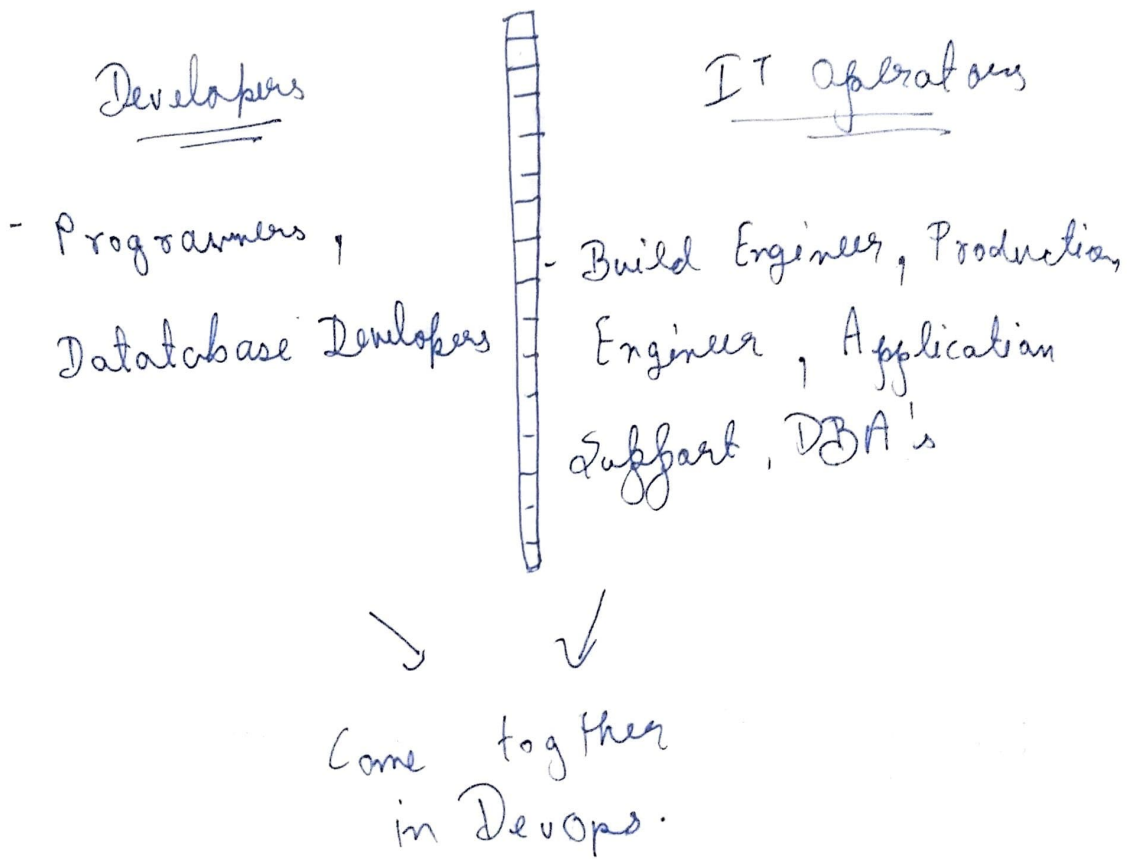


# DevOps Fundamentals

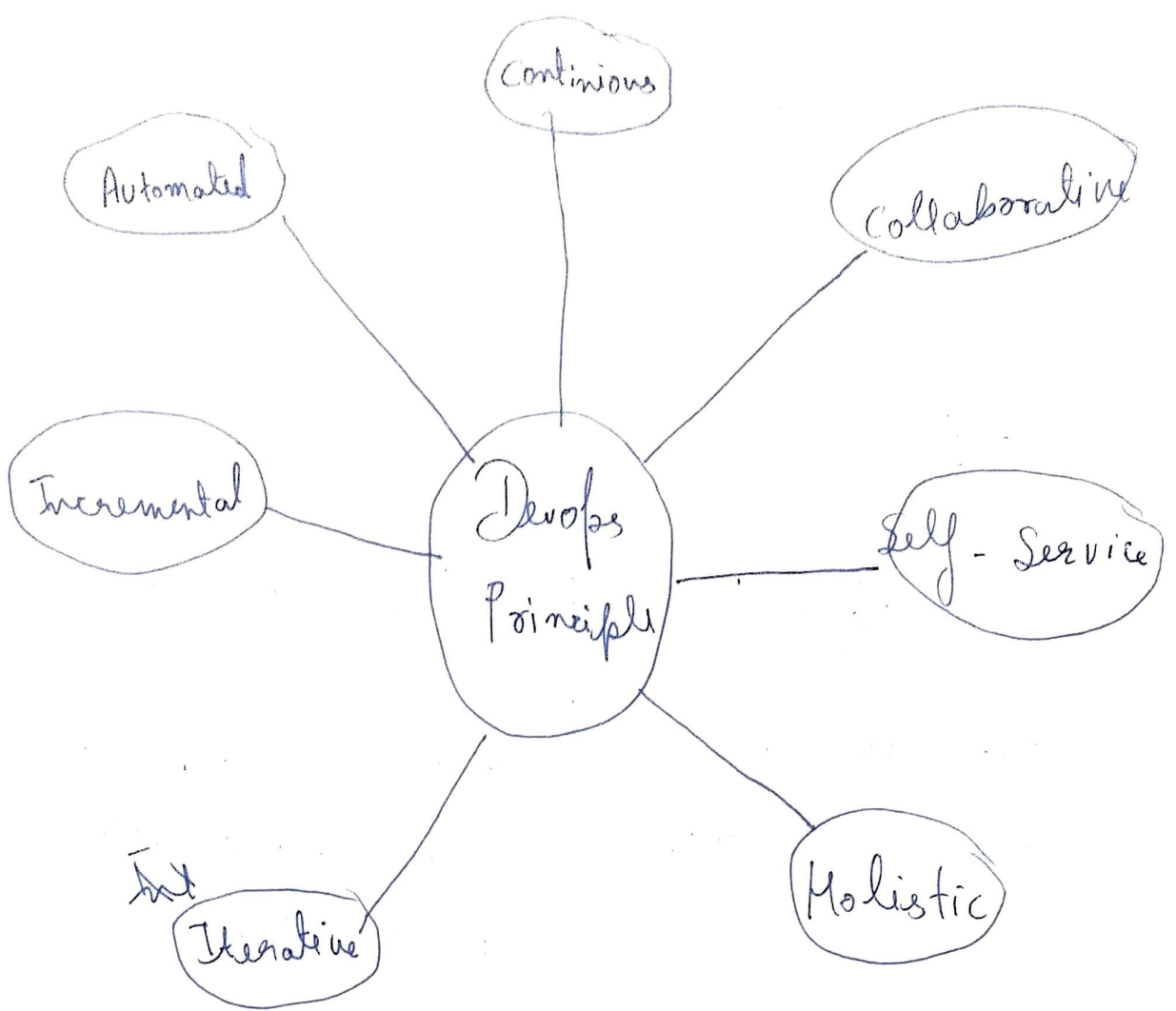
## ① Introduction



---

## ② DevOps Principals

→ Principles are the cornerstone of devops culture:



Incremental - Divide project in small tiny tasks, and deliver one at a time.

Iterative - Based on feedback of one go to another part.

Automated - Resource Efficient, Fast, Scalable, Repeatable & Less error infected.

Continuous - Integration of various & repeating this cycle multiple times a day.

Collaboration - Bringing together Developers & Operation  
teams

Self-Service - People should collaborate in a manner  
that they are interdependable.

