# EXPERIMENT-1

**AIM :** Git installation and create a repository and perform fetch, pull, branching operations.

**DESCRIPTION :**

• Understanding Version Control:

Version control systems like Git are indispensable tools in modern software development. They allow teams to collaborate effectively, track changes, revert to previous versions if necessary, and maintain a structured development workflow.

• Importance of Git Installation:

Git installation is the foundational step in establishing a versioncontrolled environment. It provides a local instance of Git on the developer's machine, enabling them to manage code versions efficiently.

• Repository Creation:

Creating a Git repository serves as the cornerstone of version control. It centralizes project files, facilitating collaborative development and ensuring a centralized location for code history and changes.

• Fetch Operation:

The fetch operation in Git is crucial for retrieving changes from a remote repository without automatically merging them into the local repository. It allows developers to review changes before integrating them into their working copy, ensuring code stability and quality.

• Pull Operation:

Pulling changes from a remote repository is vital for synchronizing the local repository with the latest updates from the central codebase. It enables seamless collaboration among team members by ensuring that everyone is working with the most recent codebase.

• Branching Operations:

Branching is a fundamental concept in Git that facilitates parallel development efforts within a project. Creating branches enables developers to work on new features, bug fixes, or experiments without affecting the main codebase. Branches promote code isolation, experimentation, and seamless integration through merging.

**PROCECURE :**

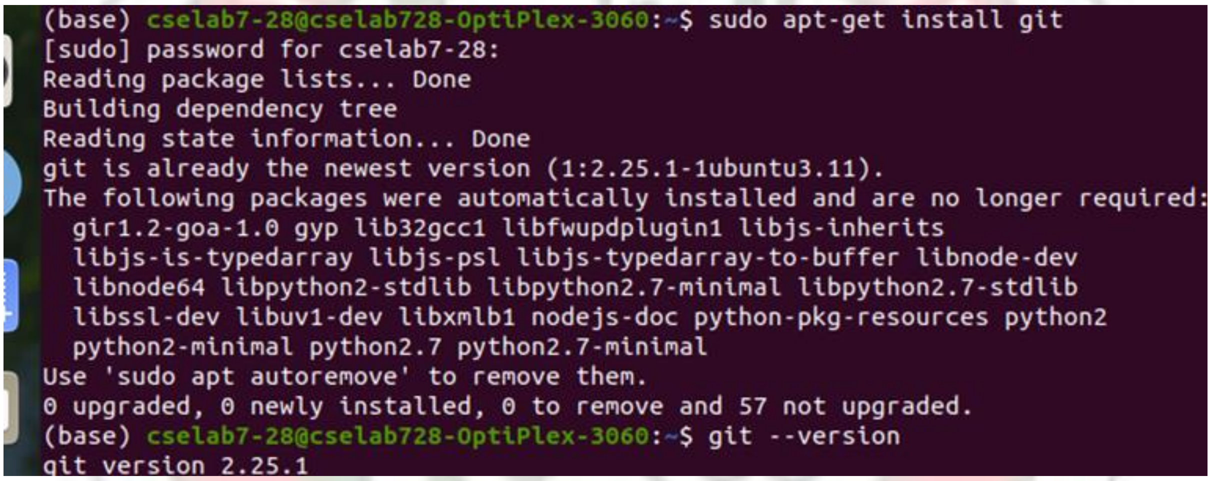
Step 1: Installing Git

Use your package manager to install Git. For example, on Ubuntu, you can run

sudo apt-get install git.

Verify Installation:

- Open a terminal (or command prompt on Windows).
- Type git --version and press Enter. You should see the installed Git version.



```
(base) cselab7-28@cselab728-OptiPlex-3060:~$ sudo apt-get install git
[sudo] password for cselab7-28:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3.11).
The following packages were automatically installed and are no longer required:
  gir1.2-goa-1.0 gyp lib32gcc1 libfwupdplugin1 libjs-inherits
  libjs-is-typedarray libjs-psl libjs-typedarray-to-buffer libnode-dev
  libnode64 libpython2-stdlib libpython2.7-minimal libpython2.7-stdlib
  libssl-dev libuv1-dev libxmlb1 nodejs-doc python-pkg-resources python2
  python2-minimal python2.7 python2.7-minimal
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
(base) cselab7-28@cselab728-OptiPlex-3060:~$ git --version
git version 2.25.1
```

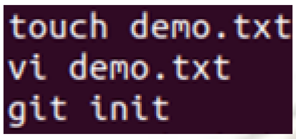
Step 2: Creating a Repository

- Create a New Directory:
- Open a terminal or command prompt.
- Navigate to the directory where you want to create your Git repository. You can use

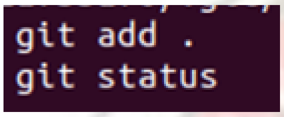
cd to change directories.

- Initialize Git Repository:
- Run the command git init. This initializes a new Git repository in the current directory.
- Add Files:
- Place the files you want to track in the repository into this directory.
- Use the command git add <file> to add files to the staging area.

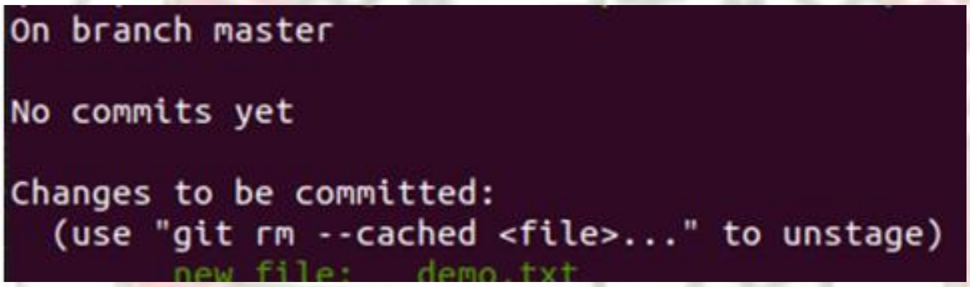
- Commit Changes:
- After adding files, commit them to the repository with git commit -a -m "Your commit message".



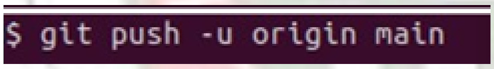
```
touch demo.txt
vi demo.txt
git init
```



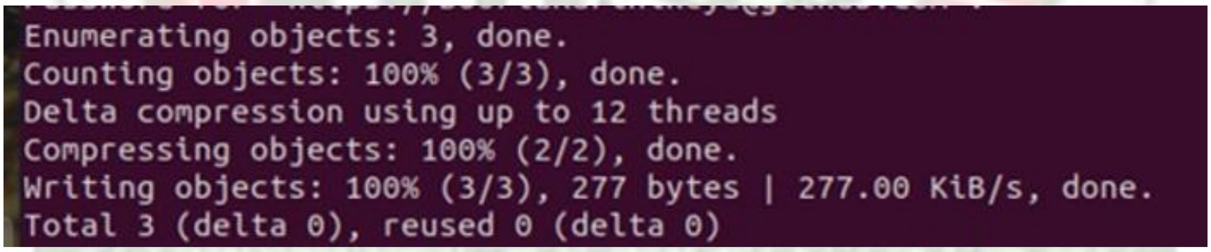
```
git add .
git status
```



```
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   demo.txt
```



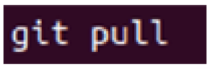
```
$ git push -u origin main
```



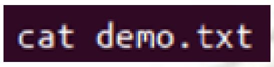
```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
```

Step 3: Basic Git Operations Pull:

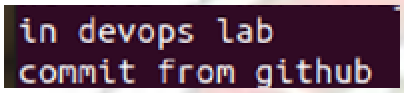
• To fetch changes from a remote repository and merge them into your current branch, use git pull



```
git pull
```



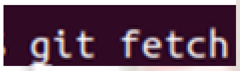
```
cat demo.txt
```



```
in devops lab
commit from github
```

Fetch:

• To fetch changes from a remote repository, use git fetch



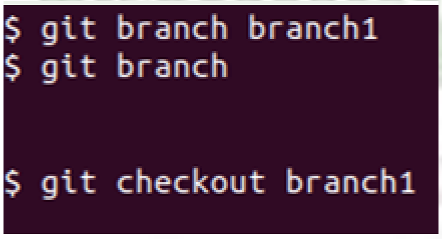
```
git fetch
```

Branching:

• To create a new branch, use git branch <branch_name>.

• To switch to a different branch, use git checkout <branch_name>.

• To create and switch to a new branch simultaneously, use git checkout -b <branch_name>.



```
$ git branch branch1
$ git branch


$ git checkout branch1
```

## EXPERIMENT-2

**AIM :** Jenkins Installation and implement continues Integration and Continues deployment, build a job using Jenkins.

**DESCRIPTION :**

• Jenkins is an open-source automation server used for continuous integration and continuous delivery (CI/CD) pipelines.

• It allows developers to automate various stages of the software development process, including building, testing, and deploying applications.

• Jenkins supports integration with various version control systems such as Git, SVN, and Mercurial.

• It offers a web-based interface for easy configuration and management of jobs and pipelines.

• Jenkins provides a wide range of plugins to extend its functionality, allowing integration with different tools and technologies.

• With Jenkins, developers can schedule and trigger builds based on code commits, time intervals, or external events.

• It supports distributed builds, allowing users to distribute work across multiple machines or nodes.

• Jenkins offers robust security features, including role-based access control and support for various authentication mechanisms.

• It provides extensive logging and monitoring capabilities to track build statuses, logs, and performance metrics.

• Jenkins is highly customizable and scalable, making it suitable for projects of any size and complexity.

**PROCEDURE :**

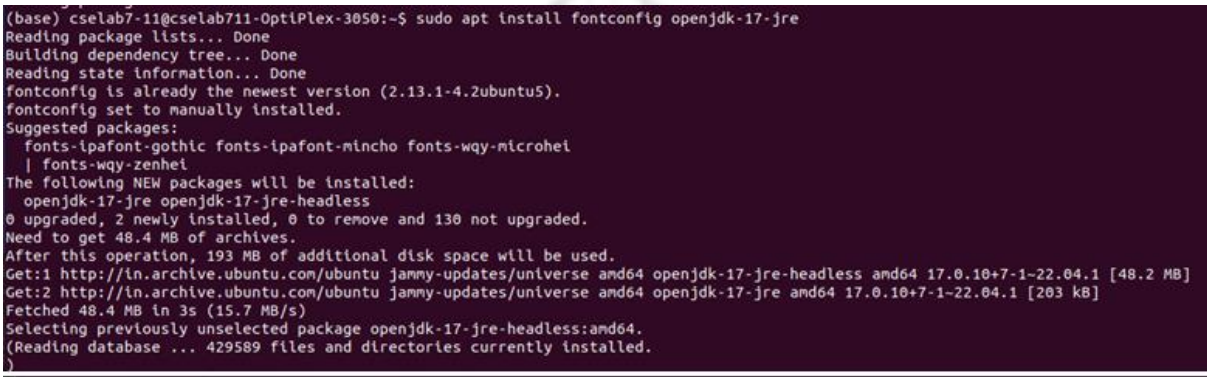
Step 1: Update Package Repository

Before installing Jenkins, it's a good practice to update the package repository on your Linux system:

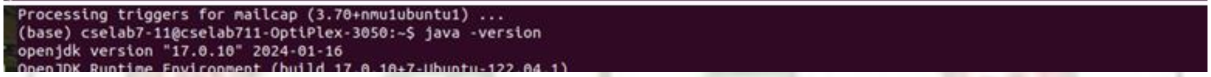
Sudo apt update Step 2: Install Java

Jenkins requires Java to run. You can install OpenJDK, which is an open-source implementation of the Java Platform:

sudo apt install fontconfig openjdk-17-jre java -version



```
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo apt install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fontconfig is already the newest version (2.13.1-4.2ubuntu5).
fontconfig set to manually installed.
Suggested packages:
  fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
  openjdk-17-jre openjdk-17-jre-headless
0 upgraded, 2 newly installed, 0 to remove and 130 not upgraded.
Need to get 48.4 MB of archives.
After this operation, 193 MB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre-headless amd64 17.0.10+7-1~22.04.1 [48.2 MB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre amd64 17.0.10+7-1~22.04.1 [203 kB]
Fetched 48.4 MB in 3s (15.7 MB/s)
Selecting previously unselected package openjdk-17-jre-headless:amd64.
(Reading database ... 429589 files and directories currently installed.
)
```



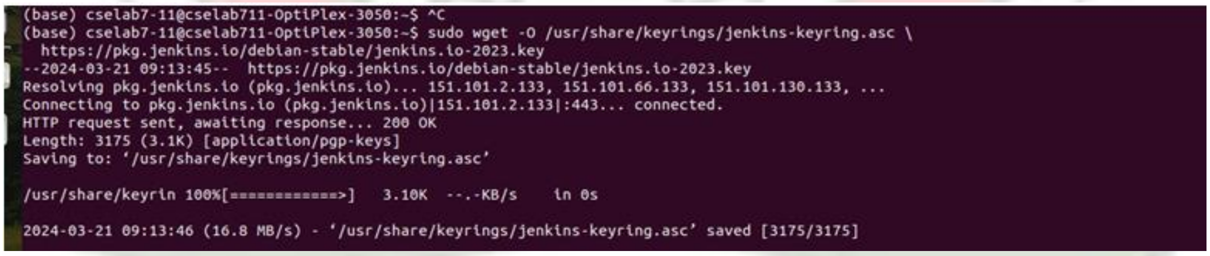
```
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
(base) cselab7-11@cselab711-OptiPlex-3050:~$ java -version
openjdk version "17.0.10" 2024-01-16
OpenJDK Runtime Environment (build 17.0.10+7-Ubuntu-122.04.1)
```

Step 3: Add Jenkins Repository Key

To ensure the authenticity of the Jenkins packages, add the Jenkins repository key to your system:

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

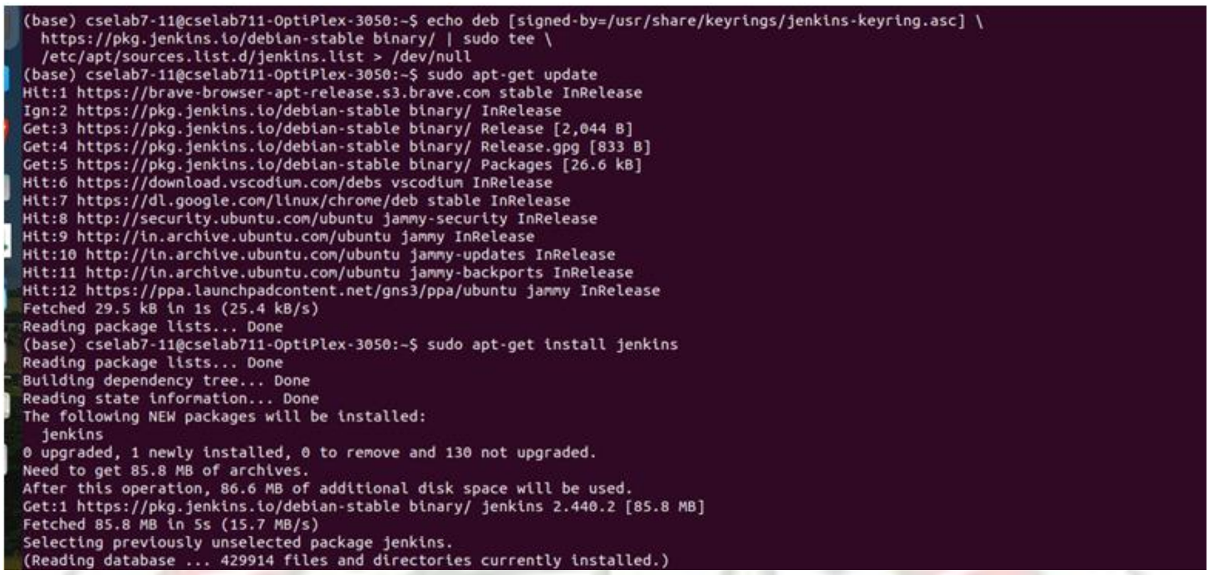


```
(base) cselab7-11@cselab711-OptiPlex-3050:~$ ^C
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-03-21 09:13:45--  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.2.133, 151.101.66.133, 151.101.130.133, ...
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.2.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrin 100%[============>]   3.10K  --.-KB/s    in 0s

2024-03-21 09:13:46 (16.8 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
```

