Amazon Route 53

Developer Guide

API Version 2010-10-01

Amazon Route 53: Developer Guide

Table of Contents

Welcome

This is the *Amazon Route 53 Developer Guide*. This guide introduces and explains how to use the Amazon Route 53 DNS service.

Amazon Route 53 has a simple API you can use to create a new DNS service or to migrate your existing DNS service. With Amazon Route 53, you can customize your DNS service and maintain all your web service resources under one provider.

How Do I...?

How Do I?	Relevant Resource
Get Started	Amazon Route 53 Getting Started Guide
Understand if Amazon Route 53 is right for my use case	Amazon Route 53
Migrate my DNS service	Migrating Your Existing DNS (p. 12)
Create a new DNS service	Creating a New Virtual DNS Service (p. 5)
Get information about the Amazon Route 53 API	Amazon Route 53 API Reference

Introduction

Topics

Amazon 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 Amazon Route 53 to route users to your infrastructure outside of AWS.

Amazon Route 53 automatically routes queries to the nearest DNS server in a global network of DNS servers, resulting in low latency. It is an authoritative DNS service, meaning it translates friendly domains names like www.example.com into IP addresses like 192.0.2.1.

You can manage your DNS records through the Amazon Route 53 API, or set account-level user and access management through the Identity and Access Management (IAM) API. Like other AWS products, there are no contracts or minimum commitments for using Amazon Route 53—you pay only for the domains you configure and the number of queries that the service answers. For more information about Amazon Route 53 pricing, see Amazon Route 53 Pricing.

About Hosted Zones

A hosted zone is a collection of resource record sets hosted by Amazon Route 53. Like a traditional DNS zone file, a hosted zone represents a collection of records 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 amazon.com hosted zone can contain records named www.amazon.com and www.aws.amazon.com, but cannot contain a record named www.amazon.ca.

You can use the Amazon Route 53 API to create, list, modify, and delete hosted zones and their resource records. The following table describes the actions you can perform on an Amazon 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 New Virtual DNS Service (p. 5). For API descriptions, go to POST CreateHostedZone.
Get your hosted zone's name server information	For how-to information, see Retrieving Your Name Servers (p. 20). For API descriptions, go to GET GetHostedZone.
Delete a hosted zone	For how-to information, see Deleting a Hosted Zone (p. 25). For API descriptions, go to DELETE DeleteHostedZone.
List your hosted zones	For how-to information, see Listing Your Hosted Zones (p. 22). For API descriptions, go to GET ListHostedZones.

Limits on Entity Counts

Amazon Route 53 entities are subject to the following limitations.

Entity	Limitation
Hosted zones	100 per AWS account.
Resource record sets	10,000 per hosted zone.
Records	100 per resource record set.
Resource records in a `ChangeResourceRecordSets` request.	1000, calculated over all `ResourceRecordSets` in the request, and including both `CREATE` and `DELETE` actions.
Characters in all the `ResourceRecord` RDATA fields. RDATA is set in the `Value` element of the `ChangeResourceRecordSet` request.	32,000, calculated over all `ResourceRecords` in all `ResourceRecordSets` in the request, and including both `CREATE` and `DELETE` actions.

DNS Constraints and Behaviors

DNS messaging is subject to factors that affect how you create and use hosted zones and resource records. 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 Amazon Route 53, Amazon Route 53 permits responses up to 4096 bytes over UDP, without truncation.

Authoritative Section Processing

For successful queries, Amazon Route 53 appends name server (NS) type resource records for the relevant hosted zone to the Authority section of the DNS response. For names that are not found

(NXDOMAIN responses), Amazon Route 53 appends the start of authority (SOA) type resource record (as defined in RFC 1035) for the relevant hosted zone to the Authority section of the DNS response.

Additional Section Processing

Amazon Route 53 appends resource records to the Additional section. If the records are known and appropriate, the service appends A or AAAA type resource records for any target of an MX, CNAME, NS, or SRV record cited in the Answer section. For more information about these formats, see Appendix A: Domain Name and Resource Record Formats (p. 41).

AWS Identity and Access Management

Amazon 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 Amazon Route 53 to control which users in your AWS Account can create a new DNS service or change your resource records.

For general information about IAM, go to:

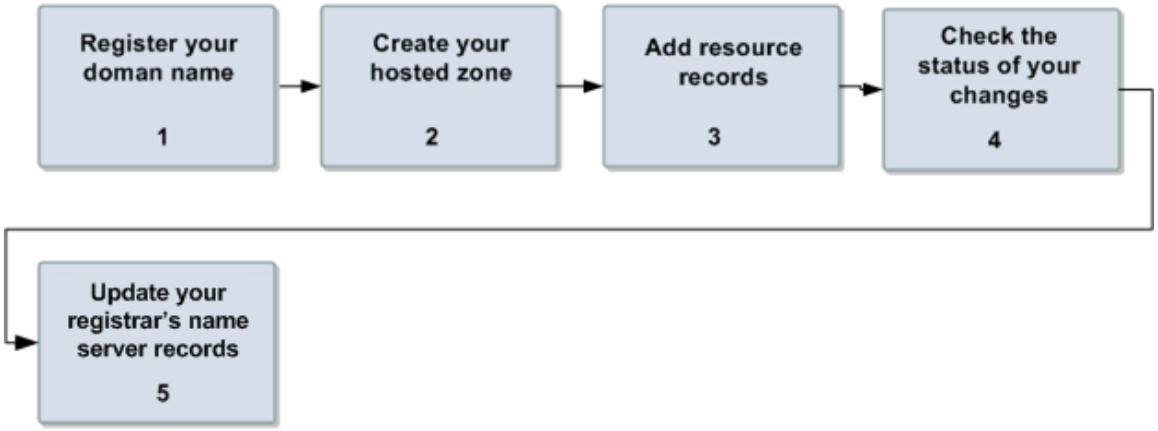
- Identity and Access Management (IAM)
- Identity and Access Management Getting Started Guide
- Using AWS Identity and Access Management