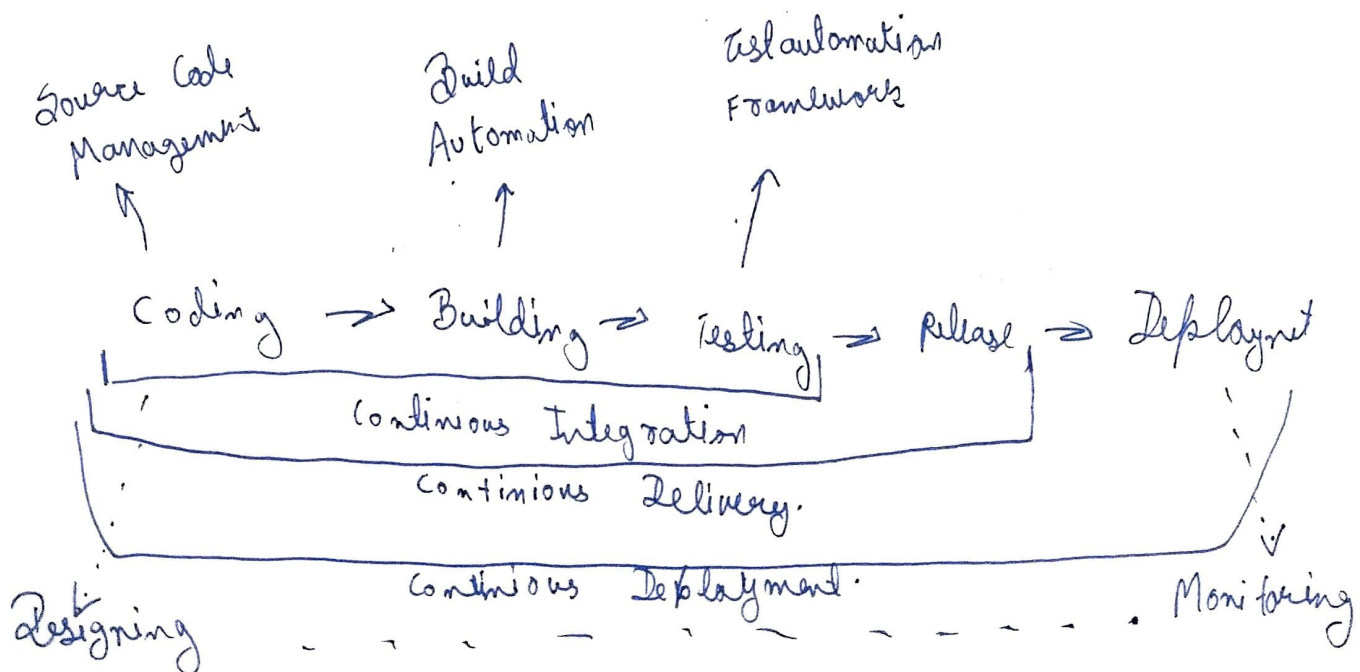
Holistic - Look at issues with better & broader  
perspective.

(Principles are still evolving)

### III. DevOps Terminology

- 2 Main principles

- Automation
- Integration



## IV DevOps Pipeline

→ Minimizing manual intervention.

→ Change in code trigger build, which when completed ~~deploys~~ testing. then server should triggers

