Day-5

- Namespaces
- Ingress
- Jobs & Cronjobs
- Daemonsets
- Statefulsets.
- ~/.kube/config-file.

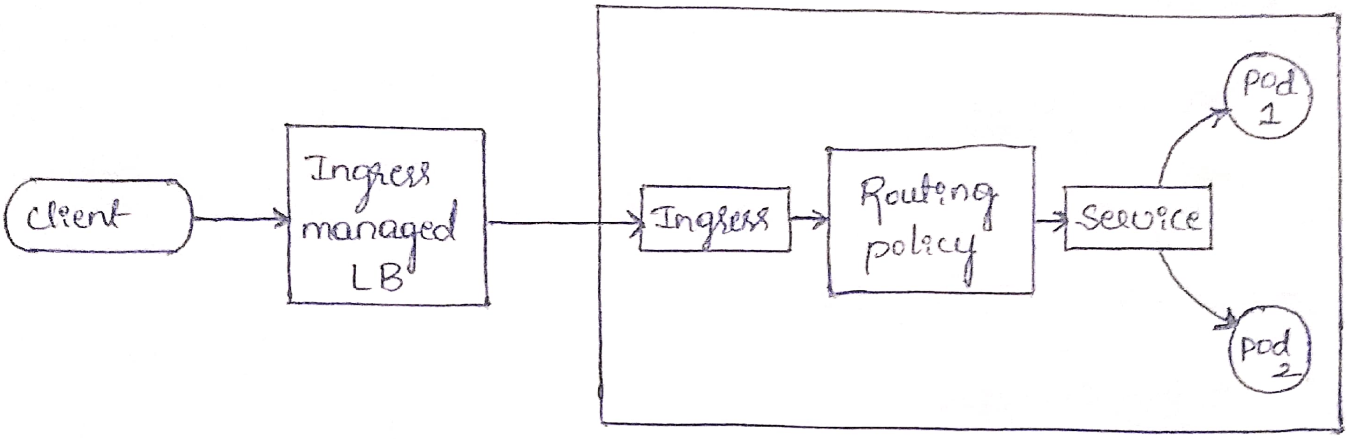
1) Namespaces:-

- Namespaces provides mechanism for isolating group of resources within a cluster.
- Name of resources in NS should be unique.
- Deployment services etc, are namespaced scoping objects.

i) default
ii) kube-node-lease
iii) kube-public
iv) kube-system.

2) Ingress:-

- An API object that manages external access to the services in a cluster typically HTTP.
- Ingress may provide load balancing, SSL termination and name based virtual hosting.



3) Jobs and Cronjobs:-

- Job creates one/more pods and will continue to retry execution of the pods until a specified number then successfully terminate.

- A cronjob creates jobs on a repeating schedule. Just like crontab in UNIX/LINUX.

4) Daemonset :-

- A daemonset ensure that all/some nodes runs a copy of pod. As nodes added to cluster. pods are added to them. As nodes are removed from cluster those pods are garbage collected. Deleting daemonset will cleanup the pods it's created.

- uses of daemonset are running.

1) cluster storage daemon on every node.

2) logs collection daemon on every node.

3) Node monitoring daemon on every node.

5). Statefull sets :-

- Statefulsets is the workload API used to manage Stateful applications.

- manages the deployment and scaling set of pods and provides guarantees about orders and uniqueness of these pods.

6) ~/.kube/config files :-

- Kubernetes configuration file called as kubeconfig.

- It is a YAML file that specifies various settings and parameters for interacting with Kubernetes

- This file is used by kubectl command to authenticate with cluster, specifies cluster and configure other options.

- Key elements in kubeconfig file are.
    i> cluster Configuration.
    ii> user Configuration.
    iii> Context Configuration.
    iv> current context.

Checkout Example "kubeconfig" below.

```
apiVersion: V1
kind: config
clusters:
  -name: my-cluster.
   cluster:
     server: https://cluster-api-server.
     certificate-authority-data: <certificate-authority-data>
users:
  -name: my-user
     client-certificate-data: <client-certificate-data>
     client-key-data: <client-key-data>
contexts:
  - name: my-context
    context:
      cluster: my-cluster.
       user: my-user
current-context: my-context.
```