For specific information about how you can control User access to Amazon Route 53, go to Integrating with Other AWS Products in *Using AWS Identity and Access Management*.

Creating a New Virtual DNS Service

You can use Amazon Route 53 to create a virtual DNS service for any registered domain name. This section explains how to create your own DNS service. To learn how to migrate an existing DNS service to Amazon Route 53, see Migrating Your Existing DNS (p. 12).

The following figure and table describe the basic process to use to create your Amazon Route 53 DNS service. These steps are explained in this chapter.



Process for Service Creation

1	You register your domain name (see Register Your Domain Name (p. 6)).
2	You create an Amazon Route 53 hosted zone (see Create Your Hosted Zone (p. 6)).
3	You add resource records to your hosted zone (see Add Resource Records (p. 7)).
4	You check for the status of your changes (see Check the Status of Your Changes (p. 9)).
5	You update your registrar's name server records (see Update the Registrar's Name Servers (p. 10)).

Register Your Domain Name

Amazon Route 53 requires you to use a registered domain name. Before creating your Amazon Route 53 DNS service, make sure that you have registered the domain name 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 your Amazon Route 53 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.

Create Your Hosted Zone

Your DNS service is based on information that resides in a *hosted zone*. To create an Amazon Route 53 hosted zone, you use the `CreateHostedZone` action to send a POST request to the `2010-10-01/hostedzone` resource.

Example Request

```
POST /2010-10-01/hostedzone HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<CreateHostedZoneRequest xmlns="https://route53.amazonaws.com/
doc/2010-10-01/">
   <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/
2010-10-01/">
   <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-4.awsdns-00.com.</NameServer>
         <NameServer>ns-748.awsdns-29.net.</NameServer>
         <NameServer>ns-1394.awsdns-46.org.</NameServer>
         <NameServer>ns-1675.awsdns-17.co.uk.</NameServer>
      </NameServers>
   </DelegationSet>
</CreateHostedZoneResponse>
```

For a detailed description of the `CreateHostedZone` action, see CreateHostedZone in the Amazon Route 53 API Reference.

Add Resource Records

By default, Amazon Route 53 creates a set of standard DNS resource records for every hosted zone that you create. These include a start of authority (SOA) record and four name server (NS) records.

To update the records, you define `ChangeBatch` elements in the `ChangeResourceRecordSets` action. You use `ChangeBatch` to specify a group of changes in a single request. All changes in a single change batch will be completed, or none will be completed. Send the `POST ChangeResourceRecordSetsRequest` to the `/2010-10-01/hostedzone/<hosted zone ID>/rrset` resource.

The following example shows a request and response to update the resource record sets using the `ChangeBatch` element. In this example, the change batch creates *A records* for www.example.com and mail.example.com, and creates a *CNAME record* that points from home.example.com to www.example.com.

Example Request

```
POST /2010-10-01/hostedzone/<hosted zone ID>/rrset HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/
doc/2010-10-01/">
   <ChangeBatch>
      <Comment>
      Adding A records for www.example.com and mail.example.com
      and a CNAME record for home.example.com that points to www.example.com.
      </Comment>
      <Changes>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>www.example.com.</Name>
               <Type>A</Type>
               <TTL>900</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.1</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>mail.example.com.</Name>
               <Type>A</Type>
               <TTL>900</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.2</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>home.example.com.</Name>
               <Type>CNAME</Type>
               <TTL>300</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>www.example.com.</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
      </Changes>
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsResponse xmlns="https://route53.amazonaws.com/doc/2010-
10-01/">
   <ChangeInfo>
      <Id>/change/C2682N5HXP0BZ4</Id>
      <Status>PENDING</Status>
      <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</ChangeResourceRecordSetsResponse>
```

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

Check the Status of Your Changes

The changes that you make to create a new hosted zone and to modify resource records take time to populate the Amazon Route 53 DNS servers. The process of populating the servers is known as *propagation*.

To find out whether a requested change was propagated to the Amazon Route 53 DNS servers, poll the servers for the status of the change. You can use the `GetChange` action to poll for the status of `CreateHostedZone` or `ChangeResourceRecordSets` requests. (For detailed information about the `GetChange` API, see GET GetChange in the Amazon Route 53 API Reference.)

A new change request always has a status of `PENDING`, indicating that the change is not complete. After a change is propagated over all Amazon Route 53 DNS servers, the status of the change becomes `INSYNC`. After you make a change request, you must wait until your changes have a status of `INSYNC` before you can proceed to the next task.

