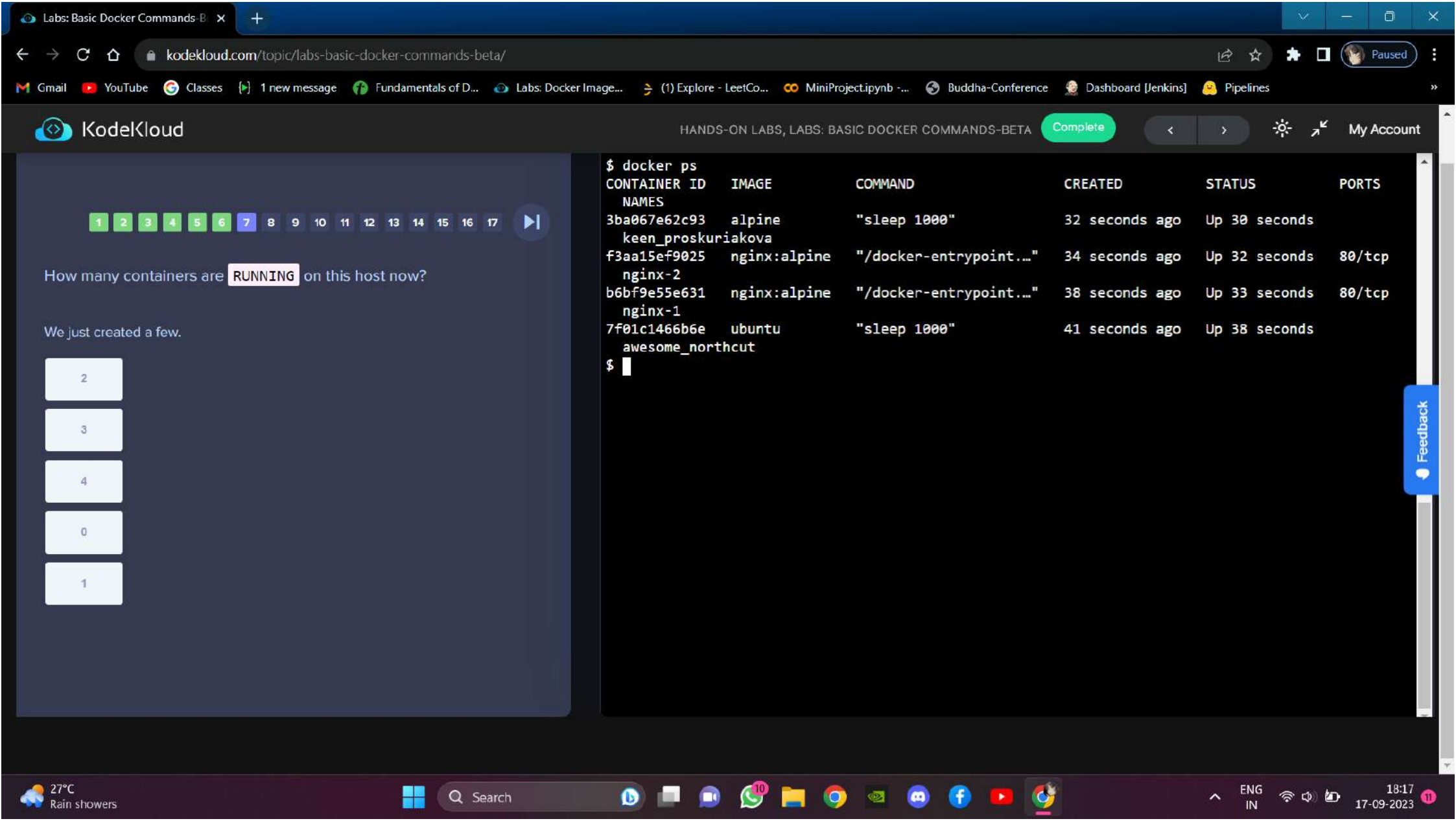
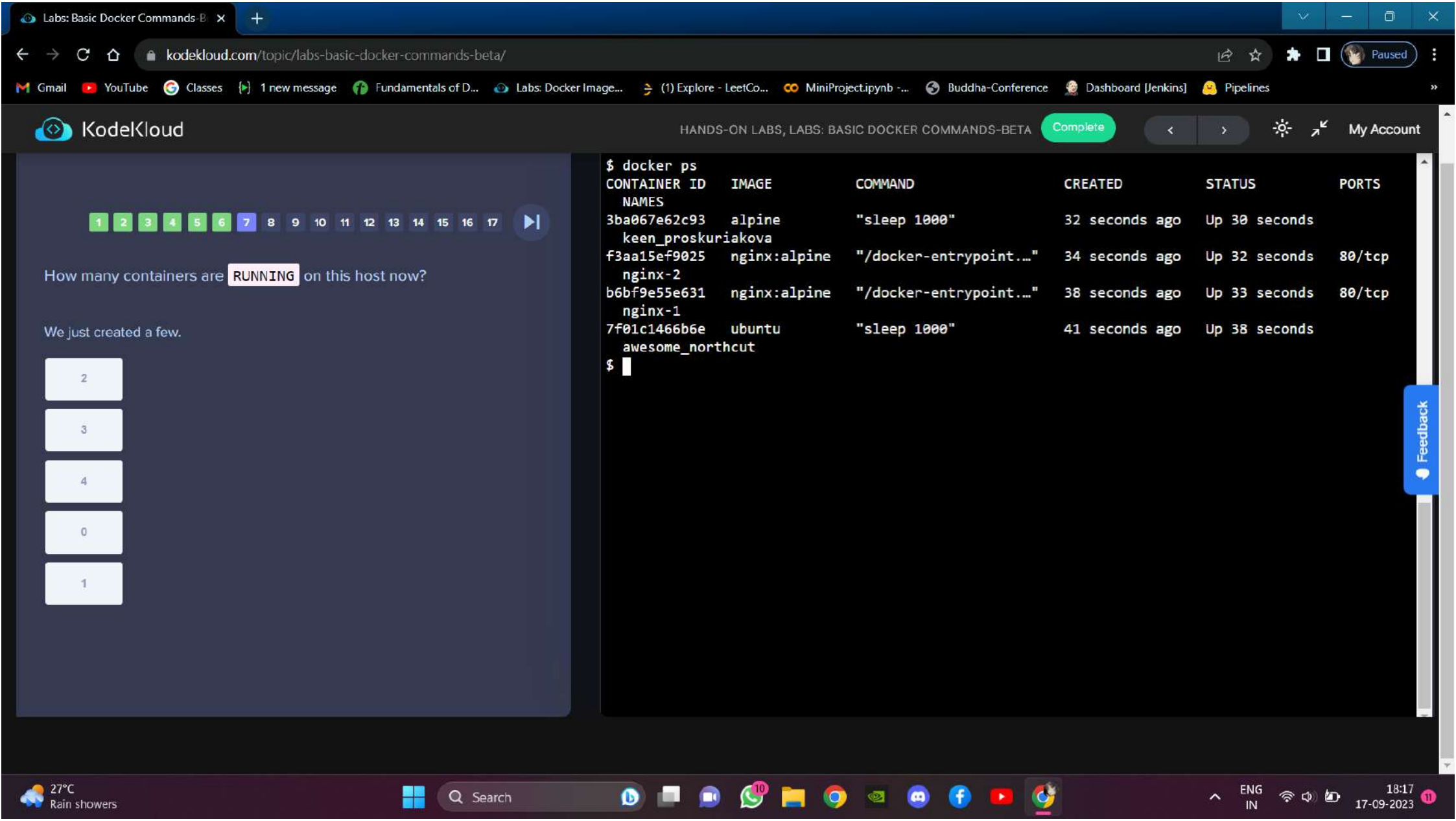
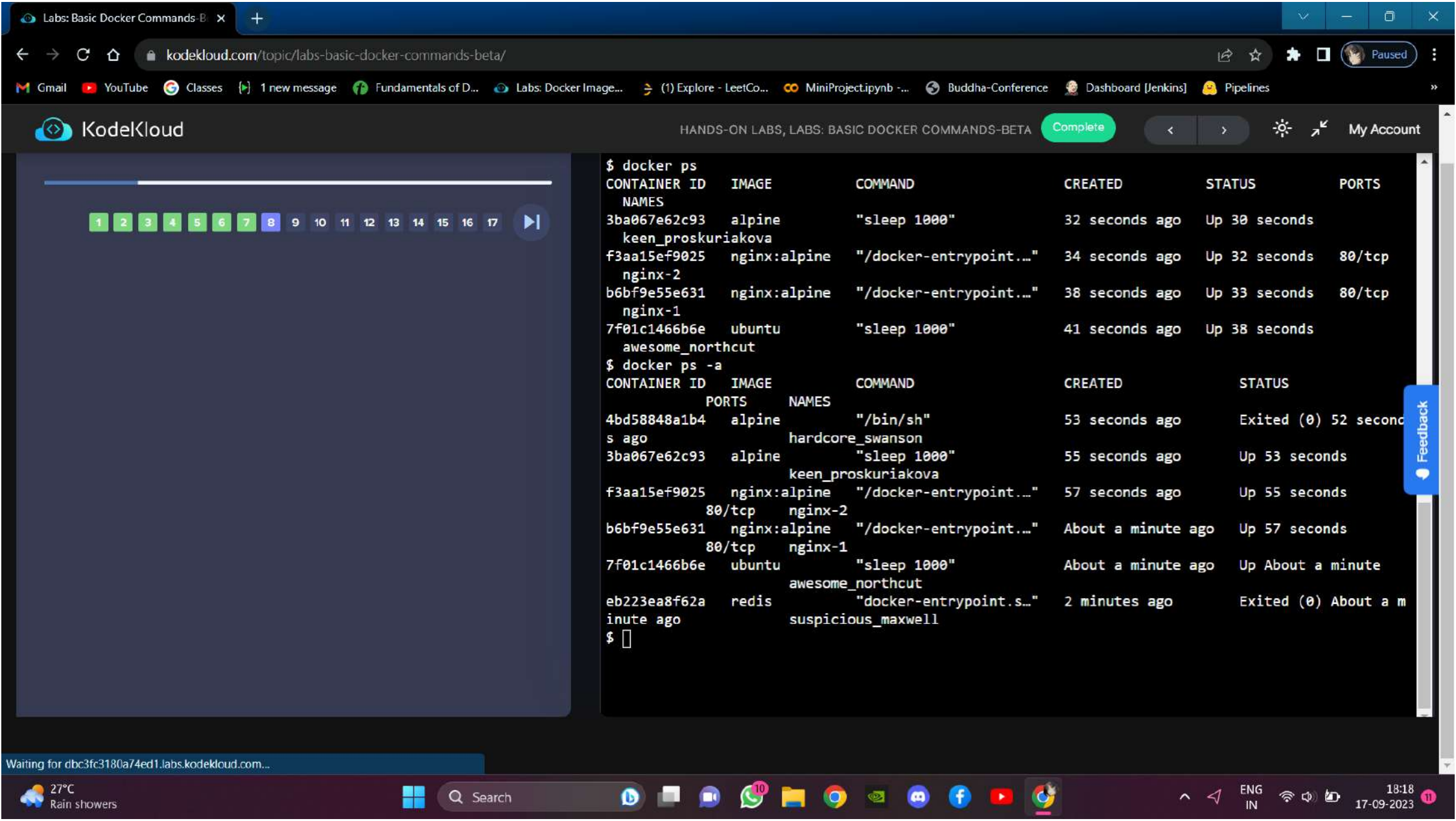
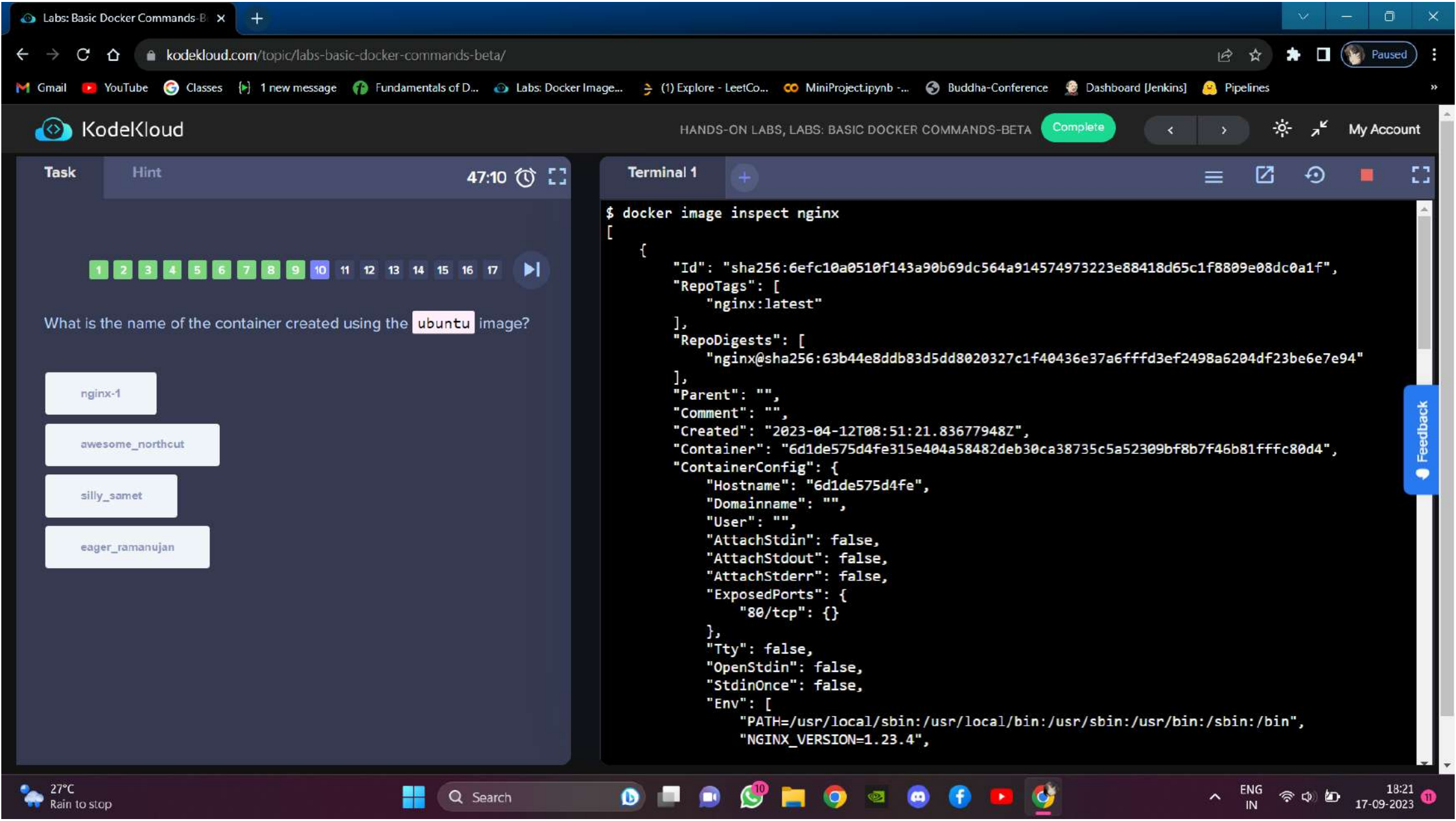

Labs: Basic Docker Commands-B
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
How many containers are RUNNING on this host now?
We just created a few.
2
3
4
0
1
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
3ba067e62c93 alpine "sleep 1000" 32 seconds ago Up 30 seconds
keen_proskuriakova
f3aa15ef9025 nginx:alpine "/docker-entrypoint.…" 34 seconds ago Up 32 seconds 80/tcp
nginx-2
b6bf9e55e631 nginx:alpine "/docker-entrypoint.…" 38 seconds ago Up 33 seconds 80/tcp
nginx-1
7f01c1466b6e ubuntu "sleep 1000" 41 seconds ago Up 38 seconds
awesome_northcut
$
Feedback
27°C
Rain showers
Search
ENG
IN
18:17
17-09-2023


Labs: Basic Docker Commands-B
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
How many containers are RUNNING on this host now?
We just created a few.
2
3
4
0
1
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
3ba067e62c93 alpine "sleep 1000" 32 seconds ago Up 30 seconds
keen_proskuriakova
f3aa15ef9025 nginx:alpine "/docker-entrypoint.…" 34 seconds ago Up 32 seconds 80/tcp
nginx-2
b6bf9e55e631 nginx:alpine "/docker-entrypoint.…" 38 seconds ago Up 33 seconds 80/tcp
nginx-1
7f01c1466b6e ubuntu "sleep 1000" 41 seconds ago Up 38 seconds
awesome_northcut
$
Feedback
27°C
Rain showers
Search
ENG
IN
18:17
17-09-2023


Labs: Basic Docker Commands-B
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
3ba067e62c93 alpine "sleep 1000" 32 seconds ago Up 30 seconds
keen_proskuriakova
f3aa15ef9025 nginx:alpine "/docker-entrypoint.…" 34 seconds ago Up 32 seconds 80/tcp
nginx-2
b6bf9e55e631 nginx:alpine "/docker-entrypoint.…" 38 seconds ago Up 33 seconds 80/tcp
nginx-1
7f01c1466b6e ubuntu "sleep 1000" 41 seconds ago Up 38 seconds
awesome_northcut
$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
4bd58848a1b4 alpine "/bin/sh" 53 seconds ago Exited (0) 52 seconds ago
hardcore_swanson
3ba067e62c93 alpine "sleep 1000" 55 seconds ago Up 53 seconds
keen_proskuriakova
f3aa15ef9025 nginx:alpine "/docker-entrypoint.…" 57 seconds ago Up 55 seconds
80/tcp nginx-2
b6bf9e55e631 nginx:alpine "/docker-entrypoint.…" About a minute ago Up 57 seconds
80/tcp nginx-1
7f01c1466b6e ubuntu "sleep 1000" About a minute ago Up About a minute
awesome_northcut
eb223ea8f62a redis "docker-entrypoint.s…" 2 minutes ago Exited (0) About a minute ago
suspicious_maxwell
$
Feedback
Waiting for dbc3fc3180a74ed1.labs.kodekloud.com...
27°C
Rain showers
Search
ENG
IN
18:18
17-09-2023


Labs: Basic Docker Commands-B
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
Task
Hint
47:10
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
What is the name of the container created using the ubuntu image?
nginx-1
awesome_northcut
silly_samet
eager_ramanujan
Terminal 1
$ docker image inspect nginx
[
{
"Id": "sha256:6efc10a0510f143a90b69dc564a914574973223e88418d65c1f8809e08dc0a1f",
"RepoTags": [
"nginx:latest"
],
"RepoDigests": [
"nginx@sha256:63b44e8ddb83d5dd8020327c1f40436e37a6fffd3ef2498a6204df23be6e7e94"
],
"Parent": "",
"Comment": "",
"Created": "2023-04-12T08:51:21.836779482",
"Container": "6d1de575d4fe315e404a58482deb30ca38735c5a52309bf8b7f46b81fffc80d4",
"ContainerConfig": {
"Hostname": "6d1de575d4fe",
"Domainname": "",
"User": "",
"AttachStdin": false,
"AttachStdout": false,
"AttachStderr": false,
"ExposedPorts": {
"80/tcp": {}
},
"Tty": false,
"OpenStdin": false,
"StdinOnce": false,
"Env": [
"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
"NGINX_VERSION=1.23.4",
Feedback
27°C
Rain to stop
Search
ENG
IN
18:21
17-09-2023

Labs: Basic Docker Commands-B

kodekloud.com/topic/labs-basic-docker-commands-beta/

Paused

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | MiniProject.ipynb -... | Buddha-Conference | Dashboard [Jenkins] | Pipelines

KodeKloud

HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA Complete My Account

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

What is the name of the container created using the `ubuntu` image?

- nginx-1
- awesome_northcut
- silly_samet
- eager_ramanujan

```
$ docker image ls -a
REPOSITORY                    TAG       IMAGE ID       CREATED         SIZE
redis                         latest    eca1379fe8b5   5 months ago    117MB
mysql                         latest    8189e588b0e8   5 months ago    564MB
nginx                         latest    6efc10a0510f   5 months ago    142MB
postgres                      latest    ceccf204404e   5 months ago    379MB
nginx                         alpine    8e75cbc5b25c   5 months ago    41MB
alpine                        latest    9ed4aefc74f6   5 months ago    7.04MB
ubuntu                        latest    08d22c0ceb15   6 months ago    77.8MB
kodekloud/simple-webapp-mysql latest    129dd9f67367   4 years ago     96.6MB
kodekloud/simple-webapp       latest    c6e3cd9aae36   4 years ago     84.8MB
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
  PORTS     NAMES
5851bf1283ec   nginx          "/docker-entrypoint.…"   2 minutes ago   Exited (0) 2 minutes ago
            stoic_benz
4bd58848a1b4   alpine         "/bin/sh"                4 minutes ago   Exited (0) 4 minutes ago
            hardcore_swanson
3ba067e62c93   alpine         "sleep 1000"             4 minutes ago   Up 4 minutes
            keen_proskuriakova
f3aa15ef9025   nginx:alpine   "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes
  80/tcp    nginx-2
b6bf9e55e631   nginx:alpine   "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes
  80/tcp    nginx-1
7f01c1466b6e   ubuntu         "sleep 1000"             5 minutes ago   Up 4 minutes
            awesome_northcut
eb223ea8f62a   redis          "docker-entrypoint.s…"   6 minutes ago   Exited (0) 5 minutes ago
            suspicious_maxwell
$
```

Feedback

27°C Rain showers

Search

ENG IN

18:22 17-09-2023

Labs: Basic Docker Commands-B

kodekloud.com/topic/labs-basic-docker-commands-beta/

Paused

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | MiniProject.ipynb -... | Buddha-Conference | Dashboard [Jenkins] | Pipelines

KodeKloud

HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA Complete My Account

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

What is the name of the container created using the `ubuntu` image?

- nginx-1
- awesome_northcut
- silly_samet
- eager_ramanujan

```
$ docker image ls -a
REPOSITORY                    TAG       IMAGE ID       CREATED         SIZE
redis                         latest    eca1379fe8b5   5 months ago    117MB
mysql                         latest    8189e588b0e8   5 months ago    564MB
nginx                         latest    6efc10a0510f   5 months ago    142MB
postgres                      latest    ceccf204404e   5 months ago    379MB
nginx                         alpine    8e75cbc5b25c   5 months ago    41MB
alpine                        latest    9ed4aefc74f6   5 months ago    7.04MB
ubuntu                        latest    08d22c0ceb15   6 months ago    77.8MB
kodekloud/simple-webapp-mysql latest    129dd9f67367   4 years ago     96.6MB
kodekloud/simple-webapp       latest    c6e3cd9aae36   4 years ago     84.8MB
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
  PORTS     NAMES
5851bf1283ec   nginx          "/docker-entrypoint.…"   2 minutes ago   Exited (0) 2 minutes ago
            stoic_benz
4bd58848a1b4   alpine         "/bin/sh"                4 minutes ago   Exited (0) 4 minutes ago
            hardcore_swanson
3ba067e62c93   alpine         "sleep 1000"             4 minutes ago   Up 4 minutes
            keen_proskuriakova
f3aa15ef9025   nginx:alpine   "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes
  80/tcp    nginx-2
b6bf9e55e631   nginx:alpine   "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes
  80/tcp    nginx-1
7f01c1466b6e   ubuntu         "sleep 1000"             5 minutes ago   Up 4 minutes
            awesome_northcut
eb223ea8f62a   redis          "docker-entrypoint.s…"   6 minutes ago   Exited (0) 5 minutes ago
            suspicious_maxwell
$
```

