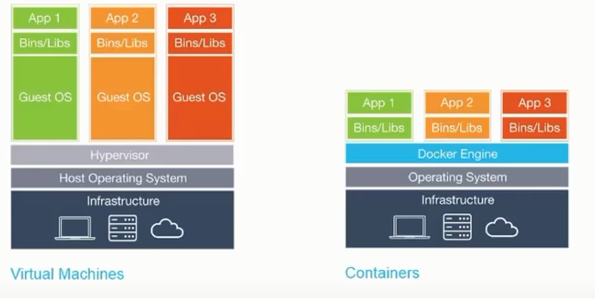
Execute automation 🡪 Docker

Introduction to Docker

Docker is a software containerization platform, it provides abstraction of OS level virtualization.

In Docker world everything is a container. It hold everything required to run a application including run time environment, libraries, dependencies and the application code itself and this is all build on a base image which is generally a scaled down version of a distro.



Docker engine helps the apps to leverage the host OS (kernel) , it is used in running these containers

With containers we do not pre-define the resources and there would be on OS which would run these containers in an isolated process at the container level however talking about the physical ram and hard disk isolation is not that good.

Docker solves the biggest problem of dependency hell because the container would be the same for everyone so you build once and can ship it anywhere and run it multiple times.

Understanding the ABC of Docker

Containers are isolated place where you run your application and it is extremely portable, in case if you were inside a container it would eventually looks like a VM however the container is leveraging the host kernel.

Docker hub is the public repo to store your images. Many of the normal application are already build and they are stored in form of images at the docker hub

You can use Kitematic in order to use the GUI feature and you can see all the images from the docker hub and run these images

You have to be signed in to Kitematics in order to use it., In Kitematics if you click on create that should create a container from that image.

Docker has a versioning system enabled where in in case you update the changes for a container and you can commit these changes and create a new image out of that commit and then these changes would be persistent

You can also roll-back using the VCS system of docker.

Docker also implements incremental uploads and downloads, in case you already have an image in the docker hub and after committing you push those changes back out to the docker hub

In such scenario it would only send the changes.

In Docker in case you already have an image created with some functionalities built in you can use that image as a base image in the docker file in order to create another image and create a new container

What is the benefit of this ?

Well you don’t have to built the features again that you developed for the first image.

Windows server 2016 is going to support containers.

Installing and Getting started with Docker for Windows.

Note :- In case you want to push more than one image in the hub you have to pay, pushing one image to the hub is free

Docker cloud is a space where you would store your private images.

Docker compose :- this is a file which is used to define how multiple containers can be run at the same time, in the compose files you would have images and these images would be linked with each other

# docker-compose up <$docker-compose.yml> will bring up all the containers in the docker-compose.yml file

Note :- Docker machine is the VM that is installed using virtual box and it is basically ubuntu loaded on the OS for the older OS as older windows OS does not support containers.

Working with Docker containers and commands

Docker can be executed via CMD or powershell in windows & Terminal in case of MAC and Linux distro.

In windows all these commands are executed in a tiny little linux distro sitting at the hyper-V

And it also automatically take cares of the network connectivity from the hyper-v machine

Famous Commands

# docker --help 🡪 Help on different commands on different flags and arguments

# docker create 🡪 Create a container from an image

# docker pull 🡪 Pull the images fom the docker hub

# docker push 🡪 Push the images from your local machine to the docker hub or your own private repo

# docker ps 🡪 List you all the running containers

# docker ps -a 🡪 all the containers that have exited or running conatiners as well

# docker rm 🡪 remove the container

# docker stop 🡪 Stop a running container, Removing a running container is not allowed. First you have to stop the container and then remove it.

# docker images 🡪 Lists all the images that are stored locally on the machine.

Note :- Since now windows has started supporting containers there is no need for a Machine to be installed at the Hyper-V and it can be done directly on the host OS

However, if you want to switch to LxC you can do that in order to do that you get an option switch to LxC.

As soon as you do that you will find a MobyLinuxVM installed over on your Hyper-V manager.

If you go ahead and check the network adaptor for this MobyLinux you will find that there is a network adaptor called DockerNat. And this DockerNAT Adaptor will be an internal switch

You can check the same via the Virtual Switch manager and it would show as internal which comes out of the box automatically.

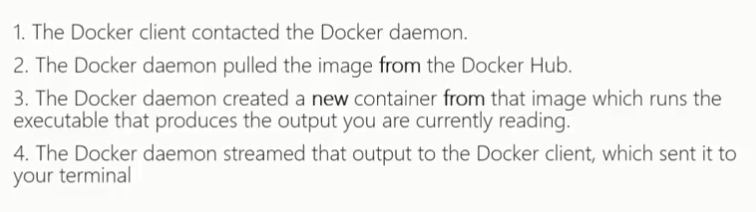
Note :- -p flags maps the host port to the exposed port of the container port. & -P would mapped the exposed port of the containers over random ports at the host

Pulling and Working with images from Docker Hub

In order to check what all operations was done on a container

# docker logs <$container-id>

Let’s understand the behind process



Linking and working with multiple containers to perform a single operation.

Links allow docker container to discover each other and securely transfer information from one container to another.

When you create a link you create a conduit between a source container and a recipient

Syntax :-

# docker run --name “Container name” –link (Name-of-the-container:image) -d (current-image that you want to run as a container and link)

Note :- In order to pass environment variable while creating the container from an image you can do that by using (-e) flag.

Docker Compose

Compose is a tool for defining and running multi-container Docker applications. With compose you create a compose file which has all the configuration related to all containers and then you run one single command and that would start all the containers.

Docker compose it’s easy to check the status and logs of a running container or all the containers.

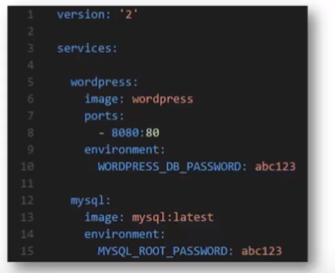
# docker-compose <$logs> 🡪 will show the logs of all the container

# docker-compose <$logs> <$Name-of-service> 🡪 Will show the logs of the specific container running the service

# docker-compose stop 🡪 Would stop all the containers

Docker compose file is a configuration file where you would define all the different configuration as you do with the CLI

It is a .YML file



In order to run a docker file you should navigate to the directory where you have docker-compose.yml file and run # docker-compose up

In case you want to mention the docker compose file which is somewhere else we use -f flag with the directory and the name of the file.