Getting Started with AWS

Hosting a Static Website

Getting Started with AWS: Hosting a Static Website

Table of Contents

Hosting a Static Website on Amazon Web Services

You can easily and inexpensively use AWS to host a website that uses client-side technologies (such as HTML, CSS, and JavaScript) and does not require server-side technologies (such as PHP and ASP.NET). This type of site is called a *static website*, and is used to display content that does not change frequently.

If you want to deploy a website that requires server-side technologies instead, see *Getting Started with AWS: Hosting a Web App for Linux* or *Getting Started with AWS: Hosting a .NET Web App*.

After you complete this tutorial, you'll know how to do the following:

- **Deploy a static website** – Host your static website using the Amazon Simple Storage Service (Amazon S3) so that it is secure, fast, protected against data loss, and can scale to support enterprise-level traffic. You'll store your website files in Amazon S3 and use Amazon S3 to deliver your content to visitors to your website.
- **Associate your domain name with your website** – Use Amazon Route 53 to tell the Domain Name System (DNS) where to find the resources for your domain, such as your website content in Amazon S3.
- **Speed up your website** – Use Amazon CloudFront to create a content delivery network (CDN) that makes your website content available from data centers around the world, called *edge locations*. Using edge locations improves the speed of your website. This is especially important if your website displays large media files such as high-resolution images, audio, or video.

Static Website Hosting Architectures

Before you create and deploy a static website, you must plan your architecture to ensure that it meets your requirements. The following table shows how Amazon S3, Amazon Route 53, and Amazon CloudFront work together to provide a seamless and cost-effective solution.

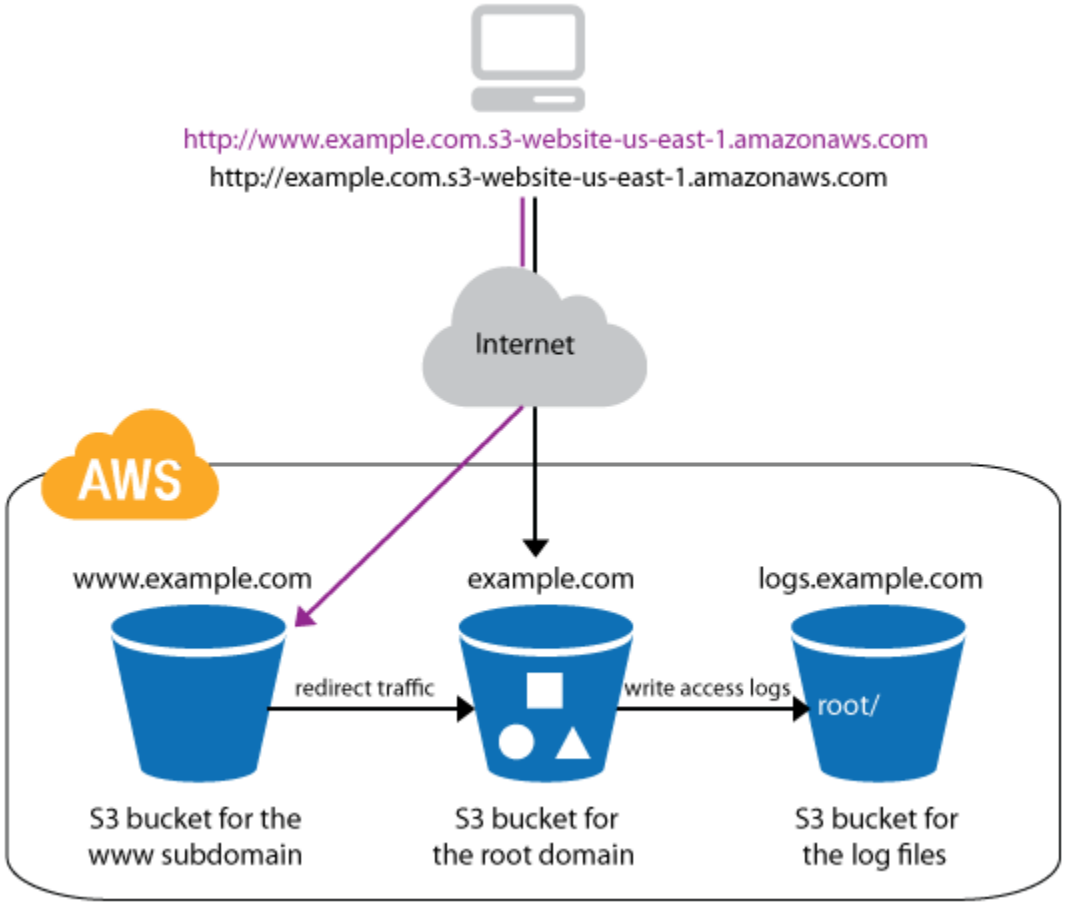
Requirement	Solution
Low-cost, reliable, online storage to host your static website	Amazon S3 is a low-cost, highly reliable web service for hosting static websites.

Requirement	Solution
A reliable and cost-effective way to route customers to your website	Amazon Route 53 maps human-readable domain names to IP addresses and AWS locations.
A way to deliver content with low latency and high data transfer speeds so that visitors to your website don't experience unnecessary delays	CloudFront speeds up the loading of streaming or downloaded static content by caching the content in edge locations. When your customer visits your site, CloudFront delivers the content from the location that is geographically closest to your customer, ensuring the lowest possible latency.

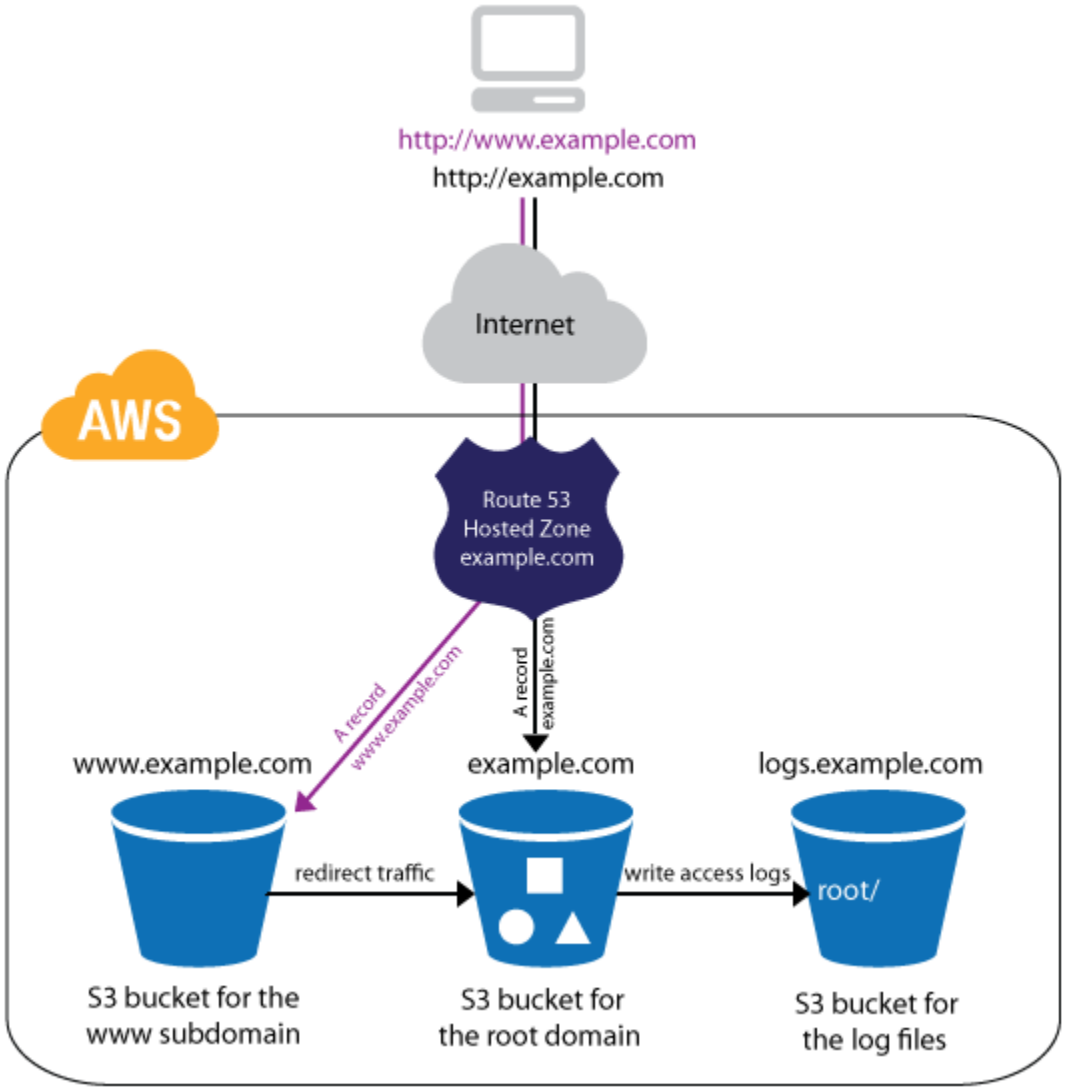
To start hosting a static website on AWS, you'll do the following:

1. Create a location, that is, an Amazon S3 *bucket*, where you will store the files, such as HTML, CSS, JavaScript, and images, for your website.
2. Upload the files to this bucket.
3. Make the files publicly viewable.
4. Configure the bucket to act as a website.

