CI Server/Jenkins Server

1) Open CentOS Machine "jenkinsserver" as shared in DATA.

2) Add following entries to /etc/hosts [[sudo vi /etc/hosts]]
    <IP Address> jenkinsserver
    <IP Address> gitserver
    <IP Address> ansibleserver

3) Install Jenkins
    ./jenkins_install.sh

4) Install Artifactory
    ./artifactory_install.sh

5) Login to Artifactory with http://IPAddress:8082 with admin/password.

6) Create Local Repository of Artifactory for Generic Packages named “hello”.

7) Generate API keys of Artifactory by going in ‘Edit Profile’.

8) Finally, after the installation of jenkins is complete you can visit the following Address in your browser http://IPAddress:8080 and install suggested plugins.

9) Install Plugins (if not already)
Manage Jenkins --> Manage Plugins —> Available
a) Git
b) Performance

10) Create “helloSetup” Job with following:

Free Style Job Name	:	helloSetup
Source Code (Git)	:	gituser@gitserver:hello.git
Poll SCM (Schedule)	:	H/5 * * * *
Build Environment	:	Delete workspace before build starts
Build	:	Execute Shell

sudo mkdir -p /var/www/hello ; sudo cp -fr * /var/www/hello/ ; sudo chmod -R 775 /var/www/hello ; sudo cp -fr /var/www/hello/hello.service /etc/systemd/system/ ; sudo systemctl daemon-reload ; sudo systemctl start hello

11) One Time Git Client Setup for Jenkins:

a) su - jenkins
b) cd /var/lib/jenkins/jobs/helloSetup
c) git config --global user.name "Sagar Mehta"
d) git config --global user.email "sagar.mehta@atgensoft.com"
e) git init
f) git remote add origin ssh://gituser@gitserver/hello.git
g) git remote set-url origin gituser@gitserver:hello.git
h) cd
i) ssh-keygen
j) ssh-copy-id -i ~/.ssh/id_rsa.pub gituser@gitserver
k) ssh-copy-id -i ~/.ssh/id_rsa.pub root@ansibleserver

12) Create “helloTest” Job with following:

Free Style Job Name	:	helloTest
Source Code (Git)	:	gituser@gitserver:hello-test.git
Build triggers	:	Build after other projects are built (helloSetup)
Build Environment	:	Delete workspace before build starts
Build	:	Invoke top-level Maven targets(Goals: clean verify)
Post-build Actions	:	
Publish Performance test result report	:	Source Data Files(**/*.jtl)
Archive the artifacts	:	
Files to archive	:	**/*jtl-report.html

13) Create “helloSave” Job with following:

Free Style Job Name	:	helloSave
Build triggers	:	Build after other projects are built (helloTest)
Build	:	Execute Shell

```
cd /var/www ; sudo tar -cvf hello.tar hello
curl -uadmin:password -T hello.tar "http://IPAddress:8081/artifactory/hello/hello.tar"
```

14) Create “helloClean” Job with following:

Free Style Job Name	:	helloClean
Build triggers	:	Build after other projects are built (helloSave)
Build	:	Execute Shell

```
sudo systemctl stop hello ; sudo rm -fr /var/www/hello ; sudo systemctl daemon-reload
```

15) Create “helloDeploy” Job with following:

Free Style Job Name	:	helloDeploy
Build triggers	:	Build after other projects are built (helloClean)
Build	:	Execute Shell

ssh root@ansibleserver ‘ansible-playbook /etc/ansible/playbooks/deploy_flask.yml’

16) deploy_flask.yml:

```
---
- hosts: localhost
  tasks:
  - name: Download TAR from artifactory
    get_url:
      url: http://172.16.130.2:8082/artifactory/hello/hello.tar
      headers:  "X-JFrog-Art-
Api:AKCp8nyNrTkK6xj6WjvTzq7oQ3Zr2EsZdYSyGWtw5kSemR15iiRicw81L5HweYpWNu7NcnL8o"
      dest: /tmp/
      mode: '0777'

  - name: Extract hello.tar into /var/www/hello
    unarchive:
      src: /tmp/hello.tar
      dest: /var/www/
      mode: '0775'

  - name: Deploy hello service
    copy:
      src: /var/www/hello/hello.service
      dest: /etc/systemd/system/hello.service
    notify:
    - restart hello service

  - name: Start hello
    service:
      name: hello
      state: started

  handlers:
  - name: restart hello service
    service:
      name: hello
      state: restarted
```