Step 4: Add Jenkins Repository

Add the Jenkins repository to your system's list of package sources:

'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'



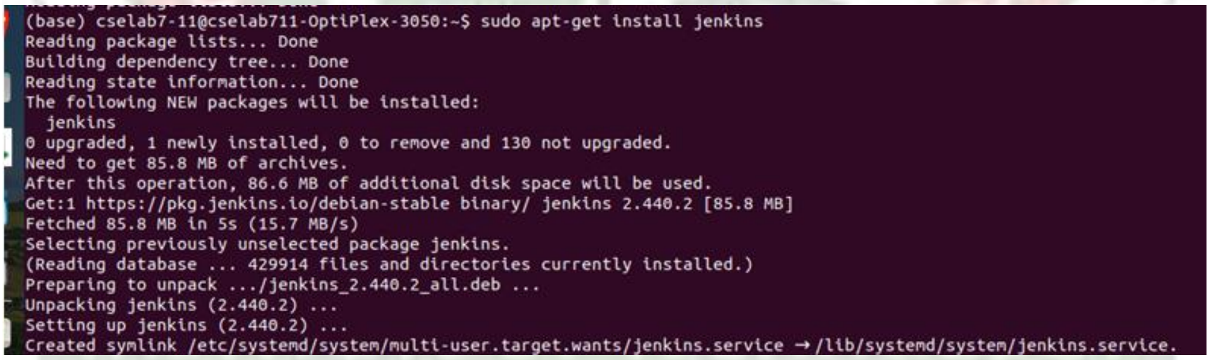
```
(base) cselab7-11@cselab711-OptiPlex-3050:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo apt-get update
Hit:1 https://brave-browser-apt-release.s3.brave.com stable InRelease
Ign:2 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:3 https://pkg.jenkins.io/debian-stable binary/ Release [2,044 B]
Get:4 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:5 https://pkg.jenkins.io/debian-stable binary/ Packages [26.6 kB]
Hit:6 https://download.vscodium.com/debs vscodium InRelease
Hit:7 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:8 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:9 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:10 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:11 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:12 https://ppa.launchpadcontent.net/gns3/ppa/ubuntu jammy InRelease
Fetched 29.5 kB in 1s (25.4 kB/s)
Reading package lists... Done
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  jenkins
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 85.8 MB of archives.
After this operation, 86.6 MB of additional disk space will be used.
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.440.2 [85.8 MB]
Fetched 85.8 MB in 5s (15.7 MB/s)
Selecting previously unselected package jenkins.
(Reading database ... 429914 files and directories currently installed.)
```

Step 5: Install Jenkins

Now, update the package repository again and install Jenkins:

sudo apt-get update

sudo apt-get install Jenkins



```
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  jenkins
0 upgraded, 1 newly installed, 0 to remove and 130 not upgraded.
Need to get 85.8 MB of archives.
After this operation, 86.6 MB of additional disk space will be used.
Get:1 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.440.2 [85.8 MB]
Fetched 85.8 MB in 5s (15.7 MB/s)
Selecting previously unselected package jenkins.
(Reading database ... 429914 files and directories currently installed.)
Preparing to unpack .../jenkins_2.440.2_all.deb ...
Unpacking jenkins (2.440.2) ...
Setting up jenkins (2.440.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
```

Step 6: Start Jenkins

You can enable the Jenkins service to start at boot with the command:

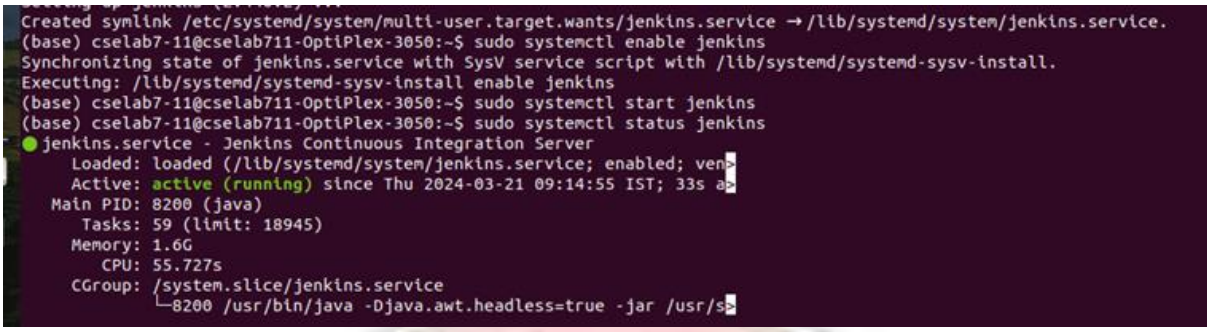
```
sudo systemctl enable jenkins
```

You can start the Jenkins service with the command:

```
sudo systemctl start jenkins
```

You can check the status of the Jenkins service using the command:

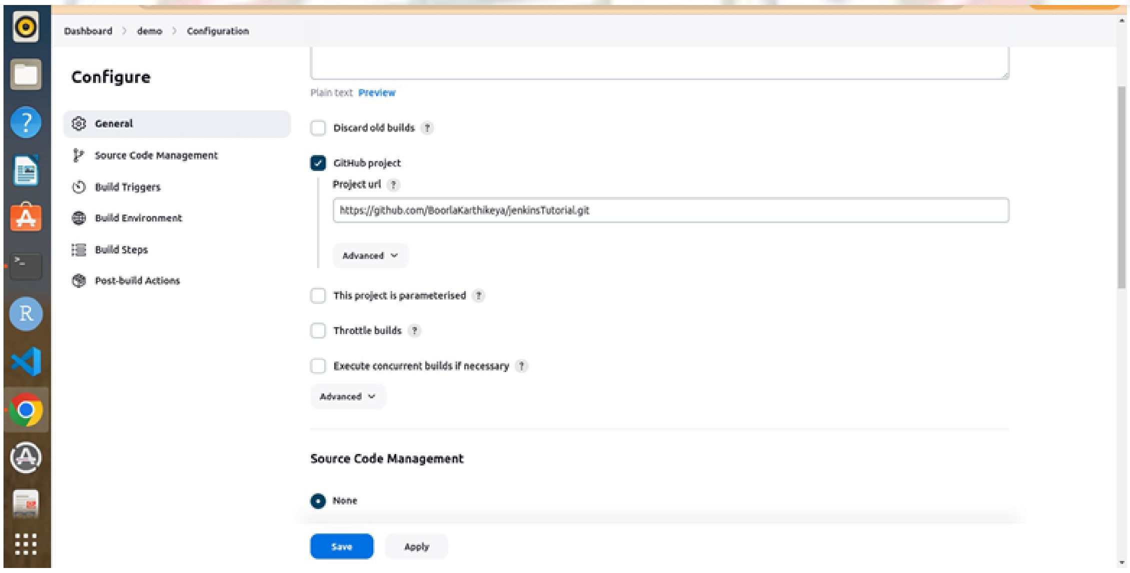
```
sudo systemctl status Jenkins
```



```
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo systemctl start jenkins
(base) cselab7-11@cselab711-OptiPlex-3050:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; ven>
     Active: active (running) since Thu 2024-03-21 09:14:55 IST; 33s a>
   Main PID: 8200 (java)
      Tasks: 59 (limit: 18945)
     Memory: 1.6G
        CPU: 55.727s
     CGroup: /system.slice/jenkins.service
             └─8200 /usr/bin/java -Djava.awt.headless=true -jar /usr/s>
```

**Step7 : configure jenkins**

Dashboard -> new Item -> github url -> save -> build -> console output



Dashboard > demo > Configuration

Configure

General / Source Code Management / Build Triggers / Build Environment / Build Steps / Post-build Actions

Plain text Preview

☐ Discard old builds

☑ GitHub project

Project url

https://github.com/BoorlaKarthikeya/jenkinsTutorial.git

Advanced

☐ This project is parameterised

☐ Throttle builds

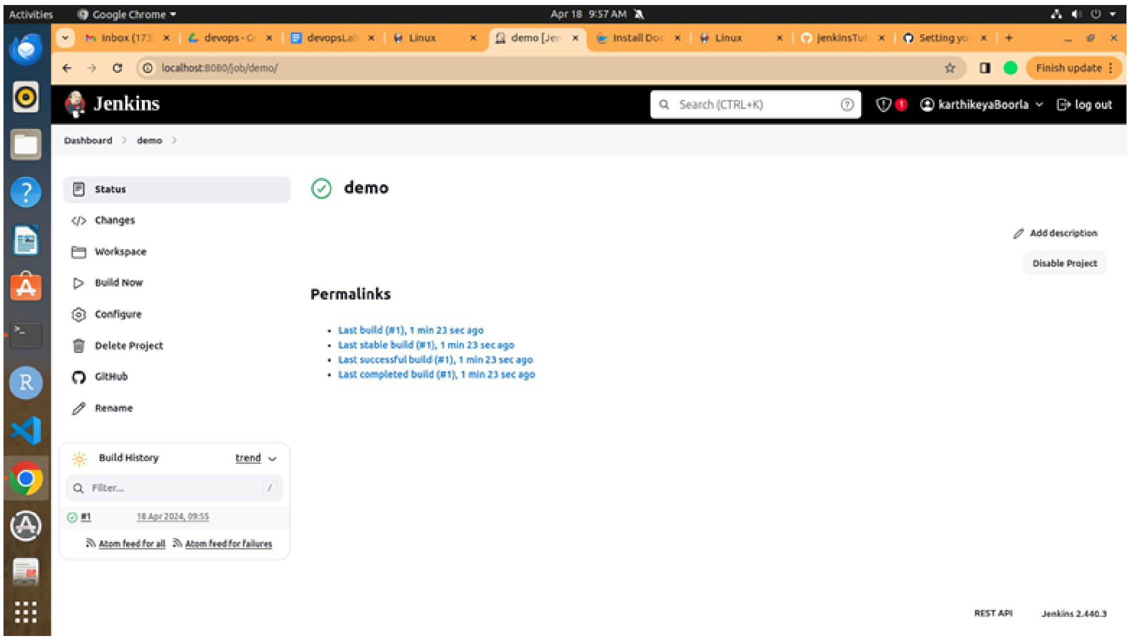
☐ Execute concurrent builds if necessary

Advanced

Source Code Management

● None

Save Apply



Activities Google Chrome Apr 18 9:57 AM

localhost:8080/job/demo/

Jenkins Search (CTRL+K) karthikeyaBoorla log out

Dashboard > demo >

Status / Changes / Workspace / Build Now / Configure / Delete Project / GitHub / Rename

demo

Add description

Disable Project

Permalinks

- Last build (#1), 1 min 23 sec ago
- Last stable build (#1), 1 min 23 sec ago
- Last successful build (#1), 1 min 23 sec ago
- Last completed build (#1), 1 min 23 sec ago

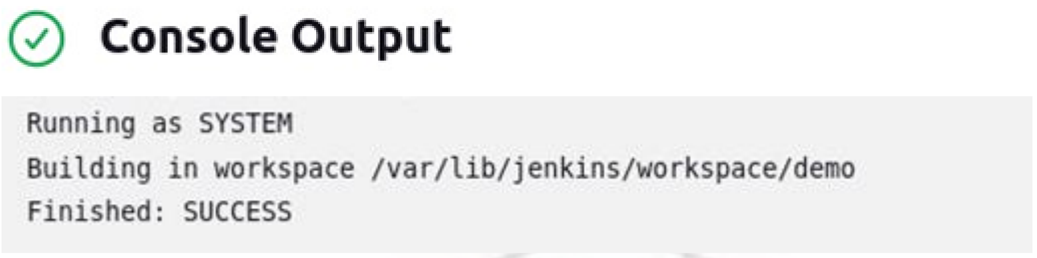
Build History trend

Filter...

#1 18 Apr 2024, 09:55

Atom feed for all Atom feed for failures

REST API Jenkins 2.440.3



## Console Output

```
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/demo
Finished: SUCCESS
```

## EXPERIMENT-3

**Aim:** To install and configure docker for creating containers for different operating systems.

**Description:**

**Installing Docker**

The following steps have to be followed to install docker and docker desktop on ubuntu.

1. **Add Docker's official GPG key:**

   sudo apt-get update

   sudo apt-get install ca-certificates curl sudo

   install -m 0755 -d /etc/apt/keyrings

   sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

   sudo chmod a+r /etc/apt/keyrings/docker.asc#

   Add the repository to Apt sources:

   echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

   $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

2. **Run the update command**

   sudo apt-get update

3. **Install docker**

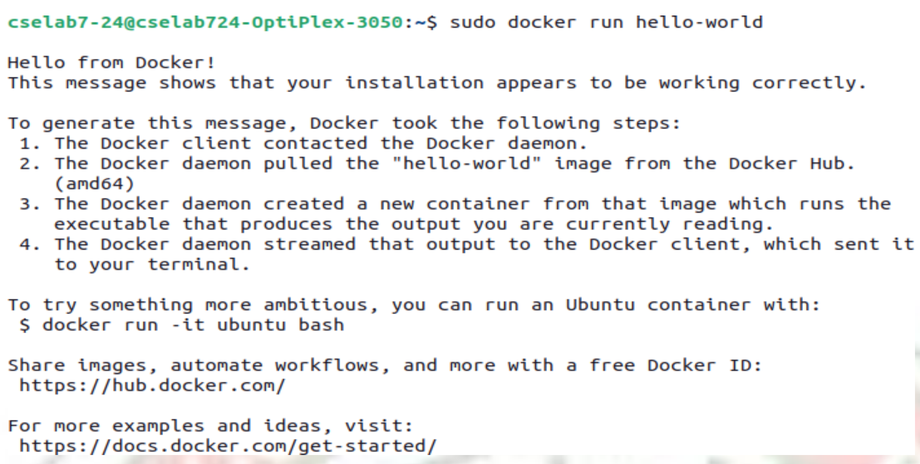
   sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

4. **If you get broken packages error fix it using**

   sudo apt --fix-broken install

5. **Run docker**

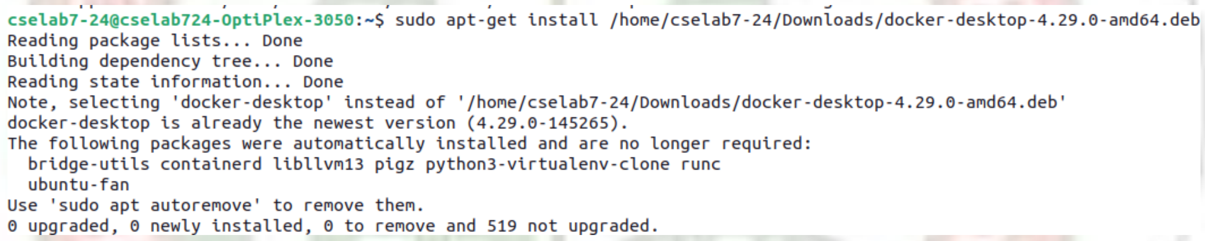
   sudo docker run hello-world



```
cselab7-24@cselab724-OptiPlex-3050:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

## 6. Installing docker desktop

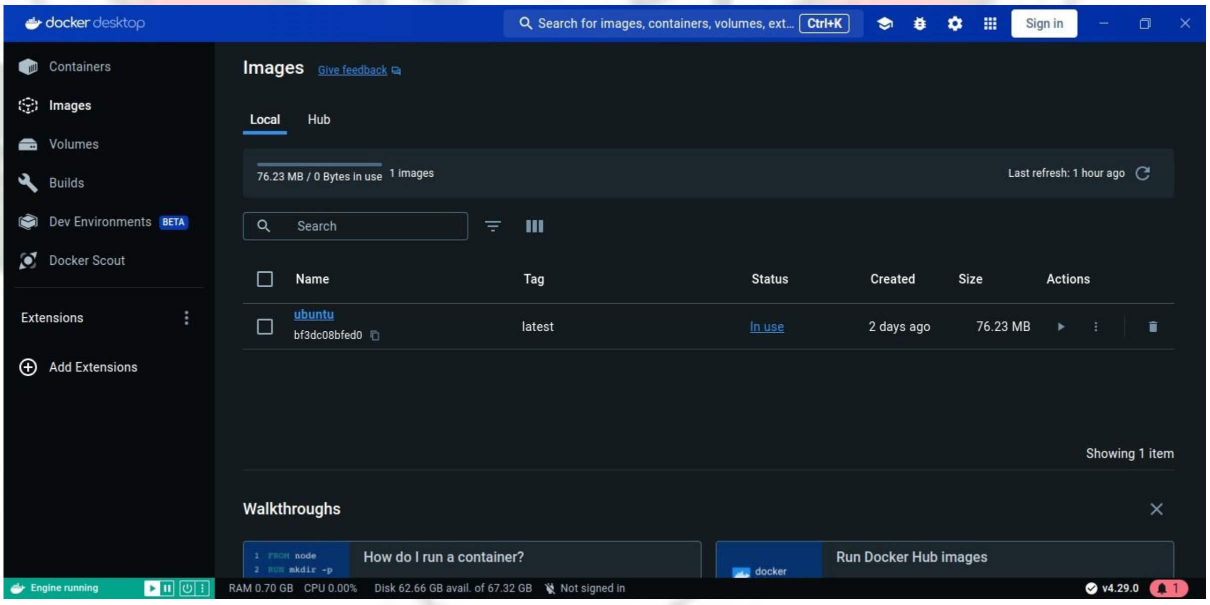
sudo apt-get install /home/cselab7-24/Downloads/docker-desktop-4.29.0-amd64.deb



