AWS CloudFormation

用户指南

API Version 2010-05-15

Amazon Web Services

AWS CloudFormation: 用户指南

Amazon Web Services

Abstract

使用 AWS CloudFormation 服务预先反复创建和配置 AWS 基础设施部署。

AWS 服务或AWS文档中描述的功能，可能因地区/位置而异。点 击 Getting Started with Amazon AWS 查看适用于中国 （北京）地区的具体区别。

欢迎

*AWS CloudFormation 用户指南*介绍如何使用 AWS CloudFormation 服务。

使用 AWS CloudFormation 可预先反复创建和预配置 AWS 基础设施部署。它可以帮助您利用 AWS 产品（如 Amazon Elastic Compute Cloud、Amazon Elastic Block Store、Amazon Simple Notification Service、Elastic Load Balancing 和 Auto Scaling）构建高度可靠、高度可扩展的具有较高成本效益的应用程序，而不必担心创建和配置底层 AWS 基础设施。AWS CloudFormation 让您能够使用模板文件将资源集合作为单个单元（堆栈）来进行创建和删除。

如何... ?

如何... ?	相关部分
确定 AWS CloudFormation 是否适合我的需求？	http://amazonaws.cn/cloudformation/
快速学会使用 AWS CloudFormation？	AWS CloudFormation Getting Started Guide
学会如何执行特定任务？	修改 AWS CloudFormation 模板 (p. 183)
了解 AWS CloudFormation 如何工作？	介绍 (p. 2)
了解有关 AWS CloudFormation 模板的信息？	使用 AWS CloudFormation 模板 (p. 99)
可在我自己的模板中使用的入门模板片段？	模板代码段 (p. 128)
查看示例模板？	示例模板 (p. 111)
了解关于模板使用和修改的信息？	使用 AWS CloudFormation 模板 (p. 99)
了解 AWS CloudFormation 工具的详细信息？	AWS 命令行界面参考 (p. 523)
了解 AWS CloudFormation 示例模板的详细信息？	模板剖析 (p. 99) 和 模板参考 (p. 217)

介绍

Abstract

以堆栈（在其中可定义堆栈的特征）为单位，一起创建和删除相关 AWS 资源。

通过 AWS CloudFormation，您可以将相关 AWS 资源组合成一个称为堆栈的单元，并对其进行创建和删除操作。您可以使用模板（符合 JSON 的文本文件）来定义堆栈参数、映射、资源属性和输出值的特性。您可以通过草稿写入您的模板，或通过我们提供的示例模板之一开始此项操作。您可以将许多 AWS 产品用于 AWS CloudFormation，例如 Amazon EC2、AWS Elastic Beanstalk 和 Amazon RDS（有关完整列表，请参阅资源属性类型参考 (p. 419)）。

Topics

堆栈

Abstract

堆栈定义为 AWS 资源集合，它说明可对堆栈执行的操作。

堆栈是 AWS 资源的一个集合。通过 AWS CloudFormation，您可以利用堆栈进行如下操作：

- 使用 `aws cloudformation create-stack` 创建 AWS CloudFormation 堆栈，提供名称，以及指定用于定义堆栈的模板。

- 使用 `aws cloudformation describe-stack-events` 跟踪创建操作的进度。AWS CloudFormation 将在堆栈创建过程中优化成员资源创建顺序，由于考虑到资源的依赖关系，因此无法预测每个资源的创建顺序。使用 `aws cloudformation describe-stack-events` 命令，您可以监控进度。
- 使用 `aws cloudformation describe-stacks` 或 `aws cloudformation list-stacks` 列出正在运行的堆栈，并按特定堆栈名称或堆栈状态进行筛选。使用 `aws cloudformation describe-stacks` 将仅列出正在运行的堆栈或处于创建或删除过程中的堆栈。您可以使用 `aws cloudformation list-stacks` 列出具有任何状态的堆栈（即使该堆栈已在过去 90 天内删除），如果您需要，可以按状态进行筛选。
- 使用 `aws cloudformation describe-stack-resources` 列出堆栈内容明细。即使正在创建或删除堆栈，您仍可进行上述操作，通过此操作，您可以查看单独成员资源的状态。
- 使用 `aws cloudformation describe-stack-events` 查看堆栈生成的事件历史记录，并可以选择按特定堆栈名称进行筛选。您可以查看多达 90 天已删除堆栈的事件。
- 使用 `aws cloudformation delete-stack` 删除堆栈。当您删除堆栈时，同时也删除了其各项成员资源。与堆栈创建一样，AWS CloudFormation 也会优化删除顺序，因此顺序无法预测。您可以使用 `aws cloudformation describe-stack-events` 跟踪删除进度，并使用 `aws cloudformation list-stacks` 列出已删除的堆栈。

AWS CloudFormation 可确保所有成员资源均已根据需要创建或删除。由于 AWS CloudFormation 将堆栈成员视为单个单元来进行处理，因此，必须为要创建的堆栈成功创建所有成员。如果由于任何原因导致成员资源无法创建，则 AWS CloudFormation 将回滚堆栈，并自动删除已创建的成员资源。

Note

将按照堆栈资源运行的时间（即使您已立即删除堆栈）向您收取费用。

有关更多信息，请参阅 修改 AWS CloudFormation 模板 (p. 183)。

模板

Abstract

说明可以在源代码控制系统中编辑和管理的模板中的 AWS 基础设施要求。

您可以在模板中说明您的 AWS 基础设施要求。模板是一个文本文件，其格式应符合 JSON 格式标准。由于模板仅为文本文件，因此，您可以通过其余源代码在源控制系统中对其进行编辑和管理。有关 JSON 格式的更多信息，请访问 http://www.json.org。

在模板中，您可以声明几个主要对象：模板的格式版本 (p. 4)及其描述 (p. 4)，以及创建堆栈所需的参数 (p. 5)、映射 (p. 6)、条件 (p. 4)、资源 (p. 4)和输出 (p. 5)。格式版本、说明、参数、映像和输出都是可选项。您只需声明一种资源。下面说明了一个有效模板，其仅需声明无属性的单一资源。

```
{
    "Resources" : {
        "MyQueue" : {
            "Type" : "AWS::SQS::Queue",
            "Properties" : {
            }
        }
    }
}
```

一般情况下，资源拥有属性部分，其包含创建此资源所需的数值。如果资源无需声明任何属性，那么您可以忽略此资源的属性部分。

如需检查您的模板文件是否存在语法错误，您可以使用 `aws cloudformation validate-template` 命令。

Note

`aws cloudformation validate-template` 的设计用途为仅检查您的模板的语法。它不能保证您已对一项资源指定的属性值对于该资源有效，也不能决定创建堆栈时存在的资源数量。

要检查操作有效性，您需要尝试创建堆栈。没有用于 AWS CloudFormation 堆栈的沙盒或测试区，因此您需要在测试期间为创建的资源支付费用。

格式版本

模板格式版本指定了编写模板时依据的 AWS CloudFormation 模板版本。

Important

模板格式版本与 API 或 WSDL 版本不同。模板格式版本可独立于 API 和 WSDL 版本，进行独立更改。

可选说明

通过可选描述属性，您可以将免费有效的 JSON 文本字符串与模板相关联。您可以记录模板的描述。

可选参数

可选参数将在参数部分列出。通过参数，您可以在运行时将数值传输至您的模板，同时也可以在模板的资源和输出部分解除参考。

大多数示例模板都会声明 Parameters 部分（请参阅示例模板 (p. 111)）。参数 (p. 5)中对参数进行了更全面的说明。另外，有关 Parameters 部分格式的技术详细信息，请参阅参数声明 (p. 101)。

可选映像

通过可选映像部分，您可以声明条件值。可在 Resources 和 Outputs 部分使用内部函数 Fn::FindInMap (p. 501) 取消映射引用。

两个示例模板均声明了 Mappings 部分（请参阅示例模板 (p. 111)）。映像 (p. 6)中对映射进行了更全面的说明。另外，有关 Mappings 部分格式的技术详细信息，请参阅映射声明 (p. 104)。

可选条件

在可选的“条件”部分，您可以定义用于控制是否创建某些资源或者是否在堆栈创建或更新过程中为某些资源属性分配值的条件。例如，您可以根据堆栈是用于生产环境还是用于测试环境来按照条件创建资源。

有关定义条件的更多信息，请参阅条件 (p. 7)。

资源

资源部分将列出堆栈的成员资源。每项资源将予以分别列明，并指定创建此特定资源所必需的资源属性。可在资源和输出部分取消资源参考。它们的属性将基于文本、资源、参数、和内部函数。有关更多信息，请参阅 资源属性 (p. 5)。

所有示例模板均会声明 Resources 部分（请参阅示例模板 (p. 111)）。资源 (p. 8)中对资源进行了更全面的说明。有关 Resources 部分格式的技术详细信息，请参阅资源声明 (p. 109)。

资源属性

如果资源无需声明任何属性，那么您可以忽略此资源的属性部分。资源属性将基于文本、资源、参数、和内部函数。

大多数示例模板声明了具有一个或多个属性的资源（请参阅示例模板 (p. 111)）。资源属性 (p. 9)中对资源属性进行了更全面的说明。另外，有关 Properties 部分格式的技术详细信息，请参阅属性声明 (p. 109)。

可选输出

在 Outputs 部分，您可以选择对响应 `aws cloudformation describe-stacks` 命令而返回的自定义值进行定义。这些输出值将包括基于文本、资源、参数、虚拟参数和内部函数的信息。

所有示例模板均声明 Outputs 部分（请参阅示例模板 (p. 111)）。输出 (p. 11)中对输出进行了更全面的说明。另外，有关 Outputs 部分格式的技术详细信息，请参阅输出声明 (p. 110)。

参数

Abstract

定义可以在模板的参数部分中设置的值，可以具有默认值也可以在运行时覆盖。

AWS CloudFormation 参数是您在模板 Parameters 部分中定义的值。参数会有默认值。如果您指定参数值作为 `aws cloudformation create-stack --parameters` 选项的一部分，那么将覆盖默认值。运行时重写的参数值将作为 `aws cloudformation describe-stacks` 命令的一部分返回，除非您通过包括值为 `true` 的 NoEcho 属性将参数值隐藏在参数声明中。如果您提供 NoEcho 属性，则参数值将显示为星号 (*****)。（您未重写的参数值将不予显示。）

可将参数声明为以下任一类型：*`String`*、*`Number`* 或 *`CommaDelimitedList`*。对于具有 *`String`* 或 *`Number`* 类型的参数，您可以定义 AWS CloudFormation 用于验证参数值的约束条件。

对于 *`String`* 类型，您可以定义以下约束条件：MinLength、MaxLength、Default、AllowedValues 和 AllowedPattern。

对于 *`Number`* 类型，您可以定义以下约束条件：MinValue、MaxValue、Default 和 AllowedValues。数字可以是一个整数或一个浮点值。

有关参数约束条件的更多信息，请参阅参数声明 (p. 101)。

请注意，所有参数值将指定为模板 JSON 中的字符串。这表示“数字”参数值必须放在引号内。例如，MyNumber 的默认值将指定一个数字，且该数字放在引号内。

```
"Parameters" : {
    "MyNumber" : {
      "Type" : "Number",
      "Default" : "10",
      "MinValue" : "1"
    }
}
```

在资源和输出部分可被解除参数参考，因此，您可以使用您声明的任何参数作为资源、资源属性、参考、函数或输出值的数值。

以下示例显示了 InstanceType 参数的声明，该参数为字符串类型，仅允许枚举值 t1.micro、m1.small 和 m1.large，并且默认值为 *m1.small*。

```
"Parameters" : {
    "InstanceType" : {
      "Type" : "String",
      "Default" : "t1.micro",
      "AllowedValues" : ["t1.micro", "m1.small", "m1.large"],
      "Description" : "Enter t1.micro, m1.small, or m1.large. Default is
t1.micro."
    }
}
```

如果您在命令行重写此数值，那么您的命令可能要类似于：

```
aws cloudformation create-stack --stack-name TestStack --template-body
file:///home/local/MyTemplate.template --parameters ParameterKey=Instance
Type,ParameterValue=m1.large
```

如果具有多个参数，请用空格分隔参数/值对。例如，假设 MyTemplate.template 需要两个参数。您可能需要通过如下命令创建基于该模板的堆栈：

```
aws cloudformation create-stack --stack-name TestStack --template-body
file:///home/local/MyTemplate.template --parameters ParameterKey=MyName,Paramet
erValue=Joe ParameterKey=MyValue,ParameterValue=10
```

请注意，在这种情况下，如果您错误键入参数名称，AWS CloudFormation 将不会创建堆栈。它会报告模板不含此参数。

大多数示例模板都会声明 Parameters 部分（请参阅示例模板 (p. 111)）。另外，有关 Parameters 部分格式的技术详细信息，请参阅参数声明 (p. 101)。

映像

Abstract

使用可选的映射部分在模板中指定条件参数值。

通过映像，您可以在您的模板中指定条件参数值。用于内部函数 Fn::FindInMap (p. 501) 时，其工作方式与 Case 语句或查找表的工作方式类似。

在可选 Mappings 部分，您可以定义一个或多个*映射*。模板内，每个映像均有一个唯一的逻辑名称，其将定义一个或多个密钥属性对。每个属性必须为一条字符串或字符串列表。您不可以基于参数、或内部函数进行映射。您可以声明您所需的若干条件映射密钥，还可以声明哪项映射密钥为默认值。以下示例显示了通过四个选项声明映射的映像部分，其中区域名称将映射至特定亚马逊系统映像 (AMI) 名称：

```
"Mappings" : {
"RegionMap" : {
    "us-east-1" : {
      "AMI" : "ami-76f0061f"
    },
    "us-west-1" : {
      "AMI" : "ami-655a0a20"
```

```
    },
    "eu-west-1" : {
      "AMI" : "ami-7fd4e10b"
    },
    "ap-southeast-1" : {
      "AMI" : "ami-72621c20"
    }
  }
}
```

要为资源属性或输出分配映射属性值时，可以使用 Fn::FindInMap 函数，向其传递映射的逻辑名称、映射密钥名称和要检索的映射属性名称。通过将参数或虚拟参数指定为向 Fn::FindInMap 传递的映射密钥名称，您可以检索在运行时使用的那个属性值。

两个示例模板会声明 Mappings 部分（请参阅示例模板 (p. 111)）。另外，有关 Mappings 部分格式的技术详细信息，请参阅映射声明 (p. 104)。

条件

Abstract

通过条件部分，使用模板中定义的内部函数指定条件。

所有条件都是在模板的“条件”部分定义的。您可以使用内部函数来定义条件，如下例所示：

```
"Parameters" : {
  "EnvType" : {
    "Description" : "Environment type.",
    "Default" : "test",
    "Type" : "String",
    "AllowedValues" : ["prod", "test"]
  }
},

"Conditions" : {
  "CreateProdInstance" : {"Fn::Equals" : [{"Ref" : "EnvType"}, "prod"]}
}
```

如果参数 EnvType 等于 prod，则 CreateProdInstance 条件的计算结果为 true。参数 EnvType 是在创建或更新堆栈时指定的输入参数。

Note

在“条件”部分，您只能引用模板的“参数”和“映射”部分中的其他条件和值。例如，您不能在条件中引用资源的逻辑 ID，但可以从输入参数引用值。

若要使用条件，您需要在模板的“资源”部分引用条件，将其与特定资源相关联。然后，每当该条件的计算结果为 true 时，就会创建该资源，如下例所示：

```
"ProductionInstance" : {
  "Type" : "AWS::EC2::Instance",
  "Condition" : "CreateProdInstance",
  "Properties" : {
```

```
    "InstanceType" : "c1.xlarge",
    "SecurityGroups" : [ { "Ref" : "ProdSecurityGroup" } ],
    "KeyName" : { "Ref" : "ProdKeyName" },
    "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" },
"AMI" ]}
  }
}
```

仅当 `CreateProdInstance` 条件的计算结果为 true 时，才会创建 `ProductionInstance` 资源。

有关条件的更多信息，请参阅条件声明 (p. 107)和条件函数 (p. 491)。

伪参数

Abstract

使用虚拟参数，而不必在模板中声明这些参数，因为 AWS CloudFormation 会为您进行声明。

虚拟参数是 AWS CloudFormation 为您声明的参数。您可以使用这些参数，而无需在您的模板中声明它们。AWS CloudFormation 声明了几个虚拟参数，可供您在任何可能使用参数名称或逻辑资源名称的地方使用。

有关虚拟参数的信息，请参阅虚拟参数参考 (p. 510)。

资源

Abstract

在模板的资源部分中，您可以声明您希望 AWS CloudFormation 管理的 AWS 资源，如 Amazon EC2 实例或 Amazon S3 存储桶。所有模板都必须声明至少包含一个资源的资源部分。必须单独声明每个资源；但是，可以指定具有相同类型的多个资源。

每个资源声明包括三部分：

- 在模板中唯一的逻辑名称
- 资源类型
- 该资源的属性

可使用逻辑名称在模板的其他部分中引用资源。例如，如果要将 Amazon Elastic Block Store 映射到 Amazon EC2 实例，可引用数据块存储和实例的逻辑 ID 来指定映射。逻辑名称必须是字母数字 (A-Za-z0-9)。有关所有资源的列表，请参阅 AWS 资源类型参考 (p. 217)。

除了逻辑 ID 之外，某些资源还有物理 ID，这是分配资源的实际名称，如 Amazon EC2 实例 ID 或 Amazon S3 存储桶名称。可以使用物理 ID 标识 AWS CloudFormation 模板外部的资源，但是仅在创建了资源之后。例如，您可以为 Amazon EC2 实例资源提供逻辑 ID `MyEC2Instance`；但是在 AWS CloudFormation 创建实例时，AWS CloudFormation 自动生成物理 ID（如 `i-28f9ba55`）并将其分配给实例。您可以使用此物理 ID 标识实例，可以使用 Amazon EC2 控制台查看其属性（如 DNS 名称）。对于支持自定义名称的资源，您可以分配自己的名称（物理 ID）以帮助您快速标识资源。例如，您可以将存储日志的 Amazon S3 存储桶命名为 `MyPerformanceLogs`。有关更多信息，请参阅 名称类型 (p. 465)。

资源属性是可以对资源指定的附加选项。例如，您可以为 Amazon RDS 数据库实例指定数据库快照属性以便从快照创建数据库实例。以下示例声明 ID 为 `myLinuxBundle-2011-12-30` 的 Amazon EC2 映像：

```
"Resources" :{
    "MySimpleImage" : {
        "Type" : "AWS::EC2::Image",
        "Properties" : {
            "ImageId" : "myLinuxBundle-2011-12-30",
        }
    }
}
```

有关资源属性的更多信息，请参阅 资源属性 (p. 9)。

有关 Resources 部分格式的技术详细信息，请参阅资源声明 (p. 109)。

资源属性

Abstract

通过在所创建资源类型的资源属性部分中进行声明，来设置特定于资源的属性值。

大多数资源需要您在创建前就设定特定资源属性值。如果资源无需声明任何属性，那么您可以忽略此资源的属性部分。

资源的 Properties 部分声明的属性将特定于要创建的资源类型，并将根据所属资源进行声明（请参阅AWS 资源类型参考 (p. 217)）。

以下示例将显示名称为“MyVolume”的资源声明，其将声明三项属性：

```
Resources : {
    "MyVolume" : {
        "Type" : "AWS::EC2::Volume",
        "Properties" : {
            "Size" : "4",
            "SnapshotId" : "snap234",
            "AvailabilityZone" : "us-east-1a"
        }
    }
 }
```

资源属性的数值将基于文本、参数参考、和内部函数。

大多数示例模板声明了具有一个或多个属性的资源（请参阅示例模板 (p. 111)）。另外，有关 Properties 部分格式的技术详细信息，请参阅属性声明 (p. 109)。

参考

Abstract

使用 Ref 函数指定任何资源的逻辑名称，以解除引用另一个资源、输出、参数或内部函数的值。

使用 Ref (p. 508) 函数，您可以指定任何资源的逻辑名称，以取消对其他资源、输出、参数或内部函数的值的引用。

例如，在资源部分，您可以通过逻辑名称“HighRestriction”声明安全组资源。在另一资源声明内的其他地方，您可以使用 `"Ref" : "HighRestriction"` 作为另一资源的属性值。

在下面的示例中，参数“MyURL”使用默认字符串值“http://amazonaws.cn”进行声明。随后在 Outputs 部分中，该值将作为 `"Ref" : "MyURL"` 取消引用。

```
"Parameters" : {
    "MyURL" : {
        "Type" : "String",
        "Default" : "http://amazonaws.cn"
    },

    ...

"Outputs" : {
    "URL" : {
        "Value" : { "Ref" : "MyURL" }
    }
}
```

AWS CloudFormation 针对取消引用的对象返回的值取决于资源类型。有关每个受支持类型的特定返回值的详细信息，请参阅资源属性类型参考 (p. 419)。

大多数示例模板会使用 Ref 函数（请参阅示例模板 (p. 111)）。另外，有关 Ref 函数的技术详细信息，请参阅Ref (p. 508)。

固有功能

Abstract

使用 AWS CloudFormation 提供的内部函数，以便传递仅在运行时可用的值。

AWS CloudFormation 提供了可用于传递仅在运行时才可用的值的函数。您可以通过“Fn::*function-name*”指定内联函数，并提供该函数在内联时需要的任何参数。上述实参可以是文字字符串或字符串列表、参数参考、或从另一函数返回的数值。

在以下示例中，创建堆栈时将由 `Fn::GetAtt` 函数根据分配给 *`MyLoadBalancer`* 负载均衡器的 *`DNSName`* 值提供 URL 输出值：

```
"Outputs" : {
    "URL" : {
        "Value" : { "Fn::GetAtt" : [ "MyLoadBalancer", "DNSName" ] }
    }
}
```

当前，AWS CloudFormation 支持以下函数：

名称	目的
Fn::Base64 (p. 491)	实参 base64 编码。
Fn::FindInMap (p. 501)	从指定映像返回密钥数值。
Fn::GetAtt (p. 502)	返回指定资源的属性值。

名称	目的
Fn::GetAZs (p. 505)	获取可以创建 AWS CloudFormation 堆栈的可用区。
Fn::Join (p. 506)	第二实参元素的串联，由第一实参进行分离。
Ref (p. 508)	返回基于逻辑名称或参数的资源或数值。

多个示例模板将使用内部函数（请参阅示例模板 (p. 111)）。另外，有关内部函数格式的技术详细信息，请参阅函数声明 (p. 110)。

输出

Abstract

使用模板的输出部分声明要传递回给模板用户的信息。

您可以使用模板输出部分声明要传递回给模板用户的信息。由 `aws cloudformation describe-stacks` 命令返回输出。

您可以使用文本值或 AWS CloudFormation 函数来声明输出信息。

Outputs 部分中的信息仅由 `aws cloudformation describe-stacks` 针对现有堆栈返回。当堆栈无法创建或删除堆栈时，不会返回输出部分中声明的值。

在以下示例中，名为 *URL* 的输出返回文本值 `http://amazonaws.cn/cloudformation`。

```
"Outputs" : {
    "URL" : {
        "Value" : "http://amazonaws.cn/cloudformation"
    }
}
```

大多数示例模板会声明 Outputs 部分（请参阅示例模板 (p. 111)）。另外，有关输出格式的技术详细信息，请参阅输出声明 (p. 110)。

AWS CloudFormation 入门

如果您对于 AWS CloudFormation 相对陌生，那么本部分中的指南会帮助您快速入门，为您提供有关通过 AWS 控制台使用 CloudFormation 的基本信息，并指导您完成使用 AWS 命令行界面 (CLI) 的全过程，以便您可以从系统的命令提示符窗口管理您的 CloudFormation 堆栈。

Topics

注册 AWS 账户

您必须首先注册 AWS 账户，然后才能使用 AWS CloudFormation 或任何 Amazon Web Services。

如需注册 AWS 账户

1. 请转至 http://amazonaws.cn，然后单击“注册”。
2. 按照屏幕上的说明进行操作。

   作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

试用

Abstract

演练使用模板声明资源，然后通过 AWS CloudFormation 创建、监控和删除堆栈。

使用合适的模板，您可以一次部署应用程序所需的全部 AWS 资源。在本节中，您将检查为 WordPress 博客声明资源的模板，创建一个 WordPress 博客作为堆栈，监视堆栈创建过程，检查堆栈上的资源，然后删除该堆栈。您需要使用 AWS Management Console完成这些任务。

步骤 1：注册服务。

注册 AWS CloudFormation 也将自动注册所需的其他 AWS 产品，如 Amazon Elastic Compute Cloud、Amazon Relational Database Service 和 Amazon Simple Notification Service。您未使用的服务，将不会向您收取任何费用。

Note

AWS CloudFormation 是一种免费服务；不过，您的堆栈中包含的 AWS 资源会按每种资源的当前费率计费。有关 AWS 定价的详细信息，请参阅 http://amazonaws.cn 上每种产品的详细信息页。

注册 AWS CloudFormation

1. 转至 http://amazonaws.cn/cloudformation，然后单击 Sign Up for AWS CloudFormation (注册 AWS CloudFormation)。
2. 按照屏幕上的说明进行操作。

如果您还没有 AWS 账户，系统会提示您在注册 AWS CloudFormation 时创建一个。

作为注册流程的一部分，您会收到一个电话，需要您使用电话键盘输入一个 PIN 码。

步骤 2：选取模板

接下来，您需要一个说明在您的堆栈中所需资源的模板。本步骤中，您可以使用一个已经准备就绪的示例模板。该示例模板使用单一 Amazon EC2 实例和 Amazon RDS 数据库实例创建基本 WordPress 博客。该模板还创建 Amazon EC2 和 Amazon RDS 安全组来控制 Amazon EC2 实例和数据库实例的防火墙设置。

Important

AWS CloudFormation 是免费的，但 AWS CloudFormation 创建的 AWS 资源是实时的（不在沙盒中运行）。您将为这些资源承担标准使用费，直到您在本教程的最后一项任务中终止这些资源为止。总费用将应该最少的。有关如何最大限度地降低费用的信息，请转至 http://amazonaws.cn/free/。

查看模板

- 您可以从以下网址下载或查看 WordPress 示例模板：https://s3.amazonaws.com/cloudformation-templates-us-east-1/WordPress_Single_Instance_With_RDS.template。

  如果您想检查模板，则无需下载。本指南中，您将稍后会使用模板 URL。

模板是一个 JavaScript Object Notation (JSON) 文本文件，它包含有关要在堆栈中创建的 AWS 资源的配置信息。当您浏览示例 WordPress 模板时，您会看到六个顶级对象：AWSTemplateFormatVersion、Description、Parameters、Mappings、Resources 和 Outputs；但是，只有 Resources 对象是必需的。

Resources 对象包含要使用该模板创建的 AWS 资源的定义。每一种资源都单独列出并指定了创建该特定资源的必需属性。模板中的以下资源声明包含 Amazon RDS 数据库实例的配置，该配置在本示例中的逻辑名称为 DBInstance：

```
"Resources" : {
```

```
  ...
  "DBInstance" : {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName"            : { "Ref" : "DBName" },
      "Engine"            : "MySQL",
      "MasterUsername"    : { "Ref" : "DBUsername" },
      "DBInstanceClass"   : { "Ref" : "DBClass" },
      "DBSecurityGroups"  : [{ "Ref" : "DBSecurityGroup" }],
      "AllocatedStorage"  : { "Ref" : "DBAllocatedStorage" },
      "MasterUserPassword": { "Ref" : "DBPassword" }
    }
  },

  "DBSecurityGroup": {
    "Type": "AWS::RDS::DBSecurityGroup",
    "Properties": {
      "DBSecurityGroupIngress": { "EC2SecurityGroupName": { "Ref": "WebServer
SecurityGroup"} },
      "GroupDescription"      : "Frontend Access"
    }
  },
  ...
},
```

如果您之前已经创建了 DB 实例，那么您将需要识别出该实例的属性，例如，引擎、DBInstanceClass 和 AllocatedStorage，这些属性决定了 DB 实例的配置。资源声明是立即指定这些所有配置设置的一种有效方法。当您将资源声明放入一个模板时，您将可以使用该模板轻易的创建和配置所有的声明资源，从而创建一个堆栈。想要启动资源的相同配置时，您所要做的是使用相同的模板创建一个新的堆栈即可。

资源声明开始于一个字符串，该字符串指定了资源的逻辑名称。如您所见，该逻辑名称可用于指代模板内的资源。

使用 *Parameters* 对象可以声明在创建堆栈时可传递给模板的值。参数是指明敏感信息的一种有效手段，这些敏感信息包括用户名和密码一类的和您不想存储在模板内部的信息。此外，它也是指定对特定应用程序可能具有唯一性的信息，或者您正在部署的配置（例如，一个域名或实例类型）的一种方法。当您在本节后面的步骤中创建 WordPress 堆栈时，您将看到模板中的参数设置在创建堆栈向导的指定参数页面显示出来，该页就是您在创建堆栈前可以指定参数的页面。

在模板中，以下参数用于指定在 Amazon RDS 数据库实例资源的属性中使用的值：

```
"Parameters" : {
  ...
  "DBName" : {
    "Default": "wordpress",
    "Description" : "The WordPress database name",
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "64",
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*",
   "ConstraintDescription" : "must begin with a letter and contain only alpha
numeric characters."
  },

  "DBUsername" : {
```

```
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "The WordPress database admin account user name",
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "16",
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*",
   "ConstraintDescription" : "must begin with a letter and contain only alpha
numeric characters."
  },

  "DBPassword" : {
    "Default": "admin",
    "NoEcho": "true",
    "Description" : "The WordPress database admin account password",
    "Type": "String",
    "MinLength": "1",
    "MaxLength": "41",
    "AllowedPattern" : "[a-zA-Z0-9]*",
    "ConstraintDescription" : "must contain only alphanumeric characters."
  },

  "DBAllocatedStorage" : {
    "Default": "5",
    "Description" : "The size of the database (Gb)",
    "Type": "Number",
    "MinValue": "5",
    "MaxValue": "1024",
    "ConstraintDescription" : "must be between 5 and 1024Gb."
  },
  ...
},
```

在 DBInstance 资源声明中，DBName parameter 参数将指定 DBName 属性：

```
  "DBInstance" : {
    "Type": "AWS::RDS::DBInstance",
    "Properties": {
      "DBName" : { "Ref" : "DBName" },
      ...
    }
  },
```

大括号内包含了使用 DBName 作为其输入对 Ref (p. 508) 函数的一次调用。Ref 函数将返回它指代的对象的值。在这种情况下，创建堆栈后，WordPressDBName 参数和 Ref 函数将为 DBName 指定值设定 DBName 属性。

Ref 函数也可以将某个资源的属性设置为另一个资源的属性值。例如，资源声明 DBInstance 包含了下列属性声明：

```
"DBInstance" : {
  "Type": "AWS::RDS::DBInstance",
  "Properties": {
    ...
```

```
    "DBSecurityGroups"  : [{ "Ref" : "DBSecurityGroup" }],
    ...
  }
},
```

DBSecurityGroups 属性将获取 Amazon RDS 数据库安全组的列表。Ref 函数将有 DBSecurityGroup 的输入，该输入为模板中一个数据安全组的逻辑名称，并将 DBSecurityGroup 的名称添加至 DBSecurityGroups 属性。

在模板中，您还会看到 *Mappings* 对象。您将使用映射来声明条件值，而该值会在一个相似的方式中作为转换语句接受评估。在实例类型中，模板将使用映射为区域和架构类型选择正确的亚马逊系统映像 (AMI)。*输出* 定义由 `aws cloudformation describe-stacks` 命令返回的自定义值，这些值位于创建堆栈后 AWS Management Console的“输出”选项卡上。您可以使用输出值来返回从堆栈中的资源得到的信息，例如，在模板中为网站创建的 URL。我们将在了解模板基础知识 (p. 19)中更加详细地介绍映射、输出和其他有关模板的信息。

现在关于模板的信息已经足够了。让我们开始创建一个堆栈吧。

步骤 3：确保您已准备好堆栈所需的所有项目

在您从一个模板创建一个堆栈之前，您必须确保模板所需的所有独立资源都是可用的。模板可以使用或引用现有 AWS 资源以及在模板中声明的资源。AWS CloudFormation 检查对模板中资源的引用，还检查对现有资源的引用，以确保这些资源存在于要创建堆栈的区域中。如果您的模板指代的是一个不存在的独立资源，那么堆栈的创建将会失败。

示例 WordPress 模板包含一个输入参数 KeyName，该参数指定在模板中声明的 EC2 实例所使用的密钥对。模板取决于从模板中创建一个堆栈的用户是否为 KeyName 参数提供一个有效的密钥对。如果您提供了一个有效的密钥对名称，则该堆栈将会创建成功。如果您未提供一个有效的密钥对名称，则该堆栈将会回滚。

在您创建堆栈 *前*，请确保您拥有有效的 EC2 密钥对，并记录下密钥对名称。

若要查看密钥对，请打开 Amazon EC2 控制台，然后在导航窗格中单击 Key Pairs (密钥对)。

Note

如果您没有一个可使用的 EC2 密钥对，那么您必须在您即将创建堆栈的区域内创建一个密钥对。有关创建密钥对的信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的获取 SSH 密钥对。

既然您已经拥有一个可用的密钥对，那就让我们使用 WordPress 模板来创建一个堆栈吧。

步骤 4：创建堆栈

您将基于之前讨论的 *WordPress-1.0.0* 文件来创建您的堆栈。此模板包含多个 AWS 资源，包括一个 Amazon Relational Database Service 数据库实例和一个 Web 服务器。

要创建 WordPress 堆栈

1. 登录 AWS 管理控制台，并通过以下网址打开 AWS CloudFormation 控制台：https://console.amazonaws.cn/cloudformation/。
2. 如果这是新的 AWS CloudFormation 账户，请单击 Create New Stack (创建新堆栈)。否则，请单击 Create Stack (创建堆栈)。
3. 在 Stack Name (堆栈名称) 框中，键入堆栈名称。在本示例中，请使用 MyWPTestStack。堆栈名中不得含有空格。

4. 选择 Provide an S3 URL (提供模板的 S3 URL)。在下面的框中，键入或粘贴示例 WordPress 模板的 URL，然后单击 Continue (继续)：

   https://s3.amazonaws.com/cloudformation-templates-us-east-1/WordPress_Single_Instance_With_RDS.template

   Note

   存储在 Amazon S3 存储段中的 AWS CloudFormation 模板必须可供创建堆栈的用户访问，并且必须位于与创建的堆栈*相同的地区*。因此，如果 Amazon S3 存储桶位于 us-east-1 区域，堆栈也必须在 us-east-1 区域中创建。

5. 在 KeyName 框中，输入要创建堆栈的同一区域中的有效 Amazon EC2 密钥对名称。

   Note

   在 Specify Parameters (指定参数) 页上，您将确认模板的 Parameters 对象中的参数。

6. 单击 Next Step (下一步)。
7. 在这种情况下，我们不会添加任何标签。单击 Next Step (下一步)。作为密钥值对的标签可帮助您识别堆栈。有关更多信息，请参阅向 AWS CloudFormation 堆栈添加标签。
8. 审核堆栈信息。如果满意该设置，则单击 Create (创建)。

创建堆栈可能需要几分钟时间 — 不过您可能不想坐等消磨时间。如果您喜欢我们的产品，那么您可能会想知道堆栈的创建是如何进行的。

步骤 5：监控堆栈创建的进展

在您完成了“创建堆栈”向导后，AWS CloudFormation 将开始创建模板中指定的资源。您的新堆栈 (MyWPTestStack) 将会在 CloudFormation 控制台顶部的列表中出现。它的状态为 CREATE_IN_PROGRESS。您可以通过查看事件了解堆栈的详细状态。

想要查看堆栈的事件

1. 在 AWS CloudFormation 控制台上，选择列表中的堆栈 MyWPTestStack。
2. 在列表下的窗格中，单击 Events (事件) 选项卡。

   控制台每 60 秒自动使用最新事件刷新事件列表。

Events (事件) 选项卡显示堆栈创建过程中的每个重要步骤（按每个事件的时间排序，最新事件位于最上面）。

第一个事件（在事件列表最底部）为堆栈创建过程的开始：

```
MyWPTestStack AWS::CloudFormation::Stack CREATE_IN_PROGRESS
```

下面是标志所有资源创建开始和完成的事件。例如，DBSecurityGroup 安全组创建后会产生以下条目：

```
2013-04-24 18:59 UTC-7 | PDT AWS::RDS::DBSecurityGroup ... CREATE_COMPLETE

2013-04-24 18:54 UTC-7 | PDT AWS::RDS::DBSecurityGroup ... CREATE_IN_PROGRESS
```

当 AWS CloudFormation 报告它已开始创建资源时，将记录 CREATE_IN_PROGRESS 事件。当资源被成功创建时，CREATE_COMPLETE 事件将被记录下来。

当 AWS CloudFormation 已成功创建堆栈时，您将会在 Events (事件) 选项卡顶部看到以下事件：

```
MyWPTestStack AWS::CloudFormation::Stack CREATE_COMPLETE
```

如果 AWS CloudFormation 无法创建资源，它将报告 CREATE_FAILED 事件并在默认情况下回滚该堆栈。Reason (原因) 列将显示导致失败的问题。例如，如果指定无效数据库密码，对于 `AWS::RDS::DBInstance` 资源，则会出现如下事件：

```
2013-04-24 19:01 UTC-7 | AWS::RDS::DBInstance | ... CREATE_FAILED | The parameter MasterUserPassword is not a valid password because it is shorter than 8 characters.
```

步骤 6：使用您的堆栈资源

当堆栈 MyWPTestStack 具有 CREATE_COMPLETE 状态时，AWS CloudFormation 已完成创建堆栈，并且您可以开始使用它的资源了。

示例 WordPress 堆栈将创建一个 WordPress 网站。您可以通过运行 WordPress 安装脚本来继续执行 WordPress 安装。

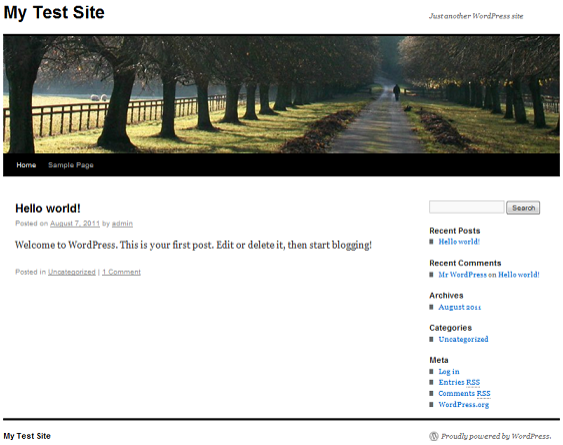
要完成 WordPress 的安装

1. 在 Outputs (输出) 选项卡上的 InstallURL 行、Value (值) 列中，单击该链接。

   InstallURL 输出值就是您使用堆栈创建的 WordPress 网站安装脚本的 URL。
2. 在 WordPress 安装的网页上，请按照屏幕上显示的指示来完成 WordPress 安装。有关安装 WordPress 的更多信息，请参阅 http://codex.wordpress.org/Installing_WordPress。

3. 返回到 AWS Management Console。在 Outputs (输出) 选项卡上的 WebsiteURL 行、Value (值) 列中，单击该链接。

如果显示了您使用此堆栈创建的 WordPress 博客网页，则表明您已使用 AWS CloudFormation 模板成功创建了 WordPress 博客。



步骤 8：清除

您已完成 AWS CloudFormation 入门任务。为了保证您不为不需要的服务承担费用，您可以通过删除堆栈和它的资源进行清理。

想要删除堆栈和它的资源

1. 在 AWS CloudFormation 控制台上，选择 `MyWPTestStack` 堆栈。
2. 单击 Delete Stack (删除堆栈)。
3. 在出现的确认消息中，单击 Yes, Delete (是，请删除)。

MyWPTestStack 的状态将会变为 DELETE_IN_PROGRESS。按照您监控堆栈的创建的同样方式，使用“事件”选项卡监控删除堆栈过程。当 AWS CloudFormation 完成删除堆栈后，它会将该堆栈从列表中移除。

恭喜您！您已经成功选取了一个模板，创建了一个堆栈，已经查看并使用了它的资源，并删除了该堆栈和它的资源。不仅如此，您还可以使用 AWS CloudFormation 模板建立 WordPress 博客。您可以在 AWS CloudFormation 示例模板库中找到其他模板。

现在，您应该了解更多有关模板的信息，以便您可以轻松地修改现有模板或创建自己的模板：了解模板基础知识 (p. 19).

了解模板基础知识

Abstract

了解模板基础知识：如何使用模板的资源、参数和其他组件，以及它们如何协作。

Topics

- 后续步骤 (p. 28)

在试用 (p. 12)中，您已学习了如何使用模板创建堆栈。您已对模板中陈述的资源有了简单的了解，并明白了它们是如何映射到堆栈中资源。我们还涉及到了输入参数，以及当您在模板中创建一个堆栈时，这些参数如何能使您传入特定值。在本节中，我们将对资源和参数进行更深入的了解。我们也将介绍模板的其他组件，以便让您了解如何综合使用这些组件来创建能产生您想要的 AWS 资源的模板。

AWS CloudFormation 模板是什么？

在我们进行进一步介绍之前，我们应该介绍一下模板的基础知识。模板是对组建堆栈的 AWS 资源的声明。模板是以一个文本文件形式保存的，其格式符合 JavaScript 对象符号 (JSON) 标准。鉴于只是文本文件，您可以在任何文本编辑器中创建编辑它们，亦可在源控制系统中用其余源代码对之进行管理。有关 JSON 格式的更多信息，请访问 http://www.json.org。

在模板中，可以使用 AWS CloudFormation 可解释的 JSON 结构声明要创建和配置的 AWS 资源。在 JSON 格式中，一个对象可被命名为值对或一组用大括号括起来的子对象名字的配对。通过逗号分隔多个同层级对象。AWS CloudFormation 模板以一个左大括号开始，以右大括号结束。在这些大括号内，您可以声明六个顶级 JSON 对象：AWSTemplateFormatVersion (p. 4)、Description (p. 4)、Parameters (p. 5)、Mappings (p. 6)、Resources (p. 4) 和 Outputs (p. 5)。唯一需要的顶级对象就是资源对象，它至少须声明一种资源。让我们先从最基础的模板开始，该模板只包含一个单一资源声明的资源对象。

资源：存储桶，你好！

该资源对象包含在大括号内的一系列资源对象。资源声明包含资源的属性，其本身声明为子对象。资源必须具有 *Type* 属性，该属性定义了要创建的 AWS 资源的类别。*Type* 属性具有特殊格式：

```
AWS::ProductIdentifier::ResourceType
```

例如，Amazon S3 存储桶的资源类型是 AWS::S3::Bucket (p. 402)。有关资源类型完整列表的信息，请参阅模板参考 (p. 217)。

让我们先来了解一下最基础的模板。以下模板声明了一个 AWS::S3::Bucket 类型的资源：其名称为 HelloBucket。

```
{
    "Resources" : {
        "HelloBucket" : {
            "Type" : "AWS::S3::Bucket"
        }
    }
}
```

句法元素是用引号括起来的字符串。如果您使用此模板创建堆栈，则 AWS CloudFormation 将创建一个 Amazon S3 存储桶。创建存储桶是很简单的，因为 AWS CloudFormation 可以使用默认设置创建存储桶。对于其他资源（如 Auto Scaling 组或 EC2 实例），AWS CloudFormation 需要更多信息。资源声明使用 *Properties* 属性来指定用于创建资源的信息。

根据资源类型，一些属性（例如 AWS::EC2::Instance (p. 272) 资源的 ImageId 属性）是必需的，而其他属性是可选的。有些属性将有默认值，如 AWS:: S3:: 存储桶资源的“AccessControl”属性，因此指定这些属性值是可选的。其他属性不是必选的，但可以添加您想要的功能，如 AWS:: S3:: 存储桶资源的 WebsiteConfiguration 属性。根据您的需要，此类属性值的指定完全是可选的。在上面的例子中，因为 AWS:: S3:: 存储桶资源只有可选属性，但我们并不需要任何可选功能，因此我们接受默认值忽略 Properties 属性。

要查看每个资源类型的属性，请参阅资源属性类型参考 (p. 419)中的主题。

资源属性和共同使用资源

通常情况下，资源属性只是一个字符串值。例如，以下模板为存储桶的 AccessControl 属性指定了一个已存的 ACL (PublicRead)。

```
{
    "Resources" : {
        "HelloBucket" : {
            "Type" : "AWS::S3::Bucket",
            "Properties" : {
               "AccessControl" : "PublicRead"
            }
        }
    }
}
```

有些资源可以有多个属性，而有些属性可以有一个或多个子属性。例如，AWS::S3::Bucket (p. 402) 资源有两个属性：AccessControl 和 WebsiteConfiguration。WebsiteConfiguration 属性有 IndexDocument 和 ErrorDocument 两个子属性。以下模板将展示我们的具有附加属性的原有存储桶资源。

```
{
    "Resources" : {
        "HelloBucket" : {
            "Type" : "AWS::S3::Bucket",
            "Properties" : {
               "AccessControl" : "PublicRead",
               "WebsiteConfiguration" : {
                    "IndexDocument" : "index.html",
                    "ErrorDocument" : "error.html"
               }
            }
        }
    }
}
```

请注意 AccessControl 和 WebsiteConfiguration、IndexDocument 和 ErrorDocument 这些同层级属性是如何用逗号进行分隔的。在一个模板中最常见的一种句法错误就是在同层级属性声明之间、各资源之间缺少逗号。

模板和 AWS CloudFormation 的最大优势之一就是能够创建一组资源，这些资源能一起工作，共同创建一个应用程序或解决方案。模板内的资源所使用的名称是一个逻辑名称。在 AWS CloudFormation 创建资源时，会产生一个基于其逻辑名称、堆栈名称和唯一 ID 的组合的实体名称。

您可能会想知道如何根据该名称或另一种资源的属性在一种资源上设置属性。例如，您可以创建受 S3 存储桶支持的 CloudFront 分配或 使用 EC2 安全组的 EC2 实例，所有这些资源可以在同一模板中创建。AWS CloudFormation 提供许多内部函数，您可以使用这些函数来引用其他资源及其属性。您可以使用 Ref 函数 (p. 508)引用资源的标识属性。通常，这是资源的实体名称；但是，有时它可以是标识符，例如，AWS::EC2::EIP (p. 269) 资源的 IP 地址或 Amazon SNS 主题的亚马逊资源名称 (ARN)。有关 Ref 函数返回的一系列值的信息，请参阅 Ref 函数 (p. 508)。以下模板包含一个 AWS::EC2::Instance (p. 272) 资源。该资源的 SecurityGroups 属性调用了 Ref 函数，以便能参阅 AWS::EC2::SecurityGroup 资源，即 InstanceSecurityGroup。

```
{
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "KeyName" : "mykey",
        "ImageId" : ""
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  }
}
```

可能您已经注意到了，就像JSON对象一样，Ref函数调用是用由冒号隔开、并用括号括起来的名值对表达的。该函数名即是该名称，且其输入参数是该值。您还会注意到，函数调用也是用方括号括起来的。在JSON中，列表是用方括号括起来的。SecurityGroups属性是一个安全组列表，且在本示例中，此列表只有一个项目。在 SecurityGroup 的属性列表中，以下模板有一个附加项目。

```
{
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" }, "MyExisting
SecurityGroup" ],
        "KeyName" : "mykey",
        "ImageId" : "ami-7a11e213"
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  }
}
```

MyExistingSecurityGroup 是指代现存 EC2 安全组，而不是模板中声明的安全组一个字符串。您可以使用文字字符串来指代现存 AWS 资源。

在上例中，AWS::EC2::Instance (p. 272) 的 KeyName 属性是文字字符串 mykey。这就意味着，名为 mykey 的密钥对必须位于正在创建堆栈的区域；否则，由于不存在密钥对，堆栈创建将失败。您所使用的密钥对会因您正在创建堆栈区域的不同而不同，或是您想与其他人共享该模板，使他们能与其 AWS 账户共用。如果是这样的话，在用户创建堆栈时您可以指定密钥对的名称。Ref 函数可指代那些在创建堆栈时指定的输入参数。以下模板添加了一个包含 KeyName 参数的参数对象，其中，KeyName 参数用于指定 AWS::EC2::Instance 资源的 KeyName 属性。

```
{
  "Parameters" : {
    "KeyName" : {
      "Description" : "The EC2 Key Pair to allow SSH access to the instance",
      "Type" : "String"
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" }, "MyExisting
SecurityGroup" ],
        "KeyName" : { "Ref" : "KeyName"},
        "ImageId" : "ami-7a11e213"
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  }
}
```

如果该参数或为资源返回的值正是您需要的，该 Ref 函数是很方便的，但可能还需要其他的资源属性。例如，如果要创建一个具有 S3 起始地址的 CloudFront 分配，就需要使用 DNS 样式地址指定存储桶位置。许多资源都具有附加属性，您可以在模板中使用这些属性的值。要获得这些属性，请使用Fn::GetAtt (p. 502) 函数。以下模板创建了一个 CloudFront 分配资源，通过使用 Fn::GetAtt 函数，该资源指定了 S3 存储桶资源的 DNS 名，以便能获得 S3 存储桶的 DomainName 属性。

```
{
    "Resources" : {
        "myBucket" : {
            "Type" : "AWS::S3::Bucket"
        },
        "myDistribution" : {
           "Type" : "AWS::CloudFront::Distribution",
           "Properties" : {
```

```
              "DistributionConfig" : {
                   "S3Origin" : {
                       "DNSName": {"Fn::GetAtt" : ["myBucket", "DomainName"]}
                   }
               }
           }
       }
   }
}
```

Fn::GetAtt 函数有两个参数，即资源的逻辑名和要检索的属性名。有关资源的可用属性的完整列表，请参阅 Fn::GetAtt (p. 502)。您将发现一个阵列中 FN::Getatt 函数将列出其两个参数。对于采用多个参数的函数，可以使用阵列来指定其参数。

使用输入参数的接收用户输入

迄今为止，您已经了解了这些资源和一点点有关如何在一个模板内共用这些资源的信息。您已经学习了如何引用输入参数，但我们还未深入研究过如何定义这些输入参数本身。让我们先来了解一下参数声明，以及如何限制并验证用户输入。

声明模板参数对象中的那些参数。一个参数包含一系列属性，这些属性定义了其值但也限制了其值。唯一需要的属性就是类型，它可以是字符串、数字或 CommaDelimitedList。您还可以添加一个说明属性，该属性会告诉用户更多有关应指定何种值的信息。当用户使用创建堆栈向导中的模板时，参数的名称和说明会出现在指定参数页面。

下列模板片段是参数对象，这些对象声明了在上述指定参数页面中使用的参数。

```
  "Parameters": {
    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access
into the WordPress web server",
      "Type": "String"
    },
    "WordPressUser": {
      "Default": "admin",
      "NoEcho": "true",
      "Description" : "The WordPress database admin account user name",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "16",
      "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"
    },
    "WebServerPort": {
      "Default": "8888",
      "Description" : "TCP/IP port for the WordPress web server",
      "Type": "Number",
      "MinValue": "1",
      "MaxValue": "65535"
    }
  },
```

KeyName 参数属于字符串类型，且有一个对应说明。您会注意到 KeyName 并没有默认属性，而其他参数却有。由于 KeyName 没有默认值，因此必须在创建堆栈时指定它：KeyName 没有值的情况下，AWS CloudFormation 将不会创建堆栈。当用户使用了创建堆栈向导中的模板，那些没有默认值的参数必须有

指定的值，否则，该向导会在该参数的旁边显示一个遗漏值警告，直到您指定一个值才会让您继续后面的操作。

省略默认属性会要求用户指定参数值，然而，要求用户输入一个值并不能保证该值是有效的。为了验证该参数值，您可以声明一些约束条件。

对于 *String* 类型，可以使用以下属性来声明约束条件：MinLength、MaxLength、Default、AllowedValues 和 AllowedPattern。在上例中，WordPressUser 参数具有三个约束条件：参数值的长度必须是 1 到 16 个字符（MinLength、MaxLength），参数值必须以字母开头，该字母后跟字母和数字的任意组合 (AllowedPattern)。

对于 *Number* 类型，您可以声明以下约束条件：MinValue、MaxValue、Default 和 AllowedValues。数字可以是一个整数或一个浮点值。 在上述例子中，WebServerPort 参数必须是一个介于 1 至 65 535（包含 65 535）之间的一个数 (MinValue，MaxValue)。

在本节之前，我们提到，参数是一个指定敏感或实现特定数据的很好的方式，如您需要使用但却不想嵌入模板本身的密码或用户名。对于敏感信息，您可以使用 NoEcho 属性让参数值不显示在控制台、命令行工具或 API 中。如果将 NoEcho 属性设置为 `true`，则将以星号 (*****) 的形式返回参数值。在上述例子中，WordPressUser 参数值对任何查看堆栈设置的任何人都是不可见的，且值都是以星号形式发回。

使用映射指定条件型值

参数是一个能帮助客户指定在堆栈资源属性中用到的独立或敏感值的绝佳方法，然而，可能会有一些视区域而定的设置，或是一些因其他条件或依赖关系而需要用户弄清的复杂设置。在这些情况下，您会需要把一些逻辑放入模板本身中，以便用户可以指定简单值（或根本没有）来获得他们希望得到的结果。在前面的示例中，我们为 EC2 实例的 ImageId 属性硬编码了 AMI ID。AMI ID 在美国东部区域操作正常，它代表了我们想要的 AMI。然而，如果用户尝试在不同区域的创建堆栈，那么他或她将会得到一个错误的 AMI 或根本得不到 AMI。（AMI ID 对一个区域来说是唯一的，因此在一个不同的区域，同一个 AMI ID 可能并不代表任何 AMI 或一个完全不同的 AMI。）

若要避免这个问题，您需要（在本示例中，创建堆栈的区域）通过一种方式根据条件型输入指定正确 AMI ID 。有两种模板功能能起到作用，即映射对象和 AWS:: 区域虚拟参数。

AWS::Region 虚拟参数是 AWS CloudFormation 可解析为创建堆栈的区域的值。创建堆栈时，由 AWS CloudFormation 解析虚拟参数。映射可让您把一个输入值用作确定另一个值的条件。类似于交换语句，一个映射将把一组值与另一组值联系起来。AWS:: 区域参数与映射相结合使用，您能确保为该区域指定一个适当的 AMI ID。以下模板包含一个名为 RegionMap 的映射对象，该对象被用于把 AMI ID 映射到合适的区域。

```
{
  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the instance",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
          "AMI" : "ami-76f0061f"
      },
      "us-west-1" : {
          "AMI" : "ami-655a0a20"
      },
      "eu-west-1" : {
```

```
          "AMI" : "ami-7fd4e10b"
      },
      "ap-southeast-1" : {
          "AMI" : "ami-72621c20"
      },
      "ap-northeast-1" : {
          "AMI" : "ami-8e08a38f"
      }
    }
  },

  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
 }, "AMI" ]},
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  }
}
```

在 RegionMap 中，每个区域都会被映射到一个名值对中。名值对是一个标签，该值是用来映射的。在 RegionMap 中，AMI 是该标签，AMI ID 是该值。要使用映射返回值，请使用 Fn::FindInMap (p. 501) 函数传递映射的名称、用于查找映射值的值，以及要返回的映射值的标签。在上述示例中，资源 Ec2Instance 的 ImageId 属性通过把 RegionMap 指定为要使用的映射，把 AWS:: 区域指定为开始映射的输入值，把 AMI 指定为映射到的需要标识的标签，使用了 Fn::FindInMap 函数来确定其值。例如，如果该模板用于创建 us-west-1 区域的堆栈，ImageId 会被设置为 ami-655a0a20。

Tip

AWS:: 区域虚拟参数能使您获得创建堆栈的区域。有些资源（例如 AWS::EC2::Instance (p. 272)、AWS::AutoScaling::AutoScalingGroup (p. 219) 和 AWS::ElasticLoadBalancing::LoadBalancer (p. 337)）具有一个用于指定可用区的属性。可以使用 Fn::GetAZs 函数 (p. 505)来获取一个区域内所有可用区的列表。

已构建的值和输出值

创建堆栈时，参数和映射是一种传递或确定特定值的极好的方式，但可能存在一些情况，例如参数值或其他资源属性值仅仅是您所需要的值的一部分。例如，在以下 WordPress 模板的片段中，通过并置 WebServerPort 参数和其他文字字符串，Fn::加入函数为 ElasticLoadBalancer 资源的 HealthCheck 属性构建了目标子属性，以便形成所需值。

```
  "Resources" : {
    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "Instances" : [ { "Ref" : "Ec2Instance1" },{ "Ref" : "Ec2Instance2" }
],
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
```

```
          "InstancePort" : { "Ref" : "WebServerPort" },
          "Protocol" : "HTTP"
        } ],
        "HealthCheck" : {
          "Target" : { "Fn::Join" : [ "", ["HTTP:", { "Ref" : "WebServerPort"
}, "/"]]},
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
      }
    },
```

Fn::加入函数有两个参数，一个是把您想要并置的值分开的分隔符，另一个是按您想要这些值出现的顺序排列的阵列。在上述示例中，Fn::加入函数指定了一个空字符串作为分隔符和 HTTP:，WebServerPort 参数的值，以及一个 / 字符被作为要并置的值。如果 WebServerPort 有一个值为 8888，该目标属性就会被设置为以下值：

```
HTTP:8888/
```

Fn::加入函数对于声明该堆栈的输出值来说也是很有用的。该模板中的输出对象包含您想要使之在堆栈创建后可用的值的声明。输出是一个用于捕获有关您的资源或输入参数的重要信息的便捷方式。例如，在 WordPress 模板中，我们声明了以下输出对象。

```
  "Outputs": {
    "InstallURL": {
      "Value": {
        "Fn::Join": [
          "",
          [
            "http://",
            {
              "Fn::GetAtt": [
                "ElasticLoadBalancer",
                "DNSName"
              ]
            },
            "/wp-admin/install.php"
          ]
        ]
      },
      "Description" : "Installation URL of the WordPress website"
    },
    "WebsiteURL": {
      "Value": {
        "Fn::Join": [
          "",
          [
            "http://",
            {
              "Fn::GetAtt": [
                "ElasticLoadBalancer",
                "DNSName" ]
            }
```

```
          ]
        ]
      }
    }
  }
```

每一个输出值都有一个名称，一个包含该返回值声明的值属性被作为输出值，和一个视需要而定的该值的说明。在前面的例子中，InstallURL 是一个由一个 Fn::加入函数调用返回的字符串，该字符串并置了 http://，资源 ElasticLoadBalancer 的 DNS 名，和 /wp-admin/install.php。该输出值可能会与以下值相似：

```
http://mywptests-elasticl-1gb51l6sl8y5v-206169572.us-east-1.elb.amazonaws.com/wp-admin/install.php
```

在入门教程中，我们使用此链接很方便地转到了已创建的 WordPress 博客的安装页面。AWS CloudFormation 在完成创建堆栈后生成了这些输出值。您可以在 AWS CloudFormation 控制台的“输出”选项卡中查看这些输出值，也可以使用 aws cloudformation describe-stacks 命令查看。

后续步骤

我们刚刚详细介绍了一个模板的基本部分，以及如何使用它们。您已了解了有关模板的以下信息：

- 声明资源及其属性
- 请参考具有 Ref 函数的其他资源和使用 Fn::GetAtt 函数的资源属性
- 使用参数来使用户在创建堆栈期间指定这些值，使用约束条件来验证参数输入
- 使用映射来确定条件值。
- 根据参数、资源属性和其他字符串，使用 Fn::加入函数来构建值。
- 使用基于值的输出来获得有关堆栈资源的信息。

我们还未介绍模板中的两个顶级对象：AWSTemplateFormatVersion 和 Description。AWSTemplateFormatVersion 只是模板格式的版本，如果您不指定，AWS CloudFormation 将使用最新版本。该说明是所有有效的 JSON 字符串，且会在创建堆栈向导的指定参数页面出现。有关更多信息，请参阅 模板格式版本声明 (p. 101) 和 模板描述声明 (p. 101)。

当然，还有很多高级的模板和堆栈功能。这里是一些您也许想要了解的重要模板和堆栈功能的列表：

可用于任何资源的 *可选属性*：

- DependsOn 属性 (p. 486)使您可以指定一个资源必须在创建另一个资源之后创建。
- DeletionPolicy 属性 (p. 485)使您可以指定 AWS CloudFormation 应如何处理资源的删除。
- Metadata (p. 488) 属性使您可以为资源指定结构化数据。

AWS::CloudFormation::Stack (p. 250) 使您可以在模板中将另一个堆栈作为资源嵌套。

演练：更新堆栈

Abstract

演练使用 AWS CloudFormation 更新正在运行的堆栈的简单进展。

借助 AWS CloudFormation，您可以更新现有堆栈中的资源的属性。更改范围可包括从更新 CloudWatch 警报上的警报阈值等简单配置更改到更新 Amazon EC2 实例上运行之亚马逊机器映像 (AMI) 等较复杂的更改。模板中的很多 AWS 资源都可进行更新，我们会继续添加对更多资源的支持。

此部分将带您逐步了解更新正在运行堆栈的简单进展。此部分将向您展示，模板的使用怎样将版本控制系统像用于您正在运行的软件那样用于您的 AWS 基础设施的配置。我们将带您逐步了解以下步骤：

1. 创建起始堆栈 (p. 36) – 使用基础 Amazon Linux AMI 创建堆栈，并使用 AWS CloudFormation 帮助程序脚本安装 Apache Web Server 和一个简单的 PHP 应用程序。
2. 更新应用程序 (p. 36) – 更新应用程序中的某一文件并使用 AWS CloudFormation 部署软件。
3. 更新实例类型 (p. 39) – 更改基础 Amazon EC2 实例的实例类型。
4. 更新 Amazon EC2 实例上的 AMI (p. 40) – 更改堆栈中的 Amazon EC2 实例的亚马逊系统映像 (AMI)。
5. 将密钥对添加到实例 (p. 41) - 将 Amazon EC2 密钥对添加到实例中，然后更新安全组以允许对实例的 SSH 访问。
6. 更新 IAM 策略 (p. 42) – 更新模板中定义的 IAM 用户权限。
7. 更改堆栈的资源 (p. 43) – 从堆栈中添加和删除资源，并通过更新模板将资源转换成自动扩展、负载平衡的应用程序。

简单的应用程序

我们将从创建可在本部分剩下的所有内容中使用的堆栈开始。我们已提供了一个简单的模板来启动在 Apache Web Server 中托管并在 Amazon Linux AMI 上运行的简单实例 PHP Web 应用程序。

Apache Web Server、PHP 和简单的 PHP 应用程序全部都由默认安装在 Amazon Linux AMI 上的 AWS CloudFormation 帮助程序脚本进行安装。以下模板代码段所示为说明待安装软件包和文件的元数据，此情况下为 Amazon Linux AMI 的 Yum 存储库中的 Apache Web Server 和 PHP 基础设施。代码段还显示 Services 部分，以确保 Apache Web Server 处于运行状态。Amazon EC2 实例定义的 Properties 部分中的 UserData 属性包含调用 cfn-init 来安装软件包和文件的 CloudInit 脚本。

```
"WebServerHost": {
  "Type" : "AWS::EC2::Instance",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"             : [],
            "php"               : []
          }
        },

        "files" : {

          "/var/www/html/index.php" : {
            "content" : { "Fn::Join" : ["", [
              "<?php\n",
             "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",

              "echo '<p>", { "Ref" : "WelcomeMessage" }, "</p>';\n",
              "?>\n"
            ]]},
            "mode"    : "000644",
            "owner"   : "apache",
            "group"   : "apache"
```

```
          },
        },

        :

        "services" : {
          "sysvinit" : {
            "httpd"    : { "enabled" : "true", "ensureRunning" : "true" },
            "sendmail" : { "enabled" : "false", "ensureRunning" : "false" }
          }
        }
      }
    },

    "Properties": {
      :
      "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
        "#!/bin/bash\n",
        "yum update -y aws-cfn-bootstrap\n",

        :

        "# Install the simple web page\n",
        "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" },
        "          -r WebServerHost ",
        "          --access-key ", { "Ref" : "WebServerKeys" },
        "          --secret-key ", {"Fn::GetAtt": ["WebServerKeys", "SecretAc
cessKey"]},
        "          --region ", { "Ref" : "AWS::Region" },
        " || error_exit 'Failed to run cfn-init'\n",

        :
      ]]}}
    }
  },
```

应用程序本身是非常简单的两行字 "Hello, World" 示例，在模板中进行了完整定义。对于真实应用程序，文件可存储在 Amazon S3、GitHub 或其他存储库中，并可通过模板进行引用。AWS CloudFormation 可下载软件包（如 RPM 或 RubyGem），并可引用单个文件以及展开 `.zip` 和 `.tar` 文件以在 Amazon EC2 实例上创建应用程序项目。

模板用于启用和配置 cfn-hup 后台程序以侦听 Amazon EC2 实例元数据中所定义配置的更改。可以通过使用 cfn-hup 后台程序来更新应用程序软件（如 Apache 或 PHP 的版本），也可以通过 AWS CloudFormation 更新 PHP 应用程序文件本身。来自于模板中同一个 EC2 资源的以下代码段显示的是，在侦测到元数据的任何更改时配置 cfn-hup 以调用 cfn-init 来更新软件所需的部件：

```
  "WebServerHost": {
    "Type" : "AWS::EC2::Instance",
    "Metadata" : {
      "AWS::CloudFormation::Init" : {
        "config" : {

          :

        "files" : {
```

```
            "/etc/cfn/cfn-credentials" : {
              "content" : { "Fn::Join" : ["", [
                "AWSAccessKeyId=", { "Ref" : "WebServerKeys" }, "\n",
               "AWSSecretKey=", {"Fn::GetAtt": ["WebServerKeys", "SecretAccess
Key"]}, "\n"
              ]]},
              "mode"    : "000400",
              "owner"   : "root",
              "group"   : "root"
            },

            "/etc/cfn/cfn-hup.conf" : {
              "content" : { "Fn::Join" : ["", [
                "[main]\n",
                "stack=", { "Ref" : "AWS::StackName" }, "\n",
                "credential-file=/etc/cfn/cfn-credentials\n",
                "region=", { "Ref" : "AWS::Region" }, "\n"
              ]]},
              "mode"    : "000400",
              "owner"   : "root",
              "group"   : "root"
            },

            "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
              "content": { "Fn::Join" : ["", [
                "[cfn-auto-reloader-hook]\n",
                "triggers=post.update\n",
                "path=Resources.WebServerHost.Metadata.AWS::CloudForma
tion::Init\n",
                "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName"
 },
                "              -r WebServerHost ",
                "              --credential-file /etc/cfn/cfn-credentials ",
               "             --region     ", { "Ref" : "AWS::Region" }, "\n",

                "runas=root\n"
              ]]}
            }
          },
          :
    },

    "Properties": {

         :

      "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [

        :

        "# Start up the cfn-hup daemon to listen for changes\n",
        "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

        :
      ]]}}
    }
  },
```

要完成堆栈，模板需要创建 Amazon EC2 安全组和使我们可通过一致性 IP 地址来参考应用程序的弹性 IP，以及会在实例 CPU 达到阈值时触发的 CloudWatch 警报。 以下是完整模板，您还可以在以下网址中下载或参考该模板：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorial+Part1.template .

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template UpdateEC2 Part 1: Sample
 template that can be used to test EC2 updates. **WARNING** This template creates
 an Amazon EC2 Instance. You will be billed for the AWS resources used if you
create a stack from this template.",

  "Parameters" : {

    "WebServerInstanceType" : {
      "Description" : "Amazon EC2 instance type for Web Server",
      "Type" : "String",
      "Default" : "t1.micro",
      "AllowedValues" : [ "t1.micro", "m1.small", "m1.large", "m1.xlarge",
"m2.xlarge", "m2.2xlarge", "m2.4xlarge", "c1.medium", "c1.xlarge" ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "32" },
      "m1.small"    : { "Arch" : "32" },
      "m1.large"    : { "Arch" : "64" },
      "m1.xlarge"   : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m2.2xlarge"  : { "Arch" : "64" },
      "m2.4xlarge"  : { "Arch" : "64" },
      "c1.medium"   : { "Arch" : "32" },
      "c1.xlarge"   : { "Arch" : "64" }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "32" : "ami-7f418316", "64" : "ami-7341831a" },
      "us-west-1"      : { "32" : "ami-951945d0", "64" : "ami-971945d2" },
      "us-west-2"      : { "32" : "ami-16fd7026", "64" : "ami-10fd7020" },
      "eu-west-1"      : { "32" : "ami-24506250", "64" : "ami-20506254" },
      "ap-southeast-1" : { "32" : "ami-74dda626", "64" : "ami-7edda62c" },
      "ap-northeast-1" : { "32" : "ami-dcfa4edd", "64" : "ami-e8fa4ee9" }
    }
  },

  "Resources" : {

    "WebServerUser" : {
      "Type" : "AWS::IAM::User",
      "Properties" : {
        "Path": "/",
        "Policies": [{
          "PolicyName": "root",
          "PolicyDocument": {
            "Version" : "2012-10-17",
```

```
            "Statement":[{
              "Effect": "Allow",
              "Action": [
                "cloudformation:DescribeStackResource"
              ],
              "Resource": "*"
            }]
          }
        }]
      }
    },

    "WebServerKeys" : {
      "Type" : "AWS::IAM::AccessKey",
      "Properties" : {
        "UserName" : {"Ref": "WebServerUser"}
      }
    },

    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp"
: "0.0.0.0/0"}
        ]
      }
    },

    "Endpoint" : {
      "Type" : "AWS::EC2::EIP",
      "Properties" : {
        "InstanceId" : { "Ref" : "WebServerHost" }
      }
    },

    "WebServerHost": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "Comment" : "Install a simple PHP application",
        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "httpd"             : [],
                "php"               : []
              }
            },

            "files" : {

              "/var/www/html/index.php" : {
                "content" : { "Fn::Join" : ["", [
                  "<?php\n",
                 "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",

                  "?>\n"
```

```
                ]]},
                "mode"    : "000644",
                "owner"   : "apache",
                "group"   : "apache"
              },

              "/etc/cfn/cfn-credentials" : {
                "content" : { "Fn::Join" : ["", [
                  "AWSAccessKeyId=", { "Ref" : "WebServerKeys" }, "\n",
                  "AWSSecretKey=", {"Fn::GetAtt": ["WebServerKeys", "SecretAc
cessKey"]},
                  "\n"
                ]]},
                "mode"    : "000400",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/cfn-hup.conf" : {
                "content" : { "Fn::Join" : ["", [
                  "[main]\n",
                  "stack=", { "Ref" : "AWS::StackName" }, "\n",
                  "credential-file=/etc/cfn/cfn-credentials\n",
                  "region=", { "Ref" : "AWS::Region" }, "\n"
                ]]},
                "mode"    : "000400",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
                "content": { "Fn::Join" : ["", [
                  "[cfn-auto-reloader-hook]\n",
                  "triggers=post.update\n",
                  "path=Resources.WebServerHost.Metadata.AWS::CloudForma
tion::Init\n",
                 "action=/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName"
 },
                  "         -r WebServerHost ",
                  "         --credential-file /etc/cfn/cfn-credentials ",
                  "         --region     ", { "Ref" : "AWS::Region" }, "\n",
                  "runas=root\n"
                ]]}
              }
            },

            "services" : {
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true" },

                "sendmail" : { "enabled" : "false", "ensureRunning" : "false"
}
              }
            }
          }
        }
      },
```

```
      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                      { "Fn::FindInMap" : [ "AWSInstanceType2Arch",
                      { "Ref" : "WebServerInstanceType" }, "Arch" ] } ] },
        "InstanceType" : { "Ref" : "WebServerInstanceType" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
          "#!/bin/bash\n",
          "yum update -y aws-cfn-bootstrap\n",

          "# Helper function\n",
          "function error_exit\n",
          "{\n",
          "  /opt/aws/bin/cfn-signal -e 1 -r \"$1\" '",
                { "Ref" : "WebServerWaitHandle" }, "'\n",
          "  exit 1\n",
          "}\n",

          "# Install the simple web page\n",
          "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" },
          "         -r WebServerHost ",
          "         --access-key ", { "Ref" : "WebServerKeys" },
          "         --secret-key ", {"Fn::GetAtt": ["WebServerKeys",
"SecretAccessKey"]},
          "         --region ", { "Ref" : "AWS::Region" },
          " || error_exit 'Failed to run cfn-init'\n",

          "# Start up the cfn-hup daemon to listen for changes\n",
          "/opt/aws/bin/cfn-hup || error_exit 'Failed to start cfn-hup'\n",

          "# All done so signal success\n",
          "/opt/aws/bin/cfn-signal -e 0 -r \"WebServer setup complete\" '",
                { "Ref" : "WebServerWaitHandle" }, "'\n"
        ]]}}
      }
    },

    "WebServerWaitHandle" : {
      "Type" : "AWS::CloudFormation::WaitConditionHandle"
    },

    "WebServerWaitCondition" : {
      "Type" : "AWS::CloudFormation::WaitCondition",
      "DependsOn" : "WebServerHost",
      "Properties" : {
        "Handle" : {"Ref" : "WebServerWaitHandle"},
        "Timeout" : "300"
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Value" : { "Fn::Join" : ["", ["http://", { "Ref" : "Endpoint" } ]] },
      "Description" : "Application URL"
    }
```

```
  }
}
```

此示例中采用的是单个 EC2 实例和弹性 IP 地址，但您可以在使用弹性负载均衡器和 Auto Scaling 组管理应用程序服务器集合的更复杂解决方案上使用相同的机制。但是，Auto Scaling 组有很多特殊注意事项。有关更多信息，请参阅 更新 Auto Scaling 组 (p. 39)。

创建起始堆栈

我们将使用 AWS 管理控制台从示例模板中创建起始堆栈以实现此示例的目的。

Caution

此程序完成后将会部署实时的 AWS 服务。只要这些服务在运行，您就要按照标准使用费率付费。

通过 AWS Management Console 创建堆栈

1. 从 https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorial+Part1.template 下载模板并将其保存在系统上的安全位置。注意保存位置，因为您需要在后续步骤中使用此文件。
2. 从 https://console.amazonaws.cn/cloudformation 登录 AWS CloudFormation 控制台。
3. 单击 Create New Stack (创建新堆栈)。
4. 在 Create New Stack (创建新堆栈) 向导的 Create A New Stack (创建新堆栈) 屏幕上的 Name (名称) 框中，键入 **`UpdateTutorial`**。在同一页上，选择 Upload template file (上传模板文件) 并浏览到在第一步中下载的文件，然后单击 Next Step (下一步)。
5. 在 Specify Parameters (指定参数) 屏幕上的 Web Server Instance Type (Web 服务器实例类型) 框中，键入 **`t1.micro`**。然后单击 Next Step (下一步)。
6. 在 Options (选项) 屏幕上，选中 I acknowledge that this template may create IAM resources (我确认，此模板可创建 IAM 资源) 复选框，然后单击 Next Step (下一步)。由于模板会创建被锁定为只允许对 `cfn-init` 和 `cfn-hup` 所需 API 操作进行访问的 IAM 用户，因此此步骤为必需步骤。IAM 用户的证书存储在新创建的 EC2 实例上。
7. 在 Review (审核) 屏幕上，确认所有设置都符合您的要求，然后单击 Create (创建)。

当您的堆栈状态变成 CREATE_COMPLETE 后，输出选项卡会显示网站的 URL。如果您单击 WebsiteURL 的输出值，您将看到新 PHP 应用程序在工作。

更新应用程序

现在您已部署好堆栈，接下来更新应用程序吧。我们将对应用程序所打印出的文本进行简单更改。要执行此操作，我们将添加回显命令至 index.php 文件，如此模板代码段所示：

```
"WebServerHost": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "AWS::CloudFormation::Init" : {
          "config" : {
              :

            "files" : {

              "/var/www/html/index.php" : {
                "content" : { "Fn::Join" : ["", [
                  "<?php\n",
```

```
              "echo '<h1>AWS CloudFormation sample PHP application</h1>';\n",

                "echo 'Updated version via UpdateStack';\n ",
                "?>\n"
              ]]},
              "mode"    : "000644",
              "owner"   : "apache",
              "group"   : "apache"
            },

            :

    }
  },
```

您可以手动编辑您先前下载的模板，或从以下网址下载更新后的模板：https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorial+Part2.template。

现在，我们将更新堆栈。

通过 AWS Management Console 更新堆栈

1. 从以下网址登录 AWS CloudFormation 控制台：https://console.amazonaws.cn/cloudformation。
2. 在 AWS CloudFormation 控制面板中，单击您之前创建的堆栈，然后单击 Update Stack (更新堆栈)。
3. 在 Update Stack (更新堆栈) 向导的 Update Stack (更新堆栈) 屏幕上，单击 Upload template file (上传模板文件)，选择修改后的模板，然后单击 Next Step (下一步)。
4. 在 Options (选项) 屏幕上，选中 I acknowledge that this template may create IAM resources (我确认，此模板可创建 IAM 资源) 复选框，然后单击 Next Step (下一步)。由于模板会创建被锁定为只允许对 `cfn-init` 和 `cfn-hup` 所需 API 操作进行访问的 IAM 用户，因此此步骤为必需步骤。IAM 用户的证书存储在新创建的 EC2 实例上。
5. 因为堆栈没有堆栈策略，所以单击 Next Step (下一步)。在没有覆盖策略的情况下，所有资源均可更新。
6. 在 Review (审核) 屏幕上，确认所有设置都符合您的要求，然后单击 Update (更新)。

如果您通过 AWS Management Console 更新堆栈，您将会注意到创建初始堆栈所用参数已预填充到 Update Stack (更新堆栈) 向导的 Parameters (参数) 页上。如果您使用 `aws cloudformation update-stack` 命令，请务必键入与您原先用于创建堆栈的参数相同的值。

当您的堆栈处于 UPDATE_COMPLETE 状态时，您可以再次单击 WebsiteURL 输出值以验证应用程序的更改已生效。默认状态下，cfn-hup 后台程序每 15 分钟运行一次，因此最多能花 15 分钟在堆栈更新后更改应用程序。

要查看已更新的资源集，请转至 AWS CloudFormation 控制台。在 Events (事件) 选项卡上，查看堆栈事件。在这种特定情况下，Amazon EC2 实例 WebServerHost 的元数据已更新，这将导致 AWS CloudFormation 同时重新计算弹性 IP 地址和 WaitCondition 资源，以确保更改没有影响更新。其他堆栈资源均未修改。AWS CloudFormation 将只更新堆栈中受堆栈的任何更改影响的资源。此类更改可直接进行，如属性或元数据更改，也可由依赖性或 Ref 和 GetAtt 中的数据流或其它内部模板函数导致。

这一简单更新对过程进行了阐述；但是，您可以对您的 Amazon EC2 实例中所部署文件和软件包进行更复杂的更改。例如，您可能会决定将 MySQL 与 MySQL 的 PHP 支持一起添加到实例中。要执行此操作，只需要将附加软件包和文件与任何附加服务一起添加到配置中，然后更新堆栈以部署更改。在以下模板代码段中，更改被红色高亮显示：

```
    "WebServerHost": {
      "Type" : "AWS::EC2::Instance",
      "Metadata" : {
        "Comment" : "Install a simple PHP application",
        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "httpd"             : [],
                "php"               : [],
                "php-mysql"         : [],
                "mysql-server"      : [],
                "mysql-devel"       : [],
                "mysql-libs"        : [],
                "mysql"             : []
              }
            },

            :

            "services" : {
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true" },

                "mysqld"   : { "enabled" : "true", "ensureRunning" : "true" },

                "sendmail" : { "enabled" : "false", "ensureRunning" : "false"
}
              }
            }
          }
        }
      },

      "Properties": {
           :
      }
    }
```

您还可以使用 UpdateStack 与 CloudFormation 元数据将应用程序所使用的软件包更新到新版本。前述示例中，每个软件包的版本属性都为空，这表示 cfn-init 应安装最新版的软件包。

```
     "packages" : {
       "yum" : {
         "httpd"             : [],
         "php"               : []
       }
```

您可以视需要指定软件包的版本字符串。如果您在后续更新堆栈调用中更改版本字符串，则会部署新版软件包。此处所示为 RubyGems 软件包版本号的使用示例。支持版本化的任何软件包都可以有特定版本。

```
  "packages" : {
    "rubygems" : {
      "mysql"           : [],
      "rubygems-update" : ["1.6.2"],
```

```
    "rake"           : ["0.8.7"],
    "rails"          : ["2.3.11"]
  }
  }
```

更新 Auto Scaling 组

如果您在模板中使用与 Amazon EC2 实例资源截然相反的 Auto Scaling 组，应用程序将会以完全相同的方式更新；但是，AWS CloudFormation 在 Auto Scaling 组中的所有 EC2 实例上都不会提供任何同步或序列化。每个主机上的 cfn-hup 后台程序都将独立运行且会按其自己的计划更新应用程序。当您使用 cfn-hup 更新实例上的配置时，每个实例都将按其自己的计划运行 cfn-hup 挂接；堆栈中的实例之间不协调。您应该考虑以下各项：

- 如果 Auto Scaling 组中所有 EC2 实例的 cfn-hup 更改都同时运行，更新期间您的服务可能不可用。
- 如果 cfn-hup 更改在不同时间运行，则新旧版软件可能会同时运行。

更改资源属性

借助 AWS CloudFormation，您可以更改堆栈中现有资源的属性。以下部分说明了解决特定问题的各种更新；但是，堆栈中支持更新的任何资源的任何属性都可视需要进行修改。

更新实例类型

我们到目前为止所建立的堆栈使用 t1.micro Amazon EC2 实例。假设您新建的网站获取的流量比 t1.micro 实例能处理的流量多，且您现在想移动到 m1.small EC2 实例类型中。如果实例类型结构从 32 位变成 64 位，则会使用不同的 AMI 创建实例。在检验模板中的映射时，您将会看到 t1.micro 和 m1.small 实例的结构都是 32 位且它们使用同一个基础 Amazon Linux AMI。

```
"Mappings" : {
  "AWSInstanceType2Arch" : {
    "t1.micro"    : { "Arch" : "32" },
    "m1.small"    : { "Arch" : "32" },
    "m1.large"    : { "Arch" : "64" },
    "m1.xlarge"   : { "Arch" : "64" },
    "m2.xlarge"   : { "Arch" : "64" },
    "m2.2xlarge"  : { "Arch" : "64" },
    "m2.4xlarge"  : { "Arch" : "64" },
    "c1.medium"   : { "Arch" : "32" },
    "c1.xlarge"   : { "Arch" : "64" }
  },
  "AWSRegionArch2AMI" : {
    "us-east-1"      : { "32" : "ami-7f418316", "64" : "ami-7341831a" },
    "us-west-1"      : { "32" : "ami-951945d0", "64" : "ami-971945d2" },
    "us-west-2"      : { "32" : "ami-16fd7026", "64" : "ami-10fd7020" },
    "eu-west-1"      : { "32" : "ami-24506250", "64" : "ami-20506254" },
    "ap-southeast-1" : { "32" : "ami-74dda626", "64" : "ami-7edda62c" },
    "ap-northeast-1" : { "32" : "ami-dcfa4edd", "64" : "ami-e8fa4ee9" }
  }
},
```

让我们使用在前述部分中进行修改的模板更改实例类型。由于 InstanceType 是模板的输入参数，我们不需要修改模板；我们只能在“堆栈更新”向导的“指定参数”页面上更改参数值。

通过 AWS Management Console 更新堆栈

1. 从以下网址登录 AWS CloudFormation 控制台：https://console.amazonaws.cn/cloudformation。
2. 在“AWS CloudFormation”控制面板中，单击您先前创建的堆栈，然后单击 Update Stack (更新堆栈)。
3. 在 Update Stack (更新堆栈) 向导的 Select Template (选择模板) 屏幕上，选择 Upload template file (上传模板文件)，选择修改后的模板，然后单击 Next Step (下一步)。

   此时将显示“指定参数”页，其中创建初始堆栈所用参数已预填充到 Specify Parameters (指定参数) 部分中。
4. 将 WebServerInstanceType 文本框的值从 `t1.micro` 更改为 `m1.small`。然后单击 Next Step (下一步)。
5. 在 Options (选项) 屏幕上，选中 I acknowledge that this template may create IAM resources (我确认，此模板可创建 IAM 资源) 复选框，然后单击 Next Step (下一步)。由于模板会创建被锁定为只允许对 `cfn-init` 和 `cfn-hup` 所需 API 操作进行访问的 IAM 用户，因此此步骤为必需步骤。IAM 用户的证书存储在新创建的 EC2 实例上。
6. 因为堆栈没有堆栈策略，所以单击 Next Step (下一步)。在没有覆盖策略的情况下，所有资源均可更新。
7. 在 Review (审核) 屏幕上，确认所有设置都符合您的要求，然后单击 Update (更新)。

可以通过启动和停止实例来动态更改 EBS 支持的 Amazon EC2 实例的实例类型。AWS CloudFormation 将尝试通过更新实例类型和重启实例来优化更改，因此实例 ID 不会更改。但是，实例重启时，实例的公用 IP 地址会更改。为了确保弹性 IP 地址在更改后正确进行绑定，AWS CloudFormation 还会更新弹性 IP 地址。您可以在 AWS CloudFormation 控制台中的“事件”选项卡上查看更改。

要通过 AWS 管理控制台检查实例类型，请打开 Amazon EC2 控制台并在其中查找您的实例。

更新 Amazon EC2 实例上的 AMI

现在让我们来看看如何更改实例上所运行的亚马逊机器映像 (AMI)。我们将通过更新堆栈来触发 AMI 更改从而使用新的 EC2 实例类型，如类型为 64 位的 m1.large。

如之前部分所述，我们将使用现有模板更改示例堆栈所使用的实例类型。在“Stack Update (堆栈更新)”向导中的“Specify Parameters (指定参数)”页面上，更改 Web 服务器实例类型的值。

在这种情况下，不能只通过启动和停止实例来修改 AMI；AWS CloudFormation 将其视为对资源不可变属性的更改。为了对不可变属性进行更改，AWS CloudFormation 必须启动替换资源，在此情况下为运行新 AMI 的新 Amazon EC2 实例。

在新实例运行之后，AWS CloudFormation 会更新堆栈中的弹性 IP 地址等其他资源，以指向新资源。所有新资源被创建且旧资源被删除的过程被称为 UPDATE_CLEANUP。此时，您将注意到堆栈中的实例的 ID 已随着更新更改。“Event”表中的事件包含说明“所请求的更新包含对不可变属性的更改，因此创建新的物理资源”，以指示资源已被替代。

如果您已将应用程序代码写入您想更新的 AMI 中，您可以使用同一堆栈更新机制更新 AMI 以加载您的新应用程序。

要更新您的堆栈中的实例之 AMI

1. 创建含有应用程序或操作系统更改的新 AMI。有关更多信息，请转至 *Amazon Elastic Compute Cloud 用户指南* 中的创建您自己的 AMI。
2. 更新您的模板以合并新 AMI ID。
3. 通过 AWS 管理控制台（如更新应用程序 (p. 36)中所述）或使用 AWS 命令 `aws cloudformation update-stack` 更新堆栈。

在您更新堆栈时，AWS CloudFormation 侦测到 AMI ID 已更改，然后用我们触发前一个更新所使用的方法触发堆栈更新。

更新 Auto Scaling 组的 Amazon EC2 启动配置

如果您使用的是 Auto Scaling 组而不是 EC2 实例，更新正在运行实例的过程会有点不同。借助 Auto Scaling 资源，可将实例类型或 AMI Id 等 EC2 实例配置封装到 Auto Scaling 启动配置中。您可以用我们在前述部分中对 EC2 实例资源进行更改时所用的方法对启动配置进行更改。但是，启动配置的更改不会影响 Auto Scaling 组中任何正在运行的 EC2 实例。更新后的启动配置只适用于更新之后创建的新实例。

如果您想将启动配置的更改传至您的 Auto Scaling 组中的所有实例上，您可以使用 Auto Scaling as-terminate-instance-in-auto-scaling-group 命令行工具如下所述替代每个实例：

```
as-terminate-instance-in-auto-scaling-group <instance_id> --no-decrement-desired-
capacity
```

有关 Auto Scaling 命令行工具的更多信息，请转至 *Auto Scaling 开发人员指南* 中的使用命令行工具。实例终止后，Auto Scaling 将会用使用新 AMI 的实例替代该实例。实例替代不是即时发生的，可能需要花点时间用 Elastic Load Balancing 注册新实例和任何其它受影响的服务。注意不要让您的组容量在更新期间产生不足。

添加资源属性

到目前为止，我们已在模板中查看了资源现有属性的更改。您还可以添加原先未在模板中指定的属性。为了阐明上述操作，我们将会添加 Amazon EC2 密钥对到现有 EC2 实例中然后在 Amazon EC2 安全组中打开端口 22，从而使您可以使用安全外壳 (SSH) 访问实例。

将密钥对添加到实例

要将 SSH 访问添加到现有 Amazon EC2 实例中

1. 将其它参数添加到模板中，从而以现有 EC2 密钥对的名称进行传递。

```
  "Parameters" : {

    "WebServerKeyName" : {
      "Description" : "Name of an existing Amazon EC2 key pair for SSH ac
cess",
      "Type" : "String"
    },
    :
  },
```

2. 将 KeyName 属性添加至 Amazon EC2 实例。

```
          "WebServerHost": {
          "Type" : "AWS::EC2::Instance",
          :
          "Properties": {
          :
          "KeyName" : { "Ref" : "WebServerKeyName" },
          :
          }
          },
```

3. 将端口 22 添加至 Amazon EC2 安全组的入口规则。

```
   "WebServerSecurityGroup" : {
     "Type" : "AWS::EC2::SecurityGroup",
     "Properties" : {
       "GroupDescription" : "Enable HTTP and SSH",
       "SecurityGroupIngress" : [
        {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp"
: …
        {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp"
: …
       ]
     }
   },
```

4. 通过 AWS Management Console（如更新应用程序 (p. 36)中所述）或使用 AWS 命令 `aws cloudformation update-stack` 更新堆栈。

您可以在以下网址下载或查看更新后的模板：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorial+Part3.template。

更新 IAM 策略

接下来，我们将更新与 IAM 用户相关的 IAM 策略，该策略会传递至 Amazon EC2 实例上运行的代码中并为其使用。假设新版应用程序需要通过实例访问 Amazon EC2 API。要启用访问，请按照以下所述在模板中更新 IAM 策略：

```
  "WebServerUser" : {
    "Type" : "AWS::IAM::User",
    "Properties" : {
      "Path": "/",
      "Policies": [{
        "PolicyName": "root",
        "PolicyDocument": {
          "Version" : "2012-10-17",
          "Statement":[{
            "Effect": "Allow",
            "Action": [
              "cloudformation:DescribeStackResource",
              "ec2:*"
            ],
            "Resource": "*"
          }]
        }
      }]
    }
    },
```

可从 https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorial+Part4.template 中下载或参考更新后的模板。此更改将为用户修改策略；它不需要堆栈中进行的其它更改。堆栈更新后，Amazon EC2 实例上的用户证书将有对 Amazon EC2 API 的访问权。

更改堆栈的资源

由于应用程序需求可能会随着时间的推移发生变化，因此 AWS CloudFormation 允许您更改构成堆栈的资源集。为了进行演示，我们将使用 添加资源属性 (p. 41) 中的单实例应用程序，并通过更新堆栈将其转换成自动扩展、负载平衡的应用程序。

要开始更改，您可以手动编辑先前下载的模板，或从以下网址下载更新后的模板：https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorial+Part3.template。

这一操作会使用弹性 IP 地址创建简单的单实例 PHP 应用程序。现在，我们将在更新时更改应用程序资源以将应用程序转变成可用性高、可自动扩展的负载均衡型应用程序。

1. 从模板中删除弹性 IP 地址资源。

```
"Endpoint" : {
  "Type" : "AWS::EC2::EIP",
  "Properties" : {
    "InstanceId" : { "Ref" : "WebServerHost" }
  }
},
```

2. 添加弹性负载均衡器资源。

```
"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "Listeners" : [ {
      "LoadBalancerPort" : "80",
      "InstancePort" : "80",
      "Protocol" : "HTTP"
    } ],
    "HealthCheck" : {
      "Target" : "HTTP:80/",
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    }
  }
},
```

3. 将模板中的 EC2 实例转换成 Auto Scaling 启动配置。由于属性都相同，所以我们只需要更改类型名称，从：

```
"WebServerHost": {
  "Type" : "AWS::EC2::Instance",
```

到：

```
"WebServerConfig": {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
```

为了使模板更清楚，我还将资源名称从 *WebServerHost* 更改为 *WebServerConfig*，因此您将需要更新由 cfn-init 和 cfn-hup 引用的资源名称（只需要搜索 WebServerHost 并将其替换为 WebServerConfig）。

4. 添加 Auto Scaling 组资源。

```
"WebServerGroup" : {
  "Type" : "AWS::AutoScaling::AutoScalingGroup",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : ""},
    "LaunchConfigurationName" : { "Ref" : "WebServerConfig" },
    "MinSize" : "1",
    "MaxSize" : "3",
    "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
  }
},
```

5. 更新安全组定义以锁定通过负载均衡器分配至实例的流量。

```
"WebServerSecurityGroup" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "Properties" : {
    "GroupDescription" : "Enable SSH access and HTTP from the load balancer
 only",
    "SecurityGroupIngress" : [{
      "IpProtocol" : "tcp",
      "FromPort" : "22",
      "ToPort" : "22",
      "CidrIp" : "0.0.0.0/0"
    }, {
      "IpProtocol" : "tcp",
      "FromPort" : "80",
      "ToPort" : "80",
      "SourceSecurityGroupOwnerId" : {"Fn::GetAtt" :
         ["ElasticLoadBalancer", "SourceSecurityGroup.OwnerAlias"]},
      "SourceSecurityGroupName" : {"Fn::GetAtt" :
         ["ElasticLoadBalancer", "SourceSecurityGroup.GroupName"]}
    }]
  }
},
```

6. 更新 Outputs 以返回弹性负载均衡器的 DNS 名称，以作为应用程序的位置，从：

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : ["", ["http://", {"Ref":"Endpoint" }]]},
  "Description" : "Application URL"
}
```

到：

```
"WebsiteURL" : {
  "Value" : { "Fn::Join" : ["", ["http://",
      { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]}]]},
  "Description" : "Application URL"
}
```

您可以从以下网址下载或参考完整模板：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/UpdateTutorialPart5.template。

如果您使用此模板更新堆栈，您会将您的单实例简单应用程序转换成可用性高、可自动扩展的、多可用区负载均衡型应用程序。只有需要更新的资源会被修改，所以，如果此应用程序有任何数据存储，该数据都会保持原样。现在，您便可以使用 AWS CloudFormation 在您的需求发生变化时增大或增强您的堆栈了。

可用性和影响注意事项

不同的属性会对堆栈中的资源造成不同的影响。您可以使用 AWS CloudFormation 更新任何属性；但是您应该在进行任何更改之前考虑以下问题：

1. 更新会如何影响资源本身？例如，更新警报阈值会使警报在更新期间处于非活动状态。正如我们所见，更改实例类型时需要停止和重启实例。AWS CloudFormation 使用底层资源的 Update 或 Modify 操作来对资源进行更改。要了解更改的影响，您应该查看特定资源的文档。
2. 更改可变还是不可变？对资源属性的某些更改，如更改 Amazon EC2 实例上的 AMI，不受基础服务的支持。如果更改可变，AWS CloudFormation 将使用适用于基础资源的“更新”或“修改”类型 API。对于不可变的属性更改，AWS CloudFormation 将用更新后的属性创建新资源，然后再删除旧资源之前将此资源链接至堆栈。虽然 AWS CloudFormation 尝试减少堆栈资源的停机时间，但替代资源是一个多步骤过程，需要时间。重新配置堆栈期间，您的应用程序不能全面运行。例如，它可能不能为请求提供服务或访问数据库。

相关资源

有关使用 AWS CloudFormation 启动应用程序的更多信息以及集成其它配置与 Puppet 和 Opscode Chef 等部署服务的更多信息，请参阅以下白皮书：

- 通过 AWS CloudFormation 启动应用程序
- AWS CloudFormation 和 Opscode Chef 集成
- AWS CloudFormation 和 Puppet 集成

此部分使用的模板为 "Hello, World" PHP 应用程序。模板库中还有一个 Amazon ElastiCache 示例模板，该模板显示如何使用 cfn-hup 和 cfn-init 集成 PHP 应用程序和 ElasticCache，以响应 Amazon ElastiCache 缓存集群配置中的更改，这所有的操作都可通过更新堆栈执行。

AWS CloudFormation 自定义资源演练

什么是自定义资源？

自定义资源是特殊的 AWS CloudFormation 资源，提供了一种使 template developer 可以在 AWS CloudFormation 堆栈中包含非 AWS 资源的方式。custom resource provider 既可以是 template developer，也可以是独立的第三方资源提供者。

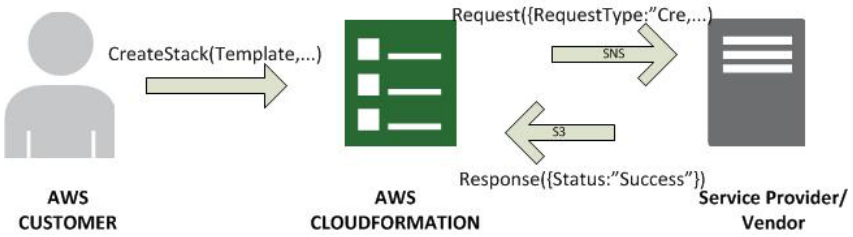
在 AWS CloudFormation 模板中，自定义资源是由 `AWS::CloudFormation::CustomResource` (p. 239) 类型或 `Custom::String` (p. 239) 指定的。

自定义资源的工作原理

对自定义资源执行的任何操作均涉及三方：template developer、AWS CloudFormation 和custom resource provider。template developer和custom resource provider可以是同一人员或实体，但过程将相同。以下步骤介绍了一般过程：

1. template developer创建、更新或删除包含自定义资源的堆栈。该模板包含自定义资源的服务令牌和所有输入/输出数据参数。
2. AWS CloudFormation 使用 SNS 主题与custom resource provider通信，向其发送请求的类型（创建、更新或删除）以及堆栈模板中存储的任何输入数据。AWS CloudFormation 向custom resource provider提供用于响应的 S3 URL。
3. custom resource provider处理消息并返回 `SUCCESS` 或 `FAILED` 响应。如果请求成功（输出数据），custom resource provider还可以发送可由template developer访问的资源属性的名称和值；如果请求失败，则发送提供有关本次失败的详细信息的字符串。
4. AWS CloudFormation 根据收到的响应设置堆栈状态，并使用 `Fn::GetAtt` (p. 502) 向 template developer提供任何自定义资源输出数据的值。

下图说明了template developer、AWS CloudFormation 和custom resource provider之间的关系：



此演练包括哪些内容？

此演练将逐步执行自定义资源过程，并对由于自定义资源堆栈创建、更新和删除而发送和接收的事件和消息序列进行说明。

此演练分为三部分：

- 第 1 部分：堆栈创建 (p. 47)
- 第 2 部分：堆栈更新 (p. 49)
- 第 3 部分：堆栈删除 (p. 50)

第 1 部分：堆栈创建

1. template developer 创建包含自定义资源的 AWS CloudFormation 堆栈；在下面的模板示例中，我们对自定义资源 `MySeleniumTest` 使用自定义资源类型名称 `Custom::SeleniumTester`。

   此自定义资源类型名称使用*服务令牌*、可选的*特定于提供商的属性*以及custom resource provider定义的可选 `Fn::GetAtt` (p. 502) 属性声明。使用这些属性和特性可以将信息从template developer传递给custom resource provider，反之亦然。自定义资源类型名称必须是字母数字字符，最大长度为 60 个字符。

   下面的示例显示了一个既有自定义属性又有返回属性的模板：

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MySeleniumTest" : {
         "Type": "Custom::SeleniumTester",
         "Version" : "1.0",
         "Properties" : {
            "ServiceToken": "arn:aws:sns:us-east-1:84969EXAMPLE:CRTest",
            "seleniumTester" : "SeleniumTest()",
             "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
 "http://search.mysite.com" ],
            "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
         }
      }
   },
   "Outputs" : {
      "topItem" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
      },
      "numRespondents" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
      }
   }
}
```

   Note

   在提供商响应 AWS CloudFormation 期间，custom resource provider会返回使用 `Fn::GetAtt` 访问的数据的名称和值。如果custom resource provider是第三方，则template developer必须从custom resource provider获取这些返回值的名称。

2. AWS CloudFormation 使用 `"RequestType" : "Create"` 向资源提供者发送一条 Amazon SNS 通知，其中包含有关堆栈的信息、堆栈模板中的自定义资源属性和用于响应的 S3 URL。

   用于发送通知的 SNS 主题嵌入在模板的 ServiceToken 属性中。要避免使用硬编码值，template developer可以使用模板参数，以便在启动堆栈时输入值。

   下面的示例所示为一个自定义资源 `Create` 请求，其中包含一个用 `MySeleniumTester` 的 `LogicalResourceId` 创建的自定义资源类型名称 `Custom::SeleniumTester`：

```
{
   "RequestType" : "Create",
```

```
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-east-1:EXAMPLE/stack-name/guid",
   "RequestId" : "unique id for this create request",
   "ResourceType" : "Custom::SeleniumTester",
   "LogicalResourceId" : "MySeleniumTester",
   "ResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
"http://search.mysite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
   }
}
```

3. custom resource provider处理template developer发送的数据，并确定 `Create` 请求是否已成功。然后，资源提供者使用 AWS CloudFormation 发送的 S3 URL 来发送 `SUCCESS` 或 `FAILED` 响应。

   根据响应类型，AWS CloudFormation 将需要不同响应字段。请参阅参考主题中的"响应"部分，了解要处理的 RequestType。

   在响应创建或更新请求时，custom resource provider可以在响应的 Data (p. 529) 字段中返回数据元素。这些是名称/值对，*名称*对应于用于堆栈模板中的自定义资源的 `Fn::GetAtt` 属性。*值*是template developer对具有该属性名称的资源调用`Fn::GetAtt` 时返回的数据。

   以下是自定义资源响应的示例：

```
{
   "Status" : "SUCCESS",
   "PhysicalResourceId" : "Tester1",
   "StackId" : "arn:aws:cloudformation:us-east-1:EXAMPLE:stack/stack-
name/guid",
   "RequestId" : "unique id for this create request",
   "LogicalResourceId" : "MySeleniumTester",
   "Data" : {
      "resultsPage" : "http://www.myexampledomain/test-results/guid",
      "lastUpdate" : "2012-11-14T03:30Z",
   }
}
```

   The *`StackId`*, *`RequestId`*, and *`LogicalResourceId`* fields must be copied verbatim from the request.

4. AWS CloudFormation 将堆栈状态声明为 `CREATE_COMPLETE` 或 `CREATE_FAILED`。如果堆栈已成功创建，template developer通过 `Fn::GetAtt` (p. 502) 访问已创建自定义资源的输出值，可以使用这些值。

   例如，用于举例说明的自定义资源模板使用 `Fn::GetAtt` 将资源输出复制到堆栈输出：

```
"Outputs" : {
   "topItem" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
   },
   "numRespondents" : {
      "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
   }
}
```

有关 Create 请求中涉及的请求和响应对象的详细信息，请参阅自定义资源参考 (p. 526)中的创建 (p. 529)。

第 2 部分：堆栈更新

要更新现有堆栈，您必须提交一个指定了堆栈资源属性更新的模板，如下面的示例所示。AWS CloudFormation 只更新模板中指定了更改的资源。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

您可以更新需要替换基础物理资源的自定义资源。在 AWS CloudFormation 模板中更新自定义资源时，AWS CloudFormation 会向相应的自定义资源发送更新请求。如果需要替换自定义资源，新的自定义资源必须使用新的物理 ID 发送响应。AWS CloudFormation 收到响应时，会比较新旧自定义资源的 PhysicalResourceId。如果不同，AWS CloudFormation 将更新识别为替换并向旧资源发送删除请求，如第 3 部分：堆栈删除 (p. 50)中所示。

1. template developer启动对包含自定义资源的堆栈的更新。在更新期间，template developer可以在堆栈模板中指定新属性。

   下面是一个使用自定义资源类型的堆栈模板 Update 示例：

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MySeleniumTest" : {
         "Type": "Custom::SeleniumTester",
         "Version" : "1.0",
         "Properties" : {
            "ServiceToken": "arn:aws:sns:us-east-1:84969EXAMPLE:CRTest",
            "seleniumTester" : "SeleniumTest()",
          "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
 "http://search.mysite.com",
               "http://mynewsite.com" ],
            "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
         }
      }
   },
   "Outputs" : {
      "topItem" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "resultsPage"] }
      },
      "numRespondents" : {
         "Value" : { "Fn::GetAtt" : ["MySeleniumTest", "lastUpdate"] }
      }
   }
}
```

2. AWS CloudFormation 会使用 "RequestType" : "Update" 向资源提供者发送一条 Amazon SNS 通知，其中包含与 Create 调用类似的信息，不同的是，*OldResourceProperties* 字段包含旧的资源属性，而 ResourceProperties 包含已更新的（如果有）资源属性。

   以下是一个 Update 请求的示例：

```
{
   "RequestType" : "Update",
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
```

```
   "StackId" : "arn:aws:cloudformation:us-east-1:EXAMPLE:stack/stack-
name/guid",
   "RequestId" : "uniqueid for this update request",
   "LogicalResourceId" : "MySeleniumTester",
   "ResourceType" : "Custom::SeleniumTester"
   "PhysicalResourceId" : "Tester1",
   "ResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
"http://search.mysite.com",
         "http://mynewsite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
   }
   "OldResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
"http://search.mysite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4" ]
   }
}
```

3. custom resource provider处理由 AWS CloudFormation 发送的数据。自定义资源执行更新并向 S3 URL 发送 SUCCESS 或 FAILED 响应。然后 AWS CloudFormation 比较新旧自定义资源的 PhysicalResourceIDs。如果不同，AWS CloudFormation 将更新识别为替换并向旧资源发送删除请求。下面的示例说明对 Update 请求的custom resource provider响应。

```
{
   "Status" : "SUCCESS",
   "StackId" : "arn:aws:cloudformation:us-east-1:EXAMPLE:stack/stack-
name/guid",
   "RequestId" : "uniqueid for this update request",
   "LogicalResourceId" : "MySeleniumTester",
   "PhysicalResourceId" : "Tester2"
}
```

   The *StackId*, *RequestId*, and *LogicalResourceId* fields must be copied verbatim from the request.

4. AWS CloudFormation 将堆栈状态声明为 UPDATE_COMPLETE 或 UPDATE_FAILED。如果更新失败，堆栈将回滚。如果堆栈更新成功，template developer可以使用 Fn::GetAtt 访问已创建自定义资源的任何新输出值。

有关 Update 请求中涉及的请求和响应对象的详细信息，请参阅自定义资源参考 (p. 526)中的更新 (p. 533)。

第 3 部分：堆栈删除

1. template developer将删除包含自定义资源的堆栈。AWS CloudFormation 将获取堆栈模板中指定的当前属性及 SNS 主题，并准备向custom resource provider发出请求。
2. AWS CloudFormation 使用 "RequestType" : "Delete" 向资源提供者发送一条 Amazon SNS 通知，其中包含有关堆栈的当前信息、堆栈模板中的自定义资源属性和用于响应的 S3 URL。

   只要删除堆栈或进行自定义资源删除或替换更新，AWS CloudFormation 都会比较新旧自定义资源的 PhysicalResourceId。如果不同，AWS CloudFormation 将更新识别为替换并向旧资源 (OldPhysicalResource) 发送删除请求，如下面的 Delete 请求示例所示。

```
{
   "RequestType" : "Delete",
   "ResponseURL" : "http://pre-signed-S3-url-for-response",
   "StackId" : "arn:aws:cloudformation:us-east-1:EXAMPLE:stack/stack-
name/guid",
   "RequestId" : "unique id for this delete request",
   "ResourceType" : "Custom::SeleniumTester",
   "LogicalResourceId" : "MySeleniumTester",
   "PhysicalResourceId" : "Tester1",
   "ResourceProperties" : {
      "seleniumTester" : "SeleniumTest()",
      "endpoints" : [ "http://mysite.com", "http://myecommercesite.com/",
"http://search.mysite.com",
         "http://mynewsite.com" ],
      "frequencyOfTestsPerHour" : [ "3", "2", "4", "3" ]
   }
}
```

`DescribeStackResource`、`DescribeStackResources` 和 `ListStackResources` 显示用户定义的名称（如果指定）。

3. custom resource provider处理由 AWS CloudFormation 发送的数据并确定 `Delete` 请求是否成功。然后，资源提供者使用 AWS CloudFormation 发送的 S3 URL 来发送 `SUCCESS` 或 `FAILED` 响应。

   以下是custom resource provider响应 `Delete` 请求的示例：

```
{
   "Status" : "SUCCESS",
   "StackId" : "arn:aws:cloudformation:us-east-1:EXAMPLE:stack/stack-
name/guid",
   "RequestId" : "unique id for this delete request",
   "LogicalResourceId" : "MySeleniumTester",
   "PhysicalResourceId" : "Tester1"
}
```

   The *`StackId`*, *`RequestId`*, and *`LogicalResourceId`* fields must be copied verbatim from the request.

