Pre-requisite:
    REST API Development

Microservices architecture
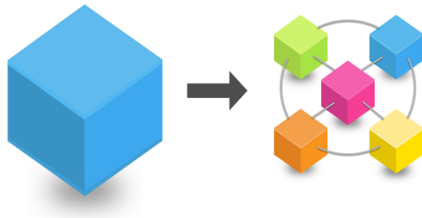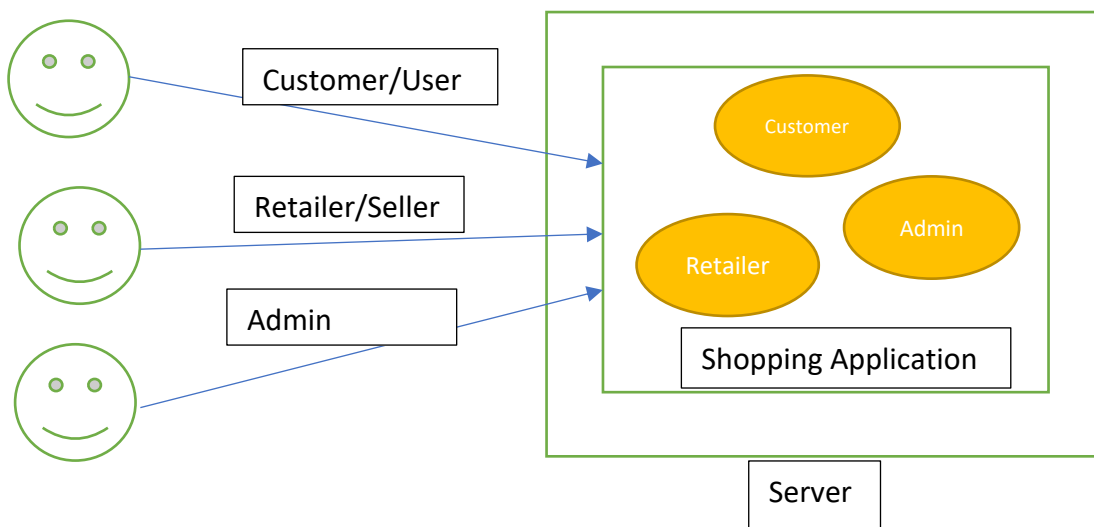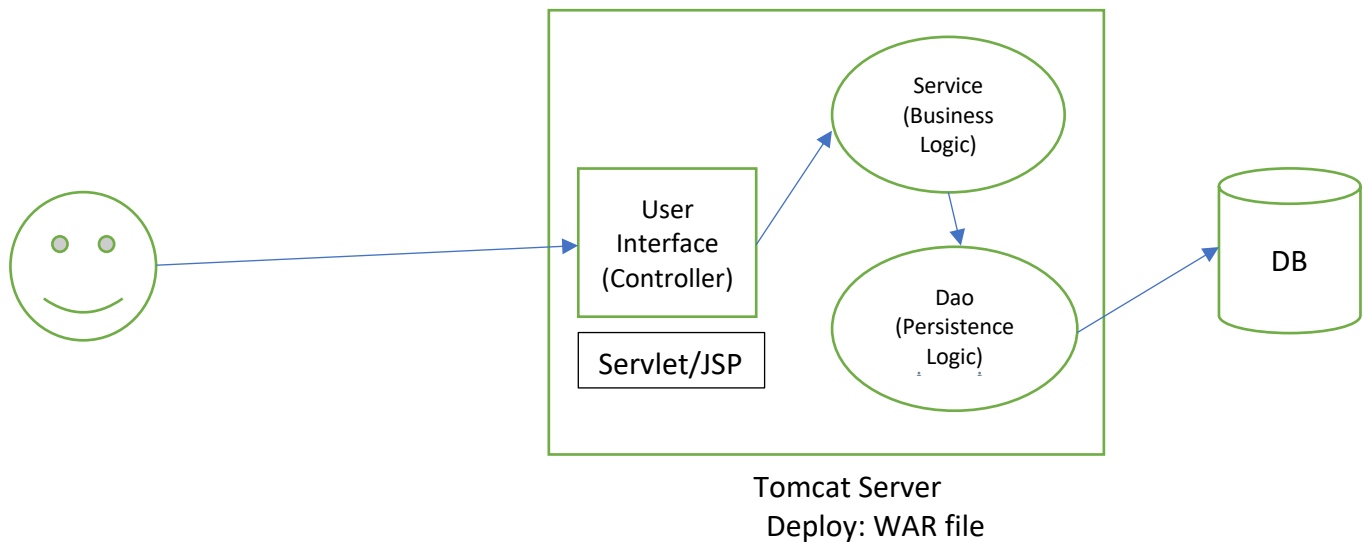