DevOps Tools

Day - 16

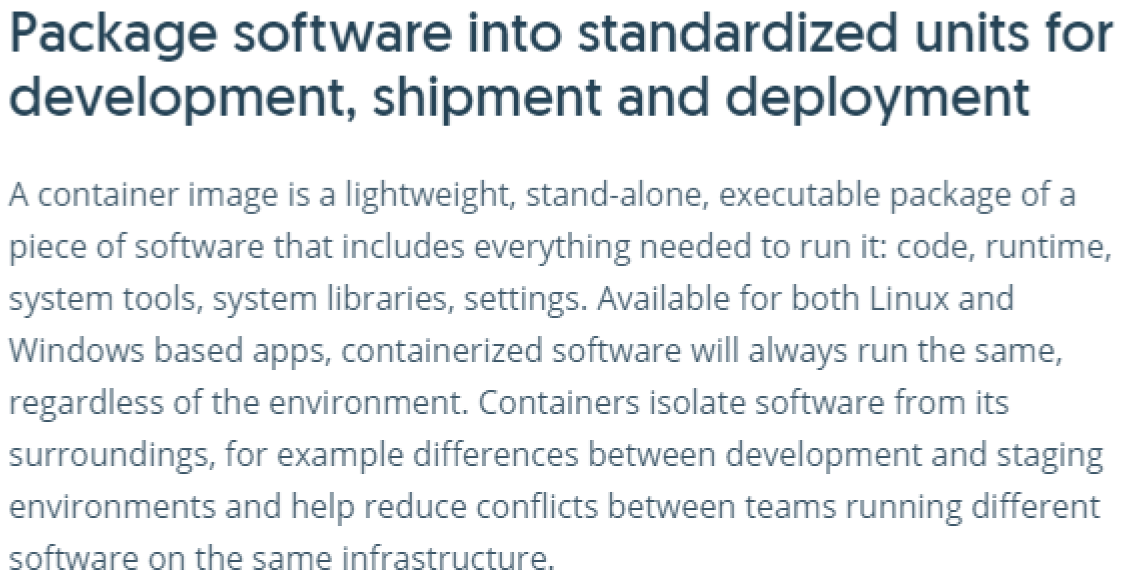
Prakash Ramamurthy
prakash.ramamurthy@wipro.com

Agenda

- Containerization and Docker
- Pros and Cons of Containerization
- Hands-On Demonstration

Containerization and Docker

Containers

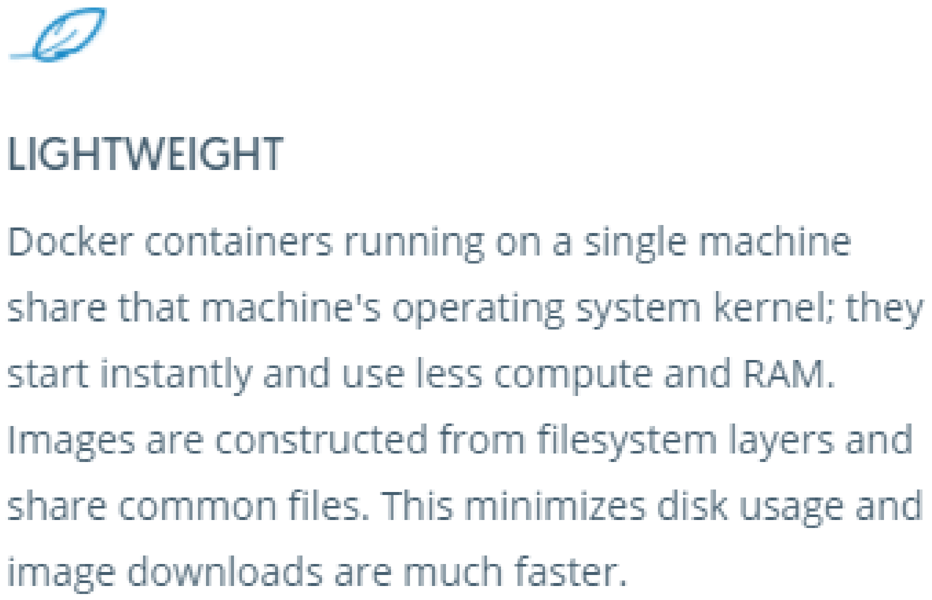


Package software into standardized units for development, shipment and deployment

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings. Available for both Linux and Windows based apps, containerized software will always run the same, regardless of the environment. Containers isolate software from its surroundings, for example differences between development and staging environments and help reduce conflicts between teams running different software on the same infrastructure.