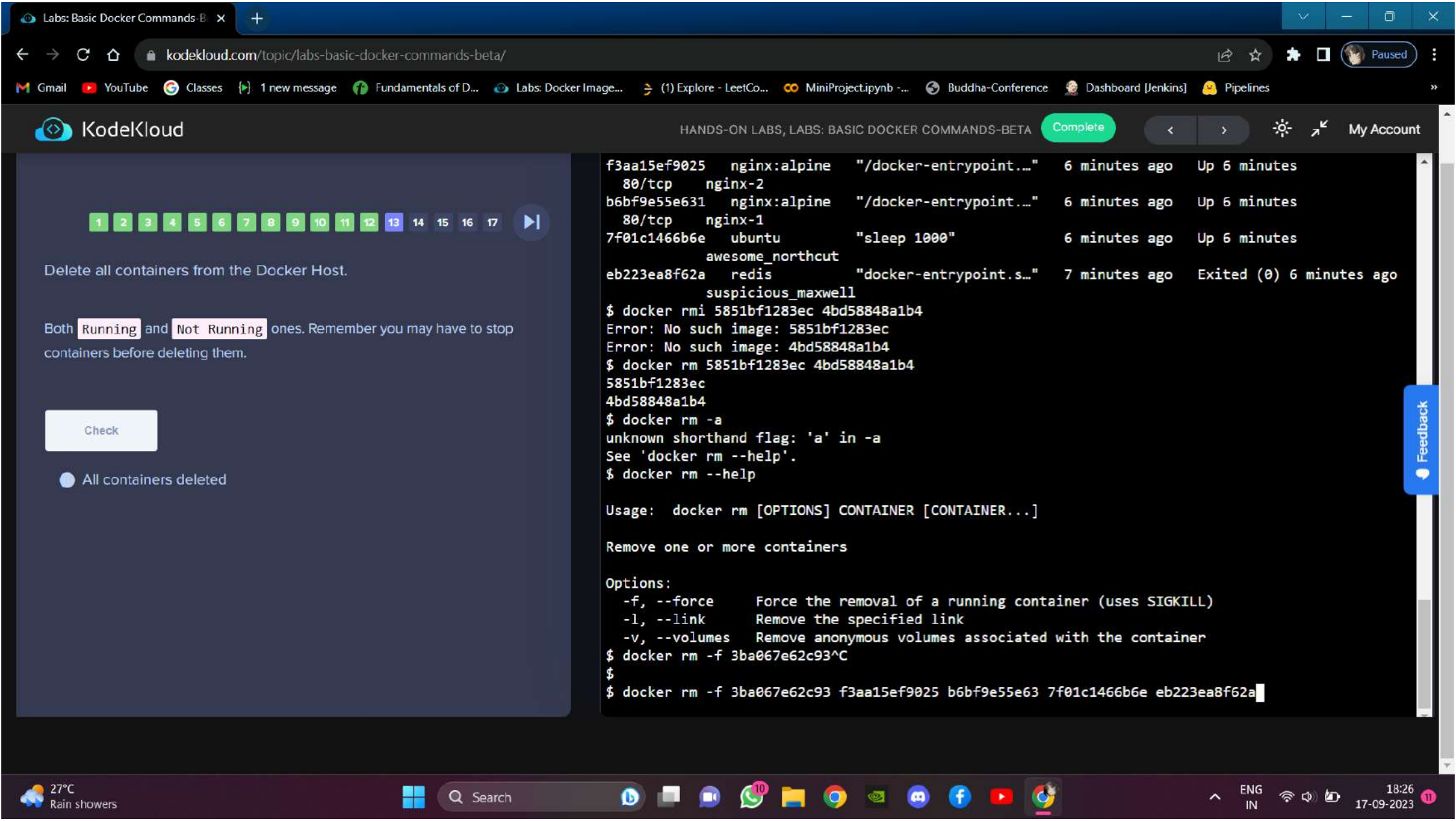
Feedback

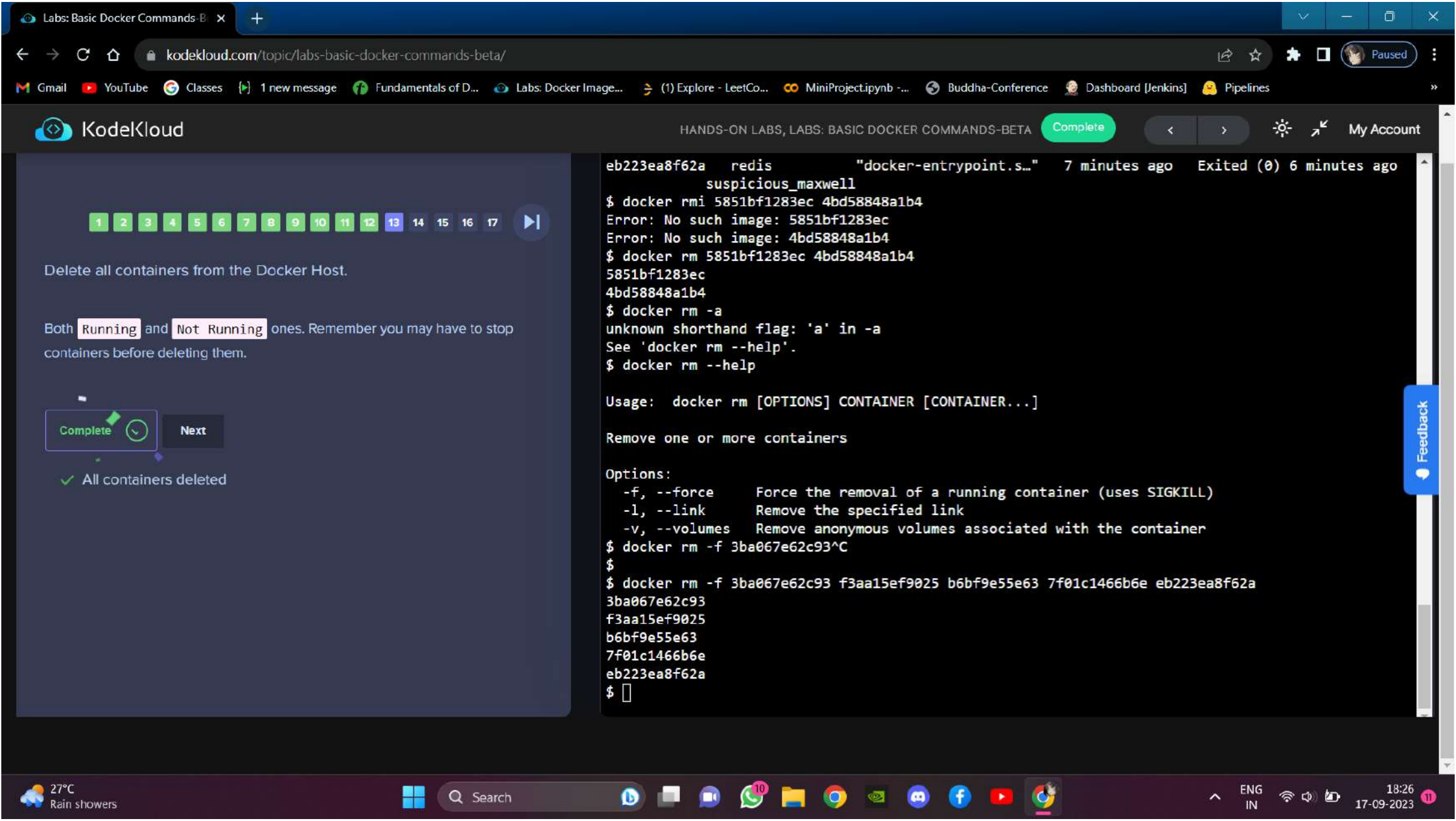
27°C Rain showers

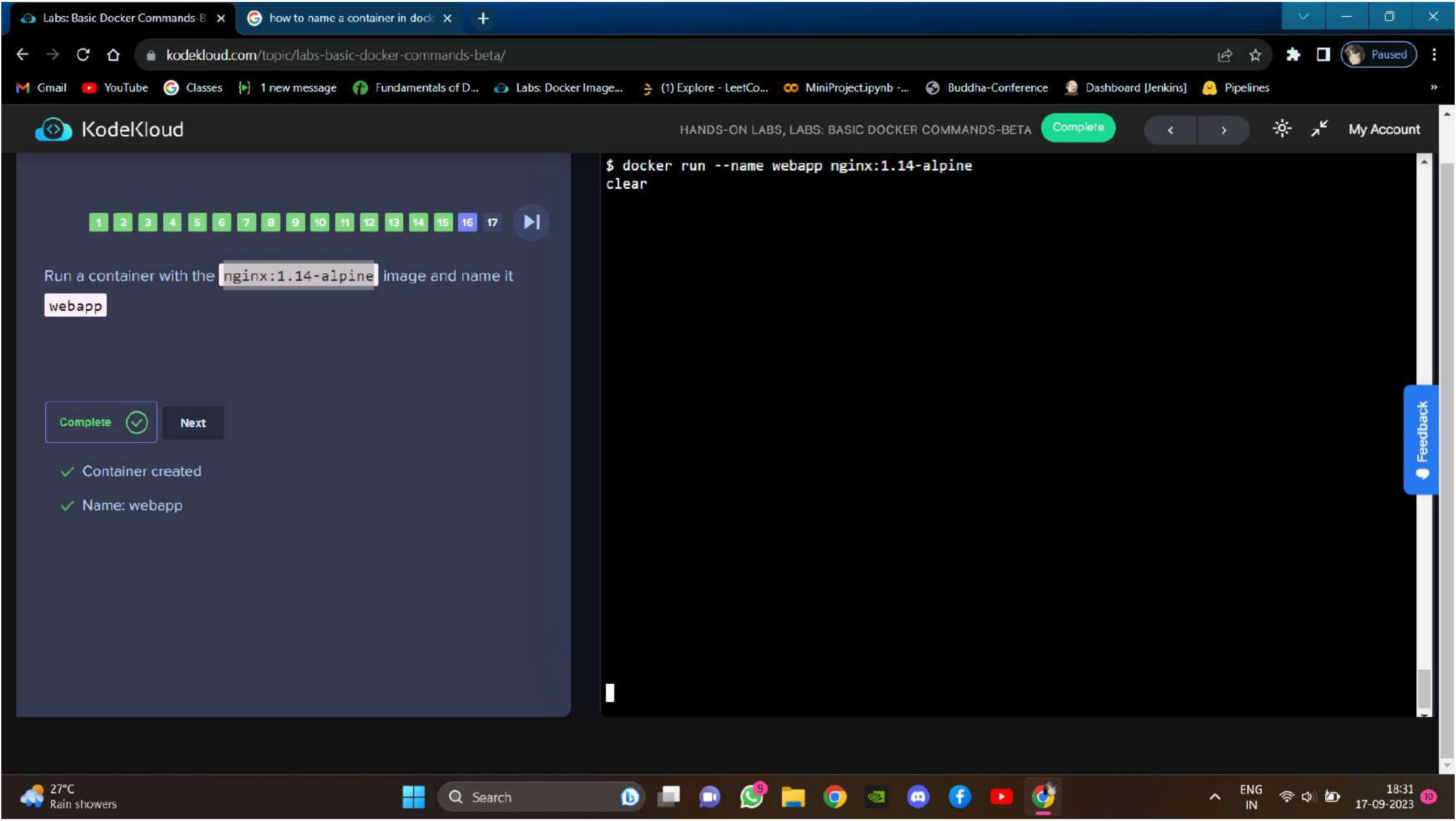
Search

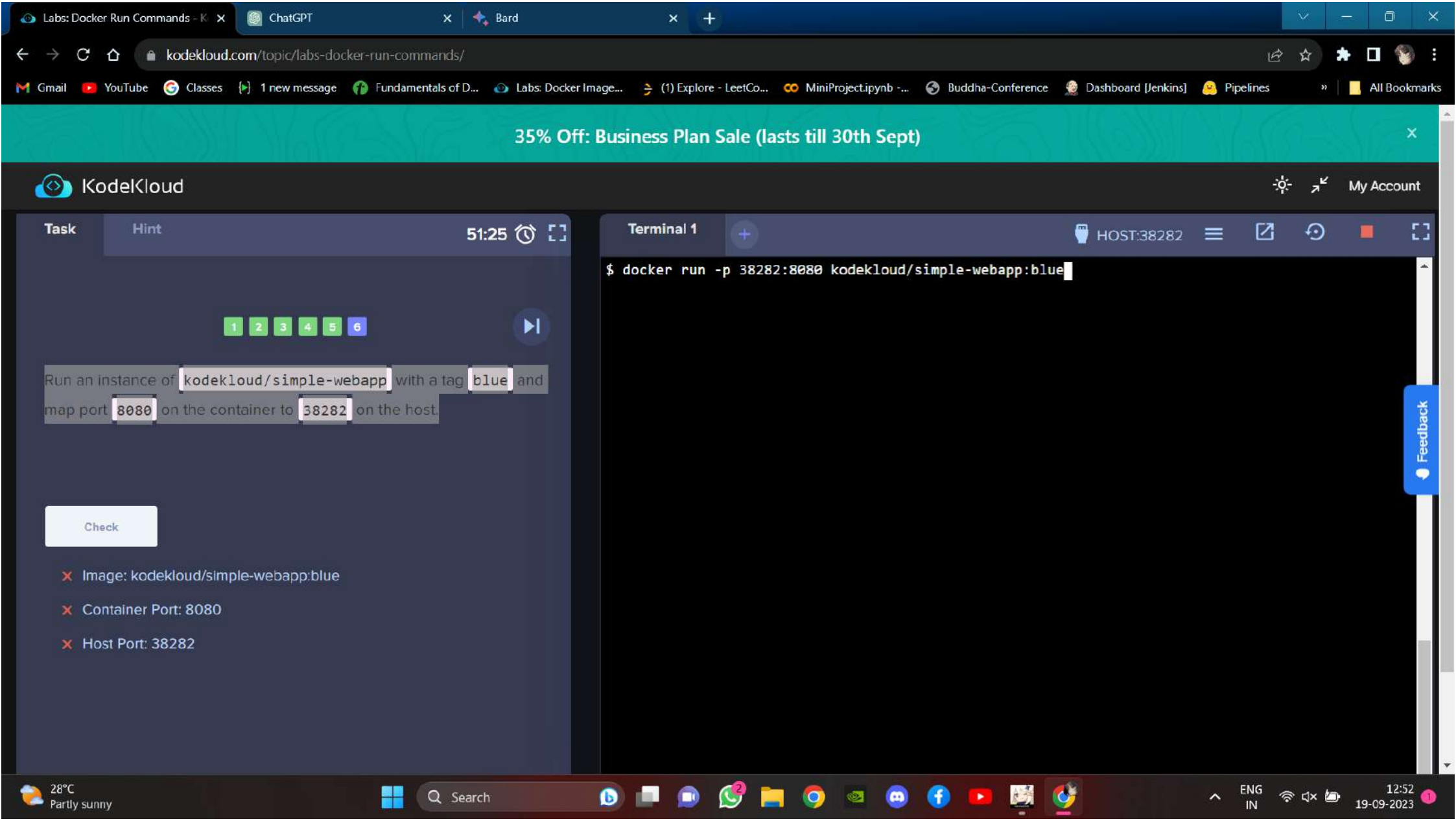
ENG IN

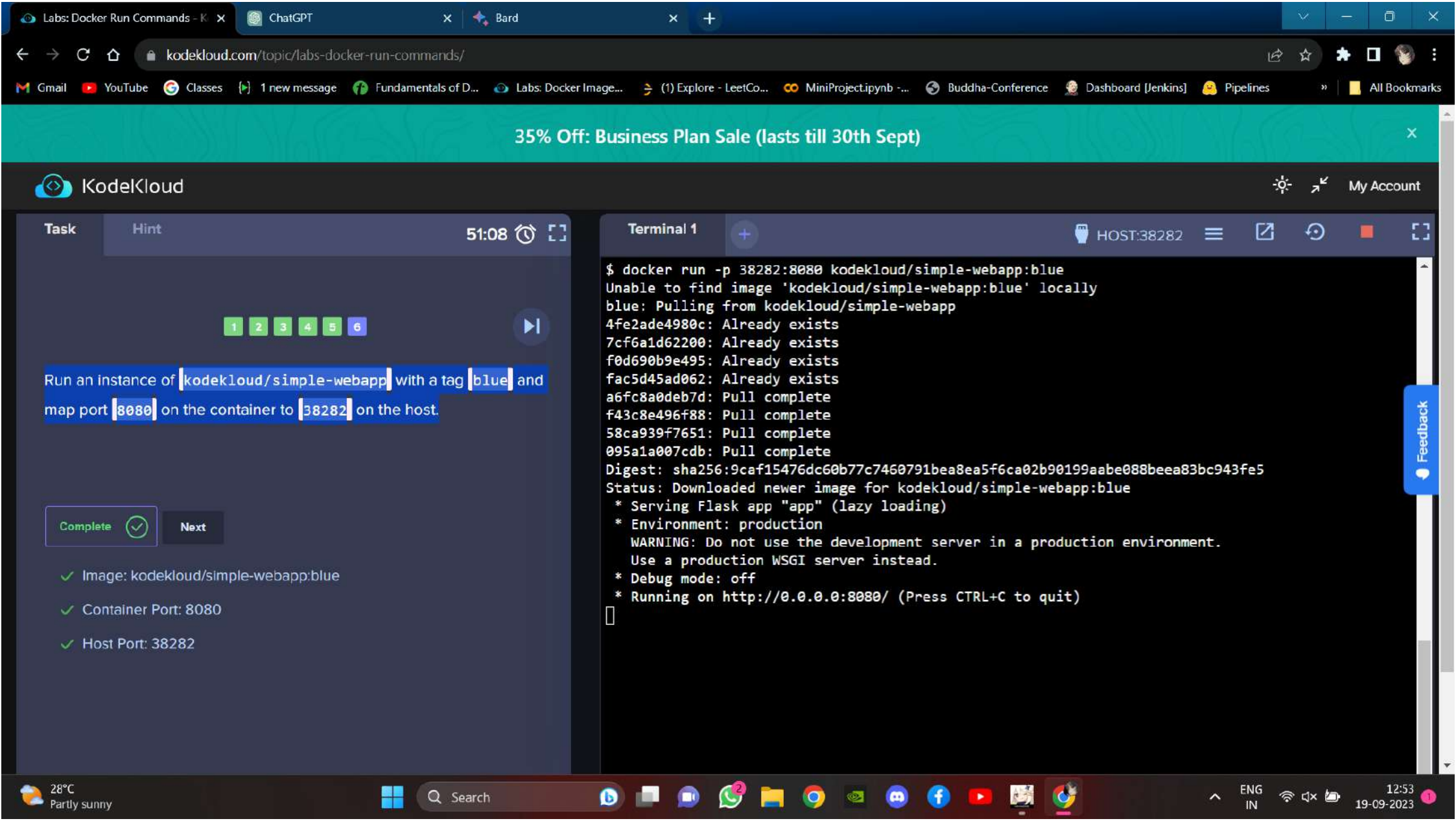
18:22 17-09-2023


Labs: Basic Docker Commands-B
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Delete all containers from the Docker Host.
Both Running and Not Running ones. Remember you may have to stop containers before deleting them.
Check
All containers deleted
f3aa15ef9025 nginx:alpine "/docker-entrypoint.…" 6 minutes ago Up 6 minutes
80/tcp nginx-2
b6bf9e55e631 nginx:alpine "/docker-entrypoint.…" 6 minutes ago Up 6 minutes
80/tcp nginx-1
7f01c1466b6e ubuntu "sleep 1000" 6 minutes ago Up 6 minutes
awesome_northcut
eb223ea8f62a redis "docker-entrypoint.s…" 7 minutes ago Exited (0) 6 minutes ago
suspicious_maxwell
$ docker rmi 5851bf1283ec 4bd58848a1b4
Error: No such image: 5851bf1283ec
Error: No such image: 4bd58848a1b4
$ docker rm 5851bf1283ec 4bd58848a1b4
5851bf1283ec
4bd58848a1b4
$ docker rm -a
unknown shorthand flag: 'a' in -a
See 'docker rm --help'.
$ docker rm --help
Usage: docker rm [OPTIONS] CONTAINER [CONTAINER...]
Remove one or more containers
Options:
-f, --force Force the removal of a running container (uses SIGKILL)
-l, --link Remove the specified link
-v, --volumes Remove anonymous volumes associated with the container
$ docker rm -f 3ba067e62c93^C
$
$ docker rm -f 3ba067e62c93 f3aa15ef9025 b6bf9e55e63 7f01c1466b6e eb223ea8f62a
Feedback
27°C
Rain showers
Search
ENG
IN
18:26
17-09-2023


Labs: Basic Docker Commands-B
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Delete all containers from the Docker Host.
Both Running and Not Running ones. Remember you may have to stop containers before deleting them.
Complete
Next
All containers deleted
eb223ea8f62a redis "docker-entrypoint.s…" 7 minutes ago Exited (0) 6 minutes ago
suspicious_maxwell
$ docker rmi 5851bf1283ec 4bd58848a1b4
Error: No such image: 5851bf1283ec
Error: No such image: 4bd58848a1b4
$ docker rm 5851bf1283ec 4bd58848a1b4
5851bf1283ec
4bd58848a1b4
$ docker rm -a
unknown shorthand flag: 'a' in -a
See 'docker rm --help'.
$ docker rm --help
Usage: docker rm [OPTIONS] CONTAINER [CONTAINER...]
Remove one or more containers
Options:
-f, --force Force the removal of a running container (uses SIGKILL)
-l, --link Remove the specified link
-v, --volumes Remove anonymous volumes associated with the container
$ docker rm -f 3ba067e62c93^C
$
$ docker rm -f 3ba067e62c93 f3aa15ef9025 b6bf9e55e63 7f01c1466b6e eb223ea8f62a
3ba067e62c93
f3aa15ef9025
b6bf9e55e63
7f01c1466b6e
eb223ea8f62a
$
Feedback
27°C
Rain showers
Search
ENG
IN
18:26
17-09-2023


Labs: Basic Docker Commands-B
how to name a container in dock
kodekloud.com/topic/labs-basic-docker-commands-beta/
Paused
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
KodeKloud
HANDS-ON LABS, LABS: BASIC DOCKER COMMANDS-BETA
Complete
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Run a container with the nginx:1.14-alpine image and name it webapp
Complete
Next
Container created
Name: webapp
$ docker run --name webapp nginx:1.14-alpine
clear
Feedback
27°C
Rain showers
Search
ENG
IN
18:31
17-09-2023


Labs: Docker Run Commands - K
ChatGPT
Bard
kodekloud.com/topic/labs-docker-run-commands/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
51:25
1 2 3 4 5 6
Run an instance of kodekloud/simple-webapp with a tag blue and map port 8080 on the container to 38282 on the host.
Check
Image: kodekloud/simple-webapp:blue
Container Port: 8080
Host Port: 38282
Terminal 1
HOST:38282
$ docker run -p 38282:8080 kodekloud/simple-webapp:blue
Feedback
28°C
Partly sunny
Search
ENG
IN
12:52
19-09-2023


Labs: Docker Run Commands - K
ChatGPT
Bard
kodekloud.com/topic/labs-docker-run-commands/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
51:08
1 2 3 4 5 6
Run an instance of kodekloud/simple-webapp with a tag blue and map port 8080 on the container to 38282 on the host.
Complete
Next
Image: kodekloud/simple-webapp:blue
Container Port: 8080
Host Port: 38282
Terminal 1
HOST:38282
$ docker run -p 38282:8080 kodekloud/simple-webapp:blue
Unable to find image 'kodekloud/simple-webapp:blue' locally
blue: Pulling from kodekloud/simple-webapp
4fe2ade4980c: Already exists
7cf6a1d62200: Already exists
f0d690b9e495: Already exists
fac5d45ad062: Already exists
a6fc8a0deb7d: Pull complete
f43c8e496f88: Pull complete
58ca939f7651: Pull complete
095a1a007cdb: Pull complete
Digest: sha256:9caf15476dc60b77c7460791bea8ea5f6ca02b90199aabe088beea83bc943fe5
Status: Downloaded newer image for kodekloud/simple-webapp:blue
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
Feedback
28°C
Partly sunny
Search
ENG
IN
12:53
19-09-2023

Labs: Docker Images - KodeKlou | ChatGPT

kodekloud.com/topic/labs-docker-images/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | MiniProject.ipynb -... | Buddha-Conference | Dashboard [Jenkins] | Pipelines | All Bookmarks

KodeKloud

My Account

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

To what location within the container is the application code copied to during a Docker build?

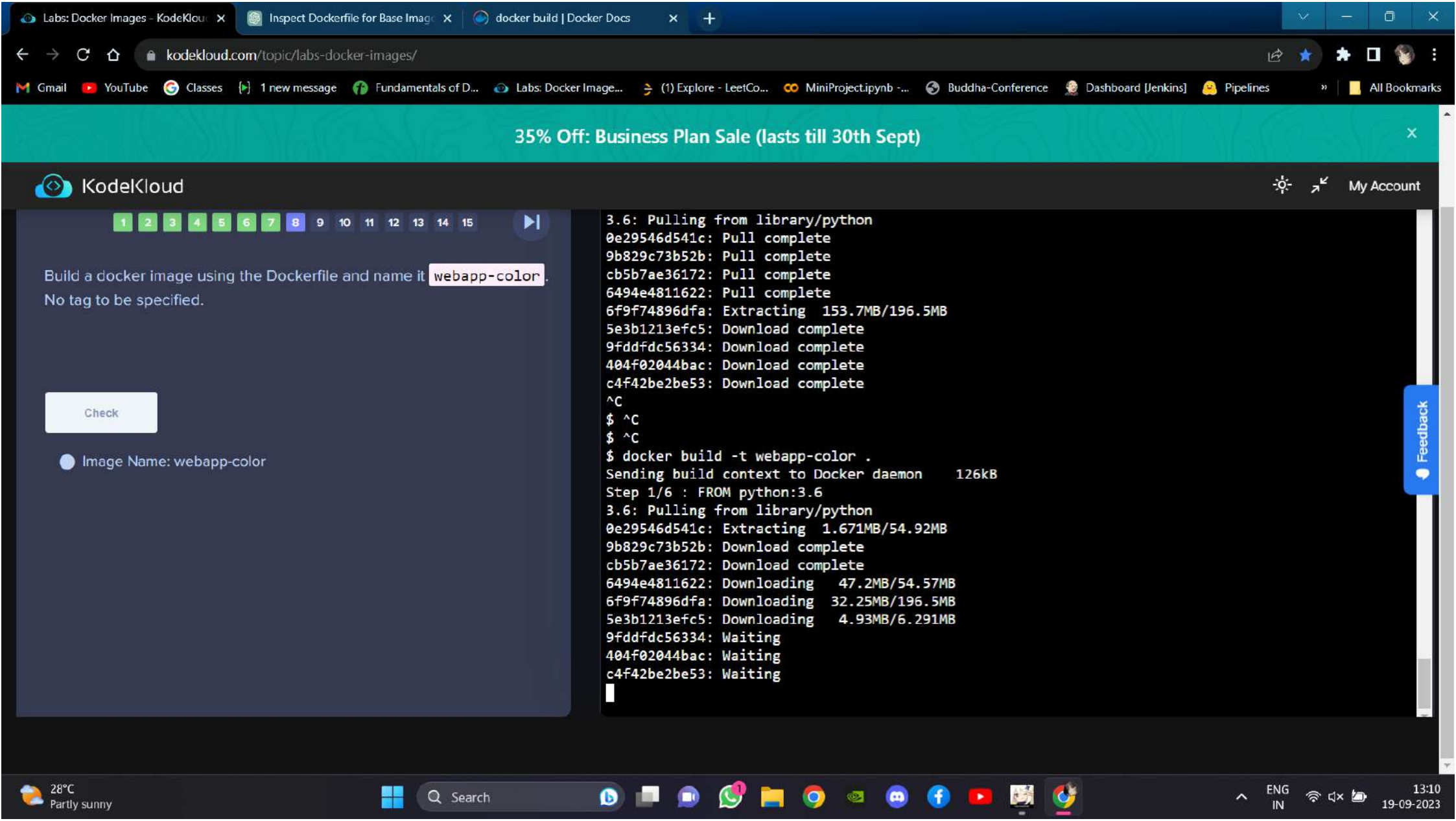
Inspect the Dockerfile in the `webapp-color` directory.

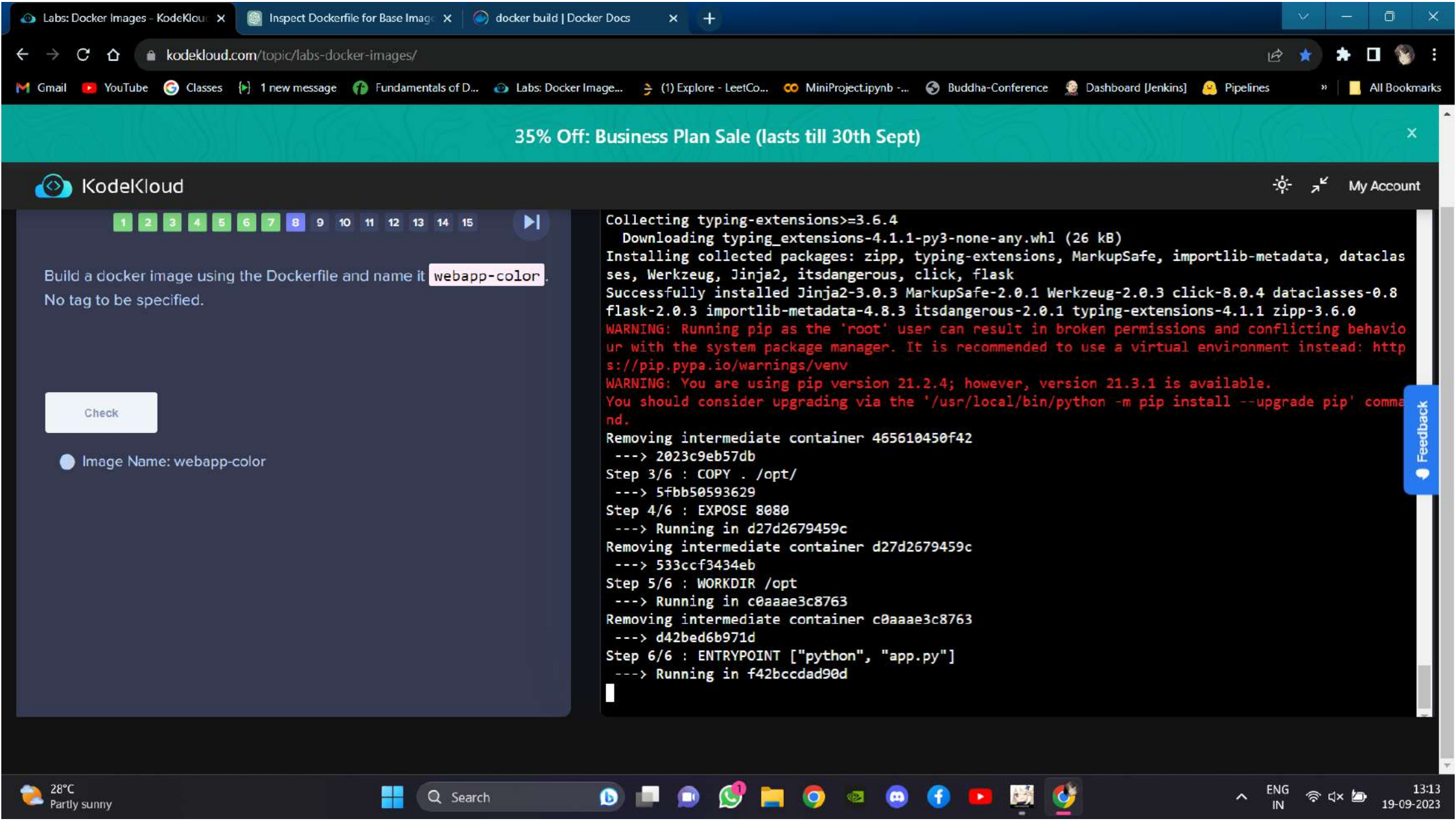
- /tmp
- /root
- /var
- /opt

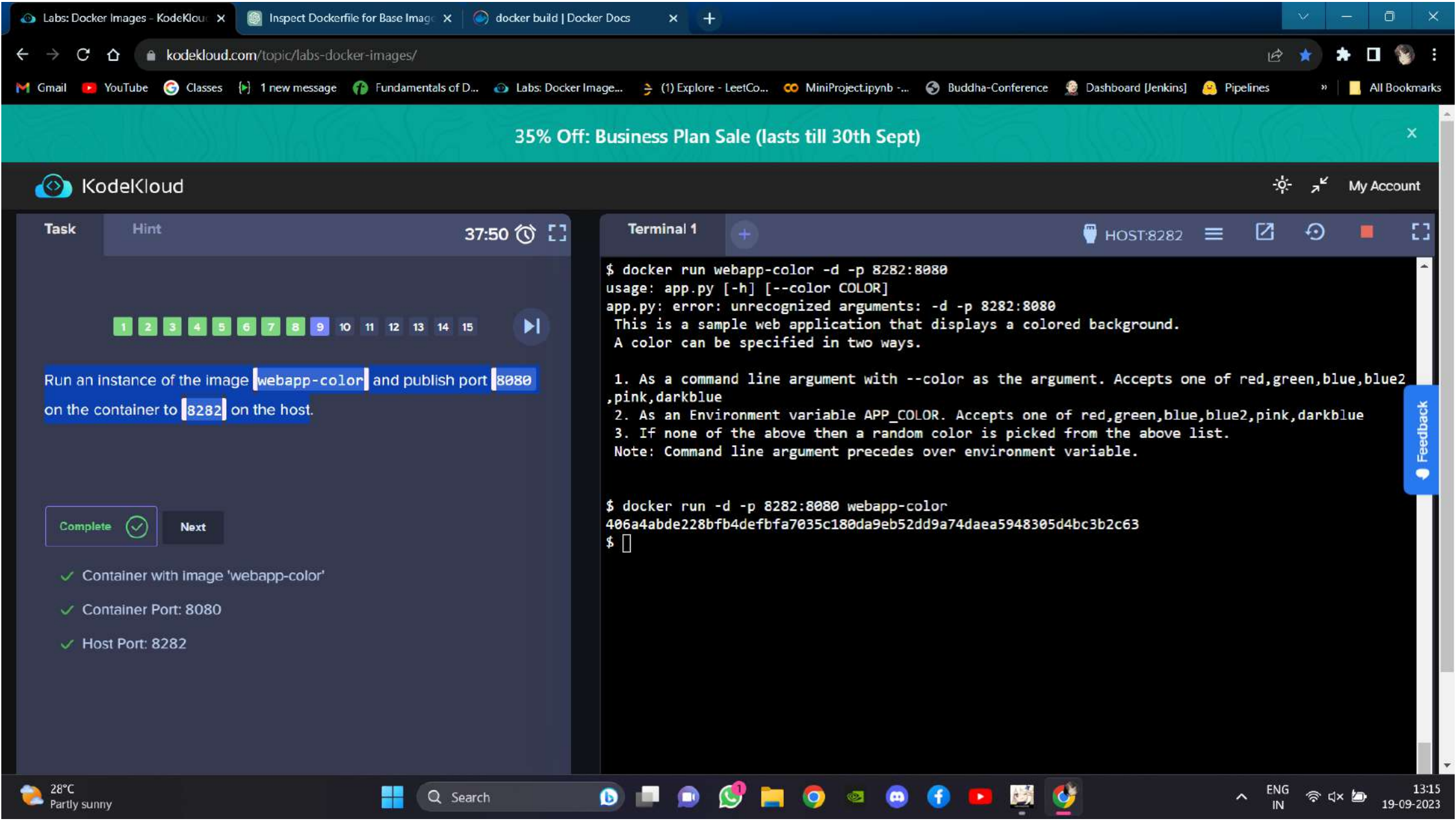
```
postgres                      latest        ceccf204404e   5 months ago   379MB
nginx                         alpine        8e75cbc5b25c   5 months ago   41MB
alpine                        latest        9ed4aefc74f6   5 months ago   7.04MB
ubuntu                        latest        08d22c0ceb15   6 months ago   77.8MB
nginx                         1.14-alpine   8a2fb25a19f5   4 years ago    16MB
kodekloud/simple-webapp-mysql latest        129dd9f67367   4 years ago    96.6MB
kodekloud/simple-webapp       latest        c6e3cd9aae36   4 years ago    84.8MB
$ ls
webapp-color
$ cd webapp-color
$ cat webapp-color
cat: webapp-color: No such file or directory
$ ls
Dockerfile  app.py  requirements.txt  templates
$ cat Dockerfile
FROM python:3.6

RUN pip install flask

COPY . /opt/

EXPOSE 8080

WORKDIR /opt

ENTRYPOINT ["python", "app.py"]
$
```

Feedback

28°C Partly sunny | Search | ENG IN | 13:02 19-09-2023


Labs: Docker Images - KodeKlou
Inspect Dockerfile for Base Imag
docker build | Docker Docs
kodekloud.com/topic/labs-docker-images/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Build a docker image using the Dockerfile and name it webapp-color.
No tag to be specified.
Check
Image Name: webapp-color
3.6: Pulling from library/python
0e29546d541c: Pull complete
9b829c73b52b: Pull complete
cb5b7ae36172: Pull complete
6494e4811622: Pull complete
6f9f74896dfa: Extracting 153.7MB/196.5MB
5e3b1213efc5: Download complete
9fddfdc56334: Download complete
404f02044bac: Download complete
c4f42be2be53: Download complete
^C
$ ^C
$ ^C
$ docker build -t webapp-color .
Sending build context to Docker daemon 126kB
Step 1/6 : FROM python:3.6
3.6: Pulling from library/python
0e29546d541c: Extracting 1.671MB/54.92MB
9b829c73b52b: Download complete
cb5b7ae36172: Download complete
6494e4811622: Downloading 47.2MB/54.57MB
6f9f74896dfa: Downloading 32.25MB/196.5MB
5e3b1213efc5: Downloading 4.93MB/6.291MB
9fddfdc56334: Waiting
404f02044bac: Waiting
c4f42be2be53: Waiting
Feedback
28°C
Partly sunny
Search
ENG
IN
13:10
19-09-2023


Labs: Docker Images - KodeKlou
Inspect Dockerfile for Base Imag
docker build | Docker Docs
kodekloud.com/topic/labs-docker-images/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Build a docker image using the Dockerfile and name it webapp-color.
No tag to be specified.
Check
Image Name: webapp-color
Collecting typing-extensions>=3.6.4
Downloading typing_extensions-4.1.1-py3-none-any.whl (26 kB)
Installing collected packages: zipp, typing-extensions, MarkupSafe, importlib-metadata, dataclasses, Werkzeug, Jinja2, itsdangerous, click, flask
Successfully installed Jinja2-3.0.3 MarkupSafe-2.0.1 Werkzeug-2.0.3 click-8.0.4 dataclasses-0.8 flask-2.0.3 importlib-metadata-4.8.3 itsdangerous-2.0.1 typing-extensions-4.1.1 zipp-3.6.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the '/usr/local/bin/python -m pip install --upgrade pip' command.
Removing intermediate container 465610450f42
---> 2023c9eb57db
Step 3/6 : COPY . /opt/
---> 5fbb50593629
Step 4/6 : EXPOSE 8080
---> Running in d27d2679459c
Removing intermediate container d27d2679459c
---> 533ccf3434eb
Step 5/6 : WORKDIR /opt
---> Running in c0aaae3c8763
Removing intermediate container c0aaae3c8763
---> d42bed6b971d
Step 6/6 : ENTRYPOINT ["python", "app.py"]
---> Running in f42bccdad90d
Feedback
28°C
Partly sunny
Search
ENG
IN
13:13
19-09-2023


Labs: Docker Images - KodeKlou
Inspect Dockerfile for Base Imag
docker build | Docker Docs
kodekloud.com/topic/labs-docker-images/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
MiniProject.ipynb -...
Buddha-Conference
Dashboard [Jenkins]
Pipelines
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
37:50
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Run an instance of the image webapp-color and publish port 8080 on the container to 8282 on the host.
Complete
Next
Container with image 'webapp-color'
Container Port: 8080
Host Port: 8282
Terminal 1
HOST:8282
$ docker run webapp-color -d -p 8282:8080
usage: app.py [-h] [--color COLOR]
app.py: error: unrecognized arguments: -d -p 8282:8080
This is a sample web application that displays a colored background.
A color can be specified in two ways.
1. As a command line argument with --color as the argument. Accepts one of red,green,blue,blue2,pink,darkblue
2. As an Environment variable APP_COLOR. Accepts one of red,green,blue,blue2,pink,darkblue
3. If none of the above then a random color is picked from the above list.
Note: Command line argument precedes over environment variable.
$ docker run -d -p 8282:8080 webapp-color
406a4abde228bfb4defbfa7035c180da9eb52dd9a74daea5948305d4bc3b2c63
$
Feedback
28°C
Partly sunny
Search
ENG
IN
13:15
19-09-2023

Labs: Docker Images - KodeKlou | Inspect Dockerfile for Base Imag | docker build | Docker Docs

kodekloud.com/topic/labs-docker-images/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | MiniProject.ipynb -... | Buddha-Conference | Dashboard [Jenkins] | Pipelines | All Bookmarks

KodeKloud

My Account

Task | Hint | 34:39

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

What is the approximate size of the `webapp-color` image?

