Topic: Docker Assignment – Advanced

Submitted by: Sumit Prasad (sumit.prasad@nagarro.com)

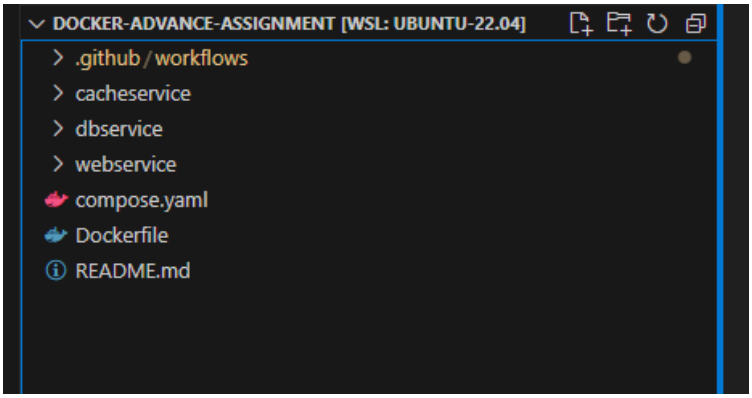
Date: 10 – Dec – 2024

Instructor: sarvesh.gupta@nagarro.com

GitHub Link: https://github.com/sumitkp11/docker-advance-assignment

- **Provide a brief overview of the multi-service application that is built and deployed using a CI/CD pipeline:**
  - The assignment comprise of 3 services:
    - Web service: A small ToDo web application based on Node.js and SQL.
    - DB service: An SQL database
    - Cache service: A Redis cache
  - The CI/CD tool used in this assignment is GitHub Actions, where two separate workflows have been created for CI and CD pipelines.
    - Continuous Integration (CI) - `.github/workflow/docker-image-ci.yml`
    - Continuous Deployment (CD) - `.github/workflow/docker-local-deploy.yml`

- **Set up a new multi-service application project:**
  - The image below shows the structure of the multi-service application project:



- Create Docker containers for at least three services: a web service, a database service and a cache service.

Create Dockerfiles for each service to build custom Docker images:

- For web service:

```
Dockerfile
webservice > Dockerfile
FROM node:lts-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

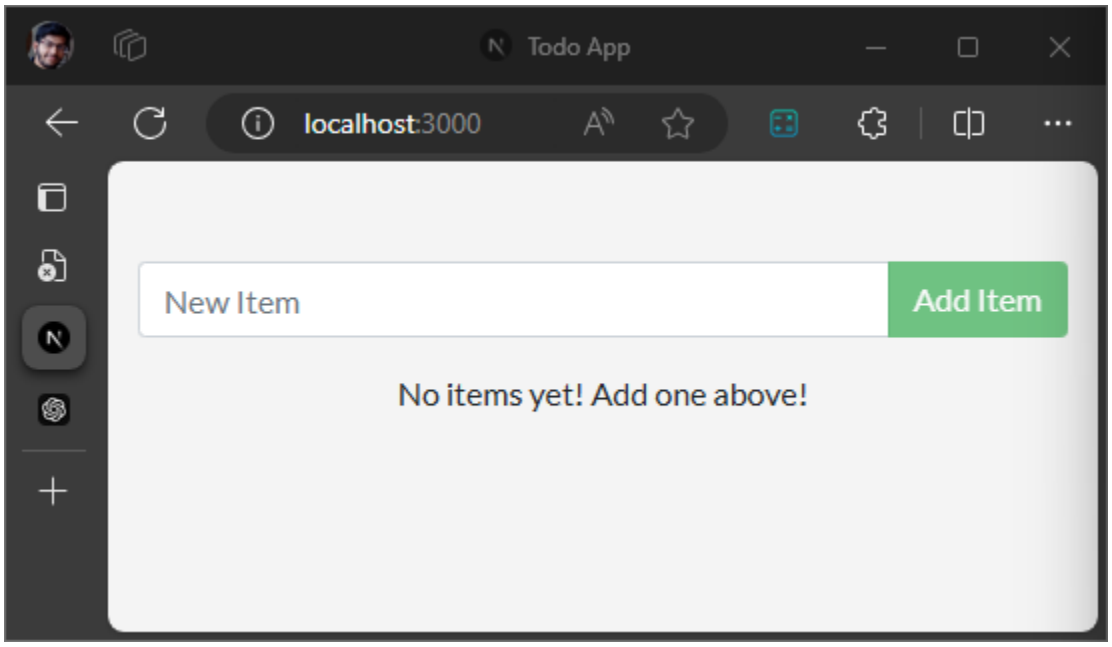
To build and run the container:

- Build the Docker image: “docker build -t webserviceimage .”

