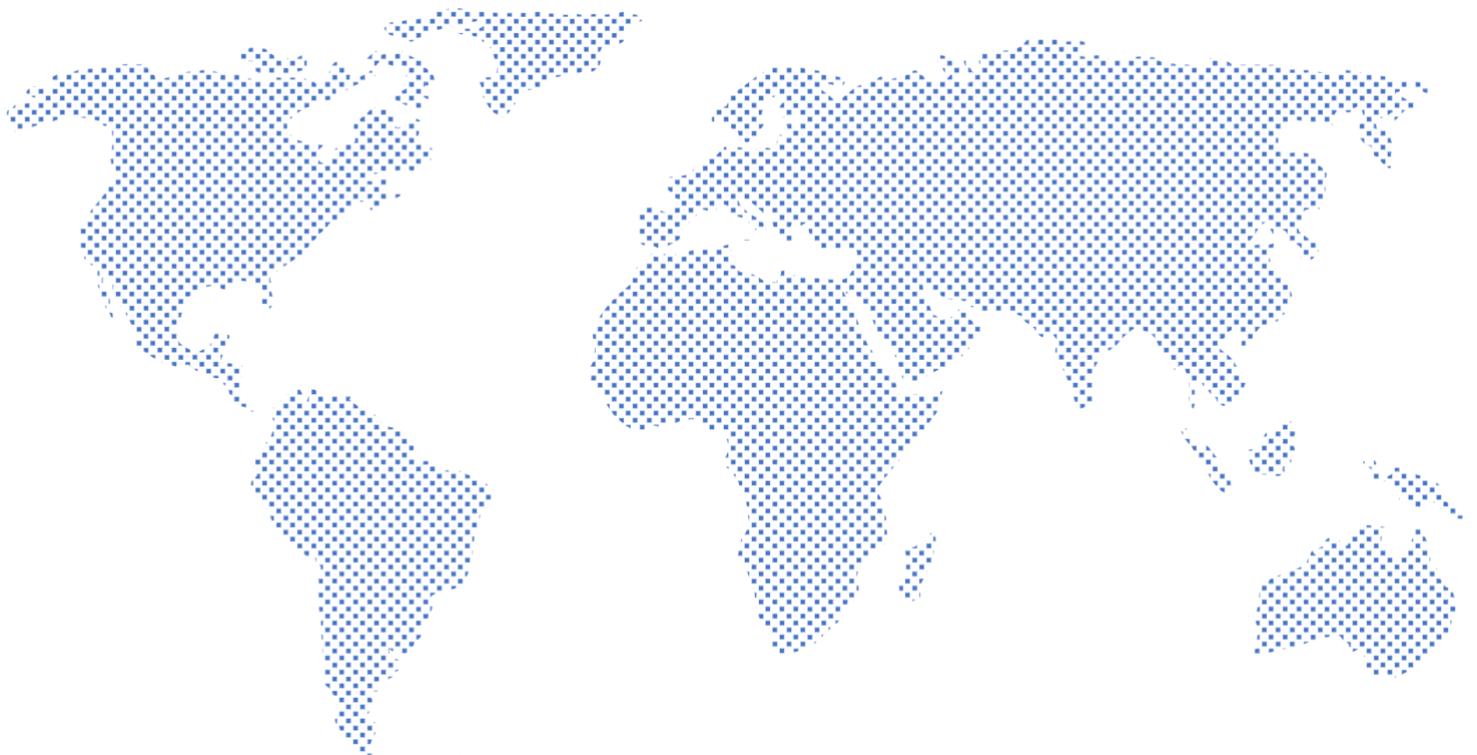


Pega Personal Edition Install on Ubuntu Server



SRINIVASA KANDRU

Sreesoft Solutions



Contents

About this document.....	3
Disclaimer & Licensing Terms.....	3
Check Prerequisites.....	3
1. Extract Pega Database Backup from PE (Personal Edition).....	4
2. Pull GitHub repository for necessary config files	6
3. Create Ubuntu Server Shell	7
4. Install Postgres Database	19
4.1 Export Backed-up Pega DB file via host 'psql' utility.....	21
5. Install Tomcat Server.....	22
6. Final configs and start Pega Server	25

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>

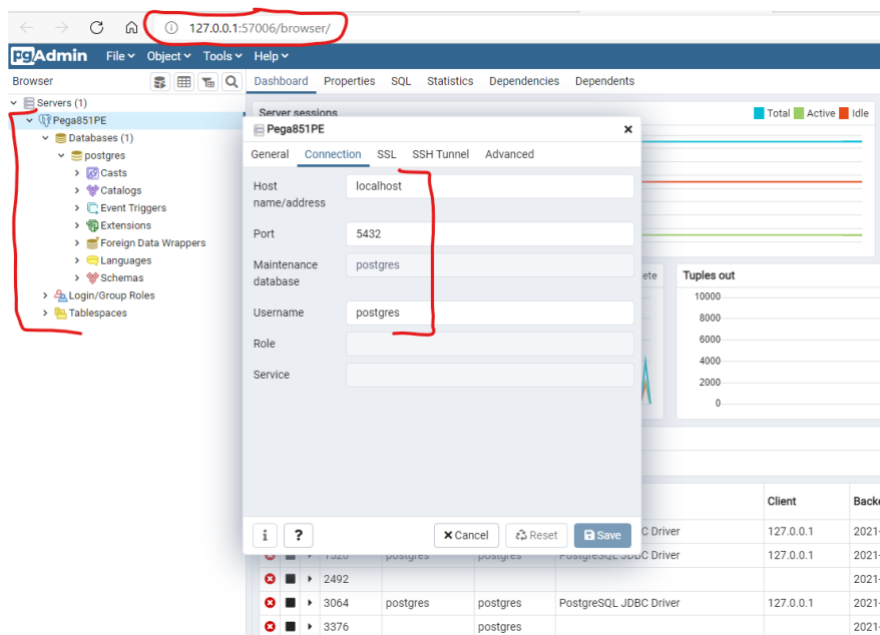
```
srinivasa@srinivasa:~$ ssh-keygen -t rsa -b 4096 -C "Srinivasa@srinivasa.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/srinivasa/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/srinivasa/.ssh/id_rsa
Your public key has been saved in /home/srinivasa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:vHL04q9Bz/r02EWG1HB5rzXLs+keSFUQesLd/z4ANls Srinivasa@srinivasa.com
The key's randomart image is:
+---[RSA 4096]-----+
|  . .oo. |
|  =o... |
|  =oo. |
|  . =E..+ |
|  . S o B..o+ |
|  . o . +..o+ |
|  o *   ....= |
|  .O + .   = |
|  .+* o   ooo |
+---[SHA256]-----+
srinivasa@srinivasa:~$
```

