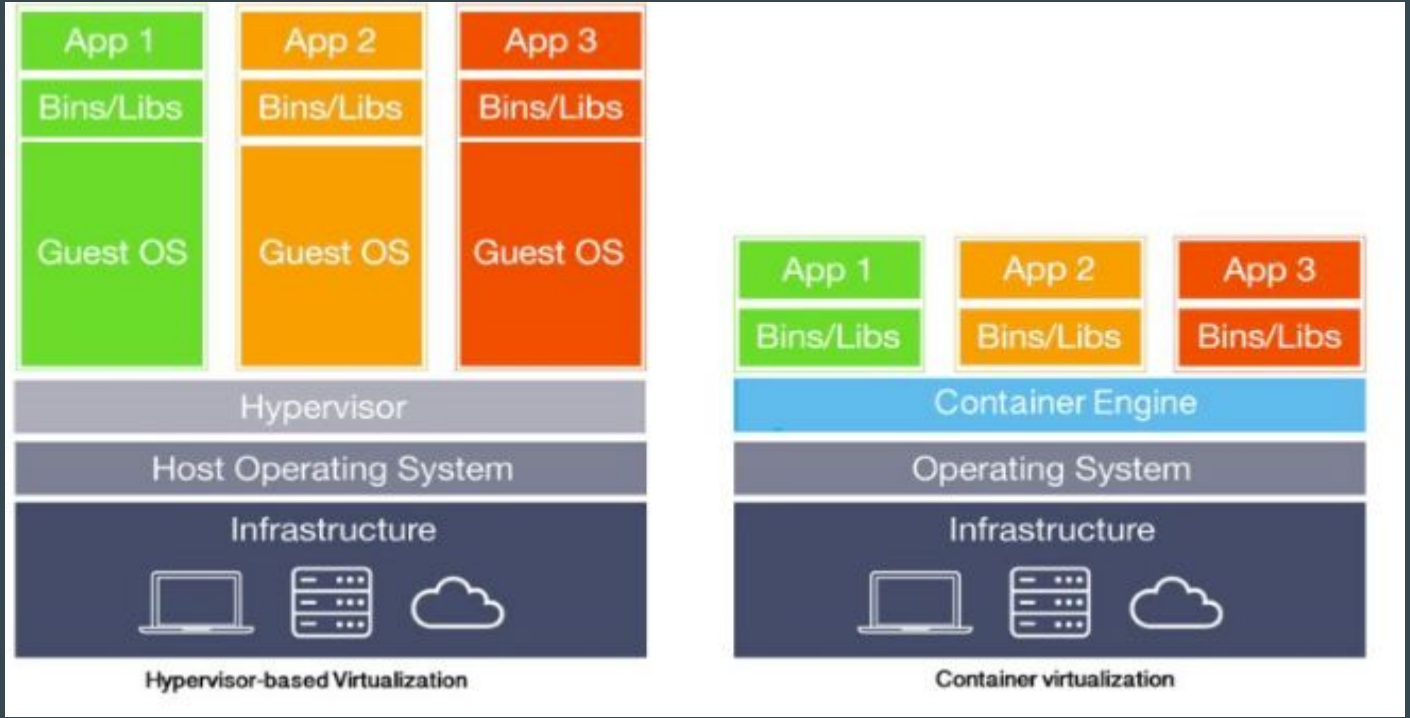
DOCKER

...

DOCKER

- Docker the container runtime and orchestration technology.
- Docker an open source project (this is now called Moby), generated the most interest in container technology in the past few years.
- A command line tool that made creating and working with containers easy for developers and administrators.
- Docker is software that runs on Windows and Linux.
- It creates, manages and orchestrate containers.

Containers VS VM:



CONTAINERS VS VM

- Each virtual machine runs a unique guest operating system
- Each VM has its own binaries, libraries, and applications
- **Container** systems usually provide service isolation between containers.
- Containers provide a way to run these isolated systems on a single server or host OS.
- Containers sit on top of a physical server and its host OS. Containers are only megabytes in size and take just seconds to start, not like VM.

INTRODUCTION TO CONTAINERS

- **Container technology**, also known as just a **container**, is a method to package an application.
- Any application can be bundled in a container can run without any worries about dependencies, libraries and binaries.
- Container creates the isolated environment with all the required dependencies, libraries and binaries to run your application without any issue.
- The application can run in any environment.

INTRODUCTION TO CONTAINERS

- A container is a standard unit of software that packages up a given code and all its dependencies so the application runs quickly and reliably from one computing environment to another.
- **Containerization** is a lightweight alternative to a virtual machine that involves encapsulating an application in a container with its own operating system.
- Containerization is the process of bundling your application code with requires packages/libraries required at runtime to execute your application quickly and reliably in any supported computing environment

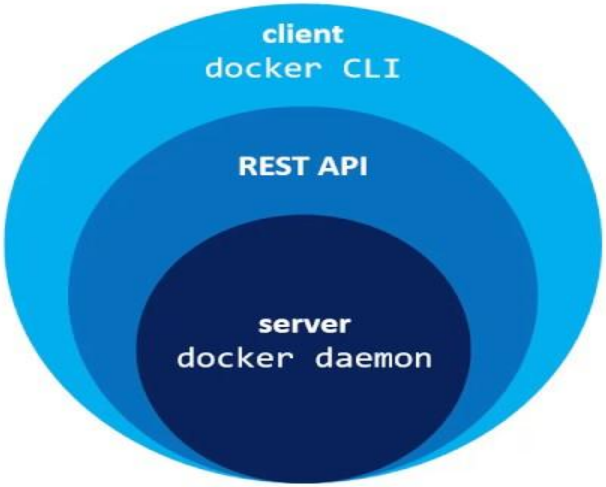
INTRODUCTION TO CONTAINERS

Monolithic applications are proved to be hard maintained, maintaining and CI/CD of such applications is time and energy intensive.

