

CI/CD Deployment Using Ansible CM Tool

Project 1

Post Graduate Program in DevOps

PG DO - Configuration Management with Chef, Puppet and
Ansible

Vignesh Dharmaraj
vikidvg@gmail.com

Table of Contents

1.	<i>Introduction to the Project</i>	2
2.	<i>Installation of pre-requisites/tools</i>	2
2.1	Installing Git	2
2.2	Creating GitHub Account	3
2.3	Setting up Jenkins	3
2.4	Install Ansible on Ubuntu	8
2.5	Jenkins config for Ansible	12
2.6	SSH Connectivity setup to Managed Nodes	13
2.7	Test SSH Connectivity to Managed Nodes	15
3.	<i>Execution of the project</i>	15
4.	<i>Testing of Project</i>	21
5.	<i>Project Results</i>	22
6.	<i>Conclusion</i>	27

1. Introduction to the Project

Objective: CI/CD Deployment Using Ansible CM Tool

Solution build should demonstrate below capabilities:

1. Configure Jenkins server as Ansible provisioning machine
2. Install Ansible plugins in Jenkins CI server
3. Prepare Ansible playbook to run Maven build on Jenkins CI server
4. Prepare Ansible playbook to execute deployment steps on the remote web container with restart of the web container post deployment

Project goal is to Automate Ansible integration with Jenkins CI server so that we can run and execute playbooks to deploy custom WAR files to a web container and then perform restart for the web container.

Tools required: Ansible, GitHub, Git, Linux (Ubuntu), Jenkins

2. Installation of pre-requisites/tools

In this section we can see how the required tools are installed to execute the project

Note: This project is implemented by the “PG DO - Certified Kubernetes Administrator (CKA) Training” Lab (Kubernetes New VM) Lab provided by SimpliLearn, so most of the tools may be already installed including the Linux (Ubuntu). Lab has three nodes with 4 GB RAM, 20 GB Hard Drive any Linux OS (Ubuntu).

2.1 Installing Git

Step 1: Verifying the Git installation

- Use the following command to check the version of Git:

git --version

```
vikidvggma@ip-172-31-29-62:~$ git --version
git version 2.7.4
```

Note: Execute **Step 2** in case you don't get any results for **git --version** command.

Step 2: Installing the latest version of Git

- Execute the following commands on the terminal to install Git:

```
sudo apt-get update
```

```
sudo apt-get install git
```

```
Get:22 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [130 kB]
Get:23 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:25 http://ppa.launchpad.net/ansible/ansible/ubuntu xenial/main amd64 Packages [696 B]
Fetched 1,206 kB in 1s (954 kB/s)
Reading package lists... Done
W: http://repo.zabbix.com/zabbix/3.0/ubuntu/dists/trusty/InRelease: Signature by key FBABD5FB20255ECAB22EE194D13D58E479EA5ED4 uses weak digest algorithm (SHA1)
vikidvggmail@ip-172-31-29-62:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.7.4-0ubuntu1.10).
0 upgraded, 0 newly installed, 0 to remove and 85 not upgraded.
```

2.2 Creating GitHub Account

Make sure you have a Github Account available. If not, please create one using the given link.

https://github.com/join?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

2.3 Setting up Jenkins

Step 1: Downloading the Java Runtime Environment

1.1 Open the terminal.

1.2 Run **sudo apt-get update** to update the package lists.

1.3 Run **sudo apt-get install openjdk-8-jdk** to install the Java Runtime Environment.

1.4 Run **java -version** to verify the installation. It will print the JDK version as shown below:

```
vikidvgmail@ip-172-31-29-62:~$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~16.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
```

Step 2: Downloading and installing the Jenkins app

1.1 Open the terminal.

1.2 Run **wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -** to install Jenkins.

```
susmitaadhyapak@susmitaadhyapak:~$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

2.3 Run **sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'** command.

```
susmitaadhyapak@susmitaadhyapak:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

2.4 Run **sudo apt-get update**

2.5 Run **sudo apt-get install jenkins** to install Jenkins.

```
susmitaadhyapak@susmitaadhyapak:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  fonts-lato javascript-common jruby libbytelist-java libhawtjni-runtime-java
  libheadius-options-java libinvokebinder-java libjansi-java
  libjansi-native-java libjcodings-java libjffi-java libjffi-jni
  libjnr-constants-java libjnr-enxio-java libjnr-ffi-java libjnr-netdb-java
  libjnr-posix-java libjnr-unixsocket-java libjnr-x86asm-java
  libjoda-time-java libjruby-joni-java libjs-jquery libjzlib-java liblua5.2-0
  libreadline7 libssl1.0.2 libtcl8.6 libunsafe-mock-java libyaml-snake-java
  libyecht-java nailgun rake vim-gui-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  daemon
The following NEW packages will be installed:
  daemon jenkins
0 upgraded, 2 newly installed, 0 to remove and 306 not upgraded.
Need to get 70.6 MB of archives.
After this operation, 71.2 MB of additional disk space will be used.
```

2.6 Run **sudo service jenkins status** to check the status of the installation. Once you verify the status as active, you can press **Ctrl+z** to exit from the process.

```
vikitv@gmail@ip-172-31-29-62:~$ sudo service jenkins status
● jenkins.service - LSB: Start Jenkins at boot time
  Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)
  Active: active (exited) since Thu 2021-07-22 09:59:50 UTC; 30min ago
    Docs: man:systemd-sysv-generator(8)
 Process: 1563 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCE
   Tasks: 0
  Memory: 0B
     CPU: 0

Jul 22 09:59:48 ip-172-31-29-62 systemd[1]: Starting LSB: Start Jenkins at boot
Jul 22 09:59:49 ip-172-31-29-62 jenkins[1563]: Correct java version found
Jul 22 09:59:49 ip-172-31-29-62 jenkins[1563]: * Starting Jenkins Automation Se
Jul 22 09:59:49 ip-172-31-29-62 su[1745]: Successful su for jenkins by root
Jul 22 09:59:49 ip-172-31-29-62 su[1745]: + ??? root:jenkins
Jul 22 09:59:49 ip-172-31-29-62 su[1745]: pam_unix(su:session): session opened f
Jul 22 09:59:50 ip-172-31-29-62 jenkins[1563]:      ...done.
Jul 22 09:59:50 ip-172-31-29-62 systemd[1]: Started LSB: Start Jenkins at boot t
lines 1-17/17 (END)
```

