Experiment No. : 10

Title: Launch an Web Application with AWS Elastic Beanstalk

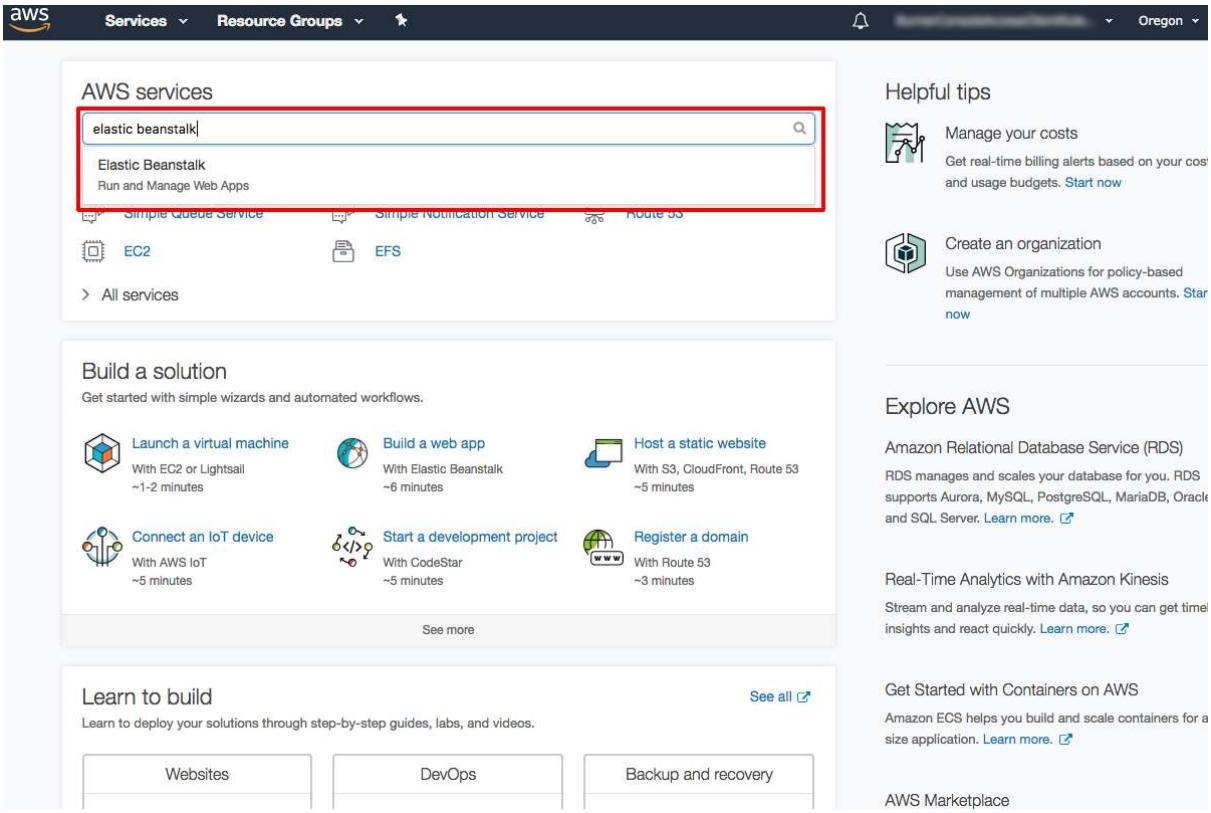
1. Sign-up for AWS

This step-by-step guide will help you get a sample PHP application up and running with AWS Elastic Beanstalk (EB). EB supports other languages besides PHP, such as Java, .NET, Node.JS, Python, Ruby, Docker, and Go, but the focus of this tutorial will be on PHP (other languages will follow the same process). You will first configure your EB application, then setup your EB environment where your application will be launched into.

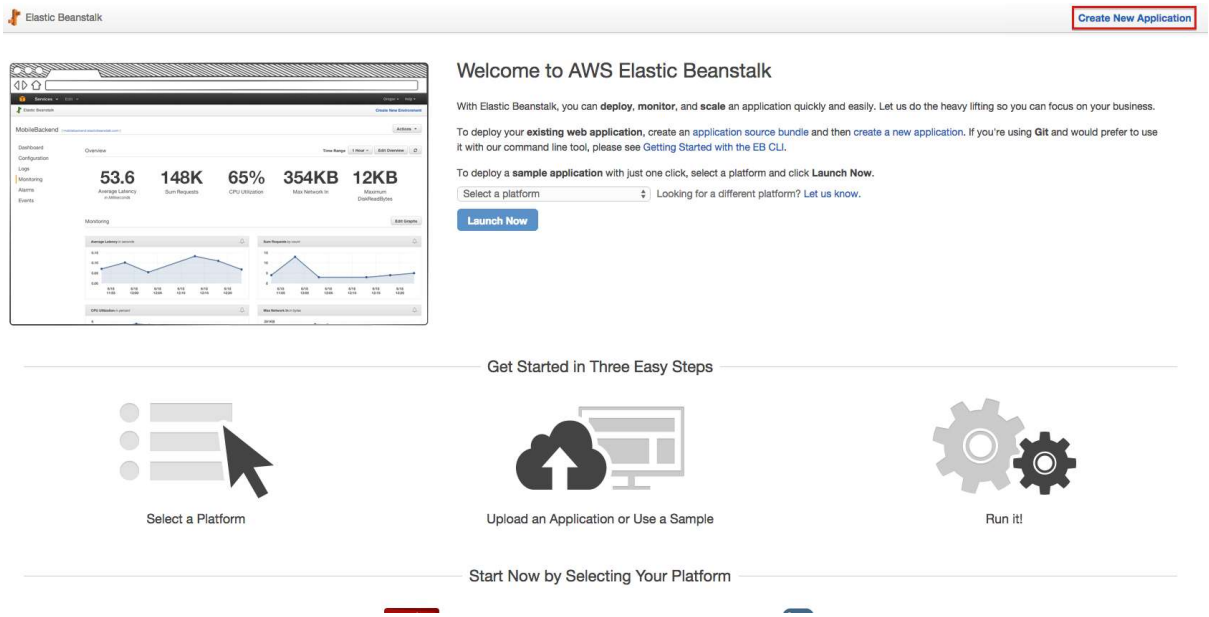
> Did you know? AWS made it even easier to launch a web application. Jumpstart your application with Amazon Lightsail >>

In this, we will be using a pre-built sample PHP application. To download this sample PHP application file, please click here.

When you click here, the AWS management console will open in a new browser window, so you can keep this step-by-step guide open. When this screen loads, enter your user name and password to get started. Then type in "elastic beanstalk" in the search bar and press Enter.

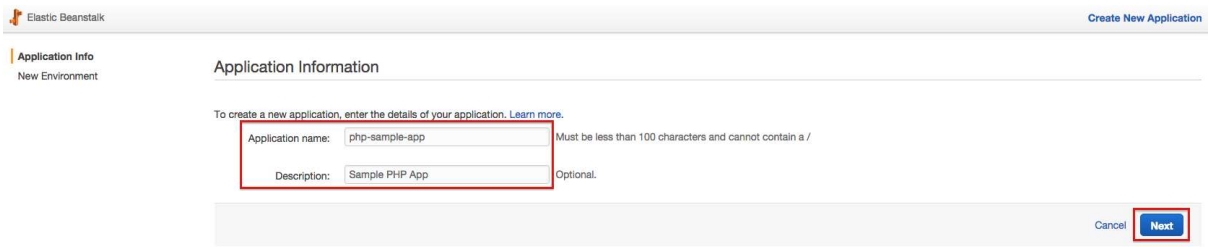


Step 1: Create a New Application
Now that you're in the AWS Elastic Beanstalk dashboard, click on Create New Application to create and configure your application.



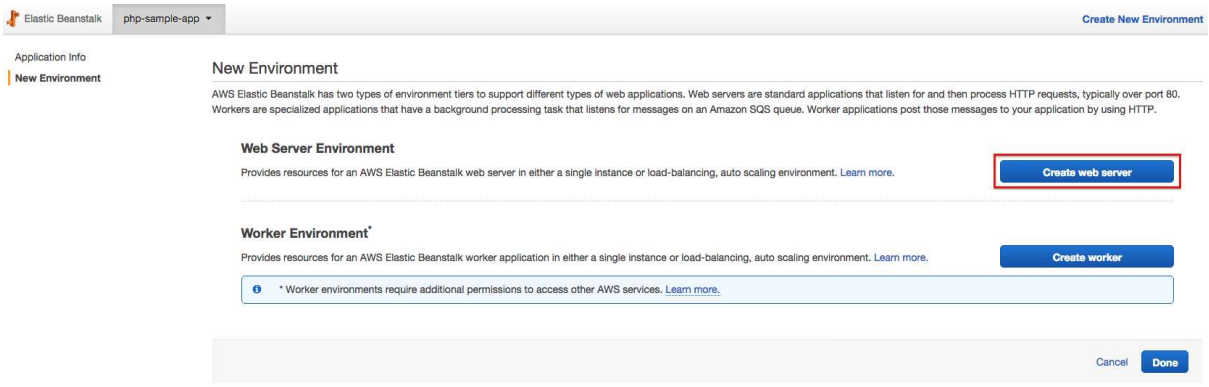
Step 2: Configure your Application

Fill out the *Application name* with php-sample-app and *Description* field with Sample PHP App. Click Next to continue.



Step 3: Configure your Environment

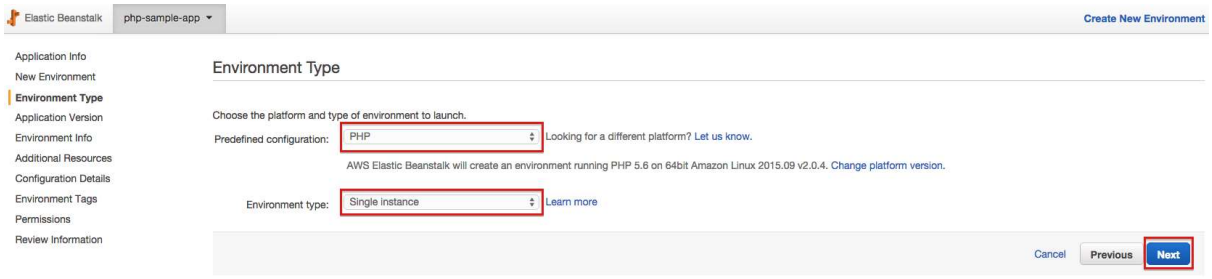
a. For this tutorial, we will be creating a web server environment for our sample PHP application. Click on Create web server.



b. Click on Select a platform next to *Predefined configuration*, then select PHP. Next, click on the drop-down menu next to *Environment type*, then select Single instance.

