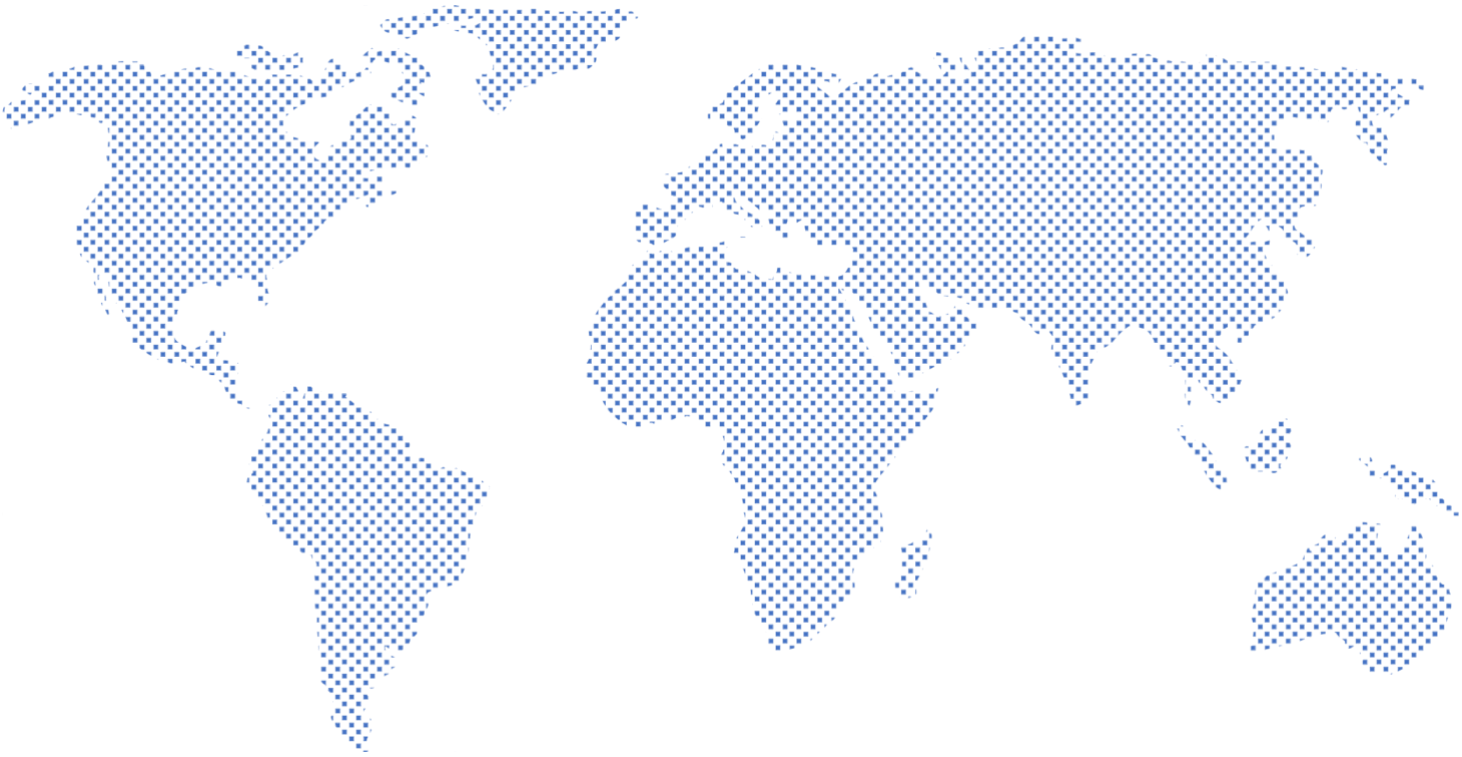
Pega Personal Edition
Install on Ubuntu Server



SRINIVASA KANDRU

Sreesoft Solutions

SreeSoft

SreeSoft

Contents

About this document

Currently Pega is available for learning in the form of personal edition (PE). PE works only on windows. There are few occasional PVS instances available from Pega Academy, but they are limited to older versions of Pega. This document details steps on creating an instance of virtual server running Pega. This virtual image can be easily run on any other OS such as Linux, Mac OS X.

For this document, we used Virtual Box to create the instance. Same steps are useful for any other virtualisation platform such as VMWare, Parallels…etc.

Disclaimer & Licensing Terms

This document is only for educational purposes. Whoever following it, must understand Pega's Personal Edition licencing terms and adhere to it. This document does not distribute Pega PE software, this document is only for who have Pega PE authorised software and wants to learn how to fit that into multiple work environments. It's purely for educational purposes.

There is no guarantee the below steps work in a specific environment. Please use caution while following the instructions.

Check Prerequisites

Before installing Pega images, ensure all the prerequisites are met and available in the system.

Ensure Virtual Box software installed and running in the target system. Virtual Box can be downloaded from [Downloads – Oracle VM VirtualBox]

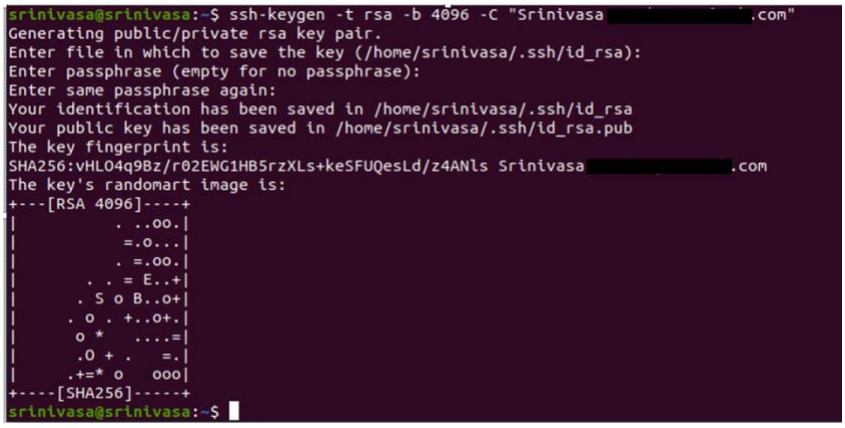
1. Once VirtualBox is downloaded and installed, verify the dashboard loaded correctly and no errors reported. Usual virtualBox dashboard looks as below.

2. Have a GitHub account and familiarise how to check-out repositories.
3. Install git command line utility as described here.
   https://www.atlassian.com/git/tutorials/install-git

   It may be required to authenticate local system with GitHub to allow check-out a repository. Please follow the below articles to resolve any issues if they arise.

   https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/adding-a-new-ssh-key-to-your-github-account

   https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent

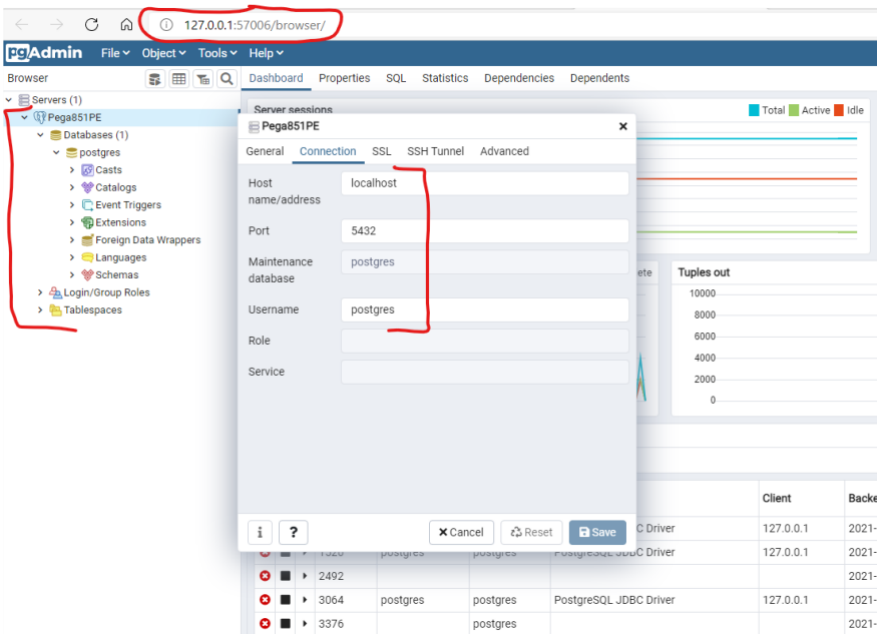


4. Install pgAdmin 4 client.
5. Keep/Extract Pega Database backup from Personal Edition. To extract, follow the related section in this document.

