

## **Task 1**

Go through the deck and make sure you are comfortable with Docker commands.

## **Task 2**

Go through the Get Started tutorial: <https://docs.docker.com/get-started/>

## **Task 3**

Learn Docker best-practices added to Appendix

## **Task 4**

Run Sample Application in docker container

<https://github.com/elisska/devops-helpers/tree/main/sample-app>

## **Task 5**

Visualize another data file with Sample application (use bind mounts & entrypoint & cmd combination)

## **Task 6**

Run 2 containers with volume and prove data sharing

## **Task 7**

Run 2 containers in the same network and prove communication between these containers

## **Task 8**

Self-learning: Understand what is multi-stage build:

<https://docs.docker.com/develop/develop-images/multistage-build/>

*For tasks 4, 5, 6, 7 provide screenshots*

*For task 8 provide link to your Github repo with application and Dockerfile + link to image pushed to DockerHub*

# Task 4

## Add .gitignore

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % touch .gitignore
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % nano .gitignore
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % git staus
git: 'staus' is not a git command. See 'git --help'.

The most similar command is
    status
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % git add .gitignore
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % git commit -m "Add .gitignore with rules for *.md and Dockerfile*"
[main 3f226a7] Add .gitignore with rules for *.md and Dockerfile*
  1 file changed, 3 insertions(+)
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 10 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 619 bytes | 619.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/idgafd/UCU_DevoPs_Course.git
  d5afab3..3f226a7  main -> main
```

I won't demonstrate the interactive method of building an image, because it is far from the best option in real development, just believe me, please, that I have already done it once 😊

## Create project structure

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % mkdir sample-app
anastasiiamazur@bttrm-amazur-pro16 UCU_DevoPs_Course % cd sample-app
anastasiiamazur@bttrm-amazur-pro16 sample-app % touch Dockerfile README.md GraphAnalysis.py
anastasiiamazur@bttrm-amazur-pro16 sample-app % ls
Dockerfile  GraphAnalysis.py  obj_dependency_data.csv  README.md
anastasiiamazur@bttrm-amazur-pro16 sample-app % cat Dockerfile
FROM python:3.7-slim
WORKDIR /app
RUN groupadd -r webservice && useradd --no-log-init -r -g webservice webservice
RUN pip install networkx dash plotly
COPY .
USER webservice:webservice
EXPOSE 8050
ENTRYPOINT ["python", "GraphAnalysis.py"]
CMD ["obj_dependency_data.csv"]
```

Used ENTRYPOINT + CMD to allow customization of arguments.

