Agenda

Containerization using Docker

Installing Docker, Virtualizing, VM vs Docker Image.

Revisit Day42

Jenkins (Build Automation Tool),

Sonar Qube, Sonar Lint and Sonar Cloud ( Code Quality Analysis Tool)

Jenkins – Is a Open Source, popular Build automation tool which supports different types of project.

Jenkins official Site – <https://Jenkins.io>

We can download and install Jenkins in all OS. It is available in Stand-alone (war) format and also available as a zip file for other operating systems. You can also download Jenkins as a docker image.

Latest version Jenkins expect JDK11.

Sonar Qube also expect JDK11. (make sure to add JDK11 to the top of the path environment variable)

Sonar Qube – Community Edition (Free Open source version) – Stand alone version (zip file)

The default port number of sonarqube is 9000 (username & password)

The default port number for Jenkins is 8080 (username & password)

Sonar Lint – Is a eclipse plugin to do code analysis (That can be added to Eclipse or STS using market place)

Docker –

Official Site : <https://docker.com>

Docker is the one of the DevOps tool which containerize the application along with all the dependencies and required libraries to run the code.

Docker will resolve the main challenge, “The code was perfectly running in my system”

If you want to Run a Java program

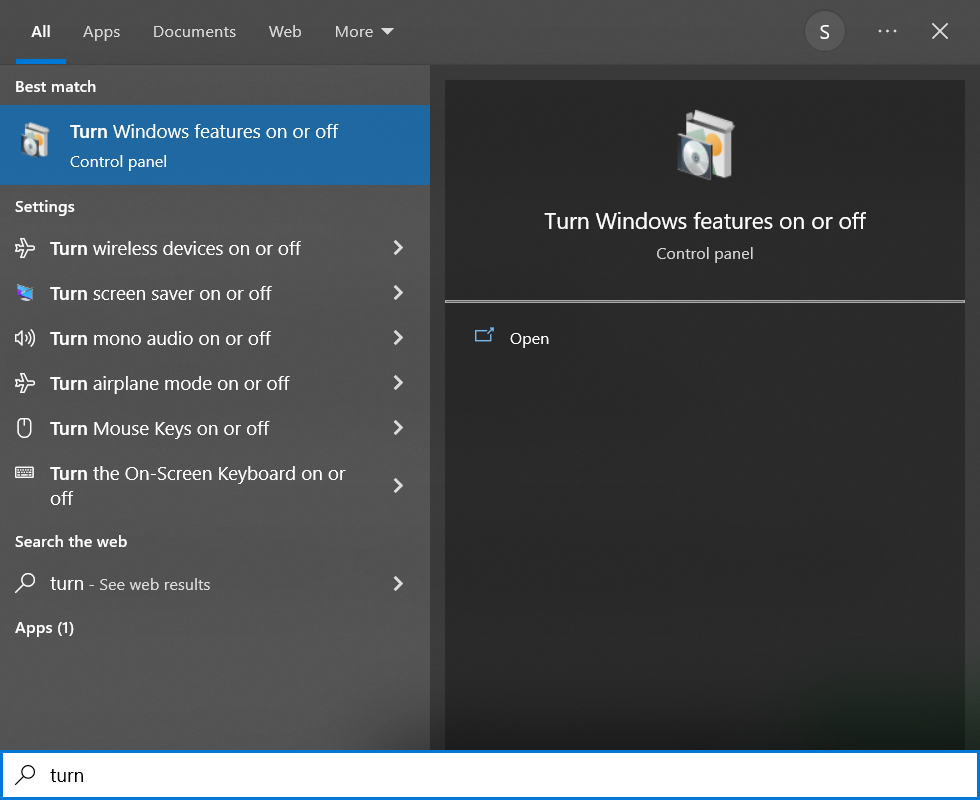
1. Java source code/ byte code
2. JVM – (JDK/JRE)
3. Libraries (3rd party library -mysql connector, spring framework jar files, springboot dependencies etc.,)

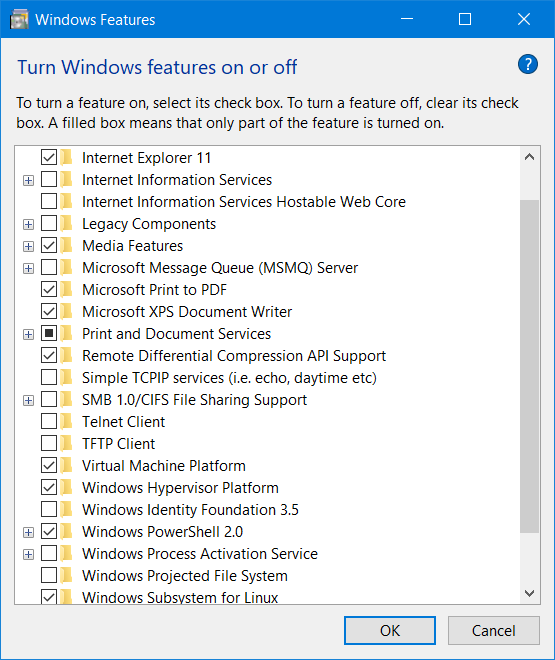
Docker will create a single file called as docker image, which is the bundle of source code along with the libraries, other dependencies and execution environment as well.

