Sunday, February 28, 2021   10:42 AM

CI/CD Deployment Using Ansible CM Tool

DESCRIPTION

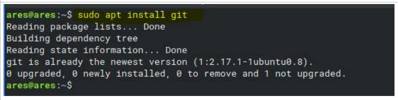
You are a DevOps engineer at XYZ Ltd. Your company is working on a Java application and wants to automate WAR file artifact deployment so that they don't have to perform WAR deployment on Tomcat/Jetty web containers. Automate Ansible integration with Jenkins CI server so that we can run and execute playbooks to deploy custom WAR files to a web container and then perform restart for the web container.

Steps to Perform:

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

INSTALLING GIT

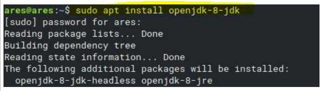
$ sudo apt update -y => this will update your repository
$ sudo apt install git => this will install the latest version of git



```
ares@ares:~$ sudo apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.8).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
ares@ares:~$
```

INSTALLING JAVA 8

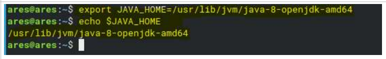
$ sudo apt install openjdk-8-jdk => this will install Java 8 in your Ubuntu 18.04LTS System



```
ares@ares:~$ sudo apt install openjdk-8-jdk
[sudo] password for ares:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openjdk-8-jdk-headless openjdk-8-jre
```

SETTING UP JAVA_HOME

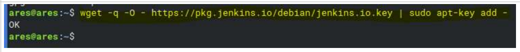
$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64 => this will import JAVA_HOME
$ echo JAVA_HOME => to verify JAVA_HOME path



```
ares@ares:~$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
ares@ares:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64
ares@ares:~$
```

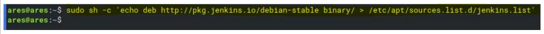
INSTALLING JENKINS

$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add - => this will add the key



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

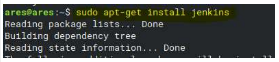
$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list' => this will add the url to your repo list



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

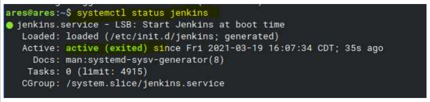
$ sudo apt update -y => this will update your repository

$ sudo apt install jenkins => this will install jenkins



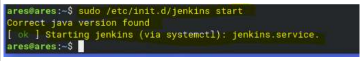
```
ares@ares:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

$ systemctl status jenkins => to check status of Jenkins service



```
ares@ares:~$ systemctl status jenkins
● jenkins.service - LSB: Start Jenkins at boot time
   Loaded: loaded (/etc/init.d/jenkins; generated)
   Active: active (exited) since Fri 2021-03-19 16:07:34 CDT; 35s ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0 (limit: 4915)
   CGroup: /system.slice/jenkins.service
```

$ sudo /etc/init.d/jenkins start => to launch jenkins at start up



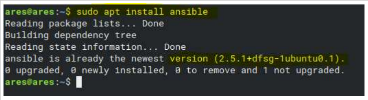
```
ares@ares:~$ sudo /etc/init.d/jenkins start
Correct java version found
[ ok ] Starting jenkins (via systemctl): jenkins.service.
ares@ares:~$
```

=> Once Jenkins in installed in your system, you can access Jenkins using the following URL in any web browser

http://localhost:8080 => Jenkins by default will be using port 8080

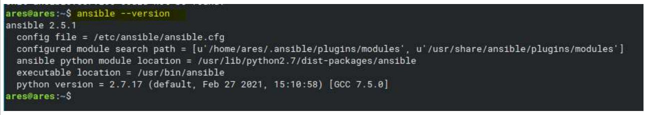
INSTALLING ANSIBLE

$ sudo apt update -y => this will update your repository
$ sudo apt install ansible => this will install ansible in your system



