DOCKER ASSIGNMENT

1. AUTOMATE DOCKER BUILT AND PUSH USING JENKINSFILE

STEP 1: Setup a Simple Flask App Project Structure

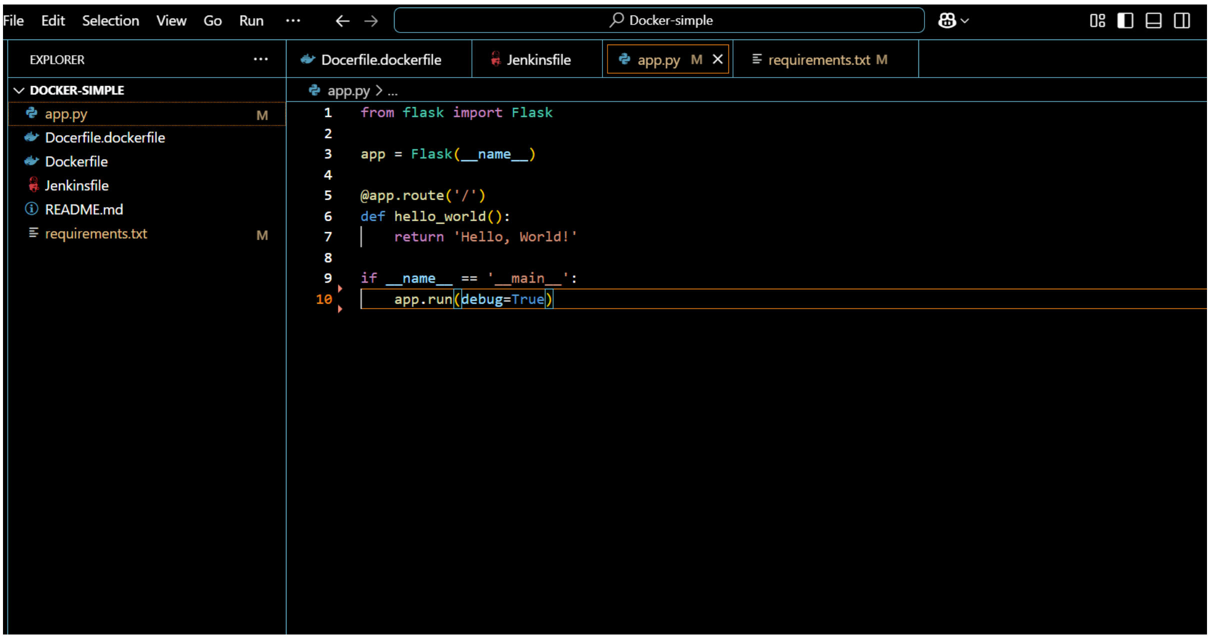
```
my-flask-app
│── app.py
│── requirements.txt
│── Dockerfile
│── Jenkinsfile
```

app.py: The main Flask application file.

requirements.txt: List of dependencies (Flask and others).

Dockerfile: Defines the Docker image for the Flask app.

Jenkinsfile: Contains the Jenkins pipeline configuration

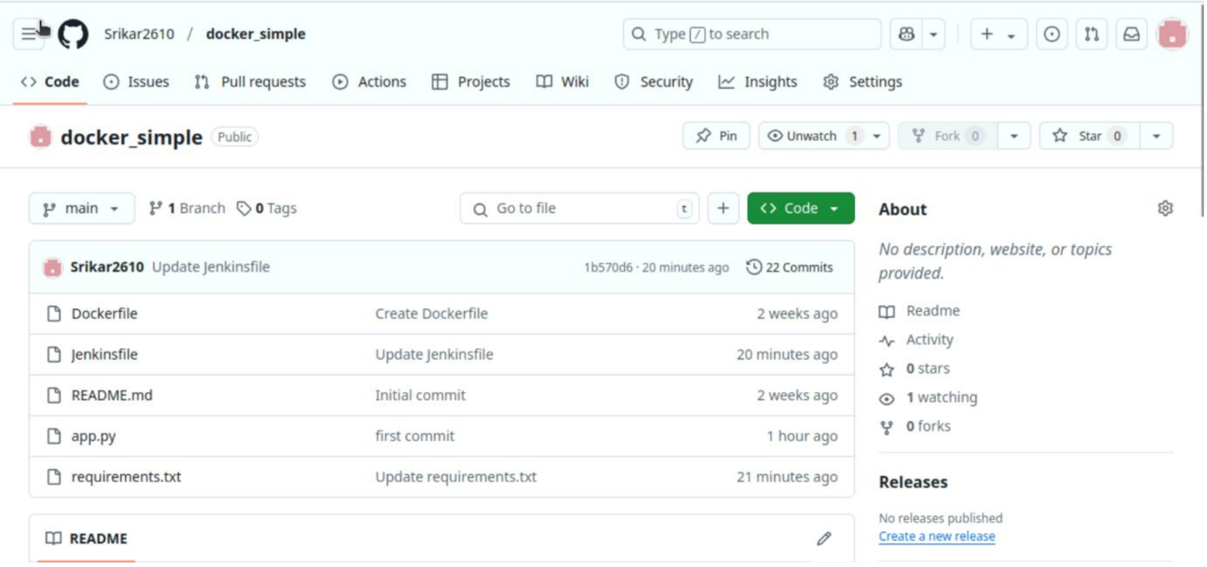


```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(debug=True)
```

STEP 2: Push the Code to GitHub

- Make sure you have a GitHub repository created for the project.
- Push all the files (app.py, requirements.txt, Dockerfile, Jenkinsfile) to the GitHub repository

GITHUB URL for the code: https://github.com/Srikar2610/docker_simple.git



Srikar2610 / docker_simple

docker_simple Public

main · 1 Branch · 0 Tags

Srikar2610 Update Jenkinsfile · 1b570d6 · 20 minutes ago · 22 Commits

Dockerfile	Create Dockerfile	2 weeks ago
Jenkinsfile	Update Jenkinsfile	20 minutes ago
README.md	Initial commit	2 weeks ago
app.py	first commit	1 hour ago
requirements.txt	Update requirements.txt	21 minutes ago

README

About

No description, website, or topics provided.

