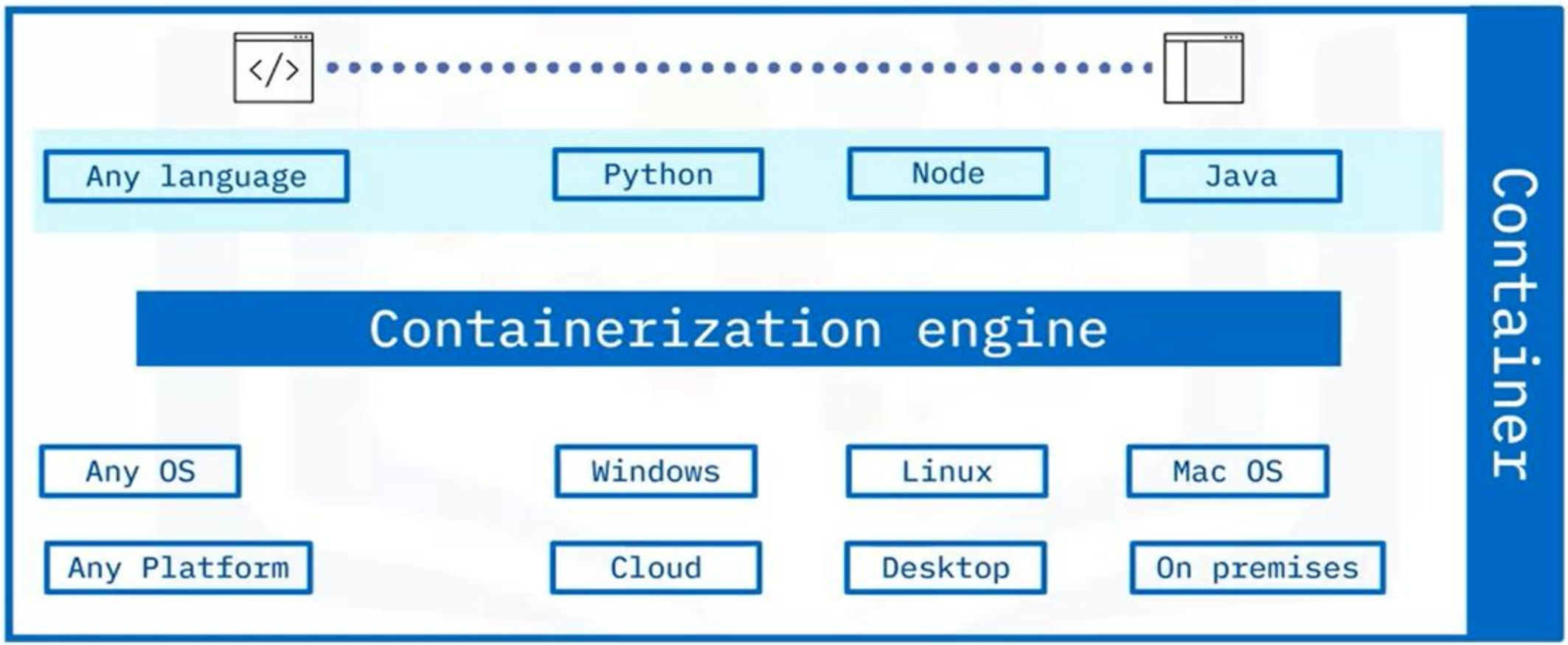

docker

Why use containers?

Isolation and Allocation	No way to define resource boundaries for apps in a physical server
Server Utilization	Not optimal because servers tend to be either over-utilized or under-utilized
Provisioning and Costs	Requires long periods for provisioning resources and expensive maintenance costs
Performance	Constrained during peak workloads
Portability	Applications are not portable across multiple environments and operating systems
Resiliency	Complex, time-consuming and expensive
Scalability	Limited scalability and resiliency
Automation	Difficult to implement for multiple platforms

Containers offer easy portability



Container benefits

Containers enable organizations to:

- Quickly create applications using automation
- Lower deployment time and costs
- Improve resource utilization (CPU, memory)
- Port across different environments
- Support next-gen applications (microservices)

Container challenges

- Security impacted if operating system affected
- Difficult to manage thousands of containers
- Complex to migrate legacy projects to container technology
- Difficult to right-size containers for specific scenarios

