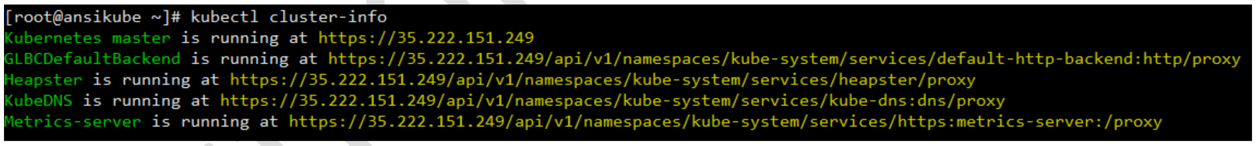
PODS

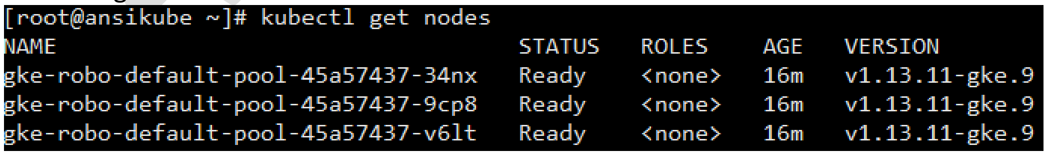
About Pod:-
Basic unit of Kubernetes is pod, the pod is thin wrapper and it can have a single container or multiple container. pod will have single network ip and under this pod, multiple container will use same ip and to communicate each other it will use loopback. means pod is single instance, under this instance multiple process can be run is called container.

Actual Readme:-
Understanding Pods
1) Pods are the smallest deployable units of computing that can be created and managed in Kubernetes.
2) A Pod is a group of one or more containers (such as Docker containers), with shared storage/network, and a specification for how to run the containers.
3) A Pod's contents are always co-located and co-scheduled and run in a shared context. A Pod models an application-specific "logical host" - it contains one or more application containers which are relatively tightly coupled — in a pre-container world, being executed on the same physical or virtual machine would mean being executed on the same logical host.

****
#kubectl api-resources
#kubectl api-versions
****
Pod Creation:-
 Two types - > Imperative method and Declarative method
Imperative method - single command line -eg = kubectl run --generator=run-pod/v1 web2 --image=nginx
Declarative Method - writing a manifest file using yaml or json

To Get cluster info:-
#kubectl cluster-info

```
[root@ansikube ~]# kubectl cluster-info
Kubernetes master is running at https://35.222.151.249
GLBCDefaultBackend is running at https://35.222.151.249/api/v1/namespaces/kube-system/services/default-http-backend:http/proxy
Heapster is running at https://35.222.151.249/api/v1/namespaces/kube-system/services/heapster/proxy
KubeDNS is running at https://35.222.151.249/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://35.222.151.249/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy
```

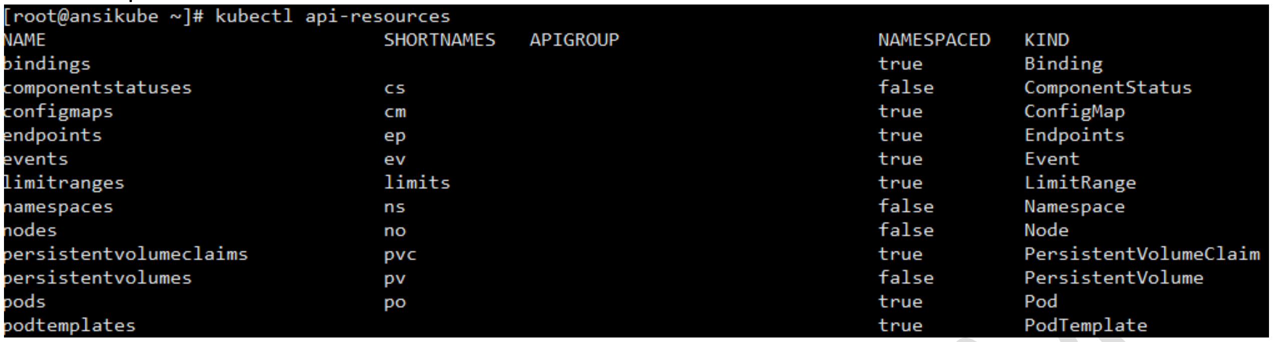
Note:- To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

