

Contents:

J query -js library//**Not Done**

Bootstrap –framework//**Not Done**

IntelliJ// **Not Done**

Title	Technology/Frameworks
1.Java	a)Core Java b)Advance Java c)JSP
2.Frame Works & Servlets	Spring ,Springboot,Microservices
3)Bug Fixing	Debugging
4)Front-end	HTML,CSS,JAVASCRIPT,REACT
5)Message Queues Services	Apache Kafka, Rabbit MQ,Putty
6)Data Bases	Mysql -database,PostgresSQL -database MongoDb –database,Oracle,Redis -cache , ScyllaDB Cassandra.
7)Servers	Eureka Server
8)Gate Ways	Spring cloud gateways , Zuul gateway
9)Spring Application Testing	Postman – Test the Rest Api's, JMeter Swagger - Provide Documentation of Rest Api
10)Build Tool	Maven,Gradle.
11)Deployment Tool	Docker- deveops Tool
12)Source control	Git & GitHub,Tortoise Svn
13)IDLE(Integrated Development and Learning Environment)	Ellipse,STS,IntelliJ
14)Code Review	Sonar Cubes

Java Version & LTS Product

1 .jdk8-LTS, Jdk-11, Jdk-17 & Latestjdk-21,industry Standards jdk-8

2.Is Java 1.8 the same as Java 8?

Ans: Yes. Oracle uses the version string 1.8 to refer to Java 8

Terminal Commands:

systemctl restart tomcat	-for tomcat restart
systemctl start tomcat	-for tomcat start
systemctl stop tomcat	-for tomcat stop
ifconfig	- for check ip address
tail -f /opt/tomcat/logs/catalina.out	- for check logs
ssh username@ipaddress	- connect to another system
ping 10.10.10.98	- for ping ip address
history	- for checking history
clear	- for clear the screen
cd	- for change directory
ls	-for see all folders and files available in directory
pwd	- for show the current working directory
mkdir <directory name>	- for create new directory
rmdir <directory name>	- for delete directory
touch <file name>	- for create the new file
cat <file name>	-for display content of file
rm <file name>	- for remove the file
head <file name>	-for displays the first 10 lines of a file
tail <file name>	-for displays the last 10 lines of a file
locate <file name>	-for search the file
telnet 192.168.1.180 9760	- for socket connection
../	- for see the back directorys
cd ..	- for back to previous directory
vim bplconfig.properties	- open the particular file
Esc -> :qa	- exit a file without editing
Esc -> I -> edit ->Esc ->:wq!	- save the edited file
cd /mnt/c	- for navigate to linux terminal to access windows c
drive	

Spring & Springboot Latest Version

Spring Latest Version:6.0.1

Springboot Latest Version:Spring Boot3.1.3 .

windows commands:

ipconfig	-for check ip address
systeminfo	-for getting total system information
cls	-for clear the screen
win+R MRT	Malicious Remove Tool

Putty

Push data to kafka topic manually :

1. connect to kafka server using putty.
2. Enter username and password
3. Go to path **cd /home/kafka/kafka/bin**
4. **./kafka-console-producer.sh --broker-list 10.10.10.98:9092 --topic <topicName>**

Git:

1. **git init** -Create a Repository for a Blank
2. **git add <file name>** - adds files to the staging area
3. **git status** -list all the untracked files
4. **git branch** - to check current branch
5. **git checkout <branch name>** - to switch branch

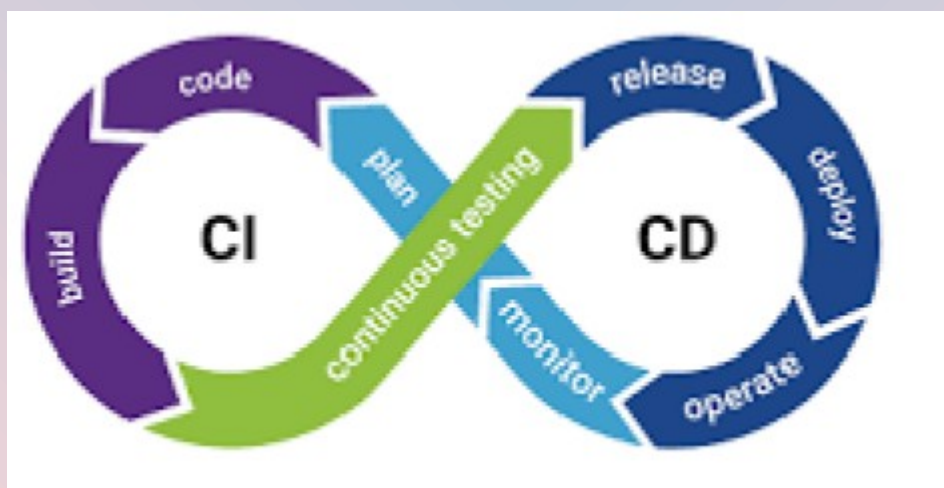
Windows Subsystem for Linux (WSL) :

