

DevOps Exercise Book

Contents

Task1: Your first Virtual Machine.....	1
Task 2: Terminal Exploration.....	3
Task 3: Creating a script file	3
Task 4: Configuring the Linux environment	4
Task 5: Using Iptables.....	4
Task 6: Vagrant scripting.....	5
For single VM installation	6
For multiple VM installation	7
Task 7: Repository Management	9
Task 8: Tooling	14
Task 1a – Setting up Jenkins.....	14
Task 2 – Setting up Jira.....	15
Task 4 – Installing Nexus	16
Task 5 – Install Zabbix	18
Docker	19
Creating AWS instance.....	19
Task 1 – Install docker.....	21
Task 2 – Deploy a Jenkins Container	21
Task 3 – Create a dockerfile	22
Task 4 – Create your own linked container	23
Task 5 - Create your own docker-compose file	24

Task1: Your first Virtual Machine

Using AWS EC2 Ubuntu box



Create instance of Ubuntu box according to needs.

Launch instance from Running Instances

Resources

You are using the following Amazon EC2 resources in the EU West (London) region:

- 0 Running Instances
- 0 Dedicated Hosts
- 1 Volumes
- 1 Key Pairs
- 0 Placement Groups

Following instance is present.

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

<div></div>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
<div></div>		i-013c02033d6adde9d	t2.micro	eu-west-2a	<div></div> stopped		None	<div></div>	-

SSH To

username@publicdns; ubuntu@...

Description	Status Checks	Monitoring	Tags
Instance ID	i-013c02033d6adde9d	Public DNS (IPv4)	-
Instance state	stopped	IPv4 Public IP	-
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-10-21.eu-west-2a
Availability zone	eu-west-2a	Private IPs	172.31.10.21
Security groups	default view inbound rules	Secondary private IPs	

Task 2: Terminal Exploration

```
root@ip-172-31-10-21: /home/QACPawan
ubuntu@ip-172-31-10-21:~$ ls
ubuntu@ip-172-31-10-21:~$ whoami
ubuntu
ubuntu@ip-172-31-10-21:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-10-21:~$ cd ..
ubuntu@ip-172-31-10-21:/home$ ls
ubuntu
ubuntu@ip-172-31-10-21:/home$ mkdir QACPawan
mkdir: cannot create directory 'QACPawan': Permission denied
ubuntu@ip-172-31-10-21:/home$ sudo mkdir QACPawan
ubuntu@ip-172-31-10-21:/home$ ls
QACPawan  ubuntu
ubuntu@ip-172-31-10-21:/home$ cd QACPawan/
ubuntu@ip-172-31-10-21:/home/QACPawan$ ls
ubuntu@ip-172-31-10-21:/home/QACPawan$ touch rand.txt
touch: cannot touch 'rand.txt': Permission denied
ubuntu@ip-172-31-10-21:/home/QACPawan$ sudo touch rand.txt
ubuntu@ip-172-31-10-21:/home/QACPawan$ echo "hello devops" >> rand.txt
-bash: rand.txt: Permission denied
ubuntu@ip-172-31-10-21:/home/QACPawan$ sudo echo "hello devops" >> rand.txt
-bash: rand.txt: Permission denied
ubuntu@ip-172-31-10-21:/home/QACPawan$ ls
rand.txt
ubuntu@ip-172-31-10-21:/home/QACPawan$ cat rand.txt
ubuntu@ip-172-31-10-21:/home/QACPawan$ echo "what" >>rand.txt
-bash: rand.txt: Permission denied
ubuntu@ip-172-31-10-21:/home/QACPawan$ user root
No command 'user' found, did you mean:
  Command 'kuser' from package 'kuser' (universe)
  Command 'fuser' from package 'psmisc' (main)
  Command 'users' from package 'coreutils' (main)
  Command 'userv' from package 'userv' (universe)
user: command not found
ubuntu@ip-172-31-10-21:/home/QACPawan$ echo "hello"
hello
ubuntu@ip-172-31-10-21:/home/QACPawan$ echo "hello" >> rand.txt
-bash: rand.txt: Permission denied
ubuntu@ip-172-31-10-21:/home/QACPawan$ sudo bash
root@ip-172-31-10-21:/home/QACPawan# echo "hello">>rand.txt
root@ip-172-31-10-21:/home/QACPawan# cat rand.txt
hello
root@ip-172-31-10-21:/home/QACPawan# ls
```

Task 3: Creating a script file

```
root@ip-172-31-10-21: /home/QACPawan/MyProject
GNU nano 2.5.3      File: adduser.sh
#!/bin/bash
read -p "Username: " username
useradd -g sudo -m $username
passwd $username
echo "Username "+$username+ "created in sudo group."
```

```
GNU nano 2.5.3      File: task1challenge.sh

#!/bin/bash
date
w
uptime
```

Task 4: Configuring the Linux environment

apt-get update

apt-get install maven

(Maven installed itself and java as well)

```
Processing triggers for libc-bin (2.23-0ubuntu7) ...
Processing triggers for systemd (229-4ubuntu16) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for ca-certificates (20160104ubuntu1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
root@ip-172-31-10-21:/home/QACPawan/MyProject# java -version
openjdk version "1.8.0_131"
OpenJDK Runtime Environment (build 1.8.0_131-8u131-b11-0ubuntu1.16.04.2-b11)
OpenJDK 64-Bit Server VM (build 25.131-b11, mixed mode)
root@ip-172-31-10-21:/home/QACPawan/MyProject# mvn -v
Warning: JAVA_HOME environment variable is not set.
Apache Maven 3.3.9
Maven home: /usr/share/maven
Java version: 1.8.0_131, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-openjdk-amd64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.4.0-1013-aws", arch: "amd64", family: "unix"
root@ip-172-31-10-21:/home/QACPawan/MyProject#
```

Task 5: Using Iptables

Start iptables:

sudo ufw enable

Stop iptables:

sudo ufw disable

Show status:

sudo ufw status

```

root@ip-172-31-10-21:/home/QACPawan/MyProject# ufw status
Status: active
root@ip-172-31-10-21:/home/QACPawan/MyProject# ufw disable
Firewall stopped and disabled on system startup
root@ip-172-31-10-21:/home/QACPawan/MyProject# ufw status
Status: inactive
root@ip-172-31-10-21:/home/QACPawan/MyProject# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ip-172-31-10-21:/home/QACPawan/MyProject#

```

Allowing ftp at port 21 rule:

```

root@ip-172-31-10-21:/home/QACPawan/MyProject# iptables -A INPUT -p tcp --dport ftp -j ACCEPT
root@ip-172-31-10-21:/home/QACPawan/MyProject# iptables -L -v -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
 421 34971 ufw-before-logging-input all -- * * 0.0.0.0/0 0.0.0.0/0
 421 34971 ufw-before-input all -- * * 0.0.0.0/0 0.0.0.0/0
 40 2944 ufw-after-input all -- * * 0.0.0.0/0 0.0.0.0/0
 40 2944 ufw-after-logging-input all -- * * 0.0.0.0/0 0.0.0.0/0
 40 2944 ufw-reject-input all -- * * 0.0.0.0/0 0.0.0.0/0
 40 2944 ufw-track-input all -- * * 0.0.0.0/0 0.0.0.0/0
 0 0 ACCEPT     tcp -- * * 0.0.0.0/0 0.0.0.0/0 tcp dpt:21

```

View rule by input chain and by line number, delete by input chain+linenumber

```

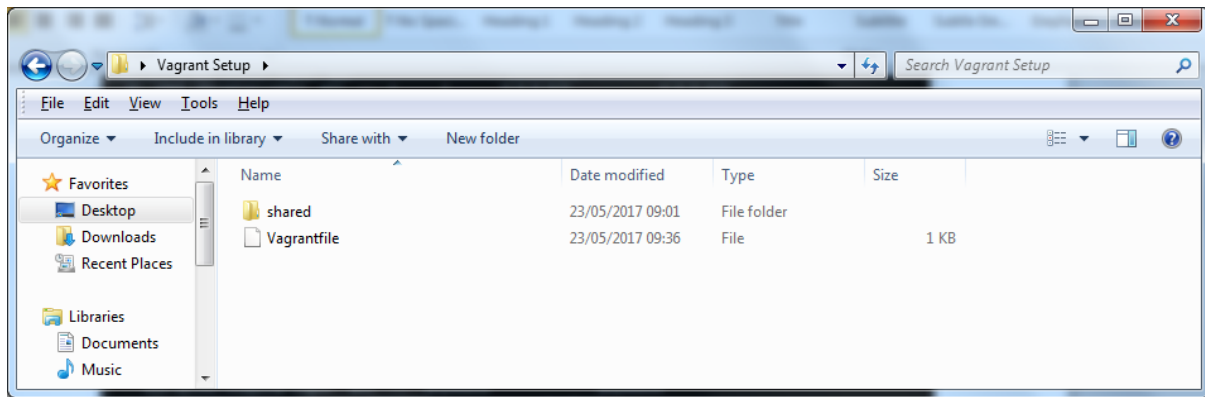
Chain ufw-user-output (1 references)
num target     prot opt source               destination
root@ip-172-31-10-21:/home/QACPawan/MyProject# iptables -L INPUT --line-number
Chain INPUT (policy DROP)
num target     prot opt source               destination
1 ufw-before-logging-input all -- anywhere anywhere
2 ufw-before-input all -- anywhere anywhere
3 ufw-after-input all -- anywhere anywhere
4 ufw-after-logging-input all -- anywhere anywhere
5 ufw-reject-input all -- anywhere anywhere
6 ufw-track-input all -- anywhere anywhere
7 ACCEPT     tcp -- anywhere anywhere tcp dpt:ftp
root@ip-172-31-10-21:/home/QACPawan/MyProject# iptables -D INPUT 7
root@ip-172-31-10-21:/home/QACPawan/MyProject# iptables -L INPUT --line-number
Chain INPUT (policy DROP)
num target     prot opt source               destination
1 ufw-before-logging-input all -- anywhere anywhere
2 ufw-before-input all -- anywhere anywhere
3 ufw-after-input all -- anywhere anywhere
4 ufw-after-logging-input all -- anywhere anywhere
5 ufw-reject-input all -- anywhere anywhere
6 ufw-track-input all -- anywhere anywhere
root@ip-172-31-10-21:/home/QACPawan/MyProject#