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

```
PS C:\Program Files\pgAdmin 4\v4\runtime> pwd
Path
----
C:\Program Files\pgAdmin 4\v4\runtime

PS C:\Program Files\pgAdmin 4\v4\runtime> ls

Directory: C:\Program Files\pgAdmin 4\v4\runtime

Mode                LastWriteTime         Length Name
----                -
d----- 07/02/2021 20:55          imageformats
d----- 07/02/2021 20:55          platforms
-a----- 18/01/2021 15:03        15360 comerr64.dll
-a----- 18/01/2021 15:03        384512 gssapi64.dll
-a----- 18/01/2021 15:03        64000 K5sprt64.dll
-a----- 18/01/2021 15:05        27136 kinit.exe
-a----- 18/01/2021 15:03       1210368 krb5_64.dll
-a----- 01/10/2020 13:34       2840064 libcrypto-1_1-x64.dll
-a----- 26/11/2020 15:30       301056 libpq.dll
-a----- 01/10/2020 13:34       675840 libssl-1_1-x64.dll
-a----- 25/01/2021 14:39       588288 pgAdmin4.exe
-a----- 26/11/2020 15:39       458704 pg_dump.exe
-a----- 26/11/2020 15:39       136192 pg_dumpall.exe
-a----- 26/11/2020 15:39       213504 pg_restore.exe
-a----- 26/11/2020 15:39       499200 psql.exe
-a----- 23/09/2020 17:00       100368 python.exe
-a----- 23/09/2020 17:00       4205584 python38.dll
-a----- 23/09/2020 17:00       98832 pythonw.exe
-a----- 25/01/2021 14:39         28 qt.conf
-a----- 27/03/2020 13:18       6193272 Qt5Core.dll
-a----- 27/03/2020 13:18       7129720 Qt5Gui.dll
-a----- 27/03/2020 13:18       1381496 Qt5Network.dll
-a----- 27/03/2020 18:41       375520 Qt5Svg.dll
-a----- 27/03/2020 13:18       5596280 Qt5Widgets.dll
-a----- 29/05/2020 13:24       84992 rlib.dll

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

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/pegaubuntumv>

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/pegaubuntumv.git](https://github.com/sreesoft/pegaubuntumv)

