

Use Case Document: Docker Custom Network (mynetwork)

Use Case Name: Docker Custom Network Setup and Verification

Description

This use case demonstrates creating a custom Docker network named mynetwork, deploying two BusyBox containers (app1 and app2) into the network, verifying their existence, checking connectivity between them (ping), and optionally verifying port access **port 80**

Preconditions

1. Docker Engine installed and running.
2. Python environment configured for running unit tests.
3. No existing containers named app1 or app2.
4. No existing network named mynetwork.

Postconditions

- mynetwork exists and both containers (app1, app2) are attached.
- Connectivity and port checks completed successfully.

Main Flow (Docker CLI Steps)

Step	Actions to be performed
1	Cleanup old containers and network
2	Create custom network according to the requirement
3	Run BusyBox containers in a network
4	Verify containers are running
5	Check connectivity (ping)
6	Inspect network
7	Verify port access

Notes / Pitfalls

- Containers must be running before inspecting networks.
- Docker inspect output must be parsed as JSON to reliably check connections.
- Ping relies on ICMP; may behave differently on Windows vs Linux.
- Containers can belong to multiple networks; the test checks only for the presence of mynetwork.
- Tests should include exception handling for robust automation.

Steps to run the projects

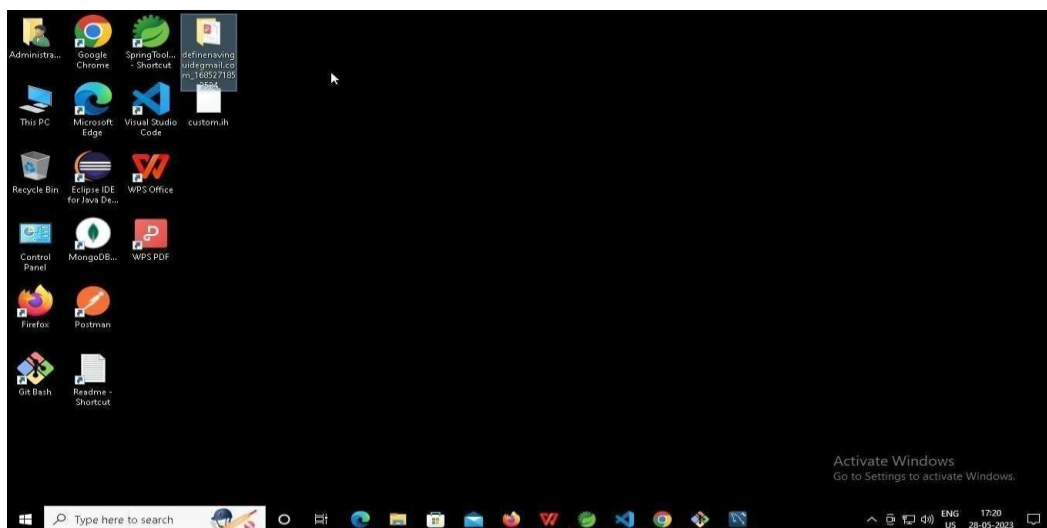
1. Open CMD type the command **pip install docker** .
2. Then use the command **pip install requests**
3. After the application is deployed your application you will need to run the testcases
4. In CMD run the python file run_tests.py
5. Wait for the python file to run the testcase
6. Next step is push the code to github