## Build Docker Image

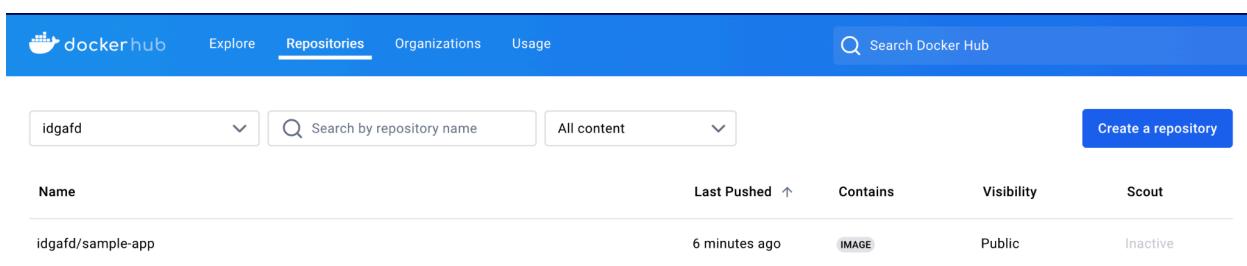
```
anastasiamazur@bttrm-amazur-pro16 sample-app % docker build -f Dockerfile -t idgafd/sample-app:dockerfile-method .  
[+] Building 22.3s (11/11) FINISHED  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 608B  
=> [internal] load metadata for docker.io/library/python:3.7-slim  
=> [auth] library/python:pull token for registry-1.docker.io  
=> [internal] load .dockerrignore  
=> => transferring context: 2B  
=> [1/5] FROM docker.io/library/python:3.7-slim@sha256:b53f496ca43e5af6994f8e316cf03af31050bf7944e0e4a308ad86c001cf028b docker:desktop-linux  
=> => resolve docker.io/library/python:3.7-slim@sha256:b53f496ca43e5af6994f8e316cf03af31050bf7944e0e4a308ad86c001cf028b 0.0s  
=> => sha256:f4e74734688847a18b13767a3a003547ed73c00a939759b369db4e08ef328ed9 11.32MB / 11.32MB 2.2s  
=> => sha256:e468929dc1a9cb521e4f19e37d59ec75f21538a41b2650d74e2163e877578f84 3.13MB / 3.13MB 0.0s  
=> => sha256:755c9c80f96367902e207fb8ea419a97d56a02867e5c5798da6258c1b2370b98 244B / 244B 0.0s  
=> => sha256:bc3b0687866d8779be6da27119f80a03882efc62ba3540c29925586d24e26a2e 3.32MB / 3.32MB 0.0s  
=> => sha256:e886f0f47ef56fcadb6ecaf2116056bbdb273e0fe07ed034498b198d386c04e 29.16MB / 29.16MB 1.4s  
=> => extracting sha256:e886f0f47ef56fcadb6ecaf2116056bbdb273e0fe07ed034498b198d386c04e 0.5s  
=> => extracting sha256:bc3b0687866d8779be6da27119f80a03882efc62ba3540c29925586d24e26a2e 0.1s  
=> => extracting sha256:f4e74734688847a18b13767a3a003547ed73c00a939759b369db4e08ef328ed9 0.2s  
=> => extracting sha256:755c9c80f96367902e207fb8ea419a97d56a02867e5c5798da6258c1b2370b98 0.0s  
=> => extracting sha256:e468929dc1a9cb521e4f19e37d59ec75f21538a41b2650d74e2163e877578f84 0.0s  
=> [internal] load build context  
=> => transferring context: 21.67kB 0.0s  
=> [2/5] WORKDIR /app 0.0s  
=> [3/5] RUN groupadd -r webservice && useradd --no-log-init -r -g webservice webservice 0.2s  
=> [4/5] RUN pip install networkx dash plotly 10.3s  
=> [5/5] COPY . . 0.1s  
=> exporting to image 6.5s  
=> => exporting layers 4.4s  
=> => exporting manifest sha256:1309485a1788d9e15e7b10fb92ed3a6a8caf0f91e30394ada1c273793200a7b5 0.0s  
=> => exporting config sha256:a1ca06d23c50a1e4e2e40574f2200094fbdfb2f61fe9760d73b3cd60ccb4a9d0 0.0s  
=> => exporting attestation manifest sha256:3fb3d326aba557d41bf0e6ac090518290a3f75c8f99fdace1a5baee16f9cc834a 0.0s  
=> => exporting manifest list sha256:8214389f152d59a2f30425b586674ed831a718bd4742e12f097fe6af83bbb9f4 0.0s  
=> => naming to docker.io/idgafd/sample-app:dockerfile-method 0.0s  
=> => unpacking to docker.io/idgafd/sample-app:dockerfile-method 2.2s
```

## Check it and push to DockerHub

```
anastasiamazur@bttrm-amazur-pro16 sample-app % docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
idgafd/sample-app	dockerfile-method	8214389f152d	23 seconds ago	559MB

```
anastasiamazur@bttrm-amazur-pro16 sample-app % docker push idgafd/sample-app:dockerfile-method  
  
The push refers to repository [docker.io/idgafd/sample-app]  
e468929dc1a9: Pushed  
7b5cdfca4f07: Pushed  
79dba863a808: Pushed  
f4e747346888: Pushed  
bc3b0687866d: Pushed  
e886f0f47ef5: Pushed  
755c9c80f963: Pushed  
c615b8efff37: Pushed  
ab605721b939: Pushed  
d67b4ca3aa4c: Pushed  
dockerfile-method: digest: sha256:8214389f152d59a2f30425b586674ed831a718bd4742e12f097fe6af83bbb9f4 size: 856
```



The screenshot shows the Docker Hub website. At the top, there's a blue header bar with the Docker Hub logo, navigation links for 'Explore', 'Repositories', 'Organizations', and 'Usage', and a search bar. Below the header, there's a search input field with the text 'Search by repository name' and a dropdown menu showing 'idgafd'. To the right of the search bar is a 'Create a repository' button. The main content area displays a table of repositories. The first row in the table has a dropdown arrow next to 'idgafd'. The second row shows the repository 'idgafd/sample-app' with details: 'Last Pushed' (6 minutes ago), 'Contains' (IMAGE), 'Visibility' (Public), and 'Scout' (Inactive). The table has columns for 'Name', 'Last Pushed', 'Contains', 'Visibility', and 'Scout'.

Name	Last Pushed	Contains	Visibility	Scout
idgafd/sample-app	6 minutes ago	IMAGE	Public	Inactive

Docker Hub interface showing the General tab for the repository `idgafd/sample-app`.

General Tab:

- Last pushed 24 minutes ago
- Add a description (incomplete)
- Add a category (incomplete)

Tags Section:

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
<code>dockerfile-method</code>		Image	24 minutes ago	24 minutes ago

[See all](#)

Docker Hub interface showing the details for the tag `dockerfile-method` of the repository `idgadf/sample-app`.

Image Details:

- INDEX DIGEST: sha256:8214389f152d59a2f30425b586674ed831a718bd4742e12f097fe6af83bb9f4
- OS/ARCH: linux/arm64/v8
- COMPRESSED SIZE: 103.19 MB
- LAST PUSHED: 41 minutes ago by `idgadf`
- TYPE: Image
- MANIFEST DIGEST: sha256:1309485a...

Image Layers Tab:

Command:

```
1 ADD file ... in / 27.81 MB ADD file:ec1a6e0aedd76c8fdc8544775f8b553f58950e9435f5cef919f39374e222cfbb in /
```

## Docker Desktop View

Docker Desktop interface showing the Images section.

Left Sidebar:

- Containers
- Images**
- Volumes
- Builds
- Docker Scout
- Extensions

Images Section:

Local Hub repositories

450.55 MB / 0 Bytes in use 1 images Last refresh: 34 minutes ago

Name	Tag	Image ID	Created	Size	Actions
<code>idgadf/sample-app</code>	<code>dockerfile-method</code>	<code>8214389f152d</code>	28 minutes ago	558.76 MB	

## Check Image history

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
8214389f152d	5 minutes ago	CMD ["obj_dependency_data.csv"]	0B	buildkit.dockerfile.v0
<missing>	5 minutes ago	ENTRYPOINT ["python" "GraphAnalysis.py"]	0B	buildkit.dockerfile.v0
<missing>	5 minutes ago	EXPOSE map[8050/tcp:{}]	0B	buildkit.dockerfile.v0
<missing>	5 minutes ago	USER webservice:webservice	0B	buildkit.dockerfile.v0
<missing>	5 minutes ago	COPY . . # buildkit	41kB	buildkit.dockerfile.v0
<missing>	5 minutes ago	RUN /bin/sh -c pip install networkx dash plo...	282MB	buildkit.dockerfile.v0
<missing>	5 minutes ago	RUN /bin/sh -c groupadd -r webservice && use...	41kB	buildkit.dockerfile.v0
<missing>	5 minutes ago	WORKDIR /app	8.19kB	buildkit.dockerfile.v0
<missing>	16 months ago	CMD ["python3"]	0B	buildkit.dockerfile.v0
<missing>	16 months ago	RUN /bin/sh -c set -eux; savedAptMark="\$(a...)	13.2MB	buildkit.dockerfile.v0
<missing>	16 months ago	ENV PYTHON_GET_PIP_SHA256=45a2bb8bf2bb5eff16...	0B	buildkit.dockerfile.v0
<missing>	16 months ago	ENV PYTHON_GET_PIP_URL=https://github.com/py...	0B	buildkit.dockerfile.v0
<missing>	16 months ago	ENV PYTHON_SETUPTOOLS_VERSION=57.5.0	0B	buildkit.dockerfile.v0
<missing>	16 months ago	ENV PYTHON_PIP_VERSION=23.0.1	0B	buildkit.dockerfile.v0
<missing>	16 months ago	RUN /bin/sh -c set -eux; for src in idle3 p...	20.5kB	buildkit.dockerfile.v0
<missing>	16 months ago	RUN /bin/sh -c set -eux; savedAptMark="\$(a...)	36.9MB	buildkit.dockerfile.v0
<missing>	16 months ago	ENV PYTHON_VERSION=3.7.17	0B	buildkit.dockerfile.v0
<missing>	16 months ago	ENV GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6...	0B	buildkit.dockerfile.v0
<missing>	16 months ago	RUN /bin/sh -c set -eux; apt-get update; a...	10.3MB	buildkit.dockerfile.v0
<missing>	16 months ago	ENV LANG=C.UTF-8	0B	buildkit.dockerfile.v0
<missing>	16 months ago	ENV PATH=/usr/local/bin:/usr/local/sbin:/usr...	0B	buildkit.dockerfile.v0
<missing>	14 months ago	/bin/sh -c #(nop) CMD ["bash"]	0B	buildkit.dockerfile.v0
<missing>	14 months ago	/bin/sh -c #(nop) ADD file:ec1a6e0aedd76c8fd...	108MB	

CMD: Runs obj\_dependency\_data.csv (0B)

ENTRYPOINT: Executes python GraphAnalysis.py (0B)

EXPOSE: Opens port 8050 for the app (0B)

USER: Sets non-root user webservice (0B)

COPY: Adds app files to container (41KB)

RUN {dependencies}: Installs networkx, dash, plotly (282MB)

RUN {user setup}: Creates user/group webservice (41KB).

WORKDIR: Sets working directory to /app (8.19KB)

Base Image (Python 3.7-slim): Adds Python environment (108MB)

Largest layer (obviously) is dependencies install (282MB). The good practices are used by setting up non-root user and slim base image. We will even make the process better with multi-stage builds to reduce image size.

## Docker Desktop View

Images / idgafd/sample-app:dockerfile-method

**idgafd/sample-app:dockerfile-method** IN USE  
8214389f152d ⚙️

CREATED 31 minutes ago SIZE 558.76 MB Recommended fixes Run ⚙️ 🗑️

Analyzed by docker scout

**Image hierarchy**