```

Task 6: Vagrant scripting

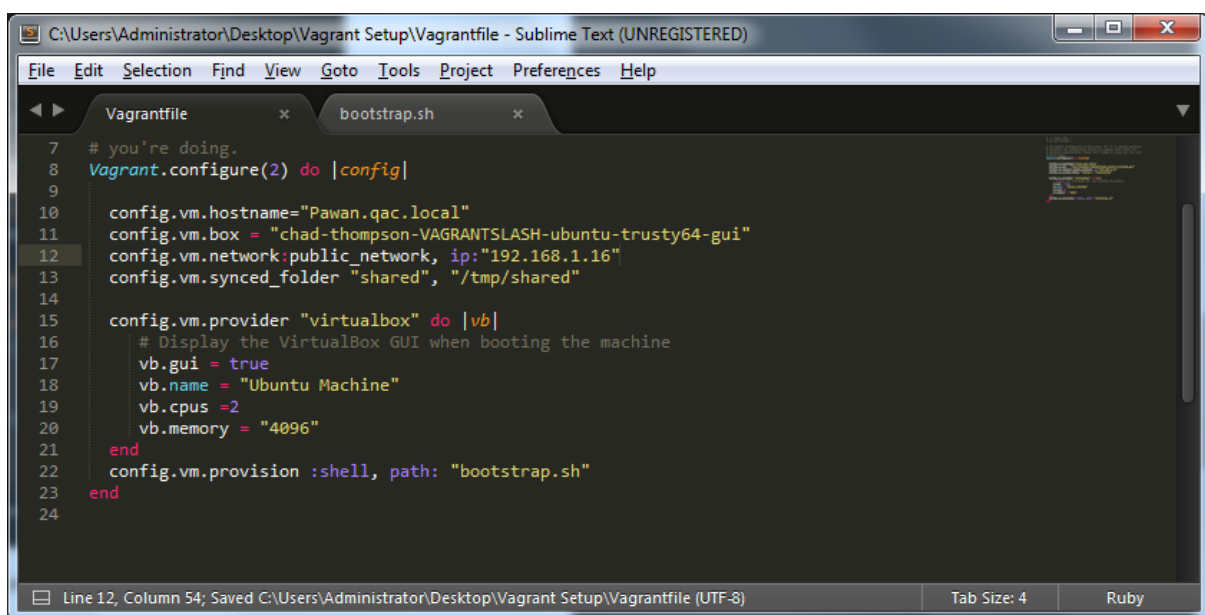
Create directory "Vagrant Setup" - in here open Git bash and use the command, *vagrant init*, to initialise a new vagrant repository. This should add a Vagrantfile to the directory

Create a shared folder to use with VM.

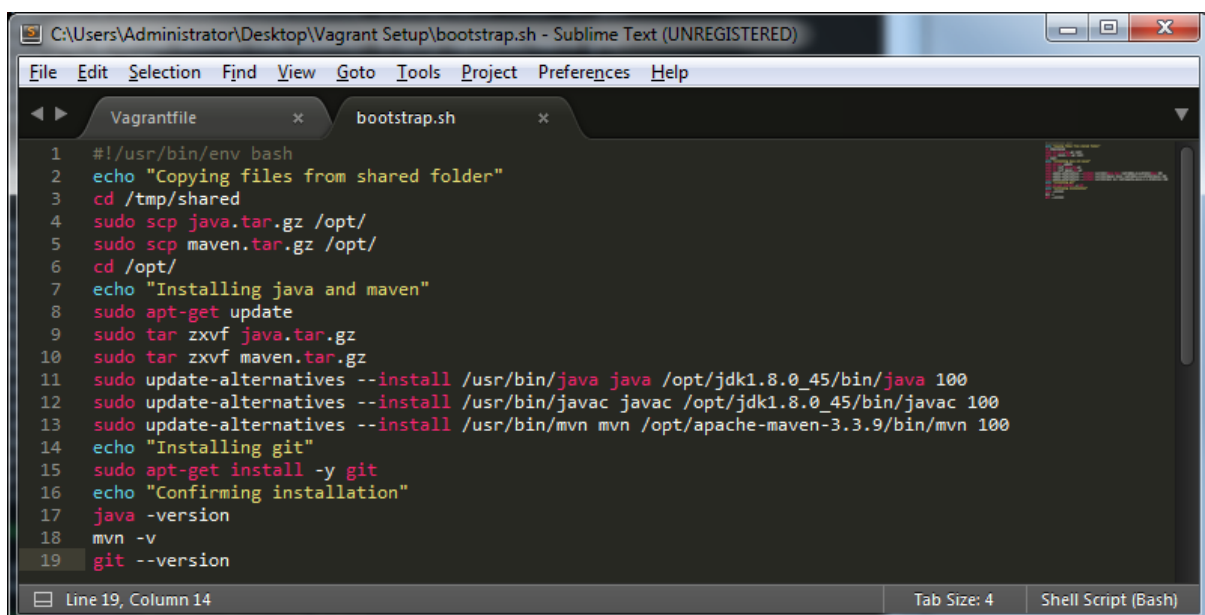


For single VM installation

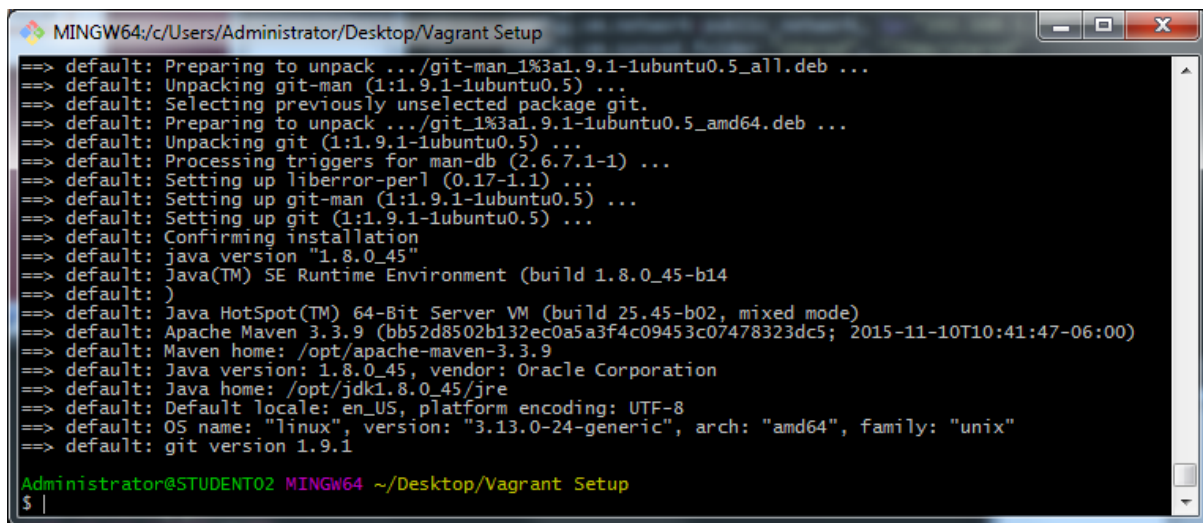
Edited vagrantfile to setup the VM with required specification.



Bash script used to install JAVA, Maven and Git on VM.



