

---

# PG DO - CI/CD Pipeline with Jenkins

Lab Guide



Get Certified. Get Ahead.

This section will guide you to:

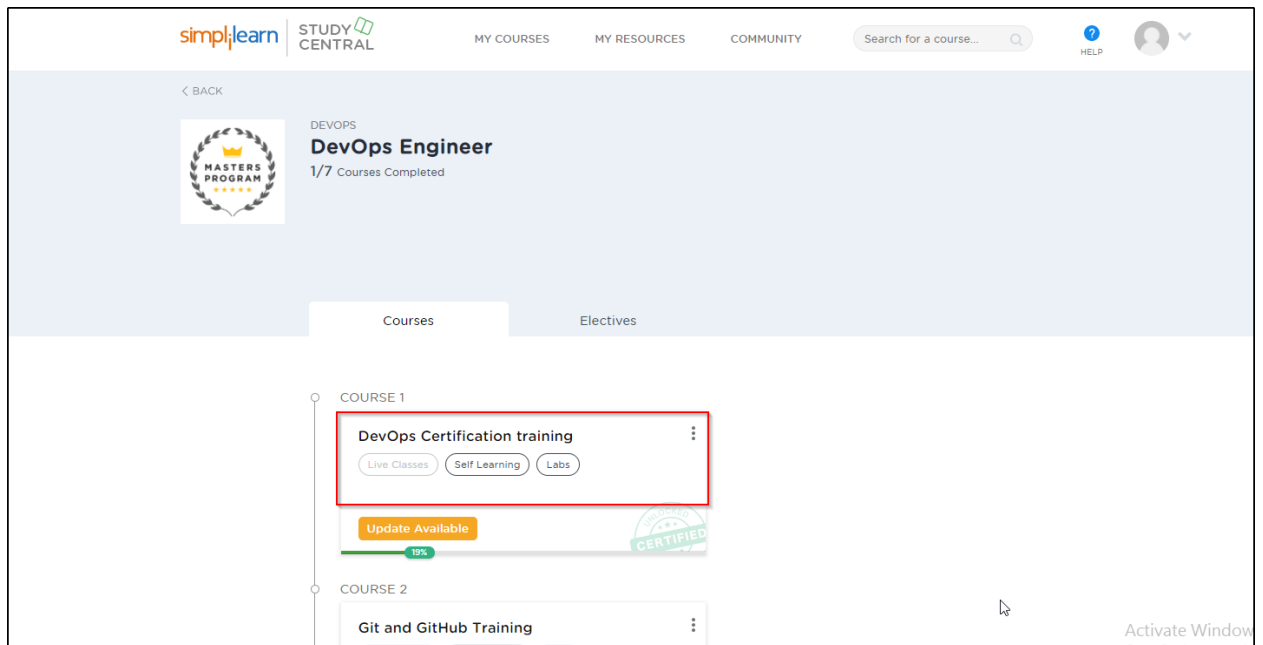
- Use labs to execute all demos included in this course

This lab has two subsections, namely:

1. Starting practice labs on LMS
2. Using different IDEs and software required for the PG DO - CI/CD Pipeline with Jenkins course

### Step 1: Starting practice labs on LMS

- Login to Simplilearn LMS
- Go to the respective course



- On the left, you will find the course ToC page
- To its left, you will find the **PRACTICE LABS** tab
- Click on it

**DevOps Certification training**  
0 Classes completed | 19% Self-Learning Videos Watched | 1/2 Projects Done

**Manage and Monitor the Docker Containers with Dry Tool**

**DESCRIPTION**  
Install Dry, demonstrate its interaction with Docker containers and images, and monitor it.

**Problem Statement Scenario:**  
Exact Technology Solutions hired you as a DevOps Engineer. The company decides to implement DevOps to develop and deliver the products. Since Exact is an Agile organization, they follow Scrum methodology to develop the projects incrementally. Stakeholders discussed implementing Docker during development and testing stages to avoid issues related to configuration and tools. As it is an additional investment on resources and tools, some of the stakeholders denied the change. Your Manager suggested a demonstration on Docker and monitoring the status of the Docker containers and images. Since you are the DevOps Engineer, you are assigned the task to demonstrate how Docker will help and how the containers can be managed effectively. You agreed to demonstrate and list the advantages, disadvantages, and document the tasks involved:

- Install Docker and create multiple containers
- Install Dry and demonstrate its interaction with Docker containers and Docker images
- Demonstrate its monitoring of Docker containers

Your goal is to demonstrate the tasks with proper documentation

**You must use the following tools:**

- Docker - To build and pack the application in a Docker container
- Linux (Ubuntu) - As the base operating system to start and execute the project

**Following requirements should be met:**

- Document the step-by-step process: initial installation to final production
- Documentation of the investment on resource utilization, cost involved in setup, and the system configurations
- Multiple containers should be running in a detached mode
- Monitoring and maintaining the report on Docker containers and Docker images

- As a new window opens, read the instructions and click on **LAUNCH LAB**
- This will launch practice labs for this course

**DevOps Certification training**  
0 Classes completed | 19% Self-Learning Videos Watched | 1/2 Projects Done

**DevOps Lab** **DevOps in AWS**

**Current Lab : DevOps Practitioner - AWS**

Access Information | Lab Details | Components | Log Details | Usage Details

**Applications**

Webconsole RDP Access

**Instance Actions**

**Start Instance**

Instance status : **Stopped**  
Last updated at : 2021-05-04 14:22

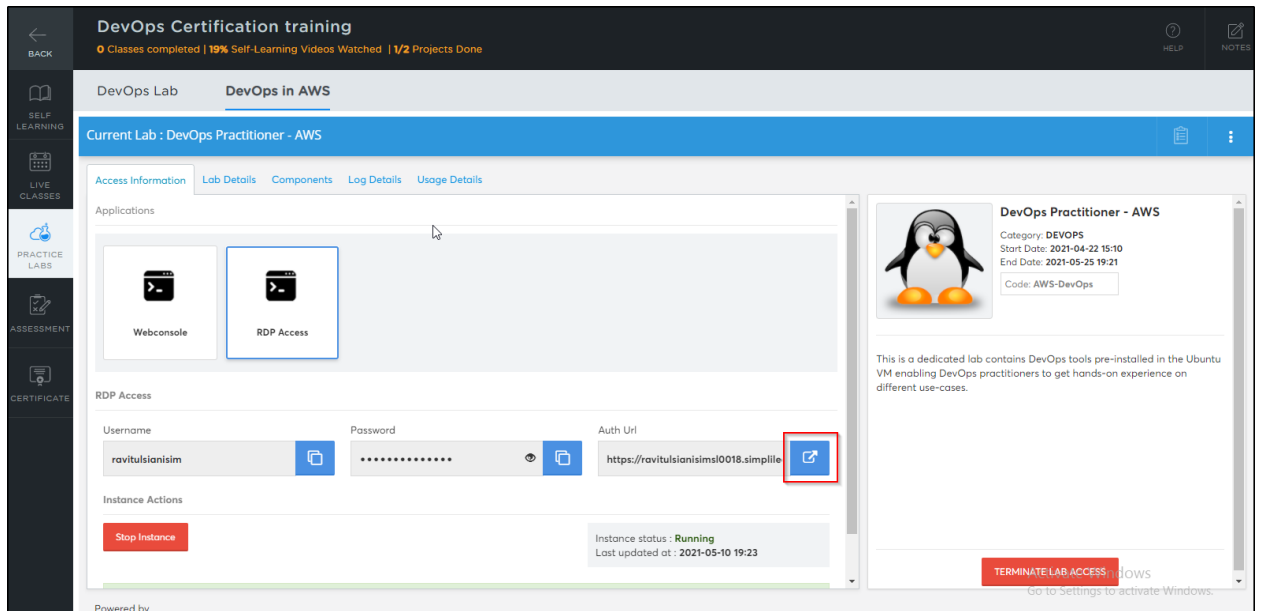
The lab environment is currently in Stopped state, so the access is suspended. Cannot connect to the lab environment until it is started. Please click on Start Instance to wake up the environment.