To poll for the status of your change, use the `GetChange` action. The following shows a sample request and response. The `Id` value is supplied by the `ChangeInfo` element contained in the `ChangeResourceRecordSetsResponse` message.

Example Request

```
GET /2010-10-01/change/C2682N5HXP0BZ4
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetChangeResponse xmlns="https://route53.amazonaws.com/doc/2010-10-01/">
  <ChangeInfo>
     <Id>/change/C2682N5HXP0BZ4</Id>
     <Status>INSYNC</Status>
     <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</GetChangeResponse>
```

Update the Registrar's Name Servers

After your changes are propagated and have a status of `INSYNC`, you can update the name servers for your hosted zone with your registrar. To change the delegations, you replace the name servers that the registrar is currently using with the Amazon Route 53 name servers associated with your Amazon Route 53 hosted zone.

Caution

Before you update the name servers of your domain to Route 53, you will want to record and preserve most existing zone records. Also, you should be aware that name servers listed in the zone file can temporarily persist after you update records to Route 53. There can be an interval of about forty eight hours after you update the name servers to Route 53 during which the default name servers, which the provider set to register the domain name, are still being used by DNS resolvers. These settings could be be cached on DNS resolvers or or client computers. For more information, see Update the Registrar's Name Servers (p. 17).

You use the `GetHostedZone` action to retrieve the set of name servers for your Amazon Route 53 hosted zone. The following request and response show the values for an example hosted zone. For more information about the `GetHostedZone` API, see GetHostedZone in the Amazon Route 53 API Reference.

Example Request

```
GET /2010-10-01/hostedzone/<hosted zone id>
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/2010-10-01/">
   <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-01.awsdns-00.com</NameServer>
         <NameServer>ns-500.awsdns-11.net</NameServer>
         <NameServer>ns-1112.awsdns-31.org</NameServer>
         <NameServer>ns-1600.awsdns-27.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</GetHostedZoneResponse>
```

Congratulations, you have created an Amazon Route 53 DNS service!

For each hosted zone you create, Amazon Route 53 creates one start of authority (SOA) record and four name server (NS) records. Don't change these records. There are other records that you will add to the new hosted zone, such as the apex A record that contains the IP address of the home page of the domain. You can add CNAME records, such as the www record or an MX record that identifies a mail server.

The new Route 53 hosted zone contains an SOA record that identifies the base DNS information about the domain:

```
ns-789.awsdns-07.net. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400
```

The elements of the SOA include:

- The host that created the SOA record, for example, `ns-789.awsdns-07.net`.
- The email address of the administrator in a format with the @ symbol replaced by a period, for example `awsdns-hostmaster.amazon.com`.
- 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`. 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`. If this time expires prior to a successful zone transfer, the secondary server will expire its zone file. This means the secondary server will stop answering queries because it considers its data too old to be reliable.
- The negative caching time, which is the time a NAME ERROR = NXDOMAIN result may be cached by any resolver. Negative caching is storing the information that something does not exist. The negative TTL is defined to be the minimum of the SOA record's TTL and the expiry field of the SOA record. The default TTL on Route 53 SOA records is 900 seconds, equivalent to 15 minutes of negative caching.

Route 53 automatically creates four NS records at the apex of your hosted zone. These records are the authoritative name servers for your zone. You should not modify these records or add more name servers into the set. An example of the values of the name server records are:

- ns-123.awsdns-07.net
- ns-1234.awsdns-29.co.uk
- ns-456.awsdns-62.com
- ns-1278.awsdns-33.org

To learn more about working with your DNS service, see the following related topics.

Related Topics

- Retrieving Your Name Servers (p. 20)
- Listing Your Hosted Zones (p. 22)
- Deleting a Hosted Zone (p. 25)
- Listing Resource Record Sets (p. 29)

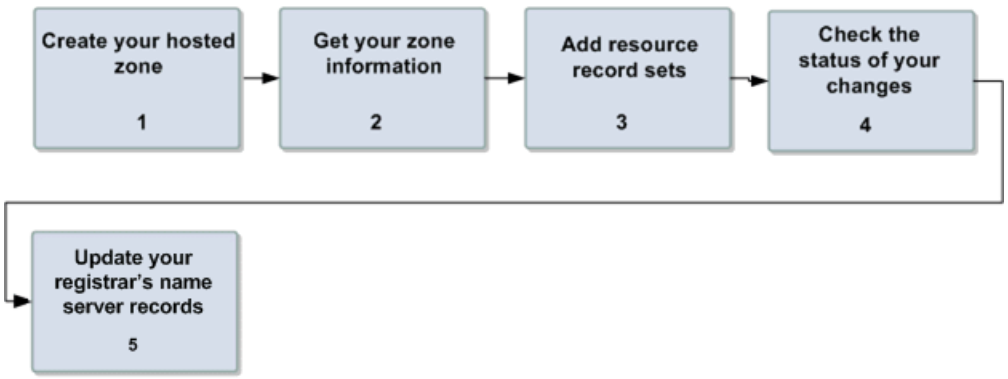
Migrating Your Existing DNS

You can migrate an existing DNS service to Amazon Route 53. To migrate an existing service, use Amazon Route 53 to create a hosted zone for your registered domain. (For information about creating a new virtual DNS service on Amazon Route 53, see Creating a New Virtual DNS Service (p. 5).)

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 figure and table describe the basic process to use to migrate your DNS zone. These steps are explained in more detail in the following sections.



