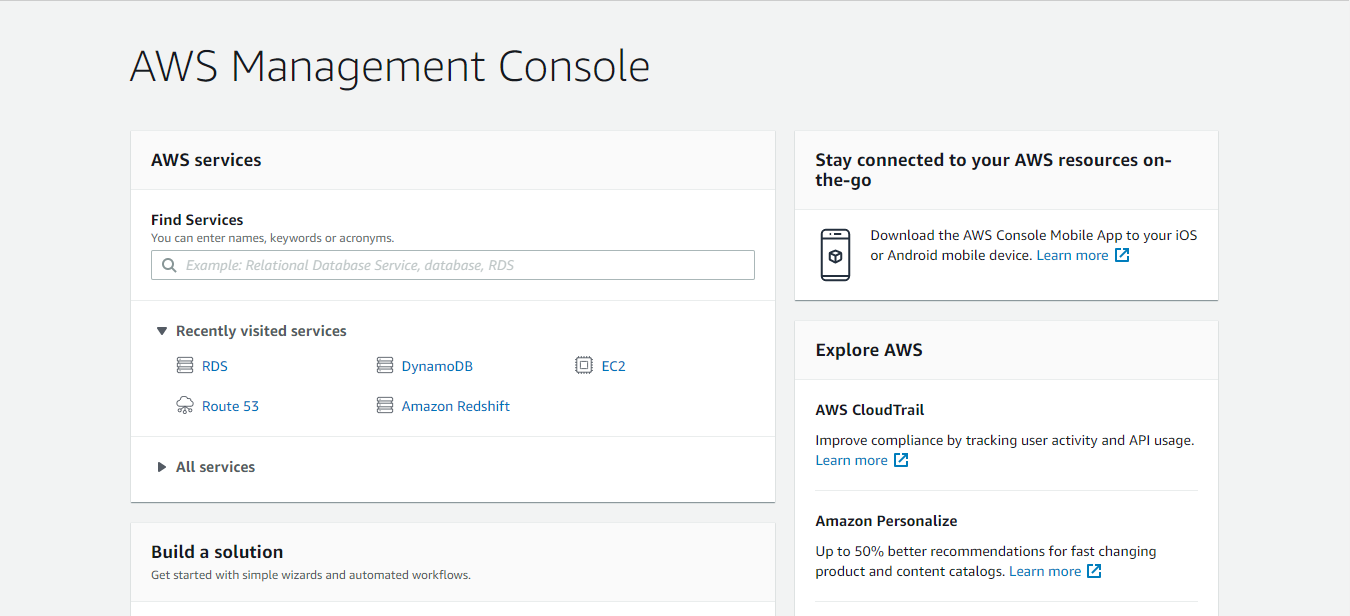
**Course-End Project: Configure and Connect a MySQL Database Instance with a Web Server**

This section will guide you to:

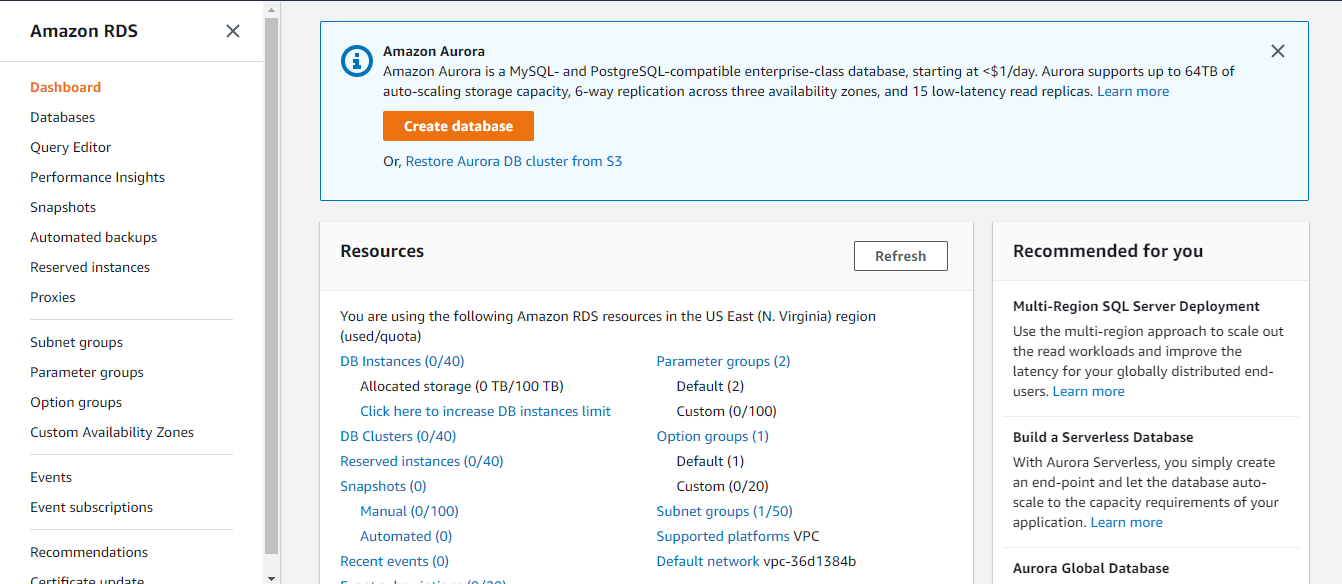
* Open the AWS console
* Create the Database in the Amazon RDS console
* Open the Amazon EC2 console
* Launch an EC2 Instance
* Convert your private key to PuTTY using PuTTYgen
* Connect to your Linux Instance
* Install an Apache web server with PHP
* Set file permissions for the Apache web server
* Connect your Apache web server to your Database Instance

**Step 1:** Open the AWS console

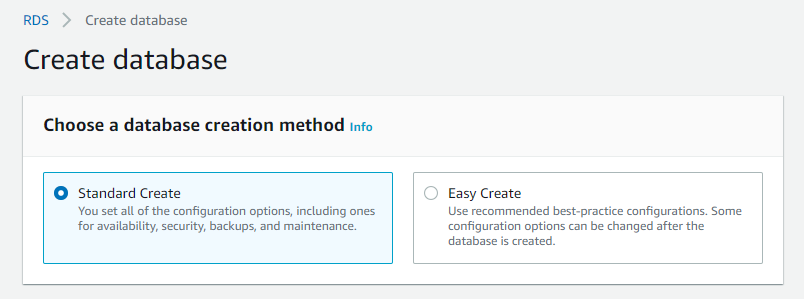


**Step 2:** Create the Database in the Amazon RDS console

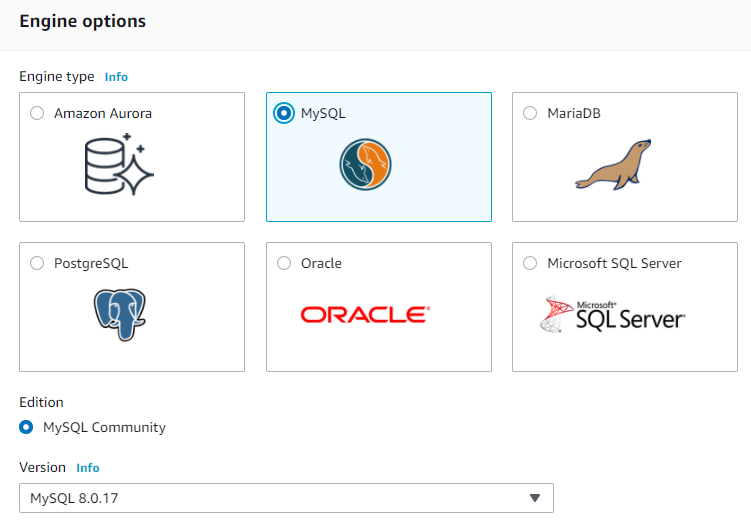
* In the console navigation pane, choose **Databases**



* Click on **Create Database**
* In the **Choose a database creation method** section, select **Standard Create**



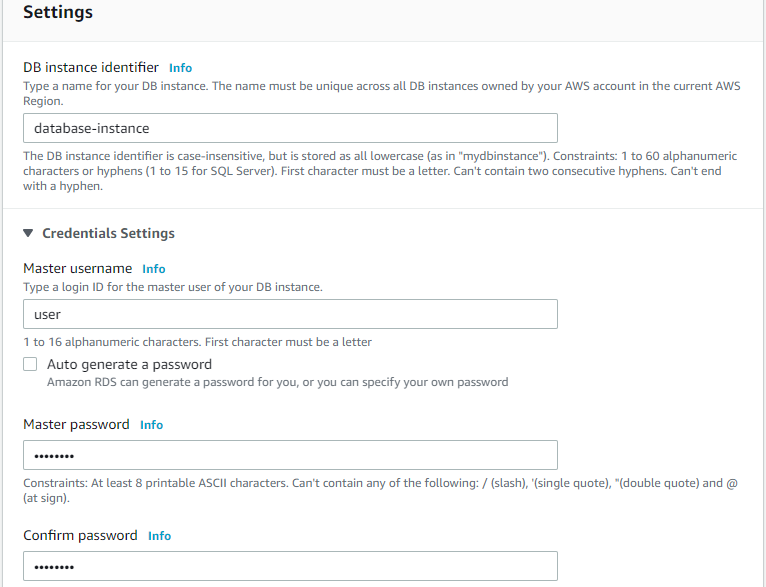
* In the **Engine options** section, select **MySQL**