```
cselab7-24@cselab724-OptiPlex-3050:~$ sudo apt-get install /home/cselab7-24/Downloads/docker-desktop-4.29.0-amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'docker-desktop' instead of '/home/cselab7-24/Downloads/docker-desktop-4.29.0-amd64.deb'
docker-desktop is already the newest version (4.29.0-145265).
The following packages were automatically installed and are no longer required:
  bridge-utils containerd libllvm13 pigz python3-virtualenv-clone runc
  ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 519 not upgraded.
```

## 7. Start Docker Desktop

systemctl --user start docker-desktop



76.23 MB / 0 Bytes in use 1 images

Last refresh: 1 hour ago

| Name | Tag | Status | Created | Size | Actions |
|---|---|---|---|---|---|
| ubuntu bf3dc08bfed0 | latest | In use | 2 days ago | 76.23 MB | |

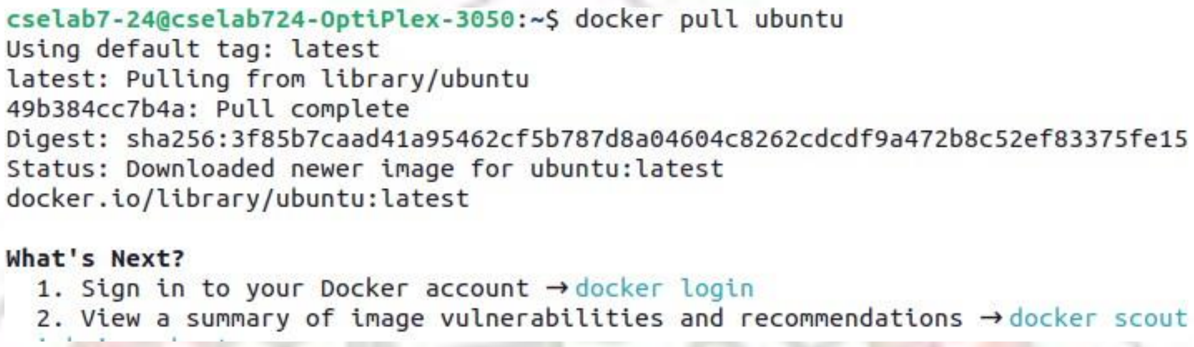
Showing 1 item

RAM 0.70 GB CPU 0.00% Disk 62.66 GB avail. of 67.32 GB Not signed in v4.29.0

**Creating containers of different OS and running commands**

## 1. Container of Ubuntu

i. Pull docker image - docker pull ubuntu

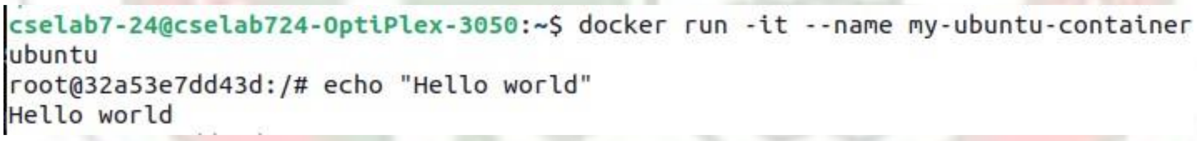


```
cselab7-24@cselab724-OptiPlex-3050:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
49b384cc7b4a: Pull complete
Digest: sha256:3f85b7caad41a95462cf5b787d8a04604c8262cdcdf9a472b8c52ef83375fe15
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout
```

ii. Run container

**docker run -it --name my-ubuntu-container ubuntu**

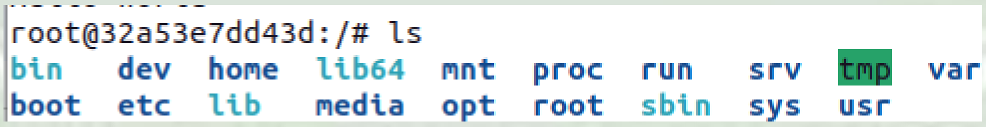
- **-it:** This flag indicates an interactive terminal session.
- **- - name my-ubuntu-container ubuntu:** Assigns a name to the container(can be replaced by your preferred name)
- **ubuntu:** specifies docker image used



```
cselab7-24@cselab724-OptiPlex-3050:~$ docker run -it --name my-ubuntu-container
ubuntu
root@32a53e7dd43d:/# echo "Hello world"
Hello world
```

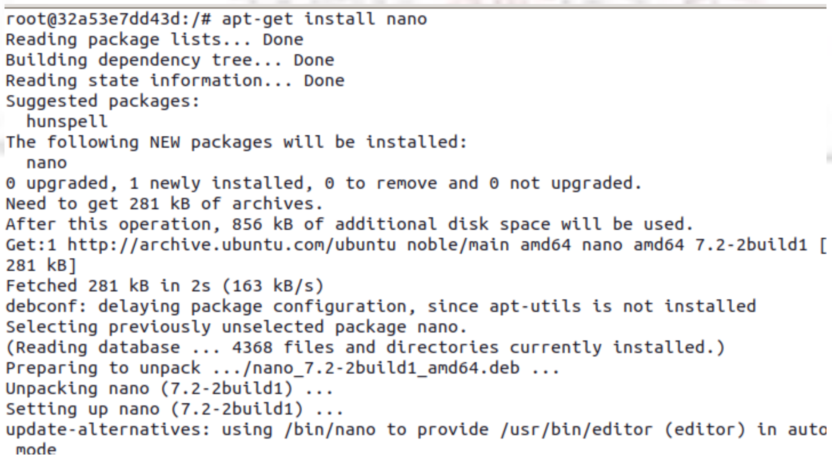
iii. Execute commands inside ubuntu container

1. ls



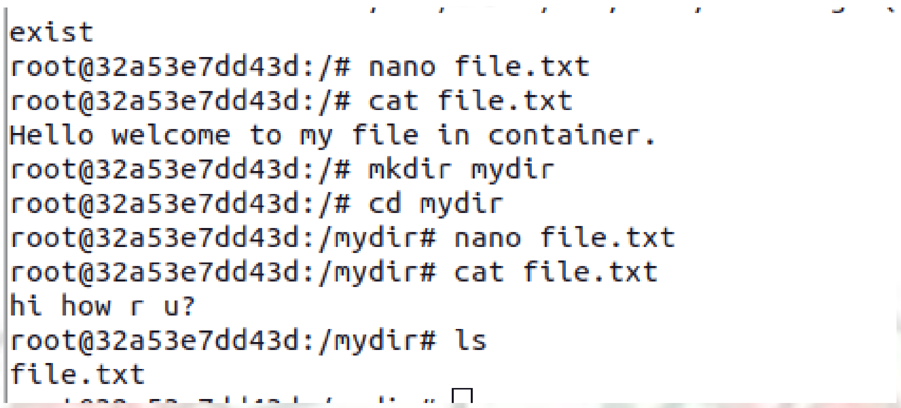
```
root@32a53e7dd43d:/# ls
bin    dev   home  lib64  mnt  proc  run   srv  tmp  var
boot   etc   lib   media  opt  root  sbin  sys  usr
```

2. Installing nano



```
root@32a53e7dd43d:/# apt-get install nano
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  hunspell
The following NEW packages will be installed:
  nano
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 281 kB of archives.
After this operation, 856 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/main amd64 nano amd64 7.2-2build1 [
281 kB]
Fetched 281 kB in 2s (163 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package nano.
(Reading database ... 4368 files and directories currently installed.)
Preparing to unpack .../nano_7.2-2build1_amd64.deb ...
Unpacking nano (7.2-2build1) ...
Setting up nano (7.2-2build1) ...
update-alternatives: using /bin/nano to provide /usr/bin/editor (editor) in auto
 mode
```

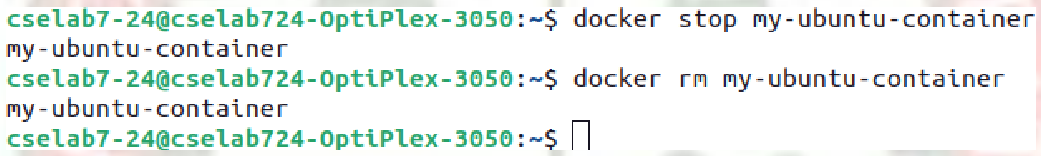
3. creating new directory - creating file in directory - displaying file - listing files



```
exist
root@32a53e7dd43d:/# nano file.txt
root@32a53e7dd43d:/# cat file.txt
Hello welcome to my file in container.
root@32a53e7dd43d:/# mkdir mydir
root@32a53e7dd43d:/# cd mydir
root@32a53e7dd43d:/mydir# nano file.txt
root@32a53e7dd43d:/mydir# cat file.txt
hi how r u?
root@32a53e7dd43d:/mydir# ls
file.txt
```

iv. Exit container using exit command

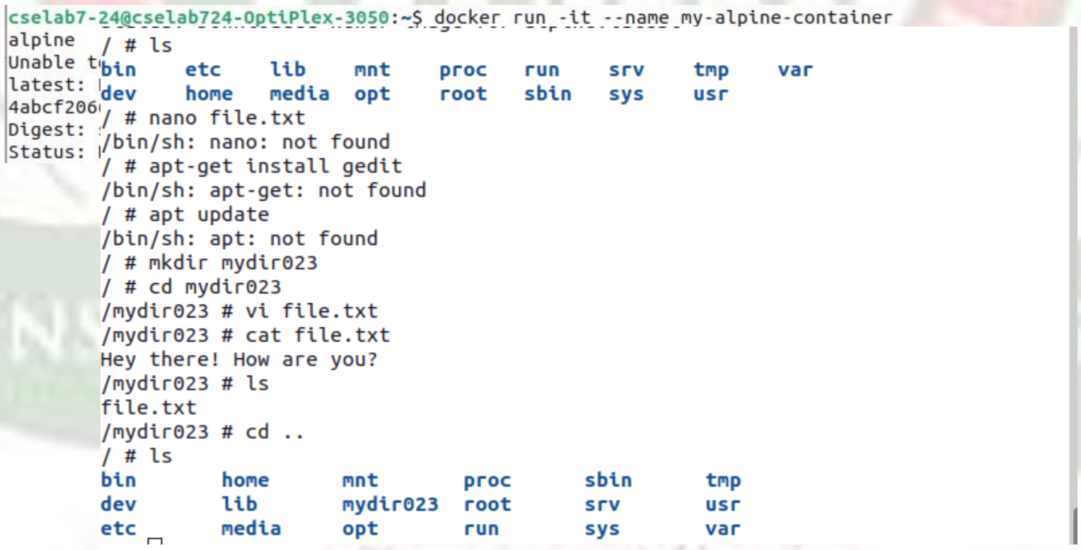
v. Stop and remove container



```
cselab7-24@cselab724-OptiPlex-3050:~$ docker stop my-ubuntu-container
my-ubuntu-container
cselab7-24@cselab724-OptiPlex-3050:~$ docker rm my-ubuntu-container
my-ubuntu-container
cselab7-24@cselab724-OptiPlex-3050:~$
```

**2. Container of Alpine**

i. Run container



```
cselab7-24@cselab724-OptiPlex-3050:~$ docker run -it --name my-alpine-container
alpine
Unable t
latest:
4abcf206
Digest:
Status:
```

```
/ # ls
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
/ # nano file.txt
/bin/sh: nano: not found
/ # apt-get install gedit
/bin/sh: apt-get: not found
/ # apt update
/bin/sh: apt: not found
/ # mkdir mydir023
/ # cd mydir023
/mydir023 # vi file.txt
/mydir023 # cat file.txt
Hey there! How are you?
/mydir023 # ls
file.txt
/mydir023 # cd ..
/ # ls
bin       home      mnt       proc      sbin      tmp
dev       lib       mydir023  root      srv       usr
etc       media     opt       run       sys       var
```

ii. Execute commands

**References:**

- https://docs.docker.com/desktop/install/ubuntu/
- https://docs.docker.com/engine/install/ubuntu/

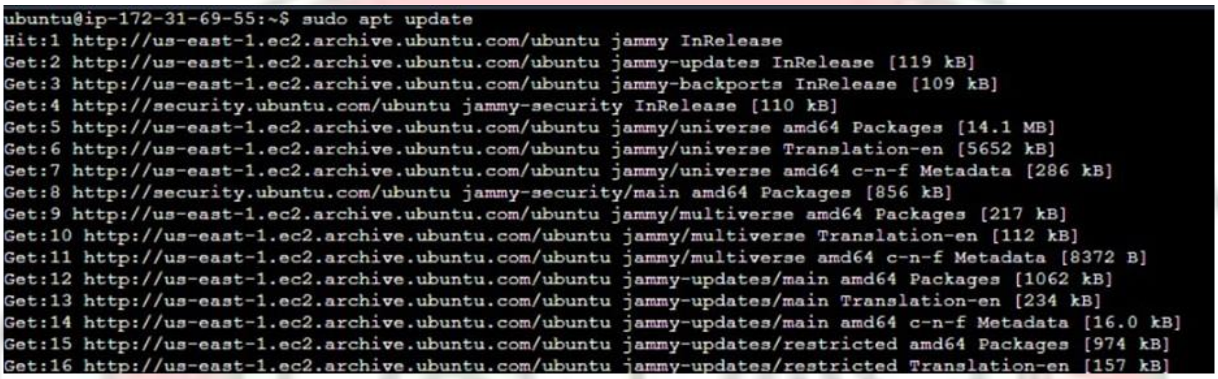
## EXPERIMENT - 4

**Aim:** Deployment Tool (Team City /Ansible) Install Docker and execute commands in aDocker and deploy the application in to Docker file

**Description:**

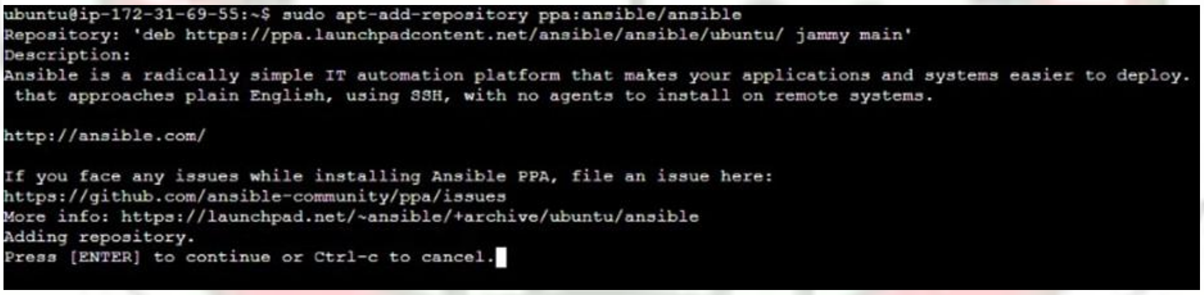
### 1. Install Dependencies

**sudo apt update**

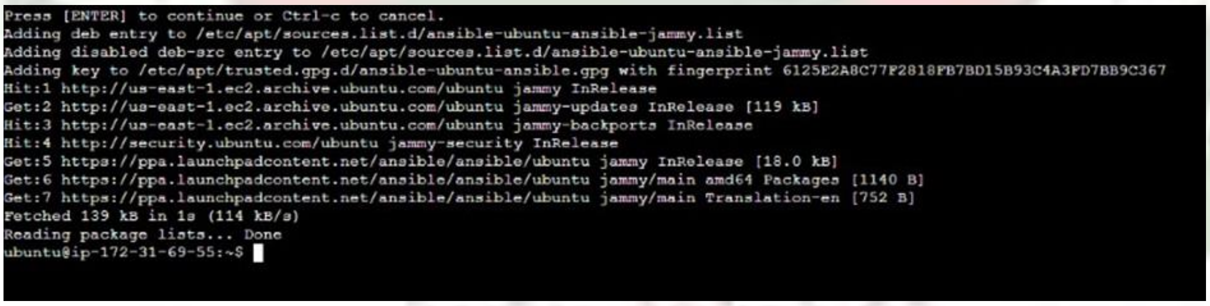