Process for Migration

1	You create an Amazon Route 53 hosted zone (see Create Your Hosted Zone (p. 13)).
2	You get your zone file information from your DNS provider (see Get Your Zone Information (p. 14)).
3	You update your Amazon Route 53 hosted zone with your zone file information (see Adding Resource Records (p. 15)).

4	You poll change status until your Amazon Route 53 hosted zone propagates (see Check the Status of Your Change (p. 17)).
5	Retrieve name server records to give to your registrar (see Update the Registrar's Name Servers (p. 17)).

Create Your Hosted Zone

The first step to migrate your existing DNS service is to create an Amazon Route 53 hosted zone. To create a hosted zone, use the `CreateHostedZone` API to send a POST request to the `2010-10-01/hostedzone` resource. The request body must include an XML document that contains a `CreateHostedZoneRequest` element. For more information on the `CreateHostedZone` API, see CreateHostedZone in the Amazon Route 53 API Reference.

The following example shows a POST request and response.

Example Request

```
POST /2010-10-01/hostedzone HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<CreateHostedZoneRequest xmlns="https://route53.amazonaws.com/
doc/2010-10-01/">
   <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/2010-10-01/">
   <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-01.awsdns-00.com</NameServer>
         <NameServer>ns-500.awsdns-11.net</NameServer>
         <NameServer>ns-1112.awsdns-31.org</NameServer>
         <NameServer>ns-1600.awsdns-27.co.uk</NameServer
      </NameServers>
   </DelegationSet>
</CreateHostedZoneResponse>
```

Get Your Zone Information

Before you can use an Amazon Route 53 hosted zone, you need to update it with information from your existing DNS zone files. How you get this information depends on your DNS service provider.

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.

Whether the zone data for your domain is that of an existing domain or a domain that you have recently created, before moving it to Route 53, you should copy and record your DNS resource record data from your current DNS provider. This is commonly provided to you as a zone file but might also be provided to you in other formats. You will continue to use most of these records (the exceptions are listed in the next paragraph) after you transfer your domain to Route 53.

Do not replace the following records in the new Route 53 hosted zone:

- The SOA record that identifies the base information about the new hosted zone
- The four NS records that identify the name servers for the new hosted zone

As part of the migration to Route 53 you will copy DNS record data from your previous DNS provider to Route 53. This involves copying records specified in your zone file to Route 53 (some exceptions have been noted previously). Common records that need to be migrated include:

- The apex A record that contains the IP address of the home page for the domain
- Mail server MX records

- CNAME records, such as the www record or subdomain records
- Other A records, CNAME records, or other supported DNS record types. (For a list of supported record types, go to Supported DNS Record Types on the AWS website.)

Adding Resource Records

For each hosted zone you create, Amazon Route 53 creates one start of authority (SOA) record and four name server (NS) records. Don't change these records. There could be other records in your zone file you want to add. For example, if your domain uses a CNAME, you need to assign the CNAME in your new resource record set. You might want to use an existing MX record that identifies a mail server.

The new Route 53 hosted zone contains an SOA record that identifies the base DNS information about the domain:

```
ns-789.awsdns-07.net. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400
```

For more information about the SOA record, see Update the Registrar's Name Servers (p. 10).

Route 53 automatically creates 4 NS records at the apex of your hosted zone. These records are the authoritative name servers for your zone and you should not modify these records or add more name servers into the set. An example of the values of the name server records are:

- ns-123.awsdns-07.net
- ns-1234.awsdns-29.co.uk
- ns-456.awsdns-62.com
- ns-1278.awsdns-33.org

To update your resource record sets, use the `ChangeBatch` element to specify a batch change that applies the information from your DNS zone file to your Amazon Route 53 hosted zone. For more information about types of resource records, see Appendix A: Domain Name and Resource Record Formats (p. 41).

The following example shows a request and response to update the resource record sets using the `ChangeBatch` element. In this example, the change batch creates *A records* for www.example.com and mail.example.com, and creates a *CNAME record* that points from home.example.com to www.example.com.

Example Request

```
POST /2010-10-01/hostedzone/<hosted zone ID>/rrset HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/doc/
2010-10-01/">
   <ChangeBatch>
      <Comment>
      Adding A records for www.example.com and mail.example.com and
      a CNAME record for home.example.com that points to www.example.com.
      </Comment>
      <Changes>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>www.example.com.</Name>
               <Type>A</Type>
               <TTL>900</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.1</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>mail.example.com.</Name>
               <Type>A</Type>
               <TTL>900</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>192.0.2.2</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
         <Change>
            <Action>CREATE</Action>
            <ResourceRecordSet>
               <Name>home.example.com.</Name>
               <Type>CNAME</Type>
               <TTL>300</TTL>
               <ResourceRecords>
                  <ResourceRecord>
                     <Value>www.example.com.</Value>
                  </ResourceRecord>
               </ResourceRecords>
            </ResourceRecordSet>
         </Change>
      </Changes>
   </ChangeBatch>
</ChangeResourceRecordSetsRequest>
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsResponse xmlns="https://route53.amazonaws.com/doc/2010-
10-01/">
   <ChangeInfo>
      <Id>/change/C2682N5HXP0BZ4</Id>
      <Status>PENDING</Status>
      <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</ChangeResourceRecordSetsResponse>
```

You can use the `GET GetChange` action to confirm that the update is complete, as described in the next section.