Source: https://www.docker.com/what-container

Containers

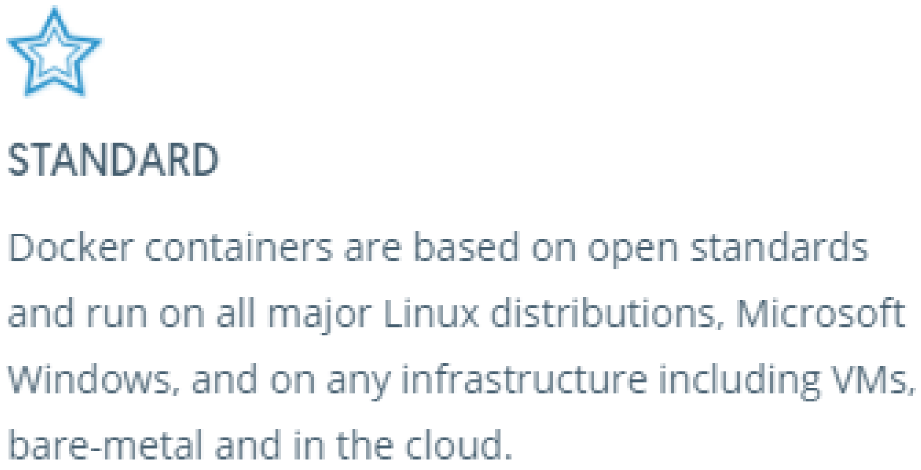


LIGHTWEIGHT

Docker containers running on a single machine share that machine's operating system kernel; they start instantly and use less compute and RAM. Images are constructed from filesystem layers and share common files. This minimizes disk usage and image downloads are much faster.

Source: https://www.docker.com/what-container

Containers

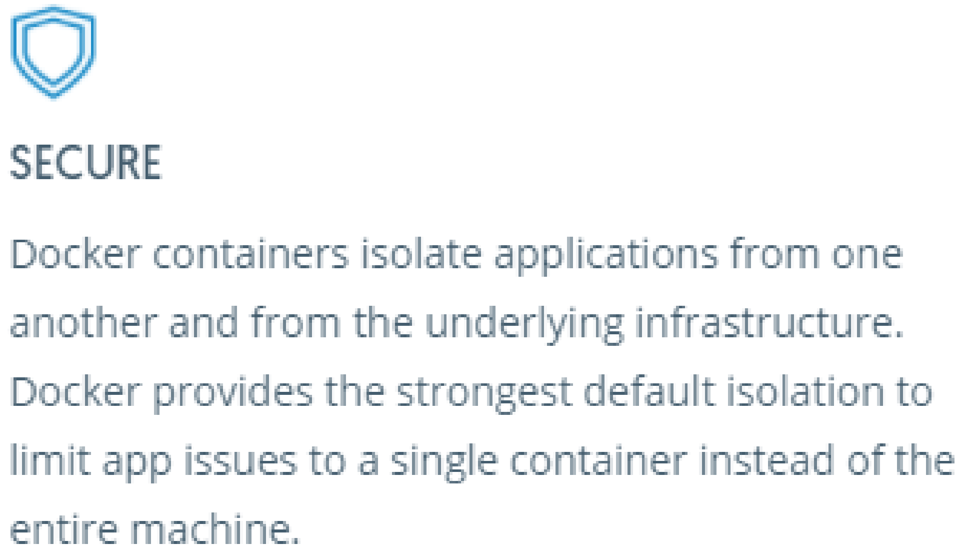


STANDARD

Docker containers are based on open standards and run on all major Linux distributions, Microsoft Windows, and on any infrastructure including VMs, bare-metal and in the cloud.

Source: https://www.docker.com/what-container

Containers

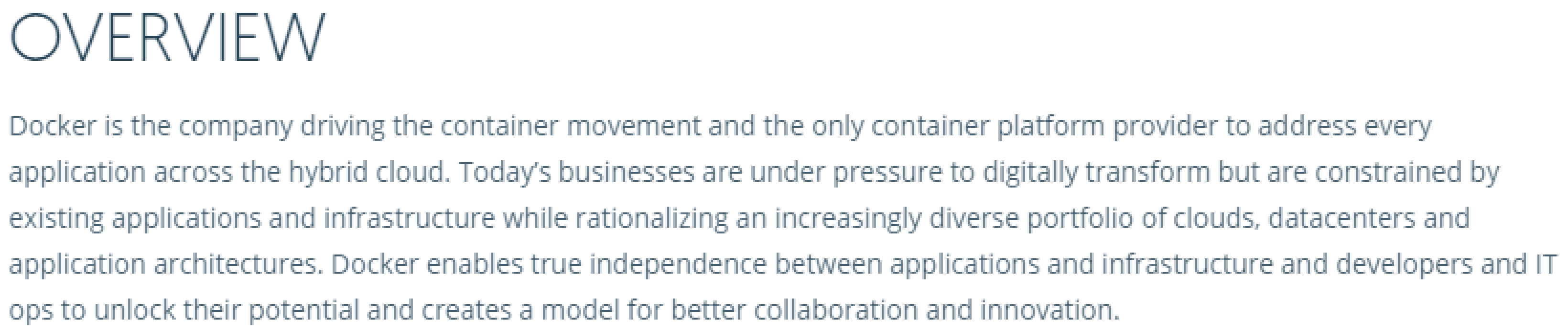


SECURE

Docker containers isolate applications from one another and from the underlying infrastructure. Docker provides the strongest default isolation to limit app issues to a single container instead of the entire machine.

