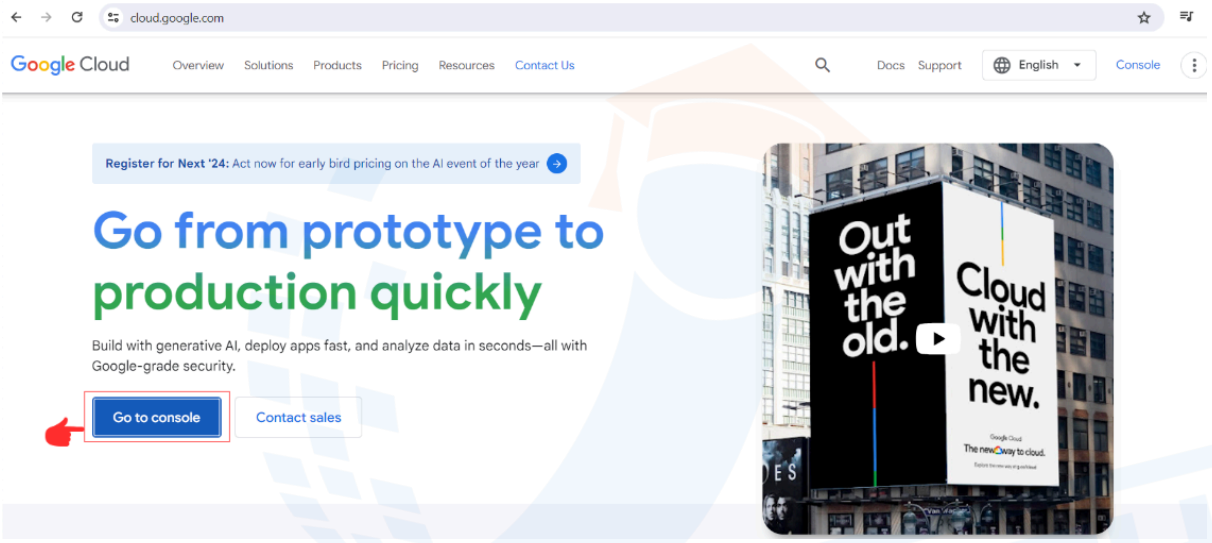
Deploying and Configuring Jenkins Server

(Usually these activities are performed by DevOps Engineers)

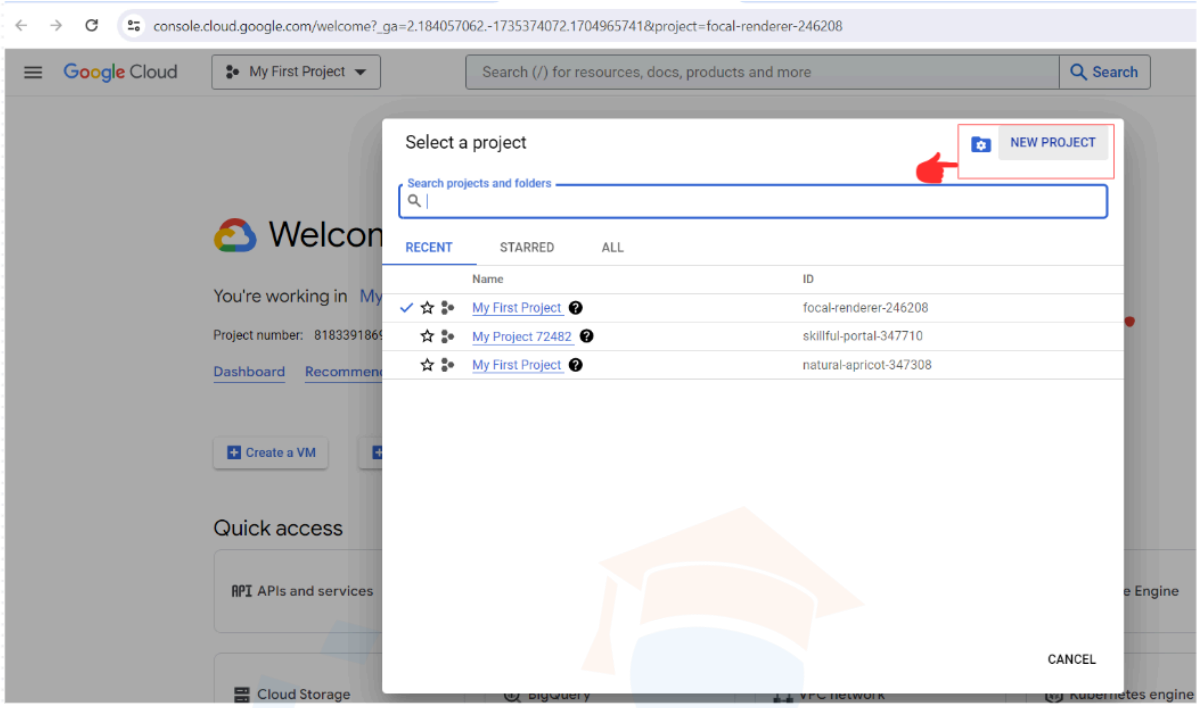
Go to cloud.google.com

Create a GCP account (provides 300$ free credits that would be valid for 3 months)

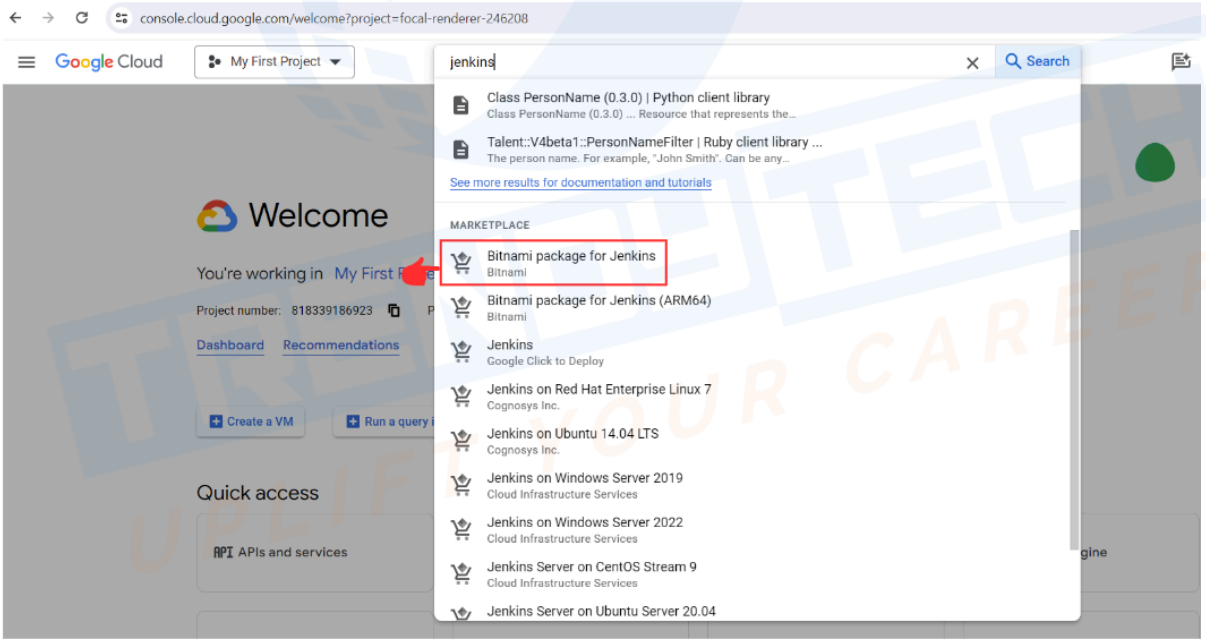
Once you create an account -> Go to Console

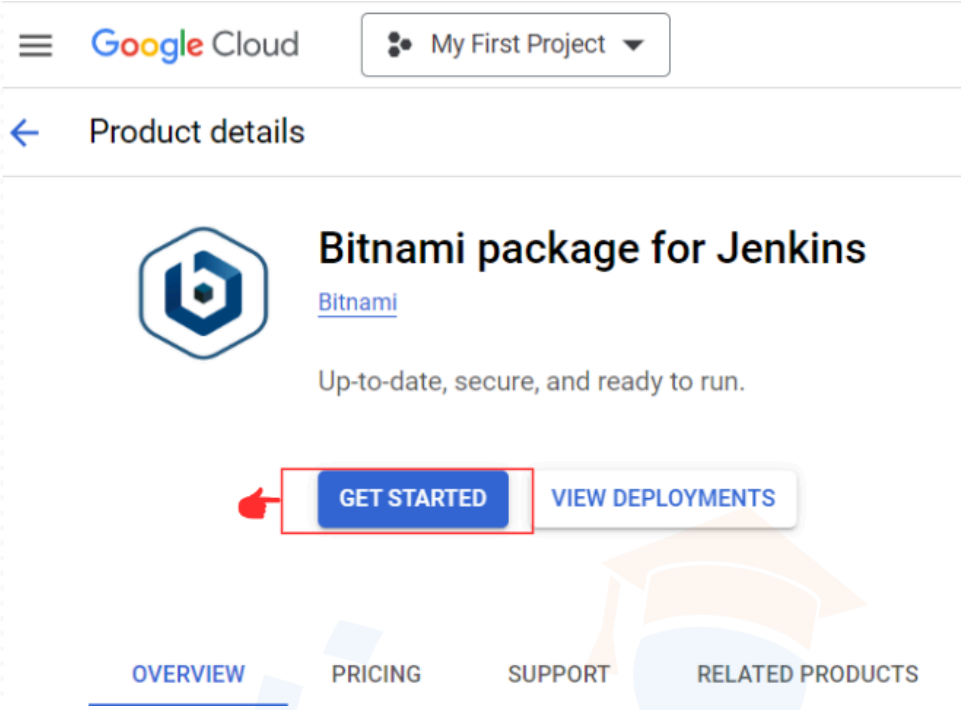


Create a new Project and Deploy Resources (Like - Jenkins)



Deploy the necessary resources - Like Jenkins

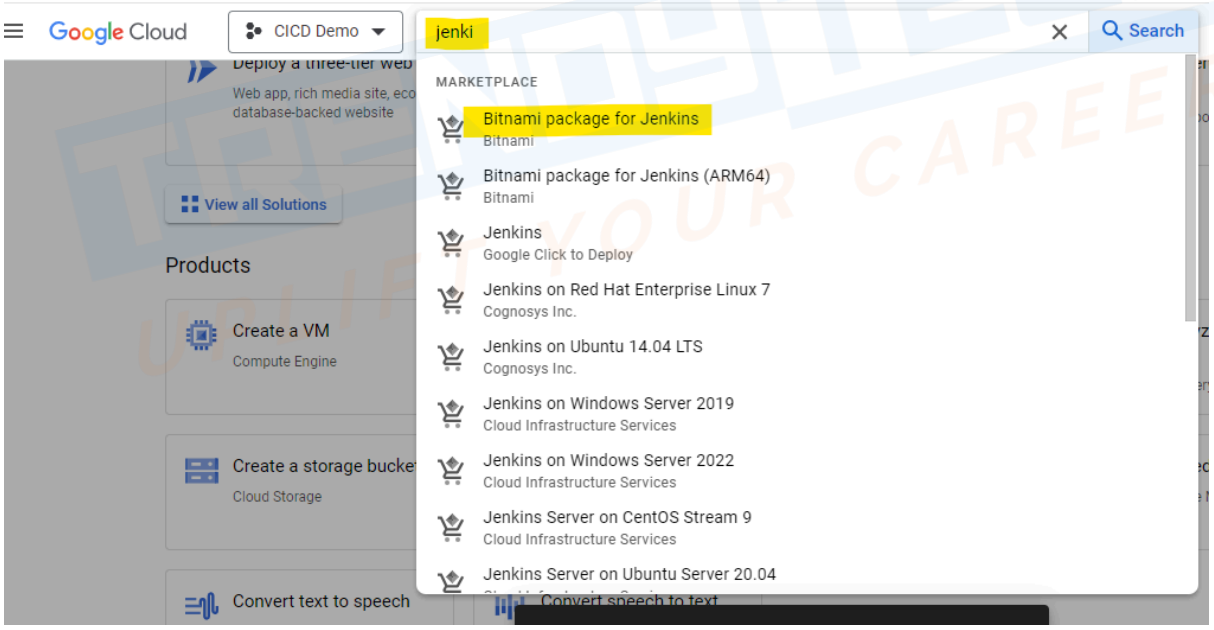


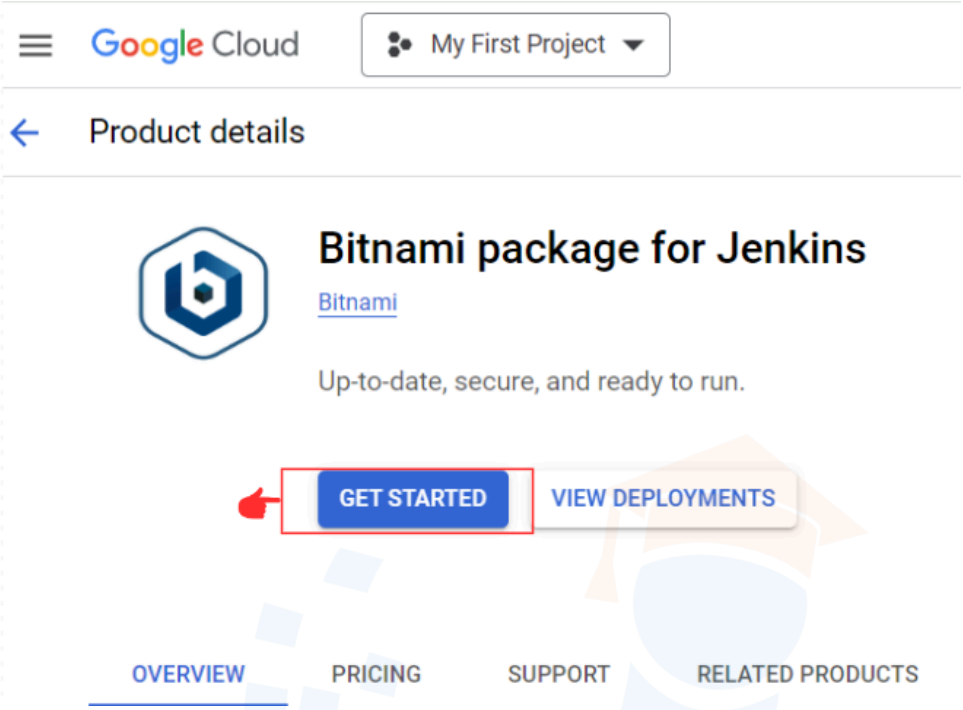


Deploy the necessary resources - Like Jenkins:

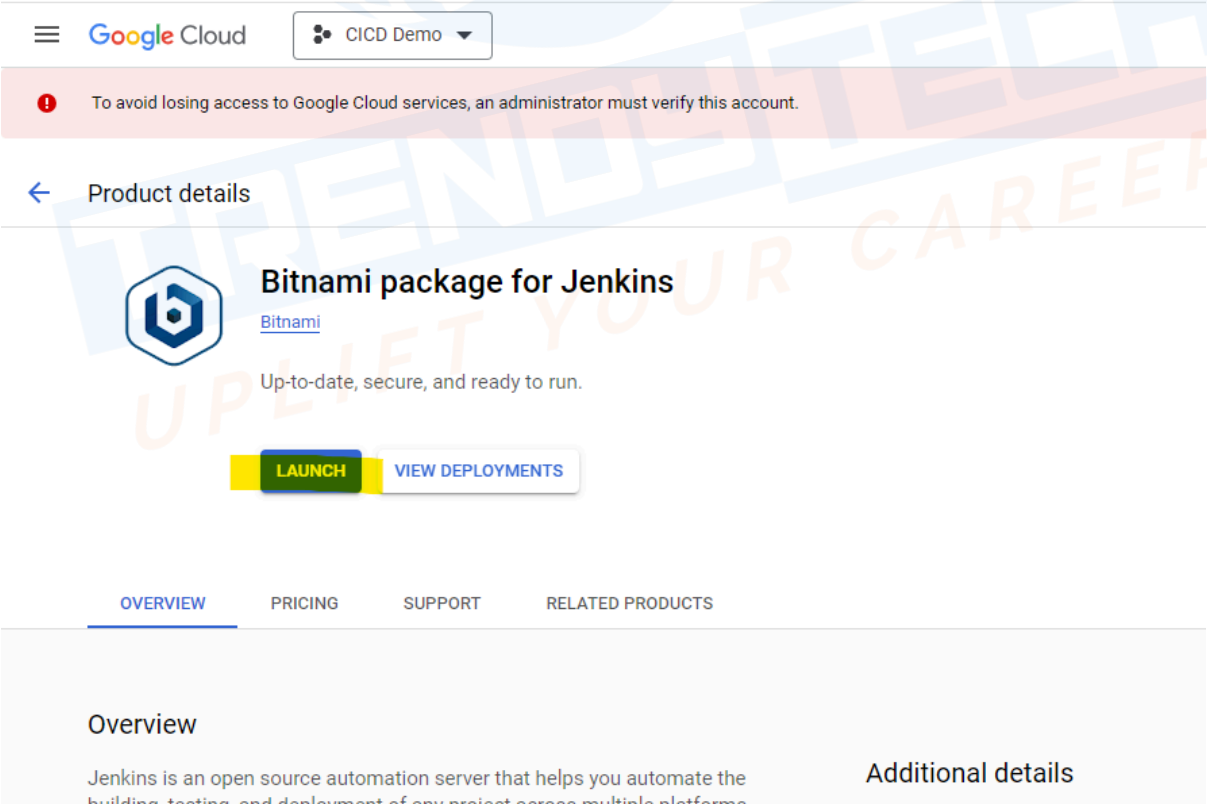
To deploy the jenkins follow below steps:

Search for **"jenkin"** in and in the marketplace you will get the option of **"Bitnami package for Jenkins"** refer attached screenshot.

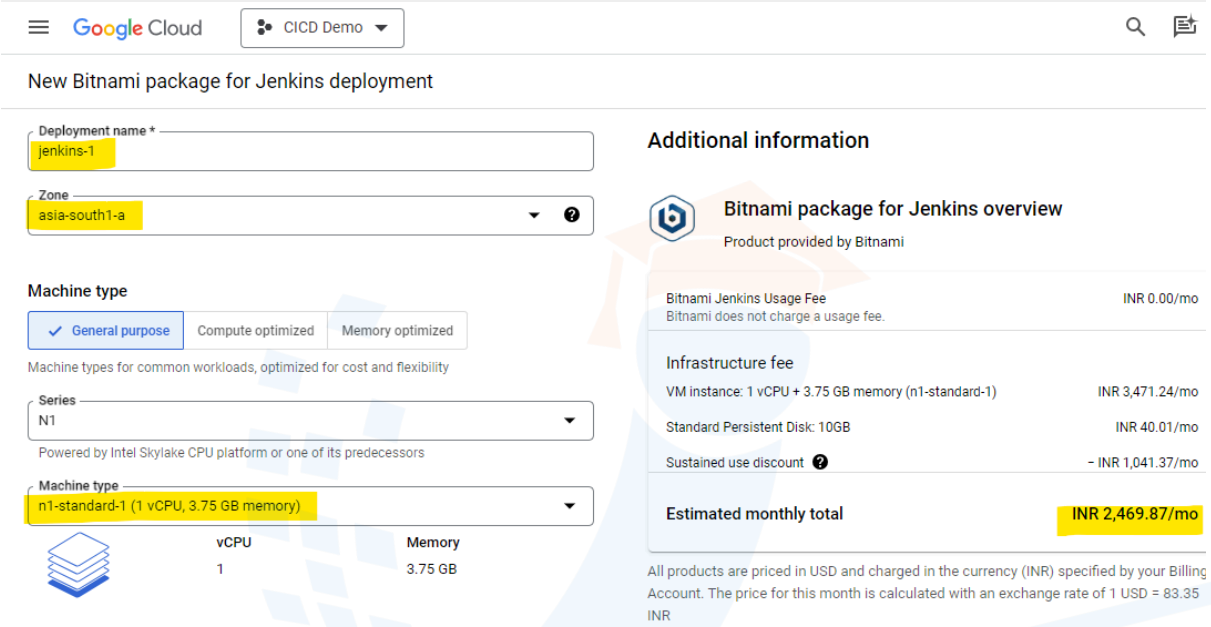




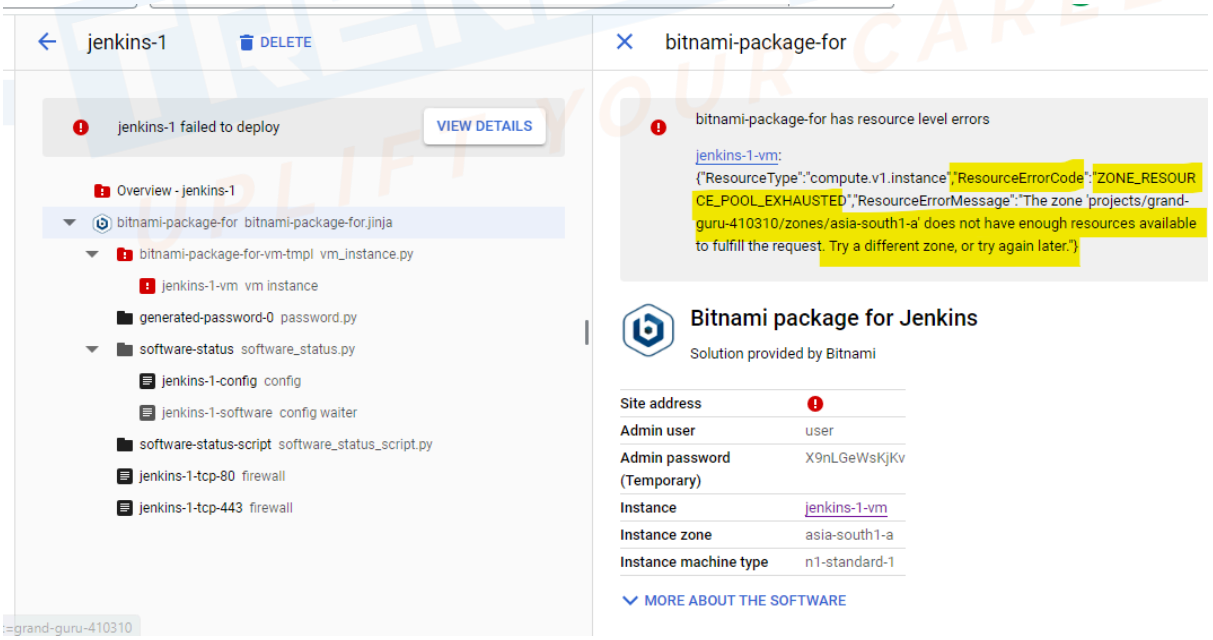
Allow the installation that it will ask for. Then you will get the option to launch the package as shown in the screenshot below.



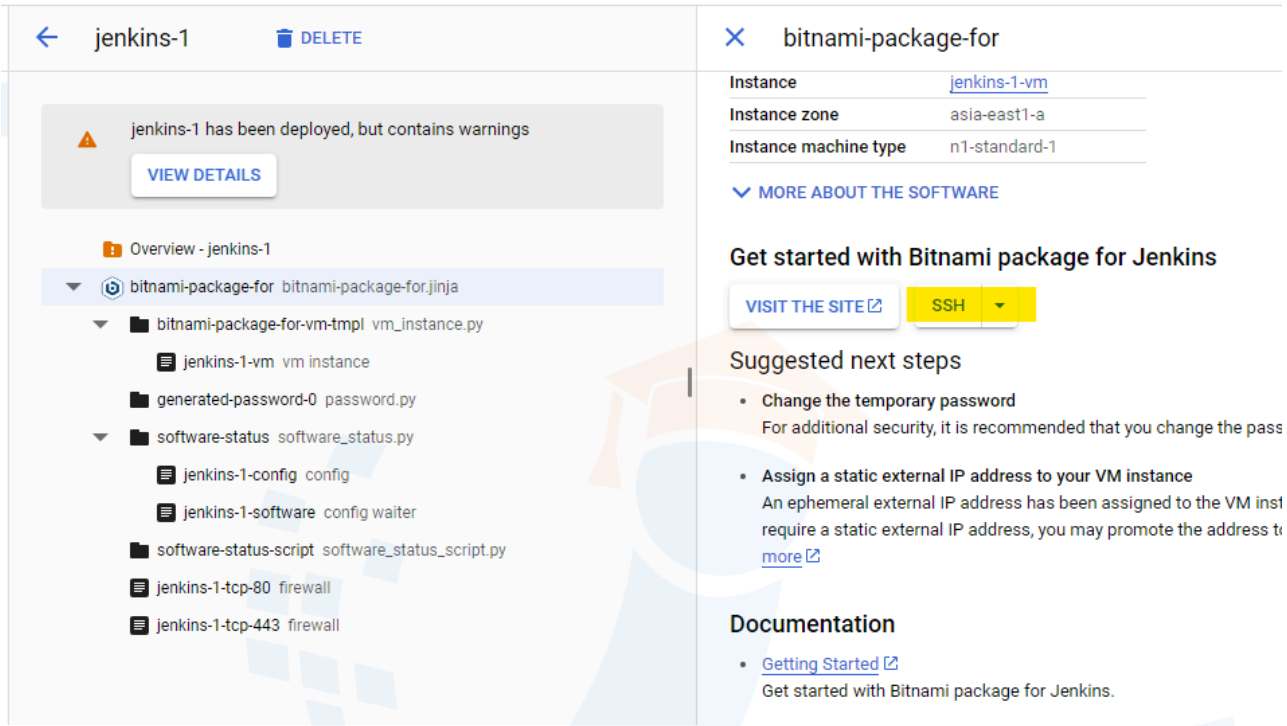
For package deployment, specify the zone and machine type as highlighted in the screenshot. Leave other configurations unchanged, and proceed to click on "Deploy."



