

DAY-2

DEVOPS

STEP -1 :INSTALL DOCKER

- 1) sudo apt update

```
jeeva@Jeeva:~$ sudo apt update
[sudo] password for jeeva:
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 3s (50.2 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them
```

- 2) sudo apt install -y docker.io

```
jeeva@Jeeva:~$ sudo apt install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base iptables libip4tc2 libip6tc2 libnetfilter-contrack3
  libnfnetlink0 libnftables1 libnftnl11 nftables pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc
  rinse zfs-fuse | zfsutils firewallld
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io iptables libip4tc2 libip6tc2 libnetfilter-contrack3
  libnfnetlink0 libnftables1 libnftnl11 nftables pigz runc ubuntu-fan
0 upgraded, 16 newly installed, 0 to remove and 13 not upgraded.
Need to get 79.6 MB of archives.
After this operation, 306 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 libip4tc2 amd64 1.8.10-3ubuntu2 [23.3 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/main amd64 libip6tc2 amd64 1.8.10-3ubuntu2 [23.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/main amd64 libnfnetlink0 amd64 1.0.2-2build1 [14.8 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/main amd64 libnetfilter-contrack3 amd64 1.0.9-6build1 [45.2 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/main amd64 libnftnl11 amd64 1.2.6-2build1 [66.0 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/main amd64 iptables amd64 1.8.10-3ubuntu2 [381 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/main amd64 libnftables1 amd64 1.0.9-1build1 [358 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/main amd64 nftables amd64 1.0.9-1build1 [69.8 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu3.1 [8599 kB]
```

STEP 2: ENABLE AND DISABLE

- 1) sudo systemctl enable docker
- 2) sudo systemctl start docker

```
jeeva@Jeeva:~$ sudo systemctl enable docker
jeeva@Jeeva:~$ sudo systemctl start docker
```

STEP 3:VERIFY THE INSTALLATION:

docker --version

```
jeeva@jeeva:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

STEP 4:INSTALL DOCKER COMPOSE

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose

```
jeeva@jeeva:~$ sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 71.4M 100 71.4M    0     0 3211k      0  0:00:22 0:00:22 --:--:-- 4038k
```

Give execution permission:

```
jeeva@jeeva:~$ sudo chmod +x /usr/local/bin/docker-compose
```

VERIFY INSTALLATION

```
jeeva@jeeva:~$ docker-compose --version
Docker Compose version v2.34.0
```

CREATE AN “HELLO WOLRD: APPLICATION

Create a project directory

```
jeeva@jeeva:~$ mkdir ~/docker-python-app
jeeva@jeeva:~$ cd ~/docker-python-app
```

Create the python Application File

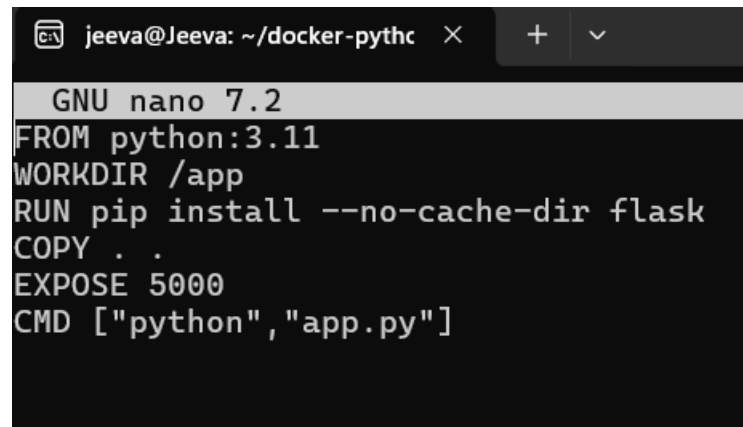
Create a file