**DevOps Practitioner - AWS**  
Category: DEVOPS  
Start Date: 2021-04-22 15:10  
End Date: 2021-05-25 19:21  
Code: AWS-DevOps

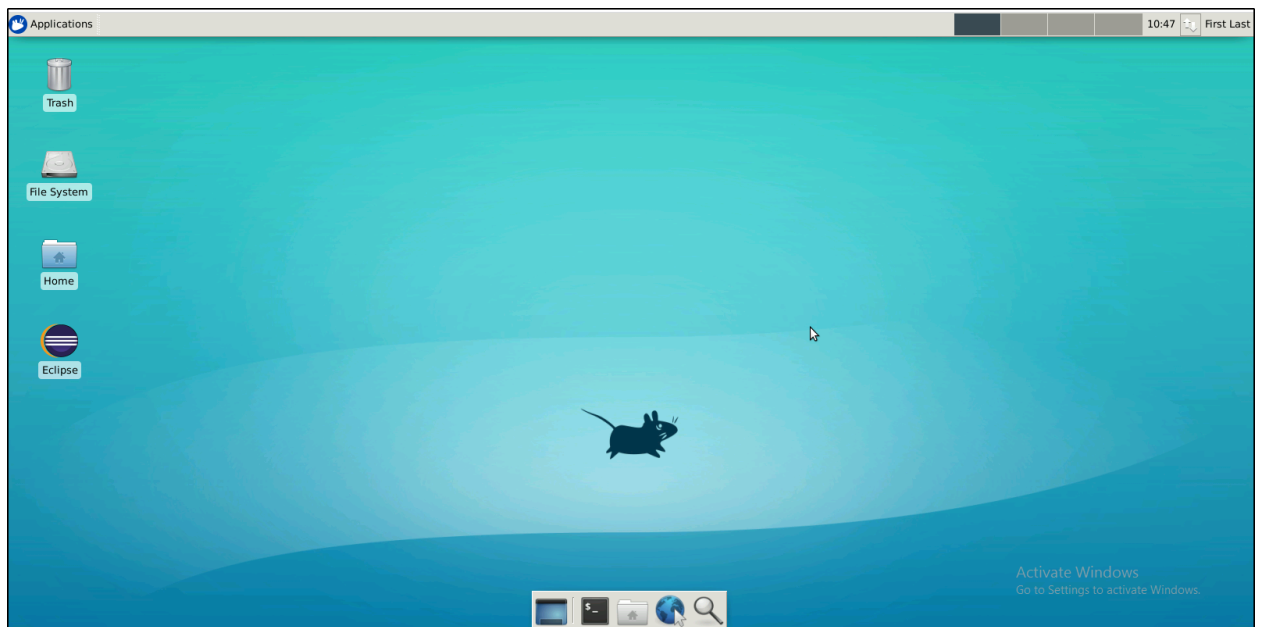
This is a dedicated lab contains DevOps tools pre-installed in the Ubuntu VM enabling DevOps practitioners to get hands-on experience on different use-cases.

