

## Creating an Amazon EC2 Linux Instance

---

edureka!

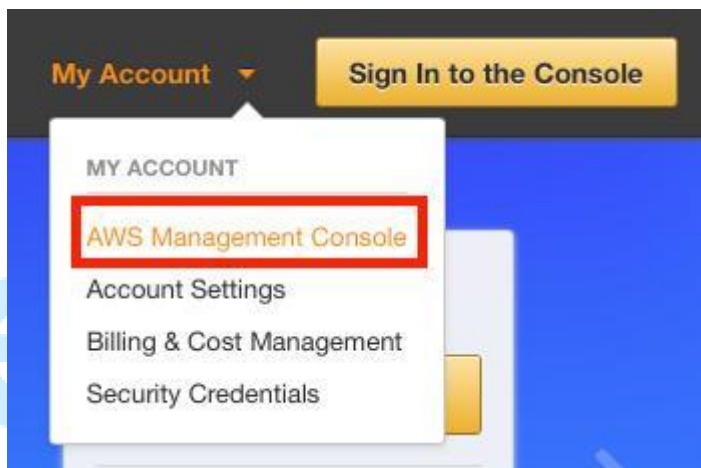
**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

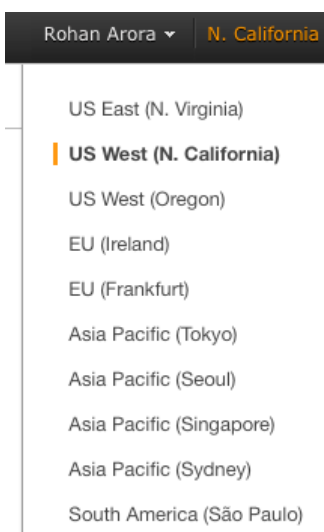
## Overview

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 EC2 instances with security groups and automatically configuring EC2 instances with bootstrapping scripts. At the end of this lab you will have deployed a simple web server which includes an informational page to display results of your virtual web server instance.

1. Login to AWS Management Console.



2. Select your preferred Region.



3. Click **EC2** under Compute section. This will take you to EC2 dashboard.

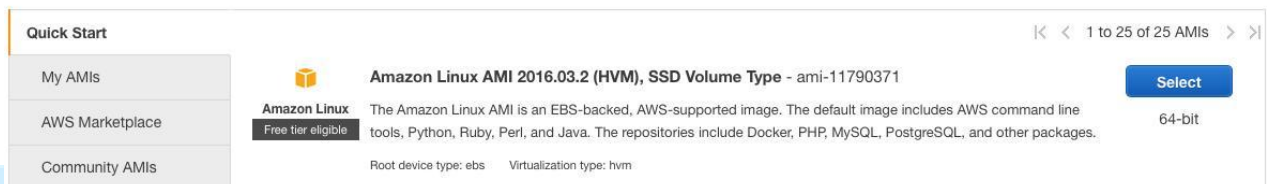


4. Click [Launch Instance](#).
5. Because you require a Linux instance, in the row for the basic 64-bit [Amazon Linux AMI](#), which will normally be the first option on the list, click [Select](#).

#### Step 1: Choose an Amazon Machine Image (AMI)

[Cancel and Exit](#)

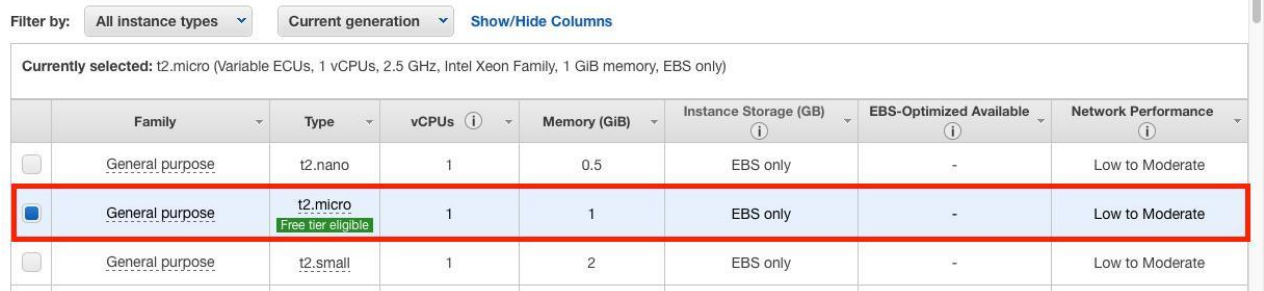
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.