500MB

913MB

920MB

80MB

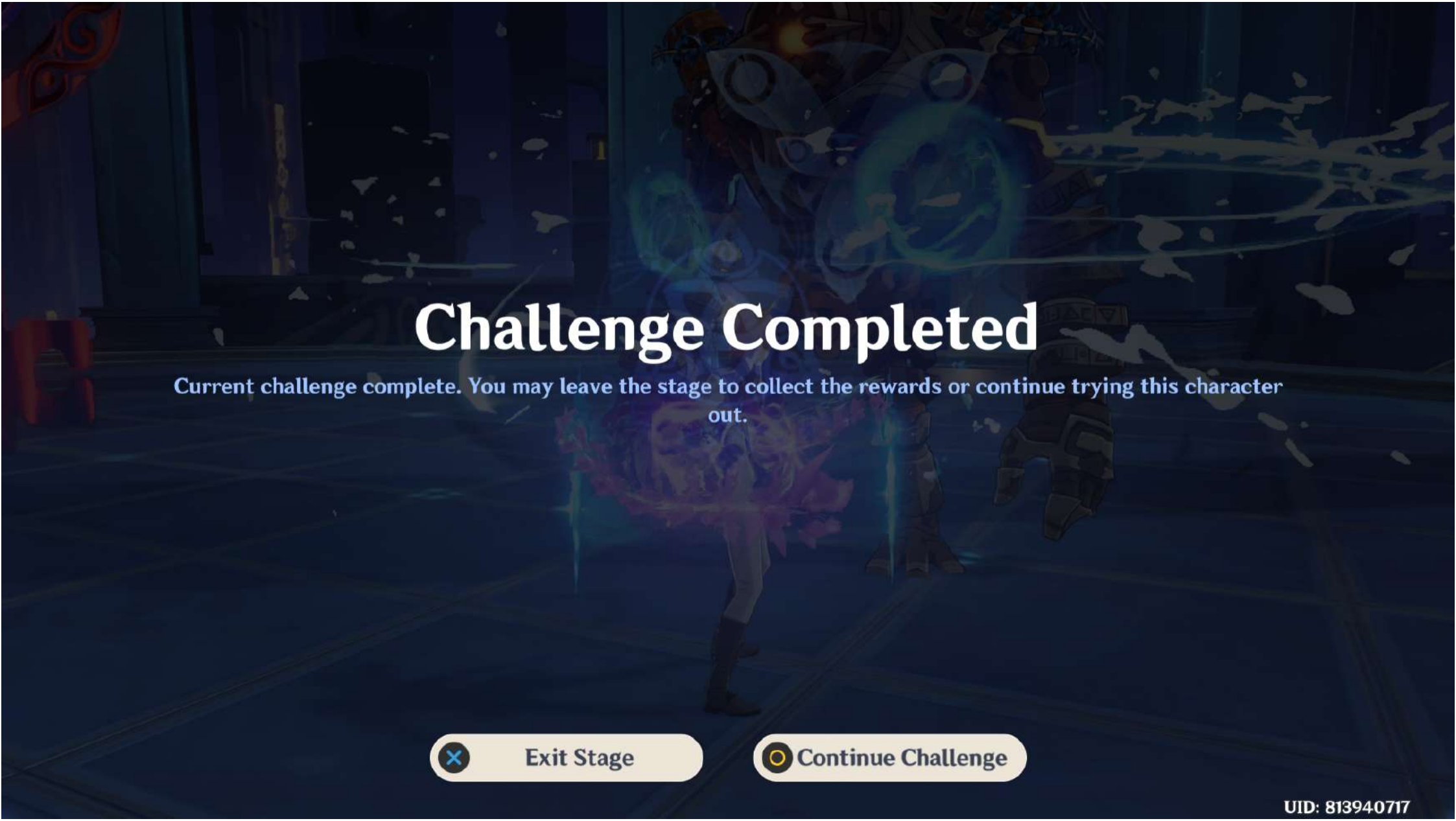
2GB

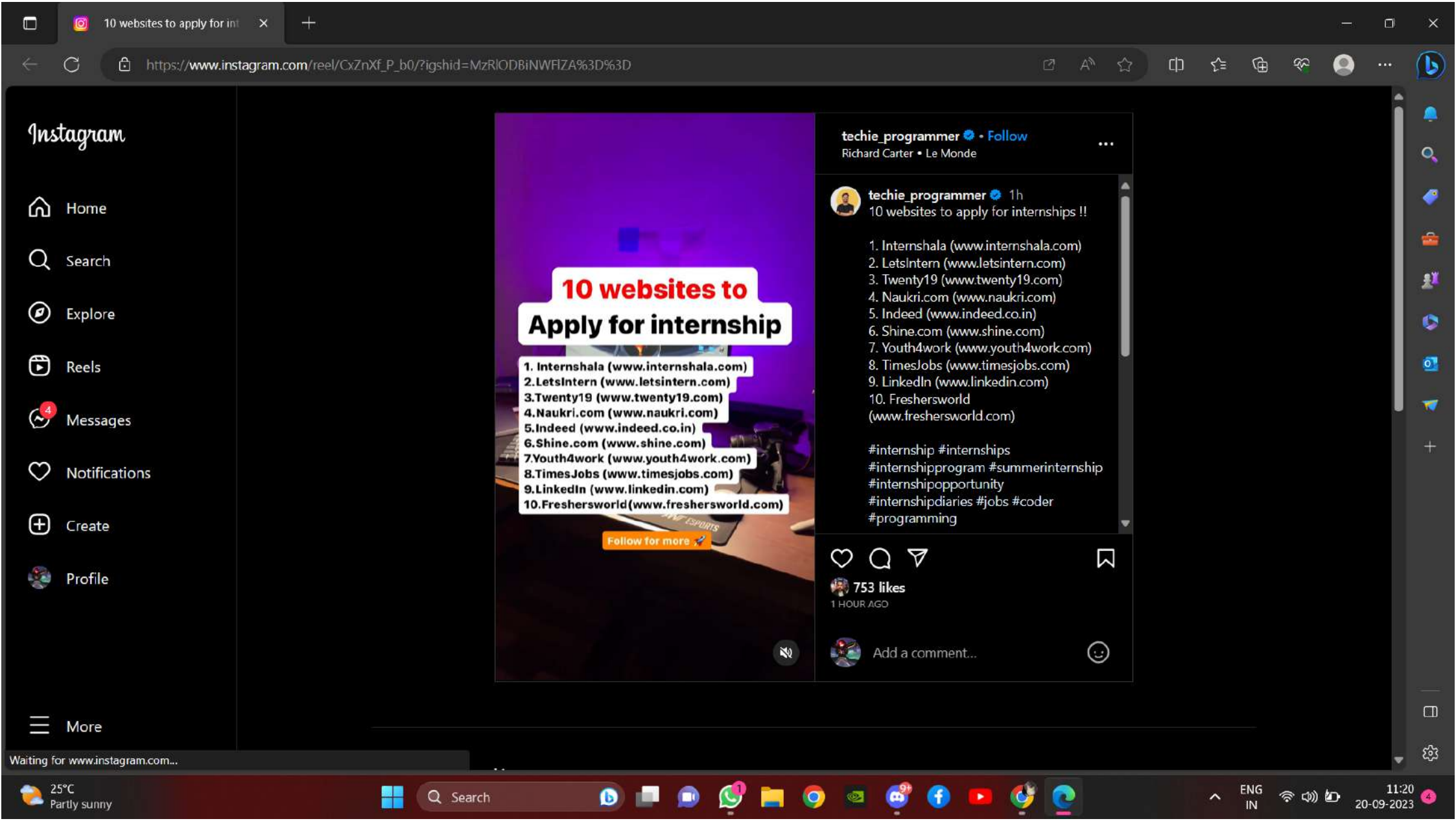
Terminal 1 | HOST:8282

```
$ docker image ls
REPOSITORY                   TAG          IMAGE ID       CREATED          SIZE
webapp-color                 latest       e6a8fb5617d3   4 minutes ago    913MB
nginx                        latest       f5a6b296b8a2   11 days ago      187MB
redis                        latest       eca1379fe8b5   5 months ago     117MB
mysql                        latest       8189e588b0e8   5 months ago     564MB
nginx                        <none>       6efc10a0510f   5 months ago     142MB
postgres                     latest       ceccf204404e   5 months ago     379MB
nginx                        alpine       8e75cbc5b25c   5 months ago     41MB
alpine                       latest       9ed4aefc74f6   5 months ago     7.04MB
ubuntu                       latest       08d22c0ceb15   6 months ago     77.8MB
python                       3.6          54260638d07c   21 months ago    902MB
nginx                        1.14-alpine  8a2fb25a19f5   4 years ago      16MB
kodekloud/simple-webapp-mysql latest      129dd9f67367   4 years ago      96.6MB
kodekloud/simple-webapp      latest       c6e3cd9aae36   4 years ago      84.8MB
$ docker run e6a8fb5617d3
 This is a sample web application that displays a colored background.
 A color can be specified in two ways.

 1. As a command line argument with --color as the argument. Accepts one of red,green,blue,blue2
,pink,darkblue
 2. As an Environment variable APP_COLOR. Accepts one of red,green,blue,blue2,pink,darkblue
 3. If none of the above then a random color is picked from the above list.
 Note: Command line argument precedes over environment variable.


No command line argument or environment variable. Picking a Random Color =green
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
```

Feedback

28°C Partly sunny | Search | ENG IN | 13:18 19-09-2023


Challenge Completed
Current challenge complete. You may leave the stage to collect the rewards or continue trying this character out.
Exit Stage
Continue Challenge
UID: 813940717


10 websites to apply for int
https://www.instagram.com/reel/CxZnXf_P_b0/?igshid=MzRlODBiNWFlZA%3D%3D
Instagram
Home
Search
Explore
Reels
Messages
Notifications
Create
Profile
More
10 websites to
Apply for internship
1. Internshala (www.internshala.com)
2.LetsIntern (www.letsintern.com)
3.Twenty19 (www.twenty19.com)
4.Naukri.com (www.naukri.com)
5.Indeed (www.indeed.co.in)
6.Shine.com (www.shine.com)
7.Youth4work (www.youth4work.com)
8.TimesJobs (www.timesjobs.com)
9.LinkedIn (www.linkedin.com)
10.Freshersworld(www.freshersworld.com)
Follow for more
techie_programmer • Follow
Richard Carter • Le Monde
techie_programmer 1h
10 websites to apply for internships !!
1. Internshala (www.internshala.com)
2. LetsIntern (www.letsintern.com)
3. Twenty19 (www.twenty19.com)
4. Naukri.com (www.naukri.com)
5. Indeed (www.indeed.co.in)
6. Shine.com (www.shine.com)
7. Youth4work (www.youth4work.com)
8. TimesJobs (www.timesjobs.com)
9. LinkedIn (www.linkedin.com)
10. Freshersworld
(www.freshersworld.com)
#internship #internships
#internshipprogram #summerinternship
#internshipopportunity
#internshipdiaries #jobs #coder
#programming
753 likes
1 HOUR AGO
Add a comment...
Waiting for www.instagram.com...
25°C
Partly sunny
Search
ENG
IN
11:20
20-09-2023

Exec – execute a command

```
docker ps -a
```

```
CONTAINER ID        IMAGE     COMMAND       CREATED             STATUS              NAMES
538d037f94a7        ubuntu    "sleep 100“   6 seconds ago       Up 4 seconds        distracted_mcclintock
```

```
docker exec distracted_mcclintock cat /etc/hosts
```

```
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      538d037f94a7
```

RUN - STDIN

```
~/prompt-application$ ./app.sh
Welcome! Please enter your name: Mumshad

Hello and Welcome Mumshad!
```

```
docker run kodekloud/simple-prompt-docker

Hello and Welcome !
```

```
docker run -i kodekloud/simple-prompt-docker
Mumshad

Hello and Welcome Mumshad!
```

RUN - STDIN

```
~/prompt-application$ ./app.sh
Welcome! Please enter your name: Mumshad

Hello and Welcome Mumshad!
```

```
docker run kodekloud/simple-prompt-docker

Hello and Welcome !
```

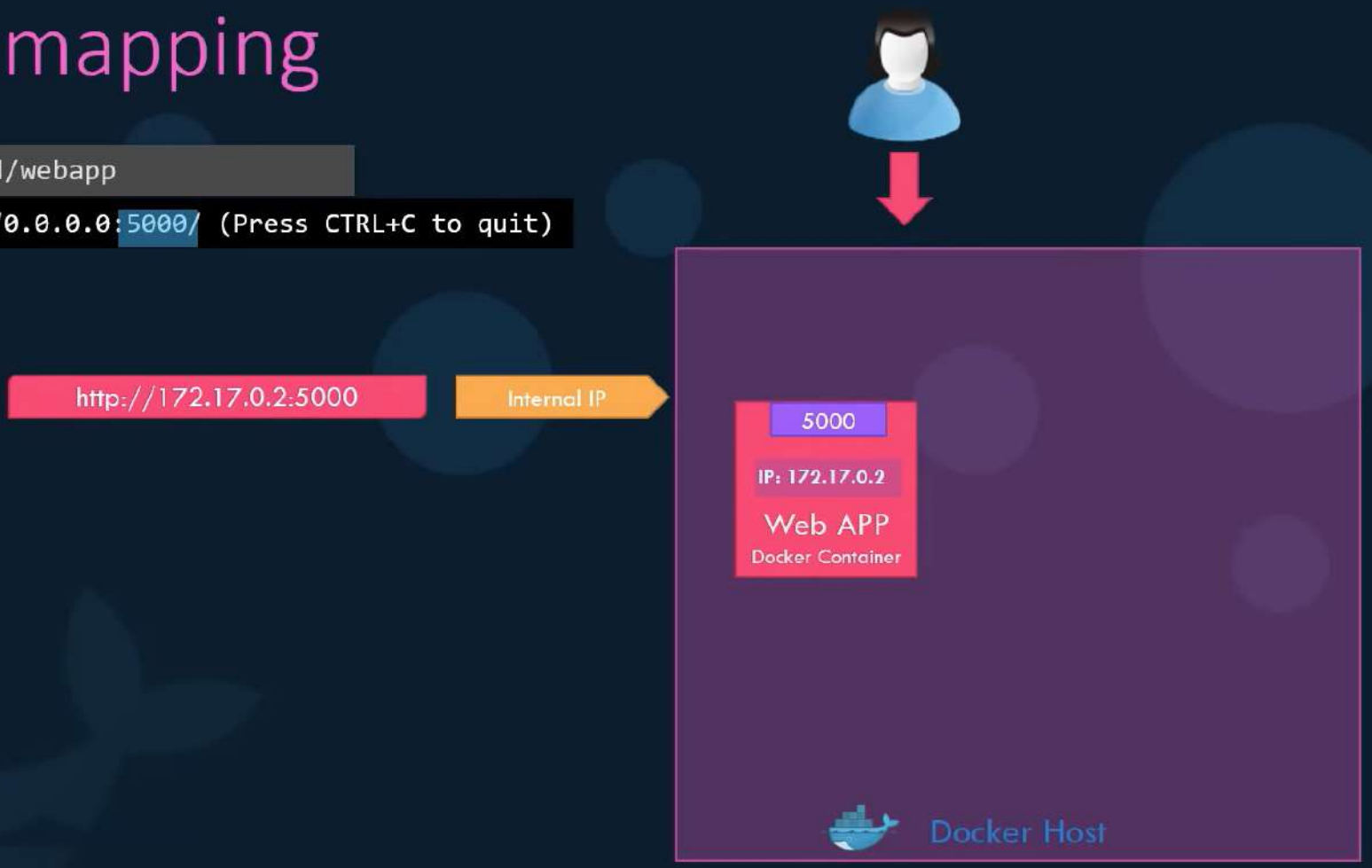
```
docker run -i kodekloud/simple-prompt-docker
Mumshad

Hello and Welcome Mumshad!
```

```
docker run -it kodekloud/simple-prompt-docker
Welcome! Please enter your name: Mumshad

Hello and Welcome Mumshad!
```

Run – PORT mapping

```
docker run kodekloud/webapp
```

```
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```



Run – PORT mapping

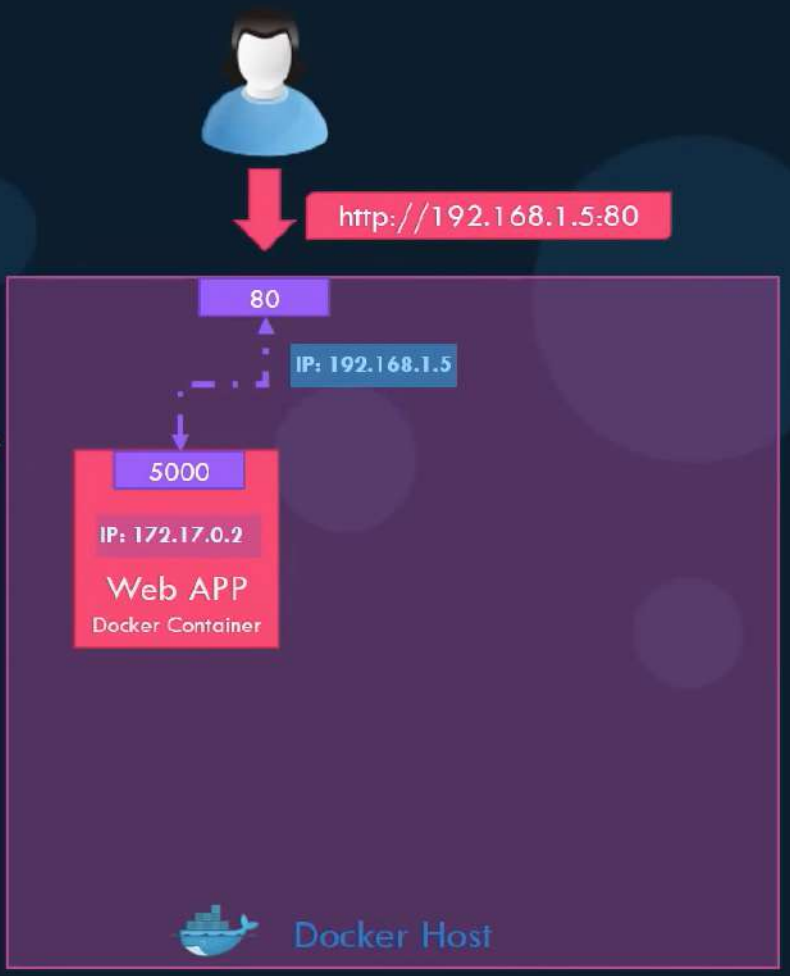
```
docker run kodekloud/webapp
```

```
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

http://172.17.0.2:5000

Internal IP

```
docker run -p 80:5000 kodekloud/simple-webapp
```

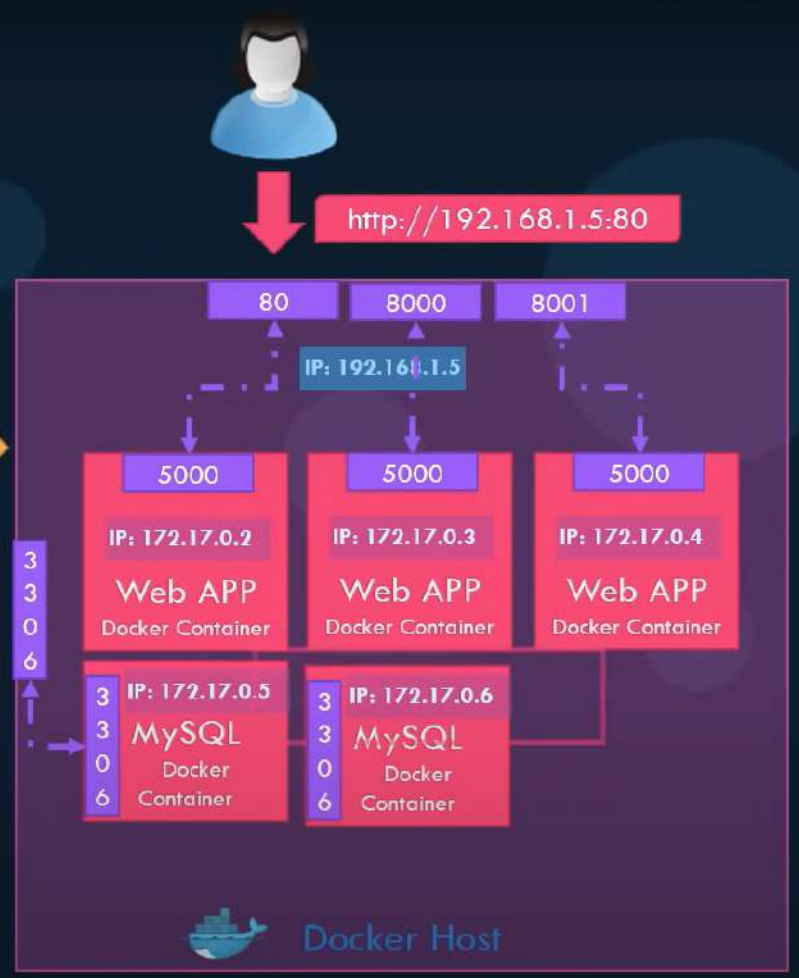


Docker Tutorial for Beginners | Docker Full Course | Access to FREE LABS [No Ads]

Run – PORT mapping

```
docker run kodekloud/webapp
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

```
docker run -p 80:5000 kodekloud/simple-webapp
docker run -p 8000:5000 kodekloud/simple-webapp
docker run -p 8001:5000 kodekloud/simple-webapp
docker run -p 3306:3306 mysql
docker run -p 8306:3306 mysql
```



39:07 / 2:09:27 • Docker Commands ›

Scroll for details

Run – PORT mapping

```
docker run kodekloud/webapp
```

```
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

http://172.17.0.2:5000 Internal IP

```
docker run -p 80:5000 kodekloud/simple-webapp
```

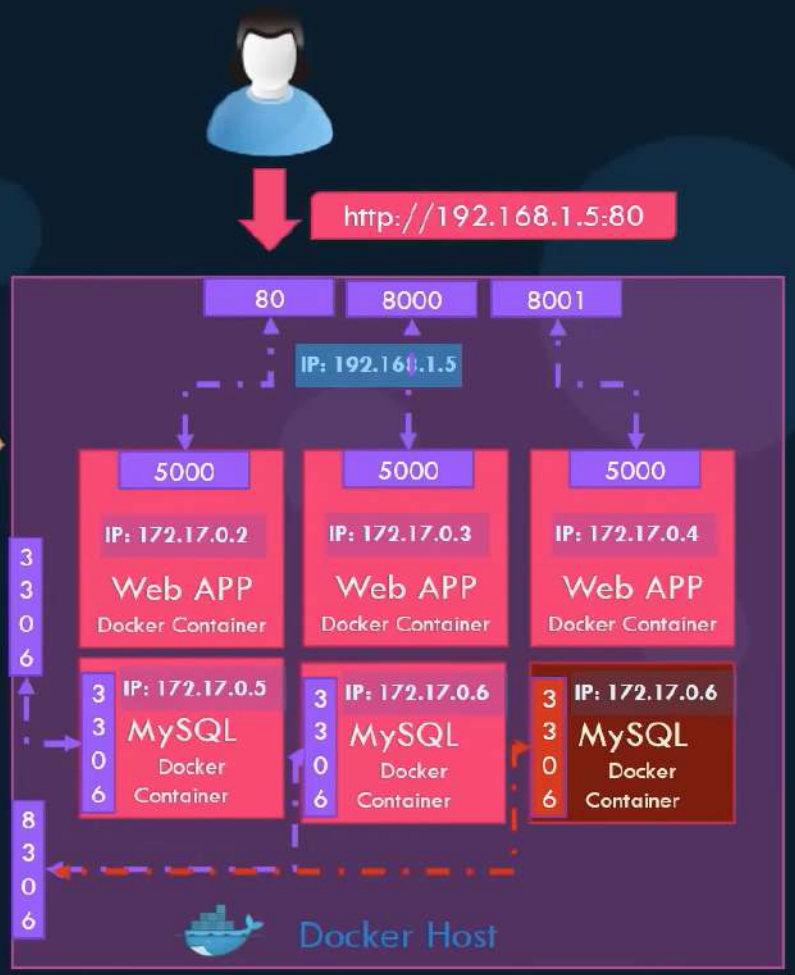
```
docker run -p 8000:5000 kodekloud/simple-webapp
```

```
docker run -p 8001:5000 kodekloud/simple-webapp
```

```
docker run -p 3306:3306 mysql
```

```
docker run -p 8306:3306 mysql
```

```
docker run -p 8306:3306 mysql
```



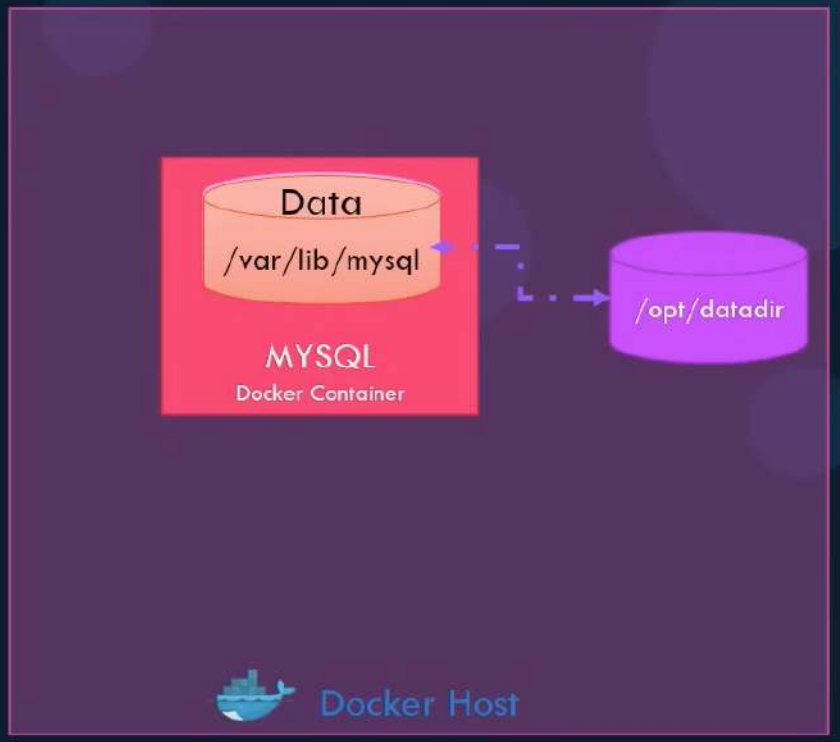

```
root@osboxes:/root # docker run -p 8306:3306 -e MYSQL_ROOT_PASSWORD=pass mysql
docker: Error response from daemon: driver failed programming external connectivity on endpoint boring_bhabha (
5079d342b7e8ee11c71d46): Bind for 0.0.0.0:8306 failed: port is already allocated.
```

RUN – Volume mapping

```
docker run mysql
```

```
docker stop mysql
docker rm mysql
```