****

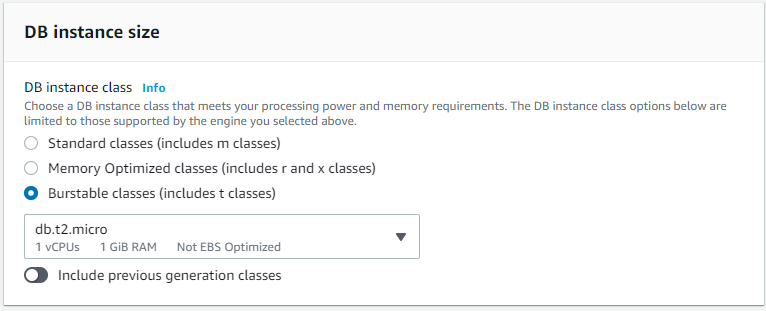
* In the **Templates** section, choose **Dev/Test**

****

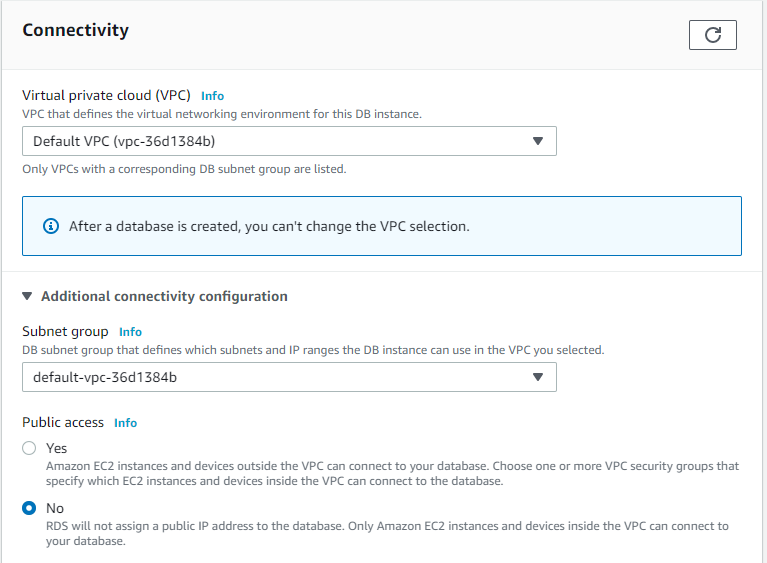
* In the **Settings** section, specify the following values:
  + DB instance identifier: database-instance
  + Master username: user
  + Auto generate a password: Disable the option
  + Master password: Choose a password
  + Confirm password: Retype the password

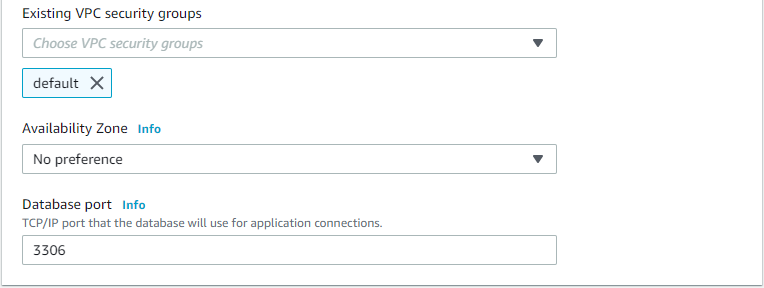


* In the **DB instance size** section, specify the following values:
  + DB instance class: Burstable classes (includes t classes)
  + db.t2.micro

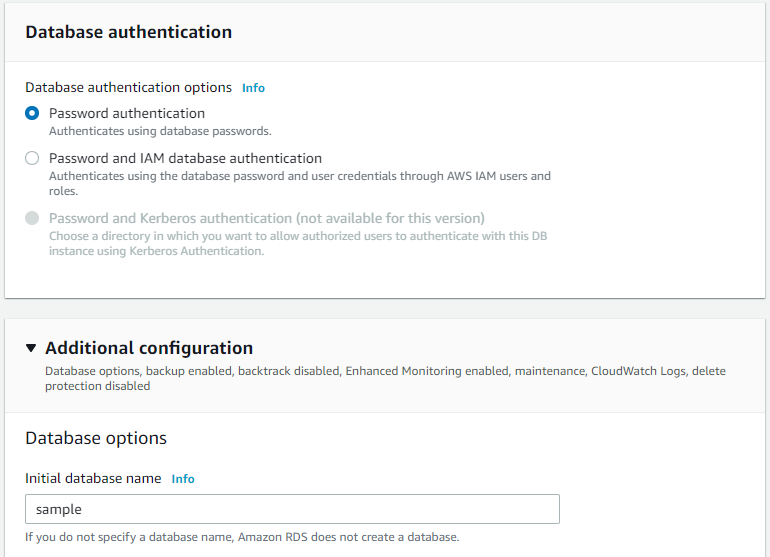


* In the **Storage** and **Availability & durability** sections, use the default values
* In the **Connectivity** section, open **Additional connectivity configuration** and specify the following values:
  + Virtual Private Cloud (VPC): Default VPC
  + Subnet group: Default subnet group
  + Publicly access: Choose No
  + VPC security groups: Choose an existing VPC security group
  + Availability Zone: No Preference
  + Database port: 3306

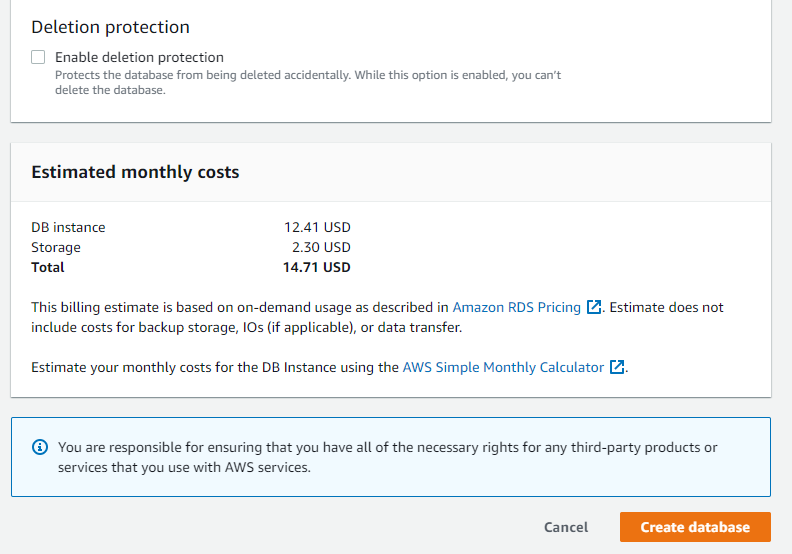




* In the **Database authentication** section, open **Additional configuration,** enter the name as **sample** for **Initial database name**, and keep the default settings for the other options