Source: https://www.docker.com/what-container

Docker

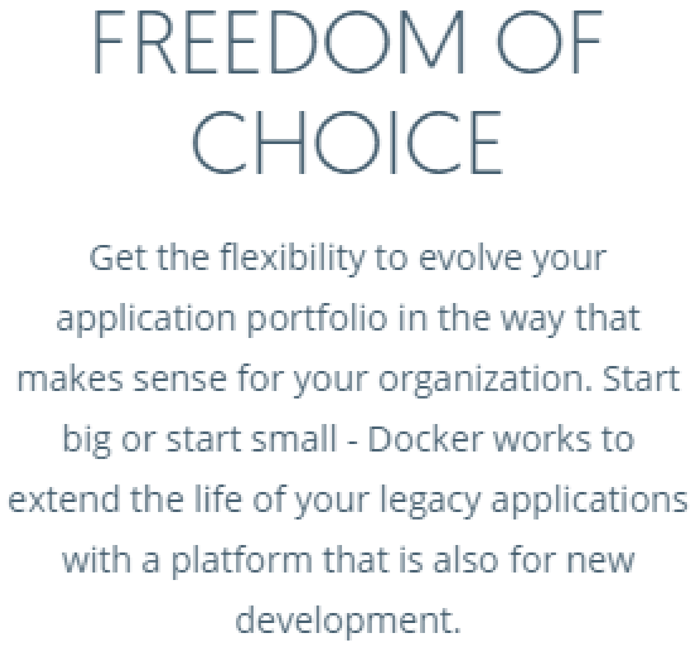


OVERVIEW

Docker is the company driving the container movement and the only container platform provider to address every application across the hybrid cloud. Today's businesses are under pressure to digitally transform but are constrained by existing applications and infrastructure while rationalizing an increasingly diverse portfolio of clouds, datacenters and application architectures. Docker enables true independence between applications and infrastructure and developers and IT ops to unlock their potential and creates a model for better collaboration and innovation.

Source: https://www.docker.com/what-docker

Docker


FREEDOM OF CHOICE

Get the flexibility to evolve your application portfolio in the way that makes sense for your organization. Start big or start small - Docker works to extend the life of your legacy applications with a platform that is also for new development.

Source: https://www.docker.com/what-docker

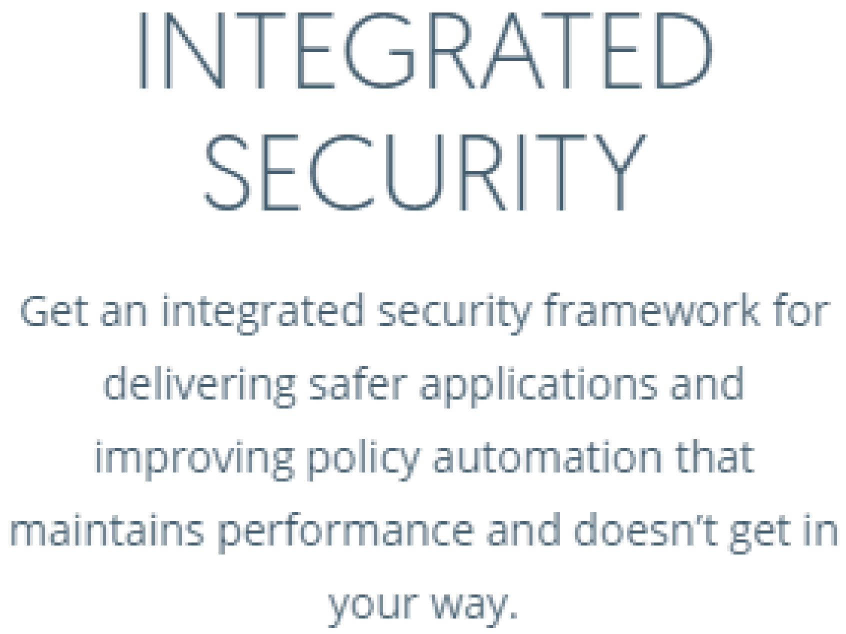
Docker

AGILE OPERATIONS

Standardize and automate the way you build, manage, and secure applications across departments. Because Docker puts developers and IT operations on the same platform, they can work smarter and faster together.

