

Step1 :

Download VMware-workstation-player 15.x or higher

<https://www.vmware.com/in/products/workstation-player/workstation-player-evaluation.html>

Clickon

**Try Workstation 15.5 Player for Windows
Download Now**

Step2:

Install VMware-workstation-player 15.x which you downloaded

Step 3: Getting VM of Centos 7:

<https://drive.google.com/file/d/1mAONTM232DVx352-TDT1cku2Hhql62NX/view?usp=sharing>

Download

Step 4:

Create my-vm folder on your windows/linux machine and extract this downloaded vm twice to the same my-vm folder. So there will be two VMs in two different folders.

If you want change the folders name to machine1 and machine2

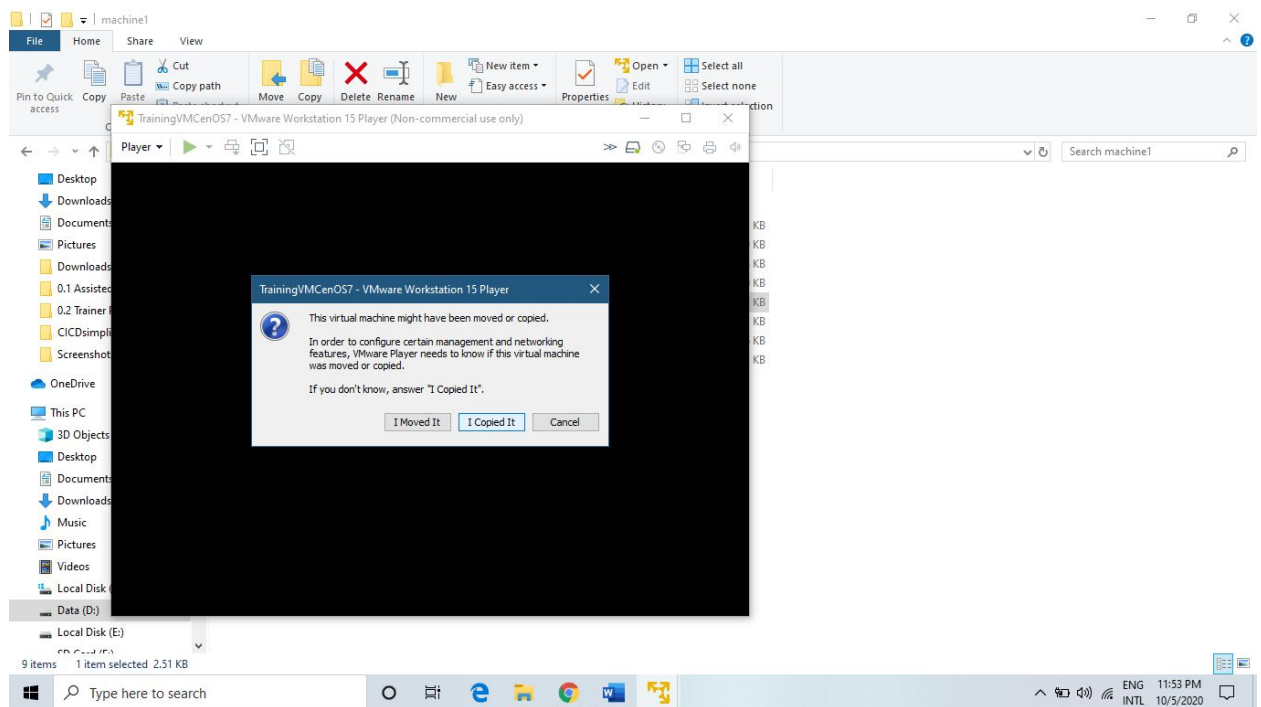
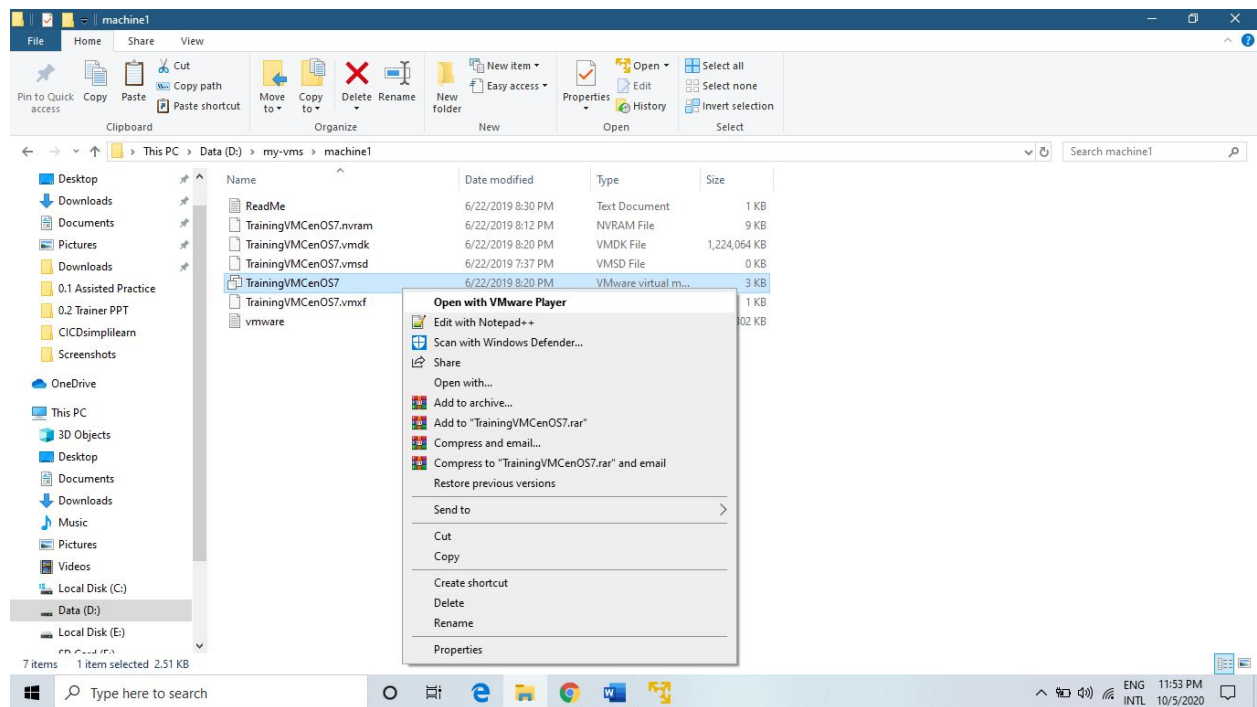
Step 5: Read readme file from machine1 and machine2 folders of my-vm to get username and password to login

Step6: start both VMs

Open machine1 folder >> right click on >>VMware virtual machine configuration (.vmx) (3KB sized file) >> SELECT "Open with VMWare Player"

