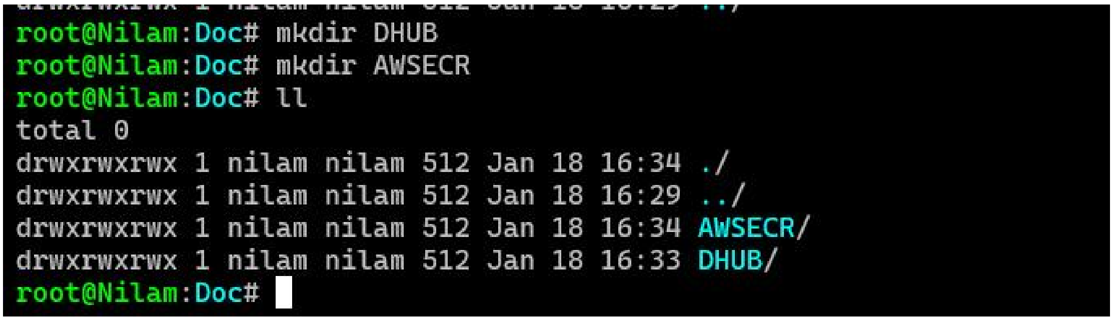
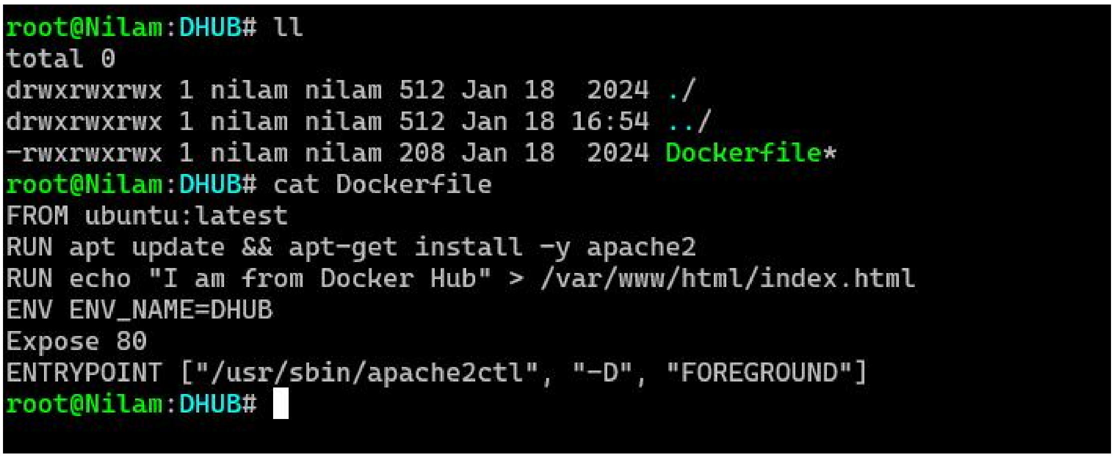
*1. **Create Directories:***
- Establish two directories named "DHUB" and "AWSECR."



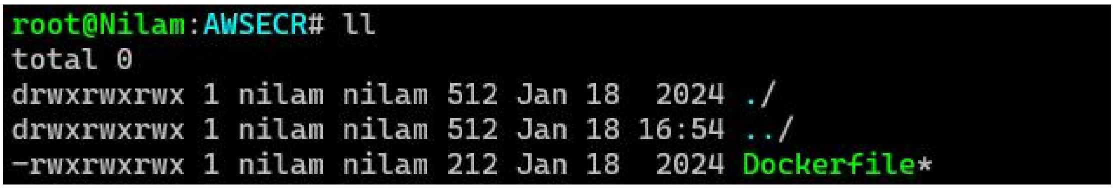
```
root@Nilam:Doc# mkdir DHUB
root@Nilam:Doc# mkdir AWSECR
root@Nilam:Doc# ll
total 0
drwxrwxrwx 1 nilam nilam 512 Jan 18 16:34 ./
drwxrwxrwx 1 nilam nilam 512 Jan 18 16:29 ../
drwxrwxrwx 1 nilam nilam 512 Jan 18 16:34 AWSECR/
drwxrwxrwx 1 nilam nilam 512 Jan 18 16:33 DHUB/
root@Nilam:Doc#
```

*2. **Dockerfile Creation:***

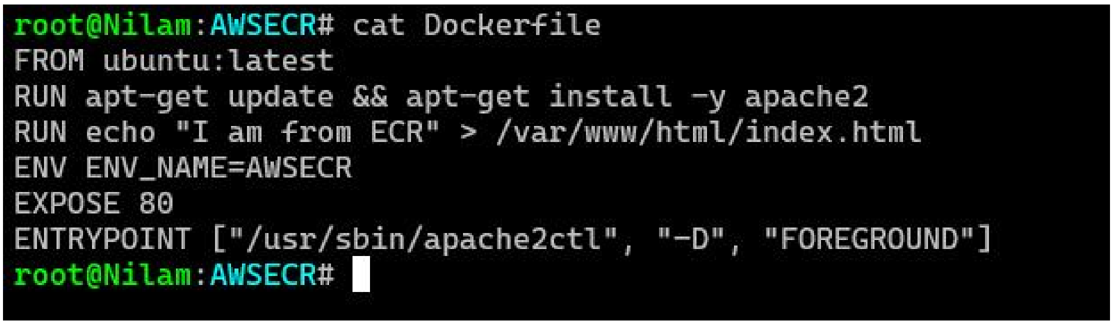
- Develop two Dockerfiles to construct custom images with the following specifications:
- Base image: Ubuntu
- Install packages: httpd
- Add "I am from Docker Hub" to the index.html file for DHUB directory and "I am from ECR" for AWSECR directory.
- Set environment variable ENV_NAME=DHUB for the DHUB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



```
root@Nilam:DHUB# ll
total 0
drwxrwxrwx 1 nilam nilam 512 Jan 18  2024 ./
drwxrwxrwx 1 nilam nilam 512 Jan 18 16:54 ../
-rwxrwxrwx 1 nilam nilam 208 Jan 18  2024 Dockerfile*
root@Nilam:DHUB# cat Dockerfile
FROM ubuntu:latest
RUN apt update && apt-get install -y apache2
RUN echo "I am from Docker Hub" > /var/www/html/index.html
ENV ENV_NAME=DHUB
Expose 80
ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
root@Nilam:DHUB#
```



