*Hosting a Web App on Amazon Web Services*

A web app is any software that users access through a web browser or specialized web client. Web apps are typically structured into logical tiers. For example, a common structure uses three tiers. The first tier is the web browser, which is responsible for presenting the user interface. The middle tier is an application server, which is responsible for the application's functionality. The third tier is a database server or file system, which is responsible for data storage.

This tutorial walks you through the process of hosting a scalable, robust web app on AWS infrastructure. We'll deploy a sample app, demonstrating best practices. By the end of this tutorial, you should be able to do the following:

* Create a virtual server, called an EC2 instance, and use it as an application server in the cloud.
* Create a database server, called a DB instance.
* Deploy a sample web app to the application server.
* Set up scaling and load balancing to distribute traffic across a minimum number of application servers.
* Associate a domain name with your web app.

This tutorial walks you through the process of hosting a web app on AWS. Following are the steps:

1. Create an Application Server
2. Create a Database Server
3. Deploy an App to Your Application Server
4. Create a custom AMI
5. Scale and Load-Balance Your Web App
6. Associate a Domain Name with Your Website
7. Clean Up

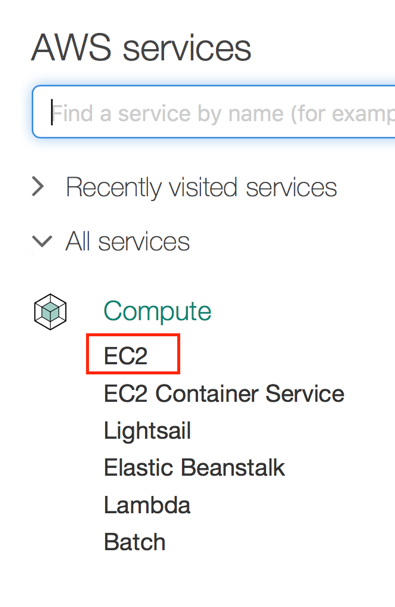
# Step 1: Create an Application Server

You can use Amazon EC2 to create a virtual server to run your web app. These virtual servers are called EC2 instances. Typically, you start from a base image called an Amazon Machine Image (AMI).

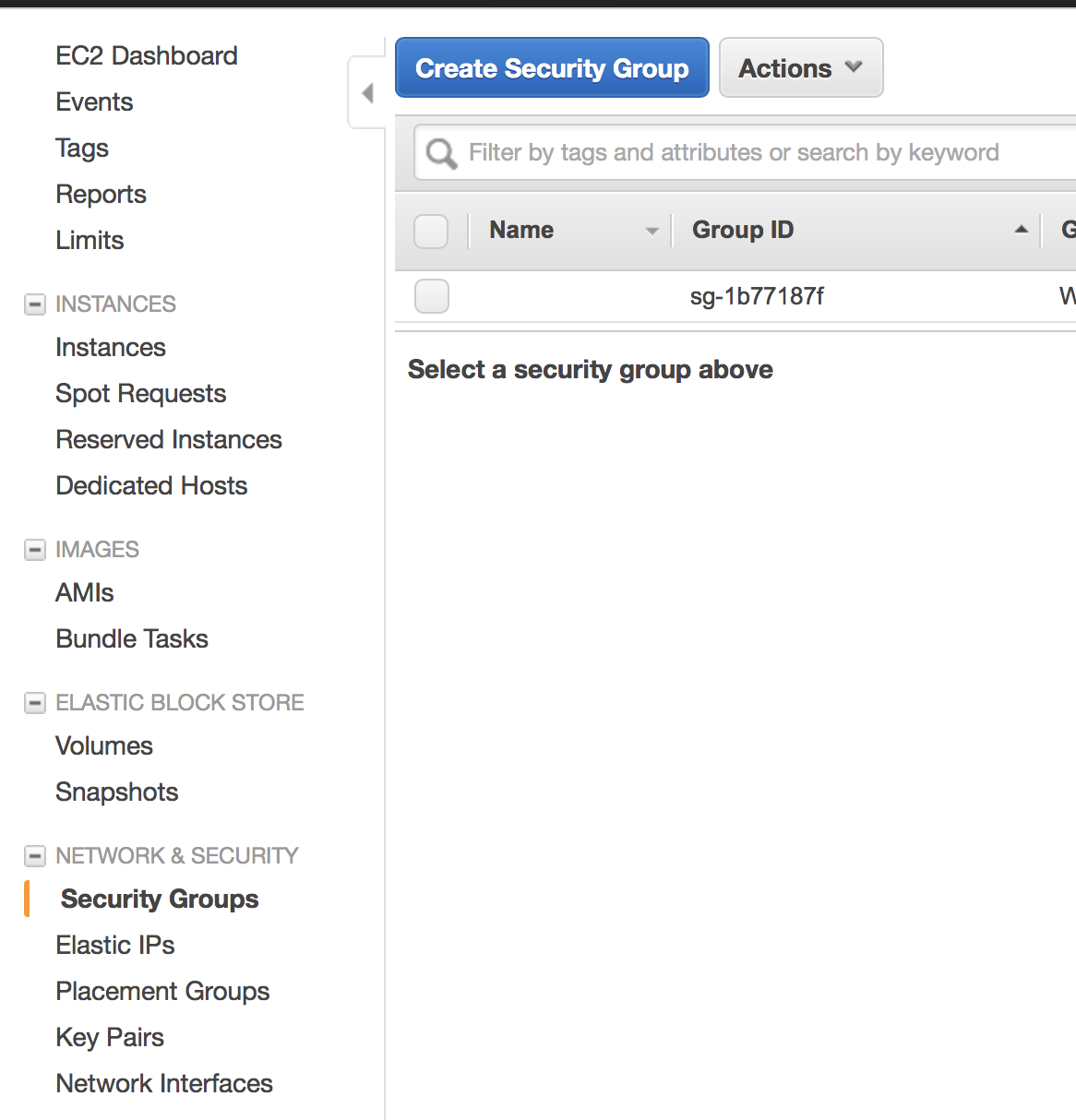
## Create Security Groups for Your Amazon EC2 and RDS DB Instances

Over here, you would be creating following two security groups: one for an EC2 instance and another for RDS DB instance

**Security group for EC2 instance**



1. Click EC2 to go to EC2 dashboard
2. In the navigation pane, click **Security Groups**, and then click **Create Security Group**.



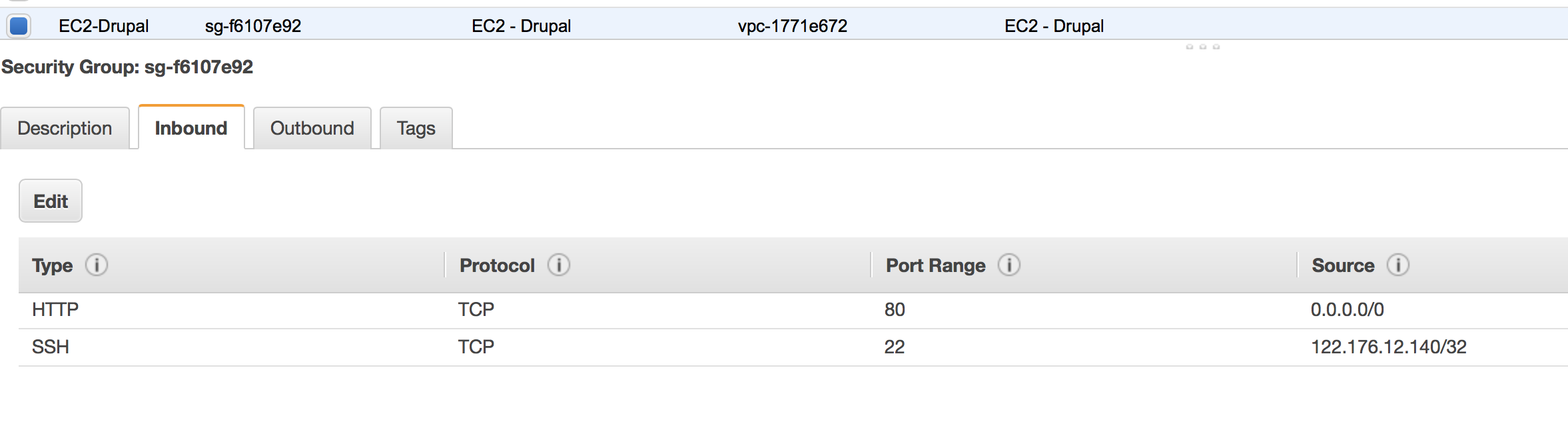
3. Enter **EC2-Drupal** as the name of the security group, and provide a description.

4. Select your VPC from the list.

5. On the **Inbound** tab, add the rules as follows:

1. Click **Add Rule**, and then select **SSH** from the **Type** list. Under **Source**, select **Custom IP** or **MyIP**.
2. Click **Add Rule**, and then select **HTTP** from the **Type** list.

6. Click **Create**.

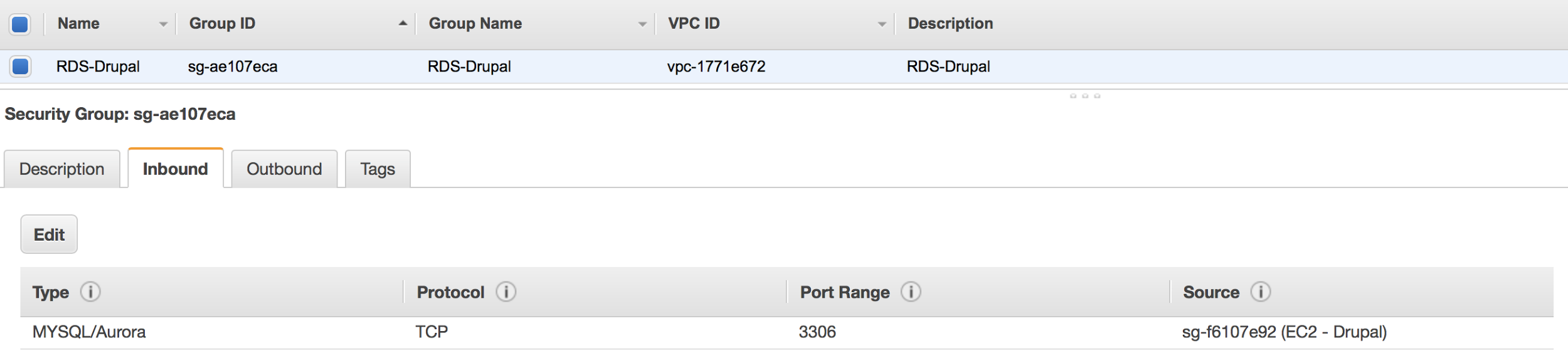


**Security group for RDS DB instance**

7. Enter **RDS-Drupal** as the name of the security group, and provide a description.

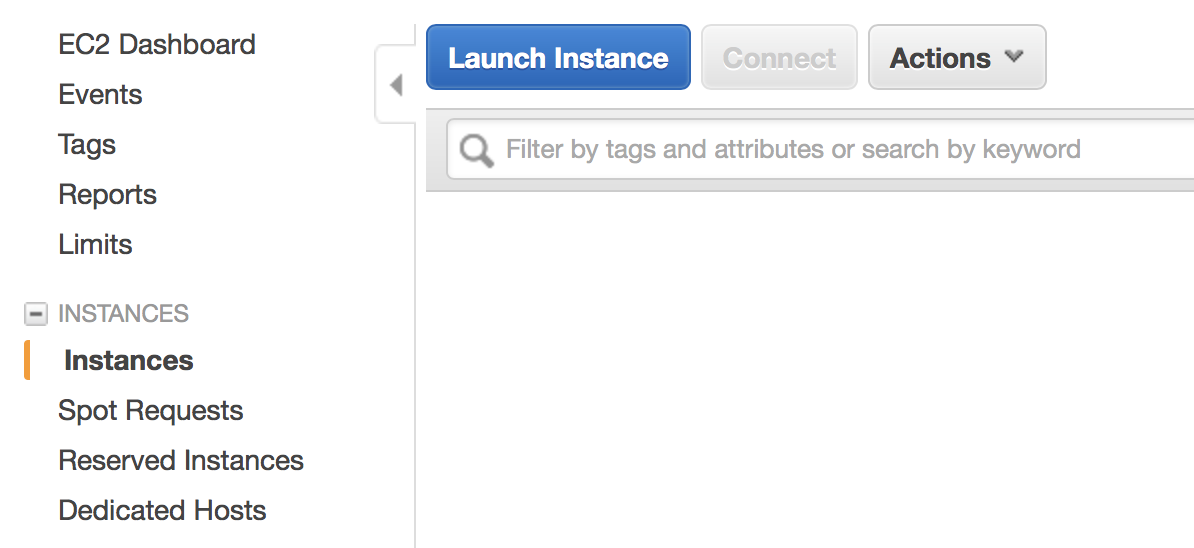
8. Select your VPC from the list.

9. On the **Inbound** tab, add a rule which allows **MYSQL/Aurora** traffic from the security group (EC2-Drupal) created for your EC2 instance.

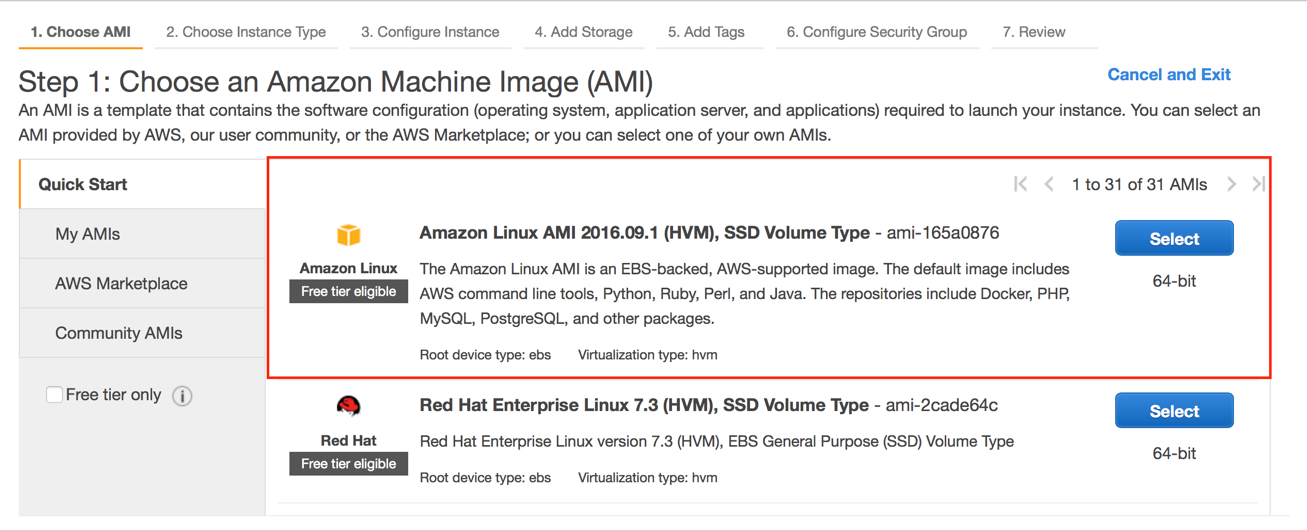


## Launch Your EC2 Instance

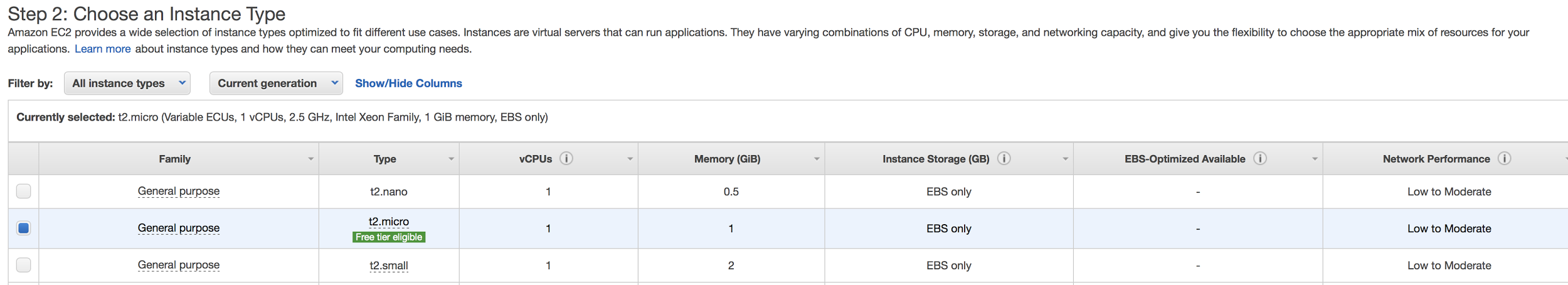
10. In the navigation pane, click **Instances**, and then click **Launch Instance**.



11. On the **Choose an Amazon Machine Image** page, choose **Amazon Linux AMI 2016.09.1 (HVM), SSD Volume Type** and click **Select**.



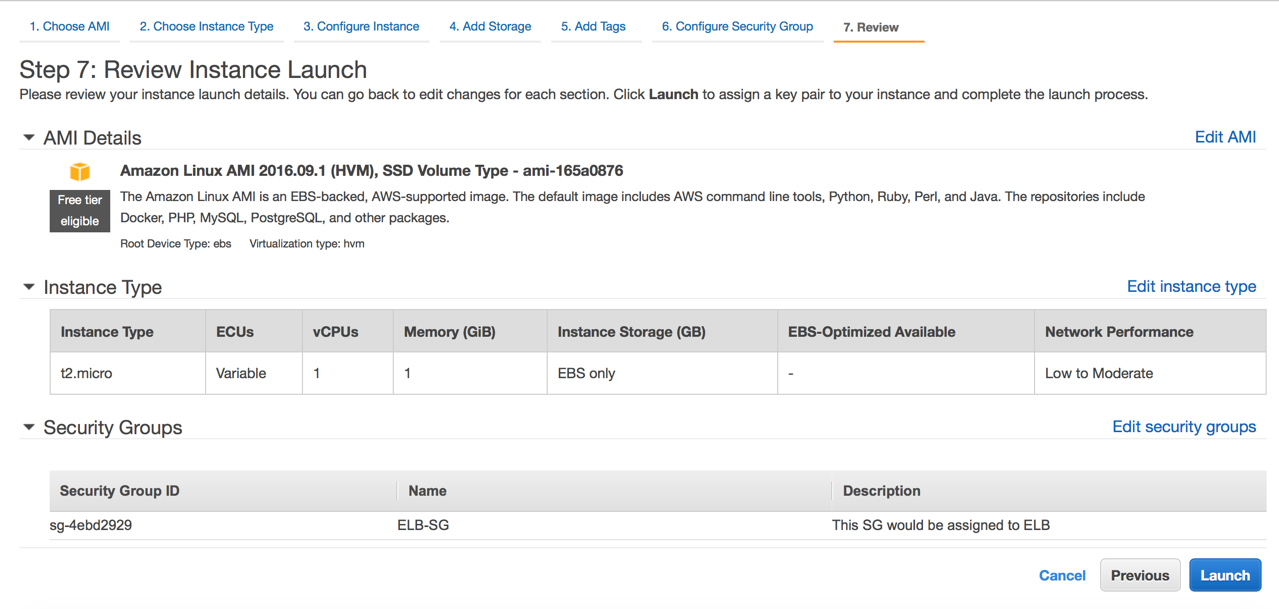
12. On the **Choose an Instance Type** page, the t2.micro instance is selected by default. To stay within the free tier, keep this instance type. Click **Next: Configure Instance Details**.



