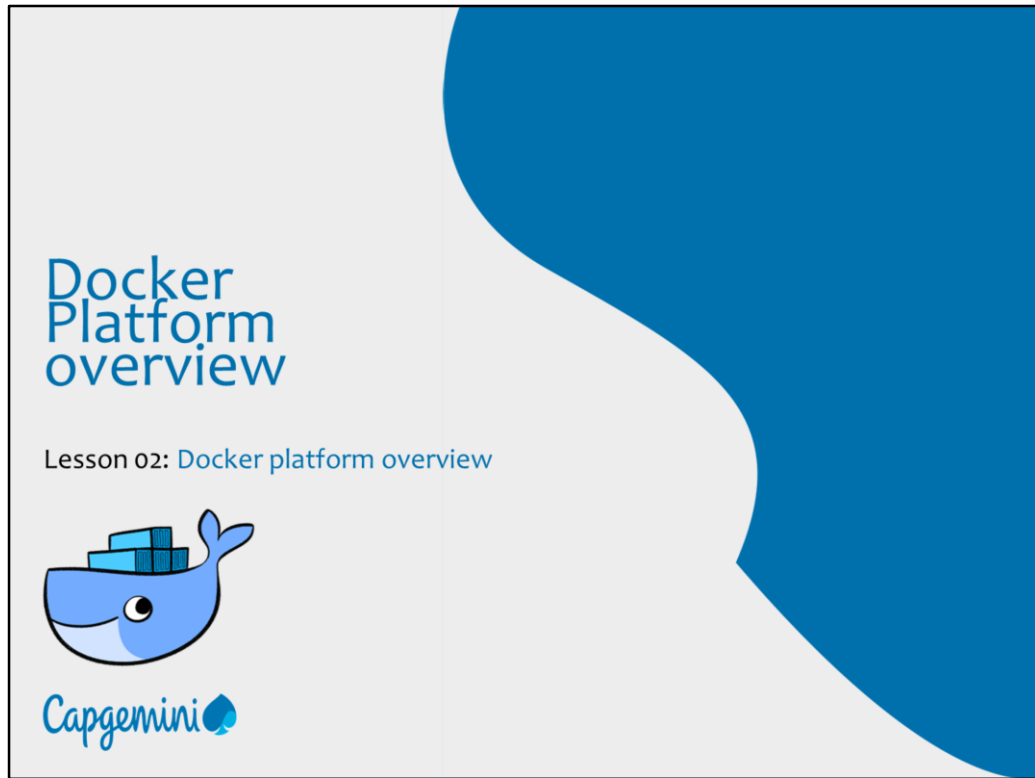


Instructor Notes:

Add instructor notes here.



Instructor Notes:

This lesson is to give an Introduction on Java Server Pages

Lesson Objectives



In this lesson, you will learn:

- Docker Platform
- Docker Engine
- Docker Images
- Docker containers
- Registry
- Repositories
- Docker Hub

Presentation Title | Author | Date

| © 2017 Capgemini. All rights reserved.

2

- Docker Platform
- Docker Engine
- Docker Images
- Docker containers
- Registry
- Repositories
- Docker Hub
- Deploying Jar file in a Docker and push the Docker image to Repository

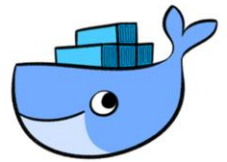
2.1: Docker Platform Docker Platform?



A container platform is complete solution that allows organizations to solve multiple problems across a diverse set of requirements.

