

Jenkins Integration

System Level Config. • Java • git • Ansible

Tool Level Config. • Maven

Plugin Level Config. • Docker

Jenkins with Ansible

Jenkins

- ① Install Java, git
- ② Config Java Environment Variable.
- ③ Install Jenkins

Install Ansible

- ④ Install Python 3
- ⑤ Install pip3
- ⑥ Install using pip3 → virtualenv.
- ⑦ Install ansible

Configure Ansible Inventory.

→ /etc/ansible/hosts → Global

→ [ws]
IP1
IP2
IP3
[DBS]
IP9
IP7

1. Order EC2 Instance

```
aws ec2 run-instances ^
--image-id ami-0f5ee92e2d63afc18 ^
--count 1 ^
--instance-type t2.micro ^
--key-name mujahed ^
--security-group-ids sg-04fb9b54c6f43819a ^
--subnet-id subnet-0f8384b366fe16fe9 ^
--profile mir
```

2. Install Jenkins Server

```
#!/bin/bash
```

```
# Update packages
sudo apt update
```

```
# Install Java
sudo apt install -y openjdk-11-jdk
```

```
# Install Jenkins dependencies
sudo apt install -y git
```

```
# Install Jenkins
# sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

```
# sudo echo "deb https://pkg.jenkins.io/debian-stable binary/" >> /etc/apt/sources.list.d/jenkins.list
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
sudo apt update
```

```
sudo apt install -y jenkins #Jenkins 2.410
```

```
# Start Jenkins and enable it to start on boot
sudo systemctl status jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

3. Install Apache Maven

```
$ sudo apt install -y maven
```

4. Configure Admin User

```
#!/bin/bash
url=http://localhost:8080
password=$(sudo cat /var/lib/jenkins/secrets/initialAdminPassword)

alias python=python3

# NEW ADMIN CREDENTIALS URL ENCODED USING PYTHON
username=$(python -c "import urllib.parse;print(urllib.parse.quote(input(), safe=''))" <<< "admin")
new_password=$(python -c "import urllib.parse;print(urllib.parse.quote(input(), safe=''))" <<< "admin")
fullName=$(python -c "import urllib.parse;print(urllib.parse.quote(input(), safe=''))" <<< "NubeEra")
email=$(python -c "import urllib.parse;print(urllib.parse.quote(input(), safe=''))" <<< "admin@gmail.com")

# GET THE CRUMB AND COOKIE
cookie_jar=$(mktemp)
full_crumb=$(curl -u "admin:$password" --cookie-jar "$cookie_jar" $url/crumbIssuer/api/xml?xpath=concat(//crumbRequestField,%22:%22,//crumb\))
arr_crumb=$(echo $full_crumb/://)
only_crumb=$(echo ${arr_crumb[1]})

# MAKE THE REQUEST TO CREATE AN ADMIN USER
curl -X POST -u "admin:$password" $url/setupWizard/createAdminUser \
  -H "Connection: keep-alive" \
  -H "Accept: application/json, text/javascript" \
  -H "X-Requested-With: XMLHttpRequest" \
  -H "$full_crumb" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  --cookie $cookie_jar \
  --data-raw "username=$username&password1=$new_password&password2=$new_password&fullName=$fullName&email=$email&Jenkins-Crumb=$only_crumb&json=%7B%
22username%22%3A%20%22$username%22%2C%20%22password1%22%3A%20%22$new_password%22%2C%20%22%24redact%22%3A%20%5B%22password1%22%2C%20%
22password2%22%5D%2C%20%22password2%22%3A%20%22$new_password%22%2C%20%22fullName%22%3A%20%22$fullName%22%2C%20%22email%22%3A%20%22$email%22%
2C%20%22Jenkins-Crumb%22%3A%20%22$only_crumb%22%7D&core%3Aapply=&Submit=Save&json=%7B%22username%22%3A%20%22$username%22%2C%20%22password1%22%
3A%20%22$new_password%22%2C%20%22%24redact%22%3A%20%5B%22password1%22%2C%20%22password2%22%5D%2C%20%22password2%22%3A%20%22$new_password%
22%2C%20%22fullName%22%3A%20%22$fullName%22%2C%20%22email%22%3A%20%22$email%22%2C%20%22Jenkins-Crumb%22%3A%20%22$only_crumb%22%7D"
```