4. AWS CloudFormation 将堆栈状态声明为 `DELETE_COMPLETE` 或 `DELETE_FAILED`。

有关 `Delete` 请求中涉及的请求和响应对象的详细信息，请参阅自定义资源参考 (p. 526)中的删除 (p. 531)。

另请参阅

- AWS CloudFormation 自定义资源参考 (p. 526)
- AWS::CloudFormation::CustomResource (p. 239)
- Fn::GetAtt (p. 502)
- Amazon Simple Notification Service Getting Started Guide
- Amazon Simple Storage Service 开发人员指南

使用 CloudFormer 可以从现有 AWS 资源创建 AWS CloudFormation 模板

Abstract

使用 CloudFormer 工具通过账户中的现有 AWS 资源创建 AWS CloudFormation 模板。

CloudFormer 是一个工具，用于从您的账户下的现有 AWS 资源创建 AWS CloudFormation 模板。基本程序如下：

1. 使用您的现有流程和工具预置并配置所需的资源。
2. 创建并启动一个 CloudFormer 堆栈。

   CloudFormer 本身是一个 AWS CloudFormation 堆栈。您可以通过从 AWS 环境启动堆栈来运行 CloudFormer。它在 t1.micro Amazon EC2 实例上运行，不需要其他资源。
3. 使用 CloudFormer 可以利用任何现有 AWS 资源创建一个模板，并将该模板保存到 Amazon S3 存储桶。
4. 关闭 CloudFormer 堆栈。

   此后，您通常不需要再使用 CloudFormer，因此可以将其关闭，从而终止关联的 Amazon EC2 实例，以避免产生额外费用。
5. 根据需要使用模板启动堆栈。

有关 CloudFormer 工作方式的一些常规说明：

- CloudFormer 支持所有 AWS CloudFormation 资源。
- CloudFormer 自动选择关联资源。

  例如，如果您在模板中包含了一个具有关联 EC2 安全组的 Amazon EC2 实例，则 CloudFormer 将自动选择安全组资源。
- 您可以完全控制要将哪些资源包含在模板中。

  您可以在适当时自动覆盖选定资源并添加其他资源。
- 您可以指定要在模板中使用的资源名称。

  默认名称基于现有资源名称。
- 您可以根据资源的可用属性添加输出参数。

本指南通过带您演练基本场景 – EC2 实例上一个使用多个资源创建模板的简单网站 – 介绍如何使用 CloudFormer。但是，本示例只是许多可能方案中的一个，CloudFormer 可以通过任何 AWS 资源集合创建模板。

Topics

步骤 1：创建 CloudFormer 堆栈

CloudFormer 本身就是一个 AWS CloudFormation 堆栈，因此第一步是创建并启动该堆栈。有多种方式可以执行此任务。

- AWS CloudFormation 控制台。
- CloudFormer 工具页上的 URL。
- AWS CloudFormation 模板页上的 URL

由于 AWS CloudFormation 控制台是学习如何使用 AWS 资源的好方法，因此该演练将通过使用此控制台来启动 CloudFormer 堆栈。

使用 AWS CloudFormation 控制台创建 CloudFormer 堆栈

1. 登录 AWS CloudFormation 控制台，然后单击 Create New Stack (创建新堆栈) 以启动堆栈创建向导。有关如何登录的说明，请参阅登录 AWS CloudFormation 控制台。
2. 在向导的 Create Stack (创建堆栈) 页上，执行以下操作：

   1. 在 Name (名称) 框中，指定此 CloudFormer 堆栈的名称。
   2. 在 Template (模板) 部分中，选择 Use a sample template (使用示例模板)，然后从列表中选择 CloudFormer - create a template from your existing resources (CloudFormer - 利用您的现有资源创建模板)。

   单击 Next Step (下一步)，进入下一页。
3. 在 Specify Parameters (指定参数) 屏幕上，执行以下操作：

   - 在访问控制下，指定可用于访问该工具的 IP 地址范围。

     默认 IP 地址范围为 0.0.0.0/0，在此设置下，该工具完全开放。我们建议您指定一个更具限制性的地址范围。
4. 单击 Next Step (下一步)。
5. 选中 I acknowledge that this template may create IAM resources (我确认，此模板可创建 IAM 资源) 复选框，然后单击 Next Step (下一步)。

   此示例不使用标签。
6. 在 Review (审核) 屏幕上，检查即将创建的堆栈的相关信息，然后单击 Create (创建)，开始创建 CloudFormer 堆栈。

   注意：CloudFormer 本身就是一个 AWS CloudFormation 堆栈，因此必须完成正常的堆栈创建过程，这需要几分钟。

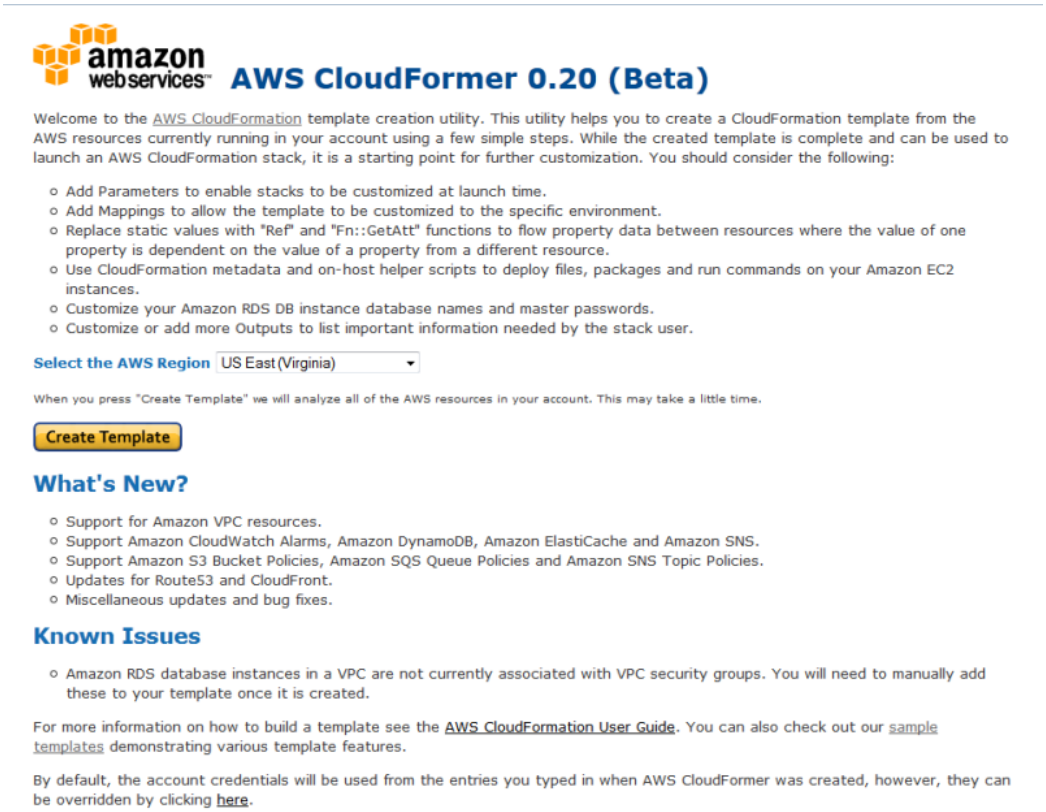
步骤 2：启动 CloudFormer 堆栈

在 CloudFormer 堆栈的状态变为 CREATE_COMPLETE 后，您就可以启动堆栈了。

启动 CloudFormer 堆栈

1. 在 AWS CloudFormation 控制台中单击 CloudFormer 堆栈的条目，然后选择堆栈信息窗格中的 Outputs (输出) 选项卡。
2. 在 Outputs Value (输出值) 列中，单击相应的 URL 以启动 CloudFormer 堆栈。

堆栈启动后，您的浏览器中将显示 CloudFormer 工具的首页，您可以按照下一节中的说明，使用该页面创建模板。



Note

CloudFormer 堆栈将启动一个 t1.micro Amazon EC2 实例，您必须在完成后手动终止该实例。

CloudFormer 堆栈创建后，将成为您的账户的堆栈集合之一。要创建其他模板，只需要重新启动 CloudFormer 堆栈。

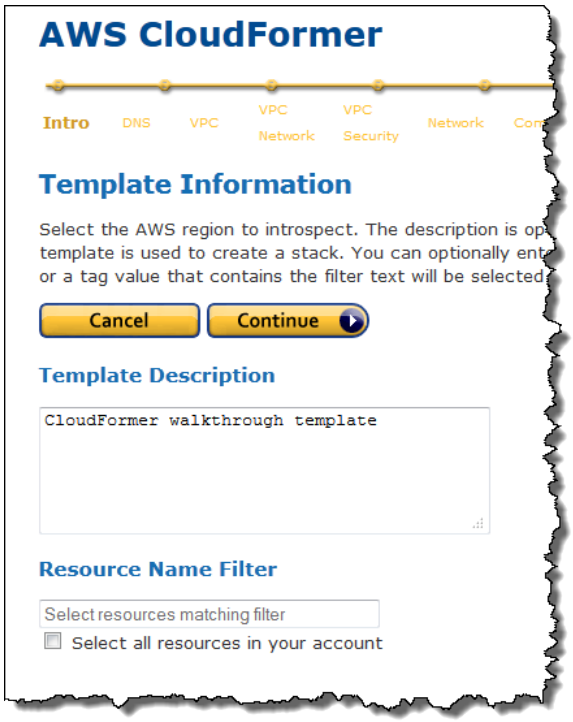
步骤 3：使用 CloudFormer 创建模板

在开始使用 CloudFormer 创建模板之前，请先确保您的账户包含您希望纳入模板的所有 AWS 资源。该演练假设您的账户包含：

- 一个 Amazon EC2 实例 (`AWS::EC2::Instance`)。
- 一个 Amazon EC2 安全组 (`AWS::EC2::SecurityGroup`)。您应该将该安全组与实例关联起来。
- 一个弹性 IP 地址 (`AWS::EC2::EIP`)。您应该将该地址与实例关联起来。

使用 CloudFormer 通过 AWS 资源创建模板

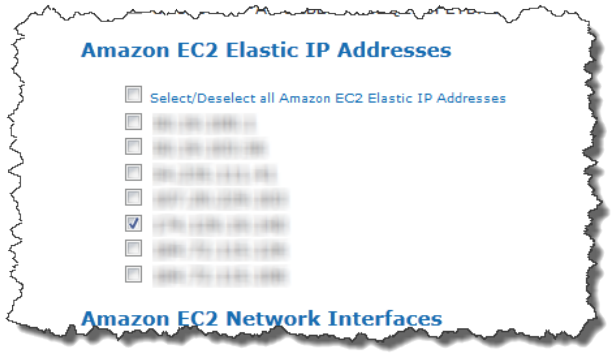
1. 在 Select the AWS Region (选择 AWS 区域) 下，从列表中选择模板的区域，然后单击 Create Template (创建模板)。该工具必须先分析您的账户，因此可能要在几分钟后才能显示 Intro (简介) 页。
2. 在 Intro (简介) 页上，输入对您的模板的说明。您也可以使用筛选条件在此页面上选择资源，或者选择您账户中的所有资源。此演练手动指定资源，因此请将 Resource Name Filter (资源名称筛选条件) 和 Select all resources in your account (选择您账户中的所有资源) 留空并分别清除，然后单击 Continue (继续)。



3. 以下页面针对本演练中未使用的资源，因此只需查看以供日后参考，然后单击 Continue (继续)。按顺序分别为：

   1. DNS Names (DNS 名称) 页，允许您包含 Route 53 记录。
   2. Virtual Private Cloud 页允许您包含 Amazon VPC。
   3. Virtual Private Cloud Network Topologies (Virtual Private Cloud 网络拓扑) 页，允许您包含 Amazon VPC 子网、网关、DHCP 配置和 VPN 连接。
   4. Virtual Private Cloud Security Configuration (Virtual Private Cloud 安全配置) 页，允许您包含网络 ACL 和路由表。

4. Network Resources (网络资源) 页，允许您包含 Elastic Load Balancing 负载均衡器、弹性 IP 地址、CloudFront 分配和 Amazon EC2 网络接口。选择要包含在模板中的弹性 IP 地址。



5. Compute Resources (计算资源) 页，允许您包含 Auto Scaling 组和 Amazon EC2 实例。在开始创建模板之前，您已将一个弹性 IP 地址与您的 Amazon EC2 实例相关联，以创建一个关联资源。当您到达 Compute Resources (计算资源) 页时，CloudFormer 将自动选择关联实例，因此只需确保您的实例已被选定，然后单击 Continue (继续) 即可。

Note

您可以根据需要手动包含额外实例。如果您不希望包含自动选定的实例，只需清除复选框。

6. 以下页面针对本演练中未使用的资源，因此只需查看以供日后参考，然后单击 Continue (继续)。按顺序分别为：

   1. Storage (存储) 页，允许您包含 Amazon EBS 卷、Amazon RDS 实例、Amazon DynamoDB 表和 Amazon S3 存储桶。
   2. Application Services (应用程序服务) 页，允许您包含Amazon ElastiCache 集群、Amazon SQS 队列、Amazon SimpleDB 域 和 Amazon SNS 主题。

      System Configuration (系统配置) 页，允许您包含 Auto Scaling 启动配置、Amazon RDS 子网组、Amazon ElastiCache 参数组和 Amazon RDS 参数组。

7. Security Groups (安全组) 页，允许您包含安全组。在开始创建模板之前，您已将一个 Amazon EC2 安全组与您的 Amazon EC2 实例相关联，以创建一个关联资源。当您到达 Security Groups (安全组) 页时，CloudFormer 将自动选择关联安全组，因此只需确保您的组已被选定，然后单击 Continue (继续) 即可。

Note

您可以根据需要手动包含其他安全组，包括 Amazon EC2 安全组、Amazon RDS 安全组等。如果您不希望包含手动选定的安全组，只需清除复选框。

8. Operational Resources (操作资源) 页，允许您包含 Auto Scaling 策略和 Amazon CloudWatch 警报。本演练不使用其中的任何一个，因此只需单击 Continue (继续)。
9. Summary (概要) 页有多项用途：

   - 它允许您审核添加到模板中的资源。

     要修改您的资源，请单击 Back (返回) 以返回至相应页面，然后根据需要修改您的选择。
   - 它允许您更改自动生成并分配给您的资源的逻辑名称。

     要修改逻辑名称，请单击 Modify (修改)，然后在 Logical Name (逻辑名称) 字段中输入名称。

- 它允许您指定提供必要信息（如站点的 IP 地址或 URL）的输出。

  要修改输出，请单击 Modify (修改)，然后从列表中选择适当的输出。

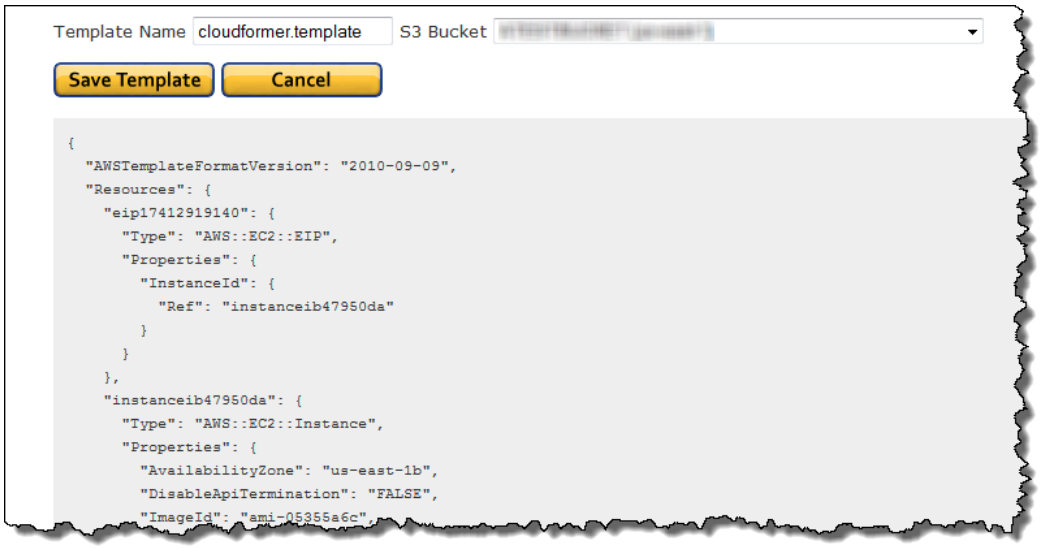
检查已选定的资源并进行必要的更改。您应该有一个弹性 IP 地址、一个 Amazon EC2 实例和一个 Amazon EC2 安全组。如果您对资源感到满意，请单击 Continue (继续) 以生成模板。

10. 此时，AWS CloudFormation Template (AWS CloudFormation 模板) 页将显示生成的模板。您可以使用该模板将资源部署为 AWS CloudFormation 组合集，或者将其用作供进一步修改的基本模板。

Note

除了显式指定的资源外，该模板还包括这些资源的关联值，如 Amazon EC2 实例的可用区。

从 S3 Bucket (S3 存储桶) 列表中选择一个 Amazon S3 存储桶，然后单击 Save Template (保存模板) 以将模板保存到该存储桶中并将其添加到您的堆栈账户集合中。



Save Template (保存模板) 为您提供了两个选项：

- Launch Stack (启动堆栈) 可将模板保存到指定的 Amazon S3 存储桶，同时立即启动堆栈。
- Create Template (创建模板) 仅将模板保存到指定的 Amazon S3 存储桶中。

  稍后，您可以像启动其他模板一样启动该堆栈，例如，使用 AWS CloudFormation 控制台启动。

11. 您有了模板，就不再需要CloudFormer堆栈了。为了避免您的账户产生不必要的费用，请转至Amazon EC2 控制台并删除 CloudFormer Amazon EC2 实例。

使用 AWS Identity and Access Management 控制访问

Abstract

使用 AWS Identity and Access Management 用户权限来控制可以在 AWS CloudFormation 中访问您的 AWS 资源的人员。

使用 AWS Identity and Access Management (IAM)，您可以创建 IAM 用户来控制哪些用户可以访问您 AWS 账户中的哪些资源。您可以使用 IAM 和 AWS CloudFormation 控制用户可以执行哪些 AWS CloudFormation 操作，如查看堆栈模板、创建堆栈或删除堆栈。此外，您还可以管理每个用户可用的 AWS 服务和资源。这样，您就可以控制用户在创建、更新或删除堆栈时可以创建或更新的资源。例如，您可以指定哪些用户可以启动 Amazon EC2 实例、终止数据库实例或更新 VPC。

有关您可以控制访问权限的所有服务的更多信息，请参阅*使用 IAM*中的支持 IAM 的 AWS 服务。

AWS CloudFormation 操作和资源

当您在 AWS 账户中创建组或 IAM 用户时，可以将一个 IAM 策略与所创建的组或用户关联起来。策略指定 IAM 用户对哪些堆栈有什么权限。例如，假设您有一个入门级开发人员组。您可以创建一个 `Junior application developers` 组并将每个入门级开发人员的 IAM 用户包含在其中。然后，将一个只允许用户查看 AWS CloudFormation 堆栈的策略与该组关联。在这种情况下，您可能有一个类似下面示例的策略：

授予查看堆栈权限的示例策略

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "cloudformation:DescribeStacks",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStackResources"
        ],
        "Resource":"*"
    }]
}
```

该策略授予 Action 元素中列出的所有操作的权限。在 Resource 元素中，我们指定了一个星号 (*)，这是一个通配符，它允许对所有 AWS CloudFormation 堆栈执行这些操作。

除 AWS CloudFormation 操作之外，创建或删除堆栈的 IAM 用户还需要与给定 AWS CloudFormation 模板中的资源相关的其他操作权限。例如，如果您有一个说明 Amazon SQS 队列的模板，用户必须具有 Amazon SQS 操作的相应 IAM 权限才能成功创建堆栈，如下面的示例策略所示：

授予创建和查看堆栈操作以及所有 Amazon SQS 操作的策略示例

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "sqs:*",
            "cloudformation:CreateStack",
            "cloudformation:DescribeStacks",
            "cloudformation:DescribeStackEvents",
            "cloudformation:DescribeStackResources",
            "cloudformation:GetTemplate",
            "cloudformation:ValidateTemplate"
        ],
        "Resource":"*"
    }]
}
```

AWS CloudFormation 还支持资源级权限，因此您可以指定对某一特定堆栈的操作，如下面的策略所示：

拒绝 MyProductionStack 的删除和更新堆栈操作的策略示例

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Deny",
        "Action":[
            "cloudformation:DeleteStack",
            "cloudformation:UpdateStack"
        ],
        "Resource":"arn:aws:cloudformation:us-east-1:123456789012:stack/MyProductionStack/*"
    }]
}
```

示例策略在堆栈名称末尾使用通配符，因此拒绝对整个堆栈 ID（如 arn:aws:cloudformation:us-east-1:123456789012:stack/MyProductionStack/abc9dbf0-43c2-11e3-a6e8-50fa526be49c）和堆栈名称（如 `MyProductionStack`）执行删除堆栈和更新堆栈操作。

有关您可以允许或拒绝的所有 AWS CloudFormation 操作的列表，请参阅 *AWS CloudFormation API 参考*。

AWS CloudFormation 条件

在 IAM 策略中，您可以选择指定控制策略生效时间的条件。AWS CloudFormation 没有特定于服务的条件。但是，您可以使用 AWS 范围的条件，如 `DateLessThan`，该条件指定策略停止生效的时间。有关 AWS 范围的条件的更多信息，请参阅*使用 IAM* 中的 IAM 策略元素参考。

Note

请勿使用 `aws:SourceIp` 条件。AWS CloudFormation 使用自身的 IP 地址而不是原始请求的 IP 地址配置资源。例如，在创建堆栈时，AWS CloudFormation 从它的 IP 地址发送请求来启动 Amazon EC2 实例或创建 Amazon S3 存储桶，而不是来自 `CreateStack` 调用或 `aws cloudformation create-stack` 命令的 IP 地址。

AWS CloudFormation 模板中的 IAM 资源

AWS CloudFormation 验证模板后，您才能创建堆栈。在验证期间，AWS CloudFormation 还会检查模板是否具有您应了解的 AWS 资源。目前，AWS CloudFormation 只检查您模板中的 IAM 资源。我们建议您检查与每个 IAM 资源关联的权限。IAM 资源（如具有完全访问权限的 IAM 用户）可以访问和修改您的 AWS 账户中的所有资源。为确保您检查了所有 IAM 资源，您必须在 AWS CloudFormation 创建堆栈前确认模板创建这些资源。

您可以使用 AWS CloudFormation 控制台、命令行或 API 确认 AWS CloudFormation 模板的功能：

- 在 AWS CloudFormation 控制台中，在“Create Stack (创建堆栈)”或“Update Stack (更新堆栈)”向导的 Specify Parameters (指定参数) 页上，选择 I acknowledge that this template may create IAM resources (我确认，此模板可创建 IAM 资源)。
- 对于 AWS Command Line Interface，在使用 `aws cloudformation create-stack` 和 `aws cloudformation update-stack` 命令时为 *`--capabilities`* 参数指定 *`CAPABILITY_IAM`* 值。
- 对于 API，在使用 CreateStack 和 UpdateStack 操作时指定 *`Capabilities.member.1=CAPABILITY_IAM`* 参数。

管理 Amazon EC2 实例上运行的应用程序的证书

如果您有一个应用程序在 Amazon EC2 实例上运行并且需要请求 AWS 资源（如 Amazon S3 存储桶或 Amazon DynamoDB 表），应用程序会要求提供 AWS 安全证书。但是，在启动的每个实例中分配并嵌入长期安全证书非常困难，而且存在潜在安全风险。不必使用类似 IAM 用户证书的长期证书，我们建议您创建一个在启动 Amazon EC2 实例时使用的 IAM 角色。然后，应用程序可以从 Amazon EC2 实例获取临时安全证书。您不必在实例上嵌入长期证书。此外，为简化证书管理工作，您可以为多个 Amazon EC2 实例指定一个角色；而无需为每个实例创建唯一证书。

有关演示如何使用角色启动实例的模板代码段，请参阅 IAM 角色模板示例 (p. 152)。

Note

使用临时安全证书的实例上的应用程序可以调用任何 AWS CloudFormation 操作。但是，因为 AWS CloudFormation 与很多其他 AWS 服务交互，您必须确定要使用的所有服务都支持临时安全证书。有关更多信息，请参阅支持 AWS STS 的 AWS 服务。

使用 IAM 角色授予临时访问权限

在某些情况下，您可能希望授予没有 AWS 证书的用户对您的 AWS 账户的临时访问权限。不必在每次授予临时访问权限时创建和删除长期证书，您可以使用 IAM 角色。通过一个 IAM 角色，您可以通过编程方式创建并分配很多临时安全证书（包括访问密钥、私有访问密钥和安全令牌）。这些证书的有效期有限，过期后不能用于访问您的 AWS 账户。您也可以创建多个 IAM 角色，向每个用户授予不同级别的权限。IAM 角色对于联合身份和单一登录等情况非常有用。

联合身份是可以跨多个系统使用的独特身份。对于已建立本地身份系统（如 LDAP 或 Active Directory）的企业用户，可以使用本地身份系统处理所有身份验证。用户经过身份验证后，您从相应的 IAM 角色提供临时安全证书。例如，您可以创建一个 `administrators` 角色和一个 `developers` 角色，其中 administrators 对 AWS 账户拥有完全访问权限，developers 只拥有使用 AWS CloudFormation 堆栈的权限。经过身份验证后，管理员有权获取 `administrators` 角色的临时安全证书。而开发人员只能获取 `developers` 角色的临时安全证书。

您还可以授予联合用户对 AWS Management Console的访问权限。使用本地身份系统对用户进行身份验证后，可以通过编程构造一个临时 URL 提供对AWS Management Console 的直接访问。用户使用临时 URL 时无需登录 AWS，因为已经过身份验证（单一登录）。此外，因为 URL 是从用户的临时安全证书构造的，所以通过这些证书提供的权限确定用户在AWS Management Console 中拥有的权限。

Note

AWS CloudFormation 与许多其他 AWS 服务交互。对 AWS CloudFormation 使用临时安全证书时，请确保要使用的所有服务都支持临时安全证书。有关更多信息，请参阅支持 AWS STS 的 AWS 服务。

有关更多信息，请参阅 *使用临时安全证书* 中的以下相关资源：

- 授予临时访问权限的情形
- 向联合用户授予直接访问 的权限AWS Management Console

AWS CloudFormation 堆栈更新

Abstract

通过提交模板或指定堆栈中资源更新的输入参数，来更新现有 AWS CloudFormation 堆栈。

您可以更新已成功创建的堆栈，以更新堆栈中的资源（如 Amazon EC2 实例），或更新堆栈的设置（如堆栈的 Amazon SNS 通知主题）。例如，如果堆栈包含一个 Amazon EC2 实例，则您可以通过更新堆栈来更新该实例。您无需创建新堆栈。您可以使用 AWS CloudFormation 控制台、`aws cloudformation update-stack` CLI 命令或 `UpdateStack` API 来更新堆栈。

对堆栈资源的更新

您可通过提交更新的模板或提交更新的输入参数来修改堆栈资源。提交更新时，AWS CloudFormation 会基于提交的内容与堆栈当前模板之间的差异来更新资源。尚未更改的资源在更新过程中会不中断地运行。根据所更新的资源和属性，可能会中断或替换更新的资源。AWS CloudFormation 使用以下方法之一更新资源：

无中断更新
: AWS CloudFormation 更新资源时不会中断该资源的运行，也不会更改该资源的物理名称。例如，如果您更新了 AWS::CloudWatch::Alarm (p. 257) 资源的任何属性，AWS CloudFormation 将更新警报的配置，但警报在更新期间将继续运行，不会出现中断。

时而中断更新
: AWS CloudFormation 更新资源时会时而中断，但保留物理名称。例如，如果您对 AWS::EC2::Instance (p. 272) 资源更新特定属性，则在 AWS CloudFormation 和 Amazon EC2 重新配置实例期间，该实例可能有时会中断。

替换
: AWS CloudFormation 会在更新过程中重新创建资源，这还会生成新的物理 ID。AWS CloudFormation 会先创建替换资源，将对其他相关资源的引用更改为指向替换资源，然后删除旧资源。例如，如果您更新 AWS::RDS::DBInstance (p. 381) 资源的 `Engine` 属性，则 AWS CloudFormation 会创建新资源并将当前 DBInstance 资源替换为新资源。

要了解有关更新特定资源的更多信息，请参阅与该资源关联的文档。例如，Amazon EC2 文档提供有关哪些更改会中断实例的详细信息。另请参阅 AWS 资源类型参考 (p. 217)，其中针对每个属性列出了更新资源的影响。

根据 AWS CloudFormation 用于修改堆栈中的每个更新资源的方法，您可以明智地决定修改资源的最佳时间，以减小这些更改对您的应用程序产生的影响。具体来说，您可以计划好更新过程中必须 *替换* 资源的时间。例如，如果您更新了 AWS::RDS::DBInstance 资源的 Port 属性，AWS CloudFormation 将使用更新的端口设置和新的物理名称创建一个新的数据库实例。为对此进行计划，您应执行以下操作：

1. 拍摄当前数据库的快照。
2. 准备一个策略，指定使用该数据库实例的应用程序在数据库实例替换期间将如何处理中断。
3. 确保使用该数据库实例的应用程序考虑更新的端口设置以及您进行的任何其他更新。
4. 使用数据库快照在新数据库实例上还原数据库。

该示例并不详尽；它旨在让您了解针对在更新过程中替换资源的情况，需要计划的事项。

Note

如果模板包括一个或多个嵌套堆栈 (p. 250)，则 AWS CloudFormation 也会为每个嵌套堆栈启动更新。这对于确定嵌套堆栈是否已修改是必要的。AWS CloudFormation 只更新嵌套堆栈中那些在相应模板中指定了更改的资源。

Topics

修改堆栈模板

如果您要修改在堆栈模板中声明的资源和属性，则必须修改堆栈的模板。要确保仅更新计划更新的资源，请将现有堆栈的模板用作起点，然后对该模板进行更新。如果您要在源控制系统中管理模板，请使用该模板的副本作为起点。否则，您可以从 AWS CloudFormation 获取堆栈模板的副本。

如果您要仅修改堆栈的参数或设置（如堆栈的 Amazon SNS 主题），则可以重用现有堆栈模板。您无需获取堆栈模板的副本或是对堆栈模板进行任何修改。

Note

如果您的模板包含不受支持的更改，AWS CloudFormation 将返回一则消息，告知不允许进行该更改。但该消息可能是异步出现的，因为默认情况下，AWS CloudFormation 创建和更新资源的顺序是不确定的。

使用控制台从 AWS CloudFormation 获取并修改堆栈的模板

1. 在 AWS CloudFormation 控制台中，选择要更新的堆栈，然后单击 Template (模板) 选项卡以查看堆栈模板。

2. 从 Template (模板) 选项卡中将模板复制到一个文本文件中。
3. 修改该模板文件，然后保存它。仅修改您要更新的资源。对于不更新的资源和属性，使用与当前堆栈配置*相同的值*。您可以通过完成以下任何操作来修改模板：

   - 添加新资源，或删除现有资源。

对于大多数是资源，更改资源的逻辑名称相当于删除该资源并将其替换为新资源。与重命名的资源关联的任何其他资源也需要更新，并且可能会导致它们被替换。其他资源需要您更新属性（不仅仅是逻辑名称）以触发更新。

- 添加、修改或删除现有资源的属性。

  有关更新特定资源属性的影响的信息，请参阅 AWS 资源类型参考 (p. 217)。对于每种属性，更新将产生以下一种影响：

  - *更新要求*：无中断 (p. 63)
  - *更新要求*：时而中断 (p. 63)
  - *更新要求*：替换 (p. 63)
- 添加、修改或删除资源的属性（Metadata、DependsOn 和 DeletionPolicy）。

  Important

  对 DeletionPolicy 或输出值声明的更改不能自行更新。只有在添加、修改或删除现有资源的属性，或者更改资源元数据时才能更改 DeletionPolicy 或输出值声明。

- 添加、修改或删除参数声明。但是，您无法添加、修改或删除不支持更新的资源使用的参数。
- 添加、修改或删除输出值声明。
- 添加、修改或删除映射声明。

一些资源或属性可能在属性值或这些值的更改方面有一些约束条件。例如，对 AWS::RDS::DBInstance (p. 381) 资源的 AllocatedStorage 属性的更改必须大于当前设置，如果为更新指定的值未满足这些约束条件，该更新将失败。有关 AllocatedStorage 更改的特定约束条件，请参阅 ModifyDBInstance。

资源更新可能对影响其他资源的属性。如果您使用 Ref 函数 (p. 508)或 Fn::GetAtt 函数 (p. 502)将已更新资源的属性指定为模板中其他资源的属性值的一部分，则 AWS CloudFormation 还将更新包含对已更改属性的引用的资源。例如，如果您更新了 AWS::RDS::DBInstance 资源的 MasterUsername 属性，并且拥有一个具有 UserData 属性（包含对使用 Ref 函数的数据库实例名称的引用）的 AWS::AutoScaling::LaunchConfiguration 资源，则 AWS CloudFormation 将使用新名称重新创建数据库实例并更新 LaunchConfiguration 资源。

4. 如果您希望在更新堆栈时将模板指定为 URL，请将更新模板上传至 Amazon S3 存储桶。存储桶必须处于与您更新的堆栈相同的区域中。

使用命令行从 AWS CloudFormation 获取并修改堆栈的模板

1. 使用命令 `aws cloudformation get-template` 获取要更新的堆栈的模板。
2. 将模板复制并粘贴到文本文件中，进行修改，然后保存它。确保您仅复制模板。该命令将模板括在引号中，但是不要复制模板周围的引号。模板本身以左括号开始，以右括号结束。在此文件中指定对堆栈资源的更改。

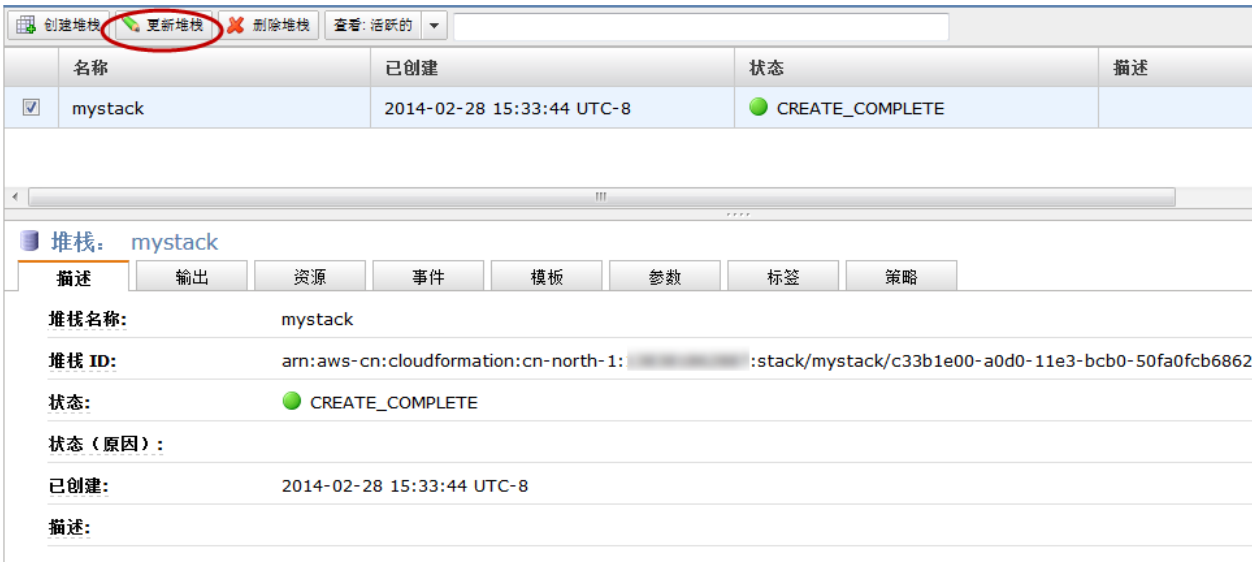
更新堆栈

更新堆栈时，您可以修改堆栈中的资源、更新堆栈设置或同时执行两种操作。例如，您可以通过更改实例类型来增加 Amazon EC2 实例的容量，也可以更新堆栈的 Amazon SNS 通知主题。

更新堆栈时，您可以更改支持更新的资源使用的参数值；但是，您必须保留当前堆栈中会对不支持更新的资源产生影响的参数的现有值。

使用控制台更新现有 AWS CloudFormation 堆栈

1. 在 AWS CloudFormation 控制台中，从堆栈列表中选择您要更新的正在运行的堆栈。
2. 单击 Update Stack (更新堆栈)。



3. 根据您是否修改了堆栈模板，您可以重用现有模板或指定其他模板。

   - 如果您未修改堆栈模板，请选择 Use existing template (使用现有模板)。
   - 如果您修改了堆栈模板，请指定更新的模板的位置：
     - 对于在计算机上本地存储的模板，选择 Upload a template to Amazon S3 (将模板上传到 Amazon S3)。输入模板文件的位置，或者单击 Browse (浏览) 以导航至该文件并选择它，然后单击 Next (下一步)。
     - 对于在 Amazon S3 存储桶中存储的模板，选择 Specify an Amazon S3 URL (指定 Amazon S3 URL)。输入或粘贴模板的 URL，然后单击 Next (下一步)。

4. 在 Specify Parameters (指定参数) 页上，输入或修改参数值，然后单击 Next (下一步)。

   AWS CloudFormation 会使用当前在堆栈中设置的值填充每个参数（使用 `NoEcho` 属性声明的参数例外）；但是，您可以通过选择 Use existing value (使用现有值)，仍然使用现有值。
5. 在 Options (选项) 页面上，您可以输入覆盖堆栈策略或更新 Amazon SNS 通知主题。覆盖堆栈策略使您可以更新受保护的资源。有关更多信息，请参阅 防止更新堆栈资源 (p. 69)。

   修改完所有选项之后，单击 Next (下一步)。
6. 审核堆栈信息。如果模板中有 IAM 资源，请选中 I acknowledge that this template may create IAM resources (我确认，此模板可创建 IAM 资源) 以指定您要使用模板中的 IAM 资源。有关使用模板中的 IAM 资源的更多信息，请参阅使用 AWS Identity and Access Management 控制访问 (p. 59)。
7. 单击 Update (更新)。

   您的堆栈将进入 UPDATE_IN_PROGRESS 状态。完成更新后，堆栈状态将设置为 UPDATE_COMPLETE。

   如果堆栈更新失败，则 AWS CloudFormation 会自动回滚任何更改，堆栈将设置为 UPDATE_ROLLBACK_COMPLETE。

Note

堆栈开始更新后，如果它仍处于 UPDATE_IN_PROGRESS 状态，那么您可以取消更新。有关更多信息，请参阅 取消堆栈更新 (p. 68)。

使用命令行更新现有 AWS CloudFormation 堆栈

- 使用命令 `aws cloudformation update-stack` 通过指定要更新的堆栈、更新后的模板、参数值和容量，来更新堆栈。

  以下示例更新堆栈命令更新 `mystack` 堆栈的模板和输入参数：

```
PROMPT> aws cloudformation update-stack --stack-name mystack --template-url
 https://s3.amazonaws.com/sample/updated.template
--parameters ParameterKey=VPCID,ParameterValue=SampleVPCID ParameterKey=Sub
netIDs,ParameterValue=SampleSubnetID1\\,SampleSubnetID2
```

  以下示例更新堆栈命令仅更新 `mystack` 堆栈的 `SubnetIDs` 参数值：

```
PROMPT> aws cloudformation update-stack --stack-name mystack --use-previous-
template
--parameters ParameterKey=VPCID,UsePreviousValue=true ParameterKey=Subnet
IDs,ParameterValue=SampleSubnetID1\\,UpdatedSampleSubnetID2
```

  以下示例更新堆栈命令向 `mystack` 堆栈添加两个堆栈通知主题：

```
PROMPT> aws cloudformation update-stack --stack-name mystack --use-previous-
template
--notification-ar-ns "arn:aws:sns:us-east-1:12345678912:mytopic"
"arn:aws:sns:us-east-1:12345678912:mytopic2"
```

  以下示例更新堆栈命令从 `mystack` 堆栈中删除所有堆栈通知主题：

```
PROMPT> aws cloudformation update-stack --stack-name mystack --use-previous-
template
--notification-ar-ns []
```

监控堆栈更新的进度

您可以通过查看堆栈事件来监控堆栈更新进度。控制台的 Events (事件) 选项卡会显示堆栈创建和更新过程中的每个重要步骤（按照每个事件的时间进行排序，最新的事件显示在最上方）。堆栈更新流程开始有堆栈 UPDATE_IN_PROGRESS 事件作为标记：

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_IN_PROGRESS
```

接下来显示的事件标明更新模板中已发生更改的各个资源的更新开始和完成。例如，更新名为 MyDB 的 AWS::RDS::DBInstance (p. 381) 资源将生成以下条目：

```
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

UPDATE_IN_PROGRESS 事件，在 AWS CloudFormation 报告它已开始更新资源时记录。UPDATE_COMPLETE 事件，在资源创建成功时记录。

AWS CloudFormation 成功更新堆栈时，您会看到以下事件：

```
2011-09-30 09:35 PDT AWS::CloudFormation::Stack MyStack UPDATE_COMPLETE
```

如果资源更新失败，AWS CloudFormation 将报告一个包含失败原因的 UPDATE_FAILED 事件。例如，如果您的更新模板指定了资源不支持的属性更改，如缩减 AWS::RDS::DBInstance (p. 381) 资源的 AllocatedStorage 大小，您将看到如下事件：

```
2011-09-30 09:36 PDT AWS::RDS::DBInstance MyDB UPDATE_FAILED Size cannot be
less than current size; requested: 5; current: 10
2011-09-30 09:35 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
```

如果资源更新失败，AWS CloudFormation 会将升级期间已更新的任何资源回滚到其更新前的配置。下面列出了您将在更新回滚期间看到的事件示例：

```
2011-09-30 09:38 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_COMPLETE
2011-09-30 09:38 PDT AWS::RDS::DBInstance MyDB UPDATE_COMPLETE
2011-09-30 09:37 PDT AWS::RDS::DBInstance MyDB UPDATE_IN_PROGRESS
2011-09-30 09:37 PDT AWS::CloudFormation::Stack MyStack UPDATE_ROLLBACK_IN_PRO
GRESS The following resource(s) failed to update: [MyDB]
```

使用控制台查看堆栈事件

1. 在 AWS CloudFormation 控制台中，选择已更新的堆栈，然后单击 Events (事件) 选项卡以查看堆栈事件。
2. 要使用最新事件更新事件列表，请单击 AWS CloudFormation 控制台中的刷新按钮。

使用命令行查看堆栈事件

- 使用 `aws cloudformation describe-stack-events` 命令查看堆栈的事件。

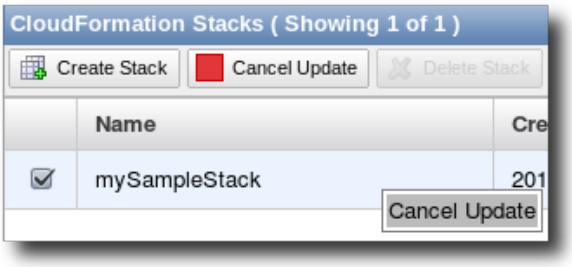
取消堆栈更新

堆栈更新开始后，如果堆栈仍处于 UPDATE_IN_PROGRESS 状态，则您可以取消堆栈更新。更新完成之后，将无法取消它。但是，您可以通过任何先前的设置再次更新堆栈。

如果您取消堆栈更新，那么堆栈将回滚到启动堆栈更新之前已存在的堆栈配置。

使用控制台取消堆栈更新

1. 在 AWS CloudFormation 控制台上的堆栈列表中，选择当前正在更新的堆栈（其状态必须为 `UPDATE_IN_PROGRESS`）。
2. 单击 Cancel Update (取消更新)。



3. 要继续取消更新，请在出现提示时单击 Yes, Cancel Update (是，取消更新)。否则，单击 Cancel (取消) 以恢复更新。

堆栈将进入 UPDATE_ROLLBACK_IN_PROGRESS 状态。更新取消过程完成后，堆栈状态将设置为 UPDATE_ROLLBACK_COMPLETE。

使用命令行取消堆栈更新

- 使用命令 `aws cloudformation cancel-update-stack` 可取消更新。

防止更新堆栈资源

Abstract

使用堆栈策略可防止堆栈资源在堆栈更新过程中意外更新或删除。

使用堆栈策略可以防止堆栈资源在堆栈更新过程中意外更新或删除。堆栈策略仅适用于堆栈更新过程中，并且只应用作故障保护功能来防止意外更新特定堆栈资源。请勿使用堆栈策略控制对 AWS 资源或操作的访问；而应使用 AWS Identity and Access Management (IAM)。

默认情况下，具有更新权限的用户可以更新堆栈中的所有资源。但是，在更新期间，一些资源可能需要中断或可能已完全替换，这可能会导致新的物理 ID 或全新的存储。为确保无人意外更新这些资源，您可以设置堆栈策略。堆栈策略可以防止他人意外更新受保护的资源。如果要更新受保护的资源，必须在堆栈更新过程中明确指定这些资源。

Important

设置堆栈策略后，将默认保护堆栈中的所有资源。必须对不想保护的资源指定 `Allow` 语句。

堆栈策略是定义对指定资源可以执行的更新操作的 JSON 文档。您只能为每个堆栈定义一个堆栈策略；不过，在一个策略中可以保护多个资源。下面的示例堆栈策略可以防止更新逻辑 ID 为 `ProductionDatabase` 的资源：

```
{
  "Statement" : [
```

```
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/ProductionDatabase"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

在示例中，我们需要防止更新，因此为 `Effect` 元素指定 `Deny` 来防止 `Action` 元素中定义的所有操作。为指定所有更新操作，我们使用了通配符（星号）。在 `Resource` 元素中，我们指定要保护的资源，即逻辑 ID 为 `ProductionDatabase` 的资源。`Principal` 元素是必需的，但它只支持通配符 (*)。

请注意，当您设置堆栈策略时，将默认保护所有资源。因此，要只保护 `ProductionDatabase` 资源，我们添加一条 `Allow` 语句来包含所有操作和资源，以便可以更新所有其他资源。即使 `Allow` 指定了所有资源，显式 `Deny` 也会覆盖所有允许。

堆栈策略适用于要更新堆栈的所有用户。您不能应用引用特定用户的堆栈策略。必须更新受保护资源的用户需要拥有使用 `SetStackPolicy` 操作的权限。有关更多信息，请参阅 更新受保护资源 (p. 71)。

有关堆栈策略语法的详细信息，请参阅堆栈策略参考 (p. 73)。

设置堆栈策略

如果需要保护堆栈资源防止意外更新，可以定义一个 JSON 格式的堆栈策略，然后在创建或更新堆栈时将其与堆栈关联起来。有关如何编写堆栈策略的更多信息，请参阅堆栈策略参考 (p. 73)。

默认情况下，堆栈没有堆栈策略，所以您可以更新任意堆栈资源。但是，在设置堆栈策略后，所有堆栈资源都默认受到保护，除非您为不想保护的资源包含 `Allow` 语句。

在创建堆栈时设置堆栈策略：

AWS Management Console

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.amazonaws.cn/cloudformation/。
2. 在 CloudFormation Stacks (CloudFormation 堆栈) 页上，单击 Create Stack (创建堆栈)。

3. 在 Create Stack (创建堆栈) 向导的 Options (选项) 屏幕上，展开 Advanced (高级) 部分。

Note

当您创建堆栈并包含策略时，无需权限就能使用 AWS CloudFormation `SetStackPolicy` 操作。但是，如果需要更新策略或更新受保护的资源，必须拥有权限才能使用 `SetStackPolicy` 操作。

4. 选择定义堆栈策略的文件或输入一个。

CLI

- 使用 `aws cloudformation create-stack` 命令，使用 `--stack-policy-body` 或 `--stack-policy-url` 选项。

修改堆栈策略（目前您不能使用控制台执行此操作）：

CLI

- 使用 `aws cloudformation set-stack-policy` 命令，使用 `--stack-policy-body` 或 `--stack-policy-url` 选项。

在堆栈更新期间修改堆栈策略（目前您不能使用控制台执行此操作）：

CLI

- 使用 `aws cloudformation update-stack` 命令，使用 `--stack-policy-body` 或 `--stack-policy-url` 选项。

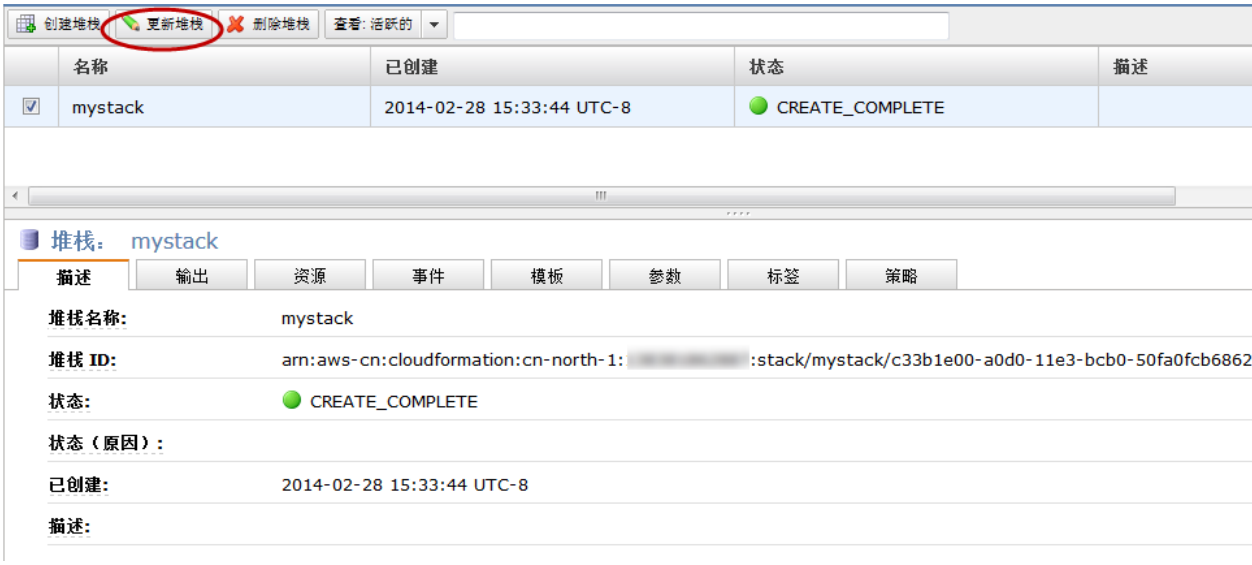
更新受保护资源

您可以使用临时策略解除对受保护资源的保护来对它们进行更新。在临时策略中指定要更新的资源。在更新堆栈时定义临时策略。在开始前，您必须拥有足够的权限才能使用 AWS CloudFormation `SetStackPolicy` 操作和定义允许更新要更改的资源的临时策略。

更新受保护资源：

AWS Management Console

1. 通过以下网址打开 AWS CloudFormation 控制台：https://console.amazonaws.cn/cloudformation/。
2. 选择要更新的堆栈，然后单击 Update Stack (更新堆栈)。



3. 在 Update Stack (更新堆栈) 向导的 Policy (策略) 屏幕上，选择定义覆盖堆栈策略的文件或输入一个文件。

   Note

   您必须拥有 AWS CloudFormation `SetStackPolicy` 操作的权限才能更新受保护的资源。

   覆盖策略应该为您要更新的受保护资源指定 `Allow`。覆盖策略是临时策略，只在此更新期间适用。

   例如，如果需要更新所有受保护资源，可以指定临时覆盖以允许所有更新：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

AWS CLI

- 使用 `aws cloudformation update-stack` 命令，使用 `--stack-policy-during-update-body` 或 `--stack-policy-during-update-url` 选项。

删除堆栈策略

在设置堆栈策略后，您不能移除或删除策略。如果要删除所有保护，必须更新策略，明确允许对所有资源执行所有操作，因为堆栈策略默认拒绝所有更新。下面的示例策略允许对所有资源进行所有更新：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

要设置更新堆栈策略，请参阅设置堆栈策略 (p. 70)。

堆栈策略参考

堆栈策略是定义用户可以执行的更新操作以及可操作的资源的 JSON 文档。这些权限是在以下元素中定义的：Effect、Action、Resource 和 Condition。默认情况下，没有资源受到保护。也就是说，允许对所有资源执行所有操作。下面的伪代码显示了堆栈策略的语法：

```
{
  "Statement" : [
    {
      "Effect" : "Deny_or_Allow",
      "Action" : "update_actions",
      "Principal" : "*",
      "Resource" : "LogicalResourceId/resource_logical_ID",
      "Condition" : {
        "StringEquals_or_StringLike" : {
          "ResourceType" : [resource_type, ...]
        }
      }
    }
  ]
}
```

效果
: 确定是拒绝还是允许对指定资源执行指定的操作。对此元素只能指定 Deny 或 Allow，如下面的代码段所示：

```
"Effect" : "Deny"
```

Important

如果堆栈策略包含任何重叠语句，Deny 始终覆盖 Allow。如果需要确保某一资源受到保护，请对该资源使用 Deny 语句。

操作
: 指定拒绝或允许的更新操作。可以指定以下操作：

`Update:Modify`
: 指定在对资源应用更改期间不会中断或有某些中断的更新操作。所有资源都保持其物理 ID。

`Update:Replace`
: 指定重新创建资源的更新操作。AWS CloudFormation 使用指定的更新创建新资源，然后删除旧资源。因为资源是重新创建的，所以资源的物理 ID 可能并不相同。

`Update:Delete`
指定删除资源的更新操作。所有从堆栈模板中完全删除资源的更新都需要此操作。

`Update:*`
指定所有更新操作。星号是通配符，代表所有更新操作。

下面的代码段显示如何只指定替换和删除操作：

```
"Action" : ["Update:Replace", "Update:Delete"]
```

您还可对操作使用 `Not`。例如，如果需要允许除 `Update:Delete` 外的所有更新操作，可以使用 `NotAction`，如下例所示：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "NotAction" : "Update:Delete",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

有关堆栈更新的更多信息，请参阅AWS CloudFormation 堆栈更新 (p. 63)。

委托人
`Principal` 元素是必需的，但它只支持通配符 (*)。

资源
指定将应用策略的资源的逻辑 ID。如果要指定资源类型，请使用 `Condition` 元素。

您可以使用逻辑 ID 指定一个资源，如下面的代码段所示：

```
"Resource" : ["LogicalResourceId/myEC2instance"]
```

您也可以对逻辑 ID 使用通配符。例如，如果使用所有相关资源的逻辑 ID 作为前缀，可以使用通配符指定全部，如下面的代码段所示：

```
"Resource" : ["LogicalResourceId/MyPrefix*"]
```

您还可以对资源使用 `Not`。例如，如果要允许更新某个资源之外的所有资源，可以使用 `NotResource`，如下例所示：

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

设置堆栈策略时，默认情况下会拒绝未显式允许的任何更新。通过允许更新 ProductionDatabase 资源之外的所有资源，会拒绝更新 ProductionDatabase 资源。

条件

指定应用策略的资源类型。如果要指定特定资源，请使用 Resource 元素。

您可以指定资源类型（如所有 Amazon EC2 实例和 Amazon RDS 数据库实例），如下例所示：

```
{
  "Statement" : [
  {
    "Effect" : "Deny",
    "Principal" : "*",
    "Action" : "Update:*",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "ResourceType" : ["AWS::EC2::Instance", "AWS::RDS::DBInstance"]
      }
    }
  },
  {
    "Effect" : "Allow",
    "Principal" : "*",
    "Action" : "Update:*",
    "Resource" : "*"
  }
  ]
}
```

设置堆栈策略时，默认情况下会拒绝未显式允许的任何更新。Allow 语句授予对 Amazon EC2 实例和 Amazon RDS 数据库实例之外的所有资源的更新权限。Deny 语句始终覆盖任何允许。

您还可以对资源类型使用通配符。例如，可以使用通配符拒绝对所有 Amazon EC2 资源（如实例、安全组和子网）的更新权限，如以下代码段所示：

```
"Condition" : {
  "StringLike" : {
    "ResourceType" : ["AWS::EC2::*"]
  }
}
```

使用通配符时，必须使用 StringLike 条件。

堆栈策略示例

防止任何对所有堆栈资源的更新

为了防止更新所有堆栈资源，下面的策略为所有资源的所有更新操作指定 Deny：

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
```

```
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

只防止更新数据库

下面的策略针对具有 MyDatabase 逻辑 ID 的数据库拒绝所有更新操作。为允许更新所有其他堆栈资源，该策略还允许对所有资源执行所有更新操作。Allow 语句不会影响 MyDatabase 资源，因为 Deny 语句始终覆盖任意允许。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyDatabase"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

实现相同结果的另一种方法是使用默认拒绝。设置堆栈策略时，默认情况下会拒绝未显式允许的任何更新。以下示例使用 NotResource 允许更新所有资源（ProductionDatabase 资源除外）。

```
{
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "NotResource" : "LogicalResourceId/ProductionDatabase"
    }
  ]
}
```

通过允许更新 ProductionDatabase 资源之外的所有资源，默认情况下会拒绝更新 ProductionDatabase 资源。但是，因为显式拒绝将覆盖任何允许，所以可以使用 Deny 语句确保保护资源。

防止任何对所有 Amazon RDS 数据库实例的更新

下面的策略针对 Amazon RDS 数据库实例资源类型拒绝所有更新操作。为了允许更新所有其他堆栈资源，该策略指定允许对所有资源执行所有更新操作。Allow 语句不会影响 Amazon RDS 数据库实例资源，因为 Deny 语句始终覆盖任意允许。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::RDS::DBInstance"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

防止对实例的替换更新

下面的策略拒绝会对具有 MyInstance 逻辑 ID 的实例造成替换的更新。为允许更新所有其他堆栈资源，该策略还允许对所有资源执行所有更新操作。不过，与往常一样，Allow 语句不会影响 MyInstance 资源，因为 Deny 语句始终覆盖任意允许。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
      "Action" : "Update:Replace",
      "Principal": "*",
      "Resource" : "LogicalResourceId/MyInstance"
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

防止更新所有嵌套堆栈

下面的策略针对 AWS CloudFormation 堆栈资源类型（嵌套堆栈）拒绝所有更新操作。为允许更新所有其他堆栈资源，该策略还允许对所有资源执行所有更新操作。不过，与往常一样，Allow 语句不会影响 AWS CloudFormation 堆栈资源，因为 Deny 语句始终覆盖任意允许。

```
{
  "Statement" : [
    {
      "Effect" : "Deny",
```

```
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "ResourceType" : ["AWS::CloudFormation::Stack"]
        }
      }
    },
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    }
  ]
}
```

使用 AWS CloudFormation 控制台

使用 AWS CloudFormation 控制台，您可以直接通过 Web 浏览器创建、监控、更新和删除堆栈。本部分包含使用 AWS CloudFormation 控制台执行常规操作的指南。

在本章节中

登录 AWS CloudFormation 控制台

使用 AWS CloudFormation 控制台可以通过基于 Web 的界面创建、监控、更新和删除 AWS CloudFormation 堆栈。它是 AWS 管理控制台的一部分。

您可以通过以下多种方式访问 AWS CloudFormation 控制台：

- 直接使用以下 URL 打开 AWS CloudFormation 控制台：`https://console.amazonaws.cn/cloudformation/`。如果您还没有登录 AWS 管理控制台，则需要先登录，然后才能使用 AWS CloudFormation 控制台。
- 如果您已登录且正在使用 AWS 管理控制台，则可以通过打开 Services (服务) 菜单并在以下任一子菜单中选择 CloudFormation 来访问 AWS CloudFormation 控制台：
  - 部署和管理

- 所有服务

如果您未在运行任何 AWS CloudFormation 堆栈，则会显示 Create a stack (创建堆栈) 选项。否则，您将看到当前运行的堆栈的列表。

另请参阅

- 创建堆栈 (p. 80)

在 AWS CloudFormation 控制台上创建堆栈

Abstract

使用向导在 AWS CloudFormation 控制台上创建堆栈。

在 AWS CloudFormation 控制台上创建堆栈是一个简单的向导驱动过程，其步骤如下：

1. 启动创建堆栈向导 (p. 81)
2. 选择堆栈模板 (p. 81)

3. 指定堆栈参数 (p. 82)
4. 设置堆栈选项 (p. 83)
5. 查看您的堆栈 (p. 83)

创建堆栈后，您可以监控堆栈的状态、查看堆栈的资源和输出、更新以及删除堆栈。有关这些操作的信息已在相关主题中提供。

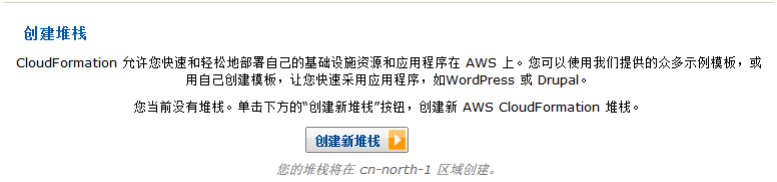
启动创建堆栈向导

在 AWS CloudFormation 控制台上创建堆栈

1. 登录 AWS 管理控制台，然后在 Services (服务) 菜单中选择 CloudFormation。
2. 使用以下其中一个选项创建堆栈：

   - 单击 Create Stack (创建堆栈)。如果您当前有正在运行的堆栈，这将是*唯一*选项。

   - 单击 CloudFormation Stacks (CloudFormation 堆栈) 主窗口中的 Create New Stack (创建新堆栈)。仅当您没有运行堆栈时，才可以看到此选项。



   - 单击 CloudFormation Stacks (CloudFormation 堆栈) 主窗口中的 Launch CloudFormer (启动 CloudFormer)，从当前正在运行的资源创建堆栈。仅当您没有运行堆栈时，才可以看到此选项。

     有关使用 CloudFormer 创建 AWS CloudFormation 堆栈的更多信息，请参阅使用 CloudFormer 创建模板 (p. 52)。

下一步，您将选择堆栈模板 (p. 81)。

在 AWS CloudFormation 控制台上选择堆栈模板

Abstract

在 AWS CloudFormation 控制台上选择堆栈名称和选择堆栈模板。

启动创建堆栈向导 (p. 81)之后，可指定堆栈名称并选择 AWS CloudFormation 用于创建堆栈的模板。

AWS CloudFormation 模板是指定构成堆栈的 AWS 资源的 JSON 文件。有关 AWS CloudFormation 模板的更多信息，请参阅模板 (p. 3)。

选择堆栈名称，选定堆栈模板：

1. 在 Create Stack (创建堆栈) 向导的 Create A New Stack (创建新堆栈) 页面上，在 Name (名称) 框中键入一个堆栈名称。

   堆栈名称必须仅包含字母数字字符且必须以字母字符为开头。名称的长度上限是 255 个字符。堆栈名称区分大小写。
2. 通过以下选项选择一个堆栈：

   使用示例模板
   从菜单上提供的模板中选择 AWS CloudFormation 模板。菜单中的可用模板列表通常与 AWS CloudFormation 示例模板网页上的模板列表相同。

   您可以从列表中选择 CloudFormer，以利用 CloudFormer 工具使用现有 AWS 资源创建堆栈。有关更多信息，请参阅 使用 CloudFormer 创建模板 (p. 52)。

   上传模板文件
   在本地系统中选择 AWS CloudFormation 模板。指定完整路径或单击 Browse (浏览) 以选择要上传的文件。

   上传的模板最多可以为 51 200 个字节。

   提供模板 URL
   指定 Amazon S3 存储桶中的模板的 URL。

   该 URL 必须指向您已对其拥有读取权限、与堆栈位于同一区域的 Amazon S3 存储桶中的模板（大小上限：460 800 字节）。该 URL 自身的长度在最大可以是 1 024 个字符。

3. 单击 Next Step (下一步) 以接受设置，然后继续指定堆栈参数 (p. 82)。

在 AWS CloudFormation 控制台上指定堆栈参数

Abstract

在 AWS CloudFormation 控制台上指定模板中定义的堆栈参数。

选择堆栈模板 (p. 81)后，可指定模板中定义的参数 (p. 101)。

您可以使用参数在创建时自定义堆栈。此处输入的数据可以在堆栈模板中通过逻辑 ID 进行引用，并且可用于修改 AWS 或自定义资源的配置方式。有关如何在 AWS CloudFormation 模板中指定参数的更多信息，请参阅参数声明 (p. 101)。

输入您堆栈的参数值

1. 在 Create Stack (创建堆栈) 向导的 Specify Parameters (指定参数) 页面上，指定堆栈模板中定义的参数。对于某些参数，默认值可能已存在。

   Note

   您可能需要在 AWS CloudFormation 创建堆栈之前确认一些资源，如 IAM 资源。有关更多信息，请参阅使用 AWS Identity and Access Management 控制访问 (p. 59)中的“AWS CloudFormation 模板中的 IAM 资源”。

2. 如果您对参数值满意，请单击 Next Step (下一步)，以继续为堆栈设置选项 (p. 83)。

设置 AWS CloudFormation 堆栈选项

Abstract

为 AWS CloudFormation 堆栈设置选项（如标签、堆栈事件通知或堆栈策略）。

指定在模板中定义的参数 (p. 101)之后，您可以为堆栈设置其他选项。

您可以设置以下堆栈选项：

标签
: 标签是任意键/值对，可用于针对成本分配等目的来标识堆栈。有关什么是标签以及如何使用标签的更多信息，请参阅 *Amazon EC2 用户指南*中的标记您的资源。

  Key (键) 可由任何字母数字字符组成，但是不得包含空格。标签键最长为 127 个字符。Value (值) 可由任何字母数字字符或空格组成。标签值最长可达 255 个字符。

Notification Options (通知选项)
: 发送有关堆栈事件的通知的新的或现有 Amazon Simple Notification Service 主题。

  如果您创建 Amazon SNS 主题，则必须指定名称和电子邮件地址（向该位置发送堆栈事件通知）。

Timeout
: 堆栈创建超时之前的分钟数。时间过期前，如果堆栈未能创建成功，那么创建将因超时而失败，堆栈将发生回滚。默认情况下，堆栈创建从不超时。

Rollback on failure (失败时回滚)
: 指定在堆栈创建失败时是否应回滚堆栈。通常情况下，您会接受默认值 Yes (是)。如果您希望即使创建失败也保留堆栈状态（例如，当您正在调试堆栈模板时），则选择 No (否)。

Stack policy (堆栈策略)
: 定义在堆栈更新期间要防止意外更新的资源。默认情况下，堆栈更新期间所有资源都可更新。有关更多信息，请参阅 防止更新堆栈资源 (p. 69)。

设置堆栈选项

1. 在 Create Stack (创建堆栈) 向导的 Options (选项) 屏幕上，您可以通过展开 Advanced (高级) 部分来指定标签或设置其他选项。
2. 输入了所有堆栈选项之后，请单击 Next (下一步) 以继续检查堆栈 (p. 83)。

审核您的堆栈并对 AWS CloudFormation 控制台上的堆栈成本进行评估

Abstract

在 AWS CloudFormation 控制台上审核您的堆栈并评估堆栈成本。

启动堆栈之前的最后一步是审核创建堆栈期间输入的值。还可以评估堆栈成本。

1. 在 Review (审核) 页面上，审核堆栈详细信息。

   如果需要在启动堆栈之前更改任何值，请单击 Back (返回) 以返回包含要更改的设置的页面。
2. （可选）您可以单击 Cost (成本) 链接，以评估堆栈的成本。AWS 简单月度成本结算器显示来自堆栈模板和启动设置的值。
3. 审核堆栈启动设置和堆栈评估成本之后，单击 Create (创建)，以启动您的堆栈。

   您的堆栈会显示在 AWS CloudFormation 堆栈列表中，其状态设置为 CREATE_IN_PROGRESS。

正在创建堆栈时（或创建之后），您可以使用堆栈详细信息窗格查看堆栈的事件、数据或资源 (p. 85)。AWS CloudFormation 会每分钟自动刷新堆栈事件。通过查看堆栈创建事件，您可以了解导致堆栈创建（或失败，如果您正在调试堆栈）的事件的顺序。

堆栈成功创建之后，其状态更改为 CREATE_COMPLETE。您随后可以选择它（如果需要），然后单击 Outputs (输出) 选项卡以查看堆栈的输出（如果在模板中定义了任何输出）。

创建 EC2 密钥对

使用某些 AWS CloudFormation 资源和模板会要求您针对验证指定 Amazon EC2 密钥对，例如当您配置对您的实例的 SSH 访问权限时。

可使用 AWS 管理控制台创建 Amazon EC2 密钥对，其过程如下：

要创建 EC2 密钥对

1. 在 AWS 管理控制台中，通过单击屏幕左上角的 Services (服务) 按钮并选择 EC2，从 AWS CloudFormation 控制台切换到 Amazon EC2 控制台。

   现在，控制台显示器显示了 Amazon EC2 控制台仪表板。
2. 在 Amazon EC2 控制台的 Navigation (导航) 窗格中，单击 Key Pairs (密钥对)。

   此时将显示 Key Pairs (密钥对) 页，其中显示了您的 Amazon EC2 密钥对。如果您尚未创建密钥对，则该列表为空，并改为显示 Create Key Pair (创建密钥对) 按钮。
3. 单击 Create Key Pair (创建密钥对) 按钮。
4. 键入密钥对名称，然后单击 Create (创建)。如何命名并不重要，重要的是要方便您记忆。

   创建密钥对后，即可开始下载私有密钥。它将称为 *name*.pem，其中 *name* 表示您为密钥对指定的名称。
5. 下载密钥对，并将权限设为 400（在 Linux 或 Mac 操作系统上）。

估算 AWS CloudFormation 堆栈的成本

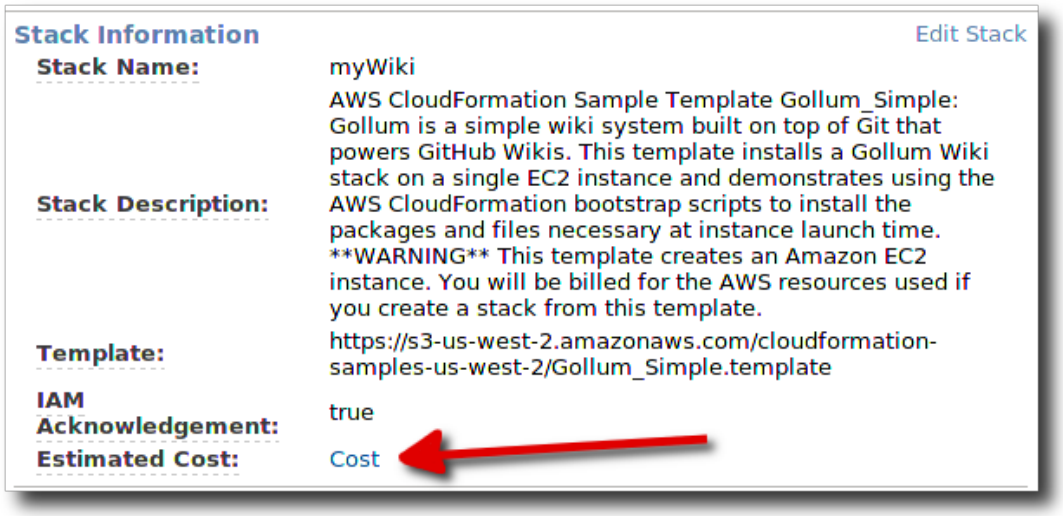
Abstract

估算使用 AWS CloudFormation 堆栈创建 AWS 资源的成本。

使用 AWS CloudFormation 无需额外付费。您为使用 AWS CloudFormation 创建的 AWS 资源（例如，Amazon EC2 实例、Elastic Load Balancing 负载均衡器等）支付的费用与手动创建它们的费用相同。

要估算堆栈的成本

1. 在 Create Stack (创建堆栈) 或 Update Stack (更新堆栈) 对话框的 Review (审核) 页上，单击 Cost (成本) 链接。



此链接将在新的浏览器页或选项卡上（具体取决于浏览器的设置）打开 AWS Simple Monthly Calculator (AWS 简单月度成本结算器)。

Note

由于您是从 AWS CloudFormation 控制台启动结算器的，因此其中已预填充模板配置和参数值。如果您可以预计您的 Amazon EC2 实例传输的数据量，则还有许多可配置的值可帮助您得到更准确的估算结果。

2. 单击 Estimate of your Monthly Bill (估算每月的账单) 选项卡以获取运行堆栈的月度估算值，以及分类显示的影响此估算值的所有因素。

使用 AWS 管理控制台查看 AWS CloudFormation 堆栈数据和资源

Abstract

使用 AWS 管理控制台查看 AWS CloudFormation 堆栈数据和资源。

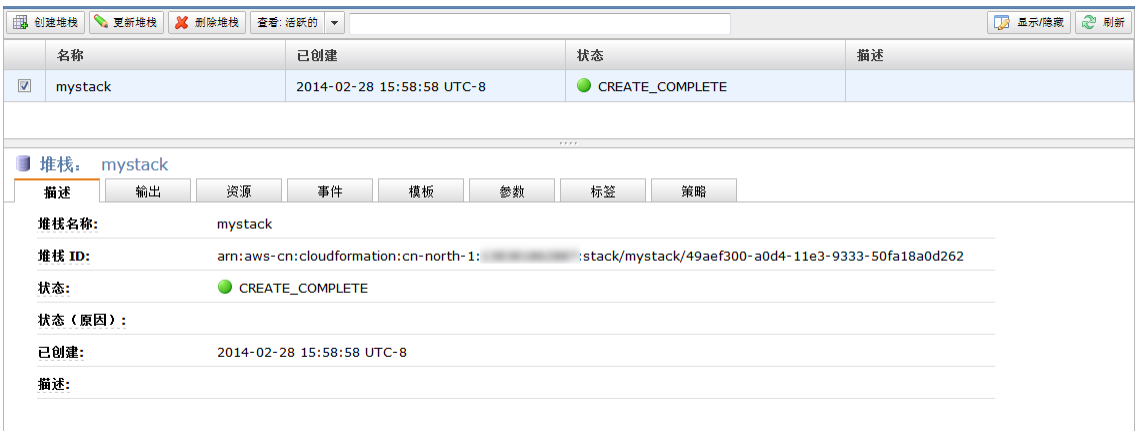
创建 AWS CloudFormation 堆栈之后，您可以使用 AWS 管理控制台来查看其数据和资源。您可查看以下堆栈信息：

- 输出
- 资源
- 事件
- 模板
- 参数
- 标签
- 策略

查看 AWS CloudFormation 堆栈的输出

1. 在 AWS CloudFormation 控制台中选择您的堆栈。这会在堆栈详细信息窗格中显示信息。
2. 在详细信息窗格中，单击选项卡可查看堆栈的相关信息。

   例如，单击 Outputs (输出) 可查看与堆栈关联的输出。



在 AWS CloudFormation 控制台上更新堆栈

Abstract

使用 AWS CloudFormation 控制台更新堆栈以修改或启动堆栈更新。

您可以使用 AWS CloudFormation 控制台来更新堆栈，步骤如下：

1. 修改堆栈和启动堆栈更新 (p. 86)
2. 选择堆栈模板更新堆栈 (p. 87)
3. 指定堆栈参数和更新策略 (p. 88)
4. *可选*：取消堆栈更新 (p. 88)

修改堆栈和启动堆栈更新

在您可以更新堆栈之前，必须首先修改其堆栈模板。为此，请执行以下操作之一：

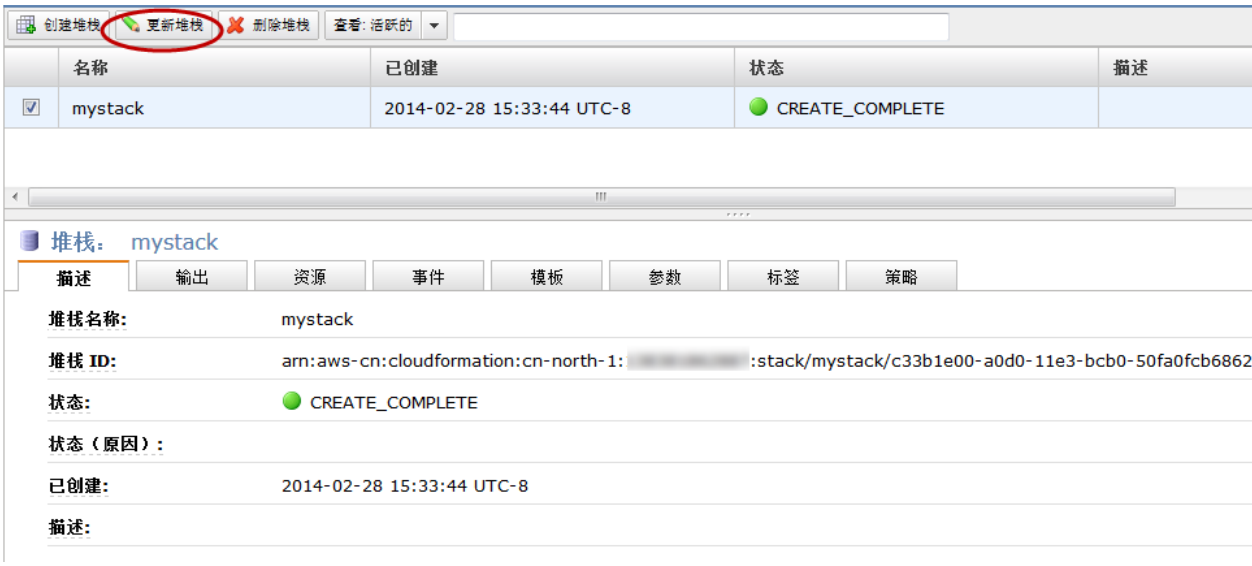
- 修改您用于创建堆栈的模板。
- 从堆栈信息窗格的 Template (模板) 选项卡中复制堆栈模板。要查看堆栈模板，请参阅查看堆栈数据和资源 (p. 85)。

启动更新堆栈向导

在您完成模板更改之后，便已准备就绪，可以启动更新堆栈向导。

使用更新堆栈向导启动堆栈更新

1. 在 AWS CloudFormation 控制台上的堆栈列表中，选择您要更新的正在运行的堆栈。
2. 单击 Update Stack (更新堆栈)。



下一步，您将选择修改后的堆栈模板 (p. 87)。

在 AWS CloudFormation 控制台上选择用于更新堆栈的堆栈模板

Abstract

在 AWS CloudFormation 控制台上为更新堆栈选择堆栈模板。

启动更新堆栈向导 (p. 86)之后，AWS CloudFormation 会提示您指定模板。

选择堆栈模板更新堆栈

1. 在 Update Stack (更新堆栈) 向导的 Update Stack (更新堆栈) 页面上，使用以下选项之一选择堆栈模板：

   使用示例模板
   : 使用标准示例模板更新堆栈。如果您在修改一个示例模板，并且要恢复为原始设置，则您可以使用此选项。如果示例模板之一已更新，并且您要利用已更新的模板配置，则您也可以使用此选项。

   上传模板文件
   : 指定路径或单击 Browse (浏览) 以选择已更新的模板。

   提供模板 URL
   : 提供 Amazon S3 存储桶中的模板的 URL。如果您的已更新的堆栈模板位于 Amazon S3 存储桶中，请使用此选项。

2. 单击 Continue (继续)，进入指定参数 (p. 88)。

另请参阅

- 修改堆栈和启动堆栈更新 (p. 86)

使用 AWS CloudFormation 控制台指定堆栈参数和更新策略

Abstract

在 AWS CloudFormation 控制台上指定更新堆栈时的堆栈参数并审核堆栈更新。

更新堆栈时，指定堆栈参数

选择模板 (p. 87)以用于更新之后，AWS CloudFormation 会提示您指定堆栈参数。

1. 在 Update Stack (更新堆栈) 向导的 Specify Parameters (指定参数) 页上，输入已更新的堆栈模板中定义的任何参数值。

   如果您进行过添加或删除参数的操作，那么此处列出的参数将与堆栈创建期间指定的参数不同。对于现有堆栈参数，创建堆栈时输入的值或者最近更新期间指定的值将显示为默认值。
2. 单击 Next Step (下一步) 以审核堆栈更新。
3. 如果所更新的堆栈具有关联的堆栈策略，则您可以通过指定覆盖堆栈策略来更新受保护的资源。然后单击 Next Step (下一步)。

   有关更多信息，请参阅 防止更新堆栈资源 (p. 69)。
4. 审核堆栈更新的设置，然后单击 Update (更新) 以开始更新堆栈。

   您的堆栈将进入 UPDATE_IN_PROGRESS 状态。完成更新后，堆栈状态将设置为 UPDATE_COMPLETE。

   如果堆栈更新失败，则它会自动回滚，堆栈将设置为 UPDATE_ROLLBACK_COMPLETE。

   Note

   堆栈开始更新后，如果它仍处于 UPDATE_IN_PROGRESS 状态，那么您可以取消更新。有关更多信息，请参阅 取消堆栈更新 (p. 88)。

在 AWS CloudFormation 控制台上取消堆栈更新

Abstract

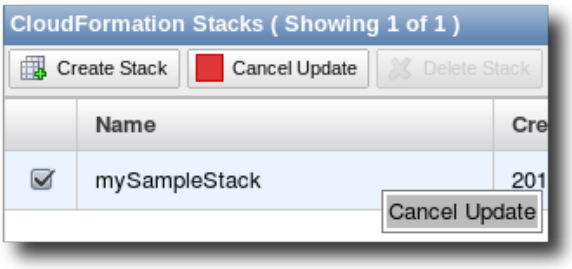
在 AWS CloudFormation 控制台上取消堆栈更新（只要更新仍在进行）。

堆栈更新开始后，如果堆栈仍处于以下状态，则您可以取消堆栈更新：UPDATE_IN_PROGRESS 状态。更新完成之后，将无法取消它。但是，您可以通过任何先前的设置再次更新堆栈。

如果您取消堆栈更新，那么堆栈将回滚到启动堆栈更新之前已存在的堆栈配置。

取消堆栈更新

1. 在 AWS CloudFormation 控制台上的堆栈列表中，选择当前正在更新的堆栈（其状态必须为 UPDATE_IN_PROGRESS）.
2. 单击 Cancel Update (取消更新)。



3. 要继续取消更新，请在出现提示时单击 Yes, Cancel Update (是，取消更新)。否则，单击 Cancel (取消) 以恢复更新。

堆栈将进入 UPDATE_ROLLBACK_IN_PROGRESS 状态。更新取消完成后，堆栈状态将为 UPDATE_ROLLBACK_COMPLETE。

另请参阅

- 更新堆栈 (p. 86)

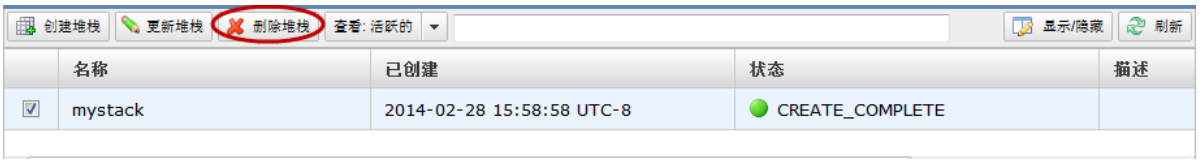
在 AWS CloudFormation 控制台上删除堆栈

Abstract

在堆栈当前正在运行的情况下，在 AWS CloudFormation 控制台上删除堆栈。

要删除堆栈

1. 在 AWS CloudFormation 控制台上的堆栈列表中，选择要删除的堆栈（该堆栈当前必须正在运行）。
2. 单击 Delete Stack (删除堆栈)。



3. 在出现提示时单击 Yes, Delete (是，删除)。

Note

堆栈删除过程开始之后，便无法中止。堆栈将进入 DELETE_IN_PROGRESS 状态。

堆栈删除过程完成之后，堆栈将处于 DELETE_COMPLETE 状态。默认情况下，处于 DELETE_COMPLETE 状态的堆栈不会显示在 AWS CloudFormation 控制台中。要显示已删除的堆栈，您必须按照查看已删除堆栈 (p. 90)中所述更改堆栈查看设置。

在 AWS CloudFormation 控制台上查看已删除的堆栈

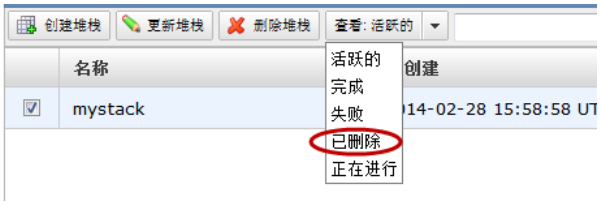
Abstract

通过首先更改堆栈视图设置使已删除的堆栈可见，在 AWS CloudFormation 控制台上查看已删除的堆栈。

默认情况下，AWS CloudFormation 控制台不显示处于 DELETE_COMPLETE 状态的堆栈。要显示有关已删除堆栈的信息，您必须更改堆栈视图。

要查看已删除的堆栈

- 在 AWS CloudFormation 控制台中，从 Filter (筛选条件) 列表中选择 Deleted (已删除)。



AWS CloudFormation 会列出所有已删除的堆栈（具有 DELETE_COMPLETE 状态的堆栈）。

另请参阅

- 删除堆栈 (p. 89)
- 查看堆栈数据和资源 (p. 85)

相关主题

- 使用 AWS CLI (p. 91)

使用 AWS 命令行界面

Abstract

使用 AWS 命令行界面在您的系统终端上创建、监控、更新和删除堆栈。

使用 AWS 命令行界面 (CLI)，可以在您的系统终端上创建、监控、更新和删除堆栈。您还可以使用 AWS CLI 通过脚本自动执行操作。有关 AWS CLI 的更多信息，请参阅 AWS Command Line Interface 用户指南。

如果您使用 Windows PowerShell，则 AWS 还提供适用于 Windows PowerShell 的 AWS 工具。

Topics

说明并列出堆栈

可以使用两个 AWS CLI 命令获取 AWS CloudFormation 堆栈的相关信息：`aws cloudformation list-stacks` 和 `aws cloudformation describe-stacks`.

aws cloudformation list-stacks

使用 `aws cloudformation list-stacks` 命令可以获取您已创建的任何堆栈的列表（甚至包括已在 90 天内删除的堆栈）。您可以使用选项按堆栈状态（如 `CREATE_COMPLETE` 和 `DELETE_COMPLETE`）筛选结果。`aws cloudformation list-stacks` 命令可返回有关任何正在运行或已删除的堆栈的汇总信息，包括名称、堆栈标识符、模板和状态。

Note

aws cloudformation list-stacks 命令可在堆栈已删除后的 90 天内返回已删除堆栈的相关信息。

以下示例显示具有 CREATE_COMPLETE 状态的所有堆栈的汇总信息：

```
PROMPT> aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
[
    {
        "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/
644df8e0-0dff-11e3-8e2f-5088487c4896",
        "TemplateDescription": "AWS CloudFormation Sample Template S3_Bucket:
Sample template showing how to create a publicly accessible S3 bucket. **WARN
ING** This template creates an
S3 bucket. You will be billed for the AWS resources used if you create a stack
 from this template.",
        "StackStatusReason": null,
        "CreationTime": "2013-08-26T03:27:10.190Z",
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE"
    }
]
```

aws cloudformation describe-stacks

`aws cloudformation describe-stacks` 命令提供正在运行的堆栈的相关信息。您可以使用选项按堆栈名称筛选结果。此命令将返回有关堆栈的信息，包括名称、堆栈标识符和状态。

以下示例显示 `myteststack` 堆栈的汇总信息：

```
PROMPT> aws cloudformation describe-stacks --stack-name myteststack
{
    "Stacks":  [
        {
            "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/a69442d0-0b8f-11e3-8b8a-500150b352e0",
          "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING**
This template creates an S3 bucket.
You will be billed for the AWS resources used if you create a stack from this
template.",
            "Tags": [],
            "Outputs": [
                {
                   "Description": "Name of S3 bucket to hold website content",

                    "OutputKey": "BucketName",
                    "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
                }
            ],
            "StackStatusReason": null,
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false
        }
    ]
}
```

如果您未使用 *--stack-name* 选项将输出限制为一个堆栈，则将返回所有正在运行的堆栈的相关信息。

堆栈状态代码

您可以指定一个或多个堆栈状态代码，只列出具有指定状态代码的堆栈。下表说明了各个堆栈状态代码：

堆栈状态	说明
CREATE_COMPLETE	成功创建一个或多个堆栈。
CREATE_IN_PROGRESS	正在创建一个或多个堆栈。
CREATE_FAILED	一个或多个堆栈创建失败。查看堆栈事件可了解所有相关错误消息。创建失败的可能原因包括：没有足够的权限使用堆栈中的所有资源，参数值被 AWS 服务拒绝，或者在资源创建期间超时。
DELETE_COMPLETE	成功删除一个或多个堆栈。已删除的堆栈在 90 天内保留可供查看。
DELETE_FAILED	一个或多个堆栈删除失败。由于删除失败，您可能有一些资源仍在运行，但是您不使用或更新堆栈。再次删除堆栈或查看堆栈事件可了解所有相关错误消息。
DELETE_IN_PROGRESS	正在删除一个或多个堆栈。
ROLLBACK_COMPLETE	在堆栈创建失败或明确取消堆栈创建后成功删除一个或多个堆栈。在创建堆栈操作期间创建的所有资源都被删除。
ROLLBACK_FAILED	在堆栈创建失败或明确取消堆栈创建后删除一个或多个堆栈失败。删除堆栈或查看堆栈事件了解所有相关错误消息。
ROLLBACK_IN_PROGRESS	在堆栈创建失败或明确取消堆栈创建后正在删除一个或多个堆栈。
UPDATE_COMPLETE	成功更新一个或多个堆栈。
UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	成功更新一个或多个堆栈后正在删除堆栈的旧资源。对于需要替换资源的堆栈更新，AWS CloudFormation 首先创建新资源，然后删除旧资源，帮助减少堆栈中断。在这种状态下，堆栈已更新可用，但 AWS CloudFormation 仍然会删除旧资源。
UPDATE_IN_PROGRESS	正在更新一个或多个堆栈。
UPDATE_ROLLBACK_COMPLETE	在堆栈更新失败后将一个或多个堆栈成功返回之前的工作状态。
UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	堆栈更新失败后正在删除一个或多个堆栈的新资源。在这种状态下，堆栈已回滚到之前的工作状态并且可用，但 AWS CloudFormation 仍会删除它在堆栈更新期间创建的所有新资源。
UPDATE_ROLLBACK_FAILED	在堆栈更新失败后将一个或多个堆栈返回之前的工作状态失败。您可以删除堆栈，或者联系客户支持将堆栈恢复到可用状态。
UPDATE_ROLLBACK_IN_PROGRESS	堆栈更新失败后正在将一个或多个堆栈返回之前的工作状态。

查看堆栈事件历史记录

您可以使用 `aws cloudformation describe-stack-events` 命令跟踪 AWS CloudFormation 创建和删除的资源的状态。创建或删除堆栈所用的时间取决于堆栈的复杂度。

下面的示例中使用 `aws cloudformation create-stack` 命令，基于模板文件创建了堆栈。创建堆栈后，使用 aws cloudformation describe-stack-events 命令显示了在堆栈创建过程中报告的事件。

以下示例使用 `sampletemplate.json` 模板文件，创建了名为 `myteststack` 的堆栈：

```
PROMPT> aws cloudformation create-stack --stack-name myteststack --template-
body file:///home/local/test/sampletemplate.json
[
    {
        "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING**
This template creates an S3 bucket.
You will be billed for the AWS resources used if you create a stack from this
template.",
        "Tags": [],
        "Outputs": [
            {
                "Description": "Name of S3 bucket to hold website content",
                "OutputKey": "BucketName",
                "OutputValue": "myteststack-s3bucket-jssofi1zie2w"
            }
        ],
        "StackStatusReason": null,
        "CreationTime": "2013-08-23T01:02:15.422Z",
        "Capabilities": [],
        "StackName": "myteststack",
        "StackStatus": "CREATE_COMPLETE",
        "DisableRollback": false
    }
]
```

以下示例说明了 `myteststack` 堆栈：

```
PROMPT> aws cloudformation describe-stack-events --stack-name myteststack
{
    "StackEvents": [
        {
            "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "EventId": "af67ef60-0b8f-11e3-8b8a-500150b352e0",
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2013-08-23T01:02:30.070Z",
            "StackName": "myteststack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/a69442d0-0b8f-11e3-8b8a-500150b352e0",
            "LogicalResourceId": "myteststack"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "EventId": "S3Bucket-CREATE_COMPLETE-1377219748025",
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceType": "AWS::S3::Bucket",
            "Timestamp": "2013-08-23T01:02:28.025Z",
```

```
            "StackName": "myteststack",
            "ResourceProperties": "{\"AccessControl\":\"PublicRead\"}",
            "PhysicalResourceId": "myteststack-s3bucket-jssofi1zie2w",
            "LogicalResourceId": "S3Bucket"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "EventId": "S3Bucket-CREATE_IN_PROGRESS-1377219746688",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceType": "AWS::S3::Bucket",
            "Timestamp": "2013-08-23T01:02:26.688Z",
            "ResourceStatusReason": "Resource creation Initiated",
            "StackName": "myteststack",
            "ResourceProperties": "{\"AccessControl\":\"PublicRead\"}",
            "PhysicalResourceId": "myteststack-s3bucket-jssofi1zie2w",
            "LogicalResourceId": "S3Bucket"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "EventId": "S3Bucket-CREATE_IN_PROGRESS-1377219743862",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceType": "AWS::S3::Bucket",
            "Timestamp": "2013-08-23T01:02:23.862Z",
            "StackName": "myteststack",
            "ResourceProperties": "{\"AccessControl\":\"PublicRead\"}",
            "PhysicalResourceId": null,
            "LogicalResourceId": "S3Bucket"
        },
        {
            "StackId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896",
            "EventId": "a69469e0-0b8f-11e3-8b8a-500150b352e0",
            "ResourceStatus": "CREATE_IN_PROGRESS",
            "ResourceType": "AWS::CloudFormation::Stack",
            "Timestamp": "2013-08-23T01:02:15.422Z",
            "ResourceStatusReason": "User Initiated",
            "StackName": "myteststack",
            "PhysicalResourceId": "arn:aws:cloudformation:us-east-
1:123456789012:stack/myteststack/a69442d0-0b8f-11e3-8b8a-500150b352e0",
            "LogicalResourceId": "myteststack"
        }
    ]
}
```

Note

您可以在堆栈正在创建时运行 aws cloudformation describe-stack-events 命令，以在报告事件时查看这些事件。

最新发生的事件先报告。下表介绍 aws cloudformation describe-stack-events 命令返回的字段：

字段	说明
EventId	事件标识符

字段	说明
StackName	事件对应的堆栈名称
StackId	事件对应的堆栈标识符
LogicalResourceId	资源的逻辑标识符
PhysicalResourceId	资源的物理标识符
ResourceProperties	资源的属性
ResourceType	资源类型
Timestamp	事件发生的时间
ResourceStatus	资源状态；可以是以下值之一：*IN_PROGRESS* | *CREATE_FAILED* | *CREATE_COMPLETE* | *DELETE_IN_PROGRESS* | *DELETE_FAILED* | *DELETE_COMPLETE*。
ResourceStatusReason	有关状态的更多信息

列出资源

在运行 `aws cloudformation create-stack` 命令后，您可以立即使用 `aws cloudformation list-stack-resources` 命令列出其资源。该命令将列出您使用 *--stack-name* 参数指定的堆栈中的每个资源的汇总信息。该报表包含堆栈的汇总信息，包括创建或删除状态。

以下示例显示了 myteststack 堆栈的资源：

```
PROMPT> aws cloudformation list-stack-resources --stack-name myteststack
{
    "StackResourceSummaries": [
        {
            "ResourceStatus": "CREATE_COMPLETE",
            "ResourceType": "AWS::S3::Bucket",
            "ResourceStatusReason": null,
            "LastUpdatedTimestamp": "2013-08-23T01:02:28.025Z",
            "PhysicalResourceId": "myteststack-s3bucket-sample",
            "LogicalResourceId": "S3Bucket"
        }
    ]
}
```

AWS CloudFormation 将报告有关任何正在运行或已删除的堆栈的资源详细信息。如果您指定了状态为 *CREATE_IN_PROCESS* 的堆栈的名称，AWS CloudFormation 将仅报告状态为 *CREATE_COMPLETE* 的资源。

Note

aws cloudformation describe-stack-resources 命令可在堆栈已删除后的 90 天内返回已删除堆栈的相关信息。

检索模板

AWS CloudFormation 将用于创建堆栈的模板存储为堆栈的一部分。您可以使用 `aws cloudformation get-template` 命令，从 AWS CloudFormation 中检索模板。

Note

`aws cloudformation get-template` 命令可在堆栈已删除后的 90 天内返回已删除堆栈模板的相关信息。

以下示例显示了 `myteststack` 堆栈的模板：

```
PROMPT> aws cloudformation get-template --stack-name myteststack
{
    "TemplateBody": {
        "AWSTemplateFormatVersion": "2010-09-09",
        "Outputs": {
            "BucketName": {
                "Description": "Name of S3 bucket to hold website content",
                "Value": {
                    "Ref": "S3Bucket"
                }
            }
        },
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
template showing how to create a publicly accessible S3 bucket. **WARNING**
This template creates an S3 bucket.
You will be billed for the AWS resources used if you create a stack from this
template.",
        "Resources": {
            "S3Bucket": {
                "Type": "AWS::S3::Bucket",
                "Properties": {
                    "AccessControl": "PublicRead"
                }
            }
        }
    }
}
```

输出将包含整个模板正文，并括在引号中。

验证模板

如需检查您的模板文件是否存在语法错误，您可以使用 `aws cloudformation validate-template` 命令。

Note

`aws cloudformation validate-template` 的设计用途为仅检查您的模板的语法。它不能保证您已对一项资源指定的属性值对于该资源有效，也不能决定创建堆栈时存在的资源数量。

要检查操作有效性，您需要尝试创建堆栈。没有用于 AWS CloudFormation 堆栈的沙盒或测试区，因此您需要在测试期间为创建的资源支付费用。

您可以使用 *--template-body* 参数在本地验证模板，也可以使用 *--template-url* 参数进行远程验证。以下示例验证在远程位置的模板：

```
PROMPT> aws cloudformation validate-template --template-url https://s3.amazon
aws.com/cloudformation-templates-us-east-1/S3_Bucket.template
{
    "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample template
 showing how to create a publicly accessible S3 bucket. **WARNING** This template
 creates an S3 bucket.
You will be billed for the AWS resources used if you create a stack from this
template.",
    "Parameters": [],
    "Capabilities": []
}
```

预期结果是无错误消息，并且列出所有参数的相关信息。

以下示例显示了本地模板文件出现的错误：

```
PROMPT> aws cloudformation validate-template --template-body file:///home/loc
al/test/sampletemplate.json
{
    "ResponseMetadata": {
        "RequestId": "4ae33ec0-1988-11e3-818b-e15a6df955cd"
    },
    "Errors": [
        {
            "Message": "Template format error: JSON not well-formed. (line 11,
 column 8)",
            "Code": "ValidationError",
            "Type": "Sender"
        }
    ],
    "Capabilities": [],
    "Parameters": []
}
A client error (ValidationError) occurred: Template format error: JSON not well-
formed. (line 11, column 8)
```

相关主题

- 使用 AWS CloudFormation 控制台 (p. 79)

使用 AWS CloudFormation 模板

Topics

充分利用 AWS CloudFormation 的关键在于对模板的充分了解。

为了让您能够快速实现对模板的更改和编写操作，本部分将陈述模板的分解详情、示例模板和模板代码段。本部分还将探讨如何更改和验证模板。

- 在模板剖析 (p. 99)中，我们提供了对每个模板对象进行编码的技术详细信息。
- 在模板代码段 (p. 128)中，我们提供了一些模板部分，这些模板部分演示了如何为模板的特定部分编写 JSON 代码。在此部分中，您可以找到用于 Amazon EC2 实例、Amazon S3 域、AWS CloudFormation 映射等的初步代码段。可以选择上述代码段涵盖一系列您可能会常常算入您模板中的资源和属性。这些代码段按使用它们进行声明的资源进行分组，其中常规用途 AWS CloudFormation 代码段在 AWS CloudFormation 模板代码段 (p. 178)中。
- 示例模板 (p. 111)部分包含若干示例模板，使用这些模板只需进行少量修改或不需要进行修改即可创建堆栈。这些示例按其复杂程度分类，并在完整应用程序的上下文中重点强调 AWS CloudFormation 模板功能的使用。有些模板需要您在命令的 `--parameters` 选项中指定值。

有关您可以在模板中使用的受支持资源、类型名称、内部函数和虚拟参数的详细信息，请参阅模板参考 (p. 217)部分。

模板剖析

Abstract

描述需要用于 AWS CloudFormation 模板的组件的详细信息。

Topics

本部分将详细描述模板的各个部分。

Note

当您写好模板后，您可以使用 `aws cloudformation validate-template` 验证其句法正确性。有关更多信息，请参阅 验证模板 (p. 97)。

模板声明

Abstract

描述 AWS CloudFormation 模板的语法和六个主要部分。

模板有六个主要部分，每个部分之间用逗号隔开。*Resources* 部分是必填部分。其它为可选部分。模板中各部分没有必需顺序。

模板的第一个字符必须为左大括号 "{"。最后一个字符必须为右大括号 "}"。各个部分用逗号隔开。

以下模板分断显示的是模板结构和各部分。这些部分可能会以任何顺序显示在模板中。

```
{

    "AWSTemplateFormatVersion" : "version date",

    "Description" : "Valid JSON strings up to 4K",

    "Parameters" : {
        set of parameters
    },

    "Mappings" : {
            set of mappings
            },

    "Conditions" : {
            set of conditions
            },

    "Resources" : {
        set of resources
     },
```

```
    "Outputs" : {
        set of outputs
    }
}
```

每个部分的更多信息如下所述：

模板格式版本声明

Abstract

通过版本声明标识模板格式的功能。

模板格式版本为可选声明，可标识模板格式的功能。由于 AWS CloudFormation 模板格式尚未更改，因此目前仅支持一个值：2010-09-09.

模板格式版本声明的值必须是文字字符串。您不能将它的值建立在参数或函数的基础上。如果该值不存在，AWS CloudFormation 将采用最新的模板格式版本。以下是有效模板格式版本声明的示例：

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

Note

模板格式版本不同于 API 或 WSDL 版本，而且能独立于这两个版本进行更改。

模板描述声明

Abstract

在模板的 Description 部分中提供有关模板的任意注释。

模板描述部分为可选部分。如果有这部分，它*必须*紧随 AWSTemplateFormatVersion 部分之后。

模板说明部分可使您对您的模板进行任意评论。说明为文字 JSON 字符串，长度在 0 到 1024 个字节之间；说明值不能基于参数或函数。说明声明被编写为密钥/值对，如下所示：

```
"Description" : "Contains an autoscaling group and a load balancer.",
```

参数声明

Abstract

在堆栈创建时将值传入模板，以通过模板的 Parameters 部分自定义每个堆栈部署。

可选 Parameters 部分可使您在创建堆栈时将值传输到您的模板中。您可以通过 Parameters 创建可为每个堆栈部署自定义的模板。通过包含参数的模板创建堆栈时，可以为这些参数指定值。在模板内，您可以使用 "Ref" 函数指定资源属性值中的参数值。例如，您可以用以下 Parameters 部分定义字符串参数：

```
"Parameters" : {
    "URL" : {
```

```
      "Type" : "String"
    }
}
```

Note

如果有，Parameters 部分必须至少声明一个参数。一个 AWS CloudFormation 模板中最多可以有 60 个参数。

在运行时，您可以使用 `aws cloudformation create-stack` 的 *`--parameters`* 选项将 URL 参数设置为特定值：

```
aws cloudformation create-stack --stack-name MyStack --template-body
file:///mytemplate.json --parameters ParameterKey=URL,ParameterValue=127.0.0.1
```

多个参数分配以空格分隔。

Note

默认情况下，aws cloudformation describe-stacks 将返回参数值。为了防止返回密码等敏感参数值，请将设置为 TRUE 的 *`NoEcho`* 属性包含在 AWS CloudFormation 模板中。

参数可具有被称为约束条件的规则，用于决定参数的有效值。通过这些约束可以在创建任何资源之前验证用户的输入。例如，您可以设置一个约束，规定某参数字符串值只能包含字母数字字符，或者数字参数值必须介于 1 和 10 之间。

参数语法和属性

Parameters 部分由密钥名称 *`Parameters`*（后跟单个冒号）组成。所有参数声明都被括在括号里。在 Parameters 部分中声明的参数用逗号隔开。

每个参数必须声明一个用双引号括起的名称，后跟冒号。参数名称必须为字母数字，并且在模板的所有逻辑名称中具有唯一性。

可将参数声明为以下任一类型：*`String`*、*`Number`* 或 *`CommaDelimitedList`*。对于具有 *`String`* 或 *`Number`* 类型的参数，您可以定义 AWS CloudFormation 用于验证参数值的约束条件。

参数有以下属性：

属性	必需	说明
类型	是	String、Number 或 CommaDelimitedList。 类型为 *`String`* 的参数只是文字字符串。 类型为 *`Number`* 的参数可以是整数或浮点数。请注意，AWS CloudFormation 将参数验证为数字，但会将模板内的参数值用作字符串。 类型为 *`CommaDelimitedList`* 的参数是由逗号分隔的文字字符串数组。成员字符串进行了空间修剪，它比特定值中的逗号多一个字符串。
默认	否	模板适当类型的值，用于未在堆栈创建时指定值的情况下。如果参数的约束条件已定义，则此值必须符合此类约束条件。

属性	必需	说明
NoEcho	否	如果为 *TRUE*，则使用 `aws cloudformation describe-stacks` 时，参数值将使用星号 (*****) 进行遮蔽。
AllowedValues	否	包含参数允许值列表的阵列。
AllowedPattern	否	*String* 约束条件。对参数字符串值所允许的模式进行表示的正则表达式。
MaxLength	否	*String* 约束条件。决定参数字符串值中最大字符数的整数值。
MinLength	否	*String* 约束条件。决定参数字符串值中最小字符数的整数值。
MaxValue	否	*Number* 约束条件。决定参数最大允许数值的数值。
MinValue	否	*Number* 约束条件。决定参数最小允许数值的数值。
说明	否	用于说明参数的 String 类型（最多为 4000 个字符）。

ConstraintDescription	否	用于解释发生约束违例时显示的约束要求的 String 类型。例如，具有 AllowedPattern "[A-Za-z0-9]+" 的参数会在用户指定无效值时显示此错误消息：
		`Malformed input-Parameter MyParameter must match pattern [A-Za-z0-9]+`
		您可以给 ConstraintDescription 添加一个值 "must only contain upper– and lowercase letters, and numbers" 来显示自定义的错误消息：
		`Malformed input-Parameter MyParameter must only contain upper and lower case letters and numbers`

参数示例

以下示例 *Parameters* 部分声明了两个参数。DBPort 参数为 `Number` 类型，具有默认值 3306、最小值 1150 和最大值 65535。DBPwd 参数为 `String` 类型，没有默认值，*NoEcho* 设置为 True 以防止 aws cloudformation describe-stacks 返回参数值，最小长度为 1，最大长度为 41，其模式允许小写和大写字母字符和数字。

```
"Parameters" : {
    "DBPort": {
      "Default": "3306",
      "Description" : "TCP/IP port for the database",
      "Type": "Number",
      "MinValue": "1150",
      "MaxValue": "65535"
    },
    "DBPwd": {
      "NoEcho": "true",
      "Description" : "The database admin account password",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "41",
      "AllowedPattern" : "[a-zA-Z0-9]*"
```

```
    }
}
```

逗号分隔列表

要在单个参数中指定多个值，请使用 CommaDelimitedList 参数类型。例如，如果您使用三个不同子网自己的 CIDR 块创建这些子网，则可以使用三个不同参数指定三个不同 CIDR 块。不过更简单的是只需使用采用三个 CIDR 块的逗号分隔列表的单个参数，如以下代码段所示：

```
"Parameters" : {
  "DbSubnetIpBlocks": {
    "Description": "Comma-delimited list of three IP blocks",
    "Type": "CommaDelimitedList",
      "Default": "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
  }
}
```

要引用列表中的特定值，请在模板的 Resources 部分中使用 Fn::Select 内部函数。可传递所需对象的索引值和对象列表，如以下代码段所示：

```
"DbSubnet1": {
  "Type": "AWS::EC2::Subnet",
    "Properties": {
      "AvailabilityZone": {"Fn::Join" : ["",[ { "Ref": "AWS::Region" }, {
"Fn::Select" : [ "0", {"Ref": "VpcAzs"} ] } ] ]} ,
      "VpcId": { "Ref": "VPC" },
      "CidrBlock": { "Fn::Select" : [ "0", {"Ref": "DbSubnetIpBlocks"} ] }
    }
},
"DbSubnet2": {
  "Type": "AWS::EC2::Subnet",
    "Properties": {
      "AvailabilityZone": {"Fn::Join" : ["",[ { "Ref": "AWS::Region" }, {
"Fn::Select" : [ "1", {"Ref": "VpcAzs"} ] } ] ]} ,
      "VpcId": { "Ref": "VPC" },
      "CidrBlock": { "Fn::Select" : [ "1", {"Ref": "DbSubnetIpBlocks"} ] }
    }
},
"DbSubnet3": {
  "Type": "AWS::EC2::Subnet",
    "Properties": {
      "AvailabilityZone": {"Fn::Join" : ["",[ { "Ref": "AWS::Region" }, {
"Fn::Select" : [ "2", {"Ref": "VpcAzs"} ] } ] ]} ,
      "VpcId": { "Ref": "VPC" },
      "CidrBlock": { "Fn::Select" : [ "2", {"Ref": "DbSubnetIpBlocks"} ] }
    }
}
```

映射声明

Abstract

通过在模板的 Mappings 部分中声明的映射将键与一组对应的指定值匹配。

映射将密钥与对应的一组命名值相匹配。例如，如果您想根据区域设置值，您可以创建将区域名称用作密钥且其中含有您想为每个特定区域指定的值的映射。

映射在 *Mappings* 部分中声明，其中每个映射用逗号分隔。映射中的密钥和值必须为文字字符串。每个映射是格式为带双引号的密钥、单个冒号和包括有数组带映射值的括号。以下示例显示了 *Mappings* 部分，其中包含一个名为 *Mapping01* 的映射。

```
"Mappings" : {
    "Mapping01" : {
        "Key01" : {
            "Value" : "Value01"
        },
        "Key02" : {
            "Value" : "Value02"
        },
        "Key03" : {
            "Value" : "Value03"
        }
    }
}
```

在映射内，每个映射都是后面加有一个逗号的密钥和一组用括号括起来的名称-值对。密钥用户标识映射，它在映射中必须是唯一的。您可以在括号内声明多个名称-值对。每一个对都是格式为带双引号的名称、单个冒号和值或值阵列。

以下示例显示了包含映射 *RegionMap* 的 Mappings 部分，该映射包含五个映射到含单字符串值的名称/值对的密钥。密钥为区域名称。对于密钥所表示区域内的 32 位 AMI，每个名称-值对均为 AMI ID。

```
"Mappings" : {
      "RegionMap" : {
        "us-east-1" : { "32" : "ami-6411e20d"},
        "us-west-1" : { "32" : "ami-c9c7978c"},
        "eu-west-1" : { "32" : "ami-37c2f643"},
        "ap-southeast-1" : { "32" : "ami-66f28c34"},
        "ap-northeast-1" : { "32" : "ami-9c03a89d"}
      }
}
```

名称-值对中有一个名称（示例中的 32）和一个值。您可以通过给值命名以将多组值映射到密钥中。以下示例包含映射到两组值的区域密钥：一组名为 32，另一组名为 64。

```
"RegionMap" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
      "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
      "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
}
```

您可以使用 Fn::FindInMap (p. 501) 函数根据指定的密钥返回命名的值。以下示例模板包含由 *FindInMap* 函数分配其 ImageId 属性的 AWS::EC2::Instance (p. 272) 资源。*FindInMap* 函数将密钥指定为创建堆栈所在的区域（使用 AWS::Region 虚拟参数 (p. 510)）并将 32 指定为要映射到的值的名称。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
```

```
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
      "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
      "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
    }
  },

  "Resources" : {
     "myEC2Instance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
         "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
 }, "32"]},
           "InstanceType" : "m1.small"
        }
     }
 }
}
```

以下示例显示的是 Mappings 部分，该部分中的映射包含三个映射到含多个字符串值的阵列中的密钥。密钥表示三个区域，经映射的值是每个区域中所用可用区的列表。
AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 资源使用 *FindInMap* 函数和 *Region2AZ* 映射指定 *AvailabilityZones* 属性。

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",

    "Mappings" : {
      "Region2AZ" : {
        "us-west-1" : { "AZ" : ["us-west-1a", "us-west-1b"] },
        "us-east-1" : { "AZ" : ["us-east-1a", "us-east-1b", "us-east-1c"] },
        "eu-west-1" : { "AZ" : ["eu-west-1a", "eu-west-1b"] }
      }
    },

    "Resources" : {
      "MyELB" : {
        "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties" : {
          "AvailabilityZones" : { "Fn::FindInMap" : [ "Region2AZ", { "Ref" :
"AWS::Region" }, "AZ" ] },
          "Listeners" : [ {
            "LoadBalancerPort" : "8888" ,
            "InstancePort" : "8888" ,
            "Protocol" : "HTTP"
          } ],
          "HealthCheck" : {
           "Target" : { "Fn::Join" : [ "", ["HTTP:", "8888", "/"]]},
            "HealthyThreshold" : "5",
            "UnhealthyThreshold" : "2",
            "Interval" : "10",
            "Timeout" : "8"
          }
```

```
        }
      }
    }
}
```

条件声明

Abstract

通过使用内部函数比较值，在模板的 Conditions 部分中定义条件。

所有条件都是使用内部函数在模板的条件部分定义的。例如，通过这些内部函数，您可以比较两个值是否相等。根据条件结果，您可以按条件创建资源。

在每个条件中都可以引用其他条件、参数值或映射。创建或更新堆栈时，可以根据指定的输入参数更改这些条件、参数值或映射。定义所有条件后，您可以在模板的资源和输出部分将它们与资源和资源属性关联起来。

在创建或更新堆栈时，AWS CloudFormation 计算模板中的所有条件，然后创建资源。会创建与 true 条件关联的所有资源，忽略与 false 条件关联的所有资源。

下面的示例模板包含一个 `EnvType` 输入参数，在这里可以指定 `prod` 来创建生产堆栈，或指定 `test` 来创建测试堆栈。对于生产环境，AWS CloudFormation 会创建一个 Amazon EC2 并向一个实例附加一个卷。对于测试环境，AWS CloudFormation 只创建 Amazon EC2 实例。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Mappings" : {
    "RegionMap" : {
      "us-east-1"      : { "AMI" : "ami-7f418316", "TestAz" : "us-east-1a" },
      "us-west-1"      : { "AMI" : "ami-951945d0", "TestAz" : "us-west-1a" },
      "us-west-2"      : { "AMI" : "ami-16fd7026", "TestAz" : "us-west-2a" },
      "eu-west-1"      : { "AMI" : "ami-24506250", "TestAz" : "eu-west-1a" },
      "sa-east-1"      : { "AMI" : "ami-3e3be423", "TestAz" : "sa-east-1a" },
     "ap-southeast-1" : { "AMI" : "ami-74dda626", "TestAz" : "ap-southeast-1a"
 },
     "ap-southeast-2" : { "AMI" : "ami-b3990e89", "TestAz" : "ap-southeast-2a"
 },
     "ap-northeast-1" : { "AMI" : "ami-dcfa4edd", "TestAz" : "ap-northeast-1a"
 }
    }
  },

  "Parameters" : {
    "EnvType" : {
      "Description" : "Environment type.",
      "Default" : "test",
      "Type" : "String",
      "AllowedValues" : ["prod", "test"],
      "ConstraintDescription" : "must specify prod or test."
    }
  },

  "Conditions" : {
    "CreateProdResources" : {"Fn::Equals" : [{"Ref" : "EnvType"}, "prod"]}
```

```
  },

  "Resources" : {
    "EC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
}, "AMI" ]}
      }
    },

    "MountPoint" : {
      "Type" : "AWS::EC2::VolumeAttachment",
      "Condition" : "CreateProdResources",
      "Properties" : {
        "InstanceId" : { "Ref" : "EC2Instance" },
        "VolumeId"  : { "Ref" : "NewVolume" },
        "Device" : "/dev/sdh"
      }
    },

    "NewVolume" : {
      "Type" : "AWS::EC2::Volume",
      "Condition" : "CreateProdResources",
      "Properties" : {
        "Size" : "100",
       "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone"
]}
      }
    }
  },

  "Outputs" : {
    "VolumeId" : {
      "Value" : { "Ref" : "NewVolume" },
      "Condition" : "CreateProdResources"
    }
  }
}
```

如果 EnvType 参数与 prod 相等，CreateProdResources 条件将计算为 true。在示例模板中，NewVolume 和 MountPoint 资源与 CreateProdResources 条件关联。因此，仅当 EnvType 参数等于 prod 时才会创建资源。

您可以使用以下内部函数定义条件：

- Fn::And
- Fn::Equals
- Fn::If
- Fn::Not
- Fn::Or

有关每个内部函数语法的更多信息，请参阅条件函数 (p. 491)。

资源声明

Abstract

在模板的 Resources 部分中声明要用作堆栈一部分的 AWS 资源。

在 Resources 部分中，您可以将您想要的 AWS 资源声明为堆栈的一部分。资源之间用逗号隔开。

每一种资源都必须在模板中有一个唯一的逻辑名称。您可以在模板的其它地方将其名称用作解除参考参数。由于各个服务对资源名称的限制有所差异，所有资源的逻辑名称必须仅为字母数字组合[a – z，A – Z，0 – 9]。

您必须指定每种资源的类型。类型名称根据资源属性类型参考 (p. 419)中所列的那些名称而定。

如果资源无需声明任何属性，那么您可以忽略此资源的属性部分。有关声明 Property 部分的信息，请参阅属性声明 (p. 109)。

以下示例显示的是典型的资源声明。其中定义了两种资源。*MyInstance* 资源包含 *MyQueue* 资源作为其 *UserData* 属性的一部分：

```
  "Resources" : {
    "MyInstance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
            "UserData" : {
                "Fn::Base64" : {
                    "Fn::Join" : [ "", [ "Queue=", { "Ref" : "MyQueue" } ] ]
                 } },
            "AvailabilityZone" : "us-east-1a",
            "ImageId" : "ami-20b65349"
        }
    },

    "MyQueue" : {
        "Type" : "AWS::SQS::Queue",
        "Properties" : {
        }
    }
}
```

属性声明

Abstract

声明资源的属性。

属性声明见资源的 Properties 部分。此外，Outputs 部分中声明的输出也会遵循属性的规则。

资源类型声明之后会立即对每种资源进行 Properties 部分声明。多个属性之间用逗号隔开。每种属性都由带双引号的名称、单个冒号和该属性的值进行声明。

属性值可以是文字字符串、字符串列表、参数引用、伪引用或者功能返回的值。如果属性值为文件字符串，该值会被双引号括起来。如果值为任一类型的列表结果，则它会被括号 ("[ ]") 括起来。如果值为内部函数或参考的结果，则它会被大括号 ("{ }") 括起来。当您将文字、列表、参考和函数合并起来获取值时，上述规则适用。

以下示例显示了声明属性的几种方法。

```
"Properties" : {
    "MyString" : "one-string-value",
    "MyLiteralList" : [ "first-value", "second-value" ],
    "MyReferenceForOneValue" :  { "Ref" : "MyLogicalResourceName" } ,
    "MyFunctionResultWithFunctionParams" : {
        "Fn::Join" : [ "%", [ "Key=", { "Ref" : "MyParameter" } ] ] }
}
```

函数声明

Abstract

在模板中使用内部函数，以便为仅在运行时可用的属性分配值。

AWS CloudFormation 内部函数是可以在模板中使用的特殊操作，用于为仅在运行时可用的属性分配值。每个函数均使用带双引号的名称、单个冒号及其参数进行声明。如果参数为文字字符串，则会将其用双引号 (**""**) 括起来。如果参数在任一类型的列表中，则会将其用方括号 (**[ ]**) 括起来。如果参数为内部函数返回的值，则会将其用大括号 (**{ }**) 括起来。

以下示例显示了用于为 *MyLBDNSName* 分配值的函数 "Fn::GetAtt"，该函数通过从名为 *MyLoadBalancer* 的 Elastic Load Balancing 负载均衡器中检索属性 *DNSName* 的值，来执行此操作。

```
"Properties" : {
   "MyMyLBDNSName" : {
      "Fn::GetAtt" : [ "MyLoadBalancer", "DNSName" ]
   }
}
```

有关内部函数的更多信息，请参阅固有功能参考 (p. 490)。

输出声明

Abstract

通过在模板的 Outputs 部分中定义输出来返回一个或多个值。

使用模板的 Outputs 部分可以向用户返回一个或多个值以响应 aws cloudformation describe-stacks 命令。

要声明 Outputs 部分，请在带双引号的密钥名称 *Outputs* 后加上单个冒号。Outputs 部分中声明的所有输出都被括在一组大括号中，且中间用逗号隔开。

Note

如果有，Outputs 部分必须至少声明一个堆栈输出。一个 AWS CloudFormation 模板中最多可以有 60 个输出。

每一个输出都由一个带双引号的密钥名称、单个冒号和一个或多个属性组成。

以下列表描述了输出属性：

Value（*必需*）
由 aws cloudformation describe-stacks 命令返回的属性值。

Condition（可选）
用于指定条件的 String。要为返回的输出值设置要求，请引用在模板的 `Conditions` 部分中定义的条件。

Description（可选）
用于描述输出值的 String 类型，最大长度为 4K。

输出属性的声明方法类似于任何其它属性。在以下示例中，如果 `CreateProdResources` 条件为 true，则名为 *`LoadBalancer`* 的输出将返回逻辑名称为 *`BackupLoadBalancer`* 的资源的信息。

```
"Outputs" : {
   "LoadBalancer" : {
      "Condition" : "CreateProdResources",
      "Value" : { "Ref" : "BackupLoadBalancer" }
   }
}
```

示例模板

编写示例 AWS CloudFormation 模板是为了显示 AWS CloudFormation 的功能，并将其作为您创建自定义堆栈的起点。我们将提供以下堆栈应用程序。在下面部分中，我们将说明模板、模板各部分及其可能具有的任何特殊功能的详细情况。也将包括模板最新源代码的链接。

Topics

访问以下网址可获取更多示例模板：http://amazonaws.cn/cloudformation/aws-cloudformation-templates/。此外，我们还将定期添加新示例模板，为新添加的支持功能提供示例。请浏览 AWS CloudFormation 开发论坛查看通告。另外，其他 AWS CloudFormation 用户可能已开发了一些模板来提供自定义解决方案，并且可能也会将其 AWS CloudFormation 解决方案发布到论坛中。

带有负载均衡器、Auto Scaling 策略和 CloudWatch 警报的 Auto Scaling 组

Topics

本模板将创建运用 Auto Scaling 和 Elastic Load Balancing 的示例网站，并配置用于使用多个可用区。模板还包含 Amazon CloudWatch 警报，在超过已定义的阈值时，这些警报将执行 Auto Scaling 策略以在 Auto Scaling 组中添加或删除实例。

Important

本模板将创建一个或多个 Amazon EC2 实例。如果您通过本模板创建堆栈，那么会针对 AWS 资源向您收取相应费用。

利用以下命令，方可通过本示例模板创建堆栈：

```
aws cloudformation create-stack --stack-name StackName --template-url ht
tps://s3.amazonaws.com/cloudformation-templates-us-east-1/AutoScalingMultiAZ
Sample.template --parameters ParameterKey=KeyName,ParameterValue=key-pair-name
```

或者单击以下链接，在 US-East Region 中创建堆栈：

https://console.aws.amazon.com/cloudformation/home?region=us-east-1#stack=AutoScalingMultiAZSample&templateURL=https://s3.amazonaws.com/cloudformation-templates-us-east-1/AutoScalingMultiAZSample.template

您可以从以下网址获取此示例模板的最新版本：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/AutoScalingMultiAZSample.template。

Auto Scaling 多可用区模板

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "Create a multi-az, load balanced, Auto Scaled sample web site.
 The Auto Scaling trigger is based on the CPU utilization of the web servers. The
 AMI is chosen based on the region in which the stack is run. This example creates
 a web service running across all availability zones in a region. The instances
are load balanced with a simple health check. The web site is available on port
80, however, the instances can be configured to listen on any port (8888 by de
fault). **WARNING** This template creates one or more Amazon EC2 instances. You
will be billed for the AWS resources used if you create a stack from this tem
plate.",

  "Parameters" : {
    "InstanceType" : {
      "Description" : "Type of EC2 instance to launch",
      "Type" : "String",
      "Default" : "m1.small"
    },
    "WebServerPort" : {
      "Description" : "The TCP port for the Web Server",
      "Type" : "String",
      "Default" : "8888"
    },
    "KeyName" : {
      "Description" : "The EC2 Key Pair to allow SSH access to the instances",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "64" },
      "m1.small"    : { "Arch" : "32" },
      "m1.large"    : { "Arch" : "64" },
      "m1.xlarge"   : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m2.2xlarge"  : { "Arch" : "64" },
      "m2.4xlarge"  : { "Arch" : "64" },
      "c1.medium"   : { "Arch" : "32" },
      "c1.xlarge"   : { "Arch" : "64" },
      "cc1.4xlarge" : { "Arch" : "64" }
```

```
    },
    "AWSRegionArch2AMI" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
      "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
      "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
    }
  },

  "Resources" : {
    "WebServerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : ""},
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : "1",
        "MaxSize" : "3",
        "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
      }
   },

    "LaunchConfig" : {
      "Type" : "AWS::AutoScaling::LaunchConfiguration",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                          { "Fn::FindInMap" : [ "AWSInstance
Type2Arch", { "Ref" : "InstanceType" },
                                          "Arch" ] } ] },
        "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }},
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "InstanceType" : { "Ref" : "InstanceType" }
      }
    },

    "WebServerScaleUpPolicy" : {
      "Type" : "AWS::AutoScaling::ScalingPolicy",
      "Properties" : {
        "AdjustmentType" : "ChangeInCapacity",
        "AutoScalingGroupName" : { "Ref" : "WebServerGroup" },
        "Cooldown" : "60",
        "ScalingAdjustment" : "1"
      }
    },
    "WebServerScaleDownPolicy" : {
      "Type" : "AWS::AutoScaling::ScalingPolicy",
      "Properties" : {
        "AdjustmentType" : "ChangeInCapacity",
        "AutoScalingGroupName" : { "Ref" : "WebServerGroup" },
        "Cooldown" : "60",
        "ScalingAdjustment" : "-1"
      }
    },

    "CPUAlarmHigh": {
     "Type": "AWS::CloudWatch::Alarm",
```

```
     "Properties": {
        "AlarmDescription": "Scale-up if CPU > 90% for 10 minutes",
        "MetricName": "CPUUtilization",
        "Namespace": "AWS/EC2",
        "Statistic": "Average",
        "Period": "300",
        "EvaluationPeriods": "2",
        "Threshold": "90",
        "AlarmActions": [ { "Ref": "WebServerScaleUpPolicy" } ],
        "Dimensions": [
          {
            "Name": "AutoScalingGroupName",
            "Value": { "Ref": "WebServerGroup" }
          }
        ],
        "ComparisonOperator": "GreaterThanThreshold"
      }
    },
    "CPUAlarmLow": {
     "Type": "AWS::CloudWatch::Alarm",
     "Properties": {
        "AlarmDescription": "Scale-down if CPU < 70% for 10 minutes",
        "MetricName": "CPUUtilization",
        "Namespace": "AWS/EC2",
        "Statistic": "Average",
        "Period": "300",
        "EvaluationPeriods": "2",
        "Threshold": "70",
        "AlarmActions": [ { "Ref": "WebServerScaleDownPolicy" } ],
        "Dimensions": [
          {
            "Name": "AutoScalingGroupName",
            "Value": { "Ref": "WebServerGroup" }
          }
        ],
        "ComparisonOperator": "LessThanThreshold"
      }
    },

    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : { "Ref" : "WebServerPort" },
          "Protocol" : "HTTP"
        } ],
        "HealthCheck" : {
          "Target" : { "Fn::Join" : [ "", ["HTTP:", { "Ref" : "WebServerPort" },
"/"]]},
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
      }
    },
```

```
    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access and HTTP access on the inbound
port",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        },
        {
          "IpProtocol" : "tcp",
          "FromPort" : { "Ref" : "WebServerPort" },
          "ToPort" : { "Ref" : "WebServerPort" },
          "SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
"SourceSecurityGroup.OwnerAlias"]},
          "SourceSecurityGroupName" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
"SourceSecurityGroup.GroupName"]}
        } ]
      }
    }
  },

  "Outputs" : {
    "URL" : {
      "Description" : "The URL of the website",
     "Value" :  { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [ "ElasticLoad
Balancer", "DNSName" ]}]]}
    }
  }
}
```

模板演练

该示例模板包含带负载均衡器的 Auto Scaling 组、用于定义入口规则的安全组、Amazon CloudWatch 警报和 Auto Scaling 策略。

该模板具有三个输入参数：InstanceType 是要用于 Auto Scaling 组的 EC2 实例的类型，其默认值为 m1.small；WebServerPort 是 Web 服务器的 TCP 端口，其默认值为 8888；KeyName 是要用于 Auto Scaling 组的 EC2 密钥对的名称。必须在堆栈创建时指定 KeyName（堆栈创建时，必须指定不带默认值的参数）。

AWS::AutoScaling::AutoScalingGroup (p. 219) 资源 WebServerGroup 声明以下 Auto Scaling 组配置：

- *AvailabilityZones* 指定将创建 Auto Scaling 组的 EC2 实例的可用区。Fn::GetAZs (p. 505) 函数调用 `{ "Fn::GetAZs" : "" }` 可为要创建堆栈的区域指定所有可用区。
- *MinSize* 和 *MaxSize* 设置 Auto Scaling 组中的 EC2 实例数的最小值和最大值。
- *LoadBalancerNames* 将列出用于将流量路由至 Auto Scaling 组的负载均衡器。该组的负载均衡器（LoadBalancer）是弹性负载均衡器（ElasticLoadBalancer）资源。

AWS::AutoScaling::LaunchConfiguration (p. 224) 资源 LaunchConfig 声明以下配置，这些配置将用于 WebServerGroup Auto Scaling 组中的 EC2 实例：

- *KeyName* 获取 KeyName 输入参数的值作为要使用的 EC2 密钥对。
- *UserData* 为 WebServerPort 参数的 Base64 编码值，该参数将传递给应用程序。
- *SecurityGroups* 为 EC2 安全组的列表，EC2 安全组包含 Auto Scaling 组中 EC2 实例的防火墙入口规则。在此示例中，仅有一个安全组，该安全组声明为 AWS::EC2::SecurityGroup (p. 292) 资源：InstanceSecurityGroup。该安全组包含两个入口规则：1) TCP 入口规则，该规则允许从端口 22（用于 SSH 访问）的所有 IP 地址 ("CidrIp" : "0.0.0.0/0") 进行访问；2) TCP 入口规则，该规则通过指定负载均衡器的源安全组来允许从 WebServerPort 端口的 ElasticLoadBalancer 资源进行访问。GetAtt (p. 502) 函数用于从 ElasticLoadBalancer 资源获取 SourceSecurityGroup.OwnerAlias 和 SourceSecurityGroup.GroupName 属性。有关 Elastic Load Balancing 安全组的更多信息，请参阅在 Amazon EC2-Classic 中管理安全组或在 Amazon VPC 中管理安全组。
- *ImageId* 是一组嵌套映射的评估值。我们将添加映射，因此，模板包括用于选择正确映像 ID 的逻辑。该逻辑基于使用 InstanceType 参数指定的实例类型（AWSInstanceType2Arch 将实例类型映射到架构 32 或 64）和将创建堆栈的区域（AWSRegionArch2AMI 将区域和架构映射到映像 ID）：

```
{ "Fn::FindInMap" : [ "AWSRegionArch2AMI",
   { "Ref" : "AWS::Region" },
   { "Fn::FindInMap" : [ "AWSInstanceType2Arch",
      { "Ref" : "InstanceType" },
      "Arch" ]
   }
   ]}
```

例如，如果您使用此模板在 us-east-1 区域中创建堆栈，并指定 InstanceType 为 m1.small，则 AWS CloudFormation 会将 AWSInstanceType2Arch 的内部映射评估为以下内容：

```
{ "Fn::FindInMap" : [ "AWSInstanceType2Arch", "m1.small", "Arch" ] }
```

在 AWSInstanceType2Arch 映射中，m1.small 密钥的架构值将映射到 32，该值用作外部映射的值。密钥是 AWS::Region 评估结果，其为将创建堆栈的区域。在此示例中，AWS::Region 为 us-east-1；因此，将按以下方式评估外部映射：

```
Fn::FindInMap" : [ "AWSRegionArch2AMI", "us-east-1", "32"]
```

在 AWSRegionArch2AMI 映射中，密钥 us-east-1 的值 32 将映射到 ami-6411e20d。这表示 ImageId would be ami-6411e20d。

AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 资源 ElasticLoadBalancer 声明以下负载均衡器配置：

- *AvailabilityZones* 是负载均衡器将分配流量的可用区列表。在此示例中，Fn::GetAZs 函数调用 `{ "Fn::GetAZs" : "" }` 指定了创建堆栈所在区域的所有可用区。
- *Listeners* 为负载均衡路由配置列表，负载均衡路由配置指定负载均衡器接受请求的端口、负载均衡器转发请求的已注册 EC2 实例上的端口，以及用于路由请求的协议。
- *HealthCheck* 是 Elastic Load Balancing 用于检查负载均衡器将流量路由到的 EC2 实例的运行状况的配置。在该示例中，HealthCheck 旨在通过 WebServerPort 在 HTTP 协议上指定的端口来获得 EC2 实例的原地址。如果 WebServerPort 为 8888，则 `{ "Fn::Join" : [ "", ["HTTP:", { "Ref" : "WebServerPort" }, "/"]]}` 函数调用将评估为字符串 `HTTP:8888/`。它还指定 EC2 实例在两项运行状况检查之间拥有 30 秒的时间间隔。Timeout（超时）将定义为 Elastic Load Balancing 等待来自运行状况检查响应的时间（本示例中为 5 秒）。超时周期失效后，Elastic Load Balancing 将标记 EC2 实例运行状况检查不合格。当 EC2 实例未通过连续 5 次运行状况检查时 (UnhealthyThreshold)，Elastic Load Balancing 会停止路由流量值该 EC2 实例，直至该实例连续 3 次运行状况检查情况良好，此时 Elastic Load Balancing 将认定 EC2 实例运行良好，并再次向该实例路由流量。

AWS::AutoScaling::ScalingPolicy (p. 230) 资源 WebServerScaleUpPolicy 为用于扩展 Auto Scaling 组 WebServerGroup 的 Auto Scaling 策略。*AdjustmentType* 属性设定为 ChangeInCapacity。这意味着 *ScalingAdjustment* 表示要添加的实例数（如果 *ScalingAdjustment* 为正值，则添加实例；若为负值，则删除实例）。在此示例中，*ScalingAdjustment* 为 1；因此，执行策略时，策略将以 1 为增量增加 EC2 实例的数量。Cooldown 属性指定 Auto Scaling 在启动任何其他策略或触发相关操作之前将等待 60 秒。

AWS::CloudWatch::Alarm (p. 257) 资源 CPUAlarmHigh 指定扩展策略 WebServerScaleUpPolicy 作为在警报处于 *ALARM* 状态 (AlarmActions) 时要执行的操作。警报将监测 WebServerGroup Auto Scaling 组中的 EC2 实例（维度）。警报每隔 300 秒（周期）测量在 WebServerGroup（维度）中实例的平均（统计数据）EC2 实例 CPU 使用率（命名空间和标准名称）。当该值（300 秒以上时间内的平均 CPU 使用率）连续 2 个周期 (EvaluationPeriod) 保持高于 90%（ComparisonOperator 和阈值）的状态，则警报将进入 *ALARM* 状态，并且 Amazon CloudWatch 将执行上述 WebServerScaleUpPolicy 策略 (AlarmActions)，扩展 WebServerGroup。

CPUAlarmLow 警报测量相同的指标，但有一项警报会在 CPU 使用率低于 75% 时触发（ComparisonOperator 和阈值），并执行 WebServerScaleDownPolicy 策略，将 1 个 EC2 实例从 Auto Scaling 组 WebServerGroup 中删除。

运行 Amazon Linux 32 位 AMI 的 Amazon EC2

此模板声明有一个参数和四个映射。资源包括一个 Amazon EC2 实例和一个安全组。映射使用 AWS::Region pseudo 参数选择适当的 AMI。Outputs 部分打印有实例的 ID、创建实例的可用区和它的公用 IP 地址。

利用以下命令，方可通过本示例模板创建堆栈：

```
aws cloudformation create-stack --stack-name StackName ----template-body
file:///home/local/EC2InstanceWithSecurityGroupSample-1.0.0.template --parameters
 ParameterKey=KeyName,ParameterValue=key-pair-name
```

您可以从以下网址获取此示例模板的最新版本：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/EC2InstanceWithSecurityGroupSample-1.0.0.template。

Amazon Linux 32 位 AMI 示例模板

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "Create an EC2 instance running the Amazon Linux 32 bit AMI. The
 AMI is chosen based on the region in which the stack is run. This example creates
 an EC2 security group for the instance to give you SSH access. **WARNING** This
template creates one or more Amazon EC2 instances. You will be billed for the AWS
 resources used if you create a stack from this template.",

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of and existing EC2 KeyPair to enable SSH access to
the instance",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
          "AMI" : "ami-76f0061f"
```

```
      },
      "us-west-1" : {
          "AMI" : "ami-655a0a20"
      },
      "eu-west-1" : {
          "AMI" : "ami-7fd4e10b"
      },
      "ap-southeast-1" : {
          "AMI" : "ami-72621c20"
      },
      "ap-northeast-1" : {
          "AMI" : "ami-8e08a38f"
      }
    }
  },

