Lab 2: Working with Namespaces and Pods

1. Via imperative commands:
    - Create a namespace called demo.

```
k create ns demo
```

    - Deploy a simple pod as my-app using the nginx Docker image in the namespace demo.

```
k run my-app --image=nginx -n demo
```

    - Confirm that the pod is running.

```
k get pod -n demo
```

    - Check pod details (for example, start date).

```
k describe pod my-app -n demo
k describe pod my-app -n demo | grep "Started:"
```

    - Check pod logs.

```
k logs my-app -n demo
```

    - List pod in the namespace app without using the option “-n”.

```
k config set-context --current --namespace=demo
k get pod
```

    - Delete pod and the namespace.

```
k delete pod my-app
k config set-context --current --namespace=default
k delete ns demo
```

2. Via imperative commands:
    - Recreate the namespace.

```
cat <<EOF >> ns.yaml
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: demo
EOF
```

```
k apply -f ns.yaml
k get ns
```

- Recreate pod. Include a label as “name : my-app” and container port as 8088.

```
cat <<EOF >> pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-app
  namespace: demo
  labels:
    name: my-app
spec:
  containers:
  - name: my-app
    image: nginx
    ports:
      - containerPort: 8088
EOF
```

```
k apply -f pod.yaml
k get pod -n demo
```

- Delete pod and namespace.

```
k delete pod my-app -n demo
k delete ns demo
```

- Extra

- Create a pod yaml from the imperative command.

```
k run my-app --image=nginx --dry-run=client -o yaml > extra.yaml
```