> Note: an "instance" is referring to Amazon's Elastic Compute Cloud (EC2) compute service. A "single instance" means we will be using one virtual server to deploy our application into.

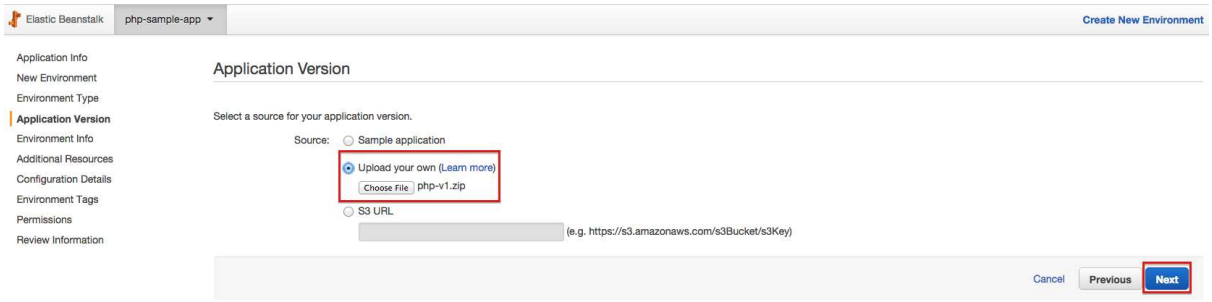
We will discuss how to scale and load balance your application in a separate tutorial.
Click Next to continue.



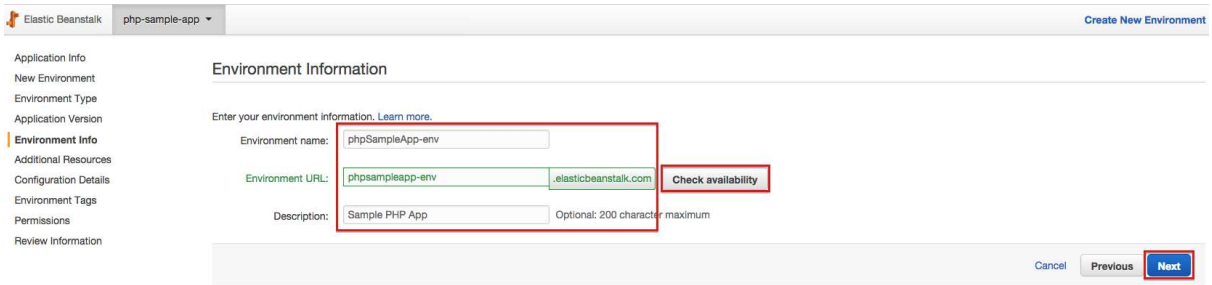
c. Under *Source*, select the Upload your own option, then click Choose File to select the sample php-v1.zip file we downloaded earlier.

Before moving on, double click on the php-v1.zip file that you downloaded to your local machine to view the contents within. This will help you better understand what your zip file should look like when working with your own PHP application. PHP does not enforce a strict file structure for applications; flat file structure will work fine.

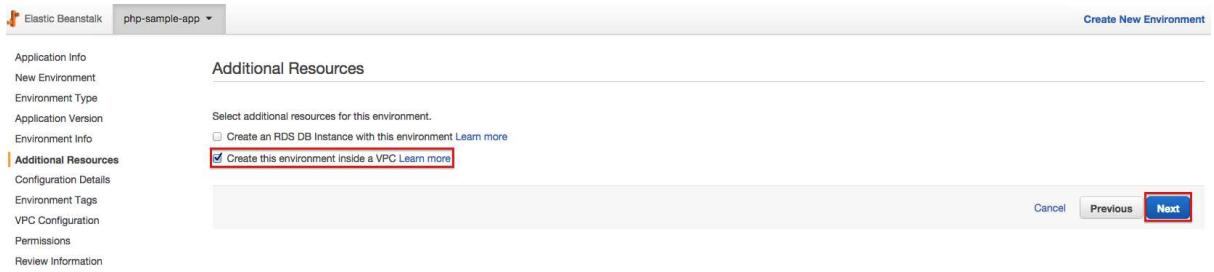
Click Next to continue.



d. Fill in the values for *Environment name* with phpSampleApp-env. For *Environment URL*, fill in a globally unique value since this will be your public-facing URL; we will use phpsampleapp-env in this tutorial, so please choose something different from this one. Lastly, fill *Description* with Sample PHP App. For the *Environment URL*, make sure to click Check availability to make sure that the URL is not taken. Click Next to continue.



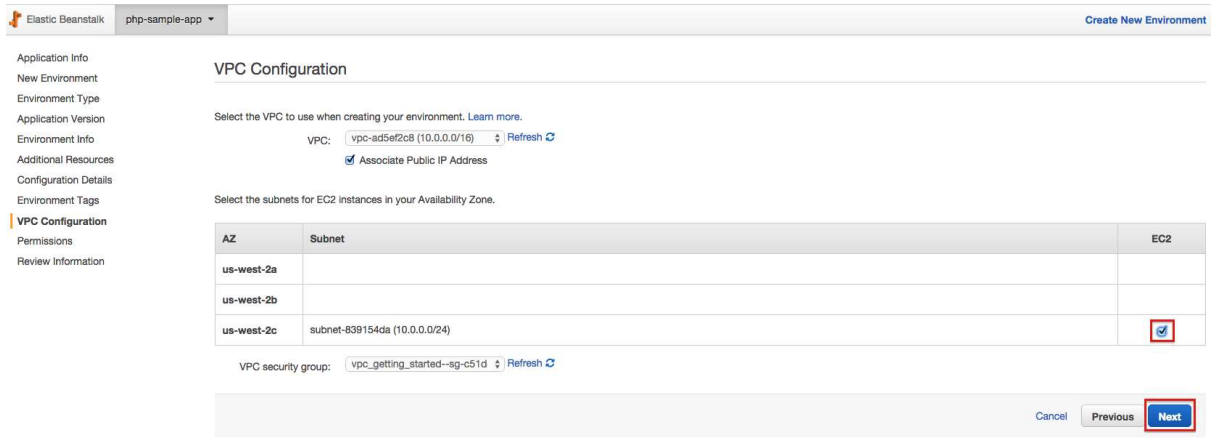
e. Check the box next to *Create this environment inside a VPC*. Click Next to continue.



f. On the Configuration Details step, you can set configuration options for the instances in your stack. For this tutorial, you don't need to change anything. Click Next.

On the Environment Tags step, you can tag all the resources in your stack. For this tutorial, you don't need to tag any resources but can if you would like. Click Next.

On the VPC Configuration step, select the first AZ listed by checking the box under the EC2 column. Your list of AZs may look different than the one shown as Regions can have different number of AZs. Click Next

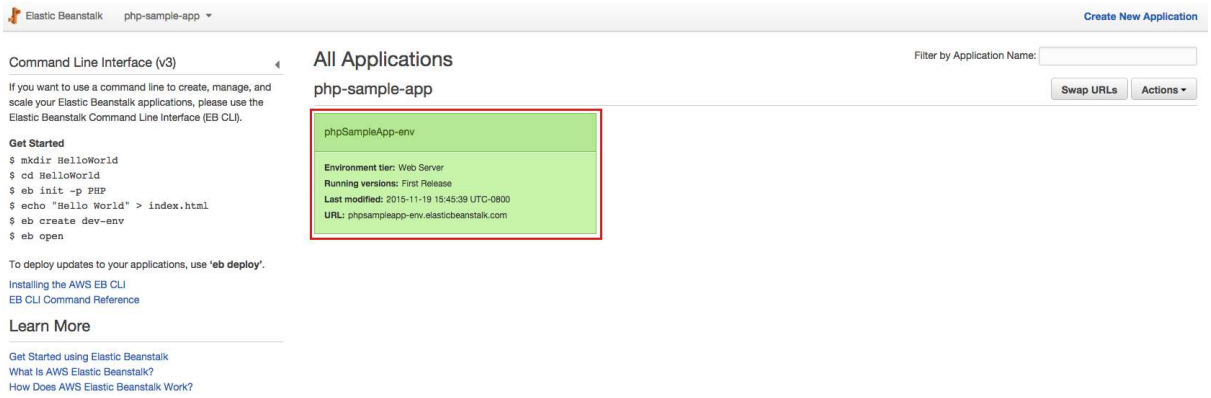


g. At the *Permissions* step, leave everything to their default values, then click Next to continue. Then review your *environment configuration* on the next screen and then click Launch to deploy your application.

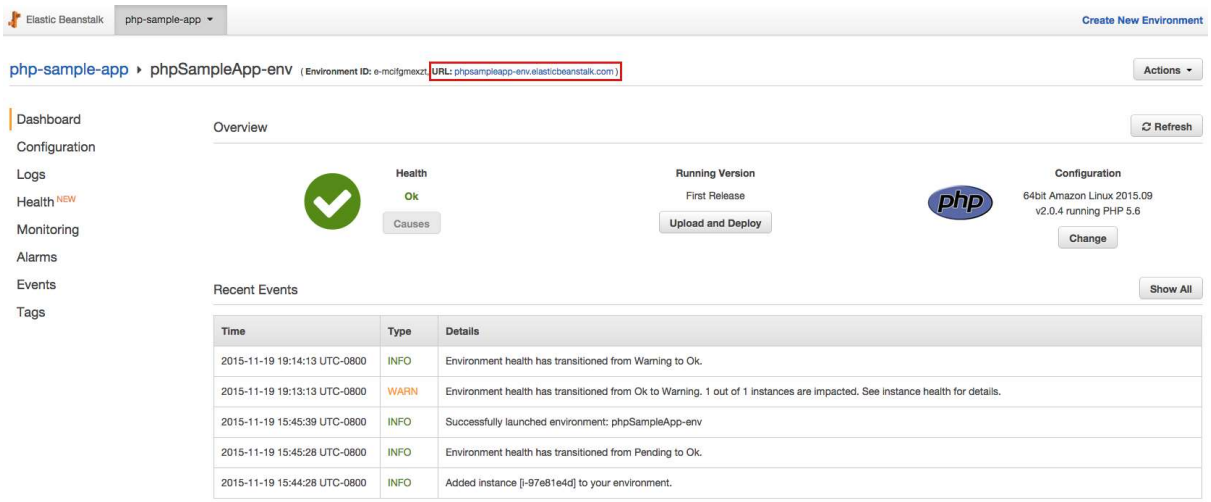
Note: Launching your application may take a few minutes.

