Getting Started with AWS

Static Website Hosting

Getting Started with AWS: Static Website Hosting

Host a Static Website on Amazon Web Services

If you want to host a website that only uses client-side technologies (such as HTML, CSS, and JavaScript) and that does not require server-side technologies (such as PHP and ASP.NET), you can easily and inexpensively host the site on Amazon Web Services (AWS). This type of site is sometimes referred to as a static website, and it is typically used to display reference material that does not change frequently, such as product manuals, informational sites, or documentation.

This guide walks you through the process of hosting a static website on AWS. We'll use the AWS Management Console to configure the AWS services we need.

By the end of this guide, you should be able to do the following:

- **Host a static website** – a static website hosted on Amazon Simple Storage Service (Amazon S3) can scale to support enterprise-level traffic. It is secure and fast, and it protects against data loss. Using Amazon S3, you can have a robust website without updating or managing server software. Amazon S3 provides a place to store your website files, and it delivers your files to site visitors.
- **Associate a custom domain name with your website** – Amazon Route 53 is a domain name service (DNS) that makes it easy to associate a custom domain name (such as example.com) with your website hosted on Amazon S3.
- **Increase the speed of your website** – Amazon CloudFront creates a content delivery network (CDN) that makes your website content available from data centers around the world, called edge locations, so your visitors can download content from a location near them. Using edge locations to serve web pages improves speed and reduces latency. Amazon CloudFront is especially useful if your site displays large media files such as high-resolution images, audio, or video.

If you want to host a website that requires server-side technologies, see Getting Started with AWS Computing Basics for Linux, which explains how to configure a virtual web server by using Amazon Elastic Compute Cloud (Amazon EC2). The guide includes instructions for installing Drupal, a popular open-source content management system (CMS) that uses PHP and MySQL. You can use the same process to create a WordPress site or a website that uses a custom PHP application. Also see Getting Started with AWS Computing Basics for Windows, which describes how to deploy DotNetNuke, an open source CMS that runs on the Microsoft Windows operating system.

What AWS Services Do I Need?

If you are creating and deploying a static website, you'll face infrastructure challenges. Where will you store your files? How will you deliver your content? How will you ensure reliability and high performance for your website? How will you route visitors to your website? To solve these challenges, AWS provides Amazon S3, Amazon Route 53, and Amazon CloudFront for seamless, and cost-effective solutions.

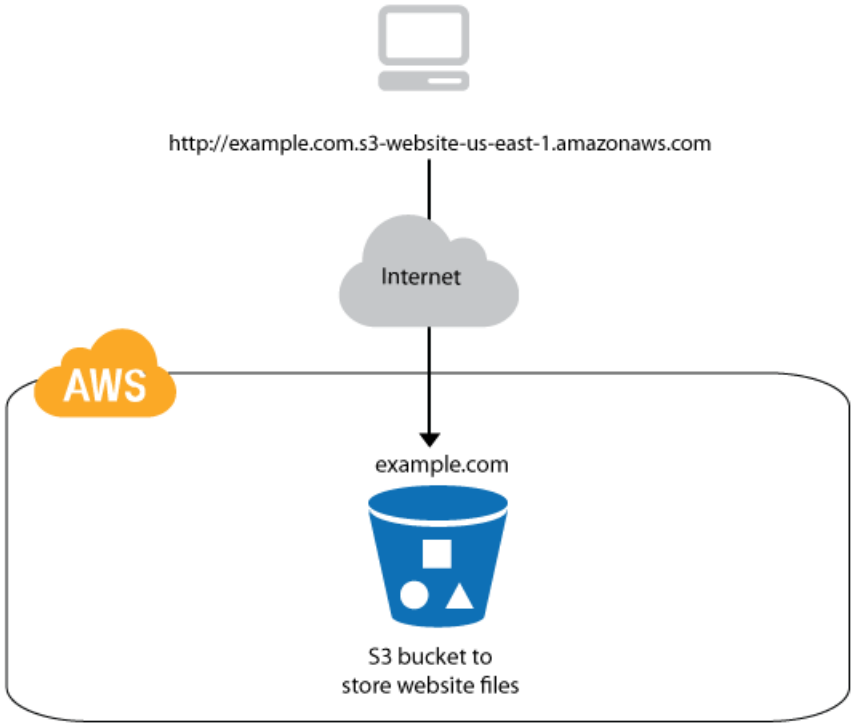
The following table gives a brief overview of the Amazon services used in this Getting Started Guide.

Challenges	Amazon Web Services	Benefits
You need a low-cost, reliable online storage web service to host static website content.	 Amazon Simple Storage Service (Amazon S3)	Amazon Simple Storage service offers a low cost, highly reliable solution for hosting static website content.
You need a reliable and cost-effective way to route end users to Internet websites.	 Amazon Route 53	Amazon Route 53 maps human-readable domain names to numeric IP addresses and AWS locations.
You need to deliver content with low-latency and high data transfer speeds so website visitors don't experience unnecessary delays.	 Amazon CloudFront	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 with the lowest latency.

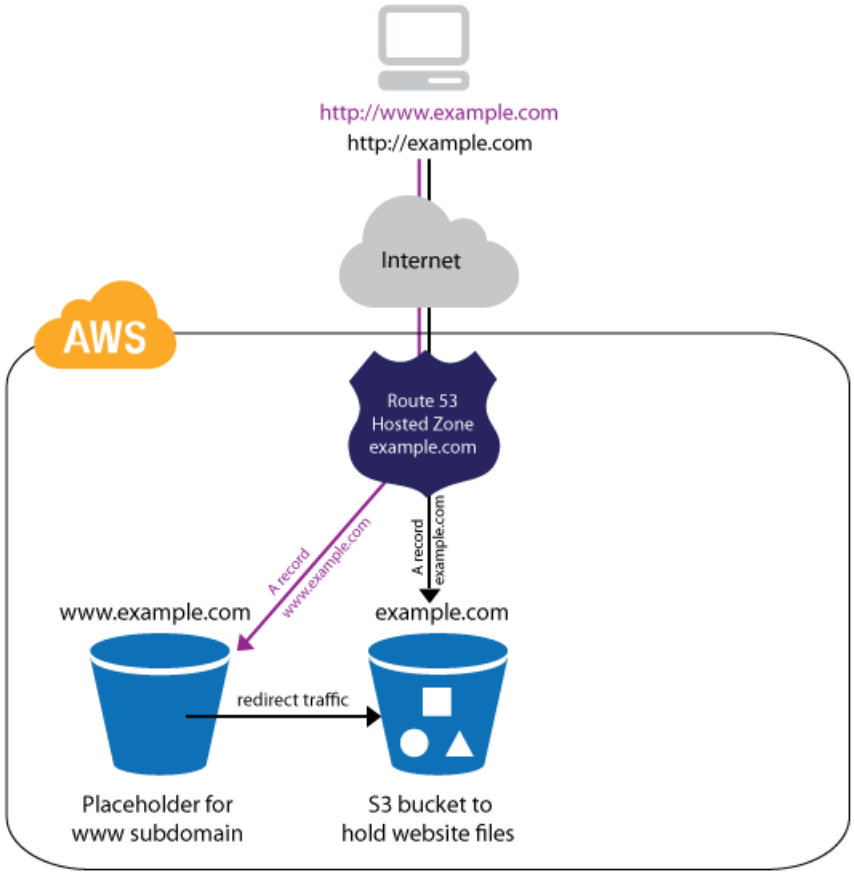
Static Website Hosting Architectures

This guide walks you through the process of hosting a static website on AWS. There are three options for how you configure your static website, depending on the functionality you require.

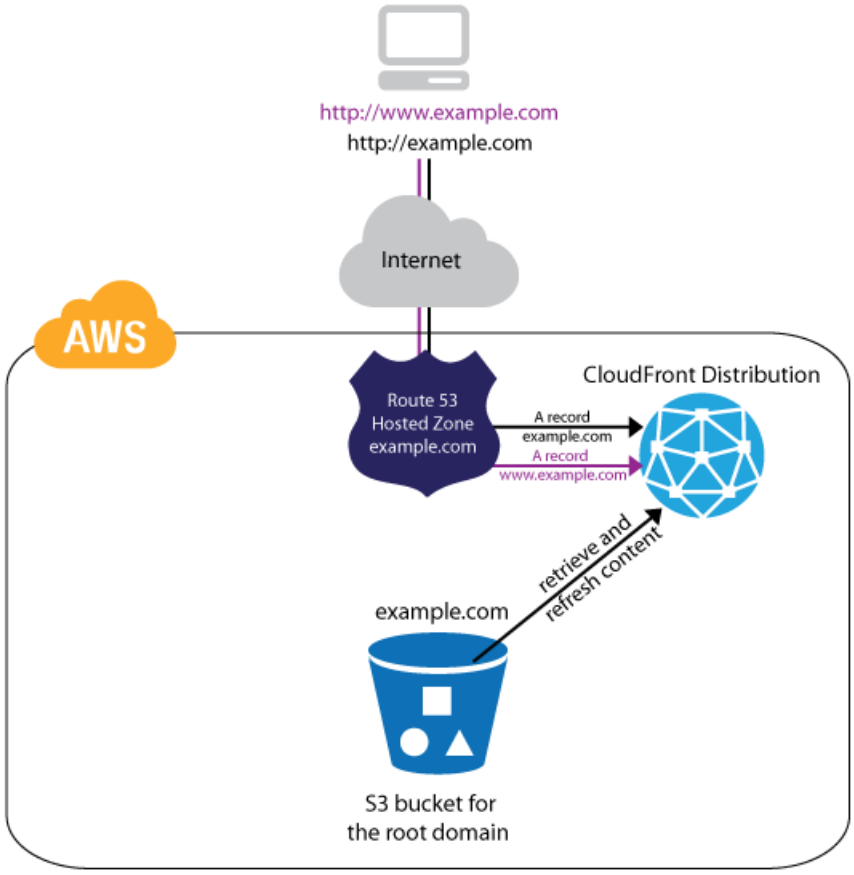
In the first option, you'll do the bare minimum to host a static website on AWS: create a place to store files, upload the website files (HTML, CSS, Javascript, images), make the files publicly viewable, and then configure the storage location to act as a website. With this option, visitors access the website with a URL of the form http://example.com.s3-website-us-east-1.amazonaws.com.



In the second option, you'll add a custom domain to your static website by configuring AWS as your DNS service provider. Visitors can now access the site with a custom URL like http://example.com.



In the third option, you'll improve performance of your website by distributing it through a content distribution network. Visitors will still access the site with the custom URL (http://example.com), but now they'll download the files from a CDN server close to them.



The following steps walk you through the tasks needed to set up each of the three options above. They follow best practices and demonstrate how different AWS products work together to deploy and run the website.

Topics

Step 1: Sign Up for the Service

If you don't already have an AWS account, you'll need to get one. Your AWS account gives you access to all services, but you will be charged only for the resources that you use. For this example walkthrough, the charges will be minimal.

To sign up for AWS

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

AWS notifies you by email when your account is active and available for you to use.

You use your AWS account to deploy and manage resources within AWS. If you give other people access to your resources, you will probably want to control who has access and what they can do. AWS Identity and Access Management (IAM) is a web service that controls access to your resources by other people. In IAM, you create users, which other people can use to obtain access and permissions that you define. For more information about IAM, go to Using IAM.

Step 2: Choose a Domain Name

When you host a website on Amazon S3, AWS assigns your website a URL based on the name of the storage location you create in Amazon S3 to hold the website files (called an S3 bucket) and the geographical region where you created the bucket. For example, if you create a bucket called `myawsbucket` on the east coast of the United States and use it to host your website, the default URL will be `http://myawsbucket.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 don't need to register a domain name. Most production sites, however, will want to brand their site with a custom domain name, such as "example.com". For the purposes of this guide, we'll assume you want to add a custom domain name to your site. If you'd rather not register a domain at this time, simply skip this step and Step 5: Associate a Domain Name with Your Website Using Amazon Route 53 (Optional) (p. 20). You can use a fictional domain name for the rest of the guide.

Domain names on the Internet are managed by the Internet Corporation for Assigned Names and Numbers (ICANN). You register a domain name through a domain name registrar, an ICANN-accredited organization that manages the registry of domain names. For a list of accredited registrars, go to http://www.internic.net/regist.html. When you have chosen a registrar, its website will have detailed instructions and pricing information for registering your domain name.

You can associate a domain name you have previously registered with your Amazon S3 website, or register a new domain name.

If you are registering a new domain name, check that the domain name you want is available with a domain name registrar. If it is available—before you pay to register the domain name—create the buckets in Amazon S3 as described in Step 3: Configure Storage on Amazon S3 (p. 7). Once you've successfully created the S3 buckets, register the domain name.

We recommend this sequence to ensure that you'll be able to use the domain name on Amazon S3 before you purchase the domain name. In order to map a domain name to an Amazon S3 bucket, the bucket name must be the same as the domain name so that Amazon S3 can properly resolve the host headers sent by web browsers. Additionally, Amazon S3 requires that bucket names be unique across of all AWS, and so to associate the domain `example.com` with a website hosted on Amazon S3, you must be able to create a bucket named "example.com". If another AWS user has already created a bucket named

"example.com", you won't be able to associate the domain `example.com` with your website hosted on Amazon S3.

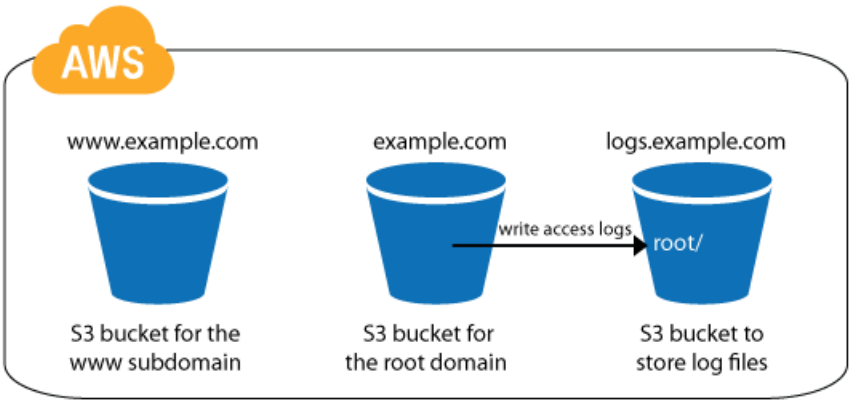
Step 3: Configure Storage on Amazon S3

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. Your files are stored in an Amazon S3 bucket as *objects*, and every object is stored in a location called a *bucket*. When you configure your Amazon S3 bucket as a website, the service delivers the files in your bucket to web browsers as if they were hosted on a web server.

Amazon S3 does not charge a fee for creating a bucket; only for storing objects and for transferring objects into and out of a bucket. The cost of storing objects is only a few cents per gigabyte per month. The Amazon S3 charges you will incur for deploying the example in this guide are minimal, approximately $0.02 a month. You may also be eligible to use Amazon S3 for free. For more information about pricing, see Amazon S3 Cost Breakdown (p. 38) and AWS Free Usage Tier.