```

FROM debian:12-slim, 12.1-slim, bookworm-20230919-slim, bookworm-slim
  FROM python:3.7-slim, 3.7-slim-bookworm, 3.7.17-slim, 3.7.17-slim-bookworm
    ALL idgafd/sample-app:dockerfile-method
  
```

**Layers (23)**

↳ 10 ENV PYTHON_SETUPTOOLS_VERSION=57.5.0	0 B	✓
↳ 11 ENV PYTHON_GET_PIP_URL=https://github.com/pypa/get-pip/...	0 B	✓
↳ 12 ENV PYTHON_GET_PIP_SHA256=45a2bb8bf2bb5eff16fdd00fa...	0 B	✓
↳ 13 RUN /bin/sh -c set -eux; savedAptMark=\$(apt-mark showmanu...	13.2 MB	!
↳ 14 CMD ["python3"]	0 B	✓
↳ 15 WORKDIR /app	8.19 KB	✓
↳ 16 RUN /bin/sh -c groupadd -r webservice && useradd --no-log-init ...	40.96 KB	✓
↳ 17 RUN /bin/sh -c pip install networkx dash plotly # buildkit	282.43 MB	!
↳ 18 COPY . . # buildkit	40.96 KB	✓
↳ 19 USER webservice:webservice	0 B	✓
↳ 20 EXPOSE map[8050/tcp:{}]	0 B	✓
↳ 21 ENTRYPOINT ["python" "GraphAnalysis.py"]	0 B	✓
↳ 22 CMD ["obj_dependency_data.csv"]	0 B	✓

**Vulnerabilities (63)**

Package or CVE name	Fixable	Show excepted	Reset filters
debian/expat 2.5.0-1	2	1	0 0 0
debian/openssl 3.0.9-1	1	1	5 6 1
debian/krb5 1.20.1-2	1	1	1 0 0
debian/glibc 2.36-9+deb12u1	0	4	1 4 2
debian/gnuthash 3.7.9-2	0	2	3 0 0
setup tools 57.5.0	0	2	0 0 0
werkzeug 2.2.3	0	1	3 0 0
debian/systemd 252.12-1~deb12u1	0	1	1 0 1
debian/perl 5.36.0-7	0	1	0 0 0
debian/tar 1.34+dfsg-1.2	0	0	1 1 0

1–10 of 23 < >

**Packages (188)**

Package name	Package
debian/acl 2.3.1-3	debian/acl 2.3.1-3
debian/adduser 3.134	debian/adduser 3.134
debian/apt 2.6.1	debian/apt 2.6.1
debian/attr 1:2.5.1-4	debian/attr 1:2.5.1-4
debian/audit 1:3.0.9-1	debian/audit 1:3.0.9-1
debian/base-files 12.4+deb12u1	debian/base-files 12.4+deb12u1
debian/base-passwd 3.6.1	debian/base-passwd 3.6.1
debian/bash 5.2.15-2+b2	debian/bash 5.2.15-2+b2
debian/bsdtar 1:2.38.1-5+b1	debian/bsdtar 1:2.38.1-5+b1
debian/bzip2 1.0.8-5	debian/bzip2 1.0.8-5

1–10 of 188 < >

We can see multiple vulnerabilities in the base image (debian:12-slim) and inherited layers, including critical issues in packages like openssl and glibc. Potentially we could try to update the base image, patch vulnerable packages, and optimize dependencies to fix some of this stuff.

## Run a container

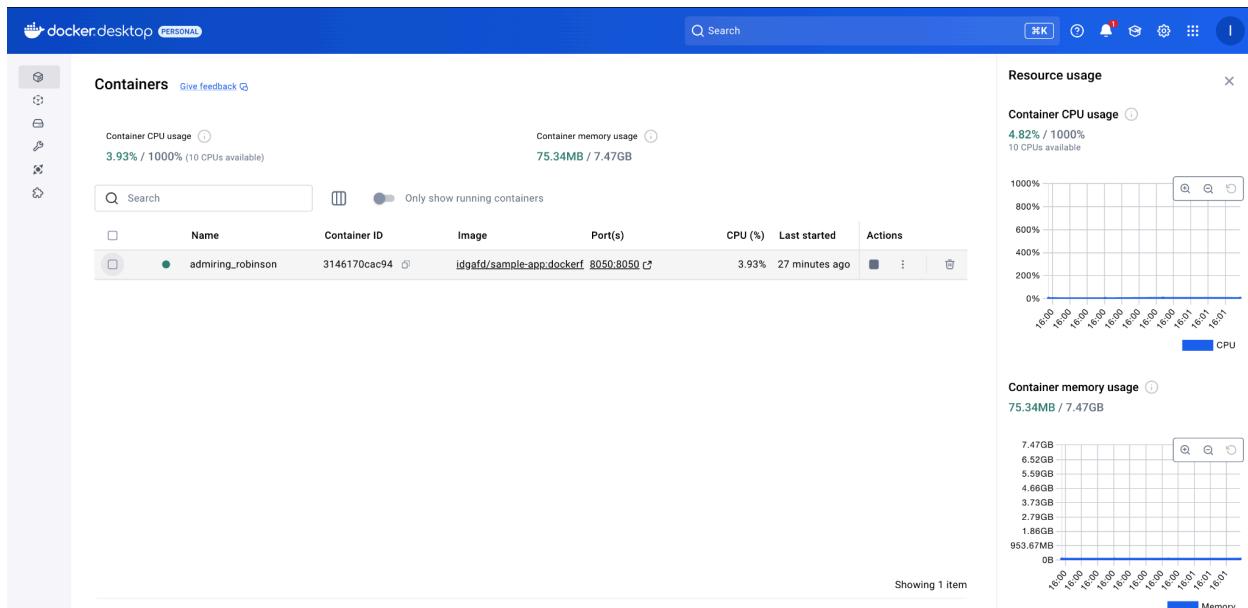
```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker run -it --rm -p 8050:8050 idgafdf/sample-app:dockerfile-method

