



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  
SINGAPORE

# Cloud Native Application - Containerization II

Cloud Infrastructure Engineering

**Nanyang Technological University  
& Skills Union - 2022/2023**

# Course Content

- Self Study Check In
- AWS ECR
- Benefits of AWS ECR
- Repeat Prev. Module Activity
- Activity - Create Repo in AWS ECR
- Activity - Push a Docker Image to ECR

**Q: You want to start using AWS Elastic Container Service(ECS), and you need a place to store your Docker Images, which of the following options would you choose?**

A - Use AWS DynamoDB to store Docker Images

B - Use AWS Relational Database Service(RDS) to store Images

C - Using EC2 Instance with EBS Volume attached

D - Using AWS Elastic Container Registry(ECR)

## **Q: What is the difference between a public and private repository in AWS Elastic Container Registry(ECR) ?**

- a) A public repository is accessible to anyone, while a private repository requires authentication to pull images
- b) A public repository is free to use, while a private repository requires payment
- c) A public repository is hosted on AWS servers, while a private repository is hosted on third-party servers

## Q: What is AWS ECS on Fargate?

- a) A managed service that helps run containerized applications on Amazon EC2 instances
- b) A managed service that helps run containerized applications on Amazon S3
- c) A managed service that helps run containerized applications on Amazon RDS
- d) A managed service that helps run containerized applications on a serverless compute engine, eliminating the need to manage servers.

## **Q: What is the main purpose of AWS Elastic Container Service(ECS) ?**

- a) To store and manage Docker Images
- b) To manage and scale containerized applications
- c) To provide load balancing for containerized applications
- d) To provide backup and recovery for containerized applications

# Activity

Instructor

- Ask to use AWS use single region for all learner for easier monitoring

# AWS ECR





# What is AWS ECR

Amazon Elastic Container Registry (ECR) is a Docker container registry that is managed fully by Amazon. It provides an effortless solution for developers to store, manage, and deploy Docker container images.

You don't need to operate your container repositories or concern yourself with scaling the underlying infrastructure, as Amazon ECR takes care of this. Your images are hosted in a highly available and scalable architecture, ensuring a dependable deployment of containers for your applications.

Amazon Identity and Access Management (IAM) integration allows you to have resource-level control over each repository. With Amazon ECR, there are no upfront costs or commitments. You will only be charged for the amount of data you store in your repositories and for data transferred to the Internet.

# AWS ECR Pricing

Source:

<https://aws.amazon.com/ecr/pricing/>

## Pricing details (beyond free tier limits)

Storage:

- Storage is \$0.10 per GB / month for data stored in private or public repositories.

Data transferred from private repositories:

Region: Asia Pacific (Singapore) ▾

	Pricing
<b>Data Transfer IN</b>	
All data transfer in	\$0.00 per GB
<b>Data Transfer OUT **</b>	
Next 9.999 TB / month	\$0.12 per GB
Next 40 TB / month	\$0.085 per GB
Next 100 TB / month	\$0.082 per GB
Greater than 150 TB / month	\$0.08 per GB

Data transferred from public repositories:

	Pricing
<b>Data Transfer IN</b>	
All data transfer in	\$0.00 per GB
<b>Data Transfer OUT ***</b>	
<b>Without using an AWS Account</b>	
500 GB / month	\$0.00 per GB
<b>When using an AWS Account</b>	
5 TB / month	\$0.00 per GB
Greater than 5 TB / month to non AWS Regions	\$0.09 per GB
Any amount to any AWS Region	\$0.00 per GB



# Benefits of AWS ECR

## **Fully managed**

Amazon ECR eliminates the need to operate and scale the infrastructure required to power your container registry. There is no software to install and manage or infrastructure to scale. Just push your container images to Amazon ECR and pull the images when you need to deploy.

## **Secure**

Amazon ECR transfers your container images over HTTPS and automatically encrypts your images at rest. You can configure policies to manage permissions and control access to your images using Amazon Identity and Access Management (IAM) users and roles without having to manage credentials directly on your EC2 instances.

## **Highly available**

Amazon ECR has a highly scalable, redundant, and durable architecture. Your container images are highly available and accessible, allowing you to reliably deploy new containers for your applications.