```
ares@ares:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
ansible is already the newest version (2.5.1+dfsg-1ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
ares@ares:~$
```

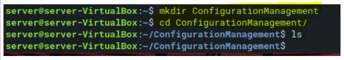
$ sudo ansible --version => to check the version of Ansible installed in your system



```
ares@ares:~$ ansible --version
ansible 2.5.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/ares/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.17 (default, Feb 27 2021, 15:10:58) [GCC 7.5.0]
ares@ares:~$
```

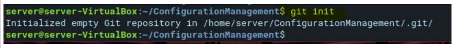
CREATING JENKINS FILE AND ANSIBLE PLAYBOOK

$ mkdir ConfigurationManagement => this will create a directory called ConfigurationManagement
$ cd ConfigurationManagement => to go inside the directory



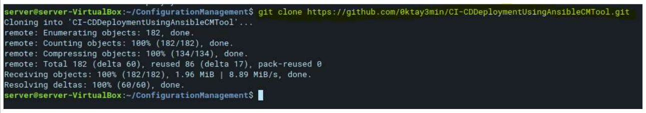
```
server@server-VirtualBox:~$ mkdir ConfigurationManagement
server@server-VirtualBox:~$ cd ConfigurationManagement/
server@server-VirtualBox:~/ConfigurationManagement$ ls
server@server-VirtualBox:~/ConfigurationManagement$
```

$ git init => to initialize git



```
server@server-VirtualBox:~/ConfigurationManagement$ git init
Initialized empty Git repository in /home/server/ConfigurationManagement/.git/
server@server-VirtualBox:~/ConfigurationManagement$
```

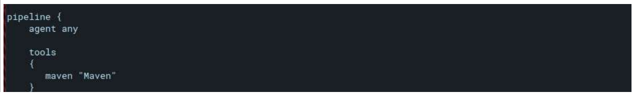
$ git clone https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git => to clone the git repo where java files are located for .war file



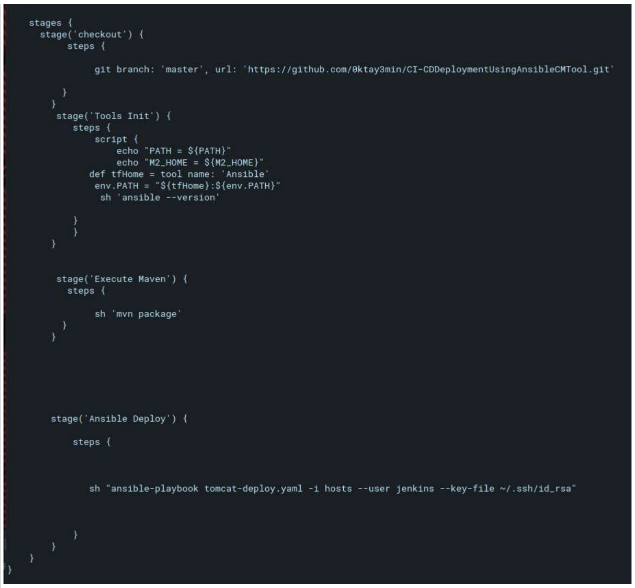
```
server@server-VirtualBox:~/ConfigurationManagement$ git clone https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git
Cloning into 'CI-CDDeploymentUsingAnsibleCMTool'...
remote: Enumerating objects: 182, done.
remote: Counting objects: 100% (182/182), done.
remote: Compressing objects: 100% (134/134), done.
remote: Total 182 (delta 60), reused 86 (delta 17), pack-reused 0
Receiving objects: 100% (182/182), 1.96 MiB | 8.89 MiB/s, done.
Resolving deltas: 100% (60/60), done.
server@server-VirtualBox:~/ConfigurationManagement$
```

$ sudo nano Jenkinsfile => This will create a Jenkinsfile. Enter the following to Jenkinsfile pipeline code below, save and exit.