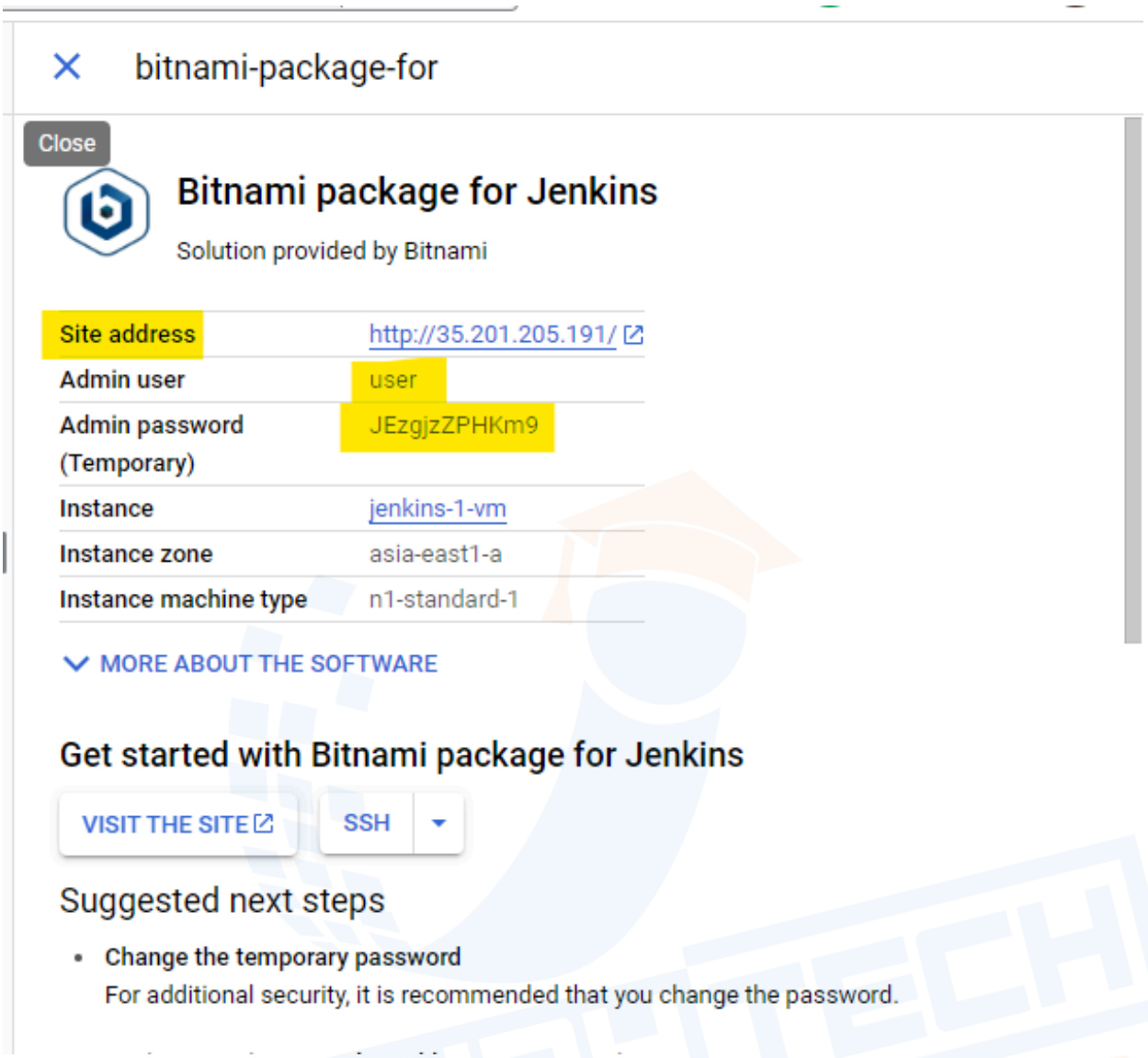
Note: If you face this issue please try to deploy it again with a different zone.



Once it is deployed you will get below screen. Now log into SSH.



Click on Site Address and username and password log into the site Refer attached screenshot.



Using the following commands
“sudo apt-get install pip”,
“sudo apt-get install sshpass“
install pip and sshpass respectively in SSH.

SSH-in-browser UPLOAD FILE

```
Linux jenkins-1-vm 5.10.0-26-cloud-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x8

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
        ___ _ _                    _
       | _ |_) |_ _ _  __ _ _ __ (_)
       | _ \ |  _| ' \/ _` | '  \| |
       |___/_|\__|_|_|\__,_|_|_|_|_|

  *** Welcome to the Bitnami package for Jenkins 2.426.2-0          ***
  *** Documentation:  https://docs.bitnami.com/google/apps/jenkins/ ***
  ***                 https://docs.bitnami.com/google/              ***
  *** Bitnami Forums: https://github.com/bitnami/vms/               ***
aratishatti15@jenkins-1-vm:~$ python --version
Python 2.7.18
aratishatti15@jenkins-1-vm:~$ python3 --version
Python 3.9.2
aratishatti15@jenkins-1-vm:~$ sudo apt-get install pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'python3-pip' instead of 'pip'
The following packages were automatically installed and are no longer required:
  libevent-2.1-7 libgnutls-dane0 libunbound8
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore l
```

SSH-in-browser UPLOAD FILE DOWNLOAD

```
Preparing to unpack .../14-python3-pip_20.3.4-4+deb11u1_all.deb ...
Unpacking python3-pip (20.3.4-4+deb11u1) ...
Setting up javascript-common (11+nmu1) ...
Setting up python3-wheel (0.34.2-1) ...
Setting up libexpat1-dev:amd64 (2.2.10-2+deb11u5) ...
Setting up python-pip-whl (20.3.4-4+deb11u1) ...
Setting up libjs-jquery (3.5.1+dfsg+~3.5.5-7) ...
Setting up python3-lib2to3 (3.9.2-1) ...
Setting up libjs-underscore (1.9.1~dfsg-3) ...
Setting up python3-distutils (3.9.2-1) ...
Setting up python3-setuptools (52.0.0-4) ...
Setting up libpython3.9-dev:amd64 (3.9.2-1) ...
Setting up python3-pip (20.3.4-4+deb11u1) ...
Setting up libjs-sphinxdoc (3.4.3-2) ...
Setting up python3.9-dev (3.9.2-1) ...
Setting up libpython3-dev:amd64 (3.9.2-3) ...
Setting up python3-dev (3.9.2-3) ...
Processing triggers for man-db (2.9.4-2) ...
aratishatti15@jenkins-1-vm:~$ sudo apt-get install sshpass
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libevent-2.1-7 libgnutls-dane0 libunbound8
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  sshpass
```

In UI, install the following plugins

Dashboard view -

This plugin allows you to create a customized dashboard view, providing a summary of information from various jobs or builds. It helps in creating a visual representation of the overall build health and status.

Github branch Source -

The GitHub plugin in Jenkins enables integration with GitHub repositories. It allows Jenkins to trigger builds automatically when changes are pushed to a GitHub repository, and it provides the ability to specify branches for building.

pipeline declarative -

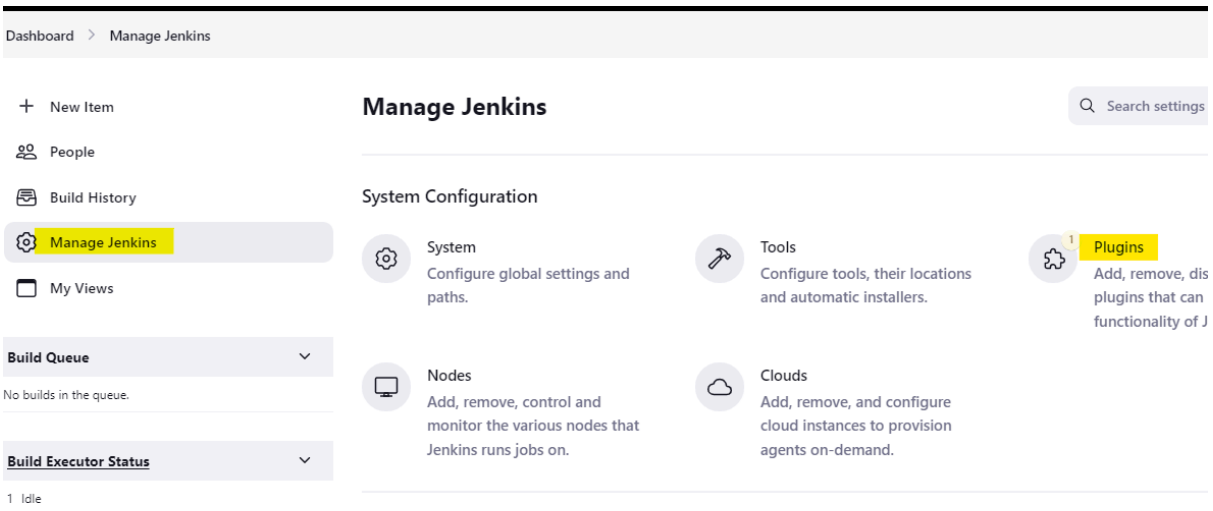
Declarative Pipeline is a feature within the Pipeline plugin that provides a simplified and opinionated syntax for defining build pipelines. It allows you to express pipelines in a more structured and readable format.

pipeline stage view -

The Stage View plugin enhances the visualization of Jenkins Pipelines. It provides a graphical representation of the stages in your pipeline, showing the progress and status of each stage.

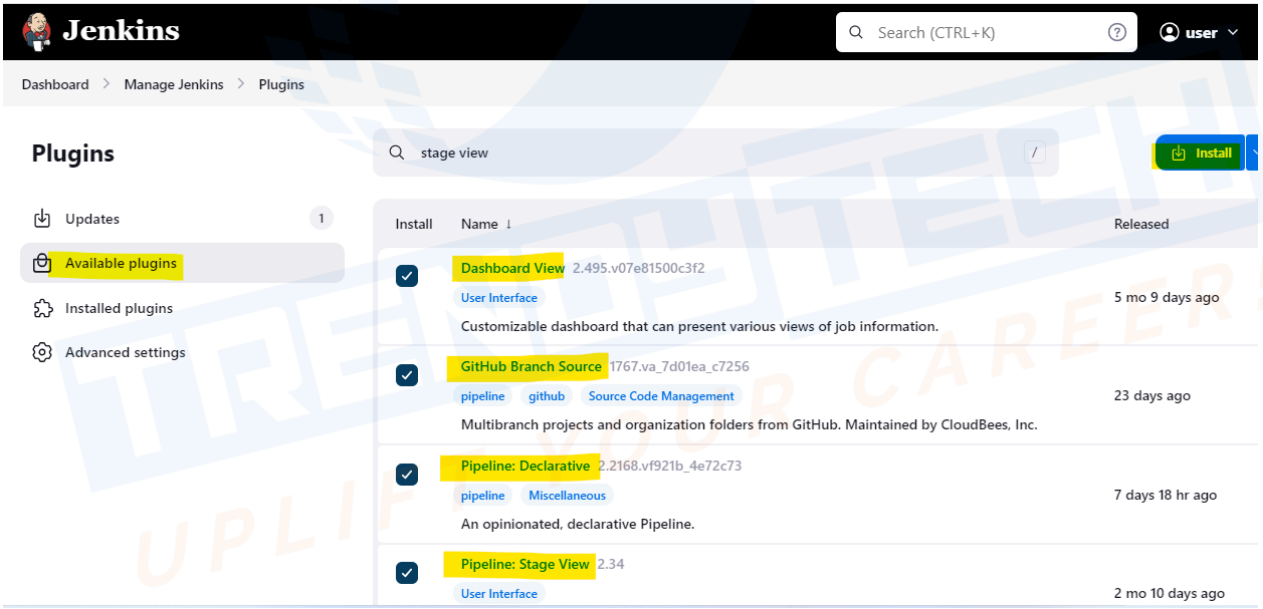
To install the plugins follow below steps:

In dashboard select “manage jenkins” => Plugins



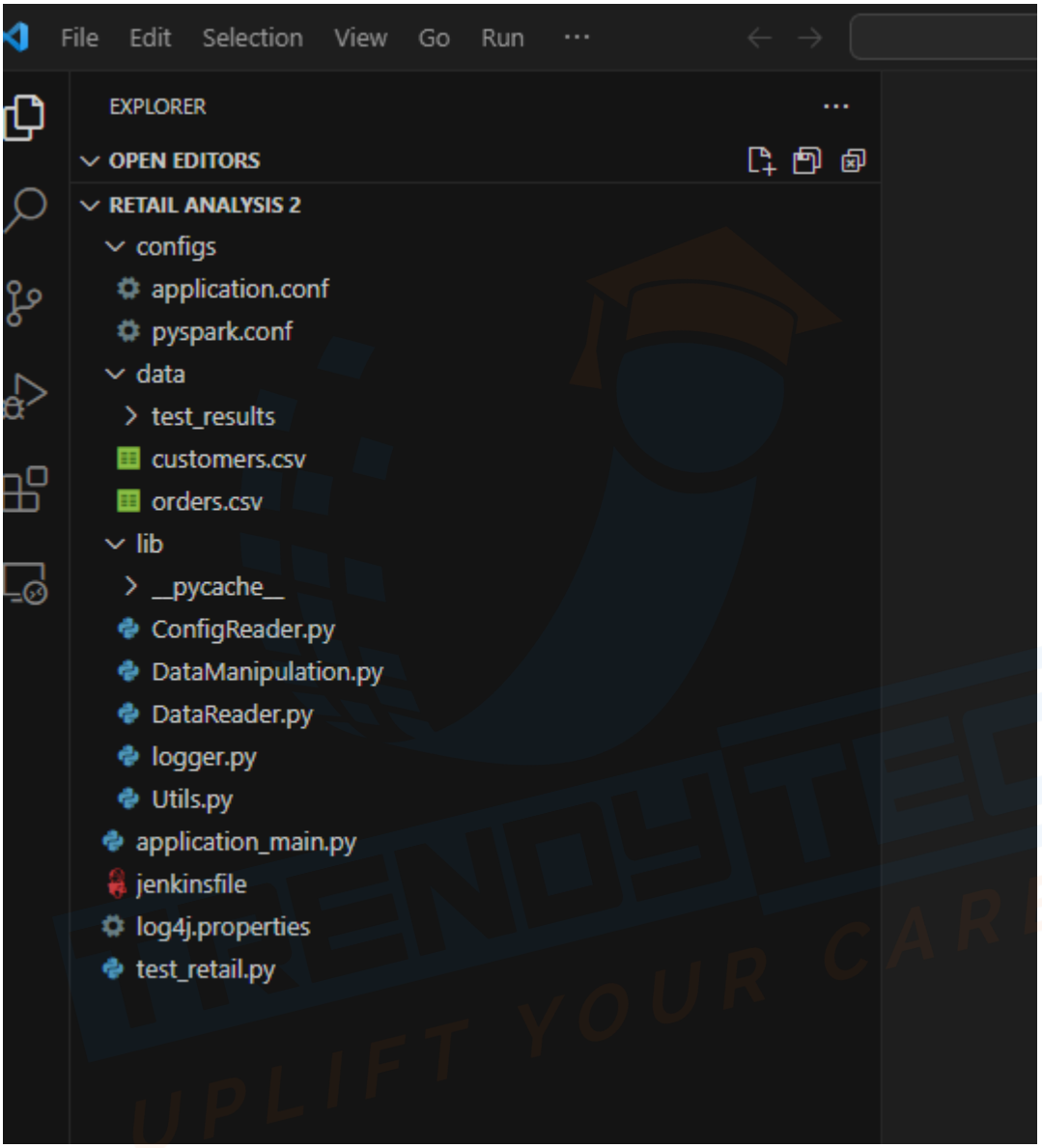
Now, go to **Available plugins => select mentioned plugin => click on install**

Refer attached screenshot.