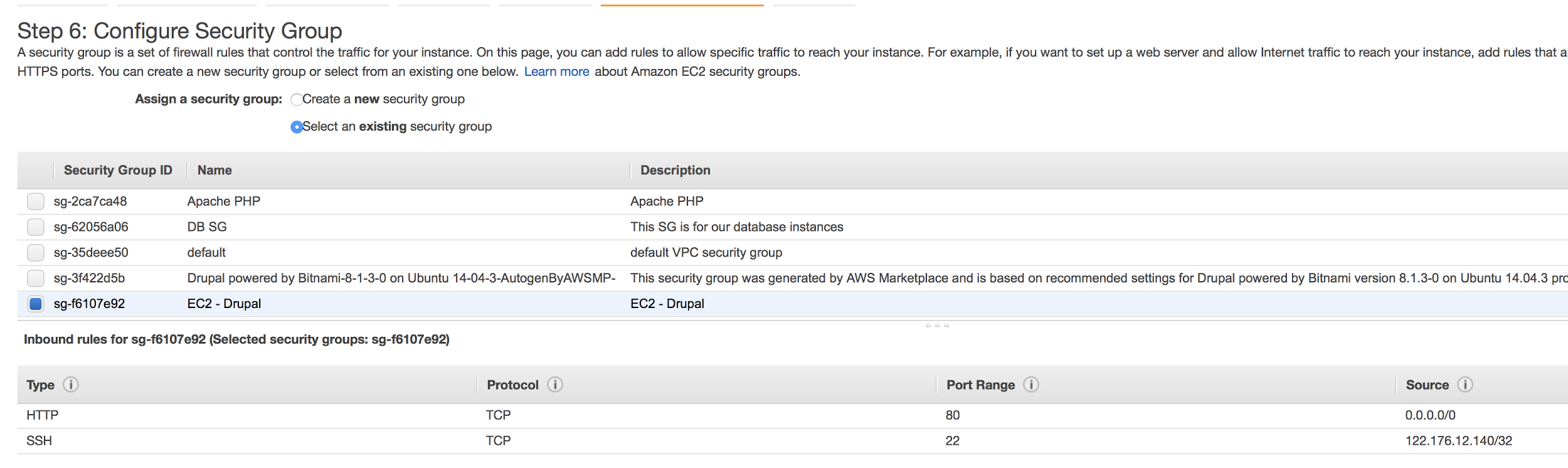
13. On the **Configure Instance Details** page, do the following:

1. T2 instances must be launched into a subnet. Select your VPC from **Network** and select one of your public subnets from **Subnet**.
2. Ensure that for **Auto-assign Public IP**, **Enable** is selected in the list. Otherwise, your instance will not get a public IP address or a public DNS name.
3. Click **Review and Launch**. If you are prompted to specify the type of root volume, make your selection and then click **Next**.

14. On the **Review Instance Launch** page, click **Edit security groups**.

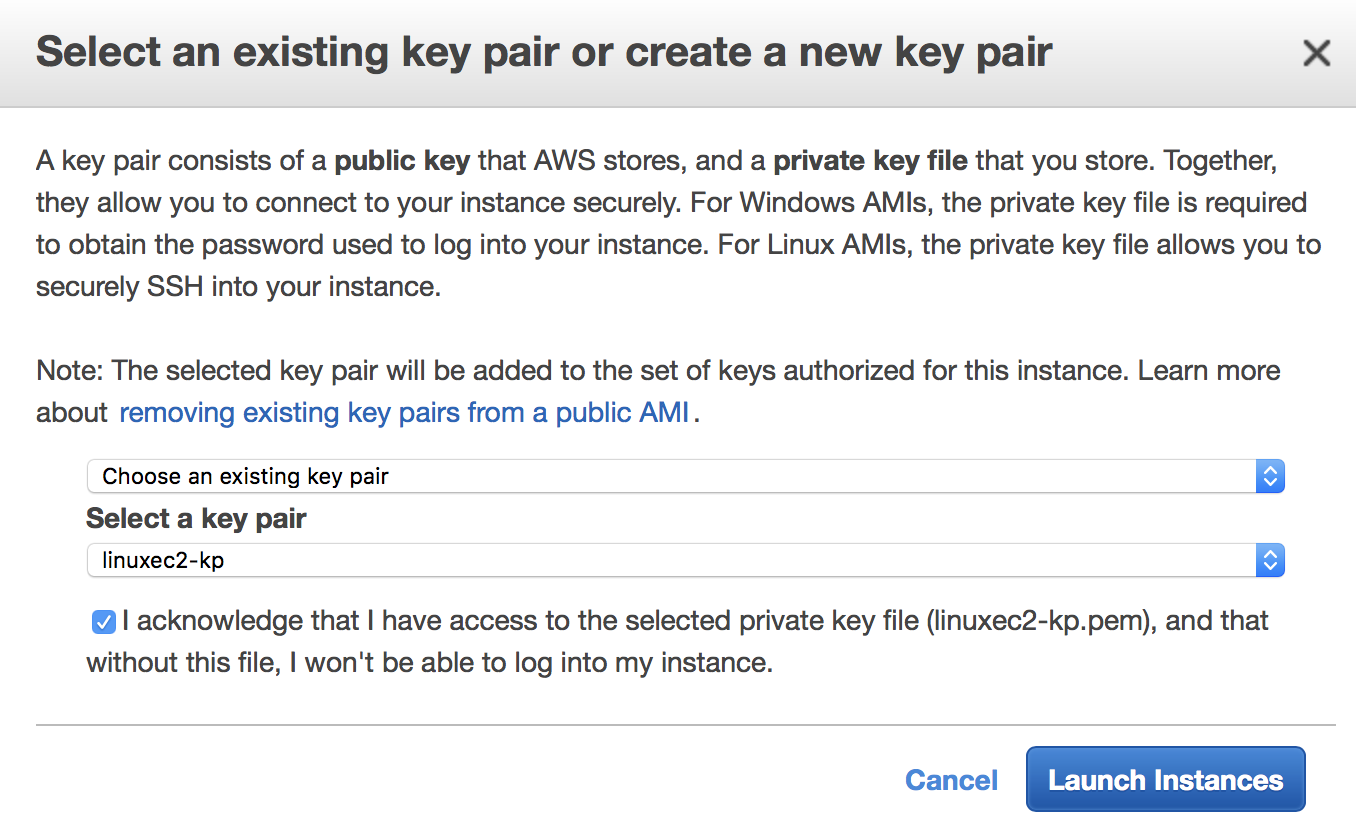


15. In the **Configure Security Group** page, click **Select an existing security group**, select the **EC2-Drupal** security group that you created, and then click **Review and Launch**.

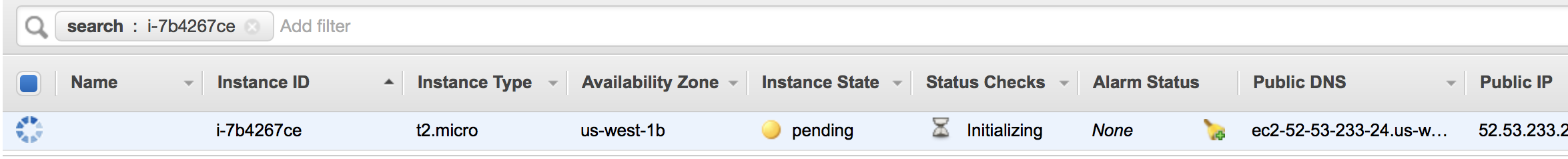


16. On the **Review Instance Launch** page, click **Launch**.

17. In the **Select an existing key pair or create a new key pair** dialog box, select **Choose an existing key pair**, then select the key pair. Click the acknowledgment check box, and then click **Launch Instances**.



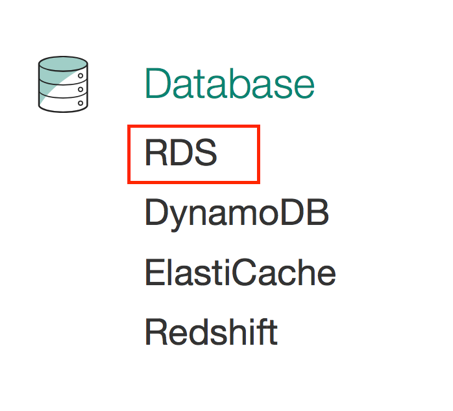
18. In the navigation pane, click **Instances** to see the status of your instance. Initially, the status of your instance is pending. After the status changes to running, your instance is ready for use.



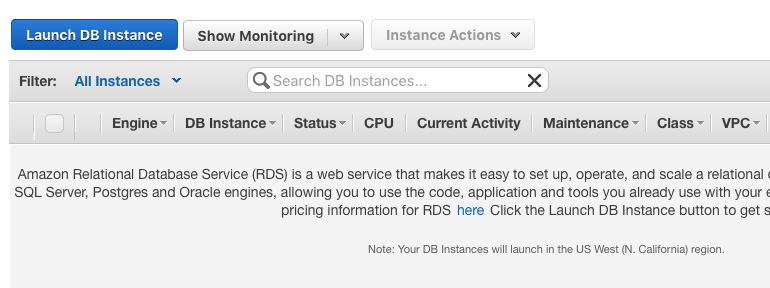
# Step 2: Create a Database Server

You can use Amazon Relational Database Service (Amazon RDS) to run your database server. In this step, you launch a Multi-AZ DB instance. This means that Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. Updates to your DB instance are synchronously replicated across Availability Zones to the standby in order to keep them in sync and protect your latest database updates against DB instance failure. During certain types of planned maintenance, or in the event of DB instance failure or Availability Zone failure, Amazon RDS automatically fails over to the standby. Because the name record for your DB instance remains the same, your app can resume database operation without the need for manual administrative steps.

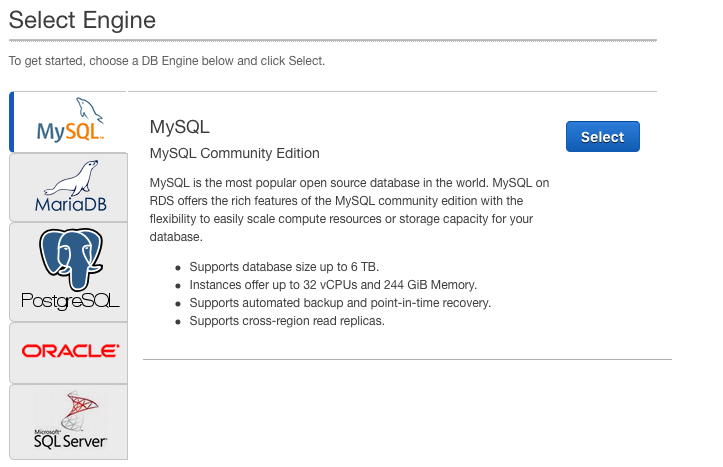
19. Open the Amazon RDS console by clicking on **RDS** within Database section.



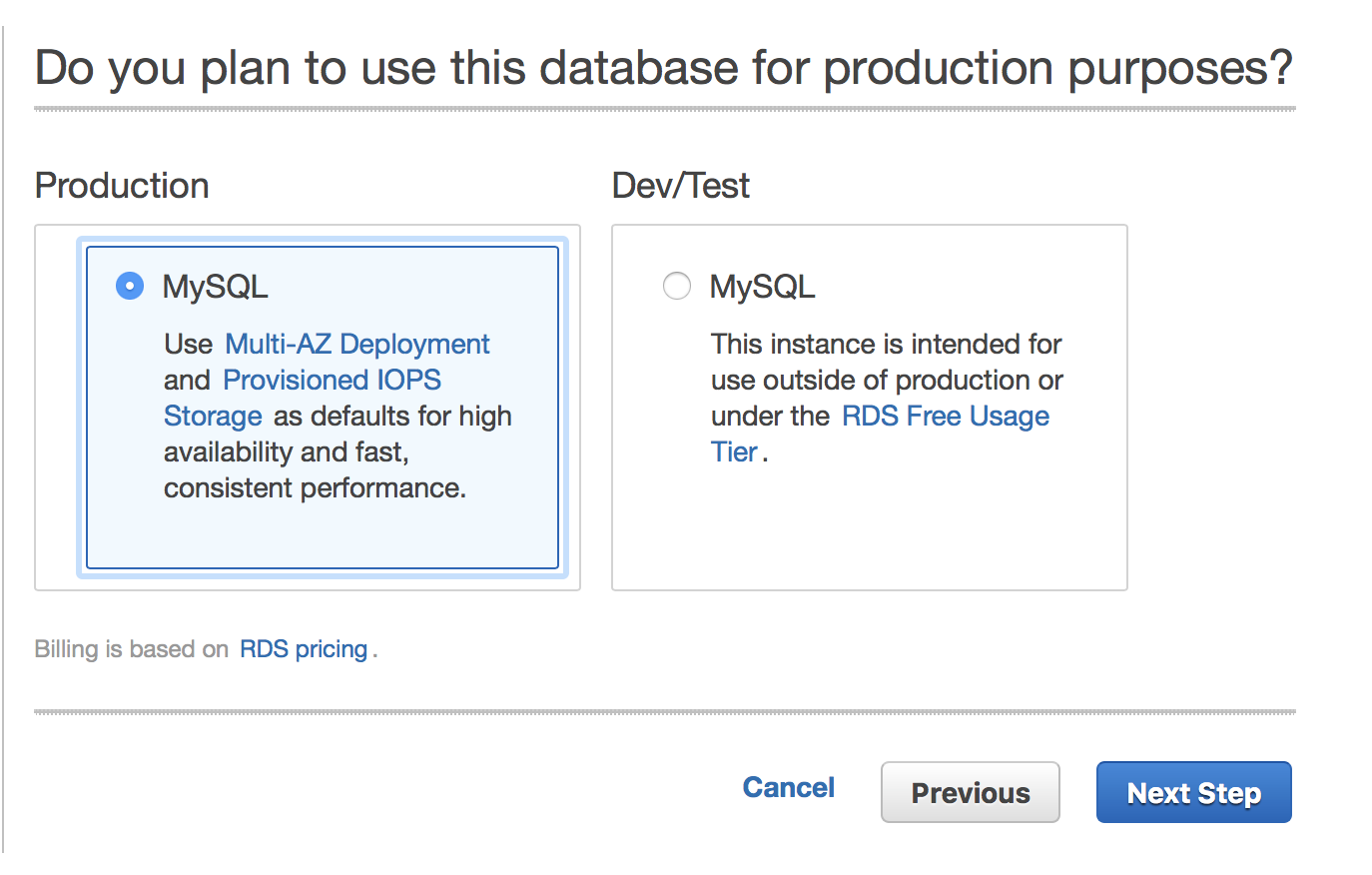
20. Choose **Launch DB Instance**. The Launch DB Instance Wizard opens on the Select Engine page.



