Amazon Route 53

Developer Guide

API Version 2011-05-05

Amazon Route 53: Developer Guide

Copyright © 2012 Amazon Web Services LLC or its affiliates. All rights reserved.

Welcome 1
What is Route 53 and How Does it Work? 2
Hosted Zones 2
DNS Domain Name Format 3
Supported DNS Resource Record Types 4
NS and SOA Records that Route 53 Creates for a Hosted Zone 6
Limits on Route 53 API Requests and Entity Counts 8
DNS Constraints and Behaviors 8
Route 53 Pricing 9
AWS Identity and Access Management 9
Getting Started with Route 53 10
Getting Started: Creating a Domain that Uses Route 53 10
How to Use the Route 53 Console, API, AWS SDKs, and Command-Line Tool 14
    The Route 53 Console 14
    The Route 53 API 15
    AWS SDKs that Support Route 53 15
    The dnscurl.pl Command-Line Tool 15
Creating a Domain that Uses Route 53 as the DNS Service 17
Migrating an Existing Domain to Route 53 22
Creating a Subdomain that Uses Route 53 without Migrating the Parent Domain 27
Migrating a Subdomain to Route 53 without Migrating the Parent Domain 32
Working with Hosted Zones 37
Creating a Hosted Zone 37
Getting the Name Servers for a Hosted Zone 39
Listing the Hosted Zones for an AWS Account 41
Deleting a Hosted Zone 45
Working with Resource Record Sets 49
Creating, Changing, and Deleting Resource Record Sets 49
Listing Resource Record Sets 56
Creating Weighted Resource Record Sets 61
Using Weighted Resource Record Sets with the 2010-10-01 Route 53 API 65
Creating Alias Resource Record Sets for Elastic Load Balancing 66
How to Create an Alias Resource Record Set 67
How to Create Weighted Alias Resource Record Sets 72
Using Aliases with the 2010-10-01 Route 53 API 76
Routing Queries to a Website That Is Hosted in an Amazon S3 Bucket 77
Controlling User Access with IAM 79
Making API Requests 83
Endpoints 83
REST Requests 84
REST Responses 86
    Request ID 86
Authenticating REST Requests 88
Example: Using dnscurl.pl 91
Additional Route 53 Resources 100
Third-Party Tools and Libraries 102
Document History 105

Welcome

The *Amazon Route 53 Developer Guide* gives developers an overview of how Route 53 works as a DNS service, explains how to use the Route 53 console and the Route 53 API to create new domains and subdomains that use Route 53 as the DNS service, and how to migrate existing domains and subdomains to Route 53, explains how to work with hosted zones and resource record sets, and explains how to make API requests.

How Do I...?

How Do I?	Relevant Topic
Get Started	Getting Started with Route 53 (p. 10)
Understand whether Route 53 is right for my use case	Amazon Route 53 detail page
Use the Route 53 console	Getting Started: Creating a Domain that Uses Route 53 (p. 10)
Create a new domain that uses Route 53 using either the console or the API	Creating a Domain that Uses Route 53 as the DNS Service (p. 17)
Migrate an existing domain to Route 53 using either the console or the API	Migrating an Existing Domain to Route 53 (p. 22)
Make API requests	Making API Requests (p. 83)
Get reference information about the Route 53 API	Amazon Route 53 API Reference

What is Route 53 and How Does it Work?

Route 53 is a scalable Domain Name System (DNS) web service. It provides secure and reliable routing to your infrastructure that uses Amazon Web Services (AWS) products, such as Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing, or Amazon Simple Storage Service (Amazon S3). You can also use Route 53 to route users to your infrastructure outside of AWS.

Route 53 is an authoritative DNS service, meaning it translates friendly domains names like www.example.com into IP addresses like 192.0.2.1. Route 53 responds to DNS queries using a global network of authoritative DNS servers, which reduces latency. For a list of the locations of Route 53 DNS servers, see The Amazon Route 53 Global Network on the Amazon Route 53 detail page.

You can manage your DNS records through the Route 53 console or the Route 53 API, or set account-level user and access management through the AWS Identity and Access Management (IAM) API.

See the following topics for more information about how Route 53 works.

Topics

- Hosted Zones (p. 2)
- DNS Domain Name Format (p. 3)
- Supported DNS Resource Record Types (p. 4)
- NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6)
- Limits on Route 53 API Requests and Entity Counts (p. 8)
- DNS Constraints and Behaviors (p. 8)
- Route 53 Pricing (p. 9)
- AWS Identity and Access Management (p. 9)

Hosted Zones

A hosted zone is a collection of resource record sets hosted by Route 53. Like a traditional DNS zone file, a hosted zone represents a collection of resource record sets that are managed together under a single domain name. Each hosted zone has its own metadata and configuration information.

The resource record sets contained in a hosted zone must share the same suffix. For example, the example.com hosted zone can contain resource record sets for www.example.com and www.aws.example.com subdomains, but cannot contain resource record sets for a www.example.ca subdomain.

You can use the Route 53 console or API to create, list, modify, and delete hosted zones and their resource record sets. The following table describes the actions you can perform on a Route 53 hosted zone and provides links to how-to and reference topics.

Action	For More Information
Create a hosted zone	For how-to information, see Creating a Hosted Zone (p. 37). For reference information about the API action, see POST CreateHostedZone.
Get information about the name servers for your hosted zone	For how-to information, see Getting the Name Servers for a Hosted Zone (p. 39). For reference information about the API action, see GET GetHostedZone.
Delete a hosted zone	For how-to information, see Deleting a Hosted Zone (p. 45). For reference information about the API action, see DELETE DeleteHostedZone.
List your hosted zones	For how-to information, see Listing the Hosted Zones for an AWS Account (p. 41). For reference information about the API action, see GET ListHostedZones.

DNS Domain Name Format

A DNS domain name consists of a series of labels separated by dots. Each label can be up to 63 bytes long. The total length of a domain name cannot exceed 255 bytes including the dots. Route 53 supports any valid domain name.

You can use any ASCII character from 0 to 255 decimal, but you must use escape codes in the format `\three-digit octal code` to use any of the following characters:

- Characters 000 to 040 octal (0x00 to 0x20 hexadecimal)
- Characters 177 to 377 octal (0x7F to 0xFF hexadecimal)
- . (period), character 056 octal (0x2E hexadecimal), when used as a character in a domain name. When using . as a delimiter between labels, you do not need to use an escape code.

For example, to create a hosted zone for the subdomain fübar under the domain example.com, you would specify `f\374bar.example.com`.

For a list of ASCII characters and the corresponding octal codes, do an Internet search on "ascii table".

You can include any of the following characters without using escape codes:
0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } ~ .

Note

For alphabetic characters, regardless of whether you specify upper-case letters, lower-case letters, or the corresponding letters in escape codes, Route 53 stores them internally as lower-case letters.

Supported DNS Resource Record Types

Route 53 supports the DNS resource record types that are listed in this section. Each record type also includes an example of how to format the `Value` element when you are accessing Route 53 using the API.

Note

For resource record types that include a domain name, enter a fully qualified domain name, for example, *www.example.com*. The trailing dot is optional; Route 53 assumes that the domain name is fully qualified. This means that Route 53 treats *www.example.com* (without a trailing dot) and *www.example.com.* (with a trailing dot) as identical.

A Format

An A record `Value` element must take the format of an IPv4 address in dotted decimal notation.

Example

```
<Value>192.0.2.1</Value>
```

AAAA Format

An AAAA record `Value` element must take the format of an IPv6 address, in colon-separated hexadecimal format.

Example

```
<Value>2001:0db8:85a3:0:0:8a2e:0370:7334</Value>
```

CNAME Format

A CNAME `Value` element is the same format as a domain name.

Important

The DNS protocol does not allow you to create a CNAME record for the top node of a DNS namespace, also known as the zone apex. For example, if you register the DNS name example.com, the zone apex is example.com. You cannot create a CNAME record for example.com, but you can create CNAME records for www.example.com, newproduct.example.com, and so on.

In addition, if you create a CNAME record for a subdomain, you cannot create any other resource record sets for that subdomain. For example, if you create a CNAME for www.example.com, you cannot create any other resource record sets for which the value of the Name field is www.example.com.

Route 53 also supports alias resource record sets, which allow you to route queries to an Elastic Load Balancing load balancer. Aliases are similar in some ways to the CNAME resource record type, however, you can create an alias for the zone apex. For more information, see Creating Alias Resource Record Sets for Elastic Load Balancing (p. 66).

Example

```
<Value>hostname.example.com</Value>
```

MX Format

An MX record `Value` element consists of two fields: a decimal number that represents the priority of the MX record, and the domain name of a mail host, for example, `mail.example.com`.

Example

```
<Value>10 mail.example.com</Value>
```

NS Format

An NS record `Value` element is the same format as a domain name.

Example

```
<Value>ns-1.example.com</Value>
```

PTR Format

A PTR record `Value` element is the same format as a domain name.

Example

```
<Value>hostname.example.com</Value>
```

SOA Format

An SOA record `Value` element consists of seven fields. The first two fields are formatted as domain names and represent the primary authority for the zone and the contact details for the zone administrator, respectively. The remaining five fields are decimal numbers representing the zone serial number, refresh time, retry time, expire time, and minimum time to live (TTL), respectively.

Example

```
<Value>ns-2048.awsdns-64.net hostmaster.awsdns.com 1 1 1 1 60</Value>
```

SPF Format

An SPF record `Value` element is the same format as a TXT format record. For information about SPF record format, refer to the applicable documentation. For information about TXT format, see TXT Format (p. 6).

Example

```
<Value>"v=spf1 ip4:192.168.0.1/16 -all"</Value>
```

SRV Format

An SRV record `Value` element consists of four space-separated values. The first three values are decimal numbers representing priority, weight, and port. The fourth value is a domain name. For information about SRV record format, refer to the applicable documentation.

Example

```
<Value>10 5 80 hostname.example.com</Value>
```

TXT Format

A TXT record `Value` element is a space separated list of double-quoted strings. A single string cannot exceed 255 characters. In addition to the characters that are permitted unescaped in domain names, space is allowed in TXT strings. All other octet values must be quoted in octal form. Unlike domain names, case is preserved in character strings, meaning that `Ab` is not the same as `aB`. You can include a literal quote in a string by escaping it.

Example

```
<Value>"this is a string" "a string with a \" quote in it" "a string with a
\100 strange character in it"</Value>
```

NS and SOA Records that Route 53 Creates for a Hosted Zone

For each hosted zone you create, Route 53 automatically creates four name server (NS) records and one SOA record. Don't change these records.

Topics

- Name Server (NS) Records (p. 6)
- The Start of Authority (SOA) Record (p. 7)

Name Server (NS) Records

The name server records that Route 53 automatically creates at the apex of your hosted zone are the authoritative name servers for your zone. You should not modify these records or add more name servers. The names of Route 53 name servers look like this:

- ns-2048.awsdns-64.com
- ns-2049.awsdns-65.net
- ns-2050.awsdns-66.org
- ns-2051.awsdns-67.co.uk

After you create a hosted zone, update your registrar's or your DNS service's name server records, as applicable, to refer to the Route 53 name servers:

- If you created a domain that uses Route 53 as the DNS service, see Updating the Registrar's Name Servers (p. 20).
- If you migrated an existing domain to Route 53, see Updating Your DNS Service's Name Server Records (p. 25).
- If you created a subdomain that uses Route 53 without migrating the parent domain, see Delegating Responsibility for the Subdomain to Route 53 (p. 30).
- If you migrated a subdomain to Route 53 without migrating the parent domain, see Delegating Responsibility for the Subdomain to Route 53 (p. 35).

The Start of Authority (SOA) Record

The SOA record identifies the base DNS information about the domain, for example:

```
ns-2048.awsdns-64.net. hostmaster.example.com. 1 7200 900 1209600 86400
```

The elements of the SOA record include:

- The host that created the SOA record, for example, `ns-2048.awsdns-64.net`.
- The email address of the administrator in a format with the `@` symbol replaced by a period, for example, `hostmaster.example.com`. The default value is an amazon.com email address that is not monitored.
- A revision number to increment when you change the zone file and distribute changes to secondary DNS servers, for example `1`.
- A refresh time in seconds that secondary DNS servers wait before querying the primary DNS server's SOA record to check for changes, for example `7200`.
- The retry interval in seconds that a secondary server waits before retrying a failed zone transfer, for example `900` (15 minutes). Normally, the retry time is less than the refresh time.
- The expire time in seconds that a secondary server will keep trying to complete a zone transfer, for example `1209600` (two weeks). If this time expires prior to a successful zone transfer, the secondary server will expire its zone file. This means that the secondary server will stop answering queries because it considers its data too old to be reliable.
- The minimum time to live (TTL). This value helps define the length of time that an NXDOMAIN result, which indicates that a domain does not exist, should be cached by a DNS resolver. Caching this negative result is referred to as negative caching. The duration of negative caching is the lesser of the SOA record's TTL or the value of the minimum TTL field. The default minimum TTL on Route 53 SOA records is 900 seconds. To change the TTL for resource record sets, including SOA resource record sets, use the `ChangeResourceRecordSets` API. For more information, see ChangeResourceRecordSets in the *Amazon Route 53 API Reference*.

Limits on Route 53 API Requests and Entity Counts

Route 53 API requests are subject to the following limitations.

Request	Limitation
`ChangeResourceRecordSets` requests	• A request cannot contain more than 100 `Change` elements. • A request cannot contain more than 1000 `ResourceRecord` elements. • The sum of the number of characters (including spaces) in all `Value` elements in a request cannot exceed 32,000 characters.
Route 53 API requests	5 requests per second per AWS account.

Route 53 entities are subject to the following limitations.

Entity	Limitation
Hosted zones	100 per AWS account. You can request a higher limit at http://aws.amazon.com/route53-request.
Resource record sets	10,000 per hosted zone. You can request a higher limit at http://aws.amazon.com/route53-request.
Resource records	100 per resource record set.

DNS Constraints and Behaviors

DNS messaging is subject to factors that affect how you create and use hosted zones and resource record sets. This section explains these factors.

Maximum Response Size

To comply with DNS standards, responses sent over UDP are limited to 512 bytes in size. Responses exceeding 512 bytes are truncated and the resolver must re-issue the request over TCP. If the resolver supports EDNS0 (as defined in RFC 2671), and advertises the EDNS0 option to Route 53, Route 53 permits responses up to 4096 bytes over UDP, without truncation.

Authoritative Section Processing

For successful queries, Route 53 appends name server (NS) resource record sets for the relevant hosted zone to the Authority section of the DNS response. For names that are not found (NXDOMAIN responses), Route 53 appends the start of authority (SOA) resource record set (as defined in RFC 1035) for the relevant hosted zone to the Authority section of the DNS response.

Additional Section Processing

Route 53 appends resource record sets to the Additional section. If the records are known and appropriate, the service appends A or AAAA resource record sets for any target of an MX, CNAME, NS, or SRV record cited in the Answer section. For more information about these DNS record types, see Supported DNS Resource Record Types (p. 4).

Route 53 Pricing

As with other AWS products, there are no contracts or minimum commitments for using Route 53—you pay only for the hosted zones you configure and the number of queries that Route 53 answers. For more information, see Route 53 Pricing.

AWS Identity and Access Management

Route 53 integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

- Create users and groups under your organization's AWS Account
- Easily share your AWS Account resources between the users in the account
- Assign unique security credentials to each user
- Granularly control users access to services and resources
- Get a single AWS bill for all users in the AWS Account

For example, you can use IAM with Route 53 to control which users in your AWS Account can create a new hosted zone or change resource record sets.

For information about using Route 53 with IAM, see Controlling User Access with IAM (p. 79).

For general information about IAM, go to:

- Identity and Access Management (IAM)
- *AWS Identity and Access Management Getting Started Guide*
- Using AWS Identity and Access Management

Getting Started with Route 53

Whether you are creating a new DNS domain or subdomain, or migrating an existing domain or subdomain, getting started with Route 53 is easy: create an AWS account if you don't already have one, create a hosted zone and some resource record sets in the Route 53 console, and then update your domain registrar to use the Route 53 name servers. The procedures in this chapter take you through the entire process for creating a new domain that uses Route 53. We also explain your other options for accessing Route 53.

Topics

- Getting Started: Creating a Domain that Uses Route 53 (p. 10)
- How to Use the Route 53 Console, API, AWS SDKs, and Command-Line Tool (p. 14)

Getting Started: Creating a Domain that Uses Route 53

You can use the procedures in the following example to create a domain that uses Route 53 as the DNS service.

Step 1: Sign Up for an AWS Account

To use Route 53, sign up for an AWS account if you don't already have one.

Note

When you sign up for an account, AWS automatically signs up the account for all services. You are charged only for the services that you use.

To sign up for an AWS account

1. Go to http://aws.amazon.com, and then click **Sign Up Now**.
2. Follow the on-screen instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.

Step 2: Register Your Domain Name

Register your domain name. For a list of registrar web sites that you can use to register your domain name, see ICANN.org. After your registrar notifies you that your domain name is successfully registered, you can create a Route 53 hosted zone for the domain.

Step 3: Create a Hosted Zone

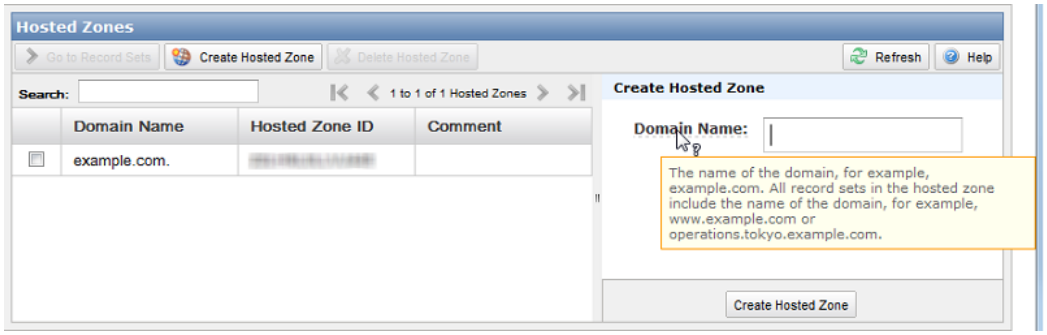
To create a domain that uses Route 53 as the DNS service, start by creating a Route 53 hosted zone. Route 53 stores information about your domain in the hosted zone.

Note

When you create a hosted zone, Route 53 automatically creates four name server (NS) records and a start of authority (SOA) record for the zone. The NS records identify the name servers that you give to your registrar or your DNS service so that queries are routed to Route 53 name servers. For more information about NS and SOA records, see NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6).

To create a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, above the left pane, click **Create Hosted Zone**.
3. In the right pane, enter a domain name and, optionally, a comment. For more information about a field, see the tool tip for the field.



4. Below the right pane, click **Create Hosted Zone**.

Step 4: Create Resource Record Sets in Your Route 53 Hosted Zone

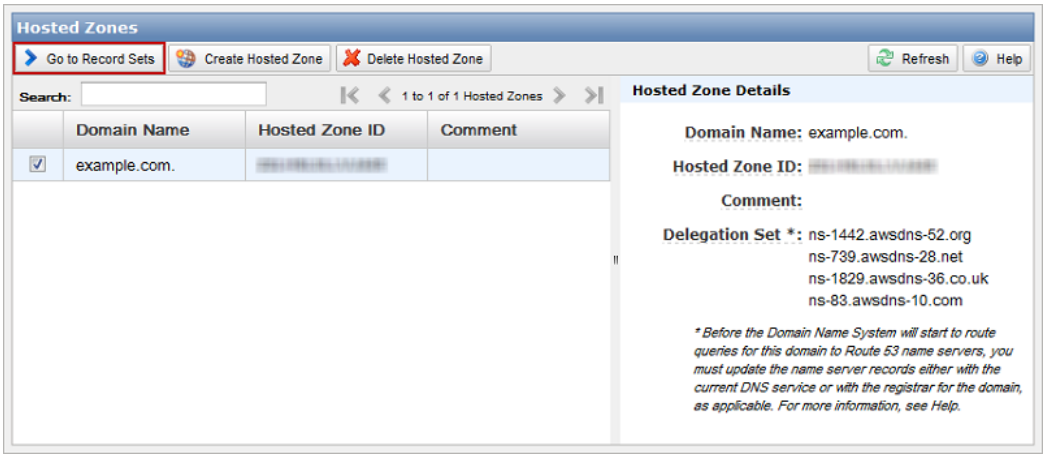
Now that you have a hosted zone, you can create resource record sets. For example, if you want a user who enters **example.com** in a web browser to be routed to a host that has the IP address 192.0.2.234, you would create a resource record set for example.com with a **Type** of A and a **Value** of 192.0.2.234.

To create resource record sets using the Route 53 console

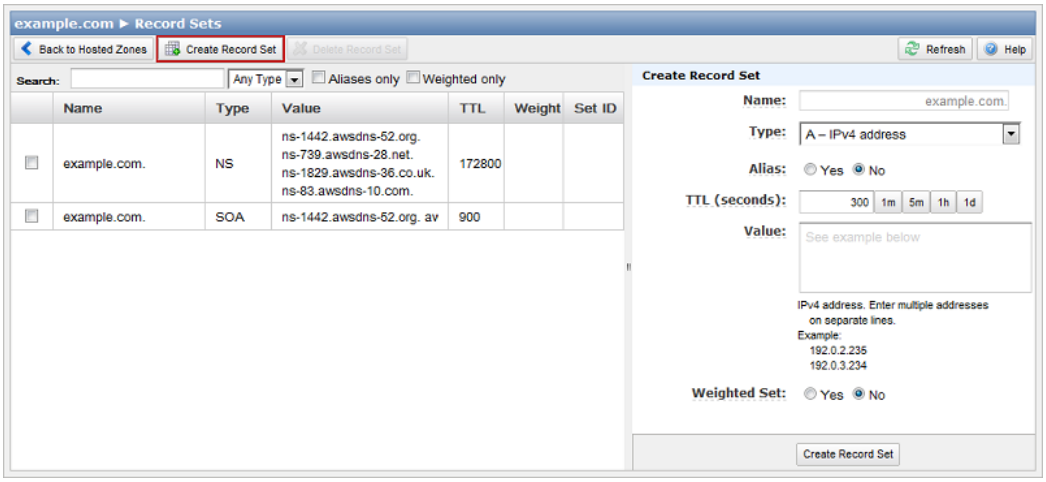
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.

2. On the Hosted Zones page, do one of the following:

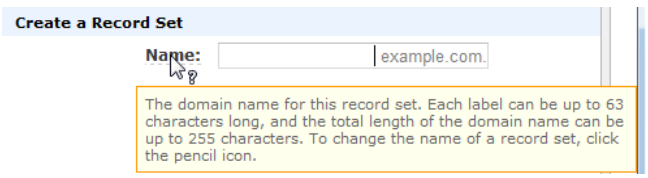
   - Click the row for the hosted zone in which you want to create record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. On the Record Sets page, above the left pane, click **Create Record Set**.



4. In the right pane, enter the applicable values. For information about a field, see the tool tip for the field.



5. Below the right pane, click **Create Record Set**.

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

Step 5: Update the Registrar's Name Server Records

Important

Use the following procedure only if you are creating a domain. If you're migrating an existing domain, or creating or migrating a subdomain, see the corresponding procedure in the applicable section:

- Migrating an existing domain to Route 53: see Updating Your DNS Service's Name Server Records (p. 25).
- Creating a subdomain that uses Route 53 without migrating the parent domain: see Delegating Responsibility for the Subdomain to Route 53 (p. 30).
- Migrating a subdomain to Route 53 without migrating the parent domain: see Delegating Responsibility for the Subdomain to Route 53 (p. 35).