Source: https://www.docker.com/what-docker

Docker

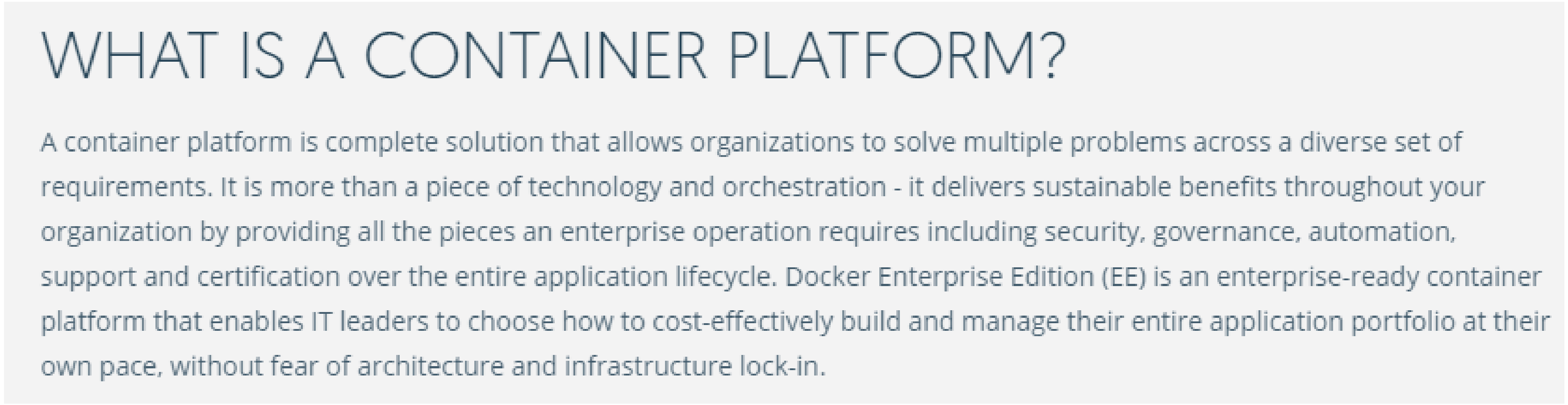


INTEGRATED SECURITY

Get an integrated security framework for delivering safer applications and improving policy automation that maintains performance and doesn't get in your way.

Source: https://www.docker.com/what-docker

Docker

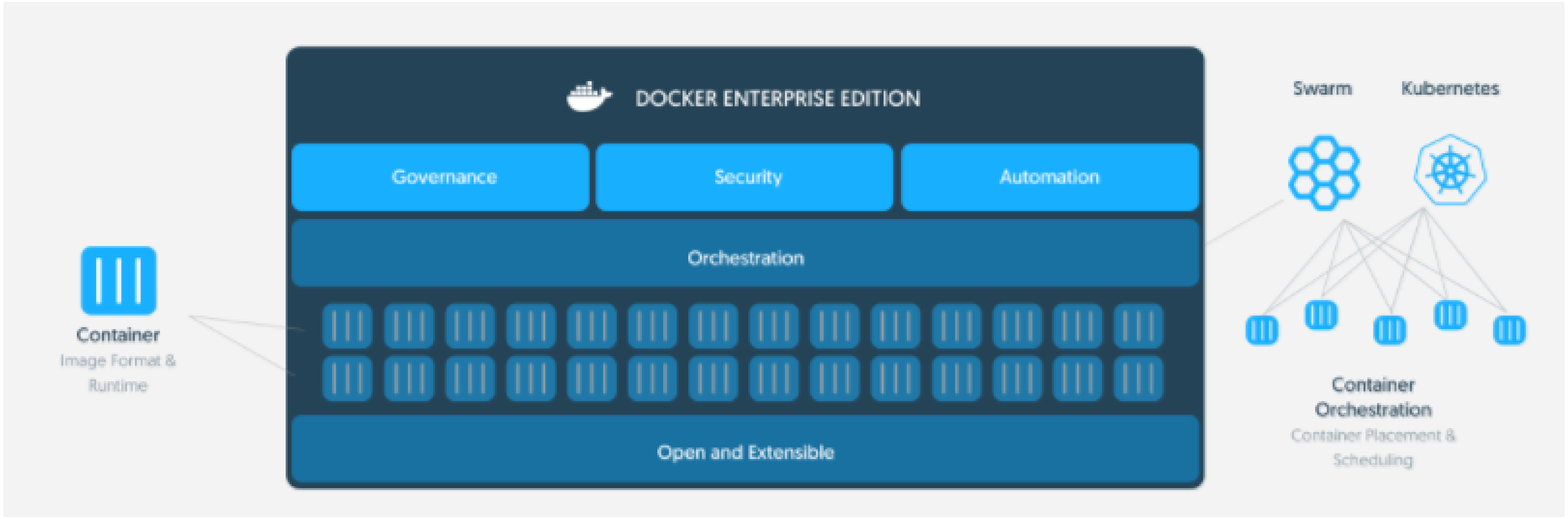


WHAT IS A CONTAINER PLATFORM?

A container platform is complete solution that allows organizations to solve multiple problems across a diverse set of requirements. It is more than a piece of technology and orchestration - it delivers sustainable benefits throughout your organization by providing all the pieces an enterprise operation requires including security, governance, automation, support and certification over the entire application lifecycle. Docker Enterprise Edition (EE) is an enterprise-ready container platform that enables IT leaders to choose how to cost-effectively build and manage their entire application portfolio at their own pace, without fear of architecture and infrastructure lock-in.