Steps for creating a project Retail Analysis, initialising Git, and establishing branches on GitHub are as follows:

Create a project Retail Analysis in VS code if you do not have



Now **create virtual environment** in it using command : **“pipenv install”**

```
C:\Users\HP\Desktop\Retail Analysis 2>pipenv install
Creating a virtualenv for this project...
Pipfile: C:\Users\HP\Desktop\Retail Analysis 2\Pipfile
Using c:/users/hp/.pyenv/pyenv-win/versions/3.9.2/python3.exe (3.9.2) to create virtualenv...
[=== ] Creating virtual environment...created virtual environment CPython3.9.2.final.0-64 in 10680ms
  creator CPython3Windows(dest=C:\Users\HP\.virtualenvs\Retail_Analysis_2-GE1CUWNa, clear=False, no_vcs_i
gnore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=
C:\Users\HP\AppData\Local\pypa\virtualenv)
    added seed packages: pip==23.3.2, setuptools==69.0.3, wheel==0.42.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActiva
tor

Successfully created virtual environment!
Virtualenv location: C:\Users\HP\.virtualenvs\Retail_Analysis_2-GE1CUWNa
Pipfile.lock (16c839) out of date, updating to (71871d)...
Locking [packages] dependencies...
Locking [dev-packages] dependencies...
Updated Pipfile.lock (467be4a3aca30b5feaeb72c4d10f547c063e1c02754d633659e116ca9371871d)!
Installing dependencies from Pipfile.lock (71871d)...
To activate this project's virtualenv, run pipenv shell.
```

Note : It will create pipfile and pipfile.lock

```
RETAIL ANALYSIS 2
  configs
    application.conf
    pyspark.conf
  data
    test_results
    customers.csv
    orders.csv
  lib
    __pycache__
    ConfigReader.py
    DataManipulation.py
    DataReader.py
    logger.py
    Utils.py
  application_main.py
  jenkinsfile
  log4j.properties
  Pipfile
  Pipfile.lock
  test_retail.py
```

Also install **pytest** using command : **“pipenv install pytest”**

```
C:\Users\HP\Desktop\Retail Analysis 2>pipenv install pytest
Installing pytest...
Resolving pytest...
Added pytest to Pipfile's [packages] ...
Installation Succeeded
Pipfile.lock (71871d) out of date, updating to (f4b28f)...
Locking [packages] dependencies...
Building requirements...
Resolving dependencies...
Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (8c4501cc7de24ae9782093e8b563a1a434860751698b668c196074c8e5f4b28f)!
Installing dependencies from Pipfile.lock (f4b28f)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
```

In the virtual environment install the python version which is present in your jenkins.
Here it is 3.9.2 so creating virtual environment with python 3.9.2

Note: Before removing the virtual environment, first **exit** it using the **“exit()”** command.
Now you can remove it using the command you can **“pipenv –rm“ and recreate it** with the required **python version.**

```
Retail Analysis 2
Pipfile
Pipfile
[[source]]
url = "https://pypi.org/simple"
verify_ssl = true
name = "pypi"

[packages]
pytest = "*"

[dev-packages]

[requires]
python_version = "3.9.2"
```

If your existing project is not under version control, use **"git init"** to start tracking changes.

```
C:\Users\HP\Desktop\Retail Analysis 2>git init
Initialized empty Git repository in C:/Users/HP/Desktop/Retail Analysis 2/.git/
```

To **rename** branch **master to main** use command : **“git branch -M main”**

Using the commands “git add .” add all the changes

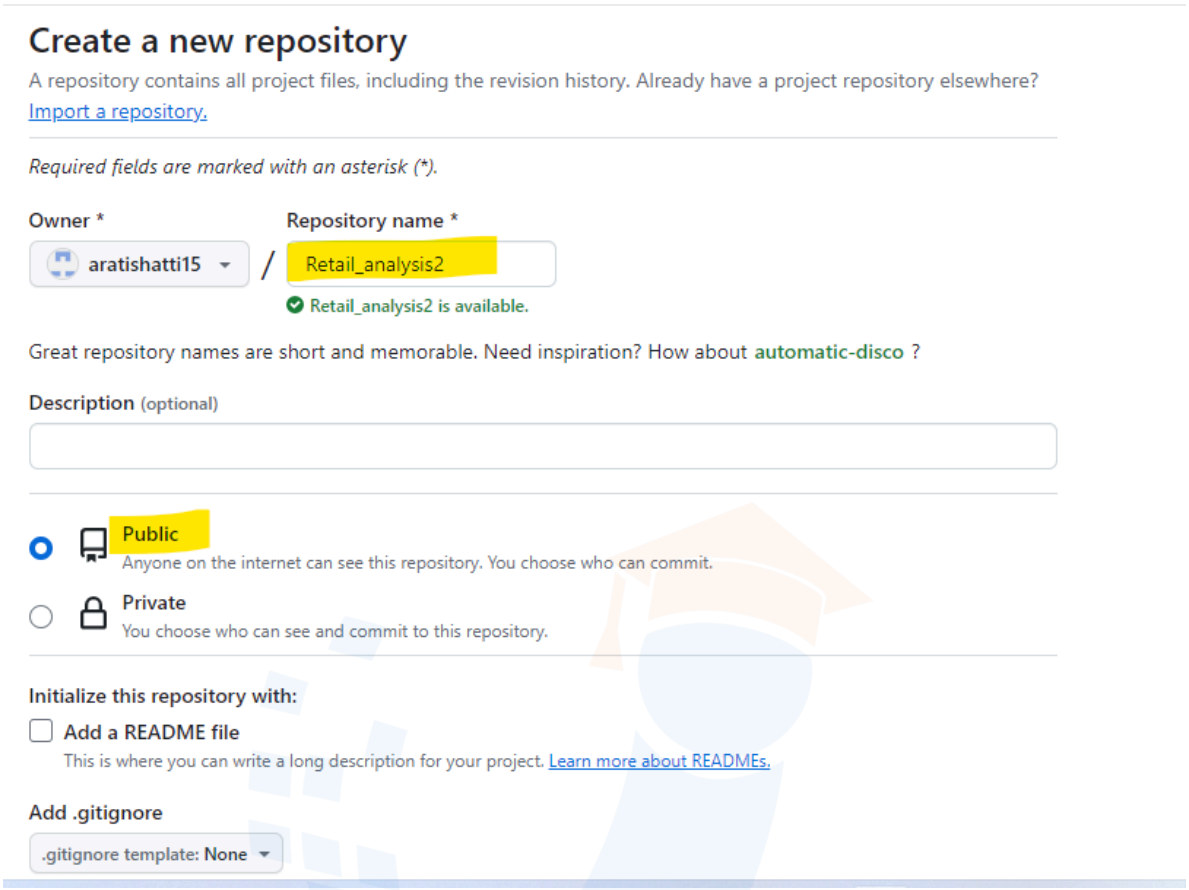
```
C:\Users\HP\Desktop\Retail Analysis 2>git add .
warning: in the working copy of 'Pipfile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Pipfile.lock', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'data/customers.csv', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'data/orders.csv', LF will be replaced by CRLF the next time Git touches it
```

And using command “git commit -m “performing initial commit”” commit all the changes

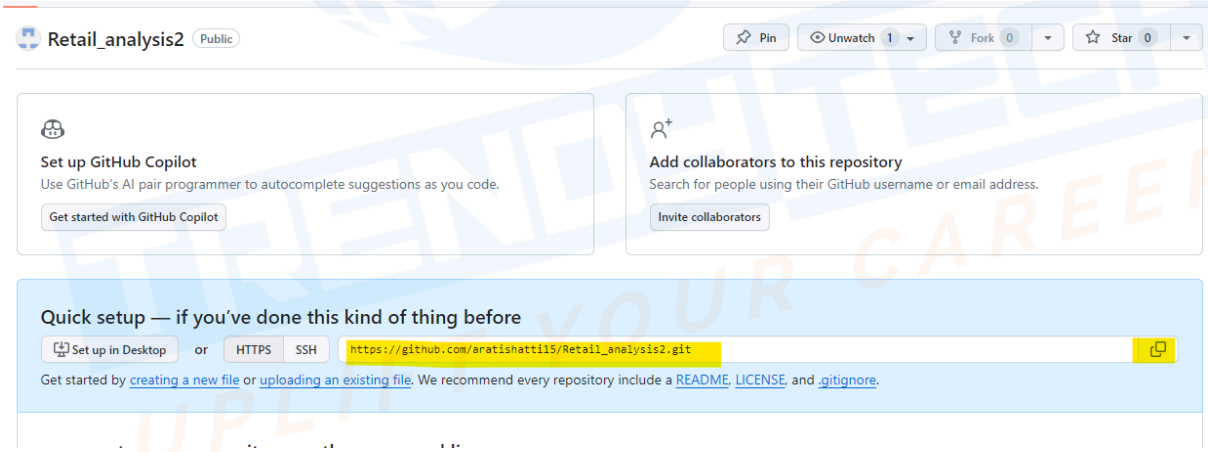
```
C:\Users\HP\Desktop\Retail Analysis 2>git commit -m "performing initial commit"
[main (root-commit) b499b8b] performing initial commit
 31 files changed, 81720 insertions(+)
 create mode 100644 Pipfile
 create mode 100644 Pipfile.lock
 create mode 100644 application_main.py
 create mode 100644 configs/application.conf
 create mode 100644 configs/pyspark.conf
 create mode 100644 data/customers.csv
 create mode 100644 data/orders.csv
 create mode 100644 data/test_results/state_aggregate.csv
 create mode 100644 jenkinsfile
 create mode 100644 lib/ConfigReader.py
```

Go to github and create a repo with the name Retail Project

Refer attached screenshot.



Also copy the ssh link as highlighted in the screenshot.