21. On the **Select Engine** page, choose the **MySQL** icon and then choose Select for the **MySQL DB engine**.

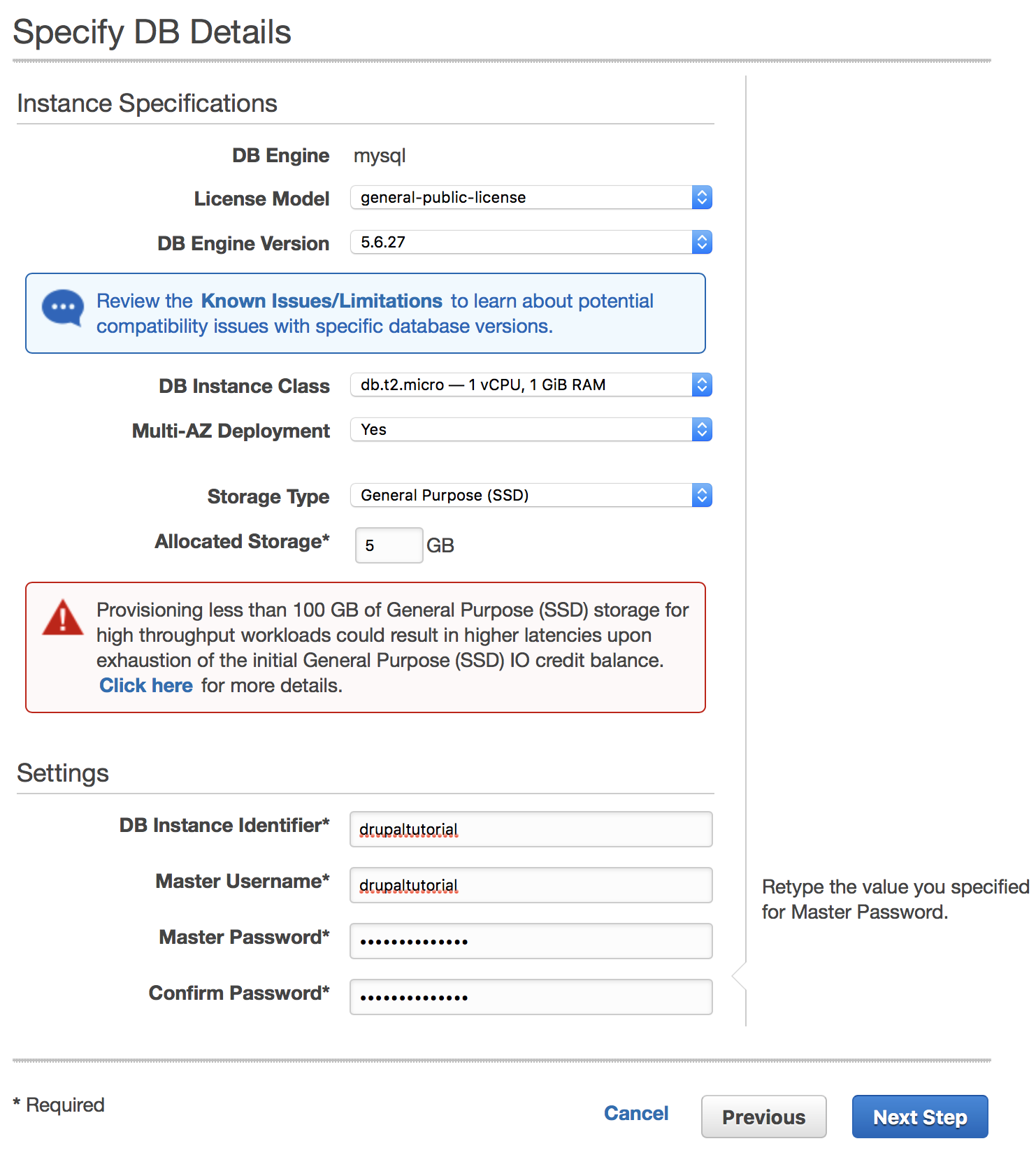


22. Choose **Production** as we need to opt for Multi-AZ Deployment.



23. On the **Specify DB Details** page, specify your DB instance information

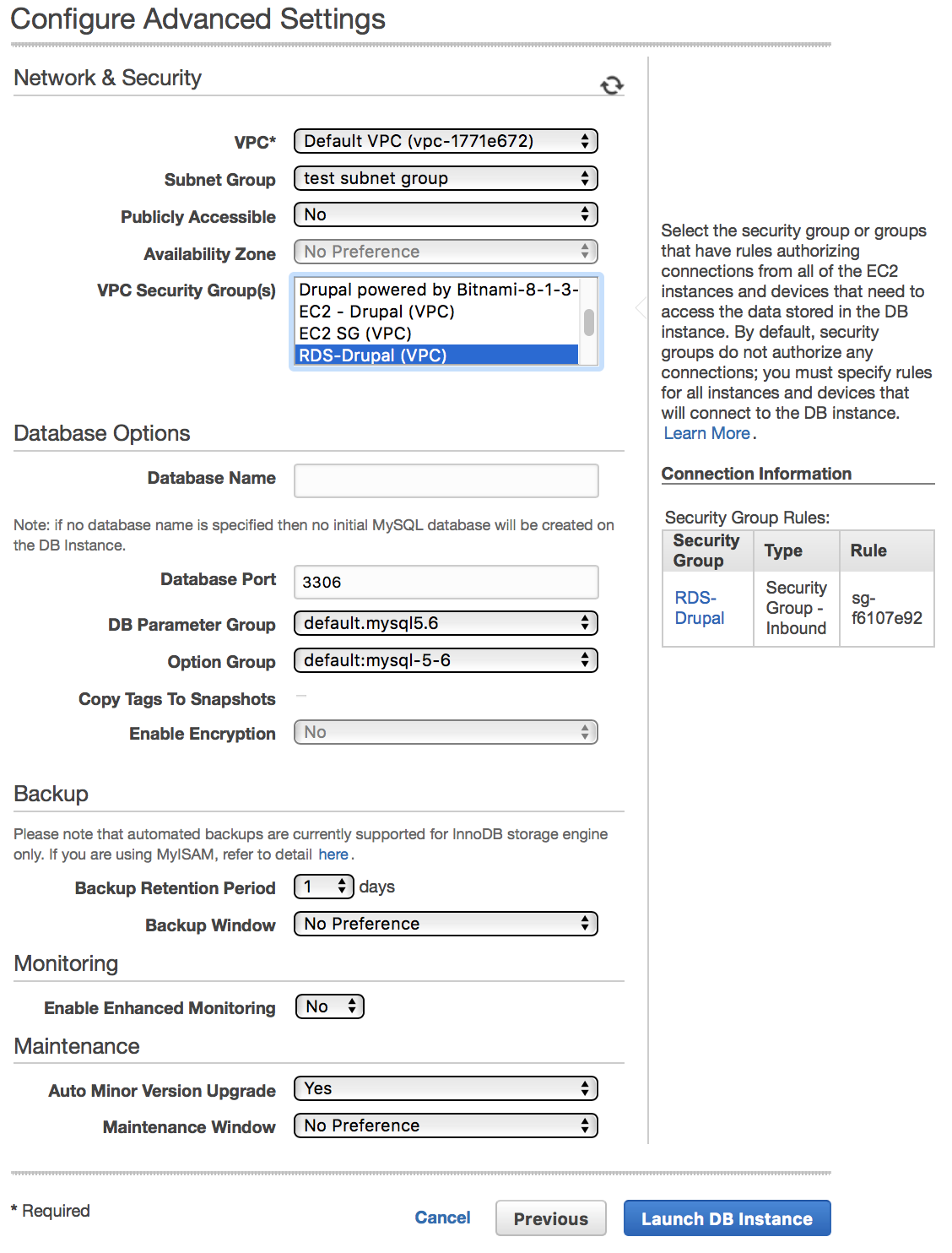
1. **License Model:** Choose the default, **general-public-license**, to use the general license agreement for MySQL. MySQL has only one license model.
2. **DB Engine Version:** Choose the default version of MySQL. Note that Amazon RDS supports multiple versions of MySQL in some regions.
3. **DB Instance Class: Choose db.t2. micro.**
4. **Multi-AZ Deployment:**  Choose **Yes**.
5. **Allocated Storage:** Type **5** to allocate 5 GB of storage for your database. In some cases, allocating a higher amount of storage for your DB instance than the size of your database can improve I/O performance.
6. **Storage Type:** Choose the storage type **Magnetic**
7. **DB Instance Identifier:** Type a name for the DB instance that is unique for your account in the region you chose.
8. **Master Username:** Type a name using alphanumeric characters that you will use as the master username to log on to your DB instance. This will be the username you use to log on to your database on the DB instance for the first time.
9. **Master Password** and **Confirm Password:** This will be the password you will use when you use the username to log on to your database. Then type the password again in the **Confirm Password** box.



24. Click **Next Step**.

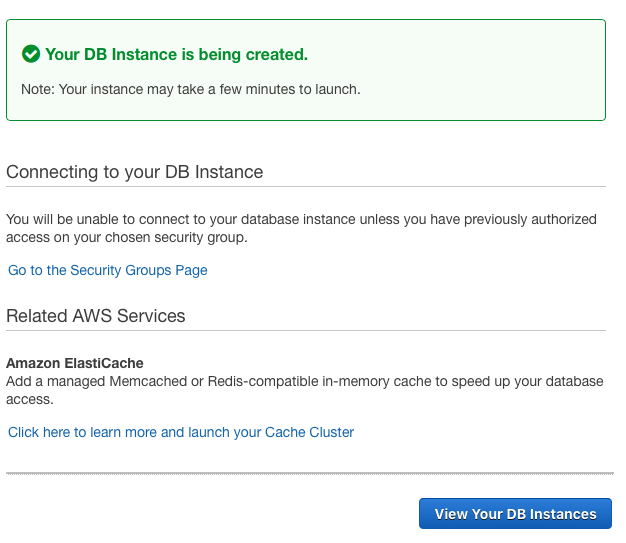
25. On the **Configure Advanced Settings** page, provide additional information that RDS needs to launch the MySQL DB instance.

1. **VPC:** Choose the name of the Virtual Private Cloud (VPC) that will host your MySQL DB instance.
2. **Availability Zone:** Determine if you want to specify a particular Availability Zone. If you chose **Yes** for the Multi-AZ Deployment parameter on the previous page, you will not have any options here.
3. **DB Security Groups:** Choose the security group you want to use with this DB instance. Over here, you will choose **RDS-Drupal** in the list.
4. **Database Name:** Leave the default value of **3306** unless you have a specific port you want to access the database through. MySQL installations default to port 3306.
5. **DB Parameter Group:** Leave the default value unless you created your own DB parameter group.
6. **Option Group:** Choose the default value because this option group is used with the MySQL version you chose on the previous page.
7. **Copy Tags To Snapshots:** Choose this option to have any DB instance tags copied to a DB snapshot when you create a snapshot.
8. **Backup Retention Period:** Set the number of days you want automatic backups of your database to be retained. For testing purposes, you can set this value to **1**.
9. **Backup Window:** Unless you have a specific time that you want to have your database backup, use the default of **No Preference**.
10. **Enable Enhanced Monitoring:** Unless you want to enable gathering metrics in real time for the operating system that your DB instance runs on, use the default of **No**.
11. **Auto Minor Version Upgrade:** Choose **Yes** to enable your DB instance to receive minor DB engine version upgrades automatically when they become available.
12. **Maintenance Window:** Choose the 30-minute window in which pending modifications to your DB instance are applied. If you the time period doesn't matter, choose **No Preference**.



26. Click **Launch DB Instance**.

27. Click **View Your DB Instances**.



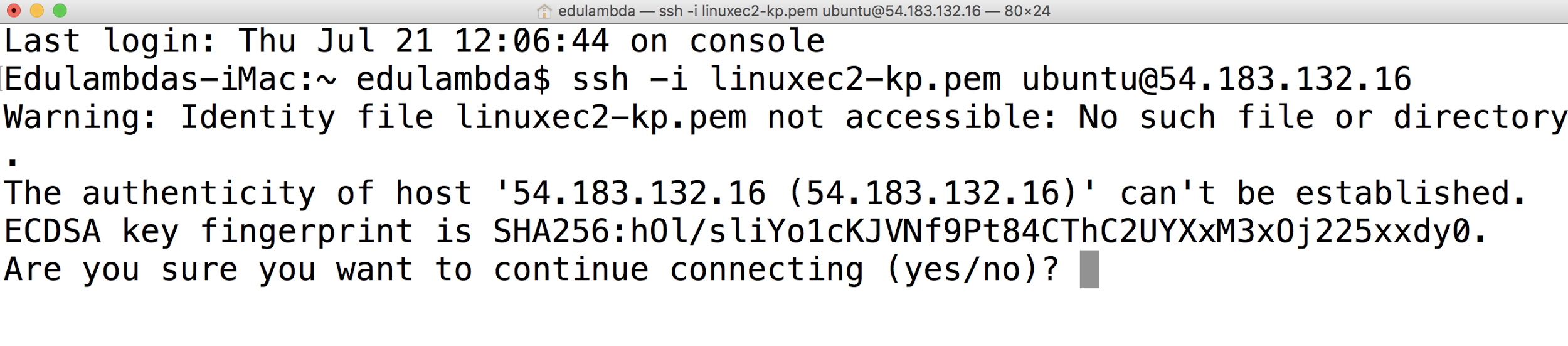
28. On the RDS console, the new DB instance appears in the list of DB instances. The DB instance will have a status of **creating** until the DB instance is created and ready for use. When the state changes to **available**, you can connect to a database on the DB instance. Depending on the DB instance class and store allocated, it could take several minutes for the new DB instance to become available.



# Step 3: Deploy Your App

Deploy the app to your EC2 instance by completing the following tasks. For this tutorial, you'll install Drupal and create a test page.

29. Connect to your Linux instance via SSH.



## Configure the EC2 Instance

To configure the instance, we'll start the web server, install the app, and then configure the app to use our database server. If you don't have the files for your web app, you can use the Drupal application, as we do in this tutorial.

**Tasks**

* Start the Web Server
* Install the App
* Configure Drupal
* Test the Website

**Start the Web Server**

Use the following procedure to start the web server.

**To start the web server**

A. To ensure that your software packages are up to date on your instance, use the following command to perform a quick software update.

[ec2-user ~]$ **sudo yum update –y**

B. Install the Apache web server, PHP, and MySQL software packages using the following command.

[ec2-user ~]$ **sudo yum install -y httpd24 php56 mysql55-server php56-mysqlnd php56-mbstring php56-gd php56-opcache**

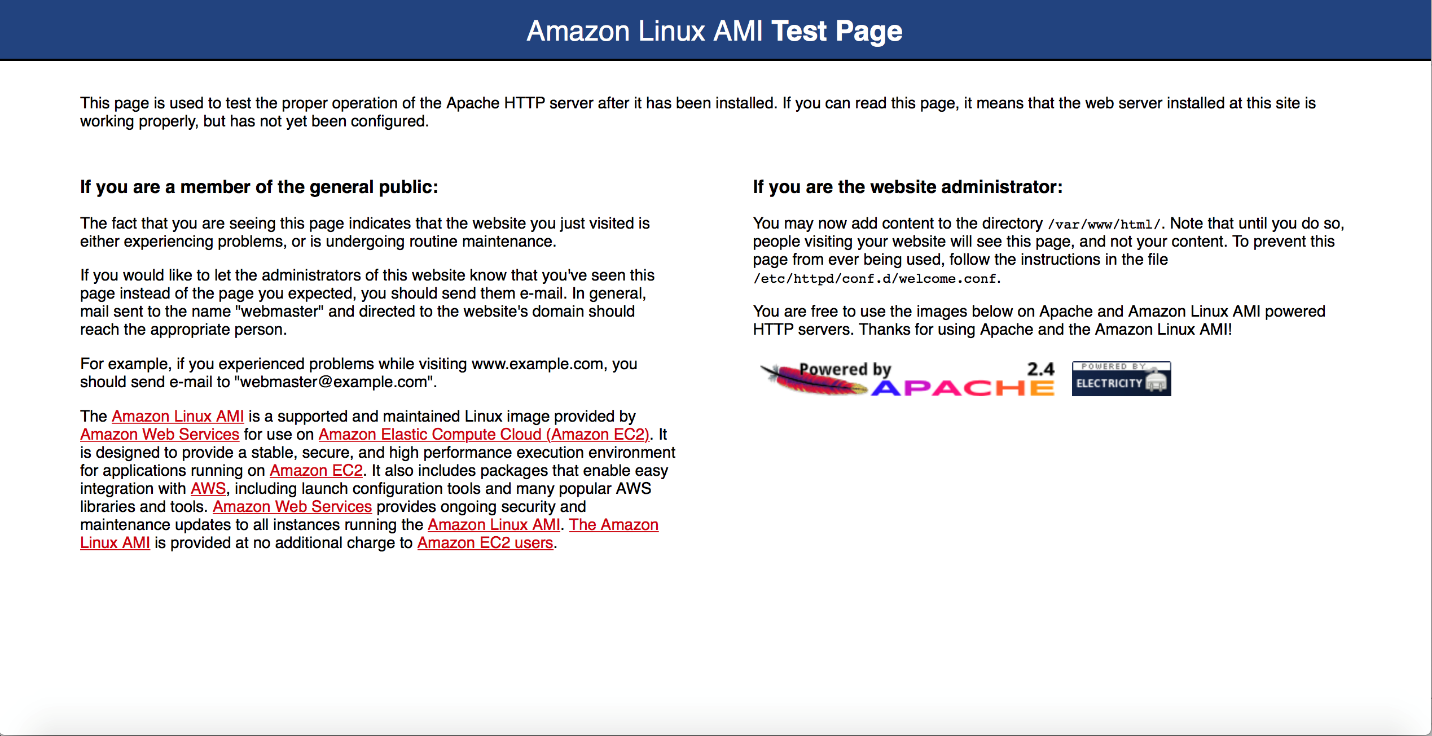
C. Start the Apache web server using the following command.

[ec2-user ~]$ **sudo service httpd start**

D. Configure the Apache web server to start at each system boot using the following command.

[ec2-user ~]$ **sudo chkconfig httpd on**

E. Before you continue, verify that the web server is running. In a web browser on your computer, paste the public DNS name of your instance into the address bar and press Enter. This displays the Apache test page. If you don't see the test page for Apache, verify that your security group allows HTTP traffic.



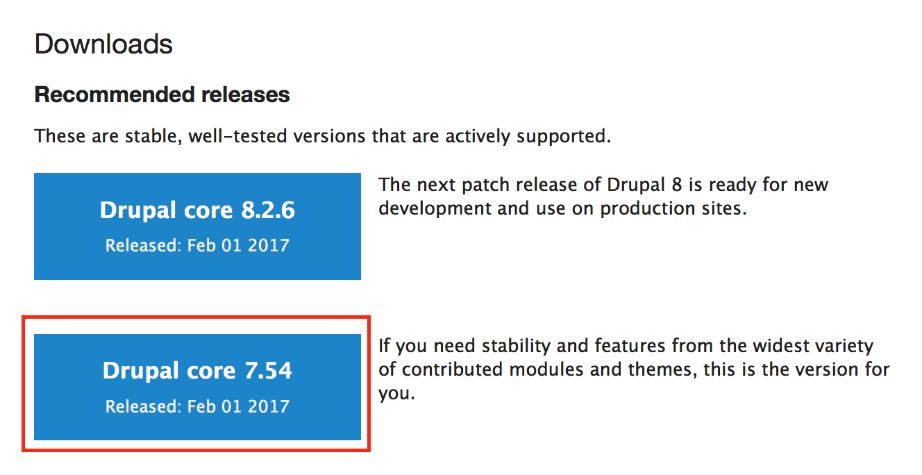
**Install the App**

Now you are ready to install and configure the web app. The procedures depend on the app that you're running. This tutorial installs Drupal to demonstrate how to download files to your instance and configure your app to use your database server.

**To install Drupal**

A. Go to <https://www.drupal.org/project/drupal> and note the version of Drupal that you'd like to use.

Note: Over here, we would be choosing version **7.54.**



B. On your instance, use the following command to download Drupal.

[ec2-user ~]$ **wget http://ftp.drupal.org/files/projects/drupal-7.54.tar.gz**

C. Extract Drupal using the following command, where x.y is the Drupal version.

[ec2-user ~]$ **tar -xzvf drupal-7.54.tar.gz**

(Optional) You can use the following command to verify that the current directory contains the compressed and uncompressed versions.

[ec2-user ~]$ **ls** drupal-7.54 drupal-7.54.tar.gz

(Optional) To remove the compressed version of Drupal, use the following command, where x.y is the Drupal version.

[ec2-user ~]$ **rm drupal-7.54.tar.gz**

**D.** Create the files directory using the following commands.

[ec2-user ~]$ **cd drupal-*7.54*/**

[ec2-user drupal-x.y]$ **mkdir sites/default/files**

Copy the *settings.php* file using the following command.

[ec2-user drupal-7.54]$ cp sites/default/default.settings.php sites/default/settings.php

E. Move Drupal to **/var/www/html/** using the following command.

[ec2-user drupal-7.54]$ **sudo rsync -avh . /var/www/html/**

(Optional) To remove the decompressed version of Drupal, you can use the following commands, where x.y is the Drupal version.

[ec2-user drupal-x.y]$ **cd ..**

[ec2-user ~]$ **rm -rf drupal-7.54/**

F. Grant ownership of the **/var/www** directory and its contents to the apache user using the following command.

[ec2-user ~]$ **sudo chown -R apache /var/www**

**G.** Enable clean URLs, as recommended by Drupal, using the following command.

[ec2-user ~]$ **sudo vim /etc/httpd/conf/httpd.conf**

Find the section **<Directory "/var/www/html">,** and set **AllowOverride** as follows:

AllowOverride All

Save the file (using Esc, :x, Enter), and then restart the Apache web server using the following command.