Update the name server (NS) records with your registrar to refer to the Route 53 name servers. Perform the following procedure.

1. If the registrar has a method to reset the TTL settings for their name servers, we recommend that you reset the settings to 900 seconds. This limits the time during which client requests will try to resolve domain names using obsolete name servers. You will need to wait for the duration of the previous TTL for resolvers and clients to stop caching the DNS records with their previous values. A common default setting is 172800 seconds (two days). After the TTL settings expire, you can safely delete the records that are stored at the previous provider and make changes only to Route 53.

   Note

   The TTL setting is only an issue if you have queried the domain, so the domain name and IP address are cached with DNS resolvers.

2. In the Route 53 console, get the name servers in the delegation set for your Route 53 hosted zone:

   a. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
   b. On the Hosted Zones page, click the name of the hosted zone.
   c. In the right pane, make note of the four servers listed for **Delegation Set**.

3. Using the method provided by the registrar for the domain, replace the name servers in the registrar's NS records with the four Route 53 name servers that you looked up in the previous step.

   Depending on the TTL settings for the name servers for the parent domain, the propagation of your changes to DNS resolvers can take 48 hours or more. During this period, DNS resolvers may still answer requests with the name servers for the registrar. In addition, client computers may continue to have the previous name servers for the domain in their cache.

For more information about using Route 53, see Additional Route 53 Resources (p. 100)

How to Use the Route 53 Console, API, AWS SDKs, and Command-Line Tool

You can access Route 53 using the Route 53 console, the Route 53 API, AWS SDKs, or the `dnscurl.pl` utility.

Topics

- The Route 53 Console (p. 14)
- The Route 53 API (p. 15)
- AWS SDKs that Support Route 53 (p. 15)
- The dnscurl.pl Command-Line Tool (p. 15)

The Route 53 Console

The Route 53 console lets you create, delete, and list Route 53 hosted zones and resource record sets.

Note

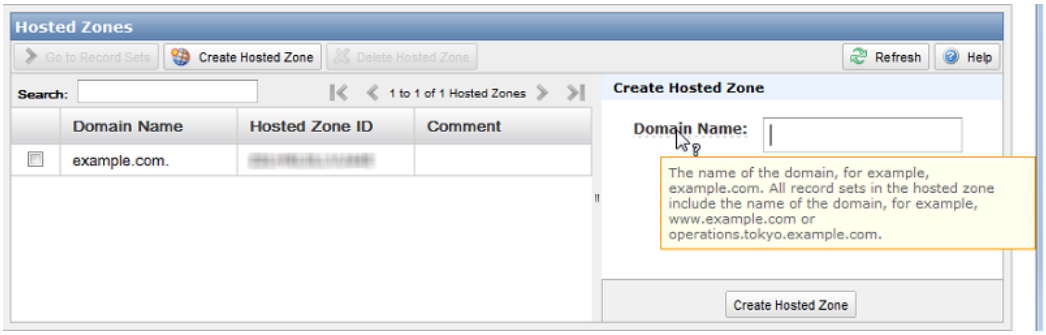
Some ad-blocking plugins for web browsers interfere with Route 53 console operations, which can cause the console to behave unpredictably. If you installed an ad-blocking plugin for your browser, we recommend that you add the URL for the Route 53 console, https://console.aws.amazon.com/route53/home, to the whitelist for the plugin.

To access the Route 53 console

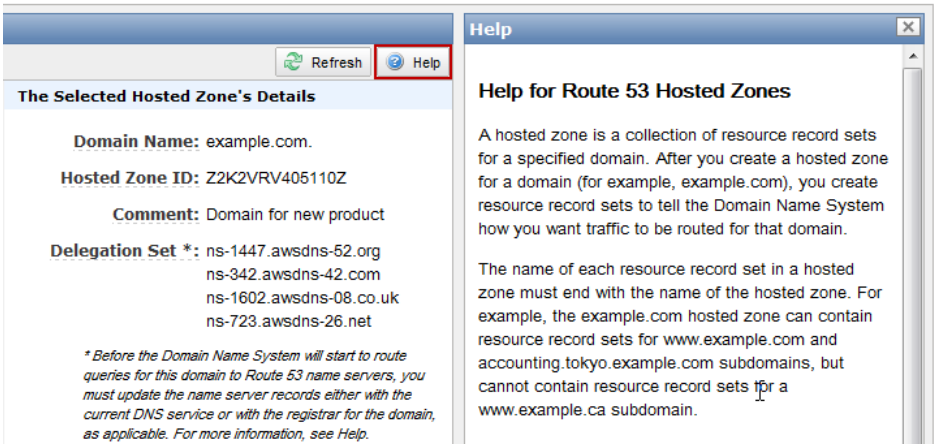
- Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.

To display help for the Route 53 console

- To display help for a field, move the cursor over the field name.



- To display help for hosted zones or for record sets, click **Help**.



The Route 53 API

The Route 53 API is a REST API that you can use to create, delete, and list Route 53 hosted zones and resource record sets. (When using the API, you change a resource record set by deleting the existing one and creating a new one.) For information about the Route 53 API, see the *Amazon Route 53 API Reference*. For information about how to use the API, including how to authenticate REST requests, see Making API Requests (p. 83).

AWS SDKs that Support Route 53

AWS SDK for Java version 1.2.13 and later includes a client for Route 53. For more information, see AWS SDK for Java.

AWS SDK for .NET version 1.4.1 and later includes a client for Route 53. For more information, see AWS SDK for .NET.

The dnscurl.pl Command-Line Tool

The `dnscurl.pl` command-line tool is a Perl script that lets you call Route 53 API actions from a Linux command line. When you run `dnscurl.pl` and specify the applicable arguments for a given action, `dnscurl.pl` calculates the Route 53 authentication signature, then calls Curl, a popular tool for interacting with HTTP services. Curl passes the arguments to Route 53, and Route 53 processes the request and returns the results.

You can get `dnscurl.pl` on the Route 53 Authentication Tool for Curl page. Save `dnscurl.pl` on the executable path, and make it executable by running the following command at a Linux command prompt:

```
chmod 755 dnscurl.pl
```

Note

We have tested `dnscurl.pl` on Linux and Mac OS.

Some customers have successfully configured Windows clients so that they can run `dnscurl.pl`. For more information, see the Route 53 forum.

Before you run `dnscurl.pl`, install the following software:

- Curl version 7.15.5 or later.
- Perl, which you can download from Perl.org. `dnscurl.pl` was tested with Perl version 5.8.8.
- The following Perl modules, which you can download from CPAN:
  - Digest::HMAC_SHA1
  - FindBin
  - MIME::Base64
  - Getopt::Long
  - File::Temp
  - File::Basename
  - Fcntl
  - IO::Handle

For an example of how to use `dnscurl.pl`, see Example: Using dnscurl.pl (p. 91).

Creating a Domain that Uses Route 53 as the DNS Service

You can use Route 53 as the DNS service for any registered domain name. This section explains how to create a domain that uses Route 53 as the DNS service.

The following table summarizes the process.

Process for Creating a Domain that Uses Route 53 as the DNS Service

1	Register your domain name. See Registering Your Domain Name (p. 17).
2	Create a Route 53 hosted zone for your domain. See Creating a Hosted Zone (p. 18).
3	Add resource record sets to your Route 53 hosted zone. See Creating Resource Record Sets (p. 18).
4	*API only:* Confirm that your changes have propagated to all Route 53 DNS servers. See Checking the Status of Your Changes (API Only) (p. 20). **Note** Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.
5	Update your registrar's name server records. See Updating the Registrar's Name Servers (p. 20).

Registering Your Domain Name

Route 53 requires you to use a registered domain name. Before creating your Route 53 DNS service, make sure that you have registered the domain name that you want to use. For a list of registrar web sites you can use to register your domain name, go to ICANN.org.

After your registrar notifies you that your domain name is successfully registered, you can create a Route 53 hosted zone for the domain.

Caution

You can create hosted zones only for domains that are registered to you. Attempting to create a hosted zone for a domain that is not registered to you constitutes abuse and puts your AWS account at risk of termination.

Creating a Hosted Zone

To create a domain that uses Route 53 as the DNS service, start by creating a Route 53 hosted zone. Route 53 stores information about your domain in the hosted zone.

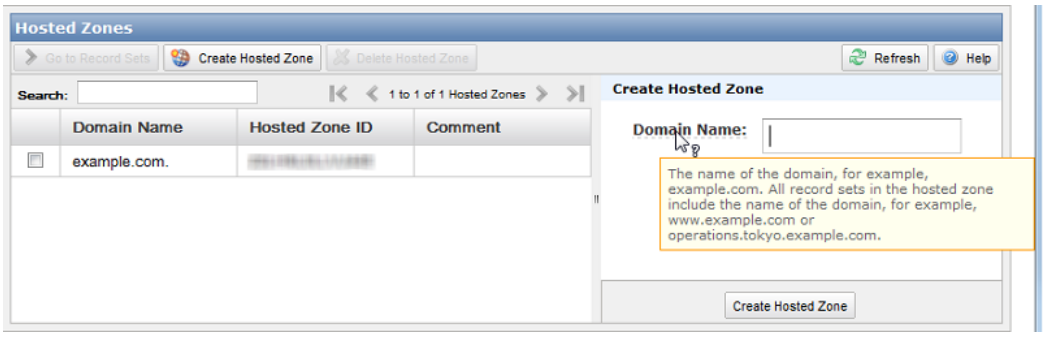
Note

When you create a hosted zone, Route 53 automatically creates four name server (NS) records and a start of authority (SOA) record for the zone. The NS records identify the name servers that you give to your registrar or your DNS service so that queries are routed to Route 53 name servers. For more information about NS and SOA records, see NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6).

To create a hosted zone using the Route 53 console, perform the following procedure. To create a hosted zone using the Route 53 API, use the `CreateHostedZone` action. For more information, see POST CreateHostedZone in the *Amazon Route 53 API Reference*.

To create a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, above the left pane, click **Create Hosted Zone**.
3. In the right pane, enter a domain name and, optionally, a comment. For more information about a field, see the tool tip for the field.



4. Below the right pane, click **Create Hosted Zone**.

Creating Resource Record Sets

You can create resource record sets using either the Route 53 console or the Route 53 API. The resource record sets that you create in Route 53 will become the resource record sets that DNS uses after you

update your registrar's name server records, as explained in Updating the Registrar's Name Servers (p. 20), later in the process.

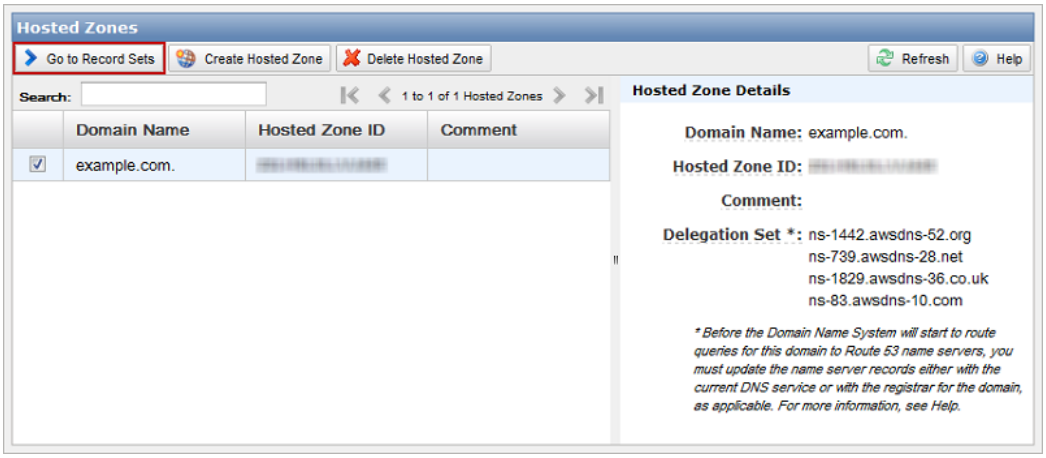
Caution

Do not create additional name serve (NS) or start of authority (SOA) records in the Route 53 hosted zone, or delete the existing records.

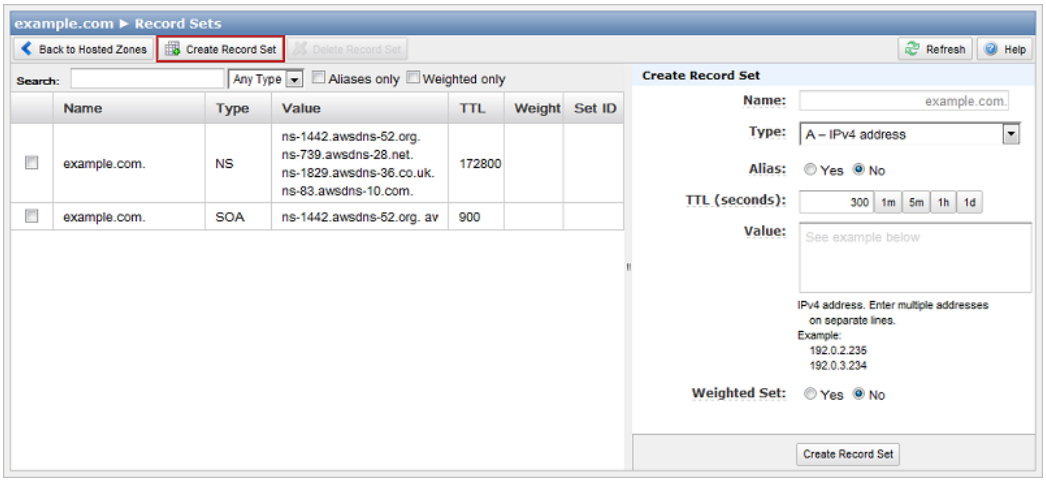
To create resource record sets using the Route 53 console, perform the following procedure. To create resource record sets using the Route 53 API, use the `ChangeResourceRecordSets` action. For more information, see POST ChangeResourceRecordSets in the *Amazon Route 53 API Reference*.

To create resource record sets using the Route 53 console

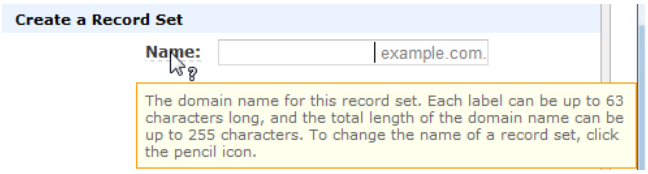
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:
   - Click the row for the hosted zone in which you want to create record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. On the Record Sets page, above the left pane, click **Create Record Set**.



4. In the right pane, enter the applicable values. For information about a field, see the tool tip for the field.



5. Below the right pane, click **Create Record Set**.

Checking the Status of Your Changes (API Only)

Creating a new hosted zone and changing resource record sets take time to propagate to the Route 53 DNS servers. If you are using the Route 53 API, you can use the `GetChange` action to determine whether your changes have propagated. For more information, see GET GetChange in the *Amazon Route 53 API Reference*.

Note

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

Updating the Registrar's Name Servers

After your changes to Route 53 resource record sets have propagated to the Route 53 DNS servers and have a status of `INSYNC`, update your registrar's name server (NS) records to refer to the Route 53 name servers. Perform the following procedure.

1. If the registrar has a method to change the TTL settings for their name servers, we recommend that you reset the settings to 900 seconds. This limits the time during which client requests will try to resolve domain names using obsolete name servers. You will need to wait for the duration of the previous TTL for resolvers and clients to stop caching the DNS records with their previous values.

A common default setting is 172800 seconds (two days). After the TTL settings expire, you can safely delete the records that are stored at the previous provider and make changes only to Route 53.

Note

The TTL setting is only an issue if you have queried the domain, so the domain name and IP address are cached with DNS resolvers.

2. In the Route 53 console, get the name servers in the delegation set for your Route 53 hosted zone:

   a. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
   b. On the Hosted Zones page, click the name of the hosted zone.
   c. In the right pane, make note of the four servers listed for **Delegation Set**.

   Alternatively, you can use the `GetHostedZone` action. For more information, see GetHostedZone in the *Amazon Route 53 API Reference*.

3. Using the method provided by the registrar for the domain, replace the name servers in the registrar's NS records with the four Route 53 name servers that were returned when you submitted the `GetHostedZone` request in the previous step.

   Depending on the TTL settings for the name servers for the parent domain, the propagation of your changes to DNS resolvers can take 48 hours or more. During this period, DNS resolvers may still answer requests with the name servers for the registrar. In addition, client computers may continue to have the previous name servers for the domain in their cache.

For more information about working with your hosted zone, see the following related topics.

Related Topics

- Getting the Name Servers for a Hosted Zone (p. 39)
- Listing the Hosted Zones for an AWS Account (p. 41)
- Deleting a Hosted Zone (p. 45)
- Listing Resource Record Sets (p. 56)

Migrating an Existing Domain to Route 53

You can use Route 53 as the DNS service for any registered domain name. The procedures in this section explain how to migrate an existing domain to use Route 53 as the DNS service.

Caution

You can create hosted zones only for domains that are registered to you. Attempting to create a hosted zone for a domain that is not registered to you constitutes abuse and puts your AWS account at risk of termination.

The following table summarizes the process.

Process for Migrating an Existing Domain to Route 53

1	Create a Route 53 hosted zone for your domain. See Creating a Hosted Zone (p. 23).
2	Get resource record sets from the current DNS service for the domain that you are migrating. See Getting Resource Record Sets from Your DNS Service Provider (p. 23).
3	Add resource record sets to your Route 53 hosted zone. See Creating Resource Record Sets (p. 24).
4	*API only:* Confirm that your changes have propagated to all Route 53 DNS servers. See Checking the Status of Your Changes (API Only) (p. 25). **Note** Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.
5	Update the name server records with your current DNS service to use the Route 53 name servers assigned to your hosted zone. See Updating Your DNS Service's Name Server Records (p. 25).

Creating a Hosted Zone

To migrate a domain from your existing DNS service, start by creating a Route 53 hosted zone. Route 53 stores information about your domain in the hosted zone.

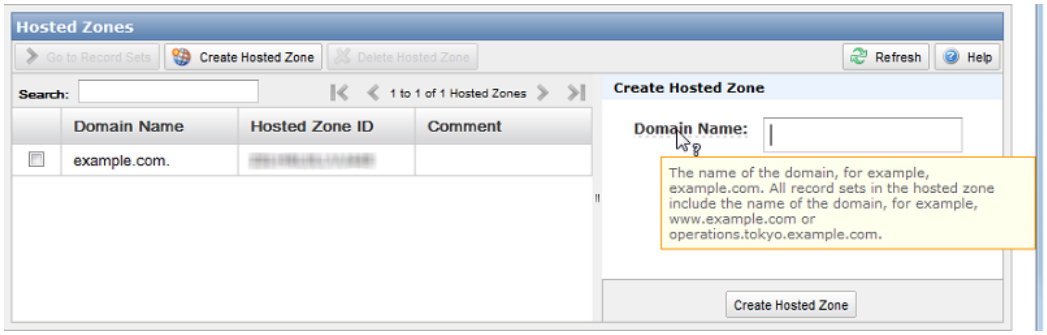
Note

When you create a hosted zone, Route 53 automatically creates four name server (NS) records and a start of authority (SOA) record for the zone. The NS records identify the name servers that you give to your registrar or your DNS service so that queries are routed to Route 53 name servers. For more information about NS and SOA records, see NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6).

To create a hosted zone using the Route 53 console, perform the following procedure. To create a hosted zone using the Route 53 API, use the `CreateHostedZone` action. For more information, see POST CreateHostedZone in the *Amazon Route 53 API Reference*.

To create a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, above the left pane, click **Create Hosted Zone**.
3. In the right pane, enter a domain name and, optionally, a comment. For more information about a field, see the tool tip for the field.



4. Below the right pane, click **Create Hosted Zone**.

Getting Resource Record Sets from Your DNS Service Provider

To simplify the process of migrating an existing domain to Route 53, get resource record sets from the DNS service provider that is currently servicing the domain. You can use this information as a basis for creating Route 53 resource record sets; you will continue to use most of the same records after you transfer your domain to Route 53.

How you get the existing resource record sets depends on which company you are currently using as your DNS service provider. Typically your DNS service provider will give you a zone file, but they might also give you the information in another format. For example, the *GoDaddy.com* service provides this information at the **My Account** link, under **List Records**. If you use another service provider, try asking customer support for your *records list* or *zone file* information.

Records that you are likely to migrate include:

- A (Address) records, which associate a domain name (example.com) with the IP address of the home page for the domain (192.0.2.3)
- Mail server (MX) records
- CNAME records, which reroute queries for one domain name (www.example.com) to another domain name (example.com)
- Other A records, CNAME records, or other supported DNS record types. For a list of supported record types, see Supported DNS Resource Record Types (p. 4).

Creating Resource Record Sets

Using the resource record sets that you got from your current DNS service provider as a starting point, create corresponding resource record sets in the Route 53 hosted zone. The resource record sets that you create in Route 53 will become the resource record sets that DNS uses after you update your current DNS service's name server records, as explained in Updating Your DNS Service's Name Server Records (p. 25), later in the process.

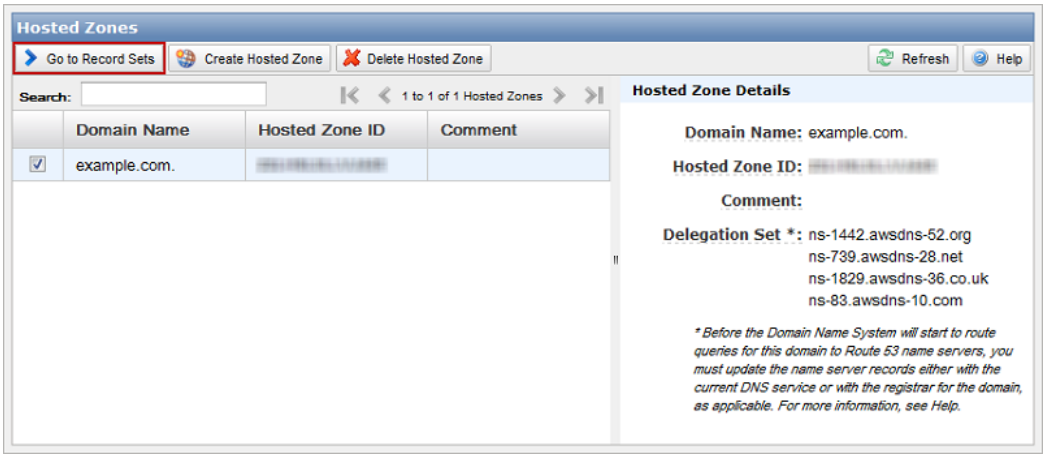
Caution

Do not create additional name serve (NS) or start of authority (SOA) records in the Route 53 hosted zone, or delete the existing records.

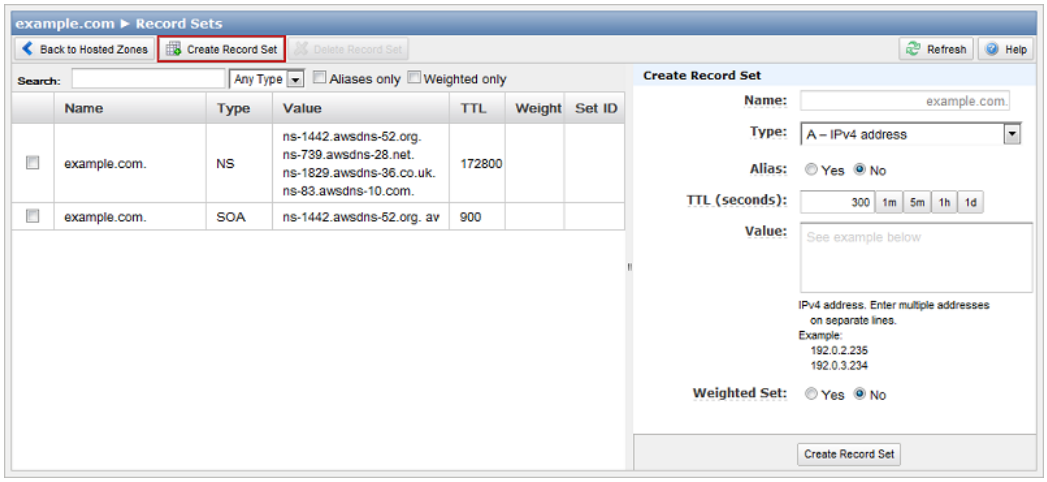
To create resource record sets using the Route 53 console, perform the following procedure. To create resource record sets using the Route 53 API, use the `ChangeResourceRecordSets` action. For more information, see POST ChangeResourceRecordSets in the *Amazon Route 53 API Reference*.