```
jeeva@jeeva:~/docker-python-app$ nano app.py
jeeva@jeeva:~/docker-python-app$ cat app.py
from flask import Flask
app=Flask(__name__)
@app.route("/")
def hello():
    return "Hello,World!"
if __name__ == "__main__":
    app.run(host="0.0.0.0",port=5000)
jeeva@jeeva:~/docker-python-app$ nano requirements.txt
```

IN REQUIREMENTS.TXT TERMINAL WILL BE OPEN TYPE flask AND SAVE THE FILE BY CTRL+X,YES,ENTER.

STEP -5 : CREATE A DOCKER FILE

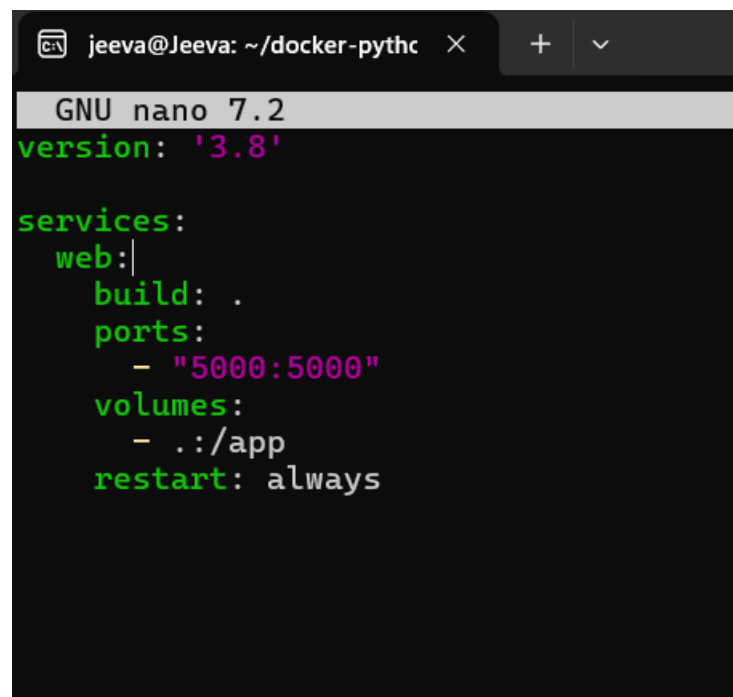
```
jeeva@Jeeva:~/docker-python-app$ nano Dockerfile
```



```
GNU nano 7.2
FROM python:3.11
WORKDIR /app
RUN pip install --no-cache-dir flask
COPY . .
EXPOSE 5000
CMD ["python", "app.py"]
```

CREATE A DOCKER COMPOSE

```
jeeva@Jeeva:~/docker-python-app$ nano docker-compose.yml
```



```
GNU nano 7.2
version: '3.8'

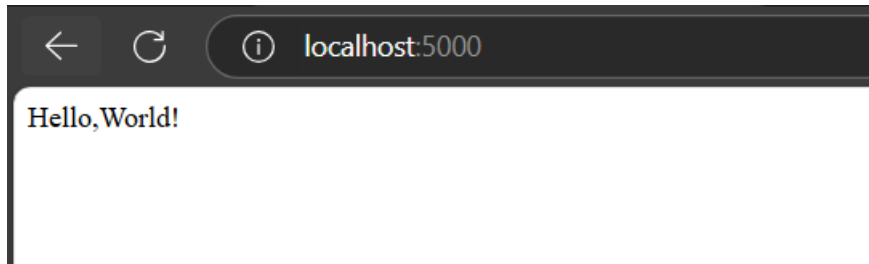
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ../app
    restart: always
```

STEP 6:BUILD AND RUN THE DOCKER CONTAINER

```
jeeva@Jeeva:~/docker-python-app$ nano docker-compose.yml
jeeva@Jeeva:~/docker-python-app$ sudo docker-compose build
```

```
jeeva@Jeeva:~/docker-python-app$ sudo docker-compose up --build
```

OPEN THE LOCALHOST:5000 IT WILL DISPLAY OUTPUT OF CODE



