LAB: Upload an Image to ECR

You need:

- An AWS Account

Duration of the Lab: 30 Minutes.

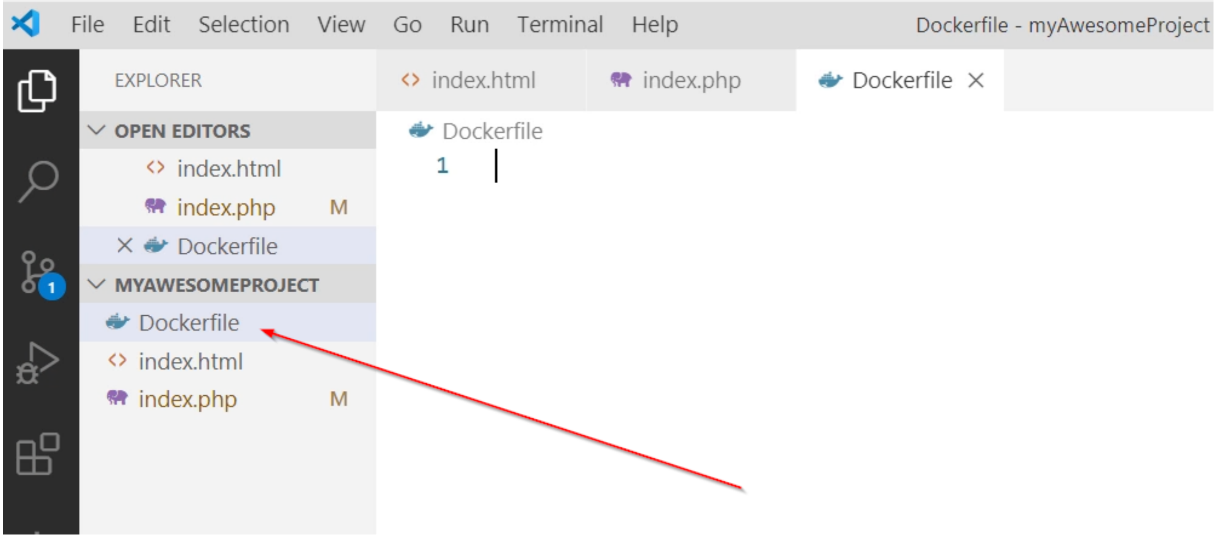
Difficulty: medium

Create a Docker Image

From the Docker-Hub we will use a PHP Image: https://hub.docker.com/_/php

And we will create a new Image from this Dockerfile.

In your Project folder create a “Dockerfile”:



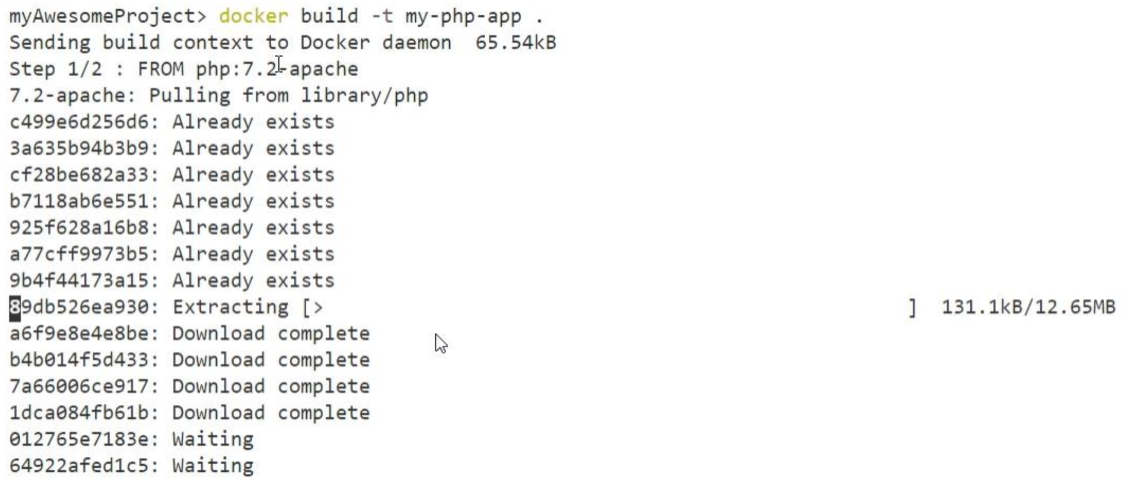
And paste the following contents:

```
FROM php:7.2-apache
COPY index.php /var/www/html
```

