

# Docker Mini Project

GitHub Repository : [Link](#)

## Part A: Application & Repository Setup

### Step 1: Create GitHub Repository, Clone it & Create the Application

The screenshot shows a GitHub repository page for 'Docker-mini-project'. The repository is public and has 0 stars. It features sections for 'Set up GitHub Copilot' and 'Add collaborators to this repository'. Below these, there's a 'Quick setup' section with instructions for cloning the repository via 'Set up in Desktop' or 'HTTPS' or 'SSH' (with the URL 'git@github.com:Abbaskashim/Docker-mini-project.git'). It also provides a command-line guide for creating a new repository:

```
echo "# Docker-mini-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:Abbaskashim/Docker-mini-project.git
git push -u origin main
```

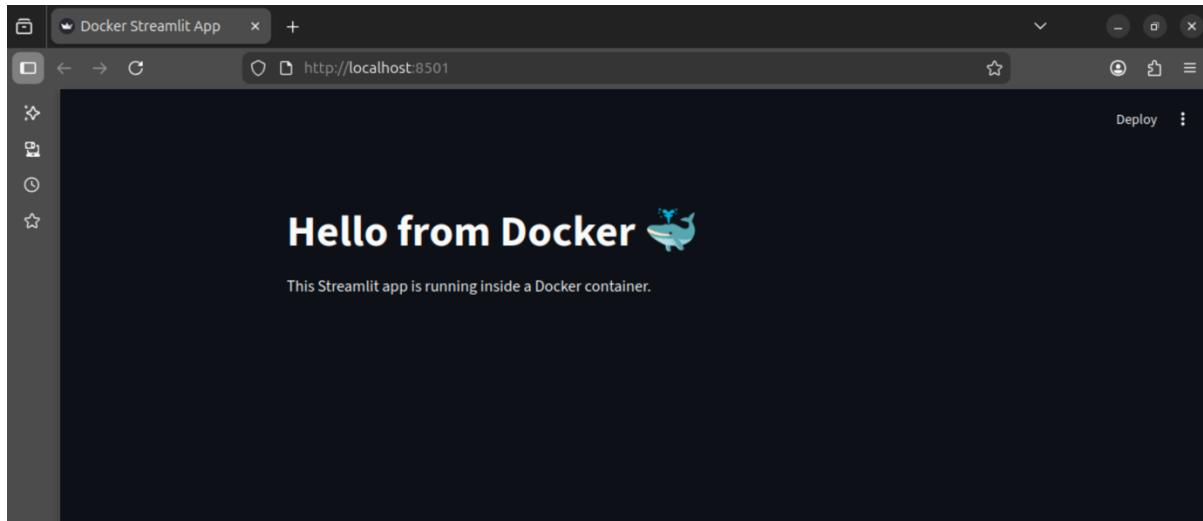
```
abbaskashim@abbakashim:~$ cd /home/abbaskashim/Desktop
abbaskashim@abbakashim:~/Desktop$ ls
app.py Minn.txt mybash.sh New operator.sh Project
abbaskashim@abbakashim:~/Desktop$ git clone git@github.com:Abbaskashim/Docker-mini-project.git
Cloning into 'Docker-mini-project'...
warning: You appear to have cloned an empty repository.
abbaskashim@abbakashim:~/Desktop$ ls
app.py Docker-mini-project Minn.txt mybash.sh New operator.sh Project
abbaskashim@abbakashim:~/Desktop$ cd Docker-mini-project
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ ls
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ nano.py
nano.py: command not found
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ nano app.py
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ 
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.20.10.3:8501

^X^C Stopping...
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$
```