```
jeeva@Jeeva:~/docker-python-app$ sudo docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES
jeeva@Jeeva:~/docker-python-app$ sudo docker ps -a
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS        NAMES
c1c028722180   docker-python-app-web  "python app.py"         31 seconds ago  Exited (137)  8 seconds ago  python-app-web-1
```

STEP-7 CREATE A NEW REPO IN GITHUB

<https://github.com/Jeeva-21BSR017/devops-sample.git>

STEP-8 GO TO THE LINK <https://github.com/settings/tokens/new>

STEP-9 CLICK TOKEN CLASSIC AND GENERATE TOKEN CLASSIC

STEP-10 IN GENERATE TOKEN CLASSIC GIVE THE NAME AND CLICK THE WORKFLOW AND ADMIN HOOK REPO

STEP-11 GENERATE TOKEN

ghp_nR2bCRC1DcFF8SQ8018UwdQm3I WV9W3zRexi

STEP-12 START THE JENKINS

Username:admin

Password:b0e507d6b0f14097ba040a5e1dd67f6d

STEP-13 Create a new ITEM AND PIPELINE THEN CLICK THE PIPELINE THEN PIPELINE SCM THEN GIT

STEP-14 PASTE THE GITHUB LINK AND IN CREDENTIALS ADD OPTIONS THEN JENKIN.

STEP-15 PROVIDE GITHUB USERNAME AND GENERATED TOKEN PASSWORD IN PASSWORD THEN GIVE THE ID AS YOUR PREFERENCE AFTER COMPLETING CLICK ADD

STEP-16 TO PUSH INTO GITHUB

1)Clone

```
jeeva@Jeeva:~$ git clone https://github.com/Jeeva-21BSR017/devops-sample.git
Cloning into 'devops-sample'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

2)check file

```
jeeva@Jeeva:~$ ls
devops-sample  docker-python-app
jeeva@Jeeva:~$ cd docker-python-app
jeeva@Jeeva:~/docker-python-app$ ls
Dockerfile  app.py  docker-compose.yml  requirements.txt
jeeva@Jeeva:~/docker-python-app$ mv app.py docker-compose.yml requirements.txt ../devops-sample/
jeeva@Jeeva:~/docker-python-app$ cd ..
jeeva@Jeeva:~$ ls
devops-sample  docker-python-app
```

3)Add to the repository

```
jeeva@Jeeva:~$ cd devops-sample
jeeva@Jeeva:~/devops-sample$ git add .
jeeva@Jeeva:~/devops-sample$ git commit -m "Initialize"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <jeeva@Jeeva.>) not allowed
jeeva@Jeeva:~/devops-sample$ git config --global user.email "jeevaa.21bsr@kongu.edu"
jeeva@Jeeva:~/devops-sample$ git config --global user.name "Jeeva-21BSR017"
jeeva@Jeeva:~/devops-sample$ git commit -m "Initialize"
[main c2779fb] Initialize
3 files changed, 19 insertions(+)
create mode 100644 app.py
create mode 100644 docker-compose.yml
create mode 100644 requirements.txt
```

```
jeeva@jeeva:~/devops-sample$ git push https://Jeeva-21BSR017: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@github.com/Jeeva-21BSR017/devops-sample.git
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 612 bytes | 612.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Jeeva-21BSR017/devops-sample.git
dd86655..c2779fb main -> main
```