```
[Srinivasas-MBP:GitHub Srinivasa$ git clone git@github.com:sreesoft/pegaubuntumv.git]
Cloning into 'pegaubuntumv'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), 1.61 MiB | 2.14 MiB/s, done.
[Srinivasas-MBP:GitHub Srinivasa$ ll]
total 0
drwxr-xr-x 13 Srinivasa staff 416 8 Apr 15:37 pegaubuntumv
[Srinivasas-MBP:GitHub Srinivasa$ cd pegaubuntumv/]
[Srinivasas-MBP:pegaubuntumv Srinivasa$ ll]
total 4024
-rwxr-xr-x 1 Srinivasa staff 3549 8 Apr 15:37 context.xml
-rwxr-xr-x 1 Srinivasa staff 4958 8 Apr 15:37 pg_hba.conf
-rwxr-xr-x 1 Srinivasa staff 340436 8 Apr 15:37 pljava.jar
-rwxr-xr-x 1 Srinivasa staff 329672 8 Apr 15:37 pljava.so
-rwxr-xr-x 1 Srinivasa staff 932808 8 Apr 15:37 postgresql-42.2.14.jar
-rwxr-xr-x 1 Srinivasa staff 27013 8 Apr 15:37 postgresql.conf
-rwxr-xr-x 1 Srinivasa staff 395800 8 Apr 15:37 prweb.war
-rwxr-xr-x 1 Srinivasa staff 313 8 Apr 15:37 setenv.sh
-rwxr-xr-x 1 Srinivasa staff 2320 8 Apr 15:37 tomcat-users.xml
-rwxr-xr-x 1 Srinivasa staff 628 8 Apr 15:37 tomcat.service
[Srinivasas-MBP:pegaubuntumv Srinivasa$]
```

4. It creates a folder 'pegaubuntumv' 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

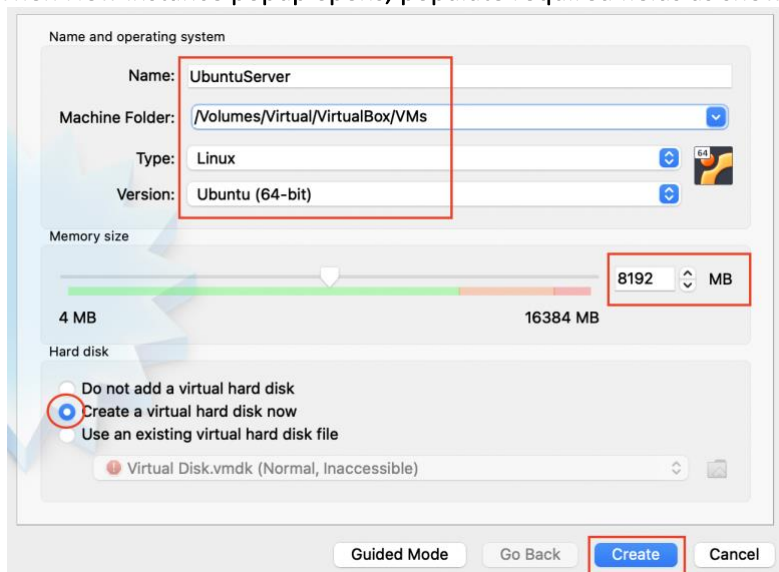
Download the latest Ubuntu Server ISO image from Ubuntu site. (At the time of this document, it is 20.04 version). Downloads of Ubuntu ISO are available at [Get Ubuntu Server | Download | Ubuntu](#)