[ec2-user ~]$ **sudo service httpd restart**

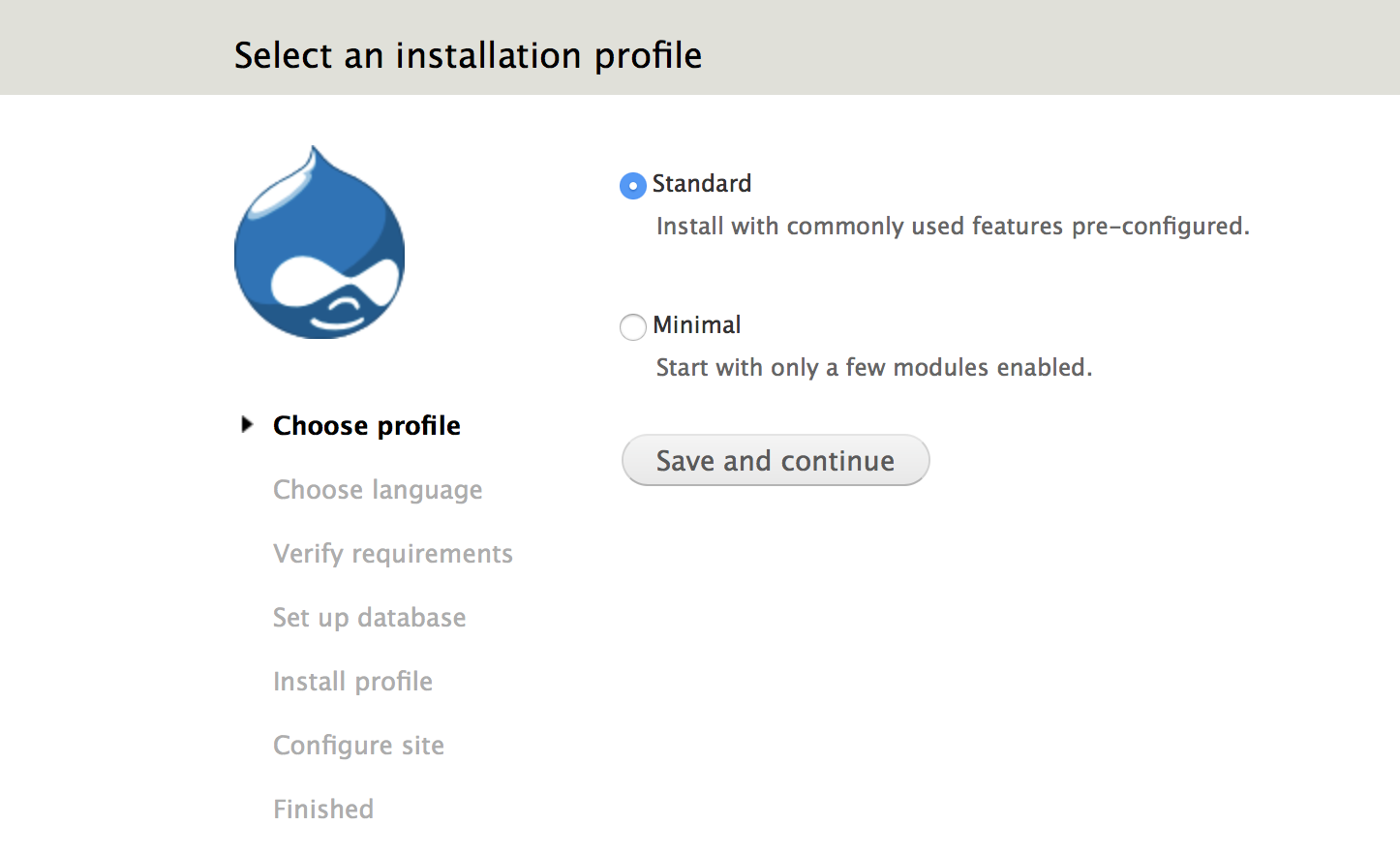
## 

## Configure Drupal

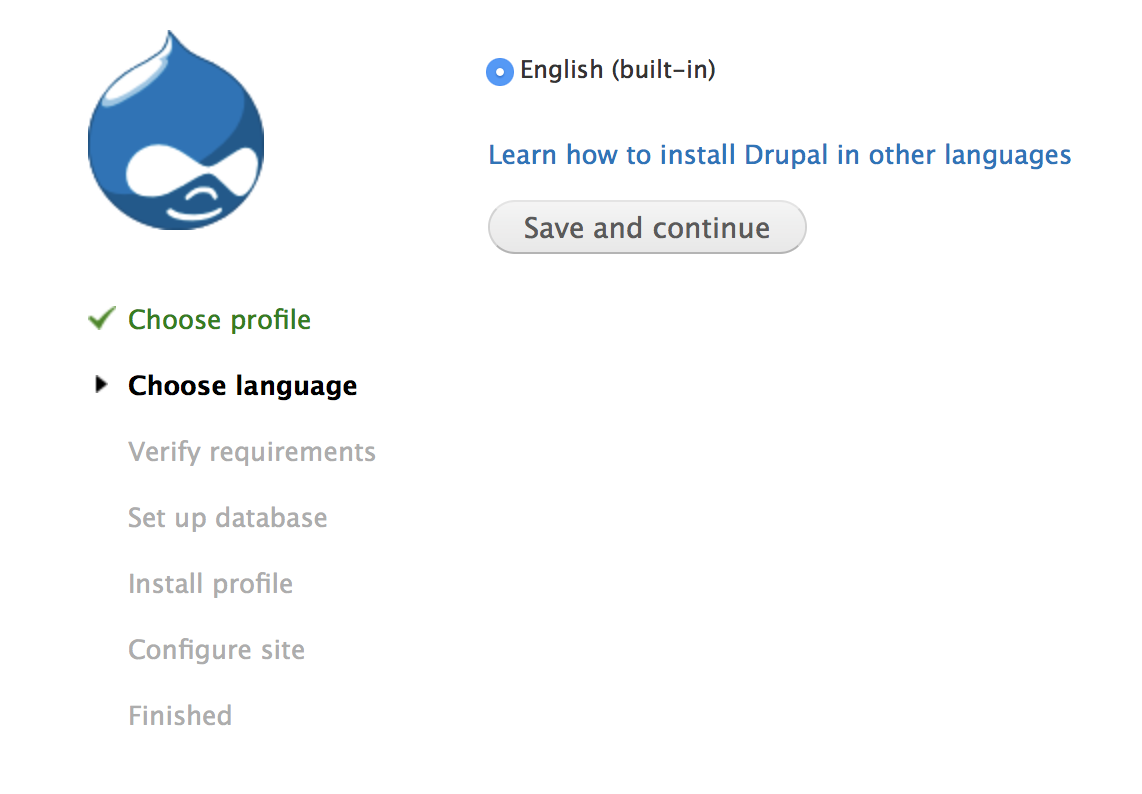
Drupal includes an installation wizard that you can use to configure your website. In the previous procedure, you installed Drupal to the Apache document root, so you can start the installation wizard by opening the website.

31. Open a web browser on your computer, and enter the public DNS address of your instance in the address bar.

32. On the **Choose profile** page, click **Standard** and then click **Save and continue**.

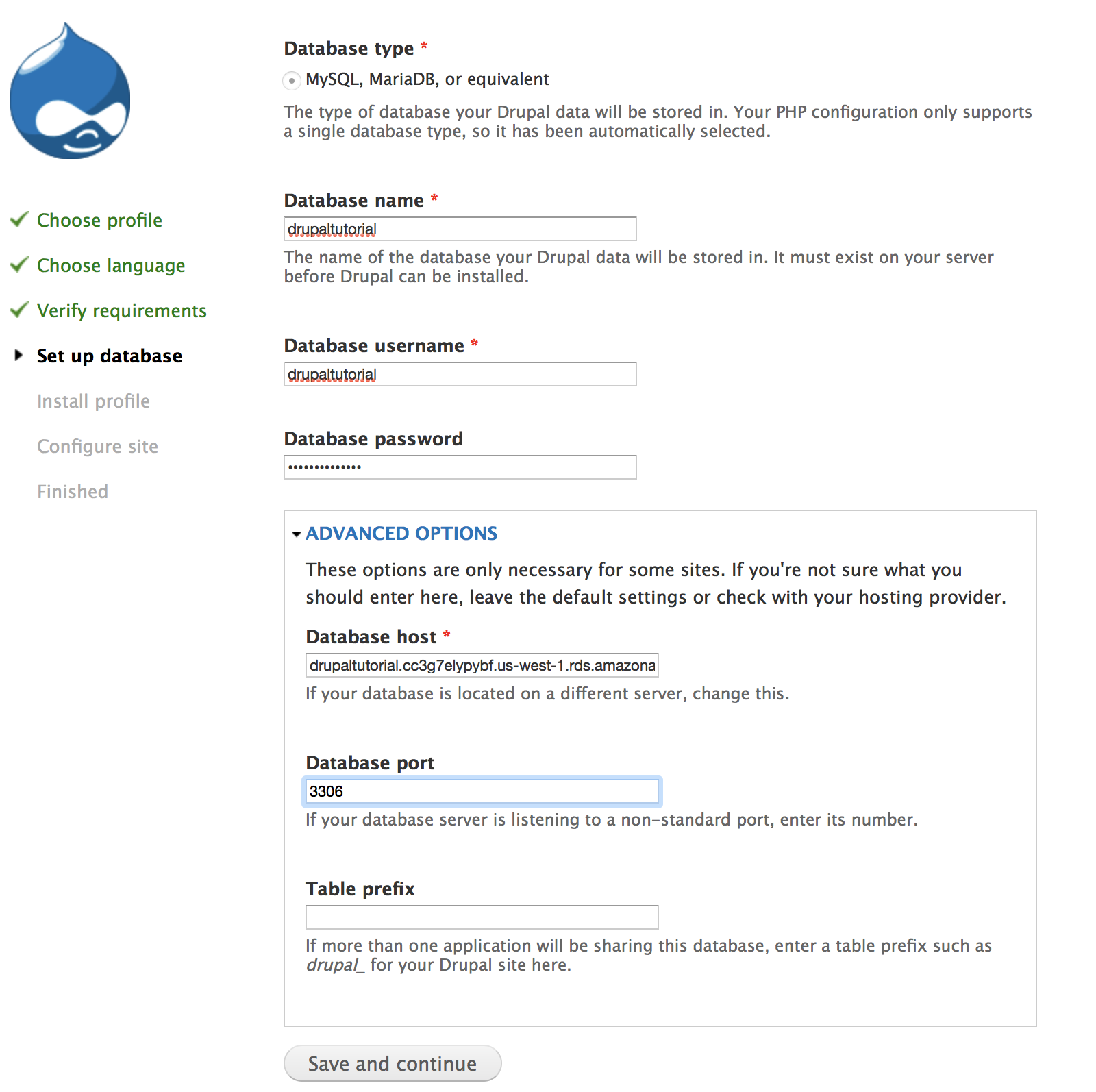


33. On the **Choose language** page, select the language and then click **Save and continue**.

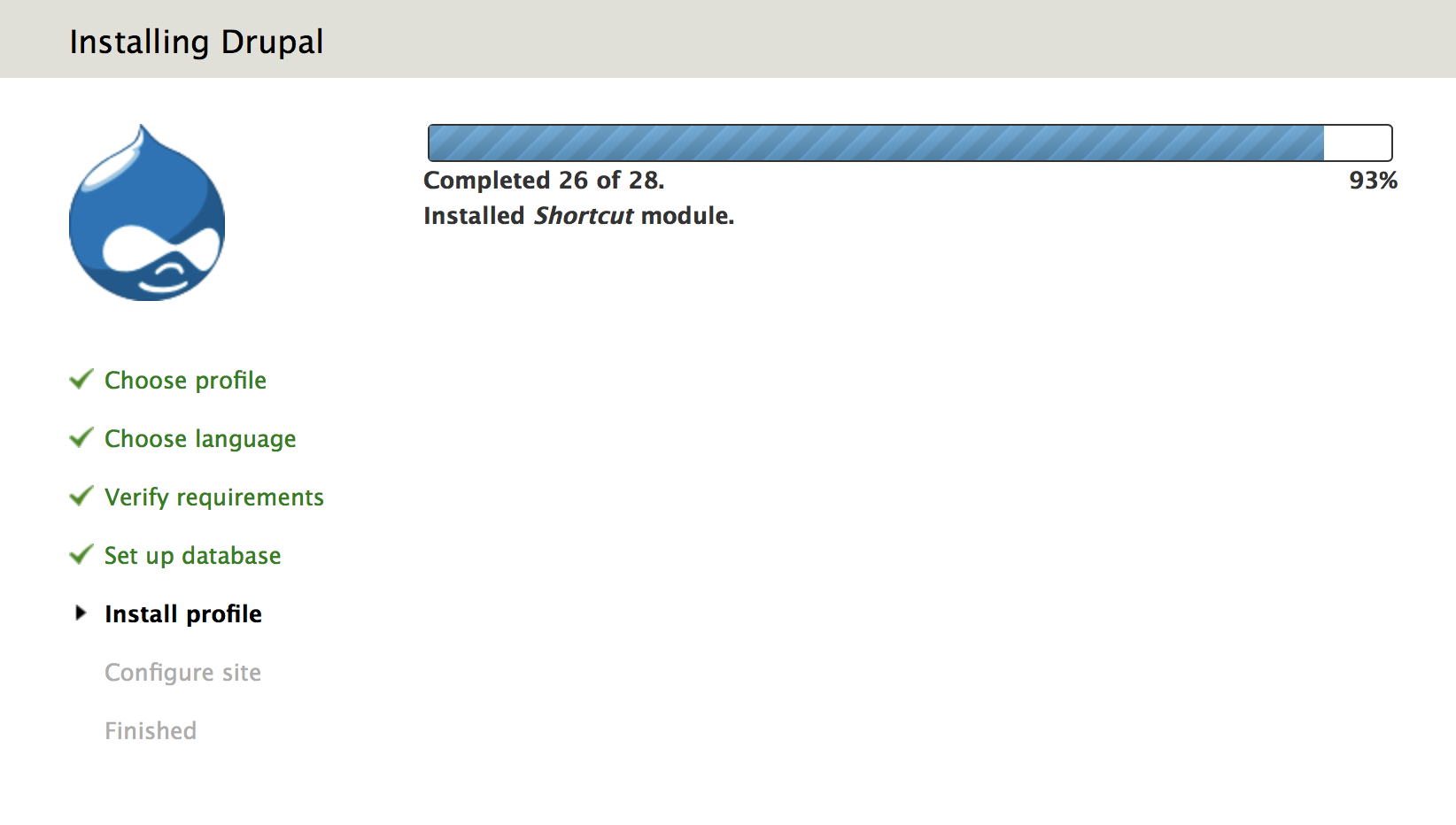


34. On the **Set up database** page, do the following:

1. Select **MySQL, MariaDB, Percona Server, or equivalent** as the database type.
2. In **Database name**, enter the name of the database on your DB instance.
3. In **Database username**, enter the username for your database.
4. In **Database password**, enter the password that you used when you created your DB instance.
5. Expand **ADVANCED OPTIONS**.
6. In **Host**, enter the endpoint of your DB instance. (To find this endpoint, select the DB instance in the Amazon RDS console. Do not include the **:3306**.)
7. Click **Save and continue**.

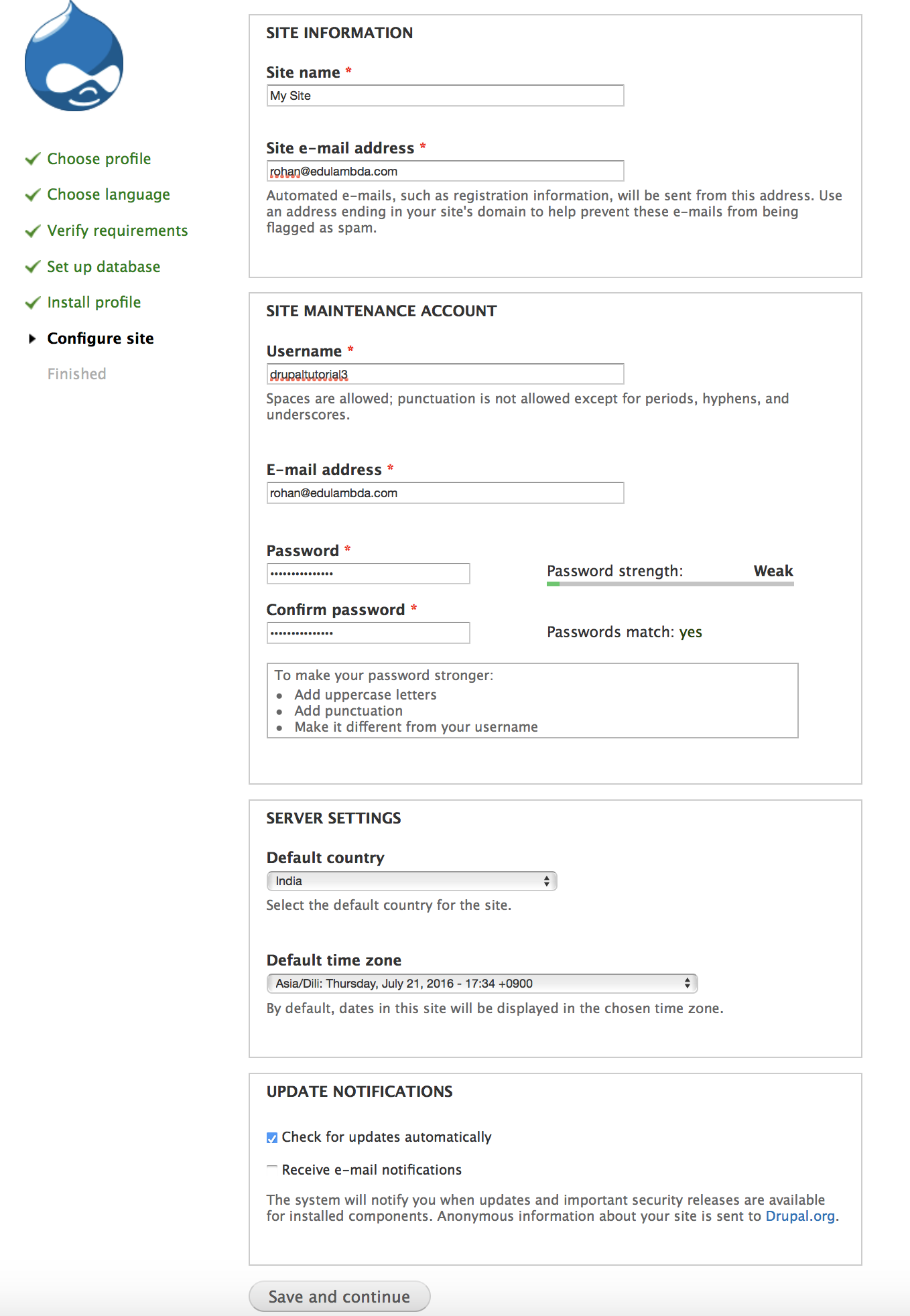


35. This will start the installation.

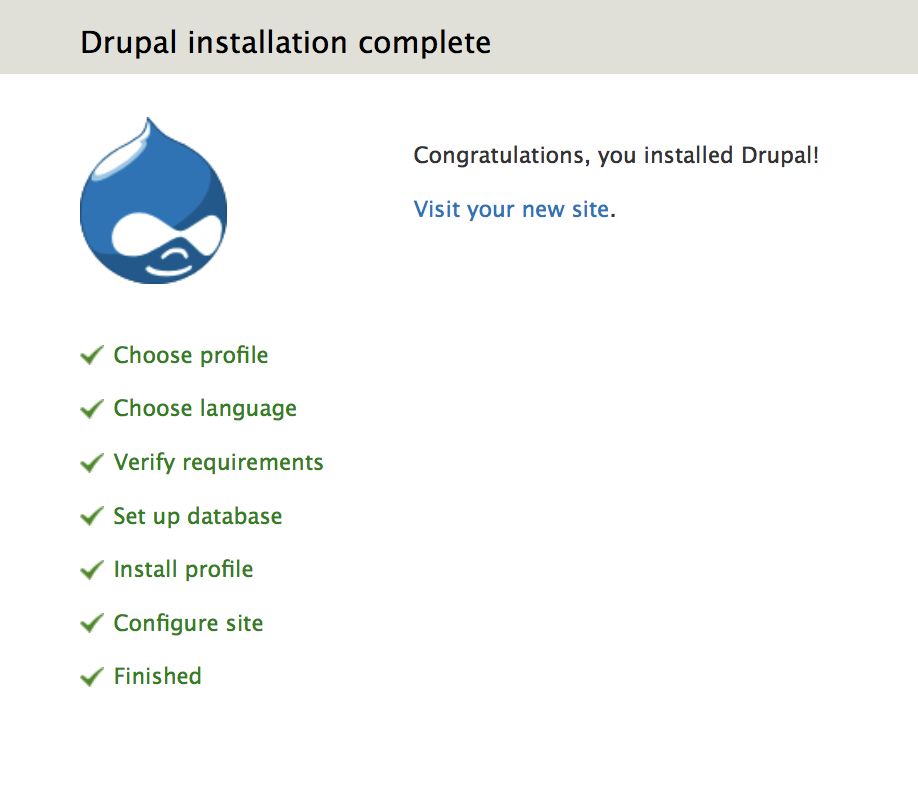


36. After the install completes, the **Configure site** page is displayed. On the **Configure site** page, do the following:

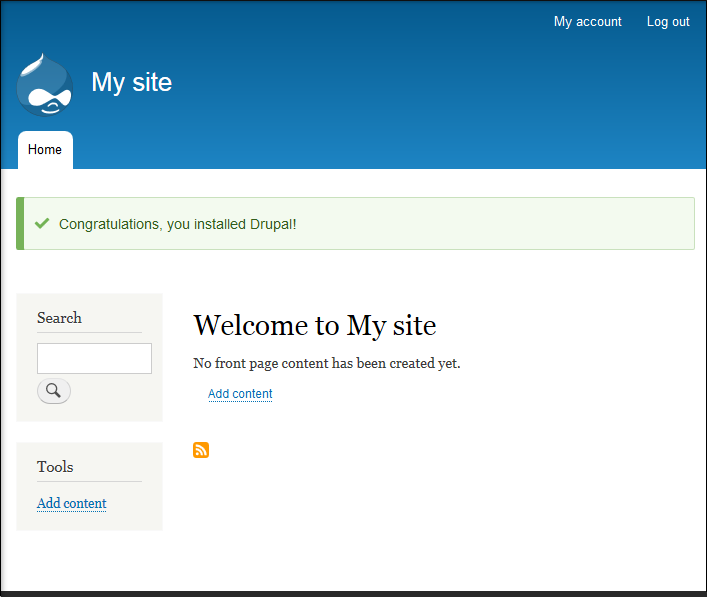
1. In **Site name**, enter a name, such as **My site**.
2. In **Site e-mail address**, enter an email address.
3. In **Username**, enter a user name.
4. In **Password**, enter a password. In **Confirm password**, enter the same password.
5. Under **Regional Settings**, select the default country and time zone.
6. Click **Save and continue**.