Step 4: Accessing your Elastic Beanstalk Application

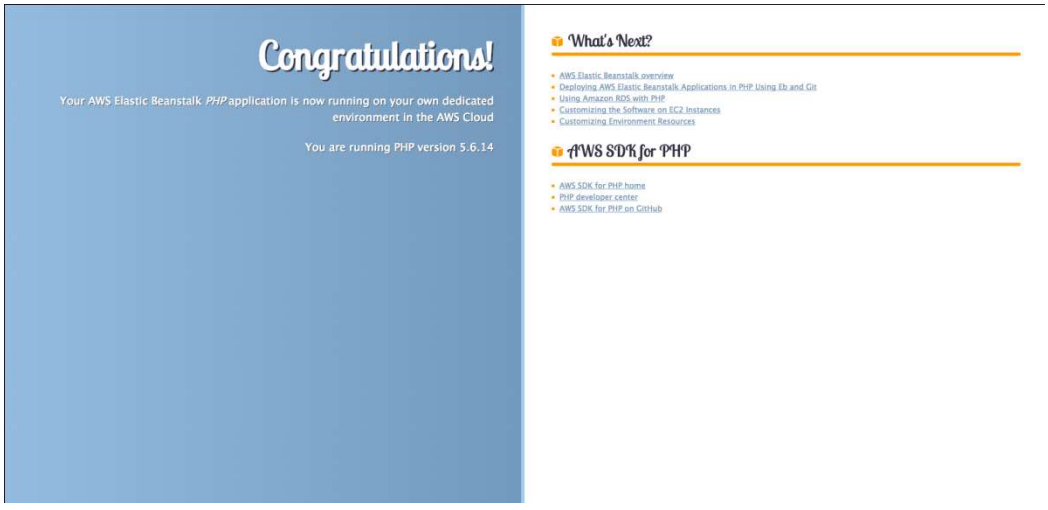
a. Go back to the main Elastic Beanstalk dashboard page by clicking on Elastic Beanstalk. When your application successfully launched, your application's environment, *phpSampleApp-env*, will show up as a green box. Click on phpSample-App-env, which is the green box.



b. At the top of the page, you should see a *URL* field, with a value that contains the *Environment URL* you specified in step 3 part d. Click on this URL field, and you should see a *Congratulations page*.



Congratulations! You have successfully launched a sample PHP application using AWS Elastic Beanstalk.



Now that you have an Elastic Beanstalk application up and running, the next part will walk you through updating your application.

Update an Application with AWS Elastic Beanstalk

This will cover how to update your existing application and then how to delete your Elastic Beanstalk environment, which includes your application. This tutorial is a continuation from the Launch an Application with AWS Elastic Beanstalk tutorial, so please go through that tutorial first if you haven't already. This tutorial will cover how to update your existing application and then how to get rid of your Elastic Beanstalk environment, which includes your application.

Step 1: Update Your Application Code

a. Navigate to the directory where you downloaded the php_v1.zip file in the previous tutorial.

Windows Users: To unzip the php-v1.zip file, right click on the *php-v1.zip* file, click Extract All..., then click Extract.

Mac Users: Double click on the *php-v1.zip* file, and this will automatically unzip the file into a *php-v1* folder in the same directory.

b. Navigate into the newly unzipped *php-v1* directory. Open index.php using your favorite text editor. You will make a small edit here that is an example of an application change. Look on line 26 for *<h1>Congratulations!</h1>*. Replace *Congratulations!*, with Application Updated! in between the <h1> and </h1> tags. Then save the *index.php* file (overwriting the original).

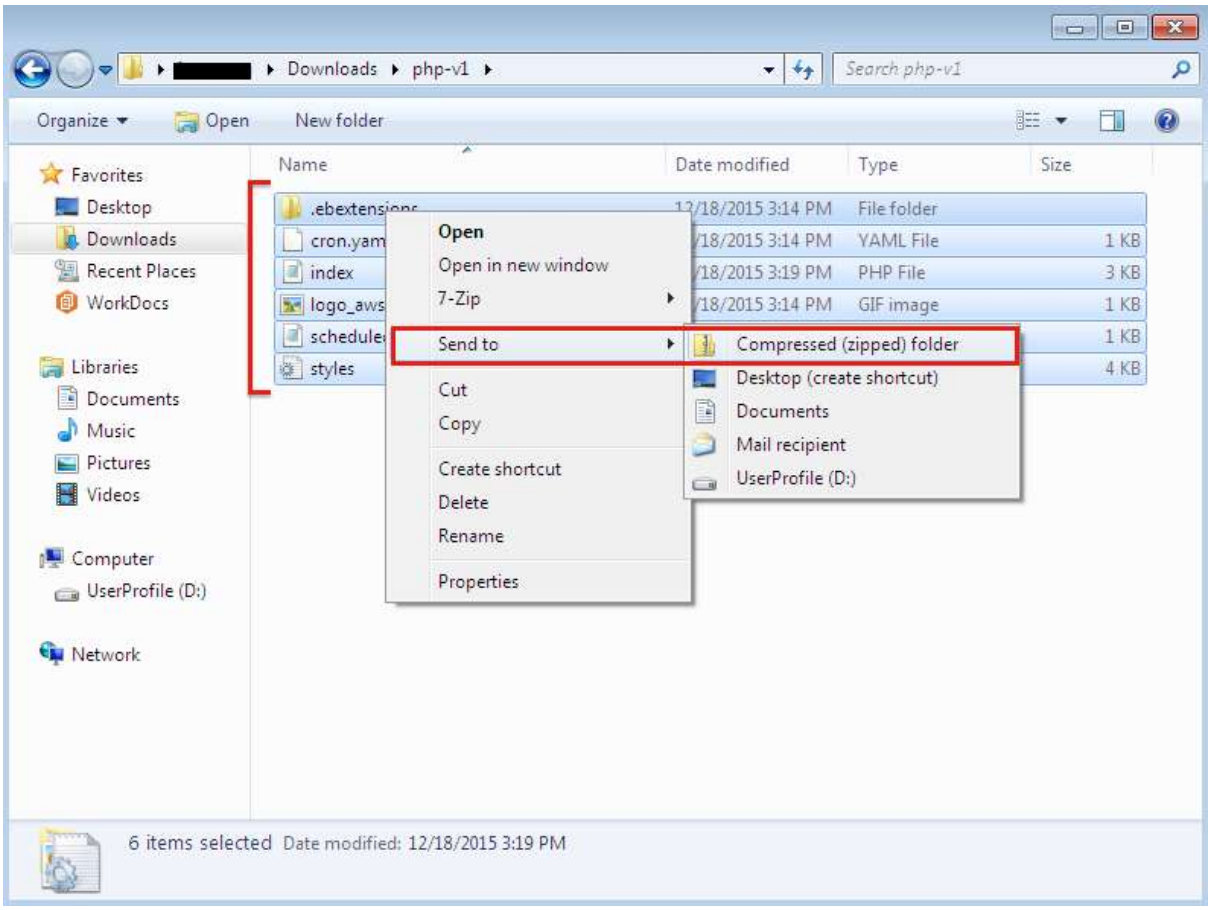
```
24  <body>
25      <section class="congratulations">
26          <h1>Application Updated!</h1>
27          <p>Your AWS Elastic Beanstalk <em>PHP</em> application is now running on your own dedicated environment in the AWS Cloud</p>
28          <p>You are running PHP version <?= phpversion() ?></p>
29      </section>
```

Next you will need to zip your application so it can be uploaded to AWS as an update package.
Windows users: Please select Windows below to see how to create the zipped application file.
Mac & Linux users: Please select Mac/Linux below to see how to create the zipped application file.

- Windows

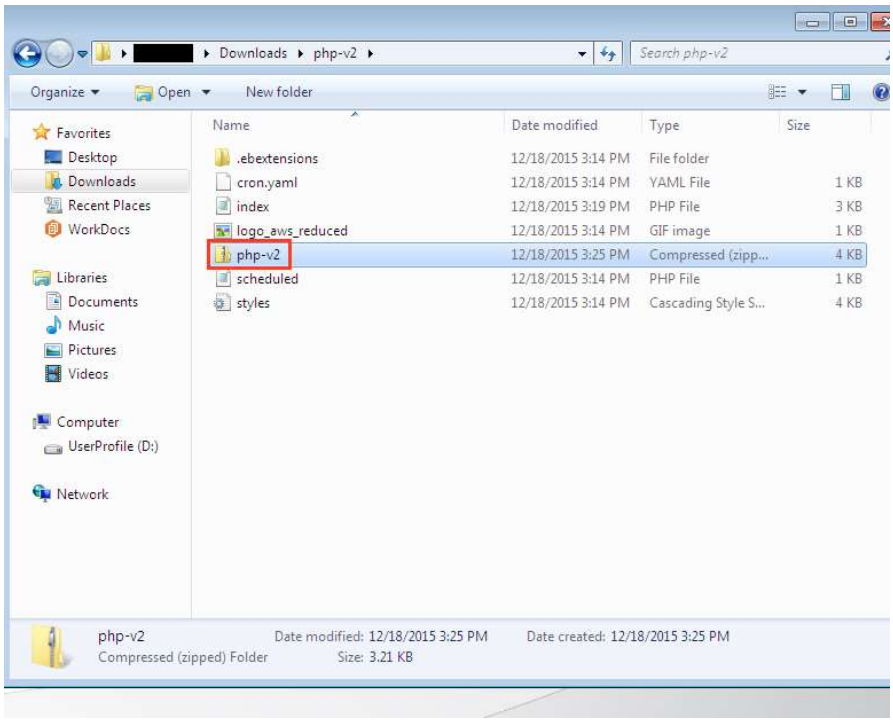
- Mac/Linux

- c. Select all 6 items (including the *.ebextensions* directory), right-click on *.ebextensions*, select Send to, and then click on Compressed (zipped) folder.



