# 6 Containers on AWS

Q What is a docker?

→ A software development platform deploy apps. Apps are packaged into containers that can run on any OS.

→ good for microservice-architecture & Lift-and-Shift apps from On-premises to AWS cloud.

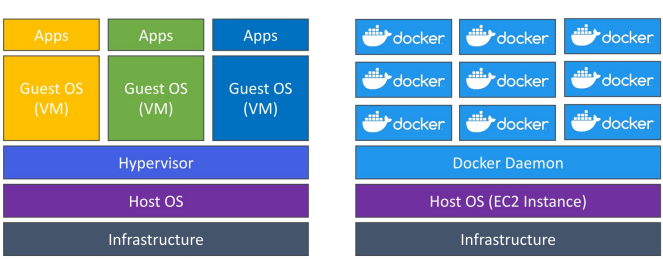
→ can run a docker container on any OS and more than one instance. (app on)

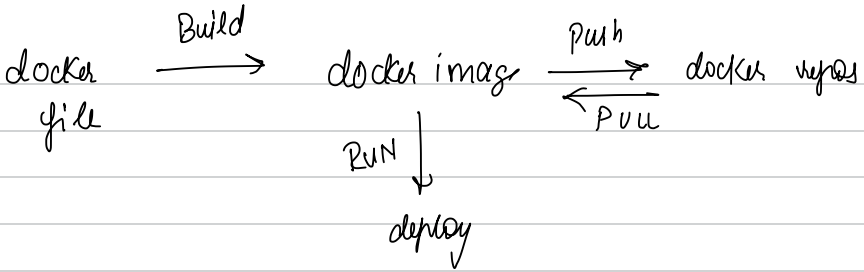
docker images can be stored in

- Docker hub
  { Public repo }
  can find base images of many Technologies & OS
- Images on ECR
  { Public (ECR public Gallery)
  Private }

## Docker vs. Virtual Machines

- Docker is "sort of" a virtualization technology, but not exactly
- Resources are shared with the host => many containers on one server






## Docker Containers Management on AWS

1. Amazon Elastic Container Service (Amazon ECS)
   - Amazon's own container platform
2. Amazon Elastic Kubernetes Service (Amazon EKS)
   - Amazon's managed Kubernetes (open source)
3. AWS Fargate
   - Amazon's own Serverless container platform
   - Works with ECS and with EKS
4. Amazon ECR:
   - Store container images

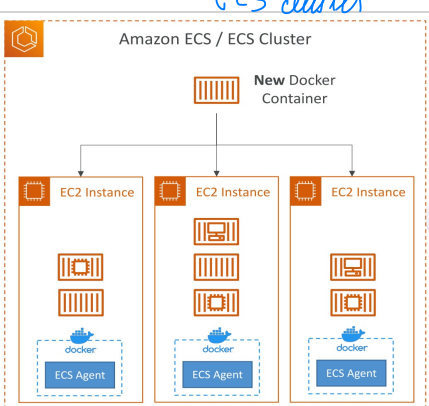

## 1 Amazon ECS

( Elastic Containers service )

- Launch dockers on AWS = Launch ECS tasks on ECS clusters.

(a) EC2 launch type → provision & maintain EC2 instances.

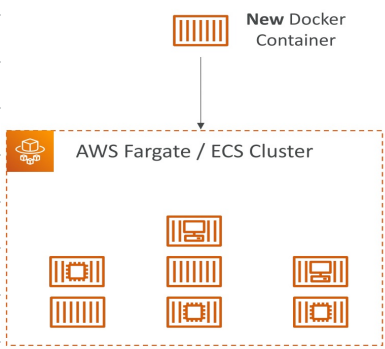
- Each EC2 instance must run "ECS Agent" to register into ECS cluster
- Once done AWS takes care of starting / stoping & distributing the containers.



(5) Fargate launch type → no need to provision EC2 instances (& the underlying infrastructure is on EC2 but no need to provision ourselves)

i.e. serverless !