In this step you will do the following:

- Create three Amazon S3 buckets: one for the root domain, one for the www subdomain, and one for log files.
- Configure permissions on your root domain bucket so that everyone can view the files it contains.
- Enable logging to record traffic to your root domain bucket.

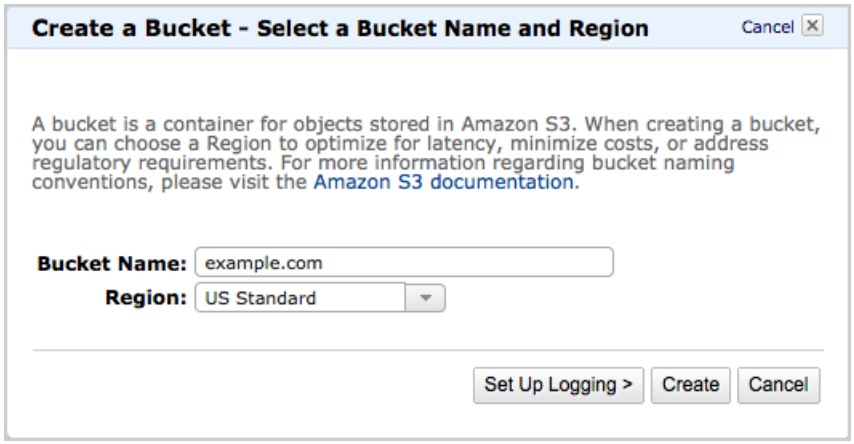


To create an S3 bucket for your static website

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Amazon S3 console, click **Create Bucket**.
3. In the **Create a Bucket – Select a Bucket Name and Region** dialog box, in the **Bucket Name** box, enter a bucket name.

   The bucket name you choose must be unique across all existing bucket names in Amazon S3. After you create a bucket, you cannot change its name. If you plan to associate a domain name with your bucket, you must give your bucket the same name as your domain name. This is a requirement of Amazon S3. For this example, we'll use `example.com`; however, you should use the domain name you selected in Step 2: Choose a Domain Name (p. 6). The name of the bucket needs to be the same as the domain name so that Amazon S3 can properly resolve the host headers sent by web browsers that request content from the site.

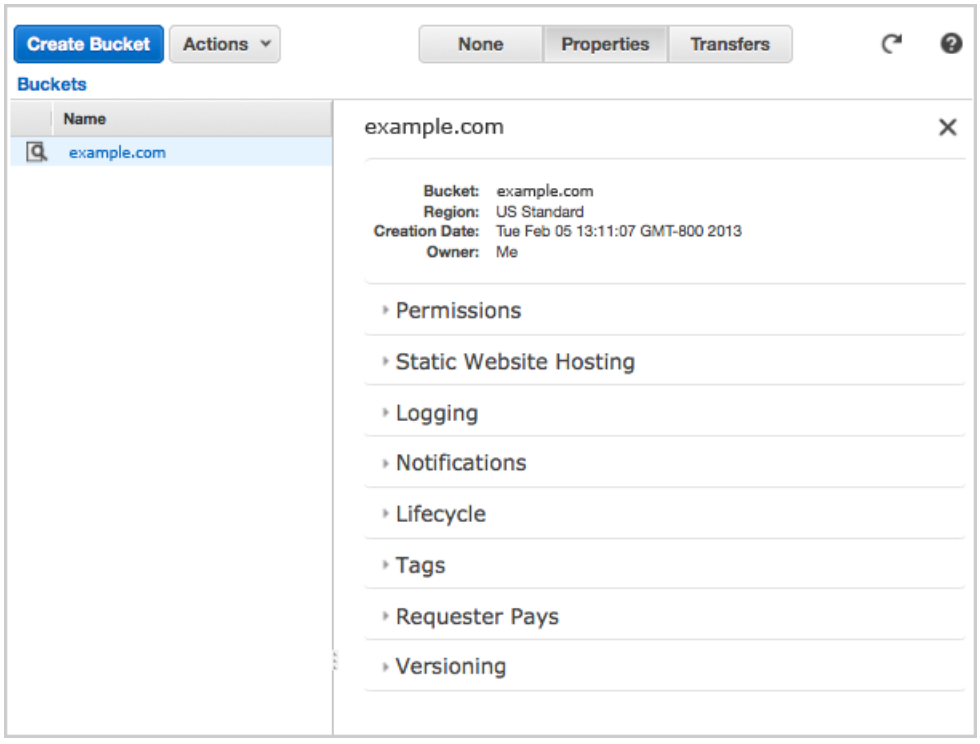
In some AWS regions, there might be additional restrictions on bucket names. For more information, go to Bucket Restrictions and Limitations in the *Amazon Simple Storage Service Developer Guide*.



4. In the **Region** box, select a region.

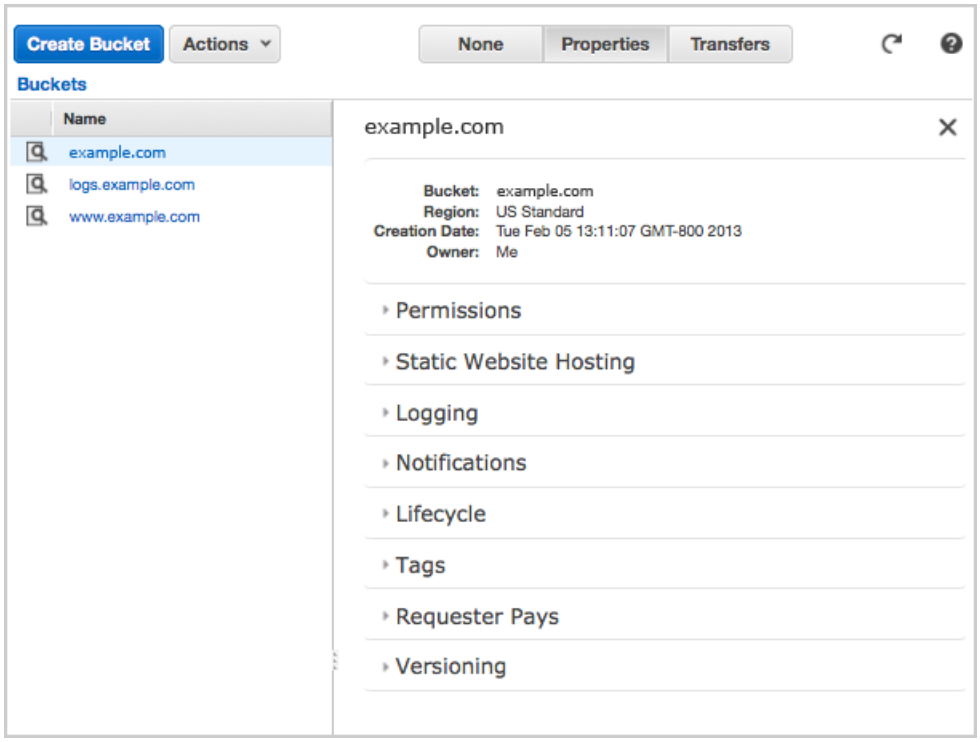
   By default, Amazon S3 creates buckets in the US Standard region. To reduce latency, minimize costs, or address regulatory requirements, you can choose a region that is closer to your website's audience. Objects stored in a region never leave that region unless you explicitly transfer them to another region. For more information about regions, go to Regions and Endpoints.
5. When the settings are as you want them, click **Create**.

   When Amazon S3 successfully creates your bucket, the console will display its name in the **Bucket** pane. This is the bucket where you'll upload your website files.



6. Repeat this procedure to create two more S3 buckets `logs.`*`example.com`* and `www.`*`example.com`*, using your domain name instead of *`example.com`*. Make sure to create these buckets in the AWS region where you created the root domain bucket. The `logs.example.com` bucket will be where

Amazon S3 logs information about traffic to your website. In Step 5: Associate a Domain Name with Your Website Using Amazon Route 53 (Optional) (p. 20), we'll configure the `www.example.com` bucket as a placeholder to redirect traffic to the root domain bucket if a user specifies the www subdomain. To host other subdomains (blog.example.com) you can create additional buckets for those. In this guide, we only show how to set up the www subdomain, but the principal is the same.



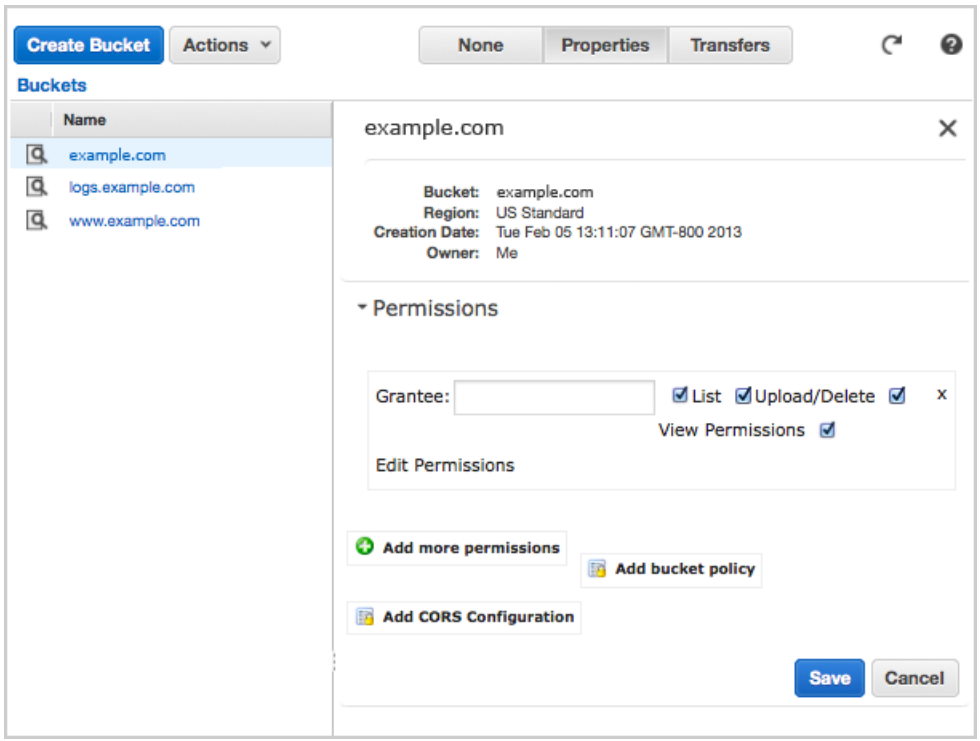
Now that you've allocated the Amazon S3 buckets we need for the website, let's set permissions on the root domain bucket so that Internet visitors will be able to access the website files you'll store there.

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 and subdomain buckets that anyone to view their contents.

Bucket policies control user access to both a bucket and the objects in it. The policies provide a fine granularity of access control for Amazon S3 resources. The policies also allow you to set permissions for a large number of objects with one statement. For more information, see Using Bucket Policies in the *Amazon Simple Storage Service Developer Guide*.

To set access permissions on your S3 bucket

1. In the Amazon S3 console, in the **Buckets** pane, right-click the root domain bucket you created (in this guide, we used `example.com`) and then click **Properties**. In the details pane, click **Permissions**.



2. Under **Permissions**, click **Add bucket policy**.
3. The following policy gives everyone permission to view any file in the `example.com` bucket. A bucket policy is a collection of JavaScript Object Notation (JSON) statements written in the access policy language. Copy the text from this document, and then paste it into the Bucket Policy Editor. Replace *`example.com`* with the name of your bucket. The date in this policy is the API version of Amazon S3 that the policy applies to. Do not modify this date. For more information about bucket policies, go to Access Control in the *Amazon Simple Storage Service Developer Guide*.

```
{
  "Version":"2008-10-17",
  "Statement":[{
        "Sid":"Allow Public Access to All Objects",
         "Effect":"Allow",
           "Principal": {
             "AWS": "*"
          },
       "Action":["s3:GetObject"],
       "Resource":["arn:aws:s3:::example.com/*"
      ]
    }
  ]
}
```

4. When you have finished revising the policy for your bucket, in the **Bucket Policy Editor**, click **Save**. In the Amazon S3 bucket, under **Permissions**, click **Save**.

Now that you've configured permissions for the root domain bucket so that everyone can access its files, let's enable logging to track visitor usage.

In order to track the number of visitors accessing your website, you need to enable logging for the root domain bucket. 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 an Amazon S3 bucket; however, any log files the system delivers to you will accrue the usual charges for storage. (You can delete the log files at any time.) Amazon S3 does not assess data transfer charges for log file delivery, but they do 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*.

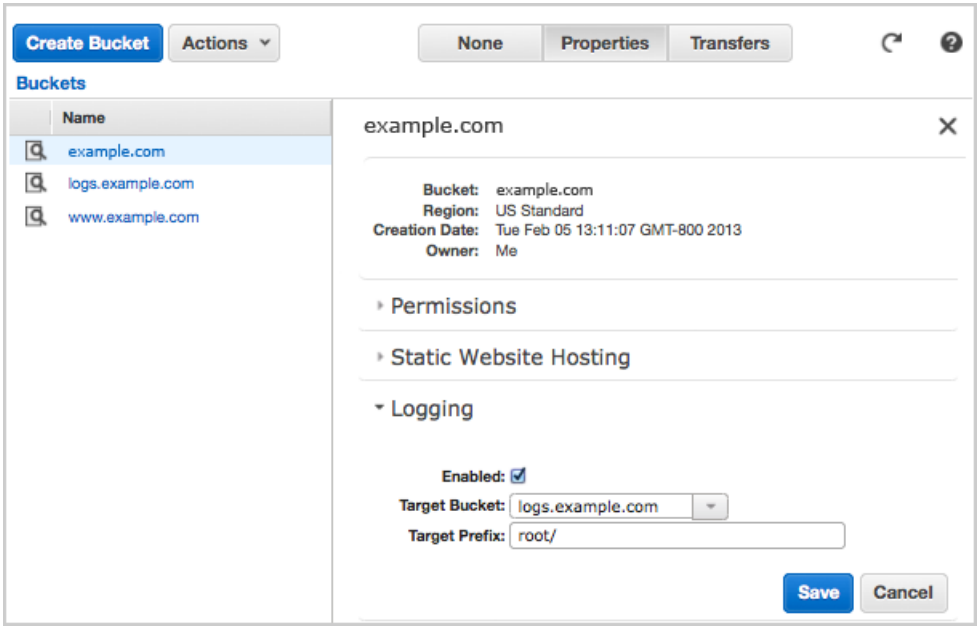
Enabling logging is optional. If you are creating a website where you don't want to track traffic to your website, you can skip the following procedure.