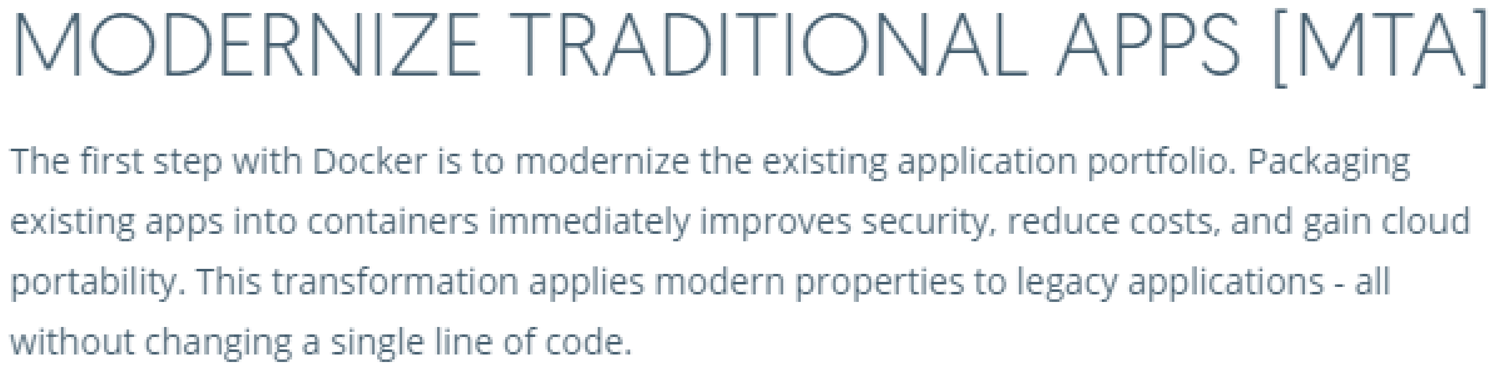
Source: https://www.docker.com/what-docker

Docker



Source: https://www.docker.com/what-docker

Docker

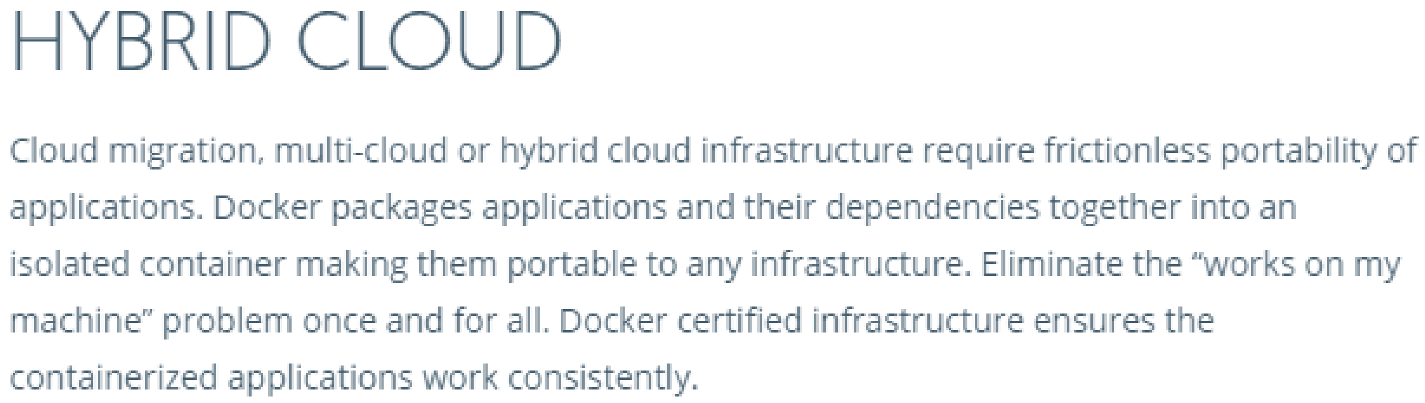


MODERNIZE TRADITIONAL APPS [MTA]

The first step with Docker is to modernize the existing application portfolio. Packaging existing apps into containers immediately improves security, reduce costs, and gain cloud portability. This transformation applies modern properties to legacy applications - all without changing a single line of code.