5. Configure URL

```
#!/bin/bash
url=http://localhost:8080

user=admin
password=admin

cookie_jar=$(mktemp)
full_crumb=$(curl -u "$user:$password" --cookie-jar "$cookie_jar" $url/crumbIssuer/api/xml?xpath=concat(//crumbRequestField,%22:%22,//crumb\))
arr_crumb=$(echo $full_crumb/://)
only_crumb=$(echo ${arr_crumb[1]})

# MAKE THE REQUEST TO DOWNLOAD AND INSTALL REQUIRED MODULES
curl -X POST -u "$user:$password" $url/pluginManager/installPlugins \
  -H 'Connection: keep-alive' \
  -H 'Accept: application/json, text/javascript, */*; q=0.01' \
  -H 'X-Requested-With: XMLHttpRequest' \
  -H "$full_crumb" \
  -H 'Content-Type: application/json' \
  -H 'Accept-Language: en,en-US;q=0.9,it;q=0.8' \
  --cookie $cookie_jar \
  --data-raw '{"dynamicLoad":true,"plugins":["cloudbees-folder","antisamy-markup-formatter","build-timeout","credentials-binding","timestamp","ws-cleanup","ant","gradle","workflow-
aggregator","github-branch-source","pipeline-github-lib","pipeline-stage-view","git","ssh-slaves","matrix-auth","pam-auth","ldap","email-ext","mailer"],"Jenkins-Crumb":{"$only_crumb"}}
```

6. Install Recommended Plugins

```
#!/bin/bash
url=http://localhost:8080

user=admin
password=admin

cookie_jar=$(mktemp)
full_crumb=$(curl -u "$user:$password" --cookie-jar "$cookie_jar" $url/crumbIssuer/api/xml?xpath=concat(//crumbRequestField,%22:%22,//crumb\))
```

```

arr_crumb=${full_crumb//:/ }
only_crumb=$(echo ${arr_crumb[1]})

# MAKE THE REQUEST TO DOWNLOAD AND INSTALL REQUIRED MODULES
curl -X POST -u "$user:$password" $url/pluginManager/installPlugins \
-H 'Connection: keep-alive' \
-H 'Accept: application/json, text/javascript, */*; q=0.01' \
-H 'X-Requested-With: XMLHttpRequest' \
-H "$full_crumb" \
-H 'Content-Type: application/json' \
-H 'Accept-Language: en,en-US;q=0.9,it;q=0.8' \
--cookie $cookie_jar \
--data-raw '{"dynamicLoad":true,"plugins":["cloudbees-folder","antisamy-markup-formatter","build-timeout","credentials-binding","timestamp","ws-cleanup","ant","gradle","workflow-aggregator","github-branch-source","pipeline-github-lib","pipeline-stage-view","git","ssh-slaves","matrix-auth","pam-auth","ldap","email-ext","mailer"],"Jenkins-Crumb":"$only_crumb"}'

```

7. Install & Configure Ansible

```

#!/bin/bash
sudo apt update
sudo mkdir /etc/ansible
sudo bash -c 'cat<<EOF > /etc/ansible/hosts
[ws]
127.0.0.1
EOF'
sudo apt install ansible -y
ssh-keygen -b 3072 -t rsa -f ~/.ssh/sshkey -q -N ""
cat ~/.ssh/sshkey.pub | cat >> ~/.ssh/authorized_keys
sudo ansible --version > /tmp/ansibleVersion.txt

```

Jenkins Ansible Version Check Pipeline

```

pipeline {
    agent any

    stages {
        stage('Deploy') {
            steps {
                sh 'ansible --version'
            }
        }
    }
}

```

Jenkins Ansible Ad-hoc Command pipeline

```

pipeline {
    agent any

    stages {
        stage('Deploy') {
            steps {
                sh 'ansible all -m ping'
            }
        }
    }
}

```

Git Repo Clone in Jenkins Pipeline

```

pipeline {
    agent any

    stages {

        stage('clone') {
            steps {
                git branch: 'main', url: 'https://github.com/NubeEra-Practice/ansible-playbooks.git'
            }
        }
    }
}

```

```

$ cat > pb1.yaml
---
- hosts: all

```

```
tasks:
- name: Print Welcome World
  debug: msg="Welcome World"
```

ansible-playbook pb1.yaml

Ansible Jenkins Pipeline with Playbook

```
pipeline {
  agent any

  stages {
    stage('clone') {
      steps {
        git branch: 'main', url: 'https://github.com/NubeEra-Practice/ansible-playbooks.git'
      }
    }
    stage('deploy') {
      steps {
        ansible-playbook pb1.yaml
      }
    }
  }
}
```