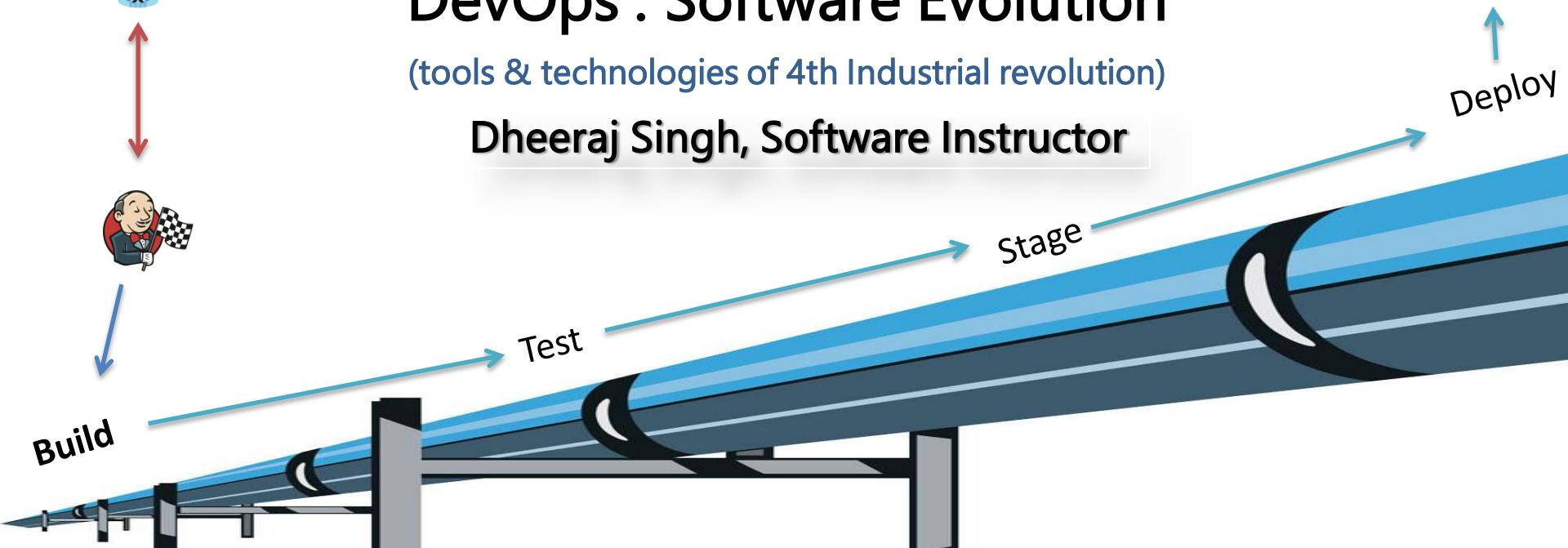


DevOps : Software Evolution

(tools & technologies of 4th Industrial revolution)

Dheeraj Singh, Software Instructor



Careers

[About](#) [Careers](#) [Contact](#) [Legal](#) [Investors](#) [Suppliers](#)

DevOps Engineer - Design Technology

Job Category [Engineering & Information Technology](#)

Location [Palo Alto, California](#)

Req. ID [50116](#)

Job Type [Full-time](#)

BUILD
THE
FUTURE

COMPANIES

2582 companies reportedly use Jenkins in their tech stacks, including Facebook, Netflix, and Instacart.



Facebook



Netflix



Instacart



Lyft



eBay



Udemy



LinkedIn



Coursera



Twitch

Agenda

Technologies of 4TH Industrial Revolution.

Uber | Tesla (Use Case)

What is DevOps & Why?

Various stages in DevOps.

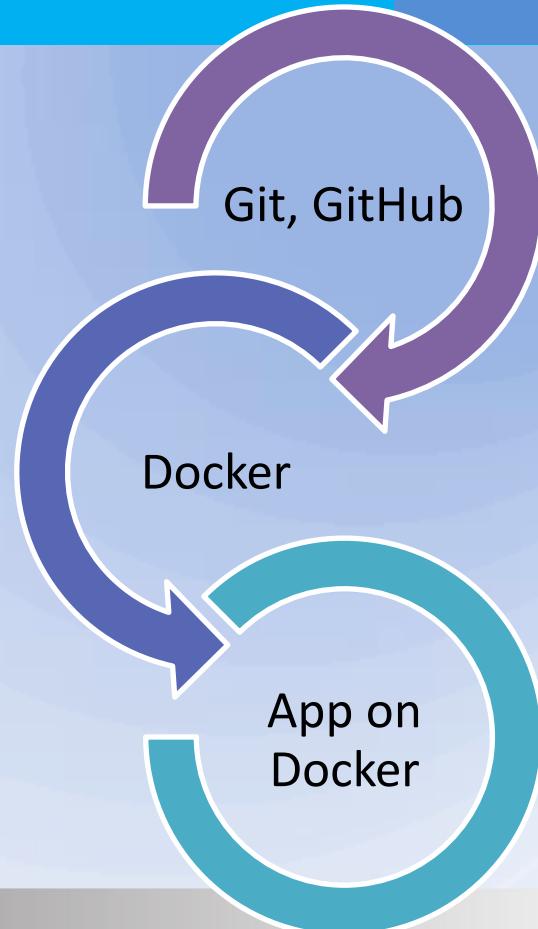
DevOps tools : Git, Maven, Jenkins, Docker.

DevOps Use-Case ()

Setting CI CD pipeline (Jenkins | Circle CI)

Chapter A:

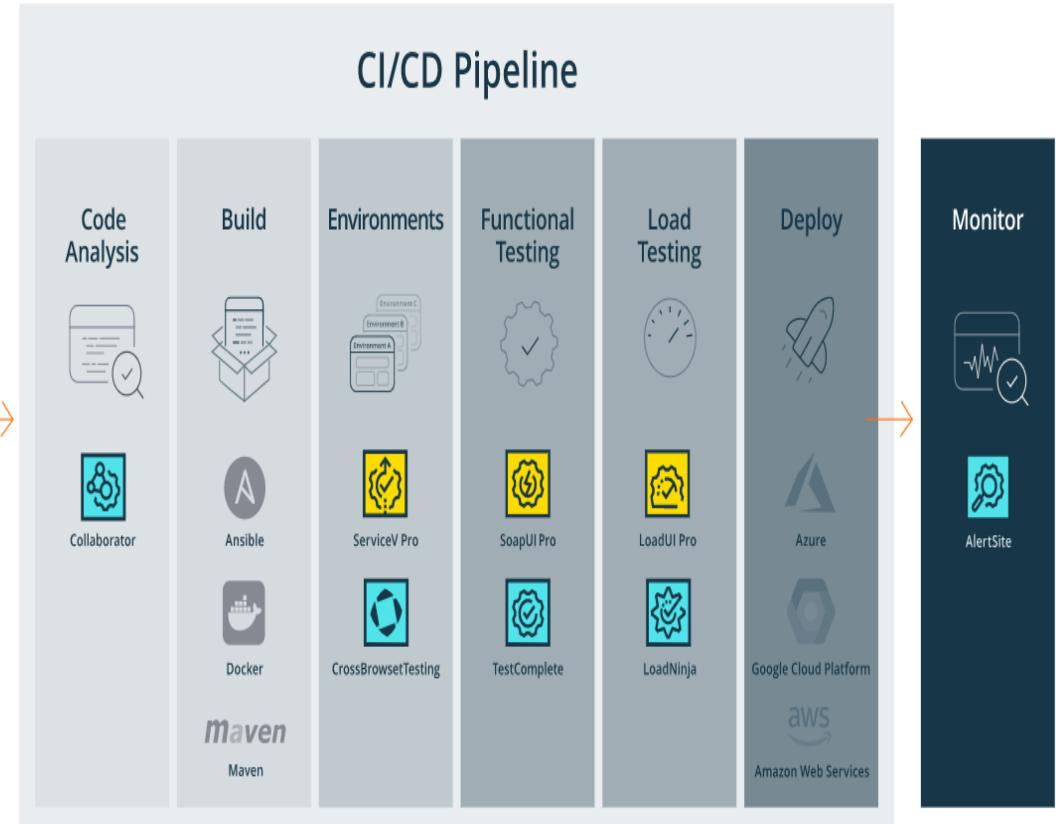
Understanding
Docker container
technologies
Build a service App
with Docker



Chapter B :

