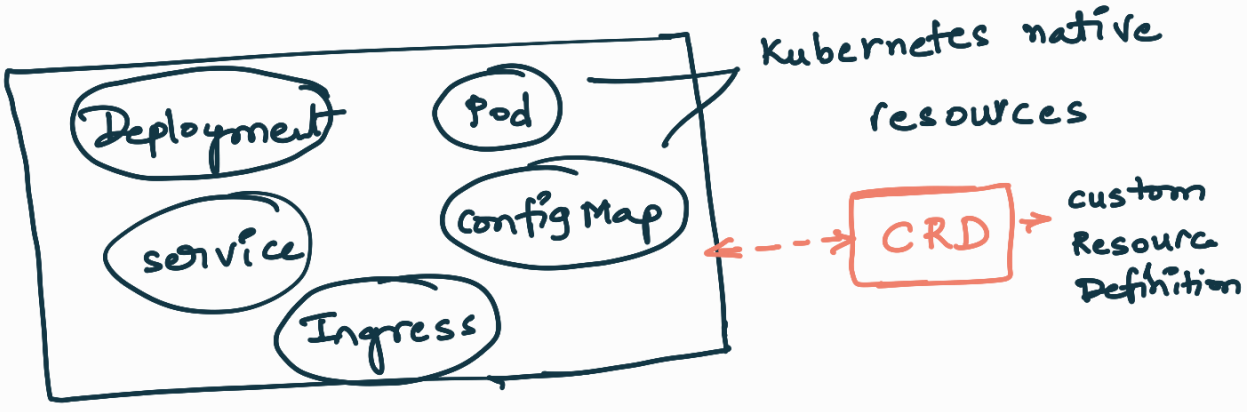
#KubernetesIn30Days challenge :-

#Day13 :-

Custom Resource Definition (CRD)



To Extend the functionality of kubernetes with API's. Kubernetes allows you to use them. it is called custom Resource Definition.

For Example, If we want to extend security features of Kubernetes you can implement any custom API to extend the functionality of kubernetes.

There are two entities/people who should act to these CRDs. they are :

1. DevOps Engineer
2. User

Ex :- Flux, Argo CD, Key clock etc.

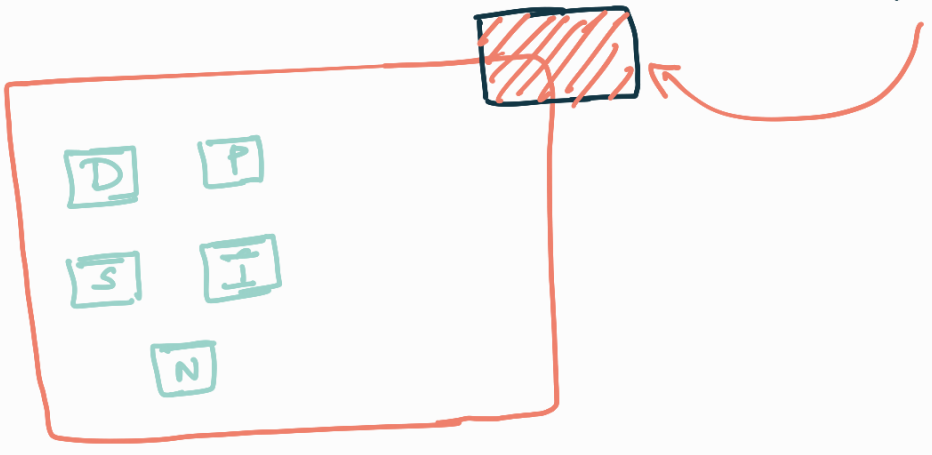
Our Main focus will be on

1. Custom Resource Definition - CRD
2. Custom Resource - CR
3. custom Controller

custom Resources of K8 APIs >>> Native K8 APIs

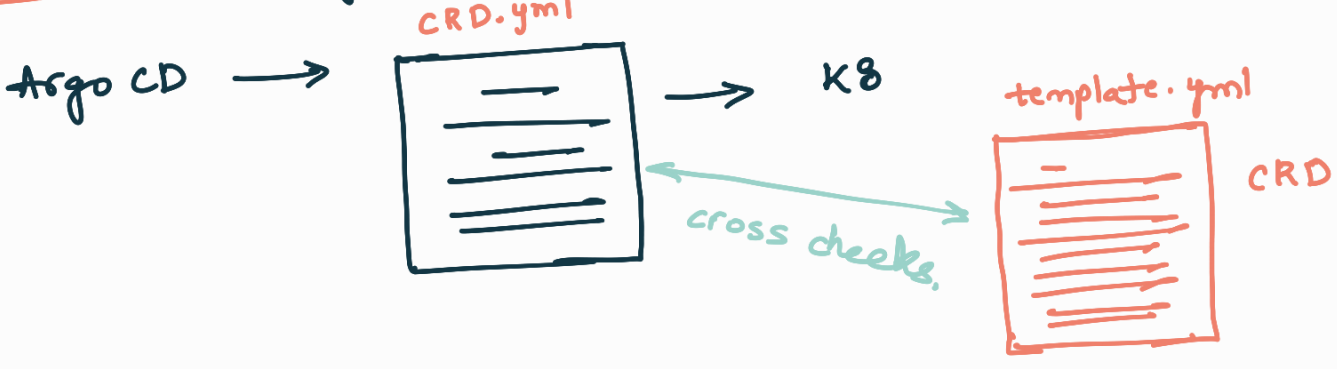
- istio
- argoCD
- keycloak
- security related
- ⋮

→ K8 cannot add logic to all custom apis. so K8 Extend capabilities of API using CRD, CR, CC



K8s cluster

① CRD :- Defining new type of API to K8s.