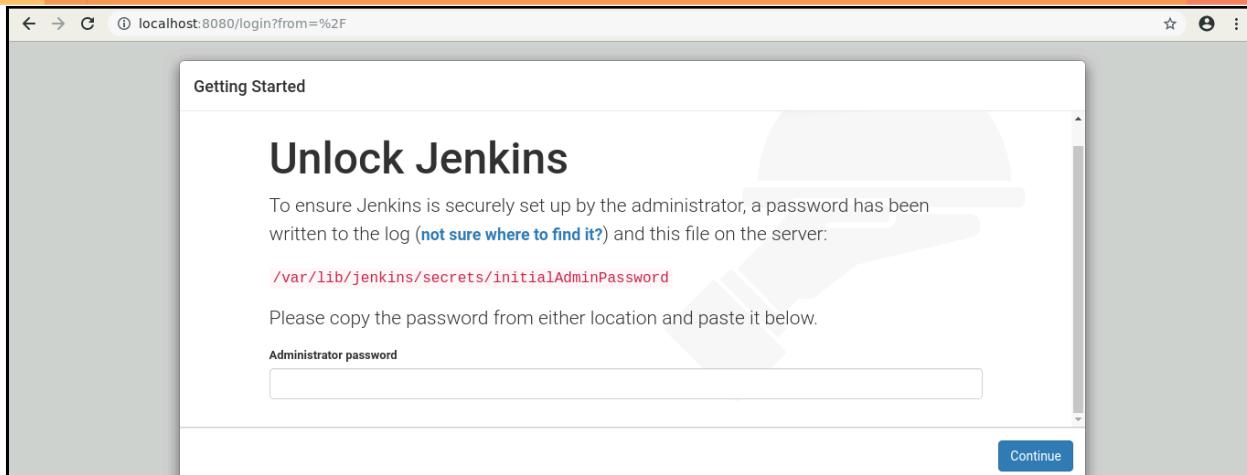
2.7 Run the following commands to start Jenkins.

```
sudo systemctl start jenkins
sudo systemctl status jenkins
```

```
susmitaadhya@susmitaadhya:~$ sudo systemctl start jenkins
susmitaadhya@susmitaadhya:~$ sudo systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
  Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)
  Active: active (exited) since Tue 2021-03-23 08:02:03 UTC; 4min 2s ago
    Docs: man:systemd-sysv-generator(8)

Mar 23 08:02:01 susmitaadhya systemd[1]: Starting LSB: Start Jenkins at boot
Mar 23 08:02:01 susmitaadhya jenkins[5672]: Correct java version found
Mar 23 08:02:01 susmitaadhya jenkins[5672]: * Starting Jenkins Automation Se
Mar 23 08:02:01 susmitaadhya su[5737]: Successful su for jenkins by root
Mar 23 08:02:01 susmitaadhya su[5737]: + ??? root:jenkins
Mar 23 08:02:01 susmitaadhya su[5737]: pam_unix(su:session): session opened f
Mar 23 08:02:03 susmitaadhya jenkins[5672]:      ...done.
Mar 23 08:02:03 susmitaadhya systemd[1]: Started LSB: Start Jenkins at boot t
Mar 23 08:05:53 susmitaadhya systemd[1]: Started LSB: Start Jenkins at boot t
lines 1-14/14 (END)
```

2.8 Open **localhost:8080** in the browser, and you will need to enter the initial password.

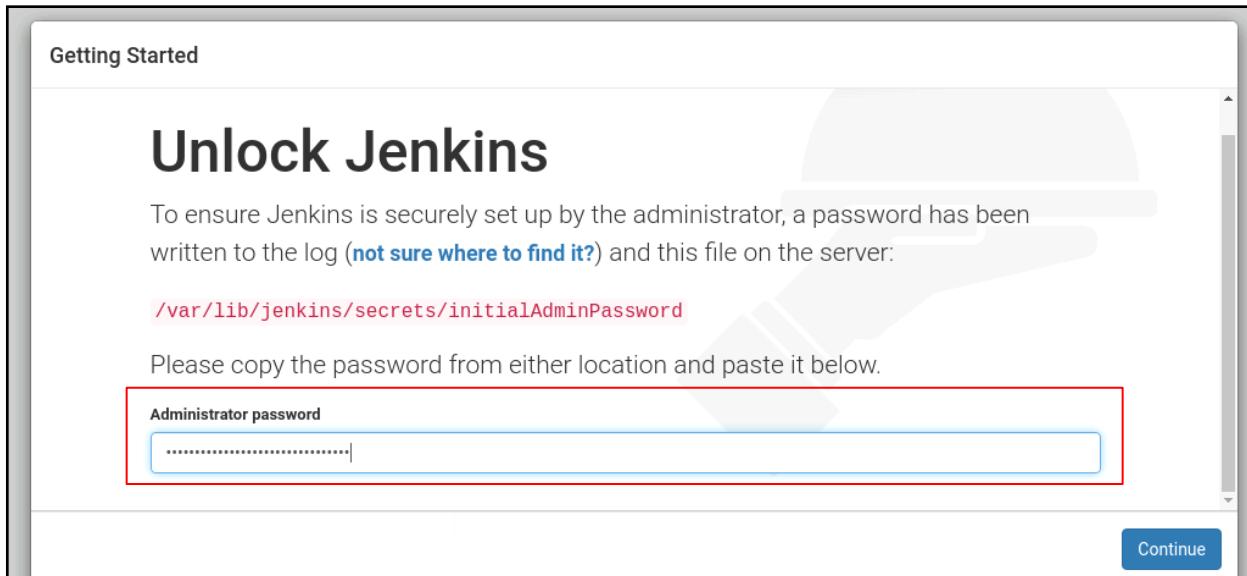


2.9 In your terminal run the following command:

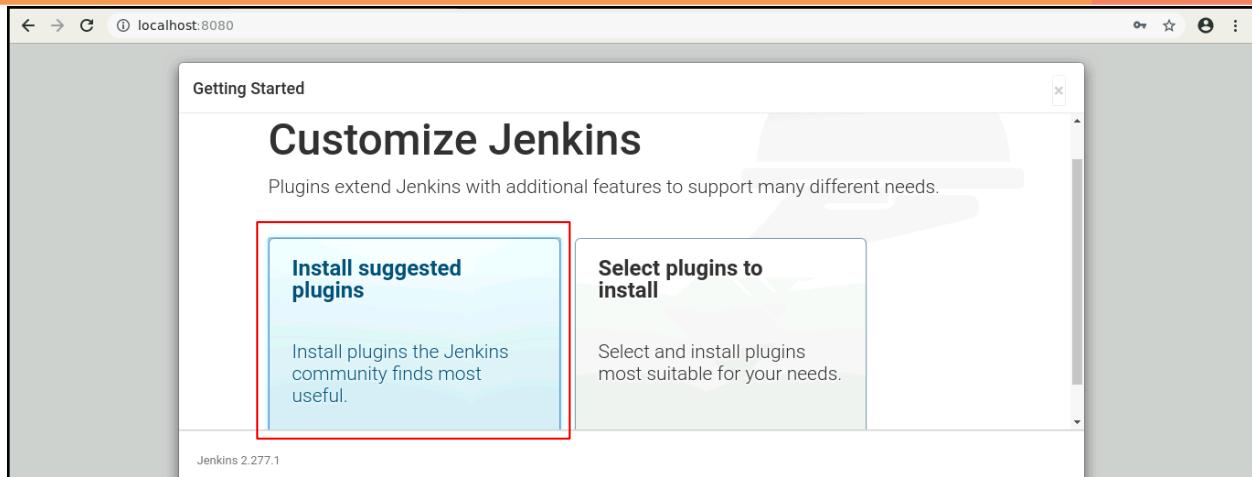
```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

```
susmitaadhyapak@susmitaadhyapak:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
876821d4689a453c87c48116a59a001b
susmitaadhyapak@susmitaadhyapak:~$ █
```

2.10 Copy this password and paste it on your Jenkins page in the browser.



2.11 Now, click on **Install the suggested plugins**.



2.12 You can either create an admin user or skip and continue as admin. Select **Skip and continue as admin**.

A screenshot of the 'Create First Admin User' form. The title is 'Create First Admin User'. It has five input fields: 'Username', 'Password', 'Confirm password', 'Full name', and 'E-mail address'. At the bottom right, there are two buttons: 'Skip and continue as admin' (highlighted with a red box) and 'Save and Continue'.

2.13 In the Instance configuration page, click on the **Start using Jenkins** button.

Getting Started

Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.277.1

2.14 Now, you can work with Jenkins as shown in the screenshot below.

The screenshot shows the Jenkins dashboard at `localhost:8080`. The left sidebar includes links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Lockable Resources', and 'New View'. The main area displays a table for the 'ansible_proj_vignesh_dharmaraj' job, which has a green checkmark icon, a blue cloud icon, and the name 'ansible_proj_vignesh_dharmaraj'. It shows the last success was 3 hr 2 min ago (#19), the last failure was 3 hr 8 min ago (#18), and the last duration was 21 sec. There are also links for 'add description', 'Icon: S M L', 'Legend', and 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

2.4 Install Ansible on Ubuntu

- Use the below commands on Ubuntu system to install ansible software:

`sudo apt-get install -f`

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

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