Use "vagrant up" on the folder with vagrantfile to create VM and run bash script.



```
Administrator@STUDENT02 MINGW64 ~/Desktop/Vagrant Setup
$ vagrant up
==> default: Preparing to unpack .../git-man_1%3a1.9.1-1ubuntu0.5_all.deb ...
==> default: Unpacking git-man (1:1.9.1-1ubuntu0.5) ...
==> default: Selecting previously unselected package git.
==> default: Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.5_amd64.deb ...
==> default: Unpacking git (1:1.9.1-1ubuntu0.5) ...
==> default: Processing triggers for man-db (2.6.7.1-1) ...
==> default: Setting up liberror-perl (0.17-1.1) ...
==> default: Setting up git-man (1:1.9.1-1ubuntu0.5) ...
==> default: Setting up git (1:1.9.1-1ubuntu0.5) ...
==> default: Confirming installation
==> default: java version "1.8.0_45"
==> default: Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
==> default: )
==> default: Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
==> default: Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T10:41:47-06:00)
==> default: Maven home: /opt/apache-maven-3.3.9
==> default: Java version: 1.8.0_45, vendor: Oracle Corporation
==> default: Java home: /opt/jdk1.8.0_45/jre
==> default: Default locale: en_US, platform encoding: UTF-8
==> default: OS name: "linux", version: "3.13.0-24-generic", arch: "amd64", family: "unix"
==> default: git version 1.9.1
Administrator@STUDENT02 MINGW64 ~/Desktop/Vagrant Setup
$
```

For multiple VM installation

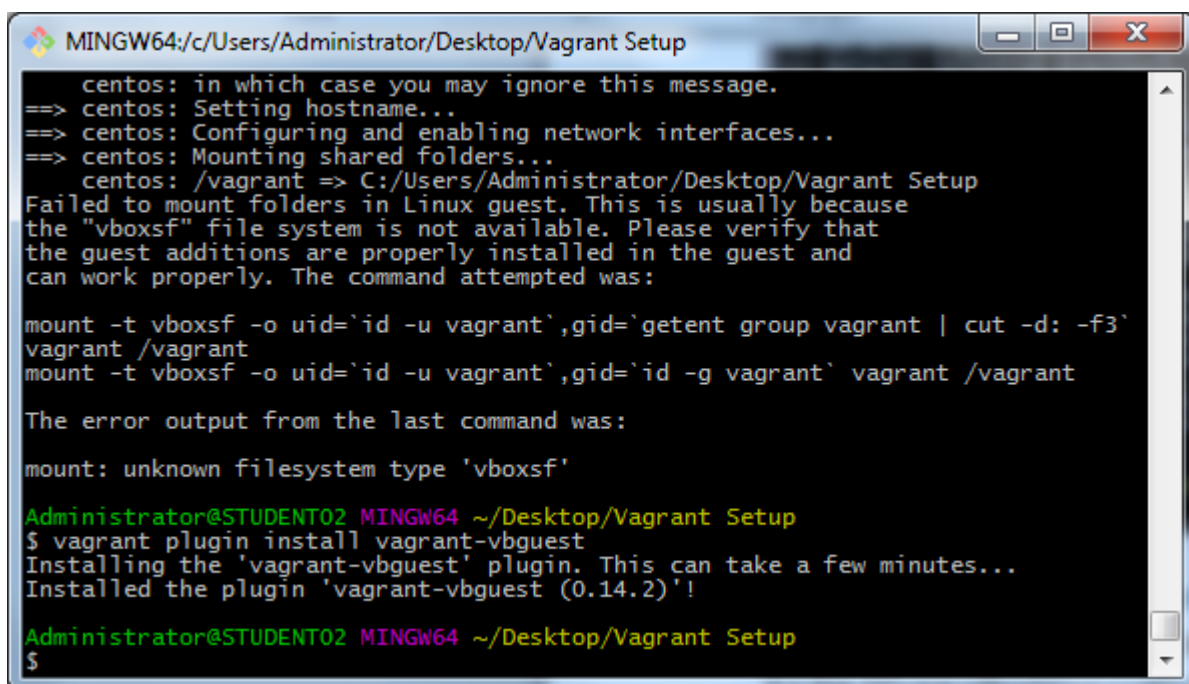
For CentOS installation, edit C:\Users\Administrator\.vagrant.d\boxes\centos-VAGRANTSLASH-7\1704.01\virtualbox\Vagrantfile

```
config.vm.synced_folder ".", "/vagrant", type: "rsync"
```

```
config.vm.synced_folder ".", "/vagrant", type: "virtualbox"
```

Rsync isn't installed.

Run **vagrant plugin install vagrant-vbguest** on git bash as the shared folder encounters problem with CentOS due to mismatch of GuestAdditions between CentOS and virtualbox.



```
Administrator@STUDENT02 MINGW64 ~/Desktop/Vagrant Setup
$ vagrant up
centos: in which case you may ignore this message.
==> centos: Setting hostname...
==> centos: Configuring and enabling network interfaces...
==> centos: Mounting shared folders...
centos: /vagrant => C:/Users/Administrator/Desktop/Vagrant Setup
Failed to mount folders in Linux guest. This is usually because
the "vboxsf" file system is not available. Please verify that
the guest additions are properly installed in the guest and
can work properly. The command attempted was:

mount -t vboxsf -o uid=id -u vagrant,gid=getent group vagrant | cut -d: -f3
vagrant /vagrant
mount -t vboxsf -o uid=id -u vagrant,gid=id -g vagrant vagrant /vagrant

The error output from the last command was:

mount: unknown filesystem type 'vboxsf'
Administrator@STUDENT02 MINGW64 ~/Desktop/Vagrant Setup
$ vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Installed the plugin 'vagrant-vbguest (0.14.2)'!
Administrator@STUDENT02 MINGW64 ~/Desktop/Vagrant Setup
$
```


Vagrantfile for multiple VM

```
8 Vagrant.configure(2) do |config|
9   config.vm.define "ubuntu" do |ubuntu|
10     ubuntu.vm.hostname="Pawan.qac.local"
11     ubuntu.vm.box = "chad-thompson-VAGRANTSLASH-ubuntu-trusty64-gui"
12     ubuntu.vm.network:public_network, ip:"192.168.1.17"
13     ubuntu.vm.synced_folder "shared", "/tmp/shared"
14
15     ubuntu.vm.provider "virtualbox" do |vb|
16       # Display the VirtualBox GUI when booting the machine
17       vb.gui = true
18       vb.name = "Ubuntu Machine"
19       vb.cpus = 2
20       vb.memory = "4096"
21     end
22   end
23   config.vm.define "centos" do |centos|
24     centos.vm.hostname="Pawan1.qac.local"
25     centos.vm.box = "centos-VAGRANTSLASH-7"
26     centos.vm.network:public_network
27     centos.vm.synced_folder "shared", "/tmp/shared"
28
29     centos.vm.provider "virtualbox" do |vb|
30       # Display the VirtualBox GUI when booting the machine
31       vb.gui = true
32       vb.name = "CentOS"
33       vb.cpus = 2
34       vb.memory = "4096"
35     end
36   end
37   config.vm.provision :shell, path: "bootstrap.sh"
38 end
```

Line 38, Column 4; Saved C:\Users\Administrator\Desktop\Vagrant Setup\Vagrantfile (UTF-8) Tab Size: 4 Ruby

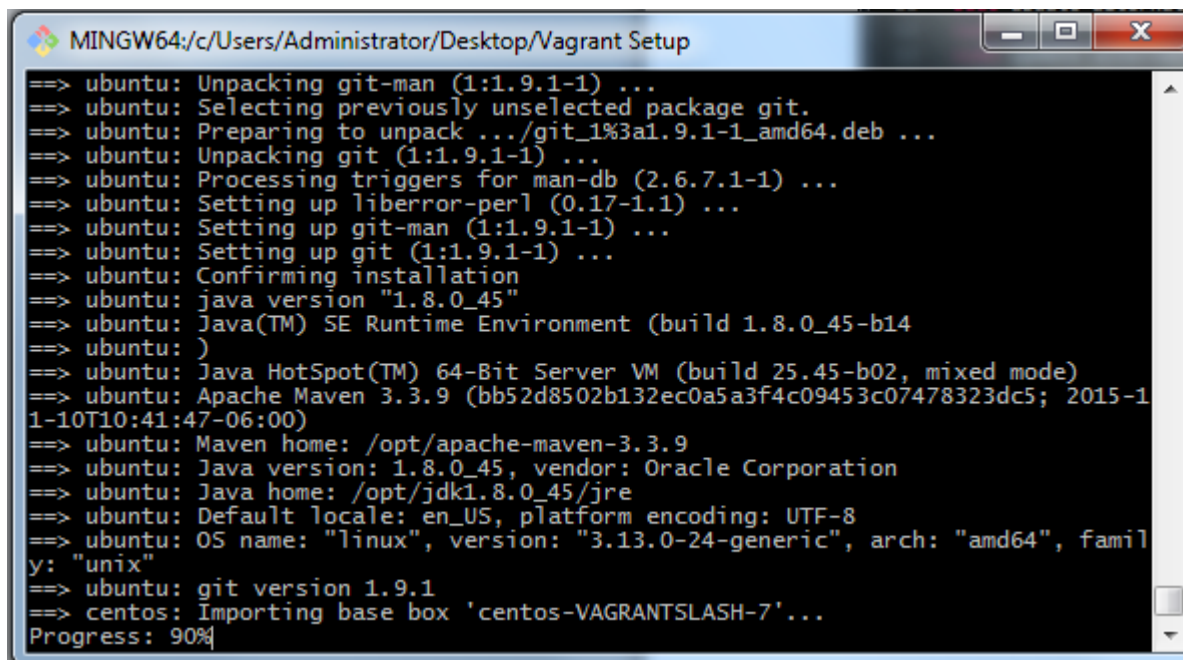
Script file for multiple VM

```
C:\Users\Administrator\Desktop\Vagrant Setup\bootstrap.sh - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Vagrantfile — Desktop\Vagrant Setup x Vagrantfile — .vagrant.d\...virtualbox x bootstrap.sh x
1 #!/usr/bin/env bash
2 echo "Copying files from shared folder"
3 cd /tmp/shared
4 sudo scp java.tar.gz /opt/
5 sudo scp maven.tar.gz /opt/
6 cd /opt/
7 echo "Installing java and maven"
8 if [ "$(. /etc/os-release; echo $NAME)" = "Ubuntu" ]; then
9   sudo apt-get install updates
10 else
11   sudo yum update
12 fi
13 sudo tar zxvf java.tar.gz
14 sudo tar zxvf maven.tar.gz
15 sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_45/bin/java 100
16 sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_45/bin/javac 100
17 sudo update-alternatives --install /usr/bin/mvn mvn /opt/apache-maven-3.3.9/bin/mvn 100
18 echo "Installing git"
19 if [ "$(. /etc/os-release; echo $NAME)" = "Ubuntu" ]; then
20   sudo apt-get install -y git
21 else
22   sudo yum install -y git
23 fi
24 echo "Confirming installation"
25 java -version
26 mvn -v
27 git --version
```

