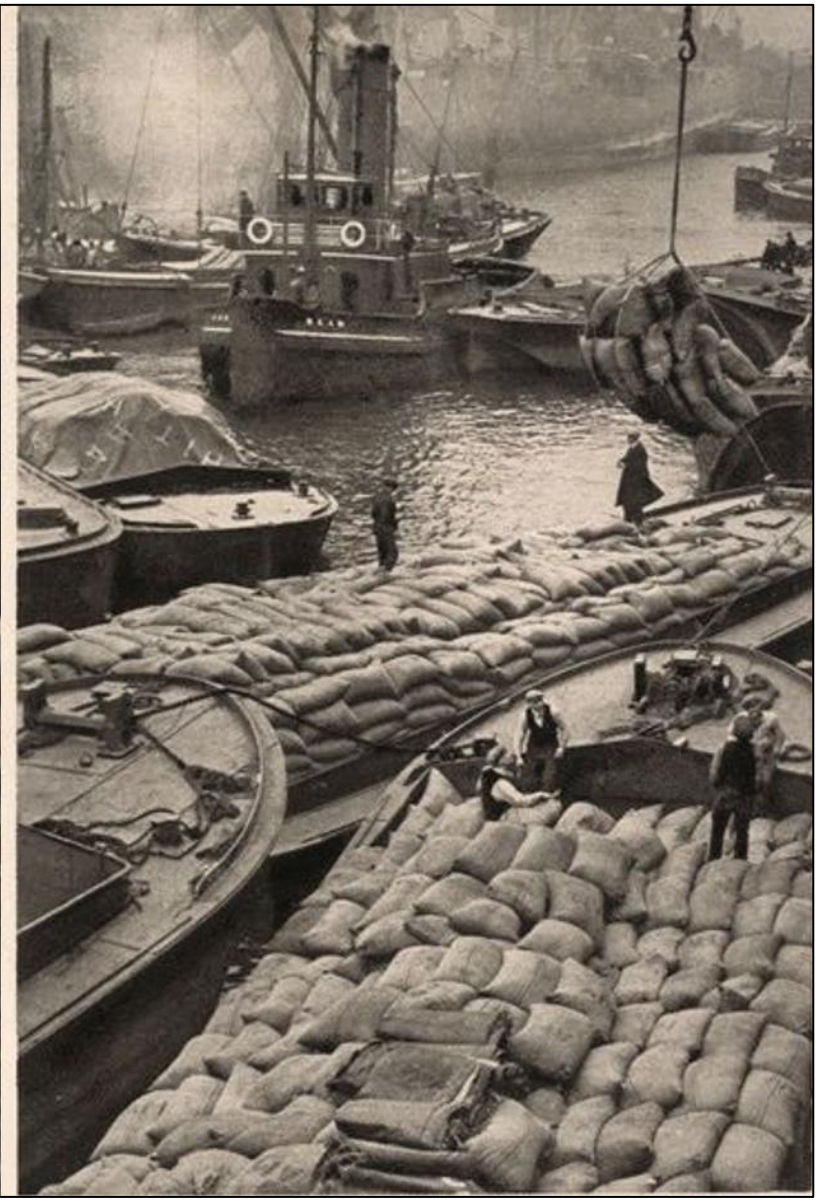
Containers in Cloud

Agenda

- Containers in Shipping
- Containers in Computing
- Containers in AWS

Containers in Shipping

Shipping before containers

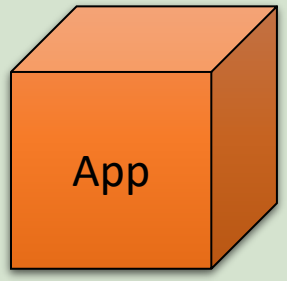
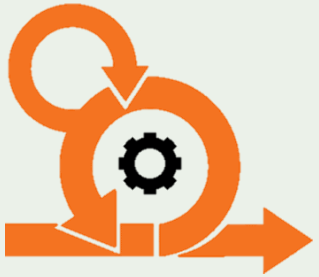
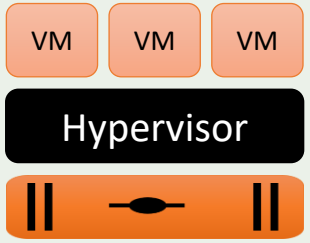
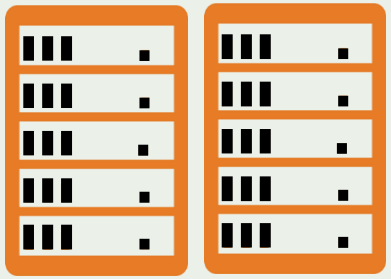
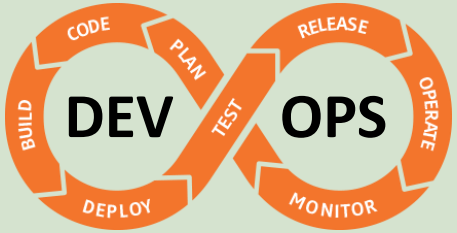
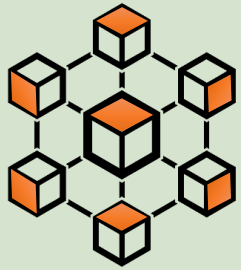
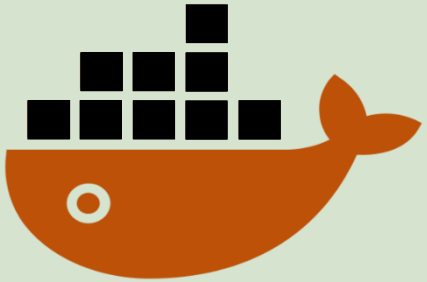