```
sudo apt-get update
```

```
sudo apt-get install ansible
```

```
anjanasinghsimp@ip-172-31-67-15:~$ sudo apt-get install -f
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-aws-headers-4.4.0-1118 linux-headers-4.4.0-1118-aws linux-image-4.4.0-1118-aws linux-modules-4.4.0-1118-aws
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.
anjanasinghsimp@ip-172-31-67-15:~$ anjanasinghsimp@ip-172-31-67-15:~$
```

```
anjanasinghsimp@ip-172-31-67-15:~$ sudo apt-add-repository ppa:ansible/ansible
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Press [ENTER] to continue or ctrl-c to cancel adding it

gpg: keyring `/tmp/tmpg3m47cka/secring.gpg' created
gpg: keyring `/tmp/tmpg3m47cka/pubring.gpg' created
gpg: requesting key 7BB9C367 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpg3m47cka/trustdb.gpg: trustdb created
gpg: key 7BB9C367: public key "Launchpad PPA for Ansible, Inc." imported
gpg: Total number processed: 1
gpg:           imported: 1  (RSA: 1)
OK
anjanasinghsimp@ip-172-31-67-15:~$
```

```
anjanasinghsimp@ip-172-31-67-15:~$ sudo apt-get update
gpg:           imported: 1  (RSA: 1)
OK
anjanasinghsimp@ip-172-31-67-15:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Hit:4 https://deb.nodesource.com/node_14.x xenial InRelease
Get:5 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:6 http://repo.zabbix.com/zabbix/3.0/ubuntu trusty InRelease
Hit:7 http://ppa.launchpad.net/ansible/ubuntu xenial InRelease
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [326 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 DEP-11 Metadata [281 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 DEP-11 Metadata [5,964 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main amd64 DEP-11 Metadata [3,328 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 DEP-11 Metadata [6,616 B]
Hit:13 http://ppa.launchpad.net/remmina-ppa-team/remmina-next-daily/ubuntu xenial InRelease
Get:14 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [93.7 kB]
Get:15 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [130 kB]
Get:16 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Fetched 1,175 kB in 0s (1,369 kB/s)
Reading package lists... Done
W: http://repo.zabbix.com/zabbix/3.0/ubuntu/dists/trusty/InRelease: Signature by key FBABD5FB20255ECAB22EE194D13D58E479EA5ED4 uses weak digest algorithm (SHA1)
anjanasinghsimp@ip-172-31-67-15:~$ anjanasinghsimp@ip-172-31-67-15:~$
```

```

File Edit View Terminal Tabs Help
Preparing to unpack .../python-ecdsa_0.13-2ubuntu0.16.04.1_all.deb ...
Unpacking python-ecdsa (0.13-2ubuntu0.16.04.1) ...
Selecting previously unselected package python-paramiko.
Preparing to unpack .../python-paramiko_1.16.0-1ubuntu0.2_all.deb ...
Unpacking python-paramiko (1.16.0-1ubuntu0.2) ...
Selecting previously unselected package python-httplib2.
Preparing to unpack .../python-httplib2_0.9.1+dfsg-1_all.deb ...
Unpacking python-httplib2 (0.9.1+dfsg-1) ...
Selecting previously unselected package sshpass.
Preparing to unpack .../sshpass_1.05-1_amd64.deb ...
Unpacking sshpass (1.05-1) ...
Selecting previously unselected package ansible.
Preparing to unpack .../ansible_2.9.19-1ppa-xenial_all.deb ...
Unpacking ansible (2.9.19-1ppa-xenial) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up python-markupsafe (0.23-2build2) ...
Setting up python-jinja2 (2.8-1ubuntu0.1) ...
Setting up python-yaml (3.11-3build1) ...
Setting up python-crypto (2.6.1-6ubuntu0.16.04.3) ...
Setting up python-ecdsa (0.13-2ubuntu0.16.04.1) ...
Setting up python-paramiko (1.16.0-1ubuntu0.2) ...
Setting up python-httplib2 (0.9.1+dfsg-1) ...
Setting up sshpass (1.05-1) ...
Setting up ansible (2.9.19-1ppa-xenial) ...
root@ip-172-31-67-15:/home/anjanasinghsimp#

```

- Establish SSH key pair in linux system to have SSH connectivity with localhost using the following commands:

ssh-keygen -t rsa

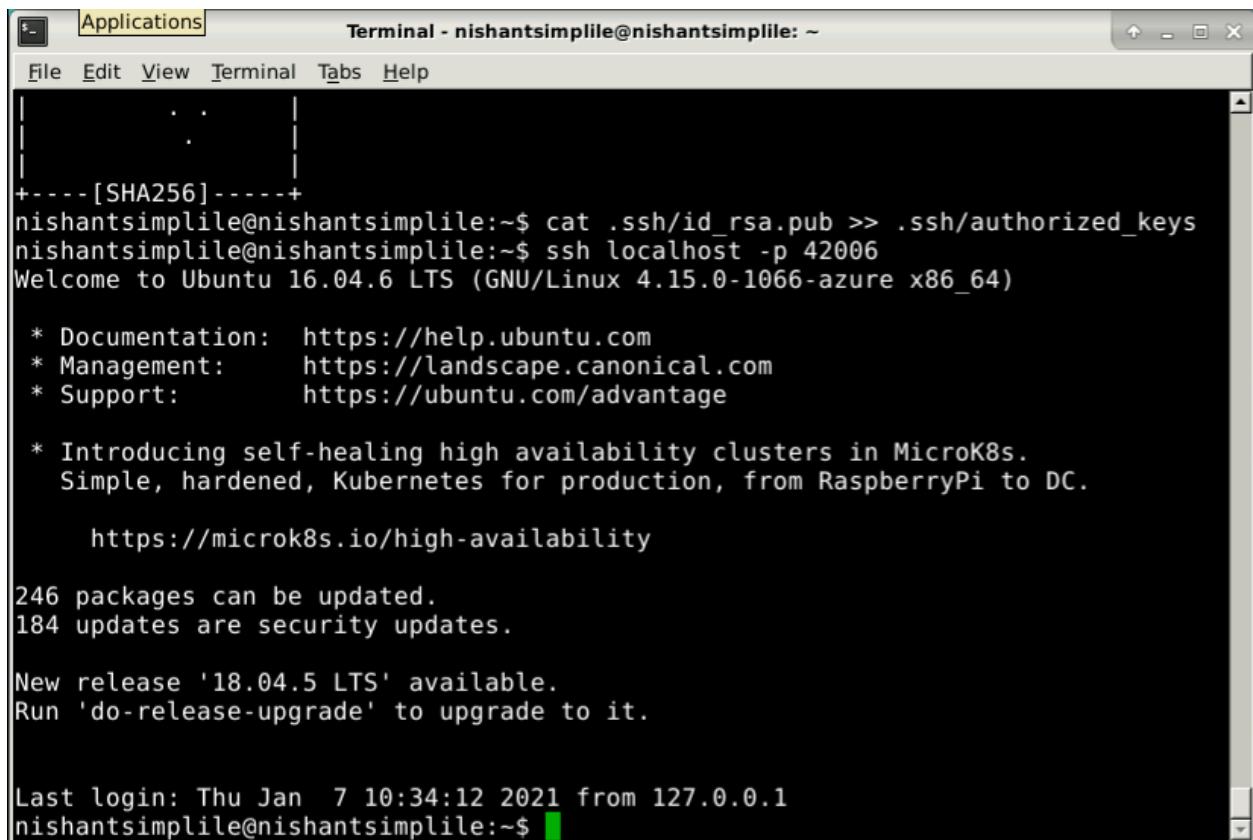
cat .ssh/id_rsa.pub >> .ssh/authorized_keys

ssh localhost -p 42006

