Amazon Elastic Container Service

Docker Introduction

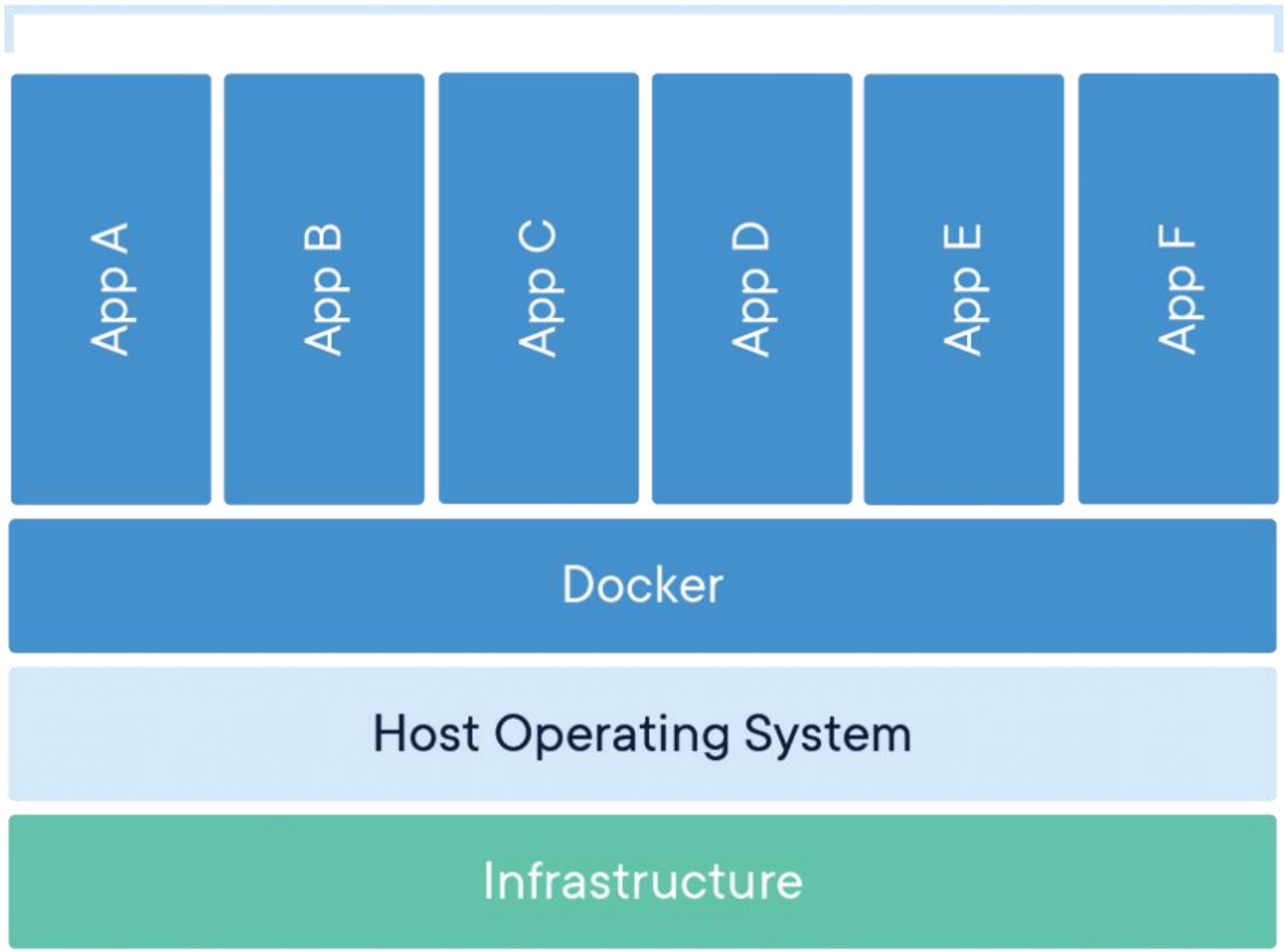
Usage of Docker

Docker architecture

Docker Introduction

- Docker is a software that performs OS level virtualization, also known as “containerization”
- Docker is not a virtual machine
- It is a popular tool to build, deploy and run applications using containers
- Apps are packaged in containers that can be run **on any OS**
- Dockers benefits include
  - Works in any machine
  - No compatibility issues
  - Predictable behavior
  - Deploy in seconds
  - Easier to maintain and deploy
  - Works with any language, any OS, any technology.