1. Extract Pega Database Backup from PE (Personal Edition)

To run Pega in Docker containers, we are relying on postgres database backup extracted from a running Pega instance. In this case, it is going to be Personal Edition of Pega. (Following steps can be applied to any other pega instance running on postgres)

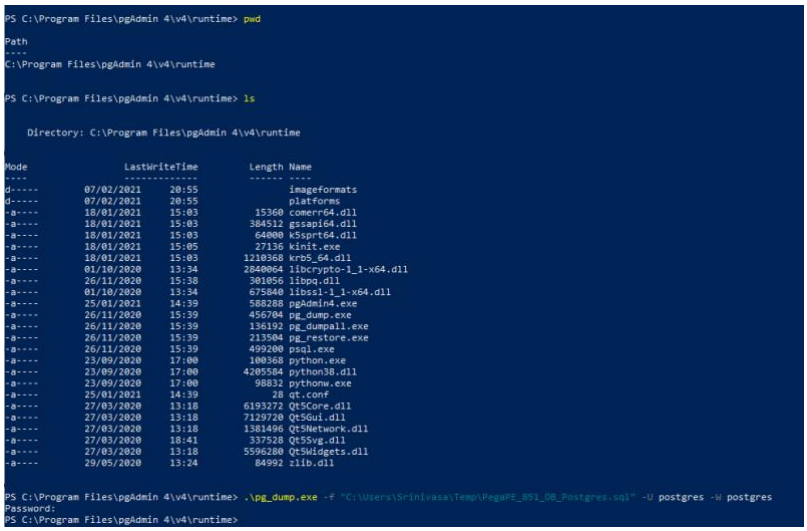
1. To extract database backup Postgres DB, we are going to use **pg_dump** tool. (this tool comes with pgAdmin 4 standard installation)
2. Install Pega personal edition before going to next step.
3. Check/Configure DB access with pre-installed pgadmin-4 client as shown in below screenshot. (password is usually either ‘postgres’ or ‘password’)



4. Below example is taken from Windows, for other OS, go through the postgres recommendations.
5. Run below command (or similar as per your environment) to extract postgres DB backup to a chosen folder (For ex : "C:\Users\Srinivasa\Temp")

In Windows :

C:\Program Files\pgAdmin 4\v4\runtime> .\pg_dump.exe -f "C:\Users\Srinivasa\Temp\PegaPE_851_DB_Postgres.sql" -U postgres -W postgres



In Linux (with remote postgres) :

/usr/bin/pg_dump --host "192.168.2.111" --port "5432" --username "postgres" --password --dbname "postgres" --file "/home/srinivasa/Downloads/PegaPE_851_ DB_Postgres.sql"

6. **Keep the backed-up copy handy for future steps.**

To restore the backup copy into fresh postgres image, below/similar commands are used. (These commands are useful in future steps too)

pg_restore if binary backup taken

/usr/bin/pg_restore --host "192.168.2.71" --port "5432" --username "postgres" --no-password --dbname "postgres" --verbose "/share/samba/PegaPE_851_DB_Postgres.sql"

psql if sql/text backup taken

/usr/bin/psql --host "192.168.2.111" --port "5432" --username "postgres" --password --dbname "postgres" --file "/home/srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"

2. Pull GitHub repository for necessary config files

To install Pega in Ubuntu server, it is required to have necessary config files. All such files are documented in GitHub and follow steps to get them to local machine.

GitHub repository is available at https://github.com/sreesoft/pegaubuntuvm

Follow step-by-step guide to clone this repository and build your own local copy.

1. First, choose a base folder to keep all files cloned from GitHub. (Example folder is visible in the below screenshot. 'GitHub')
2. Use Terminal/Command Line/Power shell (based on operating system) to go to this folder.
3. Once in this folder, use git clone command to clone 'pegapostgres' repo as shown in below screenshot.

git clone git@github.com:sreesoft/pegaubuntuvm.git

4. It creates a folder 'pegaubuntuvm' all pulled files appear inside this folder. These files are useful for further steps in this document, so keep this folder handy.

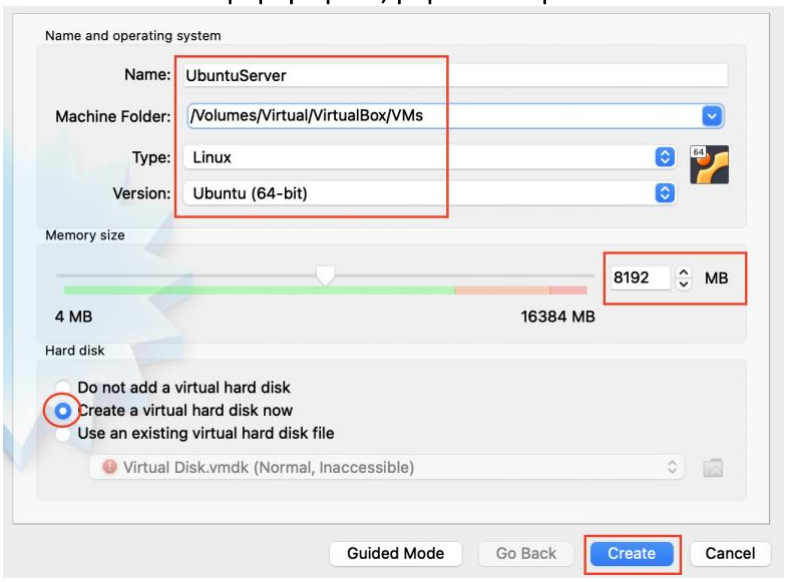
3. Create Ubuntu Server Shell

Download the latest Ubuntu Server ISO image from Ubuntu site. (At the time of this document, it is 20.04 version). Downloads of Ubuntu ISO are available at [Get Ubuntu Server | Download | Ubuntu]

Once downloaded, keep the ISO copy handy for further steps.

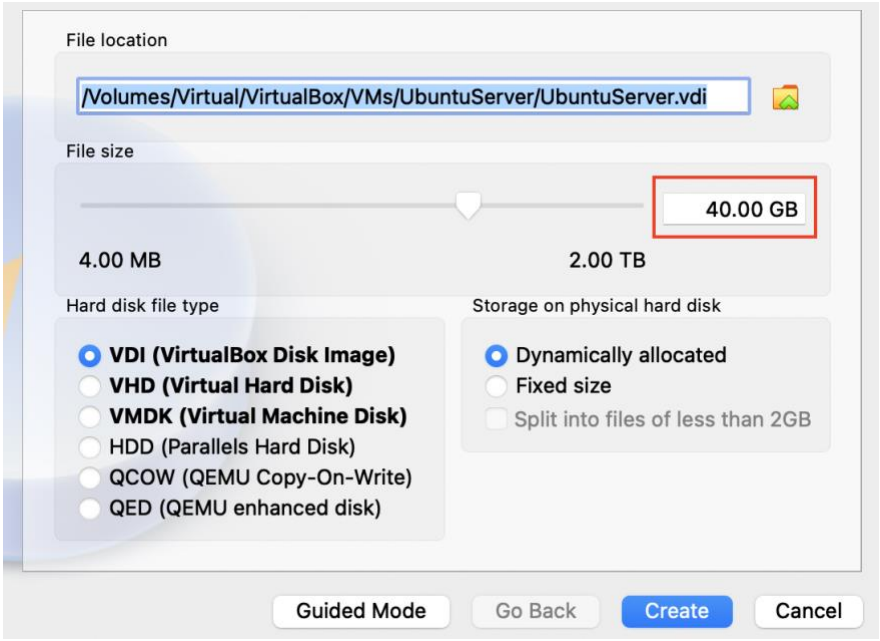
Open VirtualBox dashboard and create an instance using 'New' option as shown below.

When New instance popup opens, populate required fields as shown below...



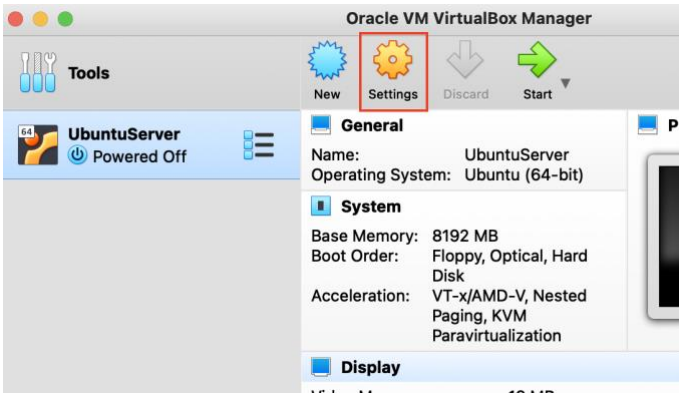
- Used Memory size as 8192MB, use as per your resource availability. More is better.
- Choose creating a virtual hard disk to expand it when needed.
- Choose Machine Folder as per target system location.
- Click on 'Create' button.

When it prompts for new Virtual Disk Image, choose options as shown in below screenshot.

