**1. Getting started with jenkins**

**2.** **Integrate with Manmon Agent**

**3. Integrate with Manmon Front**

**4. Integrate with Selenium**

**5. Integrate with Ansible**

**1. Getting started with jenkins**

**jenkins Install:**

[i.](https://phoenixnap.com/kb/install-jenkins-ubuntu) wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

ii. sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \

/etc/apt/sources.list.d/jenkins.list'

iii. sudo apt-get update

iv. sudo apt-get install jenkins

**Getting started with jenkins:**

i. cd /usr/share/jenkins

ii. java -jar jenkins.war

iii. Copy installation password

iii. Browse <ip\_address>:8080 (default port is 8080)

iv. paste installation password.

v. configure.

**Reference:**

jenkins configure: <https://www.tutorialspoint.com/jenkins/jenkins_configuration.htm>

**2. Manmon Agent:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Tool | Tool Purpose | Responsible | Time |
| Lxc container create | lxc | To create lxc container | Raihan |  |
| SSH | To communicate with |
| JVM Environment | JDK 11 |  | Joy |  |
| Maven 3.6.3 |  |
| Unit testing | Junit | For unit testing | Joy |  |
| Build Script |  |  | Joy |  |
| Jenkins Prepare | Maven plugin | To integrate with maven | Raihan |  |
| Git plugin | To communicate with git. |
| Junit plugin |  |
| Github plugin | To communicate with github. |
| Github pull |  |  | Raihan |  |
| Build and deploy | Jenkins shell |  | Raihan |  |

**Manmon Agent Jenkins prepare Steps:**

1. create a “Free Style Project”

2. “General” section,

→select “This project is parameterized”,

→select “String Parameter”,

→give branch name in default value.

2. “Source Code Management” section,

→select Git, give repository, credentials, branch.

→Additional Behaviors, select Advanced Clone Behaviors, set Timeout.

3. “Build Environment” section,

→select “Abort the build if it’s stuck” and set timeout.

4. “Build” section,

→select “Execute Shell”,

→commands:

cd manmon

./mvnw clean install

cd target

java -jar manmon-0.0.1-SNAPSHOT.jar

5. Apply and Save.

6. Build with Parameters.

**3. Manmon Front:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Tool | Tool Purpose | Responsible | Time |
| Node | Node 12.16.1 |  | Raihan/ Jyoti |  |
| npm |  |
| Testing Script | Karma | Unit testing | Jyoti |  |
| Jasmine | Unit testing |  |  |
| Karma-firefox plugin |  |  |  |
| Jenkins Prepare | Node plugin | Node | Raihan |  |
| Git plugin | To communicate with git. |
| Junit plugin |  |
| Github plugin | To communicate with github. |
| Github pull |  |  | Raihan |  |
| Build | Jenkins shell |  | Raihan |  |

**Manmon Front Jenkins prepare Steps:**

1. create a “Free Style Project”

2. “General” section,

→select “This project is parameterized”,

→select “String Parameter”,

→give branch name in default value.

2. “Source Code Management” section,

→select Git, give repository, credentials, branch.

→Additional Behaviors, select Advanced Clone Behaviors, set Timeout.

3. “Build Environment” section,

→select “Abort the build if it’s stuck” and set timeout.

4. “Build” section,

→select “Execute Shell”,

→commands:

cd manmon

./mvnw clean install

cd src/main/resources/static

karma start

5. Apply and Save.

6. Build with Parameters.

**4. Selenium:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Tool | Tool Purpose | Responsible | Time |
| Lxc container create | lxc | To create lxc container | Raihan |  |
| SSH | To communicate with |
| JVM Environment | JDK 11 |  | Farhad |  |
| Test Script | Selenium | Testing tool | Farhad |  |
| Jenkins Prepare | Maven plugin | To integrate with maven | Raihan |  |
| Git plugin | To communicate with git. |
| Junit plugin |  |
| Github plugin | To communicate with github. |
| Chromedriver plugin | To communicate with chrome. |  |
| Selenium plugin | To run selenium test cases. |  |
| Github pull |  |  | Raihan |  |
| Build and deploy | Jenkins shell |  | Raihan |  |

**Selenium Jenkins prepare Steps:**

1. create a “Free Style Project”

2. “General” section,

→select “This project is parameterized”,

→select “String Parameter”,

→give branch name in default value.

2. “Source Code Management” section,

→select Git, give repository, credentials, branch.

→Additional Behaviors, select Advanced Clone Behaviors, set Timeout.

3. “Build Environment” section,

→select “Abort the build if it’s stuck” and set timeout.

4. “Build” section,

→select “Execute Shell”,

→commands:

cd Manmon-Test

mvn clean install

cd target

java -jar manmontest-1.0-SNAPSHOT-jar-with-dependencies.jar

5. Apply and Save.

6. Build with Parameters.

**5. Ansible:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Tool | Tool Purpose | Responsible | Time |
| Lxc container create | lxc | To create lxc container | Raihan |  |
| SSH | To communicate with |
| Ansible | JDK 14 |  | Zahid |  |
| Deploy Script |  |  | Zahid |  |
| Jenkins Prepare | Ansible plugin | To integrate with ansible playbook | Raihan |  |
| Git plugin | To communicate with git. |
| Github plugin | To communicate with github. |
| Github pull |  |  | Raihan |  |
| Build and deploy | Jenkins shell |  | Raihan |  |

**Ansible Jenkins prepare Steps:**

1. create a “Free Style Project”

2. “General” section,

→select “This project is parameterized”,

→select “String Parameter”,

→give branch name in default value.

2. “Source Code Management” section,

→select Git, give repository, credentials, branch.

→Additional Behaviors, select Advanced Clone Behaviors, set Timeout.

3. “Build Environment” section,

→select “Abort the build if it’s stuck” and set timeout.

4. “Build” section,

→select “Invoke Ansible Playbook”,

→playbook path “main.yml”

→ file path “hosts”

5. Apply and Save.

6. Build with Parameters.S