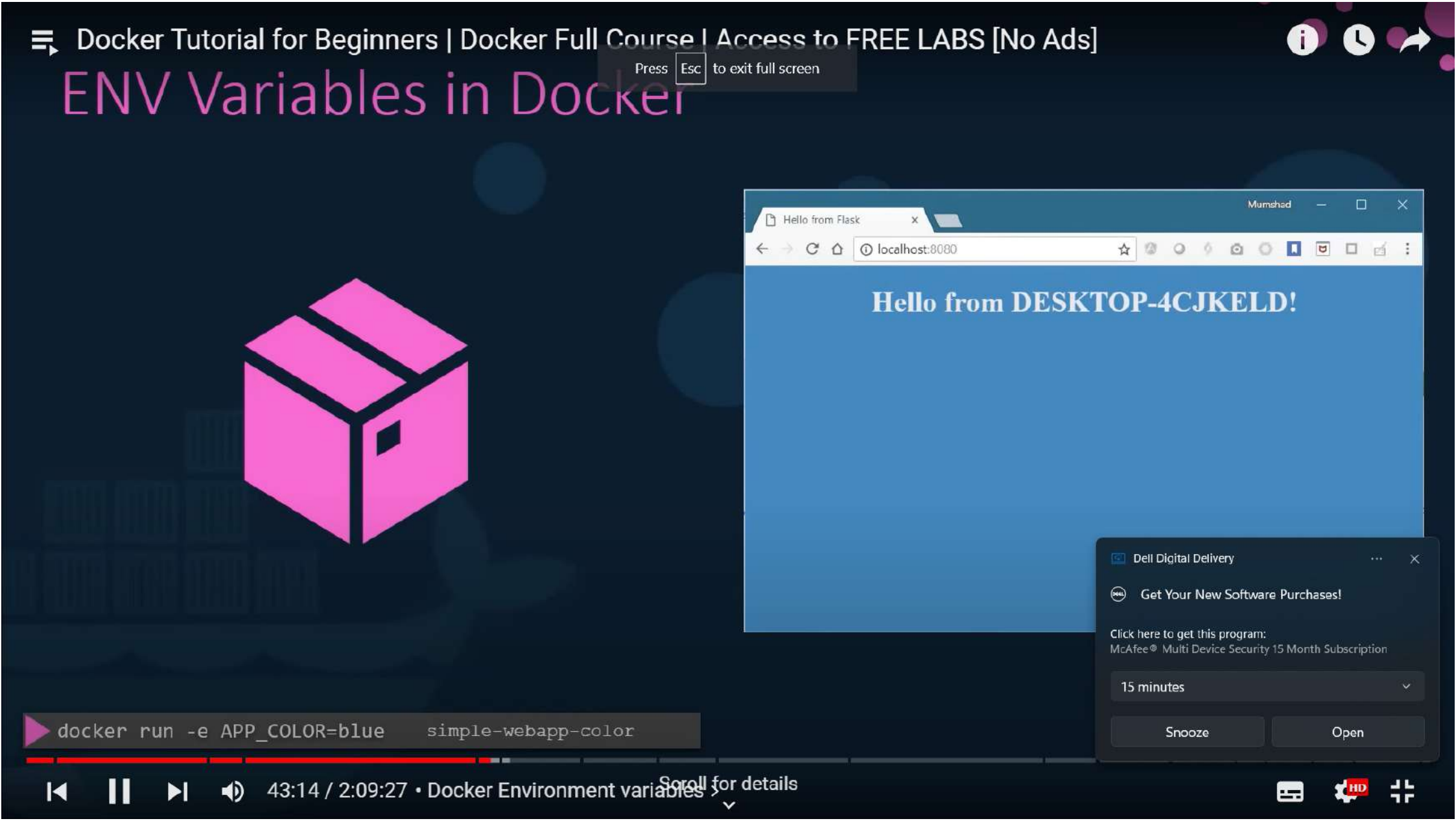
```
docker run -v /opt/datadir:/var/lib/mysql mysql
```



Inspect Container

```
docker inspect blissful_hopper
```


```
[
    {
        "Id": "35505f7810d17291261a43391d4b6c0846594d415ce4f4d0a6ffbf9cc5109048",
        "Name": "/blissful_hopper",
        "Path": "python",
        "Args": [
            "app.py"
        ],
        "State": {
            "Status": "running",
            "Running": true,
        },

        "Mounts": [],
        "Config": {
           "Entrypoint": [
                "python",
                "app.py"
            ],
        },
        "NetworkSettings": {..}
      }
]
```



Docker Tutorial for Beginners | Docker Full Course | Access to FREE LABS [No Ads]
Press Esc to exit full screen
ENV Variables in Docker
Hello from Flask
localhost:8080
Mumshad
Hello from DESKTOP-4CJKELD!
Dell Digital Delivery
Get Your New Software Purchases!
Click here to get this program:
McAfee® Multi Device Security 15 Month Subscription
15 minutes
Snooze
Open
docker run -e APP_COLOR=blue simple-webapp-color
43:14 / 2:09:27 • Docker Environment variables
Scroll for details

Inspect Environment Variable

```
docker inspect blissful_hopper
```


```
[
    {
        "Id": "35505f7810d17291261a43391d4b6c0846594d415ce4f4d0a6ffbf9cc5109048",
        "State": {
            "Status": "running",
            "Running": true,
        },

        "Mounts": [],
        "Config": {
            "Env": [
                "APP_COLOR=blue",
                "LANG=C.UTF-8",
                "GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6AA65421D",
                "PYTHON_VERSION=3.6.6",
                "PYTHON_PIP_VERSION=18.1"
            ],
            "Entrypoint": [
                "python",
                "app.py"
            ],
      }
]
```


(35) Docker Tutorial for Beginner | Labs- Environment Variables - Ko | Saving files and quitting vi | Inspecting Container Environmen

kodekloud.com/topic/labs-environment-variables-3/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

35% Off: Business Plan Sale (lasts till 30th Sept)

KodeKloud

My Account

Task | Hint

48:54

1 2 3 4

Run a container named `blue-app` using image `kodekloud/simple-webapp` and set the environment variable `APP_COLOR` to `blue`. Make the application available on port `38282` on the host. The application listens on port `8080`.

Check

- Name: blue-app
- Image: kodekloud/simple-webapp
- ENV Variable: APP_COLOR=blue

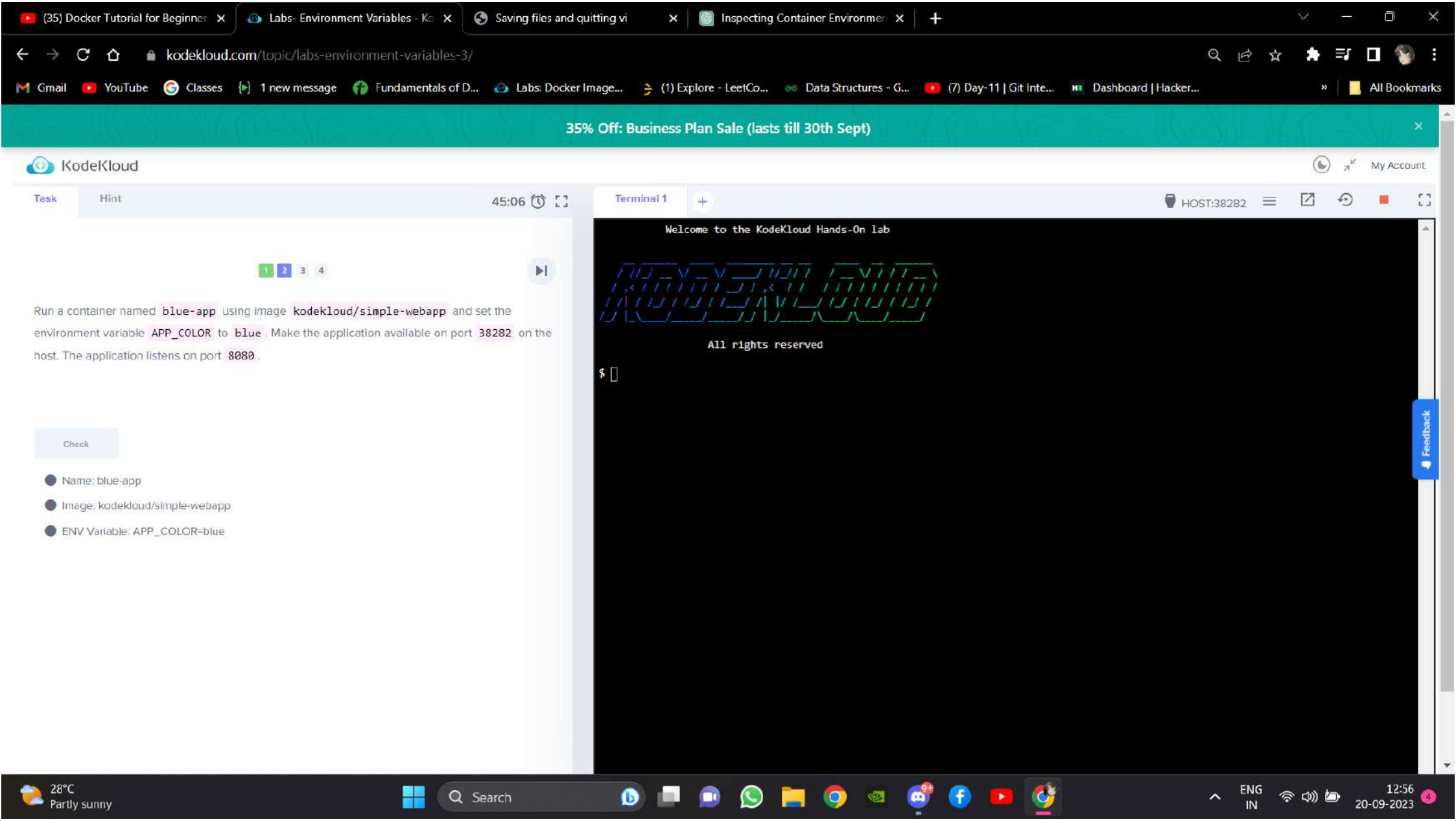
Terminal 1

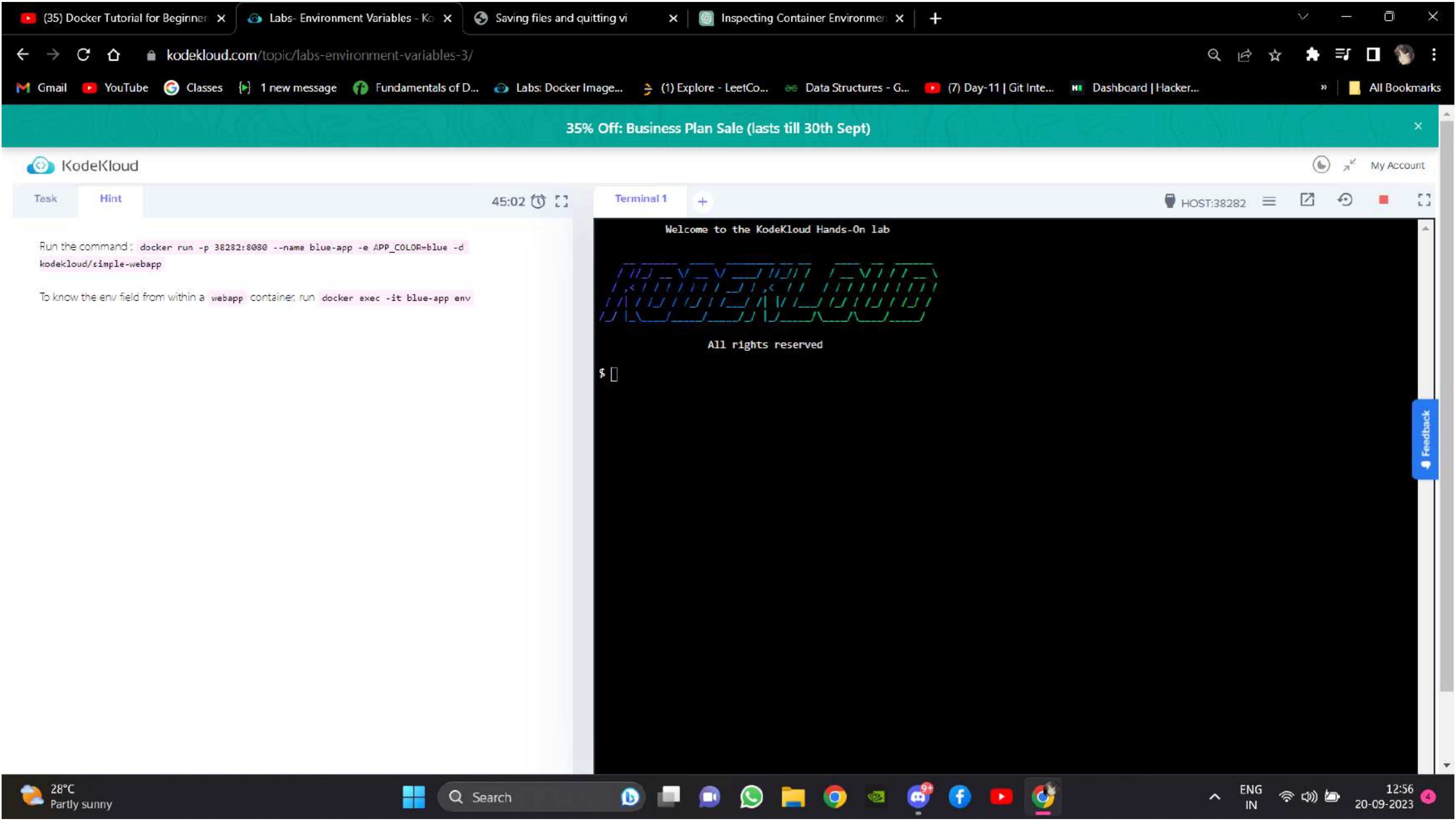
HOST:38282

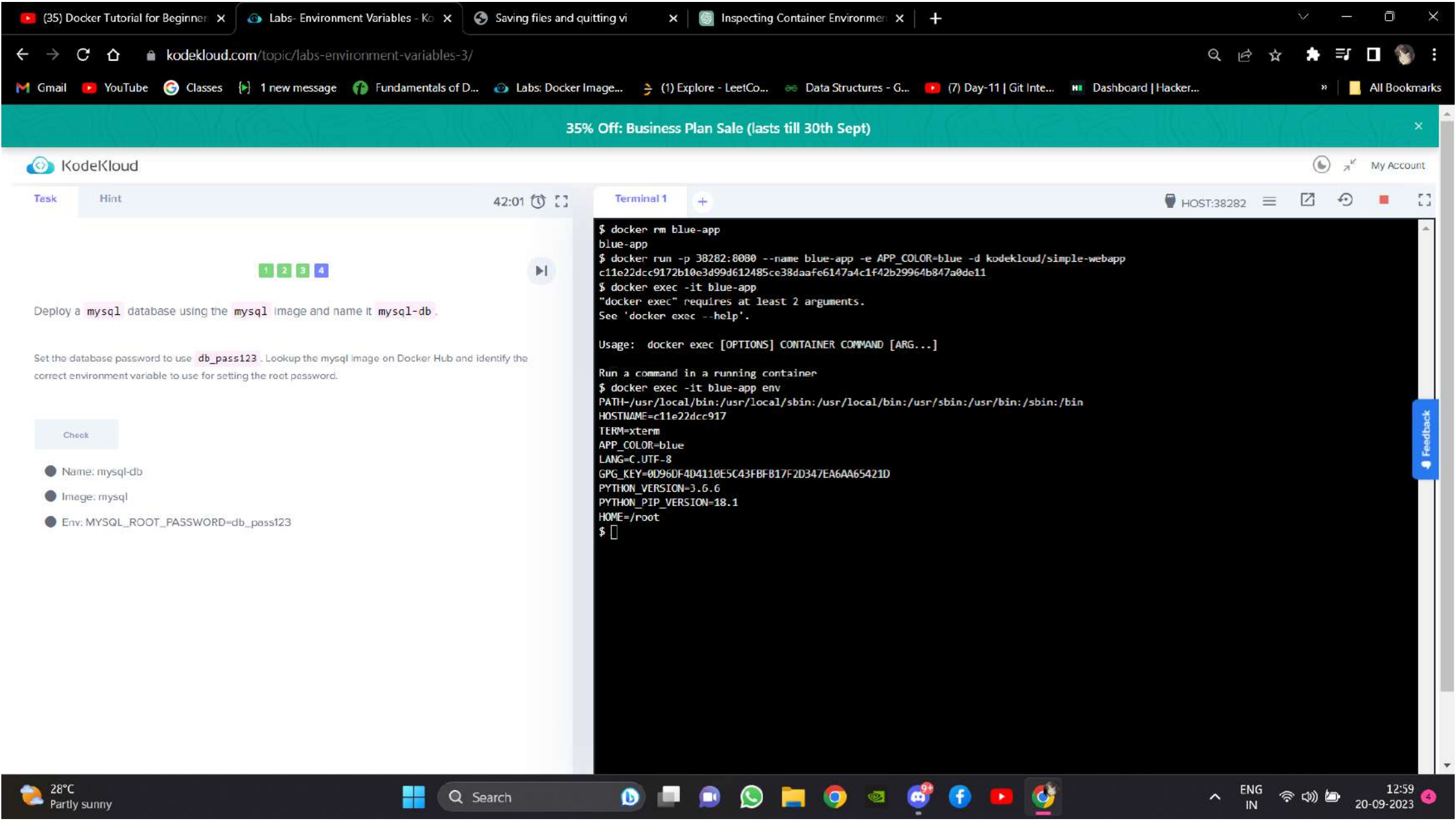
```
$ docker run -p 3282:8080 --name blue-app -e APP_COLOR=blue kodekloud/simple-webapp
 This is a sample web application that displays a colored background.
 A color can be specified in two ways.

 1. As a command line argument with --color as the argument. Accepts one of red,green,blue,blue2,pink,darkblue
 2. As an Environment variable APP_COLOR. Accepts one of red,green,blue,blue2,pink,darkblue
 3. If none of the above then a random color is picked from the above list.
 Note: Command line argument precedes over environment variable.


No Command line argument. Color from environment variable =blue
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Feedback

28°C Partly sunny | Search | ENG IN | 12:52 20-09-2023


(35) Docker Tutorial for Beginner
Labs- Environment Variables - Ko
Saving files and quitting vi
Inspecting Container Environmen
kodekloud.com/topic/labs-environment-variables-3/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
45:06
1 2 3 4
Run a container named blue-app using image kodekloud/simple-webapp and set the environment variable APP_COLOR to blue. Make the application available on port 38282 on the host. The application listens on port 8080.
Check
Name: blue-app
Image: kodekloud/simple-webapp
ENV Variable: APP_COLOR=blue
Terminal 1
HOST:38282
Welcome to the KodeKloud Hands-On lab
All rights reserved
$
Feedback
28°C
Partly sunny
Search
ENG
IN
12:56
20-09-2023


35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
Task
Hint
45:02
Run the command : docker run -p 38282:8080 --name blue-app -e APP_COLOR=blue -d kodekloud/simple-webapp
To know the env field from within a webapp container, run docker exec -it blue-app env
Terminal 1
HOST:38282
Welcome to the KodeKloud Hands-On lab
All rights reserved
$


(35) Docker Tutorial for Beginner
Labs- Environment Variables - Ko
Saving files and quitting vi
Inspecting Container Environmen
kodekloud.com/topic/labs-environment-variables-3/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
42:01
1 2 3 4
Deploy a mysql database using the mysql image and name it mysql-db.
Set the database password to use db_pass123. Lookup the mysql image on Docker Hub and identify the correct environment variable to use for setting the root password.
Check
Name: mysql-db
Image: mysql
Env: MYSQL_ROOT_PASSWORD=db_pass123
Terminal 1
HOST:38282
$ docker rm blue-app
blue-app
$ docker run -p 38282:8080 --name blue-app -e APP_COLOR=blue -d kodekloud/simple-webapp
c11e22dcc9172b10e3d99d612485ce38daafe6147a4c1f42b29964b847a0de11
$ docker exec -it blue-app
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.
Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
Run a command in a running container
$ docker exec -it blue-app env
PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=c11e22dcc917
TERM=xterm
APP_COLOR=blue
LANG=C.UTF-8
GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6AA65421D
PYTHON_VERSION=3.6.6
PYTHON_PIP_VERSION=18.1
HOME=/root
$
Feedback
28°C
Partly sunny
Search
ENG
IN
12:59
20-09-2023

(35) Docker Tutorial for Beginner | Labs- Environment Variables - Ko | Saving files and quitting vi | Inspecting Container Environmen

kodekloud.com/topic/labs-environment-variables-3/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint

41:16

Run the command: `docker run -d -e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db mysql`

To know the env field from within a `mysql-db` container, run `docker exec -it mysql-db env`

Terminal 1

HOST:38282

```
$ docker rm blue-app
blue-app
$ docker run -p 38282:8080 --name blue-app -e APP_COLOR=blue -d kodekloud/simple-webapp
c11e22dcc9172b10e3d99d612485ce38daafe6147a4c1f42b29964b847a0de11
$ docker exec -it blue-app
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
$ docker exec -it blue-app env
PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=c11e22dcc917
TERM=xterm
APP_COLOR=blue
LANG=C.UTF-8
GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6AA65421D
PYTHON_VERSION=3.6.6
PYTHON_PIP_VERSION=18.1
HOME=/root
$ docker run -d -e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db mysql
```

Feedback

28°C Partly sunny | Search | ENG IN | 13:00 20-09-2023

(35) Docker Tutorial for Beginner | Labs- Environment Variables - Ko | Saving files and quitting vi | Inspecting Container Environmen

kodekloud.com/topic/labs-environment-variables-3/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint

40:51

1 2 3 4

Deploy a `mysql` database using the `mysql` image and name it `mysql-db`.

Set the database password to use `db_pass123`. Lookup the mysql image on Docker Hub and identify the correct environment variable to use for setting the root password.

Complete | Next

- ✓ Name: mysql-db
- ✓ Image: mysql
- ✓ Env: MYSQL_ROOT_PASSWORD=db_pass123

Terminal 1

HOST:38282

```
$ docker rm blue-app
blue-app
$ docker run -p 38282:8080 --name blue-app -e APP_COLOR=blue -d kodekloud/simple-webapp
c11e22dcc9172b10e3d99d612485ce38daafe6147a4c1f42b29964b847a0de11
$ docker exec -it blue-app
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
$ docker exec -it blue-app env
PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=c11e22dcc917
TERM=xterm
APP_COLOR=blue
LANG=C.UTF-8
GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6AA65421D
PYTHON_VERSION=3.6.6
PYTHON_PIP_VERSION=18.1
HOME=/root
$ docker run -d -e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db mysql
88ac74660b0813a164ce89024efbcb75c48746636507c14b4cc55d37a3fb3713
$ docker exec -it mysql-db env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=88ac74660b08
TERM=xterm
MYSQL_ROOT_PASSWORD=db_pass123
GOSU_VERSION=1.16
MYSQL_MAJOR=8.0
MYSQL_VERSION=8.0.33-1.el8
MYSQL_SHELL_VERSION=8.0.33-1.el8
HOME=/root
$
```

Feedback

28°C Partly sunny | Search | ENG IN | 13:00 20-09-2023

How to create my own image?

1. OS - Ubuntu
2. Update apt repo
3. Install dependencies using apt
4. Install Python dependencies using pip
5. Copy source code to /opt folder
6. Run the web server using “flask” command

How to create my own image?

Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

1. OS - Ubuntu
2. Update apt repo
3. Install dependencies using apt
4. Install Python dependencies using pip
5. Copy source code to /opt folder
6. Run the web server using “flask” command

How to create my own image?

Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

1. OS - Ubuntu
2. Update apt repo
3. Install dependencies using apt
4. Install Python dependencies using pip
5. Copy source code to /opt folder
6. Run the web server using “flask” command

```
docker build Dockerfile -t mmumshad/my-custom-app
```

How to create my own image?

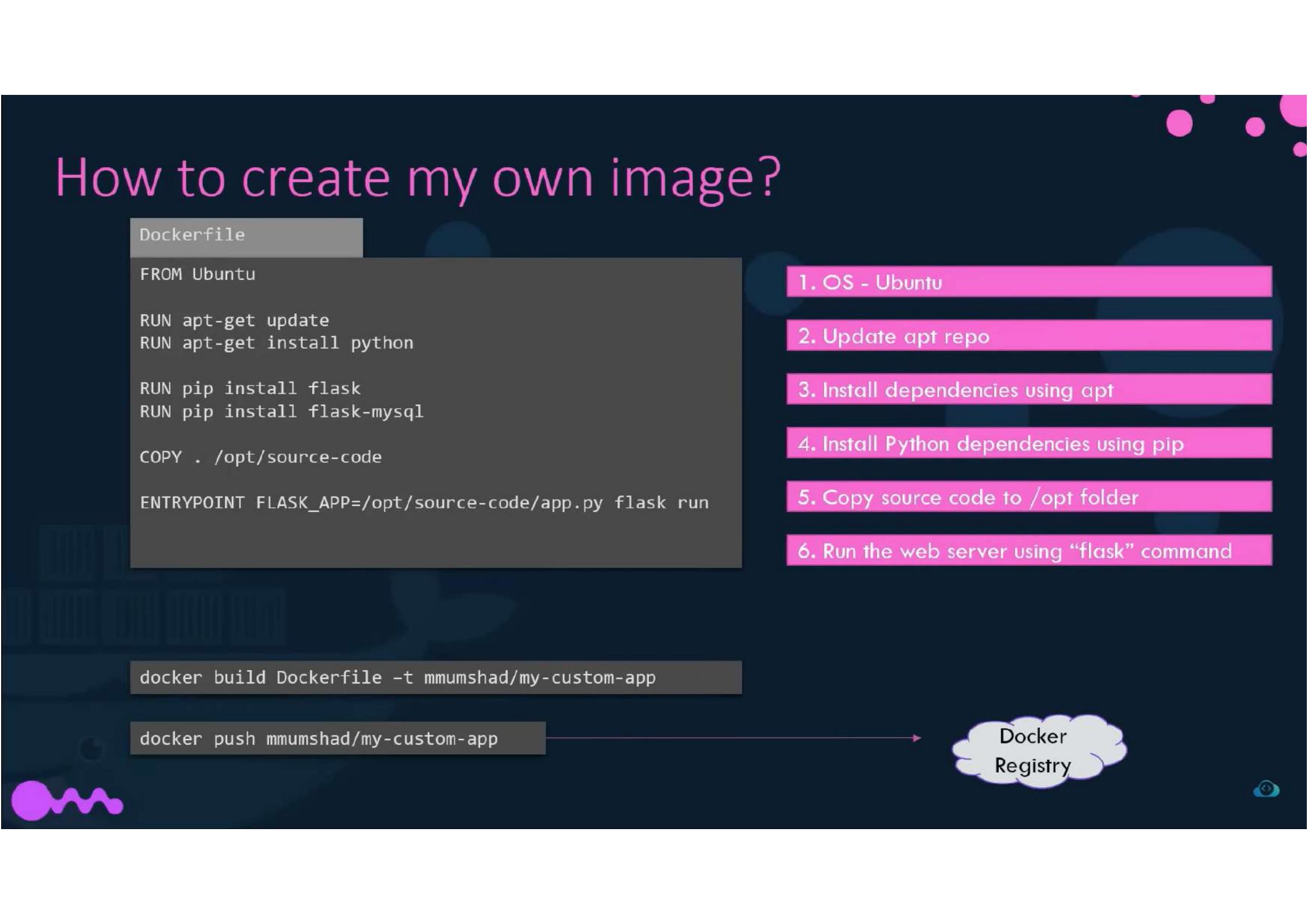
Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

1. OS - Ubuntu
2. Update apt repo
3. Install dependencies using apt
4. Install Python dependencies using pip
5. Copy source code to /opt folder
6. Run the web server using “flask” command

```
docker build Dockerfile -t mmumshad/my-custom-app
```

```
docker push mmumshad/my-custom-app
```



Dockerfile

Dockerfile

INSTRUCTION ARGUMENT

Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

Dockerfile

Dockerfile

INSTRUCTION ARGUMENT

Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

Start from a base OS or another image

Dockerfile

Dockerfile

INSTRUCTION ARGUMENT

Dockerfile

```
FROM Ubuntu

RUN apt-get update
RUN apt-get install python

RUN pip install flask
RUN pip install flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

Start from a base OS or another image

Install all dependencies

Copy source code

Specify Entrypoint

Layered architecture

Dockerfile

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```

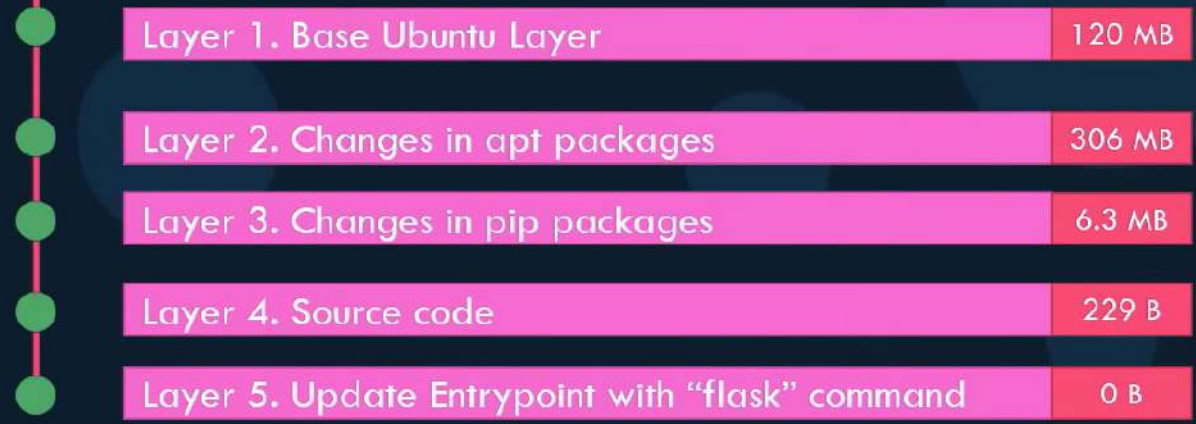
- Layer 1. Base Ubuntu Layer
- Layer 2. Changes in apt packages
- Layer 3. Changes in pip packages
- Layer 4. Source code
- Layer 5. Update Entrypoint with “flask” command

Layered architecture

Dockerfile

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```




```
root@osboxes:/root/simple-webapp-docker # docker history mmumshad/simple-webapp
IMAGE               CREATED             CREATED BY                                      SIZE                COMMENT
1a45ba829f10        About an hour ago   /bin/sh -c #(nop)  ENTRYPOINT ["/bin/sh" "...   0B
37d37ed8fe99        About an hour ago   /bin/sh -c #(nop) COPY file:29b92853d73898...   229B
d6aaebf8ded0        About an hour ago   /bin/sh -c pip install flask flask-mysql        6.39MB
e4c055538e60        About an hour ago   /bin/sh -c apt-get update && apt-get insta...   306MB
ccc7a11d65b1        2 weeks ago         /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>           2 weeks ago         /bin/sh -c mkdir -p /run/systemd && echo '...   7B
<missing>           2 weeks ago         /bin/sh -c sed -i 's/^#\s*\(deb.*universe\...   2.76kB
<missing>           2 weeks ago         /bin/sh -c rm -rf /var/lib/apt/lists/*          0B
<missing>           2 weeks ago         /bin/sh -c set -xe   && echo '#!/bin/sh' >...   745B
<missing>           2 weeks ago         /bin/sh -c #(nop) ADD file:39d3593ea220e68...   120MB
```