To get cluster nodes:-
#kubectl get nodes

```
[root@ansikube ~]# kubectl get nodes
NAME                                  STATUS   ROLES    AGE   VERSION
gke-robo-default-pool-45a57437-34nx   Ready    <none>   16m   v1.13.11-gke.9
gke-robo-default-pool-45a57437-9cp8   Ready    <none>   16m   v1.13.11-gke.9
gke-robo-default-pool-45a57437-v6lt   Ready    <none>   16m   v1.13.11-gke.9
```

To view the kubernetes object:-

There are many objects available in kubernetes, below I just pasted few output only.

#kubectl api-resources



```
[root@ansikube ~]# kubectl api-resources
NAME                         SHORTNAMES   APIGROUP                     NAMESPACED   KIND
bindings                                                               true         Binding
componentstatuses            cs                                        false        ComponentStatus
configmaps                   cm                                        true         ConfigMap
endpoints                    ep                                        true         Endpoints
events                       ev                                        true         Event
limitranges                  limits                                    true         LimitRange
namespaces                   ns                                        false        Namespace
nodes                        no                                        false        Node
persistentvolumeclaims       pvc                                       true         PersistentVolumeClaim
persistentvolumes            pv                                        false        PersistentVolume
pods                         po                                        true         Pod
podtemplates                                                           true         PodTemplate
```

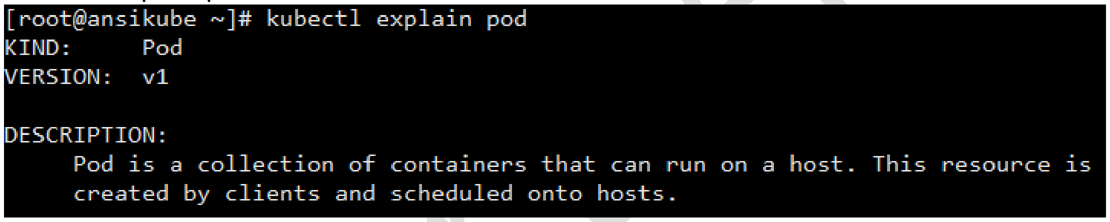
Api Version:-

(In kubernetes, all request been hit on API and will use to create resource or object, basically it will be used as extendibility and not required to touch the existing setup, we can create new API location without disturb existing. so for each resource and object have API, three type of API (Alpha,beta,stable). make sure to have knowledge to work find or create new resource or object with specific API version.

#kubectl api-versions

To find the API version of pod (Documentation of resources):-

#kubectl explain pod



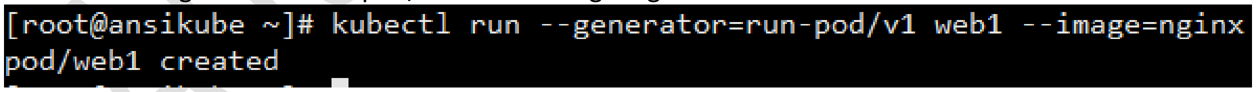
```
[root@ansikube ~]# kubectl explain pod
KIND:     Pod
VERSION:  v1

DESCRIPTION:
     Pod is a collection of containers that can run on a host. This resource is
     created by clients and scheduled onto hosts.
```

##Pod Creation##

Two types - > Imperative method and Declarative method.
Imperative method - single command line -eg = kubectl run --generator=run-pod/v1 web2 --image=nginx
Declarative Method - writing in a manifest file as yaml or json -eg = kubectl apply -f file.yml