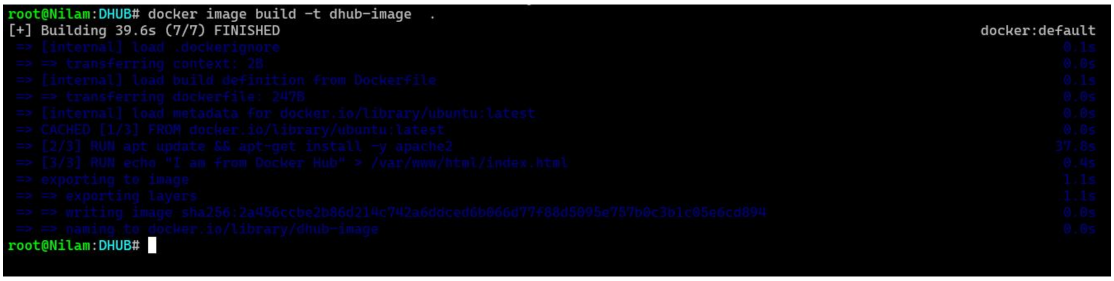
```
root@Nilam:AWSECR# ll
total 0
drwxrwxrwx 1 nilam nilam 512 Jan 18  2024 ./
drwxrwxrwx 1 nilam nilam 512 Jan 18 16:54 ../
-rwxrwxrwx 1 nilam nilam 212 Jan 18  2024 Dockerfile*
```



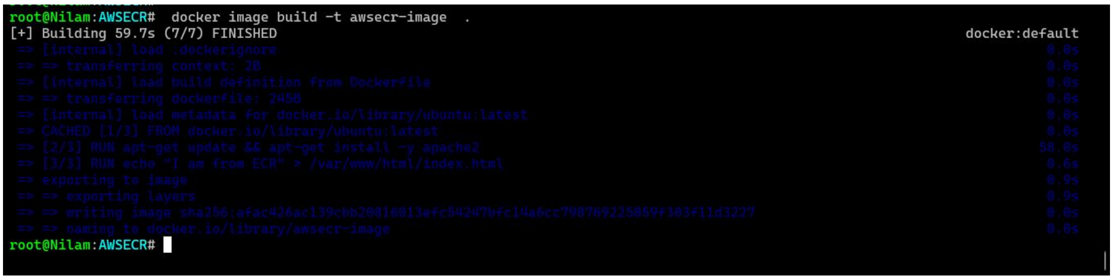
```
root@Nilam:AWSECR# cat Dockerfile
FROM ubuntu:latest
RUN apt-get update && apt-get install -y apache2
RUN echo "I am from ECR" > /var/www/html/index.html
ENV ENV_NAME=AWSECR
EXPOSE 80
ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
root@Nilam:AWSECR#
```

*3. **Build Custom Images:***

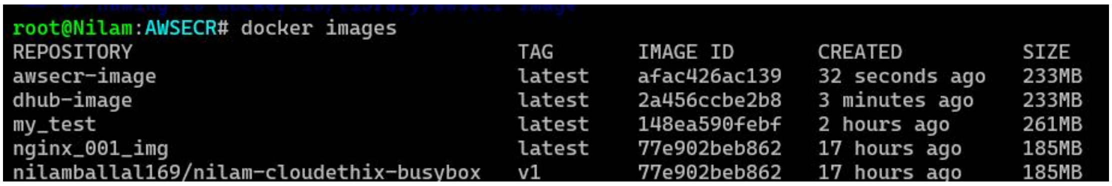
- Utilize the docker build command to build the custom images.
- List all available images using docker images.



```
root@Nilam:DHUB# docker image build -t dhub-image  .
[+] Building 39.6s (7/7) FINISHED                                                                     docker:default
 => [internal] load .dockerignore                                                                               0.1s
 => => transferring context: 2B                                                                                 0.0s
 => [internal] load build definition from Dockerfile                                                            0.1s
 => => transferring dockerfile: 247B                                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                0.0s
 => CACHED [1/3] FROM docker.io/library/ubuntu:latest                                                           0.0s
 => [2/3] RUN apt update && apt-get install -y apache2                                                         37.8s
 => [3/3] RUN echo "I am from Docker Hub" > /var/www/html/index.html                                            0.4s
 => exporting to image                                                                                          1.1s
 => => exporting layers                                                                                         1.1s
 => => writing image sha256:2a456ccbe2b86d214c742a6ddced6b066d77f88d5095e757b0c3b1c05e6cd894                    0.0s
 => => naming to docker.io/library/dhub-image                                                                   0.0s
root@Nilam:DHUB#
```