* Click on **Create database** to create your RDS MySQL Database Instance

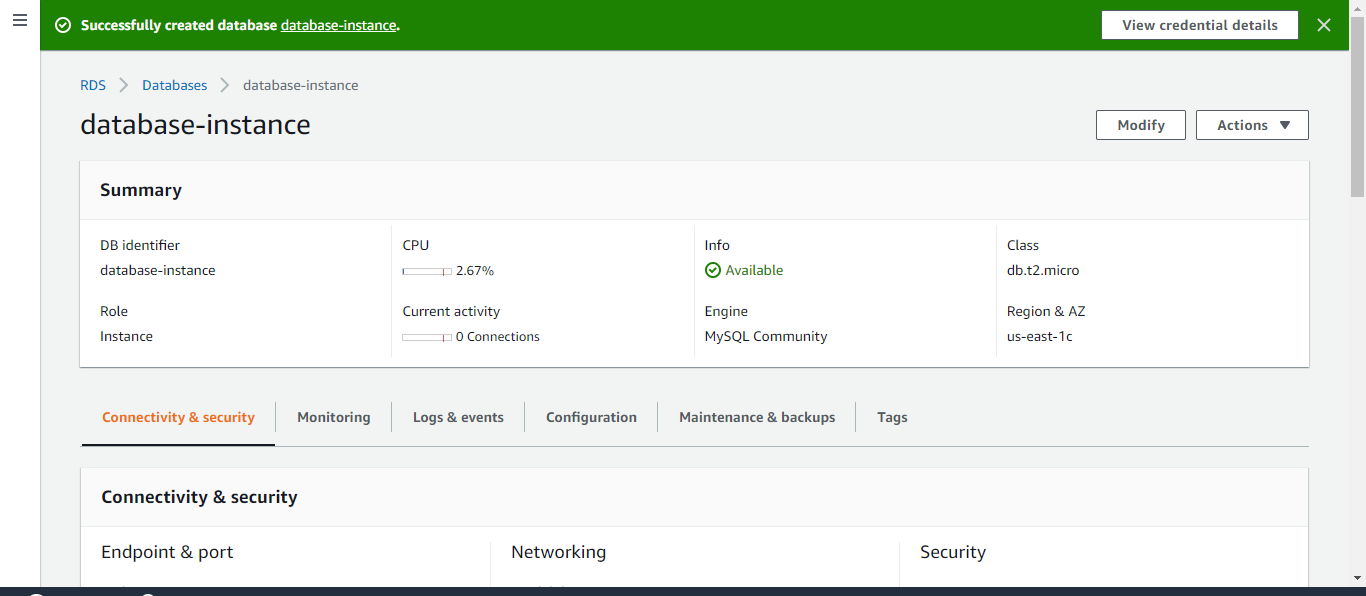


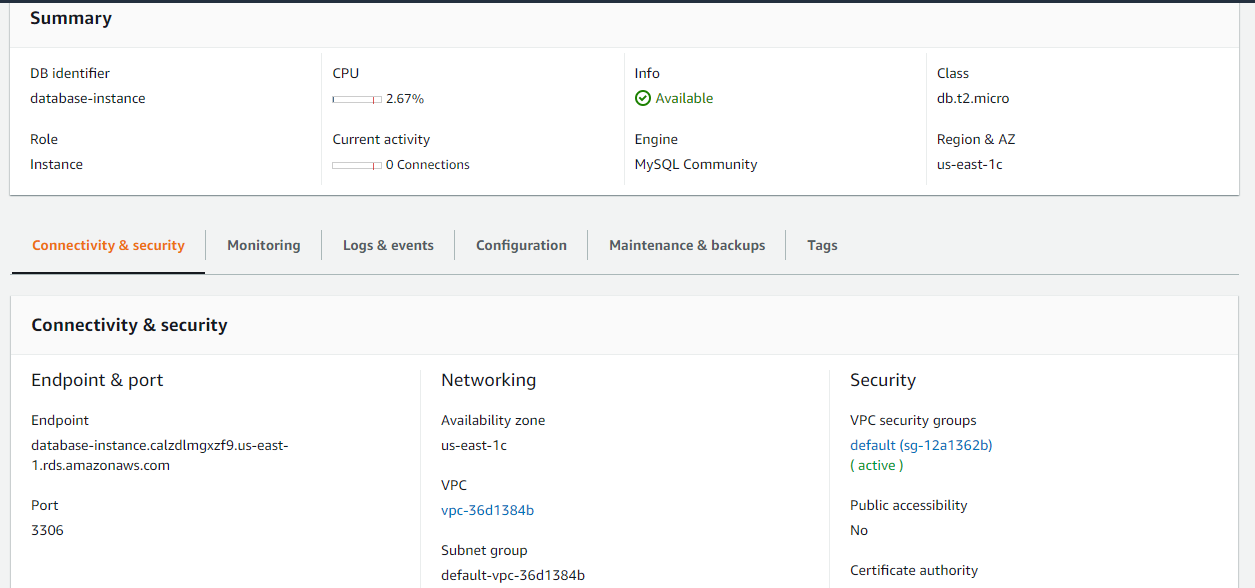
Your new Database Instance appears in the Databases list with the status as **Creating**.

Wait for the status of your new Database Instance to show as **Available**, and then choose the name of the Database Instance to view the details

* In the **Connectivity & security** section, view the **Endpoint & port** of the Database Instance

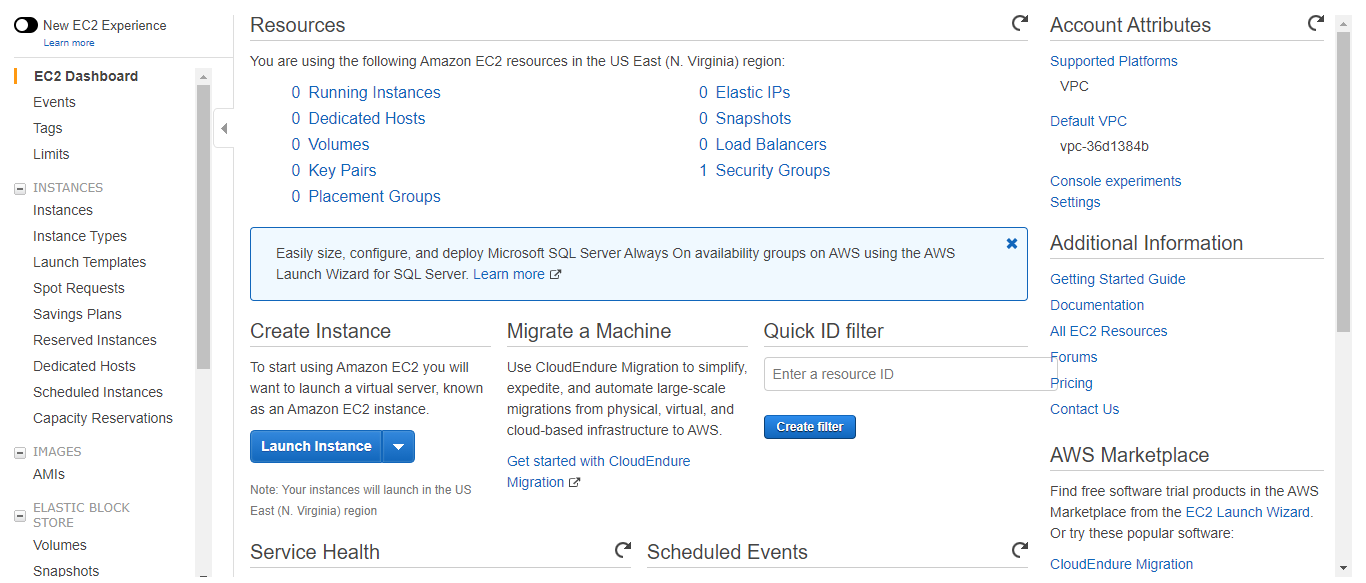
Note the endpoint and port for your Database Instance and use this information to connect the web server to your Database Instance





**Step 3:** Open the Amazon EC2 console

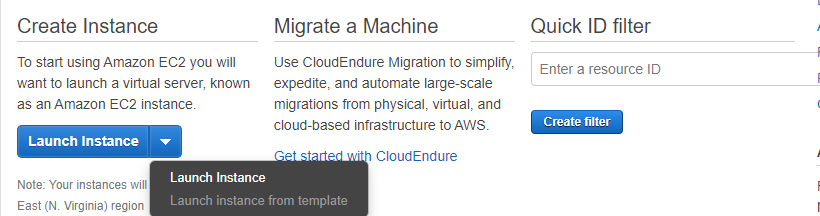
In the console navigation pane, choose **EC2 Dashboard**



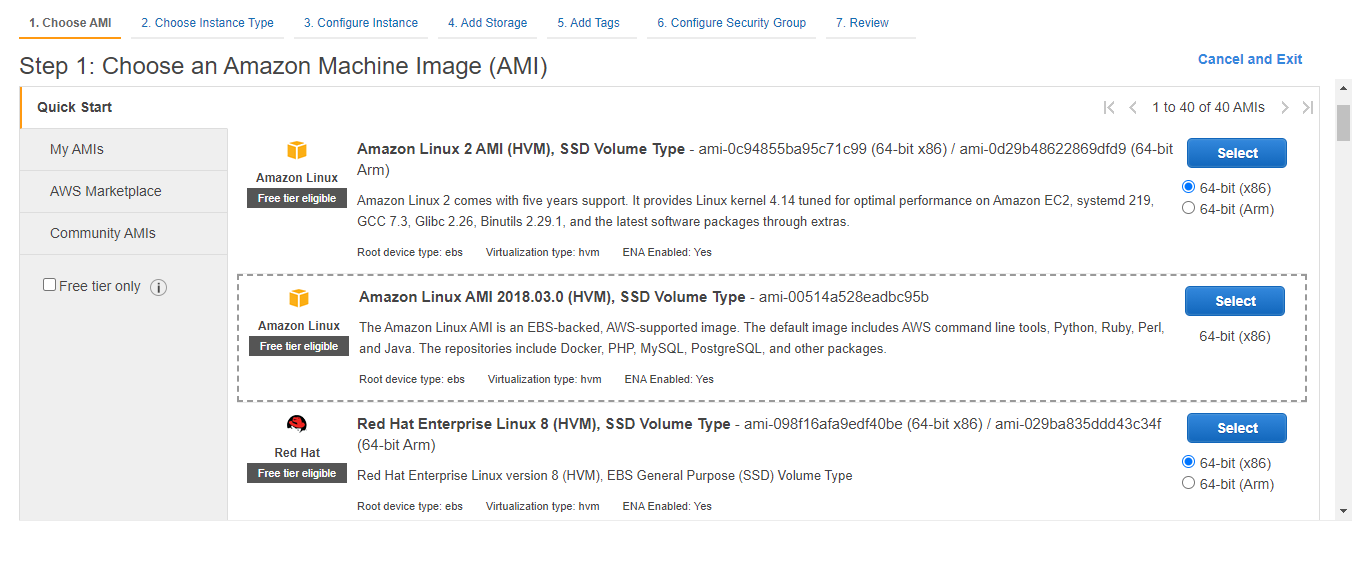
Note: In the region selector in the navigation bar, the default region is N.Virginia. All the resources required to run your instance must be created in the same region.