To enable logging for your root domain S3 bucket

1. In the Amazon S3 console, right-click the root domain bucket you created (in this guide, `example.com`) and then click **Properties**.
2. In the **Properties** pane, do the following. When all the settings are as you want them, click **Save**:

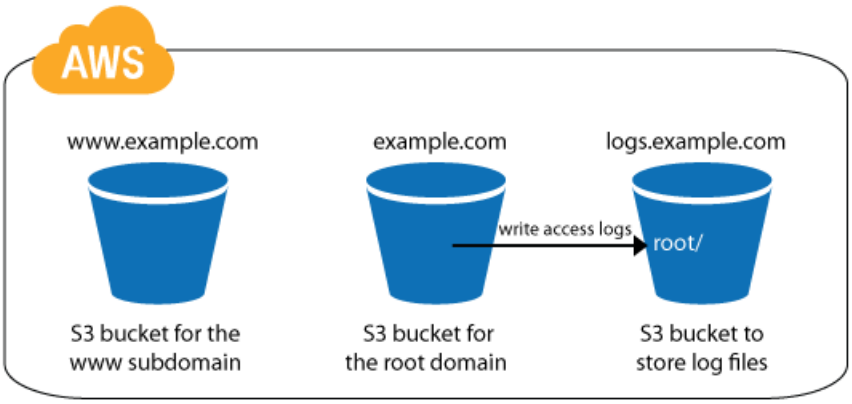
   - Click **Logging**.
   - Select the **Enabled** check box.
   - In the **Target Bucket** box, click the Amazon S3 bucket you created to hold log files (in this guide `logs.example.com`).

     You can create and select any Amazon S3 bucket to contain your log files, with a caveat. You should not store your log files in the bucket that you're collecting logs on, because that creates a situation in which viewing the log files generates more log files.
   - In the **Target Prefix** box, type `root/`. This setting causes the log data to be stored in a folder named `root` folder in the `logs.example.com` bucket. Storing your logs in a folder will be useful when we get to Step 6: Speed Up Your Website Using CloudFront (Optional) (p. 27).



Where You're At

You've allocated storage for your website on Amazon S3, configured access permissions on your root domain bucket, and set up logging to track the visitors to your website.



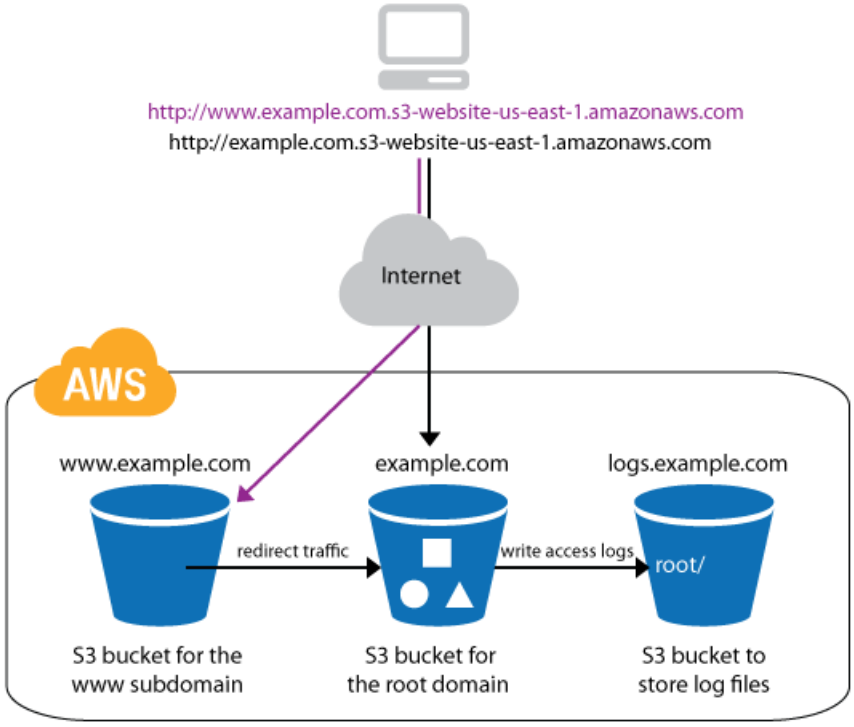
In Step 4: Launch Your Website on Amazon S3 (p. 12) you'll upload files to the root domain bucket and configure it as a website.

Step 4: Launch Your Website on Amazon S3

Now that you've created and configured your Amazon S3 buckets, you are ready to upload your website files and launch your website.

In this step you will do the following:

- Upload your static website files to your root domain Amazon S3 bucket
- Configure your root domain bucket as a website
- Redirect traffic from the www subdomain bucket to the root domain bucket
- Verify that your website works



Before you upload any files, you'll need to create a root HTML file and an error HTML file for your website. Create these files on your computer with the names `index.html` and `error.html`. To test your website, you will add simple HTML to the files.

Add the following HTML to `index.html`:

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

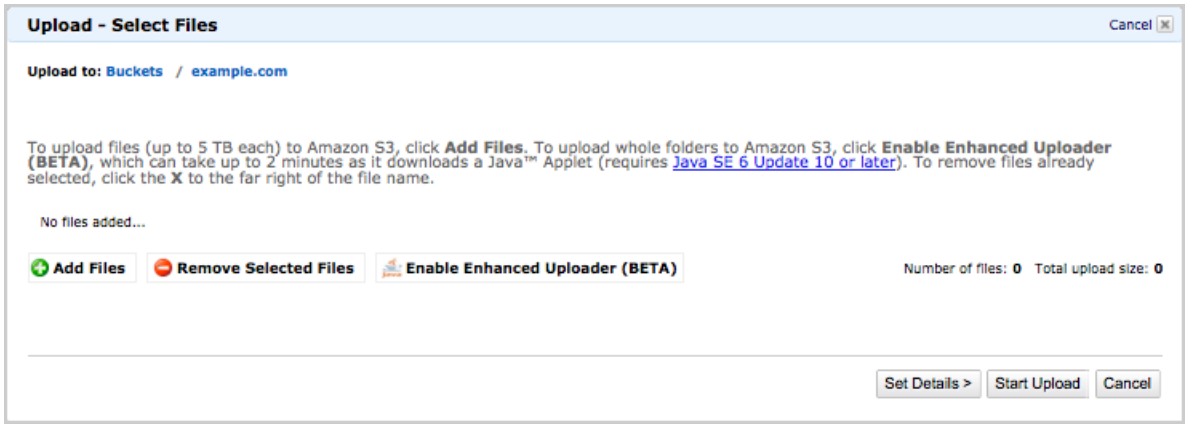
Add the following HTML to `error.html`:

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

Save these files on your computer where you can easily find them later.

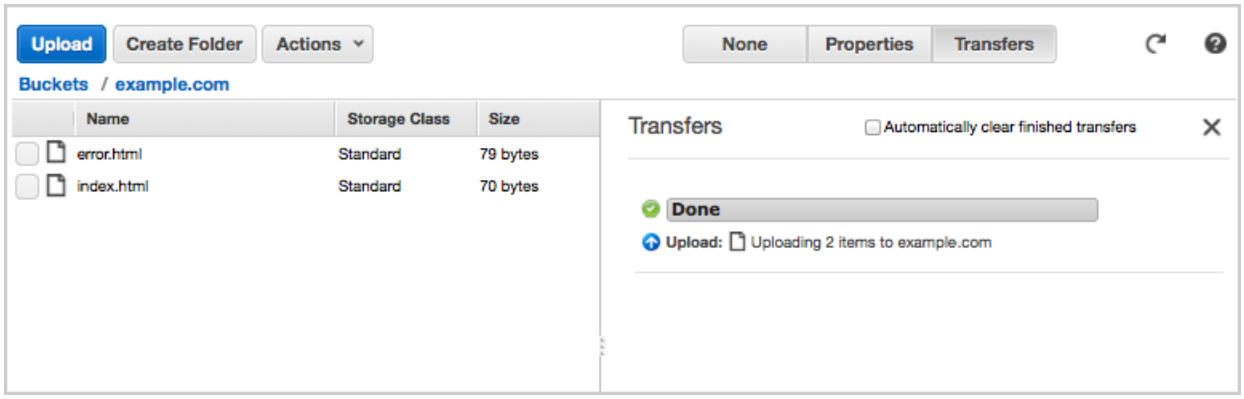
To upload your website files to your root domain Amazon S3 bucket

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the Amazon S3 console, select the root domain bucket that you previously created to contain your static website files. In our example, we will select `example.com` bucket.
3. Click **Actions**, and then click **Upload**.



4. In the **Upload — Select Files** dialog box, click **Add Files**.
5. In the **File Upload** dialog box, select the `index.html` and `error.html` files you created, and then click **Open**.
6. In the **Upload — Select Files** dialog box, click **Start Upload**.

When your files have finished uploading, they will appear as shown in the following image.

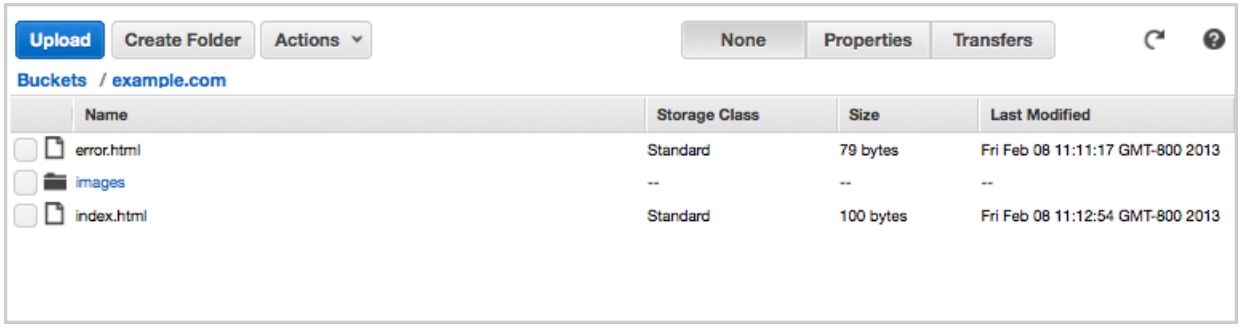


After you have uploaded your `index.html` and `error.html` files, you can add any other files you want to use for your website.

If your website files have a folder hierarchy on your local computer, such as storing image files in an `images` subdirectory, you will need to recreate that hierarchy in your buckets on Amazon S3. 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 using the following HTML.

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

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



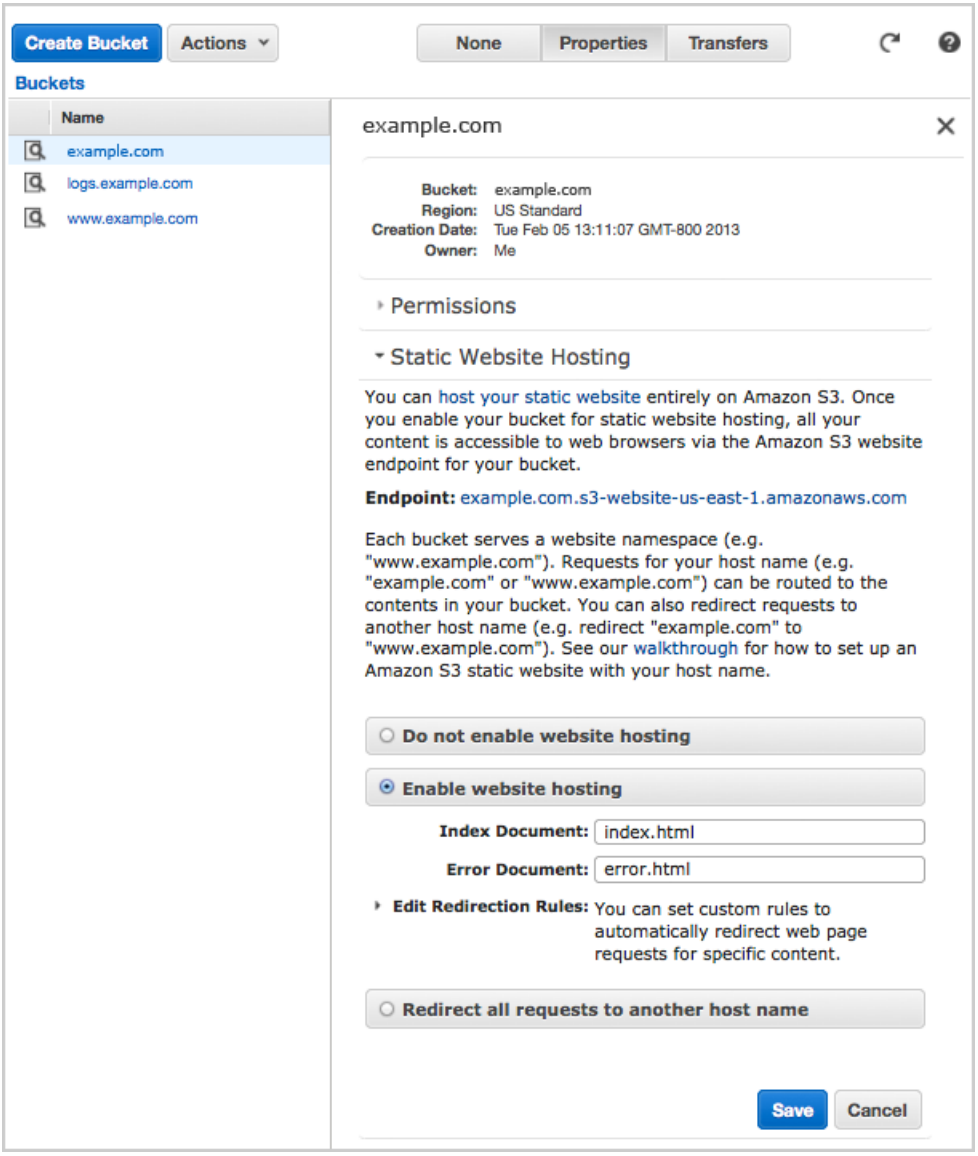
Knowing how to handle folder hierarchies on an Amazon S3 static website is important, but for the purposes of keeping this guide simple, we'll use just the two files `index.html` and `error.html` in a single bucket.

Now that you've uploaded the files of your website, it's time to configure Amazon S3 to serve those files as if they were hosted on a web server. To do so, we'll configure your root domain bucket as a website.

To configure your root domain Amazon S3 bucket as a website

1. In the Amazon S3 console, in the **Buckets** pane, right-click your root domain bucket, and then click **Properties**.
2. In the details pane, click **Static Website Hosting**.
3. Click **Enable website hosting**.
4. Copy the value of **Endpoint**, for example, **example.com.s3-website-us-east-1.amazonaws.com**. You'll need this value if you decide to perform Step 6: Speed Up Your Website Using CloudFront (Optional) (p. 27).
5. In the **Index Document** box, type `index.html`.

   In the **Error Document** box, type `error.html`.



6. Click **Save**.

To prepare for Step 5: Associate a Domain Name with Your Website Using Amazon Route 53 (Optional) (p. 20), we need to redirect traffic from the www subdomain bucket to the root domain bucket.

Amazon S3 will then forward 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 web address.