To create resource record sets using the Route 53 console

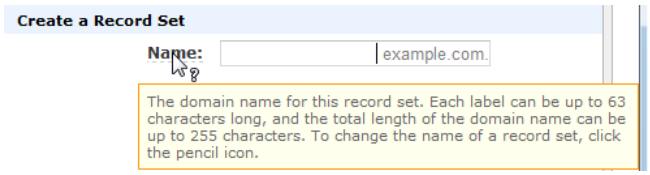
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:
   - Click the row for the hosted zone in which you want to create record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. On the Record Sets page, above the left pane, click **Create Record Set**.



4. In the right pane, enter the applicable values. For information about a field, see the tool tip for the field.



5. Below the right pane, click **Create Record Set**.

Checking the Status of Your Changes (API Only)

Creating a new hosted zone and changing resource record sets take time to propagate to the Route 53 DNS servers. If you are using the Route 53 API, you can use the `GetChange` action to determine whether your changes have propagated. For more information, see GET GetChange in the *Amazon Route 53 API Reference*.

Note

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

Updating Your DNS Service's Name Server Records

After your changes to Route 53 resource record sets have propagated to the Route 53 DNS servers and have a status of `INSYNC`, update your previous DNS service's name server (NS) records to refer to the Route 53 name servers. Perform the following procedure.

To update your DNS service's name server records to refer to Route 53 name servers

1. *Optional:* Using the method provided by your previous DNS service, back up the zone file for the domain.
2. If your previous DNS service has a method to change the TTL settings for their name servers, we recommend that you reset the settings to 900 seconds. This limits the time during which client requests will try to resolve domain names using obsolete name servers. You will need to wait for the duration of the previous TTL for resolvers and clients to stop caching the DNS records with their previous values. A common default setting is 172800 seconds (two days). After the TTL settings expire, you can safely delete the records that are stored at the previous provider and make changes only to Route 53.
3. In the Route 53 console, get the name servers in the delegation set for your Route 53 hosted zone:

   a. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
   b. On the Hosted Zones page, click the name of the hosted zone.
   c. In the right pane, make note of the four servers listed for **Delegation Set**.

   Alternatively, you can use the `GetHostedZone` action. For more information, see GetHostedZone in the *Amazon Route 53 API Reference*.
4. Using the method provided by your DNS service, replace the name servers in your previous DNS service's NS records with the four Route 53 name servers that were returned when you submitted the `GetHostedZone` request in the previous step.

   Depending on the TTL settings in your previous DNS service's NS records, the propagation of your changes to DNS resolvers can take 48 hours or more. During this period, DNS resolvers may still answer requests with the name servers for your previous DNS service. In addition, client computers may continue to have the previous name servers for the domain in their cache.

To learn more about working with your hosted zone, see the following related topics.

Related Topics

- Getting the Name Servers for a Hosted Zone (p. 39)
- Listing the Hosted Zones for an AWS Account (p. 41)
- Deleting a Hosted Zone (p. 45)
- Listing Resource Record Sets (p. 56)

Creating a Subdomain that Uses Route 53 without Migrating the Parent Domain

This section explains how to create a subdomain that uses Route 53 as the DNS service without migrating the parent domain from another DNS service.

Process for Creating a Subdomain that Uses Route 53 without Migrating the Parent Domain from another DNS Service

1	Create a Route 53 hosted zone for the subdomain. See Creating a Hosted Zone for the New Subdomain (p. 28).
2	Add resource record sets for the new subdomain to your Route 53 hosted zone. See Creating Resource Record Sets (p. 28).
3	*API only:* Confirm that your changes have propagated to all Route 53 DNS servers. See Checking the Status of Your Changes (API Only) (p. 30). **Note** Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.
4	Delegate responsibility for the subdomain to Route 53. See Delegating Responsibility for the Subdomain to Route 53 (p. 30).

Creating a Hosted Zone for the New Subdomain

When you want to use Route 53 as the DNS service for a new subdomain without migrating the parent domain, you start by creating a hosted zone for the subdomain. Route 53 stores information about your subdomain in the hosted zone.

Note

When you create a hosted zone, Route 53 automatically creates four name server (NS) records and a start of authority (SOA) record for the zone. The NS records identify the name servers that you give to your registrar or your DNS service so that queries are routed to Route 53 name servers. For more information about NS and SOA records, see NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6).

To create a hosted zone using the Route 53 console, perform the following procedure. To create a hosted zone using the Route 53 API, use the `CreateHostedZone` action. For more information, see POST CreateHostedZone in the *Amazon Route 53 API Reference*.

To create a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, above the left pane, click **Create Hosted Zone**.
3. In the right pane, enter a domain name and, optionally, a comment. For more information about a field, see the tool tip for the field.

4. Below the right pane, click **Create Hosted Zone**.

Creating Resource Record Sets

You can create resource record sets using either the Route 53 console or the Route 53 API. The resource record sets that you create in Route 53 will become the resource record sets that DNS uses after you delegate responsibility for the subdomain, to Route 53, as explained in Delegating Responsibility for the Subdomain to Route 53 (p. 30), later in the process.

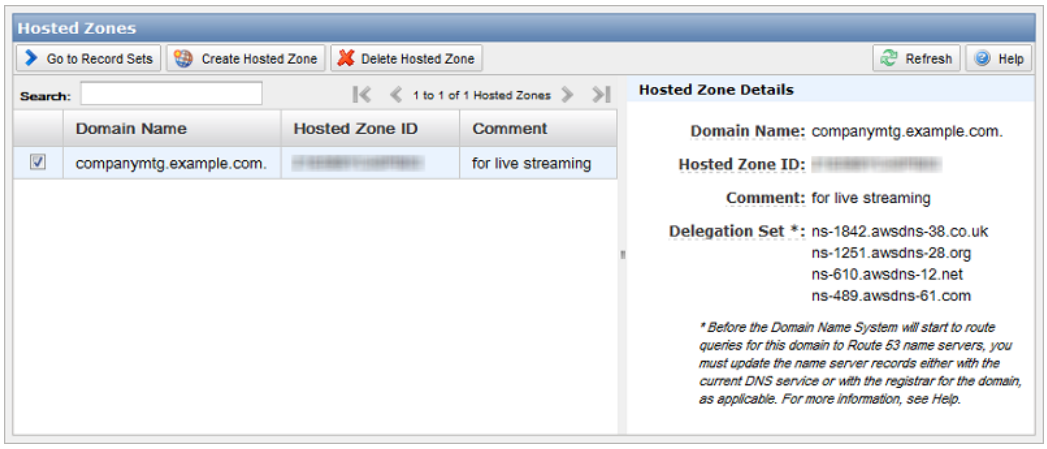
Caution

Do not create additional name serve (NS) or start of authority (SOA) records in the Route 53 hosted zone, or delete the existing records.

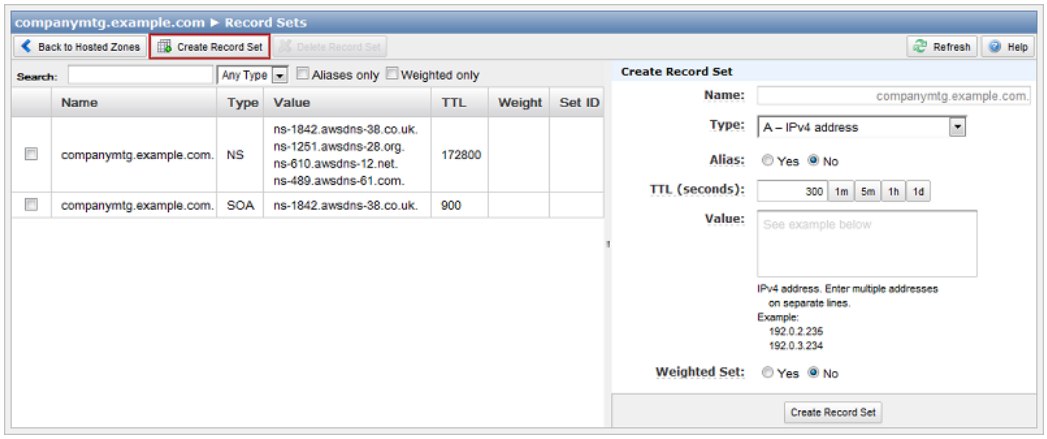
To create resource record sets using the Route 53 console, perform the following procedure. To create resource record sets using the Route 53 API, use the `ChangeResourceRecordSets` action. For more information, see POST ChangeResourceRecordSets in the *Amazon Route 53 API Reference*.

To create resource record sets using the Route 53 console

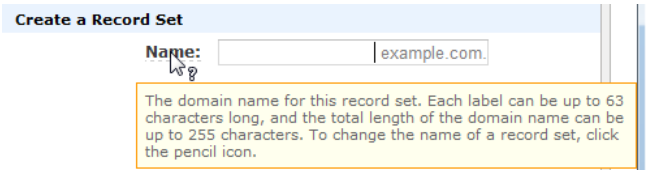
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:
   - Click the row for the hosted zone in which you want to create record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. On the Record Sets page, above the left pane, click **Create Record Set**.



4. In the right pane, enter the applicable values. For information about a field, see the tool tip for the field.



5. Below the right pane, click **Create Record Set**.

Checking the Status of Your Changes (API Only)

Creating a new hosted zone and changing resource record sets take time to propagate to the Route 53 DNS servers. If you are using the Route 53 API, you can use the `GetChange` action to determine whether your changes have propagated. For more information, see GET GetChange in the *Amazon Route 53 API Reference*.

Note

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

Delegating Responsibility for the Subdomain to Route 53

After your changes to Route 53 resource record sets have propagated and have a status of `INSYNC`, delegate responsibility for the subdomain to Route 53 by updating the zone file for the DNS service of the parent domain. Perform the following procedure.

1. Using the method provided by your DNS service, back up the zone file for the parent domain.
2. In the Route 53 console, get the name servers in the delegation set for your Route 53 hosted zone:
   a. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
   b. On the Hosted Zones page, click the name of the hosted zone.
   c. In the right pane, make note of the four servers listed for **Delegation Set**.

   Alternatively, you can use the `GetHostedZone` action. For more information, see GetHostedZone in the *Amazon Route 53 API Reference*.
3. Using the method provided by the DNS service of the parent domain, add NS records for the subdomain to the zone file for the parent domain. In these NS records, specify the four Route 53 name servers that are associated with the hosted zone that you created in Step 1.

Caution

Do not add a start of authority (SOA) record to the zone file for the parent domain. Because the subdomain will use Route 53, the DNS service for the parent domain is not the authority for the subdomain.

If your DNS service automatically added an SOA record for the subdomain, delete the record for the subdomain. However, do not delete the SOA record for the parent domain.

For more information about working with Route 53, see the following related topics.

Related Topics

- Getting the Name Servers for a Hosted Zone (p. 39)
- Listing the Hosted Zones for an AWS Account (p. 41)
- Deleting a Hosted Zone (p. 45)
- Listing Resource Record Sets (p. 56)

Migrating a Subdomain to Route 53 without Migrating the Parent Domain

This section explains how to migrate a subdomain to use Route 53 as the DNS service without migrating the parent domain from another DNS service.

Process for Migrating a Subdomain to Route 53 without Migrating the Parent Domain from another DNS Service

1	Create a Route 53 hosted zone for the subdomain. See Creating a Hosted Zone for the Subdomain that You Are Migrating (p. 33).
2	Get resource record sets from the current DNS service for the parent domain. See Getting Resource Record Sets for the Parent Domain (p. 33).
3	Add resource record sets for the subdomain to your Route 53 hosted zone. See Creating Resource Record Sets (p. 34).
4	*API only:* Confirm that your changes have propagated to all Route 53 DNS servers. See Checking the Status of Your Changes (API Only) (p. 35). **Note** Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.
5	Delegate responsibility for the subdomain to Route 53. See Delegating Responsibility for the Subdomain to Route 53 (p. 35).

Creating a Hosted Zone for the Subdomain that You Are Migrating

If you want to use Route 53 as the DNS service for a subdomain that you are migrating from another DNS service without migrating the parent domain, you start by creating a hosted zone for the subdomain that you are migrating. Route 53 stores information about your subdomain in the hosted zone.

Note

When you create a hosted zone, Route 53 automatically creates four name server (NS) records and a start of authority (SOA) record for the zone. The NS records identify the name servers that you give to your registrar or your DNS service so that queries are routed to Route 53 name servers. For more information about NS and SOA records, see NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6).

To create a hosted zone using the Route 53 console, perform the following procedure. To create a hosted zone using the Route 53 API, use the `CreateHostedZone` action. For more information, see POST CreateHostedZone in the *Amazon Route 53 API Reference*.

To create a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, above the left pane, click **Create Hosted Zone**.
3. In the right pane, enter a domain name and, optionally, a comment. For more information about a field, see the tool tip for the field.

4. Below the right pane, click **Create Hosted Zone**.

Getting Resource Record Sets for the Parent Domain

To simplify the process of migrating an existing subdomain to Route 53, get resource record sets from the DNS service provider that is currently servicing the parent domain. You can use this information as a basis for creating Route 53 resource record sets; you will continue to use most of the same records after you transfer your subdomain to Route 53.

How you get the existing resource record sets depends on which company you are currently using as your DNS service provider. Typically, your DNS service provider will give you a zone file, but they might also give you the information in another format. For example, the *GoDaddy.com* service provides this information at the **My Account** link, under **List Records**. If you use another service provider, try asking customer support for your *records list* or *zone file* information.

Creating Resource Record Sets

Using the resource record sets that you got from your current DNS service provider as a starting point, create corresponding resource record sets in the Route 53 hosted zone that you created for the subdomain. The resource record sets that you create in Route 53 will become the resource record sets that DNS uses after you delegate responsibility for the subdomain to Route 53, as explained in Delegating Responsibility for the Subdomain to Route 53 (p. 35), later in the process.

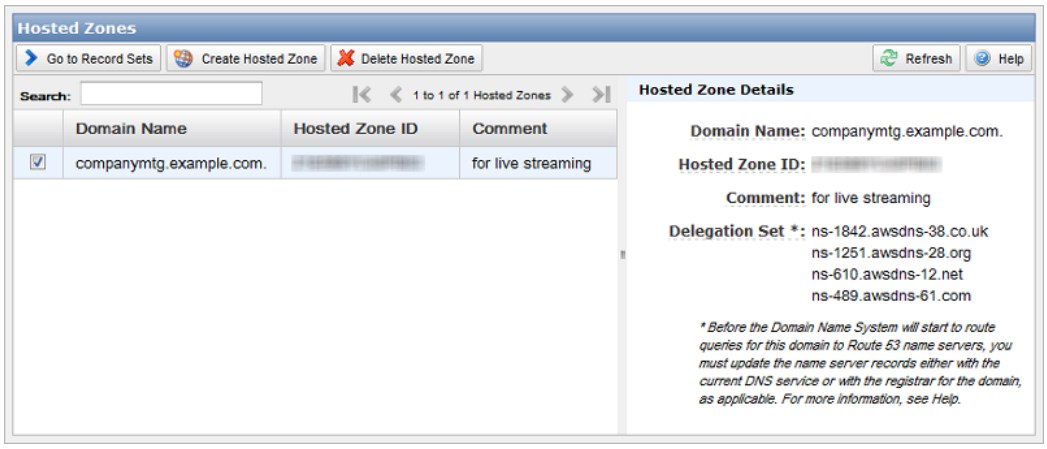
Caution

Do not create additional name serve (NS) or start of authority (SOA) records in the Route 53 hosted zone, or delete the existing records.

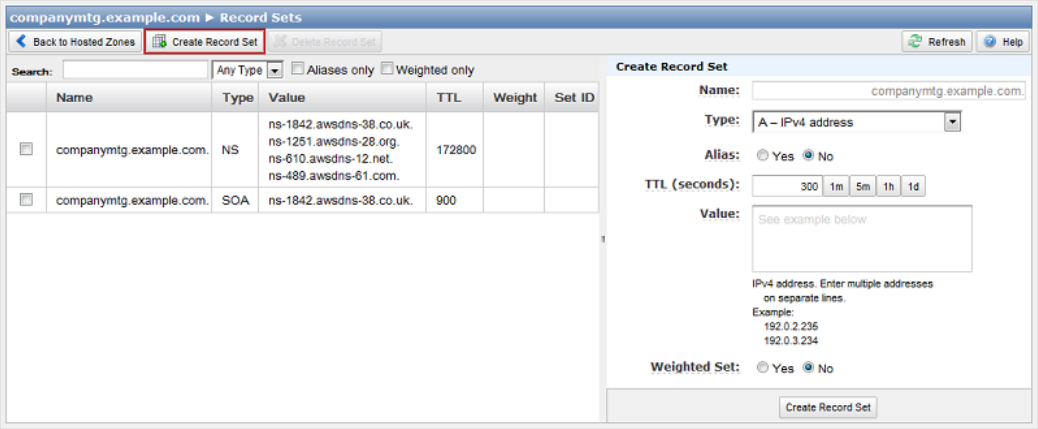
To create resource record sets using the Route 53 console, perform the following procedure. To create resource record sets using the Route 53 API, use the `ChangeResourceRecordSets` action. For more information, see POST ChangeResourceRecordSets in the *Amazon Route 53 API Reference*.

To create resource record sets using the Route 53 console

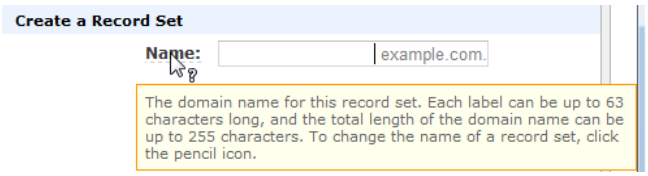
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:
   - Click the row for the hosted zone in which you want to create record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. On the Record Sets page, above the left pane, click **Create Record Set**.



4. In the right pane, enter the applicable values. For information about a field, see the tool tip for the field.



5. Below the right pane, click **Create Record Set**.

Checking the Status of Your Changes (API Only)

Creating a new hosted zone and changing resource record sets take time to propagate to the Route 53 DNS servers. If you are using the Route 53 API, you can use the `GetChange` action to determine whether your changes have propagated. For more information, see GET GetChange in the *Amazon Route 53 API Reference*.

Note

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

Delegating Responsibility for the Subdomain to Route 53

After your changes to Route 53 resource record sets have propagated and have a status of `INSYNC`, delegate responsibility for the subdomain to Route 53 by updating the zone file for the DNS service of the parent domain. Perform the following procedure.

To delegate responsibility for the subdomain to Route 53

1. Using the method provided by your DNS service, back up the zone file for the parent domain.

2. If the previous DNS service provider for the domain has a method to change the TTL settings for their name servers, we recommend that you change the settings to 900 seconds. This limits the time during which client requests will try to resolve domain names using obsolete name servers. If the current TTL is 172800 seconds (two days), which is a common default setting, you still need to wait two days for resolvers and clients to stop caching DNS records using the previous TTL. After the TTL settings expire, you can safely delete the records that are stored at the previous provider and make changes only to Route 53.
3. In the Route 53 console, get the name servers in the delegation set for your Route 53 hosted zone:

   a. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
   b. On the Hosted Zones page, click the name of the hosted zone.
   c. In the right pane, make note of the four servers listed for **Delegation Set**.

   Alternatively, you can use the `GetHostedZone` action. For more information, see GetHostedZone in the *Amazon Route 53 API Reference*.
4. Using the method provided by the DNS service of the parent domain, add NS records for the subdomain to the zone file for the parent domain. In these NS records, specify the four Route 53 name servers that are associated with the hosted zone that you created in Step 1.

   Caution

   Do not add a start of authority (SOA) record to the zone file for the parent domain. Because the subdomain will use Route 53, the DNS service for the parent domain is not the authority for the subdomain.

   If your DNS service automatically added an SOA record for the subdomain, delete the record for the subdomain. However, do not delete the SOA record for the parent domain.

   Depending on the TTL settings for the name servers for the parent domain, the propagation of your changes to DNS resolvers can take 48 hours or more. During this period, DNS resolvers may still answer requests with the name servers for the DNS service of the parent domain. In addition, client computers may continue to have the previous name servers for the subdomain in their cache.
5. After the registrar's TTL settings for the domain expire (see Step 2), delete the following resource record sets from the zone file for the parent domain:

   - The resource record sets that you added to Route 53 as described in Creating Resource Record Sets (p. 34).
   - Your DNS service's NS records. When you are finished deleting NS records, the only NS records in the zone file will be the ones that you created in Step 4.

To learn more about working with Route 53, see the following related topics.

Related Topics

- Getting the Name Servers for a Hosted Zone (p. 39)
- Listing the Hosted Zones for an AWS Account (p. 41)
- Deleting a Hosted Zone (p. 45)
- Listing Resource Record Sets (p. 56)

Working with Hosted Zones

You can use the Route 53 console or API to create, list, and delete hosted zones. See the applicable topic:

Topics

- Creating a Hosted Zone (p. 37)
- Getting the Name Servers for a Hosted Zone (p. 39)
- Listing the Hosted Zones for an AWS Account (p. 41)
- Deleting a Hosted Zone (p. 45)

Creating a Hosted Zone

A hosted zone is a collection of resource record sets for a specified domain. You create a hosted zone for a domain (for example, example.com), and then you create resource record sets to tell the Domain Name System how you want traffic to be routed for that domain.

Note

When you create a hosted zone, Route 53 automatically creates four name server (NS) records and a start of authority (SOA) record for the zone. The NS records identify the name servers that you give to your registrar or your DNS service so that queries are routed to Route 53 name servers. For more information about NS and SOA records, see NS and SOA Records that Route 53 Creates for a Hosted Zone (p. 6).

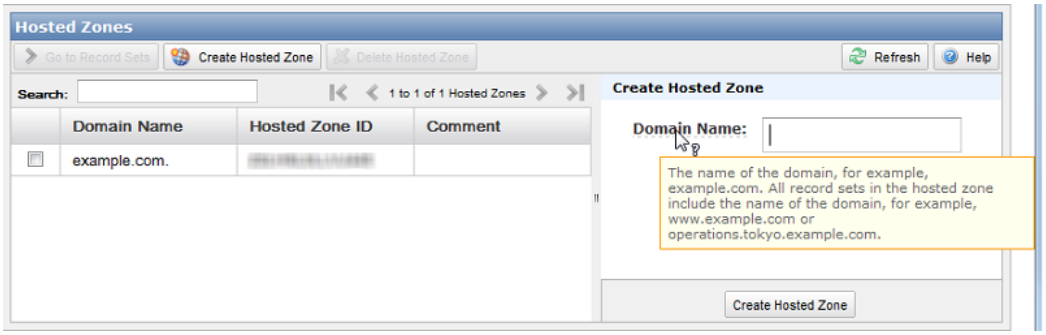
See the applicable topic:

- Creating a Hosted Zone Using the Route 53 Console (p. 38)
- Creating a Hosted Zone Using the Route 53 API (p. 38)

Creating a Hosted Zone Using the Route 53 Console

To create a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, above the left pane, click **Create Hosted Zone**.
3. In the right pane, enter a domain name and, optionally, a comment. For more information about a field, see the tool tip for the field.