It provides a native integration of Linux tools and utilities alongside Windows applications, making it easier for developers and system administrators to work with both environments simultaneously.

1. apps&features ->optional features ->more window features->enable 'windows subsystem of linux'
2. microsoft store-> install ubuntu.

Jenkins

- Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on
- The Jenkins project was originally named Hudson, and was renamed in 2011 after a dispute with Oracle.
- Run Jenkins as a Managed / SaaS Service in the Cloud .
- Jenkins is a tool that is used for automation, and it is an open-source server that allows all the developers to build, test and deploy software.
- Jenkins requires Java 11 or 17 since Jenkins 2.357 and LTS 2.361.1
- Jenkins job's build name would be named with the numeric number by default in increment manner .



What language is used in Jenkins script?

Jenkins is an automation server with hundreds of plugins, there is a huge number of technologies involved.

Is Jenkins a REST API?

It is up to the user to parse the JSON and extract the relevant pieces.

Interestingly, Jenkins itself has a REST API

Jenkins architecture has two components:

1. Jenkins Master/Server.
2. Jenkins Slave/Node/Build Server.

Steps For Downloading & Installing Jenkins In Windows

Step-1:

<https://mirrors.tuna.tsinghua.edu.cn/jenkins/windows-stable/2.414.1/jenkins.msi>

Step-2; After Installation Jenkins, go to Jenkins Installation directory Path

Navigate to Cmd

ex;c:\programFiles\Jenkins

Aftet Navigate to Cmd:execute Jenkins War file by following Command

java -jar Jenkins.war httpPort=8085[while installing it ask Select port use same port here]

it will download some plugins takes some time,after pugins installation done

by default http://localhost:pport number/

it will ask administrative password to creat jenkins account to get Password enter following path in file serach bar

C:\ProgramData\Jenkins\.jenkins\secrets\

initialAdminPassword

paste the password in Jenkins UI, select recommended plugin

Docker:

Docker is a platform for packaging, deploying, and running applications without any compatibility issues.

Docker is a container management service. Used for easy deployment process.

Dockerfile Build-----> dockerimage Run-----> docker container
--

Download & installation of Docker:

Go to official website download & install docker and sign in with username and password.

User name : ravulabikshapathi

Password: docker@123

Steps To use:

1. Develop the spring boot application.
2. Create the Dockerfile .
3. Goto project folder directory and open cmd.
4. Build the image from Dockerfile using **docker build -t <imagename> .**
5. Run the image using **docker run -p 9090:9090 <imagename>**

commands:

docker -v	- for checking version
docker help	- display all docker info
docker login	- login into docker
docker build -t imagename .	- build the docker image from docker file (here . is docker file directory)
docker images	- for view all images
docker run -p 9090:9090 imagename	- command for run the image
docker rmi imageId	- for removing docker image
docker create imageId	- create container for particular image
docker ps -a	- for view the all containers
docker start containerId	- for start the particular container
docker stop ContainerID	- for stop the particular container
docker rm ContainerID	- for remove the particular container
docker ps	- for view the running container

ScyllaDb

- **ScyllaDB** provides an advanced I/O scheduler to manage concurrency, balancing overall throughput with lowest latencies while accessing persistent storage.
- ScyllaDB runs over the same HTTP/HTTPS style connection and supports the drivers and JSON-style queries that DynamoDB does.

ScyllaDB's primary data model is CQL-based, it also offers support for storing JSON data. This means you can store JSON documents as values within columns of specific data types (e.g., text or varchar).

Note: JSON support is more about allowing you to store JSON data within a CQL-based schema.

- Data is distributed across clustered servers per CPU core and free from resource sharing operations.
- Capable of 1M+ ops per server, ScyllaDB scales linearly from gigabytes to petabytes easily, without interruption.

