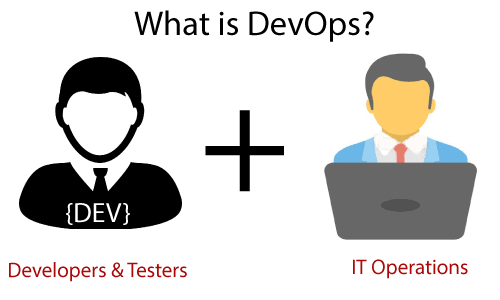
**What is DevOps?**



The DevOps is the combination of two words, one is **Development** and other is **Operations**.

It is a culture or process to promote the development and operation process collectively.

This allows a single team to handle the entire application lifecycle, from development to **testing, deployment**, **operations and monitoring**.

**Advantage of DevOps?**

DevOps helps to increase organization speed to deliver applications and services

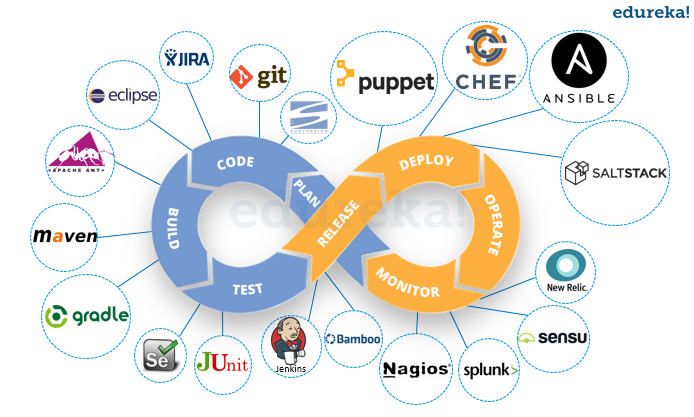
Development and IT operations with better communication and collaboration.

**Quality**, and **speed** of the application delivery has improved to a great extent

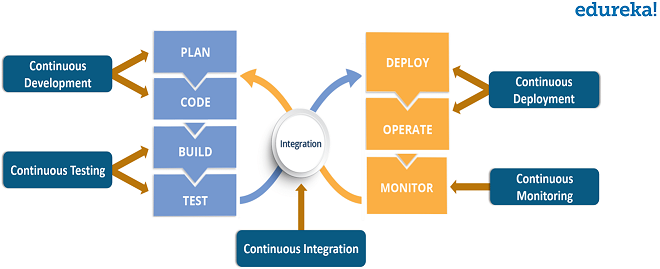
DevOps escalate business profit by decreasing software delivery time

It improves customer experience and satisfaction.

**Devops Lifecycle Phases?**



Plan – Business team and devops team to discuss goal, outcome, tools



**Continuous Integration:-**

Coding Stage - IDE

Build Stage – Maven, Gradle, Ant

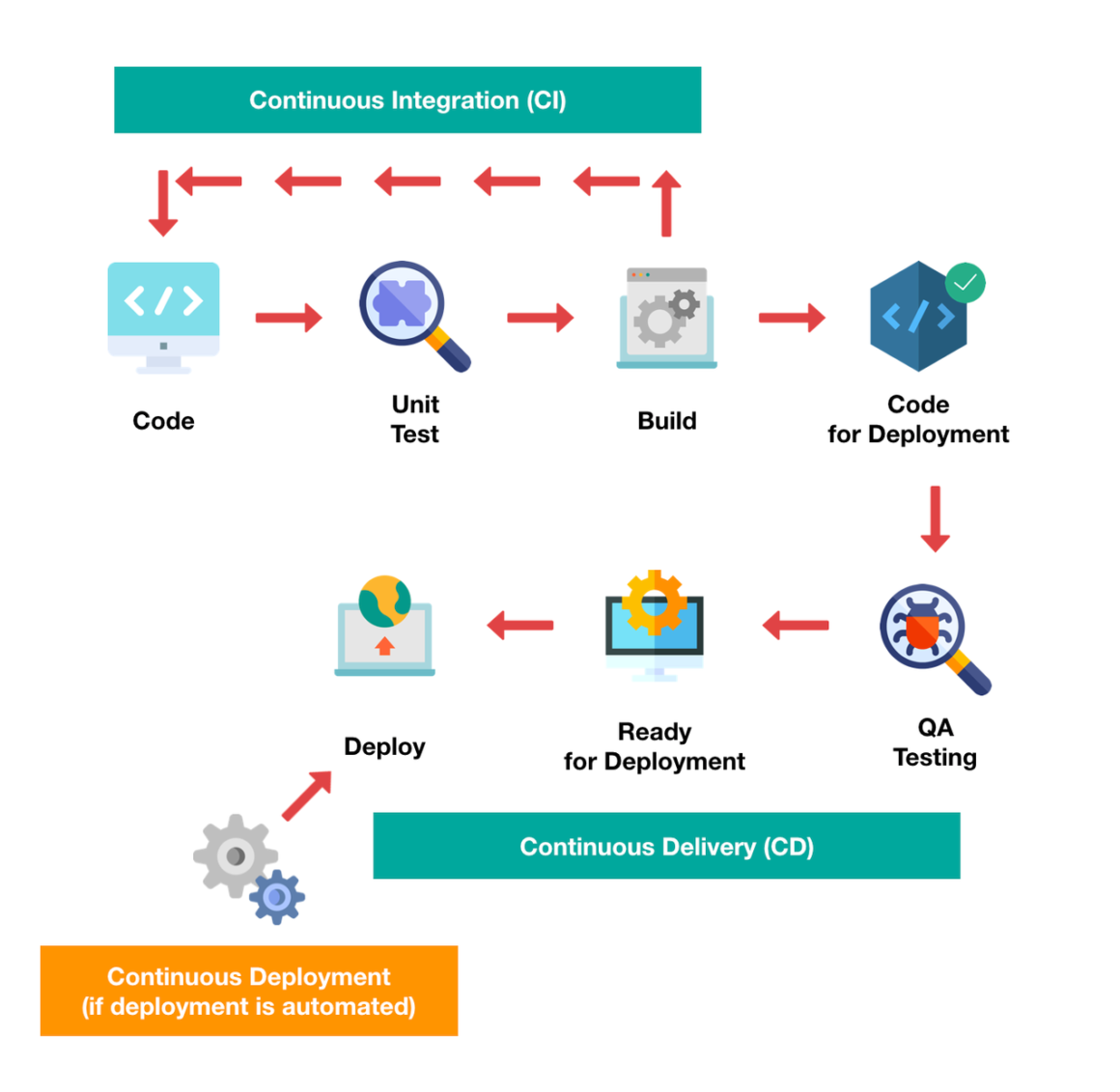
Testing Stage – Junit, selinum, TestNG , static code analysis – sonar qube

Developers practicing continuous integration merge their changes back to the main branch

The developer's made changes in the main branch then the build and running automated tests against the build will happen

By doing so, you avoid integration challenges that can happen when waiting for release day to merge changes into the release branch.

