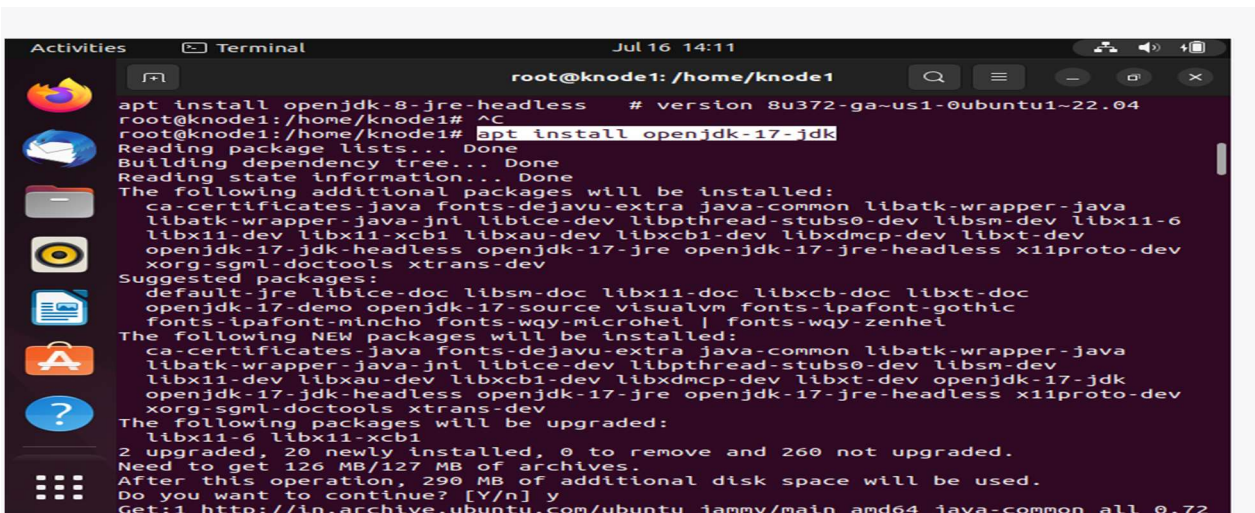


Assignment 3

Create Linux slave nodes in the Jenkins master

Step 1: Prepare the Linux Slave Node

1. Ensure that Java is installed on the Linux slave node, as Jenkins requires Java to run. If Java is not installed, you can install it using the package manager for your Linux distribution.
2.
 - For Ubuntu: `sudo apt install openjdk-11-jdk` or `sudo apt install openjdk-17-jdk`
 - For CentOS/RHEL: `sudo yum install java-11-openjdk-devel`



```
root@knode1: /home/knode1
apt install openjdk-8-jre-headless # version 8u372-ga-us1-0ubuntu1~22.04
root@knode1: /home/knode1# ^C
root@knode1: /home/knode1# apt install openjdk-17-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-6
  libx11-dev libx11-xcb1 libxau-dev libxcb1-dev libxdmcp-dev libxt-dev
  openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless x11proto-dev
  xorg-sgml-doctools xtrans-dev
Suggested packages:
  default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc
  openjdk-17-demo openjdk-17-source visualvm fonts-ipafont-gothic
  fonts-ipafont-mincho fonts-wqy-microhei | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
  libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev
  libx11-dev libx11-xcb1 libxau-dev libxcb1-dev libxdmcp-dev openjdk-17-jdk
  openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless x11proto-dev
  xorg-sgml-doctools xtrans-dev
The following packages will be upgraded:
  libx11-6 libx11-xcb1
2 upgraded, 20 newly installed, 0 to remove and 260 not upgraded.
Need to get 126 MB/127 MB of archives.
After this operation, 290 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 java-common all 0.72
```