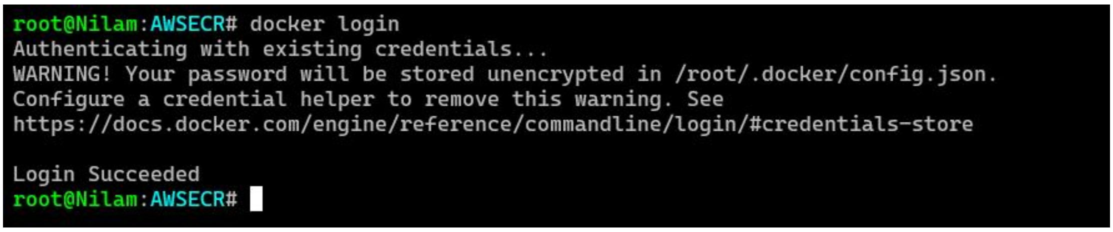
```
root@Nilam:AWSECR#  docker image build -t awsecr-image  .
[+] Building 59.7s (7/7) FINISHED                                                                   docker:default
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [internal] load build definition from Dockerfile                                                          0.0s
 => => transferring dockerfile: 245B                                                                          0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                              0.0s
 => CACHED [1/3] FROM docker.io/library/ubuntu:latest                                                         0.0s
 => [2/3] RUN apt-get update && apt-get install -y apache2                                                   58.0s
 => [3/3] RUN echo "I am from ECR" > /var/www/html/index.html                                                 0.6s
 => exporting to image                                                                                        0.9s
 => => exporting layers                                                                                       0.9s
 => => writing image sha256:afac426ac139cbb20816013efc54247bfc14a6cc798769225859f303f11d3227                  0.0s
 => => naming to docker.io/library/awsecr-image                                                               0.0s
root@Nilam:AWSECR#
```



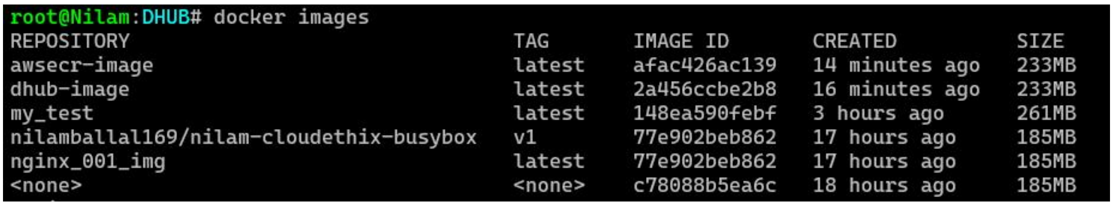
```
root@Nilam:AWSECR# docker images
REPOSITORY                              TAG       IMAGE ID       CREATED            SIZE
awsecr-image                            latest    afac426ac139   32 seconds ago     233MB
dhub-image                              latest    2a456ccbe2b8   3 minutes ago      233MB
my_test                                 latest    148ea590febf   2 hours ago        261MB
nginx_001_img                           latest    77e902beb862   17 hours ago       185MB
nilamballal169/nilam-cloudethix-busybox v1        77e902beb862   17 hours ago       185MB
```

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

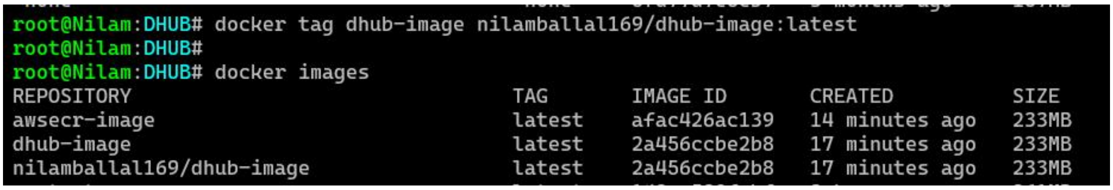
- Push the Docker image to Docker Hub.
- Push the Docker image to AWS ECR.



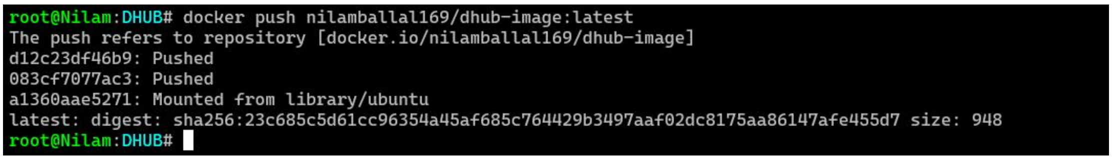
```
root@Nilam:AWSECR# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@Nilam:AWSECR#
```