Dash is running on http://0.0.0.0:8050/

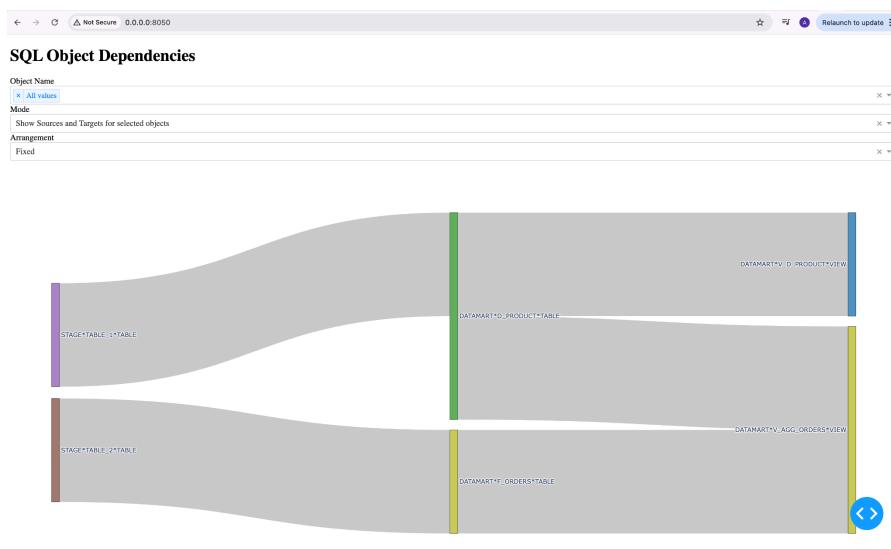
* Serving Flask app 'GraphAnalysis'
* Debug mode: on
```

- `-it`: provides an interactive terminal for the container
- `--rm`: automatically removes the container after it is stopped
- `-p 8050:8050`: binds port 8050 on my local machine to port 8050 on the container
- `--name`: didn't use that, because automatically created name are funnier 🎉

## Docker Desktop View



## Result Overview



# Task 5

## Create new .csv