Line 22, Column 26 Tab Size: 4 Shell Script (Bash)

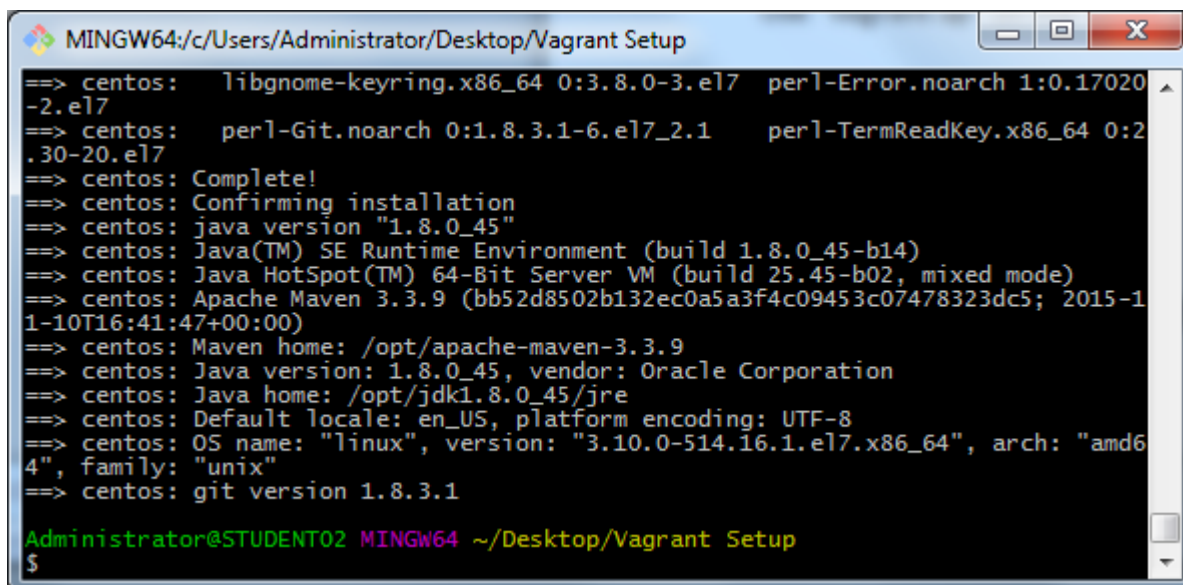
Use “vagrant up” on the folder with vagrantfile to create VM and run bash script.

Ubuntu installed with JAVA, MAVEN, and GIT.



```
MINGW64:/c/Users/Administrator/Desktop/Vagrant Setup
==> ubuntu: Unpacking git-man (1:1.9.1-1) ...
==> ubuntu: Selecting previously unselected package git.
==> ubuntu: Preparing to unpack ../git_1%3a1.9.1-1_amd64.deb ...
==> ubuntu: Unpacking git (1:1.9.1-1) ...
==> ubuntu: Processing triggers for man-db (2.6.7.1-1) ...
==> ubuntu: Setting up liberror-perl (0.17-1.1) ...
==> ubuntu: Setting up git-man (1:1.9.1-1) ...
==> ubuntu: Setting up git (1:1.9.1-1) ...
==> ubuntu: Confirming installation
==> ubuntu: java version "1.8.0_45"
==> ubuntu: Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
==> ubuntu: )
==> ubuntu: Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
==> ubuntu: Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T10:41:47-06:00)
==> ubuntu: Maven home: /opt/apache-maven-3.3.9
==> ubuntu: Java version: 1.8.0_45, vendor: Oracle Corporation
==> ubuntu: Java home: /opt/jdk1.8.0_45/jre
==> ubuntu: Default locale: en_US, platform encoding: UTF-8
==> ubuntu: OS name: "linux", version: "3.13.0-24-generic", arch: "amd64", family: "unix"
==> ubuntu: git version 1.9.1
==> centos: Importing base box 'centos-VAGRANTSLASH-7'...
Progress: 90%
```

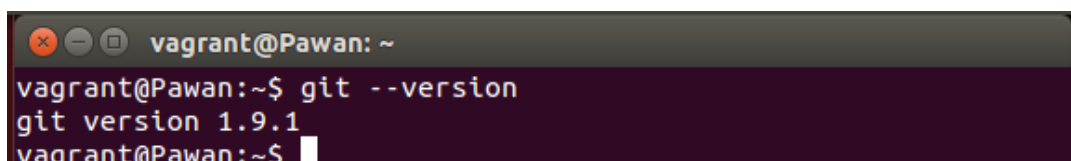
CentOS installed with JAVA, MAVEN, and GIT.



```
MINGW64:/c/Users/Administrator/Desktop/Vagrant Setup
==> centos: libgnome-keyring.x86_64 0:3.8.0-3.e17 perl-Error.noarch 1:0.17020-2.e17
==> centos: perl-Git.noarch 0:1.8.3.1-6.e17_2.1 perl-TermReadKey.x86_64 0:2.30-20.e17
==> centos: Complete!
==> centos: Confirming installation
==> centos: java version "1.8.0_45"
==> centos: Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
==> centos: Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
==> centos: Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T16:41:47+00:00)
==> centos: Maven home: /opt/apache-maven-3.3.9
==> centos: Java version: 1.8.0_45, vendor: Oracle Corporation
==> centos: Java home: /opt/jdk1.8.0_45/jre
==> centos: Default locale: en_US, platform encoding: UTF-8
==> centos: OS name: "linux", version: "3.10.0-514.16.1.e17.x86_64", arch: "amd64", family: "unix"
==> centos: git version 1.8.3.1
Administrator@STUDENT02 MINGW64 ~/Desktop/Vagrant Setup
$
```