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

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

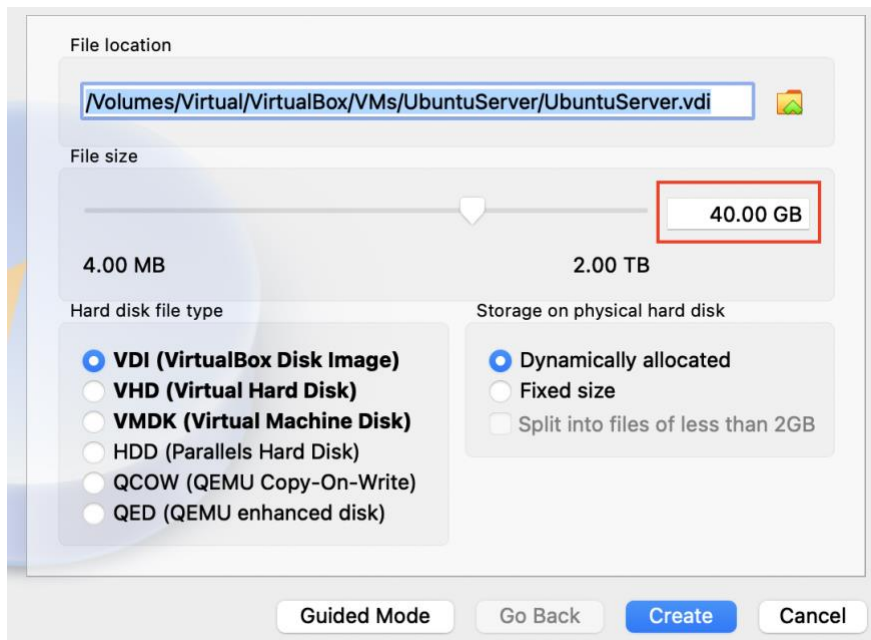


When New instance popup opens, populate required fields as shown below...



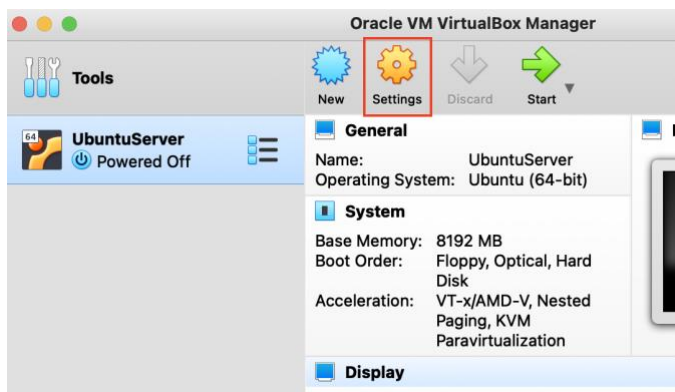
- Used Memory size as 8192MB, use as per your resource availability. More is better.
- Choose creating a virtual hard disk to expand it when needed.
- Choose Machine Folder as per target system location.
- Click on 'Create' button.

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

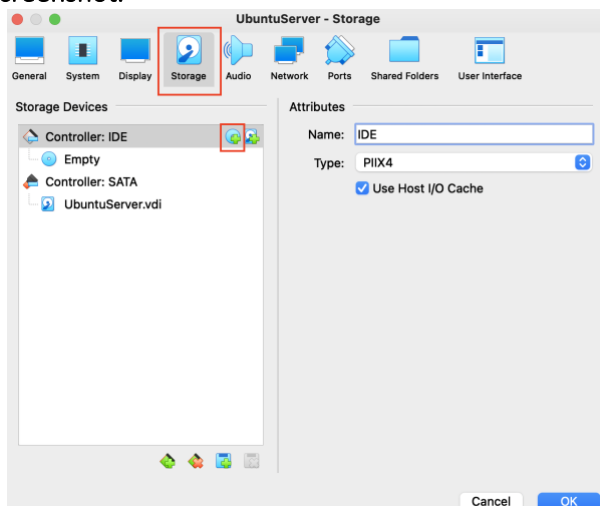


Specify at-least 40GB of disk space and Dynamically allocated.
Click on 'Create' button

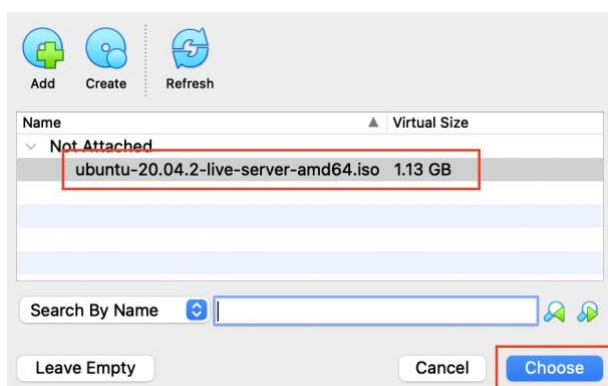
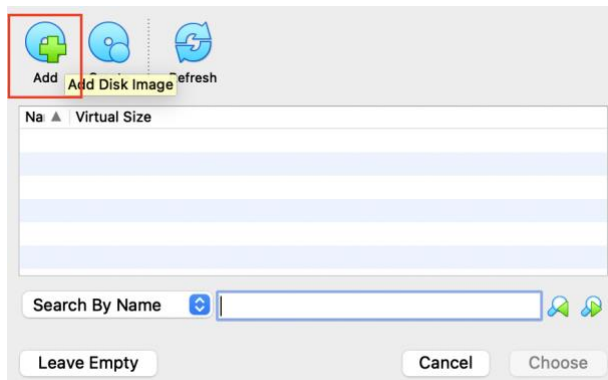
When popup completed/closed, click on 'Settings' icon on dashboard of VirtualBox.



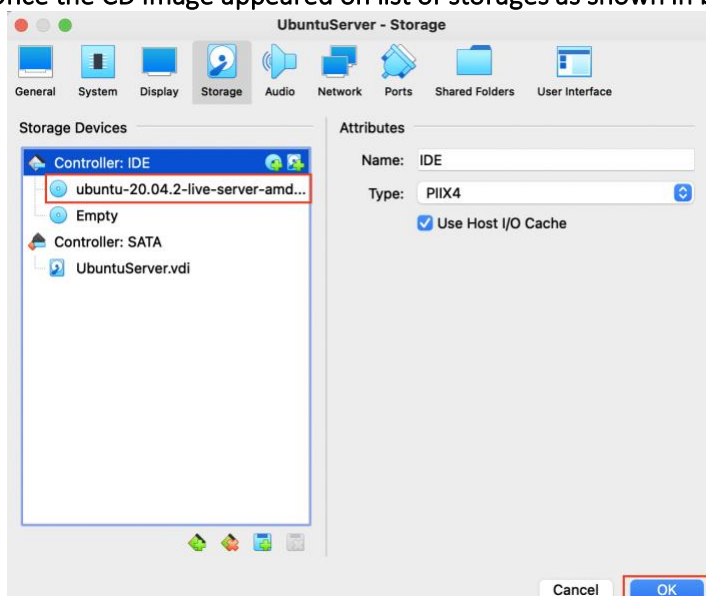
On the popup opened, select 'Storage' and add a CD image by selecting the CD icon as shown in below screenshot.



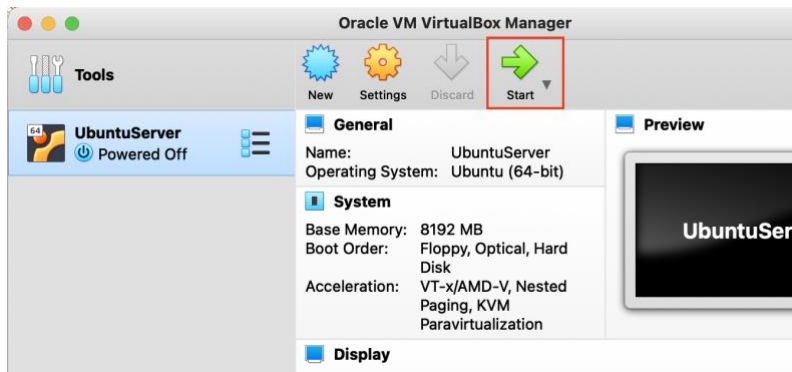
Click on 'Add' button as shown in below screenshot to choose the Ubuntu ISO downloaded earlier.



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

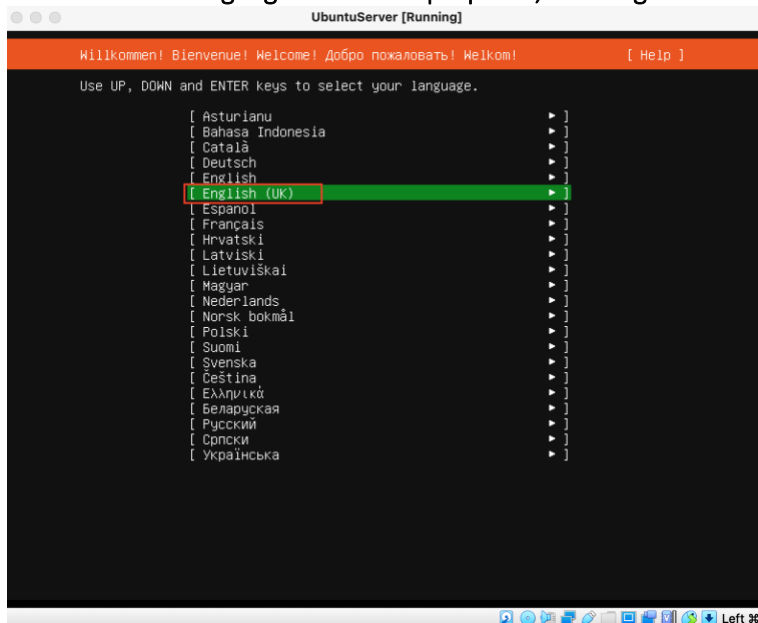


On Dashboard, click on Start icon to start the VM instance using the CD ISO image of Ubuntu Server.