Completely Practical

TASK : Docker app with DevOps (CI/CD) automated workflow



**We will try Chapter A to be achieved
today...**

What is Waterfall Model



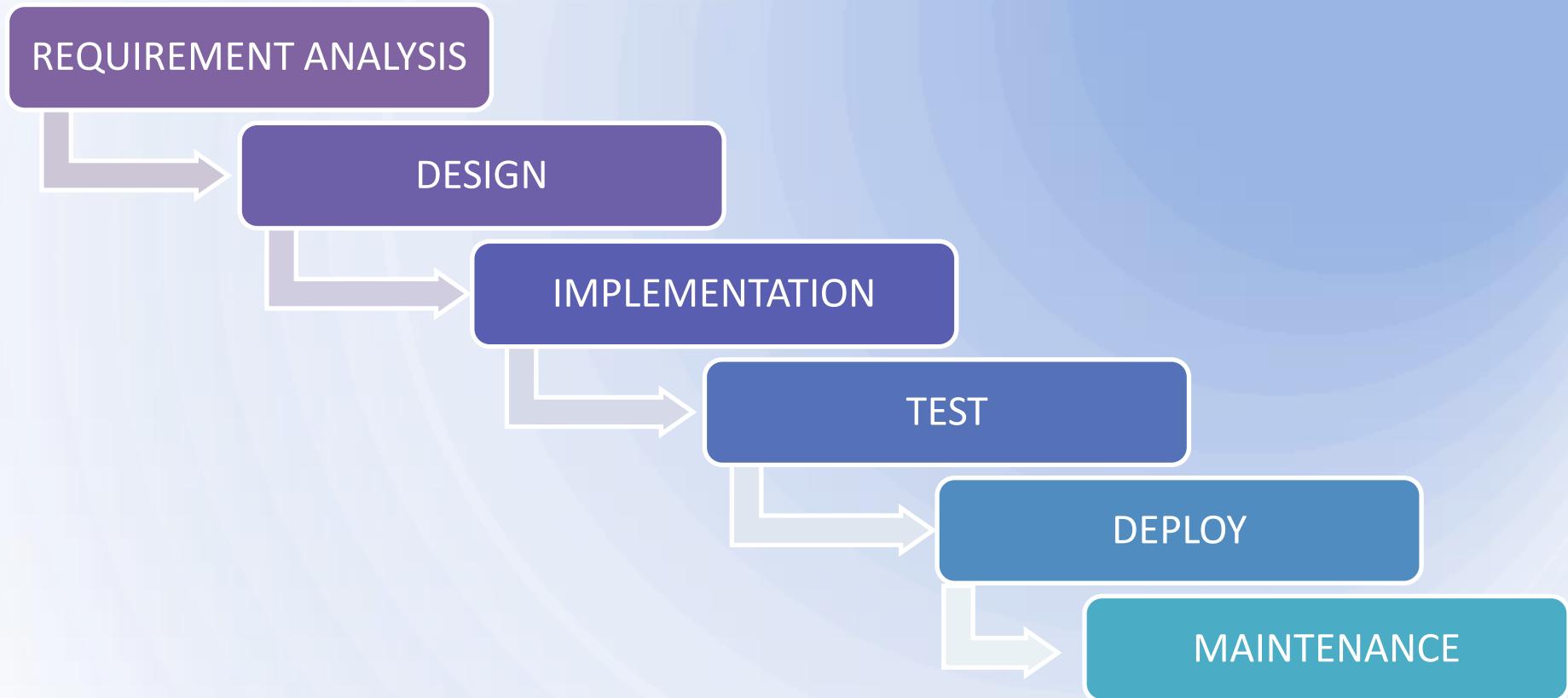
Dr. Winston W. Royce

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks.

“I believe in this concept, but the Implementation is described as risky and invites failure”

<http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>

What is Waterfall Model



Disadvantages Of Waterfall Model



Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.



No working software is produced until late during the life cycle contains High amounts of risk and uncertainty.



Not a good model for complex and object-oriented projects.
Poor model for long and ongoing projects.



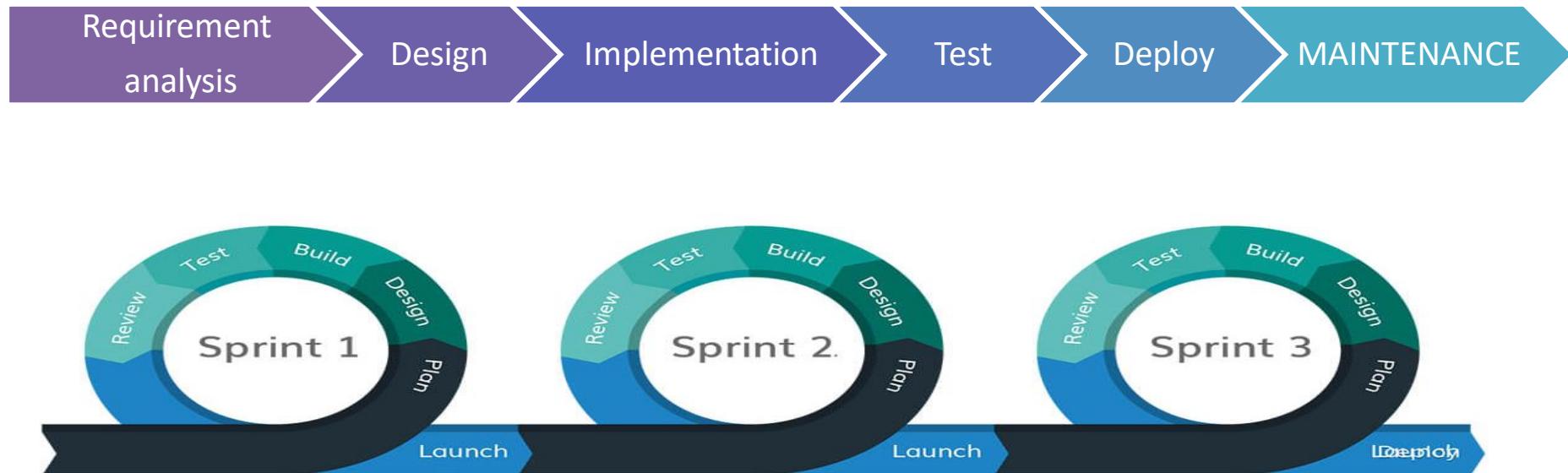
Not suitable for the projects where requirements are at a moderate to high risk of changing.

What is Agile

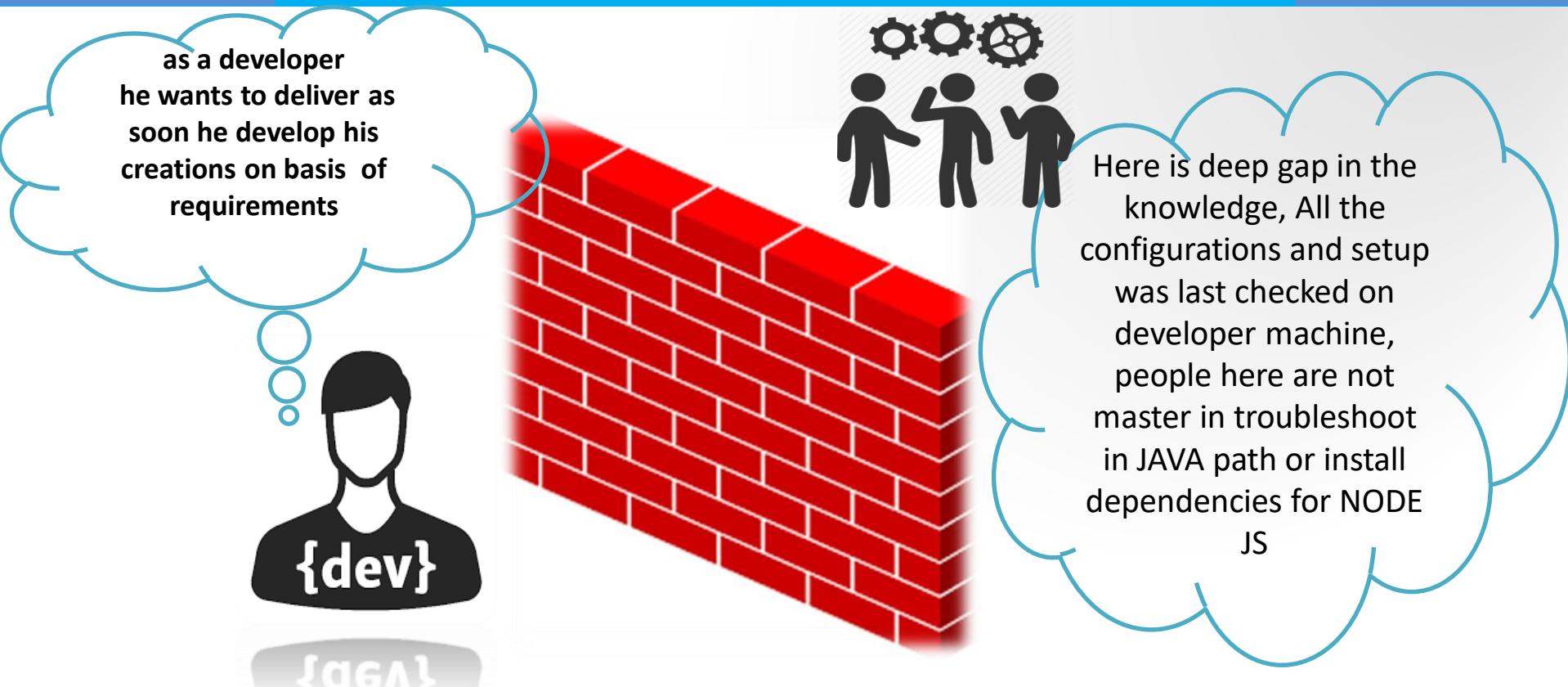
In the early 2000s, a small group of **software** industry leaders met in Snowbird, UT, to discuss these new **methodologies**. The term **agile software development** was coined in 2001 to describe the flexible nature of **software** developed in iterative stages and became a blanket term for the new **methodologies**

Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer/end user.

