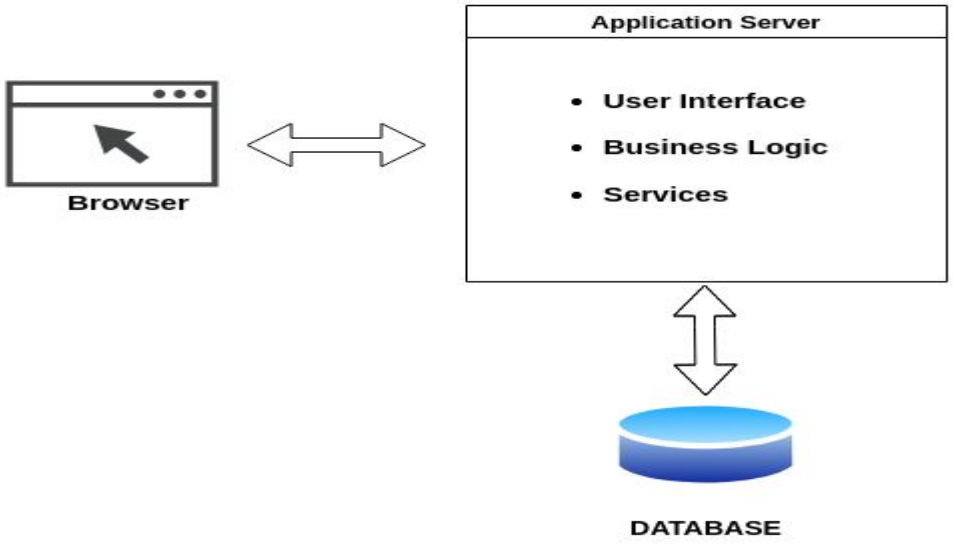
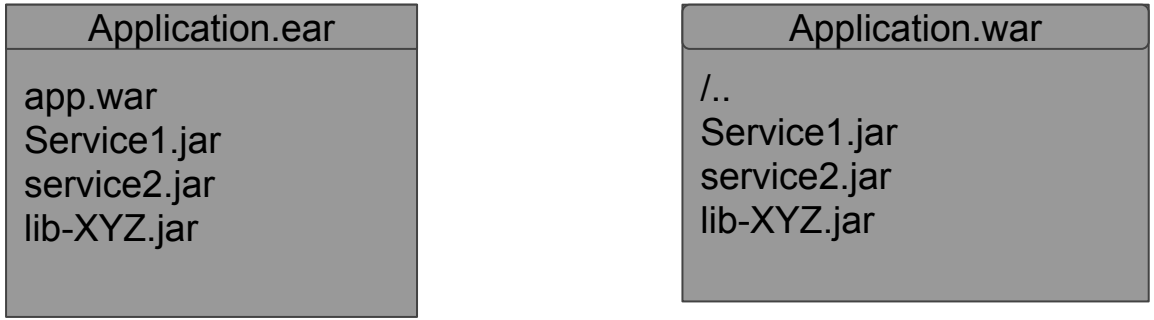
Introduction to microservices, Containers and Kubernetes

Ahmed Zaoui
26 June 2019

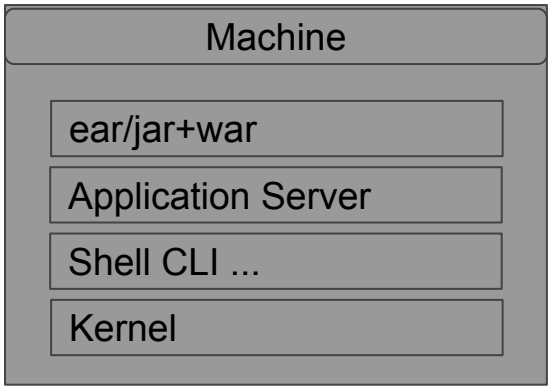
Monolithic Application



Packaging


Application.ear
app.war
Service1.jar
service2.jar
lib-XYZ.jar
Application.war
/..
Service1.jar
service2.jar
lib-XYZ.jar

Deployment



Drawback

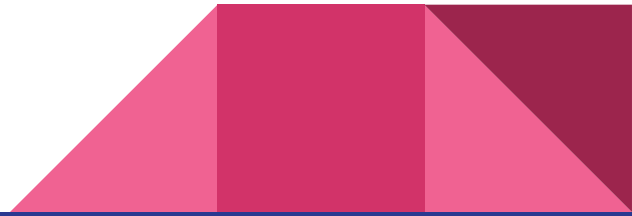
- **Deployed to a single server.**
- **Build on a single stack (such as .net or java).**
- **Frequent downtime.**
- **Hard to work with multiple team.**
- **Redeploy the entire application each new update.**
- **Run multiple instances of the application on multiple machines in order to satisfy scalability and availability. requirements.**
- **Reliability(Bug in any module e.g. memory leak can potentially bring down the entire process.)**
- **...**