```
pipeline {
        agent any
        tools
```



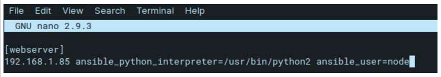
```
pipeline {
    agent any

    tools
    {
       maven "Maven"
    }
{
maven "Maven"
}
stages {
stage('checkout') {
steps {
git branch: 'master', url: 'https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git'
}
}
stage('Tools Init') {
steps {
script {
echo "PATH = ${PATH}"
echo "M2_HOME = ${M2_HOME}"
def tfHome = tool name: 'Ansible'
env.PATH = "${tfHome}:${env.PATH}"
sh 'ansible --version'
}
}
}
stage('Execute Maven') {
steps {
sh 'mvn package'
}
}
stage('Ansible Deploy') {
steps {
sh "ansible-playbook tomcat-deploy.yaml -i hosts --user jenkins --key-file ~/.ssh/id_rsa"
}
}
}
}
```



```
  stages {
    stage('checkout') {
         steps {

              git branch: 'master', url: 'https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git'

        }
      }
       stage('Tools Init') {
          steps {
              script {
                  echo "PATH = ${PATH}"
                  echo "M2_HOME = ${M2_HOME}"
             def tfHome = tool name: 'Ansible'
              env.PATH = "${tfHome}:${env.PATH}"
               sh 'ansible --version'

          }
          }
      }


       stage('Execute Maven') {
         steps {

              sh 'mvn package'
        }
      }




      stage('Ansible Deploy') {

          steps {



             sh "ansible-playbook tomcat-deploy.yaml -i hosts --user jenkins --key-file ~/.ssh/id_rsa"



          }
      }
  }
}
```

$ sudo nano hosts => this will create a file called hosts. Make sure you enter your node machine's IP address. Save the file and exit

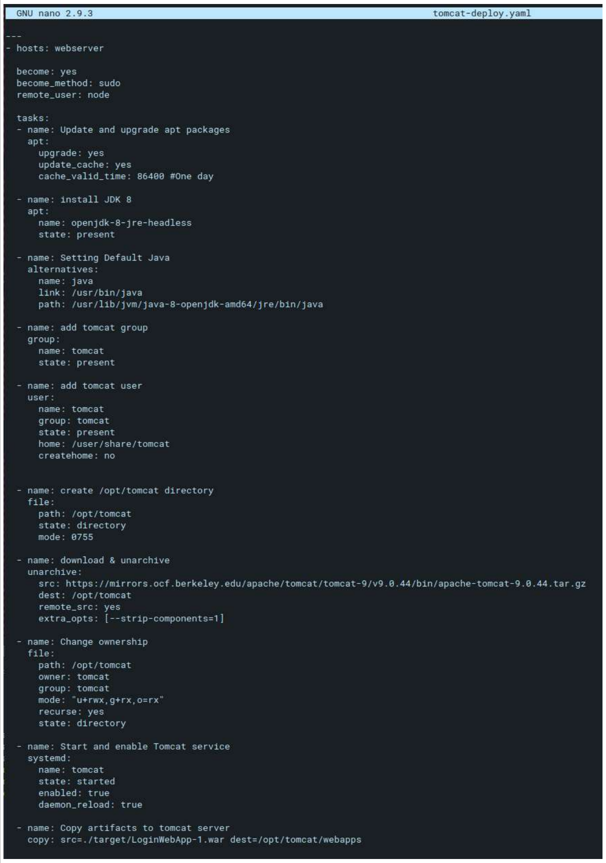
```
[webserver]
192.168.1.85 ansible_python_interpreter=/usr/bin/python2 ansible_user=node
```



```
[webserver]
192.168.1.85 ansible_python_interpreter=/usr/bin/python2 ansible_user=node
```

$ sudo nano tomcat-deploy.yaml => this will create a file called tomcat-deploy.yaml for ansible playbook. Copy the following data below save the file and exit