## **Simplified workflow**

Amazon ECR integrates with AWS ECS, AWS EKS, AWS Lambda and the Docker Command Line Interface (CLI), allowing you to simplify your development and production workflows. You can easily push your container images to Amazon ECR using the Docker CLI from your development machine, and Amazon ECS can pull them directly for production deployments.

# AWS ECR Repositories

**Private Repository:** A private repository does not offer content search capabilities and requires Amazon IAM-based authentication using AWS account credentials before allowing images to be pulled.

**Public Repository:** A public repository has descriptive content and allows anyone anywhere to pull images without needing an AWS account or using IAM credentials. Public repository images are also available in the Amazon ECR public gallery. ([ECR Public Gallery](#))

# Activity

Create new repository

**Break for 10-15 mins**

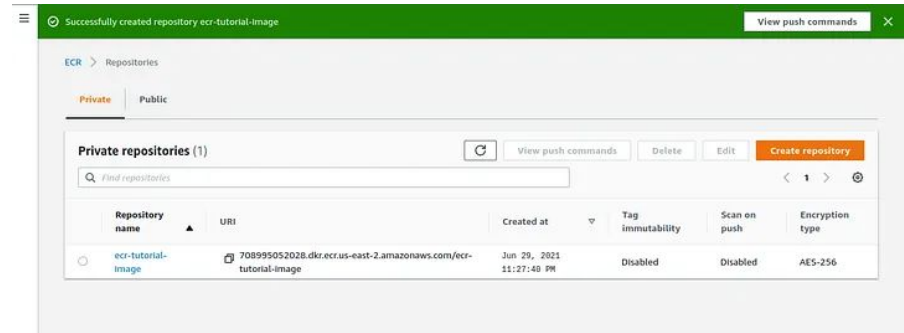
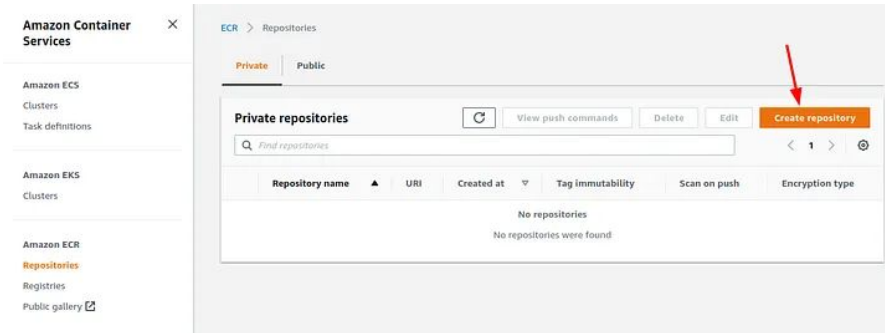


# AWS ECR



# Instructor - Activity - Create a Repo in ECR

Instructor demonstrate How to Create a Repo in ECR





# Learner - Activity - Create a Repo in ECR

Learner demonstrate How to Create a Private Repository in ECR

# Learner - Activity - Create a Repo in ECR

- \*) Clone -> `git clone git@github.com:su-ntu-ctp/6m-cloud-3.4-cloud-native-application-containerization-i.git`
- Copy file from "[hello-node-app](#)" to your own folder
- Run "npm install"
- Push everything to your github repo

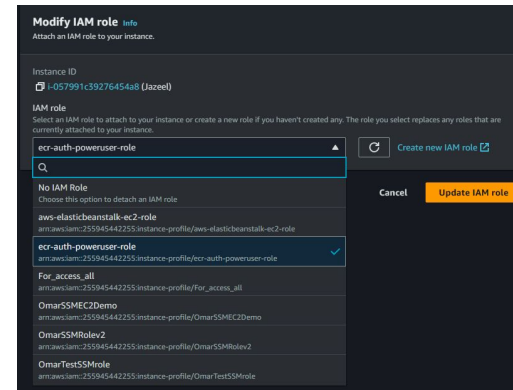
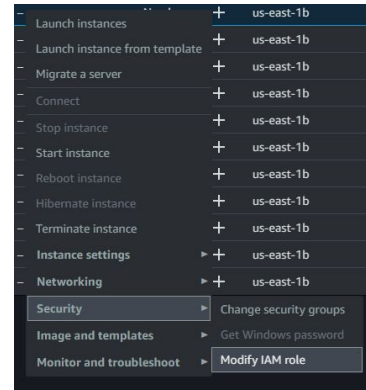
# Prerequisites - Setup

