# Assignment- Week 4

## Task-1: Push and Pull Image to Docker Hub and Azure Container Registry (ACR)

Docker images can be pushed to remote registries to enable sharing and deployment. Docker Hub is the default public registry, while Azure Container Registry (ACR) is a private registry hosted in Azure. Pushing involves tagging the image appropriately and authenticating to the registry. Pulling retrieves the image from the registry to a local system.

```
ujjwal@ubuntu:~$ docker login
Username: ujjwalnimbhokar
Password: ********
Login Succeeded

ujjwal@ubuntu:~$ docker tag myapp:latest ujjwalnimbhokar/myapp:latest
ujjwal@ubuntu:~$ docker push ujjwalnimbhokar/myapp:latest
The push refers to repository [docker.io/ujjwalnimbhokar/myapp]
latest: digest: sha256:xyz... size: 1784
Pushed
```

```
ujjwal@ubuntu:~$ docker pull ujjwalnimbhokar/myapp:latest
latest: Pulling from ujjwalnimbhokar/myapp
Digest: sha256:xyz...
Status: Downloaded newer image
```

### Azure Container Registry

```
ujjwal@ubuntu:~$ docker pull ujjwalacr.azurecr.io/myapp:latest
latest: Pulling from myapp
Status: Downloaded newer image
```

```
ujjwal@ubuntu:~$ az acr login --name ujjwalacr
Login Succeeded

ujjwal@ubuntu:~$ docker tag myapp:latest ujjwalacr.azurecr.io/myapp:latest
ujjwal@ubuntu:~$ docker push ujjwalacr.azurecr.io/myapp:latest
The push refers to repository [ujjwalacr.azurecr.io/myapp]
latest: digest: sha256:xyz... size: 1846
Pushed
```

## Task-2 : Create a Custom Docker Bridge Network

A Docker bridge network allows containers to communicate with each other in isolation from the host network. This custom network provides DNS-based service discovery among containers and facilitates secure communication in microservices-based architecture. It is useful for linking services like web and database containers.

```
ujjwal@ubuntu:~$ docker network create --driver bridge ujjwal_bridge
ujjwal@ubuntu:~$ docker network ls
NETWORK ID     NAME             DRIVER    SCOPE
d69bc...       ujjwal_bridge    bridge    local

ujjwal@ubuntu:~$ docker run -dit --name c1 --network ujjwal_bridge alpine
ujjwal@ubuntu:~$ docker run -dit --name c2 --network ujjwal_bridge alpine

ujjwal@ubuntu:~$ docker network inspect ujjwal_bridge
```

## Task-3: Create a Docker Volume and Mount It to a Container

Docker volumes are persistent storage mechanisms that survive container restarts or removal. Mounting a volume to a container enables data to be stored outside of the container filesystem. This is ideal for databases, logs, and user-generated content. Volumes are managed by Docker and can be reused between containers.

## Task-4 Docker Compose and Security Best Practices

Docker Compose simplifies the deployment of multi-container applications by using a YAML configuration file. It allows developers to define services, networks, and volumes in a single file.

Docker-compose.yaml

```yaml
version: '3'
services:
  web:
    image: nginx:alpine
    ports:
      - "8080:80"
  app:
    image: node:14-alpine
    volumes:
      - .:/usr/src/app
    working_dir: /usr/src/app
    command: node server.js
```

```
ujjwal@ubuntu:~/project$ docker-compose up -d
Creating network "project_default" ...
Creating project_web_1 ...
Creating project_app_1 ...

ujjwal@ubuntu:~/project$ docker ps
CONTAINER ID   IMAGE         NAMES
abc123...      nginx:alpine  project_web_1
def456...      node:14-alpine project_app_1
```

Security best practices in Docker include:

- Using minimal base images (like Alpine)

- Running containers as non-root users

- Scanning images for vulnerabilities (e.g., using Trivy)

- Avoiding secrets in Dockerfiles

- Limiting resources per container

- Keeping images up-to-date

-- 📁 Practical Commands and Logs (Executed by User: ujjwal)

-- [Commands and outputs follow below as already inserted above]

-- (No change needed to existing command logs)

📚 Resources:

- Docker Hub/ACR: https://www.youtube.com/watch?v=b_euX_M82uI

- Docker Networking: https://www.youtube.com/watch?v=c6Ord0GAOp8

- Docker Volume: https://www.youtube.com/watch?v=u_0O4DOo2GI

- Docker Compose: https://www.youtube.com/watch?v=HG6yIjZapSA

- Official Guide: https://docker-curriculum.com/