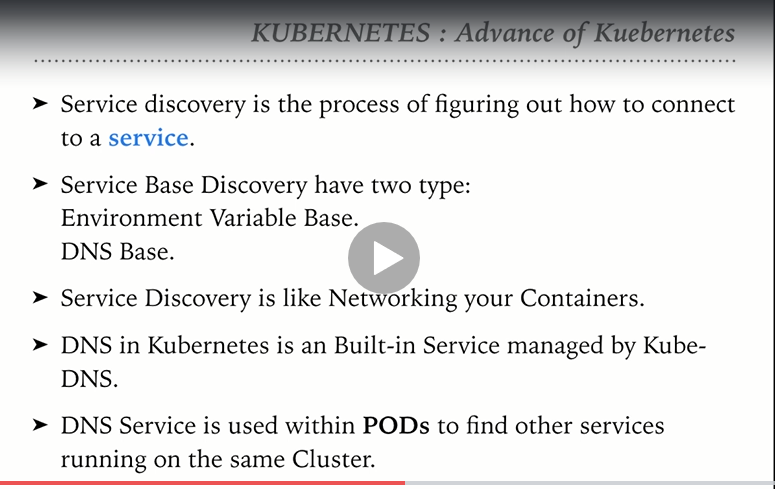
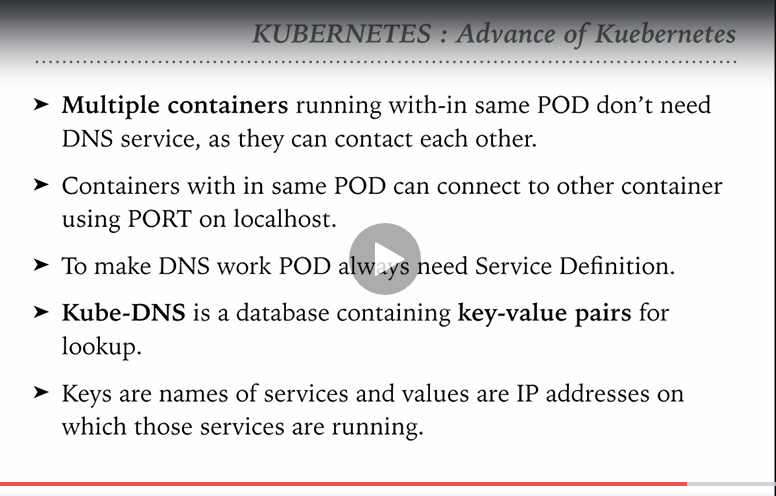
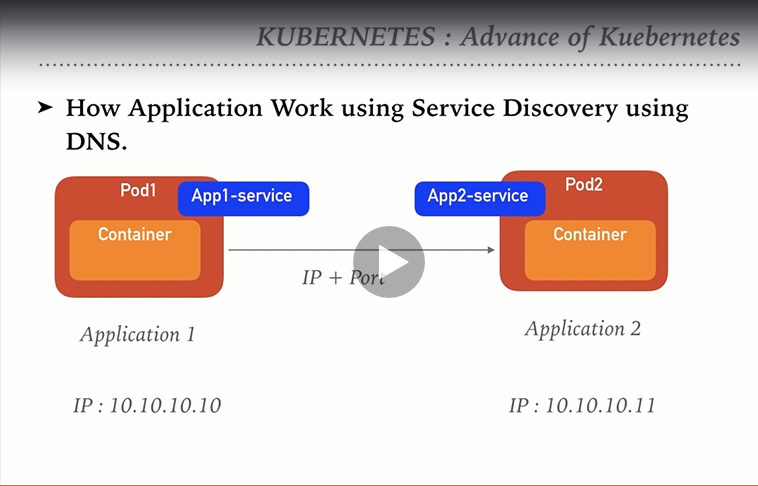
**Topic-1: Service discovery in Kubernetes using DNS**

