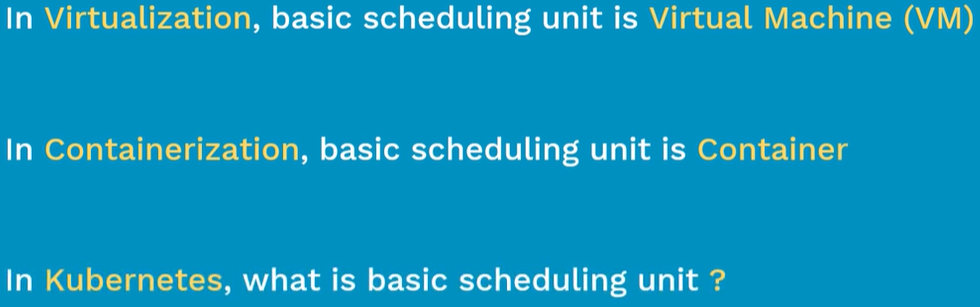
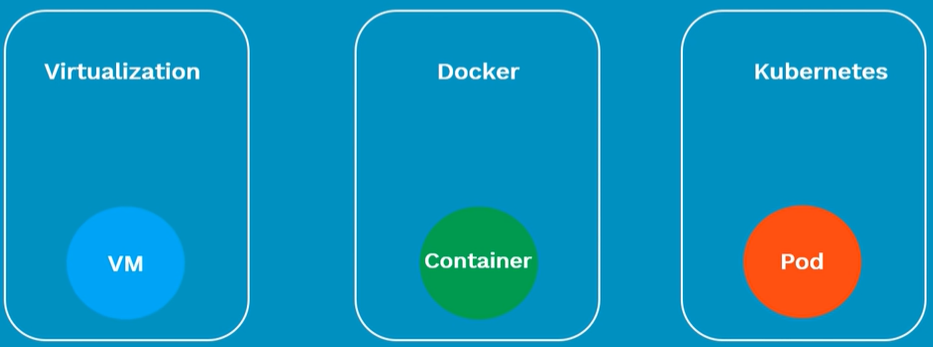
**Introduction:**

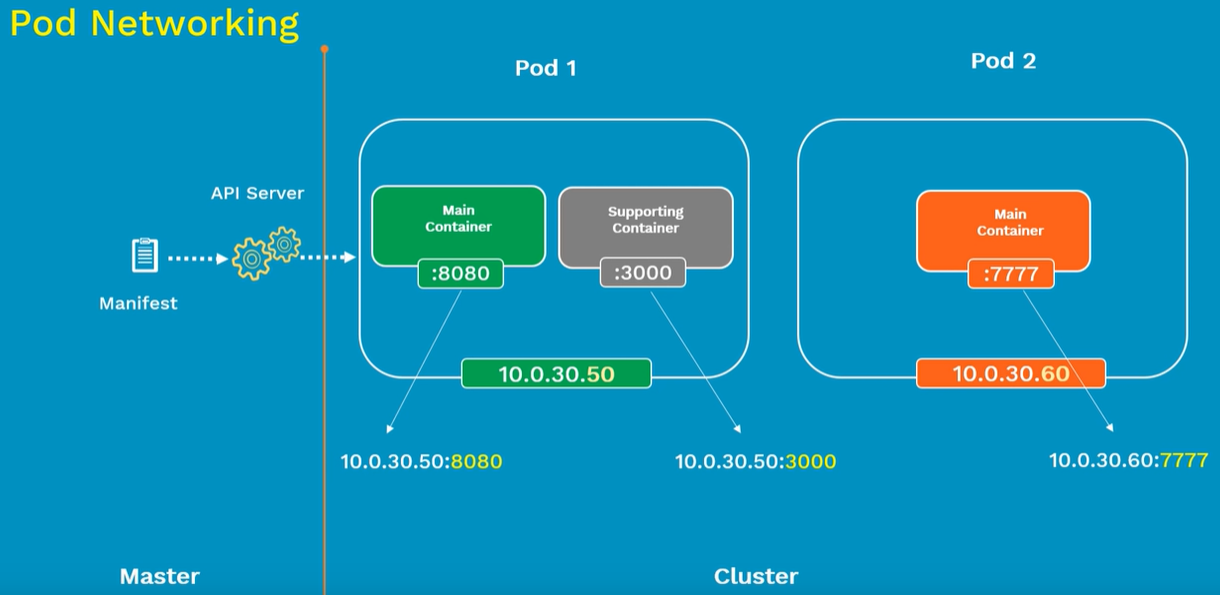




* Sometimes, we may come across a situation where we may need to run multiple containers in a single POD
* Like a container service that helps the main webapps like processing a file or uploading a file which requested by user
* When a new app container is created, the helper container will also be creating and in same case, if app container dies then the helper container will also die because they are running in same POD  
  these containers can access each other with localhost because they share same namespace