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

Check the Status of Your Change

To check whether a requested change was applied to all Amazon Route 53 DNS servers, you poll for the the status of the change. Changes that are complete have a status of `INSYNC`. Do not proceed until your change status is `INSYNC`.

To poll for the status of a requested change, use the `GetChange` action to send a GET request to the `2010-10-01/change/<change ID>` resource. In the following example, the `Id` value is the `ChangeInfo Id` value from the `ChangeResourceRecordSetsResponse` header.

Example Request

```
GET /2010-10-01/change/C2682N5HXP0BZ4
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetChangeResponse xmlns="https://route53.amazonaws.com/doc/2010-10-01/">
  <ChangeInfo>
     <Id>/change/C2682N5HXP0BZ4</Id>
     <Status>INSYNC</Status>
     <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</GetChangeResponse>
```

Update the Registrar's Name Servers

After your changes are propagated and have a status of `INSYNC`, you will update the name servers your registrar uses for your domain. Replace the name servers that your registrar is currently using with the

Amazon Route 53 name servers associated with your hosted zone. The Amazon Route 53 name server records (NS records) are your *delegation set*.

Caution

There can be an interval of about forty-eight hours after you update name servers to those provided by Route 53 during which DNS resolvers are still using the name servers previously assigned by the registrar. These settings might be be cached on DNS resolvers or or client computers. If a client uses cached NS records to do a domain name lookup, the request can go to the previous name servers and retrieve obsolete DNS records. During this transition period it is important to keep the records synchronized between both the provider you're migrating from and Route 53. The transition period will persist for the TTL interval of the name server records. During this interval the records might be cached by client computers or DNS resolvers.

Before changing the name server settings to those provided by Route 53, if the provider from which you are moving the domain has a method to reset TTL values for the name servers, you can set them to minimum values such as 900 seconds. This limits the time during which client requests will attempt to resolve domain names using obsolete name servers. If you reduce the TTL from 172800 seconds (2 days) to 900 seconds, you still need to wait 2 days to be confident that resolvers or clients are not caching DNS records with the previous TTL. After the TTL settings expire, you can safely remove records with the previous provider and make modifications only to Route 53.

To retrieve the set of name servers for your Amazon Route 53 hosted zone, use the `GetHostedZone` action. The following request and response show the values of an example hosted zone. For information about the `GetHostedZone` API, see GetHostedZone in the Amazon Route 53 API Reference.

Example Request

```
GET /2010-10-01/hostedzone/<hosted zone id>
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetHostedZoneResponse xmlns="https://route53.amazonaws.com/doc/2010-10-01/">
   <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-01.awsdns-00.com</NameServer>
         <NameServer>ns-500.awsdns-11.net</NameServer>
         <NameServer>ns-1112.awsdns-31.org</NameServer>
         <NameServer>ns-1600.awsdns-27.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</GetHostedZoneResponse>
```

Congratulations, you have migrated your DNS service to Amazon Route 53! To learn more about working with your DNS service, see the following related topics.

Related Topics

- Retrieving Your Name Servers (p. 20)
- Listing Your Hosted Zones (p. 22)
- Deleting a Hosted Zone (p. 25)
- Listing Resource Record Sets (p. 29)

Retrieving Your Name Servers

When you create a hosted zone, the `CreateHostedZone` action sends you a response containing information about the hosted zone and the request you sent to create it. The response also includes information crucial to managing DNS for your domains: your name servers names. These names are the name servers that are associated with your Amazon Route 53 DNS service.

The set of name servers is listed in the `DelegationSet` element of your `GetHostedZone` Response. Your domain name registrar requires your name servers information to correctly route your DNS service.

To retrieve the name servers for your hosted domain, send a `GET` request to the `2010-10-01/hostedzone/<hosted zone ID>` resource.

The request includes the Amazon Route 53 identifier associated with the hosted zone, this is the identifier that was provided to you in the `Id` element of the `CreateHostedZone` response. The action `CreateHostedZone` is described in more detail in the Amazon Route 53 API Reference.

Example Request

```
GET /2010-10-01/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/2010-10-01/">
   <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-01.awsdns-00.com</NameServer>
         <NameServer>ns-500.awsdns-11.net</NameServer>
         <NameServer>ns-1112.awsdns-31.org</NameServer>
         <NameServer>ns-1600.awsdns-27.co.uk</NameServer>
      </NameServers>
   </DelegationSet>
</GetHostedZoneResponse>
```

Listing Your Hosted Zones

You can use Amazon Route 53 to retrieve a list of your hosted zones. Because this list might be long, you can use pagination to retrieve a subset of the list.

This section explains how to list your hosted zones and how to paginate your list of hosted zones. For information about how to retrieve a list of resource record sets, see Listing Resource Record Sets (p. 29).

To retrieve a list of your hosted zones, send a `GET ListHostedZones` request to the 2010-10-01/hostedzone resource as follows.

