Experiment No. 6: To implement Jenkins Master-Slave Architecture with Scaling.

STEP1: In Jenkins Dashboard Click on Manage Jenkins -> Manage Nodes

STEP 2: Select New Node and enter the name of the node in the Node Name field. Select Permanent Agent and click the OK button. Initially, you will get only one option, “Permanent Agent.” Once we have one or more slaves you will get the “Copy Existing Node” option. Click Create

Create JENKINS_LAB directory in Desktop and copy its path. Paste the path in “Remote Root Directory” field.

Add label as <agentname>

Change usage field to “Only build jobs with label expressions matching this node”

STEP3: Provide jdk path under Node properties > Environment Variables. Give name is java_home.

Jdk path can be obtained by running find / -type f -name java.

STEP4: On click of ‘Save’ will display the below page with error message. Here Jenkins connect with Slave node using Java Web Start and it needs a port to establish the connection. To configure JNLP port in global security. Now go to Manage Jenkins > Security.

Set Agent port to fixed and give value 50000, then run sudo ufw allow 50000/tcp in terminal.

STEP5: Again coming back to Jenkins and navigate to Nodes -> agent2 which will display two ways to connect with Agent node. Run the commands for linux to establish connection.

Now Jenkins Slave node is ready to run any job. This node’s label name should be mentioned in the corresponding Job configuration as below:

STEP 6: Create a New Job in Jenkins dashboard

STEP7: Configure the job by checking “Restrict where this project can be run” and add label expression as name of agent that has been created. In build steps, select “Execute Shell” and write a small echo message. Build the project and check Console Output. Finally check Node Dashboard which will display projects tied to agent.