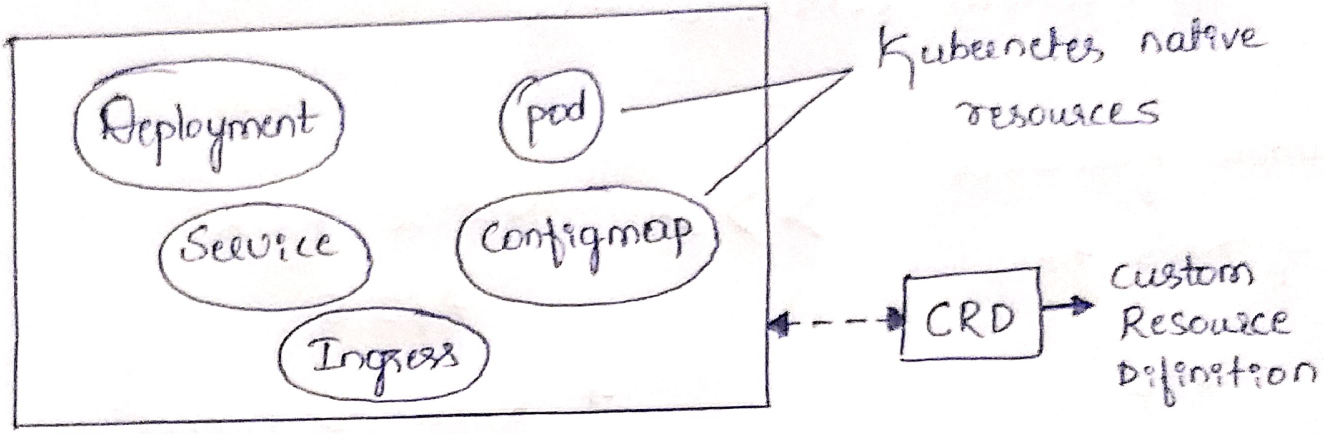
Day - 13

Custom Resource Definition (CRD)



To extend the functionality of kubernetes with API's kubernetes allows you to use then, it is called custom resource definition.

For example, If we want to extend Security features of kubernetes you can implement any custom API to extends the functionality of kubernetes.

There are two entities/people who should act to these CRD's they are:

1) DevOps Engineer

2) User

Examples: Flux, Argo CD, key clock etc.