**TERMINATE LAB ACCESS** | **HOWS**

- Once the lab instance is started, click on the *Auth Url* as shown below:

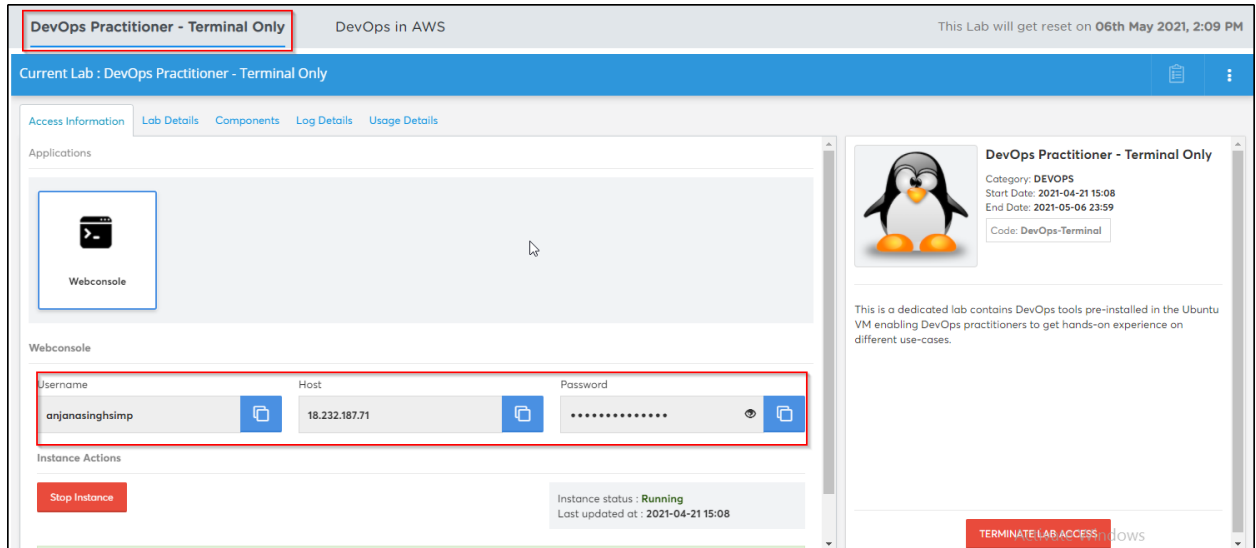


- You will be able to access IDEs and software which are present in labs

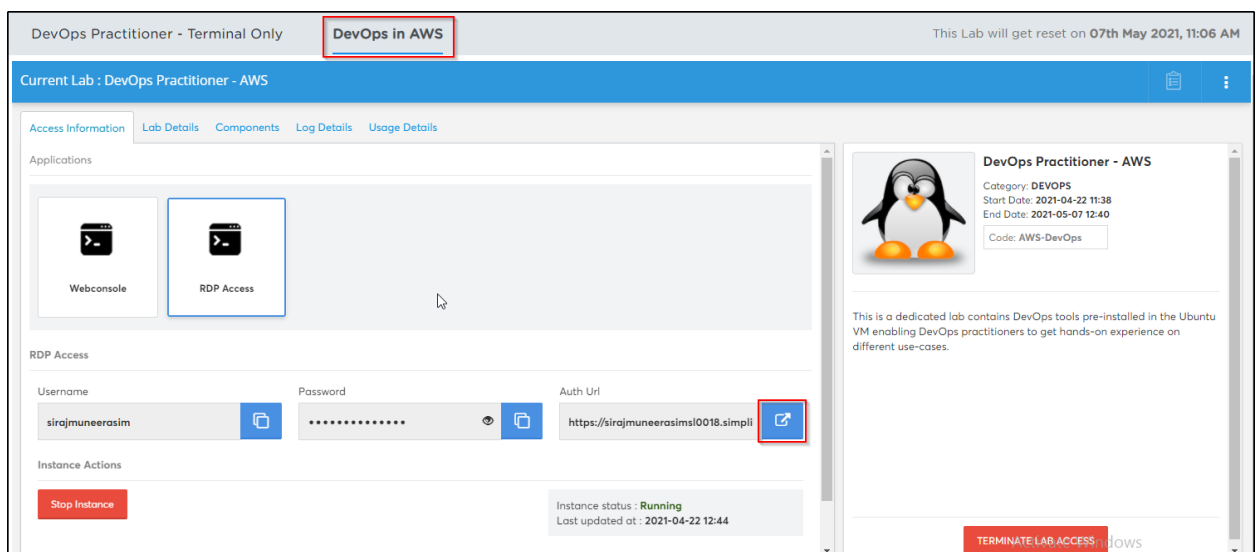


## Step 2: Accessing the terminal lab through VM.

- Click on the Terminal only lab



- Please launch the terminal only lab and save the credentials that are shown in the screenshot above
- Open the DevOps in AWS lab