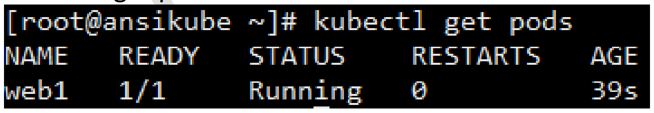
###Imperative method - single command line:-

#kubectl run --generator=run-pod/v1 web1 --image=nginx



```
[root@ansikube ~]# kubectl run --generator=run-pod/v1 web1 --image=nginx
pod/web1 created
```

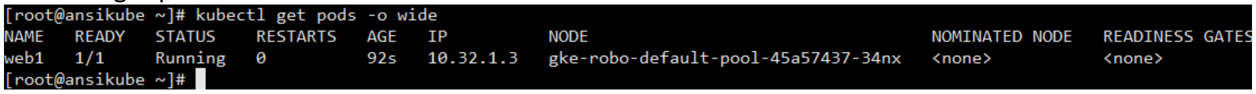
#kubectl get pods



```
[root@ansikube ~]# kubectl get pods
NAME   READY   STATUS    RESTARTS   AGE
web1   1/1     Running   0          39s
```

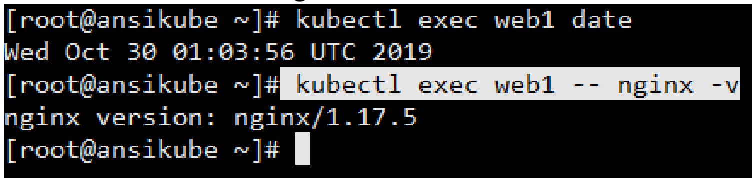
@To get wide information of pod, like on which node it has placed and ip address of the pod.

#kubectl get pods -o wide



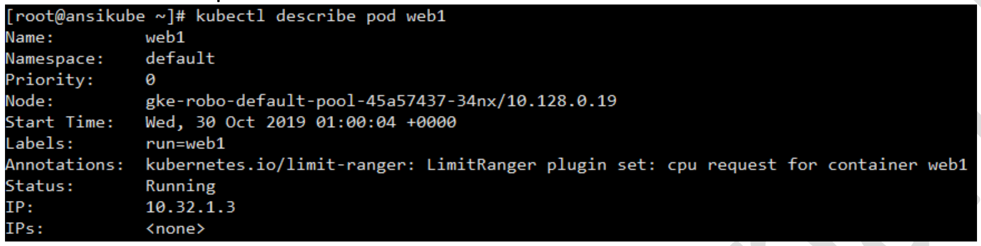
```
[root@ansikube ~]# kubectl get pods -o wide
NAME   READY   STATUS    RESTARTS   AGE   IP          NODE                                   NOMINATED NODE   READINESS GATES
web1   1/1     Running   0          92s   10.32.1.3   gke-robo-default-pool-45a57437-34nx   <none>           <none>
[root@ansikube ~]#
```

@To run command on pod, without login into pod. (kubectl exec <podname> <command>)
#kubectl exec web1 date
kubectl exec web1 -- nginx -v



```
[root@ansikube ~]# kubectl exec web1 date
Wed Oct 30 01:03:56 UTC 2019
[root@ansikube ~]# kubectl exec web1 -- nginx -v
nginx version: nginx/1.17.5
[root@ansikube ~]#
```

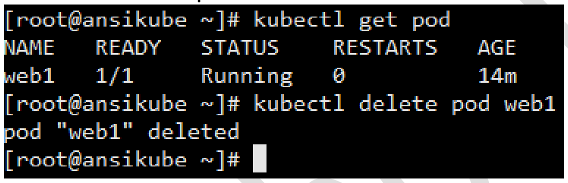
@To get information and events about pod.
#kubectl get pods
#kubectl describe pod web1



```
[root@ansikube ~]# kubectl describe pod web1
Name:         web1
Namespace:    default
Priority:     0
Node:         gke-robo-default-pool-45a57437-34nx/10.128.0.19
Start Time:   Wed, 30 Oct 2019 01:00:04 +0000
Labels:       run=web1
Annotations:  kubernetes.io/limit-ranger: LimitRanger plugin set: cpu request for container web1
Status:       Running
IP:           10.32.1.3
IPs:          <none>
```