```
Windows PowerShell   root@IN-3P96C54: ~/docker_   root@IN-3P96C54: ~
-rw-r--r-- 1 root root  126 Dec 10 12:35 Dockerfile
-rw-r--r-- 1 root root  477 Dec 10 14:05 README.md
drwxr-xr-x 2 root root 4096 Dec  5 14:08 cacheservice
-rw-r--r-- 1 root root 1067 Dec 10 13:48 compose.yaml
drwxr-xr-x 2 root root 4096 Dec  5 14:08 dbservice
drwxr-xr-x 4 root root 4096 Dec  5 14:09 webservice
root@IN-3P96C54:~/docker_training/docker-advance-assignment# cd webservice
root@IN-3P96C54:~/docker_training/docker-advance-assignment/webservice# docker build -t webserviceimage .
[+] Building 36.5s (9/9) FINISHED                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 152B                                                                          0.0s
 => [internal] load metadata for docker.io/library/node:lts-alpine                                            0.0s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [internal] load build context                                                                             0.1s
 => => transferring context: 4.60MB                                                                           0.1s
 => [1/4] FROM docker.io/library/node:lts-alpine                                                              0.0s
 => CACHED [2/4] WORKDIR /app                                                                                 0.0s
 => [3/4] COPY . .                                                                                            0.1s
 => [4/4] RUN yarn install --production                                                                      35.4s
 => exporting to image                                                                                        0.7s
 => => exporting layers                                                                                       0.7s
 => => writing image sha256:336ed598bf201e7a0d39e81b4326d7e77fdcdba0cdcb077e27a1813ba5cf25e6                  0.0s
 => => naming to docker.io/library/webserviceimage                                                            0.0s
root@IN-3P96C54:~/docker_training/docker-advance-assignment/webservice#
```

- Create container from the custom image: “docker run -p 3000:3000 webserviceimage”

```
root@IN-3P96C54:~/docker_training/docker-advance-assignment/
webservice# docker run -p 3000:3000 webserviceimage
Using sqlite database at /etc/todos/todo.db
Listening on port 3000
```



- For DB service:

```
dbservice > Dockerfile
# using mysql image
FROM mysql:8.0

# setting environment varialbles for mysql
# ENV MYSQL_ROOT_PASSWORD=rootpassword
# ENV MYSQL_DATABASE=mydatabase
# ENV MYSQL_USER=myuser
# ENV MYSQL_PASSWORD=mypassword
# RUN cat /run/secrets/my_secret > /etc/my_secret_file
RUN --mount=type=secret,id=my_secret \
cat /run/secrets/my_secret > /etc/my_secret_file

RUN echo "Secrets copied successfully"

# exposing port for db
EXPOSE 3306
```

- Build the image: “docker buildx build --secret id=my_secret,src=secrets.txt -t dbserviceadvanced . “

```
root@IN-3P96C54:dbservice$docker buildx build --secret id=my_secret,src
=secrets.txt -t dbserviceadvanced .
[+] Building 0.1s (7/7) FINISHED                          docker:default
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 464B                                0.0s
 => [internal] load metadata for docker.io/library/mysql:8.0        0.0s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => [stage-0 1/3] FROM docker.io/library/mysql:8.0                  0.0s
 => CACHED [stage-0 2/3] RUN --mount=type=secret,id=my_secret ca    0.0s
 => CACHED [stage-0 3/3] RUN echo "Secrets copied successfully"     0.0s
 => exporting to image                                              0.0s
 => => exporting layers                                             0.0s
 => => writing image sha256:429207bc503c48548cd7ac12ee7366ab6aa8    0.0s
 => => naming to docker.io/library/dbserviceadvanced                0.0s
root@IN-3P96C54:dbservice$
```

- Run the ‘dbserviceadvanced’ container: “docker run -d -e MYSQL_ROOT_PASSWORD=rootpassword dbserviceadvanced”

- -d : short for –detach | runs the container in the background. This means that Docker starts your container and returns you to the terminal prompt.
- -e : pass environment variables to the container image
- To confirm you have the database up and running, connect to the database and verify that it connects.