Steps to push the code

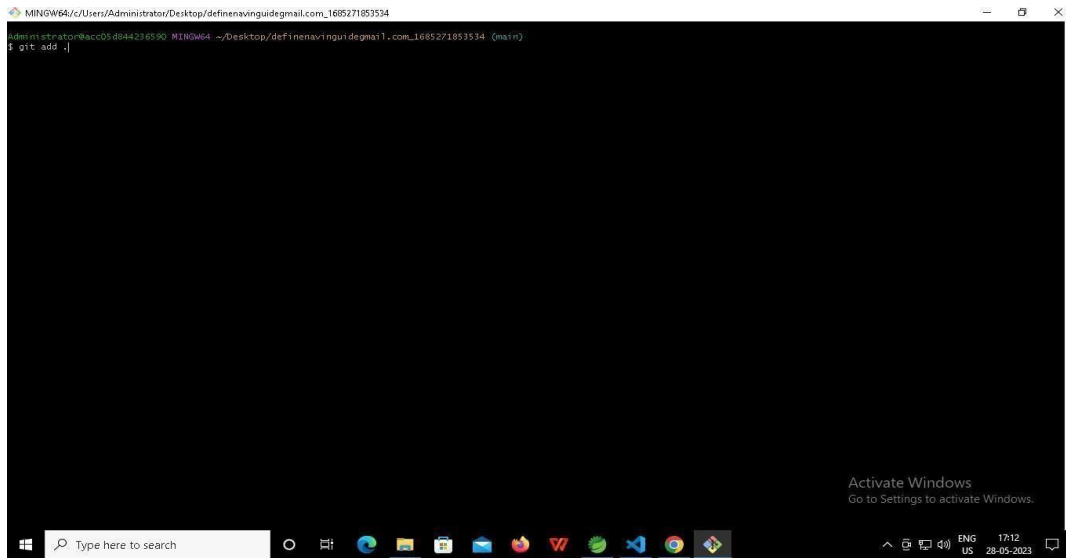
Push the code the repository

You can run test cases as many numbers of times and at any stage of Development, to check how many test cases are passed/failed and accordingly refactor your code.

1. **Make sure before final submission you commit all changes to git.** For that open the project folder available on desktop

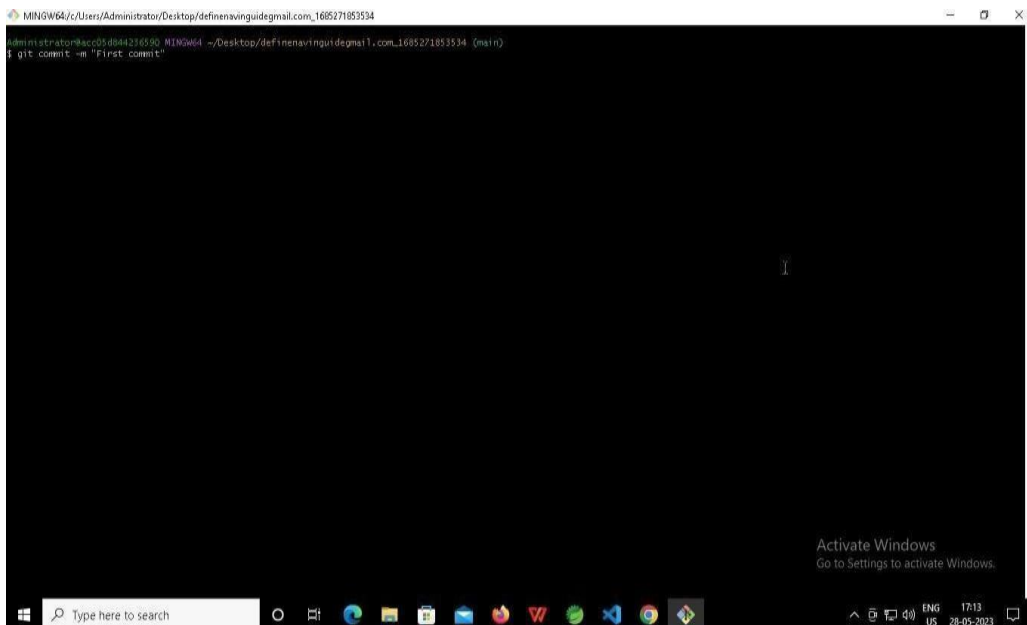


- a. Right click in folder and open Git Bash



e . git commit -m "First commit"

(You can provide any message every time you commit)



F.git push

```
MIN9W6A/c/Users/Administrator/Desktop/definensavguidemail.com_1695271853534 (main)
$ git push
Enumerating objects: 28156, done.
Counting objects: 100% (28156/28156), done.
Delta compression using up to 4 threads
Compressing objects: 100% (15330/15330), done.
Writing objects: 100% (28151/28151), 71.00 MiB | 3.27 MiB/s, done.
Total 28151 (delta 7783), reused 23848 (delta 7780), pack-reused 0
remote: Resolving deltas: 100% (7783/7783), completed with 3 local objects.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File Frontend-Angular/angular/cache/315256/angular-webpack/9c29720b052dd6ef081cb691ce119f5ada5942/D.pack is 86.24 MB; this is larger than GitHub's recommended maximum file
size of 50.00 MB
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/definensavguidemail.com_1695271853534.git
  490c626b..c102266d  main -> main
administrator@acc05d844236590: MIN9W6A - ~/Desktop/definensavguidemail.com_1695271853534 (main)
$
```

Activate Windows
Go to Settings to activate Windows.

Type here to search

ENG US 17:15 28-05-2023