If using Windows or Mac:

- Run 'aws configure' on terminal / command prompt to configure your Access and Secret access keys with at least AmazonEC2ContainerRegistryPowerUser permissions

For using EC2:

- Attach IAM role with AmazonEC2ContainerRegistryPowerUser permissions to the EC2



# Authentication / Authorization to ECR

For Public Repositories(For pushing):

Permissions needed:

- `ecr-public:GetAuthorizationToken`
- `sts:getservicetoken`

Command:

```
aws ecr-public get-login-password --region us-east-1 |  
docker login --username AWS --password-stdin  
public.ecr.aws/<registry alias>
```

IAM Policy:

`AmazonElasticContainerRegistryPublicPowerUser`  
(AWS Managed Policy)

For Private Repositories:

Permissions needed:

- `ecr:GetAuthorizationToken`

Command:

```
aws ecr get-login-password --region ap-southeast-1 |  
docker login --username AWS --password-stdin <aws  
account number>.dkr.ecr.ap-southeast-1.amazonaws.com
```

IAM Policy: `AmazonEC2ContainerRegistryPowerUser`  
(AWS Managed Policy)

# Steps to building and pushing Image to Private ECR

Command to build an image

```
docker build -t <IMAGE_NAME>:<IMAGE_TAG> .
```

Command example with my image

```
docker build -t simple-app-image .
```

Command to tag an image

```
docker tag <IMAGE_NAME>:<IMAGE_TAG> <REPOSITORY_URI>:<IMAGE_TAG>
```

Command example with our image and repository

```
docker tag simple-app-image:latest <account no>.dkr.ecr.ap-southeast-1.amazonaws.com/<REPOSITORY_NAME>:latest
```

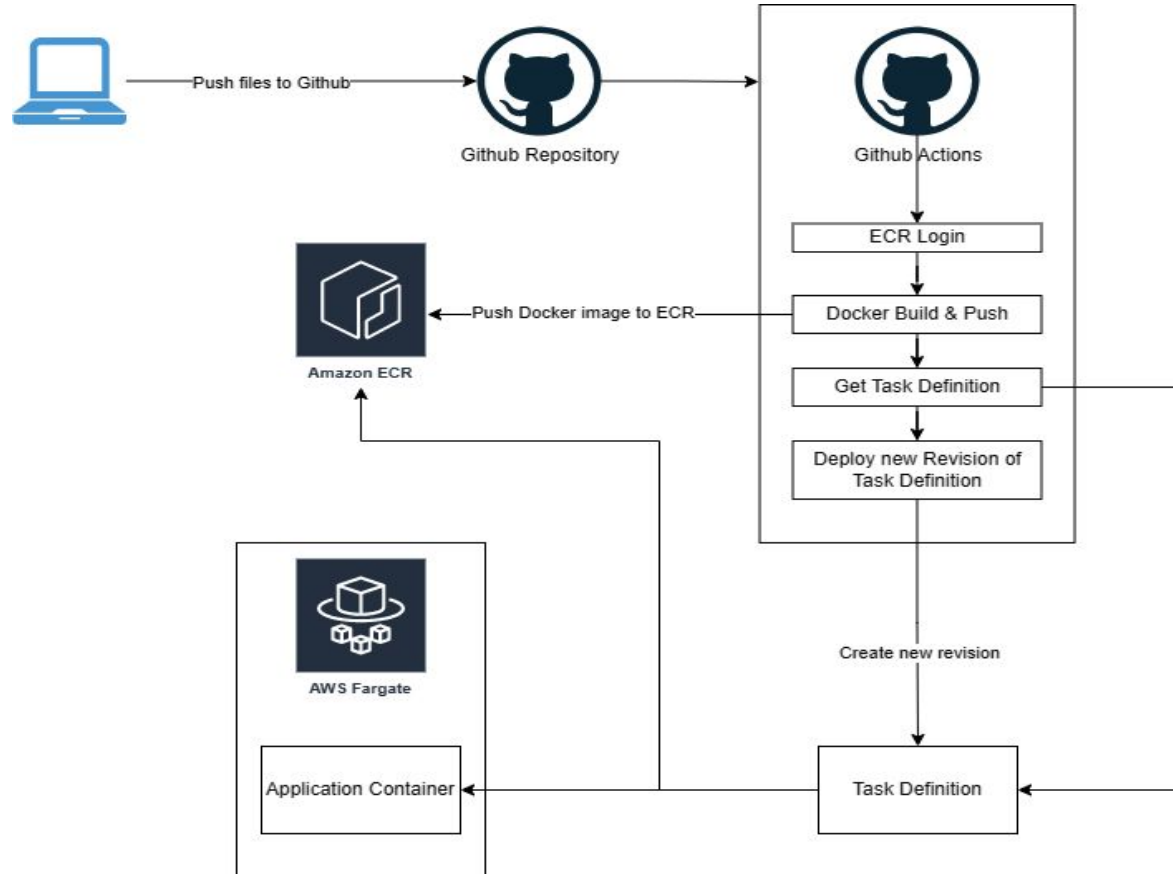
Command to push image:

```
docker push <account no>.dkr.ecr.ap-southeast-1.amazonaws.com/<REPOSITORY_NAME>:latest
```