Containerized Applications



Usage of Docker

- Fast, consistent delivery of your applications
  - Containers are great for continuous integration and continuous delivery (CI/CD) workflows
- Responsive deployment and scaling
  - Docker’s container-based platform allows for highly portable workloads.
  - Docker containers can run on a
    - Developer’s local machine
    - On physical or virtual machines
    - On cloud providers
    - On a mixture of environments
- Running more workloads on the same hardware

Docker Architecture

- Docker uses a client-server architecture.
- The Docker client talks to the Docker daemon, which does building, running, and distributing your Docker containers.
- The Docker client and daemon can run on the same system, or you can connect a Docker client to a remote Docker daemon.
- The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface.

Client

```
docker build
```

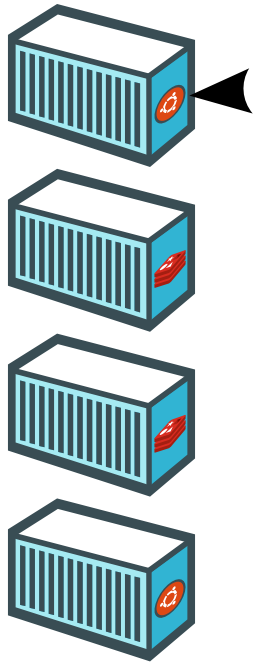
```
docker pull
```

```
docker run
```

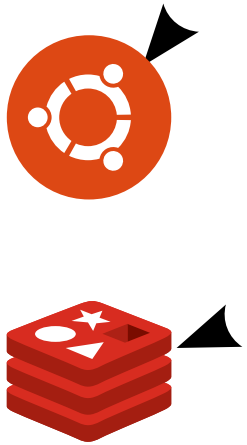
DOCKER_HOST

Docker daemon

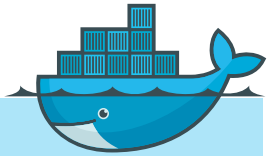
Containers



Images



Registry



Docker Terminologies

- The Docker Daemon
  - It listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes.
- The Docker Client
  - It is the primary way that many Docker users interact with Docker. The Docker client can communicate with more than one daemon.
- Docker Registries
  - It stores Docker images.
  - Docker Hub is a public registry that anyone can use.
  - Amazon ECR is another example

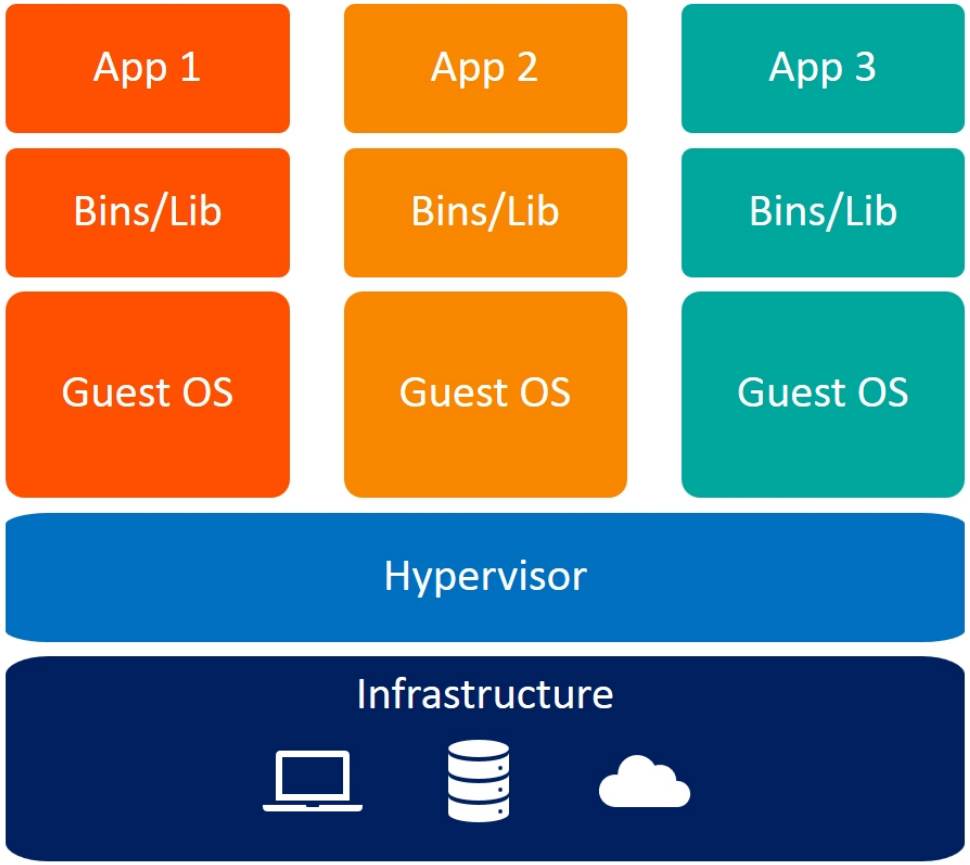
Docker Terminologies

- Docker File
  - A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.
- Docker Images
  - An image is a read-only template with instructions for creating a Docker container.
  - An image is based on another image, with some additional customization.
- Docker Container
  - A container is a runnable instance of an image.
  - You can create, start, stop, move, or delete a container using the Docker API or CLI.