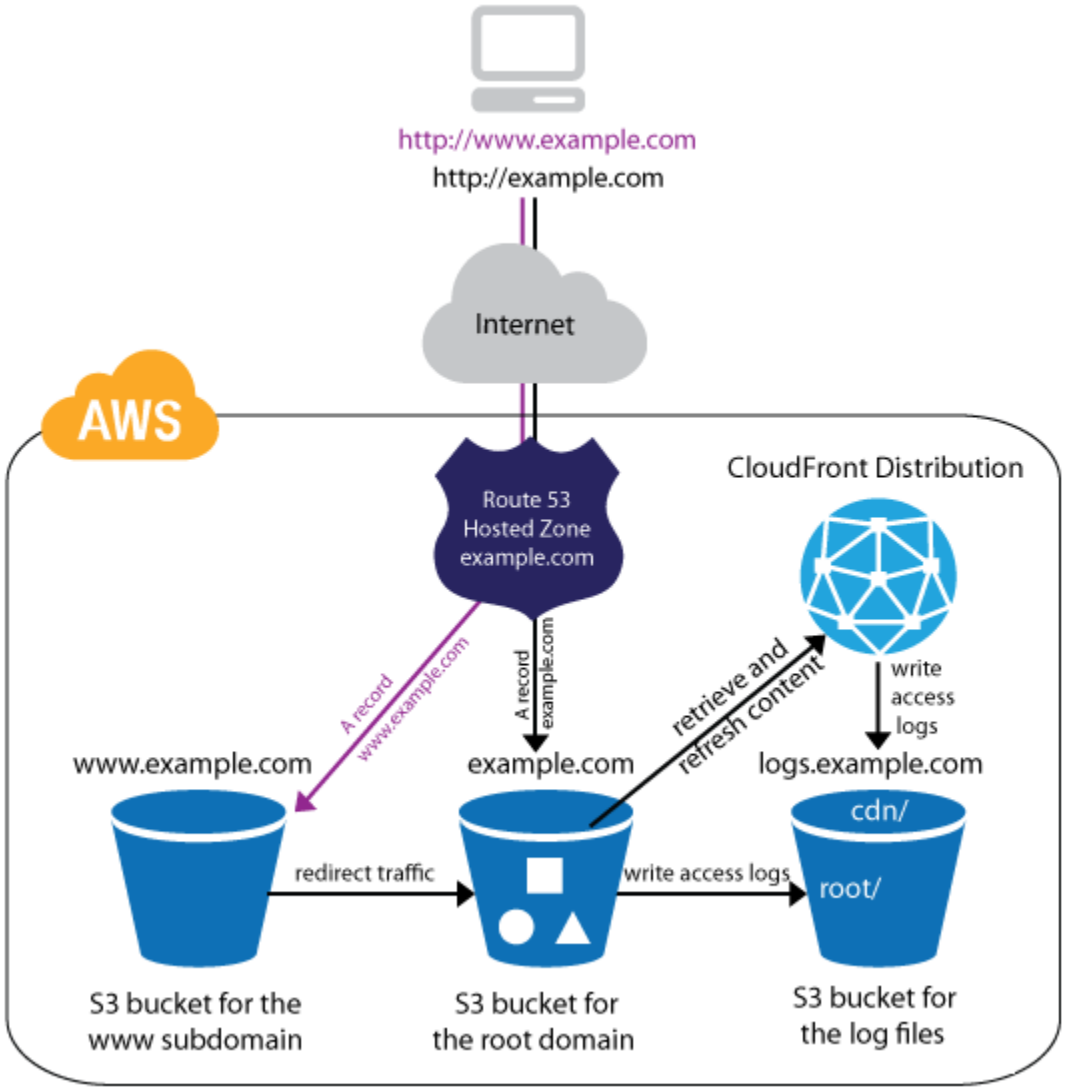
After you complete these first steps, visitors can access your website with a URL in the form http://*example.com*.s3-website-us-east-1.amazonaws.com or http://www.*example.com*.s3-website-us-east-1.amazonaws.com.


http://www.example.com.s3-website-us-east-1.amazonaws.com
http://example.com.s3-website-us-east-1.amazonaws.com
Internet
AWS
www.example.com
example.com
logs.example.com
redirect traffic
write access logs
root/
S3 bucket for the www subdomain
S3 bucket for the root domain
S3 bucket for the log files

Next, you can add a custom domain to your static website by registering a domain name and configuring AWS as your DNS service provider. Visitors can now access your website with a URL in the form http://*example.com* or http://www.*example.com*.



Finally, you can improve performance of your website by distributing it through a CDN. Visitors can still access your website with a URL in the form http://*example.com* or http://www.*example.com*, but now they'll download the files from the edge location closest to them.



Tutorial

This tutorial walks you through the process of hosting a static website on AWS. We'll use the AWS Management Console to access AWS.

1. Create the Buckets for Your Website (p. 10)
2. Register Your Domain Name (p. 14)
3. Configure Your Buckets (p. 15)
4. Deploy Your Website (p. 18)
5. Associate a Domain Name with Your Website (p. 25)

6. Speed Up Your Website (p. 29)
7. Clean Up (p. 33)

Pricing

You can use the AWS Simple Monthly Calculator to estimate what it would cost to host your static website on AWS using Amazon S3, Amazon Route 53, and CloudFront.

Note that if you created your AWS account within the last 12 months, you are eligible for the AWS Free Tier.

For more information about AWS pricing, see Pricing.

Setting Up to Host a Static Website on AWS

Before you start this tutorial, complete the following steps if you haven't already.

To set up

- Sign Up for AWS (p. 7)
- Create an IAM User (p. 7)

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS and you can start using them immediately. You are charged only for the services that you use.

If you created your AWS account less than 12 months ago, you can get started with AWS for free. For more information, see AWS Free Tier.

If you have an AWS account already, skip to the next step. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open http://aws.amazon.com/, and then click **Sign Up**.
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.

Create an IAM User

Services in AWS require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources. The console requires your password. You can create access keys for your AWS account to access the command line interface or API. However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend

that you use AWS Identity and Access Management (IAM) instead in order to better protect your AWS resources from unauthorized access.

Create an IAM user, and then add the user to an IAM group with administrative permissions or and grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console.

To create a group for administrators

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Groups**, and then choose **Create New Group**.
3. For **Group Name**, type a name for your group, such as **`Administrators`**, and then choose **Next Step**.
4. In the list of policies, select the check box next to the **AdministratorAccess** policy. You can use the **Filter** menu and the **Search** box to filter the list of policies.
5. Choose **Next Step**, and then choose **Create Group**.

Your new group is listed under **Group Name**.

To create an IAM user for yourself, add the user to the administrators group, and create a password for the user

1. In the navigation pane, choose **Users**, and then choose **Create New Users**.
2. In box **1**, type a user name. Clear the check box next to **Generate an access key for each user**. Then choose **Create**.
3. In the list of users, choose the name (not the check box) of the user you just created. You can use the **Search** box to search for the user name.
4. Choose the **Groups** tab and then choose **Add User to Groups**.
5. Select the check box next to the administrators group. Then choose **Add to Groups**.
6. Choose the **Security Credentials** tab. Under **Sign-In Credentials**, choose **Manage Password**.
7. Select **Assign a custom password**. Then type a password in the **Password** and **Confirm Password** boxes. When you are finished, choose **Apply**.