**POD networking:**

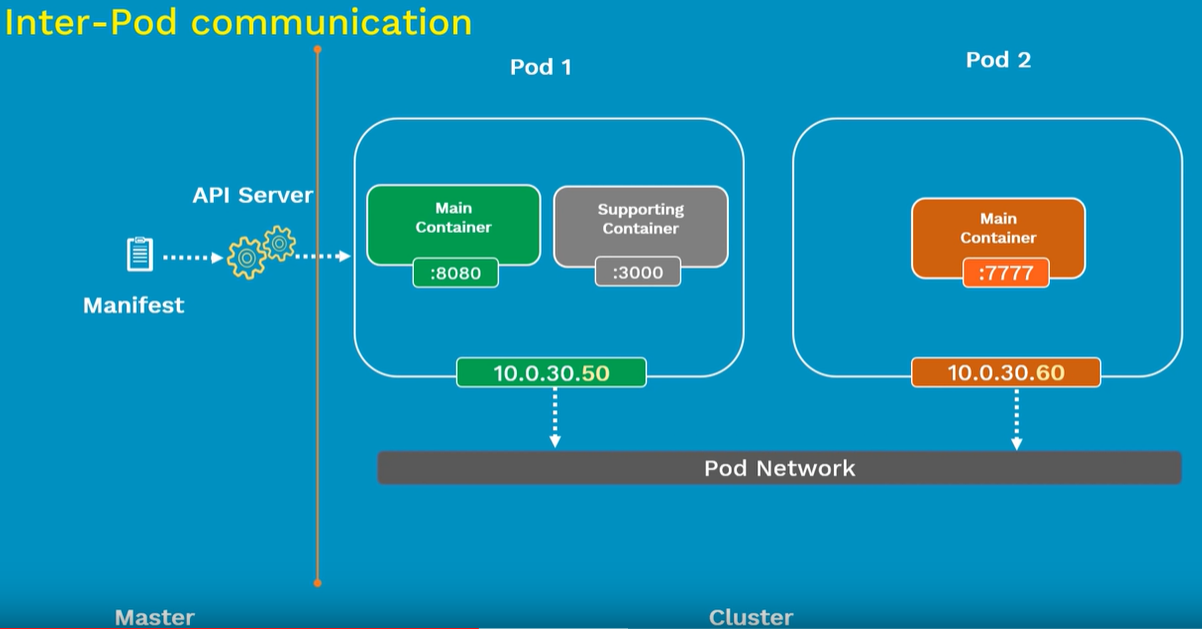
* Once we deploy the pod in kubernetes cluster, it gets a new IP address
* Every POD will have unique IP address



* Containers in the same POD will not just share the same IP address, it also shares the volumes, cgroup limits and same IPC names