Solution – Intermodal shipping container

Shipping after containers

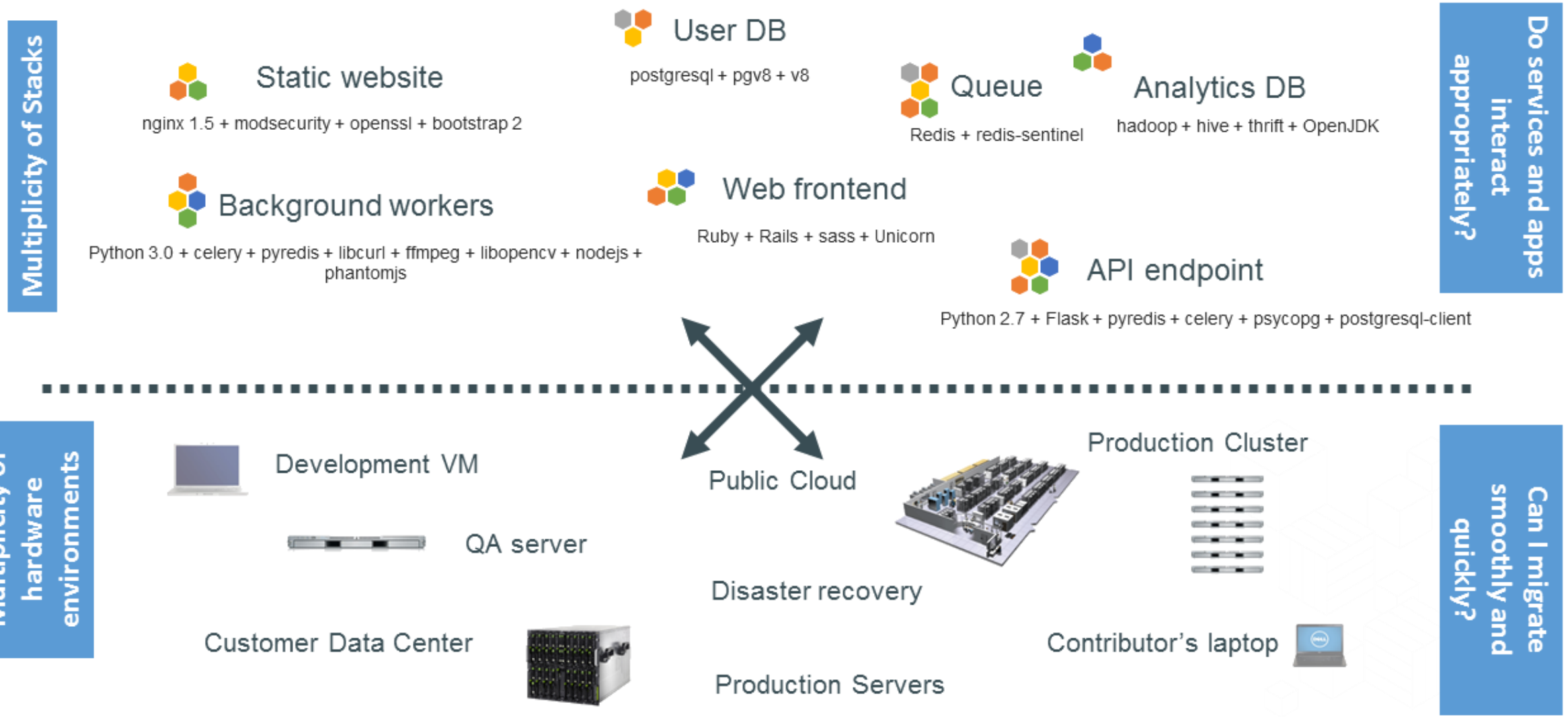
- Containers standardized the shipping industry

Containers in Computing

TimeLine	Development Process	Application Architecture	Deployment & Packaging	Application Hosting Infrastructure
1980 to 2000	 Waterfall	 Monolithic	 Physical Server	 Datacenter
2000 to 2010	 Agile	 N-Tier	 Virtual Servers	 Hosted
2010 to Current	 DevOps	 Microservices	 Containers	 Cloud

What is the Challenge?

- Maintaining multiple technology stacks across multiple environments is a nightmare