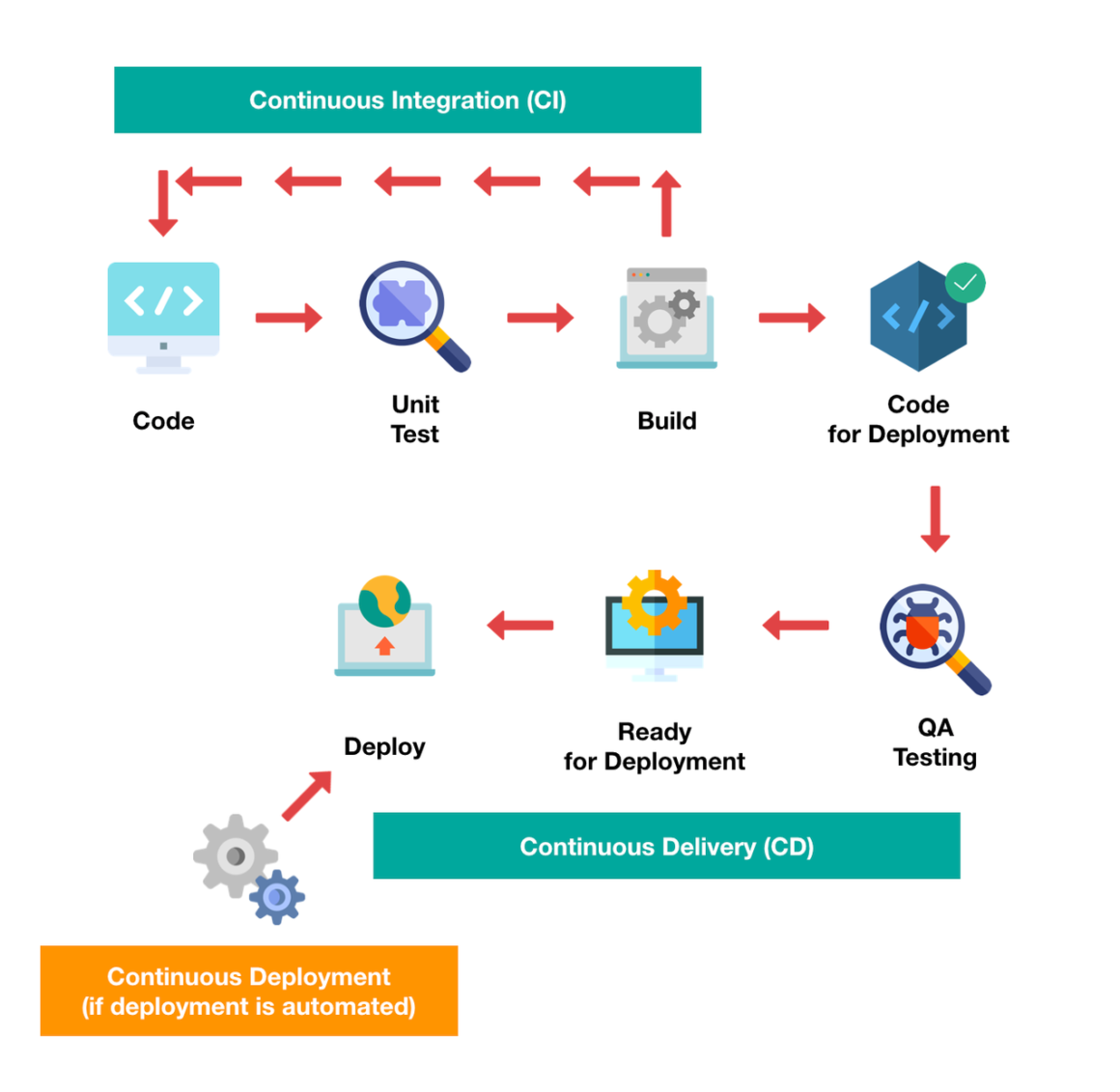
Continuous integration puts a great emphasis on testing automation to check that the application is not broken whenever new commits are integrated into the main branch.