Build the Image

Use the following command to build the image:

```
docker build -t my-php-app .
```



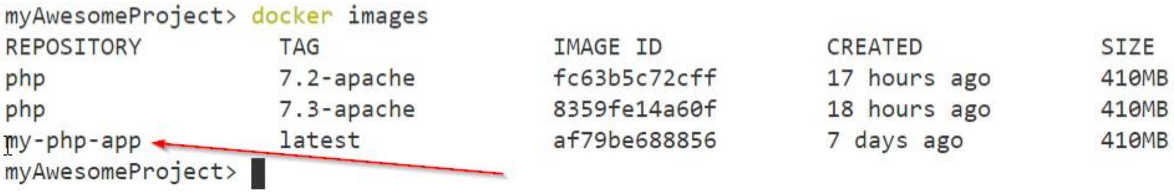
```
myAwesomeProject> docker build -t my-php-app .
Sending build context to Docker daemon  65.54kB
Step 1/2 : FROM php:7.2-apache
7.2-apache: Pulling from library/php
c499e6d256d6: Already exists
3a635b94b3b9: Already exists
cf28be682a33: Already exists
b7118ab6e551: Already exists
925f628a16b8: Already exists
a77cff9973b5: Already exists
9b4f44173a15: Already exists
89db526ea930: Extracting [>                                                  ]  131.1kB/12.65MB
a6f9e8e4e8be: Download complete
b4b014f5d433: Download complete
7a66006ce917: Download complete
1dca084fb61b: Download complete
012765e7183e: Waiting
64922afed1c5: Waiting
```

List your Images

If you run the command

```
docker images
```

then you should see your newly created image.



```
myAwesomeProject> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
php                 7.2-apache          fc63b5c72cff        17 hours ago        410MB
php                 7.3-apache          8359fe14a60f        18 hours ago        410MB
my-php-app          latest              af79be688856        7 days ago          410MB
myAwesomeProject>
```

Run the Container from the Image

Now we can run the newly created image. Mind: You *can* omit the volume mounting, because in the dockerfile we said we are already copying the index.php file into the container.

```
docker run -p 8080:80 --rm my-php-app
```

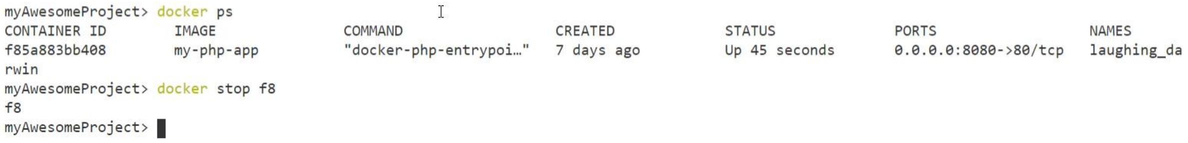
This should run the container and you can open localhost:8080 in your browser.

Stop the container

Stop the container now with ctrl+c.

Check if the container is still running (docker ps)

And stop it if it’s still there: (docker stop [containerID])