Let the vm start

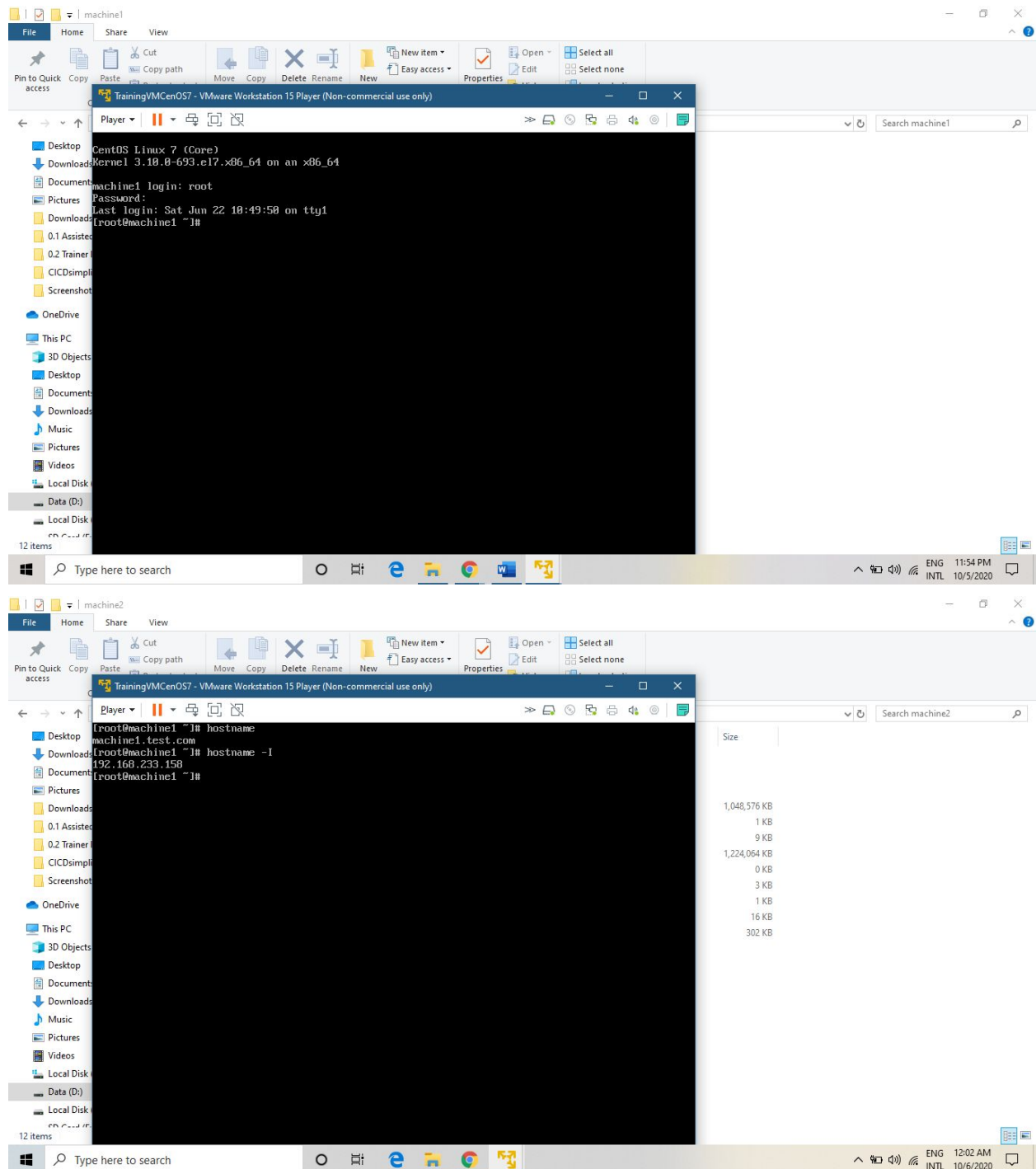
Select "I Copied IT" from pop-up



Open machine2 folder >> right click on >>VMware virtual machine configuration (.vmx)
 (3KB sized file) >> SELECT "Open with VMWare Player"
 Let the vm start
 Select "I Copied IT" from pop-up

Step7 :
 Login on both VMS by user: root password: redhat

NOTE: to take cursor out of vm box need to press Ctrl+Alt & click inside the VM box if want to type anything



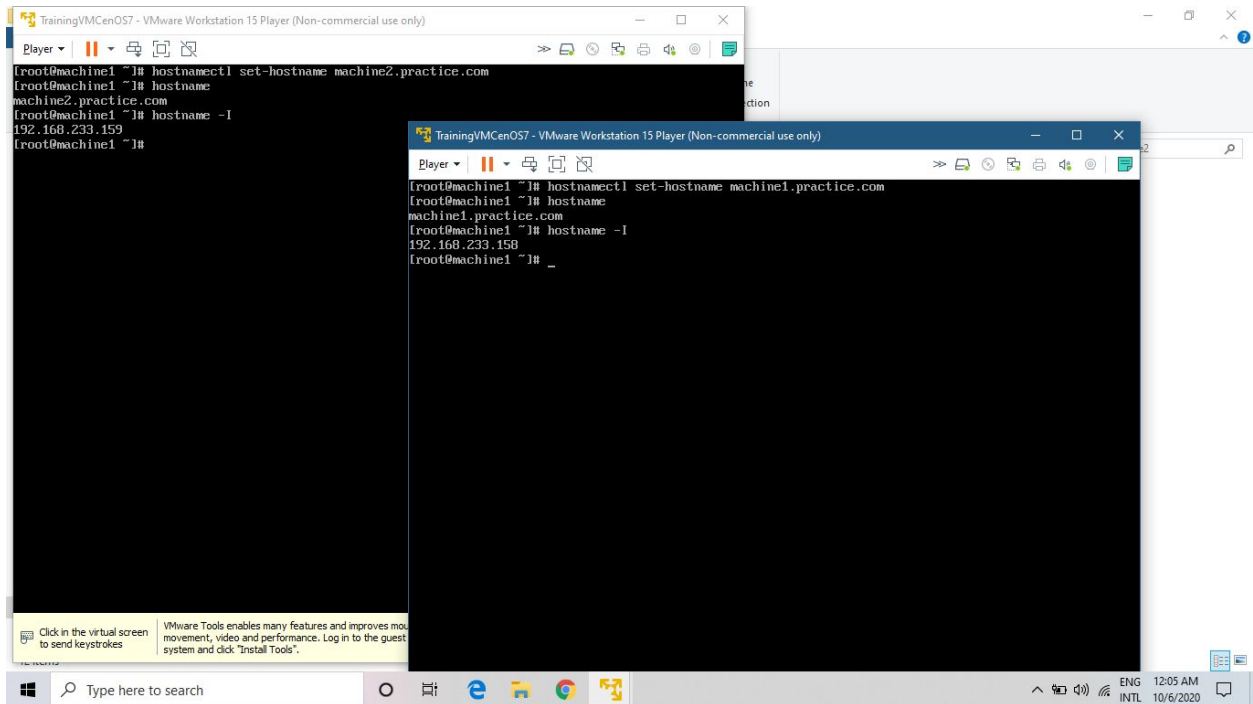
NOW BOTH MACHINES ARE READY :
Lets do the configurations now

Lets change hostnames and note down IP address as well
On machine1

```
# hostnamectl set-hostname machine1.practice.com  
# hostname  
#hostname -I
```

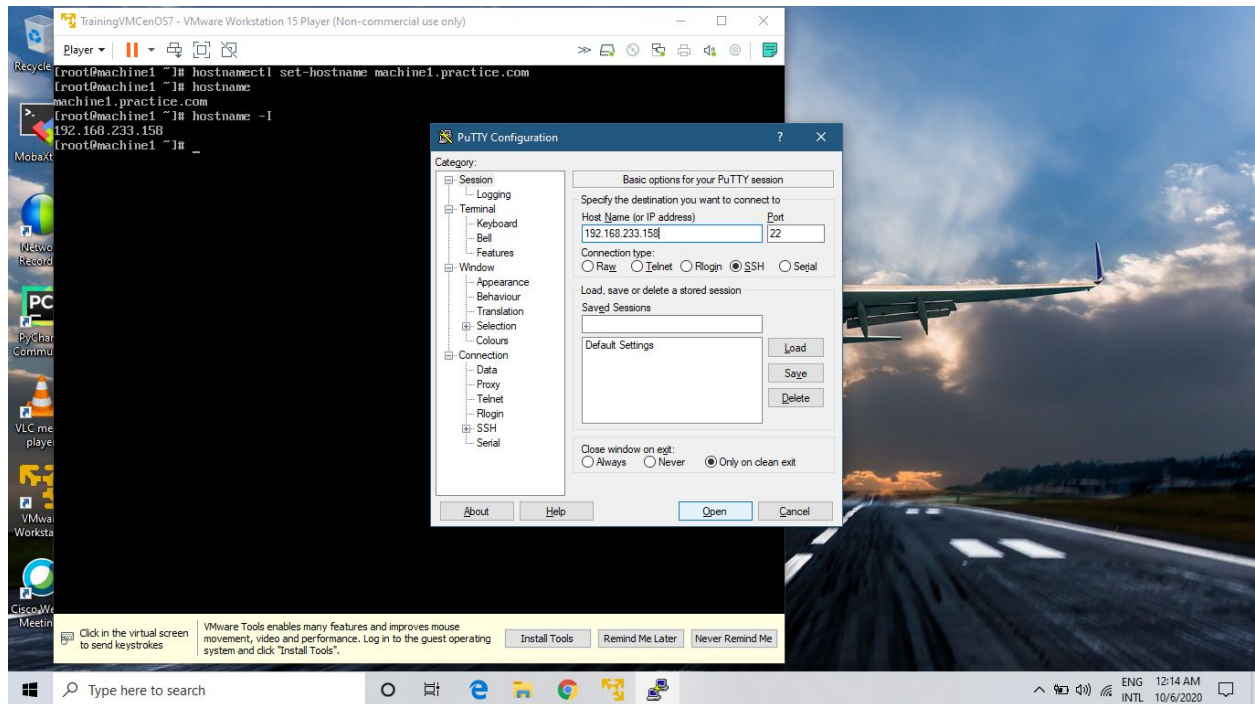
On machine2

```
# hostnamectl set-hostname machine2.practice.com  
# hostname  
#hostname -I
```