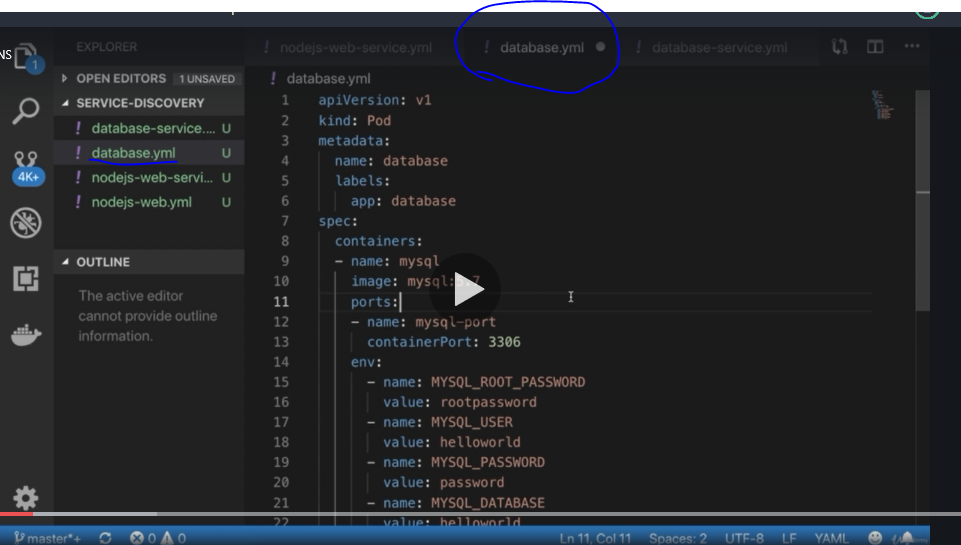


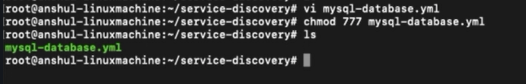




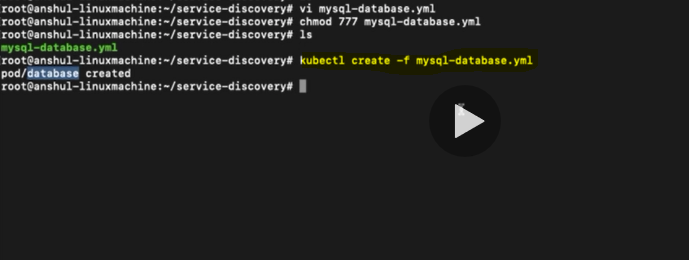
Lab:

Create a database manifest file

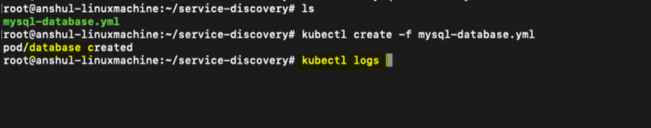




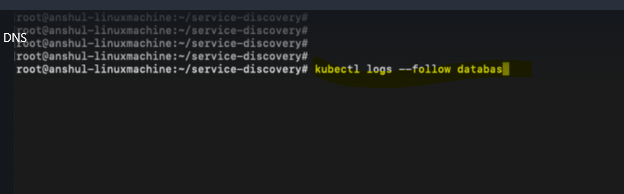
Create pod with the help of the kubectl:



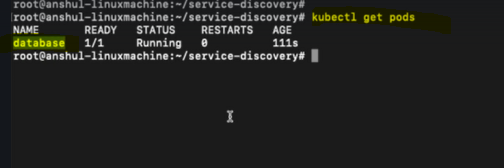
Check the logs for the database pod



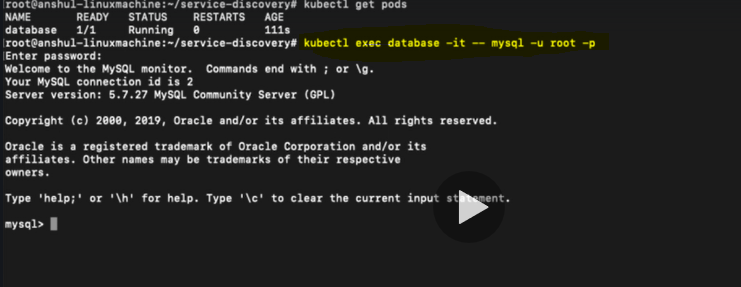
To see the live logs, use below:

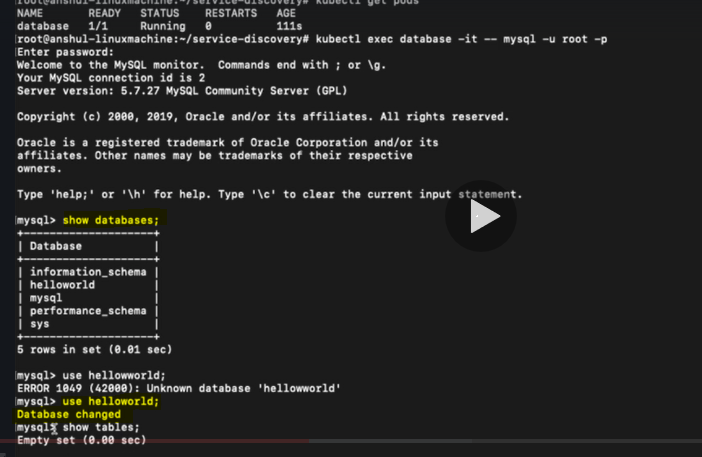


Get pods

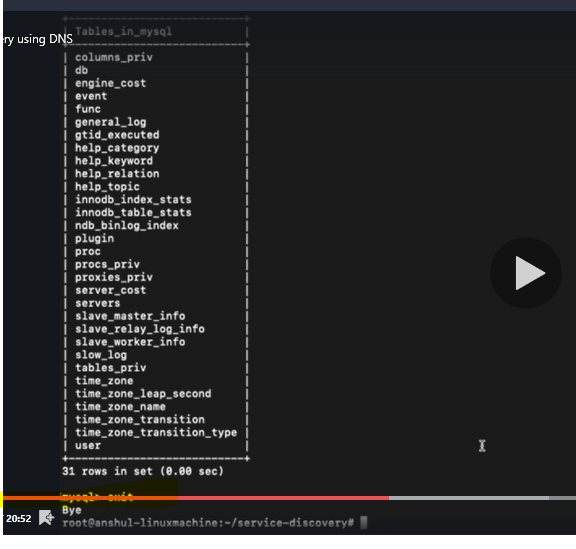


Open the pod **database** in execution mode, like exec mode in container

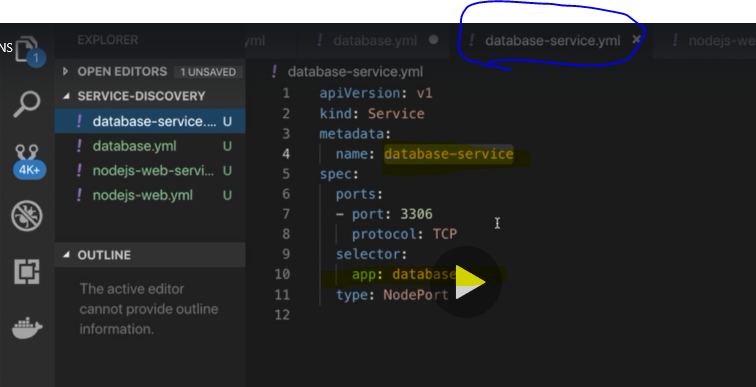




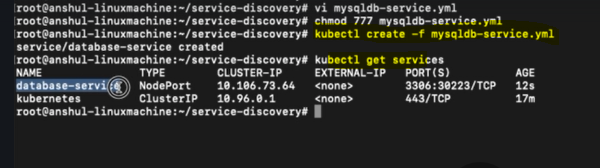
Exit from the mysql database

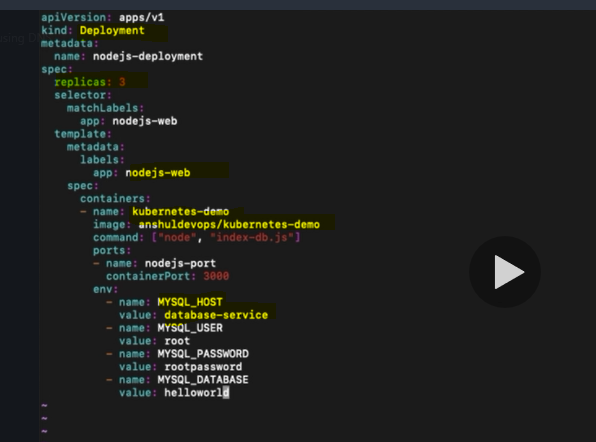


Create a service for database, This is the service for database deployment. Observe that the port name is same as that mentioned in the deployment manifest.



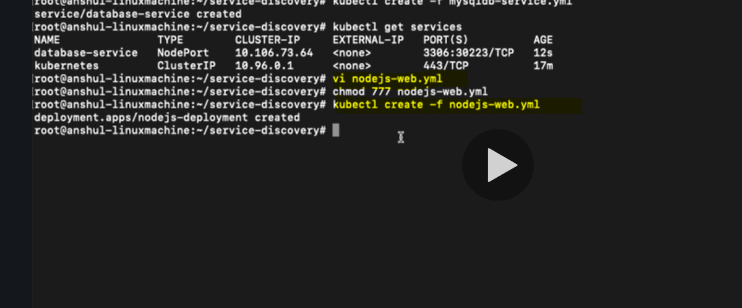