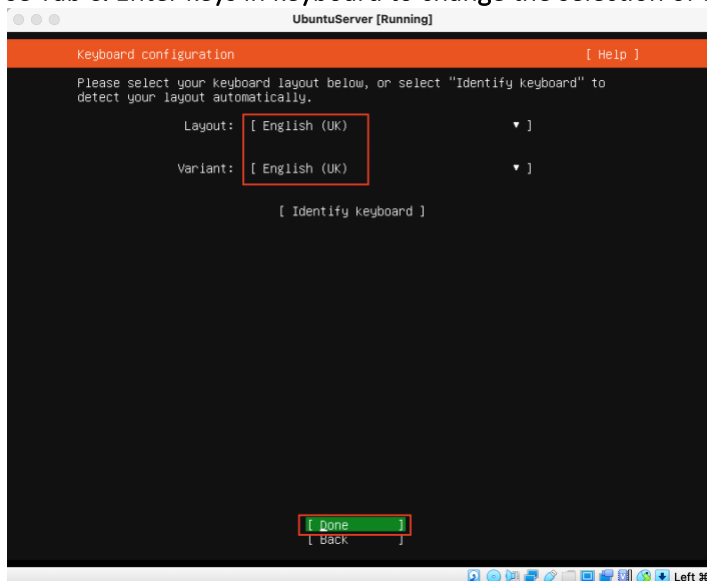


Follow onscreen instructions to install Ubuntu Server. Key steps are explained below with screenshots.

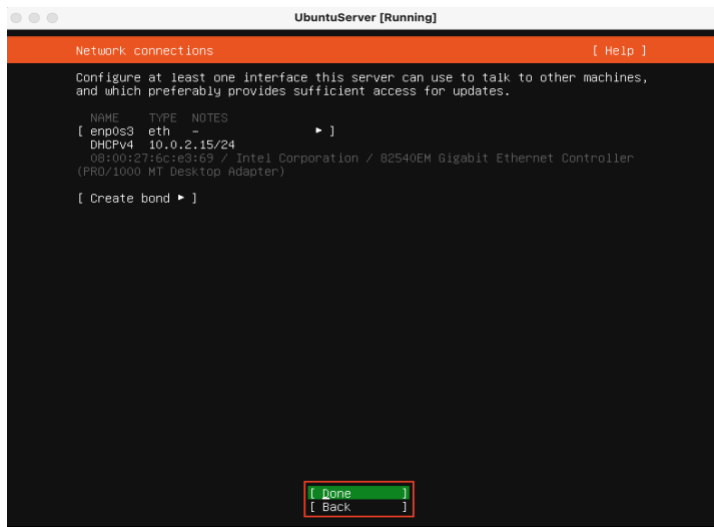
Choose relevant language. For demo purposes, it is 'English UK'.



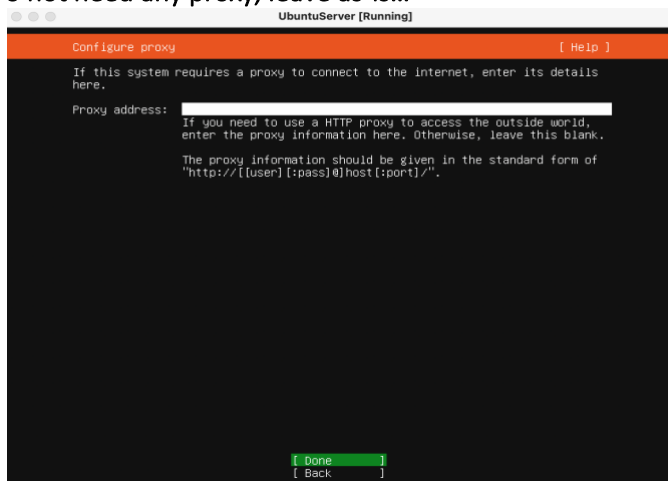
Use Tab & Enter keys in keyboard to change the selection of Layout & Variant.



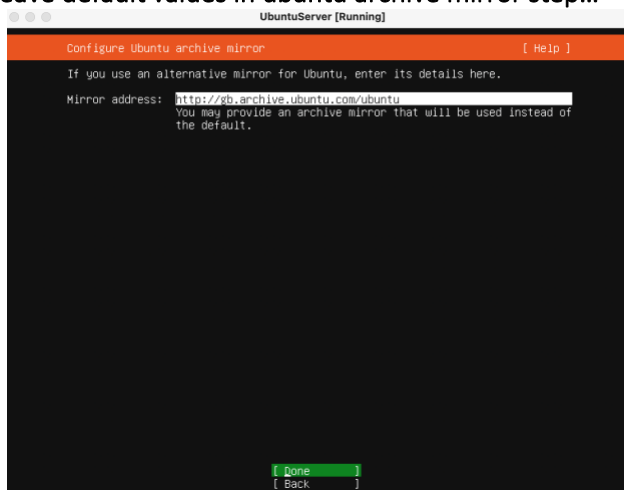
Leave network connections to defaults...



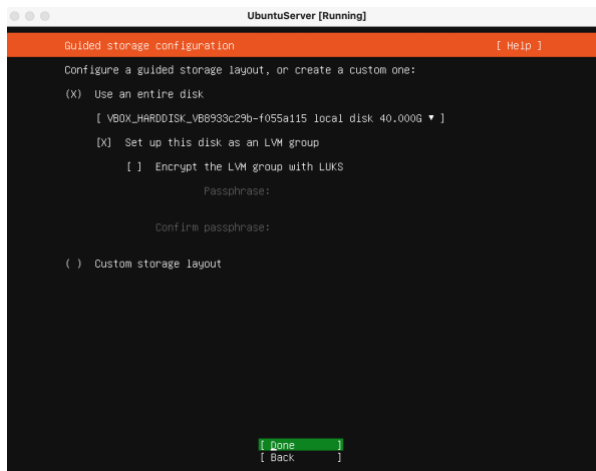
Do not need any proxy, leave as-is...



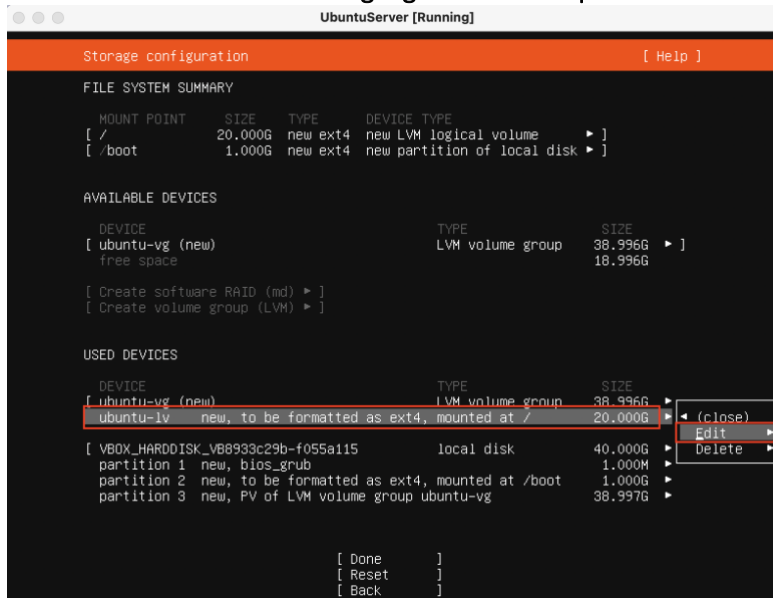
Leave default values in ubuntu archive mirror step...



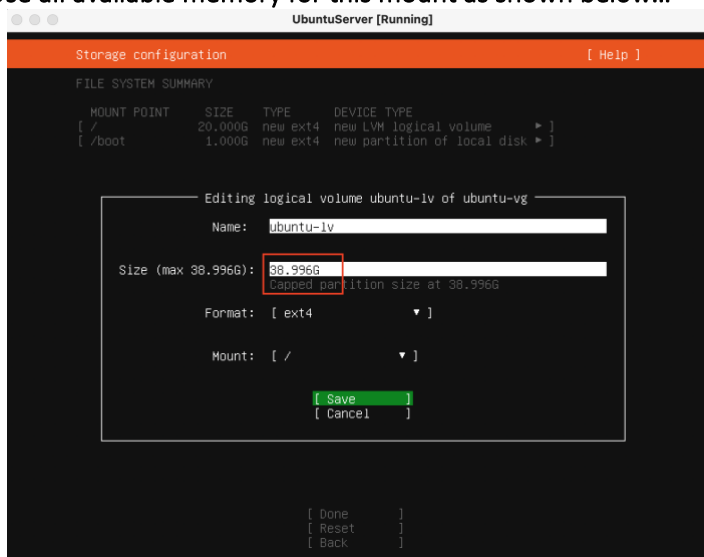
Choose entire disk option as below...



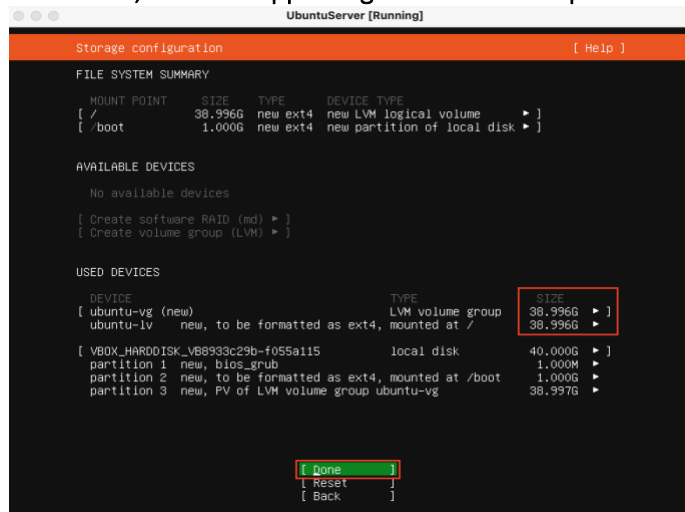
When configuring storage, map all available memory into mount point as shown below. Use Tab & Enter to select the highlighted mount option and 'Edit'...



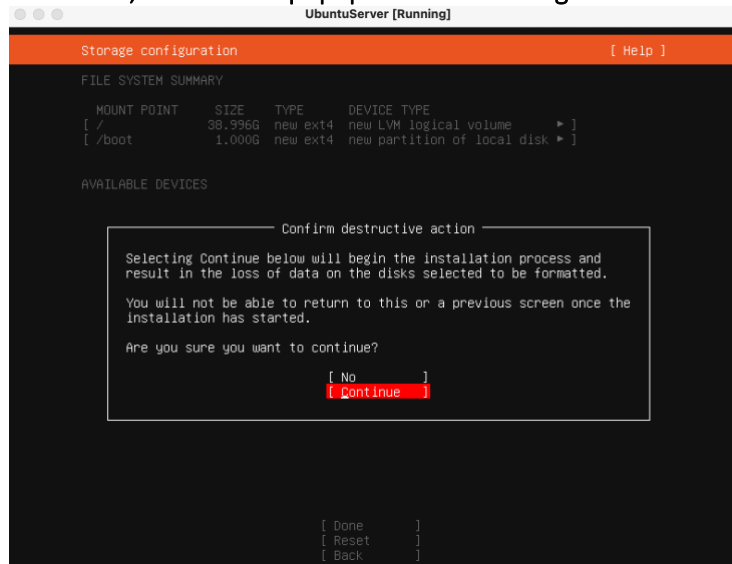
Use all available memory for this mount as shown below...



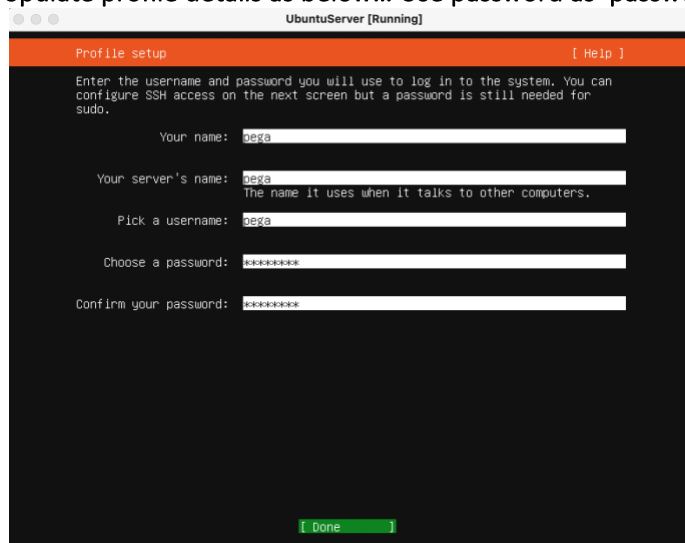
When saved, it should appear against the mount point.



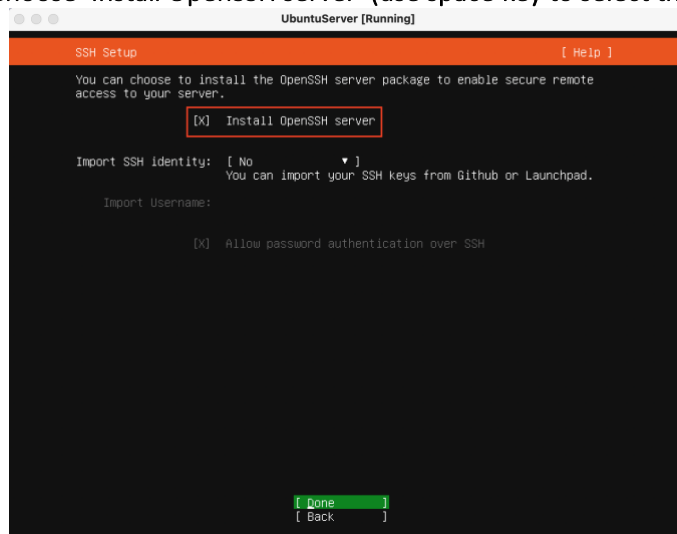
When Done, confirm the popup to use the configuration set.



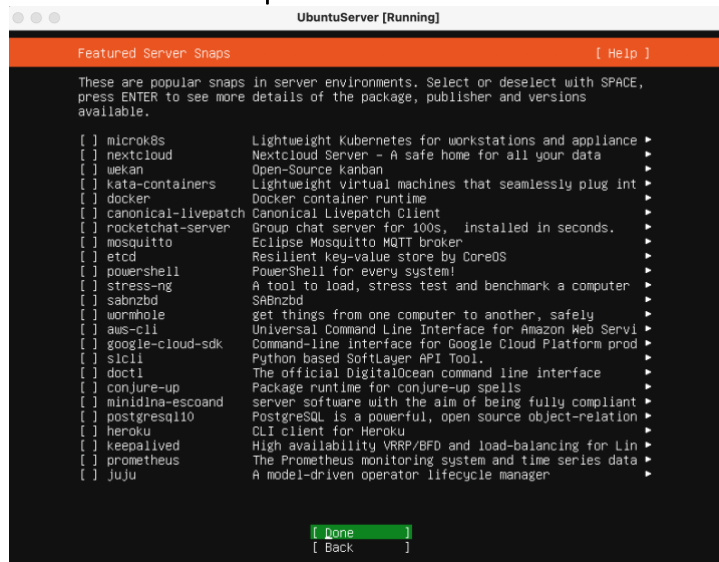
Populate profile details as below... Use password as 'password'



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.