ScyllaDB is a **wide-column NoSQL database** fully compatible with Apache Cassandra and Amazon DynamoDB.

- ScyllaDB Data modeling different from Relational DataBase Modeling.
- ScyllaDB based Around Queries.

CQL, CQLSH, and Basic CQL Syntax

Cassandra Query Language(CQL):: It is a query language that is used to interface with ScyllaDB. It allows us to perform basic functions such as insert, update, select, delete, create, and so on.

CQL-Based Data Model: ScyllaDB uses a CQL-based data model as its primary way of defining tables, schemas, and interacting with data. CQL is similar to SQL and is used to create tables, define column families, set data types, and perform CRUD (Create, Read, Update, Delete) operations on data.

Structured Data: In a CQL-based model, data is structured and organized into tables with predefined columns and data types. Each row in a table adheres to the schema defined for that table.

CQLSH: CQLSH (Cassandra Query Language Shell) is a command-line tool that is used to interact with Apache Cassandra and ScyllaDB databases using the CQL (Cassandra Query Language).

- It provides a way to execute CQL commands and queries against the database, perform administrative tasks, and manage database operations.
- CQLSH to execute various database operations, including creating keyspaces, defining column families (tables), inserting, updating, and deleting data, querying data, and creating secondary indexes.

Scylla DB Using Docker

step-1: Install Docker on windows

Step-2: login docker Account on Desktop.

Step-3:install python for Library Support

step-4:download cqlsh <https://downloads.datastax.com/enterprise/cqlsh-astra-20221114-bin.tar.gz>

step-5:open cmd type docker login

Step-6:enter docker username & password

step-7:pull the scylla dockerimage on cmd by

```
docker pull scylladb/scylla
```

step-8: Run the ScyllaDB Container

```
docker run -d --name scylla-container --network scylla-net \
  -p 9042:9042 -p 7000:7000 -p 7001:7001 -p 7199:7199 \
  -e SCYLLA_API_ADDRESS=0.0.0.0 \
  scylladb/scylla
```

step-9: Access ScyllaDB

```
docker exec -it scylla-container cqlsh
```

step-10: create keyspace

```
CREATE KEYSPACE IF NOT EXISTS keyspace_name
  WITH replication = {
    'class': 'replication_strategy',
    'replication_factor': replication_factor
  };
```

Springboot Application.properties

```
spring.data.cassandra.keyspace-name=keyspace_name
spring.data.cassandra.contact-points=host or IP
spring.data.cassandra.port=casandar port default:9042
```

Dependency:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-cassandra</artifactId>
</dependency>
```

RabbitMQ

RabbitMQ acts as a broker of messages which sits in between applications and allows them to communicate in asynchronous and reliable way.

Setup Java Development Kit ([JDK](#))

If you are running Windows and have installed the JDK in C:\jdk-11.0.11,

RabbitMQ Installation

Step-1:we need to download Erlang

https://github.com/erlang/otp/releases/download/OTP-26.0.2/otp_win64_26.0.2.exe

Step-2:

Download the RabbitMQ Latest binary from its [official downloads page](#) We've downloaded 3.9.13 as rabbitmq-server-3.9.13.exe for windows

<https://www.rabbitmq.com/install-windows.html#installer>

Step-3:

=>Navigate RabbitMQ Server path for windows

ex:C:\Program Files\RabbitMQ Server\rabbitmq_server-3.12.4\sbin

By default, RabbitMQ works as windows service. To enable Web based Administration UI, following step are needed.

1.C:\ProgramFiles\RabbitMQServer\rabbitmq_server-3.9.13\sbin>rabbitmq-plugins.batenable rabbitmq_management

2.[C:\Program](#)Files\RabbitMQServer\rabbitmq_server-3.9.13\sbin>rabbitmq-plugins enable rabbitmq_shovel rabbitmq_shovel_management

Step-4:after Navigating to RabbitMQ Server path

By Default server will start automatically after RabbitMQ Installation

Stoping:RabbitMQ Service – stop

Swagger:

Used to provide the documentation (Details) about the Rest Api to the Client , and also used to test the Rest Api.

Steps To use:

1.Add the below dependecys for spring boot project i.e springfox-swagger2 and springfox-swagger-ui.

