

INSTALLATION GUIDE FOR HYPERLEDGER FABRIC ON AN AWS EC2 INSTANCE

Overview

This Document covers the installation and setup of [Hyperledger Fabric](#) on an Amazon Web Services (AWS) Enterprise Compute Cloud (EC2) Instance on **Linux Ubuntu 16.04 LTS**.

This Document also covers all the necessary prerequisites to execute several examples that are provided by Hyperledger Fabric. This includes **curl**, **Docker CE**, **Docker Compose**, **Node.js**, **NPM**, **Go**, **Hyperledger Fabric**, **Java JDK**, and **Gradle**. Additionally, connecting to the AWS EC2 Instance will be done from a device running **Windows 10**.

Explanations into the various components/prerequisites (e.g. Docker CE, Node.js, etc.) needed to run Hyperledger Fabric and Security concepts (e.g. IAM, Security Groups, etc.) are out of the scope of this Document.

Please note that this Document was written as a compilation of information obtained online. Sources will be provided whenever possible.

Prerequisites

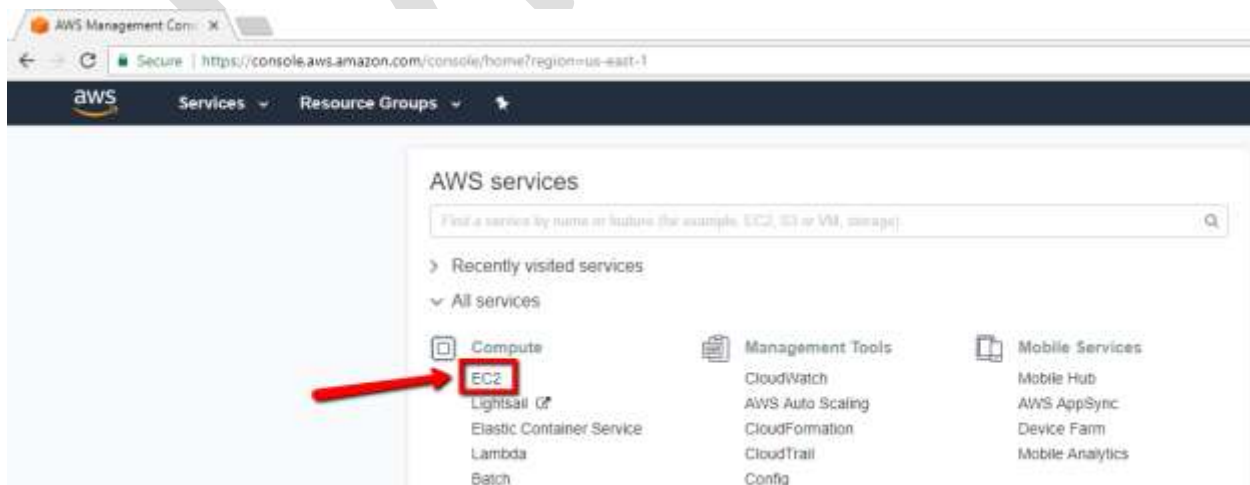
An [Amazon Web Services \(AWS\)](#) Account is required prior to starting this Installation Guide. Free AWS Accounts are available and was used to create this Document.

Installation Procedure

Set up an AWS EC2 Instance

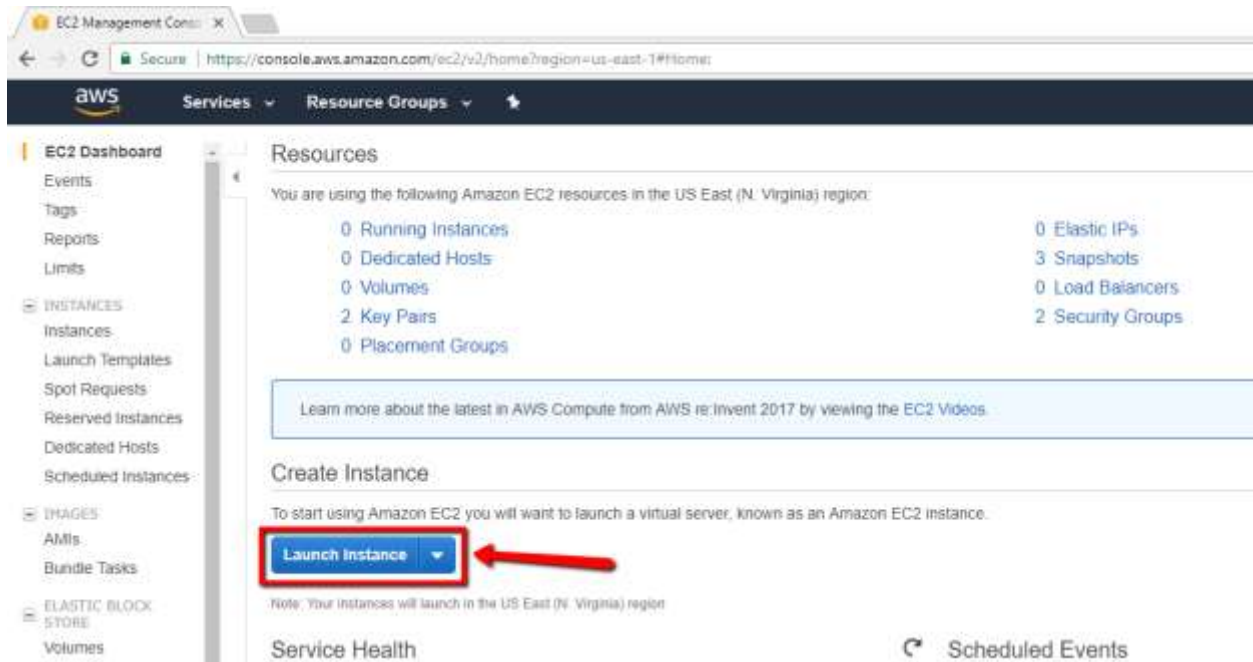
This section covers starting an AWS EC2 instance with Ubuntu 16.04 LTS. If you're already familiar with setting up and starting an AWS EC2 Instance, please proceed to Page 8.

Once logged into the AWS Console, the following screen should be visible:



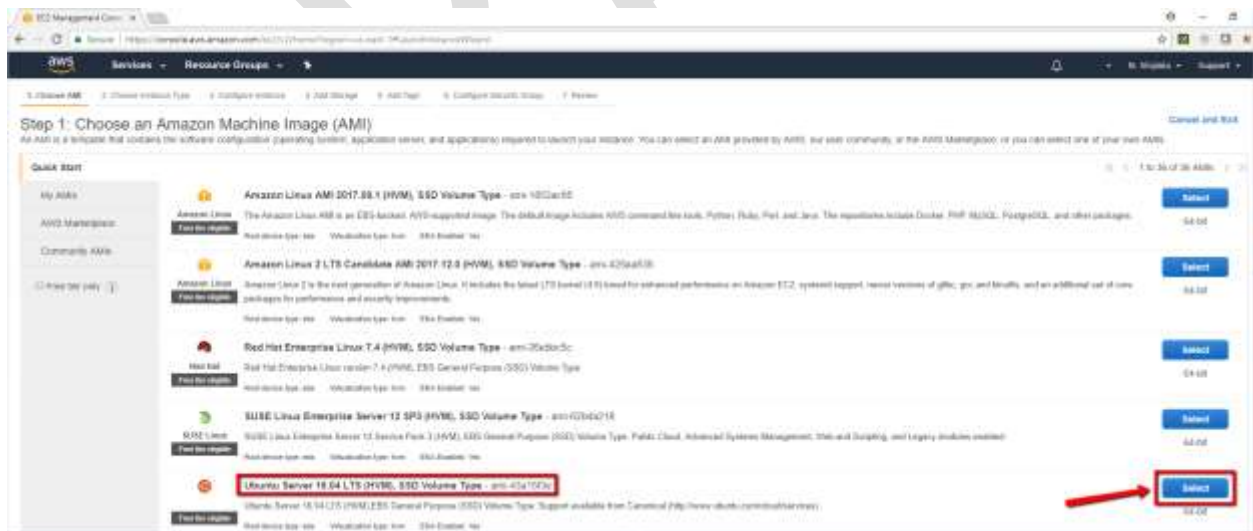
Click on the **EC2** link found under **Compute** in the **All services** section.