  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" },
 "AMI" ]}
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  },

  "Outputs" : {
    "InstanceId" : {
      "Description" : "InstanceId of the newly created EC2 instance",
      "Value" : { "Ref" : "Ec2Instance" }
    },
    "AZ" : {
      "Description" : "Availability Zone of the newly created EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone" ] }
    },
    "PublicIP" : {
      "Description" : "Public IP address of the newly created EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicIp" ] }
    }
  }
}
```

创建一个负载均衡 Apache 网站

此模板声明两个参数和四个映射。资源包括一个带侦听器和运行状况检查功能的 Elastic Load Balancing 负载均衡器、两个 Amazon EC2 实例和一个安全组。Outputs 部分打印有负载均衡器的 URL。

利用以下命令，方可通过本示例模板创建堆栈：

```
aws cloudformation create-stack --stack-name StackName ----template-body
file:///home/local/ELBSample-1.0.0.template --parameters ParameterKey=Key
Name,ParameterValue=key-pair-name
```

您可以从以下网址获取此示例模板的最新版本：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/ELBSample-1.0.0.template。

负载均衡 Apache 网站示例模板

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "Create a load balanced sample web site. The AMI is chosen based
 on the region in which the stack is run. This example creates 2 EC2 instances
behind a load balancer with a simple health check. The instances may be created
in one or more AZs. The web site is available on port 80, however, the instances
can be configured to listen on any port (8888 by default). **WARNING** This template
 creates one or more Amazon EC2 instances. You will be billed for the AWS resources
 used if you create a stack from this template.",

  "Parameters" : {
    "InstanceType" : {
      "Description" : "Type of EC2 instance to launch",
      "Type" : "String",
      "Default" : "m1.small"
    },
    "WebServerPort" : {
      "Description" : "TCP/IP port of the web server",
      "Type" : "String",
      "Default" : "8888"
    },
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the
 instances",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "64" },
      "m1.small"    : { "Arch" : "32" },
      "m1.large"    : { "Arch" : "64" },
      "m1.xlarge"   : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m2.2xlarge"  : { "Arch" : "64" },
      "m2.4xlarge"  : { "Arch" : "64" },
      "c1.medium"   : { "Arch" : "32" },
      "c1.xlarge"   : { "Arch" : "64" },
```

```
      "cc1.4xlarge" : { "Arch" : "64" }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
      "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
      "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
    }
  },

  "Resources" : {
    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "Instances" : [ { "Ref" : "Ec2Instance1" },{ "Ref" : "Ec2Instance2" } ],
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : { "Ref" : "WebServerPort" },
          "Protocol" : "HTTP"
        } ],
        "HealthCheck" : {
          "Target" : { "Fn::Join" : [ "", ["HTTP:", { "Ref" : "WebServerPort" },
"/"]]},
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
      }
    },

    "Ec2Instance1" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                          { "Fn::FindInMap" : [ "AWSInstance
Type2Arch", { "Ref" : "InstanceType" },
                                          "Arch" ] } ] },
        "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }}
      }
    },

    "Ec2Instance2" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                          { "Fn::FindInMap" : [ "AWSInstance
Type2Arch", { "Ref" : "InstanceType" },
                                          "Arch" ] } ] },
        "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }}
```

```
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access and HTTP access on the inbound
port",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        },
        {
          "IpProtocol" : "tcp",
          "FromPort" : { "Ref" : "WebServerPort" },
          "ToPort" : { "Ref" : "WebServerPort" },
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  },

  "Outputs" : {
    "URL" : {
      "Description" : "URL of the sample website",
     "Value" :  { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [ "ElasticLoad
Balancer", "DNSName" ]}]]}
    }
  }
}
```

使用竞价型实例监控SQS队列中的工作的自动扩展型工作程序

Abstract

使用此示例模板可通过竞价型实例创建监控 SQS 队列中的工作（消息）的自动扩展型工作程序。

此模板使用竞价型实例创建监控 SQS 队列中的工作（消息）的自动扩展型工作程序。应用程序根据队列中的工作量自动扩展。有工作时，Auto Scaling 扩大容量；没有工作时，Auto Scaling 缩小容量。每条消息均包含要运行的命令或脚本、输入文件位置和结果输出位置。

利用以下命令，方可通过本示例模板创建堆栈：

```
aws cloudformation create-stack --stack-name StackName --template-url ht
tps://s3.amazonaws.com/cloudformation-templates-us-east-1/worker-role.template
 --parameters ParameterKey=KeyName,ParameterValue=key-pair-name
```

或者单击以下链接，以在美国东部区域创建堆栈：

您可以从以下网址获取此示例模板的最新版本：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/worker-role.template。

WorkerRole 模板

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template WorkerRole: Create a multi-
az, Auto Scaled worker that pulls command messages from a queue and execs the
command. Each message contains a command/script to run, an input file location and
 an output location for the results. The application is Auto-Scaled based on the
amount of work in the queue. **WARNING** This template creates one or more Amazon
 EC2 instances and an Amazon SQS queue. You will be billed for the AWS resources
used if you create a stack from this template.",

  "Parameters" : {
    "InstanceType" : {
      "Description" : "Worker EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro","m1.small","m1.medi
um","m1.large","m1.xlarge","m2.xlarge","m2.2xlarge","m2.4xlarge","c1.medi
um","c1.xlarge","cc1.4xlarge","cc2.8xlarge","cg1.4xlarge"],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },

    "KeyName" : {
      "Description" : "The EC2 Key Pair to allow SSH access to the instances",
      "Type" : "String"
    },

    "MinInstances" : {
      "Description" : "The minimum number of Workers",
      "Type" : "Number",
      "MinValue" : "0",
      "Default"  : "0",
      "ConstraintDescription" : "Enter a number >=0"
    },

    "MaxInstances" : {
      "Description" : "The maximum number of Workers",
      "Type" : "Number",
      "MinValue" : "1",
      "Default"  : "1",
      "ConstraintDescription" : "Enter a number >1"
    }
  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "64" },
      "m1.small"    : { "Arch" : "64" },
      "m1.medium"   : { "Arch" : "64" },
      "m1.large"    : { "Arch" : "64" },
      "m1.xlarge"   : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m2.2xlarge"  : { "Arch" : "64" },
      "m2.4xlarge"  : { "Arch" : "64" },
      "c1.medium"   : { "Arch" : "64" },
```

```
      "c1.xlarge"   : { "Arch" : "64" },
      "cc1.4xlarge" : { "Arch" : "64HVM" },
      "cc2.8xlarge" : { "Arch" : "64HVM" },
      "cg1.4xlarge" : { "Arch" : "64HVM" }
    },

    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "32" : "ami-31814f58", "64" : "ami-1b814f72", "64HVM"
: "ami-0da96764" },
      "us-west-2"      : { "32" : "ami-38fe7308", "64" : "ami-30fe7300", "64HVM"
: "NOT_YET_SUPPORTED" },
      "us-west-1"      : { "32" : "ami-11d68a54", "64" : "ami-1bd68a5e", "64HVM"
: "NOT_YET_SUPPORTED" },
      "eu-west-1"      : { "32" : "ami-973b06e3", "64" : "ami-953b06e1", "64HVM"
: "NOT_YET_SUPPORTED" },
      "ap-southeast-1" : { "32" : "ami-b4b0cae6", "64" : "ami-beb0caec", "64HVM"
: "NOT_YET_SUPPORTED" },
      "ap-northeast-1" : { "32" : "ami-0644f007", "64" : "ami-0a44f00b", "64HVM"
: "NOT_YET_SUPPORTED" },
      "sa-east-1"      : { "32" : "ami-3e3be423", "64" : "ami-3c3be421", "64HVM"
: "NOT_YET_SUPPORTED" }
    }
  },

  "Resources" : {

    "WorkerUser" : {
      "Type" : "AWS::IAM::User",
      "Properties" : {
        "Path": "/",
        "Policies": [{
          "PolicyName": "root",
          "PolicyDocument": {
            "Version": "2012-10-17",
            "Statement":[{
              "Effect": "Allow",
              "Action": [
                "cloudformation:DescribeStackResource",
                "sqs:ReceiveMessage",
                "sqs:DeleteMessage",
                "sns:Publish"
              ],
              "Resource": "*"
            }]
          }
        }]
      }
    },

    "WorkerKeys" : {
      "Type" : "AWS::IAM::AccessKey",
      "Properties" : {
        "UserName" : {"Ref": "WorkerUser"}
      }
    },

    "InputQueue" : {
      "Type" : "AWS::SQS::Queue"
```

```
    },

    "InputQueuePolicy" : {
      "Type" : "AWS::SQS::QueuePolicy",
      "DependsOn" : "LaunchConfig",
      "Properties" : {
        "Queues" : [ { "Ref" : "InputQueue" } ],
        "PolicyDocument":  {
          "Version": "2012-10-17",
          "Id": "ReadFromQueuePolicy",
          "Statement" : [ {
            "Sid": "ConsumeMessages",
            "Effect": "Allow",
            "Principal" : { "AWS": {"Fn::GetAtt" : ["WorkerUser", "Arn"]} },
            "Action": ["sqs:ReceiveMessage", "sqs:DeleteMessage"],
            "Resource": { "Fn::GetAtt" : [ "InputQueue", "Arn" ] }
          } ]
        }
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access",
        "SecurityGroupIngress" : [ { "IpProtocol" : "tcp", "FromPort" : "22",
"ToPort" : "22", "CidrIp" : "0.0.0.0/0" } ]
      }
    },

    "LaunchConfig" : {
      "Type" : "AWS::AutoScaling::LaunchConfiguration",
      "Metadata" : {
        "Comment" : "Install a simple PHP application",
        "AWS::CloudFormation::Init" : {
          "configSets" : {
            "ALL" : ["XML", "Time", "LWP", "AmazonLibraries", "WorkerRole"]
          },
          "XML" : {
            "packages" : {
              "yum" : {
                "perl-XML-Simple"         : []
              }
            }
          },
          "Time" : {
            "packages" : {
              "yum" : {
                "perl-LWP-Protocol-https" : []
              }
            }
          },
          "LWP" : {
            "packages" : {
              "yum" : {
                "perl-Time-HiRes"         : []
              }
            }
```

```
          },
          "AmazonLibraries" : {
            "sources" : {
              "/home/ec2-user/sqs" : "http://s3.amazonaws.com/awscode/amazon-
queue/2009-02-01/perl/library/amazon-queue-2009-02-01-perl-library.zip"
            }
          },
          "WorkerRole" : {
            "files" : {
              "/etc/cron.d/worker.cron" : {
                "content" : "*/1 * * * * ec2-user /home/ec2-user/worker.pl &>
/home/ec2-user/worker.log\n",
                "mode"    : "000644",
                "owner"   : "root",
                "group"   : "root"
              },

              "/home/ec2-user/worker.pl" : {
                "content" : { "Fn::Join" : ["", [
                  "#!/usr/bin/perl -w\n",
                  "#\n",
                  "use strict;\n",
                  "use Carp qw( croak );\n",
                  "use lib qw(/home/ec2-user/sqs/amazon-queue-2009-02-01-perl-
library/src);  \n",
                  "use LWP::Simple qw( getstore );\n",
                  "\n",
                  "my $AWS_ACCESS_KEY_ID     = \"", { "Ref" : "WorkerKeys" },
"\";\n",
                 "my $AWS_SECRET_ACCESS_KEY = \"", { "Fn::GetAtt": ["WorkerKeys",
 "SecretAccessKey"]}, "\";\n",
                  "my $QUEUE_NAME            = \"", { "Ref" : "InputQueue" },
"\";\n",
                  "my $COMMAND_FILE          = \"/home/ec2-user/command\";\n",
                  "\n",
                  "eval {\n",
                  "\n",
                  "  use Amazon::SQS::Client; \n",
                  "  my $service = Amazon::SQS::Client->new($AWS_ACCESS_KEY_ID,
$AWS_SECRET_ACCESS_KEY);\n",
                  " \n",
                "  my $response = $service->receiveMessage({QueueUrl=>$QUEUE_NAME,
 MaxNumberOfMessages=>1});\n",
                  "  if ($response->isSetReceiveMessageResult) {\n",
                  "    my $result = $response->getReceiveMessageResult();\n",
                  "    if ($result->isSetMessage) {\n",
                  "      my $messageList = $response->getReceiveMessageResult()-
>getMessage();\n",
                  "      foreach(@$messageList) {\n",
                  "        my $message = $_;\n",
                  "        my $messageHandle = 0;\n",
                  "        if ($message->isSetReceiptHandle()) {\n",
                  "          $messageHandle = $message->getReceiptHandle();\n",
                  "        } else {\n",
                  "          croak \"Couldn't get message Id from message\";\n",
                  "        }\n",
                  "        if ($message->isSetBody()) {\n",
                  "          my %parameters = split(/[=;]/, $message->get
```

```
Body());\n",
                  "          if (defined($parameters{\"Input\"}) &&
defined($parameters{\"Output\"}) && defined($parameters{\"Command\"})) {\n",
                  "            getstore($parameters{\"Command\"}, $COM
MAND_FILE);\n",
                  "            chmod(0755, $COMMAND_FILE);\n",
                  "            my $command = $COMMAND_FILE . \" \" . $paramet
ers{\"Input\"} . \" \" . $parameters{\"Output\"};\n",
                  "            my $result = `$command`;\n",
                  "            print \"Result = \" . $result . \"\\n\";\n",
                  "          } else {\n",
                  "            croak \"Invalid message\";\n",
                  "          }\n",
                  "        } else {\n",
                 "          croak \"Couldn't get message body from message\";\n",

                  "        }\n",
                  "        my $response = $service->deleteMes
sage({QueueUrl=>$QUEUE_NAME, ReceiptHandle=>$messageHandle});\n",
                  "      }\n",
                  "    } else {\n",
                  "      printf \"Empty Poll\\n\";\n",
                  "    }\n",
                  "  } else {\n",
                  "    croak \"Call failed\";\n",
                  "  }\n",
                  "}; \n",
                  "\n",
                  "my $ex = $@;\n",
                  "if ($ex) {\n",
                  "  require Amazon::SQS::Exception;\n",
                  "  if (ref $ex eq \"Amazon::SQS::Exception\") {\n",
                  "    print(\"Caught Exception: \" . $ex->getMessage() .
\"\\n\");\n",
                  "  } else {\n",
                  "    croak $@;\n",
                  "  }\n",
                  "}\n"
                ]]},
                "mode"    : "000755",
                "owner"   : "ec2-user",
                "group"   : "ec2-user"
              }
            }
          }
        }
      },
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "SpotPrice" : "0.05",
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                          { "Fn::FindInMap" : [ "AWSInstance
Type2Arch", { "Ref" : "InstanceType" },
                                          "Arch" ] } ] },
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "InstanceType" : { "Ref" : "InstanceType" },
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
```

```
          "#!/bin/bash\n",
          "yum update -y aws-cfn-bootstrap\n",
          "# Install the Worker application\n",
          "/opt/aws/bin/cfn-init ",
          "         --stack ", { "Ref" : "AWS::StackName" },
          "         --resource LaunchConfig ",
          "         --configset ALL",
          "         --access-key ", { "Ref" : "WorkerKeys" },
          "         --secret-key ", {"Fn::GetAtt": ["WorkerKeys", "SecretAccess
Key"]},
          "         --region ", { "Ref" : "AWS::Region" }, "\n"
        ]]}}
      }
    },

    "WorkerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : ""},
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : { "Ref" : "MinInstances" },
        "MaxSize" : { "Ref" : "MaxInstances" }
      }
    },

    "WorkerScaleUpPolicy" : {
      "Type" : "AWS::AutoScaling::ScalingPolicy",
      "Properties" : {
        "AdjustmentType" : "ChangeInCapacity",
        "AutoScalingGroupName" : { "Ref" : "WorkerGroup" },
        "Cooldown" : "60",
        "ScalingAdjustment" : "1"
      }
    },

    "WorkerScaleDownPolicy" : {
      "Type" : "AWS::AutoScaling::ScalingPolicy",
      "Properties" : {
        "AdjustmentType" : "ChangeInCapacity",
        "AutoScalingGroupName" : { "Ref" : "WorkerGroup" },
        "Cooldown" : "60",
        "ScalingAdjustment" : "-1"
      }
    },

    "TooManyMessagesAlarm": {
     "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "Scale-Up if queue depth grows beyond 10 messages",
        "Namespace": "AWS/SQS",
        "MetricName": "ApproximateNumberOfMessagesVisible",
        "Dimensions": [{ "Name": "QueueName", "Value" : { "Fn::GetAtt" : ["In
putQueue", "QueueName"] } }],
        "Statistic": "Sum",
        "Period": "60",
        "EvaluationPeriods": "3",
        "Threshold": "1",
        "ComparisonOperator": "GreaterThanThreshold",
```

```
        "AlarmActions": [ { "Ref": "WorkerScaleUpPolicy" } ]
      }
    },

    "NotEnoughMessagesAlarm": {
     "Type": "AWS::CloudWatch::Alarm",
     "Properties": {
        "AlarmDescription": "Scale-down if there are too many empty polls, indic
ating there is not enough work",
        "Namespace": "AWS/SQS",
        "MetricName": "NumberOfEmptyReceives",
        "Dimensions": [{ "Name": "QueueName", "Value" : { "Fn::GetAtt" : ["In
putQueue", "QueueName"] } }],
        "Statistic": "Sum",
        "Period": "60",
        "EvaluationPeriods": "10",
        "Threshold": "3",
        "ComparisonOperator": "GreaterThanThreshold",
        "AlarmActions": [ { "Ref": "WorkerScaleDownPolicy" } ]
      }
    }
  },

  "Outputs" : {
    "QueueURL" : {
      "Description" : "URL of input queue",
      "Value" : { "Ref" : "InputQueue" }
    }
  }
}
```

模板代码段

Abstract

通过这些示例场景可了解 AWS CloudFormation 模板如何与其他 AWS 服务结合使用。

本部分提供了很多示例案例，您可以使用这些案例来了解如何声明各个 AWS CloudFormation 模板部分。您还可以将代码段用作您的自定义模板的起始部分。

Note

由于 AWS CloudFormation 模板必须符合 JSON，因此没有对行继续符进行预配置。如果行的长度大于 80 字符，本文件中的代码段可随机换行。

Topics

Auto Scaling 代码段

Topics

Auto Scaling 启动配置资源

此示例显示了 Auto Scaling AWS::AutoScaling::LaunchConfiguration 资源。SecurityGroups 属性指定名为 myEC2SecurityGroup 的 AWS::EC2::SecurityGroup 资源和名为 myExistingEC2SecurityGroup 的现有 EC2 安全组。BlockDeviceMappings 属性列出了两个设备：映射到 /dev/sdk 的 50 GB EBS 卷和映射到 /dev/sdc 的虚拟设备 ephemeral0。

```
"SimpleConfig" : {
   "Type" : "AWS::AutoScaling::LaunchConfiguration",
   "Properties" : {
      "ImageId" : "ami-6411e20d",
      "SecurityGroups" : [ { "Ref" : "myEC2SecurityGroup" }, "myExistingEC2Se
curityGroup" ],
      "InstanceType" : "m1.small",
      "BlockDeviceMappings" : [ {
            "DeviceName" : "/dev/sdk",
            "Ebs" : {"VolumeSize" : "50"}
         }, {
            "DeviceName" : "/dev/sdc",
            "VirtualName" : "ephemeral0"
      } ]
   }
},
```

Auto Scaling 组资源

此示例显示了 Auto Scaling AWS::AutoScaling::AutoScalingGroup (p. 219) 资源。AvailabilityZones 属性指定创建 Auto Scaling 组 EC2 实例的可用区。在此示例中，Fn::GetAZs (p. 505) 函数调用 `{ "Fn::GetAZs" : "" }` 指定了创建堆栈所在区域的所有可用区。LoadBalancerNames 属性列出了将流量路由到 Auto Scaling 组所用的 LoadBalancers。在此示例中，指定了一个负载均衡器，即 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 资源 LB。

```
"MyServerGroup" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : ""},
      "LaunchConfigurationName" : { "Ref" : "SimpleConfig" },
      "MinSize" : "1",
      "MaxSize" : "3",
      "LoadBalancerNames" : [ { "Ref" : "LB" } ]
   }
},
```

由 Amazon CloudWatch 警报触发的 Auto Scaling 策略

此示例显示了用于扩展 Auto Scaling 组 asGroup 的 AWS::AutoScaling::ScalingPolicy (p. 230) 资源。*AdjustmentType* 属性指定 ChangeInCapacity，这意味着 *ScalingAdjustment* 表示要添加（如果 *ScalingAdjustment* 为正值）或要删除（如果该项为负值）的实例数。在此示例中，*ScalingAdjustment* 为 1；因此，执行策略时，策略将以 1 为增量增加 EC2 实例的数量。

AWS::CloudWatch::Alarm (p. 257) 资源 CPUAlarmHigh 将扩展策略 ScaleUpPolicy 指定为在警报处于 *ALARM* 状态时要执行的操作 (*AlarmActions*)。

```
"ScaleUpPolicy" : {
   "Type" : "AWS::AutoScaling::ScalingPolicy",
   "Properties" : {
      "AdjustmentType" : "ChangeInCapacity",
      "AutoScalingGroupName" : { "Ref" : "asGroup" },
      "Cooldown" : "1",
      "ScalingAdjustment" : "1"
   }
},
"CPUAlarmHigh": {
   "Type": "AWS::CloudWatch::Alarm",
   "Properties": {
      "EvaluationPeriods": "1",
      "Statistic": "Average",
      "Threshold": "10",
     "AlarmDescription": "Alarm if CPU too high or metric disappears indicating
 instance is down",
      "Period": "60",
      "AlarmActions": [ { "Ref": "ScaleUpPolicy" } ],
      "Namespace": "AWS/EC2",
      "Dimensions": [ {
         "Name": "AutoScalingGroupName",
         "Value": { "Ref": "asGroup" }
      } ],
      "ComparisonOperator": "GreaterThanThreshold",
```

```
      "MetricName": "CPUUtilization"
   }
},
```

带通知的 Auto Scaling 组

此示例显示了在指定事件发生时发送 Amazon SNS 通知的 AWS::AutoScaling::AutoScalingGroup (p. 219) 资源。*NotificationConfiguration* 属性指定了 AWS CloudFormation 在其中发送通知的 SNS 主题和将导致 AWS CloudFormation 发送通知的事件。当 *NotificationTypes* 指定的事件发生时，AWS CloudFormation 会发送一条通知至 *TopicARN* 指定的 SNS 主题。在此示例中，AWS CloudFormation 会在 *autoscaling:EC2_INSTANCE_LAUNCH* 和 *autoscaling:EC2_INSTANCE_LAUNCH_ERROR* 事件发生时发送一条通知至 SNS 主题 topic1。

```
"MyAsGroupWithNotification" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Ref" : "azList" },
      "LaunchConfigurationName" : { "Ref" : "myLCOne" },
      "MinSize" : "0",
      "MaxSize" : "2",
      "DesiredCapacity" : "1",
      "NotificationConfiguration" : {
         "TopicARN" : { "Ref" : "topic1" },
         "NotificationTypes" : [
            "autoscaling:EC2_INSTANCE_LAUNCH",
            "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
            "autoscaling:EC2_INSTANCE_TERMINATE",
            "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"
         ]
      }
   }
}
```

Auto Scaling 触发资源

此示例显示了 MyServerGroup AWS::AutoScaling::AutoScalingGroup 资源的 Auto Scaling 触发。

```
"MyTrigger" : {
   "Type" : "AWS::AutoScaling::Trigger",
   "Properties" : {
      "MetricName" : "CPUUtilization",
      "Namespace" : "AWS/EC2",
      "Statistic" : "Average",
      "Period" : "300",
      "UpperBreachScaleIncrement" : "1",
      "LowerBreachScaleIncrement" : "-1",
      "AutoScalingGroupName" : { "Ref" : "MyServerGroup" },
      "BreachDuration" : "600",
      "UpperThreshold" : "90",
      "LowerThreshold" : "75",
      "Dimensions" : [ {
         "Name" : "AutoScalingGroupName",
         "Value" : { "Ref" : "MyServerGroup" }
```

```
      } ]
   }
}
```

带 UpdatePolicy 的 Auto Scaling

此示例显示了如何将 UpdatePolicy (p. 489) 用于 Auto Scaling 组。

```
"ASG1" : {
   "UpdatePolicy" : {
      "AutoScalingRollingUpdate" : {
         "MinInstancesInService" : "1",
         "MaxBatchSize" : "1",
         "PauseTime" : "PT12M5S"
      }
   },
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : { "Ref" : "AWS::Region" } },
      "LaunchConfigurationName" : { "Ref" : "ASLC" },
      "MaxSize" : "3",
      "MinSize" : "1"
   }
}
```

AWS CloudFormation Amazon EC2 模板代码段

Topics

EC2 块设备映射示例

带块设备映射的 EC2 实例

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "AWS CloudFormation Sample Template EC2_In
stance_With_Block_Device_Mapping: Example to show how to attach EBS volumes and
 modify the root device using EC2 block device mappings. **WARNING** This tem
plate creates an Amazon EC2 instance. You will be billed for the AWS resources
 used if you create a stack from this template.",
  "Parameters" : {
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro","m1.small","m1.medi
um","m1.large","m1.xlarge","m3.xlarge","m3.2xlarge","m2.xlarge","m2.2xlarge","m2.4xlarge","c1.me
di
um","c1.xlarge","cc1.4xlarge","cc2.8xlarge","cg1.4xlarge","hi1.4xlarge","hs1.8xlarge"],

      "ConstraintDescription" : "must be a valid EC2 instance type."
    },
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the web server",
      "Type" : "String"
    },
    "SSHFrom": {
      "Description": "Lockdown SSH access to the bastion host (default can be
accessed from anywhere)",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern":
"(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid CIDR range of the form
x.x.x.x/x."
    }
  },
  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "PV64" },
      "m1.small"    : { "Arch" : "PV64" },
      "m1.medium"   : { "Arch" : "PV64" },
      "m1.large"    : { "Arch" : "PV64" },
      "m1.xlarge"   : { "Arch" : "PV64" },
      "m3.xlarge"   : { "Arch" : "PV64" },
      "m3.2xlarge"  : { "Arch" : "PV64" },
      "m2.xlarge"   : { "Arch" : "PV64" },
      "m2.2xlarge"  : { "Arch" : "PV64" },
      "m2.4xlarge"  : { "Arch" : "PV64" },
      "c1.medium"   : { "Arch" : "PV64" },
      "c1.xlarge"   : { "Arch" : "PV64" },
      "cc1.4xlarge" : { "Arch" : "CLU64" },
```

```
      "cc2.8xlarge" : { "Arch" : "CLU64" },
      "cg1.4xlarge" : { "Arch" : "GPU64" },
      "hi1.4xlarge" : { "Arch" : "PV64" },
      "hs1.8xlarge" : { "Arch" : "PV64" }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "PV64" : "ami-1624987f", "CLU64" : "ami-08249861",
     "GPU64" : "ami-02f54a6b" },
      "us-west-2"      : { "PV64" : "ami-2a31bf1a", "CLU64" : "ami-2431bf14",
     "GPU64" : "NOT_YET_SUPPORTED" },
      "us-west-1"      : { "PV64" : "ami-1bf9de5e", "CLU64" : "NOT_YET_SUPPOR
TED", "GPU64" : "NOT_YET_SUPPORTED" },
      "eu-west-1"      : { "PV64" : "ami-c37474b7", "CLU64" : "ami-d97474ad",
     "GPU64" : "ami-1b02026f" },
      "ap-southeast-1" : { "PV64" : "ami-a6a7e7f4", "CLU64" : "NOT_YET_SUPPOR
TED", "GPU64" : "NOT_YET_SUPPORTED" },
      "ap-southeast-2" : { "PV64" : "ami-bd990e87", "CLU64" : "NOT_YET_SUPPOR
TED", "GPU64" : "NOT_YET_SUPPORTED" },
      "ap-northeast-1" : { "PV64" : "ami-4e6cd34f", "CLU64" : "NOT_YET_SUPPOR
TED", "GPU64" : "NOT_YET_SUPPORTED" },
      "sa-east-1"      : { "PV64" : "ami-1e08d103", "CLU64" : "NOT_YET_SUPPOR
TED", "GPU64" : "NOT_YET_SUPPORTED" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                                          { "Fn::FindInMap" : [ "AWSInstance
Type2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] },
        "KeyName" : { "Ref" : "KeyName" },
        "InstanceType" : { "Ref" : "InstanceType" },
        "SecurityGroups" : [{ "Ref" : "Ec2SecurityGroup" }],
        "BlockDeviceMappings" : [
          {
            "DeviceName" : "/dev/sda1",
            "Ebs" : { "VolumeSize" : "50" }
          },{
            "DeviceName" : "/dev/sdm",
            "Ebs" : { "VolumeSize" : "100" }
          }
        ]
      }
    },
    "Ec2SecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "HTTP and SSH access",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22", "ToPort" : "22",
          "CidrIp" : { "Ref" : "SSHFrom" }
        } ]
      }
    }
  },
```

```
  "Outputs" : {
    "Instance" : {
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicDnsName" ] },
      "Description" : "DNS Name of the newly created EC2 instance"
    }
  }
}
```

带短暂驱动的 EC2 实例

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "AWS CloudFormation Sample Template EC2_Instance_With_Ephem
eral_Drives: Example to show how to attach ephemeral drives using EC2 block
device mappings. **WARNING** This template creates an Amazon EC2 instance. You
 will be billed for the AWS resources used if you create a stack from this
template.",
  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the web server",
      "Type" : "String"
    },
    "SSHFrom": {
      "Description": "Lockdown SSH access to the bastion host (default can be
accessed from anywhere)",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern":
"(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription": "must be a valid CIDR range of the form
x.x.x.x/x."
    }
  },
  "Mappings" : {
    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "PV64" : "ami-1624987f" },
      "us-west-2"      : { "PV64" : "ami-2a31bf1a" },
      "us-west-1"      : { "PV64" : "ami-1bf9de5e" },
      "eu-west-1"      : { "PV64" : "ami-c37474b7" },
      "ap-southeast-1" : { "PV64" : "ami-a6a7e7f4" },
      "ap-southeast-2" : { "PV64" : "ami-bd990e87" },
      "ap-northeast-1" : { "PV64" : "ami-4e6cd34f" },
      "sa-east-1"      : { "PV64" : "ami-1e08d103" }
    }
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
```

```
"AWS::Region" }, "PV64" ]},
        "KeyName" : { "Ref" : "KeyName" },
        "InstanceType" : "m1.small",
        "SecurityGroups" : [{ "Ref" : "Ec2SecurityGroup" }],
        "BlockDeviceMappings" : [
          {
            "DeviceName"  : "/dev/sdc",
            "VirtualName" : "ephemeral0"
          }
        ]
      }
    },
    "Ec2SecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "HTTP and SSH access",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22", "ToPort" : "22",
          "CidrIp" : { "Ref" : "SSHFrom" }
        } ]
      }
    }
  },
  "Outputs" : {
    "Instance" : {
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicDnsName" ] },
      "Description" : "DNS Name of the newly created EC2 instance"
    }
  }
}
```

使用 AWS::EC2::EIP 代码段分配 Amazon EC2 弹性 IP

此示例显示了如何分配 Amazon EC2 弹性 IP 地址并使用 AWS::EC2::EIP 资源 (p. 269)将其分配给 Amazon EC2 实例。

```
"MyEIP" : {
 "Type" : "AWS::EC2::EIP",
 "Properties" : {
     "InstanceId" : { "Ref" : "logical name of an AWS::EC2::Instance resource"
 }
 }
}
```

将一个现有弹性 IP 指定给使用 AWS::EC2::EIPAssociation 代码段的 Amazon EC2 实例

此示例显示了如何使用 AWS::EC2::EIPAssociation 资源 (p. 270)将一个现有 Amazon EC2 弹性 IP 地址分配给 Amazon EC2 实例。

```
"IPAssoc" : {
         "Type" : "AWS::EC2::EIPAssociation",
         "Properties" : {
             "InstanceId" : { "Ref" : "logical name of an AWS::EC2::Instance
resource" },
             "EIP" : "existing Elastic IP address"
         }
     }
```

将一个现有 VPC 弹性 IP 分配给使用 AWS::EC2::EIPAssociation 代码段的 Amazon EC2 实例

此示例显示了如何使用 AWS::EC2::EIPAssociation 资源 (p. 270)将一个现有 VPC 弹性 IP 地址分配给 Amazon EC2 实例。

```
"VpcIPAssoc" : {
         "Type" : "AWS::EC2::EIPAssociation",
         "Properties" : {
             "InstanceId" : { "Ref" : "logical name of an AWS::EC2::Instance
resource" },
             "AllocationId" : "existing VPC Elastic IP allocation ID"
         }
     }
```

弹性网络接口 (ENI) 模板代码段

VPC_EC2_Instance_With_ENI

示例模板显示了如何创建具有 1 个弹性网络接口 (ENI) 的实例。它的前提是您已经创建了一个 VPC。**警告** 本模板将创建 Amazon EC2 实例。如果您通过本模板创建堆栈，那么会针对 AWS 资源向您收取相应费用。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "AWS CloudFormation Sample Template VPC_EC2_Instance_With_ENI:
 Sample template showing how to create an instance with 2 network interfaces,
one for Web assess and one for SSH access. The default ENI for the instance is
 used for Web traffic and a second ENI is created for control port traffic. It
 assumes you have already created a VPC. **WARNING** This template creates an
Amazon EC2 instance. You will be billed for the AWS resources used if you create
 a stack from this template.",
  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of and existing EC2 KeyPair to enable SSH access
to the instance",
      "Type" : "String"
    },
    "VpcId" : {
      "Type" : "String",
      "Description" : "VpcId of your existing Virtual Private Cloud (VPC)"
    },
```

```
    "SubnetId" : {
      "Type" : "String",
      "Description" : "SubnetId of an existing subnet in your Virtual Private
Cloud (VPC)"
    },
    "WebServerPort" : {
      "Description" : "TCP/IP port of the web server",
      "Type" : "String",
      "Default" : "80"
    }
  },
  "Mappings" : {
    "RegionMap" : {
      "us-east-1"      : { "AMI" : "ami-aba768c2" },
      "us-west-1"      : { "AMI" : "ami-458fd300" },
      "us-west-2"      : { "AMI" : "ami-fcff72cc" },
      "eu-west-1"      : { "AMI" : "ami-018bb975" },
      "sa-east-1"      : { "AMI" : "ami-a039e6bd" },
      "ap-southeast-1" : { "AMI" : "ami-425a2010" },
      "ap-northeast-1" : { "AMI" : "ami-7871c579" }
    }
  },
  "Resources" : {
    "ControlPortAddress" : {
      "Type" : "AWS::EC2::EIP",
      "Properties" : {
        "Domain" : "vpc"
      }
    },
    "AssociateControlPort" : {
      "Type" : "AWS::EC2::EIPAssociation",
      "Properties" : {
       "AllocationId" : { "Fn::GetAtt" : [ "ControlPortAddress", "AllocationId"
 ]},
        "NetworkInterfaceId" : { "Ref" : "controlXface" }
      }
    },
    "SSHSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "VpcId" : { "Ref" : "VpcId" },
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ { "IpProtocol" : "tcp", "FromPort" : "22",
"ToPort" : "22", "CidrIp" : "0.0.0.0/0" } ]
      }
    },
    "WebSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "VpcId" : { "Ref" : "VpcId" },
        "GroupDescription" : "Enable HTTP access via user defined port",
       "SecurityGroupIngress" : [ { "IpProtocol" : "tcp", "FromPort" : { "Ref"
 : "WebServerPort" }, "ToPort" : { "Ref" : "WebServerPort" }, "CidrIp" :
"0.0.0.0/0" } ]
      }
    },
    "controlXface" : {
      "Type" : "AWS::EC2::NetworkInterface",
```

```
      "Properties" : {
        "SubnetId" : { "Ref" : "SubnetId" },
        "Description" :"Interface for control traffic such as SSH",
        "GroupSet" : [ {"Ref" : "SSHSecurityGroup"} ],
        "SourceDestCheck" : "true",
        "Tags" : [ {"Key" : "Network", "Value" : "Control"}]
      }
    },
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
 }, "AMI" ]},
        "KeyName" : { "Ref" : "KeyName" },
        "NetworkInterfaces" : [ { "NetworkInterfaceId" : {"Ref" : "con
trolXface"}, "DeviceIndex" : "0" } ],
        "Tags" : [ {"Key" : "Role", "Value" : "Test Instance"}],
        "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }}
      }
    }
  },
  "Outputs" : {
    "InstanceId" : {
      "Value" : { "Ref" : "Ec2Instance" },
      "Description" : "Instance Id of newly created instance"
    },
    "ControlPortPublicAddress" : {
      "Value" : { "Ref" : "ControlPortAddress" },
      "Description" : "Control port public IP address of instance for SSH"
    }
  }
}
```

Amazon EC2 实例资源

此代码段所示为简单的 AWS::EC2::Instance 资源。

```
"MyInstance" : {
 "Type" : "AWS::EC2::Instance",
 "Properties" : {
     "AvailabilityZone" : "us-east-1a",
     "ImageId" : "ami-20b65349"
 }
}
```

带 UserData、卷和标签的 Amazon EC2 实例资源

此代码段显示了带有一个 Amazon EC2 卷、一个标签和 Base64 编码的 UserData 的 AWS::EC2::Instance 资源。必须在同一个模板中定义 AWS::EC2::SecurityGroup 资源、AWS::SNS::Topic 资源和 AWS::ETC::Volume 资源。另外，引用 *KeyName*、*AccessKey* 和 *SecretKey* 是必须在模板的 Parameters 部分中定义的参数。

```
"MyInstance" : {
 "Type" : "AWS::EC2::Instance",
 "Properties" : {
```

```
    "KeyName" : { "Ref" : "KeyName" },
    "SecurityGroups" : [ {
        "Ref" : "logical name of AWS::EC2::SecurityGroup resource"
    } ],
    "UserData" : {
        "Fn::Base64" : {
            "Fn::Join" : [ ":", [
                "PORT=80",
                "TOPIC=", {
                    "Ref" : "logical name of an AWS::SNS::Topic resource"
                },
                "ACCESS_KEY=", { "Ref" : "AccessKey" },
                "SECRET_KEY=", { "Ref" : "SecretKey" } ]
            ]
        }
     },
    "InstanceType" : "m1.small",
    "AvailabilityZone" : "us-east-1a",
    "ImageId" : "ami-1e817677",
    "Volumes" : [
       { "VolumeId" : {
            "Ref" : "logical name of AWS::EC2::Volume resource"
       },
       "Device" : "/dev/sdk" }
    ],

    "Tags" : [ {
        "Key" : "Name",
        "Value" : "MyTag"
    } ]
 }
}
```

带 Amazon SimpleDB 域的 Amazon EC2 实例资源

此代码段显示了一个带 UserData 中所指定 Amazon SimpleDB 域的 AWS::EC2::Instance 资源。

```
"MyInstance" : {
 "Type" : "AWS::EC2::Instance",
 "Properties" : {
     "UserData" : {
         "Fn::Base64" : {
             "Fn::Join" : [ "",
                 [ "Domain=", {
                     "Ref" : "logical name of an AWS::SDB::Domain resource"
                 } ]
             ]
         }
      },
     "AvailabilityZone" : "us-east-1a",
     "ImageId" : "ami-20b65349"
 }
}
```

带两个 CIDR 范围入口规则的 Amazon EC2 安全组资源

此代码段显示的是 AWS::EC2::SecurityGroup 资源，该资源用于说明两个授予对特定端口上 TCP 协议之特定 CIDR 范围访问权的两个入口规则。

```
"ServerSecurityGroup" : {
 "Type" : "AWS::EC2::SecurityGroup",
 "Properties" : {
     "GroupDescription" : "allow connections from specified CIDR ranges",
     "SecurityGroupIngress" : [
         {
             "IpProtocol" : "tcp",
             "FromPort" : "80",
             "ToPort" : "80",
             "CidrIp" : "0.0.0.0/0"
         },{
             "IpProtocol" : "tcp",
             "FromPort" : "22",
             "ToPort" : "22",
             "CidrIp" : "192.168.1.1/32"
         }
     ]
 }
}
```

带两个安全组入口规则的 Amazon EC2 安全组资源

此代码段显示的是说明两个安全组入口规则的 AWS::EC2::SecurityGroup 资源。第一个入口规则授予对端口 22 上 TCP 协议的现有安全组 myadminsecuritygroup（归 1234-5678-9012 AWS 账户所有）的访问权限。第二个入口规则授予对端口 80 上 TCP 的安全组 mysecuritygroupcreatedincfn 的访问权限。此入口规则使用 Ref 内部函数引用同一模板中创建的安全组（其逻辑名称为 mysecuritygroupcreatedincfn）。您必须为 `SourceSecurityGroupName` 和 `SourceSecurityGroupOwnerId` 属性声明值。

```
"ServerSecurityGroupBySG" : {
 "Type" : "AWS::EC2::SecurityGroup",
 "Properties" : {
     "GroupDescription" : "allow connections from specified source security
group",
     "SecurityGroupIngress" : [
         {
            "IpProtocol" : "tcp",
            "FromPort" : "22",
            "ToPort" : "22",
            "SourceSecurityGroupName" : "myadminsecuritygroup",
            "SourceSecurityGroupOwnerId" : "123456789012"
         },
         {
            "IpProtocol" : "tcp",
            "FromPort" : "80",
            "ToPort" : "80",
           "SourceSecurityGroupName" : {"Ref" : "mysecuritygroupcreatedincfn"}

         }
     ]
 }
}
```

带负载均衡器入口规则的 Amazon EC2 安全组资源

此代码段显示了一个包含安全组入口规则（用于授予对端口 80 上 TCP 的负载均衡器 myELB 的访问权限）的 AWS::EC2::SecurityGroup 资源。请注意，该规则使用 myELB 资源的 *SourceSecurityGroup.OwnerAlias* 和 *SourceSecurityGroup.GroupName* 属性指定负载均衡器的源安全组。

```
        "myELB" : {
                 "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
                 "Properties" : {
                     "AvailabilityZones" : [ "us-east-1a" ],
                     "Listeners" : [ {
                         "LoadBalancerPort" : "80",
                         "InstancePort" : "80",
                         "Protocol" : "HTTP"
                     } ]
                 }
             },
     "ELBIngressGroup" : {
         "Type" : "AWS::EC2::SecurityGroup",
         "Properties" : {
             "GroupDescription" : "ELB ingress group",
             "SecurityGroupIngress" : [
                 {
                    "IpProtocol" : "tcp",
                    "FromPort" : "80",
                    "ToPort" : "80",
                    "SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["myELB",
"SourceSecurityGroup.OwnerAlias"]},
                    "SourceSecurityGroupName" : {"Fn::GetAtt" : ["myELB",
"SourceSecurityGroup.GroupName"]}
                 }
             ]
         }
```

使用 AWS::EC2::SecurityGroupIngress 创建互相引用的 Amazon EC2 安全组资源

此代码段显示的是两个将相互入口规则添加到 EC2 安全组 SGroup1 和 SGroup2 的 AWS::EC2::SecurityGroupIngress 资源。SGroup1Ingress 资源可使 SGroup2 的入口通过 TCP/IP 端口 80 连至 SGroup1。SGroup2Ingress 资源可使 SGroup1 的入口通过 TCP/IP 端口 80 连至 SGroup2。

Note

如果您使用的是 Amazon VPC，则 *SecurityGroupIngress* 属性必须包含 *VpcId*，并且必须使用 *GroupId* 和 *SourceSecurityGroupId*（而不是 *GroupName* 和 *SourceSecurityGroupName*）。

```
        "SGroup1" : {
         "Type" : "AWS::EC2::SecurityGroup",
         "Properties" : {
             "GroupDescription" : "EC2 Instance access"
         }
     },
```

```
    "SGroup2" : {
        "Type" : "AWS::EC2::SecurityGroup",
        "Properties" : {
            "GroupDescription" : "EC2 Instance access"
        }
    },
    "SGroup1Ingress" : {
        "Type" : "AWS::EC2::SecurityGroupIngress",
        "Properties" : {
            "GroupName" : { "Ref" : "SGroup1" },
            "IpProtocol" : "tcp",
            "ToPort" : "80",
            "FromPort" : "80",
            "SourceSecurityGroupName" : { "Ref" : "SGroup2" }
        }
    },
    "SGroup2Ingress" : {
        "Type" : "AWS::EC2::SecurityGroupIngress",
        "Properties" : {
            "GroupName" : { "Ref" : "SGroup2" },
            "IpProtocol" : "tcp",
            "ToPort" : "80",
            "FromPort" : "80",
            "SourceSecurityGroupName" : { "Ref" : "SGroup1" }
        }
    }
```

Amazon EC2 卷资源

此代码段显示了 DeletionPolicy 属性已设置为 Snapshot 的简单 Amazon EC2 卷资源。在 DeletionPolicy 设置为 Snapshot 的情况下，AWS CloudFormation 将在堆栈删除期间于删除此卷之前创建此卷的快照。确保您为 SnapShotId 或 Size（而不是两者）指定了值。删除您不需要的值。

```
"MyEBSVolume" : {
 "Type" : "AWS::EC2::Volume",
 "Properties" : {
     "Size" : "specify a size if no SnapShotId",
     "SnapshotId" : "specify a SnapShotId if no Size",
     "AvailabilityZone" : { "Ref" : "AvailabilityZone" }
 },
 "DeletionPolicy" : "Snapshot"
}
```

Amazon EC2 VolumeAttachment 资源

此代码段显示了以下资源：在美国东部（弗吉尼亚北部）区域使用 Amazon Linux AMI 的 Amazon EC2 实例、允许 SSH 访问 IP 地址的 EC2 安全组、容量为 100 GB 且与 EC2 实例在同一个可用区的新 Amazon EBS 卷，以及用于将新卷附加到 EC2 实例的卷附加操作。

```
"Resources" : {
 "Ec2Instance" : {
   "Type" : "AWS::EC2::Instance",
   "Properties" : {
     "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
     "ImageId" : "ami-76f0061f"
```

```
    }
  },

  "InstanceSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Enable SSH access via port 22",
      "SecurityGroupIngress" : [ {
        "IpProtocol" : "tcp",
        "FromPort" : "22",
        "ToPort" : "22",
        "CidrIp" : "0.0.0.0/0"
      } ]
    }
  },

  "NewVolume" : {
    "Type" : "AWS::EC2::Volume",
    "Properties" : {
      "Size" : "100",
      "AvailabilityZone" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone"
  ]},
    }
  },

  "MountPoint" : {
    "Type" : "AWS::EC2::VolumeAttachment",
    "Properties" : {
      "InstanceId" : { "Ref" : "Ec2Instance" },
      "VolumeId"  : { "Ref" : "NewVolume" },
      "Device" : "/dev/sdh"
    }
  }
}
```

AWS Elastic Beanstalk 代码段

AWS Elastic Beanstalk 可让您迅速地在 AWS 中部署和管理应用程序，而无需为运行这些应用程序的基础设施操心。以下示例模板可以帮助您在 AWS CloudFormation 模板中描述 AWS Elastic Beanstalk 资源。

AWS Elastic Beanstalk示例 PHP

以下示例模板部署一个存储在 Amazon S3 存储桶中的示例 PHP Web 应用程序。AWS Elastic Beanstalk 环境是运行 PHP 5.3 的 64 位 Amazon Linux。该环境也是负载均衡的 Auto Scaling 环境，最少包含两个、最多包含六个 Amazon EC2 实例。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "sampleApplication": {
      "Type": "AWS::ElasticBeanstalk::Application",
      "Properties": {
        "Description": "AWS Elastic Beanstalk Sample Application"
      }
```

```
    },
    "sampleApplicationVersion": {
      "Type": "AWS::ElasticBeanstalk::ApplicationVersion",
      "Properties": {
        "ApplicationName": { "Ref": "sampleApplication" },
        "Description": "AWS ElasticBeanstalk Sample Application Version",
        "SourceBundle": {
          "S3Bucket": { "Fn::Join": [ "-", [ "elasticbeanstalk-samples", {
"Ref": "AWS::Region" } ] ] },
          "S3Key": "php-sample.zip"
        }
      }
    },
    "sampleConfigurationTemplate": {
      "Type": "AWS::ElasticBeanstalk::ConfigurationTemplate",
      "Properties": {
        "ApplicationName": { "Ref": "sampleApplication" },
        "Description": "AWS ElasticBeanstalk Sample Configuration Template",
        "OptionSettings": [
          {
            "Namespace": "aws:autoscaling:asg",
            "OptionName": "MinSize",
            "Value": "2"
          },
          {
            "Namespace": "aws:autoscaling:asg",
            "OptionName": "MaxSize",
            "Value": "6"
          },
          {
            "Namespace": "aws:elasticbeanstalk:environment",
            "OptionName": "EnvironmentType",
            "Value": "LoadBalanced"
          }
        ],
        "SolutionStackName": "64bit Amazon Linux running PHP 5.3"
      }
    },
    "sampleEnvironment": {
      "Type": "AWS::ElasticBeanstalk::Environment",
      "Properties": {
        "ApplicationName": { "Ref": "sampleApplication" },
        "Description": "AWS ElasticBeanstalk Sample Environment",
        "TemplateName": { "Ref": "sampleConfigurationTemplate" },
        "VersionLabel": { "Ref": "sampleApplicationVersion" }
      }
    }
  }
}
```

Elastic Load Balancing 代码段

Topics

Elastic Load Balancing 负载均衡器资源

此示例显示带一个侦听器但不带实例的 Elastic Load Balancing 负载均衡器。

```
"MyLoadBalancer" : {
    "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties" : {
        "AvailabilityZones" : [ "us-east-1a" ],
        "Listeners" : [ {
            "LoadBalancerPort" : "80",
            "InstancePort" : "80",
            "Protocol" : "HTTP"
        } ]
    }
}
```

带运行状况检查功能的 Elastic Load Balancing 负载均衡器资源

此示例显示带两个 Amazon EC2 实例、一个侦听器和运行状态检查功能的 Elastic Load Balancing 负载均衡器。

```
"MyLoadBalancer" : {
    "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
    "Properties" : {
        "AvailabilityZones" : [ "us-east-1a" ],
        "Instances" : [
            { "Ref" : "logical name of AWS::EC2::Instance resource 1" },
            { "Ref" : "logical name of AWS::EC2::Instance resource 2" }
        ],
        "Listeners" : [ {
            "LoadBalancerPort" : "80",
            "InstancePort" : "80",
            "Protocol" : "HTTP"
        } ],

        "HealthCheck" : {
            "Target" : "HTTP:80/",
            "HealthyThreshold" : "3",
            "UnhealthyThreshold" : "5",
            "Interval" : "30",
            "Timeout" : "5"
        }
    }
}
```

Identity and Access Management (IAM) 模板代码段

Abstract

通过 AWS CloudFormation，将这些示例模板代码段用于 Amazon Identity and Access Management 资源。

此部分包含 AWS Identity and Access Management 模板代码段。

Topics

- 声明 IAM 访问密钥资源 (p. 148)
- 声明 IAM 组资源 (p. 149)
- 添加用户到组中 (p. 150)
- 声明 IAM 策略 (p. 150)
- 声明 Amazon S3 存储桶策略 (p. 150)
- 声明 Amazon SNS 主题策略 (p. 151)
- 声明 Amazon SQS 策略 (p. 151)
- IAM 角色模板示例 (p. 152)

Important

使用包含 IAM 资源的模板创建或更新堆栈时，您必须确认 IAM 功能的使用。有关使用模板中的 IAM 资源的更多信息，请参阅使用 AWS Identity and Access Management 控制访问 (p. 59)。

声明 IAM 用户资源

此代码段显示如何声明 AWS::IAM::User (p. 355) 资源以创建 IAM 用户。用路径 "/" 和密码为 myP@ssW0rd 的登录配置文件声明用户。

名为 `giveaccesstoqueueonly` 的策略文档授予用户在 SQS 队列资源 myqueue 上执行所有 SQS 操作的权限，并拒绝对所有其他 SQS 队列资源进行访问。Fn::GetAtt (p. 502) 函数获取 AWS::SQS::Queue (p. 414) 资源 myqueue 的 Arn 属性。

将为用户添加名为 `giveaccesstotopiconly` 的策略文档以授予用户在 SNS 主题资源 mytopic 上执行所有 SNS 操作的权限，并拒绝对所有其他 SNS 资源进行访问。Ref 函数 (p. 508)获取 AWS::SNS::Topic (p. 411) 资源 mytopic 的 ARN。

```
"myuser" : {
   "Type" : "AWS::IAM::User",
   "Properties" : {
      "Path" : "/",
      "LoginProfile" : {
         "Password" : "myP@ssW0rd"
      },
      "Policies" : [ {
         "PolicyName" : "giveaccesstoqueueonly",
         "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
               "Effect" : "Allow",
               "Action" : [ "sqs:*" ],
               "Resource" : [ {
                  "Fn::GetAtt" : [ "myqueue", "Arn" ]
               } ]
            }, {
               "Effect" : "Deny",
               "Action" : [ "sqs:*" ],
               "NotResource" : [ {
                  "Fn::GetAtt" : [ "myqueue", "Arn" ]
               } ]
            }
         ] }
      }, {
```

```
         "PolicyName" : "giveaccesstotopiconly",
         "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
               "Effect" : "Allow",
               "Action" : [ "sns:*" ],
               "Resource" : [ { "Ref" : "mytopic" } ]
            }, {
               "Effect" : "Deny",
               "Action" : [ "sns:*" ],
               "NotResource" : [ { "Ref" : "mytopic" } ]
            } ]
         }
      } ]
   }
}
```

声明 IAM 访问密钥资源

此代码段显示了 AWS::IAM::AccessKey (p. 343) 资源。myaccesskey 资源创建访问密钥并将其分配给在模板中声明为 AWS::IAM::User (p. 355) 资源的 IAM 用户。

```
"myaccesskey" : {
   "Type" : "AWS::IAM::AccessKey",
   "Properties" : {
      "UserName" : { "Ref" : "myuser" }
   }
}
```

您可以使用 Fn::GetAtt (p. 502) 函数获取 AWS::IAM::AccessKey 资源的私有密钥。您只能在创建 AWS 访问密钥时获取其私有密钥。检索私有密钥的一种方法是将其放入输出值中。您可以使用 Ref 函数获取访问密钥。以下输出值声明获取 myaccesskey 的访问密钥和私有密钥。

```
"AccessKeyformyaccesskey" : {
   "Value" : { "Ref" : "myaccesskey" }
},
"SecretKeyformyaccesskey" : {
   "Value" : {
      "Fn::GetAtt" : [ "myaccesskey", "SecretAccessKey" ]
   }
}
```

您还可以将 AWS 访问密钥和私有密钥传输给模板中定义的 EC2 实例或 Auto Scaling 组中。以下 AWS::EC2::Instance (p. 272) 声明使用 UserData 属性传递 myaccesskey 资源的访问密钥和私有密钥。

```
"myinstance" : {
   "Type" : "AWS::EC2::Instance",
   "Properties" : {
      "AvailabilityZone" : "us-east-1a",
      "ImageId" : "ami-20b65349",
      "UserData" : {
```

```
         "Fn::Base64" : {
            "Fn::Join" : [
               "", [
                  "ACCESS_KEY=", {
                     "Ref" : "myaccesskey"
                  },
                  "&",
                  "SECRET_KEY=",
                  {
                     "Fn::GetAtt" : [
                        "myaccesskey",
                        "SecretAccessKey"
                     ]
                  }
               ]
            ]
         }
      }
   }
}
```

声明 IAM 组资源

此代码段显示了 AWS::IAM::Group (p. 345) 资源。该组有一个路径 ("/myapplication/")。组中会添加一个名为 myapppolicy 的策略文件，以允许组用户执行 SQS 队列资源 myqueue 上所有的 SQS 操作并拒绝对除 myqueue 之外的所有其它 SQS 资源的访问。

要分配一个策略给资源，IAM 需要该资源的亚马逊资源名称 (ARN)。在此代码段中，Fn::GetAtt (p. 502) 函数获取 AWS::SQS::Queue (p. 414) 资源队列的 ARN。

```
"mygroup" : {
   "Type" : "AWS::IAM::Group",
   "Properties" : {
      "Path" : "/myapplication/",
      "Policies" : [ {
         "PolicyName" : "myapppolicy",
         "PolicyDocument" : {
            "Version": "2012-10-17",
            "Statement" : [ {
               "Effect" : "Allow",
               "Action" : [ "sqs:*" ],
               "Resource" : [ {
                  "Fn::GetAtt" : [ "myqueue", "Arn" ]
               } ]
            },
            {
               "Effect" : "Deny",
               "Action" : [ "sqs:*" ],
               "NotResource" : [ { "Fn::GetAtt" : [ "myqueue", "Arn" ] } ]
            }
         ] }
      } ]
   }
}
```

添加用户到组中

AWS::IAM::UserToGroupAddition (p. 356) 资源将用户添加到组。在以下代码段中，addUserToGroup 资源会将以下用户添加到名为 myexistinggroup2 的现有组：现有用户 existinguser1 和在模板中声明为 AWS::IAM::User (p. 355) 资源的用户 myuser。

```
"addUserToGroup" : {
   "Type" : "AWS::IAM::UserToGroupAddition",
   "Properties" : {
      "GroupName" : "myexistinggroup2",
      "Users" : [ "existinguser1", { "Ref" : "myuser" } ]
   }
}
```

声明 IAM 策略

此代码段显示如何创建策略并使用名为 mypolicy 的 AWS::IAM::Policy (p. 348) 资源将该策略应用于多个组。mypolicy 资源中包含 PolicyDocument 属性，该属性允许 ARN arn:aws:s3:::myAWSBucket 所表示的 S3 存储桶中的对象上的 GetObject、PutObject 和 PutObjectAcl 操作。mypolicy 资源将策略应用于名为 myexistinggroup1 的现有组以及在模板中声明为 AWS::IAM::Group (p. 345) 资源的组 mygroup。此示例显示如何将策略应用于使用 Groups 属性的策略；但是您或者可以使用 Users 属性将策略文件添加到用户列表中。

```
"mypolicy" : {
   "Type" : "AWS::IAM::Policy",
   "Properties" : {
      "PolicyName" : "mygrouppolicy",
      "PolicyDocument" : {
         "Version": "2012-10-17",
         "Statement" : [ {
            "Effect" : "Allow",
            "Action" : [
               "s3:GetObject" , "s3:PutObject" , "s3:PutObjectAcl" ],
            "Resource" : "arn:aws:s3:::myAWSBucket/*"
         } ]
      },
      "Groups" : [ "myexistinggroup1", { "Ref" : "mygroup" } ]
   }
}
```

声明 Amazon S3 存储桶策略

此代码段显示如何创建策略并使用 AWS::S3::BucketPolicy (p. 409) 资源将该策略应用于 Amazon S3 存储桶。mybucketpolicy 资源会声明一个策略文件，该策略文件允许组 mygroup 执行应用该策略的 S3 存储桶中所有对象的 GetObject 操作。在此代码段中，Fn::GetAtt (p. 502) 函数获取 mygroup 资源的 ARN。mybucketpolicy 资源将策略应用于 AWS::S3::Bucket (p. 402) 资源 mybucket。Ref 函数 (p. 508)获取 mybucket 资源的存储桶名称。

```
"mybucketpolicy" : {
   "Type" : "AWS::S3::BucketPolicy",
   "Properties" : {
```

```
      "PolicyDocument" : {
         "Id" : "MyPolicy",
         "Version": "2012-10-17",
         "Statement" : [ {
            "Sid" : "ReadAccess",
            "Action" : [ "s3:GetObject" ],
            "Effect" : "Allow",
            "Resource" : { "Fn::Join" : [
                  "", [ "arn:aws:s3:::", { "Ref" : "mybucket" } , "/*" ]
               ] },
            "Principal" : {
               "AWS" : { "Fn::GetAtt" : [ "mygroup", "Arn" ] }
            }
         } ]
      },
      "Bucket" : { "Ref" : "mybucket" }
      }
   }
}
```

声明 Amazon SNS 主题策略

此代码段显示如何创建策略并使用 AWS::SNS::TopicPolicy (p. 413) 资源将该策略应用于 Amazon SNS 主题。mysnspolicy 资源包含 PolicyDocument 属性，该属性允许 AWS::IAM::User (p. 355) 资源 myuser 在 AWS::SNS::Topic (p. 411) 资源 mytopic 上执行发布操作。在此代码段中，Fn::GetAtt (p. 502) 函数获取 myuser 资源的 ARN，而 Ref (p. 508) 函数获取 mytopic 资源的 ARN。

```
"mysnspolicy" : {
   "Type" : "AWS::SNS::TopicPolicy",
   "Properties" : {
      "PolicyDocument" :  {
         "Id" : "MyTopicPolicy",
         "Version" : "2012-10-17",
         "Statement" : [ {
            "Sid" : "My-statement-id",
            "Effect" : "Allow",
            "Principal" : {
               "AWS" : { "Fn::GetAtt" : [ "myuser", "Arn" ] }
            },
            "Action" : "sns:Publish",
            "Resource" : "*"
         } ]
      },
      "Topics" : [ { "Ref" : "mytopic" } ]
   }
}
```

声明 Amazon SQS 策略

此代码段显示如何创建策略并使用 AWS::SQS::QueuePolicy (p. 418) 资源将该策略应用于 Amazon SQS 队列。PolicyDocument 属性允许现有用户 myapp（由其 ARN 指定）在现有队列（由其 URL 指定）和 AWS::SQS::Queue (p. 414) 资源 myqueue 上执行发送消息操作。Ref (p. 508) 函数获取 myqueue 资源的 URL。

```
"mysqspolicy" : {
   "Type" : "AWS::SQS::QueuePolicy",
   "Properties" : {
      "PolicyDocument" : {
         "Id" : "MyQueuePolicy",
         "Version" : "2012-10-17",
         "Statement" : [ {
            "Sid" : "Allow-User-SendMessage",
            "Effect" : "Allow",
            "Principal" : {
               "AWS" : "arn:aws:iam::123456789012:user/myapp"
            },
            "Action" : [ "sqs:SendMessage" ],
            "Resource" : "*"
         } ]
      },
      "Queues" : [
         "https://sqs.us-east-1.amazonaws.com/123456789012/myexistingqueue",
         { "Ref" : "myqueue" }
      ]
   }
}
```

IAM 角色模板示例

本部分提供 EC2 实例之 IAM 角色的 CloudFormation 模板示例。

有关 IAM 角色的更多信息，请参阅 AWS Identity and Access Management User Guide 中的*使用角色*。

带 EC2 的 IAM 角色

Example 带外部策略和连接到 EC2 实例的实例配置文件的 IAM 角色

在此示例中，实例配置文件由 EC2 实例的 IamInstanceProfile 属性引用。实例策略和角色策略都引用 AWS::IAM::Role (p. 351)。

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "myEC2Instance": {
         "Type": "AWS::EC2::Instance",
         "Version": "2009-05-15",
         "Properties": {
            "ImageId": "ami-205fba49",
            "InstanceType": "m1.small",
            "Monitoring": "true",
            "DisableApiTermination": "false",
            "IamInstanceProfile": {
               "Ref": "RootInstanceProfile"
            }
         }
      },
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "RolePolicies": {
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "root",
            "PolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "*",
                  "Resource": "*"
               } ]
            },
            "Roles": [ { "Ref": "RootRole" } ]
         }
      },
      "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ { "Ref": "RootRole" } ]
```

```
            }
        }
    }
}
```

带 AutoScaling 组的 IAM 角色

Example 带外部策略和连接到 AutoScaling 组的实例配置文件的 IAM 角色

在此示例中，实例配置文件由 AutoScaling 组启动配置的 IamInstanceProfile 属性引用。

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "myLCOne": {
         "Type": "AWS::AutoScaling::LaunchConfiguration",
         "Version": "2009-05-15",
         "Properties": {
            "ImageId": "ami-205fba49",
            "InstanceType": "m1.small",
            "InstanceMonitoring": "true",
            "IamInstanceProfile": { "Ref": "RootInstanceProfile" }
         }
      },
      "myASGrpOne": {
         "Type": "AWS::AutoScaling::AutoScalingGroup",
         "Version": "2009-05-15",
         "Properties": {
            "AvailabilityZones": [ "us-east-1a" ],
            "LaunchConfigurationName": { "Ref": "myLCOne" },
            "MinSize": "0",
            "MaxSize": "0",
            "HealthCheckType": "EC2",
            "HealthCheckGracePeriod": "120"
         }
      },
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "RolePolicies": {
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "root",
            "PolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "*",
                  "Resource": "*"
               } ]
            },
```

```
            "Roles": [ { "Ref": "RootRole" } ]
         }
      },
      "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ { "Ref": "RootRole" } ]
         }
      }
   }
}
```

AWS OpsWorks 代码段

AWS OpsWorks 是应用程序管理服务，可简化各种任务，如软件配置、应用程序部署、扩展和监控。AWS CloudFormation 是资源管理服务，可用于管理 AWS OpsWorks 资源，如 AWS OpsWorks 堆栈、层、应用程序和实例。

AWS OpsWorks 示例 PHP 应用程序

以下示例模板部署一个存储在公用 Git 存储库中的示例 AWS OpsWorks PHP Web 应用程序。该 AWS OpsWorks 堆栈包含两个应用程序服务器及一个负载均衡器，该负载均衡器用于在各服务器间均匀分布传入流量。该 AWS OpsWorks 堆栈还包含一个后端 MySQL 数据库服务器以存储数据。有关示例 AWS OpsWorks 应用程序的更多信息，请参阅演练：通过创建应用程序服务器堆栈了解 AWS AWS OpsWorks 基础知识（在 *AWS OpsWorks 用户指南* 中）。

Note

`ServiceRoleArn` 和 `DefaultInstanceProfileArn` 属性引用首次使用 AWS OpsWorks 之后创建的 IAM 角色。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "ServiceRole": {
      "Default": "aws-opsworks-service-role",
      "Description": "The OpsWorks service role",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "64",
      "AllowedPattern": "[a-zA-Z][a-zA-Z0-9-]*",
     "ConstraintDescription": "must begin with a letter and contain only alpha
numeric characters."
    },
    "InstanceRole": {
      "Default": "aws-opsworks-ec2-role",
      "Description": "The OpsWorks instance role",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "64",
      "AllowedPattern": "[a-zA-Z][a-zA-Z0-9-]*",
     "ConstraintDescription": "must begin with a letter and contain only alpha
numeric characters."
    },
```

```
    "AppName": {
      "Default": "myapp",
      "Description": "The app name",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "64",
      "AllowedPattern": "[a-zA-Z][a-zA-Z0-9]*",
     "ConstraintDescription": "must begin with a letter and contain only alpha
numeric characters."
    },
    "MysqlRootPassword" : {
      "Description" : "MysqlRootPassword",
      "NoEcho" : "true",
      "Type" : "String"
    }
  },
  "Resources": {
    "myStack": {
      "Type": "AWS::OpsWorks::Stack",
      "Properties": {
        "Name": {
          "Ref": "AWS::StackName"
        },
        "ServiceRoleArn": {
          "Fn::Join": [
            "", ["arn:aws:iam::", {"Ref": "AWS::AccountId"},
                  ":role/", {"Ref": "ServiceRole"}]
          ]
        },
        "DefaultInstanceProfileArn": {
          "Fn::Join": [
            "", ["arn:aws:iam::", {"Ref": "AWS::AccountId"},
              ":instance-profile/", {"Ref": "InstanceRole"}]
          ]
        },
        "UseCustomCookbooks": "true",
        "CustomCookbooksSource": {
          "Type": "git",
          "Url": "git://github.com/amazonwebservices/opsworks-example-cook
books.git"
        }
      }
    },
    "myLayer": {
      "Type": "AWS::OpsWorks::Layer",
      "DependsOn": "myApp",
      "Properties": {
        "StackId": {"Ref": "myStack"},
        "Type": "php-app",
     "Shortname" : "php-app",
        "EnableAutoHealing" : "true",
        "AutoAssignElasticIps" : "false",
        "AutoAssignPublicIps" : "true",
        "Name": "MyPHPApp",
        "CustomRecipes" : {
          "Configure" : ["phpapp::appsetup"]
        }
      }
```

```
    },
    "DBLayer" : {
      "Type" : "AWS::OpsWorks::Layer",
      "DependsOn": "myApp",
      "Properties" : {
        "StackId" : {"Ref":"myStack"},
        "Type" : "db-master",
     "Shortname" : "db-layer",
        "EnableAutoHealing" : "true",
        "AutoAssignElasticIps" : "false",
        "AutoAssignPublicIps" : "true",
        "Name" : "MyMySQL",
        "CustomRecipes" : {
          "Setup" : ["phpapp::dbsetup"]
        },
        "Attributes" : {
          "MysqlRootPassword" : {"Ref":"MysqlRootPassword"},
          "MysqlRootPasswordUbiquitous": "true"
        },
        "VolumeConfigurations":[{"MountPoint":"/vol/mysql","NumberOf
Disks":1,"Size":10}]
      }
    },
    "ELBAttachment" : {
      "Type" : "AWS::OpsWorks::ElasticLoadBalancerAttachment",
      "Properties" : {
        "ElasticLoadBalancerName" : { "Ref" : "ELB" },
        "LayerId" : { "Ref" : "myLayer" }
      }
    },
    "ELB" : {
      "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties": {
        "AvailabilityZones": { "Fn::GetAZs" : "" } ,
        "Listeners": [{
          "LoadBalancerPort": "80",
          "InstancePort": "80",
          "Protocol": "HTTP",
          "InstanceProtocol": "HTTP"
        }],
        "HealthCheck": {
          "Target": "HTTP:80/",
          "HealthyThreshold": "2",
          "UnhealthyThreshold": "10",
          "Interval": "30",
          "Timeout": "5"
        }
      }
    },
    "myAppInstance1": {
      "Type": "AWS::OpsWorks::Instance",
      "Properties": {
        "StackId": {"Ref": "myStack"},
        "LayerIds": [{"Ref": "myLayer"}],
        "InstanceType": "m1.small"
      }
    },
    "myAppInstance2": {
```

```
      "Type": "AWS::OpsWorks::Instance",
      "Properties": {
        "StackId": {"Ref": "myStack"},
        "LayerIds": [{"Ref": "myLayer"}],
        "InstanceType": "m1.small"
      }
    },
    "myDBInstance": {
      "Type": "AWS::OpsWorks::Instance",
      "Properties": {
        "StackId": {"Ref": "myStack"},
        "LayerIds": [{"Ref": "DBLayer"}],
        "InstanceType": "m1.small"
      }
    },
    "myApp" : {
      "Type" : "AWS::OpsWorks::App",
      "Properties" : {
        "StackId" : {"Ref":"myStack"},
        "Type" : "php",
        "Name" : {"Ref": "AppName"},
        "AppSource" : {
          "Type" : "git",
         "Url" : "git://github.com/amazonwebservices/opsworks-demo-php-simple-
app.git",
          "Revision" : "version2"
        },
        "Attributes" : {
          "DocumentRoot" : "web"
        }
      }
    }
  }
}
```

Amazon Redshift 代码段

Amazon Redshift 是一种完全托管的 PB 级云中数据仓库服务。您可以使用 AWS CloudFormation 配置和管理 Amazon Redshift 集群。

Amazon Redshift 集群

以下示例模板根据在创建堆栈时指定的参数值创建 Amazon Redshift 集群。与 Amazon Redshift 集群关联的集群参数组会启用用户活动日志记录。该模板还在模板中定义的 Amazon VPC 中启动 Amazon Redshift 集群。VPC 包含一个 Internet 网关，以便您可以从 Internet 访问 Amazon Redshift 集群。但是，还必须启用集群与 Internet 网关之间的通信，这通过路由表条目实现。

Note

该模板包含 `IsMultiNodeCluster` 条件，以便仅当 `ClusterType` 参数值设置为 `multi-node` 时才声明 `NumberOfNodes` 参数。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters" : {
```

```
    "DatabaseName" : {
      "Description" : "The name of the first database to be created when the
cluster is created",
      "Type" : "String",
      "Default" : "dev",
      "AllowedPattern" : "([a-z]|[0-9])+"
    },
    "ClusterType" : {
      "Description" : "The type of cluster",
      "Type" : "String",
      "Default" : "single-node",
      "AllowedValues" : [ "single-node", "multi-node" ]
    },
    "NumberOfNodes" : {
      "Description" : "The number of compute nodes in the cluster. For multi-
node clusters, the NumberOfNodes parameter must be greater than 1",
      "Type" : "Number",
      "Default" : "1"
    },
    "NodeType" : {
      "Description" : "The type of node to be provisioned",
      "Type" : "String",
      "Default" : "dw1.xlarge",
     "AllowedValues" : [ "dw1.xlarge", "dw1.8xlarge", "dw2.large", "dw2.8xlarge"
 ]
    },
    "MasterUsername" : {
      "Description" : "The user name that is associated with the master user
account for the cluster that is being created",
      "Type" : "String",
      "Default" : "defaultuser",
      "AllowedPattern" : "([a-z])([a-z]|[0-9])*"
    },
    "MasterUserPassword" :  {
      "Description" : "The password that is associated with the master user
account for the cluster that is being created.",
      "Type" : "String",
      "NoEcho" : "true"
    },
    "InboundTraffic" : {
      "Description" : "Allow inbound traffic to the cluster from this CIDR
range.",
      "Type" : "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default" : "0.0.0.0/0",
      "AllowedPattern" :
"(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
      "ConstraintDescription" : "must be a valid CIDR range of the form
x.x.x.x/x."
    },
    "PortNumber" : {
      "Description" : "The port number on which the cluster accepts incoming
connections.",
      "Type" : "Number",
      "Default" : "5439"
    }
  },
```

```
  "Conditions" : {
    "IsMultiNodeCluster" : {
      "Fn::Equals" : [{ "Ref" : "ClusterType" }, "multi-node" ]
    }
  },
  "Resources" : {
    "RedshiftCluster" : {
      "Type" : "AWS::Redshift::Cluster",
      "DependsOn" : "AttachGateway",
      "Properties" : {
        "ClusterType" : { "Ref" : "ClusterType" },
        "NumberOfNodes" : { "Fn::If" : [ "IsMultiNodeCluster",  { "Ref" :
"NumberOfNodes" }, { "Ref" : "AWS::NoValue" }]},
        "NodeType" : { "Ref" : "NodeType" },
        "DBName" : { "Ref" : "DatabaseName" },
        "MasterUsername" : { "Ref" : "MasterUsername" },
        "MasterUserPassword" : { "Ref" : "MasterUserPassword" },

       "ClusterParameterGroupName" : { "Ref" : "RedshiftClusterParameterGroup"
 },
        "VpcSecurityGroupIds" : [ { "Ref" : "SecurityGroup" } ],
        "ClusterSubnetGroupName" : { "Ref" : "RedshiftClusterSubnetGroup" },
        "PubliclyAccessible" : "true",
        "Port" : { "Ref" : "PortNumber" }
      }
    },
    "RedshiftClusterParameterGroup" : {
      "Type" : "AWS::Redshift::ClusterParameterGroup",
      "Properties" : {
        "Description" : "Cluster parameter group",
        "ParameterGroupFamily" : "redshift-1.0",
        "Parameters" : [{
          "ParameterName" : "enable_user_activity_logging",
          "ParameterValue" : "true"
        }]
      }
    },
    "RedshiftClusterSubnetGroup" : {
      "Type" : "AWS::Redshift::ClusterSubnetGroup",
      "Properties" : {
        "Description" : "Cluster subnet group",
        "SubnetIds" : [ { "Ref" : "PublicSubnet" } ]
      }
    },
    "VPC" : {
      "Type" : "AWS::EC2::VPC",
      "Properties" : {
        "CidrBlock" : "10.0.0.0/16"
      }
    },
    "PublicSubnet" : {
      "Type" : "AWS::EC2::Subnet",
      "Properties" : {
        "CidrBlock" : "10.0.0.0/24",
        "VpcId" : { "Ref" : "VPC" }
      }
    },
    "SecurityGroup" : {
```

```
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Security group",
        "SecurityGroupIngress" : [ {
          "CidrIp" : { "Ref": "InboundTraffic" },
          "FromPort" : { "Ref" : "PortNumber" },
          "ToPort" : { "Ref" : "PortNumber" },
          "IpProtocol" : "tcp"
        } ],
        "VpcId" : { "Ref" : "VPC" }
      }
    },
    "myInternetGateway" : {
      "Type" : "AWS::EC2::InternetGateway"
    },
    "AttachGateway" : {
      "Type" : "AWS::EC2::VPCGatewayAttachment",
      "Properties" : {
        "VpcId" : { "Ref" : "VPC" },
        "InternetGatewayId" : { "Ref" : "myInternetGateway" }
      }
    },
    "PublicRouteTable" : {
      "Type" : "AWS::EC2::RouteTable",
      "Properties" : {
        "VpcId" : {
          "Ref" : "VPC"
        }
      }
    },
    "PublicRoute" : {
      "Type" : "AWS::EC2::Route",
      "DependsOn" : "AttachGateway",
      "Properties"  : {
        "RouteTableId" : {
          "Ref" : "PublicRouteTable"
        },
        "DestinationCidrBlock" : "0.0.0.0/0",
        "GatewayId" : {
          "Ref" : "myInternetGateway"
        }
      }
    },
    "PublicSubnetRouteTableAssociation" : {
      "Type" : "AWS::EC2::SubnetRouteTableAssociation",
      "Properties" : {
        "SubnetId" : {
          "Ref" : "PublicSubnet"
        },
        "RouteTableId" : {
          "Ref" : "PublicRouteTable"
        }
      }
    }
  },
  "Outputs" : {
    "ClusterEndpoint" : {
      "Description" : "Cluster endpoint",
```

```
      "Value" : { "Fn::Join" : [ ":", [ { "Fn::GetAtt" : [ "RedshiftCluster",
"Endpoint.Address" ] }, { "Fn::GetAtt" : [ "RedshiftCluster", "Endpoint.Port"
] } ] ] }
    },
    "ClusterName" : {
      "Description" : "Name of cluster",
      "Value" : { "Ref" : "RedshiftCluster" }
    },
    "ParameterGroupName" : {
      "Description" : "Name of parameter group",
      "Value" : { "Ref" : "RedshiftClusterParameterGroup" }
    },
    "RedshiftClusterSubnetGroupName" : {
      "Description" : "Name of cluster subnet group",
      "Value" : { "Ref" : "RedshiftClusterSubnetGroup" }
    },
    "RedshiftClusterSecurityGroupName" : {
      "Description" : "Name of cluster security group",
      "Value" : { "Ref" : "SecurityGroup" }
    }
  }
}
```

另请参阅

AWS::Redshift::Cluster (p. 371)

Amazon RDS 模板代码段

Topics

- Amazon RDS 数据库实例资源 (p. 165)
- Amazon RDS Oracle Database 数据库实例资源 (p. 166)
- 适用于 CIDR 范围的 Amazon RDS DBSecurityGroup 资源 (p. 166)
- 带 Amazon EC2 安全组的 Amazon RDS DBSecurityGroup (p. 166)
- 多 VPC 安全组 (p. 167)

Amazon RDS 数据库实例资源

此示例显示 Amazon RDS 数据库实例资源。因为没有指定可选 EngineVersion 属性，所以会将默认引擎版本用于此 DB 实例。有关默认引擎版本和其他默认设置的详细信息，请参阅 CreateDBInstance。DBSecurityGroups 属性授权对名为 MyDbSecurityByEC2SecurityGroup 和 MyDbSecurityByCIDRIPGroup 的 AWS::RDS::DBSecurityGroup 资源的网络访问。有关详细信息，请参阅 AWS::RDS::DBInstance (p. 381)。DB 实例资源还有一个设置为 Snapshot 的 DeletionPolicy 属性。在 DeletionPolicy 设置为 Snapshot 的情况下，AWS CloudFormation 将在堆栈删除期间于删除此数据库实例之前创建此数据库实例的快照。

```
"MyDB" : {
 "Type" : "AWS::RDS::DBInstance",
 "Properties" : {
     "DBSecurityGroups" : [
       {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" : "MyDbSecurityByCID
RIPGroup"} ],
     "AllocatedStorage" : "5",
```

```
     "DBInstanceClass" : "db.m1.small",
     "Engine" : "MySQL",
     "MasterUsername" : "MyName",
     "MasterUserPassword" : "MyPassword"
 },
 "DeletionPolicy" : "Snapshot"
}
```

Amazon RDS Oracle Database 数据库实例资源

此示例所示为，使用库授权版许可证模型将引擎指定为 oracle-ee，从而创建 Oracle Database DB 实例资源。有关 Oracle Database 数据库实例设置的详细信息，请参阅 CreateDBInstance。DBSecurityGroups 属性授权对名为 MyDbSecurityByEC2SecurityGroup 和 MyDbSecurityByCIDRIPGroup 的 AWS::RDS::DBSecurityGroup 资源的网络访问。有关详细信息，请参阅 AWS::RDS::DBInstance (p. 381)。DB 实例资源还有一个设置为 Snapshot 的 DeletionPolicy 属性。在 DeletionPolicy 设置为 Snapshot 的情况下，AWS CloudFormation 将在堆栈删除期间于删除此数据库实例之前创建此数据库实例的快照。

```
"MyDB" : {
 "Type" : "AWS::RDS::DBInstance",
 "Properties" : {
     "DBSecurityGroups" : [
       {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" : "MyDbSecurityByCID
RIPGroup"} ],
     "AllocatedStorage" : "5",
     "DBInstanceClass" : "db.m1.small",
     "Engine" : "oracle-ee",
     "LicenseModel" : "bring-your-own-license",
     "MasterUsername" : "master",
     "MasterUserPassword" : "SecretPassword01"
 },
 "DeletionPolicy" : "Snapshot"
}
```

适用于 CIDR 范围的 Amazon RDS DBSecurityGroup 资源

此示例显示了具有以格式 ddd.ddd.ddd.ddd/dd 指定的 CIDR 范围的进入授权的 Amazon RDS DBSecurityGroup 资源。有关详细信息，请参阅 AWS::RDS::DBSecurityGroup (p. 392) 和 RDS 安全组规则 (p. 472)。

```
"MyDbSecurityByCIDRIPGroup" : {
 "Type" : "AWS::RDS::DBSecurityGroup",
 "Properties" : {
     "GroupDescription" : "Ingress for CIDRIP",
     "DBSecurityGroupIngress" : {
         "CIDRIP" : "192.168.0.0/32"
     }
 }
}
```

带 Amazon EC2 安全组的 Amazon RDS DBSecurityGroup

此示例显示了具有由 MyEc2SecurityGroup 引用的 Amazon EC2 安全组的进入授权的 AWS::RDS::DBSecurityGroup (p. 392) 资源。

要执行此操作，您要定义 EC2 安全组，然后使用内部 Ref 函数参考 DBSecurityGroup 内的 EC2 安全组。

```
"DBInstance" : {
   "Type": "AWS::RDS::DBInstance",
   "Properties": {
      "DBName"            : { "Ref" : "DBName" },
      "Engine"            : "MySQL",
      "MasterUsername"    : { "Ref" : "DBUsername" },
      "DBInstanceClass"   : { "Ref" : "DBClass" },
      "DBSecurityGroups"  : [ { "Ref" : "DBSecurityGroup" } ],
      "AllocatedStorage"  : { "Ref" : "DBAllocatedStorage" },
      "MasterUserPassword": { "Ref" : "DBPassword" }
   }
},

"DBSecurityGroup": {
   "Type": "AWS::RDS::DBSecurityGroup",
   "Properties": {
      "DBSecurityGroupIngress": { "EC2SecurityGroupName": { "Ref": "WebServer
SecurityGroup" } },
      "GroupDescription"      : "Frontend Access"
   }
},

"WebServerSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" : {
      "GroupDescription" : "Enable HTTP access via port 80 and SSH access",
      "SecurityGroupIngress" : [
         {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" :
 "0.0.0.0/0"},
         {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
 "0.0.0.0/0"}
      ]
   }
}
```

从中提取此示例的完整模板可在以下网址查看：Drupal_Single_Instance_With_RDS.template

多 VPC 安全组

本示例显示了具有 AWS::RDS::DBSecurityGroupIngress (p. 394) 中的多个 Amazon EC2 VPC 安全组的进入授权的 AWS::RDS::DBSecurityGroup (p. 392) 资源。

```
{
   "Resources" : {
      "DBinstance" : {
         "Type" : "AWS::RDS::DBInstance",
         "Properties" : {
            "AllocatedStorage" : "5",
            "DBInstanceClass" : "db.m1.small",
            "DBName" : { "MyDBName" },
            "DBSecurityGroups" : [ { "Ref" : "DbSecurityByEC2SecurityGroup" }
],
```

```
            "DBSubnetGroupName" : { "Ref" : "MyDBSubnetGroup" },
            "Engine" : "MySQL",
            "MasterUserPassword": { "MyDBPassword" }
            "MasterUsername"    : { "MyDBUsername" },
         },
          "DeletionPolicy" : "Snapshot"
      },
      "DbSecurityByEC2SecurityGroup" : {
         "Type" : "AWS::RDS::DBSecurityGroup",
         "Properties" : {
            "GroupDescription" : "Ingress for Amazon EC2 security group",
            "EC2VpcId" : { "MyVPC" },
            "DBSecurityGroupIngress" : [ {
               "EC2SecurityGroupId" : "sg-b0ff1111",
               "EC2SecurityGroupOwnerId" : "111122223333"
            }, {
               "EC2SecurityGroupId" : "sg-ffd722222",
               "EC2SecurityGroupOwnerId" : "111122223333"
            } ]
         }
      }
   }
}
```

Amazon SimpleDB 代码段

Amazon SimpleDB 域资源

此示例显示 Amazon SimpleDB 域资源。

```
"MySDBDomain" : {
    "Type" : "AWS::SDB::Domain",
    "Properties" : {
       "Description" : "Other than this AWS CloudFormation Description property,
 SDB Domains have no properties."
    }
}
```

Amazon SNS 代码段

Topics

- Amazon SNS 主题资源 (p. 168)

Amazon SNS 主题资源

此示例显示 Amazon SNS 主题资源。它需要有效的电子邮件地址。

```
"MySNSTopic" : {
    "Type" : "AWS::SNS::Topic",
    "Properties" : {
        "Subscription" : [ {
            "Endpoint" : "add valid email address",
```

```
            "Protocol" : "email"
        } ]
    }
}
```

Amazon SQS 队列代码段

此示例显示 Amazon SQS 队列。

```
"MyQueue" : {
    "Type" : "AWS::SQS::Queue",
    "Properties" : {
        "VisibilityTimeout" : "value"
    }
}
```

Amazon CloudFront 模板代码段

Topics

- 带 Amazon S3 源的 Amazon CloudFront 分配资源 (p. 169)
- 带自定义源的 Amazon CloudFront 分配资源 (p. 170)
- 带多源支持的 Amazon CloudFront 分配。 (p. 171)

带 Amazon S3 源的 Amazon CloudFront 分配资源

此示例显示了使用 S3Origin (p. 433) 的 Amazon CloudFront 分配 (p. 256)。

```
 "myDistribution" : {
   "Type" : "AWS::CloudFront::Distribution",
   "Properties" : {
      "DistributionConfig" : {
          "Origins" : [ {
              "DomainName": "mybucket.s3.amazonaws.com",
              "Id" : "myS3Origin",
              "S3OriginConfig" : {
                  "OriginAccessIdentity" : "origin-access-identity/cloud
front/E127EXAMPLE51Z"
              }
          }],
          "Enabled" : "true",
          "Comment" : "Some comment",
          "DefaultRootObject" : "index.html",
          "Logging" : {
              "Bucket" : "mylogs.s3.amazonaws.com",
              "Prefix" : "myprefix"
          },
          "Aliases" : [ "mysite.example.com", "yoursite.example.com" ],
          "DefaultCacheBehavior" : {
              "TargetOriginId" : "myS3Origin",
              "ForwardedValues" : {
                  "QueryString" : "false"
```

```
            },
            "TrustedSigners" : [ "1234567890EX", "1234567891EX" ],
            "ViewerProtocolPolicy" : "allow-all"
        }
    }
  }
}
```

带自定义源的 Amazon CloudFront 分配资源

此示例显示了使用 CustomOrigin (p. 428) 的 Amazon CloudFront 分配 (p. 256)。

```
"myDistribution": {
    "Type": "AWS: : CloudFront: : Distribution",
    "Properties": {
        "DistributionConfig": {
            "Origins": [
                {
                    "DomainName": "www.example.com",
                    "Id": "myCustomOrigin",
                    "CustomOriginConfig": {
                        "HTTPPort": "80",
                        "HTTPSPort": "443",
                        "OriginProtocolPolicy": "http-only"
                    }
                }
            ],
            "Enabled": "true",
            "Comment": "Somecomment",
            "DefaultRootObject": "index.html",
            "Logging": {
                "Bucket": "mylogs.s3.amazonaws.com",
                "Prefix": "myprefix"
            },
            "Aliases": [
                "mysite.example.com",
                "*.yoursite.example.com"
            ],
            "DefaultCacheBehavior": {
                "TargetOriginId": "myCustomOrigin",
                "ForwardedValues": {
                    "QueryString": "false"
                },
                "TrustedSigners": [
                    "1234567890EX",
                    "1234567891EX"
                ],
                "ViewerProtocolPolicy": "allow-all"
            }
        }
    }
}
```

带多源支持的 Amazon CloudFront 分配。

此模板代码段显示如何声明带多源支持的 CloudFront 分配 (p. 256)。在 DistributionConfig (p. 430) 中，提供了源列表，并且设置了 DefaultCacheBehavior (p. 429)。

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myDistribution" : {
            "Type" : "AWS::CloudFront::Distribution",
            "Properties" : {
               "DistributionConfig" : {
                   "Origins" : [ {
                           "Id" : "myS3Origin",
                           "DomainName" : "mybucket.s3.amazonaws.com",
                           "S3OriginConfig" : {
                               "OriginAccessIdentity" : "origin-access-iden
tity/cloudfront/E127EXAMPLE51Z"
                           }
                       },
                       {
                           "Id" : "myCustomOrigin",
                           "DomainName" : "www.example.com",
                           "CustomOriginConfig" : {
                               "HTTPPort" : "80",
                               "HTTPSPort" : "443",
                               "OriginProtocolPolicy" : "http-only"
                           }
                       }
                   ],
                   "Enabled" : "true",
                   "Comment" : "Some comment",
                   "DefaultRootObject" : "index.html",
                   "Logging" : {
                       "Bucket" : "mylogs.s3.amazonaws.com",
                       "Prefix" : "myprefix"
                   },
                   "Aliases" : [ "mysite.example.com", "yoursite.example.com"
],
                   "DefaultCacheBehavior" : {
                       "TargetOriginId" : "myS3Origin",
                       "ForwardedValues" : {
                           "QueryString" : "false"
                        },
                       "TrustedSigners" : [ "1234567890EX", "1234567891EX"  ],

                       "ViewerProtocolPolicy" : "allow-all",
                       "MinTTL" : "100"
                   },
                   "CacheBehaviors" : [ {
                            "TargetOriginId" : "myS3Origin",
                            "ForwardedValues" : {
                                "QueryString" : "true"
                            },
                           "TrustedSigners" : [ "1234567890EX", "1234567891EX"
 ],
```

```
                        "ViewerProtocolPolicy" : "allow-all",
                        "MinTTL" : "50",
                        "PathPattern" : "images1/*.jpg"
                    },
                    {
                        "TargetOriginId" : "myCustomOrigin",
                        "ForwardedValues" : {
                            "QueryString" : "true"
                        },
                       "TrustedSigners" : [ "1234567890EX", "1234567891EX"
  ],
                        "ViewerProtocolPolicy" : "allow-all",
                        "MinTTL" : "50",
                        "PathPattern" : "images2/*.jpg"
                    }
                ]
            }
        }
    }
}
```

Amazon Route 53 模板代码段

Topics

- 使用托管区名称或 ID 的 Amazon Route 53 资源记录集 (p. 172)
- 使用 RecordSetGroup 设置加权资源记录集 (p. 173)
- 使用 RecordSetGroup 设置别名资源记录集 (p. 174)

使用托管区名称或 ID 的 Amazon Route 53 资源记录集

创建 Amazon Route 53 资源记录集时，必须指定要添加该记录集的托管区域。AWS CloudFormation 提供了两种用于执行此操作的方法。您可以使用 HostedZoneId 属性显式指定托管区域，也可以使用 HostedZoneName 属性让 AWS CloudFormation 找到托管区域。如果使用 HostedZoneName 属性且存在多个具有相同域名的托管区域，则 AWS CloudFormation 将不会创建堆栈。

使用 HostedZoneId 添加 RecordSet

此示例使用 HostedZoneId 属性添加包含域名 "mysite.example.com" 之别名记录的 Amazon Route 53 资源记录集，以指定托管区。

```
"myDNSRecord" : {
            "Type" : "AWS::Route53::RecordSet",
            "Properties" : {
                "HostedZoneId" : "/hostedzone/Z3DG6IL3SJCGPX",
                "Comment" : "CNAME for my frontends.",
                "Name" : "mysite.example.com.",
                "Type" : "CNAME",
                "TTL" : "900",
                "ResourceRecords" : [
                    {"Fn::GetAtt":["myLB","DNSName"]}
                ]
            }
```

```
        }
```

使用 HostedZoneName 添加 RecordSet

此示例使用 HostedZoneName 属性添加包含域名 "mysite.example.com" 之 A 记录的 Amazon Route 53 资源记录集，以指定托管区。

```
"myDNSRecord2" : {
            "Type" : "AWS::Route53::RecordSet",
            "Properties" : {
                "HostedZoneName" : "example.com.",
                "Comment" : "A records for my frontends.",
                "Name" : "mysite.example.com.",
                "Type" : "A",
                "TTL" : "900",
                "ResourceRecords" : [
                    "192.168.0.1",
                    "192.168.0.2"
                ]
            }
        }
```

使用 RecordSetGroup 设置加权资源记录集

此示例使用 AWS::Route53::RecordSetGroup (p. 400) 为“example.com.”托管区域设置两条别名记录。*RecordSets* 属性包含“mysite.example.com”DNS 名称的别名记录集。每个记录集中都含有一个标识符 (SetIdentifier) 和权重 (Weight)。Frontend One 的权重是 40%（4/10），Frontend Two 是 60%（6/10）。有关加权资源记录集的更多信息，请参阅 Route 53 开发人员指南中的设置加权资源记录集。

```
        "myDNSOne" : {
            "Type" : "AWS::Route53::RecordSetGroup",
            "Properties" : {
                "HostedZoneName" : "example.com.",
                "Comment" : "Weighted RR for my frontends.",
                "RecordSets" : [
                  {
                    "Name" : "mysite.example.com.",
                    "Type" : "CNAME",
                    "TTL" : "900",
                    "SetIdentifier" : "Frontend One",
                    "Weight" : "4",
                    "ResourceRecords" : ["example-ec2.amazonaws.com"]
                  },
                  {
                    "Name" : "mysite.example.com.",
                    "Type" : "CNAME",
                    "TTL" : "900",
                    "SetIdentifier" : "Frontend Two",
                    "Weight" : "6",
                    "ResourceRecords" : ["example-ec2-larger.amazonaws.com"]
                  }
                  ]
            }
```

```
        }
```

使用 RecordSetGroup 设置别名资源记录集

此示例使用 AWS::Route53::RecordSetGroup (p. 400) 为“example.com.”托管区域设置别名资源记录集。*RecordSets* 属性包含区域顶点“example.com”的 A 记录。AliasTarget (p. 473) 属性使用 GetAtt (p. 502) 内部函数指定 myELB 负载均衡器的托管区域 ID 和 DNS 名称，以检索 myELB 资源的 CanonicalHostedZoneNameID 和 CanonicalHostedZoneName 属性。有关别名资源记录集的更多信息，请参阅 *Route 53 开发人员指南*中的创建别名资源记录集。

```
      "myELB" : {
        "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
        "Properties" : {
            "AvailabilityZones" : [ "us-east-1a" ],
            "Listeners" : [ {
                "LoadBalancerPort" : "80",
                "InstancePort" : "80",
                "Protocol" : "HTTP"
            } ]
        }
      },
      "myDNS" : {
        "Type" : "AWS::Route53::RecordSetGroup",
        "Properties" : {
          "HostedZoneName" : "example.com.",
          "Comment" : "Zone apex alias targeted to myELB LoadBalancer.",
          "RecordSets" : [
            {
              "Name" : "example.com.",
              "Type" : "A",
              "AliasTarget" : {
                 "HostedZoneId" : { "Fn::GetAtt" : ["myELB", "CanonicalHosted
ZoneNameID"] },
                  "DNSName" : { "Fn::GetAtt" : ["myELB","CanonicalHostedZone
Name"] }
              }
            }
          ]
        }
    }
```

Amazon S3 模板代码段

Topics

- 创建带默认设置的 Amazon S3 存储桶 (p. 174)
- 创建用于网站托管并带 DeletionPolicy 的 Amazon S3 存储桶 (p. 175)

创建带默认设置的 Amazon S3 存储桶

此示例使用 AWS::S3::Bucket (p. 402) 创建带默认设置的存储桶。

```
"myS3Bucket" : {
      "Type" : "AWS::S3::Bucket"
      }
```

创建用于网站托管并带 DeletionPolicy 的 Amazon S3 存储桶

此示例将存储桶创建为网站。AccessControl 属性被设置为已存 ACL PublicRead（设置虚拟主机的存储桶需要公共读取许可）。由于此存储桶资源的 DeletionPolicy 属性 (p. 485)设置为 *Retain*，因此 AWS CloudFormation 在删除堆栈时将不会删除此存储桶。Output 部分使用 *Fn::GetAtt* 来检索 S3Bucket 资源的 WebsiteURL 属性和 DomainName 属性。

```
"{
      "AWSTemplateFormatVersion" : "2010-09-09",
      "Resources" : {
          "S3Bucket" : {
             "Type" : "AWS::S3::Bucket",
             "Properties" : {
                 "AccessControl" : "PublicRead",
                 "WebsiteConfiguration" : {
                     "IndexDocument" : "index.html",
                     "ErrorDocument" : "error.html"
                 }
             },
             "DeletionPolicy" : "Retain"
          }
      },
      "Outputs" : {
          "WebsiteURL" : {
                "Value" : { "Fn::GetAtt" : [ "S3Bucket", "WebsiteURL" ] },
                "Description" : "URL for website hosted on S3"
          },
          "S3BucketSecureURL" : {
                "Value" : { "Fn::Join" : [ "", [ "https://", { "Fn::GetAtt" :
[ "S3Bucket", "DomainName" ] } ] ] },
                "Description" : "Name of S3 bucket to hold website content"
          }
      }
}
```

堆栈资源代码段

Topics

- 在模板中嵌套堆栈 (p. 175)
- 使用输入参数在模板中嵌套堆栈 (p. 176)

在模板中嵌套堆栈

此示例模板包含一个称为 myStack 的嵌套堆栈资源。当 AWS CloudFormation 从模板创建堆栈时，它会创建 myStack，其模板在 *TemplateURL* 属性中指定。 输出值 StackRef 返回 myStack 的堆栈 ID，值 OutputFromNestedStack 从 myStack 资源内返回输出值 BucketName。Outputs.*nestedstackoutputname* 格式已预留，以便指定嵌套堆栈的输出值，并且该格式可在包含模板内的任何位置使用。

有关更多信息，请参阅 AWS::CloudFormation::Stack (p. 250)。

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myStack" : {
        "Type" : "AWS::CloudFormation::Stack",
        "Properties" : {
           "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-
us-east-1/S3_Bucket.template",
              "TimeoutInMinutes" : "60"
        }
        }
    },
    "Outputs": {
       "StackRef": {"Value": { "Ref" : "myStack"}},
       "OutputFromNestedStack" : {
             "Value" : { "Fn::GetAtt" : [ "myStack", "Outputs.BucketName" ] }
       }
    }
}
```

使用输入参数在模板中嵌套堆栈

此示例模板包含指定输入参数的堆栈资源。当 AWS CloudFormation 使用此模板创建堆栈时，它会将 Parameters 属性中声明的值对用作创建 myStackWithParams 堆栈所用模板的输入参数。此示例中，InstanceType 和 KeyName 参数已指定。

有关更多信息，请参阅 AWS::CloudFormation::Stack (p. 250)。

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Resources" : {
        "myStackWithParams" : {
          "Type" : "AWS::CloudFormation::Stack",
        "Properties" : {
           "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-
us-east-1/EC2ChooseAMI.template",
            "Parameters" : {
                "InstanceType" : "t1.micro",
                "KeyName" : "mykey"
            }
           }
        }
    }
}
```

等候条件模板代码段

Topics

- 将等待条件用于 Amazon EC2 实例 (p. 177)
- 使用 Curl 发出等候条件信号 (p. 178)

将等待条件用于 Amazon EC2 实例

此示例模板使用等待条件声明 Amazon EC2 实例。等待条件 myWaitCondition 使用 myWaitConditionHandle 发出信号，使用 DependsOn 属性 (p. 486)指定等待条件将在创建 Ec2Instance 资源后触发，并使用 Timeout 属性将等待条件的持续时间指定为 4500 秒。此外，发出等候条件信号的预签署 URL 将通过 Ec2Instance 资源的 UserData 属性传输至 EC2 实例，以使应用程序或脚本可在该 EC2 实例上运行，从而检索预签署 URL 并用其发出成功或失败信号至等候条件。请注意，您需要创建发出等候条件信号的应用程序或脚本。输出值 ApplicationData 中包含从等候条件信号中传回的数据。

有关更多信息，请参阅在模板中创建等待条件 (p. 188)、AWS::CloudFormation::WaitCondition (p. 252)和 AWS::CloudFormation::WaitConditionHandle (p. 255)。

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Mappings" : {
        "RegionMap" : {
            "us-east-1" : {
                "AMI" : "ami-76f0061f"
            },
            "us-west-1" : {
                "AMI" : "ami-655a0a20"
            },
            "eu-west-1" : {
                "AMI" : "ami-7fd4e10b"
            },
            "ap-northeast-1" : {
                "AMI" : "ami-8e08a38f"
            },
            "ap-southeast-1" : {
                "AMI" : "ami-72621c20"
            }
        }
    },
    "Resources" : {
        "Ec2Instance" : {
            "Type" : "AWS::EC2::Instance",
            "Properties" : {
                "UserData" : { "Fn::Base64" : {"Ref" : "myWaitHandle"}},
                "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
"AWS::Region" }, "AMI" ]}
            }
        },
        "myWaitHandle" : {
            "Type" : "AWS::CloudFormation::WaitConditionHandle",
            "Properties" : {
            }
        },
        "myWaitCondition" : {
            "Type" : "AWS::CloudFormation::WaitCondition",
            "DependsOn" : "Ec2Instance",
            "Properties" : {
                "Handle" : { "Ref" : "myWaitHandle" },
                "Timeout" : "4500"
            }
        }
    },
    "Outputs" : {
```

```
        "ApplicationData" : {
            "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ]},
            "Description" : "The data passed back as part of signalling the
WaitCondition."
        }
    }
}
```

使用 Curl 发出等候条件信号

此示例显示将成功信号发送至等待条件的 Curl 命令行。

```
curl -T /tmp/a https://cloudformation-waitcondition-test.s3.amazon
aws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack%2Fstack-
gosar-20110427004224-test-stack-with-WaitCondition--VEYW%2Fe498ce60-70a1-11e0-
81a7-5081d0136786%2FmyWaitConditionHandle?Expires=1303976584&AWSAccessKeyId=AKI
AIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVoo%3D
```

其中文件 /tmp/a 包含以下 JSON 结构：

```
{
  "Status" : "SUCCESS",
  "Reason" : "Configuration Complete",
  "UniqueId" : "ID1234",
  "Data" : "Application has completed configuration."
}
```

此示例所示为发送同一成功信号的 Curl 命令行，将 JSON 发送为命令行参数的情况除外。

```
curl -X PUT -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason" :
 "Configuration Complete","UniqueId" : "ID1234","Data" : "Application has com
pleted configuration."}'  https://cloudformation-waitcondition-test.s3.amazon
aws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack%2Fstack-
gosar-20110427004224-test-stack-with-WaitCondition--VEYW%2Fe498ce60-70a1-11e0-
81a7-5081d0136786%2FmyWaitConditionHandle?Expires=1303976584&AWSAccessKeyId=AKI
AIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVoo%3D
```

AWS CloudFormation 模板代码段

Topics

Base64 编码 UserData 属性

此示例显示的是使用 Fn::Base64 和 Fn::Join 函数的 UserData 属性集合。引用 *MyValue* 和 *MyName* 是必须在模板的 Parameters 部分中定义的参数。文字字符串 *Hello World* 只是作为 *UserData* 的一部分由此示例传入的另一个值。

```
"UserData" : {
    "Fn::Base64" : {
        "Fn::Join" : [ ",", [
            { "Ref" : "MyValue" },
            { "Ref" : "MyName" },
            "Hello World" ] ]
    }
}
```

带 AccessKey 和 SecretKey 的 Base64 编码 UserData 属性

此示例显示的是使用 Fn::Base64 和 Fn::Join 函数的 UserData 属性集合。它包含 AccessKey 和 SecretKey 信息。引用 *AccessKey* 和 *SecretKey* 是必须在模板的 Parameters 部分中定义的参数。

```
"UserData" : {
    "Fn::Base64" : {
        "Fn::Join" : [ "", [
            "ACCESS_KEY=", { "Ref" : "AccessKey" },
            "SECRET_KEY=", { "Ref" : "SecretKey" } ]
        ]
    }
}
```

含一个文字字符串参数的 Parameters 部分

以下示例说明了有效的 Parameters 部分声明，其中声明了一个 String 类型参数。

```
"Parameters" : {
    "UserName" : {
        "Type" : "String",
        "Default" : "nonadmin",
       "Description" : "Assume a vanilla user if no command-line spec provided"

    }
}
```

含有带正则表达式约束的字符串参数的 Parameters 部分

以下示例说明了有效的 Parameters 部分声明，其中声明了一个 String 类型参数。AdminUserAccount 参数有一个默认值 admin。参数值的最小长度必须为 1，最大长度必须为 16，且其中包含字母字符和数字，但必须以字母字符开头。

```
"Parameters" : {
    "AdminUserAccount": {
      "Default": "admin",
      "NoEcho": "true",
      "Description" : "The admin account user name",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "16",
      "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"
    }
}
```

含有带 MinValue 和 MaxValue 约束的数字参数的 Parameters 部分

以下示例说明了有效的 Parameters 部分声明，其中声明了一个 Number 类型参数。WebServerPort 参数有一个默认值 80，最小值 1 和最大值 65535。

```
"Parameters" : {
    "WebServerPort": {
      "Default": "80",
      "Description" : "TCP/IP port for the web server",
      "Type": "Number",
      "MinValue": "1",
      "MaxValue": "65535"
    }
}
```

含有带 AllowedValues 约束的数字参数的 Parameters 部分

以下示例说明了有效的 Parameters 部分声明，其中声明了一个 Number 类型参数。WebServerPort 参数有一个默认值 80，且只允许值 80 和 8888。

```
"Parameters" : {
    "WebServerPortLimited": {
      "Default": "80",
      "Description" : "TCP/IP port for the web server",
      "Type": "Number",
      "AllowedValues" : ["80", "8888"]
    }
}
```

带有一个文字 CommaDelimitedList 参数的 Parameters 部分

以下示例说明了有效的 Parameters 部分声明，其中声明了一个 `CommaDelimitedList` 类型参数。NoEcho 属性设置为 `TRUE`，这会在 `aws cloudformation describe-stacks` 输出中使用星号 (*****) 对其值进行遮蔽。

```
"Parameters" : {
    "UserRoles" : {
        "Type" : "CommaDelimitedList",
        "Default" : "guest,newhire",
        "NoEcho" : "TRUE"
```

```
    }
}
```

带有基于 虚拟参数的参数值的 Parameters 部分

此示例显示基于虚拟参数“AWS::StackName”所返回值进行的参数分配。

```
"Parameters" : {
    "StackName" : {
        "Type" : "String",
        "Default" : { "Ref" : "AWS::StackName"}
    }
},
```

带有三个映射的 Mapping 部分

以下示例说明了其中包含三个映射的有效 Mapping 部分。该映射在与 *Stop*、*SlowDown* 或 *Go* 的映射密钥匹配时，将提供分配给相应 *RGBColor* 属性的 RGB 值。

```
"Mappings" : {
    "LightColor" : {
        "Stop" : {
            "Description" : "red",
            "RGBColor" : "RED 255 GREEN 0 BLUE 0"
        },
        "SlowDown" : {
            "Description" : "yellow",
            "RGBColor" : "RED 255 GREEN 255 BLUE 0"
        },
        "Go" : {
            "Description" : "green",
            "RGBColor" : "RED 0 GREEN 128 BLUE 0"
        }
    }
},
```

基于文字字符串的说明

以下示例说明了有效的说明部分声明，其中的值基于文字字符串。此代码段可用于模板、参数、资源、属性或输出。

```
"Description" : "Replace this value"
```

带有一个文件字符串输出的 Outputs 部分

此示例显示的是基于文字字符串的输出分配。

```
"Outputs" : {
    "MyPhone" : {
        "Value" : "Please call 555-5555",
       "Description" : "A random message for aws cloudformation describe-stacks"
```

```
    }
}
```

带有一个资源参考和一个虚拟参数输出的 Outputs 部分

此示例显示的是含有两个输出分配的 Outputs 部分。一个分配基于资源，另一个基于 pseudo 参考。

```
"Outputs" : {
   "SNSTopic" : { "Value" : { "Ref" : "MyNotificationTopic" } },
   "StackName" : { "Value" : { "Ref" : "AWS::StackName" } }
}
```

含有一个基于函数、文字字符串、参考和 虚拟参数的输出的 Outputs 部分

此示例显示的是带有一个输出分配的 Outputs 部分。Join 函数用于连接值，将百分比符号用作分隔符。

```
"Outputs" : {
    "MyOutput" : {
        "Value" : { "Fn::Join" :
            [ "%", [ "A-string", {"Ref" : "AWS::StackName" } ] ]
        }
    }
}
```

模板格式版本

以下代码段说明了有效的 Template Format Version 部分声明。

```
"AWSTemplateFormatVersion" : "2010-09-09"
```

AWS 标签属性

此示例显示的是 AWS 标签属性。您将在资源的 Properties 部分中执行此属性。资源被创建后，会标上您声明的标签。

```
"Tags" : [
      {
        "Key" : "keyname1",
        "Value" : "value1"
      },
      {
        "Key" : "keyname2",
        "Value" : "value2"
      }
    ]
},
```

修改 AWS CloudFormation 模板

Abstract

修改 AWS CloudFormation 模板以自定义配置。

Topics

将输入参数添加至您的模板

Abstract

通过在模板的 Parameters 部分中添加参数来配置 AWS CloudFormation 模板，以要求使用输入参数。

通过将输入参数添加至“参数”部分，您就可以配置模板，请求输入参数了。您输入的所有参数必须含有运行时的数值。您可以为每个参数指定默认值以使该参数成为可选参数。如果您未指定默认值，那么您创建堆栈时，必须提供针对该参数的数值。

可将参数声明为 *String*、*Number* 或 *CommaDelimitedList* 类型。*String* 和 *Number* 类型可以具有 AWS CloudFormation 用于验证参数值的约束条件。有关参数约束条件的更多信息，请参阅参数声明 (p. 101)。

以下示例配置了一个单独参数 *Email*：

```
"Parameters" : {
    "Email" : {
        "Type" : "String"
    }
}
```

上述参数无默认值，因此，您必须为其提供一个数值，以创建堆栈。在使用 *Email* 的值创建 CloudWatch Alarms 堆栈后，`aws cloudformation describe-stacks` 命令将返回如下内容：

```
STACK  myAlarms
arn:aws:aws cloudformation:us-east-1:165024647323:stack/f5b4cbb0-24d7-11e0-93a-
508be05d086/myAlarms
Email=Joe@Joe.com  2011-01-20T20:57:57Z  CREATE_COMPLETE
User Initiated  false  Instance=i-0723826b
```

您可以将该参数配置为不与 *NoEcho* 参数一起显示：

```
"Parameters" : {
    "Email" : {
        "Type" : "String",
        "NoEcho" : "TRUE"
```

```
    }
}
```

以下是使用同一模板创建的堆栈的输出，但其 *NoEcho* 设置为 *TRUE*：

```
STACK  myAlarms2
arn:aws:aws cloudformation:us-east-1:165024647323:stack/ff6ff540-24db-11e0-94f8-
5081b017c4b/myAlarms2
Email=******  2011-01-20T21:26:52Z  CREATE_COMPLETE  User Initiated
false  Instance=i-f734959b
```

Email 的值使用星号进行遮蔽。

要为参数提供值，可向 `aws cloudformation create-stack` 命令添加 *--parameters* 选项。

例如，以下命令为 *UserName* 和 *Password* 参数添加了值：

```
PROMPT> aws cloudformation create-stack --stack-name MyStack --template-body
file:///home/local/test/sampletemplate.json
--parameters ParameterKey=UserName,ParameterValue=Joe ParameterKey=Password,Para
meterValue=JoesPw
```

用空格将参数隔开。注意：参数名称应区分大小写。如果您在运行 `aws cloudformation create-stack` 时错误键入参数名称，则 AWS CloudFormation 将不会创建堆栈，并且会报告该模板不包含此参数。

在模板中使用“参数”和“映射”来指定值

Abstract

使用输入参数可引用处于 AWS CloudFormation 模板内的映射中的特定值。

您可以通过 `Fn::FindInMap` 函数，使用输入参数引用映射中的特定值。例如，假定您有一个映射到特定 AMI 的区域的列表。您可以通过在创建堆栈时指定区域参数来选择堆栈所用的 AMI。

1. 将一个参数添加到您想算入的所有映射的参数部分。参数就是您传输所需映射密钥的方式。
2. 创建包含密钥选项和密钥值的映像。
3. 使用 `Fn::FindInMap` 函数作为要按条件分配的资源属性或输出的值。

Note

您可以为参数和映像设定默认值。您需要为上述至少一项设定默认值。如果 `Fn::FindInMap` 在运行时未能解析密钥和值，则会发生错误，从而阻止创建堆栈。

仔细考虑下述示例。假设您希望 `aws cloudformation describe-stacks` 命令打印要基于特定区域运行的 AMI 的名称。您需要执行下述步骤完成此操作：

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "TemplateName - ShortMapExample.template",

  "Parameters" : {
```

```
    "Region" : {
      "Default" : "us-east-1",
     "Description" : " 'us-east-1' | 'us-west-1' | 'eu-west-1' | 'ap-southeast-
1' "
    }
  },

  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {
          "AMI" : "ami-76f0061f"
      },
      "us-west-1" : {
          "AMI" : "ami-655a0a20"
      },
      "eu-west-1" : {
          "AMI" : "ami-7fd4e10b"
      },
      "ap-southeast-1" : {
          "AMI" : "ami-72621c20"
      }
    }
  },

  "Resources" : {

   ...other resources...

    },

  "Outputs" : {
    "OutVal" : {
      "Description" : "Return the name of the AMI matching the RegionMap key",

      "Value" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "Region" }, "AMI"
 ]}
    }
  }

}
```

参数 *Region* 接受字符串值，在理想情况下该字符串值是模板中的区域标识符之一。Mappings 部分将声明 *RegionMap* 映射。每个映射密钥均为 *AMI* 属性分配一个值。Outputs 部分将声明 *OutVal* 输出，该输出基于从 Fn:FindInMap 返回的值来获取其值。

下面显示了基于所列命令分配给 *OutVal* 的值：

命令行	分配至 OutVal 的数值

`aws cloudformation create-stack --stack-name MyTestStack --template-body file:///home/local/test/ShortRegionExample.json --parameters ParameterKey=Region,ParameterValue=us-west-1`
`...`
`aws cloudformation describe-stacks --stack-name MyTestStack`
	ami-655a0a20

`aws cloudformation create-stack --stack-name MyTestStack --template-body file:///home/local/test/ShortRegionExample.json --parameters ParameterKey=Region,ParameterValue=eu-west-1`
`...`
`aws cloudformation describe-stacks --stack-name MyTestStack`
	ami-7fd4e10b

`aws cloudformation create-stack --stack-name MyTestStack --template-body file:///home/local/test/ShortRegionExample.json`
`...`
`aws cloudformation describe-stacks MyTestStack`
	ami-76f0061f

在前两个示例中，作为 *--parameters* 选项的一部分指定的值确定了 *OutVal* 的值。在第三个示例中，未指定映射密钥，因此将使用默认区域 *us-east-1*。

按照条件创建资源

Abstract

使用输入参数可在创建或更新堆栈时按条件创建资源。

在创建或更新堆栈时，您可以通过对输入参数和映射设置条件来创建资源。您可以设置结果各不相同的多个条件。例如，您可以指定 Amazon EC2 安全组作为输入参数，在堆栈中使用该安全组。不过，如果未提供安全组，则将创建您在模板中指定的安全组。

您可以通过完成以下步骤按照条件创建资源：

1. 在模板的“参数”部分，定义可以在条件中使用的输入参数。有关更多信息，请参阅 将输入参数添加至您的模板 (p. 183)。
2. 在模板的“条件”部分，通过针对条件使用内部函数来定义要使用的条件。有关更多信息，请参阅 条件声明 (p. 107)。
3. 在模板的“资源”和“输出”部分，将条件与相关的资源或属性关联。有关更多信息，请参阅 条件声明 (p. 107)。

有关其他示例模板以及有关条件语法的信息，请参阅条件函数 (p. 491)。

标记您的成员资源

Abstract

使用 AWS CloudFormation 自动标签可查看和查找与您的 AWS CloudFormation 堆栈相关的资源。

AWS CloudFormation 自动使用堆栈名称标记您的资源，这样，当您在 AWS 管理控制台中查看这些资源时，就可以按堆栈名称进行筛选了。

除了 AWS CloudFormation 为您添加的堆栈名称标签外，您也可以为支持设置标签的资源添加自定义标签。

Note

您添加至成员资源的标签不会显示在 `aws cloudformation describe-stack-resources` 的输出中。但是，它们将在已添加标签资源选项的 AWS 管理控制台上显示。

假设您想要自定义模板，以包括标签 *`Stage`*（标记部署阶段）和 *`QA`*（标记其值）。您可以按如下所述编写 *`MyInstance`* 资源的定义：

```
"MyInstance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
        "SecurityGroups" : [ { "Ref" : "MySecurityGroup" } ],
        "AvailabilityZone" : "us-east-1a",
        "ImageId" : "ami-20b65349",
        "Volumes" : [
            { "VolumeId" : { "Ref" : "MyEBS" },
                        "Device" : "/dev/sdk" }
        ],
        "Tags" : [
            {
                "Key" : "Stage",
                "Value" : "QA"
            }
       ]
    }
}
```

创建堆栈后，您就可以在 AWS 管理控制台中按 *`Stage`* 标签进行筛选了。

通过输出指定返回值

Abstract

使用 AWS CloudFormation 模板 Outputs 部分可指定您堆栈的返回值。

您可以使用模板 Outputs 部分指定将包括在 `aws cloudformation describe-stacks` 命令返回的值中的自定义值。您可以根据模板属性规则指定每个自定义值（资源属性 (p. 9)），这样您可以使其值基于文字、参数引用、虚拟参数、映射值和内部函数。

对于简单示例，一个示例模板声明两项输出，即 *`IPAddress`* 和 *`InstanceId`*：

```
"Outputs" : {
    "IPAddress" : {
```

```
        "Value" : { "Ref" : "MyIp" }
    },

    "InstanceId" : {
        "Value" : { "Ref" : "MyInstance" }
    }
}
```

这两个值均基于模板中声明的逻辑名称。*IPAddress* 引用逻辑名称为 *MyIp* 的 AWS::EC2::EIP 类型，而 *InstanceId* 则引用逻辑名称为 *MyInstance* 的 AWS::EC2::Instance 类型。

创建堆栈后，`aws cloudformation describe-stacks` 会将其状态报告为 `CREATE_COMPLETE`，并且还报告以下内容：

```
PROMPT> aws cloudformation describe-stacks --stack-name StackName
...
    "Outputs": [
        {
            "OutputKey": "IPAddress",
            "OutputValue": "184.72.229.56"
        },
        {
            "OutputKey": "InstanceId",
            "OutputValue": "i-47ab0a2b"
        }
    ],
...
```

在报告末尾将显示自定义输出值 `IPAddress` 和 `InstanceId`。

在模板中创建等待条件

Abstract

在模板中创建等待条件，以便暂停堆栈的创建并在继续创建堆栈前等待一个信号。

使用 AWS::CloudFormation::WaitCondition (p. 252) 和 AWS::CloudFormation::WaitConditionHandle (p. 255) 资源，您可以将等待条件置于模板中，以使 AWS CloudFormation 暂停堆栈创建，并在等到信号后再继续创建堆栈。

您可以使用等待条件将创建堆栈资源与堆栈创建外部的其他配置操作进行协调。例如，考虑 Amazon EC2 实例完成创建前，您可能想要下载和配置 Amazon EC2 上的应用程序。

以下内容展示了等待条件的运行方式：

- AWS CloudFormation 与其他任何资源一样，将创建等待条件。AWS CloudFormation 创建等待条件时，会报告等待条件的状态为 CREATE_IN_PROGRESS，并进行等待，直到它收到必需数量的成功信号或等待条件的超时期限已过期。如果在超时周期过期前，AWS CloudFormation 接收必要数量的成功信号，那么它将继续创建堆栈；否则，它会将等待条件状态设定为 CREATE_FAILED 并回滚堆栈。
- `Timeout` 属性将确定 AWS CloudFormation 等待必需数量的成功信号的时间长度。`Timeout` 是一种最低时限属性，表示超时的发生时间不会早于您指定的时间，但会在您指定的时间后很短的时间内发生。
- 通常情况下，特定资源创建后，例如 Amazon EC2 实例、RDS DB 实例或 Auto Scaling 组，您会想要等待条件立即开始。可以通过为等待条件添加 DependsOn 属性 (p. 486)来完成此操作。当您将 DependsOn 属性添加至等待条件后，仅可在特定资源创建完成后指定创建等待条件。创建等待条件后，AWS CloudFormation 将开始超时周期，并等待成功信号。

- 您还可以使用其他资源上的 DependsOn 属性。例如，创建使用数据库的 EC2 实例前，您可能想要待创建 RDS DB 实例以及在 DB 实例上配置的数据库。在这种情况下，您可以创建拥有指定 DB 实例的 DependsOn 属性的等待条件，此外，您还可以创建拥有指定等待条件的 DependsOn 属性的 EC2 实例资源。通过上述操作可以确保 DB 实例和等待条件完成后，仅可以直接创建 EC2 实例。
- AWS CloudFormation 在将等待条件状态设置为 CREATE_COMPLETE 以继续创建堆栈前，必须接收等待条件指定数量的成功信号。等待条件的计数属性将指定成功信号的数量。如果无一设置，则默认值为 1。
- 等待条件需要等待条件句柄来设置用作信号机制的预签名 URL。通过预签名 URL，您可以发送信号，而无需提供 AWS 证书。您将使用预签名 URL 发送成功或失败信号，其内嵌于 JSON 语句中。有关 JSON 语句格式，请参阅等待条件发送 JSON 格式 (p. 191)。有关向预签名 URL 发送 JSON 语句的 Curl 命令示例，请参阅 等候条件模板代码段 (p. 176).
- 如果在超时周期过期前，等待条件接受了必要数量的成功信号（按照计数属性的定义），那么 AWS CloudFormation 会将等待信号标记为 CREATE_COMPLETE，并继续创建堆栈。否则，AWS CloudFormation 将放弃等待条件，并回滚堆栈（例如，如果超时周期过期，而此时并未接收到必要数量的成功信号或者接收到失败信号）。