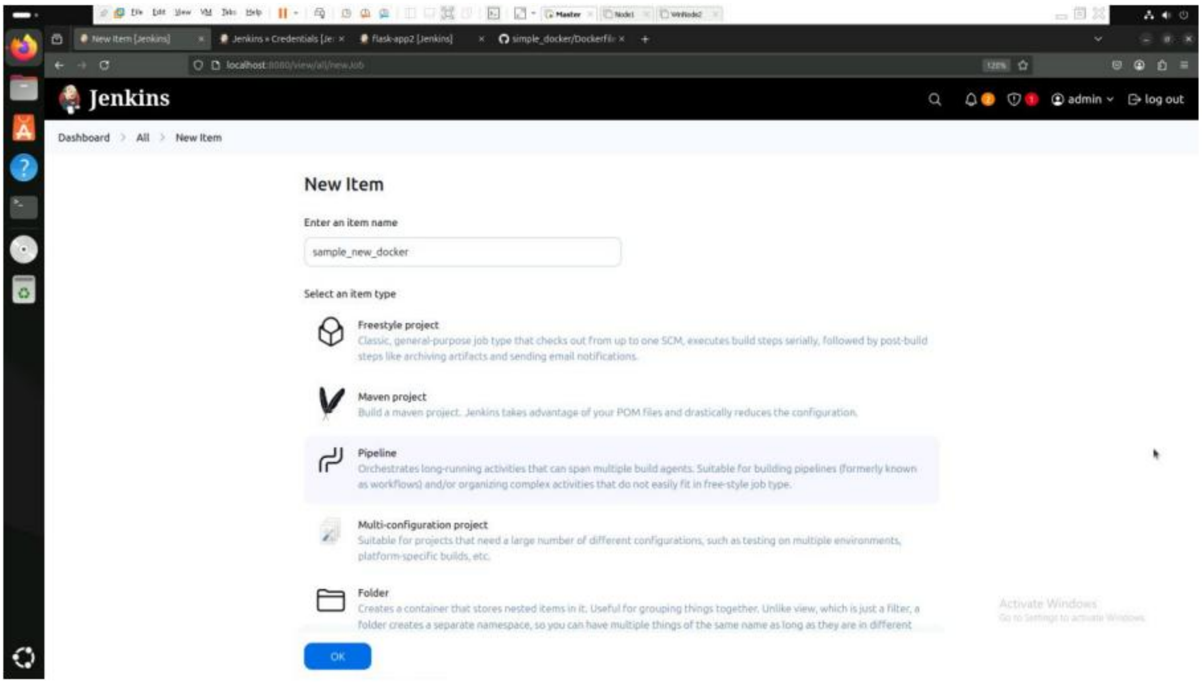
Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

STEP 3: Create a New Pipeline in Jenkins

- In Jenkins, click New Item > Pipeline.
- Enter a name for the pipeline.
- Under Pipeline Definition, select Pipeline script from SCM.
  - Select Git as the SCM.
  - Enter the GitHub repository URL (https://github.com/your-username/my□flask-app.git).
  - Set the branch (typically master or main).
- Click Save.



localhost:8080/view/all/newJob

Dashboard > All > New Item

New Item

Enter an item name

sample_new_docker

Select an item type

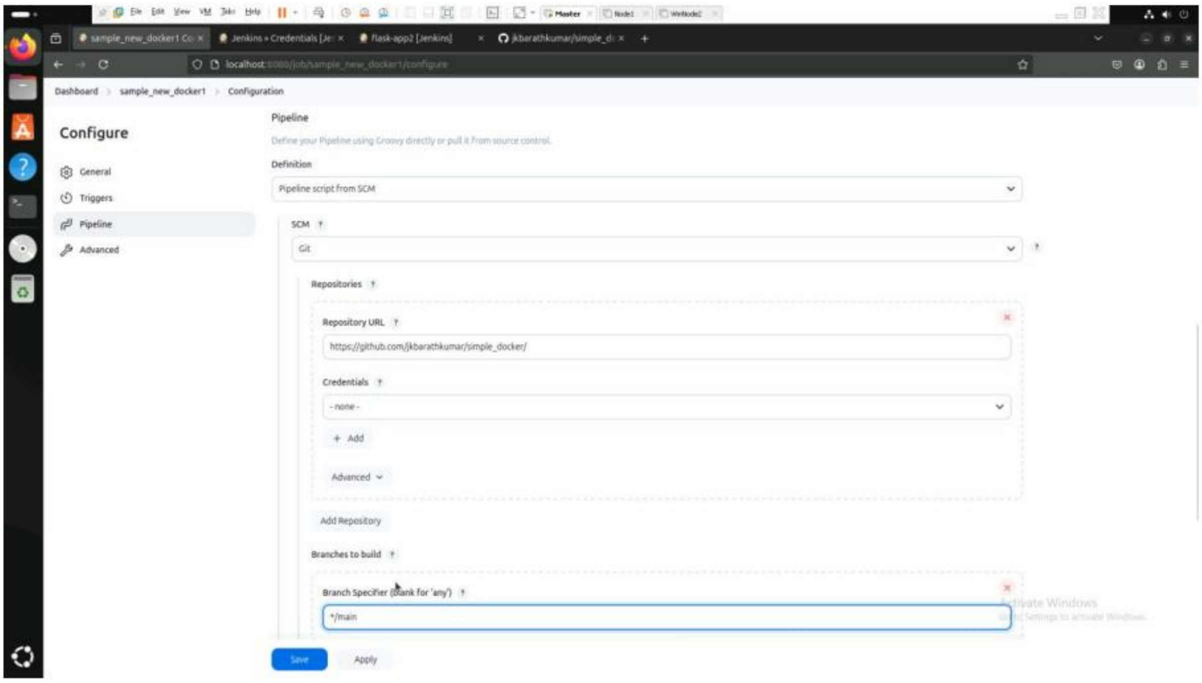
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different