STEP-17 Open docker image app.docker.com

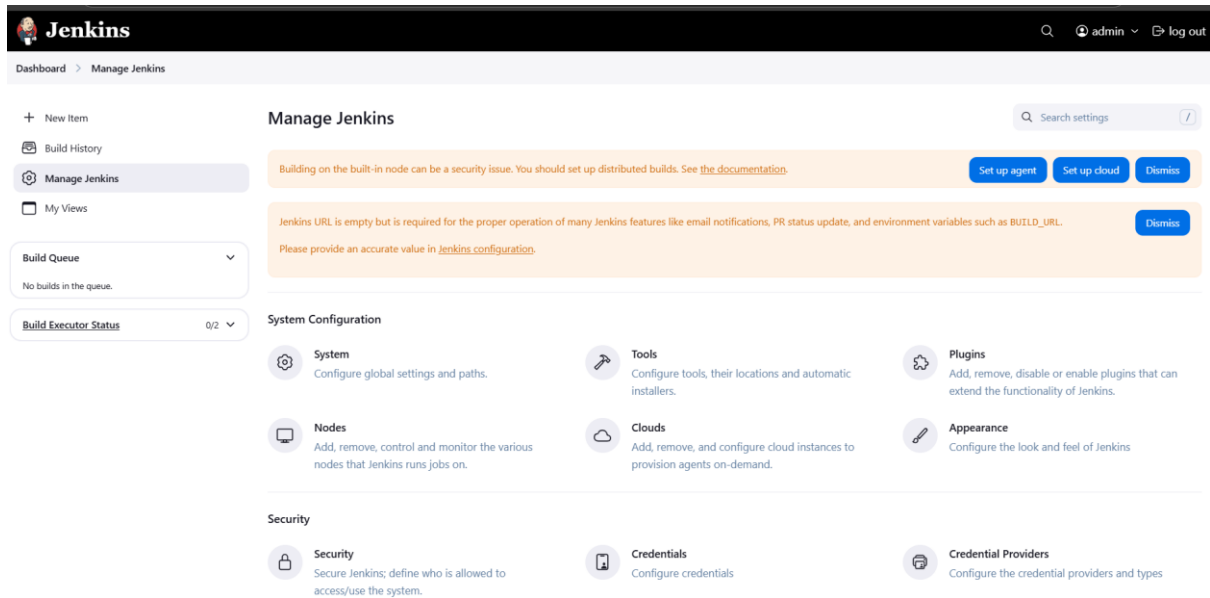
The screenshot shows a GitHub repository named 'devops-sample' owned by 'Jeeva-21BSR017'. The repository is public and has 2 commits. The main branch is selected. The file list shows: README.md (Initial commit, 2 hours ago), app.py (Initialize, 15 minutes ago), docker-compose.yml (Initialize, 15 minutes ago), and requirements.txt (Initialize, 15 minutes ago). The README section is visible, showing the repository name 'devops-sample'. On the right, there are sections for 'About' (No description, website, or topics provided), 'Releases' (No releases published), 'Packages' (No packages published), and 'Languages'.

The screenshot shows the Jenkins Dashboard. On the left, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The 'Build Queue' section shows 'No builds in the queue.' and the 'Build Executor Status' shows '0/2'. The main area displays a table of build history:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Install-Nginx	1 day 0 hr #1	N/A	12 sec
⌛	☀	jenkins	N/A	N/A	N/A

At the bottom, there is a 'Icon: S M L' selector and a 'Add description' button.

STEP-18 OPEN MANAGE JENKINS



The screenshot shows the Jenkins 'Manage Jenkins' page. The left sidebar contains navigation links: 'New Item', 'Build History', 'Manage Jenkins' (selected), and 'My Views'. Below these are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '0/2'). The main content area is titled 'Manage Jenkins' and includes a search bar. Two orange warning banners are present: one about distributed builds and another about the Jenkins URL. Below the warnings, the 'System Configuration' section is visible, containing icons and descriptions for 'System', 'Tools', 'Plugins', 'Nodes', 'Clouds', and 'Appearance'. The 'Security' section below it includes 'Security', 'Credentials', and 'Credential Providers'.

Jenkins

Dashboard > Manage Jenkins

+ New Item

Build History

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status 0/2

Manage Jenkins

Search settings

Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#).

Set up agent Set up cloud Dismiss

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as BUILD_URL.

Dismiss

Please provide an accurate value in [Jenkins configuration](#).

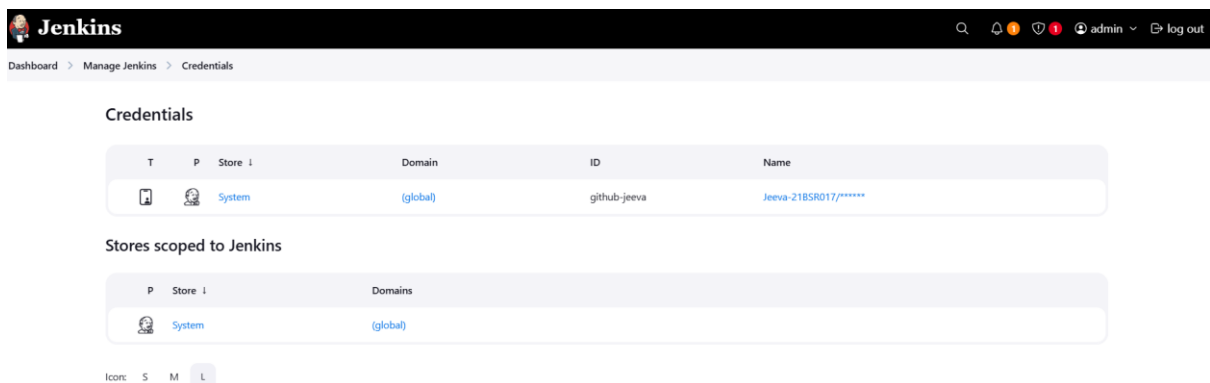
System Configuration

- System**
Configure global settings and paths.
- Tools**
Configure tools, their locations and automatic installers.
- Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Nodes**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**
Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**
Configure the look and feel of Jenkins