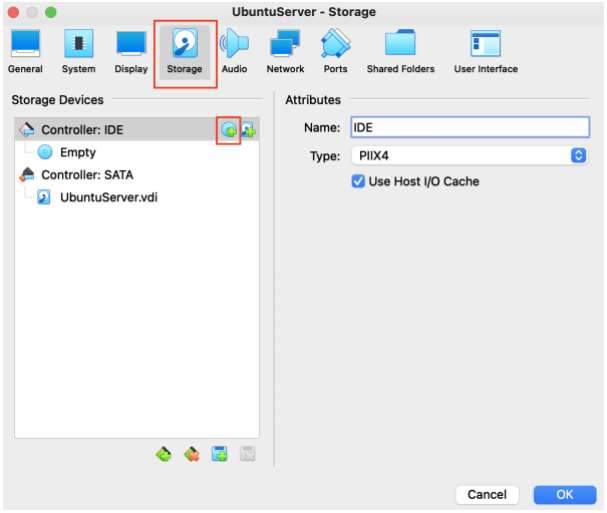


Specify at-least 40GB of disk space and Dynamically allocated.
Click on 'Create' button

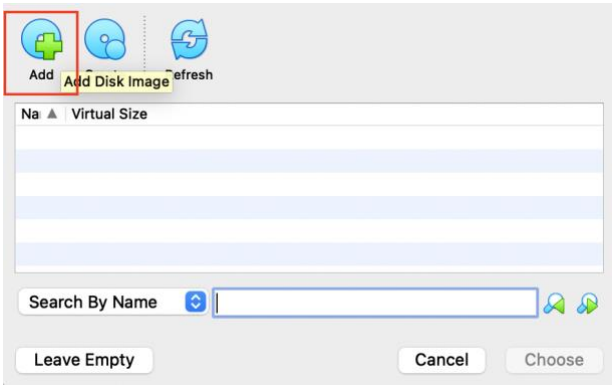
When popup completed/closed, click on 'Settings' icon on dashboard of VirtualBox.

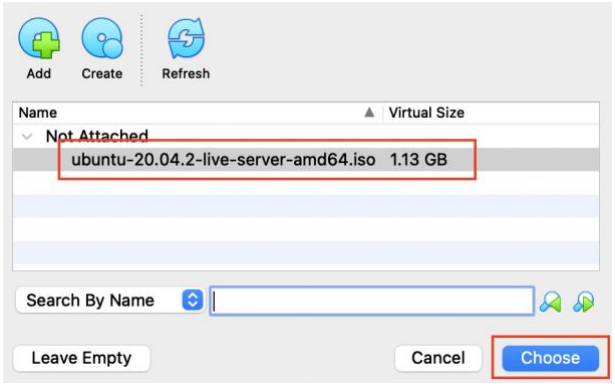


On the popup opened, select 'Storage' and add a CD image by selecting the CD icon as shown in below screenshot.

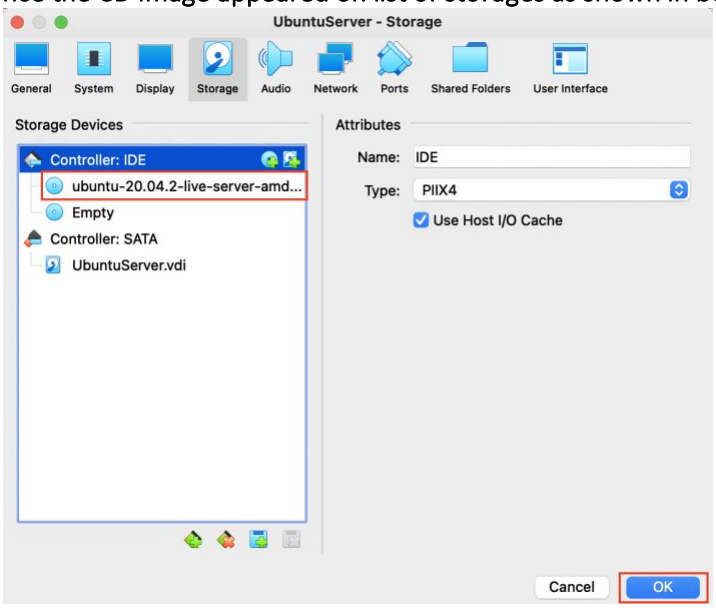


Click on ‘Add’ button as shown in below screenshot to choose the Ubuntu ISO downloaded earlier.

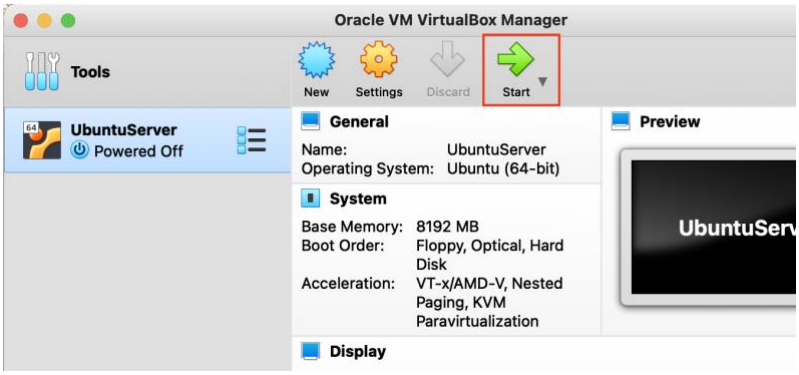




Once the CD Image appeared on list of storages as shown in below screenshot, click ‘OK’.

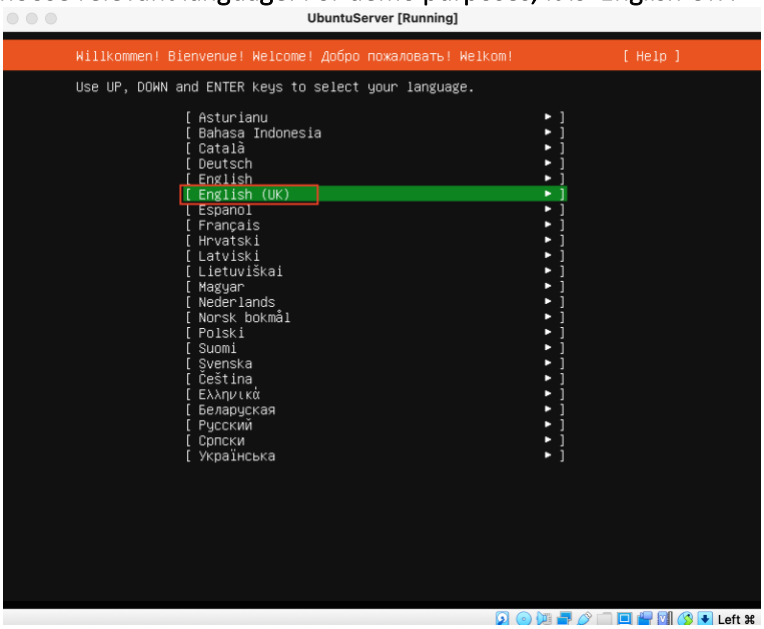


On Dashboard, click on Start icon to start the VM instance using the CD ISO image of Ubuntu Server.

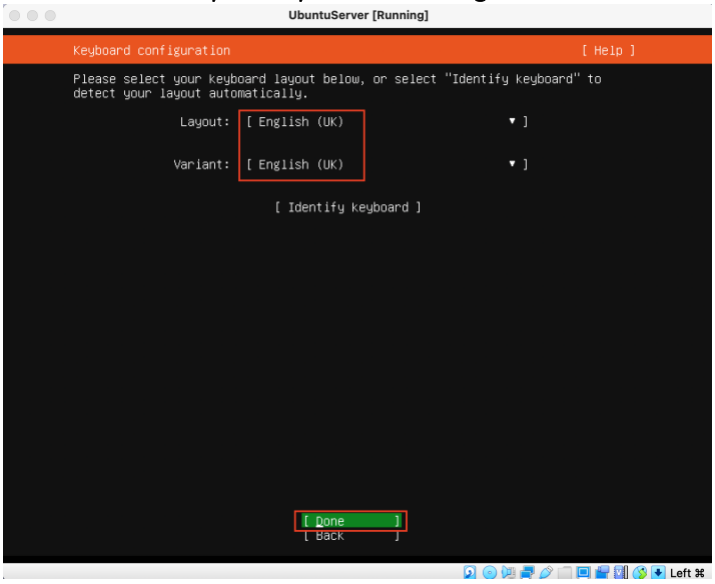


Follow onscreen instructions to install Ubuntu Server. Key steps are explained below with screenshots.

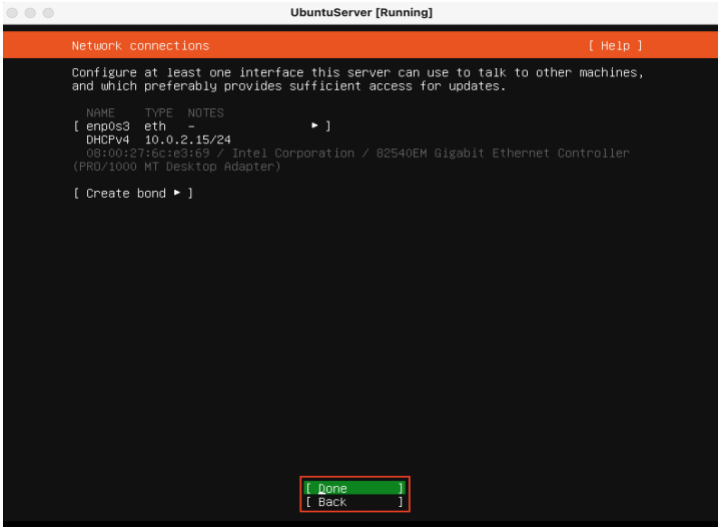
Choose relevant language. For demo purposes, it is ‘English UK’.