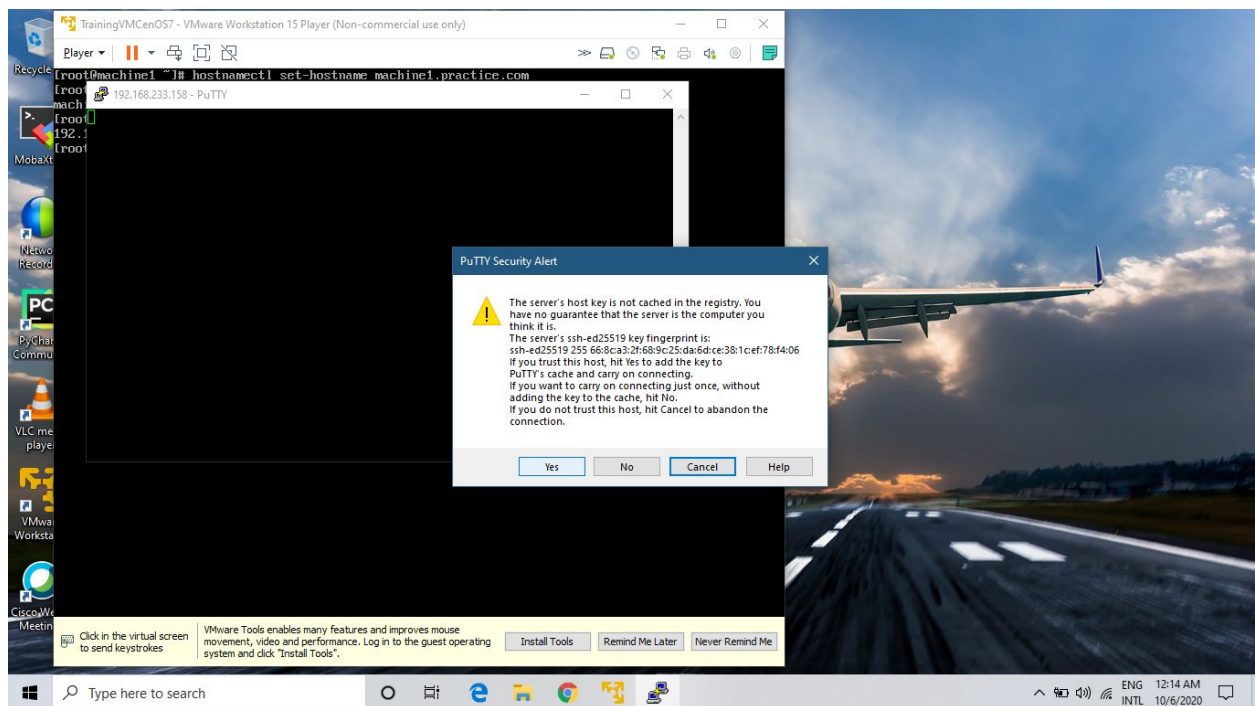


IMP:

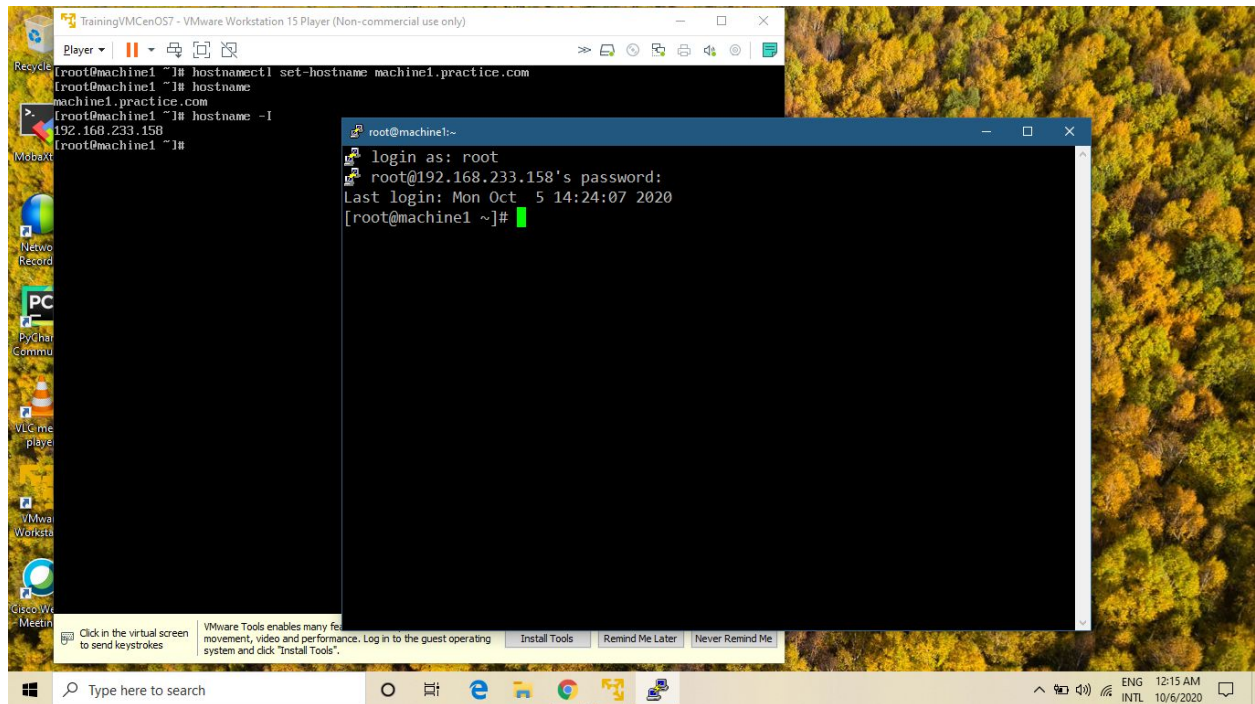
Install Putty on HOST windows machine and take ssh access on port 22 with user name root and password redhat with both machines. This is just to copy paste options for easy access. Else not needed. Remember user IP to take remote access.