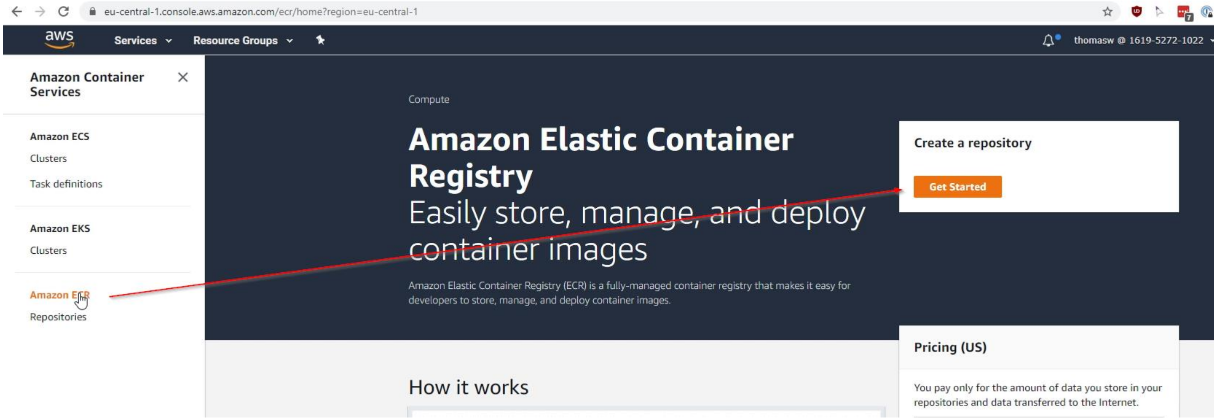


```
myAwesomeProject> docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                  NAMES
f85a883bb408        my-php-app          "docker-php-entrypoi…"   7 days ago          Up 45 seconds       0.0.0.0:8080->80/tcp   laughing_da
rwin
myAwesomeProject> docker stop f8
f8
myAwesomeProject>
```

Upload the Image to ECR

Open the Elastic Container Registry in your Console:

And choose “Get Started” to create a new repository.



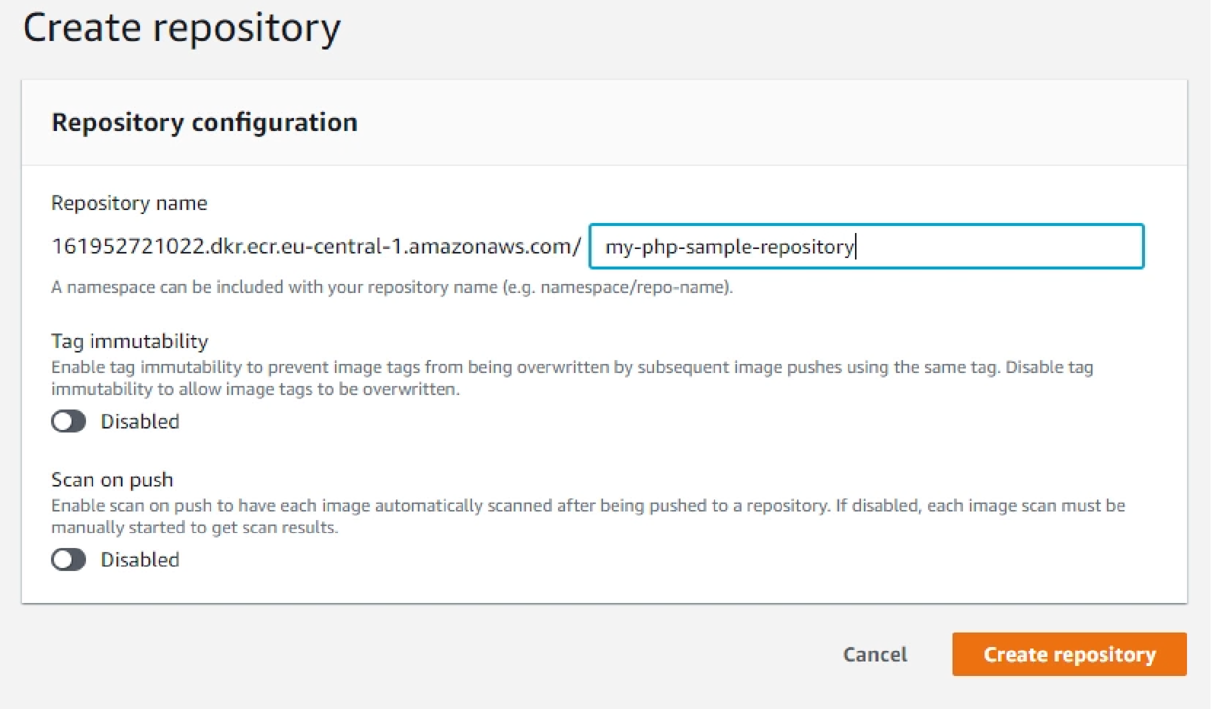
Amazon Elastic Container Registry

Easily store, manage, and deploy container images

Amazon Elastic Container Registry (ECR) is a fully-managed container registry that makes it easy for developers to store, manage, and deploy container images.