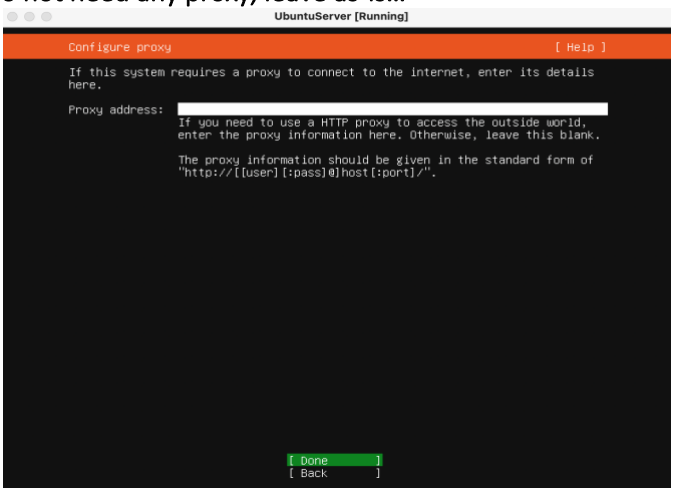
Use Tab & Enter keys in keyboard to change the selection of Layout & Variant.



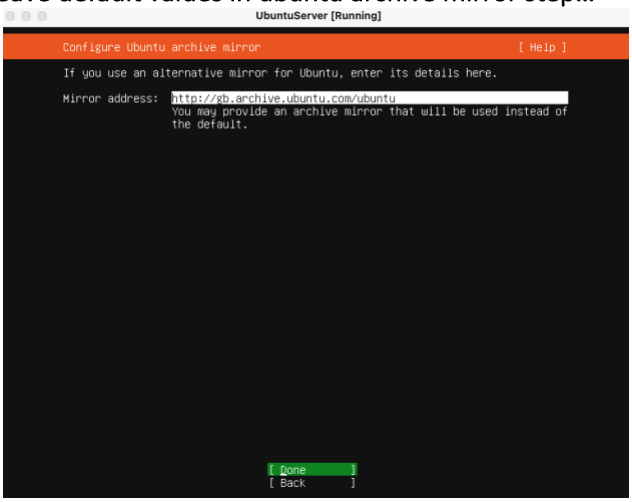
Leave network connections to defaults…



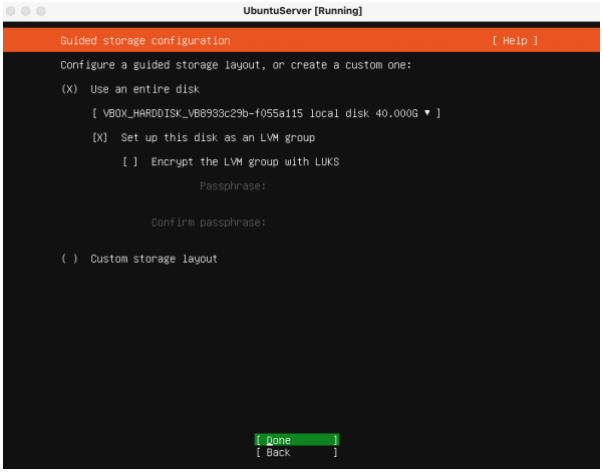
Do not need any proxy, leave as-is...



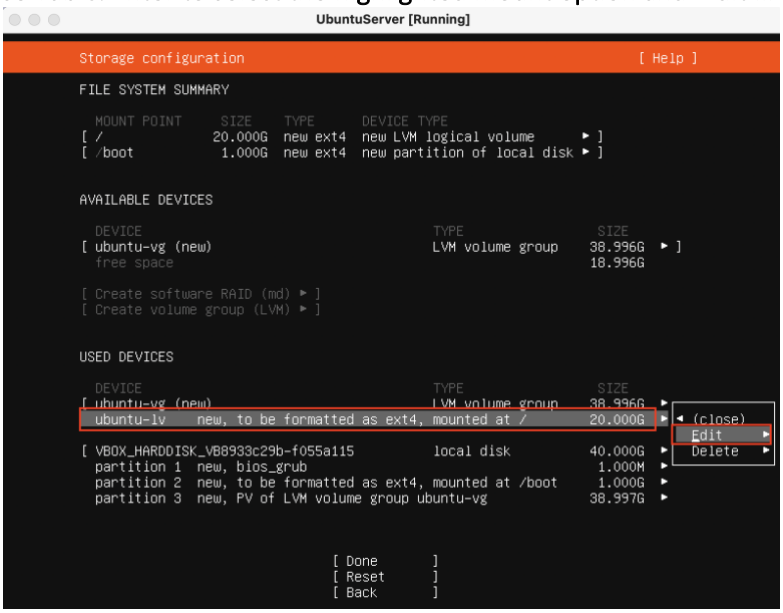
Leave default values in ubuntu archive mirror step...



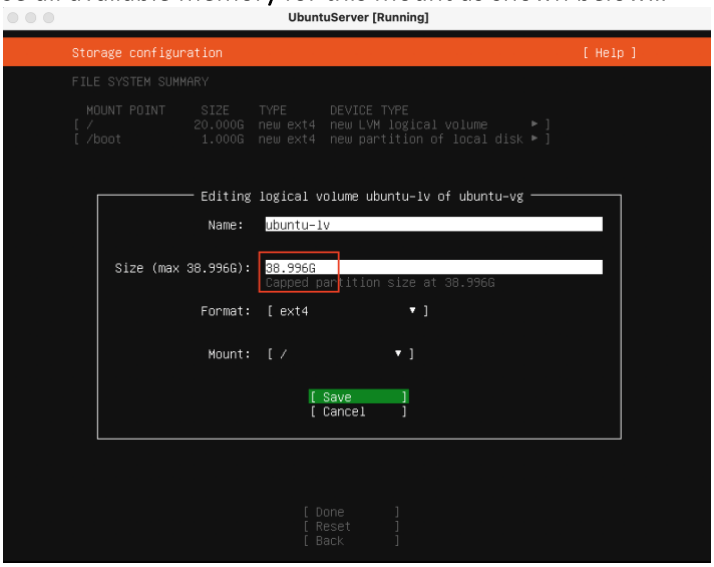
Choose entire disk option as below...



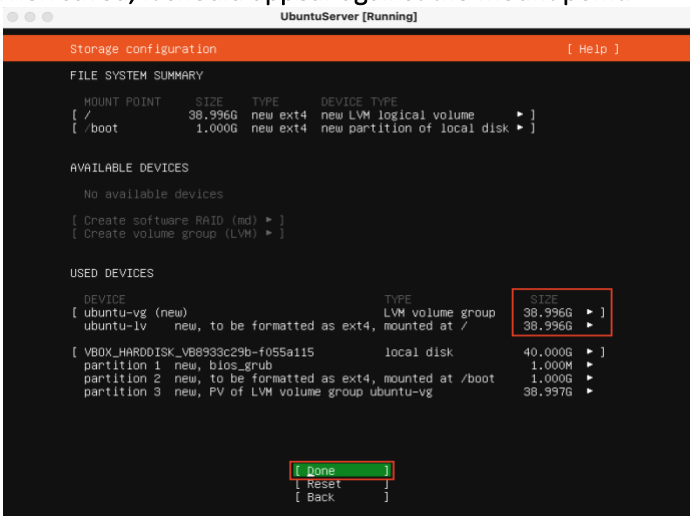
When configuring storage, map all available memory into mount point as shown below.
Use Tab & Enter to select the highlighted mount option and 'Edit'…



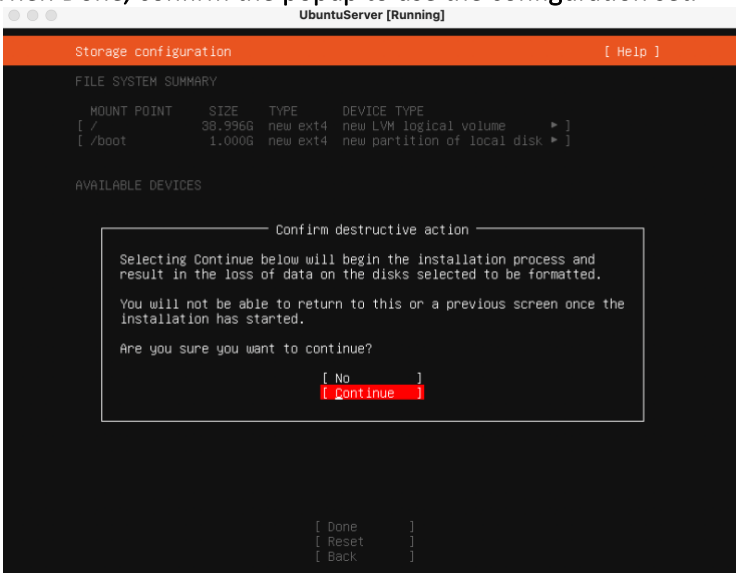
Use all available memory for this mount as shown below…



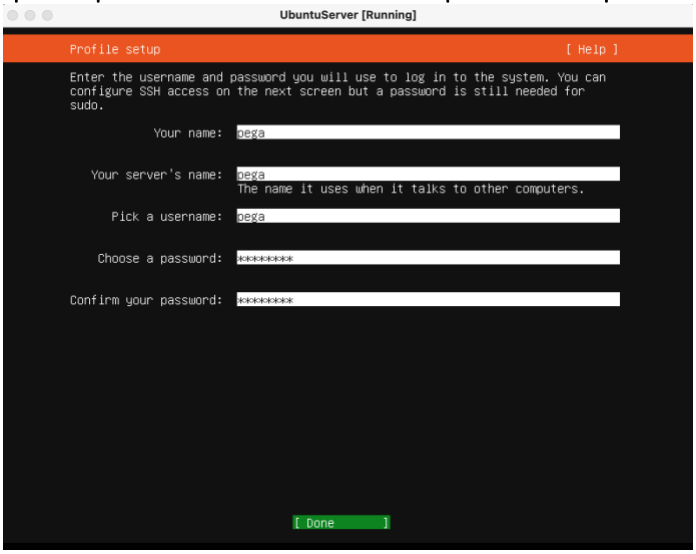
When saved, it should appear against the mount point.