Create the service and verify if the service is up and running



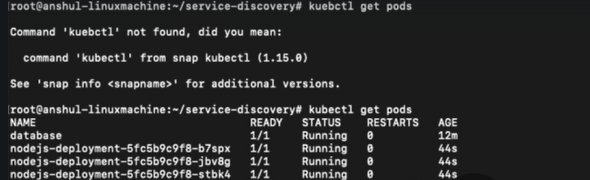
Create a deployment manifest for front end application as shown below:

In the above screenshot, Observe that we are referencing database-service under the MYSQL\_HOST

Create nodejs-web deployment as shown below:



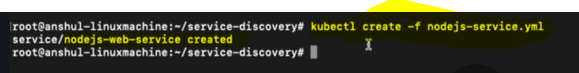
Verify that three pods are created for the above deployment:



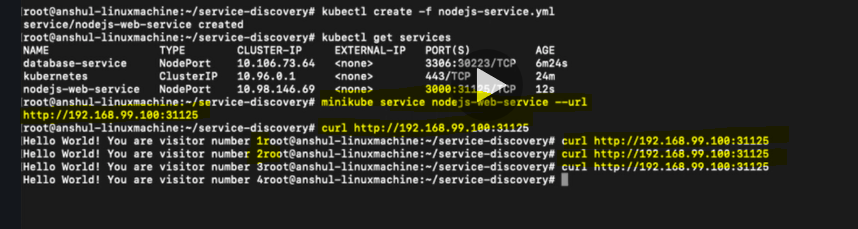
Create a service for front end:



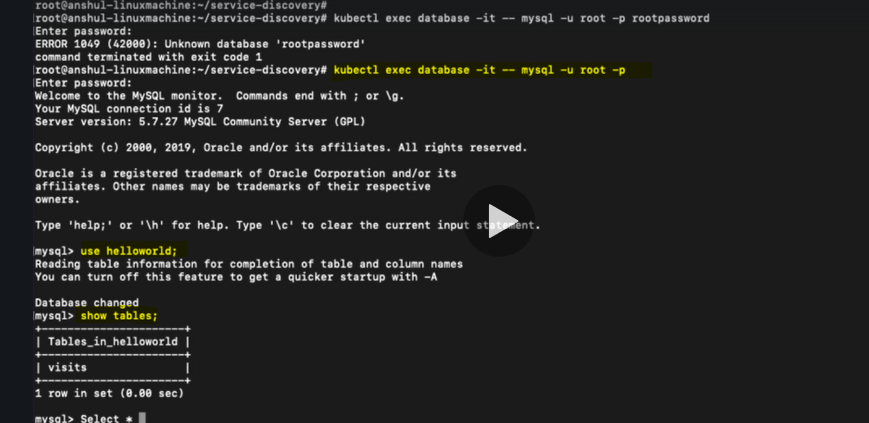
Create a service



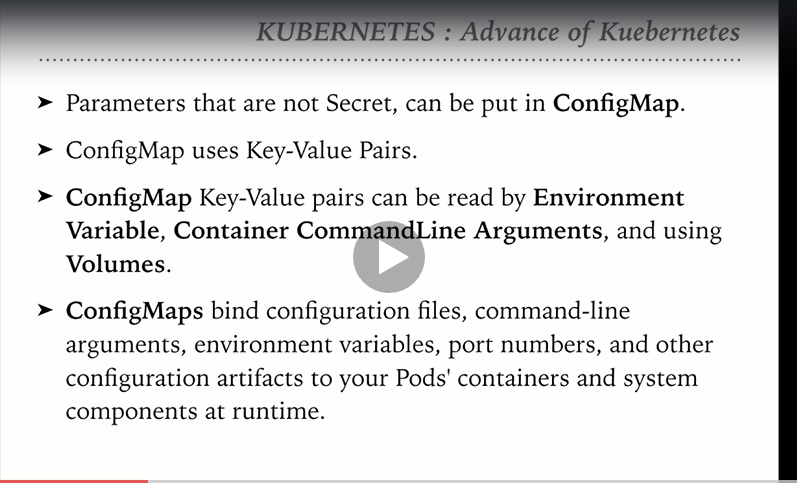
Get the url of the service and invoke the service:

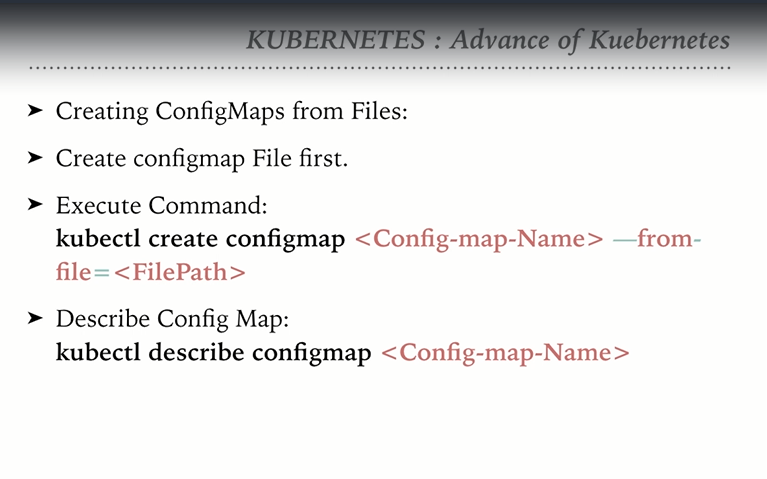


Verify that new table is created after invoking the service, by accessing the container named **database** in the interactive way

