

# DEVOPS

## ASSIGNMENT 1

### Installation of Nginx.

#### Task Description:

This task involves creating a Jenkins Freestyle job that automates the installation of Nginx on an Ubuntu server using the command `sudo apt-get install -y nginx`. This automation ensures consistent server setup and can be easily repeated, making it ideal for automated environments.

---

#### Steps:

##### 1. Update Package List:

- Run the following command to update the package list:
- `sudo apt update`

##### 2. Install Nginx:

- Install Nginx using the following command:
- `sudo apt-get install -y nginx`

##### 3. Verify Installation:

- Check if Nginx is running with the command:
- `systemctl status nginx`

##### 4. Create Jenkins Freestyle Job:

- In Jenkins, click New Item and create a Freestyle project (e.g., Install\_Nginx).

##### 5. Configure Job:

- Add a build step of type Execute shell.
- In the shell command box, enter:
- `sudo apt update`
- `sudo apt-get install -y nginx`

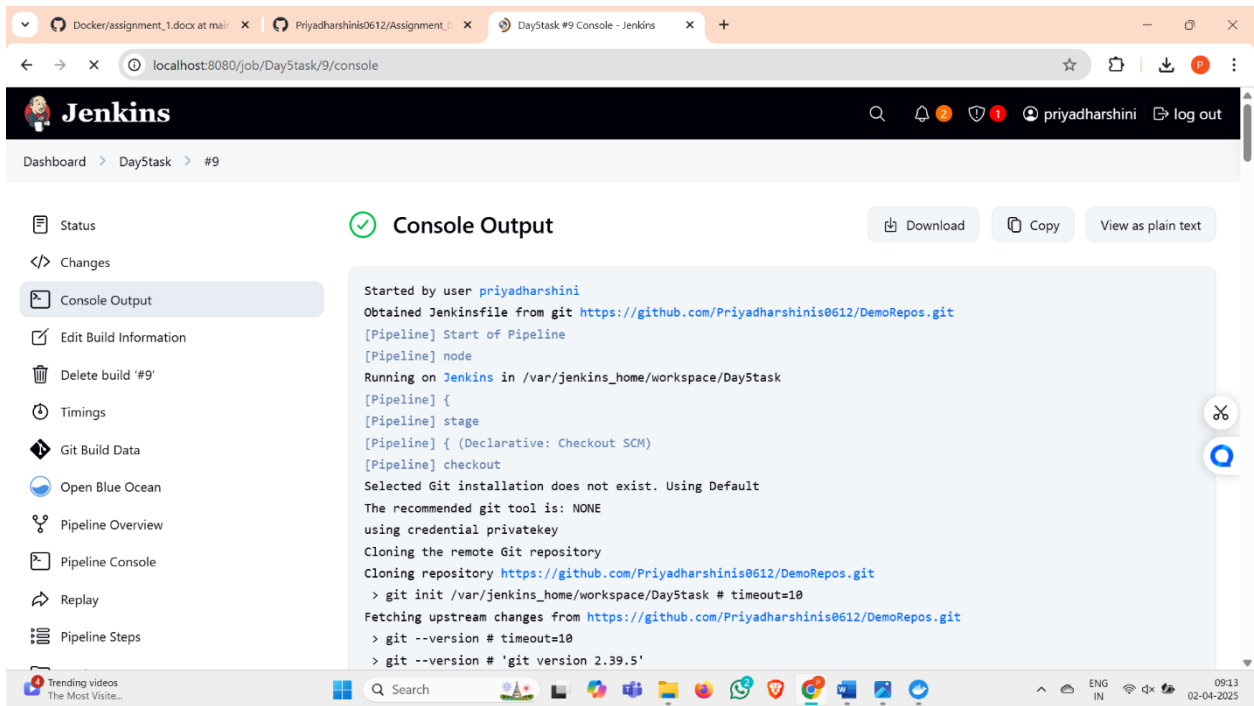
##### 6. Save and Run Job:

- Save the job and click Build Now to run it.

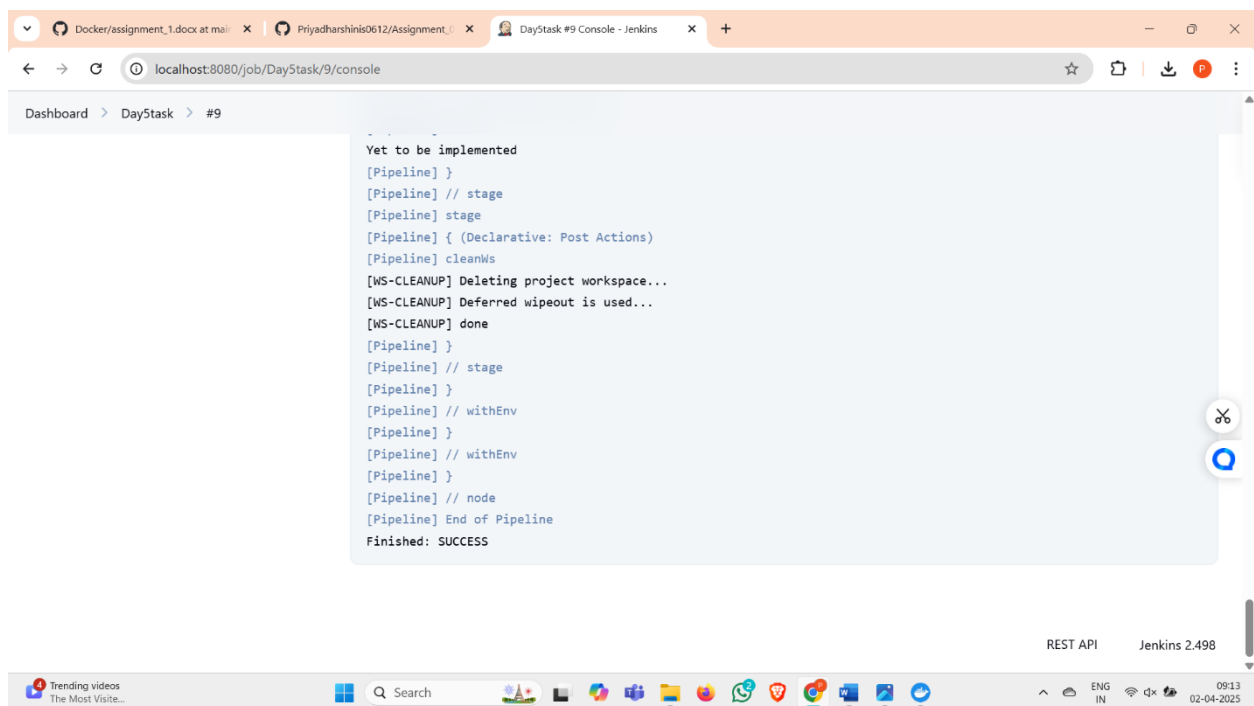
## 7. Check Console Output:

- After the job completes, view the **Console Output** to confirm the successful installation of Nginx.

## OUTPUT:



```
Started by user priyadharshini
Obtained Jenkinsfile from git https://github.com/Priyadharshinis0612/DemoRepos.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/DayStask
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential privatekey
Cloning the remote Git repository
Cloning repository https://github.com/Priyadharshinis0612/DemoRepos.git
> git init /var/jenkins_home/workspace/DayStask # timeout=10
Fetching upstream changes from https://github.com/Priyadharshinis0612/DemoRepos.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
```



```
Yet to be implemented
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] cleanWs
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```