- Once the lab is launched, open the terminal

- Execute the below commands in the terminal in order to establish an ssh connection with the terminal only lab

***ssh username@ip -p 42006***

- Type *yes* and enter the *password* when prompted.

**Note:** The username, IP and password is the one that you saved initially in step 2.

```
manikumarsimpli@ip-172-31-79-234:~$ ssh manikumarsimpli@3.239.126.172 -p 42006
The authenticity of host '[3.239.126.172]:42006 ([3.239.126.172]:42006)' can't be established.
ECDSA key fingerprint is SHA256:zmLDhSt9rrDGyEDPQ4IhscDqnSq01NIhUmMM/eQQL9I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[3.239.126.172]:42006' (ECDSA) to the list of known hosts.
manikumarsimpli@3.239.126.172's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1124-aws x86_64)

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

 * Pure upstream Kubernetes 1.21, smallest, simplest cluster ops!

    https://microk8s.io/

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

*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

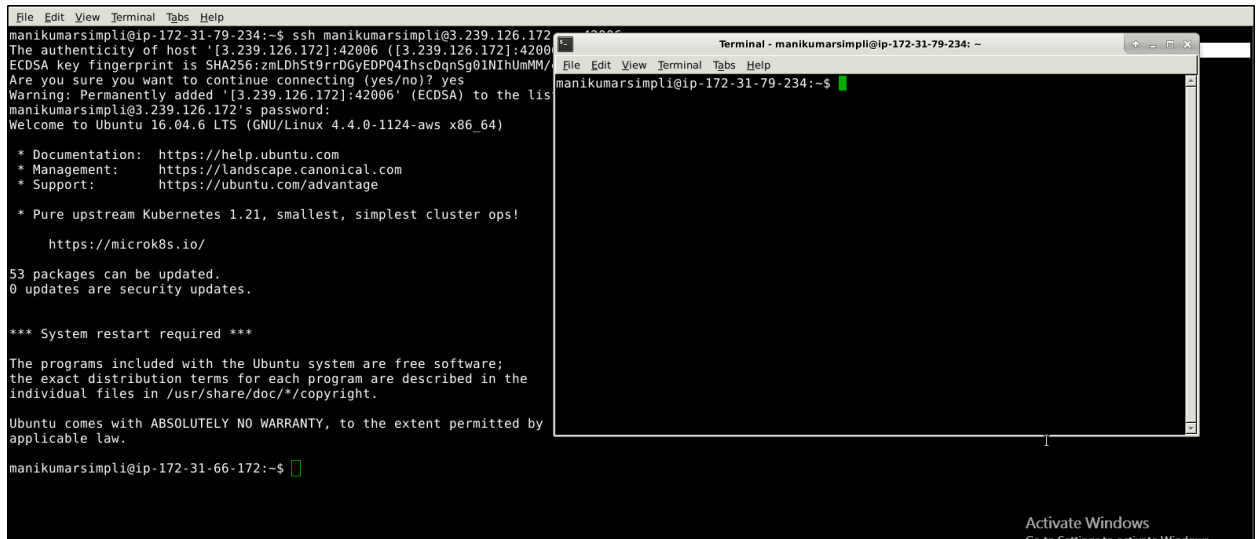
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

manikumarsimpli@ip-172-31-66-172:~$
```

- The terminal only lab has the following tools pre-installed: git, java, maven, nodejs, docker, python, python3.

**Note:** Make sure you do not close the terminal that you used to establish the ssh connection. You might end the ssh connection to the terminal lab on closing the current terminal.