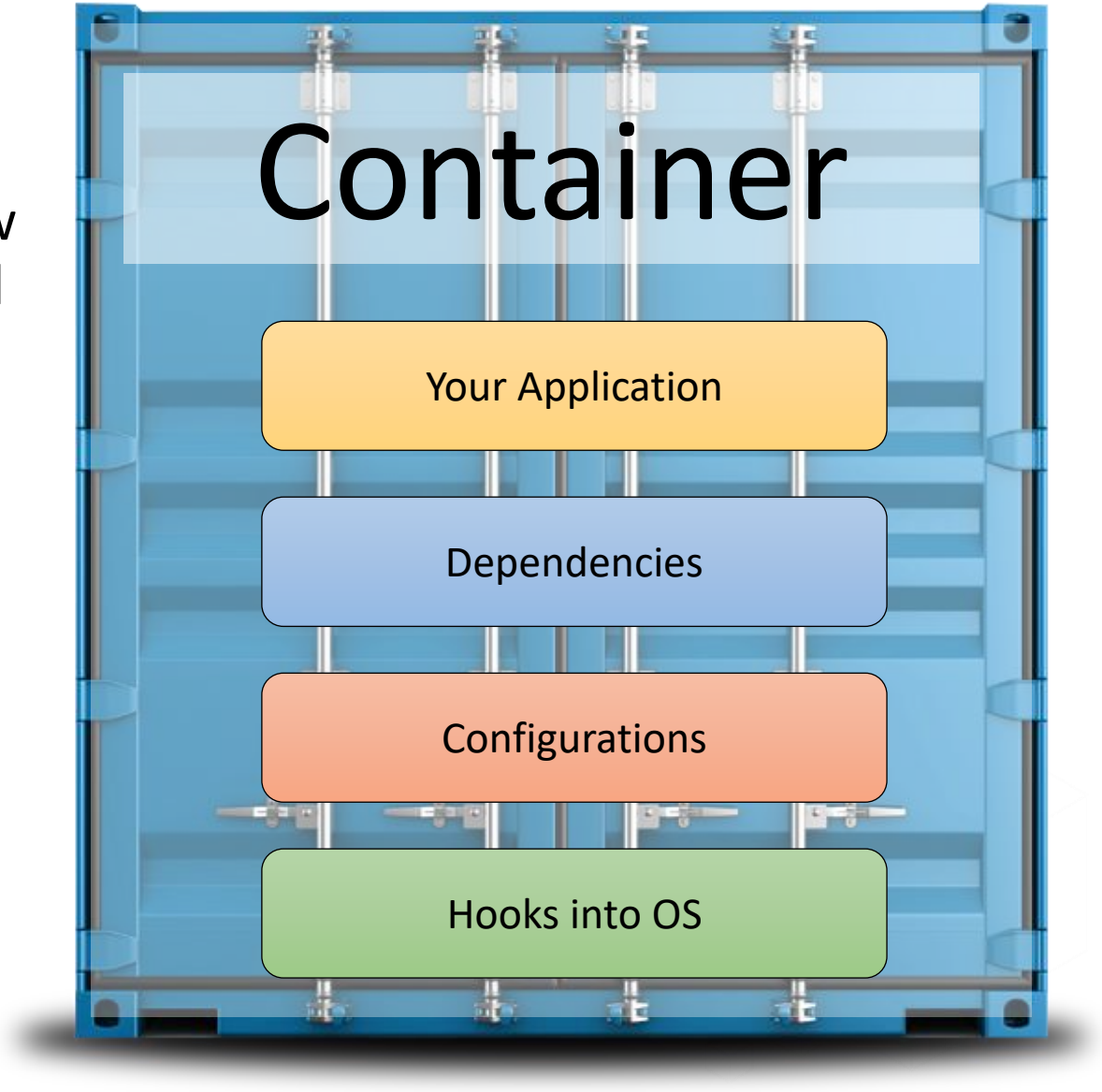
Matrix from Hell

- It is the challenge of packaging any application, irrespective of language/ frameworks / dependencies, so that it can run on any environment/cloud, irrespective of the underlying OS/hardware/infrastructure.

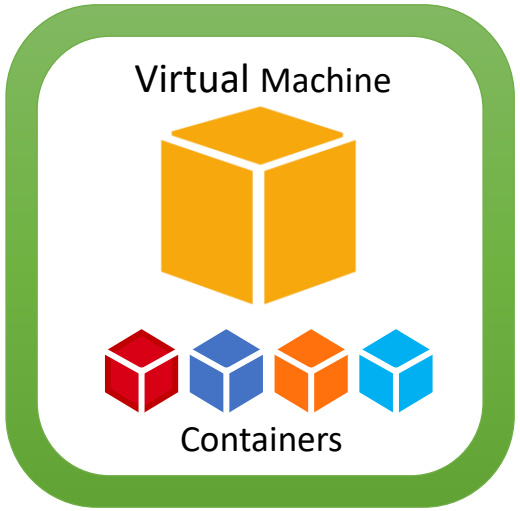
	Development VM	QA Server	Single Prod Server	Production Cluster	Public cloud	Developer's Laptop	Customer Servers
Static Website	?	?	?	?	?	?	?
Web Frontend	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

What is a Container?

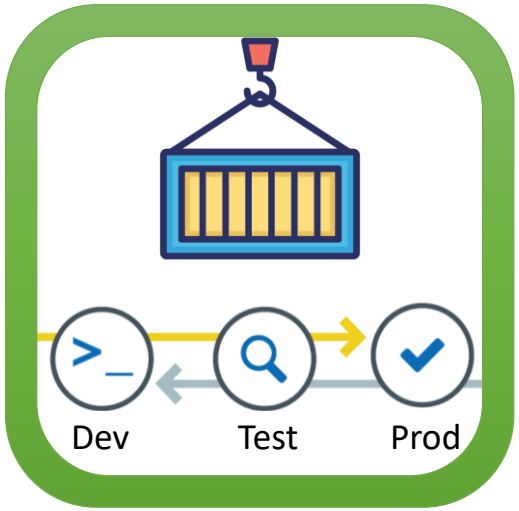
- Modeled on the success of shipping containers, an application container is designed to contain a complete deployment unit for an application to allow for automation, version tracking, and rapid deployment.
- Containers provide a standard way to package your application's code, configurations, and dependencies into a single object.