You pay only for the amount of data you store in your repositories and data transferred to the Internet.

Give the repository a name of your choice (my-php-sample-repository for example)



Create repository

Repository configuration

Repository name

161952721022.dkr.ecr.eu-central-1.amazonaws.com/ my-php-sample-repository

A namespace can be included with your repository name (e.g. namespace/repo-name).

Tag immutability

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

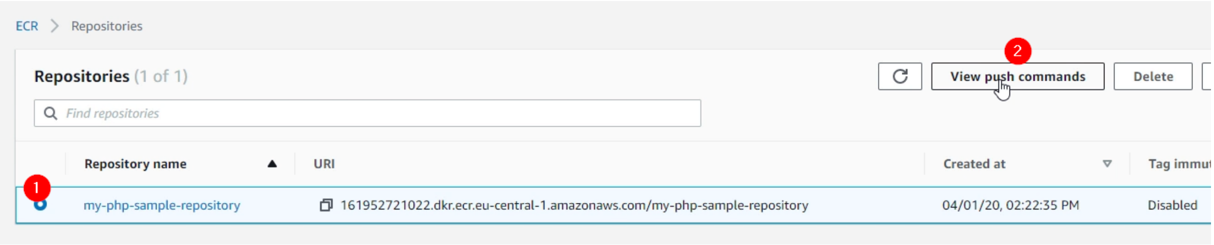
Scan on push

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

Upload your Image from your Local Repository to your ECR Repository

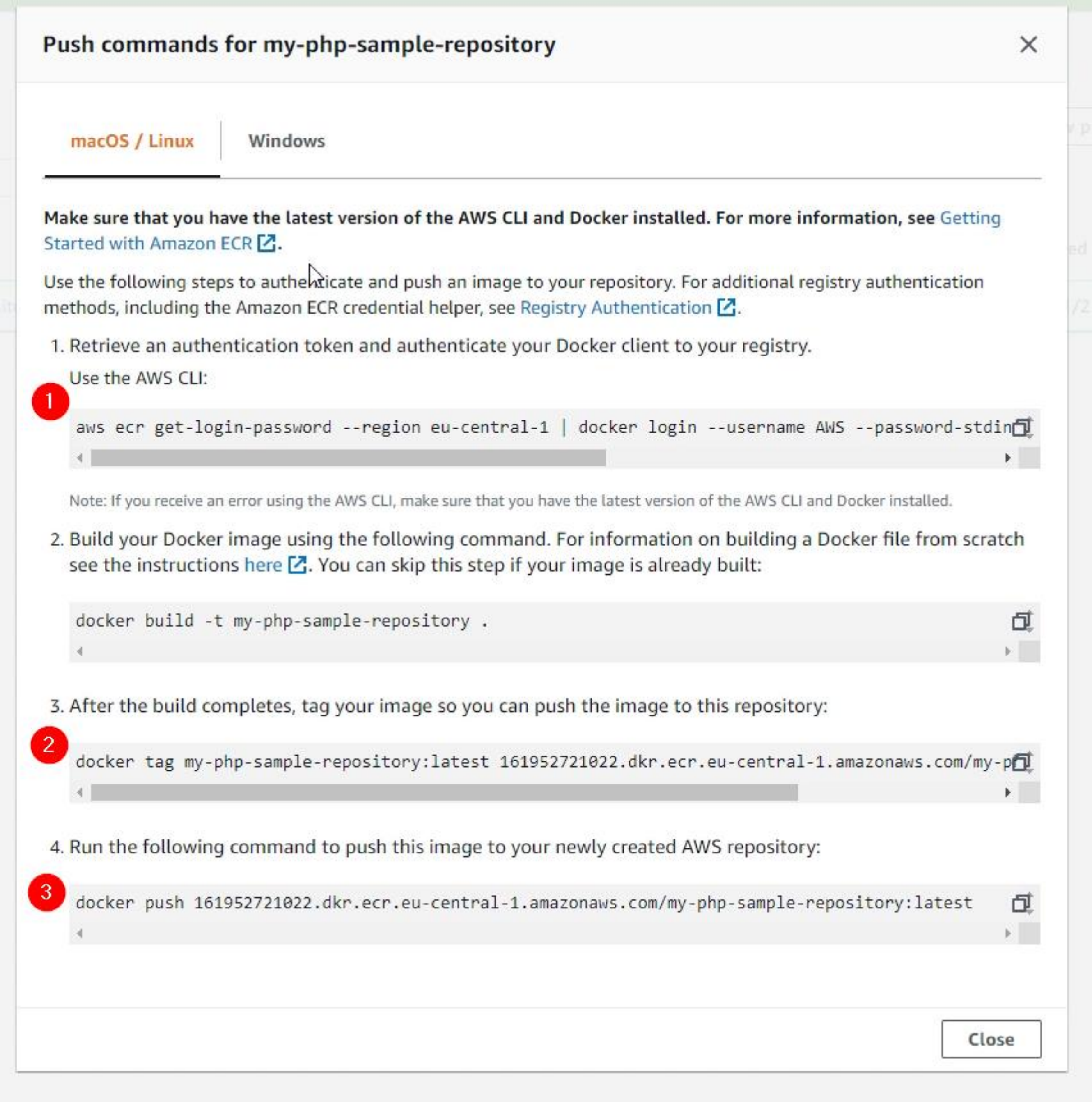
View the Push-Commands:



