dead

Lecture 6

Q) what are Microservices
Q) when & why it is used
Q) Monolithic vs microservices

→ Monolithic architecture → imp concept & backend.
so microservices came in picture in order to overcome the disadvantages of monolithic architecture.

→ If we develop any architecture it should be built in single unit

Eg Netflix / airbnb app → authentication | payment | listing
all its features / operations should be combined in single code

→ so the entire app should be built as sing unit

→ also the entire app should be deployed as single unit. So it will have single CICD pipeline

→ all code should be written within one lang ie if we choose java, then the entire app & its functions / features should be built with java only

→ in small startup or projects it it used as

adv of monolithic

→ easy to develop → easy to manage

Disadv

→ Redeployment eg in payment we made a small change we need to redeploy the entire application. as we cannot redeploy only one feature cause it it based on single unit.

→ Scaling if many users tries to login ie we need to scale up authentication, it is not possible to do only 1 feature So the entire app needs to be scale up

Eg auth (java 2.0) payment (java 2.0) } so the other code also need to be written in java 3.0 and need to redeploy.

need to update to java 3.0

and it becomes more difficult if we need to scale down the appln in future.

so not much flexibility

→ loats of dependency → if many developers are working on single app. if one developer wants to update any part of code, he needs to think twice, cause his change change might impact other function

Eg auth payment

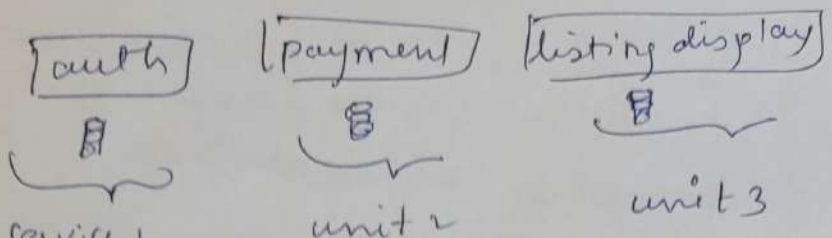
2.0 version

↳ if package version should be same

→ Eg In 2009 it followed Monolithic arch but because of drawbacks it started Microservices as solution.

Microservices

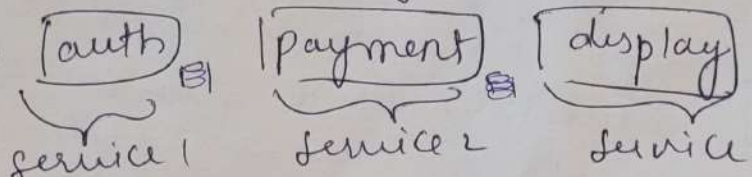
→ We divide same application into different services each service would be an individual unit.

→ 

→ Each service can be built independently
→ each service would have its own codebase and database ; each have seperate repository
& each service is loosely coupled ie not much dependent on other services.

Que when our appl^n moves from monolothic to microservices, how many services should be made

Ans it depends on features & how big the appl^n is
it depends on companys business requirement.

Eg Airbnb → 

we can have seperate services for above.

→ This is mainly used in big companies, cause there it doesnot makes sense for many developers to work on single unit and codebase

→ Each team has individual

team →
codebase →
database →
CI CD pipeline →
independent deployment

Imp Adv

Independent deployment

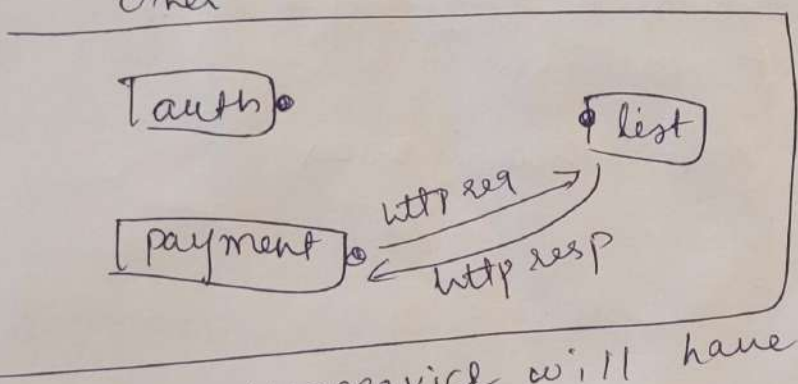
→ suppose the appln is already deployed, and auth service needs to 3.0 version (2.0 version)
so its service could be deployed with 3.0 version and other services would still run with 2.0 version.

→ flexible scaling
we can scale up and scale down individual service irrespective of what other services are doing.

→ Technology flexibility
as each service has its own fun & own database so each service can be written in its own lang

auth — java
payment — python

Imp Qu How microservices communicate with each other



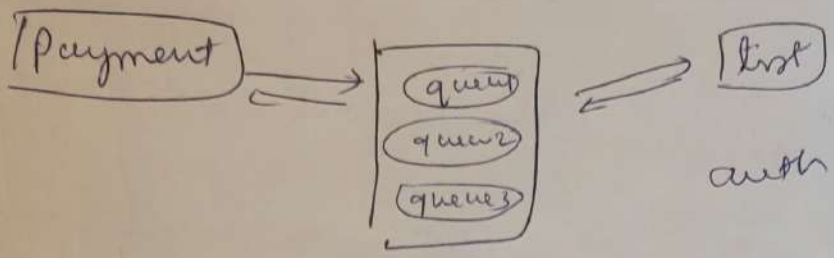
met1 ~~call~~ Synchronous communication using API call

each microservice will have its individual end point

→ Eg payment servia wants to interact with list servic so payment servia will send a http request to list servia and list servic will synchronously sends a http response to payment servic

met2 Asynchronous commun using msg brokers

msg brokes Eg rabbit Mq, apache kafka



Eg payment serv wants to interact with list service so payment service will sent the commun to msg broker & then msg broker will sent the commun to list serv

Commun 3 → Using Service Mesh Eg Istio
when our microservices are deployed on kubernetes we use Istio

---

Microservices is not perfect, its disadv

→ complex to develop
→ Management overhead is more as huge team & many services
→ as each service has its own database & its own CICD pipeline so cost is high.

So small startups use monolithic Service