```
ubuntu@ip-172-31-69-55:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [856 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1062 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [234 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [16.0 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [974 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [157 kB]
```

**sudo apt-add-repository --yes ppa:ansible/ansible**



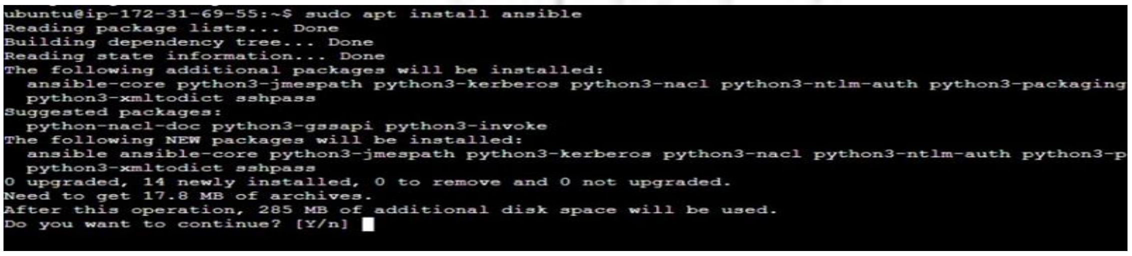
```
ubuntu@ip-172-31-69-55:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy.
 that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```



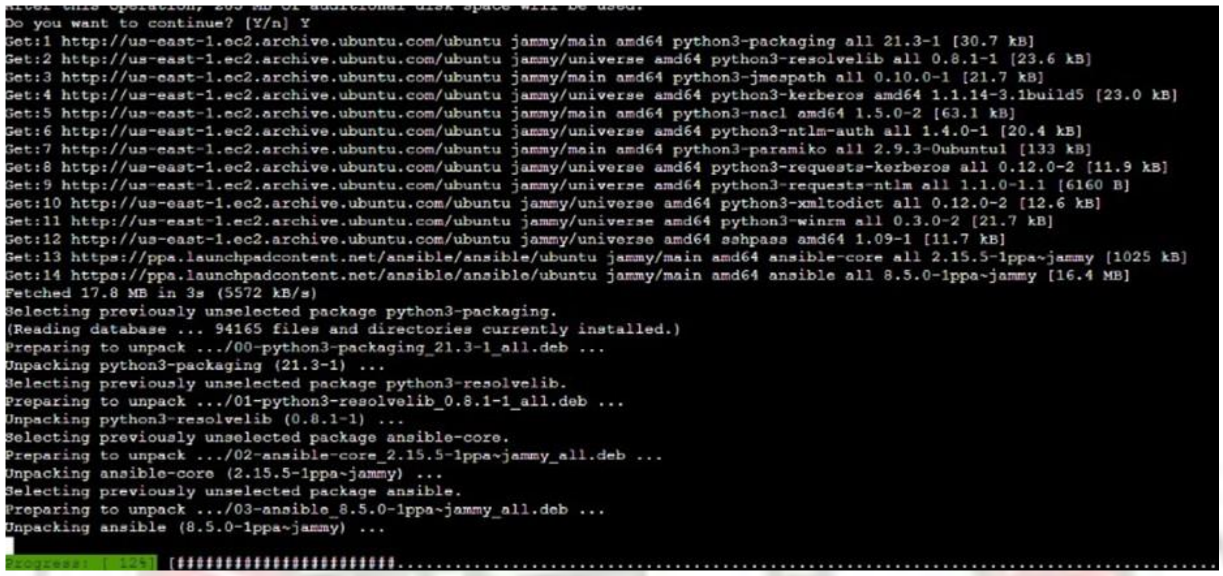
```
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.0 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1140 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [752 B]
Fetched 139 kB in 1s (114 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-69-55:~$
```

### 2. Install ansible

**sudo apt install -y ansible**



```
ubuntu@ip-172-31-69-55:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging
  python3-xmltodict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-p
  python3-xmltodict sshpass
0 upgraded, 14 newly installed, 0 to remove and 0 not upgraded.
Need to get 17.8 MB of archives.
After this operation, 285 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```



```
After this operation, 205 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-packaging all 21.3-1 [30.7 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-resolvelib all 0.8.1-1 [23.6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-jmespath all 0.10.0-1 [21.7 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-kerberos amd64 1.1.14-3.1build5 [23.0 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-nacl amd64 1.5.0-2 [63.1 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-ntlm-auth all 1.4.0-1 [20.4 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-paramiko all 2.9.3-0ubuntu1 [133 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-ntlm all 1.1.0-1.1 [6160 B]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-xmltodict all 0.12.0-2 [12.6 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-winrm all 0.3.0-2 [21.7 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 sshpass amd64 1.09-1 [11.7 kB]
Get:13 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible-core all 2.15.5-1ppa~jammy [1025 kB]
Get:14 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible all 8.5.0-1ppa~jammy [16.4 MB]
Fetched 17.8 MB in 3s (5572 kB/s)
Selecting previously unselected package python3-packaging.
(Reading database ... 94165 files and directories currently installed.)
Preparing to unpack .../00-python3-packaging_21.3-1_all.deb ...
Unpacking python3-packaging (21.3-1) ...
Selecting previously unselected package python3-resolvelib.
Preparing to unpack .../01-python3-resolvelib_0.8.1-1_all.deb ...
Unpacking python3-resolvelib (0.8.1-1) ...
Selecting previously unselected package ansible-core.
Preparing to unpack .../02-ansible-core_2.15.5-1ppa~jammy_all.deb ...
Unpacking ansible-core (2.15.5-1ppa~jammy) ...
Selecting previously unselected package ansible.
Preparing to unpack .../03-ansible_8.5.0-1ppa~jammy_all.deb ...
Unpacking ansible (8.5.0-1ppa~jammy) ...

Progress: [ 12%] [########################.....................................................................]
```

**3. Verify ansible installation**

**ansible --version**

**ERROR: A siblc ícquiícs tkc localc c codi g to bc UI'Ï-®; Dctcctcd ISO®®59-1.**

**Handling the error: It seems like Ansible is encountering an error due to the locale encoding not being set to UTF-8. To resolve this issue, you canset the locale to UTF-8. You can follow these steps:**

**i. Open the /etc/default/locale file for editing: sudo nano /etc/default/locale**

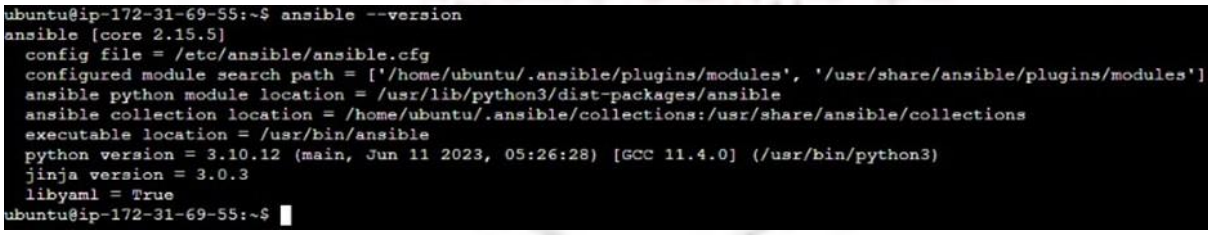
**ii. Update the file to set the locale to UTF-8. It should look like this: LANG="en_US.UTF-8"**

**LC_ALL="en_US.UTF-8"**

**iii. Save the file and exit the editor.**

**iv. Restart your system or log out and log back in to apply the changes.**

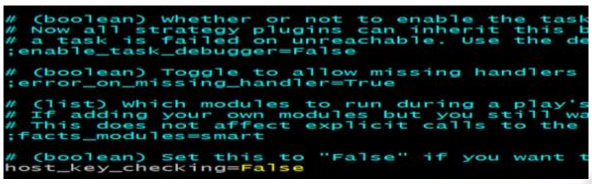
**Run version command again**



```
ubuntu@ip-172-31-69-55:~$ ansible --version
ansible [core 2.15.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
ubuntu@ip-172-31-69-55:~$
```

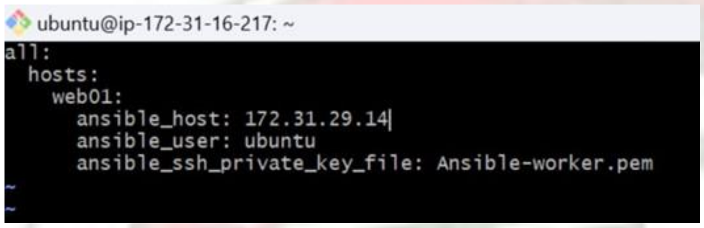
**4. Make some changes in the ansible configuration file**

**. Here assign host_key_checking to False .**



```
# (boolean) Whether or not to enable the task
# Now all strategy plugins can inherit this b
# a task is failed on unreachable. Use the de
;enable_task_debugger=False

# (boolean) Toggle to allow missing handlers
;error_on_missing_handler=True

# (list) Which modules to run during a play's
# If adding your own modules but you still wa
# This does not affect explicit calls to the
;facts_modules=smart

# (boolean) Set this to "False" if you want t
host_key_checking=False
```

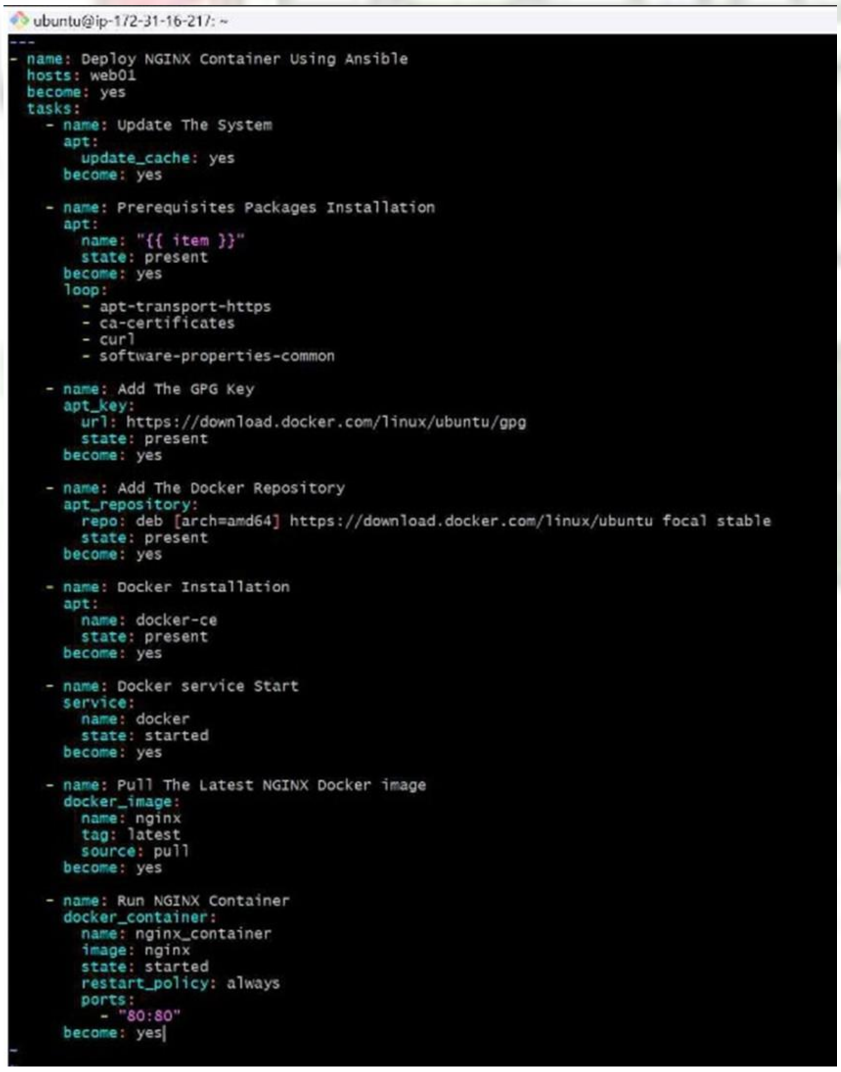
**5.** **Create an Inventory file, mentioning the private IP of worker node, machine nameand the private key.**



ubuntu@ip-172-31-16-217: ~

```
all:
  hosts:
    web01:
      ansible_host: 172.31.29.14
      ansible_user: ubuntu
      ansible_ssh_private_key_file: Ansible-worker.pem
~
~
```

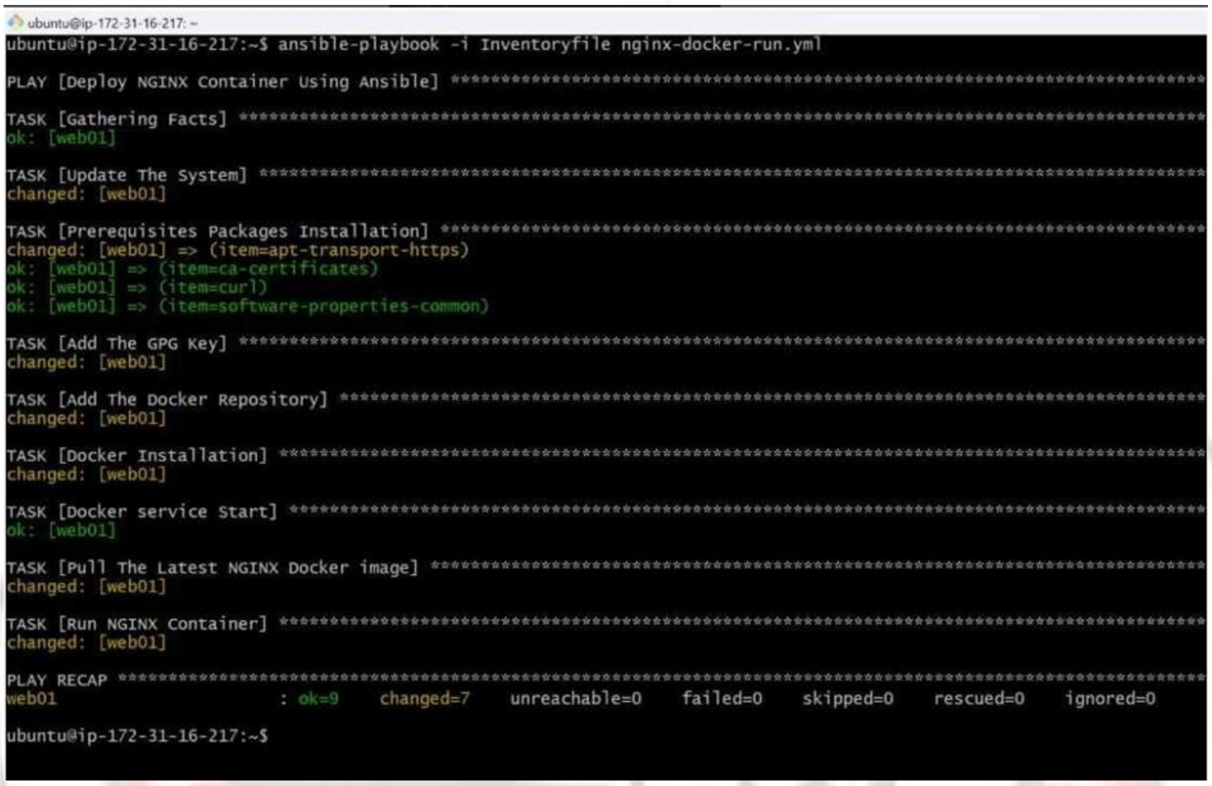
**6. Create an ansible playbook which will install docker on the worker node and runa nginx container on port 80**



ubuntu@ip-172-31-16-217: ~

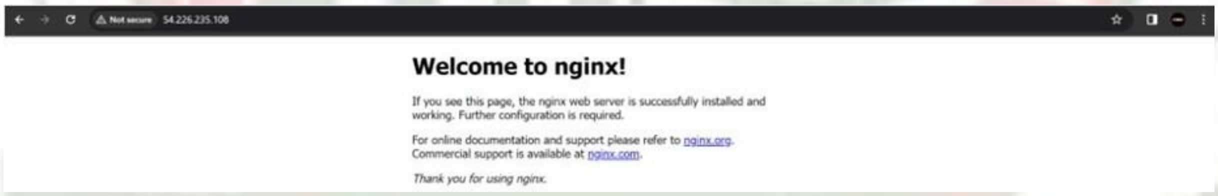
```
---
- name: Deploy NGINX Container Using Ansible
  hosts: web01
  become: yes
  tasks:
    - name: Update The System
      apt:
        update_cache: yes
      become: yes