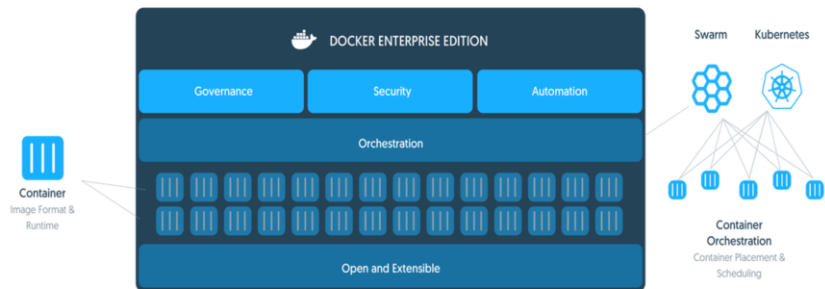
it delivers sustainable benefits throughout your organization by providing all the pieces an enterprise operation requires

- security,
- governance,
- automation,
- support and certification over the entire application lifecycle.



2.1: Docker Platform Docker Platform?

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.





2.1: Docker Platform Docker Platform Use cases?

MODERNIZE TRADITIONAL APPS :

- Packaging existing apps into containers immediately improves security, reduce costs, and gain cloud portability

Hybrid Cloud :

- Docker packages applications and their dependencies together into an isolated container can be run in any infrastructure(Cloud migration, multi-cloud or hybrid cloud infrastructure)

CONTINUOUS INTEGRATION AND DEPLOYMENT [DEVOPS] :

- Docker enables a true separation of concerns to accelerate the adoption of DevOps processes.

Microservices :

- Docker containers are lightweight by design and ideal for enabling microservices application development

Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

5

MODERNIZE TRADITIONAL APPS

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.

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. Docker certified infrastructure ensures the containerized applications work consistently.

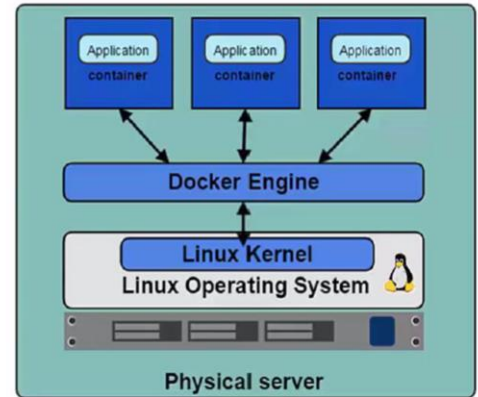


2.2: Docker Engine Docker Engine?

Docker Engine is a program which enables container to be built , ship and run.

it user Linux Kernel namespaces and control groups .

Namespace gives us the isolated workspace .

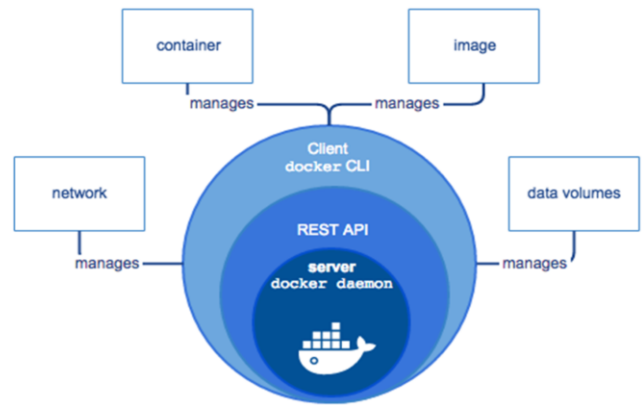


2.2: Docker Engine Docker Engine?



Docker Engine is a part of the Docker , which creates and runs the Docker container.

Docker container
is a live running
instance of a
Docker image .