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

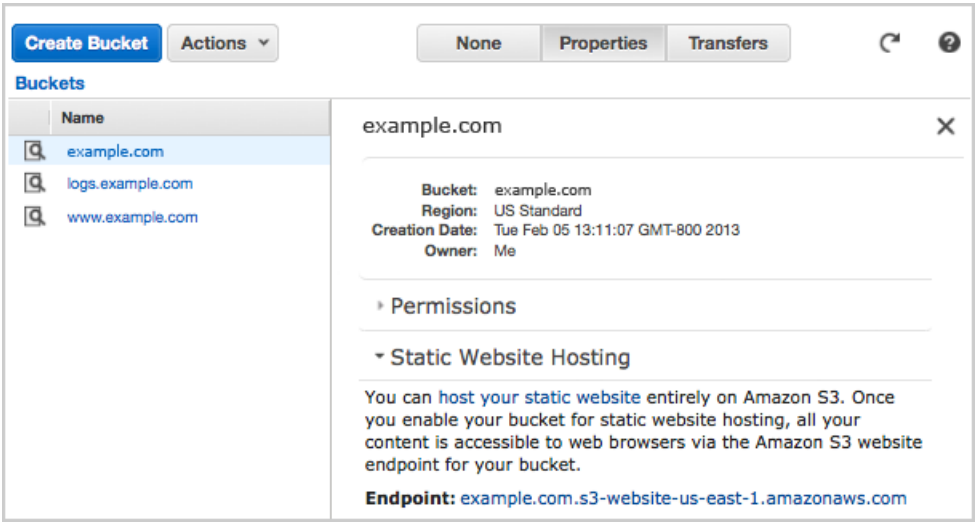
1. In the Amazon S3 console, right-click the subdomain bucket you created (in this guide, we used `www.example.com`) and select **Properties**.
2. Click **Static Website Hosting**.
3. Select the **Redirect all requests to another host name** radio button.
4. In the **Redirect all requests to** box, type the name of your root domain. In this guide we used `example.com`.

5. Click **Save**.

Your static website hosted on Amazon S3 is now live on the Internet. You can verify this with a web browser, by navigating to the default URL assigned by Amazon Web Services.

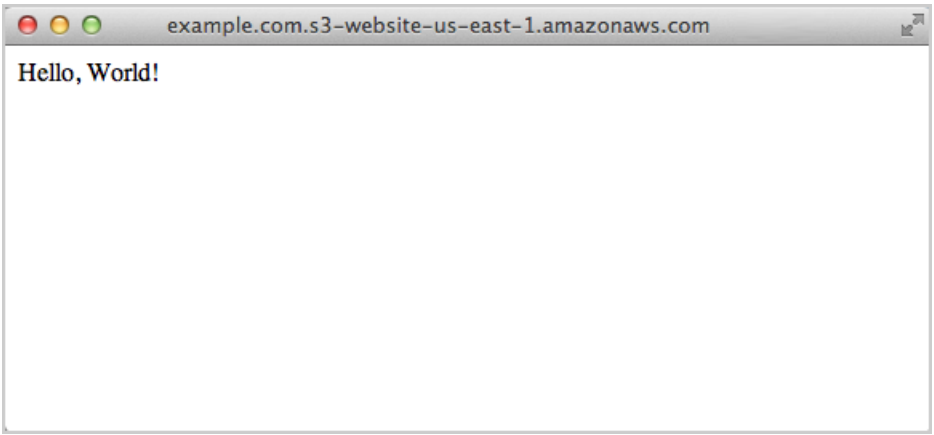
To locate the default URL of your Amazon S3 website

1. In the Amazon S3 console, right-click your root domain bucket and then click **Properties**.
2. In the details pane, click **Static website hosting**. The default URL assigned by Amazon Web Services is the **Endpoint**. In the following image, this is `example.com.s3-website-us-east-1.amazonaws.com`



To verify that your website is working

1. In a web browser, go to the default URL of the website you've hosted on Amazon S3, for example, `http://`*`example.com`*`.s3-website-us-east-1.amazonaws.com`, where *`example.com`* is replaced with the name of your custom domain. If your website is correctly deployed, you'll see your website's home page.

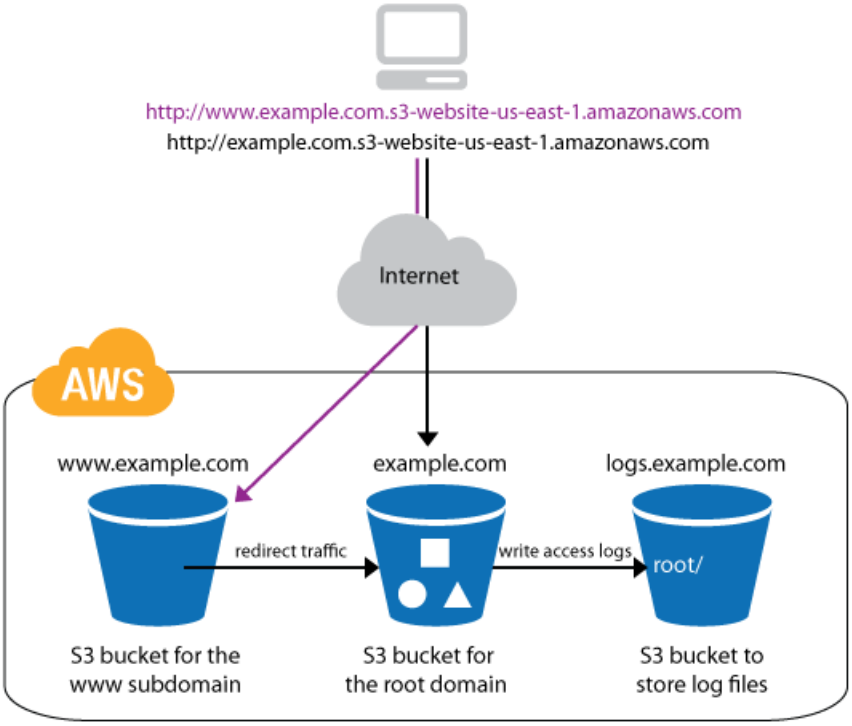


You can also test whether the subdomain bucket is properly redirecting visitors by testing the URL `http://www.example.com.s3-website-us-east-1.amazonaws.com`. You should be redirected to `http://example.com.s3-website-us-east-1.amazonaws.com`.

2. To verify that the error page is working, try to access a non-existent page on your new website, such as `http://example.com.s3-website-us-east-1.amazonaws.com/bogus.html`, where *example.com* is replaced with the name of your custom domain. If your website is correctly deployed, you'll be redirected to your custom error page.

Where You're At

Congratulations! You now have a working website deployed on Amazon S3. If you are building a website in which the default URL provided by AWS (`http://example.com.s3-website-us-east-1.amazonaws.com`) is sufficient, you can stop here.



In Step 5: Associate a Domain Name with Your Website Using Amazon Route 53 (Optional) (p. 20), we'll use Amazon Route 53 to associate a custom domain with this site so that visitors can access it with URLs like `http://example.com` and `http://www.example.com` instead of the long default URL provided by Amazon Web Services.

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

This step is associates a memorable domain name with your Amazon S3 bucket. It is optional. If you want to associate a domain name with your website and you skipped Step 2: Choose a Domain Name (p. 6), go back and do that step now. If you performed step 2, you must have used your domain name in place of *`example.com`* throughout this guide.

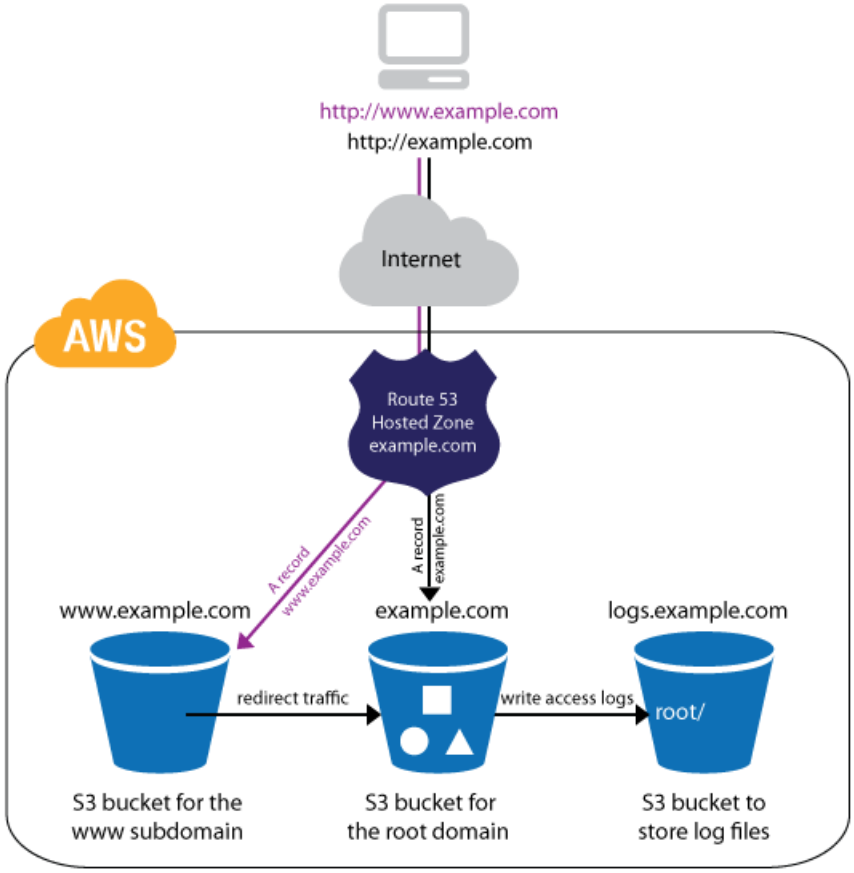
Amazon Route 53 is the service we'll use to associate your custom domain name with your static website. Amazon Route 53 is a scalable Domain Name System (DNS) service that provides secure and reliable routing to locations both inside and outside of AWS.

With Amazon Route 53, you pay only for the domains you configure and the number of queries that the service answers. The Amazon Route 53 charges that you will incur for following this guide are minimal, approximately $0.50 a month. For information about Amazon Route 53 pricing, see Amazon Route53 Cost Breakdown (p. 40).

In this step you will do the following:

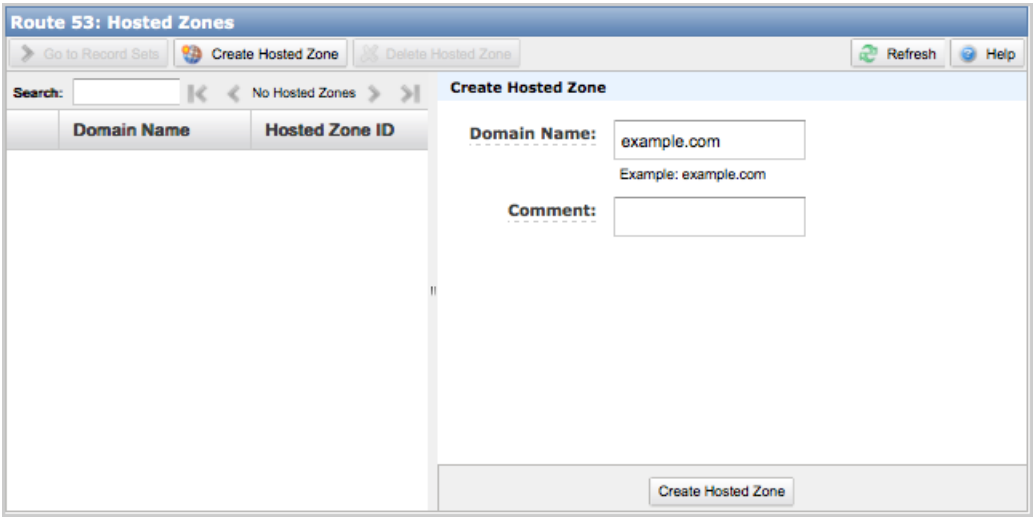
- Allocate name servers on Amazon Route 53 to route traffic for your custom domain
- Configure routing records for your root domain and www subdomain
- Set up Amazon Route 53 as your DNS provider

In this step, you'll associate an A record with the root domain and the www subdomain. The following diagram shows how Amazon Route 53 routes the root domain and www subdomain traffic after this step is completed.



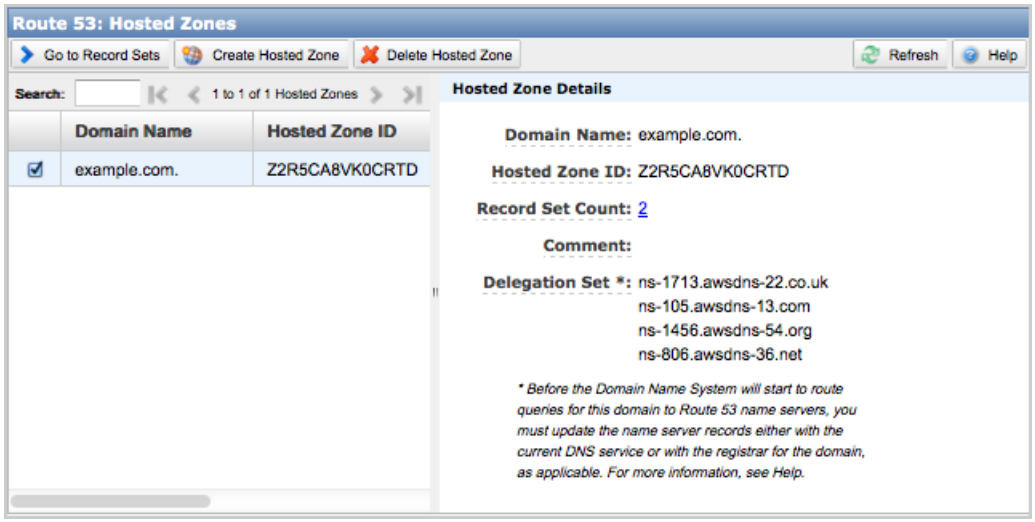
To allocate name servers on Amazon Route 53

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the **Route 53: Hosted Zones** window, click **Create Hosted Zone**.
3. In the **Domain Name** box, type your custom domain name.



4. Click the **Create Hosted Zone** button at the bottom of the pane.

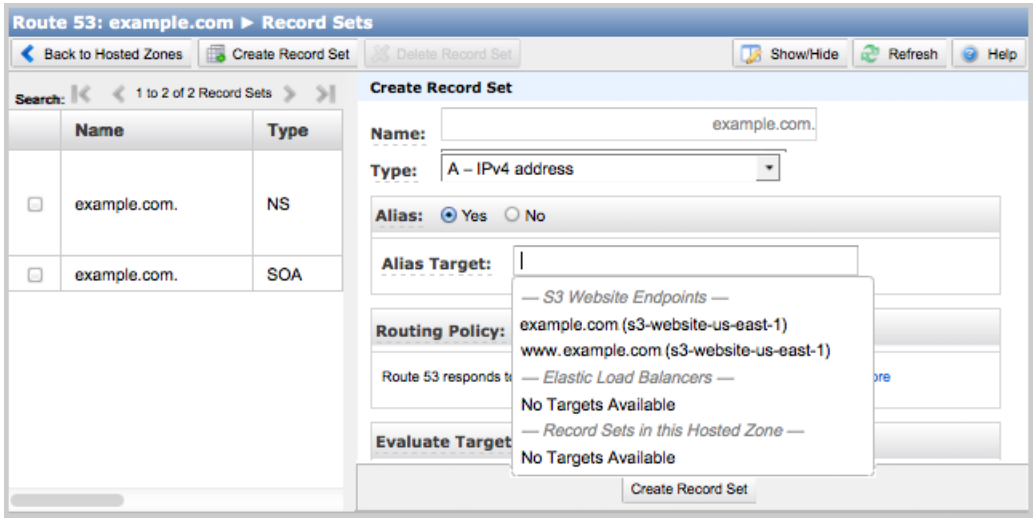
5. When the hosted zone is successfully created, in the list of domain names, select the check box that corresponds to your domain name. In the details pane, under **Delegation Set**, Amazon Route 53 displays the set of name servers that have been allocated for your domain. You'll be needing the names of these name servers later on.