4. Below the right pane, click **Create Hosted Zone**.

Creating a Hosted Zone Using the Route 53 API

To create a Route 53 hosted zone using the API, you use the `CreateHostedZone` action to send a POST request to the `2011-05-05/hostedzone` resource. The request body must include an XML document that contains a `CreateHostedZoneRequest` element. For more information about the `CreateHostedZone` action, see POST CreateHostedZone in the *Amazon Route 53 API Reference*.

Example Request

```
POST /2011-05-05/hostedzone HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<CreateHostedZoneRequest xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <Name>example.com.</Name>
   <CallerReference>myUniqueIdentifier</CallerReference>
   <HostedZoneConfig>
      <Comment>This is my first hosted zone.</Comment>
   </HostedZoneConfig>
</CreateHostedZoneRequest>
```

Example Response

```
HTTP/1.1 201 Created
<?xml version="1.0" encoding="UTF-8"?>
<CreateHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/
2011-05-05/">
   <HostedZone>
      <Id>/hostedzone/Z1PA6795UKMFR9</Id>
      <Name>example.com.</Name>
      <CallerReference>myUniqueIdentifier</CallerReference>
      <Config>
         <Comment>This is my first hosted zone.</Comment>
      </Config>
   </HostedZone>
   <ChangeInfo>
      <Id>/change/C1PA6795UKMFR9</Id>
      <Status>PENDING</Status>
      <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
   <DelegationSet>
      <NameServers>
         <NameServer>ns-2048.awsdns-64.com</NameServer>
         <NameServer>ns-2049.awsdns-65.net</NameServer>
         <NameServer>ns-2050.awsdns-66.org</NameServer>
         <NameServer>ns-2051.awsdns-67.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</CreateHostedZoneResponse>
```

Getting the Name Servers for a Hosted Zone

When you create a hosted zone, Route 53 assigns four name servers to your hosted zone; these four name servers are called the delegation set. To ensure that the Domain Name System routes queries for your domain to the Route 53 name servers, update your registrar's or your DNS service's NS records for the domain to replace the current name servers with the names of the four Route 53 name servers in the delegation set for your hosted zone. The method that you use to update the NS records depends on which registrar or DNS service you're using.

You can get the name servers in the delegation set for a hosted zone using the Route 53 console or using the `GET GetHostedZone` API action. See the applicable topic:

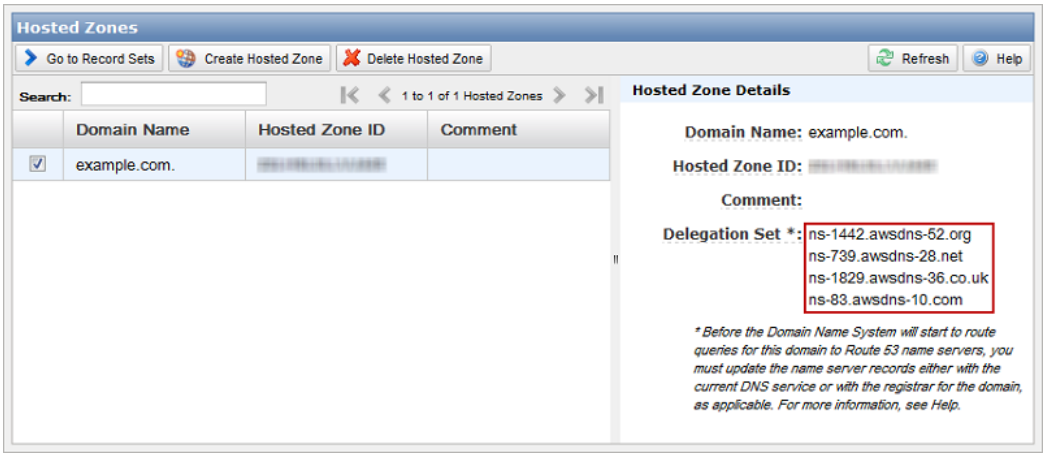
- Getting Name Servers Using the Route 53 Console (p. 39)
- Getting Name Servers Using the Route 53 API (p. 40)

Getting Name Servers Using the Route 53 Console

To get the name servers in the delegation set for a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, click the row for the hosted zone for which you want to display name servers.

The **Delegation Set** field in the right pane lists the name servers that Route 53 assigned to the hosted zone when you created it. Update your registrar's or your DNS service's NS records with the names of these name servers.



Getting Name Servers Using the Route 53 API

To get the name servers in the delegation set for a hosted zone, you use the `GetHostedZone` API action to send a `GET` request to the `2011-05-05/hostedzone/<hosted zone ID>` resource. The name servers that Route 53 assigned to the hosted zone when you created it are listed in the `DelegationSet` element of the response. For more information about the `GetHostedZone` API action, see GET GetHostedZone in the *Amazon Route 53 API Reference*.

Note

For information about using the Route 53 API, see Making API Requests (p. 83). For information about calling the Route 53 API using the Route 53 `dnscurl.pl` utility, see Example: Using dnscurl.pl (p. 91).

Example Request

```
GET /2011-05-05/hostedzone/Z1PA6795UKMFR9
```

The delegation set containing your name servers is shown in the following example response.

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/2011-05-05/">
   <HostedZone>
      <Id>/hostedzone/Z1PA6795UKMFR9</Id>
      <Name>example.com.</Name>
      <CallerReference>myUniqueIdentifier</CallerReference>
      <Config>
         <Comment>This is my first hosted zone.</Comment>
      </Config>
   </HostedZone>
   <DelegationSet>
      <NameServers>
         <NameServer>ns-2048.awsdns-64.com</NameServer>
         <NameServer>ns-2049.awsdns-65.net</NameServer>
         <NameServer>ns-2050.awsdns-66.org</NameServer>
         <NameServer>ns-2051.awsdns-67.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</GetHostedZoneResponse>
```

Listing the Hosted Zones for an AWS Account

You can create a large number of hosted zones for each AWS account, and you can list all of the hosted zones associated with an account using either the Route 53 console or the `ListHostedZones` API action.

Note

For information about how to get a list of the resource record sets in a hosted zone, see Listing Resource Record Sets (p. 56).

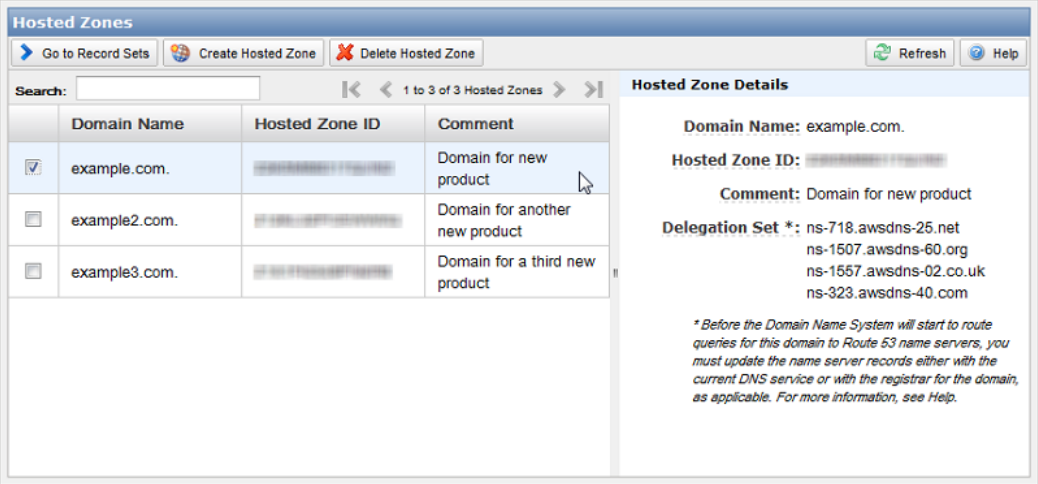
See the applicable topic:

- Listing Hosted Zones Using the Route 53 Console (p. 41)
- Listing Hosted Zones Using the Route 53 API (p. 42)

Listing Hosted Zones Using the Route 53 Console

To list the hosted zones associated with an AWS account using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, the Hosted Zones page automatically displays a list of the hosted zones that are associated with the AWS account that you are currently signed in with.



Listing Hosted Zones Using the Route 53 API

You can use the `ListHostedZones` API action to list the hosted zones that are associated with an AWS account. For more information about the `ListHostedZones` API action, see GET ListHostedZones in the *Amazon Route 53 API Reference*.

Note

For information about using the Route 53 API, see Making API Requests (p. 83).

To get a list of your hosted zones, send a `GET ListHostedZones` request to the 2011-05-05/hostedzone resource. See the following example.

Example Request

```
GET /2011-05-05/hostedzone?marker=Z2EUQ1WTGCTBG2&maxitems=10
```

The `ListHostedZones` action returns your list of hosted zones in an XML file that is contained in the body of an HTTP response.

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ListHostedZonesResponse xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <HostedZones>
      <HostedZone>
         <Id>/hostedzone/Z2EUQ1WTGCTBG2</Id>
         <Name>example2.com.</Name>
         <CallerReference>mySecondZone</CallerReference>
         <Config>
            <Comment>This is my second hosted zone.</Comment>
         </Config>
      </HostedZone>
   </HostedZones>
   <MaxItems>1</MaxItems>
   <IsTruncated>true</IsTruncated>
   <NextMarker>Z2EUQ1WTGCTBG2</NextMarker>
</ListHostedZonesResponse>
```

Each of your hosted zones is described by a `HostedZone` element that contains the domain name that is associated with your hosted zone, a Route 53 identifier of your hosted zone, and the description that you specified when you created the hosted zone. By default, the XML file that Route 53 returns lists up to 100 hosted zones, all nested in a single `HostedZones` element. If you have no hosted zones, the `HostedZones` element is empty.

If you have a lot of hosted zones, you can use pagination to get the list in groups of up to 100 hosted zones at a time. You can also paginate your list into separate XML files that you get one at a time, one with each request. The `ListHostedZones` action provides `Marker`, `NextMarker`, and `MaxItems` parameters that you can use to determine the hosted zones that are returned in the XML file included with each response. `Marker` specifies the hosted zone to begin the list with and is only included if you specified a marker parameter when you called `ListHostedZones`. After you make your initial request for three items, to get another XML file containing the next three, you make a follow-up request with the value of `Marker` set to `NextMarker`. To paginate through all of your hosted zones, make an initial request to the `ListHostedZones` action with `MaxItems` set to the number of hosted zones that you want to get in each page. As long as the `IsTruncated` element in the response is set to true, continue to make requests to the `ListHostedZones` action with the marker element set to the value of the `NextMarker` element from the previous response.

The following examples show how to use the `MaxItems` element to control the listing of hosted zones.

Example Request

The following example shows the request with the `MaxItems` element specified as 1.

```
GET /2011-05-05/hostedzone?maxitems=1
```

Example Response

This example shows the response for the previous example in which `MaxItems` is specified as 1.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ListHostedZonesResponse xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <HostedZones>
      <HostedZone>
         <Id>/hostedzone/Z2EUQ1WTGCTBG2</Id>
         <Name>example2.com.</Name>
         <CallerReference>MyUniqueIdentifier2</CallerReference>
         <Config>
            <Comment>This is my second hosted zone.</Comment>
         </Config>
      </HostedZone>
   </HostedZones>
   <MaxItems>1</MaxItems>
   <IsTruncated>true</IsTruncated>
   <NextMarker>Z2EUQ1WTGCTBG2</NextMarker>
</ListHostedZonesResponse>
```

Example Follow-up Request

This example shows the follow-up request to the previous request with the `MaxItems` element specified as 10 and the list starting with the marker set to `Z2EUQ1WTGCTBG2`.

```
GET /2011-05-05/hostedzone?marker=Z2EUQ1WTGCTBG2&maxitems=10
```

Example Follow-up Response

This example shows the response for the previous example.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ListHostedZonesResponse xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <HostedZones>
      <HostedZone>
         <Id>/hostedzone/Z3AEGXETSR30VB</Id>
         <Name>example3.com.</Name>
         <CallerReference>MyUniqueIdentifier3</CallerReference>
         <Config>
            <Comment>This is my third hosted zone.</Comment>
         </Config>
      </HostedZone>
      <HostedZone>
         <Id>/hostedzone/Z2682N5HXP0BZ4</Id>
         <Name>example.com.</Name>
         <CallerReference>MyUniqueIdentifier4</CallerReference>
         <Config>
            <Comment>This is my fourth hosted zone.</Comment>
         </Config>
      </HostedZone>
   </HostedZones>
   <MaxItems>10</MaxItems>
   <IsTruncated>false</IsTruncated>
</ListHostedZonesResponse>
```

Deleting a Hosted Zone

You can delete a hosted zone using the Route 53 console or the Route 53 API.

See the applicable topic:

- Deleting a Hosted Zone Using the Route 53 Console (p. 45)
- Deleting a Hosted Zone Using the Route 53 API (p. 47)

Deleting a Hosted Zone Using the Route 53 Console

Important

You can delete a hosted zone only if there are no resource record sets other than the default SOA and NS records. If your hosted zone contains other resource record sets, you must delete them before you can delete your hosted zone. This prevents you from accidentally deleting a hosted zone that still contains resource record sets.

To delete a hosted zone using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. Confirm that the hosted zone that you want to delete contains only an NS and an SOA resource record set. If it contains additional resource record sets, delete them:

   a. In the Route 53 console, do one of the following to display the record sets for the hosted zone that you want to delete:

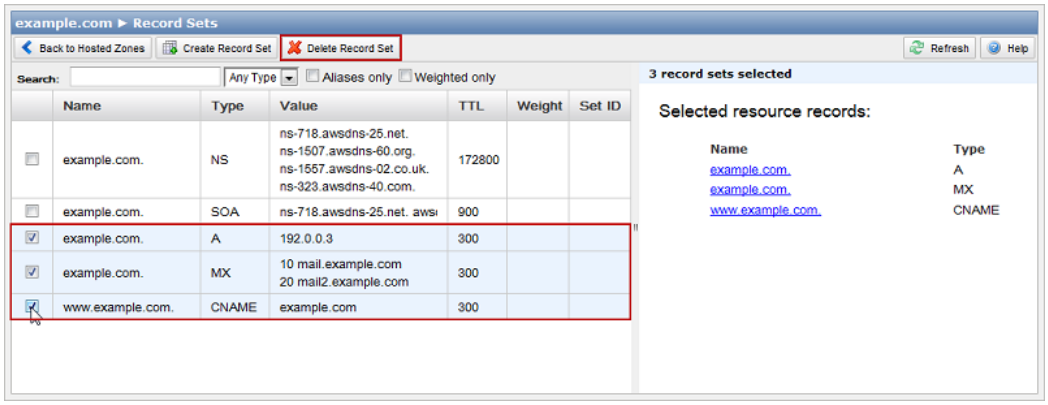
      - Click the row for the hosted zone that you want to delete, and click **Go to Record Sets**.
      - Double-click the row for the hosted zone that you want to delete.

   b. On the Record Sets page, if the list of resource record sets includes any resource record sets for which the value of the **Type** column is something other than NS or SOA, click the row, and click **Delete Record Set**.

      To select multiple, consecutive resource record sets, click the first row, press and hold the **Shift** key, and click the last row. To select multiple, non-consecutive resource record sets, click the first row, press and hold the **Ctrl** key, and click the remaining rows.

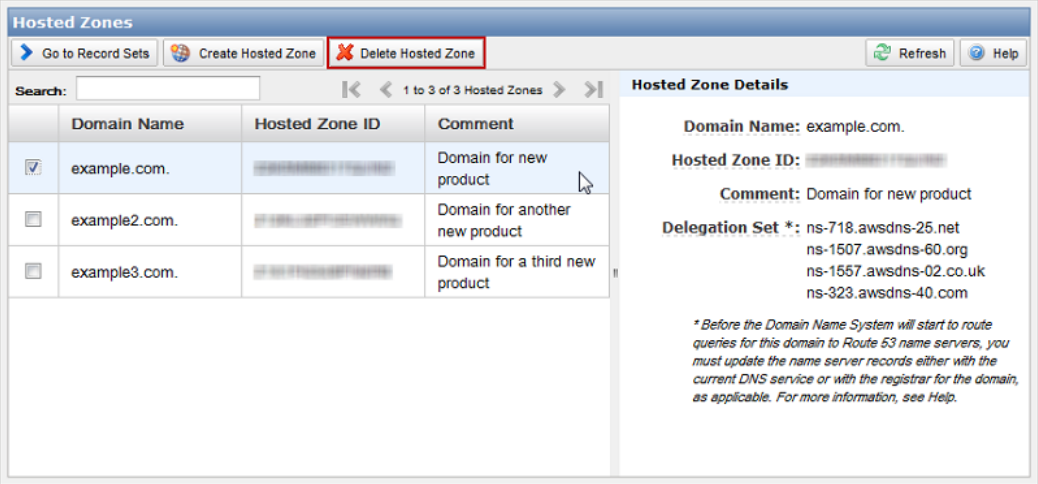
      Note

      If you created any NS records for subdomains in the hosted zone, delete those records, too.

   c. Click **Back to Hosted Zones**.

3. On the Hosted Zones page, click the row for the hosted zone that you want to delete.



4. Click **Delete Hosted Zone**.
5. Click **OK** to confirm.

Deleting a Hosted Zone Using the Route 53 API

Important

You can delete a hosted zone only if there are no resource record sets other than the default SOA and NS records. If your hosted zone contains other resource record sets, you must delete them before you can delete your hosted zone. This prevents you from accidentally deleting a hosted zone that still contains resource record sets.

To delete a hosted zone using the Route 53 API, send a `DELETE` request to the `2011-05-05/hostedzone/<hosted zone ID>` resource that specifies the ID of the hosted zone. The following examples show a request to delete a Route 53 hosted zone and the response from Route 53.

Note

For information about using the Route 53 API, see Making API Requests (p. 83). For information about calling the Route 53 API using the Route 53 `dnscurl.pl` utility, see Example: Using dnscurl.pl (p. 91).

Example Request

```
DELETE /2011-05-05/hostedzone/Z1PA6795UKMFR9
```

Example Response

```
HTTP/1.1 200 OK <?xml version="1.0" encoding="UTF-8"?>
<DeleteHostedZoneResponse xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <ChangeInfo>
      <Id>/change/C1PA6795UKMFR9</Id>
         <Status>PENDING</Status>
         <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</DeleteHostedZoneResponse>
```

To verify that the hosted zone has been deleted, do one of the following:

- Use the `GetHostedZone` action to request information about the hosted zone. For more information, see GET GetHostedZone in the *Amazon Route 53 API Reference*.
- Use the `ListHostedZones` action to get a list of the hosted zones associated with the AWS account that created the hosted zone. For more information, see GET ListHostedZones in the *Amazon Route 53 API Reference*.

Working with Resource Record Sets

Topics

- Creating, Changing, and Deleting Resource Record Sets (p. 49)
- Listing Resource Record Sets (p. 56)

After you create a hosted zone for a specified domain, for example, example.com, you create resource record sets to tell the Domain Name System how you want traffic to be routed for that domain. You might create resource record sets that cause DNS to route Internet traffic for example.com to the IPv4 or IPv6 address of a host in your data center, route email for that domain (ichiro@example.com) to a mail server (mail.example.com), and route traffic for a subdomain called operations.tokyo.example.com to the IPv4 or IPv6 address of a different host. Each resource record set includes the name of a domain or a subdomain, a record type (for example, an MX record routes email), and other information applicable to the record type (for MX records, the host name of a mail server and the priority of that server). For information about the different types of resource records, see DNS Domain Name Format (p. 3).

The name of each resource record set in a hosted zone must end with the name of the hosted zone. For example, the example.com hosted zone can contain resource record sets for www.example.com and accounting.tokyo.example.com subdomains, but cannot contain resource record sets for a www.example.ca subdomain.

Creating, Changing, and Deleting Resource Record Sets

You can use the Route 53 console or Route 53 API actions to create, change, and delete resource record sets.

Note

Creating, changing, and deleting resource record sets take time to propagate to the Route 53 DNS servers. Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

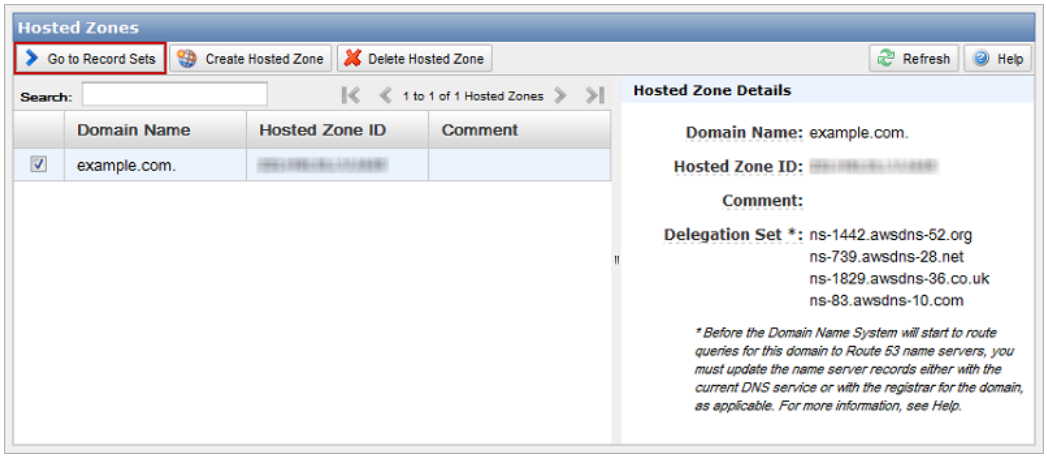
See the applicable topic:

- Creating, Changing, and Deleting Resource Record Sets Using the Route 53 Console (p. 50)
- Creating, Changing, and Deleting Resource Record Sets Using the Route 53 API (p. 53)

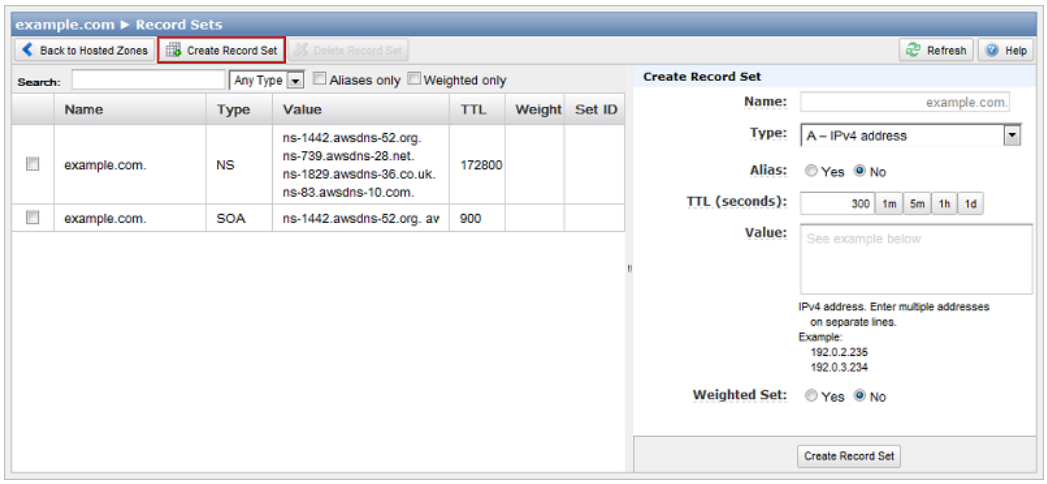
Creating, Changing, and Deleting Resource Record Sets Using the Route 53 Console

To create resource record sets using the Route 53 console

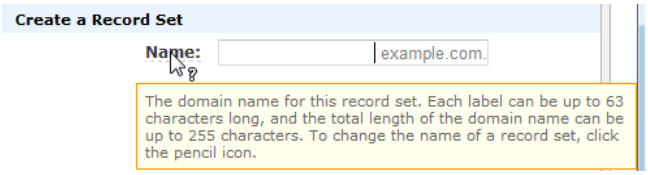
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:
   - Click the row for the hosted zone in which you want to create record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. On the Record Sets page, above the left pane, click **Create Record Set**.