Docker build output

```
root@osboxes:/root/simple-webapp-docker # docker build .
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu
 ---> ccc7a11d65b1
Step 2/5 : RUN apt-get update && apt-get install -y python python-setuptools python-dev
 ---> Running in a7840dbfad17
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:4 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [46.3 kB]
Get:5 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [440 kB]
Step 3/5 : RUN pip install flask flask-mysql
 ---> Running in a4a6c9190ba3
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
Collecting flask-mysql
  Downloading Flask_MySQL-1.4.0-py2.py3-none-any.whl
Removing intermediate container a4a6c9190ba3
Step 4/5 : COPY app.py /opt/
 ---> e7cdab17e782
Removing intermediate container faaaaf63c512
Step 5/5 : ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0
 ---> Running in d452c574a8bb
 ---> 9f27c36920bc
Removing intermediate container d452c574a8bb
Successfully built 9f27c36920bc
```

failure

- Layer 1. Base Ubuntu Layer
- Layer 2. Changes in apt packages
- Layer 3. Changes in pip packages
- Layer 4. Source code
- Layer 5. Update Entrypoint with “flask” command

```
docker build Dockerfile -t mmumshad/my-custom-app
```

```
root@osboxes:/root/simple-webapp-docker # docker build .
Sending build context to Docker daemon   5.12kB
Step 1/5 : FROM ubuntu
 ---> ccc7a11d65b1
Step 2/5 : RUN apt-get update && apt-get install -y python python-pip
 ---> Using cache
 ---> e4c055538e60
Step 3/5 : RUN pip install flask
 ---> Running in aacdaccd7403
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
Removing intermediate container aacdaccd7403
Step 4/5 : COPY app.py /opt/
 ---> af41ef57f6f3
Removing intermediate container a49cc8befc8f
Step 5/5 : ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0
 ---> Running in 3d745ff07d5a
 ---> 910416d360b6
Removing intermediate container 3d745ff07d5a
Successfully built 910416d360b6
```

```
FROM Ubuntu

CMD sleep 5
```

```
docker run ubuntu-sleeper sleep 10
```

Command at Startup: sleep 10

```
FROM Ubuntu

ENTRYPOINT ["sleep"]
```

```
docker run ubuntu-sleeper 10
```

10

Command at Startup: sleep

```
FROM Ubuntu

CMD sleep 5
```

```
docker run ubuntu-sleeper sleep 10
```

Command at Startup: sleep 10

```
FROM Ubuntu

ENTRYPOINT ["sleep"]
```

```
docker run ubuntu-sleeper 10
```

Command at Startup: sleep 10

```
docker run ubuntu-sleeper
sleep: missing operand
Try 'sleep --help' for more information.
```

Command at Startup:sleep

```
FROM Ubuntu

ENTRYPOINT ["sleep"]

CMD ["5"]
```

```
docker run ubuntu-sleeper
sleep: missing operand
Try 'sleep --help' for more information.
```

Command at Startup: sleep 5

```
docker run ubuntu-sleeper 10
```

Command at Startup: sleep 10

FROM Ubuntu

ENTRYPOINT [“sleep”]

CMD [“5”]

```
docker run ubuntu-sleeper
```

```
sleep: missing operand
Try 'sleep --help' for more information.
```

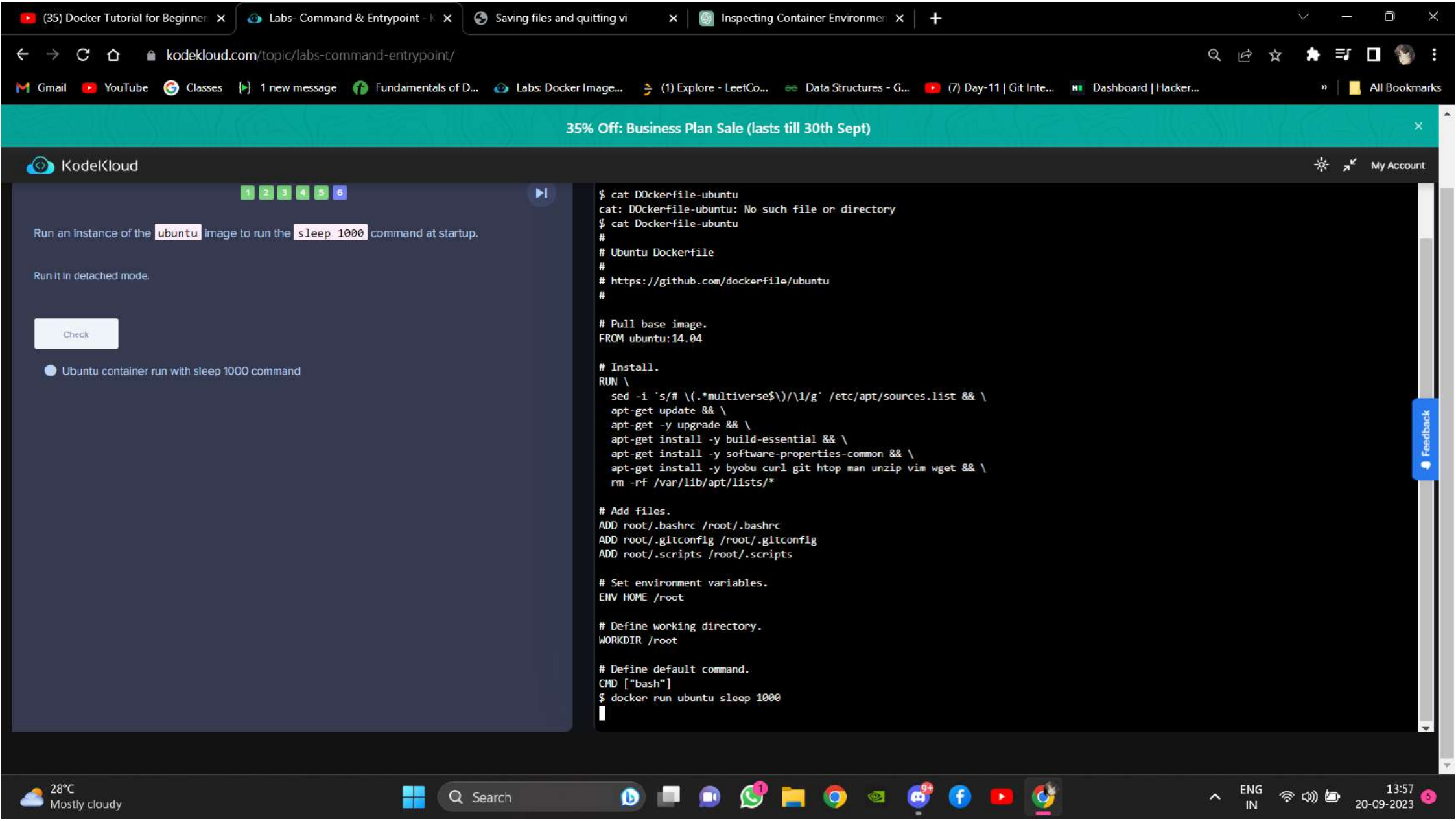
Command at Startup: sleep 5

```
docker run ubuntu-sleeper 10
```

Command at Startup: sleep 10

```
docker run --entrypoint sleep2.0 ubuntu-sleeper 10
```

Command at Startup: sleep2.0 10


(35) Docker Tutorial for Beginner
Labs- Command & Entrypoint
Saving files and quitting vi
Inspecting Container Environmer
kodekloud.com/topic/labs-command-entrypoint/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
1 2 3 4 5 6
Run an instance of the ubuntu image to run the sleep 1000 command at startup.
Run it in detached mode.
Check
Ubuntu container run with sleep 1000 command
$ cat DOckerfile-ubuntu
cat: DOckerfile-ubuntu: No such file or directory
$ cat Dockerfile-ubuntu
#
Ubuntu Dockerfile
#
https://github.com/dockerfile/ubuntu
#
Pull base image.
FROM ubuntu:14.04
Install.
RUN \
sed -i 's/# \(.*multiverse$\)/\1/g' /etc/apt/sources.list && \
apt-get update && \
apt-get -y upgrade && \
apt-get install -y build-essential && \
apt-get install -y software-properties-common && \
apt-get install -y byobu curl git htop man unzip vim wget && \
rm -rf /var/lib/apt/lists/*
Add files.
ADD root/.bashrc /root/.bashrc
ADD root/.gitconfig /root/.gitconfig
ADD root/.scripts /root/.scripts
Set environment variables.
ENV HOME /root
Define working directory.
WORKDIR /root
Define default command.
CMD ["bash"]
$ docker run ubuntu sleep 1000
Feedback
28°C
Mostly cloudy
Search
ENG
IN
13:57
20-09-2023

Default networks


```
docker run ubuntu
```


```
docker run Ubuntu --network=none
```


```
docker run Ubuntu --network=host
```

Default networks

Bridge

```
docker run ubuntu
```

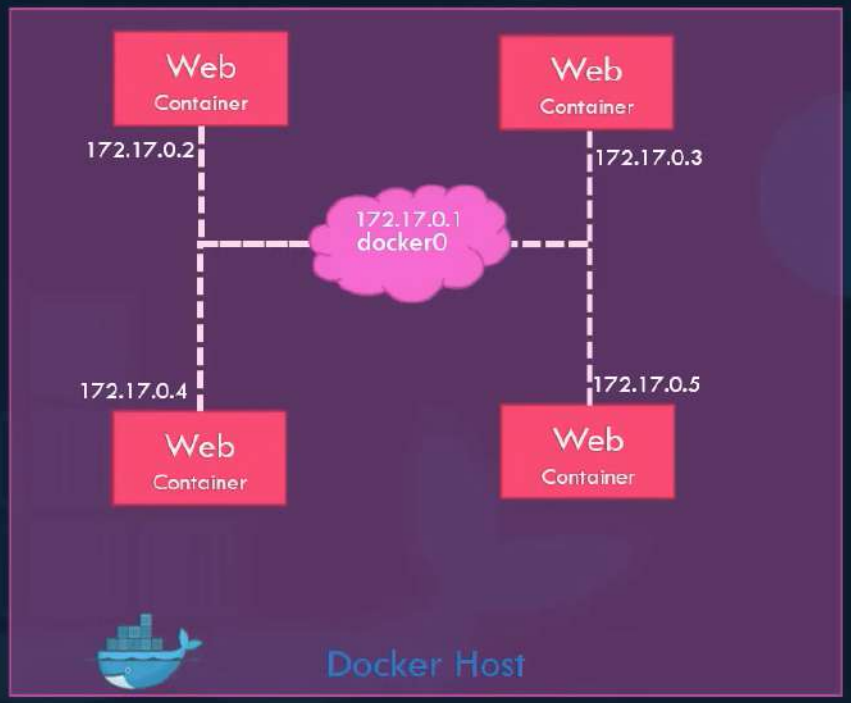
none

```
docker run Ubuntu --network=none
```

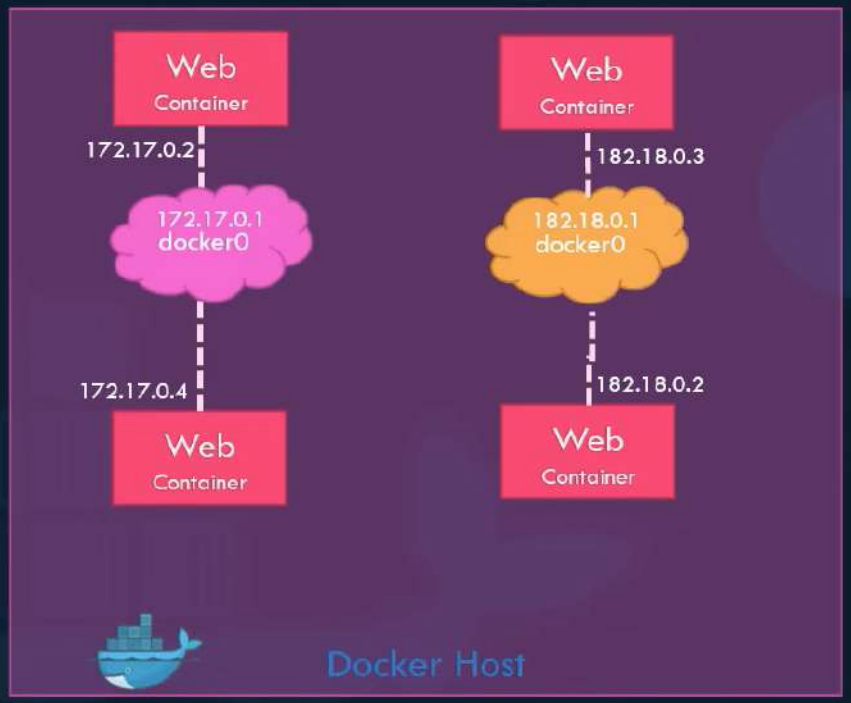
host

```
docker run Ubuntu --network=host
```

User-defined networks

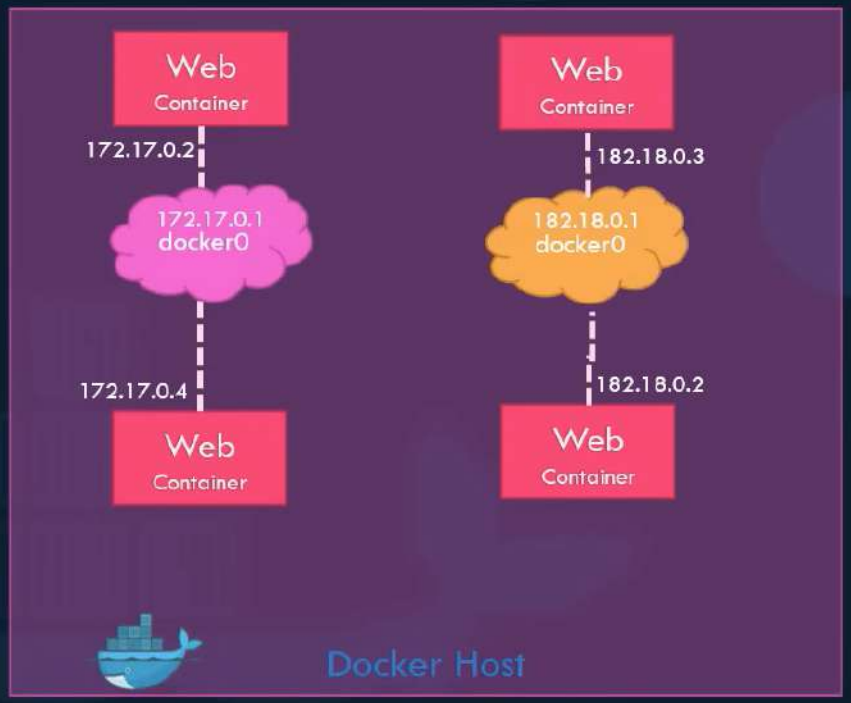


User-defined networks




```
docker network create \
    --driver bridge \
    --subnet 182.18.0.0/16
    custom-isolated-network
```

User-defined networks




```
docker network create \
    --driver bridge \
    --subnet 182.18.0.0/16
    custom-isolated-network
```

```
docker network ls
```

```
root@osboxes:/root # docker network ls
NETWORK ID          NAME                          DRIVER              SCOPE
dba0fb9370fe        bridge                        bridge              local
46d476b87cd9        customer-isolated-network     bridge              local
6de685cec1ce        docker_gwbridge               bridge              local
e29d188b4e47        host                          host                local
mmrho7vsb9rm        ingress                       overlay             swarm
d9f11695f0d6        none                          null                local
d371b4009142        simplewebappdocker_default    bridge              local
```

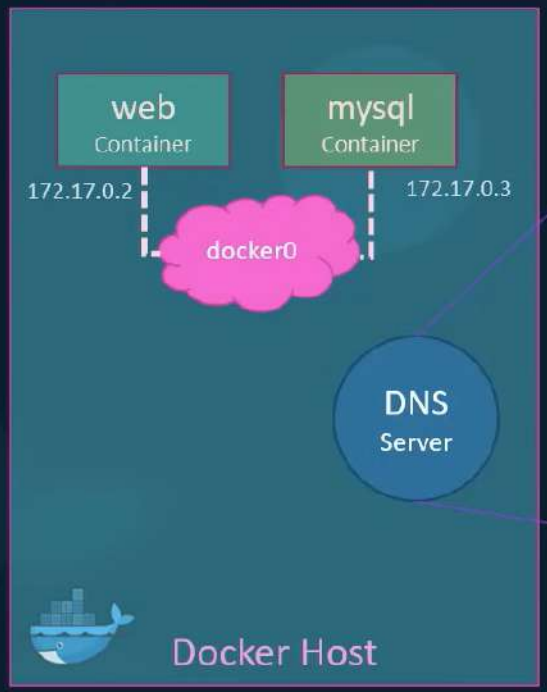
Inspect Network

```
docker inspect blissful_hopper
```

```
[
    {
        "Id": "35505f7810d17291261a43391d4b6c0846594d415ce4f4d0a6ffbf9cc5109048",
        "Name": "/blissful_hopper",
        "NetworkSettings": {
            "Bridge": "",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.6",
            "MacAddress": "02:42:ac:11:00:06",
            "Networks": {
                "bridge": {
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.6",
                    "MacAddress": "02:42:ac:11:00:06",
                }
            }
        }
]
```

Embedded DNS

```
mysql.connect(   mysql       )
```



Host	IP
web	172.17.0.2
mysql	172.17.0.3

(35) Docker Tutorial for Beginner | Labs: Docker Compose - KodeKl | Saving files and quitting vi | Inspecting Container Environmen

kodekloud.com/topic/labs-docker-compose/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint | 59:01

1 2 3 4 5

First create a redis database container called `redis`, image `redis:alpine`.

If you are unsure, check the hints section for the exact commands.

Check

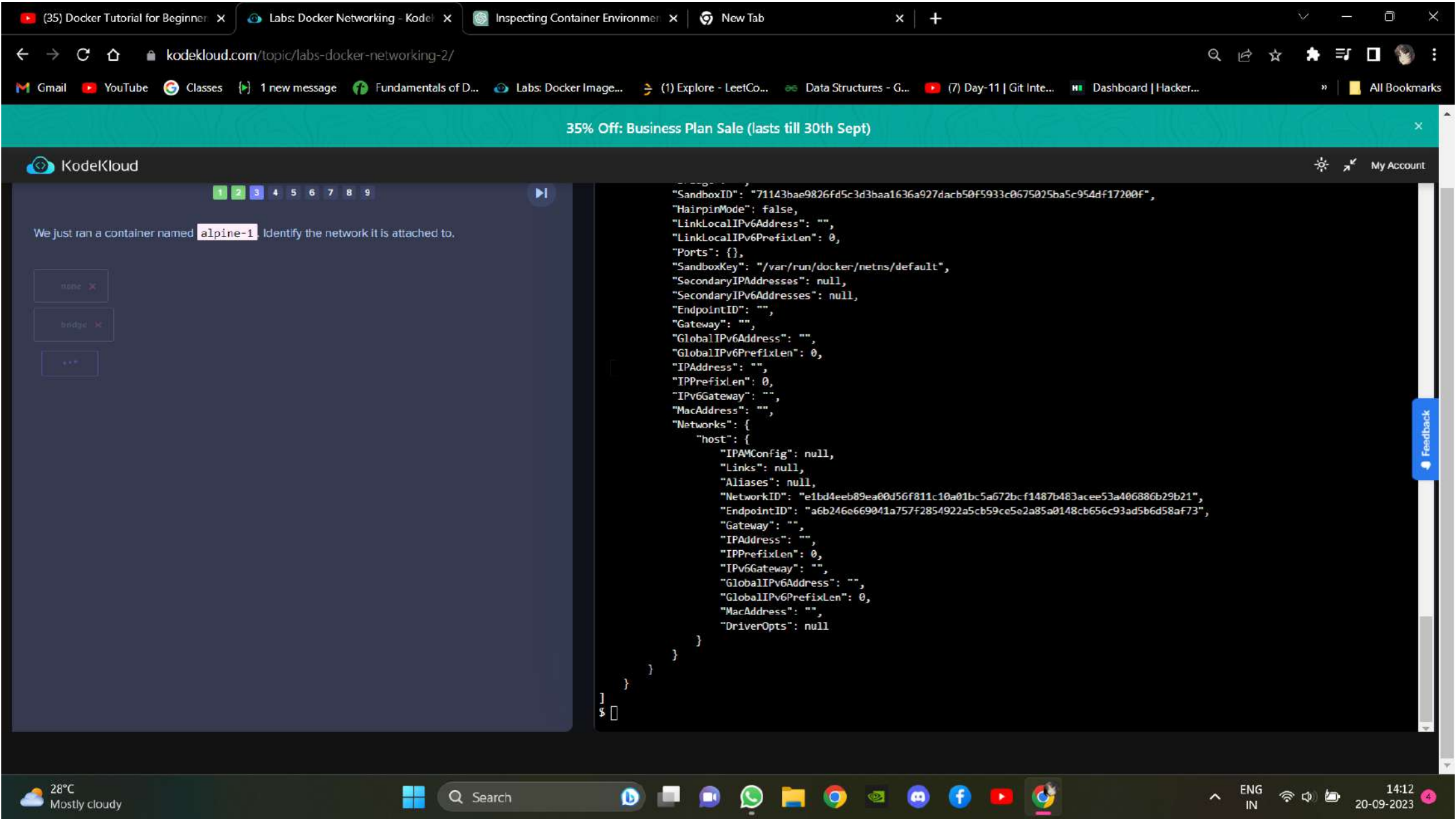
"redis" container running?

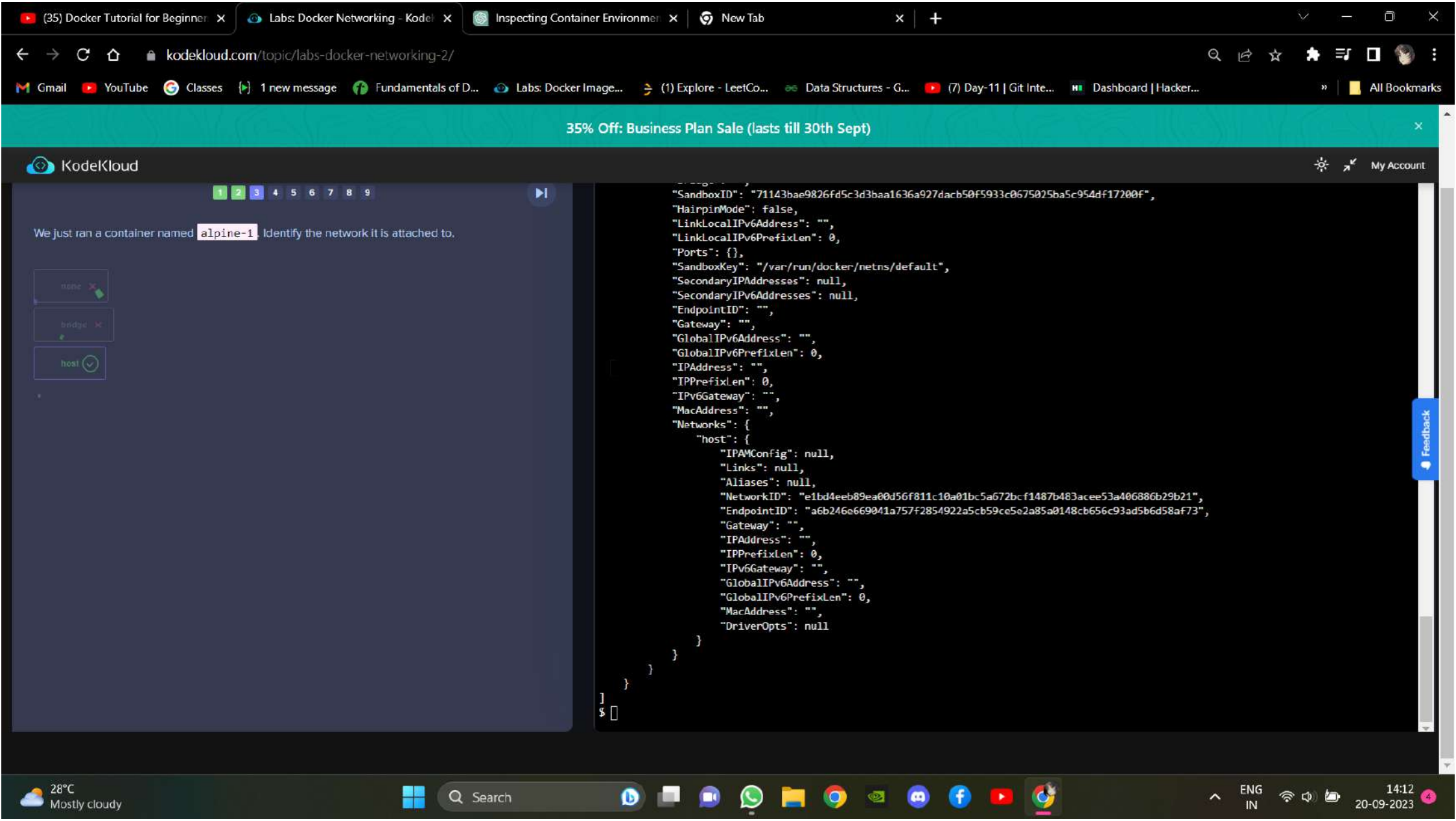
Terminal 1 | Click-Counter

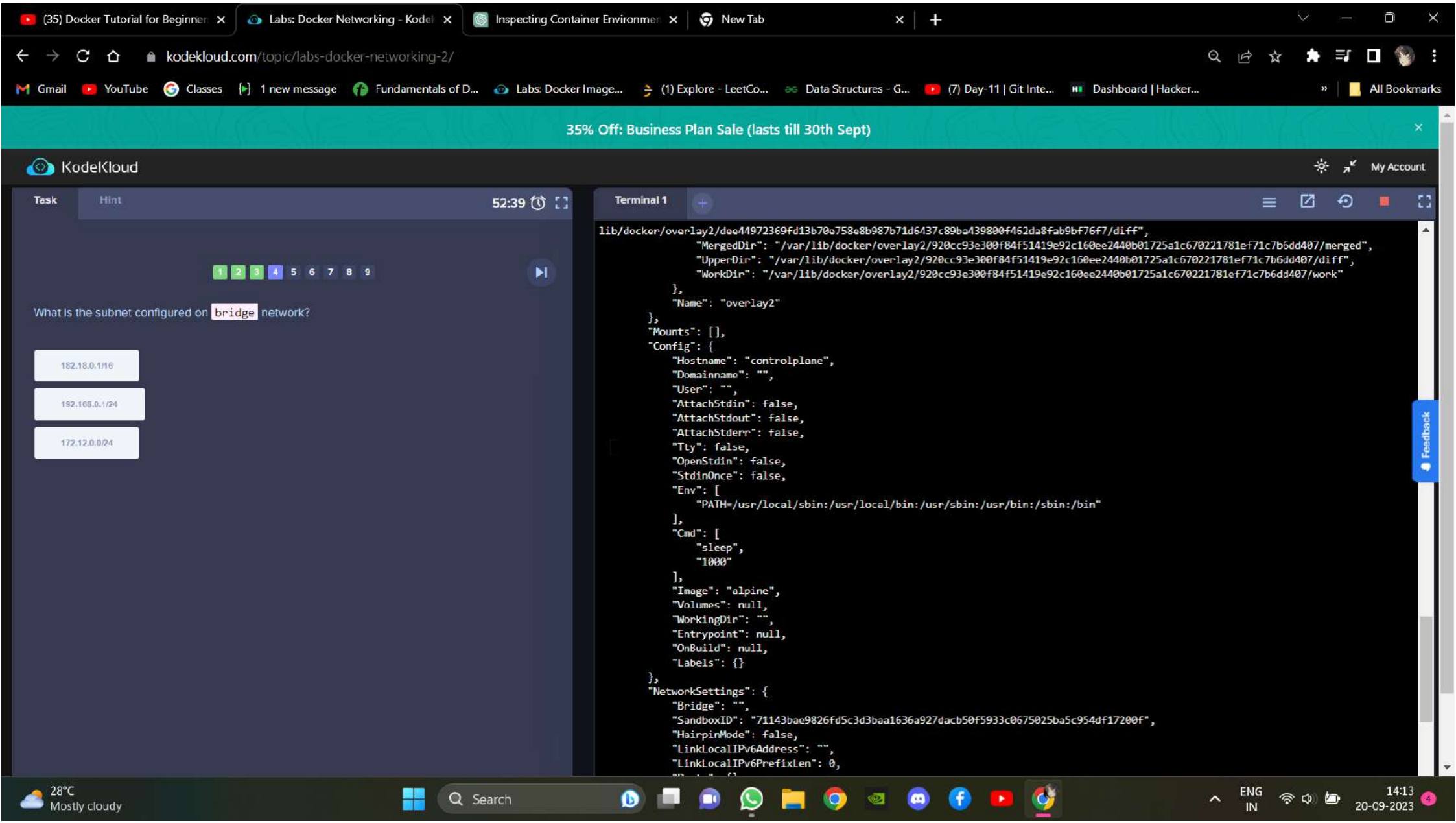
```
        Welcome to the KodeKloud Hands-On lab

                 All rights reserved

$
```