Advantages of using containers

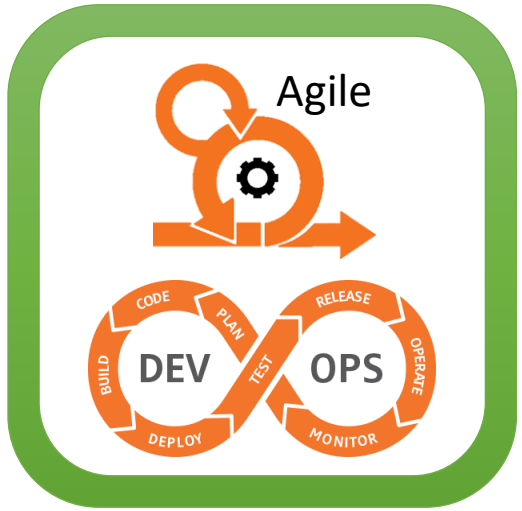


Smaller Footprint
and Less Overhead



Increased
Portability

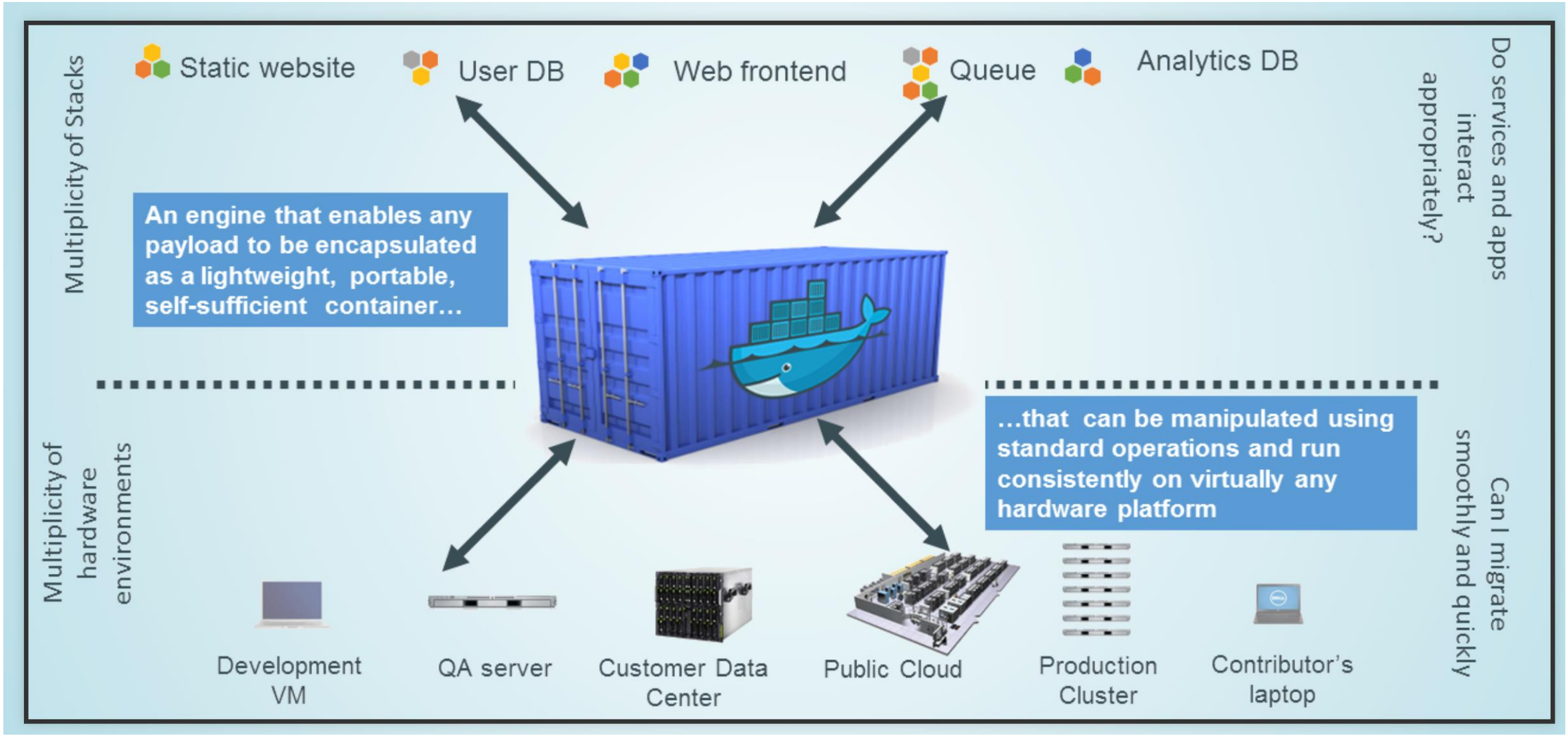
Greater Efficiency
and Smooth Scaling



Better Application
Development

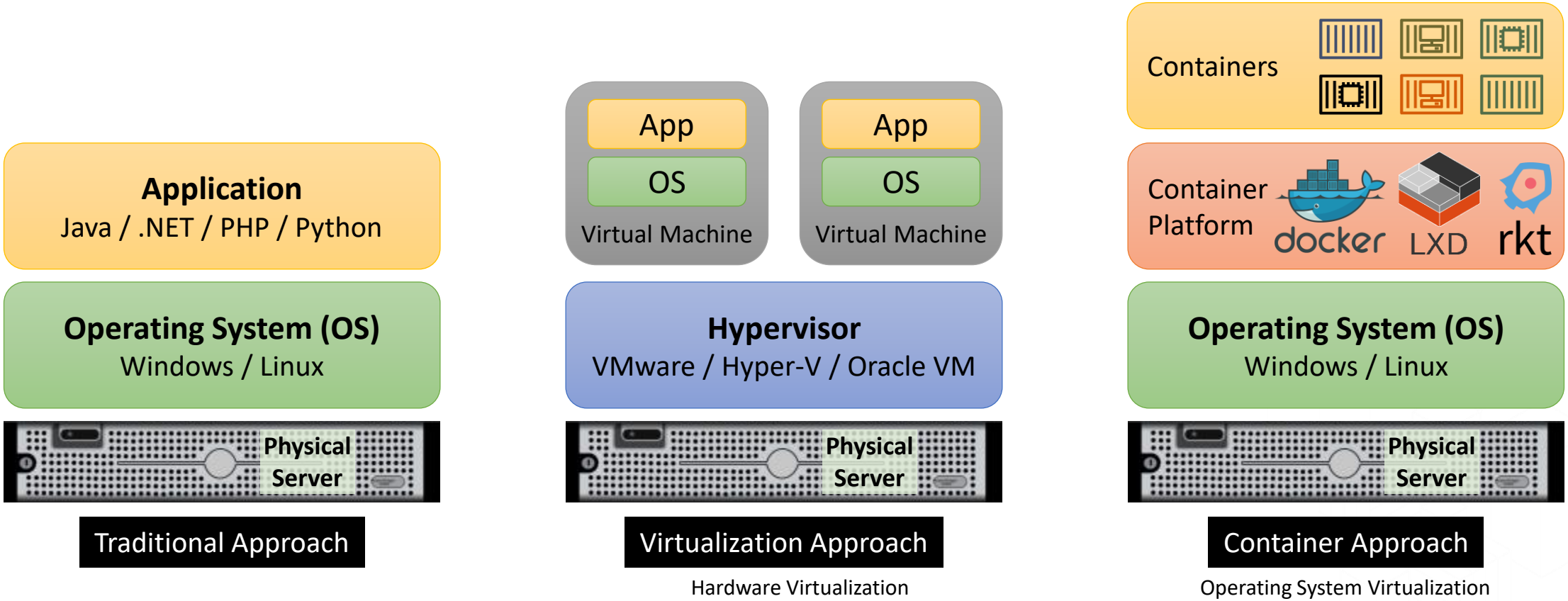
What is Docker?

- Docker is an open platform for developing, shipping, and running applications.



How containers run?

- Containers share an operating system installed on the server and run as resource-isolated processes, ensuring quick, reliable, and consistent deployments, regardless of environment.