```

File Edit View Terminal Tabs Help
nishantsimplile@nishantsimplile:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nishantsimplile/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nishantsimplile/.ssh/id_rsa.
Your public key has been saved in /home/nishantsimplile/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:BPvVlgzeS8ugw3bCFJb0K8k9Vhe4+aDlsF2LgN2tHvA nishantsimplile@nishantsimplile
The key's randomart image is:
+---[RSA 2048]----+
|   0=..o +.
|   .0= .B .
|   .+=+++
|   =.=***+o
|   @ S@+* .
|   . *o.E o
|       .
|       .
+---[SHA256]----+
nishantsimplile@nishantsimplile:~$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
nishantsimplile@nishantsimplile:~$ ssh localhost -p 42006
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1066-azure x86_64)

```



The screenshot shows a terminal window titled "Terminal - nishantsimplile@nishantsimplile: ~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content is as follows:

```
| . . |
+----[SHA256]----+
nishantsimplile@nishantsimplile:~$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
nishantsimplile@nishantsimplile:~$ ssh localhost -p 42006
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1066-azure x86_64)

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

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

   https://microk8s.io/high-availability

246 packages can be updated.
184 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Jan  7 10:34:12 2021 from 127.0.0.1
nishantsimplile@nishantsimplile:~$
```

- Now, add the host localhost in the ansible host file /etc/ansible/hosts.

sudo vi /etc/ansible/hosts

- When the file opens, add the below two lines of the code at the end of the file:

[webservers]

localhost:42006

Step 2: Establish connectivity between Ansible controller and node machine

- Execute the below command to validate host inventory file:

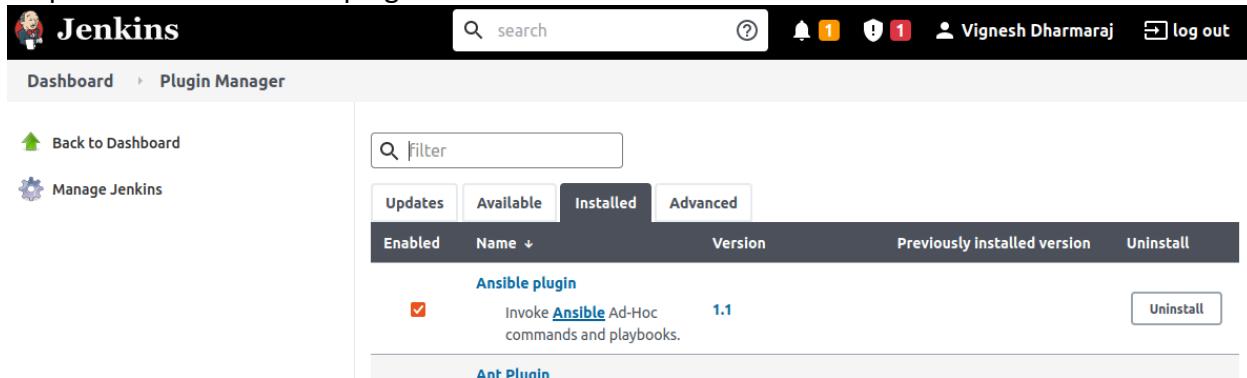
ansible -m ping webservers

```
root@docker:~# ansible -m ping webservers
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

2.5 Jenkins config for Ansible

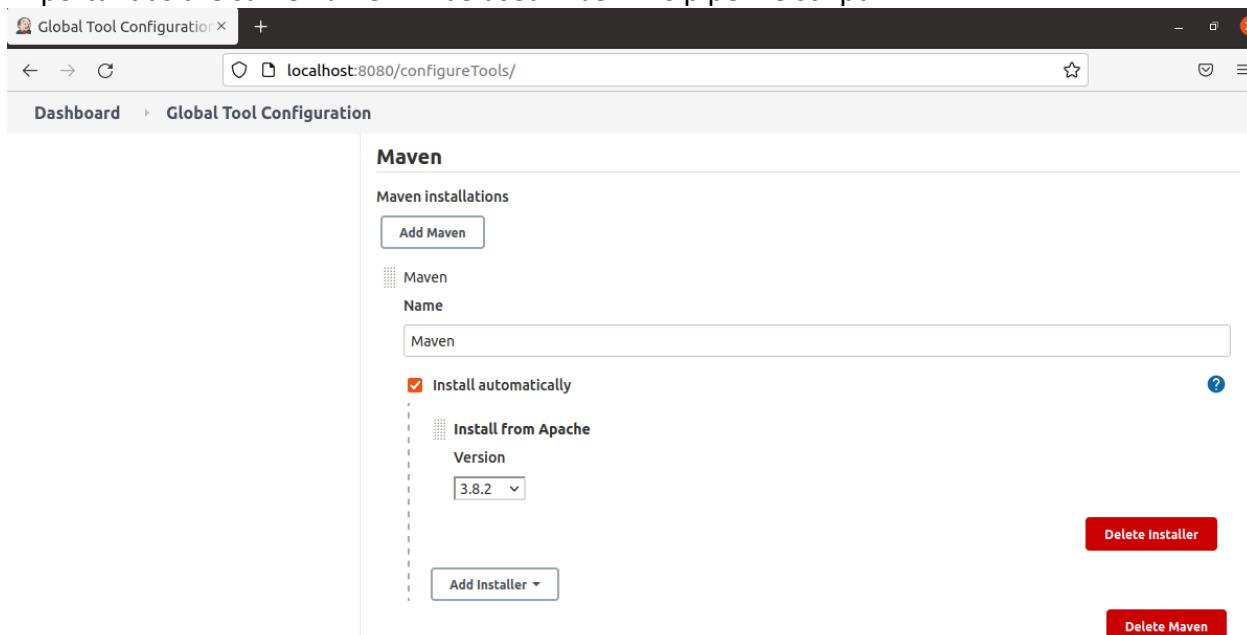
Jenkins and Ansible is already installed in the Ansible controller (master node), further Jenkins configuration has to be done as below:

Step 1: Install the Ansible plugin in Jenkins



The screenshot shows the Jenkins Plugin Manager interface. The top navigation bar includes a search bar, user information (Vignesh Dharmaraj), and a log out button. Below the header, the breadcrumb navigation shows 'Dashboard > Plugin Manager'. On the left sidebar, there are links for 'Back to Dashboard' and 'Manage Jenkins'. The main content area features a search bar labeled 'Filter' and tabs for 'Updates', 'Available', 'Installed' (which is selected), and 'Advanced'. A table lists the installed plugin, with columns for 'Enabled', 'Name', 'Version', 'Previously installed version', and 'Uninstall'. The 'Ansible plugin' row is highlighted, showing it is enabled, its name is 'Invoke Ansible Ad-Hoc commands and playbooks.', its version is '1.1', and there is a 'Uninstall' button.