Click on the **Launch Instance** () button



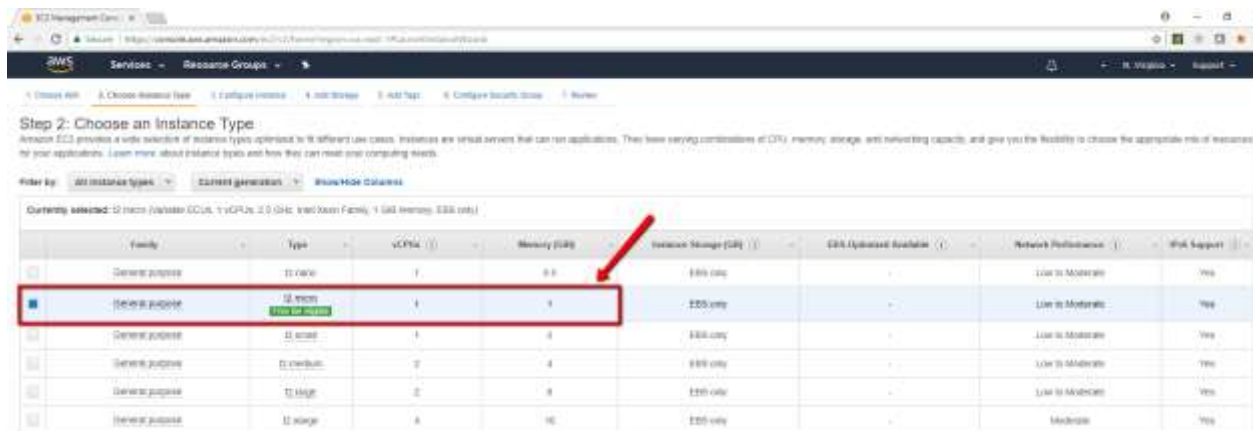
This button will start the process of creating an AWS EC2 Instance, where Hyperledger Fabric will be installed.

AWS EC2 will prompt the User to select an **Amazon Machine Image (AMI)**:



Click on the **Select** () button next to **Ubuntu Server 16.04 LTS (HVM)**.

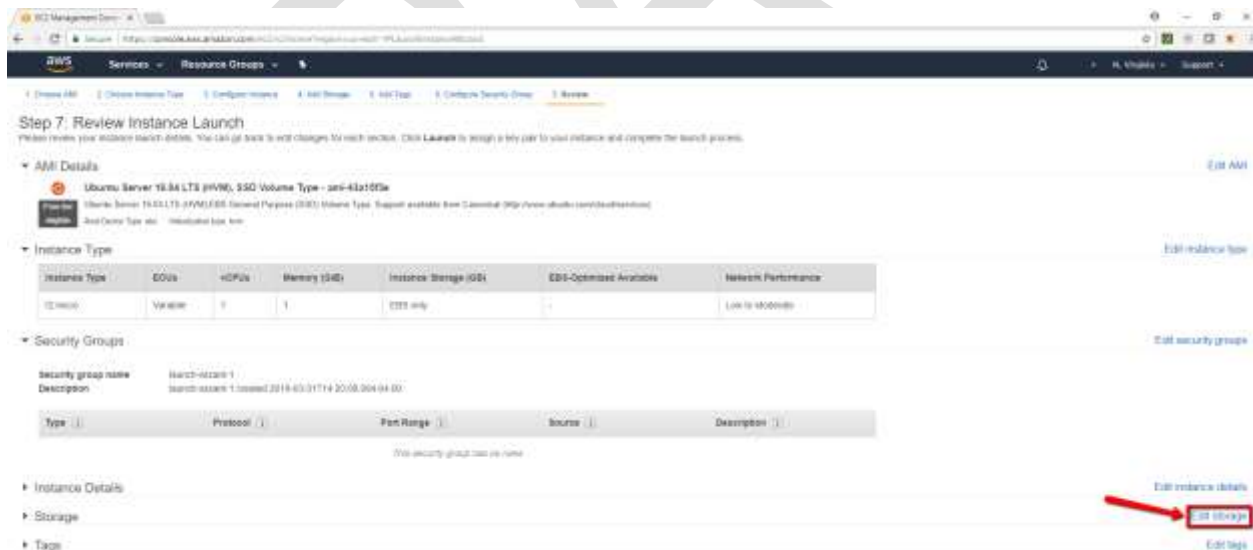
AWS EC2 will prompt the User to “configure” the Instance:



Note that the AWS Free Account has provided an option to select an Instance with one (1) vCPU and 1GB of RAM. For the purposes of this Document, this option will be sufficient.

Click on the **Review and Launch** ([Review and Launch](#)) button found at the bottom of the screen. This will take us directly to Step 7 of the process. Steps 3, 5, and 6 cover additional configuration options, but since the default entries are sufficient (for the purposes of this Document), they will be skipped.

Step 4 is relevant because the default value for the amount of Storage is insufficient to accommodate all the applications required to run/develop on Hyperledger Fabric. To adjust the Storage size, click on the **Edit Storage** link as indicated below:



This returns the User back to Step 4. By default, eight gigabytes (**8GB**) of Storage is the specified value. Since this is insufficient, change the value to a higher number (e.g. **20GB**):

aws Services - Resource Groups

1 Choose AMI 2 Choose Instance Type 3 Configure Instance 4 Add Storage 5 Add Tags 6 Configure Security Group 7 Review

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/sda1	snap-0a83ca22f09e8afbe8	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Click on the **Review and Launch** ([Review and Launch](#)) button found at the bottom of the screen. This will take the User back to Step 7 of the process:

aws Services - Resource Groups

1 Choose AMI 2 Choose Instance Type 3 Configure Instance 4 Add Storage 5 Add Tags 6 Configure Security Group 7 Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to create a key pair to your instance and complete the launch process.

AMI Details

Find AMI Ubuntu Server 16.04 LTS (HVM, SSD Volume Type - ami-42a193da) [Edit AMI](#)

Ubuntu Server 16.04 LTS (HVM, SSD Volume Type) Support available from Canonical (See [Ubuntu documentation](#))

Search Instance Type also: [InstanceType](#)

Instance Type

Instance Type	OS	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Instance	Network Performance
t2.micro	Windows	1	1	EBS only	<input checked="" type="checkbox"/>	Low to Moderate

Security Groups

Security group name: launch-ec2-1

DevOps team: launch-ec2-1 created 2016-05-01T14:20:18-04:00

Type	Protocol	Port Range	Source	Destination
This security group has no rules.				