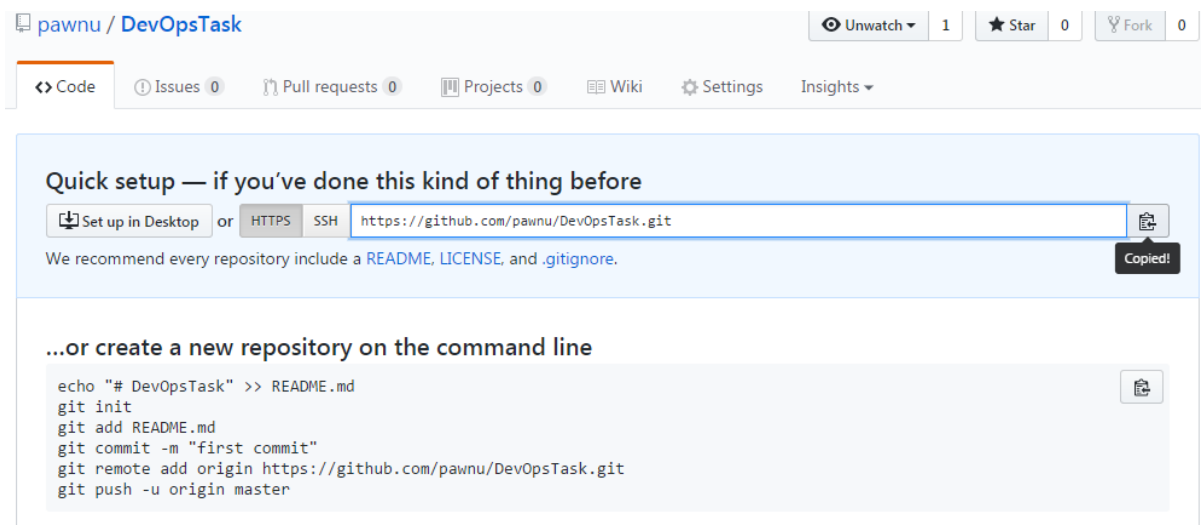
Task 7: Repository Management

Confirmed Git is installed in VM

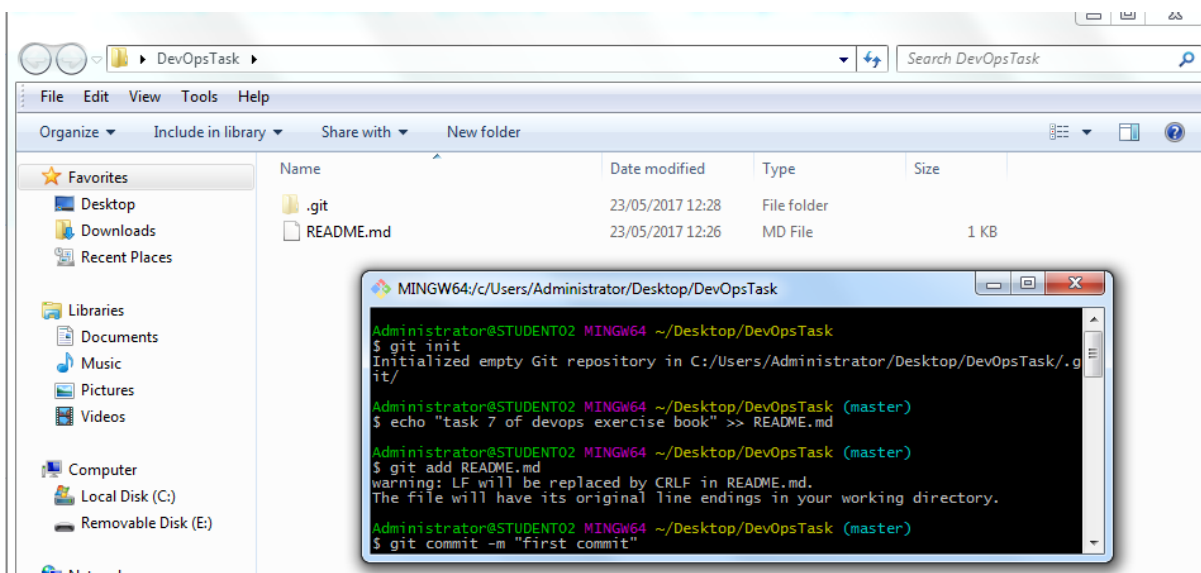


```
vagrant@Pawan: ~
vagrant@Pawan:~$ git --version
git version 1.9.1
vagrant@Pawan:~$
```

On Windows host machine, create a repository on github.



On host windows machine, create a directory for git and use "git init" command on that folder. Create a file, add and commit it.



Add the commit to your repository on github with HTTPS link and push the changes.

```
MINGW64:/c/Users/Administrator/Desktop/DevOpsTask
Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git commit -m "first commit"
[master (root-commit) c4c5727] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git remote add origin https://github.com/pawnu/DevOpsTask.git

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git push -u origin master
Username for 'https://github.com': pawan.uppadey@gmail.com
Counting objects: 3, done.
Writing objects: 100% (3/3), 243 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pawnu/DevOpsTask.git
* [new branch]      master -> master
Branch master set up to track remote branch master from origin.

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ |
```

On Ubuntu guest machine, clone and confirm the file created earlier is present.

```
vagrant@Pawan: ~/Desktop/DevOps/DevOpsTask
vagrant@Pawan:~$ ls
Desktop  Downloads  Music  Public  Videos
Documents  examples.desktop  Pictures  Templates
vagrant@Pawan:~$ cd Desktop/
vagrant@Pawan:~/Desktop$ ls
vagrant@Pawan:~/Desktop$ mkdir DevOps
vagrant@Pawan:~/Desktop$ cd DevOps/
vagrant@Pawan:~/Desktop/DevOps$ ls
vagrant@Pawan:~/Desktop/DevOps$ git clone https://github.com/pawnu/DevOpsTask.git
Cloning into 'DevOpsTask'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
vagrant@Pawan:~/Desktop/DevOps$ ls
DevOpsTask
vagrant@Pawan:~/Desktop/DevOps$ cd DevOpsTask/
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ ls
README.md
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ cat README.md
task 7 of devops exercise book
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$
```

Create a new file on Windows host machine and push change to repository.

```
MINGW64/c/Users/Administrator/Desktop/DevOpsTask

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git add example.txt
warning: LF will be replaced by CRLF in example.txt.
The file will have its original line endings in your working directory.

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git commit "added example file"
error: pathspec 'added example file' did not match any file(s) known to git.

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ ls
example.txt  README.md

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git commit -m "added example file"
[master d1e073a] added example file
1 file changed, 1 insertion(+)
create mode 100644 example.txt

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git remote -v
origin  https://github.com/pawnu/DevOpsTask.git (fetch)
origin  https://github.com/pawnu/DevOpsTask.git (push)

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git push origin master
Username for 'https://github.com': pawan.uppadey@gmail.com
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 290 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pawnu/DevOpsTask.git
 c4c5727..d1e073a  master -> master

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$
```

Confirm the file present on Ubuntu guest machine.

```
vagrant@Pawan: ~/Desktop/DevOps/DevOpsTask

vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ ls
README.md
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ cat README.md
task 7 of devops exercise book
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ git pull origin mast
er
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/pawnu/DevOpsTask
 * branch          master      -> FETCH_HEAD
   c4c5727..d1e073a  master      -> origin/master
Updating c4c5727..d1e073a
Fast-forward
 example.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 example.txt
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ ls
example.txt  README.md
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ cat example.txt
new file
```

Make change to a file on Ubuntu and push to repository

```
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ git add example.txt
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$ git commit -m "made
change to example text file"
[master 179bba0] made change to example text file
  Committer: Vagrant <vagrant@Pawan.qac.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accur
ate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

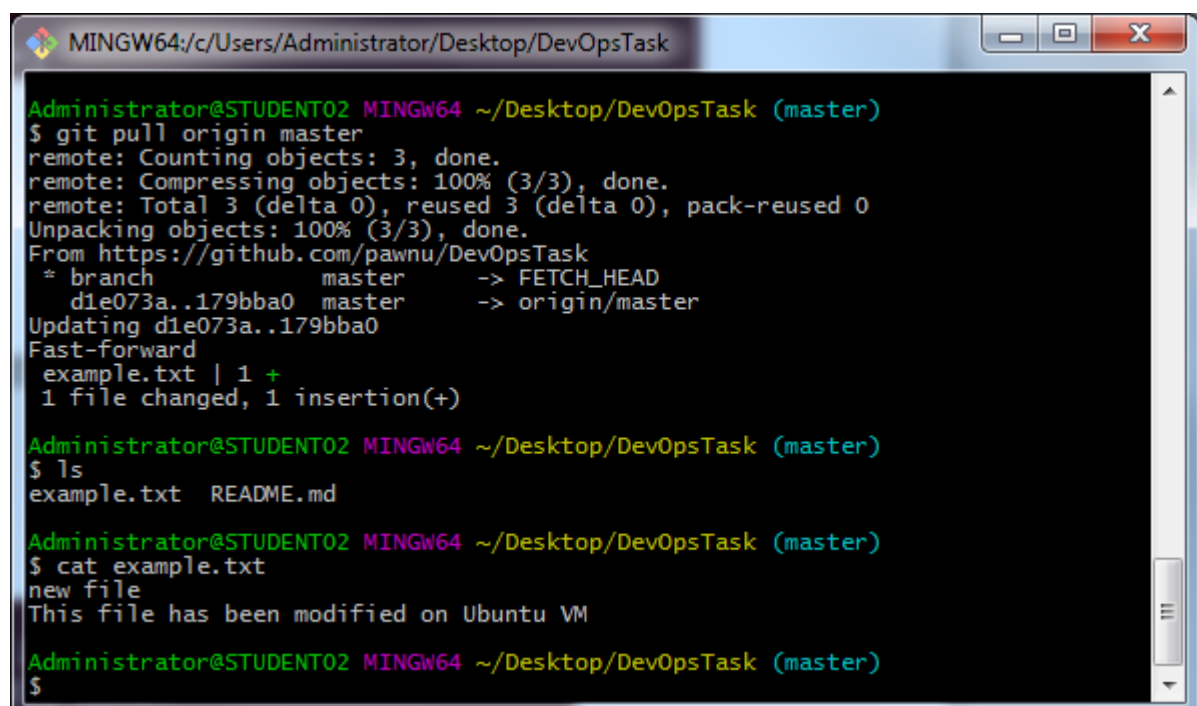
After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
```

```
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/pawnu/DevOpsTask.git
   d1e073a..179bba0  master -> master
vagrant@Pawan:~/Desktop/DevOps/DevOpsTask$
```