Microservices

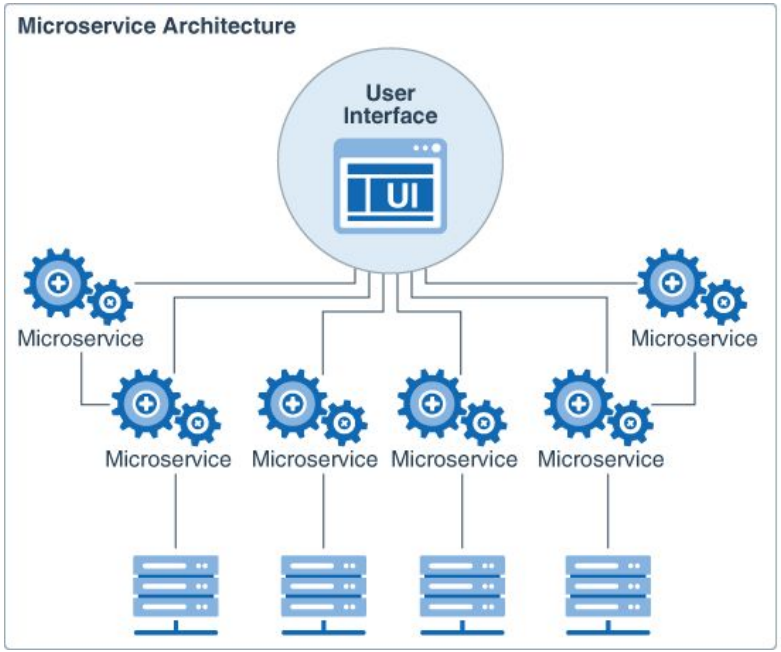
What are microservices ?

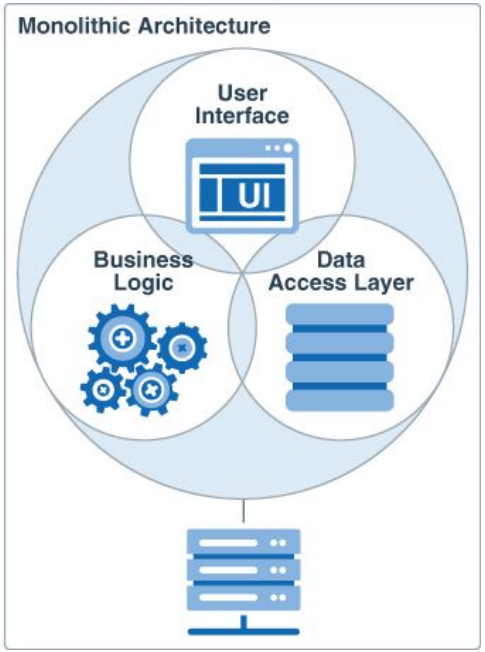
Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are:

- Highly maintainable and testable.
- Loosely coupled.
- Independently deployable.
- Organized around business capabilities.

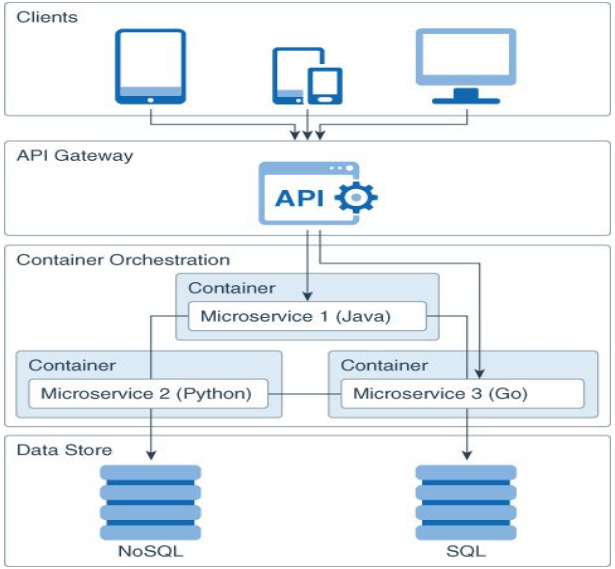


Sample Architecture





And more ...


