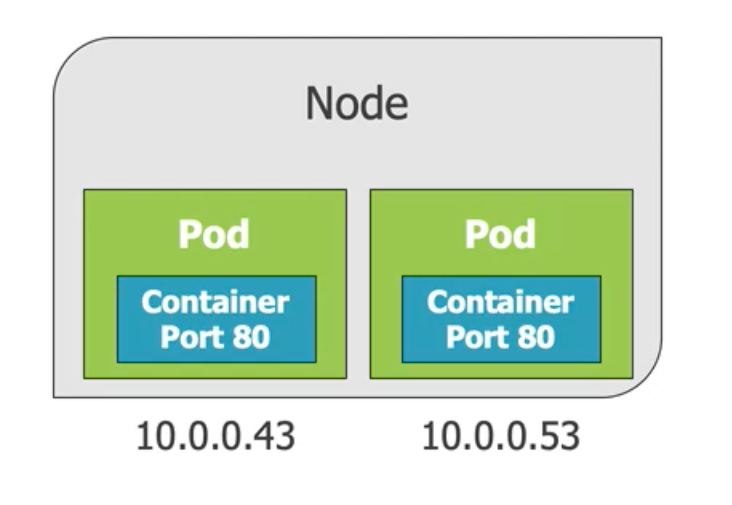
**PODS**

* Is the basic execution unit of Kubernetes application- the smallest and simplest unit in the Kubernetes object model that you create or deploy
* Environment for containers
* Can store multiple containers
* It shares its IP, memory, volumes accross containers
* Scale horizontally by adding Pod replicas
* Pods live and die but never come back to life. If Kubernetes sees the pods that is unhealthy or sick, it can automatically remove it and replace it
* **A screenshot of a cell phone

  Description automatically generated**
* Pods within a node are going to have a unique IP address and this will be a cluster IP address by default and containers in the pods will have their own unique ports
* Pod containers share the same Network namespace (share IP/port)
* Pod containers have the same loopback network interface (localhost)
* Container processes need to bind to different ports within a Pod. If you have multiple container within one pod, then containers should have unique ports
* ****
* Pods do not span on multiple nodes
* There are several different ways to schedule a Pod: a) **kubectl run** commandb) **kubectl create/apply** command. For example: **kubectl run [podname] –image=nginx:alpine**
* get list of all pods: **kubectl get pods**
* get list of all resources: **kubectl get all**
* pods and containers are only accessible within the Kubernetes cluster by default. One way to expose a container port externally: **kubectl port-forward.** For example: **kubectl port-forward [name-of-pod] 8080:80 –** 8080 is external port, 80 is internal
* Running a Pod will cause a deployment to be created. To delete a Pod use **kubectl delete pod** or find the deployment and use **kubectl delete deployment.** Fe. **kubectl delete pod [name-of-pod].** if you delete pod, kubernetes will bring new one up, so if you want to delete it forever, you should delete deployment
* Run Kubernetes locally:
  + git clone <https://github.com/kubernetes/minikube.git>
  + **minikube start** command, **minikube status**
* **A close up of a sign

  Description automatically generated**
* **Kubectl port-forward** for exposing the pod externally on port 8080. you can access it now in browser with localhost:8080
* A close up of a sign

  Description automatically generated
* To create a pod using YAML use **kubectl create -f file.pod.yml –dry-run –validate=true** (this also validates the yaml file)
* To create or apply changes to a pod using YAML use **kubectl apply -f file.pod.yml**
* **A screenshot of a cell phone

  Description automatically generated**
* use –save-config **kubectl create -f file.pod.yml –save-config**  - this will create some annotations, so when we do apply later, it will take whatever we are trying to apply, comapre it to what was there in the first place and then we can override specific settings
* **A screenshot of a cell phone

  Description automatically generated**
* **kubectl edit** you can edit your yaml file directly in console
* **kubectl delete -f file.pod.yml**
* **kubectl describe pod\_name**
* enter in pod with **kubectl exec pod\_name -it sh**
* **POD HEALTH**
  + Kubernetes relies on Probes to determine the health of a Pod container. A Probe is a diganostic performed periodically by the kubelet on container.
  + Two types of probes:  **Liveness Probe and Readiness Probe**
  + **Liveness probe** can be used to determine if a Pod is healhy and running as expected. **WHEN SHOULD A CONTAINER START RECEIVING TRAFFIC?**
  + **Readiness probe** can be used to determine if a Pod should receive requests. **WHEN SHOULD A CONTAINER RESTART?**
  + Failed pod containers are recreated by default (restartPolicy defaults to ALWAYS)

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**