**Kubernetes – NameSpaces**

In Kubernetes, namespaces provide a mechanism for isolating groups of resources within a single cluster”. Within a Kubernetes Namespace, resources must have unique names, but across different Namespaces, you can have resources with the same name.

By default Kubernetes gives you **4 Namespaces.**

1. **default**
2. **kube-node-lease**
3. **kube-public**
4. **kube-system**

**1. kube-system**

kube-system is the namespace that includes objects created by the Kubernetes system. The components that are deployed in this Namespace are the system processes – they are from Master managing processes or Kubectl etc. kube-system Namespace is not meant for our (developer’s) use. so we do not have to create anything or modify anything in this namespace.

**2. kube-public**

kube-public contains the publicly accessible data. It has a config map that contains the Cluster information which is accessible even without authentication.

**3. kube-node-lease**

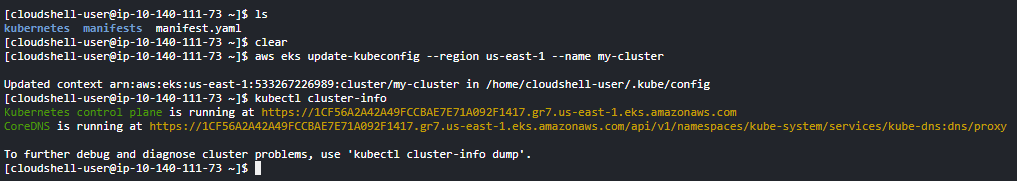
kube-node-lease Namespace is a new addition to Kubernetes. The purpose of this namespace is that it holds information about the heartbeats of Nodes. So each Node basically gets its own lease object in the Namespace. This object contains the information about that nodes availability.

**4. default**

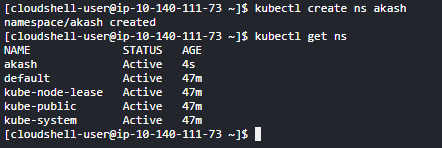
default Namespace is the Namespace that we use in order to create the resources when we create a Namespace in the beginning.

**Why there is a need to create NameSpace.**

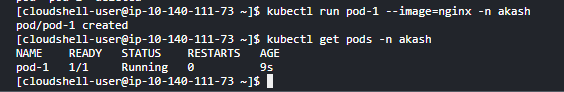
* Structuring Resources in Groups
* Preventing Conflicts between Multiple Teams
* Resource Sharing
* Limiting Access and Resources
* To use NameSpace first we have to create it manually or we can also create the NameSpace from the manifest file.
* Steps
* Create a cluster and Node Group.
* Configure the cluster on the cloudshell.



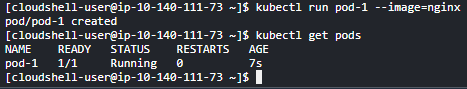
* Hit command “kubectl get ns” to view all the NameSpaces available.
* When we create a pod it is created in a default namespace.
* Now we will create a NameSpace.
* Hit command “kubectl create ns akash” this will create a namespace named akash.



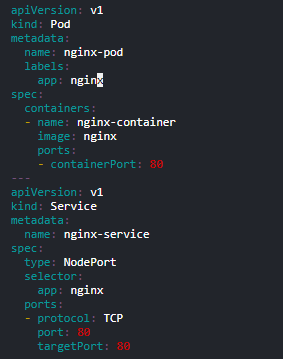
* Namespace created.
* Now we will create a pod in this namespace.
* Hit command “kubectl run pod-1 --image=nginx -n akash” this will create a pod named pod-1 in akash namespace.
* Hit command “kubectl get pods -n akash” to view the pods in akash namespace.



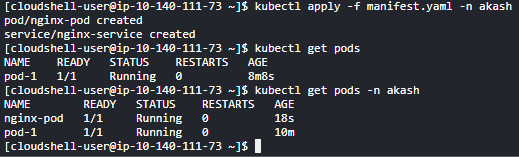
* If we create a pod without specifying the namespace it will be created in default namespace.
* Hit command “kubectl run pod-1 --image=nginx”.
* Now hit command “kubectl get pods” to view the pods.



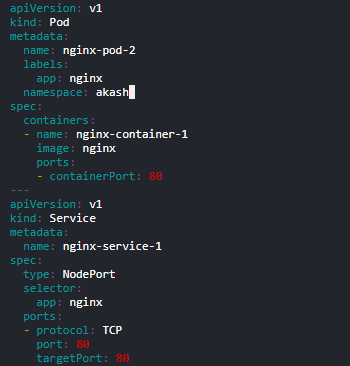
* We don’t need to specify name everytime we create a pod we can specify it while applying the manifest file also.
* Write a manifest file for the creating a pod and exposing a Port.



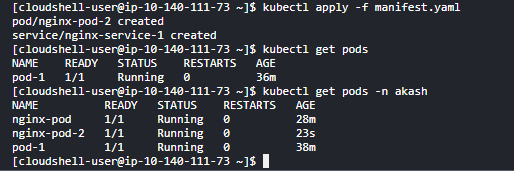
* Apply the manifest.
* Hit command “kubectl apply -f manifest.yaml -n akash” this will create a pod in akash namespace.



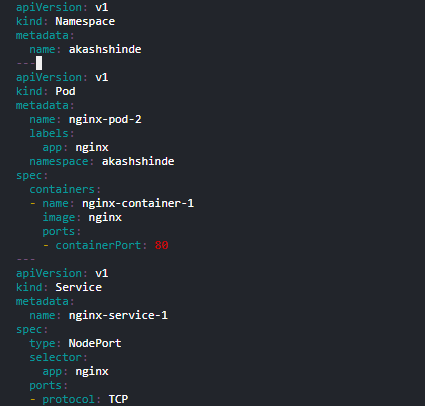
* Also we don’t have to give the namespace everytime we apply the manifest we can add the namespace in the manifest file.

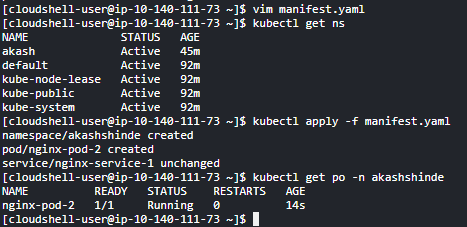


* Now apply the service and see where the pod is created.



* In the above cases we have to first create a namespace manually before adding in the manifest file.
* But we can also write a configuration to to create a namespace.

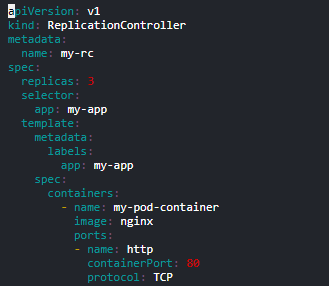




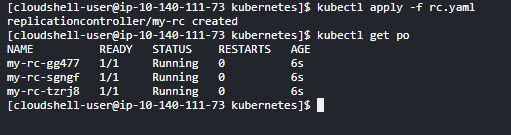
* We can delete these pods by using command “kubectl delete pod pod-name”.

**Replication Controller**

* **Replication Controller manifest.**

****

* Apply the manifest and check the pods.
* Hit command “kubectl apply -f rc.yaml”.

****