

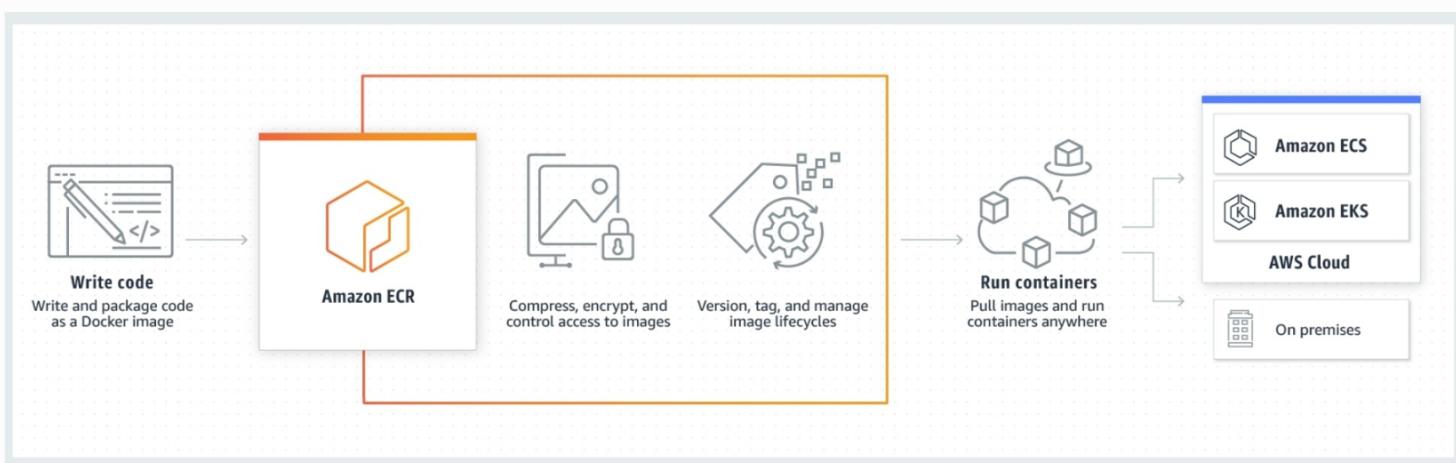
# AWS - 105

## ECR - Elastic Container Registry

Amazon Elastic Container Registry (ECR) is a fully managed container registry that makes it easy to store, manage, share, and deploy your container images and artifacts anywhere

Amazon ECR works with Amazon Elastic Kubernetes Service (EKS), Amazon Elastic Container Service (ECS), and AWS Lambda, simplifying your development to production workflow

### How it works



You can use your preferred CLI to push, pull, and manage Docker images, Open Container Initiative (OCI) images, and OCI compatible artifacts.

### Components of Amazon ECR

## Registry

An Amazon ECR registry is provided to each AWS account; you can create image repositories in your registry and store images in them. For more information, see [Amazon ECR private registries](#).

## Authorization token

Your client must authenticate to Amazon ECR registries as an AWS user before it can push and pull images. For more information, see [Private registry authentication](#).

## Repository

An Amazon ECR image repository contains your Docker images, Open Container Initiative (OCI) images, and OCI compatible artifacts. For more information, see [Amazon ECR private repositories](#).

## Repository policy

You can control access to your repositories and the images within them with repository policies. For more information, see [Repository policies](#).

## Image

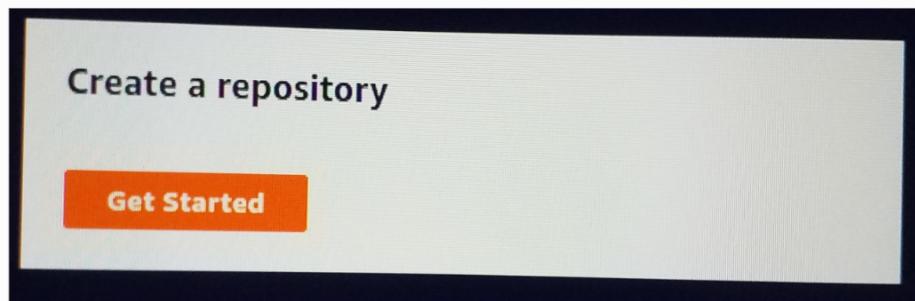
You can push and pull container images to your repositories. You can use these images locally on your development system, or you can use them in Amazon ECS task definitions and Amazon EKS pod specifications. For more information, see [Using Amazon ECR images with Amazon ECS](#) and [Using Amazon ECR Images with Amazon EKS](#).

Ques:

NOTE: To perform the below steps, please install AWS CLI and configure it first

→ Creating an AWS ECR repository and pushing docker image

- 1) Go to AWS console and search ECR
- 2) Click 'Get Started' under 'Create a repository'



- 3) Enter the repository name : <your choice>
- 4) Click 'Create Repository'
- 5) One repository is created follow as shown below:

Step 2: Click to view push commands

Repository name	URI	Created at	Tag immutability	Scan on push
spring_boot_kubernetes	023194539562.dkr.ecr.ap-south-1.amazonaws.com/spring_boot_kubernetes	12/18/20, 02:58:36 PM	Disabled	Disabled

Step 1: Select the repository

- 6) Read and follow the shown command, these command mainly has 3 things :
- a) Authentication of docker client to your registry
  - b) Tag the docker image to push
  - c) Run command to push the docker image