## Output:



## Push Code to GitHub

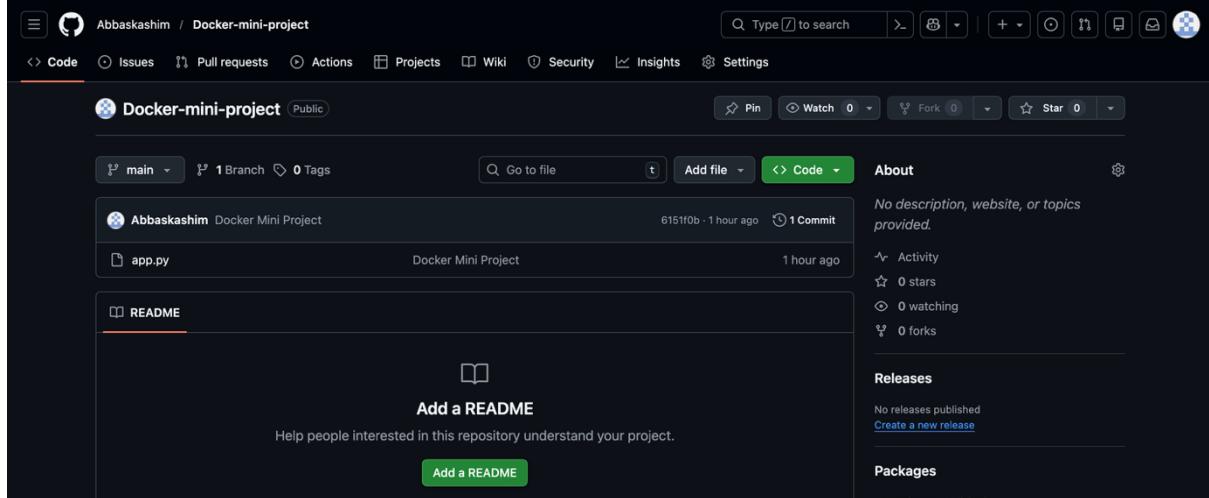
```
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ 
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ 
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git add .
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Dockerfile
    new file:   app.py

abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git config --global user.mail "abbaskashimsardhar@gmail.com"
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git commit -m "Docker Mini Project"
[main (root-commit) 666ef24] Docker Mini Project
 2 files changed, 20 insertions(+)
  create mode 100644 Dockerfile
  create mode 100644 app.py
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 568 bytes | 568.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Abbaskashim/Docker-mini-project.git
 * [new branch]      main -> main
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$
```

## Output:



## Part B: Dockerfile Creation & Image Build

### Step 2: Create Dockerfile

Write a valid Dockerfile

```
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ nano Dockerfile
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$
```

### Step 3: Build & Run Docker Image

docker build -t <image\_name> .

```
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ sudo -i
[sudo] password for abbaskashim:
root@abbakashim:~# cd /home/abbaskashim/Desktop/Docker-mini-project
root@abbakashim:/home/abbaskashim/Desktop/Docker-mini-project# docker build -t streamlit-docker-app .
[+] Building 2.7s (12/12) FINISHED                                            docker:default
=> [internal] load build definition from Dockerfile                      0.0s
=> => transferring dockerfile: 283B                                         0.0s
=> [internal] load metadata for docker.io/library/ubuntu:24.04            2.6s
=> [internal] load .dockerignore                                           0.0s
=> => transferring context: 2B                                           0.0s
=> [1/7] FROM docker.io/library/ubuntu:24.04@sha256:cd1dba651b3080c3686e  0.0s
=> [internal] load build context                                         0.0s
=> => transferring context: 30.78kB                                       0.0s
=> CACHED [2/7] RUN apt update                                           0.0s
=> CACHED [3/7] RUN apt install python3 -y                            0.0s
=> CACHED [4/7] RUN apt install python3-pip -y                         0.0s
=> CACHED [5/7] RUN pip3 install streamlit --break-system-packages    0.0s
=> CACHED [6/7] WORKDIR /app                                           0.0s
=> [7/7] COPY . .                                                 0.0s
=> => exporting to image                                              0.0s
=> => exporting layers                                              0.0s
=> => writing image sha256:326e66c30559867882ad676aef7d96bea6c010f19cba 0.0s
=> => naming to docker.io/library/streamlit-docker-app                  0.0s
root@abbakashim:/home/abbaskashim/Desktop/Docker-mini-project#
```