    - name: Prerequisites Packages Installation
      apt:
        name: "{{ item }}"
        state: present
      become: yes
      loop:
        - apt-transport-https
        - ca-certificates
        - curl
        - software-properties-common

    - name: Add The GPG Key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present
      become: yes

    - name: Add The Docker Repository
      apt_repository:
        repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable
        state: present
      become: yes

    - name: Docker Installation
      apt:
        name: docker-ce
        state: present
      become: yes

    - name: Docker service Start
      service:
        name: docker
        state: started
      become: yes

    - name: Pull The Latest NGINX Docker image
      docker_image:
        name: nginx
        tag: latest
        source: pull
      become: yes

    - name: Run NGINX Container
      docker_container:
        name: nginx_container
        image: nginx
        state: started
        restart_policy: always
        ports:
          - "80:80"
      become: yes
-
```

7. **Runinng the ansible playbook.**



```
ubuntu@ip-172-31-16-217: ~
ubuntu@ip-172-31-16-217:~$ ansible-playbook -i Inventoryfile nginx-docker-run.yml

PLAY [Deploy NGINX Container Using Ansible] ************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [web01]

TASK [Update The System] *******************************************************************************
changed: [web01]

TASK [Prerequisites Packages Installation] *************************************************************
changed: [web01] => (item=apt-transport-https)
ok: [web01] => (item=ca-certificates)
ok: [web01] => (item=curl)
ok: [web01] => (item=software-properties-common)

TASK [Add The GPG Key] *********************************************************************************
changed: [web01]

TASK [Add The Docker Repository] ***********************************************************************
changed: [web01]

TASK [Docker Installation] *****************************************************************************
changed: [web01]

TASK [Docker service Start] ****************************************************************************
ok: [web01]

TASK [Pull The Latest NGINX Docker image] **************************************************************
changed: [web01]

TASK [Run NGINX Container] *****************************************************************************
changed: [web01]

PLAY RECAP *********************************************************************************************
web01                      : ok=9    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-16-217:~$
```

8. **After this connect the public IP of worker node with port 80 on the browser.4**



Not secure 54.226.235.108

**Welcome to nginx!**

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

*Thank you for using nginx.*

**References:**

- https://www.geeksforgeeks.org/automating-docker-deployments-with-ansible/

## EXPERIMENT -5

**AIM**: Test the Application using selenium tool

**Description:**

Selenium is a popular open-source software testing framework used for automating web applications.
It is widely used for functional testing, regression testing, and performance testing.
Selenium supports multiple programming languages, including Java, C#, Python, and Ruby,
making it accessible to a wide range of developers.

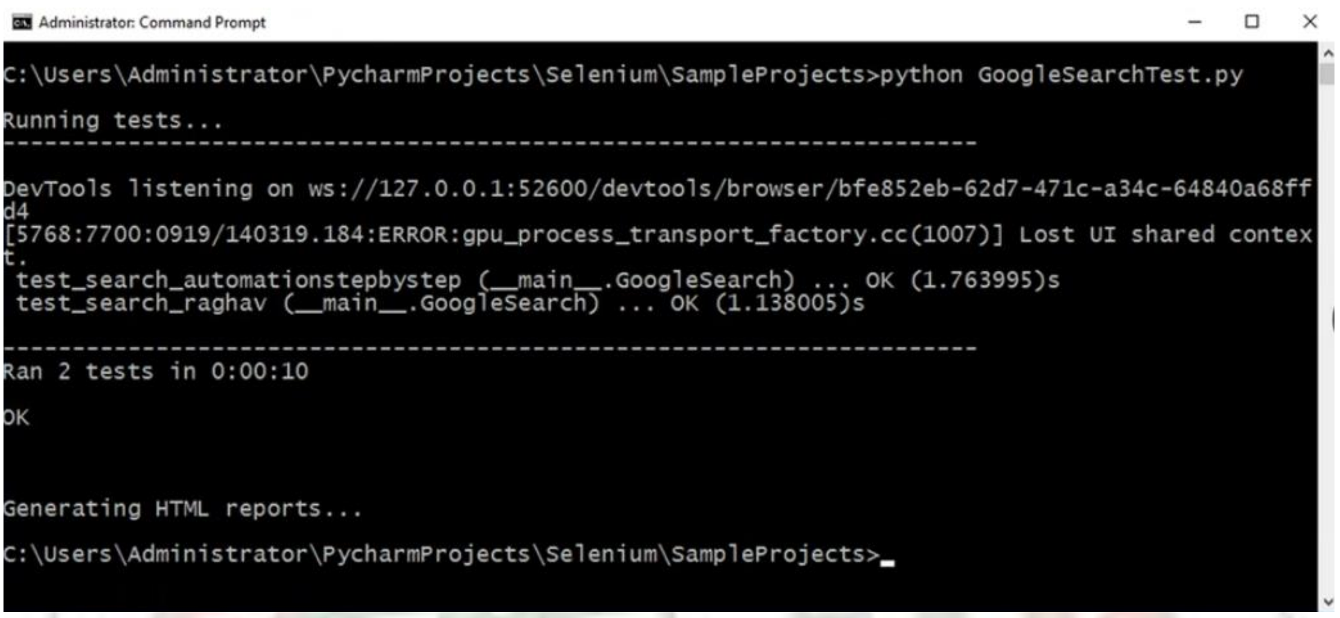
Selenium is an Automation Tool and portable software testing tool for web applications.
A test domain-specific language is also provided, to write test cases one can use programming languages, including C#, Java, Perl, PHP, Python, Ruby, Scala, and Groovy.
It does not support RIA(Rich Internet Application) Technology such as Silverlight JavaFX and Flex\Flash.
Selenium can be easily used on platforms like Windows, Linux, Solaris, and Macintosh.
Selenium also supports Operating Systems for mobile applications like Android, iOS, and Windows. Selenium uses many programming languages like Ruby, python C#, Java, Perl, and PHP. Safari, Internet Explorer, Mozilla Firefox, and Google Chrome are some of the browsers that are supported by Selenium.

1. Selenium provides several tools that allow developers to automate their tests, including:
2. Selenium WebDriver: A tool for automating browser interactions and for controlling web browsers programmatically.
3. Selenium IDE: A browser-based tool for recording and playing back user interactions with a web application.
4. Selenium Grid: A tool for running tests in parallel on multiple machines, allowing for faster test execution and improved resource utilization.
5. Selenium is an important tool in the software engineering process as it enables developers to automate repetitive and time-consuming testing tasks, freeing up their time to focus on other aspects of the software development process. The ability to automate testing also helps to ensure that software is thoroughly tested and that issues are identified and resolved more quickly, improving the overall quality of the software.

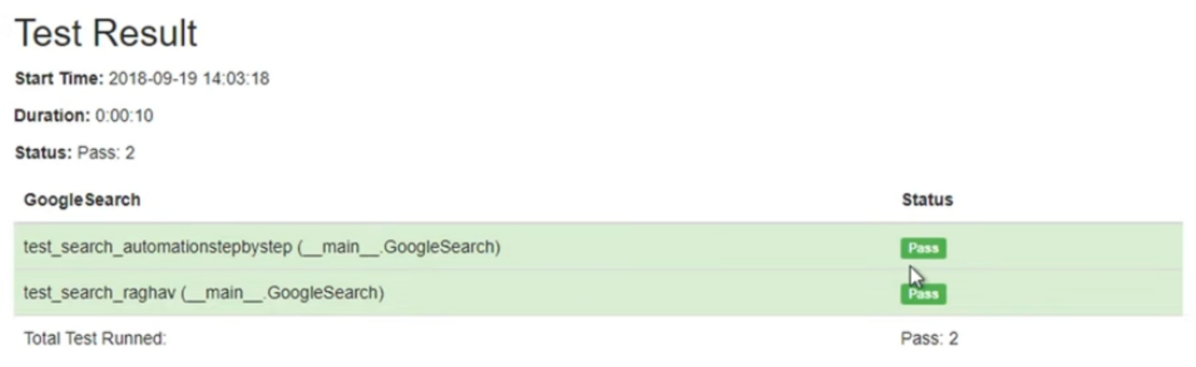
**Procedure:**

Write the Code for Unit Testing of Google.com

```
from selenium import webdriver
import unittest
import HtmlTestRunner

class GoogleSearch(unittest.TestCase):

    @classmethod
    def setUpClass(cls):
        cls.driver = webdriver.Chrome(executable_path='../drivers/chromedriver.exe')
        cls.driver.implicitly_wait(10)
        cls.driver.maximize_window()

    def test_search_automationstepbystep(self):
        self.driver.get("https://google.com")
        self.driver.find_element_by_name("q").send_keys("DEVOPS")
        self.driver.find_element_by_name("btnK").click()

    def test_search_raghav(self):
        self.driver.get("https://google.com")
        self.driver.find_element_by_name("q").send_keys("CBIT")
        self.driver.find_element_by_name("btnK").click()

    @classmethod
    def tearDownClass(cls):
        cls.driver.close()
        cls.driver.quit()
        print("Test Completed")


if __name__ == '__main__':

unittest.main(testRunner=HtmlTestRunner.HTMLTestRunner(output='C:/Users/Administrator/PycharmProjects/
Selenium/reports'))
```



```
Administrator: Command Prompt
C:\Users\Administrator\PycharmProjects\Selenium\SampleProjects>python GoogleSearchTest.py

Running tests...
----------------------------------------------------------------------

DevTools listening on ws://127.0.0.1:52600/devtools/browser/bfe852eb-62d7-471c-a34c-64840a68ff
d4
[5768:7700:0919/140319.184:ERROR:gpu_process_transport_factory.cc(1007)] Lost UI shared contex
t.
 test_search_automationstepbystep (__main__.GoogleSearch) ... OK (1.763995)s
 test_search_raghav (__main__.GoogleSearch) ... OK (1.138005)s

----------------------------------------------------------------------
Ran 2 tests in 0:00:10

OK


Generating HTML reports...

C:\Users\Administrator\PycharmProjects\Selenium\SampleProjects>_
```



Test Result

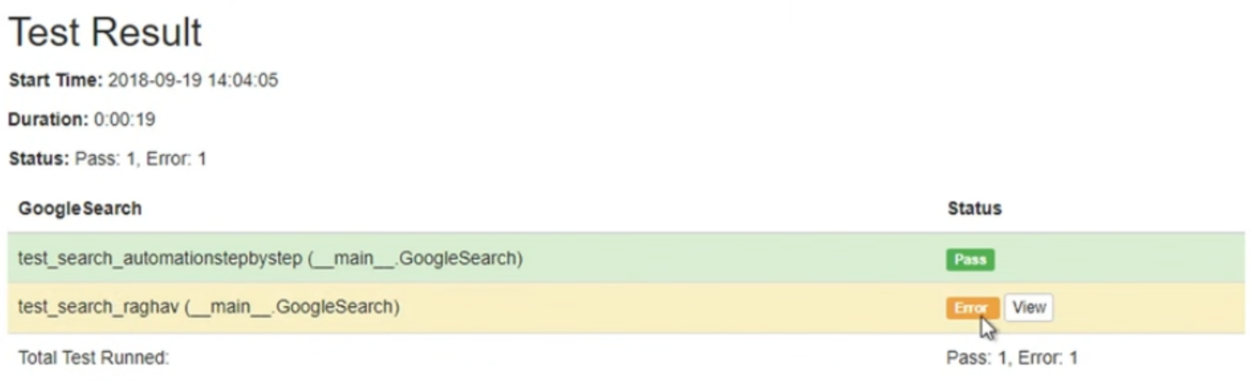
**Start Time:** 2018-09-19 14:03:18

**Duration:** 0:00:10

**Status:** Pass: 2

| GoogleSearch | Status |
|---|---|
| test_search_automationstepbystep (__main__.GoogleSearch) | Pass |
| test_search_raghav (__main__.GoogleSearch) | Pass |
| Total Test Runned: | Pass: 2 |

If you deliberately throw an invalid syntax. This is the error



## Test Result

**Start Time:** 2018-09-19 14:04:05

**Duration:** 0:00:19

**Status:** Pass: 1, Error: 1

| GoogleSearch | Status |
|---|---|
| test_search_automationstepbystep (__main__.GoogleSearch) | Pass |
| test_search_raghav (__main__.GoogleSearch) | Error View |
| Total Test Runned: | Pass: 1, Error: 1 |

# EXPERIMENT-6

**Aim:** Configuring and Establishing Connection between puppet agent and puppet master.

**Description:**

Puppet is an open source configuration management system. With its domain-specific language, Puppet allows system administrators to perform system configuration in a declarative way. Since Puppet configurations consist of text, they can easily be versioned, which means you can easily run diffs against competing versions to understand exactly what changed.

**Puppet Agent: The Basics**

Puppet's architecture follows the server-client model, although it uses a slightly different nomenclature. A client in this model is called a Puppet agent. One or more Puppet servers are installed on the server

Agents are installed on the machines that are to be managed. Puppet works on a pull model, which means that agents periodically query the centralized server for updates. The server compiles these updates in the form of catalogs and sends them back to the agents.

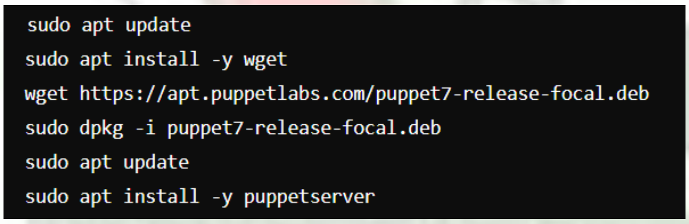
Upon receiving its catalog, each node analyzes it, iterating over its resources and comparingit with the state of its actual components. When the node detects that something needs to be changed, it carries out the necessary tasks to undergo the transformation.

After the changes are applied, the agents send a report to the centralized server

**Procedure:**

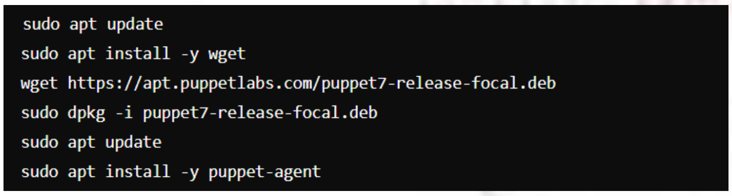
**Step 1: Install Puppet on Both Master and Agent**

1. Puppet Master Installation: On the Puppet master server, add the Puppet repositoryand install Puppet server.