Dashboard > sample_new_docker1 > Configuration

Configure

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/jkbarathkumar/simple_docker/

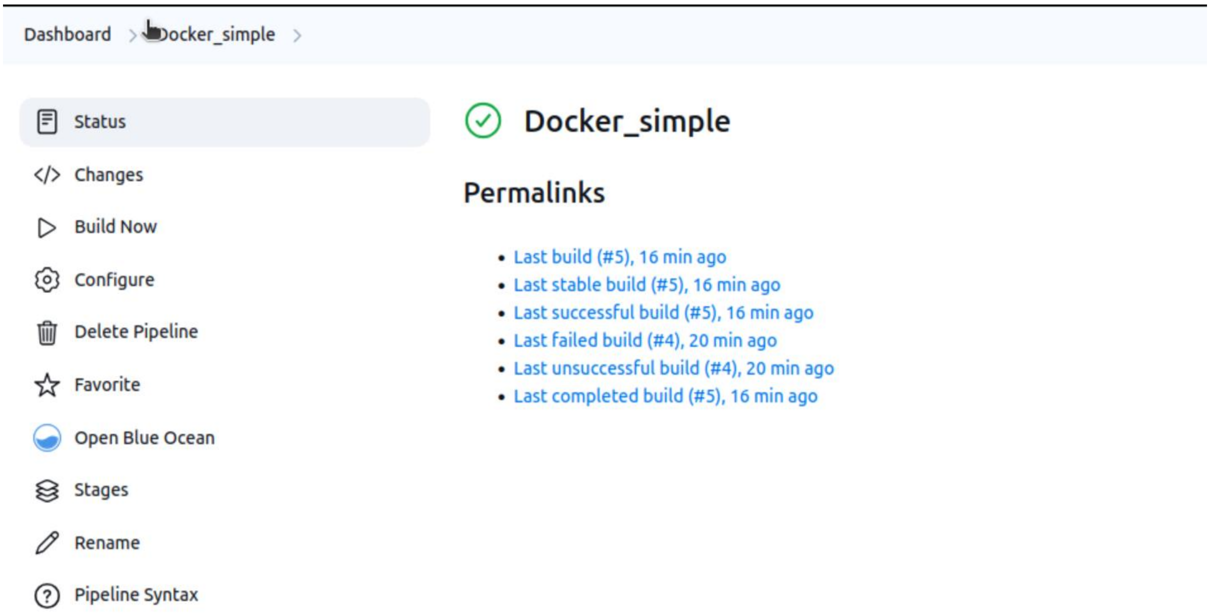
Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */main

STEP 4: Click Build Now

- Click Build Now in Jenkins to trigger the build.
- Jenkins will:
  - Checkout the code from GitHub.
  - Build the Docker image.
  - Push the image to Docker Hub