→ Our main focus will be on:-

1) Custom resource definition — CRD

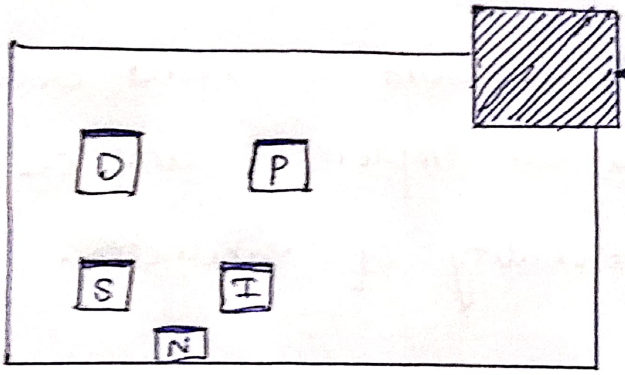
2) Custom resource — CR

3) Custom Controller — CC

Custom Resources of K8's APIs >>> Native K8's API's

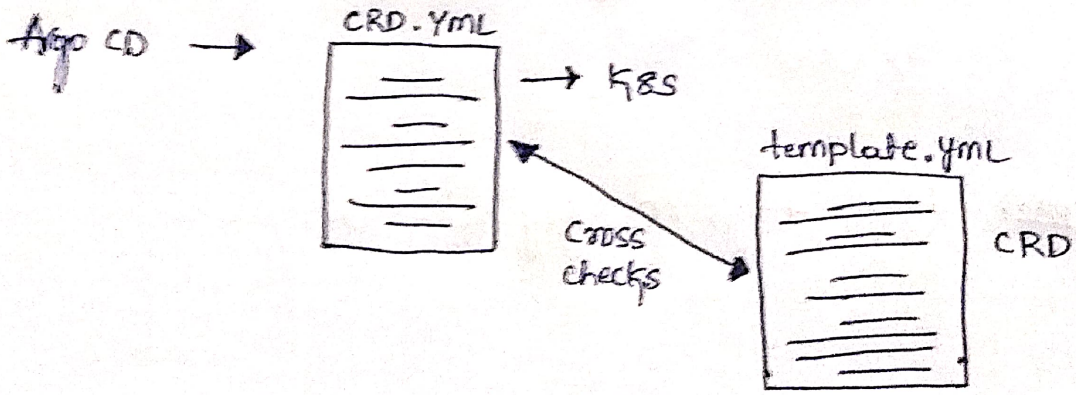
- istio
- ArgoCD
- Keyclock
- Security related

→ Kubernetes cannot add logic to all custom API's. So K8s extend capacilities of API using CRD, CR, CC

