**INTRODUCTION TO DEVOPS & DOCKER**

**DevOps Introduction:**

**SDLC Phase:**

* Requirement gathering
* Planning
* Design
* Coding
* Implementation
* Testing
* Deployment
* Maintenance

**What is waterfall?**

* First documented SDLC
* Introduced on 1970
* A ladder methodology

**Limitations:**

**Agile SDLC Model:**

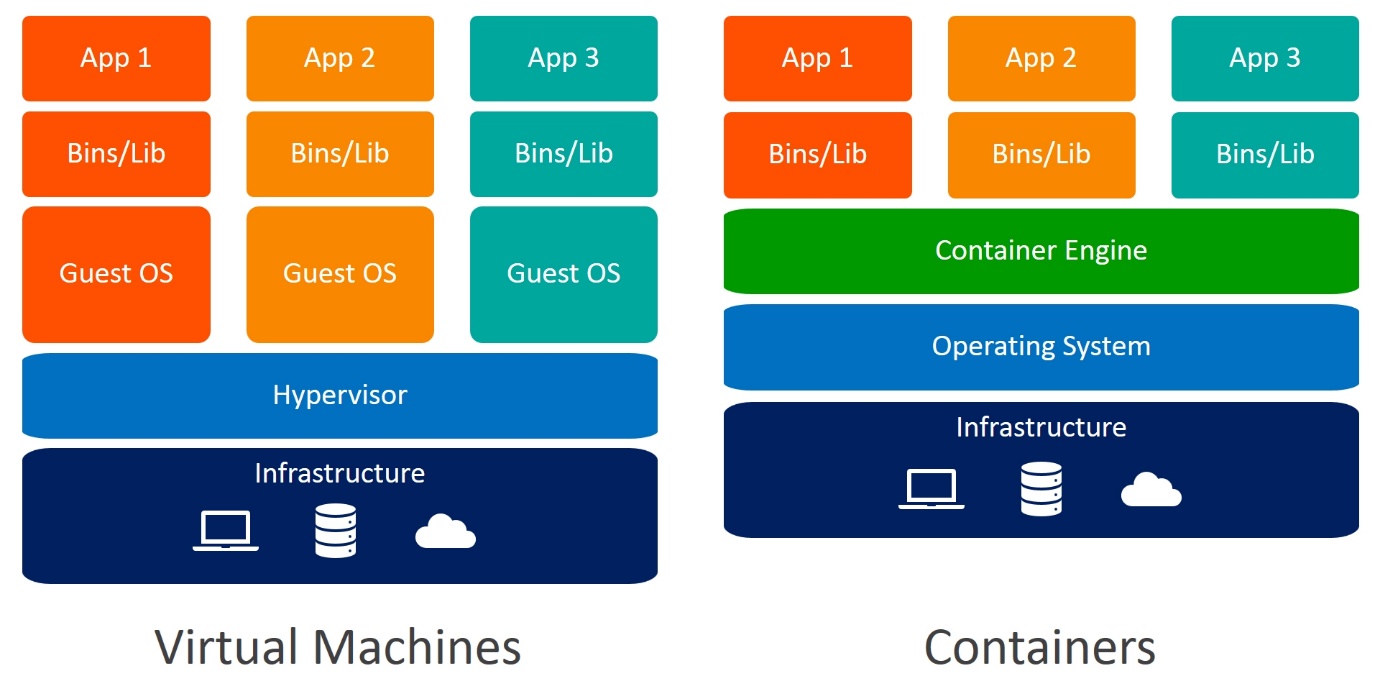
**What is Agile?**

* A combination of iterations and incremental process and mainly it focuses on process adoptability and customer satisfaction
* Divides the build into incremental builds and same will be provided to iterations
* 50% manual work and 50% tool

**DevOps Methodology:**

* 100% automation tools
* Is an enterprise software development life cycle means to have an agile relationship between Development and Operation teams and which acts like a bridge between both business units
* Automation of tools + AGILE

**Docker:**

* A containerization tool
* Docker is a software platform that allows you to build, test, and deploy applications quickly.
* Docker packages software into standardized units called [containers](https://aws.amazon.com/containers/) that have everything the software needs to run including libraries, system tools, code, and runtime.
* Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.
* Running Docker on AWS provides developers and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale.
* 

**How to create docker containers?**

* Create a new instance (say ‘Docker-Vinaya’)
* In security group inbound rules, add ALL TCP and Source-Anywhere IPv4-0
* Connect the instance with putty

**Installing docker:**

|  |
| --- |
| sudo su -  yum update -y  yum install docker -y  docker --version |

**Starting docker:**

|  |
| --- |
| service docker start |

**Pull image from docker hub:**

|  |
| --- |
| docker pull httpd |

**To list all images:**

|  |
| --- |
| docker images |

**To launch container from image:**

to launch container from image

-i = interactive

-t = terminal

-d = detached

-p = port

|  |
| --- |
| docker run -itd -p 80:80 httpd  docker run -itd -p 8080:80 httpd |

**To list all containers:**

|  |
| --- |
| docker ps |

or

|  |
| --- |
| docker container ls |

**Entering into the container:**

|  |
| --- |
| docker exec -it xxxx /bin/bash  xxxx-container id first four digits |

**To list all directories in the container:**

|  |
| --- |
| ls |

**Output: bin build cgi-bin conf error htdocs icons include logs modules**

**Go to htdocs directory:**

|  |
| --- |
| cd htdocs |

**To list all directories in htdocs:**

|  |
| --- |
| ls |

**Output: index.html**

**To display content in index.html:**

|  |
| --- |
| cat index.html |

**Output:**

**<html><body><h1>It works!</h1></body></html>**

**To change the content in index.html:**

|  |
| --- |
| echo "<html><body><h1>HTTP CONTAINER</h1></body></html>" > index.html |

**To exit container:**

|  |
| --- |
| exit |

**To stop container:**

|  |
| --- |
| docker stop xxxxxxxxxxxxxxxxxx yyyyyyyyyy zzzzzzzzzz |

xxxxxxxxxxxxxxxx, yyyyyy,zzzzzzzz-docker container ID

Now when the public IP address is open in brower with :8080

The changed content will be displayed

Similarly we can pull NGNIX, which is another image available in docker hub

|  |
| --- |
| docker pull nginx |

**How to push image to docker hub?**

**How to create a docker file?**

mkdir mydockerdir

ls

cd mydockerdir

ls -lrt

//touch Dockerfile

vi Dockerfile

i-insert

FROM httpd:latest //base image

MAINTAINER Vinaya

RUN yum update -y //command execution

esc:wq

**How to build :**

docker build -t myhttpd .

docker images

**To delete container:**

Docker rm -f

**To delete image:**

Docker rmi -f <image\_id>