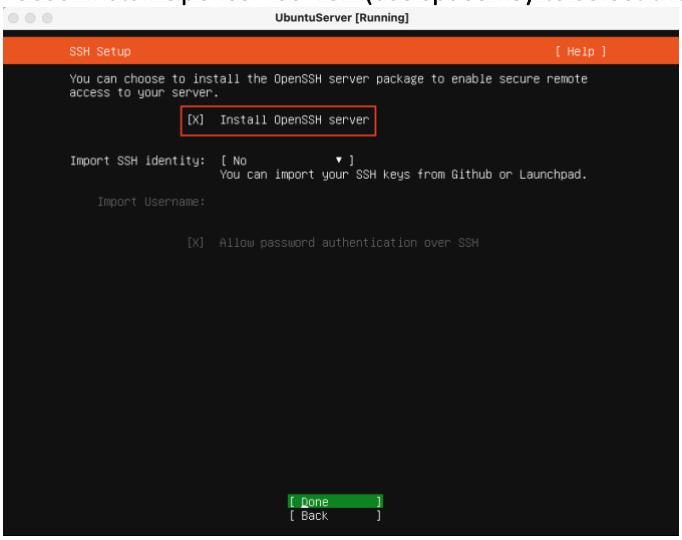
When Done, confirm the popup to use the configuration set.



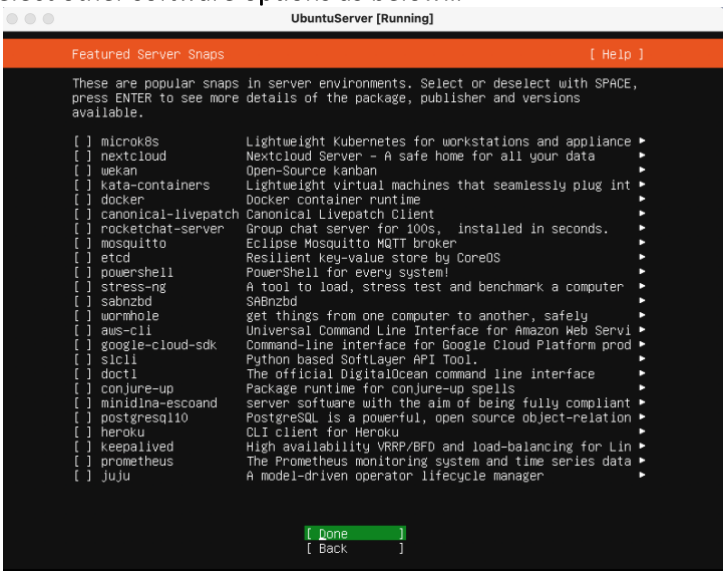
Populate profile details as below… Use password as ‘password’



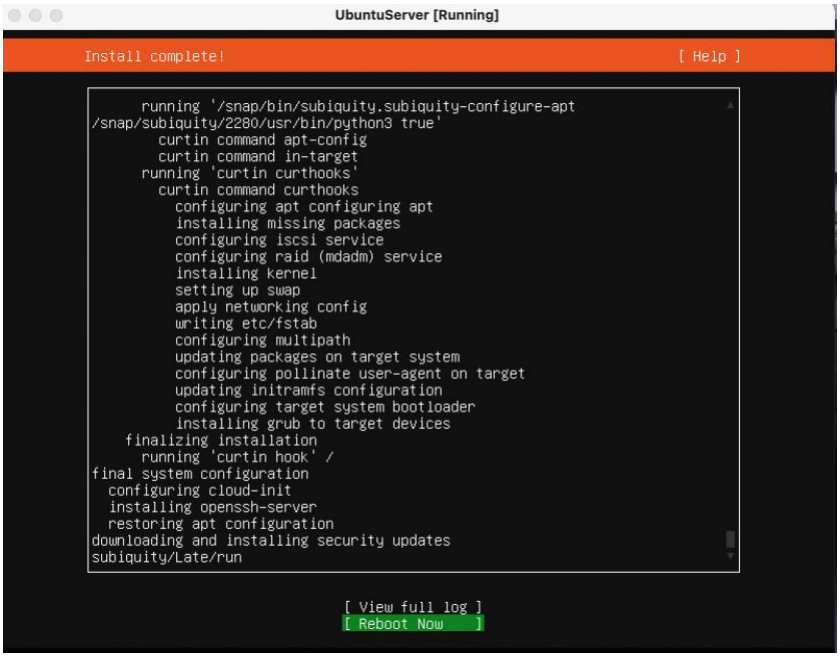
Choose ‘Install OpenSSH server’ (use space key to select the option)



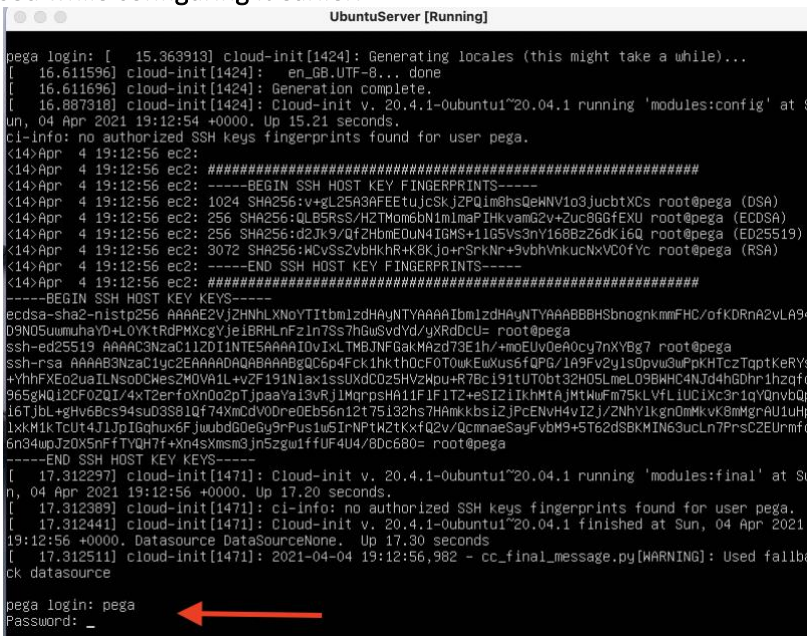
Select other software options as below…



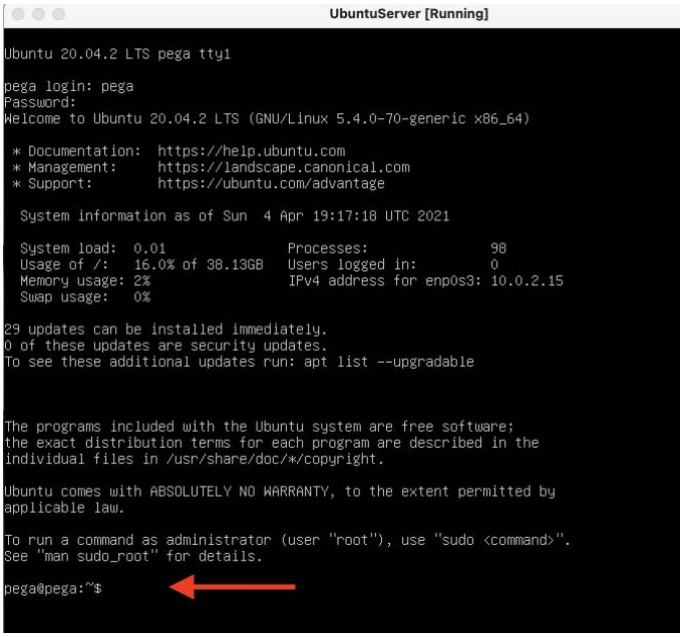
When installation completes, ‘Reboot Now’ option will be visible. Select it to reboot the system.



When successfully rebooted, it appears as below with login prompt. Use password as ‘password’ or the one used while configuring it earlier.