Instance Details

Storage

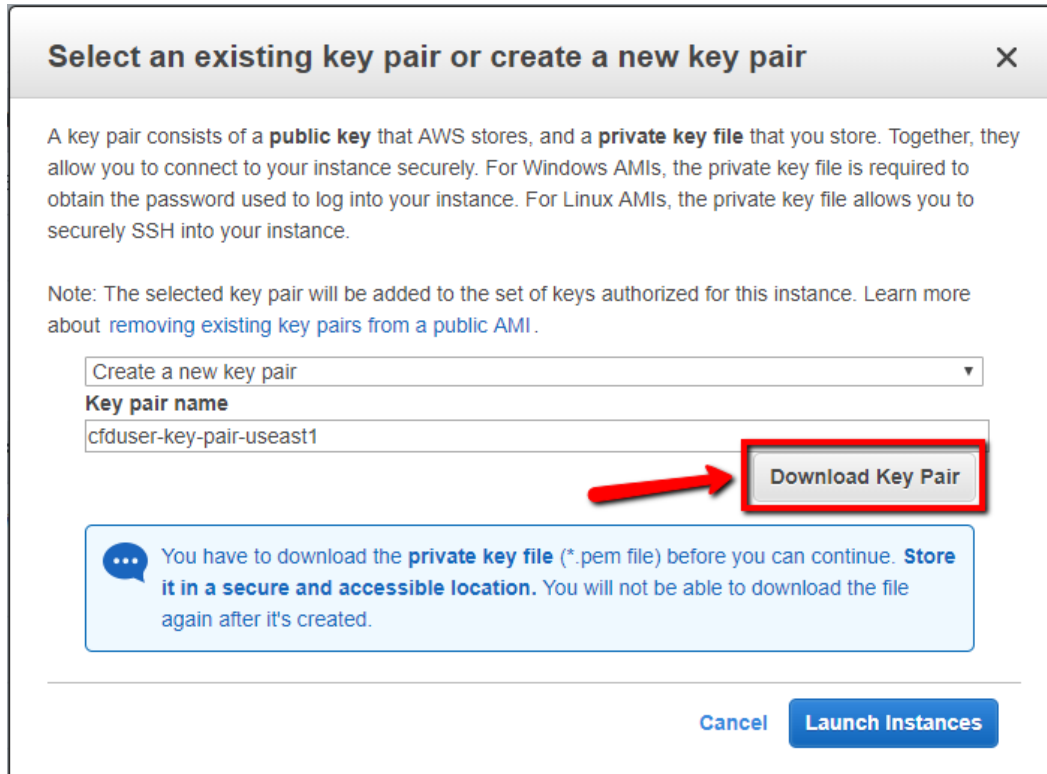
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0a83ca22f09e8afbe8	8	GP2	100 / 3000	N/A	Yes	Not Encrypted

Tags

[Cancel](#) [Previous](#) [Launch](#) [Edit tags](#)

Click on the **Launch** ([Launch](#)) button to start the creation of the AWS EC2 Instance.

AWS EC2 will display a dialog box asking the User to select an existing **Key Pair** or if a new one will be created:



Select an existing key pair or create a new key pair X

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](#).

Create a new key pair ▼

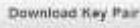
Key pair name

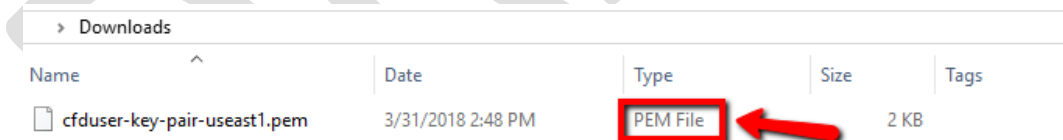
cfduser-key-pair-useast1


Download Key Pair


You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

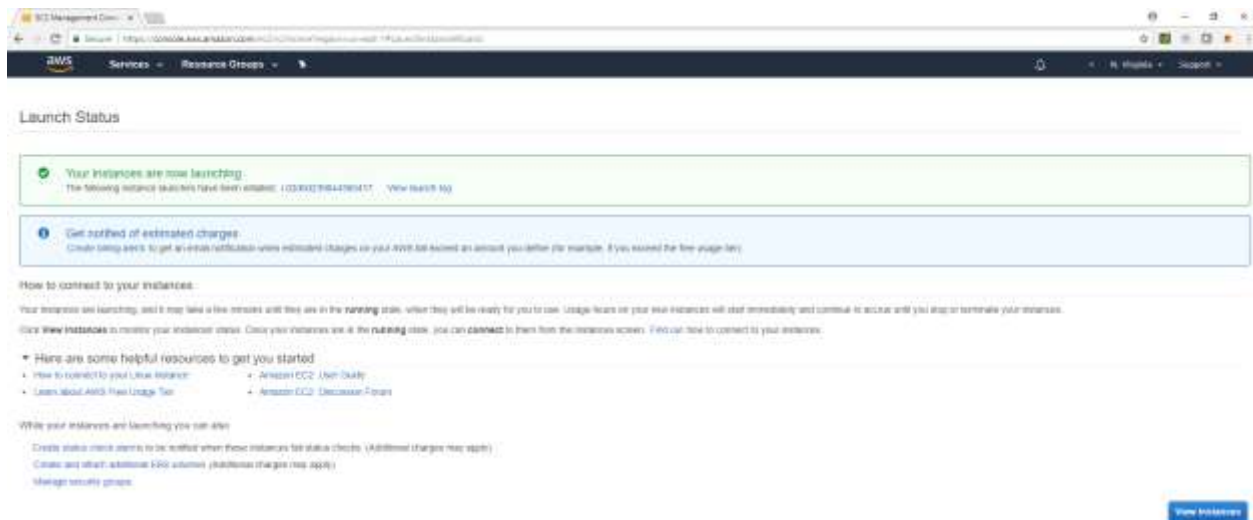
Select **Create a new key pair**, enter a corresponding name (e.g. **cfduser-key-pair-useast1**), and click on the **Download Key Pair** () button. This will download the Key Pair, as a **.pem file** into the **Downloads** folder:



Name	Date	Type	Size	Tags
 cfduser-key-pair-useast1.pem	3/31/2018 2:48 PM	PEM File	2 KB	

Once the Key Pair (.pem) file has been downloaded, click on the **Launch Instances** () button to launch the AWS EC2 Instance.

AWS EC2 will display a confirmation screen showing the Launch Status of the Instance:

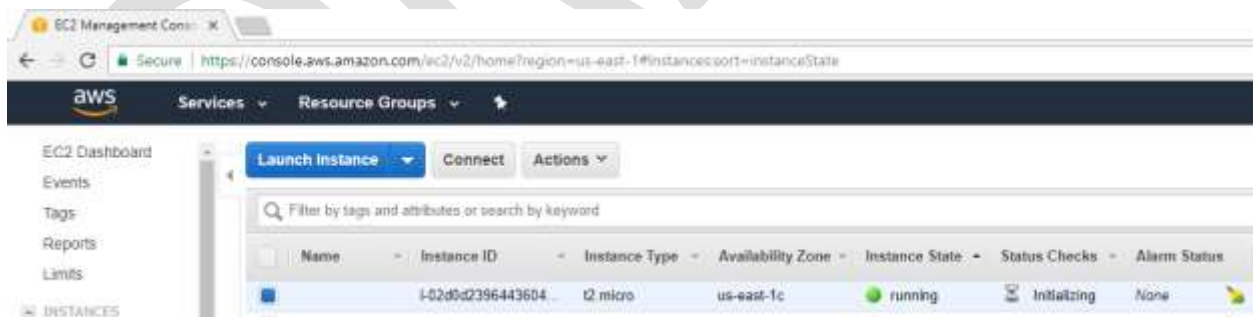


Note that there is a message stating that the instance is launching. Additionally, AWS provides a [link](#) to the documentation regarding how to connect to the newly-created instance. For example, when using Windows to connect to the AWS EC2 instance, AWS provides a detailed procedure using [PuTTY for Windows](#).

Alternatively, there's a [Youtube video](#) that also shows how to connect to the AWS EC2 Instance using PuTTY and PuTTYgen using the **.pem** file generated by the creation of the Key Pair.

Click on the **View Instances** ([View Instances](#)) button to display the AWS EC2 Instance.

AWS EC2 will display a screen showing the AWS EC2 Instance:



Note that the newly created AWS EC2 Instance is running.

To start the process of connecting to the AWS EC2 Instance, select the instance (indicated by the blue square) and then click on the **Connect** ([Connect](#)) button.

AWS EC2 will display the details to connect to the Instance:

Connect To Your Instance

I would like to connect with

☒ A standalone SSH client
☐ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))

2. Locate your private key file (cfduser-key-pair-useast1.pem). The wizard automatically detects the key you used to launch the instance.