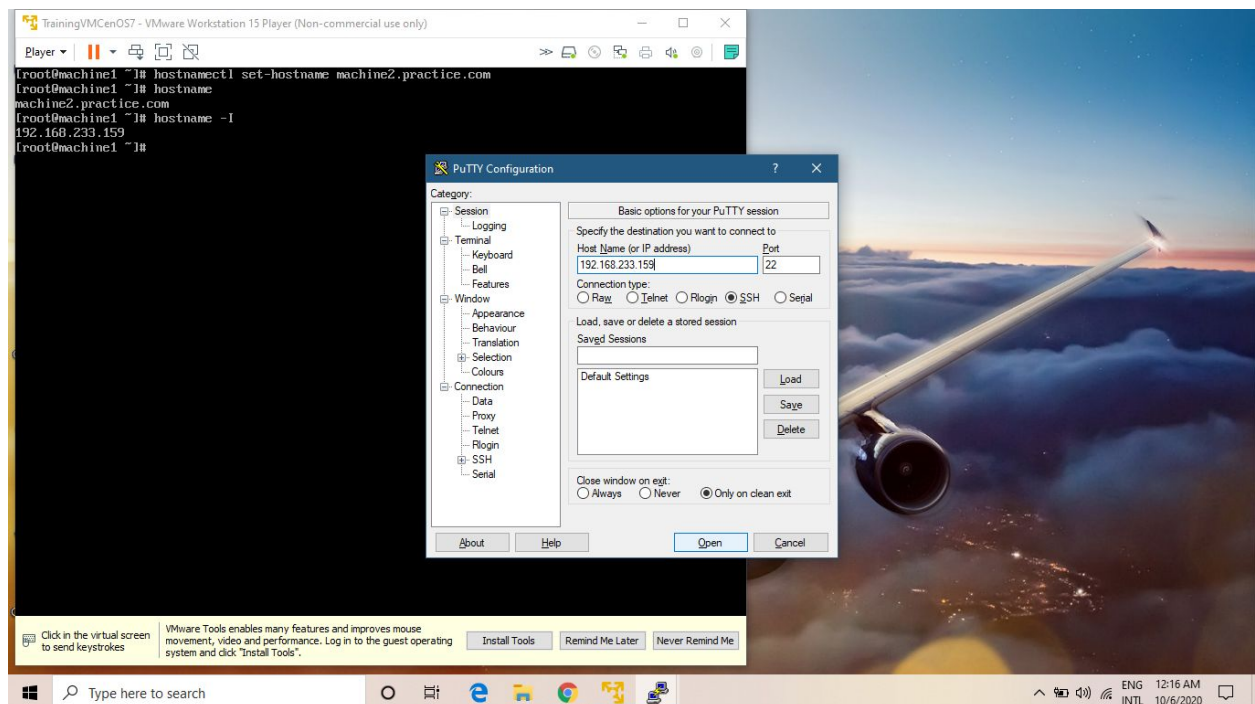
Select YES

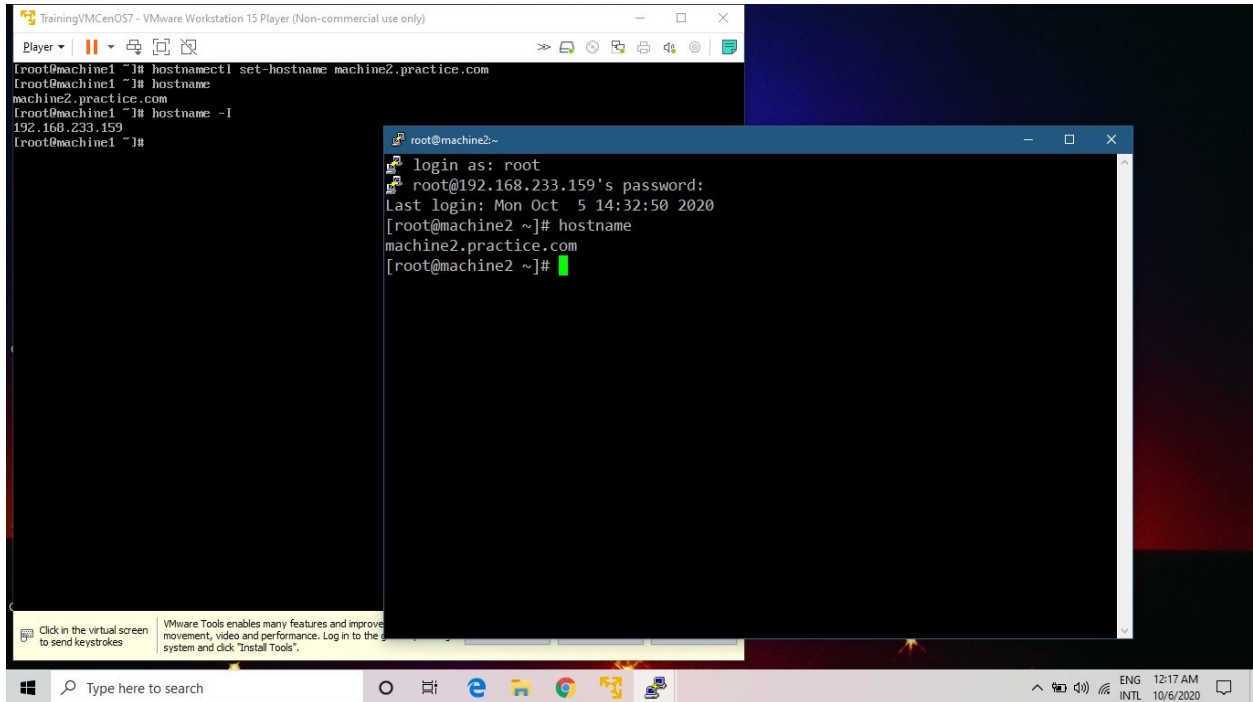


Login by user: root
Password: redhat



SAME for machine2:





=====

On machine 1 Install Jenkins now

Installation

Link:

<https://wiki.jenkins.io/display/JENKINS/Installing+Jenkins+on+Red+Hat+distributions>

Installation of an LTS version

There is also [a LTS YUM repository](#) for the [LTS Release Line](#)

yum install wget

- `wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo`
- `rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key`
- `yum upgrade`
- `yum install jenkins java-1.8.0-openjdk-devel`
- `systemctl daemon-reload`

- service jenkins start
- chkconfig jenkins on

Configure the firewall

```
#firewall-cmd --permanent --add-port=8080/tcp
```

```
#firewall-cmd --reload
```

```
#firewall-cmd --list-all
```

=====

Phase 2

Installation of git and Maven

```
# yum install git -y
```

```
# yum install maven -y
```

```
# git --version
```

```
# java -version
```

```
# mvn --version
```

=====

Add Git, Maven and Junit plugins to jenkins

Step1: Note down IP of machine1.practice.com

Open the browser of host machine and open

http://<IP_of_machine1>:8080

Ex.

```
#hostname -I
```


<http://192.168.233.158:8080>

Initial login password in file

```
cat /var/lib/jenkins/secrets/initialAdminPassword
```

Copy the password and login

Sign in [Jenkins]

Not secure | 192.168.233.158:8080/login?from=%2F

Incognito

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue

Setup Wizard [Jenkins]

Not secure | 192.168.233.158:8080

Incognito

Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

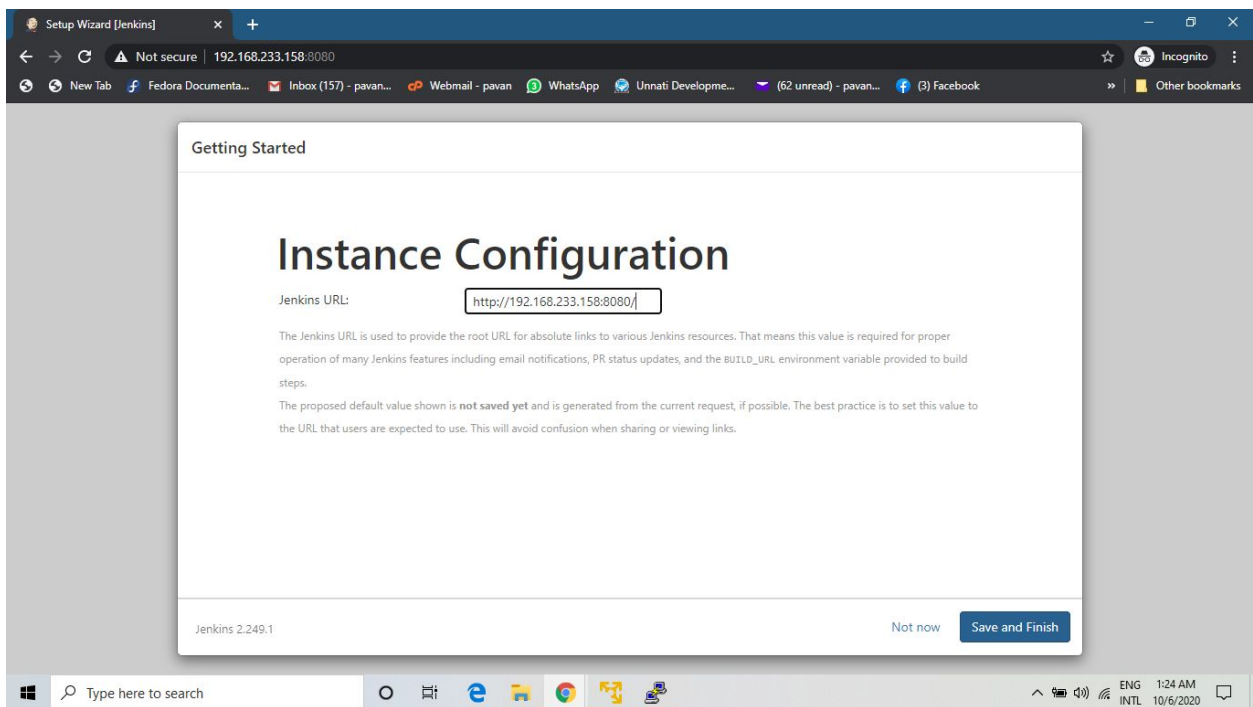
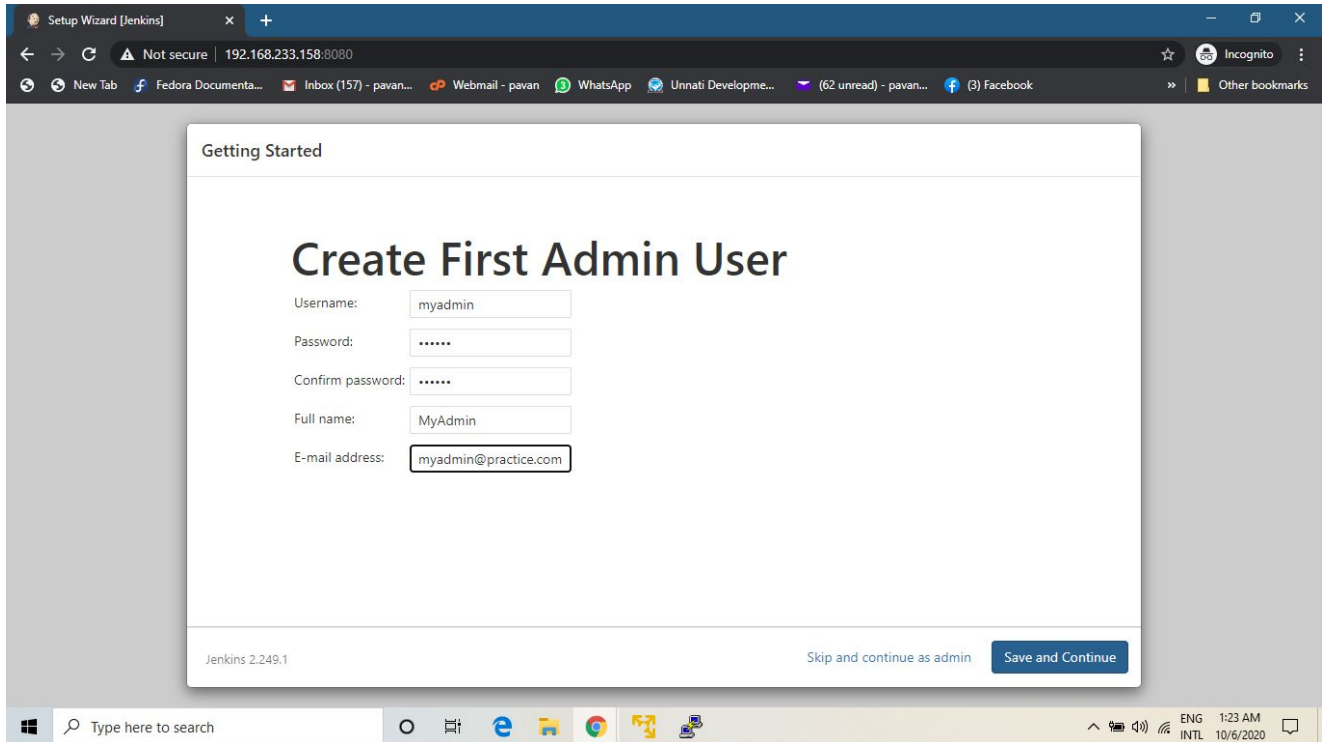
Install suggested plugins