## docker images

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker images
Info → U In Use
IMAGE           ID          DISK USAGE   CONTENT SIZE   EXTRA
streamlit-docker-app:01  326e66c30559    1.1GB        0B
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project#
```

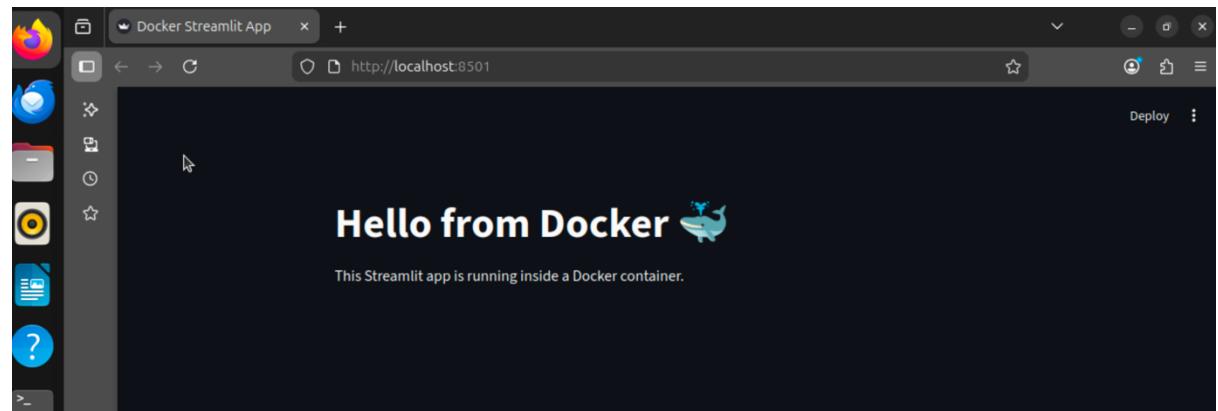
```
docker run -it -p 8501:8501 streamlit-app:01
```

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker run -it -p
8501:8501 streamlit-docker-app:01

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.17.0.2:8501
External URL: http://152.57.103.200:8501
```



```
docker ps
```

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      NAMES
STATUS         PORTS
c8a71044f2be  streamlit-docker-app:01 "streamlit run app.p..."  30 seconds ago
Up 29 seconds  0.0.0.0:8501->8501/tcp, [::]:8501->8501/tcp  streamlit-conta
iner
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project#
```

## Part C: Docker Command Explanation

### **Step 4: Explain All Docker Commands Used**

**docker build -t streamlit-docker-app:01 .**

- docker build → Builds Docker image
- -t → Tags the image
- streamlit-docker-app:01 → Image name
- . → Current directory (Dockerfile location)

**docker run -it -p 8501:8501 streamlit-app:01**

- docker run → Creates & starts container
- -it → Commonly used together to run the container **in the foreground with terminal access.**
- -p 8501:8501 → Port mapping
- Host: 8501
- Container: 8501
- streamlit-docker-app → Image name
- 01 → Image tag

**docker ps**

- Lists running containers

**docker images**

- Lists available Docker images

## Part D: Docker Compose Implementation

### **Step 5: Create Docker Compose File**

Create docker-compose.yml

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# nano docker-compose.yml
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project#
```