```
<dependency>
  <groupId> springfox</groupId>
  <artifactId>springfox-swagger-ui</artifactId>
  <version>2.6.1</version>
</dependency>
```

```
<dependency>
<groupId>io.springfox</groupId>
<artifactId>springfox-swagger2</artifactId>
<version>2.6.1</version>
</dependency>
```

2.Create configuration for swagger

```
@Configuration
@EnableSwagger2
public class SwaggerConfig {
    @Bean
    public Docket restApi() {
        return new Docket(DocumentationType.SWAGGER_2)
            .select()
            .apis(RequestHandlerSelectors
                .basePackage("com . project_Name . controller"))
            .build();
    }
}
```

3. For swagger Ui api documentation : <http://localhost:8080/swagger-ui.html>

4.For swagger json api documentation : <http://localhost:8080/v2/api-docs>

Redis:

Redis cache is a popular and powerful in-memory data structure store that is used as a cache.

We can store data in Strings, hashes, lists, sets format. It stores data in Key , value pairs <k,V>

It is used like In memory Database and cache and message broker

In Memory Database:

1.Download and Install Redis server <https://github.com/tporadowski/redis/releases>

2.Add the dependencies

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

3.Add in Application.properties

```
spring.redis.host localhost
spring.redis.port 6379
spring.cache.type redis
spring.cache.redis.cache-null-values true
#Automatically erase data after specific time
spring.cache.redis.time-to-live 60000
```

4. Create Configuration for Redis

@Configuration

```
public class AppConfig {  
    // Redis Connection  
    @Bean  
    public RedisConnectionFactory cf() {  
        return new LettuceConnectionFactory();  
    }  
    // Redis Template  
    @Bean  
    public RedisTemplate<String, Student> rt() {  
        RedisTemplate<String, Student> template = new RedisTemplate<>();  
        template.setConnectionFactory(this.cf());  
        return template;  
    }  
}
```

5. And create dao classes and perform operation

@Resource(name = "rt")

```
private HashOperations<String, String, Student> opr;
```

Cache Management:

It is a Temp memory, it is used between Server (App) and database.

This is used to reduce network calls between server and database if we are trying to fetch same data multiple times from database.

Do not implement cache for all modules select modules which are actually mostly used.

Eg: Gmail -> Inbox

@EnableCaching // Enable cache

@Cacheable (value = "employee")

//Annotation used for store the

cache

@CachePut (value = "employee", key = "#id") //Used for Update cache

@CacheEvict (value = "employee", key = "#id", allEntries = true) //Used for delete cache

Eureka Naming Server:

- Eureka is a server
- Eureka server acts as a central registry that keeps track of all the microservices instances running in the system. Each microservice instance (Eureka client) registers itself with the Eureka server when it starts up, and it also sends periodic heartbeats to indicate that it is still alive and functioning.
- The main purpose of Eureka Server is to facilitate service registration, discovery, and load balancing in a distributed system.

Create Eureka server and client applications then client will register in server.

Eureka Server Property file

```
eureka.client.fetch-registry= false
eureka.client.registerWithEureka=false
server.port= 8761
spring.application.name= Server
```

```
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-
server<artifactId>
      </dependency>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-zuul</artifactId>
      <version>2.2.10.RELEASE</version>
    </dependency>
```

Application.property file for service Register On Eureka

```
eureka.client.serviceUrl.defaultZone: ${EUREKA_URI:http://localhost:8761/eureka}
eureka.instance.prefer-ip-address=true
```

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-eureka-client<artifactId>
  </dependency>
```