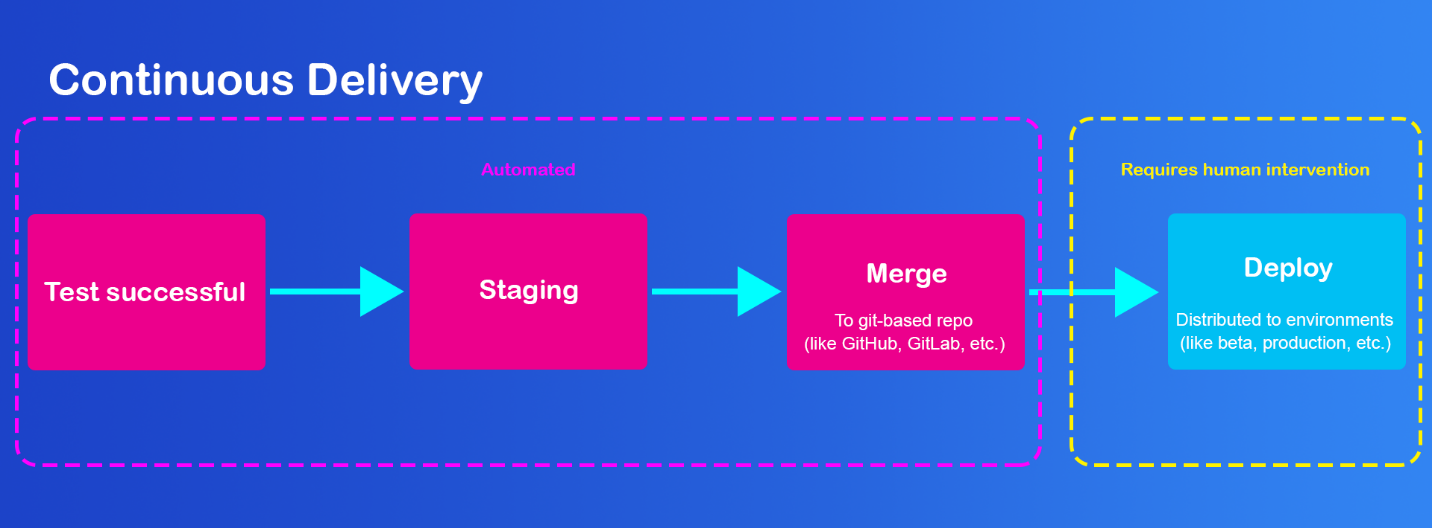


**Continuous Delivery**

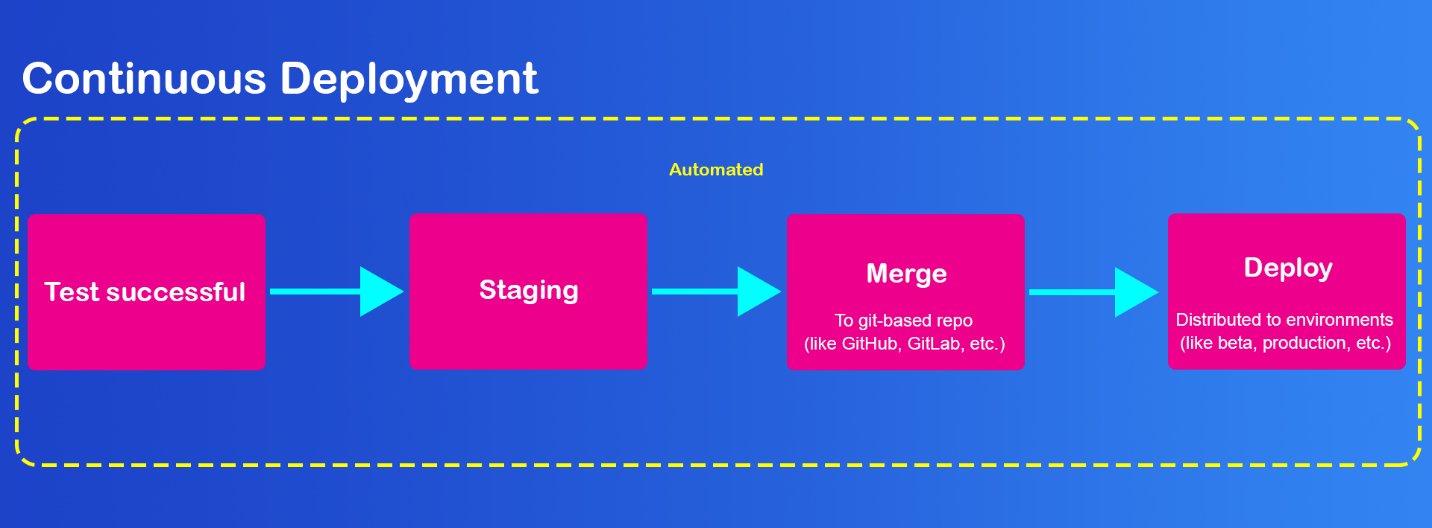
Monitoring Stage

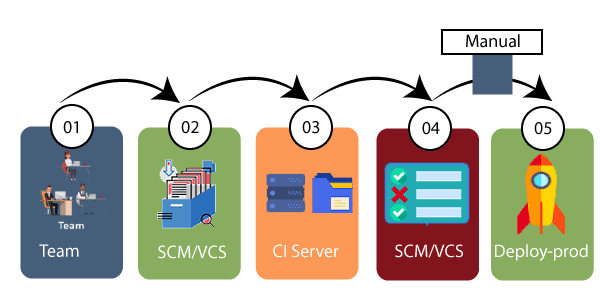
Deployment Stage





Continuous Deployment:-





Continuous Monitoring:

**What is Jenkins?**

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes.

Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

**Why Jenkins**

* Developers and testers use Jenkins to detect defects in the software development lifecycle and automate the testing of builds.
* They use it to continuously monitor the code in real-time and integrate changes into the build.
* Jenkins as it turns out, is a great fit for building a CI/CD pipeline because of its plugin-capabilities, and simple-to-use nature.

### **What are the features of Jenkins?**

Some of the crucial features of Jenkins are the following:

* It is a free and open-source automation tool
* Jenkins provides a vast number of plugins
* It is easy to set up and install on multiple operating systems
* Provides pipeline support
* Fast release cycles
* Easy upgrades

### **What are the ways to install Jenkins?**

Jenkins can be installed using -

Native System Package Manager like - apt (Linux), brew (Mac), etc.

Docker (popular docker images for Jenkins is available for different platforms like Unix/Mac/Windows in the docker registry)

Kubernetes (available as a helm chart and can be installed on our Kubernetes clusters)

### **What is a Jenkins job?**

A Job/Project is the fundamental unit of a work (like a software build, an automation task, test execution, etc) using the Jenkins automation server and other required plugins, configurations & infrastructures.

Jobs can be of different types like - a freestyle project, a multi-configuration project, a pipeline project, a multi-branch project, etc.

### **What is Jenkins pipeline?**

Creating a chain of jobs in Jenkins is the process of automatically starting the sequential job after one job is executed successfully.

Jenkins Pipeline, sometimes simply called Pipeline, is a set of plugins, which supports the integration and use of continuous delivery pipelines in Jenkins. The continuous delivery pipeline is the process where automated builds, tests, and deployments are planned and arranged as one release workflow. Simply put, it is the process the code changes will go through for delivering software from the version control to clients and users.

### **What is Jenkinsfile?**

A Jenkinsfile is essentially a text file containing the steps for running a Jenkins Pipeline and is registered into the source control repository of a project.

### **What are the types of Jenkins pipelines?**

Jenkins Pipelines can be either - a Declarative pipeline or a Scripted Pipeline.

**Declarative pipelin**e makes use of numerous, generic, predefined build steps/stages (i.e. code snippets) to build our job according to our build/automation needs whereas,

**Scripted pipelines**, the steps/stages can be custom-defined & used using a groovy syntax which provides better control & fine-tuned execution levels.

### **What are the ways to trigger a Jenkins Job/Pipeline?**

There are many ways we can trigger a job in Jenkins. Some of the common ways are as below

* Trigger an API (POST) request to the target job URL with the required data.
* Trigger it manually from the Jenkins web application.
* Trigger it using Jenkins CLI from the master/slave nodes.
* Time-based Scheduled Triggers like a cron job.
* Event-based Triggers like SCM Actions (Git Commit, Pull Requests), WebHooks, etc.
* Upstream/Downstream triggers by other Jenkins jobs.