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is `1234-5678-9012`, your AWS account ID is `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "*your_user_name* @ *your_aws_account_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM dashboard, click **Customize** and enter an alias, such as your company name. To sign in after you create an account alias, use the following URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

Step 1: Create the Buckets for Your Website

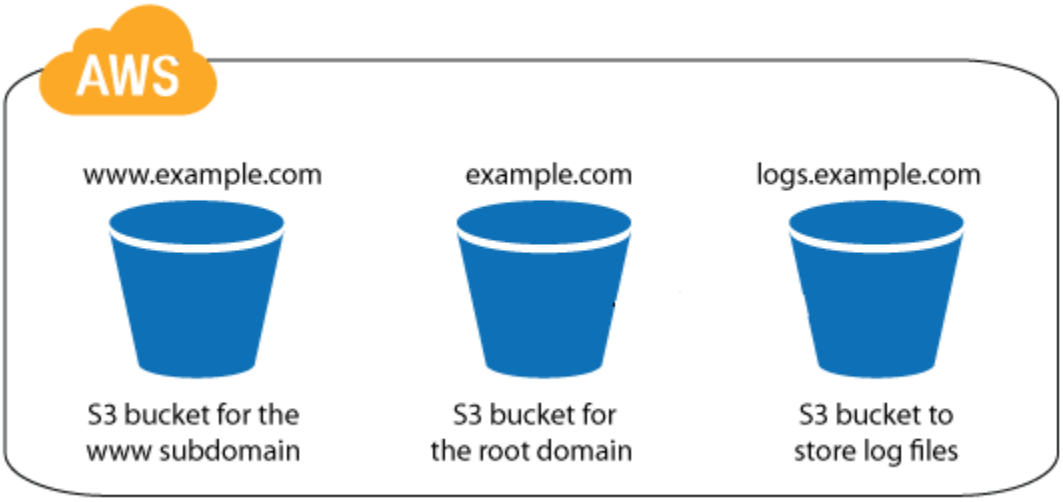
You can use Amazon Simple Storage Service (Amazon S3) to store all the content that makes up your static website, including HTML pages, images, CSS files, videos, and JavaScript files. Each file is stored in Amazon S3 as an *object* in a location called a *bucket*.

Contents

Bucket Requirements

Amazon S3 requires that you give your bucket the same name as your domain. This is so that Amazon S3 can properly resolve the host headers sent by web browsers when a user requests content from your website. Therefore, we recommend that you create your buckets for your website in Amazon S3 before you pay to register your domain name. (If the domain name that you want to use is not available to register, you'll have to delete your bucket and create a new one, because you can't change the name of your bucket after you create it.)

In addition to creating the `example.com` *root domain* bucket, create the `logs.example.com` and `www.example.com` buckets. Be sure to create these buckets in the same AWS region where you created the `example.com` bucket. Amazon S3 stores log information about traffic to your website in the `logs.example.com` bucket. You'll set up the `www.example.com` bucket so that you can redirect traffic to the root domain bucket if a user specifies the www subdomain.



Buckets and Website URLs

When you host a website on Amazon S3, AWS assigns it a URL based on the name of the bucket that you create to store the website files and the region where you created the bucket. For example, if you create a bucket (for example, *example.com*) in the US East (N. Virginia) region, the default URL for your website is as follows:

```
http://example.com.s3-website-us-east-1.amazonaws.com/
```

If this URL is acceptable for your purposes, such as creating a prototype website for a client to review, you can simply use the default URL and skip the steps in this tutorial that are related to registering a custom domain name and associating it with your website.

Creating the Buckets

To create the buckets for your website, use Amazon S3 to complete the following procedure. Note that you must replace "example.com" with the name of your domain.

To create the buckets for your website

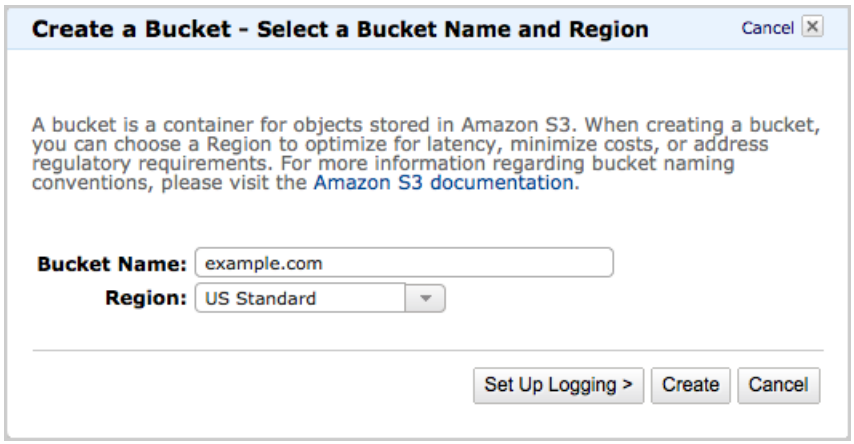
1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Click **Create Bucket**.
3. In the **Create a Bucket** dialog box, do the following:

   a. In the **Bucket Name** box, enter a name for the bucket where you'll upload the files for your website (that is, the bucket for the root domain). You must use the same name that you intend to use for your domain. This name must also be unique across all existing bucket names in Amazon S3. In some AWS regions, there might be additional restrictions on bucket names. For more information, see Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.

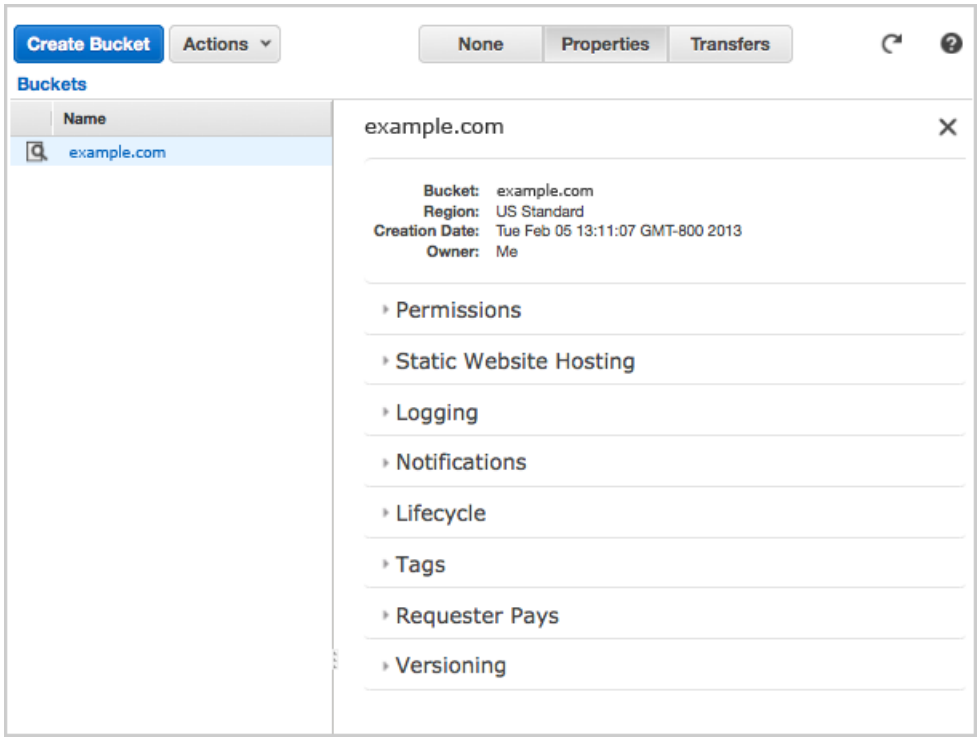
   b. In the **Region** box, select a region. By default, Amazon S3 creates buckets in the US Standard region. To reduce latency, minimize costs, or address regulatory requirements, you can choose

a region that is closer to the users for your website. Objects that you store in a region never leave that region unless you explicitly transfer them to another region.