Example Request

```
GET /2010-10-01/hostedzone?marker=Z2EUQ1WTGCTBG2&maxitems=10
```

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

Example Responses

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<ListHostedZonesResponse xmlns="https://route53.amazonaws.com/
doc/2010-10-01/">
   <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 containing the domain name associated with your hosted zone, an Amazon Route 53 identifier of your hosted zone, and your simple description of the hosted zone. By default, the XML file you receive lists a maximum of 100 hosted zones, all nested in a single HostedZones element. If you have no hosted zones, the HostedZones element is empty. For detailed information about the action elements, see the Amazon Route 53 API Reference.

You can also paginate your list into separate XML files that you retrieve 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 a marker parameter was supplied when making the ListHostedZones call. After making your initial request for three items, to retrieve 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 your hosted zones make an initial request to the ListHostedZones action with MaxItems set to the number of hosted zones you would like to retrieve in each page. Until the IsTruncated element in the response you receive is set to false you should 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 /2010-10-01/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/2010-10-01/">
   <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 /2010-10-01/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/2010-10-01/">
   <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

This section explains how to delete an Amazon Route 53 hosted zone.

To delete a hosted zone, send a `DELETE` request to the `2010-10-01/hostedzone/<hosted zone ID>` resource specifying the ID of the hosted zone to delete. The following examples show a request to delete an Amazon Route 53 hosted zone and its response.

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 resource records other than the SOA and NS records, you must delete those resource records before you can delete your hosted zone. If you try to delete a hosted zone that contains additional records, Amazon Route 53 denies your request with a `HostedZoneNotEmpty` error. Deleting records from your hosted zone is described in Making Changes to Your Resource Record Sets (p. 26).

Example Request

```
DELETE /2010-10-01/hostedzone/Z1PA6795UKMFR9
```

Example Response

```
HTTP/1.1 200 OK <?xml version="1.0" encoding="UTF-8"?>
<DeleteHostedZoneResponse xmlns="https://route53.amazonaws.com/
doc/2010-10-01/">
   <ChangeInfo>
      <Id>/change/C1PA6795UKMFR9</Id>
         <Status>PENDING</Status>
         <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</DeleteHostedZoneResponse>
```

To verify the deletion of the hosted zone, use the `GetHostedZone` action to request information about the hosted zone. You can also retrieve an exhaustive listing of your current Amazon Route 53 hosted zones by using the `GET ListedHostedZones` action. For more information about these actions, see the Amazon Route 53 API Reference.

Working with Resource Record Sets

Topics

A resource record set contains information about one or more of the DNS records in your hosted zone. Amazon Route 53 uses this information to resolve DNS queries. You can use Amazon Route 53 API actions on resource records to change your DNS information or to list information about your resource records. For information about the different types of resource records, see Appendix A: Domain Name and Resource Record Formats (p. 41).

Adding Resource Records

By default, Amazon Route 53 creates a set of standard DNS resource records for every hosted zone that you create. These records include a start of authority (SOA) record and four name server (NS) records. Because Amazon Route 53 creates a resource record set for every hosted zone, the procedure you use to add records is a change request. For more information on making changes to your resource records, see the following section Making Changes to Your Resource Record Sets (p. 26).

Making Changes to Your Resource Record Sets

To update your resource record sets, you use the Amazon Route 53 API `ChangeResourceRecordSets` action to create batch changes. These changes apply the information from your change request to your Amazon Route 53 hosted zone. Batch changes are safe because they are transactional changes.

Transactional Changes

When you use the Amazon Route 53 API to make changes to your resource records, your changes are transactions. The transaction model processes modifications as a list of items in a single transaction.

For example, if you use the `ChangeResourceRecordSets` action to delete a resource record set and then replace it, you delete and replace the record set in a single operation. The new record set replaces the existing record set, unless a `DELETE` or `CREATE` action fails. If a single action fails, then the whole batch fails. Because these transactions are contained within a single transaction, they are safe modifications.

Creating a Batch Change

To create a batch change, 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 adding records, then you only need to use `CREATE` actions.

Note

A `ChangeResourceRecordSets` request cannot reference more than 1000 `ResourceRecord` elements. Additionally, the count of characters contained in all `RDATA` fields in the `ResourceRecord` elements of a single `ChangeResourceRecordSets` request cannot exceed 32000. RDATA is set in the `Value` element of the `ChangeResourceRecordSet` request. This limit is calculated across all the `ResourceRecord` elements over all referenced `ResourceRecordSets` and applies to both `CREATE` and `DELETE` actions.

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

Example of a change batch request

This example creates an *A 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 /2010-10-01/hostedzone/Z1PA6795UKMFR9/rrset HTTP/1.1
<?xml version="1.0" encoding="UTF-8"?>
<ChangeResourceRecordSetsRequest xmlns="https://route53.amazonaws.com/
doc/2010-10-01/">
   <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>
```

For information about checking the status of your change request, see Checking the Status of Your Change (p. 29).

Checking the Status of Your Change

After you send your change request, you should check to ensure your change was applied to all Amazon Route 53 DNS servers. To check the status of your change, you poll for the status of the change. When your change is complete, its status becomes `INSYNC`.

To poll for the status of a requested change, use the `GetChange` action to send a GET request to the `2010-10-01/change/<change ID>` resource. In the following example, the `Id` value is the `ChangeInfo` `Id` value from the `ChangeResourceRecordSetsResponse` header.

Example Request

```
GET /2010-10-01/change/C2682N5HXP0BZ4
```

Example Response

```
HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<GetChangeResponse xmlns="https://route53.amazonaws.com/doc/2010-10-01/">
  <ChangeInfo>
     <Id>/change/C2682N5HXP0BZ4</Id>
     <Status>INSYNC</Status>
     <SubmittedAt>2010-09-10T01:36:41.958Z</SubmittedAt>
   </ChangeInfo>
</GetChangeResponse>
```

Listing Resource Record Sets

To retrieve a list of your resource record sets 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 walks you through 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. 31).

To retrieve a list of your resource records

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

   For example:

```
GET /2010-10-01/hostedzone/Z1PA6795UKMFR9/rrset?maxitems=1
```

Your response might look like the following.