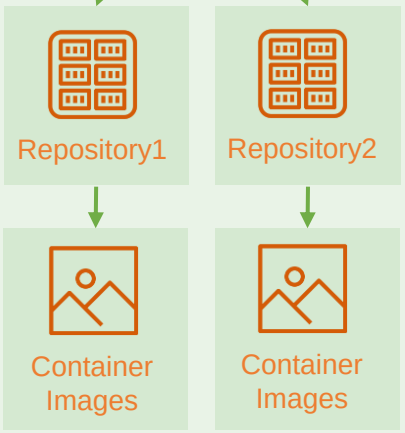
Containers in AWS

Running container on AWS

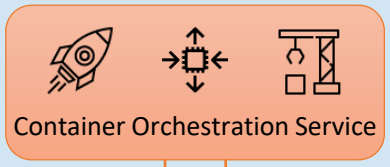
Image Storage

Store, encrypt, and manage container images

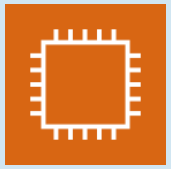
Amazon Elastic Container Registry (ECR)



Compute



Run containers with server level control



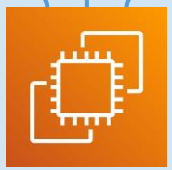
Self provisioned Cluster on EC2 Instance

Run containers using fully managed container orchestration service

Amazon Elastic Container Service (ECS)

Manage containers with Kubernetes

Amazon Elastic Kubernetes Service (EKS)