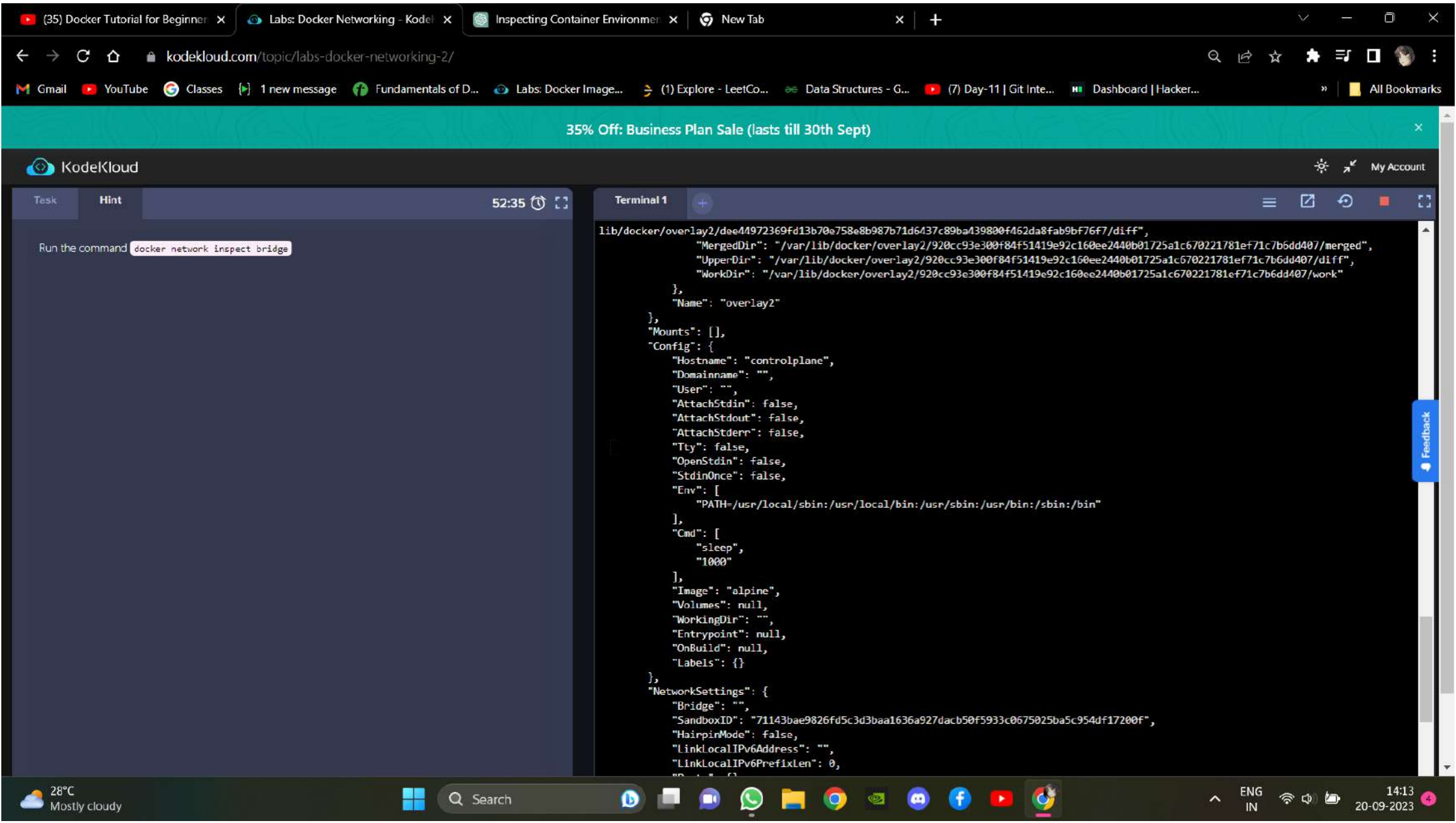
Feedback

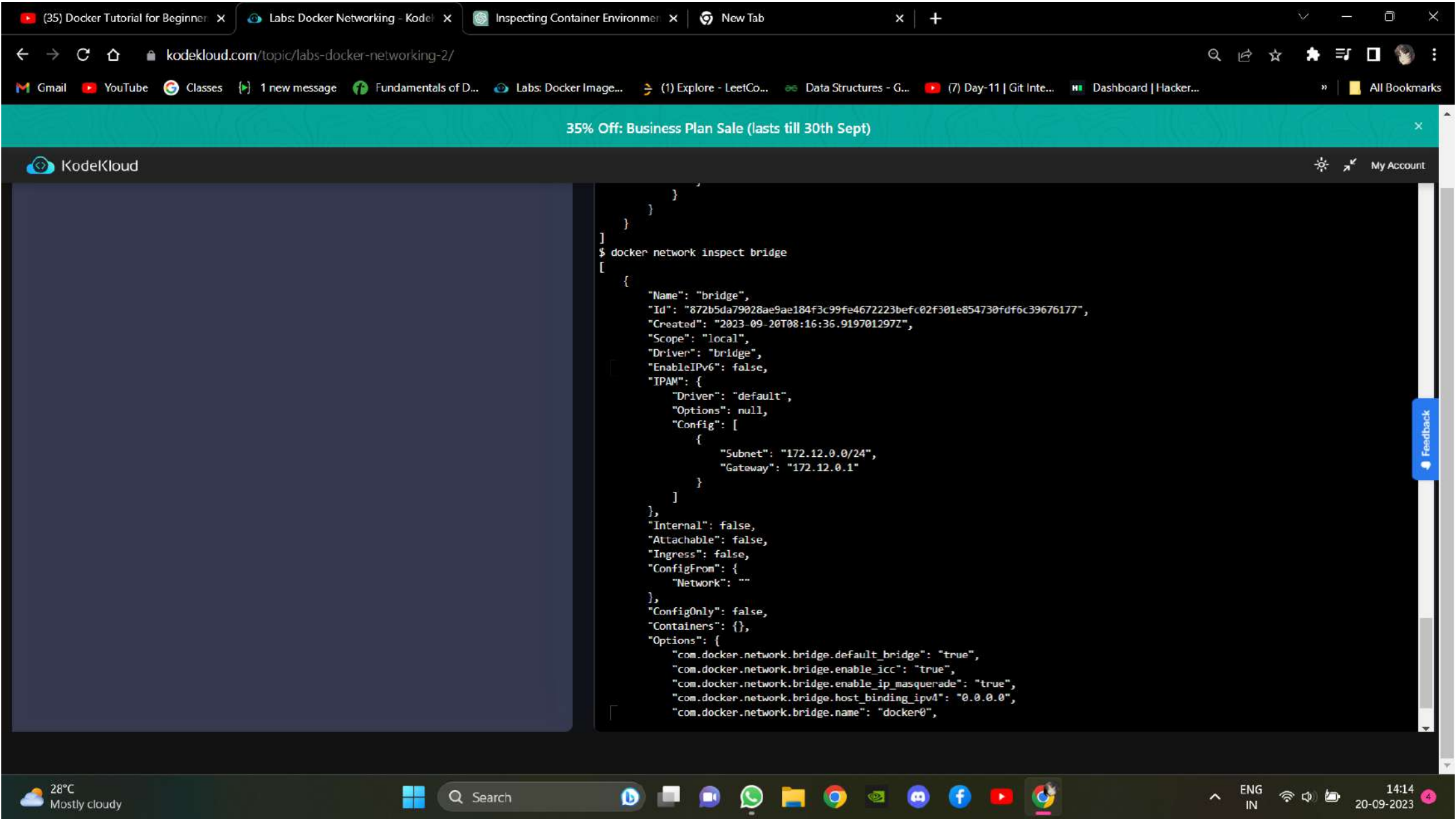
28°C Mostly cloudy | Search | ENG IN | 14:05 20-09-2023

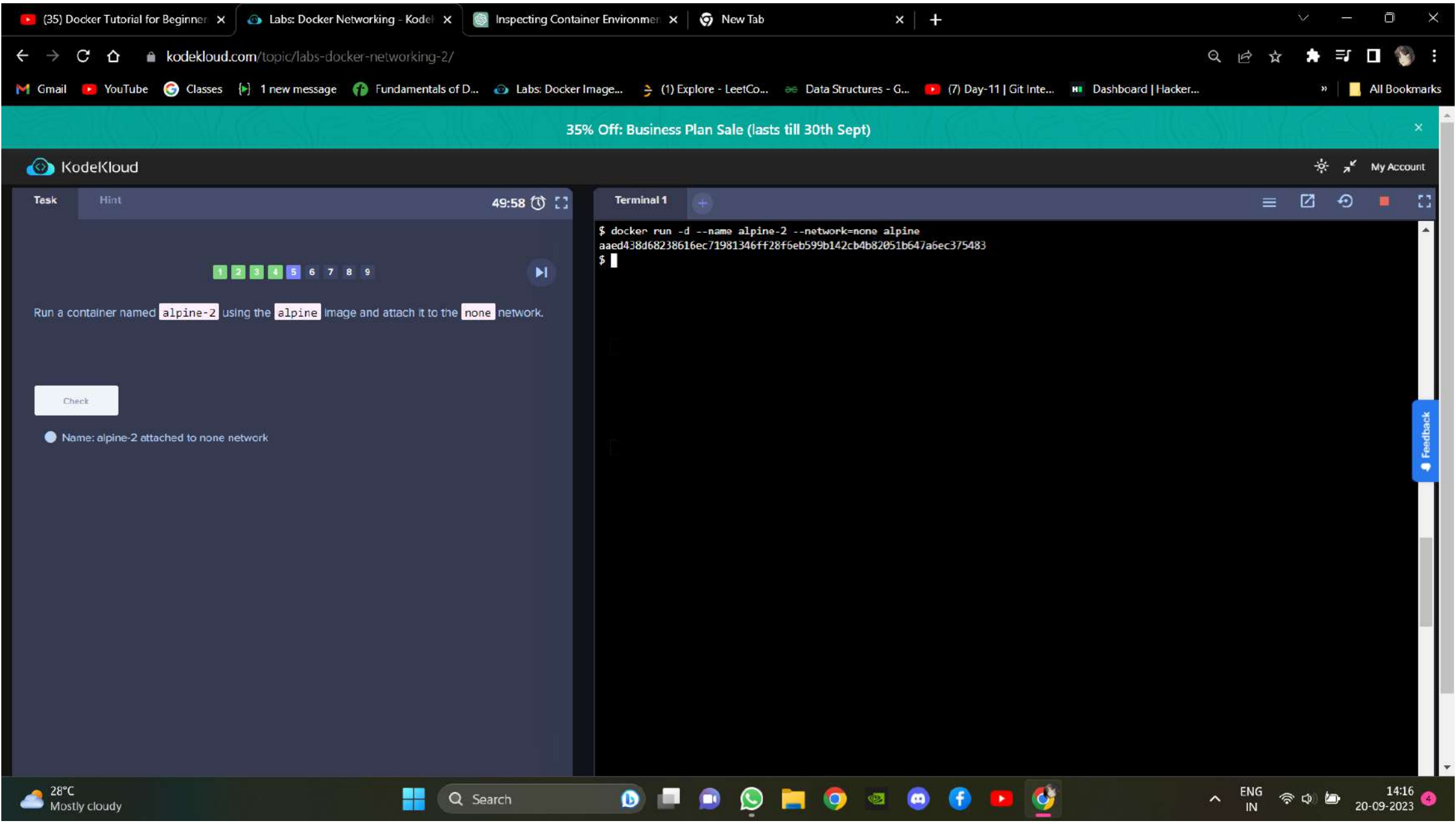

(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
1 2 3 4 5 6 7 8 9
We just ran a container named alpine-1. Identify the network it is attached to.
none
bridge
...
"SandboxID": "71143bae9826fd5c3d3baa1636a927dacb50f5933c0675025ba5c954df17200f",
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"Ports": {},
"SandboxKey": "/var/run/docker/netns/default",
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "",
"Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "",
"IPPrefixLen": 0,
"IPv6Gateway": "",
"MacAddress": "",
"Networks": {
"host": {
"IPAMConfig": null,
"Links": null,
"Aliases": null,
"NetworkID": "e1bd4eeb89ea00d56f811c10a01bc5a672bcf1487b483acee53a406886b29b21",
"EndpointID": "a6b246e669041a757f2854922a5cb59ce5e2a85a0148cb656c93ad5b6d58af73",
"Gateway": "",
"IPAddress": "",
"IPPrefixLen": 0,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"MacAddress": "",
"DriverOpts": null
}
}
}
}
]
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:12
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
1 2 3 4 5 6 7 8 9
We just ran a container named alpine-1. Identify the network it is attached to.
none
bridge
host
"SandboxID": "71143bae9826fd5c3d3baa1636a927dacb50f5933c0675025ba5c954df17200f",
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"Ports": {},
"SandboxKey": "/var/run/docker/netns/default",
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "",
"Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "",
"IPPrefixLen": 0,
"IPv6Gateway": "",
"MacAddress": "",
"Networks": {
"host": {
"IPAMConfig": null,
"Links": null,
"Aliases": null,
"NetworkID": "e1bd4eeb89ea00d56f811c10a01bc5a672bcf1487b483acee53a406886b29b21",
"EndpointID": "a6b246e669041a757f2854922a5cb59ce5e2a85a0148cb656c93ad5b6d58af73",
"Gateway": "",
"IPAddress": "",
"IPPrefixLen": 0,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"MacAddress": "",
"DriverOpts": null
}
}
}
}
]
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:12
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
52:39
1 2 3 4 5 6 7 8 9
What is the subnet configured on bridge network?
182.18.0.1/16
192.168.0.1/24
172.12.0.0/24
Terminal 1
lib/docker/overlay2/dee44972369fd13b70e758e8b987b71d6437c89ba439800f462da8fab9bf76f7/diff",
"MergedDir": "/var/lib/docker/overlay2/920cc93e300f84f51419e92c160ee2440b01725a1c670221781ef71c7b6dd407/merged",
"UpperDir": "/var/lib/docker/overlay2/920cc93e300f84f51419e92c160ee2440b01725a1c670221781ef71c7b6dd407/diff",
"WorkDir": "/var/lib/docker/overlay2/920cc93e300f84f51419e92c160ee2440b01725a1c670221781ef71c7b6dd407/work"
},
"Name": "overlay2"
},
"Mounts": [],
"Config": {
"Hostname": "controlplane",
"Domainname": "",
"User": "",
"AttachStdin": false,
"AttachStdout": false,
"AttachStderr": false,
"Tty": false,
"OpenStdin": false,
"StdinOnce": false,
"Env": [
"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
],
"Cmd": [
"sleep",
"1000"
],
"Image": "alpine",
"Volumes": null,
"WorkingDir": "",
"Entrypoint": null,
"OnBuild": null,
"Labels": {}
},
"NetworkSettings": {
"Bridge": "",
"SandboxID": "71143bae9826fd5c3d3baa1636a927dacb50f5933c0675025ba5c954df17200f",
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:13
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
52:35
Run the command docker network inspect bridge
Terminal 1
lib/docker/overlay2/dee44972369fd13b70e758e8b987b71d6437c89ba439800f462da8fab9bf76f7/diff",
"MergedDir": "/var/lib/docker/overlay2/920cc93e300f84f51419e92c160ee2440b01725a1c670221781ef71c7b6dd407/merged",
"UpperDir": "/var/lib/docker/overlay2/920cc93e300f84f51419e92c160ee2440b01725a1c670221781ef71c7b6dd407/diff",
"WorkDir": "/var/lib/docker/overlay2/920cc93e300f84f51419e92c160ee2440b01725a1c670221781ef71c7b6dd407/work"
},
"Name": "overlay2"
},
"Mounts": [],
"Config": {
"Hostname": "controlplane",
"Domainname": "",
"User": "",
"AttachStdin": false,
"AttachStdout": false,
"AttachStderr": false,
"Tty": false,
"OpenStdin": false,
"StdinOnce": false,
"Env": [
"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
],
"Cmd": [
"sleep",
"1000"
],
"Image": "alpine",
"Volumes": null,
"WorkingDir": "",
"Entrypoint": null,
"OnBuild": null,
"Labels": {}
},
"NetworkSettings": {
"Bridge": "",
"SandboxID": "71143bae9826fd5c3d3baa1636a927dacb50f5933c0675025ba5c954df17200f",
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:13
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
$ docker network inspect bridge
[
{
"Name": "bridge",
"Id": "872b5da79028ae9ae184f3c99fe4672223befc02f301e854730fdf6c39676177",
"Created": "2023-09-20T08:16:36.9197012972",
"Scope": "local",
"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
"Driver": "default",
"Options": null,
"Config": [
{
"Subnet": "172.12.0.0/24",
"Gateway": "172.12.0.1"
}
]
},
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
"Network": ""
},
"ConfigOnly": false,
"Containers": {},
"Options": {
"com.docker.network.bridge.default_bridge": "true",
"com.docker.network.bridge.enable_icc": "true",
"com.docker.network.bridge.enable_ip_masquerade": "true",
"com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
"com.docker.network.bridge.name": "docker0",
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:14
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
49:58
1 2 3 4 5 6 7 8 9
Run a container named alpine-2 using the alpine image and attach it to the none network.
Check
Name: alpine-2 attached to none network
Terminal 1
$ docker run -d --name alpine-2 --network=none alpine
aaed438d68238616ec71981346ff28f6eb599b142cb4b82051b647a6ec375483
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:16
20-09-2023

(35) Docker Tutorial for Beginner | Labs: Docker Networking - Kode | Inspecting Container Environmer | New Tab

kodekloud.com/topic/labs-docker-networking-2/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint | 47:43

1 2 3 4 5 6 7 8 9

Create a new network named `wp-mysql-network` using the `bridge` driver. Allocate subnet `182.18.0.1/24`. Configure `Gateway 182.18.0.1`

Check

- Name: wp-mysql-network
- Driver: bridge
- subnet: 182.18.0.1/24
- Gateway: 182.18.0.1

Terminal 1

```
$ docker network create -- driber bridge --subnet 182.18.0.1/24 --gateway 182.18.0.1 wp-mysql-network
"docker network create" requires exactly 1 argument.
See 'docker network create --help'.

Usage:  docker network create [OPTIONS] NETWORK

Create a network
$ docker network create -- driver bridge --subnet 182.18.0.1/24 --gateway 182.18.0.1 wp-mysql-network
"docker network create" requires exactly 1 argument.
See 'docker network create --help'.

Usage:  docker network create [OPTIONS] NETWORK

Create a network
$ docker network create --driver bridge --subnet 182.18.0.1/24 --gateway 182.18.0.1 wp-mysql-network
64b44cb3d41e0aac2a421e0faa98f21a21481e1831dbac970cd2704879f02cb8
$
```

Feedback

28°C Mostly cloudy | Search | ENG IN | 14:18 20-09-2023

(35) Docker Tutorial for Beginner | Labs: Docker Networking - Kode | Inspecting Container Environmer | New Tab

kodekloud.com/topic/labs-docker-networking-2/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint | 38:35

1 2 3 4 5 6 7 8 9

Deploy a `mysql` database using the `mysql:5.6` image and name it `mysql-db`. Attach it to the newly created network `wp-mysql-network`

Set the database password to use `db_pass123`. The environment variable to set is `MYSQL_ROOT_PASSWORD`.

Check

- ✕ Name: mysql-db
- ✕ Image: mysql:5.6
- ✕ Env: MYSQL_ROOT_PASSWORD=db_pass123
- ✕ Network: wp-mysql-network

Terminal 1

```
$ docker run -d --name mysql-db e MYSQL_ROOT_PASSWORD=db_pass123 --network wp-mysql-network mysql:5.6
8e2e49bcde3a4bbee054e49662a2bea9a5080efa93c46506f8e1dd362364bca1
$ docker stop 8e
8e
$ docker rm 82
Error: No such container: 82
$ docker rm 8e
8e
$ docker run -d - e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db --network wp-mysql-network mysql:5.6
docker: invalid reference format.
See 'docker run --help'.
$ docker run -d -e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db --network wp-mysql-network mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
35b2232c987e: Pull complete
fc55c00e48f2: Pull complete
0030405130e3: Pull complete
e1fef7f6a8d1: Pull complete
1c76272398bb: Pull complete
f57e698171b6: Pull complete
f5b825b269c0: Pull complete
dcb0af686073: Pull complete
27bbfeb886d1: Extracting [---------------------------------->                ]  44.01MB/64.27MB
6f70cc868145: Download complete
1f6637f4600d: Download complete
```

Feedback

28°C Mostly cloudy | Search | ENG IN | 14:27 20-09-2023

(35) Docker Tutorial for Beginner | Labs: Docker Networking - Kode | Inspecting Container Environmer | New Tab

kodekloud.com/topic/labs-docker-networking-2/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud | My Account

Task | Hint | 38:27

1 2 3 4 5 6 7 8 9

Deploy a `mysql` database using the `mysql:5.6` image and name it `mysql-db`. Attach it to the newly created network `wp-mysql-network`

Set the database password to use `db_pass123`. The environment variable to set is `MYSQL_ROOT_PASSWORD`.

Complete | Next

- Name: mysql-db
- Image: mysql:5.6
- Env: MYSQL_ROOT_PASSWORD=db_pass123
- Network: wp-mysql-network

Terminal 1

```
$ docker run -d --name mysql-db e MYSQL_ROOT_PASSWORD=db_pass123 --network wp-mysql-network mysql:5.6
8e2e49bcde3a4bbee054e49662a2bea9a5080efa93c46506f8e1dd362364bca1
$ docker stop 8e
8e
$ docker rm 82
Error: No such container: 82
$ docker rm 8e
8e
$ docker run -d - e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db --network wp-mysql-network mysql:5.6
docker: invalid reference format.
See 'docker run --help'.
$ docker run -d -e MYSQL_ROOT_PASSWORD=db_pass123 --name mysql-db --network wp-mysql-network mysql:5.6
Unable to find image 'mysql:5.6' locally
5.6: Pulling from library/mysql
35b2232c987e: Pull complete
fc55c00e48f2: Pull complete
0030405130e3: Pull complete
e1fef7f6a8d1: Pull complete
1c76272398bb: Pull complete
f57e698171b6: Pull complete
f5b825b269c0: Pull complete
dcb0af686073: Pull complete
27bbfeb886d1: Pull complete
6f70cc868145: Pull complete
1f6637f4600d: Pull complete
Digest: sha256:20575ecebe6216036d25dab5903808211f1e9ba63dc7825ac20cb975e34cfcae
Status: Downloaded newer image for mysql:5.6
db5d0f55c5c1879b7839a4fd99db3d772c4c7ad861ebbab37ab59487821e1c51
$
```

Feedback

28°C Mostly cloudy | Search | ENG IN | 14:27 20-09-2023

(35) Docker Tutorial for Beginner | Labs: Docker Networking - Kode | Inspecting Container Environmer | New Tab

kodekloud.com/topic/labs-docker-networking-2/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint | 34:39

1 2 3 4 5 6 7 8 9

Deploy a web application named `webapp` using the `kodekloud/simple-webapp-mysql` image. Expose the port to 38080 on the host.

The application makes use of two environment variable:

1: `DB_Host` with the value `mysql-db`.

2: `DB_Password` with the value `db_pass123`.

Make sure to attach it to the newly created network called `wp-mysql-network`.

Also make sure to link the `MySQL` and the `webapp` container.

Check

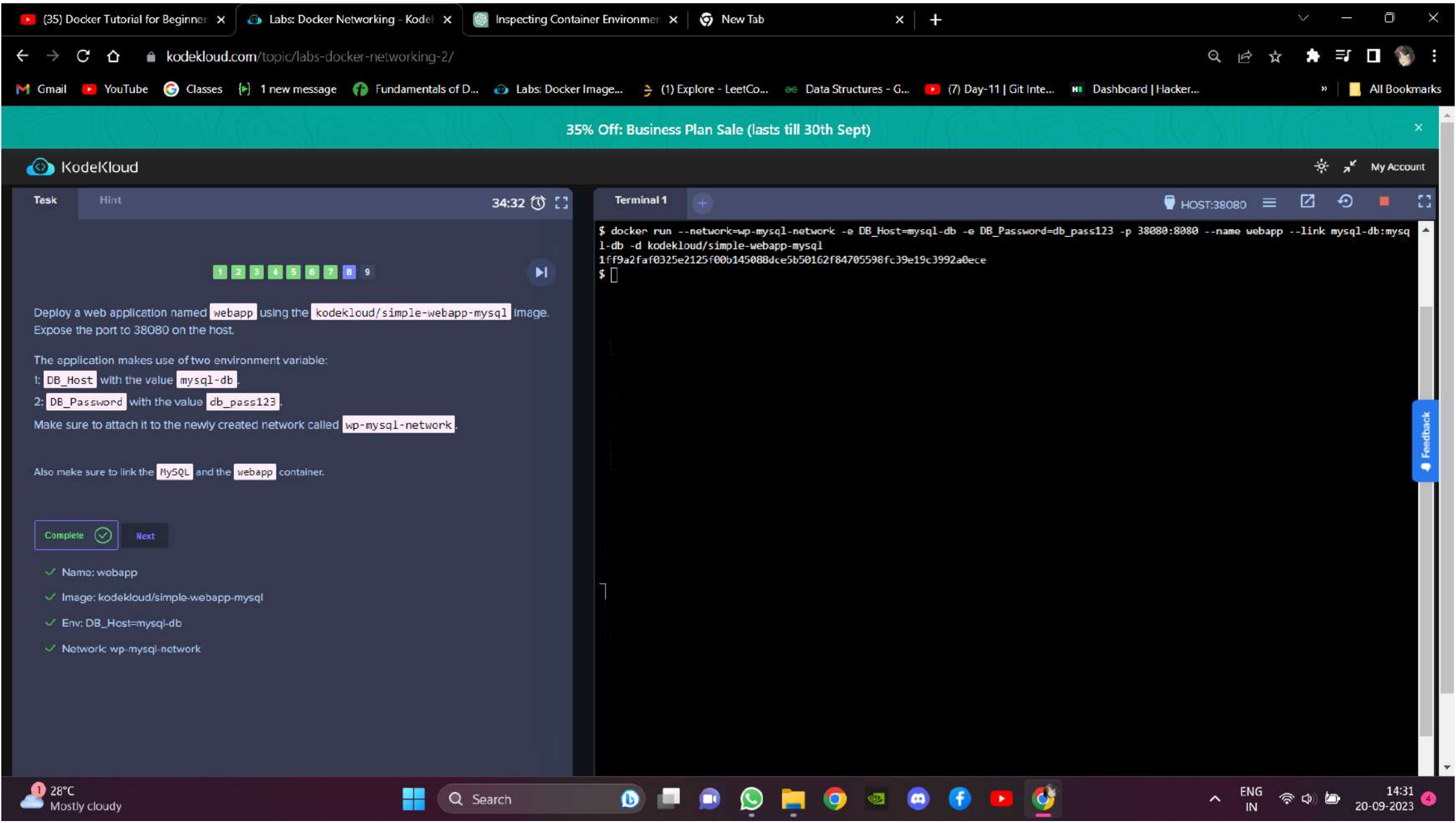
- Name: webapp
- Image: kodekloud/simple-webapp-mysql
- Env: DB_Host=mysql-db
- Network: wp-mysql-network

Terminal 1 | HOST:38080

```
$ docker run --network=wp-mysql-network -e DB_Host=mysql-db -e DB_Password=db_pass123 -p 38080:8080 --name webapp --link mysql-db:mysql-db -d kodekloud/simple-webapp-mysql
```

Feedback

28°C Mostly cloudy | Search | ENG IN | 14:31 20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Networking - Kodel
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-networking-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
34:32
1 2 3 4 5 6 7 8 9
Deploy a web application named webapp using the kodekloud/simple-webapp-mysql image. Expose the port to 38080 on the host.
The application makes use of two environment variable:
1: DB_Host with the value mysql-db.
2: DB_Password with the value db_pass123.
Make sure to attach it to the newly created network called wp-mysql-network.
Also make sure to link the MySQL and the webapp container.
Complete
Next
Name: webapp
Image: kodekloud/simple-webapp-mysql
Env: DB_Host=mysql-db
Network: wp-mysql-network
Terminal 1
HOST:38080
$ docker run --network=wp-mysql-network -e DB_Host=mysql-db -e DB_Password=db_pass123 -p 38080:8080 --name webapp --link mysql-db:mysql-db -d kodekloud/simple-webapp-mysql
1ff9a2faf0325e2125f00b145088dce5b50162f84705598fc39e19c3992a0ece
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:31
20-09-2023

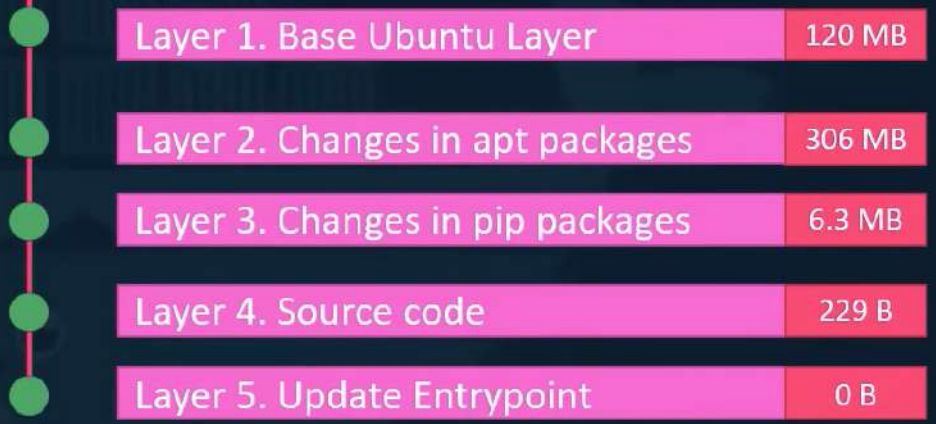
File system

Layered architecture

Dockerfile

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask
run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```



Layered architecture

Dockerfile

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```

- Layer 1. Base Ubuntu Layer — 120 MB
- Layer 2. Changes in apt packages — 306 MB
- Layer 3. Changes in pip packages — 6.3 MB
- Layer 4. Source code — 229 B
- Layer 5. Update Entrypoint — 0 B

Dockerfile2

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY app2.py /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app2.py flask run
```

```
docker build Dockerfile2 -t mmumshad/my-custom-app-2
```

- Layer 1. Base Ubuntu Layer
- Layer 2. Changes in apt packages
- Layer 3. Changes in pip packages
- Layer 4. Source code
- Layer 5. Update Entrypoint

Layered architecture

Dockerfile

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY . /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app.py flask
run
```

```
docker build Dockerfile -t mmumshad/my-custom-app
```

- Layer 1. Base Ubuntu Layer — 120 MB
- Layer 2. Changes in apt packages — 306 MB
- Layer 3. Changes in pip packages — 6.3 MB
- Layer 4. Source code — 229 B
- Layer 5. Update Entrypoint — 0 B

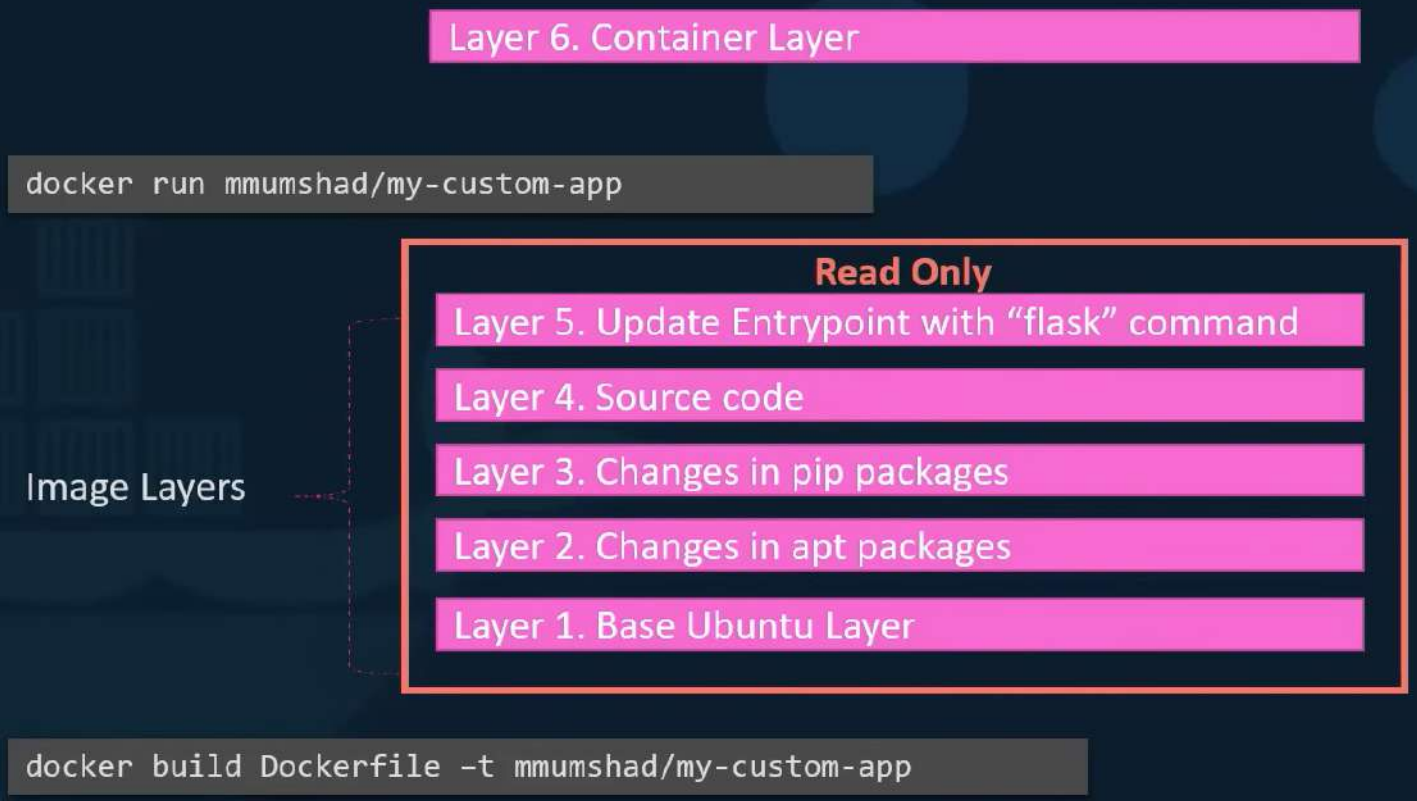
Dockerfile2

```
FROM Ubuntu

RUN apt-get update && apt-get -y install python

RUN pip install flask flask-mysql

COPY app2.py /opt/source-code

ENTRYPOINT FLASK_APP=/opt/source-code/app2.py flask
run
```