Source: https://www.docker.com/what-docker

Docker

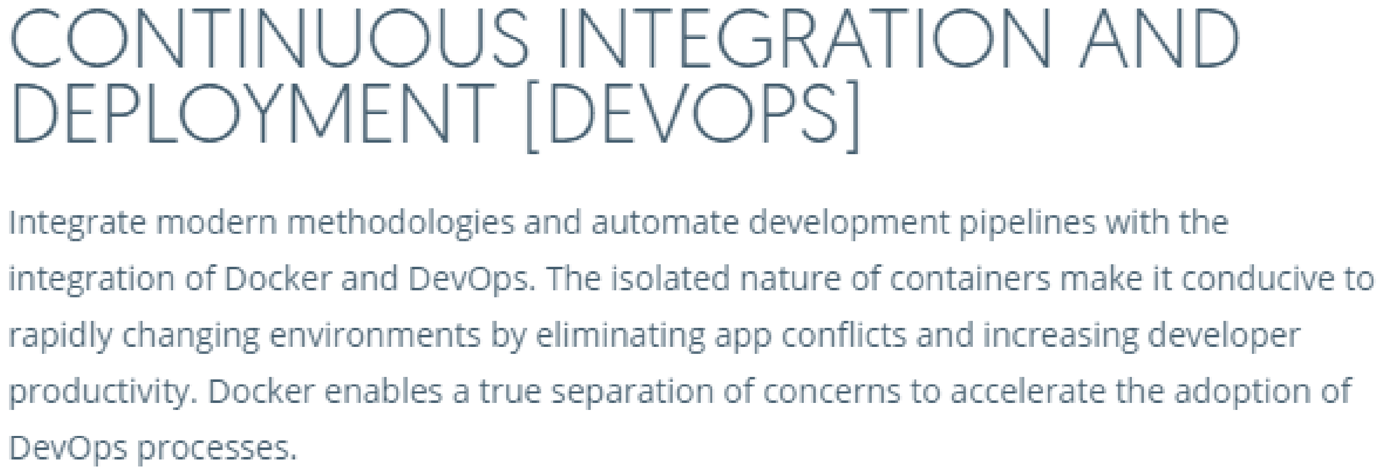


HYBRID CLOUD

Cloud migration, multi-cloud or hybrid cloud infrastructure require frictionless portability of applications. Docker packages applications and their dependencies together into an isolated container making them portable to any infrastructure. Eliminate the "works on my machine" problem once and for all. Docker certified infrastructure ensures the containerized applications work consistently.

Source: https://www.docker.com/what-docker

Docker

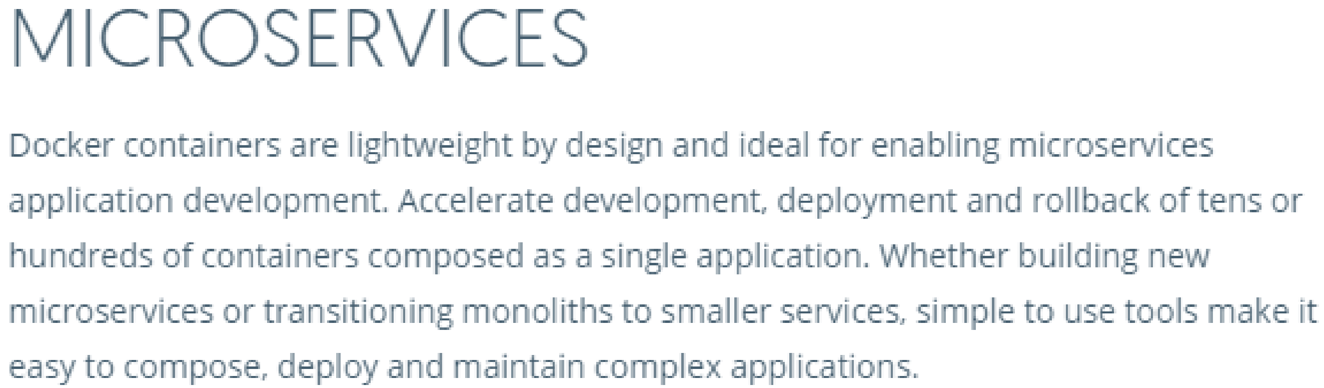


CONTINUOUS INTEGRATION AND DEPLOYMENT [DEVOPS]

Integrate modern methodologies and automate development pipelines with the integration of Docker and DevOps. The isolated nature of containers make it conducive to rapidly changing environments by eliminating app conflicts and increasing developer productivity. Docker enables a true separation of concerns to accelerate the adoption of DevOps processes.