# Learner - Activity - Push a Docker Image to ECR

Learner demonstrate How to Push a Docker Image to ECR

# Sample ECS CI/CD Demo



**ECR FAQ:** [Amazon ECR FAQs | Docker Container Registry | Amazon Web Services](#)





# Questions?



# Activity

Learner:

- Clean up AWS.
- Remove/delete/terminate all service/ resources that you created.

Instructor

- Clean up AWS.
- Remove/delete/terminate all service/ resources that you created.
- Check the AWS account after learner clean up.

**END**



# Instructor - Activity - Push a Docker Image to ECR

## Instructor demonstrate How Push a Docker Image to ECR

For Docker to push the image to ECR, first we have to authenticate our Docker credentials with AWS. We use the `get-login-password` command that retrieves and displays an authentication token using the `GetAuthorizationToken` API that we can use to authenticate to an Amazon ECR registry.

```
aws ecr get-login-password --region <region_name>
```

Use the same `region_name` that you used while creating a repo. Store the encrypted token somewhere for a moment.

Take a deep breath now. We need two things I told you to save somewhere. The first is the token I just mentioned and second is the repository URI from the previous step.

The following command pushes the local Docker file to the remote ECR repository. Depending on the image size, it will take some time to finish.

```
docker push <ecr-repo-uri>
```

# Activity

## **Repeat Prev. Module Activity**

Take 2-5 Mins for Learners create new repository.

Take 5-10 Mins for Learners create Dockerfile file with the sample-app.

# Steps to building and pushing Image to Private ECR

# Command to build an image

```
$docker image build -t <IMAGE_NAME>:<IMAGE_TAG> .
```

# Command example with my image

```
$docker image build -t simple-app-image .
```

# Command to tag an image

```
$docker image tag <IMAGE_NAME>:<IMAGE_TAG> <REPOSITORY_URI>:<IMAGE_TAG>
```

# Command example with our image and repository

```
$docker image tag simple-app-image:latest xxxxxxxxxxxx.dkr.ecr.us-east-2.amazonaws.com/ecr-tutorial-image:latest
```

## **Q: What is Docker?**

- A - A programming language
- B - A containerization platform
- C - A cloud computing service
- D - A data storage solution

## **Q: What is the main advantage of using Docker for containerization?**

- a) It allows applications to run on any operating system
- b) It provides built-in security features
- c) It allows for seamless integration with other cloud services
- d) It provides a highly scalable infrastructure



## **Q: What is AWS ECR?**

- a) A container orchestration tool
- b) A managed registry service for Docker container images
- c) An infrastructure-as-code tool for managing AWS resources