如需在堆栈中使用等待条件：

1. 在堆栈模板中声明 AWS::CloudFormation::WaitConditionHandle 资源。等待条件句柄并无属性；但是，WaitConditionHandle 资源参考将解析您用于向 WaitCondition 发送成功或失败信号的预签名 URL。例如：

```
"myWaitHandle" : {
     "Type" : "AWS::CloudFormation::WaitConditionHandle",
     "Properties" : {
     }
}
```

2. 在堆栈模板中声明 AWS::CloudFormation::WaitCondition 资源。WaitCondition 资源具有两个必需属性：Handle 是对模板中声明的 WaitConditionHandle 的引用，而 Timeout 是 AWS CloudFormation 要等待的秒数。另外，您可以选择设置计数属性，其将确定 AWS CloudFormation 重新开始创建堆栈前，等待条件必须接收的成功信号数量。

   您应在等待条件中设置 DependsOn 属性，才能在等待条件触发后进行控制操作。DependsOn 子句将资源与等待条件相关联。在 AWS CloudFormation 创建 DependsOn 资源后，它将阻止进一步创建堆栈资源，直到发生以下事件之一：a) 超时期限到期 b) 收到必需数量的成功信号 c) 收到失败信号。

   此处为等待条件的一项示例，其在 Ec2Instance 资源成功创建后开始，它运用 myWaitHandle 资源作为 WaitConditionHandle，超时为 4500 秒，默认计数为 1（因尚未指定计数属性）：

```
"myWaitCondition" : {
    "Type" : "AWS::CloudFormation::WaitCondition",
    "DependsOn" : "Ec2Instance",
    "Properties" : {
        "Handle" : { "Ref" : "myWaitHandle" },
         "Timeout" : "4500"
    }
}
```

3. 获取用于发送信号的预签名 URL。

   在模板中，可通过将 AWS::CloudFormation::WaitConditionHandle 资源的逻辑名称传输至 Ref 内部函数，实现对预签名 URL 的检索。例如，您可以使用 AWS::EC2::Instance 资源的 UserData 属性将预签名 URL 传递给 Amazon EC2 实例，以使这些实例上运行的脚本或应用程序可以向 AWS CloudFormation 发送成功或失败信号：

```
"UserData" : {
   "Fn::Base64" : {
       "Fn::Join" : [ "", ["SignalURL=", { "Ref" : "myWaitHandle" } ] ]
   }
}
```

注意：在 AWS 管理控制台或 AWS CloudFormation 命令行工具中，预签名 URL 将显示为等待条件句柄资源的物理 ID。

4. 当堆栈进入等待条件后，选择检测方法。

   如果您在启用通知时创建堆栈，则 AWS CloudFormation 会针对每个堆栈事件将一个通知发布到指定主题。如果您或您的应用程序订阅了该主题，您可以监测针对等待条件句柄创建事件的通知，并通过通知消息来检索预签名 URL。

   您还可以使用 AWS 管理控制台、AWS CloudFormation 命令行工具或 AWS CloudFormation API 来监控堆栈事件。

5. 利用预签名 URL 发送成功或失败信号。

   您需要通过预签名 URL 发送 HTTP 请求消息之后，才能发送信号。请求方法必须为 PUT，并且内容类型标题必须为空字符串或省略。请求消息必须采用等待条件发送 JSON 格式 (p. 191)中指定的表单的 JSON 结构。

   您需要发送 Count 属性指定的数量的成功信号，才能使 AWS CloudFormation 继续创建堆栈。如果您的计数超过 1，则在发送至特定等待条件的所有信号中，每个信号的 UniqueId 值必须是唯一的。

   Curl 命令是发送信号的一种方式。以下示例显示了向等待条件发送成功信号的 Curl 命令行。

```
curl -T /tmp/a "https://cloudformation-waitcondition-test.s3.amazon
aws.com/arn%3Aaws%3Acloudformation%3Aus-east-1%3A034017226601%3Astack%2Fstack-
gosar-20110427004224-test-stack-with-WaitCondition--VEYW%2Fe498ce60-70a1-
11e0-81a7-5081d0136786%2FmyWaitConditionHandle?Expires=1303976584&AWSAccess
KeyId=AKIAIOSFODNN7EXAMPLE&Signature=ik1twT6hpS4cgNAw7wyOoRejVoo%3D"
```

其中文件 /tmp/a 包含以下 JSON 结构：

```
{
   "Status" : "SUCCESS",
   "Reason" : "Configuration Complete",
   "UniqueId" : "ID1234",
   "Data" : "Application has completed configuration."
}
```

此示例显示发送同一成功信号的 Curl 命令行，不同的是，它将 JSON 结构作为命令行参数发送。

```
curl -X PUT -H 'Content-Type:' --data-binary '{"Status" : "SUCCESS","Reason"
 : "Configuration Complete","UniqueId" : "ID1234","Data" : "Application has
 completed configuration."}'  https://cloudformation-waitcondition-
test.s3.amazonaws.com/arn%3Aaws%3Acloudformation%3Aus-east-
1%3A034017226601%3Astack%2Fstack-gosar-20110427004224-test-stack-with-Wait
Condition--VEYW%2Fe498ce60-70a1-11e0-81a7-5081d0136786%2FmyWaitCondition
Handle?Expires=1303976584&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Signa
ture=ik1twT6hpS4cgNAw7wyOoRejVoo%3D
```

等待条件发送 JSON 格式

当您发送等待条件的信号时，您必须使用以下 JSON 格式：

```
{
  "Status" : "StatusValue",
  "UniqueId" : "Some UniqueId",
  "Data" : "Some Data",
  "Reason" : "Some Reason"
  }
```

其中：

StatusValue 必须为以下值之一：

- *SUCCESS* 指示成功信号。
- *FAILURE* 指示失败信号，并触发失败的等待条件和堆栈回滚。

UniqueId 标识发送至 AWS CloudFormation 的信号。如果等待条件的 Count 属性大于 1，那么在针对特定等待条件发送的所有信号中，UniqueId 值必须唯一；否则，AWS CloudFormation 将认为此信号为先前发送的具有相同 UniqueId 的信号的重新传输，因此将忽略该信号。

Data 为您想要通过信号发送回的任何信息。可以通过调用模板中的 Fn::GetAtt 函数 (p. 502)来访问 Data 值。例如，如果您针对等待条件 mywaitcondition 创建以下输出值，则可以使用 `aws cloudformation describe-stacks` 命令、DescribeStacks 操作 或 CloudFormation 控制台的“输出”选项卡，来查看通过有效信号发送至 AWS CloudFormation 的 Data：

```
        "WaitConditionData" : {
            "Value" : { "Fn::GetAtt" : [ "mywaitcondition", "Data" ]},
            "Description" : "The data passed back as part of signalling the
WaitCondition"
        },
```

Fn::GetAtt 函数将返回 UniqueId 和 Data 作为 JSON 结构中的名称/值对。下面已定义 WaitConditionData 输出值返回的 Data 属性的示例：

```
{"Signal1":"Application has completed configuration."}
```

Reason 为字符串，除了要符合 JSON 格式外，对其内容无任何其他限制。

AWS CloudFormation 终端节点

为了缩短应用程序中的数据延迟，大多数 Amazon Web Services 产品都允许您选择区域终端节点来发出请求。终端节点是作为 Web 服务入口点的 URL。

AWS CloudFormation 终端节点包括：

区域	终端节点
美国东部（弗吉尼亚北部）区域	cloudformation.us-east-1.amazonaws.com

区域	终端节点
美国西部（俄勒冈）区域	cloudformation.us-west-2.amazonaws.com
美国西部（加利福尼亚北部）区域	cloudformation.us-west-1.amazonaws.com
欧洲（爱尔兰）区域	cloudformation.eu-west-1.amazonaws.com
亚太地区（新加坡）区域	cloudformation.ap-southeast-1.amazonaws.com
亚太地区（悉尼）区域	cloudformation.ap-southeast-2.amazonaws.com
亚太地区（东京）区域	cloudformation.ap-northeast-1.amazonaws.com
南美洲（圣保罗）区域	cloudformation.sa-east-1.amazonaws.com

Note

所有 AWS CloudFormation 终端节点均使用 HTTPS 协议进行访问。

有关 AWS CloudFormation 及其他服务的区域和终端节点的更多信息，请转至 *Amazon Web Services General Reference*中的区域和终端节点。

在 AWS CloudFormation 模板中使用正则表达式

正则表达式（通常称为 Regex）可在 AWS CloudFormation 模板中的多个位置指定，例如，在创建模板参数 (p. 101)时，可以指定用于 AllowedPattern 属性的正则表达式。

AWS CloudFormation 中的正则表达式遵循 Java 正则表达式语法。可在 Java 文档中查看该语法的完整描述及其构造，相应的网址为：java.util.regex.Pattern。

Important

由于 AWS CloudFormation 模板使用 JSON 语法来指定对象和数据，因此，您将需要为正则表达式中的任何反斜杠字符再添加一条反斜杠，否则 JSON 会将其解释为转义字符。

例如，如果您在正则表达式中包含 `\d` 以匹配数字字符，则您需要在模板中将其写为 `\\d`。

使用 AWS CloudFormation 和 Cloud-Init 自动化应用程序安装

作者：Chris Whitaker，2011 年 5 月

Abstract

通过将 WaitCondition 资源与 AWS CloudFormation 和 Cloud-Init 结合使用，在启动时自动启动并动态配置应用程序。

本部分将介绍如何将 Amazon Linux AMI 与 AWS CloudFormation 配合使用，以便在系统启动时启动和动态配置应用程序。示例使用了 AWS CloudFormation 支持的新 WaitCondition 资源，等到系统任务堆栈创建成功之后，再配置 Ruby on Rails 应用程序。该示例还利用了适用于 Cloud-init 的 Amazon Linux AMI 支持，而 Cloud-init 是按规范构建的开源应用程序。Cloud-init 使您能够使用 Amazon Elastic Compute Cloud (Amazon EC2) `UserData` 参数指定启动时要在实例上运行的操作。该实例将采用这一机制来指定应用程序配置命令，以及向等待条件发送成功信号，以便继续创建堆栈的命令。

Note

此外，也可以使用 AWS CloudFormation 帮助程序脚本（cfn-init 和 cfn-signal）自动安装应用程序。有关更多信息，请参阅 使用 AWS CloudFormation 部署应用程序 (p. 198)。

首先，让我们先创建一个简单的 Rails 应用程序。开始先键入命令：

```
$ rails new <path to application>
```

您必须先安装几个数据包和 RubyGem，然后才能在 Amazon Linux 上执行此命令。以下是创建简单应用程序所需的完整命令集：

```
#!/bin/bash -ex
yum -y install gcc-c++ make
yum -y install mysql-devel sqlite-devel
yum -y install ruby-rdoc rubygems ruby-mysql ruby-devel
gem install --no-ri --no-rdoc rails
gem install --no-ri --no-rdoc mysql
gem install --no-ri --no-rdoc sqlite3
rails new myapp
cd myapp
rails server -d
```

通常，您可以从 AWS 管理控制台启动运行 Amazon Linux AMI 的新 Amazon EC2 实例。在控制台中，您可以通过 SSH 连接实例，然后键入上方所列的命令以设置和运行 Rails 应用程序。

使用 AWS CloudFormation 创建应用程序可以为您带来很多好处：

1. 要使用创建的应用程序，您需要允许该程序访问 Amazon EC2 实例上的多个 TCP/IP 端口。特别要注意的是，您必须打开端口 3000 以便连接至 Rails 应用程序。您可能需要打开端口 22 以通过 SSH 连接实例，从而在其启动并运行时对其进行管理。要启用这些端口的访问权限，您需要创建一个 Amazon EC2 实例，并且具有一个正确配置的 EC2 安全组。通过 AWS CloudFormation，您可以在定义实例的同时定义 Amazon EC2 安全组，这让您可以将应用程序的整个 AWS 资源配置保存在某个位置。
2. 您可以使用 AWS CloudFormation 模板创建多个应用程序实例。每个实例保证都相同。所有应用程序配置和安装脚本都保存在统一位置。通过利用模板中的多个工具，您可以使用同一模板创建不同Amazon EC2 区域的应用程序实例。例如，您可以创建一个美国东部（弗吉尼亚北部）区域的实例和一个欧洲（爱尔兰）区域的实例，并且可以确保这两个应用程序的配置完全相同。
3. 通过使用 AWS CloudFormation 模板中的 WaitCondition 资源，您可以准确获悉应用程序何时可以接受流量。

以下代码显示了可用于在任何 Amazon EC2 区域创建和配置示例 Rails 应用程序的完整模板。

```
{
 "AWSTemplateFormatVersion" : "2010-09-09",
 "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the instance",
      "Type" : "String"
    }
  },

  "Mappings" : {
    "RegionMap" : {
```

```
      "us-east-1" : {"AMI" : "ami-8c1fece5"},
      "us-west-1" : {"AMI" : "ami-3bc9997e"},
      "eu-west-1" : {"AMI" : "ami-47cefa33"},
      "ap-southeast-1" : {"AMI" : "ami-6af08e38"},
      "ap-northeast-1" : {"AMI" : "ami-300ca731"}
    }
  },

  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
 }, "AMI" ]},
        "UserData" : { "Fn::Base64" : { "Fn::Join" : ["",[
            "#!/bin/bash -ex","\n",
            "yum -y install gcc-c++ make","\n",
            "yum -y install mysql-devel sqlite-devel","\n",
            "yum -y install ruby-rdoc rubygems ruby-mysql ruby-devel","\n",
            "gem install --no-ri --no-rdoc rails","\n",
            "gem install --no-ri --no-rdoc mysql","\n",
            "gem install --no-ri --no-rdoc sqlite3","\n",
            "rails new myapp","\n",
            "cd myapp","\n",
            "rails server -d","\n",
            "curl -X PUT -H 'Content-Type:' --data-binary '{\"Status\" :
 \"SUCCESS\",",
                                                              "\"Reason\" : \"The
 application myapp is ready\",",
                                                               "\"UniqueId\" :
 \"myapp\",",
                                                          "\"Data\" : \"Done\"}'
 ",
                  "\"", {"Ref" : "WaitForInstanceWaitHandle"},"\"\n" ]]}}
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable Access to Rails application via port 3000
 and SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : "0.0.0.0/0"
        }, {
          "IpProtocol" : "tcp",
          "FromPort" : "3000",
          "ToPort" : "3000",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    },
```

```
    "WaitForInstanceWaitHandle" : {
      "Type" : "AWS::CloudFormation::WaitConditionHandle",
      "Properties" : {
      }
    },

    "WaitForInstance" : {
      "Type" : "AWS::CloudFormation::WaitCondition",
      "DependsOn" : "Ec2Instance",
      "Properties" : {
        "Handle" : {"Ref" : "WaitForInstanceWaitHandle"},
        "Timeout" : "600"
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Description" : "The URL for the newly created Rails application",
      "Value" : { "Fn::Join" : ["", [ "http://", { "Fn::GetAtt" : [ "Ec2In
stance", "PublicIp" ] }, ":3000" ]]}
    }
  }
}
```

该模板假定您希望 SSH 可访问正在运行的 Amazon EC2 实例。它需要您输入账户中某个现有 Amazon EC2 密钥对的名称，将其作为创建堆栈时的输入参数：

```
"Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the instance",
      "Type" : "String"
    }
},
```

您键入的 *KeyName* 将在 Amazon EC2 实例资源定义中引用：

```
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
...
```

如果您目前还没有密钥对，则可以访问 AWS Management Console 并打开 Amazon EC2 控制台，从中创建一个新的密钥对。请记得保存创建的密钥文件，以便稍后使用它来以 SSH 方式连接实例。有关创建和使用 Amazon EC2 密钥对的更多信息，请参阅 *Amazon EC2 用户指南*中的获取 SSH 密钥对。

AMI ID 特定于区域，因此，要启动的实际 AMI 取决于堆栈创建时所在的区域。该模板使用“映射”功能基于区域选择适当的 AMI：

```
    "Mappings" : {
    "RegionMap" : {
      "us-east-1" : {"AMI" : "ami-8c1fece5"},
```

```
    "us-west-1" : {"AMI" : "ami-3bc9997e"},
    "eu-west-1" : {"AMI" : "ami-47cefa33"},
    "ap-southeast-1" : {"AMI" : "ami-6af08e38"},
    "ap-northeast-1" : {"AMI" : "ami-300ca731"}
  }
 },
```

要使用的实际 AMI 在 Amazon EC2 实例资源定义中通过使用 *FindInMap* 内部函数和虚拟参数 *AWS::Region*（返回表示要构建堆栈的区域的字符串）定义：

```
  "Ec2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
       ...
      "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
}, "AMI" ]},
       ...
    }
  },
```

为了打开用于访问 SSH 的端口（TCP/IP 端口 22）以及允许访问新创建的 Rails 应用程序的端口（TCP/IP 端口 3000），模板定义了新的安全组：

```
  "InstanceSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Enable Access to Rails application via port 3000
and SSH access via port 22",
      "SecurityGroupIngress" : [ {
        "IpProtocol" : "tcp",
        "FromPort" : "22",
        "ToPort" : "22",
        "CidrIp" : "0.0.0.0/0"
      }, {
        "IpProtocol" : "tcp",
        "FromPort" : "3000",
        "ToPort" : "3000",
        "CidrIp" : "0.0.0.0/0"
      } ]
    }
  },
```

Amazon EC2 实例应用程序中引用了该模板：

```
  "Ec2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
       ...
      "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
       ...
    }
  },
```

最后，要完成应用程序配置，您需要配置 Cloud-init 以安装所需的数据包和 RubyGem，然后启动应用程序。Cloud-init 使用 Amazon EC2 *UserData* 字段传递配置信息。如果 *UserData* 字段以 `#!` 开头，将认

为 *UserData* 的内容包含要在启动时执行的脚本。CloudFormation 模板中的所有文本都必须符合 JSON 结构要求，因此必须避开一些脚本。示例模板使用 *Fn::Base64* 函数对用户数据进行 base64 编码，并允许在运行时在脚本中替换模板中的参数和引用（在本例中为对 WaitConditionHandle 的引用）。此示例使用 *Fn::Join* 函数连接脚本的各个部分。

```
       "UserData" : { "Fn::Base64" : { "Fn::Join" : ["",[
           "#!/bin/bash -ex","\n",
           "yum -y install gcc-c++ make","\n",
           "yum -y install mysql-devel sqlite-devel","\n",
           "yum -y install ruby-rdoc rubygems ruby-mysql ruby-devel","\n",
           "gem install --no-ri --no-rdoc rails","\n",
           "gem install --no-ri --no-rdoc mysql","\n",
           "gem install --no-ri --no-rdoc sqlite3","\n",
           "rails new myapp","\n",
           "cd myapp","\n",
           "rails server -d","\n",
           "curl -X PUT -H 'Content-Type:' --data-binary '{\"Status\" :
\"SUCCESS\",",
                                                          "\"Reason\" : \"The
 application myapp is ready\",",
                                                           "\"UniqueId\" :
\"myapp\",",
                                                       "\"Data\" : \"Done\"}'
 ",
                "\"", {"Ref" : "WaitForInstanceWaitHandle"},"\"\n" ]]}}
```

因此，在已安装程序包并且运行应用程序之前，堆栈不会指示 *CREATE_COMPLETE*，我们将使用新的 WaitCondition 资源。您可以在上面的 Amazon EC2 实例资源定义中看到，*UserData* 脚本的最后一行是 *CURL* 命令，该命令可使用称为 WaitForInstanceWaitHandle 的 WaitConditionHandle 资源发送 WaitCondition 信号。

WaitCondition 本身的定义如下所示：

```
   "WaitForInstance" : {
     "Type" : "AWS::CloudFormation::WaitCondition",
     "DependsOn" : "Ec2Instance",
     "Properties" : {
       "Handle" : {"Ref" : "WaitForInstanceWaitHandle"},
       "Timeout" : "600"
     }
   }
```

WaitCondition 定义使用 *DependsOn* 结构。这可以确保 WaitForInstance WaitCondition 资源仅在 EC2 实例资源创建后直接创建。为什么这一点很重要？WaitCondition 中指定的 Timeout 值（在本例中为 600 秒）会在 WaitCondition 对象置于 *CREATE_IN_PROGRESS* 状态时开始计时。在本模板中，我们希望为 Ruby 应用程序留出一些启动时间，但时间不能过长，以防实例出现不好的情况。使 WaitCondition 与 Amazon EC2 实例相关后，只有当 EC2 实例进入 EC2 运行状态，且 Cloud-init 脚本启动时，才会创建 WaitCondition 资源。使用 *DependsOn* 可确保配置脚本具有 600 秒的运行时间。如果脚本没有通过 *CURL* 命令发送信号，则堆栈创建会在 WaitCondition 超时触发时失败。

注意：WaitCondition 资源可用于同步模板中其他资源的创建，而不仅仅是堆栈创建。例如，您可以选择在应用程序运行前，不将实例与弹性 IP 地址关联。通过为模板中引用 WaitCondition 的其他资源添加 DependsOn 子句，您可以确保 WaitCondition 不会在 WaitCondition 信号发出之前创建。阅读更多有关了解如何从模板中的应用程序传回数据的文章，以及在应用程序被视为正常运行之前，如何使用 WaitCondition 对象等待多个实例启动并运行。

如果要下载、修改或试用 Rails 示例模板，可从 AWS CloudFormation 示例模板中获得。

相关资源

- *Amazon Elastic Compute Cloud 用户指南*中的 Amazon EC2 AMI 基础知识
- *Amazon Elastic Compute Cloud 用户指南*中的 Amazon Linux AMI 基础知识
- 使用 AWS CloudFormation 部署应用程序 (p. 198)
- Ubuntu CloudInit documentation
- Using Amazon's CloudFormation, cloud-init, chef and fog to automate infrastructure

使用 AWS CloudFormation 部署应用程序

Topics

AWS CloudFormation 部署应用程序概述

您可以使用 AWS CloudFormation 来自动安装、配置和启动应用程序。执行此操作可帮助您节省许多时间和工作量。例如，您可以用许多方法来部署 Amazon 实例，使其能够运行 LAMP（Linux、Apache、MySQL 和 PHP）Web 服务器，如下所示：

- 手动安装。手动安装可靠但也耗时。您可以通过选择包含所有必要软件的 AMI 来减少工作量，但是您仍然要连接产生的 Amazon EC2 实例，以便完成配置过程并启动应用程序。
- Cloud-init。最新的 Linux AMI 中包含 cloud-init，一种根据您提供的脚本配置和启动您的应用程序的开源程序。如果您要更改配置或安装新应用程序或更新，则必须连接到实例并手动进行更改。

使用 AWS CloudFormation 来自动执行应用程序部署有两项优势。首先，当所有安装、配置和启动命令都包含在 CloudFormation 模板中时，更容易复制部署。其次，您可以在不直接连接实例的情况下，更新现有安装。

AWS CloudFormation 包含一组基于 cloud-init 的帮助应用程序（即：cfn-init、cfn-signal、cfn-get-metadata 和 cfn-hup）。这些帮助应用程序不仅提供类似于 cloud-init 功能性，还允许您在实例和应用程序正在运行时更新元数据。您可以在部署后更新元数据，因为 AWS CloudFormation 存储元数据。以这种方式提升灵活性确实需要执行一些额外设置，即您需要为实例创建安全证书，以便实例可以调用 CloudFormation API 来检索已更新的元数据。

以下部分说明了如何创建可说明 LAMP 堆栈并使用 cfn-init 来安装、配置以及启动 Apache、MySQL 和 PHP 的模板。我们将从用于设置运行 Linux 的基础 Amazon EC2 实例的简单模板入手，然后继续向模板添加内容，直至部署说明中的说明为 LAMP 堆栈已满为止。

使用 CloudFormation 创建基础 Amazon EC2 实例

我们从基础模板开始，该模板定义了带有安全组的单个 Amazon EC2 实例，此安全组允许 SSH 流量通过端口 22，HTTP 流量通过端口 80。该模板包含五个顶级 JSON 对象：

- 说明—EC2_Single_Instance

EC2_Single_Instance 可指定一个 EC2 实例及相应的安全组，可支持 SSH 和 HTTP 访问。

- 参数—KeyName 和 InstanceType

  KeyName 参数指定要用于 SSH 访问的 EC2 密钥对，而 InstanceType 参数指定 EC2 实例的类型。
- 映像—AWSInstanceType2Arch 和 AWSRegionArch2AMI

  AWSInstanceType2Arch 可将适合的架构变换到实例大小，因此模板用户只需指定实例大小即可。AWSRegionArch2AMI 可将 AMI ID 映射到各自的特定区域，这样模板用户便不用搜索各区域内可用的 AMI ID。
- 资源—WebServer 和 WebServerSecurityGroup

  WebServer 资源定义了 Amazon EC2 实例，而 WebServerSecurityGroup 定义了允许传入流量通过端口 22 (SSH) 和端口 80 (HTTP) 的安全组。
- 输出—WebsiteURL

  WebsiteURL 会返回新创建网站的公有 URL。

此处为模板：

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template EC2_Single_Instance:
Create a single EC2 instance. **WARNING** This template creates an Amazon EC2
instance. You will be billed for the AWS resources used if you create a stack
from this template.",

  "Parameters" : {

    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the instances",
      "Type" : "String",
      "MinLength": "1",
      "MaxLength": "64",
      "AllowedPattern" : "[-_ a-zA-Z0-9]*",
      "ConstraintDescription" : "can contain only alphanumeric characters,
spaces, dashes and underscores."
    },

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro", "m1.small", "m1.large", "m1.xlarge",
"m2.xlarge", "m2.2xlarge", "m2.4xlarge", "c1.medium", "c1.xlarge" ],
      "ConstraintDescription" : "must be a valid EC2 instance type."
    }

  },

  "Mappings" : {
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "32" },
      "m1.small"    : { "Arch" : "32" },
      "m1.large"    : { "Arch" : "64" },
```

```
      "m1.xlarge"   : { "Arch" : "64" },
      "m2.xlarge"   : { "Arch" : "64" },
      "m2.2xlarge"  : { "Arch" : "64" },
      "m2.4xlarge"  : { "Arch" : "64" },
      "c1.medium"   : { "Arch" : "32" },
      "c1.xlarge"   : { "Arch" : "64" }
    },
    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "32" : "ami-7f418316", "64" : "ami-7341831a" },
      "us-west-1"      : { "32" : "ami-951945d0", "64" : "ami-971945d2" },
      "us-west-2"      : { "32" : "ami-16fd7026", "64" : "ami-10fd7020" },
      "eu-west-1"      : { "32" : "ami-24506250", "64" : "ami-20506254" },
      "sa-east-1"      : { "32" : "ami-3e3be423", "64" : "ami-3c3be421" },
      "ap-southeast-1" : { "32" : "ami-74dda626", "64" : "ami-7edda62c" },
      "ap-northeast-1" : { "32" : "ami-dcfa4edd", "64" : "ami-e8fa4ee9" }
    }
  },

  "Resources" : {

    "WebServer": {
      "Type": "AWS::EC2::Instance",

      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                        { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref"
 : "InstanceType" }, "Arch" ] } ] },
        "InstanceType"   : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "KeyName"        : { "Ref" : "KeyName" }

      }
    },


    "WebServerSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80",
        "SecurityGroupIngress" : [
          {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp"
: "0.0.0.0/0"},
          {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp"
: "0.0.0.0/0"}
        ]
      }
    }
  },

  "Outputs" : {
    "WebsiteURL" : {
      "Value" : { "Fn::Join" : ["", ["http://", { "Fn::GetAtt" : [ "WebServer",
 "PublicDnsName" ]}]] },
      "Description" : "URL for newly created EC2 Instance"
    }
  }
}
```

使用 CloudFormation 自动执行 LAMP 安装

在此部分中，我们将基于基础 Amazon EC2 模板创建可自动安装 Apache、MySQL 和 PHP 的 LAMP 模板。在稍后的部分中，我们将配置和启动应用程序。在向模板添加必要的元数据和 shell 命令之前，我们需要设置有权读取 AWS CloudFormation 元数据的 IAM 用户。

为了从在模板中创建的 Amazon EC2 实例读取元数据，我们建议您在模板中创建一个 IAM 用户，并将 AWS 访问密钥 ID 和秘密访问密钥作为元数据传递到实例。授予 IAM 用户仅调用 DescribeStackResource 操作的权限。创建 IAM 用户会锁定账户，因此它只能检索数据，而无法执行其他操作。通过使用模板中的 IAM 功能，可将用户的存续与堆栈的使用寿命捆绑起来。每个新堆栈都有一个单独的唯一用户。有关在 AWS CloudFormation 中使用 IAM 的更多信息，请参阅使用 AWS Identity and Access Management 控制访问 (p. 59)。

更新后的模板包含几个新部分。此模板创建的堆栈安装了 Apache、MySQL 和 PHP，但是不会配置或启动其中的任意一个。新的模板部分如下所示：

- 元数据密钥 —`AWS::CloudFormation::Init`

  Cfn-init 可读取并安装此密钥中列出的软件包（例如，httpd、mysql 和 php）。Cfn-init 还检索并扩展按来源列出的文件。
- 属性资源 - `UserData`

  UserData 密钥允许您执行 shell 命令。此模板发出两个 shell 命令：第一个命令用于安装 AWS CloudFormation 帮助程序脚本；第二个命令用于执行 cfn-init 脚本。

以下示例模板可创建自动安装 Apache、MySQL 和 PHP 的 LAMP 模板。为了简洁，将删除所有用省略号 (...) 标记的部分。模板的新增部分以红色斜体文本显示。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template LAMP_Install_Only: ...",


  "Parameters" : {

    "KeyName" : { ... },

    "InstanceType" : { ...  },

  "Mappings" : { ... },

  "Resources" : {
    "WebServer": {
      "Type": "AWS::EC2::Instance",
      "Metadata" : {
       "Comment1" : "Configure the bootstrap helpers to install the Apache Web
 Server and PHP",
       "Comment2" : "The website content is downloaded from the CloudFormation
PHPSample.zip file",

        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "mysql"        : [],
```

```
                "mysql-server" : [],
                "mysql-libs"   : [],
                "httpd"        : [],
                "php"          : [],
                "php-mysql"    : []
              }
            },

            "sources" : {
              "/var/www/html" : "https://s3.amazonaws.com/cloudformation-ex
amples/CloudFormationPHPSample.zip"
            }
          }
        }
      },
      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                          { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref"
 : "InstanceType" }, "Arch" ] } ] },
        "InstanceType"   : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "KeyName"        : { "Ref" : "KeyName" },
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
          "#!/bin/bash -v\n",
          "yum update -y aws-cfn-bootstrap\n",

          "# Install LAMP packages\n",
          "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" }, " -r Web
Server ",
          "    --region ", { "Ref" : "AWS::Region" }, " || error_exit 'Failed
to run cfn-init'\n"
    ]]}}
      }
    },


    "WebServerSecurityGroup" : { ...  }
  },

  "Outputs" : { ... }
}
```

使用 CloudFormation 自动执行 LAMP 配置

在此部分中，我们将使用 AWS CloudFormation 来自动配置并启动 Apache、MySQL 和 PHP。

- 参数 - `DBName`、`DBUsername`、`DBPassword` 和 `DBRootPassword`

  这些参数允许模板用户指定数据库名称和密码。此参数还包含各种限制，可在开始创建堆栈之前，抓取格式不正确的值。
- 元数据 —`files` 和 `services`

  `files` 元数据密钥用于创建 MySQL 设置文件。服务元数据密钥确保 httpd 和 mysqld 服务不仅在 cfn-init 结束时继续运行（ensureRunning 设为 true），而且当重启时这些服务也会随之重启（enabled 设备为 true）。
- 属性资源 - `UserData`

UserData 密钥包含用于创建数据库和用户的 MySQL shell 命令，现在这些命令紧跟在对 cfn-init 的调用之后。另外，shell 命令可配置 PHP 以遵循 MySQL shell 命令。

以下示例模板可创建 LAMP 堆栈、自动安装 Apache、MySQL 和 PHP，然后配置并启动每个服务。为了简洁，将删除所有用省略号 (...) 标记的部分。模板的新增部分以红色斜体文本显示。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template LAMP_Single_Instance:
Create a LAMP stack using a single EC2 instance and a local MySQL database for
 storage. This template demonstrates using the AWS CloudFormation bootstrap
scripts to install the packages and files necessary to deploy the Apache web
server, PHP and MySQL at instance launch time. **WARNING** This template creates
 an Amazon EC2 instance. You will be billed for the AWS resources used if you
create a stack from this template.",

  "Parameters" : {

    "KeyName" : { ...  },

    "DBName": {
      "Default": "MyDatabase",
      "Description" : "MySQL database name",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "64",
      "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*",
      "ConstraintDescription" : "must begin with a letter and contain only al
phanumeric characters."
    },

    "DBUsername": {
      "NoEcho": "true",
      "Description" : "Username for MySQL database access",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "16",
      "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*",
      "ConstraintDescription" : "must begin with a letter and contain only al
phanumeric characters."
    },

    "DBPassword": {
      "NoEcho": "true",
      "Description" : "Password for MySQL database access",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "41",
      "AllowedPattern" : "[a-zA-Z0-9]*",
      "ConstraintDescription" : "must contain only alphanumeric characters."
    },

    "DBRootPassword": {
      "NoEcho": "true",
      "Description" : "Root password for MySQL",
      "Type": "String",
```

```
      "MinLength": "1",
      "MaxLength": "41",
      "AllowedPattern" : "[a-zA-Z0-9]*",
      "ConstraintDescription" : "must contain only alphanumeric characters."
    },

    "InstanceType" : { ...  }

  },

  "Mappings" : {
  ...
  },

  "Resources" : {
    "WebServer": {
      "Type": "AWS::EC2::Instance",
      "Metadata" : {
       "Comment1" : "Configure the bootstrap helpers to install the Apache Web
 Server and PHP",
       "Comment2" : "The website content is downloaded from the CloudFormation
PHPSample.zip file",

        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "mysql"        : [],
                "mysql-server" : [],
                "mysql-libs"   : [],
                "httpd"        : [],
                "php"          : [],
                "php-mysql"    : []
              }
            },

            "sources" : {
              "/var/www/html" : "https://s3.amazonaws.com/cloudformation-ex
amples/CloudFormationPHPSample.zip"
            },

            "files" : {
              "/tmp/setup.mysql" : {
                "content" : { "Fn::Join" : ["", [
                  "CREATE DATABASE ", { "Ref" : "DBName" }, ";\n",
                  "GRANT ALL ON ", { "Ref" : "DBName" }, ".* TO '", { "Ref" :
"DBUsername" }, "'@localhost IDENTIFIED BY '", { "Ref" : "DBPassword" }, "';\n"

                ]]},
                "mode"  : "000644",
                "owner" : "root",
                "group" : "root"
              }
            },

            "services" : {
              "sysvinit" : {
                "mysqld" : {
```

```
                "enabled"       : "true",
                "ensureRunning" : "true"
              },
              "httpd" : {
                "enabled"       : "true",
                "ensureRunning" : "true"
              }
            }

          }
        }
      }
    },
    "Properties": {
      "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                        { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref"
 : "InstanceType" }, "Arch" ] } ] },
      "InstanceType"   : { "Ref" : "InstanceType" },
      "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
      "KeyName"        : { "Ref" : "KeyName" },
      "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
        "#!/bin/bash -v\n",
        "yum update -y aws-cfn-bootstrap\n",

        "# Install LAMP packages\n",
        "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" }, " -r Web
Server ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n",

        "# Setup MySQL, create a user and a database\n",
        "mysqladmin -u root password '", { "Ref" : "DBRootPassword" }, "\n",

        "mysql -u root --password='", { "Ref" : "DBRootPassword" }, "\n",

       "# Configure the PHP application - in this case, fix up the page with
 the right references to the database\n",
        "sed -i \"s/REPLACE_WITH_DATABASE/localhost/g\" /var/www/html/in
dex.php\n",
        "sed -i \"s/REPLACE_WITH_DBUSER/", { "Ref" : "DBUsername" }, "/g\"
/var/www/html/index.php\n",
       "sed -i \"s/REPLACE_WITH_DBPASSWORD/", { "Ref" : "DBPassword" }, "/g\"
 /var/www/html/index.php\n"

      ]]}}
    }
  },

  "WebServerSecurityGroup" : { ... }
},

"Outputs" : { ... }
}
```

使用 AWS CloudFormation 等待条件

到目前为止，该模板将创建堆栈资源并尝试启动所有指定资源。此时，它会将堆栈的状态标记为 CREATE_COMPLETE，即使一个或多个服务启动失败。为防止在成功启动所有服务之前状态更改为 CREATE_COMPLETE，AWS CloudFormation 支持您使用 WaitCondition 资源来确认是否已成功创建所有堆栈资源还是完全没有创建。在满足指定条件或超时之前，WaitCondition 资源会暂停执行模板。

为了等待应用程序就绪，我们可以扩展之前的模板，以创建 WaitCondition 和 WaitConditionHandle，并使用 cfn-signal 帮助程序脚本发送应用程序已安装的信号。WaitConditionHandle 资源是信号发送机制，用于通知 WaitCondition 资源的状态变化，后者决定部署是否可以继续或必须回滚。有关等待条件的更多信息，请参阅在模板中创建等待条件 (p. 188)。

为了添加 WaitCondition 和 WaitConditionHandle 资源，我们将以下内容添加到模板：

- 资源 > WebServer > 属性 —`UserData error_exit` 帮助程序函数

  error_exit 函数通过发送出错信号的退出代码调用 cfn-signal。如果 cfn-init 失败或设置 MySQL 的尝试失败，UserData 脚本会调用 error_exit。
- 资源 > WebServer > 属性 —`UserData cfn-signal`（成功时）

  如果所有服务均已成功配置并启动，则 UserData 脚本会通过发送成功信号的退出代码调用 cfn-signal。
- 资源 > WaitHandle —`AWS::CloudFormation::WaitConditionHandle`

  `AWS::CloudFormation::WaitConditionHandle` 密钥用于创建用作信号发送机制的预签名 URL。
- 资源 > WaitCondition —`AWS::CloudFormation::WaitCondition`

  `AWS::CloudFormation::WaitCondition` 密钥用于创建特殊资源，该资源可在指定的一段时间内等待成功信号。如果在指定时间段内没有收到成功信号，等待条件的状态会变为 CREATE_FAILED，且整个堆栈会回滚。

现在，此模板如下所示：模板的新增部分以红色斜体文本显示。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template LAMP_Single_Instance:
...",

  "Parameters" : { ... },

  "Mappings" : { ... },

  "Resources" : {
    "WebServer": {
      "Type": "AWS::EC2::Instance",
      "Metadata" : { ...
      },
      "Properties": {
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
                        { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref"
 : "InstanceType" }, "Arch" ] } ] },
        "InstanceType"   : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ {"Ref" : "WebServerSecurityGroup"} ],
        "KeyName"        : { "Ref" : "KeyName" },
        "UserData"       : { "Fn::Base64" : { "Fn::Join" : ["", [
```

```
          "#!/bin/bash -v\n",
          "yum update -y aws-cfn-bootstrap\n",

          "# Helper function\n",
          "function error_exit\n",
          "{\n",
          "  /opt/aws/bin/cfn-signal -e 1 -r \"$1\" '", { "Ref" : "WaitHandle"
 }, "'\n",
          "  exit 1\n",
          "}\n",

          "# Install LAMP packages\n",
          "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" }, " -r Web
Server ",
          "    --region ", { "Ref" : "AWS::Region" },
          " || error_exit 'Failed to run cfn-init'\n",

          "# Setup MySQL, create a user and a database\n",
          "mysqladmin -u root password '", { "Ref" : "DBRootPassword" },
          "' || error_exit 'Failed to initialize root password'\n",
          "mysql -u root --password='", { "Ref" : "DBRootPassword" },
        "' < /tmp/setup.mysql || error_exit 'Failed to initialize database'\n",


         "# Configure the PHP application - in this case, fix up the page with
 the right references to the database\n",
          "sed -i \"s/REPLACE_WITH_DATABASE/localhost/g\" /var/www/html/in
dex.php\n",
          "sed -i \"s/REPLACE_WITH_DBUSER/", { "Ref" : "DBUsername" }, "/g\"
/var/www/html/index.php\n",
         "sed -i \"s/REPLACE_WITH_DBPASSWORD/", { "Ref" : "DBPassword" }, "/g\"
 /var/www/html/index.php\n",

          "# All is well so signal success\n",
          "/opt/aws/bin/cfn-signal -e 0 -r \"LAMP Stack setup complete\" '", {
 "Ref" : "WaitHandle" }, "'\n"
        ]]}}
      }
    },

    "WaitHandle" : {
      "Type" : "AWS::CloudFormation::WaitConditionHandle"
    },

    "WaitCondition" : {
      "Type" : "AWS::CloudFormation::WaitCondition",
      "DependsOn" : "WebServer",
      "Properties" : {
        "Handle" : {"Ref" : "WaitHandle"},
        "Timeout" : "300"
      }
    },

    "WebServerSecurityGroup" : { ...
    }
  },

  "Outputs" : {
```

```
    "WebsiteURL" : { ...
    }
  }
}
```

可在以下网址获取完整的模板：

https://s3.amazonaws.com/cloudformation-templates-us-east-1/LAMP_Single_Instance.template

在 AWS CloudFormation 上使用 Microsoft Windows 堆栈

通过 AWS CloudFormation，您可以基于 Amazon EC2 Windows 亚马逊系统映像 (AMI) 创建 Microsoft Windows 堆栈，还可以安装软件、使用远程桌面访问您的堆栈，以及更新和配置您的堆栈。

本部分中的主题旨在演示如何通过 AWS CloudFormation 完成与 Windows 实例创建和管理相关的常见任务。

在本章节中

Microsoft Windows 亚马逊系统映像 (AMI) 和 AWS CloudFormation 模板

Abstract

列出能够用于创建 Microsoft Windows 堆栈的可用预配置 AWS CloudFormation 模板。

通过 AWS CloudFormation，您可以创建 Microsoft Windows 堆栈，以运行 Windows 服务器实例。可直接从 “AWS CloudFormation 实例模板”页启动若干个预配置模板，例如以下模板：

- Windows_Single_Server_SharePoint_Foundation.template - 在 Microsoft Windows Server® 2008 R2 上运行的 SharePoint® Foundation 2010
- Windows_Single_Server_Active_Directory.template - 创建运行在 Microsoft Windows Server® 2008 R2 上的 Active Directory 的单一服务器安装。
- Windows_Roles_And_Features.template - 创建单一服务器，指定 Microsoft Windows Server® 2008 R2 上运行的服务器角色。
- ElasticBeanstalk_Windows_Sample.template - 在运行 IIS 7.5 的 Windows Server 2008 R2 上启动 AWS Elastic Beanstalk 示例应用程序。

Note

Microsoft、Windows Server 和 SharePoint 是 Microsoft 公司集团的商标。

尽管上述堆栈已配置，您仍可以使用任意 EC2 Windows AMI 作为 AWS CloudFormation Windows 堆栈的基础。

启动 AWS CloudFormation Windows 堆栈

Abstract

使用 AWS CloudFormation 可引导 Windows 堆栈。

本主题将介绍如何引导 Windows 堆栈并排除堆栈创建问题。如果您要创建自己的用于 CloudFormation 的 Windows 映像，请参阅 *Amazon EC2 Microsoft Windows 指南*中的使用 EC2ConfigService 配置 Windows 实例来获得相关说明。您必须使用 EC2ConfigService 设置 Windows 实例，才能使该实例与 AWS CloudFormation 引导工具一起工作。

Topics

启动 Windows 堆栈的示例

出于举例说明的目的，我们将查看 AWS CloudFormation 单实例 Sharepoint 服务器模板，可以在以下 URL 对该模板进行整体查看：

- https://s3.amazonaws.com/cloudformation-templates-us-east-1/Windows_Single_Server_SharePoint_Foundation.template

本示例将展示如何：

- 创建访问实例的 IAM 用户和安全组
- 配置初始化文件：`cfn-credentials`、`cfn-hup.conf` 和 `cfn-auto-reloader.conf`
- 下载程序包，例如 Sharepoint Foundation 2010，并将其安装在服务器实例上。
- 使用 WaitCondition 确定资源准备就绪
- 通过 Amazon 弹性 IP (EIP) 检索实例的 IP。

AWS CloudFormation 帮助程序脚本 `cfn-init` 将用于根据 Windows Single Server Sharepoint Foundation 模板上 AWS::CloudFormation::Init (p. 241) 资源中的信息，执行上述各项操作。

AWS::CloudFormation::Init 部分名称为“SharePointFoundation”，并通过标准声明开始：

```
"SharePointFoundation": {
   "Type" : "AWS::EC2::Instance",
   "Metadata" : {
     "AWS::CloudFormation::Init" : {
       "config" : {
```

此后，将声明 AWS::CloudFormation::Init 的 files (文件) 部分：

```
"files" : {
  "c:\\cfn\\cfn-hup.conf" : {
    "content" : { "Fn::Join" : ["", [
      "[main]\n",
      "stack=", { "Ref" : "AWS::StackName" }, "\n",
      "region=", { "Ref" : "AWS::Region" }, "\n"
      ]]}
  },
  "c:\\cfn\\hooks.d\\cfn-auto-reloader.conf" : {
    "content": { "Fn::Join" : ["", [
      "[cfn-auto-reloader-hook]\n",
      "triggers=post.update\n",
     "path=Resources.SharePointFoundation.Metadata.AWS::CloudFormation::Init\n",

      "action=cfn-init.exe -v -s ", { "Ref" : "AWS::StackName" },
                                    " -r SharePointFoundation",
                                    " --region ", { "Ref" : "AWS::Region" },
"\n"
    ]]}
  },
  "C:\\SharePoint\\SharePointFoundation2010.exe" : {
    "source" : "http://d3adzpja92utk0.cloudfront.net/SharePointFoundation.exe"

  }
},
```

将在此处创建三个文件，并将其置于服务器实例上的 C:\cfn 目录中。它们是：

- cfn-hup.conf，cfn-hup 的配置文件。
- cfn-auto-reloader.conf，为钩子的配置文件，当 AWS::CloudFormation::Init 中的元数据更改时，cfn-hup 将用该配置文件初始化更新（调用 cfn-init）。

此外，还有一个下载到服务器的文件：SharePointFoundation.exe.该文件用于将 SharePoint 安装在服务器实例上。

Important

由于 Windows 上的路径使用反斜杠 ('\') 字符，因此，您必须始终记住，每当您引用 AWS CloudFormation 模板中的 Windows 路径时，都应通过预置另一反斜杠来正确地将所有反斜杠转义。

下面是 commands (命令) 部分，它将如下所示：

```
"commands" : {
  "1-extract" : {
    "command" : "C:\\SharePoint\\SharePointFoundation2010.exe /extract:C:\\Share
Point\\SPF2010 /quiet /log:C:\\SharePoint\\SharePointFoundation2010-extract.log"

  },
  "2-prereq" : {
    "command" : "C:\\SharePoint\\SPF2010\\PrerequisiteInstaller.exe /unattended"

  },
```

```
  "3-install" : {
    "command" : "C:\\SharePoint\\SPF2010\\setup.exe /config C:\\Share
Point\\SPF2010\\Files\\SetupSilent\\config.xml"
  }
```

由于实例中的命令会*按名称的字母顺序*进行处理，因此，每个命令前已预置数字来指示其所需的执行顺序。因此，我们能够确保首先提取安装程序包，随后安装所有系统必备项，最后，开始安装 SharePoint。

下面是 Properties (属性) 部分：

```
"Properties": {
  "InstanceType" : { "Ref" : "InstanceType" },
  "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region"
 },
              { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "Instan
ceType" }, "Arch" ] } ] },
  "SecurityGroups" : [ {"Ref" : "SharePointFoundationSecurityGroup"} ],
  "KeyName" : { "Ref" : "KeyPairName" },
  "UserData" : { "Fn::Base64" : { "Fn::Join" : ["", [
    "<script>\n",

    "cfn-init.exe -v -s ", { "Ref" : "AWS::StackName" },
    " -r SharePointFoundation",
    " --region ", { "Ref" : "AWS::Region" }, "\n",

    "cfn-signal.exe -e %ERRORLEVEL% ", { "Fn::Base64" : { "Ref" : "SharePoint
FoundationWaitHandle" }}, "\n",

    "</script>"
    ]]}}
  }
```

在此部分中，UserData 属性包含 `cmd.exe` 脚本，该脚本由 <script> 标签括起，将由 `cfn-init` 执行。您可以通过将脚本用 <powershell> 标签括起，来在此处改用 Windows Powershell 脚本。对于 Windows 堆栈，您必须再次对等待条件句柄 URL 进行 base64 编码。

此处引用了 SharePointFoundationWaitHandle，并将其与 `cfn-signal` 一起运行。模板的下一部分中声明了 WaitConditionHandle 和关联的 WaitCondition：

```
"SharePointFoundationWaitHandle" : {
   "Type" : "AWS::CloudFormation::WaitConditionHandle"
},

"SharePointFoundationWaitCondition" : {
   "Type" : "AWS::CloudFormation::WaitCondition",
   "DependsOn" : "SharePointFoundation",
   "Properties" : {
     "Handle" : {"Ref" : "SharePointFoundationWaitHandle"},
     "Timeout" : "3600"
   }
}
```

由于执行所有步骤和安装 SharePoint 需要一定的时间，但无需一个小时，因此在超时前，WaitCondition 将等待一个小时 (3600) 秒。

如果一切顺利，弹性 IP 将用于提供对 SharePoint 实例的访问权限：

```
"Outputs" : {
 "SharePointFoundationURL" : {
   "Value" : { "Fn::Join" : ["", ["http://", { "Ref" : "SharePointFoundationEIP"
 } ]] },
   "Description" : "SharePoint Team Site URL. Please retrieve Administrator
password of the instance and use it to access the URL"
 }
```

堆栈创建完成后，由 EIP 提供的 IP 地址将显示在 AWS CloudFormation 控制台的 Outputs (输出) 选项卡中。但是，访问实例前，您需要检索为实例而自动生成的临时管理员密码。访问 AWS CloudFormation Windows 实例 (p. 213)主题中提供了如何执行此操作的相关说明。

如何管理 Windows 服务

除了使用 `windows` 密钥而不使用 `sysvinit` 之外，管理 Linux 服务的方式与管理 Windows 服务的方式相同。以下示例将启动 `cfn-hup` 服务，将其设置为“自动”，并在 cfn-init 修改了 `c:\cfn\cfn-hup.conf` 或 `c:\cfn\hooks.d\cfn-auto-reloader.conf` 配置文件时重启服务。

```
"services" : {
  "windows" : {
    "cfn-hup" : {
      "enabled" : "true",
      "ensureRunning" : "true",
      "files" : ["c:\\cfn\\cfn-hup.conf", "c:\\cfn\\hooks.d\\cfn-auto-reload
er.conf"]
    }
  }
}
```

您可以通过使用名称（而非显示名称）引用服务，来以相同方式管理其他 Windows 服务。

如何解决堆栈创建故障问题

如果您的堆栈在创建中出现问题，默认行为将为失败时回滚。正常情况下，这属于一项良好默认，因为它免除了不必要的收费，但是它会让调试堆栈创建失败的原因变得非常困难。

要关闭此行为，请在使用 AWS CloudFormation 控制台创建堆栈时，单击 Show Advanced Options (显示高级选项)，然后单击 Rollback on failure (失败时回滚) 旁边的 No (否) 选择器。通过此操作，您可以登录您的实例，并查看日志文件，并精确查找运行启动脚本时出现的问题。

须查看的重要日志文件包括：

- EC2 配置日志位于：`C:\Program Files\Amazon\Ec2ConfigService\Logs\Ec2ConfigLog.txt`
- cfn-init 日志位于： `C:\cfn\log\cfn-init.log`

访问 AWS CloudFormation Windows 实例

在 AWS CloudFormation 上成功创建 Microsoft Windows 堆栈后，您就可以通过远程桌面登录您的实例，对其进行手动配置。此操作包括若干步骤：

1. 查找您 Windows 实例的物理 ID。
2. 使用物理 ID 从 Amazon EC2 中检索登录证书。
3. 使用登录证书，通过远程桌面 Remote Desktop 访问您的实例。

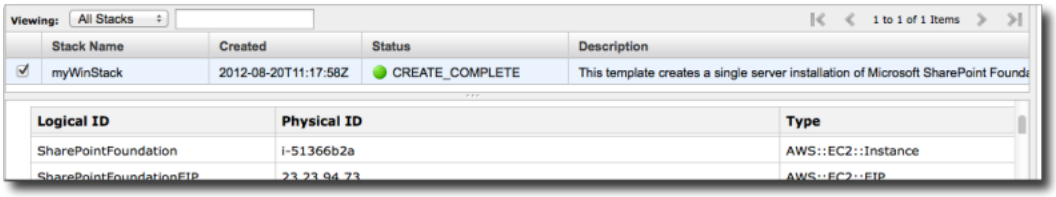
Note

在开始之前，您需要运行 AWS CloudFormation Windows 堆栈，还需要在创建实例时所用密钥对的私有密钥。有关生成 Amazon EC2 密钥对的信息，请参阅创建 EC2 密钥对 (p. 84)。

要检索 AWS CloudFormation Windows 实例的物理 ID，请执行以下操作：

1. 在 AWS CloudFormation 控制台上，单击基于 Windows 的堆栈。窗口下面的窗格中，将显示您的堆栈信息。
2. 单击 Resources (资源) 选项卡，然后查找 AWS::EC2::Instance (p. 272) 的 Physical ID (物理 ID)。它将如下所示：`i-51366b2a`.

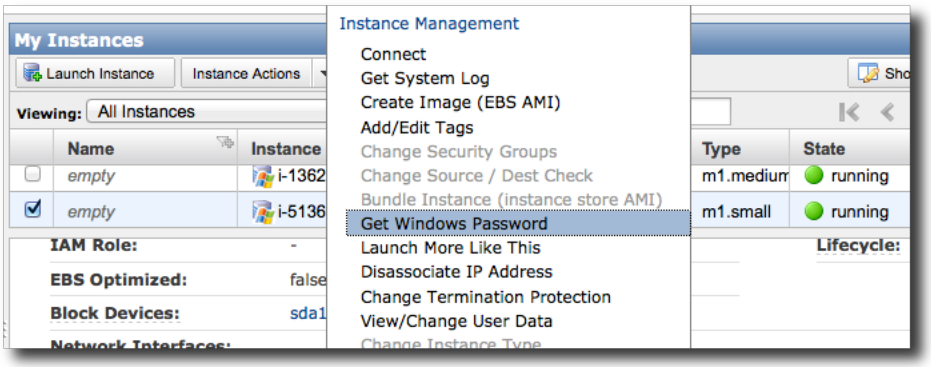
   如果您运行多项实例，那么您可能需要记住您实例的物理 ID 或将其写下来。您需要用它来恢复管理员密码，以便登录您的实例。



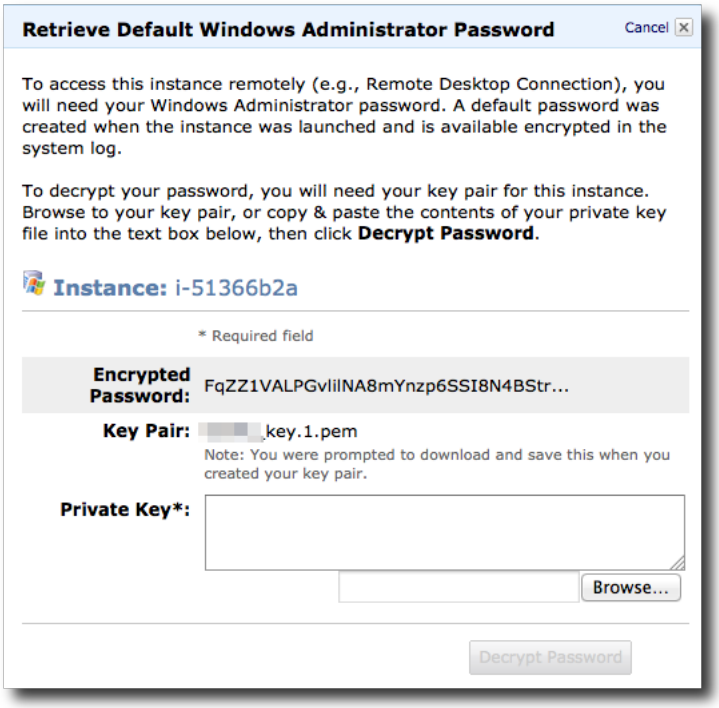
拥有实例的物理 ID 之后，您就可以用该其来检索管理员密码了。

为您 Windows 实例检索的管理员密码：

1. 在 AWS CloudFormation 控制台的左上角，单击 Services (服务)，然后单击 EC2。这会将您转到 Amazon EC2 Console Dashboard (Amazon EC2 控制台控制面板)。
2. 在 Navigation Bar (导航栏) 上，单击 Instances (实例)。这将显示标题为 My Instances (我的实例) 的列表。
3. 在列表中，通过物理 ID 查找您的实例。找到您的实例后，右击列表上的条目。此时将显示 Instance Management (实例管理) 上下文菜单。



4. 在该上下文菜单上，单击 Get Windows Password (获取 Windows 密码)。此时将显示一个称为 Retrieve Default Windows Administrator Password (检索默认 Windows 管理员密码) 的对话框。在此对话框上，将显示加密密码，以及您创建 AWS CloudFormation Windows 堆栈时使用的 Amazon EC2 密钥对。



5. 执行以下项中的*一项*操作（这些操作等效）：

   - 找到您下载的与显示的密钥对相对应的私有密钥文件，将其内容复制到剪贴板上，然后将其粘贴到该对话框的 Private Key (私有密钥) 框中。
   - 单击 Browse (浏览) 按钮，浏览到您系统中的私有密钥文件。选择该文件时，该文件的内容将显示在 Private Key (私有密钥) 框中。

6. 单击 Decrypt Password (解密密码)。随后将显示您实例的连接信息，包括：

   - 远程实例的 IP 地址。
   - 要在登录时使用的用户名称。
   - 已解密密码。

Note

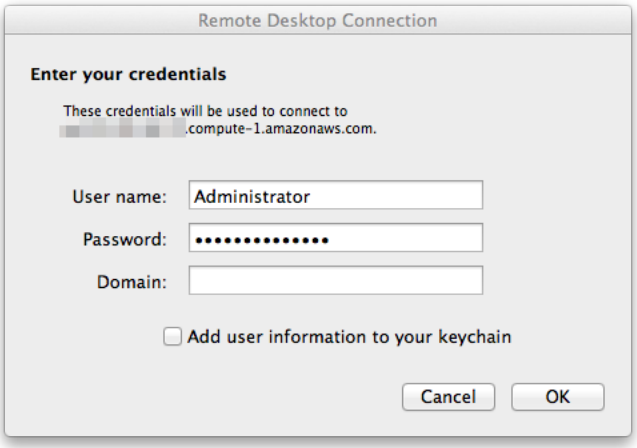
此密码为临时密码。一旦登录您的实例之后，您应将其更改为您自身选择之一。

通过 Remote Desktop 登录 Windows 实例时，可使用这些用户证书。

要登录 AWS CloudFormation Windows 堆栈，请执行以下操作：

1. 启动您的 Remote Desktop 客户端。

2. 当提示您输入 Server (服务器) 时，输入您从 EC2 中检索到的您的实例的服务器名称。



3. 输入从 EC2 中检索到的 User name (用户名称) (“Administrator”) 和 Password (密码)。
4. 如果提示您输入 Domain (域)，请将该字段留空。
5. 单击 OK (确定) 以完成连接。

登录服务器后，您可以按照自己的方式对其进行配置。您也可以使用该证书信息登录您堆栈创建的任何安全输出，例如 Sharepoint 站点。它是您的 Windows 实例，随心所欲地操作它吧！

模板参考

本部分详细说明了 AWS CloudFormation 模板中使用的受支持资源、类型名称、内部函数和虚拟参数。

Topics

AWS 资源类型参考

Abstract

列出有关 AWS CloudFormation 支持的所有 AWS 资源类型的详细参考信息。

本节包含 AWS CloudFormation 所支持的所有 AWS 资源的参考信息。

资源类型标识符总是采取以下形式：

```
AWS::aws-product-name::data-type-name
```

Topics

AWS::AutoScaling::AutoScalingGroup

Abstract

使用 AWS::AutoScaling::AutoScalingGroup 资源创建 Auto Scaling 组。

AWS::AutoScaling::AutoScalingGroup 类型可创建 Auto Scaling 组。

您可为您的 Auto Scaling 组添加 UpdatePolicy (p. 489) 属性，以控制对 Auto Scaling 组的启动配置 (p. 224)或子网组成员关系 (p. 222)进行更改后如何执行滚动升级。

语法

```
{
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones (p. 220)" : [ String, ... ],
      "Cooldown (p. 220)" : String,
      "DesiredCapacity (p. 220)" : String,
      "HealthCheckGracePeriod (p. 220)" : Integer,
```

```
      "HealthCheckType (p. 220)" : String,
      "InstanceId (p. 221)" : String,
      "LaunchConfigurationName (p. 221)" : String,
      "LoadBalancerNames (p. 221)" : [ String, ... ],
      "MaxSize (p. 221)" : String,
      "MetricsCollection (p. 222)" : [ MetricsCollection, ... ]
      "MinSize (p. 222)" : String,
      "NotificationConfiguration (p. 222)" : NotificationConfiguration,
      "Tags (p. 222)" : [ Auto Scaling Tag, ..., ],
      "TerminationPolicies (p. 222)" : [ String, ..., ],
      "VPCZoneIdentifier (p. 222)" : [ String, ... ]
   }
}
```

属性

AvailabilityZones
包含该组的可用区列表。

Required: Yes.

Type: A list of strings.

更新要求：无中断 (p. 63)

冷却时间
从一项扩展活动完成到其他扩展活动开始之间的秒数。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

DesiredCapacity
指定 Auto Scaling 组的所需容量。

如果没有在 AWS::AutoScaling::LaunchConfiguration (p. 224) 内为该 Auto Scaling 组设置 *SpotPrice*，则 Auto Scaling 将基于 *DesiredCapacity* 开始启用实例。CloudFormation 只有在所需的容量达到之后才会将该 Auto Scaling 组标记为“成功”（通过将其状态设置为 CREATE_COMPLETE）。

如果 *SpotPrice 已设置*，则不会将 *DesiredCapacity* 作为成功的标准。因为实例只有与竞价价格相匹配时才会运行。但一旦与竞价价格相匹配，Auto Scaling 就会将使用 *DesiredCapacity* 作为启用实例的目标容量。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

HealthCheckGracePeriod
新 EC2 实例开始运行之后，即 Auto Scaling 开始检查其运行状况的时长（以秒计）。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

HealthCheckType
您希望检查其运行状况的服务，Amazon EC2 或 Elastic Load Balancer。有效值为 "EC2" 或 "ELB"。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

实例 ID
要用于创建 Auto Scaling 组的 Amazon EC2 实例的 ID。如果要使用现有 Amazon EC2 实例（而不是启动配置）创建 Auto Scaling 组，请使用此属性。

使用 Amazon EC2 实例创建 Auto Scaling 组时，会先创建新的启动配置，然后将该配置与 Auto Scaling 组关联。新启动配置从该实例派生其所有属性（`BlockDeviceMapping` 和 `AssociatePublicIpAddress` 除外）。

Required: Conditional..如果不指定 `LaunchConfigurationName` 属性，则必须指定此属性。

Type: String.

更新要求：替换 (p. 63)

LaunchConfigurationName
指定相关联的 AWS::AutoScaling::LaunchConfiguration (p. 224) 的名称。

Note

如果此资源具有公有 IP 地址并且还处于同一模板中定义的 VPC 内，则您必须使用 `DependsOn` 属性声明对 VPC 网关连接的依赖关系。有关更多信息，请参阅 DependsOn 属性 (p. 486)。

必需：（有条件）如果不指定 `InstanceId` 属性，则必须指定此属性。

Type: String.

更新要求：无中断 (p. 63)

Important

更新 `LaunchConfigurationName` 时，现有 Amazon EC2 实例将继续采用它们最初启动时采用的配置运行。要更新现有实例，请为此 Auto Scaling 组指定更新策略属性。有关更多信息，请参阅 UpdatePolicy (p. 489)。

LoadBalancerNames
与该 Auto Scaling 组关联的负载均衡器列表。

Required: No.

Type: A list of strings.

更新要求：替换 (p. 63)

Important

更新 LoadBalancerNames 时，替换整个 Auto Scaling 组。

MaxSize
Auto Scaling 组的最大大小。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

MetricsCollection
启用 Auto Scaling 组的组指标监控。

Required: No.

类型：Auto Scaling MetricsCollection (p. 423) 列表

更新要求：无中断 (p. 63)

MinSize
Auto Scaling 组的最小大小。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

NotificationConfiguration
一种嵌入式属性，可将 Auto Scaling 组配置为在指定事件发生时发出通知。

Required: No.

类型： NotificationConfiguration (p. 423)

更新要求：无中断 (p. 63)

标签
要与此资源相连接的标签。

有关标签的更多信息，请参阅 *Auto Scaling 开发人员指南* 中的 Tagging Auto Scaling Groups and Amazon EC2 Instances。

Required: No.

类型：Auto Scaling 标签 (p. 424)列表

更新要求：无中断 (p. 63)

TerminationPolicies
用于选择要终止的实例的终止策略或终止策略列表。将按照排列顺序执行策略。

有关为 Auto Scaling 组配置终止策略的更多信息，请参阅 *Auto Scaling 开发人员指南* 中的Auto Scaling 组的实例终止策略。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

VPCZoneIdentifier
Amazon Virtual Private Cloud (Amazon VPC) 的子网标识符的列表。

为 `VPCZoneIdentifier` 指定的子网必须驻留在您通过 `AvailabilityZones` 参数指定的可用区中。

有关更多信息，请参阅 *Auto Scaling 开发人员指南* 中的 Using EC2 Dedicated Instances Within Your VPC。

Required: No.

Type: A list of strings.

更新要求：时而中断 (p. 63)

Note

更新 VPCZoneIdentifier 时，替换实例，但不替换 Auto Scaling 组。

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyASGroup" }
```

对于逻辑 ID 为“MyASGroup”的 Auto Scaling 组，`Ref` 将返回：

```
mystack-myasgroup-NT5EUXTNTXXD
```

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

具有弹性负载均衡器、启动配置和指标收集功能的 Auto Scaling 组

```
"WebServerGroup" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "2",
      "MaxSize" : "2",
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ],
      "MetricsCollection": [
         {
            "Granularity": "1Minute",
            "Metrics": [
               "GroupMinSize",
               "GroupMaxSize"
            ]
         }
      ]
   }
}
```

具有更新策略的 Auto Scaling 组

以下示例介绍如何通过包含 UpdatePolicy (p. 489) 属性来配置更新。该属性包含一个 AutoScalingRollingUpdate 嵌入式对象，其中有三个指定更新策略设置的属性。

```
"ASG1" : {
   "UpdatePolicy" : {
      "AutoScalingRollingUpdate" : {
         "MinInstancesInService" : "1",
         "MaxBatchSize" : "1",
         "PauseTime" : "PT12M5S"
```

```
      }
   },
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : { "Ref" : "AWS::Region" } },
      "LaunchConfigurationName" : { "Ref" : "ASLC" },
      "MaxSize" : "3",
      "MinSize" : "1"
   }
}
```

要查看更多 Auto Scaling 示例，请参阅 Auto Scaling 代码段 (p. 129)。

另请参阅

- UpdatePolicy (p. 489)
- *Auto Scaling API Reference* 中的 UpdateAutoScalingGroup
- AWS CloudFormation 堆栈更新 (p. 63)

AWS::AutoScaling::LaunchConfiguration

Abstract

使用 AWS::AutoScaling::LaunchConfiguration 资源创建 Auto Scaling 启动配置以配置 Amazon EC2 实例。

AWS::AutoScaling::LaunchConfiguration 类型可创建 Auto Scaling 启动配置，Auto Scaling 组可使用该配置在 Auto Scaling 组中配置 Amazon EC2 实例。

Important

在您更新 LaunchConfiguration 资源时，AWS CloudFormation 会删除该资源，并使用更新过的属性和新名称新建启动配置。这一更新操作不会对 Auto Scaling 组中正在运行的 Amazon EC2 实例部署任何更改。换句话说，更新只会替换 LaunchConfiguration。因此，当 Auto Scaling 组启动新实例时，它们将获得更新后的配置，但现有实例将继续采用启动时的原始配置运行。如果您手动对 Auto Scaling 组进行了任何类似的更改，此情况同样适用。

如果要在更新 LaunchConfiguration 资源时更新现有实例，您必须为 `AWS::AutoScaling::AutoScalingGroup` 资源指定更新策略属性。有关更多信息，请参阅 UpdatePolicy (p. 489)。

语法

```
{
   "Type" : "AWS::AutoScaling::LaunchConfiguration",
   "Properties" : {
      "AssociatePublicIpAddress (p. 225)" : Boolean,
      "BlockDeviceMappings (p. 225)" : [ BlockDeviceMapping, ... ],
      "EbsOptimized (p. 225)" : Boolean,
      "IamInstanceProfile (p. 225)" : String,
      "   ID (p. 225)" : String,
      "   ID (p. 226)" : String,
      "InstanceMonitoring (p. 226)" : Boolean,
```

```
      "     (p. 226)" : String,
      "KernelId (p. 226)" : String,
      "KeyName (p. 226)" : String,
      "RamDiskId (p. 226)" : String,
      "SecurityGroups (p. 227)" : [ SecurityGroup, ... ],
      "SpotPrice (p. 227)" : String,
      "UserData (p. 227)" : String
   }
}
```

属性

AssociatePublicIpAddress
对 VPC 中的 Amazon EC2 实例，指示 Auto Scaling 组中的实例是否接收公有 IP 地址。如果您指定 `true`，则 Auto Scaling 中的每个实例都会接收一个公有 IP 地址。

Note

如果此资源具有公有 IP 地址并且还处于同一模板中定义的 VPC 内，则您必须使用 DependsOn 属性声明对 VPC 网关连接的依赖关系。有关更多信息，请参阅 DependsOn 属性 (p. 486)。

Required: No.

Type: Boolean.

更新要求：替换 (p. 63)

BlockDeviceMappings
指定块存储设备如何对实例开放。您可以指定虚拟设备和 EBS 卷。

Required: No.

类型：BlockDeviceMappings (p. 421) 的列表。

更新要求：替换 (p. 63)

EbsOptimized
指定启动配置是否已针对 EBS I/O 进行优化。此优化可为 Amazon EBS 提供专用吞吐量，并提供优化的配置堆栈以实现最佳 EBS I/O 性能。

使用 EBS 优化实例会产生额外费用。有关费用和支持的实例类型的更多信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的 EBS 优化实例。

Required: No. 如果未指定该属性，则将使用“false”。

Type: Boolean.

更新要求：替换 (p. 63)

IamInstanceProfile
提供与实例的 IAM 角色关联的实例配置文件的名称或亚马逊资源名称 (ARN)。该实例配置文件包含 IAM 角色。

Required: No.

Type: String. (1 到 1600 个字符)

更新要求：替换 (p. 63)

映像 ID
提供注册期间分配的亚马逊系统映像 (AMI) 的唯一 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

实例 ID
要用于创建启动配置的 Amazon EC2 实例的 ID。如果希望启动配置使用现有 Amazon EC2 实例中的设置，请使用此属性。

使用某个实例创建启动配置时，所有属性都派生自该实例（`BlockDeviceMapping` 和 `AssociatePublicIpAddress` 除外）。您可以通过在启动配置中指定属性来覆盖实例中的任何属性。

Required: No.

Type: String.

更新要求：替换 (p. 63)

InstanceMonitoring
指明是否应该为该 Auto Scaling 组启用实例监控。该功能默认已启用。要将其关闭，请将 *`InstanceMonitoring`* 设置为“false”。

Required: No. 默认值为“true”。

Type: Boolean.

更新要求：替换 (p. 63)

实例类型
指定 EC2 实例的实例类型。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

KernelId
提供与 EC2 AMI 关联的内核的 ID。

Required: No.

Type: String.

更新要求：替换 (p. 63)

KeyName
提供 EC2 密钥对的名称。

Required: No.

Type: String.

更新要求：替换 (p. 63)

RamDiskId
供选择的 RAM 磁盘的 ID。一些内核启动时需要额外的驱动器。请查看内核要求，了解有关是否需要指定 RAM 磁盘的信息。要查找内核要求，请参阅 AWS 资源中心并搜索相应的内核 ID。

Required: No.

Type: String.

更新要求：替换 (p. 63)

SecurityGroups
一个包含 EC2 安全组的列表，这些安全组将分配给 Auto Scaling 组中的 Amazon EC2 实例。此列表可包含现有 EC2 安全组的名称或对在模板中创建的 AWS::EC2::SecurityGroup 资源的引用。如果实例是在 VPC 内启动的，请指定 Amazon VPC 安全组 ID。

Required: No.

Type: EC2 安全组列表。

更新要求：替换 (p. 63)

SpotPrice
该 Auto Scaling 组的竞价价格。如果已设置竞价价格，则 Auto Scaling 组将在当前竞价价格低于模板中指定的金额时启动。

如果您已为 Auto Scaling 组指定了竞价价格，则只有在达到竞价价格时，Auto Scaling 组才会启动，而不考虑该组 `DesiredCapacity` 中的设置。

有关为 Auto Scaling 组配置竞价价格的更多信息，请参阅 *Auto Scaling Developer Guide* 中的 Using Auto Scaling to Launch Spot Instances。

Required: No.

Type: String.

更新要求：替换 (p. 63)

Note

如果您通过创建新的启动配置更改了投标价格，则目前正在运行的实例将继续运行，直至它们的投标价格超出当前竞价价格。

UserData
适用于已启动的 EC2 实例的用户数据。

Required: No.

Type: String.

更新要求：替换 (p. 63)

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyAutoScalingGroup" }
```

对于逻辑 ID 为“MyAutoScalingGroup”的资源，`Ref` 将返回 AutoScaling 启动配置名称，如：`mystack-mylaunchconfig-1DDYF1E3B3I`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板示例

Example LaunchConfig（具有块储存设备）

此示例介绍的启动配置说明两个 Amazon Elastic Block Store 映射。

```
"LaunchConfig" : {
   "Type" : "AWS::AutoScaling::LaunchConfiguration",
   "Properties" : {
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : {
         "Fn::FindInMap" : [
            "AWSRegionArch2AMI",
            { "Ref" : "AWS::Region" },
            {
               "Fn::FindInMap" : [
                  "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch"
               ]
            }
         ]
      },
      "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }},
      "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
      "InstanceType" : { "Ref" : "InstanceType" }
      "BlockDeviceMappings" : [
         {
           "DeviceName" : "/dev/sda1",
           "Ebs" : { "VolumeSize" : "50", "VolumeType" : "io1", "Iops" : 200 }

         },
         {
           "DeviceName" : "/dev/sdm",
           "Ebs" : { "VolumeSize" : "100", "DeleteOnTermination" : "true"}
         }
      ]
   }
}
```

Example Autoscaling 组中具有竞价价格的 LaunchConfig

此例显示了 AutoScaling 组中具有竞价价格的启动配置。该启动配置仅在当前竞价价格低于模板规格中的金额 (0.05) 时有效。

```
"LaunchConfig" : {
   "Type" : "AWS::AutoScaling::LaunchConfiguration",
   "Properties" : {
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : {
         "Fn::FindInMap" : [
            "AWSRegionArch2AMI",
            { "Ref" : "AWS::Region" },
            {
               "Fn::FindInMap" : [
                  "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch"
               ]
            }
         ]
      },
      "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
      "SpotPrice" :  "0.05",
      "InstanceType" : { "Ref" : "InstanceType" }
   }
}
```

Example 具有 IAM 实例配置文件的 LaunchConfig

下面是一个使用 IamInstanceProfile (p. 225) 属性的启动配置。

仅显示了 AWS::AutoScaling::LaunchConfiguration 规范。如需了解完整模板，包括此处引用为“RootInstanceProfile”的 AWS::IAM::InstanceProfile (p. 346) 对象的定义和进一步参考，请参阅：auto_scaling_with_instance_profile.template。

```
"myLCOne": {
   "Type": "AWS::AutoScaling::LaunchConfiguration",
   "Properties": {
      "ImageId": {
         "Fn::FindInMap": [
            "AWSRegionArch2AMI",
            { "Ref": "AWS::Region" },
            {
               "Fn::FindInMap": [
                  "AWSInstanceType2Arch", { "Ref": "InstanceType" }, "Arch"
               ]
            }
         ]
      },
      "InstanceType": { "Ref": "InstanceType" },
      "IamInstanceProfile": { "Ref": "RootInstanceProfile" }
   }
}
```

Example 具有指定 PIOPS 的 EBS 优化卷

您可以使用所包含的 EBS 优化卷具有指定 PIOPS 的自动扩展实例创建一个 AWS CloudFormation 堆栈。这可以提高受 EBS 支持的实例的性能，*Amazon Elastic Compute Cloud User Guide* 中的 Increasing EBS Performance 对此有详细介绍。

Caution

使用 EBS 优化实例会产生额外费用。有关详细信息，请参阅 *Amazon Elastic Compute Cloud User Guide* 中的 EBS-Optimized Instances。

由于您无法重写 Auto Scaling 启动配置中的 PIOPS 设置，因此必须通过指定所需 PIOPS 的块存储设备映射对启动配置中的 AMI 进行配置。您可以通过创建具有以下特性的自有 EC2 AMI 来执行此操作：

- `m1.large` 或更大的实例类型。要进行 EBS 优化，则必需采用此设置。
- 卷类型为 "io1" 的 EBS 支持型 AMI，以及您希望由 Auto Scaling 启动的实例达到的 IOPS 次数。
- EBS 卷的大小必须能够配合您所需的 IOPS。IOPS 和存储量（吉字节，GiB）的比率为 10:1，因此 PIOPS 要达到 100，至少需要 10 GiB 根卷存储量。

请在您的 Auto Scaling 启动配置中使用此 AMI。例如，一个具有 PIOPS 且 AMI ID 为 `ami-7430ba44` 的 EBS 优化 AMI 在启动配置中的使用情况如下：

```
"LaunchConfig" : {
   "Type" : "AWS::AutoScaling::LaunchConfiguration",
   "Properties" : {
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : { "ami-7430ba44" },
      "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" } },
      "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
      "InstanceType" : { "m1.large" },
      "EbsOptimized" : "true"
   }
},
```

务必将 *`InstanceType`* 设置为至少 *m1.large*，并将 *`EbsOptimized`* 设置为 *true*。

当您创建此类启动配置时，您的已启动实例所包含的优化后 EBS 根卷将具有您在创建 AMI 时选定的 PIOPS。

要查看更多 LaunchConfiguration 代码段，请参阅 Auto Scaling 启动配置资源 (p. 129)。

另请参阅

- *Amazon Elastic Compute Cloud User Guide* 中的 Creating Your Own AMIs。
- *Amazon Elastic Compute Cloud User Guide* 中的 Block Device Mapping。

AWS::AutoScaling::ScalingPolicy

Abstract

使用 AWS::AutoScaling::ScalingPolicy 资源向 Auto Scaling 组添加扩展策略。

AWS::AutoScaling::ScalingPolicy 资源可向 Auto Scaling 组添加扩展策略。扩展策略将指明是增加还是减少 Auto Scaling 组，以及增加或减少的数量。有关扩展策略的更多信息，请参阅 Auto Scaling Developer Guide 中的 Scaling by Policy。

扩展策略可与 Amazon CloudWatch 警报配合使用。Amazon CloudWatch 警报可根据您指定的参数自动为您启动操作。扩展策略就是警报可以发起的操作类型之一。有关介绍如何创建由 Amazon CloudWatch 警报触发的 Auto Scaling 策略的代码段，请参阅 由 Amazon CloudWatch 警报触发的 Auto Scaling 策略 (p. 130)。

此类型支持更新。有关更新此资源的更多信息，请参阅 PutScalingPolicy。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type" : "AWS::AutoScaling::ScalingPolicy",
   "Properties" : {
      "AdjustmentType (p. 231)" : String,
      "AutoScalingGroupName (p. 231)" : String,
      "Cooldown (p. 231)" : String,
      "ScalingAdjustment (p. 231)" : String
   }
}
```

属性

AdjustmentType
: 指定 *ScalingAdjustment* 是绝对数还是当前容量所占的百分比。有效值包括 *ChangeInCapacity*、*ExactCapacity* 和 *PercentChangeInCapacity*。

必需：是

类型：字符串

更新要求：无中断 (p. 63)

AutoScaling 组名
: 您希望策略与其关联的 Auto Scaling 组的名称或亚马逊资源名称 (ARN)。

必需：是

类型：字符串

更新要求：无中断 (p. 63)

冷却时间
: 从一项扩展活动完成到其他与触发相关的扩展活动开始之间的时长（秒）。

必需：否

类型：字符串

更新要求：无中断 (p. 63)

ScalingAdjustment
: 要扩展的实例数量。AdjustmentType 可确定此数字的解释（例如，作为绝对数或现有 Auto Scaling 组大小所占的百分比）。正增量会为当前容量增加相应容量，负值则会从当前容量中删除相应容量。

必需：是

类型：字符串

更新要求：无中断 (p. 63)

返回值

如果将 AWS::AutoScaling::ScalingPolicy 类型指定为 `Ref` 函数的参数，AWS CloudFormation 将返回策略名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

AWS::AutoScaling::ScheduledAction

Abstract

为 Auto Scaling 组创建计划扩展操作。

为 Auto Scaling 组创建计划扩展操作，从而更改可用于应用程序的服务器数以响应可预测的负载变化。

语法

```
{
  "Type" : "AWS::AutoScaling::ScheduledAction",
  "Properties" : {
    "AutoScaling    (p. 232)" : String,
    "DesiredCapacity (p. 232)" : Integer,
    "EndTime (p. 232)" : Time stamp,
    "MaxSize (p. 233)" : Integer,
    "MinSize (p. 233)" : Integer,
    "   (p. 233)" : String,
    "StartTime (p. 233)" : Time stamp,
  }
}
```

属性

AutoScaling 组名
: Auto Scaling 组的名称或 ARN。

  Required: Yes.

  Type: String.

  更新要求：替换 (p. 63)

DesiredCapacity
: 应在 Auto Scaling 组中运行的 Amazon EC2 实例数。

  Required: No.

  Type: Integer

  更新要求：无中断 (p. 63)

EndTime
: 计划的结束时间（采用 UTC 表示）。例如，`2010-06-01T00:00:00Z`。

  Required: No.

  类型：时间戳

更新要求：无中断 (p. 63)

MaxSize
Auto Scaling 组中的最大 Amazon EC2 实例数。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

MinSize
Auto Scaling 组中的最小 Amazon EC2 实例数。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

循环
未来重复操作的开始时间（UTC 格式）。按照 Unix cron 语法格式指定开始时间。有关 cron 语法的更多信息，请参阅 http://en.wikipedia.org/wiki/Cron。

通过 `Recurrence` 属性指定 `StartTime` 和 `EndTime` 属性可组成重复操作的开始和停止边界。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

StartTime
计划的开始时间（采用 UTC 表示）。例如，`2010-06-01T00:00:00Z`。

Required: No.

类型：时间戳

更新要求：无中断 (p. 63)

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyScheduledAction" }
```

对于具有逻辑 ID `MyScheduledAction` 的计划 Auto Scaling 操作，`Ref` 返回计划操作名称。例如：

```
mystack-myscheduledaction-NT5EUXTNTXXD
```

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Auto Scaling 计划操作代码段

以下模板代码段包含两个扩展 Auto Scaling 组中的实例数的计划操作。`ScheduledActionUp` 操作每天上午 7 点开始，将 Auto Scaling 组设置为至少 5 个 Amazon EC2 实例，最多 10 个。`ScheduledActionDown` 操作每天晚上 7 点开始，将 Auto Scaling 组设置为最小值和最大值都为 1 个 Amazon EC2 实例。

```
"ScheduledActionUp": {
  "Type": "AWS::AutoScaling::ScheduledAction",
  "Properties": {
    "AutoScalingGroupName": {
      "Ref": "WebServerGroup"
    },
    "MaxSize": "10",
    "MinSize": "5",
    "Recurrence": "0 7 * * *"
  }
},
"ScheduledActionDown": {
  "Type": "AWS::AutoScaling::ScheduledAction",
  "Properties": {
    "AutoScalingGroupName": {
      "Ref": "WebServerGroup"
    },
    "MaxSize": "1",
    "MinSize": "1",
    "Recurrence": "0 19 * * *"
  }
}
```

AWS::CloudFormation::Authentication

Abstract

使用 AWS::CloudFormation::Authentication 资源为文件或源指定身份验证证书。

使用 AWS::CloudFormation::Authentication 资源可为通过 AWS::CloudFormation::Init (p. 241) 资源指定的文件或源指定身份验证证书。

若要包括使用 AWS::CloudFormation::Init 指定的文件或来源的身份验证信息，请使用 `uris` 属性（如果该资源是 URI）或存储桶属性（如果该资源是 Amazon S3 存储桶）。有关文件的更多信息，请参阅 文件 (p. 245)。有关源的更多信息，请参阅 来源 (p. 249)。

您也可以直接在 AWS::CloudFormation::Init 资源中指定文件的验证信息。资源的文件密钥包含名为 `authentication` 的属性。您可以使用 `authentication` 属性将在 an AWS::CloudFormation::Authentication 资源中定义的身份验证信息直接与文件相关联。

对于文件，AWS CloudFormation 会按以下顺序查找身份验证信息：

1. AWS::CloudFormation::Init `files` 密钥的 `authentication` 属性。
2. AWS::CloudFormation::Authentication 资源的 `uris` 或 `buckets` 属性。

对于源，AWS CloudFormation 会在 AWS::CloudFormation::Authentication 资源的 `uris` 或 `buckets` 属性中查找身份验证信息。

语法

与大多数 AWS CloudFormation 资源不同的是，AWS::CloudFormation::Authentication 类型不包含名为“Properties”的块，而是包含用户命名块的一个列表，每个用户命名块都包含其自己的身份验证属性。

并不是所有属性都与每个身份验证类型相关；有关详细信息，请参阅 type (p. 235) 属性。

```
{
   "Type" : "AWS::CloudFormation::Authentication" {
      "String" : {
         "accessKeyId (p. 235)" : String,
         "buckets (p. 235)" : [ String, ... ],
         "password (p. 235)" : String,
         "secretKey (p. 235)" : String,
         "type (p. 235)" : String,
         "uris (p. 235)" : [ String, ... ],
         "username (p. 236)" : String,
         "roleName (p. 236)" : String
      },
      ...
   }
}
```

属性

accessKeyId
: 指定 S3 验证的访问密钥 ID。

  Required: Conditional. 只有在将 type 属性设置为 "S3" 时才能指定。

  Type: String.

存储桶
: 要与 S3 身份验证凭据关联的 Amazon S3 存储桶的逗号分隔列表。

  Required: Conditional. 只有在将 type 属性设置为 "S3" 时才能指定。

  Type: A list of strings.

密码
: 指定基础验证密码。

  Required: Conditional. 只有在将 type 属性设置为 "basic" 时才能指定。

  Type: String.

secretKey
: 指定 S3 验证的访问密钥 ID。

  Required: Conditional. 只有在将 type 属性设置为 "S3" 时才能指定。

  Type: String.

类型
: 指定身份验证方案是使用用户名称和密码（“basic”）还是使用访问密钥 ID 和私有密钥（“S3”）。

  如果指定 "basic"，则必须还要指定 username、password 和 uris 属性。

  如果指定 "S3"，则必须还要指定 accessKeyId、secretKey 和 buckets 属性。

  Required: Yes.

  Type: String. 有效值为“basic”或“S3”

uris
: 要与基础级验证凭证关联的 URI 列表（以逗号分隔）。该授权适用于指定的 URI 和任何更多特定 URI。例如，如果指定 http://www.example.com，则该身份验证还将应用于 http://www.example.com/test。

Required: Conditional. 只有在将 type 属性设置为 `"basic"` 时才能指定。

Type: A list of strings.

username
指定基础身份验证的用户名称。

Required: Conditional. 只有在将 type 属性设置为 `"basic"` 时才能指定。

Type: String.

roleName
说明基于角色的身份验证的角色。

Required: Conditional. 只有在将 type 属性设置为 `"S3"` 时才能指定。

Type: String..

示例

Example EC2 Web 服务器身份验证

本模板片段显示了如何从 EC2 实例内的私有 S3 存储桶获取文件。用于身份验证的凭据是在 AWS::CloudFormation::Authentication 资源中定义的，并由 *files* 节中的 AWS::CloudFormation::Init 资源引用。

```
"WebServer": {
   "Type": "AWS::EC2::Instance",
   "DependsOn" : "BucketPolicy",
   "Metadata" : {
      "AWS::CloudFormation::Init" : {
         "config" : {
            "packages" : { "yum" : { "httpd" : [] } },
            "files" : {
               "/var/www/html/index.html" : {
                  "source" : {
                     "Fn::Join" : [
                      "", [ "http://s3.amazonaws.com/", { "Ref" : "BucketName"
 }, "/index.html" ]
                     ]
                  },
                  "mode"   : "000400",
                  "owner"  : "apache",
                  "group"  : "apache",
                  "authentication" : "S3AccessCreds"
               }
            },
            "services" : {
               "sysvinit" : {
                  "httpd" : { "enabled" : "true", "ensureRunning" : "true" }
               }
            }
         }
      },
      "AWS::CloudFormation::Authentication" : {
         "S3AccessCreds" : {
            "type" : "S3",
            "accessKeyId" : { "Ref" : "CfnKeys" },
            "secretKey" : { "Fn::GetAtt": [ "CfnKeys", "SecretAccessKey" ] }
         }
      }
   },
   "Properties": {
       ... EC2 Resource Properties ...
   }
}
```

Example 指定基本和 S3 身份验证

下面的示例代码段包含 *basic* 和 *S3* 身份验证类型。

```
"AWS::CloudFormation::Authentication" : {
   "testBasic" : {
      "type" : "basic",
      "username" : { "Ref" : "UserName" },
      "password" : { "Ref" : "Password" },
      "uris" : [ "http://www.example.com/test" ]
   },
   "testS3" : {
      "type" : "S3",
      "accessKeyId" : { "Ref" : "AccessKeyID" },
      "secretKey" : { "Ref" : "SecretAccessKeyID" },
      "buckets" : [ "myawsbucket" ]
   }
}
```

Example IAM 角色

下面的示例说明如何使用 IAM 角色。

```
"AWS::CloudFormation::Authentication": {
    "rolebased" : {
        "type": "s3",
        "buckets": [ "myBucket" ],
        "roleName": { "Ref": "myRole" }
    }
}
```

示例假定以下各项：

- `myRole` 是 AWS::IAM::Role (p. 351) 资源。
- 运行 cfn-init 的 Amazon EC2 实例通过实例配置文件与 `myRole` 相关联。
- 与正常 Amazon S3 身份验证一样，此示例使用存储桶属性来指定身份验证。也可以通过名称来指定身份验证。

完整模板示例

关于使用 AWS::CloudFormation::Authentication 资源的完整模板示例，请查看 AWS CloudFormation 示例模板 网页上的以下模板：

- S3Bucket_Auth_1.template
- S3Bucket_Auth_2.template
- S3Bucket_SourceAuth.template

AWS::CloudFormation::CustomResource

Abstract

使用 AWS::CloudFormation::CustomResource 资源指定自定义资源，以便 AWS CloudFormation 堆栈中可以包含非 AWS 资源。

自定义资源是特殊的 AWS CloudFormation 资源，提供了一种使 template developer 可以在 AWS CloudFormation 堆栈中包含非 AWS 资源的方式。custom resource provider 既可以是 template developer，也可以是独立的第三方资源提供者。

在模板中，自定义资源由 `AWS::CloudFormation::CustomResource` 或 `Custom::String` 表示。

语法

```
{
   "Type" : "AWS::CloudFormation::CustomResource",
   "Version" : "1.0",
   "Properties" : {
      "ServiceToken (p. 240)" : String,
      ... provider-defined properties ...
   }
}
```

或者

```
{
   "Type" : "Custom::String",
   "Version" : "1.0",
   "Properties" : {
      "ServiceToken (p. 240)" : String,
      ... provider-defined properties ...
   }
}
```

Note

AWS 为自定义资源仅定义一个属性：`ServiceToken`.其他所有属性都是服务供应商定义的。

自定义资源类型名称

对于自定义资源，您可以指定 `AWS::CloudFormation::CustomResource` 作为资源类型，也可以指定自己的资源类型名称。例如，您可以使用 `Custom::String`，而不使用 `AWS::CloudFormation::CustomResource`。自定义资源类型名称必须是字母数字字符，最大长度为 60 个字符。在更新期间，不能更改类型。

使用自己的资源类型名称有助于快速区分堆栈中自定义资源的类型。例如，如果有执行两种不同 ping 测试的两个自定义资源，则可以将自定义资源类型命名为 `Custom::PingTester`（而不使用 `AWS::CloudFormation::CustomResource`）以便方便地识别为 ping 测试器。

属性

ServiceToken
为访问该服务而由服务提供程序提供给 template developer 的服务令牌。

Required: Yes.

Type: String.

返回值

对于自定义资源，返回值由 custom resource provider 定义，通过对提供程序定义的属性调用 `Fn::GetAtt` (p. 502) 检索。

示例

在模板中创建自定义资源定义

下面的示例演示如何在模板中创建自定义资源定义。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MyFrontEndTest" : {
         "Type": "AWS::CloudFormation::CustomResource",
         "Version" : "1.0",
         "Properties" : {
            "ServiceToken": "arn:aws:sns:us-east-1:84969EXAMPLE:CRTest",
            "key1" : "string",
            "key2" : [ "list" ],
            "key3" : { "key4" : "map" }
         }
      }
   },
   "Outputs" : {
      "CustomResourceAttribute1" : {
         "Value" : { "Fn::GetAtt" : ["MyFrontEndTest", "responseKey1"] }
      },
      "CustomResourceAttribute2" : {
         "Value" : { "Fn::GetAtt" : ["MyFrontEndTest", "responseKey2"] }
      }
   }
}
```

除 `ServiceToken` 外的所有属性以及所有 `Fn::GetAtt` 资源属性都由 custom resource provider 定义。

为自定义资源创建用户定义资源类型

下面的示例演示如何为自定义资源创建类型名称。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MyFrontEndTest" : {
         "Type": "Custom::PingTester",
```

```
        "Version" : "1.0",
        "Properties" : {
           "ServiceToken": "arn:aws:sns:us-east-1:84969EXAMPLE:CRTest",
           "key1" : "string",
           "key2" : [ "list" ],
           "key3" : { "key4" : "map" }
        }
     }
  },
  "Outputs" : {
     "CustomResourceAttribute1" : {
        "Value" : { "Fn::GetAtt" : ["MyFrontEndTest", "responseKey1"] }
     },
     "CustomResourceAttribute2" : {
        "Value" : { "Fn::GetAtt" : ["MyFrontEndTest", "responseKey2"] }
     }
  }
}
```

在更新期间替换自定义资源

您可以更新需要替换基础物理资源的自定义资源。在 AWS CloudFormation 模板中更新自定义资源时，AWS CloudFormation 会向相应的自定义资源发送更新请求。如果该自定义资源需要替换，则新的自定义资源必须用新的物理 ID 发送响应。AWS CloudFormation 收到响应时，会比较新旧自定义资源的 `PhysicalResourceId`。如果物理资源 ID 不同，则 AWS CloudFormation 将该更新视为替换，并向旧资源发送删除请求。有关此过程的分步演练，请参阅堆栈更新 (p. 49)。

请注意以下几点：

- 您可以在 Events (事件) 选项卡上监视更新进度。有关更多信息，请参阅 查看堆栈数据和资源 (p. 85)。
- 有关更新期间的资源性能的更多信息，请参阅AWS CloudFormation 堆栈更新 (p. 63)。

另请参阅

- AWS CloudFormation 自定义资源演练 (p. 46)

AWS::CloudFormation::Init

Abstract

使用 AWS::CloudFormation::Init 资源类型在 Amazon EC2 实例上包括用于 cfn-init 帮助程序脚本的元数据。

Topics

使用 AWS::CloudFormation::Init 类型可在 Amazon EC2 实例上包括用于 cfn-init 帮助程序脚本的元数据。如果您的模板调用 cfn-init 脚本，该脚本会查询来源于 AWS::CloudFormation::Init 元数据密钥内的资源元数据。有关 cfn-init 的更多信息，请参阅 cfn-init (p. 513)。

元数据分成数个配置密钥，您可以将它们分成几个 configset。当您在模板中调用 cfn-init 时，可以指定 configset。如果您不指定配置集，则 cfn-init 会寻找名为 *config* 的单个配置密钥。

配置过程分为几个部分。以下模板代码段演示如何在模板内将 cfn-init 的元数据附加到 Amazon EC2 实例资源。

```
"Resources": {
  "MyInstance": {
    "Type": "AWS::EC2::Instance",
    "Metadata" : {
      "AWS::CloudFormation::Init" : {
        "config" : {
          "packages" : {
            :
          },
          "groups" : {
            :
          },
          "users" : {
            :
          },
          "sources" : {
            :
          },
          "files" : {
            :
          },
          "commands" : {
            :
          },
          "services" : {
            :
          }
        }
      }
    },
    "Properties": {
      :
    }
  }
}
```

Note

cfn-init 帮助程序脚本按以下顺序处理这些配置部分：软件包、组、用户、资源、文件、命令，随后是服务。如果您需要不同的顺序，请将各个部分分隔为不同的配置密钥，然后使用配置集指定处理配置密钥应采用的顺序。

`cfn-init` 支持 Linux 系统的所有元数据类型。在满足下面部分中描述的条件时，它还支持 Windows 的元数据类型。

有关使用 AWS::CloudFormation::Init 和 cfn-init 帮助程序脚本的示例，请参阅 使用 AWS CloudFormation 部署应用程序 (p. 198)。

有关说明如何使用 cfn-init 创建 Windows 堆栈的示例，请参阅启动 AWS CloudFormation Windows 堆栈 (p. 210)。

Configset

如果您创建多个配置密钥并使 cfn-init 以特定的顺序处理这些密钥，则可创建一个 configset，其中包含以特定顺序排列的多个配置密钥。例如，以下模板代码段创建名为 `ascending` 和 `descending` 的配置集，每个配置集包含两个配置密钥。

```
"AWS::CloudFormation::Init" : {
    "configSets" : {
        "ascending" : [ "config1" , "config2" ],
        "descending" : [ "config2" , "config1" ]
    },
    "config1" : {
        "commands" : {
            "test" : {
                "command" : "echo \"$CFNTEST\" > test.txt",
                "env" : { "CFNTEST" : "I come from config1." },
                "cwd" : "~"
            }
        }
    },
    "config2" : {
        "commands" : {
            "test" : {
                "command" : "echo \"$CFNTEST\" > test.txt",
                "env" : { "CFNTEST" : "I come from config2" },
                "cwd" : "~"
            }
        }
    }
}
```

以下示例 cfn-init 调用适用于前述示例 configset。为清楚起见，对示例调用进行了简略；有关完整语法，请参阅 cfn-init (p. 513)。

- 如果对 cfn-init 的调用指定 `ascending` 配置集：

  ```
  cfn-init -c ascending
  ```

  该脚本会处理 `config1`，然后处理 `config2`，并且 test.txt 文件将包含文本 `I come from config2`。
- 如果对 cfn-init 的调用指定 `descending` 配置集：

  ```
  cfn-init -c descending
  ```

  该脚本会处理 `config2`，然后处理 `config1`，并且 test.txt 文件将包含文本 `I come from config1`。

您可以创建多个 configset，并使用 cfn-init 脚本对它们进行一连串调用。每个 configset 都包含配置密钥列表或对其他 configset 的引用的列表。例如，以下模板片段可创建三个 configset。第一个配置集 (`test1`) 包含一个名为 `1` 的配置密钥。第二个配置集 (`test2`) 包含对 `test1` 配置集的引用以及一个名为 `2` 的配置密钥。第三个配置集 (default) 包含对配置集 `test2` 的引用。

```
"AWS::CloudFormation::Init" : {
    "configSets" : {
        "test1" : [ "1" ],
        "test2" : [ { "ConfigSet" : "test1" }, "2" ],
        "default" : [ { "ConfigSet" : "test2" } ]
    },
    "1" : {
        "commands" : {
            "test" : {
                "command" : "echo \"$MAGIC\" > test.txt",
                "env" : { "MAGIC" : "I come from the environment!" },
                "cwd" : "~"
            }
        }
    },
    "2" : {
        "commands" : {
            "test" : {
                "command" : "echo \"$MAGIC\" >> test.txt",
                "env" : { "MAGIC" : "I am test 2!" },
                "cwd" : "~"
            }
        }
    }
}
```

以下 cfn-init 调用适用于前述模板片段中申明的 configSet。为清楚起见，对示例调用进行了简略；有关完整语法，请参阅 cfn-init (p. 513)。

- 如果仅指定 `test1`：

  ```
  cfn-init -c test1
  ```

  cfn-init 仅处理配置密钥 `1`。
- 如果仅指定 `test2`：

  ```
  cfn-init -c test2
  ```

  cfn-init 处理配置密钥 `1`，然后处理配置密钥 `2`。
- 如果指定 `default` 配置集（或不指定任何配置集）：

  ```
  cfn-init -c default
  ```

  行为将与指定配置集 `test2` 时相同。

命令

您可以使用命令密钥在 EC2 实例上执行命令。其执行顺序即为命令名称的字母顺序。

键	说明
命令	必需。阵列或者字符串都可以指定要运行的命令。如果使用阵列，那么不需要转义空格字符，或者在命令参数周围使用引号。

键	说明
env	可选。设置该命令的环境变量。这个属性会覆盖而不是追加到现有的环境。
cwd	可选。工作目录
测试	可选。一个为了让 cfn-init 处理命令密钥中包含的命令而必须返回值 `true` 的命令。
ignoreErrors	可选。一个布尔值，用于确定在命令密钥中包含的命令失败时（返回非零值）cfn-init 是否继续运行。如果要使 cfn-init 在该命令失败的情况下继续运行，则设置为 `true`。如果要使 cfn-init 在该命令失败时停止运行，则设置为 `false`。默认值为 `false`。
waitAfterCompletion	可选。仅限于 Windows 系统。指定命令执行结束后，在该命令引起重启的情况下需要等待的时间长短（以秒为单位）。默认值是 60 秒，而值“forever”会使 cfn-init 退出，并且仅在重启完成后继续。

以下示例片段调用了 echo 命令。

```
"commands" : {
    "test" : {
        "command" : "echo \"$MAGIC\" > test.txt",
        "env" : { "MAGIC" : "I come from the environment!" },
        "cwd" : "~",
        "test" : "test ! -e ~/test.txt",
        "ignoreErrors" : "false"
    }
}
```

文件

您可以使用 `files` 密钥在 EC2 实例上创建文件。其内容既可以来自模板，也可以从 URL 抽取。这些文件会按词典顺序写入磁盘。下表列出了支持的密钥。

键	说明
内容	字符串或格式正确的 JSON 对象。如果您将 JSON 对象用作内容，则 JSON 将被写到磁盘上的文件中。您必须在 JSON 对象写入磁盘之前，评估所有固有函数，如 Fn::GetAtt 或 Ref。
来源	用于加载文件的 URL。不能使用内容密钥指定该选项。
编码	编码格式。仅限在内容为字符串时使用。如果使用的是源，则不会应用编码。 有效值：`plain` |`base64`
组	拥有这个文件的组的名称。在 Windows 系统中不受支持。
所有人	拥有这个文件的用户的名称。在 Windows 系统中不受支持。
模式	是一个六位数的八进制值，表示这个文件的模式。在 Windows 系统中不受支持。
身份验证	要使用的身份验证方法的名称。这会覆盖任何默认的身份验证。您可以使用此属性选择通过 AWS::CloudFormation::Authentication (p. 234) 资源定义的身份验证方法。

键	说明
context	指定需要作为 Mustache 模板处理的文件的上下文。若要使用此密钥，必须已安装 aws-cfn-bootstrap 1.3-11 或更高版本以及 pystache。

以下示例片段将文件 setup.mysql 创建为大型安装的一部分。

```
"files" : {
  "/tmp/setup.mysql" : {
    "content" : { "Fn::Join" : ["", [
      "CREATE DATABASE ", { "Ref" : "DBName" }, ";\n",
      "CREATE USER '", { "Ref" : "DBUsername" }, "'@'localhost' IDENTIFIED BY
'",
                       { "Ref" : "DBPassword" }, "';\n",
      "GRANT ALL ON ", { "Ref" : "DBName" }, ".* TO '", { "Ref" : "DBUsername"
 },
                       "'@'localhost';\n",
      "FLUSH PRIVILEGES;\n"
      ]]},
    "mode"  : "000644",
    "owner" : "root",
    "group" : "root"
  }
},
```

此处提供了完整模板：
https://s3.amazonaws.com/cloudformation-templates-us-east-1/Drupal_Single_Instance.template

Mustache 模板主要用于创建配置文件。例如，您可以将配置文件存储在一个 S3 存储桶中，并从模板插入 Refs 和 GetAtts，而不是使用 Fn::Join (p. 506)。下面的示例代码段将“Content for test9”输出到 `/tmp/test9.txt`。

```
"files" : {
    "/tmp/test9.txt" : {
        "content" : "Content for {{name}}",
        "context" : { "name" : "test9" }
    }
}
```

在使用 Mustache 模板时，请注意以下几点：

- 必须存在用于要处理的文件的上下文密钥。
- 上下文密钥必须是密钥值映射，但可以嵌套。
- 您可以使用内容密钥来处理含有内联内容的文件，并使用源密钥来处理远程文件。
- Mustache 支持取决于 pystache 版本。版本 0.5.2 支持 Mustache 1.1.2 规范。

组

您可以使用组密钥创建 Linux/UNIX 组，并分配组 ID。该组密钥在 Windows 系统中不受支持。

要创建组，请添加新的密钥值对，将新的组名映射到可选组 ID。组密钥可以包含一个或多个组名。下表列出了可用的密钥。

键	说明
gid	组 ID 编号。 如果指定了组 ID，且存在该名称的组，那么创建组将失败。如果其他组已使用指定的 ID，则操作系统可能会拒绝创建该组。 示例：`{ "gid" : "23" }`

示例片段

以下代码段指定名为 `groupOne` 的组而不分配组 ID，并指定名为 `groupTwo` 的组（指定组 ID 值为 `45`）。

```
"groups" : {
    "groupOne" : {},
    "groupTwo" : { "gid" : "45" }
}
```

软件包

您可以使用包密钥下载和安装各种预包装的应用程序和组件。在 Windows 系统上，软件包密钥仅支持 MSI 安装程序。

支持的软件包格式

Cfn-init 目前支持以下软件包格式：apt、msi、python、rpm、rubygems 和 yum。软件包处理顺序如下：rpm、yum/apt，然后是 rubygems 和 python。rubygems 和 python 之间没有处理顺序，且不保证以任何顺序安装每个包管理器内的包。

指定版本

在每个包管理器内，会使用包名和一系列版本对每个包进行指定。版本可以是字符串、一系列版本、空字符串或者空列表。空字符串或者空列表表示您需要最新的版本。对于 rpm 管理器，版本是以磁盘或 URL 上的文件的路径的方式指定的。

如果您指定软件包的版本，cfn-init 将尝试安装该版本，即使已在实例上安装该软件包的更新版本。一些包管理器支持多个版本，但其他的包管理器可能不支持。有关更多信息，请查看软件包管理器的文档。如果您不指定版本并且已安装该软件包的某个版本，则 cfn-init 脚本不会安装新版本 — 它会假设您要保留并使用现有版本。

示例代码段

下面的代码段为 rpm 指定版本 URL，从 yum 请求最新版本并从 rubygems 请求 chef 的版本 0.10.2：

```
"rpm" : {
  "epel" : "http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-
4.noarch.rpm"
},
"yum" : {
  "httpd" : [],
  "php" : [],
  "wordpress" : []
},
"rubygems" : {
```

```
  "chef" : [ "0.10.2" ]
}
```

下面的代码段为 MSI 包指定 URL：

```
"msi" : {
  "awscli" : "https://s3.amazonaws.com/aws-cli/AWSCLI64.msi"
}
```

Services

您可以使用此服务密钥来定义实例启动后应启用或禁用哪些服务。在 Linux 系统上，将通过使用 sysvinit 来支持此密钥。在 Windows 系统上，将通过使用 Windows 服务管理器来支持此密钥。

服务密钥还支持您指定源、软件包和文件之间的相关性，这样如果由于安装文件导致重启时，cfn-init 会处理服务重启部分。例如，如果您下载 Apache HTTP Server 软件包，则在堆栈创建过程中，软件包安装将自动启动 Apache HTTP Server。然而，如果 Apache HTTP Server 配置在堆栈创建过程的后期被更新，则要等到重启 Apache 服务器之后，更新才会生效。您可以使用服务密钥来确保 Apache HTTP 服务启动。

下表列出了支持的密钥。

键	说明
ensureRunning	设为 true 可确保 cfn-init 结束后服务仍保持运行。 设为 false 可确保 cfn-init 结束后服务不再运行。 忽略该密钥，将不更改服务状态。
已启用	设为 true 可确保启动时自动启动服务。 设为 false 可确保启动时不会自动启动服务。 忽略该密钥，将不更改这个属性。
文件	一系列文件。如果 cfn-init 直接通过文件块更改其中一个文件，则该服务将重启。
来源	一系列目录。如果 cfn-init 将存档扩展到这些目录之一，则此服务将重启。
软件包	软件包管理器映射至软件包名称列表。如果 cfn-init 安装或更新其中一个软件包，则此服务将重启。
命令	一系列命令名称。如果 cfn-init 运行指定的命令，则此服务将重启。

下面的 Linux 代码段配置服务如下：

- 如果 cfn-init 修改了 /etc/nginx/nginx.conf 或 /var/www/html，则 nginx 服务会重启。
- 如果使用 yum 安装或更新 php 或 spawn-fcgi，则 php-fastcgi 服务将重启。
- 发送邮件服务不会停止和停用。

```
"services" : {
  "sysvinit" : {
```

```
    "nginx" : {
      "enabled" : "true",
      "ensureRunning" : "true",
      "files" : ["/etc/nginx/nginx.conf"],
      "sources" : ["/var/www/html"]
    },
    "php-fastcgi" : {
      "enabled" : "true",
      "ensureRunning" : "true",
      "packages" : { "yum" : ["php", "spawn-fcgi"] }
    },
    "sendmail" : {
      "enabled" : "false",
      "ensureRunning" : "false"
    }
  }
}
```

下面的 Windows 代码段启动 `cfn-hup` 服务，将其设置为自动，并在 `cfn-init` 修改指定配置文件的情况下重新启动该服务：

```
"services" : {
  "windows" : {
    "cfn-hup" : {
      "enabled" : "true",
      "ensureRunning" : "true",
      "files" : ["c:\\cfn\\cfn-hup.conf", "c:\\cfn\\hooks.d\\cfn-auto-reload
er.conf"]
    }
  }
}
```

来源

您可以使用此资源密钥来下载存档文件并在 EC2 实例上的目标目录中取出文件。此密钥在 Linux 和 Windows 系统中完全受支持。

支持的格式

支持的格式包括 tar、tar+gzip、tar+bz2 以及 zip。

GitHub

如果您将 GitHub 用作源控制系统，则可以使用 cfn-init 和源打包机制来提取特定的应用程序版本。GitHub 允许您通过 URL 从特定版本创建 zip 或 tar，如下所示：

```
https://github.com/<your directory>/(zipball|tarball)/<version>
```

例如，以下代码段会将版本 *master* 作为一个 `.tar` 文件拉取。

```
"sources" : {
  "/etc/puppet" : https://github.com/user1/cfn-demo/tarball/master
  }
```

示例

以下示例会从 Amazon S3 存储桶下载 ZIP 文件，并将其解包到 /etc/myapp 中：

```
"sources" : {
  "/etc/myapp" : "https://s3.amazonaws.com/mybucket/myapp.tar.gz"
  }
```

您可以对来源使用身份验证证书，但不能将身份验证密钥放在源数据块中。相反，您可以将存储桶密钥包含在 S3AccessCreds 数据块中。有关示例，请参阅 示例模板.有关 Amazon S3 身份验证凭据的更多信息，请参阅 AWS::CloudFormation::Authentication (p. 234)。

用户

您可以使用用户密钥在 EC2 实例上创建 Linux/UNIX 用户。该用户密钥在 Windows 系统中不受支持。

下表列出了支持的密钥。

键	说明
uid	用户 ID。如果用户名还有另一个用户 ID，那么此创建过程会失败。如果该用户 ID 已分配给现有用户，则操作系统可能会拒绝创建请求。
组	一系列组名。该用户将被添加到列表的每个组中。
homeDir	用户的主目录。

创建的用户属于非交互式系统用户，带有 /sbin/nologin 外壳。这是特意设计的，无法修改。

```
"users" : {
    "myUser" : {
        "groups" : ["groupOne", "groupTwo"],
        "uid" : "50",
        "homeDir" : "/tmp"
    }
}
```

AWS::CloudFormation::Stack

Abstract

使用 AWS::CloudFormation::Stack 属性在父模板中将堆栈作为资源进行嵌套。

AWS::CloudFormation::Stack 类型在顶层模板中将堆栈作为资源进行嵌套。

您可以从该包含模板内的一个嵌套堆栈来添加输出值。您可以将 GetAtt (p. 502) 函数与该嵌套堆栈的逻辑名称以及嵌套堆栈中 Outputs.*NestedStackOutputName* 格式的输出值的名称结合使用。

在应用模板更改以更新顶层堆栈时，AWS CloudFormation 将更新顶层堆栈，并开始对其嵌套堆栈进行更新。AWS CloudFormation 会更新已修改的嵌套堆栈的资源，但不更新未修改的嵌套堆栈的资源。有关更多信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

AWS::CloudFormation::Stack 代码段：堆栈资源代码段 (p. 175).

Note

嵌套堆栈要求您确认 IAM 功能，即使嵌套堆栈不包含任何 IAM 资源。有关确认 IAM 功能的更多信息，请参阅使用 AWS Identity and Access Management 控制访问 (p. 59)中的“AWS CloudFormation 模板中的 IAM 资源”。

语法

```
{
   "Type" : "AWS::CloudFormation::Stack",
   "Properties" : {
      "NotificationARNs (p. 251)" : [ String, ... ],
      "   (p. 251)" : { CloudFormation          (p. 425) },
      "TemplateURL (p. 251)" : String,
      "TimeoutInMinutes (p. 251)" : String
   }
}
```

属性

NotificationARNs
向其发送有关堆栈事件的通知的现有 Amazon SNS 主题列表。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

参数
创建此嵌套堆栈时传递到 AWS CloudFormation 的参数集。

Note

如果您使用 `ref` 函数将参数值传递到嵌套堆栈，则逗号分隔列表参数必须属于 `String` 类型。换句话说，您不能将属于类型 `CommaDelimitedList` 的值传递给嵌套堆栈。

必需：有条件（如果嵌套堆栈需要输入参数，则是必需的）。

类型：CloudFormation 堆栈参数属性类型 (p. 425)

更新要求：更新是否会导致中断取决于所更新的资源。更新绝不会导致替换嵌套堆栈。

TemplateURL
模板的 URL 指定您要作为资源创建的堆栈。模板必须存储在 Amazon S3 存储桶上，因此，URL 必须采用以下格式：`https://s3.amazonaws.com/.../TemplateName.template`

Required: Yes.

类型：字符串

更新要求：更新是否会导致中断取决于所更新的资源。更新绝不会导致替换嵌套堆栈。

TimeoutInMinutes
AWS CloudFormation 等待嵌套堆栈达到 CREATE_COMPLETE 状态的时间长度，单位为分钟。默认值为无超时。当 AWS CloudFormation 检测到嵌套堆栈已达到 CREATE_COMPLETE 状态时，它

会在父堆栈中将该嵌套堆栈资源标记为 CREATE_COMPLETE，然后继续创建父堆栈。如果在嵌套堆栈达到 CREATE_COMPLETE 之前超时期结束，AWS CloudFormation 则将嵌套堆栈标记为已失败，并回滚嵌套堆栈和父堆栈。

Required: No.

类型：字符串

更新要求：不支持更新

返回值

Ref

对于 AWS::CloudFormation::Stack，`Ref` 返回堆栈 ID。例如：

```
arn:aws:cloudformation:us-east-1:123456789012:stack/mystack-mynestedstack-sgg
frhxhum7w/f449b250-b969-11e0-a185-5081d0136786
```

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

Outputs.*NestedStackOutputName*
返回值：指定嵌套堆栈的输出值，其中，*NestedStackOutputName* 是输出值的名称。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

AWS::CloudFormation::WaitCondition

Abstract

添加一个等待条件，该条件会暂停 AWS CloudFormation 堆栈创建，直至该条件收到所需数量的信号。

使用 AWS::CloudFormation::WaitCondition 和 AWS::CloudFormation::WaitConditionHandle 资源，您可以在模板中放置一个等待条件，以便 AWS CloudFormation 可将堆栈的创建暂停并在继续创建堆栈前等待一个信号。

语法

```
{
   "Type" : "AWS::CloudFormation::WaitCondition",
   "Properties" : {
      "Count (p. 253)" : String,
      "Handle (p. 253)" : String,
      "Timeout (p. 253)" : String
   }
}
```

属性

数量
: AWS CloudFormation 继续堆栈创建过程之前必须接收的成功信号的数目。当等待条件接收必需数目的成功信号后，AWS CloudFormation 会恢复堆栈的创建。如果等待条件在超时期结束前未接收到指定数目的成功信号，则 AWS CloudFormation 假定该等待条件已失败并将该堆栈回滚。

  Required: No.

  Type: String.

  更新要求：不支持更新

Handle
: 对用于发送此等待条件信号的等待条件句柄的引用。使用 `Ref` 内部函数来指定一个 AWS::CloudFormation::WaitConditionHandle (p. 255) 资源。

  只要您在堆栈更新过程中添加 WaitCondition 资源，便必须将等待条件与新的 WaitConditionHandle 资源关联。请勿重用已在模板中定义的旧等待条件句柄。如果重用等待条件句柄，则等待条件可能会计算来自上一个创建或更新堆栈命令的旧信号。

  Required: Yes.

  Type: String.

  更新要求：不支持更新

Timeout
: 等待 Count 属性所指定的信号数目的时间长度（以秒为单位）。`Timeout` 是最低限制属性，即超时不会早于指定时间发生，而可在该时间之后不久发生。为该属性指定的最长时间为 12 小时（43 200 秒）。

  Required: Yes.

  Type: String.

  更新要求：不支持更新

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Data
: *返回值*：一个 JSON 对象，它包含指定等待条件的等待信号中的 *`UniqueId`* 和 *`Data`* 值。有关等待条件信号的更多信息，请参阅 等待条件发送 JSON 格式 (p. 191)。

  具有 2 个信号的等待条件的返回值示例：

```
{ "Signal1" : "Step 1 complete." , "Signal2" : "Step 2 complete." }
```

有关使用 Fn:GetAtt 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

Example 具有一个 WaitConditionHandle 和 GetAtt 的简单 WaitCondition 示例

```
"Resources" : {
   "myWaitHandle" : {
      "Type" : "AWS::CloudFormation::WaitConditionHandle",
      "Properties" : { }
   },
   "myWaitCondition" : {
      "Type" : "AWS::CloudFormation::WaitCondition",
      "Properties" : {
         "Handle" : { "Ref" : "myWaitHandle" },
         "Timeout" : "300"
      }
   }
},
"Outputs" : {
   "ApplicationData" : {
      "Value" : { "Fn::GetAtt" : [ "myWaitCondition", "Data" ]},
      "Description" : "The data passed back as part of signalling the WaitCon
dition"
   }
}
```

Example 等待 Windows 服务器实例启动的 WaitCondition

```
"WindowsServerWaitHandle" : {
   "Type" : "AWS::CloudFormation::WaitConditionHandle"
},
"WindowsServerWaitCondition" : {
   "Type" : "AWS::CloudFormation::WaitCondition",
   "DependsOn" : "WindowsServer",
   "Properties" : {
      "Handle" : { "Ref" : "WindowsServerWaitHandle" },
      "Timeout" : "1800"
   }
}
```

Example 等待 Web 服务器组中的实例达到所需数量的 WaitCondition

```
"WebServerGroup" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "1",
      "MaxSize" : "5",
      "DesiredCapacity" : { "Ref" : "WebServerCapacity" },
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
   }
},

"WaitHandle" : {
   "Type" : "AWS::CloudFormation::WaitConditionHandle"
},

"WaitCondition" : {
   "Type" : "AWS::CloudFormation::WaitCondition",
   "DependsOn" : "WebServerGroup",
   "Properties" : {
      "Handle"  : { "Ref" : "WaitHandle" },
      "Timeout" : "300",
      "Count"   : { "Ref" : "WebServerCapacity" }
   }
}
```

另请参阅

- 在模板中创建等待条件 (p. 188)
- 等候条件模板代码段 (p. 176)
- DependsOn 属性 (p. 486)

AWS::CloudFormation::WaitConditionHandle

Abstract

创建一个预签名 URL，用于指定接收为 AWS CloudFormation WaitCondition 资源发送的信号。

类型 AWS::CloudFormation::WaitConditionHandle 没有属性。在使用 `Ref` 函数引用 WaitConditionHandle 资源时，AWS CloudFormation 会返回预签名 URL。您可将此 URL 传递给在 Amazon EC2 实例上运行的应用程序或脚本，以向该 URL 发送信号。一个关联的 AWS::CloudFormation::WaitCondition (p. 252) 资源会检查该 URL 是否有所需数目的成功信号或是否有失败信号。

Important

只要您在堆栈更新过程中添加 WaitCondition 资源，便必须将等待条件与新的 WaitConditionHandle 资源关联。请勿重用已在模板中定义的旧等待条件句柄。如果重用等待条件句柄，则等待条件可能会计算来自上一个创建或更新堆栈命令的旧信号。

语法

```
{
   "Type" : "AWS::CloudFormation::WaitConditionHandle",
   "Properties" : {
   }
}
```

Note

此资源不支持更新。

相关资源

- 有关如何使用等待条件的信息，请参阅 在模板中创建等待条件 (p. 188)。
- 有关 AWS::CloudFormation::WaitCondition 代码段，请参阅等候条件模板代码段 (p. 176)。

AWS::CloudFront::Distribution

Abstract

使用 AWS::CloudFront::Distribution 资源创建 Amazon CloudFront Web 分配。

创建 Amazon CloudFront Web 分配。有关 CloudFront 分配的一般信息，请参阅 *Amazon CloudFront 开发人员指南* 中的 Introduction to Amazon CloudFront。有关创建 CloudFront Web 分配的特定信息，请参阅 *Amazon CloudFront API 参考* 中的 POST 分配。

语法

```
{
   "Type" : "AWS::CloudFront::Distribution",
   "Properties" : {
      "DistributionConfig (p. 256)" : DistributionConfig
   }
}
```

属性

DistributionConfig
分配的配置信息。

Required: Yes.

类型：DistributionConfig (p. 430) 类型

更新要求：无中断 (p. 63)

返回值

Ref

返回值：CloudFront 分配 ID。例如：`E27LVI50CSW06W`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

DomainName
返回值：资源的域名。例如：`d2fadu0nynjpfn.cloudfront.net`.

