How to Jenkins-Ansible integration.

Ansible configurations:-

Agenda: Create a ansible role for deploying the artifacts from nexus to destination.

1. first we need to install the ansible in the linux machine i.e ubuntu/centos, for package mangement may vary based on flavours of linux. Ansible is not supporting for windows.

```
On ubuntu
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo apt-add-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
On Fedora:
$ sudo dnf install ansible
On RHEL and CentOS:
$ sudo yum install ansible
```

2. Check the version.

```
root@ip-address:/home/ubuntu# ansible --version
ansible 2.8.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.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 = 2.7.12 (default, Oct  8 2019, 14:14:10) [GCC 5.4.0
20160609]
```

3.Please install the supported python libraries for Ansible, i.e lxml etc.

4. Ansible related folders.

```
$ whereis ansible
ansible: /usr/bin/ansible ---> bin folder
/etc/ansible   ----> ansible home deafult directory
/usr/share/man/man1/ansible.1.gz ---> check the sub commands.
```

5.Go to ansible home directory.

```
ansible.cfg  ---> configuration related to ansible configurations like
below
#inventory      = /etc/ansible/hosts
#library        = /usr/share/my_modules/
#module_utils   = /usr/share/my_module_utils/
#remote_tmp     = ~/.ansible/tmp
#local_tmp      = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks          = 5
#poll_interval  = 15
#sudo_user      = root
#ask_sudo_pass = True
#ask_pass      = True
#transport      = smart
#remote_port    = 22
#module_lang    = C
#module_set_locale = False
------------------------------------------------------------------------
------------
hosts        ---> you can store the hosts here.
roles        ----> defaults role can install here.
```

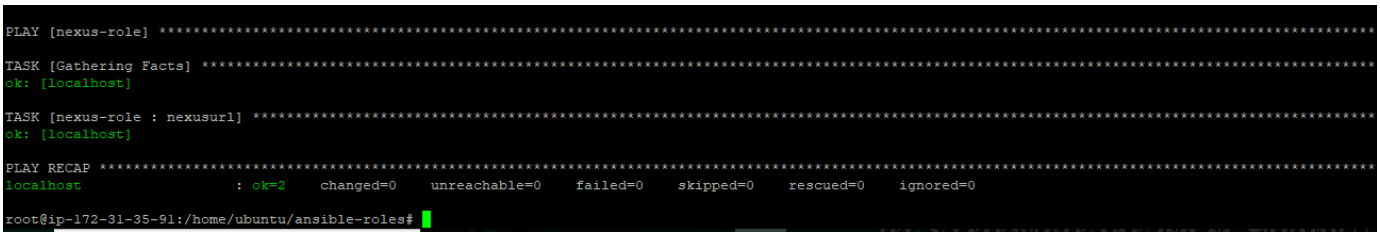
6. Create a file with naming playbook.yml and structure is below.

```
---
  - hosts: localhost  ---> hosts which you mentioned in inventoryfile
    name: nexus-role
    gather_facts: true
    roles:
      - nexus-role
    vars:
      versioning: latest
```

7. Create a role and struture is below.

```
$ ansible-galaxy init nexus-role.
$ tree
.
 defaults
    main.yml
 files
 handlers
    main.yml
 meta
    main.yml
 README.md
 tasks
    main.yml
 templates
 tests
    inventory
    test.yml
 vars
     main.yml
this yml files are defaults file for placing the tasks, vars, templates,
handlers.
```

8. Try to run the playbook with mapping role.



9. I'm trying to deploy the nexus artifacts from nexus server to destination.

```
task main.yml:- we are using WAR files in the TOMCAT destination
---
-
  maven_artifact:
    artifact_id: consumerBanking
    dest: $(TOMCAT_HOME)/webapps/    ----> mentions tomcat home
directory& hit the tomcat url localhost:8080/consumerbanking
    group_id: com.companyname.bank
    password: {{ servicetoken }}
    repository_url: "http://localhost/repository/novartis-maven"
    username: admin
    version: {{ versioning}}
  name: nexusurl

var main.yml:-
---
servicetoken: XXXXXXXXXXXXXXXX
```

```
task main.yml:- we are using JAR files in the Application destination
---
-
  maven_artifact:
    artifact_id: consumerBanking
    dest: $(Application path) ---> specify the path of destination to
deploy
    group_id: com.companyname.bank
    password: {{ servicetoken }}
    repository_url: "http://localhost/repository/novartis-maven"
    username: admin
    version: {{ versioning}}
  name: nexusurl
- name: deploying
  shell: java -jar consumerBanking-latest.jar    ---> specify the
location and name of jar file

var main.yml:-
---
servicetoken: XXXXXXXXXXXXXXXX
```