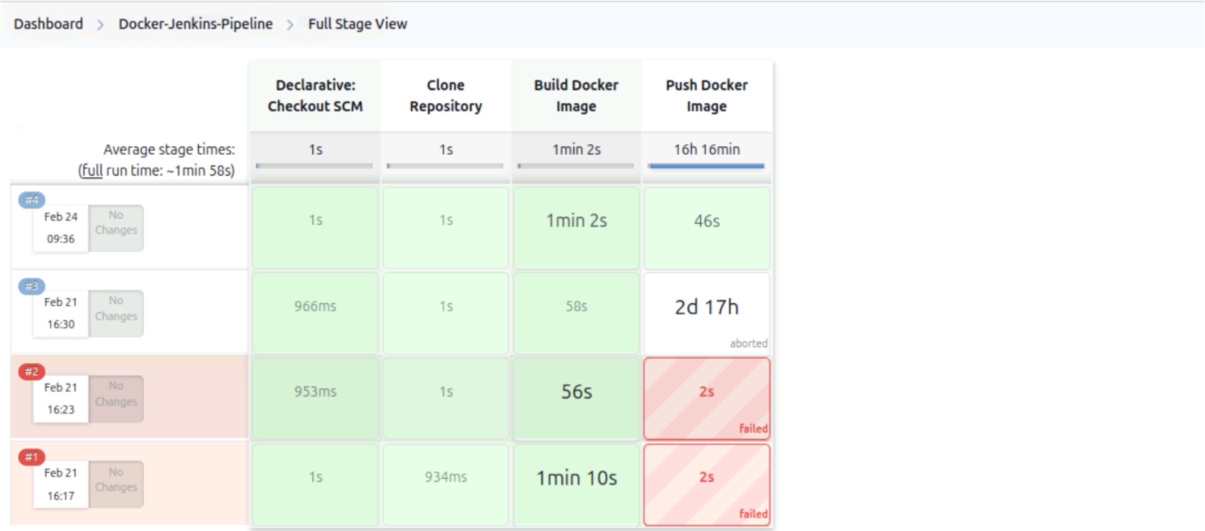


Dashboard > Docker_simple >

Docker_simple

Permalinks

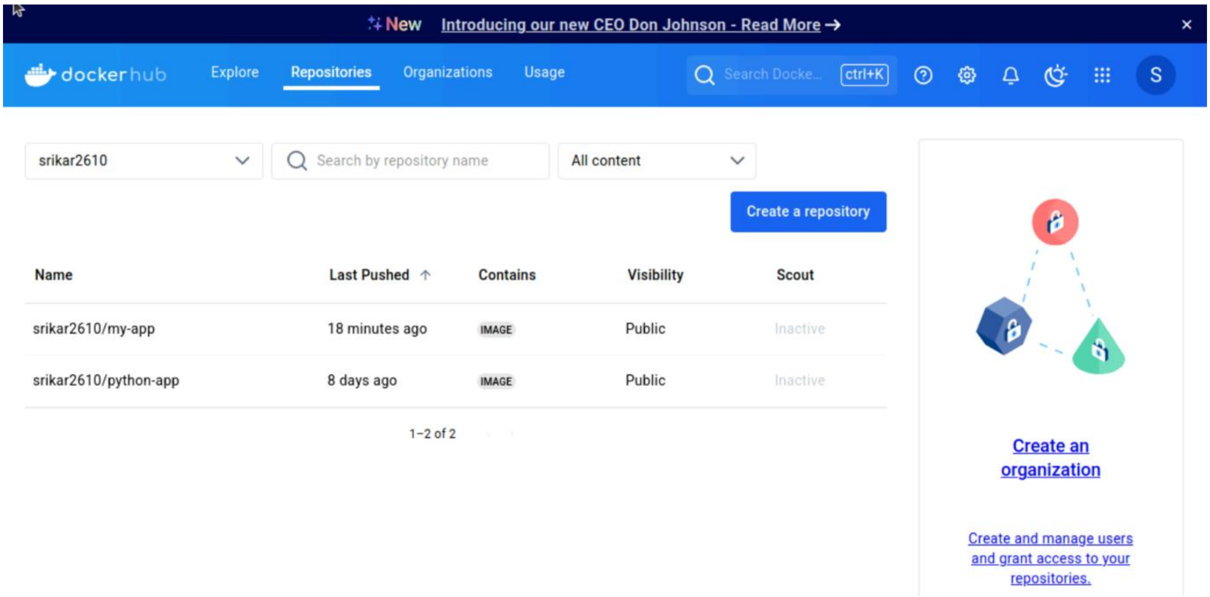
- Last build (#5), 16 min ago
- Last stable build (#5), 16 min ago
- Last successful build (#5), 16 min ago
- Last failed build (#4), 20 min ago
- Last unsuccessful build (#4), 20 min ago
- Last completed build (#5), 16 min ago



	Declarative: Checkout SCM	Clone Repository	Build Docker Image	Push Docker Image
Average stage times: (full run time: ~1min 58s)	1s	1s	1min 2s	16h 16min
#4 Feb 24 09:36 No Changes	1s	1s	1min 2s	46s
#3 Feb 21 16:30 No Changes	966ms	1s	58s	2d 17h aborted
#2 Feb 21 16:23 No Changes	953ms	1s	56s	2s failed
#1 Feb 21 16:17 No Changes	1s	934ms	1min 10s	2s failed

STEP 5: Verify Docker Image on Docker Hub

- After the build finishes, log into your Docker Hub account.
- You should see the my-flask-app image under Repositories with the latest tag.



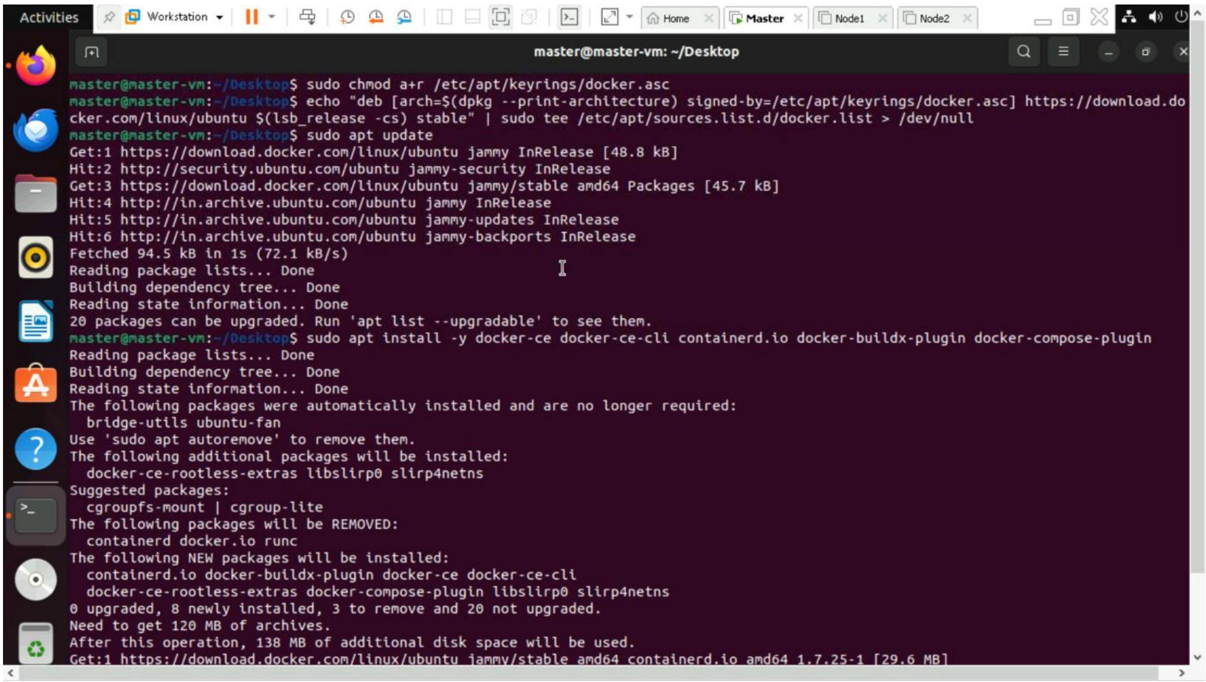
Name	Last Pushed ↑	Contains	Visibility	Scout
srikar2610/my-app	18 minutes ago	IMAGE	Public	Inactive
srikar2610/python-app	8 days ago	IMAGE	Public	Inactive