Confirm changes on Windows host machine side



```
MINGW64:/c:/Users/Administrator/Desktop/DevOpsTask
Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/pawnu/DevOpsTask
 * branch            master       -> FETCH_HEAD
   d1e073a..179bba0  master       -> origin/master
Updating d1e073a..179bba0
Fast-forward
 example.txt | 1 +
 1 file changed, 1 insertion(+)

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ ls
example.txt  README.md

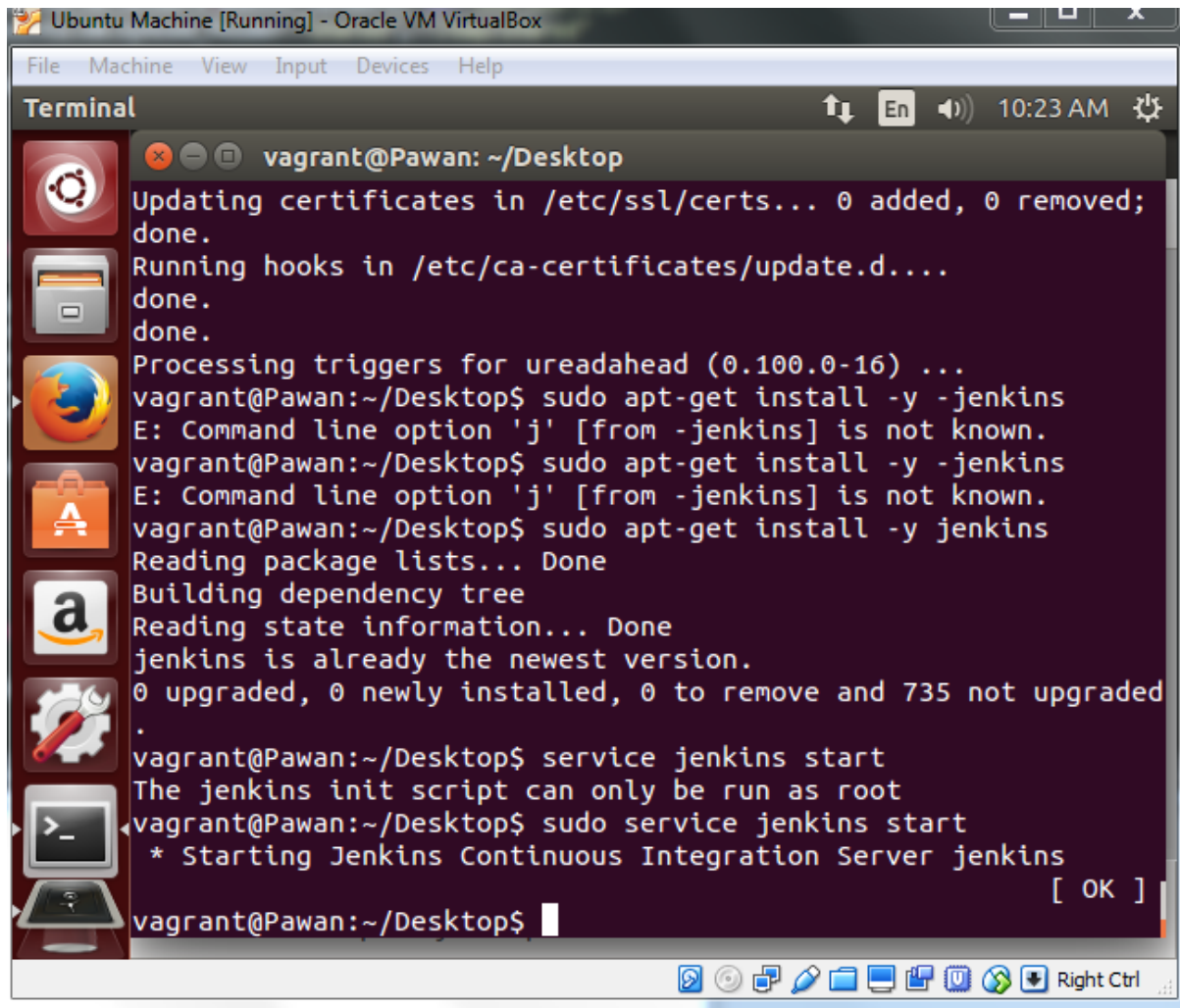
Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$ cat example.txt
new file
This file has been modified on Ubuntu VM

Administrator@STUDENT02 MINGW64 ~/Desktop/DevOpsTask (master)
$
```


Task 8: Tooling

Task 1a – Setting up Jenkins

```
sudo scp jenkins_2.1_all.deb /home/vagrant/Desktop/  
cd /home/vagrant/Desktop  
sudo dpkg -i jenkins_2.1_all.deb  
sudo apt-get install -y -f  
sudo apt-get install -y jenkins  
sudo service jenkins start
```



```
Ubuntu Machine [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Terminal 10:23 AM  
vagrant@Pawan: ~/Desktop  
Updating certificates in /etc/ssl/certs... 0 added, 0 removed;  
done.  
Running hooks in /etc/ca-certificates/update.d....  
done.  
done.  
Processing triggers for ureadahead (0.100.0-16) ...  
vagrant@Pawan:~/Desktop$ sudo apt-get install -y -jenkins  
E: Command line option 'j' [from -jenkins] is not known.  
vagrant@Pawan:~/Desktop$ sudo apt-get install -y -jenkins  
E: Command line option 'j' [from -jenkins] is not known.  
vagrant@Pawan:~/Desktop$ sudo apt-get install -y jenkins  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
jenkins is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 735 not upgraded  
.  
vagrant@Pawan:~/Desktop$ service jenkins start  
The jenkins init script can only be run as root  
vagrant@Pawan:~/Desktop$ sudo service jenkins start  
* Starting Jenkins Continuous Integration Server jenkins  
[ OK ]  
vagrant@Pawan:~/Desktop$
```



Task 2 – Setting up Jira

cd /opt/

sudo chmod a+x jira.bin

sudo ./jira.bin

Choose custom install, install on port 8081, custom port 80

Please wait a few moments while JIRA starts up.

Launching JIRA ...

Installation of JIRA 6.4.9 is complete

Your installation of JIRA 6.4.9 is now ready and can be accessed via your browser.

JIRA 6.4.9 can be accessed at http://localhost:8081

Finishing installation ...

vagrant@Pawan:/opt\$



Task 4 – Installing Nexus

```
cd /opt/  
sudo scp nexus-2.14.4-03-bundle.tar.gz /usr/local  
cd /usr/local  
sudo tar xvfz nexus-2.14.4-03-bundle.tar.gz  
sudo ln -s nexus-2.14.4-03 nexus  
echo "1" | sudo update-alternatives --config java  
sudo chown -R vagrant nexus* sonatype-work  
cd /usr/local/nexus  
./bin/nexus console  
./bin/nexus start
```

Notes:

nexus-3.* didn't work with this setup
nexus won't run with sudo, change a user to be owner of nexus and sonawork directory to run
nexus requires JVM 1.8. JVM might show as 1.7. Make sure java 1.8 is installed.

Use **sudo update-alternatives - --config java** and select the 1.8 version of java to run nexus

```

vagrant@Pawan: /opt
vagrant@Pawan:/usr/local/nexus$ sudo update-alternatives --conf
ig java
There are 2 choices for the alternative java (providing /usr/bi
n/java).

  Selection    Path
Priority    Status
-----
* 0            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java
1071         auto mode
  1            /opt/jdk1.8.0_45/bin/java
  1            manual mode
  2            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java
1071         manual mode

Press enter to keep the current choice[*], or type selection nu
mber: 1
update-alternatives: using /opt/jdk1.8.0_45/bin/java to provide
 /usr/bin/java (java) in manual mode
vagrant@Pawan:/usr/local/nexus$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

```

Nexus by default runs on port 8081, to change it edit the **conf/properties** file

```

# Sonatype Nexus
# =====
# This is the most basic configuration of Nexus.

# Jetty section
application-port=8085
application-host=0.0.0.0
nexus-webapp=${bundleBasedir}/nexus
nexus-webapp-context-path=/nexus

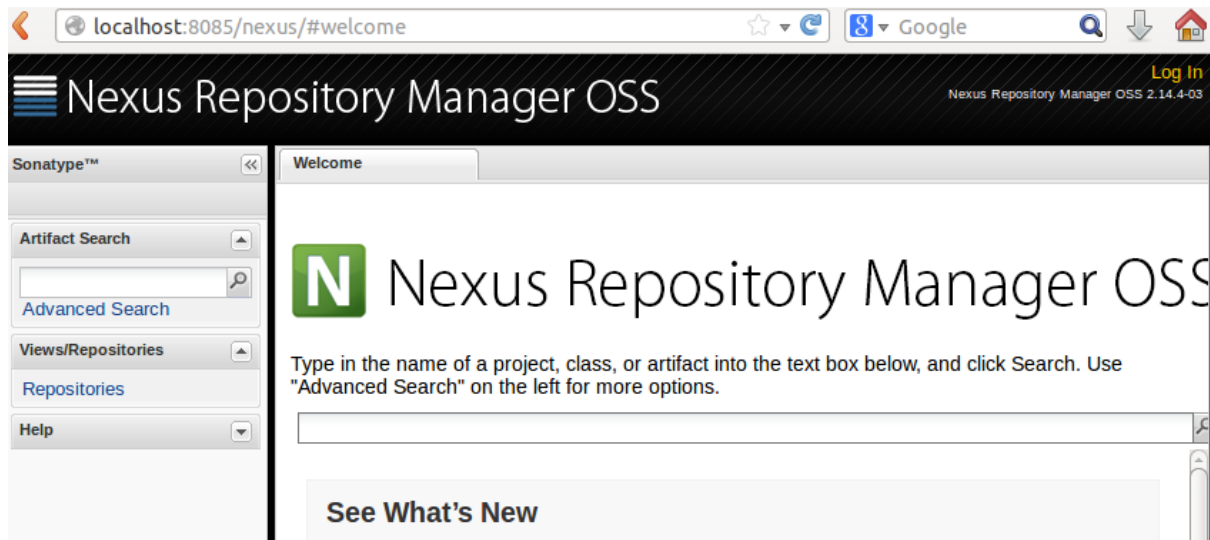
```