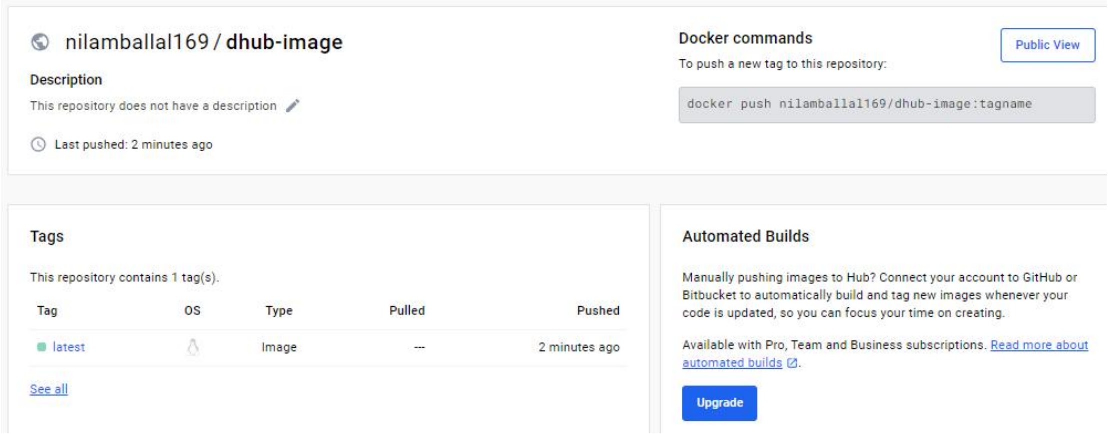
```
root@Nilam:DHUB# docker images
REPOSITORY                                 TAG       IMAGE ID       CREATED          SIZE
awsecr-image                               latest    afac426ac139   14 minutes ago   233MB
dhub-image                                 latest    2a456ccbe2b8   16 minutes ago   233MB
my_test                                    latest    148ea590febf   3 hours ago      261MB
nilamballal169/nilam-cloudethix-busybox    v1        77e902beb862   17 hours ago     185MB
nginx_001_img                              latest    77e902beb862   17 hours ago     185MB
<none>                                     <none>    c78088b5ea6c   18 hours ago     185MB
```



```
root@Nilam:DHUB# docker tag dhub-image nilamballal169/dhub-image:latest
root@Nilam:DHUB#
root@Nilam:DHUB# docker images
REPOSITORY                                 TAG       IMAGE ID       CREATED          SIZE
awsecr-image                               latest    afac426ac139   14 minutes ago   233MB
dhub-image                                 latest    2a456ccbe2b8   17 minutes ago   233MB
nilamballal169/dhub-image                  latest    2a456ccbe2b8   17 minutes ago   233MB
```



```
root@Nilam:DHUB# docker push nilamballal169/dhub-image:latest
The push refers to repository [docker.io/nilamballal169/dhub-image]
d12c23df46b9: Pushed
083cf7077ac3: Pushed
a1360aae5271: Mounted from library/ubuntu
latest: digest: sha256:23c685c5d61cc96354a45af685c764429b3497aaf02dc8175aa86147afe455d7 size: 948
root@Nilam:DHUB#
```



nilamballal169 / **dhub-image**

Description

This repository does not have a description

Last pushed: 2 minutes ago

Docker commands

To push a new tag to this repository:

```
docker push nilamballal169/dhub-image:tagname
```

Tags

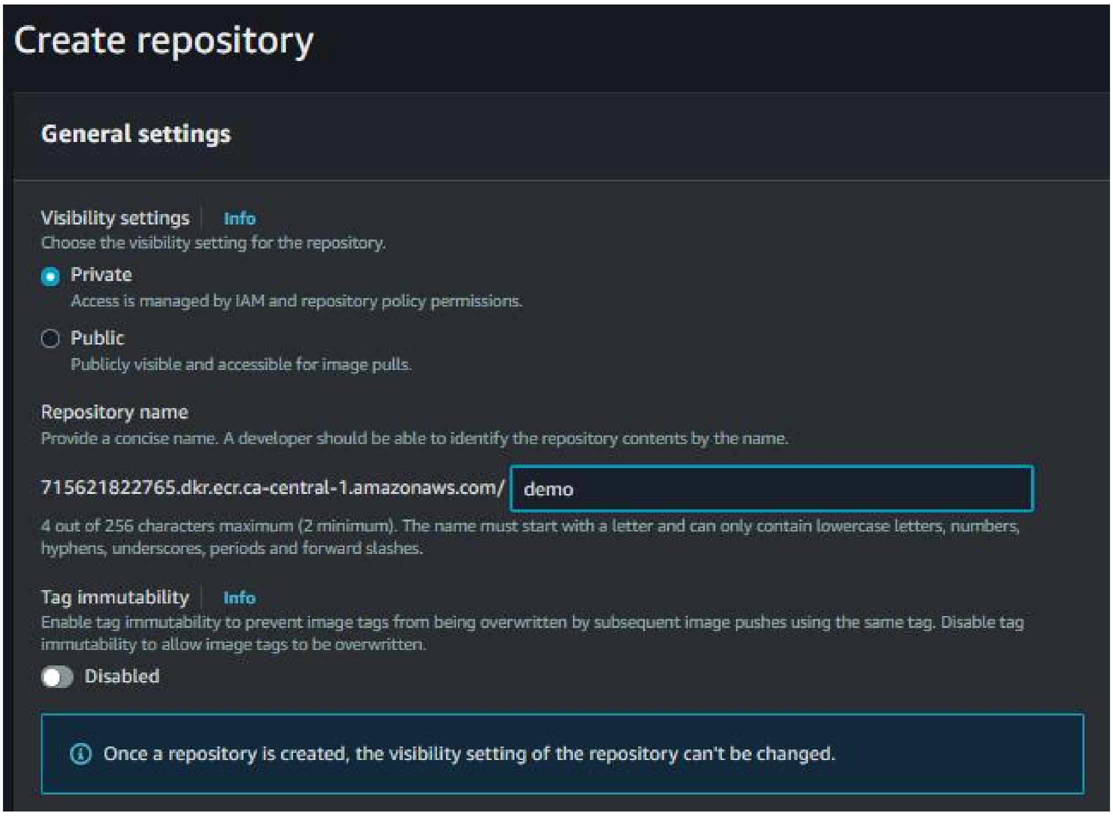
This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	2 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.



Create repository

General settings

Visibility settings | Info

Choose the visibility setting for the repository.

● Private

Access is managed by IAM and repository policy permissions.

○ Public

Publicly visible and accessible for image pulls.

Repository name

Provide a concise name. A developer should be able to identify the repository contents by the name.