1–2 of 2

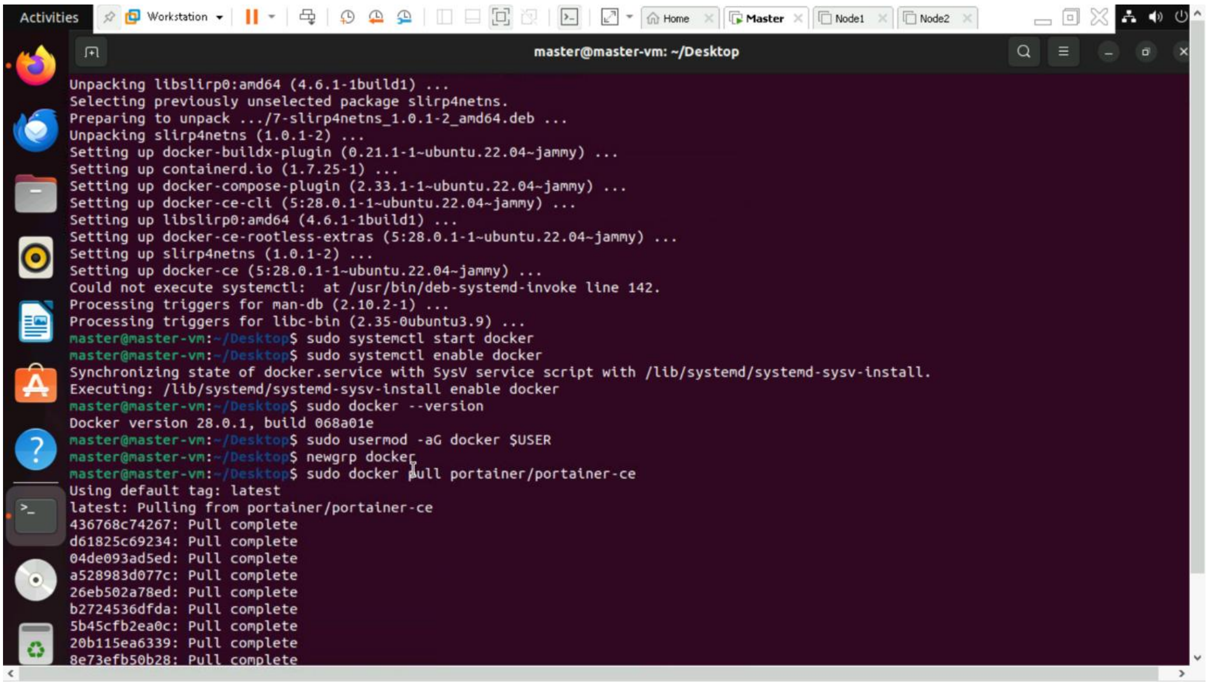
Create an organization

Create and manage users and grant access to your repositories.