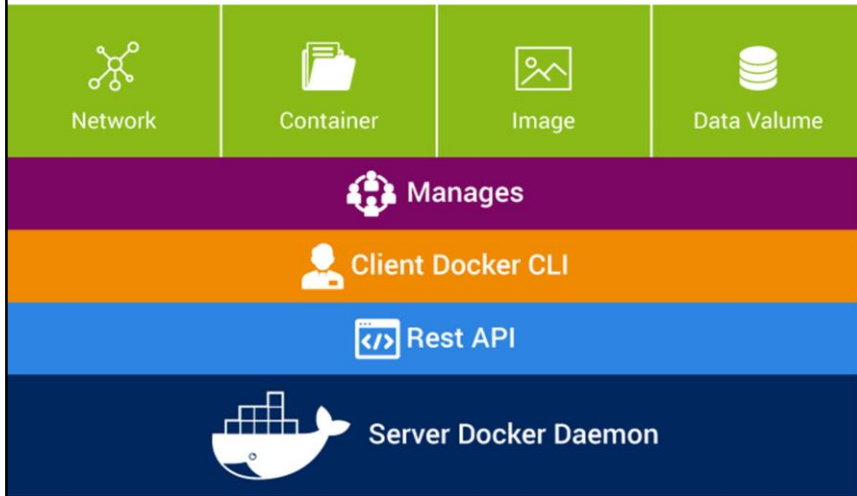
Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

7

The Docker engine is a part of Docker which create and run the Docker containers. The docker container is a live running instance of a docker image. Docker Engine is a client-server based application with following components -
A server which is a continuously running service called a daemon process.
A REST API which interfaces the programs to use talk with the daemon and give instruct it what to do.
A command line interface client.

2.2: Docker Engine Docker Engine?



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

8

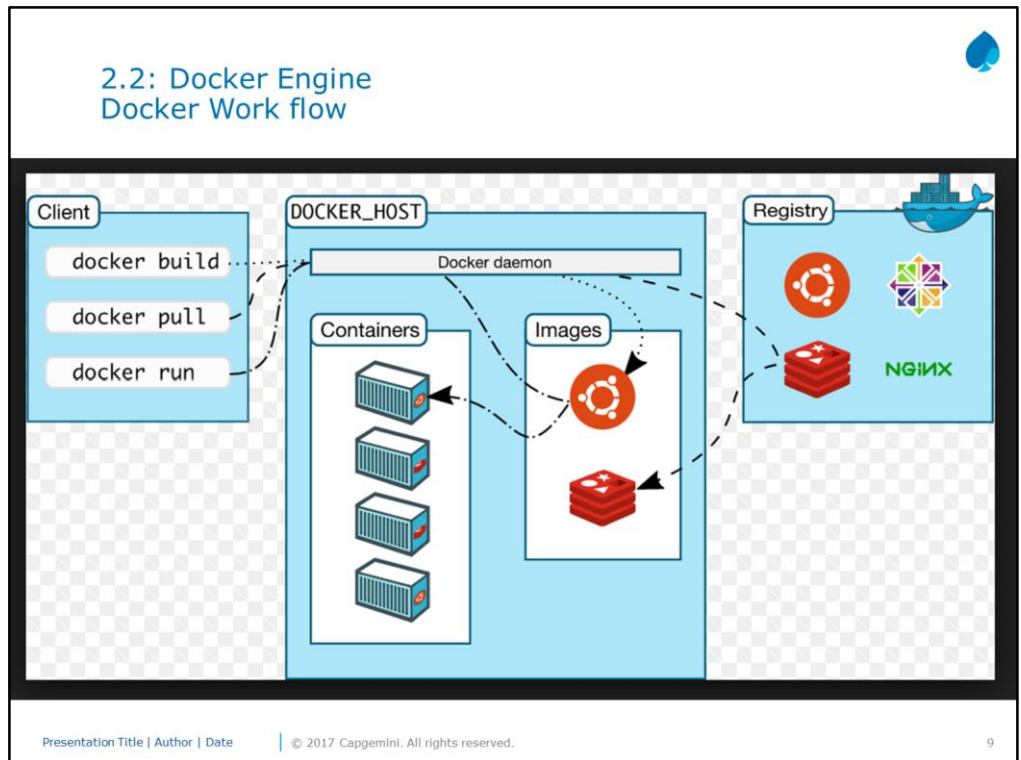
The command line interface client uses the Docker REST API to interact with the Docker daemon through using CLI commands.

Many other Docker applications also use the API and CLI.

The Docker daemon listens to only Docker API requests and handles Docker images, containers, networks, and volumes.

It also communicates with other daemons to manage Docker services .