715621822765.dkr.ecr.ca-central-1.amazonaws.com/ demo

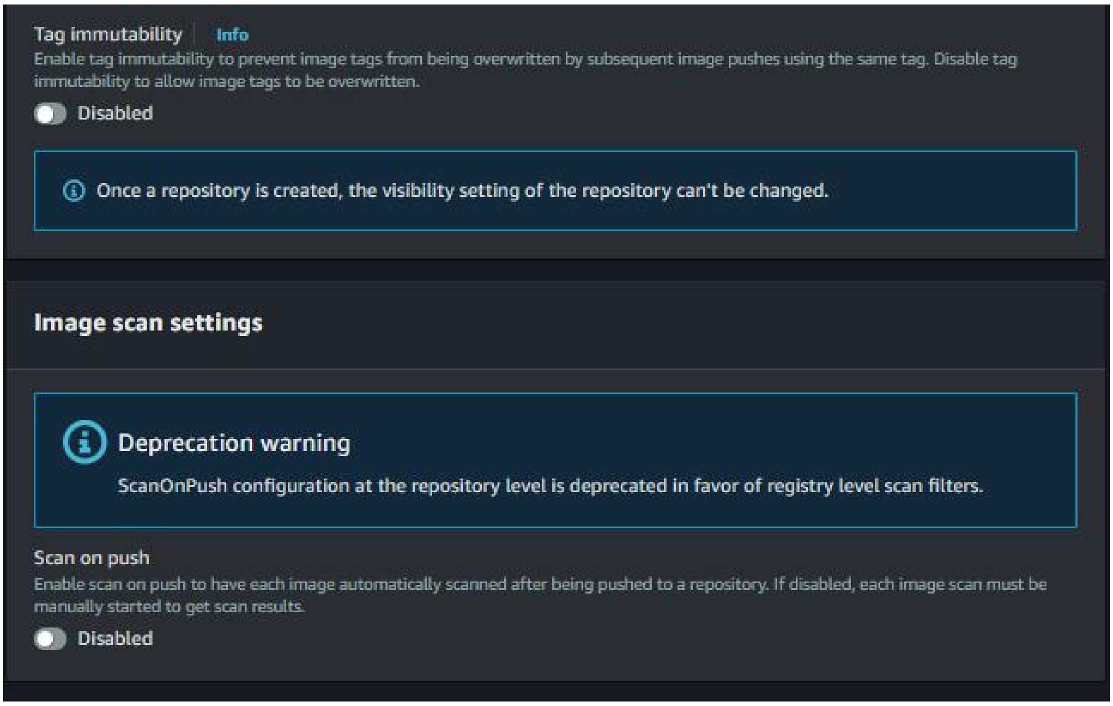
4 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

Tag immutability | Info

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

Disabled

Once a repository is created, the visibility setting of the repository can't be changed.



Tag immutability | Info

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

Disabled

Once a repository is created, the visibility setting of the repository can't be changed.

Image scan settings

Deprecation warning

ScanOnPush configuration at the repository level is deprecated in favor of registry level scan filters.

Scan on push

Enable scan on push to have each image automatically scanned after being pushed to a repository. If disabled, each image scan must be manually started to get scan results.

Disabled

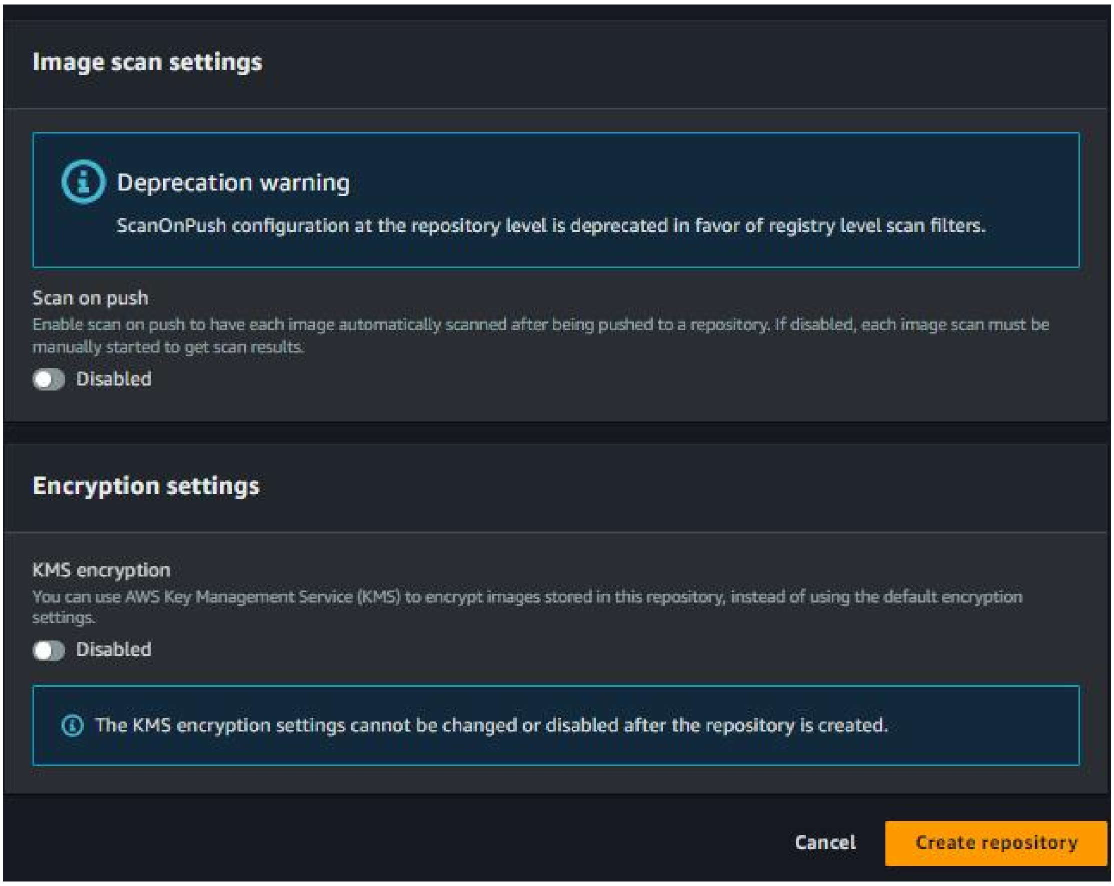


Image scan settings

Deprecation warning

ScanOnPush configuration at the repository level is deprecated in favor of registry level scan filters.