```
---
- hosts: webserver
become: yes
become_method: sudo
remote_user: node
tasks:
- name: Update and upgrade apt packages
apt:
upgrade: yes
update_cache: yes
cache_valid_time: 86400 #One day
- name: install JDK 8
apt:
name: openjdk-8-jre-headless
state: present
- name: Setting Default Java
alternatives:
name: java
link: /usr/bin/java
path: /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
- name: add tomcat group
group:
name: tomcat
state: present
- name: add tomcat user
user:
name: tomcat
group: tomcat
state: present
home: /user/share/tomcat
createhome: no
- name: create /opt/tomcat directory
file:
path: /opt/tomcat
state: directory
mode: 0755
- name: download & unarchive
unarchive:
src: https://mirrors.ocf.berkeley.edu/apache/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz
dest: /opt/tomcat
remote_src: yes
extra_opts: [--strip-components=1]
- name: Change ownership
file:
path: /opt/tomcat
owner: tomcat
group: tomcat
mode: "u+rwx,g+rx,o=rx"
recurse: yes
```

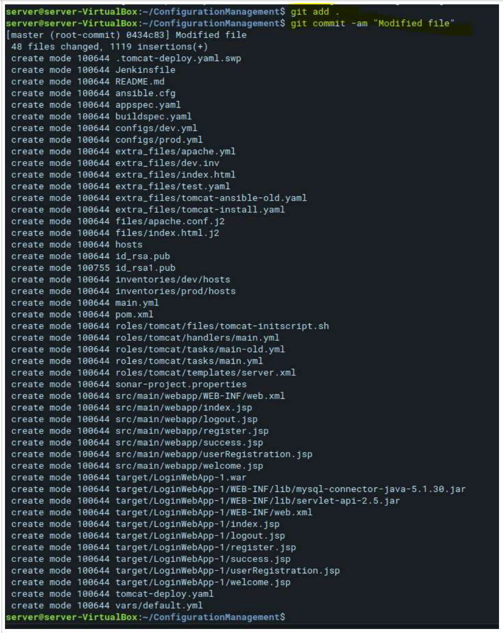


```
---
- hosts: webserver

  become: yes
  become_method: sudo
  remote_user: node

  tasks:
  - name: Update and upgrade apt packages
    apt:
      upgrade: yes
      update_cache: yes
      cache_valid_time: 86400 #One day

  - name: install JDK 8
    apt:
      name: openjdk-8-jre-headless
      state: present

  - name: Setting Default Java
    alternatives:
      name: java
      link: /usr/bin/java
      path: /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java

  - name: add tomcat group
    group:
      name: tomcat
      state: present

  - name: add tomcat user
    user:
      name: tomcat
      group: tomcat
      state: present
      home: /user/share/tomcat
      createhome: no


  - name: create /opt/tomcat directory
    file:
      path: /opt/tomcat
      state: directory
      mode: 0755

  - name: download & unarchive
    unarchive:
      src: https://mirrors.ocf.berkeley.edu/apache/tomcat/tomcat-9/v9.0.44/bin/apache-tomcat-9.0.44.tar.gz
      dest: /opt/tomcat
      remote_src: yes
      extra_opts: [--strip-components=1]

  - name: Change ownership
    file:
      path: /opt/tomcat
      owner: tomcat
      group: tomcat
      mode: "u+rwx,g+rx,o=rx"
      recurse: yes
      state: directory

  - name: Start and enable Tomcat service
    systemd:
      name: tomcat
      state: started
      enabled: true
      daemon_reload: true

  - name: Copy artifacts to tomcat server
    copy: src=./target/LoginWebApp-1.war dest=/opt/tomcat/webapps
```

```
state: directory
- name: Start and enable Tomcat service
systemd:
name: tomcat
state: started
enabled: true
daemon_reload: true
- name: Copy artifacts to tomcat server
copy: src=./target/LoginWebApp-1.war dest=/opt/tomcat/webapps
```

PUSHING YOUR CODE TO GITHUB

$ git add . => to add all files and directories to staging area
$ git commit -am "initial commit" => to commit changes

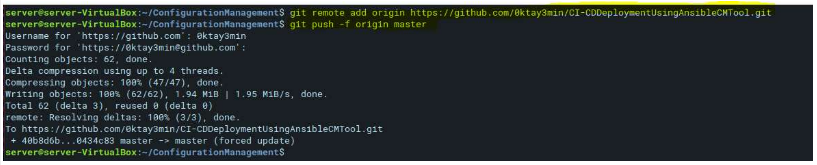


