

# Creating Amazon EC2 Instances (for Linux)

## LAB Overview

This lab leads you through the steps to launch and configure your first virtual machine in the Amazon cloud. You will learn about using Amazon Machine Images to launch Amazon EC2 instances, creating key pairs for SSH authentication, securing network access to Amazon EC2 instances with security groups, automatically configuring Amazon EC2 instances with bootstrapping scripts, and attaching Elastic IPs to Amazon EC2 instances to provide static Internet addresses. At the end of this lab you will have deployed a simple web server which includes an informational page to display details of your virtual web server instance.

---

## Download PuTTY (MS Windows)

If you do not already have the PuTTY client installed on your machine, you can download and then launch it from here:

[www.chiark.greenend.org.uk/~sgtatham/putty/](http://www.chiark.greenend.org.uk/~sgtatham/putty/)

---

## Task 1: Launch a Linux instance

In this lab, you will launch a default Amazon Linux instance with an Apache PHP web server installed on initialization.

1. On the **Services** menu, click **EC2**.
2. Click **Launch Instance**.
3. Because you require a Linux instance, in the row for the basic 64-bit **Amazon Linux AMI 2018.03.0**, which will normally be the second option on the list, click **Select**. When you create an instance, AWS will ask you which instance family you want to use. The family you choose determines how much throughput and processing cycles are available to your instance.
4. On the **Choose an Instance Type** page, select the **t2.micro** instance type, which is the lowest-cost option, should be automatically selected.
5. Click **Next: Configure Instance Details**.
6. On the **Configure Instance Details** page provide the following information:
  - **Number of instance:** 1
  - **Network:** StudentX\_MyVPC
  - **Subnet:** StudentX\_Public1
  - **Auto-assign Public IP:** Enable

7. Scroll down and expand the **Advanced Details** section.
8. For **User data**, select **As text**.

Since you will be using your Amazon EC2 instance as a web server, you need to ensure that the Apache HTTPd server is up and running, and that the PHP programming language is installed. We can accomplish this with a simple Linux shell script. The script below installs HTTPd and PHP using the yum package manager, and then starts the HTTPd server.

9. Copy the initialization script seen below. Paste the script into the **User data** box:

```
#!/bin/sh
yum -y install httpd php php-mysql
chkconfig httpd on
/etc/init.d/httpd start
```

This will automatically install and start the Apache Web server when the instance is created and launches.

**Note:** You can copy script from **bootstrap.sh**. If you type the script, use SHIFT+ENTER for new lines in the text box.

10. Click **Next: Add Storage**.
11. Click **Next: Add Tags** to accept the default storage device configuration.
12. On the **Add Tags** page, click **Add Tag**, type a **Name** for a Key box and `studentX_01` in the Value box. This name, more correctly known as a tag, will appear in the console when the instance launches. It makes it easy to keep track of running machines in a complex environment.
13. Click **Next: Configure Security Group**.
14. For **Assign a security group**, select a WebServers Security Group you created in LAB2-VPC.
15. Click **Review and Launch**.

**Note:** You may see a warning on this screen that **Your security group is open to the world**. This is a result of not restricting SSH access to your machine, as described above. For the purposes of this lab only, you may ignore this warning.

16. Review your choices, and then click **Launch**.
17. In the key pair dialog box, select **Create a new key pair**.
18. Download a key.
19. Click **Launch Instances**.
20. On the status page, which notifies you that your instances are launching, click **View Instances**.

The Instances page of the Amazon EC2 Dashboard displays the list of all running Amazon EC2 instances in the currently selected region. You can see the status of your instance here. If the status is not Running, wait a few minutes and refresh the list.

21. Select your instance to display a list of details and status update in the lower pane.
22. Copy the **Public DNS (IPv4)** value to your Clipboard. It will look something like ec2-54-84-236-205.compute-1.amazonaws.com.

---

## Task 2: Connect to your EC2 instance

### Instructions for Windows Users: Connecting to your Amazon EC2 Instance via SSH

**Note** This section is for Windows users only. If you are running Mac OS or Linux, skip to the next section.

In this section, you will use the PuTTY Secure Shell (SSH) client and your server's public DNS address to connect to your server. The downloaded private key in step 17 is in **pem** format. To use it with putty you need to convert it.

23. Start PuTTYgen (for example, from the **Start** menu, choose **All Programs > PuTTY > PuTTYgen**).
24. Under **Type of key to generate**, choose **RSA**. (**SSH-2 RSA** in older putty versions).
25. Choose **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of all types.
26. Select your .pem file for the key pair that you specified when you launched your instance, and then choose **Open**. Choose **OK** to dismiss the confirmation dialog box.
27. Choose **Save private key** to save the key in the format that PuTTY can use. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.
28. Specify the same name for the key that you used for the key pair (for example, my-key-pair). PuTTY automatically adds the .ppk file extension.
29. Start PuTTY (from the **Start** menu, choose **All Programs > PuTTY > PuTTY**).
30. In the **Host Name** box, enter **ec2-user@<public DNS from step 21>**.
31. In the **Category** list, expand **SSH**.
32. Click **Auth** (don't expand it).
33. In the **Private key file for authentication box**, browse to the your PPK file and double-click it.
34. Click **Open**.
35. Click **Yes** when prompted to allow a first connection to this remote SSH server.

### Instructions for Mac OS and Linux Users: Connecting to your Amazon EC2 Instance via SSH

**Note** This section is for Mac OS and Linux users only. If you are running Windows, skip to the next section.

36. Open your favourite Terminal application.
37. Correct permission to the key file by typing: **chmod 600** <path\_to\_your\_pem\_file>.
38. Connect to your EC2 instance by typing: **ssh -i** <path\_to\_your\_pem\_file> **ec2-user@**<public DNS from step 21>.

---

### Task 3: Create a PHP Web Page on Your EC2

The AMI has already been customized with the installation of Apache and PHP from the script you entered as user data when the instance was launched. Modify the web server by adding an index.php file.

39. Type the following into PuTTY or Terminal window in order to create an index.php file at the root of your HTTP web server HTML document directory:

```
cd /var/www/html
sudo nano index.php
```

40. Copy and paste the following code into the console (file index.php in lab folder):

```
<?php
$url = "http://169.254.169.254/latest/meta-data/instance-id";
$instance_id = file_get_contents($url);
echo "<h1>Instance ID: " . $instance_id . "</h1><br/>";
$url = "http://169.254.169.254/latest/meta-data/placement/availability-zone";
$zone = file_get_contents($url);
echo "<h2>Zone: " . $zone . "</h2><br/>";
$url = "http://169.254.169.254/latest/meta-data/local-ipv4";
$private_ip = file_get_contents($url);
echo "<h3>Private ip: " . $private_ip . "</h3><br/>";
?>
```

41. Press CTRL+O, ENTER to save your document as index.php.
42. Press CTRL+X to exit the Nano editor.
43. Open a new browser window, and then paste the public DNS value into the address bar.

Your instance ID and Availability Zone should be displayed in the browser:



---

## END Your Lab

This is the end of the lab. Don't terminate the instance, it will be used in the next labs.