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

AWS::CloudWatch::Alarm

Abstract

使用 AWS::CloudFront::Distribution 资源创建 Amazon CloudFront Web 分配。

AWS::CloudWatch::Alarm 类型可创建 Amazon CloudWatch 警报。

此类型支持更新。有关更新此资源的更多信息，请参阅 PutMetricAlarm。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type" : "AWS::CloudWatch::Alarm",
   "Properties" : {
      "ActionsEnabled (p. 258)" : Boolean,
      "AlarmActions (p. 258)" : [ String, ... ],
      "AlarmDescription (p. 258)" : String,
      "AlarmName (p. 258)" : String,
      "ComparisonOperator (p. 258)" : String,
      "   (p. 259)" : [ Metric dimension, ... ],
      "EvaluationPeriods (p. 259)" : String,
      "InsufficientDataActions (p. 259)" : [ String, ... ],
      "MetricName (p. 259)" : String,
      "     (p. 259)" : String,
      "OKActions (p. 259)" : [ String, ... ],
      "   (p. 259)" : String,
      "     (p. 260)" : String,
      "   (p. 260)" : String,
      "Unit (p. 260)" : String
   }
```

```
}
```

属性

ActionsEnabled
指明是否应该在对警报状态进行更改期间执行操作。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

AlarmActions
当此警报从任何其他状态转换为 ALARM 状态时，要执行的操作列表。每个操作都指定为一个亚马逊资源编号 (ARN)。目前，受支持的唯一操作正发布至 Amazon SNS 主题或 Amazon Auto Scaling 策略。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

AlarmDescription
对警报的描述。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

AlarmName
警报的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作警报名称。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

Required: No.

Type: String.

更新要求：替换 (p. 63)

ComparisonOperator
将指定统计数据与阈值进行比较时使用的算术运算符。指定的统计值将作为第一个操作数。

可以指定以下值：`GreaterThanOrEqualToThreshold` | `GreaterThanThreshold` | `LessThanThreshold` | `LessThanOrEqualToThreshold`

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

维度
警报相关指标的维度。

Required: No.

类型：指标维度 (p. 434)列表

更新要求：无中断 (p. 63)

EvaluationPeriods
其间的数据将与指定阈值进行比较的期间数。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

InsufficientDataActions
当此警报从任何其他状态转换为 INSUFFICIENT_DATA 状态时，要执行的操作列表。每个操作都指定为一个亚马逊资源编号 (ARN)。目前，受支持的唯一操作正发布至 Amazon SNS 主题或 Amazon Auto Scaling 策略。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

MetricName
警报相关指标的名称。有关您可指定的指标的更多信息，请参阅 *Amazon CloudWatch 开发者指南* 中的 Amazon CloudWatch Namespaces, Dimensions, and Metrics Reference。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

命名空间
警报相关指标的命名空间。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

OKActions
当此警报从任何其他状态转换为 OK 状态时，要执行的操作列表。每个操作都指定为一个亚马逊资源编号 (ARN)。目前，受支持的唯一操作正发布至 Amazon SNS 主题或 Amazon Auto Scaling 策略。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

周期
在其间应用指定统计数据的期间（秒数）。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

统计数据
要应用至警报相关指标的统计数据。

可以指定以下值：`SampleCount` | `Average` | `Sum` | `Minimum` | `Maximum`

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

阈值
指定统计数据的比较值。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

Unit
警报相关指标的单位。

可以指定以下值：秒 | 微秒 | 毫秒 | 字节 | 千字节 | 兆字节 | 千兆字节 | 太兆字节 | 位 | 千位 | 兆位 | 千兆位 | 太兆位 | 百分比 | 计数 | 字节/秒 | 千字节/秒 | 兆字节/秒 | 千兆字节/秒 | 太兆字节/秒 | 位/秒 | 千位/秒 | 兆位/秒 | 千兆位/秒 | 太兆位/秒 | 计数/秒 | 无

Required: No.

Type: String.

更新要求：无中断 (p. 63)

返回值

Ref

如果将 AWS::CloudWatch::Alarm 类型指定为 `Ref` 函数的参数，AWS CloudFormation 将返回 *`AlarmName`* 的值。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

AWS::DynamoDB::Table

Abstract

使用 AWS::DynamoDB::Table 资源创建 Amazon DynamoDB 表。

创建 Amazon DynamoDB 表。

Note

AWS CloudFormation 通常并行创建 Amazon DynamoDB 表。但是，如果模板包含具有索引的 Amazon DynamoDB 表，则必须声明依赖关系，才能按顺序创建表。有关示例代码段，请参阅具有 DependsOn 属性的 Amazon DynamoDB 表 (p. 264)。

语法

```
{
   "Type" : "AWS::DynamoDB::Table",
   "Properties" : {
      "AttributeDefinitions (p. 261)" : [ AttributeDefinitions, ... ],
      "GlobalSecondaryIndexes (p. 261)" : [ GlobalSecondaryIndexes, ... ],
      "KeySchema (p. 261)" : [ KeySchema, ... ],
      "LocalSecondaryIndexes (p. 261)" : [ LocalSecondaryIndexes, ... ],
      "ProvisionedThroughput (p. 261)" : { ProvisionedThroughput },
      "    (p. 262)" : String
   }
}
```

属性

AttributeDefinitions
说明表和索引的键架构的 AttributeName 和 AttributeValue 对象列表。

必需：是

类型：DynamoDB 属性定义 (p. 435)

更新要求：替换 (p. 63)

GlobalSecondaryIndexes
要对表创建的全局二级索引。可以创建最多 5 个全局二级索引。

必需：否

类型：DynamoDB 全局二级索引 (p. 436)

更新要求：替换 (p. 63)

KeySchema
指定组成表主键的属性。KeySchema 属性中的属性还必须在 AttributeDefinitions 属性中定义。

必需：是

类型：DynamoDB 键架构 (p. 437)

更新要求：替换 (p. 63)

LocalSecondaryIndexes
要对表创建的本地二级索引。可以创建最多5个本地二级索引。每个索引的范围都限定到给定哈希键值。每个哈希键的大小最大可以为 10 GB。

必需：否

类型：DynamoDB 本地二级索引 (p. 438)

更新要求：替换 (p. 63)

ProvisionedThroughput
指定表的吞吐量，由 ReadCapacityUnits 值和 WriteCapacityUnits 值组成。有关已配置吞吐量结构的更多信息，请参阅 DynamoDB 预置吞吐量 (p. 439)。

必需：是

类型：DynamoDB 预置吞吐量 (p. 439)

更新要求：无中断 (p. 63)

表名称

表的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作表名称。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

必需：否

类型：名称类型 (p. 465)

更新要求：替换 (p. 63)

Note

有关 Amazon DynamoDB 中的限制的详细信息，请参阅 *Amazon DynamoDB 开发者指南* 中的 Amazon DynamoDB 中的限制。

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyResource" }
```

对于逻辑 ID 为 `myDynamoDBTable` 的资源，`Ref` 返回 Amazon DynamoDB 表名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

具有本地和二级索引的 Amazon DynamoDB 表

以下示例创建一个 Amazon DynamoDB 表，以 `Album`、`Artist` 和 `Sales` 作为属性。主键包含 `Album` 属性作为哈希键，包含 `Artist` 作为范围键。该表还包含一个全局和一个二级索引。查询给定艺术家的销量时，全局二级索引使用 `Sales` 属性作为哈希键，使用 `Artist` 属性作为范围键。查询专辑销量时，本地二级索引使用与表相同的哈希键，但是使用 `Sales` 属性作为范围键。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Resources" : {
    "myDynamoDBTable" : {
      "Type" : "AWS::DynamoDB::Table",
      "Properties" : {
        "AttributeDefinitions" : [
          {
            "AttributeName" : "Album",
            "AttributeType" : "S"
          },
          {
            "AttributeName" : "Artist",
            "AttributeType" : "S"
          },
          {
```

```
        "AttributeName" : "Sales",
        "AttributeType" : "N"
      }
    ],
    "KeySchema" : [
      {
        "AttributeName" : "Album",
        "KeyType" : "HASH"
      },
      {
        "AttributeName" : "Artist",
        "KeyType" : "RANGE"
      }
    ],
    "ProvisionedThroughput" : {
      "ReadCapacityUnits" : "5",
      "WriteCapacityUnits" : "5"
    },
    "TableName" : "myTableName",
    "GlobalSecondaryIndexes" : [{
      "IndexName" : "myGSI",
      "KeySchema" : [
        {
          "AttributeName" : "Sales",
          "KeyType" : "HASH"
        },
        {
          "AttributeName" : "Artist",
          "KeyType" : "RANGE"
        }
      ],
      "Projection" : {
        "NonKeyAttributes" : ["Album"],
        "ProjectionType" : "INCLUDE"
      },
      "ProvisionedThroughput" : {
        "ReadCapacityUnits" : "5",
        "WriteCapacityUnits" : "5"
      }
    }],
    "LocalSecondaryIndexes" :[{
      "IndexName" : "myLSI",
      "KeySchema" : [
        {
          "AttributeName" : "Album",
          "KeyType" : "HASH"
        },
        {
          "AttributeName" : "Sales",
          "KeyType" : "RANGE"
        }
      ],
      "Projection" : {
        "NonKeyAttributes" : ["Artist"],
        "ProjectionType" : "INCLUDE"
      }
    }]
  }
```

```
    }
  }
}
```

具有 DependsOn 属性的 Amazon DynamoDB 表

如果单个模板包含多个具有索引的 Amazon DynamoDB 表，则必须包含依赖关系，才能按顺序创建表。以下示例假设 `myFirstDDBTable` 表在与 `mySecondDDBTable` 表相同的模板中声明，两个表都包含一个二级索引。`mySecondDDBTable` 表包含对 `myFirstDDBTable` 表的依赖关系，从而 AWS CloudFormation 一次创建一个表。

```
"mySecondDDBTable" : {
  "Type" : "AWS::DynamoDB::Table",
  "DependsOn" : "myFirstDDBTable" ,
  "Properties" : {
    "AttributeDefinitions" : [
      {
        "AttributeName" : "ArtistId",
        "AttributeType" : "S"
      },
      {
        "AttributeName" : "Concert",
        "AttributeType" : "S"
      },
      {
        "AttributeName" : "TicketSales",
        "AttributeType" : "S"
      }
    ],
    "KeySchema" : [
      {
        "AttributeName" : "ArtistId",
        "KeyType" : "HASH"
      },
      {
        "AttributeName" : "Concert",
        "KeyType" : "RANGE"
      }
    ],
    "ProvisionedThroughput" : {
      "ReadCapacityUnits" : {"Ref" : "ReadCapacityUnits"},
      "WriteCapacityUnits" : {"Ref" : "WriteCapacityUnits"}
    },
    "GlobalSecondaryIndexes" : [{
      "IndexName" : "myGSI",
      "KeySchema" : [
        {
          "AttributeName" : "TicketSales",
          "KeyType" : "HASH"
        }
      ],
      "Projection" : {
        "ProjectionType" : "KEYS_ONLY"
      },
      "ProvisionedThroughput" : {
        "ReadCapacityUnits" : {"Ref" : "ReadCapacityUnits"},
```

```
        "WriteCapacityUnits" : {"Ref" : "WriteCapacityUnits"}
      }
    }]
  }
}
```

AWS::EC2::CustomerGateway

Abstract

使用 AWS::EC2::CustomerGateway 资源向 AWS 提供有关您的 VPN 客户网关设备的信息。

向 AWS 提供有关您的 VPN 客户网关设备的信息。

语法

```
{
   "Type" : "AWS::EC2::CustomerGateway",
   "Properties" : {
      "BgpAsn (p. 265)" : Number,
      "IpAddress (p. 265)" : String,
      "   (p. 265)" :  [ Resource Tag, ... ],
      "Type (p. 265)" : String
   }
}
```

属性

BgpAsn
客户网关的边界网关协议 (BGP) 自治系统编号 (ASN)。

Required: Yes.

Type: Number. BgpAsn 始终是整数值。

更新要求：替换 (p. 63)

IpAddress
可在互联网上路由的 IP 地址，用于客户网关的外部接口。该地址必须是静态的。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

标签
需要附加到资源的标签。

Required: No.

类型：AWS CloudFormation 资源标签 (p. 471).

更新要求：无中断 (p. 63).

类型
客户网关支持的 VPN 连接类型。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

示例：`ipsec.1`

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyResource" }
```

对于具有逻辑 ID“MyResource”的资源，`Ref` 将返回 AWS 资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myCustomerGateway" : {
         "Type" : "AWS::EC2::CustomerGateway",
         "Properties" : {
            "Type" : "ipsec.1",
            "BgpAsn" : "64000",
            "IpAddress" : "1.1.1.1"
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateCustomerGateway。

AWS::EC2::DHCPOptions

Abstract

使用 AWS::EC2::DHCPOptions 资源为您的 VPC 创建一组 DHCP 选项。

为您的 VPC 创建 DHCP 选项集。

有关更多信息，请参阅 *Amazon Elastic Compute Cloud API 参考* 中的 CreateDhcpOptions。

语法

```
{
   "Type" : "AWS::EC2::DHCPOptions",
   "Properties" : {
      "DomainName (p. 267)" : String,
      "DomainNameServers (p. 267)" : [ String, ... ],
      "NetbiosNameServers (p. 267)" : [ String, ... ],
      "NetbiosNodeType (p. 267)" : Number,
      "NtpServers (p. 268)" : [ String, ... ],
      "Tags (p. 268)" : [ EC2 Tag, ... ]
   }
}
```

属性

DomainName
您选择的域名。

必需：有条件；请参阅备注 (p. 268)。

Type: String.

更新要求：替换 (p. 63)

Example: `"example.com"`

DomainNameServers
域名服务器的 IP (IPv4) 地址。您最多可以指定四个地址。

必需：有条件；请参阅备注 (p. 268)。

Type: A list of strings.

更新要求：替换 (p. 63)

Example: `"DomainNameServers" : [ "10.0.0.1", "10.0.0.2" ]`

NetbiosNameServers
NetBIOS 名称服务器的 IP (IPv4) 地址。您最多可以指定四个地址。

必需：有条件；请参阅备注 (p. 268)。

Type: A list of strings.

更新要求：替换 (p. 63)

Example: `"NetbiosNameServers" : [ "10.0.0.1", "10.0.0.2" ]`

NetbiosNodeType
表示 NetBIOS 节点类型的整数值：

- 1:广播（“B”）
- 2:点对点（“P”）
- 4:混合模式（“M”）
- 8:混合（“H”）

有关这些值以及 NetBIOS 节点类型的更多信息，请参阅 RFC 2132、RFC 1001 和 RFC 1002。建议此时仅使用值 2（目前不支持广播和多播）。

Required: 如果指定了 `NetBiosNameServers`，则是必需的；否则为可选。

Type: A list of numbers.

更新要求：替换 (p. 63)

Example: `"NetbiosNodeType" : 2`

NtpServers
网络时间协议 (NTP) 服务器的 IP (IPv4) 地址。您最多可以指定四个地址。

必需：有条件；请参阅备注 (p. 268)。

Type: A list of strings.

更新要求：替换 (p. 63)

Example: `"NtpServers" : [ "10.0.0.1" ]`

标签
要与此资源相连接的标签。

Required: 编号

Type: EC2 标签 (p. 453)的列表

更新要求：无中断 (p. 63)

有关标签的更多信息，请参阅 *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags。

条件属性

必须指定下列属性中的*至少一个*：

- DomainNameServers (p. 267)
- NetbiosNameServers (p. 267)
- NtpServers (p. 268)

满足此条件后，其余的属性为选填项。

如果指定 NetbiosNameServers，则需要有 NetbiosNodeType。

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myDhcpOptions" : {
```

```
        "Type" : "AWS::EC2::DHCPOptions",
        "Properties" : {
           "DomainName" : "example.com",
           "DomainNameServers" : [ "AmazonProvidedDNS" ],
           "NtpServers" : [ "10.2.5.1" ],
           "NetbiosNameServers" : [ "10.2.5.1" ],
           "NetbiosNodeType" : 2,
           "Tags" : [ { "Key" : "foo", "Value" : "bar" } ]
        }
     }
  }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateDhcpOptions
- *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags。
- RFC 2132 - *DHCP 选项和 BOOTP 供应商扩展*，网络工作组，1997
- RFC 1001 - *TCP/UDP 传输 NetBIOS 服务的协议标准：概念和方法*，网络工作组，1987
- RFC 1002 - *TCP/UDP 传输 NetBIOS 服务的协议标准：详细规范*，网络工作组，1987

AWS::EC2::EIP

Abstract

使用 AWS::EC2::EIP 资源分配弹性 IP 地址，并且根据需要将其与 Amazon EC2 实例关联。

AWS::EC2::EIP 资源将分配一个弹性 IP (EIP) 地址，并且可以根据需要将其与 Amazon EC2 实例关联。

语法

```
{
   "Type" : "AWS::EC2::EIP",
   "Properties" : {
      "InstanceId (p. 269)" : String,
      "Domain (p. 269)" : String
   }
}
```

属性

实例 ID
: 您希望将其与此弹性 IP 地址关联的 Amazon EC2 实例的实例 ID。

  必需：否

  类型：字符串

  更新要求：无中断 (p. 63)

域
: 设置为 vpc 以将地址分配给 Virtual Private Cloud (VPC)。不支持任何其他值。

Note

如果您定义弹性 IP 地址并且将其与相同模板中定义的 VPC 关联，则必须对此资源使用 `DependsOn` 属性，声明对 VPC 网关连接的依赖关系。有关更多信息，请参阅 DependsOn 属性 (p. 486)。

有关更多信息，请参阅 *Amazon Elastic Compute Cloud API 参考* 中的 AllocateAddress。有关 VPC 中的弹性 IP 地址的更多信息，请参阅 *Amazon VPC User Guide* 中的 IP Addressing in Your VPC。

必需：条件。向 VPC 分配地址时必填。

类型：字符串

更新要求：替换 (p. 63)

返回值

Ref

如果将 AWS::EC2::EIP 对象的逻辑 ID 指定为 `Ref` 函数的参数，AWS CloudFormation 将返回实例的 *`PublicIp`* 值。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

AllocationId
: AWS 分配的 ID，用于表示与 Amazon VPC 配合使用的地址的分配。该值仅针对 VPC 弹性 IP 地址返回。示例返回值：`eipalloc-5723d13e`

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

要查看 AWS::EC2::EIP 代码段，请参阅 使用 AWS::EC2::EIP 代码段分配 Amazon EC2 弹性 IP (p. 136)。

AWS::EC2::EIPAssociation

Abstract

使用 AWS::EC2::EIPAssociation 资源将弹性 IP 地址与 Amazon EC2 实例关联。

AWS::EC2::EIPAssociation 资源类型可将弹性 IP 地址与 Amazon EC2 实例关联。该弹性 IP 地址可以是现有弹性 IP 地址，也可以是通过 AWS::EC2::EIP 资源 (p. 269)分配的弹性 IP 地址。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type": "AWS::EC2::EIPAssociation",
```

```
   "Properties": {
      "AllocationId (p. 271)": String,
      "EIP (p. 271)": String,
      "InstanceId (p. 271)": String,
      "NetworkInterfaceId (p. 271)": String,
      "PrivateIpAddress (p. 271)": String
   }
}
```

属性

AllocationId
: 您希望与 VPC 中的 Amazon EC2 实例关联的 VPC Elastic IP 地址的分配 ID。

  必需：条件。对于 VPC 为必需属性。

  类型：字符串

  更新要求：替换 (p. 63) - 在还要更改 `InstanceId` 或 `NetworkInterfaceId` 属性时。如果不更改，则更新要求无中断 (p. 63)。

EIP
: 您希望将其与 `InstanceId` 属性指定的 Amazon EC2 实例关联的弹性 IP 地址。您可以指定一个现有的弹性 IP 地址，也可以引用使用 AWS::EC2::EIP 资源 (p. 269)分配的弹性 IP 地址。

  必需：条件。对于弹性 IP 地址为必需属性，供在 EC2-Classic 中使用。

  类型：字符串

  更新要求：替换 (p. 63) - 在还要更改 `InstanceId` 或 `NetworkInterfaceId` 属性时。如果不更改，则更新要求无中断 (p. 63)。

实例 ID
: 您希望将其与 EIP 属性指定的弹性 IP 地址关联的 Amazon EC2 实例的实例 ID。

  必需：否

  类型：字符串

  更新要求：替换 (p. 63) - 在还要更改 `AllocationId` 或 `EIP` 属性时。如果不更改，则更新要求无中断 (p. 63)。

NetworkInterfaceId
: 将与弹性 IP 地址（仅限 VPC）关联的网络接口的 ID。

  必需：否

  类型：字符串

  更新要求：替换 (p. 63) - 在还要更改 `AllocationId` 或 `EIP` 属性时。如果不更改，则更新要求无中断 (p. 63)。

PrivateIpAddress
: 要与弹性 IP 地址关联的私有 IP 地址。私有 IP 地址仅限与网络接口关联的主辅私有 IP 地址。默认情况下，与 EIP 关联的私有 IP 地址是网络接口的主要私有 IP 地址。

  必需：否

  类型：字符串

  更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

有关 AWS::EC2::EIPAssociation 代码段的信息，请参阅 使用 AWS::EC2::EIP 代码段分配 Amazon EC2 弹性 IP (p. 136)。

AWS::EC2::Instance

Abstract

使用 AWS::EC2::Instance 资源时创建 Amazon EC2 实例。

AWS::EC2::Instance 类型创建 Amazon EC2 实例。

如果您的实例已与某个弹性 IP 地址关联，AWS CloudFormation 会在更新实例后重新关联该弹性 IP 地址。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type" : "AWS::EC2::Instance",
   "Properties" : {
      "AvailabilityZone (p. 273)" : String,
      "BlockDeviceMappings (p. 273)" : [ EC2 Block Device Mapping, ... ],
      "DisableApiTermination (p. 273)" : Boolean,
      "EbsOptimized (p. 273)" : Boolean,
      "IamInstanceProfile (p. 273)" : String,
      "ImageId (p. 274)" : String,
      "InstanceType (p. 274)" : String,
      "KernelId (p. 274)" : String,
      "KeyName (p. 274)" : String,
      "Monitoring (p. 274)" : Boolean,
      "NetworkInterfaces (p. 274)" : [ EC2 Network Interface, ... ],
      "PlacementGroupName (p. 275)" : String,
      "PrivateIpAddress (p. 275)" : String,
      "RamdiskId (p. 275)" : String,
      "SecurityGroupIds (p. 275)" : [ String, ... ],
      "SecurityGroups (p. 275)" : [ String, ... ],
      "SourceDestCheck (p. 276)" : Boolean,
      "SubnetId (p. 276)" : String,
      "Tags (p. 276)" : [ EC2 Tag (p. 453), ... ],
      "Tenancy (p. 276)" : String,
      "UserData (p. 276)" : String,
      "Volumes (p. 276)" : [ EC2 MountPoint (p. 444), ... ]
   }
}
```

属性

可用区
指定实例所在的可用区的名称。

有关 AWS 区域和可用区的更多信息，请参阅 *Amazon EC2 用户指南*中的 Regions and Availability Zones。

Required: No..如果未指定该值，系统将根据该区域的负载平衡标准，自动为您选择一个可用区。

Type: String.

更新要求：替换 (p. 63)

BlockDeviceMappings
定义一组 Amazon Elastic Block Store 块储存设备映射、实例存储块储存设备短暂映射或两者。有关更多信息，请参阅 *Amazon Elastic Compute Cloud 开发人员指南* 中的 Amazon Elastic Block Store 或 Amazon EC2 实例存储。

Required: No.

Type: Amazon EC2 块储存设备映射属性 (p. 440) 列表。

更新要求：替换 (p. 63) - 在要更改 DeleteOnTermination 属性时。如果不更改，则更新要求无中断 (p. 63)。

DisableApiTermination
指定是否可通过 API 终止实例。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

EbsOptimized
指定是否针对 EBS I/O 优化实例。此优化可为 Amazon EBS 提供专用吞吐量，并提供优化的配置堆栈以实现最佳 EBS I/O 性能。

以下实例类型可作为 EBS 优化实例启动：

- 大型 (m1.large)
- 超大型 (m1.xlarge)
- 高内存四倍超大型 (m2.4xlarge)

使用 EBS 优化实例会产生额外费用。有关详细信息，请参阅 *Amazon Elastic Compute Cloud User Guide* 中的 EBS-Optimized Instances。

Required: No..如果该属性未指定，则将使用 "false"。

Type: Boolean.

更新要求：

- *更新要求*：时而中断 (p. 63)（对于由 Amazon EBS 支持的实例）
- *更新要求*：替换 (p. 63)（对于由实例存储支持的实例）

IamInstanceProfile
实例配置文件或对 AWS::IAM::InstanceProfile (p. 346) 资源的引用的物理 ID。

有关 IAM 角色的更多信息，请参阅 AWS Identity and Access Management User Guide 中的*使用角色*。

Required: No.

Type: String.

更新要求：替换 (p. 63)

映像 ID
提供注册期间分配的亚马逊系统映像 (AMI) 的唯一 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

实例类型
实例类型，如 "m1.xlarge"。默认类型为 "m1.small"。要查看实例类型列表，请参阅 Instance Families and Types。

Required: No.

Type: String.

更新要求：

- *更新要求*：时而中断 (p. 63)（对于由 Amazon EBS 支持的实例）
- *更新要求*：替换 (p. 63)（对于由实例存储支持的实例）

KernelId
内核 ID。

Required: No.

Type: String.

更新要求：

- *更新要求*：时而中断 (p. 63)（对于由 Amazon EBS 支持的实例）
- *更新要求*：替换 (p. 63)（对于由实例存储支持的实例）

KeyName
提供 Amazon EC2 密钥对的名称。

Required: No.

Type: String.

更新要求：替换 (p. 63)

正在监控
指明是否已为实例启用了监控。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

NetworkInterfaces
说明与此实例关联的网络接口的 NetworkInterface (p. 445) 嵌入式对象的列表。

Note

如果此资源具有公有 IP 地址并且还处于同一模板中定义的 VPC 内，则您必须使用 DependsOn 属性声明对 VPC 网关连接的依赖关系。有关更多信息，请参阅 DependsOn 属性 (p. 486)。

Required: No.

Type: NetworkInterface (p. 445) 对象的列表。

更新要求：替换 (p. 63)

PlacementGroupName
您希望从中启动实例的现有替换组名称（对于群集实例）。

Required: No.

Type: String.

更新要求：替换 (p. 63)

PrivateIpAddress
此实例的私有 IP 地址。

Important

如果对需要替换的实例进行更新，必须分配新的私有 IP 地址。替换期间，AWS CloudFormation 会创建新实例，但在成功更新堆栈之前不会删除旧实例。如果堆栈更新失败，则 AWS CloudFormation 使用旧实例将该堆栈回滚到上一个工作状态。旧实例和新实例的私有 IP 地址不能相同。

如果您正在使用 Amazon VPC，则可以根据需要使用此参数从子网（例如，10.0.0.25）为实例分配一个特定的可用 IP 地址。默认情况下，Amazon VPC 会从子网中为实例选择一个 IP 地址。

Required: No.

Type: String.

更新要求：替换 (p. 63)

RamdiskId
供选择的 RAM 磁盘的 ID。一些内核启动时需要额外的驱动器。请查看内核要求，了解有关是否需要指定 RAM 磁盘的信息。要查找内核要求，请参阅 AWS 资源中心并搜索相应的内核 ID。

Required: No.

Type: String.

更新要求：

- *更新要求*：时而中断 (p. 63)（对于由 Amazon EBS 支持的实例）
- *更新要求*：替换 (p. 63)（对于由实例存储支持的实例）

SecurityGroupIds
一个列表，其中包含要分配给 Amazon EC2 实例的 VPC 安全组的安全组 ID。如果指定了 `NetworkInterfaces` 属性，则不指定此属性。

必需：条件。VPC 安全组必填。

Type: A list of strings.

更新要求：

- *更新要求*：无中断 (p. 63) - 对于 VPC 中的实例。
- *更新要求*：替换 (p. 63) - 对于不在 VPC 中的实例。

SecurityGroups
仅对 Amazon EC2 安全组有效。一个列表，其中包含要分配给 Amazon EC2 实例的 Amazon EC2 安全组。该列表可能包含现有 Amazon EC2 的名称、对模板中创建的 AWS::EC2::SecurityGroup 资源的引用，或同时包含两者。

Required: No.

Type: A list of strings.

更新要求：替换 (p. 63).

SourceDestCheck
控制是否在实例上启用了源/目标检查，并且确定 VPC 中的实例是否会执行网络地址转换 (NAT)。

"true" 值表明已启用源/目标检查，"false" 值则表明已禁用该检查。对于要执行 NAT 的实例，该值*必须*为 "false"。有关更多信息，请参阅 *Amazon Virtual Private Cloud 用户指南*中的 NAT Instances。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

SubnetId
如果您正在使用 Amazon VPC，则该属性可指定您希望从中启动实例的子网 ID。如果指定了 `NetworkInterfaces` 属性，则不指定此属性。

Required: No.

Type: String.

更新要求：替换 (p. 63)

标签
要与实例关联的标签。

Required: No.

类型：EC2 标签 (p. 453)的列表。

更新要求：无中断 (p. 63)

租期
要启动的实例的租赁。该值可为 "default" 或 "dedicated"。*`tenancy`* 值为 `"dedicated"` 的实例在单租户硬件上运行，并且仅可在 VPC 中启动。有关更多信息，请参阅 *Amazon Virtual Private Cloud User Guide* 中的 Using EC2 Dedicated Instances Within Your VPC。

Required: No.

Type: String.

更新要求：替换 (p. 63)

UserData
适用于实例的 Base64 编码 MIME 用户数据。

Required: No.

Type: String.

更新要求：

- *更新要求*：时而中断 (p. 63)（对于由 Amazon EBS 支持的实例）
- *更新要求*：替换 (p. 63)（对于由实例存储支持的实例）

卷
要与实例关联的 Amazon EBS 卷。

Note

断开卷之前，在操作系统中卸载设备上的任何文件系统。如果不卸载文件系统，则卷可能会在断开期间陷入繁忙状态

Required: No.

类型：EC2 MountPoint (p. 444) 列表。

更新要求：无中断 (p. 63)

返回值

Ref

当您将 AWS::EC2::Instance 对象的逻辑 ID 发送至内部 `Ref` 函数时，将返回该对象的 InstanceId。例如：`i-636be302`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

可用区
: 指定实例启动时所在的可用区。例如：`us-east-1b`.

  您可以使用 Fn::GetAZs (p. 505) 内部函数来检索某个区域的所有可用区的列表。

PrivateDnsName
: 指定实例的私有 DNS 名称。例如：`ip-10-24-34-0.ec2.internal`.

PublicDnsName
: 指定实例的公有 DNS 名称。例如：`ec2-107-20-50-45.compute-1.amazonaws.com`.

PrivateIp
: 指定实例的私有 IP 地址。例如：`10.24.34.0`.

PublicIp
: 指定实例的公有 IP 地址。例如：`192.0.2.0`.

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

具有 EBS Block Device Mapping 的 EC2 实例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Description" : "Ec2 block device mapping",
   "Resources" : {
      "MyEC2Instance" : {
         "Type" : "AWS::EC2::Instance",
         "Properties" : {
            "ImageId" : "ami-79fd7eee",
            "KeyName" : "testkey",
            "BlockDeviceMappings" : [
               {
                  "DeviceName" : "/dev/sdm",
                  "Ebs" : {
                     "VolumeType" : "io1",
                     "Iops" : "200",
```

```
                    "DeleteOnTermination" : "false",
                    "VolumeSize" : "20"
                 }
              },
              {
                 "DeviceName" : "/dev/sdk",
                 "NoDevice" : {}
              }
           ]
        }
     }
  }
}
```

其他示例

您可以下载介绍如何使用 AWS::EC2::Instance 以创建 Virtual Private Cloud (VPC) 的模板：

- 单个子网中的单一实例
- 多个具有 ELB 和 Auto Scaling 组的子网

有关具有 IAM 实例配置文件的 AWS::EC2::Instance 的更多信息，请参阅：Create an EC2 instance with an associated instance profile。

有关 EC2 模板示例的更多信息，请参阅：Amazon EC2 代码段 (p. 132).

另请参阅

- *Amazon Elastic Compute Cloud API Reference* 中的 RunInstances
- *Amazon Elastic Compute Cloud User Guide* 中的 EBS-Optimized Instances

AWS::EC2::InternetGateway

Abstract

使用 AWS::EC2::InternetGateway 资源在您的 AWS 账户中创建新 Internet 网关。

在您的 AWS 账户中创建新的 Internet 网关。创建 Internet 网关后，您可以将它连接到 VPC。

语法

```
{
   "Type" : "AWS::EC2::InternetGateway",
   "Properties" : {
      "Tags (p. 278)" : [ { "Key" : "keyname1", "Value" : "value1" }, ... ]
   }
}
```

属性

标签
    要与此资源相连接的标签。

Type: EC2 标签 (p. 453)的列表。

Required: No.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myInternetGateway" : {
         "Type" : "AWS::EC2::InternetGateway",
         "Properties" : {
            "Tags" : [ {"Key" : "foo", "Value" : "bar"}]
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateInternetGateway。
- *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags。

AWS::EC2::NetworkAcl

Abstract

使用 AWS::EC2::NetworkAcl 资源在 VPC 中创建新网络 ACL。

在 VPC 中创建新的 ACL。

语法

```
{
   "Type" : "AWS::EC2::NetworkAcl",
   "Properties" : {
      "Tags (p. 280)" : [ EC2 Tags (p. 453) ],
      "VpcId (p. 280)" : String
   }
}
```

属性

标签
: 要与此资源相连接的标签。

  有关标签的更多信息，请参阅 *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags。

  Required: No.

  Type: EC2 标签 (p. 453)的列表

  更新要求：无中断 (p. 63)

VpcId
: 要在其中创建网络 ACL 的 VPC 的 ID。

  Required: Yes.

  Type: String.

  更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myNetworkAcl" : {
         "Type" : "AWS::EC2::NetworkAcl",
         "Properties" : {
            "VpcId" : { "Ref" : "myVPC" },
            "Tags" : [ { "Key" : "foo", "Value" : "bar" } ]
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateNetworkAcl
- *Amazon Virtual Private Cloud User Guide* 中的 Network ACLs。

AWS::EC2::NetworkAclEntry

Abstract

使用 AWS::EC2::NetworkAclEntry 资源在网络 ACL 中创建具有指定规则编号的条目。

使用您指定的规则编号在网络 ACL 中创建条目（如，规则）。每个网络 ACL 都拥有一套已编号的输入规则和单独的一套已编号输出规则。

语法

```
{
   "Type" : "AWS::EC2::NetworkAclEntry",
   "Properties" : {
      "CidrBlock (p. 281)" : String,
      "Egress (p. 281)" : Boolean,
      "Icmp (p. 281)" : EC2 ICMP,
      "NetworkAclId (p. 281)" : String,
      "PortRange (p. 282)" : EC2 PortRange,
      "Protocol (p. 282)" : Integer,
      "RuleAction (p. 282)" : String,
      "RuleNumber (p. 282)" : Integer
   }
}
```

属性

CidrBlock
要接受或拒绝的 CIDR 范围（以 CIDR 表示法显示，如 172.16.0.0/24）。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

输出
该规则是否应用到子网的输出流量 ("true") 或应用到子网的输入流量 ("false")。

Required: Yes.

Type: Boolean.

更新要求：替换 (p. 63).

Icmp
互联网控制信息协议 (ICMP) 代码和类型。

Required: Conditional. 如果为该协议参数指定 1 (ICMP)，则是必需的。

Type: EC2 ICMP 属性类型 (p. 443)

更新要求：无中断 (p. 63)

NetworkAclId
将在其中创建条目的 ACL 的 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63).

PortRange
适用于 UDP/TCP 协议的端口编号范围。

Required: Conditional. 如果为该协议参数指定 6 (TCP) 或 17 (UDP)，则是必需的。

Type: EC2 PortRange 属性类型 (p. 449)

更新要求：无中断 (p. 63)

协议
该规则适用 IP 协议。您可以使用 -1 来表示所有协议。此值必需是 -1 或协议号（请参阅 iana.org 中的协议号）。

Required: Yes.

Type: Number.

更新要求：无中断 (p. 63)

RuleAction
是否允许或拒绝与该规则匹配的流量；有效值为“allow”或“deny”。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

RuleNumber
分配给条目的规则编号（例如 100）。此值必须是 1 至 32766 之间的整数。

Required: Yes.

Type: Number.

更新要求：替换 (p. 63).

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myNetworkAclEntry" : {
         "Type" : "AWS::EC2::NetworkAclEntry",
         "Properties" : {
            "NetworkAclId" : { "Ref" : "myNetworkAcl" },
            "RuleNumber" : "100",
```

```
            "Protocol" : "-1",
            "RuleAction" : "allow",
            "Egress" : "true",
            "CidrBlock" : "172.16.0.0/24",
            "Icmp" : { "Code" : "-1", "Type" : "-1" },
            "PortRange" : { "From" : "53", "To" : "53" }
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 NetworkAclEntry
- *Amazon Virtual Private Cloud User Guide* 中的 Network ACLs。

AWS::EC2::NetworkInterface

Abstract

使用 AWS::EC2::NetworkInterface 资源描述 Amazon EC2 实例中的网络接口。

描述 AWS CloudFormation 的 Elastic Compute Cloud (EC2) 实例中的网络接口。
AWS::EC2::Instance (p. 272) 的 NetworkInterfaces 属性中的一个列表中提供了此接口。

语法

```
{
   "Type" : "AWS::EC2::NetworkInterface",
   "Properties" : {
      "Description (p. 283)" : String,
      "GroupSet (p. 284)" : [ String, ... ],
      "PrivateIpAddress (p. 284)" : String,
      "PrivateIpAddresses (p. 284)" : [ PrivateIpAddressSpecification, ... ],
      "SecondaryPrivateIpAddressCount (p. 284)" : Integer,
      "SourceDestCheck (p. 284)" : Boolean,
      "SubnetId (p. 284)" : String,
      "Tags (p. 285)" : [ EC2 Tags, ...],
   }
}
```

属性

说明
: 有关此网络接口的说明。

  Required: No.

  Type: String.

  更新要求：无中断 (p. 63).

GroupSet
此网络接口相关的安全组 ID 列表。

Required: No.

类型：字符串列表。

更新要求：无中断 (p. 63)

PrivateIpAddress
将单个私有 IP 地址分配给网络接口，该地址用作主要私有 IP 地址。如果您想指定多个私有 IP 地址，请使用 `PrivateIpAddresses` 属性。

Required: No.

Type: String.

更新要求：替换 (p. 63).

PrivateIpAddresses
将一组私有 IP 地址分配给网络接口。您可以通过在 `PrivateIpAddressSpecification` 属性中将 `Primary` 属性的值设置为 `true` 来指定主要私有 IP 地址。如果您要让 Amazon EC2 自动分配专用 IP 地址，请使用 `SecondaryPrivateIpAddressCount` 属性，而不要指定此属性。

有关私有 IP 地址的最大数量的信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的 Private IP Addresses Per ENI Per Instance Type。

Required: No.

类型：PrivateIpAddressSpecification (p. 449) 的列表。

更新要求：替换 (p. 63) - 在更改主要私有 IP 地址时。如果不更改，则更新要求无中断 (p. 63)。

SecondaryPrivateIpAddressCount
Amazon EC2 自动分配给该网络接口的辅助专用 IP 地址的数目。Amazon EC2 使用 `PrivateIpAddress` 属性的值以作为主专用 IP 地址。如果您不指定该属性，则 Amazon EC2 会自动分配主专用 IP 地址和辅助专用 IP 地址。

如果您想指定自己的私有 IP 地址列表，请使用 `PrivateIpAddresses` 属性，并且不指定此属性。

有关私有 IP 地址的最大数量的信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的 Private IP Addresses Per ENI Per Instance Type。

Required: No.

类型：整数。

更新要求：无中断 (p. 63).

SourceDestCheck
表示实例的进出流量是否有效的标记。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63).

SubnetId
要与网络接口关联的子网的 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63).

标签
与此网络相关的标记的列表。

Required: No.

Type: EC2 标签 (p. 453) 的列表。

更新要求：无中断 (p. 63).

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

PrimaryPrivateIpAddress
返回网络接口的主要私有 IP 地址。例如，`10.0.0.192`。

SecondaryPrivateIpAddresses
返回网络接口的辅助私有 IP 地址。例如，`["10.0.0.161", "10.0.0.162", "10.0.0.163"]`。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

Tip

有关更多 NetworkInterface 模板示例，请参阅 Elastic Network Interface (ENI) 模板代码段 (p. 137)。

简单的独立 ENI

这是一个使用所有可用属性的简单的独立 Elastic Network Interface (ENI)。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Description" : "Simple Standalone ENI",
   "Resources" : {
      "myENI" : {
         "Type" : "AWS::EC2::NetworkInterface",
         "Properties" : {
            "Tags": [{"Key":"foo","Value":"bar"}],
            "Description": "A nice description.",
            "SourceDestCheck": "false",
            "GroupSet": ["sg-75zzz219"],
            "SubnetId": "subnet-3z648z53",
            "PrivateIpAddress": "10.0.0.16"
```

```
         }
      }
   }
}
```

EC2 实例上的 ENI

此为 EC2 实例上的 ENI 示例。在此示例中，会将一个 ENI 添加到该实例中。如果您想添加一个以上的 ENI，则可以为 `NetworkInterface` 属性指定一个列表。但是，只有在所有 ENI 都仅有专用 IP 地址（没有关联的公共 IP 地址）的情况下，才可以指定多个 ENI。如果您有带公共 IP 地址的 ENI，请将其指定，然后使用 `AWS::EC2::NetworkInterfaceAttachment` 资源来添加其他 ENI。

```
"Ec2Instance" : {
   "Type" : "AWS::EC2::Instance",
   "Properties" : {
      "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
}, "AMI" ]},
      "KeyName" : { "Ref" : "KeyName" },
      "SecurityGroupIds" : [{ "Ref" : "WebSecurityGroup" }],
      "SubnetId" : { "Ref" : "SubnetId" },
      "NetworkInterfaces" : [ {
         "NetworkInterfaceId" : {"Ref" : "controlXface"}, "DeviceIndex" : "1"
} ],
      "Tags" : [ {"Key" : "Role", "Value" : "Test Instance"}],
      "UserData" : { "Fn::Base64" : { "Ref" : "WebServerPort" }}
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API Reference* 中的 NetworkInterfaceType

AWS::EC2::NetworkInterfaceAttachment

Abstract

使用 AWS::EC2::NetworkInterfaceAttachment 资源将弹性网络接口 (ENI) 附加到 Amazon EC2 实例。

将弹性网络接口 (ENI) 附加到 Amazon EC2 实例。您可以使用此资源类型，在不中断操作的情况下将其他网络接口附加到实例。

语法

```
{
   "Type" : "AWS::EC2::NetworkInterfaceAttachment",
   "Properties" : {
      "DeleteOnTermination (p. 287)": Boolean,
      "DeviceIndex (p. 287)": String,
      "InstanceId (p. 287)": String,
      "NetworkInterfaceId (p. 287)": String,
   }
}
```

属性

DeleteOnTermination
是否要在实例终止时删除网络接口。默认情况下，此值设置为 `True`。

Required: No.

类型：布尔值。

更新要求：无中断 (p. 63)

DeviceIndex
网络接口在关联顺序中的位置。例如，第一个附加的网络接口的 `DeviceIndex` 为 `0`。

必需：可以。

类型：字符串。

更新要求：无中断 (p. 63)

实例 ID
将 ENI 附加到的实例的 ID。

必需：可以。

类型：字符串。

更新要求：无中断 (p. 63)

NetworkInterfaceId
要附加的 ENI 的 ID。

必需：可以。

类型：字符串。

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example 将 **`MyNetworkInterface`** 附加到 **`MyInstance`**

```
"NetworkInterfaceAttachment" : {
    "Type" : "AWS::EC2::NetworkInterfaceAttachment",
        "Properties" : {
            "InstanceId" : {"Ref" : "MyInstance"},
            "NetworkInterfaceId" : {"Ref" : "MyNetworkInterface"},
            "DeviceIndex" : "1"
        }
}
```

AWS::EC2::Route

Abstract

使用 AWS::EC2::Route 资源在 VPC 内的路由表中创建路由。

在 VPC 中的路由表内创建新的路由。此路由目标既可以是连接到 VPC 的网关，也可以连接 VPC 中的 NAT 实例。

语法

```
{
   "Type" : "AWS::EC2::Route",
   "Properties" : {
      "DestinationCidrBlock (p. 288)" : String,
      "GatewayId (p. 288)" : String,
      "InstanceId (p. 288)" : String,
      "NetworkInterfaceId (p. 288)" : String,
      "RouteTableId (p. 289)" : String
   }
}
```

属性

DestinationCidrBlock
用于此目标匹配的 CIDR 地址块。例如，`"0.0.0.0/0"`。路由判断是根据最具体的匹配确定的。

必需：是

类型：字符串

更新要求：替换 (p. 63)

网关 ID
与您的 VPC 关联的网关的 ID。例如：`"igw-eaad4883"`.

对于指定网关的路由条目，您必须指定对网关连接资源的依赖关系。有关更多信息，请参阅 DependsOn 属性 (p. 486)。

必需：条件。在以下项目中，您必须*仅提供一个*：`GatewayID`、`InstanceID` 或 `NetworkInterfaceId`。

类型：字符串

更新要求：无中断 (p. 63)

实例 ID
您的 VPC 中的 NAT 实例。例如，`"i-1a2b3c4d"`。

必需：条件。在以下项目中，您必须*仅提供一个*：`GatewayID`、`InstanceID` 或 `NetworkInterfaceId`。

类型：字符串

更新要求：无中断 (p. 63)

NetworkInterfaceId
允许网络接口 ID 路由。

必需：条件。在以下项目中，您必须*仅提供一个*：`GatewayID`、`InstanceID` 或 `NetworkInterfaceId`。

类型：字符串

更新要求：无中断 (p. 63)

RouteTableId
将添加路由的路由表 (p. 290)的 ID。

必需：是

类型：字符串

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example 带有网关 ID 的路由

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myRoute" : {
         "Type" : "AWS::EC2::Route",
         "DependsOn" : "GatewayToInternet",
         "Properties" : {
            "RouteTableId" : { "Ref" : "myRouteTable" },
            "DestinationCidrBlock" : "0.0.0.0/0",
            "GatewayId" : { "Ref" : "myInternetGateway" }
         }
      }
   }
}
```

Example 带有实例 ID 的路由

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myRoute" : {
         "Type" : "AWS::EC2::Route",
         "Properties" : {
            "RouteTableId" : { "Ref" : "myRouteTable" },
            "DestinationCidrBlock" : "0.0.0.0/0",
            "InstanceId" : { "Ref" : "myInstance" }
         }
      }
   }
}
```

Example 带有网络接口 ID 的路由。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myRoute" : {
         "Type" : "AWS::EC2::Route",
         "Properties" : {
            "RouteTableId" : { "Ref" : "myRouteTable" },
            "DestinationCidrBlock" : "0.0.0.0/0",
            "NetworkInterfaceId" : { "Ref" : "eni-1a2b3c4d" }
         }
      }
   }
}
```

另请参阅

- AWS::EC2::RouteTable (p. 290)
- *Amazon Elastic Compute Cloud API 参考* 中的 CreateRoute
- *Amazon Virtual Private Cloud User Guide* 中的 Route Tables。

AWS::EC2::RouteTable

Abstract

使用 AWS::EC2::RouteTable 资源在 VPC 中创建路由表。

在 VPC 内创建新的路由表。当您创建新路由表后，您可以添加路由并将该表与子网关联。

语法

```
{
   "Type" : "AWS::EC2::RouteTable",
```

```
   "Properties" : {
      "VpcId (p. 291)" : String,
      "Tags (p. 291)" : [ EC2 Tag, ... ]
   }
}
```

属性

VpcId
: 要在其中创建路由表的 VPC 的 ID。

  示例：vpc-11ad4878

  必需：是

  类型：字符串

  更新要求：替换 (p. 63)

标签
: 要与此资源相连接的标签。

  必需：否

  类型：EC2 标签 (p. 453)的列表。

  更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example

下面的示例代码段使用在同一模板的其他位置声明过的名为 *myVPC* 的 VPC 中的 VPC ID。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myRouteTable" : {
         "Type" : "AWS::EC2::RouteTable",
         "Properties" : {
            "VpcId" : { "Ref" : "myVPC" },
            "Tags" : [ { "Key" : "foo", "Value" : "bar" } ]
         }
      }
   }
}
```

另请参阅

- AWS::EC2::Route (p. 288)
- *Amazon Elastic Compute Cloud API 参考* 中的 CreateRouteTable
- *Amazon Virtual Private Cloud User Guide* 中的 Route Tables
- *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags

AWS::EC2::SecurityGroup

Abstract

使用 AWS::EC2::SecurityGroup 资源创建 Amazon EC2 安全组。

创建一个 Amazon EC2 安全组。要创建 VPC 安全组，请使用 VpcId (p. 293) 属性。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" :
   {
      "GroupDescription (p. 292)" : String,
      "SecurityGroupEgress (p. 292)" : [ Security Group Rule, ... ],
      "SecurityGroupIngress (p. 292)" : [ Security Group Rule, ... ],
      "   (p. 293)" :  [ Resource Tag, ... ],
      "VpcId (p. 293)" : String
   }
}
```

属性

GroupDescription
安全组的说明。

类型：字符串

必需：是

更新要求：替换 (p. 63)

SecurityGroupEgress
Amazon EC2 安全组传出规则列表。

类型：EC2 安全组规则 (p. 450)

必需：否

更新要求：无中断 (p. 63)

SecurityGroupIngress
Amazon EC2 安全组传入规则列表。

类型：EC2 安全组规则 (p. 450)

必需：否

更新要求：无中断 (p. 63)

标签
需要附加到资源的标签。

Required: No.

类型：AWS CloudFormation 资源标签 (p. 471).

更新要求：无中断 (p. 63).

VpcId
VPC 的物理 ID。可使用对 AWS::EC2::VPC (p. 310) 的引用获取，如：`{ "Ref" : "myVPC" }`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

类型：字符串

必需：对于 VPC 安全组为必需

更新要求：替换 (p. 63)

Note

有关 VPC 安全组的更多信息，请参阅 *Amazon Virtual Private Cloud User Guide* 中的 Security Groups。

返回值

Ref

指定 AWS::EC2::SecurityGroup 类型作为 `Ref` 函数的参数时，AWS CloudFormation 返回 VPC 安全组的安全组名称或安全组 ID。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

GroupId
指定安全组的组 ID，如 `sg-94b3a1f6`。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

AWS::EC2::SecurityGroup 是 AWS CloudFormation 模板内的顶层元素。示例如下：

```
"InstanceSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" : {
      "GroupDescription" : "Allow http to client host",
      "VpcId" : {"Ref" : "myVPC"},
      "SecurityGroupIngress" : [{
```

```
            "IpProtocol" : "tcp",
            "FromPort" : "80",
            "ToPort" : "80",
            "CidrIp" : "0.0.0.0/0"
         }],
      "SecurityGroupEgress" : [{
         "IpProtocol" : "tcp",
         "FromPort" : "80",
         "ToPort" : "80",
         "CidrIp" : "0.0.0.0/0"
      }]
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud 用户指南* 中的 Using Security Groups。
- *Amazon Virtual Private Cloud User Guide* 中的 Security Groups。

AWS::EC2::SecurityGroupEgress

Abstract

使用 AWS::EC2::SecurityGroupEgress 资源向 Amazon VPC 安全组添加传出规则。

类型 AWS::EC2::SecurityGroupEgress 用于向 Amazon VPC 安全组添加传出规则。

有关向 Amazon VPC 安全组添加传出规则的更多信息，请参阅 *Amazon Elastic Compute Cloud API 参考* 中的 AuthorizeSecurityGroupEgress。

Important

仅在必要时使用 `AWS::EC2::SecurityGroupIngress` 和 `AWS::EC2::SecurityGroupEgress`，它们通常允许安全组在传入和传出规则中相互引用。否则，请使用 AWS::EC2::SecurityGroup (p. 292) 的嵌入式传入和传出规则。有关更多信息，请参阅 Amazon EC2 Security Groups。

语法

```
{
   "CidrIp (p. 295)" : String,
   "DestinationSecurityGroupId (p. 295)" : String,
   "FromPort (p. 295)" : Number,
   "GroupId (p. 295)" : String,
   "IpProtocol (p. 295)" : String,
   "ToPort (p. 295)" : Number
}
```

属性

Note

如果您更改逻辑 ID，则还必须更新属性值以便为此资源触发更新。

CidrIp
CIDR 范围。

Type: String.

Required: Conditional. 不能在指定目标安全组时使用。

更新要求：替换 (p. 63)

DestinationSecurityGroupId
指定目标 Amazon VPC 安全组的 GroupId。

Type: String.

Required: Conditional. 不能在指定 CIDR IP 地址时使用。

更新要求：替换 (p. 63)

FromPort
TCP 和 UDP 协议端口范围的起始端口，或者某个 ICMP 类型编号。如果为 `IpProtocol` 属性指定 `icmp`，则可以将 -1 指定为通配符（即，任何 ICMP 类型编号）。

Type: String.

Required: Yes.

更新要求：替换 (p. 63)

GroupId
要修改的 Amazon VPC 安全组的 ID。此值可以是对具有有效 VpcId 属性的 AWS::EC2::SecurityGroup (p. 292) 资源的引用，也可以是现有 Amazon VPC 安全组的 ID。

Type: String.

Required: Yes.

更新要求：替换 (p. 63)

IpProtocol
IP 协议名称或编号。要查看有效值，请参阅 AuthorizeSecurityGroupIngress 中的 IpProtocol 参数。

Type: String.

Required: Yes.

更新要求：替换 (p. 63)

ToPort
TCP 和 UDP 协议端口范围的终止端口，或者某个 ICMP 代码。如果为 `IpProtocol` 属性指定 `icmp`，则可以将 -1 指定为通配符（即，任何 ICMP 代码）。

Type: String.

Required: Yes.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

VPC 安全组和传出规则

下面的模板代码段使用一个传出规则来创建 VPC 安全组，该传出规则允许该安全组中任何其他主机从端口 80 传出流量。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "SGBase": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId" : "vpc-e063f789",
        "GroupDescription": "Base Security Group",
        "SecurityGroupEgress": [
          {
            "IpProtocol": "tcp",
            "CidrIp": "0.0.0.0/0",
            "FromPort": "22",
            "ToPort": "22"
          }
        ]
      }
    },
    "SGBaseEgress": {
      "Type": "AWS::EC2::SecurityGroupEgress",
      "Properties": {
        "GroupId": { "Ref": "SGBase" },
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "DestinationSecurityGroupId": { "Ref": "SGBase" }
      }
    }
  }
}
```

AWS::EC2::SecurityGroupIngress

Abstract

使用 AWS::EC2:SecurityGroupIngress 属性类型向 Amazon EC2 或 VPC 安全组添加传入规则。

AWS::EC2::SecurityGroupIngress 类型可为 Amazon EC2 或 VPC 安全组添加传入规则。

有关为 Amazon EC2 或 VPC 安全组添加传入规则的更多信息，请参阅 *Amazon Elastic Compute Cloud API 参考* 中的 AuthorizeSecurityGroupIngress。

Important

仅在必要时使用 `AWS::EC2::SecurityGroupIngress` 和 `AWS::EC2::SecurityGroupEgress`，它们通常允许安全组在传入和传出规则中相互引用。否则，请使用 AWS::EC2::SecurityGroup (p. 292) 的嵌入式传入和传出规则。有关更多信息，请参阅 Amazon EC2 Security Groups。

语法

```
{
   "GroupName (p. 297)" : String
   "GroupId (p. 297)" : String
   "IpProtocol (p. 297)" : String
   "CidrIp (p. 297)" : String
   "SourceSecurityGroupName (p. 298)" : String
   "SourceSecurityGroupId (p. 298)" : String
   "SourceSecurityGroupOwnerId (p. 298)" : String
   "FromPort (p. 298)" : Number
   "ToPort (p. 298)" : Number
}
```

属性

Note

如果您更改逻辑 ID，则还必须更新属性值以便为此资源触发更新。

GroupName
要修改的 EC2 安全组的名称。该值可以是对 AWS::EC2::SecurityGroup (p. 292) 资源的引用，也可以是现有 EC2 安全组的名称。
类型：字符串
必需：可以用来*替代* EC2 安全组的 GroupId。
更新要求：替换 (p. 63)

GroupId
要修改的 EC2 或 VPC 安全组的 ID。该组必须属于您的账户。
类型：字符串
必需：对于 VPC 安全组为必需；可以用来替代 EC2 安全组的 GroupName
更新要求：替换 (p. 63)

IpProtocol
IP 协议名称或编号。要查看有效值，请参阅 AuthorizeSecurityGroupIngress 中的 IpProtocol 参数。
类型：字符串
必需：是
更新要求：替换 (p. 63)

CidrIp
指定一个 CIDR 范围。
有关 CIDR 范围的概述，请访问 Wikipedia Tutorial。
条件：如果您指定了 SourceSecurityGroupName，就不要指定 CidrIp。

类型：字符串

必需：有条件 - 如果您指定 SourceSecurityGroupName，就不要指定 CidrIp。

更新要求：替换 (p. 63)

SourceSecurityGroupName
指定 Amazon EC2 安全组的名称以便访问，或者使用 Ref 内部函数以引用同一模板中定义的安全组的逻辑名称。对于 VPC 中的实例，请指定 `SourceSecurityGroupId` 属性。

类型：字符串

必需：有条件 - 如果您指定 CidrIp，就不要指定 SourceSecurityGroupName。

更新要求：替换 (p. 63)

SourceSecurityGroupId
指定源安全组的 ID 或使用 Ref 内部函数，以引用同一模板中定义的安全组的逻辑名称。

条件：如果已指定 CidrIp，则不需再指定 SourceSecurityGroupId。

类型：字符串

必需：有条件 - 如果您指定 CidrIp，就不要指定 SourceSecurityGroupId。

更新要求：替换 (p. 63)

SourceSecurityGroupOwnerId
指定在 SourceSecurityGroupName 属性中所指定 Amazon EC2 安全组的所有者的 AWS 账户 ID。

类型：字符串

必需：有条件 - 如果指定 SourceSecurityGroupName，并且该安全组的所有者并非创建堆栈的账户，则必须指定 SourceSecurityGroupOwnerId；否则，可根据需要选择是否指定此属性。

更新要求：替换 (p. 63)

FromPort
TCP 和 UDP 协议端口范围的起始端口，或者某个 ICMP 类型编号。如果为 `IpProtocol` 属性指定 `icmp`，则可以将 -1 指定为通配符（即，任何 ICMP 类型编号）。

类型：字符串

必需：是，对于 ICMP 和使用端口的任何协议。

更新要求：替换 (p. 63)

ToPort
TCP 和 UDP 协议端口范围的终止端口，或者某个 ICMP 代码。如果为 `IpProtocol` 属性指定 `icmp`，则可以将 -1 指定为通配符（即，任何 ICMP 代码）。

类型：字符串

必需：是，对于 ICMP 和使用端口的任何协议。

更新要求：替换 (p. 63)

示例

EC2 安全组和传入规则

要创建 Amazon EC2（非 VPC）安全组和传入规则，请在传入规则中使用 SourceSecurityGroupName 属性。

以下模板代码段可创建具有以下特点的 EC2 安全组：其传入规则允许安全组中的任何其他主机从端口 80 传入流量。该代码段使用内部函数 Ref (p. 508) 来指定 SourceSecurityGroupName 的值。

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
```

```
        "SGBase": {
            "Type": "AWS::EC2::SecurityGroup",
            "Properties": {
                "GroupDescription": "Base Security Group",
                "SecurityGroupIngress": [
                    {
                        "IpProtocol": "tcp",
                        "CidrIp": "0.0.0.0/0",
                        "FromPort": "22",
                        "ToPort": "22"
                    }
                ]
            }
        },
        "SGBaseIngress": {
            "Type": "AWS::EC2::SecurityGroupIngress",
            "Properties": {
                "GroupName": { "Ref": "SGBase" },
                "IpProtocol": "tcp",
                "FromPort": "80",
                "ToPort": "80",
                "SourceSecurityGroupName": { "Ref": "SGBase" }
            }
        }
    }
}
```

VPC 安全组和传入规则

要创建 Amazon VPC 安全组和传入规则，请执行以下操作：

- 在安全组中指定 VpcId 属性。
- 在传入规则中使用 SourceSecurityGroupId 属性。

以下模板代码段可创建具有以下特点的 VPC 安全组：其传入规则允许安全组中的任何其他主机从端口 80 传入流量。该代码段使用内部函数 Ref (p. 508) 来指定 SourceSecurityGroupId 的值。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "SGBase": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "VpcId" : "vpc-12345678",
        "GroupDescription": "Base Security Group",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "CidrIp": "0.0.0.0/0",
            "FromPort": "22",
            "ToPort": "22"
          }
        ]
      }
    },
    "SGBaseIngress": {
```

```
      "Type": "AWS::EC2::SecurityGroupIngress",
      "Properties": {
        "GroupId": { "Ref": "SGBase" },
        "IpProtocol": "tcp",
        "FromPort": "80",
        "ToPort": "80",
        "SourceSecurityGroupId": { "Ref": "SGBase" }
      }
    }
  }
}
```

允许 Ping 请求

若要允许 Ping 请求，请添加 ICMP 协议类型，为 ICMP 类型指定 8（回显请求），为 ICMP 代码指定 0 或 -1（全部）。

```
"SGPing" : {
  "Type" : "AWS::EC2::SecurityGroup",
  "DependsOn": "VPC",
  "Properties" : {
    "GroupDescription" : "SG to test ping",
    "VpcId" : {"Ref" : "VPC"},
    "SecurityGroupIngress" : [
      { "IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" :
"10.0.0.0/24" },
      { "IpProtocol" : "icmp", "FromPort" : "8", "ToPort" : "-1", "CidrIp" :
"10.0.0.0/24" }
    ]
  }
}
```

AWS::EC2::Subnet

Abstract

使用 AWS::EC2::Subnet 资源在现有 VPC 中创建子网。

可在现有 VPC 中创建子网。

语法

```
{
   "Type" : "AWS::EC2::Subnet",
   "Properties" : {
      "AvailabilityZone (p. 301)" : String,
      "CidrBlock (p. 301)" : String,
      "Tags (p. 301)" : [ EC2 Tag, ... ],
      "VpcId (p. 301)" : { "Ref" : String }
   }
}
```

属性

可用区
您要在其中创建子网的可用区。默认值：AWS 为您选择区域（推荐）。

必需：否

类型：字符串

更新要求：替换 (p. 63)

Note

如果您更新此属性，则还必须更新 `CidrBlock` 属性。

CidrBlock
您要让该子网覆盖的 CIDR 块（例如，`"10.0.0.0/24"`）。

必需：是

类型：字符串

更新要求：替换 (p. 63)

Note

如果您更新此属性，则还必须更新 `AvailabilityZone` 属性。

标签
您要添加到此资源的标签。

必需：否

类型：EC2 标签 (p. 453)的列表。

更新要求：无中断 (p. 63)

VpcId
Ref 结构，其中包含您要在其上创建子网的 VPC 的 ID。提供 VPC ID 作为“Ref”属性的值，例如：`{ "Ref": "VPCID" }`.

必需：是

类型：引用 ID

更新要求：替换 (p. 63)

Note

如果您更新此属性，则还必须更新 `CidrBlock` 属性。

返回值

您可以将资源的逻辑 ID 传送给内部函数，获得从资源返回的值。这一返回的值取决于所用函数。

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 Ref 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

Fn::GetAtt返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

可用区
返回此子网的可用区（例如，"us-east-1a"）。

示例：

```
{ "Fn::GetAtt" : [ "mySubnet", "AvailabilityZone" ] }
```

有关使用 Fn:GetAtt 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

下面的示例代码段使用在同一模板的其他位置声明过的名为 *myVPC* 的 VPC 中的 VPC ID。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "mySubnet" : {
         "Type" : "AWS::EC2::Subnet",
         "Properties" : {
            "VpcId" : { "Ref" : "myVPC" },
            "CidrBlock" : "10.0.0.0/24",
            "AvailabilityZone" : "us-east-1a",
            "Tags" : [ { "Key" : "foo", "Value" : "bar" } ]
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateSubnet
- *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags

AWS::EC2::SubnetNetworkAclAssociation

Abstract

使用 AWS::EC2::SubnetNetworkAclAssociation 资源将子网与网络 ACL 关联。

关联子网与网络 ACL。

有关更多信息，请参阅 *Amazon Elastic Compute Cloud API 参考* 中的 ReplaceNetworkAclAssociation。

Note

EC2 API 参考中将 *SubnetId* 参数作为 *AssociationId*。

语法

```
"Type" : "AWS::EC2::SubnetNetworkAclAssociation",
"Properties" : {
   "SubnetId (p. 303)" : { String }
   "NetworkAclId (p. 303)" : { String }
}
```

属性

SubnetId
: 表示原始网络 ACL 和子网之间当前关联性的 ID。
必需：是
类型：字符串
更新要求：替换 (p. 63)

NetworkAclId
: 要关联到子网的新 ACL 的 ID。
必需：是
类型：字符串
更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

AssociationId
: 返回此对象的 SubnetId (p. 303) 属性的值。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

Example

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "mySubnetNetworkAclAssociation" : {
         "Type" : "AWS::EC2::SubnetNetworkAclAssociation",
         "Properties" : {
            "SubnetId" : { "Ref" : "mySubnet" },
            "NetworkAclId" : { "Ref" : "myNetworkAcl" },
         }
      }
   }
}
```

AWS::EC2::SubnetRouteTableAssociation

Abstract

使用 AWS::EC2::SubnetRouteTableAssociation 资源将子网与路由表关联。

将子网与路由表关联

语法

```
{
   "Type" : "AWS::EC2::SubnetRouteTableAssociation",
   "Properties" : {
      "RouteTableId (p. 304)" : String,
      "SubnetId (p. 304)" : String,
   }
}
```

属性

RouteTableId
路由表的 ID。此 ID 通常会写为在模板的其他位置声明的路由表的引用。例如：

```
"RouteTableId" : { "Ref" : "myRouteTable" }
```

必需：是

类型：字符串

更新要求：无中断 (p. 63).然而，当路由表 ID 发生更改时，物理 ID 也会随之更改。

SubnetId
子网的 ID。此 ID 通常会写为对在模板其他位置声明的子网的引用。例如：

```
"SubnetId" : { "Ref" : "mySubnet" }
```

必需：是

类型：字符串

更新要求：替换 (p. 63)

返回值

当该资源的逻辑 ID 提供给 Ref 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyRTA" }
```

对于与逻辑 ID “MyRTA”关联的子网路由表，Ref 将返回 AWS 资源名称。

有关使用 Ref 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "mySubnetRouteTableAssociation" : {
         "Type" : "AWS::EC2::SubnetRouteTableAssociation",
         "Properties" : {
            "SubnetId" : { "Ref" : "mySubnet" },
            "RouteTableId" : { "Ref" : "myRouteTable" }
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 AssociateRouteTable

AWS::EC2::Volume

Abstract

使用 AWS::EC2::Volume 创建 Amazon EBS 卷。

AWS::EC2::Volume 类型可创建新的 Elastic Block Store 卷。

您可以为卷设置一个删除策略，以控制 AWS CloudFormation 在堆栈被删除时如何处理卷。对于 Elastic Block Store 卷，您可以选择*保留*卷、*删除*卷或为卷*创建快照*。有关更多信息，请参阅 DeletionPolicy 属性 (p. 485)。

Note

如果您设置的删除策略是创建快照，卷上的所有标签都会包含在快照中。

语法

```
{
   "Type":"AWS::EC2::Volume",
   "Properties" : {
      "AvailabilityZone (p. 306)" : String,
      "Iops (p. 306)" : Number,
      "Size (p. 306)" : String,
      "SnapshotId (p. 306)" : String,
      "Tags (p. 307)" : [ EC2 Tag, ...  ],
      "VolumeType (p. 307) : String
   }
}
```

属性

可用区
从中创建卷的可用区。

Required: Yes.

Type: String.

更新要求：不支持更新

Iops
卷支持的每秒 I/O 操作数 (IOPS)。它可以是 1–4000 之间的任何整数值。

Required: Conditional. 如果卷类型为 "io1"，则为*必需属性*；不与标准卷配合使用。

Type: Number.

更新要求：不支持更新

大小
卷的大小，以 gibibytes (GiB) 为单位。该参数可为 10–1024 之间的任何值。

Note

EBS 卷的大小必须能够配合您所需的 IOPS。IOPS 和存储量（吉字节，GiB）的比率为 10:1，因此 PIOPS 要达到 100，至少需要 10 GiB 根卷存储量。

Required: Conditional. 如果您不从快照中创建卷，则为*必需属性*。如果您已指定 Size，则*不要*指定 SnapshotId。

Type: String.

更新要求：不支持更新

SnapshotId
从中创建新卷的快照。

Required: Conditional. 如果您要从快照中创建卷，则为*必需属性*。如果您没有为“SnapshotId”指定值，则必须为“Size”指定值。

Type: String.

更新要求：不支持更新

标签
: 要与卷关联的标签。

Required: No.

类型：EC2 标签 (p. 453)的列表。

更新要求：不支持更新

VolumeType
: 卷类型。该参数可为 "standard" 或 "io1"。如果未指定值，则将使用 "standard"。

Required: No.

Type: String.

更新要求：不支持更新

返回值

Ref

如果将 AWS::EC2::Volume 类型指定为 `Ref` 函数的参数，AWS CloudFormation 将返回卷的物理 ID。例如：`vol-5cb85026`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example DeletionPolicy 将在删除时拍摄快照的 EBS 卷

```
"NewVolume" : {
   "Type" : "AWS::EC2::Volume",
   "Properties" : {
      "Size" : "100",
     "AvailabilityZone" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone"
 ] },
      "Tags" : [ {
         "Key" : "MyTag",
         "Value" : "TagValue"
      } ]
   },
   "DeletionPolicy" : "Snapshot"
}
```

Example 具有 100 个预置 IOPS 的 EBS 卷。

```
"NewVolume" : {
   "Type" : "AWS::EC2::Volume",
   "Properties" : {
     "Size" : "100",
     "VolumeType" : "io1",
     "Iops" : "100",
     "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone"
 ] }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API Reference* 中的 CreateVolume
- DeletionPolicy 属性 (p. 485)

AWS::EC2::VolumeAttachment

Abstract

使用 AWS::EC2::VolumeAttachment 资源将 Amazon EBS 卷附加到正在运行的实例，然后将其公开给具有指定设备名称的实例。

将 Amazon EBS 卷附加到正在运行的实例，然后将其公开给具有指定设备名称的实例。

Important

在删除此资源（并随之断开卷）之前，您必须先卸载实例中的卷。如果没有执行此操作，当系统尝试卸载时，会导致卷卡在忙碌状态，而这可能会损害文件系统或其中包含的数据。

如果 Amazon EBS 卷是实例的根设备，则不可以在实例处于“正在运行”状态时断开。要断开根卷，请先停止实例。

如果根卷与带有 AWS Marketplace 产品代码的实例断开，那么该卷的 AWS Marketplace 产品代码就不再与该实例关联。

语法

```
{
   "Type":"AWS::EC2::VolumeAttachment",
   "Properties" : {
      "Device (p. 309)" : String,
      "InstanceId (p. 309)" : String,
      "VolumeId (p. 309)" : String
   }
}
```

属性

设备
: 设备如何对实例开放（如 /dev/sdh 或 xvdh）。

必需：是

类型：字符串

更新要求：不支持更新

实例 ID
: 与卷关联的实例的 ID。该值可以是对 AWS::EC2::Instance (p. 272) 资源的引用，也可以是现有 EC2 实例的物理 ID。

必需：是

类型：字符串

更新要求：不支持更新

卷 ID
: Amazon EBS 卷的 ID。卷和实例必须位于同一可用区内。该值可以是对 AWS::EC2::Volume (p. 305) 资源的引用，也可以是现有 Amazon EBS 卷的卷 ID。

必需：是

类型：字符串

更新要求：不支持更新

示例

本示例将 EC2 EBS 卷与逻辑名称为 "Ec2Instance" 的 EC2 实例关联。

```
"NewVolume" : {
   "Type" : "AWS::EC2::Volume",
   "Properties" : {
      "Size" : "100",
     "AvailabilityZone" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone"
 ] },
      "Tags" : [ {
         "Key" : "MyTag",
         "Value" : "TagValue"
      } ]
   }
},

"MountPoint" : {
   "Type" : "AWS::EC2::VolumeAttachment",
   "Properties" : {
      "InstanceId" : { "Ref" : "Ec2Instance" },
      "VolumeId"  : { "Ref" : "NewVolume" },
      "Device" : "/dev/sdh"
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud User Guide* 中的 Amazon Elastic Block Store (Amazon EBS)。
- *Amazon Elastic Compute Cloud User Guide* 中的 Attaching a Volume to an Instance
- *Amazon Elastic Compute Cloud User Guide* 中的 Detaching an Amazon EBS Volume from an Instance
- *Amazon Elastic Compute Cloud API Reference* 中的 AttachVolume
- *Amazon Elastic Compute Cloud API Reference* 中的 DetachVolume

AWS::EC2::VPC

Abstract

使用 AWS::EC2::VPC 资源创建具有指定 CIDR 块的 Virtual Private Cloud。

利用您指定的 CIDR 块创建 Virtual Private Cloud (VPC)。

语法

```
{
   "Type" : "AWS::EC2::VPC",
   "Properties" : {
      "CidrBlock (p. 310)" : String,
      "EnableDnsSupport (p. 310)" : Boolean,
      "EnableDnsHostnames (p. 310)" : Boolean,
      "InstanceTenancy (p. 311)" : String,
      "Tags (p. 311)" : [ EC2 Tag, ... ]
   }
}
```

属性

CidrBlock
: 您希望 VPC 覆盖的 CIDR 块。例如："10.0.0.0/16".
必需：是
类型：字符串
更新要求：替换 (p. 63)

EnableDnsSupport
: 指定是否支持 VPC 的 DNS 解析。如果此属性为 `true`，则 Amazon DNS 服务器将实例的 DNS 主机名解析为相应的 IP 地址；否则不进行解析。默认情况下，该值设置为 `true`。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

EnableDnsHostnames
: 指定在 VPC 中启动的实例是否获取 DNS 主机名。如果此属性为 `true`，则 VPC 中的实例获取 DNS 主机名；否则不获取主机名。只有在也将 `EnableDnsSupport` 属性设置为 `true` 的情况下，才可以将 `EnableDnsHostnames` 设置为 `true`。默认情况下，该值设置为 `false`。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

InstanceTenancy
允许租户启动的实例已发布到 VPC 中。

- `"default"`:实例可通过任何租户启动。
- `"dedicated"`:任何启动到 VPC 中的实例都会自动变为专用，而无论您在启动该实例时指定的租户选项是什么。

Required: No.
类型：字符串
有效值：`"default"` 或者`"dedicated"`
更新要求：替换 (p. 63)

标签
您要添加到此实例的标签。
类型：EC2 标签 (p. 453)的列表。
必需：否
更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myVPC" : {
         "Type" : "AWS::EC2::VPC",
         "Properties" : {
            "CidrBlock" : "10.0.0.0/16",
         "EnableDnsSupport" : "false",
         "EnableDnsHostnames" : "false",
            "InstanceTenancy" : "dedicated",
            "Tags" : [ {"Key" : "foo", "Value" : "bar"} ]
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateVpc。

AWS::EC2::VPCDHCPOptionsAssociation

Abstract

使用 AWS::EC2::VPCDHCPOptionsAssociation 资源将一组 DHCP 选项与指定 VPC 关联。

将一套 DHCP 选项（您先前创建的）与指定的 VPC 关联。

语法

```
{
   "Type" : "AWS::EC2::VPCDHCPOptionsAssociation",
   "Properties" : {
      "DhcpOptionsId (p. 312)" : String,
      "VpcId (p. 312)" : String
   }
}
```

属性

DhcpOptionsId
要与 VPC 关联的 DHCP 选项的 ID。如果您不希望 VPC 使用 DHCP 选项，请指定 `default`。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

VpcId
要关联此 DHCP 选项集的 VPC 的 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myVPCDHCPOptionsAssociation" : {
         "Type" : "AWS::EC2::VPCDHCPOptionsAssociation",
```

```
        "Properties" : {
            "VpcId" : {"Ref" : "myNetworkAcl"},
            "DhcpOptionsId" : {"Ref" : "myDhcpOption"},
        }
     }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 AssociateDhcpOptions。

AWS::EC2::VPCGatewayAttachment

Abstract

使用 AWS::EC2::VPCGatewayAttachment 资源将网关附加到 VPC。

将网关附加至 VPC。

语法

```
{
   "Type" : "AWS::EC2::VPCGatewayAttachment",
   "Properties" : {
      "InternetGatewayId (p. 313)" : String,
      "VpcId (p. 313)" : String,
      "VpnGatewayId (p. 313)" : String
   }
}
```

属性

InternetGatewayId
: Internet 网关的 ID。

  Required: Conditional. 您必须指定 InternetGatewayId 或 VpnGatewayId，但不能两者都指定。

  Type: String.

  更新要求：无中断 (p. 63)

VpcId
: 要关联此网关的 VPC 的 ID。

  Required: Yes.

  Type: String.

  更新要求：无中断 (p. 63)

VpnGatewayId
: 要连接到 VPC 的虚拟专用网络 (VPN) 网关的 ID。

  Required: Conditional. 您必须指定 InternetGatewayId 或 VpnGatewayId，但不能两者都指定。

Type: String.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example 将 Internet 网关和 VPN 网关连接到 VPC

若要将 Internet 网关和 VPN 网关都附加到 VPC，必须分开指定两个 AWS::EC2::VPCGatewayAttachment 资源：

```
"AttachGateway" : {
   "Type" : "AWS::EC2::VPCGatewayAttachment",
   "Properties" : {
      "VpcId" : { "Ref" : "VPC" },
      "InternetGatewayId" : { "Ref" : "myInternetGateway" }
   }
},

"AttachVpnGateway" : {
   "Type" : "AWS::EC2::VPCGatewayAttachment",
   "Properties" : {
      "VpcId" : { "Ref" : "VPC" },
      "VpnGatewayId" : { "Ref" : "myVPNGateway" }
   }
},
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 AttachVpnGateway。

AWS::EC2::VPNConnection

Abstract

使用 AWS::EC2::VPNConnection 资源在现有虚拟专用网关与 VPN 客户网关之间创建 VPN 连接。

在现有虚拟专用网关和 VPN 客户网关之间创建新的 VPN 连接。

有关更多信息，请参阅 *Amazon Elastic Compute Cloud API 参考* 中的 CreateVpnConnection。

语法

```
{
   "Type" : "AWS::EC2::VPNConnection",
   "Properties" : {
      "Type (p. 315)" : String,
      "CustomerGatewayId (p. 315)" : GatewayID,
      "StaticRoutesOnly (p. 315)" : Boolean,
      "   (p. 315)" :  [ Resource Tag, ... ],
      "VpnGatewayId (p. 315)" : GatewayID
   }
}
```

属性

类型
此虚拟私有网关支持的 VPN 连接类型。

示例："ipsec.1"

必需：是

类型：字符串

更新要求：替换 (p. 63)

CustomerGatewayId
客户网关的 ID。此 ID 既可能是嵌入式 JSON 对象也可能是网关 ID 的引用。

必需：是

类型：字符串
更新要求：替换 (p. 63)

StaticRoutesOnly
指明 VPN 连接是否要求使用静态路由。

必需：条件：如果要为不支持边界网关协议 (BGP) 的设备创建 VPN 连接，则必须指定 `true`。

类型：布尔值

更新要求：替换 (p. 63)

标签
需要附加到资源的标签。

Required: No.

类型：AWS CloudFormation 资源标签 (p. 471).

更新要求：无中断 (p. 63).

VpnGatewayId
虚拟专用网关的 ID。此 ID 既可能是嵌入式 JSON 对象也可能是网关 ID 的引用。

必需：是

类型：字符串
更新要求：替换 (p. 63)

返回值

当该资源的逻辑 ID 提供给 Ref 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyVPNConnection" }
```

对于具有逻辑 ID“MyVPNConnection”的 VPNConnection，Ref 将返回该 VPN 连接的资源名称。

有关使用 Ref 功能的更多信息，请参阅参考 (p. 508)。

模板示例

Example VPNConnection

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myVPNConnection" : {
         "Type" : "AWS::EC2::VPNConnection",
         "Properties" : {
            "Type" : "ipsec.1",
         "StaticRoutesOnly" : "true",
            "CustomerGatewayId" : {"Ref" : "myCustomerGateway"},
            "VpnGatewayId" : {"Ref" : "myVPNGateway"}
         }
      }
   }
}
```

AWS::EC2::VPNConnectionRoute

Abstract

使用 AWS::EC2::VPNConnectionRoute 资源在现有虚拟专用网关与 VPN 客户网关之间建立静态路由。

与现有虚拟专用网关和 VPN 客户网关之间的 VPN 连接相关联的静态路由。通过该静态路由，流量可以从虚拟专用网关路由至 VPN 客户网关。

语法

```
{
   "Type" : "AWS::EC2::VPNConnectionRoute",
   "Properties" : {
      "DestinationCidrBlock (p. 316)" : String
      "VpnConnectionId (p. 317)" : String,
   }
}
```

属性

DestinationCidrBlock
    与客户网络的本地子网关联的 CIDR 块。

必需：可以。

Type: String.

更新要求：替换 (p. 63)

VpnConnectionId
VPN 连接的 ID。

必需：可以。

Type: String.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example 指定静态路由

```
"MyConnectionRoute0" : {
   "Type" : "AWS::EC2::VPNConnectionRoute",
   "Properties" : {
      "DestinationCidrBlock" : "10.0.0.0/16",
      "VpnConnectionId" : {"Ref" : "Connection0"}
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateVpnConnectionRoute。

AWS::EC2::VPNGateway

Abstract

使用 AWS::EC2::VPNGateway 资源创建虚拟专用网关。

创建虚拟专用网关。虚拟专用网关是 VPN 连接的 VPC 侧终端节点。

语法

```
{
   "Type" : "AWS::EC2::VPNGateway",
   "Properties" : {
      "Type (p. 318)" : String,
```

```
      "Tags (p. 318)" : [ Tag1, ... ]
   }
}
```

属性

类型
: 此虚拟私有网关支持的 VPN 连接类型。唯一有效的值是 "ipsec.1"。

必需：是

类型：字符串

更新要求：替换 (p. 63)

标签
: 您要添加到此实例的标签。

必需：否

类型：EC2 标签 (p. 453)的列表。
更新要求：无中断 (p. 63)

返回值

当该资源的逻辑 ID 提供给 Ref 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyVPNGateway" }
```

对于具有逻辑 ID“MyVPNGateway”的 VPN 网关，Ref 将返回该网关的资源名称。

有关使用 Ref 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myVPNGateway" : {
         "Type" : "AWS::EC2::VPNGateway",
         "Properties" : {
            "Type" : "ipsec.1",
            "Tags" : [ { "Key" : "Use", "Value" : "Test" } ]
         }
      }
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 CreateVpnGateway。

AWS::EC2::VPNGatewayRoutePropagation

Abstract

使用 AWS::EC2::VPNGatewayRoutePropagation 资源类型启用虚拟专用网关以将路由传播至 VPC 的路由表。

启用虚拟专用网关 (VGW) 以将路由传播至 VPC 的路由表。

语法

```
{
   "Type" : "AWS::EC2::VPNGatewayRoutePropagation",
   "Properties" : {
      "RouteTableIds (p. 319)" : [ String, ... ],
      "VpnGatewayId (p. 319)" : String
   }
}
```

属性

RouteTableIds
与 VPC 关联的路由表 ID 的列表。路由表必须与虚拟专用网关所附加到的 VPC 关联。

Required: Yes.

类型：路由表 ID 列表

更新要求：无中断 (p. 63)

VpnGatewayId
附加到 VPC 的虚拟专用网关的 ID。虚拟专用网关必须附加到路由表所关联的 VPC。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myVPNGatewayRouteProp" }
```

对于逻辑 ID 为 *myVPNGatewayRouteProp* 的 VPN 网关，`Ref` 将返回该网关的资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
"myVPNGatewayRouteProp" : {
  "Type" : "AWS::EC2::VPNGatewayRoutePropagation",
```

```
  "Properties" : {
    "RouteTableIds" : [{"Ref" : "PrivateRouteTable"}],
    "VpnGatewayId" : {"Ref" : "VPNGateway"}
  }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API 参考* 中的 EnableVgwRoutePropagation。

AWS::ElastiCache::CacheCluster

Abstract

使用 AWS::ElastiCache::CacheCluster 资源创建 Amazon ElastiCache 缓存集群。

类型 AWS::ElastiCache::CacheCluster 可用于创建 Amazon ElastiCache 缓存集群。

语法

```
{
   "Type" : "AWS::ElastiCache::CacheCluster",
   "Properties" :
   {
      "AutoMinorVersionUpgrade (p. 320)" : Boolean,
      "CacheNodeType (p. 321)" : String,
      "CacheParameterGroupName (p. 321)" : String,
      "CacheSecurityGroupNames (p. 321)" : [ String, ... ],
      "CacheSubnetGroupName (p. 321)" : String,
      "ClusterName (p. 321)" : String,
      "Engine (p. 321)" : String,
      "EngineVersion (p. 322)" : String,
      "NotificationTopicArn (p. 322)" : String,
      "NumCacheNodes (p. 322)" : String,
      "Port (p. 322)" : Integer,
      "PreferredAvailabilityZone (p. 322)" : String,
      "PreferredMaintenanceWindow (p. 322)" : String,
      "SnapshotArns (p. 322)" : [String, ... ],
      "VpcSecurityGroupIds (p. 323)" : [String, ...]
   }
}
```

属性

AutoMinorVersionUpgrade
表示在维护窗口期间，将对该缓存集群自动应用次要引擎升级。

必需：否

类型：布尔值

默认值：`true`

更新要求：无中断 (p. 63)

CacheNodeType
缓存集群中的节点的计算和内存容量。

必需：是

类型：字符串

更新要求：替换 (p. 63)

CacheParameterGroupName
此缓存集群相关缓存参数组的名称。

必需：否

类型：字符串

更新要求：时而中断 (p. 63)

CacheSecurityGroupNames
与此缓存集群相关的缓存安全组名称列表。如果您的缓存集群在 VPC 中，请改为指定 `VpcSecurityGroupIds` 属性。

必需：如果您的缓存集群不在 VPC 中，则必须指定此属性。

类型：字符串列表

更新要求：无中断 (p. 63)

CacheSubnetGroupName
您与缓存集群关联的缓存子网组。

必需：否

类型：字符串

更新要求：替换 (p. 63)

ClusterName
缓存集群的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用于缓存集群。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

必需：否

类型：字符串

更新要求：替换 (p. 63)

引擎
要用于此缓存集群的缓存引擎的名称，如 `memcached` 或 `redis`。

Note

AWS CloudFormation 目前不支持 Redis 的复制组和只读副本。

必需：是

类型：字符串

更新要求：替换 (p. 63)

引擎版本
要用于此集群的缓存引擎的版本。

必需：否

类型：字符串

更新要求：时而中断 (p. 63)

NotificationTopicArn
向其发送通知的目标 Amazon Simple Notification Service (SNS) 主题的 Amazon 资源名称 (ARN)。

必需：否

类型：字符串

更新要求：无中断 (p. 63)

NumCacheNodes
缓存集群应该拥有的缓存节点的数量。

必需：是

类型：字符串

更新要求：无中断 (p. 63)

端口
各缓存节点接受连接的端口编号。

必需：否

类型：整数

更新要求：替换 (p. 63)

PreferredAvailabilityZone
用于创建缓存集群的 EC2 可用区。

必需：否

类型：字符串

更新要求：替换 (p. 63)

PreferredMaintenanceWindow
可能进行系统维护的每周时间范围（按世界协调时间计算）。

必需：否

类型：字符串

更新要求：无中断 (p. 63)

SnapshotArns
要用来为新 Redis 缓存集群做种的快照文件的 ARN。如果您在 Amazon ElastiCache 之外管理 Redis 实例，则可使用存储在 Amazon S3 存储桶中的快照文件在 Amazon ElastiCache 中创建新缓存集群。

必需：否

类型：字符串列表

更新要求：替换 (p. 63)

VpcSecurityGroupIds
VPC 的安全组 ID 列表。如果您的缓存集群不在 VPC 中，请指定 `CacheSecurityGroupNames` 属性。

Note

您必须使用 `AWS::EC2::SecurityGroup` 资源而不是 `AWS::ElastiCache::SecurityGroup` 资源来指定处于 VPC 中的 Amazon ElastiCache 安全组。此外，如果您将默认 VPC 用于您的 AWS 账户，则必须使用 `Fn::GetAtt` 函数和 `GroupId` 属性检索安全组 ID（而不是 `Ref` 函数）。要查看示例模板，请参阅“模板代码段”部分。

必需：如果您的缓存集群在 VPC 中，则必须指定此属性。

类型：字符串列表

更新要求：无中断 (p. 63)

备注

获取缓存集群节点地址

Amazon ElastiCache 缓存集群没有单一终端节点，但是，通过定义 `get-cache-nodes` 脚本并将其安装在模板的 AWS::CloudFormation::Init (p. 241) 部分，可以获得各缓存节点的终端节点。

您可以查看完整示例模板了解实现详细信息：

- 有关 Memcached 的信息，请参阅 https://s3.amazonaws.com/cloudformation-templates-us-east-1/ElastiCache.template
- 有关 Redis 的信息，请参阅 https://s3.amazonaws.com/cloudformation-templates-us-east-1/ElastiCache_Redis.template

Amazon ElastiCache 模板使用 AWS CloudFormation 引导脚本 cfn-hup (p. 520) 来检测 Amazon ElastiCache 缓存集群配置发生的更改，例如缓存集群中实例的数量。然后，运行脚本来更新应用程序在主机上的配置。

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Note

目前，只能将 `Fn::GetAtt` 用于 Memcached 缓存集群。

ConfigurationEndpoint.Address
Memcached 缓存集群的配置终端节点的 DNS 地址。

ConfigurationEndpoint.Port
Memcached 缓存集群的配置终端节点的端口号。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板代码段

以下代码段描述处于默认 VPC 中的安全组中的一个 Amazon ElastiCache 集群。通常，VPC 中的安全组要求指定 VPC ID。在此例中无需 VPC ID，因为安全组使用默认 VPC。

对于缓存集群，`VpcSecurityGroupIds` 属性用于将集群与安全组关联。因为 `VpcSecurityGroupIds` 属性需要安全组 ID（而不是安全组名称），所以模板代码段对 `ElasticacheSecurityGroup` 资源使用 `Fn::GetAtt` 函数而不是 `Ref` 函数。因为安全组未指定 VPC ID，所以 `Ref` 函数将返回安全组名称。

```
"ElasticacheSecurityGroup": {
  "Type": "AWS::EC2::SecurityGroup",
  "Properties": {
    "GroupDescription": "Elasticache Security Group",
    "SecurityGroupIngress": [ {
      "IpProtocol": "tcp",
      "FromPort": "11211",
      "ToPort": "11211",
      "SourceSecurityGroupName": {"Ref": "InstanceSecurityGroup"}
    } ]
  }
},
"ElasticacheCluster": {
  "Type": "AWS::ElastiCache::CacheCluster",
  "Properties": {
    "AutoMinorVersionUpgrade": "true",
    "Engine": "memcached",
    "CacheNodeType": "cache.t1.micro",
    "NumCacheNodes": "1",
    "VpcSecurityGroupIds": [{"Fn::GetAtt": [ "ElasticacheSecurityGroup",
"GroupId"]}]
  }
}
```

另请参阅

- *Amazon ElastiCache API Reference Guide* 中的 CreateCacheCluster
- *Amazon ElastiCache API Reference Guide* 中的 ModifyCacheCluster

AWS::ElastiCache::ParameterGroup

Abstract

使用 AWS::ElastiCache::ParameterGroup 资源创建缓存参数组。

类型 AWS::ElastiCache::ParameterGroup 可用于创建新的缓存参数组。缓存参数组控制缓存集群的参数。

语法

```
{
   "Type": "AWS::ElastiCache::ParameterGroup",
   "Properties": {
      "CacheParameterGroupFamily" : String,
      "Description" : String,
      "Properties" : { "prop1" : "value1", ... }
   }
}
```

属性

CacheParameterGroupFamily
缓存参数组可以使用的缓存参数组系列的名称。

Required: Yes.

Type: String.

更新要求：不支持更新

说明
缓存参数组的相关说明。

Required: Yes.

Type: String.

更新要求：不支持更新

属性
参数名称/值对的列表（以逗号隔开）。有关更多信息，请转至 *Amazon ElastiCache API Reference Guide* 中的 ModifyCacheParameterGroup。

示例：

```
"Properties" : {
   "cas_disabled" : "1",
   "chunk_size_growth_factor" : "1.02"
}
```

Required: Yes.

Type: JSON object.

更新要求：不支持更新

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
"MyParameterGroup": {
   "Type": "AWS::ElastiCache::ParameterGroup",
   "Properties": {
      "Description": "MyNewParameterGroup",
      "CacheParameterGroupFamily": "memcached1.4",
      "Properties" : {
         "cas_disabled" : "1",
         "chunk_size_growth_factor" : "1.02"
      }
   }
}
```

另请参阅

- *Amazon ElastiCache API Reference Guide* 中的 CreateCacheParameterGroup
- *Amazon ElastiCache API Reference Guide* 中的 ModifyCacheParameterGroup
- AWS CloudFormation 堆栈更新 (p. 63)

AWS::ElastiCache::SecurityGroup

Abstract

使用 `AWS::ElastiCache::SecurityGroup` 资源创建缓存安全组。

`AWS::ElastiCache::SecurityGroup` 资源会创建缓存安全组。有关缓存安全组的更多信息，请参阅 *Amazon ElastiCache User Guide* 中的 Cache Security Groups，或 *Amazon ElastiCache API Reference Guide* 中的 CreateCacheSecurityGroup。

要在 VPC 中创建 Amazon ElastiCache 集群，请使用 AWS::EC2::SecurityGroup (p. 292) 资源。有关更多信息，请参阅 AWS::ElastiCache::CacheCluster (p. 320) 资源中的 `VpcSecurityGroupIds` 属性。

语法

```
{
  "Type" : "AWS::ElastiCache::SecurityGroup",
  "Properties" :
  {
    "    (p. 326)" : String
  }
}
```

属性

说明
: 缓存安全组的描述。

类型：字符串

Required: No.

更新要求：不支持更新

返回值

Ref

当您将 `AWS::ElastiCache::SecurityGroup` 资源指定为 `Ref` 函数的参数时，AWS CloudFormation 会返回缓存安全组的 *`CacheSecurityGroupName`* 属性。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

AWS::ElastiCache::SecurityGroupIngress

Abstract

使用 AWS::ElastiCache::SecurityGroupIngress 资源授予从指定 Amazon EC2 安全组中的主机传入缓存安全组的权限。

类型 AWS::ElastiCache::SecurityGroupIngress 可授予从指定 Amazon EC2 安全组中的主机传入缓存安全组的权限。有关 ElastiCache 安全组传入权限的更多信息，请参阅 *Amazon ElastiCache API Reference Guide* 中的 AuthorizeCacheSecurityGroupIngress。

语法

```
{
  "Type" : "AWS::ElastiCache::SecurityGroupIngress",
  "Properties" :
  {
    "CacheSecurityGroupName (p. 327)" : String,
    "EC2SecurityGroupName (p. 327)" : String,
    "EC2SecurityGroupOwnerId (p. 327)" : String
  }
}
```

属性

CacheSecurityGroupName
要授权的缓存安全组的名称。

Type: String.

Required: Yes.

更新要求：不支持更新

EC2SecurityGroupName
要在授权中包含的 EC2 安全组的名称。

Type: String.

Required: Yes.

更新要求：不支持更新

EC2SecurityGroupOwnerId
指定 EC2 安全组（EC2SecurityGroupName 属性指定的）的所有人的 AWS 账户 ID。AWS 访问密钥 ID 不是认可的值。

Type: String.

Required: No.

更新要求：不支持更新

AWS::ElastiCache::SubnetGroup

Abstract

使用 AWS::ElastiCache::SubnetGroup 资源创建缓存子网组。

创建缓存子网组。有关缓存子网组的更多信息，请参阅 *Amazon ElastiCache User Guide* 中的 Cache Subnet Groups，或转到 *Amazon ElastiCache API Reference Guide* 中的 CreateCacheSubnetGroup。

在将 AWS::ElastiCache::SubnetGroup 类型指定为 `Ref` 函数的参数时，AWS CloudFormation 返回缓存子网组的名称。

语法

```
"SubnetGroup" : {
    "Type" : "AWS::ElastiCache::SubnetGroup",
    "Properties" : {
        "Description (p. 328)" : String,
        "SubnetIds (p. 328)" : [ String, ... ]
    }
}
```

属性

说明
缓存子网组的说明。

类型：字符串

必需：是

更新要求：无中断 (p. 63)

SubnetId
缓存子网组的 Amazon EC2 子网 ID。

类型：字符串列表

必需：是

更新要求：无中断 (p. 63)

示例

```
"SubnetGroup" : {
    "Type" : "AWS::ElastiCache::SubnetGroup",
    "Properties" : {
        "Description" : "Cache Subnet Group",
        "SubnetIds" : [ { "Ref" : "Subnet1" }, { "Ref" : "Subnet2" } ]
    }
}
```

AWS::ElasticBeanstalk::Application

Abstract

使用 AWS::ElasticBeanstalk::Application 资源创建 AWS Elastic Beanstalk 应用程序。

创建 AWS Elastic Beanstalk 应用程序。

语法

```
{
   "Type" : "AWS::ElasticBeanstalk::Application",
   "Properties" : {
      "ApplicationName (p. 329)" : String,
      "Description (p. 329)" : String
   }
}
```

属性

ApplicationName
AWS Elastic Beanstalk 应用程序的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作应用程序名称。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

Required: No.

Type: String.

更新要求：替换 (p. 63)

说明
对此应用程序的可选描述。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
{
   "Type" : "AWS::ElasticBeanstalk::Application",
   "Properties" : {
      "ApplicationName" : "SampleAWSElasticBeanstalkApplication",
      "Description" : "AWS Elastic Beanstalk PHP Sample Application"
   }
}
```

另请参阅

- 有关完整 AWS Elastic Beanstalk 示例模板，请参阅 AWS Elastic Beanstalk 代码段 (p. 144)。

AWS::ElasticBeanstalk::ApplicationVersion

Abstract

为 AWS Elastic Beanstalk 应用程序创建应用程序版本。

为 AWS Elastic Beanstalk 应用程序创建应用程序版本（可部署代码的迭代）。

语法

```
{
  "Type" : "AWS::ElasticBeanstalk::ApplicationVersion",
  "Properties" : {
    "ApplicationName (p. 330)" : String,
    "   (p. 330)" : String,
    "SourceBundle (p. 330)" : { SourceBundle }
  }
}
```

成员

ApplicationName
与此应用程序版本关联的 AWS Elastic Beanstalk 应用程序的名称。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

说明
此应用程序版本的描述。

Required: No.

Type: String.

更新要求：时而中断 (p. 63)

SourceBundle
此版本的源捆绑的位置。

Required: No.

Type: 源包 (p. 456)

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

```
"myAppVersion" :{
  "Type" : "AWS::ElasticBeanstalk::ApplicationVersion",
  "Properties" : {
    "ApplicationName" : {"Ref" : "myApp"},
    "Description" : "my sample version",
    "SourceBundle" : {
      "S3Bucket" : { "Fn::Join" :
        ["-", [ "elasticbeanstalk-samples", { "Ref" : "AWS::Region" } ] ] },
      "S3Key" : "php-sample.zip"
    }
  }
}
```

另请参阅

- 有关完整 AWS Elastic Beanstalk 示例模板，请参阅 AWS Elastic Beanstalk 代码段 (p. 144)。

AWS::ElasticBeanstalk::ConfigurationTemplate

Abstract

为 AWS Elastic Beanstalk 应用程序创建配置模板。

为 AWS Elastic Beanstalk 应用程序创建配置模板。您可以通过在配置模板中定义的配置设置，使用配置模板部署不同版本的应用程序。

语法

```
{
  "Type" : "AWS::ElasticBeanstalk::ConfigurationTemplate",
  "Properties" : {
    "ApplicationName (p. 332)" : String,
    "   (p. 332)" : String,
    "EnvironmentId (p. 332)" : String,
    "OptionSettings (p. 332)" : [ OptionSetting, ... ],
    "SolutionStackName (p. 332)" : String,
    "SourceConfiguration (p. 332)" : Source configuration
```

```
  }
}
```

成员

ApplicationName
与此配置模板关联的 AWS Elastic Beanstalk 应用程序的名称。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

说明
对此配置的可选描述。

Type: String.

Required: No.

更新要求：时而中断 (p. 63)

EnvironmentId
要使用其设置创建配置模板的环境。如果不指定 `SolutionStackName` 或 `SourceConfiguration` 属性，则必须指定此属性。

Type: String.

Required: Conditional.

更新要求：替换 (p. 63)

OptionSettings
此 Elastic Beanstalk 配置的 OptionSettings (p. 455) 列表。有关 Elastic Beanstalk 配置选项的完整列表，请参阅 *AWS Elastic Beanstalk Developer Guide* 中的 Option Values。

Type: OptionSettings (p. 455) 列表。

Required: No.

更新要求：时而中断 (p. 63)

SolutionStackName
此配置将使用的 AWS Elastic Beanstalk 解决方案堆栈的名称。解决方案堆栈可指定配置模板的操作系统、架构和应用程序服务器，如 `64bit Amazon Linux 2013.09 running Tomcat 7 Java 7`。有关详细信息，请参阅 *AWS Elastic Beanstalk 开发人员指南* 中的支持的平台。

如果不指定 `EnvironmentId` 或 `SourceConfiguration` 属性，则必须指定此属性。

Type: String.

Required: Conditional.

更新要求：替换 (p. 63)

SourceConfiguration
与其他 AWS Elastic Beanstalk 应用程序关联的配置模板。如果您指定 `SolutionStackName` 属性和 `SourceConfiguration` 属性，则源配置模板中的解决方案堆栈必须与您为 `SolutionStackName` 属性指定的值匹配。

如果不指定 `EnvironmentId` 或 `SolutionStackName` 属性，则必须指定此属性。

类型：AWS Elastic Beanstalk SourceConfiguration 属性类型 (p. 457)

Required: Conditional.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

此 ElasticBeanstalk ConfigurationTemplate 示例在 AWS CloudFormation 示例模板 ElasticBeanstalkSample.template 中，该模板还提供了它在 AWS::ElasticBeanstalk::Application 中的使用示例。

```
"myConfigTemplate" : {
  "Type" : "AWS::ElasticBeanstalk::ConfigurationTemplate",
  "Properties" : {
    "ApplicationName" :{"Ref" : "myApp"},
    "Description" : "my sample configuration template",
    "EnvironmentId" : "",
    "SourceConfiguration" : {
      "ApplicationName" : {"Ref" : "mySecondApp"},
      "TemplateName" : {"Ref" : "mySourceTemplate"}
    },
    "SolutionStackName" : "64bit Amazon Linux running PHP 5.3",
    "OptionSettings" : [ {
      "Namespace" : "aws:autoscaling:launchconfiguration",
      "OptionName" : "EC2KeyName",
      "Value" : { "Ref" : "KeyName" }
    } ]
  }
}
```

另请参阅

- AWS::ElasticBeanstalk::Application (p. 329)
- *AWS Elastic Beanstalk Developer Guide* 中的 Option Values
- 有关完整 AWS Elastic Beanstalk 示例模板，请参阅 AWS Elastic Beanstalk 代码段 (p. 144)。

AWS::ElasticBeanstalk::Environment

Abstract

使用 AWS::ElasticBeanstalk::Environment 资源创建或更新 AWS Elastic Beanstalk 环境。

创建或更新 AWS Elastic Beanstalk 环境。

语法

```
{
   "Type" : "AWS::ElasticBeanstalk::Environment",
   "Properties" : {
      "ApplicationName (p. 334)" : String,
      "CNAMEPrefix (p. 334)" : String,
      "Description (p. 334)" :  String,
      "EnvironmentName (p. 334)" :  String,
      "OptionSettings (p. 335)" : [ OptionSettings, ... ],
      "SolutionStackName (p. 335)" : String,
      "TemplateName (p. 335)" : String,
      "   (p. 335)" : Environment Tier,
      "VersionLabel (p. 335)" : String
   }
}
```

属性

ApplicationName
与此环境关联的应用程序的名称。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

CNAMEPrefix
AWS Elastic Beanstalk 环境 URL 的前缀。

Required: No.

Type: String.

更新要求：替换 (p. 63)

说明
帮助您标识此环境的描述。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

EnvironmentName
AWS Elastic Beanstalk 环境的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作环境名称。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

Required: No.

Type: String.

更新要求：替换 (p. 63)

OptionSettings
为此环境定义配置选项的键/值对。这些选项会覆盖在解决方案堆栈或配置模板中定义的值。如果在堆栈更新过程中删除任何选项，则删除的选项会恢复为默认值。

Required: No.

Type: OptionSettings (p. 455) 列表。

更新要求：时而中断 (p. 63)

SolutionStackName
此配置将使用的 AWS Elastic Beanstalk 解决方案堆栈的名称。有关详细信息，请参阅 *AWS Elastic Beanstalk 开发人员指南* 中的支持的平台。您必须指定此参数或 AWS Elastic Beanstalk 配置模板名称。

Required: No.

Type: String.

更新要求：替换 (p. 63)

TemplateName
要与环境配合使用的 AWS Elastic Beanstalk 配置模板的名称。您必须指定此参数或解决方案堆栈名称。

Required: No.

Type: String.

更新要求：时而中断 (p. 63)

套餐
指定要用于创建环境的层。选择的环境层确定 AWS Elastic Beanstalk 配置资源是支持处理 HTTP(S) 请求的 Web 应用程序还是处理后台处理任务的 Web 应用程序。

Required: No.

类型：AWS Elastic Beanstalk 环境层属性类型 (p. 454)

更新要求：请参阅 AWS Elastic Beanstalk 环境层属性类型 (p. 454)

VersionLabel
要与环境关联的版本。

Required: No.

Type: String.

更新要求：时而中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

EndpointURL
: 此环境的负载均衡器的 URL。

  示例：

  awseb-myst-myen-132MQC4KRLAMD-1371280482.us-east-1.elb.amazonaws.com

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

简单环境

```
{
   "Type" : "AWS::ElasticBeanstalk::Environment",
   "Properties" : {
      "ApplicationName" : { "Ref" : "sampleApplication" },
      "Description" :  "AWS Elastic Beanstalk Environment running PHP Sample
Application",
      "EnvironmentName" :  "SamplePHPEnvironment",
      "TemplateName" : "DefaultConfiguration",
      "VersionLabel" : "Initial Version"
   }
}
```

具有嵌入式选项设置的环境

```
{
   "Type" : "AWS::ElasticBeanstalk::Environment",
   "Properties" : {
      "ApplicationName" : { "Ref" : "sampleApplication" },
     "Description" :  "AWS Elastic Beanstalk Environment running Python Sample
 Application",
      "EnvironmentName" :  "SamplePythonEnvironment",
      "SolutionStackName" : "64bit Amazon Linux running Python",
      "OptionSettings" : [ {
         "Namespace" : "aws:autoscaling:launchconfiguration",
         "OptionName" : "EC2KeyName",
         "Value" : { "Ref" : "KeyName" }
      } ],
      "VersionLabel" : "Initial Version"
   }
}
```

另请参阅

- *AWS Elastic Beanstalk Developer Guide* 中的 Launching New Environments
- *AWS Elastic Beanstalk Developer Guide* 中的 Managing Environments

- 有关完整 AWS Elastic Beanstalk 示例模板，请参阅 AWS Elastic Beanstalk 代码段 (p. 144)。

AWS::ElasticLoadBalancing::LoadBalancer

Abstract

使用 AWS::ElasticLoadBalancing::LoadBalancer 资源创建负载均衡器。

AWS::ElasticLoadBalancing::LoadBalancer 类型可创建 LoadBalancer。

Note

如果此资源具有公有 IP 地址并且还处于同一模板中定义的 VPC 内，则您必须使用 DependsOn 属性声明对 VPC 网关连接的依赖关系。有关更多信息，请参阅 DependsOn 属性 (p. 486)。

语法

```
{
   "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
   "Properties": {
      "AccessLoggingPolicy (p. 337)" : AccessLoggingPolicy,
      "AppCookieStickinessPolicy (p. 337)" : [ AppCookieStickinessPolicy, ...
],
      "AvailabilityZones (p. 338)" : [ String, ... ],
      "ConnectionDrainingPolicy (p. 338)" : ConnectionDrainingPolicy,
      "CrossZone (p. 338)" : Boolean,
      "HealthCheck (p. 338)" : HealthCheck,
      "Instances (p. 338)" : [ String, ... ],
      "LBCookieStickinessPolicy (p. 338)" : [ LBCookieStickinessPolicy, ... ],
      "        (p. 338)" : String,
      "Listeners (p. 339)" : [ Listener, ... ],
      "Policies (p. 339)" : [ ElasticLoadBalancing Policy, ... ],
      "Scheme (p. 339)" : String,
      "SecurityGroups (p. 339)" : [ Security Group, ... ],
      "Subnets (p. 339)" : [ String, ... ]
   }
}
```

属性

AccessLoggingPolicy
获取对负载均衡器进行的所有请求的相关详细信息，如收到请求的时间、客户端的 IP 地址、延迟、请求路径和服务器响应。

Required: No.

Type: Elastic Load Balancing AccessLoggingPolicy (p. 458)

更新要求：无中断 (p. 63)

AppCookieStickinessPolicy
可生成一个或多个粘性策略，其粘性会话生命周期取决于应用程序生成的 cookie 的生命周期。这些策略只能与 HTTP/HTTPS 侦听器关联。

Required: No.

Type: AppCookieStickinessPolicy (p. 458) 对象列表。

更新要求：无中断 (p. 63)

AvailabilityZones
从中创建负载均衡器的可用区。您可以指定“AvailabilityZones”或“Subnets”，但不能同时指定两者。

Required: No.

Type: A list of strings.

更新要求：替换 (p. 63) - 在未指定可用区并且要添加一个时，或在要删除所有可用区时。否则，更新要求无中断 (p. 63)。

ConnectionDrainingPolicy
已取消注册或运行状况不佳的实例是否可以完成所有处于飞行状态的请求。

Required: No.

Type: Elastic Load Balancing ConnectionDrainingPolicy (p. 459)

更新要求：无中断 (p. 63)

CrossZone
是否为负载均衡器启用跨区域负载均衡。凭借跨区域负载均衡，负载均衡器节点将流量路由到跨所有可用区的后端实例。默认情况下，`CrossZone` 属性是 `false`。

Required: No.

类型：布尔值

更新要求：无中断 (p. 63)

HealthCheck
实例的应用程序运行状况检查。

Required: No.

类型：ElasticLoadBalancing HealthCheck 类型 (p. 460).

更新要求：替换 (p. 63) - 在未指定运行状况检查并且要添加一个时，或在要删除运行状况检查时。否则，更新要求无中断 (p. 63)。

实例
负载均衡器的 EC2 实例 ID 列表。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

LBCookieStickinessPolicy
生成一个粘性策略，其粘性会话生命周期由浏览器 (user-agent) 的生命周期控制，或者在指定期限后到期。此策略只能与 HTTP/HTTPS 侦听器关联。

Required: No.

Type: LBCookieStickinessPolicy (p. 461) 对象列表。

更新要求：无中断 (p. 63)

负载均衡器名称
负载均衡器的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用于负载均衡器。名称必须在负载均衡器组中是唯一的。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

必需：否

类型：字符串

更新要求：替换 (p. 63)

侦听器

适用于此负载均衡器的一个或多个侦听器。每个侦听器都必须注册一个特定端口，一个给定端口不能具有多个侦听器。

Important

如果您更新了 `Listeners` 属性指定的侦听器的属性值，AWS CloudFormation 将删除现有侦听器并使用更新后的属性创建一个新的侦听器。AWS CloudFormation 执行此操作期间，客户端将无法连接至负载均衡器。

Required: Yes.

Type:ElasticLoadBalancing Listener 属性类型 (p. 462) 对象列表。

更新要求：无中断 (p. 63)

策略

要应用至此弹性负载均衡器的 Elastic Load Balancing 策略列表。

Required: No.

Type: ElasticLoadBalancing 策略 (p. 463)对象列表。

更新要求：无中断 (p. 63)

Scheme

对于连接至 Amazon VPC 的负载均衡器，此参数可用于指定要使用的负载均衡器类型。指定 **`“internal”`**将创建一个 DNS 名称可解析为私有 IP 地址的内部负载均衡器。

Required: No.

Type: String.

更新要求：替换 (p. 63)

SecurityGroups

Required: No.

Type: Virtual Private Cloud (VPC) 中分配至您的负载均衡器的安全组的列表。

更新要求：无中断 (p. 63)

子网

Virtual Private Cloud (VPC) 中将与您的负载均衡器关联的子网 ID 列表。您可以指定“AvailabilityZones”*或*“Subnets”，但不能同时指定两者。

有关在 VPC 中使用 Elastic Load Balancing 的更多信息，请参阅 *Elastic Load Balancing Developer Guide* 中的 How Do I Use Elastic Load Balancing in Amazon VPC。

Required: No.

Type: A list of strings.

更新要求：替换 (p. 63) - 在未指定子网并且要添加一个时，或在要删除所有子网时。否则，更新要求无中断 (p. 63)。

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如，`mystack-myelb-1WQN7BJGDB5YQ`

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

CanonicalHostedZoneName
与负载均衡器关联的 Route 53 托管区域的名称。

Important

如果您指定 `internal` 作为 Elastic Load Balancing 模式，请改用 `DNSName`。对于 `internal` 模式，负载均衡器没有 `CanonicalHostedZoneName` 值。

示例：`mystack-myelb-15HMABG9ZCN57-1013119603.us-east-1.elb.amazonaws.com`

CanonicalHostedZoneNameID
与负载均衡器关联的 Route 53 托管区域名称的 ID。

示例：`Z3DZXE0Q79N41H`

DNSName
负载均衡器的 DNS 名称。

示例：`mystack-myelb-15HMABG9ZCN57-1013119603.us-east-1.elb.amazonaws.com`

SourceSecurityGroup.GroupName
您可以作为负载均衡器后端 Amazon EC2 应用程序实例入站规则的一部分使用的安全组。

示例：`amazon-elb`

SourceSecurityGroup.OwnerAlias
源安全组的所有者。

例如：`amazon-elb-sg`

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

具有运行状况检查和访问日志的负载均衡器

```
"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "Instances" : [ { "Ref" : "Ec2Instance1" },{ "Ref" : "Ec2Instance2" } ],
    "Listeners" : [ {
```

```
      "LoadBalancerPort" : "80",
      "InstancePort" : { "Ref" : "WebServerPort" },
      "Protocol" : "HTTP"
    } ],
    "HealthCheck" : {
      "Target" : {
        "Fn::Join" : [ "", [ "HTTP:", { "Ref" : "WebServerPort" }, "/" ] ]
      },
      "HealthyThreshold" : "3",
      "UnhealthyThreshold" : "5",
      "Interval" : "30",
      "Timeout" : "5"
    },
    "AccessLoggingPolicy": {
      "S3BucketName": {
        "Ref": "S3LoggingBucket"
      },
      "S3BucketPrefix": "MyELBLogs",
      "Enabled": "true",
      "EmitInterval" : "60"
    },
  "DependsOn": "S3LoggingBucketPolicy"
  }
}
```

启用了访问日志记录的负载均衡器

以下示例代码段创建具有一个存储桶策略的 Amazon S3存储桶，该策略允许负载均衡器在 Logs/AWSLogs/*AWS account number*/ 文件夹中存储信息。负载均衡器还包括对存储桶策略的显式依赖关系，需要先建立这种依赖关系，然后负载均衡器才能写入存储桶。

```
"S3LoggingBucket": {
  "Type": "AWS::S3::Bucket"
},
"S3LoggingBucketPolicy": {
  "Type": "AWS::S3::BucketPolicy",
  "Properties": {
    "Bucket": {
      "Ref": "S3LoggingBucket"
    },
    "PolicyDocument": {
      "Version": "2008-10-17",
      "Statement": [ {
        "Sid": "ELBAccessLogs20130930",
        "Effect": "Allow",
        "Resource": {
          "Fn::Join": [
            "",
            [
              "arn:aws:s3:::",
              { "Ref": "S3LoggingBucket" },
              "/",
              "Logs",
              "/AWSLogs/",
              { "Ref": "AWS::AccountId" },
              "/*"
```

```
            ]
          ]
        },
        "Principal": { "AWS": "*" },
        "Action": [
          "s3:PutObject"
        ]
      } ]
    }
  }
},
"ElasticLoadBalancer": {
  "Type": "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties": {
    "AvailabilityZones": { "Fn::GetAZs": "" },
    "Listeners": [{
      "LoadBalancerPort": "80",
      "InstancePort": "80",
      "Protocol": "HTTP"
    }],
    "HealthCheck": {
      "Target": "HTTP:80/",
      "HealthyThreshold": "3",
      "UnhealthyThreshold": "5",
      "Interval": "30",
      "Timeout": "5"
    },
    "AccessLoggingPolicy": {
      "S3BucketName": {
        "Ref": "S3LoggingBucket"
      },
      "S3BucketPrefix": "Logs",
      "Enabled": "true",
      "EmitInterval" : "60"
    }
  },
  "DependsOn": "S3LoggingBucketPolicy"
}
```

具有连接耗尽策略的负载均衡器

以下代码段启用一个连接耗尽策略，该策略在 60 秒之后结束与取消注册或运行状况不佳的实例的连接。

```
"ElasticLoadBalancer" : {
  "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
  "Properties" : {
    "AvailabilityZones" : { "Fn::GetAZs" : "" },
    "Instances" : [ { "Ref" : "Ec2Instance1" },{ "Ref" : "Ec2Instance2" } ],
    "Listeners": [{
      "LoadBalancerPort": "80",
      "InstancePort": "80",
      "Protocol": "HTTP"
    }],
    "HealthCheck": {
      "Target": "HTTP:80/",
      "HealthyThreshold": "3",
      "UnhealthyThreshold": "5",
```

```
      "Interval": "30",
      "Timeout": "5"
    },
    "ConnectionDrainingPolicy": {
      "Enabled" : "true",
      "Timeout" : "60"
    }
  }
}
```

更多示例

您可以从 AWS CloudFormation 示例模板中查看和下载 AWS CloudFormation 模板的示例。包括：

- ELBSample.template：具有运行状况检查功能的负载均衡器。
- ELBStickinessSample.template：使用基于 Cookie 的粘性配置的负载均衡器示例。
- ELBWithLockedDownEC2Instances.template：一种负载均衡器，其实例仅接收来自负载均衡器的流量。
- ELBWithLockedDownAutoScaledInstances.template：具有仅从负载均衡器接收流量的 Auto Scaling 组的负载均衡器。
- ELBZoneApex.template：可将负载均衡器映射至 DNS 区域顶点。

另请参阅

- *Elastic Load Balancing API 参考* 中的 CreateLoadBalancer

AWS::IAM::AccessKey

Abstract

使用 AWS::IAM::AccessKey 资源生成私有访问密钥，并将其分配给 IAM 用户或 AWS 账户。

AWS::IAM::AccessKey 资源类型用于生成秘密访问密钥，并将其分配到 IAM 用户或 AWS 账户。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type": "AWS::IAM::AccessKey",
   "Properties": {
      "Serial (p. 344)": Integer,
      "Status (p. 344)": String,
      "UserName (p. 344)": String
   }
}
```

属性

序列号
: 此值特定于 AWS CloudFormation，并且只能*递增*。将此值递增就是通知 AWS CloudFormation 您需要轮换访问密钥。在更新堆栈时，AWS CloudFormation 会用新密钥替换现有访问密钥。

  必需：否

  类型：整数

  更新要求：替换 (p. 63)

状态
: 访问密钥的状态。

  必需：是

  类型：字符串

  有效值：`"Active" or "Inactive"`

  更新要求：无中断 (p. 63)

UserName
: 新密钥所属的用户的名称。

  必需：是

  类型：字符串

  更新要求：替换 (p. 63)

返回值

Ref

将此资源 ID 指定给内部 `Ref` 函数会返回 *`AccessKeyId`*。例如：`AKIAIOSFODNN7EXAMPLE`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

SecretAccessKey
: 返回指定 AWS::IAM::AccessKey 资源的私有访问密钥。例如：
  `wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY`.

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

若要查看 AWS::IAM::AccessKey 代码段，请参阅 声明 IAM 访问密钥资源 (p. 148)。

AWS::IAM::Group

Abstract

使用 AWS::IAM::Group 资源创建或更新 AWS Identity and Access Management 组。

AWS::IAM::Group 类型创建 Identity and Access Management (IAM) 组。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type": "AWS::IAM::Group",
   "Properties": {
      "   (p. 345)": String,
      "   (p. 345)": [ Policy1, ... ]
   }
}
```

属性

路径
: 该组的路径。有关路径的更多信息，请参阅 *Using IAM* 中的 Identifiers for IAM Entities。
必需：否
类型：字符串
更新要求：无中断 (p. 63)

策略
: 要添加到组中的策略。有关策略的信息，请参阅 *Using IAM* 中的 Overview of Policies。
必需：否
类型：AWS::IAM::Policy (p. 348) 类型列表
更新要求：无中断 (p. 63)

返回值

Ref

将此资源 ID 指定到内部 `Ref` 函数会返回 GroupName。例如：`mystack-mygroup-1DZETITOWEKVO`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Arn
: 返回 AWS::IAM::Group 资源的亚马逊资源名称 (ARN)。例如：
`arn:aws:iam::123456789012:group/mystack-mygroup-1DZETITOWEKVO`.

