**Microservice Architecture:**

Main objective of the Microservices Architecture is to split up the application as separate services(Spring Boot Applications) for each REST API service functionality and can be deployed independently.

Microservices architecture allows avoiding monolith application for large system. It provides loose coupling between collaborating processes which running independently in different environments with tight cohesion.

**Advantage of Microservice Architecture over Monolithic Architecture:**

1. Independent deployment, since each Microservice is an individual Spring Boot Application, hence changes of one MIcroservice can be deployed without affecting other Microservices
2. Ease for Developer as a Developer need not have entire code base on a single IDE, he/she need to develop/change/build code of his service
3. Build/Deployment time reduce
4. Easy to scale as individual component. And each MicroService can be scaled differently, based on the load it expects.
5. Technology diversity i.e. we can mix libraries, languages, databases, frameworks etc.
6. Fault isolation i.e. a process failure should not bring whole system down.
7. Better support for smaller and parallel team.
8. Release Cycle of each Microservice can be different

**Spring Cloud Tools**

Spring Cloud Framework provides below Tools to facilitate Microservice Architecture

|  |  |  |
| --- | --- | --- |
| **Spring Cloud Tool Name** | **Purpose** |  |
| **Zuul Proxy** | **API Gateway** |  |
| **Eureka Service Registry/Discovery** | **Service Registry and Discovery** |  |
| **Ribbon** | **Load Balancer** |  |

For more details, refer:

PPT(Mandatory): <https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/MicroServices/Microservice_Introduction%20(1).pptx>

RestTemplate examples(Mandatory): <https://drive.google.com/open?id=1qxh9YiGnfedMSzHDf-dOYP9ft1TQRnhr>

FeignClient examples: <https://www.javacodegeeks.com/2018/10/making-rest-communication-easy-with-feign-clients.html>

More Microservices details(Optional): <https://drive.google.com/open?id=1CxVUCI4dKsheferXDzOUPYxIt_Xj9Ivf>

**Jenkins:**

Jenkins is a CI/CD(Continuous Integration/Continuous Deployment) Tool, which automatically starts build whenever source code is committed to Git. Advantages are detecting build failures immediately so that respective developer can fix and check in the fixes. Earlier integration was done manually, which causes multiple integration issues(like build failures) at later point of time, it would be difficult to back track older changes, and may delay Project delivery and other milestones.

Steps in Apart from building includes Perform Unit Testing, deploying, etc...

Video(Mandatory): <https://www.youtube.com/watch?v=-GsvomI4CCQ>

Jenkins Setup Step by Step(Mandatory): <https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/Jenkins/Jenkins.pptx>

CI/CD Basics(Optional): <https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/Jenkins/ci_cd_brief1.pdf>

**Docker:**

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

By doing so, the developer can rest assured that the application will run on any other Linux machine regardless of any customized settings that machine might have that could differ from the machine used for writing and testing the code.

**Advantages of Dockerizing:**

1. Fast Deployment since it is single image file.
2. Deployment is less error prone
3. Docker Image can be deployed on Docker Host irrespective of Technology

**Docker Concepts:** <https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/DockerBasics/1_Docker_Concepts.docx>

**Docker Commands:**

<https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/DockerBasics/2_Docker_commands.pptx>

**How to Dockerize simple Spring Boot Application:** <https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/DockerBasics/3_DOckerize_spring_boot_app.docx>

**How to Setup Docker Client(on IIHT Cloud):**

<https://github.com/vskreddy652/Genc_BatchB/blob/master/ConceptsAndExamples/DockerBasics/Docker%20Remote%20Host%20Access%20%20Steps.docx>

**More details(Optional):**

<https://drive.google.com/open?id=1-0fNQ_ZLQaBRNMve6ZcdTbOsvBoxFcGX>

**JMeter:**

JMeter is a Performance or Load or Stress Testing Tool

**Video(Mandatory):** <https://www.youtube.com/watch?v=9433A7jHe2U>