# Working with Pods & Namespaces

1. **Create a directory login to the directory and perform the below.**

**2 . Create a YAML (vim mypod.yaml) for Pod and name the Pod as mypod**

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-demo
    image: nginx:1.9.1
    ports:
    - containerPort: 80
```

**Create a Pod**

$ kubectl create -f mypod.yaml

**Create a Namespace**

$ kubectl create ns test

**Create a Pod in a given Namespace**

$ kubectl create -f mypod.yaml -n test

**List the Pods**

$ kubectl get pods

**List the Pods with additional detail**

$ kubectl get pods -o wide

**List the Pods in a given namespace**

$ kubectl get pods -n test

## Create a Multi-Container Pod YAML (multi-c-pod.yaml)

```
apiVersion: v1
kind: Pod
metadata:
  name: multicontainer
spec:
  volumes:
  - name: html
    emptyDir: {}
  containers:
  - name: con1
    image: nginx
    volumeMounts:
    - name: html
      mountPath: /usr/share/nginx/html
  - name: con2
    image: debian
    volumeMounts:
    - name: html
      mountPath: /html
    command: ["/bin/sh", "-c"]
    args:
```

```
- while true; do
    date >> /html/index.html;
    sleep 1;
  done
```

**Deploy a Multi-Container Pod**

$ kubectl apply -f multi-c-pod.yaml

**List the Pods**

$ kubectl get pods

**List the Pods with all namespaces**

$ kubectl get pods --all-namespaces

**Delete the Pods.**

$ kubectl delete pod mypod -n test

kubectl delete pod --all

Working with Namespaces:

```
kubectl get namespace
kubectl get ns
kubectl delete ns namspacesnames
```