Step 2: Global tools config to be done for Ansible and Maven as below, the 'Name' configured is important as the same name will be used in Jenkins pipeline script.



The screenshot shows the Jenkins Global Tool Configuration page for Maven. The top navigation bar includes a search bar, user information (Vignesh Dharmaraj), and a log out button. Below the header, the breadcrumb navigation shows 'Dashboard > Global Tool Configuration'. The main content area is titled 'Maven' and contains a section for 'Maven installations'. It features a 'Add Maven' button and a table for managing Maven installations. One entry is visible: 'Maven' with 'Name' set to 'Maven', 'Install automatically' checked, and 'Install from Apache' selected with 'Version' set to '3.8.2'. Buttons for 'Delete Installer' and 'Delete Maven' are located at the bottom right.

The screenshot shows a web-based tool configuration interface. At the top, there's a header bar with a logo and navigation links. Below it, the main content area has a title "Global Tool Configuration". Under this, there's a section for "Ansible" with a sub-section for "Ansible installations". It includes fields for "Name" (set to "Ansible") and "Path to ansible executables directory" (set to "/usr/bin"). There's also a checkbox for "Install automatically" and a "Delete Ansible" button. A "Add Ansible" button is located at the bottom left of this section.

2.6 SSH Connectivity setup to Managed Nodes

SSH connectivity to be setup between the Ansible provisioning node and the managed nodes. As the Ansible provisioning is done through Jenkins, a user called 'Jenkins' need to be created in all the managed nodes and SSH connectivity has to be done with that user.

Step 1: Create user 'Jenkins' in the master node (already created as part of Jenkins installation) & managed nodes and do the necessary config for the user to execute all the tasks with elevated user privilege

```
labsuser@ip-172-31-22-46:~$ id jenkins
id: 'jenkins': no such user
labsuser@ip-172-31-22-46:~$ sudo -i
root@ip-172-31-22-46:~# useradd -m -s /bin/bash jenkins
root@ip-172-31-22-46:~# passwd jenkins
New password:
Retype new password:
passwd: password updated successfully
root@ip-172-31-22-46:~# echo -e 'jenkins\tALL=(ALL)\tNOPASSWD:\tALL' > /etc/sudoers.d/jenkins
root@ip-172-31-22-46:~#
```

Step 2: Ensure SSHD config file (/etc/ssh/sshd_config) has below parameters enabled

PasswordAuthentication yes

PermitRootLogin yes

After modifying the SSSH config file restart SSSH service with below command:
 sudo systemctl restart sshd

note the ip of the managed node by using the below command:
 ip a

```
root@ip-172-31-22-46:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 02:b0:cc:75:c3:93 brd ff:ff:ff:ff:ff:ff
        inet 172.31.22.46/20 brd 172.31.31.255 scope global dynamic ens5
            valid_lft 2238sec preferred_lft 2238sec
        inet6 fe80::b0:ccff:fe75:c393/64 scope link
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:a4:c5:ac:1c brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
root@ip-172-31-22-46:~#
```

Ens5: inet has the ip address

Step 3: From the master node ssh-copy-id to the managed node ip to establish the ssh connectivity

Azure Resourcegroup Kubernetes New VM This Lab will get reset on 06th Oct 2021

master worker1 worker2

Used 38.8 of 50 hours in Sep, 2021 ▶ Start Lab ⏺

Save

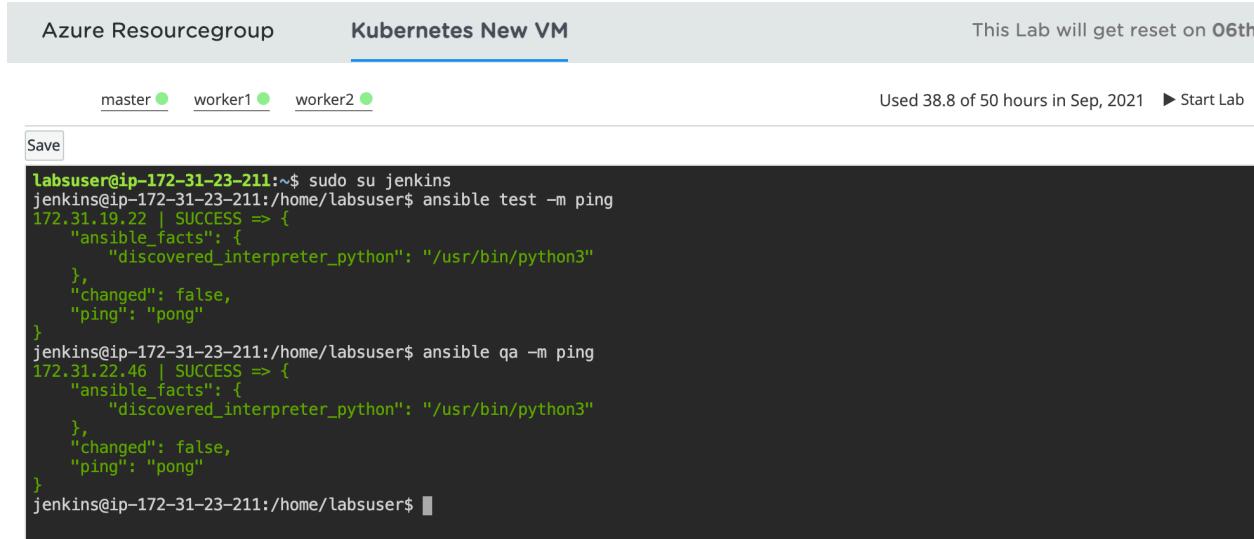
```
labsuser@ip-172-31-23-211:~$ ls
Desktop  Downloads  Pictures  Templates  ansible.cfg  loop1.yaml  myinventory  os_version.yaml  tera  tomcat.yaml
Documents  Music  Public  Videos  eignore.yaml  loop2.yaml  nohup.out  roles  test  tomcat_restart.yaml
labsuser@ip-172-31-23-211:~$ sudo su jenkins
jenkins@ip-172-31-23-211:/home/labsuser$ ssh-copy-id 172.31.22.46
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/jenkins/.ssh/id_rsa.pub"
The authenticity of host '172.31.22.46 (172.31.22.46)' can't be established.
ECDSA key fingerprint is SHA256:5rHB4C9XMSfZ25fzVtkASpFB0qmM+b8idi5p7SIGBE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
jenkins@172.31.22.46's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '172.31.22.46'"
and check to make sure that only the key(s) you wanted were added.
```

2.7 Test SSH Connectivity to Managed Nodes

After all the above tools installation and configuration steps are followed, Now the SSH connectivity is established between the ansible controller machine and the managed nodes with Jenkins user, tested as shown below:



The screenshot shows a cloud lab interface with tabs for 'Azure Resourcegroup' and 'Kubernetes New VM'. The 'Kubernetes New VM' tab is selected. A status bar at the top right indicates 'This Lab will get reset on 06th'. Below the tabs, there are three nodes listed: 'master' (green), 'worker1' (green), and 'worker2' (green). A message 'Used 38.8 of 50 hours in Sep, 2021' and a 'Start Lab' button are also visible. A 'Save' button is located in the top-left corner of the main content area. The main content area displays a terminal session output:

```
Save
labsuser@ip-172-31-23-211:~$ sudo su jenkins
jenkins@ip-172-31-23-211:/home/labsuser$ ansible test -m ping
172.31.19.22 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
jenkins@ip-172-31-23-211:/home/labsuser$ ansible qa -m ping
172.31.22.46 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
jenkins@ip-172-31-23-211:/home/labsuser$
```

Connectivity has been setup with 2 managed nodes and the ansible provisioning required by the project will be deployed to both the nodes below:

Ansible managed nodes / machines

172.31.19.22

172.31.22.46

3. Execution of the project

All the necessary code is created in the below GitHub repository, this repo can be cloned to execute the Jenkins pipeline

https://github.com/vdharraja/PGDO_CM_Project_1

Below steps can be followed by anyone to execute the project:

Step 1:

with \$git clone https://github.com/vdharraja/PGDO_CM_Project_1.git command you can clone the project into your machine and the changes to be made according to your setup is explained in further steps. You can also fork/clone the project from github, as it is a public repository.