37. Click **Visit your new site** to view the web page.



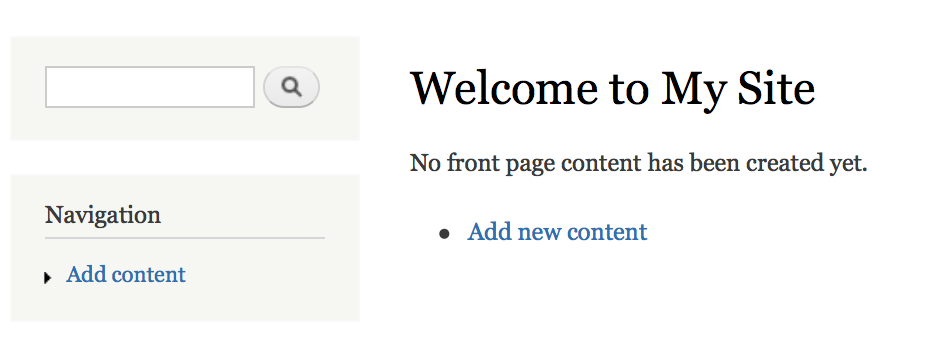
38. Upon success, your website is displayed.



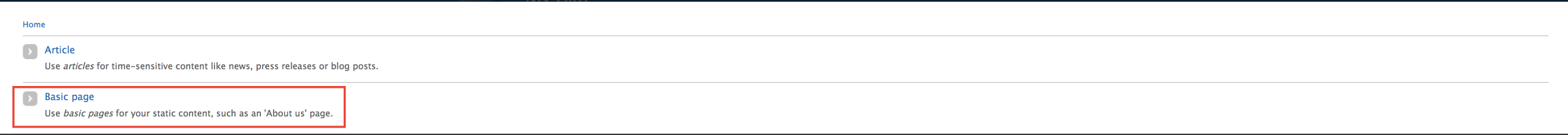
If you'd like, you can create a front page.

**To add the page**

39. From your Drupal website, click **Add content**.

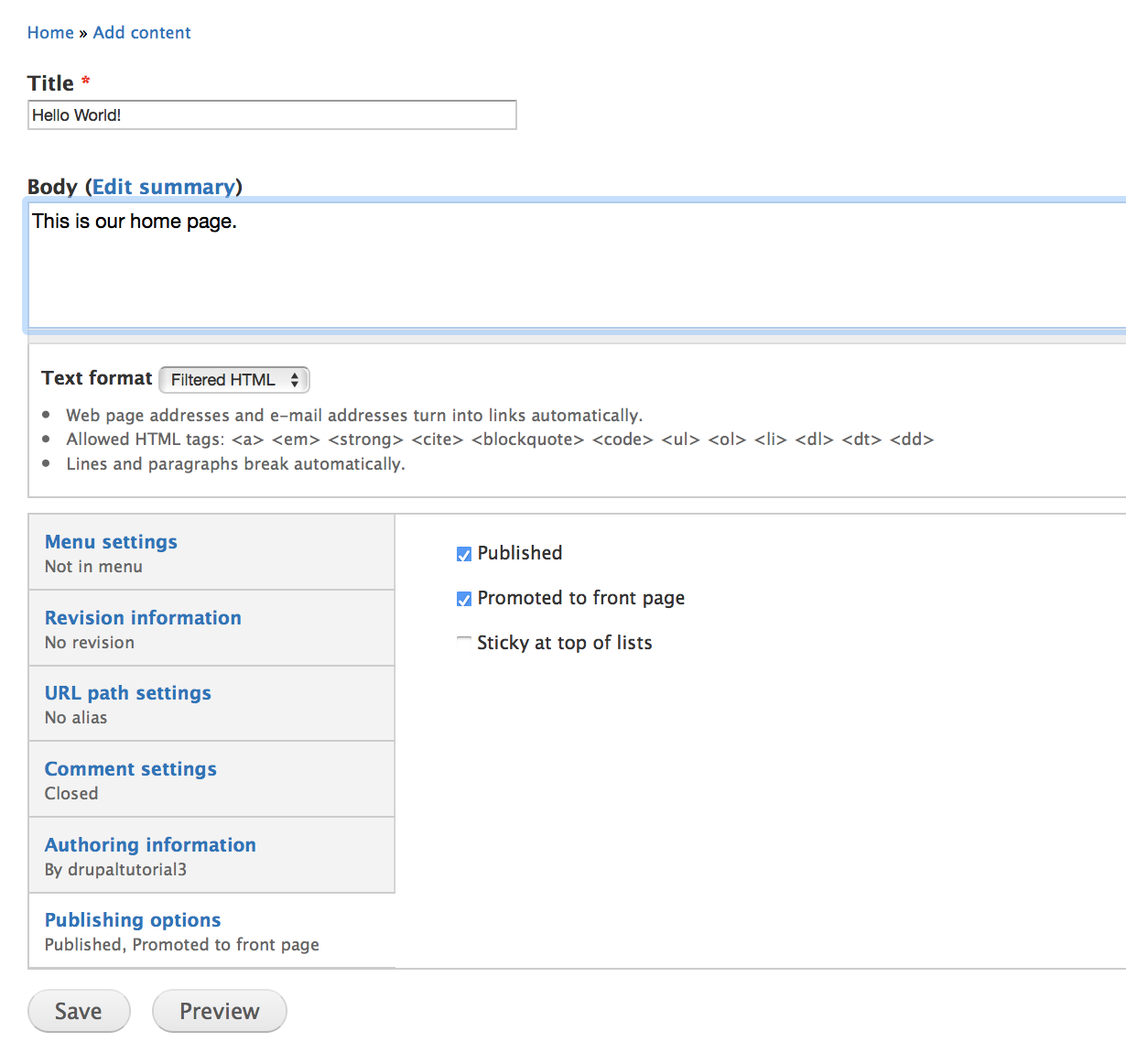


40. Choose **Basic Page**.

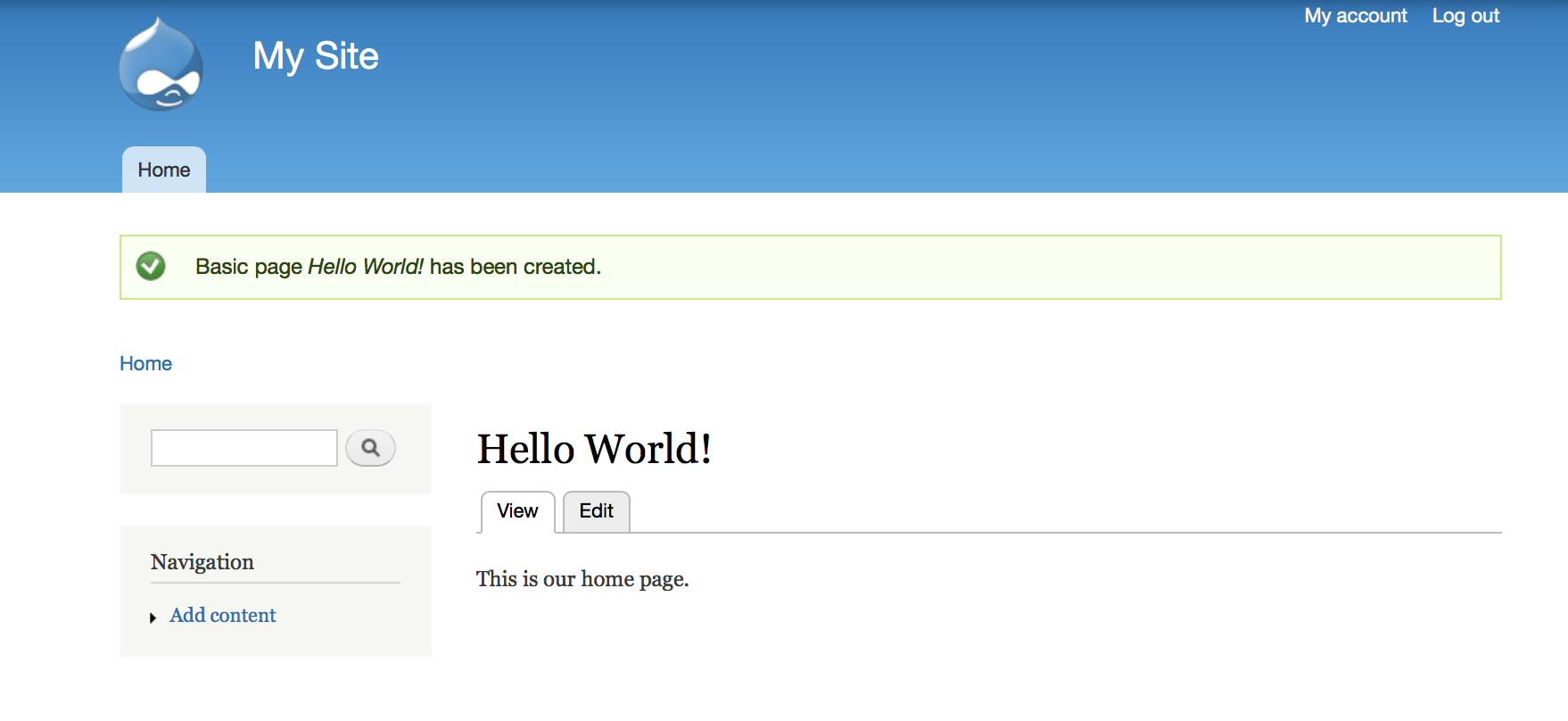


42. In **Title**, enter **Hello World!** Enter some text in **Body** field.

43. Expand **Promotion Options** and then select **Promoted to front page**.



44. Click **Save and publish**. Your updated website looks something like this.

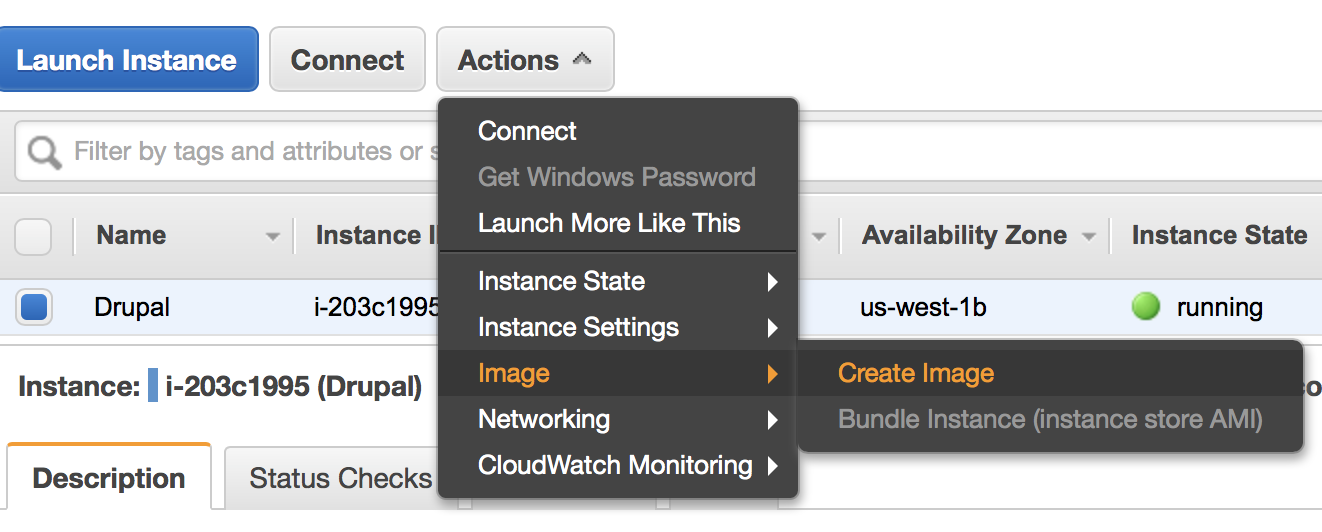


# Step 4: Create a custom AMI

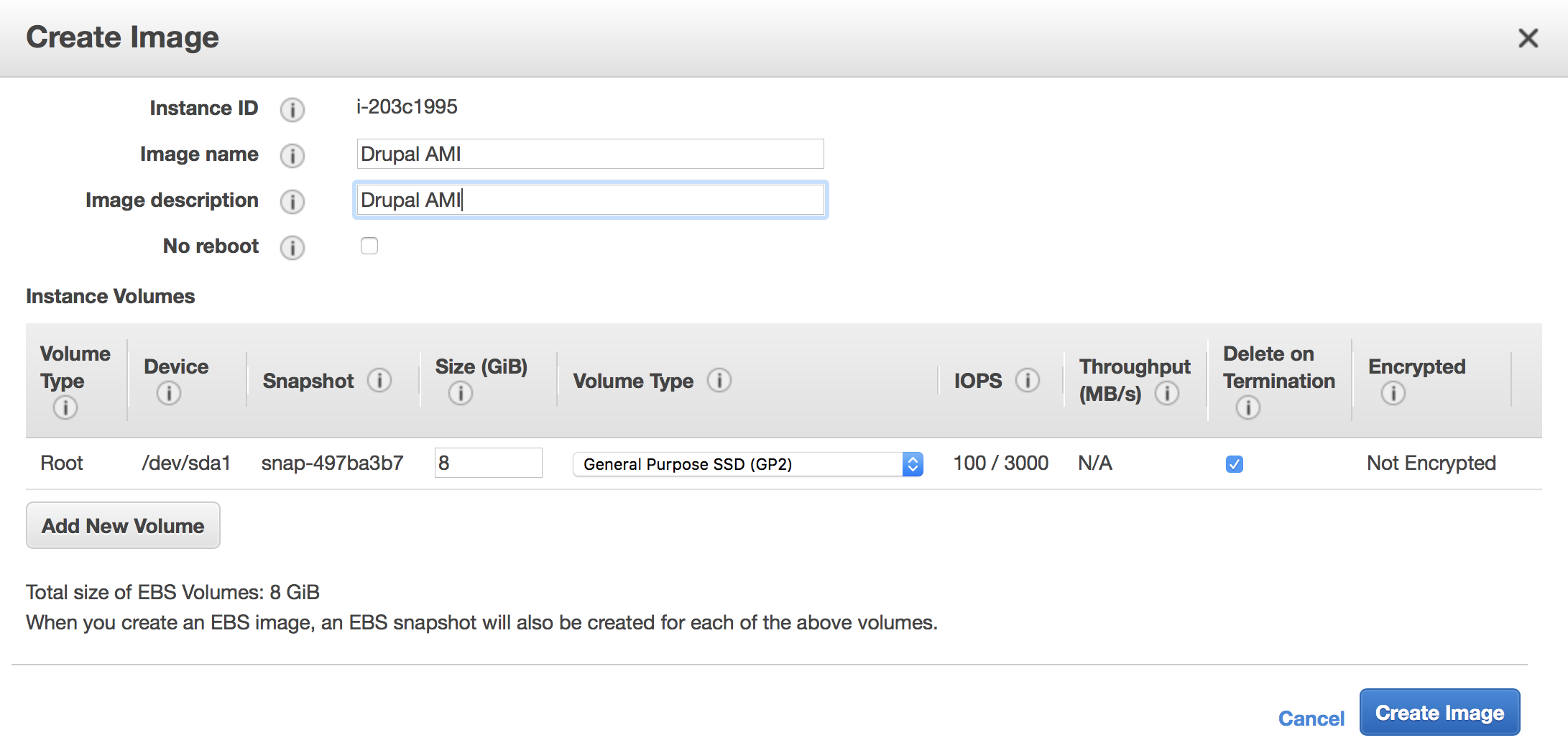
Now that you have customized your EC2 instance, you can create your own AMI. With your own AMI, you can quickly launch a new EC2 instance with the same configuration as this one.

45. Go to **Instance** Dashboard**.**

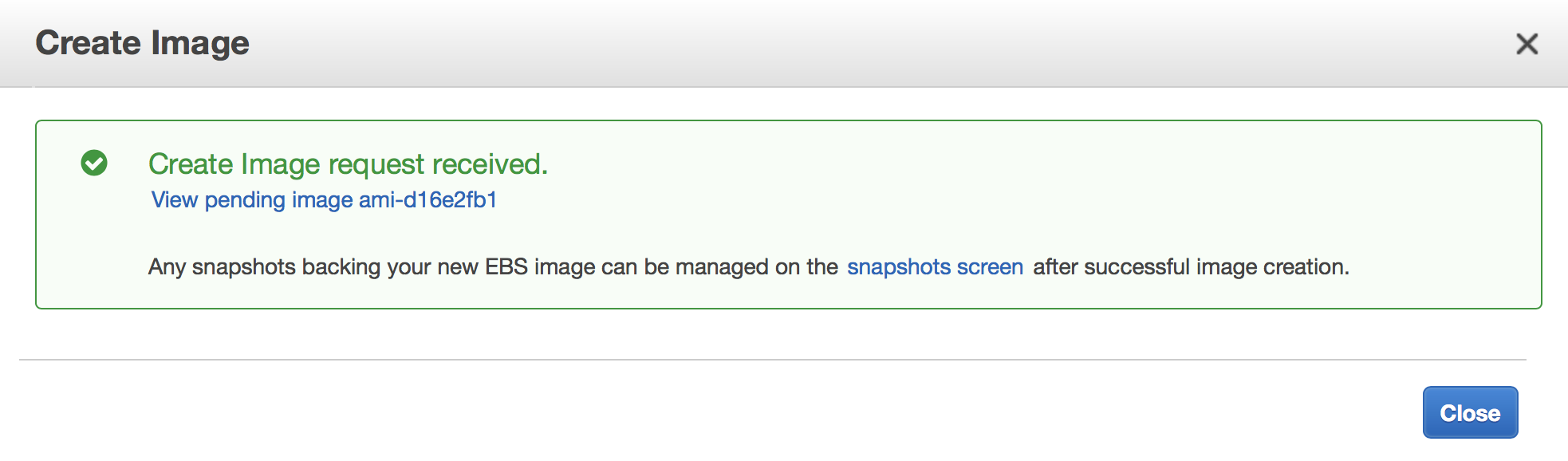
46. On the Instances page, select your Drupal instance, click **Actions**, select **Image**, and then click **Create Image.**



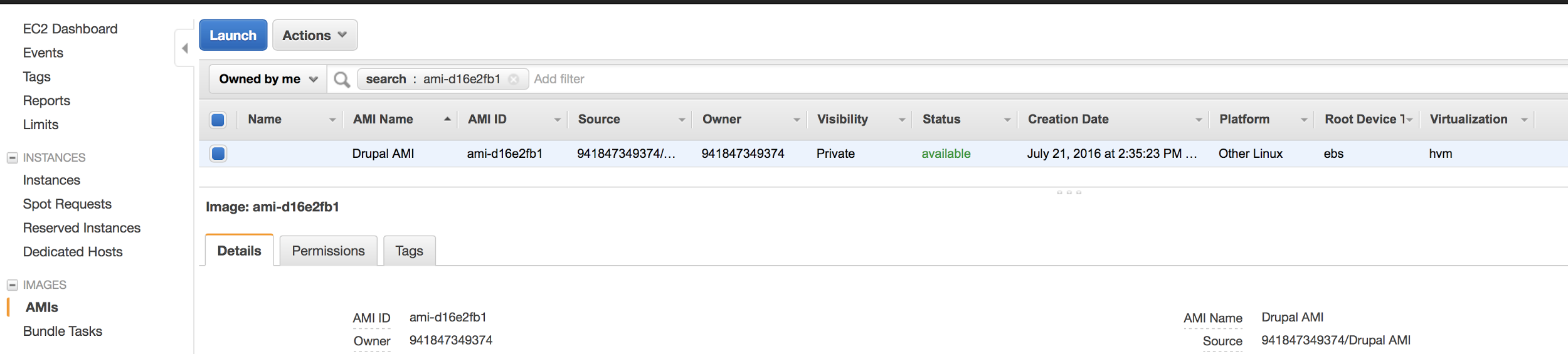
47. In the **Create Image** dialog box, specify a unique image name and an optional description of the image (up to 255 characters), and then click **Create Image**.