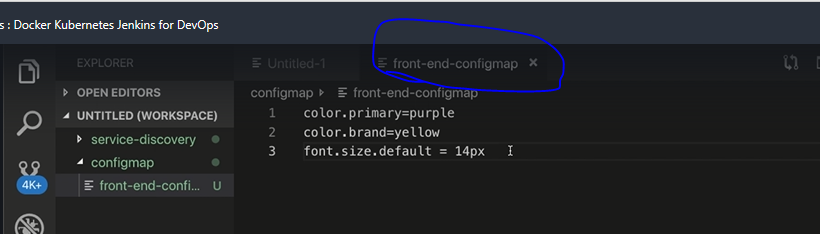


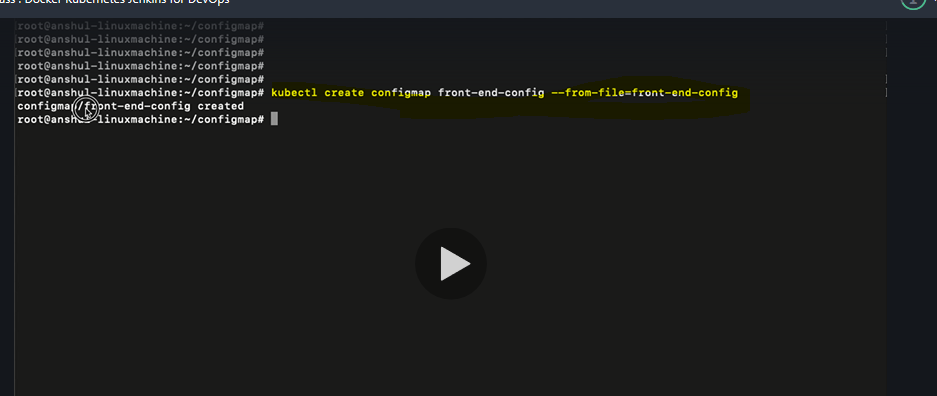
**Topic-2: ConfigMap**

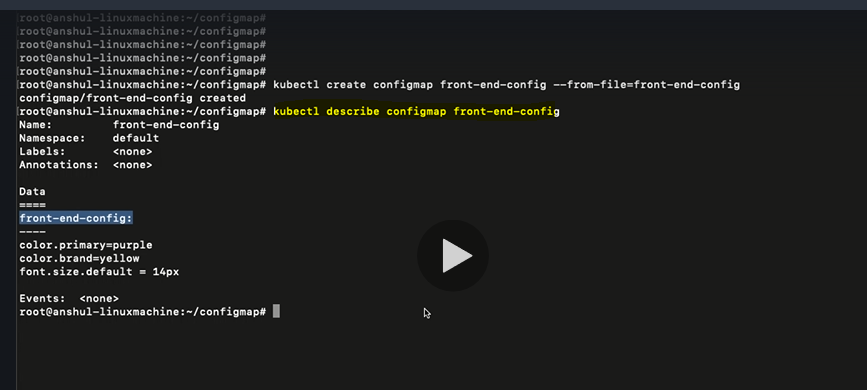


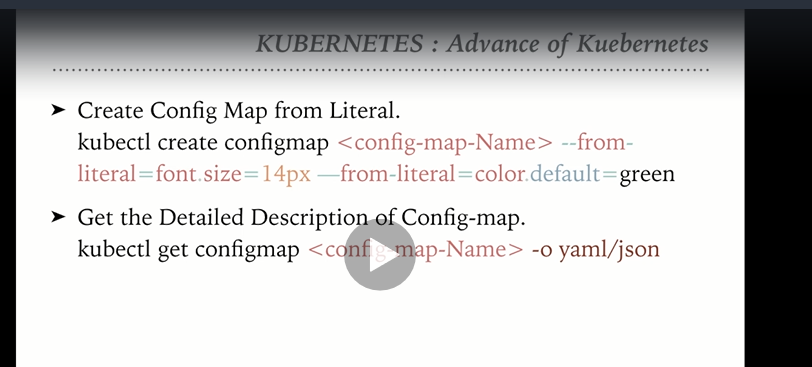


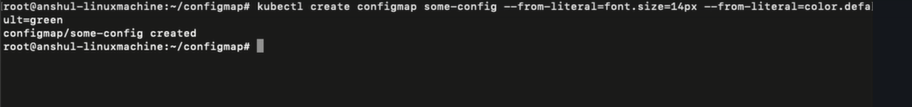
File content



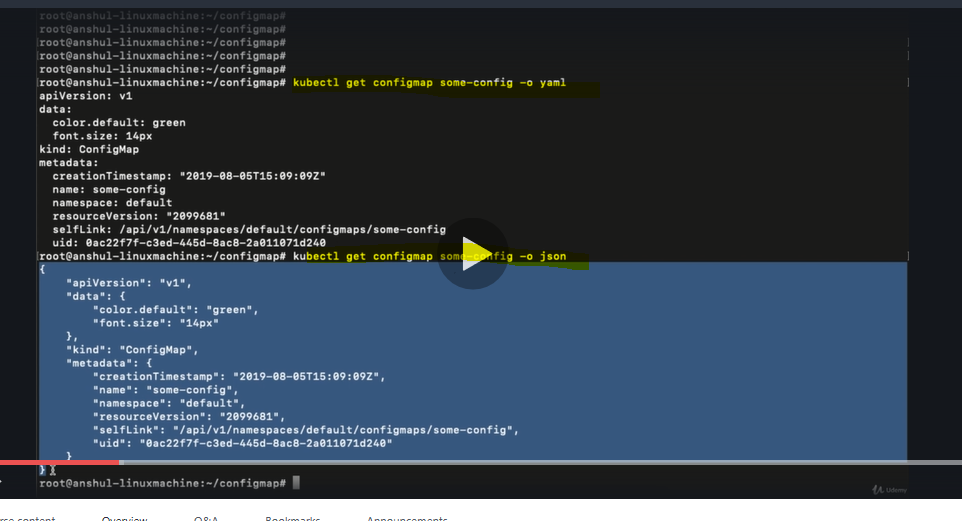