Source: https://www.docker.com/what-docker

Docker

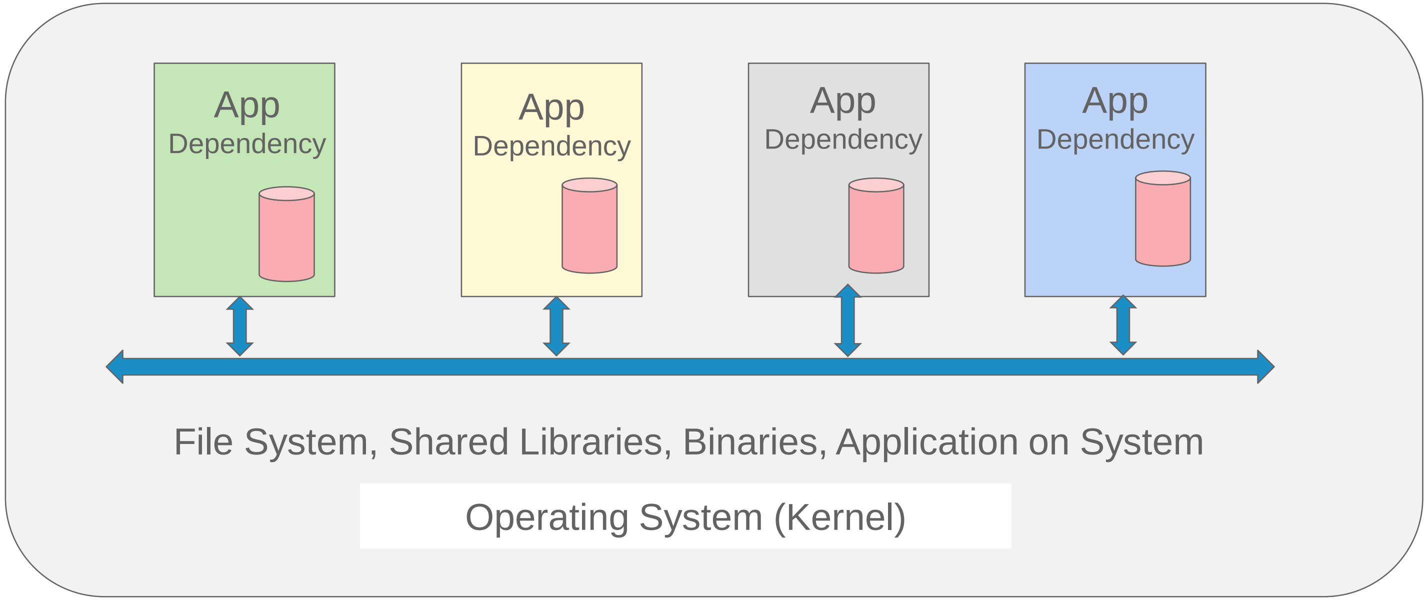


MICROSERVICES

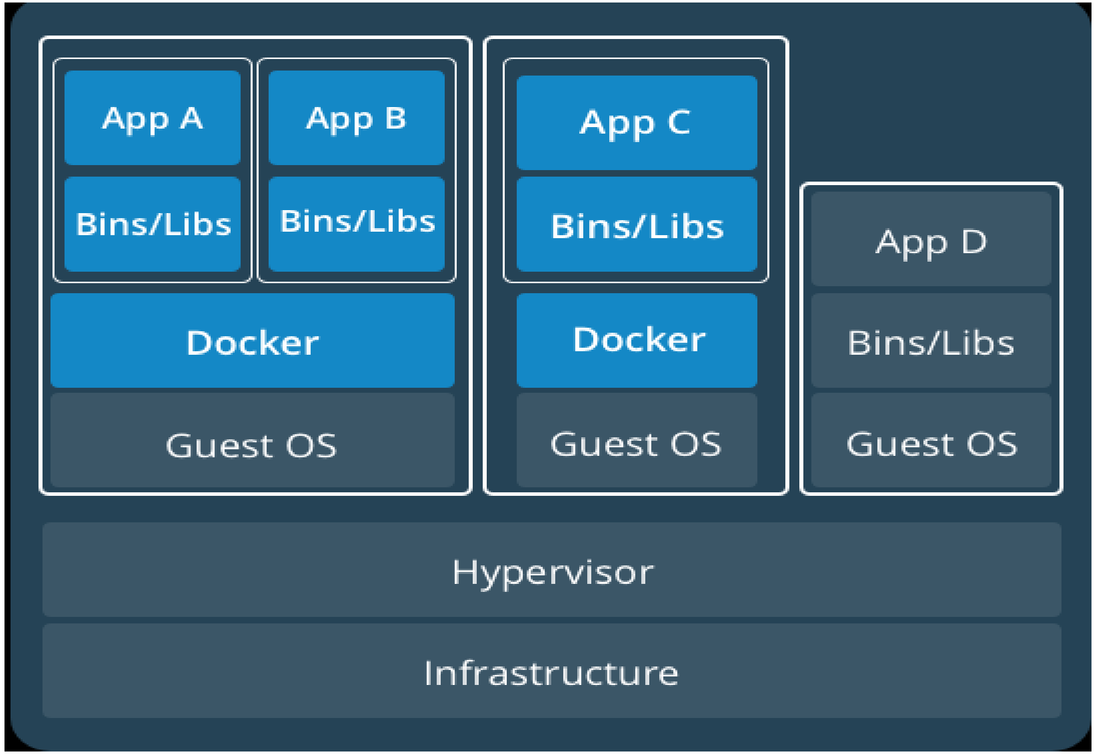
Docker containers are lightweight by design and ideal for enabling microservices application development. Accelerate development, deployment and rollback of tens or hundreds of containers composed as a single application. Whether building new microservices or transitioning monoliths to smaller services, simple to use tools make it easy to compose, deploy and maintain complex applications.

Source: https://www.docker.com/what-docker

Containers on a Server


App
Dependency
App
Dependency
App
Dependency
App
Dependency
File System, Shared Libraries, Binaries, Application on System
Operating System (Kernel)

Containers and Virtual Machines



Source: https://www.docker.com/sites/default/files/containers-vms-together.png

Docker Reference

Reference Sites:

- https://www.docker.com/what-docker
- https://www.docker.com/what-container

Hands-On Demonstration

Handson Demonstration

Docker availability

- Check the status of docker service

```
$ service docker status
$ service docker start
```

- Check the version of docker available

```
$ docker version
```

- Create a test container from hello-world image

```
$ docker run hello-world
```

Handson Demonstration

Working on Container

- Know the images available in your machine

```
$ docker image ls
```

- Create a container with interactive access (-it)

```
$ docker container run -it ubuntu:16.04
```

- List the containers

```
$ docker container ls
```

Handson Demonstration

Change the container environment

- Create a folder and file inside container and exit from container

```
:/# mkdir MASTER
:/# touch testfile
:/# exit
```

- Start and connect (attach) with the container again for interaction

```
$ docker container start 301a4c3c816c
$ docker attach naughty_brown
```