**Step 4:** Launch an EC2 Instance

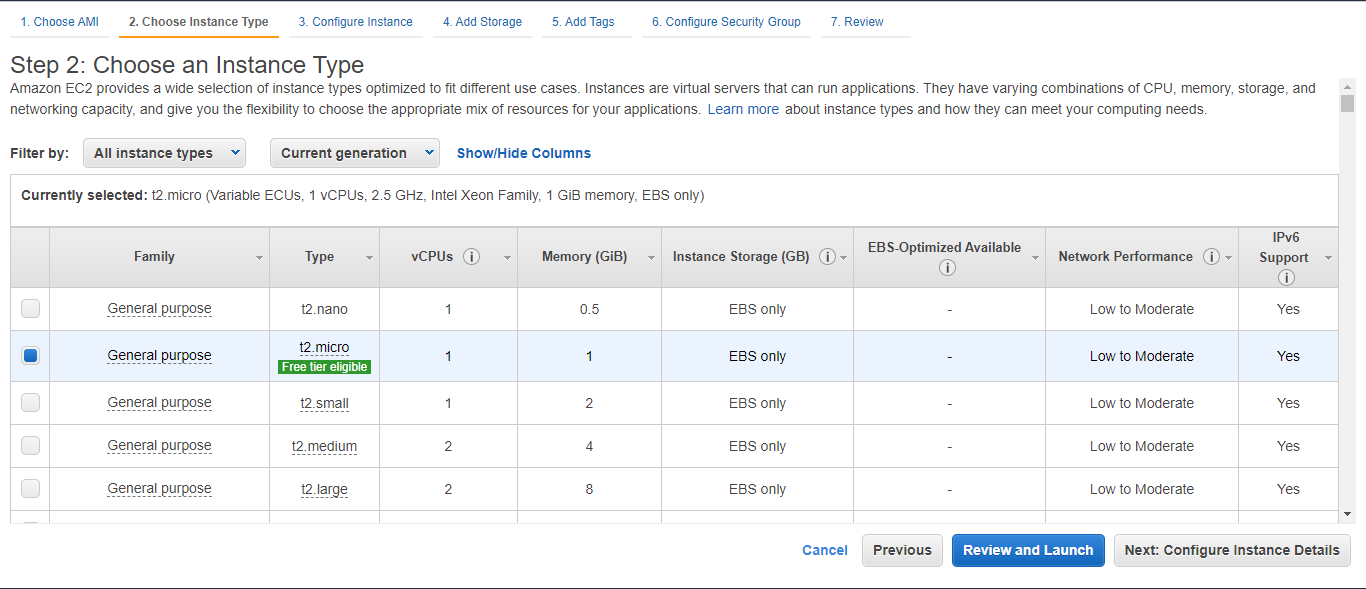
* In the Launch instance section of the console, click on **Launch instance**

****

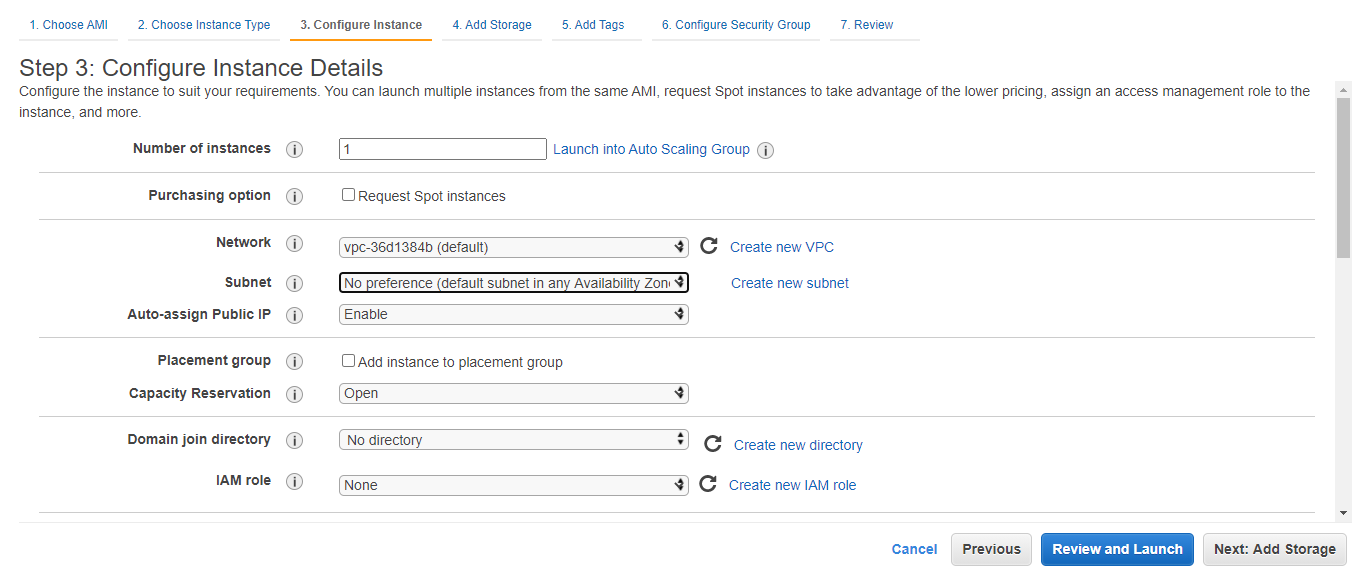
* On the **Step 1: Choose an Amazon Machine Image (AMI)** page, find the Amazon Linux AMI Free tier eligible AMI, and then click on **Select**

****

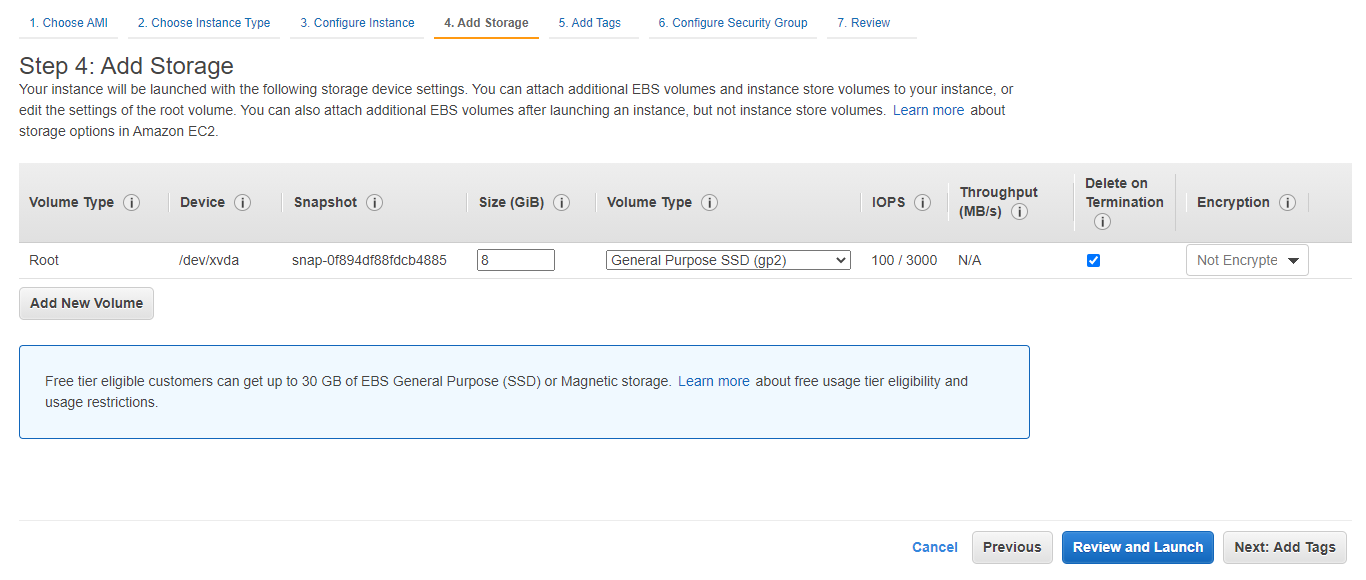
* On the **Step 2: Choose an Instance Type** page, choose General Purpose **t2.micro** Instance type, and then click on **Next: Configure Instance Details**