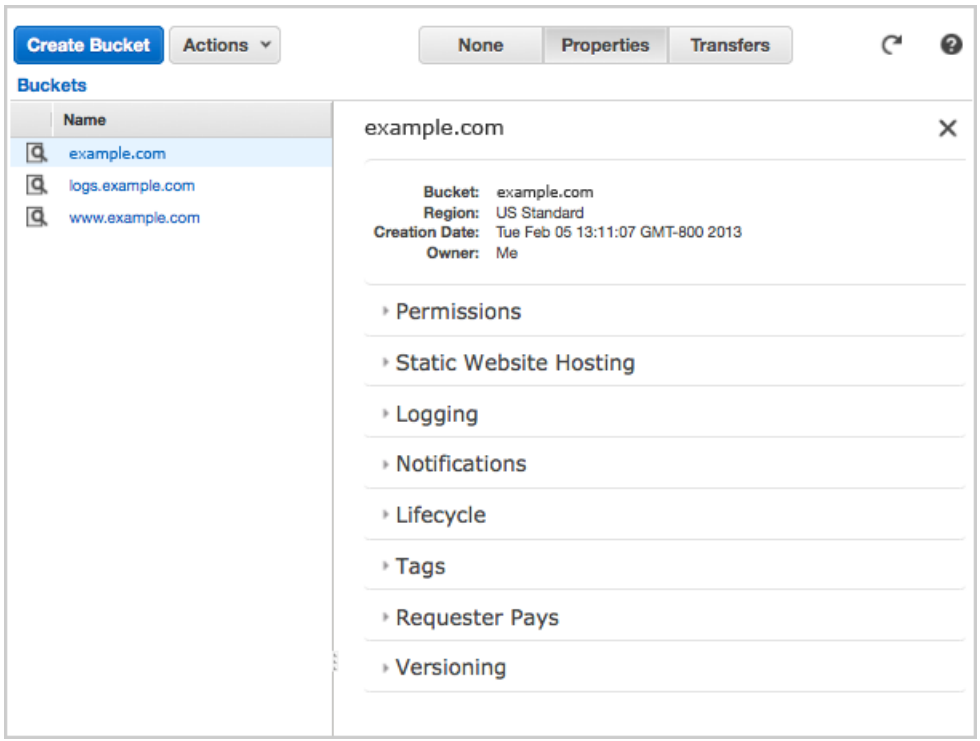
c. Click **Create**.



4. After Amazon S3 creates your bucket, the console displays it in the **Buckets** pane, similar to the following.



5. Register your domain name (p. 14). If the domain name that you want is not available, delete the bucket that you just created (select the bucket, click **Actions**, and then click **Delete**) and then repeat step 3.
6. Repeat step 3 to create two additional subdomain buckets, logs.*example.com* (for the log files) and www.*example.com* (for the www subdomain). When you are finished, the console displays all three buckets, similar to the following.


Create Bucket
Actions
None
Properties
Transfers
Buckets
Name
example.com
logs.example.com
www.example.com
example.com
Bucket: example.com
Region: US Standard
Creation Date: Tue Feb 05 13:11:07 GMT-800 2013
Owner: Me
Permissions
Static Website Hosting
Logging
Notifications
Lifecycle
Tags
Requester Pays
Versioning

Step 2: Register Your Domain Name

If you haven't already done so, register your domain name. The Internet Corporation for Assigned Names and Numbers (ICANN) manages domain names on the Internet. You register a domain name using a *domain name registrar*, an ICANN-accredited organization that manages the registry of domain names. The website for your registrar will provide detailed instructions and pricing information for registering your domain name. For more information, see the following resources:

- To use Amazon Route 53 to register a domain name, see Registering Domain Names Using Amazon Route 53 in the *Amazon Route 53 Developer Guide*.
- For a list of accredited registrars, see the Accredited Registrar Directory.

First, check that the domain name that you used when you created your buckets in Amazon S3, per Step 1: Create the Buckets for Your Website (p. 10), is available with a domain name registrar. If the domain name is not available, you should delete the buckets in Amazon S3 and create new buckets. Be sure to create the buckets in Amazon S3 before you pay to register the domain name, because Amazon S3 requires that you give your bucket the same name as your domain when you host a static website, and you can't change the name of a bucket after you create it.

Step 3: Configure Your Buckets

When you configure your root domain bucket in Amazon S3 as a website, Amazon S3 delivers the files in that bucket to web browsers as if they were hosted on a web server. First, you must add permissions to your root domain bucket so that everyone can view the files for your website. We also recommend that you enable logging to record information about traffic to your website.

To configure the buckets for your website, use Amazon S3 to complete the following tasks.

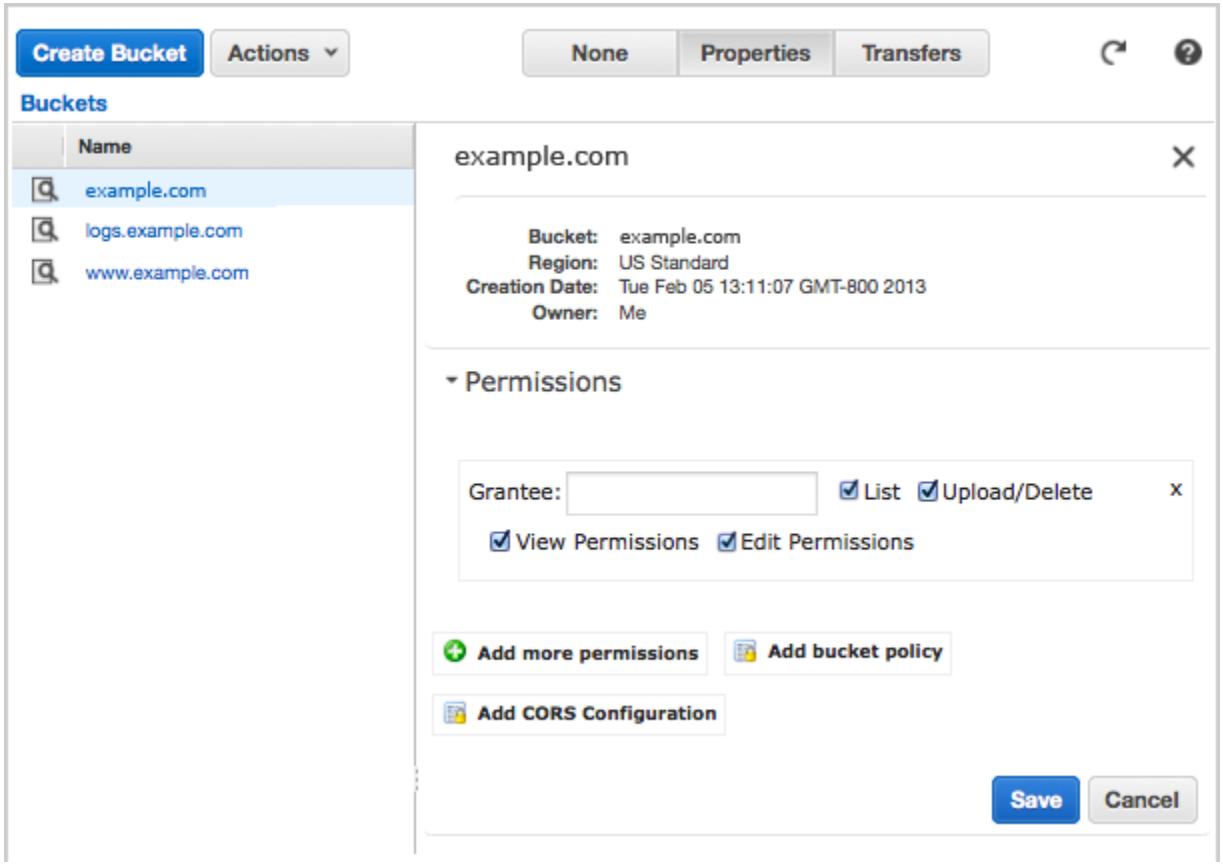
Tasks

Add Permissions

When you first create an Amazon S3 bucket, only you can access the bucket and its contents. This default behavior ensures that you do not accidentally expose your data to other users. The point of a website, however, is to be visited, so we'll apply a policy to the root domain bucket so that anyone can view its contents. For more information, see Using Bucket Policies and User Policies in the *Amazon Simple Storage Service Developer Guide*.

To add permissions for the contents of your root domain bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** pane, select your root domain bucket, click **Properties**, click **Permissions**, and then click **Add bucket policy**.



3. The following policy gives everyone permission to view any file in the *example.com* bucket. Copy the policy and then paste it into the **Bucket Policy Editor**. Replace *example.com* with the name of your bucket, and then click **Save**.

```
{
  "Version":"2012-10-17",
  "Statement": [{
    "Sid": "Allow Public Access to All Objects",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::example.com/*"
  }
 ]
}
```

   For more information, see Using Bucket Polices and User Policies in the *Amazon Simple Storage Service Developer Guide*.