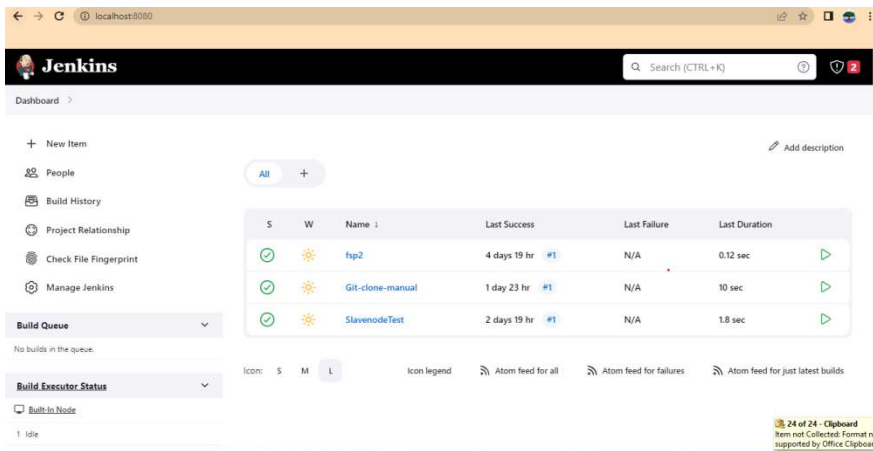
3. Set up SSH key authentication on the Jenkins master and copy the public key to the Linux slave node. This will allow the Jenkins master to communicate with the slave node securely over SSH.

Step 2: Configure Jenkins Master

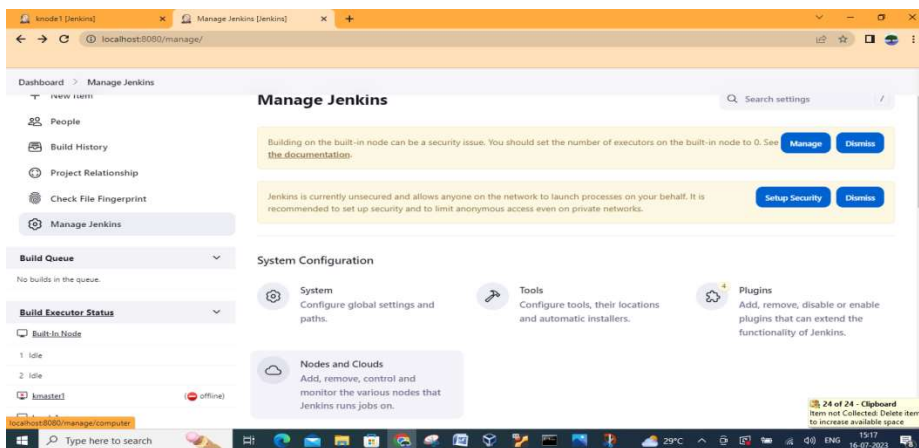
1. Open your web browser and navigate to the Jenkins web interface by entering <http://localhost:8080>.
2. If you haven't done so already, log in to Jenkins using your admin credentials.



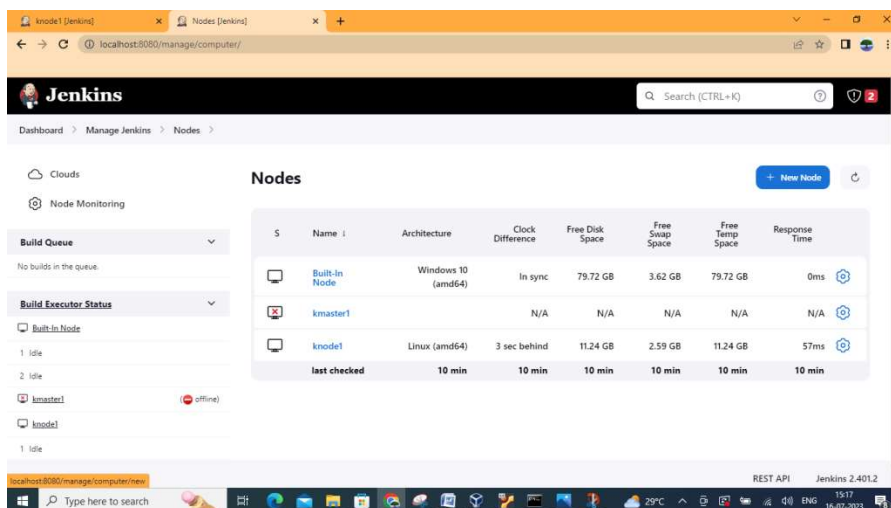
Once logged in, click on "Manage Jenkins" in the left-hand side menu.