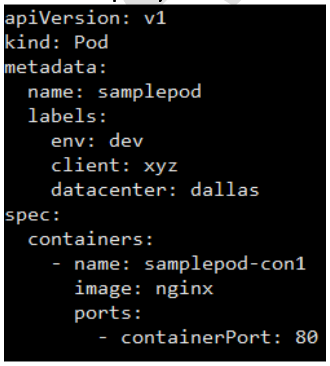
Note:- With describe command, will be display the pod detailed information and to read logs or event about pod from start to end, can see it at events, even it helpful for troubleshoot.

@To delete pod:-
#kubectl get pod
#kubectl delete pod web1



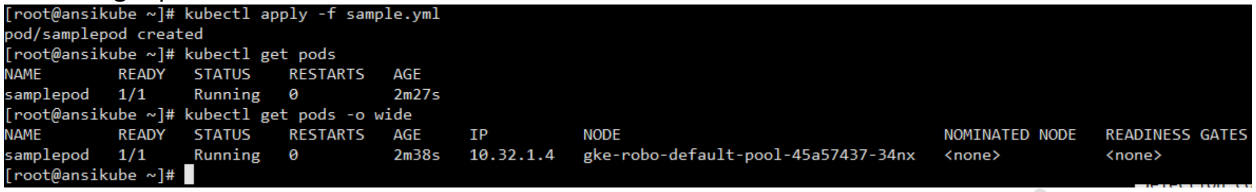
```
[root@ansikube ~]# kubectl get pod
NAME   READY   STATUS    RESTARTS   AGE
web1   1/1     Running   0          14m
[root@ansikube ~]# kubectl delete pod web1
pod "web1" deleted
[root@ansikube ~]#
```

###Declarative Method - writing a manifest file using yaml or json.
:- To create a pod, mandatory to have 4 types of parameters (apiVersion,KIND,metadata,spec)
Note:- Under metadata, labels are very important, and it will be in epar value and in kubernetes pod will be identify with label only, we can add multiple label to pods.
#kubectl explain pod
@@Create a single container pod.
#cat sample.yml



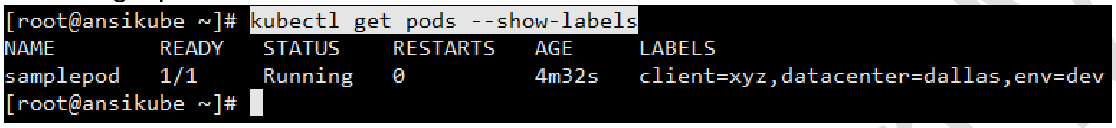
```
apiVersion: v1
kind: Pod
metadata:
  name: samplepod
  labels:
    env: dev
    client: xyz
    datacenter: dallas
spec:
  containers:
    - name: samplepod-con1
      image: nginx
      ports:
        - containerPort: 80
```

#kubectl apply -f sample.yml
#kubectl get pods
#kubectl get pods -o wide



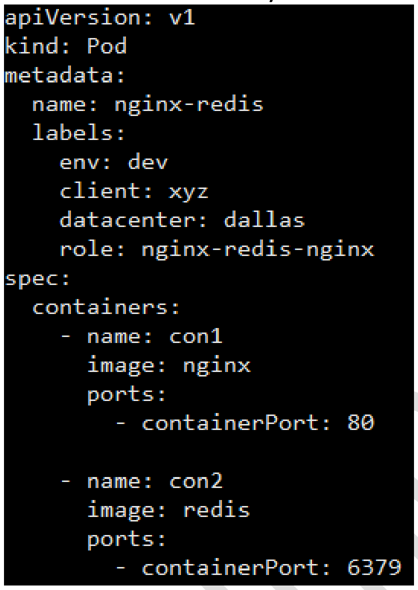
```
[root@ansikube ~]# kubectl apply -f sample.yml
pod/samplepod created
[root@ansikube ~]# kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
samplepod   1/1     Running   0          2m27s
[root@ansikube ~]# kubectl get pods -o wide
NAME        READY   STATUS    RESTARTS   AGE     IP          NODE                                    NOMINATED NODE   READINESS GATES
samplepod   1/1     Running   0          2m38s   10.32.1.4   gke-robo-default-pool-45a57437-34nx   <none>           <none>
[root@ansikube ~]#
```