4. Under **Permissions**, click **Save**.

Enable Logging

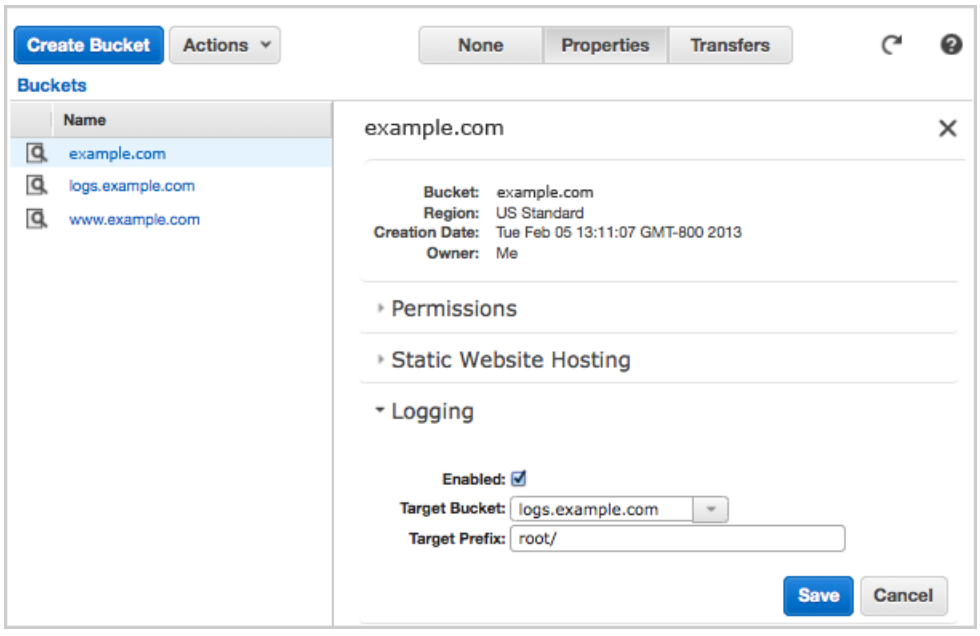
To track the number of visitors accessing your website, you must enable logging for the root domain bucket. Enabling logging is optional. If you don't want to track traffic to your website, you can skip the following procedure.

With logging enabled, you can track information such as data in and out of your bucket and the IP addresses of whoever is accessing your bucket. There is no extra charge for enabling logging on a bucket; however, you will accrue charges to store the resulting log files in the bucket that you specify. (You can delete the log files from this bucket at any time.) Amazon S3 does not assess data transfer charges for log file delivery, but does charge the normal data transfer rate for accessing the log files. For more information, see Server Access Logging in the *Amazon Simple Storage Service Developer Guide*.

To enable logging for your root domain bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** pane, select your root domain bucket, click **Properties**, and then click **Logging**.
3. Complete the **Logging** pane as follows:

   a. Select the **Enabled** check box.
   b. In the **Target Bucket** list, select the bucket that you created for the log files, logs.`example.com`.
   c. In the **Target Prefix** box, enter `root/`. This setting groups the log data files in a folder named `root` in the bucket so that they will be easy for you to locate later.
   d. Click **Save**.



Step 4: Deploy Your Website

Now that you've created and configured your Amazon S3 buckets, you are ready to deploy your website. If you don't already have files for a website, you can just use the simple HTML files we create in this step.

To deploy your static website, use Amazon S3 to complete the following tasks.

Tasks

Create an Index Document and a Custom Error Document

The *index document* is the default page of a website. When you configure your website with a *custom error document*, Amazon S3 returns that error document for HTTP 4xx error codes. Create these files on your computer with the names `index.html` and `error.html` and save them where you can easily find them.

Add the following HTML to `index.html`:

```
<html>
  <body>
    <p>Hello, World!</p>
  </body>
</html>
```

Add the following HTML to `error.html`:

```
<html>
  <body>
    <p>This is an error page.</p>
```

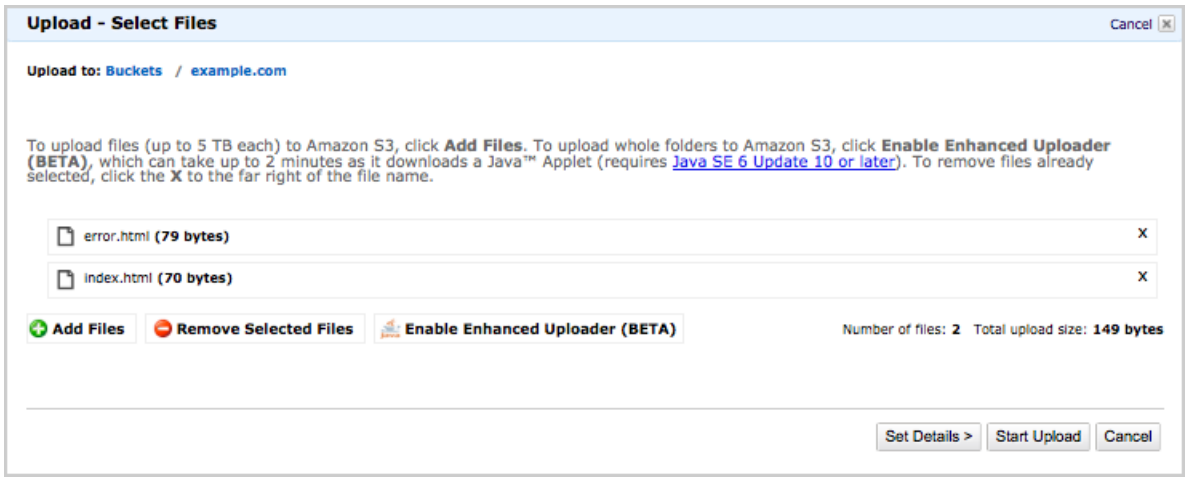
```
  </body>
</html>
```

For more information, see Index Document Support and Custom Error Document Support in the *Amazon Simple Storage Service Developer Guide*.

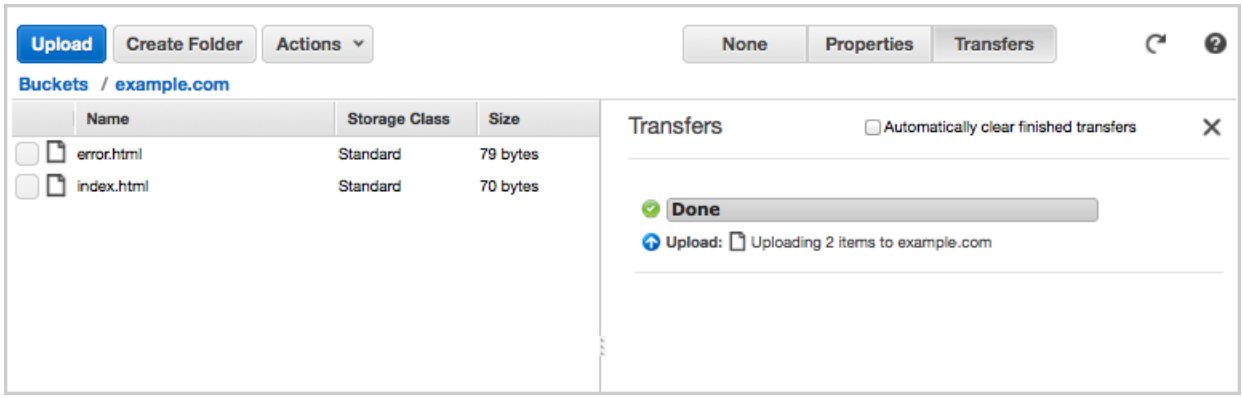
Upload Files to Your Bucket

To upload your website files to your root domain Amazon S3 bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Select your root domain bucket.
3. Click **Actions**, and then click **Upload**.
4. In the **Upload — Select Files** dialog box, do the following:
   a. Click **Add Files**.
   b. In the **File Upload** dialog box, select the `index.html` and `error.html` files that you created, and then click **Open**.
   c. Click **Start Upload**.



