# AWS ECR - Elastic Container Registry

## What is Elastic Container Registry (ECR)

- Elastic Container Registry (ECR) is a fully-managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images.
- ECR is integrated with Elastic Kubernetes Service (EKS), simplifying our development to production workflow.
- ECR eliminates the need to operate our own container repositories or worry about scaling the underlying infrastructure.
- ECR hosts our images in a highly available and scalable architecture, allowing us to reliably deploy containers for our applications.
- Integration with AWS Identity and Access Management (IAM) provides resource-level control of each repository.
- With Amazon ECR, there are no upfront fees or commitments. We pay only for the amount of data you store in your repositories and data transferred to the Internet.

## Benefits of Elastic Container Registry (ECR)

- Fully managed
- Secure
- Highly Available
- Scan container for security or vulnerability or of the box

## Step-01: What are we going to learn?

- We are going to build a Docker image
- Push to ECR Repository
- And run the image to test the application

## Step-02: ECR Terminology

- **Registry:** An ECR registry is provided to each AWS account; we can create image repositories in our registry and store images in them.
- **Repository:** An ECR image repository contains our Docker images.
- **Repository policy:** We can control access to our repositories and the images within them with repository policies.
- **Authorization token:** `With docker, we use username and password through docker login`. Our Docker client must authenticate to Amazon ECR registries as an AWS user before it can push and pull images. The AWS CLI get-login command provides us with authentication credentials to pass to Docker.
- **Image:** We can push and pull container images to our repositories.

## Step-03: Pre-requisites

- Install required CLI Version 2 software on your local desktop
- Docker
- Create a user to access AWS ECR
- Configure AWS CLI with Authorization

```
aws configure
AWS Access Key ID: ****
AWS Secret Access Key: ****
Default region name: us-east-1
Default output format: json
```

**Test the connection**

```
aws s3 ls
```

Step-04: Create ECR Repository

- Create a simple ECR repository via AWS Console
  - Repository Name: aws-ecr-docker
  - Tag Immutability: Enable
  - Scan on Push: Enable (to scan docker image for security)
- Explore ECR console.

**Create ECR Repository using AWS CLI**

```
aws ecr create-repository --repository-name aws-ecr-docker --region us-
east-1
aws ecr create-repository --repository-name <your-repo-name> --region
<your-region>

788210522308.dkr.ecr.us-east-1.amazonaws.com/aws-ecr-docker
```

Step-05: Create Docker Image locally and test it

- Create docker image locally
- Run it locally and test

**Build Docker Image**

```
vim Dockerfile

FROM httpd
LABEL maintainer="Tia M"
RUN apt-get -y update && \
    apt-get -y install wget && \
    apt-get install -y apt-utils && \
    apt-get -y install unzip

WORKDIR /usr/local/apache2/htdocs/

RUN rm -rf * && \
    wget https://linux-devops-course.s3.amazonaws.com/WEB+SIDE+HTML
/covid19.zip && \
    unzip covid19.zip && \
    cp -R covid19/* .

CMD ["httpd-foreground"]
EXPOSE 80
docker build -t <ECR-REPOSITORY-URI>:<TAG> .

docker build -t 788210522308.dkr.ecr.us-east-1.amazonaws.com/aws-ecr-
docker:covid19 .
```

**Run Docker Image locally & Test**

```
docker run --name <name-of-container> -p 80:80 -d <ECR-REPOSITORY-URI>:
<TAG>

docker run --name covid19 -p 8010:80 -d 788210522308.dkr.ecr.us-east-1.
amazonaws.com/aws-ecr-docker:covid19
```

**Access Application locally**

```
http://<IP>:8010/
http://10.0.0.94:8010/
```

Get Login Password

```
aws ecr get-login-password --region <your-region> | docker login --
username AWS --password-stdin <ECR-REPOSITORY-URI>

aws ecr get-login-password --region us-east-1 | docker login --username
AWS --password-stdin 788210522308.dkr.ecr.us-east-1.amazonaws.com/aws-
ecr-docker
```

Push the Docker Image to ECR

```
docker push <ECR-REPOSITORY-URI>:<TAG>

docker push 788210522308.dkr.ecr.us-east-1.amazonaws.com/aws-ecr-docker:
covid19
```

- Verify the newly pushed docker image on AWS ECR.
- Verify the vulnerability scan results.

Clean up, pull image from ECR and test it

1. Clean up

```
for cont in $(docker ps -a |awk '{print$1}'); do docker stop $cont;
docker rm -f $cont; done

for im in $(docker images |awk '{print$3}'); do docker rmi -f $im; done
```

1. Login first

```
aws ecr get-login-password --region us-east-1 | docker login --username
AWS --password-stdin 788210522308.dkr.ecr.us-east-1.amazonaws.com/aws-
ecr-docker
```

1. Pull the image down

```
docker pull 788210522308.dkr.ecr.us-east-1.amazonaws.com/aws-ecr-docker:
covid19
```

1. Run the image and test

```
docker run --name covid19 -p 8010:80 -d 788210522308.dkr.ecr.us-east-1.
amazonaws.com/aws-ecr-docker:covid19

http://<IP>:8010/
http://10.0.0.94:8010/
```