Security

- Security**
Secure Jenkins; define who is allowed to access/use the system.
- Credentials**
Configure credentials
- Credential Providers**
Configure the credential providers and types

STEP-19 OPEN CREDENTIALS



The screenshot shows the Jenkins 'Credentials' page. The left sidebar shows the navigation path: 'Dashboard > Manage Jenkins > Credentials'. The main content area is titled 'Credentials' and features a table with columns 'T', 'P', 'Store', 'I', 'Domain', 'ID', and 'Name'. A single credential is listed with 'System' as the provider, 'github-jeeva' as the ID, and 'jeeva-21BSR017/*****' as the name. Below the table, the 'Stores scoped to Jenkins' section shows a table with 'System' as the provider and 'global' as the domain. At the bottom, there are icons for 'Icon', 'S', 'M', and 'L'.

Jenkins

Dashboard > Manage Jenkins > Credentials

Credentials

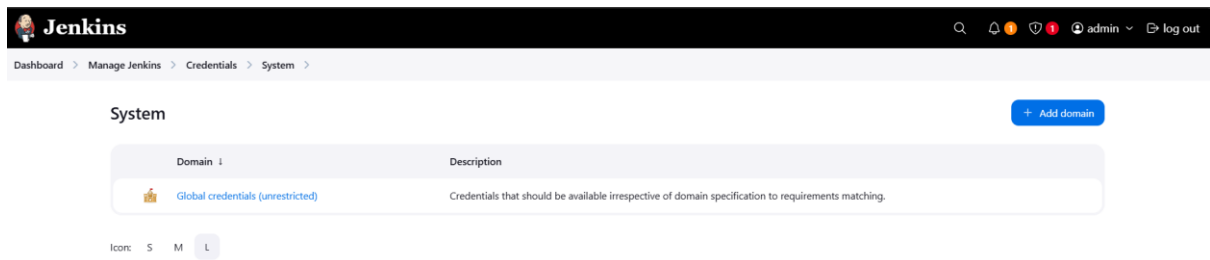
T	P	Store	I	Domain	ID	Name
		System		(global)	github-jeeva	jeeva-21BSR017/*****

Stores scoped to Jenkins

P	Store	I	Domains
	System		(global)