3. Click on "Manage Nodes and Clouds."

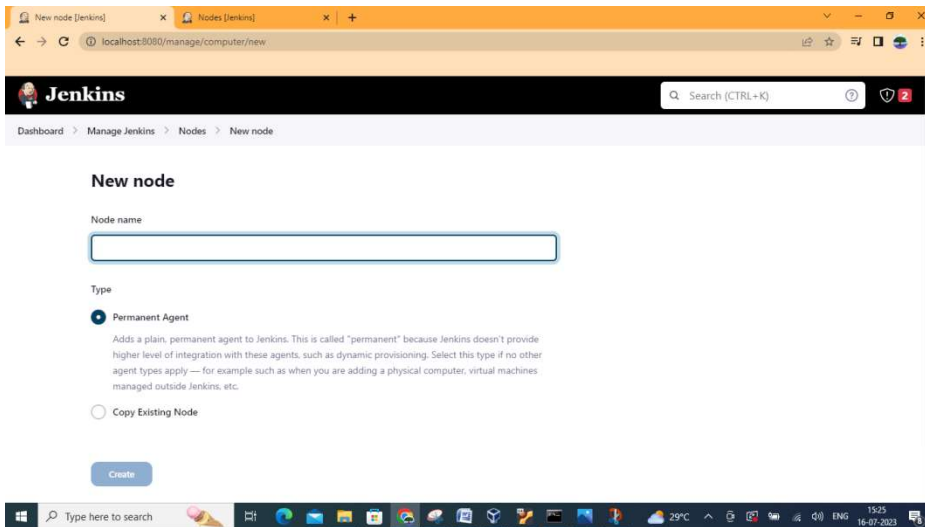


4. Click on "New Node" on the left-hand side menu to add a new slave node.



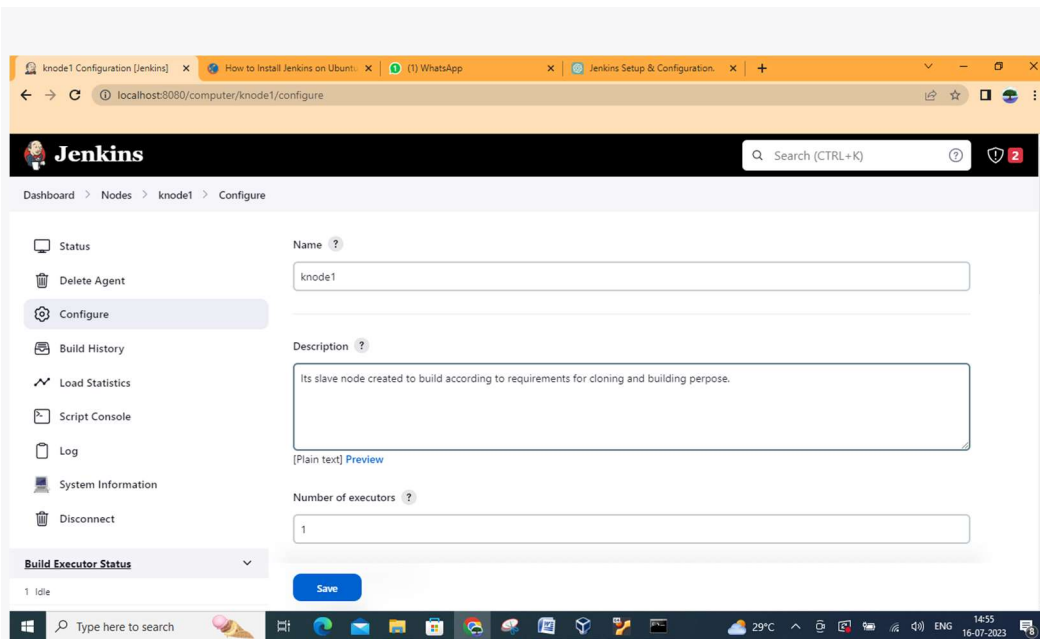
5. Provide a node name, e.g., "knode1" or any descriptive name for your slave node.