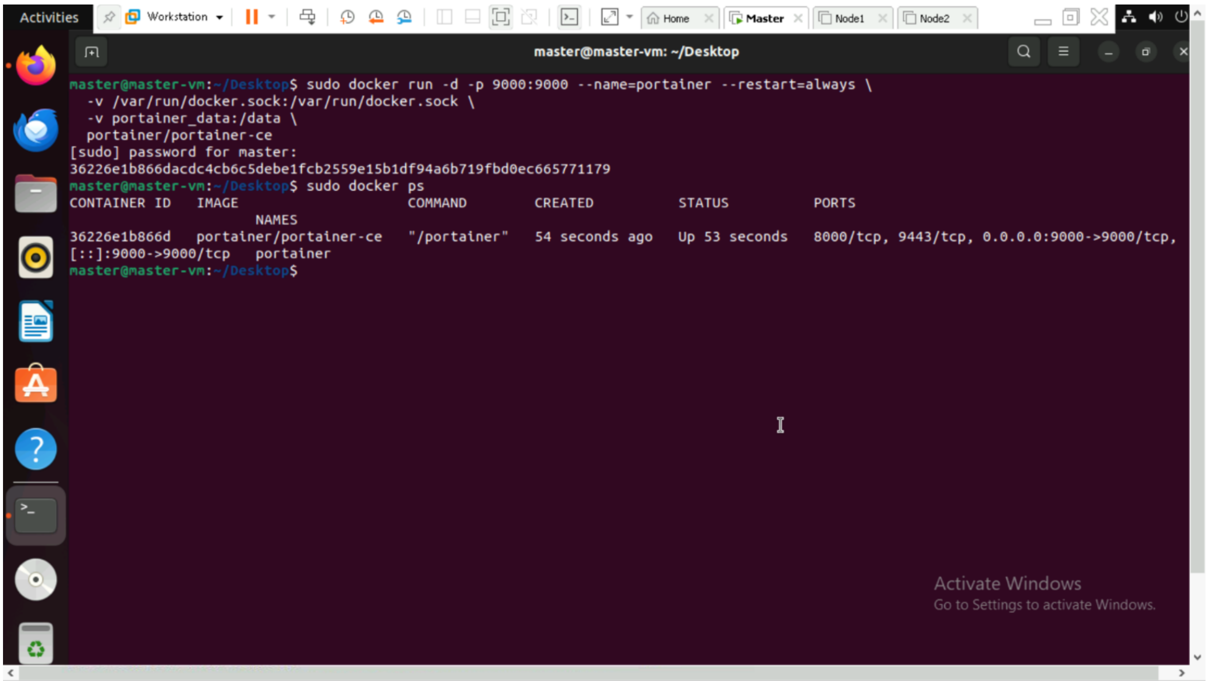
2. DOCKERS AND NGINX



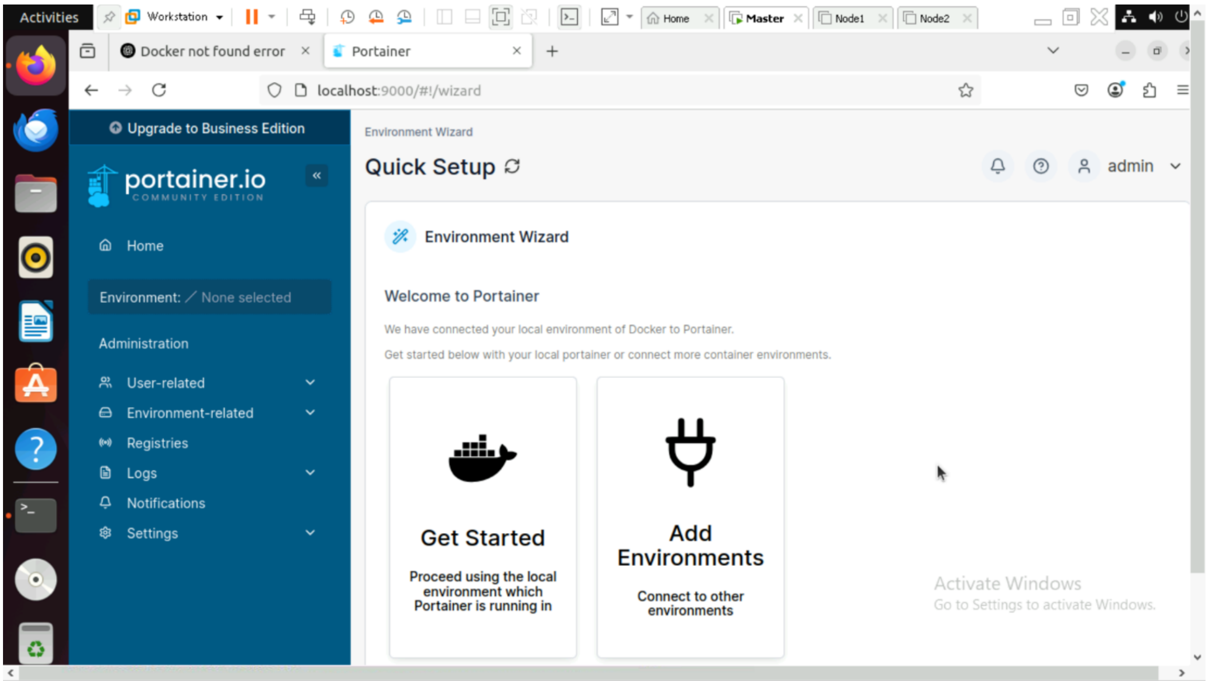
```
master@master-vm:~/Desktop$ sudo chmod a+r /etc/apt/keyrings/docker.asc
master@master-vm:~/Desktop$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.do
cker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
master@master-vm:~/Desktop$ sudo apt update
Get:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:3 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [45.7 kB]
Hit:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 94.5 kB in 1s (72.1 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
20 packages can be upgraded. Run 'apt list --upgradable' to see them.
master@master-vm:~/Desktop$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-ce-rootless-extras libslirp0 slirp4netns
Suggested packages:
  cgroupfs-mount | cgroup-lite
The following packages will be REMOVED:
  containerd docker.io runc
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin libslirp0 slirp4netns
0 upgraded, 8 newly installed, 3 to remove and 20 not upgraded.
Need to get 120 MB of archives.
After this operation, 138 MB of additional disk space will be used.
Get:1 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.7.25-1 [29.6 MB]
```



```
Unpacking libslirp0:amd64 (4.6.1-1build1) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../7-slirp4netns_1.0.1-2_amd64.deb ...
Unpacking slirp4netns (1.0.1-2) ...
Setting up docker-buildx-plugin (0.21.1-1~ubuntu.22.04~jammy) ...
Setting up containerd.io (1.7.25-1) ...
Setting up docker-compose-plugin (2.33.1-1~ubuntu.22.04~jammy) ...
Setting up docker-ce-cli (5:28.0.1-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up docker-ce-rootless-extras (5:28.0.1-1~ubuntu.22.04~jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:28.0.1-1~ubuntu.22.04~jammy) ...
Could not execute systemctl:  at /usr/bin/deb-systemd-invoke line 142.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.9) ...
master@master-vm:~/Desktop$ sudo systemctl start docker
master@master-vm:~/Desktop$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
master@master-vm:~/Desktop$ sudo docker --version
Docker version 28.0.1, build 068a01e
master@master-vm:~/Desktop$ sudo usermod -aG docker $USER
master@master-vm:~/Desktop$ newgrp docker
master@master-vm:~/Desktop$ sudo docker pull portainer/portainer-ce
Using default tag: latest
latest: Pulling from portainer/portainer-ce
436768c74267: Pull complete
d61825c69234: Pull complete
04de093ad5ed: Pull complete
a528983d077c: Pull complete
26eb502a78ed: Pull complete
b2724536dfda: Pull complete
5b45cfb2ea0c: Pull complete
20b115ea6339: Pull complete
8e73efb50b28: Pull complete
```


Activities
Workstation
Home
Master
Node1
Node2
master@master-vm: ~/Desktop
Activate Windows
Go to Settings to activate Windows.

```
master@master-vm:~/Desktop$ sudo docker run -d -p 9000:9000 --name=portainer --restart=always \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -v portainer_data:/data \
  portainer/portainer-ce
[sudo] password for master:
36226e1b866dacdc4cb6c5debe1fcb2559e15b1df94a6b719fbd0ec665771179
master@master-vm:~/Desktop$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND        CREATED          STATUS          PORTS
                     NAMES
36226e1b866d   portainer/portainer-ce   "/portainer"   54 seconds ago   Up 53 seconds   8000/tcp, 9443/tcp, 0.0.0.0:9000->9000/tcp,
[::]:9000->9000/tcp   portainer
master@master-vm:~/Desktop$
```



Environment Wizard

Welcome to Portainer

We have connected your local environment of Docker to Portainer.

