**Elastic Container Service (ECS)**

* + - highly scalable, high performance container management service that supports Docker containers and allows you to easily run applications on a managed cluster of EC2 instances
    - Eliminates the need for you to install, operate, and scale your own cluster management infrastructure
    - Using API calls you can launch and stop container-enabled applications, query the complete state of clusters, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes and IAM roles
    - **No charge** for ECS
      * You do pay for the resources ECS deploys
    - You can assosciate a service on ECS to an ALB
    - Launch Types
      * 2 types
        + EC2
        + Fargate

This is used when you want to run ECS containers without having to manage servers or clusters of EC2 instances

**serverless**

You pay for the vCPU and memory allocated to the containers you run with Fargate

* + - ECS Terms

|  |  |
| --- | --- |
| Cluster | Logical Grouping of EC2 instances |
| Container instance | EC2 instance running the ECS agent |
| Task Definition | Blueprint that describes how a docker container should launch |
| Task | A running container using settings in a Task Definition |
| Service | Defines long running tasks - can control task count with ASG and attach an ELB |

* + - Images
      * Containers are created from a read-only template called an image which has the instructions for creating a Docker container
      * Images are built from a Dockerfile
      * Only Docker containers are currently supported
      * An image contains instructions for creating a Docker container
      * Images are stored in a registry such as DockerHuB
      * ECR(Elastic Container Registry) is AWS' version of DockerHub
    - Tasks
      * A task definition is required to run Docker containers in ECS
      * A task definition is a text file in JSON that describes one or more containers, up to a max of 10
      * Task definitions use Docker images to launch containers
      * You specify the number of tasks to run
      * Some of the parameters you can specify in a task definition include
        + How much CPU and memory to use with each container
        + Whether the containers are linked together in a task
        + The command the container should run when it is started
    - ECS Clusters
      * Logical grouping of container instances that you can place tasks on
      * A default cluster is created but you can then create multiple clusters to separate resources
      * Clusters can contain tasks using Fargate and EC2
      * Clusters are region specific
    - ECS Container Agent
      * Allows container instances to connect to the cluster
    - Service Auto Scaling
      * ECS can optionally be configured to use Service Auto Scaling to adjust the desired task count up or down automatically
      * ECS auto scaling follows two types of scaling policies
        + Target Tracking Polcies: Increase or decrease the number of tasks that your service runs based on a target value for a specific CloudWatch metric. This is similar to the way that your thermostat maintains the temperature of your home
        + Step Scaling Policies: Increase or decrease the number of tasks that your service runs in response to CW alarms. Step scaling is based on a set of scaling adjustments, known as step adjustments, which vary based on the size of the alarm breach
      * Cluster Auto Scaling
        + Uses ECS resource type called Capacity Provider
        + Capacity Provider can be associated with an EC2 ASG

Table

Description automatically generated