To configure the A record for your root domain

1. In the Amazon Route 53 console, in the details pane, select the check box that corresponds to your domain name.
2. Click **Go to Record Sets**.
3. Click **Create Record Set**.
4. Under **Create Record Set**, do the following:

   - In the **Name** box, accept the default. The root domain is already entered for you.
   - In the **Type** box, select **A — IPv4 address** to create an A record for your domain.
   - In **Alias**, click **Yes**. This option causes the A record you create to point to an AWS resource—in this case, an Amazon S3 bucket—instead of an IP address. An alias makes it possible to associate your custom domain with the website you have hosted in Amazon S3. Normally, an A record can be associated only with a numerical IP address (such as 192.0.2.255). When you create the A record as an alias, AWS maps the A record to your Amazon S3 endpoint. Amazon Route 53 is the only DNS provider that can map the A record of a domain to an S3 bucket.
   - Click **Alias Target**. Select your root domain website endpoint (`example.com`) from the dropdown list that appears. You created this endpoint in Step 4: Launch Your Website on Amazon S3 (p. 12). Take care not to select the www subdomain endpoint instead (`www.example.com`).
   - In the **Routing Policy** box, select **Simple**.
   - Leave **Evaluate Target Health** set to **No**.



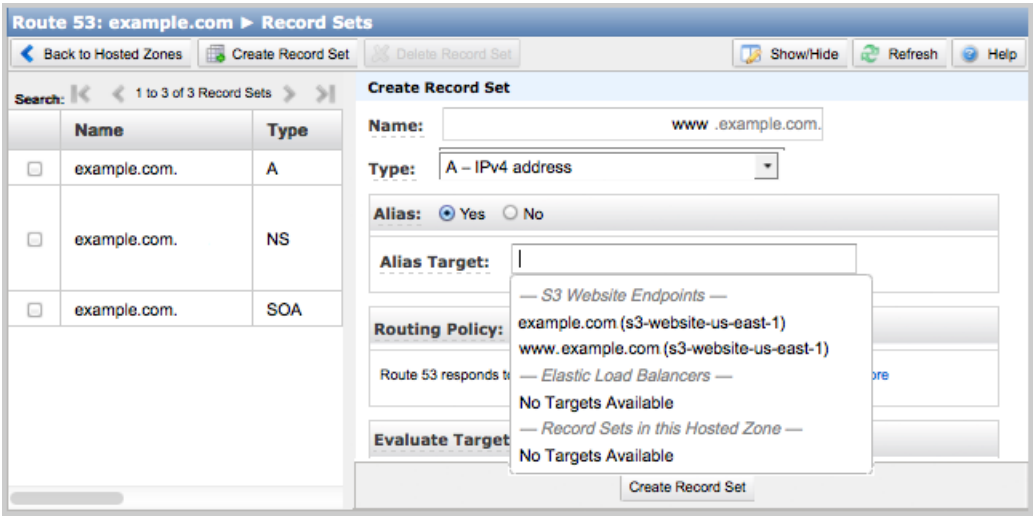
5. Click the **Create Record Set** button at the bottom of the pane.

Next, you'll create an A record that points to the www subdomain. Amazon Route 53 does not charge for queries made against aliases. If you create A records pointing to aliases for both the root domain and the www subdomain the only Amazon Route 53 charge is for the domain hosting. If you had used a CNAME instead of an A record for the www subdomain, you would also be charged for queries on that CNAME. By using an A record, you also save your site's visitors a second DNS lookup to resolve the CNAME.

To configure the A record for your www subdomain

1. In the Amazon Route 53 console, in the details pane, select the check box that corresponds to your domain name.
2. Click **Go to Record Sets**.
3. Click **Create Record Set**.
4. Under Create Record Set, do the following:

   - In the **Name** box, type `www`. The root domain is already entered for you, and the connecting period (.) is entered for you when you start typing.
   - In the **Type** box, select **A — IPv4 address** to create an A record for your domain.
   - In **Alias**, click **Yes**.
   - In the **Alias Target** box, click your www subdomain website endpoint (`www.example.com`) from the dropdown list that appears. You created this endpoint in Step 4: Launch Your Website on Amazon S3 (p. 12). Take care not to select the root domain endpoint instead (`example.com`).
   - In the **Routing Policy** box, click **Simple**.
   - Leave **Evaluate Target Health** set to **No**.

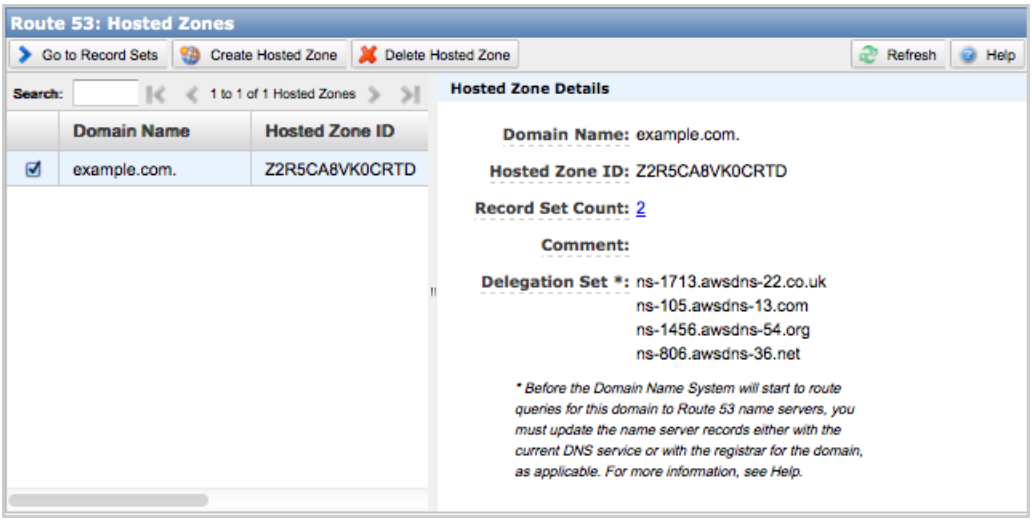


5. Click the **Create Record Set** button at the bottom of the pane.

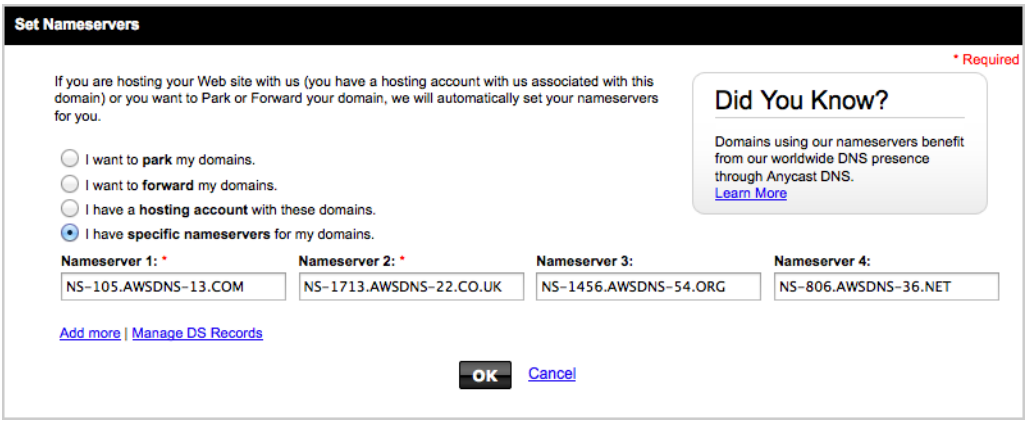
If you registered a new domain name in Step 2: Choose a Domain Name (p. 6) and have used that name during this guide, you're ready to set up Amazon Route 53 as your DNS provider. If you're reusing a domain name that was previously associated with another website, you may need to transfer other DNS records from your current DNS provider to Amazon Route 53 in order to ensure the continued c of the services hosted under the domain name. For example, Mail Exchange (MX) records specify routing for email associated with the domain. If you do not replicate your MX records in Amazon Route 53 before you make Amazon Route 53 your DNS provider, your email will not be properly routed. To determine what DNS records you need to 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.

To set up Amazon Route 53 as your DNS service provider

1. In the Amazon Route 53 console, in the details pane, select the check box that corresponds to your domain name.
2. Make a note of the values in **Delegation Set**.



3. Log into the domain name registrar that you used to register your domain name in Step 2: Choose a Domain Name (p. 6). Use the registrar's web interface to change the name servers for your domain to the name server values you noted in the previous step. How you do this depends on the registrar that you used. For the specific procedure, see their online help. Below is an example of updating the name servers on a domain provider's website.



Some registrars allow you to specify name servers only by using IP addresses; they don't allow you to specify fully qualified domain names. If your registrar requires using IP addresses, you can get the IP addresses for your name servers by using a command line utility such as dig (for Mac OS X, Unix, or Linux) or nslookup (for Windows). The following example shows how to locate the IP address associated with the domain name of the name server `ns-330.awsdns-41.com` using dig.

```
dig ns-105.awsdns-13.com
```

The value you would use to set the name server if you have to use an IP address instead of a domain name is the IP address that is returned under ANSWER SECTION in the dig output. You should use the IP address only if it is required by your domain registrar. The preferred method of setting the name servers is by using the domain name. Using the domain name protects you if the IP address associated with a allocated name server ever changes.

```
;; ANSWER SECTION:
ns-105.awsdns-13.com. 172800 IN A 203.0.113.0
```