```
sudo apt update
sudo apt install -y wget
wget https://apt.puppetlabs.com/puppet7-release-focal.deb
sudo dpkg -i puppet7-release-focal.deb
sudo apt update
sudo apt install -y puppetserver
```

2. Puppet Agent Installation: On the Puppet agent node, add the Puppet repository and install the Puppet agent.



```
sudo apt update
sudo apt install -y wget
wget https://apt.puppetlabs.com/puppet7-release-focal.deb
sudo dpkg -i puppet7-release-focal.deb
sudo apt update
sudo apt install -y puppet-agent
```

**Step 2: Configure Puppet Master**

1. Configure puppetserver settings in /etc/puppetlabs/puppet/puppet.conf:

```
[main]
certname = puppetmaster.example.com
server = puppetmaster.example.com
environment = production
runinterval = 1h

[master]
dns_alt_names = puppet,puppetmaster,puppetmaster.example.com
```

2. Start and enable the Puppet master service:

```
sudo systemctl start puppetserver
sudo systemctl enable puppetserver
```

**Step 3: Configure Puppet Agent**

1. Configure the Puppet agent settings in /etc/puppetlabs/puppet/puppet.conf:

```
[main]
certname = puppetagent.example.com
server = puppetmaster.example.com
environment = production
```

2. Start and enable the Puppet agent service:

```
sudo systemctl start puppet
sudo systemctl enable puppet
```

**Step 4: Establish SSL Certificate**

1. On the Puppet Agent, initiate a certificate signing request (CSR):

```
sudo puppet agent --test --waitforcert 60
```

2. On the Puppet Master, list and sign the certificate request:

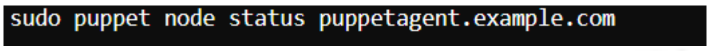
```
sudo puppetserver ca list --all
sudo puppetserver ca sign --certname puppetagent.example.com
```

3. Verify the connection on the Puppet agent:

```
sudo puppet agent --test
```

**Step 5: Verify Communication**

1. On the Puppet Master, verify the agent node:

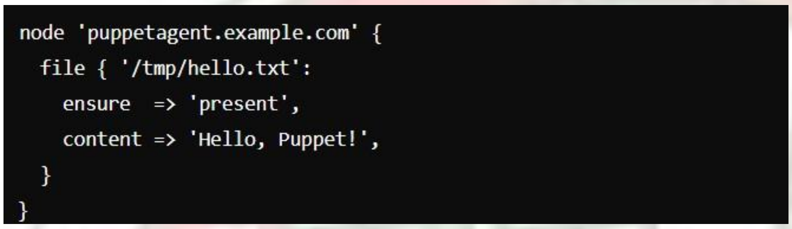


```
sudo puppet node status puppetagent.example.com
```

2. Check the reports on the Puppet Master to ensure that the agent is reportingproperly.

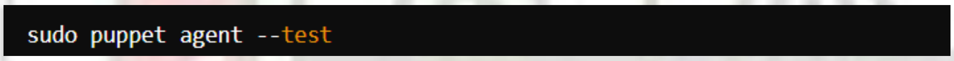
**Step 6: Manage Puppet Modules**

1. Create a simple manifest in /etc/puppetlabs/code/environments/production/manifests/site.pp:



```
node 'puppetagent.example.com' {
  file { '/tmp/hello.txt':
    ensure  => 'present',
    content => 'Hello, Puppet!',
  }
}
```

2. Run the Puppet agent to apply the configuration:



```
sudo puppet agent --test
```

**References:**

- https://www.cloudbees.com/blog/install-and-configure-puppet-agent

## EXPERIMENT-7

**AIM:** Install code monitoring tools ex: Nagios..Perform operations

**DESCRIPTION:**

Nagios is a free and open-source software application for computer systems. It is used for monitoring the systems, networks and infrastructure.

Its original name was NetSaint developed by Ethan Galstad with the group of some developers in the year 1999.

This software application mainly provides the services of monitoring and alerting for switches, applications, and servers inside the DevOps culture.

It is also used to notify the users when the things go bad, and also alerts them when the things become better.

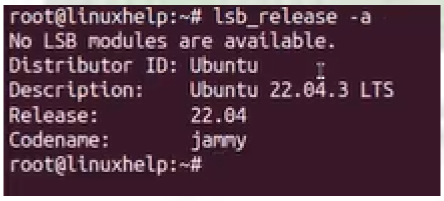
Why Nagios?

Following are some reasons for using the Nagios software application:

It is used for monitoring the performance issues of servers.
It helps the users of this software application to easily find the root cause of any problem.
It is also used to detect all the possible networks.
We can easily maintain the issues of security and also detect the availability of the services.
This application automatically fix the problems or issues when occurs.
Users can easily run it on any operating system.
Before the failure of a system, it helps you to update the infrastructure.
Using this application, user can quickly detect any type of infrastructure issues.
It also monitors the various servers of database such as SQL Server, MySQL.

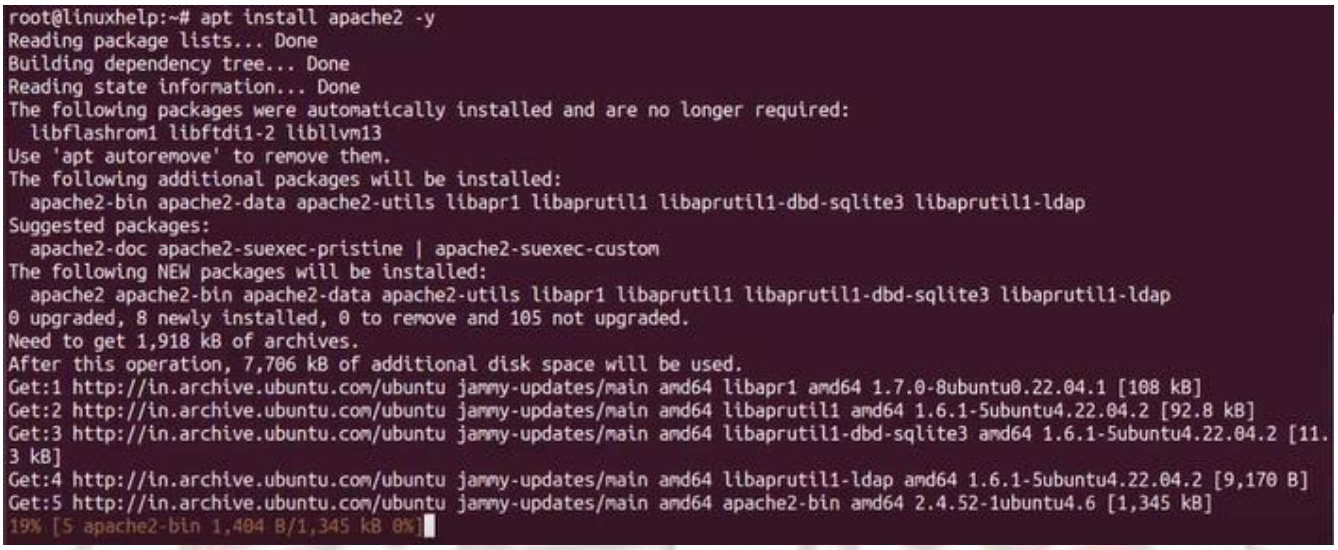
**PROCEDURE:**

Step-1 : Check LSB Release



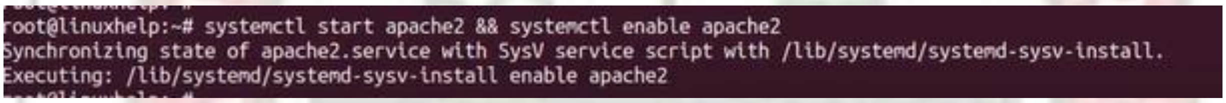
```
root@linuxhelp:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
root@linuxhelp:~#
```

Step-2: Install APACHE (apt install apache2 -y)



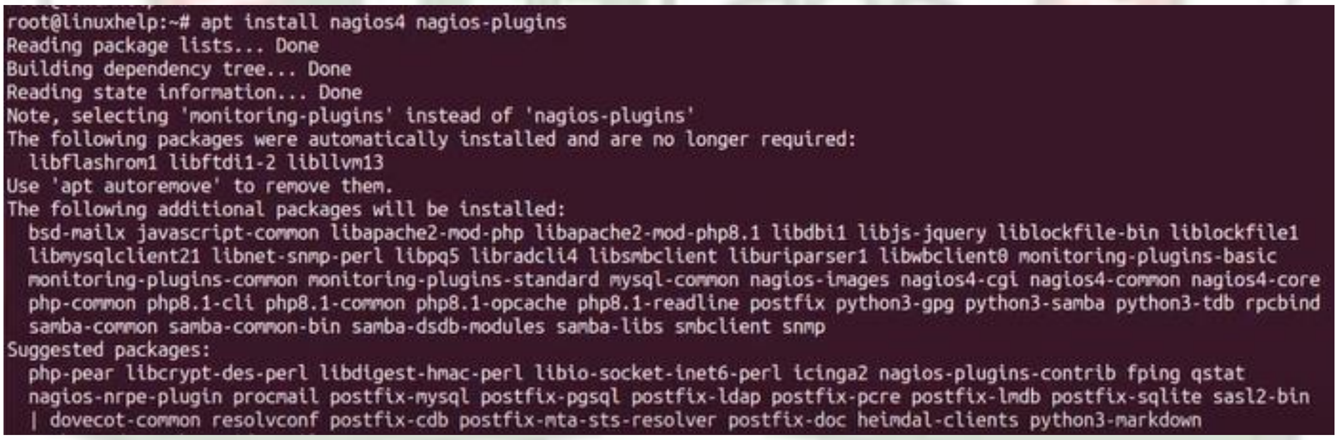
```
root@linuxhelp:~# apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 105 not upgraded.
Need to get 1,918 kB of archives.
After this operation, 7,706 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.
3 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9,170 B]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.6 [1,345 kB]
19% [5 apache2-bin 1,404 B/1,345 kB 0%]
```

STEP-3: Enable the apache service (systemctl start apache2 && systemctl enable apache2)



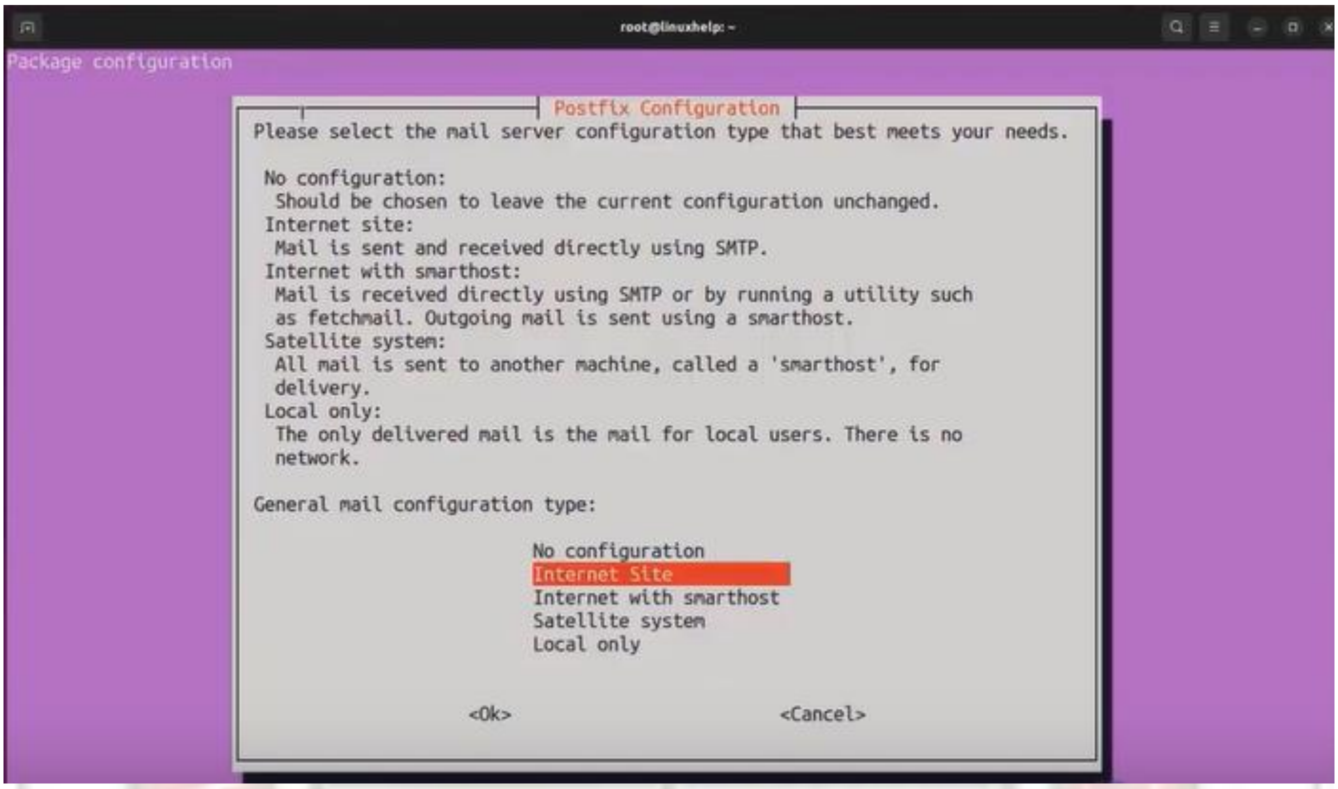
```
root@linuxhelp:~# systemctl start apache2 && systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

STEP-4: Install Nagios (apt install nagios4 nagios-plugins)



```
root@linuxhelp:~# apt install nagios4 nagios-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'monitoring-plugins' instead of 'nagios-plugins'
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  bsd-mailx javascript-common libapache2-mod-php libapache2-mod-php8.1 libdbi1 libjs-jquery liblockfile-bin liblockfile1
  libmysqlclient21 libnet-snmp-perl libpq5 libradcli4 libsmbclient liburiparser1 libwbclient0 monitoring-plugins-basic
  monitoring-plugins-common monitoring-plugins-standard mysql-common nagios-images nagios4-cgi nagios4-common nagios4-core
  php-common php8.1-cli php8.1-common php8.1-opcache php8.1-readline postfix python3-gpg python3-samba python3-tdb rpcbind
  samba-common samba-common-bin samba-dsdb-modules samba-libs smbclient snmp
Suggested packages:
  php-pear libcrypt-des-perl libdigest-hmac-perl libio-socket-inet6-perl icinga2 nagios-plugins-contrib fping qstat
  nagios-nrpe-plugin procmail postfix-mysql postfix-pgsql postfix-ldap postfix-pcre postfix-lmdb postfix-sqlite sasl2-bin
  | dovecot-common resolvconf postfix-cdb postfix-mta-sts-resolver postfix-doc heimdal-clients python3-markdown
```

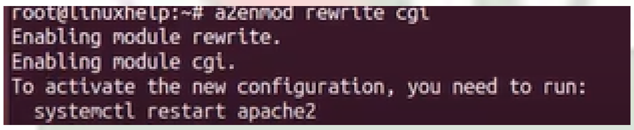
STEP-5 POSTFIX CONFIGURATION



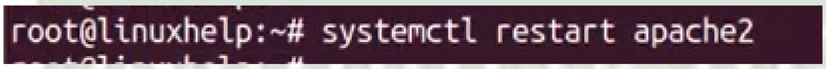
```
root@linuxhelp: ~
Package configuration

                          Postfix Configuration
 Please select the mail server configuration type that best meets your needs.

  No configuration:
   Should be chosen to leave the current configuration unchanged.
  Internet site:
   Mail is sent and received directly using SMTP.
  Internet with smarthost:
   Mail is received directly using SMTP or by running a utility such
   as fetchmail. Outgoing mail is sent using a smarthost.
  Satellite system:
   All mail is sent to another machine, called a 'smarthost', for
   delivery.
  Local only:
   The only delivered mail is the mail for local users. There is no
   network.

 General mail configuration type:

                          No configuration
                          Internet Site
                          Internet with smarthost
                          Satellite system
                          Local only

                 <Ok>                           <Cancel>
```