6. On the [Choose an Instance Type](#) page, choose **t2.micro**, which is free tier eligible.

#### Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.



7. Click [Next: Configure Instance Details](#).
8. On the [Configure Instance Details](#) page, scroll down and expand [Advanced Details](#) section.
9. For [User Data](#), select [As Text](#).
10. Copy and paste following script into the [User Data](#) box.

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

### Step 3: Configure Instance Details

**IAM role** ⓘ None [Create new IAM role](#)

---

**Shutdown behavior** ⓘ Stop

**Enable termination protection** ⓘ ☐ Protect against accidental termination

**Monitoring** ⓘ ☐ Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

**Tenancy** ⓘ Shared - Run a shared hardware instance   
[Additional charges will apply for dedicated tenancy.](#)

▼ **Advanced Details**

**User data** ⓘ ☒ As text ☐ As file ☐ Input is already base64 encoded

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

#### 11. Click [Next: Add Storage](#).

→ This page displays which EBS volumes are attached to your image. When you launch an EC2 instance, the root volume contains the image used to boot the instance. Instances that use EBS for root device automatically have an EBS volume attached. When an EBS-backed instance is launched, an EBS volume is created for each EBS snapshot referenced by the AMI. You must have at least one snapshot that denotes the root device; the others are optional and denote additional volumes to be created from other snapshots.

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encrypted ⓘ
Root	/dev/xvda	snap-ea4eaa1b	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
<a href="#">Add New Volume</a>								

#### 12. Click Next: [Tag Instance](#) to accept the default storage device configuration.

13. On the [Tag Instance](#) page, type a name for your instance in the [Value](#) box. This name, more correctly known as tag, will appear in the console when the instance launches. It makes it easy to keep track of running machines in a complex environment. Use a name that you can easily recognize and remember.

**Step 5: Add Tags**

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	Edureka_test
<a href="#">Add another tag</a> (Up to 50 tags maximum)	

14. Click [Next: Configure Security Group](#).

→ Now you will create security group. A security group acts as a firewall that controls the traffic allowed into a group of instances. When you launch an EC2 instance, you can assign it to one or more security groups. For each security group, you add rules that govern the allowed inbound traffic to instances in the group. All other inbound traffic is discarded. You can modify rules for a security group at any time. The new rules are automatically enforced for all existing and future instances in the group.

15. For [Assign a security group](#), click [Create a new Security group](#).

16. In the [Security group name](#) box, type a name that you would like to assign to this security group.

17. (Optional) type a description for your security group.

→ By default, AWS creates a rule that allows Secure Shell (SSH) access from any IP address. It is highly recommended that you restrict terminal access to the ranges of IP addresses (e.g., IPs assigned to machines within your company) that have a legitimate business need to administer to your EC2 instance.

18. Click [Add Rule](#) to open a new port.

19. In the [Type](#) drop-down list, click [HTTP](#).

→ This will add a default handler for HTTP that will allow requests from anywhere on the internet. Since you want this web server to be accessible to the general public, you can leave this rule as is without any further configuration.

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a **new** security group  
☐ Select an **existing** security group

Security group name:

Description:

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 182.69.53.0/24
HTTP	TCP	80	Anywhere 0.0.0.0/0

Add Rule

20. Click [Review and Launch](#).

21. Review your choices, and then click [Launch](#).

**Step 7: Review Instance Launch**

▼ AMI Details [Edit AMI](#)

**Amazon Linux AMI 2016.03.2 (HVM), SSD Volume Type - ami-11790371**  
 The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.  
 Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

Security Group ID	Name	Description
-------------------	------	-------------

Cancel Previous **Launch**

22. Choose an existing key pair and select the acknowledgement check box.

**Select an existing key pair or create a new key pair** ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

Choose an existing key pair

**Select a key pair**

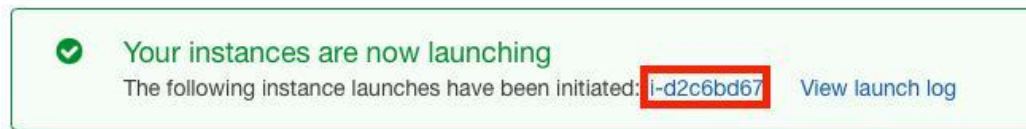
☒ I acknowledge that I have access to the selected private key file (linuxec2-kp.pem), and that without this file, I won't be able to log into my instance.