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

要查看 AWS::IAM::Group 片段，请参阅 声明 IAM 组资源 (p. 149)

AWS::IAM::InstanceProfile

Abstract

使用 AWS::IAM::InstanceProfile 资源创建可用于 IAM 角色的 IAM 实例配置文件。

创建可用于 EC2 实例的 IAM 角色的 AWS Identity and Access Management (IAM) 实例配置文件。

有关 IAM 角色的更多信息，请参阅 AWS Identity and Access Management User Guide 中的*使用角色*。

语法

```
{
   "Type": "AWS::IAM::InstanceProfile",
   "Properties": {
      "Path (p. 346)": String,
      "Roles (p. 346)": [ IAM Roles ]
   }
}
```

属性

路径
: 与此 IAM 实例配置文件相关的路径。有关 IAM 路径的信息，请参阅 *AWS Identity and Access Management User Guide* 中的 Friendly Names and Paths。
必需：是
类型：字符串
更新要求：替换 (p. 63)

角色
: 与此 IAM 实例配置文件相关的角色。
必需：是
类型：对 AWS::IAM::Role 的引用的列表。目前，最多只能为实例配置文件分配一个角色。
更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyProfile" }
```

对于具有逻辑 ID“MyProfile”的 IAM::InstanceProfile，`Ref` 将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Arn
: 返回实例配置文件的 Amazon 资源名称 (ARN)。例如：

```
{"Fn::GetAtt" : ["MyProfile", "Arn"] }
```

它将返回一个值，如“`arn:aws:iam::1234567890:instance-profile/MyProfile-ASDNSDLKJ`”。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

Example 带内嵌策略和实例配置文件的 IAM 角色

此示例将显示 IAM::Role 中的内嵌策略。此策略被指定内联在 IAM::Role Policies 属性中。

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/",
            "Policies": [ {
               "PolicyName": "root",
               "PolicyDocument": {
                  "Version" : "2012-10-17",
                  "Statement": [ {
                     "Effect": "Allow",
                     "Action": "*",
                     "Resource": "*"
                  } ]
               }
               } ]
            }
         },
         "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ {
               "Ref": "RootRole"
            } ]
         }
      }
   }
}
```

AWS::IAM::Policy

Abstract

使用 AWS::IAM::Policy 资源将 IAM 策略应用于用户或组或更新该策略。

AWS::IAM::Policy 类型将 Identity and Access Management (IAM) 策略应用于用户或组。有关 IAM 策略的更多信息，请参阅 *AWS Identity and Access Management 用户指南*中的 Overview of Policies。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type": "AWS::IAM::Policy",
   "Properties": {
      "Groups (p. 349)" : [ String, ... ],
      "PolicyDocument (p. 349)" : JSON,
      "PolicyName (p. 349)" : String,
      "Roles (p. 349)" : [ String, ...
      "Users (p. 349)" : [ String, ... ],
   }
}
```

属性

组
要向其添加策略的组的名称。

Required: Conditional.

Type: A list of strings.

更新要求：无中断 (p. 63)

PolicyDocument
策略文档，其中包含要添加到指定用户或组的权限。

Required: Yes.

Type: JSON object.

更新要求：无中断 (p. 63)

PolicyName
策略的名称。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

角色
要附加到此策略的 AWS::IAM::Role (p. 351) 的名称。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

用户
您要为其添加策略的用户的名称。

Required: Conditional.

Type: A list of strings.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

带有策略组的 IAM 策略

```
{
   "Type" : "AWS::IAM::Policy",
   "Properties" : {
      "PolicyName" : "CFNUsers",
      "PolicyDocument" : {
         "Version" : "2012-10-17",
         "Statement": [ {
         "Effect"   : "Allow",
         "Action"   : [
            "cloudformation:Describe*",
            "cloudformation:List*",
            "cloudformation:Get*"
         ],
         "Resource" : "*"
         } ]
      },
      "Groups" : [ { "Ref" : "CFNUserGroup" } ]
   }
}
```

此代码段来自 IAM_Users_Groups_and_Policies.template

带有指定角色的 IAM 策略

```
{
   "Type": "AWS::IAM::Policy",
   "Properties": {
      "PolicyName": "root",
      "PolicyDocument": {
         "Version" : "2012-10-17",
         "Statement": [
            { "Effect": "Allow", "Action": "*", "Resource": "*" }
         ]
      },
      "Roles": [ { "Ref": "RootRole" } ]
   }
}
```

此代码段来自 auto_scaling_with_instance_profile.template。

若要查看更多 AWS::IAM::Policy 代码段，请参阅 声明 IAM 策略 (p. 150)。

AWS::IAM::Role

Abstract

使用 AWS::IAM::Role 资源创建 AWS Identity and Access Management 角色。

创建 AWS Identity and Access Management (IAM) 角色。通过 IAM 角色，在 Amazon EC2 实例上运行的应用程序能够安全访问您的 AWS 资源。

有关 IAM 角色的更多信息，请参阅 AWS Identity and Access Management User Guide 中的*使用角色*。

语法

```
{
   "Type": "AWS::IAM::Role",
   "Properties": {
      "AssumeRolePolicyDocument (p. 351)": { JSON },
      "Path (p. 351)": String,
      "Policies (p. 351)": [ IAM policy, ... ]
   }
}
```

属性

AssumeRolePolicyDocument
与此角色关联的 IAM 代入角色策略。

Required: Yes.

Type: 一个 JSON 策略文档。

更新要求：无中断 (p. 63)

Note

一个角色只能与一个代入角色策略关联。有关代入角色策略的示例，请参阅模板示例 (p. 353)。

路径
与角色相关的路径。有关 IAM 路径的信息，请参阅 *使用 IAM* 中的友好名称和路径。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

策略
要与此角色关联的嵌入式策略的列表。策略也可以在外部指定。有关演示嵌入式和外部策略的示例模板，请参阅 模板示例 (p. 353)。

Required: No.

Type: IAM 策略的列表。

更新要求：无中断 (p. 63)

IAM 角色策略备注

有关 IAM 策略和策略文档的一般信息，请参阅 *使用 IAM* 中的如何编写策略。

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "RootRole" }
```

对于具有逻辑 ID“RootRole”的 IAM::Role，`Ref` 将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Arn
: 返回实例配置文件的 Amazon 资源名称 (ARN)。例如：

```
{"Fn::GetAtt" : ["MyRole", "Arn"] }
```

这会返回一个值，如 “`arn:aws:iam::1234567890:role/MyRole-AJJHDSKSDF`”。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

Example 带内嵌策略和实例配置文件的 IAM 角色

此示例将显示 IAM::Role 中的内嵌策略。此策略被指定内联在 IAM::Role Policies 属性中。

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/",
            "Policies": [ {
               "PolicyName": "root",
               "PolicyDocument": {
                  "Version" : "2012-10-17",
                  "Statement": [ {
                     "Effect": "Allow",
                     "Action": "*",
                     "Resource": "*"
                  } ]
               }
               } ]
            }
         },
         "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ {
               "Ref": "RootRole"
            } ]
         }
      }
   }
}
```

Example 带外部策略和实例配置文件的 IAM 角色

在此示例中，Policy 和 InstanceProfile 资源被指定在 IAM 角色外部。此类资源是通过在其各自的角色属性中指定名称“RootRole”的角色。

```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
      "RootRole": {
         "Type": "AWS::IAM::Role",
         "Properties": {
            "AssumeRolePolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Principal": {
                     "Service": [ "ec2.amazonaws.com" ]
                  },
                  "Action": [ "sts:AssumeRole" ]
               } ]
            },
            "Path": "/"
         }
      },
      "RolePolicies": {
         "Type": "AWS::IAM::Policy",
         "Properties": {
            "PolicyName": "root",
            "PolicyDocument": {
               "Version" : "2012-10-17",
               "Statement": [ {
                  "Effect": "Allow",
                  "Action": "*",
                  "Resource": "*"
               } ]
            },
            "Roles": [ {
               "Ref": "RootRole"
            } ]
         }
      },
      "RootInstanceProfile": {
         "Type": "AWS::IAM::InstanceProfile",
         "Properties": {
            "Path": "/",
            "Roles": [ {
               "Ref": "RootRole"
            } ]
         }
      }
   }
}
```

另请参阅

- Identity and Access Management (IAM) 模板代码段 (p. 146)

- AWS::IAM::InstanceProfile (p. 346)

AWS::IAM::User

Abstract

使用 AWS::IAM::User 资源创建或更新 AWS Identity and Access Management 用户。

AWS::IAM::User 类型创建用户。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type": "AWS::IAM::User",
   "Properties": {
      "Path (p. 355)": String,
      "Groups (p. 355)": [ Group, ... ],
      "LoginProfile (p. 355)": { "Password" : String },
      "Policies (p. 356)": [ Embedded IAM Policies ]
   }
}
```

属性

路径
: 用户名称的路径。更多有关路径的信息，请参阅 Using AWS Identity and Access Management 的 IAM Entities 部分。

  必需：否

  类型：字符串

  更新要求：无中断 (p. 63)

组
: 您要向其添加用户的组的名称。

  必需：否

  类型：组列表

  更新要求：无中断 (p. 63)

LoginProfile
: 为用户创建登录配置文件，以便用户访问 AWS 管理控制台之类的 AWS 服务。

  LoginProfile 类型是 AWS::IAM::User 类型中的一个嵌入式属性。LoginProfile 属性只包含一个字段：*Password*，该字段将一个字符串用作其值。例如：

  ```
  "LoginProfile": { "Password": "myP@ssW0rd" }
  ```

  必需：否

  类型：LoginProfile 类型

更新要求：无中断 (p. 63)

策略
对用户应用指定的策略。有关策略的信息，请参阅 Overview of Policies，位于[使用 IAM].

必需：否

类型：AWS::IAM::Policy (p. 348) 类型列表

更新要求：无中断 (p. 63)

返回值

Ref

将此资源 ID 指定到内部 Ref 函数将返回 UserName。例如：`mystack-myuser-1CCXAFG2H2U4D`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Arn
返回指定 AWS::IAM::User 资源的 Amazon 资源名称 (ARN)。例如：
`arn:aws:iam::123456789012:user/mystack-myuser-1CCXAFG2H2U4D`.

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

模板示例

要查看 AWS::IAM::User 代码段，请参阅：声明 IAM 用户资源 (p. 147)

AWS::IAM::UserToGroupAddition

Abstract

使用 AWS::IAM::UserToGroupAddition 资源将 IAM 用户添加到组。

AWS::IAM::UserToGroupAddition 类型将 AWS Identity and Access Management (IAM) 用户添加到组。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

语法

```
{
   "Type": "AWS::IAM::UserToGroupAddition",
   "Properties": {
      "GroupName (p. 357)": String,
      "Users (p. 357)": [ User1, ... ]
   }
}
```

属性

GroupName
向其添加用户的目标组名。
必需：是
类型：字符串
更新要求：无中断 (p. 63)

用户
必需：是
类型：用户列表
更新要求：无中断 (p. 63)

返回值

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyUserToGroupAddition" }
```

对于具有逻辑 ID“MyUserToGroupAddition”的 AWS::IAM::UserToGroupAddition，`Ref` 将返回 AWS 资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板示例

若要查看 AWS::IAM::UserToGroupAddition 代码段，请参阅 添加用户到组中 (p. 150)。

AWS::Kinesis::Stream

Abstract

使用 AWS::Kinesis::Stream 资源可创建 Amazon Kinesis 流。

创建捕获和传输从数据源发出的数据记录的 Amazon Kinesis 流。有关创建流的特定信息，请参阅 *Amazon Kinesis API Reference* 中的 CreateStream。

语法

```
{
   "Type" : "AWS::Kinesis::Stream",
   "Properties" : {
      "ShardCount (p. 357)" : Integer
   }
}
```

属性

ShardCount
流使用的分片数。要获得更大的配置吞吐量，请增加分片数。

Required: Yes.

Type: Integer

更新要求：替换 (p. 63)

返回值

Ref

指定 AWS::Kinesis::Stream 资源作为 `Ref` 函数的参数时，AWS CloudFormation 返回流名称（物理 ID）。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

AWS::OpsWorks::App

Abstract

使用 AWS::OpsWorks::App 资源类型定义 AWS OpsWorks 应用程序。

为 AWS OpsWorks 堆栈定义 AWS OpsWorks 应用程序。该应用程序表示您要在应用程序服务器上运行的代码。

语法

```
{
  "Type": "AWS::OpsWorks::App",
  "Properties": {
    "AppSource (p. 358)" : Source,
    "    (p. 358)" : { String:String, ... },
    "   (p. 359)" : String,
    "  (p. 359)" :  [ String, ... ],
    "EnableSsl (p. 359)" : Boolean,
    "   (p. 359)" : String,
    "Shortname (p. 359)" : String,
    "SslConfiguration (p. 359)" : { SslConfiguration },
    "   ID (p. 359)" : String,
    "   (p. 360)" : String
  }
}
```

属性

AppSource
包含从存储库检索应用程序所需的信息。

Required: No.

类型：AWS OpsWorks Source 类型 (p. 467)

更新要求：无中断 (p. 63)

属性
要添加到堆栈属性包的一个或多个用户定义键/值对。

Required: No.

类型：键/值对的列表

更新要求：无中断 (p. 63)

说明
应用程序的描述。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

域
应用程序虚拟主机设置（多个域以逗号分隔）。例如，'www.example.com、example.com'。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

EnableSsl
是否为此应用程序启用 SSL。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

名称
AWS OpsWorks 应用程序名称。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

Shortname
由 AWS OpsWorks 和 Chef 配方在内部使用的应用程序短名。

Required: No.

Type: String.

更新要求：替换 (p. 63)

SslConfiguration
SSL 配置

Required: No.

类型：AWS OpsWorks SslConfiguration 类型 (p. 468)

更新要求：无中断 (p. 63)

堆栈 ID
此应用程序将关联的 AWS OpsWorks 堆栈 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

类型
: 应用程序类型。每种支持的类型与特定层关联。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateApp。

  Required: Yes.

  Type: String.

  更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myApp" }
```

对于 AWS OpsWorks 堆栈 `myApp`，`Ref` 返回 AWS OpsWorks 应用程序 ID。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段创建在 Git 存储库中使用 PHP 应用程序的 AWS OpsWorks 应用程序：

```
"myApp" : {
  "Type" : "AWS::OpsWorks::App",
  "Properties" : {
    "StackId" : {"Ref":"myStack"},
    "Type" : "php",
    "Name" : {"myPHPapp"},
    "AppSource" : {
      "Type" : "git",
      "Url" : "git://github.com/amazonwebservices/opsworks-demo-php-simple-
app.git",
      "Revision" : "version1"
    }
  }
}
```

另请参阅

- AWS::OpsWorks::Stack (p. 368)
- AWS::OpsWorks::Layer (p. 364)
- AWS::OpsWorks::Instance (p. 361)

AWS::OpsWorks::ElasticLoadBalancerAttachment

Abstract

使用 AWS::OpsWorks::ElasticLoadBalancerAttachment 资源类型在 AWS OpsWorks 层上定义 Elastic Load Balancing 附加操作。

将 Elastic Load Balancing 负载均衡器附加到指定的 AWS OpsWorks 层。

语法

```
{
  "Type": "AWS::OpsWorks::ElasticLoadBalancerAttachment",
  "Properties": {
    "ElasticLoadBalancerName (p. 361)" : String,
    "  ID (p. 361)" : String
  }
}
```

属性

ElasticLoadBalancerName
Elastic Load Balancing 负载均衡器名称。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

层 ID
Elastic Load Balancing 负载均衡器将附加到的 AWS OpsWorks 层 ID。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

模板代码段

以下代码段指定将一个负载均衡器附加到 AWS OpsWorks 层（在相同模板中的其他位置描述了两者）：

```
"ELBAttachment" : {
  "Type" : "AWS::OpsWorks::ElasticLoadBalancerAttachment",
    "Properties" : {
      "ElasticLoadBalancerName" : { "Ref" : "ELB" },
      "LayerId" : { "Ref" : "Layer" }
    }
}
```

另请参阅

- AWS::OpsWorks::Layer (p. 364)

AWS::OpsWorks::Instance

Abstract

为 AWS OpsWorks 堆栈创建实例。

为 AWS OpsWorks 堆栈创建实例。这些实例表示 Amazon EC2 实例，例如用于处理应用程序服务和流量均衡工作的实例。

语法

```
{
  "Type": "AWS::OpsWorks::Instance",
  "Properties": {
    "AmiId (p. 362)" : String,
    "   (p. 362)" : String,
    "    (p. 362)" : String,
    "InstallUpdatesOnBoot (p. 362)" : Boolean,
    "     (p. 363)" : String,
    "LayerIds (p. 363)" :  [ String, ... ],
    "Os (p. 363)" : String,
    "RootDeviceType (p. 363)" : String,
    "SshKeyName (p. 363)" : String,
    "   ID (p. 363)" : String,
    "SubnetId (p. 363)" : String
  }
}
```

属性

AmiId
: 要用于创建实例的自定义 AMI 的 ID。AMI 应基于标准 AWS OpsWorks API 之一。

  Required: No.

  Type: String.

  更新要求：时而中断 (p. 63)

架构
: 实例架构。

  Required: No.

  Type: String.

  更新要求：时而中断 (p. 63)

可用区
: 实例可用区。

  Required: No.

  Type: String.

  更新要求：替换 (p. 63)

InstallUpdatesOnBoot
: 是否在实例启动时安装操作系统和软件包更新。

  Required: No.

  Type: Boolean.

  更新要求：时而中断 (p. 63)

实例类型
AWS OpsWorks 必须支持的实例类型。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateInstance。

Required: Yes.

Type: String.

更新要求：时而中断 (p. 63)

LayerIds
将与此实例关联的 AWS OpsWorks 层的 ID。

Required: Yes.

Type: A list of strings.

更新要求：时而中断 (p. 63)

Os
实例操作系统。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateInstance。

Required: No.

Type: String.

更新要求：时而中断 (p. 63)

RootDeviceType
实例根设备类型。

Required: No.

Type: String.

更新要求：替换 (p. 63)

SshKeyName
实例 SSH 密钥名称。

Required: No.

Type: String.

更新要求：时而中断 (p. 63)

堆栈 ID
此实例将关联的 AWS OpsWorks 堆栈的 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

SubnetId
实例子网的 ID。如果堆栈在 VPC 中运行，则您可以使用此参数覆盖堆栈的默认子网 ID 值和指示 AWS OpsWorks 在不同子网中启动实例。

Required: No.

Type: String.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myInstance1" }
```

对于 AWS OpsWorks 实例 `myInstance1`，`Ref` 返回 AWS OpsWorks 实例 ID。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段创建两个与 `myStack` AWS OpsWorks 堆栈和 `myLayer` AWS OpsWorks 层关联的 AWS OpsWorks 实例：

```
"myInstance1" : {
  "Type" : "AWS::OpsWorks::Instance",
  "Properties" : {
    "StackId" : {"Ref":"myStack"},
    "LayerIds" : [{"Ref":"myLayer"}],
    "InstanceType" : "m1.small"
  }
},

"myInstance2" : {
  "Type" : "AWS::OpsWorks::Instance",
  "Properties" : {
    "StackId" : {"Ref":"myStack"},
    "LayerIds" : [{"Ref":"myLayer"}],
    "InstanceType" : "m1.small"
  }
}
```

另请参阅

- AWS::OpsWorks::Stack (p. 368)
- AWS::OpsWorks::Layer (p. 364)
- AWS::OpsWorks::App (p. 358)

AWS::OpsWorks::Layer

Abstract

使用 AWS::OpsWorks::Layer 资源类型创建 AWS OpsWorks 层。

创建 AWS OpsWorks 层。例如，层定义安装的软件包和应用程序及其配置方式。

语法

```
{
  "Type": "AWS::OpsWorks::Layer",
  "Properties": {
    "    (p. 365)" : { String:String, ... },
    "AutoAssignElasticIps (p. 365)" : Boolean,
    "AutoAssignPublicIps (p. 365)" : Boolean,
    "CustomInstanceProfileArn (p. 365)" : String,
    "CustomRecipes (p. 366)" : Recipes,
    "CustomSecurityGroupIds (p. 366)" :  [ String, ... ],
    "EnableAutoHealing (p. 366)" : Boolean,
    "InstallUpdatesOnBoot (p. 366)" : Boolean,
    "   (p. 366)" : String,
    "    (p. 366)" : [ String, ... ],
    "Shortname (p. 366)" : String,
    "   ID (p. 367)" : String,
    "   (p. 367)" : String,
    "VolumeConfigurations (p. 367)" : [ VolumeConfiguration, ... ]
  }
}
```

属性

属性
要添加到堆栈属性包的一个或多个用户定义键/值对。

Required: No.

类型：键/值对的列表

更新要求：无中断 (p. 63)

AutoAssignElasticIps
是否自动将弹性 IP 地址分配给此层中的 Amazon EC2 实例。

Required: Yes.

Type: Boolean.

更新要求：无中断 (p. 63)

AutoAssignPublicIps
对于在 VPC 运行的 AWS OpsWorks 堆栈，是否自动将公有 IP 地址分配给此层中的 Amazon EC2 实例。

Required: Yes.

Type: Boolean.

更新要求：无中断 (p. 63)

CustomInstanceProfileArn
要用于此层中 Amazon EC2 实例的 IAM 实例配置文件的亚马逊资源名称 (ARN)。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

CustomRecipes
此层的自定义事件配方。

Required: No.

类型：AWS OpsWorks Recipes 类型 (p. 466)

更新要求：无中断 (p. 63)

CustomSecurityGroupIds
此层的自定义安全组 ID。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

EnableAutoHealing
是否自动修复断开连接或超时的 Amazon EC2 实例。

Required: Yes.

Type: Boolean.

更新要求：无中断 (p. 63)

InstallUpdatesOnBoot
是否在实例启动时安装操作系统和软件包更新。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

名称
AWS OpsWorks 层名称。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

软件包
用于此层的软件包。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

Shortname
由 AWS OpsWorks 和 Chef 配方在内部使用的层短名。短名也用作安装应用程序文件的目录的名称。

该名称可以最多包含 200 个字符，限制为字母数字字符、“-”、“_”和“.”。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

堆栈 ID
: 此层将关联的 AWS OpsWorks 堆栈的 ID。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

类型
: 层类型。堆栈不能具有相同类型的多个层。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateLayer。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

VolumeConfigurations
: 描述此层的 Amazon EBS 卷。

Required: No.

类型：AWS OpsWorks VolumeConfiguration 类型 (p. 470)

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myLayer" }
```

对于 AWS OpsWorks 层 `myLayer`，`Ref` 返回 AWS OpsWorks 层 ID。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段创建与 `myStack` AWS OpsWorks 堆栈关联的 AWS OpsWorks PHP 层。该层依赖于 `myApp` AWS OpsWorks 应用程序。

```
"myLayer": {
  "Type": "AWS::OpsWorks::Layer",
  "DependsOn": "myApp",
  "Properties": {
    "StackId": {"Ref": "myStack"},
 "Name": "PHP App Server",
    "Type": "php-app",
    "Shortname" : "php-app",
    "EnableAutoHealing" : "true",
    "AutoAssignElasticIps" : "false",
    "AutoAssignPublicIps" : "true",
    "Name": "MyPHPApp"
```

```
  }
}
```

另请参阅

- AWS::OpsWorks::Stack (p. 368)
- AWS::OpsWorks::App (p. 358)
- AWS::OpsWorks::Instance (p. 361)

AWS::OpsWorks::Stack

Abstract

使用 AWS::OpsWorks::Stack 资源类型创建 AWS OpsWorks 堆栈。

创建 AWS OpsWorks 堆栈。AWS OpsWorks 堆栈表示要共同管理的一组实例，原因通常是它们具有共同用途（例如为 PHP 应用程序提供服务）。

语法

```
{
  "Type": "AWS::OpsWorks::Stack",
  "Properties": {
    "    (p. 368)" : { String:String, ... },
    "ConfigurationManager (p. 368)" : { StackConfigurationManager },
    "CustomCookbooksSource (p. 369)" : { Source },
    "CustomJson (p. 369)" : JSON,
    "DefaultAvailabilityZone (p. 369)" : String,
    "DefaultInstanceProfileArn (p. 369)" : String,
    "DefaultOs (p. 369)" : String,
    "DefaultRootDeviceType (p. 369)" : String,
    "DefaultSshKeyName (p. 370)" : String,
    "DefaultSubnetId (p. 370)" : String,
    "HostnameTheme (p. 370)" : String,
    "   (p. 370)" : String,
    "ServiceRoleArn (p. 370)" : String,
    "UseCustomCookbooks (p. 370)" : Boolean,
    "VpcId (p. 370)" : String
  }
}
```

属性

属性
: 要添加到堆栈属性包的一个或多个用户定义键/值对。

  Required: No.

  类型：键/值对的列表

  更新要求：无中断 (p. 63)

ConfigurationManager
: 配置管理器。创建堆栈时，使用配置管理器指定 Chef 版本。

Required: No.

类型：AWS OpsWorks StackConfigurationManager 类型 (p. 469)

更新要求：无中断 (p. 63)

CustomCookbooksSource
包含从存储库检索食谱所需的信息。

Required: No.

类型：AWS OpsWorks Source 类型 (p. 467)

更新要求：无中断 (p. 63)

CustomJson
包含用户定义的自定义 JSON 的字符串。自定义 JSON 用于覆盖对应默认堆栈配置 JSON 值。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateStack。

Important

AWS CloudFormation 以字符串形式提交所有 JSON 属性，包括任何布尔值或数值属性。如果您有需要布尔值或数值的配方，则必须修改配方以接受字符串并将这些字符串解释为布尔值或数值。

Required: No.

Type: JSON object.

更新要求：无中断 (p. 63)

DefaultAvailabilityZone
堆栈的默认可用区（必须处于指定区域中）。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

DefaultInstanceProfileArn
作为堆栈的所有 Amazon EC2 实例的默认配置文件的 IAM 实例配置文件亚马逊资源名称 (ARN)。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

DefaultOs
堆栈的默认操作系统。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateStack。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

DefaultRootDeviceType
默认根设备类型。此值在默认情况下用于堆栈中的所有实例，不过可以在创建实例时覆盖它。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateStack。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

DefaultSshKeyName
堆栈实例的默认 SSH 密钥。创建或更新实例时，可以覆盖此值。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

DefaultSubnetId
堆栈的默认子网 ID。所有实例都在此子网中启动，除非您在创建实例时指定其他子网 ID。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

HostnameTheme
堆栈的主机名主题（空格替换为下划线）。该主题用于为堆栈实例生成主机名。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 CreateStack。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

名称
AWS OpsWorks 堆栈的名称。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

ServiceRoleArn
AWS OpsWorks 用于代表您使用 AWS 资源的 AWS Identity and Access Management (IAM) 角色。您必须指定现有 IAM 角色的亚马逊资源名称 (ARN)。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

UseCustomCookbooks
堆栈是否使用自定义食谱。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

VpcId
堆栈将在其中启动到 VPC（必须处于指定区域中）的 ID。所有实例都在此 VPC 中启动。

Required: No.

Type: String.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myStack" }
```

对于 AWS OpsWorks 堆栈 `myStack`，`Ref` 返回 AWS OpsWorks 堆栈 ID。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段创建一个使用默认服务角色和 Amazon EC2 角色（在首次使用 AWS OpsWorks 之后创建）的 AWS OpsWorks 堆栈：

```
"myStack" : {
  "Type" : "AWS::OpsWorks::Stack",
  "Properties" : {
    "Name" : {"Ref":"OpsWorksStackName"},
   "ServiceRoleArn" : { "Fn::Join": ["", ["arn:aws:iam::", {"Ref":"AWS::Accoun
tId"}, ":role/aws-opsworks-service-role"]] },
    "DefaultInstanceProfileArn" : { "Fn::Join": ["", ["arn:aws:iam::",
{"Ref":"AWS::AccountId"}, ":instance-profile/aws-opsworks-ec2-role"]] },
    "DefaultSshKeyName" : {"Ref":"KeyName"}
  }
}
```

有关完整示例 AWS OpsWorks 模板，请参阅 AWS OpsWorks 代码段 (p. 158)。

另请参阅

- AWS::OpsWorks::Layer (p. 364)
- AWS::OpsWorks::App (p. 358)
- AWS::OpsWorks::Instance (p. 361)

AWS::Redshift::Cluster

Abstract

使用 AWS::Redshift::Cluster 资源类型创建 Amazon Redshift 集群。

创建 Amazon Redshift 集群。集群是由计算节点集组成的完全托管的数据仓库。有关默认值和有效值的更多信息，请参阅 *Amazon Redshift API Reference* 中的 CreateCluster。

语法

```
{
  "Type": "AWS::Redshift::Cluster",
  "Properties": {
    "AllowVersionUpgrade (p. 372)" : Boolean,
```

```
    "AutomatedSnapshotRetentionPeriod (p. 372)" : Integer,
    "    (p. 372)" : String,
    "ClusterParameterGroupName (p. 372)" : String,
    "ClusterSecurityGroups (p. 373)" : [ String, ... ],
    "ClusterSubnetGroupName (p. 373)" : String,
    "ClusterType (p. 373)" : String,
    "ClusterVersion (p. 373)" : String,
    "DBName (p. 373)" : String,
    "ElasticIp (p. 373)" : String,
    "Encrypted (p. 373)" : Boolean,
    "HsmClientCertificateIdentifier (p. 374)" : String,
    "HsmConfigurationIdentifier (p. 374)" : String,
    "MasterUsername (p. 374)" : String,
    "MasterUserPassword (p. 374)" : String,
    "NodeType (p. 374)" : String,
    "NumberOfNodes (p. 374)" : Integer,
    "OwnerAccount (p. 374)" : String,
    "   (p. 374)" : Integer,
    "PreferredMaintenanceWindow (p. 375)" : String,
    "PubliclyAccessible (p. 375)" : Boolean,
    "SnapshotClusterIdentifier (p. 375)" : String,
    "SnapshotIdentifier (p. 375)" : String,
    "VpcSecurityGroupIds (p. 375)" : [ String, ... ]
  }
}
```

属性

AllowVersionUpgrade
: 发布新版本的 Amazon Redshift 时，指示升级是否可以应用于在集群上运行的引擎。升级在维护时段中应用。

  Required: No.

  Type: Boolean.

  更新要求：无中断 (p. 63)

AutomatedSnapshotRetentionPeriod
: 保留自动快照的天数。如果将值设置为 `0`，则禁用自动快照。

  Required: No.

  Type: Integer

  更新要求：无中断 (p. 63)

可用区
: 要在其中配置 Amazon Redshift 集群的 Amazon EC2 可用区。例如，如果您在特定可用区中运行多个 Amazon EC2 实例，则可能希望在相同区域中配置集群以便减小网络延迟。

  Required: No.

  Type: String.

  更新要求：替换 (p. 63)

ClusterParameterGroupName
: 要与此集群关联的参数组的名称。

  Required: No.

Type: String.

更新要求：时而中断 (p. 63)

ClusterSecurityGroups
要与此集群关联的安全组的列表。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

ClusterSubnetGroupName
要与此集群关联的集群子网组的名称。

Required: No.

Type: String.

更新要求：替换 (p. 63)

ClusterType
集群的类型。可以指定 `single-node` 或 `multi-node`。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

ClusterVersion
要在集群上部署的 Amazon Redshift 引擎版本。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

DBName
创建集群时创建的第一个数据库的名称。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

ElasticIp
集群的弹性 IP (EIP) 地址。

Required: No.

Type: String.

更新要求：替换 (p. 63)

Encrypted
指示集群中的数据是否进行静态加密。

Required: No.

Type: Boolean.

更新要求：替换 (p. 63)

HsmClientCertificateIdentifier
指定 Amazon Redshift 集群用于检索 HSM 中存储的数据加密密钥的 HSM 客户端证书名称。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

HsmConfigurationIdentifier
指定 HSM 配置的名称，该配置包含 Amazon Redshift 集群可以用于在 HSM 中检索和存储密钥的信息。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

MasterUsername
与此集群的主用户账户关联的用户名称。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

MasterUserPassword
与此集群的主用户账户关联的密码。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

NodeType
为此集群配置的节点类型。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

NumberOfNodes
集群中的计算节点数。如果您为 `ClusterType` 参数指定 `multi-node`，则必须指定大于 1 的数字。

Required: Conditional.

Type: Integer

更新要求：无中断 (p. 63)

OwnerAccount
从其他 AWS 账户中的快照还原时，包含该快照的 12 位 AWS 账户 ID。

Required: No.

Type: String.

更新要求：替换 (p. 63)

端口
集群用于接受传入连接的端口号。

Required: No.

Type: Integer

更新要求：替换 (p. 63)

PreferredMaintenanceWindow
可以进行自动集群维护的每周时间范围（采用 UTC 格式）。时间范围的格式是 `ddd:hh24:mi-ddd:hh24:mi`。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

PubliclyAccessible
指示是否可以从公用网络访问集群。

Required: No.

Type: Boolean.

更新要求：替换 (p. 63)

SnapshotClusterIdentifier
希望使用其快照的集群标识符。

Required: No.

Type: String.

更新要求：替换 (p. 63)

SnapshotIdentifier
请求的快照的唯一标识符。此标识符必须对 AWS 账户中的所有快照是唯一的。

Required: No.

Type: String.

更新要求：替换 (p. 63)

VpcSecurityGroupIds
与此集群关联的 VPC 安全组的列表。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myCluster" }
```

对于 Amazon Redshift 集群 `myCluster`，`Ref` 返回集群的名称。

有关使用 Ref 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

Fn::GetAtt返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Endpoint.Address
: Amazon Redshift 集群的连接终端节点。例如：examplecluster.cg034hpkmmjt.us-east-1.redshift.amazonaws.com .

Endpoint.Port
: Amazon Redshift 集群用于接受连接的端口号。例如：5439.

模板代码段

以下代码段描述一个单节点 Amazon Redshift 集群。主用户密码引用自相同模板中的输入参数。

```
"myCluster" : {
  "Type": "AWS::Redshift::Cluster",
  "Properties": {
    "MasterUsername" : "master",
    "MasterUserPassword" : { "Ref" : "MasterUserPassword" },
    "NodeType" : "dw.hs1.xlarge",
    "ClusterType" : "single-node"
  }
}
```

AWS::Redshift::ClusterParameterGroup

Abstract

使用 AWS::Redshift::ClusterParameterGroup 资源类型创建 Amazon Redshift 参数组。

创建可以与 Amazon Redshift 集群关联的 Amazon Redshift 参数组。该组中的参数应用于在集群中创建的所有数据库。

语法

```
{
  "Type": "AWS::Redshift::ClusterParameterGroup",
  "Properties": {
    "    (p. 376)" : String,
    "ParameterGroupFamily (p. 377)" : String,
    "    (p. 377)" : [ Parameter, ... ]
  }
}
```

属性

说明
: 参数组的描述。

  Required: Yes.

  Type: String.

更新要求：替换 (p. 63)

ParameterGroupFamily
应用于此集群参数组的 Amazon Redshift 引擎版本。集群引擎版本确定可在 `Parameters` 属性中指定的参数集。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

参数
在 `ParameterGroupFamily` 属性中指定的 Amazon Redshift 引擎版本允许的参数名称和值的列表。有关更多信息，请参阅 *Amazon Redshift Management Guide* 中的 Amazon Redshift 参数组。

Required: No.

类型：Amazon Redshift 参数类型 (p. 471)

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myClusterParameterGroup" }
```

对于 Amazon Redshift 集群参数组 `myClusterParameterGroup`，`Ref` 返回集群参数组的名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段描述具有指定的一个参数的参数组：

```
"myClusterParameterGroup" : {
  "Type" : "AWS::Redshift::ClusterParameterGroup",
  "Properties" : {
    "Description" : "My parameter group",
    "ParameterGroupFamily" : "redshift-1.0",
    "Parameters" : [ {
      "ParameterName" : "enable_user_activity_logging",
      "ParameterValue" : "true"
    }]
  }
}
```

以下代码段使用 `wlm_json_configuration` 参数修改工作负载管理配置。该参数值是 JSON 对象，必须采用括在引号 (") 中的字符串形式传递。在 JSON 对象中仅使用单引号 (')。

```
"RedshiftClusterParameterGroup" : {
  "Type" : "AWS::Redshift::ClusterParameterGroup",
  "Properties" : {
```

```
    "Description" : "Cluster parameter group",
    "ParameterGroupFamily" : "redshift-1.0",
    "Parameters" : [{
      "ParameterName" : "wlm_json_configuration",
      "ParameterValue" : "[{'user_group':['ex
ample_user_group1'],'query_group':['example_query_group1'],'query_concur
rency':7},{'query_concurrency':5}]"
    }]
  }
}
```

AWS::Redshift::ClusterSecurityGroup

Abstract

使用 AWS::Redshift::ClusterSecurityGroup 资源类型创建 Amazon Redshift 安全组。

创建一个 Amazon Redshift 安全组。使用安全组可控制对不在 VPC 中的 Amazon Redshift 集群的访问。

语法

```
{
  "Type": "AWS::Redshift::ClusterSecurityGroup",
  "Properties": {
    "    (p. 378)" : String
  }
}
```

属性

说明
安全组的描述。

Required: Yes.

Type: String.

更新要求：替换 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myClusterSecurityGroup" }
```

对于 Amazon Redshift 集群安全组 `myClusterSecurityGroup`，`Ref` 返回集群安全组的名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段创建可以将集群安全组入口规则与之关联的 Amazon Redshift 集群安全组：

```
"myClusterSecurityGroup" : {
  "Type": "AWS::Redshift::ClusterSecurityGroup",
  "Properties": {
    "Description" : "Security group to determine where connections to the Amazon
 Redshift cluster can come from"
  }
}
```

另请参阅

- AWS::Redshift::ClusterSecurityGroupIngress (p. 379)

AWS::Redshift::ClusterSecurityGroupIngress

Abstract

使用 AWS::Redshift::ClusterSecurityGroupIngress 资源类型为 Amazon Redshift 安全组指定入站规则。

为 Amazon Redshift 安全组指定入站（传入）规则。

语法

```
{
  "Type": "AWS::Redshift::ClusterSecurityGroupIngress",
  "Properties": {
    "ClusterSecurityGroupName (p. 379)" : String,
    "CIDRIP (p. 379)" : String,
    "EC2SecurityGroupName (p. 380)" : String,
    "EC2SecurityGroupOwnerId (p. 380)" : String
  }
}
```

属性

ClusterSecurityGroupName
: 将与传入规则关联的 Amazon Redshift 安全组的名称。

  Required: Yes.

  Type: String.

  更新要求：替换 (p. 63)

CIDRIP
: 对 Amazon Redshift 安全组具有入站访问权限的 IP 地址范围。

  Required: No.

  Type: String.

  更新要求：替换 (p. 63)

EC2SecurityGroupName
将添加 Amazon Redshift 安全组的 Amazon EC2 安全组。

Required: No.

Type: String.

更新要求：替换 (p. 63)

EC2SecurityGroupOwnerId
`EC2SecurityGroupName` 参数指定的 Amazon EC2 安全组拥有者的 12 位 AWS 账户。

Required: No.

Type: String.

更新要求：替换 (p. 63)

模板代码段

以下代码段描述 Amazon Redshift 集群安全组的一个传入规则：

```
"myClusterSecurityGroupIngressIP" : {
  "Type": "AWS::Redshift::ClusterSecurityGroupIngress",
  "Properties": {
    "ClusterSecurityGroupName" : {"Ref":"myClusterSecurityGroup"},
    "CIDRIP" : "10.0.0.0/16"
  }
}
```

另请参阅

- AWS::Redshift::ClusterSecurityGroup (p. 378)

AWS::Redshift::ClusterSubnetGroup

Abstract

使用 AWS::Redshift::ClusterSubnetGroup 资源类型创建 Amazon Redshift 子网组。

创建 Amazon Redshift 子网组。创建 Amazon Redshift 子网组时，必须提供处于现有 Amazon VPC 中的一个或多个子网的列表。

语法

```
{
  "Type": "AWS::Redshift::ClusterSubnetGroup",
  "Properties": {
    "    (p. 381)" : String,
    "SubnetId (p. 381)" : [ String, ... ]
  }
}
```

属性

说明
: 子网组的描述。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

SubnetId
: VPC 子网 ID 的列表。您可以修改最多 20 个子网。

Required: Yes.

Type: A list of strings.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "myClusterSubnetGroup" }
```

对于 Amazon Redshift 集群子网组 `myClusterSubnetGroup`，`Ref` 返回集群子网组的名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板代码段

以下代码段为 Amazon Redshift 集群子网组指定一个子网。

```
"myClusterSubnetGroup" : {
  "Type": "AWS::Redshift::ClusterSubnetGroup",
  "Properties": {
    "Description" : "My ClusterSubnetGroup",
    "SubnetIds" : ["subnet-7fbc2813"]
  }
}
```

AWS::RDS::DBInstance

Abstract

使用 AWS::RDS::DBInstance 资源创建 Amazon RDS 数据库实例。

类型 AWS::RDS::DBInstance 创建 Amazon RDS 数据库实例。有关配置 RDS 数据库实例的详细信息，请参阅 CreateDBInstance。

Important

如果数据库实例在更新过程中进行删除或替换，则所有自动快照都会删除。但是，手动数据库快照会保留。在需要替换的更新过程中，您可以应用堆栈策略以防止替换数据库实例。有关更多信息，请参阅 防止更新堆栈资源 (p. 69)。

语法

```
{
   "Type" : "AWS::RDS::DBInstance",
   "Properties" :
   {
      "AllocatedStorage (p. 382)" : String,
      "AutoMinorVersionUpgrade (p. 383)" : Boolean,
      "AvailabilityZone (p. 383)" : String,
      "BackupRetentionPeriod (p. 383)" : String,
      "DBInstanceClass (p. 383)" : String,
      "          (p. 383)" : String,
      "DBName (p. 384)" : String,
      "DBParameterGroupName (p. 384)" : String,
      "DBSecurityGroups (p. 384)" : [ String, ... ],
      "DBSnapshotIdentifier (p. 384)" : String,
      "DBSubnetGroupName (p. 384)" : String,
      "Engine (p. 385)" : String,
      "EngineVersion (p. 385)" : String,
      "Iops (p. 385)" : Number,
      "LicenseModel (p. 385)" : String,
      "MasterUsername (p. 385)" : String,
      "MasterUserPassword (p. 385)" : String,
      "MultiAZ (p. 386)" : Boolean,
      "Port (p. 386)" : String,
      "PreferredBackupWindow (p. 386)" : String,
      "PreferredMaintenanceWindow (p. 386)" : String,
      "SourceDBInstanceIdentifier (p. 386)" : String,
      "Tags (p. 386)" : [ Resource Tag, ..., ],
      "VPCSecurityGroups (p. 387)" : [ String, ... ]
   }
}
```

属性

AllocatedStorage
已分配存储的大小（以 GB 计算）。

如果在 *Iops* 参数中使用了任何值，则 *AllocatedStorage* 必须至少为 100 GB，这相当于最小 *Iops* 值 1000。如果 *Iops* 增加（以 1000 IOPS 增量），则 *AllocatedStorage* 必须也相应增加（以 100 GB 增量）。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

AllowMajorVersionUpgrade
指示是否允许主要版本升级。更改此参数不会导致中断，所做更改会尽快以异步方式应用。

约束：若指定不同于数据库实例的当前主要版本的 `EngineVersion`，则必须将此参数设置为 `true`。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)

AutoMinorVersionUpgrade
指示在维护窗口期间，将对该数据库实例自动应用次要引擎升级。默认值为 `true`。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63)或时而中断 (p. 63)。有关更多信息，请参阅 *Amazon Relational Database Service API 参考* 中的 ModifyDBInstance。

可用区
数据库实例所在的可用区的名称。如果 *`MultiAZ`* 参数设为 true，则无法设置 *`AvailabilityZone`* 参数。

Required: No.

Type: String.

更新要求：替换 (p. 63)

BackupRetentionPeriod
自动拍摄的数据库快照的保留天数。

Important

如果此数据库实例在更新过程中进行删除或替换，则所有自动快照都会删除。但是，手动数据库快照会保留。

Required: No.

Type: String.

更新要求：无中断 (p. 63)或时而中断 (p. 63)。有关更多信息，请参阅 *Amazon Relational Database Service API 参考* 中的 ModifyDBInstance。

DBInstanceClass
数据库实例的计算和内存容量级别名称。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

数据库实例标识符
数据库实例的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用于数据库实例。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

必需：否

类型：字符串

更新要求：无中断 (p. 63)或时而中断 (p. 63)。有关更多信息，请参阅 *Amazon Relational Database Service API 参考* 中的 ModifyDBInstance。

DBName
创建时提供的此示例的初始数据库的名称（如果指定了名称）。在数据库实例的使用期间会始终返回同一名称。

Required: No.

Type: String.

更新要求：替换 (p. 63)

DBParameterGroupName
现有数据库参数组的名称，或对模板中所创建的 AWS::RDS::DBParameterGroup (p. 389) 资源的引用。

Required: No.

Type: String.

更新要求：无中断 (p. 63)或时而中断 (p. 63)。有关更多信息，请参阅 *Amazon Relational Database Service API 参考* 中的 ModifyDBInstance。此外，如果引用的参数组的任何数据成员在更新期间发生更改，就可能要重新启动该数据库实例，从而造成运行中断。

DBSecurityGroups
要分配到 Amazon RDS 实例的数据库安全组的列表。该列表可包含现有数据库安全组的名称，或是对模板中所创建的 AWS::RDS::DBSecurityGroup (p. 392) 资源的引用。

如果您要设置 DBSecurityGroups，则不能设置 VPCSecurityGroups (p. 387)，反之亦然。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63)

DBSnapshotIdentifier
要用来恢复数据库的数据库快照的标识符。

通过指定此属性，您可以从指定的数据库快照创建数据库实例。如果 DBSnapshotIdentifier 属性是空字符串或 AWS::RDS::DBInstance 申明没有 DBSnapshotIdentifier 属性，则该数据库是作为新数据库创建的。如果此属性包含一个值（而不是空字符串），则 AWS CloudFormation 会从指定快照创建数据库。如果不存在具有指定名称的快照，则表明数据库创建失败，堆栈将回滚。

Required: No.

Type: String.

更新要求：替换 (p. 63)

DBSubnetGroupName
要与此数据库实例关联的数据库子网组。

如果不存在数据库子网组，则该实例为非 VPC 数据库实例。

有关在 VPC 中使用 Amazon RDS 的更多信息，请参阅 *Amazon Relational Database Service 开发人员指南* 中的 Using Amazon RDS with Amazon Virtual Private Cloud (VPC)。

Required: No.

Type: String.

更新要求：替换 (p. 63)

引擎
该数据库实例所使用的数据库引擎的名称。指定 `DBSnapshotIdentifier` 属性创建数据库实例时，此属性是可选的。

Required: Conditional.

Type: String.

更新要求：替换 (p. 63)

引擎版本
要使用的数据库引擎的版本号。

Required: No.

Type: String.

更新要求：时而中断 (p. 63)

Iops
数据库每秒应调配的 I/O 操作的次数 (IOPS)。此值可能是 1 000 - 10 000 之间的任意整数值，以 1 000 IOPS 为增量。

如果在 *`Iops`* 参数中使用了任何值，则 *`AllocatedStorage`* 必须至少为 100 GB，这相当于最小 *`Iops`* 值 1000。如果 *`Iops`* 增加（以 1000 IOPS 增量），则 *`AllocatedStorage`* 必须也相应增加（以 100 GB 增量）。

有关此参数的更多信息，请参阅 *Amazon Relational Database Service 用户指南*中的 Working with Provisioned IOPS Storage。

Required: No.

Type: Number.

更新要求：无中断 (p. 63)

LicenseModel
数据库实例的许可模型信息。

Required: No.

Type: String.

更新要求：替换 (p. 63).

MasterUsername
数据库实例的主用户名。指定 `DBSnapshotIdentifier` 属性创建数据库实例时，此属性是可选的。

Required: Conditional.

Type: String.

更新要求：替换 (p. 63).

MasterUserPassword
数据库实例的主密码。指定 `DBSnapshotIdentifier` 属性创建数据库实例时，此属性是可选的。

Required: Conditional.

Type: String.

更新要求：无中断 (p. 63).

MultiAZ
指定数据库实例是否是多可用区部署。如果 *MultiAZ* 参数设为 true，则无法设置 *AvailabilityZone* 参数。

Required: No.

Type: Boolean.

更新要求：无中断 (p. 63).

端口
实例的端口。

Required: No.

Type: String.

更新要求：替换 (p. 63).

PreferredBackupWindow
如果启用自动备份，该属性指定自动执行备份日常时间范围，如 BackupRetentionPeriod 所规定。

Required: No.

Type: String.

更新要求：无中断 (p. 63).

PreferredMaintenanceWindow
可能进行系统维护的每周时间范围（按世界协调时间计算）。

Required: No.

Type: String.

更新要求：无中断 (p. 63)或时而中断 (p. 63)。有关更多信息，请参阅 *Amazon Relational Database Service API 参考* 中的 ModifyDBInstance。

SourceDBInstanceIdentifier
如果您需要创建只读副本数据库实例，请指定源数据库实例的 ID。每个数据库实例都可具有特定数目的只读副本。有关更多信息，请参阅 *Amazon Relational Database Service 开发人员指南* 中的 Working with Read Replicas。

`SourceDBInstanceIdentifier` 属性决定了数据库实例是否是只读副本。如果您从当前模板中删除 `SourceDBInstanceIdentifier` 属性，然后更新堆栈，则将删除只读副本，并创建新的数据库实例（不是只读副本）。

如果您指定 `SourceDBInstanceIdentifier`，请勿将 `MultiAZ` 属性设置为 `true`，并且请勿指定 `DBSnapshotIdentifier` 属性。不能在多可用区中部署只读副本，也不能从快照创建只读副本。

Important

- 只读副本不支持删除策略。将会忽略任何与只读副本关联的删除策略。
- 您必须创建与源数据库实例处于相同区域的只读副本。当前不支持跨区域副本。

Required: No.

Type: String.

更新要求：替换 (p. 63).

标签
此 RDS 数据库实例的任意标签组（密钥/值对）。

Required: No.

Type: AWS CloudFormation 资源标签 (p. 471)

更新要求：无中断 (p. 63).

VPCSecurityGroups
要分配到 Amazon RDS 实例的 VPC 安全组的列表。该列表可包含现有 VPC 安全组的物理 ID 或对模板中所创建的 AWS::EC2::SecurityGroup (p. 292) 资源的引用。

如果您要设置 VPCSecurityGroups，则不能设置 DBSecurityGroups (p. 384)，反之亦然。

Important

您可以将堆栈中的数据库实例从 RDS 数据库安全组迁移到 VPC 安全组，但应该注意以下几点：

- 一旦建立了 VPC 安全组成员关系，就不能还原为使用 RDS 安全组。
- 在将数据库实例迁移至 VPC 安全组时，如果堆栈更新由于数据库实例更新中出现其他故障或者其他 AWS CloudFormation 资源中出现更新故障而发生回滚，则该回滚操作将由于无法还原至 RDS 安全组而失败。

因此，您仅应在该迁移操作是堆栈模板中的*唯*一更改时，将数据库实例迁移为使用 VPC 安全组。

Required: No.

Type: A list of strings.

更新要求：无中断 (p. 63).

更新和删除 AWS:RDS::DBInstances

Caution

在对标记为“*更新要求*：替换 (p. 63)”的属性执行更新时，AWS CloudFormation 首先会创建替换数据库实例资源，然后将引用从其他相关资源更改为指向该替换资源，最后才删除旧资源。

如果您没有先创建数据库快照，然后再更新堆栈，那么当您的数据库实例被替换后，您将丢失所有数据。为了保护数据，您应该采取以下预防措施：

1. 静默任何使用数据库实例的应用程序，确保未对数据库实例进行任何操作。
2. 创建数据库实例的快照。有关创建数据库快照的更多信息，请参阅 Creating a DB snapshot。
3. 如果您要用数据库快照还原实例，请使用该数据库实例的更改部分修改更新模板，并添加 DBSnapshotIdentifier 属性，其中包含要使用的数据库快照的 ID。
4. 更新堆栈。

有关更新此资源上的其他属性的更多信息，请参阅 ModifyDBInstance。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

您可以为数据库实例设置删除策略以控制 AWS CloudFormation 在堆栈删除后处理该实例的方式。对于 Amazon RDS 实例数据库，您可以选择*保留*实例，*删除*实例，或*创建实例的快照*。有关更多信息，请参阅 DeletionPolicy 属性 (p. 485)。

返回值

Ref

在将 RDS 数据库实例的逻辑名称提供给 `Ref` 内部函数时，`Ref` 将返回 DBInstanceIdentifier。例如：`mystack-mydb-ea5ugmfvuaxg`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

- Endpoint.Address

  数据库的连接终端节点。例如：`mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-1.rds.amazonaws.com`.
- Endpoint.Port

  数据库实例接受连接的端口编号。例如：`3306`.

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

Example 具有设定 MySQL 版本、标签和 DeletionPolicy 的 DBInstance

此示例说明如何设置 MySQL 版本，并设置了一个 DeletionPolicy 属性 (p. 485)。在设置了 "Snapshot" DeletionPolicy 后，AWS CloudFormation 将在堆栈删除期间，在删除此数据库实例之前获取此数据库实例的快照。此外，该示例还设置了包含简明数据库名称的标签。

```
"MyDB" : {
   "Type" : "AWS::RDS::DBInstance",
   "Properties" : {
      "DBName" : { "Ref" : "DBName" },
      "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
      "DBInstanceClass" : { "Ref" : "DBInstanceClass" },
      "Engine" : "MySQL",
      "EngineVersion" : "5.5",
      "MasterUsername" : { "Ref" : "DBUser" },
      "MasterUserPassword" : { "Ref" : "DBPassword" },
      "Tags" : [ { "Key" : "Name", "Value" : "My SQL Database" } ]
   },
   "DeletionPolicy" : "Snapshot"
}
```

Example 具有预置 IOPS 的 DBInstance

此示例在 Iops (p. 385) 属性中设置预置 IOPS 值。请注意，AllocatedStorage (p. 382) 属性是根据 IOPS 和 GiB 存储之间 10 :1 比率设置的。

```
"MyDB" : {
   "Type" : "AWS::RDS::DBInstance",
   "Properties" : {
      "AllocatedStorage" : "100",
      "DBInstanceClass" : "db.m1.small",
      "Engine" : "MySQL",
      "EngineVersion" : "5.5",
      "Iops" : "1000",
      "MasterUsername" : { "Ref" : "DBUser" },
      "MasterUserPassword" : { "Ref" : "DBPassword" }
   }
}
```

Example 只读副本 DBInstance

此示例为 `MyDB` 数据库实例创建名为 `MyDBreadreplica` 的只读副本。

```
"MyDB" : {
   "Type" : "AWS::RDS::DBInstance",
   "Properties" : {
      "DBName" : { "Ref" : "DBName" },
      "AllocatedStorage" : { "Ref" : "DBAllocatedStorage" },
      "DBInstanceClass" : { "Ref" : "DBClass" },
      "Engine" : "MySQL",
      "EngineVersion" : "5.6",
      "MasterUsername" : { "Ref" : "DBUser" } ,
      "MasterUserPassword" : { "Ref" : "DBPassword" },
      "Port" : "5804",
      "Tags" : [{"Key" : "Role", "Value" : "Primary"}]
   }
},

"MyDBreadreplica" : {
   "Type": "AWS::RDS::DBInstance",
   "Properties": {
      "SourceDBInstanceIdentifier": { "Ref" : "MyDB" },
      "Port" : "5802",
      "Tags" : [{"Key" : "Role", "Value" : "ReadRep"}]
      }
   }
}
```

若要查看更多 AWS::RDS::DBInstance 模板代码段，请参阅 Amazon RDS 模板代码段 (p. 165)。

AWS::RDS::DBParameterGroup

Abstract

使用 AWS::RDS::DBParameterGroup 资源为 Amazon RDS 数据库系列创建自定义参数组。

为 RDS 数据库系列创建自定义参数组。有关 RDS 参数组的更多信息，请参阅 *Amazon Relational Database Service User Guide* 中的 Working with DB Parameter Groups。

此类型可在模板中进行声明，并在 AWS::RDS::DBInstance (p. 381) 的 *DBParameterGroupName* 参数中进行引用。

Note

对 DBInstance 应用 ParameterGroup 可能需要重启实例，从而造成重启期间数据库运行中断。

语法

```
{
   "Type": "AWS::RDS::DBParameterGroup",
   "Properties" : {
      "Description (p. 390)" : String,
      "Family (p. 390)" : String,
      "Parameters (p. 390)" : DBParameters,
      "    (p. 391)" : [ Resource Tag, ... ]
   }
}
```

属性

说明
: 对 RDS 参数组的简明说明。例如，"My Parameter Group"。

Required: Yes.

*类型：*字符串

更新要求：无中断 (p. 63)

系列
: 此 RDS 参数组的数据库系列。例如，"MySQL5.1"。

Required: Yes.

*类型：*字符串

更新要求：无中断 (p. 63)

参数
: 要为 RDS 参数组设置的参数。

Required: No.

*类型：*DBParameters，一个包含字符串密钥/值对的 JSON 对象。例如：

```
"Parameters" : {
   "Key1" : "Value1",
   "Key2" : "Value2",
   "Key3" : "Value3"
}
```

更新要求：无中断 (p. 63)

标签
要附加到该 RDS 参数组的标签。

Required: No.

类型：资源标签 (p. 471)列表。

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyDBParameterGroup" }
```

对于具有逻辑 ID“MyDBParameterGroup”的 RDS::DBParameterGroup，`Ref` 将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

AWS::RDS::DBSubnetGroup

Abstract

使用 AWS::RDS::DBSubnetGroup 资源创建至少包含一个子网的 Amazon RDS 数据库子网组。

类型 AWS::RDS::DBSubnetGroup 可用于创建 RDS 数据库子网组。子网组在区域的两个可用区必须包含至少一个子网。

语法

```
{
   "Type" : "AWS::RDS::DBSubnetGroup",
   "Properties" : {
      "DBSubnetGroupDescription (p. 391)" : String,
      "SubnetIds (p. 391)" : [ String, ... ],
      "   (p. 392)" : [ Resource Tag, ... ]
   }
}
```

属性

DBSubnetGroupDescription
数据库子网组的说明。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

SubnetId
数据库子网组的 EC2 子网 ID。

Required: Yes.

Type: A list of strings.

更新要求：无中断 (p. 63)

标签

要附加到 RDS 数据库子网组的标签。

Required: No.

类型：资源标签 (p. 471)列表。

更新要求：无中断 (p. 63)

示例

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myDBSubnetGroup" : {
         "Type" : "AWS::RDS::DBSubnetGroup",
         "Properties" : {
            "DBSubnetGroupDescription" : "description",
            "SubnetIds" : [ "subnet-7b5b4112", "subnet-7b5b4115" ],
            "Tags" : [ {"key" : "value", "key2" : "value2"} ]
         }
      }
   }
}
```

另请参阅

- *Amazon Relational Database Service API 参考* 中的 CreateDBSubnetGroup
- *Amazon Relational Database Service API 参考* 中的 ModifyDBSubnetGroup
- AWS CloudFormation 堆栈更新 (p. 63)

AWS::RDS::DBSecurityGroup

Abstract

使用 AWS::RDS::DBSecurityGroup 资源创建或更新 Amazon RDS 数据库安全组。

AWS::RDS::DBSecurityGroup 类型用于创建或更新 Amazon RDS 数据库安全组。有关数据库安全组的更多信息，请参阅 *Amazon Relational Database Service 开发人员指南* 中的 Working with DB Security Groups。

有关数据库安全组设置的详细信息，请参阅 CreateDBSecurityGroup。

在将 AWS::RDS::DBSecurityGroup 作为参数指定给 `Ref` 函数时，AWS CloudFormation 返回 *`DBSecurityGroupName`* 的值。

语法

```
{
   "Type" : "AWS::RDS::DBSecurityGroup",
   "Properties" :
   {
      "EC2VpcId (p. 393)" : { "Ref" : "myVPC" },
     "DBSecurityGroupIngress (p. 393)" : [ RDS Security Group Rule (p. 472) object
 1, ... ],
      "GroupDescription (p. 393)" : String,
      "   (p. 393)" : [ Resource Tag, ... ]
   }
}
```

属性

EC2VpcId
: VPC 的 ID。标识此数据库安全组所属的 VPC。
类型：字符串
必需：条件。为 VPC 创建数据库安全组时必须指定；否则，不必指定。
更新要求：替换 (p. 63)

DBSecurityGroupIngress
: 针对 Amazon EC2 安全组或 IP 地址范围的网络接入授权。
类型：RDS 安全组规则 (p. 472)列表。
必需：是
更新要求：无中断 (p. 63)

GroupDescription
: 安全组的说明。
类型：字符串
必需：是
更新要求：替换 (p. 63)

标签
: 要附加到 Amazon RDS 数据库安全组的标签。

Required: No.

类型：资源标签 (p. 471)列表。

更新要求：无中断 (p. 63)

模板示例

Tip

有关更多 RDS 模板示例，请参阅 Amazon RDS 模板代码段 (p. 165)。

单一 VPC 安全组

此模板片段为 EC2SecurityGroupName 所引用，可用于创建/更新单一 VPC 安全组。

```
"DBSecurityGroup": {
   "Type": "AWS::RDS::DBSecurityGroup",
   "Properties": {
      "EC2VpcId" : { "Ref" : "VpcId" },
      "DBSecurityGroupIngress": [
         {"EC2SecurityGroupName": { "Ref": "WebServerSecurityGroup"}}
      ],
      "GroupDescription": "Frontend Access"
   }
},
```

多 VPC 安全组

此模板片段可创建/更新多个 VPC 安全组。

```
{
   "Resources" : {
      "DBinstance" : {
         "Type" : "AWS::RDS::DBInstance",
         "Properties" : {
            "DBSecurityGroups" : [ {"Ref" : "DbSecurityByEC2SecurityGroup"} ],

            "AllocatedStorage" : "5",
            "DBInstanceClass" : "db.m1.small",
            "Engine" : "MySQL",
            "MasterUsername" : "YourName",
            "MasterUserPassword" : "YourPassword"
         },
         "DeletionPolicy" : "Snapshot"
      },
      "DbSecurityByEC2SecurityGroup" : {
         "Type" : "AWS::RDS::DBSecurityGroup",
         "Properties" : {
            "GroupDescription" : "Ingress for Amazon EC2 security group",
            "DBSecurityGroupIngress" : [ {
                  "EC2SecurityGroupId" : "sg-b0ff1111",
                  "EC2SecurityGroupOwnerId" : "111122223333"
               }, {
                  "EC2SecurityGroupId" : "sg-ffd722222",
                  "EC2SecurityGroupOwnerId" : "111122223333"
               } ]
         }
      }
   }
}
```

AWS::RDS::DBSecurityGroupIngress

Abstract

使用 AWS::RDS::DBSecurityGroupIngress 资源启用对 DBSecurityGroup 的传入。

类型 AWS::RDS::DBSecurityGroupIngress 使用两种授权形式之一来启动对 DBSecurityGroup 的输入。第一，如果使用该数据库的应用程序是在 EC2 或 VPC 实例上运行的，则可向 DBSecurityGroup 添加 EC2 或 VPC 安全组。第二，如果访问您数据库的应用程序是在 Internet 上运行的，则 IP 范围可用。有关数据库安全组的更多信息，请参阅 Working with DB security groups。

此类型支持更新。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

有关数据库安全组传入的详细信息，请参阅 AuthorizeDBSecurityGroupIngress。

语法

```
{
   "CIDRIP (p. 395)": String,
   "DBSecurityGroupName (p. 395)": String,
   "EC2SecurityGroupId (p. 395)": String,
   "EC2SecurityGroupName (p. 395)": String,
   "EC2SecurityGroupOwnerId (p. 395)": String
}
```

属性

CIDRIP
要授权的 IP 范围。

有关 CIDR 范围的概述，请访问 Wikipedia Tutorial。
Type: String.

更新要求：无中断 (p. 63)

DBSecurityGroupName
要添加此传入的 AWS::RDS::DBSecurityGroup (p. 392) 的名称 (ARN)。
Type: String.
Required: Yes.

更新要求：无中断 (p. 63)

EC2SecurityGroupId
要授权的 VPC 或 EC2 安全组 ID。

对于 VPC 数据库安全组，请使用 EC2SecurityGroupId。对于 EC2 安全组，可使用 EC2SecurityGroupOwnerId 和 EC2SecurityGroupName 或 EC2SecurityGroupId。
Type: String.
Required: No.

更新要求：无中断 (p. 63)

EC2SecurityGroupName
要授权的 EC2 安全组的名称。

对于 VPC 数据库安全组，请使用 EC2SecurityGroupId。对于 EC2 安全组，可使用 EC2SecurityGroupOwnerId 和 EC2SecurityGroupName 或 EC2SecurityGroupId。
Type: String.
Required: No.

更新要求：无中断 (p. 63)

EC2SecurityGroupOwnerId
指定 EC2 安全组（在 EC2SecurityGroupName 参数中指定的）所有人的 AWS 账户。AWS 访问密钥 ID 不是认可的值。

对于 VPC 数据库安全组，请使用 EC2SecurityGroupId。对于 EC2 安全组，可使用 EC2SecurityGroupOwnerId 和 EC2SecurityGroupName 或 EC2SecurityGroupId。
Type: String.
Required: No.

更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

另请参阅

- *Amazon Relational Database Service API 参考* 中的 AuthorizeDBSecurityGroupIngress

AWS::Route53::RecordSet

Abstract

通过将 AWS::Route53::RecordSet 资源用作独立资源或嵌入式属性，向现有区域添加记录集。

AWS::Route53::RecordSet 类型可用作独立资源，或用作 AWS::Route53::RecordSetGroup (p. 400) 类型中的嵌入式属性。请注意，某些 AWS::Route53::RecordSet 属性仅在 AWS::Route53::RecordSetGroup 内使用时才有效。

请注意，在使用 AWS CloudFormation 向托管区域添加记录集时，必须已在 Amazon Route 53 中创建该托管区域。AWS CloudFormation 不会创建新的托管区域。

此类型支持更新。有关资源记录集更新的详细信息和约束，请参阅 Changing Your Resource Record Sets。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

有关每个属性的约束和值的更多信息，请参阅托管区域的 POST CreateHostedZone 以及资源记录集的 POST ChangeResourceRecordSet。

语法

```
{
   "Type" : "AWS::Route53::RecordSet",
   "Properties" : {
      "AliasTarget (p. 397)" : AliasTarget (p. 473),
      "Comment (p. 397)" : String,
      "HostedZoneId (p. 397)" : String,
      "HostedZoneName (p. 397)" : String,
      "Name (p. 397)" : String,
      "Region (p. 398)" : String,
      "ResourceRecords (p. 398)" : [ String ],
      "SetIdentifier (p. 398)" : String,
      "TTL (p. 399)" : String,
      "Type (p. 399)" : String,
```

```
      "Weight (p. 399)" : Integer
   }
}
```

属性

别名目标
*仅限别名资源记录集：*有关流量重定向到的域的信息。

如果指定了此属性，则不指定 TTL 属性。别名使用来自别名目标记录的 TTL 值。

有关别名资源记录集的更多信息，请参阅 *Route 53 开发人员指南*中的创建别名资源记录集以及 Route 53 API 引用中的 POST ChangeResourceRecordSets。

Note

目前，Amazon Route 53 仅支持 Elastic Load Balancing 的别名。

Required: Conditional. 若要创建别名资源记录集，则是必需的。

类型：别名目标 (p. 473)

更新要求：无中断 (p. 63)

评论
您要包含的有关托管区的所有评论。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

HostedZoneId
托管区的 ID。

Required: Conditional. 您必须指定 `HostedZoneName` 或 `HostedZoneId`，但不能两者都指定。

Type: String.

更新要求：替换 (p. 63)

HostedZoneName
您要向其添加记录集的托管区的域名。

在使用指定 `HostedZoneName` 的 AWS::Route53::RecordSet 创建堆栈时，AWS CloudFormation 会尝试查找名称与 HostedZoneName 匹配的托管区域。如果 AWS CloudFormation 找不到具有匹配域名称的托管区域，或者有一个以上具有指定域名称的托管区域，则 AWS CloudFormation 不会创建堆栈。

如果您有多个具有相同域名称的托管区域，则必须使用 `HostedZoneId` 显式指定托管区域。

Required: Conditional. 您必须指定 `HostedZoneName` 或 `HostedZoneId`，但不能两者都指定。

Type: String.

更新要求：替换 (p. 63)

名称
域的名称。必须为完整指定的域，以句点结尾（表示最后一个标签指示）。如果您删除了最后的句点，Amazon Route 53 会假定该域与根相关。

Required: Yes.

Type: String.

更新要求：无中断 (p. 63)

区域
仅限延迟资源记录集：在此资源记录集中指定的资源所在的 Amazon EC2 区域。一般而言，该资源可以为 AWS 资源，例如 Amazon EC2 实例或 Elastic Load Balancing 负载均衡器，并为 IP 地址或 DNS 域名所引用，具体取决区域记录的类型。

当 Route 53 收到查询您已创建延迟资源记录集的域的名称和类型的 DNS 查询时，Route 53 会选择在最终用户和相关 Amazon EC2 区域之间延迟时间最短的延迟资源记录集。然后，Route 53 会返回与所选资源记录集相关的值。

必须遵循以下限制：

- 您只能为每个延迟资源记录集指定一个 ResourceRecord。
- 您只能为每个 Amazon EC2 区域创建一个延迟资源记录集。
- 您不必为所有 Amazon EC2 区域创建延迟资源记录集。Route 53 会从您已创建延迟资源记录集的区域中选择延迟性能最佳的区域。
- 您不能同时创建名称和类型元素的值相同的权重资源记录集和延迟资源记录集。

区域名称的有效值：

- 亚太地区（东京）区域："ap-northeast-1"
- 亚太地区（新加坡）区域："ap-southeast-1"
- 亚太地区（悉尼）区域："ap-southeast-2"
- 欧洲（爱尔兰）区域："eu-west-1"
- 南美洲（圣保罗）区域："sa-east-1"
- 美国东部（弗吉尼亚北部）区域："us-east-1"
- 美国西部（加利福尼亚北部）区域："us-west-1"
- 美国西部（俄勒冈州）区域："us-west-2"

ResourceRecords
要添加的资源记录的列表。每个记录都应该采用适合由 *Type* 属性指定的记录类型的格式。有关不同记录类型及其记录格式的信息，请参阅 附录：Domain Name Format（在 *Route 53 Developer Guide* 中）。

Required: Conditional. 如果设置了 *TTL* 或 *SetIdentifier*，则为必需。此外，如果您设置 *ResourceRecords*，则必须还要设置 *TTL* 或 *SetIdentifier*。

Note

如果要创建别名资源记录集，则应省略 *ResourceRecords*。

Type: A list of strings.

更新要求：无中断 (p. 63)

SetIdentifier
区分具有相同 DNS 名称和类型组合的多个资源记录集的唯一标识符。

Required: Conditional. 如果要创建加权资源记录集，则为必需。必须还要设置 *ResourceRecords*。

有关加权资源记录集的更多信息，请参阅 *Route 53Developer Guide* 中的 Setting Up Weighted Resource Record Sets。

Type: String.

更新要求：无中断 (p. 63)

TTL
资源记录缓存存续时间 (TTL)，以秒计算。如果指定了此属性，则不指定 `AliasTarget` 属性。对于别名目标记录，别名使用来自目标的 TTL 值。

如果指定了 *`TTL`*，则 *`ResourceRecords`* 也是必需的。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

类型
要添加的记录的类型。

Required: Yes.

Type: String.

有效值：A | AAAA | CNAME | MX | NS | PTR | SOA | SPF | SRV | TXT

更新要求：无中断 (p. 63)

权重
*仅限加权资源记录集：*在具有相同的 DNS 名称和类型组合的资源记录集中，将把用于确定当前资源记录集的流量部分的值路由至关联位置。

有关加权资源记录集的更多信息，请参阅 *Route 53Developer Guide* 中的 Setting Up Weighted Resource Record Sets。

Required: Conditional. 如果要创建加权资源记录集，则是必需的。

Type: Number. 加权应为整数值。

更新要求：无中断 (p. 63)

返回值

在将 AWS::Route53::RecordSet 类型指定为 `Ref` 函数的参数时，AWS CloudFormation 会返回该记录集的域名称值。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

示例

Example 将 Amazon Route 53 的记录“A”映射到 EC2 实例的公有 IP 地址

```
"Resources" : {
   "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
         "ImageId" : { "Fn::FindInMap" : [
            "RegionMap", { "Ref" : "AWS::Region" }, "AMI"
         ] }
      }
   },
   "myDNSRecord" : {
      "Type" : "AWS::Route53::RecordSet",
      "Properties" : {
         "HostedZoneName" : {
            "Fn::Join" : [ "", [
               { "Ref" : "HostedZone" }, "."
            ] ]
         },
         "Comment" : "DNS name for my instance.",
         "Name" : {
            "Fn::Join" : [ "", [
               {"Ref" : "Ec2Instance"}, ".",
               {"Ref" : "AWS::Region"}, ".",
               {"Ref" : "HostedZone"} ,"."
            ] ]
         },
         "Type" : "A",
         "TTL" : "900",
         "ResourceRecords" : [
            { "Fn::GetAtt" : [ "Ec2Instance", "PublicIp" ] }
         ]
      }
   }
},
```

有关其他 AWS::Route53::RecordSet 代码段，请参阅 Amazon Route 53 模板代码段 (p. 172)。

AWS::Route53::RecordSetGroup

Abstract

使用 AWS::Route53::RecordSetGroup 资源向现有区域添加记录集。

Note

在使用 AWS CloudFormation 向托管区域添加记录集之前，必需已在 Amazon Route 53 中创建该托管区域。AWS CloudFormation 不会创建新托管区域。

此类型支持更新。有关资源记录集更新的详细信息和约束，请参阅 Changing Your Resource Record Sets。有关更新堆栈的详细信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。

有关每个属性的约束和值的更多信息，请参阅托管区域的 POST CreateHostedZone 以及资源记录集的 POST ChangeResourceRecordSet。

语法

```
{
   "Type" : "AWS::Route53::RecordSetGroup",
   "Properties" : {
      "HostedZoneId (p. 401)" : String,
      "HostedZoneName (p. 401)" : String,
      "RecordSets (p. 401)" : [ RecordSet1, ... ],
      "Comment (p. 401)" : String,
   }
}
```

属性

HostedZoneId
: 托管区的 ID。
必需：条件：您必须指定 *HostedZoneName* 或 *HostedZoneId*，但不能两者都指定。
类型：字符串
更新要求：替换 (p. 63)

HostedZoneName
: 您要向其添加记录集的托管区的域名。
在使用指定 *HostedZoneName* 的 AWS::Route53::RecordSet 创建堆栈时，AWS CloudFormation 会尝试查找名称与 HostedZoneName 匹配的托管区域。如果 AWS CloudFormation 找不到具有匹配域名称的托管区域，或者有一个以上具有指定域名称的托管区域，则 AWS CloudFormation 不会创建堆栈。
如果您有多个具有相同域名称的托管区域，则必须使用 *HostedZoneId* 显式指定托管区域。
必需：条件。您必须指定 *HostedZoneName* 或 *HostedZoneId*，但不能两者都指定。
类型：字符串
更新要求：替换 (p. 63)

RecordSets
: 要添加的资源记录集的列表。
必需：是
类型：AWS::Route53::RecordSet (p. 396)的列表
更新要求：无中断 (p. 63)

评论
: 您要包含的有关托管区的所有评论。
必需：否
类型：字符串
更新要求：无中断 (p. 63)

返回值

当该资源的逻辑 ID 提供给 Ref 内部函数时，它将返回资源名称。 例如：

```
{ "Ref": "MyRecordSetGroup" }
```

对于带有逻辑 ID“MyRecordSetGroup”的资源，`Ref` 将返回 AWS 资源名称。

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

模板示例

有关 AWS::Route53::RecordSetGroup 代码段，请参阅 Amazon Route 53 模板代码段 (p. 172)。

AWS::S3::Bucket

Abstract

使用 AWS::S3::Bucket 资源创建 Amazon S3 存储桶。

AWS::S3::Bucket 类型创建 Amazon S3 存储桶。

您可以为该存储桶设置删除策略，以控制删除堆栈时 AWS CloudFormation 处理该存储桶的方式。对于 Amazon S3 存储桶，您可以选择*保留* 该存储桶或*删除* 该存储桶。有关更多信息，请参阅 DeletionPolicy 属性 (p. 485)。

Important

只能删除空的 Amazon S3 存储桶。如果存储桶中包含内容，则删除操作会失败。

语法

```
{
   "Type" : "AWS::S3::Bucket",
   "Properties" : {
      "AccessControl (p. 402)" : String,
      "BucketName (p. 403)" : String,
      "CorsConfiguration (p. 403)" : CORS Configuration,
      "LifecycleConfiguration (p. 403)" : Lifecycle Configuration,
      "LoggingConfiguration (p. 403)" : Logging Configuration,
      "NotificationConfiguration (p. 403)" : Notification Configuration,
      "   (p. 403)" : [ Resource Tag, ... ],
      "VersioningConfiguration (p. 404)" : Versioning Configuration,
      "WebsiteConfiguration (p. 404)" : Website Configuration Type
   }
}
```

属性

AccessControl

可授予预定义存储桶权限的标准访问控制列表 (ACL)。有关标准 ACL 的更多信息，请参阅 Canned ACLs in the Amazon S3 documentation。

Required: No.

Type: String.

Valid values: `Private` | `PublicRead` | `PublicReadWrite` | `AuthenticatedRead` | `LogDeliveryWrite` | `BucketOwnerRead` | `BucketOwnerFullControl`

更新要求：无中断 (p. 63)

BucketName
存储桶的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作存储桶名称。有关更多信息，请参阅 名称类型 (p. 465)。存储桶名称必须仅包含小写字母、数字、句点 (.) 和短划线 (-)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

Required: No.

Type: String.

更新要求：替换 (p. 63)

CorsConfiguration
定义此存储桶中对象的跨源资源共享的规则。有关更多信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的启用跨源资源共享部分。

Required: No.

类型：Amazon S3 Cors 配置 (p. 474)

更新要求：无中断 (p. 63)

LifecycleConfiguration
定义 Amazon S3 如何在对象的生命周期内管理这些对象的规则。有关更多信息，请参阅 *Amazon Simple Storage Service 开发者指南* 中的对象生命周期管理。

Required: No.

类型：Amazon S3 生命周期配置 (p. 476)

更新要求：无中断 (p. 63)

LoggingConfiguration
定义日志存储位置的设置。

Required: No.

类型：Amazon S3 日志记录配置 (p. 478)

更新要求：无中断 (p. 63)

NotificationConfiguration
定义要将消息发送到的 Amazon SNS 主题以及要报告的事件的配置。

Required: No.

类型：Amazon S3 通知配置 (p. 479)

更新要求：无中断 (p. 63)

标签
此 Amazon S3 存储桶的任意标签（键/值对）组。

Required: No.

Type: AWS CloudFormation 资源标签 (p. 471)

更新要求：无中断 (p. 63)

VersioningConfiguration
: 允许此存储桶中的所有对象存在多个变体。您可能会启用版本控制来防止对象被错误删除或覆盖，或者是将对象存档，以便检索对象的早期版本。

  Required: No.

  Type: Amazon S3 版本控制配置 (p. 480)

  更新要求：无中断 (p. 63)

WebsiteConfiguration
: 用于将存储桶配置为静态网站的信息。有关更多信息，请参阅 Hosting Websites on Amazon S3。

  Required: No.

  Type: 网站配置类型 (p. 480)

  更新要求：无中断 (p. 63)

返回值

Ref

当该资源的逻辑 ID 提供给 `Ref` 内部函数时，它将返回资源名称。

示例：mystack-mybucket-kdwwxmddtr2g

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

DomainName
: 返回指定存储桶的 DNS 名称。

  示例：mystack-mybucket-kdwwxmddtr2g.s3.amazonaws.com

WebsiteURL
: 指定存储桶的 Amazon S3 网站终端节点。

  示例：http://mystack-mybucket-kdwwxmddtr2g.s3-website-us-east-1.amazonaws.com/

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

Example 具有路由规则的静态网站配置

在此示例中，AWS::S3::Bucket 的 `Fn::GetAtt` 值用于提供输出。在发生 HTTP 404 错误时，路由规则将请求重定向到 Amazon EC2 实例，并在重定向中插入对象键前缀 `report-404/`。例如，如果您请求页面 `ExamplePage.html`，而它导致了 HTTP 404 错误，该请求将路由到指定实例上的页面 `report-404/ExamplePage.html`。对于所有其他 HTTP 错误代码，会返回 `error.html`。

```
"Resources" : {
   "S3Bucket" : {
      "Type" : "AWS::S3::Bucket",
      "Properties" : {
         "AccessControl" : "PublicRead",
         "BucketName" : "PublicBucket",
         "WebsiteConfiguration" : {
            "IndexDocument" : "index.html",
            "ErrorDocument" : "error.html",
            "RoutingRules": [
                {
                    "RoutingRuleCondition": {
                        "HttpErrorCodeReturnedEquals": "404",
                        "KeyPrefixEquals": "out1/"
                    },
                    "RedirectRule": {
                       "HostName": "ec2-11-22-333-44.compute-1.amazonaws.com",

                        "ReplaceKeyPrefixWith": "report-404/"
                    }
                }
            ]
         }
      },
      "DeletionPolicy" : "Retain"
   }
},

"Outputs" : {
   "WebsiteURL" : {
      "Value" : { "Fn::GetAtt" : [ "S3Bucket", "WebsiteURL" ] },
      "Description" : "URL for website hosted on S3"
   },
   "S3BucketSecureURL" : {
      "Value" : { "Fn::Join" : [
         "", [ "https://", { "Fn::GetAtt" : [ "S3Bucket", "DomainName" ] } ]
      ] },
      "Description" : "Name of S3 bucket to hold website content"
   }
}
```

Example 启用跨源资源共享

以下示例模板演示具有两个跨源资源共享规则的 Amazon S3 存储桶。

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "AccessControl": "PublicReadWrite",
                "CorsConfiguration": {
                    "CorsRules": [
                        {
                            "AllowedHeaders": [
                                "*"
                            ],
                            "AllowedMethods": [
                                "GET"
                            ],
                            "AllowedOrigins": [
                                "*"
                            ],
                            "ExposedHeaders": [
                                "Date"
                            ],
                            "Id": "myCORSRuleId1",
                            "MaxAge": "3600"
                        },
                        {
                            "AllowedHeaders": [
                                "x-amz-*"
                            ],
                            "AllowedMethods": [
                                "DELETE"
                            ],
                            "AllowedOrigins": [
                                "http://www.example1.com",
                                "http://www.example2.com"
                            ],
                            "ExposedHeaders": [
                                "Connection",
                                "Server",
                                "Date"
                            ],
                            "Id": "myCORSRuleId2",
                            "MaxAge": "1800"
                        }
                    ]
                }
            }
        }
    },
    "Outputs": {
        "BucketName": {
            "Value": {
                "Ref": "S3Bucket"
            },
```

```
            "Description": "Name of the sample Amazon S3 bucket with CORS en
abled."
        }
    }
}
```

Example 管理 Amazon S3 对象的生命周期

以下示例模板演示具有一个生命周期配置规则的 Amazon S3 存储桶。该规则应用于键前缀为 `glacier` 的所有对象。对象在一天之后转移到 Amazon Glacier，在一年之后删除。

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "AccessControl": "PublicReadWrite",
                "LifecycleConfiguration": {
                    "Rules": [
                        {
                            "Id": "GlacierRule"
                            "Prefix": "glacier",
                            "Status": "Enabled",
                            "ExpirationInDays": "365",
                            "Transition": {
                                "TransitionInDays": "1",
                                "StorageClass": "Glacier"
                            }
                        }
                    ]
                }
            }
        },
    "Outputs": {
        "BucketName": {
            "Value": {
                "Ref": "S3Bucket"
            },
           "Description": "Name of the sample Amazon S3 bucket with a lifecycle
 configuration."
        }
    }
}
```

Example 针对特定存储桶的日志访问请求

以下示例模板创建两个 Amazon S3 存储桶。LoggingBucket 存储桶用于存储来自 S3Bucket 存储桶的日志。日志记录存储桶需要日志传输写入权限才能从 S3Bucket 存储桶接收日志。

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "AccessControl": "PublicRead",
                "LoggingConfiguration": {
                    "DestinationBucketName": {"Ref" : "LoggingBucket"},
                    "LogFilePrefix": "testing-logs"
                }
            }
        },
        "LoggingBucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "AccessControl": "LogDeliveryWrite"
            }
        }
    },
    "Outputs": {
        "BucketName": {
            "Value": {
                "Ref": "S3Bucket"
            },
            "Description": "Name of the sample Amazon S3 bucket with a logging
 configuration."
        }
    }
}
```

Example 接收发送到 Amazon SNS 主题的存储桶通知

以下示例模板演示具有一个通知配置的 Amazon S3 存储桶，该配置在 Amazon S3 丢失对象的所有副本时向指定主题发送事件。

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "S3Bucket": {
            "Type": "AWS::S3::Bucket",
            "Properties": {
                "AccessControl": "PublicReadWrite",
                "NotificationConfiguration": {
                    "TopicConfigurations": [
                        {
                            "Topic": "arn:aws:sns:us-east-
1:123456789012:TestTopic",
                            "Event": "s3:ReducedRedundancyLostObject"
                        }
                    ]
                }
            }
        },
    "Outputs": {
        "BucketName": {
            "Value": {
                "Ref": "S3Bucket"
            },
           "Description": "Name of the sample Amazon S3 bucket with a notific
ation configuration."
        }
    }
}
```

有关更多示例，请参阅 Amazon S3 模板代码段 (p. 174)。

另请参阅

- DeletionPolicy 属性 (p. 485)
- *Amazon Simple Storage Service Developer Guide* 中的 Access Control List (ACL) Overview
- *Amazon Simple Storage Service Developer Guide* 中的 Hosting a Static Website on Amazon S3

AWS::S3::BucketPolicy

Abstract

使用 AWS::S3::BucketPolicy 资源将策略应用于 Amazon S3 存储桶。

AWS::S3::BucketPolicy 类型将 Amazon S3 存储桶策略应用于 Amazon S3 存储桶。

AWS::S3::BucketPolicy 片段：声明 Amazon S3 存储桶策略 (p. 150)

语法

```
{
   "Type" : "AWS::S3::BucketPolicy",
   "Properties" : {
      "    (p. 410)" : String,
      "PolicyDocument (p. 410)" : JSON
   }
}
```

属性

存储桶
应用策略的 Amazon S3 存储桶。

Required: Yes.

Type: String.

您无法更新此属性。如果您要对存储桶策略添加或删除存储桶，则必须通过创建新存储桶策略资源并删除旧资源来修改 AWS CloudFormation 模板。然后使用修改的模板更新 AWS CloudFormation 堆栈。

PolicyDocument
策略文档，其中包含可向指定存储桶添加的权限。

Required: Yes.

Type: JSON object.

更新要求：无中断 (p. 63)

示例

Example 允许从特定引用站点进行 GET 请求的存储桶策略

以下示例是一个存储桶策略，该策略附加到 myExampleBucket 存储桶，允许源自 www.example.com 和 example.com 的 GET 请求：

```
"SampleBucketPolicy" : {
  "Type" : "AWS::S3::BucketPolicy",
  "Properties" : {
    "Bucket" : {"Ref" : "myExampleBucket"},
    "PolicyDocument": {
      "Statement":[{
     "Action":["s3:GetObject"],
     "Effect":"Allow",
    "Resource": { "Fn::Join" : ["", ["arn:aws:s3:::", { "Ref" : "myExampleBuck
et" } , "/*" ]]},
     "Principal":"*",
        "Condition":{
          "StringLike":{
            "aws:Referer":[
              "http://www.example.com/*",
              "http://example.com/*"
            ]
          }
        }
      }]
    }
  }
}
```

AWS::SDB::Domain

Abstract

指定 AWS::SDB::Domain 类型时，返回 DomainName 的值。

AWS::SDB::Domain 类型没有任何属性。

此资源不支持更新。

在将 AWS::SDB::Domain 类型指定为 Ref 函数的参数时，AWS CloudFormation 返回 *DomainName* 的值。

AWS::SNS::Topic

Abstract

使用 AWS::SNS::Topic 资源创建 Amazon SNS 主题。

AWS::SNS::Topic 类型创建 Amazon SNS 主题。

语法

```
{
   "DisplayName (p. 412)" : String,
   "    (p. 412)" : [ SNS Subscription, ... ],
   "      (p. 412)" : String
}
```

属性

Important

创建 Amazon SNS 主题之后，您无法使用 AWS CloudFormation 更新其属性。可以使用 AWS Management Console修改 Amazon SNS 主题。

DisplayName
开发人员定义的字符串，可用于识别此 SNS 主题。

Required: No.

Type: String.

更新要求：不支持更新

订阅
此主题的 SNS 订阅（终端节点）。

Required: No.

Type: SNS 订阅 (p. 484)列表

更新要求：不支持更新

主题名称
主题的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作主题名称。有关更多信息，请参阅 名称类型 (p. 465)。

必需：否

类型：名称类型 (p. 465)

更新要求：不支持更新

返回值

Ref

对于 `AWS::SNS::Topic` 资源，`Ref` 内部函数返回主题 ARN，例如：`arn:aws:sns:us-east-1:123456789012:mystack-mytopic-NZJ5JSMVGFIE`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

主题名称
返回 Amazon SNS 主题的名称。

有关使用 `Fn:GetAtt` 的更多信息，请参阅 Fn::GetAtt (p. 502)。

示例

两个 SQS 队列订阅一个 SNS 主题的示例：

```
"MySNSTopic" : {
   "Type" : "AWS::SNS::Topic",
   "Properties" : {
      "Subscription" : [
         { "Endpoint" : { "Fn::GetAtt" : [ "MyQueue1", "Arn" ] }, "Protocol" :
 "sqs" },
         { "Endpoint" : { "Fn::GetAtt" : [ "MyQueue2", "Arn" ] }, "Protocol" :
 "sqs" }
      ],
      "TopicName" : "SampleTopic"
   }
}
```

另请参阅

- *Amazon Simple Notification Service Developer Guide* 中的 Using an AWS CloudFormation Template to Create a Topic that Sends Messages to Amazon SQS Queues

AWS::SNS::TopicPolicy

Abstract

使用 AWS::SNS::TopicPolicy 资源将策略应用于 SNS 主题。

AWS::SNS::TopicPolicy 类型将策略应用于 SNS 主题。

此资源不支持更新。

AWS::SNS::TopicPolicy Snippet:声明 Amazon SNS 主题策略 (p. 151)

属性	类型	必需	备注
PolicyDocument	JSON	是	一份策略文件，包含要添加给指定 SNS 主题的权限。
主题：	SNS 主题 ARN 的阵列	是	您希望为其添加策略的主题的亚马逊资源名称 (ARN)。您可以使用 Ref 函数 (p. 508)来指定 AWS::SNS::Topic (p. 411) 资源。

AWS::SQS::Queue

Abstract

使用 AWS::SQS::Queue 资源创建 Amazon SQS 队列。

AWS::SQS::Queue 类型创建 Amazon SQS 队列。

语法

```
{
   "Type": "AWS::SQS::Queue",
   "Properties": {
      "DelaySeconds (p. 414)": Integer,
      "MaximumMessageSize (p. 414)": Integer,
      "MessageRetentionPeriod (p. 414)": Integer,
      "     (p. 414)": String,
      "ReceiveMessageWaitTimeSeconds (p. 415)": Integer,
      "RedrivePolicy (p. 415)": RedrivePolicy,
      "VisibilityTimeout (p. 415)": Integer
   }
}
```

属性

DelaySeconds
队列中所有消息的传输延迟时间（以秒为单位）。您可以指定 0 到 900（15 分钟）之间的整数值。默认值为 0。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

MaximumMessageSize
一条消息可以包含的字节数的限制，超出此限 Amazon SQS 就会拒绝。您可以指定 1024 字节 (1 KiB) 到 262144 字节 (256 KiB) 之间的整数值。默认值是 262144 (256 KiB)。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

MessageRetentionPeriod
Amazon SQS 保留消息的秒数。您可以指定 60 秒（1 分钟）到 1209600 秒（14 天）之间的整数值。默认值是 345600 秒（4 天）。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

队列名称
队列的名称。如果不指定名称，则 AWS CloudFormation 生成一个唯一物理 ID 并将该 ID 用作队列名称。有关更多信息，请参阅名称类型 (p. 465)。

Important

如果您指定一个名称，您将无法执行需要替换此资源的更新。不过，如果更新操作不需要或者只需要时而中断，则您仍然可以对此资源执行更新。

Required: No.

类型：名称类型 (p. 465)

更新要求：替换 (p. 63)

ReceiveMessageWaitTimeSeconds
指定 `ReceiveMessage` 操作调用等待消息进入队列以包括在响应中的持续时间（以秒为单位），这与尚无可以返回的消息时返回空消息的情况相反。您可以指定 `1` 到 `20` 之间的整数。短轮询用作默认值或在您为此属性指定 `0` 时使用。有关更多新信息，请参阅 Amazon SQS 长轮询。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

RedrivePolicy
指定现有死信队列在源队列（此队列）无法处理消息的次数达到指定次数之后接收消息。

Required: No.

类型：Amazon SQS RedrivePolicy (p. 485)

更新要求：无中断 (p. 63)

VisibilityTimeout
队列交付消息后，可供查看的时间长度。这样可以阻止其他组件收到相同的信息，并为初始组件预留足够的时间来处理并删除队列上的信息。

值必须在 0 到 43 200 秒之间（12 小时）。如果没有指定值，将使用默认值（即 30 秒）。

有关 SQS 队列可见性超时的更多信息，请参阅 *Amazon Simple Queue Service Developer Guide* 中的 Visibility Timeout。

Required: No.

Type: Integer

更新要求：无中断 (p. 63)

返回值

Ref

AWS::SQS::Queue 类型返回队列 URL，例如：
`https://sqs.us-east-1.amazonaws.com/123456789012/aa4-MyQueue-Z5NOSZO2PZE9`.

有关使用 `Ref` 功能的更多信息，请参阅参考 (p. 508)。

Fn::GetAtt

`Fn::GetAtt`返回一个此类型指定属性的值。此部分列出了可用属性和相应的返回值。

Arn
: 返回队列的 Amazon 资源名称 (ARN)。例如：
`arn:aws:sqs:us-east-1:123456789012:mystack-myqueue-15PG5C2FC1CW8`

队列名称
: 返回队列名称。例如：

mystack-myqueue-1VF9BKQH5BJVI

示例

带有 Cloudwatch 警报的 SQS 队列。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template SQS_With_CloudWatch_Alarms:
 Sample template showing how to create an SQS queue with Amazon CloudWatch
alarms on queue depth. **WARNING** This template creates an Amazon SQS queue
and one or more Amazon CloudWatch alarms. You will be billed for the AWS re
sources used if you create a stack from this template.",

  "Parameters" : {
    "AlarmEmail": {
      "Default": "nobody@amazon.com",
      "Description": "Email address to notify if operational problems arise",
      "Type": "String"
    }
  },

  "Resources" : {
    "MyQueue" : {
      "Type" : "AWS::SQS::Queue",
      "Properties" : {
         "QueueName" : "SampleQueue"
      }
    },
    "AlarmTopic": {
      "Type": "AWS::SNS::Topic",
      "Properties": {
        "Subscription": [{
          "Endpoint": { "Ref": "AlarmEmail" },
          "Protocol": "email"
        }]
      }
    },
    "QueueDepthAlarm": {
      "Type": "AWS::CloudWatch::Alarm",
      "Properties": {
        "AlarmDescription": "Alarm if queue depth grows beyond 10 messages",
        "Namespace": "AWS/SQS",
        "MetricName": "ApproximateNumberOfMessagesVisible",
        "Dimensions": [{
          "Name": "QueueName",
          "Value" : { "Fn::GetAtt" : ["MyQueue", "QueueName"] }
        }],
        "Statistic": "Sum",
```

```
        "Period": "300",
        "EvaluationPeriods": "1",
        "Threshold": "10",
        "ComparisonOperator": "GreaterThanThreshold",
        "AlarmActions": [{
          "Ref": "AlarmTopic"
        }],
        "InsufficientDataActions": [{
          "Ref": "AlarmTopic"
        }]
      }
    }
  },
  "Outputs" : {
    "QueueURL" : {
      "Description" : "URL of newly created SQS Queue",
      "Value" : { "Ref" : "MyQueue" }
    },
    "QueueARN" : {
      "Description" : "ARN of newly created SQS Queue",
      "Value" : { "Fn::GetAtt" : ["MyQueue", "Arn"]}
    },
    "QueueName" : {
      "Description" : "Name newly created SQS Queue",
      "Value" : { "Fn::GetAtt" : ["MyQueue", "QueueName"]}
    }
  }
}
```

具有死信队列的 SQS 队列

以下示例创建一个源队列和一个死信队列。因为源队列在其重新驱动策略中指定死信队列，所有源队列依赖于死信队列的创建。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Resources" : {
    "MySourceQueue" : {
      "Type" : "AWS::SQS::Queue",
      "Properties" : {
        "RedrivePolicy": {
         "deadLetterTargetArn" : {"Fn::GetAtt" : [ "MyDeadLetterQueue" , "Arn"
 ]},
          "maxReceiveCount" : 5
        }
      }
    },
    "MyDeadLetterQueue" : {
      "Type" : "AWS::SQS::Queue"
    }
  },

  "Outputs" : {
    "SourceQueueURL" : {
      "Description" : "URL of the source queue",
      "Value" : { "Ref" : "MySourceQueue" }
```

```
    },
    "SourceQueueARN" : {
      "Description" : "ARN of the source queue",
      "Value" : { "Fn::GetAtt" : ["MySourceQueue", "Arn"]}
    },
    "DeadLetterQueueURL" : {
      "Description" : "URL of the dead letter queue",
      "Value" : { "Ref" : "MyDeadLetterQueue" }
    },
    "DeadLetterQueueARN" : {
      "Description" : "ARN of the dead letter queue",
      "Value" : { "Fn::GetAtt" : ["MyDeadLetterQueue", "Arn"]}
    }
  }
}
```

另请参阅

- *Amazon Simple Queue Service API 参考* 中的 CreateQueue
- 什么是 Amazon Simple Queue Service？：*Amazon Simple Queue Service 开发人员指南*

AWS::SQS::QueuePolicy

Abstract

使用 AWS::SQS::QueuePolicy 资源将策略应用于 Amazon SQS 队列。

类型 AWS::SQS::QueuePolicy 适用于对 SQS 队列的策略。

AWS::SQS::QueuePolicy 片段：声明 Amazon SQS 策略 (p. 151)

语法

```
{
   "Type" : "AWS::SQS::QueuePolicy",
   "Properties" : {
      "PolicyDocument (p. 418)" : JSON,
      "   (p. 419)" : [ String, ... ]
   }
}
```

属性

PolicyDocument
策略文档，其中包含可向指定 SNS 队列添加的权限。

Required: Yes.

Type: JSON object.

更新要求：无中断 (p. 63)

队列
: 您要向其添加策略的队列的 URL。您可以使用 Ref 函数 (p. 508)来指定 AWS::SQS::Queue (p. 414) 资源。

Required: Yes.

Type: A list of strings.

更新要求：无中断 (p. 63)

资源属性类型参考

Abstract

列出可以为 AWS CloudFormation 中支持的资源设置的资源特定属性。

此部分详细介绍 AWS CloudFormation 支持的各种资源的特定属性。

Topics

AWS CloudFormation AutoScaling 块存储设备映射属性类型

Abstract

介绍嵌入式 AutoScaling 块储存设备映射属性类型。

AutoScaling Block Device Mapping 类型是 AWS::AutoScaling::LaunchConfiguration (p. 224) 类型的嵌入式属性。

语法

```
{
   "DeviceName (p. 421)" : String,
   "Ebs (p. 421)" : AutoScaling EBS Block Device,
   "NoDevice (p. 421)" : Boolean,
   "VirtualName (p. 421)" : String
}
```

属性

DeviceName
: Amazon EC2 内的设备的名称。

  Required: Yes.

  Type: String.

Ebs
: Amazon Elastic Block Store 卷信息。

  Required: Conditional. 您可以指定 `VirtualName` 或 `Ebs`，但不能两者都指定。

  Type: AutoScaling EBS 块存储设备 (p. 422).

NoDevice
: 隐藏设备映射。如果 `NoDevice` 对于根设备设置为 true，则实例可能无法通过 Amazon EC2 运行状况检查。如果实例无法通过运行状况检查，则 Auto Scaling 会启动替代实例。

  Required: No.

  Type: Boolean.

VirtualName
: 虚拟设备的名称。该名称必须采用 `ephemeral`*X* 形式，其中 *X* 是从零 (0) 开始的数字，例如 `ephemeral0`。

  Required: Conditional. 您可以指定 `VirtualName` 或 `Ebs`，但不能两者都指定。

  Type: String.

AWS CloudFormation Auto Scaling EBS 块储存设备属性类型

Abstract

介绍嵌入式 AutoScaling EBS 块储存设备属性类型。

AutoScaling EBS 块存储设备类型是 AutoScaling 块存储设备映射 (p. 421) 类型的嵌入式属性。

语法

```
{
   "DeleteOnTermination (p. 422)" : Boolean,
   "Iops (p. 422)" : Integer,
   "SnapshotId (p. 422)" : String,
   "VolumeSize (p. 422)" : Integer,
   "VolumeType (p. 422)" : String
}
```

属性

DeleteOnTermination
指示在实例终止时是否删除卷。默认情况下，Auto Scaling 使用 `true`。

Required: No.

Type: Boolean.

Iops
卷支持的每秒 I/O 操作数 (IOPS)。IOPS 与卷大小的最大比率为 30。

Required: No.

类型：整数。

SnapshotId
待使用卷的快照 ID。

Required: Conditional. 如果同时指定 `SnapshotId` 和 `VolumeSize`，则 `VolumeSize` 必须等于或大于快照大小。

Type: String.

VolumeSize
卷大小，以 Gibibyte (GiB) 为单位。它可以是 1 – 1024 范围内的数字。如果卷类型是 EBS 优化，则最小值为 10。有关指定卷类型的更多信息，请参阅 AWS::AutoScaling::LaunchConfiguration (p. 224) 中的 EbsOptimized。

Required: Conditional. 如果同时指定 `SnapshotId` 和 `VolumeSize`，则 `VolumeSize` 必须等于或大于快照大小。

类型：整数。

更新要求：时而中断 (p. 63)

VolumeType
卷类型。默认情况下，Auto Scaling 使用 `standard` 卷类型。

Required: No.

Type: String.

示例

有关 AutoScaling EBS 块储存设备代码段，请参阅 Auto Scaling 启动配置资源 (p. 129)。

Auto Scaling MetricsCollection

Abstract

描述 Auto Scaling 组发送到 Amazon CloudWatch 的用于描述该组（而非其任何实例）的组指标。

`MetricsCollection` 是 AWS::AutoScaling::AutoScalingGroup (p. 219) 资源的属性，描述 Auto Scaling 组发送到 Amazon CloudWatch 的组指标。这些指标描述组而非其任何实例。有关更多信息，请参阅 *Auto Scaling API 参考* 中的 EnableMetricsCollection。

语法

```
{
  "   (p. 423)" : String,
  "   (p. 423)" : [ String, ... ]
}
```

属性

粒度
: Auto Scaling 向 Amazon CloudWatch 发送聚合数据的频率。例如，您可以指定 `1Minute` 以便每分钟向 Amazon CloudWatch 发送聚合数据。

  Required: Yes.

  Type: String.

指标
: 要收集的指标的列表。如果不指定任何指标，则启用所有指标。

  Required: No.

  Type: A list of strings.

Auto Scaling NotificationConfiguration 属性类型

Abstract

使用 NotificationConfiguration 属性可配置 Auto Scaling 组以发送指定事件的通知。

NotificationConfiguration 属性可对 Auto Scaling 组进行配置，使其在指定事件发生时发送通知。

NotificationConfiguration 是 AWS::AutoScaling::AutoScalingGroup (p. 219) 类型的嵌入式属性。

有关 NotificationConfiguration 代码段的信息，请参阅 带通知的 Auto Scaling 组 (p. 131)。

属性	类型	必需	备注
TopicARN	字符串	是	Amazon Simple Notification Service (SNS) 主题的亚马逊资源名称 (ARN)。
NotificationTypes	字符串列表	是	将触发通知的事件类型。将触发通知的事件的列表，可以包括以下任意或所有的事件：autoscaling:EC2_INSTANCE_LAUNCH、autoscaling:EC2_INSTANCE_LAUNCH_ERROR、autoscaling:EC2_INSTANCE_TERMINATE、autoscaling:EC2_INSTANCE_TERMINATE_ERROR 和 autoscaling:TEST_NOTIFICATION。有关事件类型的更多信息，请参阅 DescribeAutoScalingNotificationTypes。

Auto Scaling 标签属性类型

Abstract

介绍嵌入式 Auto Scaling 标签属性，通过这些属性，AWS CloudFormation 可以向所有 Auto Scaling 组和关联实例添加标签。

Auto Scaling 标签属性是 AWS::AutoScaling::AutoScalingGroup (p. 219) 类型的嵌入式属性。有关标签的更多信息，请参阅 *Auto Scaling 开发人员指南* 中的 Tagging Auto Scaling Groups and Amazon EC2 Instances。

AWS CloudFormation 会为所有 Auto Scaling 组及其相关实例添加以下标签：

- aws:cloudformation:stack-name
- aws:cloudformation:stack-id
- aws:cloudformation:logical-id

语法

```
{
   "Key (p. 424)" : String,
   "Value (p. 424)" : String,
   "PropagateAtLaunch (p. 425)" : Boolean
}
```

属性

键

标签的密钥名称。

必需：是

类型：字符串

值

标签的值。

必需：是

类型：字符串

PropagateAtLaunch
: 如果您希望 AWS CloudFormation 将标签复制到作为 Auto Scaling 组的一部分启动的 EC2 实例，请将此参数设置为 `true`。如果您希望标签仅与 Auto Scaling 组连接，而不复制到任何作为 Auto Scaling 组的一部分启动的实例，请将该参数设置为 `false`。

必需：是

类型：布尔值

示例

下面的示例模板代码段将创建两个 Auto Scaling 标签。第一个标签 `MyTag1` 将与名为 `WebServerGroup` 的 Auto Scaling 组连接，并且会复制到作为 Auto Scaling 组的一部分启动的任何 EC2 实例。第二个标签 `MyTag2` 将仅与名为 `WebServerGroup` 的 Auto Scaling 组连接。

```
"WebServerGroup" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "1",
      "MaxSize" : "2",
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ],
      "Tags" : [ {
         "Key" : "MyTag1",
         "Value" : "Hello World 1",
         "PropagateAtLaunch" : "true"
      }, {
         "Key" : "MyTag2",
         "Value" : "Hello World 2",
         "PropagateAtLaunch" : "false"
      } ]
   }
}
```

CloudFormation 堆栈参数属性类型

Parameters 类型是 AWS::CloudFormation::Stack (p. 250) 类型的一个嵌入式属性。

Parameters 类型包含一组值对，这些值对表示将传递到用于创建 AWS::CloudFormation::Stack 资源的模板的参数。每个参数都具有对应于该嵌入式模板中定义的参数的名称，以及表示要为该参数设置的值。例如，示例模板 EC2ChooseAMI.template 包含以下参数部分：

```
"Parameters" : {
   "InstanceType" : {
      "Type" : "String",
      "Default" : "m1.small",
      "Description" : "EC2 instance type, e.g. m1.small, m1.large, etc."
   },
   "WebServerPort" : {
      "Type" : "String",
      "Default" : "80",
      "Description" : "TCP/IP port of the web server"
   },
```

```
   "KeyName" : {
      "Type" : "String",
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to
 the web server"
   }
}
```

您可以使用以下模板来嵌入使用 EC2ChooseAMI.template 的堆栈 (myStackWithParams)，并使用 AWS::CloudFormation::Stack 资源中的 Parameters 属性来指定 InstanceType 和 KeyName：

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myStackWithParams" : {
         "Type" : "AWS::CloudFormation::Stack",
         "Properties" : {
           "TemplateURL" : "https://s3.amazonaws.com/cloudformation-templates-
us-east-1/EC2ChooseAMI.template",
            "Parameters" : {
               "InstanceType" : "t1.micro",
               "KeyName" : "mykey"
            }
         }
      }
   }
}
```

CloudFront CacheBehavior

Abstract

使用 CacheBehavior 属性描述请求的 URL 与模式匹配时的 Amazon CloudFront 缓存行为。

CacheBehavior 是 DistributionConfig (p. 430) 属性的属性，用于描述请求的 URL 与模式匹配时的 Amazon CloudFront 缓存行为。

语法

```
{
   "TargetOriginId (p. 426)" : String,
   "ForwardedValues (p. 427)" : ForwardedValues,
   "TrustedSigners (p. 427)" : [ String, ... ],
   "ViewerProtocolPolicy (p. 427)" : String,
   "MinTTL (p. 427)" : String
   "PathPattern (p. 427)" : String
}
```

属性

TargetOriginId
在请求与 PathPattern 匹配时，您希望 CloudFront 将请求发送到的目标源的 ID 值。

Required: Yes.

Type: String.

ForwardedValues
指定 CloudFront 处理查询字符串的方式。

Required: Yes.

类型：ForwardedValues (p. 427) 类型

TrustedSigners
您希望允许为私有内容创建签名 URL 的 AWS 账户列表。

Required: No.

Type: A list of strings.

ViewerProtocolPolicy
指定当请求与 PathPattern 匹配时，用户访问 TargetOriginId 指定的源中的文件时所使用的协议。

Required: Yes.

Type: String.

MinTTL
您希望在 CloudFront 查询源以了解对象是否已更新之前，对象保留在缓存中的最短时长。

Required: No.

Type: String.

PathPattern
您希望此缓存行为应用的模式（例如，"images/*.jpg"）。

CloudFront 接收到最终用户请求时，系统会按照缓存行为在堆栈规格列表中等待分配的顺序，将请求路径与路径模式进行比较。

Required: Yes.

Type: String.

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudFront ForwardedValues

Abstract

使用 ForwardedValues 属性指示 Amazon CloudFront 如何为缓存行为处理查询字符串。

`ForwardedValues` 是 DefaultCacheBehavior (p. 429) 和 CacheBehavior (p. 426) 属性的属性，用于描述 Amazon CloudFront 是否转发查询字符串。

语法

```
{
```

```
   "QueryString (p. 428)" : Boolean
}
```

属性

QueryString
指明是否希望 CloudFront 将查询字符串转发到与此缓存行为关联的源。如果是，请指定 "true"，否则指定 "false"。

Required: Yes.

Type: Boolean.

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudFront CustomOrigin

Abstract

使用 CustomOrigin 属性描述用作 Amazon CloudFront 的自定义源的 HTTP 服务器。

CustomOrigin CloudFront Origin (p. 432) 属性的属性，用于描述 HTTP 服务器。

语法

```
{
   "HTTPPort (p. 428)" : String
   "HTTPSPort (p. 428)" : String
   "OriginProtocolPolicy (p. 429)" : String
}
```

属性

Note

有关每个属性的限制条件和值的更多信息，请参阅 *Amazon CloudFront API 参考* 中的 DistributionConfig Complex Type。

HTTPPort
自定义源侦听的 HTTP 端口。

Required: No.

Type: String.

HTTPSPort
自定义源侦听的 HTTPS 端口。

Required: No.

Type: String.

OriginProtocolPolicy
要应用至源的源协议策略。

Required: Yes.

Type: String.

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudFront DefaultCacheBehavior

Abstract

使用 DefaultCacheBehavior 属性描述 Amazon CloudFront 分配的默认缓存行为。

`DefaultCacheBehavior` 是 DistributionConfig (p. 430) 属性的属性，用于描述 Amazon CloudFront 分配的默认缓存行为。

语法

```
{
   "TargetOriginId (p. 429)" : String,
   "ForwardedValues (p. 429)" : ForwardedValues,
   "TrustedSigners (p. 429)" : [ String, ... ],
   "ViewerProtocolPolicy (p. 429)" : String,
   "MinTTL (p. 430)" : String
}
```

属性

TargetOriginId
您希望 CloudFront 在默认缓存行为适用于请求时将请求发送到的源的 ID 值。

Required: Yes.

Type: String.

ForwardedValues
指定 CloudFront 如何处理查询字符串。

Required: Yes.

类型：ForwardedValues (p. 427) 类型

TrustedSigners
您希望允许为私有内容创建签名 URL 的 AWS 账户列表。

Required: No.

Type: A list of strings.

ViewerProtocolPolicy
指定当默认缓存行为适用于请求时，用户访问 TargetOriginId 指定的源中的文件所使用的协议。

Required: Yes.

Type: String.

MinTTL
您希望在 CloudFront 查询源以了解对象是否已更新之前，对象保留在缓存中的最短时长。

Required: No.

Type: String.

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudFront DistributionConfig

Abstract

使用 DistributionConfig 属性指示 Amazon CloudFront 当用户通过网站或应用程序请求文件时从哪些原始服务器获取文件。

DistributionConfig 是 AWS::CloudFront::Distribution (p. 256) 属性的属性，用于描述在用户通过网站或应用程序请求文件时从哪些原始服务器获取文件。

语法

```
{
   "Aliases (p. 430)" : [ String, ... ],
   "CacheBehaviors (p. 430)" : [ CacheBehavior, ... ],
   "Comment (p. 430)" : String,
   "DefaultCacheBehavior (p. 431)" : DefaultCacheBehavior,
   "DefaultRootObject (p. 431)" : String,
   "Enabled (p. 431)" : Boolean,
   "Logging (p. 431)" : Logging,
   "Origins (p. 431)" : [ Origin, ... ]
}
```

属性

Aliases
分配的别名记录（替代域名）（如果有）。

Required: No.

Type: A list of strings.

CacheBehaviors
分配的 CacheBehavior 类型列表。

Required: No.

类型：CacheBehavior (p. 426) 列表

评论
您希望包含的任何关于分配的评论。

Required: No.

Type: String.

DefaultCacheBehavior
在以下条件下触发的默认缓存行为：未指定 CacheBehavior 元素；或者文件未与 CacheBehavior 元素中的任何 PathPattern 值匹配。

Required: Yes.

类型：DefaultCacheBehavior 类型 (p. 429)

DefaultRootObject
您希望在请求分配的根 URL 时（例如，"http://example.com/"），CloudFront 从您的源中请求的对象（例如，"index.html"）。

Note

指定一个默认根对象，以避免公开分配的内容。

Required: No.

Type: String.

已启用
控制是否已启用分配以接受对内容的最终用户请求。

Required: Yes.

Type: Boolean.

日志记录
控制是否为分配编写访问日志。要启用访问日志，请包括此属性。

Required: No.

类型：Logging (p. 431) 类型

Origins
此 CloudFront 分配的源列表。对于每个源，您可以指定它是 *S3* 还是*自定义*源。

Required: Yes.

类型：源 (p. 432)列表。

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

另请参阅

- *Amazon CloudFront API 参考* 中的 DistributionConfig Complex Type。

CloudFront Logging

Abstract

使用 Logging 属性针对 Amazon CloudFront 分配启用有关用户请求的访问日志记录。

`Logging` 是 DistributionConfig (p. 430) 属性的属性，使 CloudFront 可将各个分配的访问日志传输给 Amazon S3 存储桶。

语法

```
{
   "Bucket (p. 432)" : String,
   "Prefix (p. 432)" : String
}
```

属性

Note

有关每个属性的限制条件和值的更多信息，请参阅 DistributionConfig Complex Type。

存储桶
用于存储访问日志的 Amazon S3 存储桶。

Required: Yes.

Type: String.

前缀
一个可选字符串，定义后将用作此分配的访问日志的文件名前缀。

Required: No.

Type: String.

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudFront Origin

Abstract

使用 Origin 属性描述 Amazon CloudFront 分配源。

`Origin` 是 DistributionConfig (p. 430) 属性的属性，用于描述 Amazon CloudFront 分配源。

语法

```
{
   "DomainName (p. 433)" : String,
   "Id (p. 433)" : String,
   "S3OriginConfig (p. 433)" : S3 Origin,
   "CustomOriginConfig (p. 433)" : CustomOrigin,
}
```

属性

DomainName
您希望 CloudFront 从中获取此源的对象的 Amazon S3 存储桶或 HTTP 服务器的 DNS 名称。

Required: Yes.

Type: String.

Id
源的标识符。Id 值在分配范围内必须具有唯一性。

Required: Yes.

Type: String.

用于指定 &S3; 源的源信息。
用于指定 Amazon S3 源的源信息。

Required: Conditional..您不能在同一分配中同时使用 S3Origin 和 CustomOrigin，但*必须*指定其中一项。

类型：S3Origin (p. 433) 类型

CustomOriginConfig
用于指定自定义源的源信息。

Required: Conditional..您不能在同一分配中同时使用 CustomOrigin 和 S3 源，但*必须*指定其中一项。

类型：CustomOrigin (p. 428) 类型

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudFront S3Origin

Abstract

使用 S3Origin 属性描述 Amazon S3 源。

S3Origin 是 Origin (p. 432) 属性的属性，用于描述要与源关联的 Amazon S3 源。

语法

```
{
   "OriginAccessIdentity (p. 434)" : String
}
```

属性

Note

有关 OriginAccessIdentity 的限制条件和值的更多信息，请参阅 *Auto Scaling API 参考* 中的 DistributionConfig Complex Type。

OriginAccessIdentity
要与源关联的 CloudFront 源访问标识。该参数用于对源进行配置，使最终用户只能通过 CloudFront 访问 Amazon S3 存储桶中的对象。

Required: No.

Type: String.

模板示例

要查看 AWS::CloudFront::Distribution 代码段，请参阅 Amazon CloudFront 模板代码段 (p. 169)。

CloudWatch 指标维属性类型

Abstract

使用嵌入式指标维度属性为与 CloudWatch 指标关联的给定指标指定最多 10 个维度。

指标维度是 AWS::CloudWatch::Alarm (p. 257) 类型的嵌入式属性。维度是可与 CloudWatch 指标关联的任意名称/值对。您最多可以为给定指标指定 10 个维度。

语法

```
{
   "Name" : String,
   "Value" : String
}
```

属性

名称
维度的名称，长度在 1–255 个字符之间。

Required: Yes.

Type: String.

值
表示维度衡量标准的值，长度在 1–255 个字符之间。

Required: Yes.

Type: String.

示例

两个 CloudWatch 警报，具有 Ref 函数提供的维度值

Ref (p. 508) 和 Fn::GetAtt (p. 502) 内部函数通常用于为 CloudWatch 指标维度提供值。下面是 `Ref` 函数的使用示例。

```
"CPUAlarmHigh": {
```

```
   "Type": "AWS::CloudWatch::Alarm",
   "Properties": {
      "AlarmDescription": "Scale-up if CPU is greater than 90% for 10 minutes",

      "MetricName": "CPUUtilization",
      "Namespace": "AWS/EC2",
      "Statistic": "Average",
      "Period": "300",
      "EvaluationPeriods": "2",
      "Threshold": "90",
      "AlarmActions": [ { "Ref": "WebServerScaleUpPolicy" } ],
      "Dimensions": [
         {
            "Name": "AutoScalingGroupName",
            "Value": { "Ref": "WebServerGroup" }
         }
      ],
      "ComparisonOperator": "GreaterThanThreshold"
   }
},
"CPUAlarmLow": {
   "Type": "AWS::CloudWatch::Alarm",
   "Properties": {
      "AlarmDescription": "Scale-down if CPU is less than 70% for 10 minutes",

      "MetricName": "CPUUtilization",
      "Namespace": "AWS/EC2",
      "Statistic": "Average",
      "Period": "300",
      "EvaluationPeriods": "2",
      "Threshold": "70",
      "AlarmActions": [ { "Ref": "WebServerScaleDownPolicy" } ],
      "Dimensions": [
         {
            "Name": "AutoScalingGroupName",
            "Value": { "Ref": "WebServerGroup" }
         }
      ],
      "ComparisonOperator": "LessThanThreshold"
   }
}
```

另请参阅

- *Amazon CloudWatch API Reference* 中的 Dimension
- *Amazon CloudWatch Developer Guide*中的 Amazon CloudWatch Metrics, Namespaces, and Dimensions Reference

DynamoDB 属性定义

Abstract

为 `AWS::DynamoDB::Table` 资源提供属性定义列表。每个元素由 `AttributeName` 和 `AttributeType` 组成。

AWS::DynamoDB::Table (p. 260) 资源的属性定义列表。每个元素由 `AttributeName` 和 `AttributeType` 组成。

语法

```
{
  "AttributeName (p. 436)" : String,
  "AttributeType (p. 436)" : String
}
```

属性

AttributeName
: 属性的名称。属性名称长度在 1–255 个字符之间，没有字符限制。

Required: Yes.

Type: String.

AttributeType
: 属性的数据类型。可以为字符串数据指定 `S`，为数值数据指定 `N`，或为二进制数据指定 `B`。

Required: Yes.

Type: String.

示例

有关示例，请参阅 AWS::DynamoDB::Table (p. 260)。

DynamoDB 全局二级索引

Abstract

说明 `AWS::DynamoDB::Table` 资源的全局二级索引。

说明 AWS::DynamoDB::Table (p. 260) 资源的全局二级索引。

语法

```
{
  "IndexName (p. 436)" : String,
  "KeySchema (p. 437)" : [ KeySchema, ... ],
  "Projection (p. 437)" : { Projection },
  "ProvisionedThroughput (p. 437)" : { ProvisionedThroughput }
}
```

属性

IndexName
: 全局二级索引的名称。索引名称长度在 3 – 255 个字符之间，没有字符限制。

Required: Yes.

Type: String.

KeySchema
全局二级索引的完整索引键架构，包括一个或多个属性名称和键类型对。

Required: Yes.

类型：DynamoDB 键架构 (p. 437)

Projection
从源表复制（投影）到索引中的属性。这些属性是主键属性和索引键属性（这些属性会自动投影）之外的属性。

Required: Yes.

类型：DynamoDB 投影对象 (p. 439)

ProvisionedThroughput
索引的预配置吞吐量设置。

Required: Yes.

类型：DynamoDB 预置吞吐量 (p. 439)

示例

有关已声明全局二级索引的示例，请参阅 AWS::DynamoDB::Table (p. 260)。

DynamoDB 键架构

Abstract

说明 `AWS::DynamoDB::Table` 资源的主键或索引的键架构。每个元素由 `AttributeName` 和 `KeyType` 组成。

说明 AWS::DynamoDB::Table (p. 260) 资源的主键或索引的键架构。每个元素由 `AttributeName` 和 `KeyType` 组成。

对于仅包含哈希属性的 Amazon DynamoDB 表的主键，请指定一个 `KeyType` 为 `HASH` 的元素。对于仅包含哈希和范围属性的 Amazon DynamoDB 表的主键，请指定两个元素：一个元素的 `KeyType` 为 `HASH`，另一个元素的 `KeyType` 为 `RANGE`。

有关 Amazon DynamoDB 主键的完整讨论，请参阅 *Amazon DynamoDB 开发者指南* 中的主键。

语法

```
{
  "AttributeName (p. 437)" : String,
  "KeyType (p. 438)" : "HASH"
},
{
  "AttributeName (p. 437)" : String,
  "KeyType (p. 438)" : "RANGE"
}
```

属性

AttributeName
用作表主键的属性名称。主密钥元素的名称长度在 1–255 个字符之间，没有字符限制。

Required: Yes.

Type: String.

KeyType
表示属性数据，由数据类型和属性值本身组成。可以指定 HASH 或 RANGE。

Required: Yes.

Type: String.

示例

有关已声明键架构的示例，请参阅 AWS::DynamoDB::Table (p. 260)。

DynamoDB 本地二级索引

Abstract

描述 AWS::DynamoDB::Table 资源的本地二级索引。

说明 AWS::DynamoDB::Table (p. 260) 资源的本地二级索引。每个索引的范围都限定到给定哈希键值。具有一个或多个本地二级索引的表都受制于项目集合大小限制，其中给定项目集合中的数据量不能超过 10 GB。

语法

```
{
  "IndexName (p. 438)" : String,
  "KeySchema (p. 438)" : [ KeySchema, ...],
  "Projection (p. 438)" : { Projection }
}
```

属性

IndexName
本地二级索引的名称。索引名称长度在 3 – 255 个字符之间，没有字符限制。

Required: Yes.

Type: String.

KeySchema
本地二级索引的完整索引键架构，包括一个或多个属性名称和键类型对。对于本地二级索引，哈希键必须与源表的哈希键相同。

Required: Yes.

类型：DynamoDB 键架构 (p. 437)

Projection
从源表复制（投影）到索引中的属性。这些属性是s主键属性和索引键属性（这些属性会自动投影）之外的属性。

Required: Yes.

类型：DynamoDB 投影对象 (p. 439)

示例

有关已声明本地二级索引的示例，请参阅 AWS::DynamoDB::Table (p. 260)。

DynamoDB 投影对象

Abstract

说明从源表复制（投影）到索引中的属性。

从源表复制（投影）到索引中的属性。这些属性是主键属性和索引键属性（这些属性会自动投影）之外的属性。

语法

```
{
  "NonKeyAttributes (p. 439)" : [ String, ... ],
  "ProjectionType (p. 439)" : String
}
```

属性

NonKeyAttributes
: 投影到索引中的非键属性名称。

  对于本地二级索引，所有本地二级索引中 NonKeyAttributes 的数量总和不能超过 20。如果将同一属性名称投影到两个不同的索引中，在确定总量时计为两个不同的属性。

  Required: No.