Docker:-

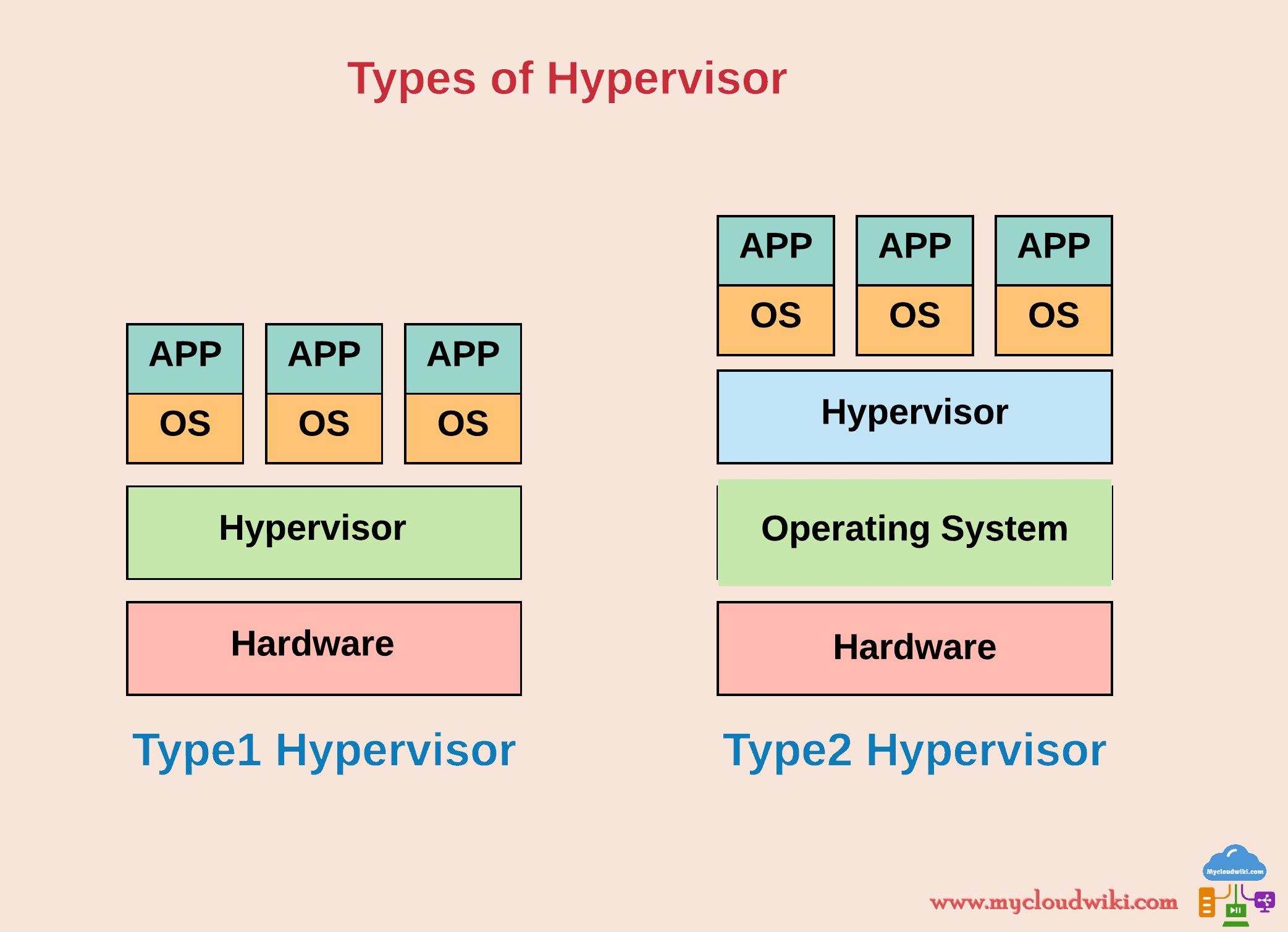
**What is Hypervisor?**

A software helps in making **virtualization** possible.

Its other name is **Virtual Machine Monitor**.

Its main task is to divide the host system. It also does the **allocation of resources** to the individually **divided virtual environments or virtual machines.**

The two types of Hypervisors are:



### **Type – I, Bare Metal Hypervisor or Native Hypervisor:** This type of hypervisor runs directly on your underlying host system. As it has access to your host hardware, it does not need any base server operating system.

### **VMware vSphere / ESXi, Microsoft Windows Server 2012 Hyper-V , Xen / Citrix XenServer, Oracle VM, KVM, Red Hat Enterprise Virtualization**

### **Type –II Hosted Hypervisor:** This hypervisor uses an underlying host operating system and thus the name, hosted hypervisor.

### **Oracle VM VirtualBox, VMware Workstation Pro/VMware Fusion, Windows Virtual PC, Parallels Desktop**

**What is Virtualization?**

Virtualization can be defined as a process by which we create a virtual, software-based version of anything such as servers, computer storage, applications, etc.

It can be done with just a physical single hardware system. A software named Hypervisor comes in use to split a single system into various different sections. These split sections, in turn, work like a distinct, separate individual system.

**What is containerization?**

Usually, in the software development process, code developed on one machine might not work perfectly fine on any other machine because of the dependencies.

This problem was solved by the containerization concept.

So basically, an application that is being developed and deployed is bundled and wrapped together with all its configuration files and dependencies.

This bundle is called a container. Now when you wish to run the application on another system, the container is deployed which will give a bug-free environment as all the dependencies and libraries are wrapped together.

Most famous containerization environments are Docker and Kubernetes.

**What is Docker?**

Docker is a containerization platform, which packages your application and all its dependencies together in the form of containers to ensure that your application works seamlessly in any environment, be it development, test or production.

Docker containers, wrap a piece of software in a complete file system that contains everything needed to run: code, runtime, system tools, system libraries, etc. It wraps anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment.

**What is containers?**

Containers are runtime instances of Docker images.

Any containerized application can run on any platform regardless of the underlying operating system

A Docker container is known as an executable package of the software, which contains all components essential to execute an application. It contains libraries, system tools, runtime, code, and settings.

**How to create an image**

Dockerfile

FROM openjdk:11

COPY ./target/icicibankapp.jar icicibankapp.jar

CMD ["java","-jar","icicibankapp.jar"]

Types of Images:-

Own images – by your

Predefined Images

Images can be stored in registry

**How to create a Docker container?**

Use the below command and a Docker image to create a Docker container

$ docker run -it -d <image\_name>

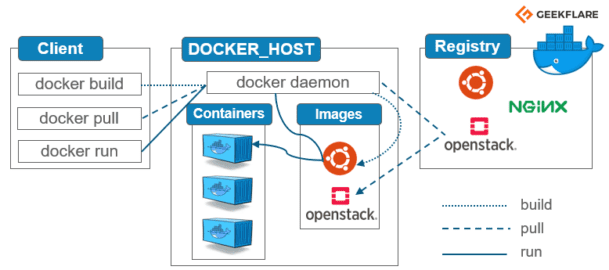
**What is Docker images:-**

The Docker image help to create Docker containers. You can create the Docker image with the build command. Every Docker images are stored in the Docker registry.

A [Docker](https://searchitoperations.techtarget.com/definition/Docker) image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker [container](https://searchitoperations.techtarget.com/definition/container-containerization-or-container-based-virtualization), like a template

An image consists of a collection of files (or layers) that pack together all the necessities—such as dependencies, source code, and libraries—needed to set up a completely functional container environment.