5. When your files have finished uploading, they appear as follows.

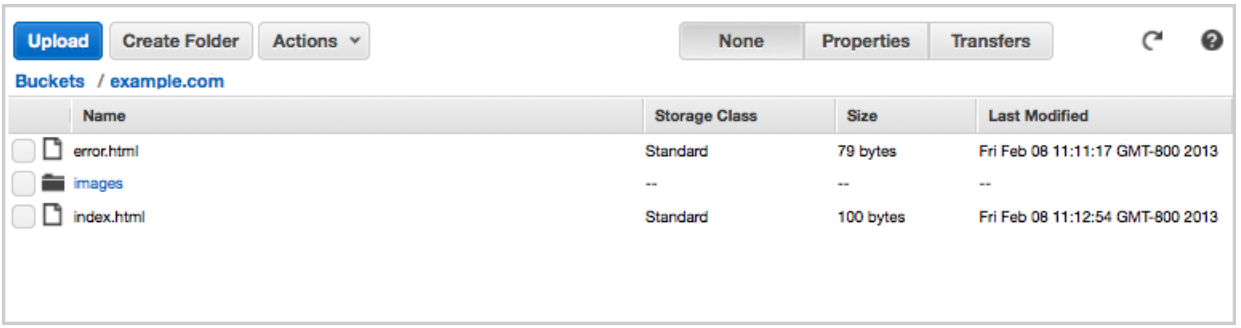


6. (Optional) Upload any other files for your website.

   If your website files have a folder hierarchy on your local computer, such as storing image files in an `images` subfolder, you need to recreate that hierarchy in your buckets. To do so, simply create folders inside the root domain bucket that match your folder hierarchy. For example, consider the case where you have a file `/images/check.gif` referenced in `index.html` as follows.

```
<html>
  <body>
    <p>Hello, World!</p>
    <img src="/images/check.gif">
  </body>
</html>
```

   To create this folder, open your root domain bucket, click **Create Folder**, create the `images` folder, and then upload `check.gif` to the new `images` folder.



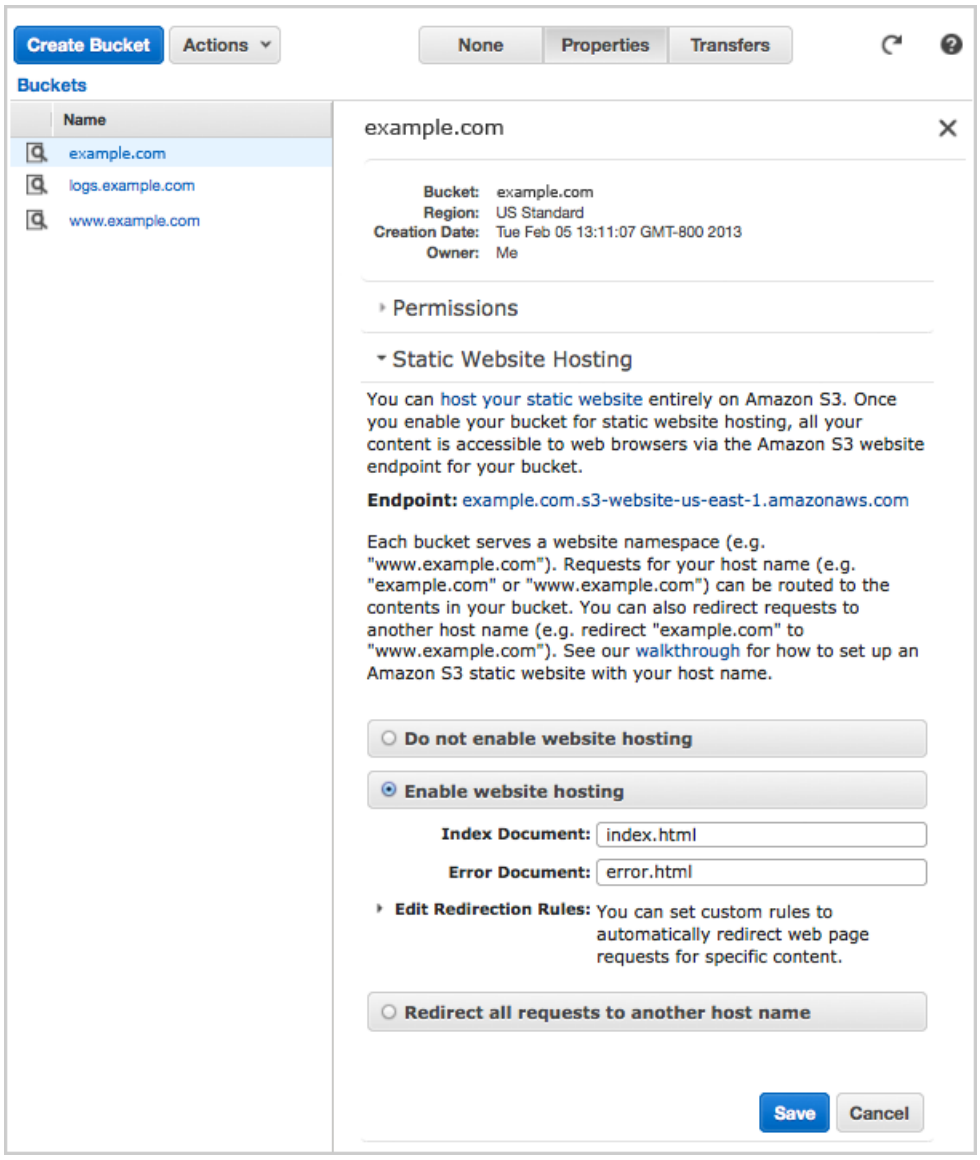
Configure Your Bucket as a Website

After you configure your bucket as a website, Amazon S3 can serve the files in the bucket as if they were hosted on a web server.

To configure your root domain bucket as a website

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** pane, select your root domain bucket, click **Properties**, and then click **Static Website Hosting**.
3. Make a note of the value of **Endpoint**; for example, *`example.com`*`.s3-website-us-east-1.amazonaws.com`. You'll need this value if you decide to set up a CDN.
4. Complete the **Static Website Hosting** pane as follows:

   a. Click **Enable website hosting**.
   b. In the **Index Document** box, enter `index.html`.
   c. In the **Error Document** box, enter `error.html`.
   d. Click **Save**.



Set Up a Redirect

Before you can associate your domain name with your website, you must redirect traffic from the www subdomain bucket to the root domain bucket. Then, Amazon S3 forwards any requests that are sent to the www subdomain bucket to the root domain bucket instead. By redirecting traffic in this way, you can maintain a single version of your website files in Amazon S3 while still supporting both the root and www subdomain versions of your website's address.

To redirect traffic from your www subdomain bucket to your root domain bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Select the subdomain bucket you created, `www.example.com`, click **Properties**, and then click **Static Website Hosting**.
3. Complete the **Static Website Hosting** pane as follows:

a. Click **Redirect all requests to another host name**.
b. In the **Redirect all requests to** box, enter the name of your root domain.
c. Click **Save**.

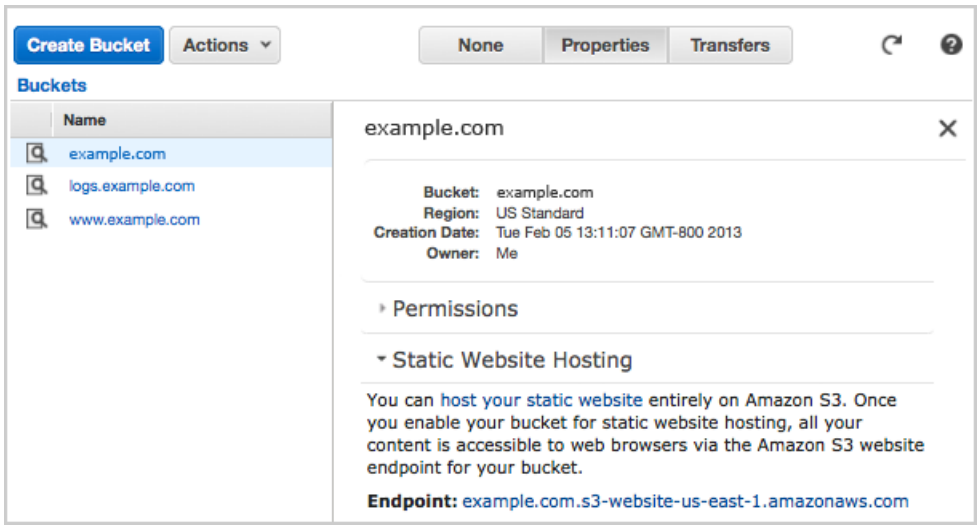
Test Your Website

You can verify that your static website is live on the Internet by using a web browser to navigate to the default URL assigned by Amazon Web Services.