Hyper-V option should be enabled.

WSL-2 🡺 Windows Sub System for Linux.

In windows, search “Turn windows feature on or off”





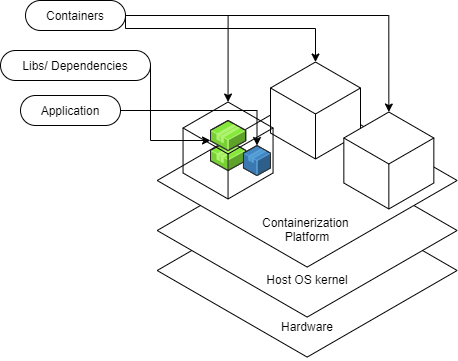
Download Docker-Desktop (Stand-alone Docker Client)

<https://www.docker.com/products/docker-desktop/>

Docker helps to containerize the application.

Containers – Applications packaged as a single unit along with all the dependencies

Containers – Apps running in an isolated sandbox. (No other libraries needed to run a container)



Docker is a Containerization platform which allows to run containers and also helps to create containers.

Using Docker

1. Create a container (Image file)
2. Run a Container (Running an image)

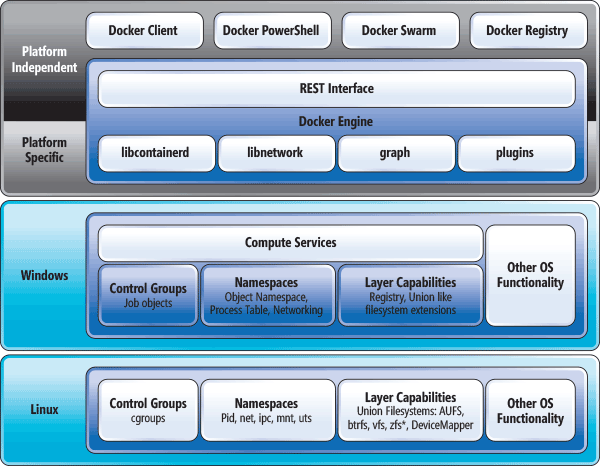
Images --- Class (MySQL, Oracle, Windows)

Container – Objects (Running version of an image)

Docker Architecture

Linux Containers

Windows Containers



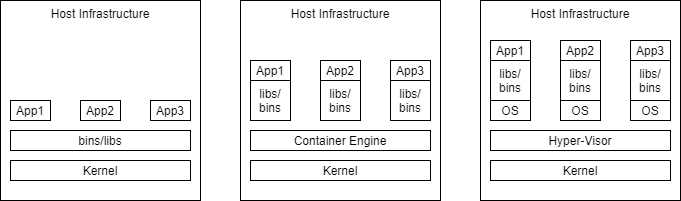
VM vs Containers

VM – Virtual Machines.

Currently in my laptop I’m using Windows Operating System.

Installing another OS

1. Dual Boot (Install windows and linux side by side using different partitions). I can choose to load any one OS at the time of booting.
2. I can install Linux inside the Windows OS. (Virtualizing)