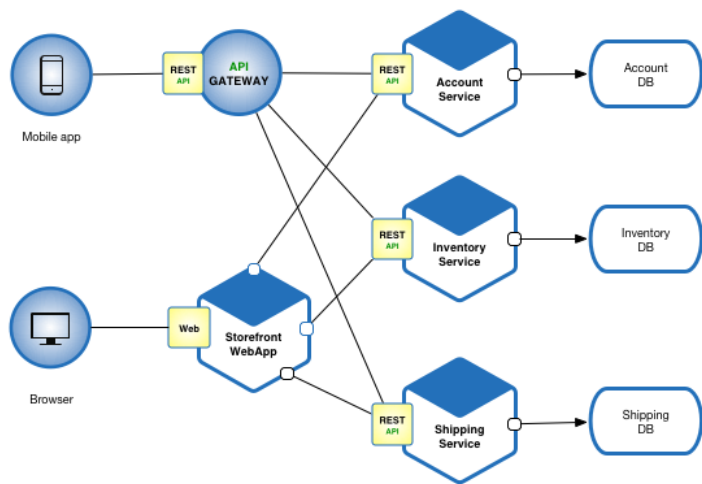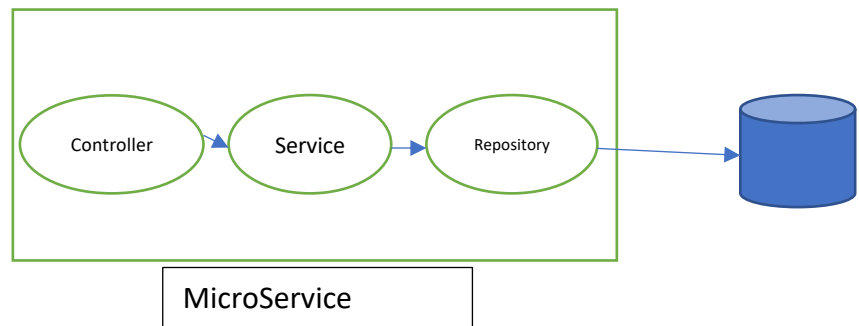- Modern software development technique in which an application is broken down into parts and each part is developed independently

