# CI/CD

Continuous Integration / Continuous Delivery or Deployment

CI = Continuous Build + Continuous Testing

# Docker

### Container

**To start docker**

Service docker start

* **To check docker status**
* **To stop docker**
* **To check existing images**
* **To create container from image**
* **To search images for a keyword on docker hub**
* **To create image from container**

# Jenkins

* Jenkins is an open-source tool, build in java that runs on windows, macOS and other Unix-like operating systems. It is free and community supported and lots of plug-ins available for it. It might be your first choice for CI.
* Its default port is 8080.
* Jenkins automates the entire cycle of software development.
* Jenkins was originally developed by Sun Microsystems under the name Hudson in 2004. Later Oracle purchased the company and changed the project name to Jenkins.
* Hudson is enterprise edition of Hudson.
* It can run on any major platform without any compatibility issue.

## Workflow of Jenkins

* We can attach git, Selenium, maven, artifactory and many other plugins to Jenkins.
* Once developer push code into GitHub, Jenkins pull that code and send to maven to build it.
* Once build is done, Jenkins pull that code and send to selenium for testing.
* Once testing is done, Jenkins pull that code and send to artifactory (archiving purpose) as per requirement and so on.
* We can also deploy that code.

## Advantages of Jenkins

* It has lots of plugins available
* You can write your own plugin for Jenkins
* You can use community plugins
* Jenkin is not just a tool but a framework too. You can do whatever you want, all you need is a plugin.
* Jenkins works on master-slave architecture. We can attach salves (nodes) to Jenkins’s master. It instructs slave to do job, if slaves are not available, Jenkins itself does the job.
* Jenkins also behaves as Crone Server replacement that can schedule jobs
* It can create labels i.e., who or which slave will do the task?

## SCM : Source Code Polling

Scheduled vs SCM

* Only Admin can create a job or add a plugin to Jenkins.
* Add “Role based authorization strategy” to add and manage users.

## Master / Slave

* Every slave has its own directory to contain its job
* Agent.jar to communicate between master and slave
* Set label to dedicate job to specific slave

## Monolithic applications

# Kubernetes

* Orchestration tool
* It’s a container management tool
* Automatically increase and decrease number of containers based on requirement
* Api gateways to communicate microservices to each other
* Open-source container management tool which automates container deployment, containers scaling and load balancing.
* It schedules, runs and manages isolated containers which are running on virtual/physical/cloud machines.
* All top cloud providers support k8s
* It is written in Golang language.
* You can write script in Kubernetes script in JSON or Yaml

### Problems with scaling up the containers

* Containers cannot communicate with each other
* Auto scaling and load balancing was not possible
* Containers had to be managed carefully

### Features of Kubernetes

* Orchestration (clustering of any number of containers running on different network)
* Auto scaling (vertical and horizontal scaling)
* Auto healing
* Load Balancing
* Platform independent (virtual machine/physical machine/cloud)
* Fault tolerant (Node/Pod failure)
* Rollback (going back to previous version of application)
* Health monitoring of containers
* Batch execution (one time, sequential, parallel)

|  |  |  |
| --- | --- | --- |
| **Feature** | **Kubernetes** | **Docker Swarm** |
| Installation and cluster configuration | Complicated and time consuming | Fast and easy |
| Supports | K8s can work with almost all container types like docker, Rocket and containerD | Works only with Docker |
| GUI | Available | Not available |
| Data volumes | Only shared with containers in same pod | Can be shared with any other container |
| Updates and Rollback | Process scheduling to maintain services while updating | Progressive updates and services health monitoring through the update |
| Autoscaling | Supports vertical and horizontal scaling | Does not support autoscaling |
| Logging and monitoring | Have inbuilt tool | Use 3rd party tool |

## Kubernetes Architecture

* Its used master/slave or server/client architecture.
* Its cluster can have one master -> one node or one master -> multiple nodes or multiple masters -> multiple nodes
* Kubernetes talks to pod instead of container and pod talks to container
* Generally, we have one container in one pod

Master Node

Controller manager