Step 2:

In the inventory file of the playbook change your managed host according to your need, after testing the connectivity to your host



```
main ▾ PGDO_CM_Project_1 / inventories / dev / hosts

vdharmaraj added a new host ...
1 contributor

3 lines (3 sloc) | 32 Bytes

1 [dev]
2 172.31.19.22
3 172.31.22.46
```

Step 4:

In the Jenkinsfile, checkout stage related git clone url and branch has to be changed according to your github links.

This stage in Jenkinsfile is just used for displaying the Git repository fetch.

main ➔ PGDO_CM_Project_1 / Jenkinsfile

vdharmaraj jenkins file initial commit

1 contributor

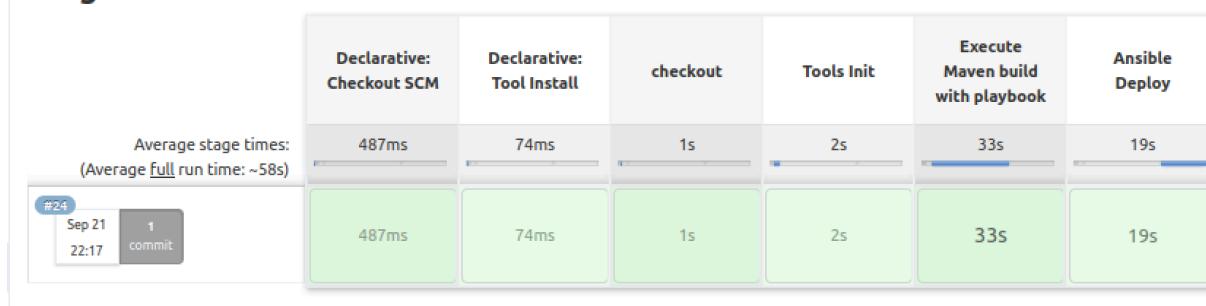
56 lines (35 sloc) | 1.08 KB

```
1 pipeline {  
2     agent any  
3  
4     tools  
5     {  
6         maven "Maven"  
7     }  
8  
9     stages {  
10        stage('checkout') {  
11            steps {  
12                git branch: 'main', url: 'https://github.com/vdharmaraj/PGDO_CM_Project_1.git'  
13            }  
14        }  
15    }  
16}
```

Step 5:

Create new pipeline job in Jenkins to poll the SCM (Git) and execute the Jenkins pipeline file with below stages:

Stage View



Jenkins pipeline job is created with below configuration:

The screenshot shows the Jenkins interface for creating a new item. The title bar says 'localhost:8080/view/all/newJob'. The main area has a heading 'Enter an item name' with a text input field containing 'new pipeline job'. Below it, a note says '» Required field'. There are three project types listed: 'Freestyle project' (selected), 'Pipeline', and 'Multi-configuration project'.

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

The screenshot shows the Jenkins Pipeline configuration page for the job 'ansible_proj_vignesh_dharmaraj'. The top navigation bar includes 'Dashboard', 'ansible_proj_vignesh_dharmaraj', 'General', 'Build Triggers', 'Advanced Project Options', and 'Pipeline'. The 'General' tab is selected.

Description
new pipeline job to execute ansible for doing build (create war file) and deploy the changes in tomcat server created in managed hosts

General Configuration Options

- Discard old builds
- Do not allow concurrent builds
- Do not allow the pipeline to resume if the controller restarts
- GitHub project
- Pipeline speed/durability override
- Preserve stashes from completed builds
- This project is parameterized

To poll the SCM for new commits every minute

The screenshot shows the Jenkins interface for configuring build triggers. The 'Build Triggers' tab is selected. Under the 'Schedule' section, the cron expression '*' is displayed, indicating a poll every minute. A warning message at the bottom states: '⚠ Do you really mean "every minute" when you say "* * * *"? Perhaps you meant "H * * * *" to poll once per hour'. Below the schedule, there are checkboxes for 'Ignore post-commit hooks' and 'Disable this project', both of which are unchecked.

Repository URL is configured in the below step which will be used to poll the SCM for any commits in the code repository.

The screenshot shows the Jenkins Pipeline configuration page. The 'Pipeline' tab is selected. In the 'Definition' section, the 'Pipeline script from SCM' dropdown is set to 'SCM'. Under 'SCM', 'Git' is selected, and the 'Repository URL' field contains the URL 'https://github.com/vdharma...git'. The 'Credentials' section shows a dropdown menu set to '- none -' and an 'Add' button. There is also an 'Advanced...' button at the bottom right.

Jenkinsfile reference from the repository is given in the 'Script Path' as below:

The screenshot shows the Jenkins Pipeline configuration page for a job named 'ansible_proj_vignesh_dharmaraj'. The 'Pipeline' tab is selected. The configuration includes:

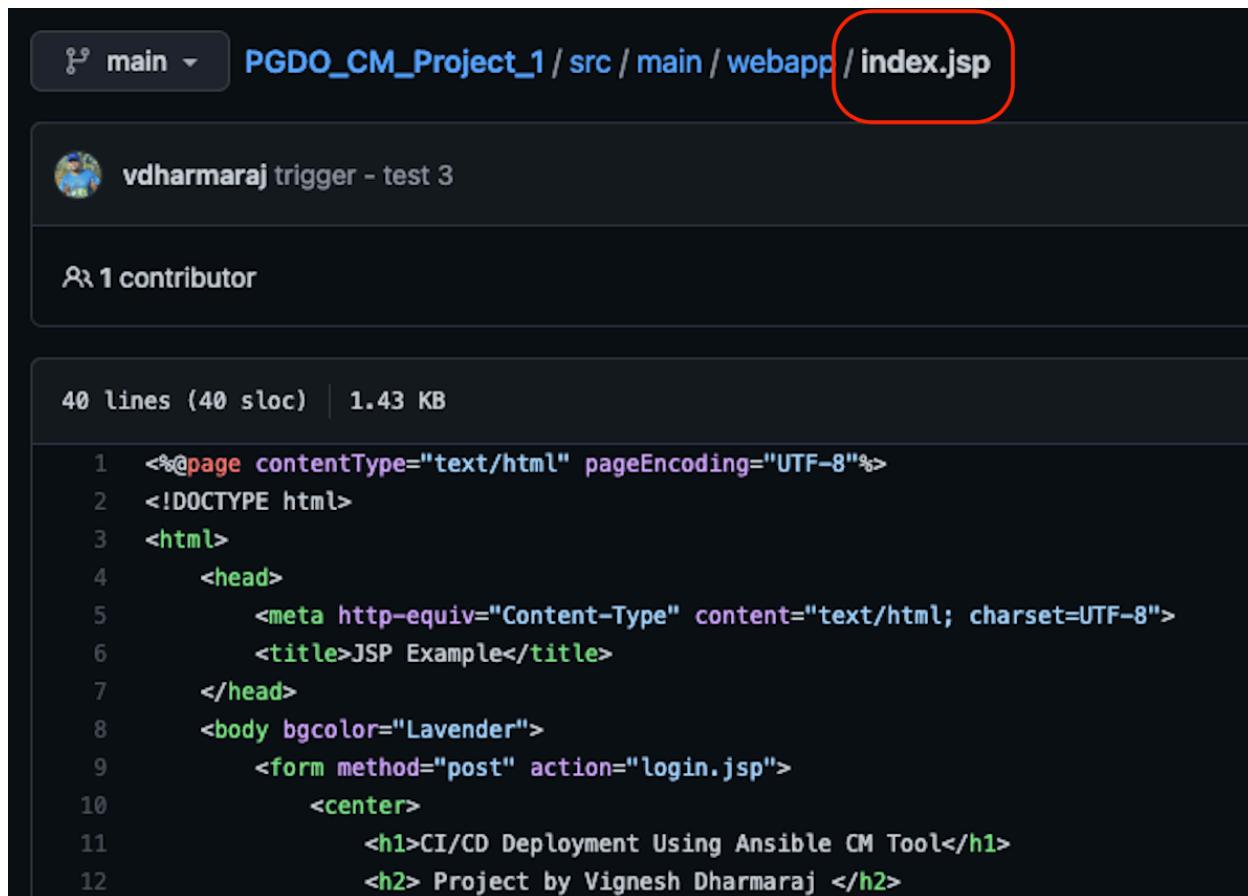
- Branches to build:** Branch Specifier (blank for 'any') set to */main, with an 'Add Branch' button.
- Repository browser:** Set to (Auto).
- Additional Behaviours:** An 'Add' button.
- Script Path:** Jenkinsfile, with a checked checkbox for Lightweight checkout.
- Pipeline Syntax:** Pipeline Syntax link.

Save and apply will save the Jenkins pipeline job and will be scheduled to poll the SCM (git) every minute for changes committed to the project repository.

This step completes all the project configurations required.

4. Testing of Project

In our Repository the index.jsp file, which is part of java web App source code, this servlet file can be modified and the changes can be committed to the repository which will trigger the Jenkins Poll SCM job and the CI/CD process will be triggered to deploy the new application changes in the managed hosts with the help of ansible



The screenshot shows a GitHub repository interface. At the top, there is a dropdown menu labeled "main" and a path "PGDO_CM_Project_1 / src / main / webapp / index.jsp". The "index.jsp" part of the path is circled in red. Below the path, there is a profile picture of a person and the name "vdharmaraj trigger - test 3". Underneath that, it says "1 contributor". At the bottom, there is a code editor displaying the content of the index.jsp file. The code is as follows:

```
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4      <head>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <title>JSP Example</title>
7      </head>
8      <body bgcolor="Lavender">
9          <form method="post" action="login.jsp">
10             <center>
11                 <h1>CI/CD Deployment Using Ansible CM Tool</h1>
12                 <h2> Project by Vignesh Dharmaraj </h2>
```

5. Project Results

Result 1:

Jenkins job will be triggered automatically corresponding to the repository commit version, this can be seen in the Job build history

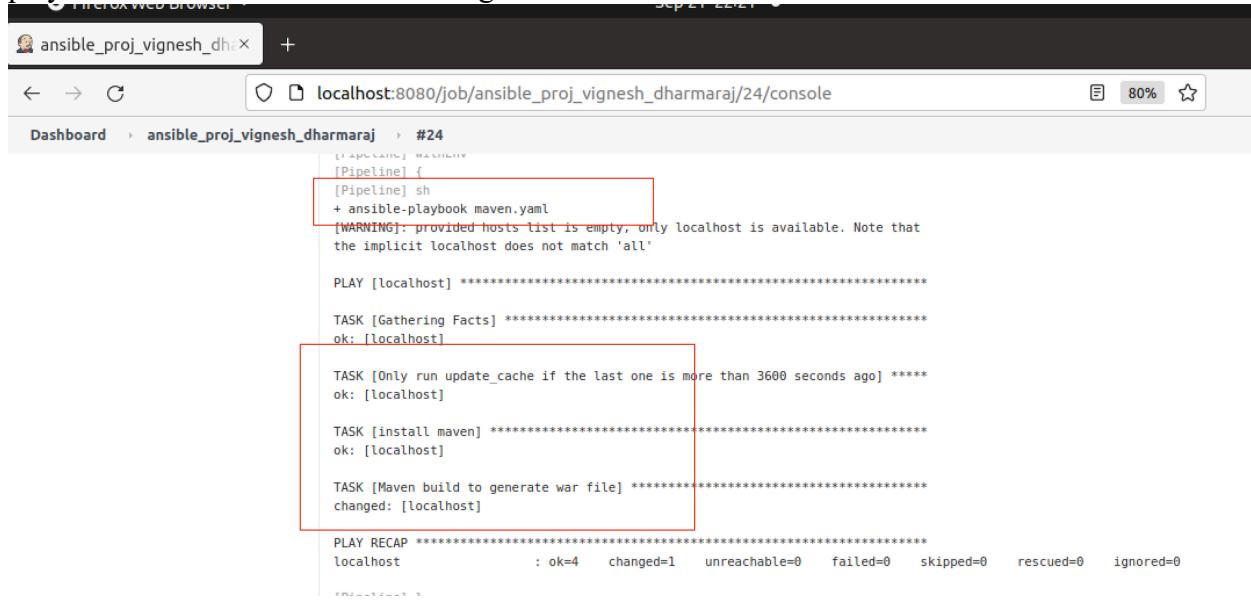
The screenshot shows the Jenkins Stage View interface. On the left, there's a sidebar with various options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Rename, Pipeline Syntax, and Git Polling Log. Below that is a Build History section with two entries: #21 (Sep 21, 2021 8:03 PM) and #20 (Sep 21, 2021 7:08 PM). The main area is titled "Stage View" and displays a table of build stages. The columns are: Declarative: Checkout SCM, Declarative: Tool Install, checkout, Tools Init, Execute Maven, and Ansible Deploy. The table includes average stage times and a full run time of ~33s. A tooltip over a commit in row #21 indicates it was triggered by "trigger - test 3".

	Declarative: Checkout SCM	Declarative: Tool Install	checkout	Tools Init	Execute Maven	Ansible Deploy
Average stage times: (Average full run time: ~33s)	696ms	141ms	789ms	1s	7s	15s
#21 Sep 21 20:03	1 commit	1 commit c650d9 trigger - test 3	118ms	977ms	2s	9s
#20 Sep 21 19:08	1 commit		318ms	1s	1s	9s
#19 Sep 21	1	425ms	61ms	378ms	741ms	4s

The screenshot shows the GitHub repository page for "vdharmaraj / PGDO_CM_Project_1". The main navigation bar includes Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. The repository is public. The commit history for "main" is shown, with a red box highlighting a specific commit from Sep 22, 2021. This commit is labeled "trigger - test 3" and was made by "vdharmaraj" 14 minutes ago. Below this commit, another message says "added a new host ..." and was also made by "vdharmaraj" 1 hour ago. Other commits listed include "Commits on Sep 21, 2021".