To test your website

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Select your root domain bucket, click **Properties**, and then click **Static Website Hosting**.

   The default URL assigned by AWS is the **Endpoint**. In the following image, this is `example.com.s3-website-us-east-1.amazonaws.com`.



3. Click the endpoint. If your website is correctly deployed, you'll see its home page.

4. (Optional) To verify that the subdomain bucket is properly redirecting visitors, try to access `http://www.example.com.s3-website-us-east-1.amazonaws.com`. If your website is correctly deployed, you are redirected to `http://example.com.s3-website-us-east-1.amazonaws.com`.
5. (Optional) To verify that the error page is working, try to access a page on your new website that doesn't exist, such as

`http://example.com.s3-website-us-east-1.amazonaws.com/bogus.html`. If your website is correctly deployed, you are redirected to your custom error page.

Step 5: Associate a Domain Name with Your Website Using Amazon Route 53

The easiest way for your customers to access your website is through a memorable domain name. In the procedures on this page, replace "example.com" with your domain name.

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. It is designed as an extremely reliable and cost-effective way to route visitors to websites by translating domain names (such as `www.example.com`) into the numeric IP addresses (such as `192.0.2.1`) that computers use to connect to each other. With Amazon Route 53, you pay only for the domains you configure and the number of queries that the service answers. For more information, see Amazon Route 53.

To associate a domain name with your website, use Amazon Route 53 to complete the following tasks.

Tasks

Register a Domain Name

If you haven't done so already, register your domain name (p. 14).

Create a Hosted Zone for Your Domain

A *hosted zone* is a container for the information about how you want to route traffic on the Internet for a domain (such as `example.com`) and its subdomains (such as `www.example.com`).

To create a hosted zone

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. If you are new to Amazon Route 53, you see a welcome page; click **Get Started Now** under **DNS Management**. Otherwise, click **Hosted Zones** in the navigation pane.
3. Click **Create Hosted Zone**.
4. In **Domain Name**, enter your domain name.
5. Click **Create**.

Create Record Sets for Your Domain and Subdomain

Create an alias resource record set that routes queries for your domain name to the Amazon S3 domain name for your bucket.

To configure the alias record set for your root domain

1. On the **Hosted Zones** page, select the hosted zone that you created for your domain.
2. Click **Go to Record Sets**.
3. Click **Create Record Set**.
4. Under **Create Record Set**, do the following:
    a. Leave the default name, which is the root domain.
    b. From **Type**, select **A — IPv4 address**.
    c. In **Alias**, click **Yes**. An alias enables Amazon Route 53 to associate your domain name with the Amazon S3 bucket that you configured as a website endpoint.
    d. Click **Alias Target**. Select your root domain website endpoint from the list (for example, `example.com`). Do not select the www subdomain endpoint (for example, `www.example.com`).
    e. In **Routing Policy**, select **Simple**.
    f. Leave **Evaluate Target Health** set to **No**.
    g. Click **Create**.

Next, create an alias resource record set that routes queries for your www subdomain name to the Amazon S3 domain name for your bucket.

To configure the alias resource record set for your www subdomain

1. On the **Hosted Zones** page, select the hosted zone that you created for your domain.
2. Click **Go to Record Sets**.
3. Click **Create Record Set**.
4. Under **Create Record Set**, do the following:
    a. In **Name**, type `www`. The root domain is already specified for you, and the connecting period (.) appears when you start typing.
    b. From **Type**, select **A — IPv4 address**.
    c. In **Alias**, click **Yes**.
    d. Click **Alias Target**. Select your www subdomain website endpoint from the list (for example, `www.example.com`). Do not select the root domain endpoint (for example, `example.com`).

e. From **Routing Policy**, select **Simple**.
f. Leave **Evaluate Target Health** set to **No**.
g. Click **Create**.

Set Up a DNS Provider

If you registered a new domain name and have used that name while doing this tutorial, you're ready to set up Amazon Route 53 as your DNS provider.

Alternatively, if you're reusing a domain name that was previously associated with another website, you might need to transfer other DNS records from your current DNS provider to Amazon Route 53 in order to ensure the continued availability of the services hosted under the domain name. To determine which DNS records you must replicate in Amazon Route 53, check the DNS record settings configured for the domain in your current DNS provider. Two records that you should not transfer to Amazon Route 53 are the Start of Authority (SOA) and Name Server (NS) records. These records were set by Amazon Route 53 when the name servers were allocated, and they should not be changed.

First, log into the domain name registrar that you used to register your domain name. Use the web interface provided by the registrar to set the name servers for your domain to the name server values displayed under **Name Servers** in the details for the hosted zone. How you do this depends on the registrar that you used.

Wait between two to 48 hours for the Internet DNS resolver network to propagate name server changes. To see if the name server change has gone through, use a command line utility such as dig (for Mac OS X, Unix, or Linux) or nslookup (for Windows). The following example shows how use dig to see which name servers are associated with your domain.

```
dig example.com
```

When the AUTHORITY SECTION of the output shows the AWS name servers that you allocated using Amazon Route 53, the DNS changes have propagated through the DNS resolver network.

```
;; AUTHORITY SECTION:
example.com. 118928 IN NS ns-806.awsdns-36.net.
example.com. 118928 IN NS ns-1456.awsdns-54.org.
example.com. 118928 IN NS ns-1713.awsdns-22.co.uk.
example.com. 118928 IN NS ns-105.awsdns-13.com.
```

After your DNS changes have propagated, you'll be able to view your website using your custom domain name.

If you open your www subdomain (such as `www.example.com`) in your web browser, it redirects to your domain (such as `example.com`).

Step 6: Speed Up Your Website Using CloudFront

You can use Amazon CloudFront to improve the performance of your website. CloudFront makes your website's files (such as HTML, images, and video) available from data centers around the world (called *edge locations*). When a visitor requests a file from your website, the request is automatically redirected to a copy of the file at the nearest edge location, which results in faster download times than if the visitor had requested the content from a data center farther away. CloudFront caches content at edge locations for a period of time that you specify. When a visitor requests content that has been cached for longer than the expiration date, CloudFront checks the origin server to see if a newer version of the content is available. If a newer version is available, CloudFront copies the new version to the edge location. In this manner, changes that you make to the original content are replicated to edge locations as visitors request the content.

To speed up your website, use CloudFront to complete the following tasks.

Tasks

Create a CloudFront Distribution

First, you'll create a CloudFront distribution, which makes your website available from data centers around the world.

To create a distribution with an Amazon S3 origin

1. Open the CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Click **Create Distribution**.
3. On the **Select a delivery method for your content** page, under **Web**, click **Get Started**.
4. On the **Create Distribution** page, under **Origin Settings**, enter the Amazon S3 static website hosting endpoint for your bucket in the **Origin Domain Name** box. For example, `example.com.s3-website-us-east-1.amazonaws.com`.

Important
Do not select the name of your bucket from the list, for example, `example.com.s3.amazonaws.com`.

The **Origin ID** value is filled in for you.

5. Leave the values under **Default Cache Behavior Settings** at their default settings. For more information about these configuration options, see Values that You Specify When You Create or Update a Web Distribution in the *Amazon CloudFront Developer Guide*.
6. Under **Distribution Settings**, do the following:
   a. Leave **Price Class** set to **Use All Edge Locations (Best Performance)**.
   b. Set **Alternate Domain Names (CNAMEs)** to the root domain and www subdomain (in this tutorial, `example.com, www.example.com`). These values must be set in order to create A record aliases from the specified domain names to the CloudFront distribution.
   c. Set **Default Root Object** to `index.html`. This is the default page that the CloudFront distribution returns if the URL used to access the distribution does not contain a file name. This value should match the index document value that you set in Step 4: Deploy Your Website (p. 18).
   d. Set **Logging** to **On**.
   e. In **Bucket for Logs**, select the logging bucket that you created (`logs.`*`example.com`*).
   f. Set **Log Prefix** to `cdn/`, so that the logs generated by traffic to the CloudFront distribution are stored in a folder named `cdn` in the log bucket.
   g. Leave the other settings at their default values.
7. Click **Create Distribution**.