be configured & we should continue with deployment

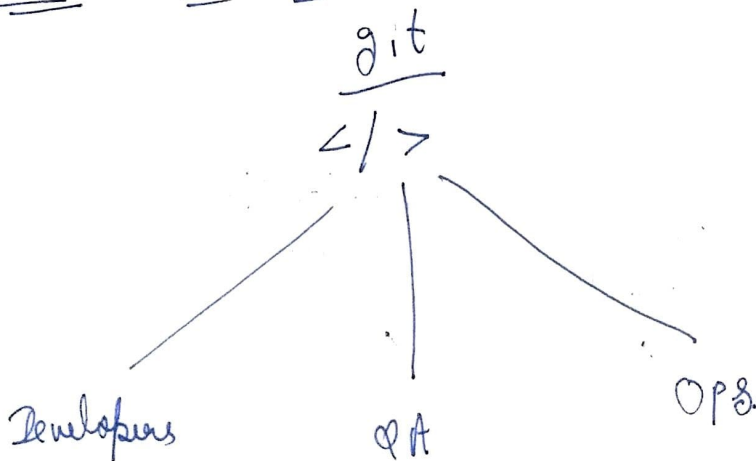
———— Pipeline ————

Monitoring

Designing

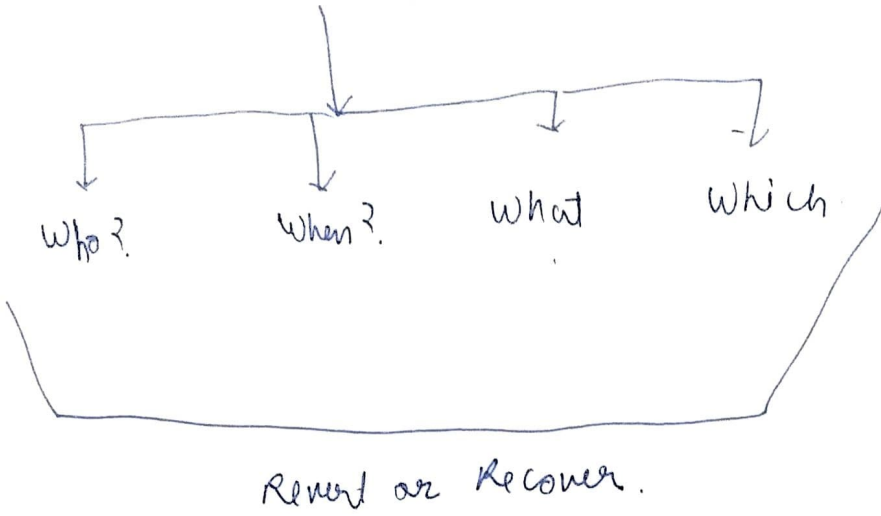
---

## V Introduction to git.



→ Version control System

↓  
Tracks



→ Git works in a distributed manner.

→ Git is used in backend.

↳ Works on ssh & http protocol.

↳ Allsian Bitbucket

↳ Github

↳ Gitlab

↳ AWS Code Commit

---

Git installation & Repository Setup

// Demo

## (VI) Git installation and repository session.

</Demo>

---

## VII Git - Tracking your changes

\* git log - gives all the info.

\* git show - all the details of changes.

\* git log --help → To view all the options.

\* git log --pretty=short = short info

\* " " = oneline = single line "

---

## (VIII) Git Understanding index area and local repository

\* vim file1.c → To add a file.

\* git status → will show uncommitted files



\* git add file3.c → suggested by git to add (green)

\* git commit -m "Adding file3.c" → will commit & shown nothing  
↓  
~~commit~~  
man to commit

\* git push → to push it to remote repository.

---

## ①x Git Branches

Branches are

→ It is a linear series of commit object.

→ Default Branch → master

→ git branch → to view all branches & master.

→ git branch devbranch → add branch ~~now~~ named devbranch

→ git checkout devbranch → change branch to

## Types of Branches

Master Branch

Feature Branch.

Develop Branch

Release Branch.

(depend on User/Company)

---

### ① Merging Branches in hit

→ git diff master ... devbranch - to check differences in branch.

→ change in file of both branches

→ git merge devbranch → devbranch will be merged into current branch.

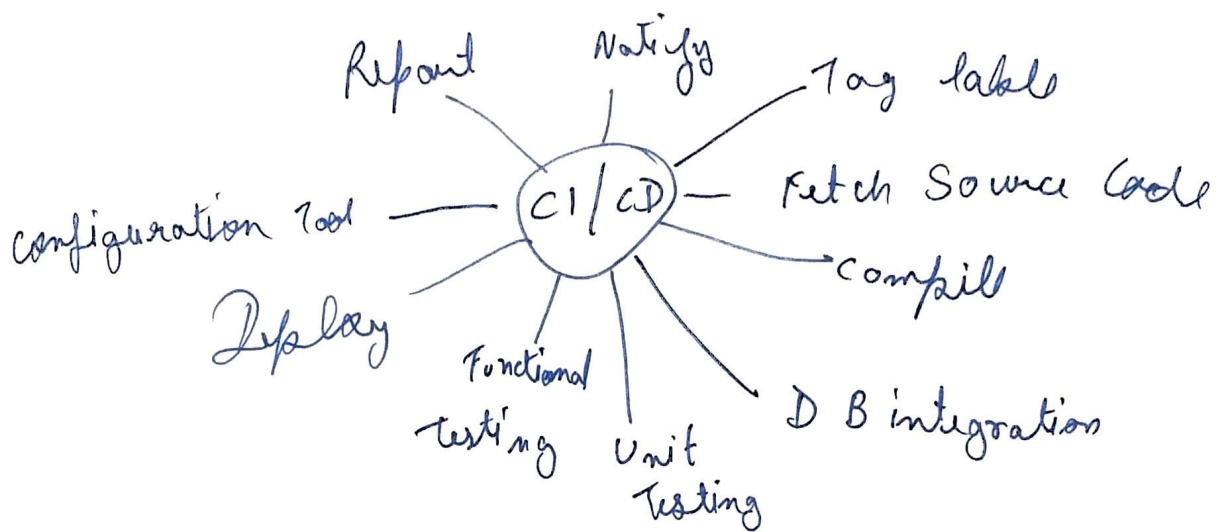
---



## \* Introduction to Jenkins

→ Jenkins makes continuous Integration  
" Delivery

" Deployment. possible.