```
root@IN-3P96C54: dbservice
root@IN-3P96C54:dbservice$docker exec -it f485f1ef7f7d mysql -u root -
p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.40 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input st
atement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.01 sec)

mysql>
```

For Cache service:

```
Dockerfile
cacheservice > Dockerfile
FROM redis:alpine
EXPOSE 6379
CMD ["redis-server"]
```

- Build the Dockerfile:

```
root@IN-3P96C54: cacheservi
root@IN-3P96C54:cacheservice$docker build -t cacheserviceadvanced .
[+] Building 2.6s (6/6) FINISHED                          docker:default
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 88B                                 0.0s
 => [internal] load metadata for docker.io/library/redis:alpine     2.5s
 => [auth] library/redis:pull token for registry-1.docker.io        0.0s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => CACHED [1/1] FROM docker.io/library/redis:alpine@sha256:c1e     0.0s
 => exporting to image                                              0.0s
 => => exporting layers                                             0.0s
 => => writing image sha256:9ae919608e960a276e08dbee195ce0d1f31     0.0s
 => => naming to docker.io/library/cacheserviceadvanced             0.0s
root@IN-3P96C54:cacheservice$
```

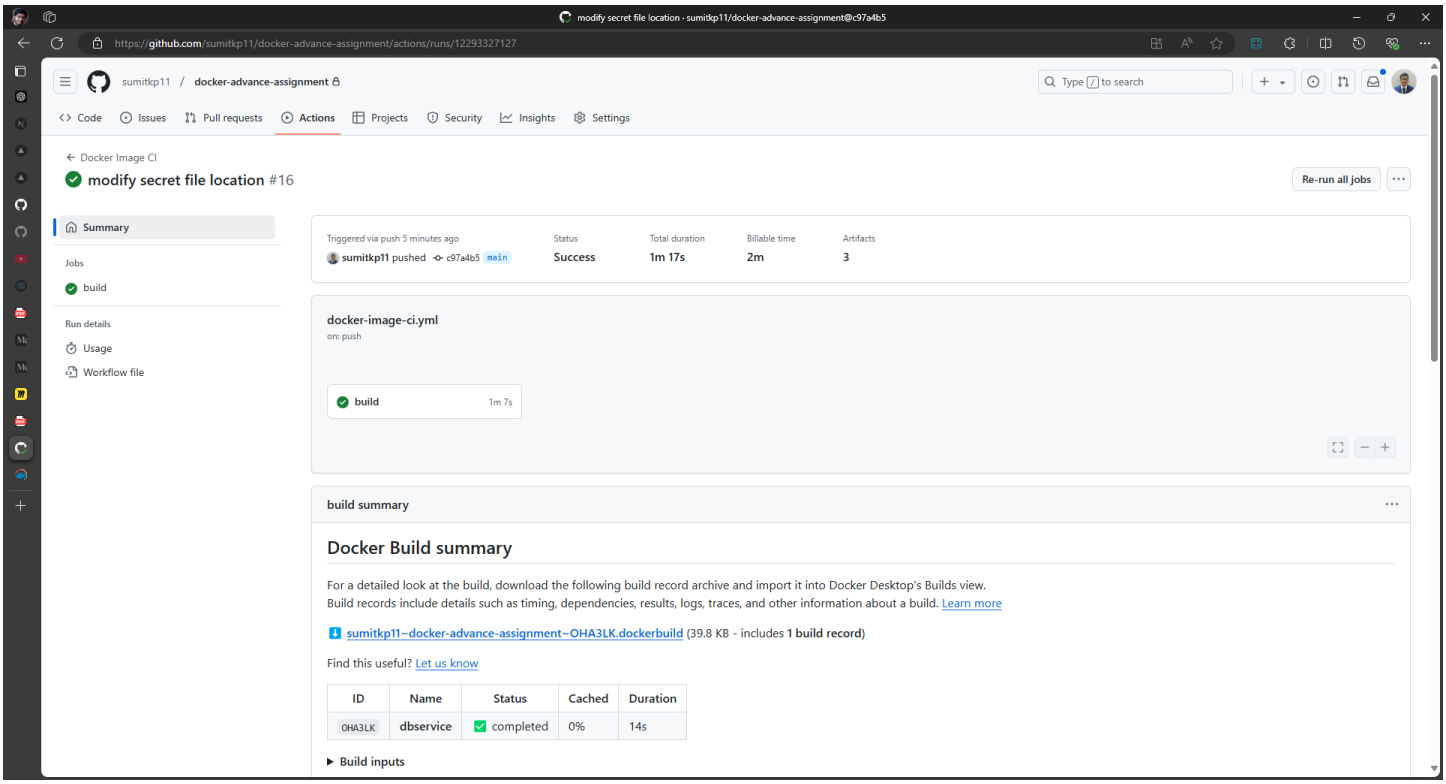
- Run the container image: $docker run -d cacheserviceadvanced

Implement a CI/CD pipeline to automate the build, test and deployment process using Docker and Docker Compose.

Steps:

Step 1 – Implement Continuous Integration workflow:

- The workflow for the assignment can be found at ".github/workflows/docker-image-ci.yml".
- The steps for the workflow are:
    - Checkout the repository
    - Set up Docker Buildx
    - Login into DockerHub
    - Generate labels and tags for web, DB and cache services
    - Build and Push Docker Image for web, DB and cache services