EC2 Hosted

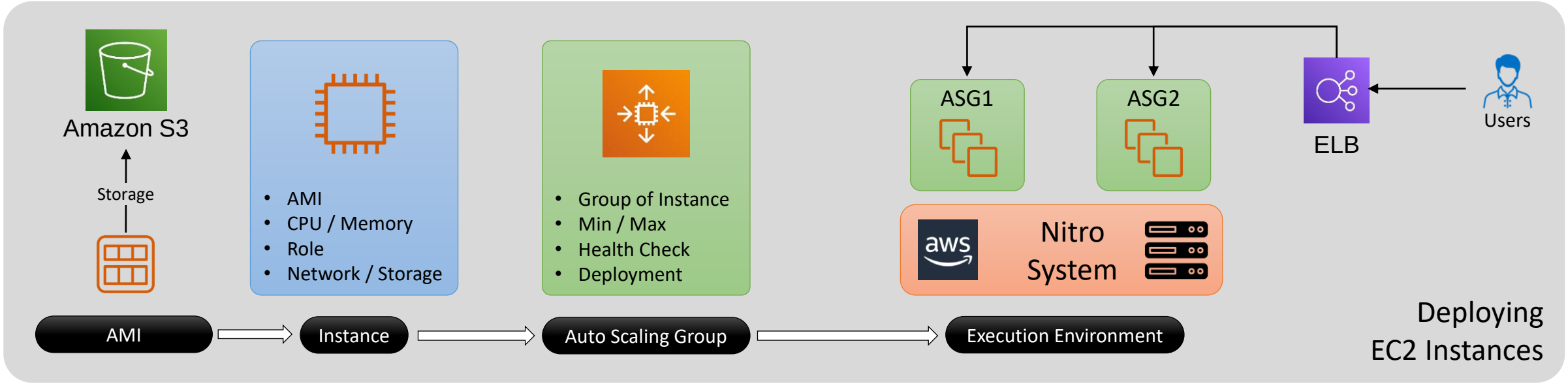
Run containers without managing servers

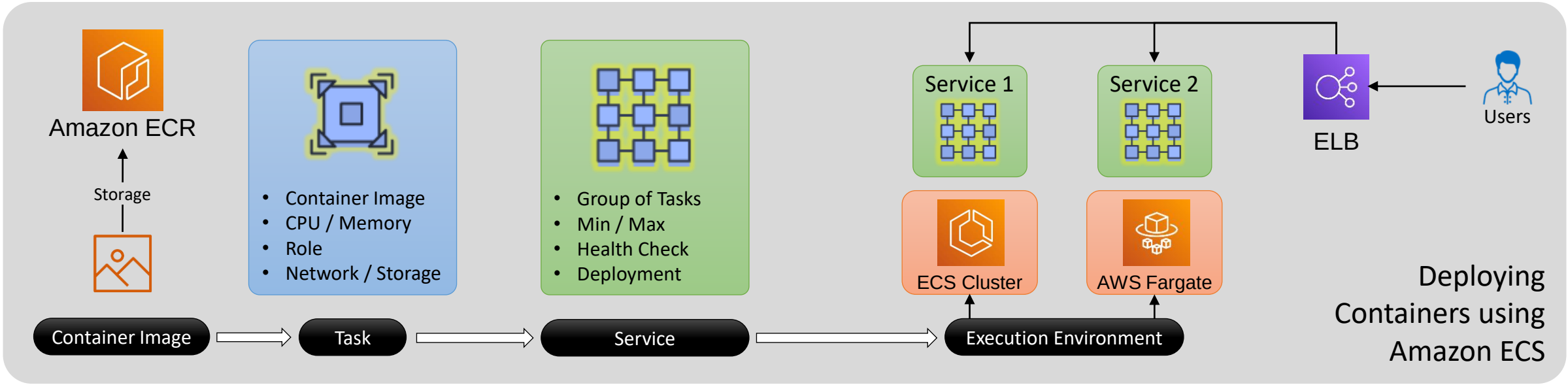
AWS Fargate

Serverless

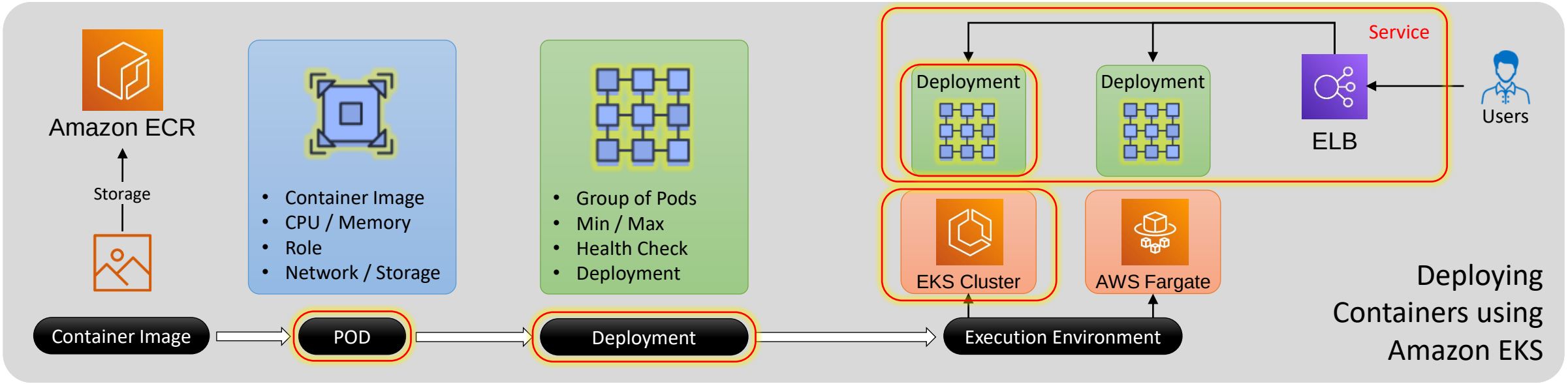
EC2 Instance vs. Containers

EC2 Instance vs Containers





Amazon EKS


Amazon ECR
Storage
Container Image
CPU / Memory
Role
Network / Storage
Group of Pods
Min / Max
Health Check
Deployment
Service
Deployment
Deployment
ELB
Users
EKS Cluster
AWS Fargate
Container Image
POD
Deployment
Execution Environment
Deploying Containers using Amazon EKS

Amazon Elastic Kubernetes Service (Amazon EKS)

Kubernetes

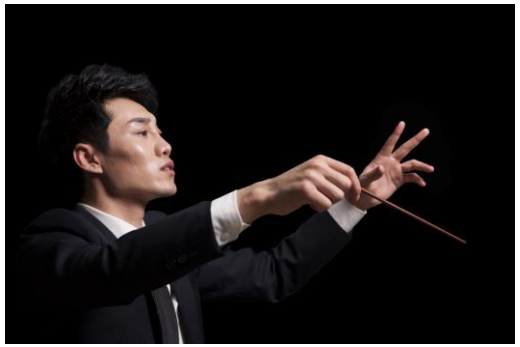
- Kubernetes is an open source orchestration system that helps deploy and scale your containerized applications

Open source container management platform governed by the community.