STEP-6 : Enable Module

(a2emod rewrite cgi)
Also run systemctl restart apache2

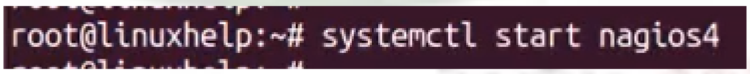


```
root@linuxhelp:~# a2enmod rewrite cgi
Enabling module rewrite.
Enabling module cgi.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

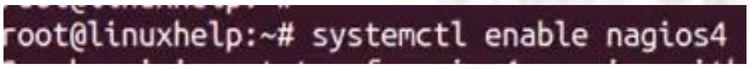


```
root@linuxhelp:~# systemctl restart apache2
```

STEP-7: Enable Nagios service (systemctl start nagios4) & (systemctl enable nagios4)

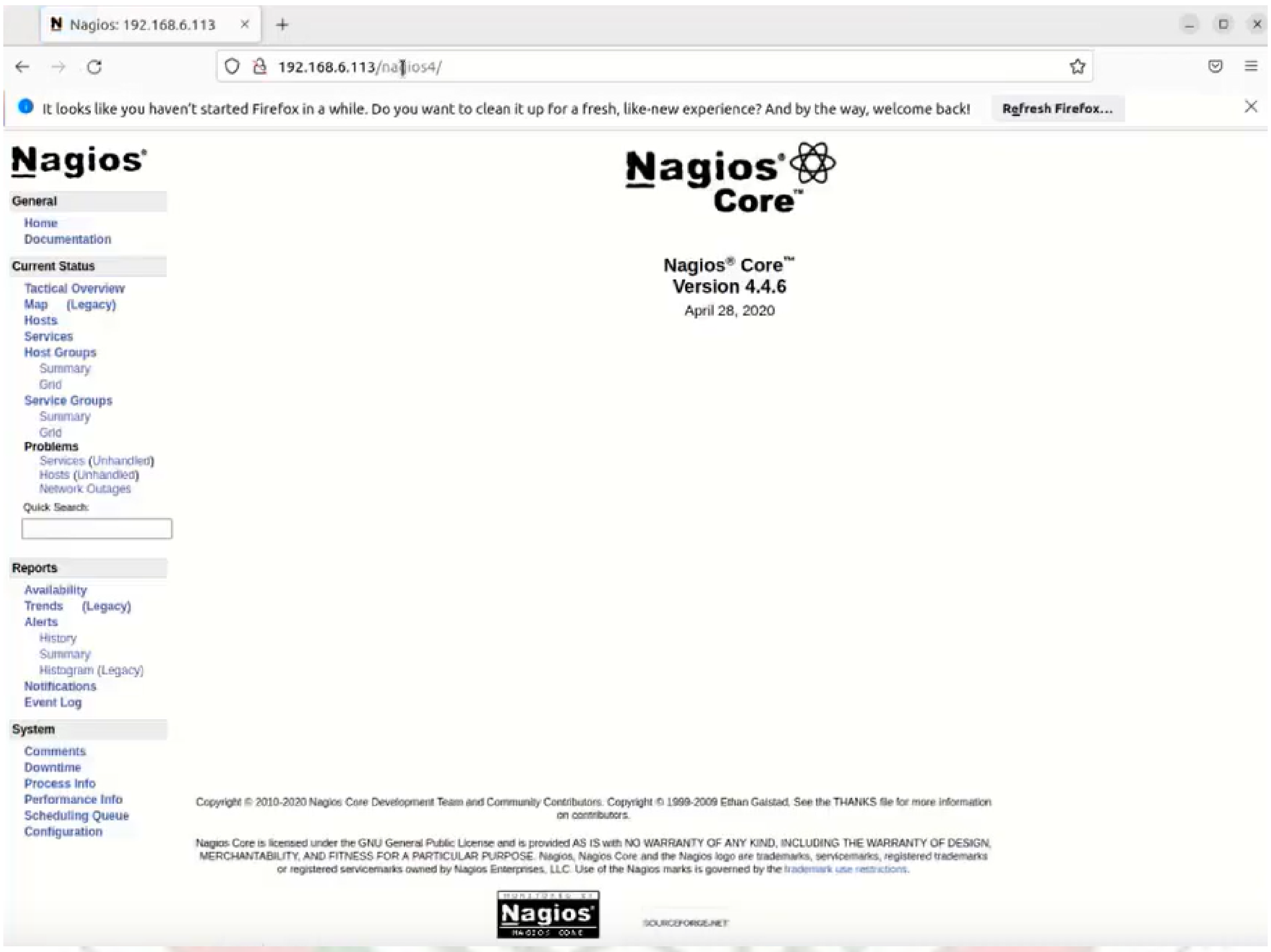