Repositories (1 of 1)

	Repository name	URI	Created at	Tag immut
●	my-php-sample-repository	161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-php-sample-repository	04/01/20, 02:22:35 PM	Disabled

First you need to Login, but we already have our image built in the local repository on your Host Computer. That means all we have to do is tagging and uploading:



Push commands for my-php-sample-repository

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 eu-central-1 | docker login --username AWS --password-stdin
```

   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 my-php-sample-repository .
```

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

```
docker tag my-php-sample-repository:latest 161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-p
```

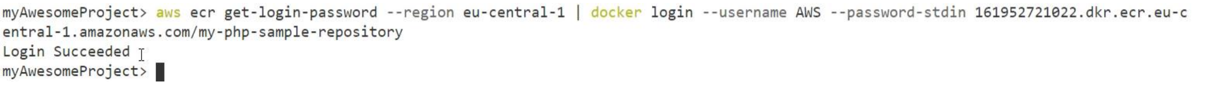
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-php-sample-repository:latest
```

Copy the commands and run them:

Login to ECR

Copy the first Command which logs into ECR for your Docker Desktop:



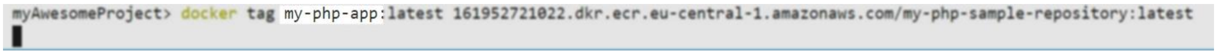
```
myAwesomeProject> aws ecr get-login-password --region eu-central-1 | docker login --username AWS --password-stdin 161952721022.dkr.ecr.eu-c
entral-1.amazonaws.com/my-php-sample-repository
Login Succeeded
myAwesomeProject>
```

Tag Your Image

For Docker to know which Image to put in which remote Repository you need to tag it the right way.

docker tag [local-image-name]:latest [ecr-repsitory-name]:latest