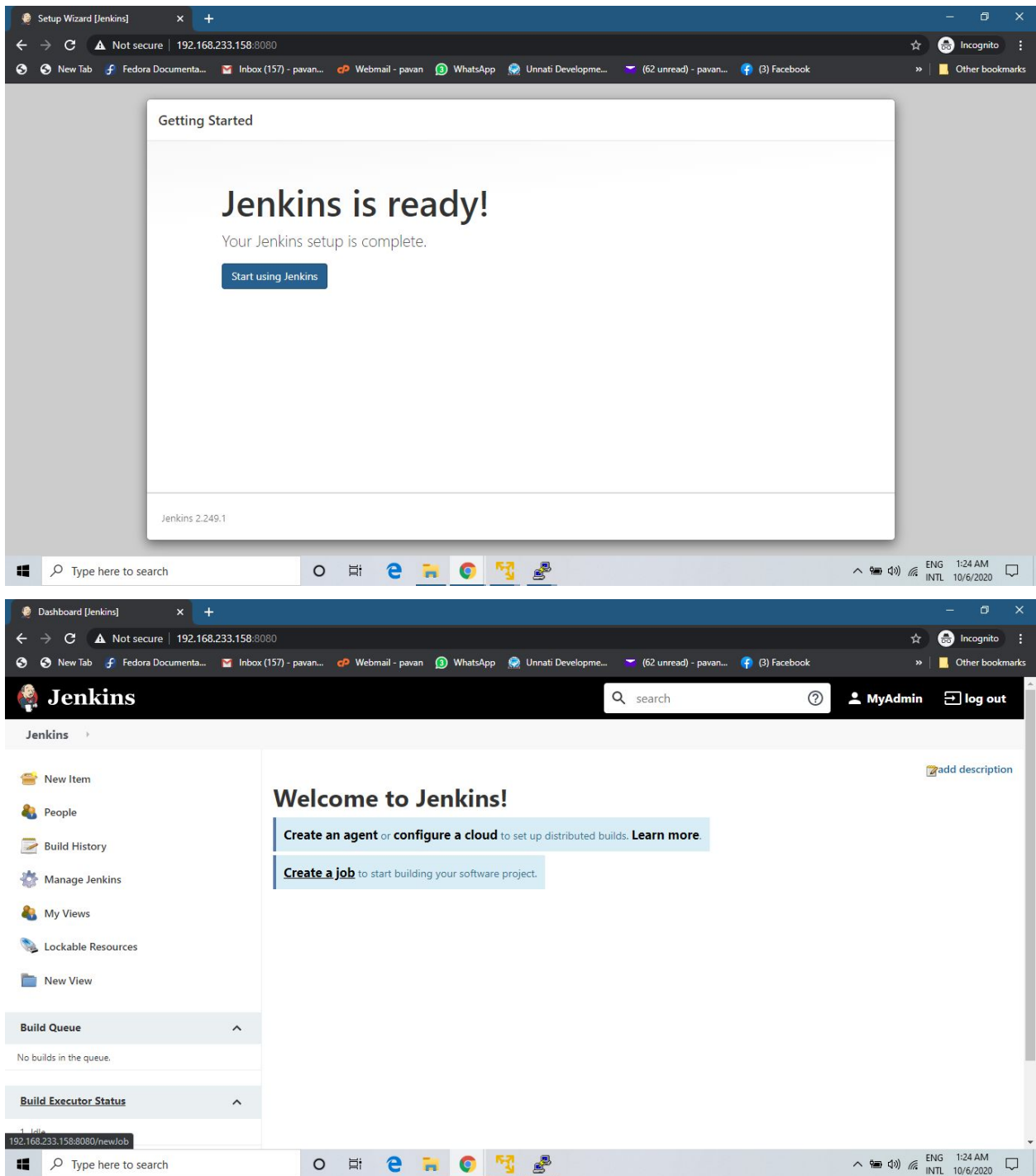
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.249.1





Select Install suggested plugins

Step: Add jenkins user to privilege users as sudo
visudo

```
jenkins    ALL=(ALL)    NOPASSWD: ALL  
(Add above line after 100th line)
```

=====

On Machine2.practice.com

Lets install Tomcat

Login as user root and password redhat

Step1: Install Tomcat

```
# yum update      (note:Optional step)
```

```
# yum install tomcat
```

```
# yum install tomcat-webapps tomcat-admin-webapps tomcat-docs-webapp  
tomcat-javadoc
```

Step2: Stop HTTPD service and start Tomcat service

```
# systemctl stop httpd
```

```
# systemctl start tomcat
```

```
# systemctl enable tomcat
```

```
# systemctl status tomcat
```

Step3: Create user “forjenkins” with manager access with password “forjenkins”

In order to use Tomcat's web management interface, you will need to create a user.
Open the tomcat-users.xml file with the command:

Scroll down to below the line which reads <tomcat-users> and add the information for your user account:

```
<user username="[username]" password="[password]"  
roles="manager-gui,admin-gui"/>
```

vi /usr/share/tomcat/conf/tomcat-users.xml

For example, to add the user “forjenkins” with password “forjenkins” this section will read:

```
<user username="forjenkins" password="forjenkins"  
roles="manager-gui,admin-gui,manager-script"/>
```

Save and exit the file. Restart the Tomcat service for the changes to take effect:

```
# systemctl restart tomcat
```

Step4:

Add port 8080 to firewall

```
# firewall-cmd --permanent --add-port=8080/tcp
```

```
# firewall-cmd --reload
```

```
# hostname -l
```

Step5:

In a browser, visit the URL http://<IP_of_machine2>:8080 to see the Tomcat welcome page.

