118. Multi-Container pods

⇒ The idea of decoupling a large monolithic application into sub components known as microservices enables us to develop and deploy a set of independent small & reusable code

⇒ Micro Services architecture can then help us scale up, down, as well as modify each services as required as opposed to modifying entire application.

⇒ However at times, we need 2 services to work together such as webserver & logging service.

* we need one agent instance per webserver instance paired together, we dont want to merge & load the code of tool services, as each of them target different functionalities & you would still like them to be developed & deployed seperately.

* we only need 2 functionality to work together, we need one agent per web server instance paired together that can scale up & down together & that is why you have multi container pods that sharing the same life cycle which means they are created together & destroyed together, they share the same n/w space, which means they can refer to each other as local host & they have access to the same storage volumes. This way we do not have to establish volume sharing or services b/w pods to enable communication b/w them

To Create a multi-container pod

Add the new containers information to pod-definition file.

Remember,

⇒ The containers section under the spec section in a pod definition file is an array, & the reason it is an array, is to allow multiple containers in a single pod

⇒ In this case, we add a new container named log agent to our existing pod.

pod-definition.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: simple-webapp
  labels:
    name: simple-webapp
spec:
  containers:
  - name: simple-webapp
    image: simple-webapp
    ports:
      - containerPort: 8080
  - name: log-agent
    image: log-agent
```