4. Wait for 2 to 48 hours. This is the time it takes the Internet DNS resolver network to propagate name server changes. To see if the name server change has gone, 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.
```

If you don't want to use a command line utility to determine when the new name server values have taken effect, you can try to go to `http://example.com` in a web browser every so often. When you see your "Hello World" web page, you know the DNS changes have propagated out.

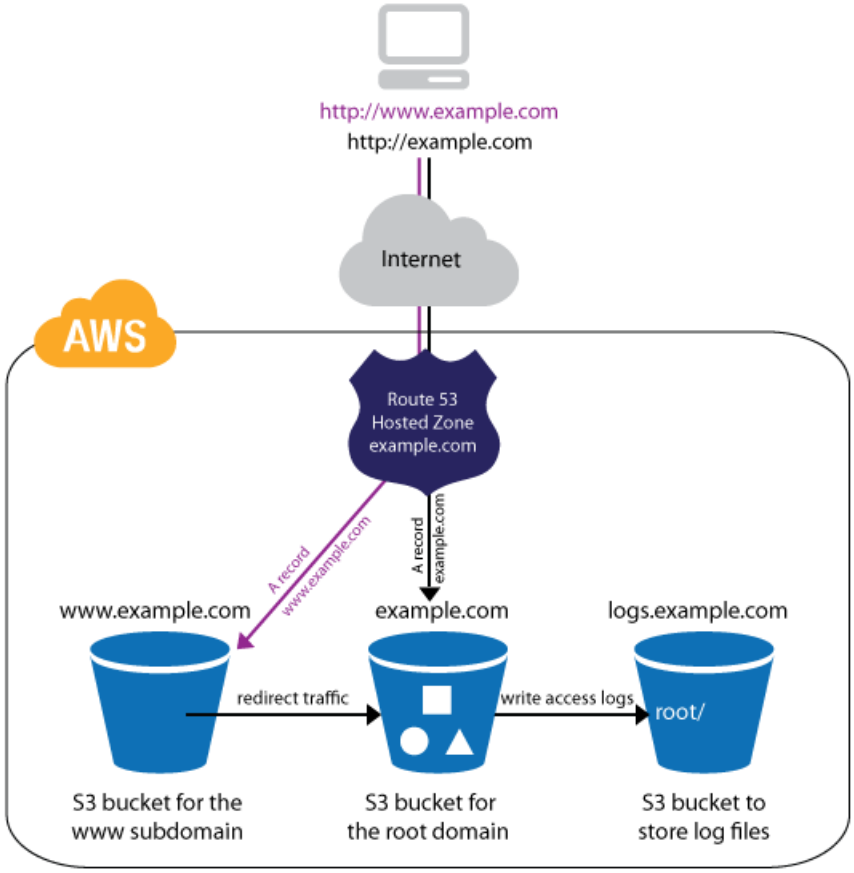
After your DNS changes have propagated, you'll be able to view your website using your custom domain name.

If you open `www.example.com` in your web browser, it will redirect to `example.com`.

Where You're At

Congratulations! You've successfully associated a custom domain with your website hosted on Amazon S3.

If your website has few visitors, does not include large files, and most users are in the same geographical area as your root domain bucket, you probably don't need a CloudFront distribution network yet. You can add a CloudFront distribution to your site later, when traffic increases.



In Step 6: Speed Up Your Website Using CloudFront (Optional) (p. 27), we'll use Amazon CloudFront to create a content delivery network (CDN) that will improve the speed of your website. CloudFront sends visitors to edge locations, automatically updating the content cached at that location if a new version is available. The improvement in performance is especially notable if you are delivering large files, such as high resolution images, audio, or video.

Step 6: Speed Up Your Website Using CloudFront (Optional)

Amazon CloudFront is the service that we'll use to increase 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, he or she is invisibly redirected to a copy of the file at the nearest edge location, which results in faster download times than if the visitor had accessed 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 new version of the content is available. If there is, CloudFront copies the new version to the edge location. In this manner, changes that you make to the original content are replicated out to edge locations as visitors request the content.

With CloudFront, the fee that you pay is based on the global edge locations you deploying to, and the volume of data that is transferred from those edge locations. The CloudFront charges that you will incur

for the example deployment in this guide are minimal, approximately $4.70 a month. For more information about pricing, go to CloudFront Cost Breakdown (p. 41).

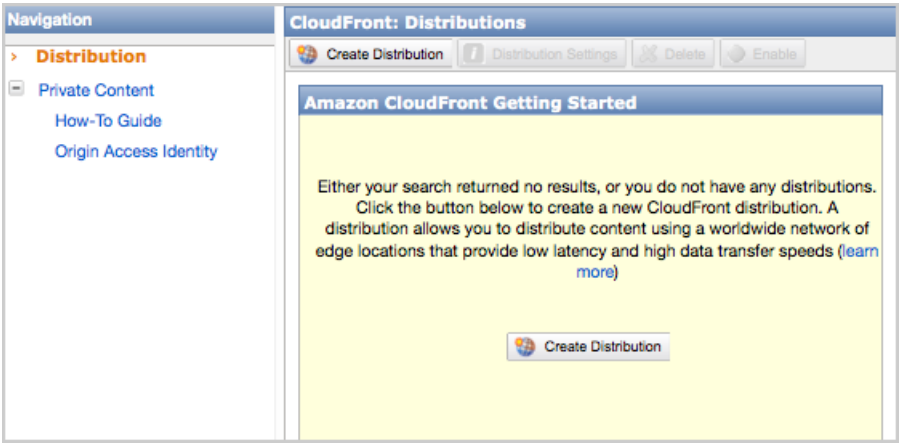
In this step, you will do the following:

- Create a CloudFront distribution that makes your website available from data centers around the world
- Update the A records in Amazon Route 53 to point to the CloudFront distribution

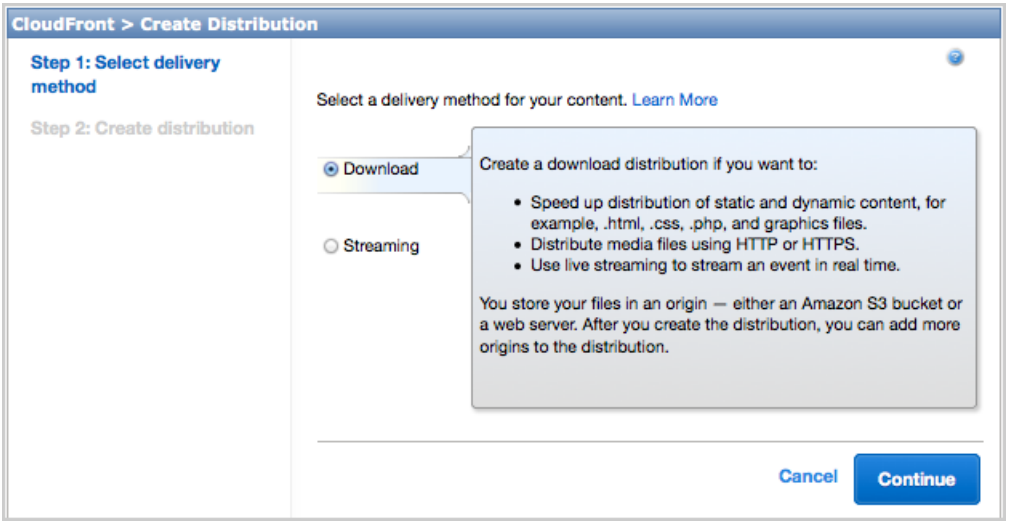
To create a distribution with an Amazon S3 origin, we will use the AWS Management Console.

To create a CloudFront distribution

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/.
2. In the CloudFront console, click **Create Distribution**.



3. On the **Step 1: Select delivery method** page, accept the default selection, **Web**, and then click **Continue**.



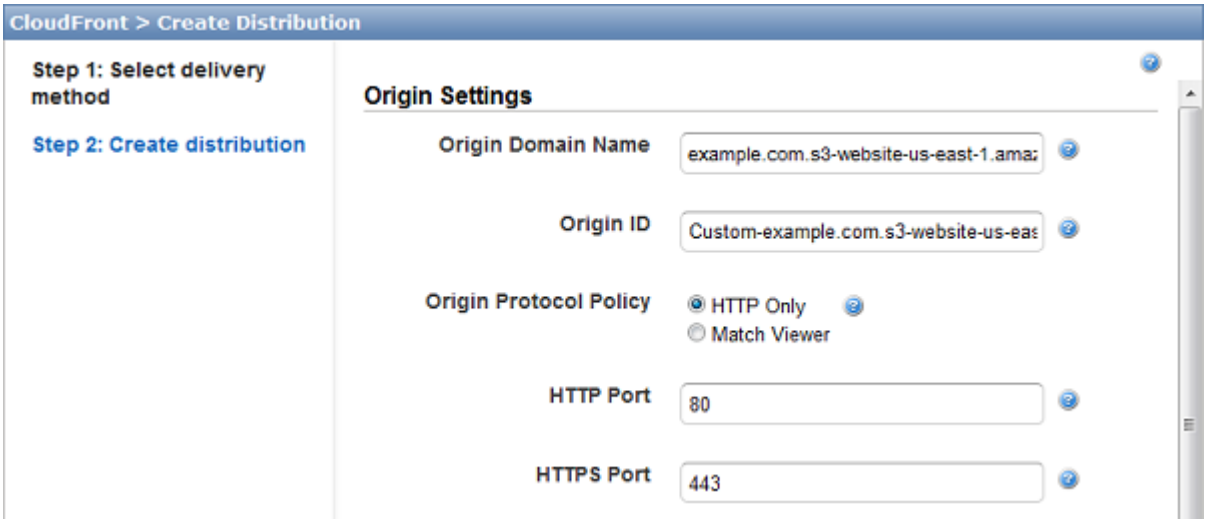
4. On the **Step 2: Create Distribution** page, do the following:

   - In the **Origin Domain Name** box, enter the Amazon S3 static website hosting endpoint for your bucket. This is the value that you copied in Step 4 of the procedure To configure your root domain Amazon S3 bucket as a website (p. 16), for example, **example.com.s3-website-us-east-1.amazonaws.com**.

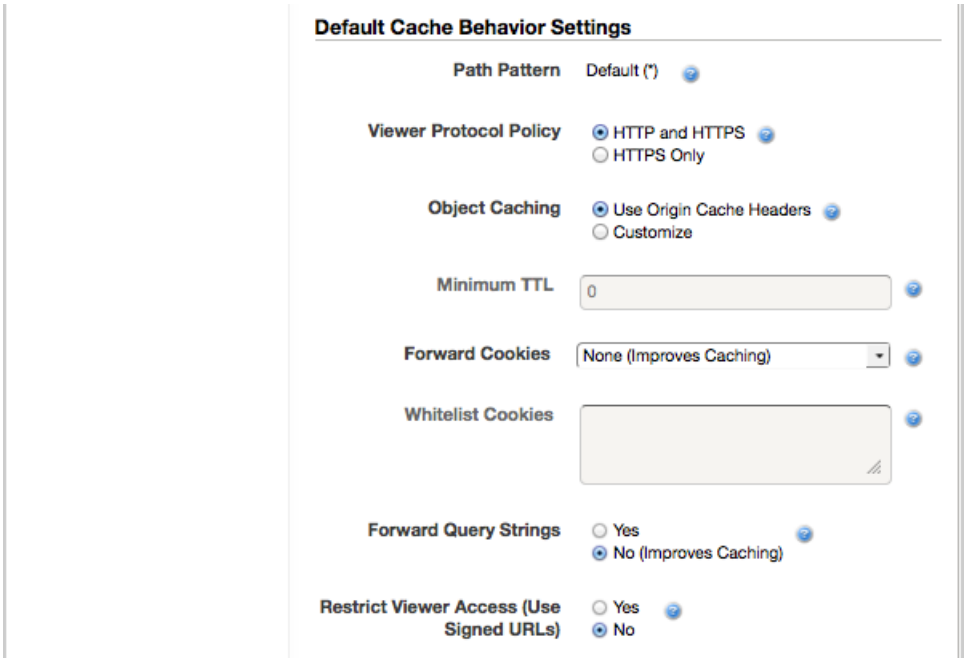
Important
Do not select the name of your bucket from the list, for example, **example.com.s3.amazonaws.com**

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

  Leave **Origin Protocol Policy**, **HTTP Port**, and **HTTPS Port** at their default settings.



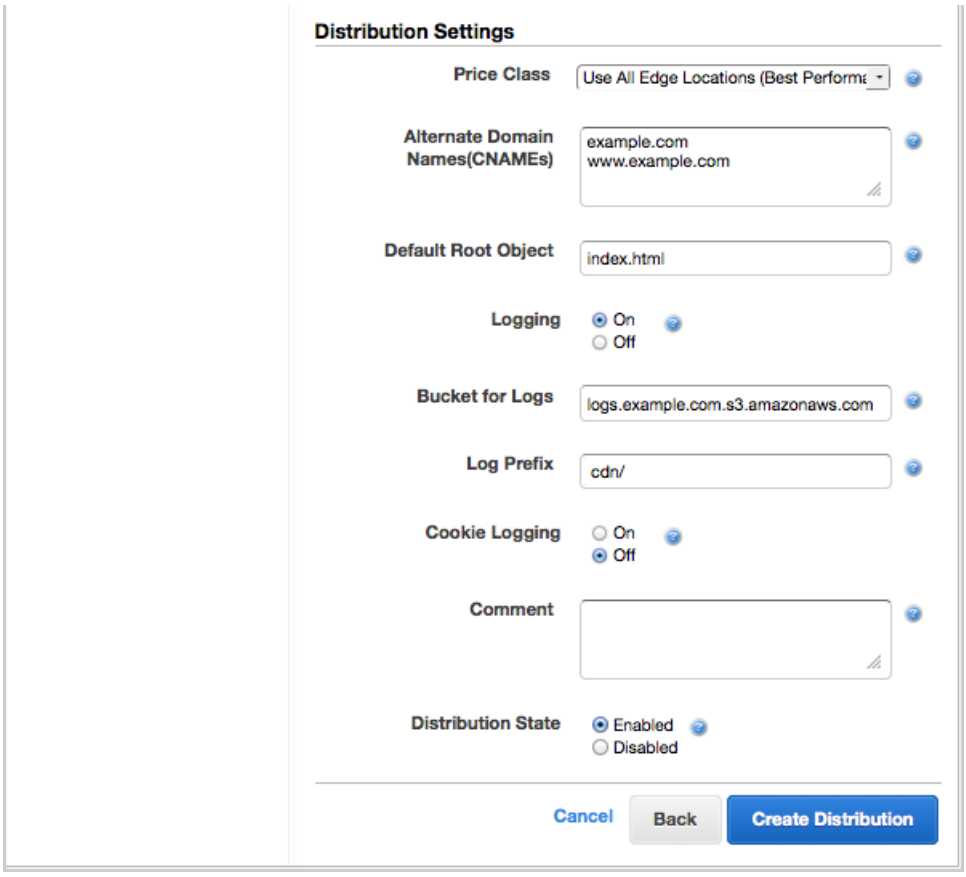
- Leave the values under **Default Cache Behavior Settings** at their default settings. For more information about these configuration options, see Values that You Specify When You Create or Update a Web Distribution in the *Amazon CloudFront Developer Guide*.



- Under **Distribution Settings**, leave **Price Class** set to **Use All Edge Locations (Best Performance)**. Set **Alternate Domain Names (CNAMEs)** to the root domain and www subdomain

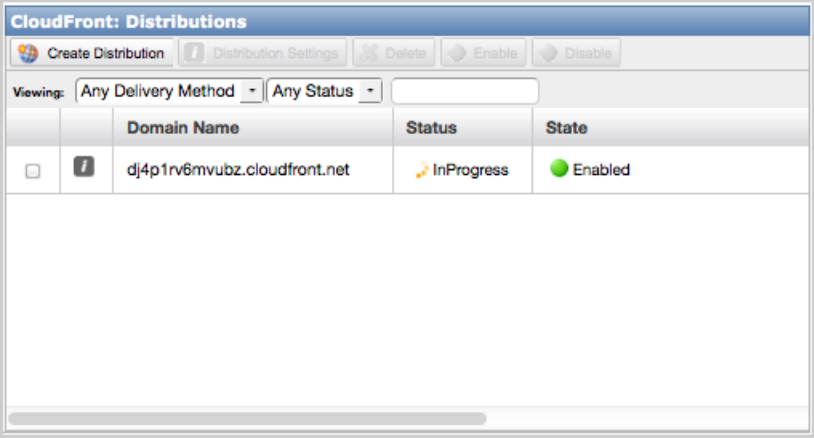
(in this guide, `example.com, www.example.com`). These values must be set in order to create A record aliases from the specified domain names to the CloudFront distribution.

- 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: Launch Your Website on Amazon S3 (p. 12).
- Set **Logging** to **On**.
- In **Bucket for Logs** select the logging bucket that you created in Step 3: Configure Storage on Amazon S3 (p. 7) (in this guide `logs.example.com`).
- Set **Log Prefix** to cdn/. This setting stores the logs generated by traffic to the CloudFront distribution to a folder `cdn` inside the log bucket.
- Leave **Cookie Logging** set to **Off**.
- Leave **Comment** blank.
- Leave **Distribution State** set to **Enabled**.



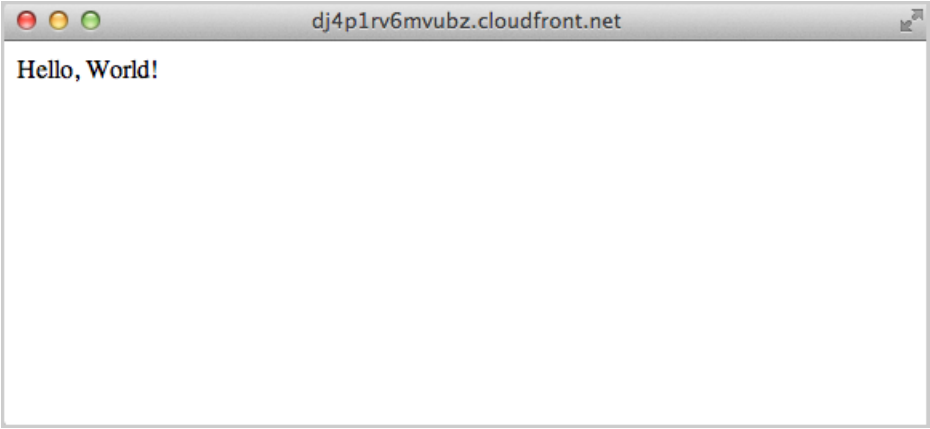
5. Click **Create Distribution**.

After you create the distribution, it may take up to 15 minutes to deploy for the distribution. The distribution's current status is displayed in the console under **Status**. A status of **InProgress** indicates that the distribution is not yet fully deployed.

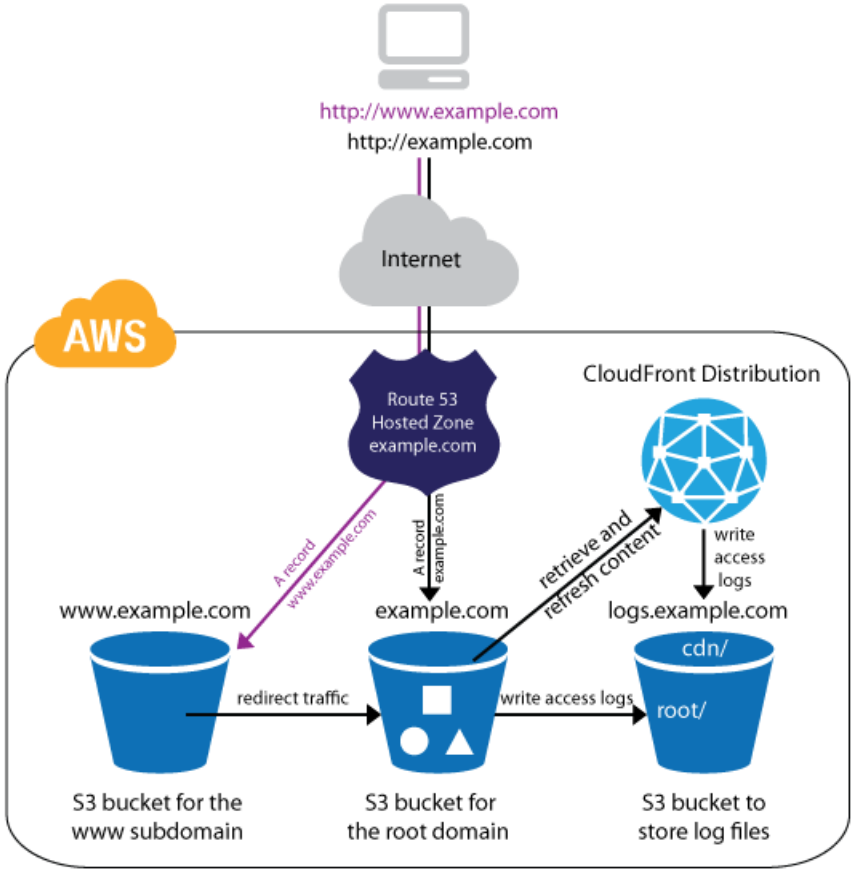


When your distribution is deployed, you are ready to reference your content with your new CloudFront domain name. Make a note of the value of **Domain Name** in the CloudFront console. You'll need this value in the next step. In this example, the value is `dj4p1rv6mvubz.cloudfront.net`.

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



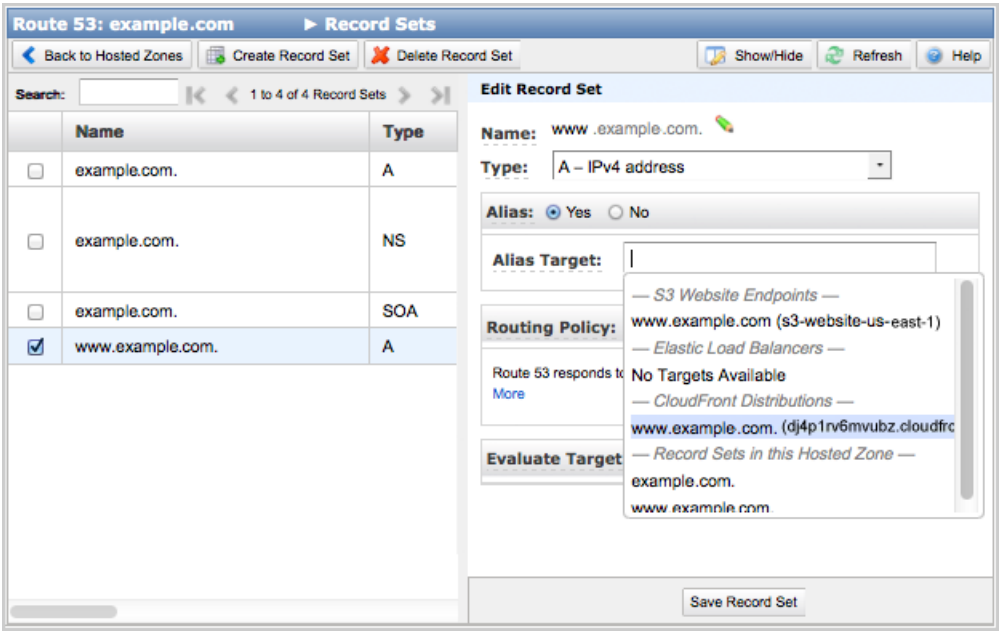
Your AWS infrastructure is currently configured as shown in the image below.



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 in Amazon Route 53 to point to a CloudFront distribution

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. Select the hosted zone that you created for your domain. In this guide, the zone is `example.com`.
3. Click **Go to Record Sets**.
4. In the list of records, select the check box that corresponds to the A record you created for the www subdomain. In this guide, the A record for the www subdomain is `www.example.com`.
5. Under **Alias Target**, select the CloudFront distribution.

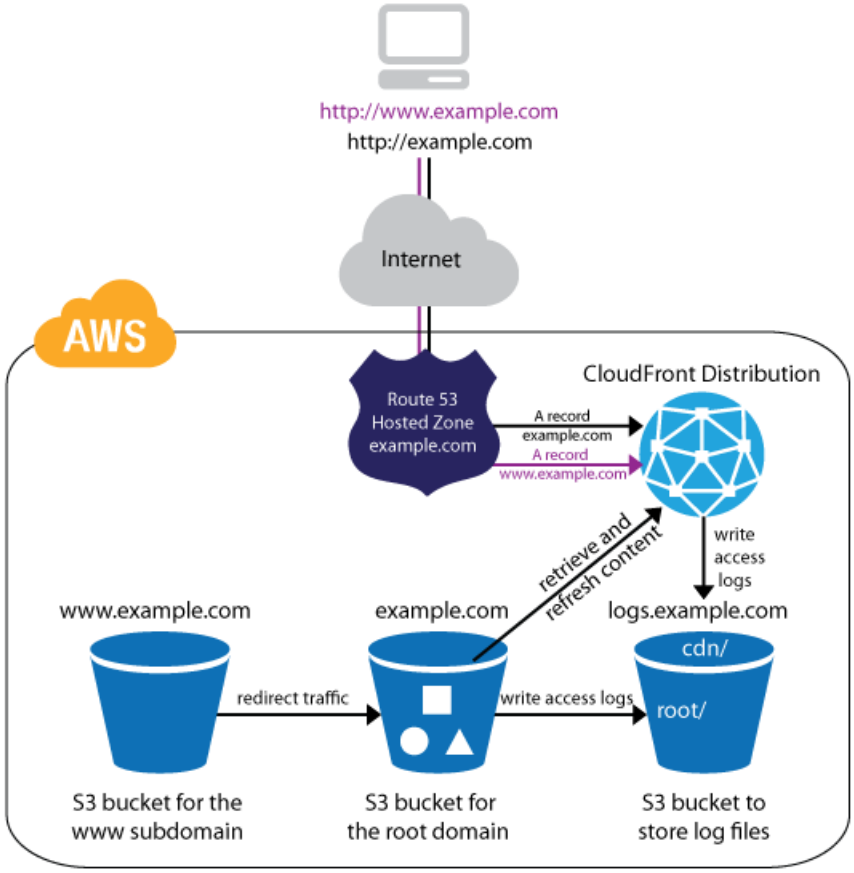


6. Click **Save Record Set**.
7. Repeat this procedure to redirect the A record for the root domain to the CloudFront distribution. In step 4, use `example.com` instead of `www.example.com`.