Tomcat server is ready

=====

Lets Add machine 2 as jenkins node to jenkins master on machine1 using SSH

=====

Step1: From Master machine : machine1

Adding Node to Jenkins as SSH known host way(Another way is to have through JNLP client)

```
#usermod -s /bin/bash jenkins
```

```
# passwd jenkins
```

```
# visudo
```

```
jenkins    ALL=(ALL)    NOPASSWD: ALL
```

(Add above line after 100th line)

```
# su - jenkins
```


Step2: From Node machine : machine2

Client Machine:

Create user jenkins, set some password

Create a dir ex. Name : /work which should

=====

```
# useradd jenkins
# passwd jenkins
# mkdir /work
# chown jenkins:jenkins /work
# visudo
jenkins    ALL=(ALL)    NOPASSWD: ALL
```

(Add above line after 100th line)

```
21 yum install java -y
```

Step3:

FROM MASTER Machine Jenkins user

```
# su - jenkins
```

```
# ssh-keygen
```

```
# ssh-copy-id jenkins@192.168.233.159
```

This IP is of machine2

Step 4:

Open jenkins dashboard and join machine2 as Node2

Dashboard [Jenkins] x Apache Tomcat/7.0.76 x +

Not secure | 192.168.233.158:8080

Jenkins

search

pavan log out

Jenkins

- New Item
- People
- Build History
- Manage Jenkins
 - Manage Nodes and Clouds
- My Views
- Lockable Resources
- New View
- Lockable Resources
- New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

System Configuration

- Configure System
- Global Tool Configuration
- Manage Plugins

Security

- Configure Global Security
- Manage Credentials
- Configure Credential Providers
- Manage Users

Status Information

- System Information
- System Log
- Load Statistics
- About Jenkins

Troubleshooting

- Manage Old Data

Tools and Actions

- Reload Configuration from Disk
- Jenkins CLI
- Script Console
- Prepare for Shutdown

192.168.233.158:8080/computer

REST API Jenkins 2.249.1

Nodes [Jenkins] x Apache Tomcat/7.0.76 x +

Not secure | 192.168.233.158:8080/computer/

Jenkins

search

pavan log out

Jenkins

Nodes

- Back to Dashboard
- Manage Jenkins
- New Node
- Configure Clouds
- Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

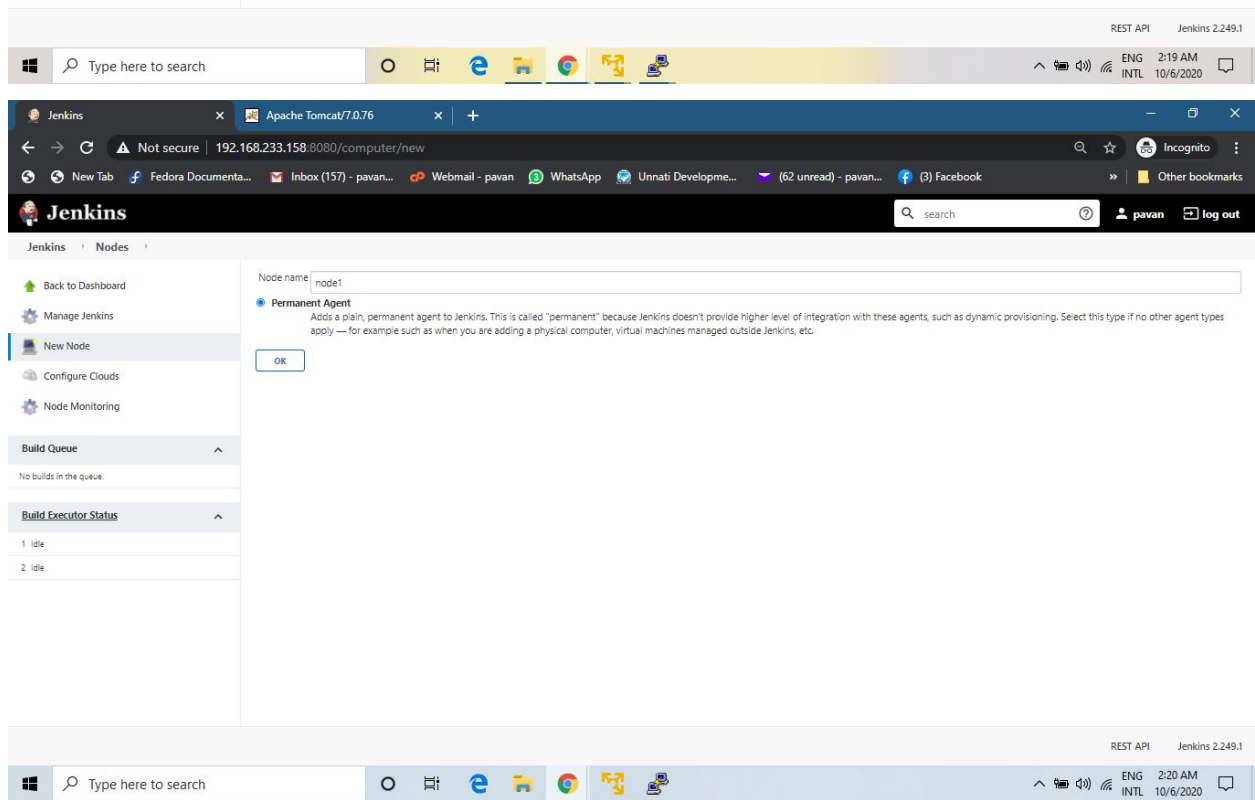
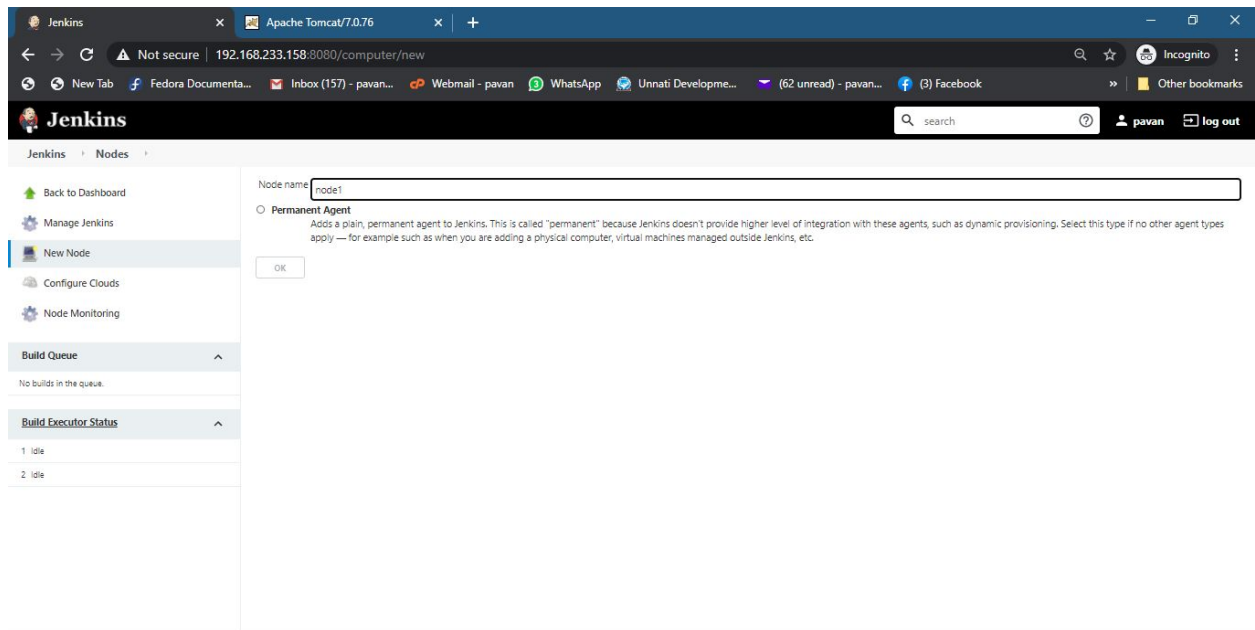
2 Idle

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	4.07 GB	816.45 MB	4.07 GB	0ms
	Data obtained	3 min 5 sec	3 min 5 sec	3 min 5 sec	3 min 5 sec	3 min 5 sec	3 min 5 sec

Refresh status

192.168.233.158:8080/computer/new

REST API Jenkins 2.249.1



Jenkins

Apache Tomcat/7.0.76

Not secure | 192.168.233.158:8080/computer/createItem

search

pavan log out

Jenkins Nodes

Back to Dashboard
Manage Jenkins
New Node
Configure Clouds
Node Monitoring

Build Queue
No builds in the queue.

Build Executor Status
1 idle
2 idle

Name: node1
Description: Tomcat server
of executors: 2
Remote root directory: /work
Labels: node1
Usage: Only build jobs with label expressions matching this node
Launch method: Launch agents via SSH
Host: 192.168.233.159
Credentials: - none - Add
Host Key Verification Strategy: Known hosts file Ver Jenkins Credentials Provider
Availability: Keep this agent online as much as possible
Advanced...