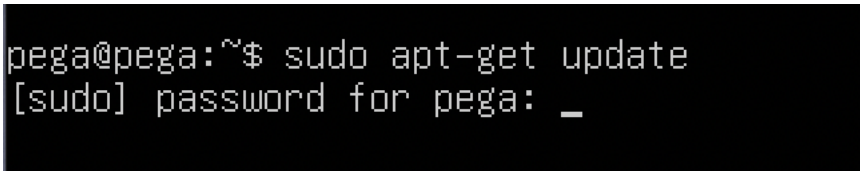


On successful logon screen appears as below…



Now run the below commands to install few more software's.

sudo apt-get update



sudo apt-get install -y net-tools

sudo apt-get install -y openjdk-8-jdk

Once Java is installed, it is time to set JAVA_HOME permanently in the system.

First check the Java version and run the below commands.

java -version
sudo update-java-alternatives -l (Copy the result, here it is /usr/lib/jvm/java-1.8.0-openjdk-amd64)
sudo vi /etc/profile

```
pega@pega:~$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~20.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
pega@pega:~$ sudo update-java-alternatives -l
[sudo] password for pega:
java-1.8.0-openjdk-amd64       1081       /usr/lib/jvm/java-1.8.0-openjdk-amd64
pega@pega:~$ vi /etc/profile
pega@pega:~$ sudo vi /etc/profile
```

In profile file add the below lines at the end.

```
    . $i
  fi
 done
 unset i
fi

JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
```

JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH

Test JAVA_HOME variable using the below command. It should display the path of Java.

```
pega@pega:/pega/logs$ echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk-amd64
pega@pega:/pega/logs$
```

Create below folders for Pega purposes.

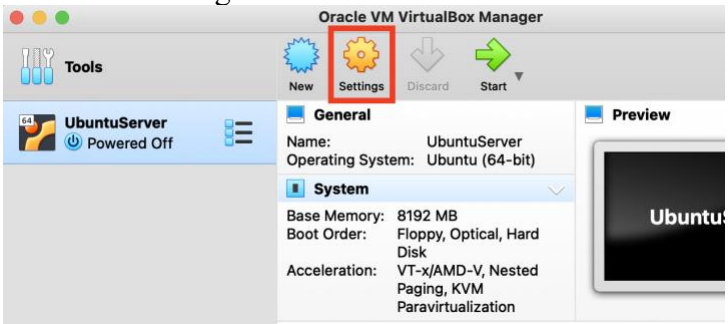
sudo mkdir /pega
sudo mkdir /pega/logs
sudo mkdir /pega/index
sudo mkdir /pega/temp
sudo mkdir /pega/cassandra_data

Now restart the system using the below command.

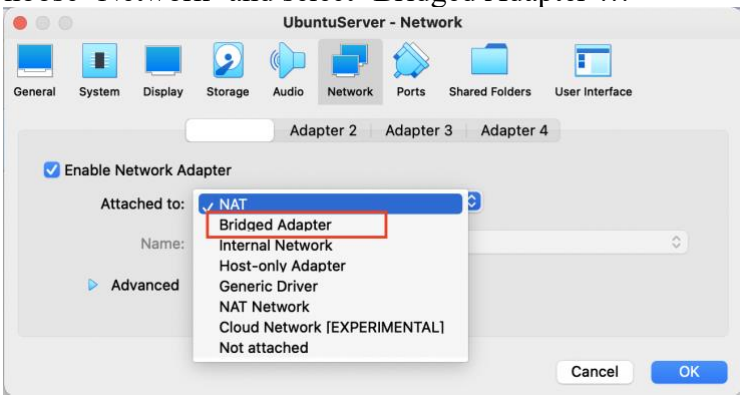
shutdown -h now

```
pega@pega:~$
pega@pega:~$
pega@pega:~$
pega@pega:~$ shutdown -h now
```

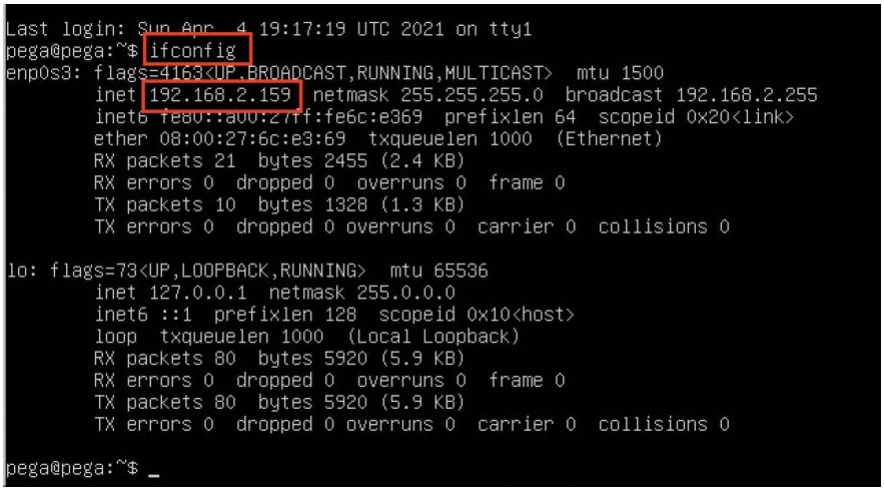
Before starting the system again, set VirtualBox Image network adaptor as bridged as shown below. First select 'Settings'…



Choose ‘Network’ and select ‘Bridged Adapter’…



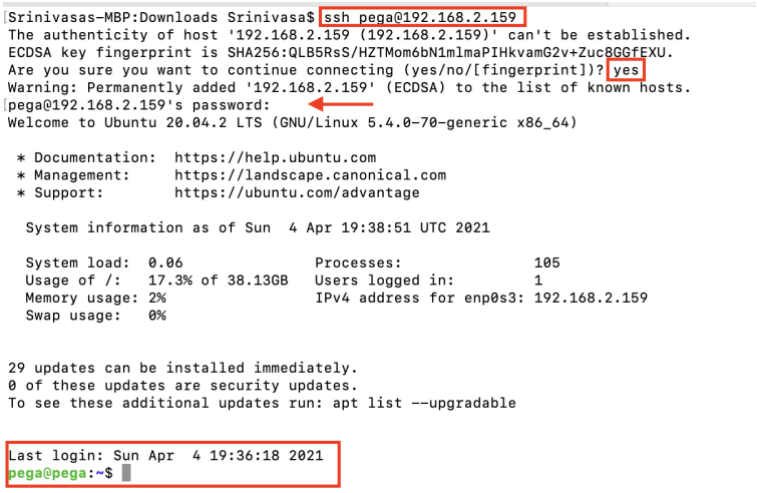
Start the VM image again.
Login using pega/password, run command ifconfig to get the IP address.



IP Address identified is useful to connect this system remotely via Putty or Terminal type tools. Since OpenSSH Server already installed, we can straight away use one of such tools. For this document, Terminal of Mac OS is used.

Open Terminal app and run ssh command as below.

ssh pega@192.168.2.159 (alternatively it can be used as ‘ssh pega@pega’)



From here, all commands can be run remotely using ssh session.

4. Install Postgres Database

Now we have plain Ubuntu server without any database or app server. This section details the installation of Database Server which is ‘postgres’.

Run below commands via ssh session and install postgres.

```
sudo apt-get update
sudo apt install -y postgresql postgresql-contrib
sudo -u postgres createuser –interactive
sudo -i -u postgres
psql
alter role postgres with password 'postgres';
```

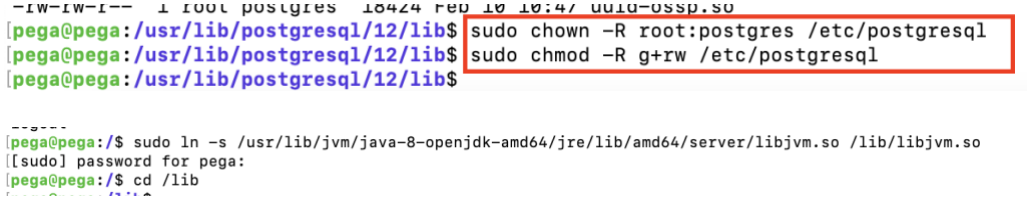
```
pega@pega:/$ sudo -u postgres createuser –interactive
pega@pega:/$ sudo -i -u postgres
postgres@pega:~$ psql
psql (12.6 (Ubuntu 12.6-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# alter role postgres with password 'postgres';
ALTER ROLE
postgres=#
postgres=#
```

```
exit
exit
```

```
postgres=#
postgres=# exit
postgres@pega:~$ exit
logout
pega@pega:/$
```