```
UbuntuServer [Running]

Install complete! [ Help ]

running '/snap/bin/subiquity.subiquity-configure-apt
/snap/subiquity/2280/usr/bin/python3 true'
curtin command apt-config
curtin command in-target
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
installing missing packages
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
configuring target system bootloader
installing grub to target devices
finalizing installation
running 'curtin hook' /
final system configuration
configuring cloud-init
installing openssh-server
restoring apt configuration
downloading and installing security updates
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

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

```
UbuntuServer [Running]

pega login: [ 15.363913] cloud-init[1424]: Generating locales (this might take a while)...
[ 16.611596] cloud-init[1424]: en_GB.UTF-8... done
[ 16.611696] cloud-init[1424]: Generation complete.
[ 16.887918] cloud-init[1424]: Cloud-init v. 20.4.1-0ubuntu1~20.04.1 running 'modules:config' at S
un, 04 Apr 2021 19:12:54 +0000. Up 15.21 seconds.
ci-info: no authorized SSH keys fingerprints found for user pega.
<14>Apr 4 19:12:56 ec2:
<14>Apr 4 19:12:56 ec2: #####
<14>Apr 4 19:12:56 ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Apr 4 19:12:56 ec2: 1024 SHA256:v+gL25A3AFEEtJcSkJZPQim8hsQeANV1o3juctbXCs root@pega (DSA)
<14>Apr 4 19:12:56 ec2: 256 SHA256:QLB5RSs/HZTMom6bN1mImaPIHkvamG2v+Zuc8GGfEXU root@pega (ECDSA)
<14>Apr 4 19:12:56 ec2: 256 SHA256:d2Jk3/qf2HbmEUuN4IGMS+11G5Vs3nY168B226dk16Q root@pega (ED25519)
<14>Apr 4 19:12:56 ec2: 3072 SHA256:WCVSS2vbHkhR+K8Kjo+rSrKnR+9vbhVnkucNxvCofYc root@pega (RSA)
<14>Apr 4 19:12:56 ec2: -----END SSH HOST KEY FINGERPRINTS-----
<14>Apr 4 19:12:56 ec2: #####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHsbnogknmmFHC/ofKDRnA2vLA9
D9N05uuumuhyd+LOYktRdPMXcgYje1BRHLnFzln7Ss7hGuSvdYd/yXRdDcU= root@pega
ssh-ed25519 AAAAC3NzaC1l2D1lNTE5AAAAI0vIXLTMbJNFgAKMAzd73E1h/+moEUvDeA0cy7nXYBg7 root@pega
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC6p4Fck1hkth0CF0T0wkEwXus6fQP6/1A9Fv2y1s0pvu3wPpKHTCzTqptKeRY
+YhhFXEo2uaILNs0dCHes2M0VA1l+v2F191N1ax1ssUXdC0z5HVZwpu+R7Bc191tU0bt32H05LmeL09BWHC4NJD4h6Dhr1hzqf
965gHQ12CF02Q1/4xT2erfoXn002pTjpaaya13vRj1MqrpsHA11F1F1T2+eS121IkHmtAJMtWuFm75kLVFL1UC1Xc3r1qYQnvvQ
16Tjbl+ghV6Bcs94s03S81Qf74XmCdV0Dre0E056n12t75132hs7HAMkks12JPCENvH4v1ZJ/ZNhY1kgnd0mMkvK8mMgRAU1uH
1xkM1KtU14J1JpIqghux6FJuubdG0eGy9rPus1w5IrNPtW2tKxq2v/QcmnaeSayFvbM9+5T62dSBKMIN63ucLn7PrsC2Eurmfc
6n34upJ2DX5NffTYQH7f+Xn4sXmsm3Jn5Zgw1ffUF4U4/8Dc680= root@pega
-----END SSH HOST KEY KEYS-----
[ 17.312297] cloud-init[1471]: Cloud-init v. 20.4.1-0ubuntu1~20.04.1 running 'modules:final' at S
n, 04 Apr 2021 19:12:56 +0000. Up 17.20 seconds.
[ 17.312389] cloud-init[1471]: ci-info: no authorized SSH keys fingerprints found for user pega.
[ 17.312441] cloud-init[1471]: Cloud-init v. 20.4.1-0ubuntu1~20.04.1 finished at Sun, 04 Apr 2021
19:12:56 +0000. DataSource DataSourceNone. Up 17.30 seconds
[ 17.312511] cloud-init[1471]: 2021-04-04 19:12:56,982 - cc_final_message.py[WARNING]: Used failb
ck datasource

pega login: pega
Password: _
```