**Docker Architecture**



### **The Docker daemon**

The Docker daemon (dockerd) listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. A daemon can also communicate with other daemons to manage Docker services.

### **The Docker client**

The Docker client (docker) is the primary way that many Docker users interact with Docker. When you use commands such as docker run, the client sends these commands to dockerd, which carries them out. The docker command uses the Docker API. The Docker client can communicate with more than one daemon.

### **Docker registries**

A Docker *registry* stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

When you use the docker pull or docker run commands, the required images are pulled from your configured registry. When you use the docker push command, your image is pushed to your configured registry.

### **Docker objects**

When you use Docker, you are creating and using images, containers, networks, volumes, plugins, and other objects. This section is a brief overview of some of those

objects.

**Docker compose:**

**Docker hub?**

Docker Hub is a service provided by Docker for finding and sharing container images. The default version of Hub is the cloud-based registry that hosts all the public docker images like Ubuntu, Linux, etc.

We need to create repositories to push and pull the docker images, allowing us to share container images within our team, organization, customers. In the case of public repositories, we can share the images with the entire Docker community.

Docker images are pushed to Docker Hub through the ‘Docker push’ command. A single Docker Hub repository can hold many Docker images.

How to install Jenkins in AWS Amazon Linux EC2 instance

|  |
| --- |
|  |
|  | ------------------------------ |
|  | sudo yum -y install openjdk-8-jdk |
|  | sudo yum -y install git |
|  | sudo yum -y install maven |
|  | sudo yum -y wget ( optional ) |
|  |  |
|  | Steps to install Jenkins |
|  | ------------------------------ |
|  | sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo |
|  | sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key> |
|  | sudo yum -y install Jenkins  In case any error while installing the jenkins  ----------------------------------------------  sudo amazon-linux-extras install epel -y  sudo vi vim /etc/yum.repos.d/epelfordaemonize.repo  Add the following:  [daemonize]  baseurl=https://download-ib01.fedoraproject.org/pub/epel/7/x86\_64/  gpgcheck=no  enabled=yes  -------------------------------- |
|  | sudo systemctl start jenkins |
|  | sudo systemctl enable --now Jenkins  cat /var/lib/jenkins/secrets/initialAdminPassword  Open browser enter the public ip:8080 it will open Jenkins home page and ask for initial password |

Steps to install Docker Amazon linux2

======================================

sudo yum update -y

sudo amazon-linux-extras install docker

sudo service docker start

sudo usermod -a -G docker ec2-user

sudo usermod -a -G docker Jenkins(Optional if Jenkins is installed on the same machine)

docker info

Steps to containerize the postgresql

Step 1: docker pull postgres

Step 2: docker run --name posgresdb -p 5432:5432 -e POSTGRES\_DB=postgres -e POSTGRES\_USER=postgres -e POSTGRES\_PASSWORD=root12345 5861c038d674

Step3: docker container exec -it containerid bash

Step 4:su postgres

Step 5: psql

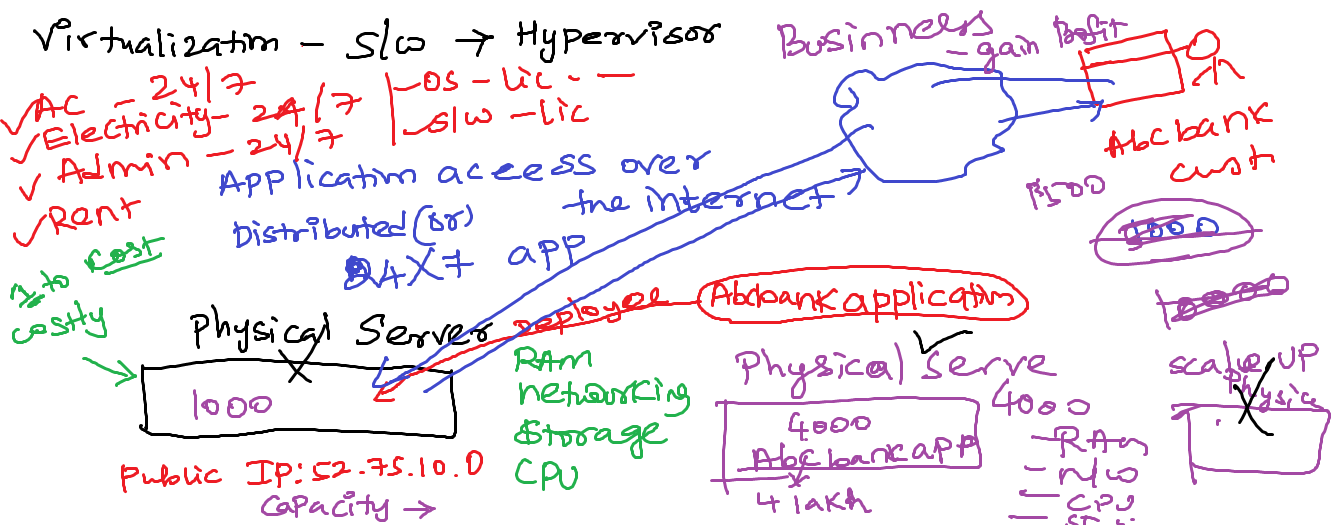
Step 6: use postgres

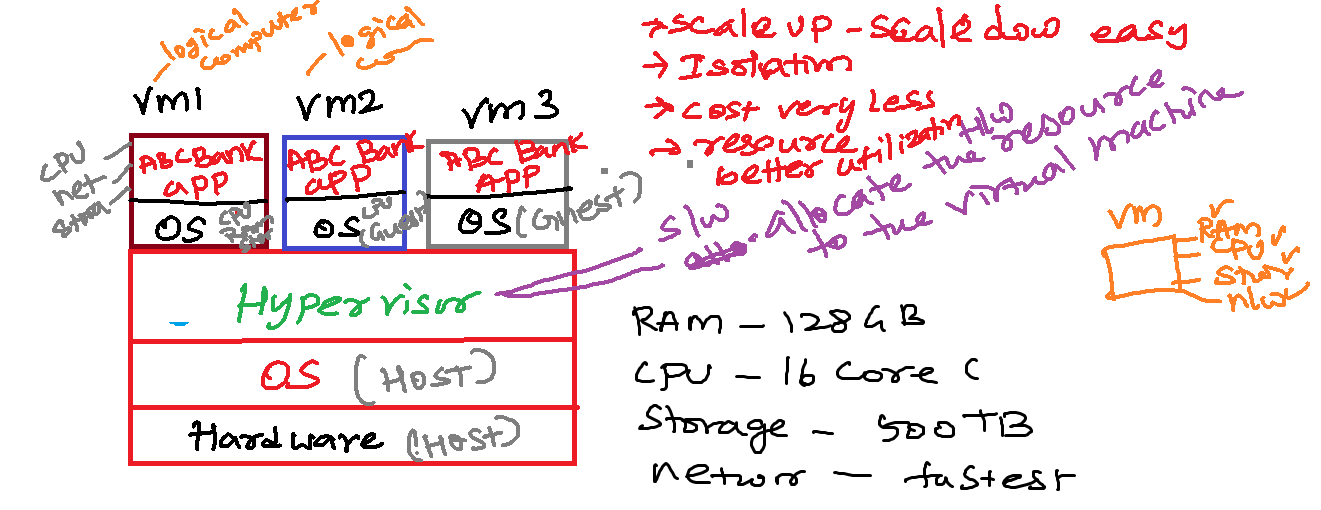
Step 7: create schema schemaname;