Cancel **Launch Instances**

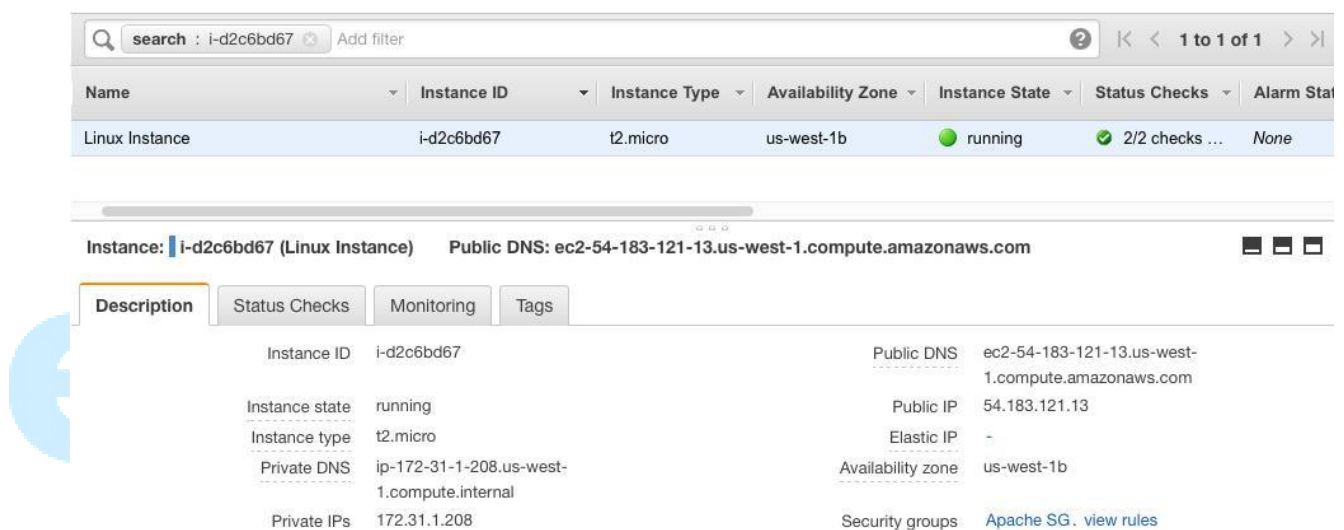
23. Click [Launch Instances](#).

24. On the status page, which notifies you that your instances have been initiated, click on instance ID.

### Launch Status



25. Select your instance to display a list of details and status update in the lower pane.



### Instructions for Windows Users: Connecting to EC2 instance via SSH

- In this section, you will use the PuTTY Secure Shell (SSH) client and your server's public DNS address to connect to your server.
- All EC2 instances are assigned two IP addresses at launch: a private IP address (RFC 1918) and a public IP address that are directly mapped to each other through Network Address Translation (NAT). Private IP addresses are only reachable from within Amazon EC2 network. Public IP addresses are reachable from the internet.



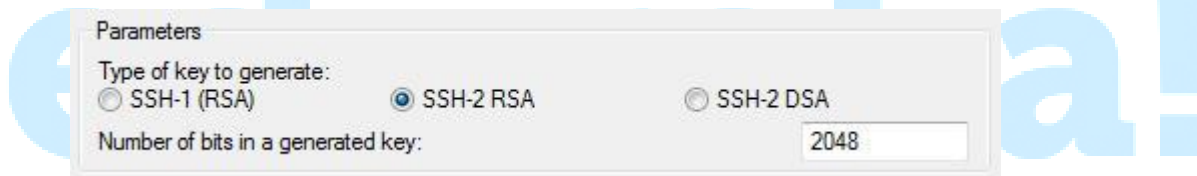
→ EC2 also provides an internal DNS name and public DNS name that map to the private and public IP addresses, respectively. The internal DNS name can only be resolved within Amazon EC2. The public DNS name resolves to the public IP address outside the EC2 network, and to the private IP address within the EC2 network.