<u>@To find the label info:-</u>
#kubectl get pods --show-labels



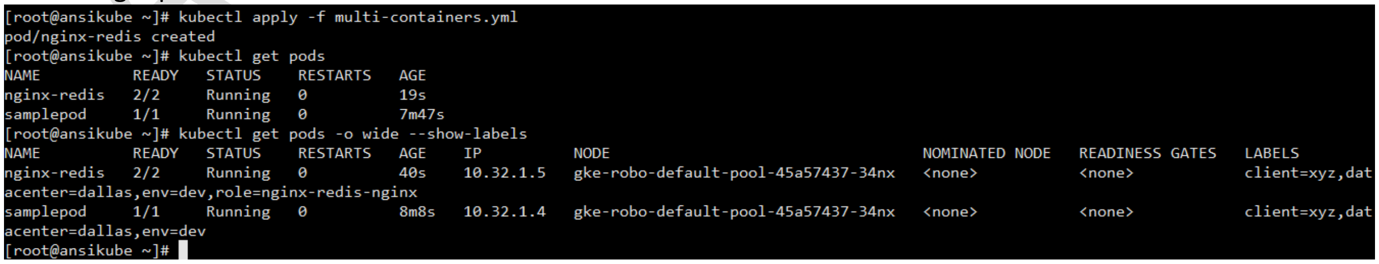
```
[root@ansikube ~]# kubectl get pods --show-labels
NAME        READY   STATUS    RESTARTS   AGE     LABELS
samplepod   1/1     Running   0          4m32s   client=xyz,datacenter=dallas,env=dev
[root@ansikube ~]#
```

<u>@@Create a multiple container pods.</u>
#cat multi-containers.yml



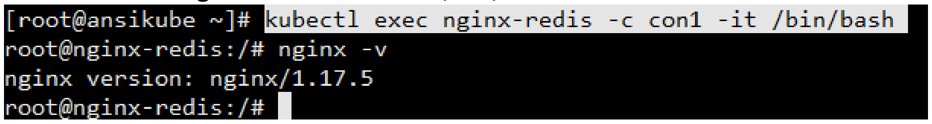
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-redis
  labels:
    env: dev
    client: xyz
    datacenter: dallas
    role: nginx-redis-nginx
spec:
  containers:
    - name: con1
      image: nginx
      ports:
        - containerPort: 80