## \* Jenkins

↳ open source

↳ easy to use

↳ extensibility (plugins)

↳ Instant Report

↳ Distributed (Master/Slave)

↳ Email Notification.

↳ customisable.

## XII Jenkins Installation

2/1 demo

## XIII DevOps Pipeline Using Jenkins

→ tree - to see whole code & directory

Structure -

↳ can add description.

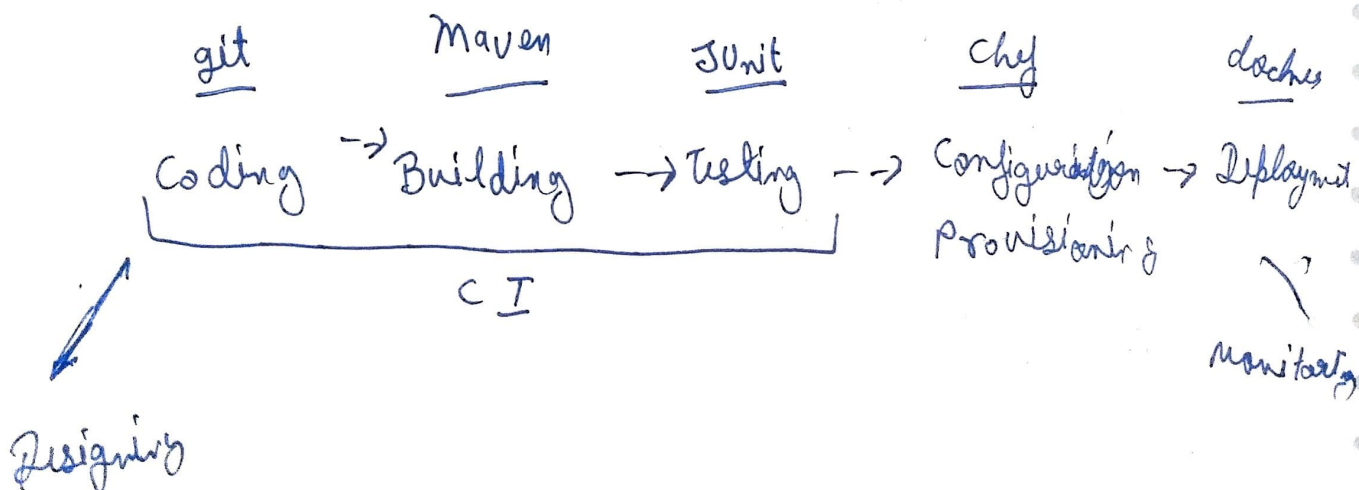
↳ give URL of Github.

↳ may build trigger.

↳ may add test cases.

↳ build.