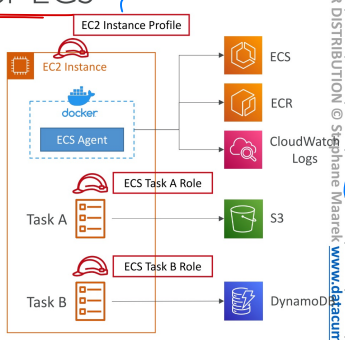
- create tasks & AWS just runs them based on the CPU / RAM needed
- it will scale automatically.




# Amazon ECS – IAM Roles for ECS

an IAM policy!

- EC2 Instance Profile (EC2 Launch Type only):
  - Used by the ECS agent
  - Makes API calls to ECS service
  - Send container logs to CloudWatch Logs
  - Pull Docker image from ECR
  - Reference sensitive data in Secrets Manager or SSM Parameter Store
- ECS Task Role:
  - Allows each task to have a specific role
  - Use different roles for the different ECS Services you run
  - Task Role is defined in the task definition



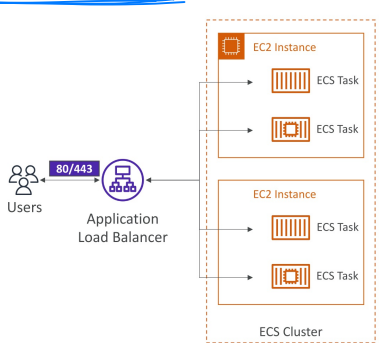

(1) EC2 instance profile role {instance level}

(2) EC2 task role {task level}



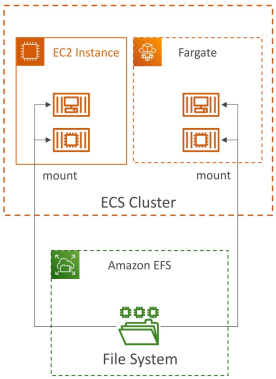
# Amazon ECS – Load Balancer Integrations

- Application Load Balancer supported and works for most use cases
- Network Load Balancer recommended only for high throughput / high performance use cases, or to pair it with AWS Private Link
- Classic Load Balancer supported but not recommended (no advanced features – no Fargate)



# Amazon ECS – Data Volumes (EFS)

- Mount EFS file systems onto ECS tasks
- Works for both EC2 and Fargate launch types
- Tasks running in any AZ will share the same data in the EFS file system
- Fargate + EFS = Serverless (ultimate)
- Use cases: persistent multi-AZ shared storage for your containers
- Note:
  - Amazon S3 cannot be mounted as a file system




(1) first create a ECS cluster, use the VPC wizard & this will create a ECS cluster if EC2 launch type then an Auto Scaling Group is also to be created.

Ephemeral storage ?

(2) To create a task go to task definition & infrastructure requirements

task role is for containers to interact with other services of AWS.

# ECS Service Auto Scaling

- Automatically increase/decrease the desired number of ECS tasks
- Amazon ECS Auto Scaling uses **AWS Application Auto Scaling** (metrics)
  - ECS Service Average CPU Utilization
  - ECS Service Average Memory Utilization - Scale on RAM
  - ALB Request Count Per Target – metric coming from the ALB
- **Target Tracking** – scale based on target value for a specific CloudWatch metric
- **Step Scaling** – scale based on a specified CloudWatch Alarm
- **Scheduled Scaling** – scale based on a specified date/time (predictable changes)

(scaling methods)

- ECS Service Auto Scaling (task level) ≠ EC2 Auto Scaling (EC2 instance level)
- Fargate Auto Scaling is much easier to setup (because **Serverless**)