### Converting Your Private Key Using PuTTYgen

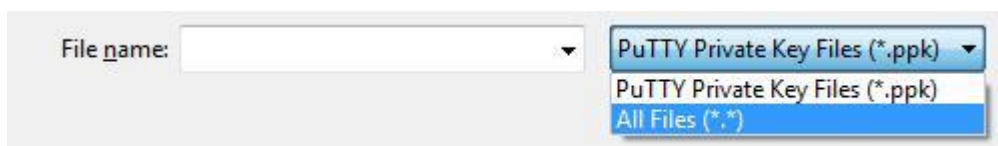
→ PuTTY does not natively support the private key format (.pem) generated by Amazon EC2. PuTTY has a tool named PuTTYgen, which can convert keys to the required PuTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PuTTY.

26. Start PuTTYgen (for example, from the [Start](#) menu, click [All Programs > PuTTY > PuTTYgen](#)).

27. Under [Type of key to generate](#), select [SSH-2 RSA](#).



28. Click [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.



29. Select your .pem file for the key pair that you specified when you launch your instance, and then click [Open](#). Click [OK](#) to dismiss the confirmation dialog box.

30. Click [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. Click [Yes](#).

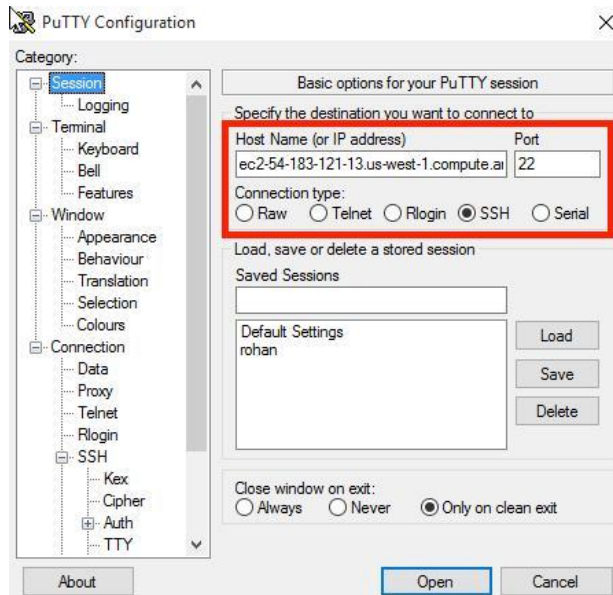
31. Specify the same name for the key that you used for the key pair (for example, my-key-pair). PuTTY automatically adds the .ppk file extension.