  类型：字符串列表

ProjectionType
: 投影到索引中的属性集：

  KEYS_ONLY
  : 只有索引和主键才投影到索引中。

  INCLUDE
  : 只有指定表属性才投影到索引中。投影属性的列表处于 NonKeyAttributes 中。

  全部
  : 所有表属性都投影到索引中。

  必需：否

  类型：字符串

示例

有关示例，请参阅 AWS::DynamoDB::Table (p. 260)。

DynamoDB 预置吞吐量

Abstract

说明资源的一组预配置吞吐量值，用于分配足够的资源，以提供所请求的吞吐量。

说明 AWS::DynamoDB::Table (p. 260) 资源的一组预置吞吐量值。Amazon DynamoDB 使用这些容量单位来分配足够的资源，以提供所请求的吞吐量。

有关 Amazon DynamoDB 预置吞吐量值的全面讨论，请参阅 *Amazon DynamoDB Developer Guide* 中的 Specifying Read and Write Requirements。

语法

```
{
   "ReadCapacityUnits (p. 440)" : Number,
   "WriteCapacityUnits (p. 440)" : Number
}
```

参数

ReadCapacityUnits
为指定表设置在 Amazon DynamoDB 均衡负载之前，项目（最大为 1KB）每秒需要达到的最小一致性读取数。

Required: Yes.

Type: Number.

WriteCapacityUnits
为指定表设置在 Amazon DynamoDB 均衡负载之前，项目（最大为 1KB）每秒需要达到的最小一致性写入数。

Required: Yes.

Type: Number.

Note

有关 Amazon DynamoDB 中的预置吞吐量值限制的详细信息，请参阅 *Amazon DynamoDB Developer Guide* 中的 Limits in Amazon DynamoDB。

示例

有关预置吞吐量值的示例，请参阅 AWS::DynamoDB::Table (p. 260)。

Amazon EC2 块储存设备映射属性

Abstract

描述 Amazon EC2 实例的 Amazon EC2 块储存设备映射。

Amazon EC2 块储存设备映射属性是 AWS::EC2::Instance (p. 272) 资源的嵌入式属性。有关 Auto Scaling 启动配置的块储存设备映射，请参阅 AutoScaling 块存储设备映射 (p. 421)。

语法

```
{
   "DeviceName (p. 441)" : String,
```

```
   "Ebs (p. 441)" : EC2 EBS Block Device,
   "NoDevice (p. 441)" : {},
   "VirtualName (p. 441)" : String
}
```

属性

DeviceName
: Amazon EC2 内的设备的名称。

  Required: Yes.

  Type: String.

Ebs
: *Required*: Conditional. 您可以指定 `VirtualName` 或 `Ebs`，但不能两者都指定。

  Type: Amazon Elastic Block Store 块储存设备属性 (p. 442).

NoDevice
: 该属性可用于取消映射已定义的设备。

  Required: No.

  Type: 空映射：{}.

VirtualName
: 虚拟设备的名称。该名称必须采用 `ephemeral`*X* 形式，其中 *X* 是从零 (0) 开始的数字，例如 `ephemeral0`。

  Required: Conditional. 您可以指定 `VirtualName` 或 `Ebs`，但不能两者都指定。

  Type: String.

示例

具有两个 EBS 卷的块存储设备映射

本示例将受 EBS 支持的根设备 (/dev/sda1) 大小设置为 50 GiB，而另一个受 EBS 支持的设备则映射至大小为 100 GiB 的 /dev/sdm。

```
"BlockDeviceMappings" : [
   {
      "DeviceName" : "/dev/sda1",
      "Ebs" : { "VolumeSize" : "50" }
   },
   {
      "DeviceName" : "/dev/sdm",
      "Ebs" : { "VolumeSize" : "100" }
   }
]
```

具有暂存驱动器的块存储设备映射

本示例将暂存驱动器映射至 /dev/sdc。

```
"BlockDeviceMappings" : [
   {
      "DeviceName"  : "/dev/sdc",
      "VirtualName" : "ephemeral0"
   }
]
```

取消映射 AMI 定义的设备

要取消在 AMI 中定义的设备的映射，请将 NoDevice 属性设置为空映射，如下所示：

```
{
   "DeviceName":"/dev/sde",
   "NoDevice": {}
}
```

另请参阅

- *Amazon Elastic Compute Cloud User Guide* 中的 Amazon EC2 Instance Store

Amazon Elastic Block Store 块储存设备属性

Abstract

描述 Amazon EC2 块储存设备映射属性的 Amazon Elastic Block Store 嵌入式属性。

Amazon Elastic Block Store 块储存设备类型是 Amazon EC2 块储存设备映射属性 (p. 440) 属性的嵌入式属性。

语法

```
{
   "DeleteOnTermination (p. 442)" : Boolean,
   "Iops (p. 442)" : Number,
   "SnapshotId (p. 443)" : String,
   "VolumeSize (p. 443)" : String,
   "VolumeType (p. 443)" : String
}
```

属性

DeleteOnTermination
: 确定是否在实例终止时删除卷。默认值为 `true`。

  Required: No.

  Type: Boolean.

Iops
: 卷支持的每秒 I/O 操作数 (IOPS)。该参数可为 100–2000 之间的任意整数。

Required: Conditional. 如果卷类型 (p. 443)为 "io1"，则为必需属性；不与标准卷配合使用。

Type: Number.

SnapshotId
用于创建块存储设备的卷的快照 ID。

Required: Conditional. 如果同时指定 `SnapshotId` 和 `VolumeSize`，则 `VolumeSize` 必须等于或大于快照大小。

Type: String.

VolumeSize
卷大小，以 Gibibyte (GiB) 为单位。该参数可为 1 – 1 024 之间的任意数字。如果卷类型为 `io1`，则最小值为 10。

Required: Conditional. 如果同时指定 `SnapshotId` 和 `VolumeSize`，则 `VolumeSize` 必须等于或大于快照大小。

Type: String.

更新要求：时而中断 (p. 63)

VolumeType
卷类型。有效值为 `standard`（默认值）和 `io1`。如果您将类型设置为 `io1`，则还必须设置 Iops 属性。

Required: No.

Type: String.

示例

```
{
   "DeviceName":"/dev/sdc",
   "Ebs":{
      "SnapshotId":"snap-xxxxxx",
      "VolumeSize":"50",
      "VolumeType":"io1",
      "Iops":"1000",
      "DeleteOnTermination":"false"
   }
}
```

另请参阅

- *Amazon Elastic Compute Cloud API Reference* 中的 CreateVolume

EC2 ICMP 属性类型

Abstract

指定可用于 EC2 ICMP 属性类型的属性。

EC2 ICMP 属性是 AWS::EC2::NetworkAclEntry (p. 281) 类型的嵌入式属性。

以下属性适用于 EC2 ICMP 类型。

属性	类型	必需	备注
代码	整数	条件	Internet 控制消息协议 (ICMP) 代码。您可以使用 -1 为给定 ICMP 类型指定的所有 ICMP 代码。 条件：如果指定 1 (ICMP) 作为 CreateNetworkAclEntry 协议参数，则为必需属性。
类型	整数	条件	Internet 控制消息协议 (ICMP) 类型。您可以使用 -1 指定所有 ICMP 类型。 条件：如果指定 1 (ICMP) 作为 CreateNetworkAclEntry 协议参数，则为必需属性。

EC2 MountPoint 属性类型

Abstract

描述 Amazon EC2 的 MountPoint 嵌入式属性的语法和详细信息。

EC2 MountPoint 属性是 AWS::EC2::Instance (p. 272) 类型的嵌入式属性。

语法

```
{
   "Device (p. 444)" : String,
   "VolumeId (p. 444)" : String
}
```

属性

设备
: 设备如何对实例（如 /dev/sdh 或 xvdh）开放。

  Required: Yes.

  Type: String.

卷 ID
: Amazon EBS 卷的 ID。卷和实例必须位于同一可用区内，而且实例必须处于运行状态。

  Required: Yes.

  Type: String.

示例

该安装点（在 EC2 实例的 *Volumes* 属性中指定）引用一个已命名的 EBS 卷："NewVolume"。

```
"Ec2Instance" : {
   "Type" : "AWS::EC2::Instance",
   "Properties" : {
```

```
      "AvailabilityZone" : {
         "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "TestAz"
]
      },
      "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : {
         "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]
      },
      "Volumes" : [
         { "VolumeId" : { "Ref" : "NewVolume" }, "Device" : "/dev/sdk" }
      ]
   }
},
"NewVolume" : {
   "Type" : "AWS::EC2::Volume",
   "Properties" : {
      "Size" : "100",
      "AvailabilityZone" : {
         "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "TestAz"
]
      }
   }
}
```

另请参阅

- AWS::EC2::Instance (p. 272)
- AWS::EC2::Volume (p. 305)

EC2 NetworkInterface 嵌入式属性类型

Abstract

描述 Amazon EC2 的 NetworkInterface 嵌入式属性的语法和详细信息。

EC2 Network Interface 类型是 AWS::EC2::Instance (p. 272) 类型的嵌入式属性。它指定将关联的网络接口。

语法

```
{
   "AssociatePublicIpAddress (p. 446)" : Boolean,
   "DeleteOnTermination (p. 446)" : Boolean,
   "Description (p. 446)" : String,
   "DeviceIndex (p. 446)" : String,
   "GroupSet (p. 446)" : [ String, ... ],
   "NetworkInterfaceId (p. 446)" : String,
   "PrivateIpAddress (p. 446)" : String,
   "PrivateIpAddresses (p. 446)" : [ PrivateIpAddressSpecification, ... ],
   "SecondaryPrivateIpAddressCount (p. 447)" : Integer,
   "SubnetId (p. 447)" : String
}
```

属性

AssociatePublicIpAddress
指示网络接口是否接收公有 IP 地址。您只能将公有 IP 地址与设备索引为 `eth0` 的网络接口关联。有关更多信息，请参阅 Amazon EC2 Instance IP Addressing。

Required: No.

Type: Boolean..

DeleteOnTermination
是否要在实例终止时删除网络接口。

Required: No.

Type: Boolean..

说明
有关此网络接口的说明。

Required: No.

Type: String.

DeviceIndex
网络接口在关联顺序中的位置。

Required: Yes.

Type: String.

GroupSet
此网络接口相关的安全组 ID 列表。

Required: No.

类型：字符串列表。

NetworkInterfaceId
网络接口 ID。

Required: No.

Type: String.

PrivateIpAddress
将单个私有 IP 地址分配给网络接口，该地址用作主要私有 IP 地址。如果您想指定多个私有 IP 地址，请使用 `PrivateIpAddresses` 属性。

Required: No.

Type: String.

PrivateIpAddresses
将一组私有 IP 地址分配给网络接口。您可以通过在 `PrivateIpAddressSpecification` 属性中将 `Primary` 属性的值设置为 `true` 来指定主要私有 IP 地址。如果您希望 Amazon EC2 自动分配私有 IP 地址，请使用 `SecondaryPrivateIpCount` 属性，并且不指定此属性。

有关私有 IP 地址的最大数量的信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的 Private IP Addresses Per ENI Per Instance Type。

Required: No.

类型：PrivateIpAddressSpecification (p. 449) 的列表

SecondaryPrivateIpAddressCount
Amazon EC2 自动分配到网络接口的辅助私有 IP 地址的数量。Amazon EC2 将 `PrivateIpAddress` 属性的值用作主要私有 IP 地址。如果不指定该属性，则 Amazon EC2 将自动分配主要和辅助私有 IP 地址。

如果您想指定自己的私有 IP 地址列表，请使用 `PrivateIpAddresses` 属性，并且不指定此属性。

有关私有 IP 地址的最大数量的信息，请参阅 *Amazon Elastic Compute Cloud 用户指南* 中的 Private IP Addresses Per ENI Per Instance Type。

Required: No.

类型：整数。

SubnetId
要与网络接口关联的子网的 ID。

必需：条件。如果不指定 `NetworkInterfaceId` 属性，则必须指定此属性。

Type: String.

EC2 网络接口关联

Abstract

描述 Elastic Network Interface 属性类型的网络接口关联。

描述 Elastic Network Interface (ENI) 的网络接口关联。AWS::EC2::NetworkInterface (p. 283) 在其 Association 属性中使用此类型的对象。

语法

```
{
   "AttachmentID" : String,
   "InstanceID" : String,
   "PublicIp" : String,
   "IpOwnerId" : String
}
```

属性

AttachmentID
网络接口连接的 ID。
必需：是
类型：字符串

实例 ID
与此网络接口关联的实例的 ID。
必需：是
类型：字符串

PublicIp
与网络接口关联的弹性 IP 地址的地址。
必需：是
类型：字符串

IpOwnerId
弹性 IP 地址所有者的 ID。
必需：是
类型：字符串

EC2 网络接口连接

Abstract

描述 Elastic Network Interface 属性类型的网络接口连接。

描述 Elastic Network Interface (ENI) 的网络接口连接。AWS::EC2::NetworkInterface (p. 283) 在其 Attachment 属性中使用此类型的对象。

语法

```
{
   "AttachmentID" : String,
   "InstanceID" : String
}
```

属性

AttachmentID
网络接口连接的 ID。
必需：是
类型：字符串

实例 ID
与此网络接口关联的实例的 ID。
必需：是
类型：字符串

EC2 网络接口组项目

Abstract

使用 GroupSet 属性在组集中按 ID 或名称添加单个 EC2 安全组。

按 ID 或组集中的名称引用单个 EC2 安全组。AWS::EC2::NetworkInterface (p. 283) 在其 GroupSet 属性中使用此类型的对象的列表。

语法

```
{
   "GroupId" : String,
   "GroupName" : String
}
```

属性

键
: 安全组的 ID。
必需：是
类型：字符串

值
: 安全组名称。
必需：是
类型：字符串

EC2 网络接口的私有 IP 规范

Abstract

描述 Amazon EC2 的 PrivateIpAddressSpecification 嵌入式属性的语法和详细信息。

`PrivateIpAddressSpecification` 类型是 AWS::EC2::NetworkInterface (p. 283) 类型的嵌入式属性。

语法

```
{
   "PrivateIpAddress" : String,
   "Primary" : Boolean
}
```

属性

PrivateIpAddress
: 网络接口的私有 IP 地址。
必需：是
类型：字符串

主要
: 将私有 IP 地址设置为主要私有地址。您只能设置一个主要私有 IP 地址。如果您不指定主要私有 IP 地址，Amazon EC2 会自动分配主要私有 IP 地址。
必需：是
类型：布尔值

EC2 PortRange 属性类型

Abstract

指定可用于嵌入式 EC2 PortRange 属性类型的属性。

EC2 PortRange 属性是 AWS::EC2::NetworkAclEntry (p. 281) 类型的嵌入式属性。

以下属性适用于 EC2 PortRange 类型。

属性	类型	必需	备注
付款人	整数	条件	范围中的第一个端口。 条件：如果指定 6 (TCP) 或 17 (UDP) 作为 CreateNetworkAclEntry 协议参数，则为必需属性。
收款人	整数	条件	范围中的最后一个端口。 条件：如果指定 6 (TCP) 或 17 (UDP) 作为 CreateNetworkAclEntry 协议参数，则为必需属性。

EC2 安全组规则属性类型

Abstract

描述 Amazon EC2 的安全组规则嵌入式属性的语法和详细信息。

EC2 安全组规则是 AWS::EC2::SecurityGroup (p. 292) 类型的嵌入式属性。

语法 SecurityGroupIngress

```
{
   "CidrIp (p. 450)" : String,
   "FromPort (p. 451)" : String,
   "IpProtocol (p. 451)" : String,
   "SourceSecurityGroupId (p. 451)" : String,
   "SourceSecurityGroupName (p. 451)" : String,
   "SourceSecurityGroupOwnerId (p. 451)" : String,
   "ToPort (p. 451)" : String
}
```

语法 SecurityGroupEgress

```
{
   "CidrIp (p. 450)" : String,
   "FromPort (p. 451)" : String,
   "IpProtocol (p. 451)" : String,
   "DestinationSecurityGroupId (p. 451)" : String,
   "ToPort (p. 451)" : String
}
```

属性

CidrIp
: 指定一个 CIDR 范围。

  Type: String.

  Required: Conditional.如果您指定 SourceSecurityGroupName 或 SourceSecurityGroupId，请不要指定 CidrIp。

DestinationSecurityGroupId（仅限 SecurityGroupEgress）
指定目标 Amazon VPC 安全组的 GroupId。

Type: String.

Required: Conditional. 不能在指定 CIDR IP 地址时使用。

FromPort
TCP 和 UDP 协议端口范围的起始端口，或者某个 ICMP 类型编号。ICMP 类型编号为 -1 时表示通配符（例如，任何 ICMP 类型编号）。

Type: String.

Required: Yes.

IpProtocol
一个 IP 协议名称或编号。要查看有效值，请参阅 AuthorizeSecurityGroupIngress 中的 IpProtocol 参数

Type: String.

Required: Yes.

SourceSecurityGroupId（仅限 SecurityGroupIngress）
仅限 VPC 安全组。指定允许访问的 Amazon EC2 安全组的 ID。您可以使用 `Ref` 内部函数来引用同一模板中定义的安全组的逻辑 ID。

Type: String.

Required: Conditional..如果您指定 `CidrIp`，请不要指定 `SourceSecurityGroupId`。

SourceSecurityGroupName（仅限 SecurityGroupIngress）
指定将用于访问的 Amazon EC2 安全组的名称。您可以使用 `Ref` 内部函数来引用同一模板中定义的安全组的逻辑名称。

Type: String.

Required: Conditional..如果您指定 CidrIp，请不要指定 SourceSecurityGroupName。

SourceSecurityGroupOwnerId（仅限 SecurityGroupIngress）
指定在 SourceSecurityGroupName 属性中指定的 Amazon EC2 安全组的所有者 AWS 账户 ID。

Type: String.

Required: Conditional..如果指定 SourceSecurityGroupName，并且该安全组的所有者并非创建堆栈的账户，则必须指定 SourceSecurityGroupOwnerId；否则，可根据需要选择是否指定此属性。

ToPort
TCP 和 UDP 协议端口范围的终止端口，或者某个 ICMP 代码。ICMP 代码 -1 表示通配符（例如，任何 ICMP 代码）。

Type: String.

Required: Yes.

示例

具有 CidrIp 的安全组

```
"InstanceSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
```

```
   "Properties" : {
      "GroupDescription" : "Enable SSH access via port 22",
      "SecurityGroupIngress" : [ {
         "IpProtocol" : "tcp",
         "FromPort" : "22",
         "ToPort" : "22",
         "CidrIp" : "0.0.0.0/0"
      } ]
   }
}
```

具有安全组 Id 的安全组

```
"InstanceSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" : {
      "GroupDescription" : "Enable HTTP access on the configured port",
      "VpcId" : { "Ref" : "VpcId" },
      "SecurityGroupIngress" : [ {
         "IpProtocol" : "tcp",
         "FromPort" : { "Ref" : "WebServerPort" },
         "ToPort" : { "Ref" : "WebServerPort" },
         "SourceSecurityGroupId" : { "Ref" : "LoadBalancerSecurityGroup" }
      } ]
   }
}
```

具有多个传入规则的安全组

该代码段可通过 CidrIp 授予 SSH 访问权，通过 SourceSecurityGroupName 授予 HTTP 访问权。`Fn::GetAtt` 用于从弹性负载均衡器派生 SourceSecurityGroupName 和 SourceSecurityGroupOwnerId 的值。

```
"ElasticLoadBalancer" : {
   "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "Listeners" : [ {
         "LoadBalancerPort" : "80",
         "InstancePort" : { "Ref" : "WebServerPort" },
         "Protocol" : "HTTP"
      } ],
      "HealthCheck" : {
         "Target" : { "Fn::Join" : [ "", ["HTTP:", { "Ref" : "WebServerPort"
}, "/"]]},
         "HealthyThreshold" : "3",
         "UnhealthyThreshold" : "5",
         "Interval" : "30",
         "Timeout" : "5"
      }
   }
},
```

```
"InstanceSecurityGroup" : {
   "Type" : "AWS::EC2::SecurityGroup",
   "Properties" : {
      "GroupDescription" : "Enable SSH access and HTTP from the load balancer
only",
      "SecurityGroupIngress" : [ {
         "IpProtocol" : "tcp",
         "FromPort" : "22",
         "ToPort" : "22",
         "CidrIp" : "0.0.0.0/0"
      }, {
         "IpProtocol" : "tcp",
         "FromPort" : { "Ref" : "WebServerPort" },
         "ToPort" : { "Ref" : "WebServerPort" },
        "SourceSecurityGroupOwnerId" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
 "SourceSecurityGroup.OwnerAlias"]},
         "SourceSecurityGroupName" : {"Fn::GetAtt" : ["ElasticLoadBalancer",
"SourceSecurityGroup.GroupName"]}
      } ]
   }
}
```

另请参阅

- *Amazon EC2 User Guide* 中的 Amazon EC2 Security Groups。

EC2 标签

Abstract

描述 EC2 标签嵌入式属性的语法和详细信息。

EC2 标签是 AWS::EC2::Instance (p. 272)、AWS::EC2::RouteTable (p. 290)、AWS::EC2::NetworkAcl (p. 279)、AWS::EC2::NetworkInterface (p. 283)、AWS::EC2::Subnet (p. 300)、AWS::EC2::VPC (p. 310) 和 AWS::EC2::Volume (p. 305) 类型的嵌入式属性。

语法

```
{
   "Key (p. 453)" : String,
   "Value (p. 453)" : String
}
```

属性

密钥
: 该项目的密钥术语。
必需：是
类型：字符串

值
: 与密钥术语关联的值。
必需：是

类型：字符串

示例

EC2 标签通常在列表中提供。您也可以使用 Ref (p. 508) 或 Fn::GetAtt (p. 502) 内部函数为标签提供一个值，在标签中包含有用信息。例如：

```
"Tags": [
   { "Key" : "Role", "Value": "Test Instance" },
   { "Key" : "Application", "Value" : { "Ref" : "AWS::StackName"} }
]
```

使用标签的 EC2 资源以及 Amazon EC2 代码段 (p. 132)页面中提供了更多标签示例。

另请参阅

- *Amazon Elastic Compute Cloud User Guide* 中的 Using Tags

AWS Elastic Beanstalk 环境层属性类型

Abstract

为 AWS Elastic Beanstalk 环境层指定选项组。

说明 AWS::ElasticBeanstalk::Environment (p. 333) 资源的环境层。有关更多信息，请参阅 *AWS Elastic Beanstalk 开发人员指南* 中的环境层。

语法

```
{
   "   (p. 454)" : String,
   "   (p. 454)" : String,
   "   (p. 455)" : String
}
```

成员

名称
: 环境层名称。可以指定 `WebServer` 或 `Worker`。

  Required: No.

  Type: String.

  更新要求：替换 (p. 63)

类型
: 此环境层的类型。您可以为 `WebServer` 层指定 `Standard`，或是为 `Worker` 层指定 `SQS/HTTP`。

  Required: No.

  Type: String.

更新要求：替换 (p. 63)

版本
环境层的版本。

Required: No.

Type: String.

更新要求：无中断 (p. 63)

示例

```
"Tier" : {
  "Type" : "SQS/HTTP",
  "Name" : "Worker",
  "Version" : "1.0"
}
```

AWS Elastic Beanstalk OptionSettings 属性类型

Abstract

指定 AWS Elastic Beanstalk 环境或配置模板使用的选项的列表。

`OptionSettings` 是 AWS::ElasticBeanstalk::Environment (p. 333) 和 AWS::ElasticBeanstalk::ConfigurationTemplate (p. 331) 资源的嵌入式属性。您可使用 `OptionSettings` 属性为 AWS Elastic Beanstalk 环境指定选项组。

Note

您可以使用 `elastic-beanstalk-describe-configuration-settings` 命令来获取 AWS Elastic Beanstalk 配置的有效设置集。有关更多信息，请参阅 *AWS Elastic Beanstalk Developer Guide* 中的 elastic-beanstalk-describe-configuration-settings。

语法

```
{
   "Namespace (p. 455)" : String,
   "OptionName (p. 456)" : String,
   "Value (p. 456)" : String
}
```

成员

命名空间
标识与选项关联的 AWS 资源的唯一命名空间。

Required: Yes.

Type: String.

OptionName
: 配置选项的名称。有关此处可使用的选项的列表，请参阅 *AWS Elastic Beanstalk 开发人员指南* 中的选项值。

  Required: Yes.

  Type: String.

值
: 设置的值。

  Required: Yes.

  Type: String.

示例

使用 `OptionSettings` 的这个示例在 AWS CloudFormation 示例模板中提供：在 ElasticBeanstalkSample.template 中，该模板还提供了它在 `AWS::ElasticBeanstalk::Application`.

```
"OptionSettings" : [ {
   "Namespace" : "aws:autoscaling:launchconfiguration",
   "OptionName" : "EC2KeyName",
   "Value" : { "Ref" : "KeyName" }
} ]
```

另请参阅

- *AWS Elastic Beanstalk Developer Guide* 中的 ConfigurationOptionSetting
- *AWS Elastic Beanstalk Developer Guide* 中的 Option Values

AWS Elastic Beanstalk SourceBundle 属性类型

Abstract

描述 AWS Elastic Beanstalk 应用程序版本的 SourceBundle 嵌入式属性类型。

SourceBundle 属性是 AWS::ElasticBeanstalk::ApplicationVersion (p. 330) 资源的嵌入式属性。

语法

```
{
   "S3Bucket (p. 456)" : String,
   "S3Key (p. 457)" : String
}
```

成员

S3Bucket
: 数据所在的 Amazon S3 存储桶。

  Required: Yes.

Type: String.

S3Key
数据所在的 Amazon S3 密钥。

Required: Yes.

Type: String.

示例

```
{
   "S3Bucket" : { "Fn::Join" :
      ["-", [ "elasticbeanstalk-samples", { "Ref" : "AWS::Region" } ] ] },
   "S3Key" : "php-sample.zip"
}
```

AWS Elastic Beanstalk SourceConfiguration 属性类型

Abstract

使用 AWS Elastic Beanstalk 配置模板中的设置。

将其他 AWS Elastic Beanstalk 配置模板中的设置用于 AWS::ElasticBeanstalk::ConfigurationTemplate (p. 331) 资源类型。

语法

```
{
   "ApplicationName (p. 457)" : String,
   "TemplateName (p. 457)" : String
}
```

成员

ApplicationName
包含要使用的配置模板的 AWS Elastic Beanstalk 应用程序名称。

Required: Yes.

Type: String.

TemplateName
配置模板名称。

Required: Yes.

Type: String.

Elastic Load Balancing AccessLoggingPolicy

Abstract

描述 Elastic Load Balancing 负载均衡器的 AccessLoggingPolicy 属性。

`AccessLoggingPolicy` 属性描述为 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 资源存储访问日志的位置和方式。

语法

```
{
  "EmitInterval (p. 458)" : Integer,
  "    (p. 458)" : Boolean,
  "S3BucketName (p. 458)" : String,
  "S3BucketPrefix (p. 458)" : String
}
```

属性

EmitInterval
发布访问日志的间隔（以分钟为单位）。您可以指定 5 分钟或 60 分钟的间隔。

Required: No.

Type: Integer

已启用
是否为负载均衡器启用日志记录。

Required: Yes.

Type: Boolean.

S3BucketName
存储访问日志文件的 Amazon S3 存储桶的名称。

Required: No.

Type: String.

S3BucketPrefix
所有日志对象键的前缀，如 `my-load-balancer-logs/prod`。如果您在单个存储桶中存储多个源的日志文件，则可以使用前缀区分每个日志文件及其源。

Required: No.

Type: String.

ElasticLoadBalancing AppCookieStickinessPolicy 类型

Abstract

描述嵌入式 AppCookieStickinessPolicy 属性。

AppCookieStickinessPolicy 类型是 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 类型的嵌入式属性。

语法

```
{
   "CookieName (p. 459)" : String,
   "PolicyName (p. 459)" : String
}
```

属性

CookieName
用于粘性的应用程序 cookie 的名称。

必需：是

类型：字符串

PolicyName
正在创建的策略的名称。该名称在此负载均衡器的策略集中必须具有唯一性。

必需：是

类型：字符串

另请参阅

- AWS::ElasticLoadBalancing::LoadBalancer (p. 337)
- ElasticLoadBalancing 策略类型 (p. 463)
- ElasticLoadBalancing LBCookieStickinessPolicy 类型 (p. 461)
- *Elastic Load Balancing API 参考* 中的 CreateAppCookieStickinessPolicy。

Elastic Load Balancing ConnectionDrainingPolicy

Abstract

描述 Elastic Load Balancing 负载均衡器的 ConnectionDrainingPolicy 属性。

`ConnectionDrainingPolicy` 属性描述取消注册或运行状况不佳的实例如何为 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 资源处理处于飞行状态的请求。连接耗尽功能可确保在实例已取消注册或运行状况不佳时，负载均衡器可为已注册实例所有处于飞行状态的请求完成服务。没有连接耗尽功能时，负载均衡器可关闭与取消注册或运行状况不佳的实例的连接，并且不会完成任何处于飞行状态的请求。

有关连接耗尽和默认值的更多信息，请参阅 *Elastic Load Balancing 开发人员指南* 中的为负载均衡器启用或禁用连接耗尽。

语法

```
{
```

```
   "    (p. 460)" : Boolean,
   "Timeout (p. 460)" : Integer
}
```

属性

已启用
: 是否为负载均衡器启用连接耗尽功能。

Required: Yes.

Type: Boolean.

Timeout
: 负载均衡器关闭与取消注册或运行状况不佳的实例的所有连接之后的时间（以秒为单位）。

Required: No.

Type: Integer

ElasticLoadBalancing HealthCheck 类型

Abstract

描述 ElasticLoadBalancing 的嵌入式 HealthCheck 属性。

ElasticLoadBalancing HealthCheck 是 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 类型的嵌入式属性。

语法

```
{
   "HealthyThreshold (p. 460)" : String,
   "Interval (p. 460)" : String,
   "Target (p. 461)" : String,
   "Timeout (p. 461)" : String,
   "UnhealthyThreshold (p. 461)" : String
}
```

属性

HealthyThreshold
: 指定实例在转换为“Healthy”状态之前，需要连续多少次成功通过健康运行探测。

必需：是

类型：字符串

Interval
: 指定单个实例的健康运行检查之间的大致间隔时间（秒）。

必需：是

类型：字符串

目标
指定要检查的实例的协议和端口。协议可以是 TCP、HTTP、HTTPS 或 SSL。有效端口范围从 1 到 65535。

必需：是

类型：字符串

Note

对于 TCP 和 SSL，指定一个端口对。例如，您可以指定 `TCP:5000` 或 `SSL:5000`。运行状况检查尝试在指定端口上打开与实例的 TCP 或 SSL 连接。如果运行状况检查未能在配置的超时期限内连接，则会将实例视为运行状况不佳。

对于 HTTP 或 HTTPS，指定端口和用于发送 Ping 命令的路径 (*`HTTP or HTTPS`*`:`*`port`*`/`*`PathToPing`*)。例如，您可以指定 `HTTP:80/weather/us/wa/seattle`。在这种情况下，系统会在指定端口和路径上向实例发出 HTTP GET 请求。如果运行状况检查在配置的超时期限内收到 `200 OK` 之外的任何响应，则会将实例视为运行状况不佳。HTTP 或 HTTPS ping 目标的总长度不能多于 1024 个 16 位 Unicode 字符。

Timeout
指定检查时长（秒），在此期间无响应意味着未通过健康运行探测。该值必须小于 *`Interval`* 的值。

必需：是

类型：字符串

UnhealthyThreshold
指定实例在转换为“Unhealthy”状态之前，需要连续多少次未通过健康运行探测。

必需：是

类型：字符串

ElasticLoadBalancing LBCookieStickinessPolicy 类型

Abstract

描述 ElasticLoadBalancing 的嵌入式 LBCookieStickinessPolicy 属性。

LBCookieStickinessPolicy 类型是 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 类型的嵌入式属性。

语法

```
{
   "CookieExpirationPeriod (p. 461)" : String,
   "PolicyName (p. 462)" : String
}
```

属性

CookieExpirationPeriod
cookie 在此后应被视为“过期”的时间段（秒）。如果此参数未指定，则粘度会话的持续时间与浏览器会话的持续时间相同。

必需：否

类型：字符串

PolicyName
正在创建的策略的名称。该名称在此负载均衡器的策略集中必须具有唯一性。

另请参阅

- AWS::ElasticLoadBalancing::LoadBalancer (p. 337)
- ElasticLoadBalancing 策略类型 (p. 463)
- ElasticLoadBalancing AppCookieStickinessPolicy 类型 (p. 458)
- *Elastic Load Balancing API 参考* 中的 CreateLBCookieStickinessPolicy

ElasticLoadBalancing Listener 属性类型

Abstract

描述 ElasticLoadBalancing 的嵌入式 Listener 属性。

Listener 属性是 AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 类型的嵌入式属性。

语法

```
{
   "InstancePort (p. 462)" : String,
   "InstanceProtocol (p. 462)" : String,
   "LoadBalancerPort (p. 463)" : String,
   "PolicyNames (p. 463)" :  [ String, ... ],
   "Protocol (p. 463)" : String,
   "SSLCertificateId (p. 463)" : String
}
```

属性

InstancePort
指定实例服务器正在侦听的 TCP 端口。该属性在负载均衡器的使用寿命内无法调整。

Required: Yes.

Type: String.

InstanceProtocol
指定将流量发送至后端实例时使用的协议 – HTTP、HTTPS、TCP 或 SSL。该属性在负载均衡器的使用寿命内无法调整。

Required: No.

Type: String.

Note

- 如果前端协议为 HTTP 或 HTTPS，则 *InstanceProtocol* 也需要处于相同的协议层，即 HTTP 或 HTTPS。同样地，如果前端协议为 TCP 或 SSL，*InstanceProtocol* 也需要为 TCP 或 SSL。
- 如果另一侦听器也采用相同的 InstancePort，并且该端口也采用安全的 *InstanceProtocol*（例如，HTTPS 或 SSL），则该侦听器也需要采用安全的 *InstanceProtocol*（例如，HTTPS 或 SSL）。如果另一侦听器也采用相同的 InstancePort，并且该端口的 *InstanceProtocol* 为 HTTP 或 TCP，则该侦听器的 *InstanceProtocol* 也必须为 HTTP 或 TCP。

LoadBalancerPort
: 指定外部负载均衡器端口号。该属性在负载均衡器的使用寿命内无法调整。

Required: Yes.

Type: String.

PolicyNames
: 将与该侦听器关联的 ElasticLoadBalancing 策略 (p. 463)名称的列表。

Required: No.

Type: A list of strings.

协议
: 指定用于路由的负载均衡器传输协议 – HTTP、HTTPS、TCP 或 SSL。该属性在负载均衡器的使用寿命内无法调整。

Required: Yes.

Type: String.

SSLCertificateId
: 要使用的 SSL 证书的 ARN。有关 SSL 证书的更多信息，请参阅 AWS Identity and Access Management 文档中的 Managing Server Certificates。

Required: No.

Type: String.

ElasticLoadBalancing 策略类型

Abstract

描述为嵌入式 Policy 属性包含侦听器的 PolicyNames 字段的策略。

ElasticLoadBalancing 策略类型是 AWS::ElasticLoadBalancing::Listener (p. 462) 类型的嵌入式属性。它用于说明将包括在侦听器 *PolicyNames* 字段中的策略。

语法

```
{
   "Attributes (p. 464)" : [ { "Name", String, "Value", String }, ... ],
   "InstancePorts (p. 464)" : [ String, ... ],
   "LoadBalancerPorts (p. 464)" : [ String, ... ],
   "PolicyName (p. 464)" : String,
```

```
   "PolicyType (p. 464)" : String
}
```

属性

属性
该策略的任意属性列表。

必需：否

类型：JSON 名称–值对列表。

InstancePorts
该策略的实例端口列表。这些是与后端服务器关联的端口。

必需：否

类型：字符串列表

LoadBalancerPorts
该策略的外部负载均衡器端口列表。

必需：仅对于一些策略为必需。有关更多信息，请参阅 *Elastic Load Balancing 开发人员指南*。

类型：字符串列表

PolicyName
对负载均衡器具有唯一性的策略名称。

必需：是

类型：字符串

PolicyType
该策略的策略类型名称。该参数必须为 Elastic Load Balancing DescribeLoadBalancerPolicyTypes 操作报告的类型之一。

必需：是

类型：字符串

示例

本示例显示了 ELB 侦听器“Policies”部分的代码段。

```
"Policies" : [
   {
      "PolicyName" : "MySSLNegotiationPolicy",
      "PolicyType" : "SSLNegotiationPolicyType",
      "Attributes" : [
         { "Name" : "Protocol-TLSv1", "Value" : "true" },
         { "Name" : "Protocol-SSLv2", "Value" : "true" },
         { "Name" : "Protocol-SSLv3", "Value" : "false" },
         { "Name" : "DHE-RSA-AES256-SHA", "Value" : "true" } ]
   }, {
      "PolicyName" : "MyAppCookieStickinessPolicy",
      "PolicyType" : "AppCookieStickinessPolicyType",
      "Attributes" : [
```

```
         { "Name" : "CookieName", "Value" : "MyCookie"} ]
   }, {
      "PolicyName" : "MyPublicKeyPolicy",
      "PolicyType" : "PublicKeyPolicyType",
      "Attributes" : [ {
         "Name" : "PublicKey",
         "Value" : { "Fn::Join" : [
            "\n", [
               "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDh/51Aohx5Vrpm
lfGHZCzciMBa",
               "fkHve+MQYYJcxmNUKMdsWnz9WtVfKxxWUU7Cfor4lorYmENGCG8FWqCoLD
MFs7pN",
               "yGEtpsrlKhzZWtgY1d7eGrUrBil03bI90E2KW0j4qAwGYAC8xix
OkNClicojeEz4",
               "f4rr3sUf+ZBSsuMEuwIDAQAB" ]
         ] }
      } ]
   }, {
      "PolicyName" : "MyBackendServerAuthenticationPolicy",
      "PolicyType" : "BackendServerAuthenticationPolicyType",
      "Attributes" : [
         { "Name" : "PublicKeyPolicyName", "Value" : "MyPublicKeyPolicy" } ],
      "InstancePorts" : [ "8443" ]
   }
]
```

此示例说明使用代理协议的 ELB 的“Policies”部分中的代码段。

```
"Policies" : [{
   "PolicyName" : "EnableProxyProtocol",
   "PolicyType" : "ProxyProtocolPolicyType",
   "Attributes" : [{
      "Name"  : "ProxyProtocol",
      "Value" : "true"
   }],
   "InstancePorts" : [{"Ref" : "WebServerPort"}]
}]
```

另请参阅

- AWS::ElasticLoadBalancing::LoadBalancer (p. 337)
- ElasticLoadBalancing AppCookieStickinessPolicy 类型 (p. 458)
- ElasticLoadBalancing LBCookieStickinessPolicy 类型 (p. 461)

名称类型

Abstract

为某些资源指定更易于读取和识别的自定义名称。

对于某些资源，您可以指定自定义名称。默认情况下，AWS CloudFormation 会生成用于为资源命名的唯一物理 ID。例如，AWS CloudFormation 可能会用物理 ID `stack123123123123-s3bucket-abcdefghijk1` 来命名 Amazon S3 存储桶。借助自定义名称，您可以指定一个更易于读取和识别的名称，如 `production-app-logs` 或 `business-metrics`。

如果您重复使用模板来创建多个堆栈，则必须从模板来更改或删除自定义名称。每个堆栈的资源名称必须是唯一的。在您删除名称时，AWS CloudFormation 会生成用于为该资源命名的唯一物理 ID。

Important

如果自定义命名的资源在更新过程中需要替换，则无法更新这些资源。

如果您想要使用自定义名称，请在 AWS CloudFormation 模板中为该资源指定名称属性。支持自定义名称的每个资源都具有可以指定的自身属性。例如，若要命名一个 Amazon DynamoDB 表，您可以使用 `TableName` 属性，如下例所示：

```
"myDynamoDBTable" : {
   "Type" : "AWS::DynamoDB::Table",
   "Properties" : {
      "KeySchema" : {
         "HashKeyElement": {
            "AttributeName" : "AttributeName1",
            "AttributeType" : "S"
         },
         "RangeKeyElement" : {
            "AttributeName" : "AttributeName2",
            "AttributeType" : "N"
         }
      },
      "ProvisionedThroughput" : {
         "ReadCapacityUnits" : "5",
         "WriteCapacityUnits" : "10"
      },
      "TableName" : "Sample Table"
   }
}
```

请勿管理 AWS CloudFormation 外部的堆栈资源。例如，如果您重命名作为堆栈一部分的一个 Amazon S3 存储桶而不使用 AWS CloudFormation，则您在尝试更新或删除该堆栈时，随时有可能获取错误。

AWS OpsWorks Recipes 类型

Abstract

描述 AWS OpsWorks 层的自定义事件配方。

描述 AWS OpsWorks 在标准事件配方之后运行的 AWS::OpsWorks::Layer (p. 364) 资源类型自定义配方。有关更多信息，请参阅 *AWS OpsWorks 用户指南* 中的 AWS OpsWorks 生命周期事件。

语法

```
{
  "   (p. 467)" : [ String, ... ],
  "Deploy (p. 467)" : [ String, ... ],
  "Setup (p. 467)" : [ String, ... ],
  "   (p. 467)" : [ String, ... ],
  "Undeploy (p. 467)" : [ String, ... ]
}
```

属性

配置
在 Configure 事件之后运行的自定义配方名称。该事件在实例进入或退出在线状态时，在堆栈的所有实例上发生。

Required: No.

Type: A list of strings.

Deploy
在 Deploy 事件之后运行的自定义配方名称。该事件在您运行 `deploy` 命令时发生，通常用于将应用程序部署到一组应用程序服务器实例。

Required: No.

Type: A list of strings.

Setup
在 Setup 事件之后运行的自定义配方名称。此事件在新实例成功启动之后，在该实例上发生。

Required: No.

Type: A list of strings.

关闭
在 Shutdown 事件之后运行的自定义配方名称。此事件在您指示 AWS OpsWorks 关闭实例之后以及关联 Amazon EC2 实例实际终止之前发生。

Required: No.

Type: A list of strings.

Undeploy
在 Undeploy 事件之后运行的自定义配方名称。此事件在您删除应用程序或运行 `undeploy` 命令以从一组应用程序服务器实例中删除应用程序时发生。

Required: No.

Type: A list of strings.

AWS OpsWorks Source 类型

Abstract

描述 AWS OpsWorks 堆栈的源存储库信息。

描述针对 AWS::OpsWorks::Stack (p. 368) 或 AWS::OpsWorks::App (p. 358) 资源类型从存储库检索食谱或应用程序所需的信息。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 Source。

语法

```
{
  "   (p. 468)" : String,
  "Revision (p. 468)" : String,
  "SshKey (p. 468)" : String,
  "   (p. 468)" : String,
  "Url (p. 468)" : String,
```

```
  "    (p. 468)" : String
}
```

属性

密码
: 此参数取决于存储库类型。对于 Amazon S3 捆绑，请将 `Password` 设置为合适的 IAM 私有访问密钥。对于 HTTP 捆绑、Git 存储库和 Subversion 存储库，请将 `Password` 设置为合适的密码。

  Required: No.

  Type: String.

Revision
: 应用程序版本。借助 AWS OpsWorks，您可以部署新版本的应用程序。一种最简单的方法是在存储库中包含表示可以部署的不同版本的分支或修订版。

  Required: No.

  Type: String.

SshKey
: 存储库的 SSH 密钥。

  Required: No.

  Type: String.

类型
: 存储库类型。

  Required: No.

  Type: String.

Url
: 源 URL。

  Required: No.

  Type: String.

用户名
: 此参数取决于存储库类型。对于 Amazon S3 捆绑，请将 `Username` 设置为合适的 IAM 访问密钥 ID。对于 HTTP 捆绑、Git 存储库和 Subversion 存储库，请将 `Username` 设置为合适的用户名称。

  Required: No.

  Type: String.

AWS OpsWorks SslConfiguration 类型

Abstract

描述 AWS OpsWorks 应用程序的 SSL 配置。

描述 AWS::OpsWorks::App (p. 358) 资源类型的 SSL 配置。

语法

```
{
  "Certificate (p. 469)" : String,
  "Chain (p. 469)" : String,
  "PrivateKey (p. 469)" : String
}
```

属性

Certificate
证书的 domain.crt 文件的内容。

Required: Yes.

Type: String.

Chain
中间证书颁发机构密钥或客户端身份验证。

Required: No.

Type: String.

PrivateKey
私有密钥；证书的 domain.kex 文件的内容。

Required: Yes.

Type: String.

AWS OpsWorks StackConfigurationManager 类型

Abstract

描述 AWS OpsWorks 堆栈的堆栈配置管理器。

描述 AWS::OpsWorks::Stack (p. 368) 资源类型的堆栈配置管理器。有关更多信息，请参阅 *AWS OpsWorks API Reference* 中的 StackConfigurationManager。

语法

```
{
  "   (p. 469)" : String,
  "   (p. 470)" : String
}
```

属性

名称
配置管理器的名称。

Required: No.

Type: String.

版本
Chef 版本。

Required: No.

Type: String.

AWS OpsWorks VolumeConfiguration 类型

Abstract

描述 AWS OpsWorks 层的卷配置。

描述 AWS::OpsWorks::Layer (p. 364) 资源类型的 Amazon EBS 卷。

语法

```
{
  "MountPoint (p. 470)" : String,
  "NumberOfDisks (p. 470)" : Number,
  "RaidLevel (p. 470)" : Number,
  "   (p. 470)" : Number
}
```

属性

MountPoint
卷安装点，如 `/dev/sdh`。

Required: Yes.

Type: String.

NumberOfDisks
卷中的磁盘数。

Required: Yes.

Type: Number.

RaidLevel
卷 RAID 级别。

Required: No.

Type: Number.

大小
卷大小。

Required: Yes.

Type: Number.

Amazon Redshift 参数类型

Abstract

描述 AWS::Redshift::ClusterParameterGroup 资源类型的 Amazon Redshift 参数。

描述 AWS::Redshift::ClusterParameterGroup (p. 376) 资源类型的参数。

语法

```
{
  "ParameterName (p. 471)" : String,
  "ParameterValue (p. 471)" : String
}
```

属性

ParameterName
参数的名称。

Required: Yes.

Type: String.

ParameterValue
参数的值。

Required: Yes.

Type: String.

AWS CloudFormation 资源标签类型

Abstract

使用资源标签属性类型将用户定义的标签应用于支持的资源类型。

AWS CloudFormation 资源标签属性用于将用户定义的标签应用于 AWS CloudFormation 中支持的资源类型。标签可用于帮助您识别和分类资源，以用于各种用途。

除了定义的任何标签之外，AWS CloudFormation 还会自动创建带有前缀“aws:”的以下堆栈级别标签：

- aws:cloudformation:logical-id
- aws:cloudformation:stack-id
- aws:cloudformation:stack-name

所有堆栈级标签，包括自动创建的标签在内，都会被传递给支持添加标签的资源。但是，当前标签不会传播到从块储存设备映射创建的 Amazon EBS 卷。

AWS CloudFormation 支持将标签用于支持它的资源。有关更多信息，请参阅 AWS 资源类型参考 (p. 217) 中的各个资源。

Note

标签实现方式可能因资源而异。例如，AWS::AutoScaling::AutoScalingGroup 提供了一个附加的必需 PropagateAtLaunch 属性以作为其标签方案的一部分。

语法

```
{
   "Key (p. 472)" : String,
   "Value (p. 472)" : String
}
```

属性

键

标签的密钥名称。

必需：是

类型：字符串

值

标签的值。

必需：是

类型：字符串

另请参阅

- 设置堆栈选项 (p. 83)
- 查看堆栈数据和资源 (p. 85)

RDS 安全组规则

Abstract

描述 RDS 安全组规则嵌入式属性类型的语法和详细信息。

RDS 安全组规则是 AWS::RDS::DBSecurityGroup (p. 392) 类型的嵌入式属性。

语法

```
{
   "CIDRIP (p. 473)": String,
   "EC2SecurityGroupId (p. 473)": String,
   "EC2SecurityGroupName (p. 473)": String,
   "EC2SecurityGroupOwnerId (p. 473)": String
}
```

属性

CIDRIP
要授权的 IP 范围。

有关 CIDR 范围的概述，请访问 Wikipedia Tutorial。
类型：字符串
必需：否

更新要求：替换 (p. 63)

EC2SecurityGroupId
要授权的 VPC 或 EC2 安全组 ID。

对于 VPC 数据库安全组，请使用 EC2SecurityGroupId。对于 EC2 安全组，可使用 EC2SecurityGroupOwnerId 和 EC2SecurityGroupName 或 EC2SecurityGroupId。
类型：字符串
必需：否

更新要求：替换 (p. 63)

EC2SecurityGroupName
要授权的 EC2 安全组的名称。

对于 VPC 数据库安全组，请使用 EC2SecurityGroupId。对于 EC2 安全组，可使用 EC2SecurityGroupOwnerId 和 EC2SecurityGroupName 或 EC2SecurityGroupId。
类型：字符串
必需：否

更新要求：替换 (p. 63)

EC2SecurityGroupOwnerId
指定 EC2 安全组（在 EC2SecurityGroupName 参数中指定的）所有人的 AWS 账户。AWS 访问密钥 ID 不是认可的值。

对于 VPC 数据库安全组，请使用 EC2SecurityGroupId。对于 EC2 安全组，可使用 EC2SecurityGroupOwnerId 和 EC2SecurityGroupName 或 EC2SecurityGroupId。
类型：字符串
必需：否

更新要求：替换 (p. 63)

Route 53 AliasTarget 属性

Abstract

描述 Amazon Route 53 资源的别名目标。

`AliasTarget` 是 AWS::Route53::RecordSet (p. 396) 资源的属性。

有关别名资源记录集的更多信息，请参阅 *Amazon Route 53 开发人员指南* 中的创建别名资源记录集。

语法

```
{
  "DNSName (p. 474)" : String,
  "EvaluateTargetHealth (p. 474)" : Boolean,
```

```
  "HostedZoneId (p. 474)" : String
}
```

属性

DNSName
作为别名目标的负载均衡器的 DNS 名称。

Type: String.

Required: Yes.

EvaluateTargetHealth
响应 DNS 查询时，Route 53 是否在别名目标中检查资源记录集的运行状况。有关使用此属性的更多信息，请参阅 *Amazon Route 53 API 参考* 中的 EvaluateTargetHealth。

Type: Boolean.

Required: No.

HostedZoneId
作为别名目标的负载均衡器的托管区域 ID。

Type: String.

Required: Yes.

Amazon S3 Cors 配置

Abstract

用于 Amazon S3 存储桶中的对象的一组源和方法（希望允许的跨源访问）。

描述用于 AWS::S3::Bucket (p. 402) 资源中的对象的跨源访问配置。

语法

```
{
  "CorsRule (p. 474)" : [ Cors Rule, ... ]
}
```

属性

CorsRule
您允许的一组源和方法。

Required: Yes.

类型：Amazon S3 Cors 配置规则 (p. 474)

Amazon S3 Cors 配置规则

Abstract

描述用于 Amazon S3 存储桶中的对象的跨源访问规则。

描述用于 Amazon S3 Cors 配置 (p. 474) 属性的跨源访问规则。

语法

```
{
  "AllowedHeaders (p. 475)" : [ String, ... ],
  "AllowedMethods (p. 475)" : [ String, ... ],
  "AllowedOrigins (p. 475)" : [ String, ... ],
  "ExposedHeaders (p. 475)" : [ String, ... ],
  "Id (p. 475)" : String,
  "MaxAge (p. 475)" : Integer
}
```

属性

AllowedHeaders
在 `Access-Control-Request-Headers` 标头中指定的标头。在预检 OPTIONS 请求中允许使用这些标头。作为对任何预检 OPTIONS 请求的响应，Amazon S3 将返回允许的任意请求的标头。

Required: No.

Type: A list of strings.

AllowedMethods
您允许源执行的 HTTP 方法。有效值包括 `GET`、`PUT`、`HEAD`、`POST` 和 `DELETE`。

Required: Yes.

Type: A list of strings.

AllowedOrigins
您允许发送跨域请求的源。

Required: Yes.

Type: A list of strings.

ExposedHeaders
响应中可供客户端应用程序访问的一个或多个标头（例如，来自 JavaScript XMLHttpRequest 对象）。

Required: No.

Type: A list of strings.

Id
此规则的唯一标识符。值不能多于 255 个字符。

Required: No.

Type: String.

MaxAge
浏览器为指定资源缓存预检响应的时间（以秒为单位）。

Required: No.

Type: Integer

Amazon S3 生命周期配置

Abstract

描述用于 Amazon S3 中的对象的生命周期配置。

描述用于 AWS::S3::Bucket (p. 402) 资源中的对象的生命周期配置。

语法

```
{
  "   (p. 476)" : [ Lifecycle Rule, ... ]
}
```

属性

规则
用于存储桶中的各个对象的生命周期规则。

Required: Yes.

类型：Amazon S3 生命周期规则 (p. 476)

Amazon S3 生命周期规则

Abstract

描述用于 Amazon S3 存储桶中的对象的生命周期规则。

描述用于 Amazon S3 生命周期配置 (p. 476) 属性的生命周期规则。

语法

```
{
  "ExpirationDate (p. 476)" : String,
  "ExpirationInDays (p. 477)" : Integer,
  "Id (p. 477)" : String,
  "   (p. 477)" : String,
  "   (p. 477)" : String,
  "Transition (p. 477)" : Transition type
}
```

属性

ExpirationDate
指示从 Amazon S3 和 Amazon Glacier 中删除对象的时间。该值必须采用 ISO 8601 格式。时间始终为午夜 UTC。

您必须指定以下至少一个属性：ExpirationDate、ExpirationInDays 或 Transition。如果您指定过期和转换时间，则必须对两个属性使用相同的时间单位（按天或按日期）。过期时间还必须晚于转换时间。

Required: Conditional.

Type: String.

ExpirationInDays
指示在创建之后的多少天从 Amazon S3 和 Amazon Glacier 中删除对象。

您必须指定以下至少一个属性：`ExpirationDate`、`ExpirationInDays` 或 `Transition`。如果您指定过期和转换时间，则必须对两个属性使用相同的时间单位（按天或按日期）。过期时间还必须晚于转换时间。

Required: Conditional.

Type: Integer

Id
此规则的唯一标识符。值不能多于 255 个字符。

Required: No.

Type: String.

前缀
标识要应用此规则的一个或多个对象的对象键前缀。

Required: No.

Type: String.

状态
指定 `Enabled` 或 `Disabled`。如果您指定 `Enabled`，则 Amazon S3 按计划执行此规则。如果您指定 `Disabled`，则 Amazon S3 忽略此规则。

Required: Yes.

Type: String.

Transition
描述对象转换为指定存储类的时间。

您必须指定以下至少一个属性：`ExpirationDate`、`ExpirationInDays` 或 `Transition`。如果您指定过期和转换时间，则必须对两个属性使用相同的时间单位（按天或按日期）。转换时间还必须早于过期时间。

Required: Conditional.

类型：Amazon S3 生命周期规则转换 (p. 477)

Amazon S3 生命周期规则转换

Abstract

描述用于 Amazon S3 规则属性的转换规则。

描述对于 Amazon S3 生命周期规则 (p. 476) 属性，对象转换为指定存储类的时间。

语法

```
{
  "StorageClass (p. 478)" : String,
  "TransitionDate (p. 478)" : String,
  "TransitionInDays (p. 478)" : Integer
}
```

属性

StorageClass
要将对象转换为的存储类。当前只能指定 `Glacier`。

Required: Yes.

Type: String.

TransitionDate
指示在创建之后的多少天将对象转换为指定存储类。

Required: Conditional.

Type: String.

TransitionInDays
指示对象转换为指定存储类的时间。该值必须采用 ISO 8601 格式。时间始终为午夜 UTC。

Required: Conditional.

Type: Integer

Amazon S3 日志记录配置

Abstract

描述用于跟踪对 Amazon S3 存储桶资源的请求的日志记录配置。

描述对于 AWS::S3::Bucket (p. 402) 资源，存储日志的位置以及 Amazon S3 分配给所有日志对象键的前缀。这些日志跟踪对 Amazon S3 存储桶的请求。有关更多信息，请参阅 *Amazon Simple Storage Service API 参考* 中的 PUT Bucket 日志记录。

语法

```
{
  "DestinationBucket (p. 478)" : String,
  "LogFilePrefix (p. 478)" : String
}
```

属性

DestinationBucket
Amazon S3 用于存储服务器访问日志文件的 Amazon S3 存储桶的名称。您可以在您拥有的任何存储桶中存储日志文件。默认情况下，日志存储在定义了 `LoggingConfiguration` 属性的存储桶中。

Required: No.

Type: String.

LogFilePrefix
所有日志对象键的前缀。如果您在单个存储桶中存储来自多个 Amazon S3 存储桶的日志文件，则可以使用前缀区分哪些日志文件来自哪个存储桶。

Required: No.

Type: String.

Amazon S3 通知配置

Abstract

描述用于 Amazon S3 存储桶的通知配置。

描述用于 AWS::S3::Bucket (p. 402) 资源的通知配置。

语法

```
{
  "TopicConfigurations (p. 479)" : [ Topic Configuration, ... ]
}
```

属性

TopicConfigurations
向其发送通知的主题以及为其生成通知的事件。

Required: Yes.

类型：Amazon S3 通知主题配置 (p. 479)

Amazon S3 通知主题配置

Abstract

描述通知配置属性的主题和事件。

描述 Amazon S3 通知配置 (p. 479) 属性的主题和事件。

语法

```
{
  "   (p. 479)" : String,
  "   (p. 479)" : String
}
```

属性

事件
针对其发送通知的 Amazon S3 存储桶事件。当前，`s3:ReducedRedundancyLostObject` 是通知唯一支持的事件。

Required: Yes.

Type: String.

主题
Amazon S3 向其报告指定事件的 Amazon SNS 主题。

Required: Yes.

Type: String.

Amazon S3 版本控制配置

Abstract

描述 Amazon S3 存储桶资源的版本控制状态。

描述 AWS::S3::Bucket (p. 402) 资源的版本控制状态。有关更多信息，请参阅 *Amazon Simple Storage Service API 参考* 中的 PUT Bucket 版本控制。

语法

```
{
  "    (p. 480)" : String
}
```

属性

状态
: Amazon S3 存储桶的版本控制状态。如果您启用了版本控制，则必须暂停版本控制才能禁用它。

  Required: Yes.

  Type: String.

Amazon S3 网站配置属性

Abstract

描述 Amazon S3 存储桶的网站配置属性。

WebsiteConfiguration 是 AWS::S3::Bucket (p. 402) 资源的嵌入式属性。

语法

```
"WebsiteConfiguration" : {
   "ErrorDocument (p. 480)" : String,
   "IndexDocument (p. 480)" : String,
   "RedirectAllRequestsTo (p. 481)" : Redirect all requests rule,
   "RoutingRules (p. 481)" : [ Routing rule, ... ]
}
```

属性

ErrorDocument
: 网站错误文档的名称。

  Required: No.

  Type: String.

IndexDocument
: 网站索引文档的名称。

Required: No.

Type: String.

RedirectAllRequestsTo
对此存储桶网站终端节点进行的每个请求的重定向行为。

Important

如果您指定此属性，则无法指定任何其他属性。

Required: No.

类型：Amazon S3 网站配置“重定向所有请求至”属性 (p. 481)

RoutingRules
定义重定向的应用时间和重定向行为的规则。

Required: No.

类型：Amazon S3 网站配置路由规则属性 (p. 482)

示例

```
"S3Bucket" : {
   "Type" : "AWS::S3::Bucket",
   "Properties" : {
      "AccessControl" : "PublicRead",
      "WebsiteConfiguration" : {
         "IndexDocument" : "index.html",
         "ErrorDocument" : "error.html"
      }
   }
}
```

另请参阅

- *Amazon Simple Storage Service Developer Guide* 中的 Custom Error Document Support
- *Amazon Simple Storage Service Developer Guide* 中的 Index Document Support

Amazon S3 网站配置“重定向所有请求至”属性

Abstract

描述对 Amazon S3 存储桶网站终端节点进行的所有请求的重定向行为。

`RedirectAllRequestsTo` 代码是 Amazon S3 网站配置属性 (p. 480) 属性的嵌入式属性，用于描述对 Amazon S3 存储桶网站终端节点进行的所有请求的重定向行为。

语法

```
"RedirectAllRequestsTo" : {
   "HostName (p. 482)" : String,
```

```
   "    (p. 482)" : String
}
```

属性

HostName
: 请求重定向到的主机的名称。

  Required: Yes.

  Type: String.

协议
: 重定向请求时要使用的协议（`http` 或 `https`）。默认值是原始请求中使用的协议。

  Required: No.

  Type: String.

Amazon S3 网站配置路由规则属性

Abstract

描述重定向行为以及对 Amazon S3 存储桶网站终端节点应用重定向的时间。

`RoutingRules` 属性是 Amazon S3 网站配置属性 (p. 480) 属性的嵌入式属性。此属性描述重定向行为以及应用重定向的时间。

语法

```
"RoutingRules" : {
   "RedirectRule (p. 482)" : Redirect rule,
   "RoutingRuleCondition (p. 482)" : Routing rule condition
}
```

属性

RedirectRule
: 将请求重定向到其他主机、重定向到其他页面或使用其他协议重定向。

  Required: Yes.

  类型：Amazon S3 网站配置路由规则重定向规则属性 (p. 483)

RoutingRuleCondition
: 定义应用重定向的时间的规则。

  Required: No.

  类型：Amazon S3 网站配置路由规则条件属性 (p. 484)

Amazon S3 网站配置路由规则重定向规则属性

Abstract

描述如何为 Amazon S3 存储桶网站终端节点重定向请求。

`RedirectRule` 属性是 Amazon S3 网站配置路由规则属性 (p. 482) 的嵌入式属性，用于描述如何重定向请求。出现错误时，您可以指定不同的错误代码进行返回。

语法

```
"RedirectRule" : {
   "HostName (p. 483)" : String,
   "HttpRedirectCode (p. 483)" : String,
   "   (p. 483)" : String,
   "ReplaceKeyPrefixWith (p. 483)" : String,
   "ReplaceKeyWith (p. 483)" : String
}
```

属性

HostName
: 请求重定向到的主机的名称。

  Required: No.

  Type: String.

HttpRedirectCode
: 要在响应时使用的 HTTP 重定向代码。

  Required: No.

  Type: String.

协议
: 要在重定向请求中使用的协议。

  Required: No.

  Type: String.

ReplaceKeyPrefixWith
: 要在重定向请求中使用的对象键前缀。例如，要将对前缀为 `docs/` 的所有页面（`docs/` 文件夹中的对象）进行的请求重定向到 `documents/` 前缀，可以在路由条件属性中将 `KeyPrefixEquals` 属性设置为 `docs/`，并将 *ReplaceKeyPrefixWith* 属性设置为 `documents/`。

  Important

  如果指定此属性，则无法指定 `ReplaceKeyWith` 属性。

  Required: No.

  Type: String.

ReplaceKeyWith
: 要在重定向请求中使用的特定对象键。例如，将请求重定向到 `error.html`。

Important

如果指定此属性，则无法指定 `ReplaceKeyPrefixWith` 属性。

Required: No.

Type: String.

Amazon S3 网站配置路由规则条件属性

Abstract

描述为应用重定向而必须满足的条件。

`RoutingRuleCondition` 属性是 Amazon S3 网站配置路由规则属性 (p. 482) 的嵌入式属性，用于描述为应用重定向而必须满足的条件。

语法

```
"RoutingRuleCondition" : {
   "HttpErrorCodeReturnedEquals (p. 484)" : String,
   "KeyPrefixEquals (p. 484)" : String
}
```

属性

HttpErrorCodeReturnedEquals
: 发生错误时，如果错误代码等于此值，则应用此重定向。

  Required: Conditional..您必须至少指定一个条件属性。

  Type: String.

KeyPrefixEquals
: 应用重定向时的对象键名称前缀。例如，要重定向对 `ExamplePage.html` 进行的请求，请将键前缀设置为 `ExamplePage.html`。要重定向对前缀为 `docs/` 的所有页面进行的请求，请将键前缀设置为 `docs/`，这会标识 `docs/` 文件夹中的所有对象。

  Required: Conditional..您必须至少指定一个条件属性。

  Type: String.

SNS 订阅属性类型

Abstract

指定可用于嵌入式 SNS 订阅属性类型的属性。

SNS 订阅是 AWS::SNS::Topic (p. 411) 类型的嵌入式属性。

以下属性可用于 Amazon SNS 订阅类型。

属性	类型	必需	备注
终端节点	字符串	是	订阅的终端节点（格式视协议而定）。
协议	字符串	是	订阅的协议。

Amazon SQS RedrivePolicy

Abstract

说明 Amazon SQS 队列的重新驱动策略。

RedrivePolicy 类型是 AWS::SQS::Queue (p. 414) 资源的属性。

语法

```
{
  "deadLetterTargetArn (p. 485)" : String,
  "maxReceiveCount (p. 485)" : Integer
}
```

属性

deadLetterTargetArn
在超过 `maxReceiveCount` 值之后向其发送消息的死信队列的亚马逊资源名称 (ARN)。

Required: No.

Type: String.

maxReceiveCount
消息在发送到死信队列之前传输给源队列的次数。

Required: No.

Type: Integer

资源属性引用

本部分将详细介绍可添加至某一资源，从而控制其他行为和关系的属性。

Topics

- DeletionPolicy 属性 (p. 485)
- DependsOn 属性 (p. 486)
- Metadata 属性 (p. 488)
- UpdatePolicy 属性 (p. 489)

DeletionPolicy 属性

Abstract

指定如何使用 DeletionPolicy 属性在 AWS CloudFormation 中处理资源删除。

DeletionPolicy 属性可用于指定 AWS CloudFormation 如何处理资源删除。通过向资源添加 DeletionPolicy，您可以控制 AWS CloudFormation 在堆栈被删除时处理资源的方式。AWS CloudFormation 会默认删除不具有 DeletionPolicy 属性的资源。您可以为 AWS CloudFormation 指定 *Retain*，以保留资源而不将其删除。对于支持 AWS::RDS::DBInstance 和 AWS::EC2::Volume 等快照的资源，您可以为 AWS CloudFormation 指定 *Snapshot*，从而在删除资源前拍摄快照。

例如，以下模板包含具有 *Retain* 删除策略的 Amazon S3 存储桶资源。此堆栈被删除时，AWS CloudFormation 将保留该存储桶，不将其删除。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "myS3Bucket" : {
         "Type" : "AWS::S3::Bucket",
         "DeletionPolicy" : "Retain"
      }
   }
}
```

DeletionPolicy 选项

删除
: 默认值。该策略指示 AWS CloudFormation 在删除堆栈时删除资源及其所有内容（如果适用）。您可以向任何资源类型添加此删除策略。

Note

对于 S3 存储桶，只有当存储桶清空时，才算成功完成删除。

保留
: 该策略指示 AWS CloudFormation 在删除堆栈时保留资源且不删除资源或其内容（如果适用）。您可以向任何资源类型添加此删除策略。请注意，AWS CloudFormation 完成堆栈删除后，堆栈将处于 Delete_Complete 状态，但采用“保留”策略的资源将继续保留，并且仍将产生相应费用，直至您删除这些资源。

快照
: 此策略仅适用于支持快照的资源：AWS::EC2::Volume、AWS::RDS::DBInstance 和 AWS::Redshift::Cluster。此策略指示 AWS CloudFormation 在删除资源之前为其创建快照。请注意，AWS CloudFormation 完成堆栈删除后，堆栈将处于 Delete_Complete 状态，但采用此策略创建的快照将继续保留，并且仍将产生相应费用，直至您删除这些快照。

DependsOn 属性

Abstract

使用 DependsOn 属性可指定特定资源的创建取决于 AWS CloudFormation 中的另一个资源。

使用 *DependsOn* 属性可以指定特定资源紧跟着另一个资源创建。在您为资源添加 DependsOn 属性时，该资源仅在创建 DependsOn 属性中指定的资源之后创建。您可以对任何资源使用 DependsOn 属性。下面是一些一般用途：

- 确定等待条件生效的时间。有关更多信息，请参阅 在模板中创建等待条件 (p. 188)。

- 为必须按特定顺序创建或删除的资源声明依赖关系。如果您的 AWS CloudFormation 模板定义 VPC、网关和网关连接，VPC 中具有公有 IP 地址的所有资源都依赖于网关连接。有关更多信息，请参阅 需要 DependsOn 属性的资源 (p. 488)。
- 在创建、更新或删除资源时会覆盖默认的并行机制。AWS CloudFormation 尽可能并行地创建、更新和删除资源。它自动确定模板中的哪些资源可以并行处理，哪些资源具有要求其他操作先完成的依赖关系。您可以使用 DependsOn 明确指定依赖关系，这些依赖关系会覆盖默认的并行机制，指示 CloudFormation 按指定的顺序对这些资源执行操作。

语法

DependsOn 属性可以包含一个字符串或字符串列表。

```
"DependsOn" : [ String, ... ]
```

示例

以下模板包含一个 AWS::EC2::Instance (p. 272) 资源，该资源具有指定 myDB 即 AWS::RDS::DBInstance (p. 381) 的 DependsOn 属性。AWS CloudFormation 在创建此堆栈时，将首先创建 myDB，然后创建 Ec2Instance。

```
{
    "AWSTemplateFormatVersion" : "2010-09-09",
    "Mappings" : {
        "RegionMap" : {
            "us-east-1" : { "AMI" : "ami-76f0061f" },
            "us-west-1" : { "AMI" : "ami-655a0a20" },
            "eu-west-1" : { "AMI" : "ami-7fd4e10b" },
            "ap-northeast-1" : { "AMI" : "ami-8e08a38f" },
            "ap-southeast-1" : { "AMI" : "ami-72621c20" }
        }
    },
    "Resources" : {
        "Ec2Instance" : {
            "Type" : "AWS::EC2::Instance",
            "Properties" : {
                "ImageId" : {
                   "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" },
 "AMI" ]
                }
            },
            "DependsOn" : "myDB"
        },
        "myDB" : {
            "Type" : "AWS::RDS::DBInstance",
            "Properties" : {
               "AllocatedStorage" : "5",
               "DBInstanceClass" : "db.m1.small",
               "Engine" : "MySQL",
               "EngineVersion" : "5.5",
               "MasterUsername" : "MyName",
               "MasterUserPassword" : "MyPassword"
            }
        }
```

```
    }
}
```

需要 DependsOn 属性的资源

VPC 中具有关联公有 IP 地址的资源取决于网关连接。当前，以下资源取决于 VPC 网关连接：

- Auto Scaling 组
- Amazon EC2 实例
- Elastic Load Balancing 负载均衡器
- 弹性 IP 地址

例如，如果 VPC 和 InternetGateway 资源也在相同模板中声明，则具有公有 IP 地址的 Amazon EC2 实例依赖于 VPC 网关连接。以下代码段演示一个示例网关连接和依赖于该网关连接的 Amazon EC2 实例：

```
"GatewayToInternet" : {
  "Type" : "AWS::EC2::VPCGatewayAttachment",
  "Properties" : {
    "VpcId" : { "Ref" : "VPC" },
    "InternetGatewayId" : { "Ref" : "InternetGateway" }
  }
},

"EC2Host" : {
  "Type" : "AWS::EC2::Instance",
  "DependsOn" : "GatewayToInternet",
    "Properties" : {
      "InstanceType" : { "Ref" : "EC2InstanceType" },
      "KeyName"  : { "Ref" : "KeyName" },
      "ImageId"  : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" },
       { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "EC2InstanceType"
 }, "Arch" ] } ] },
      "NetworkInterfaces" : [{
        "GroupSet"                 : [{ "Ref" : "EC2SecurityGroup" }],
        "AssociatePublicIpAddress" : "true",
        "DeviceIndex"              : "0",
        "DeleteOnTermination"      : "true",
        "SubnetId"                 : { "Ref" : "PublicSubnet" }
      }]
    }
}
```

Metadata 属性

Metadata 属性能够将结构化数据与资源相关联。通过为资源添加 Metadata 属性，您可以将 JSON 格式的数据添加到资源声明中。此外，您还可以使用 Metadata 属性中的内部函数（如 GetAtt (p. 502) 和 Ref (p. 508)）、参数和虚拟参数来添加解释的值。

Note

AWS CloudFormation 不验证 Metadata 属性中的 JSON。

您可以通过 AWS 命令 `aws cloudformation describe-stack-resource` 或 DescribeStackResource 操作来检索此数据。

示例

以下模板包含一个具有 Metadata 属性的 Amazon S3 存储桶资源。

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MyS3Bucket" : {
         "Type" : "AWS::S3::Bucket",
         "Metadata" : { "Object1" : "Location1",  "Object2" : "Location2" }
      }
   }
}
```

UpdatePolicy 属性

您可通过 UpdatePolicy 属性指定 AWS CloudFormation 如何处理某个特定资源的滚动更新。

Note

AWS::AutoScaling::AutoScalingGroup (p. 219) 是目前唯一一个支持更新策略的 AWS CloudFormation 资源。对于 Auto Scaling 组，更新策略是由启动配置或自动扩展组的子网组成员关系变化所调用的。

语法

```
"UpdatePolicy" : {
   "AutoScalingRollingUpdate" : {
      "MaxBatchSize" : String,
      "MinInstancesInService" : String,
      "PauseTime" : String
   }
}
```

UpdatePolicy 选项

UpdatePolicy 包含一个 AutoScalingRollingUpdate 嵌入式对象，该对象指定以下策略设置：

MaxBatchSize
在给定时间内被终止的最多实例数量。

Type: String.

Required: No. 默认值为一（“1”）。

MinInstancesInService
终止废弃实例时，在 Auto Scaling 组内必须有最少实例数处于运行中。

Type: String.

Required: No. 默认值为零（“0”）。

PauseTime
在对资源进行更改（例如，在增加或终止 Auto Scaling 组中的实例时添加或删除自动扩展实例）之前需要暂停的时长。

值必须为 ISO8601 持续时间格式，其形式如下：“PT#H#M#S”，其中每个 # 分别代表小时数、分钟数和/或秒数。暂停时间可指定的最大时长为一小时 ("PT1H")。

Type: String.

Required: No. 默认值为零秒（“PT0S”）。

示例

下面的例子介绍了如何为 Auto Scaling 组添加或更新策略。

```
"ASG1" : {
   "UpdatePolicy" : {
      "AutoScalingRollingUpdate" : {
         "MinInstancesInService" : "1",
         "MaxBatchSize" : "1",
         "PauseTime" : "PT12M5S"
      }
   },
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : { "Ref" : "AWS::Region" } },
      "LaunchConfigurationName" : { "Ref" : "ASLC" },
      "MaxSize" : "3",
      "MinSize" : "1"
   }
}
```

固有功能参考

AWS CloudFormation 提供多个内置函数以帮助您管理堆栈。

Note

您只能在模板的特定部分使用内部函数。目前，您可以在资源属性、元数据属性和更新策略属性中使用内部函数。

Topics

Fn::Base64

内部函数 `Fn::Base64` 返回输入字符串的 Base64 表示形式。此函数通常用于通过 UserData 属性将编码数据传递给 Amazon EC2 实例。

声明

{ "Fn::Base64" :*valueToEncode* }

参数

valueToEncode
　　您想转换成 Base64 的字符串值。

返回值：

用 Base64 表示方法的原始字符串。

示例

```
        { "Fn::Base64" : "AWS CloudFormation" }
```

另请参阅

- 固有功能参考 (p. 490)

条件函数

Abstract

使用内部函数可按条件创建堆栈资源。

您可以使用内部函数（如 `Fn::If`、`Fn::Equals` 和 `Fn::Not`）按条件创建堆栈资源。这些条件根据您在创建或更新堆栈时声明的输入参数进行计算。定义所有条件后，您可以在模板的资源和输出部分将它们与资源或资源属性关联起来。

所有条件（`Fn::If` 条件除外）都是在模板的条件部分中定义的。您可以在模板 Resources 和 Outputs 部分的元数据属性、更新策略属性和属性值中使用 `Fn::If` 条件。

如果需要重新使用模板创建一个类似的堆栈（如模拟生产堆栈的测试堆栈），可能会使用条件。在模板中，您可以添加 `EnvironmentType` 输入参数，它接受 **`prod`** 或 **`test`** 作为输入。对于生产环境，您可能需要 Amazon EC2 实例包括更多功能。使用条件，您可以定义对每个环境类型创建哪些资源以及如何配置它们。

有关条件部分的更多信息，请参阅条件声明 (p. 107)。

Note

您只能从模板的参数和映射部分引用其他条件和值。例如，您不能在条件中引用资源的逻辑 ID，但可以从输入参数引用值。

Topics

引用条件

在一个条件中引用另一个条件或将条件与资源关联时，要使用 `Condition:` 关键字。对于 `Fn::If` 函数，只需指定条件名称。

下面的代码段来自模板的条件部分。`MyAndCondition` 条件引用 `SomeOtherCondition` 条件：

```
"MyAndCondition": {
   "Fn::And": [
      {"Fn::Equals": ["sg-mysggroup", {"Ref": "ASecurityGroup"}]},
      {"Condition": "SomeOtherCondition"}
   ]
}
```

下面的代码段来自模板的资源部分，显示如何将资源与条件相关联。`NewVolume` 资源与 `CreateProdResources` 条件关联。

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
  "Condition" : "CreateProdResources",
  "Properties" : {
     "Size" : "100",
     "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone"
 ]}
}
```

下面的代码段显示如何使用 `Fn::If` 来按条件为 `NewVolume` 资源指定资源属性值。如果 `CreateLargeSize` 条件为 true，则 AWS CloudFormation 将卷大小设置为 100。如果条件为 false，则 AWS CloudFormation 将卷大小设置为 10。

```
"NewVolume" : {
  "Type" : "AWS::EC2::Volume",
  "Properties" : {
    "Size" : {
      "Fn::If" : [
        "CreateLargeSize",
        {"Ref" : "100"},
        {"Ref" : "10"}
      ]},
    "AvailabilityZone" : { "Fn::GetAtt" : [ "Ec2Instance", "AvailabilityZone"
]}
  },
  "DeletionPolicy" : "Snapshot"
}
```

Fn::And

如果所有指定条件计算为 true，则返回 `true`，如果任意条件计算为 false，则返回 `false`。`Fn::And` 用作 AND 运算符。您最少可以包含两个条件，最多可以包含 10 个条件。

声明

```
"Fn::And": [{condition}, {...}]
```

参数

`condition`
    计算为 `true` 或 `false` 的条件。

示例

当引用的安全组名称等于 `sg-mysggroup` 并且 `SomeOtherCondition` 计算为 true 时，下面的 `MyAndCondition` 计算为 true：

```
"MyAndCondition": {
   "Fn::And": [
      {"Fn::Equals": ["sg-mysggroup", {"Ref": "ASecurityGroup"}]},
      {"Condition": "SomeOtherCondition"}
   ]
}
```

Fn::Equals

比较两个值是否相等。如果两个值相等，则返回 `true`，如果不等，则返回 `false`。

声明

```
"Fn::Equals" : ["value_1", "value_2"]
```

参数

`value`
    要比较的任意类型的值。

示例

如果 `EnvironmentType` 参数的值等于 `prod`，则下面的 `UseProdCondition` 条件计算为 true：

```
"UseProdCondition" : {
   "Fn::Equals": [
      {"Ref": "EnvironmentType"},
      "prod"
   ]
}
```

Fn::If

如果指定的条件计算为 `true`，则返回一个值，如果指定的条件计算为 `false`，则返回另一个值。当前，AWS CloudFormation 在模板 Resources 部分和 Outputs 部分的元数据属性、更新策略属性和属性值中支持 `Fn::If` 内部函数。您可以使用 `AWS::NoValue` 虚拟参数作为返回值来删除相应的属性。

声明

```
"Fn::If": [condition_name, value_if_true, value_if_false]
```

参数

`condition_name`
: 条件部分中对条件的引用。使用条件名称引用它。

`value_if_true`
: 当指定的条件计算为 `true` 时要返回的值。

`value_if_false`
: 当指定的条件计算为 `false` 时要返回的值。

示例

下面的代码段在 Amazon EC2 资源的 `SecurityGroups` 属性中使用了一个 `Fn::If` 函数。如果 `CreateNewSecurityGroup` 条件计算为 true，则 AWS CloudFormation 使用 `NewSecurityGroup` 的引用值指定 `SecurityGroups` 属性；否则，AWS CloudFormation 使用 `ExistingSecurityGroup` 的引用值。

```
"SecurityGroups" : [{
  "Fn::If" : [
    "CreateNewSecurityGroup",
    {"Ref" : "NewSecurityGroup"},
    {"Ref" : "ExistingSecurityGroup"}
  ]
}]
```

在模板的 Output 部分中，您可以使用 `Fn::If` 函数按条件输出信息。在以下代码段中，如果 `CreateNewSecurityGroup` 条件的计算结果为 true，则 AWS CloudFormation 输出 `NewSecurityGroup` 资源的安全组 ID。如果条件为 false，则 AWS CloudFormation 输出 `ExistingSecurityGroup` 资源的安全组 ID。

```
"Outputs" : {
  "SecurityGroupId" : {
    "Description" : "Group ID of the security group used.",
    "Value" : {
      "Fn::If" : [
        "CreateNewSecurityGroup",
        {"Ref" : "NewSecurityGroup"},
        {"Ref" : "ExistingSecurityGroup"}
      ]
    }
  }
}
```

下面的代码段在 Fn::If 函数中使用 AWS::NoValue 虚拟参数。仅当提供了快照 ID 时，该条件才对 Amazon RDS 数据库实例使用快照。如果 UseDBSnapshot 条件计算为 true，则 AWS CloudFormation 对 DBSnapshotIdentifier 属性使用 DBSnapshotName 参数值。如果条件计算为 false，则 AWS CloudFormation 删除 DBSnapshotIdentifier 属性。

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.m1.small",
    "Engine" : "MySQL",
    "EngineVersion" : "5.5",
    "MasterUsername" : { "Ref" : "DBUser" },
    "MasterUserPassword" : { "Ref" : "DBPassword" },
    "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
    "DBSnapshotIdentifier" : {
      "Fn::If" : [
        "UseDBSnapshot",
        {"Ref" : "DBSnapshotName"},
        {"Ref" : "AWS::NoValue"}
      ]
    }
  }
}
```

以下代码段仅当 RollingUpdates 条件计算为 true 时，才提供 Auto Scaling 更新策略。如果条件计算为 false，则 AWS CloudFormation 删除 AutoScalingRollingUpdate 更新策略。

```
"UpdatePolicy": {
  "AutoScalingRollingUpdate": {
    "Fn::If": [
      "RollingUpdates",
      {
        "MaxBatchSize": "2",
        "MinInstancesInService": "2",
        "PauseTime": "PT0M30S"
      },
      {
        "Ref" : "AWS::NoValue"
      }
    ]
  }
}
```

要查看更多示例，请参阅示例模板 (p. 496)。

Fn::Not

对计算为 false 的条件返回 true，对计算为 true 的条件返回 false。Fn::Not 用作 NOT 运算符。

声明

```
"Fn::Not": [{condition}]
```

参数

`condition`
    计算为 `true` 或 `false` 的条件（如 `Fn::Equals`）。

示例

当 `EnvironmentType` 参数的值不等于 `prod` 时，下面的 `EnvCondition` 条件计算为 true：

```
"MyNotCondition" : {
   "Fn::Not" : [{
      "Fn::Equals" : [
         {"Ref" : "EnvironmentType"},
         "prod"
      ]
   }]
}
```

Fn::Or

如果任意一个指定条件计算为 true，则返回 `true`，如果所有条件都计算为 false，则返回 `false`。`Fn::Or` 用作 OR 运算符。您最少可以包含两个条件，最多可以包含 10 个条件。

声明

```
"Fn::Or": [{condition}, {...}]
```

参数

`condition`
    计算为 `true` 或 `false` 的条件。

示例

如果引用的安全组名称等于 `sg-mysggroup` 或者 `SomeOtherCondition` 计算为 true，则下面的 `MyOrCondition` 计算为 true：

```
"MyOrCondition" : {
   "Fn::Or" : [
      {"Fn::Equals" : ["sg-mysggroup", {"Ref" : "ASecurityGroup"}]},
      {"Condition" : "SomeOtherCondition"}
   ]
}
```

示例模板

Abstract

使用这些示例模板作为起点，创建如下所示的堆栈并根据需要进行自定义。

按条件为生产、开发或测试堆栈创建资源

在某些情况下，您可能需要创建类似但略有不同的堆栈。例如，您可能有一个用于生产应用程序的模板。您需要创建相同的生产堆栈来用于开发或测试。但是，对于开发和测试，您可能不需要生产级堆栈中包含的所有额外容量。您可以使用环境类型输入参数按条件创建特定于生产、开发或测试的堆栈资源，如下例所示：

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Mappings" : {
    "RegionMap" : {
      "us-east-1"      : { "AMI" : "ami-aecd60c7"},
      "us-west-1"      : { "AMI" : "ami-734c6936"},
      "us-west-2"      : { "AMI" : "ami-48da5578"},
      "eu-west-1"      : { "AMI" : "ami-6d555119"},
      "sa-east-1"      : { "AMI" : "ami-fe36e8e3"},
      "ap-southeast-1" : { "AMI" : "ami-3c0b4a6e"},
      "ap-southeast-2" : { "AMI" : "ami-bd990e87"},
      "ap-northeast-1" : { "AMI" : "ami-2819aa29"}
    }
  },

  "Parameters" : {
    "EnvType" : {
      "Description" : "Environment type.",
      "Default" : "test",
      "Type" : "String",
      "AllowedValues" : ["prod", "dev", "test"],
      "ConstraintDescription" : "must specify prod, dev, or test."
    }
  },

  "Conditions" : {
    "CreateProdResources" : {"Fn::Equals" : [{"Ref" : "EnvType"}, "prod"]},
    "CreateDevResources" : {"Fn::Equals" : [{"Ref" : "EnvType"}, "dev"]}
  },

  "Resources" : {
    "EC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
}, "AMI" ]},
        "InstanceType" : { "Fn::If" : [
          "CreateProdResources",
          "c1.xlarge",
          {"Fn::If" : [
            "CreateDevResources",
            "m1.large",
            "m1.small"
          ]}
        ]}
      }
    },

    "MountPoint" : {
      "Type" : "AWS::EC2::VolumeAttachment",
```

```
      "Condition" : "CreateProdResources",
      "Properties" : {
        "InstanceId" : { "Ref" : "EC2Instance" },
        "VolumeId"  : { "Ref" : "NewVolume" },
        "Device" : "/dev/sdh"
      }
    },

    "NewVolume" : {
      "Type" : "AWS::EC2::Volume",
      "Condition" : "CreateProdResources",
      "Properties" : {
        "Size" : "100",
       "AvailabilityZone" : { "Fn::GetAtt" : [ "EC2Instance", "AvailabilityZone"
 ]}
      }
    }
  }
}
```

您可以为 EnvType 参数指定 prod、dev 或 test。对于每种环境类型，模板都指定一个不同的实例类型。实例类型范围可以从大型计算优化实例类型到小型通用实例类型。为了按条件指定实例类型，该模板在模板的条件部分定义了两个条件：CreateProdResources，它在 EnvType 参数值等于 prod 时计算为 true，还有 CreateDevResources，它在参数值等于 dev 时计算为 true。

在 InstanceType 属性中，该模板嵌套了两个 Fn::If 内部函数来确定使用哪个实例类型。如果 CreateProdResources 条件为 true，则实例类型为 c1.xlarge。如果条件为 false，则计算 CreateDevResources 条件。如果 CreateDevResources 条件为 true，则实例类型为 m1.large，否则实例类型为 m1.small。

除实例类型之外，生产环境还向实例创建并附加一个 Amazon EC2 卷。MountPoint 和 NewVolume 资源与 CreateProdResources 条件相关联，目的是仅当条件计算为 true 时才创建资源。

按条件分配资源属性

在此示例中，您可以从快照创建 Amazon RDS 数据库实例。如果指定 DBSnapshotName 参数，AWS CloudFormation 在创建数据库实例时将使用该参数值作为快照名称。如果您保留默认值（空字符串），AWS CloudFormation 将删除 DBSnapshotIdentifier 属性并从头创建数据库实例。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Parameters": {
    "DBUser": {
      "NoEcho": "true",
      "Description" : "The database admin account username",
      "Type": "String",
      "MinLength": "1",
      "MaxLength": "16",
      "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*",
      "ConstraintDescription" : "must begin with a letter and contain only al
phanumeric characters."
    },
    "DBPassword": {
      "NoEcho": "true",
      "Description" : "The database admin account password",
      "Type": "String",
```

```
      "MinLength": "1",
      "MaxLength": "41",
      "AllowedPattern" : "[a-zA-Z0-9]*",
      "ConstraintDescription" : "must contain only alphanumeric characters."
    },
    "DBSnapshotName": {
      "Description": "The name of a DB snapshot (optional)",
      "Default": "",
      "Type": "String"
    }
  },

  "Conditions": {
    "UseDBSnapshot": {"Fn::Not": [{"Fn::Equals" : [{"Ref" : "DBSnapshotName"},
""]}]}
  },

  "Resources" : {
    "MyDB" : {
      "Type" : "AWS::RDS::DBInstance",
      "Properties" : {
        "AllocatedStorage" : "5",
        "DBInstanceClass" : "db.m1.small",
        "Engine" : "MySQL",
        "EngineVersion" : "5.5",
        "MasterUsername" : { "Ref" : "DBUser" },
        "MasterUserPassword" : { "Ref" : "DBPassword" },
        "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
        "DBSnapshotIdentifier" : {
          "Fn::If" : [
            "UseDBSnapshot",
            {"Ref" : "DBSnapshotName"},
            {"Ref" : "AWS::NoValue"}
          ]
        }
      }
    },

    "MyRDSParamGroup" : {
      "Type": "AWS::RDS::DBParameterGroup",
      "Properties" : {
        "Family" : "MySQL5.5",
        "Description" : "CloudFormation Sample Database Parameter Group",
          "Parameters" : {
            "autocommit" : "1" ,
            "general_log" : "1",
            "old_passwords" : "0"
        }
      }
    }
  }
}
```

仅当 `DBSnapshotName` 不是空字符串时，`UseDBSnapshot` 条件才计算为 true。如果 `UseDBSnapshot` 条件计算为 true，则 AWS CloudFormation 对 `DBSnapshotIdentifier` 属性使用 `DBSnapshotName` 参数值。如果条件计算为 false，则 AWS CloudFormation 删除 `DBSnapshotIdentifier` 属性。用作返回值时，`AWS::NoValue` 虚拟参数会删除相应的资源属性。

按条件使用现有资源

在此示例中，您可以使用已创建的 Amazon EC2 安全组或创建新的安全组，这是在模板中指定的。对于 ExistingSecurityGroup 参数，您可以指定 default 安全组名称或 NONE。如果指定 default，AWS CloudFormation 将使用已经创建的名为 default 的安全组。如果指定 NONE，AWS CloudFormation 会创建模板中定义的安全组。

```
{
  "Parameters" : {
    "ExistingSecurityGroup" : {
      "Description" : "An existing security group ID (optional).",
      "Default" : "NONE",
      "Type" : "String",
      "AllowedValues" : ["default", "NONE"]
    }
  },

  "Conditions" : {
    "CreateNewSecurityGroup" : {"Fn::Equals" : [{"Ref" : "ExistingSecurity
Group"}, "NONE"] }
  },

  "Resources" : {
    "MyInstance" : {
      "Type" : "AWS::EC2::Instance",
        "Properties" : {
          "ImageId" : "ami-1b814f72",
          "SecurityGroups" : [{
            "Fn::If" : [
              "CreateNewSecurityGroup",
              {"Ref" : "NewSecurityGroup"},
              {"Ref" : "ExistingSecurityGroup"}
            ]
          }]
        }
    },

    "NewSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Condition" : "CreateNewSecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "80",
          "ToPort" : "80",
          "CidrIp" : "0.0.0.0/0"
        } ]
      }
    }
  },

  "Outputs" : {
    "SecurityGroupId" : {
      "Description" : "Group ID of the security group used.",
      "Value" : {
        "Fn::If" : [
          "CreateNewSecurityGroup",
```

```
          {"Ref" : "NewSecurityGroup"},
          {"Ref" : "ExistingSecurityGroup"}
        ]
      }
    }
  }
}
```

为了确定是否创建 NewSecurityGroup 资源，该资源与 CreateNewSecurityGroup 条件关联。仅当条件为 true（当 ExistingSecurityGroup 参数等于 NONE）时，才会创建资源。

在 SecurityGroups 属性中，模板使用 Fn::If 内部函数确定要使用的安全组。如果 CreateNewSecurityGroup 条件计算为 true，则安全组属性引用 NewSecurityGroup 资源。如果 CreateNewSecurityGroup 条件计算为 false，则安全组属性引用 ExistingSecurityGroup 参数（default 安全组）。

最后，模板按条件输出安全组 ID。如果 CreateNewSecurityGroup 条件的计算结果为 true，则 AWS CloudFormation 输出 NewSecurityGroup 资源的安全组 ID。如果条件为 false，则 AWS CloudFormation 输出 ExistingSecurityGroup 资源的安全组 ID。

Fn::FindInMap

内部函数 Fn::FindInMap 将与键对应的值返回到 Mappings 部分声明的双层映射中。

声明

"Fn::FindInMap" :["*MapName*", "*TopLevelKey*", "*SecondLevelKey*"]

参数

MapName
: Mappings 部分中所声明映射的逻辑名称，包含键和值。

TopLevelKey
: 顶层键名称。其值是一个键/值对列表。

SecondLevelKey
: 第二层键名称，该项设置为分配给 *TopLevelKey* 的列表中的一个键。

返回值：

分配给 *SecondLevelKey* 的值。

示例

以下示例显示了对于具有包含单个映射 RegionMap（该映射将 AMI 与 AWS 区域相关联）的 Mappings 部分的模板，如何使用 Fn::FindInMap。

- 该映射具有 5 个对应多个 AWS 区域的顶层键。
- 每个顶层键都分配有一个列表，其中包含两个与 AMI 架构对应的第二层键（"32" 和 "64"）。
- 每个第二层密钥都分配有一个适当的 AMI 名称。

```
{
  ...
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : { "32" : "ami-6411e20d", "64" : "ami-7a11e213" },
      "us-west-1" : { "32" : "ami-c9c7978c", "64" : "ami-cfc7978a" },
      "eu-west-1" : { "32" : "ami-37c2f643", "64" : "ami-31c2f645" },
      "ap-southeast-1" : { "32" : "ami-66f28c34", "64" : "ami-60f28c32" },
      "ap-northeast-1" : { "32" : "ami-9c03a89d", "64" : "ami-a003a8a1" }
    }
  },

  "Resources" : {
     "myEC2Instance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
          "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region"
 }, "32"]},
           "InstanceType" : "m1.small"
        }
     }
 }
}
```

此示例模板包含由 `FindInMap` 函数设置其 `ImageId` 属性的 `AWS::EC2::Instance` 资源。

- 可按个人喜好为映射设置 *`MapName`*，在本例中为 `"RegionMap"`。
- *`TopLevelKey`* 设置为创建堆栈所在的区域，该值通过使用 `"AWS::Region"` 虚拟参数确定。
- *`SecondLevelKey`* 设置为所需的架构，在本例中为 `"32"`。

`FindInMap` 返回分配给 `FindInMap` 的 AMI。对于 us-east-1 中的 32 位实例，`FindInMap` 将返回 `"ami-6411e20d"`。

Fn::GetAtt

Abstract

使用 Fn::GetAtt 内部函数从模板中的资源返回属性的值。

内部函数 `Fn::GetAtt` 返回模板中的资源的属性值。

声明

"Fn::GetAtt" :["*`logicalNameOfResource`*", "*`attributeName`*"]

参数

logicalNameOfResource
: 包含您想要的属性之资源的逻辑名称。

attributeName
: 您想要获得其值的资源特定属性名称。有关该资源类型之可用属性的详细信息，请参阅资源的引用页面。

返回值

属性值。

示例

此示例返回一个字符串，其中包含逻辑名称为 *MyLB* 的负载均衡器的 DNS 名称。

```
"Fn::GetAtt" : [ "MyLB" , "DNSName" ]
```

属性

您可以使用 `Fn::GetAtt` 检索以下属性：

资源类型名称	属性	说明
AWS::CloudFormation::WaitCondition (p. 252)	Data	JSON 格式的字符串包含指定等待条件的等待条件信号中的 UniqueId 和 Data 值。有关等待条件信号的更多信息，请参阅等待条件发送 JSON 格式 (p. 191)。 具有 2 个信号的等待条件示例： `{"Signal1":"Step 1 complete.","Signal2":"Step 2 complete."}`
AWS::CloudFormation::Stack (p. 250)	Outputs.*NestedStackOutputName*	您指定的嵌套堆栈的输出值，其中 *NestedStackOutputName* 是该输出值的名称。
AWS::CloudFront::Distribution (p. 256)	DomainName	示例：`http://d2fadu0nynjpfn.cloudfront.net/`
AWS::EC2::EIP (p. 269)	AllocationId	AWS 分配的 ID，用于表示与 Amazon VPC 配合使用的地址的分配。该值仅针对 VPC 弹性 IP 地址返回。 示例：`eipalloc-5723d13e`
AWS::EC2::Instance (p. 272)	可用区	启动您指定的实例时所在的可用区。 示例： us-east-1b
AWS::EC2::Instance (p. 272)	PrivateDnsName	您指定的实例的私有 DNS 名称。 示例：`ip-10-24-34-0.ec2.internal`
AWS::EC2::Instance (p. 272)	PublicDnsName	您指定的实例的公有 DNS 名称。 示例：`ec2-107-20-50-45.compute-1.amazonaws.com`
AWS::EC2::Instance (p. 272)	PrivateIp	您指定的实例的私有 IP 地址。 示例：`10.24.34.0`
AWS::EC2::Instance (p. 272)	PublicIp	您指定的实例的公有 IP 地址。 示例：`192.0.2.0`

资源类型名称	属性	说明
AWS::EC2::NetworkInterface (p. 283)	PrimaryPrivateIpAddress	您指定的网络接口的主要私有 IP 地址。 示例：`10.0.0.192`
AWS::EC2::NetworkInterface (p. 283)	SecondaryPrivateIpAddresses	您指定的网络接口的辅助私有 IP 地址。 例如：`["10.0.0.161", "10.0.0.162", "10.0.0.163"]`
AWS::EC2::SecurityGroup (p. 292)	GroupId	指定安全组的组 ID。 例如：`sg-94b3a1f6`
AWS::EC2::VPCNetworkAcl (p. 302)	AssociationId	与子网关联的网络 ACL 的关联 Id
AWS::ElastiCache::CacheCluster (p. 320)	ConfigurationEndpoint.Address	缓存集群的配置终端节点的 DNS 地址。
AWS::ElastiCache::CacheCluster (p. 320)	ConfigurationEndpoint.Port	缓存集群的配置终端节点的端口编号。
AWS::ElasticBeanstalk::Environment (p. 333)	EndpointURL	指向此环境的负载均衡器的 URL。 示例： awseb-myst-myen-132MQC4KRLAMD-1371280482.us-east-1.elb.amazonaws.com
AWS::ElasticLoadBalancing::LoadBalancer (p. 337)	CanonicalHostedZoneName	与负载均衡器关联的 Route 53 托管区域的名称。 示例： mystack-myelb-15HMAB89ZON57-1013119603.us-east-1.elb.amazonaws.com
AWS::ElasticLoadBalancing::LoadBalancer (p. 337)	CanonicalHostedZoneNameID	与负载均衡器关联的 Route 53 托管区域名称的 ID。 示例：`Z3DZXE0Q79N41H`
AWS::ElasticLoadBalancing::LoadBalancer (p. 337)	DNSName	负载均衡器的 DNS 名称。 示例： mystack-myelb-15HMAB89ZON57-1013119603.us-east-1.elb.amazonaws.com
AWS::ElasticLoadBalancing::LoadBalancer (p. 337)	SourceSecurityGroup.GroupName	可作为负载均衡器后端 Amazon EC2 应用程序实例入站规则的一部分使用的安全组。 示例：`amazon-elb`
AWS::ElasticLoadBalancing::LoadBalancer (p. 337)	SourceSecurityGroup.OwnerAlias	源安全组的所有者。 示例：`amazon-elb-sg`
AWS::IAM::AccessKey (p. 343)	SecretAccessKey	指定访问密钥的秘密访问密钥。 示例：`wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY`
AWS::IAM::Group (p. 345)	Arn	例如： arn:aws:iam::123456789012:group/mystack-mygroup-1DZETITOWEKVO
AWS::IAM::User (p. 355)	Arn	例如： arn:aws:iam::123456789012:group/mystack-myuser-1CCXAFG2H2U4D

资源类型名称	属性	说明
AWS::Redshift::Cluster (p. 371)	Endpoint.Address	集群的连接终端节点。 例如： examplecluster.cg034hpkmmjt.us-east-1.redshift.amazonaws.com
AWS::Redshift::Cluster (p. 371)	Endpoint.Port	集群的连接终端节点。 例如：5439
AWS::RDS::DBInstance (p. 381)	Endpoint.Address	数据库的连接终端节点。 示例： mystack-mydb-1apw1j4phylrk.cg034hpkmmjt.us-east-1.rds.amazonaws.com
AWS::RDS::DBInstance (p. 381)	Endpoint.Port	数据库实例接受连接的端口编号。 示例：3306
AWS::S3::Bucket (p. 402)	DomainName	指定存储桶的 DNS 名称。 示例： mystack-mybucket-kdwwxmddtr2g.s3-us-east-1.amazonaws.com
AWS::S3::Bucket (p. 402)	WebsiteURL	指定存储桶的 Amazon S3 网站终端节点。 例如： http://mystack-mybucket-kdwwxmddtr2g.s3-website-us-east-1.amazonaws.com/
AWS::SNS::Topic (p. 411)	主题名称	Amazon SNS 主题的名称。 例如：my-sns-topic
AWS::SQS::Queue (p. 414)	Arn	指定队列的 ARN。 例如： arn:aws:sqs:us-east-1:123456789012:mystack-myqueue-15PG5C2FC1CW8
AWS::SQS::Queue (p. 414)	队列名称	Amazon SQS 队列的名称。 例如：mystack-myqueue-1VF9BKQH5BJVI

Fn::GetAZs

内部函数 Fn::GetAZs 返回列出指定区域的所有可用区的数组。

由于客户可访问不同的可用区，模板作者可以使用内部函数 Fn::GetAZs 编写适合调用用户访问的模板。这一操作让你不需要为特定区域硬编码可用区的完整列表。

声明

"Fn::GetAZs" :"*region*"

参数

区域
    您想获得其可用区的区域之名称。

您可以使用 *AWS::Region* 虚拟参数指定创建堆栈所在的区域。指定空字符串等效于指定 *AWS::Region*。

返回值

该区域可用区列表。

示例

```
{ "Fn::GetAZs" : "" }
```

```
{ "Fn::GetAZs" : "us-east-1" }
```

在前面两个示例中，AWS CloudFormation 在假设用户已在 us-east-1 区域创建堆栈的情况下对以下数组计算 Fn::GetAZs：

```
[ "us-east-1a", "us-east-1b", "us-east-1c" ]
```

Fn::Join

内部函数 `Fn::Join` 将一组值追加到单个值中，这些值用指定的分隔符分隔。如果分隔符为空字符串，则该组值不通过分隔符被连接在一起。

声明

"Fn::Join" :["*delimiter*", [*comma-delimited list of values*]]

参数

分隔符
: 您希望发生在片断之间的值。分隔符只会发生在片断之间。它不会终止终值。

listOfValues
: 您想组合的值之列表。

返回值

组合的字符串。

示例

```
"Fn::Join" : [ ":", [ "a", "b", "c" ] ]
```

此示例返回："a:b:c".

Fn::Select

Abstract

使用 Fn::Select 内部函数按索引返回对象列表中的单个对象。

内部函数 `Fn::Select` 按索引返回对象列表中的单个对象。

Important

Fn::Select 不会检查空值，或检查索引是否超出阵列边界。两种条件都可导致堆栈错误，所以您应该确保您选择的索引有效且列表中包含非空值。

声明

{ "Fn::Select" :[*index*, *listOfObjects*] }

参数

索引
: 待检索对象的索引。索引必须是零到 N-1 之间的某个值，其中 N 代表阵列中元素的数量。

listOfObjects
: 选择对象的列表。该列表不得为空，且不能包含空项目。

返回值

选定的对象。

示例

```
{ "Fn::Select" : [ "1", [ "apples", "grapes", "oranges", "mangoes" ] ] }
```

此示例返回："grapes".

逗号分隔列表参数类型

您可以使用 `Fn::Select` 从 `CommaDelimitedList` 参数选择一个对象。您可以使用 `CommaDelimitedList` 参数合并相关参数的值，这样可减少模板中的参数总数。例如，以下参数指定包含三个 CIDR 块的逗号分隔列表：

```
"Parameters" : {
  "DbSubnetIpBlocks": {
    "Description": "Comma-delimited list of three CIDR blocks",
    "Type": "CommaDelimitedList",
      "Default": "10.0.48.0/24, 10.0.112.0/24, 10.0.176.0/24"
  }
}
```

要指定三个 CIDR 块之一，请在相同模板的 Resources 部分中使用 `Fn::Select`，如以下示例代码段所示：

```
"Subnet0": {
  "Type": "AWS::EC2::Subnet",
    "Properties": {
      "VpcId": { "Ref": "VPC" },
      "CidrBlock": { "Fn::Select" : [ "0", {"Ref": "DbSubnetIpBlocks"} ] }
```

```
    }
},
```

Ref

Abstract

使用 Ref 内部函数返回有关指定参数或资源的信息。

内部函数 `Ref` 返回指定的*参数*或*资源*的值。

- 它在您指定参数逻辑名称时返回参数值。
- 在您指定资源的逻辑名称时，它会返回一个您一般用于引用该资源的值。

当您正在模板中声明资源并且您需要用名称指定另一个模板资源时，您可以使用 `Ref` 引用那个其他资源。通常，`Ref` 返回资源的名称。例如，对 AWS::AutoScaling::AutoScalingGroup (p. 219) 的引用返回该 Auto Scaling 组资源的名称。

对于某些资源，会返回在资源环境中有另一种重要意义的标识符。例如，AWS::EC2::EIP (p. 269) 资源返回 IP 地址，而 AWS::EC2::Instance (p. 272) 返回实例 ID。

本主题的底部有一个表格列出了针对很多公共资源类型返回的值。有关特定资源或属性的 `Ref` 返回值的更多信息，可在该资源或属性的文档中找到。

Tip

您还可以使用 `Ref` 将值添加到 Output 消息中。

声明

"Ref" :"*logicalName*"

参数

logicalName
: 您想解除引用的资源或参数之逻辑名称。

返回值

MyInputParameter 参数的值。

示例

弹性 IP 地址的以下资源声明需要 EC2 实例的实例 ID，并使用 `Ref` 函数指定 MyEC2Instance 资源的实例 ID：

```
"MyEIP" : {
   "Type" : "AWS::EC2::EIP",
   "Properties" : {
      "InstanceId" : { "Ref" : "MyEC2Instance" }
```

```
    }
}
```

资源返回示例

此部分列出了 `Ref` 所返回的特定 AWS CloudFormation 资源的示例值。有关特定资源或属性的 `Ref` 返回值的更多信息，请参阅该资源或属性的文档。

资源类型	参考值	示例返回值
AWS::AutoScaling::AutoScalingGroup (p. 219)	名称	mystack-myasgroup-NT5EUXTNTXXD
AWS::AutoScaling::LaunchConfiguration (p. 224)	名称	mystack-mylaunchconfig-1DDYF1E3B3I
AWS::AutoScaling::ScalingPolicy (p. 230)	名称	mystack-myaspolicy-1DDYF1E3B3I
AWS::AutoScaling::ScheduledAction (p. 232)	名称	mystack-myscheduledaction-NT5EUXTNTXXD
AWS::CloudFormation::Stack (p. 250)	堆栈 ID	[illegible]
AWS::CloudFormation::WaitCondition (p. 252)	名称	[illegible]
AWS::CloudFormation::WaitConditionHandle (p. 255)	等候条件信号 URL	[illegible]
AWS::CloudFront::Distribution (p. 256)	分配 ID	E27LVI50CSW06W
AWS::CloudWatch::Alarm (p. 257)	名称	mystack-myalarm-3AOHFRGOXR5T
AWS::EC2::Volume (p. 305)	卷 ID	vol-3cdd3f56
AWS::EC2::VolumeAttachment (p. 308)	名称	mystack-myvola-ERXHJITXMRLT
AWS::EC2::EIP (p. 269)	弹性 IP 地址	192.0.2.0
AWS::EC2::EIPAssociation (p. 270)	名称	mystack-myeipa-1NU3IL8LJ313N
AWS::EC2::Instance (p. 272)	实例 ID	i-636be302
AWS::EC2::SecurityGroup (p. 292)	名称或安全组 ID（对于 VPC 安全组）	mystack-mysecuritygroup-QQB406M8FISX 或 sg-94b3a1f6
AWS::EC2::SecurityGroupIngress (p. 296)	名称	mysecuritygroupingress
AWS::EC2::Subnet (p. 300)	名称	subnet-e19f0178
AWS::ElasticBeanstalk::Application (p. 329)	名称	mystack-myapplication-FM6BIXY7U8PK
AWS::ElasticBeanstalk::ApplicationVersion (p. 330)	名称	mystack-myapplicationversion-iy8ptveuxjly
AWS::ElasticBeanstalk::ConfigurationTemplate (p. 331)	名称	mystack-myconfigurationtemplate-108RPH64J195
AWS::ElasticBeanstalk::Environment (p. 333)	名称	mystack-myenv-LKGNQSFHO1DB
AWS::ElastiCache::SubnetGroup (p. 328)	名称	myCachesubnetgroup
AWS::ElasticLoadBalancing::LoadBalancer (p. 337)	名称	mystack-myelb-1WQN7BJGDB5YQ
AWS::IAM::AccessKey (p. 343)	AccessKeyId	AKIAIOSFODNN7EXAMPLE
AWS::IAM::Group (p. 345)	GroupName	mystack-mygroup-1DZETITOWEKVO
AWS::IAM::User (p. 355)	UserName	mystack-myuser-1CCXAFG2H2U4D

资源类型	参考值	示例返回值
AWS::Kinesis::Stream (p. 357)	名称	mystack-mystream-1NAOH4L1RIQ7I
AWS::OpsWorks::App (p. 358)	AWS OpsWorks 应用程序 ID	4fee5b96-0d10-4af1-bcc5-25f92e3c6acf
AWS::OpsWorks::Instance (p. 361)	AWS OpsWorks 实例 ID	aa2e9ae2-2b4b-491c-aeb6-8bf3ce9400fe
AWS::OpsWorks::Layer (p. 364)	AWS OpsWorks 层 ID	730b238b-f7c4-461d-b7c0-3feb7ef1152a
AWS::OpsWorks::Stack (p. 368)	AWS OpsWorks 堆栈 ID	5c9f04e8-370e-4bd3-ae09-a4bbcc2998bb
AWS::Redshift::Cluster (p. 371)	名称	mystack-myredshiftcluster-ranmiv3f0mad
AWS::Redshift::ClusterParameterGroup (p. 376)	名称	mysta-mypar-1AJYM1FL3WQBW
AWS::Redshift::ClusterSecurityGroup (p. 378)	名称	mystack-myredshiftclustersecuritygroup-bjy2afmhy3ee
AWS::Redshift::ClusterSubnetGroup (p. 380)	名称	mystack-myredshiftclustersubnetgroup-aq8sdq8rp71
AWS::RDS::DBInstance (p. 381)	名称	mystack-mydb-ea5ugmfvuaxg
AWS::RDS::DBSecurityGroup (p. 392)	名称	mystack-mydbsecuritygroup-1k5u5dxjb0nxs
AWS::S3::Bucket (p. 402)	名称	mystack-mys3bucket-1hbsmonr9mytq
AWS::SDB::Domain (p. 411)	名称	mystack-mysdbdomain-IVNAOZTDFVXL
AWS::SNS::Topic (p. 411)	主题 ARN	arn:aws:sns:us-east-1:123456789012:mystack-mytopic-NZJ5JSMVGFIE
AWS::SQS::Queue (p. 414)	队列 URL	https://sqs.us-east-1.amazonaws.com/123456789012/MyQueue-ZNOSZOPPZF9
虚拟参数 (p. 510)	AWS::AccountId	123456789012
虚拟参数 (p. 510)	AWS::NotificationARNs	[arn:aws:sns:us-east-1:123456789012:MyTopic]
虚拟参数 (p. 510)	AWS::NoValue	不返回值。
虚拟参数 (p. 510)	AWS::Region	us-east-1
虚拟参数 (p. 510)	AWS::StackId	arn:aws:cloudformation:us-east-1:123456789012:stack/MyStack/466df9e0-0dff-08fa-6a52-caa12b135a15
虚拟参数 (p. 510)	AWS::StackName	MyStack

虚拟参数参考

虚拟参数是 AWS CloudFormation 预定义的参数。请您不要在您的模板中声明它们。就像使用参数一样将虚拟参数用作 `Ref` 函数的参数。

例如，以下片断分配了 *`AWS::Region`* 虚拟参数的值：

```
"Outputs" {
   "MyStacksRegion" : { "Value" : { "Ref" : "AWS::Region" } }
}
```

此处所列为目前可用的 pseudo 参数。

AWS::AccountId
　　返回在其中创建堆栈的账户的 AWS 账户 ID。

AWS::NotificationARNs
返回当前堆栈的通知亚马逊资源名称 (ARN) 列表。

例如：

```
{
   "AWSTemplateFormatVersion" : "2010-09-09",
   "Resources" : {
      "MyNestedStack" : {
         "Type" : "AWS::CloudFormation::Stack",
         "Properties" : {
            "TemplateURL" : "https://my-website.com/stack-spec.json",
            "NotificationARNs" : {"Ref" : "AWS::NotificationARNs"}
         }
      }
   }
}
```

要从列表中获取单个 ARN，请使用 Fn::Select (p. 506)：

```
"myASGrpOne" : {
   "Type" : "AWS::AutoScaling::AutoScalingGroup",
   "Version" : "2009-05-15",
   "Properties" : {
      "AvailabilityZones" : [ "us-east-1a" ],
      "LaunchConfigurationName" : { "Ref" : "MyLaunchConfiguration" },
      "MinSize" : "0",
      "MaxSize" : "0",
      "NotificationConfiguration" : {
         "TopicARN" : { "Fn::Select" : [ "0", { "Ref" : "AWS::Notification
ARNs" } ] },
         "NotificationTypes" : [ "autoscaling:EC2_INSTANCE_LAUNCH", "auto
scaling:EC2_INSTANCE_LAUNCH_ERROR" ]
      }
   }
}
```

AWS::NoValue
如果在 Fn::If 内部函数中指定为返回值，则删除相应的资源属性。例如，如果需要对 Amazon RDS 数据库实例使用快照（仅当提供了快照 ID 时），您可以使用 AWS::NoValue 参数，如以下代码段所示：

```
"MyDB" : {
  "Type" : "AWS::RDS::DBInstance",
  "Properties" : {
    "AllocatedStorage" : "5",
    "DBInstanceClass" : "db.m1.small",
    "Engine" : "MySQL",
    "EngineVersion" : "5.5",
    "MasterUsername" : { "Ref" : "DBUser" },
    "MasterUserPassword" : { "Ref" : "DBPassword" },
    "DBParameterGroupName" : { "Ref" : "MyRDSParamGroup" },
    "DBSnapshotIdentifier" : {
      "Fn::If" : [
```

```
        "UseDBSnapshot",
        {"Ref" : "DBSnapshotName"},
        {"Ref" : "AWS::NoValue"}
      ]
    }
  }
}
```

如果 `UseDBSnapshot` 条件计算为 true，则 AWS CloudFormation 对 `DBSnapshotIdentifier` 属性使用 `DBSnapshotName` 参数值。如果条件计算为 false，则 AWS CloudFormation 删除 `DBSnapshotIdentifier` 属性。

AWS::Region
: 返回代表包容性资源在其中创建的 AWS 区域的字符串。

AWS::StackId
: 返回使用 `aws cloudformation create-stack` 命令指定的堆栈 ID。

AWS::StackName
: 返回使用 `aws cloudformation create-stack` 命令指定的堆栈名称。

CloudFormation 帮助程序脚本参考

Abstract

描述可以用于通过 AWS CloudFormation 安装软件和启动服务的 Python 帮助程序脚本。

Topics

AWS CloudFormation 提供了一组 Python 帮助程序脚本，您可以使用这些脚本在作为堆栈的一部分创建的 Amazon EC2 实例上安装软件和启动服务。您可以从您的模板中直接调用帮助程序脚本。该脚本可与您在同一模板中定义的资源元数据共同运行。在堆栈创建过程中，这些帮助程序脚本在 Amazon EC2 实例上运行。

可在 Amazon Linux AMI 最新版本上预先安装该帮助程序脚本。可从 Amazon Linux yum 存储库中获取帮助程序脚本，与其他 UNIX/Linux AMI 一起使用。

当前，AWS CloudFormation 提供以下帮助程序：

- cfn-init (p. 513):用于检索和解释资源元数据、安装程序包、创建文件和启动服务。
- cfn-signal (p. 515):用于发送 CloudFormation WaitCondition 信号的简单包装程序，该程序允许您将堆栈中的其他资源与准备就绪的应用程序进行同步。
- cfn-get-metadata (p. 518):包装程序脚本，旨在便于您轻松检索为资源定义的所有元数据，或者资源元数据的特定键或子树的路径。
- cfn-hup (p. 520):一个后台程序，用于检查元数据更新，并在检测出变更时，执行自定义钩子。

这些脚本按照默认方式安装在 /opt/aws/bin 中的最新版本 Amazon Linux AMI 上。针对 Amazon Linux AMI 的先前版本，可从 Amazon Linux AMI yum 存储库中获取这些脚本在，其亦可通过针对 Linux/Unix 分配的 RPM 获取。还可利用 Python for Windows 将这些脚本安装在 Microsoft Windows 上。

脚本并非以默认方式执行的。您必须纳入对执行特定帮助程序脚本的调用操作。

AWS CloudFormation 帮助程序脚本通过以下位置可用：

- Amazon Linux AMI 最新版本，包括以默认方式安装在 /opt/aws/bin 中的 AWS CloudFormation 帮助程序脚本。
- 针对 Amazon Linux AMI 的先前版本，可从 Amazon Linux AMI yum 存储库中获取 AWS 帮助程序脚本（程序包名称为：aws-cfn-bootstrap）。
- 其他格式的帮助程序亦可供使用：
  - https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-latest.amzn1.noarch.rpm
  - https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-latest.tar.gz 可通过 Python 简易安装工具安装帮助程序脚本。
  - https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-latest.zip
  - https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-latest.msi 用于在 Microsoft Windows 上进行安装。
- 可从 https://s3.amazonaws.com/cloudformation-examples/aws-cfn-bootstrap-latest.src.rpm 获取脚本源代码，这些源代码可用于 Amazon Linux AMI 之外的 Linux 分配。

Note

可用帮助程序脚本的完整列表及其用法的相关信息可在 AWS CloudFormation 工具页上找到：使用 AWS CloudFormation 启动应用程序。

cfn-init

Abstract

从 AWS::CloudFormation::Init 资源读取模板元数据并完成操作，如在 Amazon EC2 实例上安装和配置应用程序。

说明

cfn-init 帮助程序脚本读取来自 AWS::CloudFormation::Init 密钥的模板元数据并进行相应操作：

- 提取和解析来自 CloudFormation 的元数据
- 安装程序包
- 将文件写入磁盘
- 启用/禁用以及启动/停止服务

Note

如果您使用 cfn-init 来更新现有文件，则它将在同一目录下为原始文件创建一个扩展名为 .bak 的备份副本。例如，如果您更新了 */path/to/file_name*，则该操作将生成两个文件：*/path/to/file_name*.bak 包含原始文件的内容，而 */path/to/file_name* 包含更新后的内容。

有关模板元数据的信息，请参阅AWS::CloudFormation::Init (p. 241)。

Note

cfn-init 不要求提供证书，因此您不需要使用 `--access-key`、`--secret-key`、`--role` 或 `--credential-file` 选项。

语法

```
cfn-init --stack|-s stack.name.or.id \
         --resource|-r logical.resource.id \
         --region region
         --access-key access.key \
         --secret-key secret.key \
         --role rolename\
         --credential-file|-f credential.file \
         --configsets|-c config.sets \
         --url|-u service.url \
         -v
```

选项

名称	说明	必需
`-s, --stack`	堆栈名称。 *Type*: String. *默认值*：无 *示例*：`-s { "Ref" : "AWS::StackName" },`	是
`-r, --resource`	包含元数据的资源的逻辑资源 ID。 *Type*: String. *示例*：`-r WebServerHost`	是
`--region`	待派生 CloudFormation URL 的区域。 *Type*: String. *默认值*：无 *示例*：`--region ", { "Ref" : "AWS::Region" },`	否
`--access-key`	针对拥有权限的账户 AWS 访问密钥，调用 CloudFormation 上的 DescribeStackResource。 *Type*: String. 条件：证书文件参数取代此参数。	条件
`--secret-key`	AWS 密钥对应于指定 AWS 访问密钥。 *Type*: String. 条件：证书文件参数取代此参数。	条件
`--role`	IAM 角色名称。 *Type*: String. 条件：证书文件参数取代此参数。	

名称	说明	必需
`-f, --credential-file`	同时包含密钥和访问密钥的文件。 *Type*: String. 条件：证书文件参数取代 --role、--access-key 和 --secret-key 参数。	条件
`-c, --configsets`	待运行逗号分隔 configsets 列表（按顺序）。 *Type*: String. *默认值*：`default`	否
`-u, --url`	待点击 CloudFormation 服务 URL。 *Type*: String.	否
`-v`	详细的输出。这对于调试 cfn-init 无法初始化的案例非常有用。  Note 您应该打开 DisableRollback，以便调试初始化操作。您可以通过使用 CloudFormation 控制台，选择*显示高级选项*，然后将“失败时回滚”设定为“否”来完成此操作。随后您可以将 SSH 置于控制台中，读取 /var/log/cfn-init.log 上的日志。	否

示例

以下代码段与名称为 WebServer 的资源关联。

```
"/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" },
"    -r WebServer ",
"    --region ", { "Ref" : "AWS::Region" }, \n",
```

多个 AWS CloudFormation 示例模板使用 cfn-init，包括以下模板。

- LAMP：带有本地 MySQL 数据库的单一 EC2 实例
- WordPress：带有本地 MySQL 数据库的单一 EC2 实例
- Drupal：带有本地 MySQL 数据库的单一 EC2 实例

cfn-signal

Abstract

暂停堆栈创建过程。

说明

您可以利用 cfn-signal 帮助程序脚本暂停堆栈创建进程。与两个 AWS CloudFormation 资源结合使用 cfn-signal 脚本：

- AWS::CloudFormation::WaitCondition (p. 252)
- AWS::CloudFormation::WaitConditionHandle (p. 255)

有关如何在模板中使用等待条件的更多信息，请参阅在模板中创建等待条件 (p. 188)。

语法

```
cfn-signal --success|-s signal.to.send \
        --reason|-r resource.status.reason \
        --data|-d data \
        --id|-i unique.id \
        --exit-code|-e exit.code \
        waitconditionhandle.url
```

选项

名称	说明	必需
`-s, --success`	如为真实的，则信号“成功”，否则“失败”。 类型：布尔值 *默认值*：`true`	否
`-r, --reason`	如成功为错误信息，资源事件状态原因（当前仅用于故障）则将默认为“配置失败”。 *Type*: String.	否
`-d, --data`	通过 waitConditionHandle 发送回数据。默认值待留空。 *Type*: String. *默认值*：空	否
`-i, --id`	待通过 WaitConditionHandle 发送回的唯一 id。 *Type*: String. *默认值*：Amazon EC2 实例的 ID。如果无法解析该 ID，则将返回计算机的完全限定域名 (FQDN)。	否
`-e, --exit-code`	进程中出现错误代码，用于确定成功或失败。如果指定，则将忽略 `--success` 选项。 *Type*: String. *示例*：`-e $?`（适用于 Linux），`-e %ERRORCODE%`（适用于 Windows）	否

名称	说明	必需
waitconditionhandle.url	您可以用于向关联 WaitCondition 发送成功或失败信号的预签名 URL *Type*: String.	是

示例

示例 1

常规使用模式将同时使用 cfn-init 和 cfn-signal。cfn-signal 调用利用了 cfn-init 调用的返回状态（使用 $? 外壳程序构造）。如果应用程序安装失败，那么 WaitCondition 将无法创建，堆栈将回滚。

```
"MyInstance": {
   "Type": "AWS::EC2::Instance",
   "Metadata": {
      :
   },
   "Properties": {
      "ImageId" : "ami-12345678",
      "UserData" : {
         "Fn::Base64" : {
            "Fn::Join" : ["", [
               "#!/bin/bash\n",
               "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" },
               "         -r MyInstance ",
               "         --region ", { "Ref" : "AWS::Region" },
               "\n",
              "/opt/aws/bin/cfn-signal -e $? '", { "Ref" : "MyHandle" }, "'\n",

            ] ]
         }
      }
   }
},
```

示例 2

以下代码段显示了某项函数调用 cfn-signal 时的另一个常规使用模式。环绕函数中的 cfn-signal 能够使从多个位置调用 cfn-signal 变得更为简单。对于 Windows 堆栈，您必须再次对等待条件句柄 URL 进行 base64 编码。有关示例，请参阅启动 AWS CloudFormation Windows 堆栈 (p. 210)。

```
"# Helper function\n",
          "function error_exit\n",
          "{\n",
          "  /opt/aws/bin/cfn-signal -e 1 -r \"$1\" '", { "Ref" : "WaitHandle"
 }, "'\n",
          "  exit 1\n",
          "}\n",
```

以下代码段显示了两条命令中调用的 error_exit 函数。

```
"# Setup MySQL, create a user and a database\n",
"mysqladmin -u root password '", { "Ref" : "DBRootPassword" },
"' || error_exit 'Failed to initialize root password'\n",

"mysql -u root --password='", { "Ref" : "DBRootPassword" },
"' < /tmp/setup.mysql || error_exit 'Failed to initialize database'\n",
```

示例模板中的示例

若干 AWS CloudFormation 示例模板使用 cfn-signal，包括以下模板。

- LAMP：带有本地 MySQL 数据库的单一 EC2 实例
- WordPress：带有本地 MySQL 数据库的单一 EC2 实例
- Drupal：带有本地 MySQL 数据库的单一 EC2 实例

cfn-get-metadata

Abstract

从 AWS CloudFormation 提取元数据块并进行打印。

说明

您可以使用 cfn-get-metadata 帮助程序脚本从 CloudFormation 中提取元数据块，并将其打印进行标准输出。如果您指定密钥，您还可以打印元数据块的子树。但是，仅支持具有最高级别的密钥。

Note

cfn-get-metadata 不要求提供证书，因此您不需要使用 `--access-key`、`--secret-key` 或 `--credential-file` 选项。

语法

```
cfn-get-metadata --access-key access.key \
                 --secret-key secret.key \
                 --credential-file|f credential.file \
                 --key|k key \
                 --stack|-s stack.name.or.id \
                 --resource|-r logical.resource.id \
                 --url|-u service.url \
                 --region region
```

选项

名称	说明	必需
`-k, --key`	对于键/值对，返回指定的值所对应的键的名称。 *Type*: String. *示例*：对于 `{ "SampleKey1" : "Key1", "SampleKey2" : "Key2" }`，`cfn-get-metadata -k Key2` 返回 `SampleKey2`。	否
`-s, --stack`	堆栈名称。 *Type*: String. *默认值*：无 *示例*：`-s { "Ref" : "AWS::StackName" },`	是
`-r, --resource`	包含元数据的资源的逻辑资源 ID。 *Type*: String. *示例*：`-r WebServerHost`	是
`--region`	待派生 CloudFormation URL 的区域。 *Type*: String. *默认值*：无 *示例*：`--region ", { "Ref" : "AWS::Region" },`	否
`--access-key`	针对拥有权限的账户 AWS 访问密钥，调用 CloudFormation 上的 DescribeStackResource。 *Type*: String. 条件：证书文件参数取代此参数。	条件
`--secret-key`	AWS 密钥对应于指定 AWS 访问密钥。 *Type*: String. 条件：证书文件参数取代此参数。	条件
`-f, --credential-file`	同时包含密钥和访问密钥的文件。 *Type*: String. 条件：证书文件参数取代 --access-key 和 --secret-key 参数。	条件

cfn-hup

Abstract

检测任何资源元数据中的更改并在检测到更改时运行用户指定的操作。

说明

cfn-hup 帮助程序作为一项后台程序，旨在检测资源元数据中出现的变更，并在检测出变更的情况下，运行用户指定操作。通过此项操作，您可以通过 UpdateStack API 操作对您正在运行的 Amazon EC2 实例进行配置更新。

语法

```
cfn-hup --config|-c config.dir \
        --no-daemon \
        --verbose|-v
```

选项

名称	说明	必需
`--config|-c config.dir`	指定 cfn-hup 脚本用于查找 `cfn-hup.conf` 和 `hooks.d` 目录的路径。在 Windows 上，默认路径为 *`system_drive`*`\cfn`。在 Linux 上，默认路径为 `/etc/cfn`。	否
`--no-daemon`	指定此选项可运行 cfn-hup 脚本一次，然后退出。	否
`-v, --verbose`	指定此选项可使用详细模式。	否

cfn-hup.conf 配置文件

cfn-hup.conf 文件存储了堆栈的名称和 cfn-hup 后台程序目标 AWS 证书。cfn-hup.conf 文件采用以下格式：

```
[main]
stack=<stack-name-or-id>
```

名称	说明	必需
`stack`	堆栈名称或 ID。 *Type*: String.	是
`credential-file`	仅用户可用证书文件，与用于命令行工具的格式相同。 示例： Note cfn-hup 不要求提供证书，因此您不需要使用 `--credential-file` 选项。	否

名称	说明	必需
region	包含堆栈的 AWS 区域名称。 *示例*：us-east-1	否
interval	用于检查资源元数据更改的时间间隔（以分钟为单位） 类型：数字 *默认值*：10	否
verbose	指定是否要使用详细日志记录。 类型：布尔值 *默认值*：false	否

hooks.conf 配置文件

cfn-hup 后台程序定期调用的用户操作已在 hooks.conf 配置文件中予以定义。hooks.conf 文件采用以下格式：

```
[hookname]
triggers=post.add|post.update|post.remove
path=Resources.<logicalResourceId> (.Metadata|PhysicalResourceId)(.optional
Metadatapath)
action=<arbitrary shell command>
runas=<runas user>
```

操作运行时，它将在当前环境（cfn-hup 处于其中）的副本中运行，此时 CFN_OLD_METADATA 设定为路径的先前数值，且 CFN_NEW_METADATA 设定为当前值。

钩子配置文件仅可通过 cfn-hup 后台程序启动加载，因此需要后台程序才能重新启动新的钩子。先前元数据值的缓存存储在 /var/lib/cfn-hup/data/metadata_db（仅供机器读取）中，您可以删除此缓存，以强制 cfn-hup 再次运行所有 post.add 操作。

名称	说明	必需
hookname	此钩子的唯一名称 *Type*: String.	是
triggers	待检测条件的逗号分隔列表。 *有效值*：post.add | post.update | post.remove *示例*：post.add, post.update	是

名称	说明	必需
`path`	元数据对象路径。支持元数据块中的随机深度路径。 路径格式选择 • Resources.*<LogicalResourceId>* - 监控资源的上次更新时间，在对资源进行任何更改时触发。 • Resources.*<LogicalResourceId>*.PhysicalResourceId - 监控资源的物理 ID，仅当关联的资源标识更改（例如，新的 EC2 实例）时触发。 • Resources.*<LogicalResourceId>*.Metadata(*.optional path*) - 监控资源元数据的更改情况（可将元数据子路径指定为任意深度级别，以便监控特定值）。	是
`action`	按照既定运行的任意外壳程序命令。	是
`runas`	运行命令的用户。Cfn-hup 使用 su 命令切换至用户。	是

hooks.d 目录

要通过部署更改通知钩子支持若干应用程序组合，cfn-hup 要支持位于钩子配置目录中名为 hooks.d 的目录。您可以将一个或多个附加钩子配置文件置于 hooks.d 目录之中。附加钩子文件必须使用与 hooks.conf 文件相同的版式。

cfn-hup 后台程序将解析并加载此目录中的所有文件。如果 hooks.d 中的任何钩子与 hooks.conf 中的钩子同名，则应将上述钩子进行合并（表示 hooks.d 将重写 hooks.conf 针对上述两个文件指定的任何数值）。

AWS 命令行界面参考

Amazon Web Services (AWS) 现在提供 AWS 命令行界面 (CLI)，该命令行界面是一个用于控制和管理多个 AWS 服务的工具。AWS CLI 是推荐的 AWS CloudFormation CLI 工具。

有关如何使用 AWS CLI 命令的更多信息，请参阅使用 AWS CLI (p. 91)。有关 AWS CloudFormation 命令的参考信息，请参阅 *AWS Command Line Interface Reference*中的 cloudformation。

Note

早期的 AWS CloudFormation CLI 工具仍然可用，但不推荐使用。如果您需要有关早期 AWS CloudFormation CLI 工具的信息，请参阅文档档案中的 AWS CloudFormation CLI 参考。

AWS CloudFormation 限制

Abstract

描述 AWS CloudFormation 限制。

您的 AWS 账户具有 AWS CloudFormation 限制，您在创作模板和创建堆栈时可能需要了解这些限制。下表总结了这些 AWS CloudFormation 限制。

AWS CloudFormation 限制

限制	说明	值	优化策略
cfn-signal 等待条件数据 (p. 515)	cfn-signal 可以传递的最大数据量。	4,096 bytes	要传递更大的数据量，请将数据发送到某个 Amazon S3 存储桶，然后使用 cfn-signal 将 Amazon S3 URL 传递给该存储桶。
自定义资源响应 (p. 239)	自定义资源提供者可以传递的最大数据量。	4,096 bytes	
映像 (p. 6)	在 AWS CloudFormation 模板中可以声明的最大映射数。	100 个映射	要指定更多映射，请将模板分为多个模板（例如使用嵌套堆栈 (p. 250)）。
映射属性 (p. 6)	在 AWS CloudFormation 模板中可以为每个映射声明的最大映射属性数。	30 个属性	要指定更多映射属性，请将属性分为多个映射。
映射名称和映射属性名称 (p. 6)	每个映射名称的最大大小。	255 个字符	
输出 (p. 11)	在 AWS CloudFormation 模板中可以声明的最大输出数。	60 个输出	
输出名称 (p. 11)	输出名称的最大大小。	255 个字符	

限制	说明	值	优化策略
参数 (p. 5)	在 AWS CloudFormation 模板中可以声明的最大参数数量。	60 个参数	要指定更多参数，可以使用映射或逗号分隔列表以将多个值分配给一个参数。
参数名称 (p. 5)	参数名称的最大大小。	255 个字符	
参数值 (p. 5)	参数值的最大大小。	4,096 bytes	要使用更大的参数值，请创建多个参数，然后使用 `Fn::Join` 将多个值附加到单个值。
资源 (p. 8)	在 AWS CloudFormation 模板中可以声明的最大资源数。	200 个资源	要指定更多资源，请将模板分为多个模板（例如使用嵌套堆栈 (p. 250)）。
资源名称 (p. 8)	资源名称的最大大小。	255 个字符	
堆栈 (p. 2)	您可以创建的最大 AWS CloudFormation 堆栈数。	20 个堆栈	要创建更多堆栈，请删除不需要的堆栈，或请求增加 AWS 账户中的最大堆栈数。有关更多信息，请参阅 *AWS General Reference* 中的 AWS 服务限制。
请求中的模板正文大小 (p. 3)	可以在 `CreateStack`、`UpdateStack` 或 `ValidateTemplate` 请求中传递的模板正文的最大大小。	51,200 bytes	要使用更大的模板正文，请将模板分为多个模板（例如使用嵌套堆栈 (p. 250)）。或将模板上传到 Amazon S3 存储桶。
Amazon S3 对象中的模板正文大小 (p. 3)	对于使用 Amazon S3 模板 URL 的 `CreateStack`、`UpdateStack` 或 `ValidateTemplate` 请求，可以在 Amazon S3 对象中传递的模板正文的最大大小。	460,800 bytes	要使用更大的模板正文，请将模板分为多个模板（例如使用嵌套堆栈 (p. 250)）。
模板描述 (p. 3)	模板描述的最大大小。	1,024 bytes	

自定义资源参考

此部分提供有关以下内容的详细信息：

- 提供自定义资源时，发送到或发送自 AWS CloudFormation 的消息中使用的 JSON 请求和响应字段。
- 响应堆栈创建、堆栈更新和堆栈删除时，对custom resource provider发出请求或进行响应所需的字段。

在本章节中

自定义资源请求对象

模板开发人员请求属性

template developer使用 AWS CloudFormation 资源 AWS::CloudFormation::CustomResource (p. 239) 在模板中指定自定义资源。

在 AWS::CloudFormation::CustomResource 中，所有属性均由custom resource provider定义。只有一个必需属性：ServiceToken.