Container vendors

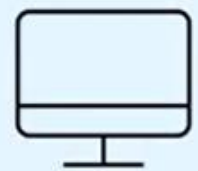
Docker	• Robust and most popular container platform today
Podman	• Daemon-less architecture providing more security than Docker containers
LXC	• Preferred for data-intensive apps and ops
Vagrant	• Offers highest levels of isolation on the running physical machine

Docker defined

Available since 2013, Docker is an open platform, or engine, where programmers can:

Develop

Ship

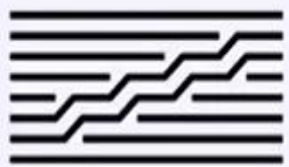


Run

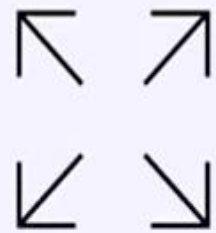
Containers

Docker becomes popular

Docker became popular due to:



Simple architecture



Scalability

Easy portability

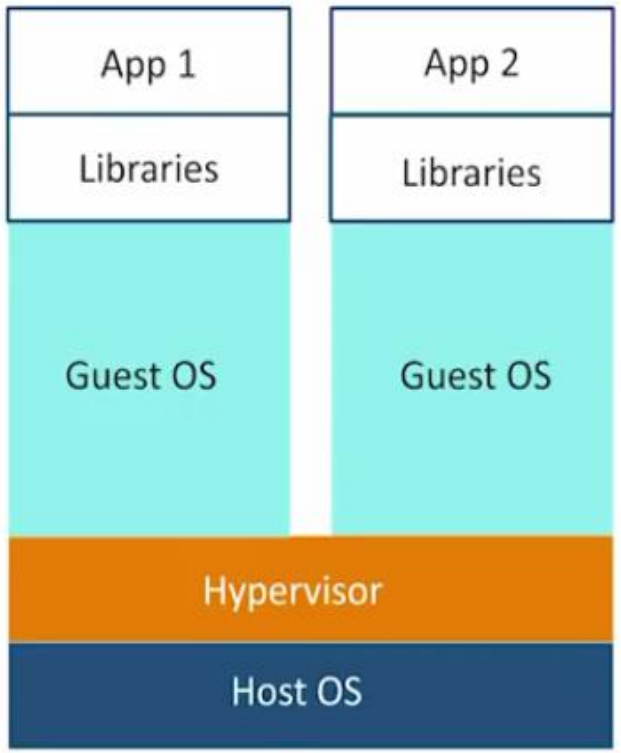
Docker’s underlying technology

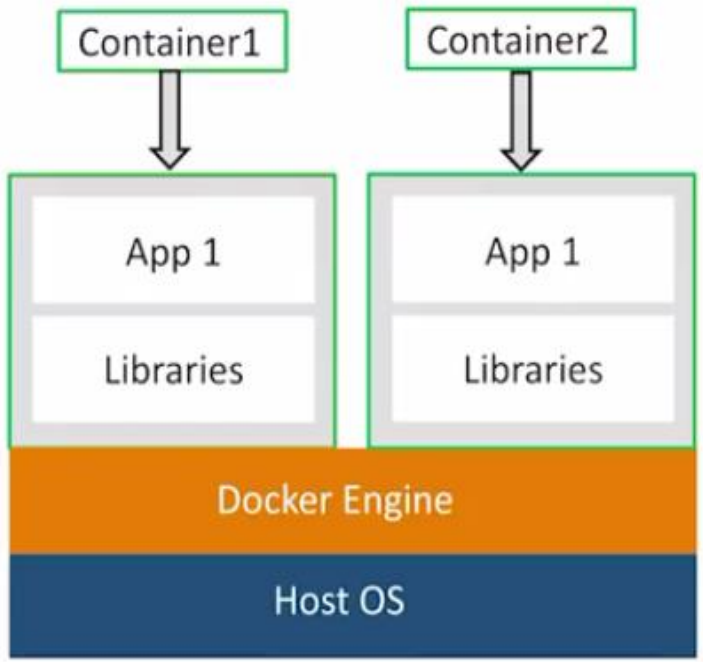
- Written in Go programming language
- Uses Linux kernel’s features to deliver functionality
- Uses the namespaces technology to provide an isolated workspace called “container”
- Creates a set of namespaces for every container and each aspect runs in a separate namespace with access limited to that namespace