## Connect to EC2 instance using SSH and PuTTY

32. Open PuTTY.exe

33. In the **Host Name box**, enter either **Public DNS** or **Public IP** of your instance.

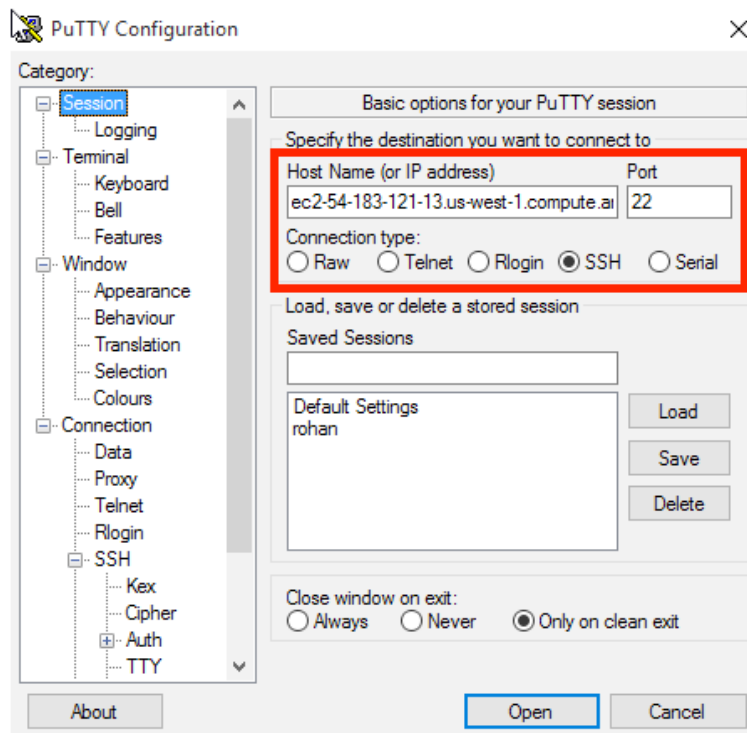


34. In the **Category** list, expand SSH.

35. Click **Auth** (don't expand it).

36. In the **Private Key file for authentication** box, browse to the PPK file that you downloaded and double-click it.

37. Click **Open**.



38. Click **Yes** when prompted to allow a first connection to this remote SSH server.

Because you are using a key pair for authentication, you will not be prompted for a password.

39. Type in **ec2-user** when prompted for login ID.

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Tue Jun 28 07:15:24 2016 from 122.162.216.136

  _ |  _ | _ )
 _ | ( _ | /   Amazon Linux AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2016.03-release-notes/
1 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-1-208 ~]$
```

## Connect to EC2 instance using OpenSSH CLI client

40. Open the Terminal application.

41. Enter the following commands.

```
chmod 400 <private key file>
ssh -i <private key file> ec2-user@<public IP address>
```

```
Rohans-MacBook-Pro:~ rohanarora$ cd Documents
Rohans-MacBook-Pro:Documents rohanarora$ chmod 400 linuxec2-kp.pem
Rohans-MacBook-Pro:Documents rohanarora$ ssh -i linuxec2-kp.pem ec2-user@54.183.121.13
The authenticity of host '54.183.121.13 (54.183.121.13)' can't be established.
ECDSA key fingerprint is SHA256:f0g3aIfYT7DQdnb5Qy5lHvjv75A8vUMsC00QhkMRZEc.
Are you sure you want to continue connecting (yes/no)? yes
```

## Create a PHP Web Page on Your Linux Web Server

→ 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.

42. Copy the following commands into PuTTY. This will create an index.php file at the root of your HTTP web server's HTML document directory.

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

43. Copy the following code and paste to Nano:

```
<?php
$url = "http://169.254.169.254/latest/meta-data/instance-id";
$instance_id = file_get_contents($url);
echo "Instance ID: <b>" . $instance_id . "</b><br/>";
$url = "http://169.254.169.254/latest/meta-data/placement/availability-zone";
$zone = file_get_contents($url);
echo "Zone: <b>" . $zone . "</b><br/>";
?>
```

```

GNU nano 2.5.3      File: index.php
?php
$url = "http://169.254.169.254/latest/meta-data/instance-id";
$instance_id = file_get_contents($url);
echo "Instance ID: <b>" . $instance_id . "</b><br/>";
$url = "http://169.254.169.254/latest/meta-data/placement/availability-zone";
$zone = file_get_contents($url);
echo "Zone: <b>" . $zone . "</b><br/>";
?>

```

```

[ Read 9 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Uncut Text ^T To Spell  ^_ Go To Line

```

44. Press CTRL+O, ENTER to save your document as index.php.
45. Press CTRL+X to exit the Nano editor.
46. Close your PuTTY or Terminal window.

### View Your Website

- In this section, you will navigate to your new website and see the content of the page that you just created.
47. Return to AWS Management Console.
  48. In your list of running EC2 instances, select the instance to display the instance details.
  49. Copy and paste either the **Public IP** or **Public DNS** name in your browser. Your instance ID and Availability Zone should be displayed in the browser.



## Conclusion

Congratulations! You have now successfully:

- Learned about basic concepts and terminology of the Amazon Elastic Compute Cloud (EC2) service.
- Created your own EC2 server instance running Linux in the AWS cloud.
- Modified it to run a web server with a page that displays machine-specific information.

edureka!