3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 cfduser-key-pair-useast1.pem
```

4. Connect to your instance using its Public DNS:

ec2-34-235-157-236.compute-1.amazonaws.com

Example:

```
ssh -i "cfduser-key-pair-useast1.pem" ubuntu@ec2-34-235-157-236.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

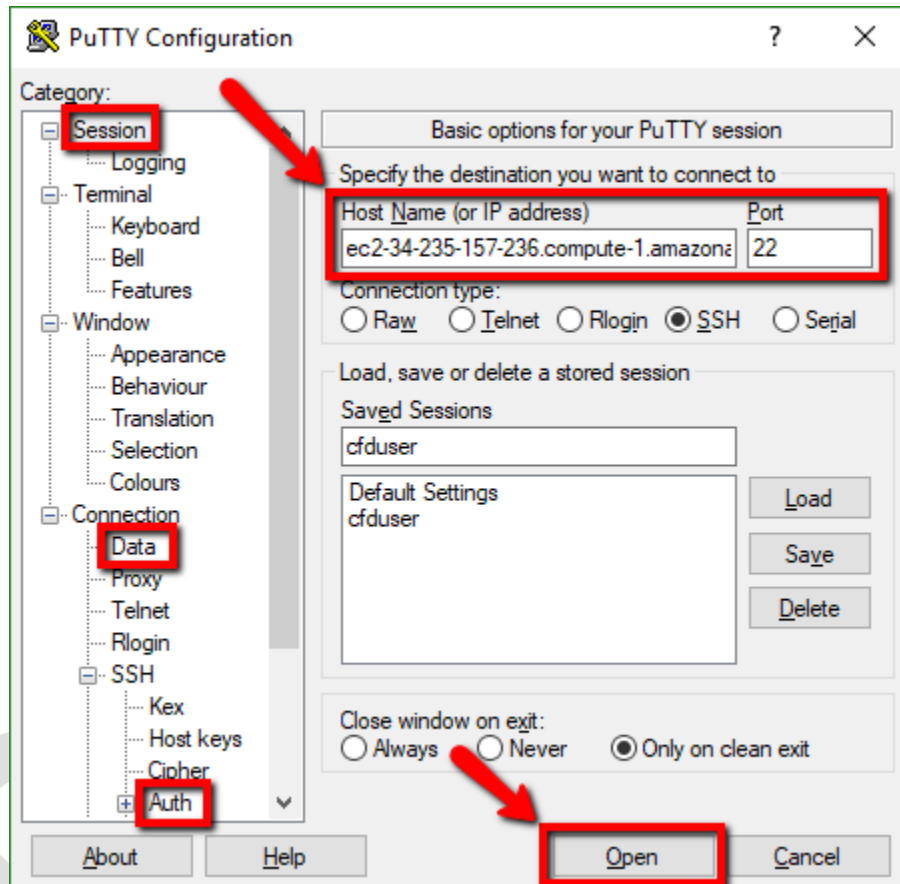
Close


Note the Public DNS value. Highlight the Public DNS, right-click and then select **Copy** (Ctrl+C).

Connect to the AWS EC2 Instance from a Windows device using PuTTY


Open PuTTY, and complete the following:

1. Under **Session**, enter the Public DNS of the AWS EC2 Instance in the **Host Name** field (Port should stay as **22**)
2. Under **Session** → **Connection** → **Data**, enter the User Name (e.g. **ubuntu**)
3. Under **Session** → **Connection** → **SSH** → **Auth**, specify the Key Pair (**.ppk**) file

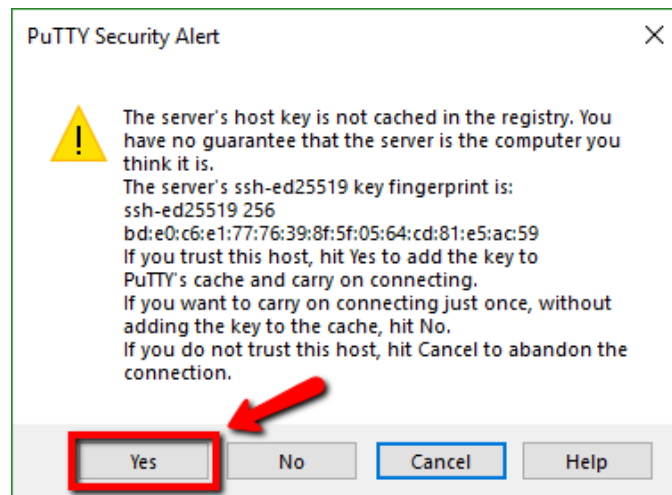


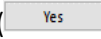
The above-mentioned entries can be saved (for future use) by creating a **Session**. This is done by entering a value in the field beneath **Saved Session** and then clicking on the **Save** () button.

To load a saved Session, select it from the list and then click on the **Load** () button.

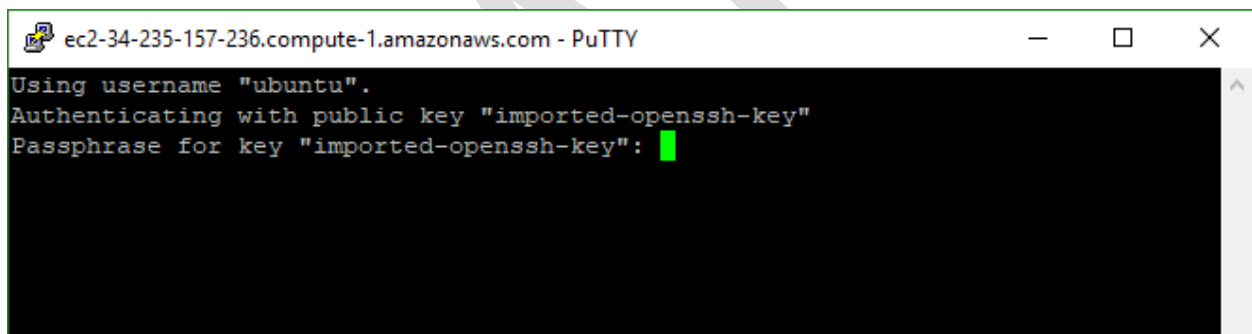
To connect to the AWS EC2 Instance, click on the **Open** () button.

PuTTY will prompt a Security Alert:



Since the host (i.e. the AWS EC2 Instance) is “trusted”, click on the **Yes** () button.

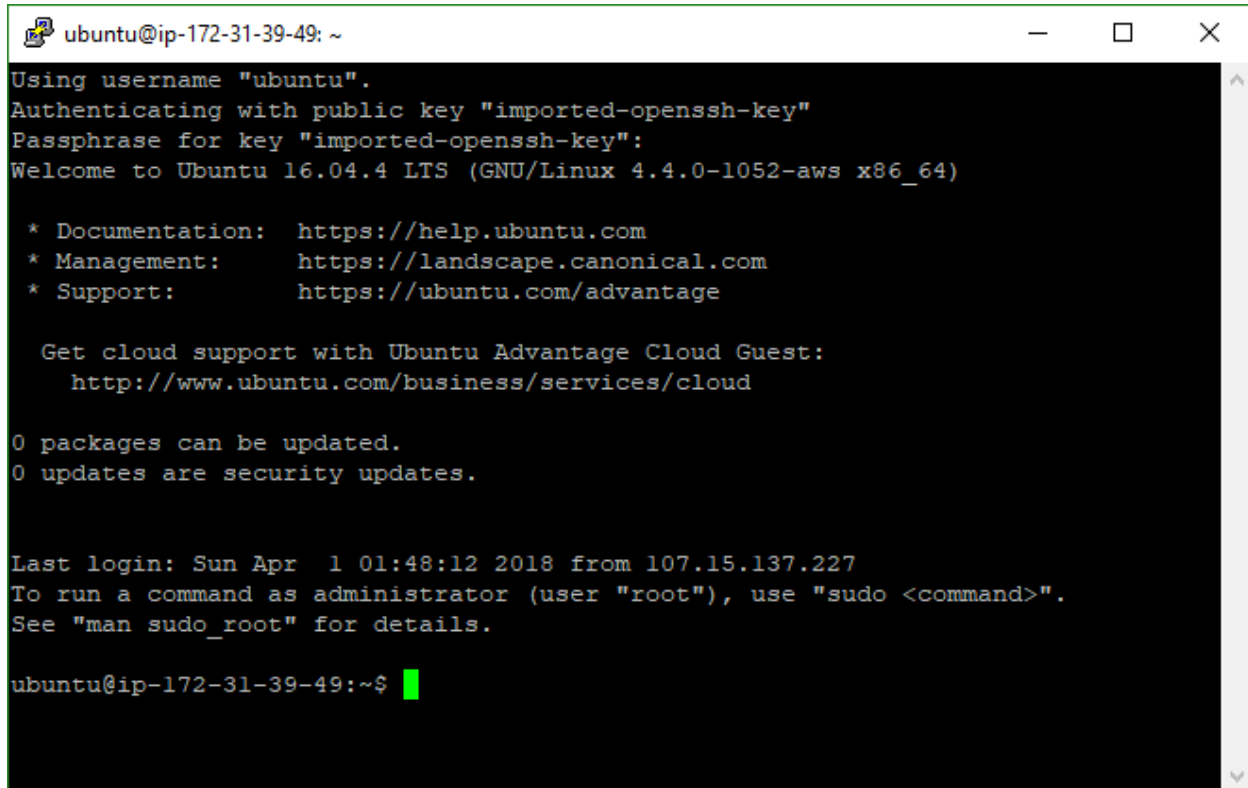
PuTTY will display a Terminal window asking for the User to enter a Password:



The Password is the one that was specified when creating the **Key Pair**.

Enter the Password and press the **Enter** Key.

Once the Password is authenticated, a Welcome Message is displayed:

A terminal window titled 'ubuntu@ip-172-31-39-49: ~' with standard window controls. The terminal output shows the login process for 'ubuntu' using a public key. It displays the Ubuntu version (16.04.4 LTS) and provides links for documentation, management, and support. It also shows that 0 packages can be updated and 0 security updates are available. The last login was on Sun Apr 1 01:48:12 2018 from 107.15.137.227. The prompt is 'ubuntu@ip-172-31-39-49:~\$' with a green cursor.

```
ubuntu@ip-172-31-39-49: ~
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-1052-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Sun Apr  1 01:48:12 2018 from 107.15.137.227
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-39-49:~$
```

At this point, the AWS EC2 Instance is running and is being accessed via a Terminal.

Install cURL

[cURL](#) is a command line tool for getting or sending files using URL syntax. For the purposes of this Document, it will be used to download the other applications into the Instance.

To install cURL, enter the following command in the Terminal Window and press the **Enter** Key:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get install curl
sudo apt install curl
```

The command should install the latest version of cURL. If the latest version is already installed, a corresponding message will be displayed. To check the version of cURL, enter the following command in the Terminal Window and press the **Enter** Key:

```
ubuntu@ip-172-31-39-49:~$ curl -V
curl -V
```

This command will display not only the cURL version, but will also the version of other related components (e.g. libcurl, GnuTLS, etc.). However, they are outside the scope of this Document.

Install Docker CE

[Docker](#) is a computer program that performs operating-system-level virtualization also known as containerization.

This section is the application of the installation procedure found in the official [Docker documentation](#).

Before installing Docker, check for updates of existing packages by entering the following command in the Terminal Window and pressing the **Enter** Key:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update
sudo apt-get update
```

The next step is to install packages that are needed to use repositories over HTTPS. Enter the following command in the Terminal Window and pressing the **Enter** Key:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get install \apt-transport-https \ca-certific
ates \curl \software-properties-common
sudo apt-get install \apt-transport-https \ca-certificates \curl \software-properties-common
```

When prompted by the system, enter **Y** (Yes) and press the **Enter** Key.

The next step is to add Docker's official GNU Privacy Guard (GPG) Key by entering the following command in the Terminal Window and pressing the **Enter** Key:

```
ubuntu@ip-172-31-39-49:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg
| sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

The next step is to confirm that the Instance has the key with the correct fingerprint. This is done by searching using the last eight (8) characters of the fingerprint in the Terminal Window as follows:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-key fingerprint 0EBFCD88
sudo apt-key fingerprint 0EBFCD88
```

Please note that the character before the "E" is a zero (0). Press the **Enter** key to continue. The Terminal window will display the full fingerprint (see below):

```
ubuntu@ip-172-31-39-49:~$ sudo apt-key fingerprint 0EBFCD88
pub 4096R/0EBFCD88 2017-02-22
    Key fingerprint = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid          Docker Release (CE deb) <docker@docker.com>
sub 4096R/F273FCD8 2017-02-22
ubuntu@ip-172-31-39-49:~$
```

The next step is to set up the stable Docker repository by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Before installing Docker CE, check again for any updates to packages by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update
```

```
sudo apt-get update
```

The next step is to install Docker CE by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get install docker-ce
```

```
sudo apt-get install docker-ce
```

When prompted by the system, enter **Y** (Yes) and press the **Enter** Key.

To verify that Docker CE was installed successfully, display the version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ docker -v
```

```
docker -v
```

If the Docker CE version is displayed, it means that Docker was installed. The next step is to test Docker CE by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo systemctl status docker
```

```
sudo systemctl status docker
```

If Docker CE was installed successfully, a screen similar to the one shown below should appear:

```
ubuntu@ip-172-31-39-49: ~
ubuntu@ip-172-31-39-49:~$ docker -v
Docker version 18.03.0-ce, build 0520e24
ubuntu@ip-172-31-39-49:~$ clear
ubuntu@ip-172-31-39-49:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese
   Active: active (running) since Sun 2018-04-01 02:06:57 UTC; 5min ago
     Docs: https://docs.docker.com
    Main PID: 13899 (dockerd)
      CGroup: /system.slice/docker.service
              └─13899 /usr/bin/dockerd -H fd://
                  └─13905 docker-containerd --config /var/run/docker/containerd/con

Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.57
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.57
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.57
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.57
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.70
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.74
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.75
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.75
Apr 01 02:06:57 ip-172-31-39-49 systemd[1]: Started Docker Application Conta
Apr 01 02:06:57 ip-172-31-39-49 dockerd[13899]: time="2018-04-01T02:06:57.77
lines 1-19/19 (END)
```

To exit this screen, press **Ctrl+C**.

As a final test of Docker CE, run the **Hello World** Test by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo docker run hello-world
```

sudo docker run hello-world

The Test is successful if the following message appears:

```
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://cloud.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/

ubuntu@ip-172-31-39-49:~$
```

The last step is to assign the User by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo usermod -aG docker ${USER}
```

sudo usermod -aG docker \${USER}

Install Docker Compose

As per the official [Docker Compose Documentation](#), **Compose** is a tool for defining and running multi-container Docker applications.

This section is an application of the [installation procedure](#).

The first step is to download the latest version of Docker Compose by entering the following command in the Terminal Window:

```
ubuntu@ip-172-31-39-49:~$ sudo curl -L https://github.com/docker/compose/releases/download/1.20.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
sudo curl -L https://github.com/docker/compose/releases/download/1.20.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

Note that the **-o** in the command is a dash followed by a small letter **o**.

The next step is to apply executable permissions to the binary by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo chmod +x /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

The last step is to verify that Docker Compose was installed successfully, display the version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ docker-compose --version
docker-compose --version
```

Note that there are two (2) dashes before the word **version**.

Install Node.js

As per the [Node.js website](#), **Node.js**® is a JavaScript runtime built on [Chrome's V8 JavaScript Engine](#). It basically allows for the execution of JavaScript on the server side.

This section is an application of the [installation procedure](#) found in the Node.js website.

The first step is to download Node.js by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -  
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
```

Install Node.js by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get install -y nodejs  
sudo apt-get install -y nodejs
```

To verify that Node.js was installed successfully, check the version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ node -v  
node -v
```

As part of installing Node.js, **npm** is also installed. As per the [NPM website](#), **npm** is the package manager for JavaScript and the world's largest software registry.