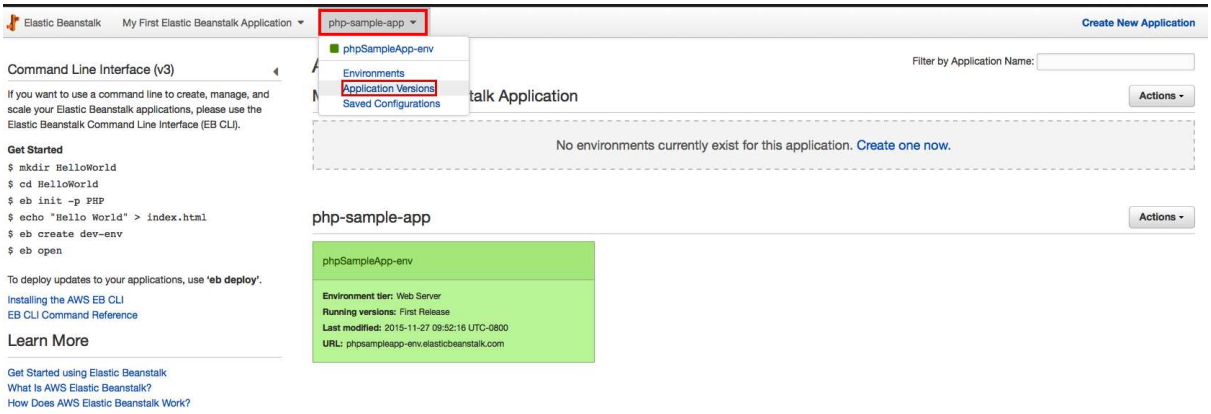
d. Rename the newly created zip file to php-v2.zip.

Note: On some Windows systems the *.zip* part of the file name may be hidden (see the example image).

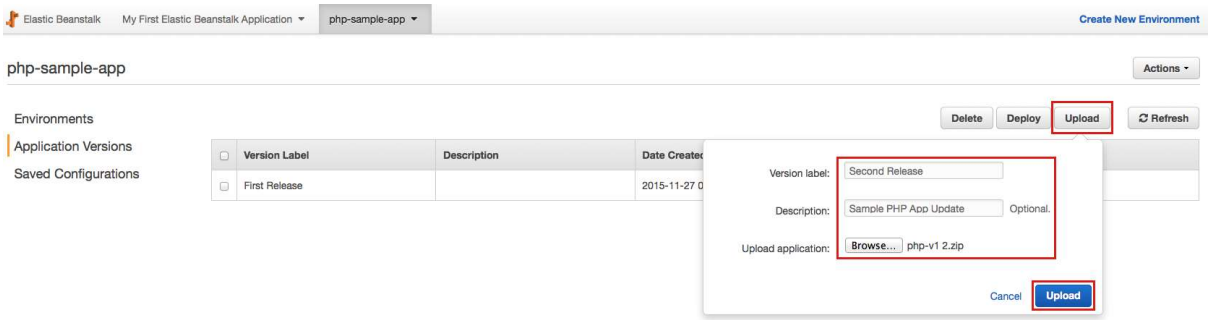


Step 2: Upload Your Updated Application to Elastic Beanstalk

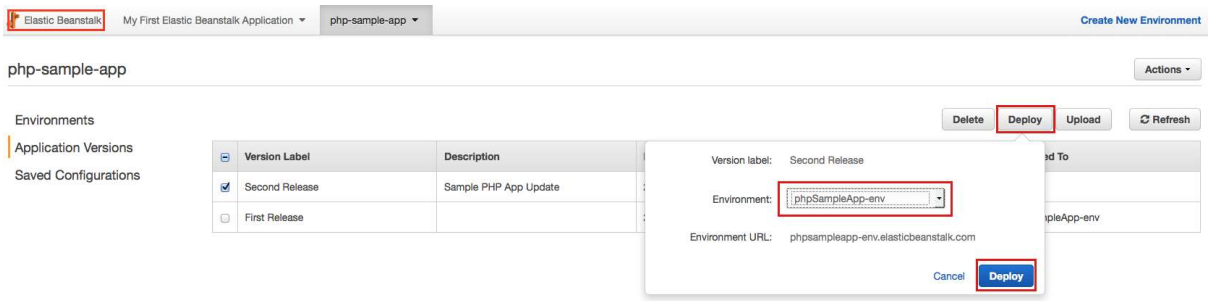
a. Click here to open the Elastic Beanstalk console. Within the Elastic Beanstalk dashboard, click on php-sample-app at the top of your screen, and this should show a drop-down menu where you should select Application Versions.



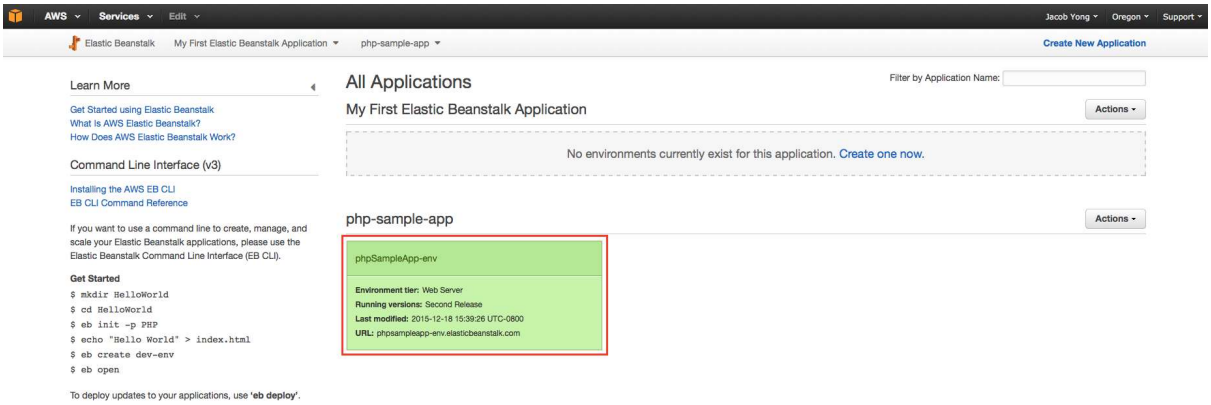
b. Here you should see one entry in the *Version Label* column titled *First Release*. The *Source* column for this entry should show the *php-v1.zip* file we uploaded in the previous tutorial. Click on Upload, enter Second Release for *Version* label, then Sample PHP App Update for *Description*. Click Browse, then navigate to the location where your *php-v2.zip* file is located, Select the php-v2.zip file and click Upload.



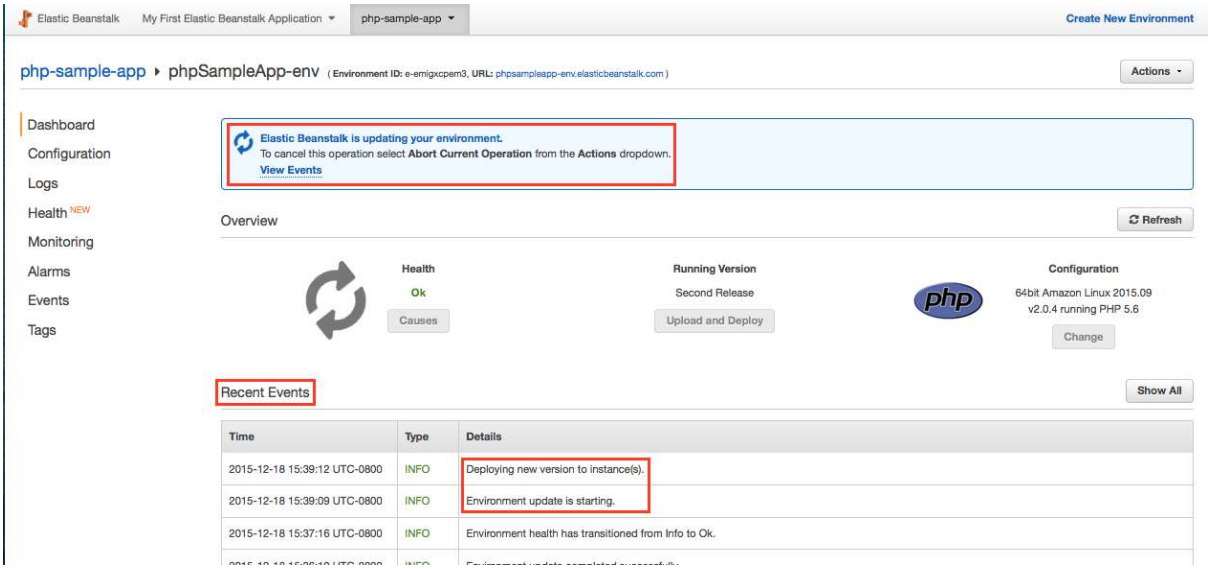
c. You should now see *Second Release* within the application versions table. Check the box for *Second Release*, then click Deploy. You should see your *Environment* defaulted to *phpSampleApp-env*. Leave the default settings here and click Deploy. Lastly, click Elastic Beanstalk in the top left hand corner of the web page.



d. Click on the green box titled *phpSampleApp-env* to see the view of your application's environment.