- In case you want to work on the DevOps AWS lab's terminal, you will need to open a new terminal without closing the previous one



### Step 3: Using different IDEs and software required

#### Linux OS:

- The virtual machines that we use in the labs are Linux OS
- To verify the version of the linux installation execute the below command in the terminal:

***lsb\_release -a***

```
manikumarsimpli@ip-172-31-79-234:/$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.6 LTS
Release:        16.04
Codename:       xenial
manikumarsimpli@ip-172-31-79-234:/$
```

#### Java:

- Java 1.8 is already installed in the labs
- Open the terminal and type **java -version** to find whether Java is installed or not

```
manikumarsimpli@ip-172-31-79-234:~$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~16.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
manikumarsimpli@ip-172-31-79-234:~$
```

- If Java is not installed in your system, then use the following commands to install the same:

```
sudo apt-get install openjdk-8-jdk
sudo apt-get install openjdk-8-jre
```

## Git:

- Git is already installed in the labs
- To check whether Git is installed properly or not run the following command:  
***git --version***

```
manikumarsimpli@ip-172-31-79-234:~$ git --version
git version 2.31.1
manikumarsimpli@ip-172-31-79-234:~$
```

- If git is not installed in your system, then use the following commands to install the same:

```
sudo apt-get update
sudo apt install git-all
```

## Maven:



- Maven is already installed in your practice labs
- You can use the following command to verify the installation:

***mvn -v***

```
manikumarsimpli@ip-172-31-79-234:/$ mvn -v
Apache Maven 3.3.9
Maven home: /usr/share/maven
Java version: 1.8.0_282, vendor: Private Build
Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.4.0-1126-aws", arch: "amd64", family: "unix"
manikumarsimpli@ip-172-31-79-234:/$
```

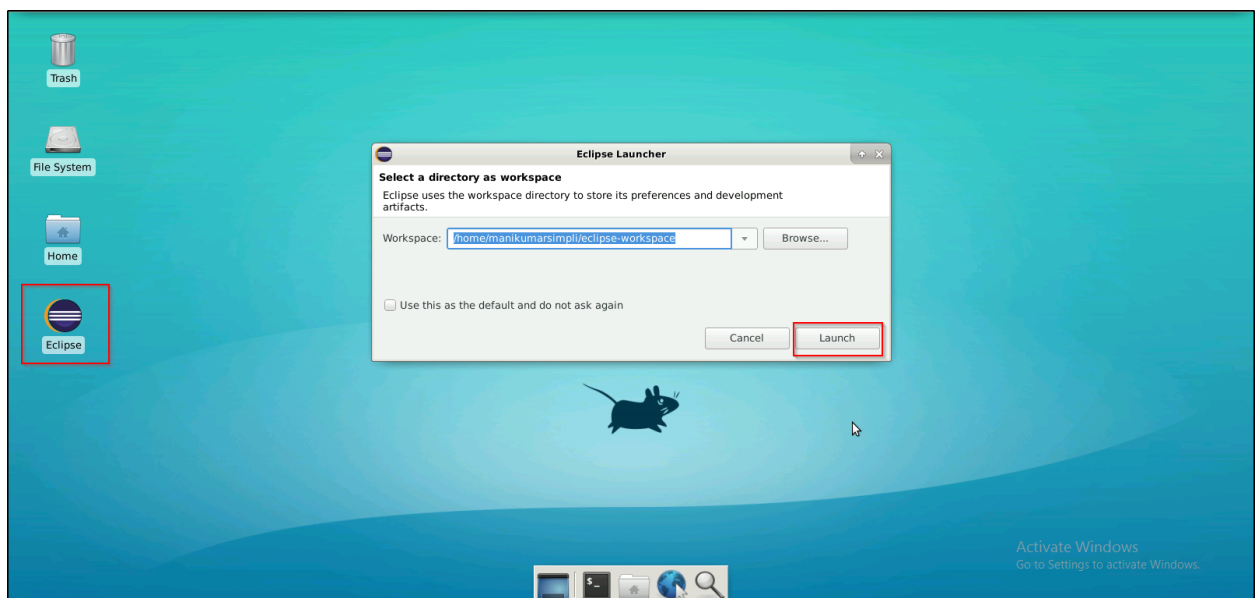
- In case Maven is not installed in your system, you can install it using the commands:

*sudo apt-get update*

*sudo apt-get install maven*

## Eclipse:

- Double-click on the Eclipse icon
- Select a directory where you want to save your programs
- Select the **Use this as the default and do not ask again** checkbox and click on **Launch**

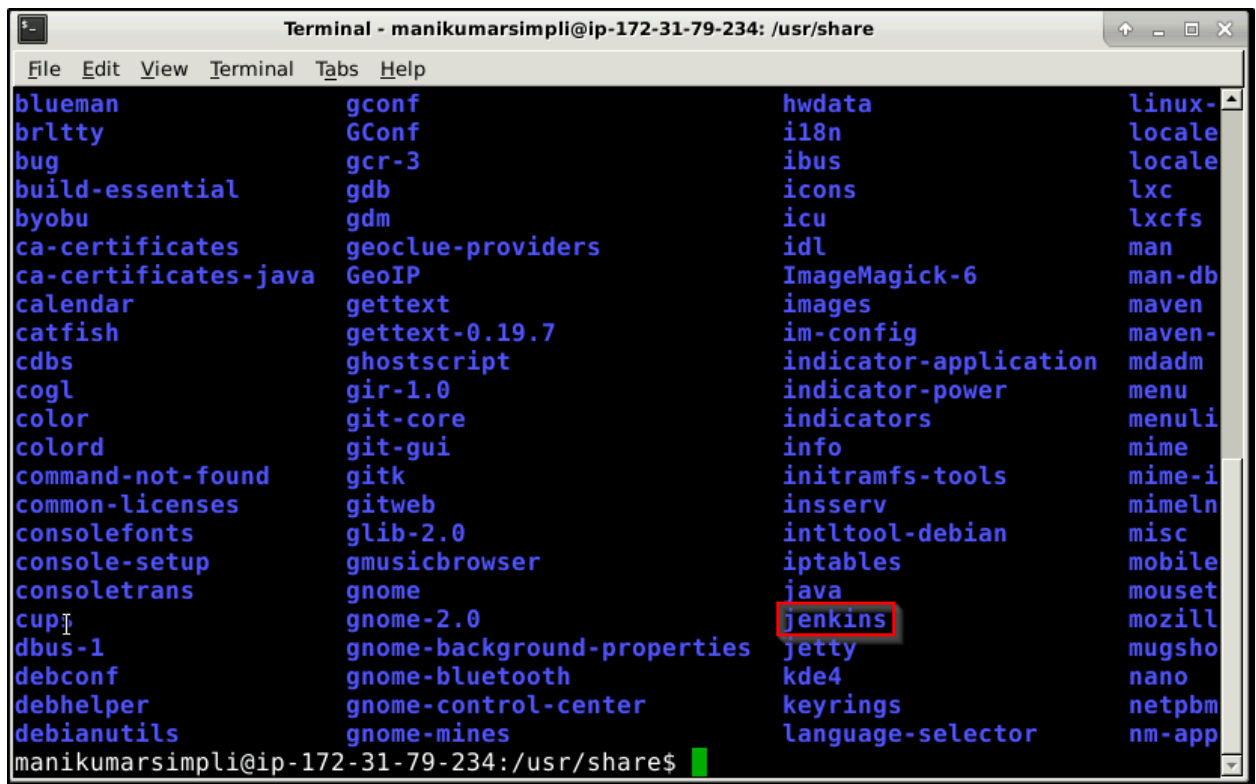


## Jenkins:

- Jenkins (version 2.287) is already installed in your practice lab
- You will find it in the directory `/usr/share`
- Use the following commands to navigate to the above-mentioned directory