    - name: con2
      image: redis
      ports:
        - containerPort: 6379
```

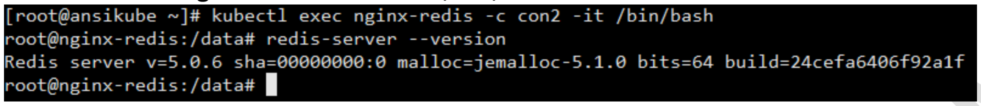
#kubectl apply -f multi-containers.yml
#kubectl get pods
#kubectl get pods -o wide --show-labels



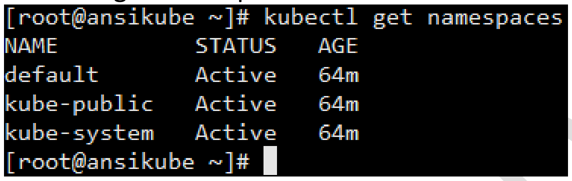
```
[root@ansikube ~]# kubectl apply -f multi-containers.yml
pod/nginx-redis created
[root@ansikube ~]# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-redis   2/2     Running   0          19s
samplepod     1/1     Running   0          7m47s
[root@ansikube ~]# kubectl get pods -o wide --show-labels
NAME          READY   STATUS    RESTARTS   AGE    IP          NODE                                    NOMINATED NODE   READINESS GATES   LABELS
nginx-redis   2/2     Running   0          40s    10.32.1.5   gke-robo-default-pool-45a57437-34nx   <none>           <none>            client=xyz,dat
acenter=dallas,env=dev,role=nginx-redis-nginx
samplepod     1/1     Running   0          8m8s   10.32.1.4   gke-robo-default-pool-45a57437-34nx   <none>           <none>            client=xyz,dat
acenter=dallas,env=dev
[root@ansikube ~]#
```

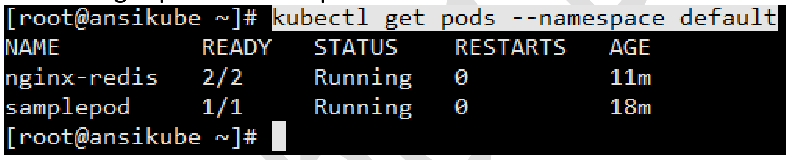
Note:- To verify the containers status which created under pod, just login to container by mentioning pod and container details with kubectl command.

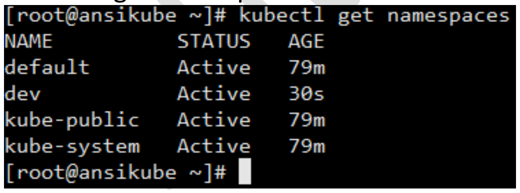
#kubectl get pods
#kubectl describe pod nginx-redis
kubectl exec nginx-redis -c con1 -it /bin/bash

```
[root@ansikube ~]# kubectl exec nginx-redis -c con1 -it /bin/bash
root@nginx-redis:/# nginx -v
nginx version: nginx/1.17.5
root@nginx-redis:/#
```

kubectl exec nginx-redis -c con2 -it /bin/bash

```
[root@ansikube ~]# kubectl exec nginx-redis -c con2 -it /bin/bash
root@nginx-redis:/data# redis-server --version
Redis server v=5.0.6 sha=00000000:0 malloc=jemalloc-5.1.0 bits=64 build=24cefa6406f92a1f
root@nginx-redis:/data#
```

<u>##Namespace##</u>
 Below are default namespaces. for kubernetes all system component pods are running in kube-system only. (API server, kubelet , DNS server etc... the Namespace will be creating as a virtual boundaries inside kubenetes cluster, example:- we will be having (dev,stage,test,prod) environment and for those infrastructure will be separate , so namespace will be used for separating the env with limiting boundaries like access between two env and become secure inside the Kubernetes cluster.

#kubectl get namespaces

```
[root@ansikube ~]# kubectl get namespaces
NAME          STATUS   AGE
default       Active   64m
kube-public   Active   64m
kube-system   Active   64m
[root@ansikube ~]#
```

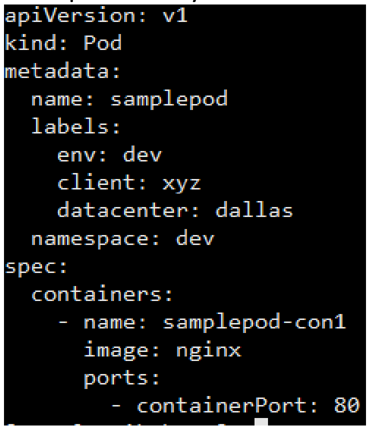
#kubectl get pods --namespace default

```
[root@ansikube ~]# kubectl get pods --namespace default
NAME          READY   STATUS    RESTARTS   AGE
nginx-redis   2/2     Running   0          11m
samplepod     1/1     Running   0          18m
[root@ansikube ~]#
```

<u>@Creating new namespace in Kubernetes cluster.</u>
#kubectl create namespace dev
#kubectl get namespaces