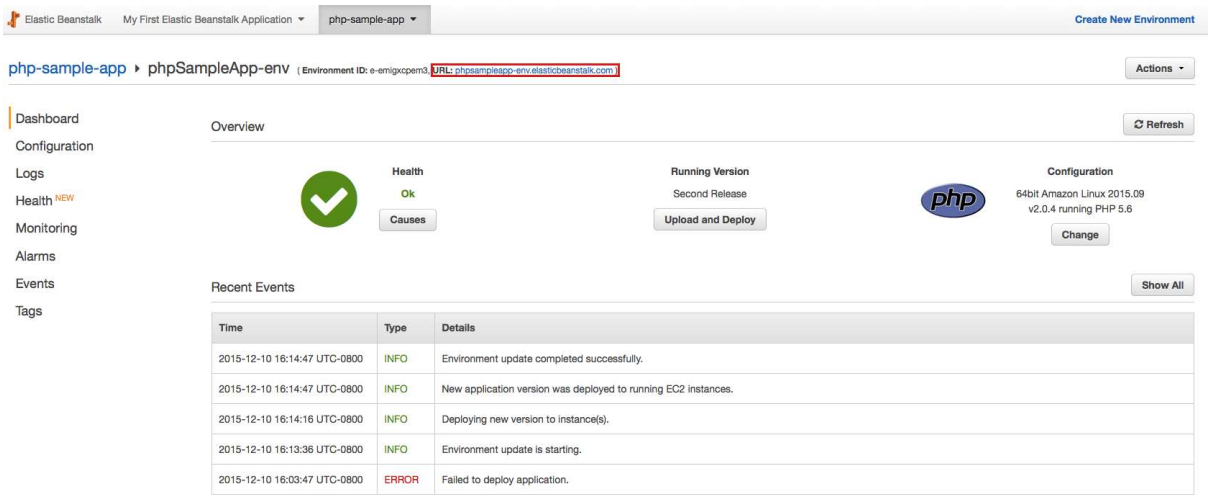


e. Here, you can see a *Recent Events* section that displays your application being updated.



Step 3: Accessing Your Updated Elastic Beanstalk Application

a. Once you see *Environment update completed successfully* within *Recent Events*, click on your application URL toward the center top of the screen to view your updated application.

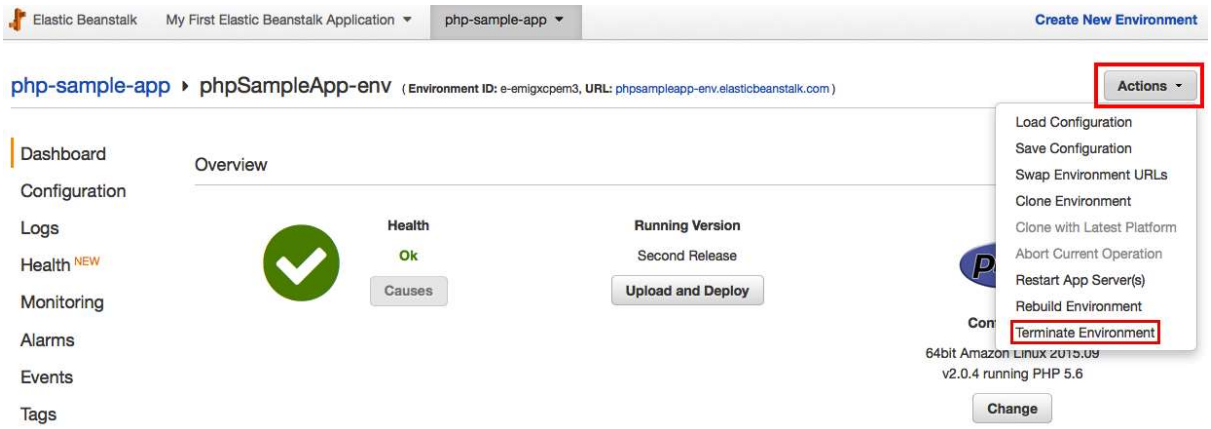


b. You will see that instead of the *Congratulations!* text that existed in version 1 of your application the text has been updated to version 2 with the heading *Application Updated!*

Congratulations! You have successfully updated your AWS Elastic Beanstalk application.

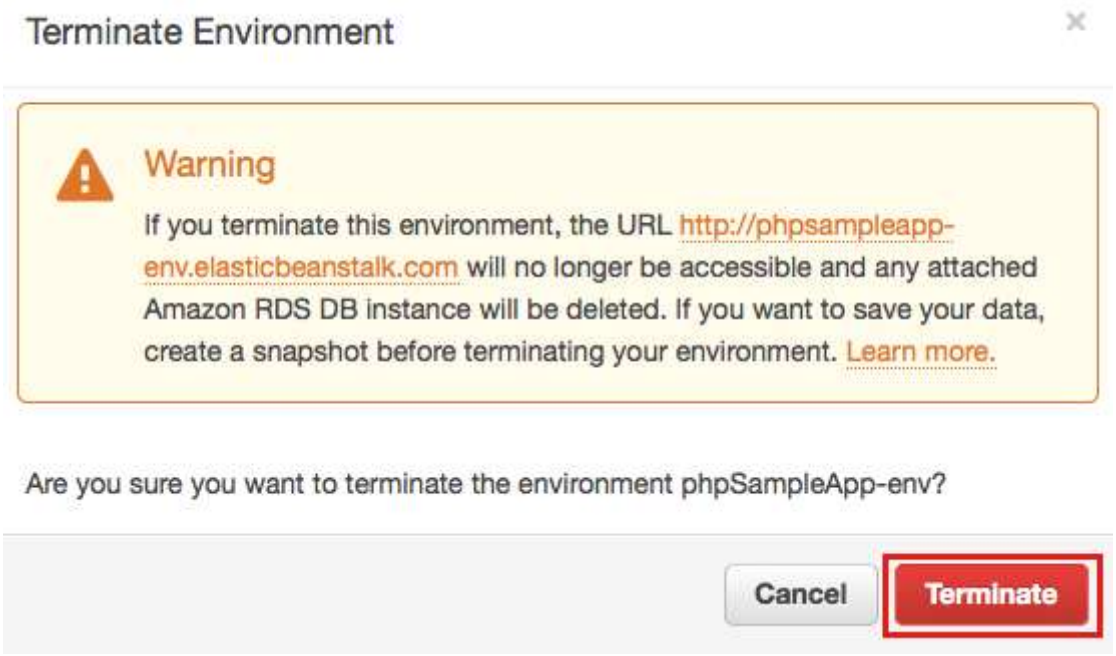
Step 4: Terminate Your Environment

a. To delete your application (and stop using the AWS resources associated with your application), access your Elastic Beanstalk application dashboard, click on Actions in the top right-hand corner, then select Terminate Environment.



b. You will be presented with a warning/confirmation screen. Click Terminate to continue.

Note: It may take a couple of minutes for the environment to completely shut down.



Next Steps

Now that you have an Elastic Beanstalk application up and running, the next part will walk you through registering a domain name so your website/application can be easy to access.

Register a Domain Name with Amazon Route 53

In this tutorial you will register a new domain name for your website. You will then connect that domain name through the Domain Name System (DNS) to a currently running EC2 instance (such as a WebApp, or website running WordPress, Apache, NGINX, IIS, or other

Website platform). If you already have a domain name registered, do step 1 and then refer to your domain registrar's documentation for how to set the DNS record for your new site.

Cost implications:

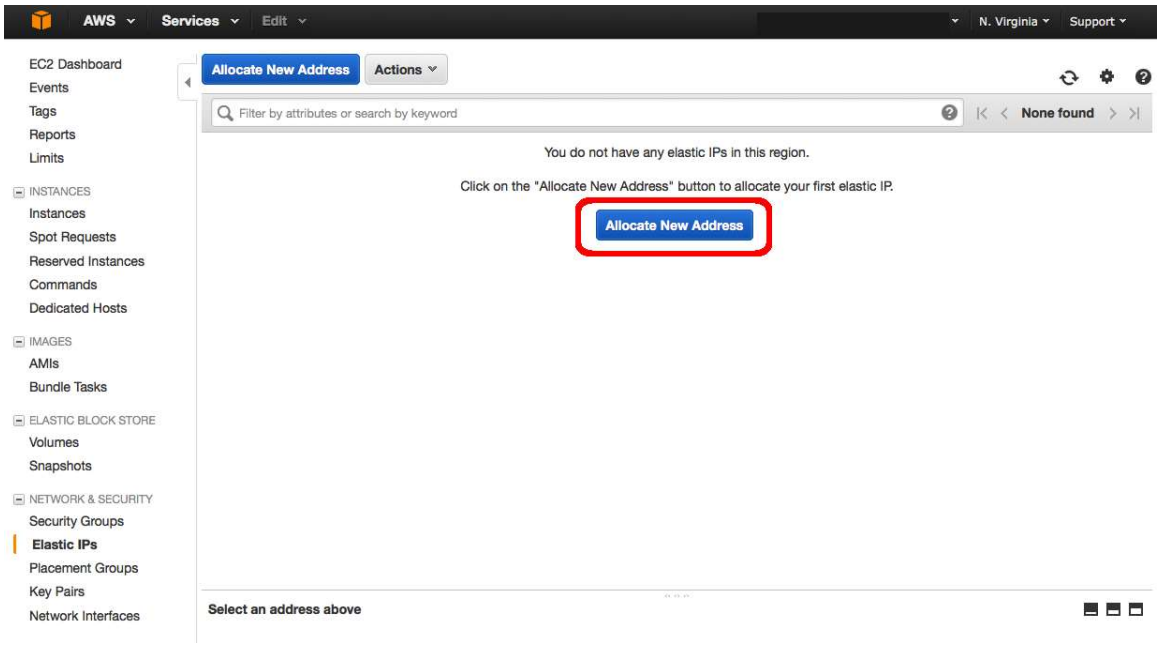
There's an annual fee to register a domain, ranging from $9 to several hundred dollars, depending on the top-level domain, such as .com. For more information, see Amazon Route 53 Pricing for Domain Registration. This fee is not refundable.

When you register a domain, we automatically create a hosted zone that has the same name as the domain. You use the hosted zone to specify where you want Amazon Route 53 to route traffic for your domain. The fee for a hosted zone is $0.50 per month. You can delete the hosted zone if you want to avoid this charge.