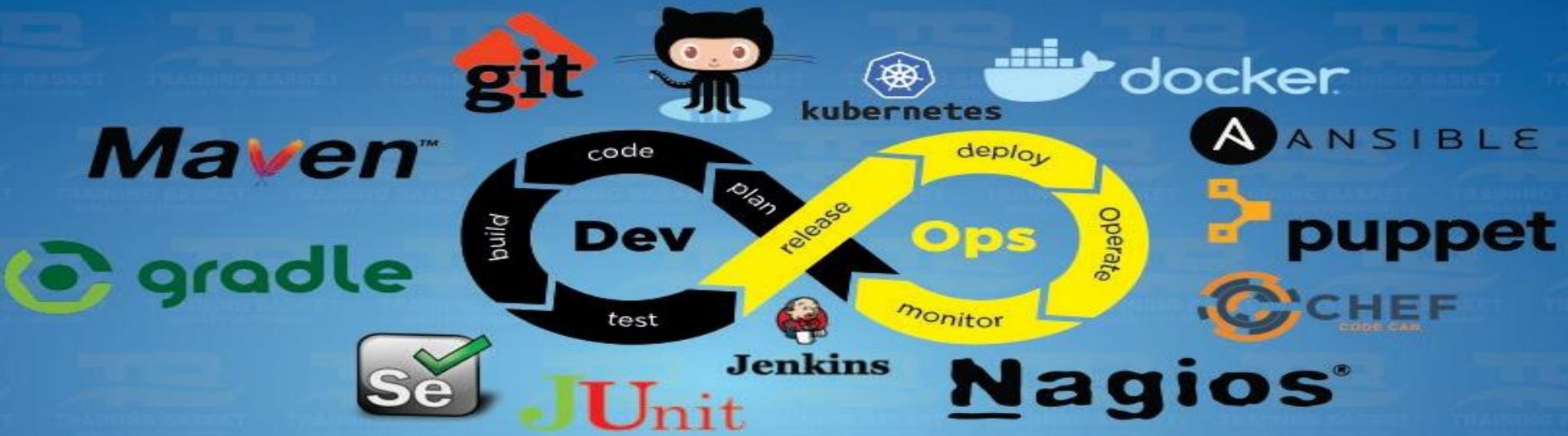
What is Agile



Limitation with agile



Limitation with agile



There Is One And Only Solution, Adopt DevOps

Limitation with agile

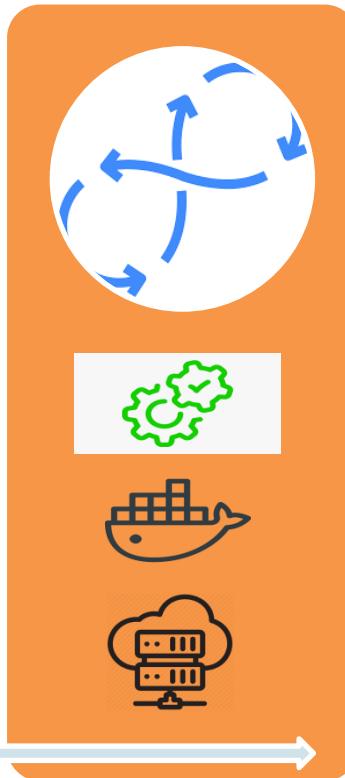
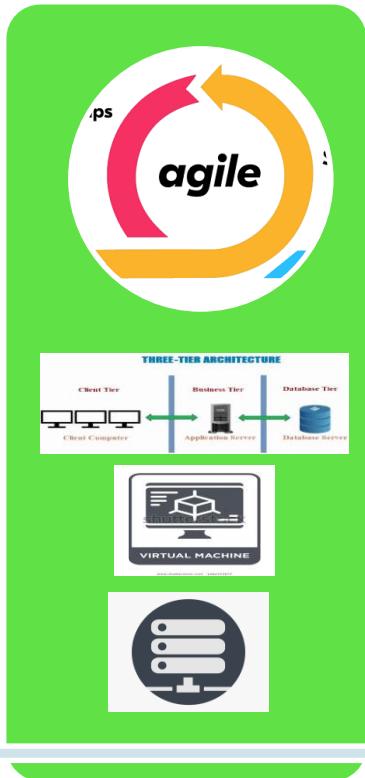
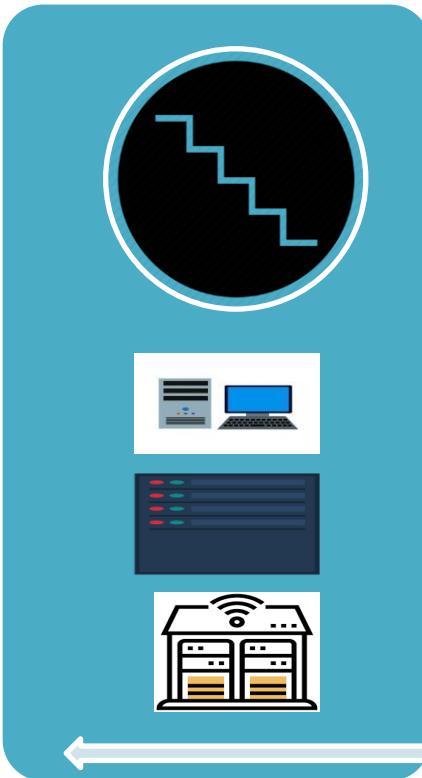
Let us sum up what we have discussed by now

LET'S SUM UP

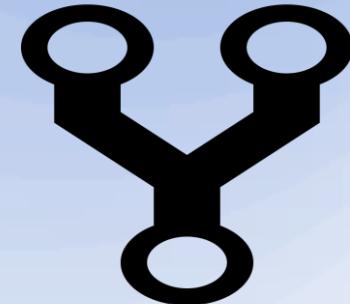
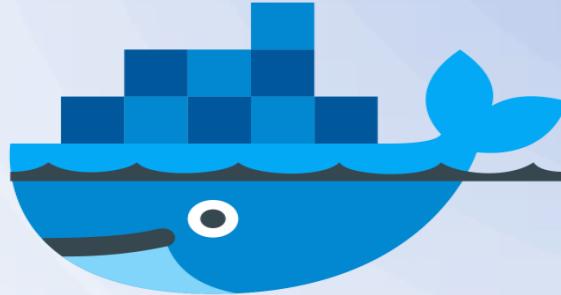
Architecture

Packaging

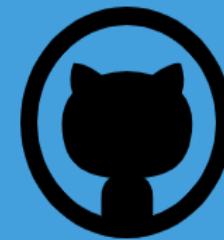
Infrastructure



Lets do hands dirty with most of the tools that would help individuals to setup CI CD pipelines, following tools will be there today :



Version Control with Git and GitHub



LET'S SUM UP

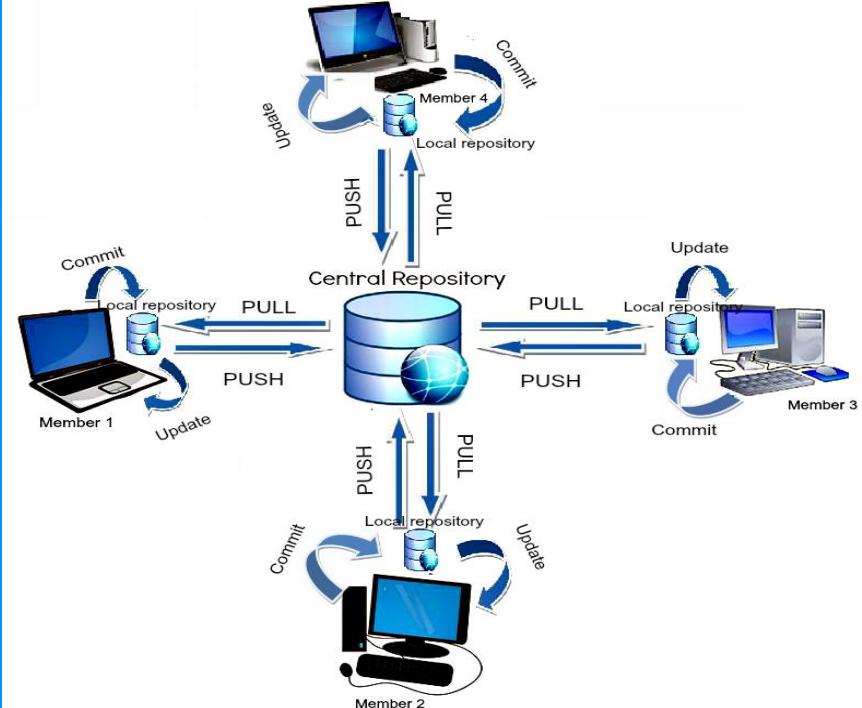
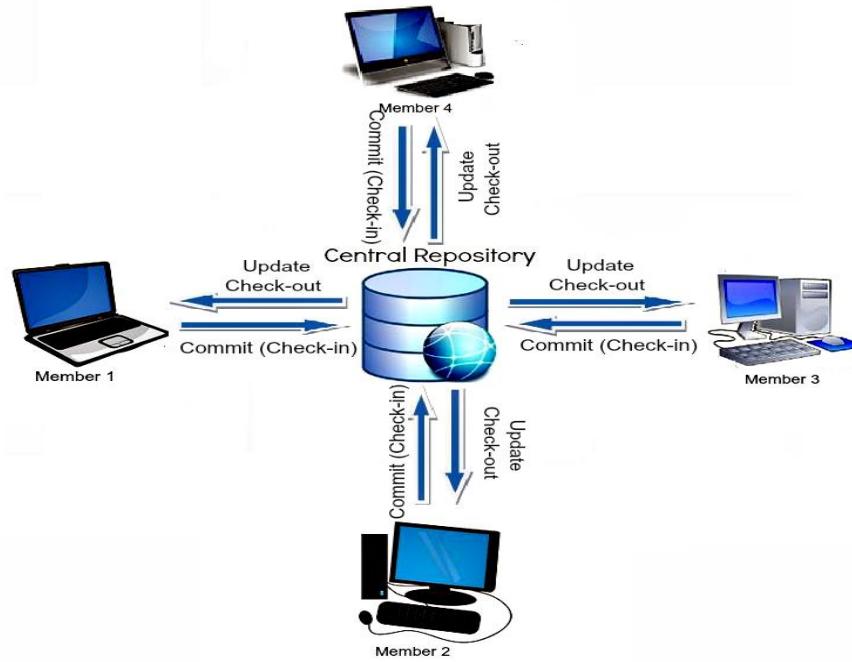
Version Control is:

The process of managing changes to source code or set of files over time.

Version control software keeps track of every modification to the code in a special kind of database.

If a mistake is made, developers can restore and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members or contributors.

LET'S SUM UP



Benefits of Version Control

- Backups
- Preserves efficiency and agility
- Source Code Management
- Traceability
- Branching and Merging
- Reverting Changes
- Collaboration

Variety Of Version Control Systems

Software	Network architecture
 Git	Distributed
 Mercurial	Distributed
 SVN	Client-server
 CVS	Client-server

Benefits of Version Control

Git is a **distributed** and actively maintained **open source project** originally developed in 2005 by **Linus Torvalds**, the famous creator of the Linux operating system kernel.

Unlike older centralized version control systems such as SVN and CVS, Git is distributed: every developer has the full history of their code repository locally. Git also works well on a wide range of operating systems and IDEs (Integrated Development Environments).

Git is a Free and Open Source Distributed Version Control System

Demo Time



Demo Time

```
1 git init // creates a new git repository
2 // create a new file (README.md), do something new!
3
4 git add README.md or git add * or git add --all
5 // Proposes this new change and adds it to the index
6
7 git status
8 // check what has been going on in git, see whats staged or not.
9
10 git commit -m "commit description"
11 // tell us what you did and add to HEAD
12
13 git push origin master
14 // push HEAD to the remote repository
15
16 git log
17 // see the repository history
18
19 git pull
20 // grabs the latest directory with updates from the remote repository
21
22 git clone <url>
23 // clone a remote repository to local
```

What is GitHub then?

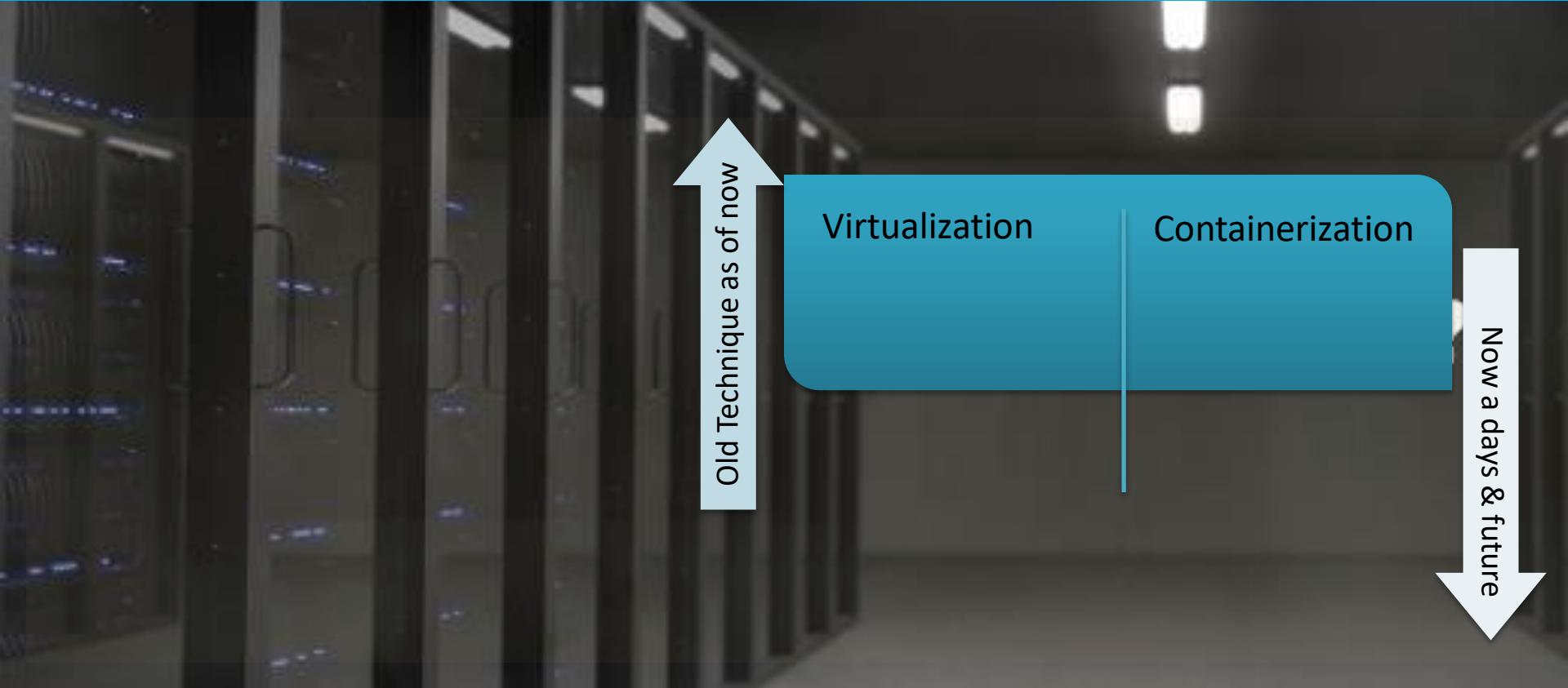


Why should I learn Virtualization

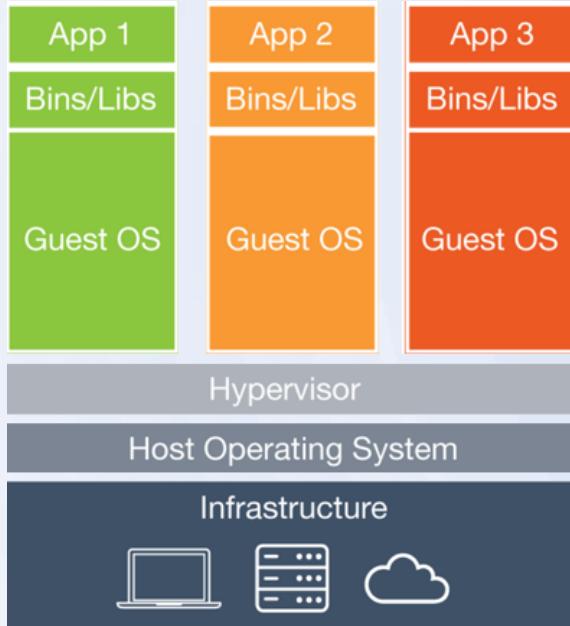
The most important function of virtualization is the capability of running multiple operating systems and applications on a single computer or server. This means increased productivity achieved by fewer servers

For being multilingual and do hands on any technology or language

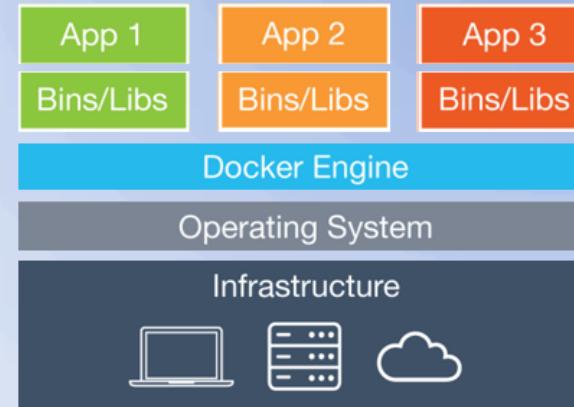
Virtualization



Virtual Machines vs Containers

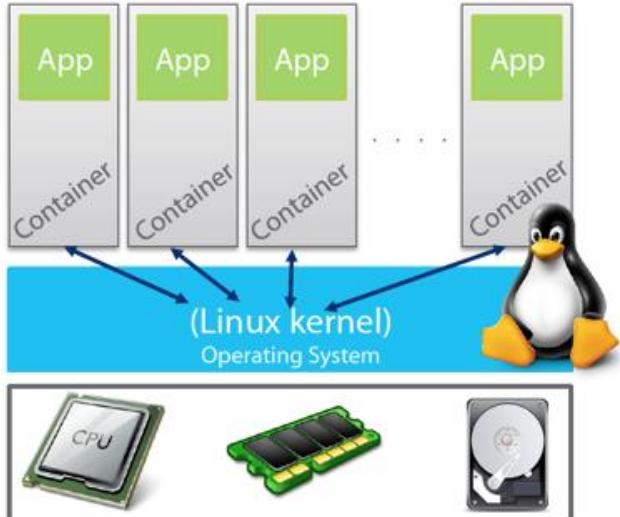


Virtual Machines



Containers

Container



Containers consume less CPU, RAM and disk resource than Virtual Machines

Get Started

- Install Docker Engine on host machine
- Windows or Mac: Docker toolbox

Get Started....