To verify that npm was installed successfully, check the version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ npm -v  
npm -v
```

Although npm was included in the Node.js installation, it is not the latest version. To get the latest version of npm, enter the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo npm install npm@latest -g  
sudo npm install npm@latest -g
```

To verify that npm was updated successfully, check the version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ npm -v  
npm -v
```


Install Go Programming Language

As per the [Go website](#), Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.

This section references an installation procedure found in a [Medium](#) article written by [Patrick Dahlke](#) and was updated to use the latest version of Go (when the Document was created).

The first step is to update and upgrade packages by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update && sudo apt-get -y upgrade
sudo apt-get update && sudo apt-get -y upgrade
```

The next step is to create a subdirectory by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ mkdir Downloads
mkdir Downloads
```

Change directories by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ cd Downloads
cd Downloads
```

The next step is to download the Go package, into the **Downloads** subdirectory, by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo curl -O https://storage.googleapis.com/golang/go1.10.1.linux-amd64.tar.gz
sudo curl -O https://storage.googleapis.com/golang/go1.10.1.linux-amd64.tar.gz
```

The next step is to unpack the package into the same **Downloads** subdirectory by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo tar -xvf go1.10.1.linux-amd64.tar.gz
sudo tar -xvf go1.10.1.linux-amd64.tar.gz
```

Once the Go package has been unpacked, the next step is to move the **go** subdirectory to a different location by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo mv go /usr/local
sudo mv go /usr/local
```

The next step is to set the PATH value by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.profile
echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.profile
```

Refresh the profile by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ source ~/.profile
source ~/.profile
```

The last step is to check the version of Go by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ go version
go version
```

If the Go version value is returned, it confirms that the Go Programming Language was installed successfully.

In Hyperledger Fabric, “[Chaincode](#)” is written in Go.

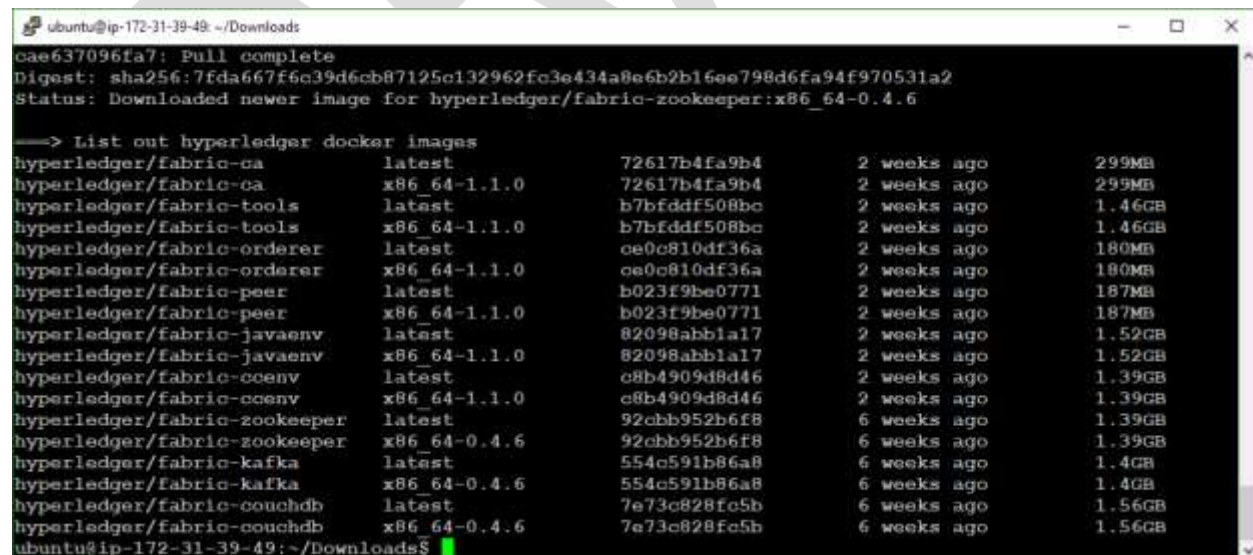
At this point, all the prerequisites for installing Hyperledger Fabric have been installed into the AWS EC2 Instance.

Install Hyperledger Fabric

The first step is to download the Platform-specific Binaries as specified in the official [Hyperledger Fabric Documentation](#):

```
ubuntu@ip-172-31-39-49:~/Downloads$ curl -sSL https://goo.gl/6wtTN5 | sudo bash -s 1.1.0
curl -sSL https://goo.gl/6wtTN5 | bash -s 1.1.0
```

This step may take a couple of minutes to complete since it's downloading all the Docker Images. Once completed, the result would be a screen similar to the one shown below:



```
ca6637096fa7: Pull complete
Digest: sha256:7fda667f6c39d6cb87125c132962fc3e434a8e6b2b16ee798d6fa94f970531a2
Status: Downloaded newer image for hyperledger/fabric-zookeeper:x86_64-0.4.6

--> List out hyperledger docker images
hyperledger/fabric-ca          latest          72617b4fa9b4   2 weeks ago    299MB
hyperledger/fabric-ca          x86_64-1.1.0   72617b4fa9b4   2 weeks ago    299MB
hyperledger/fabric-tools       latest          b7bfddf508bc   2 weeks ago    1.46GB
hyperledger/fabric-tools       x86_64-1.1.0   b7bfddf508bc   2 weeks ago    1.46GB
hyperledger/fabric-orderer     latest          ce0c810df36a   2 weeks ago    180MB
hyperledger/fabric-orderer     x86_64-1.1.0   ce0c810df36a   2 weeks ago    180MB
hyperledger/fabric-peer        latest          b023f9be0771   2 weeks ago    187MB
hyperledger/fabric-peer        x86_64-1.1.0   b023f9be0771   2 weeks ago    187MB
hyperledger/fabric-javaenv     latest          82098abb1a17   2 weeks ago    1.52GB
hyperledger/fabric-javaenv     x86_64-1.1.0   82098abb1a17   2 weeks ago    1.52GB
hyperledger/fabric-ccenv       latest          c8b4909d8d46   2 weeks ago    1.39GB
hyperledger/fabric-ccenv       x86_64-1.1.0   c8b4909d8d46   2 weeks ago    1.39GB
hyperledger/fabric-zookeeper   latest          92cbb952b6f8   6 weeks ago    1.39GB
hyperledger/fabric-zookeeper   x86_64-0.4.6   92cbb952b6f8   6 weeks ago    1.39GB
hyperledger/fabric-kafka       latest          554c591b86a8   6 weeks ago    1.4GB
hyperledger/fabric-kafka       x86_64-0.4.6   554c591b86a8   6 weeks ago    1.4GB
hyperledger/fabric-couchdb     latest          7e73c828fc5b   6 weeks ago    1.56GB
hyperledger/fabric-couchdb     x86_64-0.4.6   7e73c828fc5b   6 weeks ago    1.56GB
ubuntu@ip-172-31-39-49:~/Downloads$
```

The important part is that for each of the components (e.g. fabric-ca, fabric-tools, etc.), there is a line with **latest**.

The next step is to download the Hyperledger Fabric samples by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ git clone -b master https://github.com/hyperledger/fabric-samples.git
git clone -b master https://github.com/hyperledger/fabric-samples.git
```

The next step is to export the PATH where the **bin** directory is located by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ export PATH=~/Downloads/bin:$PATH
export PATH=~/Downloads/bin:$PATH
```

At this point, Hyperledger Fabric and sample programs have been installed into the AWS EC2 Instance.

Install Java JDK 8

Java JDK 8

This section references the installation procedure found on this [website](#).