Now Postgres is installed and time to make configuration changes…

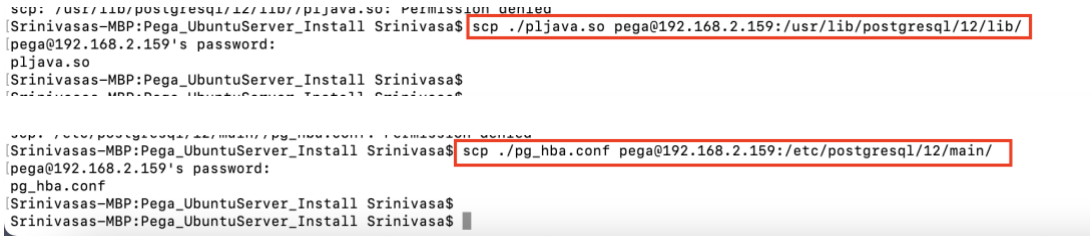
```
sudo chown -R root:postgres /usr/lib/postgresql
sudo chmod -R g+rw /usr/lib/postgresql
sudo chown -R root:postgres /etc/postgresql
sudo chmod -R g+rw /etc/postgresql
sudo usermod -a -G postgres pega
sudo ln -s /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so /lib/libjvm.so
```



Copy/upload pre-prepared config files (cloned from GitHub earlier in this document) from host machine to ubuntu server using one of the below methods. Whatever is comfortable…

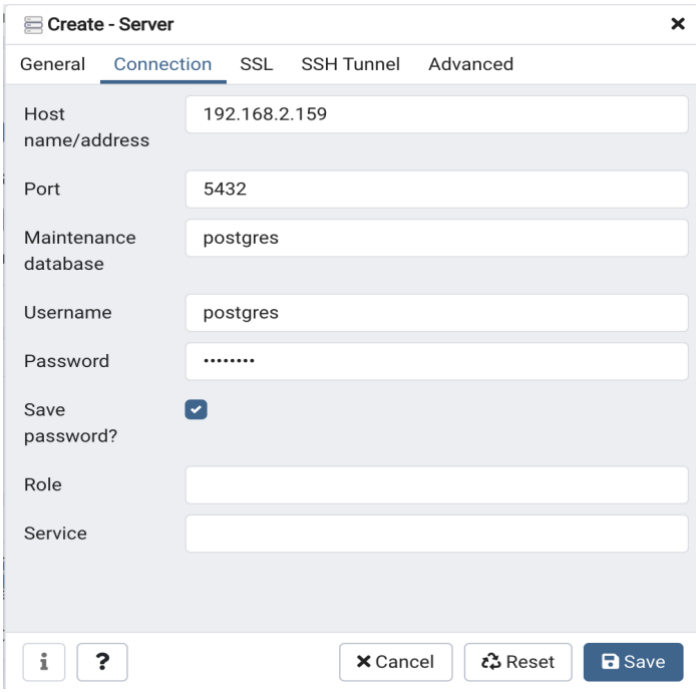
Use Terminal/Command Line utility to run scp commands to copy files from host to ubuntu server.
Go to base folder location of these config file pulled from GitHub earlier and run scp commands. Please note IP Address may vary, use appropriately.

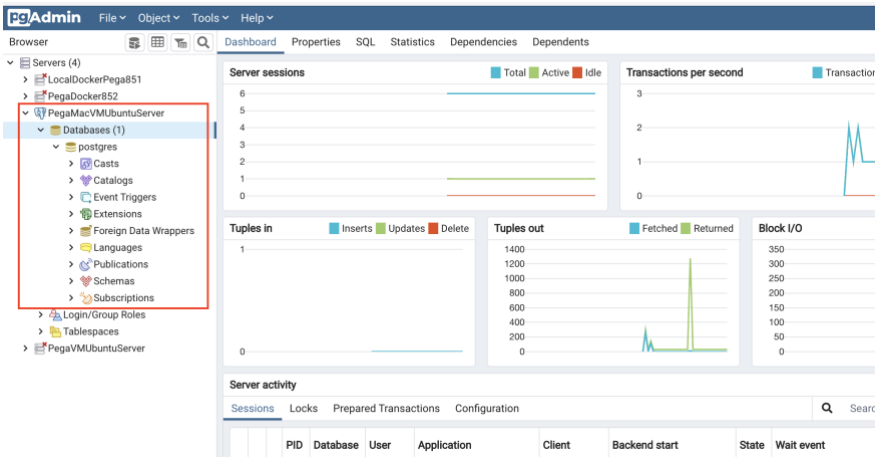
```
scp ./pljava.so pega@192.168.2.159:/usr/lib/postgresql/12/lib/
scp ./pljava.jar pega@192.168.2.159:/usr/lib/postgresql/12/lib/
scp ./pg_hba.conf pega@192.168.2.159:/etc/postgresql/12/main/
scp ./postgresql.conf pega@192.168.2.159:/etc/postgresql/12/main/
```



Alternatively, use **filezilla** or similar tool to upload them to target locations using GUI options.

Restart Ubuntu server as explained earlier and try connecting postgres from pgadmin4 client.
Create a new connection and use connection details as below. Password is ‘postgres’.





4.1 Export Backed-up Pega DB file via host ‘psql’ utility.

Find ‘psql’ utility in host machine. If its Mac, it usually available in the shown path below.
Once found, run the psql command as shown. File paths & IP Address may vary.

./psql --host "192.168.2.159" --port "5432" --username "postgres" --password --dbname "postgres" --file "/Users/Srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"

Password to use here is ‘postgres’

Once password is accepted, it will start loading all tables and objects into Postgres container. It may take a while depends on target system capacity. Once completed, it will show similar to below.

```
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
Srinivasas-MBP:SharedSupport Srinivasa$
```

If no errors are presented, that’s it. DB is successfully loaded.

5. Install Tomcat Server

Tomcat is used for running Pega platform for this document. This section describes the steps to install it.

First, create tomcat user and group with the below command via ssh terminal.

```
sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
```

```
pega@pega:~$ sudo useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat
[sudo] password for pega:
pega@pega:~$
```

Find the latest version of Tomcat 9 at the Tomcat 9 Downloads page. At the time of writing, the latest version is **9.0.44**, if any later stable version available, can be used too. For this document purpose, we used tomcat 9.0.44 version and its link as below.

https://downloads.apache.org/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz

Run the command to download latest tomcat package into /tmp folder.

```
wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz -P /tmp
```

```
pega@pega:~$ wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz -P /tmp
--2021-04-05 12:13:16--  https://downloads.apache.org/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 2a01:4f8:10a:201a::2
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11487016 (11M) [application/x-gzip]
Saving to: '/tmp/apache-tomcat-9.0.44.tar.gz'

apache-tomcat-9.0.44.tar.gz                         100%[==================================================>]

2021-04-05 12:13:18 (5.36 MB/s) - '/tmp/apache-tomcat-9.0.44.tar.gz' saved [11487016/11487016]

pega@pega:~$
pega@pega:~$
pega@pega:~$
```

Extract tomcat contents into /opt/tomcat folder and create link for latest version of tomcat…

sudo tar xf /tmp/apache-tomcat-9*.tar.gz -C /opt/tomcat
sudo ln -s /opt/tomcat/apache-tomcat-9.0.44 /opt/tomcat/latest

```
[pega@pega:~$ sudo tar xf /tmp/apache-tomcat-9*.tar.gz -C /opt/tomcat
[[sudo] password for pega:
[pega@pega:~$ sudo ln -s /opt/tomcat/apache-tomcat-9.0.44 /opt/tomcat/latest
[pega@pega:~$
[pega@pega:~$
```

Set all permissions of key config folders of tomcat with tomcat group.

sudo chown -RH tomcat: /opt/tomcat/latest
sudo sh -c 'chmod +x /opt/tomcat/latest/bin/*.sh'
sudo sh -c 'chmod g+rx /opt/tomcat/latest/conf/'
sudo sh -c 'chmod g+r /opt/tomcat/latest/conf/*'

Get Java path and update tomcat.service file pulled from GitHub.

sudo update-java-alternatives -l

```
[pega@pega:~$ sudo sh -c 'chmod g+rx /opt/tomcat/latest/conf/'
[pega@pega:~$ sudo sh -c 'chmod g+r /opt/tomcat/latest/conf/*'
[pega@pega:~$ sudo update-java-alternatives -l
java-1.8.0-openjdk-amd64       1081       /usr/lib/jvm/java-1.8.0-openjdk-amd64
pega@pega:~$
```

tomcat.service

```
[Unit]
Description=Tomcat 9 servlet container
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"

Environment="CATALINA_BASE=/opt/tomcat/latest"
Environment="CATALINA_HOME=/opt/tomcat/latest"
Environment="CATALINA_PID=/opt/tomcat/latest/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/latest/bin/startup.sh
ExecStop=/opt/tomcat/latest/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```

Either use vi editor to add contents of this file or upload it via host terminal.

sudo vi /etc/systemd/system/tomcat.service (This command to run in Ubuntu server shell and copy paste the contents)

Alternatively, upload this file to target location from host terminal via scp command and move…