- Find that the environment inside the container is intact

```
:/# ls
```

Handson Demonstration

Replicating the Container

- Extract the image of container

```
$ docker container commit 301a4c3c816c
```

- Create another container from this extracted image

```
$ docker container run -it testimage /bin/bash
```

- Change container names

```
$ docker container rename naughty_brown original
```

Handson Demonstration

Docker Image Sharing as tarfile

- Export the image of a continer as a single tar file

```
$ docker container export -o test.tar original
```

- Import the image from the tar file

```
$ docker image import ./test.tar newimage
```

- Check whether the new container is replica of the former

```
$ docker container run -it --name replica2 newimage /bin/bash
```

Handson Demonstration

Docker image sharing via docker hub

- Create an account in docker hub: (https://hub.docker.com)
- Login into docker hub

```
$ docker login
```

- Change the image tag (Name) as required

```
$ docker image tag testimage prakaram/ti001
```

- Push the image to docker hub

```
$ docker image push prakaram/ti001
```

- Logout from docker hub

```
$ docker logout
```

Handson Demonstration

Docker image sharing via docker hub

- Pulling an image from docker hub

```
$ docker image pull docker/whalesay
```

- Creating a container

```
$ docker container run docker/whalesay cowsay "Always an Important Message"
```

```
 _____________________________
< Always an Important Message >
 -----------------------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
```

Handson Demonstration

Removing containers & images

- Removing a container

```
$ docker container rm replica2
```

- Removing all stopped containers

```
$ docker container prune
```

- Removing an image

```
$ docker image rm prakaram/ti001
```

- Removing all images not attached to any containers

```
$ docker image prune -a
```

Handson Demonstration

Docker subcommands

- Docker container specific subcommands

```
$ docker container
```

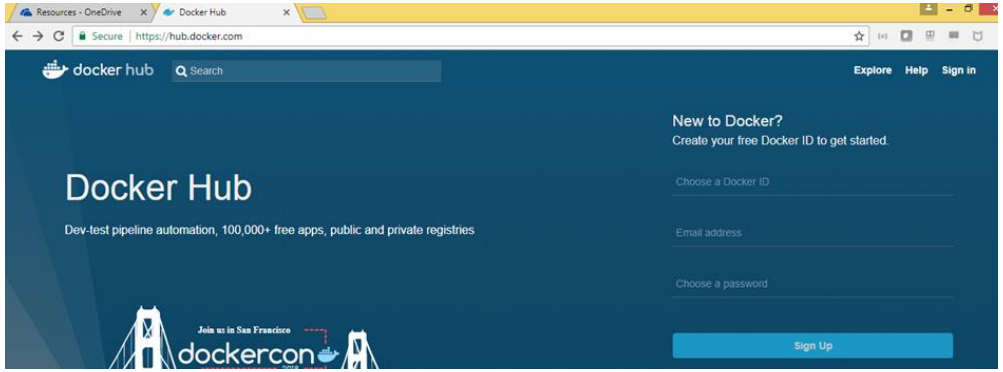
- Docker Image specific subcommands

```
$ docker image
```

- Old general list of docker subcommands

```
$ docker
```

Docker account creation and login screen



Resources - OneDrive
Docker Hub
Secure | https://hub.docker.com

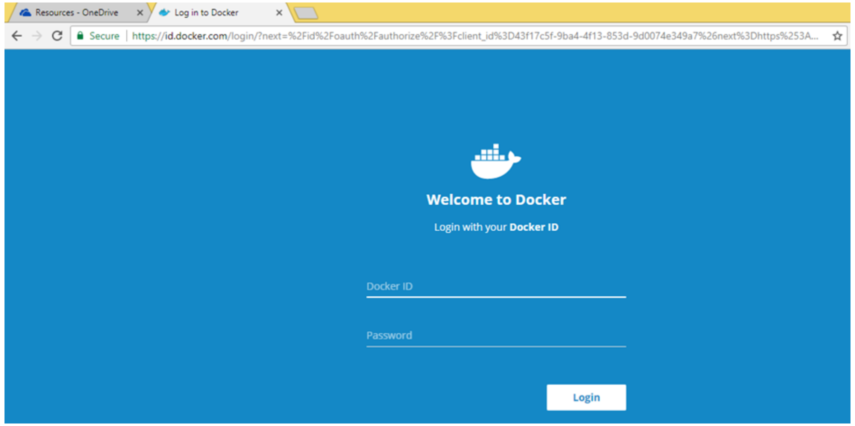
docker hub
Search
Explore Help Sign in

Docker Hub

Dev-test pipeline automation, 100,000+ free apps, public and private registries

New to Docker?
Create your free Docker ID to get started.

Choose a Docker ID
Email address
Choose a password
Sign Up



Resources - OneDrive
Log in to Docker
Secure | https://id.docker.com/login/?next=%2Fid%2Foauth%2Fauthorize%2F%3Fclient_id%3D43f17c5f-9ba4-4f13-853d-9d0074e349a7%26next%3Dhttps%253A...

Welcome to Docker
Login with your **Docker ID**

Docker ID
Password
Login

wipro

Thank You