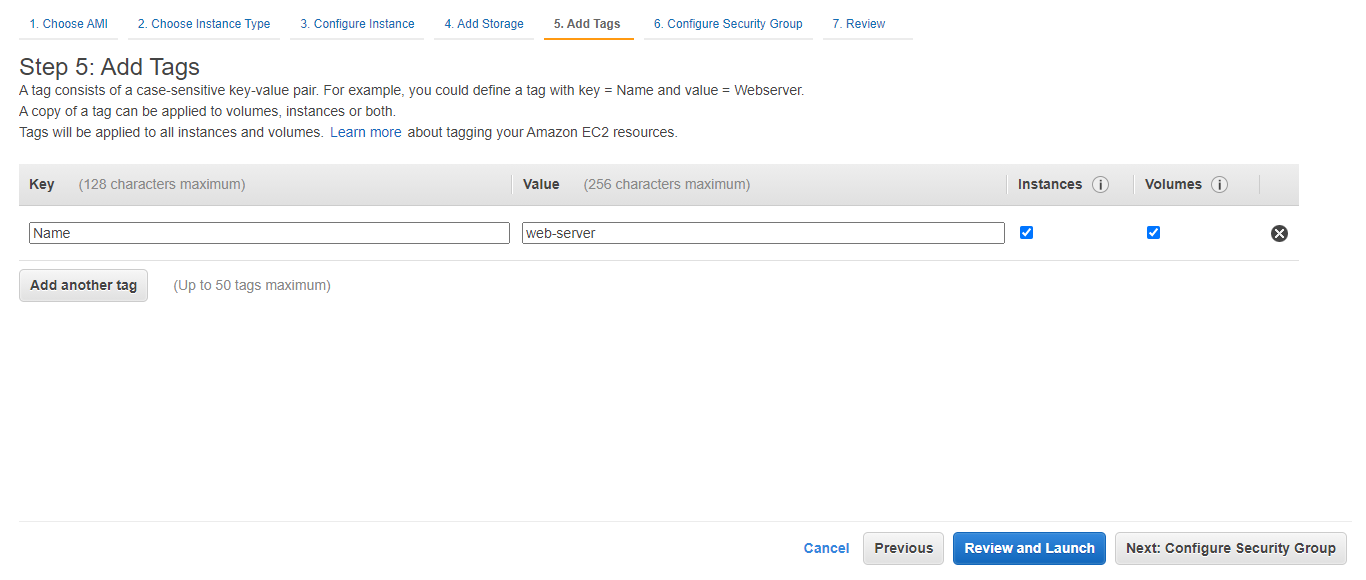
* On the **Step 3: Configure Instance Details** page, accept the following default selections or configure them, and then choose **Next: Add Storage**
  + Network: Choose the VPC with both public and private subnets that you chose for the Database Instance
  + Subnet: Select the default option
  + Auto-assign Public IP: Select Enable



* On the **Step 4: Add Storage** page, keep the default values, and then choose **Next: Add Tags**

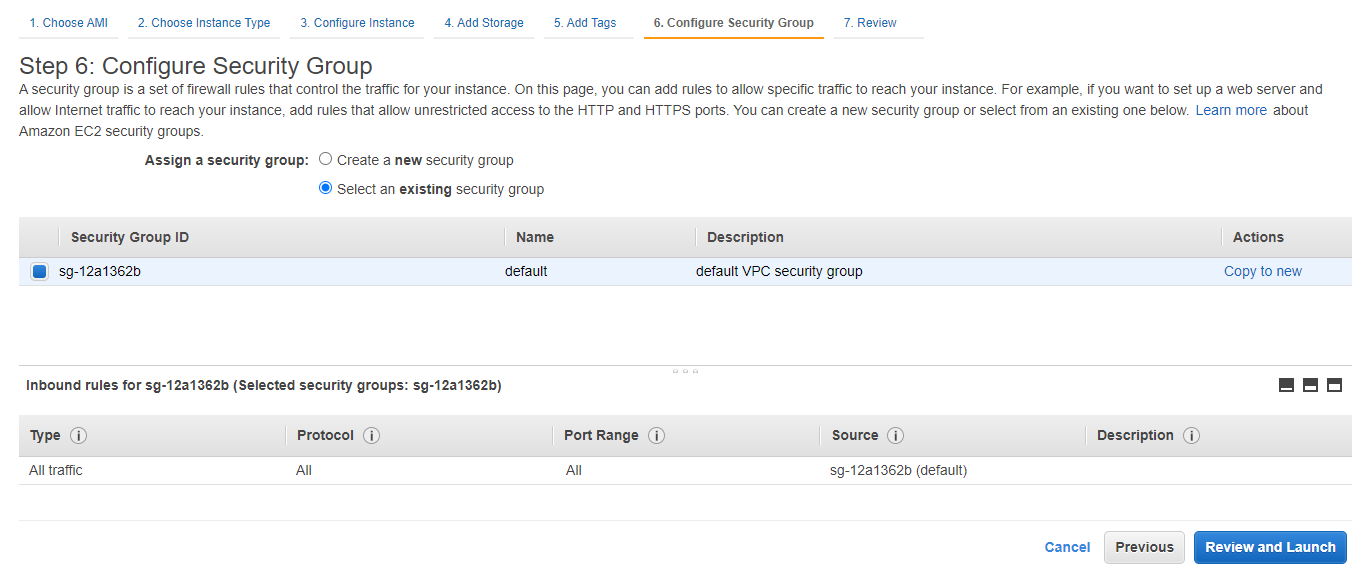


* On the **Step 5: Add Tags** page, choose **Add Tag**, enter **Name** for **Key** and enter **web-server** for **Value,** and then click on **Next: Configure Security Group**

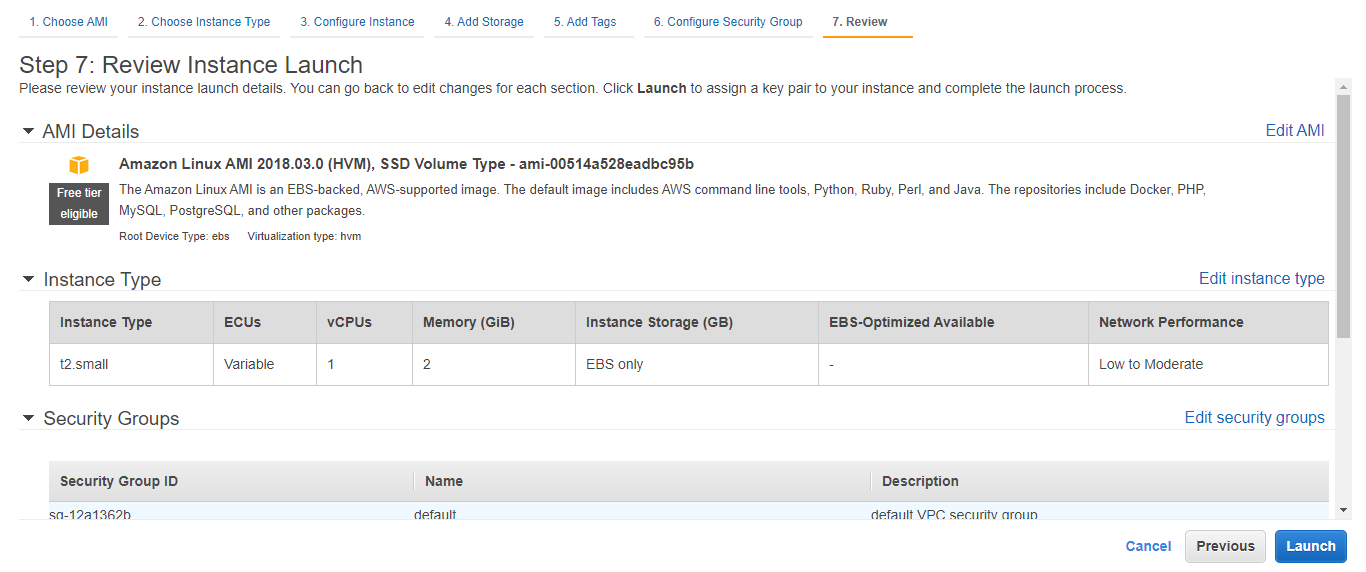


* On the **Step 6:** **Configure Security Group** page, choose **Select an existing security group**, and then click on **Review and Launch**

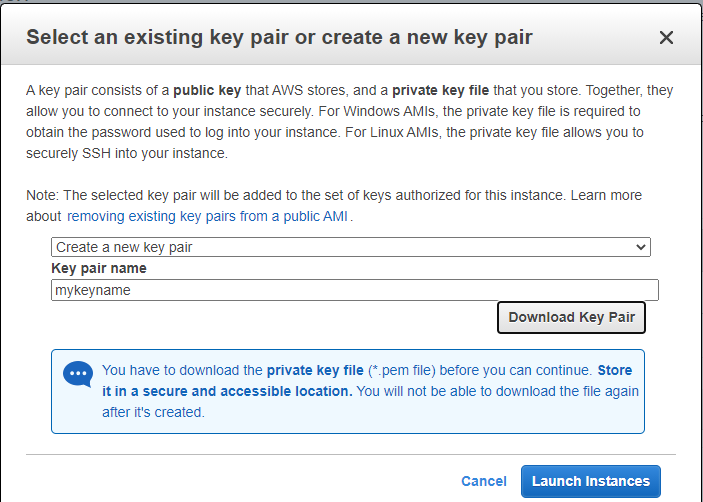
Note: Make sure that the security group that you choose includes the inbound rules for Secure Shell (SSH) and HTTP access.