### **Step 6: Run Application Using Docker Compose**

#### **Mandatory Commands:**

docker-compose up -build

docker-compose ps

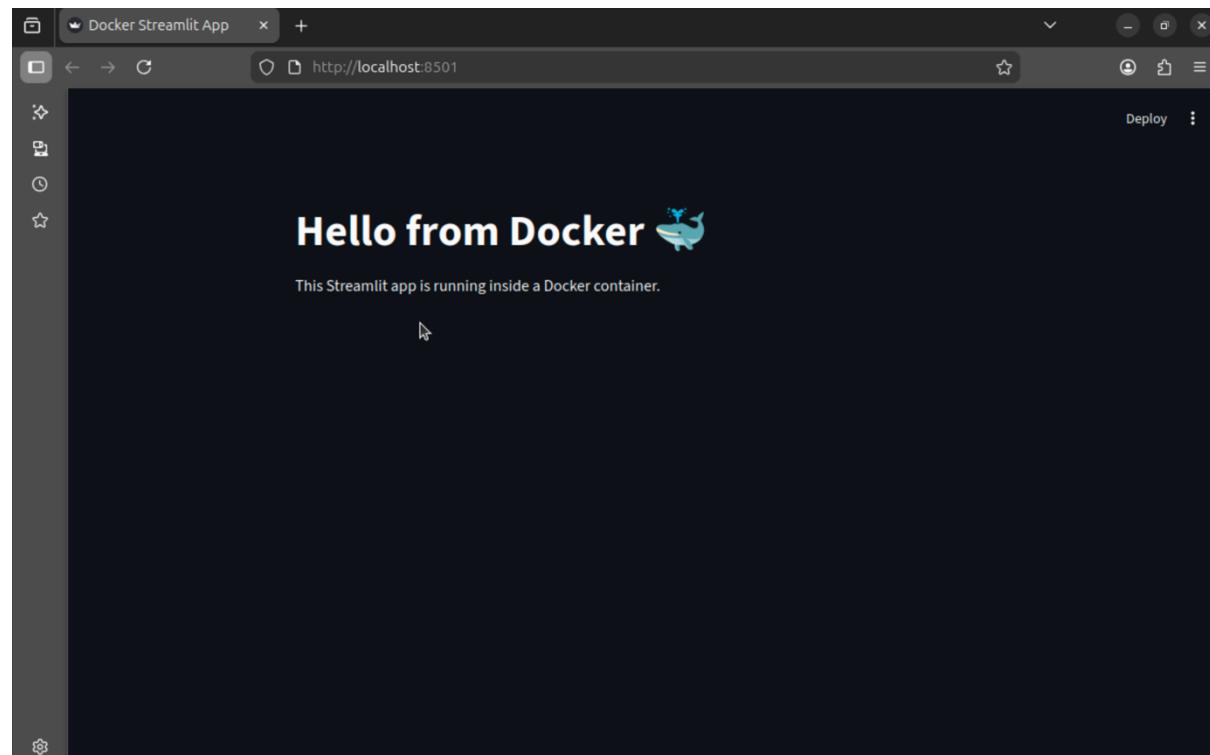
docker-compose logs

docker-compose down

## docker-compose up --build

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker-compose up --build
Building streamlit-app
[+] Building 1.0s (12/12) FINISHED
  => [internal] load build definition from Dockerfile
  => transferring dockerfile: 283B
  => [internal] load metadata for docker.io/library/ubuntu:24.04
  => [internal] load .dockerrcignore
  => transferring context: 2B
  => [1/7] FROM docker.io/library/ubuntu:24.04@sha256:cd1dba651b3080c3686ecf4e3c4220f026b521fb769788
  => [internal] load build context
  => => transferring context: 2.12kB
  => CACHED [2/7] RUN apt update
  => CACHED [3/7] RUN apt install python3 -y
  => CACHED [4/7] RUN apt install python3-pip -y
  => CACHED [5/7] RUN pip3 install streamlit --break-system-packages
  => CACHED [6/7] WORKDIR /app
  => CACHED [7/7] COPY . .
  => exporting to image
  => => exporting layers
  => => writing image sha256:759ca686e9f1ef6f841dd9c2d7a419a566af1f2f819d89a4ad726790797873ac
  => => naming to docker.io/library/streamlit-app:01
streamlit_container is up-to-date
Attaching to streamlit_container
streamlit_container |
streamlit_container |   Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.
streamlit_container |
streamlit_container |   You can now view your Streamlit app in your browser.
streamlit_container |
streamlit_container |   Local URL: http://localhost:8501
streamlit_container |   Network URL: http://172.18.0.2:8501
streamlit_container |   External URL: http://49.207.153.104:8501
```

## Output:



### **docker-compose ps**

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker-compose ps
      Name           Command    State    Ports
-----+-----+-----+-----+
streamlit_container   streamlit run app.py    Up      0.0.0.0:8501-
          --ser ...
                                         >8501/tcp,:::8501-
                                         >8501/tcp
```