```
server@server-VirtualBox:~/ConfigurationManagement$ git add .
server@server-VirtualBox:~/ConfigurationManagement$ git commit -am "Modified file"
[master (root-commit) 0434c83] Modified file
 48 files changed, 1119 insertions(+)
 create mode 100644 .tomcat-deploy.yaml.swp
 create mode 100644 Jenkinsfile
 create mode 100644 README.md
 create mode 100644 ansible.cfg
 create mode 100644 appspec.yaml
 create mode 100644 buildspec.yaml
 create mode 100644 configs/dev.yml
 create mode 100644 configs/prod.yml
 create mode 100644 extra_files/apache.yml
 create mode 100644 extra_files/dev.inv
 create mode 100644 extra_files/index.html
 create mode 100644 extra_files/test.yaml
 create mode 100644 extra_files/tomcat-ansible-old.yaml
 create mode 100644 extra_files/tomcat-install.yaml
 create mode 100644 files/apache.conf.j2
 create mode 100644 files/index.html.j2
 create mode 100644 hosts
 create mode 100644 id_rsa.pub
 create mode 100755 id_rsa1.pub
 create mode 100644 inventories/dev/hosts
 create mode 100644 inventories/prod/hosts
 create mode 100644 main.yml
 create mode 100644 pom.xml
 create mode 100644 roles/tomcat/files/tomcat-initscript.sh
 create mode 100644 roles/tomcat/handlers/main.yml
 create mode 100644 roles/tomcat/tasks/main-old.yml
 create mode 100644 roles/tomcat/tasks/main.yml
 create mode 100644 roles/tomcat/templates/server.xml
 create mode 100644 sonar-project.properties
 create mode 100644 src/main/webapp/WEB-INF/web.xml
 create mode 100644 src/main/webapp/index.jsp
 create mode 100644 src/main/webapp/logout.jsp
 create mode 100644 src/main/webapp/register.jsp
 create mode 100644 src/main/webapp/success.jsp
 create mode 100644 src/main/webapp/userRegistration.jsp
 create mode 100644 src/main/webapp/welcome.jsp
 create mode 100644 target/LoginWebApp-1.war
 create mode 100644 target/LoginWebApp-1/WEB-INF/lib/mysql-connector-java-5.1.30.jar
 create mode 100644 target/LoginWebApp-1/WEB-INF/lib/servlet-api-2.5.jar
 create mode 100644 target/LoginWebApp-1/WEB-INF/web.xml
 create mode 100644 target/LoginWebApp-1/index.jsp
 create mode 100644 target/LoginWebApp-1/logout.jsp
 create mode 100644 target/LoginWebApp-1/register.jsp
 create mode 100644 target/LoginWebApp-1/success.jsp
 create mode 100644 target/LoginWebApp-1/userRegistration.jsp
 create mode 100644 target/LoginWebApp-1/welcome.jsp
 create mode 100644 tomcat-deploy.yaml
 create mode 100644 vars/default.yml
server@server-VirtualBox:~/ConfigurationManagement$
```

$ git remote add origin https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git => to add git remote URL

$ git push -f origin master => this will push all changes to your remote repository

Please Note: You will need to enter your GitHub login credentials.

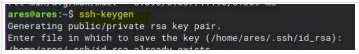