Wait from 2 to 48 hours for this change to take effect. 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.

> **Note**
> 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.)

Your AWS infrastructure is currently configured as shown in the image below.



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

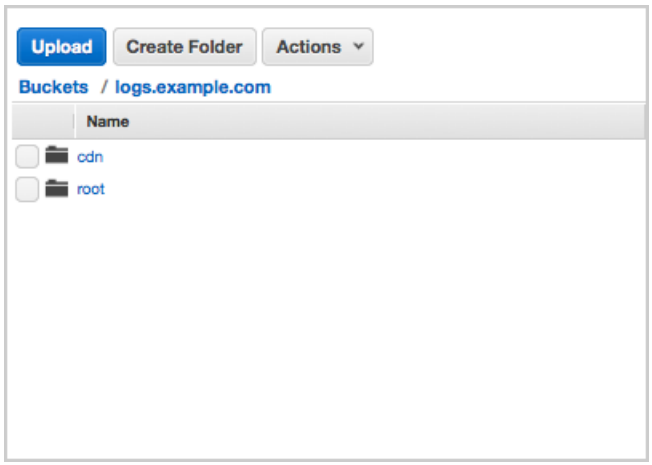
Congratulations! You have now improved the performance of your website for all visitors, whether they're accessing the site by using `http://www.example.com` or `http://example.com`.

The last thing to check on the site is that the access logs are being correctly written to Amazon S3. These files tell us how many people are visiting the site, and they contain valuable business data that you can analyze with other services, such as Amazon Elastic MapReduce (Amazon EMR).

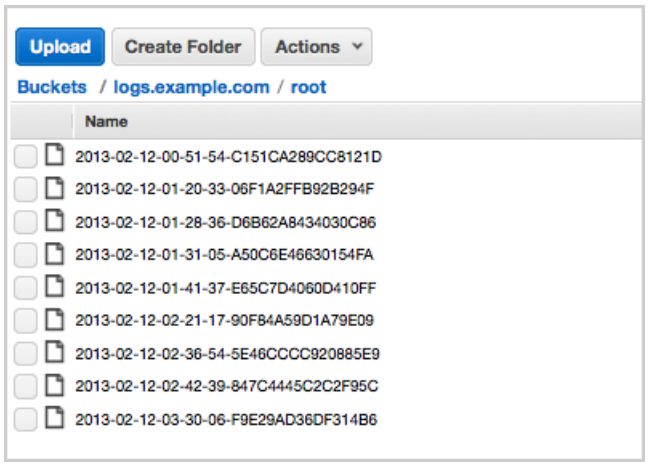
When you check the log files (in this guide, they are stored in the `logs.example.com` bucket) you should see older Amazon S3 log files in the folder `root`. All new log files should be CloudFront logs written in the folder `cdn`. Amazon S3 website access logs are written to your log bucket every 2 hours. 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. In the Amazon S3 console, click the logging bucket you created (in this guide, we used `logs.example.com`).
2. Click either `cdn` or `root` to view the log files stored within.



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



Where You're At

You're finished! You have created a static website hosted on Amazon S3, assigned a custom domain to the site using Amazon Route 53, and improved performance with a CloudFront distribution.

If the site you've created is one you want to keep, you're done. You may want to review Pricing (p. 38) to understand the ongoing costs associated with the site you've created.

If you created your site as a learning exercise only, it's time to delete the resources you allocated so that you no longer accrue charges. Continue on to Step 7: Clean Up (p. 35). After you delete your AWS resources, your website will no longer be available.

Step 7: Clean Up

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

Topics

Delete a Amazon Route 53 Hosted Zone

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

To delete A 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. In the dialog box that appears, 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. In the dialog box that appears, click **Confirm**.

   Amazon Route 53 deletes the hosted zone.

Disable and Delete a CloudFront Distribution

An Amazon CloudFront distribution can be disabled or deleted. A disabled distribution is no longer functional and you will no longer accrue charges, but you can enable it again at any time. A deleted distribution is no longer accessible and is lost forever.

You can only delete a CloudFront distribution when it is disabled. If you created a CloudFront distribution as part of this guide, you will need to first disable the distribution and then delete it.

To disable an CloudFront distribution

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

To delete a CloudFront distribution

1. Continuing from the previous procedure, right-click a disabled distribution, and then click **Delete**.
2. In the confirmation message that appears, click **Yes, Delete**.

Delete Objects and an Amazon S3 Bucket

Before you can delete an Amazon S3 bucket, all objects within the bucket must be deleted.

You should also ensure that logging for your Amazon S3 bucket is disabled; otherwise, logs might be immediately written to your bucket after you delete your bucket's objects.

To disable logging

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

To delete an object

1. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
2. In the **Buckets** pane, click the bucket where the objects are stored.
3. In the list of objects, right-click the object that you want to delete and then click **Delete**.

   A dialog box shows the actions you can take on the selected object(s).

   Tip
   You can use the **`SHIFT`** and **`CRTL`** keys to select multiple objects and perform the same action on them simultaneously.
4. In the confirmation message that appears, click **Yes, Delete**.

To delete a bucket, you must first delete all of the objects in it.

To delete a bucket

1. Continuing from the previous procedure, right-click the bucket you want to delete, and then click **Delete**.
2. In the confirmation message that appears, click **Yes, Delete**.

Pricing

Because AWS pricing is based on your actual usage of the service, we have to make some assumptions about the your static website's size and traffic in order to do a pricing estimate. For this pricing walk through, we used the following values.

- The site comprises 20 website files of about 50 KB each, for a total file size for the website of 1 MB.
- Files are updated twice a month.
- There is one custom domain name.
- Content is distributed to four edge locations: United States, Europe, Hong Kong and Singapore, and Japan.
- The site receives approximately 30,000 visits a month.

If you anticipate significantly different website traffic or file sizes, you can use the AWS Simple Monthly Calculator to do a more accurate pricing estimate. The AWS Simple Monthly Calculator estimates your monthly bill. It provides a per-service cost breakdown, as well as an aggregate monthly estimate. You can also use the calculator to see an estimate and breakdown of costs for common solutions.

The following topics walk you through an example of using the AWS Simple Monthly Calculator to estimate your monthly bill.

Topics

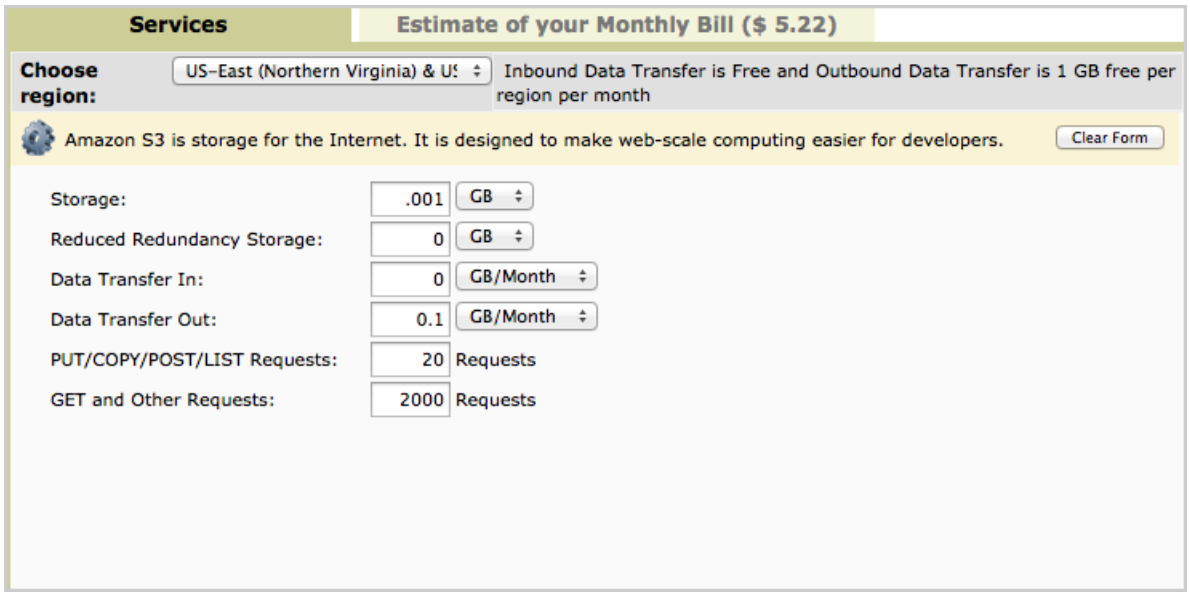
> **Note**
> AWS pricing that you see in this documentation is current at the time of publication. For the latest pricing information, go to AWS Service Pricing Overview. For more information on how AWS pricing works, download the PDF file How AWS Pricing Works.

Amazon S3 Cost Breakdown

The following table shows the characteristics for Amazon S3 we have identified for this static website hosting architecture.

Characteristic	Metric	Description
Storage	0.001 GB/month	20 JPEG files (objects) @ 50 KB = 1 MB Total of 1.0 MB for 20 objects = 0.001 GB
Requests	PUT requests: 20/month GET requests: 2000/month	We will plan to update the objects twice a month. 1 PUT request * 20 objects = 20 requests (PUT requests include from any location including Amazon EC2) We will transfer the objects 10 times per month to each of the 10 CloudFront edge locations. 10 GET requests * 10 CloudFront Nodes * 20 objects = 2000 requests (GET requests include from any location including Amazon EC2)
Data Transfer	Data out: 0.1 GB/month	If the average object size is 50 KB, and we make about 2000 requests per month, the average data transfer is approximately 0.1 GB.