10. Now our task is done, next we can integrate with Jenkins and implement the same.

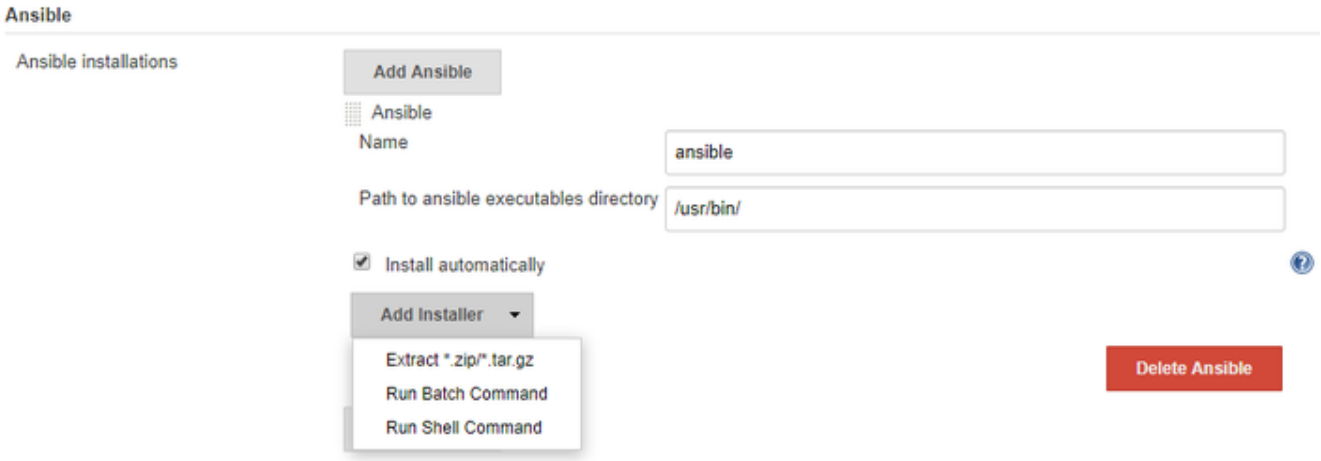
Jenkins configurations:-

Jenkins-Ansible(Master node) configuration,

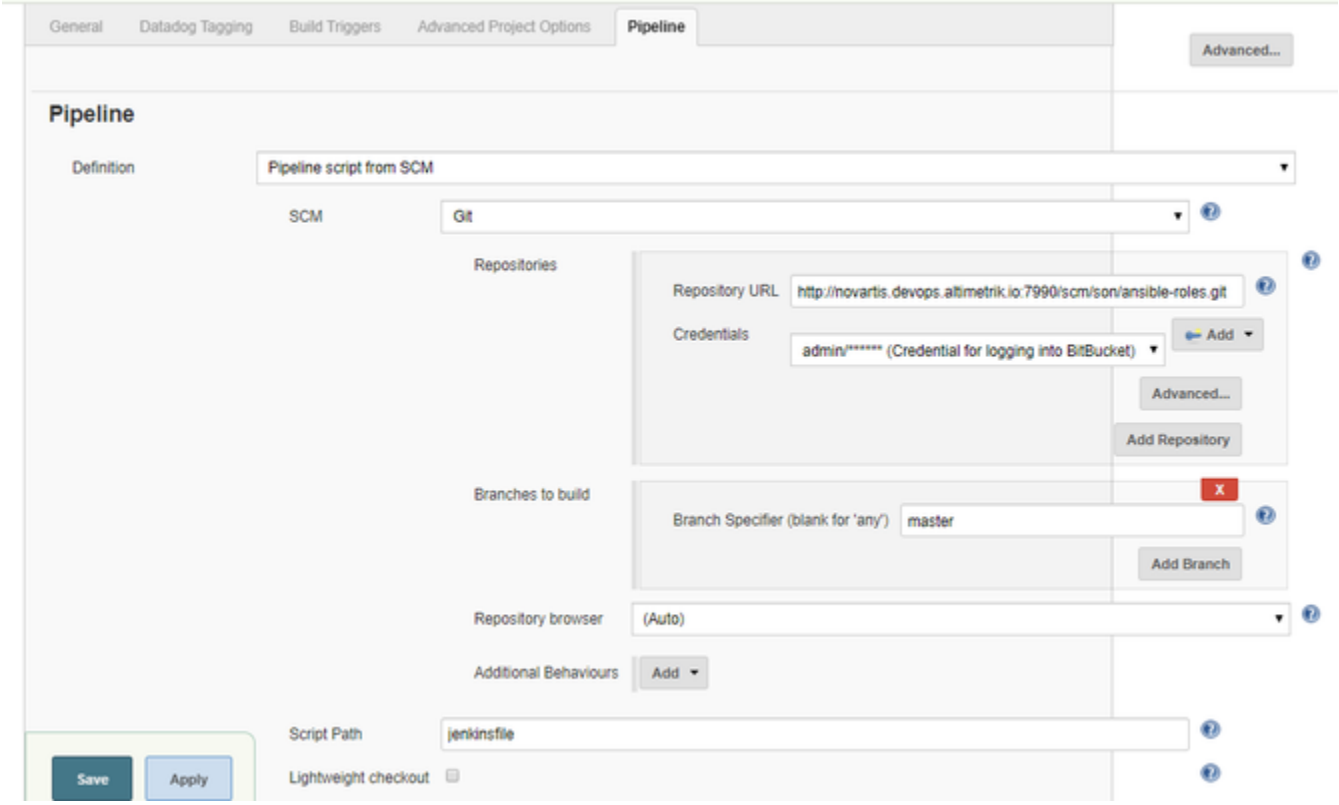
Note: Ansible should install in master node. Ansible-playbook will run in master node.