Clients
API Gateway
API
Container Orchestration
Container
Microservice 1 (Java)
Container
Microservice 2 (Python)
Container
Microservice 3 (Go)
Data Store
NoSQL
SQL

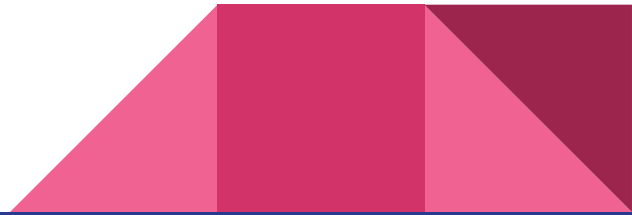
API Gateway

Problem: How do the clients of a Microservices-based application access the individual services ?

Solution : single entry point for all clients.

An API gateway that is the single entry point for all clients. The API gateway handles requests in one of two ways. Some requests are simply proxied/routed to the appropriate service. It handles other requests by fanning out to multiple services.

Example :
https://github.com/Netflix/zuul



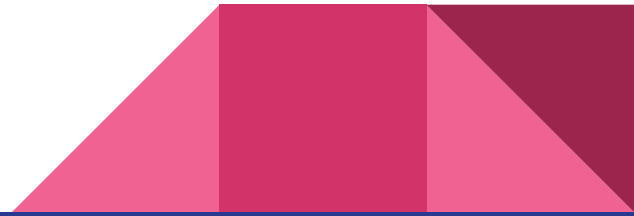
Framework to easy implement Microservices:

- **Microprofile : https://microprofile.io/**
- **SpringBoot : https://spring.io/projects/spring-boot**
- **Thorntail : https://thorntail.io/**

Demo :
https://github.com/azaoui/presentation

How to run microservices application ?

The best choice for running a microservices application architecture is application containers.



What are Containers

1. **A lightweight approach to virtualisation.**
2. **An executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.**
3. **An abstraction approach from the environment in which they actually run.**

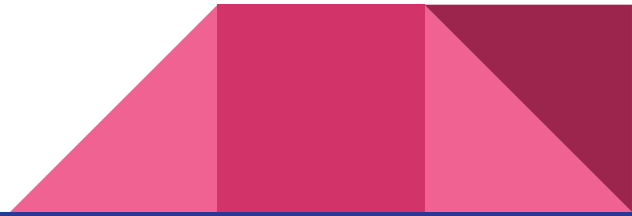
Demo :
https://github.com/azaoui/presentation

Containers orchestration

1. **Availability of containers.**
2. **Scaling up or removing containers.**
3. **Load balancing and service discovery between containers.**
4. **Monitoring.**
5. **Configuration.**

Kubernetes

1. **An open-source container-orchestration.**
2. **Originally designed by Google**



Kubernetes basic Objects:

Node :

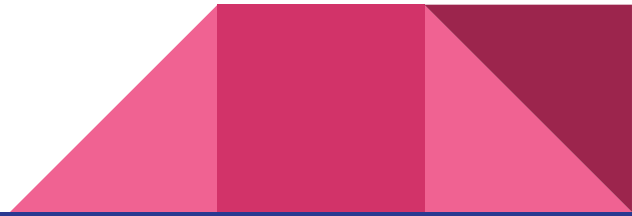
1. **A worker machine in Kubernetes**
2. **A node may be a VM or physical machine**

A Pod :

1. **The smallest deployable unit on a node**
2. **A group of one or more containers which must run together.**
3. **A Pod usually contains one container.**
4. **It represent a microservice running on Kubernetes.**

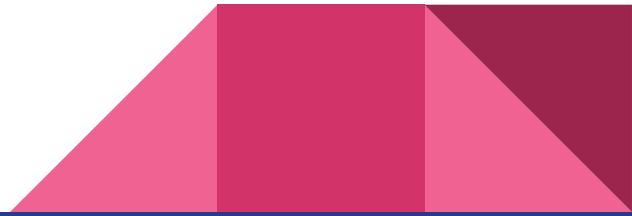
Deployment :

1. **Replication Controllers**
2. **A supervisor for pods**



Service :

An abstract way to expose an application running on a set of Pods as a network service.



Demo

Before you begin:

1. **Install kubectl**
2. **Install minikupe(the quickest way to go. It makes it easy to run a single-node Kubernetes locally)**

https://kubernetes.io/docs/tasks/tools/install-minikube/#before-you-begin

Thank you!