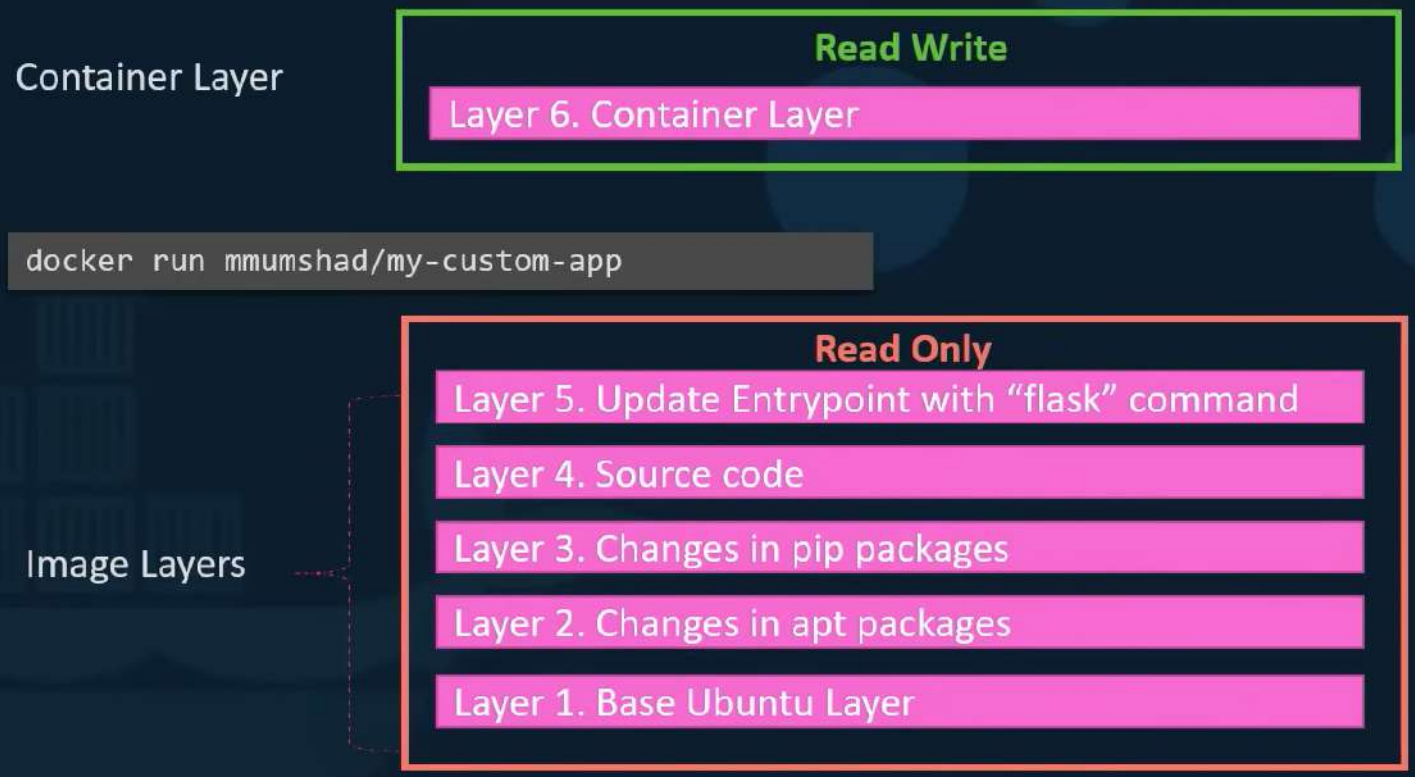
```
docker build Dockerfile2 -t mmumshad/my-custom-app-2
```

- Layer 1. Base Ubuntu Layer — 0 MB
- Layer 2. Changes in apt packages — 0 MB
- Layer 3. Changes in pip packages — 0 MB
- Layer 4. Source code — 229 B
- Layer 5. Update Entrypoint — 0 B

Layered architecture

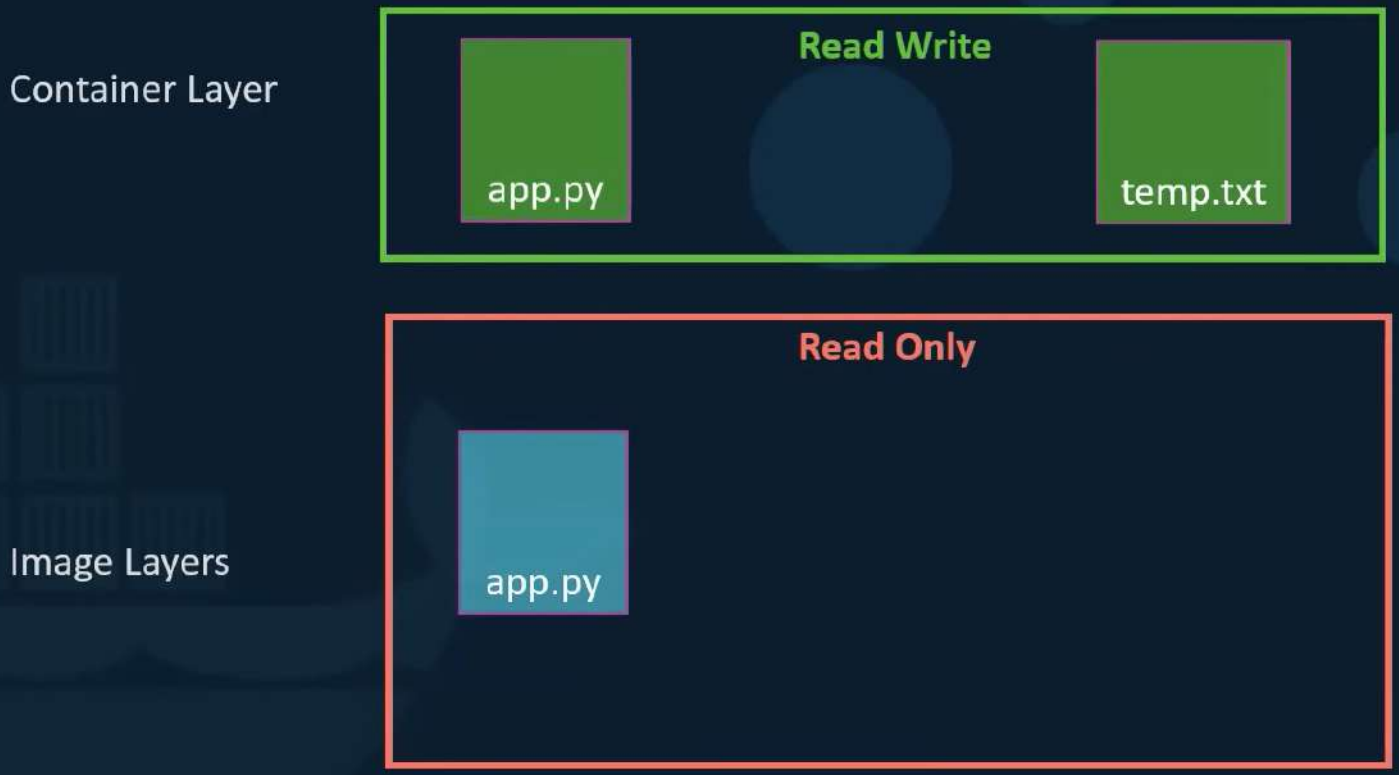


Layered architecture




Container Layer
Read Write
app.py
temp.txt
Read Only
app.py
Image Layers

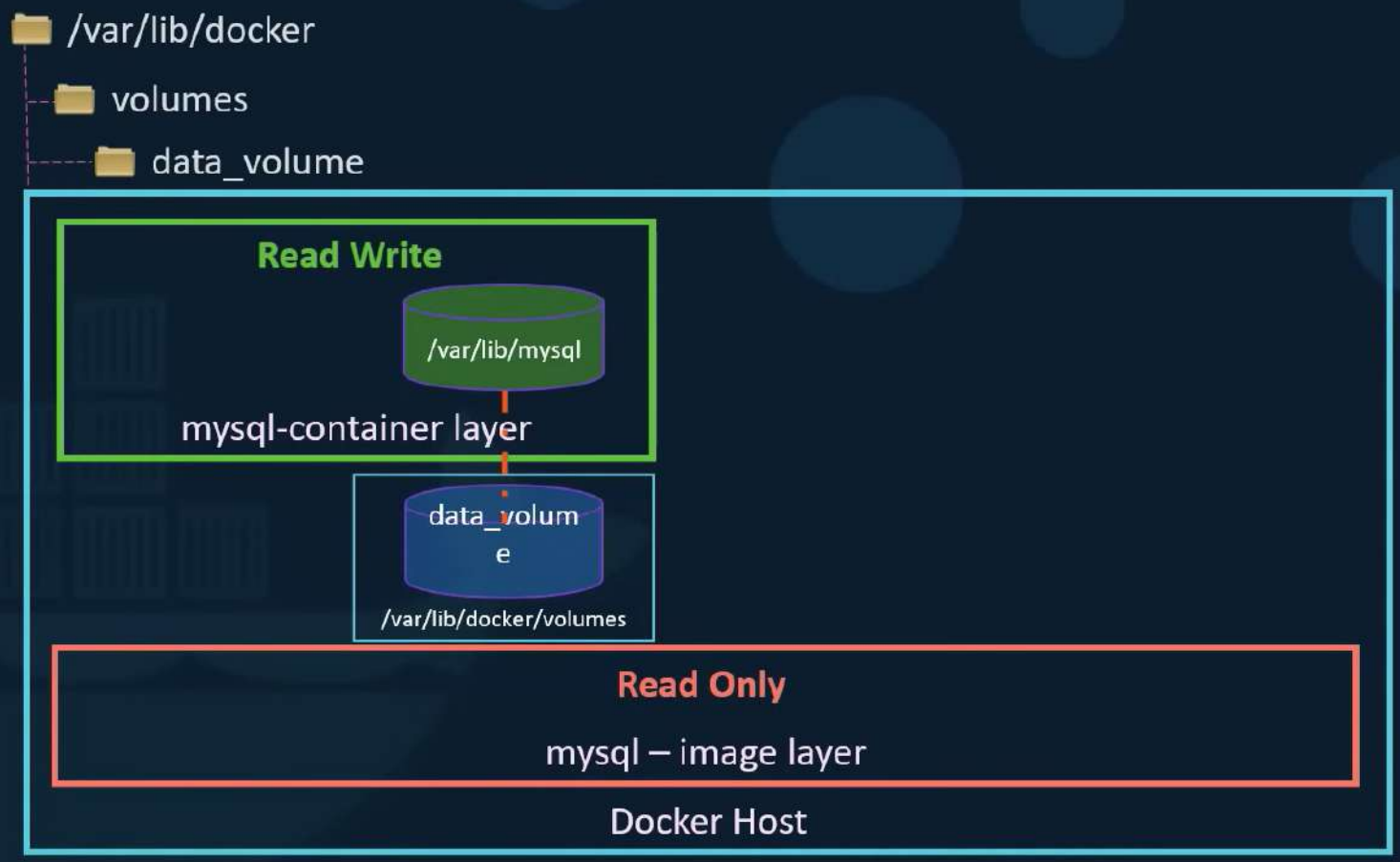
COPY-ON-WRITE



volumes

```
docker run -v data_volume:/var/lib/mysql mysql
```

```
docker volume create data_volume
```



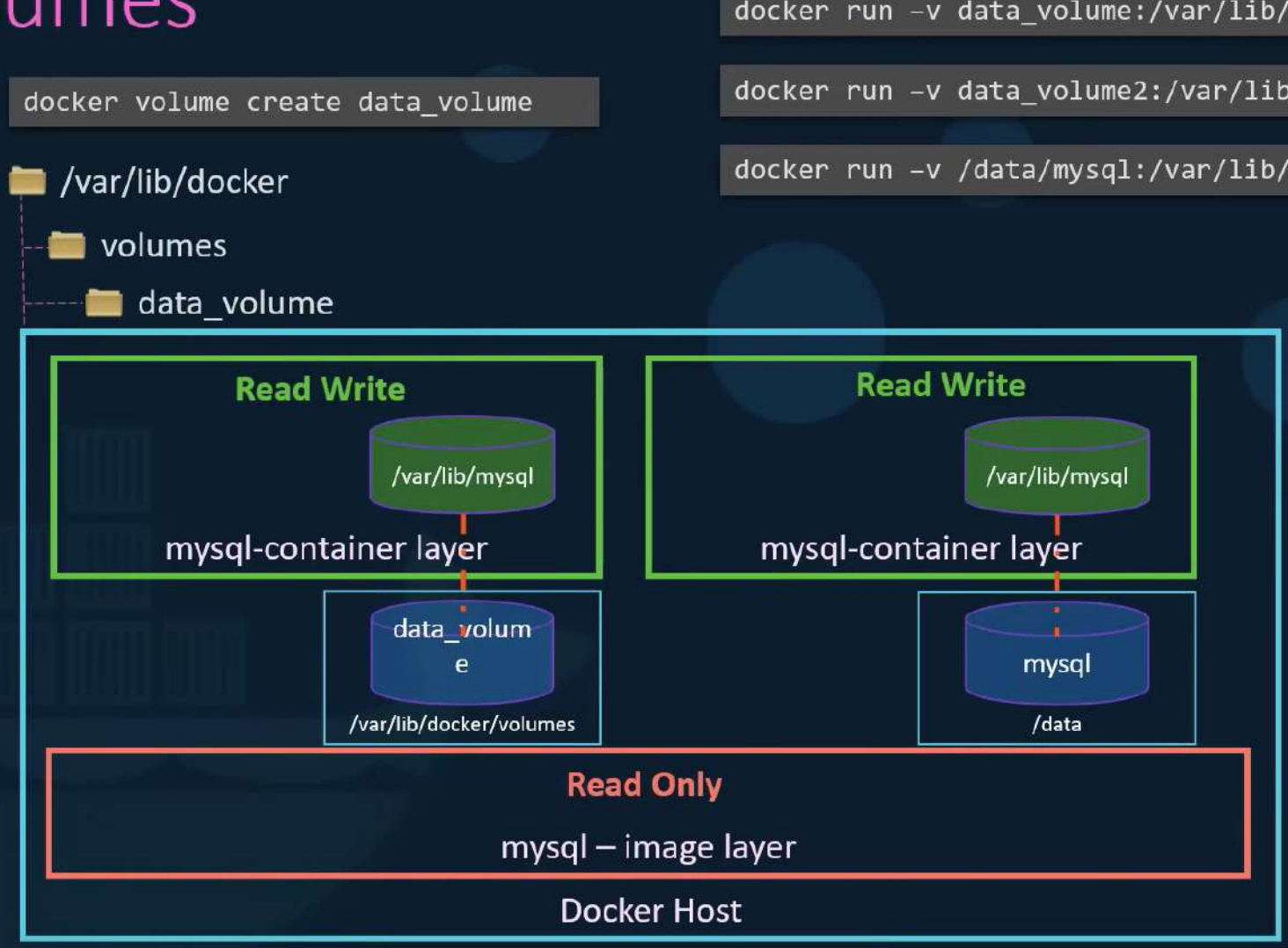
volumes

```
docker volume create data_volume
```

```
docker run -v data_volume:/var/lib/mysql mysql
```

```
docker run -v data_volume2:/var/lib/mysql mysql
```

```
docker run -v /data/mysql:/var/lib/mysql mysql
```



volumes

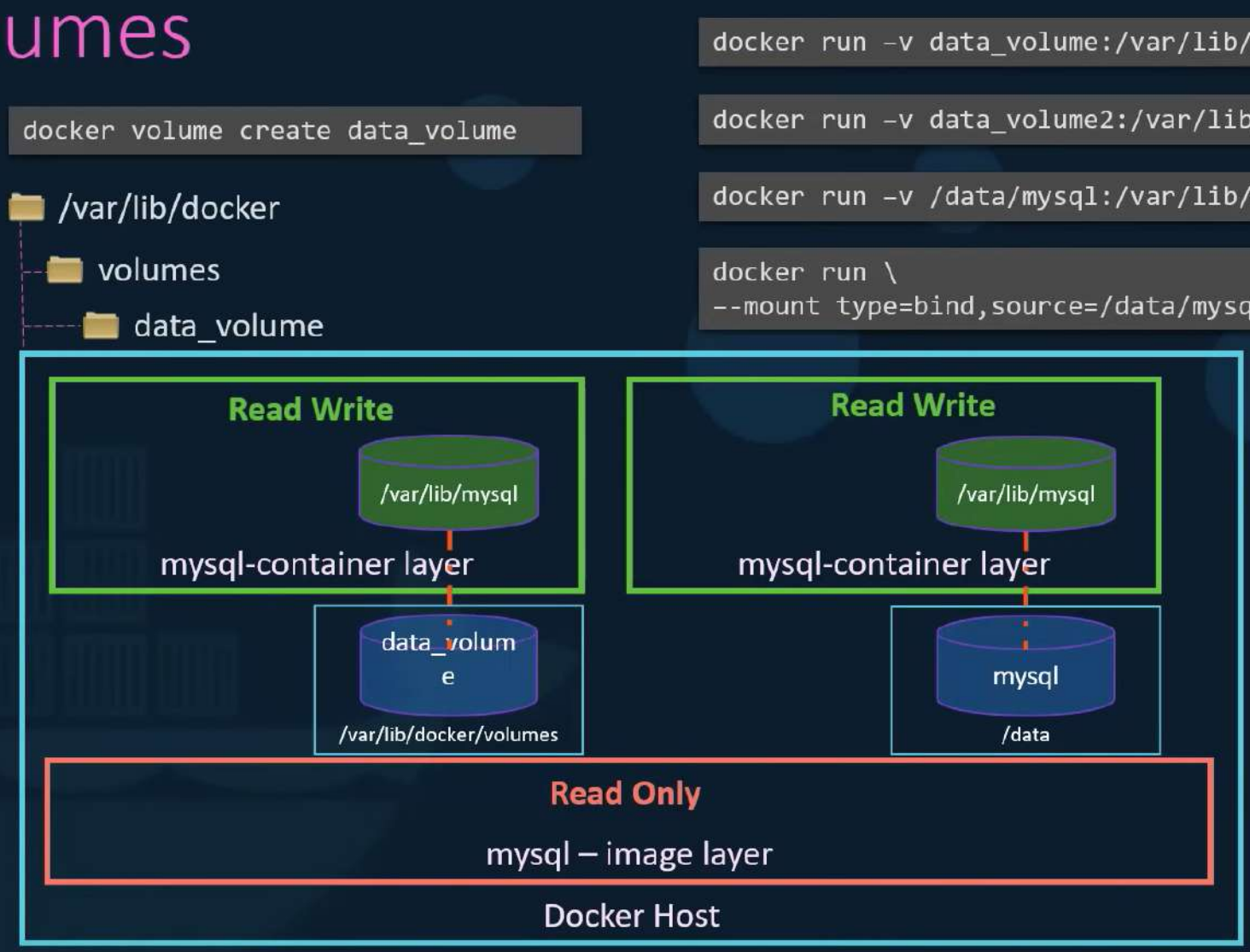
```
docker volume create data_volume
```

```
docker run -v data_volume:/var/lib/mysql mysql
```

```
docker run -v data_volume2:/var/lib/mysql mysql
```

```
docker run -v /data/mysql:/var/lib/mysql mysql
```

```
docker run \
--mount type=bind,source=/data/mysql,target=/var/lib/mysql mysql
```



Storage drivers

- AUFS
- ZFS
- BTRFS
- Device Mapper
- Overlay
- Overlay2

(35) Docker Tutorial for Beginner | Labs: Docker Storage - KodeKlou | Inspecting Container Environmer | New Tab

kodekloud.com/topic/labs-docker-storage-2/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

35% Off: Business Plan Sale (lasts till 30th Sept)

KodeKloud

My Account

Task | Hint

59:06

1 2 3 4 5 6 7 8 9 10

What location are the files related to the docker containers and images stored?

/tmp/docker

/opt/docker

/etc/docker

...

Terminal 1

Welcome to the KodeKloud Hands-On lab

$

Feedback

28°C Mostly cloudy

Search

ENG IN

14:45 20-09-2023

(35) Docker Tutorial for Beginner | Labs: Docker Storage - KodeKlou | Inspecting Container Environmer | New Tab

kodekloud.com/topic/labs-docker-storage-2/

Gmail | YouTube | Classes | 1 new message | Fundamentals of D... | Labs: Docker Image... | (1) Explore - LeetCo... | Data Structures - G... | (7) Day-11 | Git Inte... | Dashboard | Hacker... | All Bookmarks

KodeKloud

My Account

Task | Hint | Solution

58:12

1 2 3 4 5 6 7 8 9 10

What directory under `/var/lib/docker` are the files related to the container `alpine-3` image stored?

075cf53c0518446811e8af3f7abb5397ea161a5a13b6e20f7c19cfd358c8a859

6fa416a7d24c669c661697863b6ef4f375c86761a2c6d12bfa2560a3a48ed95e

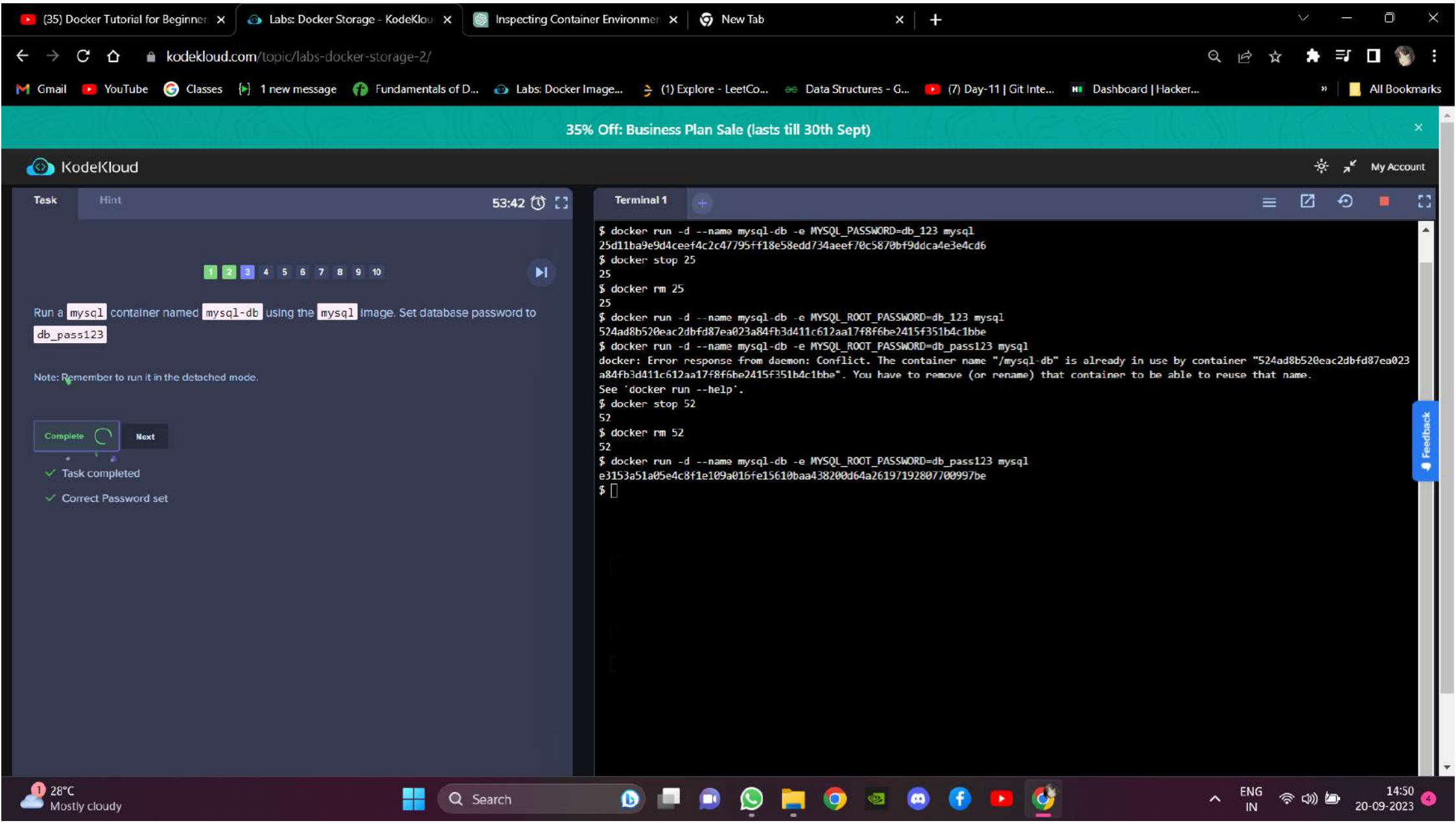
643b9cf496c099293d77cf530ed627c815321d2c22def65ef21cc29071d80a4a

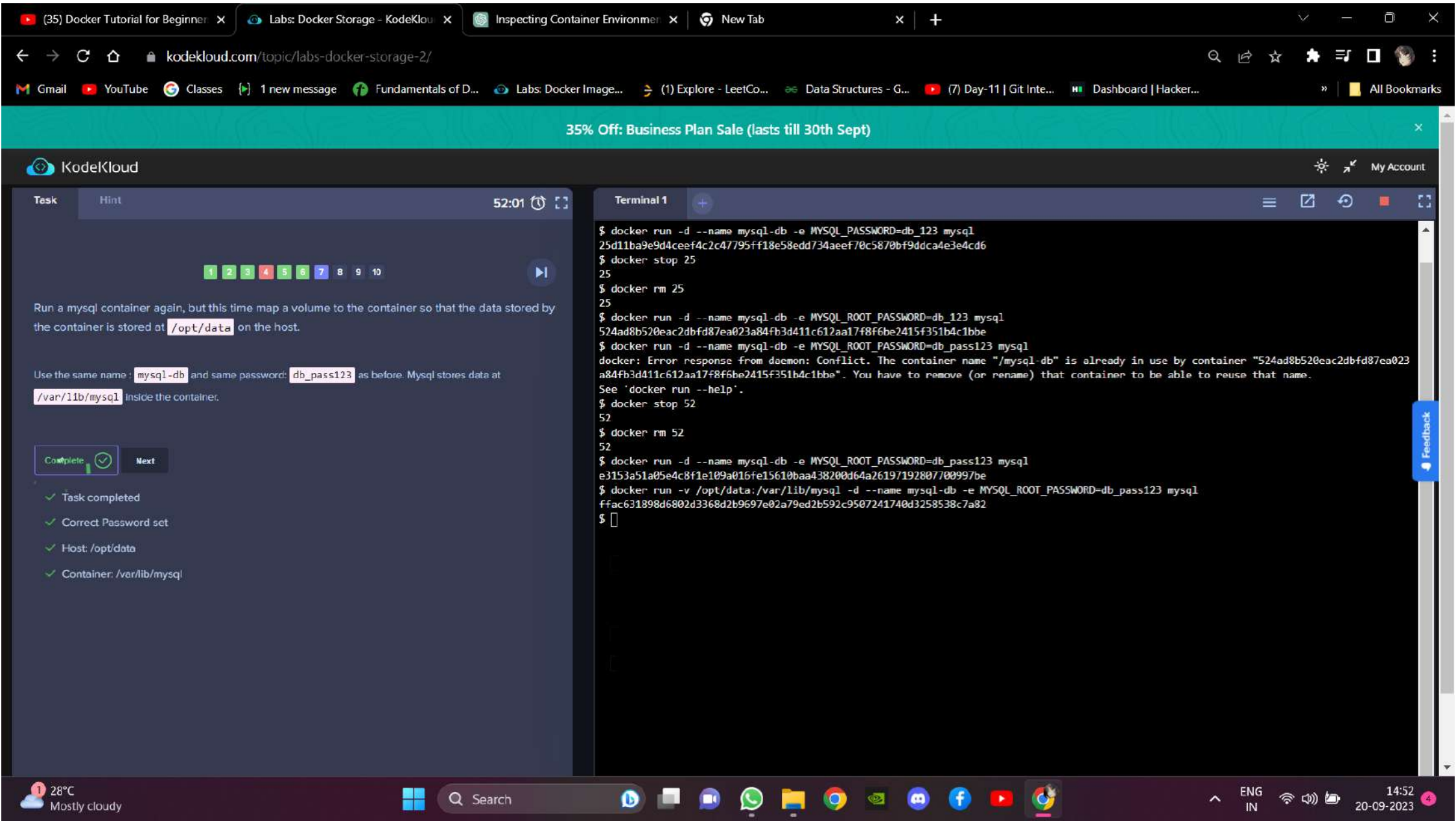
Terminal 1

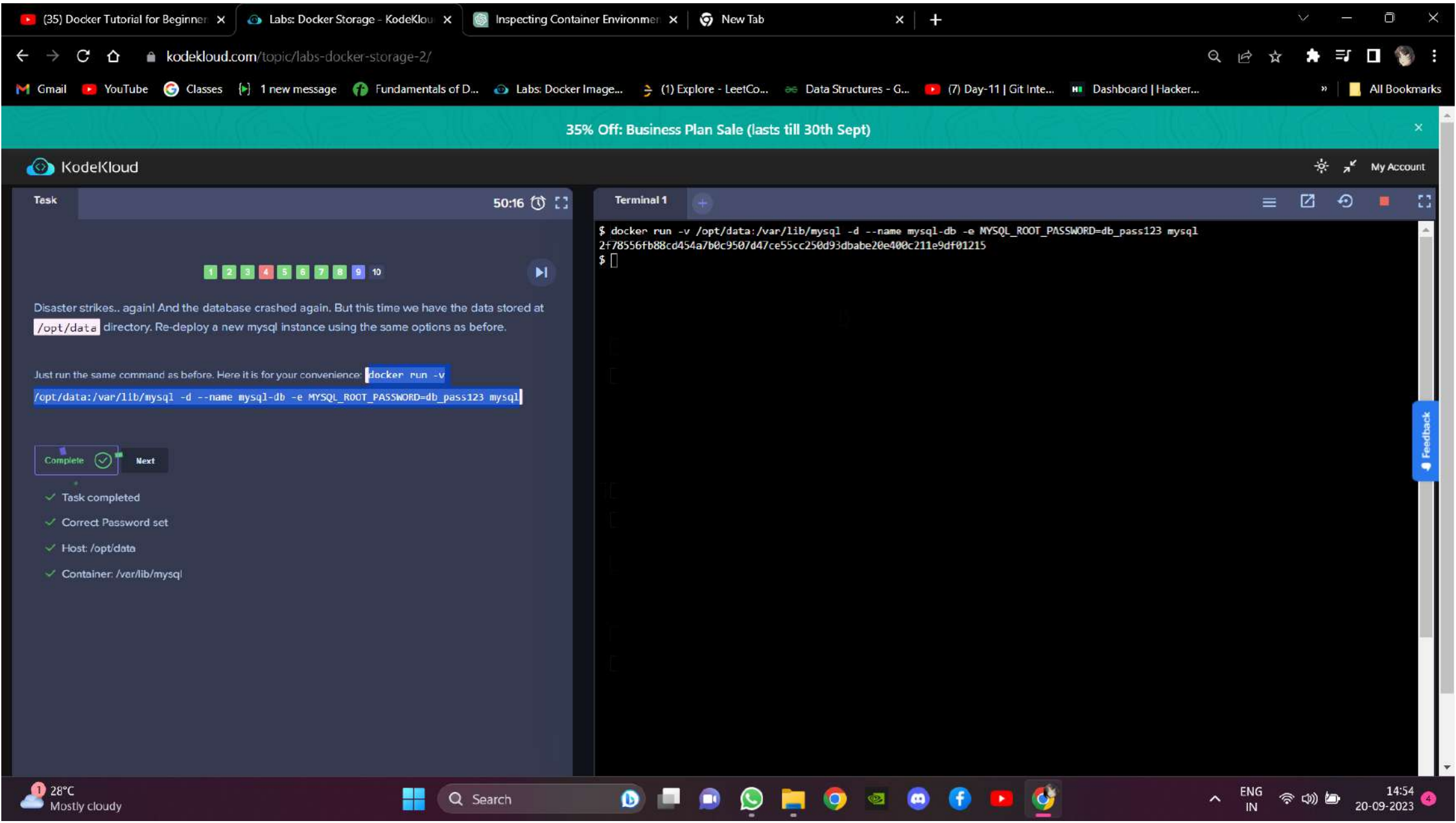
```
        Welcome to the KodeKloud Hands-On lab

                 All rights reserved

$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
$ docker image ls
REPOSITORY                    TAG       IMAGE ID       CREATED        SIZE
redis                         latest    eca1379fe8b5   5 months ago   117MB
mysql                         latest    8189e588b0e8   5 months ago   564MB
nginx                         latest    6efc10a0510f   5 months ago   142MB
postgres                      latest    ceccf204404e   5 months ago   379MB
nginx                         alpine    8e75cbc5b25c   5 months ago   41MB
alpine                        latest    9ed4aefc74f6   5 months ago   7.04MB
ubuntu                        latest    08d22c0ceb15   6 months ago   77.8MB
kodekloud/simple-webapp-mysql latest    129dd9f67367   4 years ago    96.6MB
kodekloud/simple-webapp       latest    c6e3cd9aae36   4 years ago    84.8MB
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND     CREATED          STATUS                      PORTS     NAMES
075cf53c0518   alpine    "/bin/sh"   56 seconds ago   Exited (0) 55 seconds ago             alpine-3
a81101cbcd2c   alpine    "/bin/sh"   57 seconds ago   Exited (0) 56 seconds ago             alpine-2
afe0a8dca76d   alpine    "/bin/sh"   58 seconds ago   Exited (0) 57 seconds ago             alpine-1
$
```

