123 Init Containers

In a multi-container pod, each container is expected to run a process that stays alive as long as the POD's lifecycle.

For eg:-

In the multi-container pod that we talked about earlier that has a web application & logging agent, both the containers are expected to stay alive at all times. The process running in the log agent container is expected to stay alive as long as the web application is running. If any of them fails, the POD restarts.

But at times you may want to run a process that runs to completion in a container. For eg:- a process that pulls a code or binary from a repository that will be used by the main web application. That is a task that will be run only one time when the pod is first created or a process that waits for an external service or database to be up before the actual application starts, Thats where init Containers comein.

An init Containers is configured in pod like all other container, except that it is specified inside a init Containers section like this

- - - - - - - - - - - - - - - - -

⇒ Init Containers runs first, finish their job & only then do the main Container starts

⇒ A special Container that runs before the main app starts

⇒ It prepares things like downloading files, setting up permission

⇒ If it fails, the main app wont even start

```
apiVersion : V1
kind: Pod
metadata:
   name: myapp-pod
   labels:
     app: myapp
  spec:
    containers:
    - name: myapp-container
      image: busybox : 1.28
      command: ['sh', '-c', 'echo the app is running! &&
                                    sleep 3600']

     initContainers:
      - name: init-myservice
        image: busybox
        command: ['sh', '-c', 'git clone <some-repository-
                 that-will-be-used-by-application>; done; ']
```

when a POD is first created the init Container is run, and the process in the init Container must run to a completion before the real container hosting the application starts

you can configure multiple such init Containers as well, like how we did for multi-container pod. In that case each init Container is run one at a time in sequential order.

If any of the init containers fail to complete, K8S restarts the pod repeatedly until the init Container succeeds

```
apiVersion: v1
kind: Pod
metadata:
    name: myapp-pod
    labels:
        app: myapp
spec:
    containers:
    - name: myapp-container
      image: busybox:1.28
      command: ['sh', '-c', 'echo The app is running! && 
                                   sleep 3600']

    initContainers:
      - name: init-myservice
        image: busybox:1.28
        command: ['sh', '-c', 'until nslookup myservice; do
              echo waiting for myservice; sleep 2; done;']

      - name: init-mydb
        image: busybox:1.28
        command: ['sh', '-c'; until nslookup mydb;
              do echo waiting for mydb; sleep 2; done;]
```

do in practice task - Init containers

Diff b/w Init & Side Car container.

Init container	Side Car Container.
Runs before main containers	Runs alongside main containers
Must finish first	keeps running as long as main container runs