It can take up to 15 minutes to deploy the distribution. To view the current status of the distribution, find it in the console and check the **Status** column. A status of `InProgress` indicates that the distribution is not yet fully deployed.

When your distribution is deployed, you are ready to reference your content with your new CloudFront domain name. Make a note of the value of **Domain Name** in the CloudFront console. You'll need this value in the next step. In this example, the value is `dj4p1rv6mvubz.cloudfront.net`.

To verify that your CloudFront distribution is working, enter the domain name of the distribution in a web browser. If it is working, you will see your website display.

Update the Record Sets for Your Domain and Subdomain

Now that you have successfully created a CloudFront distribution, the next step is to update the A records in Amazon Route 53 to point to the new CloudFront distribution.

To update A records to point to a CloudFront distribution

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. On the **Hosted Zones** page, select the hosted zone that you created for your domain.
3. Click **Go to Record Sets**.
4. Select the A record that you created for the www subdomain.
5. Under **Alias Target**, select the CloudFront distribution.
6. Click **Save Record Set**.
7. Repeat this procedure to redirect the A record for the root domain to the CloudFront distribution.

This change will take effect within two to 48 hours. You can tell that the new A records have taken effect when going to `http://www.example.com` in a browser no longer redirects you to `http://example.com`. This change in behavior occurs because traffic routed by the *old* A record to the www subdomain S3 bucket is redirected by the settings in Amazon S3 to the root domain. When the new A record has taken effect, traffic routed by the new A record to the CloudFront distribution will not be redirected to the root domain.

> **Tip**
> Browsers can cache redirect settings. If you think the new A record settings should have taken effect, but you are still seeing `http://www.example.com` redirect to `http://example.com`, try clearing your browser history and cache, closing and reopening your browser application, or using a different web browser (if you have more than one installed).

At this point, any visitors who reference the site by using `http://example.com` or `http://www.example.com` are redirected to the nearest CloudFront edge location, where they will benefit from faster download times.

If you created your site as a learning exercise only, you can delete the resources that you allocated so that you no longer accrue charges. Continue on to Step 7: Clean Up Your Resources (p. 33). After you delete your AWS resources, your website will no longer be available.

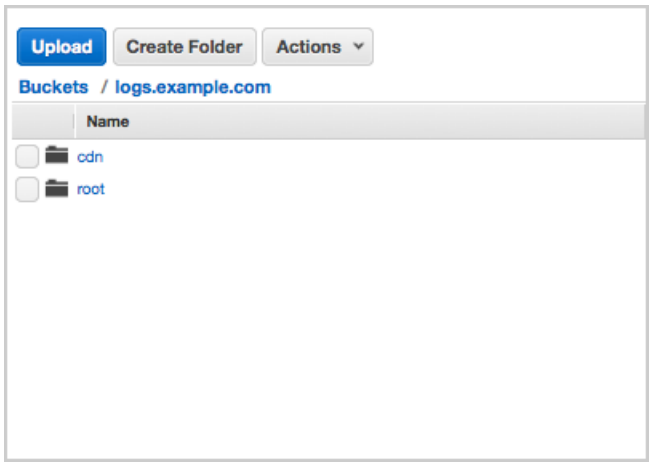
(Optional) Check the Log Files

The access logs tell you how many people are visiting the website, and they contain valuable business data that you can analyze with other services, such as Amazon Elastic MapReduce (Amazon EMR).

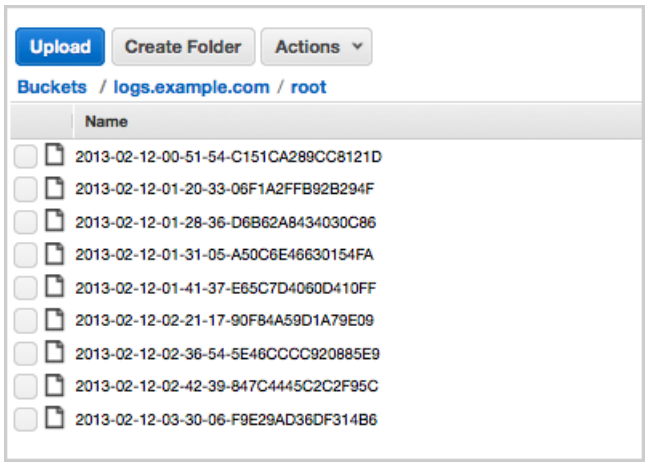
When you check the log files in your bucket, you should see older Amazon S3 log files in the folder `root`. All new log files should be CloudFront logs written in the folder `cdn`. Amazon S3 website access logs are written to your log bucket every two hours. CloudFront logs are written to your log bucket within 24 hours of the corresponding requests, so you may have to wait for them to show up.

To view the log files of your website

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Select the logging bucket for your website.
3. Click either `cdn` or `root` to view the log files stored within.



4. Double-click a log file to open it in the browser (as text files written by Amazon S3) or download it (as GZip files written by CloudFront).



Step 7: Clean Up Your Resources

If you created your static website as a learning exercise only, be sure to delete the AWS resources that you allocated so that you no longer accrue charges. After you delete your AWS resources, your website is no longer available.

Tasks

Delete the Amazon Route 53 Hosted Zone

Before you delete the hosted zone, you must delete the record sets that you created. You do not need to delete the NS and SOA records; these are automatically deleted when you delete the hosted zone.

To delete the record sets

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the list of domain names, select the check box that corresponds to your domain name, and then click **Go to Record Sets**.
3. In the list of record sets, select the check boxes that correspond to the A records that you created. The type of each record set is listed in the **Type** column.
4. Click **Delete Record Set**.
5. When prompted for confirmation, click **Confirm**.

To delete an Amazon Route 53 hosted zone

1. Continuing from the previous procedure, click **Back to Hosted Zones**.
2. Select the check box that corresponds to your domain name, and then click **Delete Hosted Zone**.
3. When prompted for confirmation, click **Confirm**.

Delete the CloudFront Distribution

Before you delete a CloudFront distribution, you must disable it. A disabled distribution is no longer functional and does not accrue charges. You can enable a disabled distribution at any time. After you delete a disabled distribution, it is no longer available.

To disable the CloudFront distribution

1. Open the CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. Right-click the distribution that you want to disable, and then click **Disable**.
3. When prompted for confirmation, click **Yes, Disable**.

To delete a CloudFront distribution

1. Continuing from the previous procedure, right-click a disabled distribution, and then click **Delete**.
2. When prompted for confirmation, click **Yes, Delete**.

Delete the Amazon S3 Bucket

Before you delete your Amazon S3 bucket, you should ensure that logging is disabled for the bucket; otherwise, we will continue to write logs to your bucket as you delete it.

To disable logging for a bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Select your bucket and then click **Properties**.
3. In the **Properties** pane, click **Logging**.
4. Clear the **Enabled** check box.
5. Click **Save**.

To delete a bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. Open the context (right-click) menu for the bucket and click **Delete Bucket**.
3. When prompted for confirmation, type the name of the bucket in the field, and then click **Delete**.

Related Resources

The following table lists some of the AWS resources that you'll find useful as you work with AWS.

Resource	Description
AWS Products & Services	Information about the products and services that AWS offers.
AWS Documentation	Official documentation for each AWS product, including service introductions, service features, and API reference.
AWS Discussion Forums	Community-based forums for discussing technical questions about Amazon Web Services.
Contact Us	A central contact point for account questions such as billing, events, and abuse. For technical questions, use the forums.
AWS Support Center	The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
AWS Support	The home page for AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
AWS Architecture Center	Provides the necessary guidance and best practices to build highly scalable and reliable applications in the AWS cloud. These resources help you understand the AWS platform, its services and features. They also provide architectural guidance for design and implementation of systems that run on the AWS infrastructure.
AWS Security Center	Provides information about security features and resources.
AWS Economics Center	Provides access to information, tools, and resources to compare the costs of Amazon Web Services with IT infrastructure alternatives.
AWS Technical Whitepapers	Provides technical whitepapers that cover topics such as architecture, security, and economics. These whitepapers have been written by the Amazon team, customers, and solution providers.

Resource	Description
AWS Blogs	Provides blog posts that cover new services and updates to existing services.
AWS Podcast	Provides podcasts that cover new services, existing services, and tips.