```
HTTP/1.1 200 OK
<ListResourceRecordSetsResponse xmlns="https://route53.amazonaws.com/doc/2010-
10-01/">
  <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 /2010-10-01/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/2010-
10-01/">
  <ResourceRecordSets>
    <ResourceRecordSet>
      <Name>example.com.</Name>
      <Type>NS</Type>
      <TTL>172800</TTL>
      <ResourceRecords>
        <ResourceRecord>
          <Value>ns-1421.awsdns-49.org.</Value>
        </ResourceRecord>
        <ResourceRecord>
          <Value>ns-1667.awsdns-16.co.uk.</Value>
        </ResourceRecord>
        <ResourceRecord>
          <Value>ns-888.awsdns-47.net.</Value>
```

```
        </ResourceRecord>
        <ResourceRecord>
          <Value>ns-31.awsdns-03.com.</Value>
        </ResourceRecord>
      </ResourceRecords>
    </ResourceRecordSet>
    <ResourceRecordSet>
      <Name>example.com.</Name>
      <Type>SOA</Type>
      <TTL>900</TTL>
      <ResourceRecords>
        <ResourceRecord>
          <Value>ns-1667.awsdns-16.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 your list requests this way, you use the `Name` and `Type` elements to control the length and ordering of the list of resource record sets. The ways 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 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`.
If you specify `Type` but not `Name`	Amazon Route 53 returns the `InvalidInput` error.

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 Amazon Route 53 DNS servers.

Concurrent Changes

You should avoid retrieving a list that is being changed. After you request list information, any changes that occur in that list won't appear in your results.

The only way to get a consistent multi-page snapshot of all resource record sets in a zone is to stop making changes while pagination is in progress.

Important

Do not use the `ListResourceRecordSets` action to list your resource records set when you are still using the `ChangeResourceRecordSets` action to change them. Your results will not be consistent.

If you must call `ListResourceRecordSets` while making changes, the results from `ListResourceRecordSets` will be consistent within a page. For example, if you are making change calls concurrently with `ListResourceRecordSets`, the result of each change call will either be completely visible in your results or not at all visible. You will not see partial changes, or changes that do not succeed.

The results from `ListResourceRecordSets` are consistent with `ChangeResourceRecordSets`. If a single process makes a call to `ChangeResourceRecordSets` and receives a successful response, the effects of that change will be visible in a subsequent call to `ListResourceRecordSets` by that process.

Making API Requests

Topics

This section describes how to make REST requests to the Amazon 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/2010-10-01/<hosted zone ID>
```

Related Topics

- REST Requests (p. 34)
- The Amazon Route 53 Global Network (a list on the AWS website of the Amazon 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

Amazon 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 Amazon 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 Amazon Route 53, and waiting for the response. Like any HTTP request, a REST request to Amazon 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 Amazon Route 53 API you use (for example, 2010-10-01). 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*).

```
/2010-10-01/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 Amazon Route 53 API actions require you to include XML in the body of the request. The XML conforms to the Amazon 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 /2010-10-01/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/2010-10-01/">
   <Name>example.com.</Name>
   <CallerReference>myUniqueIdentifier</CallerReference>
   <HostedZoneConfig>
      <Comment>This is my hosted zone.</Comment>
   </HostedZoneConfig>
</CreateHostedZoneRequest>
```

Related Topics

- Authenticating REST Requests (p. 38)
- REST Responses (p. 36)

REST Responses

Amazon Route 53 responses are just standard HTTP responses. Some of the Amazon Route 53 actions return special information specific to Amazon 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 Amazon 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 Amazon 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/
2010-10-01/">
   <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-01.awsdns-00.com</NameServer>
         <NameServer>ns-500.awsdns-11.net</NameServer>
         <NameServer>ns-1112.awsdns-31.org</NameServer>
         <NameServer>ns-1600.awsdns-27.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/2010-10-01/">
   <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. 34)
- Authenticating REST Requests (p. 38)

Authenticating REST Requests

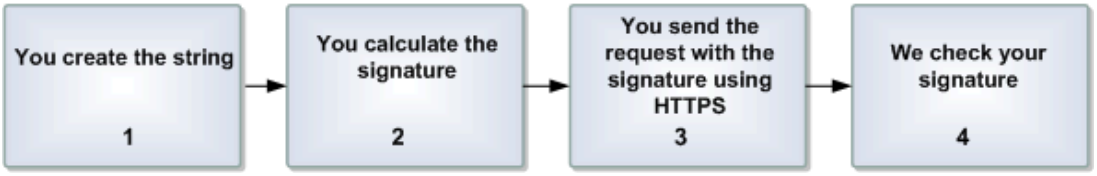
Topics

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

The Amazon Route 53 REST API uses a custom HTTP scheme based on a keyed-HMAC (Hash Message Authentication Code) for authentication. The following figure and table describe 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. 39).
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. 39).
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. 39).
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. 40)).

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. 40).

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. 34). 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. 39)) 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	`/Ml61L9VxlzloZ091/lkqVV5X1/YvaJtI9hW4Wr9`
Base64 encoded signature	`4cP0hCJsdCxTJ1jPXo7+e/YSu0g=`

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/2010-10-01/">
   <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 Amazon 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.

Appendices

Topics

These appendices include additional information about Amazon Route 53, such as related resources, document conventions, and document history.

For definitions of AWS terms, go to the Amazon Web Services Glossary.

Appendix A: Domain Name and Resource Record Formats

Topics

This appendix describes acceptable formats for domain names and for resource records.

Domain Name Format

A domain name consists of a series of labels separated by dots. Each label can be 63 bytes in length at most, and the total size of the domain name cannot exceed 255 bytes. With the exception of the @pseudo-domain-name, Amazon Route 53 supports any legal domain name, including support for arbitrary octet values in domain names. You include arbitrary octet values by using an escaped octal encoding. For example, to encode a label containing a literal dot, you could use `foo\056bar.example.com`.

You can include the following characters literally, without escaping:
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

Even when you specify characters using octal encoding, Amazon DNS considers uppercase and lowercase ASCII characters equivalent and stores them internally as lowercase.

Resource Record Type Value Formats

How you format the resource record `value` element depends on the type of resource record you use. This section describes how to format the `value` element for each record type.

A Format

An A record value element must take the format of an IPv4 address in dotted decimal representation.

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:db8::1</Value>
```