    - End
- Screenshot of successful workflow:



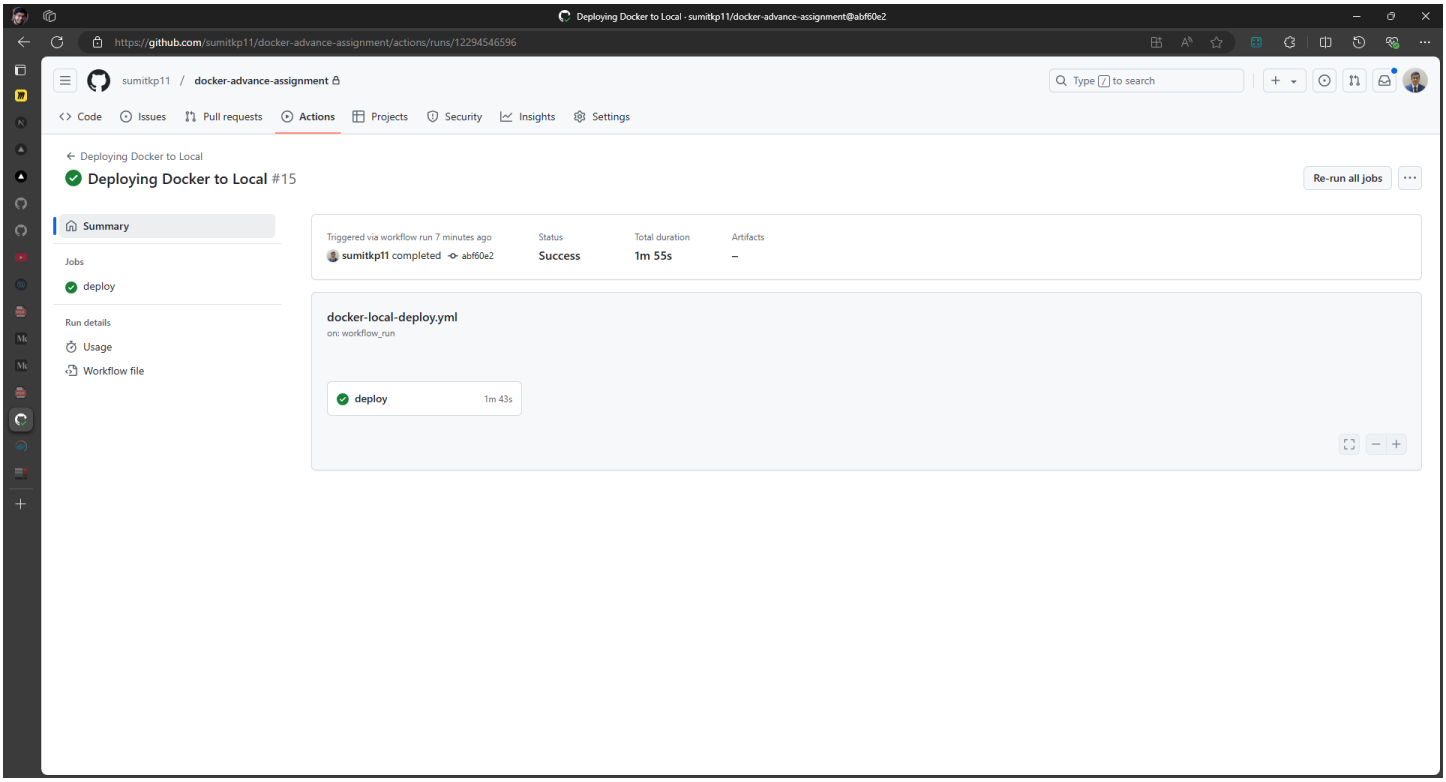
Step 2 – Implement Continuous Deployment workflow:

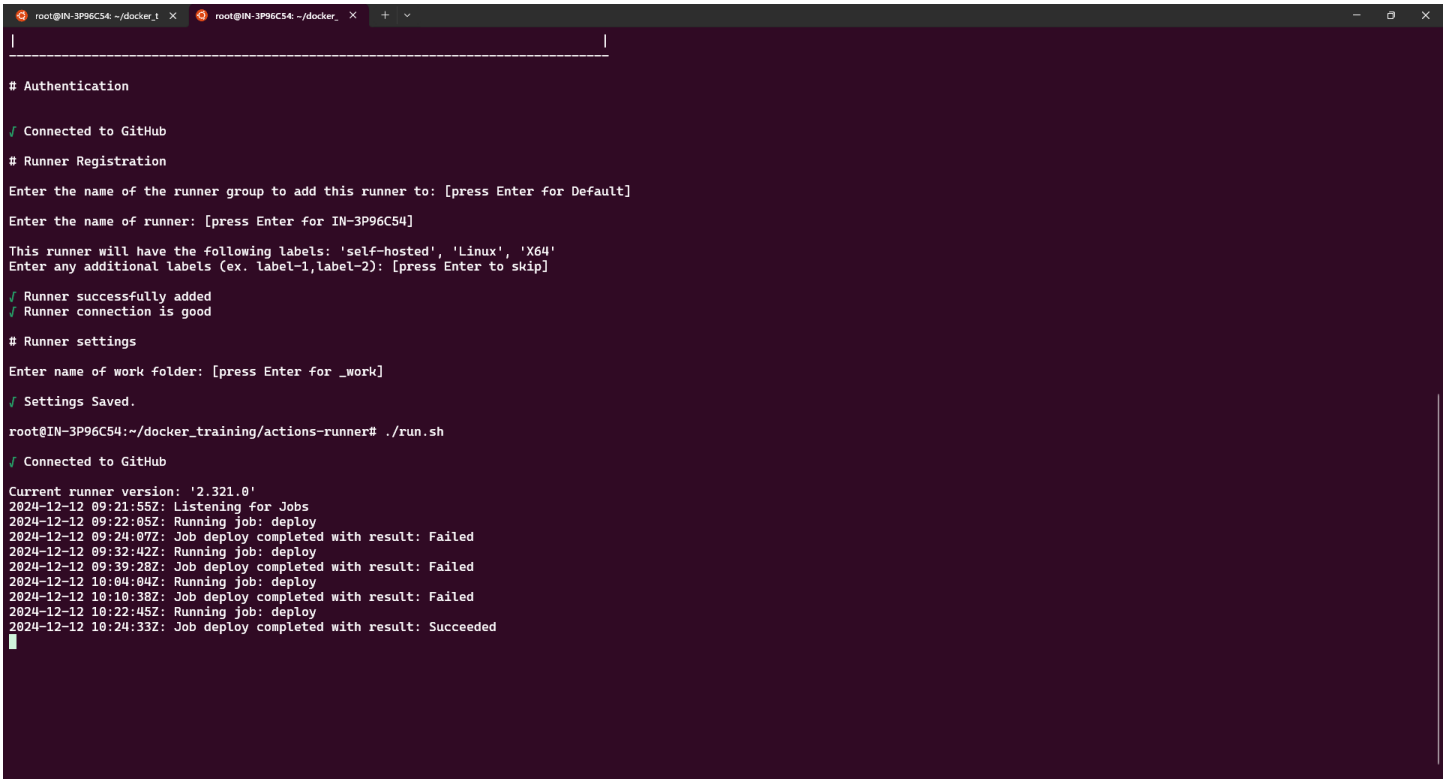
- For this, I am using Github's self hosted runner. In order to set it up, goto Settings > Actions > Runners > New self-hosted runner. On next page, follow the setup instructions according to the Operating System. I am using Linux for this purpose.

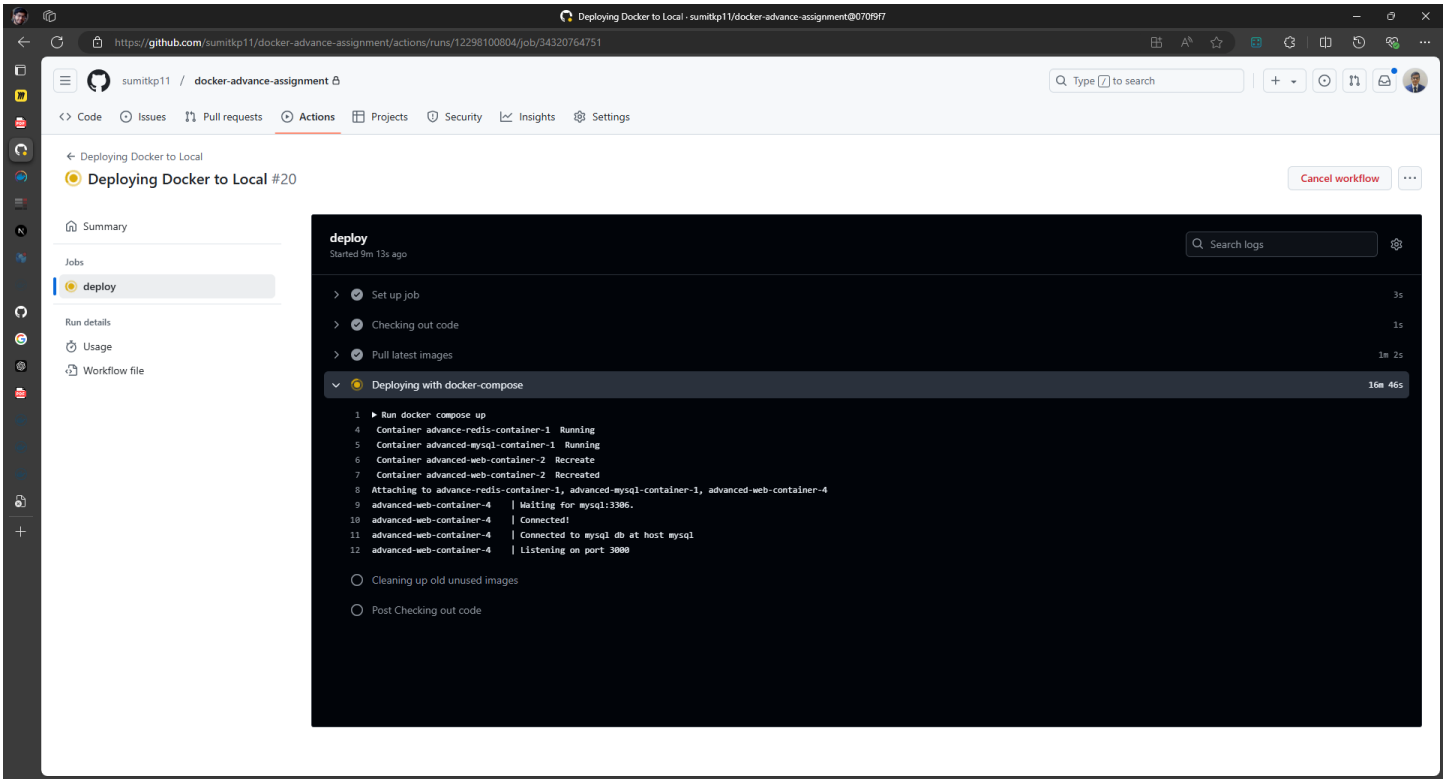
```
root@IN-3P96C54:~/docker_training/actions-runner# ./config.sh --url https://github.com/sumitkp11/docker-advance-ass
ignment --token ATDXD6FX3PW7RF22ROJD44THLK5QE