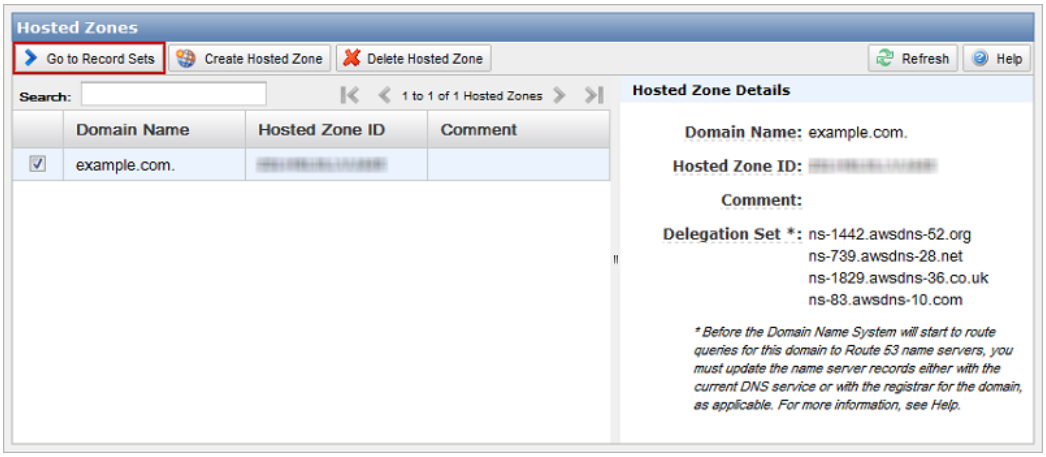
4. In the right pane, enter the applicable values. For information about a field, see the tool tip for the field.



5. Below the right pane, click **Create Record Set**.

To change resource record sets using the Route 53 console

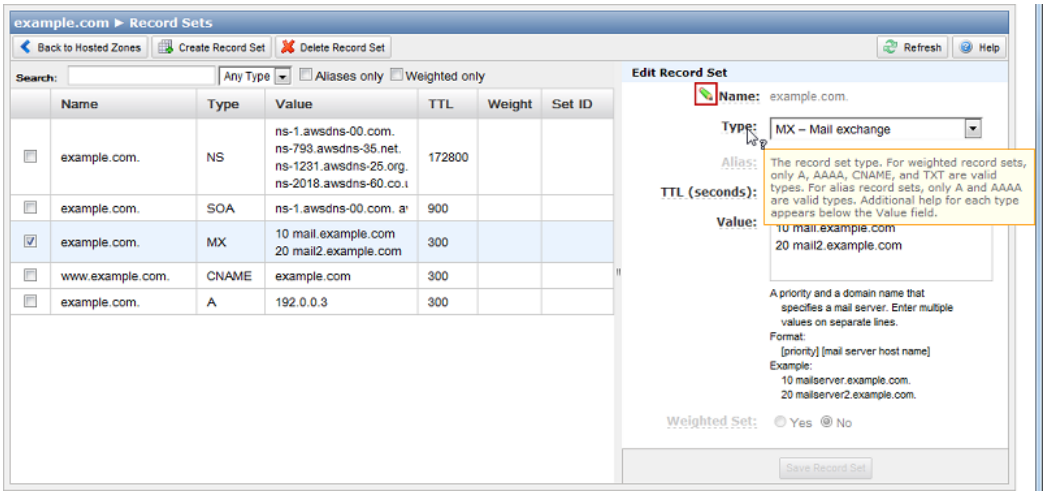
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:
   - Click the row for the hosted zone in which you want to change record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. In the left pane, select the resource record set that you want to change.
4. Change the applicable values. For information about a field, see the tool tip for the field.

Note

To change the name of the record set, click the pencil icon.

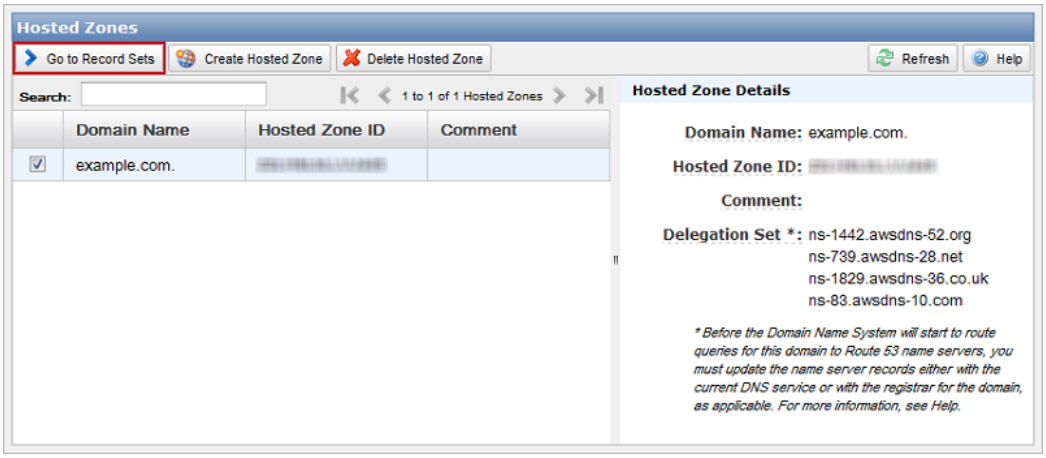


5. Click **Save**.

To delete resource record sets using the Route 53 console

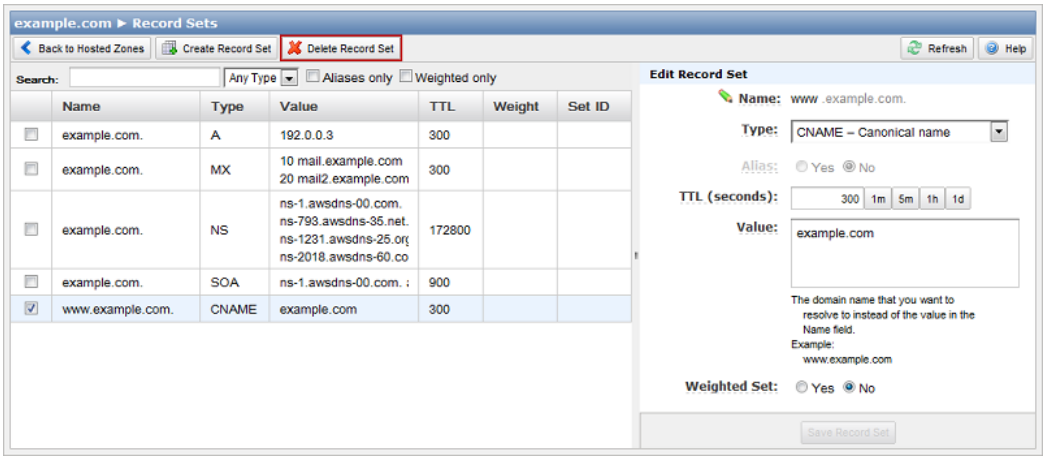
1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:

   - Click the row for the hosted zone in which you want to change record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



3. In the left pane, select the resource record set that you want to delete.

   To select multiple, consecutive resource record sets, click the first row, press and hold the **Shift** key, and click the last row. To select multiple, non-consecutive resource record sets, click the first row, press and hold the **Ctrl** key, and click the remaining rows.
4. Click **Delete Record Set**.



5. Click **OK** to confirm.

Creating, Changing, and Deleting Resource Record Sets Using the Route 53 API

To create, change, or delete resource record sets using the Route 53 API, use the Route 53 API `ChangeResourceRecordSets` action. These changes apply the information from your change batch request to your Route 53 hosted zone.

How Change Batch Requests Work

When you use the Route 53 API to change resource record sets, Route 53 either makes all of the changes or makes none of the changes in a change batch request. This ensures that Route 53 never partially implements the changes that you intended to make to the resource record sets in a hosted zone.

For example, suppose you create a change batch request that deletes the A record for example.com and creates a replacement A record that has a different IP address. Route 53 deletes the first resource record set and creates the second resource record set in a single operation. If either the `DELETE` or the `CREATE` action fails, then both changes (plus any other changes in the batch) fail, and the original A record continues to exist.

Creating a Change Batch Request

To create a change batch request, use the `ChangeResourceRecordSets` action `ChangeBatch` element. You use `CREATE` and `DELETE` actions within the `ChangeBatch` element for each record that you want to update. If you are only creating records, then you will only use `CREATE` actions.

Note the following limitations on a `ChangeResourceRecordSets` request:

- A request cannot contain more than 100 `Change` elements.
- A request cannot contain more than 1000 `ResourceRecord` elements.
- The sum of the number of characters (including spaces) in all `Value` elements in a request cannot exceed 32,000 characters.

The following is an example of a `ChangeResourceRecordSets` change batch request. For more information about the `ChangeResourceRecordSets` API action, see POST ChangeResourceRecordSets in the *Amazon Route 53 API Reference*.

Note

For information about using the Route 53 API, see Making API Requests (p. 83). For information about calling the Route 53 API using the Route 53 `dnscurl.pl` utility, see Example: Using dnscurl.pl (p. 91).

Example of a Change Batch Request

This example creates a *TXT record* for `www.example.com` and changes the *A record* for `foo.example.com` from `192.0.2.3` to `192.0.2.1`.

```
POST /2011-05-05/hostedzone/Z1PA6795UKMFR9/rrset HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <ChangeBatch>
      <Comment>
      This change batch creates a TXT record for www.example.com. and
      changes the A record for foo.example.com. from
      192.0.2.3 to 192.0.2.1.
      </Comment>
      <Changes>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>www.example.com.</Name>
               <Type>TXT</Type>
               <TTL>600</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>"item 1" "item 2" "item 3"</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>DELETE</Action>
            <ResourceRecordSet>
               <Name>foo.example.com.</Name>
               <Type>A</Type>
               <TTL>600</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.3</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>foo.example.com.</Name>
               <Type>A</Type>
               <TTL>600</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.1</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
      </Changes>
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

Checking the Status of Your Change

After you send your change request, you should check to ensure that your change has propagated to all Route 53 DNS servers. To check the status of your change, poll for the status of the change. Use the `GetChange` action to send a GET request to the `2011-05-05/change/<change ID>` resource. When your change is complete, its status becomes `INSYNC`. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

For more information about the `GetChange` API action, see GET GetChange in the *Amazon Route 53 API Reference*.

In the following example, the `Id` value is the `ChangeInfo Id` value from the `ChangeResourceRecordSetsResponse` header.

Note

For information about using the Route 53 API, see Making API Requests (p. 83). For information about calling the Route 53 API using the Route 53 `dnscurl.pl` utility, see Example: Using dnscurl.pl (p. 91).

Example Request

```
GET /2011-05-05/change/C2682N5HXP0BZ4
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetChangeResponse xmlns="https://route53.amazonaws.com/doc/2011-05-05/">
  <ChangeInfo>
     <Id>/change/C2682N5HXP0BZ4</Id>
     <Status>INSYNC</Status>
     <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</GetChangeResponse>
```

Listing Resource Record Sets

You can use the Route 53 console or Route 53 API actions to list the resource record sets in a hosted zone. See the applicable topic:

- Listing Resource Record Sets in the Route 53 Console (p. 57)
- Listing Resource Record Sets Using the Route 53 API (p. 57)

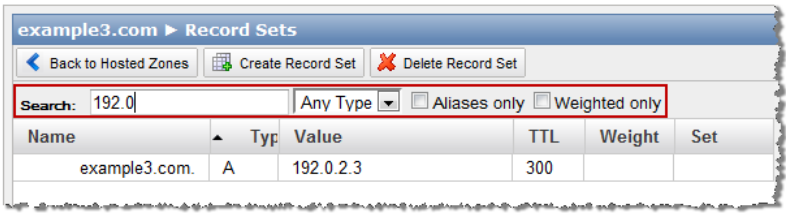
Listing Resource Record Sets in the Route 53 Console

To list resource record sets in the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the Route 53 console, click the name of the hosted zone for which you want to list resource record sets, and the Record Sets page appears.

   To display only selected resource record sets, enter the applicable search criteria above the list of resource record sets:

   - To display the resource record sets that have specific values in either the **Name** or **Value** field, enter a value in the **Search** field. For example, to display the resource record sets that have an IP address beginning with **192.0**, type that value in the **Search** field.
   - To display only the resource record sets that have the same DNS record type, select the type in the drop down list.
   - To display only alias resource record sets, check the **Alias Only** check box.
   - To display only weighted resource record sets, check the **Weighted Only** check box.



Listing Resource Record Sets Using the Route 53 API

To get a list of resource record sets in a hosted zone using the Route 53 API, you use the `ListResourceRecordSets` action. For this action, the format of the response depends on how you specify the `ListResourceRecordSets` elements in your request. This section describes how to specify the `Type` and `Name` elements and what you need to consider when listing records concurrently with a `ChangeResourceRecordsSets` request.

The `ListResourceRecordSets` action sorts results first by DNS name (ordered by top-level domain, for example `com.amazon.www`), and second by the record type. This procedure explains how to specify the request to control the response format. For more information about how `Name` and `Type` interact, see the table Name and Type Specification and Result (p. 59).

Note

For information about using the Route 53 API, see Making API Requests (p. 83). For information about calling the Route 53 API using the Route 53 `dnscurl.pl` utility, see Example: Using dnscurl.pl (p. 91).

To get a list of resource record sets in a hosted zone

1. To retrieve the first page of resource record sets, send a `ListResourceRecordSets` request and do not specify `Name` or `Type` values.

   For example:

```
GET /2011-05-05/hostedzone/Z1PA6795UKMFR9/rrset?maxitems=1
```

   Your response might look like the following.

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ListResourceRecordSetsResponse xmlns="https://route53.amazonaws.com/doc/2011-
05-05/">
  <ResourceRecordSets>
    <ResourceRecordSet>
      <Name>example.com.</Name>
      <Type>A</Type>
      <TTL>600</TTL>
      <ResourceRecords>
        <ResourceRecord>
          <Value>192.0.2.3</Value>
        </ResourceRecord>
      </ResourceRecords>
    </ResourceRecordSet>
  </ResourceRecordSets>
  <IsTruncated>true</IsTruncated>
  <NextRecordName>example.com.</NextRecordName>
  <NextRecordType>NS</NextRecordType>
  <MaxItems>1</MaxItems>
</ListResourceRecordSetsResponse>
```

2. For subsequent calls, use the `Name` and `Type` elements to control the length and ordering of the list. Set `Name` and `Type` to the `NextRecordName` and `NextRecordType` values from the previous response.

   The follow-up request and response might look like the following.

```
GET /2011-05-05/hostedzone/Z1PA6795UKMFR9/rrset?type=NS&name=example.com&max
items=10
```

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ListResourceRecordSetsResponse xmlns="https://route53.amazonaws.com/doc/2011-
05-05/">
  <ResourceRecordSets>
    <ResourceRecordSet>
      <Name>example.com.</Name>
      <Type>NS</Type>
      <TTL>172800</TTL>
      <ResourceRecords>
        <ResourceRecord>
          <Value>ns-2048.awsdns-64.org.</Value>
        </ResourceRecord>
        <ResourceRecord>
```

```
          <Value>ns-2049.awsdns-65.co.uk.</Value>
        </ResourceRecord>
        <ResourceRecord>
          <Value>ns-2050.awsdns-66.net.</Value>
        </ResourceRecord>
        <ResourceRecord>
          <Value>ns-2051.awsdns-67.com.</Value>
        </ResourceRecord>
      </ResourceRecords>
    </ResourceRecordSet>
    <ResourceRecordSet>
      <Name>example.com.</Name>
      <Type>SOA</Type>
      <TTL>900</TTL>
      <ResourceRecords>
        <ResourceRecord>
          <Value>ns-2048.awsdns-64.co.uk. awsdns-hostmaster.amazon.com. 1
7200 900 1209600 86400</Value>
        </ResourceRecord>
      </ResourceRecords>
    </ResourceRecordSet>
  </ResourceRecordSets>
  <IsTruncated>false</IsTruncated>
  <MaxItems>10</MaxItems>
</ListResourceRecordSetsResponse>
```

When you make list requests this way, use the `Name` and `Type` elements to control the length and the order of the list of resource record sets. The ways in which you might use `Name` and `Type` are described in the following table.

Name and Type Specification and Result

Specification	Results
If you do not specify `Name` or `Type`	The results begin with the first resource record set that the hosted zone contains.
If you specify `Name` but not `Type`	The results begin with the first resource record set in the list whose name is greater than or equal to `Name`.
If you specify `Type` but not `Name`	Route 53 returns the `InvalidInput` error.
If you specify both `Name` and `Type`	The results begin with the first resource record set in the list whose name is greater than or equal to `Name`, and whose type is greater than or equal to `Type`.

For more information about the `ListResourceRecordSets` action and descriptions of its elements, see ListResourceRecordSets in the Amazon Route 53 API Reference.

Note

The `ListResourceRecordSets` action returns the most current version of the records. This includes records that are `PENDING` and not yet available on all Route 53 DNS servers.

What Happens If You Change Resource Record Sets While You're Listing Them

To ensure that you get an accurate listing of the resource record sets for a hosted zone at a point in time, do not submit a `ChangeResourceRecordSets` request while you are paging through the results of a `ListResourceRecordSets` request. If you do, some pages may display results without the latest changes while other pages display results with the latest changes.

For example, suppose you're viewing the results for the example.com domain. You have A records for the following subdomains:

- sub1.example.com
- sub3.example.com
- sub5.example.com

You submit a `ChangeResourceRecordSets` request that makes the following changes:

- Change the IP address of the A record for sub1.example.com by deleting the existing resource record set and creating a new one.
- Add an A record for sub2.example.com.
- Change the IP address of the A record for sub3.example.com.
- Add an A record for sub4.example.com.
- Change the IP address of the A record for sub5.example.com.

When you submitted the `ListResourceRecordSets` request, you specified 1 for the value of the `maxitems` parameter, so you're viewing the resource record sets in the example.com hosted zone one at a time. `ListResourceRecordSets` returns the results in alphabetical order by domain name. You've already viewed the A record for sub1.example.com, and you're viewing the A record for sub3.example.com when you submit the change request. The change request completes before you request the next resource record set from the `ListResourceRecordSets` request. Here is what happens:

- You don't see the change to the A records for sub1.example.com or sub3.example.com because the `ListResourceRecordSets` request got the data on those resource record sets before the change request was submitted.
- You don't see the new A record for sub2.example.com because sub2.example.com appears alphabetically before the A record for sub3.example.com.
- When you request the page after sub3.example.com, you don't see the new A record for sub4.example.com. When Route 53 returned data about the A record for sub3.example.com, it also returned a marker that identifies the next record by resource record set name and type, and the next record at that time was the A record for sub5.example.com.
- When you request the page after sub3.example.com, you do see the change to the IP address for the A record for sub5.example.com. The change request deleted the old record and created a new record with the new IP address before you requested the data for the record that appears after sub3.example.com.

Creating Weighted Resource Record Sets

Weighted resource record sets let you associate multiple answers with a single DNS name. This can be useful for a variety of purposes including load balancing and testing new versions of software. To create a group of weighted resource record sets, you create two or more resource record sets that have the same combination of DNS name and type, and you assign each resource record set a unique identifier and a relative weight.

When processing a DNS query, Route 53 searches for a resource record set that matches the specified name and type. If a group of resource record sets have the same name and type, Route 53 selects one from that group. The probability of any one resource record set being selected depends on its weight as a proportion of the total weight for all resource record sets in the group:

$$\frac{\text{Weight for a given resource record set}}{\text{Sum of the weights for the resource record sets}}$$

For example, suppose you create three resource record sets for www.example.com. The three A records have weights of 1, 1, and 3 (sum = 5). On average, Route 53 selects each of the first two resource record sets one-fifth of the time, and returns the third resource record set three-fifths of the time.

You can create weighted resource record sets using the Route 53 console or the Route 53 API. See the applicable topic:

- Creating Weighted Resource Record Sets Using the Route 53 Console (p. 62)
- Creating Weighted Resource Record Sets Using the Route 53 API (p. 63)

Creating Weighted Resource Record Sets Using the Route 53 Console

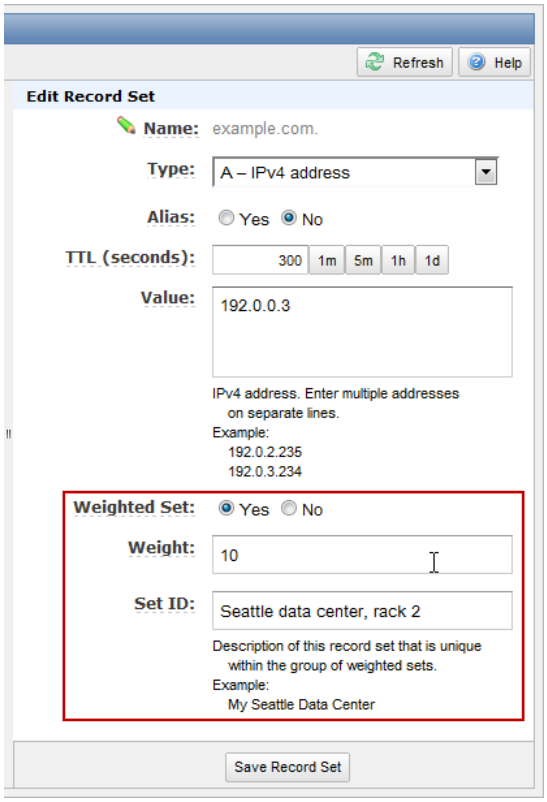
To create weighted resource record sets using the Route 53 console

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the Hosted Zones page, do one of the following:

   - Click the row for the hosted zone in which you want to create weighted resource record sets, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.

3. Click **Create Record Set**.
4. Enter the applicable values. For information about a field, see the tool tip for the field. Note the following:

   - For **Name**, specify the same value for all of the resource record sets that you want to include in the same group of weighted resource record sets.
   - For **Type**, select A, AAAA, CNAME, or TXT. Select the same value for all of the resource record sets that you want to include in the same group of weighted resource record sets.
   - For the value of **Weight**, enter an integer between 0 and 255. To disable routing to a resource, set **Weight** to 0. If you set **Weight** to 0 for all of the resource record sets in the group, traffic is routed to all resources with equal probability. This ensures that you don't accidentally disable routing for a group of weighted resource record sets.
   - For **Set ID**, enter a value that is unique for the current combination of **Name** and **Type**, for example, **Seattle data center, rack 2, position 4**.



5. Click **Create Record Set**.
6. Repeat steps 3 through 5 for each of the weighted resource record sets that you want to add to the group.

Creating Weighted Resource Record Sets Using the Route 53 API

To use the Route 53 API to create a group of weighted resource record sets for a given combination of `Name` and `Type`, use the Route 53 API `ChangeResourceRecordSets` action and include `SetIdentifier` and `Weight` elements in each resource record set. Because `Name` and `Type` will be the same for each resource record set in the group, you must specify unique values for `SetIdentifier` to differentiate resource record sets from one another. This unique value is also useful when you retrieve resource records sets using `ListResourceRecordSets`.

For more information about the `ChangeResourceRecordSets` action, see POST ChangeResourceRecordSets in the Amazon Route 53 API Reference.

Weighted resource record sets are supported only for A, AAAA, CNAME, and TXT record types.

To disable routing to a resource, set `Weight` to 0. If you set `Weight` to 0 for all of the resource record sets in the group, traffic is routed to all resources with equal probability. This ensures that you don't accidentally disable routing for a group of weighted resource record sets.

The following example deletes the A record for www.example.com, recreates it with `SetIdentifier` and `Weight` elements, and creates an additional A record for www.example.com with a different `SetIdentifier`, `Weight`, and `Value` (IP address). One-fourth of traffic (1/(1+3)) for www.example.com

will be routed to 192.0.2.1, and three-fourths of traffic (3/(1+3)) will be routed to 192.0.2.2. We delete the existing A record for www.example.com because you cannot have both weighted and unweighted resource record sets that have the same `Name` and `Type`.