In Docker, client can interact with more than one daemon process.

Instructor Notes:

The command line interface client uses the Docker REST API to interact with the Docker daemon through using CLI commands. Many other Docker applications also use the API and CLI. The Docker daemon listens to only Docker API requests and handles Docker images, containers, networks, and volumes.

It also communicates with other daemons to manage Docker services .

In Docker, client can interact with more than one daemon process.

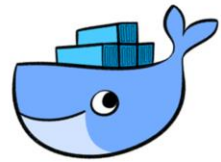
Instructor Notes:

2.3: Docker Images Docker Images



Docker Image :

- It is a Read only templates used to create container
- Build by you or other Docker user
- Stored in Docker Hub or in your local Registry



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

10

Instructor Notes:

2.3: Docker Images Docker Images

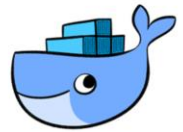


Docker Images are the most build part of docker life cycle.

We can build an image which is based on the centos image .

An image can be created based on another image, with some additional customization in the image.

We can create our own images or only use those created by others and published in registry directory.



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

11

We can build an image which is based on the centos image, which can install the Nginx web server with required application and configuration details which need to make the application run.

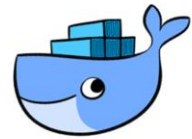
2.3: Docker Images

Steps to Docker Images



Follow the Steps :

- Dockerfile needs to be created with some syntax contains the steps that needed to create the image and make to run it.
- Each instruction in a Dockerfile creates a new layer in the image.
- For any modification in Dockerfile later on needed rebuild the image and the layers which have changed are rebuilt.



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

12

We can build an image which is based on the centos image, which can install the Nginx web server with required application and configuration details which need to make the application run.

Instructor Notes:

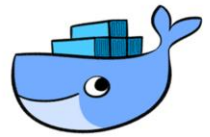
2.3: Docker Images DockerFile



It is a text file which consists of all commands so that user can call on the command line to build an image .

The docker file can be considered as the source code and images to make compile for our container which is running code .

The Dockerfile are portable files which can be shared, stored and updated as required.



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

13

We can build an image which is based on the centos image, which can install the Nginx web server with required application and configuration details which need to make the application run.

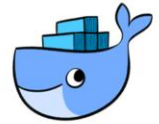
Instructor Notes:

2.3: Docker Images Docker File - Commands



Docker file instruction:

- **From** : Set the Base image for instruction
- **Maintainer** : - Shows Author of DOC file
- **Run** : Allows to execute the command
- **CMD** :- It shows the command , which are used in image
- **Label** :- It shows metadata(Key – Value pairs)
- **Expose** : listen on specific as required by application servers
- **ENV** : - set the environment variables in the Docker file for the container
- **COPY** : - copy file from source to destination
- **WORKDIR** : -- set the current working directory for the other instruction, i.e., RUN, CMD, COPY, etc.



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

14

Some of the docker files instruction is -

FROM - This is used for to set the base image for the instructions. It is very important to mention this in the first line of docker file.

MAINTAINER - This instruction is used to indicate the author of the docker file and its non-executable.

RUN - This instruction allows us to execute the command on top of the existing layer and create a new layer with the result of command execution.

CMD - This instruction doesn't perform anything during the building of docker image. It Just specifies the commands that are used in the image.

LABEL - This Instruction is used to assign the metadata in the form key-value pairs. It is always best to use few LABEL instructions as possible.

EXPOSE - This instruction is used to listen on specific as required by application servers.

ENV - This instruction is used to set the environment variables in the Docker file for the container.

COPY - This instruction is used to copy the files and directory from specific folder to destination folder.

WORKDIR - This instruction is used to set the current working directory for the other instruction, i.e., RUN, CMD, COPY, etc.

Instructor Notes:

2.4: Docker Container Docker Container

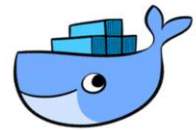


One can create , run ,stop or delete a container using Docker CLI

One can connect the Container to more than one network or even create a Image based on it's current state.