Node Properties
☐ Disable deferred wipeout on this node
☐ Environment variables
☐ Tool Locations
Save

Jenkins

Apache Tomcat/7.0.76

Not secure | 192.168.233.158:8080/computer/createItem

search

pavan log out

Jenkins Nodes

Back to Dashboard
Manage Jenkins
New Node
Configure Clouds
Node Monitoring

Build Queue
No builds in the queue.

Build Executor Status
1 idle
2 idle

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)
Kind: Username with password
Scope: Global (Jenkins, nodes, items, all child items, etc)
Username: jenkins
Password:
ID:
Description:
Add Cancel
Save

The screenshot displays the Jenkins 'New Node' configuration interface. The browser's address bar indicates the URL 'https://192.168.233.158:8080/computer/creatItem'. The Jenkins sidebar on the left provides navigation options such as 'Back to Dashboard', 'Manage Jenkins', 'New Node', 'Configure Clouds', and 'Node Monitoring'. The main configuration area includes the following fields:

- Name:** node1
- Description:** Tomcat server
- # of executors:** 2
- Remote root directory:** /work
- Labels:** node1
- Usage:** Only build jobs with label expressions matching this node
- Launch method:** Launch agents via SSH
- Host:** 192.168.233.159
- Credentials:** jenkins/*****
- Host Key Verification Strategy:** Known hosts file Verification Strategy
- Availability:** Keep this agent online as much as possible
- Node Properties:**
 - ☐ Disable deferred wipeout on this node
 - ☐ Environment variables
 - ☐ Tool Locations

A 'Save' button is located at the bottom left of the configuration area.

node1 [Jenkins] x Apache Tomcat/7.0.76 x +

Not secure | 192.168.233.158:8080/computer/node1/

Jenkins

node1

Back to List

Status

Delete Agent

Configure

Build History

Load Statistics

Log

Build Executor Status

Agent node1 (Tomcat server)

This node is being launched. [See log for more details](#)

Relaunch agent

Projects tied to node1

None

Mark this node temporarily offline

Nodes [Jenkins] x Apache Tomcat/7.0.76 x +

Not secure | 192.168.233.158:8080/computer/

Jenkins

Nodes

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

master

1 idle

2 idle

node1

1 idle

2 idle

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	master	Linux (amd64)	In sync	4.07 GB	815.60 MB	4.07 GB	0ms
	node1	Linux (amd64)	In sync	4.97 GB	819.99 MB	4.97 GB	302ms
Data obtained		3 sec	2.7 sec	2.8 sec	2.9 sec	2.8 sec	2.7 sec

Refresh status

=====

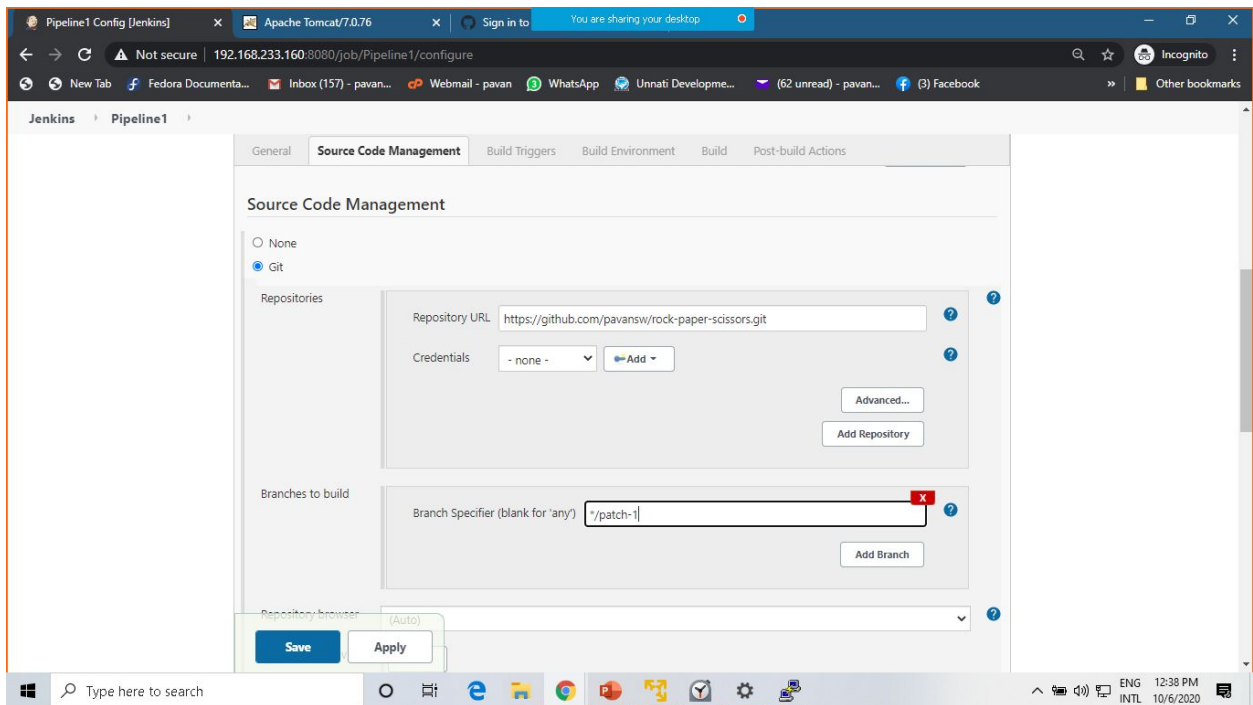
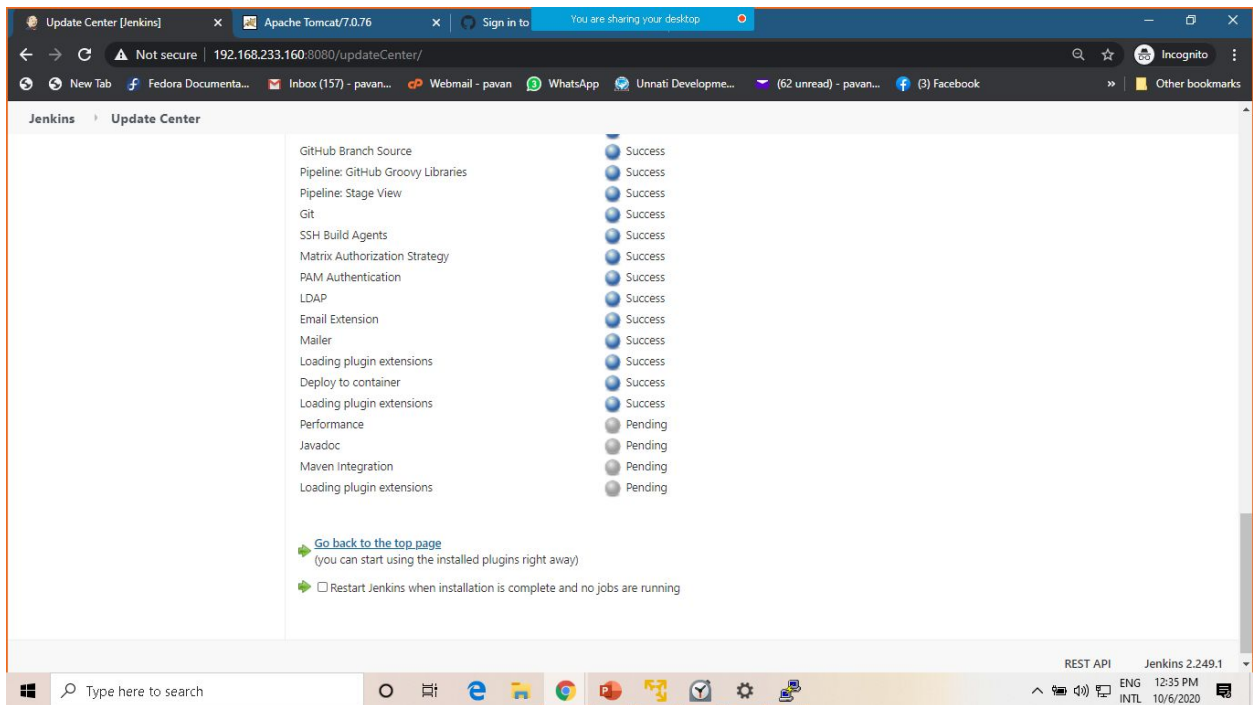
TOMCAT deployment