```
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <ChangeBatch>
      <Comment>
      This change batch deletes the A record for www.example.com,
      recreates it as a weighted record, and creates another A record with a
      different SetIdentifier, Weight, and Value (IP address).
      </Comment>
      <Changes>
         <Change>
            <Action>DELETE</Action>
            <ResourceRecordSet>
               <Name>www.example.com.</Name>
               <Type>A</Type>
               <TTL>600</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.1</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>www.example.com.</Name>
               <Type>A</Type>
               <SetIdentifier>Rack 2, Position 4</SetIdentifier>
               <Weight>1</Weight>
               <TTL>600</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.1</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>www.example.com.</Name>
               <Type>A</Type>
               <SetIdentifier>Rack 5, Position 1</SetIdentifier>
               <Weight>3</Weight>
               <TTL>600</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.2</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
      </Changes>
```

```
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

After you send your change request, use the `GetChange` API action to poll for the status of the change to confirm that it has propagated to all Route 53 DNS servers. When your change has finished propagating, its status changes from `PENDING` to `INSYNC`.

Note

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

For more information about using the `GetChange` API action, see GET GetChange in the *Amazon Route 53 API Reference*.

For more information about working with resource record sets and the Route 53 API, see the following related topics.

Related Topics

- Creating, Changing, and Deleting Resource Record Sets (p. 49)
- Listing Resource Record Sets (p. 56)
- Making API Requests (p. 83)
- Getting Started with Route 53 (p. 10)

Using Weighted Resource Record Sets with the 2010-10-01 Route 53 API

Weighted resource record sets were added starting with the 2011-05-05 version of the Route 53 API. If a zone contains any weighted resource record sets, we recommend that you use the 2011-05-05 API. Note the following about using weighted resource record sets with the 2010-10-01 API.

- The 2010-10-01 version of the `ChangeResourceRecordSets` action cannot create or delete resource record sets that include `SetIdentifier` and `Weight` elements.
- The 2010-10-01 version of the `ListResourceRecordSets` action can list resource record sets that include the `SetIdentifier` and `Weight` elements, but those elements are not included in the output. Instead, the `Value` element contains a message that says the resource record set includes an unsupported attribute.
- The 2010-10-01 version of the `ListResourceRecordSets` action can paginate resource record sets that include `SetIdentifier` and `Weight` elements, but the combination of `Name` and `Type` is no longer guaranteed to correspond with a single resource record set. To ensure that the 2010-10-01 `ListResourceRecordSets` paginator makes progress and doesn't repeat a partial list of the resource record sets for a given `Name` and `Type`, specify a page size that is long enough to accommodate the largest number of resource record sets that are associated with any `Name` and `Type` combination in the zone.

Creating Alias Resource Record Sets for Elastic Load Balancing

Ordinary Route 53 resource record sets are direct: They contain the DNS resource records that Route 53 uses to respond to DNS queries.

Route 53 *alias* resource record sets are indirect: They do not contain DNS resource records. Instead, an alias resource record set contains a pointer to a resource record set that contains DNS resource records. When a DNS query arrives that matches the name and type in an alias resource record set, Route 53 follows the pointer and responds as if the query had asked for the resource record set that is referenced by the pointer.

Note

Currently, alias resource record sets are supported for use only with Elastic Load Balancing.

The indirection of alias resource record sets can save you time because Route 53 automatically recognizes changes in the resource record sets that are referenced by the **Alias Target** field in the Route 53 console or by the `HostedZoneId` and `DNSName` elements in a `POST ChangeResourceRecordSets` API action. This means that you do not need to manually change the alias resource record sets. For example, suppose an alias resource record set for *example.com* points to a load balancer at *lb1-1234.us-east-1.elb.amazonaws.com*. If the IP address of the load balancer changes, Route 53 will automatically reflect those changes in DNS answers for *example.com* without any changes to the hosted zone that contains resource record sets for *example.com*.

Alias resource records sets are similar to CNAME records, but there are some important differences:

- You cannot create a CNAME record for the top node of a DNS namespace, also known as the *zone apex*. (For example, if you register the DNS name example.com, the zone apex is example.com.) You can create an alias resource record set for the zone apex.
- A CNAME record redirects traffic for a domain name regardless of record type. Route 53 follows the pointer in an alias resource record set only when the record type also matches.
- A CNAME record can point to any DNS record hosted anywhere. Currently, alias resource record sets can only point to Elastic Load Balancing load balancers.
- A CNAME record is visible in the answer section of a reply from a Route 53 DNS server. An alias resource record set is only visible in the Route 53 XML control API.

- A CNAME record is followed by a recursive resolver. An alias resource record set is only followed inside Route 53. This means that both the alias resource record set and its target must exist in Route 53.

Note

You cannot set the time to live (TTL) for alias records; Route 53 honors the Elastic Load Balancing TTL. For the current TTL value for Elastic Load Balancing, see the introduction to Using Domain Names With Elastic Load Balancing in the *Elastic Load Balancing Developer Guide*.

The following table describes how you create an alias or weighted alias resource record sets.

Process for Creating an Alias Resource Record Set or Weighted Alias Resource Record Sets

1	Use Elastic Load Balancing to set up a load balancer. If you're creating weighted alias resource record sets, create one load balancer for each weighted alias resource record set. For more information, see User Scenarios in the *Elastic Load Balancing Developer Guide*. **Note** Give the load balancer a name that will help you remember what it's for later. The name you specify when you create a load balancer is the name you'll choose when you create an alias or weighted alias resource record set in Route 53.
2	Create a Route 53 hosted zone. For more information, see Creating a Domain that Uses Route 53 as the DNS Service (p. 17). **Note** For information about using the Route 53 API, see Making API Requests (p. 83). For information about calling the Route 53 API using the Route 53 `dnscurl.pl` utility, see Example: Using dnscurl.pl (p. 91).
3	Create an alias resource record set or weighted alias resource record sets in your hosted zone. For more information, see the applicable section: • How to Create an Alias Resource Record Set (p. 67) • How to Create Weighted Alias Resource Record Sets (p. 72)
4	Check on the status of your changes. For more information, see Checking the Status of Your Change (p. 56).

How to Create an Alias Resource Record Set

You can create an alias resource record set using the Route 53 console or the Route 53 API. See the applicable topic:

- Creating an Alias Resource Record Set Using the Route 53 Console (p. 68)

- Creating an Alias Resource Record Set Using the Route 53 API (p. 70)

Creating an Alias Resource Record Set Using the Route 53 Console

To create an alias resource record set using the Route 53 console

1. If you created your Route 53 hosted zone and your Elastic Load Balancing load balancer using the same AWS account, skip to Step 2.

   If you created the hosted zone and the load balancer using different accounts, perform the following steps to get the DNS name for the load balancer:

   a. Sign in to the AWS Management Console using the AWS account that was used to create the load balancer for which you want to create an alias resource record set.

   b. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

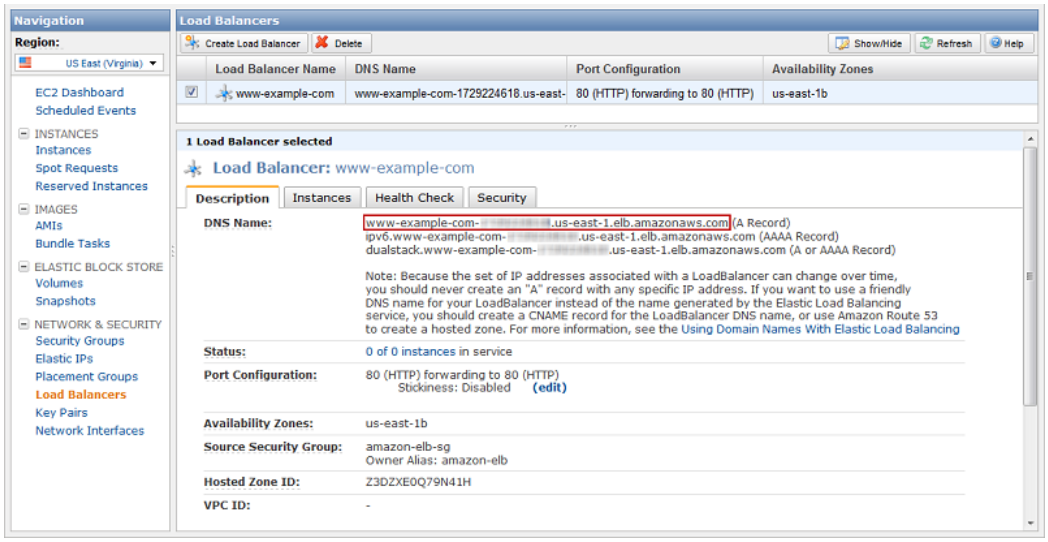
   c. In the **Navigation** pane, click **Load Balancers**.

   d. In the **Load Balancers** pane, select the load balancer for which you want to create a Route 53 alias resource record set.

   e. In the bottom pane, on the **Description** tab, get the A record value for **DNS Name**.

   Important

   Do not use the value for the AAAA record or the dualstack (A or AAAA record) record.

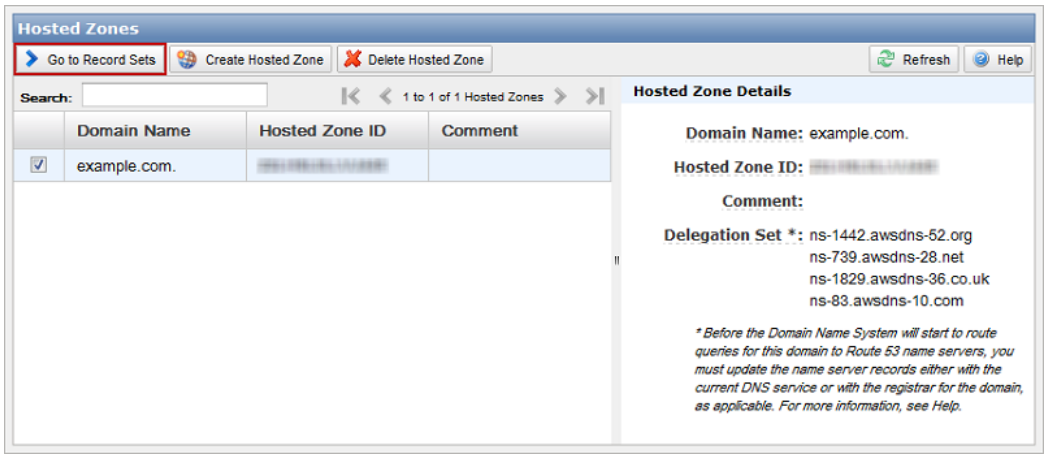


   f. Sign out of the AWS Management Console.

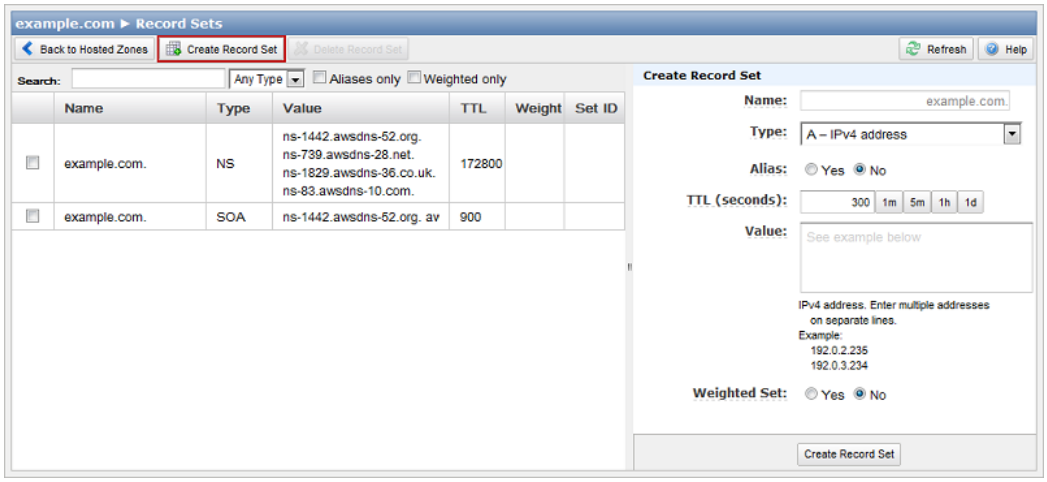
   g. Sign in to the AWS Management Console again using the AWS account that was used to create the Route 53 hosted zone.

2. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
3. In the Route 53 console, do one of the following to display the record sets for the hosted zone in which you want to create alias resource record sets:

- Click the row for the hosted zone, and click **Go to Record Sets**.
- Double-click the row for the hosted zone.

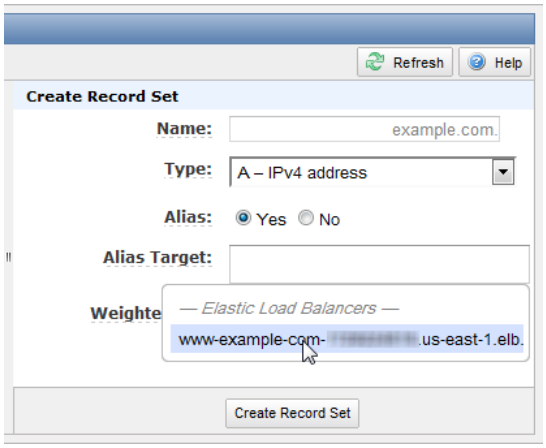


4. Click **Create Record Set**.



5. Enter the applicable values. For information about a field, see the tool tip for the field. Note the following:

   - For **Type**, select A or AAAA.
   - For **Alias**, click **Yes**.
   - For **Alias Target**, do one of the following:
     - If the load balancer and the hosted zone were created by the same AWS account, click the **Alias Target** field, and choose a load balancer from the list. If you have a lot of load balancers, you can type the first few characters of the DNS name to filter the list.
     - If the load balancer and the hosted zone were created using different AWS accounts, enter the value that you got in Step 1 for the DNS name of the load balancer.
   - The value of **Alias Hosted Zone ID** appears automatically based on the value that you selected or entered for **Alias Target**.



6. Click **Create Record Set**.

Creating an Alias Resource Record Set Using the Route 53 API

To create an alias resource record set, you use the `ChangeResourceRecordSets` action to create one or more resource record sets that define the alias.

In the following example:

- *Z222222222* is the ID of the Route 53 hosted zone in which you're creating the alias resource record set.
- *example.com* is the zone apex for which you want to create an alias.
- *www.example.com* is a subdomain for which you also want to create an alias.
- *Z1111111111111* is an example of a hosted zone ID for the load balancer. You can get the value of the load balancer's hosted zone ID, `CanonicalHostedZoneNameID`, in the following ways:
  - Use the AWS Management Console. On the EC2 page, click **Load Balancers** in the **Navigation** pane.
  - Use the `DescribeLoadBalancers` action. For more information, see "Actions" in the *Elastic Load Balancing API Reference*.
  - Use the Elastic Load Balancing `elb-describe-lbs` CLI command. For more information about this CLI command, see the *Elastic Load Balancing Quick Reference Card* or the Elastic Load Balancing CLI help.
- *example-load-balancer-1111111111.us-east-1.elb.amazonaws.com* is an example of a load balancer domain name with which Route 53 responds to queries for *example.com* and *www.example.com*.

  Use the DNS name that was returned by the Elastic Load Balancing `CreateLoadBalancer` action, which is described in the "Actions" chapter of the *Elastic Load Balancing API Reference*. You can also get the domain name for a load balancer by using the same methods that you used to get the hosted zone ID for the load balancer.

Note

You can also create an alias resource record set by using the `elb-associate-route53-hosted-zone` CLI command. For more information about the CLI

command, see the *Elastic Load Balancing Quick Reference Card* or the Elastic Load Balancing CLI help.

```
POST /2011-05-05/hostedzone/Z222222222/rrset HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <ChangeBatch>
      <Comment>
      This change batch creates two alias resource record sets, one
      for the zone apex, example.com, and one for www.example.com, that both
      point to example-load-balancer-1111111111.us-east-1.elb.amazonaws.com.
      </Comment>
      <Changes>
         <Change>
            <Action>CREATE</Action>
               <ResourceRecordSet>
                  <Name>example.com.</Name>
                  <Type>A</Type>
                  <AliasTarget>
                     <HostedZoneId>Z1111111111111</HostedZoneId>
                     <DNSName>example-load-balancer-1111111111.us-east-
1.elb.amazonaws.com.</DNSName>
                  </AliasTarget>
               </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
               <ResourceRecordSet>
                  <Name>www.example.com.</Name>
                  <Type>A</Type>
                  <AliasTarget>
                     <HostedZoneId>Z1111111111111</HostedZoneId>
                     <DNSName>example-load-balancer-1111111111.us-east-
1.elb.amazonaws.com.</DNSName>
                  </AliasTarget>
               </ResourceRecordSet>
         </Change>
      </Changes>
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

For more information about changing resource record sets, see Creating, Changing, and Deleting Resource Record Sets (p. 49). For information about listing weighted resource record sets, see GET ListResourceRecordSets in the "Actions on Resource Records Sets" section of the *Amazon Route 53 API Reference*.

After you send your change request, check to ensure that your change was applied to all Route 53 DNS servers by polling for the status of the change. When the change is complete, its status becomes `INSYNC`. For more information about polling for the status of a change request, see Checking the Status of Your Change (p. 56).

How to Create Weighted Alias Resource Record Sets

You can create weighted alias resource record sets using the Route 53 console or the Route 53 API. See the applicable topic:

- Creating Weighted Alias Resource Record Sets Using the Route 53 Console (p. 72)
- Creating Weighted Alias Resource Record Sets Using the Route 53 API (p. 74)

Creating Weighted Alias Resource Record Sets Using the Route 53 Console

To create weighted alias resource record sets using the Route 53 console

1. If you created your Route 53 hosted zone and your Elastic Load Balancing load balancer using the same AWS account, skip to Step 2.

   If you created the hosted zone and the load balancer using different accounts, perform the following steps to get the DNS name for the load balancer:

   a. Sign in to the AWS Management Console using the AWS account that was used to create the load balancer for which you want to create an alias resource record set.

   b. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

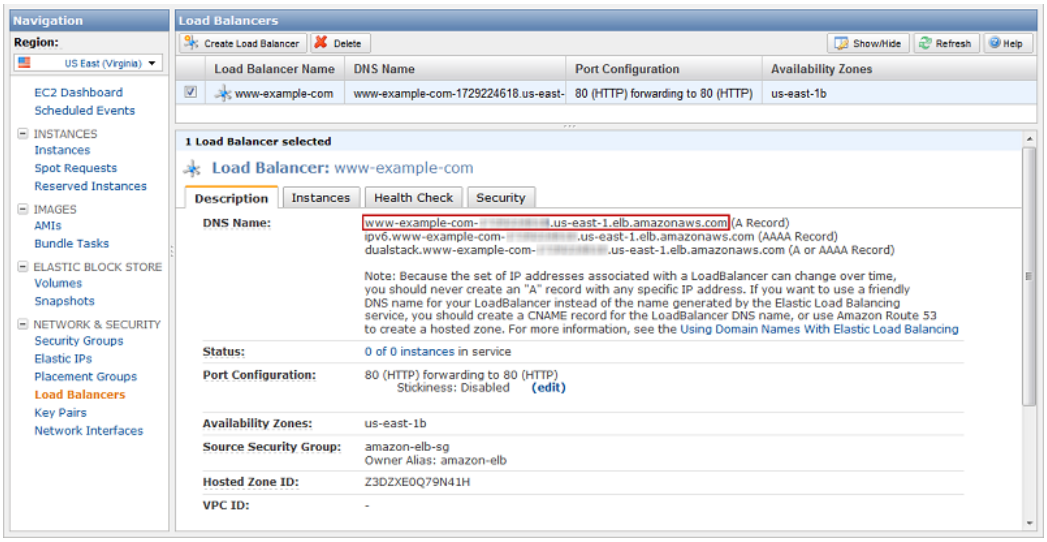
   c. In the **Navigation** pane, click **Load Balancers**.

   d. In the **Load Balancers** pane, select the load balancer for which you want to create a Route 53 alias resource record set.

   e. In the bottom pane, on the **Description** tab, get the A record value for **DNS Name**.

   Important

   Do not use the value for the AAAA record or the dualstack (A or AAAA record) record.

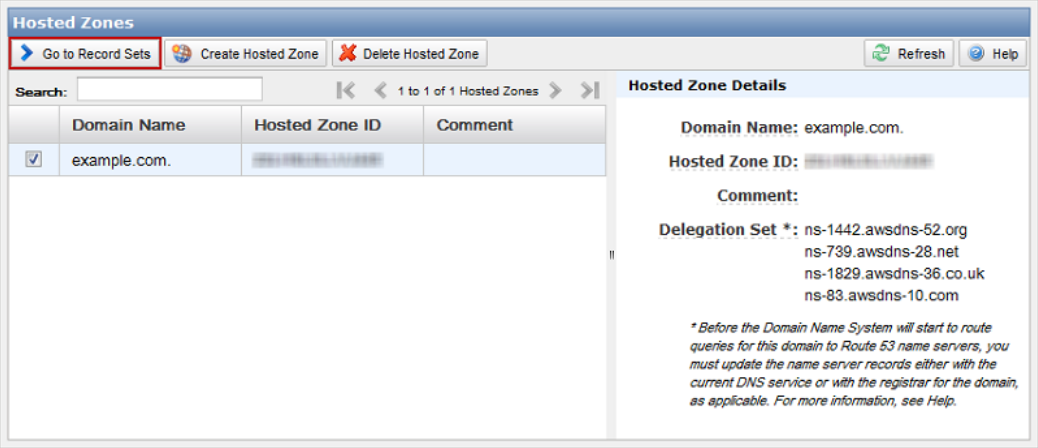


f. Sign out of the AWS Management Console.

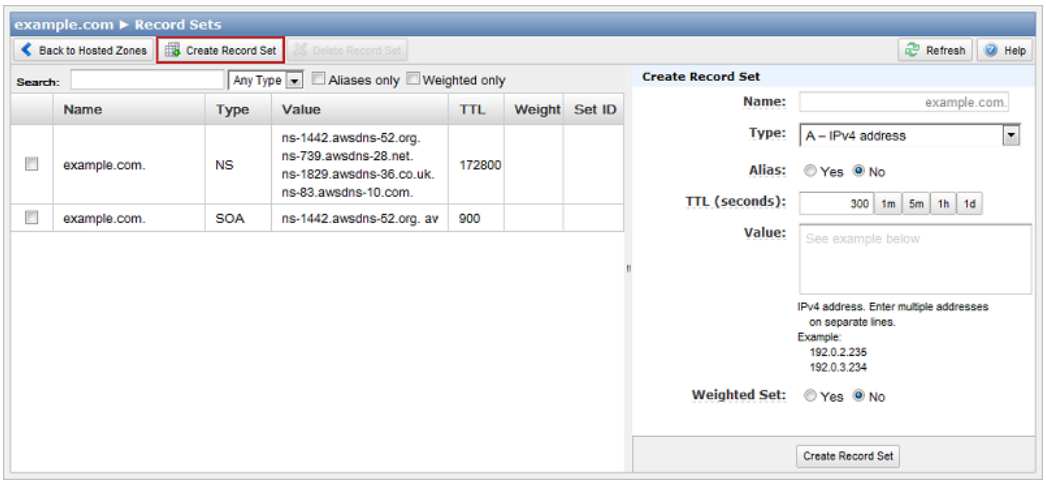
g. Sign in to the AWS Management Console again using the AWS account that was used to create the Route 53 hosted zone.

2. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
3. In the Route 53 console, do one of the following to display the record sets for the hosted zone in which you want to create weighted alias resource record sets:

   - Click the row for the hosted zone, and click **Go to Record Sets**.
   - Double-click the row for the hosted zone.



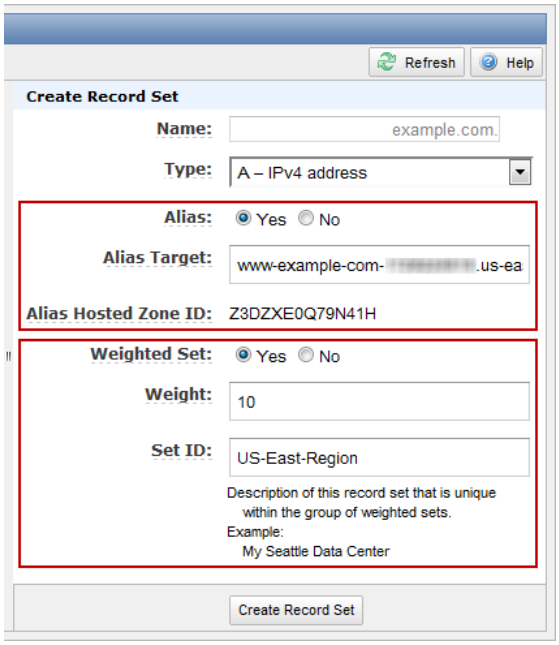
4. Click **Create Record Set**.



5. Enter the applicable values. For information about a field, see the tool tip for the field. Note the following:

   - For **Name**, specify the same value for all of the resource record sets that you want to include in the same group of weighted alias resource record sets.
   - For **Type**, select A or AAAA. Select the same value for all of the resource record sets that you want to include in the same group of weighted resource record sets.

- For **Alias**, click **Yes**.
- For **Alias Target**, do one of the following:
  - If the load balancer and the hosted zone were created using the same AWS account, click the **Alias Target** field, and choose a load balancer from the list. If you have a lot of load balancers, you can type the first few characters of the DNS name to filter the list.
  - If the load balancer and the hosted zone were created using different AWS accounts, enter the value that you got in Step 1 for the DNS name of the load balancer.
- The value of **Alias Hosted Zone ID** appears automatically based on the value that you selected or entered for **Alias Target**.
- For **Weighted Set**, click **Yes**.
- For the value of **Weight**, enter an integer between 0 and 255. To disable routing to a resource, set **Weight** to 0. If you set `Weight` to 0 for all of the resource record sets in the group, traffic is routed to all resources with equal probability. This ensures that you don't accidentally disable routing for a group of weighted alias resource record sets.
- For **Set ID**, enter a value that is unique for the current combination of **Name** and **Type**, for example, **US-East-Region**.



6. Click **Create Record Set**.
7. Repeat steps 4 through 6 for each of the weighted resource record sets that you want to add to the group.

Creating Weighted Alias Resource Record Sets Using the Route 53 API

You can also use the `ChangeResourceRecordSets` action to create weighted alias resource record sets.

In the following example:

- *example.com* is the domain for which you want to create weighted alias resource record sets.
- `SetIdentifier` differentiates the two weighted alias resource record sets from one another. This element is required because the `Name` and `Type` elements have the same values for both resource record sets.
- *Z1111111111111* and *Z3333333333333* are examples of hosted zone IDs for the load balancer specified by the value of `DNSName`. You can get the value of a load balancer's hosted zone ID, `CanonicalHostedZoneNameID`, in the following ways:
  - Use the AWS Management Console. On the EC2 page, click **Load Balancers** in the **Navigation** pane.
  - Use the `DescribeLoadBalancers` action, which is described in the "Actions" chapter of the *Elastic Load Balancing API Reference*.
  - Use the Elastic Load Balancing `elb-describe-lbs` CLI command. For more information about the CLI command, see the *Elastic Load Balancing Quick Reference Card* or the Elastic Load Balancing CLI help.
- *example-load-balancer-2222222222.us-east-1.elb.amazonaws.com* and *example-load-balancer-4444444444.us-east-1.elb.amazonaws.com* are examples of Elastic Load Balancing domains from which Route 53 responds to queries for *example.com*.

  Use the DNS name that was returned by the Elastic Load Balancing `CreateLoadBalancer` action, which is described in the "Actions" chapter of the *Elastic Load Balancing API Reference*. You can also get the domain name for a load balancer by using the same methods that you used to get the hosted zone ID for the load balancer.

