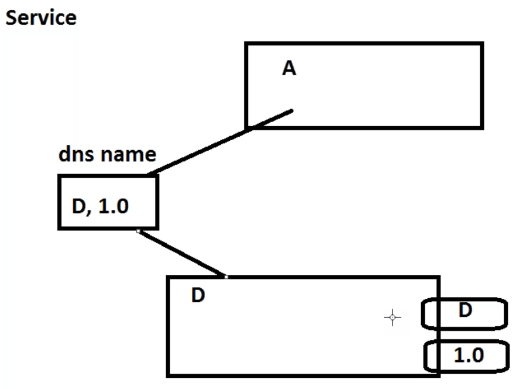
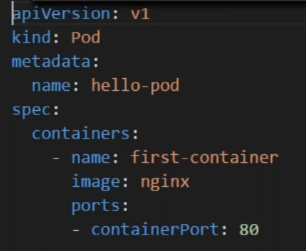
**Manifest:**

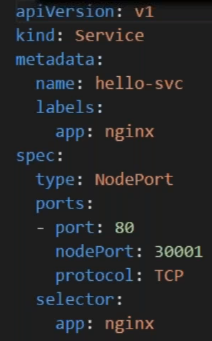
* When we install Kubernetes, first we will have k8s cluster
* Nodes were also called as minions
* Container engine in node is nothing but docker daemon
* Kubernetes will not create containers. It creates pods
* Pod is an execution area where we have network and some other kernel namespaces
* If we have pod A and B, between that we have service. If A wants to communicate with B. then it can’t do directly. With pod IP. It can be possible with service
* Service can identify the pod by the concept called tag or label
* If B has label D and 1.0 and if we requested for the same. It will search for D,1.0 and load balance
* Service will do it only for healthy pods



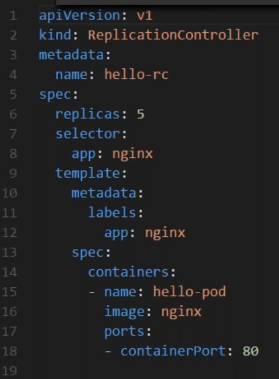
* We achieve all of these by writing manifest



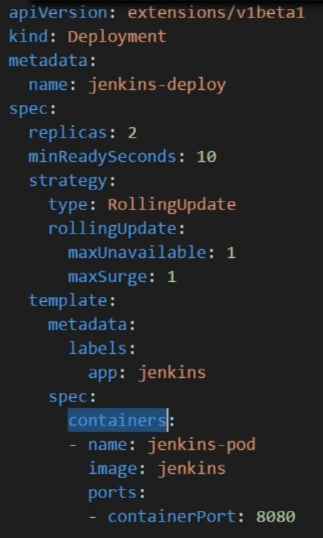
* Kind describes what we need to do
* If we want pod, we need to mention pod as kind
* Apiversion, kubernetes has v1 and v2. Most of the production is in version1
* Metadata name is what we need to call the kind. As above, we need pod called as hello-pod
* And specs, we are naming the container as first-container and the image is nginix which will be called from docker hub



* Just like the same, we are creating service as above
* We are naming it as hello-svc and we are interested in labels called nginix
* And under specs, we configure port and nodeport
* Selector means how we want to find the pod



* As above, we create replication controller



* Above is the manifest file for deployment
* All of these written in kubernetes docs. We would be using them and edit what we want
* We will be writing these files and we will use a component called as kubectl to get all of our job done. We will be interacting with kubectl