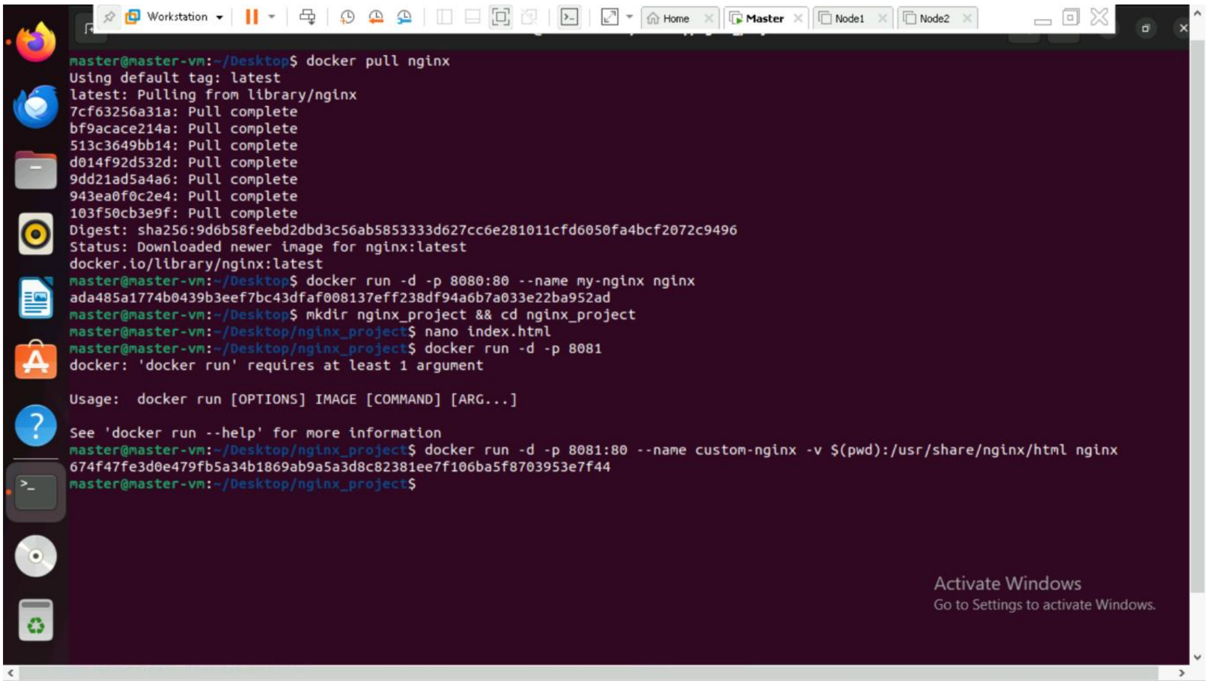
Get started below with your local portainer or connect more container environments.

Get Started

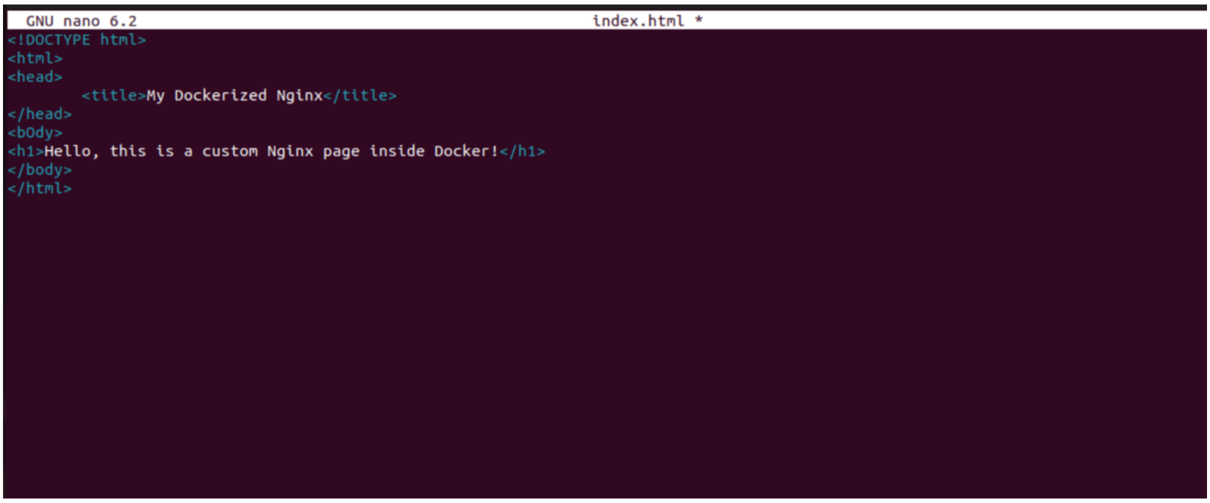
Proceed using the local environment which Portainer is running in