```
POST /2011-05-05/hostedzone/Z5555555555/rrset HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <ChangeBatch>
      <Comment>
      This change batch creates two weighted alias resource record sets.
      Route 53 responds to queries for example.com with the first ELB
      domain 3/4ths of the times and the second one 1/4th of the time.
      </Comment>
      <Changes>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>example.com.</Name>
               <Type>A</Type>
               <SetIdentifier>1</SetIdentifier>
               <Weight>3</Weight>
               <AliasTarget>
                  <HostedZoneId>Z1111111111111</HostedZoneId>
                  <DNSName>example-load-balancer-2222222222.us-east-
1.elb.amazonaws.com.</DNSName>
               </AliasTarget>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>example.com.</Name>
               <Type>A</Type>
               <SetIdentifier>2</SetIdentifier>
```

```
               <Weight>1</Weight>
               <AliasTarget>
                  <HostedZoneId>Z3333333333333</HostedZoneId>
                  <DNSName>example-load-balancer-4444444444.us-east-
1.elb.amazonaws.com.</DNSName>
               </AliasTarget>
            </ResourceRecordSet>
         </Change>
      </Changes>
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

Using Aliases with the 2010-10-01 Route 53 API

Aliases were added with the 2011-05-05 version of the Route 53 API. If a zone contains any alias resource record sets, we recommend that you use the 2011-05-05 API. Note the following about using alias resource record sets with the 2010-10-01 API.

- The 2010-10-01 version of the `ChangeResourceRecordSets` action cannot create or delete resource record sets that include `AliasTarget`, `HostedZoneId`, or `DNSName` elements.
- The 2010-10-01 version of the `ListResourceRecordSets` action can list resource record sets that include `AliasTarget`, `HostedZoneId`, and `DNSName` elements, but those elements are not included in the output. Instead, the `Value` element contains a message that says the resource record set includes an unsupported attribute.

Routing Queries to a Website That Is Hosted in an Amazon S3 Bucket

If you want to host a website in an Amazon S3 bucket and you want to use Route 53 as the DNS service for your domain, do the following procedure.

To configure Route 53 and Amazon S3 to route queries to a website that is hosted in an Amazon S3 bucket

1. Create an Amazon S3 bucket and set up your website. For more information, see Setting Up a Website in the *Amazon Simple Storage Service Developer Guide*.

   If you want users to access your website using your domain name (for example, `newproduct.example.com`) instead of the Amazon S3 domain name for your bucket (for example, `myawsbucket.example.com.s3.amazonaws.com`), give the Amazon S3 bucket the name of a subdomain in your domain. In addition, in step 3 of this procedure, create a Route 53 CNAME record that routes traffic from your domain name to the Amazon S3 domain name for your bucket.

   Important

   The DNS protocol does not allow you to create a CNAME record for the top node of a DNS namespace, also known as the zone apex. For example, if you register the DNS name example.com, the zone apex is example.com. You cannot create a CNAME record for example.com, but you can create CNAME records for subdomains like www.example.com, newproduct.example.com, and so on. Therefore, if you want users to access your website using your domain name, you must use a subdomain (www.example.com) instead of the zone apex (example.com).

2. Create a domain or a subdomain that uses Route 53 as the DNS service, or migrate an existing domain or subdomain to Route 53. See the applicable topic:

   - Creating a Domain that Uses Route 53 as the DNS Service (p. 17)
   - Creating a Subdomain that Uses Route 53 without Migrating the Parent Domain (p. 27)
   - Migrating an Existing Domain to Route 53 (p. 22)
   - Migrating a Subdomain to Route 53 without Migrating the Parent Domain (p. 32)

3. In the Route 53 console, create a CNAME record that reroutes traffic for your domain name to the Amazon S3 domain name:

   - For more information from the Amazon S3 perspective, see Customizing Amazon S3 URLs with CNAMEs in the *Amazon Simple Storage Service Developer Guide*.
   - For information about creating a CNAME resource record set in Route 53, see Creating, Changing, and Deleting Resource Record Sets (p. 49).

Controlling User Access with IAM

Topics

- Route 53 ARNs (p. 79)
- Route 53 Actions (p. 80)
- Route 53 Keys (p. 80)
- Example IAM Policies for Route 53 (p. 80)

Amazon Route 53 integrates with AWS Identity and Access Management (IAM) so that you can specify which Route 53 API actions a User can perform on which Route 53 resources. For example, you can create an IAM policy that gives certain Users in your organization permission to update resource record sets of specific hosted zones that your organization owns.

Important

Using Route 53 with IAM doesn't change how you use Route 53. There are no changes to Route 53 actions.

For an example of a policy that covers Route 53 actions, see Example IAM Policies for Route 53 (p. 80).

Route 53 ARNs

You can specify two types of Route 53 resources in an IAM policy: hosted zones and changes. Route 53 Amazon Resource Names (ARNs) have the following general format:

```
arn:aws:route53:::resource/ID
```

`Resource` is either `hostedzone` or `change`, and `ID` is the ID of the hosted zone or the change.

The following are examples of a hosted zone ARN and a change ARN, respectively.

```
arn:aws:route53:::hostedzone/Z148QEXAMPLE8V
arn:aws:route53:::change/C2RDJ5EXAMPLE2
```

You can use wildcards (*) in place of the ID. For example, the following could specify all hosted zone resources owned by the AWS account.

```
arn:aws:route53:::hostedzone/*
```

Route 53 supports all valid values for `namespace` and `relative-id` in an ARN, including federated users.

For more information about ARNs, go to *ARNs* at Identifiers for IAM Entities in *Using IAM*.

Route 53 Actions

Currently, all Route 53 API actions can be referred to in an IAM policy. Each action name must be prefixed with the lowercase string `route53:`. For example: `route53:CreateHostedZone`, `route53:GetChange` or `route53:*` (for all actions).

Most actions can be authorized to act on a specific resource or a set of resources using a wildcard ARN. However, because the `CreateHostedZone` and `ListHostedZones` actions do not act on specific resources, policies for these actions must specify * as the resource. For a list of Route 53 actions, refer to the API action names in the Amazon Route 53 API Reference.

Route 53 Keys

Route 53 implements the following policy keys, but no others. For more information about policy keys, go to *Available Keys* in *Condition* at Element Descriptions in *Using Identity and Access Management*.

AWS-Wide Policy Keys

- `aws:CurrentTime` (for date/time conditions)
- `aws:EpochTime` (the date in epoch or UNIX time, for use with date/time conditions)
- `aws:SecureTransport` (Boolean representing whether the request was sent using SSL)
- `aws:SourceIp` (the requester's IP address, for use with IP address conditions)
- `aws:UserAgent` (information about the requester's client application, for use with string conditions)

If you use `aws:SourceIp`, and the request comes from an Amazon EC2 instance, we evaluate the instance's public IP address to determine if access is allowed.

For services that use only SSL, such as Amazon RDS and Route 53, the `aws:SecureTransport` key has no meaning.

The key names are case insensitive. For example, `aws:CurrentTime` is equivalent to `AWS:currenttime`.

Example IAM Policies for Route 53

This section shows a simple policy for controlling User access to Route 53.

Note

In the future, Route 53 might add new actions that should logically be included in one of the following policies, based on the policy's stated goals.

Example 1: Allow Users read access to all hosted zones

This policy allows the group it is attached to (for example, the AllUsers group) read access to all of the AWS Account's hosted zones.

```
{
   "Statement":[{
         "Effect":"Allow",
         "Action":["route53:GetHostedZone",
            "route53:ListResourceRecordSets"],
         "Resource":"arn:aws:route53::123456789012:change/*"
      }
      {
         "Effect":"Allow",
         "Action":["route53:ListHostedZones"],
         "Resource":"*"
      }
   ]
}
```

Example 2: Allow creation and deletion of hosted zones

This policy allows the group it is attached to (for example, the Managers group) to manage the creation and deletion of hosted zones.

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":["route53:CreateHostedZone"],
         "Resource":"*"
      }
      {
         "Effect":"Allow",
         "Action":["route53:DeleteHostedZone"],
         "Resource":"arn:aws:route53::123456789012:change/*"
      }
      {
         "Effect":"Allow",
         "Action":["route53:GetChange"],
         "Resource":"arn:aws:route53::123456789012:change/*"
      }
   ]
}
```

Example 3: Allow modifications to a specific zone

This policy allows the group it is attached to (for example, a SysAdmins group) to make modifications to a specific zone. It also allows the group to query the status of the change.

```
{
   "Statement":[
      {
         "Effect":"Allow",
         "Action":["route53:ChangeResourceRecordSets"],
         "Resource":"arn:aws:route53::123456789012:hostedzone/Z148QEXAMPLE8V"
      }
      {
         "Effect":"Allow",
         "Action":["route53:GetChange"],
         "Resource":"arn:aws:route53::123456789012:change/*"
      }
   ]
}
```

Making API Requests

Topics

- Endpoints (p. 83)
- REST Requests (p. 84)
- REST Responses (p. 86)
- Authenticating REST Requests (p. 88)

This section describes how to make REST requests to the Route 53 control API, which you use to create and manage your distributions. The various topics acquaint you with the components of requests, the content of responses, and how to authenticate requests.

Endpoints

When you make a REST request you use the following format, where *<hosted zone ID>* is the hosted zone that you are asking to take action on in your request.

```
https://route53.amazonaws.com/2011-05-05/<hosted zone ID>
```

Related Topics

- REST Requests (p. 84)
- The Route 53 Global Network (a list on the AWS website of the Route 53 DNS servers world-wide locations)
- Regions and Endpoints (information about AWS product endpoints and regions in the *Amazon Web Services General Reference*)

REST Requests

Route 53 REST requests are HTTPS requests, as defined by RFC 2616 (for more information, go to http://www.ietf.org/rfc/rfc2616.txt). This section describes the structure of an Route 53 REST request. For detailed descriptions of the actions you can perform, go to the Amazon Route 53 API Reference.

A typical REST action consists of sending a single HTTPS request to Route 53, and waiting for the response. Like any HTTP request, a REST request to Route 53 contains a request method, a URI, request headers, and sometimes a query string or request body. The response contains an HTTP status code, response headers, and sometimes a response body.

Request URI

The request URI always starts with a forward slash and then the version of the Route 53 API you use (for example, 2011-05-05). The remainder of the URI indicates the particular resource you want to act on. For example, following is the URI you use when creating a new hosted zone (for more information, go to POST CreateHostedZone in the *Amazon Route 53 API Reference*).

```
/2011-05-05/hostedzone
```

About the Request Time Stamp

You must provide the time stamp in either the HTTP `Date` header or the AWS `x-amz-date` header (some HTTP client libraries don't let you set the `Date` header). When an `x-amz-date` header is present, the system ignores any `Date` header when authenticating the request.

The time stamp must be within 5 minutes of the AWS system time when the request is received. If it isn't, the request fails with the `RequestExpired` error code. This is to prevent replays of your requests by an adversary.

The date must be specified in one of the following three formats, as specified in the HTTP/1.1 RFC:

- Sun, 06 Nov 1994 08:49:37 GMT (RFC 822, updated by RFC 1123)
- Sunday, 06-Nov-94 08:49:37 GMT (RFC 850, obsoleted by RFC 1036)
- Sun Nov 6 08:49:37 1994 (ANSI C's asctime() format)

Request Body

Many of the Route 53 API actions require you to include XML in the body of the request. The XML conforms to the Route 53 schema.

Example Request

The following example request uses a simple XML statement to create a hosted zone named example.com with the reference identifier, *myUniqueIdentifier*.

```
POST /2011-05-05/hostedzone HTTP/1.1
Host: route53.amazonaws.com
X-Amzn-Authorization: [AWS authentication string]
[Other required headers]

<?xml version="1.0" encoding="UTF-8"?>
<CreateHostedZoneRequest xmlns="https://route53.amazonaws.com/
doc/2011-05-05/">
   <Name>example.com.</Name>
   <CallerReference>myUniqueIdentifier</CallerReference>
   <HostedZoneConfig>
      <Comment>This is my hosted zone.</Comment>
   </HostedZoneConfig>
</CreateHostedZoneRequest>
```

Related Topics

- Authenticating REST Requests (p. 88)
- REST Responses (p. 86)

REST Responses

Route 53 responses are just standard HTTP responses. Some of the Route 53 actions return special information specific to Route 53 in the form of an HTTP header or XML in the body of the response. The specific details are covered in the API reference topic for the particular action.

Request ID

Each response contains a request ID that you can use if you need to troubleshoot a request with Amazon Route 53. The ID is contained in an HTTP header called `x-amz-request-id`. An example of a request ID is `647cd254-e0d1-44a9-af61-1d6d86ea6b77`.

The following example shows a response to a request to create a hosted zone. The `CreatedHostedZoneResponse` element contains information about the hosted zone including an Route 53 identifier, the domain that the hosted zone is associated with, and a reference description and comment. The change request itself is associated with a submittal time, an identifier and a status, shown as `PENDING`. Most importantly, the `CreatedHostedZoneResponse` includes the Route 53 name servers assigned to the hosted zone; this information is contained in the `DelegationSet` element.

Example Response

```
HTTP/1.1 201 Created
x-amz-request-id: request_id

<?xml version="1.0" encoding="UTF-8"?>
<CreateHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/
2011-05-05/">
   <HostedZone>
      <Id>/hostedzone/Z1PA6795UKMFR9</Id>
      <Name>example.com.</Name>
      <CallerReference>myUniqueIdentifier</CallerReference>
      <Config>
         <Comment>This is my first hosted zone.</Comment>
      </Config>
   </HostedZone>
   <ChangeInfo>
      <Id>/change/C1PA6795UKMFR9</Id>
      <Status>PENDING</Status>
      <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
   <DelegationSet>
      <NameServers>
         <NameServer>ns-2048.awsdns-64.com</NameServer>
         <NameServer>ns-2049.awsdns-65.net</NameServer>
         <NameServer>ns-2050.awsdns-66.org</NameServer>
         <NameServer>ns-2051.awsdns-67.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</CreateHostedZoneResponse>
```

Error Responses

If a REST request results in an error, the HTTP response has:

- An XML error document as the response body
- Content-Type header: text/xml
- An appropriate 3xx, 4xx, or 5xx HTTP status code

Following is an example of the XML error document in a REST error response.

```
<ErrorResponse xmlns="https://route53.amazonaws.com/doc/2011-05-05/">
   <Error>
      <Type>Sender</Type>
      <Code>InvalidInput</Code>
      <Message>The input is not valid.</Message>
   </Error>
   <RequestId>410c2a4b-e435-49c9-8382-3770d80d7d4c</RequestId>
</ErrorResponse>
```

Related Topics

- REST Requests (p. 84)
- Authenticating REST Requests (p. 88)

Authenticating REST Requests

Topics

- Overview of the Authentication Process (p. 88)
- The String to Sign (p. 89)
- Calculate the Signature (p. 89)
- The X-Amzn-Authorization Header (p. 89)
- Authentication Errors (p. 89)
- Fetch the Date (p. 90)

Every request you make to the Route 53 control API must be authenticated. AWS and others in the coding community provide tools that automatically sign your requests as required for Route 53.

Overview of the Authentication Process

Authentication is how you prove your identity to the system. You must prove your identity in all your requests to the Route 53 control API. The following sections describe how.

The Route 53 REST API uses a custom HTTP scheme based on a keyed-HMAC (Hash Message Authentication Code) for authentication. The following table describes the basic process for authentication.

Process for Request Authentication

1	You create a string based on specific information in the request. For more information, see The String to Sign (p. 89).
2	You calculate a *signature* using your AWS Secret Access Key, the string from task 1, and a HmacSHA1 or HmacSHA256 algorithm. Informally, we call this process *signing the request*, and we call the output of the HMAC algorithm the signature because it simulates the security properties of a real signature. For instructions on creating the signature, see Calculate the Signature (p. 89).
3	You include the signature in the request and send the request to AWS using HTTPS (HTTP requests are not accepted). For information about where to put the signature in the request, see The X-Amzn-Authorization Header (p. 89).
4	We check your signature. When we receive your request, we fetch the AWS Secret Access Key that you claim to have and use it in the same way you did to compute a signature for the message. We then compare the signature we calculated to the signature you presented in the request. If the two signatures match, we accept and process the request. Otherwise, we reject the request and respond with an error message (for more information, see Authentication Errors (p. 89)).

Note

We also confirm the request time stamp is within 5 minutes of the AWS server time. For more information, see Fetch the Date (p. 90).

The String to Sign

In the first task in the preceding process, you form a string. The string is simply the UTF-8 encoded value of the `Date` header in the request (for example, Thu, 19 Nov 2009 19:37:58 GMT). Your request must include either the `Date` header, the `x-amz-date` header, or both (if both are present, we ignore the `Date` header when authenticating the request). You might decide to include the `x-amz-date` header if your HTTP client doesn't let you set the `Date` header.

The format you use for the header value must be one of the full date formats specified in About the Request Time Stamp (p. 84), for example, `Wed, 05 Apr 2006 21:12:00 GMT`.

Calculate the Signature

Calculating the value to include in the request is a simple procedure.

To calculate the signature

1. Calculate an RFC 2104-compliant HmacSHA1 or HmacSHA256 hash, using the string (see The String to Sign (p. 89)) and your Secret Access Key as the key.
2. Convert the resulting value to base64.

   The result is the signature you include in the request.

The following table shows a string, a fake Secret Access Key, and what the resulting base64 encoded signature would be.

String	`Thu, 14 Aug 2008 17:08:48 GMT`
Secret Access Key	`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`
Base64 encoded signature	`4cP0hCJsdCxTJ1jPXo7+EXAMPLE=`

The X-Amzn-Authorization Header

To pass the signature to AWS, you include it as part of the standard HTTP `X-Amzn-Authorization` header. You include both the signature and your AWS Access Key ID in the header using the following format.

```
AWS3-HTTPS AWSAccessKeyId=MyAccessKey,Algorithm=ALGORITHM,Signature=Base64(
Algorithm((ValueOfDateHeader), SigningKey) )
```

The algorithm can be HmacSHA1 or HmacSHA256.

Authentication Errors

If the signature we create based on your request and Secret Access Key doesn't match the signature you sent in the request, we return the following error.