```
server@server-VirtualBox:~/ConfigurationManagement$ git remote add origin https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git
server@server-VirtualBox:~/ConfigurationManagement$ git push -f origin master
Username for 'https://github.com': 0ktay3min
Password for 'https://0ktay3min@github.com':
Counting objects: 62, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (47/47), done.
Writing objects: 100% (62/62), 1.94 MiB | 1.95 MiB/s, done.
Total 62 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git
 + 40b8d6b...0434c83 master -> master (forced update)
server@server-VirtualBox:~/ConfigurationManagement$
```

SSH CONNECTION BETWEEN 2 MACHINES

Master machine

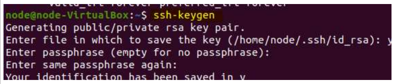
$ ssh-keygen => this will create a rsa key inside .ssh directory



```
ares@ares:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ares/.ssh/id_rsa):
/home/ares/.ssh/id_rsa already exists.
```

Node machine

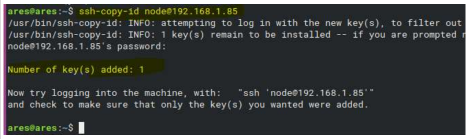
$ ssh-keygen => this will create a rsa key inside .ssh directory



```
    valid_lft forever preferred_lft forever
node@node-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/node/.ssh/id_rsa): y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in y
```

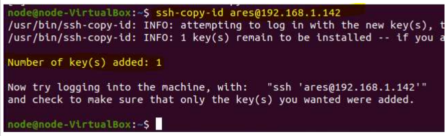
Now Copy id_rsa.pub between Master and Node machines

$ ssh-copy-id node@192.168.1.85 => this will copy id_rsa.pub from master machine to node machine.



```
ares@ares:~$ ssh-copy-id node@192.168.1.85
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted r
node@192.168.1.85's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'node@192.168.1.85'"
and check to make sure that only the key(s) you wanted were added.

ares@ares:~$
```

$ ssh-copy-id ares@192.168.1.85 => this will copy id_rsa.pub from master machine to node machine.



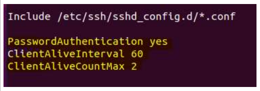
```
node@node-VirtualBox:~$ ssh-copy-id ares@192.168.1.142
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), t
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you a

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ares@192.168.1.142'"
and check to make sure that only the key(s) you wanted were added.

node@node-VirtualBox:~$
```

Modify the sshd_config file on both machines

$ sudo nano /etc/ssh/sshd_config => to open sshd_config file and add the following

PasswordAuthentication yes
ClientAliveInterval 60
ClientAliveCountMax 2

$ sudo systemctl restart sshd => to restart sshd services

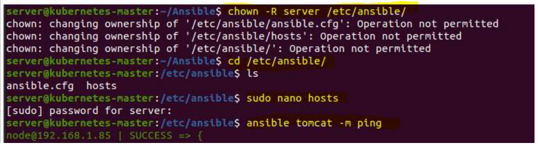


```
Include /etc/ssh/sshd_config.d/*.conf

PasswordAuthentication yes
ClientAliveInterval 60
ClientAliveCountMax 2
```

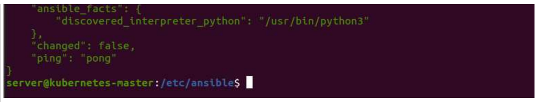
CONFIGURING JENKINS USER

$ sudo -su jenkins => to switch to "jenkins" user
$ ssh-keygen =>to generate ssh key
$ eval $(ssh-agen -s) => to make sure ssh-agent is running in your system
$ ssh-add ~/.ssh/id_rsa => to add ssh-key
$ sudo chown -R server /etc/ansible => to give directory permission to user called server to perform any changes in ansible directory

$ cd /etc/ansible => to see all files and directories inside Ansible directory



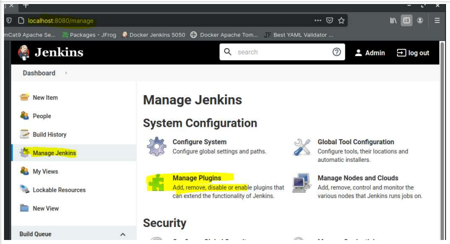
```
server@kubernetes-master:~/Ansible$ chown -R server /etc/ansible/
chown: changing ownership of '/etc/ansible/ansible.cfg': Operation not permitted
chown: changing ownership of '/etc/ansible/hosts': Operation not permitted
chown: changing ownership of '/etc/ansible/': Operation not permitted
server@kubernetes-master:~/Ansible$ cd /etc/ansible/
server@kubernetes-master:/etc/ansible$ ls
ansible.cfg  hosts
server@kubernetes-master:/etc/ansible$ sudo nano hosts
[sudo] password for server:
server@kubernetes-master:/etc/ansible$ ansible tomcat -m ping
node@192.168.1.85 | SUCCESS => {
```