```
[anastasiiamazur@bttrm-amazur-pro16 sample-app % touch funny_data.csv
anastasiiamazur@bttrm-amazur-pro16 sample-app % echo "SOURCE|TARGET|TYPE|SYSTEM|OBJECT|OBJECT_TYPE
COFFEE_MACHINE|CUP|ACTION|KITCHEN|V_FILL_CUP|SUCCESS
BROWSER|STACKOVERFLOW|SEARCH|INTERNET|V_QUESTION|LOOP
BED|USER|ACTION|BEDROOM|V_WAKE_UP|FAIL
FRIDGE|USER|SEARCH|KITCHEN|V_SNACK|SUCCESS
KEYBOARD|MONITOR|INPUT|DESK|V_TYPING|ERROR
USER|COFFEE_MACHINE|ACTION|KITCHEN|V_REFILL_BEANS|WARNING
CHAIR|USER|SUPPORT|OFFICE|V_SIT_DOWN|SUCCESS
IDE|CONSOLE|RUN|COMPUTER|V_DEBUG|FAIL
MOUSE|DESKTOP|ACTION|COMPUTER|V_CLICK|SUCCESS
BROWSER|CAT_VIDEO|SEARCH|INTERNET|V_WATCH|SUCCESS" > funny_data.csv
```

## Run a container

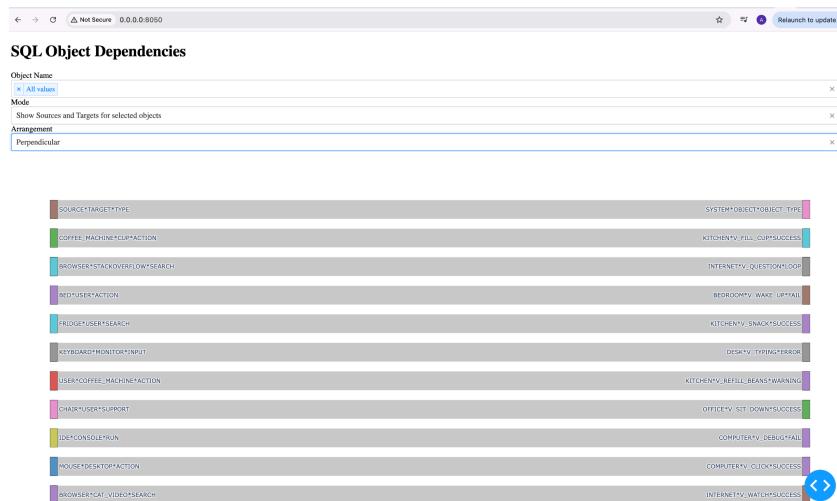
```
anastasiiamazur@bttrm-amazur-pro16 sample-app % docker run -it --rm -p 8050:8050 \
-v $(pwd)/funny_data.csv:/app/funny_data.csv \
idgafd/sample-app:dockerfile-method funny_data.csv
```

Dash is running on <http://0.0.0.0:8050>

```
* Serving Flask app 'GraphAnalysis'
* Debug mode: on
```

- `-v $(pwd)/funny_data.csv:/app/funny_data.csv`: mounts the file `funny_data.csv` from the current directory (`$(pwd)`) into the container at `/app/funny_data.csv`
- `funny_data.csv`: passes `funny_data.csv` as an argument to the container's entrypoint (`GraphAnalysis.py` in our case)

## Result Overview



# Task 6

For this example, I will run two containers using a simple image alpine.

## Create a shared volume

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker volume create shared_volume  
shared_volume
```

## Run the first container and mount the volume

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker run -it --rm -v shared_volume:/app --name container1 alpine  
Unable to find image 'alpine:latest' locally  
latest: Pulling from library/alpine  
9986a736f7d3: Download complete  
Digest: sha256:1e42bbe2508154c9126d48c2b8a75420c3544343bf86fd041fb7527e017a4b4a  
Status: Downloaded newer image for alpine:latest
```

- `-v shared_volume:/app:` mounts the shared\_volume to /app inside the container
- `--name container1:` names the first container

Docker couldn't find the alpine:latest image on my local system, so it automatically downloaded it from the Docker Hub, and now it is here 🎉

Images [Give feedback](#)

Local [Hub repositories](#)

0 Bytes / 0 Bytes in use 2 images Last refresh: 2 hours ago

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	idgafdf/sample-app	dockerfile-method	8214389f152d	2 hours ago	558.76 MB	<a href="#">▶</a> <a href="#">⋮</a> <a href="#">🔗</a>
<input type="checkbox"/>	alpine	latest	1e42bbe25081	3 months ago	13.57 MB	<a href="#">▶</a> <a href="#">⋮</a> <a href="#">🔗</a>

## Write data to the volume in container1 and exit

```
/ # echo "Hello from container1" > /app/shared_file.txt  
/ # exit
```

## Run the second container and mount the same volume

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker run -it --rm -v shared_volume:/app --name container2 alpine
```

## Read the data from the shared volume in container2 and exit

```
/ # cat /app/shared_file.txt  
Hello from container1  
[/ # exit
```

# Task 7

## Create a custom network

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker network create my_network  
4382748a73455aec3de2d3bf3deb82a0ef04b1810fa664792c64d7706ed651bc
```

## Run the first container in the network

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker run -d --name container1 --network my_network alpine sleep 3600  
d5b80f83c8bca74c40e52e0dc2211af7dc266343d116be4c737ced4009394c08
```