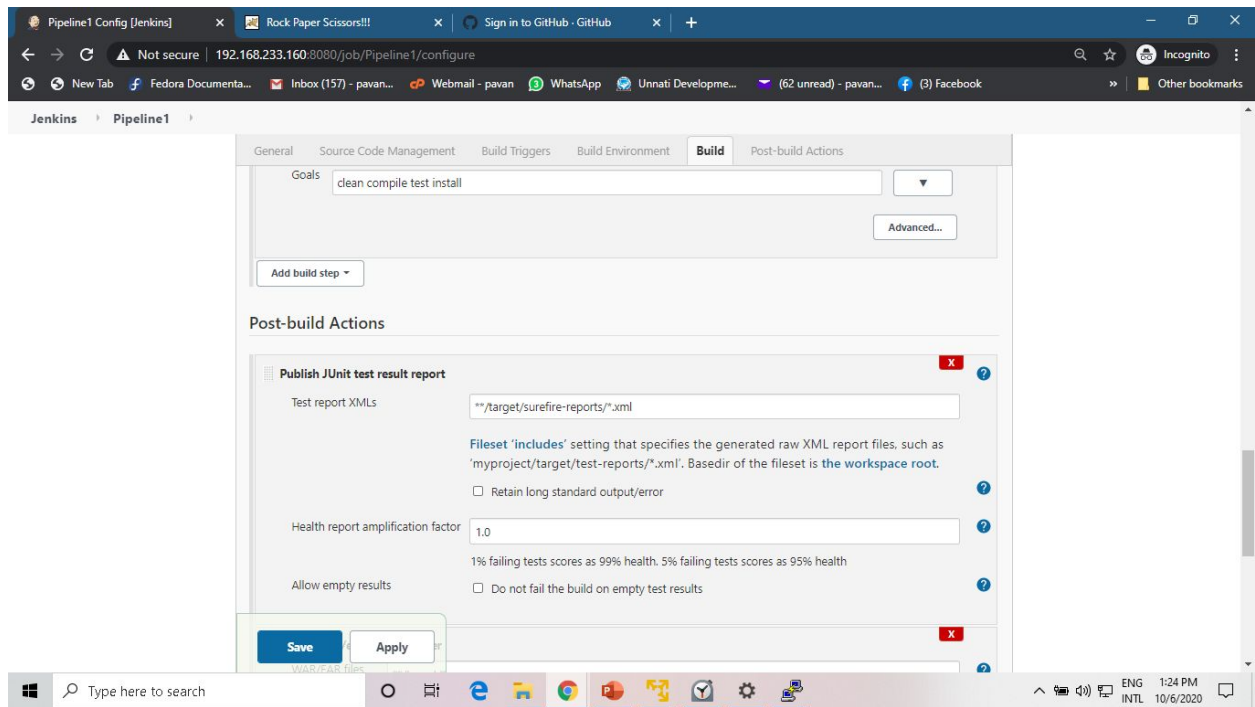
=====

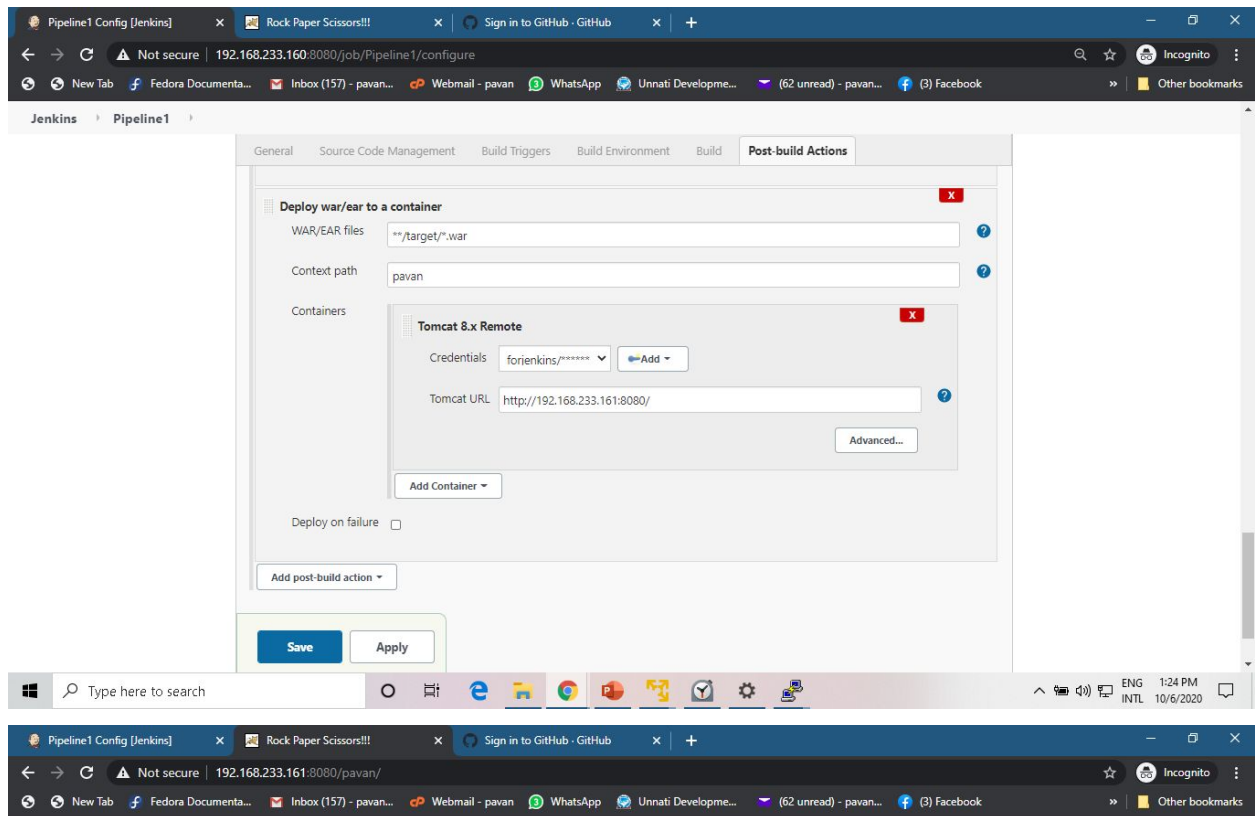
The screenshot shows the Jenkins Plugin Manager interface. The top navigation bar includes links for Back to Dashboard, Manage Jenkins, and Update Center. The main content area is titled 'Jenkins Plugin Manager' and features a search bar with the text 'container'. Below the search bar, there are tabs for 'Updates', 'Available', 'Installed', and 'Advanced'. The 'Available' tab is selected, displaying a table of plugins. The table has columns for 'Install', 'Name', 'Version', and 'Released'. The plugins listed are:

- Docker Pipeline**: Build and use Docker containers from pipelines. Version 1.24, released 1 mo 18 days ago. It includes tags for Deployment, DevOps, docker, and pipeline.
- Deploy to container**: This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment. Version 1.15, released 1 yr 2 mo ago. It includes a tag for Artifact Uploaders.
- Docker Slaves**: Uses Docker containers to run Jenkins build agents. Version 1.0.7, released 3 yr 2 mo ago. It includes a tag for docker.
- Amazon Elastic Container Service (ECS) / Fargate**: Use Amazon EC2 Container Service to provide elastic agents. Version 1.37, released 12 days ago. It includes tags for aws, Cluster Management and Distributed Build, and Agent Launchers and Controllers.

At the bottom of the table, there are buttons for 'Install without restart', 'Download now and install after restart', and 'Check now'. The status bar at the bottom indicates 'Update information obtained: 53 min ago'.







Which one will it be for you today?
[rock paper scissors](#)



EXAMPLE Pipeline code

```
pipeline {
node any

stages {

        stage('code') {

            steps {

                git branch: 'patch-1', url:
                'https://github.com/pavansw/rock-paper-scissors.git'

            }

        }

        stage('compile') {

            steps {

                sh label: "", script: 'mvn clean compile'

            }

        }

        stage('testing') {

            steps {

                sh label: "", script: 'mvn test'

            }

        }

        stage('report') {

            steps {

                junit '**/*.xml'

            }

        }

    }
}
```

```
stage('package') {  
    steps {  
        sh label: "", script: 'mvn install'  
    }  
}  
  
stage('Deploy Tomcat on node1') {  
    steps {  
        deploy adapters: [tomcat8(credentialsId:  
            'c9177718-53c2-497d-b98a-535c3a755758', path: "", url:  
            'http://192.168.233.161:8080/)], contextPath: 'application1', war:  
            '**/target/*.war'  
        }  
    }  
}  
}
```

=====