**`cd /usr/share`**

**`ls`**



```
Terminal - manikumarsimpli@ip-172-31-79-234: /usr/share
File Edit View Terminal Tabs Help
blueman gconf hwdata linux-
brltty GConf i18n locale
bug gcr-3 ibus locale
build-essential gdb icons lxc
byobu gdm icu lxcfs
ca-certificates geoclue-providers idl man
ca-certificates-java GeoIP ImageMagick-6 man-db
calendar gettext images maven
catfish gettext-0.19.7 im-config maven-
cddb ghostscript indicator-application mdadm
cogl gir-1.0 indicator-power menu
color git-core indicators menuli
colord git-gui info mime
command-not-found gitk initramfs-tools mime-i
common-licenses gitweb insserv mimeln
consolefonts glib-2.0 intltool-debian misc
console-setup gmusicbrowser iptables mobile
consoletrans gnome java mouset
cup jenkins mozilla
dbus-1 gnome-2.0 jetty mugsho
debconf gnome-background-properties kde4 nano
debhelper gnome-bluetooth keyrings netpbm
debianutils gnome-control-center language-selector nm-app
manikumarsimpli@ip-172-31-79-234:/usr/share$
```

- To login to Jenkins dashboard navigate to **`localhost:8080`** from the browser in the lab
- In case Jenkins is not installed in your practice lab, you can install it using the commands:

`sudo apt update`

`sudo apt install jenkins`

## Node JS:

- Node JS 14.16.0 version is installed in your practice labs
- To verify the installation, open the terminal and execute the command:

***node -v***

```
manikumarsimpli@ip-172-31-79-234:/$ node -v
v14.16.0
manikumarsimpli@ip-172-31-79-234:/$
```

- The command mentioned above displays the Node JS version installed in your practice lab
- If Node JS is not installed in your practice lab, you can install it by using the commands:

*sudo apt-get update*

*sudo apt-get install nodejs*

## **Docker:**

- Docker version 18.09.7 is already installed in your practice lab
- To verify the installation, open the terminal and execute the following:

***docker --version***

```
manikumarsimpli@ip-172-31-79-234:/$ docker --version
Docker version 19.03.14, build 5eb3275d40
manikumarsimpli@ip-172-31-79-234:/$
```

- In case Docker is not installed in your practice lab, you can install it following these steps:

1. Set up the Docker repository using the following commands:

*sudo apt-get update*

```
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

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

2. Install Docker using the command:

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

## Ansible:

- Ansible 2.9.19 is already installed in your practice lab
- To verify the installation:
  1. Open the command-line interface
  2. Type the command:

***ansible --version***

```
manikumarsimpli@ip-172-31-79-234:/$ ansible --version
ansible 2.9.19
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/manikumarsimpli/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.12 (default, Mar 1 2021, 11:38:31) [GCC 5.4.0 20160609]
manikumarsimpli@ip-172-31-79-234:/$
```

- If ansible is not installed in your practice lab, you can install it by using the commands:

```
sudo apt-get install -f
```

```
sudo apt-get install software-properties-common
```

```
sudo apt-add-repository ppa:ansible/ansible
```

```
sudo apt-get update
```

---

*sudo apt-get install ansible*

## Python:

- Python is already installed in your practice lab
- To verify the installation, open the command-line interface and type the command:

***python --version***

***python3 --version***

```
manikumarsimpli@ip-172-31-79-234:/$ python --version
Python 2.7.12
manikumarsimpli@ip-172-31-79-234:/$ python3 --version
Python 3.9.4
manikumarsimpli@ip-172-31-79-234:/$ █
```

- If python is not installed in your practice lab, you can install it by using the commands:

*sudo apt-get update*

*sudo apt-get install python*

*sudo apt-get update*

*sudo apt-get install python3*