Result 2:

Jenkins job console output is verified to see if the mvn packaging/build is created by a ansible playbook in the Jenkins CI server to generate new war file



The screenshot shows the Jenkins job console output for a job named 'ansible_proj_vignesh_dharmaraj' run #24. The output is displayed in a terminal window with several red boxes highlighting specific parts of the log.

```
[Pipeline] sh
+ ansible-playbook maven.yaml
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

PLAY [localhost] ****
TASK [Gathering Facts] ****
ok: [localhost]

TASK [Only run update_cache if the last one is more than 3600 seconds ago] ****
ok: [localhost]

TASK [install maven] ****
ok: [localhost]

TASK [Maven build to generate war file] ****
changed: [localhost]

PLAY RECAP ****
localhost : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Result 3:

Jenkins job console output is verified to check if new war file is kept in tomcat server folder and then the tomcat is started and restarted in both the managed nodes

← → ⌂

localhost:8080/job/ansible_proj_vignesh_dharmaraj/

Jenkins

Dashboard > ansible_proj_vignesh_dharmaraj >

Back to Dashboard

Pipeline ansible_proj_vignesh

Stage Logs (Ansible Deploy)

```
① Use a tool from a predefined Tool Installation -- Maven (self time 24ms)
② Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 0ms)
③ Shell Script – ansible-playbook main.yml -i inventories/dev/hosts (self time 16s)
ok: [172.31.19.22]

TASK [tomcat : Create sample directory] *****
ok: [172.31.19.22]
ok: [172.31.22.46]
④ TASK [tomcat : copy war file] *****
⑤ changed: [172.31.19.22]
⑥ changed: [172.31.22.46]
```

```

23
24   - name: Symlink install directory
25     file: src=/opt/apache-tomcat-7.0.61 path=/usr/share/tomcat state=link
26
27   - name: Change ownership of Tomcat installation
28     file: path=/usr/share/tomcat/ owner=tomcat group=tomcat state=directory recurse=yes
29
30   - name: Configure Tomcat server
31     template: src=server.xml dest=/usr/share/tomcat/conf/
32
33
34   - name: Create sample directory
35     file:
36       path: "/opt/apache-tomcat-7.0.61/webapps/samples"
37       state: directory
38       mode: 0777
39     become: true
40
41
42   - name: copy war file
43     copy: src=./target/LoginWebApp-1.war dest=/opt/apache-tomcat-7.0.61/webapps/
44
45
46   notify: restart tomcat
47
48   - name: Install Tomcat init script
49     copy: src=tomcat-initscript.sh dest=/etc/init.d/tomcat.sh mode=0755
50
51   - name: Start Tomcat
52     shell: /etc/init.d/tomcat.sh start
53
54   - name: wait for tomcat to start
55     wait_for: port={{http_port}}

```

localhost:8080/job/ansible_proj_vignesh_dharmaraj/

Stage Logs (Ansible Deploy)

- Use a tool from a predefined Tool Installation -- Maven (self time 24ms)
- Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step. (self time 33ms)
- Shell Script -- ansible-playbook main.yml -i inventories/dev/hosts (self time 16s)
 - ok: [172.31.19.22]
 - ok: [172.31.22.46]

TASK [tomcat : Start Tomcat] *****
changed: [172.31.22.46]
changed: [172.31.19.22]

Started tomcat in both nodes

TASK [tomcat : Wait for tomcat to start] *****
ok: [172.31.22.46]
ok: [172.31.19.22]

RUNNING HANDLER [tomcat : restart tomcat] *****
changed: [172.31.22.46]
changed: [172.31.19.22]

Restarted tomcat in both the nodes

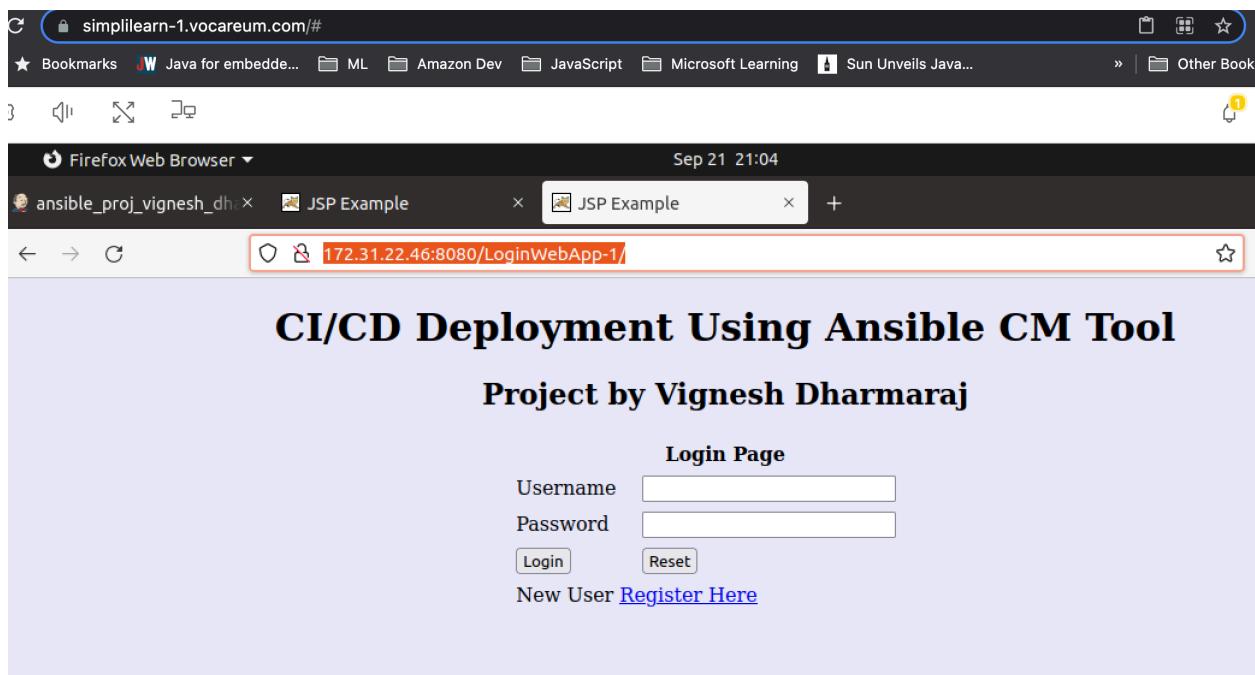
PLAY RECAP *****
172.31.19.22 : ok=15 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
172.31.22.46 : ok=15 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

#20 Sep 21, 2021 7:08 PM	#19 Sep 21, 2021 2:39 PM	14:39	1 commit	425ms	61ms	378ms	741ms	4s	14s
--------------------------	--------------------------	-------	----------	-------	------	-------	-------	----	-----

Result 4:

Deployed application with the changes can be tested by accessing the URL

http://<node_ip>:8080/LoginWebApp-1/



6. Conclusion

All the below objectives is achieved and verified.

1. Configure Jenkins server as Ansible provisioning machine
2. Install Ansible plugins in Jenkins CI server
3. Prepare Ansible playbook to run Maven build on Jenkins CI server
4. Prepare Ansible playbook to execute deployment steps on the remote web container with restart of the web container post deployment

We have automated the WAR file deployment using Ansible and Jenkins.

CI server Jenkins had ansible playbooks to do the maven build and the deployment to remote hosts including install/start/restart of the tomcat server is also done by ansible playbook & ansible role.

Finally we had seen all the evidences to Automate Ansible integration with Jenkins CI server so that we can run and execute playbooks to deploy custom WAR files to a web container and then perform restart for the web container.