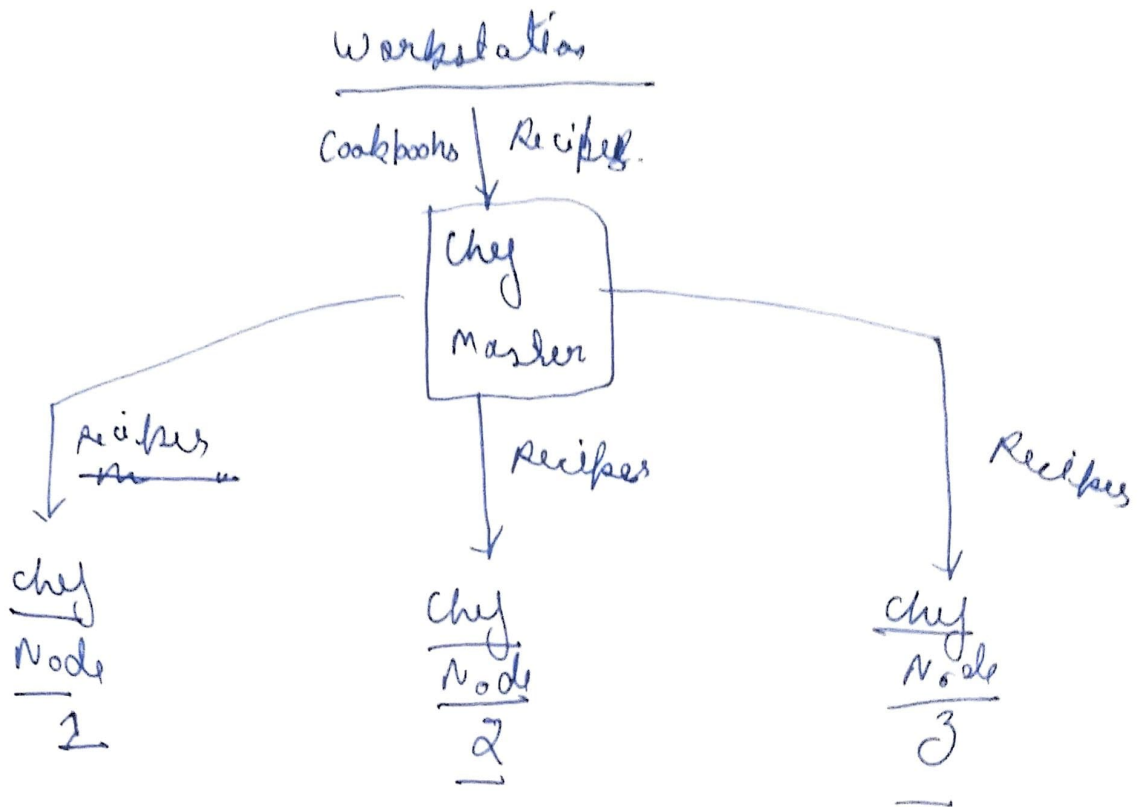
↳ add auto trigger when code is pushed



(XIV)

## Introduction to Chef.

→ Chef is basically a Infrastructure Automation & Configuration management tool.



(XV)

## Introduction to Containers

→ Containers can be called very light weight virtual machines.

→ Docker made it very easy to manage, create, or delete containers.

## \* Features of Containers

- ↳ Scalability
- ↳ Isolation
- ↳ Accurate Testing
- ↳ Replicable Environment
- ↳ Resource optimisation
- ↳ Performance
- ↳ High Availability

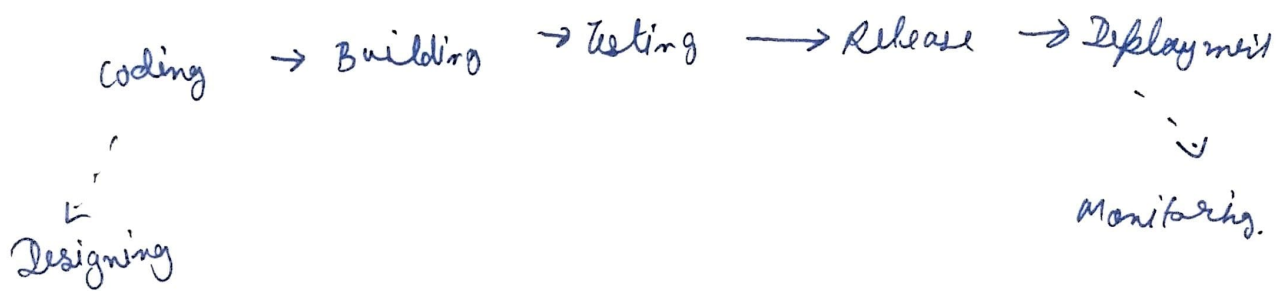
## Short Summary

→ In DevOps Developers & IT Operators come together and helps each other.

→ DevOps has few principles such as-

- Continuous, - Collaboration, - Holistic, Iterative,
- Self Service, - Incremental, - Automated, Continuous.

→ It's terminology is as:-



→ DevOps Pipeline helps minimize manual interventions

→ Git can be used in backend, it is a version control system which works on ssh & http.

→ Jenkins makes Continuous Integration, Continuous Delivery and Continuous Deployment possible. it is also used in devops pipeline with other tools such as, git, maven, junit, chef, docker etc.

→ Chef is a Infrastructure Automation & Configuration management tool.

→ Containers can be called Very light weight Virtual machines.  
→ Docker made it easy to manage create or delete Containers.