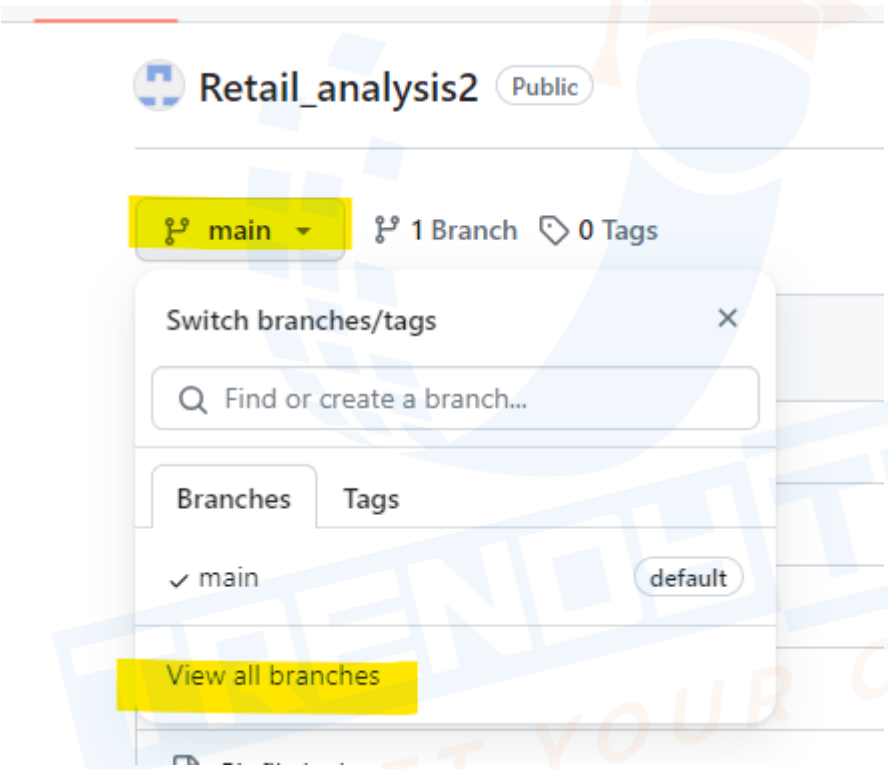
git remote add origin <url of project>

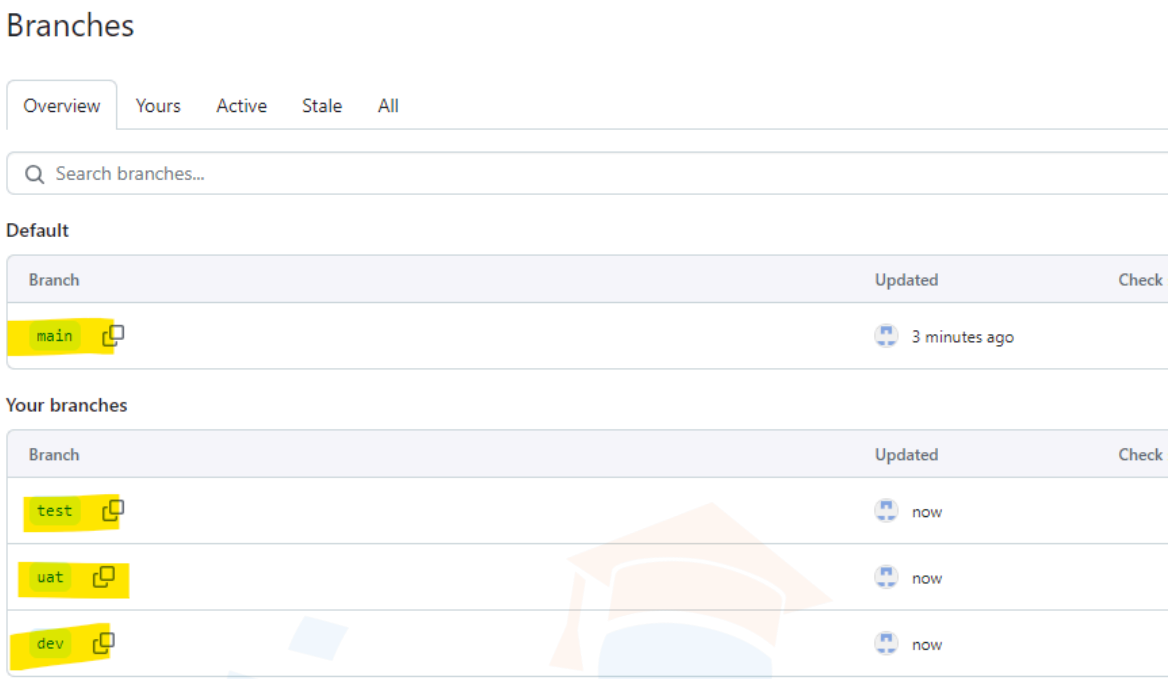
```
C:\Users\HP\Desktop\Retail Analysis 2>git remote add origin https://github.com/aratishatti15/Retail_analysis2.git
```

To push the code to the origin main run the command: **"git push origin main"**

```
C:\Users\HP\Desktop\Retail Analysis 2>git push origin main
Enumerating objects: 38, done.
Counting objects: 100% (38/38), done.
Delta compression using up to 2 threads
Compressing objects: 100% (36/36), done.
Writing objects: 100% (38/38), 715.50 KiB | 3.22 MiB/s, done.
Total 38 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
To https://github.com/aratishatti15/Retail_analysis2.git
 * [new branch]      main -> main
```

Click on view all branches and create the new branches : **main, dev, uat, test**





All have the same code

Create and switch to a new branch in a Git repository on your local machine using the command :

git checkout -b feature-rp-50001

```
C:\Users\HP\Desktop\Retail Analysis 2>git checkout -b feature-rp-50001
Switched to a new branch 'feature-rp-50001'
```

After making modifications, commit all the changes, and then push the changes to the remote repository using the command

git push origin feature-rp-50001

```
C:\Users\HP\Desktop\Retail Analysis 2>git add .

C:\Users\HP\Desktop\Retail Analysis 2>git commit -m "initial commit in feature"
On branch feature-rp-50001
nothing to commit, working tree clean

C:\Users\HP\Desktop\Retail Analysis 2>git push origin feature-rp-50001
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature-rp-50001' on GitHub by visiting:
remote:      https://github.com/aratishatti15/Retail_analysis2/pull/new/feature-rp-50001
remote:
To https://github.com/aratishatti15/Retail_analysis2.git
 * [new branch]      feature-rp-50001 -> feature-rp-50001
```

when you execute the above a new feature branch will be created in github

Configuring Jenkins to interpret GitHub events.

Making configurations in jenkins so that it can read the github events like branch creation, git push, pull request ..etc

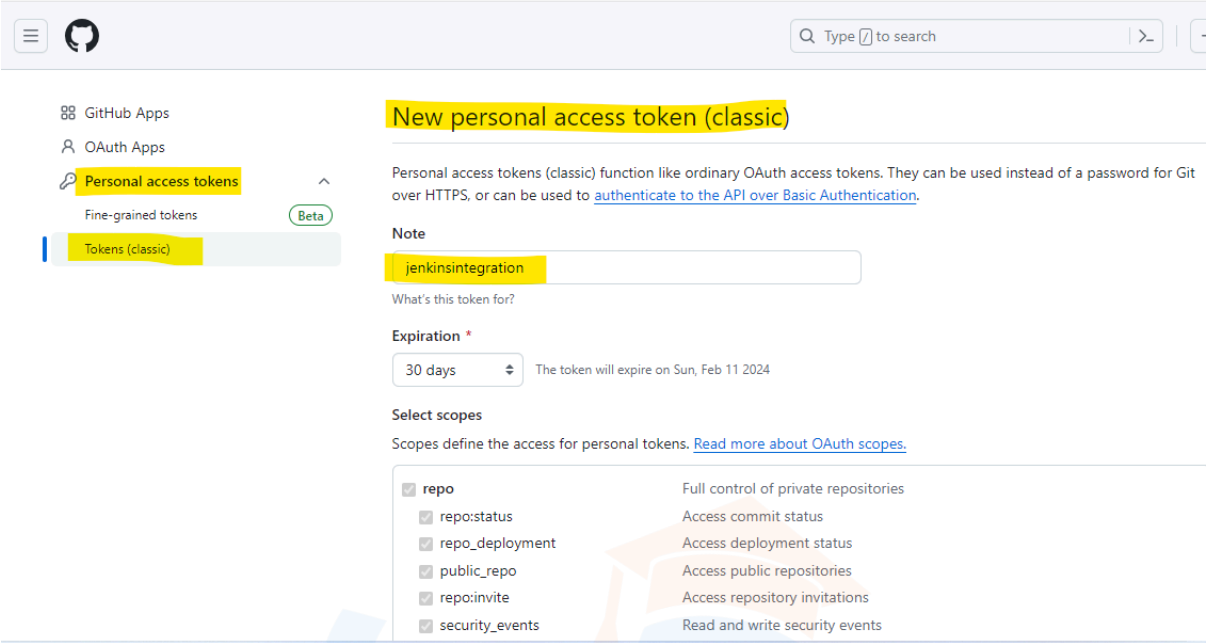
To establish the connection between github and jenkins, it has to be done from both ways.

Steps to follow in Jenkins:

Pre-step:

For creation of token in Github:

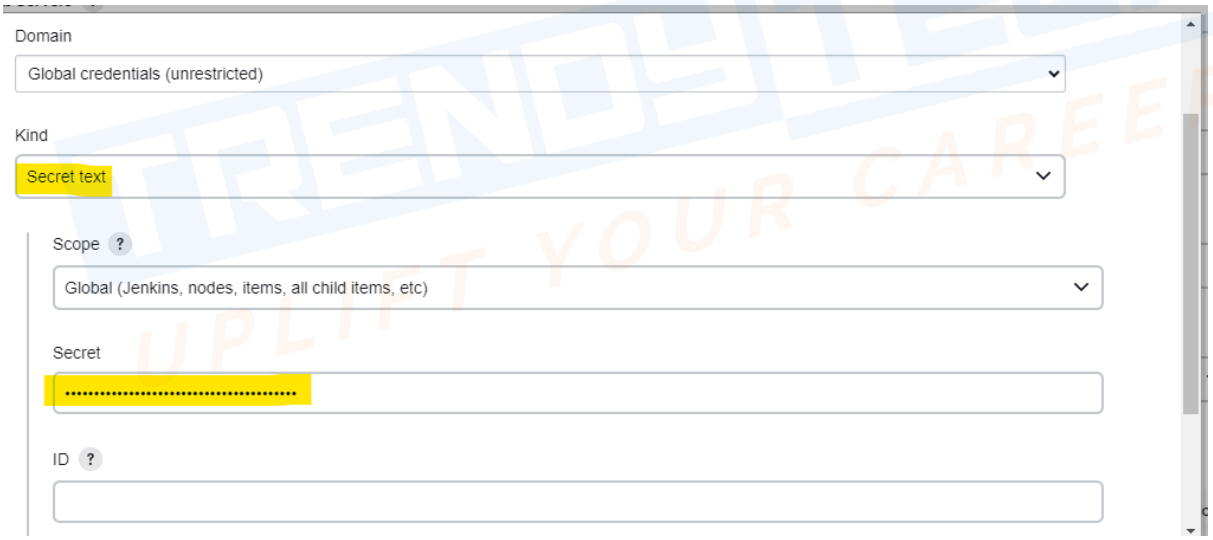
Go to Github => Settings => Developer Settings => Personal Access token => Tokens (Classic) => select all the scopes => generate token => Copy the token



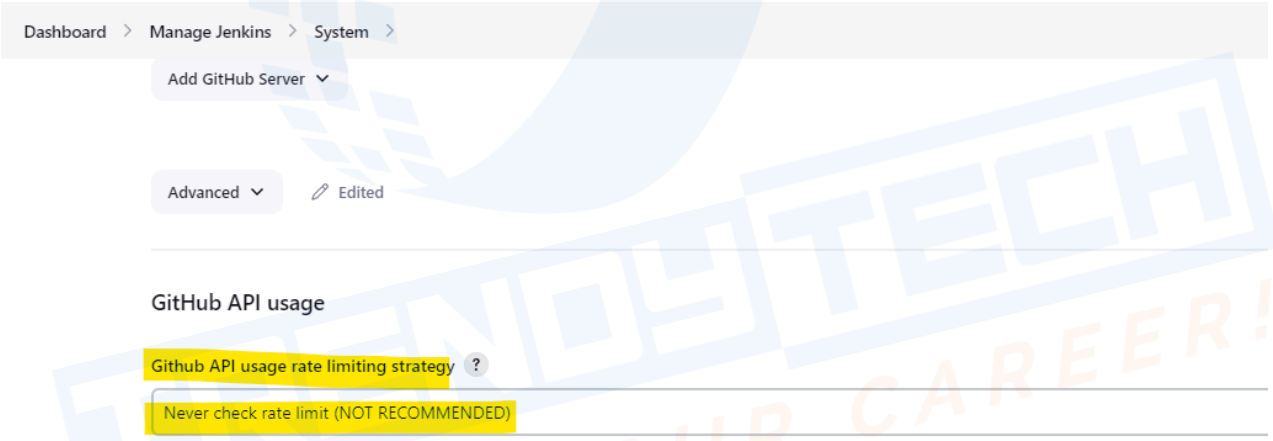
Go to Jenkins

Dashboard => Manage Jenkins => System

Mention the github token while creating secret text.



Also set Github Api usage limit strategy as shown below and then save it.

