**Resource limits**

* Created a file called pod yaml ,with the below content

apiVersion: v1

kind: Pod

metadata:

  name: web-server-pod

  labels:

    app: web-server

    environment: production

  annotations:

    description: This pod runs the web server

spec:

  containers:

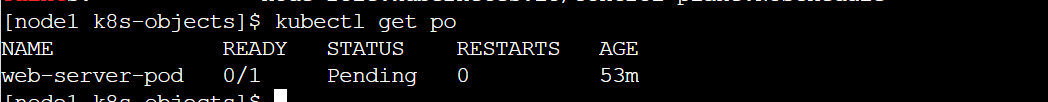
  - name: web-server

    image: nginx:1.14.2

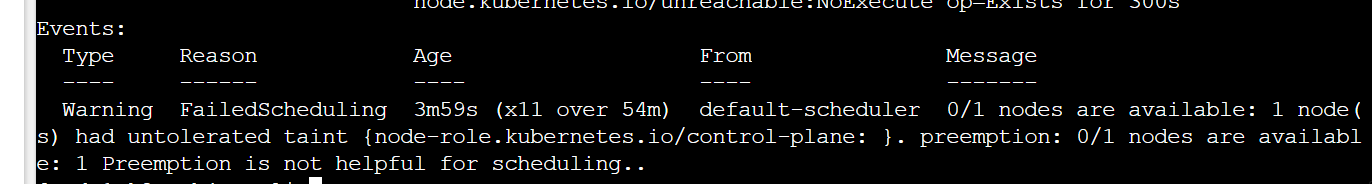
    ports:

    - containerPort: 80

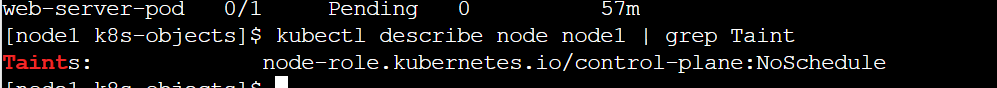
* **command** – kubectl create -f <podname.yaml>



* The pod went into Pending state, describe the pod



* This is because the control plane node has a taint for scheduling, adding a toleration in the pod spec would help



  tolerations:

  - key: "node-role.kubernetes.io/control-plane"

    effect: "NoSchedule"

* After adding the above tolerations in yaml file, the pod entered into running state
* Taint is a spec that will be looking for a toleration, If a taint is defined in a node, and we want to run the deployment on that node then toleration has to be added in yaml file, if not this app will be deployed in the node which has no taint

**Resource limits and requests**

* Request will be allocated by the scheduler to the container when starting up, after exceeding the limit the application not run as expected
* Limits is something that will be not be exceeded , If the container exceeds the limits it would enter into OOMKilled state

apiVersion: v1

kind: Pod

metadata:

  name: resource-limits-pod

spec:

  containers:

  - name: nginx-container

    image: nginx:latest

    resources:

      limits:

        memory: "256Mi"  # Maximum memory usage allowed

        cpu: "0.5"       # Maximum CPU usage allowed (0.5 cores)

      requests:

        memory: "128Mi"  # Initial memory request

        cpu: "0.25"      # Initial CPU request