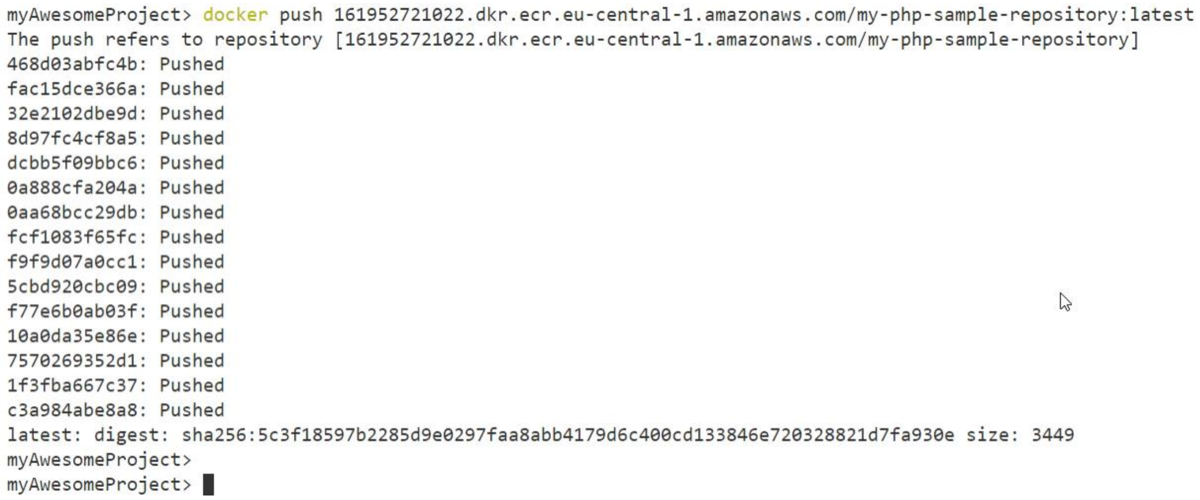
Copy the command where is the red (2) but be aware that you have to change the image name most likely:



```
myAwesomeProject> docker tag my-php-app:latest 161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-php-sample-repository:latest
```

Upload your Image

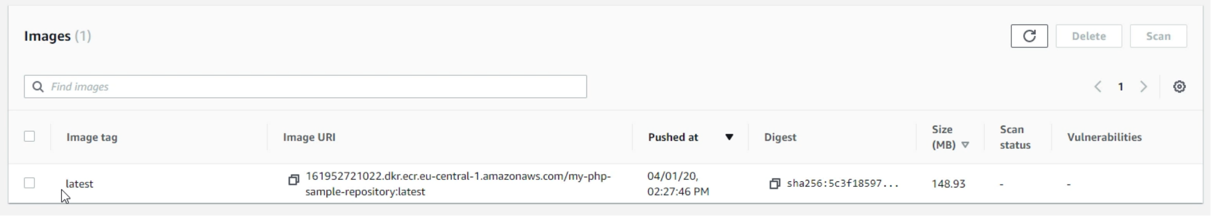
Then simply push your image to ECR by copying the command where is the red (3) in the image above to push the image:



```
myAwesomeProject> docker push 161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-php-sample-repository:latest
The push refers to repository [161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-php-sample-repository]
468d03abfc4b: Pushed
fac15dce366a: Pushed
32e2102dbe9d: Pushed
8d97fc4cf8a5: Pushed
dcbb5f09bbc6: Pushed
0a888cfa204a: Pushed
0aa68bcc29db: Pushed
fcf1083f65fc: Pushed
f9f9d07a0cc1: Pushed
5cbd920cbc09: Pushed
f77e6b0ab03f: Pushed
10a0da35e86e: Pushed
7570269352d1: Pushed
1f3fba667c37: Pushed
c3a984abe8a8: Pushed
latest: digest: sha256:5c3f18597b2285d9e0297faa8abb4179d6c400cd133846e720328821d7fa930e size: 3449
myAwesomeProject>
myAwesomeProject> █
```

Check if your Image is in ECR

Just reload the ECR Dashboard:



□	Image tag	Image URI	Pushed at	Digest	Size (MB)	Scan status	Vulnerabilities
□	latest	161952721022.dkr.ecr.eu-central-1.amazonaws.com/my-php-sample-repository:latest	04/01/20, 02:27:46 PM	sha256:5c3f18597...	148.93	-	-

Lab End