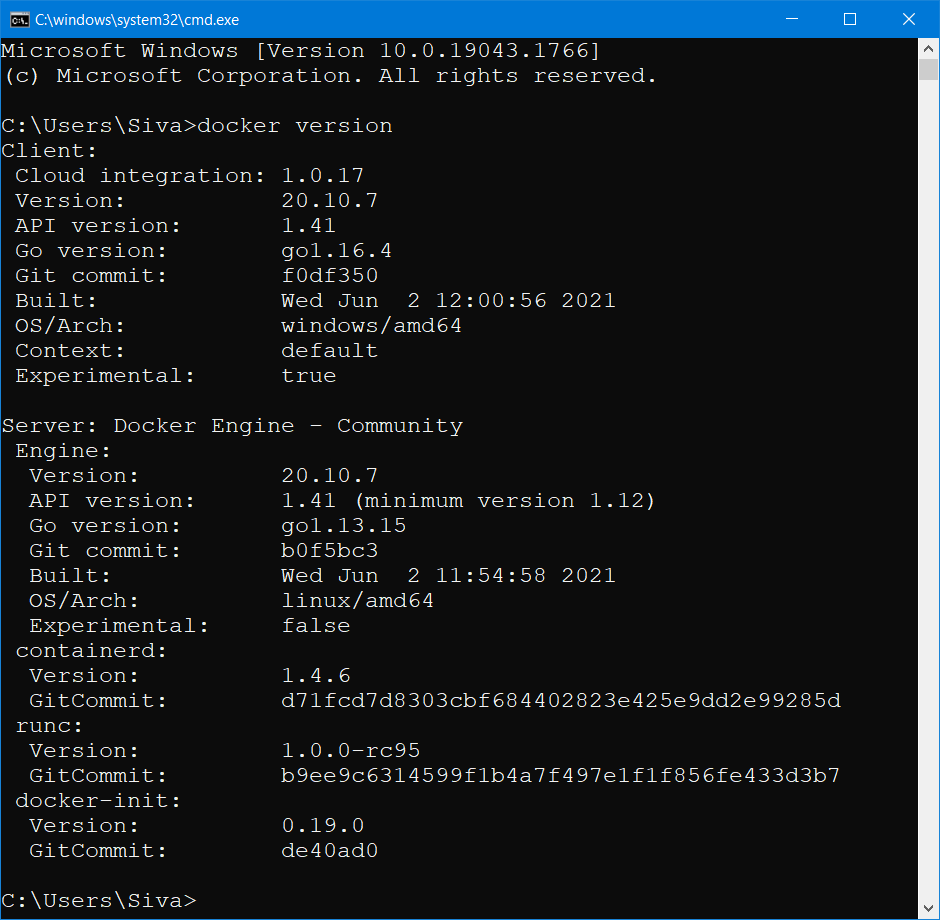
<https://www.javatpoint.com/docker-tutorial>

Docker Architecture

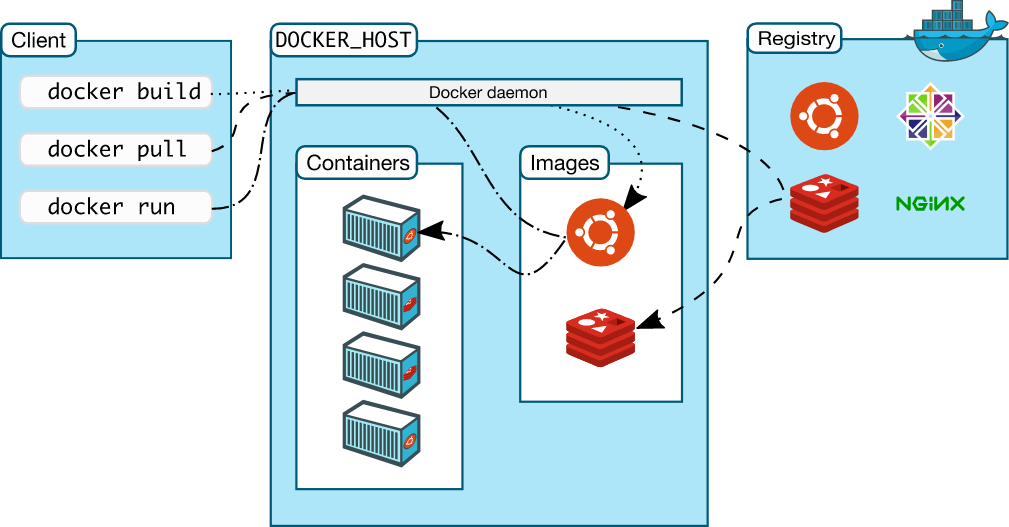
1. Docker Server or Docker Daemon
2. REST API – Docker Client (CLI/ GUI – Docker Desktop) – [Git]
3. Docker Hub (Cloud Repo) <https://hub.dockerhub.com> 🡪 <https://hub.docker.com> [Github]

Open command prompt

“docker version”



1. Docker Client [CLI client (docker commands to run in command prompt/ GUI client (Docker Desktop)]
2. Docker Server / Docker Engine/Daemon
3. Docker Registry (Docker hub)



<https://www.javatpoint.com/docker-java-example>

FROM java:8

COPY . /var/www/java

WORKDIR /var/www/java

RUN javac HelloWorld.java

CMD ["java", "HelloWorld"]

<https://www.amazonaws.cn/en/getting-started/tutorials/deploy-docker-containers/>

<https://aws.amazon.com/blogs/devops/build-and-deploy-docker-images-to-aws-using-ec2-image-builder/>

<https://medium.com/geekculture/docker-basics-and-easy-steps-to-dockerize-spring-boot-application-17608a65f657>

**Spring Boot Starter Project (Java Backend)**

1. Spring Web
2. Spring Data JPA
3. Spring Security
4. Spring Boot Dev Tools
5. Lombok
6. MySQL, Postgres, H2 (in-memory database)
7. OAS – Open API Documentation [Swagger v3]

Angular Front End

1. Login Page (Forgot Password)
2. Register/Sign-in Page
3. Landing Page (Different Landing page for different roles) Currently logged user name and signout button.
4. Use BootStrap/Material UI