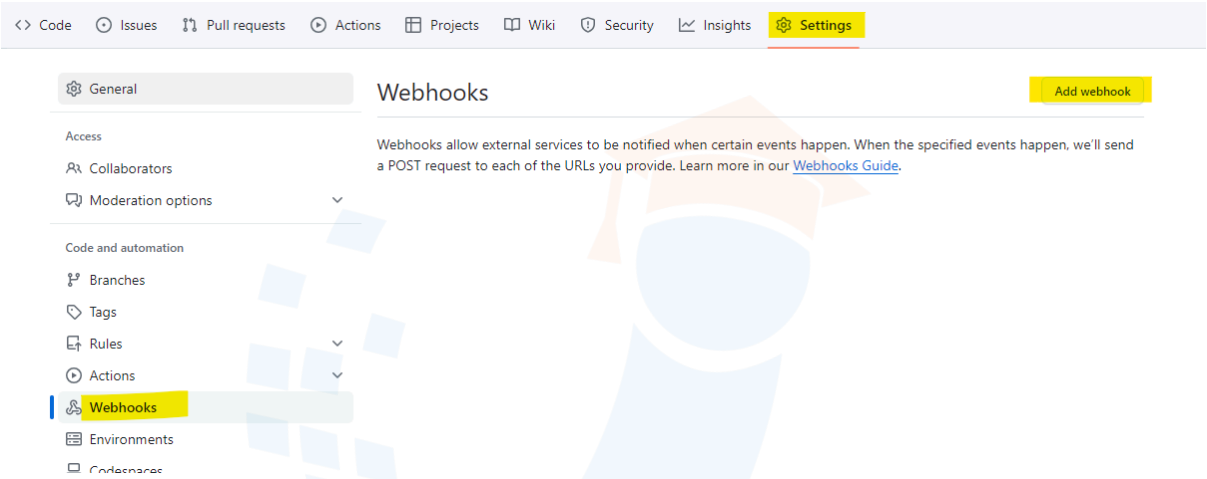


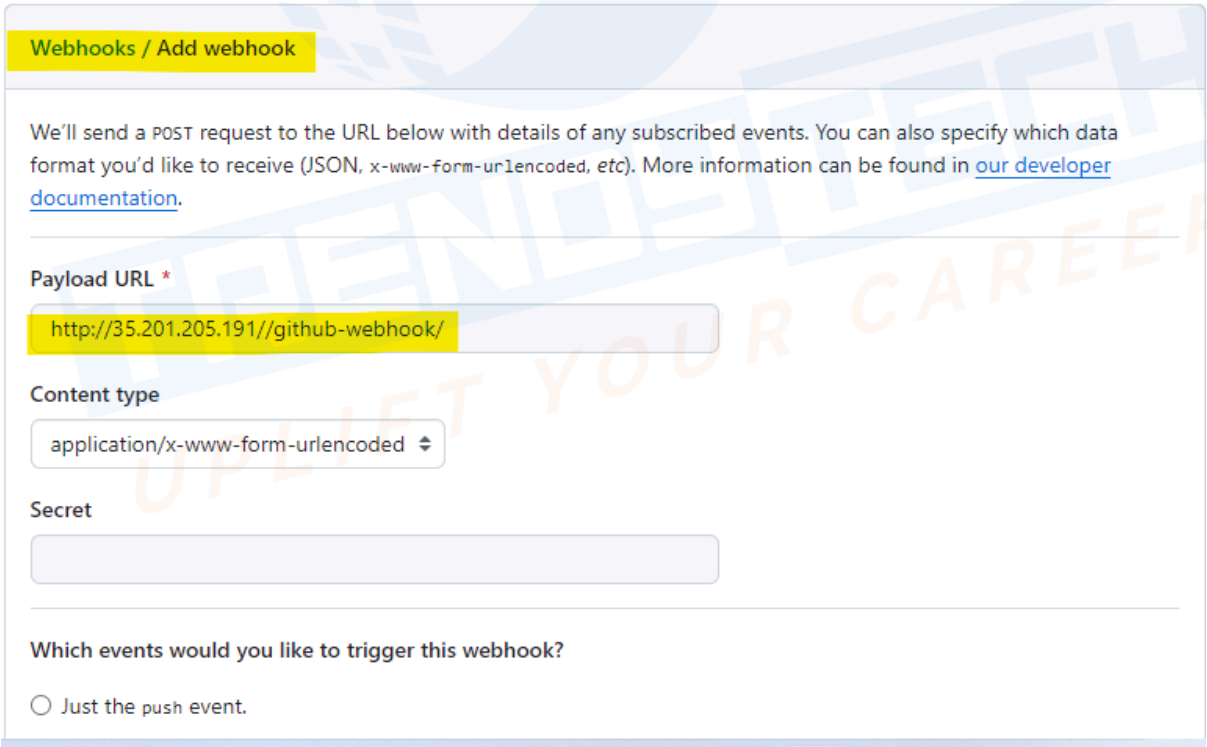
In Github

Go to the Repository for Retail Analysis => Setting => Webhooks

Now create a new webhook in which you have to mention jenkins URL and "/github-webhook/" as shown below.

<jenkinsurl>/github-webhook/

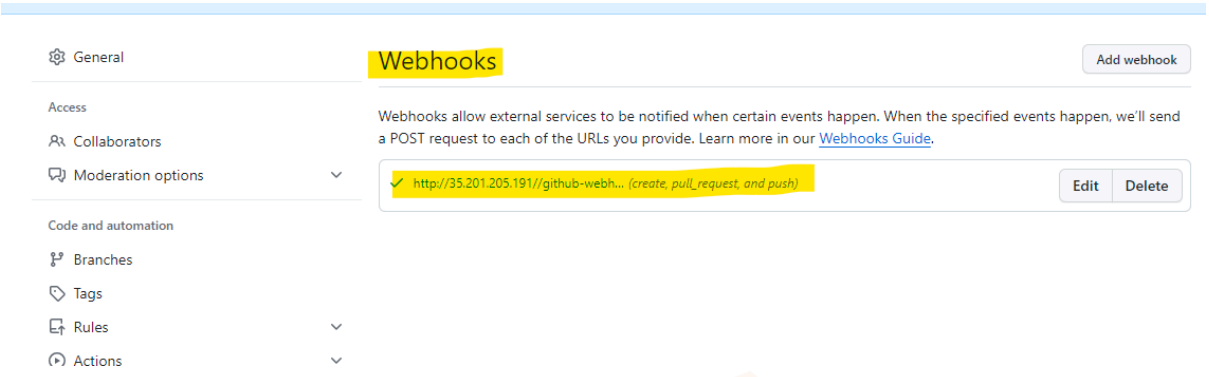




Which events would you like to trigger this webhook? => Let me select individual events.
=> Branch or tag creation, Pull requests, Pushes

Refer attached screenshot.

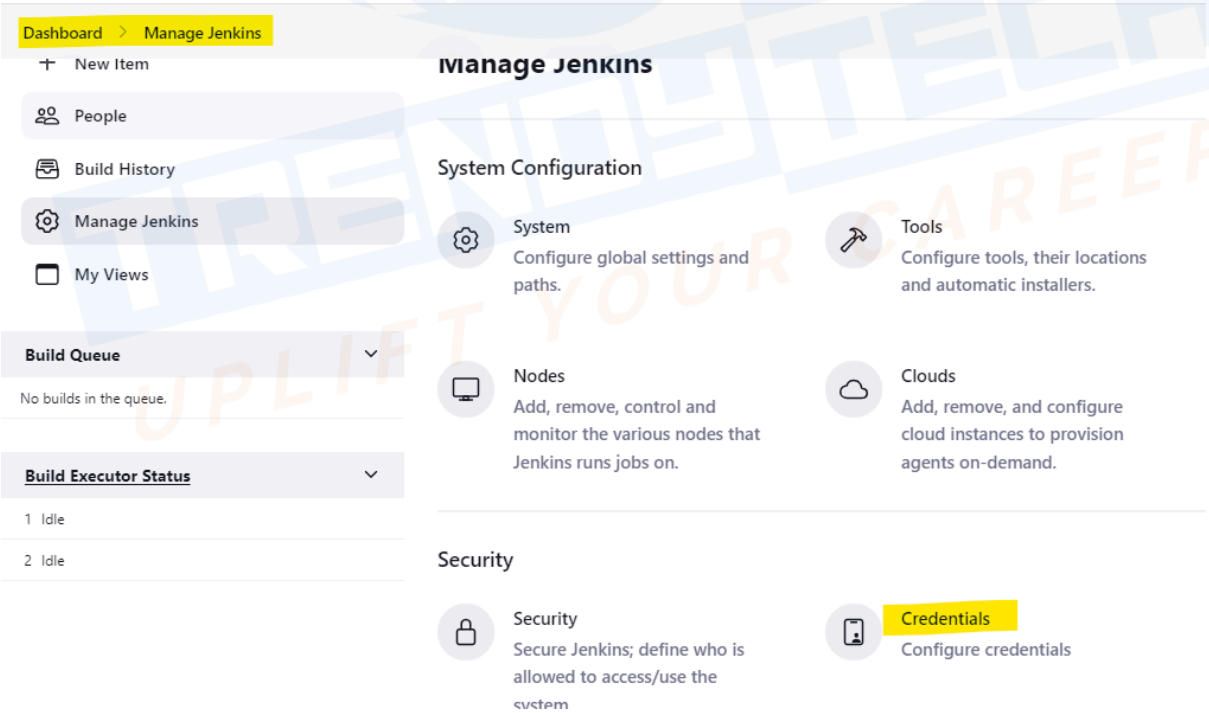
And click on “Add Webhook”. The webhook will be listed as shown below.

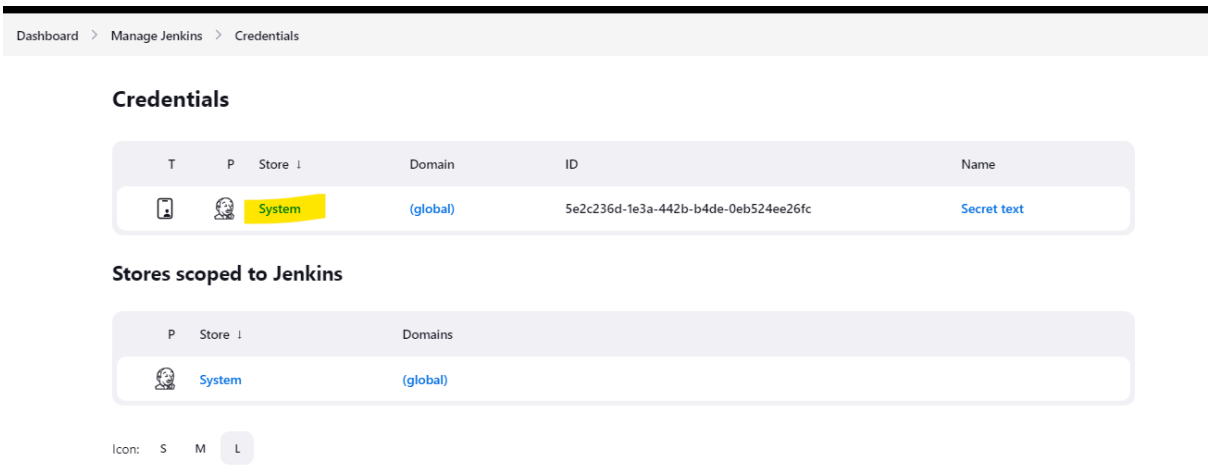


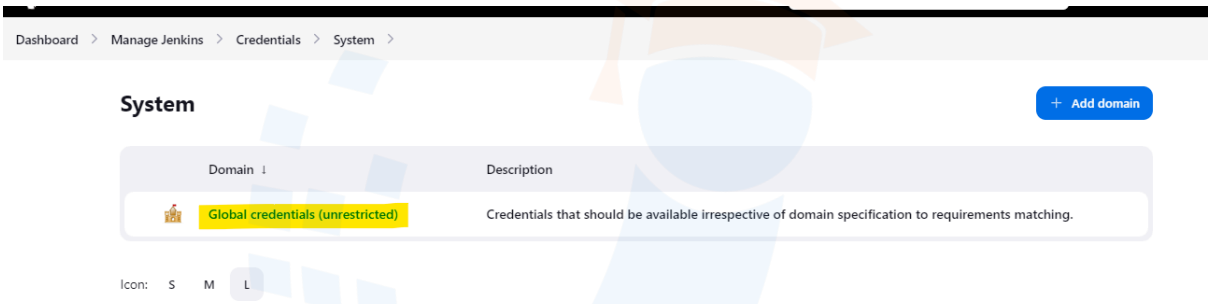
With these steps Both ways connectivity has been done.

Also set the credentials for your multi lab cluster.

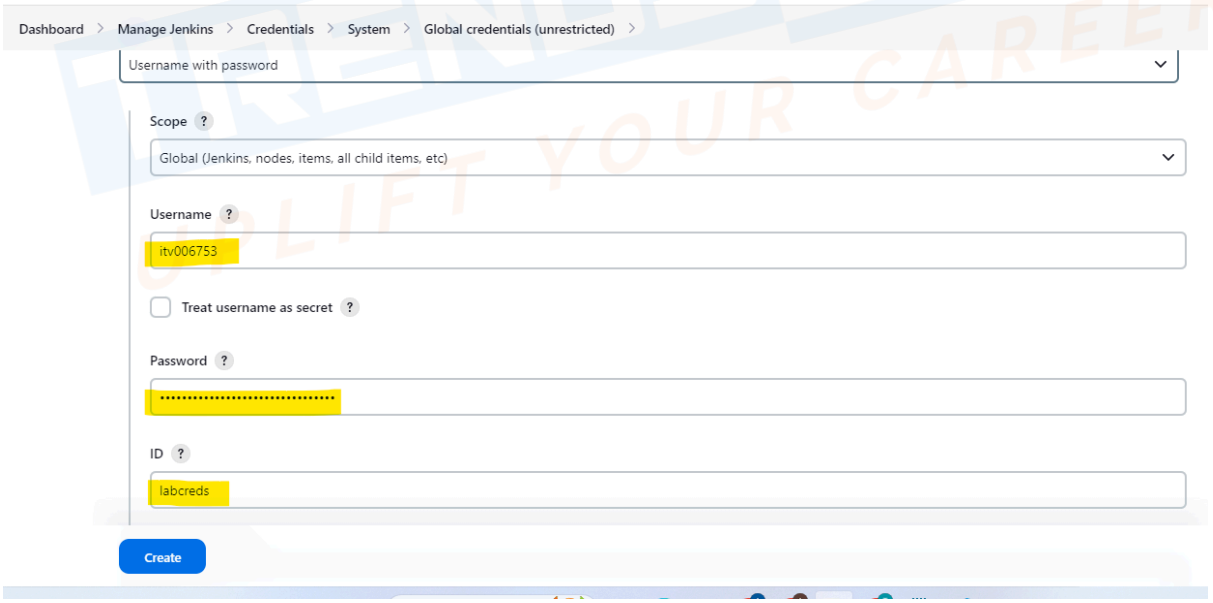
Go to jenkins => Manage jenkins => Credentials => System => Global Credentials => Add Credentials

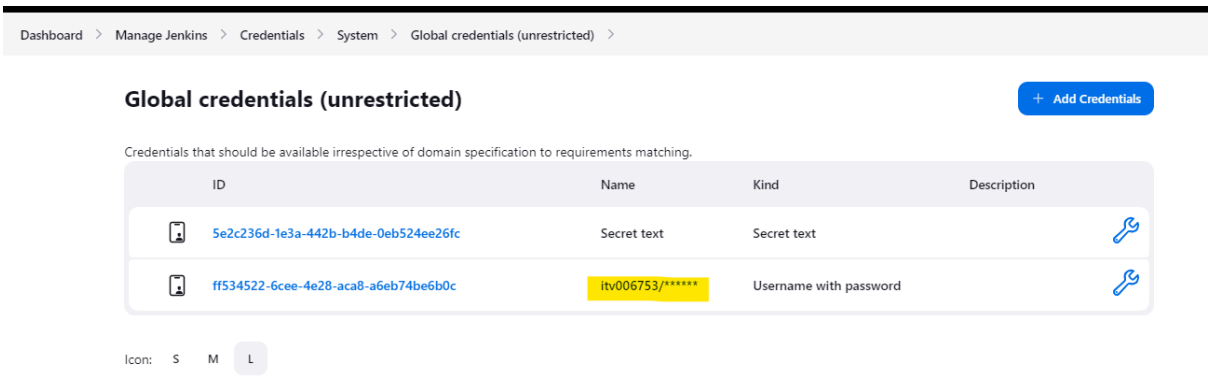






Provide the username, password and id for your "Multi Lab Cluster," and the system will generate the corresponding credentials as depicted below.