Icon: S M L

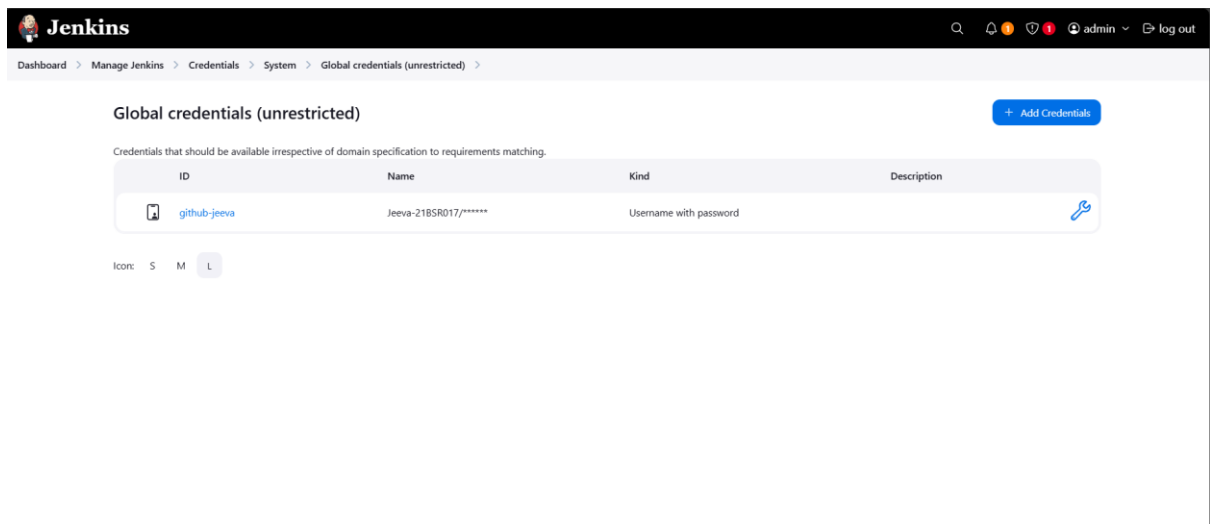
STEP-20 OPEN SYSTEM



The Jenkins System configuration page shows a table with one entry: 'Global credentials (unrestricted)'. The table has columns for 'Domain 1' and 'Description'. The description states: 'Credentials that should be available irrespective of domain specification to requirements matching.' There is a '+ Add domain' button in the top right corner. Below the table, there are icons for 'Icon: S M L'.

Domain 1	Description
Global credentials (unrestricted)	Credentials that should be available irrespective of domain specification to requirements matching.

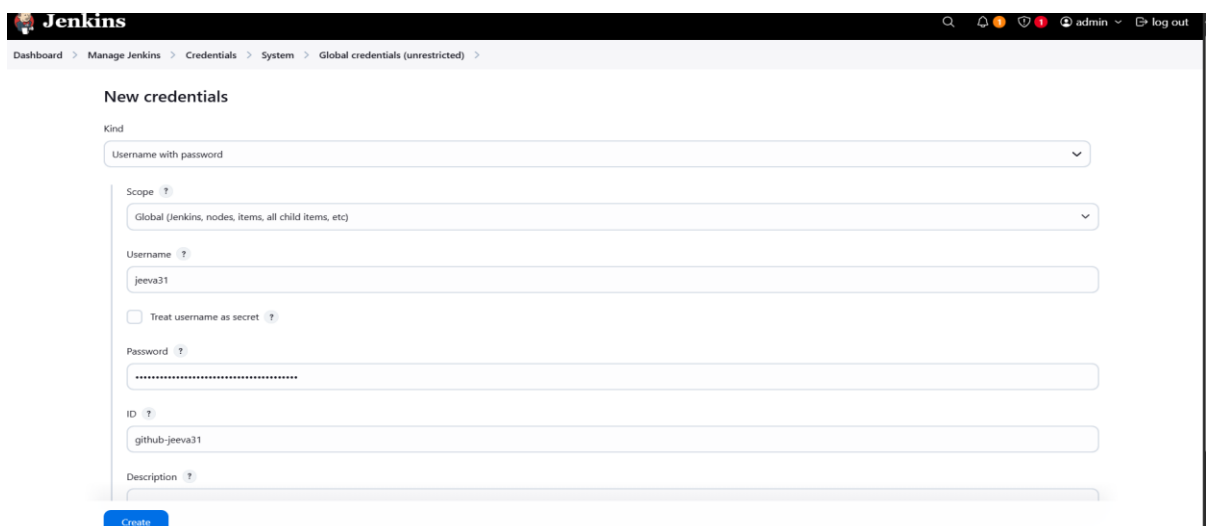
STEP-21 OPEN MANAGE JENKINS



The Jenkins Global credentials (unrestricted) page shows a table with one entry: 'github-jeeva'. The table has columns for 'ID', 'Name', 'Kind', and 'Description'. The description states: 'Credentials that should be available irrespective of domain specification to requirements matching.' There is a '+ Add Credentials' button in the top right corner. Below the table, there are icons for 'Icon: S M L'.

ID	Name	Kind	Description
github-jeeva	Jeeva-21BSR017/*****	Username with password	

STEP-22 OPEN GLOBAL CREDENTIALS



The Jenkins New credentials form is shown. It has fields for 'Kind' (Username with password), 'Scope' (Global (Jenkins, nodes, items, all child items, etc)), 'Username' (jeeva31), 'Treat username as secret' (checkbox), 'Password' (masked), 'ID' (github-jeeva31), and 'Description'. There is a 'Create' button at the bottom.