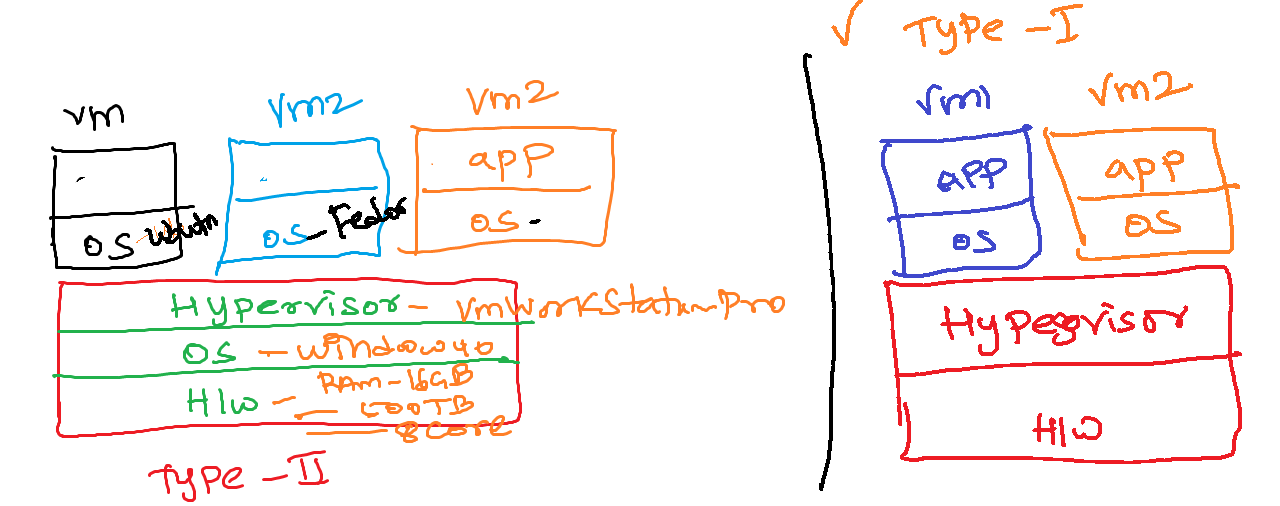
Step 8: \q

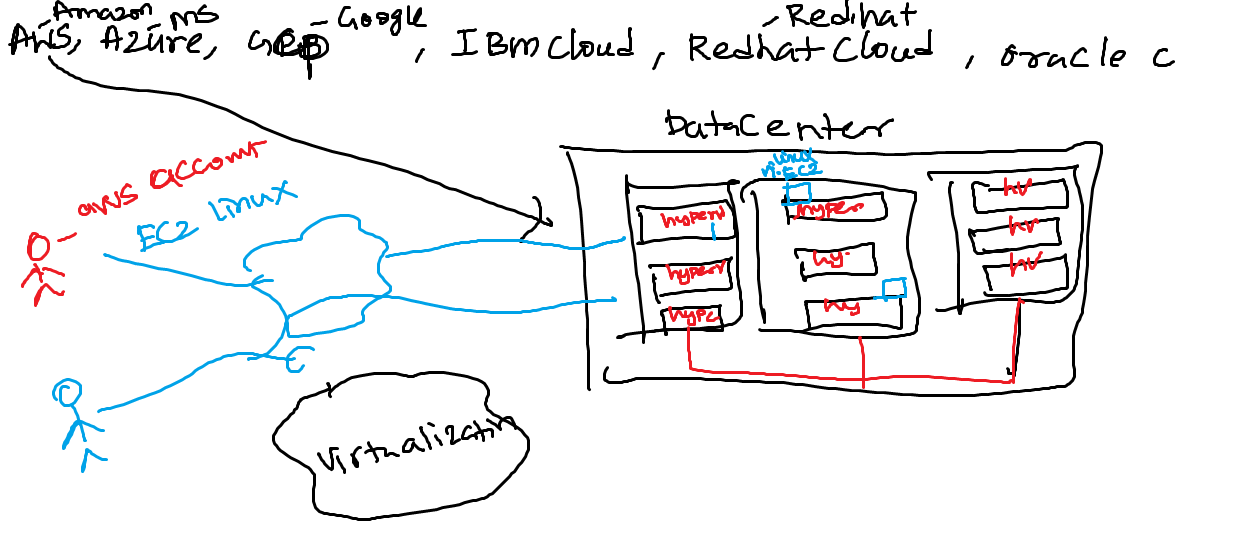
Step 9: exit

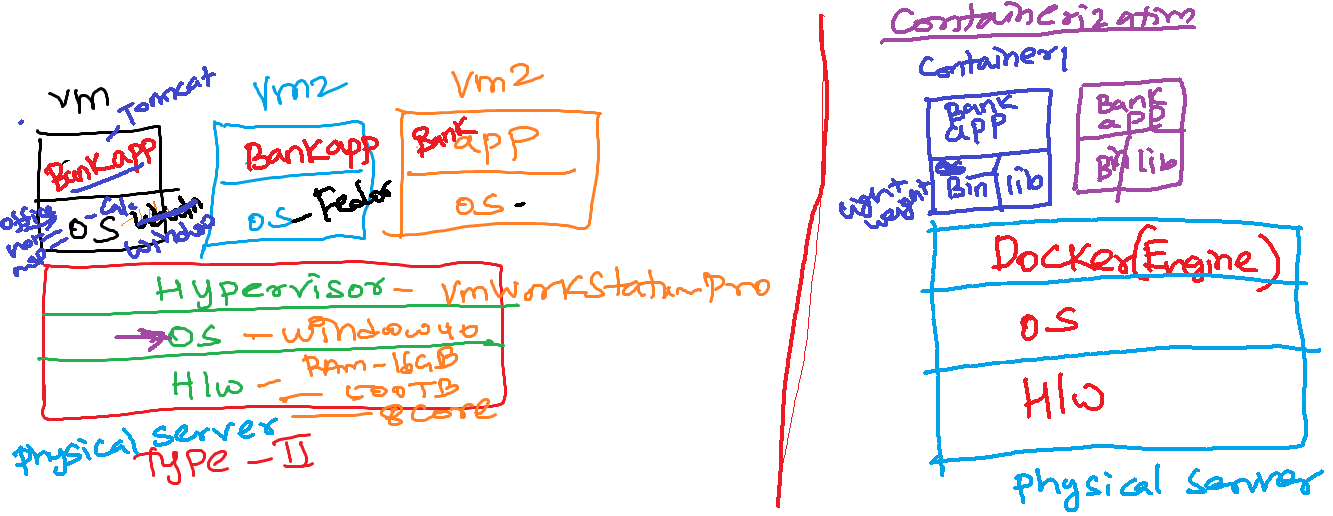
Step 10: docker run --name railway1 -p 8888:9848 --link posgresdb e654f73c7959