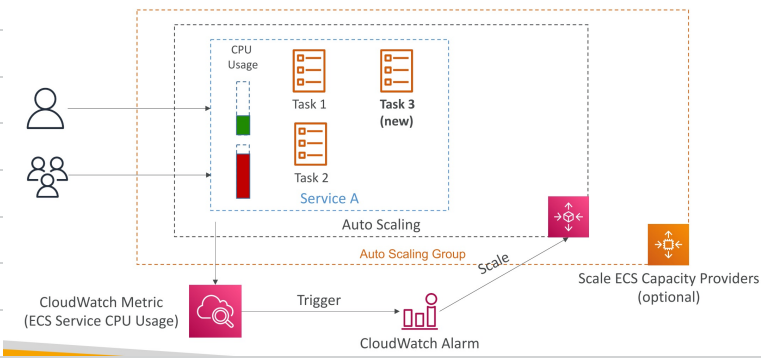
# EC2 Launch Type – Auto Scaling EC2 Instances

- Accommodate ECS Service Scaling by adding underlying EC2 Instances

- Auto Scaling Group Scaling
  - Scale your ASG based on CPU Utilization
  - Add EC2 instances over time

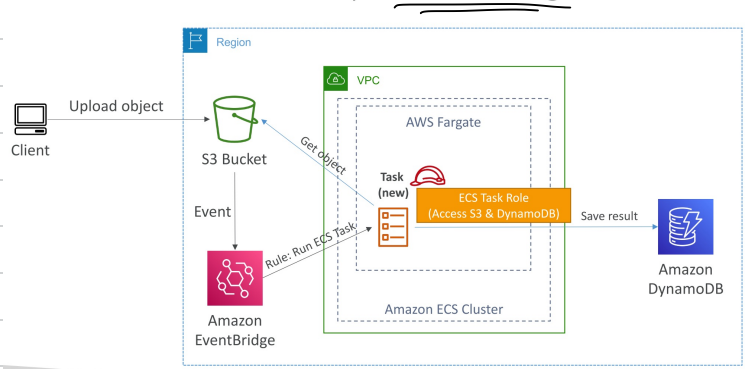
better!

- ECS Cluster Capacity Provider
  - Used to automatically provision and scale the infrastructure for your ECS Tasks
  - Capacity Provider paired with an Auto Scaling Group
  - Add EC2 Instances when you're missing capacity (CPU, RAM…)

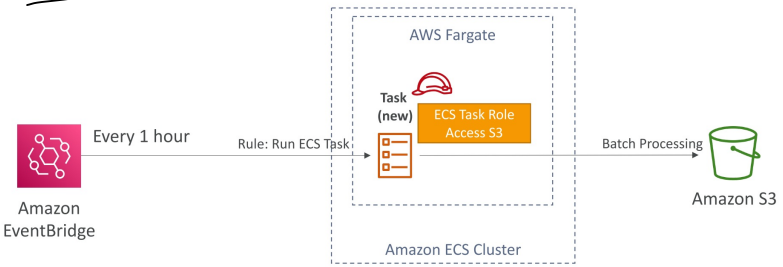



* Serverless solution to process data & add them to a database.