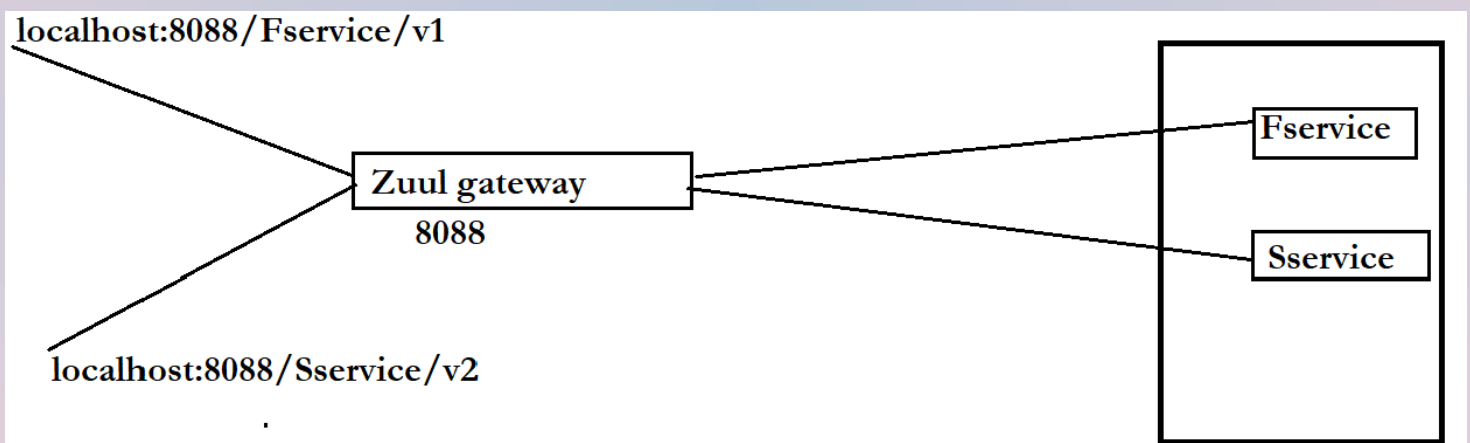

API Gateways :

- The API Gateway is a server .It is a single entry point into a system. API Gateway encapsulates the internal system architecture.
- All the requests made by the client go through the API Gateway. After that, the API Gateway routes requests to the appropriate microservice.
- The API Gateway can also perform various functions such as authentication, load balancing, caching, request routing, and request/response transformation.

Some of Api gateways :

Spring cloud api gate way , Netflix Zuul api gateway, Kong Api gateway

Netflix Zuul api gateway



Gate Way Property file

```
spring.application.name= zuul-gateway
server.port=8088
##### Zuul Configuraiton #####

zuul.ignoredServices=*
#ribbon.eureka.enabled=false
zuul.routes.Fservice.path=/Fservice/v1
zuul.routes.Fservice.url=http://localhost:8081
```

```
zuul.routes.Fservice.serviceId=Fservice
zuul.routes.Fservice.stripPrefix=false
```

```
zuul.routes.Sservice.path=/Sservice/v2
zuul.routes.Sservice.url=http://localhost:8082
zuul.routes.Sservice.serviceId=Sservice
zuul.routes.Sservice.stripPrefix=false
```

First Service: Controller

```
@RestController
@RequestMapping("Fservice")
public class FController {
    @GetMapping("/v1")
    public String requestMethodName() {

        return "Hello this first service";
    }
}
```

Second Service:Controller

```
@RestController
@RequestMapping("Sservice")
public class SControl {
    @GetMapping("v2")
    public String requestMethodName() {
        return "{ \"msg\": \" Hello this is second
Service\"}";
    }
}
```

```
}
```

```
package com.microservices.netflixzuulapigatewayserver;  
  
import org.springframework.boot.SpringApplication;  
  
import  
org.springframework.boot.autoconfigure.SpringBootApplication;  
  
import  
org.springframework.cloud.client.discovery.EnableDiscoveryClient  
;  
  
import  
org.springframework.cloud.netflix.zuul.EnableZuulProxy;  
  
@SpringBootApplication
```

```
@EnableZuulProxy
```



```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-netflix-zuul</artifactId>  
  <version>2.2.10.RELEASE</version>  
</dependency>
```

```
public class NetflixZuulApiGatewayServerApplication  
{  
  
    public static void main(String[] args)  
    {  
  
        SpringApplication.run(NetflixZuulApiGatewayServerApplica  
tion.class, args);  
  
    }  
  
}
```

J Query:

jQuery is a small and lightweight JavaScript library.
jQuery simplifies AJAX call and DOM manipulation.

React Js :

- It is a JavaScript library .**Facebook** developed ReactJS in **2011** in its newsfeed section, but it was released to the public in the month of **May 2013**
- It uses virtual DOM instead of regular DOM.
- DOM is an object which is created by the browser each time a web page is loaded
- It dynamically adds or removes the data at the back end and when any modifications were done, then each time a new DOM is created for the same page. This repeated creation of DOM makes unnecessary memory wastage and reduces the performance of the application.

Download & Installation :

1. NodeJS and NPM are the platforms need to develop any ReactJS application.

node -v - Check node js version

npm -v Check the npm version

node js : Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to run JavaScript code outside of a web browser.