Add Environments

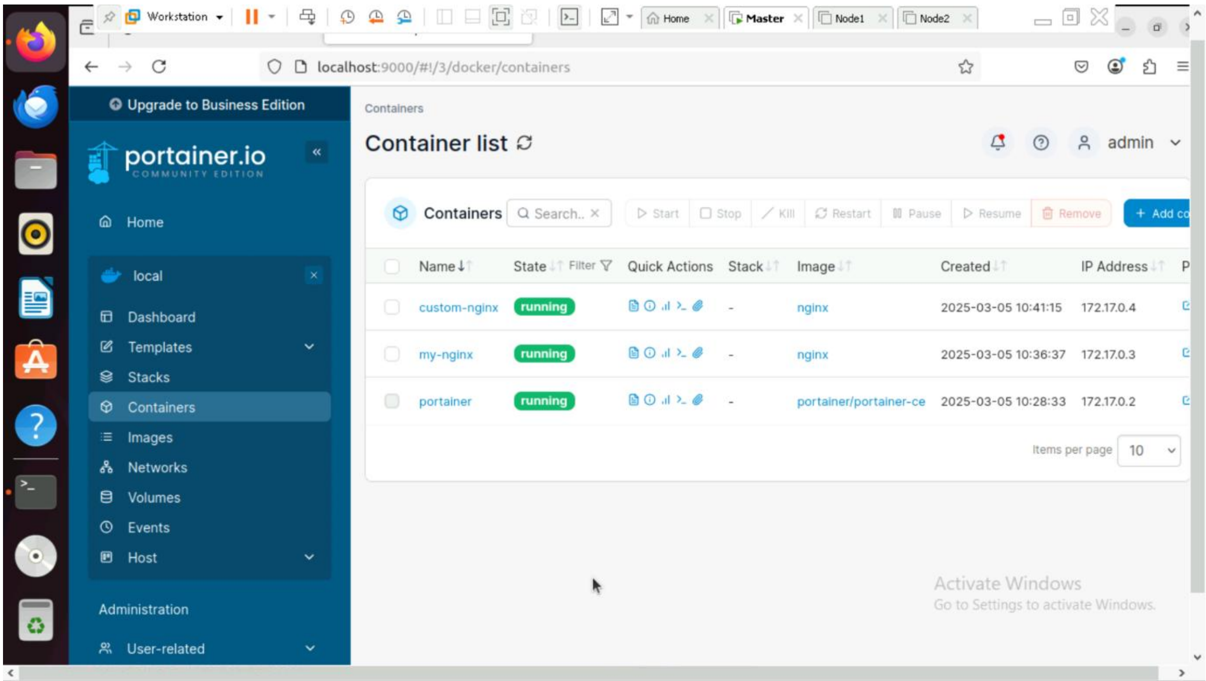
Connect to other environments



```
master@master-vm:~/Desktop$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
master@master-vm:~/Desktop$ docker run -d -p 8080:80 --name my-nginx nginx
ada485a1774b0439b3eef7bc43dfaf008137eff238df94a6b7a033e22ba952ad
master@master-vm:~/Desktop$ mkdir nginx_project && cd nginx_project
master@master-vm:~/Desktop/nginx_project$ nano index.html
master@master-vm:~/Desktop/nginx_project$ docker run -d -p 8081
docker: 'docker run' requires at least 1 argument

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

See 'docker run --help' for more information
master@master-vm:~/Desktop/nginx_project$ docker run -d -p 8081:80 --name custom-nginx -v $(pwd):/usr/share/nginx/html nginx
674f47fe3d0e479fb5a34b1869ab9a5a3d8c82381ee7f106ba5f8703953e7f44
master@master-vm:~/Desktop/nginx_project$
```



```
  GNU nano 6.2                                          index.html *
<!DOCTYPE html>
<html>
<head>
        <title>My Dockerized Nginx</title>
</head>
<body>
<h1>Hello, this is a custom Nginx page inside Docker!</h1>
</body>
</html>
```



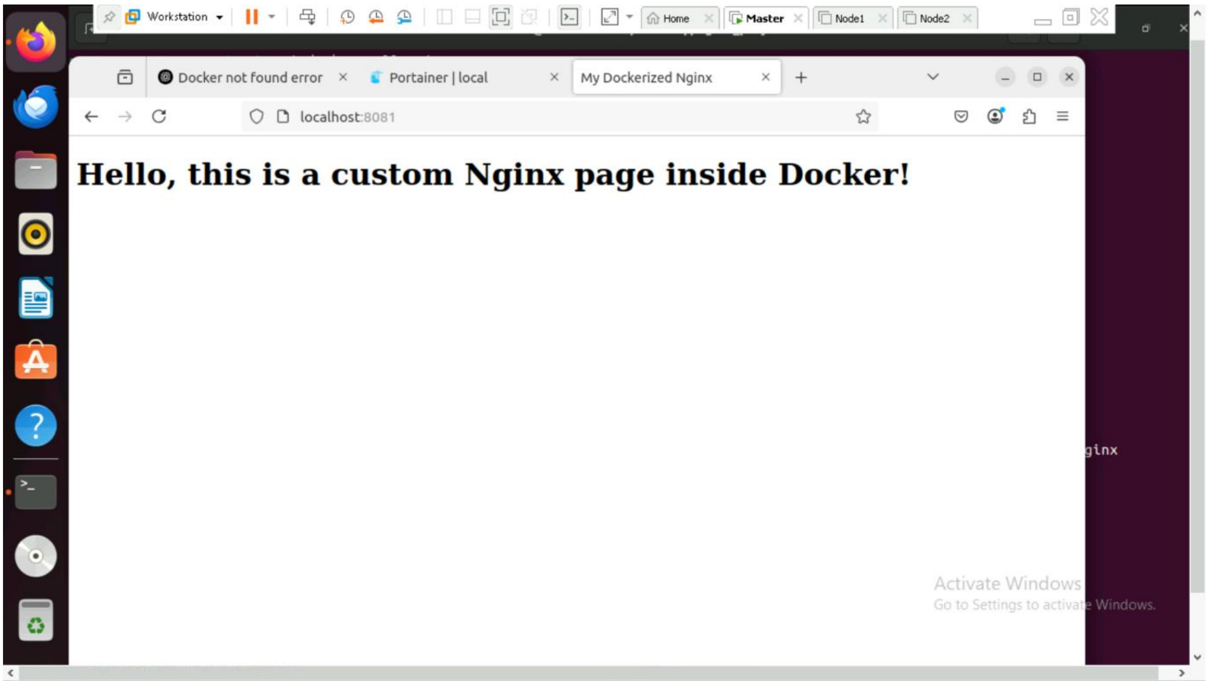
Containers

Container list

Containers Search..

Name	State	Quick Actions	Stack	Image	Created	IP Address
custom-nginx	running		-	nginx	2025-03-05 10:41:15	172.17.0.4
my-nginx	running		-	nginx	2025-03-05 10:36:37	172.17.0.3
portainer	running		-	portainer/portainer-ce	2025-03-05 10:28:33	172.17.0.2

Items per page 10



Hello, this is a custom Nginx page inside Docker!