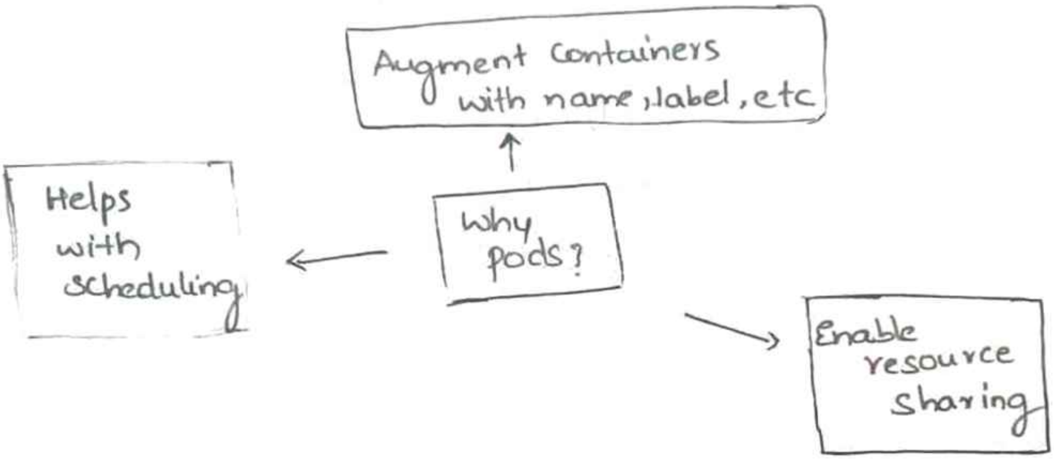
Pods :-

Controllers gift Pods with super powers like self-healing, Scaling, etc.

* Labels let you group pods.

Why pods ?



Pods are mortal. That is why applications should always store State & data outside the Pod.

Deploying static Pods :-

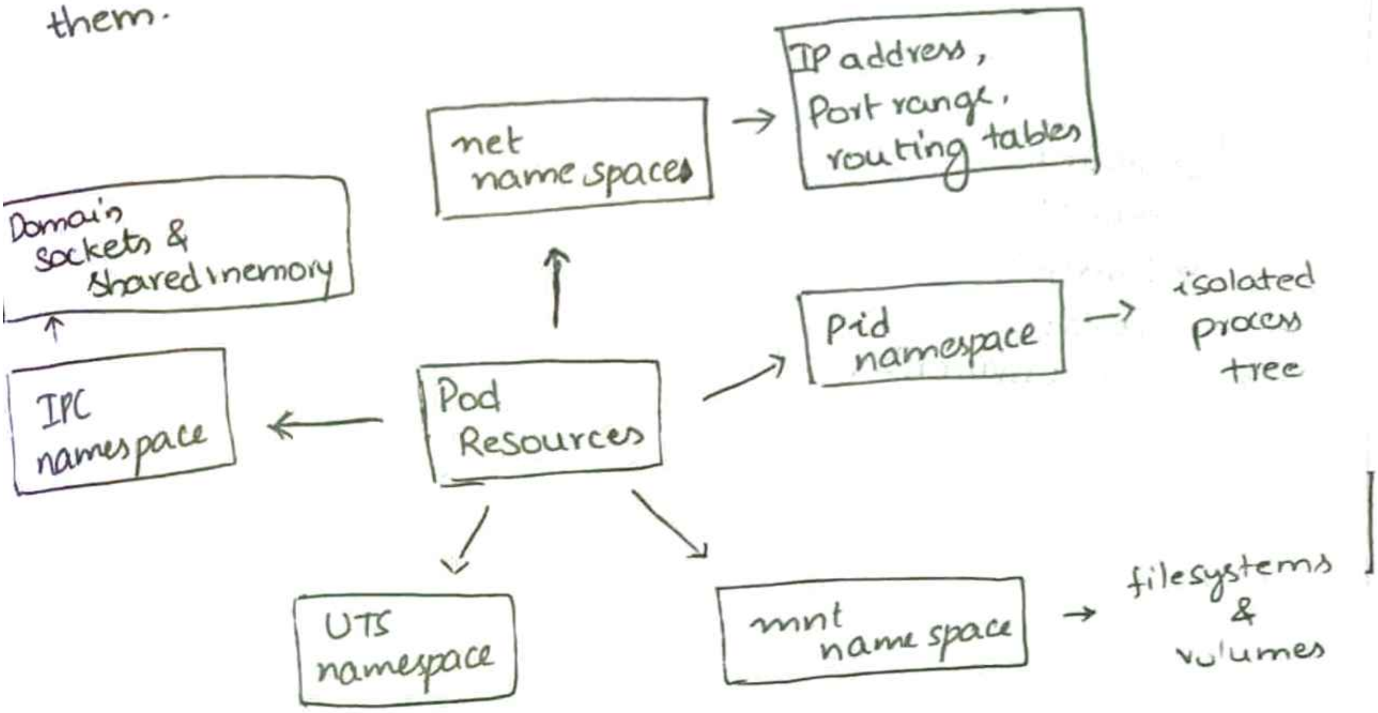
1. Define it in YAML manifest file.
2. POST the YAML to API Server.
3. API server authenticates & authorizes the request.
4. The YAML configuration is validated.
5. Scheduler deploys the pod to a healthy node
6. Local kubelet monitors it.

Think of Pod as OS

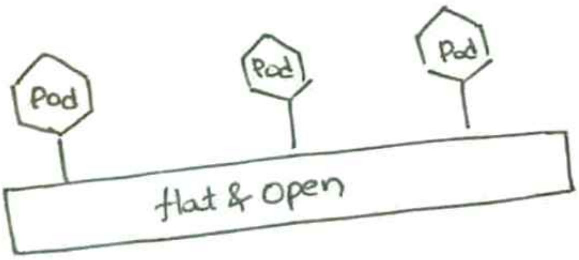
&

Containers as processes running inside

Pod is a collection of resources & Containers inherit them.



Pod networking :-



* Every Pod has unique IP address.
* Pod network is flat & open meaning any pod in the cluster could communicate with other Pod in the cluster.

Short-lived & long-lived Pods

↳ batch jobs etc

↳ web servers, etc

Pod manifest files :-

kind : Pod

apiVersion : V1

metadata : (names, labels, annotations, NameSpaces etc)

spec :
↳ where containers are defined.

```
spec:
  containers:
   - name:
   o image:
     ports:
      - ContainerPort: 8080.
```

Name Spaces :-

, native way to divide a single k8s cluster into multiple virtual clusters.

* If you don't explicitly define a Namespace for a namespaced object, it will be deployed in a default Namespace

Almost every k8s Object.

```
kubectl createns newNamespace
```

→ create imperatively.

shield-ns.yml

```
kind : NameSpace
apiversion : V1
metadeta:
      name : shield.
```

```
kubectl apply -f shield_ns.yml
```