Docker benefits

- ✓Consistent and isolated environments
- ✓Fast deployment
- ✓Repeatability and automation
- ✓Supports Agile and CI/CD DevOps practices
- ✓Versioning for easy testing, rollbacks, and redeployments
- ✓Collaboration, modularity, and scaling
- ✓Easy portability and flexibility

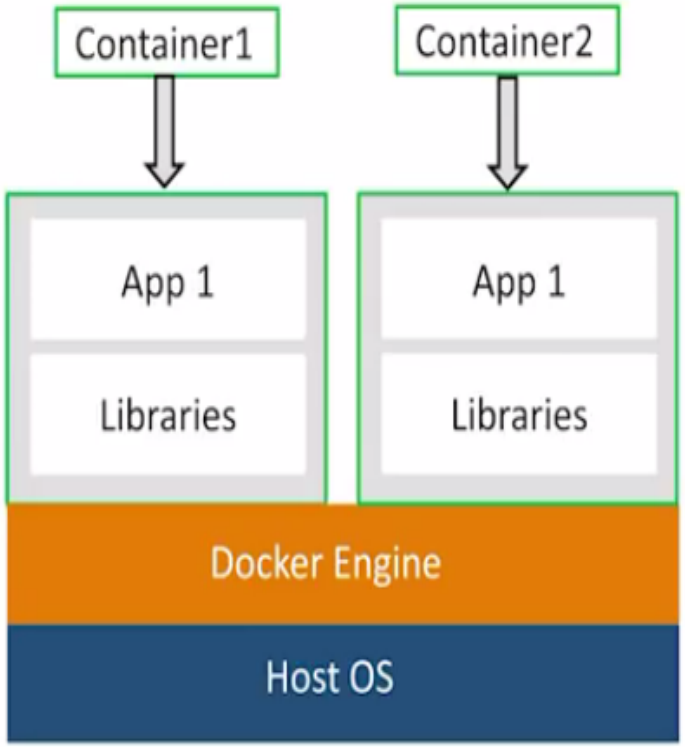
Virtualisation Vs Containerisation





What is Docker?

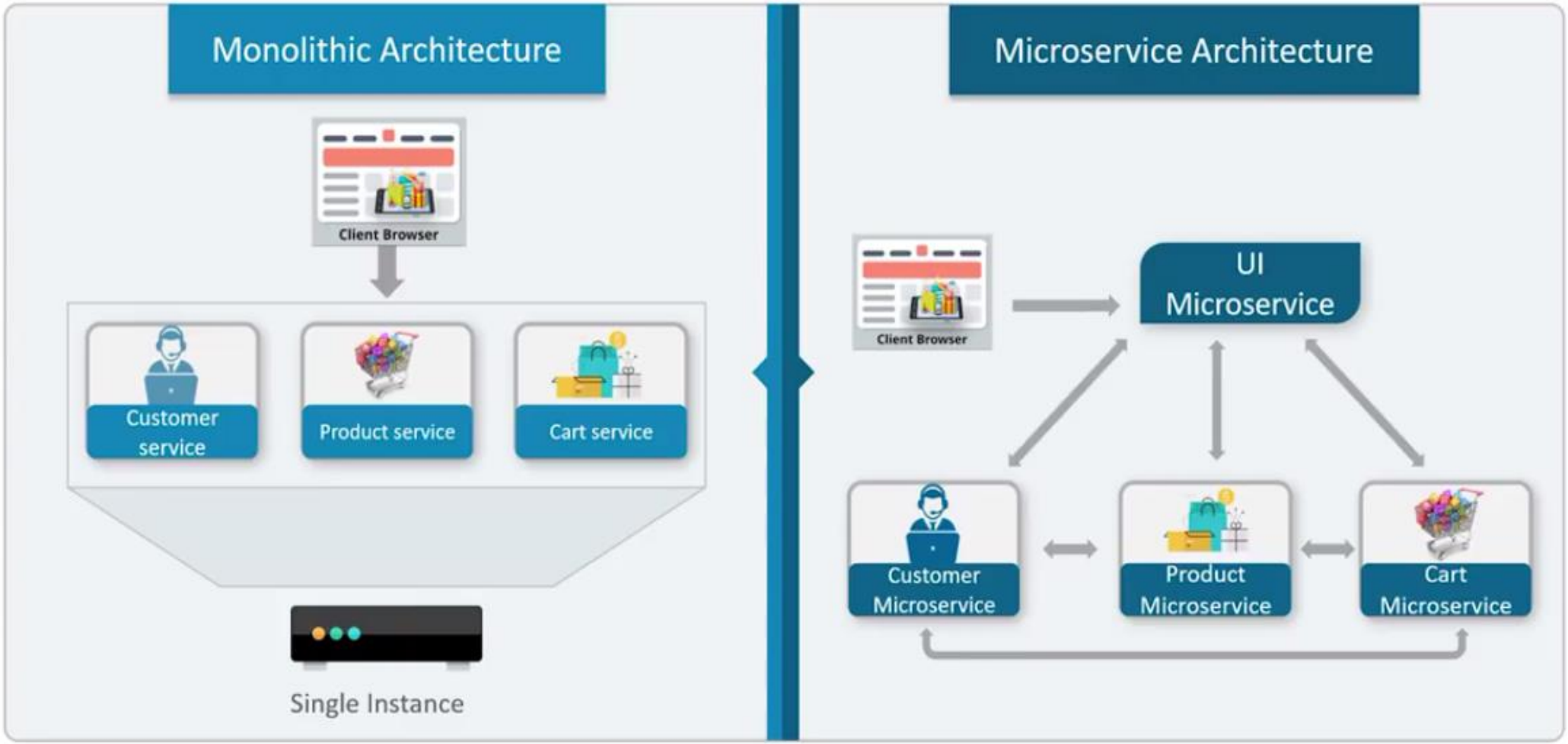
- Runs applications within Docker containers
- Alternative to VMs & use host's OS