On successful login screen appears as below...


```
UbuntuServer [Running]
Ubuntu 20.04.2 LTS pega tty1
pega login: pega
Password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-70-generic x86_64)

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

System information as of Sun  4 Apr 19:17:18 UTC 2021

System load:  0.01          Processes:           98
Usage of /:   16.0% of 38.13GB   Users logged in:    0
Memory usage: 2%            IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%

29 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

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.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

pega@pega:~$
```

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

```
sudo apt-get update
```

```
pega@pega:~$ sudo apt-get update
[sudo] password for pega: _
```

```
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.

```

    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

```

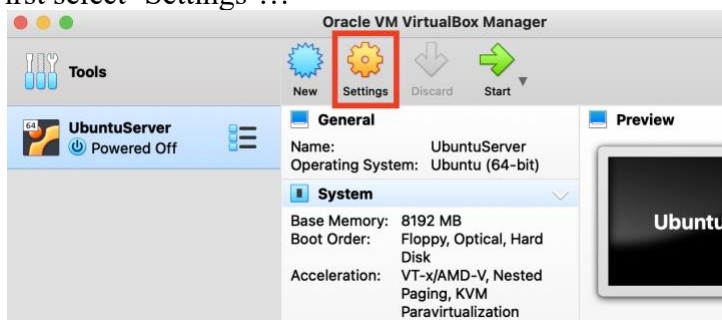
Now restart the system using the below command.

```

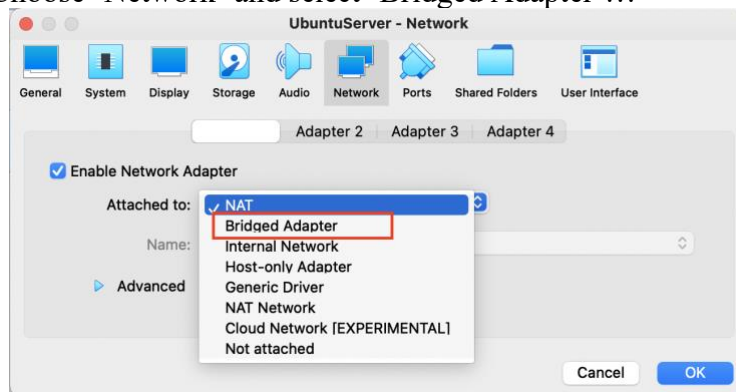
shutdown -h now
pega@pega:~$
pega@pega:~$
pega@pega:~$
pega@pega:~$ shutdown -h now

```

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.

```
Last login: Sun Apr  4 19:17:19 UTC 2021 on tty1
pega@pega:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.159 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::a00:27ff:fe6c:e369 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:6c:e3:69 txqueuelen 1000 (Ethernet)
    RX packets 21 bytes 2455 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 1328 (1.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 80 bytes 5920 (5.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 80 bytes 5920 (5.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pega@pega:~$ _
```

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')
```

```

Srinivasas-MBP:Downloads Srinivasas$ ssh pega@192.168.2.159
The authenticity of host '192.168.2.159 (192.168.2.159)' can't be established.
ECDSA key fingerprint is SHA256:QLB5Rss/HZTMom6bNim1maPIHkvamG2v+Zuc8GGfEXU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.159' (ECDSA) to the list of known hosts.
pega@192.168.2.159's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-70-generic x86_64)

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

System information as of Sun  4 Apr 19:38:51 UTC 2021

System load:  0.06          Processes:            105
Usage of /:   17.3% of 38.13GB Users logged in:       1
Memory usage: 2%           IPv4 address for enp0s3: 192.168.2.159
Swap usage:   0%

29 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

```

```

Last login: Sun Apr  4 19:36:18 2021
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
```

```
--1W-1W-1-- 1 root postgres 10424 Feb 10 10:47 uu10-0SSP.SU
[pega@pega:/usr/lib/postgresql/12/lib$ sudo chown -R root:postgres /etc/postgresql
[pega@pega:/usr/lib/postgresql/12/lib$ sudo chmod -R g+rw /etc/postgresql
[pega@pega:/usr/lib/postgresql/12/lib$
```

```
-----
[pega@pega:/$ sudo ln -s /usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64/server/libjvm.so /lib/libjvm.so
[[sudo] password for pega:
[pega@pega:/$ cd /lib
```