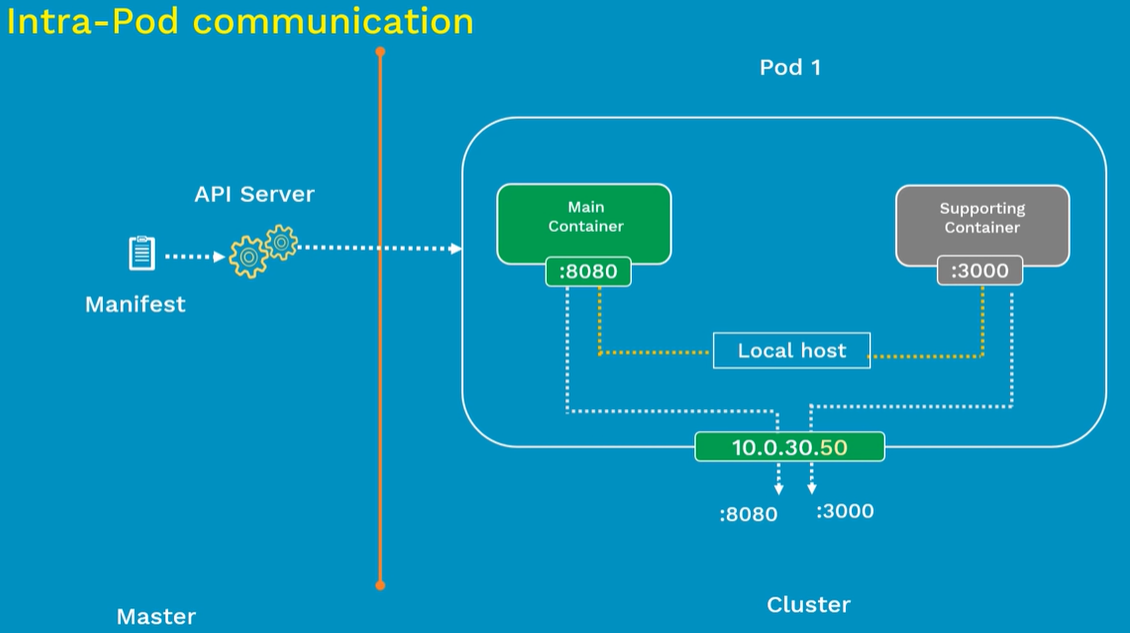
**Inter POD communication:**

* When we setup kubernetes cluster, we use POD network plugin to create the POD network
* Once we are done with that, every POD gets its own IP address. Which means every POD can talk to other PODS directly



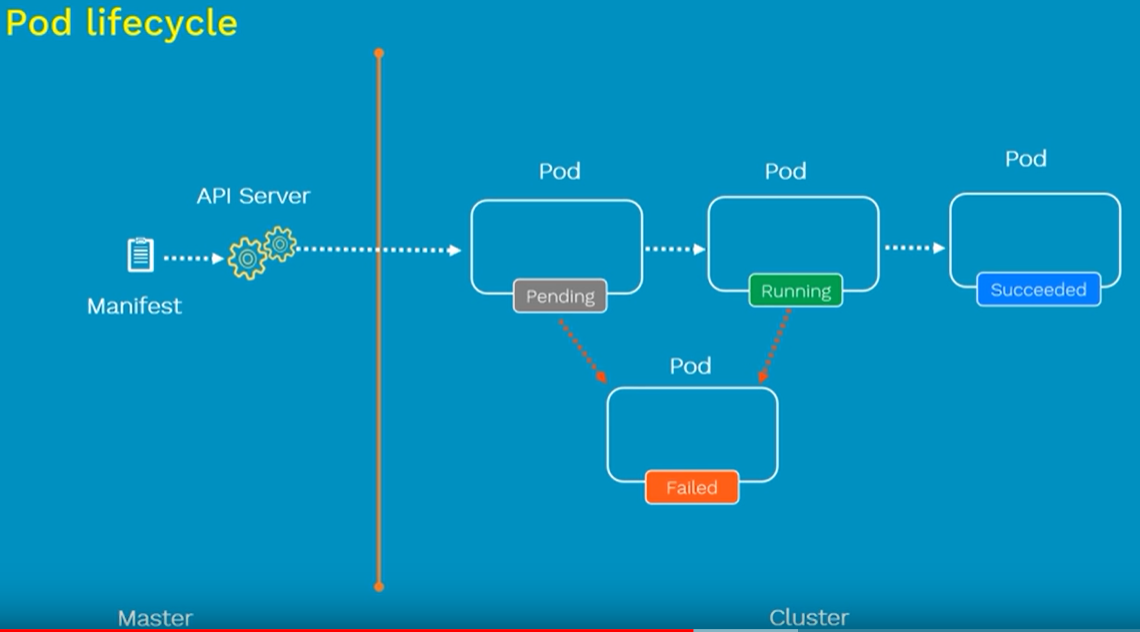
**Intra POD communication:**

* The containers inside the same POD uses the shared local host
* But if we want to run multiple containers in single POD, the containers should run on different ports