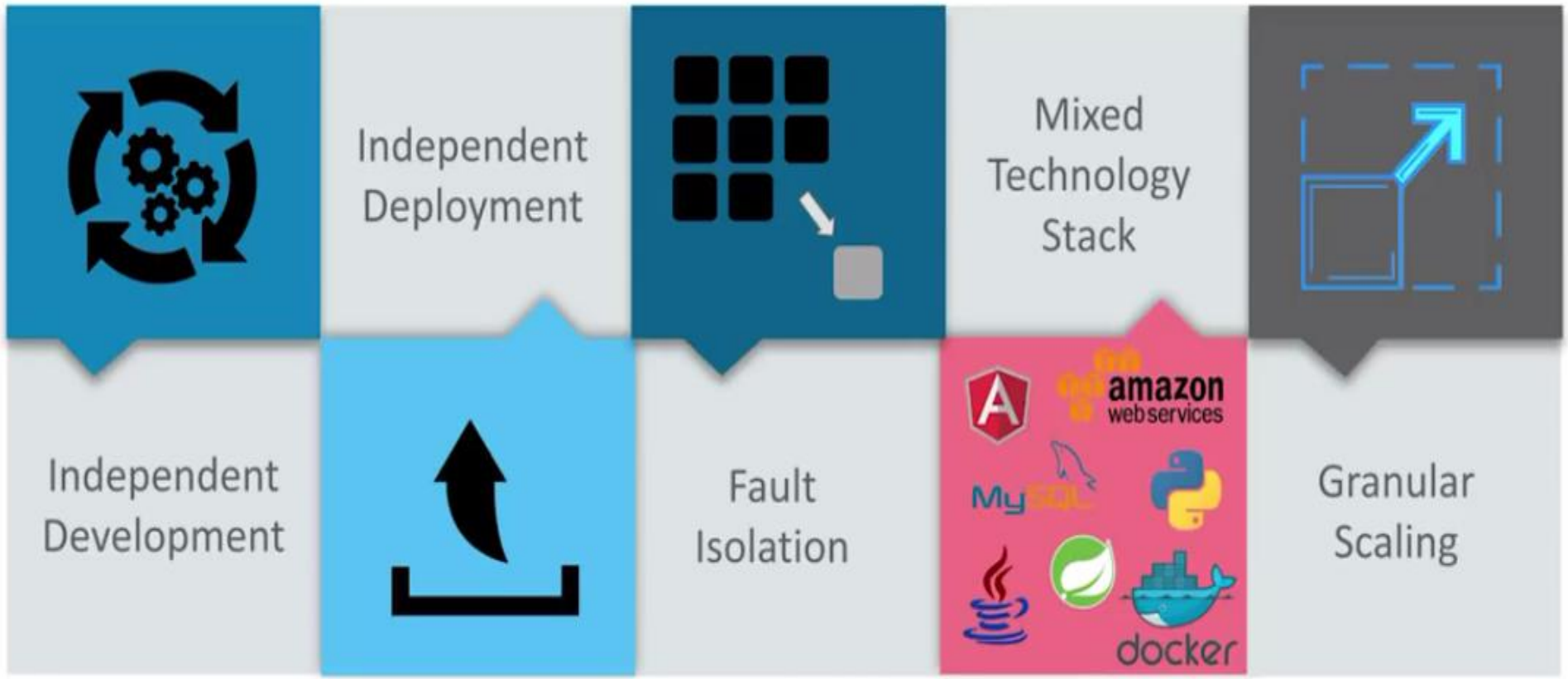
3 terminologies to remember

- Docker Image is built using a Dockerfile
- Dockerfile contains all the application dependencies
- Docker container is an instance of a docker image

What are Microservices?