Scan on push

Enable scan on push to have each image automatically scanned after being pushed to a repository. If disabled, each image scan must be manually started to get scan results.

Disabled

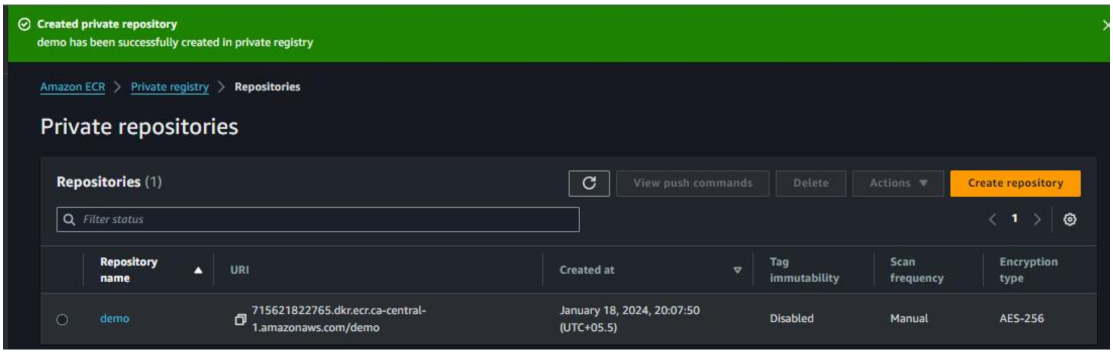
Encryption settings

KMS encryption

You can use AWS Key Management Service (KMS) to encrypt images stored in this repository, instead of using the default encryption settings.

Disabled

The KMS encryption settings cannot be changed or disabled after the repository is created.

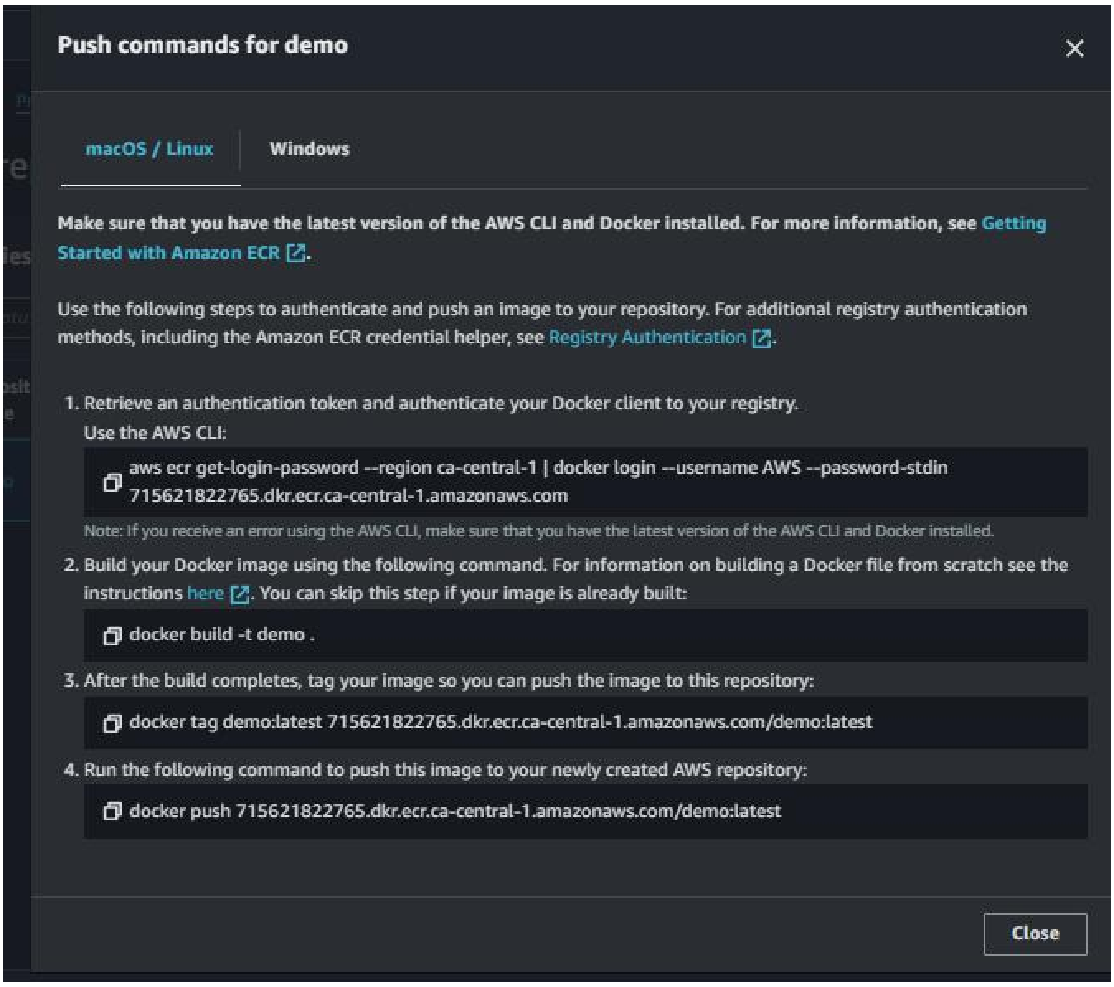


Created private repository
demo has been successfully created in private registry

Private repositories

Repositories (1)

	Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type
○	demo	715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo	January 18, 2024, 20:07:50 (UTC+05.5)	Disabled	Manual	AES-256



Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry.
   Use the AWS CLI:

   ```
   aws ecr get-login-password --region ca-central-1 | docker login --username AWS --password-stdin 715621822765.dkr.ecr.ca-central-1.amazonaws.com
   ```

   Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:

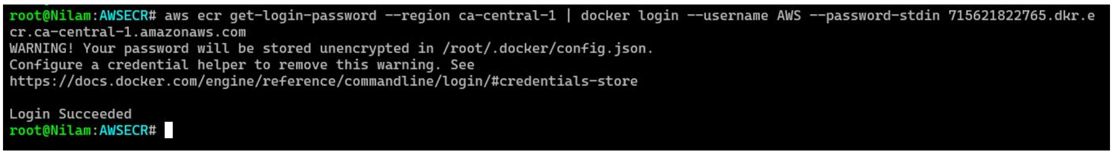
   ```
   docker build -t demo .
   ```

3. After the build completes, tag your image so you can push the image to this repository:

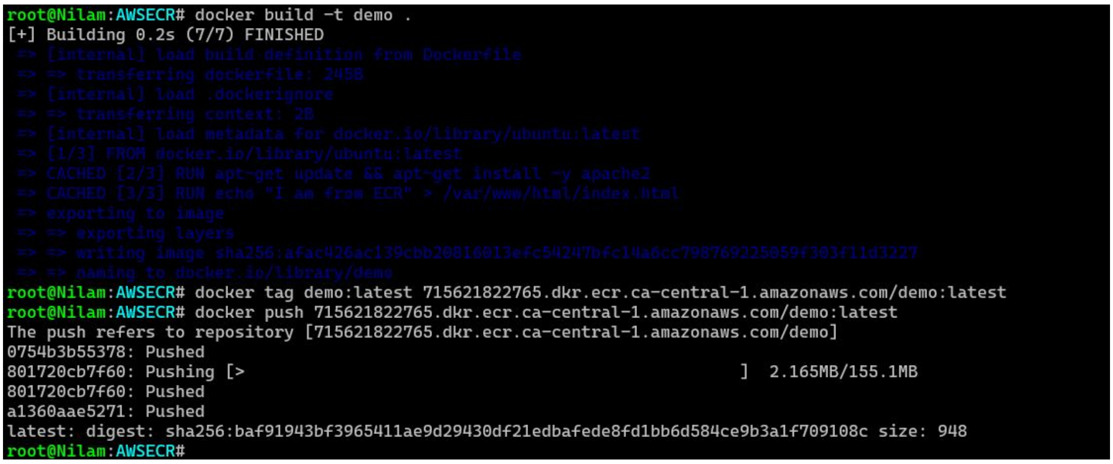
   ```
   docker tag demo:latest 715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo:latest
   ```

4. Run the following command to push this image to your newly created AWS repository:

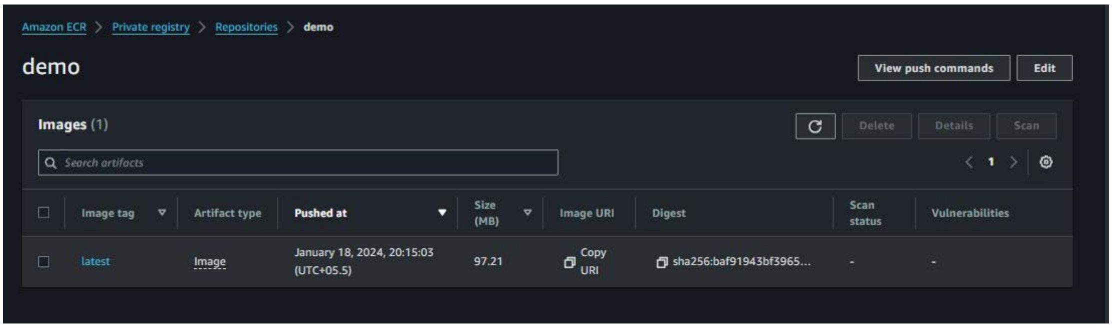
   ```
   docker push 715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo:latest
   ```



```
root@Nilam:AWSECR# aws ecr get-login-password --region ca-central-1 | docker login --username AWS --password-stdin 715621822765.dkr.e
cr.ca-central-1.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
root@Nilam:AWSECR#
```



```
root@Nilam:AWSECR# docker build -t demo .
[+] Building 0.2s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 245B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [1/3] FROM docker.io/library/ubuntu:latest
 => CACHED [2/3] RUN apt-get update && apt-get install -y apache2
 => CACHED [3/3] RUN echo "I am from ECR" > /var/www/html/index.html
 => exporting to image
 => => exporting layers
 => => writing image sha256:afac426ac139cbb20816013efc54247bfc14a6cc798769225059f303f11d3227
 => => naming to docker.io/library/demo
root@Nilam:AWSECR# docker tag demo:latest 715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo:latest
root@Nilam:AWSECR# docker push 715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo:latest
The push refers to repository [715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo]
0754b3b55378: Pushed
801720cb7f60: Pushing [>                                                  ]  2.165MB/155.1MB
801720cb7f60: Pushed
a1360aae5271: Pushed
latest: digest: sha256:baf91943bf3965411ae9d29430df21edbafede8fd1bb6d584ce9b3a1f709108c size: 948
root@Nilam:AWSECR#
```



demo

Images (1)