--------------------------------------------------------------------------------
|        ____ _ _   _   _       _          _     _   _                        |
|       / ___(_) |_| | | |_   _| |__      / \   ___| |_(_) ___  _ __   ___     |
|      | |  _| | __| |_| | | | | '_ \    / _ \ / __| __| |/ _ \| '_ \/ __|    |
|      | |_| | | |_|  _  | |_| | |_) |  / ___ \ (__| |_| | (_) | | | \__ \    |
|       \____|_|\__|_| |_|\__,_|_.__/  /_/   \_\___|\__|_|\___/|_| |_|___/    |
|                                                                              |
|                       Self-hosted runner registration                        |
|                                                                              |
--------------------------------------------------------------------------------

# Authentication


√ Connected to GitHub

# Runner Registration

Enter the name of the runner group to add this runner to: [press Enter for Default]

Enter the name of runner: [press Enter for IN-3P96C54]

This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]

√ Runner successfully added
√ Runner connection is good

# Runner settings

Enter name of work folder: [press Enter for _work]

√ Settings Saved.

root@IN-3P96C54:~/docker_training/actions-runner#
```

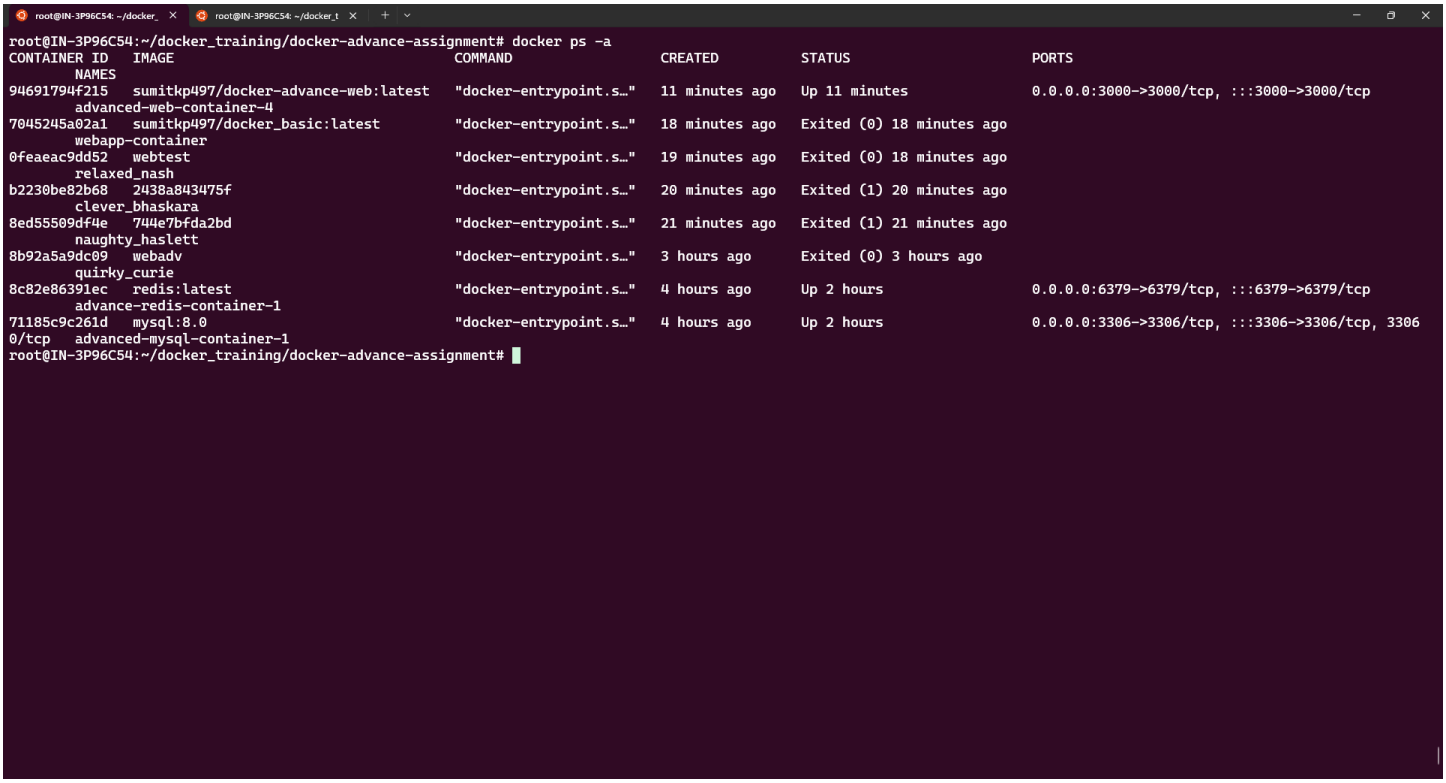
- The CD workflow for this assignment can be found under ".github/workflows/docker-local-deploy.yml"
- The steps for the workflow are:
    - The CD workflow only runs on one condition, that is, when the previous CI workflow has been completed.
    - The jobs deploy self-hosted runners.
    - Checks out the code
    - Pull latest docker images from Docker Hub from sumitkp497 username

- Screenshot of successful workflow:


Deploying Docker to Local - sumitkp11/docker-advance-assignment@abf60e2
https://github.com/sumitkp11/docker-advance-assignment/actions/runs/12294546596
sumitkp11 / docker-advance-assignment
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
Deploying Docker to Local
Deploying Docker to Local #15
Re-run all jobs
Summary
Jobs
deploy
Run details
Usage
Workflow file
Triggered via workflow run 7 minutes ago
sumitkp11 completed abf60e2
Status
Success
Total duration
1m 55s
Artifacts
–
docker-local-deploy.yml
on: workflow_run
deploy
1m 43s


root@IN-3P96C54: ~/docker_t
root@IN-3P96C54: ~/docker_
Authentication
√ Connected to GitHub
Runner Registration
Enter the name of the runner group to add this runner to: [press Enter for Default]
Enter the name of runner: [press Enter for IN-3P96C54]
This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]
√ Runner successfully added
√ Runner connection is good
Runner settings
Enter name of work folder: [press Enter for _work]
√ Settings Saved.
root@IN-3P96C54:~/docker_training/actions-runner# ./run.sh
√ Connected to GitHub
Current runner version: '2.321.0'
2024-12-12 09:21:55Z: Listening for Jobs
2024-12-12 09:22:05Z: Running job: deploy
2024-12-12 09:24:07Z: Job deploy completed with result: Failed
2024-12-12 09:32:42Z: Running job: deploy
2024-12-12 09:39:28Z: Job deploy completed with result: Failed
2024-12-12 10:04:04Z: Running job: deploy
2024-12-12 10:10:38Z: Job deploy completed with result: Failed
2024-12-12 10:22:45Z: Running job: deploy
2024-12-12 10:24:33Z: Job deploy completed with result: Succeeded