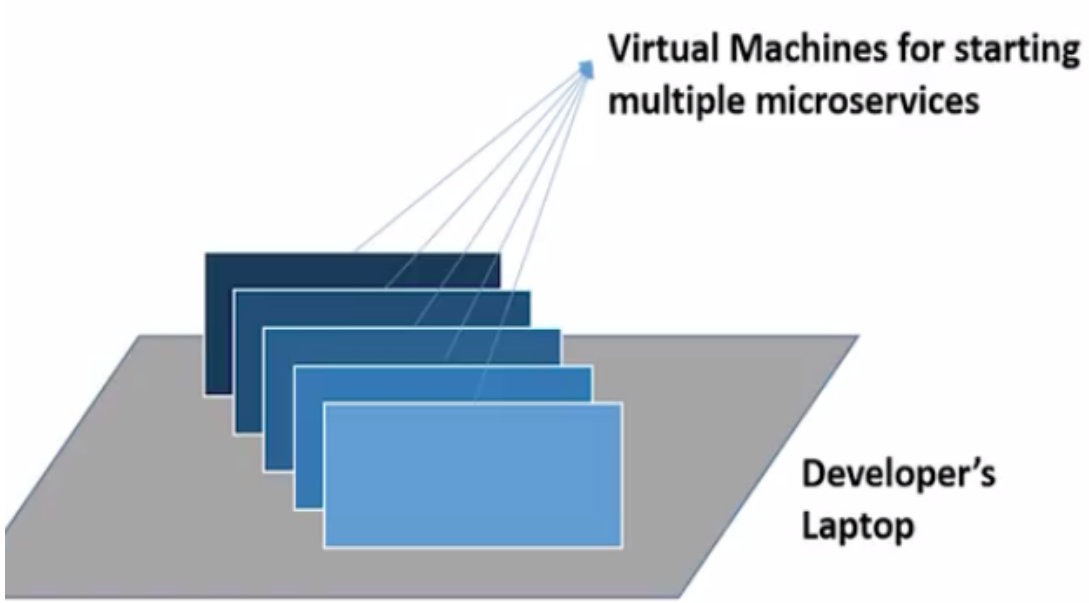


Advantages of Microservice Architecture

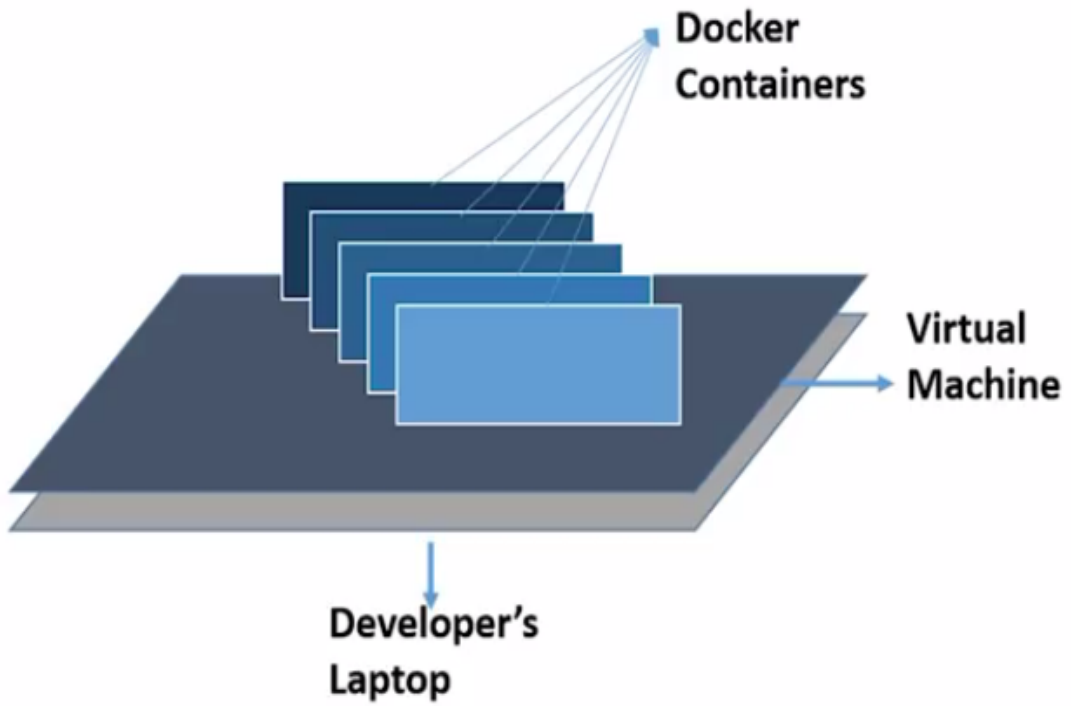


VMs vs Docker Containers For Microservices

Developing an application requires starting several of microservices in one machine. So if you are starting five of those services you require five VMs on that machine.



You can run several microservices in the same VM by running various Docker containers for each microservice.