1. start the Jenkins server and install the Ansible plugin.

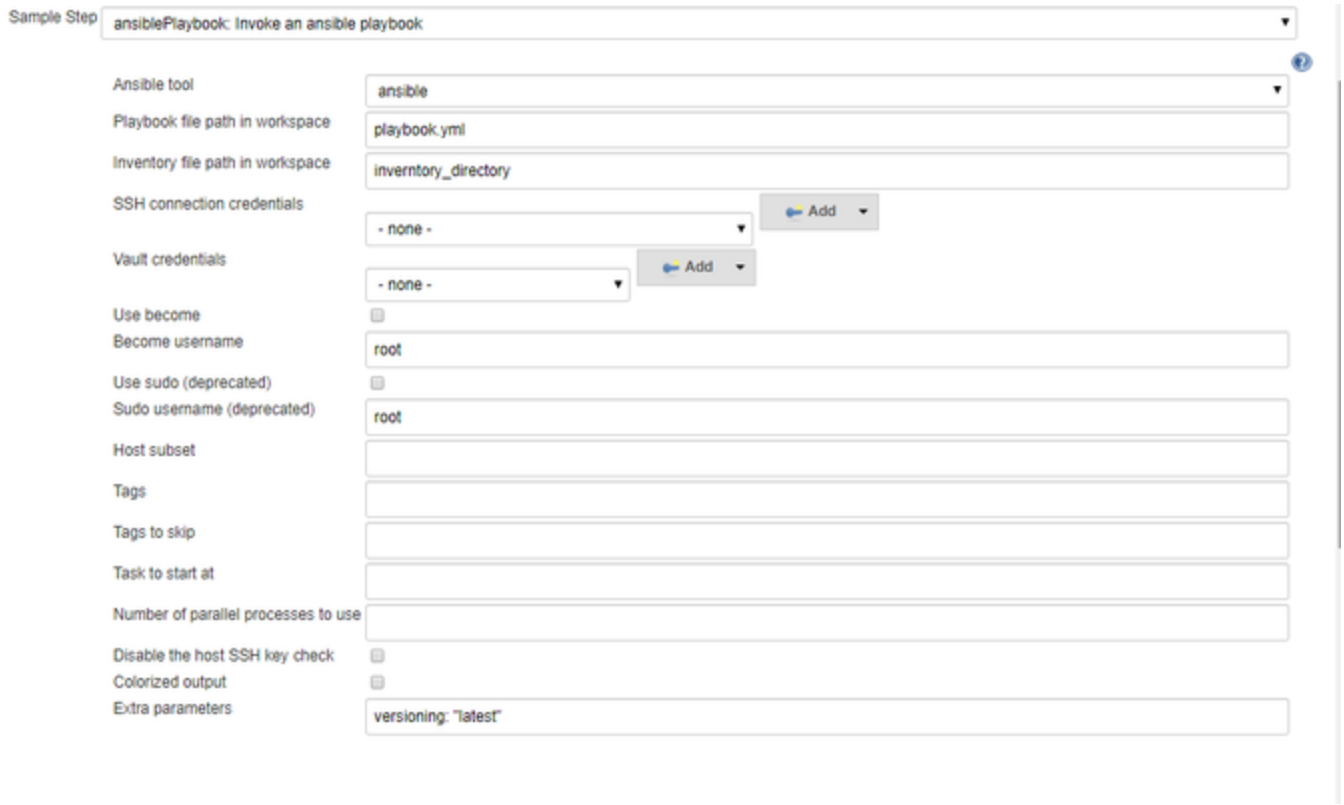
2. Configure the Ansible as below pic.