```
root@linuxhelp:~# systemctl start nagios4
```



```
root@linuxhelp:~# systemctl enable nagios4
```

STEP-8 GO to browser and type this URL (192.168.6.133/nagios4)



Nagios® Core™
Version 4.4.6
April 28, 2020

Copyright © 2010-2020 Nagios Core Development Team and Community Contributors. Copyright © 1999-2009 Ethan Galstad. See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Use of the Nagios marks is governed by the trademark use restrictions.

SOURCEFORGE.NET

# EXPERIMENT-8

**AIM:** Install issue tracker and monitor the workflow of any application and track the issues JIRA tool (Agile management tool)

**DESCRIPTION:**

JIRA is a popular project management tool developed by Atlassian, designed primarily for issue tracking and agile project management. It allows teams to plan, track, and manage software development projects efficiently. JIRA supports various agile methodologies, including Scrum and Kanban, making it a versatile tool for different project management needs.

Installing JIRA is the first step in leveraging its powerful features for managing projects. It provides a centralized platform to track issues, monitor workflows, and facilitate communication among team members.

JIRA provides a centralized platform for managing tasks, bugs, and other types of issues, and it helps teams organize and prioritize their work. JIRA is designed for agile software development teams and it supports multiple methodologies such as Scrum, Kanban, and custom workflows.

JIRA is used for:

Project Management: JIRA provides a centralized platform for managing software development projects, with support for multiple projects and workflows.

Task Management: Teams can create, assign, and track tasks, bugs, and other types of issues.

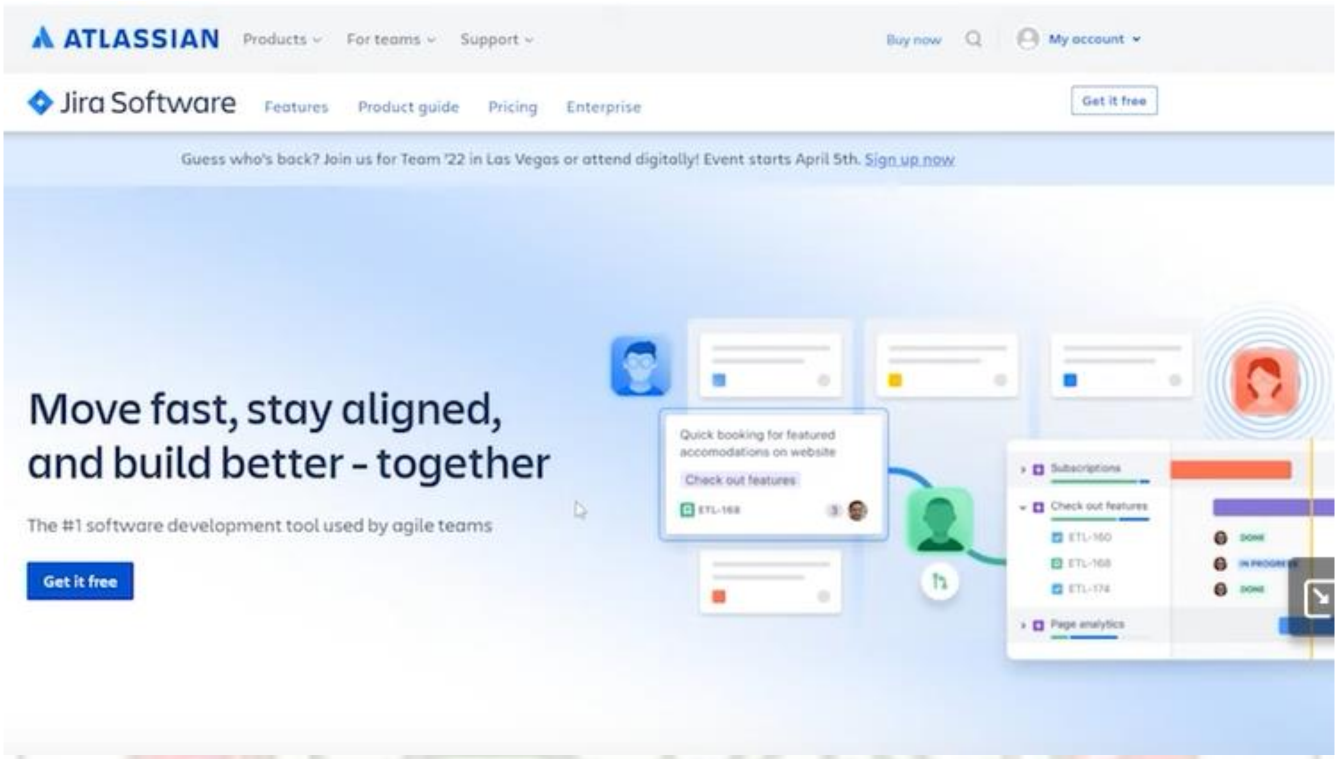
Agile Planning: JIRA supports agile methodologies such as Scrum and Kanban and provides tools for planning and tracking sprints, backlogs, and releases.

Reporting and Dashboards: JIRA provides various reports and dashboards that help teams get a real-time view of their work and make data-driven decisions.

Collaboration: JIRA allows teams to collaborate and communicate effectively, with features such as comments, notifications, and alerts.

**PROCEDURE:**

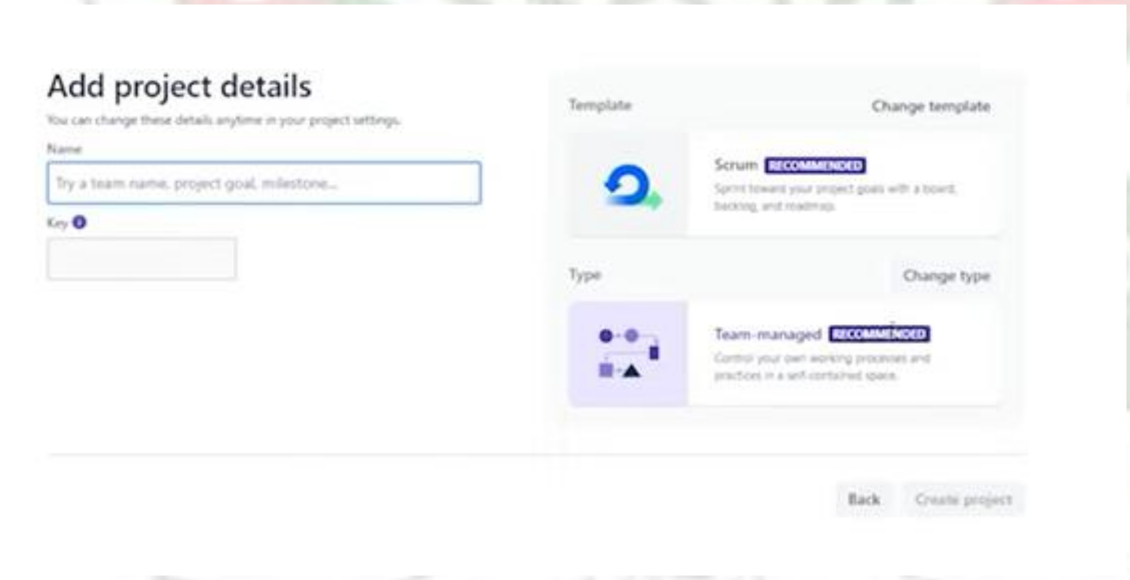
Step-1: Go to Atlassian Website (Atlassian.com)


ATLASSIAN
Products
For teams
Support
Buy now
My account
Jira Software
Features
Product guide
Pricing
Enterprise
Get it free
Guess who's back? Join us for Team '22 in Las Vegas or attend digitally! Event starts April 5th. Sign up now
Move fast, stay aligned, and build better - together
The #1 software development tool used by agile teams
Get it free

Click “Get it free”

Sign Up – Add Your Site

Step-2 : Add Project Details


Add project details

You can change these details anytime in your project settings.

Name

Try a team name, project goal, milestone...

Key

Template — Change template

Scrum RECOMMENDED — Sprint toward your project goals with a board, backlog, and roadmap.

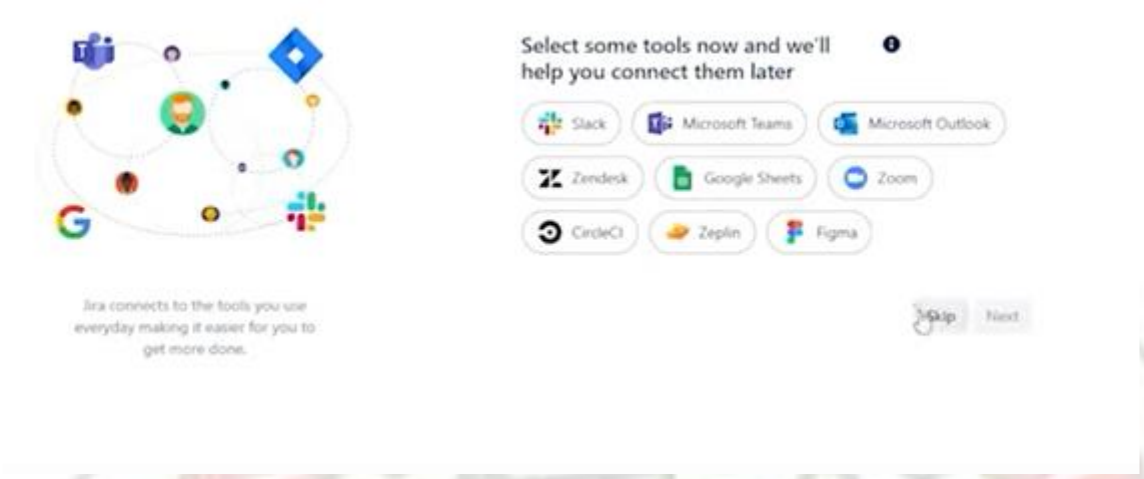
Type — Change type

Team-managed RECOMMENDED — Control your own working processes and practices in a self-contained space.

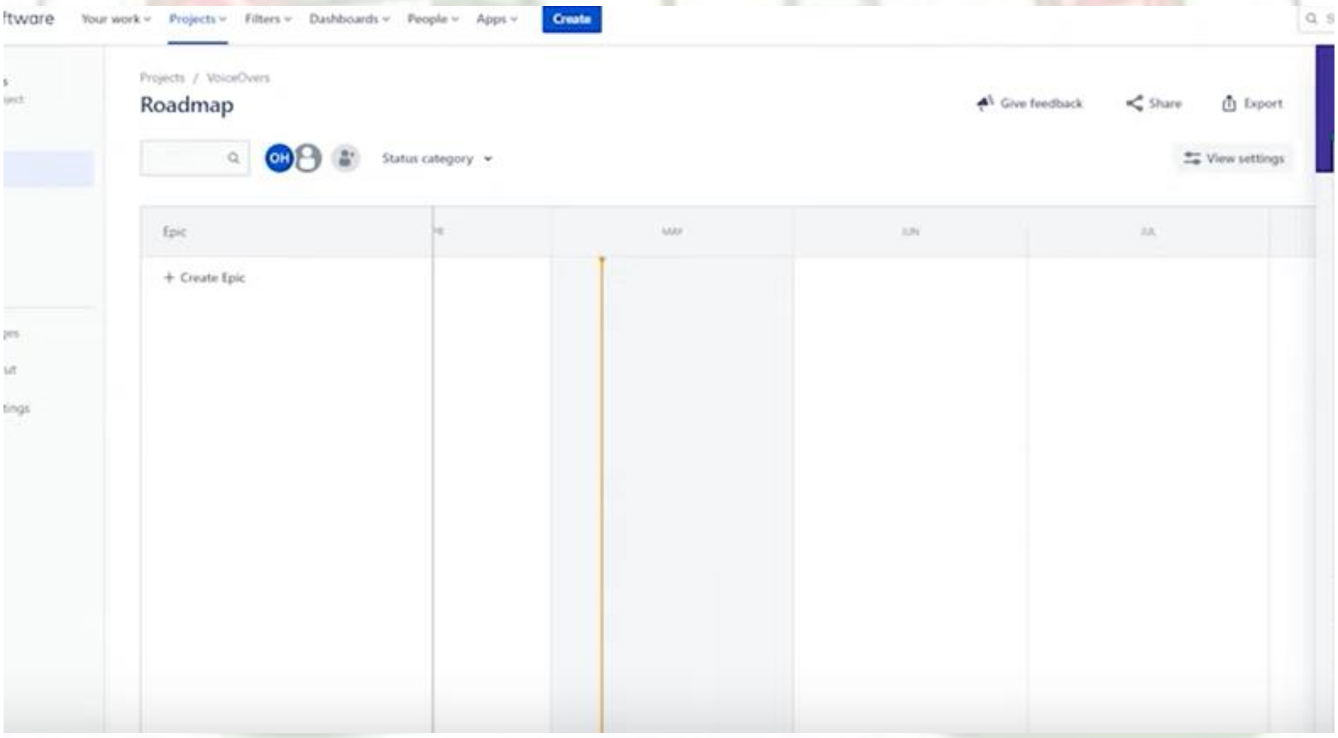
Back | Create project

Choose Template of Your wish (Kanban in this case)

STEP- 3: Optional (You can Add other tools too)

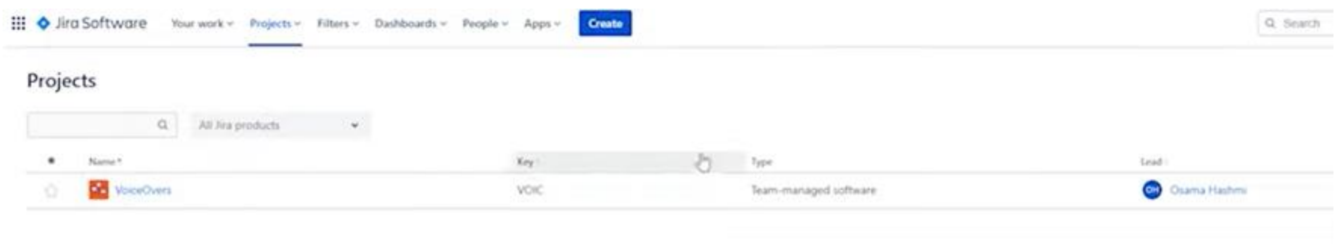

Select some tools now and we'll help you connect them later

STEP-4 : Invite Team Members



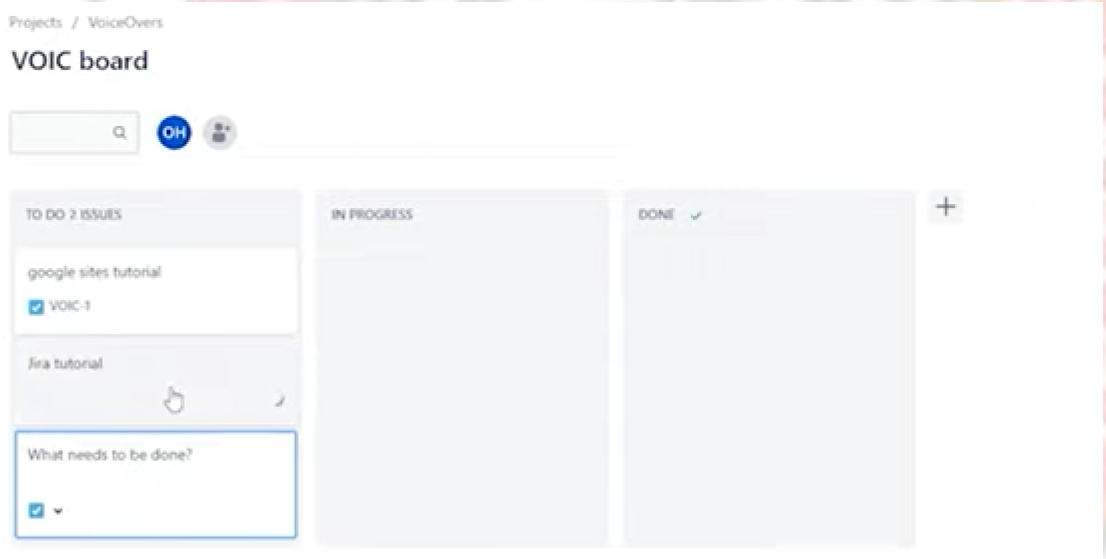
Dashboard of the JIRA

STEP-5: Create New Project

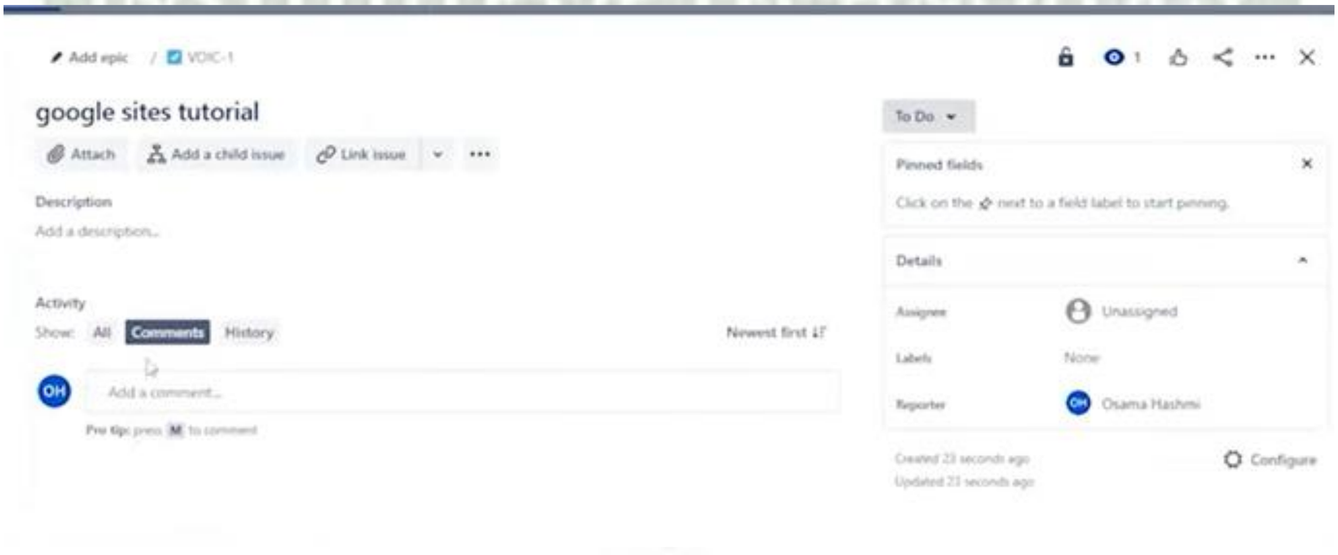


| Name | Key | Type | Lead |
|---|---|---|---|
| VoiceOvers | VOIC | Team-managed software | Osama Hashmi |

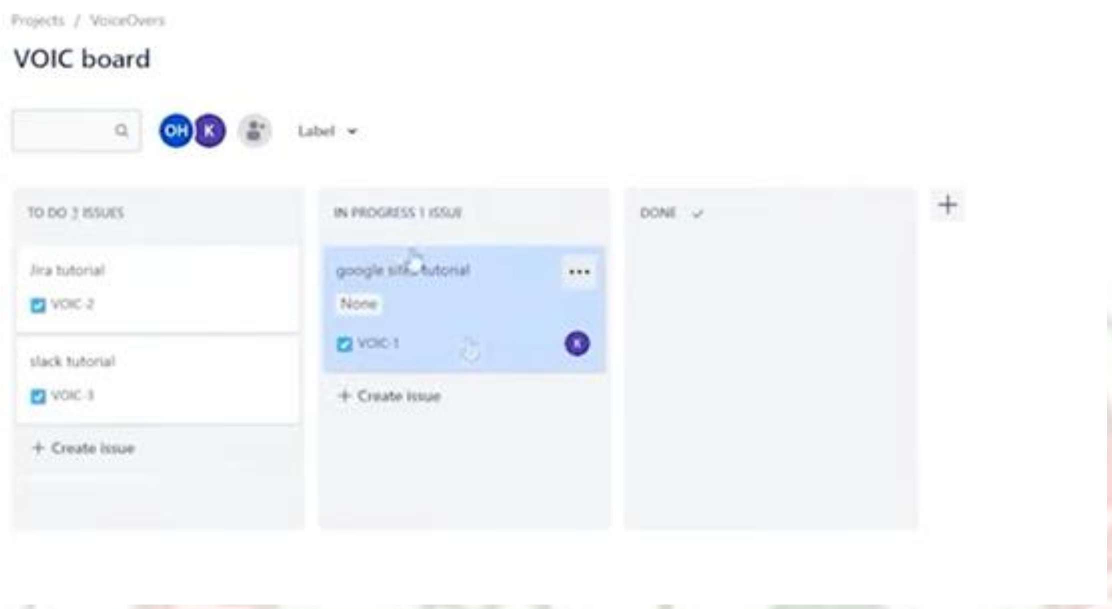
STEP-6: Create Boards



You can Assign People, Lables, Description too !

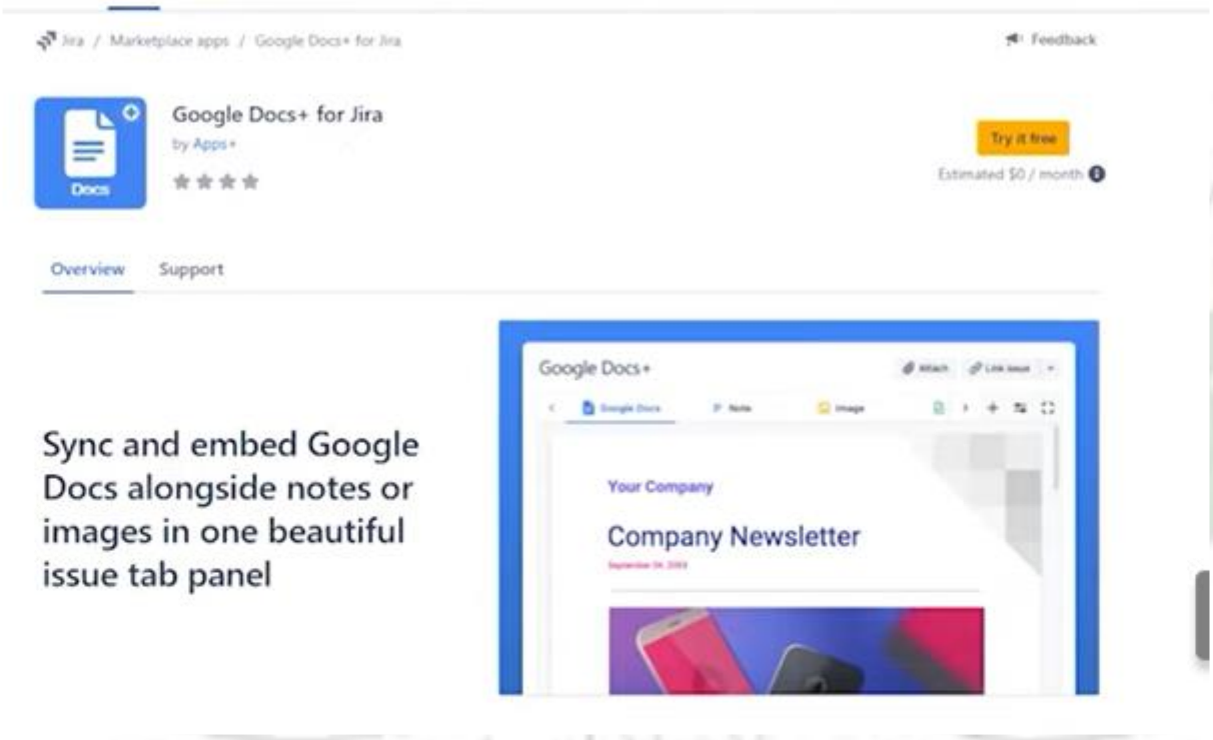


Once the Process in Progress. You can Shift it to IN PROGRESS Area



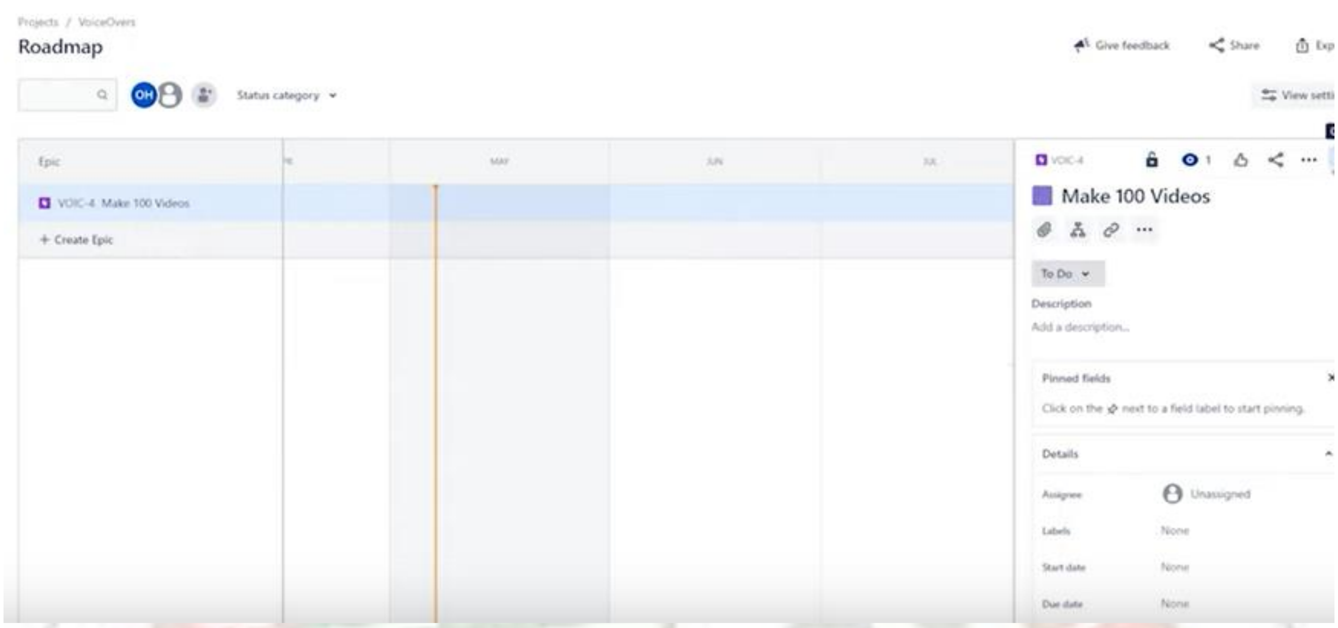
And if done change it to DONE.

STEP-7: Can add different integrations too


Jira / Marketplace apps / Google Docs+ for Jira
Feedback
Google Docs+ for Jira
by Apps+
Try it free
Estimated $0 / month
Overview
Support
Sync and embed Google Docs alongside notes or images in one beautiful issue tab panel
Google Docs+
Your Company
Company Newsletter

Step-8: Roadmap

Can add Epics and Deadlines


Projects / VoiceOvers

Roadmap

Give feedback · Share · Exp

Status category

View setti

| Epic | MAY | JUN | JUL |
|---|---|---|---|
| VOIC-4 Make 100 Videos | | | |
| + Create Epic | | | |

VOIC-4

Make 100 Videos

To Do

Description

Add a description...

Pinned fields

Click on the next to a field label to start pinning.

Details

| | |
|---|---|
| Assignee | Unassigned |
| Labels | None |
| Start date | None |
| Due date | None |