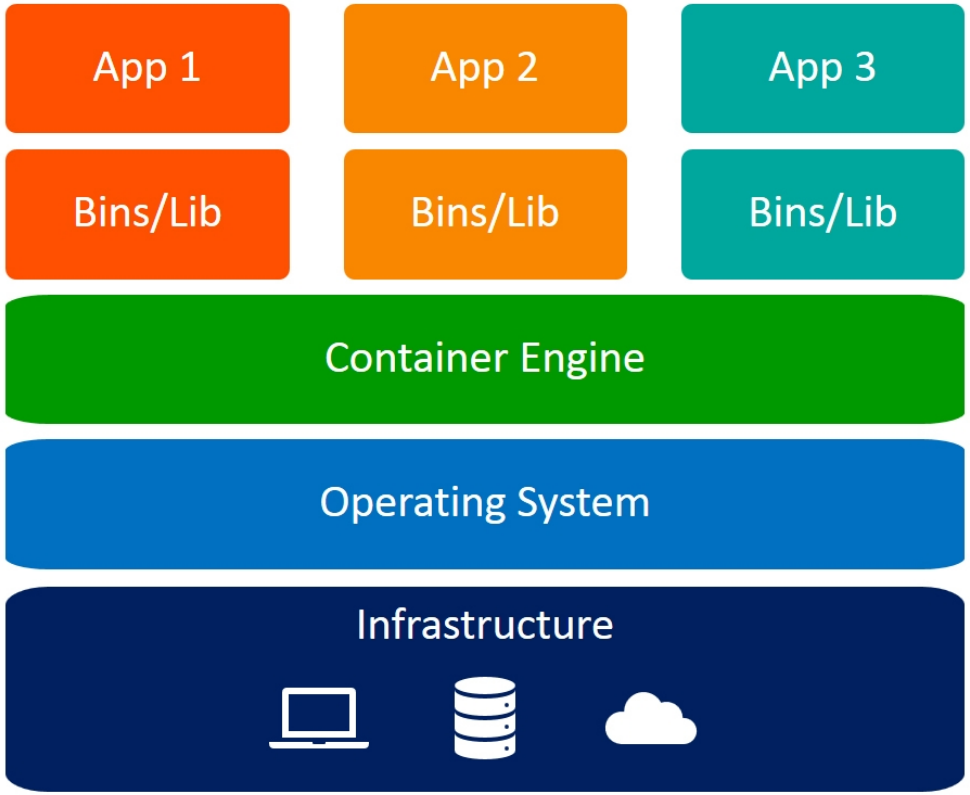
Docker Image Repository

- Docker images are stored in Docker Repositories
- Public Repo: Docker Hub https://hub.docker.com/
  - Find base images for many technologies or OS:
  - Ubuntu
  - MySQL
  - NodeJS, Java…
- Private Repo:
  - Amazon ECR (Elastic Container Registry)
  - JFrog
  - Nexus