The following image shows the cost breakdown for Amazon S3 in the AWS Simple Monthly Calculator.



Variable	Formula	Calculation
Provisioned Storage	Storage rate x Storage Amount (GB) ----------	$0.125 x 0.001 ------- $0.00
PUT Requests	Request Rate x Number of requests (per 1000) ----------	$0.01 x 1 ----- $0.01
GET Requests	Request Rate x Number of requests (per 10000) ----------	$0.01 x 1 ----- $0.01

We use the AWS Simple Monthly calculator to obtain this estimate. According to the calculator, the total cost for Amazon S3 is $0.02.

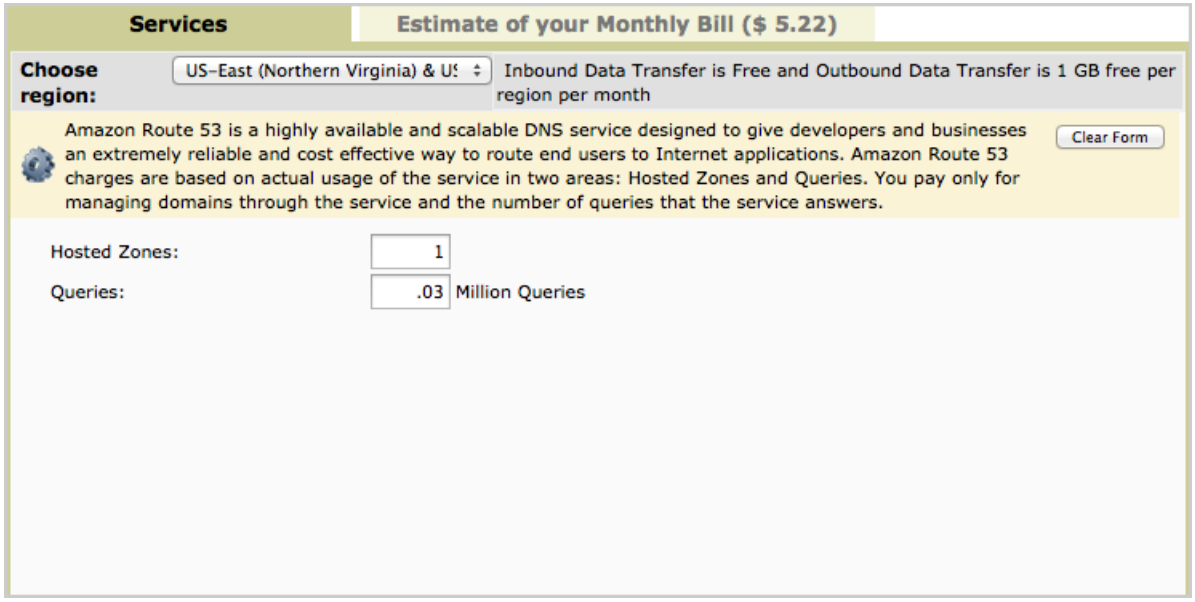
Amazon Route53 Cost Breakdown

The following table shows the characteristics for Amazon Route 53 we have identified for this web application hosting architecture.

Amazon Route 53 does not charge for queries made against aliases. Since you created A records pointing to aliases for both the root domain and the www subdomain, the only charge is for the domain hosting. Had you used a CNAME instead of an A record for the www subdomain, the monthly charge for .03 Million queries would have been $0.02 a month.

Characteristic	Metric	Description
Hosted Zones	1	Number of domain names hosted
Queries	30,000 / month	Number of DNS lookups handled by the name servers

The following image shows the cost breakdown for Amazon Route 53 in the AWS Simple Monthly Calculator.



The total monthly cost is the sum of the hosted zones and queries costs.

Variable	Formula	Calculation
Hosted Zones	Hosted Zones x Rate ----------	1 x 0.50 ------ $0.50
Queries	Monthly Queries (millions) x Rate ----------	.03 x 0.00 ------ $0.00

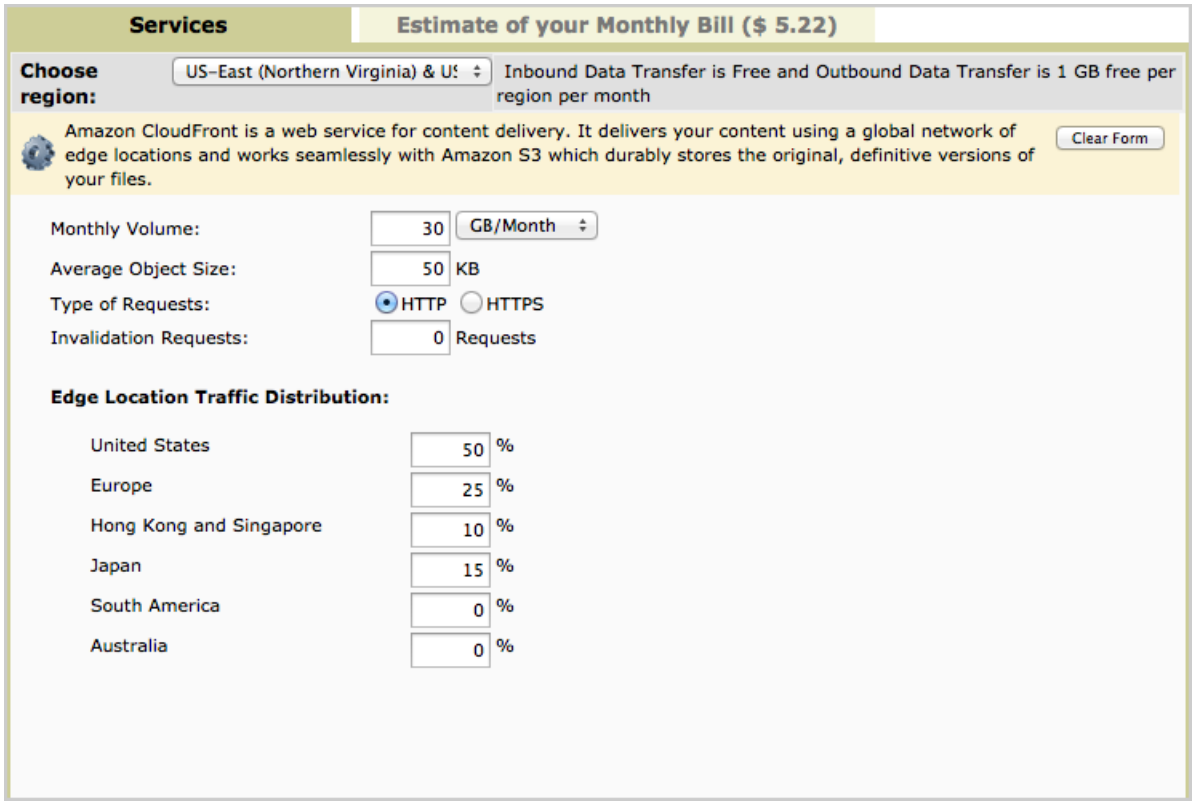
We use the AWS Simple Monthly calculator to obtain this estimate. With the calculator, the total cost for Amazon Route 53 is $0.50.

CloudFront Cost Breakdown

The following table shows the characteristics for CloudFront that we have identified for this web application hosting architecture.

Characteristic	Metric	Description
Traffic Distribution	50% US 25% EU 10% HK 15% JP 0% SA	Distribution of traffic across regions
Request Type	HTTP	Type of requests that customers make to the cached locations
Data Transfer Out	30.0 GB/month	1.0 MB x 30.5 days x 1,000 hits/day

The following image shows the cost breakdown for CloudFront in the AWS Simple Monthly Calculator.



The total monthly cost is the sum of the data transfer out plus the requests costs for each of the regions.

Variable	Formula	Calculation
Data Transfer Out for US	Monthly Volume (GB) Traffic Distribution (%) x Data Out Rate ----------	30.00 0.50 x 0.12 ------ $1.80
Data Transfer Out for EU	Monthly Volume (GB) Traffic Distribution (%) x Data Out Rate ----------	30.00 0.25 x 0.12 ------ $0.90
Data Transfer Out for HK/Singapore	Monthly Volume (GB) Traffic Distribution (%) x Data Out Rate ----------	30.00 0.10 x 0.19 ------ $0.57
Data Transfer Out for JP	Monthly Volume (GB) Traffic Distribution (%) x Data Out Rate ----------	30.00 0.15 x 0.201 ------- $0.90
Requests for US	Request Rate Traffic Distribution (%) x (Monthly Volume/Object Size (per 10,000 requests)) ----------	$0.0075 0.50 x (30 GB/50 KB/10K) ---------- $0.24
Requests for EU	Request Rate Traffic Distribution (%) x (Monthly Volume/Object Size (per 10,000 requests)) ----------	$0.009 0.25 x (30.00 GB/50 KB/10K) ---------- $0.15

Variable	Formula	Calculation
Requests for HK/Singapore	Request Rate Traffic Distribution (%) x (Monthly Volume/Object Size (per 10,000 requests)) ----------	$0.0075 0.10 x (30.00 GB/50 KB/10K) ---------- $0.05
Requests for JP	Request Rate Traffic Distribution (%) x (Monthly Volume/Object Size (per 10,000 requests)) ----------	$0.0095 0.15 x (30.00 GB/50 KB/10K) ---------- $0.09

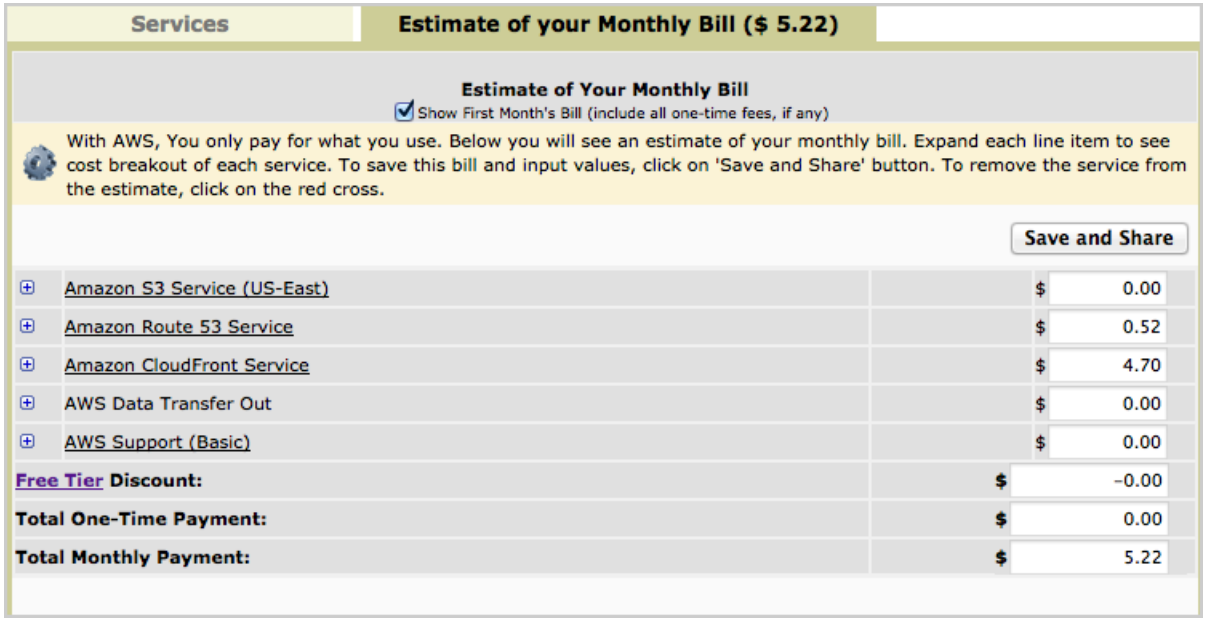
We use the AWS Simple Monthly calculator to obtain this estimate. According to the calculator, the total cost for CloudFront is $4.70.

Summing It All Up

To calculate the total cost for this example, we add the cost for Amazon S3, Amazon Route 53, and CloudFront. If you are eligible for the AWS Free Usage Tier, you can subtract the costs of Amazon S3 and CloudFront from this amount.

The total AWS data transfer out represents aggregate usage across Amazon S3 and any other Amazon services you might use. For Amazon S3, we have 0.1 GB per month. Since up to 1 GB per month of data transferred out is free, there is no charge.

The following image shows an example of your monthly estimate.



Services | Estimate of your Monthly Bill ($ 5.22)

Estimate of Your Monthly Bill

Show First Month's Bill (include all one-time fees, if any)

With AWS, You only pay for what you use. Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, click on the red cross.

Save and Share

⊞	Amazon S3 Service (US-East)	$ 0.00
⊞	Amazon Route 53 Service	$ 0.52
⊞	Amazon CloudFront Service	$ 4.70
⊞	AWS Data Transfer Out	$ 0.00
⊞	AWS Support (Basic)	$ 0.00
Free Tier Discount:		$ -0.00
Total One-Time Payment:		$ 0.00
Total Monthly Payment:		$ 5.22

The total cost of this static website is estimated at $5.22 per month.

Related Resources

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

Resource	Description
AWS Products and Services	A comprehensive list of products and services AWS offers.
Documentation	Official documentation for each AWS product including service introductions, service features, and API references, and other useful information.
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 Economics Center	Provides access to information, tools, and resources to compare the costs of Amazon Web Services with IT infrastructure alternatives.
AWS Cloud Computing Whitepapers	Features a comprehensive list of technical AWS whitepapers covering topics such as architecture, security, and economics. These whitepapers have been authored either by the Amazon team or by AWS customers or solution providers.
Videos and Webinars	Previously recorded webinars and videos about products, architecture, security, and more.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and AWS Premium Support. (subscription required).
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.

Resource	Description
Form for questions related to your AWS account: Contact Us	This form is *only* for account questions. For technical questions, use the Discussion Forums.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Document History

The following table describes the important changes to the documentation since the last release of *Getting Started with AWS AWS Static Website Hosting*.

Latest documentation update: June 14, 2013

Change	Description	Release Date
Custom Domain	Added instructions for associating a custom domain name with the static website.	June 14, 2013
CloudFront	Added instructions for creating a CloudFront distribution.	May 17, 2012
New content	Initial release of this document.	October 17, 2011