Namespace provides an additional qualification to a resource name. This is helpful when multiple teams are using the same cluster and there is a potential of name collision. It can be as a virtual wall between multiple clusters.

Functionality of Namespace

Following are some of the important functionalities of a Namespace in Kubernetes −

* Namespaces help pod-to-pod communication using the same namespace.
* Namespaces are virtual clusters that can sit on top of the same physical cluster.
* They provide logical separation between the teams and their environments.

Create a Namespace

The following command is used to create a namespace.

apiVersion: v1

kind: Namespce

metadata

name: elk

Control the Namespace

The following command is used to control the namespace.

$ kubectl create –f namespace.yml ---------> 1

$ kubectl get namespace -----------------> 2

$ kubectl get namespace <Namespace name> ------->3

$ kubectl describe namespace <Namespace name> ---->4

$ kubectl delete namespace <Namespace name>

In the above code,

* We are using the command to create a namespace.
* This will list all the available namespace.
* This will get a particular namespace whose name is specified in the command.
* This will describe the complete details about the service.
* This will delete a particular namespace present in the cluster.

Using Namespace in Service - Example

Following is an example of a sample file for using namespace in service.

apiVersion: v1

kind: Service

metadata:

name: elasticsearch

namespace: elk

labels:

component: elasticsearch

spec:

type: LoadBalancer

selector:

component: elasticsearch

ports:

- name: http

port: 9200

protocol: TCP

- name: transport

port: 9300

protocol: TCP

In the above code, we are using the same namespace under service metadata with the name of **elk**.