Example :-

YAML

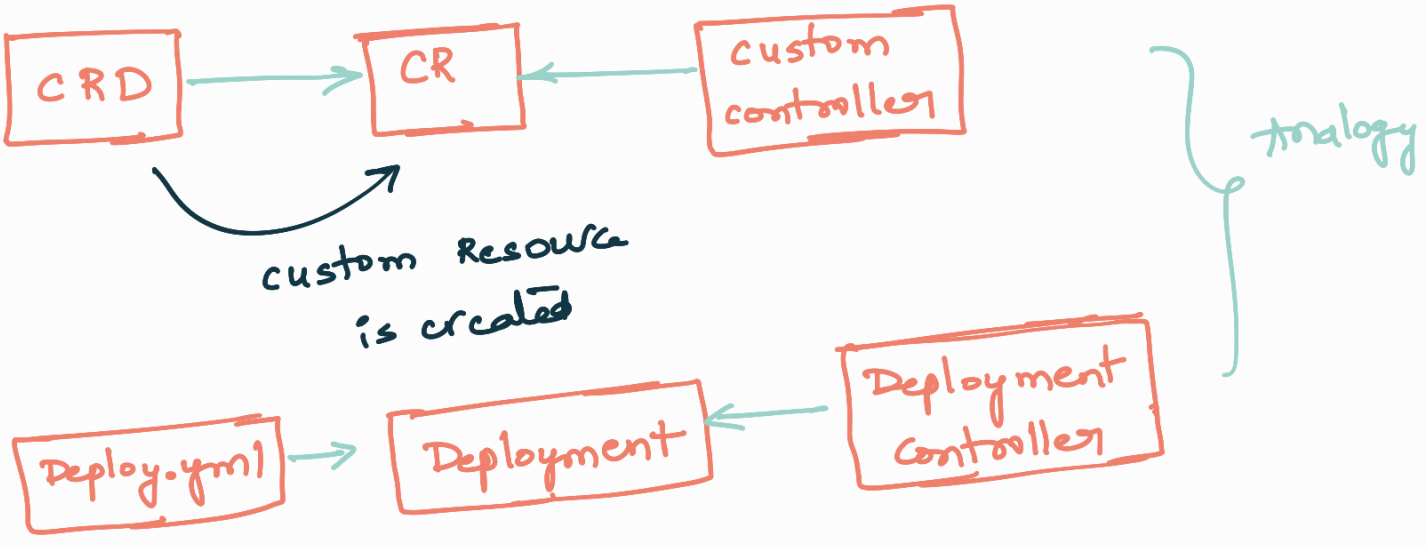
Deployment.yml

Resource Definition template

CRD

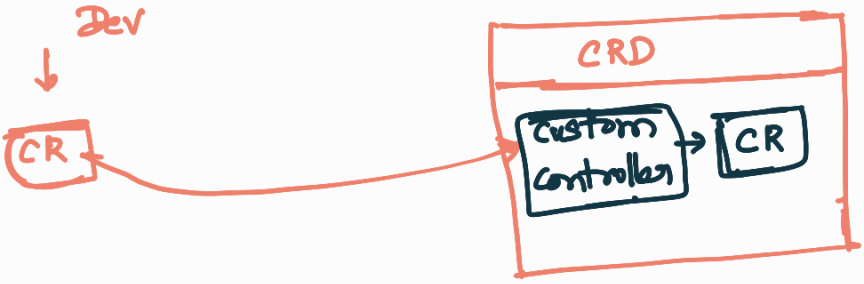
K8 will check whether Deploy.yml is upto standard.

Here CRD will be std template it'll validated against CRD template.

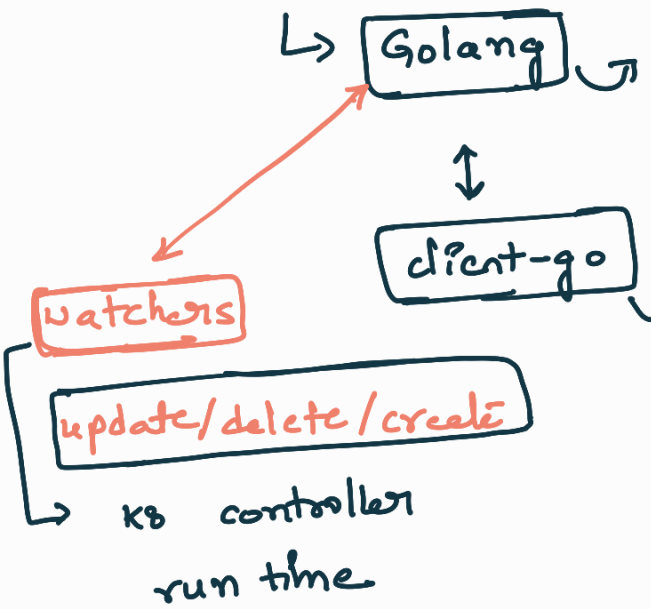


This CR will be created inside cluster.

It'll have the custom controller.



→ custom controller



Most popular language

(K8s itself it built by Golang)

(Now it supports most of languages)

Use Respective installation documentation to install the custom controller. It'll vary with the tool your using.

Deploying of CR, CRD and custom controller is a responsibility of DevOps Engineer and. Managing of it is also a part of it.