3. Create a pipeline job and add the declarative pipeline , add repository, map the jenkinfile.

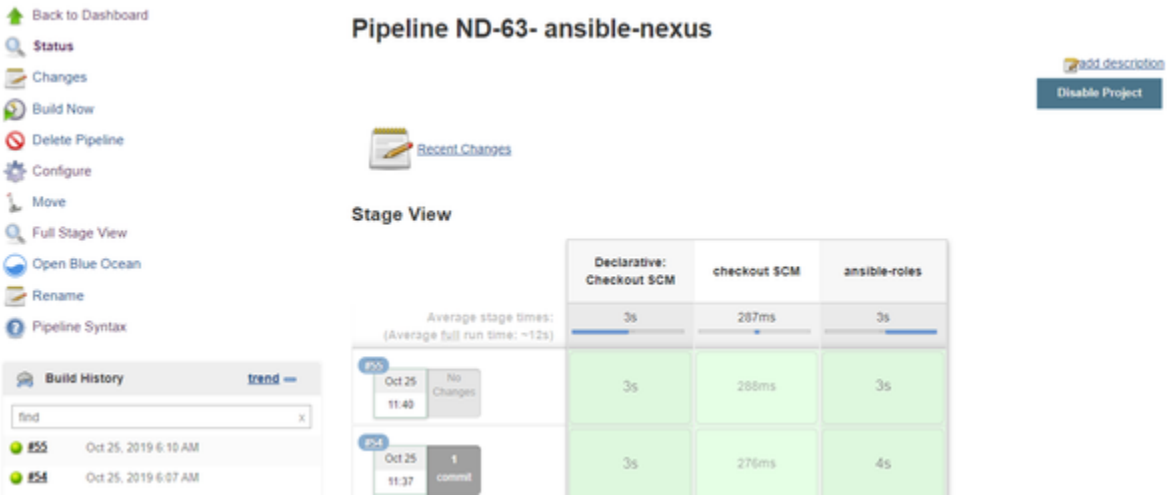


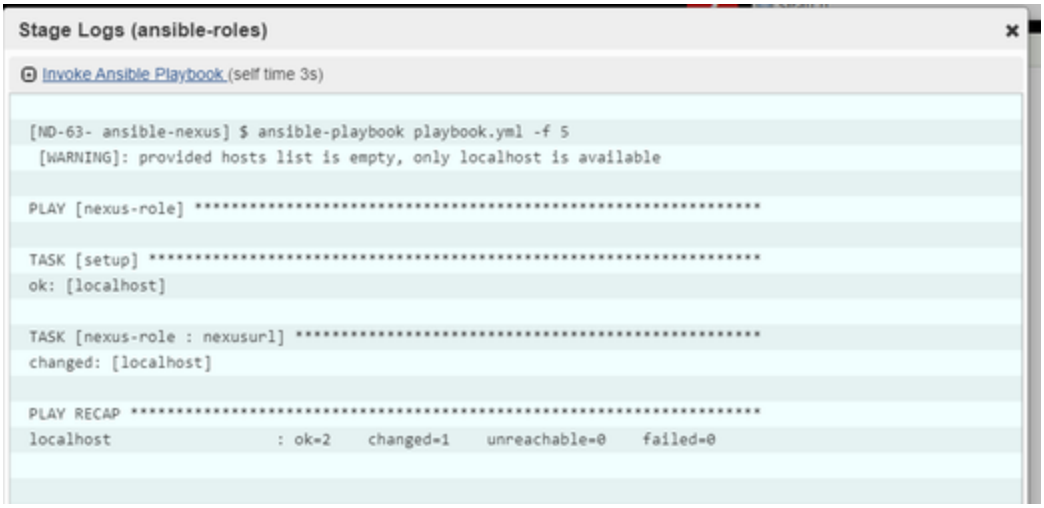
4. For declarative synatx, add the ansible playbook step and generate the syntax, add in jenkinsfile.