Monolithic   Microservices

Monolithic Architecture   Microservice Architecture

REST API — API GATEWAY — REST API — Account Service — Account DB

Mobile app

REST API — Inventory Service — Inventory DB

Web — Storefront WebApp

Browser

REST API — Shipping Service — Shipping DB

User Interface (Controller)

Servlet/JSP

Service (Business Logic)

Dao (Persistence Logic)

DB

Tomcat Server
Deploy: WAR file

Customer/User

Retailer/Seller

Admin

Customer

Admin

Retailer

Shopping Application

Server

MakeMyTrip diagram with connections to:
- API — Air India
- JSON — API — SpiceJet
- API — Vistara

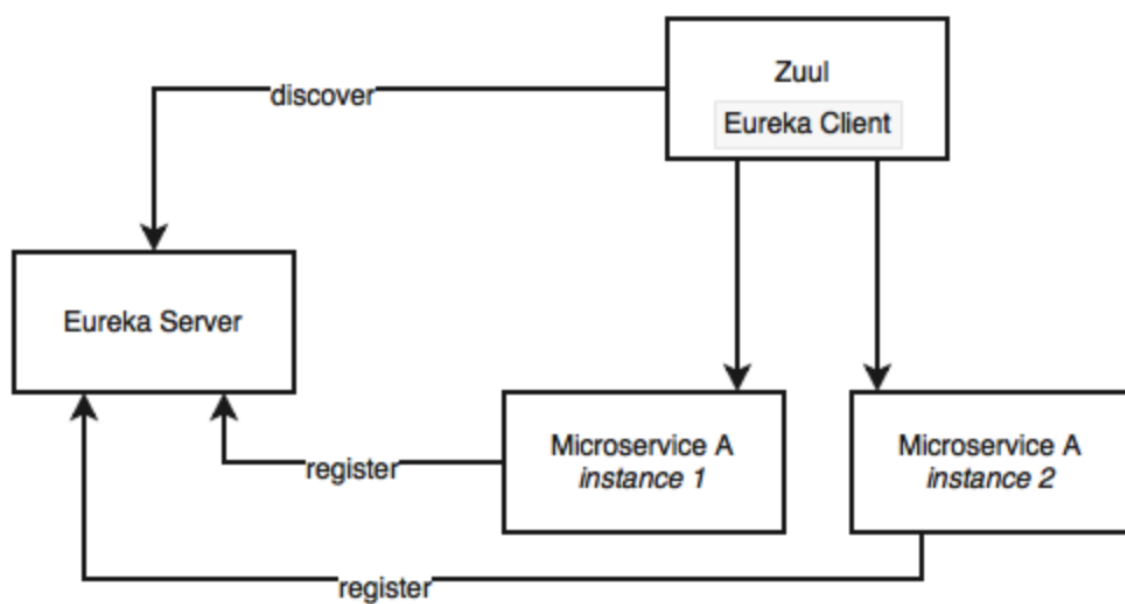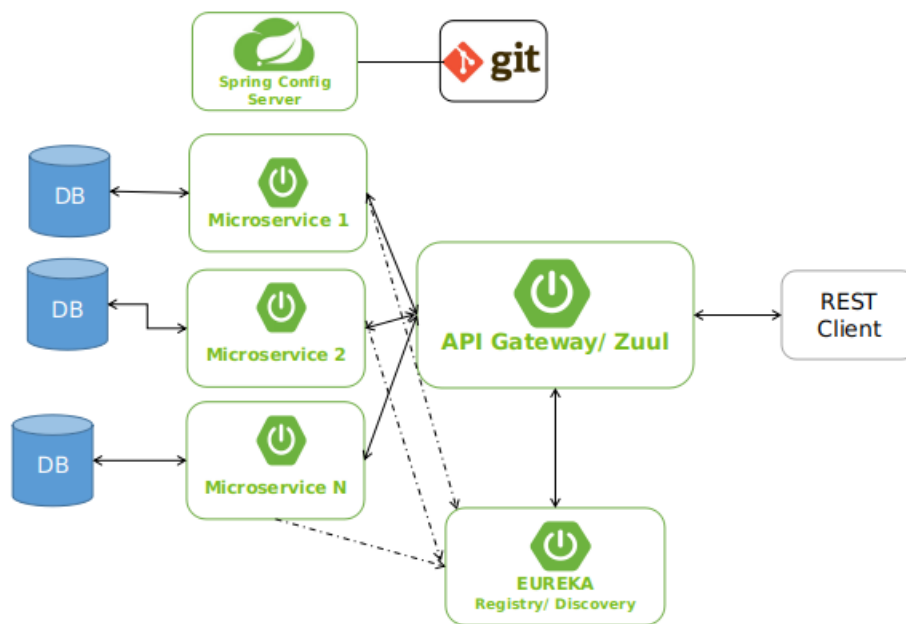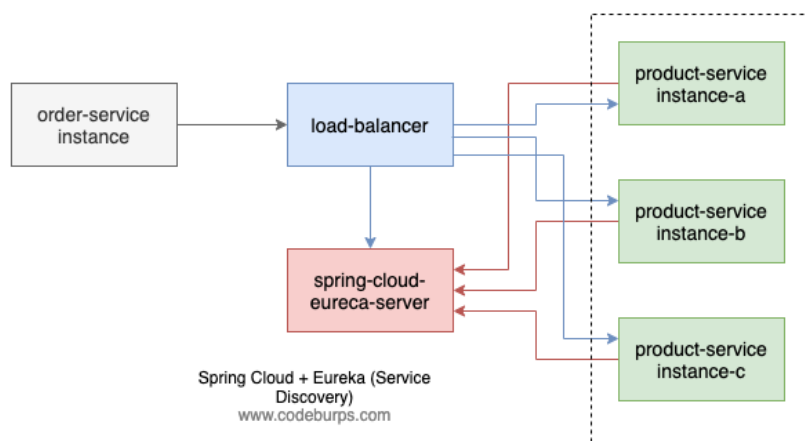Controller → Service → Repository → (database)