JavaScript was mainly used for client-side scripting, running within web browsers to enhance web pages and provide interactivity.

Node.js extends JavaScript's capabilities by enabling it to be executed on the server side as well.

npm (Node Package Manager): It is a command-line tool that comes bundled with Node.js and is used to manage and install packages (libraries and modules) written in JavaScript.

npm allows developers to easily install, update, and remove packages and their dependencies.

props V/s state:

PROPE	STATE
<ul style="list-style-type: none">• The Data is passed from one component to another.	<ul style="list-style-type: none">• The Data is passed within the component only.
<ul style="list-style-type: none">• It is Immutable (cannot be modified).	<ul style="list-style-type: none">• It is Mutable (can be modified).
<ul style="list-style-type: none">• Props can be used with state and functional components.	<ul style="list-style-type: none">• The state can be used only with the state components/class component (Before 16.0).

****Latest Hooks**

*****Hooks cannot be used with class components*****

Why the need for Hooks?

- ◆ Use of '*this*' keyword:

classes don't concise efficiently which leads to hot reloading being unreliable which can be solved using Hooks

- ◆ **Reusable stateful logics**

Here is no particular way to reuse stateful component logic to React.

implemented using custom React Hooks and Context API. React Hooks, introduced in

React 16.8, provide a way to add state and lifecycle features to functional components,

making it easier to create reusable stateful logic.

What is React Router?

- React Router is a standard library for routing in React.
- It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

Installing React Route

```
npm i react-router-dom
```

Importing React Router

```
// Importing  
import { BrowserRouter, Routes, Route } from "react-router-dom";
```

- **BrowserRouter:** BrowserRouter is a router implementation that uses the HTML5 history API(pushState, replaceState, and the popstate event) to keep your UI in sync with the URL.
- **Routes:** It's a new component introduced in the v6 and an upgrade of the component. The main advantages of Routes over Switch are:
- **Link:** The link component is used to create links to different routes and implement navigation around the application. It works like an HTML anchor tag.

```
<Router>  
  <div className="App">  
    <ul className="App-header">  
      <li>  
        <Link to="/">Home</Link>  
      </li>  
      <li>  
        <Link to="/about">About Us</Link>  
      </li>  
      <li>  
        <Link to="/contact">Contact Us</Link>  
      </li>  
    </ul>  
    <Routes>  
      <Route exact path="/" element={< Home />}></Route>  
      <Route exact path="/about" element={< About />}></Route>  
      <Route exact path="/contact" element={< Contact />}></Route>  
    </Routes>  
  </div>  
</Router>
```

Functional Component V/s Class Component

Functional Component	Class Component
A functional component Accepts Props As param, returns React Element(JSX)	class component requires you to extend from React. Component and create a render function that returns a React element.
There is no render method used in functional components.	It must have the render() method returning JSX (which is syntactically similar to HTML)
Functional components run from top to bottom and once the function is returned it can't be kept alive.	The class component is instantiated and different life cycle method is kept alive and is run and invoked depending on the phase of the class component.
React lifecycle methods (for example, componentDidMount) cannot be used in functional components.	Stateful components because they implement logic and state.
Hooks can be easily used in functional components to make them Stateful. Example: <pre>const [name,SetName]= React.useState(' ')</pre>	It requires different syntax inside a class component to implement hooks. Example: <pre>constructor(props) { super(props); this.state = {name: ' '} }</pre>
Constructors are not used.	Constructor is used as it needs to store state.

kubernetes

- **Kubernetes** is a tool that helps us to run and manage applications in containers. It was developed by Google Lab in 2014, and it is also known as k8s.
- Platform that automates the **deployment, management, and scaling** of container-based applications in different kinds of environments like physical, virtual, and cloud-native computing foundations.
- Containers are isolated from each other so that multiple containers can run on the same machine without interrupting anyone else.
- It allows us to deploy and manage container-based applications across a Kubernetes cluster of machines.

The benefits of **Kubernetes** are:

- Scalability
- High availability
- Portability
- Security

Installation & Configuration Of Kubernetes

1. Play-with-K8s: Ready Made kubernetes Cluster, No need Of Implementation

For installation go Through the following link

<https://labs.play-with-k8s.com/>

Login with either Docker or Github

→ Create Instance & start

kubernetes architecture