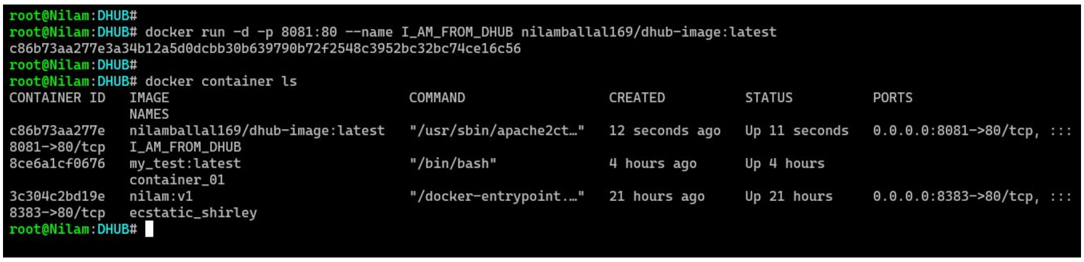
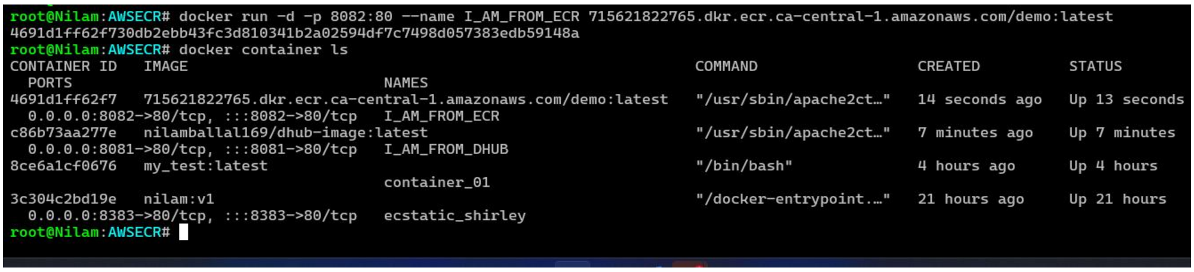
□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities
□	latest	Image	January 18, 2024, 20:15:03 (UTC+05.5)	97.21	Copy URI	sha256:baf91943bf3965...	-	-

*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.

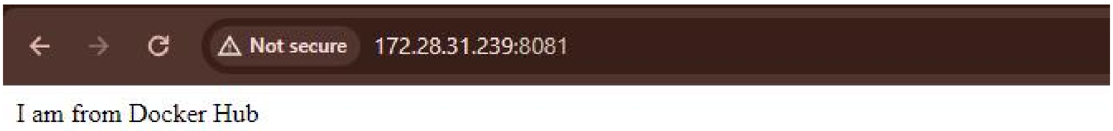
```
docker run -d -p 8081:80 --name I_AM_FROM_DHUB your_docker_hub_image
```

- 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@Nilam:DHUB#
root@Nilam:DHUB# docker run -d -p 8081:80 --name I_AM_FROM_DHUB nilamballal169/dhub-image:latest
c86b73aa277e3a34b12a5d0dcbb30b639790b72f2548c3952bc32bc74ce16c56
root@Nilam:DHUB#
root@Nilam:DHUB# docker container ls
CONTAINER ID   IMAGE                              COMMAND                  CREATED          STATUS          PORTS
               NAMES
c86b73aa277e   nilamballal169/dhub-image:latest   "/usr/sbin/apache2ct…"   12 seconds ago   Up 11 seconds   0.0.0.0:8081->80/tcp, :::
8081->80/tcp   I_AM_FROM_DHUB
8ce6a1cf0676   my_test:latest                     "/bin/bash"              4 hours ago      Up 4 hours
               container_01
3c304c2bd19e   nilam:v1                           "/docker-entrypoint.…"   21 hours ago     Up 21 hours     0.0.0.0:8383->80/tcp, :::
8383->80/tcp   ecstatic_shirley
root@Nilam:DHUB#
```



```
root@Nilam:AWSECR# docker run -d -p 8082:80 --name I_AM_FROM_ECR 715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo:latest
4691d1ff62f730db2ebb43fc3d810341b2a02594df7c7498d057383edb59148a
root@Nilam:AWSECR# docker container ls
CONTAINER ID   IMAGE                                                        COMMAND                  CREATED          STATUS
  PORTS                                     NAMES
4691d1ff62f7   715621822765.dkr.ecr.ca-central-1.amazonaws.com/demo:latest   "/usr/sbin/apache2ct…"   14 seconds ago   Up 13 seconds
  0.0.0.0:8082->80/tcp, :::8082->80/tcp   I_AM_FROM_ECR
c86b73aa277e   nilamballal169/dhub-image:latest                             "/usr/sbin/apache2ct…"   7 minutes ago    Up 7 minutes
  0.0.0.0:8081->80/tcp, :::8081->80/tcp   I_AM_FROM_DHUB
8ce6a1cf0676   my_test:latest                                               "/bin/bash"              4 hours ago      Up 4 hours
                                          container_01
3c304c2bd19e   nilam:v1                                                     "/docker-entrypoint.…"   21 hours ago     Up 21 hours
  0.0.0.0:8383->80/tcp, :::8383->80/tcp   ecstatic_shirley
root@Nilam:AWSECR#
```
```

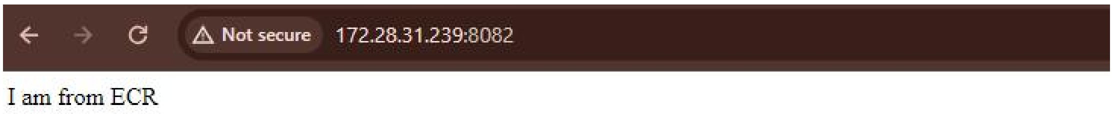
6. **Access Pages from Browser:**

- Open a web browser and access the pages:
  - For Docker Hub: http://localhost:8081



I am from Docker Hub

  - For AWS ECR: http://localhost:8082



I am from ECR

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.