CNAME Format

A CNAME (an NS record value element) is the same format as a domain name.

Note

No other resource record set can have the same name as a CNAME.

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 a domain name.

Example

```
<Value>10 hostname.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 negative time to live (TTL), respectively.

Example

```
<Value>ns-500.awsdns-11.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 TXT format, see TXT Format (p. 43).

SRV Format

An SRV record value element consists of four space-separated fields: decimal numbers representing priority, weight, and port, followed by a domain name.

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 must not exceed 255 characters in length. 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>
```

Appendix B: Amazon 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 Getting Started Guide	The getting started guide provides instructions for using the service for the first time.
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.
Amazon 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 Developer Tools	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon 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.
Amazon Route 53 product information	The primary web page for information about Amazon Route 53.
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse, etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Appendix C: Document Conventions

This section lists the common typographical and symbol use conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical use conventions.

Convention	Description/Example
Call-outs	A call-out is a number in the body text to give you a visual reference. The reference point is for further discussion elsewhere. You can use this resource regularly. 
Code in text	Inline code samples (including XML) and commands are identified with a special font. You can use the command `java -version`.
Code blocks	Blocks of sample code are set apart from the body and marked accordingly. `# ls -l /var/www/html/index.html -rw-rw-r-- 1 root root 1872 Jun 21 09:33 /var/www/html/index.html # date Wed Jun 21 09:33:42 EDT 2006`
Emphasis	Unusual or important words and phrases are marked with a special font. You *must* sign up for an account before you can use the service.
Internal cross references	References to a section in the same document are marked. See Appendix C: Document Conventions (p. 45).
Logical values, constants, and regular expressions, abstracta	A special font is used for expressions that are important to identify, but are not code. If the value is `null`, the returned response will be `false`.
Product and feature names	Named AWS products and features are identified on first use. Create an *Amazon Machine Image* (AMI).
Operations	In-text references to operations. Use the `GetHITResponse` operation.
Parameters	In-text references to parameters. The operation accepts the parameter *`AccountID`*.
Response elements	In-text references to responses. A container for one `CollectionParent` and one or more `CollectionItems`.
Technical publication references	References to other AWS publications. If the reference is hyperlinked, it is also underscored. For detailed conceptual information, see the *Amazon Mechanical Turk Developer Guide*.
User entered values	A special font marks text that the user types. At the password prompt, type **`MyPassword`**.
User interface controls and labels	Denotes named items on the UI for easy identification. On the **File** menu, click **Properties**.
Variables	When you see this style, you must change the value of the content when you copy the text of a sample to a command line. % ec2-register *`<your-s3-bucket>`*/image.manifest See also the following symbol convention.

Symbol Conventions

This section describes the common use of symbols.

Convention	Symbol	Description/Example
Mutually exclusive parameters	(Parentheses | and | vertical | bars)	Within a code description, bar separators denote options from which one must be chosen. `% data = hdfread (start | stride | edge)`

Optional parameters
XML variable text
	[square brackets]	Within a code description, square brackets denote completely optional commands or parameters.
		`% sed [-n, -quiet]`
		Use square brackets in XML examples to differentiate them from tags.
		`<CustomerId>[ID]</CustomerId>`

Variables	<arrow brackets>	Within a code sample, arrow brackets denote a variable that must be replaced with a valid value. `% ec2-register <your-s3-bucket>/image.manifest`

Appendix D: Document History

This Document History is associated with the 2010-10-01 release of Amazon Route 53. This guide was last updated on 29 March 2011.

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

Change	Description	Release Date
Initial Release	This is the first release of *Amazon Route 53 Developer Guide*.	In this release