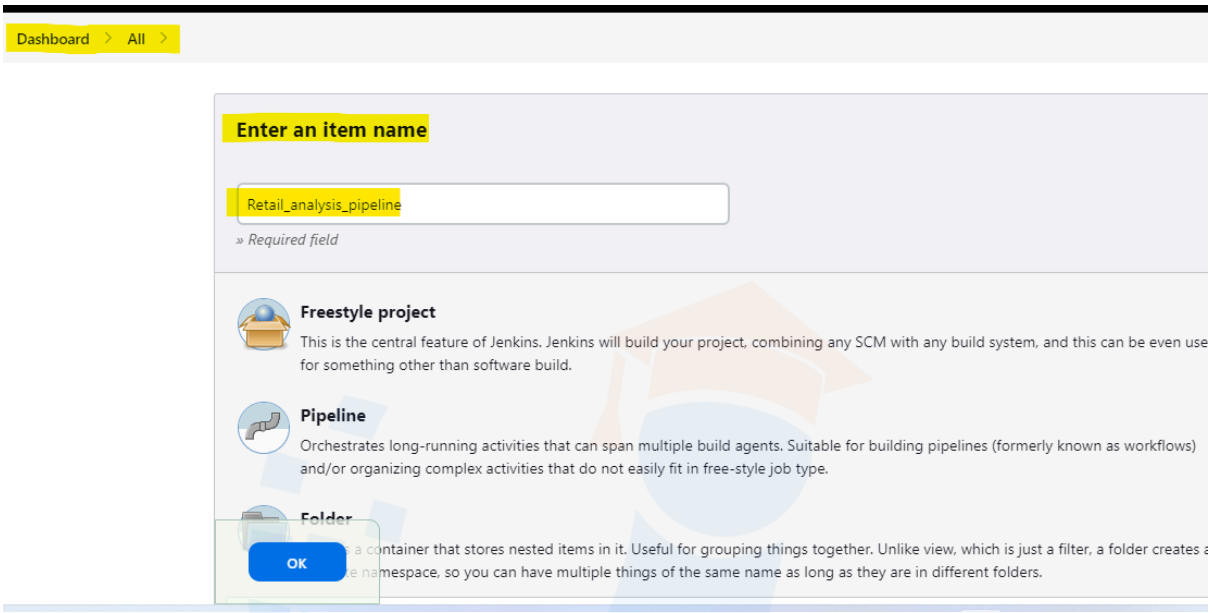


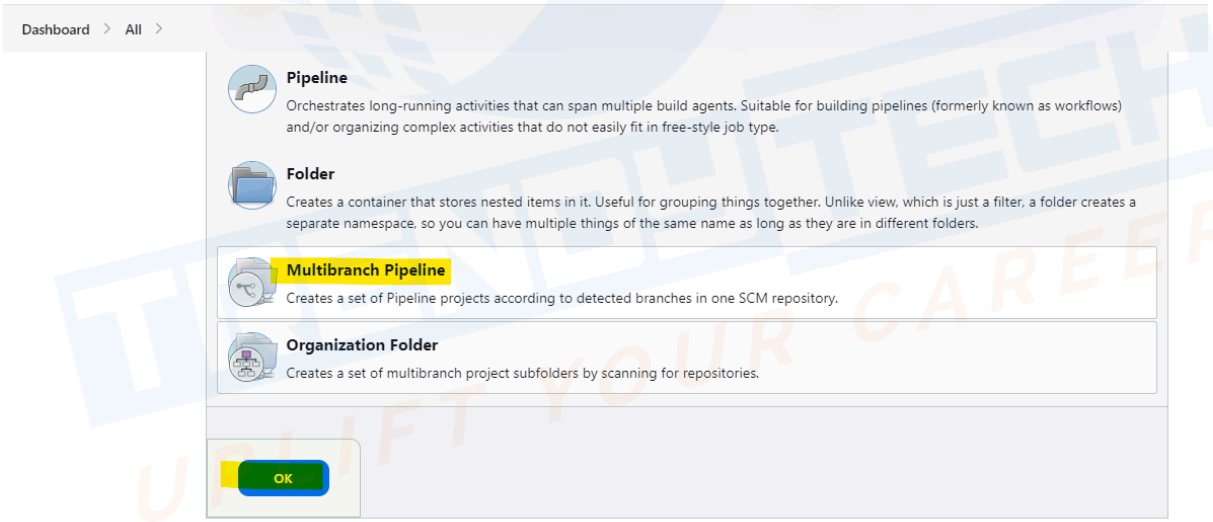
Note: Modify the Jenkinsfile with the provided code, and proceed to add, commit and push the changes.

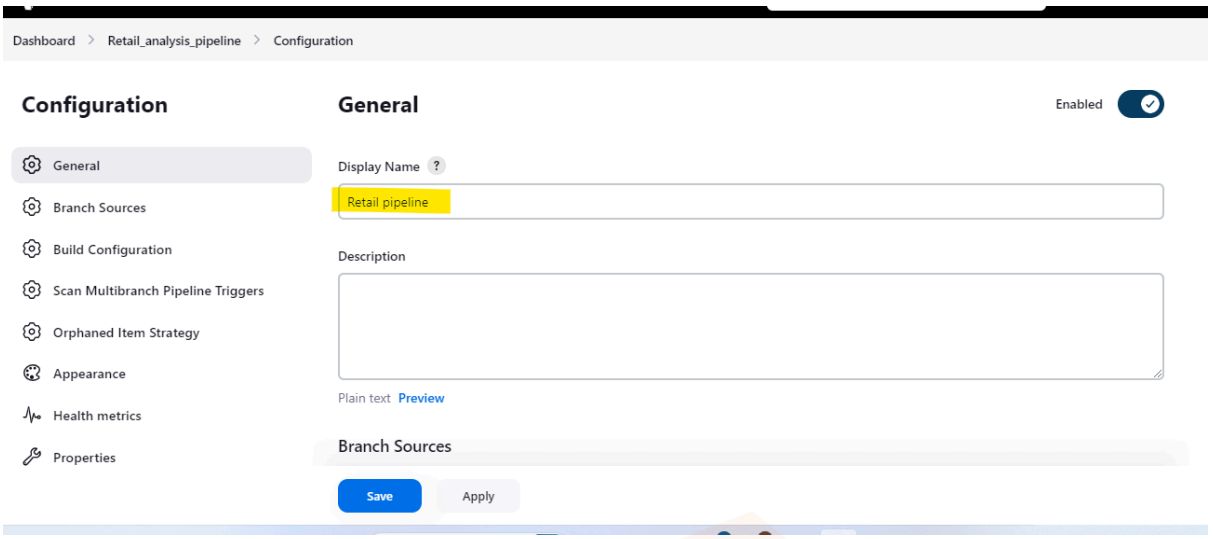
```
C:\Users\HP\Desktop\Retail Analysis 2>git add .

C:\Users\HP\Desktop\Retail Analysis 2>git commit -m "Edited jenkins file"
[feature-rp-50001 ba153a1] Edited jenkins file
 1 file changed, 27 insertions(+)

C:\Users\HP\Desktop\Retail Analysis 2>git push origin feature-rp-50001
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 395 bytes | 395.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/aratishatti15/Retail_analysis2.git
   b499b8b..ba153a1  feature-rp-50001 -> feature-rp-50001
```

Creation of Multibranch pipeline in Jenkins

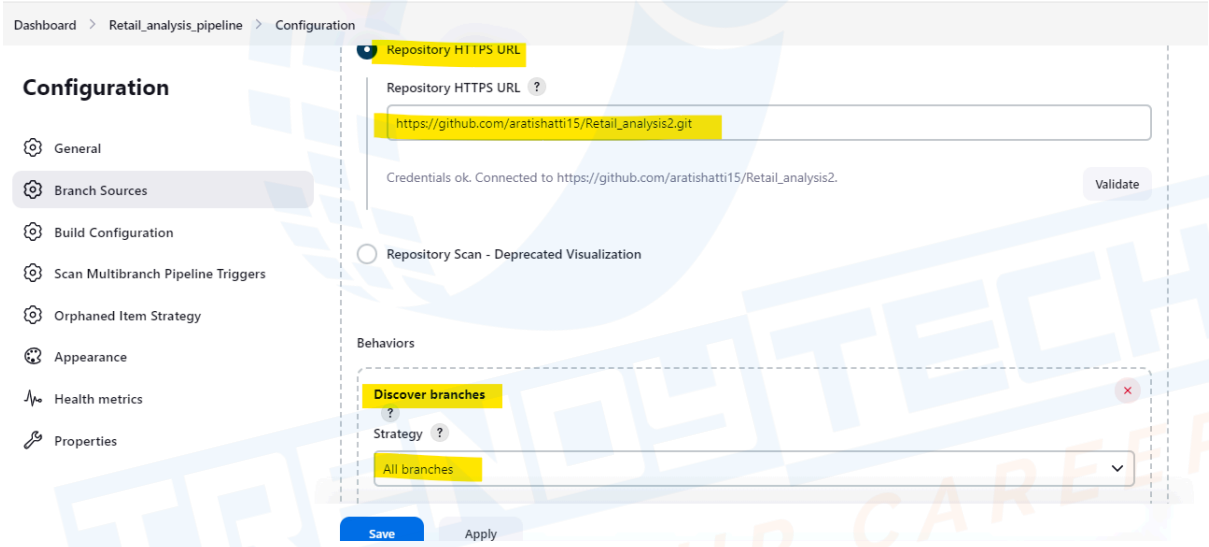
Go to the Jenkins UI (Dashboard) => New items => Create the multibranch pipeline



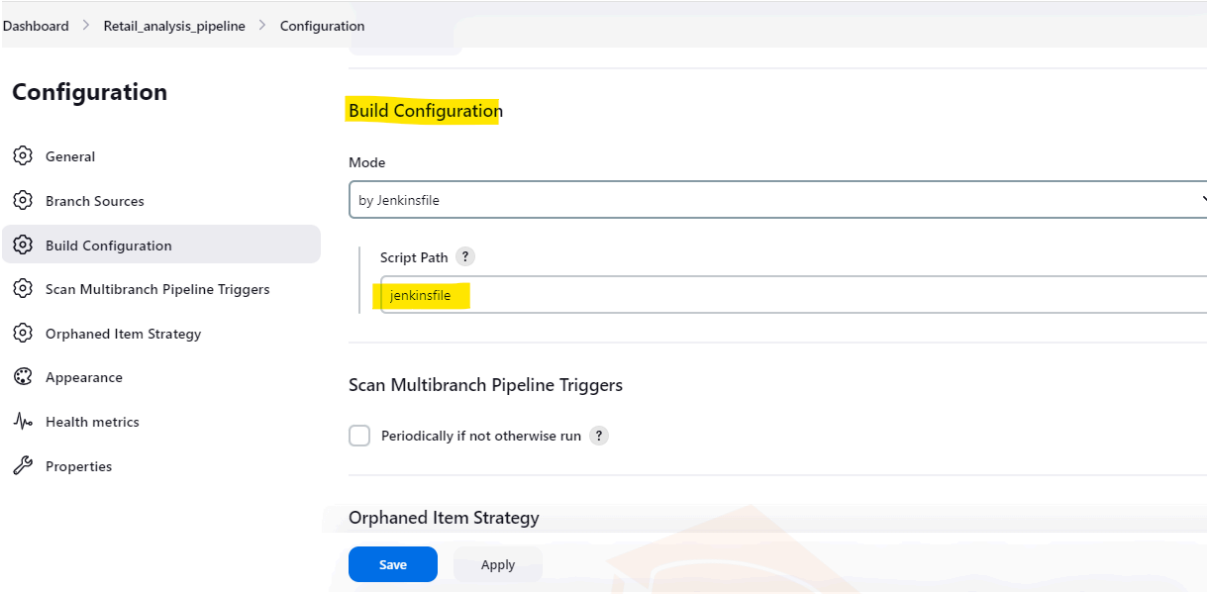




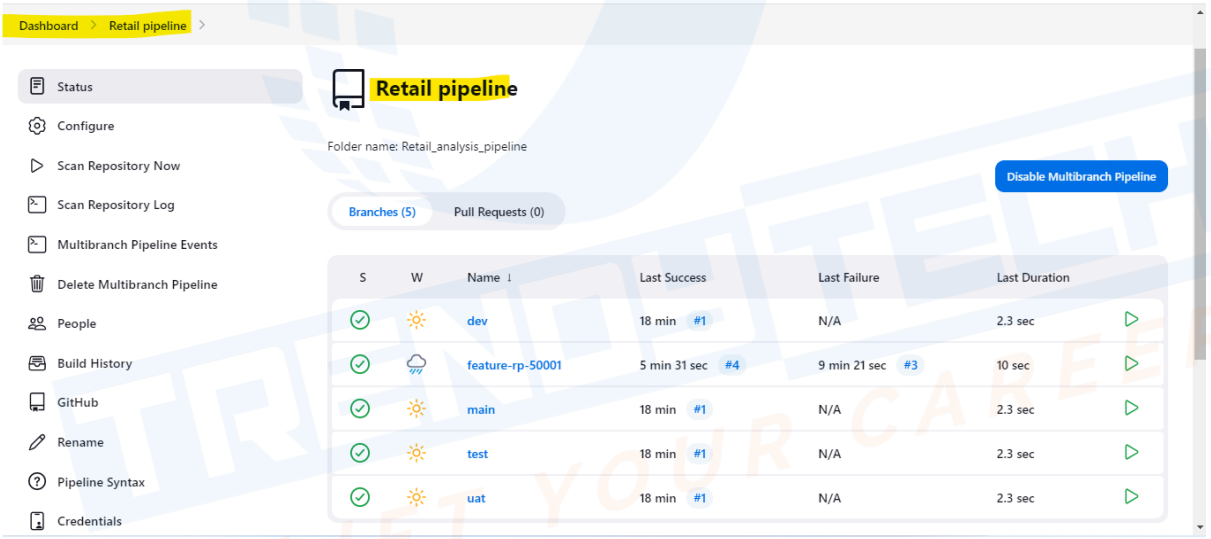
Add the URL of Retail Analysis repository and in discover branches option mention “All branches”



Note: Mention the name of jenkins file as you have created as it is case sensitive



After the pipeline is triggered and successfully executed, you'll observe the following output in the Jenkins UI.