Run nexus as follows:

```

vagrant@PUPPADEY:/usr/local/nexus$ ./bin/nexus stop
Stopping Nexus OSS...
Nexus OSS was not running.
vagrant@PUPPADEY:/usr/local/nexus$ ./bin/nexus start
Starting Nexus OSS...
Started Nexus OSS.
vagrant@PUPPADEY:/usr/local/nexus$ ./bin/nexus status
Nexus OSS is running (18921).

```



Task 5 – Install Zabbix

```
cd /opt/
```

```
wget http://repo.zabbix.com/zabbix/2.4/ubuntu/pool/main/z/zabbix-release/zabbix-release_2.4-1+trusty_all.deb
```

```
sudo dpkg -i zabbix-release_2.4-1+trusty_all.deb
```

```
sudo apt-get install -y zabbix-server-mysql zabbix-frontend-php php5-mysql
```

Edit **/etc/php5/apache2/php.ini**

```
post_max_size = 16M
```

```
max_execution_time = 300
```

```
max_input_time = 300
```

```
date.timezone = Europe/London
```

```
sudo service apache2 restart
```

Create **/etc/zabbix/apache.conf**

```
# Define /zabbix alias, this is the default
```

```
<IfModule mod_alias.c>
```

```
Alias /zabbix /usr/share/zabbix
```

```
</IfModule>
```

```
sudo cp /etc/zabbix/apache.conf /etc/apache2/conf-available/zabbix.conf
```

```
sudo a2enconf zabbix.conf
```

```
sudo service apache2 restart
```

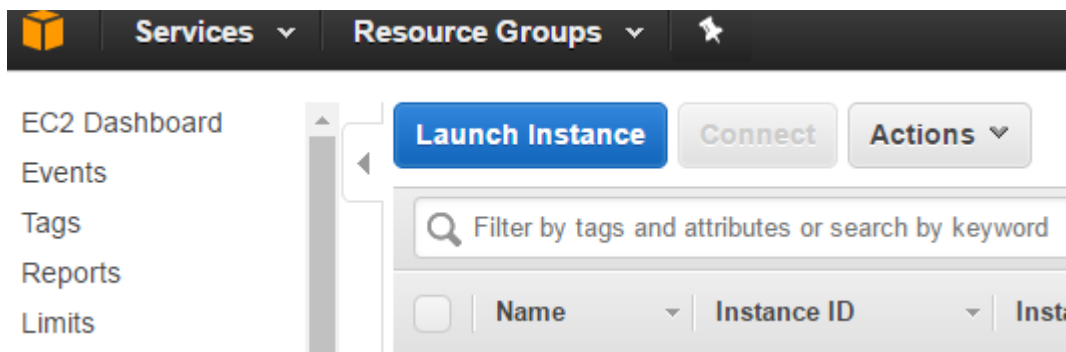
```
sudo service zabbix-server start
```



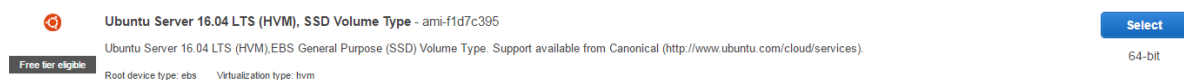
Docker

Creating AWS instance

EC2 – Launch Instance



Install Ubuntu



Go to security groups – default to configure firewall

Public DNS (IPv4)	IPv4 Public IP	IPv6	Key Name	Monitoring	Launch Time	Security Groups
-	-	-	Keypair2	disabled	May 24, 2017 at 3:44:30 PM...	default
-	-	-	dockerpair	disabled	May 25, 2017 at 10:34:30 AM..	default
ec2-35-176-57-21...	35.176.57.212	-	dockerpair	disabled	May 25, 2017 at 10:52:25 AM..	default

Security Group: sg-fd128894

Description

Inbound

Outbound

Tags

Edit

Type ⓘ

Protocol ⓘ

Custom TCP Rule

TCP

Custom TCP Rule

TCP

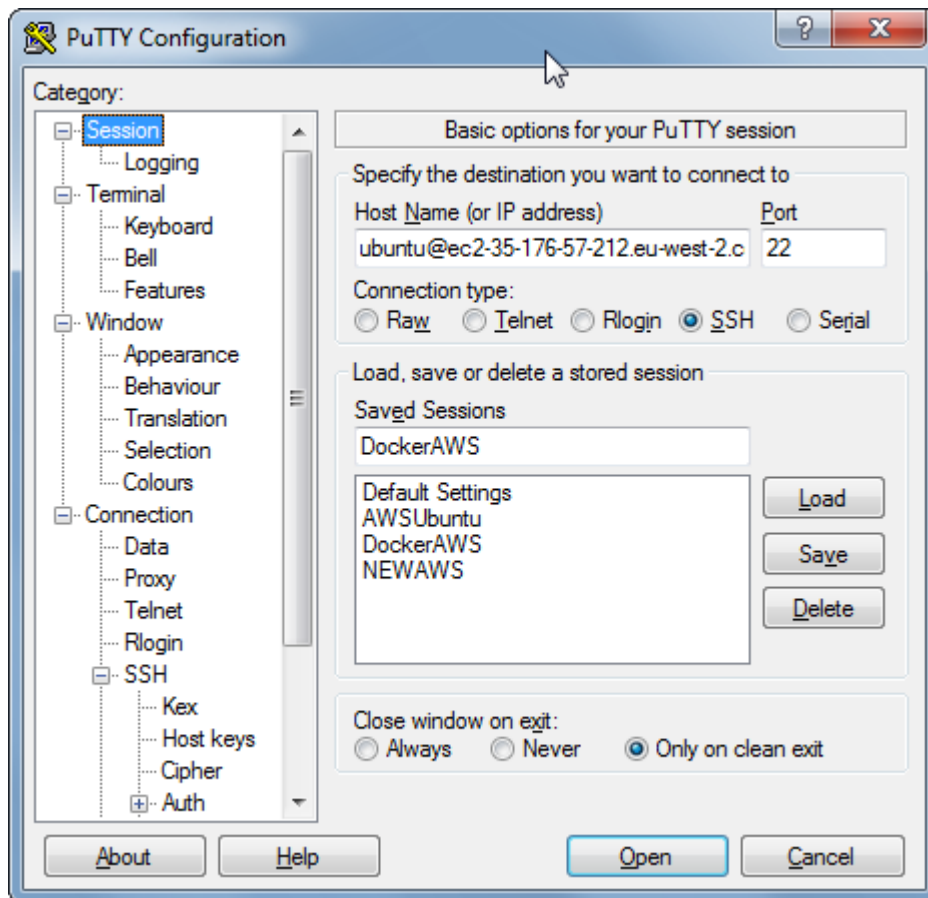
Add port 22 for SSH, and others as required e.g. 8080

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	
Custom TCP Rule	TCP	8080	Custom 0.0.0.0/0	✕
Custom TCP Rule	TCP	8080	Custom ::/0	✕
All traffic	All	0 - 65535	Custom sg-fd128894	✕
SSH	TCP	22	Custom 0.0.0.0/0	✕
SSH	TCP	22	Custom ::/0	✕
Custom TCP Rule	TCP	8081	Custom 0.0.0.0/0	✕
Custom TCP Rule	TCP	8081	Custom ::/0	✕
Add Rule				

Public DNS (IPv4)	ec2-35-176-57-212.eu-west-2.compute.amazonaws.com	Copy	Ctrl+C
IPv4 Public IP	35.176.57.212	Go to ec2-35-176-57-212.eu-west-2.compute.amazonaws.com	
IPv6 IPs	-	Print...	Ctrl+P
Private DNS	ip-172-31-71-100.eu-west-2.compute.amazonaws.com	Inspect	Ctrl+Shift+I
Private IPs	172.31.71.100		
Secondary private IPs			
VPC ID	vpc-0154e368		
Subnet ID	subnet-aeaf74d5		
Network interfaces	eth0		
Source/dest. check	True		

The .pem file provided by AWS to access SSH can be broken down into a private/public key pair. The private key can be created by using the .pem file on PuTTYgen – click generate private key.