Step 1: Obtain a Static URL

Note: If you are using Elastic Load Balancing (Elastic Load Balancing is done automatically if you launched your app with Amazon Elastic Beanstalk) then you do not need to obtain a static IP address and can go directly to **step 2.**

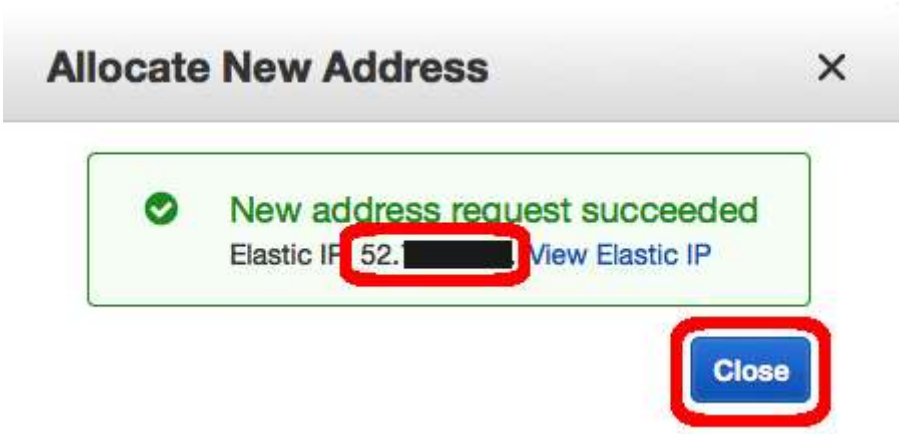
a. Click https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fec2%2Fhome%3Fregion%3Dus-east-1%26state%3DhashArgs%2523Addresses%253Asort%253DpublicIp%26isauthcode%3Dtrue&client_id=arn%3Aaws%3Aiam%3A%3A015428540659%3Auser%2Fec2&forceMobileApp=0 to open the *Elastic IP*s part of the *EC2 console* in a new window and click Allocate New Address.



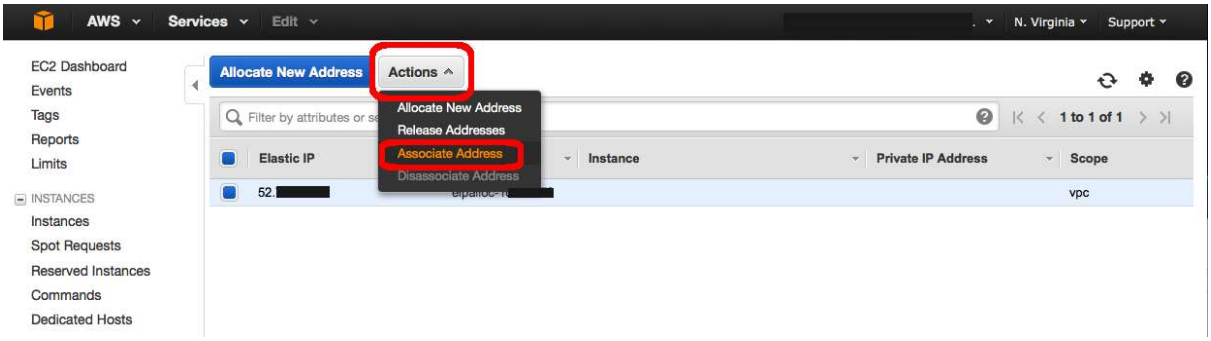
b. Set *EIP used in:* to VPC and click Yes, Allocate.

Note: There is no charge for Elastic IP addresses (EIPs) that are connected to running instances. If you remove the instance (e.g. the EIP is no longer connected to a running instance) then there is a cost of $0.005/hr for the EIP).

c. Note your new *IP address* and click Close.

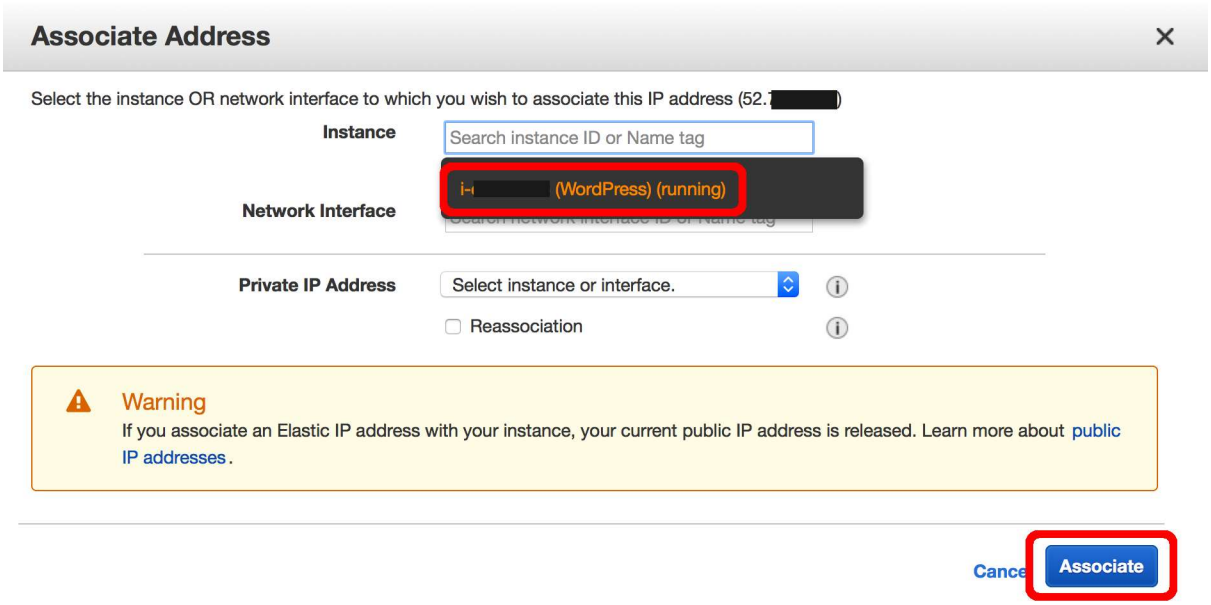


d. Select the new IP address in the *Elastic IP* column. Press the Actions button and choose the Associate Address option.

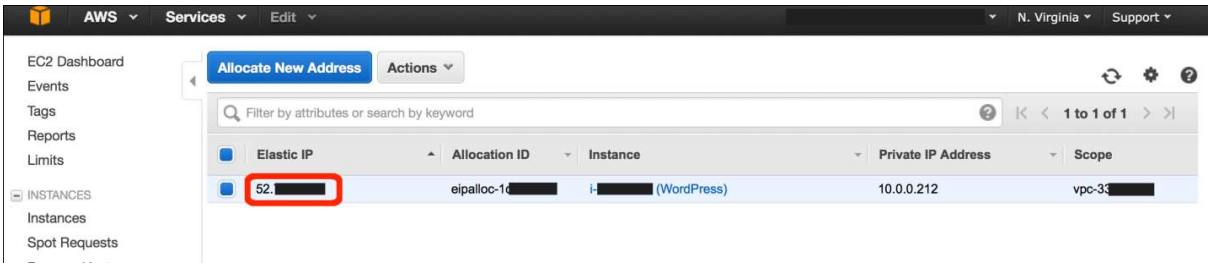


e. Click in the Instance text box and choose the option that has your instance name.

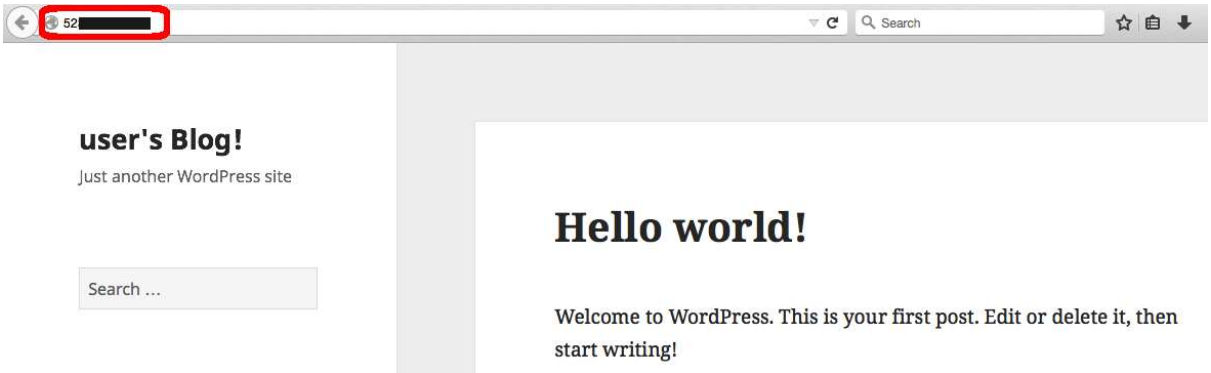
Note: in the WordPress tutorial we named this machine *WordPress*.



f. Make a note of your new IP address in the *Elastic IP* column.



g. Verify that your new Elastic IP address is working by typing it into your web browser.



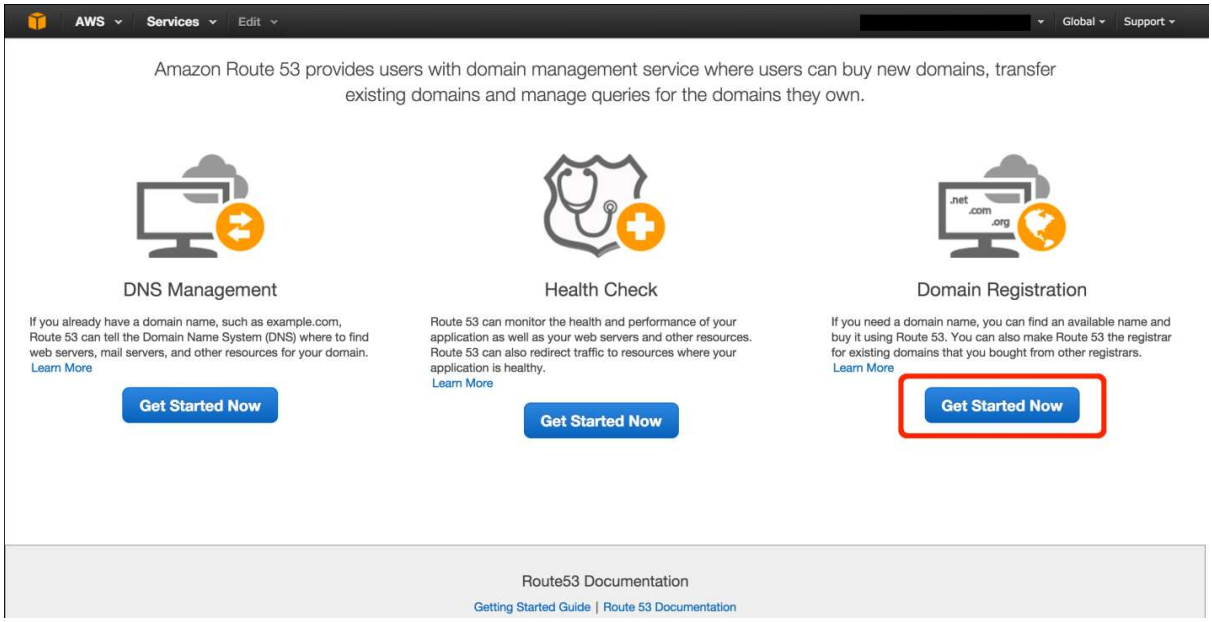
Step 2: Register a Domain Name

Now that you have an IP address associated with your instance, we will need to configure the Domain Name System (DNS) to point to this address so that people can find your website.