- Run few basic commands:`docker info`
- `docker images`
- `docker ps`
- `docker ps -a`

Run a software image in a container

- docker **run** *hello-world*
- docker **run -i -t ubuntu** /bin/bash
- Re-run commands:
 - docker info
 - docker images
 - docker ps
 - docker ps -a
 - docker history <ImageId>

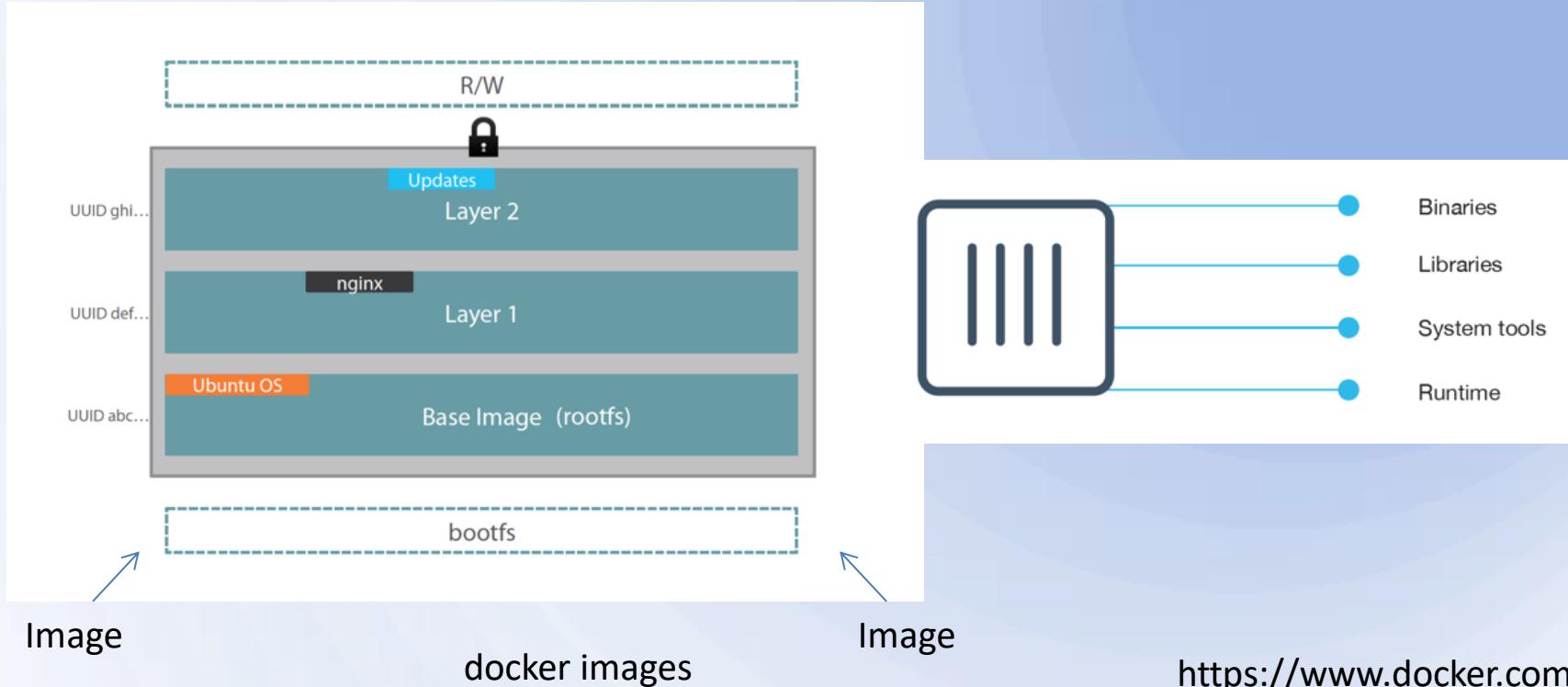
Docker hub

Signup on hub.docker.com

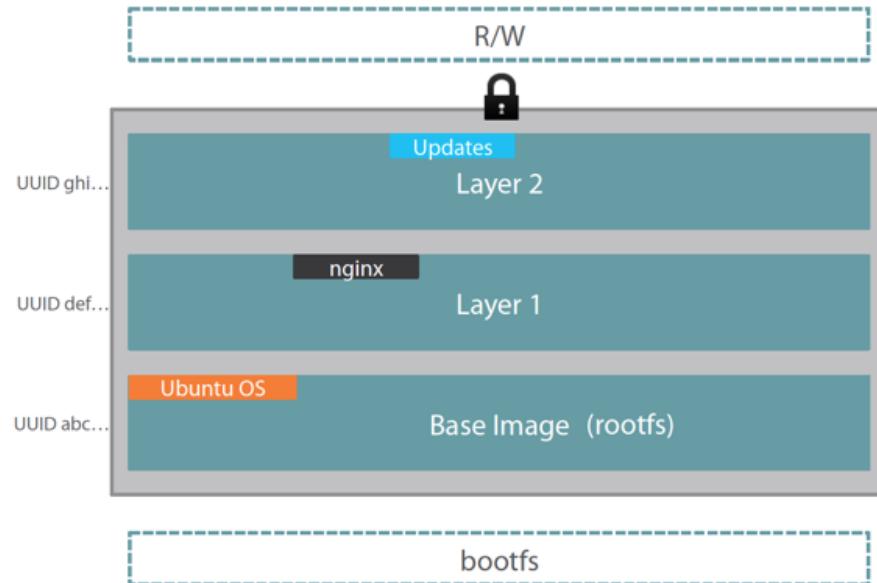
Docker image hosting provider



Docker image & Containers



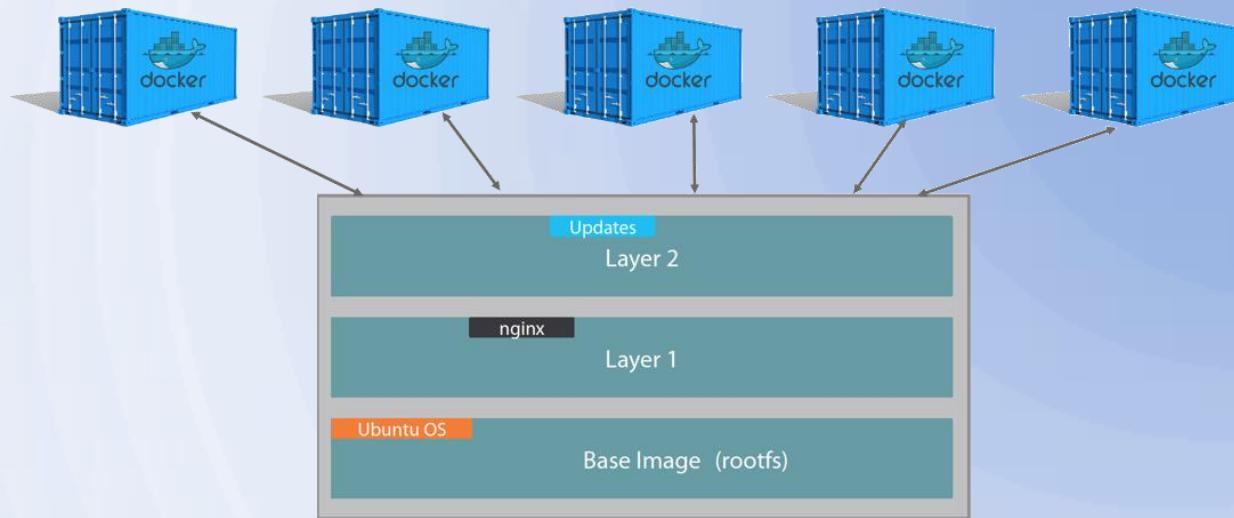
docker commit



docker commit command saves changes in a container as a new image.