```
root@IN-3P96C54:~/docker_training/docker-advance-assignment# docker ps -a
CONTAINER ID   IMAGE                                  COMMAND                  CREATED          STATUS                      PORTS
        NAMES
94691794f215   sumitkp497/docker-advance-web:latest   "docker-entrypoint.s…"   11 minutes ago   Up 11 minutes               0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
        advanced-web-container-4
7045245a02a1   sumitkp497/docker_basic:latest         "docker-entrypoint.s…"   18 minutes ago   Exited (0) 18 minutes ago
        webapp-container
0feaeac9dd52   webtest                                "docker-entrypoint.s…"   19 minutes ago   Exited (0) 18 minutes ago
        relaxed_nash
b2230be82b68   2438a843475f                           "docker-entrypoint.s…"   20 minutes ago   Exited (1) 20 minutes ago
        clever_bhaskara
8ed55509df4e   744e7bfda2bd                           "docker-entrypoint.s…"   21 minutes ago   Exited (1) 21 minutes ago
        naughty_haslett
8b92a5a9dc09   webadv                                 "docker-entrypoint.s…"   3 hours ago      Exited (0) 3 hours ago
        quirky_curie
8c82e86391ec   redis:latest                           "docker-entrypoint.s…"   4 hours ago      Up 2 hours                  0.0.0.0:6379->6379/tcp, :::6379->6379/tcp
        advance-redis-container-1
71185c9c261d   mysql:8.0                              "docker-entrypoint.s…"   4 hours ago      Up 2 hours                  0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 3306
0/tcp   advanced-mysql-container-1
root@IN-3P96C54:~/docker_training/docker-advance-assignment#
```