```
[root@ansikube ~]# kubectl get namespaces
NAME          STATUS   AGE
default       Active   79m
dev           Active   30s
kube-public   Active   79m
kube-system   Active   79m
[root@ansikube ~]#
```

<u>@Create a pod in specific dev namespace.</u>

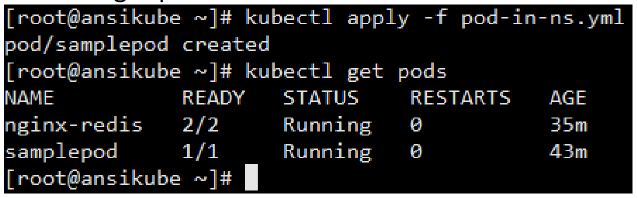
Note:- below is an example to create a pod on dev namespace environment.

# cat pod-in-ns.yml



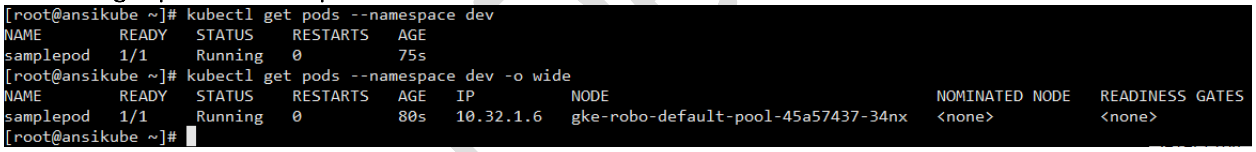
```
apiVersion: v1
kind: Pod
metadata:
  name: samplepod
  labels:
    env: dev
    client: xyz
    datacenter: dallas
  namespace: dev
spec:
  containers:
    - name: samplepod-con1
      image: nginx
      ports:
        - containerPort: 80
```

#kubectl apply -f pod-in-ns.yml
#kubectl get pods



```
[root@ansikube ~]# kubectl apply -f pod-in-ns.yml
pod/samplepod created
[root@ansikube ~]# kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-redis   2/2     Running   0          35m
samplepod     1/1     Running   0          43m
[root@ansikube ~]#
```

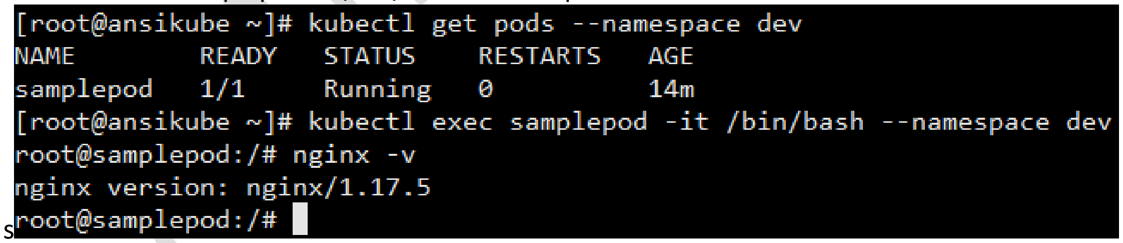
#kubectl get pods --namespace dev
#kubectl get pods --namespace dev -o wide



```
[root@ansikube ~]# kubectl get pods --namespace dev
NAME        READY   STATUS    RESTARTS   AGE
samplepod   1/1     Running   0          75s
[root@ansikube ~]# kubectl get pods --namespace dev -o wide
NAME        READY   STATUS    RESTARTS   AGE   IP          NODE                                   NOMINATED NODE   READINESS GATES
samplepod   1/1     Running   0          80s   10.32.1.6   gke-robo-default-pool-45a57437-34nx    <none>           <none>
[root@ansikube ~]#
```

<u>@login into specific namespace pod.</u>

kubectl get pods --namespace dev
#kubectl exec samplepod -it /bin/bash --namespace dev



```
[root@ansikube ~]# kubectl get pods --namespace dev
NAME        READY   STATUS    RESTARTS   AGE
samplepod   1/1     Running   0          14m
[root@ansikube ~]# kubectl exec samplepod -it /bin/bash --namespace dev
root@samplepod:/# nginx -v
nginx version: nginx/1.17.5
root@samplepod:/#
```

s