Docker container:

- is a live running instance of a Docker image .
- It is isolated application platform which contains every thing that needed to run your application based on one or more images
- It is isolated from other container and system machines



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

15

A container is the instance of an image. We can create, run, stop, or delete a container using the Docker CLI. We can connect a container to more than one networks, or even create a new image based on its current state.

By default, a container is well isolated from other containers and its system machine. A container defined by its image or configuration options that we provide during to create or run it.

Instructor Notes:

2.4: Docker Container Namespace



It is a service in Docker , provides isolated environment called as container.

When one run the Container , Docker creates a set of namespace for that container.

It provides a layer of isolation:

- Namespace PID :-- It provides isolation for the allocation of the process, lists of processes with details
- Namespace network :-- isolates the network interface controllers, IP tables firewall rules, routing tables etc.



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

16

Namespace PID : - It provides isolation for the allocation of the process, lists of processes with details. In new namespace is isolated from other processes in its "parent" namespace still see all processes in child namespace

Namespace network isolates the network interface controllers, IP tables firewall rules, routing tables etc. Network namespaces can be connected with each other using the virtual Ethernet device.

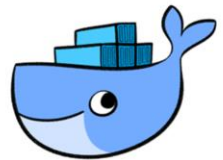
2.4: Docker Container Control groups



Docker Engine in Linux relies on named control groups , which limits the application to a predefined set of resources.

Docker Engine use the Control groups to share the available Hardware resources to containers.

Using control groups, one can define the memory available to a particular container.



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

17

Namespace PID : - It provides isolation for the allocation of the process, lists of processes with details. In new namespace is isolated from other processes in its "parent" namespace still see all processes in child namespace
Namespace network isolates the network interface controllers, IP tables firewall rules, routing tables etc. Network namespaces can be connected with each other using the virtual Ethernet device.

Instructor Notes:

2.4: Docker Container Union File systems and container format



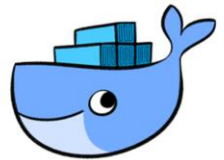
Union file systems is a file system which is used by creating layers, making them lightweight and faster.

Docker Engine using UFS provide the building blocks to containers.

Using control groups, one can define the memory available to a particular container.

Container Format :- It is a file which Docker Engine use to adds the namespaces, control groups & UnionFS .

The default size for the container is lib container.



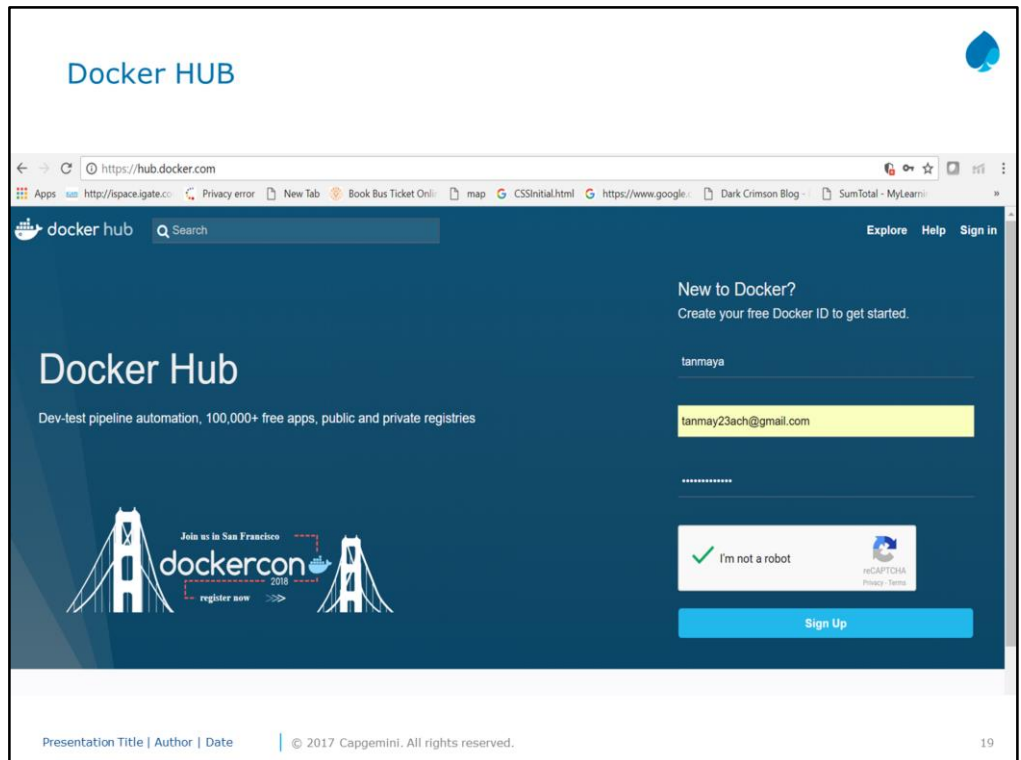
Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