Problem Statement

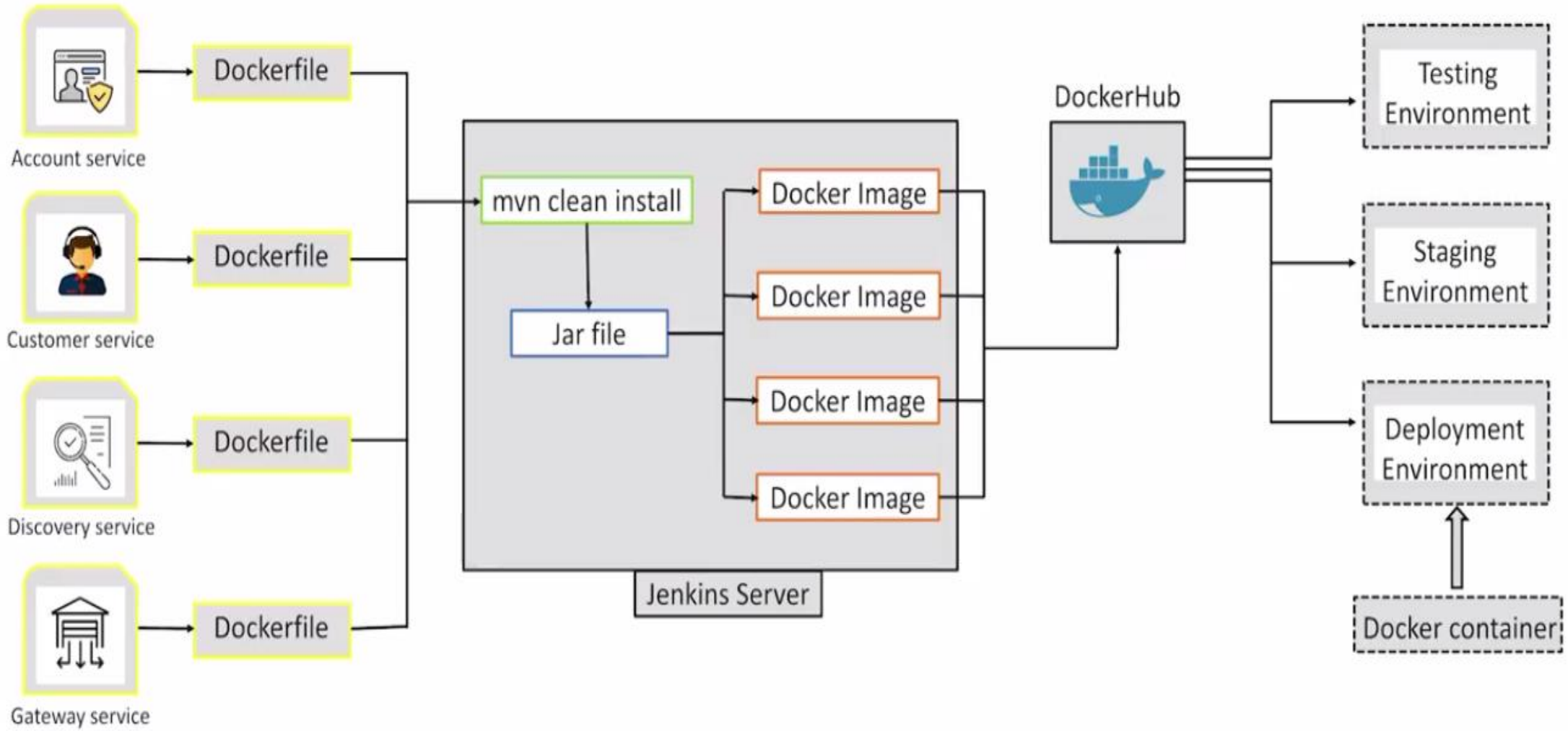
Dev server

Test server

Prod server

Inconsistent computing environment

Solution


Account service
Dockerfile
Customer service
Dockerfile
Discovery service
Dockerfile
Gateway service
Dockerfile
mvn clean install
Jar file
Docker Image
Docker Image
Docker Image
Docker Image
Jenkins Server
DockerHub
Testing Environment
Staging Environment
Deployment Environment
Docker container


https://docs.docker.com/docker-for-windows/install/
Search the docs
Guides
Product manuals
Glossary
Reference
Samples
Install Docker Desktop on Windows
Estimated reading time: 6 minutes
Docker Desktop for Windows is the Community version of Docker for Microsoft Windows. You can download Docker Desktop for Windows from Docker Hub.
Download from Docker Hub
By downloading Docker Desktop, you agree to the terms of the Docker Software End User License Agreement and the Docker Data Processing Agreement.

Most Used Docker Commands

```
docker --version
docker --help
docker pull
docker run
docker build
docker login
docker push
```

```
docker ps
docker images
docker stop
docker kill
docker rm
docker rmi
docker exec
```

```
docker commit
docker import
docker export
docker container
docker compose
docker swarm
docker service
```

docker pull

```
$ docker pull ubuntu
```

This command pulls a new Docker image from the Docker Hub

docker images

```
$ docker images
```

This command lists down all the images in your local repo

docker run

```
$ docker run ubuntu
```

This command executes a Docker image on your local repo & creates a running Container out of it

docker build

```
$ docker build -t MyUbuntuImage .
```

This command is used to compile the Dockerfile, for building custom Docker images based on the

docker login

```
$ docker login
```