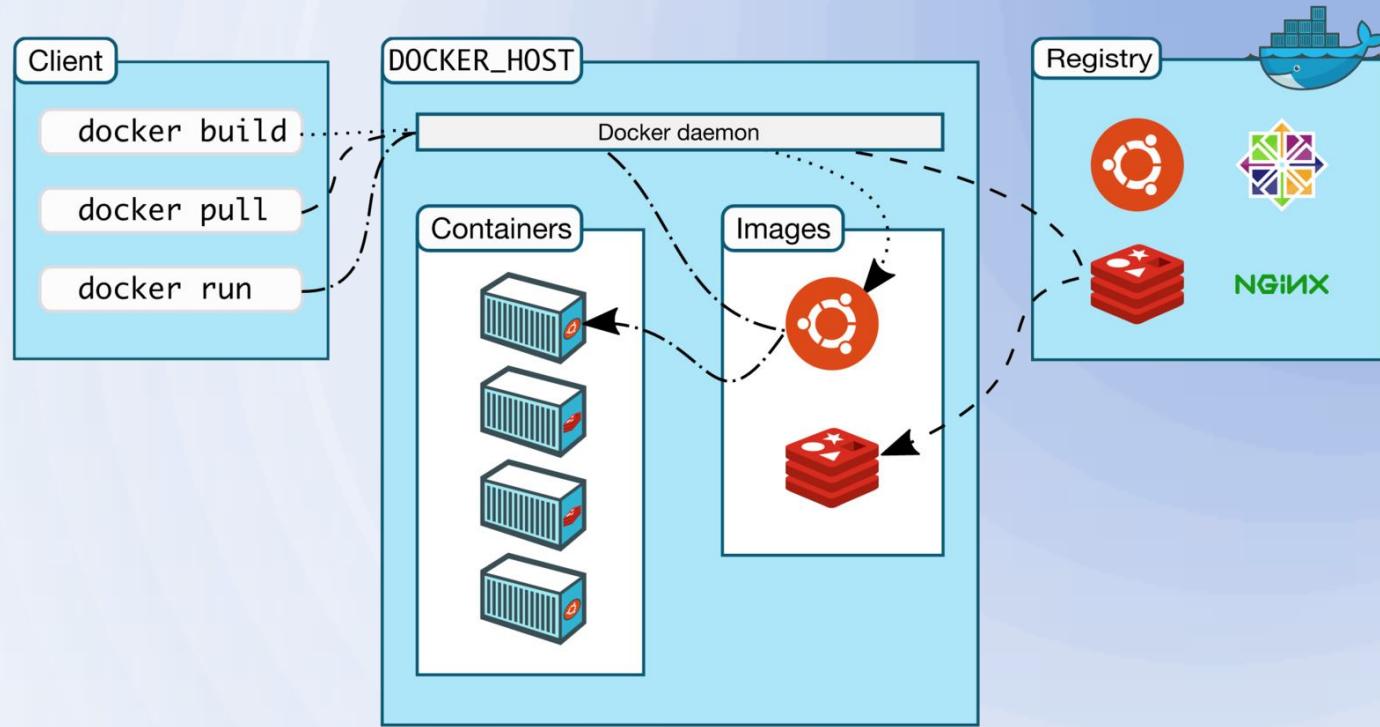
`docker commit [options] [container id] [repository:tag]`

Docker Containers

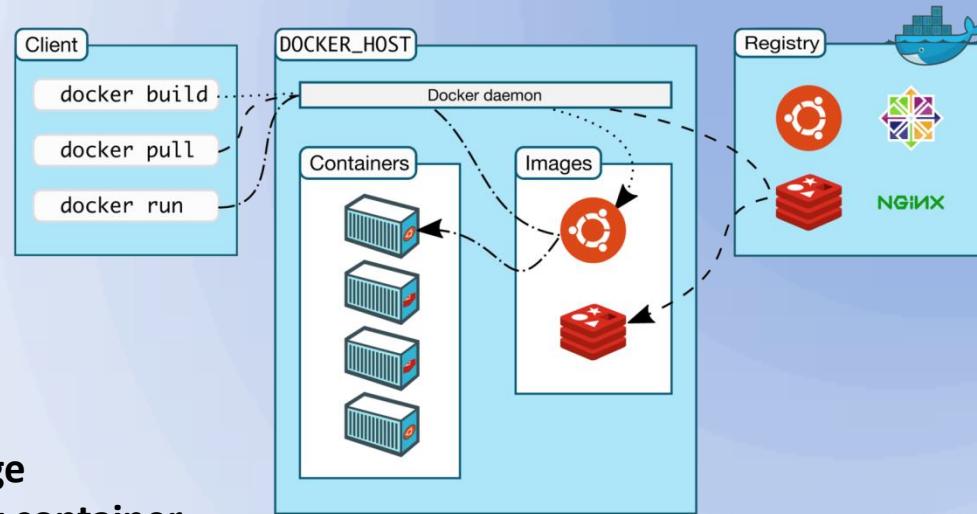


- ✓ Run container from a image: `docker run`
- ✓ Check running containers: `docker ps`
- ✓ Check all (Running, Stopped, Paused) containers: `docker ps -a`
- ✓ Inspect a container: `docker inspect`

Docker's architecture



docker run



1. Pulls the image
2. Creates a new container
3. Allocates a filesystem and mounts a read-write layer
4. Allocates a network / bridge interface
5. Sets up an IP address
6. Executes a process that you specify
7. Captures and provides application output



Docker Application

5 Container default command / Entry point

4 Expose container port to host

3 Download & Install the app

2 Install tools for the app

1 Download Base Image (OS)

Demo Node Web App

https://github.com/dheerajthedev/Demo_DockerCI-CD

Node Web Docker App



5 CMD ["npm", "start"]

4 EXPOSE 8080

3 COPY ./usr/src/app & RUN npm install

2

FROM node:argon

1

Dockerfile

A Dockerfile is a configuration file that contains instructions for building a Docker Image

Demo

1. Login to host machine & check docker command
2. git clone https://github.com/ashishapy/Demo_DockerCI-CD.git
3. cd Demo_DockerCI-CD
4. docker build -t <username>/demo_dockerci-cd .
5. docker images
6. docker run -p 80:8080 -d <username>/demo_dockerci-cd
7. Enter host machine ip address on browser.

Push to Docker hub

- docker login
- docker push <username>/demo_dockerci-cd

OOO!



We have our own image now!

Can we use it to quickly spin node-web-app?

Run own image

```
docker run -p 80:8080 -d <username>/demo_dockerci-ci
```



QUESTION



congratulations!

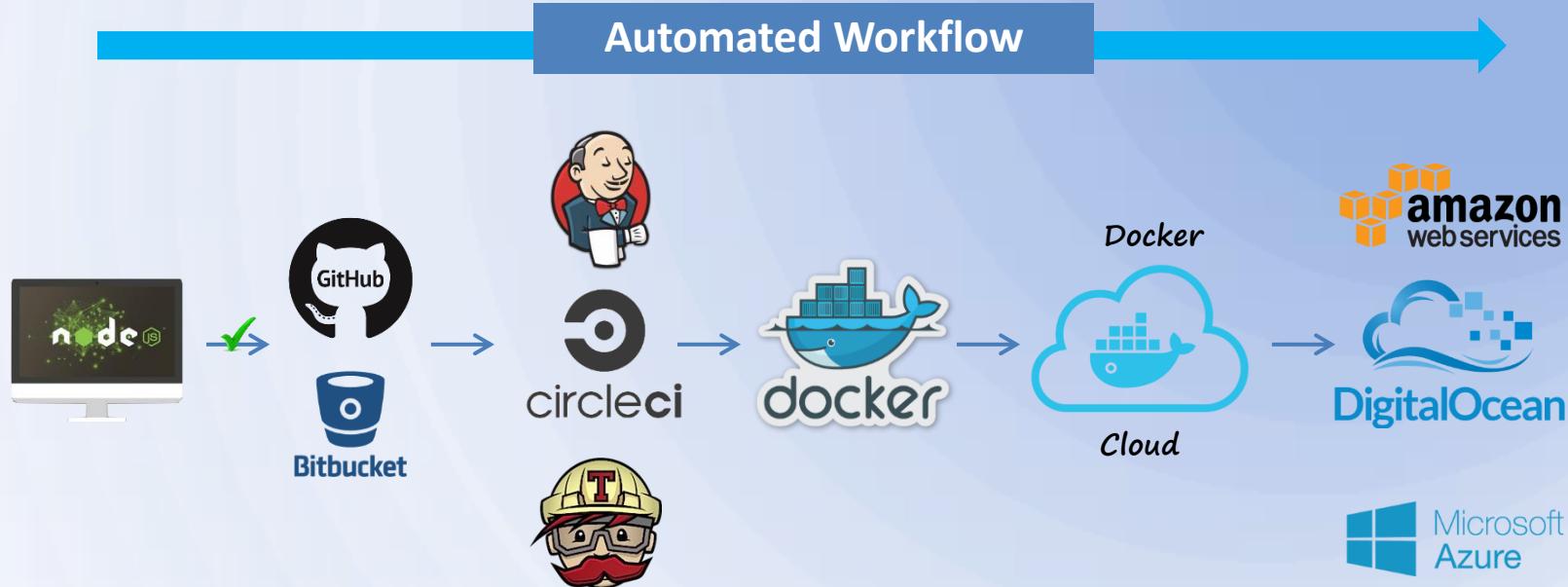
Achievement
unlocked!