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/
```

```
scp: /usr/lib/postgresql/12/lib/pljava.so: permission denied
[Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$ scp ./pljava.so pega@192.168.2.159:/usr/lib/postgresql/12/lib/
pega@192.168.2.159's password:
pljava.so
[Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$
-----
scp: /etc/postgresql/12/main/pg_hba.conf: permission denied
[Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$ scp ./pg_hba.conf pega@192.168.2.159:/etc/postgresql/12/main/
pega@192.168.2.159's password:
pg_hba.conf
[Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$
[Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$
```

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'.

Create - Server

General

Connection

SSL

SSH Tunnel

Advanced

Host name/address

192.168.2.159

Port

5432

Maintenance database

postgres

Username

postgres

Password

.....

Save password?

☒

Role

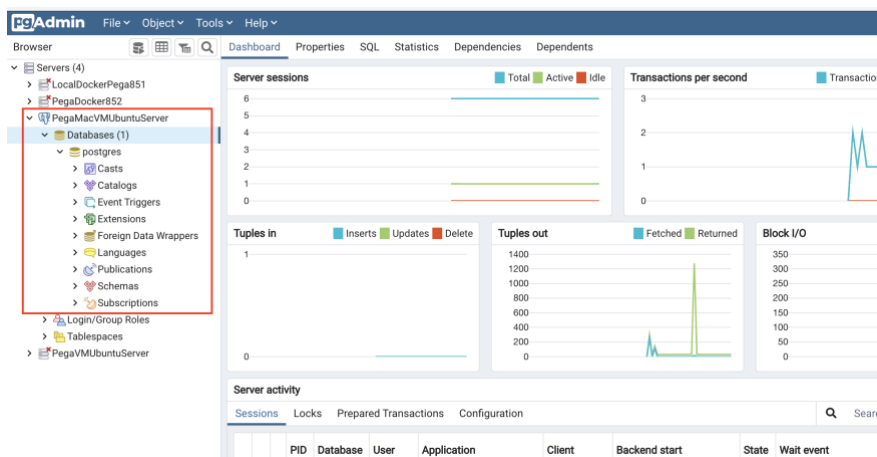
Service

?

Cancel

Reset

Save



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"
```

```
Srinivasas-MBP:~$ cd /Applications/pgAdmin/ 4.app/Contents/SharedSupport/
Srinivasas-MBP:SharedSupport Srinivasas$ ll
total 3384
-rwxr-xr-x@ 1 Srinivasa  admin  551456 23 Mar 14:01 pg_dump
-rwxr-xr-x@ 1 Srinivasa  admin  147408 23 Mar 14:01 pg_dumpall
-rwxr-xr-x@ 1 Srinivasa  admin  259312 23 Mar 14:01 pg_restore
-rwxr-xr-x@ 1 Srinivasa  admin   768224 23 Mar 14:01 psql
Srinivasas-MBP:SharedSupport Srinivasas$ ./psql --host "192.168.2.159" --port "5432" --username "postgres" --password --dbname "postgres" --file "/Users/Srinivasa/Downloads/PegaPE_851_DB_Postgres.sql"
Password:
SET
SET
SET
SET
```

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
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
GRANT
GRANT
GRANT
GRANT
GRANT
Srinivasas-MBP:SharedSupport Srinivasas$
```

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](https://downloads.apache.org/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz) 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:~$
```

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

```

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:~$ █

```

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:~$ 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:~$ 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:~$ █

```

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
```

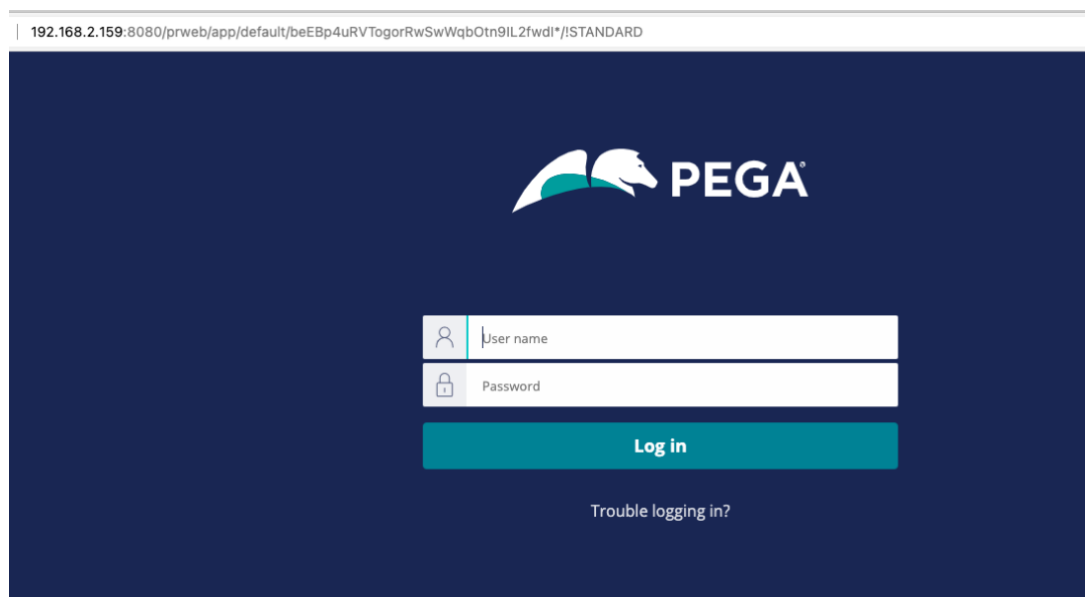
```
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$ scp context.xml pega@192.168.2.159:/opt/tomcat/latest/conf
pega@192.168.2.159's password:
context.xml
Srinivasas-MBP:Pega_UbuntuServer_Install Srinivasas$ scp tomcat-users.xml pega@192.168.2.159:/opt/tomcat/latest/conf
pega@192.168.2.159's password: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

6. Final configs and start Pega Server

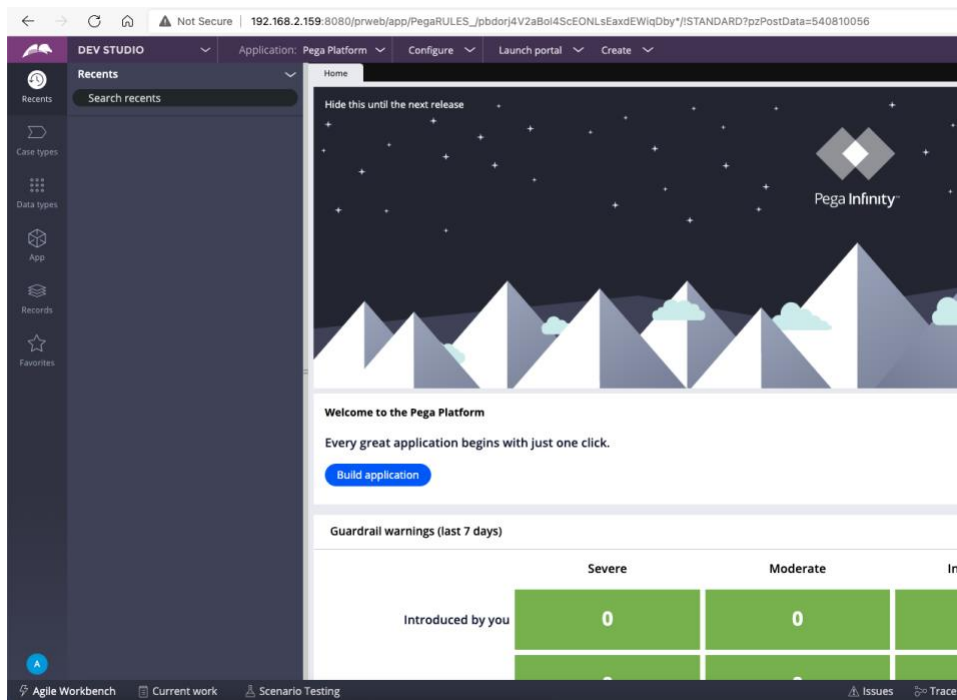
Final permission updates for tomcat group with below commands...

```
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... 😊