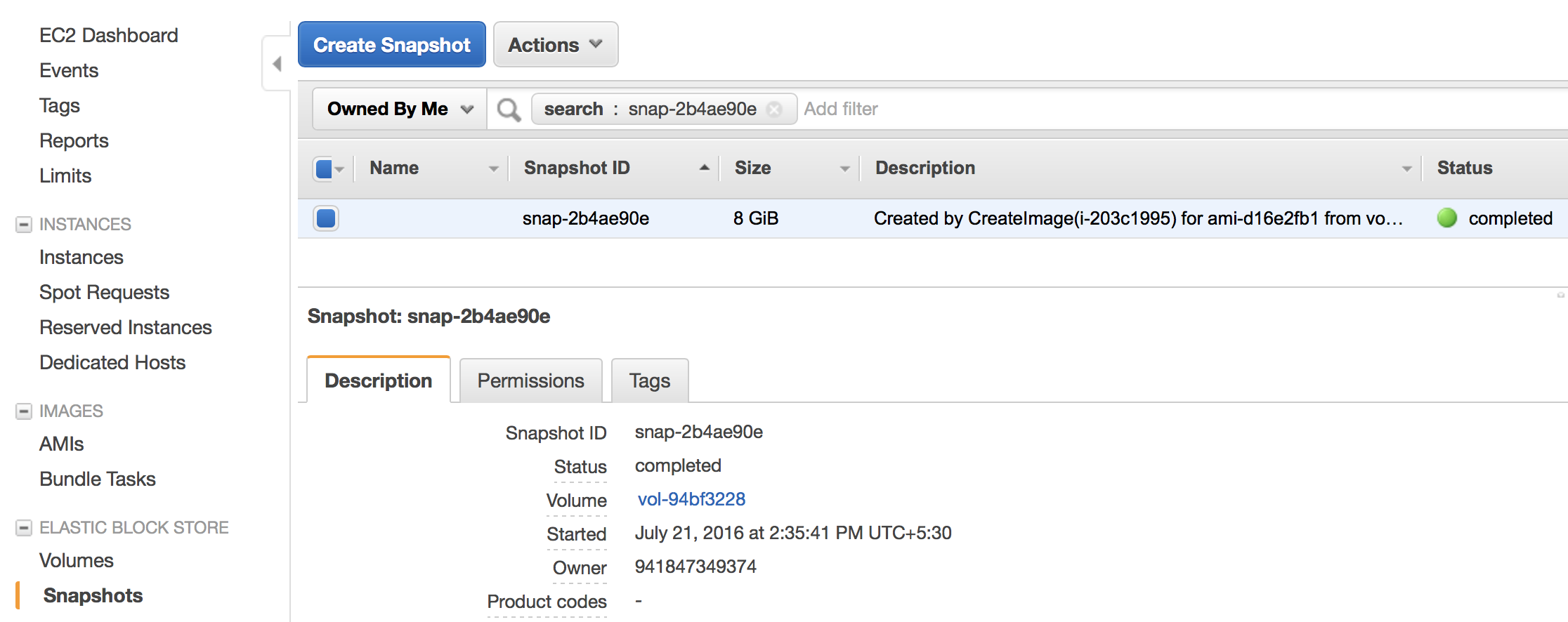
48. Click **Close**.



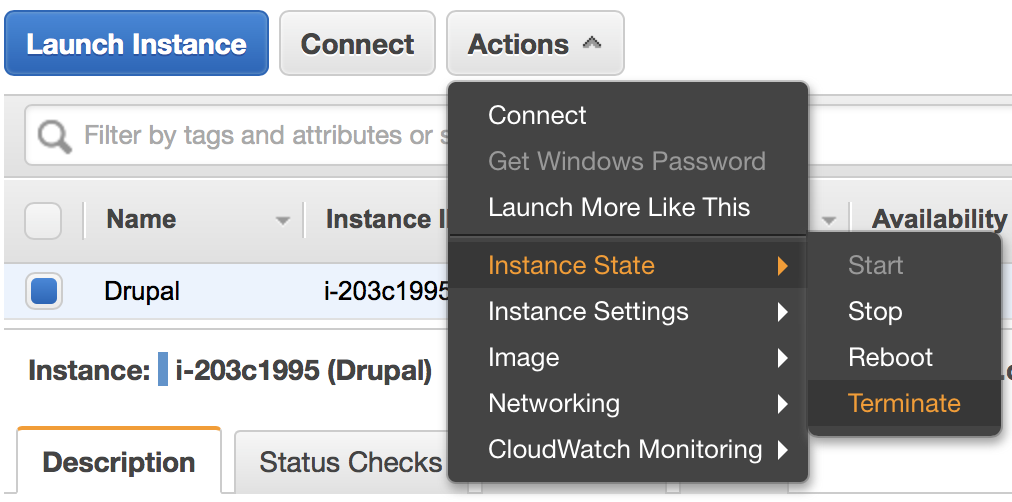
49. To view the status of your AMI, go to the **AMIs** page. While the AMI is being created, its status is pending.



50. If you go to the **Snapshots** page, you'll see that we created a snapshot that is used to create the root device volume of any instance that you launch using your new custom AMI.



51. Now that you have a custom AMI, you no longer need the instance that you created it from, because we'll use Auto Scaling to launch new instances in the next step. To terminate the instance, go to the **Instances** page, select it, click **Actions**, select **Instance State**, and then click **Terminate**, and then click **Yes, Terminate**.



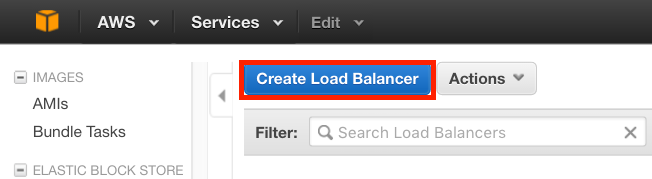
# Step 5: Scale and Load-Balance Your Web App

Now, you'll set up the basic infrastructure for a web app that is load-balanced and auto-scaled with a desired capacity of two instances. We configure Auto Scaling to keep the group at its initial size. When you deploy your own web app, you should follow best practices to ensure that your app can run even if it loses one Availability Zone. You can also create additional policies, such as a scale-in policy.

52. In the console’s left navigation pane, click **Load Balancers**. You may need to scroll down to see the link.



53. Click **Create Load Balancer**.

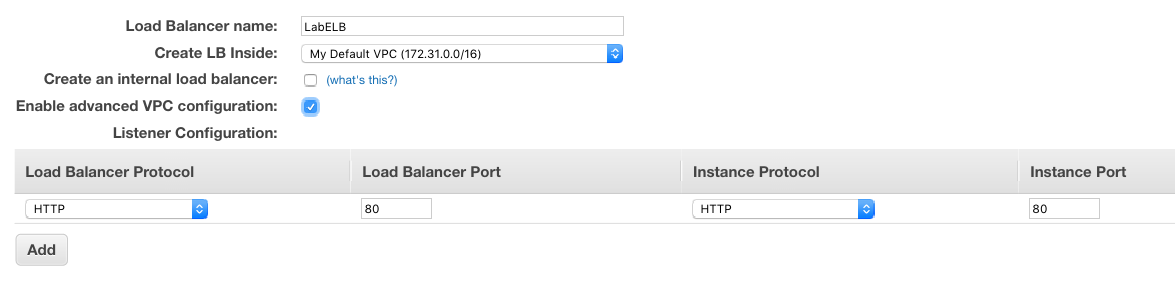


The name of your load balancer must be unique within your set of load balancers for the region, can have a maximum of 32 characters, can contain only alphanumeric characters and hyphens, and must not begin or end with a hyphen.

54. For **Create LB inside**, select the same network that you selected for your instances: EC2-Classic or a specific VPC.

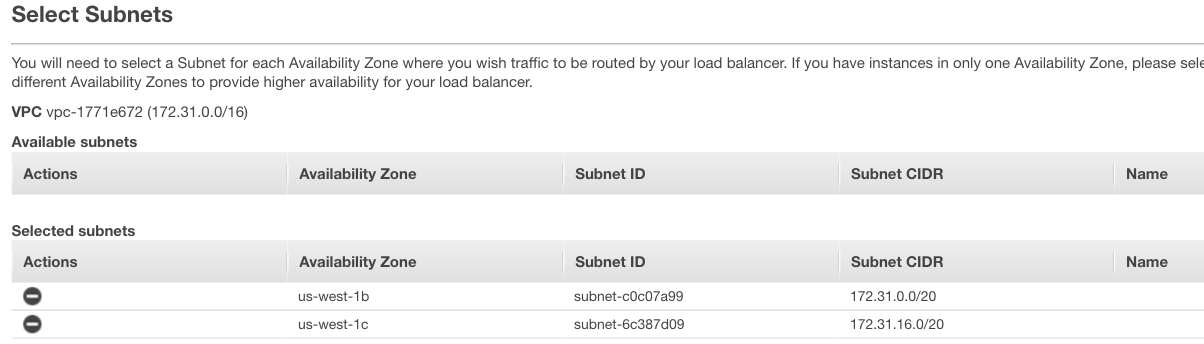
55. [Default VPC] If you selected a default VPC and would like to choose the subnets for your load balancer, select **Enable advanced VPC configuration**.

56. Leave the default listener configuration.



57. Under **Select Subnets**, all the subnets in which our web servers have been launched into.

The available subnets for the VPC for your load balancer are displayed under **Available Subnets**. Select public subnets that are in the same Availability Zones as your instances. Click the icon in the **Action** column for each subnet to attach. These subnets are moved under **Selected Subnets**. You can select at most one subnet per Availability Zone. If you select a subnet from an Availability Zone where there is already a selected subnet, this subnet replaces the currently selected subnet for the Availability Zone.



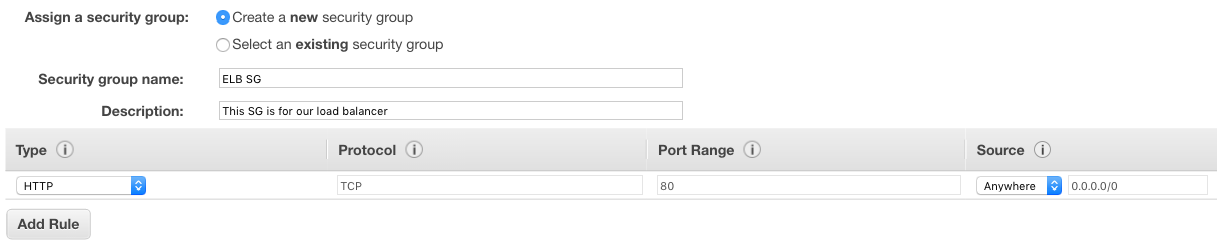
58. Click **Next: Assign Security Groups**.

## Assign Security Groups to Your Load Balancer in a VPC

If you selected a VPC as your network, you must assign your load balancer a security group that allows inbound traffic to the ports that you specified for your load balancer and the health checks for your load balancer.

59. On the **Assign Security Groups** page, select **Create a new security group**.

60. Enter a name and description for your security group, or leave the default name and description. This new security group contains a rule that allows traffic to the port that you configured your load balancer to use. Over here, we would be allowing **HTTP** traffic from anywhere i.e. 0.0.0.0/0.



61. Click **Next: Configure Security Settings**.

62. Skip Step 3: Configure Security Settings by clicking on **Next: Configure Health Check**.

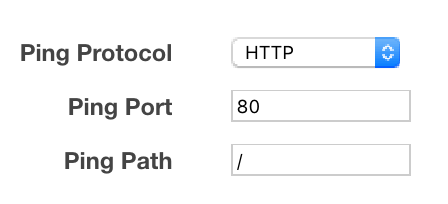
## Configure Health Checks for Your EC2 Instances

Elastic Load Balancing automatically checks the health of the EC2 instances for your load balancer. If Elastic Load Balancing finds an unhealthy instance, it stops sending traffic to the instance and reroutes traffic to healthy instances. In this step, you customize the health checks for your load balancer.

**To configure health checks for your instances**

63.On the **Configure Health Check** page, do the following:

1. Leave **Ping Protocol** set to its default value, HTTP.
2. Leave **Ping Port** set to its default value, 80.
3. In the **Ping Path** field, replace the default value with a single forward slash ("/"). Delete the text **index.html**.
4. Leave the other fields set to their default values.

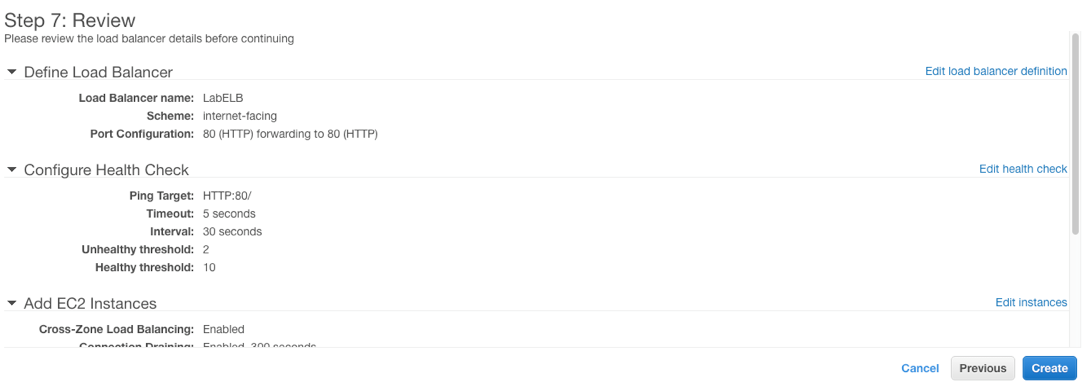


64. Click **Next: Add EC2 Instances**.

65. Click **Next: Add Tags** to skip the **Add Instances to Load Balancer** page. We'll use Auto Scaling to add our EC2 instances to the load balancer.

67. Click **Review and Create** to skip adding tags.

68. Review your settings and then click **Create**. After the load balancer is created, click **Close**.



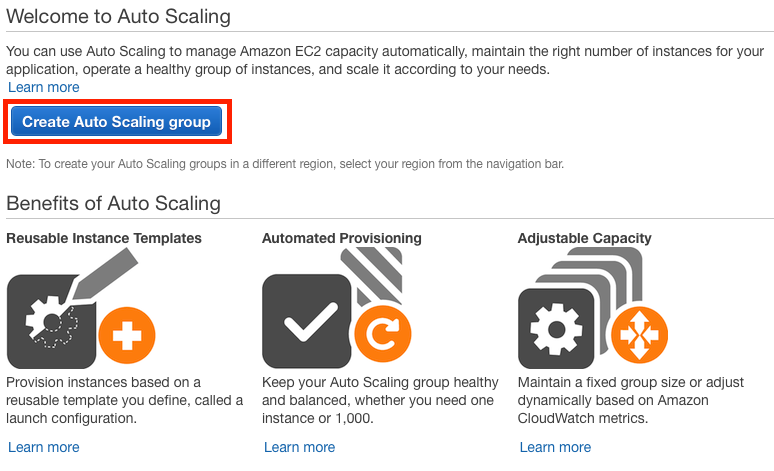
## Create a Launch Configuration

A launch configuration specifies the type of EC2 instance that Auto Scaling creates for you. You create the launch configuration by including information such as the Amazon Machine Image (AMI) ID to use for launching the EC2 instance, the instance type, key pairs, security groups, and block device mappings, among other configuration settings.

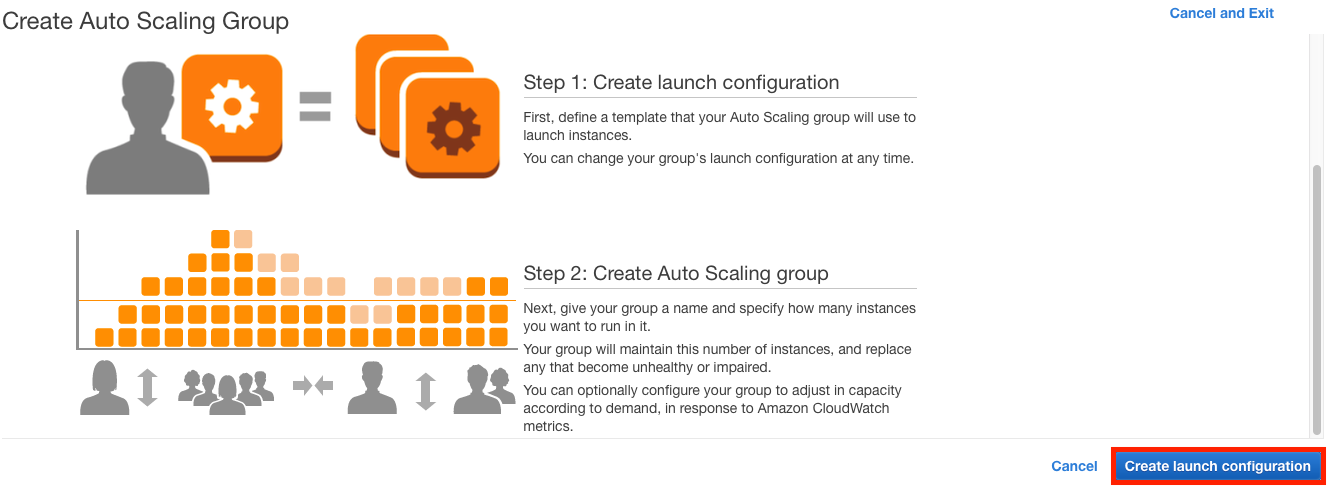
69. On the navigation pane, under **Auto Scaling**, choose **Launch Configurations**.