18

Docker Engine uses many UnionFS variants some of including are AUFS, btrfs, vfs, Device Mapper, etc.

Instructor Notes:



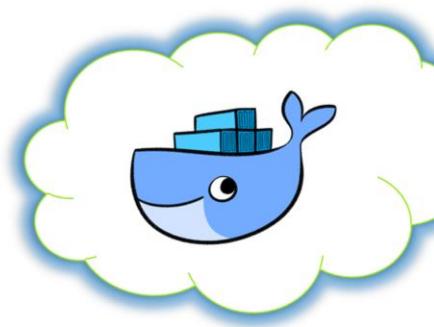
Docker Engine uses many UnionFS variants some of including are AUFS, btrfs, vfs, Device Mapper, etc.

Instructor Notes:

2.8: Deploying jar file in a Docker Demo

Demo on:

- Deploying Jar file into Docker and Push the image to Docker hub(Steps given in Note section)



Presentation Title | Author | Date
© 2017 Capgemini. All rights reserved.
20

Step 1 : Develop a simple Helloworld App and import the jar File to your EC2 instance and note the public IP address to connect .

Step 2: In your Window platform Use the puttygen and putty.exe files to connect to your Linux server instance in AWS .

Type the command to create an image and check the application deployment

Step 3 : Once the docker service started , Type below command

sudo usermod -a -

|

Instructo **G docker ec2-user**

Step 4: Type the command to check the images and container Id (Optional)

Step 5: Create a Dockerfile with below lines

```
FROM frolovlad/alpine-oraclejdk8:slim
ADD DockerHelloDemo.jar app.jar
ENTRYPOINT [ "sh", "-c", "java -jar app.jar" ]
```

Step 6 : Type the command to build an image

Docker build -t mydockerfstdemo .

Repeat Step 4 to check your image and container

|

|

Instruct

Type the command to run the Docker container

Step 7: **docker run mydockerfstdemo**

Step 8 : Now push the Docker image to Dockerhub repo . Type the command

```
export
DOCKER_ID_USER="tanmay23"
docker login
docker tag
mydockerfstdemo
$DOCKER_ID_USER/fstdockerdemo
docker push
$DOCKER_ID_USER/fstdockerdemo
```

Step 9: **Login to dockerhub repo and see your Docker.**

|

Instructor Notes:

Summary



In this lesson, you have learnt:

- Introduction to Docker
- How Docker provides Solution in SW or Application impl
- Limitation of VM
- What is Containerization
- The Differences between VM and Container
- Docker Architecture
- Docker community



Presentation Title | Author | Date

© 2017 Capgemini. All rights reserved.

21

Instructor Notes:

Answers for the Review Questions:

Answer 1: Docker engine

Answer 2: Docker container

Review – Questions



Question 1: _____ is a part of the Docker , which creates and runs the Docker container Docker container is light weight then VM.

Question 2 : _____ is a live running instance of a Docker image