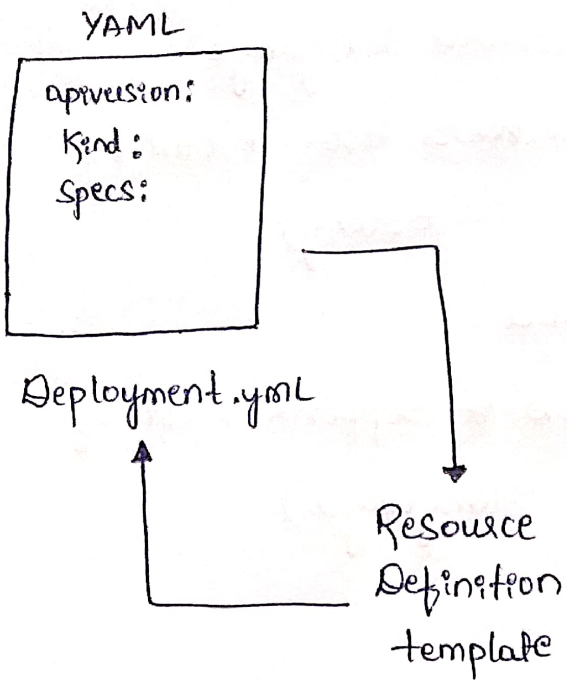


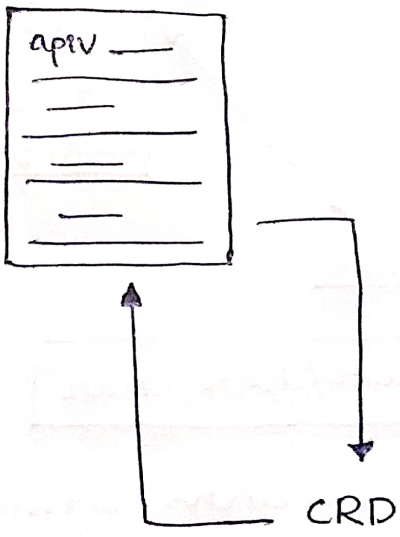
Kubernetes cluster

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



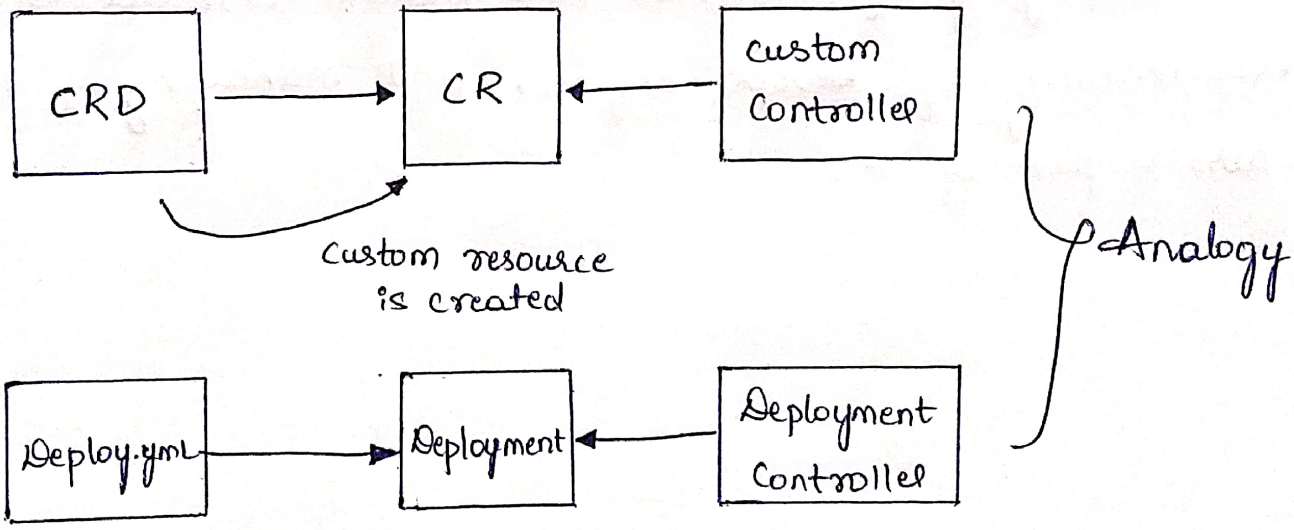
Example :-



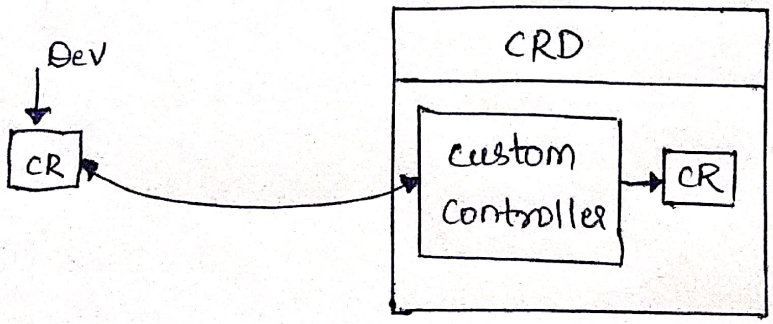


Kubernetes will check whether deploy.yml is upto Standard.

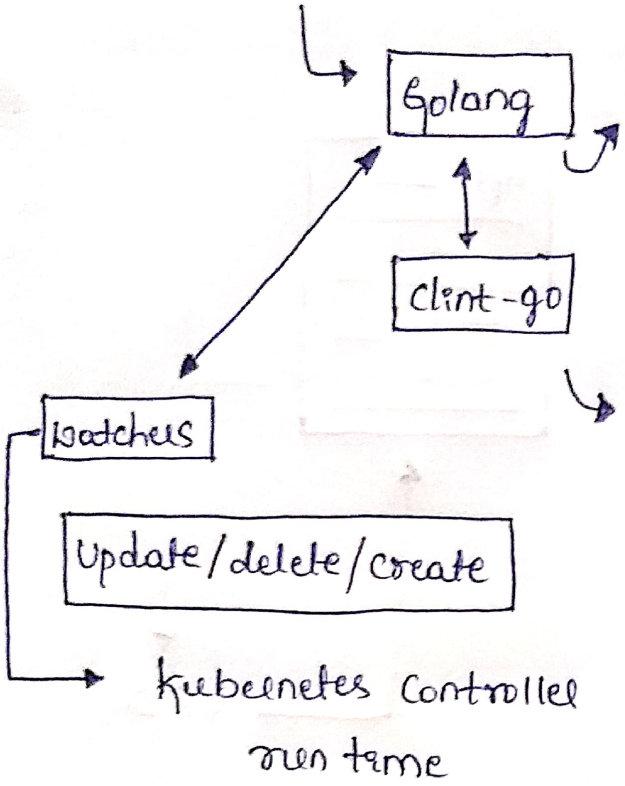
Here CRD will be Standard template it will validated CRD template.



This custom resource will be created inside cluster. It will have the custom controller.



→ Custom Controller.



Use respective installation documentation to install the Custom Controller. It will 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.