Successfully completed Docker basics

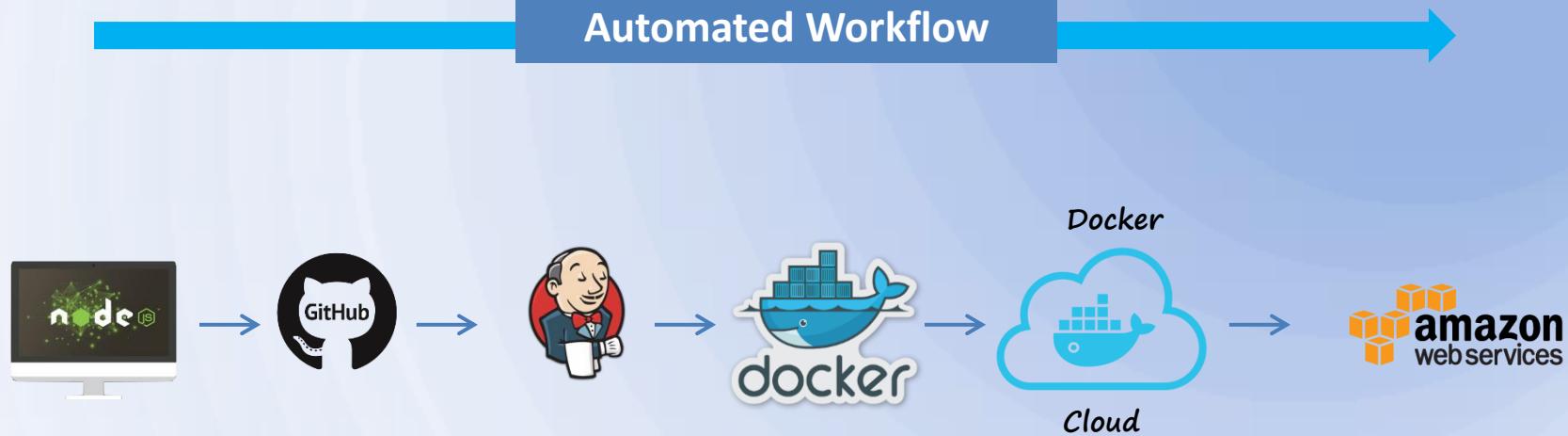
Chapter - B

Docker app with DevOps (CI/CD)
automated workflow

DevOps (CI/CD) Workflow

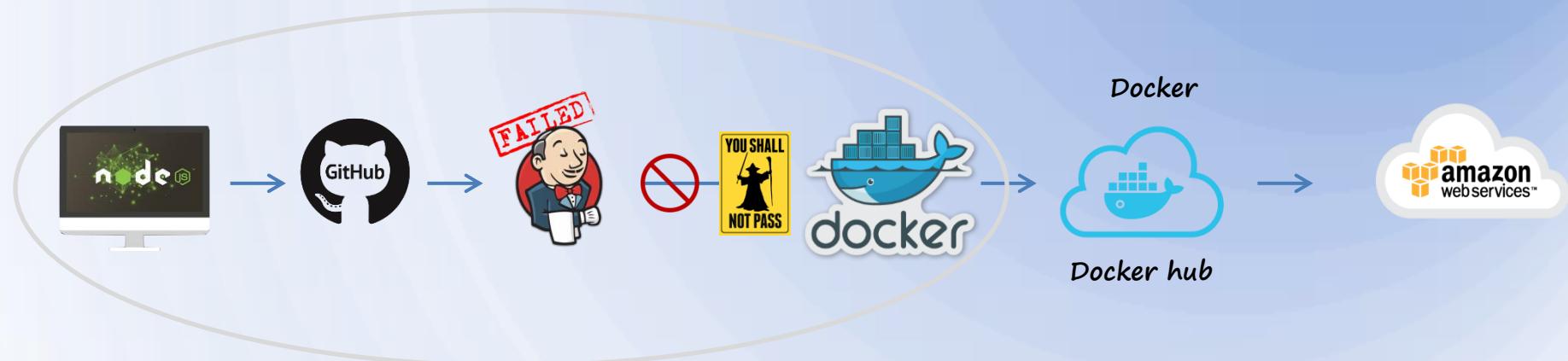


DevOps (CI/CD) Workflow



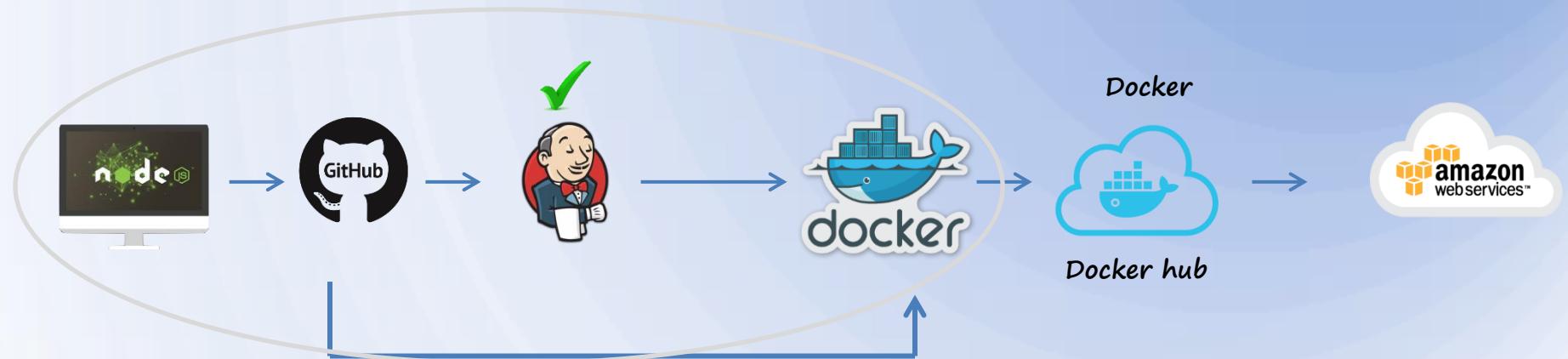
Continuous Integration

Automated Workflow



Continuous Integration

Automated Workflow

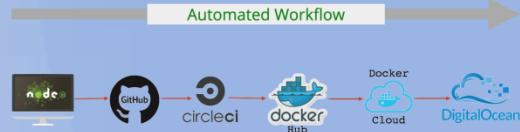


Docker Hub



1. Link your GitHub account with Docker Hub [here](#).
2. Go to "Create" -> "Create Automated Build" and create.
3. Under "Build Settings" of the repo, uncheck "When active, builds will happen automatically on pushes." option & save.
4. Go to "Build Triggers" section, click on "Activate Trigger" button.
5. Copy "Trigger URL" generated from previous step.

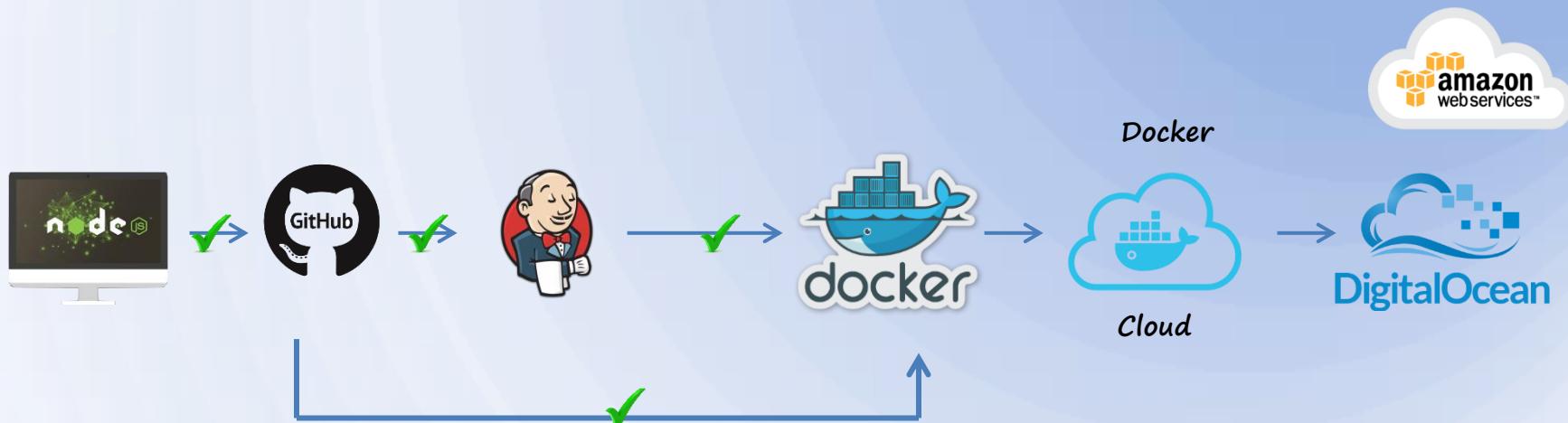
CircleCI



1. Signup with CircleCI using your GitHub account.
2. On CircleCI "Dashboard", click on "Add Projects".
3. Select your GitHub account then "Build Project" for the repo.
4. If all is well then it should successfully pass the project.
5. Go to "Project Settings" -> "Environment Variables" & add environment variable as Name: DOCKER_HUB_TRIGGER; Value: curl -H "Content-Type": application/json" --data '{"build": true}' -X POST <Docker_Hub_TRIGGER_URL>
6. Change circle.yml file to include "deployment" section.
7. Push the changes to GitHub repo and watch all auto builds.