Containerization offers the following benefits:

- Portability of distributed applications
- Reproducibility of the application
- Scaling based on requirements
- Lifecycle management of containers
- Memory, CPU, and storage efficiency compared to VM hosting and hence cluster improvisation

Visualizing Docker's Architecture



Client -> Server Architecture

Visualizing Docker’s Architecture

- **The Docker Daemon**
  - Docker daemon is a service that runs on your host operating system.
- **Docker Daemon REST API**
  - The Docker daemon itself exposes a REST API From here, a number of different tools can talk to the daemon through this API.
- **Docker CLI**
  - The most widespread tool is the Docker CLI. It is a command line tool that lets you talk to the Docker daemon. When you install Docker, you get both the Docker daemon and the Docker CLI tools together.

Images

- A Docker image is an object that contains an OS file system and an application.
- A container **image** is an inert, immutable, file that's essentially a binary packaged snapshot of a **container**.
- An image is the application we want to run.
- It is like a virtual machine template.
- You can think of an image as a class.

Docker Installation

- **Official Ubuntu Repositories**
  - $ sudo apt-get install docker.io
- **Another Way TO install Docker from Official Site**
  - https://docs.docker.com/install/linux/docker-ce/ubuntu/
- **Verify the installation**
  - $ sudo docker -v

DOCKER

- **Verify that Docker CE is installed correctly by running the hello-world image**.
  - $ sudo docker run hello-world
- **CHECK IMAGES:**
  - $ sudo docker image ls
  - $ sudo docker images

DOCKER COMMANDS`

- docker image pull ubuntu:latest
- docker images or docker image ls
- docker container run -it ubuntu:latest
- ps -elf
- touch myfile.txt
- pressed Ctrl-PQ
- docker container ls (copy the container name)
- docker exec -it <container-name> bash

DOCKER COMMANDS

- Stop the container and kill it using the docker container stop and docker container rm commands.
  - docker container stop <container-name>
  - docker stop $(docker ps -a -q)
  - docker container start <container-name>
  - docker container rm <container-name>
- Verify that the container was successfully deleted by running another docker container ls command.
  - docker container ls

To Build Docker Image:

Create index.html file

Hello EveryOne.

To Build Docker Image:

- **Create Dockerfile**
  - Add the following Instructions in Dockerfile:
  - **FROM** instruction to set the application's base image.
    - FROM nginx:alpine
  - **COPY** files from a specific location into a **Docker** image.
    - COPY index.html /usr/share/nginx/html/index.html

To Build Docker Image:

- **BUILD IMAGE:**
  - The "-t" flag adds a tag to the image so that it gets a nice name and tag.
  - At the end in the below command “.” which tells Docker to use the Dockerfile in the current directory.
    - docker image build -t <image-name> .

DOCKER

- **CHECK IMAGES AGAIN:**
  - $ sudo docker image ls
- **Run IMAGE:**
  - Get Image Name from above command.
    - $ sudo docker container run -p=8080:80 <image-name>
  - To run the container in the background
    - $ sudo docker container run -d --name <container-name> -p=8080:80 <image>

DOCKER-HUB

- Create account on Docker-hub.
  - https://hub.docker.com/
- Login to docker hub.
- Now connect your machine to docker hub using this command:
  - $ sudo docker login --username=yourhubusername --email=youremail@company.com
- Create repository on docker hub.

DOCKER-HUB

- Check the image ID using:
  - $ sudo docker images
- Tag your image using:
  - $ sudo docker tag <image-id> yourhubusername/repository-name:tag
- Push your image to the repository you created.
  - $ sudo docker push yourhubusername/repository-name:tag

DOCKER COMMANDS

- docker image ls
  - To get the list of images
- docker pull
  - Pull an image from the registry.
  - docker pull <image-name>
- docker run
  - docker run command to launch a container.

DOCKER COMMANDS

docker container run -it --name os1 ubuntu:latest bash

exit

docker exec -it os1 bash

docker start os1

docker exec -it os1 bash

docker volume create <volume-name>

docker volume ls

docker run -d -p 84:80 -v <vol-name>/:/usr/share/nginx/html/ 22061996/hello-world:v5

docker run -d -p 84:80 -v ~/Desktop/cnc/:/usr/share/nginx/html/ 22061996/hello-world:v5

DOCKER Volume

docker run -d -p=8080:80 -v prac:/usr/share/nginx/html --name prac 22061996/hello-world:v5

THANKS