- -d: runs the container in detached mode
- --network my\_network: connects the container to the network

## Run the second container in the same network

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % docker run -it --rm --name container2 --network my_network alpine
```

## Test communication by ICMP Echo Request

```
/ # ping container1  
PING container1 (172.18.0.2): 56 data bytes  
64 bytes from 172.18.0.2: seq=0 ttl=64 time=0.881 ms  
64 bytes from 172.18.0.2: seq=1 ttl=64 time=0.619 ms  
64 bytes from 172.18.0.2: seq=2 ttl=64 time=0.476 ms  
64 bytes from 172.18.0.2: seq=3 ttl=64 time=0.214 ms  
64 bytes from 172.18.0.2: seq=4 ttl=64 time=0.123 ms  
64 bytes from 172.18.0.2: seq=5 ttl=64 time=0.351 ms  
64 bytes from 172.18.0.2: seq=6 ttl=64 time=0.754 ms  
64 bytes from 172.18.0.2: seq=7 ttl=64 time=0.467 ms  
64 bytes from 172.18.0.2: seq=8 ttl=64 time=55.039 ms  
64 bytes from 172.18.0.2: seq=9 ttl=64 time=0.606 ms  
64 bytes from 172.18.0.2: seq=10 ttl=64 time=0.074 ms  
64 bytes from 172.18.0.2: seq=11 ttl=64 time=0.194 ms  
64 bytes from 172.18.0.2: seq=12 ttl=64 time=0.266 ms  
64 bytes from 172.18.0.2: seq=13 ttl=64 time=0.817 ms  
64 bytes from 172.18.0.2: seq=14 ttl=64 time=0.350 ms  
64 bytes from 172.18.0.2: seq=15 ttl=64 time=0.364 ms  
64 bytes from 172.18.0.2: seq=16 ttl=64 time=0.166 ms  
64 bytes from 172.18.0.2: seq=17 ttl=64 time=0.416 ms  
64 bytes from 172.18.0.2: seq=18 ttl=64 time=0.352 ms  
64 bytes from 172.18.0.2: seq=19 ttl=64 time=0.835 ms  
64 bytes from 172.18.0.2: seq=20 ttl=64 time=0.422 ms  
64 bytes from 172.18.0.2: seq=21 ttl=64 time=0.390 ms  
64 bytes from 172.18.0.2: seq=22 ttl=64 time=0.302 ms  
64 bytes from 172.18.0.2: seq=23 ttl=64 time=0.438 ms  
^C  
--- container1 ping statistics ---  
24 packets transmitted, 24 packets received, 0% packet loss  
round-trip min/avg/max = 0.074/2.704/55.039 ms
```

# Task 8

## Create project structure

```
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % mkdir ml-app  
anastasiiamazur@bttrm-amazur-pro16 UCU_DevOps_Course % cd ml-app  
anastasiiamazur@bttrm-amazur-pro16 ml-app % touch app.py  
  
anastasiiamazur@bttrm-amazur-pro16 ml-app % touch requirements.txt  
  
anastasiiamazur@bttrm-amazur-pro16 ml-app % touch Dockerfile
```

## Build Docker Image

```
anastasiiamazur@bttrm-amazur-pro16 ml-app % docker build -t idgafd/ml-app:multistage .  
[+] Building 2.1s (16/16) FINISHED docker:desktop-linux  
=> [internal] load build definition from Dockerfile 0.0s  
=> transferring dockerfile: 1.08kB 0.0s  
=> [internal] load metadata for docker.io/library/python:3.9-slim 0.0s  
=> [internal] load .dockerrcignore 0.0s  
=> transferring context: 2B 0.0s  
=> [build 1/6] FROM docker.io/library/python:3.9-slim@sha256:4ee0613170ac55ebc693a03b6655a5c6f387126f6bc3390e739c2e6c337880ef 0.0s  
=> => resolve docker.io/library/python:3.9-slim@sha256:4ee0613170ac55ebc693a03b6655a5c6f387126f6bc3390e739c2e6c337880ef 0.0s  
=> [internal] load build context 0.0s  
=> => transferring context: 5.89kB 0.0s  
=> CACHED [build 2/6] WORKDIR /app 0.0s  
=> CACHED [build 3/6] COPY requirements.txt . 0.0s  
=> CACHED [build 4/6] RUN pip install --no-cache-dir -r requirements.txt 0.0s  
=> [build 5/6] COPY app.py . 0.0s  
=> [build 6/6] COPY static ./static 0.2s  
=> CACHED [stage-1 3/7] COPY --from=build /usr/local/lib/python3.9/site-packages /usr/local/lib/python3.9/site-packages 0.0s  
=> CACHED [stage-1 4/7] COPY --from=build /usr/local/bin /usr/local/bin 0.0s  
=> [stage-1 5/7] COPY app.py . 0.0s  
=> [stage-1 6/7] COPY static ./static 0.0s  
=> [stage-1 7/7] RUN groupadd -r webservice && useradd --no-log-init -r -g webservice webservice 0.2s  
=> exporting to image 0.0s  
=> => exporting layers 0.1s  
=> => exporting manifest sha256:a6ec6f28c5fb64063f79ecd700ad904af7331c7ca8c0d43433ac0f7400181b1 0.0s  
=> => exporting config sha256:d4a9e3302d73d5923c9a6bad2ff0fe27991838297e032c3fa451e56b15afadcb 0.0s  
=> => exporting attestation manifest sha256:ba56de35bf51c4684666e560bdd2598f03a0858692b05704f96473945ee2ea29f 0.0s  
=> => exporting manifest list sha256:73978eea4fbabcbe436a3b635eedb7ac580efbec22eb13f3a582aac4a822d241b 0.0s  
=> => naming to docker.io/idgafd/ml-app:multistage 0.0s  
=> => unpacking to docker.io/idgafd/ml-app:multistage 0.0s
```

## Run a container