6. Select "Permanent Agent" and click "OK."

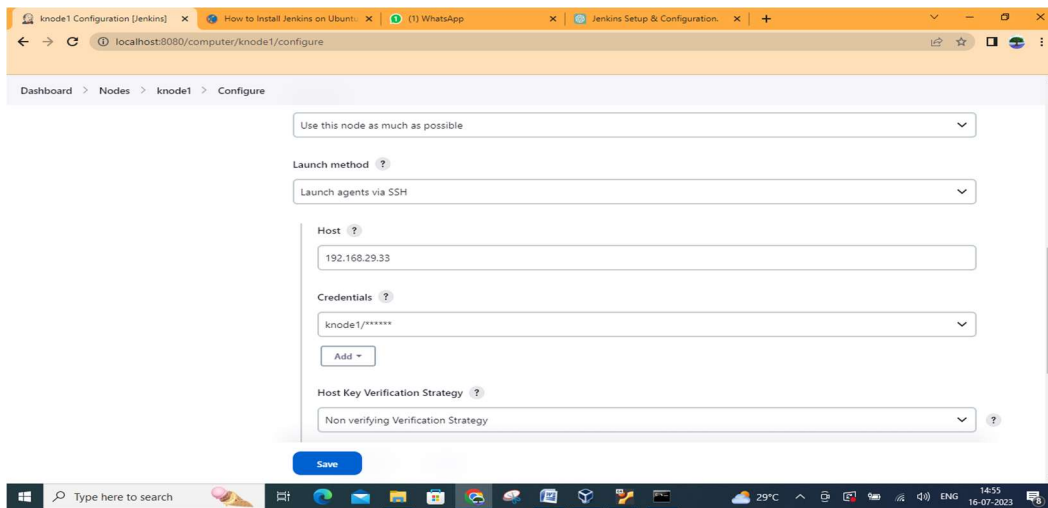


7. Now, configure the node settings as follows:

- **# Of executors:** The number of concurrent builds that this slave node can handle. Set this based on the resources available on the slave node.



- **Remote root directory:** The path to the home directory of the user on the slave node where Jenkins will perform the builds.
- **Labels:** You can add any labels that identify the capabilities of the slave node (e.g., "linux," "knode1," etc.).



8. For the **Launch method**, select "Launch agents via SSH."
9. In the **Host** field, enter the IP address or domain name of the Linux slave node.
10. For **Credentials**, click on "Add" to add the SSH credentials for the user on the slave node.
 - Select "Kind: SSH Username with private key."
 - Enter the **Username [knode1]** for the user on the slave node.
 - In the **Private Key** section, select "Enter directly" and paste the public key of the Jenkins master.
12. Click "Save" to add the Linux slave node to Jenkins.

Step 3: Verify the Slave Node Connection

1. After saving the configuration, Jenkins will attempt to connect to the Linux slave node using the provided SSH credentials.
2. To verify the connection, go back to the "Manage Nodes and Clouds" page and check if the slave node shows "Online" status.

Step 4: Run Builds on the Linux Slave Node

Now that the Linux slave node is connected to the Jenkins master, you can configure your Jenkins jobs to use this node. In the job configuration, under the "Restrict where this project can be run" section, specify the labels you added earlier to distribute builds among the slave nodes.

With this configuration, Jenkins will automatically distribute builds between the master and the Linux slave node based on the labels specified in the job configuration.