First, packages are checked if there are any updates available. Should there be any updates available, the package should be upgraded prior to the installation of any new packages. This is done by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update && sudo apt-get upgrade
sudo apt-get update && sudo apt-get upgrade
```

The next step is to install Java JDK 8 by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get install default-jdk
sudo apt-get install default-jdk
```

To confirm that Java JDK 8 was installed successfully, display the version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ javac -version
javac -version
```

At the time this Document was written, Java JDK 9 (i.e. **v9.0.4**) has been designated as a stable release. The reason why Java JDK 9 was chosen to be part of this installation is because there are still issue with using this version with Gradle.

Install Unzip Package

The Gradle application requires the **Unzip** package to be installed in the AWS EC2 Instance.

The first step is to check for updates. Should there be any updates available, the package should be upgraded prior to the installation of any new packages. This is done by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update && sudo apt-get upgrade
sudo apt-get update && sudo apt-get upgrade
```

The next step is to install the **Unzip** package by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt install unzip
sudo apt install unzip
```

Install Gradle

As per the [Gradle website](#), Gradle helps team build, automate and deliver better software, faster.

This section is an application of the [installation instructions](#) found on the Gradle website.

The first step is to check for updates. Should there be any updates available, the package should be upgraded prior to the installation of any new packages. This is done by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update && sudo apt-get upgrade
sudo apt-get update && sudo apt-get upgrade
```

After checking for updates, the next step is to download Gradle using the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ wget https://services.gradle.org/distributions/gradle-4.6-bin.zip
wget https://services.gradle.org/distributions/gradle-4.6-bin.zip
```

The next step is to unzip Gradle into a given location by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo unzip gradle-4.6-bin.zip -d /usr/local
sudo unzip gradle-4.6-bin.zip -d /usr/local
```

The next step is to create a shortcut to the Gradle distribution by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo ln -s gradle-4.6 gradle
sudo ln -s gradle-4.6 gradle
```

Before installing Gradle, another check for package updates is done by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update && sudo apt-get upgrade  
sudo apt-get update && sudo apt-get upgrade
```

The next step is to install Gradle by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo apt install gradle  
sudo apt install gradle
```

When prompted, enter **Y** to continue with the installation.

The next step is to check the Gradle version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ gradle -v  
gradle -v
```

Gradle is installed, but it needs to be updated. To update Gradle, the repository will be downloaded using the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ sudo add-apt-repository ppa:cwchien/gradle  
sudo add-apt-repository ppa:cwchien/gradle
```

Before updating Gradle, another check for package updates is done by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get update  
sudo apt-get update
```

The next step is to update Gradle by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~$ sudo apt-get upgrade gradle  
sudo apt-get upgrade gradle
```

The last step is to check the Gradle version by entering the following command in the Terminal window:

```
ubuntu@ip-172-31-39-49:~/Downloads$ gradle -v  
gradle -v
```

The Gradle version should correspond to a more recent release.

At this point, the AWS EC2 Instance has the required components to be part of a Hyperledger Fabric network. To close the Terminal window, enter the following command:

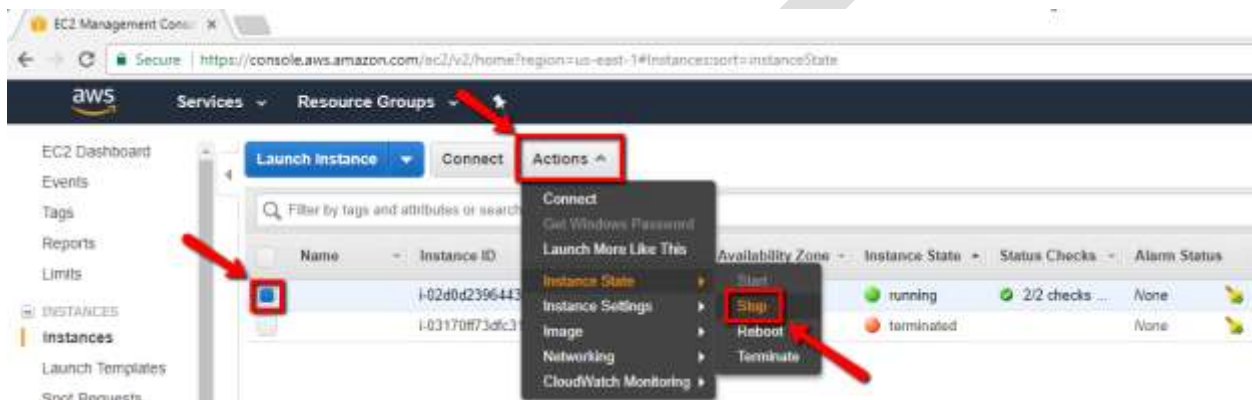
```
ubuntu@ip-172-31-39-49:~$ exit  
Exit
```

Stop an AWS EC2 Instance

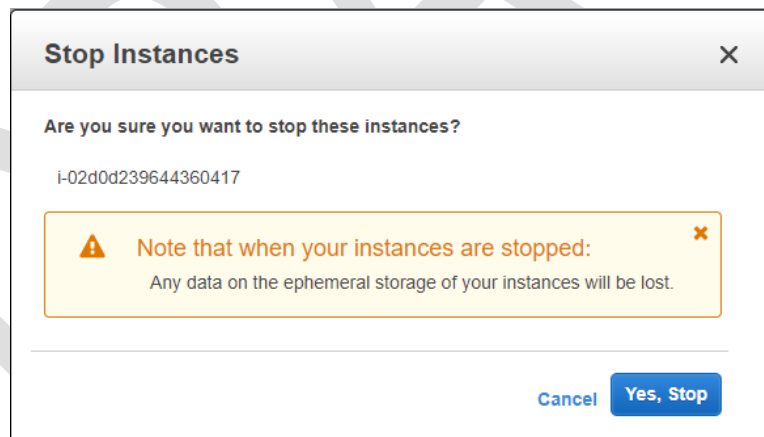
At the end of the installation process, the last step was to exit from the Terminal window. Although the Terminal window is closed, the AWS EC2 Instance is still running (i.e. clock is still running).


For a detailed explanation regarding the stopping and starting of AWS EC2 Instances, please refer to the official [Documentation](#).

To stop an AWS EC2 Instance, return to the AWS EC2 Dashboard, select the AWS EC2 Instance to be stopped, and then follow the menu path **Actions** → **Instance State** → **Stop** (see below):



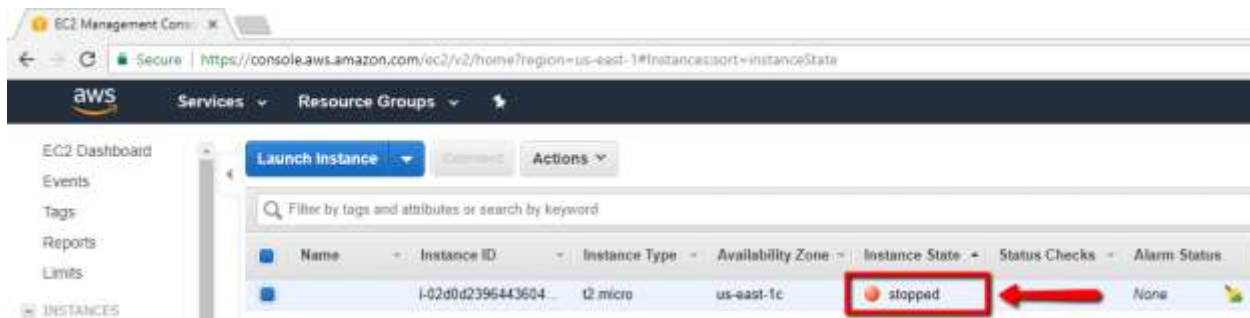
AWS EC2 will prompt the User to confirm that the AWS EC2 Instance will be stopped (see below):



To stop the AWS EC2 Instance, click on the **Yes, Stop** () button.

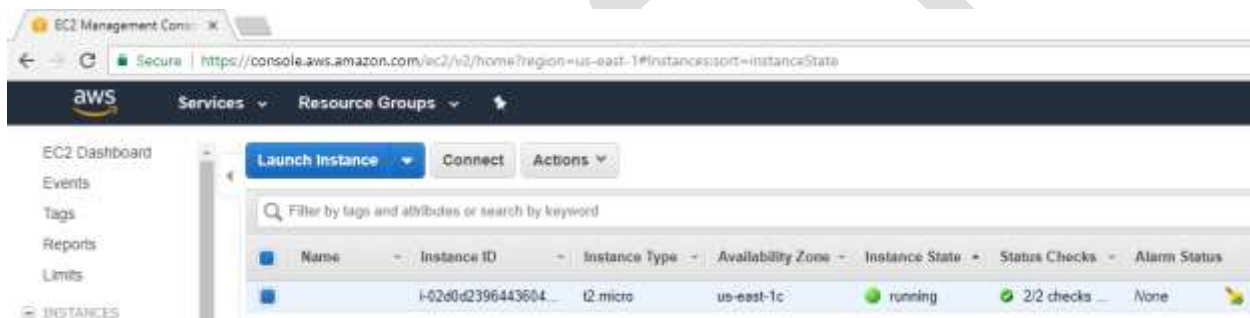
At this point, AWS EC2 will begin the process of stopping the AWS Instance.

After a short time, the AWS EC2 Console will show that the AWS EC2 Instance has been stopped:



Restarting the AWS EC2 Instance uses an almost identical procedure. The only difference is that the menu path is **Actions → Instance State → Start**

Once the AWS EC2 Instance has been restarted, the AWS EC2 Console will automatically refresh the status:



Please note that the Public DNS may have changed. So, when connecting to the restarted AWS EC2 Instance, the **Host Name** value must be changed in the **Saved Session** in **PuTTY**.

Alternatively, if the User had selected to **terminate** the AWS EC2 Instance (instead of stopping it), then there's no option to restart the instance at a later time.

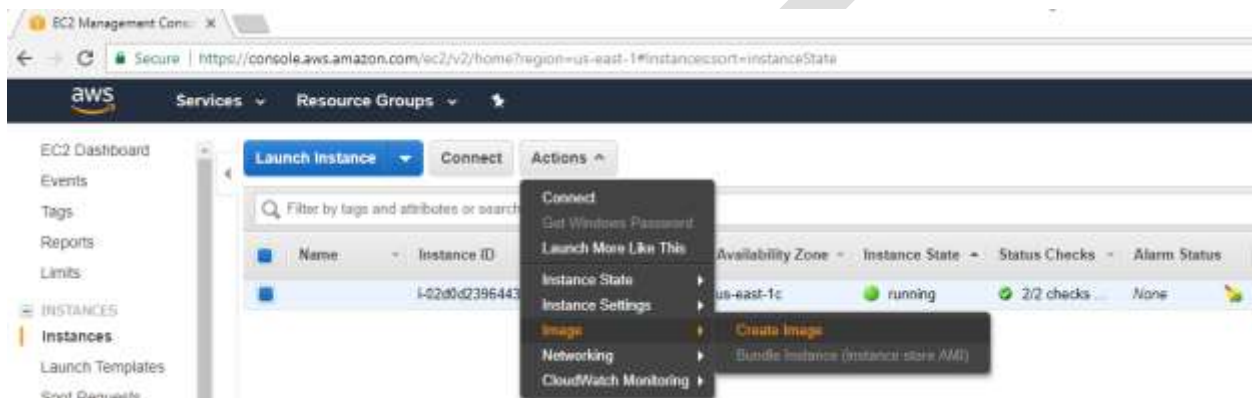
Please refer to the official AWS [Documentation](#) that explains the difference between stopping and terminating an AWS EC2 Instance.

Create an Amazon Machine Image (AMI)

As per the official AWS [Documentation](#) of Amazon Machine Images (AMI), contains all the information required to launch an Instance.

For example, an AMI of the AWS EC2 Instance can be created. Once the AMI has been created, it can be used to launch a second AWS EC2 Instance without having to repeat the installation procedure covered by this Document.

Creating the AMI uses a similar procedure but the menu path is **Actions** → **Image** → **Create Image**:



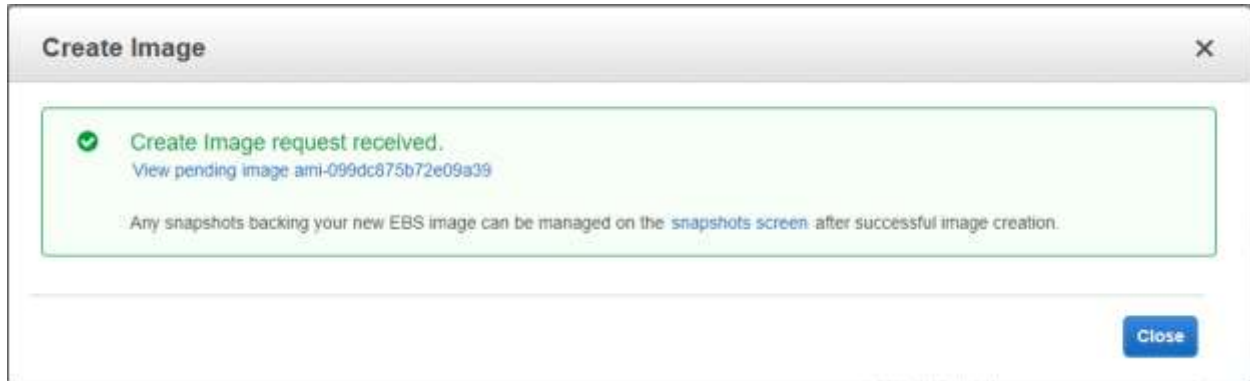
AWS EC2 will prompt the User to enter details for the AMI to be created:

A screenshot of the 'Create Image' dialog box in the AWS Management Console. The dialog contains the following fields and options:

- Instance ID**: i-02d0d239644360417
- Image name**: Hyperledger Fabric on Ubuntu 16.04
- Image description**: Hyperledger Fabric v1.1 on Ubuntu 16.04
- No reboot**: ☐
- Instance Volumes**: A table with columns: Volume Type, Device, Snapshot, Size (GiB), Volume Type, IOPS, Throughput (MB/s), Delete on Termination, and Encrypted. The table contains one row for the root volume: Root, /dev/sda1, snap-0a83a22928e9a1be9, 20, General Purpose SSD (GP2), 100 / 3000, N/A, ☒, Not Encrypted.
- Add New Volume**: A button to add additional volumes.
- Total size of EBS Volumes**: 20 GiB
- When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.**
- Buttons**: Cancel and Create Image.

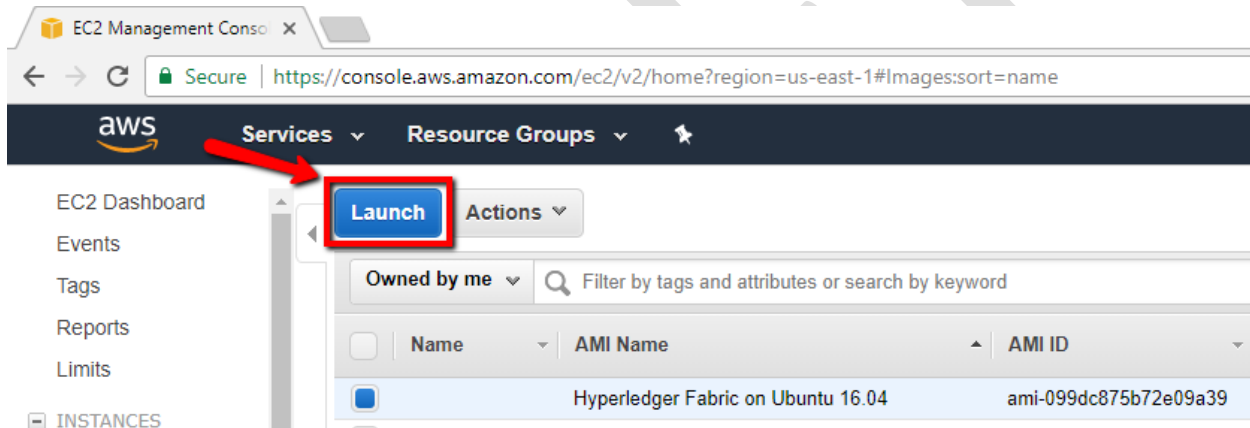
Enter values for **Image name**, **Image description**, and click on the **Create Image** ([Create image](#)) button to create the AMI.

AWS EC2 will display a confirmation screen stating that the AMI is being created:



Click on the **Close** (Close) button to close the message.

After a short while, the newly created AMI will be available in the AMI section of AWS:



Additionally, the User can launch an AWS EC2 Instance from the AMI Console.