```
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$ ll
total 42776
-rwxr-xr-x@ 1 Srinivasa  staff  16773922  5 Apr 13:34 Pega_PersonalEdition_Install_On_UbuntuServer.docx
-rwxr-xr-x@ 1 Srinivasa  staff   3060443 28 Mar 13:53 Pega_UbuntuServer_Install.docx
-rwxr-xr-x@ 1 Srinivasa  staff      3549 22 Mar 16:53 context.xml
-rwxr-xr-x  1 Srinivasa  staff      4958 22 Mar 16:58 pg_hba.conf
-rwxr-xr-x  1 Srinivasa  staff    340436 20 Dec 08:12 pljava.jar
-rwxr-xr-x  1 Srinivasa  staff    329672 20 Dec 08:12 pljava.so
-rwxr-xr-x  1 Srinivasa  staff    932808 20 Dec 08:13 postgresql-42.2.14.jar
-rwxr-xr-x  1 Srinivasa  staff     27013 22 Mar 16:58 postgresql.conf
-rwxr-xr-x  1 Srinivasa  staff    395800 26 Nov 23:33 prweb.war
-rwxr-xr-x  1 Srinivasa  staff       313 20 Dec 08:13 setenv.sh
-rwxr-xr-x  1 Srinivasa  staff      2320 22 Mar 16:53 tomcat-users.xml
-rwxr-xr-x  1 Srinivasa  staff       628 23 Mar 11:52 tomcat.service
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$ scp tomcat.service pega@192.168.2.159:/tmp
pega@192.168.2.159's password:
tomcat.service
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$
```

Once uploaded to /tmp folder, move it to target location…

sudo cp /tmp/tomcat.service /etc/systemd/system/

```
pega@pega:~$ sudo update-java-alternatives -l
java-1.8.0-openjdk-amd64       1081       /usr/lib/jvm/java-1.8.0-openjdk-amd64
pega@pega:~$ sudo cp /tmp/tomcat.service /etc/systemd/system/
pega@pega:~$
pega@pega:~$
```

Update system daemon to add tomcat service and start tomcat…

sudo systemctl daemon-reload
sudo systemctl start tomcat
sudo service tomcat status
sudo systemctl enable tomcat
sudo ufw allow 8080/tcp

```
pega@pega:~$
pega@pega:~$ sudo systemctl daemon-reload
pega@pega:~$ sudo systemctl start tomcat
pega@pega:~$ sudo service tomcat status
● tomcat.service - Tomcat 9 servlet container
     Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
     Active: active (running) since Mon 2021-04-05 12:40:35 UTC; 13s ago
    Process: 2685 ExecStart=/opt/tomcat/latest/bin/startup.sh (code=exited, status=0/SUCCESS)
   Main PID: 2692 (java)
      Tasks: 29 (limit: 9450)
     Memory: 154.5M
     CGroup: /system.slice/tomcat.service
             └─2692 /usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/latest/conf/lc

Apr 05 12:40:35 pega systemd[1]: Starting Tomcat 9 servlet container...
Apr 05 12:40:35 pega startup.sh[2685]: Tomcat started.
Apr 05 12:40:35 pega systemd[1]: Started Tomcat 9 servlet container.
pega@pega:~$
pega@pega:~$ sudo systemctl enable tomcat
Created symlink /etc/systemd/system/multi-user.target.wants/tomcat.service → /etc/systemd/system/tomcat.service.
pega@pega:~$ sudo ufw allow 8080/tcp
Rules updated
Rules updated (v6)
pega@pega:~$
pega@pega:~$
pega@pega:~$
```

Update permissions of all key tomcat files/folders…

sudo usermod -a -G tomcat pega
sudo chmod g+w /opt/tomcat/latest/bin
sudo chmod g+w /opt/tomcat/latest/lib
sudo chmod g+w /opt/tomcat/latest/conf
sudo chmod g+w /opt/tomcat/latest/conf/*
sudo chmod g+w /opt/tomcat/latest/webapps

Upload config files to target locations via scp from host machine or upload via sftp client such as **FileZilla** into target locations.

```
scp postgresql-42.2.14.jar pega@192.168.2.159:/opt/tomcat/latest/lib
scp context.xml pega@192.168.2.159:/opt/tomcat/latest/conf
scp tomcat-users.xml pega@192.168.2.159:/opt/tomcat/latest/conf
scp setenv.sh pega@192.168.2.159:/opt/tomcat/latest/bin
scp prweb.war pega@192.168.2.159:/opt/tomcat/latest/webapps
```

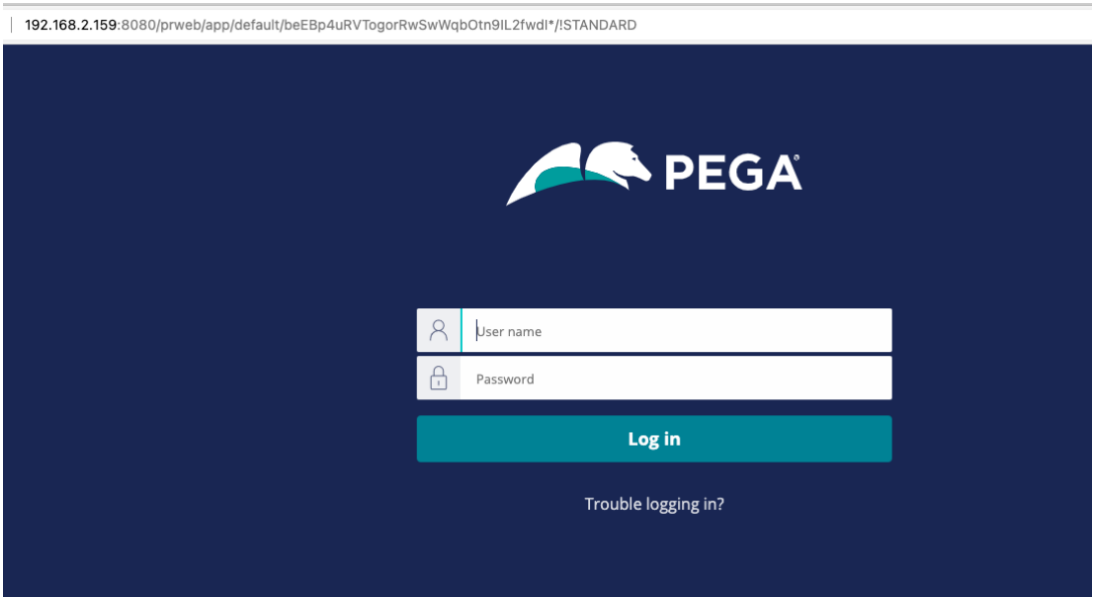
```
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$ scp context.xml pega@192.168.2.159:/opt/tomcat/latest/conf
pega@192.168.2.159's password:
context.xml
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$ scp tomcat-users.xml pega@192.168.2.159:/opt/tomcat/latest/conf
pega@192.168.2.159's password:
tomcat-users.xml
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$ scp setenv.sh pega@192.168.2.159:/opt/tomcat/latest/bin
pega@192.168.2.159's password:
setenv.sh
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasa$
```

6. Final configs and start Pega Server

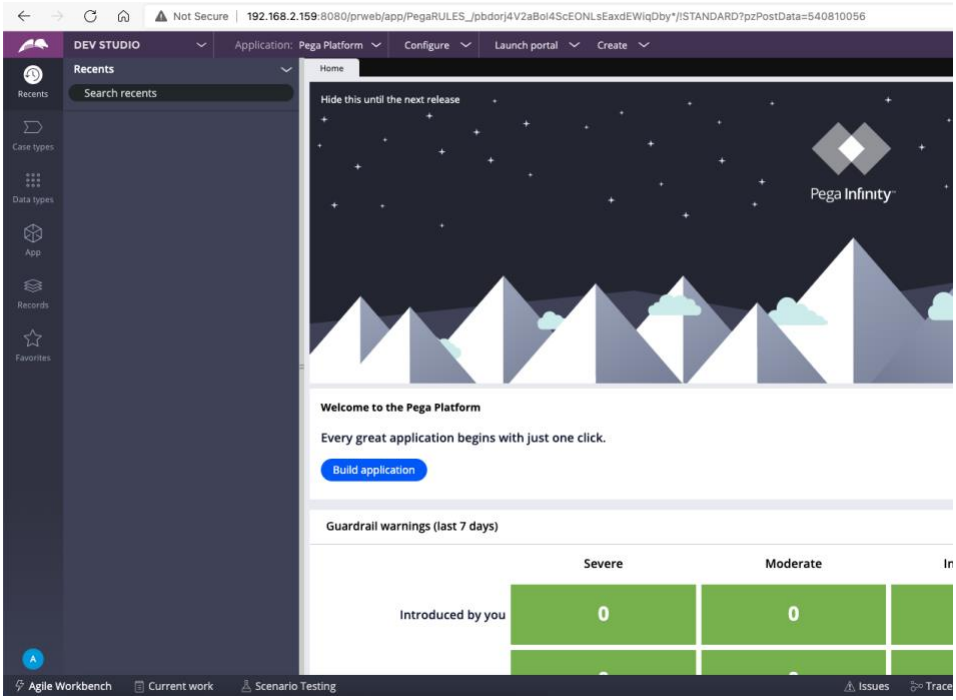
Final permission updates for tomcat group with below commands…

```
sudo chown -R tomcat:tomcat /opt/tomcat/*
sudo chown -R tomcat:tomcat /pega
```

Restart system and check logs at /pega/logs. It should start tomcat and Pega. Access it via http://192.168.2.159:8080/prweb/



Administrator@pega.com / install



Happy adventure… 😊