ServiceToken
: 从custom resource provider获取的用于访问服务的服务令牌。

  Required: Yes.

  Type: String.

资源属性中的所有其他字段是可选的，并将通过请求的 `ResourceProperties` 字段逐字发送到custom resource provider。提供者定义这些字段的名称和有效内容。

Custom Resource Provider 请求字段

这些字段将以 JSON 请求形式从 AWS CloudFormation 发送到提供者为实现此目的而配置的 SNS 主题中的custom resource provider。

RequestType
: 请求类型由template developer为包含自定义资源的堆栈启动的 AWS CloudFormation 堆栈操作（创建堆栈、更新堆栈或删除堆栈）设置。

  必须为以下值之一：`Create`、`Update` 或 `Delete`。

  Required: Yes.

  Type: String.

ResponseURL
: The response URL identifies a pre-signed Amazon S3 bucket that receives responses from the custom resource provider to AWS CloudFormation.

  Required: Yes.

  Type: String.

堆栈 ID
: The Amazon Resource Name (ARN) that identifies the stack containing the custom resource.

  Combining the *`StackId`* with the *`RequestId`* forms a value that can be used to uniquely identify a request on a particular custom resource.

  Required: Yes.

  Type: String.

请求 ID
: A unique ID for the request.

  Combining the *`StackId`* with the *`RequestId`* forms a value that can be used to uniquely identify a request on a particular custom resource.

  Required: Yes.

  Type: String.

ResourceType
: AWS CloudFormation 模板中模板开发人员选择的自定义资源的资源类型。

  Required: Yes.

  Type: String.

LogicalResourceId
: The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. 这是为了便于在custom resource provider与template developer之间进行通信而提供的。

  Required: Yes.

  Type: String.

PhysicalResourceId
: A required custom resource provider-defined physical ID that is unique for that provider.

  Required: 始终与 `Update` 和 `Delete` 请求一起发送，从不与 `Create` 请求一起发送。

  Type: String.

ResourceProperties
: This field contains the contents of the Properties object sent by the template developer. Its contents are defined by the custom resource provider.

  Required: No.

  Type: JSON object.

OldResourceProperties
: Used only for `Update` requests. Contains the resource properties that were declared previous to the update request.

  Required: Yes.

  Type: JSON object.

自定义资源响应对象

Custom Resource Provider 响应字段

状态
: The status value sent by the custom resource provider in response to an AWS CloudFormation-generated request.

  必须是 `SUCCESS` 或 `FAILED`。

  Required: Yes.

  Type: String.

原因
: Describes the reason for a failure response.

  Required: 如果 *`Status`* 为 `FAILED`，则是必需项；否则为可选项。

  Type: String.

PhysicalResourceId
: This value should be an identifier unique to the custom resource vendor, and can be up to 1Kb in size.

  Required: Yes.

  Type: String.

堆栈 ID
: The Amazon Resource Name (ARN) that identifies the stack containing the custom resource. This response value should be copied *verbatim* from the request.

  Required: Yes.

  Type: String.

请求 ID
: A unique ID for the request. This response value should be copied *verbatim* from the request.

  Required: Yes.

  Type: String.

LogicalResourceId
The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. This response value should be copied *verbatim* from the request.

Required: Yes.

Type: String.

Data
Optional, custom resource provider-defined name/value pairs to send with the response. The values provided here can be accessed by name in the template with `Fn::GetAtt`.

Required: No.

Type: JSON object.

自定义资源请求类型

当template developer创建、更新或删除包含自定义资源的堆栈时，请求类型通过由 AWS CloudFormation 发送的供应商请求对象 (p. 526)的 *`RequestType`* 字段发送。

每种请求类型均包含一组与请求一起发送的特定字段，包括自定义资源提供者提供的响应的 S3 URL。提供者使用 `SUCCESS` 或 `FAILED` 结果对 S3 存储桶做出响应。每个结果也包含 AWS CloudFormation 所需的一组特定字段。

此部分提供每种类型请求的请求字段和响应字段的相关信息，并提供相应示例。

在本章节中

创建

当template developer创建包含自定义资源的堆栈时，将发送 *`RequestType`* 已设置为 `"Create"` 的自定义资源提供者请求。

请求

创建请求包含以下字段：

RequestType
将为“Create”。

请求 ID
A unique ID for the request.

ResponseURL
The response URL identifies a pre-signed Amazon S3 bucket that receives responses from the custom resource provider to AWS CloudFormation.

ResourceType
AWS CloudFormation 模板中模板开发人员选择的自定义资源的资源类型。

LogicalResourceId
The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template.

堆栈 ID
The Amazon Resource Name (ARN) that identifies the stack containing the custom resource.

ResourceProperties
This field contains the contents of the Properties object sent by the template developer. Its contents are defined by the custom resource provider.

示例

```
{
   "RequestType" : "Create",
   "RequestId" : "unique id for this create request",
   "ResponseURL" : "pre-signed-url-for-create-response",
   "ResourceType" : "Custom::MyCustomResourceType",
   "LogicalResourceId" : "name of resource in template",
   "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-
name/guid",
   "ResourceProperties" : {
      "key1" : "string",
      "key2" : [ "list" ],
      "key3" : { "key4" : "map" }
   }
}
```

响应

成功

创建请求成功时，必须向 S3 存储桶发送包含以下字段的响应：

状态
必须为“SUCCESS”。

LogicalResourceId
The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. This response value should be copied *verbatim* from the request.

请求 ID
A unique ID for the request. This response value should be copied *verbatim* from the request.

堆栈 ID
The Amazon Resource Name (ARN) that identifies the stack containing the custom resource. This response value should be copied *verbatim* from the request.

PhysicalResourceId
This value should be an identifier unique to the custom resource vendor, and can be up to 1Kb in size.

Data
Optional, custom resource provider-defined name/value pairs to send with the response. The values provided here can be accessed by name in the template with `Fn::GetAtt`.

示例

```
{
   "Status" : "SUCCESS",
```

```
   "LogicalResourceId" : "name of resource in template (copied from request)",

   "RequestId" : "unique id for this create request (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-name/guid
 (copied from request)",
   "PhysicalResourceId" : "required vendor-defined physical id that is unique
for that vendor",
   "Data" : {
      "keyThatCanBeUsedInGetAtt1" : "data for key 1",
      "keyThatCanBeUsedInGetAtt2" : "data for key 2"
   }
}
```

已失败

创建请求失败时，必须向 S3 存储桶发送包含以下字段的响应：

状态
: 必须为“FAILED”。

原因
: Describes the reason for a failure response.

LogicalResourceId
: The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. This response value should be copied *verbatim* from the request.

请求 ID
: A unique ID for the request. This response value should be copied *verbatim* from the request.

堆栈 ID
: The Amazon Resource Name (ARN) that identifies the stack containing the custom resource. This response value should be copied *verbatim* from the request.

示例

```
{
   "Status" : "FAILED",
   "Reason" : "Required failure reason string",
   "LogicalResourceId" : "name of resource in template (copied from request)",

   "RequestId" : "unique id for this create request (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-name/guid
 (copied from request)"
}
```

删除

当template developer删除包含自定义资源的堆栈时，将发送 *`RequestType`* 已设置为 `"Delete"` 的自定义资源提供者请求。

请求

删除请求包含以下字段：

RequestType
: 将为“Delete”。

请求 ID
A unique ID for the request.

ResourceType
AWS CloudFormation 模板中模板开发人员选择的自定义资源的资源类型。

ResponseURL
The response URL identifies a pre-signed Amazon S3 bucket that receives responses from the custom resource provider to AWS CloudFormation.

LogicalResourceId
The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template.

堆栈 ID
The Amazon Resource Name (ARN) that identifies the stack containing the custom resource.

PhysicalResourceId
A required custom resource provider-defined physical ID that is unique for that provider.

ResourceProperties
This field contains the contents of the Properties object sent by the template developer. Its contents are defined by the custom resource provider.

示例

```
{
   "RequestType" : "Delete",
   "RequestId" : "unique id for this delete request",
   "ResponseURL" : "pre-signed-url-for-delete-response",
   "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-
name/guid",
   "ResourceType" : "Custom::MyCustomResourceType",
   "LogicalResourceId" : "name of resource in template",
   "PhysicalResourceId" : "custom resource provider-defined physical id",
   "ResourceProperties" : {
      "key1" : "string",
      "key2" : [ "list" ],
      "key3" : { "key4" : "map" }
   }
}
```

响应

成功

删除请求成功时，必须向 S3 存储桶发送包含以下字段的响应：

状态
必须为“SUCCESS”。

LogicalResourceId
The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. This response value should be copied *verbatim* from the request.

请求 ID
A unique ID for the request. This response value should be copied *verbatim* from the request.

堆栈 ID
The Amazon Resource Name (ARN) that identifies the stack containing the custom resource. This response value should be copied *verbatim* from the request.

PhysicalResourceId
: This value should be an identifier unique to the custom resource vendor, and can be up to 1Kb in size.

示例

```
{
   "Status" : "SUCCESS",
   "LogicalResourceId" : "name of resource in template (copied from request)",

   "RequestId" : "unique id for this delete request (copied from request)",
   "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-name/guid
 (copied from request)",
   "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

已失败

删除请求失败时，必须向 S3 存储桶发送包含以下字段的响应：

状态
: 必须为“FAILED”。

原因
: 失败的原因。

LogicalResourceId
: 从删除请求 (p. 531)复制的 *LogicalResourceId* 值。

请求 ID
: 从删除请求 (p. 531)复制的 *RequestId* 值。

堆栈 ID
: 从删除请求 (p. 531)复制的 *StackId* 值。

PhysicalResourceId
: custom resource provider定义的必需物理 ID，对该提供者唯一。

示例

```
{
"Status" : "FAILED",
"Reason" : "Required failure reason string",
"LogicalResourceId" : "name of resource in template (copied from request)",
"RequestId" : "unique id for this delete request (copied from request)",
"StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-name/guid
(copied from request)",
"PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

更新

当template developer更新包含自定义资源的堆栈时，将发送 *RequestType* 已设置为 "Update" 的自定义资源提供者请求。

请求

更新请求包含以下字段：

RequestType
: 将为“Update”。

请求 ID
: A unique ID for the request.

ResponseURL
: The response URL identifies a pre-signed Amazon S3 bucket that receives responses from the custom resource provider to AWS CloudFormation.

堆栈 ID
: The Amazon Resource Name (ARN) that identifies the stack containing the custom resource.

ResourceType
: AWS CloudFormation 模板中模板开发人员选择的自定义资源的资源类型。

LogicalResourceId
: The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template.

PhysicalResourceId
: A required custom resource provider-defined physical ID that is unique for that provider.

ResourceProperties
: template developer在更新后的 AWS CloudFormation 模板中声明的新资源属性值。

OldResourceProperties
: template developer以前在 AWS CloudFormation 模板中声明的资源属性值。

示例

```
{
   "RequestType" : "Update",
   "RequestId" : "unique id for this update request",
   "ResponseURL" : "pre-signed-url-for-update-response",
   "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-
name/guid",
   "ResourceType" : "Custom::MyCustomResourceType",
   "LogicalResourceId" : "name of resource in template",
   "PhysicalResourceId" : "custom resource provider-defined physical id",
   "ResourceProperties" : {
      "key1" : "new-string",
      "key2" : [ "new-list" ],
      "key3" : { "key4" : "new-map" }
   }
   "OldResourceProperties" : {
      "key1" : "string",
      "key2" : [ "list" ],
      "key3" : { "key4" : "map" }
   }
}
```

响应

成功

如果custom resource provider可以成功更新资源，AWS CloudFormation 会预期响应中的状态设置为 "SUCCESS"。

状态
: 必须为“SUCCESS”。

堆栈 ID
: The Amazon Resource Name (ARN) that identifies the stack containing the custom resource. This response value should be copied *verbatim* from the request.

请求 ID
: A unique ID for the request. This response value should be copied *verbatim* from the request.

LogicalResourceId
: The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. This response value should be copied *verbatim* from the request.

PhysicalResourceId
: This value should be an identifier unique to the custom resource vendor, and can be up to 1Kb in size.

示例

```
{
   "Status" : "SUCCESS",
  "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-name/guid
 (copied from request)",
   "RequestId" : "unique id for this update request (copied from request)",
   "LogicalResourceId" : "name of resource in template (copied from request)",

   "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

已失败

如果无法使用新属性集更新资源，则 AWS CloudFormation 会预期响应中的状态设置为“失败”并且响应中会提供失败原因。

状态
: 必须为“FAILED”。

原因
: Describes the reason for a failure response.

LogicalResourceId
: The template developer-chosen name (logical ID) of the custom resource in the AWS CloudFormation template. This response value should be copied *verbatim* from the request.

请求 ID
: A unique ID for the request. This response value should be copied *verbatim* from the request.

堆栈 ID
: The Amazon Resource Name (ARN) that identifies the stack containing the custom resource. This response value should be copied *verbatim* from the request.

PhysicalResourceId
: This value should be an identifier unique to the custom resource vendor, and can be up to 1Kb in size.

示例

```
{
   "Status" : "FAILED",
   "Reason" : "Required failure reason string",
   "LogicalResourceId" : "name of resource in template (copied from request)",

   "RequestId" : "unique id for this update request (copied from request)",
  "StackId" : "arn:aws:cloudformation:us-east-1:namespace:stack/stack-name/guid
 (copied from request)",
   "PhysicalResourceId" : "custom resource provider-defined physical id"
}
```

在 AWS CloudTrail 中记录 AWS CloudFormation API 调用

Abstract

使用 AWS CloudTrail 记录 AWS CloudFormation API 调用。

AWS CloudFormation 与 AWS CloudTrail 集成在一起，是用于捕获由 AWS 账户或代表 AWS 账户进行的 API 调用的服务。这种信息经过收集后，写入存储在所指定的 Amazon S3 存储桶中的日志文件。当您使用 AWS CloudFormation API、AWS CloudFormation 控制台、后端控制台或 AWS CLI 时，就会记录 API 调用。通过使用 CloudTrail 收集的信息，您可以确定向 AWS CloudFormation 发出了什么请求、发出请求的源 IP 地址以及发出请求的时间等。

如需了解更多关于 CloudTrail 的信息（包括如何配置和启用），请参阅 *AWS CloudTrail User Guide*。

Topics

AWS CloudFormation 信息CloudTrail

如果启用了 CloudTrail 日志，则会捕获所有 AWS CloudFormation 操作的调用并记录在日志文件中。所有 AWS CloudFormation 操作都在 *AWS CloudFormation API 参考* 中进行了说明。例如，对 CreateStack、DeleteStack 和 ListStacks 操作进行的调用在 CloudTrail 日志文件中生成相关条目。

每个日志条目都包含请求发出方的信息。例如，如果发出请求以列出 AWS CloudFormation 堆栈 (ListStacks)，则 CloudTrail 会记录发出请求的人员或服务的用户身份。用户身份信息有助于确定发出的请求是否具有根或 IAM 用户证书，是否具有角色或联合用户临时安全证书，或者是否是由其他 AWS 服务发出的。有关 CloudTrail 字段的更多信息，请参阅 *AWS CloudTrail User Guide* 中的 CloudTrail 事件参考。

可以将日志文件存储在存储桶中，时间不限，不过也可以定义 Amazon S3 生命周期规则以自动存档或删除日志文件。默认情况下，您的日志文件使用 Amazon S3 服务器端加密 (SSE) 进行加密。

了解 AWS CloudFormation 日志文件条目

CloudTrail 日志文件可包含一个或多个日志条目，每个条目由多个 JSON 格式的事件组成。一个日志条目表示来自任何源的一个请求，包含所请求的操作、任何输入参数以及操作的日期和时间等信息。这些日志条目不是按任何特定顺序出现的。也就是说，它们并不表示公用 API 调用的有序堆栈跟踪。

以下示例记录显示一个用于 CreateStack 操作的 CloudTrail 日志条目。该操作由名为 Alice 的 IAM 用户执行。

Note

仅记录输入参数键名称；而不记录参数值。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:02:43Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "CreateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "https://s3.amazonaws.com/Alice-dev/create_stack",
    "tags": [
      {
        "key": "test",
        "value": "tag"
      }
    ],
    "stackName": "my-test-stack",
    "disableRollback": true,
    "parameters": [
      {
        "parameterKey": "password"
      },
      {
        "parameterKey": "securitygroup"
      }
    ]
  },
  "responseElements": {
    "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-
stack/a38e6a60-b397-11e3-b0fc-08002755629e"
  },
  "requestID": "9f960720-b397-11e3-bb75-a5b75389b02d",
  "eventID": "9bf6cfb8-83e1-4589-9a70-b971e727099b"
}
```

以下示例记录显示 Alice 对 `my-test-stack` 堆栈调用了 UpdateStack 操作：

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:04:29Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "UpdateStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "templateURL": "https://s3.amazonaws.com/Alice-dev/create_stack",
    "parameters": [
      {
        "parameterKey": "password"
      },
      {
        "parameterKey": "securitygroup"
      }
    ],
    "stackName": "my-test-stack"
  },
  "responseElements": {
    "stackId": "arn:aws:cloudformation:us-east-1:012345678910:stack/my-test-
stack/a38e6a60-b397-11e3-b0fc-08002755629e"
  },
  "requestID": "def0bf5a-b397-11e3-bb75-a5b75389b02d",
  "eventID": "637707ce-e4a3-4af1-8edc-16e37e851b17"
}
```

以下示例记录显示 Alice 调用了 ListStacks 操作。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:03:16Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "ListStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": null,
```

```
  "responseElements": null,
  "requestID": "b7d351d7-b397-11e3-bb75-a5b75389b02d",
  "eventID": "918206d0-7281-4629-b778-b91eb0d83ce5"
}
```

以下示例记录显示 Alice 对 `my-test-stack` 堆栈调用了 DescribeStacks 操作。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:06:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DescribeStacks",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "224f2586-b398-11e3-bb75-a5b75389b02d",
  "eventID": "9e5b2fc9-1ba8-409b-9c13-587c2ea940e2"
}
```

以下示例记录显示 Alice 对 `my-test-stack` 堆栈调用了 DeleteStack 操作。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAABCDEFGHIJKLNMOPQ",
    "arn": "arn:aws:iam::012345678910:user/Alice",
    "accountId": "012345678910",
    "accessKeyId": "AKIDEXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2014-03-24T21:07:15Z",
  "eventSource": "cloudformation.amazonaws.com",
  "eventName": "DeleteStack",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-cli/1.2.11 Python/2.7.4 Linux/2.6.18-164.el5",
  "requestParameters": {
    "stackName": "my-test-stack"
  },
  "responseElements": null,
  "requestID": "42dae739-b398-11e3-bb75-a5b75389b02d",
  "eventID": "4965eb38-5705-4942-bb7f-20ebe79aa9aa"
}
```

文档历史记录

下表描述了自上次发行 AWS CloudFormation 以来对文档所做的重要更改。

- API 版本：2010-05-15

更改	说明	发行日期
更新堆栈增强功能	AWS CloudFormation 支持用于更新堆栈的其他功能： • 您可以更新 AWS CloudFormation 堆栈参数，而无需重新提交堆栈的模板。 • 您可以添加或删除 AWS CloudFormation 堆栈的 Amazon SNS 通知主题。 有关更多信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。	2014 年 5 月 12 日
Amazon Kinesis	您可以使用 AWS CloudFormation 创建可从数据源捕获并传输数据记录的 Amazon Kinesis 流。有关更多信息，请参阅 AWS::Kinesis::Stream (p. 357)。	2014 年 5 月 6 日
Amazon S3	AWS CloudFormation 支持其他 Amazon S3 存储桶属性： • 跨源资源共享 (CORS) 定义存储桶中对象的跨源资源共享。 • 生命周期定义 Amazon S3 如何在对象的生命周期内管理对象。 • 访问日志记录策略捕获有关对存储桶进行的请求的信息。 • 通知定义了要报告的事件以及要将消息发送到的 Amazon SNS 主题。 • 版本控制允许存储桶中的所有对象存在多个变体。 • 重定向和路由规则管理对存储桶网站终端节点进行的请求的重定向行为。 有关更多信息，请参阅 AWS::S3::Bucket (p. 402)。	2014 年 5 月 5 日

更改	说明	发行日期
Auto Scaling	AWS CloudFormation 支持 Auto Scaling 组的指标集合。有关更多信息，请参阅 AWS::AutoScaling::AutoScalingGroup (p. 219)。	2014 年 5 月 5 日
`Fn::If` 更新	您可以在模板的输出部分中使用 `Fn::If` 内部函数。有关更多信息，请参阅 条件函数 (p. 491)。	2014 年 5 月 5 日
使用 AWS CloudTrail 进行 API 日志记录	您可以使用 AWS CloudTrail 记录 AWS CloudFormation 请求。借助 AWS CloudTrail，您可以获取您账户的 AWS CloudFormation API 调用的历史记录。有关更多信息，请参阅 在 AWS CloudTrail 中记录 AWS CloudFormation API 调用 (p. 537)。	2014 年 4 月 2 日
Elastic Load Balancing 更新	您可以指定访问日志记录策略，以捕获有关对负载均衡器进行的请求的信息。您还可以指定连接耗尽策略，用于描述在实例取消注册或运行状况不佳时如何处理处于飞行状态的请求。有关更多信息，请参阅 AWS::ElasticLoadBalancing::LoadBalancer (p. 337)。	2014 年 3 月 20 日
AWS OpsWorks 支持	您可以使用 AWS CloudFormation 配置和管理 AWS OpsWorks 堆栈。有关更多信息，请参见 AWS::OpsWorks::Stack (p. 368) 或 AWS OpsWorks 代码段 (p. 158)。	2014 年 3 月 3 日
提高上限	您可以在 Amazon S3 中指定最大为 460 800 字节的模板大小。	2014 年 2 月 18 日
Amazon Redshift 支持	您可以使用 AWS CloudFormation 配置和管理 Amazon Redshift 集群。有关更多信息，请参见 Amazon Redshift 代码段 (p. 161) 或 AWS::Redshift::Cluster (p. 371)。	2014 年 2 月 10 日
Amazon S3 存储桶和存储桶策略更新	您可以更新 Amazon S3 存储桶和存储桶策略资源的一些属性。有关更多信息，请参见 AWS::S3::Bucket (p. 402) 或 AWS::S3::BucketPolicy (p. 409)。	2014 年 2 月 10 日
AWS Elastic Beanstalk 环境和应用程序版本更新	您可以更新 AWS Elastic Beanstalk 环境配置和应用程序版本。有关更多信息，请参见 AWS::ElasticBeanstalk::Environment (p. 333)、AWS::ElasticBeanstalk::ConfigurationTemplate (p. 331)或 AWS::ElasticBeanstalk::ApplicationVersion (p. 330)。	2014 年 2 月 10 日
Amazon SQS 更新	可以为 Amazon SQS 队列指定死信队列。有关更多信息，请参阅 AWS::SQS::Queue (p. 414)。	2014 年 1 月 29 日
Auto Scaling 计划操作	您可以基于计划在 Auto Scaling 组中扩展 Amazon EC2 实例数。通过使用计划，可以扩展应用程序来响应可预测的负载变化。有关更多信息，请参阅 AWS::AutoScaling::ScheduledAction (p. 232)。	2014 年 1 月 27 日
Amazon DynamoDB 二级索引	您可以为 Amazon DynamoDB 数据库创建本地和全局二级索引。通过使用二级索引，可以通过主键之外的属性高效地访问数据。有关更多信息，请参阅 AWS::DynamoDB::Table (p. 260)。	2014 年 1 月 27 日
Auto Scaling 更新	您可以为 Auto Scaling 组或启动配置指定实例 ID。还可以指定其他 Auto Scaling 块储存设备属性。有关更多信息，请参阅 AWS::AutoScaling::AutoScalingGroup (p. 219) 或 AWS::AutoScaling::LaunchConfiguration (p. 224)。	2014 年 1 月 2 日

更改	说明	发行日期
Amazon SQS 更新	您可以更新 Amazon SQS 队列并指定其他属性。有关更多信息，请参阅 AWS::SQS::Queue (p. 414)。	2014 年 1 月 2 日
限制提高	您可以在 AWS CloudFormation 模板中指定最多 60 个参数和 60 个输出	2014 年 1 月 2 日
新控制台	新 AWS CloudFormation 控制台添加了一些功能，如自动刷新堆栈事件和堆栈参数的字母数字排序	2013 年 12 月 19 日
跨区域负载均衡	凭借跨区域负载均衡，您可以将流量路由到所有可用区的后端实例。有关更多信息，请参阅 AWS::ElasticLoadBalancing::LoadBalancer (p. 337)。	2013 年 12 月 19 日
AWS Elastic Beanstalk 环境层	您可以指定 AWS Elastic Beanstalk 配置资源是支持 Web 服务器还是处理后台处理任务。有关更多信息，请参阅 AWS::ElasticBeanstalk::Environment (p. 333)。	2013 年 12 月 19 日
资源名称	您可以向以下资源分配名称（物理 ID）： • Amazon ElastiCache 集群 • Elastic Load Balancing 负载均衡器 • Amazon Relational Database Service 数据库实例 有关更多信息，请参阅 名称类型 (p. 465)。	2013 年 12 月 19 日
VPN 支持	您可以启用虚拟专用网关 (VGW) 将路由传播至 VPC 的路由表。有关更多信息，请参阅 AWS::EC2::VPNGatewayRoutePropagation (p. 319)。	2013 年 11 月 22 日
有条件地创建资源并分配属性	通过使用输入参数，您可以在 AWS CloudFormation 模板中定义条件，从而控制指定堆栈资源的创建和设置。例如，您可以使用条件为生产环境创建堆栈资源。使用相同的模板，您还可以为测试环境创建具有较低容量的类似堆栈资源。有关更多信息，请参阅 条件函数 (p. 491)。	2013 年 11 月 8 日
防止意外更新堆栈资源	您可以防止可能导致无意间更改堆栈资源的堆栈更新。例如，如果堆栈的数据库层很少更新，可以设置一个堆栈策略来防止大多数用户更新该数据库层。有关更多信息，请参阅 防止更新堆栈资源 (p. 69)。	2013 年 11 月 8 日
名称资源	您可以向某些资源分配名称，而不是使用 AWS CloudFormation 生成的物理 ID。以下 AWS CloudFormation 资源支持命名： • Amazon CloudWatch 警报 • Amazon DynamoDB 表 • AWS Elastic Beanstalk 应用程序和环境 • Amazon S3 存储桶 • Amazon SNS 主题 • Amazon SQS 队列 有关更多信息，请参阅 名称类型 (p. 465)。	2013 年 11 月 8 日

更改	说明	发行日期
分配自定义资源类型	在模板中，您可以为 AWS CloudFormation 自定义资源 (`AWS::CloudFormation::CustomResource`) 指定自己的资源类型。通过使用您自己的自定义资源类型名称，您可以快速识别堆栈中自定义资源的类型。例如，您可以指定 `"Type": "Custom::MyCustomResource"`。有关更多信息，请参阅 AWS::CloudFormation::CustomResource (p. 239)。	2013 年 11 月 8 日
添加虚拟参数	现在，通过引用 `AWS::AccountID` 虚拟参数，您可以在 AWS CloudFormation 模板内引用 AWS AccountID。有关更多信息，请参阅 虚拟参数参考 (p. 510)。	2013 年 11 月 8 日
在 IAM 策略中指定堆栈	您可以允许或拒绝 IAM 用户、组或角色对特定 AWS CloudFormation 堆栈进行的操作。例如，您可以拒绝对特定堆栈 ID 执行的删除堆栈操作。有关更多信息，请参阅 使用 AWS Identity and Access Management 控制访问 (p. 59)。	2013 年 11 月 8 日
联合支持	AWS CloudFormation 支持 IAM 角色的临时安全证书，从而可以实现联合和单一登录 AWS Management Console这样的方案。您还可以从 Amazon EC2 调用 AWS CloudFormation，无需通过使用 IAM 角色来嵌入长期安全证书。有关 AWS CloudFormation 和 IAM 的更多信息，请参阅使用 AWS Identity and Access Management 控制访问 (p. 59)。	2013 年 10 月 14 日
Amazon RDS 只读副本支持	您现在可以从源数据库实例创建 Amazon RDS 只读副本。有关更多信息，请参阅 AWS::RDS::DBInstance (p. 381) 资源中的 SourceDBInstanceIdentifier 属性。	2013 年 9 月 24 日
将公有 IP 地址与 Auto Scaling 组中的实例关联。	您现在可以将公有 IP 地址与 Auto Scaling 组中的实例关联。有关更多信息，请参阅 AWS::AutoScaling::LaunchConfiguration (p. 224)。	2013 年 9 月 19 日
其他 VPC 支持。	AWS CloudFormation 添加了多项可支持 VPC 和 VPN 功能的增强功能： • 您可以将一个公有 IP 地址和多个私有 IP 地址与 Amazon EC2 网络接口关联。有关更多信息，请参阅 AWS::EC2::NetworkInterface (p. 283)。您还可以将一个主要私有 IP 地址与一个弹性 IP 地址 (EIP) 关联。 • 您可以启用 DNS 支持并指定 DNS 主机名称。有关更多信息，请参阅 AWS::EC2::VPC (p. 310)。 • 您可以在虚拟专用网关与您的 VPN 网关之间指定静态路由。有关更多信息，请参阅 AWS::EC2::VPNConnectionRoute (p. 316)。	2013 年 9 月 17 日
对 Amazon ElastiCache 的 Redis 和 VPC 安全组支持。	您现在可以将 Redis 指定为 Amazon ElastiCache 集群的缓存引擎。您现在还可以对 Amazon ElastiCache 集群分配 VPC 安全组。有关更多信息，请参阅 AWS::ElastiCache::CacheCluster (p. 320)。	2013 年 9 月 3 日
并行执行堆栈创建、更新和删除，以及嵌套堆栈更新。	CloudFormation 现在可并行创建、更新和删除资源，从而提高了这些操作的性能。如果更新一个顶级模板，CloudFormation 将自动更新已发生更改的任何嵌套堆栈。有关更多信息，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。	2013 年 8 月 12 日

更改	说明	发行日期
现在可以在 AWS RDS 实例中设置 VPC 安全组	您现在可以使用 AWS CloudFormation 将 VPC 安全组分配到 Amazon RDS 实例。有关更多信息，请参阅 AWS::RDS::DBInstance (p. 381) 中的 VPCSecurityGroups (p. 387) 属性。	2013 年 2 月 28 日
Auto Scaling 组的回滚部署	AWS CloudFormation 现在支持 Auto Scaling 组的更新策略，这些策略说明如何在 Auto Scaling 组添加或移除实例时替换或修改 Auto Scaling 组中的实例。在创建堆栈更新堆栈过程中，您可以修改这些设置。 有关更多信息以及示例，请参阅 UpdatePolicy (p. 489)。	2013 年 2 月 20 日
堆栈更新的取消和回滚操作	AWS CloudFormation 支持取消堆栈更新。在发出更新请求时，堆栈必须处于 UPDATE_IN_PROGRESS 状态。以下主题中提供了更多信息： • 取消堆栈更新 (p. 68) • `aws cloudformation cancel-update-stack` • *AWS CloudFormation API 参考* 中的 CancelUpdateStack	2013 年 2 月 20 日
Auto Scaling 组的 EBS 优化实例	您现在可以在 Auto Scaling 组中部署 EBS 优化实例，为自动扩展的实例中的 Amazon EBS 提供专用吞吐量。具体的部署实施类似于以前针对优化的 EBS EC2 实例所发布的支持的实施。 有关更多信息，请参阅 AWS::AutoScaling::LaunchConfiguration (p. 224) 中的新 *`EbsOptimized`* 属性。	2013 年 2 月 20 日
新文档	AWS::EC2::Instance (p. 272) 现在提供了 BlockDeviceMappings 属性，以便您设置 EC2 实例的块储存设备映射。 通过这一更改，增加了两个新类型： • Amazon EC2 块储存设备映射属性 (p. 440) • Amazon Elastic Block Store 块储存设备属性 (p. 442)	2012 年 12 月 21 日
新文档	增加了新章节，来说明使用最近重新设计过的 AWS 管理控制台来创建和查看堆栈的流程。这些章节如下： • 创建堆栈 (p. 80) • 查看堆栈数据和资源 (p. 85)	2012 年 12 月 21 日

更改	说明	发行日期
新文档	自定义资源是特殊的 AWS CloudFormation 资源，提供了一种使 template developer 可以在 AWS CloudFormation 堆栈中包含非 AWS 资源的方式。custom resource provider 既可以是 template developer，也可以是独立的第三方资源提供者。 以下主题提供有关自定义资源的信息： • AWS CloudFormation 自定义资源演练 (p. 46) • AWS::CloudFormation::CustomResource (p. 239) • 自定义资源参考 (p. 526)	2012 年 11 月 15 日
更新的文档	AWS CloudFormation 现在支持为 Amazon RDS 实例指定部署的每秒 I/O 操作数 (IOPS)。您可以使用 AWS::RDS::DBInstance (p. 381) 中的新 IOPS (p. 385) 属性以 1 000 为增量将此值设置为从 1 000 到 10 000。 有关为 RDS 实例指定 IOPS 的更多信息，请参阅 *Amazon Relational Database Service 用户指南* 中的预配置 IOPS 部分。	2012 年 11 月 15 日
新增和更新的文档	重新组织了主题，以便更清楚地提供有关使用 AWS 管理控制台和使用 AWS CloudFormation 命令行界面 (CLI) 的特定信息。 文档中增加了关于标记 AWS CloudFormation 堆栈的信息，包括新指南和更新过的参考主题： • 有关使用控制台的新主题：设置堆栈选项 (p. 83). • *AWS CloudFormation API 参考* 中有关标签的新信息：CreateStack、堆栈和标签。 有关处理 Windows 堆栈 (p. 209)的新信息： • Microsoft Windows 亚马逊系统映像 (AMI) 和 AWS CloudFormation 模板 (p. 209) • 启动 AWS CloudFormation Windows 堆栈 (p. 210) • 访问 AWS CloudFormation Windows 实例 (p. 213) 新主题：在 AWS CloudFormation 模板中使用正则表达式 (p. 192).	2012 年 8 月 27 日

更改	说明	发行日期
新功能	AWS CloudFormation 现在通过 Amazon EC2 提供对 Virtual Private Cloud (VPC) 安全性的完全支持。您现在可以使用单个 AWS CloudFormation 模板并利用每一种类型的 VPC 资源（子网、网关、网络 ACL，路由表等）创建和填充整个 VPC。 可下载用于演示新 VPC 功能的模板： 单个子网中的单一实例 具备 Elastic Load Balancing (ELB) 功能的多个子网和一个 Auto Scaling 组 有关以下资源类型的文档已更新： AWS::EC2::SecurityGroup (p. 292) AWS::EC2::SecurityGroupIngress (p. 296) AWS::EC2::SecurityGroupEgress (p. 294) AWS::EC2::Instance (p. 272) AWS::AutoScaling::AutoScalingGroup (p. 219) AWS::EC2::EIP (p. 269) AWS::EC2::EIPAssociation (p. 270) AWS::ElasticLoadBalancing::LoadBalancer (p. 337) 文档中增加了新资源类型： AWS::EC2::VPC (p. 310) AWS::EC2::InternetGateway (p. 278) AWS::EC2::DHCPOptions (p. 266) AWS::EC2::DHCPOptions (p. 290) AWS::EC2::RouteTable (p. 288) AWS::EC2::NetworkAcl (p. 279) AWS::EC2::NetworkAclEntry (p. 281) AWS::EC2::Subnet (p. 300) AWS::EC2::VPNGateway (p. 317) AWS::EC2::CustomerGateway (p. 265)	2012 年 4 月 25 日
新功能	AWS CloudFormation 现在允许您在更新堆栈时在堆栈中添加或移除元素。AWS CloudFormation 堆栈更新 (p. 63) 已更新，并且演练中增加了一个新的部分： 更改堆栈的资源 (p. 43)，它说明了如何在更新堆栈时添加和移除资源.	2012 年 4 月 13 日

更改	说明	发行日期
新功能	AWS CloudFormation 现在为现有 Amazon Virtual Private Cloud (VPC) 中的资源提供支持。借助此版本，您可以： • 在现有 VPC 中启动 EC2 专用实例。有关更多信息，请参阅 AWS::EC2::Instance (p. 272)。 • 设置驻留在现有 VPC 中的 Amazon EC2 实例的 SourceDestCheck 属性。有关更多信息，请参阅 AWS::EC2::Instance (p. 272)。 • 在现有 VPC 中创建 Amazon 弹性 IP 地址。有关更多信息，请参阅 AWS::EC2::EIP (p. 269)。 • 使用 CloudFormation 在现有 VPC 中创建 VPC 安全组和传入/传出规则。有关更多信息，请参阅 AWS::EC2::SecurityGroup (p. 292)。 • 通过设置 AWS::AutoScaling::AutoScalingGroup 资源的 VPCZoneIdentifier 属性，将 Auto Scaling 组与现有 Amazon VPC 关联。有关更多信息，请参阅 AWS::AutoScaling::AutoScalingGroup (p. 219)。 • 将一个 Elastic Load Balancing 负载均衡器附加到 VPC 子网，并为该负载平衡器创建安全组。有关更多信息，请参阅 AWS::ElasticLoadBalancing::LoadBalancer (p. 337)。 • 在现有 VPC 中创建 RDS 实例。有关更多信息，请参阅 AWS::RDS::DBInstance (p. 381)。	2012 年 2 月 2 日
新功能	您现在可以在现有堆栈中更新以下资源的属性： • AWS::EC2::SecurityGroupIngress (p. 296) • AWS::EC2::SecurityGroupEgress (p. 294) • AWS::EC2::EIPAssociation (p. 270) • AWS::RDS::DBSubnetGroup (p. 391) • AWS::RDS::DBSecurityGroup (p. 392) • AWS::RDS::DBSecurityGroupIngress (p. 394) • AWS::Route53::RecordSetGroup (p. 400) 有关可更新资源的完整列表，以及更新堆栈时要考虑的详细事项，请参阅 AWS CloudFormation 堆栈更新 (p. 63)。	2012 年 2 月 2 日
新编指南	现有章节已重新组织为新章节：使用 AWS CloudFormation 模板 (p. 99)和管理堆栈。模板参考 (p. 217)移至目录的最高一层。估算 AWS CloudFormation 堆栈的成本 (p. 84)移至“入门”这一节。	2012 年 2 月 2 日

更改	说明	发行日期
新增内容	增加了三个 新章节： • 演练：更新堆栈 (p. 28)是说明更新 LAMP 堆栈的详细步骤的教程。 • 使用 AWS CloudFormation 部署应用程序 (p. 198)介绍如何使用 AWS CloudFormation 帮助程序用模板中存储的元数据部署应用程序。 • CloudFormation 帮助程序脚本参考 (p. 512)提供 AWS CloudFormation 帮助程序脚本（cfn-init、cfn-get-metadata、cfn-signal 和 cfn-hup）的参考材料。	2012 年 2 月 2 日
新功能	AWS CloudFormation 现在提供 aws cloudformation list-stacks 命令，您可使用此命令来按堆栈状态列出筛选出的堆栈。删除的堆栈在删除后最多还可在 90 天内列出。 有关更多信息，请参阅说明并列出堆栈 (p. 91).	2011 年 5 月 26 日
新功能	使用 aws cloudformation describe-stack-resources 和 aws cloudformation get-template 命令，现在可从已经删除了 90 天的堆栈中获取信息。 有关更多信息，请参阅列出资源 (p. 96)和 检索模板 (p. 97).	2011 年 5 月 26 日
新链接	Amazon Web Services General Reference 中现在还提供 AWS CloudFormation 的终端节点的信息。有关更多信息，请参阅 Amazon Web Services 常规参考中的“区域和终端节点”。	2011 年 3 月 1 日
首次发行	这是 AWS CloudFormation 的第一个公开发行版。	2011 年 2 月 25 日

AWS Glossary

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

Numbers and Symbols

100-continue	A method that enables a client to see if a server can accept a request before actually sending it. For large PUTs, this method can save both time and bandwidth charges.

A

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

ABT	Amazon Payments Balance Transfer. A method to approve or deny payments synchronously. It requires payment approval before processing continues.
access control list	A document that defines who can access a particular bucket or object. Each bucket and object in Amazon S3 has an ACL. The document defines what each type of user can do, such as write and read permissions.
access identifiers	See credentials.
access key ID	A string that AWS distributes to uniquely identify each AWS user; it is an alphanumeric token associated with your secret access key (p. 577).
access key rotation	A method to increase security by changing the AWS access key ID. This method enables you to retire an old key at your discretion.
access policy language	A language for writing documents (that is, *policies*) that specify who can access a particular AWS resource and under what conditions.
account	The AWS account associated with a particular AWS login ID and password. IAM: The AWS account that centrally controls all the resources created under its umbrella and pays for all AWS activity for those resources. The AWS account has permission to do anything and everything with all the AWS account resources. This is in contrast to the user (p. 582).

account activity	A web page showing your month-to-date AWS usage and costs. The account activity page is located at http://amazonaws.cn/account-activity.
ACH	Amazon bank account debit (aka *bank account withdrawal*, *bank account transaction*, and *Automated Clearing House*). An asynchronous bank account debit payment method.
action	An API function. Also called *operation* or *call*. The activity the principal (p. 571) has permission to perform. The action is B in the statement "A has permission to do B to C where D applies." For example, Jane sends a request to Amazon SQS with Action=ReceiveMessage. Amazon CloudWatch: The response initiated by the change in an alarm's state: for example, from OK to ALARM. The state change may be triggered by a metric reaching the alarm threshold, or by a SetAlarmState request. Each alarm can have one or more actions assigned to each state. Actions are performed once each time the alarm changes to a state that has an action assigned, such as an Amazon Simple Notification Service notification, an Auto Scaling policy execution or an Amazon EC2 instance stop/terminate action.
activate URL	The location where your customers can generate a new activation key (p. 551) for your product, should you request it. The activate URL is http://www.amazon.com/dp-activate.
activation	The process your product goes through to prepare itself for the customer's use. During activation, your product obtains the required credentials for the customer.
activation key	An encoded string that represents the relationship between a customer and a Amazon DevPay product the customer has purchased. AWS generates this value when the customer completes the purchase of the product. You use the key to obtain credentials related to the customer and product.
active authorization	When you inform customers of a price change for your product, they have to take action to accept the price change. If they don't take the action to accept the price change, their access to your product is canceled when the price change takes effect.
active trusted signers	A list showing each of the trusted signers you've specified and the IDs of the corresponding active key pairs that CloudFront is aware of. To be able to create working signed URLs, a trusted signer must appear in this list with at least one key pair ID.
administrative suspension	Auto Scaling might suspend processes for Auto Scaling group (p. 554) that repeatedly fail to launch instances. Auto Scaling groups that most commonly experience administrative suspension have zero running instances, have been trying to launch instances for more than 24 hours, and have not succeeded in that time.
alarm	An item that watches a single metric over a specified time period, and triggers an Amazon SNS topic (p. 582) or an Auto Scaling policy (p. 571) if the value of the metric crosses a threshold value over a predetermined number of time periods.
allow	An allow results from a statement that has effect=allow, assuming any stated conditions are met. Example: Allow requests received before 1:00 p.m. on April 30, 2010. An allow overrides any default deny (p. 559), but never an explicit deny (p. 561).
Amazon CloudFront	An AWS content delivery service that helps you improve the performance, reliability, and availability of your websites and applications. See Also http://amazonaws.cn/cloudfront.

Amazon CloudSearch	A fully-managed service in the cloud that makes it easy to set up, manage, and scale a search solution for your website.
Amazon CloudWatch	A web service that enables you to monitor and manage various metrics, and configure alarm actions based on data from those metrics. See Also http://amazonaws.cn/cloudwatch.
Amazon DevPay	An easy-to-use online billing and account management service that makes it easy for you to sell an Amazon EC2 AMI or an application built on Amazon S3. See Also http://amazonaws.cn/devpay.
Amazon Elastic Block Store	A service that provides block level storage volumes for use with EC2 instances. See Also http://amazonaws.cn/ebs.
Amazon EBS-backed AMI	Instances launched from this type of AMI use an Amazon EBS volume as their root device. Compare this with instances launched from Amazon S3-backed AMIs, which use the instance store as the root device.
Amazon Elastic Compute Cloud	A web service that enables you to launch and manage Linux/UNIX and Windows server instances in Amazon's data centers. See Also http://amazonaws.cn/ec2.
Amazon EC2 VM Import Connector	See http://amazonaws.cn/ec2/vmimport.
Amazon Elastic MapReduce	A web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop processing combined with several AWS products to do such tasks as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing. See Also http://amazonaws.cn/elasticmapreduce.
Amazon Flexible Payments Service	A web service that enables developers to accept payments on their websites. See Also http://amazonaws.cn/fps.
Amazon Machine Image	An encrypted machine image stored in Amazon Elastic Block Store (p. 552) or Amazon Simple Storage Service. AMIs are like a template of a computer's root drive. They contain the operating system and can also include software and layers of your application, such as database servers, middleware, web servers, and so on.
Mechanical Turk	Provides an on-demand, scalable, human workforce to complete jobs that humans can do better than computers. See Also http://amazonaws.cn/mturk.
Amazon Payments	See http://payments.amazon.com.
Amazon Relational Database Service	A web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. See Also http://amazonaws.cn/rds.
Amazon Resource Name	A standardized way to refer to an AWS resource. For example: arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Bob.
Amazon Route 53	A web service you can use to create a new DNS service or to migrate your existing DNS service to the cloud. See Also http://amazonaws.cn/route53.
Amazon S3	See Amazon Simple Storage Service.

Amazon S3-Backed AMI	Instances launched from this type of AMI use the instance store as their root device. Compare this with instances launched from Amazon EBS-backed AMIs, which use an Amazon EBS volume as the root device.
Amazon Simple Email Service	An easy-to-use, cost-effective email solution for applications. See Also http://amazonaws.cn/ses.
Amazon Simple Notification Service	A web service that enables applications, end-users, and devices to instantly send and receive notifications from the cloud. See Also http://amazonaws.cn/sns.
Amazon Simple Pay	Incorporates a subset of Amazon FPS functionality. See Also https://payments.amazon.com.
Amazon Simple Queue Service	Reliable and scalable hosted queues for storing messages as they travel between computers. See Also http://amazonaws.cn/sqs.
Amazon Simple Storage Service	Storage for the internet. You can use it to store and retrieve any amount of data at any time, from anywhere on the web. See Also http://amazonaws.cn/s3.
Amazon SimpleDB	A highly-available, scalable, and flexible non-relational data store that enables you to store and query data items using web service requests. See Also http://amazonaws.cn/simpledb.
Amazon Virtual Private Cloud	A web service that enables you to create a virtual network for your AWS resources. See Also http://amazonaws.cn/vpc.
Amazon Web Services	An infrastructure web services platform in the cloud for companies of all sizes. See Also http://amazonaws.cn.
AMI	See Amazon Machine Image.
application	A logical collection of AWS Elastic Beanstalk components, including environments, versions, and environment configurations. An application is conceptually similar to a folder.
Application Billing	The location where your customers manage the Amazon DevPay products they've purchased. This is the URL http://www.amazon.com/dp-applications.
application version	A specific, labeled iteration of an application that represents a functionally consistent set of deployable application code. A version points to an Amazon S3 object (a JAVA WAR file) that contains the application code.
approval	If a Worker's response satisfies your Human Intelligence Task (p. 563), you approve the assignment. When you approve an assignment Mechanical Turk transfers the HIT reward from your Mechanical Turk account to the Worker's Amazon Payments account.
ARN	See Amazon Resource Name.
assignment	When a worker (p. 584) finds a Human Intelligence Task (p. 563) (HIT) to complete, the worker accepts the HIT. Mechanical Turk creates an *assignment* to track the work to completion and store the answer the worker submits. The assignment belongs exclusively to the worker who accepted it and guarantees that the worker can submit results and be eligible for a reward—up until the HIT or assignment expires.

asynchronous bounce	A type of bounce (p. 555) that occurs when a receiver (p. 574) initially accepts an email message for delivery and then subsequently fails to deliver it.
attribute	Similar to a column on a spreadsheet, an attribute represents a data category. In Amazon SimpleDB, an attribute has a name (such as *color*), which has a value (such as *blue*) when applied to a data item.
authentication	The process of proving your identity to a system.
Auto Scaling	A web service designed to launch or terminate instance (p. 564)s automatically based on user-defined policies, schedules, and health checks. See Also http://amazonaws.cn/autoscaling.
Auto Scaling group	A representation of multiple Amazon Elastic Compute Cloud (p. 552) instance (p. 564)s that share similar characteristics, and that are treated as a logical grouping for the purposes of instance scaling and management.
Availability Zone	A distinct location within a region (p. 574) that is insulated from failures in other Availability Zones, and provides inexpensive, low-latency network connectivity to other Availability Zones in the same region.
AWS	See Amazon Web Services.
AWS CloudFormation	A service for writing or changing templates that create and delete related AWS resources together as a unit. See Also http://amazonaws.cn/cloudformation.
AWS Consolidated Billing	A billing option that lets you get a single bill for multiple AWS accounts. See Also http://amazonaws.cn/consolidated-billing.
AWS Elastic Beanstalk	See Also http://amazonaws.cn/elasticbeanstalk.
AWS Import/Export	A service for transferring large amounts of data between AWS and portable storage devices. See Also http://amazonaws.cn/importexport.
AWS Identity and Access Management	A web service that enables Amazon Web Services (p. 553) customers to manage users and user permissions within AWS. See Also http://amazonaws.cn/iam.
AWS Management Console	A graphical interface to manage compute, storage, and other cloud resources. See Also http://amazonaws.cn/console.
AWS Multi-Factor Authentication	An optional AWS account security feature. Once you enable AWS MFA, you must provide a six-digit, single-use code in addition to your sign-in credentials whenever you access secure AWS web site pages or the AWS Management Console. You get this single-use code from an authentication device that you keep in your physical possession. See Also http://amazonaws.cn/mfa.
AWS Resources	See resource.
AWS VPN CloudHub	Enables secure communication between branch offices using a simple hub-and-spoke model, with or without a VPC.

B

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

basic monitoring	Monitoring of AWS-provided metrics derived at a 5-minute frequency.
batch	A collection of add and delete document operations in Search Data Format (SDF). You use the document service API to submit add and delete document operations to update the data in your search domain.
BGP ASN	Border Gateway Protocol Autonomous System Number. A unique identifier for a network, for use in BGP routing. Amazon EC2 supports all 2-byte ASN numbers in the range of 1 - 65334, with the exception of 7224, which is reserved.
blacklist	A list of IP addresses, email addresses, or domains that an Internet Service Provider (p. 564) suspects to be the source of spam (p. 579). The ISP blocks incoming emails from these addresses or domains.
block	A data set. Amazon EMR breaks large amounts of data into subsets. Each subset is called a data block. Amazon EMR assigns an ID to each block and uses a hash table to keep track of block processing.
block device	A storage device that supports reading and (optionally) writing data in fixed-size blocks, sectors, or clusters.
block device mapping	A mapping structure for every AMI and instance that specifies the block devices attached to the instance.
bootstrap action	A user-specified default or custom action that runs a script or an application on all nodes of a job flow before Hadoop starts.
Border Gateway Protocol Autonomous System Number	See BGP ASN.
bounce	A failed email delivery attempt.
breach	The condition in which a user-set threshold (upper or lower boundary) is passed. If the duration of the breach is significant, as set by a breach duration parameter, it can possibly start a scaling activity (p. 576).
bucket	A container for objects stored in Amazon S3. Every object is contained in a bucket. For example, if the object named `photos/puppy.jpg` is stored in the `johnsmith` bucket, then authorized users can access the object with the URL `http://johnsmith.s3.amazonaws.com/photos/puppy.jpg`.
bucket owner	Just as Amazon is the only owner of the domain name Amazon.com, only one person or organization can own a bucket in Amazon S3.
bundling	A commonly used term for creating an Amazon Machine Image (p. 552). It specifically refers to creating Amazon S3-backed AMIs.
buyer	Amazon FPS: The customer making a purchase. Also called a sender (p. 577). The buyer pays the seller (p. 577) for a product or service. Amazon Simple Pay: The customer who sends a payment using an Amazon Simple Pay button.

C

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

cache cluster	A logical cache distributed over multiple cache node (p. 556)s. A cache cluster can be set up with a specific number of cache nodes.
cache cluster identifier	Customer-supplied identifier for the cache cluster that must be unique for that customer in an AWS region.
cache engine version	The version of the Memcached service that is running on the cache node.
cache node	A fixed-size chunk of secure, network-attached RAM. Each cache node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of cache nodes are supported, each with varying amounts of associated memory.
cache node type	EC2 instance type used to run the cache node.
cache parameter group	A container for cache engine parameter values that can be applied to one or more cache clusters.
cache security group	A group maintained by ElastiCache that combines ingress authorizations to cache nodes for hosts belonging to Amazon EC2 security groups specified through the console or the API or command line tools.
caller	A developer who facilitates payment between a sender (p. 577) and a recipient (p. 574).
caller reference	A unique value that you provide and AWS uses to prevent replays of your request.
canned access policy	A standard access control policy that you can apply to a bucket or object. Options include: private, public-read, public-read-write, and authenticated-read.
canonicalization	The process of converting data into a standard format that a service such as Amazon S3 can recognize.
capacity	Each Auto Scaling group (p. 554) is defined with a minimum and maximum compute size. The amount of available compute size at any time is the current capacity. A scaling activity (p. 576) increases or decreases the capacity—within the defined minimum and maximum values.
Cascading	Cascading is an open-source Java library that provides a query API, a query planner, and a job scheduler for creating and running Hadoop MapReduce applications. Applications developed with Cascading are compiled and packaged into standard Hadoop-compatible JAR files similar to other native Hadoop applications.
CBUI	See Co-Branded User Interface.
certificate	A credential that some AWS products use to authenticate AWS accounts and users. Also known as an X.509 certificate. The certificate is paired with a private key.
chargeable resources	Features or services whose use incurs fees. Although some AWS products are free, others include charges. For example, in an AWS CloudFormation stack (p. 579), AWS resources that have been created incur charges. The amount

	charged depends on the usage load. Use the Amazon Web Services Simple Monthly Calculator at http://calculator.s3.amazonaws.com/calc5.html to estimate your cost prior to creating instances, stacks, or other resources.
chargeback	A payment reversal that a bank issues when the buyer disputes a charge.
CIDR block	Classless Inter-Domain Routing. An Internet protocol address allocation and route aggregation methodology. See Also http://en.wikipedia.org/wiki/CIDR_notation.
CloudHub	See AWS VPN CloudHub.
cluster compute instance	A type of instance (p. 564) that provides a great amount of CPU power coupled with increased networking performance, making it well suited for High Performance Compute (HPC) applications and other demanding network-bound applications.
cluster placement group	A logical cluster compute instance (p. 557) grouping to provide lower latency and high-bandwidth connectivity between the instances.
CNAME	Canonical Name Record. A type of resource record in the Domain Name System (DNS) that specifies that the domain name is an alias of another, canonical domain name. More simply, it is an entry in a DNS table that lets you alias one fully qualified domain name to another.
Co-Branded service	The web service underlying the Co-Branded User Interface (p. 557).
Co-Branded User Interface	For the buyer: A series of web pages hosted by Amazon Payments that enables the buyer to authorize a payment. For the merchant: A series of web pages that Amazon Payments hosts so that a website owner can register a merchant's product for sale on the website.
co-branding	Running a business logo on a site or service that another company provides. Co-branding with Amazon Simple Pay, for example, is merely adding an independent merchant logo to each of the payment authorization web pages.
complaint	The event in which a recipient (p. 574) who does not want to receive an email message clicks "Mark as Spam" within the email client, and the Internet Service Provider (p. 564) sends a notification to Amazon SES.
condition	Any restriction or detail about a permission. The condition is *D* in the statement "A has permission to do B to C where D applies." Conditions are always optional.
conditional parameter	See mapping.
configuration API	The API that you use to create, configure, and manage Amazon CloudSearch domains.
configuration template	A series of key–value pairs that define parameters for various AWS products so that AWS Elastic Beanstalk can provision them for an environment.
confirmation email	The email Amazon Payments sends to your customers to notify them that a price change you scheduled has taken effect.
consistency model	The method a service uses to achieve high availability. For example, it could involve replicating data across multiple servers in a data center. See Also eventual consistency.
consistent read	When data is written or updated successfully, all copies of the data are updated in all AWS regions. However, it takes time for the data to propagate to all storage locations. A consistent read returns a result that reflects any writes that received

	a successful response before the read request—regardless of the region. By contrast, an eventually consistent read returns data from only one region and might not show the most recent write information. See Also eventual consistency.
console	See AWS Management Console.
Consolidated Billing	See AWS Consolidated Billing.
cooldown period	Amount of time during which Auto Scaling does not allow the desired size of the Auto Scaling group (p. 554) to be changed by any other notification from a CloudWatch alarm (p. 551).
core node	An EC2 instance (p. 560) that runs Hadoop map and reduce tasks and stores data using the Hadoop Distributed File System (HDFS). Core nodes are managed by the master node (p. 567), which assigns Hadoop tasks to nodes and monitors their status. The EC2 instances you assign as core nodes are capacity that must be allotted for the entire job flow run. Because core nodes store data, you can't remove them from a job flow. However, you can add more core nodes to a running job flow. Core nodes run both the DataNodes and TaskTracker Hadoop daemons.
corpus	A collection of data that you want to search.
credentials	Also called *access credentials* or *security credentials*. In authentication and authorization, a system uses credentials to identify who is making a call and whether to allow the requested access. In AWS, these credentials are typically the access key ID (p. 550) and the secret access key (p. 577).
customer gateway	A router or software application on your side of a VPN tunnel that is managed by Amazon VPC. The internal interfaces of the customer gateway are attached to one or more devices in your home network. The external interface is attached to the VPG (p. 583) across the VPN tunnel.

D

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

dashboard	See service health dashboard.
database engine	The database software and version running on the DB instance (p. 558).
database name	The name of a database hosted in a DB instance (p. 558). A DB instance can host multiple databases, but databases hosted by the same DB instance must each have a unique name within that instance.
DB compute class	Size of the database compute platform used to run the instance.
DB instance	An isolated database environment running in the cloud. A DB instance can contain multiple user-created databases.
DB instance identifier	User-supplied identifier for the DB instance. The identifier must be unique for that user in an AWS region (p. 574).
DB parameter group	A container for database engine parameter values that apply to one or more DB instance (p. 558)s.

DB security group	A method that controls access to the DB instance (p. 558). By default, network access is turned off to DB instances. After ingress is configured for a security group, the same rules apply to all DB instances associated with that group.
DB snapshot	A user-initiated point backup of a DB instance.
Dedicated Instance	An instance that is physically isolated at the host hardware level and launched within a VPC.
Dedicated Reserved Instance	An option you purchase to guarantee that sufficient capacity will be available to launch Dedicated Instances into a VPC.
default deny	The default result from a policy (p. 571) in the absence of an allow (p. 551) or explicit deny (p. 561). For example: if a user (p. 582) requests to use Amazon SQS, but the only policy that applies to the user states that the user can use Amazon SimpleDB, then that policy results in a default deny.
delete marker	An object with a key and version ID, but without content. Amazon S3 inserts delete markers automatically into versioned buckets when an object is deleted.
deliverability	The likelihood that an email message will arrive at its intended destination.
deliveries	The number of emails, sent through Amazon SES, that were accepted by an Internet Service Provider (p. 564) for delivery to recipient (p. 574)s over a period of time.
detailed monitoring	Monitoring of AWS-provided metrics derived at a 1-minute frequency.
Description property	A property added to parameters, resources, resource properties, mappings, and outputs, to help you to document AWS CloudFormation template elements.
DevPay Activity	The location (Amazon DevPay Activity) where you manage the Amazon DevPay products you've created.
dimension	A name/value pair (for example, InstanceType=m1.small, or EngineName=mysql), that contains additional information to identify a metric.
discussion forums	A place where AWS users can post technical questions and feedback to help accelerate their development efforts and to engage with the AWS community. The discussion forums are located at http://amazonaws.cn/forums.
distributed cache	A Hadoop feature that allow you to transfer files from a distributed file system to the local file system. It can distribute data and text files as well as more complex types such as archives and JARs.
distribution	A link between an origin server (such as an Amazon S3 bucket) and a domain name, which CloudFront automatically assigns. Through this link, CloudFront identifies the object you have stored in your origin server (p. 570).
DKIM	DomainKeys Identified Mail. A standard that email senders use to sign their messages. ISPs use those signatures to verify that messages are legitimate. For more information, see http://www.dkim.org.
DNS	See Domain Name System (DNS).
document	Represents an item that can be returned as a search result. Each document has a collection of fields that contain the data that can be searched or returned. The value of a field can be either a string or a number. Each document must have a unique ID, a version number, and at least one field.

document ID (docid)	A unique alpha-numeric identifier for a document. This is the `id` attribute that's specified in an add or delete operation when using the document service API to update your search domain.
document service API	The API that you use to submit SDF batches to update the data in an Amazon CloudSearch domain.
document service endpoint	The URL that you connect to when sending document updates to a search domain. Each search domain has a unique document service endpoint.
document version	In SDF, each document has a numeric version number that's used to guarantee that a search domain always reflects the most recent document updates. Document updates are applied only if the version number specified in the add or delete operation is *greater* than the existing version number.
domain	All Amazon SimpleDB information is stored in domains. Domains are like tables that contain similar data. You can execute queries against a domain, but cannot execute joins between domains. See Also search domain.
Domain Name System (DNS)	A distributed naming system that associates network information with human-readable domain names on the Internet.
Donation button	An HTML-coded button to provide an easy and secure way for US-based, IRS-certified 501(c)3 nonprofit organizations to solicit donations.

E

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

EBS	See Amazon Elastic Block Store.
EC2	See Amazon Elastic Compute Cloud.
EC2 compute unit	An AWS standard for compute CPU and memory. This measure enables you to evaluate the CPU capacity of different EC2 instance types.
EC2 instance	In Amazon EC2, this is simply an instance. Other AWS services use the term EC2 instance to distinguish these instances from other types of instances they support.
edge location	A site that CloudFront uses to cache copies of your content for faster delivery to users at any location.
Edit Token API	A Co-Branded service (p. 557) API that enables you to view an existing token's details and to change the payment instrument (p. 570) for the token.
Elastic Block Store	See Amazon Elastic Block Store.
elastic IP address	A fixed (static) IP address that you have allocated in Amazon EC2 or Amazon VPC and then attached to an instance. Elastic IP addresses are associated with your account, not a specific instance. They are *elastic* because you can easily allocate, attach, detach, and free them as your needs change. Unlike traditional static IP addresses, elastic IP addresses allow you to mask instance or Availability Zone failures by rapidly remapping your public IP addresses to another instance.

Elastic Load Balancing	A web service that improves an application's availability by distributing incoming traffic between two or more EC2 instance (p. 560)s. See Also http://amazonaws.cn/elasticloadbalancing.
elastic network interface	An additional network interface that can be attached to an instance (p. 564). ENIs include a primary private IP address, one or more secondary private IP addresses, an elastic IP address (optional), a MAC address, membership in specified security groups, a description, and a source/destination check flag. You can create an ENI, attach it to an instance, detach it from an instance, and attach it to another instance.
endpoint	A URL that identifies a host and port as the entry point for a web service. Every web service request contains an endpoint. Most AWS products provide regional endpoints to enable faster connectivity. For more information, see Regions and Endpoints in the *Amazon Web Services General Reference* ElastiCache: The DNS name of a cache node (p. 556). Amazon RDS: The DNS name of a DB instance (p. 558). AWS CloudFormation: The DNS name or IP address of the server that receives an HTTP request.
endpoint port	ElastiCache: The port number used by a cache node (p. 556). Amazon RDS: The port number used by a DB instance (p. 558).
ephemeral store	See instance store.
epoch	The date from which time is measured. For most Unix environments, the epoch is January 1, 1970.
eventual consistency	The method through which AWS products achieve high availability, which involves replicating data across multiple servers in Amazon's data centers. When data is written or updated and "Success" is returned, all copies of the data are updated. However, it takes time for the data to propagate to all storage locations. The data will eventually be consistent, but an immediate read might not show the change. Consistency is usually reached within seconds, but a high system load might increase this time.
eventually consistent read	See consistent read.
eviction	An *eviction* occurs when CloudFront deletes an object from an edge location (p. 560) before its expiration time. If an object in an edge location isn't frequently requested, CloudFront might evict the object (remove the object before its expiration date) to make room for objects that are more popular.
environment	A specific running instance of an application (p. 553). The application has a CNAME and includes an application version and a customizable configuration (which is inherited from the default container type).
environment configuration	A collection of parameters and settings that define how an environment and its associated resources behave.
expiration	*Expiration* occurs when CloudFront stops serving an object from an edge location (p. 560). The next time the edge location needs to serve that object, CloudFront gets a new copy from the origin server (p. 570).
explicit deny	An *explicit deny* results from a statement that has effect=deny, assuming that any stated conditions are met. Example: *Deny all requests from Antarctica*. Any request

	that comes from Antarctica will always be denied no matter what any other policy (p. 571) might allow.
explicit launch permission	An Amazon Machine Image (p. 552) launch permission granted to a specific AWS account.
exponential backoff	A strategy that incrementally increases the wait between retry attempts in order to reduce the load on the system and increase the likelihood that repeated requests will succeed. For example, client applications might wait up to 400 milliseconds before attempting the first retry, up to 1600 milliseconds before the second, up to 6400 milliseconds (6.4 seconds) before the third, and so on.

F

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

facet	A search index field that represents a category that you want to use to refine and filter search results.
facet constraint	A particular facet value that you want to count when searching.
facet enabled	A search index field option that enables facet information to be calculated for the field.
FBL	See feedback loop.
feedback loop	The mechanism by which a mailbox provider (for example, an Internet Service Provider (p. 564)) forwards a recipient (p. 574)'s complaint (p. 557) back to the sender (p. 577).
field weight	The relative importance of a text field in a search index. Field weights control how much matches in particular text fields affect a document's text_relevance score.
filter	A criterion you specify to limit the results when you list or describe your Amazon EC2 resources.
format version	See template format version.
forums	See discussion forums.
FPS	See Amazon Flexible Payments Service.
function	See intrinsic function.
funding token	A payment token used to fund a prepaid instrument (p. 571).

G

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

gibibyte	A contraction of giga binary byte, a gibibyte is 2^30 bytes or 1,073,741,824 bytes. A gigabyte is 10^9 or 1,000,000,000 bytes.

H

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

Hadoop	See http://hadoop.apache.org.
hard bounce	A persistent email delivery failure such as "mailbox does not exist."
hardware VPN	A hardware-based IPsec VPN connection over the Internet.
HDFS	Hadoop Distributed File System. The HDFS file system stores large files across multiple machines. It achieves reliability by replicating the data across multiple hosts, and hence does not require RAID storage on hosts.
health check	A system call to check on the health status of each instance in an Auto Scaling group.
high-quality email	Email that recipients find valuable and want to receive. Value means different things to different recipients and can come in the form of offers, order confirmations, receipts, newsletters, etc.
hit	A document that matches the criteria specified in the search request. Also referred to as a *search result*.
HIT	See Human Intelligence Task.
Hive	An open source, data warehouse and analytic package that runs on top of Hadoop. Hive scripts use an SQL-like language called Hive QL (query language) that abstracts the MapReduce programming model and supports typical data warehouse interactions.
HMAC	Hash-based Message Authentication Code. A specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key. You can use it to verify both the data integrity and the authenticity of a message at the same time. AWS calculates the HMAC using a standard, cryptographic hash algorithm, such as SHA-256.
hosted zone	A collection of resource record sets that Amazon Route 53 hosts. Like a traditional DNS zone file, a hosted zone represents a collection of records that are managed together under a single domain name.
Human Intelligence Task	A task that a Requester (p. 575) submits to Mechanical Turk for workers (p. 584) to perform. A HIT represents a single, self-contained task, for example, "Identify the car color in the photo." HITs contain all of the information a worker needs to answer a question, including the kinds of answers you would consider valid.
HVM virtualization	Hardware Virtual Machine virtualization. Lets the guest VM run as though it is on a native hardware platform, except that it still uses para-virtual (PV) network and storage drivers for improved performance.

I

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

image	See Amazon Machine Image.
import/export station	A machine that uploads or downloads your data to, or from, Amazon S3.
import log	A report that contains details about how AWS Import/Export processed your data.
inbound request	A button click or other form request to Amazon Payments.
index	See search index.
index field	A name-value pair that is included in a search domain's index. An index field can contain text, literal, or unsigned integer data.
index field name	The name of a text, literal, or uint field in a search index.
indexing options	Configuration settings that define a search domain's index fields, how SDF data is mapped to those index fields, and how the index fields can be used.
instance	A copy of an Amazon Machine Image running as a virtual server in the AWS cloud.
instance family	A general instance type (p. 564) grouping using either storage or CPU capacity.
instance group	A Hadoop cluster contains one master instance group that contains one master node (p. 567), a core instance group containing one or more core node (p. 558) and an optional task node (p. 581) instance group, which can contain any number of task nodes.
instance store	Disk storage that is physically attached to the host computer for an EC2 instance, and therefore has the same lifespan as the instance. When the instance terminates, you lose any data in the instance store.
instance store-backed AMI	Instances launched from this type of AMI use an instance store volume as the root device. Compare this with instances launched from Amazon EBS-backed AMIs, which use an Amazon EBS volume as the root device.
instance type	A specification that defines the memory, CPU, storage capacity, and hourly cost for an instance. Some instance types are designed for standard applications, whereas others are designed for CPU-intensive, memory-intensive applications, and so on.
Instant Payment Notification	Also called IPN. A notification that Amazon Payments sends whenever a payment, refund, or reserved payment completes successfully or fails. The caller must provide Amazon Payments with an endpoint that we can send the IPN to. A notification (separate from the buyer redirect) that is sent whenever a payment, refund, or reserved payment completes successfully or fails. The developer must host this notification service and provide Amazon Simple Pay with an IPN response URL.
Internet gateway	Connects a network to the Internet. You can route traffic for IP addresses outside your VPC (p. 583) to the Internet gateway.
Internet Service Provider	A company that provides subscribers with access to the Internet. Many ISPs are also mailbox provider (p. 567)s. Mailbox providers are sometimes referred to as ISPs, even if they only provide mailbox services.
intrinsic function	A special action in a template that assigns values to properties not available until runtime. These functions follow the format *Fn::Attribute*, such as `Fn::GetAtt`. Arguments for intrinsic functions can be parameters, pseudo parameters, or the output of other intrinsic functions.

IP address — All EC2 instances are assigned two IP addresses at launch, which are directly mapped to each other through network address translation (NAT): a private IP address (following RFC 1918) and a public IP address. Instances launched in a VPC are assigned only a private IP address. Instances launched in your default VPC are assigned both a private IP address and a public IP address.

ISP — See Internet Service Provider.

issuer — The issuer is the person who writes a policy to grant permissions to a resource. The issuer (by definition) is always the resource owner. AWS does not permit Amazon SQS users to create policies for resources they don't own. If John is the resource owner, AWS authenticates John's identity when he submits the policy he's written to grant permissions for that resource.

item — Similar to rows on a spreadsheet, items represent individual objects that contain one or more value-attribute pairs.

item name — An identifier for an item. The identifier must be unique within the domain (p. 560).

J

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

job flow — A job flow specifies the complete processing of the data. It's comprised of one or more steps, which specify all of the functions to be performed on the data.

job ID — A five-character, alphanumeric string that uniquely identifies a storage device in your shipment. AWS issues the job ID in response to a `CREATE JOB` email command.

job prefix — The AWS Import/Export process generates a log file. The log file name always ends with the phrase *import-log-* followed by your Job ID. There is a remote chance that you already have an object with this name. To avoid a key collision, you can add an optional prefix to the log file.
See Also key prefix.

JSON — JavaScript Object Notation. A lightweight data-interchange format. For information about JSON, see http://www.json.org/.

junk folder — The location where email messages that various filters determine to be of lesser value are collected so that they do not arrive in the recipient (p. 574)'s inbox, but are still accessible to the recipient. This is also referred to as a spam (p. 579) or bulk folder.

K

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

key — A credential that identifies an AWS account or user to AWS (such as the AWS secret access key (p. 577)).
Amazon S3, Amazon EMR: The unique identifier for an object in a bucket. Every object in a bucket has exactly one key. Because a bucket and key together uniquely identify each object, you can think of Amazon S3 as a basic data map

between the *bucket + key*, and the object itself. You can uniquely address every object in Amazon S3 through the combination of the web service endpoint, bucket name, and key, for example:
`http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl`, where doc is the name of the bucket, and `2006-03-01/AmazonS3.wsdl` is the key.
AWS Import/Export: The name of an object in Amazon S3. It is a sequence of Unicode characters whose UTF-8 encoding cannot exceed 1024 bytes. If a key, for example, logPrefix + import-log-JOBID, is longer than 1024 bytes, AWS Elastic Beanstalk returns an `InvalidManifestField` error.
IAM: In the context of writing a policy (p. 571): A specific characteristic that is the basis for restricting access (such as the current time, or the IP address of the requester).

key pair

A set of security credentials you use to prove your identity electronically. A key pair consists of a private key and a public key.

key prefix

A logical grouping of the objects in a bucket (p. 555). The prefix value is similar to a directory name that enables you to store similar data under the same directory in a bucket.

L

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

launch configuration

A set of descriptive parameters used to create new EC2 instances in an Auto Scaling activity.

A template that an Auto Scaling group (p. 554) uses to launch new EC2 instances. The launch configuration contains information such as the Amazon Machine Image (p. 552) ID, the instance type, key pairs, security groups, and block device mappings, among other configuration settings.

launch permission

An Amazon Machine Image (p. 552) (AMI) attribute that allows users to launch an AMI.

lifecycle

The lifecycle state of the EC2 instance (p. 560) contained in an AutoScalingGroup. EC2 instances progress through several states over their lifespan; these include *Pending*, *InService*, *Terminating* and *Terminated*.

load balancer

A load balancer is a combination of a DNS name and a set of ports, which together provide a destination for all requests intended for your application. A load balancer can distribute traffic to multiple application instances across every Availability Zone (p. 554) within a region (p. 574). Load balancers can span multiple Availability Zones within an Amazon EC2 region, but they cannot span multiple regions.

logical name

A case-sensitive unique string within an AWS CloudFormation template that identifies a resource (p. 575), mapping (p. 567), parameter, or output. In an AWS CloudFormation template, each parameter, resource, property, mapping, and output must be declared with a unique logical name. You use the logical name when dereferencing these items using the `Ref` function.

M

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

machine utilization
: The amount of machine capacity used to complete a particular request (for example SELECT, GET, PUT, and so on), normalized to the hourly capacity of a standard processor. Machine utilization is measured in machine hour increments.

Mail Transfer Agent (MTA)
: Software that transports email messages from one computer to another by using a client-server architecture.

mailbox provider
: An organization that provides email mailbox hosting services. Mailbox providers are sometimes referred to as Internet Service Provider (p. 564)s, even if they only provide mailbox services.

mailbox simulator
: A set of email addresses that you can use to test an Amazon SES-based email sending application without sending messages to actual recipients. Each email address represents a specific scenario (such as a bounce or complaint) and generates a typical response that is specific to the scenario.

main route table
: The default route table that any new VPC subnet uses for routing. You can associate a subnet with a different route table of your choice. You can also change which route table is the main route table.

manifest
: When sending a *create job* request for an import or export operation you describe your job in a text file called a manifest. The manifest file is a YAML-formatted file that specifies how to transfer data between your storage device and the AWS cloud.

MapReduce
: See http://hadoop.apache.org/docs/r1.2.0/mapred_tutorial.html.

mapper
: An executable that splits the raw data into key/value pairs. The reducer uses the output of the mapper, called the *intermediate results*, as its input.

mapping
: A way to add conditional parameter values to an AWS CloudFormation template. You specify mappings in the template's optional Mappings section and retrieve the desired value using the `FN::FindInMap` function.

marker
: See pagination.

marketplace
: Amazon FPS: An environment in which the caller charges a fee for facilitating a transaction between a sender and a recipient.

  Amazon Simple Pay: A feature that allows a third party to charge for hosting a merchant's offers and facilitating payment.

Marketplace button
: An HTML-coded button to display and sell the goods of other sellers, optionally charging them a fee for the service.

master node
: A process running on an Amazon Machine Image (p. 552) that keeps track of the work its core and task nodes complete.

maximum price
: The maximum price you will pay to launch one or more Spot Instances. If your maximum price exceeds the current Spot Price (p. 579) and your restrictions are met, Amazon EC2 launches instances on your behalf.

maximum send rate	The maximum number of emails that you can send per second using Amazon SES.
member resources	See resource.
message ID	Amazon SES: A unique identifier that is assigned to every email message that is sent. Amazon SQS: The identifier returned when you send a message to a queue.
metadata	Amazon S3, Amazon EMR: A set of name/value pairs that describe the object. These include default metadata such as the date last modified and standard HTTP metadata such as Content-Type. Users can also specify custom metadata at the time they store an object. Amazon EC2: Data about an EC2 instance (p. 560) that the instance can retrieve to determine things about itself, such as, the instance type, the IP address, and so on.
metric	An element of time-series data defined by a unique combination of exactly one namespace, exactly one metric name, and between zero and ten dimensions. Metrics and the statistics derived from them are the basis of Amazon CloudWatch.
metric name	The primary identifier of a metric, used in combination with a namespace and optional dimensions.
MFA	See AWS Multi-Factor Authentication.
micro instance	A type of EC2 instance (p. 560) that is more economical to use if you have occasional bursts of high CPU activity.
MIME	See Multipurpose Internet Mail Extensions (MIME).
MTA	See Mail Transfer Agent (MTA).
Multi-AZ deployment	A primary DB instance (p. 558) that has a synchronous standby replica in a different Availability Zone (p. 554). The primary DB instance is synchronously replicated across Availability Zones to the standby replica.
Multi-Factor Authentication	See AWS Multi-Factor Authentication.
multi-use token	A usage-based payment instrument (p. 570) that allows the caller (p. 556) to charge the sender (p. 577) multiple times, without requiring the sender to repeatedly authorize the payments (compare to a single-use token (p. 578)).
Multi-Use Token API	The Co-Branded service (p. 557) API that creates a multi-use token (p. 568).
multi-valued attribute	An attribute with more than one value.
multipart upload	A feature that allows you to upload a single object as a set of parts.
Multipurpose Internet Mail Extensions (MIME)	An Internet standard that extends the email protocol to include non-ASCII text and non-text elements like attachments.
Multitool	A Cascading (p. 556) application that provides a simple command-line interface for managing large datasets.

N

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

namespace	An abstract container that provides context for the items (names, or technical terms, or words) it holds, and allows disambiguation of homonym items residing in different namespaces.
NAT	Network address translation.
NAT instance	An instance that is configured to perform NAT (p. 569) in a VPC. A NAT instance enables private instances in the VPC to initiate Internet-bound traffic without being directly reachable from the Internet.
network ACL	An optional layer of security that acts as a firewall for controlling traffic in and out of a subnet. You can associate multiple subnets with a single network ACL, but a subnet can be associated with only one network ACL at a time.
node	After an Amazon Machine Image (p. 552) is launched, the resulting running system is referred to as a node. All instances based on the same AMI are identical at start-up. Any information about the node is lost when the node terminates or fails.
NoEcho	A property of AWS CloudFormation parameters that will prevent the otherwise default reporting of names and values of a template parameter. Declaring the *`NoEcho`* property causes the parameter value to be masked with asterisks in the report by the `cfn-describe-stacks` command.
notification email	The email Amazon Payments sends to your customers to notify them of an upcoming price change you've scheduled.
null object	A null object is one whose version ID is null. Amazon S3 adds a null object to a bucket when versioning (p. 583) for that bucket is suspended. It is possible to have only one null object for each key in a bucket.

O

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

object	Amazon S3: The fundamental entity type stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3. CloudFront: Any entity that can be served either over HTTP or a version of RTMP.
on-demand instance	An Amazon EC2 pricing option that charges you for compute capacity by the hour with no long-term commitment.
one time payment	An Amazon FPS payment processed with a single-use token (p. 578). After the payment is made, that token is no longer valid.
operation	An API function. Also called an *action*.
order pipeline	The process that an order passes through between the time a customer selects an item and the time that customer's payment instrument (p. 570) is charged.

origin access identity	Also called OAI. A virtual identity you use when giving your distribution permission to fetch a private object from your origin server (Amazon S3 bucket).
origin server	The Amazon S3 bucket or custom origin containing the definitive original version of the content you deliver through CloudFront.
outbound notification	A response from Amazon Payments to your Amazon FPS (or Amazon Simple Pay) application via a Return URL or an Instant Payment Notification (IPN).

P

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

pagination	Some APIs that return a potentially large list of records can return a subset by using a value to set the maximum number of returned records. They then provide a marker, which identifies the last record returned so that in a subsequent call, the user can get the next sequence of records.
paid AMI	An Amazon Machine Image (AMI) that you sell to other Amazon EC2 users using Amazon DevPay.
part	In a multipart upload request, each part is a contiguous portion of the object's data.
passive authorization	A passive authorization happens when you inform customers at least 14 days in advance of a price change for your product, and they don't take any action; the price change is accepted automatically.
PAT	Port address translation.
payment instrument	The method of payment a customer chooses to use with Amazon Payments. These include credit cards (CC), Amazon Payments account balance (ABT), and bank account debits (ACH).
payment method failure	An error caused by an irregularity in the customer's chosen payment method, such as an insufficient bank balance or an expired credit card.
period	See sampling period.
permission	A statement within a policy (p. 571) that allows or disallows access to a particular resource. You can state any permission like this: "A has permission to do B to C where D applies." For example, Jane (A) has permission to read messages (B) from John's Amazon SQS queue (C), as long as she asks to receive only a maximum of 10 messages from the queue at a time (D). Whenever Jane sends a request to Amazon SQS to use John's queue, the service checks to see if she has permission and if the request satisfies the conditions John set forth in the permission.
persistent identifier	Also called PID. An encoded string that represents the relationship between a customer and the owner of Amazon DevPay products. After a customer purchases one of your products, you can use the PID to confirm the status of the customer's subscription to the product.
persistent storage	A long-term data storage solution. Options within AWS are: Amazon S3, Amazon EBS, and Amazon SimpleDB.

physical name	A unique label AWS CloudFormation assigns to each resource when creating a stack (p. 579). Some AWS CloudFormation commands accept the physical name as a value with the `--physical-name` parameter.
Pig	An open-source Apache library that runs on top of Hadoop. The library takes SQL-like commands written in a language called Pig Latin and converts those commands into MapReduce job flows.
policy	A policy is the formal description of the permissions for a resource. The access policy language distinguishes between a *policy* and a *statement*. A policy is the complete document that can contain many different permissions for a given resource. A statement is the description of an individual permission. Therefore a policy can contain multiple statements. For example, a policy could specify that Jane can use John's queue (one statement), and Bob cannot use John's queue (another statement). Auto Scaling: An object that stores the information needed to launch or terminate instances for an Auto Scaling group. Executing the policy causes instances to be launched or terminated. You can configure an alarm (p. 551) to invoke an Auto Scaling policy.
postpaid credit instrument	A payment instrument (p. 570) that is like a credit card used to make incremental purchases on your website. As purchases are made, the accumulated debt on the postpaid credit instrument increases until a credit limit is reached, or until you have arranged to make a settlement of the debt, such as a monthly payment.
postpaid payment token	A payment token used whenever a buyer (p. 555) wants to make a purchase using a postpaid credit instrument (p. 571).
Postpaid Token API	The Co-Branded service (p. 557) API that creates a postpaid payment token (p. 571) that a sender uses as a payment instrument (p. 570) and then funds, much like a credit card.
pre-signed URL	A URL that uses query string authentication (p. 573).
prefix	See job prefix.
Premium Support	A one-on-one, fast-response support channel that AWS customers can subscribe to for support for AWS infrastructure services. See Also https://amazonaws.cn/premiumsupport/.
prepaid instrument	A payment instrument (p. 570) that is like a gift card with an associated prepaid balance. A buyer (p. 555) can purchase a prepaid instrument on your website and use it to make incremental payments over a period of time, in line with whatever constraints that you or the buyer previously set up.
prepaid payment token	A payment token used whenever a buyer (p. 555) wants to make a purchase using a prepaid instrument (p. 571).
Prepaid Token API	The Co-Branded service (p. 557) API that creates a prepaid payment token (p. 571) that a sender (p. 577) funds and then uses as a payment instrument (p. 570).
principal	The principal is the person or persons who receive the permission in the policy (p. 571). The principal is A in the statement "A has permission to do B to C where D applies." In a policy, you can set the principal to "anyone" (that is, you can specify a wildcard to represent all people). You might do this, for example, if you don't want to restrict access based on the actual identity of the requester, but instead on some other identifying characteristic such as the requester's IP address.

	The concept of principals doesn't apply to a IAM policy, because these policies are attached to users or groups.
private IP address	All EC2 instances are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC are assigned only a private IP address.
private subnet	A VPC subnet whose instances cannot be reached from the Internet.
product activation	See activation.
product code	The product code is an eight-character string that identifies your registered product to AWS.
product identification token	See product token.
product token	The product token is a long encoded string that identifies the product to AWS. You might also see the product token referred to as the *product identification token*.
properties	See resource property.
property rule	A JSON (p. 565)-compliant markup standard for declaring properties, mappings, and output values in an AWS CloudFormation template.
provisioned IOPS	A storage option designed to deliver fast, predictable, and consistent I/O performance. When you specify an IOPS rate while creating a DB Instance, Amazon RDS provisions that IOPS rate for the lifetime of the DB Instance.
pseudo parameter	A predefined setting, such as `AWS:StackName` that can be used in AWS CloudFormation templates without having to declare them. You can use pseudo parameters anywhere you can use a regular parameter.
public AMI	An Amazon Machine Image (p. 552) that all AWS accounts have permission to launch.
public data set	A large set of public data that can be seamlessly integrated into AWS cloud-based applications. Amazon stores public data sets at no charge to the community and, like all AWS services, users pay only for the compute and storage they use for their own applications. These data sets currently include data from the Human Genome Project, the U.S. Census, Wikipedia, and other sources. See Also http://amazonaws.cn/publicdatasets.
public IP address	All EC2 instances are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC are assigned only a private IP address.
public subnet	A subnet whose instances can be reached from the Internet.
purchase URL	The URL your customers use to purchase your product. When you advertise your product, you provide the purchase URL as the sign-up link for customers to use.

Q

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

Qualification	A property associated with a worker (p. 584) that represents that worker's skill, ability, or reputation. A Requester (p. 575) can use Qualifications to control which workers can perform HITs. Each worker can have multiple Qualifications.
Qualification requirements	A Human Intelligence Task (p. 563) can have Qualification requirements that a worker's Qualifications (q.v.) must meet before the worker can accept that HIT.
Qualification test	A form, similar to a HIT, containing a set of questions that the worker must complete successfully to receive a particular Qualification (p. 573).
Qualification type	Just as each worker (p. 584) has one or more Qualification (p. 573), each Human Intelligence Task (p. 563) has one or more Qualification type. These types specify what Qualifications the worker must have.
Query	A type of HTTP-based request interface that generally uses only the GET or POST HTTP method and a query string with parameters. See Also REST, REST-Query.
query string authentication	An AWS feature that lets you place the authentication information in the HTTP request query string instead of in the Authorization header. For example: with Amazon DevPay, query string authentication enables your product to give anyone easy, URL-based access to objects in the customer's bucket.
query time rank expression	A rank expression (p. 573) that's defined within a search request. You can use a query time rank expression to rank results for a request or set a threshold for the search results.
queue	A sequence of messages or jobs held in temporary storage awaiting transmission or processing.
queue URL	A URL that uniquely identifies a queue.
quota	Amazon RDS: The maximum number of DB instance (p. 558)s and available storage you can use. ElastiCache: The maximum number of the following items: • The number of cache clusters for each AWS account • The number of cache nodes per cache cluster • The total number of cache nodes per AWS account across all cache clusters created by that AWS account

R

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

range GET	A range GET specifies a byte range of data to get for a download. If an object is large, you can break up a download into smaller units by sending multiple range GET requests that each specify a different byte range to GET.
rank expression	A numeric expression that you can use to control how search hits are ranked. You can construct rank expressions using uint fields, other rank expressions, a document's default text_relevance score, and standard numeric operators and functions. When you use the `rank` option to specify a rank expression in a search

	request, the expression is evaluated for each search hit and the hits are listed according to their rank expression values.
raw email	A type of *sendmail* request that allows you to specify the email headers and MIME types.
RDS	See Amazon Relational Database Service.
read replica	An active copy of another DB instance. Any updates to the data on the source DB instance are replicated to the read replica DB instance using the built-in replication feature of MySQL 5.1.
receipt handle	An identifier you get when you receive a message from the queue. This identifier is required to delete a message from the queue or when changing a message's visibility timeout.
receiver	The entity that consists of the network systems, software, and policies that manage email delivery for a recipient (p. 574).
recipient	Amazon FPS: A seller who receives a payment from a buyer (sender) in exchange for a service or product. Amazon SES: The person or entity receiving an email message. For example, a person named in the "To" field of a message.
recipient token	A general term to cover all types of payment tokens associated with recipients (for example, single-use token, multi-use token, and recurring-use token).
recurring payment	An Amazon FPS payment processed with a recurring payment token. Payments can occur periodically using the same payment token. The token is valid until it expires.
recurring-use token	A type of multi-use token (p. 568) that is restricted to a predetermined fixed amount and a regular payment interval.
Recurring-Use Token API	The Co-Branded service (p. 557) API that creates a recurring-use token (p. 574).
redirect URL	The page on your own website that you want customers to see at the end of the purchase process for your product. You provide the URL when you register the product with Amazon DevPay.
reducer	An executable in the MapReduce process that uses the intermediate results from the mapper and processes them into the final output.
reference	A means of inserting a property from one AWS resource into another. For example, you could insert an Amazon EC2 security group property into an Amazon RDS resource.
region	A named set of AWS resources in the same geographical area. A region comprises at least two Availability Zones.
reply path	The email address to which an email reply is sent. This is different from the return path (p. 576).
reputation	1. An Amazon SES metric, based on factors that might include bounces, complaints, and other metrics, regarding whether or not a customer is sending high-quality emails. 2. A measure of confidence, as judged by an Internet Service Provider (p. 564) or other entity that an IP address that they are receiving emails from is not the source of spam (p. 579).

requester	A requester is a person who sends a request to an AWS service and asks for access to a particular resource. The requester sends a request to AWS that essentially says: "Can A do B to C where D applies?" In this question, the requester is A. See Also Requester.
Requester	(Note capitalization) A company, organization, or person that creates and submits tasks (a Human Intelligence Task (p. 563)) to Mechanical Turk; for workers (p. 584) to perform.
Requester Pays	An Amazon S3 feature that allows a bucket owner (p. 555) to specify that anyone who requests access to objects in a particular bucket must pay the data transfer and request costs.
reservation	A collection of EC2 instances started as part of the same launch request. Not to be confused with a Reserved Instance (p. 575).
reserve	The amount that is held in reserve against a credit card, but not charged. Later, the transaction is settled, that is charged (typically after the product is actually shipped).
Reserved Instance	A pricing option that lets you make a low, one-time payment for each instance to reserve and receive a significant discount on the hourly usage charge for that instance.
Reserved Instance Marketplace	Matches sellers who have reserved capacity that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instances that you purchase from third-party sellers will have less than a full standard term remaining and can be sold at different upfront prices. The usage or reoccurring fees will remain the same as the fees set when the Reserved Instances were originally purchased. Full standard terms for Reserved Instances available from AWS run for one year or three years.
resource	1. The objects you work with on AWS. This includes buckets, domains, instances, queues, and so on. 2. Tools, code, and documents that AWS provides to support users. 3. An object that the principal (p. 571) requests access to. The resource is C in the statement "A has permission to do B to C where D applies." 4. A required element of an AWS CloudFormation stack (p. 579). Each stack contains at least one resource, such as an Auto Scaling LaunchConfiguration. All resources in a stack must be created successfully for the stack to be created.
resource property	A value required when including an AWS resource in an AWS CloudFormation stack (p. 579). Each resource may have one or more properties associated with it. For example, an `AWS::EC2::Instance` resource may have a `UserData` property. In an AWS CloudFormation template, resources must declare a properties section, even if the resource has no properties.
resource record	Also called *resource record set*. Standard DNS terminology. See Also http://en.wikipedia.org/wiki/Domain_Name_System.
REST	A type of HTTP-based request interface that generally uses only the GET or POST HTTP method and a query string with parameters. Sometimes known as Query. In some implementations of a REST interface, other HTTP verbs besides GET and POST are used.

REST-Query	Also known as Query or HTTP Query. This is a type of HTTP request that generally uses only the GET or POST HTTP method and a query string with parameters. Compare this with REST, which is a type of HTTP request that uses any HTTP method (GET, DELETE, POST, etc.), a resource, HTTP headers, and possibly a query string with parameters.
result enabled	An index field option that enables a text or literal field's values to be returned in the search results.
return path	The email address to which bounced emails are returned. The return path is specified in the header of the original email. This is different from the reply path (p. 574).
reward	The money a Requester (p. 575) pays a worker (p. 584) for satisfactory work done on the Requester's Human Intelligence Task (p. 563)s.
rollback	A return to a previous state that follows the failure to create an object, such as AWS CloudFormation stack (p. 579). All resources associated with the failure are deleted during the rollback. For AWS CloudFormation, you can override this behavior using the `--disable-rollback` option on the command line.
root device volume	Contains the image used to boot the instance. If you launched the instance from an AMI backed by instance store, this is an instance store volume created from a template stored in Amazon S3. If you launched the instance from an AMI backed by Amazon EBS, this is an Amazon EBS volume created from an Amazon EBS snapshot.
route table	A set of routing rules that controls the traffic leaving any subnet that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.

S

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

sampling period	A defined duration of time, such as one minute, over which CloudWatch computes a statistic (p. 579).
sandbox	A testing location where you can test the functionality of your application without affecting production, incurring charges, or purchasing products. Amazon SES: An Amazon SES environment that is designed for developers to test and evaluate the service. In the sandbox, you have full access to the Amazon SES API, but you can only send messages to verified email addresses and the mailbox simulator. To get out of the sandbox, you need to apply for production access. Accounts in the sandbox also have lower sending limits (p. 578) than production accounts.
scaling activity	A process that changes the size, configuration, or makeup of an Auto Scaling group (p. 554) by launching or terminating instances. For more information, see Auto Scaling Concepts in the Auto Scaling Developer Guide.
SDF	See Search Data Format (SDF).
search API	The API that you use to submit search requests to an Amazon CloudSearch domain.

Search Data Format (SDF)	The format that you use to describe the data to add or delete from an Amazon CloudSearch domain. SDF can be represented as either JSON (p. 565) or XML.
search domain	Encapsulates your searchable data and the search instances that handle your search requests. You set up a separate domain for each different collection of data that you want to search.
search domain configuration	A search domain's indexing options, text options, access policies, and rank expressions.
search domain name	A user-specified name that is used to construct a unique identifier for a search domain.
search enabled	An search index field option that enables the field data to be searched.
search index	A representation of your searchable data that facilitates fast and accurate data retrieval.
search instance	A compute resource that indexes your data and processes search requests. A search domain has one or more search instances, each with a finite amount of RAM and CPU resources. As your data volume grows, more search instances or larger search instances are deployed to contain your indexed data. When necessary, your index is automatically partitioned across multiple search instances. As your request volume or complexity increases, each search partition is automatically replicated to provide additional processing capacity.
search request	A request that is sent to a search domain to retrieve documents that match particular search criteria.
search result	A document that matches a search request. Also referred to as a *search hit*.
search service endpoint	The URL that you connect to when sending search requests to a search domain.
secret access key	A key that Amazon Web Services assigns to you when you sign up for an AWS account. Sometimes called simply a "secret key."
security group	A named set of allowed inbound network connections for an instance. (Security groups in Amazon VPC also include support for outbound connections.) Each security group consists of a list of protocols, ports, and IP address ranges. A security group can apply to multiple instances, and multiple groups can regulate a single instance.
seller	Amazon FPS, Amazon Simple Pay: A seller receives money from a buyer in exchange for a service or product. Amazon Simple Pay: Individual who receives a payment from a buyer using an Amazon Simple Pay button. The seller receives money from a buyer in exchange for a service or product.
sender	Amazon FPS: A sender (also known as the buyer) pays a recipient for a product or service. Amazon SES: The person or entity sending an email message.
Sender ID	A Microsoft-controlled version of SPF. An email authentication and anti-spoofing system. For more information about Sender ID, go to http://wikipedia.org/wiki/Sender_ID.

sender token	A general term to cover all types of payment tokens that can be associated with a sender (for example, single-use token, recurring use token, postpaid token, and so on).
sending limits	The sending quota (p. 578) and maximum send rate (p. 568) that are associated with every Amazon SES account.
sending quota	The maximum number of emails that you can send using Amazon SES in a 24-hour period.
service endpoint	See endpoint.
server-side signature verification	Method used to validate Instant Payment Notification (IPN) and Return URL responses.
service health dashboard	A web page showing up-to-the-minute information about AWS service availability. The dashboard is located at http://status.amazonaws.cn.
settle	To complete a transaction that has been reserved. If you don't charge the sender (p. 577) immediately when the purchase is initiated (and instead reserve the amount against the sender's credit card), you settle the transaction later, typically after you ship the product to the sender. In FPS, *Settling* moves the reserved amount from the sender to the recipient. Amazon Simple Pay does not support settling purchases. You must use the Amazon Simple Pay API `Settle` to implement that functionality.
settlement token	A payment token used to settle the debt accumulated with a postpaid credit instrument (p. 571).
SHA	Secure Hash Algorithm. SHA1 is an earlier version of the algorithm, which AWS has deprecated in favor of SHA256.
shared AMI	An Amazon Machine Image (p. 552) that a developer builds and makes available for others to use.
shutdown action	A predefined bootstrap action that launches a script that executes a series of commands in parallel before terminating the job flow.
signature	Refers to a *digital signature*, which is a mathematical way to confirm the authenticity of a digital message. AWS uses signatures to authenticate the requests you send to our web services. For more information, to http://amazonaws.cn/security.
SIGNATURE file	A file you copy to the root directory of your storage device. The file contains a job ID, manifest file, and a signature.
Simple Mail Transfer Protocol	See SMTP.
Single-AZ DB Instance	A standard (non-Multi-AZ) DB instance (p. 558) that is deployed in one Availability Zone (p. 554), without a standby replica in another Availability Zone. See Also Multi-AZ deployment.
single-use token	A payment instrument (p. 570) that allows the caller (p. 556) to charge the sender (p. 577) only once (compare to a multi-use token (p. 568)).
Single-Use Token API	The Co-Branded service (p. 557) API that creates a single-use token (p. 578).
single-valued attribute	An attribute with one value.

SMTP	Simple Mail Transfer Protocol. The standard that is used to exchange email messages between internet hosts for the purpose of routing and delivery.
snapshot	Amazon Elastic Block Store (p. 552) creates *snapshots* or backups of your volumes and stores them in Amazon S3. You can use these snapshots as the starting point for new Amazon EBS volumes or to protect your data for long-term durability.
soft bounce	A temporary email delivery failure such as "mailbox full."
software VPN	A software appliance-based VPN connection over the Internet.
source	An SDF document field that is used to populate a search index field.
source/destination checking	A security measure to verify that an EC2 instance is the origin of all traffic that it sends and the ultimate destination of all traffic that it receives, that is, that the instance is not relaying traffic. Source/destination checking is enabled by default. For instances that function as gateways, such as VPC NAT instances, source/destination checking must be disabled.
spam	Unsolicited bulk email.
spamtrap	An email address that is set up by an anti-spam (p. 579) entity, not for correspondence, but to monitor unsolicited email. This is also called a *honeypot*.
SPF	Sender Policy Framework. A standard for authenticating email. See Also http://www.openspf.org.
Spot Instance	A type of EC2 instance (p. 560) that you can bid on to take advantage of unused Amazon EC2 capacity.
Spot Price	The price for a Spot Instance (p. 579) at any given time. If your maximum price exceeds the current price and your restrictions are met, Amazon EC2 launches instances on your behalf.
stack	AWS CloudFormation: A collection of AWS resources you create and delete as a single unit. AWS OpsWorks: A set of instances you manage collectively, typically because they have a common purpose such as serving PHP applications. A stack serves as a container and handles tasks that apply to the group of instances as a whole, such as managing applications and cookbooks.
Standard button	An HTML-coded button to offer Amazon Simple Pay as a standalone payment method for one-time purchases.
station	A place at an AWS facility where we transfer your AWS Import/Export data on to, or off of, your storage device.
statistic	One of five functions of the values submitted for a given sampling period (p. 576). These functions are "Maximum", "Minimum," "Sum," "Average," and "SampleCount."
stem	The common root or substring shared by a set of related words.
stemming	The process of mapping related words to a common stem. This enables matching on variants of a word. For example, a search for "horse" could return matches for horses, horseback, and horsing, as well as horse.
stemming dictionary	A domain-specific collection of mappings of words to their stems. Amazon CloudSearch does not define a default stemming dictionary.

step	A single function applied to the data in a job flow (p. 565). The sum of all steps comprises a job flow.
step type	The type of work done in a step. There are a limited number of step types, such as moving data from Amazon S3 to Amazon EC2 or from Amazon EC2 to Amazon S3.
sticky session	A feature of the load balancer that binds a user's session to a specific application instance so that all requests coming from the user during the session are sent to the same application instance. By contrast, a load balancer defaults to route each request independently to the application instance with the smallest load.
stopping	The process of filtering stop words from an index or search request.
stopword	A word that is not indexed and is automatically filtered out of search requests because it is either insignificant or so common that including it would result in too many matches to be useful. Stop words are language-specific.
stopword dictionary	A domain-specific collection of stopwords. Amazon CloudSearch defines a default stopword dictionary for English that you can use as-is, or customize to suit your collection of data.
streaming	Amazon EMR: A utility that comes with Hadoop that enables you to develop MapReduce executables in languages other than Java. CloudFront: The ability to use a media file in real time—as it is transmitted in a steady stream from a server.
streaming distribution	A special kind of distribution (p. 559) that serves streamed media files using a Real Time Messaging Protocol (RTMP) connection.
string-to-sign	Before you calculate an HMAC signature, you first assemble the required components in a canonical order. The pre-encrypted string is the string-to-sign.
subnet	A segment of the IP address range of a VPC (p. 583) that EC2 instances can be attached to. You can create subnets to group instances according to security and operational needs.
Subscription button	An HTML-coded button that enables an easy way to charge customers a recurring fee.
supported AMI	An Amazon Machine Image (p. 552) similar to a paid AMI (p. 570), except that the owner charges for additional software or a service that customers use with their own AMIs.
synchronous bounce	A type of bounce (p. 555) that occurs while the email servers of the sender (p. 577) and receiver (p. 574) are actively communicating.
synonym	A word that is the same or nearly the same as an indexed word and that should produce the same results when specified in a search request. For example, a search for "Rocky Four" or "Rocky 4" should return the fourth *Rocky* movie. This can be done by designating that `four` and `4` are synonyms for `IV`. Synonyms are language-specific.
synonym dictionary	A domain-specific collection of synonym mappings. Amazon CloudSearch does not define a default synonym dictionary.
system Qualifications	The set of Qualifications (p. 573) that represent a worker's (p. 584) history and reputation. The Mechanical Turk system assigns these Qualifications to each worker, and continuously updates the values as they use the system.

T

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

tag	Metadata (consisting of up to 10 key/value pairs) that you can define and assign to Amazon EC2 resources.
tagging	Also called *labeling*. A way to format return path (p. 576) email addresses so that you can specify a different return path for each recipient of a message. Tagging enables you to support VERP (p. 583). For example, if Andrew manages a mailing list, he can use the return paths andrew+recipient1@example.net and andrew+recipient2@example.net so that he can determine which email bounced.
task node	An EC2 instance (p. 560) that runs Hadoop map and reduce tasks, but does not store data. Task nodes are managed by the master node (p. 567), which assigns Hadoop tasks to nodes and monitors their status. While a job flow is running you can increase and decrease the number of task nodes. Because they don't store data and can be added and removed from a job flow, you can use task nodes to manage the EC2 instance capacity your job flow uses, increasing capacity to handle peak loads and decreasing it later. Task nodes only run a TaskTracker Hadoop daemon.
tebibyte	A contraction of tera binary byte, a tebibyte is 2^40 bytes or 1,099,511,627,776 bytes. A terabyte is 10^12 or 1,000,000,000,000 bytes.
template format version	The version of an AWS CloudFormation template design that determines the available features. If you omit the `AWSTemplateFormatVersion` section from your template, AWS CloudFormation assumes the most recent format version.
template validation	The process of confirming the use of JSON (p. 565) code in an AWS CloudFormation template. You can validate any AWS CloudFormation template using the `cfn-validate-template` command.
text options	Domain-specific stopword, stemming, and synonym dictionaries used during text processing when building a search index. Stopwords and stems are also used at search time to process the search terms before looking for matching documents in the index.
text_relevance	A built-in relevance score that's based on the repetition of search terms in the document and proximity of search terms to each other in each matching index field in the document. A document's text_relevance score is an integer value from 0 to 1000 (inclusive).
throttling	The means by which Amazon SES rejects your attempts to send email because you have exceeded your sending limits (p. 578).
time series data	Data provided as part of a metric. The time value is assumed to be when the value occurred. A metric is the fundamental concept for CloudWatch and represents a time-ordered set of data points. You publish metric data points into CloudWatch and later retrieve statistics about those data points as a time-series ordered data set.
time stamp	A date/time string in ISO 8601 format.
TLS	See Transport Layer Security.

tokenization	Part of the text processing that Amazon CloudSearch performs when indexing and processing search requests. During indexing, the contents of each text field are split into a collection of tokens that can be indexed separately. Punctuation is stripped and each word (that isn't in the stopword list) becomes a token. For example, the string "spider-man" would be split into two tokens: *spider* and *man*. At search time, the search terms are tokenized using the same rules before being matched against the indexed tokens.
topic	A communication channel to send messages and subscribe to notifications. It provides an access point for publishers and subscribers to communicate with each other.
Transport Layer Security	A cryptographic protocol that provides security for communication over the Internet. Its predecessor is Secure Sockets Layer (SSL).
trusted signers	AWS accounts that the CloudFront distribution owner has given permission to create signed URLs for a distribution's content.
tuning	Selecting the number and type of AMIs (p. 552) to run a Hadoop job flow most efficiently.
tunnel	A route for transmission of private network traffic that uses the Internet to connect nodes in the private network. The tunnel uses encryption and secure protocols such as PPTP to prevent the traffic from being intercepted as it passes through public routing nodes.

U

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

unbounded	The number of potential occurrences is not limited by a set number. This value is often used when defining a data type that is a list (for example, `maxOccurs="unbounded"`), in Web Services Description Language (p. 584).
unit	Standard measurement for the values submitted to CloudWatch as metric data. Units include Seconds, Percent, Bytes, Bits, Count, Bytes/Second, Bits/Second, Count/Second, and None.
usage report	An AWS report giving details of your usage of a particular AWS service. You can generate and download usage reports from http://amazonaws.cn/usage-reports.
user	A person or application under an account (p. 550) that needs to make API calls to AWS products. Each user has a unique name within the AWS account, and a set of security credentials not shared with other users. These credentials are separate from the AWS account's security credentials. Each user is associated with one and only one AWS account.
user token	A customer credential returned to your product during product activation (p. 551). The user token is a long, encoded string that AWS uses to identify the customer. Your product provides the customer's user token in each request for Amazon S3 the product makes on behalf of the customer. Every user token generated for a particular customer differs from the others because the creation time is one of the items making up the token value.

V

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

validation
: See template validation.

value
: Instances of attributes (p. 554) for an item, such as cells in a spreadsheet. An attribute might have multiple values.

value-add
: The amount you charge each customer on top of the cost of the AWS services they used.

Variable Envelope Return Path
: See VERP.

verification
: The process of confirming that you own an email address or a domain so that you can send emails from or to it.

VERP
: Variable Envelope Return Path. A way in which email sending applications can match bounced emails with the undeliverable address that caused the bounce by using a different return path (p. 576) for each recipient. VERP is typically used for mailing lists. With VERP, the recipient's email address is embedded in the address of the return path, which is where bounced emails are returned. This makes it possible to automate the processing of bounced emails without having to open the bounce messages, which may vary in content.

versioning
: Every object in Amazon S3 has a key and a version ID. Objects with the same key, but different version IDs can be stored in the same bucket. Versioning is enabled at the bucket layer using PUT Bucket versioning.
See Also document version.

virtual private cloud
: See VPC.

virtual private gateway
: See VPG.

visibility timeout
: The period of time that a message is invisible to the rest of your application after an application component gets it from the queue. During the visibility timeout, the component that received the message usually processes it, and then deletes it from the queue. This prevents multiple components from processing the same message.

VPC
: Virtual private cloud. An elastic network populated by infrastructure, platform, and application services that share common security and interconnection.

VPG
: Virtual private gateway. The Amazon side of a VPN connection that maintains connectivity. The internal interfaces of the virtual private gateway connect to your VPC via the VPN attachment and the external interfaces connect to the VPN connection, which leads to the customer gateway.

VPN CloudHub
: See AWS VPN CloudHub.

VPN connection
: Although VPN connection is a general term, we specifically mean the IPsec connection between a VPC (p. 583) and some other network, such as a corporate data center, home network, or co-location facility.

W

Numbers and Symbols (p. 550) | A (p. 550) | B (p. 555) | C (p. 556) | D (p. 558) | E (p. 560) | F (p. 562) | G (p. 562) | H (p. 563) | I (p. 563) | J (p. 565) | K (p. 565) | L (p. 566) | M (p. 567) | N (p. 569) | O (p. 569) | P (p. 570) | Q (p. 572) | R (p. 573) | S (p. 576) | T (p. 581) | U (p. 582) | V (p. 583) | W (p. 584) | X, Y, Z (p. 584)

Web Services Description Language	A language used to describe the actions that a web service can perform, along with the syntax of action requests and responses. Your SOAP or other toolkit interprets a WSDL file to provide your application access to the actions provided by the web service. For most toolkits, your application calls a service action using routines and classes provided or generated by the toolkit.
website owner	A developer who uses Amazon FPS, such as by creating an Amazon Simple Pay button.
worker	A person who performs the tasks specified by a Requester (p. 575) in a Human Intelligence Task (p. 563).

X, Y, Z

No entries