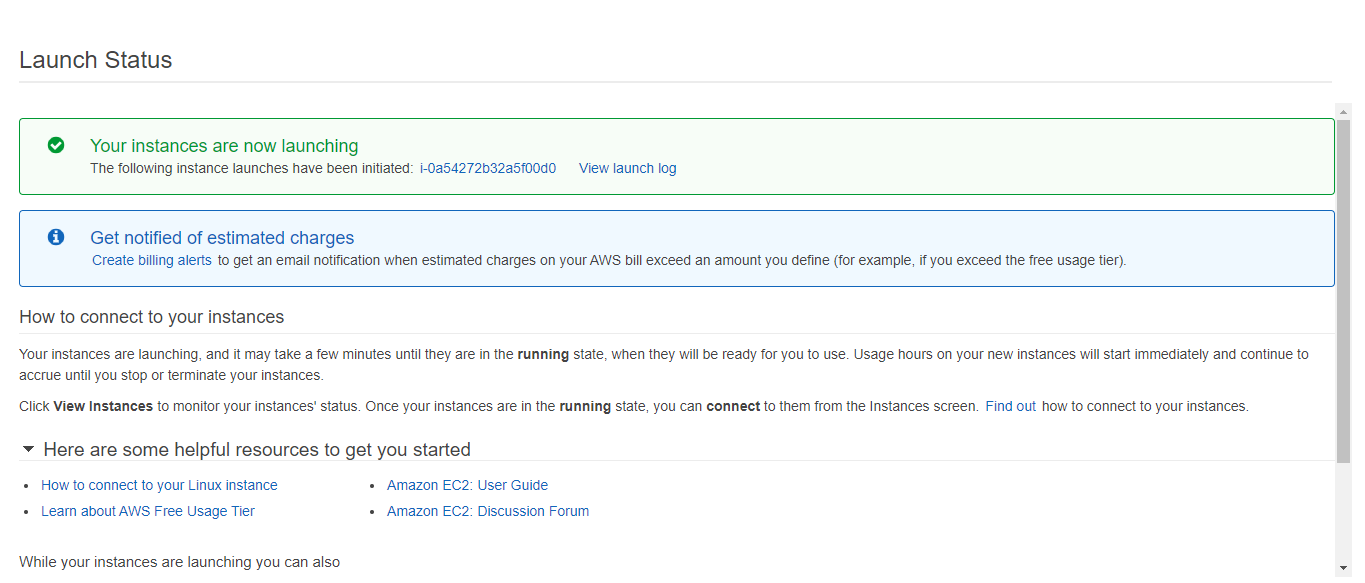


* On the **Step 7: Review Instance Launch** page, review your instance launch details, and then click on **Launch**

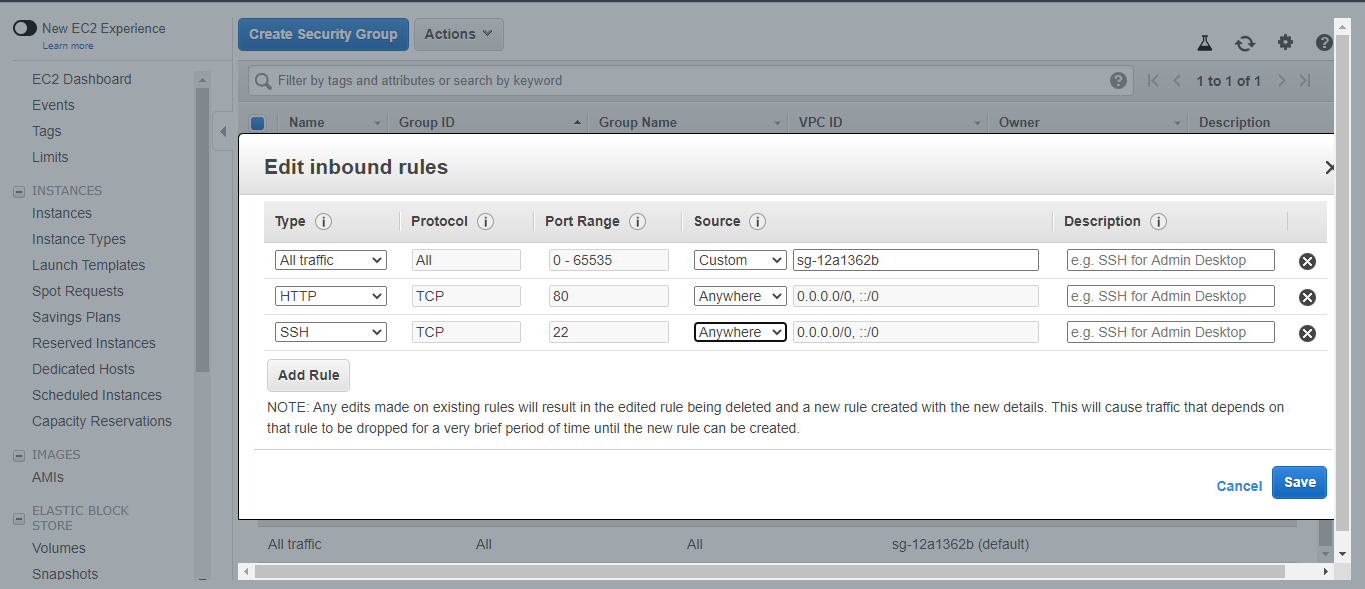
****

* Select an existing key pair or create a new key pair dialog box, create a new key pair from the drop-down list that will be used to access the instance. Give the name of the key pair, download it in your system, and then click on **Launch Instances**

****

Your instances are now launching.****

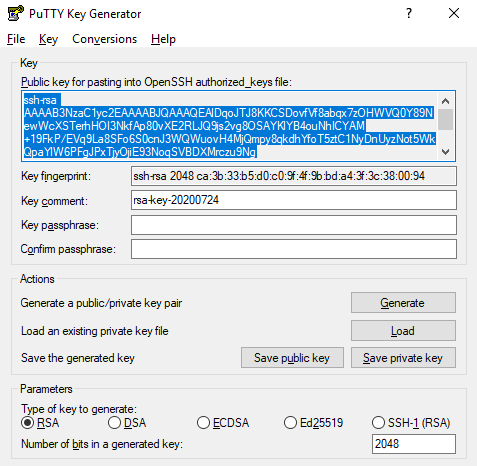
* On the navigation pane, choose **Security Group,** select the security group that you chose for the EC2 Instance, edit the inbound rules by adding **HTTP** and **SSH** type, and then click on **Save**

****

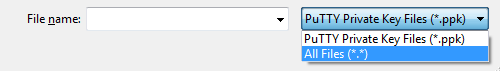
* On the **Launch Status** page, in the **Your instances are now launching** section, click on the **View instance** button to view the status of the instances. When the **Status Checks** are completed successfully and status of the Instance State is running, then connect to your instance

**Step 5:** Convert your private key to PuTTY using PuTTYgen

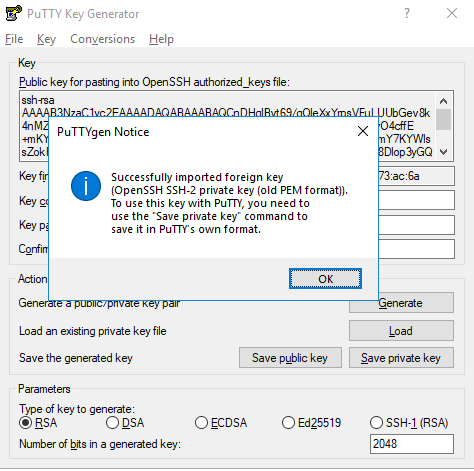
* From the Start menu, choose **All Programs**, and then **PuTTYgen**
* Under Type of key to generate, choose **RSA**