Declarative pipeline for the Ansible-playbook, please find the code in below code string.

6. Build the pipeline job and see the output, check the logs by going to corresponding logs.





7. The output of the build stage can be observe in the job dashboard.

Jenkins-Ansible(Docker slave node) configuration,

Note: Ansible should install in Docker node. Ansible-playbook will run in container node, JDK is mandatory in order to run the slaves in any nodes.

For more details, please reference the this document - How To - Elastic Ephemeral Build Jenkins Slaves.

- **Note:** if you are using public docker image just mentions name, else you can create customized Dockerfile, please test it and push to registry, right-now i'm using customized docker file in my local, find below code.

```
FROM jenkinsci/slave:3.35-3
USER root
RUN apt-get update && \
   apt-get install --no-install-recommends -y \
   software-properties-common \
   ansible \
   && rm -rf /tmp/downloaded_packages/ /tmp/*.rds \
   && rm -rf /var/lib/apt/lists/* \
   && apt-get install -y python-lxml

USER jenkins
```

Jenkins-job execution:

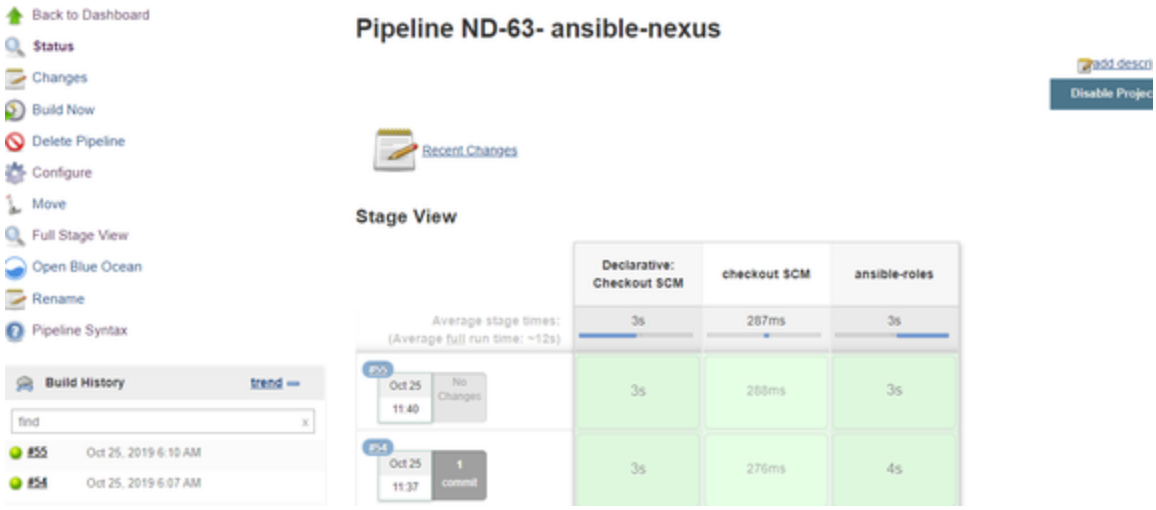
- Create a pipeline job and map the repository and jenkinsfile as i show above.
- Write the Jenkins file as below, mention the node label.

```
pipeline {
 agent { label 'ansible-slave' }  ------> docker slave which i
mentioned in docker node template.
 stages {
  stage("checkout SCM") {
   steps {
    checkout changelog: false, scm: [$class: 'GitSCM', branches:
[[name: '*/master']], doGenerateSubmoduleConfigurations: false,
extensions: [], submoduleCfg: [], userRemoteConfigs:
[[credentialsId: 'BitCred', url:
'http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.gi
t']]]
   }
  }
  stage("ansible-roles") {
   steps {
   step([$class: 'AnsiblePlaybookBuilder', additionalParameters:
'', ansibleName: 'ansibles', becomeUser: '', credentialsId: '',
forks: 5, inventory: [$class: 'InventoryDoNotSpecify'], limit: '',
playbook: 'playbook.yml', skippedTags: '', startAtTask: '',
sudoUser: '', tags: '', vaultCredentialsId: ''])
            //step {
             //sh 'ansible-playbook playbook.yml'
            //}
   }
  }
 }
}
```

see the outputs in the build section, console output.