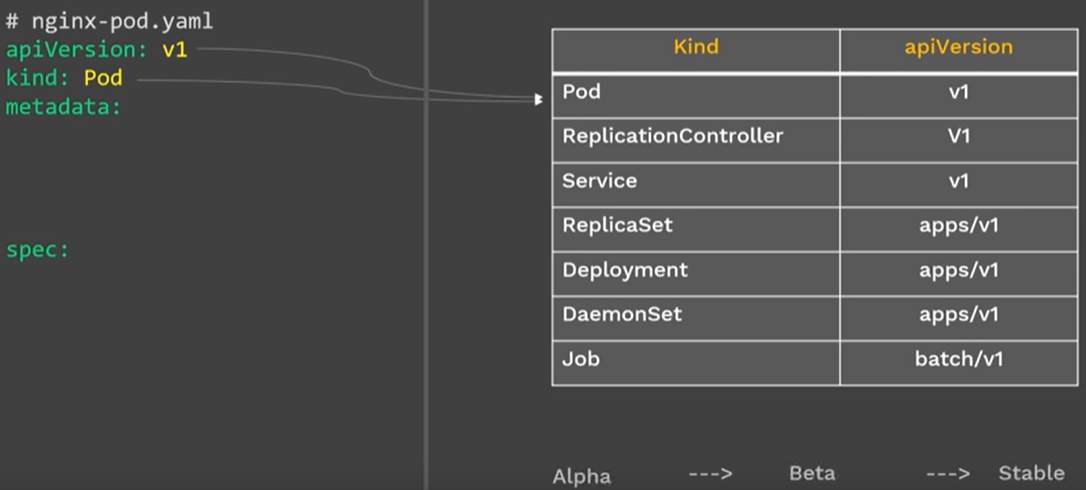
**POD lifecycle:**

* First, we define a pod configuration inside a yaml or json file
* Then we submit the yaml file to API server in master node
* Then it will get scheduled on a worker node inside the cluster. Once it is scheduled, it will be in pending state. During this pending state, node will download the container image and start running the container
* Once that is all done, it goes to running state and if the main purpose of POD is achieved, it gets shutdown and goes to succeeded state
* It goes to failed stage if something goes wrong. Then we need to launch the new POD with same configuration

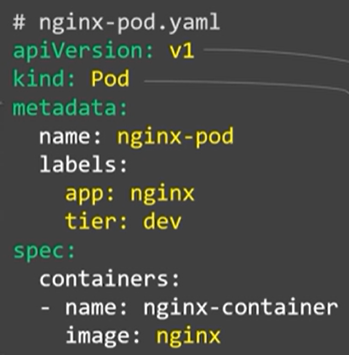


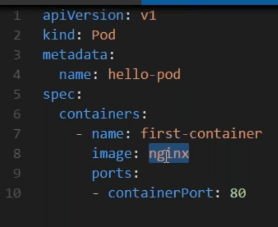
**POD yaml file:**

* Apiversion defines the version number which this kubernetes object belongs to

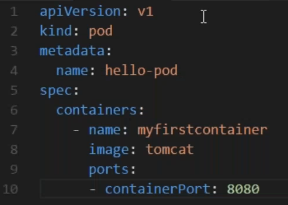


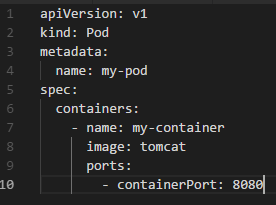
* Any component in kubernetes starts with alpha, then beta and it goes to stable
* The stable version does not contain its name in the version like alpha or beta as above. So, it’s all safe to use
* “metadata” contains two fields. One is “name”, and another is “label”
* Meta data is that what we want to tell about pod. It is like extra information
* Label is just a tag to the POD. If there are many pods running in container then the labels will be used to filter.
* Labels are optional, but it is good to use
* Then the “spec” section contains the container details like name of the container and the container details



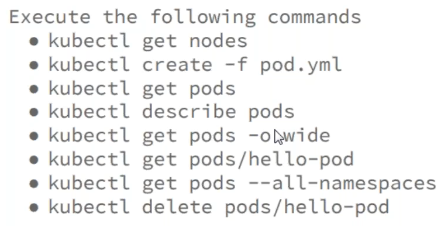


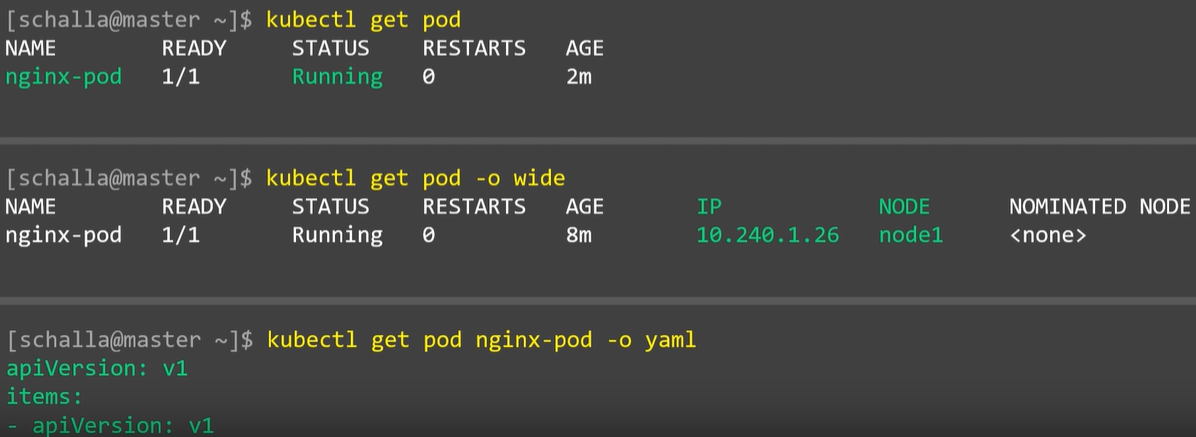
* Basically, we can create any pod like this





**POD commands:**





* Describe command used to describe the pod

“-o wide” is used to view more details of POD

“-o yaml” is used to view the yaml file of a POD