```
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
server@kubernetes-master:/etc/ansible$
```

INSTALLING MAVEN AND ANSIBLE PLUGINS IN JENKINS

1- Go to Jenkins using your browser and enter http://localhost:8080
2- Click "Manage Jenkins"
3- Click "Manage Plugins"
4- Click "Available" tab and then search for Maven.
5- Select "Maven Integration" and then press "Install without restart" button
6- Perform the same steps for Ansible plugin



Manage Jenkins

System Configuration

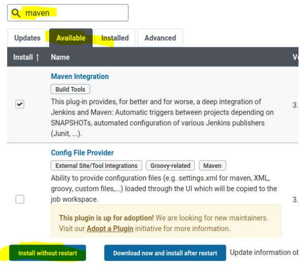
Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

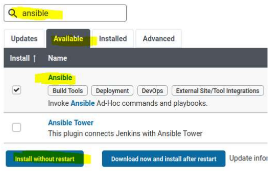


☑ Maven Integration — This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).

□ Config File Provider — Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace.

This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.

Update information ol



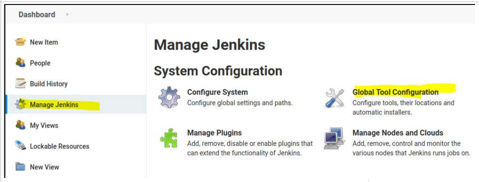
☑ Ansible — Invoke Ansible Ad-Hoc commands and playbooks.

□ Ansible Tower — This plugin connects Jenkins with Ansible Tower

Update infor

CONFIGURING MAVEN AND ANSIBLE IN JENKINS

1- Click "Manage Jenkins"
2- Click "Global Tool Configuration"
3- Click "Add Maven" and enter any name
4- Select "Install Automatically" so Jenkins will use any Maven version available. If you want to configure your own maven, you will have to provide maven home directory information.
5- Click "Add Ansible" and then give any name "Ansible". The same name must go inside the Jenkins file that we will be pulling from Github repository



Manage Jenkins

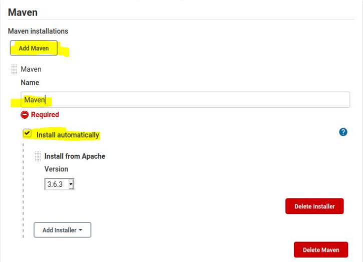
System Configuration

Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

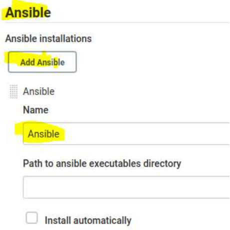
Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.



Maven

Maven installations

Maven
Name: Maven
Required
☑ Install automatically
Install from Apache
Version: 3.6.3



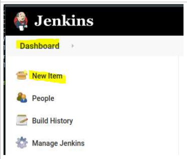
Ansible

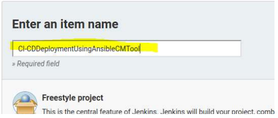
Ansible installations

Ansible
Name: Ansible
Path to ansible executables directory:
□ Install automatically