Put your privatekey to SSH-Auth on PuTTY Config. Insert the username (Ubuntu) public IP address and port 22 to access SSH.



Task 1 – Install docker

Get the script file to install docker and its dependencies from docker website and pipe it to the shell.

```
wget -qO- https://get.docker.com/ | sh
```

Task 2 – Deploy a Jenkins Container

```
docker pull jenkins
```

```
docker run -p 8080:8080 -p 50000:50000 jenkins
```



A container of Jenkins has been created, use ctrl+z to get out of the terminal.

sudo docker ps -a

Lists all processes, find the container id of Jenkins

sudo docker start containerID

```
ubuntu@ip-172-31-7-183:~$ sudo docker start 3ee2780a2a10
3ee2780a2a10
ubuntu@ip-172-31-7-183:~$
```

```
ubuntu@ip-172-31-7-183:~$ sudo docker ps -a -f status=running
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
35f39df042c9	mysql:8.0	"docker-entrypoint..."	17 hours ago	Up 11 minutes
3ee2780a2a10	jenkins	"/bin/tini -- /usr..."	22 hours ago	Up 10 minutes

```
ubuntu@ip-172-31-7-183:~$
```

Task 3 – Create a dockerfile

FROM ubuntu:16.04

#Always update your running system

RUN sudo apt-get update -y

#You may or may not need to run these commands

RUN sudo apt-get install -y wget

RUN sudo apt-get install -y tar

#installs the libraries needed to run the GUI

RUN sudo apt-get install -y libgtk2.0

RUN sudo apt-get install -y mesa-utils

RUN sudo apt-get install -y libXtst6

#RUN sudo apt-get install -y openjdk-7-jre

#RUN java -version

#Now install the Java Compiler

#RUN sudo apt-get install -y openjdk-7-jdk

#RUN javac -version

#Add java from file and install

WORKDIR /opt

ADD files /opt

RUN sudo tar xzvf /opt/java.tar.gz

RUN sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_74/bin/java 100

RUN sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_74/bin/javac 100

```
# Install OpenJDK-8
RUN apt-get update && \
    apt-get install -y openjdk-8-jdk && \
    apt-get install -y ant && \
    apt-get clean;

# Fix certificate issues
RUN apt-get update && \
    apt-get install ca-certificates-java && \
    apt-get clean && \
    update-ca-certificates -f;

# Setup JAVA_HOME -- useful for docker commandline
ENV JAVA_HOME /usr/lib/jvm/java-8-openjdk-amd64/
RUN export JAVA_HOME
```

Create java program

```
public class javaprogram{
    public static void main(String[] args){
        System.out.println("Hello world");
    }
}
```

Compile and run it.

```
root@a613e9cf0f1d:/opt# javac javaprogram.java
root@a613e9cf0f1d:/opt# ls
javaprogram.class  javaprogram.java
```

```
root@a613e9cf0f1d:/opt# java javaprogram
Hello world
```

Task 4 – Create your own linked container

To create mysql container that stores data in the host volume, create a directory

```

ubuntu@ip-172-31-7-183: ~
ubuntu@ip-172-31-7-183:/home$ ls
ubuntu
ubuntu@ip-172-31-7-183:/home$ cd ubuntu/
ubuntu@ip-172-31-7-183:~$ ls
dockerfile
ubuntu@ip-172-31-7-183:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-7-183:~$ sudo mkdir mysqldir
ubuntu@ip-172-31-7-183:~$ ls
dockerfile  mysqldir

```

docker run --name some-mysql -v /home/ubuntu/mysqldir:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:tag

where **my-secret-pw** is password, **tag** is version number e.g. 8.0

e.g. **docker run --name some-mysql -v /home/ubuntu/mysqldir:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=abcde -d mysql:8.0**

Find running process with

sudo docker ps -a -f status=running

Execute interactive terminal –it inside a container using exec and containerID

sudo docker exec -it containerID /bin/bash

```

ubuntu@ip-172-31-7-183:~/mysqldir$ sudo docker ps -a -f status=running
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
2f8542bc24de       php:7.0-apache     "docker-php-entryp..." 4 minutes ago      Up 4 minutes       80/tcp
ea11b44952de       ubuntu             "bash"              2 hours ago        Up 2 hours
35f39df042c9       mysql:8.0          "docker-entrypoint..." 19 hours ago       Up 2 hours         3306/tcp
3ee2780a2a10       jenkins            "/bin/tini -- /usr..." 24 hours ago       Up 2 hours         0.0.0.0:8080-
0:50000->50000/tcp  dreamy_jang
ubuntu@ip-172-31-7-183:~/mysqldir$ sudo docker exec -it 35f39df042c9 /bin/bash
root@35f39df042c9:/# ls
bin      dev          entrypoint.sh  home  lib64  mnt  proc  run  srv  tmp  var
boot    docker-entrypoint-initdb.d  etc          lib   media  opt  root  sbin  sys  usr
root@35f39df042c9:/# cd /var/lib/mysql
root@35f39df042c9:/var/lib/mysql# ls
auto.cnf      client-cert.pem  ib_logfile0  ibtmp1          performance_schema  server-cert.pem  sys_4.SDI
ca-key.pem    client-key.pem   ib_logfile1  mysql           private_key.pem     server-key.pem
ca.pem        ib_buffer_pool  ibdata1      performance_sche_3.SDI  public_key.pem      sys
root@35f39df042c9:/var/lib/mysql#

```

For PHP project, pull php image:

sudo docker pull php

When running the container mount the directory to a newdirectory in php file path

sudo docker run -it -v /home/ubuntu/mysqldir:/home/newdir php:7.0-apache bash

```

ubuntu@ip-172-31-7-183:~/mysqldir$ sudo docker run -it -v /home/ubuntu/mysqldir:/home/newdir php:7.0-apache bash
root@2f8542bc24de:/var/www/html# cd /home/newdir
root@2f8542bc24de:/home/newdir# ls
auto.cnf      client-cert.pem  ib_logfile0  ibtmp1          performance_schema  server-cert.pem  sys_4.SDI
ca-key.pem    client-key.pem   ib_logfile1  mysql           private_key.pem     server-key.pem
ca.pem        ib_buffer_pool  ibdata1      performance_sche_3.SDI  public_key.pem      sys

```

Task 5 - Create your own docker-compose file

Create a directory to be used as shared folder.

/home/Ubuntu/mysqldir contains many files from task4

```
ubuntu@ip-172-31-7-183:~$ cd mysqldir/
ubuntu@ip-172-31-7-183:~/mysqldir$ ls
auto.cnf      ca.pem        client-key.pem  ibdata1       ib_logfile1  performance_sche_3.SDI  phptext.txt
ca-key.pem    client-cert.pem  ib_buffer_pool  ib_logfile0  mysql        performance_schema      private_key.pem
```

Create **docker-compose.yml** file as below:

version : '2'

services:

db:

image: mysql:8.0

ports:

- "3333:3333"

volumes:

- /home/ubuntu/mysqldir:/var/www/html

environment:

MYSQL_ROOT_PASSWORD: passwerd

php:

image: php:7.0-apache

links:

- db:db

ports:

- "80:80"

volumes:

- /home/ubuntu/mysqldir:/var/www/html

PHP an mysql are running

```
ubuntu@ip-172-31-7-183:~$ sudo docker-compose up -d
Starting ubuntu_db_1
Recreating ubuntu_php_1
ubuntu@ip-172-31-7-183:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
ac8da15d92a4        php:7.0-apache     "docker-php-entryp..." 6 seconds ago       Up 5 seconds        0.0.0.0:80->80/tcp
4557d923d586        mysql:8.0           "docker-entrypoint..." 22 minutes ago      Up 5 seconds        3306/tcp,
```

Execute the first container and check shared directory which contains files as expected.

```
ubuntu@ip-172-31-7-183:~$ sudo docker exec -ti ac8da15d92a4 bash
root@ac8da15d92a4:/var/www/html# ls
auto.cnf      ca.pem        client-key.pem  ib_logfile0  ibdata1  performance_sche_3.SDI  phptext.txt
ca-key.pem    client-cert.pem  ib_buffer_pool  ib_logfile1  mysql    performance_schema      private_key.pem
```

Execute the second container and it can be seen that the directory is successfully mounted as the files expected are present.

```
ubuntu@ip-172-31-7-183:~$ sudo docker exec -ti 4557d923d586 bash
root@4557d923d586:/# ls
bin boot dev docker-entrypoint-initdb.d  entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  rd
root@4557d923d586:/# cd /var/www/html
root@4557d923d586:/var/www/html# ls
auto.cnf      ca.pem        client-key.pem  ib_logfile0  ibdata1  performance_sche_3.SDI  phptext.txt
ca-key.pem    client-cert.pem  ib_buffer_pool  ib_logfile1  mysql    performance_schema      private_key.pem
```