```
anastasiiamazur@bttrm-amazur-pro16 ml-app % docker run -it --rm -p 8050:8050 idgafd/ml-app:multistage  
* Serving Flask app 'app'  
* Debug mode: off  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:8050  
* Running on http://172.17.0.2:8050  
Press CTRL+C to quit  
172.17.0.1 - - [05/Dec/2024 17:55:50] "GET / HTTP/1.1" 200 -  
172.17.0.1 - - [05/Dec/2024 17:56:14] "POST /predict-form HTTP/1.1" 200 -
```

- \* Serving Flask app 'app': Flask application named “app” is being served
- WARNING: this is a warning that the Flask development server is not secure or optimized for production use. In production, a WSGI server like gunicorn or uwsgi are being used, but I’m too lazy to make everything properly (sorry) 😊
- \* Running on all addresses (0.0.0.0: the app is listening on all available network interfaces

- \* Running on `http://127.0.0.1:8050` and \* Running on `http://172.17.0.2:8050`: the app can be accessed via `127.0.0.1` (localhost) for requests from inside the container or `172.17.0.2` for requests within Docker's internal network
- `172.17.0.1 - - [05/Dec/2024 17:55:50] "GET / HTTP/1.1" 200 -`: a successful GET request to the root endpoint `(/)` with HTTP status code `200`, that means the page was served successfully
- `172.17.0.1 - - [05/Dec/2024 17:56:14] "POST /predict-form HTTP/1.1" 200 -`: a POST request to the `/predict-form` route was processed successfully
- `172.17.0.1 - - [05/Dec/2024 17:56:14] "GET /static/iris_setosa.jpg HTTP/1.1" 304 -`: a GET request to retrieve the image `iris_setosa.jpg` from the `/static` directory with HTTP status code `304`, that means that the resource was not modified and is being loaded from the browser cache

## Result Overview

### Predict Iris Flower

**Feature 1:**

**Feature 2:**

**Feature 3:**

**Feature 4:**

**Predict**

### Predict Iris Flower

**Feature 1:**

**Feature 2:**

**Feature 3:**

**Feature 4:**

**Predict**

**Prediction: Setosa**



## Push Docker Image to DockerHub

```
anastasiiamazur@bttrm-amazur-pro16 ml-app % docker tag idgafd/ml-app:multistage idgafd/ml-app:latest
anastasiiamazur@bttrm-amazur-pro16 ml-app % docker push idgafd/ml-app:latest
The push refers to repository [docker.io/idgafd/ml-app]
e88090b39a19: Pushed
cd87c688b22f: Pushed
db13dac0be0a: Pushed
b17061f37db5: Pushed
264404b27a4a: Pushed
5cb15a4160d8: Pushed
f6745ff4cc0b: Pushed
32cbdb1397f6: Pushed
0b3c05c3c4f0: Pushed
bad02e1bcd0d: Pushed
bb3f2b52e6af: Pushed
latest: digest: sha256:73978eea4fbacbe436a3b635eedb7ac580efbec22eb13f3a582aac4a822d241b size: 856
```

Now docker pull idgafd/ml-app:latest can be used to pull the image 

Digest:

sha256:73978eea4fbacbe436a3b635eedb7ac580efbec22eb13f3a582aac4a822d241b

Link: <https://hub.docker.com/r/idgafd/ml-app>

GitHub Repository Link:

[https://github.com/idgafd/UCU\\_DevOps\\_Course/tree/main/ml-app](https://github.com/idgafd/UCU_DevOps_Course/tree/main/ml-app)