CREATING PIPELINE PROJECT

1- Click "New Item" on Jenkins Dashboard





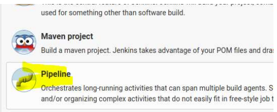
Enter an item name

CI-CDDeploymentUsingAnsibleCMTool

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, comb

2- Enter an item name and press Pipeline Project tab and click "OK"



used for something other than software build.

Maven project

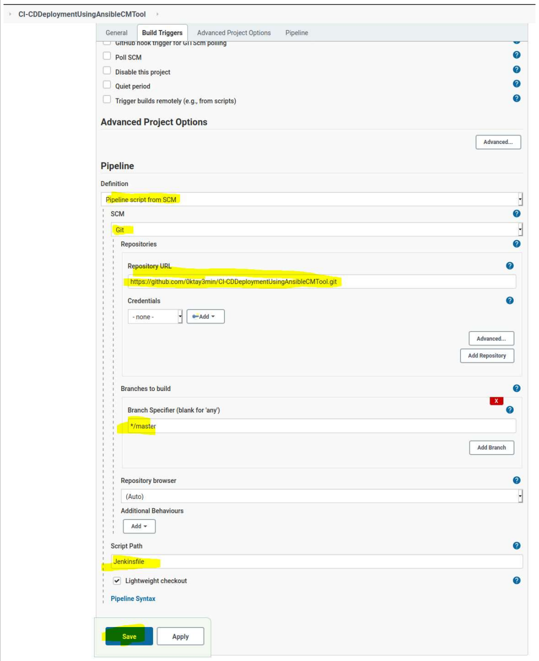
Build a maven project. Jenkins takes advantage of your POM files and dra

Pipeline

Orchestrates long-running activities that can span multiple build agents. S and/or organizing complex activities that do not easily fit in free-style job t

----

3- In the Project management windows go to Build Triggers and select "Pipeline script from SCM"
4- Select Git from SCM menu and copy your GitHub Project URL
5- Enter your login credentials if there are any.
6- Select your branch name and press "Save" button.



CI-CDDeploymentUsingAnsibleCMTool

□ GitHub hook trigger for GITScm polling
□ Poll SCM
□ Disable this project
□ Quiet period
□ Trigger builds remotely (e.g., from scripts)

Advanced Project Options

Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/0ktay3min/CI-CDDeploymentUsingAnsibleCMTool.git

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

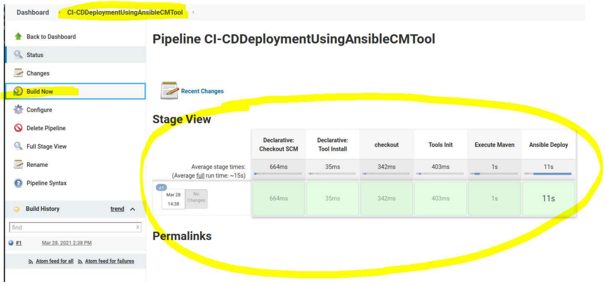
Repository browser: (Auto)

Additional Behaviours

Script Path: Jenkinsfile

☑ Lightweight checkout

7- Go back to your project and click "Build Now" button



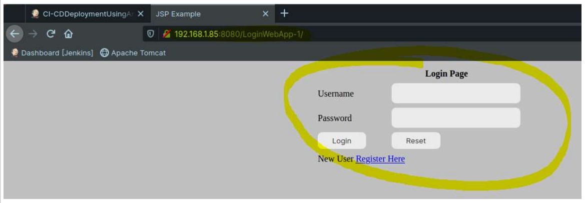
Pipeline CI-CDDeploymentUsingAnsibleCMTool

Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	checkout	Tools Init	Execute Maven	Ansible Deploy
Average stage times: (Average full run time: ~15s)	664ms	35ms	342ms	403ms	1s	11s
#1 Mar 28 14:38 No Changes	664ms	35ms	342ms	403ms	1s	11s

#1 Mar 28, 2021 2:38 PM

8- Open you browser and check the deployed web project
Project URL is your node IP address, port number and .war name



192.168.1.85:8080/LoginWebApp-1/

Login Page

Username

Password

New User Register Here

EX: http://192.168.1.85:8080/LoginWebApp-1/