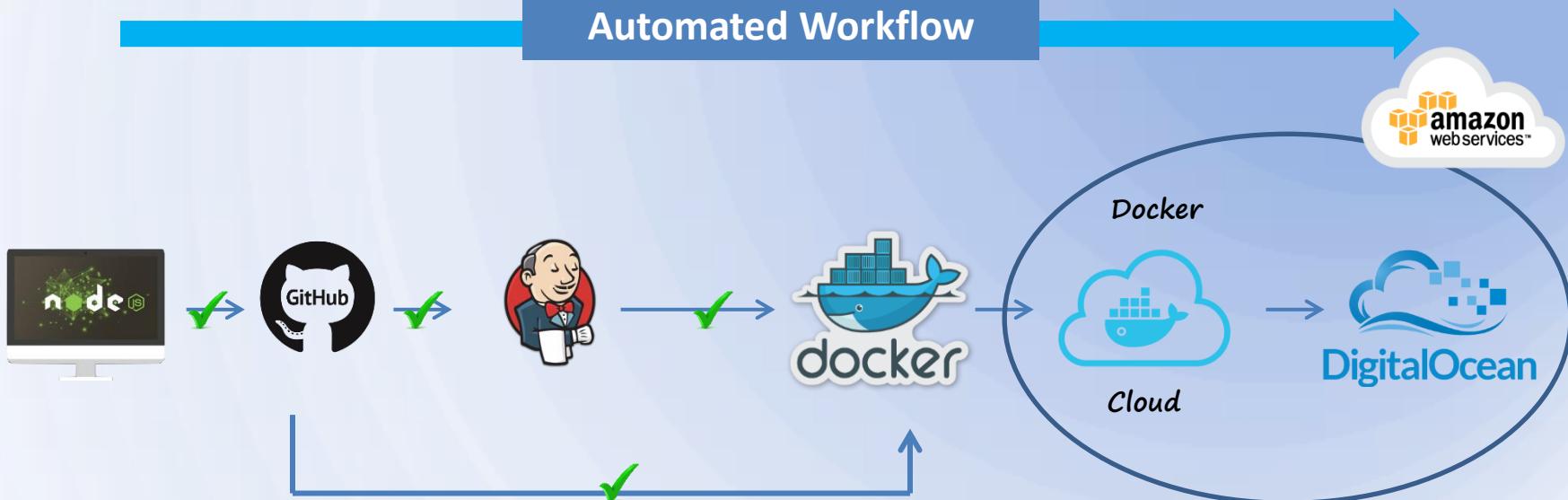
Story till now ...

Automated Workflow

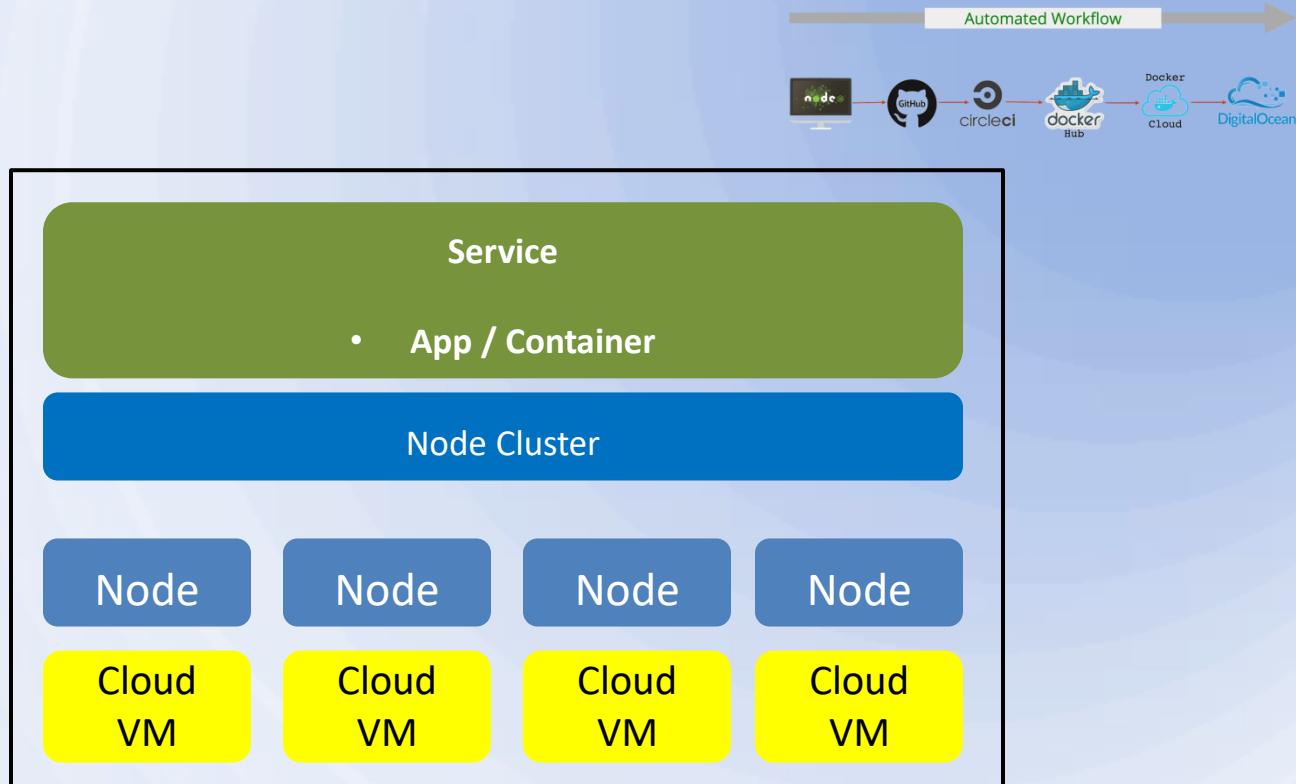




Story till now ...



Reference Architecture



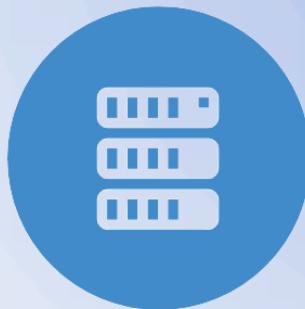
Docker Cloud

cloud.docker.com

Automated Workflow



Link a cloud provider



Deploy a node



Create a service

Everything have been taken care of by
"Docker Cloud"

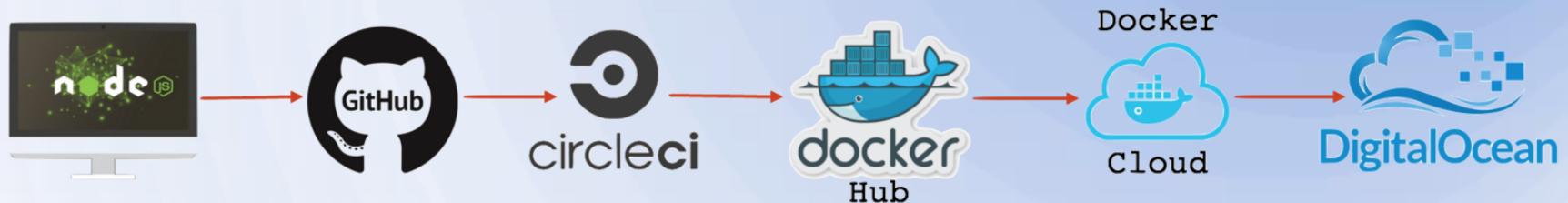


Check you should have one instance
running, created by "Docker Cloud" ->
"Node"



Incremental Change

Automated Workflow



Push a change to Github, and monitor changes at every step.

Container as a Service (CaaS)

- Registry: Docker Hub
- Clustering: Docker Swarm
- Orchestration: Docker Compose
- Deploy, manage, and monitor: Docker Universal Control Plane

Also check Kubernetes by Google and Apache Mesos to Manage a cluster of Linux containers

For Enquiry

- ✉ support@trainingbasket.in
- ☎ 9015-887-887

 **SUBSCRIBE**



THANK YOU

Visit : www.trainingbasket.in



VISIT OUR
PLACEMENT PORTAL

www.trainingbasket.in

For Reviews & Video Testimonials of our
Recently Placed Students



Thank You

For more information please visit our website
www.online.trainingbasket.in