\*\*Task List: Docker Image Creation and Deployment\*\*

1. \*\*Create Directories:\*\*  
   - Establish two directories named "DHUB" and "AWSECR."

root@DESKTOP-PDOJQQA:~# mkdir DHUB

root@DESKTOP-PDOJQQA:~# mkdir AWSECR

# 2. \*\*Dockerfile Creation:\*\*

# - Develop two Dockerfiles to construct custom images with the following specifications:      - Base image: Ubuntu      - Install packages: httpd      - Add "I am from Docker Hub" to the index.html file for DHUB directory and "I am from ECR" for AWSECR directory.      - Set environment variable ENV\_NAME=DHUB for the DHUB directory and ENV\_NAME=AWSECR for the AWSECR directory.   - Start http service using ENTRYPOINT    -  Expose port 80.

# root@DESKTOP-PDOJQQA:~# cd DHUB/

# root@DESKTOP-PDOJQQA:DHUB# ll

# total 8

# drwxr-xr-x 2 root root 4096 Jan 19 15:05 ./

# drwx------ 17 root root 4096 Jan 19 15:06 ../

# root@DESKTOP-PDOJQQA:DHUB# vi Dockerfile

# root@DESKTOP-PDOJQQA:DHUB# vi index.html

FROM ubuntu

RUN apt-get update && apt-get install -y apache2

ENV ENV\_NAME=DHUB

EXPOSE 80

ENTRYPOINT ["apache2ctl", "-D", "FOREGROUND"]

root@DESKTOP-PDOJQQA:~# **cd AWSECR/**

root@DESKTOP-PDOJQQA:AWSECR# **ll**

total 16

drwxr-xr-x 2 root root 4096 Jan 19 15:24 ./

drwx------ 17 root root 4096 Jan 19 15:24 ../

-rw-r--r-- 1 root root 141 Jan 19 15:24 Dockerfile

-rw-r--r-- 1 root root 58 Jan 19 15:23 index.file

# 3. \*\*Build Custom Images:\*\*

# - Utilize the docker build command to build the custom images.    - List all available images using docker images.

# root@DESKTOP-PDOJQQA:AWSECR# docker image build -t my\_awscer\_image .

# [+] Building 32.3s (6/6) FINISHED docker:default

# => [internal] load build definition from Dockerfile 0.1s

# => => transferring dockerfile: 180B 0.0s

# => [internal] load .dockerignore 0.1s

# => => transferring context: 2B 0.0s

# => [internal] load metadata for docker.io/library/ubuntu:latest 0.0s

# => CACHED [1/2] FROM docker.io/library/ubuntu 0.0s

# => [2/2] RUN apt-get update && apt-get install -y apache2 30.6s

# => exporting to image 1.6s

# => => exporting layers 1.5s

# => => writing image sha256:4da104e09c5bd86ed0cda343a180cad5cdb91da2d2bf1af7a89cb7e8db83c058 0.0s

# => => naming to docker.io/library/my\_awscer\_image 0.0s

# root@DESKTOP-PDOJQQA:DHUB# docker image build -t my\_dhub\_image .

# [+] Building 0.2s (6/6) FINISHED docker:default

# => [internal] load .dockerignore 0.0s

# => => transferring context: 2B 0.0s

# => [internal] load build definition from Dockerfile 0.0s

# => => transferring dockerfile: 185B 0.0s

# => [internal] load metadata for docker.io/library/ubuntu:latest 0.0s

# => [1/2] FROM docker.io/library/ubuntu:latest 0.0s

# => CACHED [2/2] RUN apt-get update && apt-get install -y apache2 0.0s

# => exporting to image 0.0s

# => => exporting layers 0.0s

# => => writing image sha256:8c6a6516a7fa4045cd9a90818b97240b490cb790d90b5fdba19739252d8d1ba3 0.0s

# => => naming to docker.io/library/my\_dhub\_image 0.0s

# root@DESKTOP-PDOJQQA:DHUB# docker image ls

# REPOSITORY TAG IMAGE ID CREATED SIZE

# my\_awscer\_image latest 4da104e09c5b 5 minutes ago 233MB

# my\_dhub\_image latest 8c6a6516a7fa 5 minutes ago 233MB

# 4. \*\*Push Images to Repositories:\*\*

# - Push the Docker image to Docker Hub.    - Push the Docker image to AWS ECR.

# root@DESKTOP-PDOJQQA:DHUB# docker image tag my\_dhub\_image:latest vivekmore5292/local\_app\_images:v1

# root@DESKTOP-PDOJQQA:DHUB# docker image tag my\_awscer\_image:latest vivekmore5292/local\_app\_images:v2

# root@DESKTOP-PDOJQQA:DHUB# docker image push vivekmore5292/local\_app\_images:v1

# The push refers to repository [docker.io/vivekmore5292/local\_app\_images]

# 272b0b3264af: Pushed

# a1360aae5271: Mounted from library/ubuntu

# v1: digest: sha256:78218648f1c5cbc772ae3d33f124690ee614dd89cecf074a9a8157fc32880d69 size: 741

# root@DESKTOP-PDOJQQA:DHUB# docker image push vivekmore5292/local\_app\_images:v2

# The push refers to repository [docker.io/vivekmore5292/local\_app\_images]

# 272b0b3264af: Layer already exists

# a1360aae5271: Layer already exists

# v2: digest: sha256:ed305d182ebfcabc47c678f25cb8d2f7fcf9994f3d378ec63382dc2880485e35 size: 741

# Push the Docker image to AWS ECR:

# root@DESKTOP-PDOJQQA:DHUB# aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 891301382865.dkr.ecr.us-east-2.amazonaws.com

# WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

# Configure a credential helper to remove this warning. See

# https://docs.docker.com/engine/reference/commandline/login/#credentials-store

# Login Succeeded

# Tag images for AWS

# root@DESKTOP-PDOJQQA:DHUB# docker tag 8c6a6516a7fa 891301382865.dkr.ecr.us-east-2.amazonaws.com/my-localrepo:v1

# root@DESKTOP-PDOJQQA:DHUB# docker tag 4da104e09c5b 891301382865.dkr.ecr.us-east-2.amazonaws.com/my-localrepo:v2

# Push image into AWS ECR:

# root@DESKTOP-PDOJQQA:DHUB# docker push 891301382865.dkr.ecr.us-east-2.amazonaws.com/my-localrepo:v1

# The push refers to repository [891301382865.dkr.ecr.us-east-2.amazonaws.com/my-localrepo]

# 272b0b3264af: Pushed

# a1360aae5271: Pushed

# v1: digest: sha256:3c67c707414265bd3e0dc7ba49bbd1a3f3a2907c8aa5e655af089320ea866c52 size: 741

# root@DESKTOP-PDOJQQA:DHUB# docker push 891301382865.dkr.ecr.us-east-2.amazonaws.com/my-localrepo:v2

# The push refers to repository [891301382865.dkr.ecr.us-east-2.amazonaws.com/my-localrepo]

# 272b0b3264af: Layer already exists

# a1360aae5271: Layer already exists

# v2: digest: sha256:47a74ed45023ddb13fefcec1f7eb7098d6e35837ab9ee4983656b84a941bc895 size: 741

# 

# Steps: To create a repository (AWS Management Console)

# Open the Amazon ECR console

From the navigation bar, choose the Region to create your repository in.

# In the navigation pane, choose Repositories

On the Repositories page, choose the Private tab, and then choose Create repository.

For Visibility settings, verify that Private is selected.

# For Repository name, enter a unique name for your repository

# For Login AWS ECR on CLI

# aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws\_account\_id.dkr.ecr.region.amazonaws.com.

# For tag Image in AWS ECR Format:

# docker tag e9ae3c220b23 (image id) 891301382865 (aws\_account\_id).dkr.ecr.us-west-2(zone).amazonaws.com/my-repository:tag (aws repo name with : tag)

# Push the image using the docker push command into AWS ECR:

# docker push 891301382865 (aws\_account\_id).dkr.ecr.us-west-2(Zone).amazonaws.com/my-repository:tag (aws repo name with : tag)

# 5. \*\*Run Containers:\*\*

# - Execute a container from the Docker Hub image, naming it "I\_AM\_FROM\_DHUB," and mapping host port 8081 to the container.

# docker run -d -p 8081:80 --name I\_AM\_FROM\_DHUB your\_docker\_hub\_image

# Execute a container from the AWS ECR image, naming it "I\_AM\_FROM\_ECR," and mapping host port 8082 to the container.

# docker run -d -p 8082:80 --name I\_AM\_FROM\_ECR your\_aws\_ecr\_image.

# root@DESKTOP-PDOJQQA:~# docker container run -itd --name=DHUB -p 8081:80 my\_dhub\_image:latest

# 04cdca9997f804462f1af90d8534999ac0a215c3277d5059691989ec27a86178

# root@DESKTOP-PDOJQQA:~# docker container run -itd --name=AWSECR -p 8082:80 my\_awscer\_image:latest

# 4a0908f7f8b357583b51299caada4e6261b02f2d940a91038e0327de22d6fae9

# root@DESKTOP-PDOJQQA:~# docker container ls

# CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS

# NAMES

# 4a0908f7f8b3 my\_awscer\_image:latest "apache2ctl -D FOREG…" 12 seconds ago Up 11 seconds 0.0.0.0:8082->80/tcp, :::8082->80/tcp AWSECR

# 04cdca9997f8 my\_dhub\_image:latest "apache2ctl -D FOREG…" About a minute ago Up About a minute 0.0.0.0:8081->80/tcp, :::8081->80/tcp DHUB

# root@DESKTOP-PDOJQQA:~# docker container exec -it DHUB /bin/bash

# root@04cdca9997f8:/# echo $ENV\_NAME

# DHUB

# root@DESKTOP-PDOJQQA:~# docker container exec -it AWSECR /bin/bash

# root@04cdca9997f8:/# echo $ENV\_NAME

# AWSECR

# 6. \*\*Access Pages from Browser:\*\*

# - Open a web browser and access the pages:      - For Docker Hub: <http://localhost:8081>      - For AWS ECR: <http://localhost:8082>

# 

# This task list guides you through the process of creating custom Docker images, pushing them to Docker Hub and AWS ECR, running containers, and accessing the pages from a browser.

# root@DESKTOP-PDOJQQA:~# curl 172.17.0.6

# I am from Docker Hub

# root@DESKTOP-PDOJQQA:~# curl 172.17.0.7

# I am from ECR