Docker versus Virtual Machines



Virtual Machines



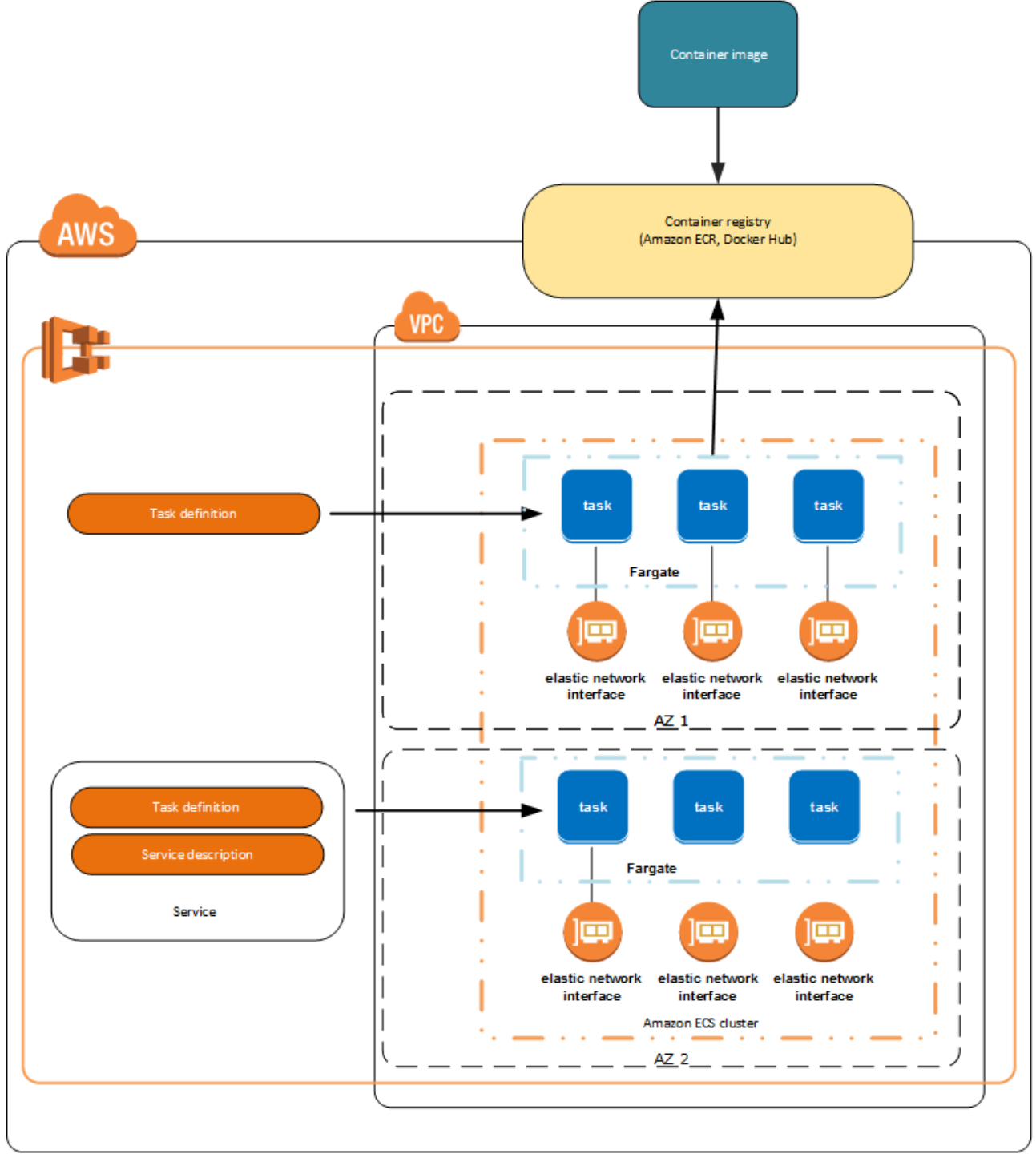
Containers

AWS Docker Containers Management

- Amazon Elastic Container Service (ECS): Amazon’s platform
- Fargate: Amazon’s own Serverless Platform
- Elastic Kubernetes Service: Amazon’s managed Kubernetes (open source)

Amazon Elastic Container Service

- Amazon Elastic Container Service is a highly scalable, fast container management service.
- Amazon ECS enables you to launch and stop your container-based applications by using simple API calls
- Regional service.
- Amazon ECS clusters created within a new or existing VPC.
- You can create task definitions that define which container images run across your clusters.
- Your task definitions are used to run tasks or create services.
- Container images are stored in container registries, for example, the Amazon Elastic Container Registry.


Container image
Container registry
(Amazon ECR, Docker Hub)
AWS
VPC
Task definition
task
task
task
Fargate
elastic network interface
elastic network interface
elastic network interface
AZ 1
Task definition
Service description
Service
task
task
task
Fargate
elastic network interface
elastic network interface
elastic network interface
Amazon ECS cluster
AZ 2

ECS Clusters Overview

- ECS Clusters are logical grouping of EC2 instances
- EC2 instances run the ECS agent (Docker container)
- The ECS agents registers the instance to the ECS cluster
- The EC2 instances run a special AMI, made specifically for ECS
- Clusters are Region-specific
- Type
  - Fargate
  - EC2 instance

ECS Task Definitions

- A task definition is required to run Docker containers in Amazon ECS.
- How to run a Docker Container
- The Docker image details provided
- Port Binding for Container and Host
- Amount of CPU and memory to be used
- The logging configuration to use for your tasks
- Environment variables
- Networking information
- IAM Role configuration

ECS Service

- ECS Services define the tasks details to be run and maintain a specified number of tasks
- Maintaining the desired number of tasks in your service
- If any task fail or stop for any reason, ECS service scheduler launches another tasks in the service
- Demo

ECR - Amazon Elastic Container Registry

- ECR is a private Docker image repository
- Amazon ECR provides API operations to create, monitor, and delete image repositories
- Access is controlled through IAM policy
- AWS CLI v1 login command
  - aws ecr get-login --no-include-email --region ap-south-1
- AWS CLI v2 login command
  - aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin test-acc.dkr.ecr.ap-south-1.amazonaws.com
- Docker Push & Pull:
  - docker push test-acc.dkr.ecr.ap-south-1.amazonaws.com/test:latest
  - docker pull test-acc.dkr.ecr.ap-south-1.amazonaws.com/test:latest