This command is used to Login to Docker Hub repo from the CLI

docker push

```
$ docker push vardhanns/MyUbuntuImage
```

This command pushes a Docker image on your local repo to the Docker Hub

docker ps

```
$ docker ps
```

This command lists all the running containers in the host
If '–a' flag is specified, shutdown containers are also displayed

```
$ docker ps -a
```

docker stop

```
$ docker stop fe6e370a1c9c
```

This command shuts down the container whose Container ID is specified in arguments. Container is shut down gracefully by waiting for other dependencies to shut

docker kill

```
$ docker kill fe6e370a1c9c
```

This command kills the container by stopping its execution immediately. Its similar to force kill

docker rm

```
$ docker rm fe6e370a1c9c
```

This command removes the container whose Container ID is specified in arguments

Docker commands

- docker run ubuntu
- docker ps
- // list all running containers
- docker ps -a
- // shows all running as well as previous conatiners
- //for stoping the container
- docker stop // conatiner name or ID
- //for removing the container
- docker rm //container name or id
- //for listing all the images
- docker images
- //for removing the images
- docker rmi ubuntu
- //rm is used to remove containers
- //rmi is used to remove images
- //for downloading the image first time use the following command

Docker commands

- docker run ubuntu
- //thereafter we have to use pull command
- docker pull ubantu

- docker run ubuntu sleep 1000
- //Execute a command inside a conatiner
- docker exec "//put the name of container " cat /etc/hosts

- //you can use any image available on http://hub.docker.com or you can create your own image
- docker run centos

- docker run -it centos bash
- //for exit
- exit

- docker run -d centos sleep 2000