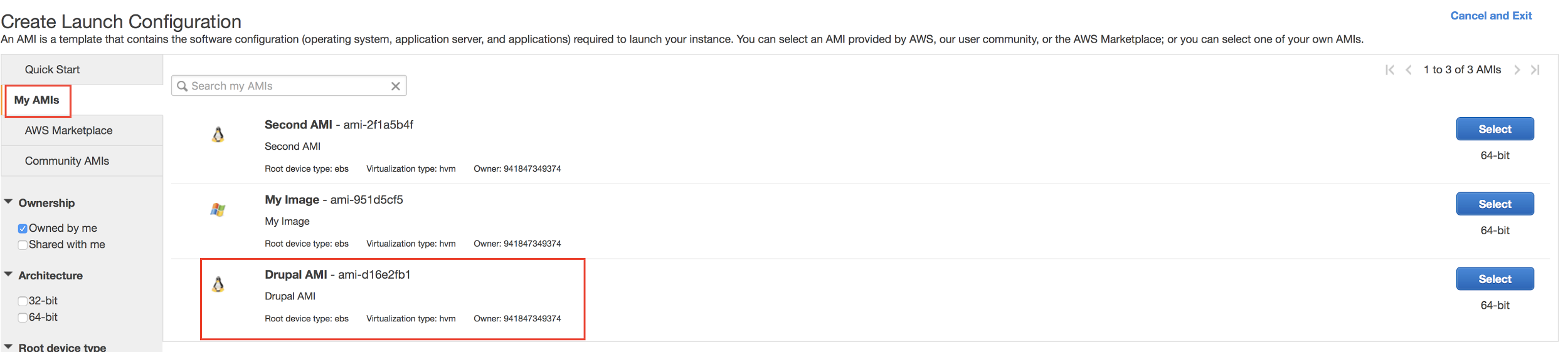
70. On the **Welcome to Auto Scaling** page, choose **Create Auto Scaling group**.



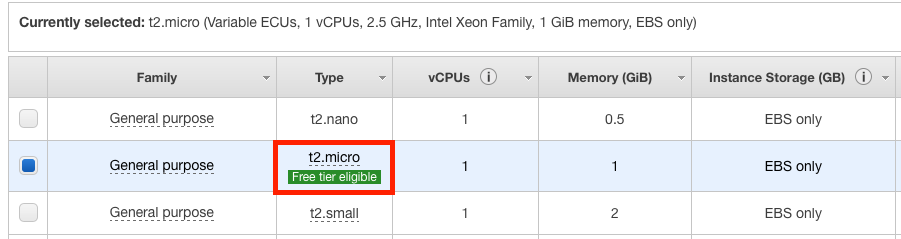
71. On the **Create Auto Scaling Group** page, choose **Create launch configuration**.



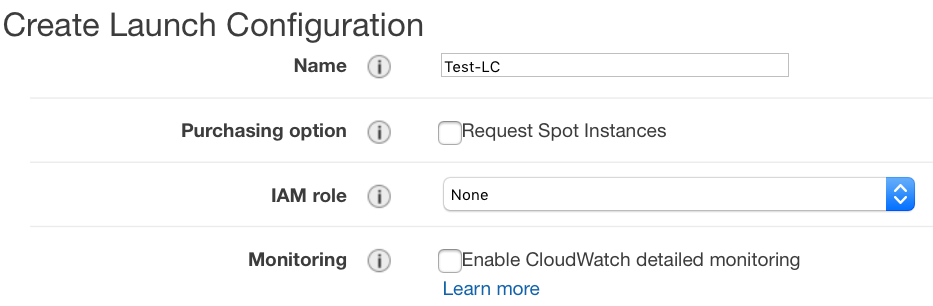
72. On the **Choose AMI** page, select the **My AMIs** tab, and then select the AMI that you created in Create a Custom AMI.



73. On the **Choose Instance Type** page, select a hardware configuration for your instance. We recommend that you keep the default, a *t2.micro* instance. Choose **Next: Configure details**.



74. On the Configure Details page, type a name for your launch configuration in the **Name** section.



75. In **User data**, select **As text** and then enter the following script.

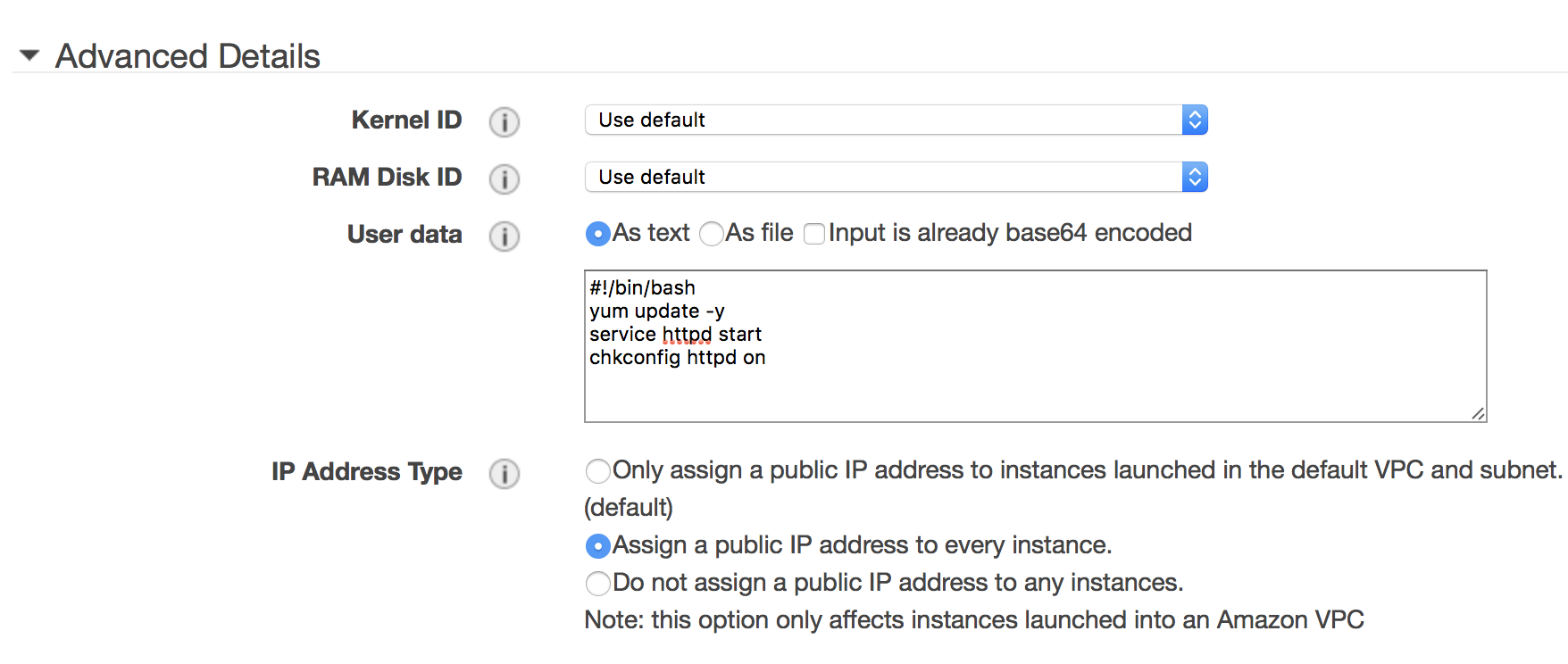
#!/bin/bash

yum update -y

service httpd start

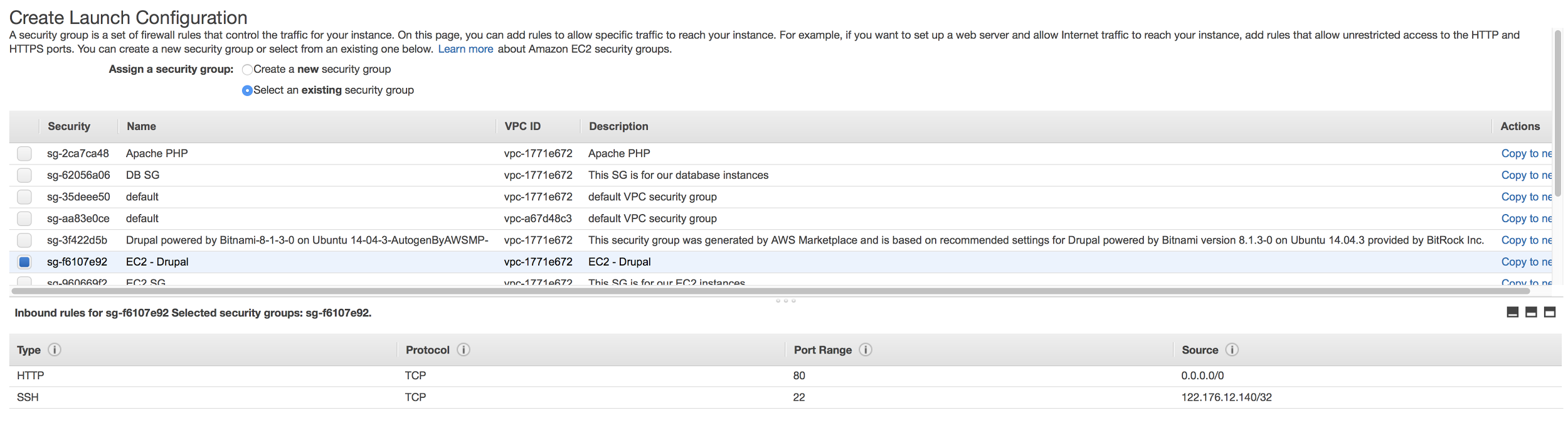
chkconfig httpd on

76. Select **Assign a public IP address to every instance**.

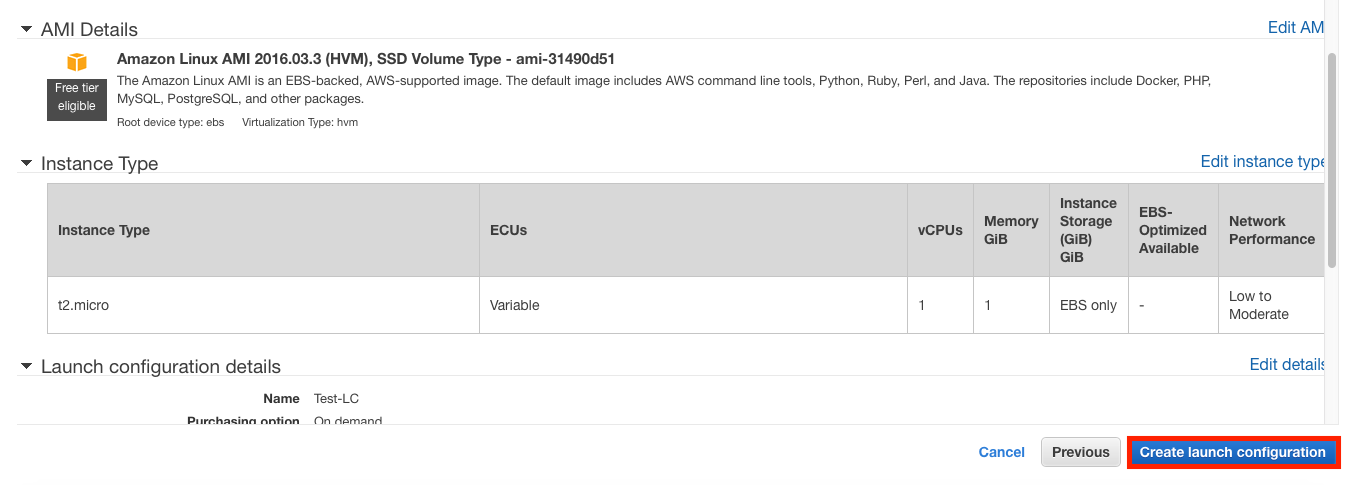


77. Keep other settings to default and click **Skip to review**.

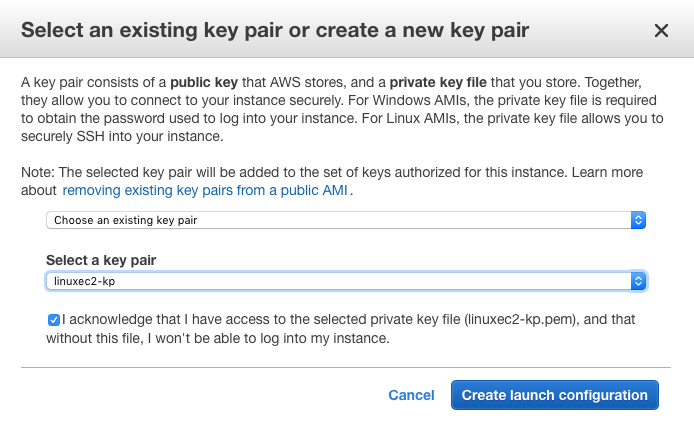
78. On the **Review**page, choose **Edit security groups**. Follow the instructions to choose an existing security group **(EC2-Drupal)**, and then choose **Review**.



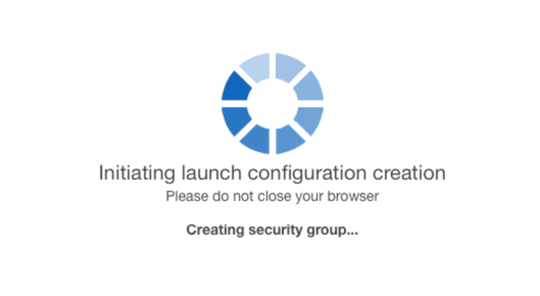
79. On the **Review**page, choose **Create launch configuration**.



80. On the **Select an existing key pair or create a new key pair** page, select one of the listed options.

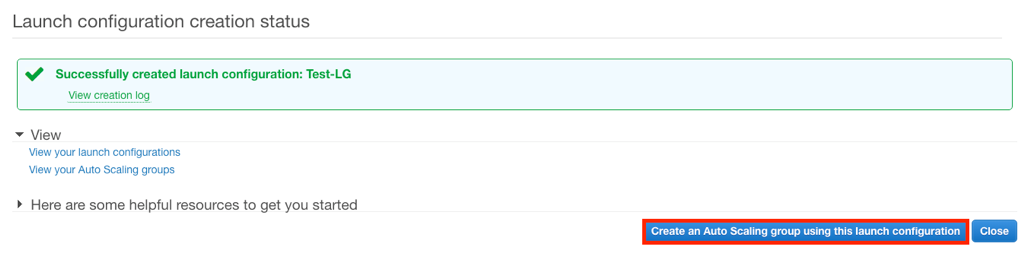


81. Choose **Create launch configuration**. This initiates the process to create this Launch Configuration.



## Create an Auto Scaling Group

82. Click **Create an Auto Scaling group using this launch configuration**.

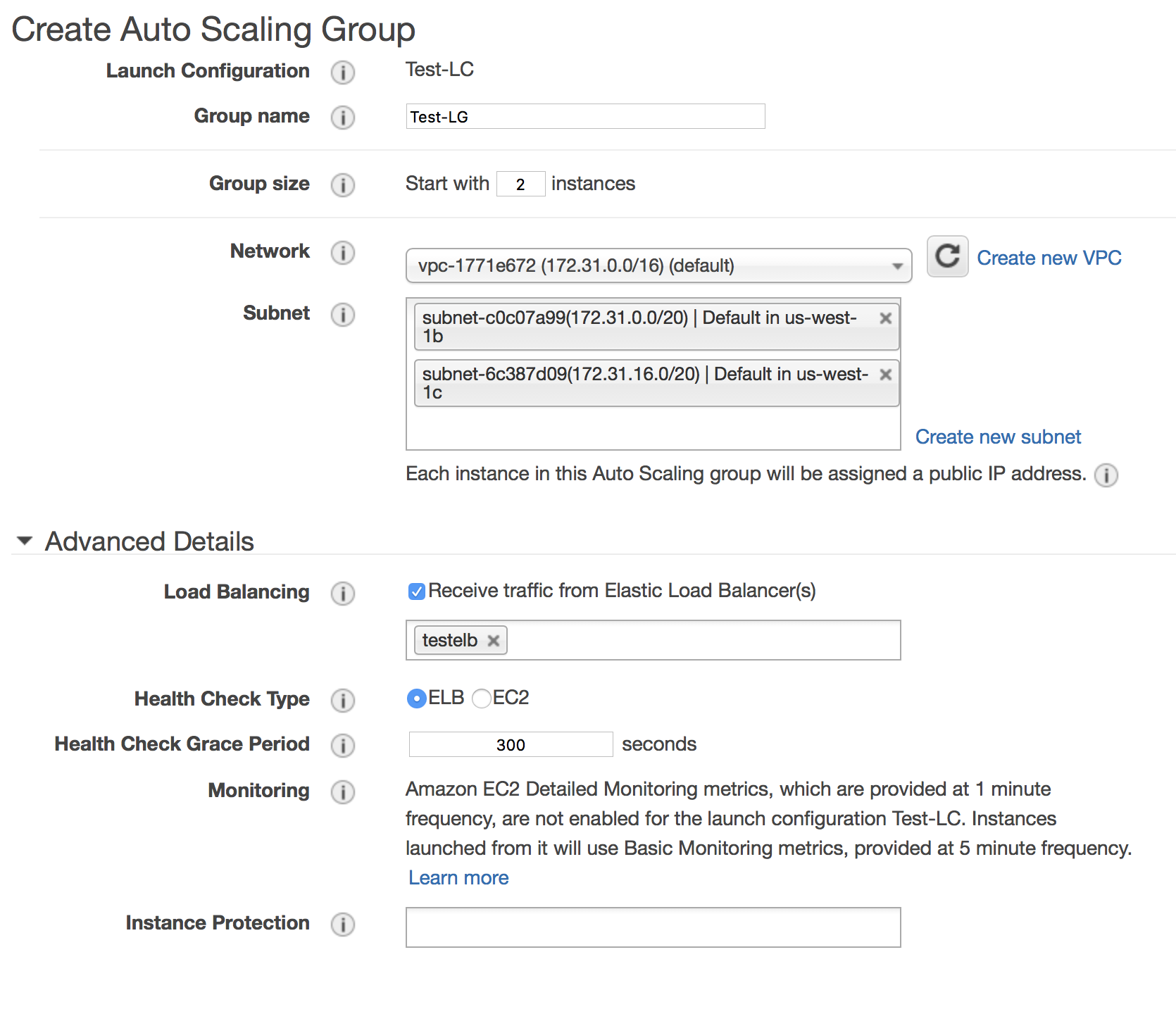


83. For **Group name**, type a name for your Auto Scaling group.

84. Keep **Group size**set to the default value of **2** instances for this tutorial.

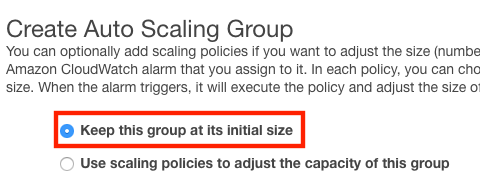
85. Choose VPC in **Network** and its associated subnets in **Subnet.** In this example, we select our default VPC and all its subnets. This is a best practice for building fault-tolerant apps. If one Availability Zone experiences an outage, traffic will be routed to the other Availability Zone.

86. Expand **Advanced Details**. Select **Receive traffic from Elastic Load Balancer(s)**. Select your load balancer from the text field.