Run containers at scale integrating networking, storage, and compute.

Take advantage of open source tools that are designed to run on Kubernetes.

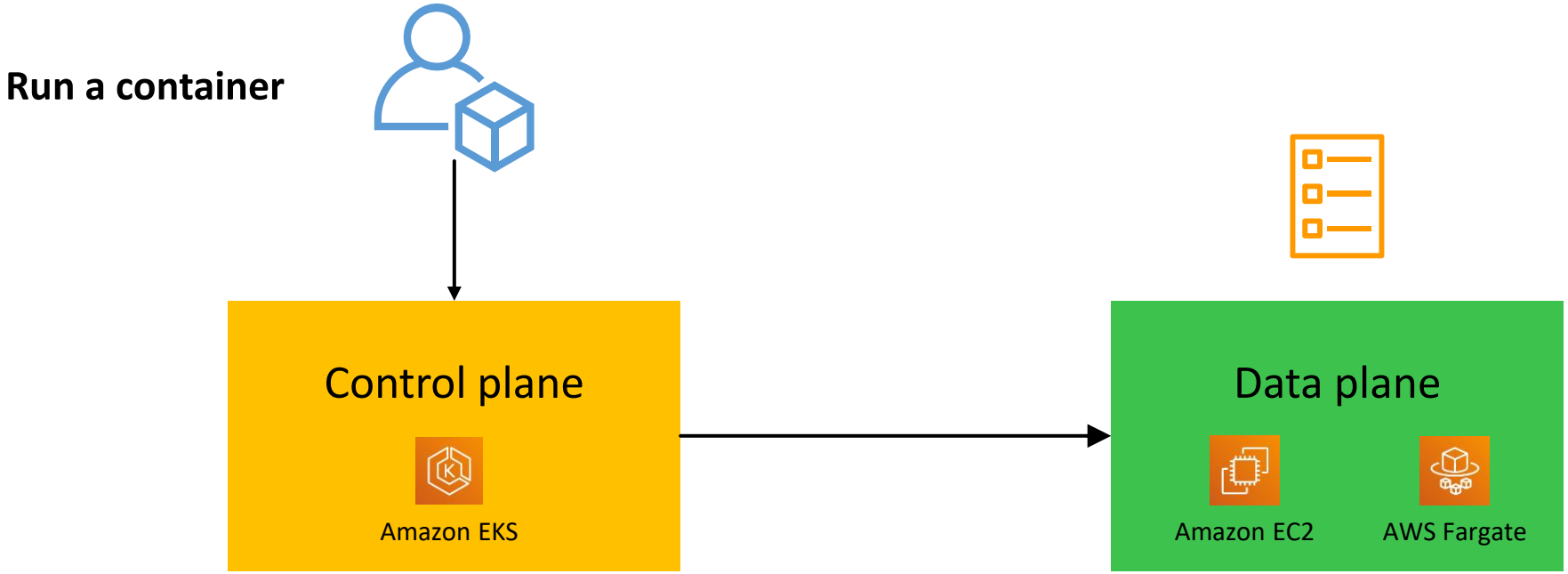
Structure of Kubernetes



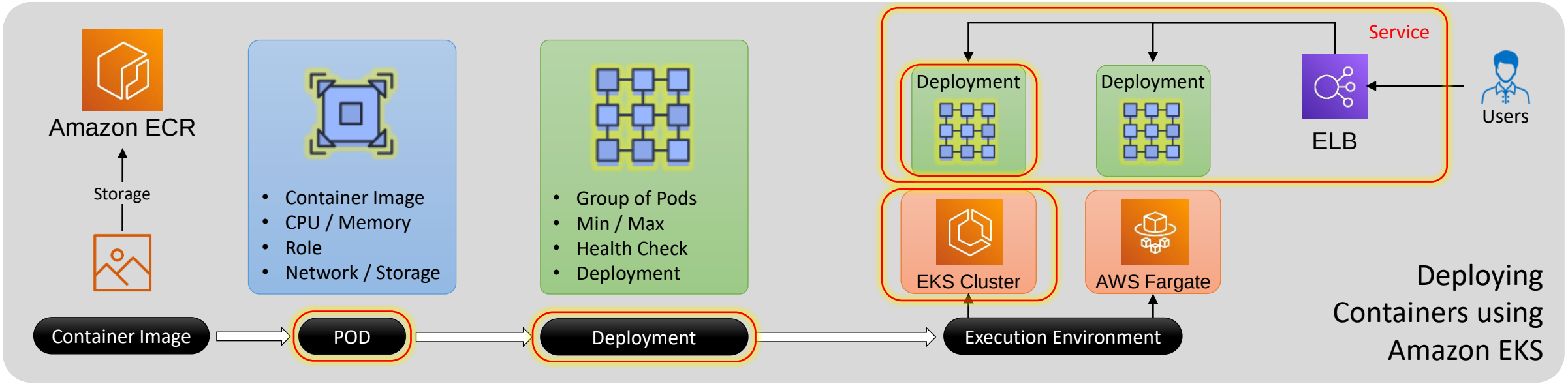
Control plane

Data plane

Structure of Kubernetes



Amazon EKS


Amazon ECR
Storage
Container Image
CPU / Memory
Role
Network / Storage
Group of Pods
Min / Max
Health Check
Deployment
Service
Deployment
Deployment
ELB
Users
EKS Cluster
AWS Fargate
Container Image
POD
Deployment
Execution Environment
Deploying Containers using Amazon EKS

Amazon EKS Portfolio

Amazon EKS

Amazon EKS
in Local Zones

Amazon EKS
in Wavelength Zones

Amazon EKS
on Outposts

Amazon EKS Anywhere

Amazon EKS Distro

AWS Managed

Customer Managed

Reference:

FAQs

Category:

Containers

Amazon Elastic Container Registry (Amazon ECR)

What?