MicroService

TODO:
- When working on Microservices we come across services like:
  - Service Registry & Discovery (ex: Eureka)
  - API Gateway (ex: Zuul)
  - Circuit Breaker (ex: Hystrix)

Client

Instance 1   Instance 2

Instance 3

Microservice A

Service Registry & Discovery
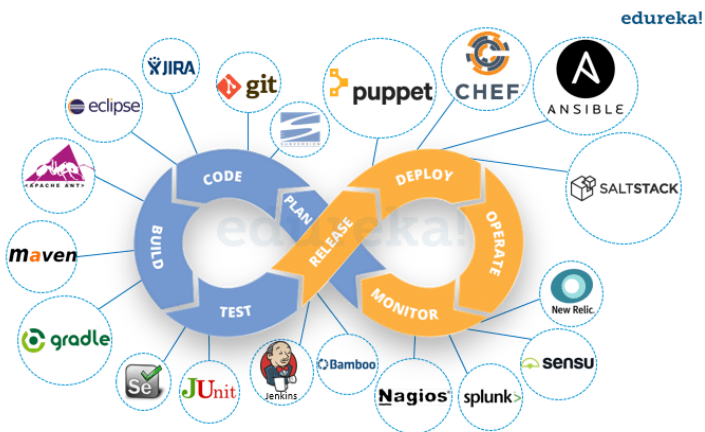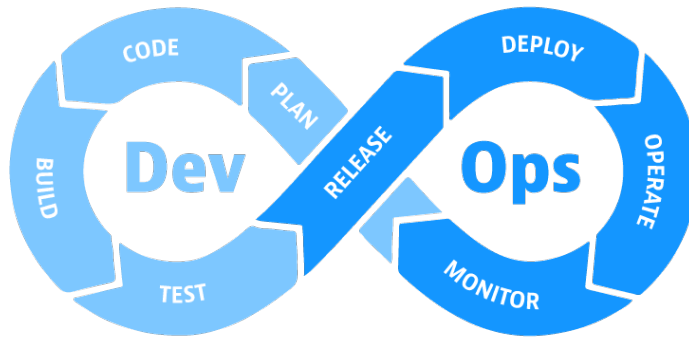
How can the client know which instance to connect to?
- We need someone(Service Registry & Discovery Service) who will keep track of current available microservices and their IP address

order-service instance → load-balancer → product-service instance-a / product-service instance-b / product-service instance-c

load-balancer → spring-cloud-eureca-server

Spring Cloud + Eureka (Service Discovery)
www.codeburps.com

Spring Config Server — git

DB → Microservice 1
DB → Microservice 2
DB → Microservice N

API Gateway/ Zuul → REST Client

EUREKA Registry/ Discovery

Zuul
Eureka Client

discover → Eureka Server

Microservice A instance 1 — register → Eureka Server
Microservice A instance 2 — register → Eureka Server
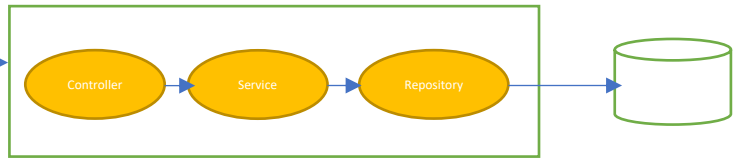
Serverless architecture

Cloud computing

DevOps

Flight Management Microservice

API of our Flight Management Microservice

GET : /flights/{source}/{destination}/{date}

GET : /flights/seats-available?flightNo={flightNo}&travelDate={travelDate}&noOfSeats={noOfSeats}

POST : /flights/block-seats

Flight Booking Microservice
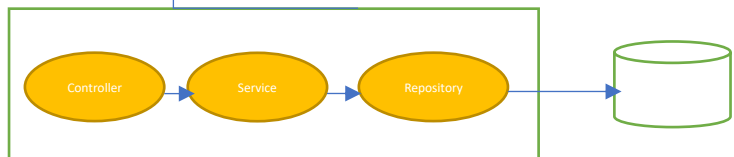
API of our Booking Microservice
POST : /book-flight

Booking
Service

Flight
Service

Instance1

Flight
Service

Instance2