Console Output

```
Started by user admin
Checking out git http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git into /var/jenkins_home/workspace/ND-63- ansible-nexus@script to read jenkinsfile
using credential BitCred
 > git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git # timeout=10
Fetching upstream changes from http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git
 > git --version # timeout=10
using GIT_ASKPASS to set credentials Credential for logging into BitBucket
 > git fetch --tags --progress -- http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git +refs/heads/*:refs/remotes/origin/*
 > git rev-parse origin/master^{commit} # timeout=10
Checking out Revision 46a80cbce16d204c6593df9c3e9faad0835dc926 (origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 46a80cbce16d204c6593df9c3e9faad0835dc926
Commit message: "main.yml edited online with Bitbucket"
 > git rev-list --no-walk 46a80cbce16d204c6593df9c3e9faad0835dc926 # timeout=10
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on ansible-slave-0000jx5cn4kpg on docker1 in /home/jenkins/workspace/ND-63- ansible-nexus
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
using credential BitCred
Cloning the remote Git repository
Cloning repository http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git
 > git init /home/jenkins/workspace/ND-63- ansible-nexus # timeout=10
Fetching upstream changes from http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git
 > git --version # timeout=10
using GIT_ASKPASS to set credentials Credential for logging into BitBucket
 > git fetch --tags --progress -- http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git +refs/heads/*:refs/remotes/origin/*
 > git config remote.origin.url http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git # timeout=10
Fetching upstream changes from http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git
using GIT_ASKPASS to set credentials Credential for logging into BitBucket
 > git fetch --tags --progress -- http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git +refs/heads/*:refs/remotes/origin/*
Checking out Revision 46a80cbce16d204c6593df9c3e9faad0835dc926 (origin/master)
Commit message: "main.yml edited online with Bitbucket"
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (checkout SCM)
[Pipeline] checkout
using credential BitCred
Fetching changes from the remote Git repository
Checking out Revision 46a80cbce16d204c6593df9c3e9faad0835dc926 (refs/remotes/origin/master)
 > git rev-parse origin/master^{commit} # timeout=10
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 46a80cbce16d204c6593df9c3e9faad0835dc926
Commit message: "main.yml edited online with Bitbucket"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (ansible-roles)
[Pipeline] step
[ND-63- ansible-nexus] $ ansible-playbook playbook.yml -f 5
 > git rev-parse --is-inside-work-tree # timeout=10
 > git config remote.origin.url http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git # timeout=10
Fetching upstream changes from http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git
 > git --version # timeout=10
using GIT_ASKPASS to set credentials Credential for logging into BitBucket
 > git fetch --tags --progress -- http://novartis.devops.altimetrik.io:7990/scm/son/ansible-roles.git +refs/heads/*:refs/remotes/origin/*
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 46a80cbce16d204c6593df9c3e9faad0835dc926
 [WARNING]: provided hosts list is empty, only localhost is available

PLAY [nexus-role] **************************************************************

TASK [setup] *******************************************************************
ok: [localhost]

TASK [nexus-role : nexusurl] ***************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
```



check the destination server for deployment to conform.

```
root@ip-172-31-35-91:/home/ubuntu# ls
:  ansible-roles  consumerBanking-latest.jar  Docke
root@ip-172-31-35-91:/home/ubuntu# java -jar consur
no main manifest attribute, in consumerBanking-late
root@ip-172-31-35-91:/home/ubuntu# cat playbook.yml
```

Deployment to App server(Tomcat):-

- In the ansible-playbook, you need to write the exact destination of deployment, in our case we are deploying in Apache Tomcat, deployment path is $TOMCAT_HOME/WEB_APPS. So we need to specify the destination path in dest.

- After deployment is done, please restart the tomcat server and access the service by hitting the url of application like www.localhost:8080/consumerbanking-1.0.
- Configure the tomcat server based on your requirement.
- if you the develop the JAR application, you can start below in the specified port.

```
java -jar consumerBanking-latest.jar --http:localhost:6070
```

thats all !! done.