- 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.
- You can easily access Amazon ECR from any Docker environment, whether in the cloud, on-premises, or on your local machine.

Why?

- Amazon ECR eliminates the need to operate and scale the infrastructure required to power your container registry.
- You can configure policies to manage permissions for each repository and restrict access to IAM users, roles, or other AWS accounts. Amazon ECR is integrated with third-party developer tools.

When?

- You want to use the familiar Docker CLI, or your preferred client, to push, pull, and manage images.
- You want to use lifecycle policies to manage your images, for example expiring images based on age or count.
- You want to scan software vulnerabilities in your container images on push using Amazon Inspector.

Where?

- Amazon ECR is a Regional service. Amazon ECR stores your container images and artifacts in Amazon S3.
- It also supports cross-Region and cross-account replication.

Who?

- You can easily push your container images to Amazon ECR using the Docker CLI from your development machine.
- You can set up AWS PrivateLink endpoints to pull images from your private repositories without traversing internet.

Complete book:

Click Here

How?

- After you create a repository, you can use your preferred CLI to push, pull, and manage Docker images, Open Container Initiative (OCI) images, and OCI compatible artifacts. Amazon ECR transfers your container images over HTTPS and automatically encrypts your images at rest. You can configure resource-based permissions using IAM.

Created by:

Ashish Prajapati

How much?

- You pay only for the amount of data you store in your public or private repositories, and data transferred to the internet.
- Data transferred between Amazon Elastic Container Registry and Amazon EC2 within a single region is free of charge.
- Data transferred between Amazon ECR and Amazon EC2 in different regions is charged at Internet Data Transfer rates.

Reference:

FAQs

Category:

Containers

Amazon Elastic Container Service (Amazon ECS)

Complete book:

Click Here

Created by:

Ashish Prajapati

What?

- Amazon ECS is a fully managed container orchestration service that makes it easy for you to deploy, manage, and scale containerized applications.

Why?

- Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure.
- You can use Amazon ECS to schedule container placement across your cluster based on your resource needs and availability requirements.

When?

- You want to schedule long-running applications, services, and batch processes using Docker containers.
- You want to maintains application availability and ability to scale your containers up or down to meet your application's capacity requirements.

Where?

- Amazon ECS is a regional service which can run and scale your container workloads across availability zones
- On your infrastructure with *Amazon ECS Anywhere*.

Who?

- Amazon ECS is a fully-managed container orchestration service, with AWS configuration and operational best practices built-in, and no control plane, nodes, or add-ons for you to manage.
- Customers define Tasks, Service, Capacity Providers and deployment process.

How?

- Amazon ECS allows you to easily run applications on a managed cluster of Amazon EC2 instances.
- Your containers are defined in a task definition that you use to run an individual tasks or task within a service.
- Amazon ECS is integrated with familiar features like Elastic Load Balancing, EBS volumes, Amazon VPC and IAM.

How much?

- There is no additional charge for Amazon ECS. You pay for AWS resources (e.g. Amazon EC2 instances or EBS volumes) you create to store and run your application.
- You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments.

Reference:

FAQs

Category:

Containers

Amazon Elastic Kubernetes Service (Amazon EKS)

What?

- Amazon EKS is a managed service that makes it easy for you to run Kubernetes on AWS without installing and operating your own Kubernetes control plane or worker nodes.

Why?

- Amazon EKS provisions and scales the Kubernetes control plane, including the API servers and backend persistence layer, across multiple AWS Availability Zones (AZs) for high availability and fault tolerance.
- Amazon EKS automatically detects and replaces unhealthy control plane nodes and patches the control plane.

When?

- You don't want operational burden of managing the Kubernetes control plane.
- You want to maintain existing applications that run on upstream Kubernetes and want to use plugins and tooling from the Kubernetes community.

Where?

- Amazon EKS is a regional service.
- It runs and scales the Kubernetes control plane across multiple AWS Availability Zones to ensure high availability.

Who?

- Amazon EKS handles provisioning, scaling, and managing the Kubernetes control plane. It also automatically detects and replaces unhealthy masters, and it provides automated version upgrades and patching for the masters.
- Customers provision an EKS cluster, Deploy Compute nodes, connect to EKS and run Kubernetes applications.

Complete book:

Click Here

How?

- Amazon EKS runs a single tenant Kubernetes control plane for each cluster. It integrates with many AWS services to provide scalability and security for your applications
- It uses Amazon VPC network policies to restrict traffic between control plane components to a single cluster.

Created by:

Ashish Prajapati

How much?

- You pay $0.10 per hour for each Amazon EKS cluster you create and for the AWS resources you create to run your Kubernetes worker nodes. You only pay for what you use, as you use it; there are no minimum fees and no upfront commitments.
- If you are using AWS Fargate, pricing is calculated based on the vCPU and memory resources used.