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: jeeva31

☐ Treat username as secret

Password: [masked]

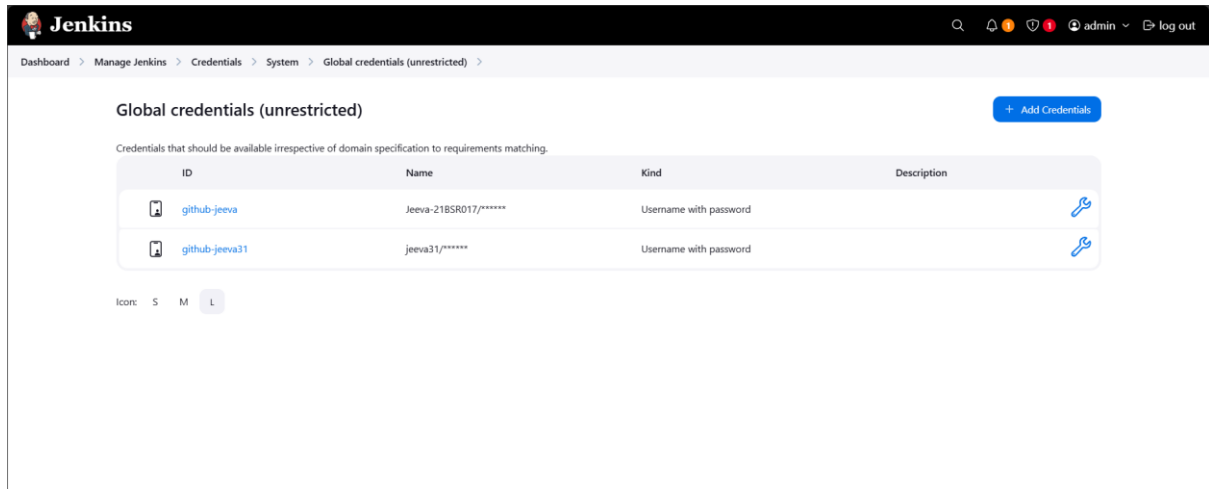
ID: github-jeeva31

Description: [empty]

Create

STEP-23 OPEN MANAGE JENKINS

=>IT DISPLAYS THE GITHUB CREDENTIALS ID AND DOCKER CREDENTIALS ID



STEP 24:OPEN UBUNTU AND CREATE nano Jenkinsfile

STEP 25:

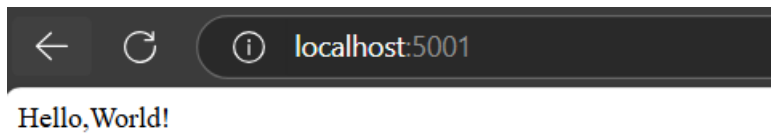
```
if [ "$(docker ps -aq -f name=$CONTAINER_NAME)" ]; then
    docker stop $CONTAINER_NAME || true
    docker rm $CONTAINER_NAME || true
fi
'''
}
}
}

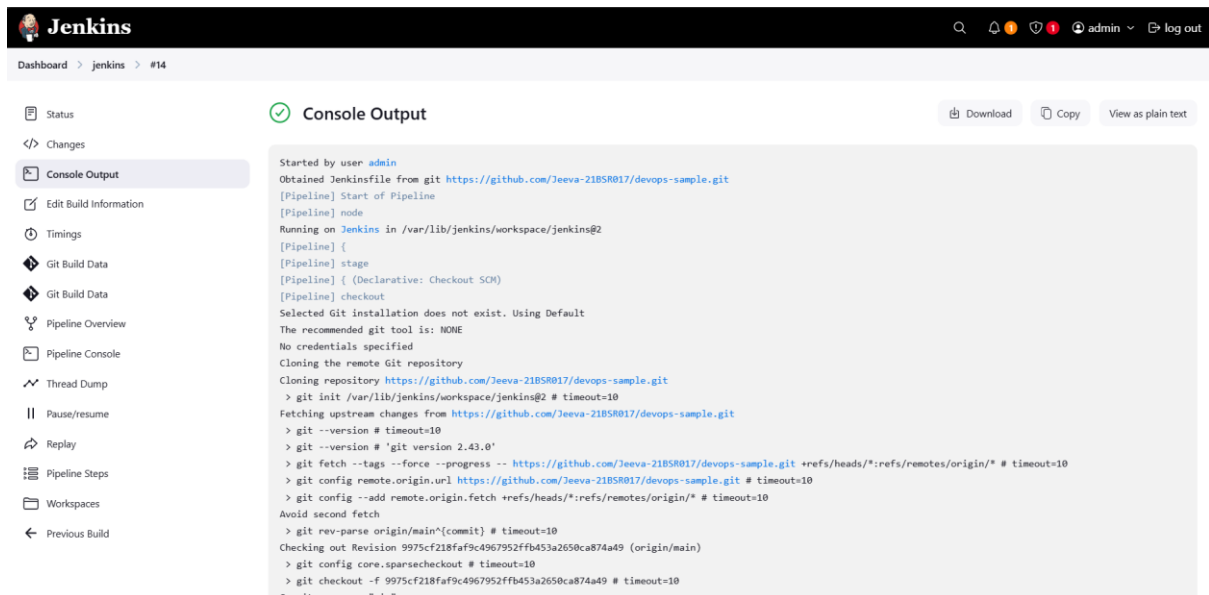
stage('Run Docker Container') {
    steps {
        sh 'docker run -d -p 5001:5000 --name $CONTAINER_NAME
$DOCKER_IMAGE'
    }
}
}
```

```
post {  
  success {  
    echo "Build, push, and container execution successful!"  
  }  
  failure {  
    echo "Build or container execution failed."  
  }  
}  
}
```