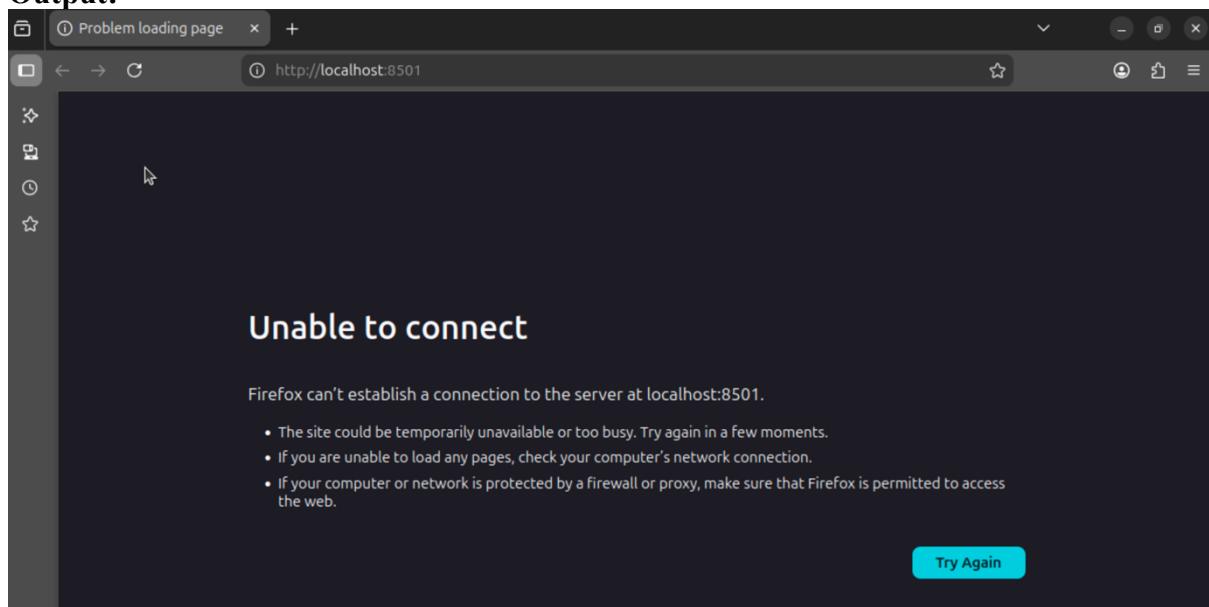
### **docker-compose logs**

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker-compose logs
Attaching to streamlit_container
streamlit_container |
streamlit_container | Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.
streamlit_container |
streamlit_container |
streamlit_container | You can now view your Streamlit app in your browser.
streamlit_container |
streamlit_container | Local URL: http://localhost:8501
streamlit_container | Network URL: http://172.18.0.2:8501
streamlit_container | External URL: http://49.207.153.104:8501
streamlit_container |
```

### **docker-compose down**

```
root@abbakashim:/home/abbakashim/Desktop/Docker-mini-project# docker-compose down
Stopping streamlit_container ... done
Removing streamlit_container ... done
Removing network docker-mini-project_default
```

### **Output:**



Now I have pushed all the files to GitHub Repository.

```
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git add .
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  Dockerfile
    new file:  docker-compose.yml

abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git commit -m "Docker Mini Project"
[main 74b6c99] Docker Mini Project
 2 files changed, 23 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 docker-compose.yml
abbaskashim@abbakashim:~/Desktop/Docker-mini-project$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 608 bytes | 152.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Abbaskashim/Docker-mini-project.git
 6151f0b..74b6c99  main -> main
```

## Output:

The screenshot shows the GitHub repository page for 'Docker-mini-project'. The repository is public and owned by 'Abbaskashim'. It has 1 branch (main) and 0 tags. The repository was created by 'Abbaskashim' 1 minute ago. There are 2 commits in the main branch, both of which were made by 'Abbaskashim' 1 minute ago. The commits are 'Dockerfile' and 'app.py'. The 'About' section indicates no description, website, or topics provided. The 'Activity' section shows 0 stars, 0 watching, and 0 forks. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Languages' section shows Dockerfile at 57.1% and Python at 42.9%. A 'README' section is present with a button to 'Add a README'.

## **Part F: Dockerfile vs Docker Compose Comparison**

Dockerfile and Docker Compose are both essential Docker tools, but they serve **different purposes** in the containerization workflow.

### **Comparison Table**

Aspect	Dockerfile	Docker Compose
Purpose	Used to create a Docker image by defining step-by-step build instructions	Used to run and manage multiple containers as services
Usage	Builds images using the docker build command	Runs services using docker-compose up
Complexity	Low – focuses only on image creation	Medium – manages services, networking, volumes
Version Control	Yes – stored as <b>Dockerfile</b> in Git repositories	Yes – stored as <b>docker-compose.yml</b> in Git repositories
Use Case	Best for single-container applications	Best for multi-container systems (app + DB, etc.)