```
<ErrorResponse xmlns="https://route53.amazonaws.com/doc/2011-05-05/">
   <Error>
      <Type>Sender</Type>
```

```
      <Code>SignatureDoesNotMatch</Code>
      <Message>The request signature we calculated
        does not match the signature you provided.
        Check your AWS Secret Access Key and signing
        method. Consult the service documentation for details.
      </Message>
   </Error>
   <RequestId>a1170c87-d04d-47c9-964f-54e1a4883f4e</RequestId>
</ErrorResponse>
```

Fetch the Date

To avoid replays of your requests, AWS requires the time stamp in the request to be within 5 minutes of the AWS system time. To avoid clock synchronization errors, we recommend you fetch the current date from the Route 53 server and then use that as the time stamp for your request and the string for your signature.

To fetch the date

- Send an unauthenticated HTTPS request for the date resource.

```
GET /date HTTP/1.1
Host: route53.amazonaws.com
```

We return the current server date as the value of the `Date` response header (note that the HTTP status code might or might not be a 200). The date uses the RFC 1123 format (e.g., Thu, 19 Nov 2009 19:37:58 GMT). For more information, go to the RFC 1123 specification.

Example: Using dnscurl.pl

The `dnscurl.pl` command-line tool is a Perl script that lets you call Route 53 API actions from a Linux command line. When you run `dnscurl.pl` and specify the applicable arguments for a given action, `dnscurl.pl` calculates the Route 53 authentication signature, then calls Curl, a popular tool for interacting with HTTP services. Curl passes the arguments to Route 53, and Route 53 processes the request and returns the results.

You can use the following example to create a domain that uses Route 53 as the DNS service. For the steps in which you send requests to Route 53, we show how to use `dnscurl.pl`, which is easier for some users than using the Route 53 REST API.

Note

The examples in this section explain how to send both GET and PUT requests. From these examples and the documentation about using the Route 53 API, you can extrapolate to use `dnscurl.pl` to send any supported request to Route 53.

For information about downloading and installing `dnscurl.pl`, see The dnscurl.pl Command-Line Tool (p. 15).

For information about using the Route 53 console or the Route 53 REST API to create a domain that uses Route 53 as the DNS service, see Creating a Domain that Uses Route 53 as the DNS Service (p. 17).

To get started:

1	Create an AWS account if you haven't already. For more information, see Step 1: Sign Up for an AWS Account (p. 10).
2	Register your domain name. For a list of registrar web sites that you can use to register your domain name, see ICANN.org. After your registrar notifies you that your domain name is successfully registered, you can create a Route 53 hosted zone for the domain. **Caution** You can create hosted zones only for domains that are registered to you. Attempting to create a hosted zone for a domain that is not registered to you constitutes abuse and puts your AWS account at risk of termination.

3	Get `dnscurl.pl` and install the required software. For more information, see The dnscurl.pl Command-Line Tool (p. 15).
4	Create a Route 53 hosted zone. For more information, see Creating a Hosted Zone (p. 92).
5	Add resource record sets (for example, A, MX, and CNAME records) to your hosted zone. For more information, see Creating Resource Record Sets in Your Route 53 Hosted Zone (p. 94).
6	Confirm that your new DNS records have propagated to the Route 53 servers. For more information, see Checking the Status of Your Changes (p. 96).
7	Use your registrar's web site to update the name servers for your domain. For more information, see Updating the Registrar's Name Server Records (p. 98).

Creating a Hosted Zone

Creating a hosted zone involves creating an XML request and using the Route 53 `dnscurl.pl` tool to send your request, as described in the following steps. Text shown in red indicates where you need to modify the values.

To create a hosted zone

1. Create an XML file containing the `CreateHostedZoneRequest` request. In this example, the filename is MyCreateZoneRequest.xml. Your request should look similar to the example.

   Specify the registered domain name for which you're creating the hosted zone in the `Name` element, and specify a description for the domain in the `CallerReference` element. CallerReference can be any string; you might choose to use a string that identifies your project, such as `MyDNSMigration_01`. Specify a different value for `CallerReference` each time you create another hosted zone. The `HostedZoneConfig` value is optional. You can use it to record notes such as the purpose for creating the hosted zone.

   ```
   <CreateHostedZoneRequest xmlns="https://route53.amazonaws.com/doc/2011-05-
   05/">
      <Name>example.com.</Name>
      <CallerReference>a_unique_reference</CallerReference>
      <HostedZoneConfig>
         <Comment>Migrate an existing domain to Amazon Route 53</Comment>
      </HostedZoneConfig>
   </CreateHostedZoneRequest>
   ```

   For more information about the `CreateHostedZone` action, see POST CreateHostedZone in the *Amazon Route 53 API Reference*.

2. Create a file that contains your AWS Access Key ID and your Secret Access Key:

   a. Create a new file. The default filename is .aws-secrets. If you want to use a different filename, edit `dnscurl.pl`.

   b. Go to the AWS Security Credentials page and look up your Access Credentials.

   c. Copy your Access Key ID and Secret Access Key into .aws-secrets.

   d. Save the file in your home directory.

   The .aws-secrets file should look similar to the following example.

```
%awsSecretAccessKeys = (
   "my-aws-account" => {
      id => "AKIAIOSFODNN7EXAMPLE",
      key => "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
   },
);
```

3. To set the permissions on the .aws-secrets file, run the following Linux CLI command:

```
chmod 600 ~/.aws-secrets
```

4. Send the create request. At a Linux command prompt, enter:

```
dnscurl.pl --keyname my-aws-account -- -H "Content-Type: text/xml; char
set=UTF-8" -X POST
--upload-file ./MyCreateZoneRequest.xml https://route53.amazonaws.com/2011-
05-05/hostedzone
```

where:

- `dnscurl.pl` is the name of the command
- `--keyname` *`my-aws-account`* is the `dnscurl.pl` option that specifies the name of the key within the .aws-secrets file that contains your Access Key ID and Secret Access Key.
- `--` indicates the end of the `dnscurl.pl` options.
- `-H "Content-Type: text/xml; charset=UTF-8"` is the Curl option that specifies that you're including an extra HTTP header. In this case, the header specifies the content type and the character set to use for the request.
- `-X POST` is a Curl option that indicates that you're submitting an HTTP POST request method.
- `--upload-file ./`*`MyCreateZoneRequest.xml`* is a Curl option that indicates that the XML document that Curl will POST to Route 53 to create a hosted zone is contained in the file `./`*`MyCreateZoneRequest.xml`*.
- `https://route53.amazonaws.com/2011-05-05/hostedzone` is the URL of the Route 53 endpoint to which you're submitting the create request, along with the contents of `./`*`MyCreateZoneRequest.xml`*.

The response that Route 53 returns includes information similar to the following.

```
<?xml version="1.0">
<CreateHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/2011-05-
05/">
   <HostedZone>
      <Id>/hostedzone/Z1111111111111</Id>
      <Name>example.com.</Name>
      <CallerReference>a_unique_reference</CallerReference>
      <Config>
         <Comment>Migrate an existing domain to Route 53</Comment>
      </Config>
   </HostedZone>
   <ChangeInfo>
      <Id>/change/C24LD0DUV5VOVE</Id>
```

```
      <Status>PENDING</Status>
      <SubmittedAt>2010-12-02T01:34:20.633Z</SubmittedAt>
   </ChangeInfo>
   <DelegationSet>
      <NameServers>
         <NameServer>ns-2048.awsdns-64.com</NameServer>
         <NameServer>ns-2049.awsdns-65.net</NameServer>
         <NameServer>ns-2050.awsdns-66.org</NameServer>
         <NameServer>ns-2051.awsdns-67.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</CreateHostedZoneResponse>
```

This response includes the following information:

- The `Id` element contains the hosted zone ID, which is an alpha-numeric string that begins with "Z".

Tip

Note the hosted zone ID. You need this value to perform any operations on the hosted zone, for example, listing name servers, or adding or deleting resource record sets.

- The `Status` element tells you the status of the request: `PENDING` indicates that the changes in this request have not propagated to all Route 53 DNS servers.
- The `SubmittedAt` element shows when your request was submitted. The `Z` after the time indicates that the time is listed in Coordinated Universal Time (UTC), which is synonymous with Greenwich Mean Time in this context.
- The `NameServers` element lists the Route 53 name servers that are assigned to your hosted zone.

Note

If you don't receive a response to your request, you can resubmit the request with the same caller reference. If necessary, you can use the `ListHostedZones` action to find the caller reference for your request. For information about the `ListHostedZones` action, see ListHostedZones.

Creating Resource Record Sets in Your Route 53 Hosted Zone

Now that you have a hosted zone, you can create resource record sets. For example, if you want to use your domain to access an Amazon S3 bucket and a CloudFront distribution, you will want to create two new CNAME resource record sets.

For more information about the DNS record types that Route 53 supports, see Supported DNS Resource Record Types (p. 4).

To create resource record sets in your hosted zone:

1. Create a new file (`MyCreateRecordsRequest.xml` in this example), and enter the XML code to create two resource record sets:

   - A CNAME record that specifies that myawsbucket.example.com is a pointer to myawsbucket.example.com.s3.amazonaws.com. When a user enters myawsbucket.example.com in a web browser, Route 53 returns myawsbucket.example.com.s3.amazonaws.com, and the DNS resolver restarts the search using myawsbucket.example.com.s3.amazonaws.com.
   - A CNAME record that specifies that cdn.example.com is an alias for d1234abc.cloudfront.net. When a user enters cdn.example.com in a web browser, Route 53 returns d1234abc.cloudfront.net, and DNS restarts the search using d1234abc.cloudfront.net.

   Your request might look like the following. Each new resource record set is defined in a separate `Change` element:

```
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazon
aws.com/doc/2011-05-05/">
   <ChangeBatch>
      <Comment>
      Migrate existing records to Route 53
      </Comment>
      <Changes>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>myawsbucket.example.com.</Name>
               <Type>CNAME</Type>
               <TTL>14400</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                    <Value>myawsbucket.example.com.s3.amazonaws.com</Value>

                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>cdn.example.com.</Name>
               <Type>CNAME</Type>
               <TTL>14400</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>d1234abc.cloudfront.net</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
      </Changes>
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

2. Send the request to create resource record sets. At a Linux command prompt, enter:

```
dnscurl.pl --keyname my-aws-account -- -H "Content-Type: text/xml;
charset=UTF-8" -X POST --upload-file ./MyCreateRecordsRequest.xml
https://route53.amazonaws.com/2011-05-05/hostedzone/HostedZoneID/rrset
```

where:

- `dnscurl.pl` is the name of the command
- `--keyname my-aws-account` is the `dnscurl.pl` option that specifies the name of the key within the .aws-secrets file that contains your Access Key ID and Secret Access Key.
- `--` indicates the end of the `dnscurl.pl` options.
- `-H "Content-Type: text/xml; charset=UTF-8"` is the Curl option that specifies that you're including an extra HTTP header. In this case, the header specifies the content type and the character set to use for the request.
- `-X POST` is a Curl option that indicates that you're submitting an HTTP POST request method.
- `--upload-file ./MyCreateRecordsRequest.xml` is a Curl option that indicates that the XML code that Route 53 requires to create a hosted zone is contained in the file `./MyCreateRecordsRequest.xml`.
- `https://route53.amazonaws.com/2011-05-05/hostedzone/HostedZoneID/rrset` is the Route 53 namespace to which you're submitting the create request along with the contents of `./MyCreateRecordsRequest.xml`. Replace *HostedZoneId* with the alpha-numeric string beginning with "Z" that Route 53 returned in the `Id` element when you created the hosted zone.

Your response will look similar to the following. The status of the requested change (identified in the `Id` element as `/change/CVSYPFWLOWW74`) is `PENDING`.

```
<ChangeResourceRecordSetsResponse xmlns="https://route53.amazon
aws.com/doc/2011-05-05/">
   <ChangeInfo>
      <Id>/change/CVSYPFWLOWW74</Id>
      <Status>PENDING</Status>
      <SubmittedAt>2010-11-27T03:14:40.707Z</SubmittedAt>
   </ChangeInfo>
</ChangeResourceRecordSetsResponse>
```

You have added two CNAME records to your hosted zone.

For information about changing resource record sets, see Creating, Changing, and Deleting Resource Record Sets (p. 49).

Checking the Status of Your Changes

Creating a new hosted zone, and adding or changing resource record sets take time to propagate to the Route 53 DNS servers. After you send the request to create resource record sets, you should check the status of the change request to confirm that your changes have propagated to all Route 53 DNS servers. When propagation is complete, the status changes from `PENDING` to `INSYNC`.

Note

Currently, the only way to verify that changes have propagated is by using the GetChange API action. Changes generally propagate to all Route 53 name servers in a couple of minutes. In rare circumstances, propagation can take up to 30 minutes.

To check the status of your create request:

- Enter the following `dnscurl.pl` command using the value of the `Id` element from the `ChangeResourceRecordSetsResponse` that was returned after you submitted the request to create resource record sets.

In the previous example, the value of the `Id` element was `CVSYPFWLOWW74`. This is a unique value that references a single change request.

```
dnscurl.pl --keyname my-aws-account -- -H "Content-Type: text/xml;
 charset=UTF-8" https://route53.amazonaws.com/2011-05-05/change/CVSYPFWLOWW74
```

where:

- `dnscurl.pl` is the name of the command
- `--keyname my-aws-account` is the `dnscurl.pl` option that specifies the name of the key within the .aws-secrets file that contains your Access Key ID and Secret Access Key.
- `--` indicates the end of the `dnscurl.pl` options.
- `-H "Content-Type: text/xml; charset=UTF-8"` is the Curl option that specifies that you're including an extra HTTP header. In this case, the header specifies the content type and the character set to use for the request.
- `https://route53.amazonaws.com/2011-05-05/change/CVSYPFWLOWW74` is the Route 53 namespace to which you're submitting the request. Replace `CVSYPFWLOWW74` with the value of the `Id` element that was returned after you submitted the request to create resource record sets.

Note

The `-X POST` and `--upload-file` Curl options that appeared in previous `dnscurl.pl` commands do not appear in this command because Curl is performing a GET request. The commands in previous examples performed a POST request, which submits XML data to Route 53.

The response to the status request will look similar to the following:

```
<?xml version="1.0"?>
<GetChangeResponse xmlns="https://route53.amazonaws.com/doc/2011-05-05/">
   <ChangeInfo>
      <Id>/change/CVSYPFWLOWW74</Id>
      <Status>INSYNC</Status>
      <SubmittedAt>2010-11-27T03:14:40.707Z</SubmittedAt>
   </ChangeInfo>
</GetChangeResponse>
```

The example response indicates that the status of the change request, identified by the change ID, is `INSYNC`. If the status is `PENDING`, it indicates your changes have not yet fully propagated to all of the

Route 53 DNS servers. In that case, you should check again after waiting 30 to 60 seconds. After the status changes to `INSYNC`, you know Route 53 has finished propagating your changes.

Updating the Registrar's Name Server Records

Caution

Use the following procedure only if you are creating a domain. If you are migrating an existing domain, or creating or migrating a subdomain, see the corresponding procedure in the applicable section:

- Migrating an Existing Domain to Amazon Route 53
- Creating a Subdomain that Uses Amazon Route 53 without Migrating the Parent Domain
- Migrating a Subdomain to Amazon Route 53 without Migrating the Parent Domain

After your changes to Route 53 resource record sets have propagated to the Route 53 DNS servers and have a status of `INSYNC`, update the name server records with your registrar to refer to the Route 53 name servers. Perform the following procedure.

1. If the registrar has a method to reset TTL settings for their name servers, we recommend that you reset the settings to 900 seconds. This limits the time during which client requests will try to resolve domain names using obsolete name servers. You will need to wait for the duration of the previous TTL for resolvers and clients to stop caching the DNS records with their previous values. A common default setting is 172800 seconds (two days). After the TTL settings expire, you can safely remove records with the previous provider and make modifications only to Route 53.

   Note

   The TTL setting is only an issue if you have queried the domain, so the domain name and IP address are cached with DNS resolvers.

2. Use the `GetHostedZone` action to retrieve the set of name servers for your Route 53 hosted zone. In the `dnscurl.pl` example that follows, the request and response show the values for an example hosted zone. For more information about the `GetHostedZone` action that is called by `dnscurl.pl`, see GetHostedZone in the *Amazon Route 53 API Reference*.
3. Using the method provided by the registrar for the domain, replace the name servers in the registrar's NS records with the four Route 53 name servers that were returned when you submitted the `GetHostedZone` request in the previous step.

   Depending on the TTL settings for the name servers for the parent domain, the propagation of your changes to DNS resolvers can take 48 hours or more. During this period, DNS resolvers may still answer requests with the name servers for the registrar. In addition, client computers may continue to have the previous name servers for the domain in their cache.

To retrieve your name server information:

- Enter the following `dnscurl.pl` command to get your hosted zone ID.

You will need the hosted zone ID you received when you created the hosted zone. In our example, it was Z1111111111111. Your request will look like the following example.

```
dnscurl.pl --keyname my-aws-account -- -H "Content-Type: text/xml; charset=UTF-
8"
https://route53.amazonaws.com/2011-05-05/hostedzone/HostedZoneId
```

where:

- `dnscurl.pl` is the name of the command
- `--keyname` *`my-aws-account`* is the `dnscurl.pl` option that specifies the name of the key within the .aws-secrets file that contains your Access Key ID and Secret Access Key.
- `--` indicates the end of the `dnscurl.pl` options.
- `-H "Content-Type: text/xml; charset=UTF-8"` is the Curl option that specifies that you're including an extra HTTP header. In this case, the header specifies the content type and the character set to use for the request.
- `https://route53.amazonaws.com/2011-05-05/hostedzone/`*`HostedZoneId`* is the Route 53 namespace to which you're submitting the request. Replace *`HostedZoneId`* with the alpha-numeric string beginning with "Z" that Route 53 returned in the `Id` element when you created the hosted zone.

Note

The `-X POST` and `--upload-file` Curl options that appeared in previous `dnscurl.pl` commands do not appear in this command because Curl is performing a GET request. The commands in previous examples performed a POST request, which submits XML data to Route 53.

The response to the confirmation request will look similar to the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/2011-05-
05/hostedzone">
   <HostedZone>
      <Id>/hostedzone/<hosted zone id></Id>
      <Name>example.com.</Name>
      <CallerReference>myUniqueIdentifier</CallerReference>
      <Config>
      <Comment>This is my first hosted zone.</Comment>
     </Config>
   </HostedZone>
   <DelegationSet>
      <NameServers>
         <NameServer>ns-2048.awsdns-64.com</NameServer>
         <NameServer>ns-2049.awsdns-65.net</NameServer>
         <NameServer>ns-2050.awsdns-66.org</NameServer>
         <NameServer>ns-2051.awsdns-67.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</GetHostedZoneResponse>
```

For more information about using Route 53, see Additional Route 53 Resources (p. 100)

Additional Route 53 Resources

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon Route 53 API Reference	The API reference gives the schema location; complete descriptions of the API actions, parameters, and data types; and a list of errors that the service returns.
Route 53 Release Notes	The release notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS::Route53::RecordSet Type in the *AWS CloudFormation User Guide*	Use Route 53 with CloudFormation to create customized DNS names for your CloudFormation stacks.
AWS Developer Tools	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
AWS Management Console	The console allows you to perform the functions of Route 53 without programming.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Route 53.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support (if you are subscribed to this program).
AWS Premium Support Information	The primary web page for information about AWS Premium Support, a one-on-one, fast-response support channel to help you build and run applications on AWS Infrastructure Services.
Route 53 product information	The primary web page for information about Route 53.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse, etc.

Resource	Description
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Third-Party Tools and Libraries

A variety of third-party tools and libraries that work with Route 53 are available.

Topics

- Tools and Libraries (p. 102)
- Graphical User Interfaces (p. 103)

Tools and Libraries

The tools and libraries listed in the following table are available for use with Route 53.

Tool or Library	Description
Amazon Route 53 Authentication Tool for Curl (dnscurl.pl)	Curl is a popular command-line tool for interacting with HTTP services. This Perl script, `dnscurl.pl`, calculates the proper Route 53 authentication signature, then calls Curl with the appropriate arguments. For more information about using `dnscurl.pl`, see Example: Using dnscurl.pl (p. 91).
Amazon Route 53 to BIND Conversion Tool	A BIND zone file describes a DNS zone in a common text-based format. This Perl script converts the XML-formatted text that is returned by the Route 53 `ListResourceRecordSets` API action to BIND zone file format.
Amazon Route 53 Zone Creation Tool	This Perl script generates `CreatedHostedZoneRequest` XML for a given zone origin to create a zone in Route 53.
AmazonRoute53AppsScript (via webos-goodies)	Google spreadsheet management of Route 53
AWS Component for .NET (via SprightlySoft)	SprightlySoft .NET Component for Amazon Web Services with support for REST operations and Route 53

Tool or Library	Description
BIND to Amazon Route 53 Conversion Tool	A BIND zone file describes a DNS zone in a common text-based format. This Perl script converts a BIND zone file to the XML-formatted text that is required by the Route 53 `ChangeResourceRecordSets` API action to add or remove records from Route 53.
Boto API download (via github)	Boto Python interface to Amazon Web Services
cli53 (via github)	Command line interface for Route 53
Dasein Cloud API	Java-based API
dev.SquareCows.com (blog and script download) script download (via github)	Python script to transfer a BIND zone to Route 53
easyRoute53 (easyDNS)	GUI tools, registrar services, and zone transfer services
PHP library for Query-based Route 53 requests	A simple PHP library for interacting with Route 53.
R53.py (via github)	Maintains your own canonical version of your DNS configs under source control, and calculates the minimal changeset required to accomplish a DNS change.
RIAForge	ColdFusion based components for managing DNS using Route 53
RightScripts (via RightScale)	Scripts to configure or update your RightScale server for use with Route 53
RightScale Support Tutorials	RightScale tutorial for domain setup with Route 53
route53d	DNS front-end to Route 53 API (enables incremental zone transfer (IXFR))
Route53Manager (via github)	Web-based interface
Ruby Fog (via github)	The Ruby cloud services library
Slick53	Python interface for Route 53
Valet (via github)	Java API, including a one-way-sync utility for Windows DNS server files
WebService::Amazon::Route53 (via CPAN)	Perl interface to Amazon Route 53 API

Graphical User Interfaces

The following third-party tools provide graphical user interfaces (GUIs) for working with Route 53:

- DNS30
- easyRoute53 (easyDNS)

- Interstate 53
- Nephelai
- R53 Fox
- Ylastic

Document History

This Document History is associated with the 2011-05-05 release of Amazon Route 53. This guide was last updated on 21 February 2012.

The following table describes important changes since the last release of the *Amazon Route 53 Developer Guide*.

Change	Description	Release Date
New Feature	With this release, the Route 53 console in the AWS Management Console lets you create an alias resource record set by choosing an Elastic Load Balancer from a list instead of manually entering the hosted zone ID and the DNS name of the load balancer. New functionality is documented in the *Amazon Route 53 Developer Guide*.	In this release
New Feature	With this release, you can use the Route 53 console in the AWS Management Console to create and delete hosted zones, and to create, change, and delete resource record sets. New functionality is documented throughout the *Amazon Route 53 Developer Guide*, as applicable.	16 November 2011
Updated Documentation	The *Amazon Route 53 Getting Started Guide* was merged into the *Amazon Route 53 Developer Guide*, and the *Developer Guide* was reorganized to enhance usability.	18 October 2011
New Feature	This release of Amazon Route 53 introduces alias resource record sets, which allow you to create zone apex aliases; weighted resource record sets; a new API (2011-05-05); and a service-level agreement. In addition, after six months in beta, Route 53 is now generally available. For more information, see the Amazon Route 53 product page or the following topics in the *Amazon Route 53 Developer Guide*: Creating Alias Resource Record Sets for Elastic Load Balancing (p. 66) and How to Create Weighted Alias Resource Record Sets (p. 72).	24 May 2011
Initial Release	This is the first release of *Amazon Route 53 Developer Guide*.	05 December 2010