Note: In this example we will be acquiring a new domain name and associating it with the Elastic IP address we just created (which is attached to your instance). If you already have a domain name, or if you choose to use another domain registrar to get a domain name, please refer to their documentation on configuring DNS for your instance.

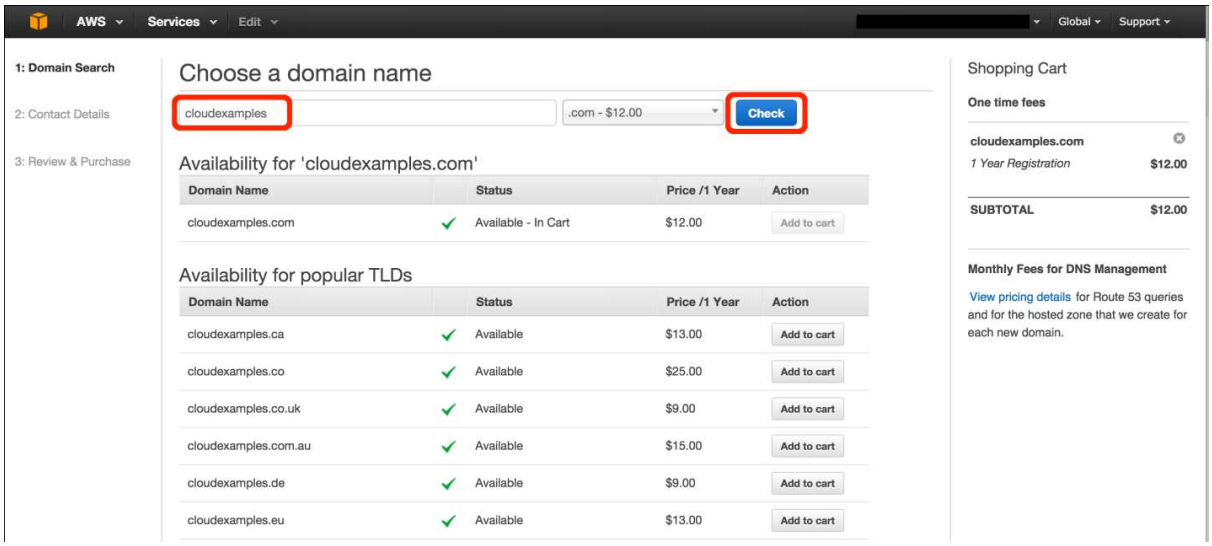
a. Click here to open the Route 53 console in a new window (Route 53 is AWS's DNS service). You can register new domain names with Route 53 as well as manage DNS records for your domain.

Select Get Started Now under *Domain Registration*.

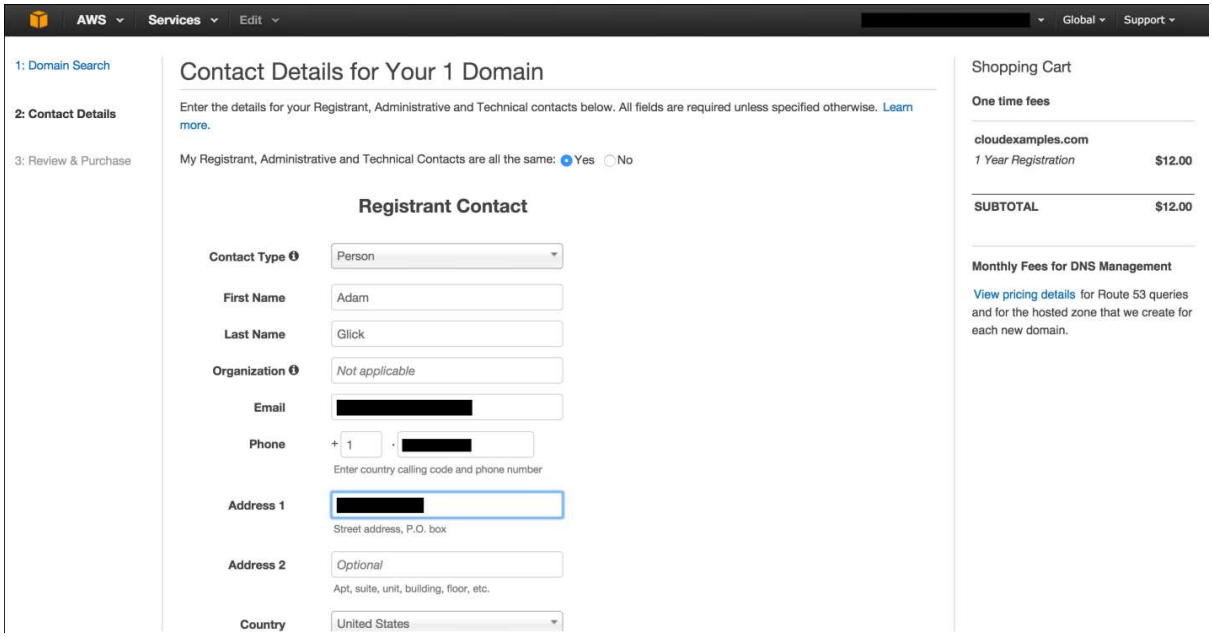


b. Click the Register Domain button. On the next screen, enter the domain you want in the *Choose a Domain* box (cloudexamples is shown in the image, then select a Top Level Domain (TLD) (e.g. .com, .org, .co.uk, etc.) And click the Check button to see if the domain is available. If the domain is available, click the Add to cart button and scroll to the bottom of the page to click Continue.

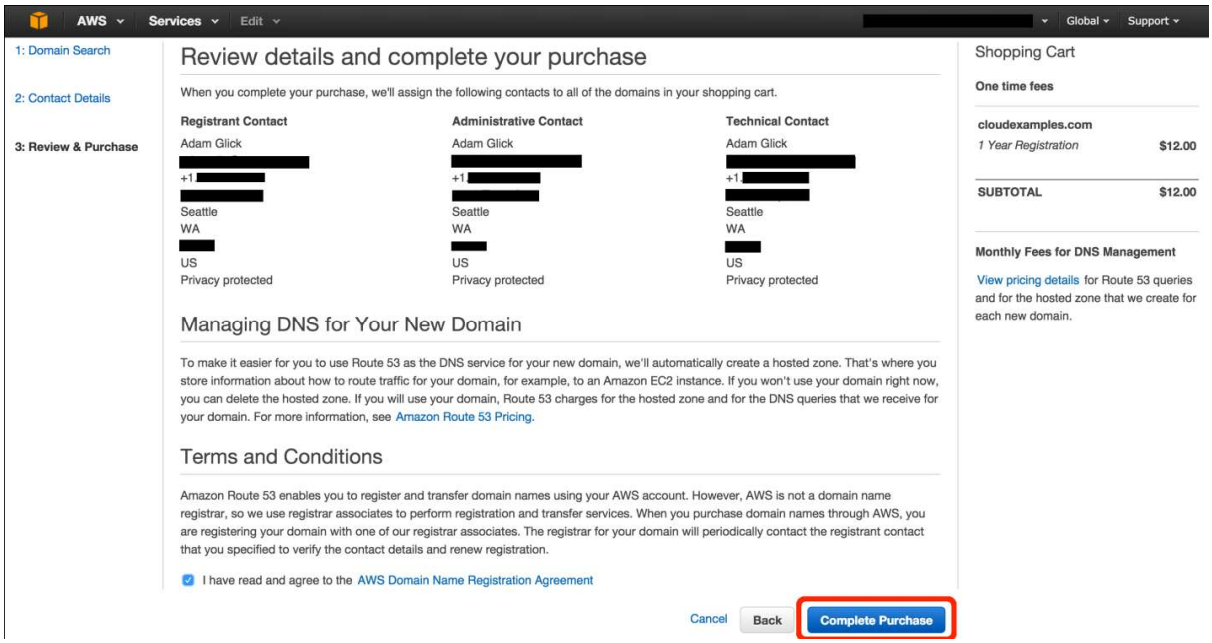
Note: Domains are not part of the free tier so you will be charged for any domain you register.



c. Enter your Contact Details. These are the details that will be associated with your domain name. When you are done, click Continue at the bottom of the page.



d. Review the details as they are listed and, if they are correct, check the box titled *I have read and agree to the AWS Domain Name Registration Agreement*. Then click the Complete Purchase button.

