Creating HTTP Healthchecks

- We will be create a simple container first
- We will be adding liveness probe to it.
- We will custom make livesness probe fail and see how k8s restarts the pods.

Create a simple nginx pod without any probe -pod-nginx.yaml

```
vim pod-nginx.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-demo
    image: nginx:1.15-alpine
    ports:
    - containerPort: 80
```

Create the Deployment

```
kubectl apply -f pod-nginx.yaml
```

Working with HTTP probe

Creating a nginx pod with liveness probe with given configuration as below.

```
vim liveness-http-pod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-liveness
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-demo
    image: nginx:1.15-alpine
    ports:
```

```
    - containerPort: 80
    livenessProbe:
      httpGet:
        path: /
        port: 80
      initialDelaySeconds: 15
      periodSeconds: 5
      timeoutSeconds: 1
```

Deploy the pod

```
kubectl apply -f liveness-http-pod.yaml
```

Example demonstrating failing liveness probe.

```
vim liveness-http-fail-pod.yaml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-liveness-fail
  labels:
     app: nginx
spec:
  containers:
  - name: nginx-demo
    image: nginx:1.9.1
    ports:
    - containerPort: 80
    livenessProbe:
      httpGet:
        path: /
        port: 81
      initialDelaySeconds: 15
      timeoutSeconds: 1
```

```
kubectl apply -f liveness-http-fail-pod.yaml
```

Regularly check output of kubectl get pods ..It would be failing