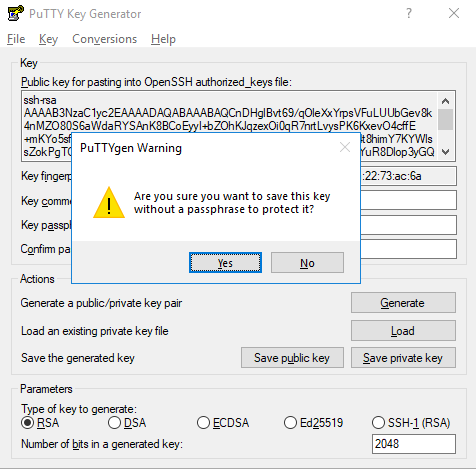
* Select the **Load** button. By default, PuTTYgen displays only the files with the extension .ppk. To locate your .pem file, choose the option to display all types of files



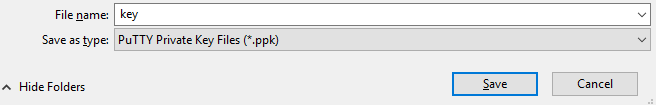
* Select your .pem file for the key pair that you specified while launching your instance, and then click on **Open**. PuTTYgen displays a notice that the .pem file was successfully imported. Finally, click on the **OK** button



* To save the key in the format that PuTTY can use, click on **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Click on the **Yes** button



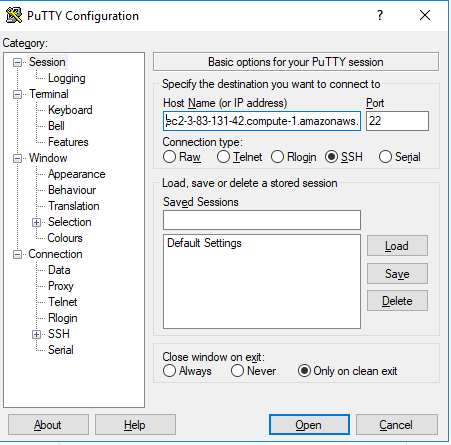
* Specify the same name for the key that you used for the key pair (for example, key) and click on the **Save** button. PuTTY automatically adds the .ppk file extension.



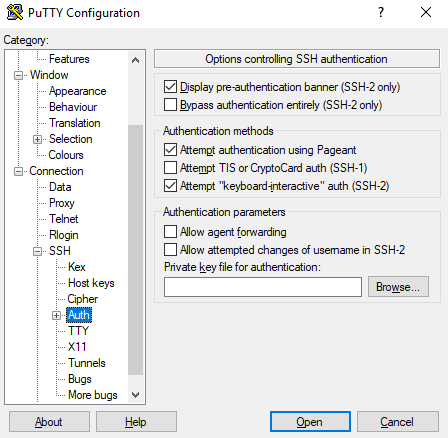
Your private key is now in the correct format for use with PuTTY.

**Step 6:** Connect to your Linux Instance

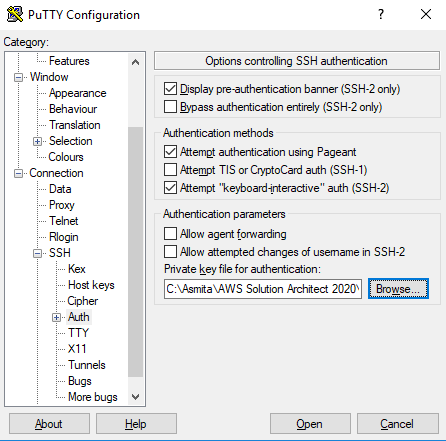
* From the Start menu, choose **All Programs** and **PuTTY**
* In the Category pane, choose **Session** and to connect using your instance's public DNS name, enter the public DNS in the host name and ensure that the Port value is 22



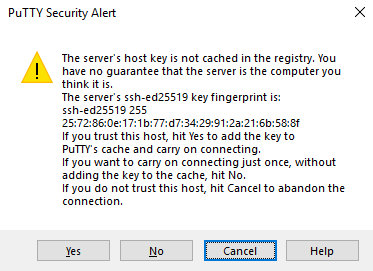
* In the Category pane, expand **Connection,** and then **SSH**. Choose **Auth**, and then click on **Browse**



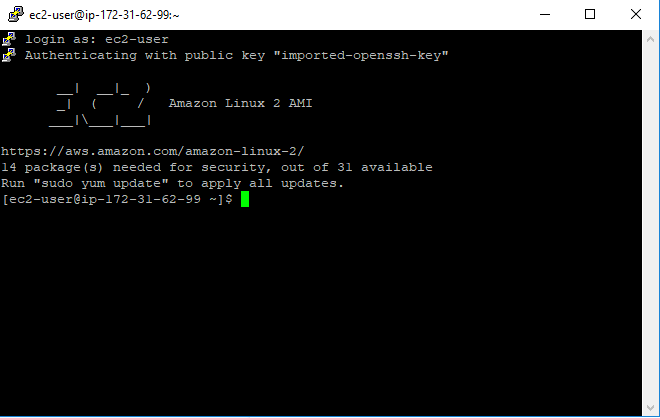
* Select the .ppk file that you generated for your key pair and click on **Open**



* If this is the first time that you are connecting to this instance, PuTTY will display a security alert dialog box that will ask whether you trust the host to which you are connecting. When you click on the **Yes** button, a window will open, and you will be connected to your instance.



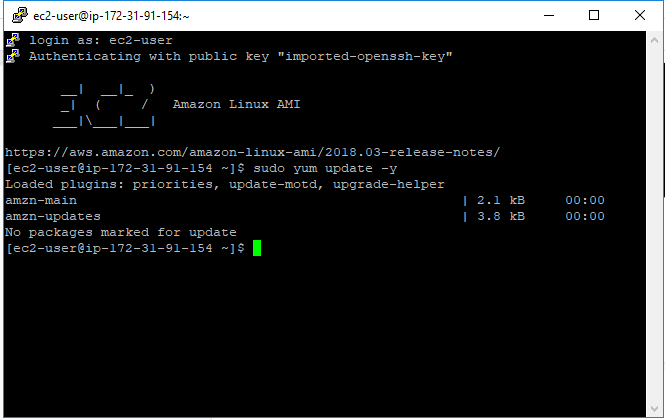
Give the command **ec2-user**. You have created a Linux-based EC2 Instance.



**Step 7:** Install an Apache web server with PHP

* Get the latest bug fixes and security updates by updating the software on your EC2 Instance. To do this, use the following command:

**sudo yum update -y**

****

* After the update is complete, install the **Apache web server** with the PHP software package using the **yum** install command. The following command installs multiple software packages and related dependencies at the same time:

**sudo yum install -y httpd24 php56 php56-mysqlnd**

You can view your version of Amazon Linux with the following command:

**cat /etc/system-release**

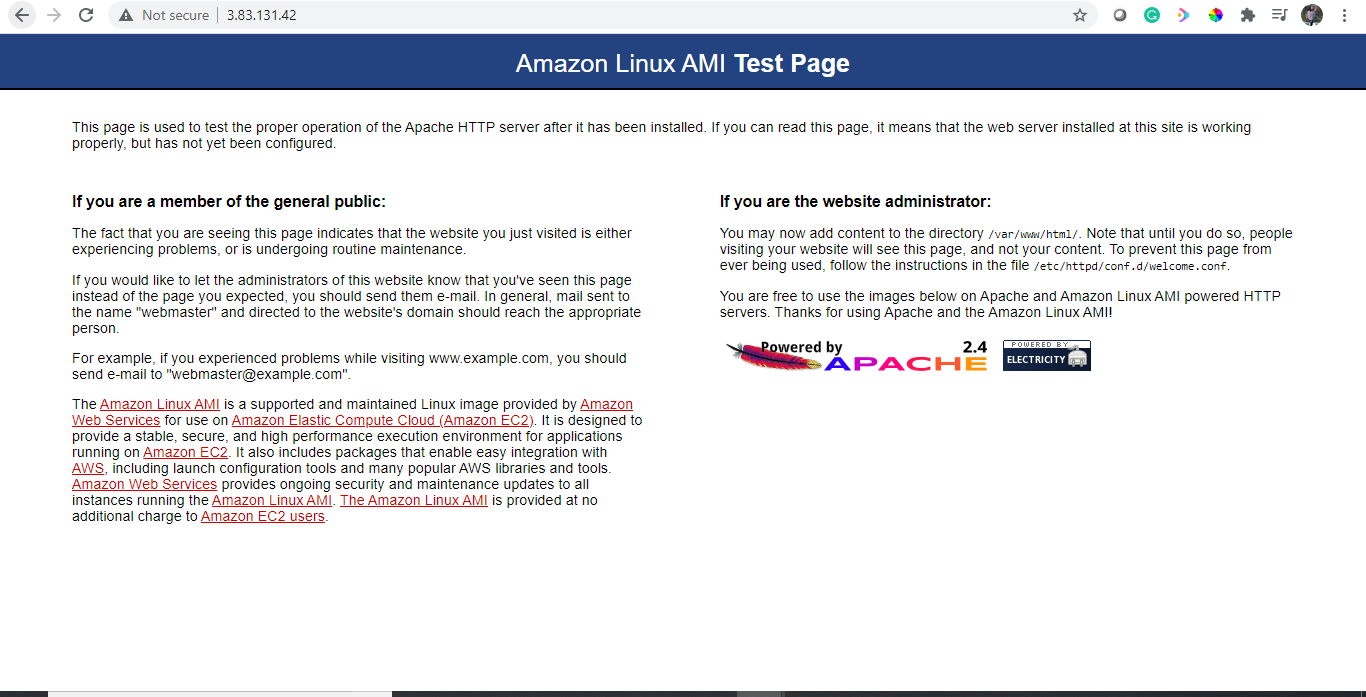


* Start the web server with the following command:

**sudo service httpd start**

****

You can test the installation and status of your web server. To do this, enter the public IP address in the address bar of the web browser



* Configure the web server to start with each system boot using the following command:

**sudo chkconfig httpd on**

**Step 8:** Set file permissions for the Apache web server

* Add the www group to EC2 Instance with the following command:

**sudo groupadd www**

* Add the ec2-user to the www group with the following command:

**sudo usermod -a -G www ec2-user**

* Log out to refresh your permissions and include the new www group with the following command:

**exit**

* Log in again and verify the existence of the www group with the following command:

**groups**

****

* Change the group ownership of the **/var/www** directory and its contents to the www group with the following command:

**sudo chgrp -R www /var/www**

* Change the directory permissions of /var/www and its subdirectories to add the group write permissions, and then set the group ID on the subdirectories that will be created in the future with the following commands:

**sudo chmod 2775 /var/www**

**find /var/www -type d -exec sudo chmod 2775 {} +**

* Recursively change the permissions for files in the /var/www directory and its subdirectories to add the group write permissions with the following command:

**find /var/www -type f -exec sudo chmod 0664 {} +**

**Step 9:** Connect your Apache web server to your Database Instance

* Add content to the Apache web server that connects to your Database Instance
* While you are still connected to your EC2 Instance, change the directory to **/var/www** and create a new subdirectory named **inc** with the following command:

**cd /var/www**

**mkdir inc**

**cd inc**

****

* Create a new file in the **inc** directory named **dbinfo.inc**, and then edit the file by the nano command (or the editor of your choice)

**>dbinfo.inc**

**nano dbinfo.inc**

* Add the following contents to the dbinfo.inc file. Here, db\_instance\_endpoint is your Database Instance endpoint, and the master password is the password that was given by you while creating the Database Instance

**<?php**

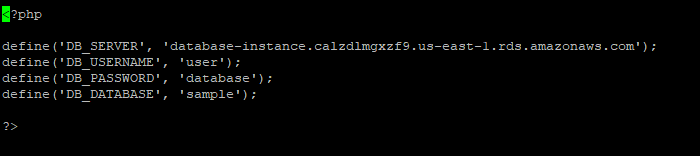
**define('DB\_SERVER', 'db\_instance\_endpoint');**

**define('DB\_USERNAME', 'user');**

**define('DB\_PASSWORD', 'master password');**

**define('DB\_DATABASE', 'sample');**

**?>**

****

* Save and close the dbinfo.inc file
* Change the directory to **/var/www/html** with the following command:

**cd /var/www/html**

* Create a new file in the html directory named **SamplePage.php**, and then edit the file by the nano command (or the editor of your choice)

**>SamplePage.php**

**nano SamplePage.php**

* Add the following contents to the SamplePage.php file:

**<?php include "../inc/dbinfo.inc"; ?>**

**<html>**

**<body>**

**<h1>Sample page</h1>**

**<?php**

**/\* Connect to MySQL and select the database. \*/**

**$connection = mysqli\_connect(DB\_SERVER, DB\_USERNAME, DB\_PASSWORD);**

**if (mysqli\_connect\_errno()) echo "Failed to connect to MySQL: " . mysqli\_connect\_error();**

**$database = mysqli\_select\_db($connection, DB\_DATABASE);**

**/\* Ensure that the EMPLOYEES table exists. \*/**

**VerifyEmployeesTable($connection, DB\_DATABASE);**

**/\* If input fields are populated, add a row to the EMPLOYEES table. \*/**

**$employee\_name = htmlentities($\_POST['NAME']);**

**$employee\_address = htmlentities($\_POST['ADDRESS']);**

**if (strlen($employee\_name) || strlen($employee\_address)) {**

**AddEmployee($connection, $employee\_name, $employee\_address);**

**}**

**?>**

**<!-- Input form -->**

**<form action="<?PHP echo $\_SERVER['SCRIPT\_NAME'] ?>" method="POST">**

**<table border="0">**

**<tr>**

**<td>NAME</td>**

**<td>ADDRESS</td>**

**</tr>**

**<tr>**

**<td>**

**<input type="text" name="NAME" maxlength="45" size="30" />**

**</td>**

**<td>**

**<input type="text" name="ADDRESS" maxlength="90" size="60" />**

**</td>**

**<td>**

**<input type="submit" value="Add Data" />**

**</td>**

**</tr>**

**</table>**

**</form>**

**<!-- Display table data. -->**

**<table border="1" cellpadding="2" cellspacing="2">**

**<tr>**

**<td>ID</td>**

**<td>NAME</td>**

**<td>ADDRESS</td>**

**</tr>**

**<?php**

**$result = mysqli\_query($connection, "SELECT \* FROM EMPLOYEES");**

**while($query\_data = mysqli\_fetch\_row($result)) {**

**echo "<tr>";**

**echo "<td>",$query\_data[0], "</td>",**

**"<td>",$query\_data[1], "</td>",**

**"<td>",$query\_data[2], "</td>";**

**echo "</tr>";**

**}**

**?>**

**</table>**

**<!-- Clean up. -->**

**<?php**

**mysqli\_free\_result($result);**

**mysqli\_close($connection);**

**?>**

**</body>**

**</html>**

**<?php**

**/\* Add an employee to the table. \*/**

**function AddEmployee($connection, $name, $address) {**

**$n = mysqli\_real\_escape\_string($connection, $name);**

**$a = mysqli\_real\_escape\_string($connection, $address);**

**$query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";**

**if(!mysqli\_query($connection, $query)) echo("<p>Error adding employee data.</p>");**

**}**

**/\* Check whether the table exists and, if not, create it. \*/**

**function VerifyEmployeesTable($connection, $dbName) {**

**if(!TableExists("EMPLOYEES", $connection, $dbName))**

**{**

**$query = "CREATE TABLE EMPLOYEES (**

**ID int(11) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,**

**NAME VARCHAR(45),**

**ADDRESS VARCHAR(90)**

**)";**

**if(!mysqli\_query($connection, $query)) echo("<p>Error creating table.</p>");**

**}**

**}**

**/\* Check for the existence of a table. \*/**

**function TableExists($tableName, $connection, $dbName) {**

**$t = mysqli\_real\_escape\_string($connection, $tableName);**

**$d = mysqli\_real\_escape\_string($connection, $dbName);**

**$checktable = mysqli\_query($connection,**

**"SELECT TABLE\_NAME FROM information\_schema.TABLES WHERE TABLE\_NAME = '$t' AND TABLE\_SCHEMA = '$d'");**

**if(mysqli\_num\_rows($checktable) > 0) return true;**

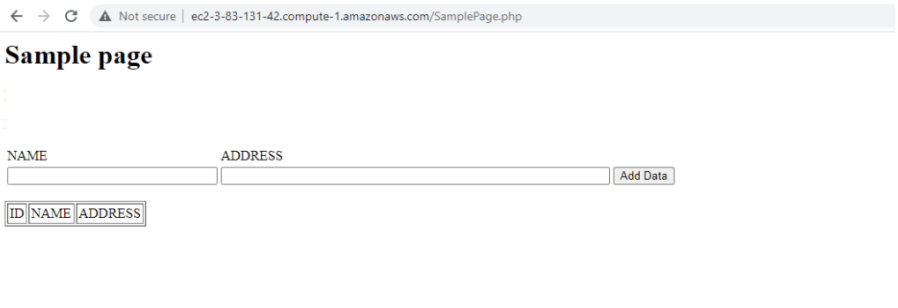
**return false;**

**}**

**?>**

* Save and close the SamplePage.php file
* Verify that your web server successfully connects to your Database Instance by opening a web browser and browsing to the following URL:

[**http://ec2-3-83-131-42.compute-1.amazonaws.com/SamplePage.php**](http://ec2-3-83-131-42.compute-1.amazonaws.com/SamplePage.php)

****