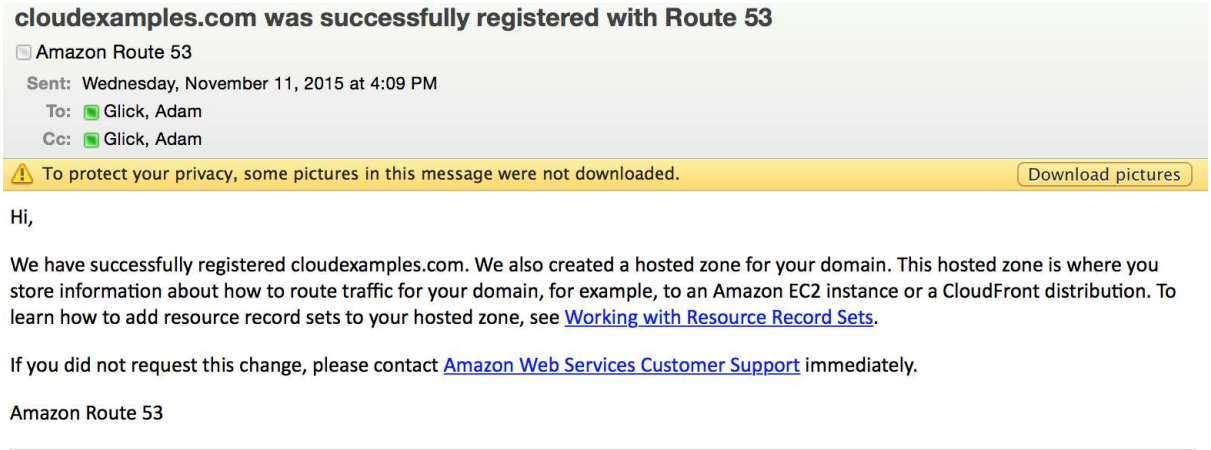


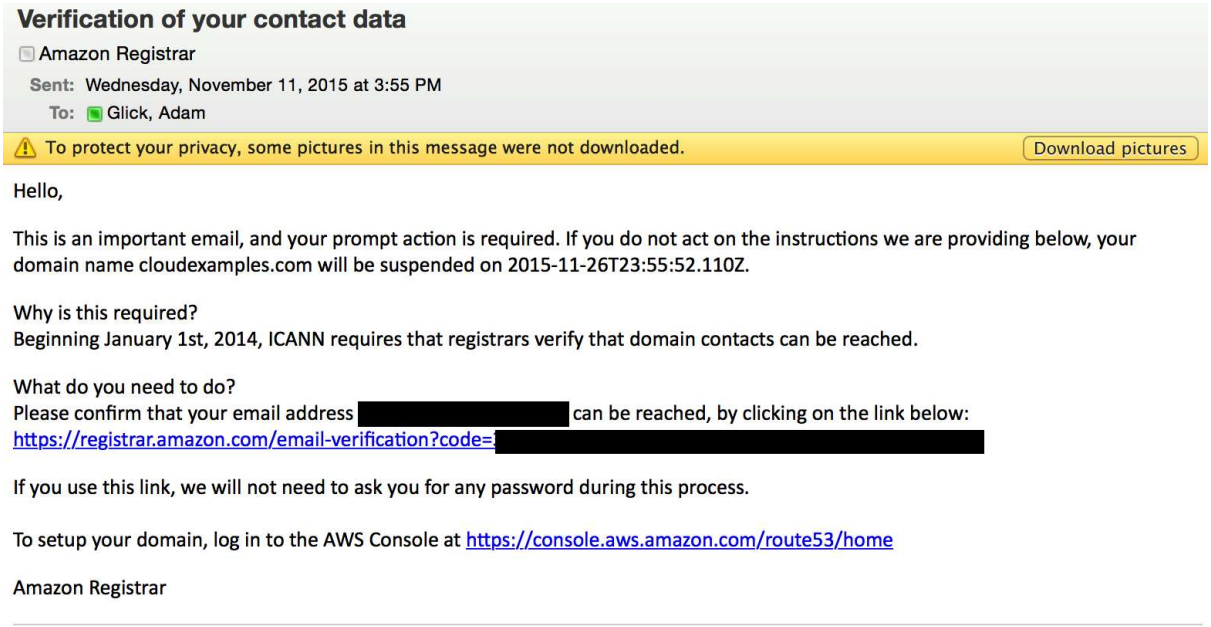
e. If you registered a domain that has a generic top-level domain (such as .com), you'll receive an email that asks you to confirm your email address. (We don't send an email if we already have confirmation that the email address is valid.)

You must follow the link in this email to confirm your email address, or the domain won't be registered.

For all domains, you'll receive an email when your domain registration has been approved.

> Note: it can take a few minutes for the system to confirm the registration of your new domain.





Verification of your contact data

Amazon Registrar

Sent: Wednesday, November 11, 2015 at 3:55 PM

To: Glick, Adam

To protect your privacy, some pictures in this message were not downloaded. Download pictures

Hello,

This is an important email, and your prompt action is required. If you do not act on the instructions we are providing below, your domain name cloudexamples.com will be suspended on 2015-11-26T23:55:52.110Z.

Why is this required?
Beginning January 1st, 2014, ICANN requires that registrars verify that domain contacts can be reached.

What do you need to do?
Please confirm that your email address [redacted] can be reached, by clicking on the link below:
https://registrar.amazon.com/email-verification?code=[redacted]

If you use this link, we will not need to ask you for any password during this process.

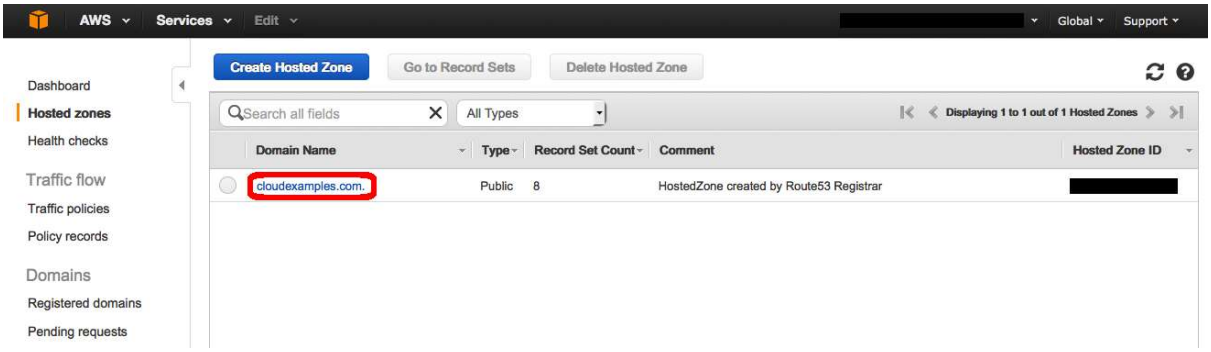
To setup your domain, log in to the AWS Console at https://console.aws.amazon.com/route53/home

Amazon Registrar

Step 3: Configure DNS

Our last step is to configure the DNS so that the new domain we created in step 2 can point to the address we have for our server. This can be the static IP address (from step 1) or a fully qualified domain name (FQDN) that is automatically created if you are using Amazon Elastic Beanstalk.

a. Open the *Hosted Zones* part of the Route 53 console by clicking here. Next, click on the domain name you created in step 2 (in this example we are using cloudexamples.com but your domain will be different).



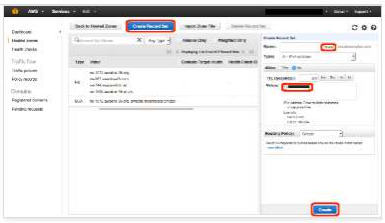
Below are Tabs to help you choose the scenario that is most applicable to you.

If you have a static IP address for your website, virtual server, or service; select Static IP Address below.

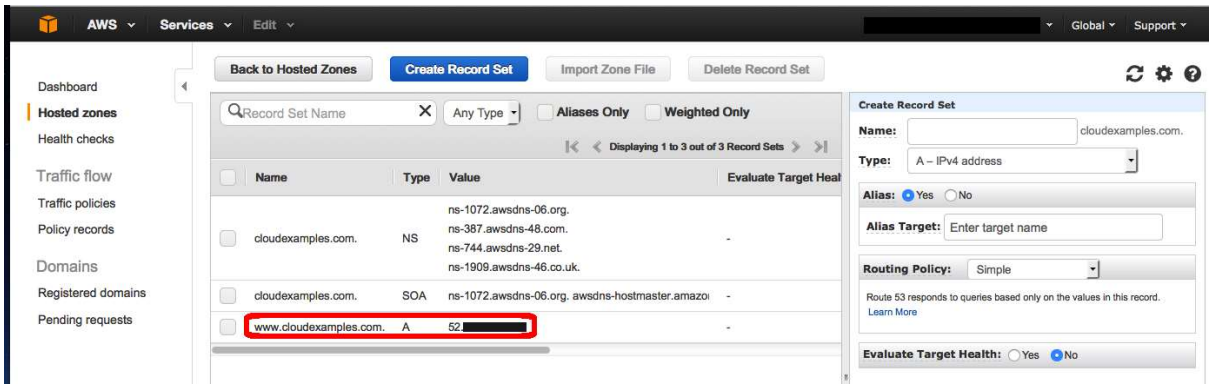
If you have a Fully Qualified Domain Name (FQDN) for your resource (this is common for applications launced by Elastic Beanstalk, Lambda functions, S3 static sites and more advanced deployments using Elastic Load Balancing) please select Fully Qualified Domain Name (FQDN) below.

Static IP Address Fully Qualified Domain Name (FQDN)

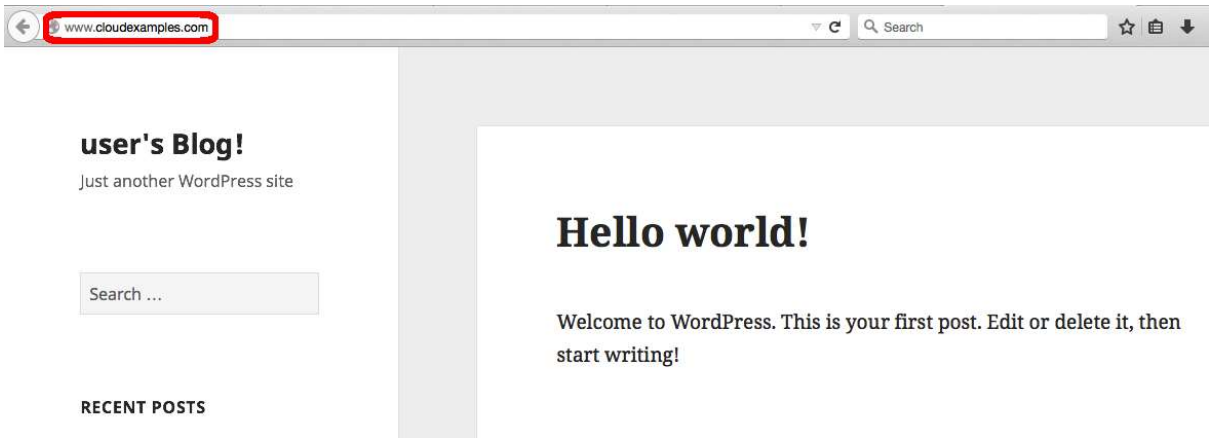
b. Click the **Create Record Set** button. On the right side of the window, enter **www** in the *Name* text box. **Enter the Elastic IP address you created in step 1** in the *Value* box and then click **Create**.



c. Verify that you have a new entry in the main table with the value you entered.



d. Verify that your website is now available at your new domain by typing your new website address into your web browser.



Congratulations! Users can now access your website through the web address you selected.