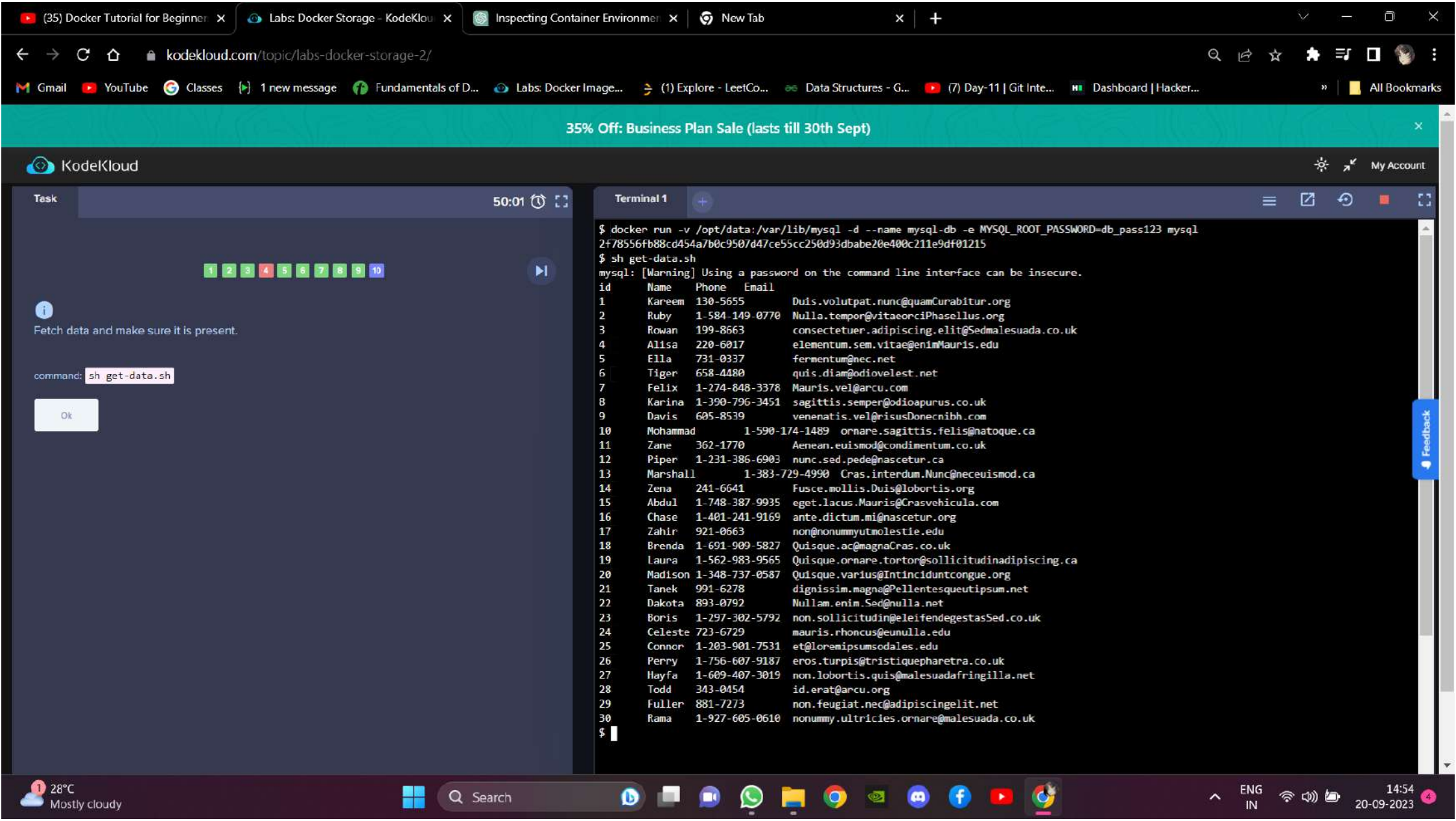
Feedback

28°C Mostly cloudy | Search | ENG IN | 14:46 20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Storage - KodeKlou
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-storage-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
53:42
1 2 3 4 5 6 7 8 9 10
Run a mysql container named mysql-db using the mysql image. Set database password to db_pass123
Note: Remember to run it in the detached mode.
Complete
Next
Task completed
Correct Password set
Terminal 1
$ docker run -d --name mysql-db -e MYSQL_PASSWORD=db_123 mysql
25d11ba9e9d4ceef4c2c47795ff18e58edd734aeef70c5870bf9ddca4e3e4cd6
$ docker stop 25
25
$ docker rm 25
25
$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_123 mysql
524ad8b520eac2dbfd87ea023a84fb3d411c612aa17f8f6be2415f351b4c1bbe
$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
docker: Error response from daemon: Conflict. The container name "/mysql-db" is already in use by container "524ad8b520eac2dbfd87ea023a84fb3d411c612aa17f8f6be2415f351b4c1bbe". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
$ docker stop 52
52
$ docker rm 52
52
$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
e3153a51a05e4c8f1e109a016fe15610baa438200d64a26197192807700997be
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:50
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Storage - KodeKlou
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-storage-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
Hint
52:01
1 2 3 4 5 6 7 8 9 10
Run a mysql container again, but this time map a volume to the container so that the data stored by the container is stored at /opt/data on the host.
Use the same name : mysql-db and same password: db_pass123 as before. Mysql stores data at /var/lib/mysql inside the container.
Complete
Next
Task completed
Correct Password set
Host: /opt/data
Container: /var/lib/mysql
Terminal 1
$ docker run -d --name mysql-db -e MYSQL_PASSWORD=db_123 mysql
25d11ba9e9d4ceef4c2c47795ff18e58edd734aeef70c5870bf9ddca4e3e4cd6
$ docker stop 25
25
$ docker rm 25
25
$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_123 mysql
524ad8b520eac2dbfd87ea023a84fb3d411c612aa17f8f6be2415f351b4c1bbe
$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
docker: Error response from daemon: Conflict. The container name "/mysql-db" is already in use by container "524ad8b520eac2dbfd87ea023a84fb3d411c612aa17f8f6be2415f351b4c1bbe". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
$ docker stop 52
52
$ docker rm 52
52
$ docker run -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
e3153a51a05e4c8f1e109a016fe15610baa438200d64a2619719280770099 7be
$ docker run -v /opt/data:/var/lib/mysql -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
ffac631898d6802d3368d2b9697e02a79ed2b592c9507241740d3258538c7a82
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:52
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Storage - KodeKlou
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-storage-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
50:16
1 2 3 4 5 6 7 8 9 10
Disaster strikes.. again! And the database crashed again. But this time we have the data stored at /opt/data directory. Re-deploy a new mysql instance using the same options as before.
Just run the same command as before. Here it is for your convenience: docker run -v /opt/data:/var/lib/mysql -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
Complete
Next
Task completed
Correct Password set
Host: /opt/data
Container: /var/lib/mysql
Terminal 1
$ docker run -v /opt/data:/var/lib/mysql -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
2f78556fb88cd454a7b0c9507d47ce55cc250d93dbabe20e400c211e9df01215
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:54
20-09-2023


(35) Docker Tutorial for Beginner
Labs: Docker Storage - KodeKlou
Inspecting Container Environmer
New Tab
kodekloud.com/topic/labs-docker-storage-2/
Gmail
YouTube
Classes
1 new message
Fundamentals of D...
Labs: Docker Image...
(1) Explore - LeetCo...
Data Structures - G...
(7) Day-11 | Git Inte...
Dashboard | Hacker...
All Bookmarks
35% Off: Business Plan Sale (lasts till 30th Sept)
KodeKloud
My Account
Task
50:01
1 2 3 4 5 6 7 8 9 10
Fetch data and make sure it is present.
command: sh get-data.sh
Ok
Terminal 1
$ docker run -v /opt/data:/var/lib/mysql -d --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql
2f78556fb88cd454a7b0c9507d47ce55cc250d93dbabe20e400c211e9df01215
$ sh get-data.sh
mysql: [Warning] Using a password on the command line interface can be insecure.
id Name Phone Email
1 Kareem 130-5655 Duis.volutpat.nunc@quamCurabitur.org
2 Ruby 1-584-149-0770 Nulla.tempor@vitaeorciPhasellus.org
3 Rowan 199-8663 consectetuer.adipiscing.elit@Sedmalesuada.co.uk
4 Alisa 220-6017 elementum.sem.vitae@enimMauris.edu
5 Ella 731-0337 fermentum@nec.net
6 Tiger 658-4480 quis.diam@odiovelest.net
7 Felix 1-274-848-3378 Mauris.vel@arcu.com
8 Karina 1-390-796-3451 sagittis.semper@odioapurus.co.uk
9 Davis 605-8539 venenatis.vel@risusDonecnibh.com
10 Mohammad 1-590-174-1489 ornare.sagittis.felis@natoque.ca
11 Zane 362-1770 Aenean.euismod@condimentum.co.uk
12 Piper 1-231-386-6903 nunc.sed.pede@nascetur.ca
13 Marshall 1-383-729-4990 Cras.interdum.Nunc@neceuismod.ca
14 Zena 241-6641 Fusce.mollis.Duis@lobortis.org
15 Abdul 1-748-387-9935 eget.lacus.Mauris@Crasvehicula.com
16 Chase 1-401-241-9169 ante.dictum.mi@nascetur.org
17 Zahir 921-0663 non@nonummyutmolestie.edu
18 Brenda 1-691-909-5827 Quisque.ac@magnaCras.co.uk
19 Laura 1-562-983-9565 Quisque.ornare.tortor@sollicitudinadipiscing.ca
20 Madison 1-348-737-0587 Quisque.varius@Intinciduntcongue.org
21 Tanek 991-6278 dignissim.magna@Pellentesqueutipsum.net
22 Dakota 893-0792 Nullam.enim.Sed@nulla.net
23 Boris 1-297-302-5792 non.sollicitudin@eleifendegestasSed.co.uk
24 Celeste 723-6729 mauris.rhoncus@eunulla.edu
25 Connor 1-203-901-7531 et@loremipsumsodales.edu
26 Perry 1-756-607-9187 eros.turpis@tristiquepharetra.co.uk
27 Hayfa 1-609-407-3019 non.lobortis.quis@malesuadafringilla.net
28 Todd 343-0454 id.erat@arcu.org
29 Fuller 881-7273 non.feugiat.nec@adipiscingelit.net
30 Rama 1-927-605-0610 nonummy.ultricies.ornare@malesuada.co.uk
$
Feedback
28°C
Mostly cloudy
Search
ENG
IN
14:54
20-09-2023

Docker compose

```
docker run mmumshad/simple-webapp
```

```
docker run mongodb
```

```
docker run redis:alpine
```

```
docker run ansible
```

docker-compose.yml

```
services:
     web:
        image: “mmumshad/simple-webapp"
     database:
        image: “mongodb“
     messaging:
        image: "redis:alpine“
     orchestration:
        image:  “ansible“
```

```
docker-compose up
```

docker run --links

```
docker run -d --name=redis redis
```

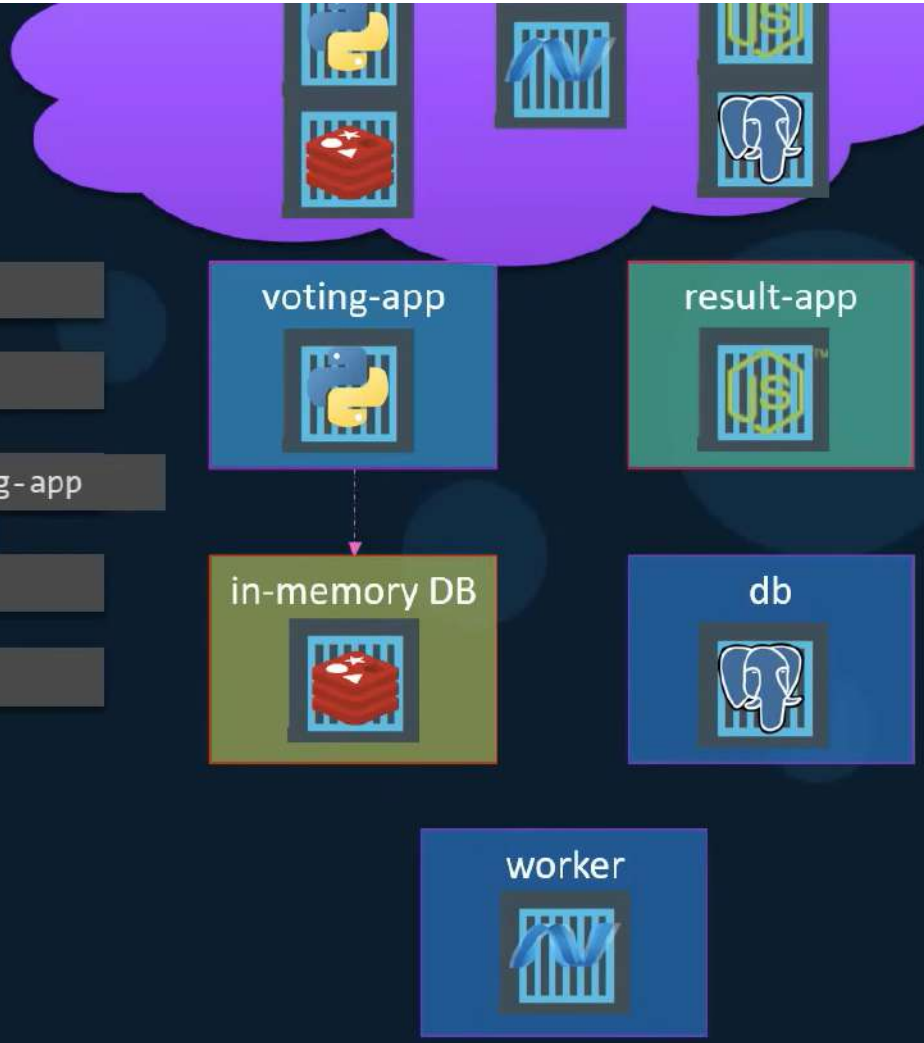
```
docker run -d --name=db postgres:9.4  result-app
```

```
docker run -d --name=vote -p 5000:80 --link redis:redis voting-app
```

```
docker run -d --name=result -p 5001:80
```

```
docker run -d --name=worker  worker
```

```
def get_redis():
    if not hasattr(g, 'redis'):
        g.redis = Redis(host="redis", db=0, socket_timeout=5)
    return g.redis
```



Docker compose

```
docker run -d --name=redis redis
```

```
docker run -d --name=db postgres:9.4
```

```
docker run -d --name=vote -p 5000:80 --link redis:redis voting-app
```

```
docker run -d --name=result -p 5001:80 --link db:db result-app
```

```
docker run -d --name=worker --link db:db --link redis:redis worker
```

docker-compose.yml

Docker compose

```
docker run -d --name=redis redis
```

```
docker run -d --name=db postgres:9.4
```

```
docker run -d --name=vote -p 5000:80 --link redis:redis voting-app
```

```
docker run -d --name=result -p 5001:80 --link db:db result-app
```

```
docker run -d --name=worker --link db:db --link redis:redis worker
```

docker-compose.yml

```
redis:
  image: redis
db:
  image: postgres:9.4
vote:
  image: voting-app
  ports:
    - 5000:80
  links:
    - redis
result:
  image: result-app
  ports:
    - 5001:80
  links:
    - db
worker:
    image: worker
    links:
      - redis
      - db
```

Docker compose - build

docker-compose.yml

```
redis:
    image: redis
db:
    image: postgres:9.4
vote:
    image: voting-app
    ports:
        - 5000:80
    links:
        - redis
result:
    image: result
    ports:
        - 5001:80
    links:
        - db
worker:
    image: worker
    links:
        - db
        - redis
```

docker-compose.yml

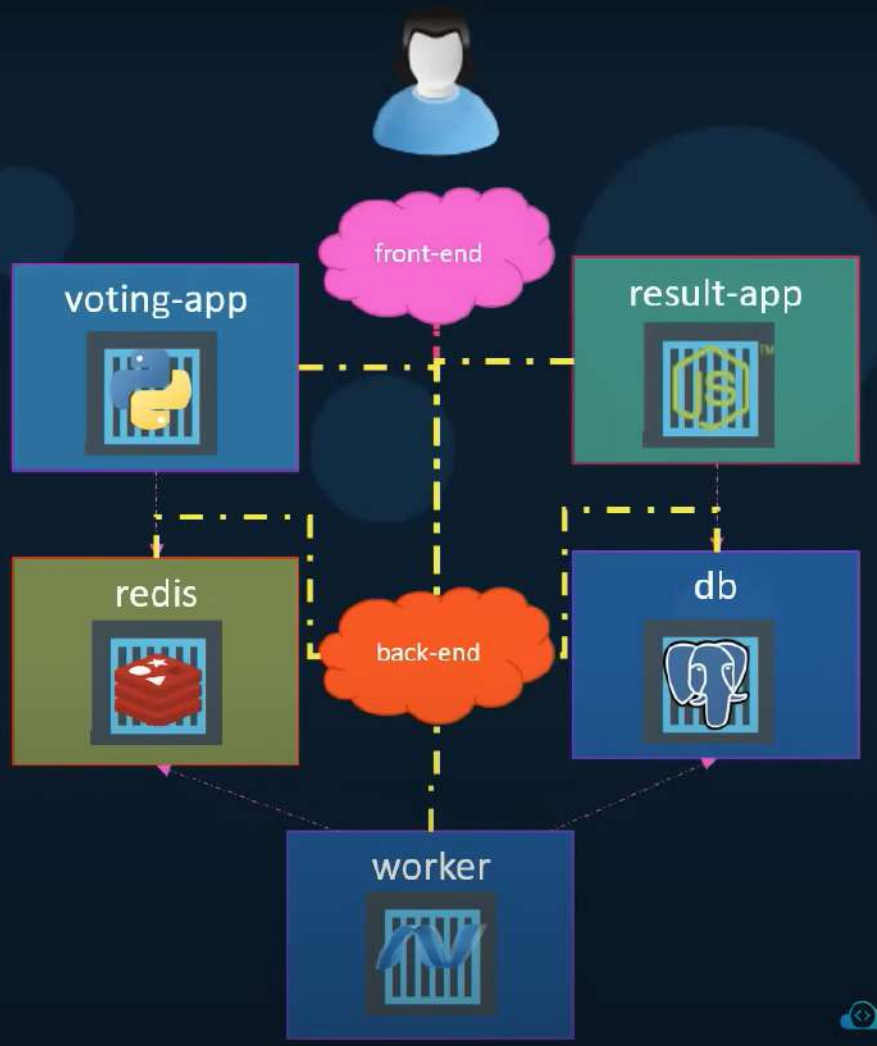
```
redis:
    image: redis
db:
    image: postgres:9.4
vote:
    build: ./vote
    ports:
        - 5000:80
    links:
        - redis
result:
    build: ./result
    ports:
        - 5001:80
    links:
        - db
worker:
    build: ./worker
    links:
        - db
        - redis
```

Docker Tutorial for Beginners | Docker Full Course | Access to FREE LABS [No Ads]

Docker compose

docker-compose.yml

```
version: 2
services:
     redis:
          image: redis

          networks:
               - back-end
     db:
          image: postgres:9.4
          networks:
               - back-end
     vote:
          image: voting-app
          networks:
               - front-end
               - back-end
     result:
          image: result
          networks:
               - front-end
               - back-end

networks:
    front-end:
    back-end:
```



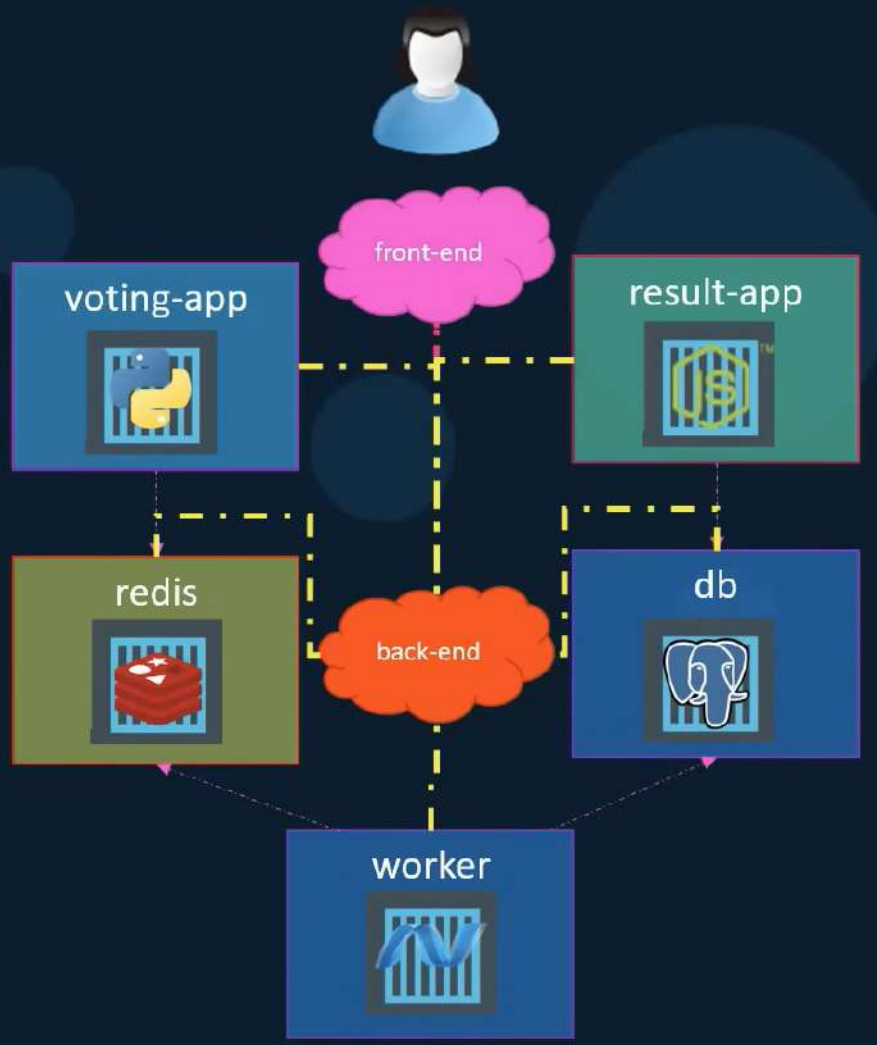
1:34:16 / 2:09:27 • Docker Compose ›

Scroll for details

Docker compose

docker-compose.yml

```
version: 2
services:
    redis:
        image: redis
        networks:
            - back-end
    db:
        image: postgres:9.4
        networks:
            - back-end
    vote:
        image: voting-app
        networks:
            - front-end
            - back-end
    result:
        image: result
        networks:
            - front-end
            - back-end

networks:
    front-end:
    back-end:
```



Image

```
image: nginx/nginx
```

User/ Account

Image/ Repository

Image

docker.io
Docker Hub

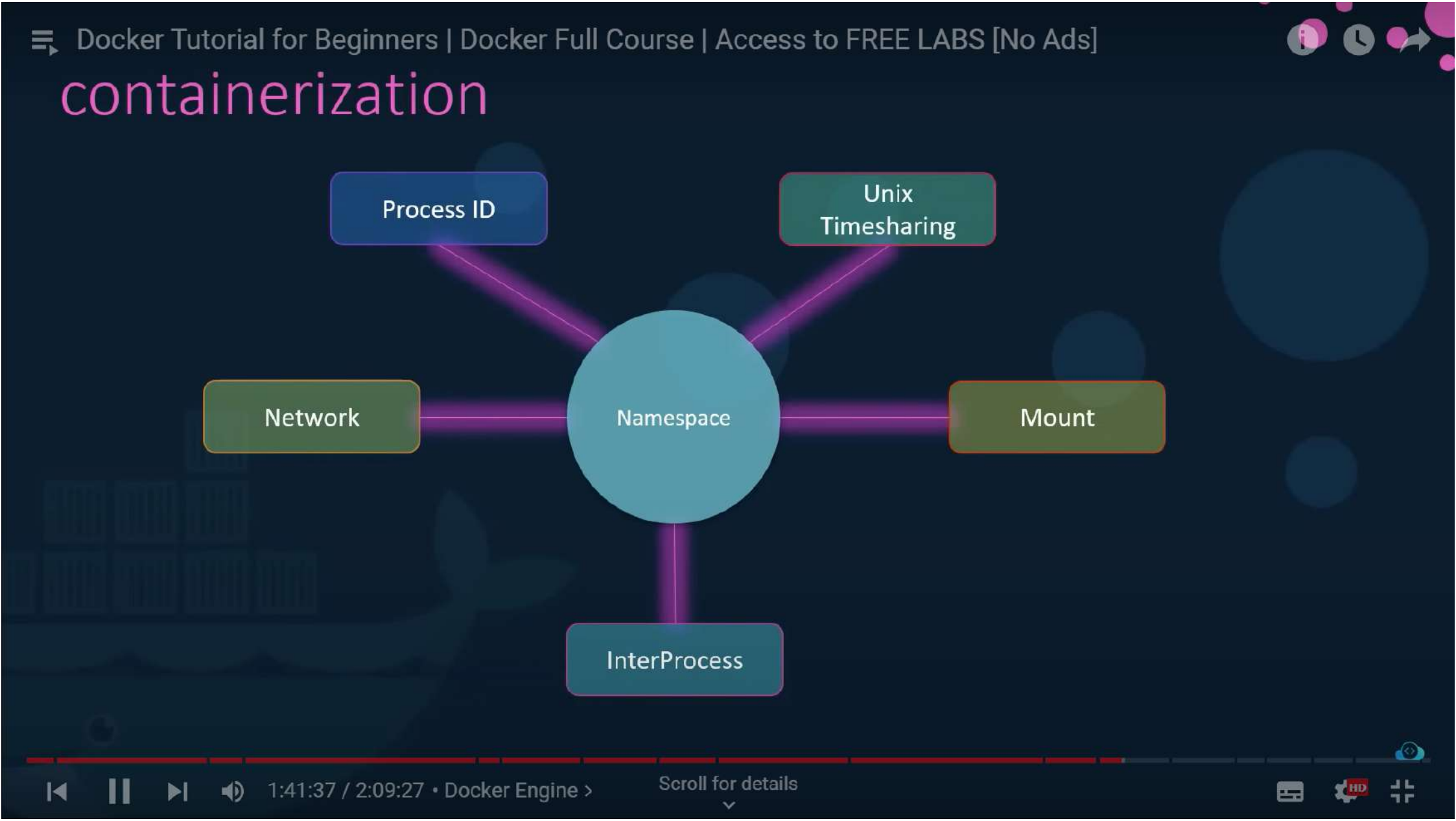
```
image: docker.io/nginx/nginx
```

Registry

User/
Account

Image/
Repository

Docker Engine


Docker Tutorial for Beginners | Docker Full Course | Access to FREE LABS [No Ads]
containerization
Process ID
Unix Timesharing
Network
Namespace
Mount
InterProcess
1:41:37 / 2:09:27 • Docker Engine >
Scroll for details