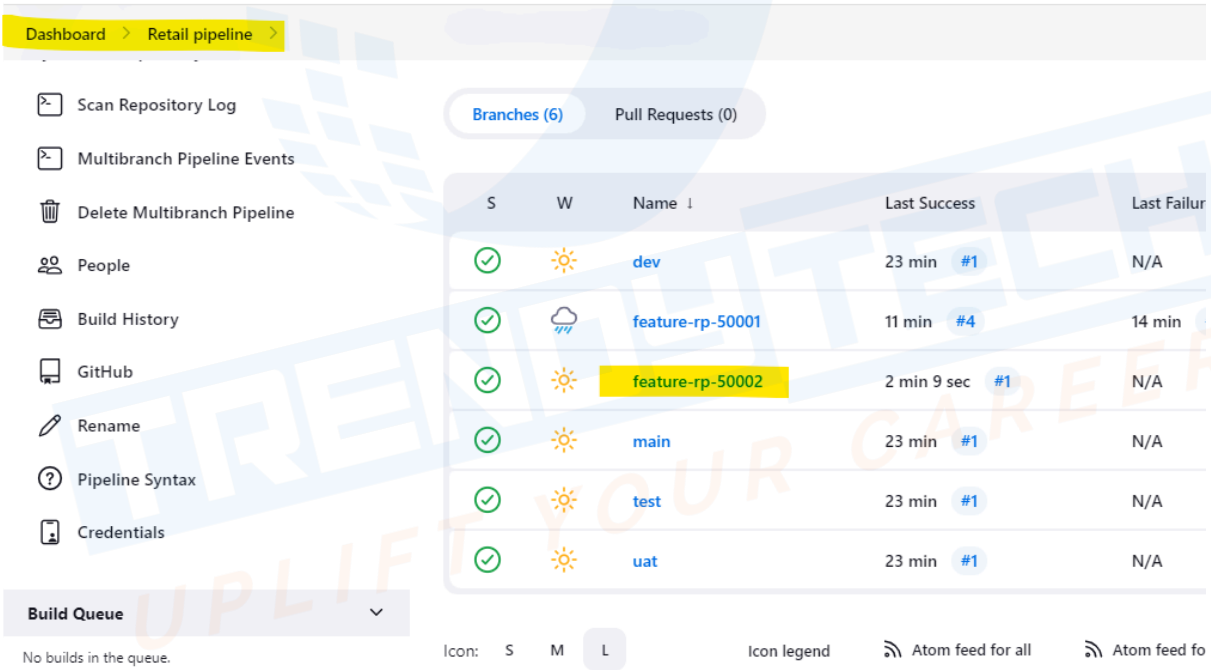


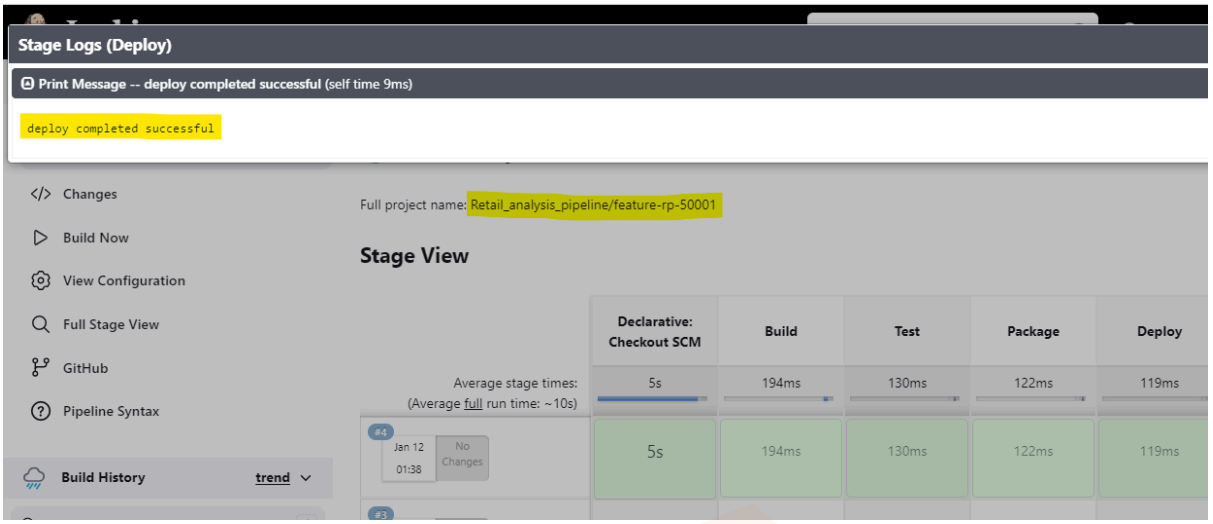
This indicates that the pipeline is successfully triggered upon the git push.

Now create the new branch "**git checkout -b feature-rp-50001** " and push changes to remote using the command "**git push origin feature-rp-50001**".

```
C:\Users\HP\Desktop\Retail Analysis 2>git checkout -b feature-rp-50002
Switched to a new branch 'feature-rp-50002'

C:\Users\HP\Desktop\Retail Analysis 2>git push origin feature-rp-50002
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'feature-rp-50002' on GitHub by visiting:
remote:      https://github.com/aratishatti15/Retail_analysis2/pull/new/feature-rp-50002
remote:
To https://github.com/aratishatti15/Retail_analysis2.git
 * [new branch]      feature-rp-50002 -> feature-rp-50002

C:\Users\HP\Desktop\Retail Analysis 2>
```

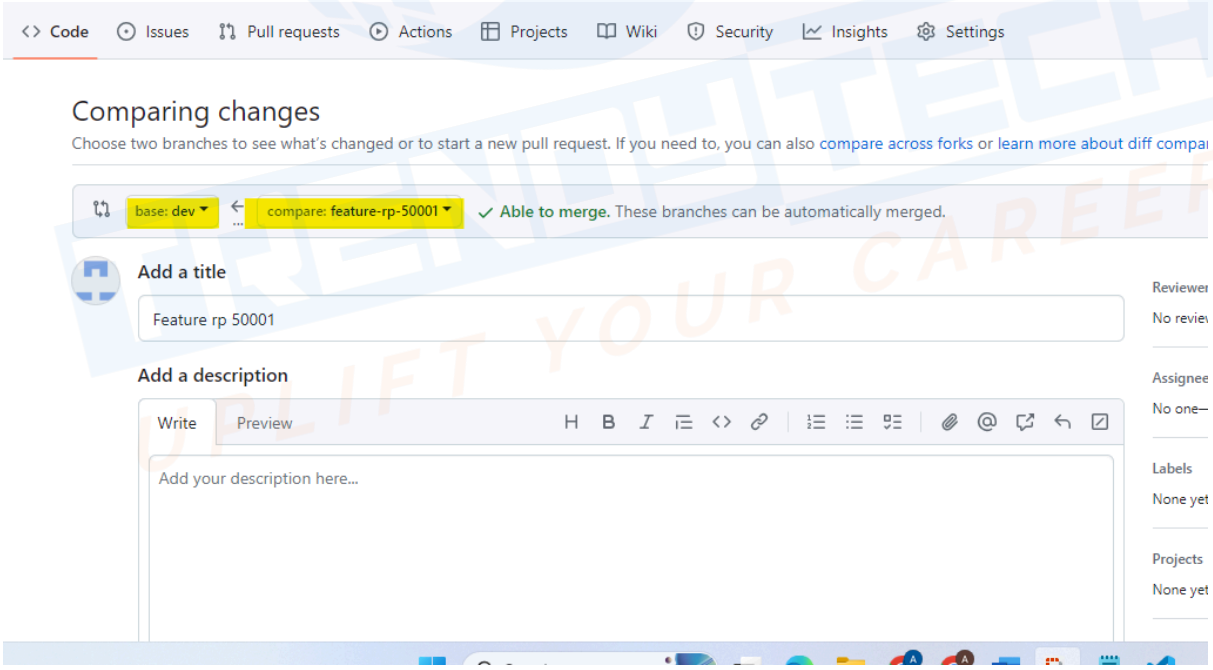
After the pipeline is triggered and successfully executed, you'll observe the following output in the Jenkins UI.

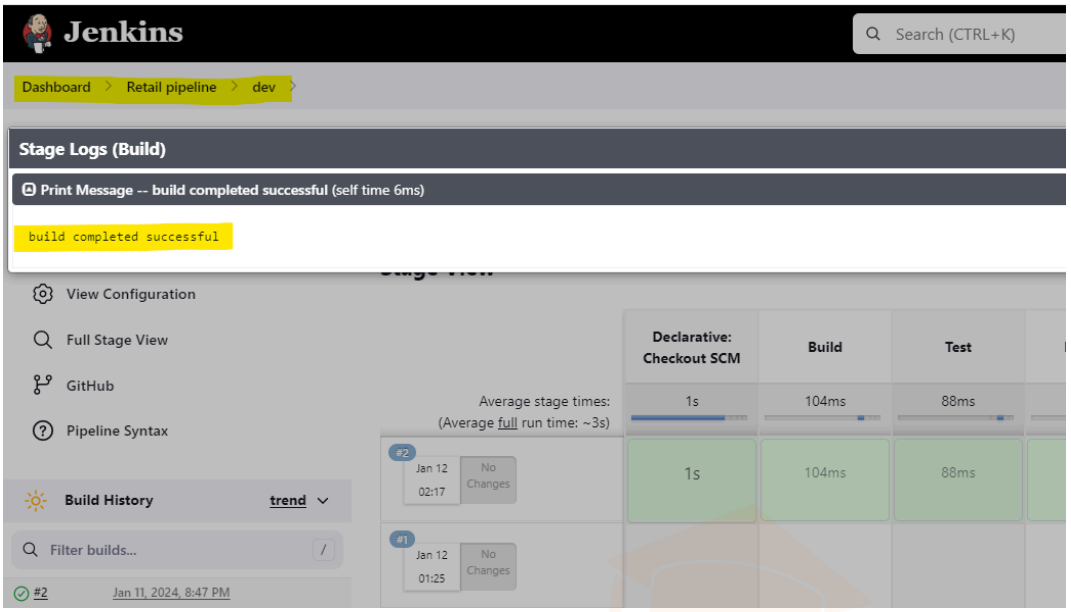




This indicates that the pipeline is successfully triggered upon the creation of a new branch.

Now create and approve the pull request for merging changes of branch "**feature-rp-50001" into "dev" branch.**





This indicates that the pipeline is successfully triggered upon the creation of a new branch.

To facilitate the deployment of the Retail Project code from our local system to the edge node within our Multi-node lab, we'll craft a Jenkins file tailored to this specific use case.

Modify the Jenkinsfile with the provided code in feature-rp-50002, and proceed to add, commit and push the changes.

```
PS C:\Users\HP\Desktop\Retail Analysis 2> git add .
PS C:\Users\HP\Desktop\Retail Analysis 2> git commit -m "Modified jenkins file"
[feature-rp-50002 9934cfd] Modified jenkins file
 1 file changed, 14 insertions(+), 9 deletions(-)
PS C:\Users\HP\Desktop\Retail Analysis 2> git push origin feature-rp-50002
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 579 bytes | 13.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/aratishatti15/Retail_analysis2.git
   e462ae2..9934cfd  feature-rp-50002 -> feature-rp-50002
PS C:\Users\HP\Desktop\Retail Analysis 2>
```

Now the pipeline will run and it will zip all the files in the Retail project folder as per code and using scp command it will copy files to the mentioned path in the code of edge node and you can see the retail project folder in edge node of multi node lab.