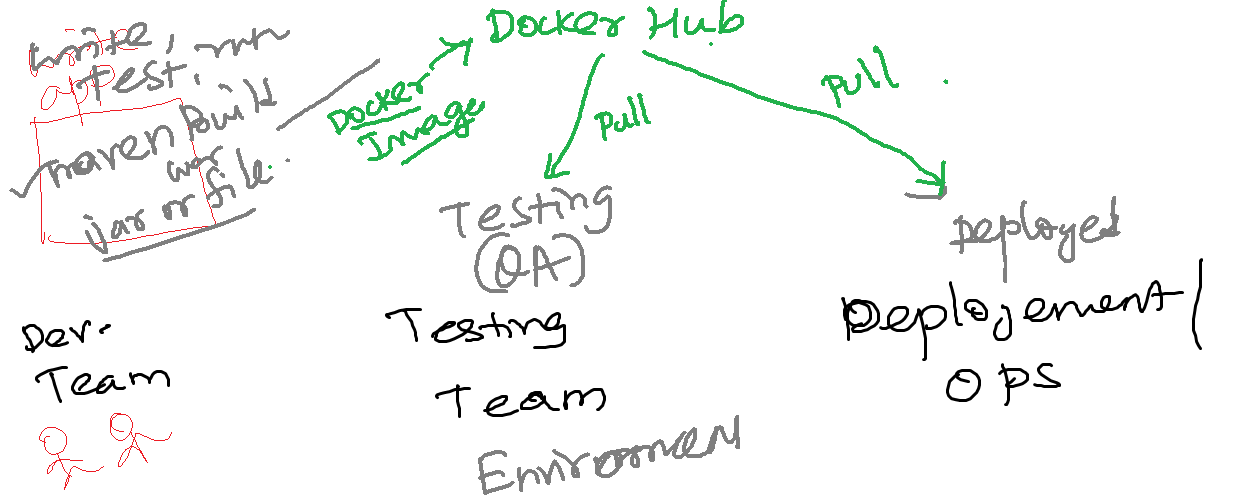


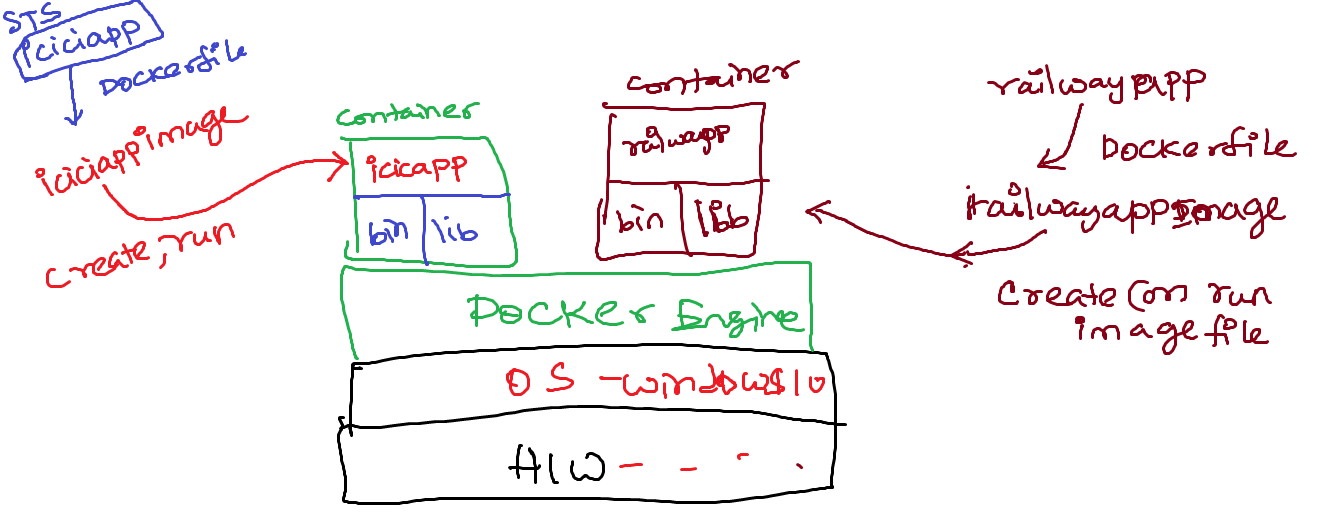


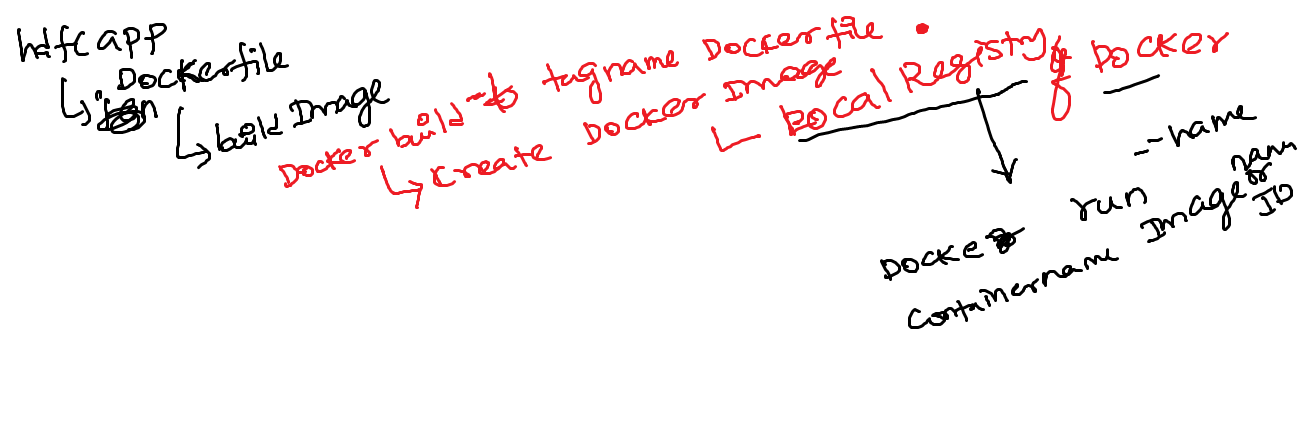


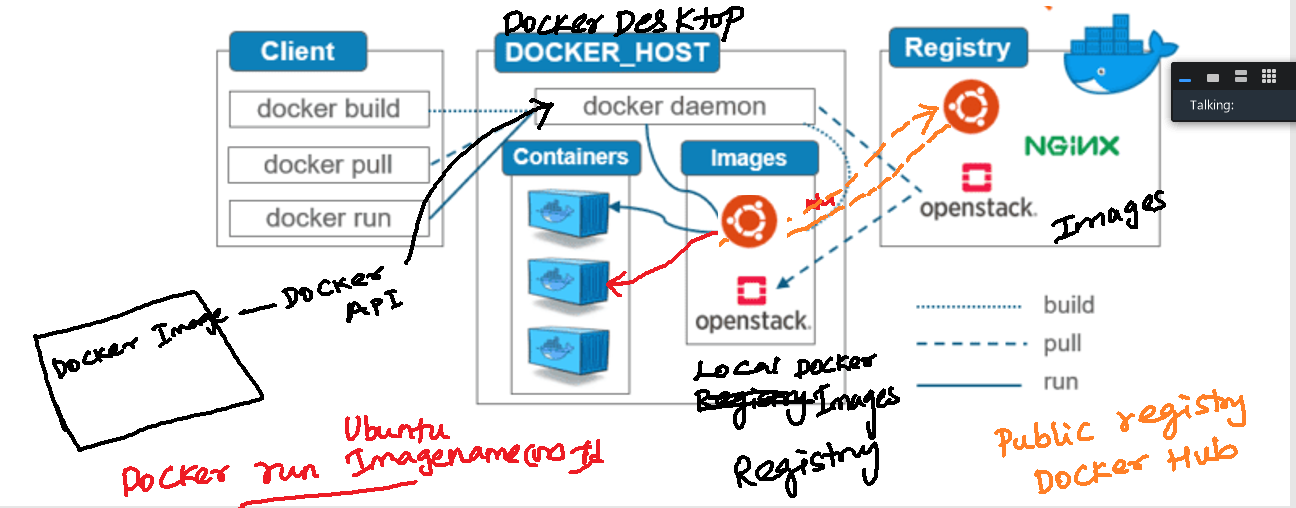


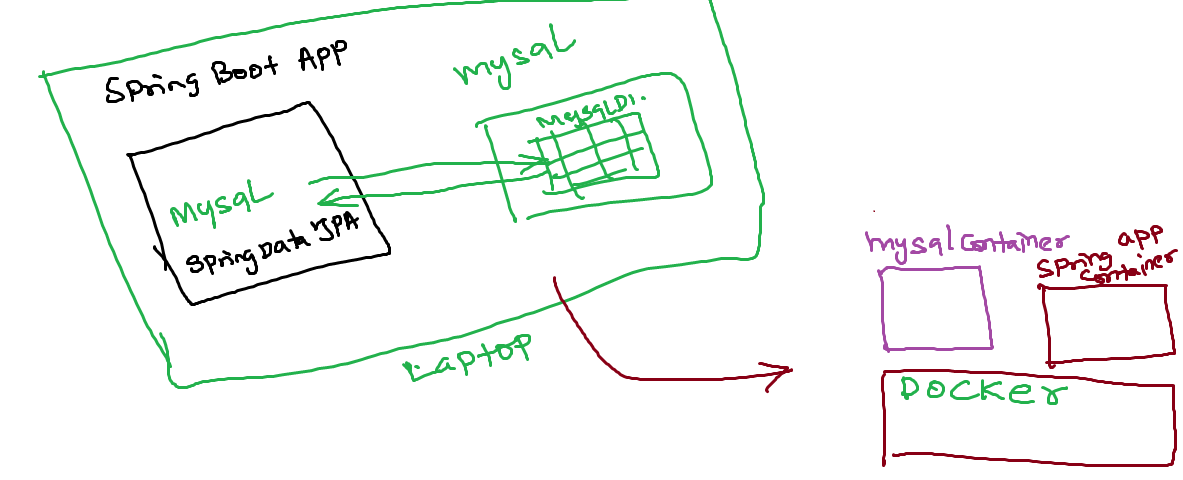












Git:

* Source code management system
* Source – java project
* Repository – A location where your project files get stored
  + Local git repository – git -> project files gets stored(repository) in the same machine
  + Remote git repository – github -> project files gets stored(repository) in the remote machine
* Developer -> java code -> git or github

Git commands developer can talk to the git or github

*Git init*

*Git add*

*Git commit*

*Git status*

*Git branch*

*Git checkout*

$ git config --global credential.helper wincred

Generate a token in github

Enter the token during authentication

* How to install git on windows
  + Download the git software

*Project Configuration tool or project management tool*

* *Create or modify the project*
* *Compile the project*

*Build the project*

* *Test*
* *Run the project*

*1. Ant*

*2. Apache maven – pom.xml*

*Install the maven*

*Lifecycle*

*Maven clean*

*Maven build*

* validate - validate the project is correct and all necessary information is available
* compile - compile the source code of the project
* test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