## ECS tasks invoked by Event Bridge

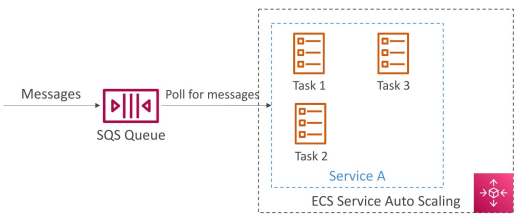


* Example 2



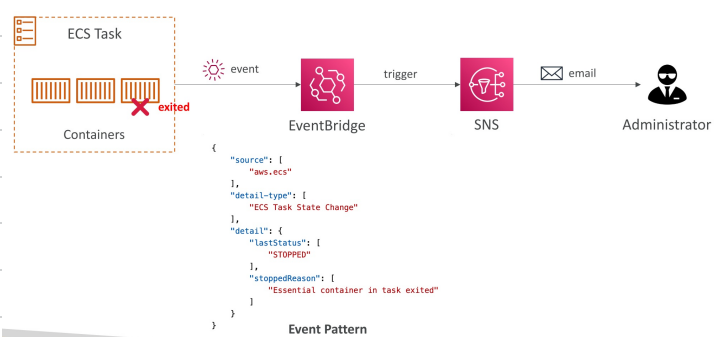

* Example 3

## ECS – SQS Queue Example




* Example 3

## ECS – Intercept Stopped Tasks using EventBridge

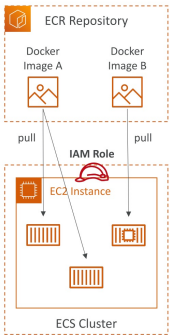



hence Event bridge allows us to understand the lifecycle of ECS containers

# Amazon ECR

storing docker images

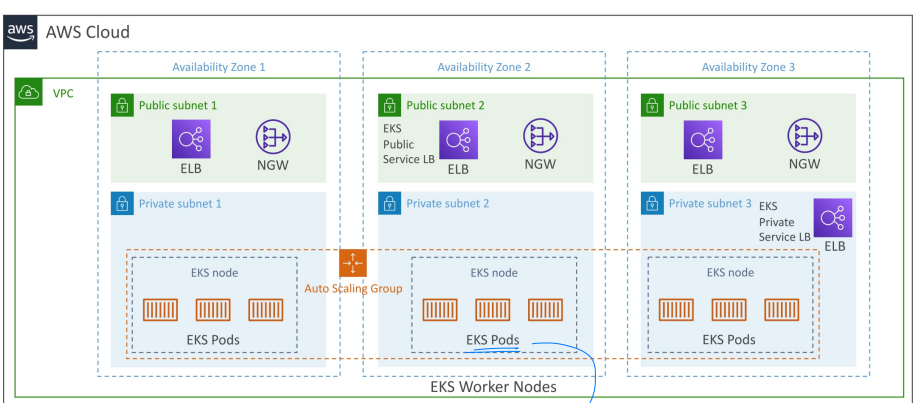
- ECR = Elastic Container Registry
- Store and manage Docker images on AWS
- Private and Public repository (Amazon ECR Public Gallery https://gallery.ecr.aws)
- Fully integrated with ECS, backed by Amazon S3
- Access is controlled through IAM (permission errors => policy)
- Supports image vulnerability scanning, versioning, image tags, image lifecycle, …




Any permission denied is due to IAM.

# Amazon EKS Overview

- Amazon EKS = Amazon Elastic Kubernetes Service
- It is a way to launch managed Kubernetes clusters on AWS
- Kubernetes is an open-source system for automatic deployment, scaling and management of containerized (usually Docker) application
- It's an alternative to ECS, similar goal but different API
- EKS supports EC2 if you want to deploy worker nodes or Fargate to deploy serverless containers
- Use case: if your company is already using Kubernetes on-premises or in another cloud, and wants to migrate to AWS using Kubernetes
- Kubernetes is cloud-agnostic (can be used in any cloud – Azure, GCP…)
- For multiple regions, deploy one EKS cluster per region
- Collect logs and metrics using CloudWatch Container Insights




similar to tasks

# Amazon EKS – Node Types

- Managed Node Groups
  - Creates and manages Nodes (EC2 instances) for you
  - Nodes are part of an ASG managed by EKS
  - Supports On-Demand or Spot Instances
- Self-Managed Nodes
  - Nodes created by you and registered to the EKS cluster and managed by an ASG
  - You can use prebuilt AMI - Amazon EKS Optimized AMI
  - Supports On-Demand or Spot Instances
- AWS Fargate
  - No maintenance required; no nodes managed

# Amazon EKS – Data Volumes

- Need to specify StorageClass manifest on your EKS cluster
- Leverages a Container Storage Interface (CSI) compliant driver

- Support for…
  - Amazon EBS
  - Amazon EFS (works with Fargate)
  - Amazon FSx for Lustre
  - Amazon FSx for NetApp ONTAP


# AWS App Runner

- Fully managed service that makes it easy to deploy web applications and APIs at scale
- No infrastructure experience required
- Start with your source code or container image
- Automatically builds and deploy the web app
- Automatic scaling, highly available, load balancer, encryption
- VPC access support
- Connect to database, cache, and message queue services
- Use cases: web apps, APIs, microservices, rapid production deployments