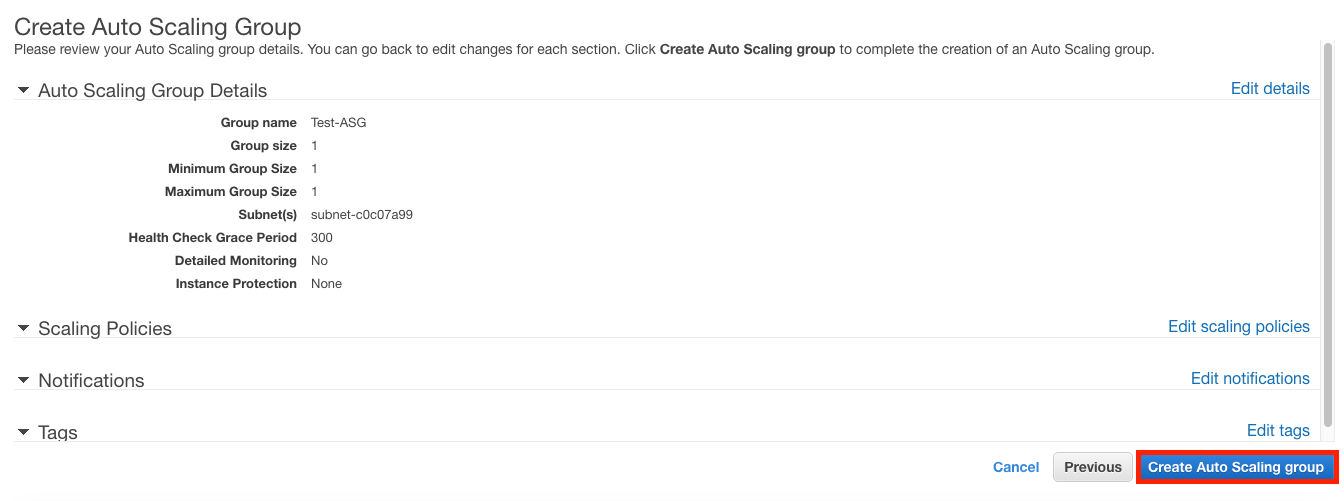
87. Choose **Next: Configure scaling policies**.

88. On the **Configure scaling policies** page, **select Keep this group at its initial size** and choose **Review**.

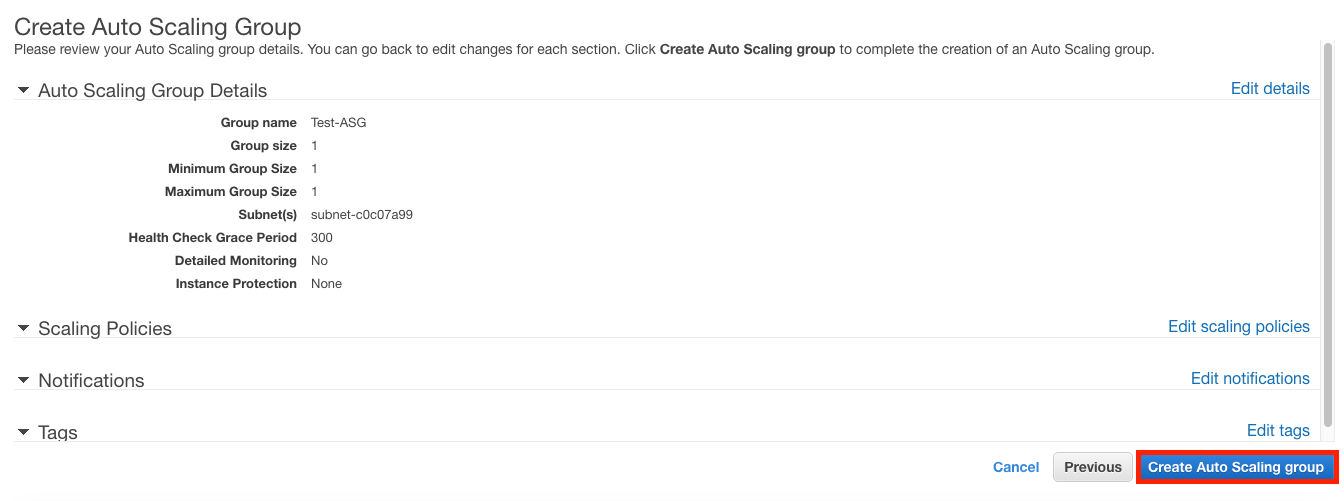
\\

89. On the **Review**page, choose **Create Auto Scaling group**.

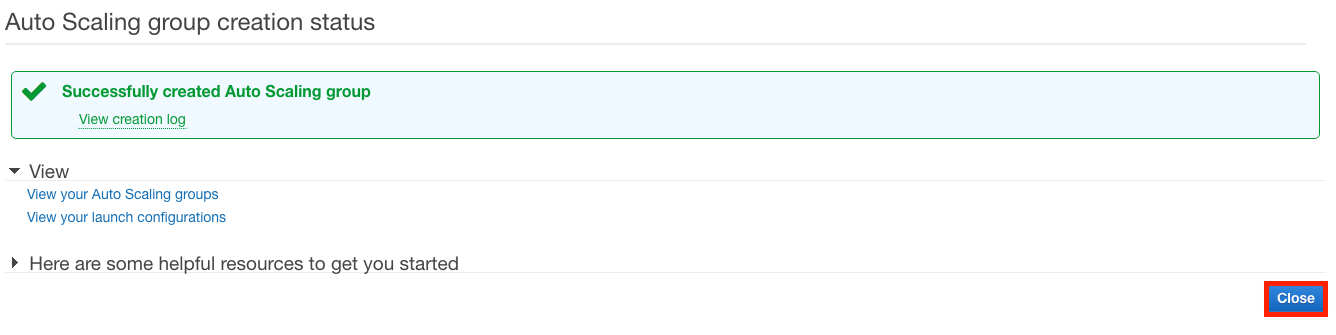
\



90. On the **Review**page, choose **Create Auto Scaling group**.

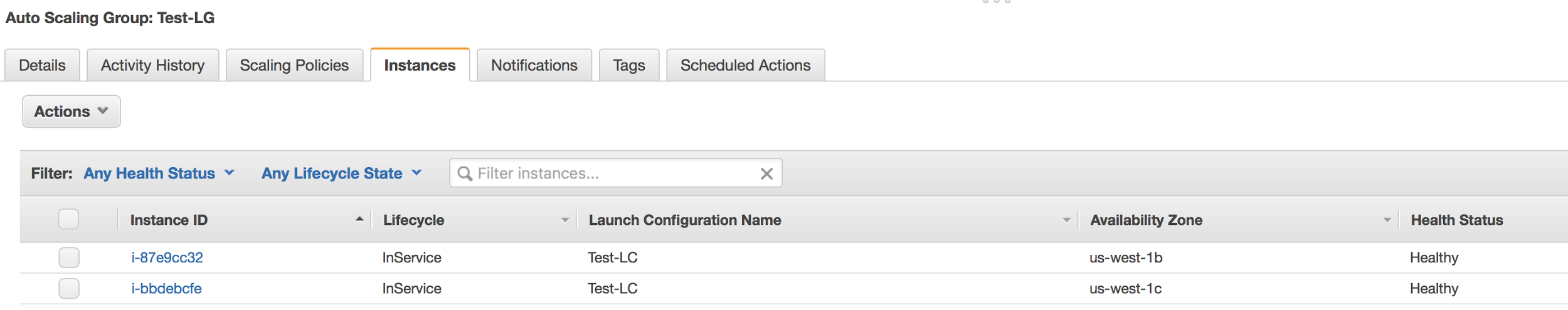


91. On the **Auto Scaling group creation status** page, choose **Close**.



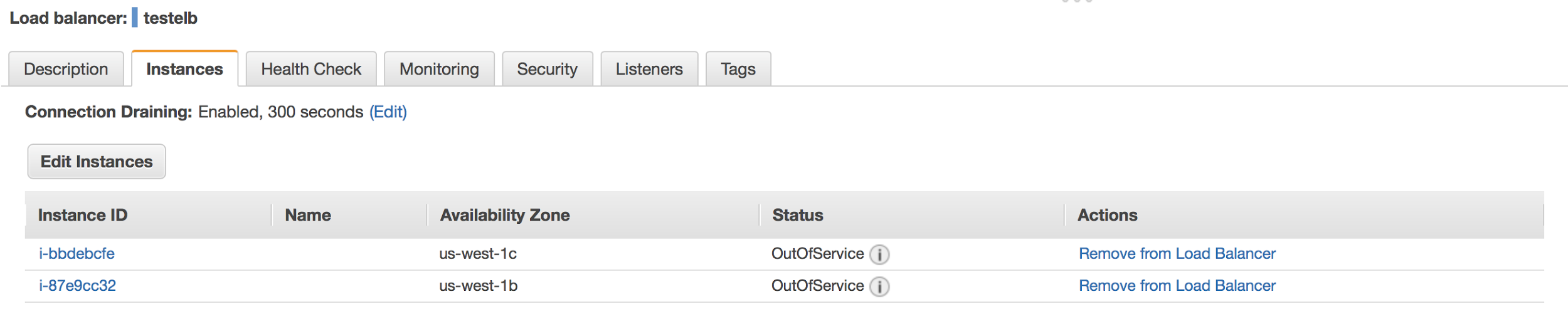
## Test Your Load Balancer

92. First, verify that your instances are ready. From the Auto Scaling Groups page, select your Auto Scaling group, and then select the Instances tab.



Initially, your instances are in the Pending state. When their states are InService, they are ready for use.

93. Next, verify that your instances are registered with the load balancer. From the **Load Balancers** page, select your load balancer, and then select the **Instances** tab.



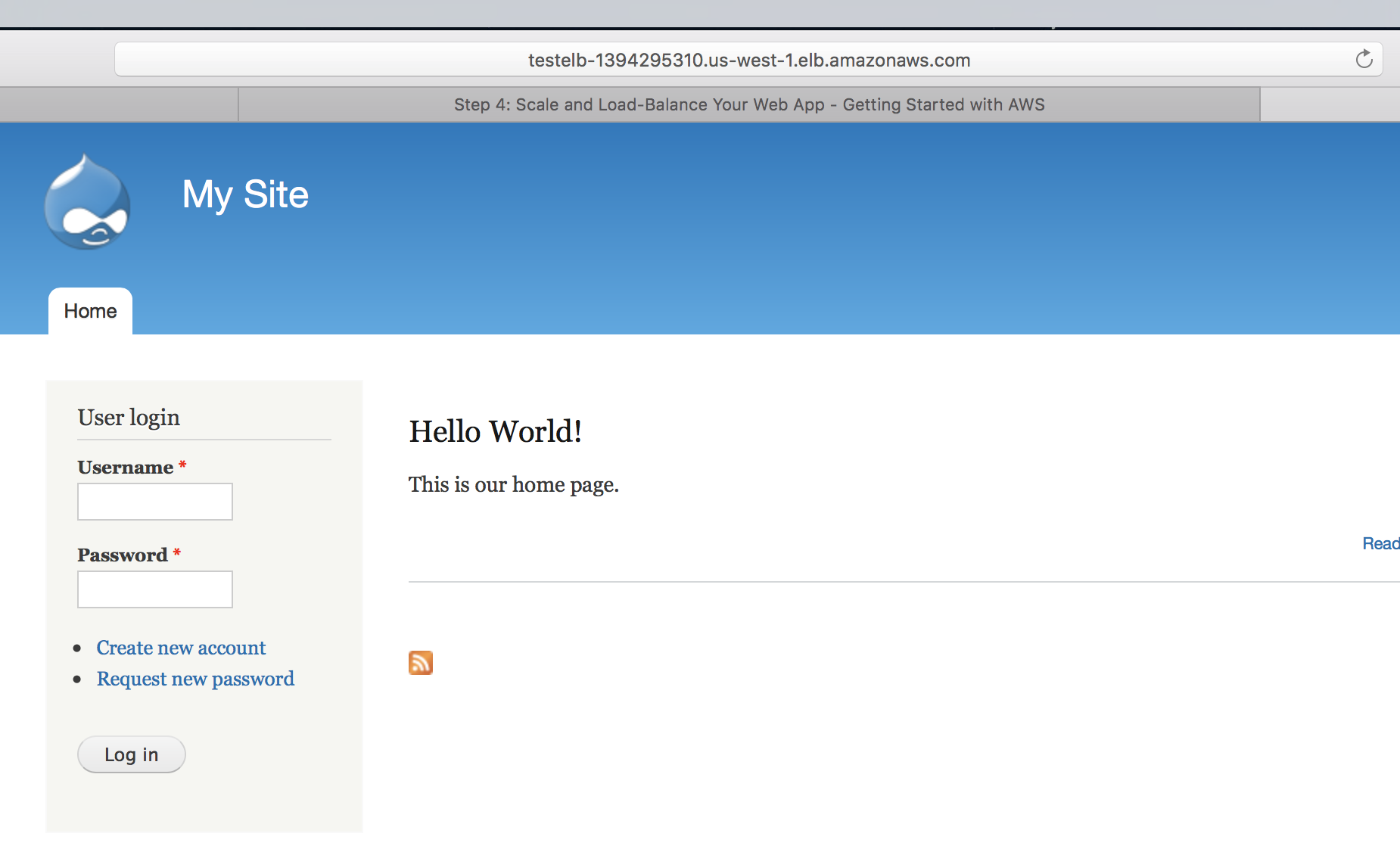
If the state of your instances is OutOfService, it's possible that they are still registering. When their states are InService, they are ready for use. After your instances are ready, you can test your load balancer as follows.

**To test your load balancer**

94. From the **Load Balancers** page, select your load balancer.

95. On the **Description** tab, locate the DNS name.

96. In a web browser, paste the DNS name for the load balancer into the address bar and press Enter. You'll see your website displayed.



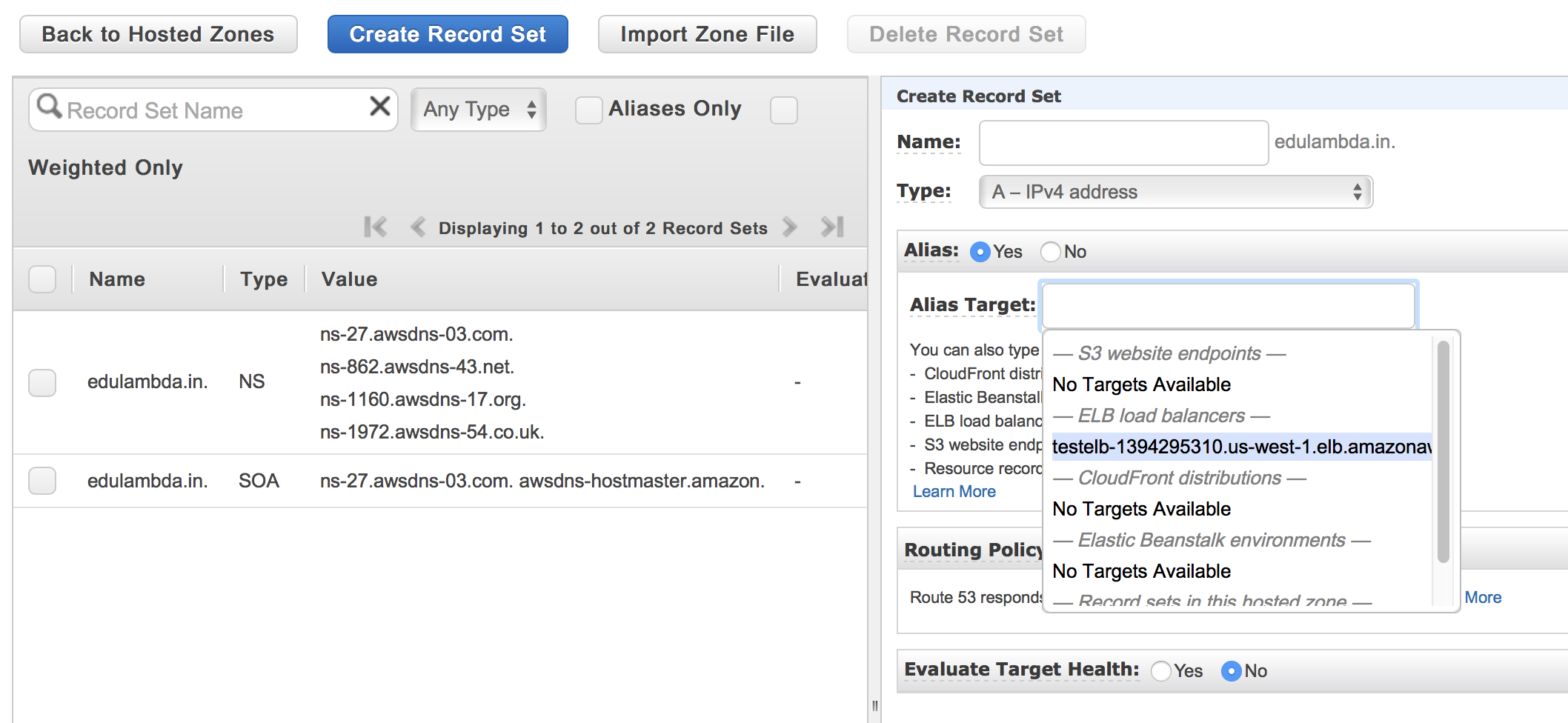
You will get to another instance’s web page each time you refresh the page.

# Step 6: Associate a Domain Name with Your Website

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

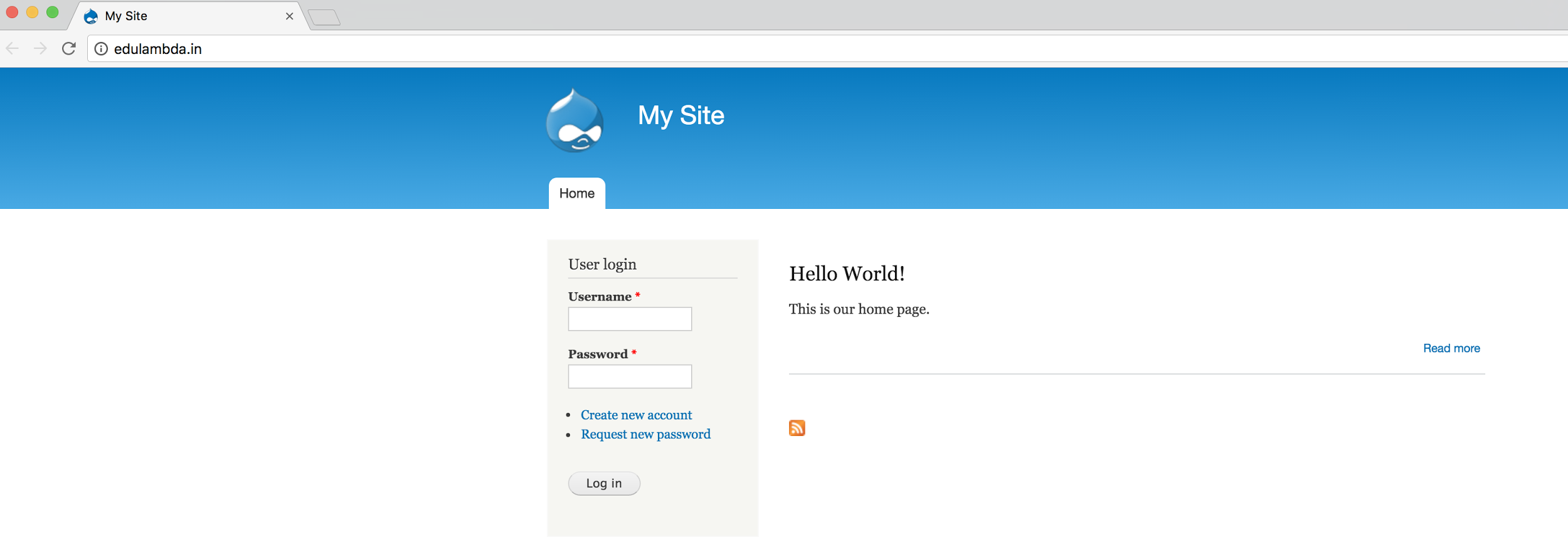
97. Create a hosted zone for your domain.

98. Create a record set pointing towards the load balancer.



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

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



# Step 7: Clean Up

After completing this tutorial, be sure to delete the AWS resources that you created so that you no longer accrue charges.