* package - take the compiled code and package it in its distributable format, such as a JAR.
* verify - run any checks on results of integration tests to ensure quality criteria are met
* install - install the package into the local repository, for use as a dependency in other projects locally
* deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

*.jar – java archive file*

*.war – web archive file*

*.ear – enterprise archive file*

*3. Gradle – advanced, more performance, Groovy or Kotlin gradle.build*

*Maven:*

*Project Management tool*

*Jenkins:-*

*How to download the jenkins*

*1. download jenkins.war file and the run the war file*

*java -jar jenkins.war*

*Jenkins by default start and runs on localhost:8080. If the port no 8080 is not available*

*Jenkins will not start give you exception*

*java -jar jenkins.war –httpPort=7777*

*2. download the Jenkins.msi file(install)*

*Pattern -> Regex ->*

*. –> any character a-zA-Z0-9allspecialcharacters*

*\* -> zero or more character*

*abc -> match word Abc*

*[abc] -> a or b or c*

*[] – range of characters*

*[a-z] – a or b or c … z*

*Regex Universal Characters*

* *\*- zero or more*
* *+ - one or more*
* *? – zero or one*
* *| - OR*

*Gradle Instalation*

*GRADLE\_HOME C:\gradle\gradle-6.9.1*

*Path C:\gradle\gradle-6.9.1\bin*