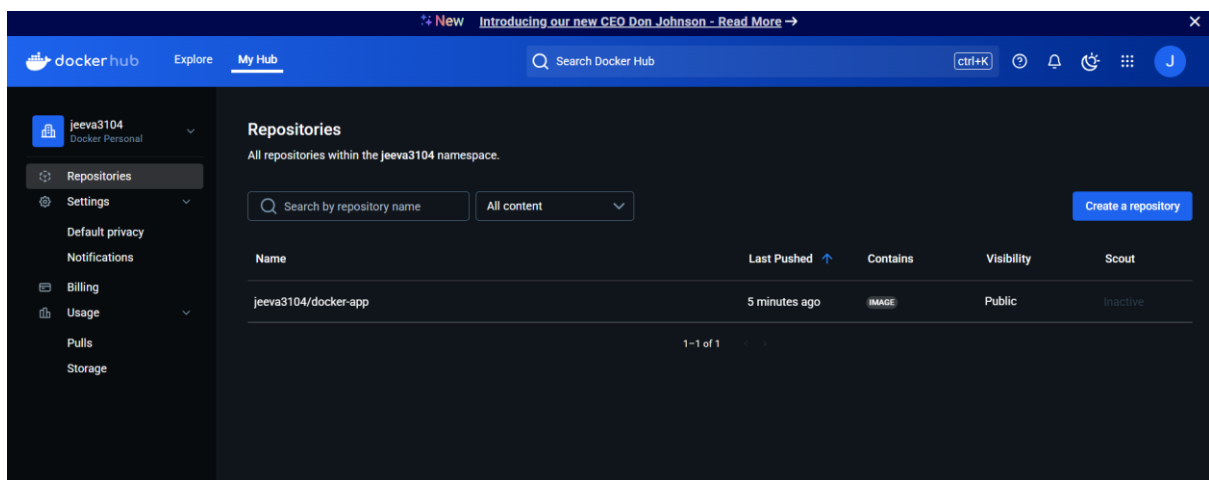
STEP 26:IN JENKINS BUILD NOW THE Jenkins


STEP 27:IN LOCALHOST:5001 IT DISPLAYS THE OUTPUT.





STEP-28 : IN DOCKERHUB THE PROCESS ARE TO DONE AND THEN THE LINUX IMAGE HAS TO BE DISPLAY.



 jeeva3104
Docker Personal

▼

Repositories

▼

Settings

▼

Default privacy

Notifications

Billing

Usage

▼

Pulls

Storage

Repositories / docker-app / General

jeeva3104/docker-app

Last pushed 6 minutes ago

Add a description

Add a category

General

Tags

Image Management

Collaborators

Webhooks

Settings

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	6 minutes

[See all](#)