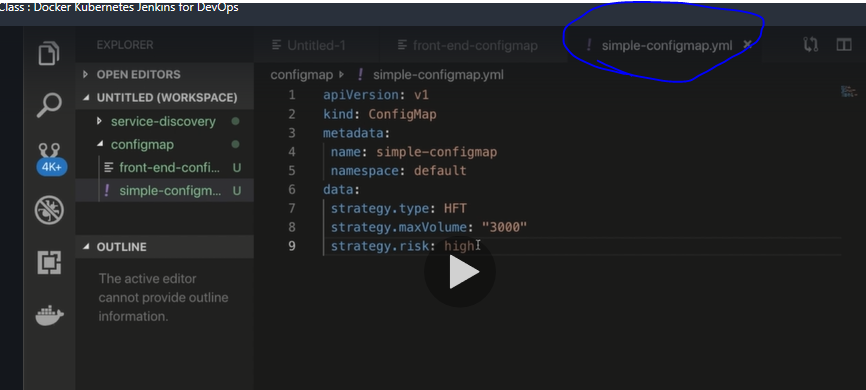


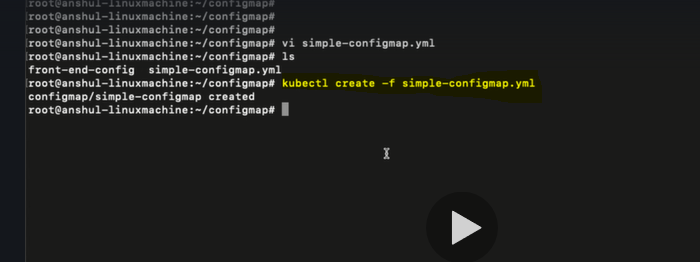


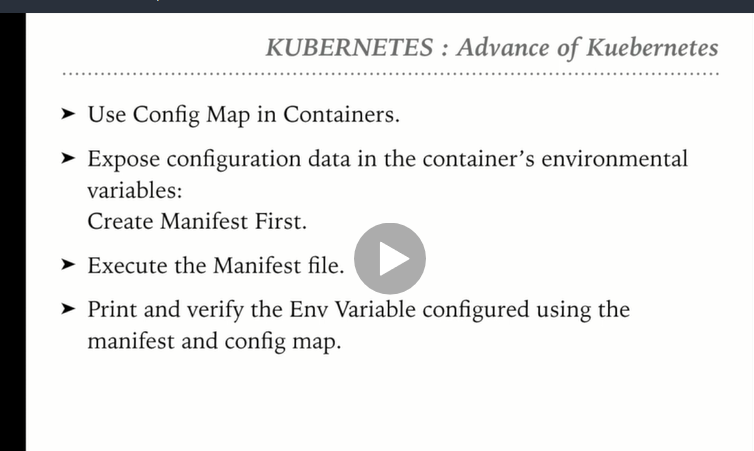
How to show the data inside config map:



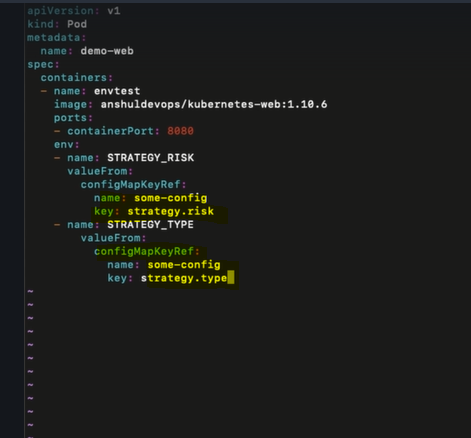
Creating by manifest file:



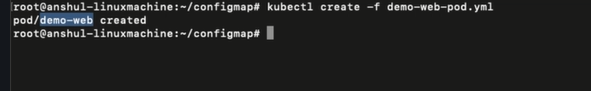




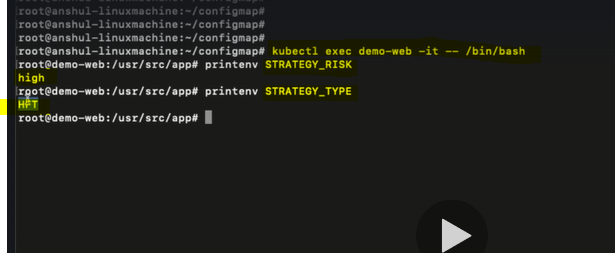
Create a manifest file which creates a pod and container and references the configMap which is already created.

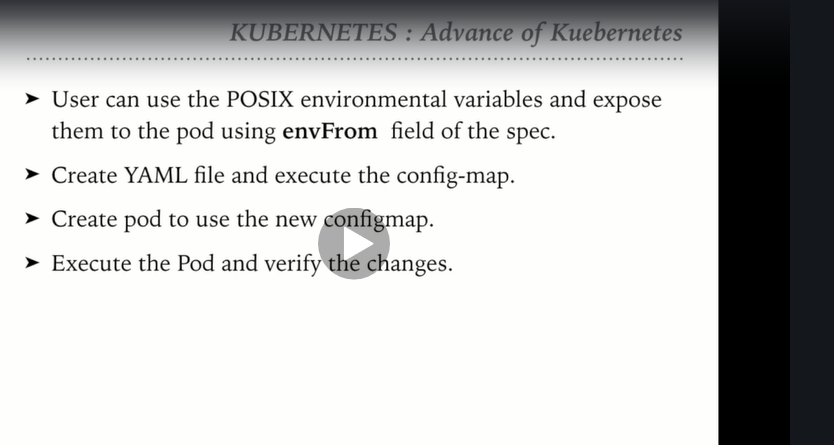


Create the pod with the above manifest file:



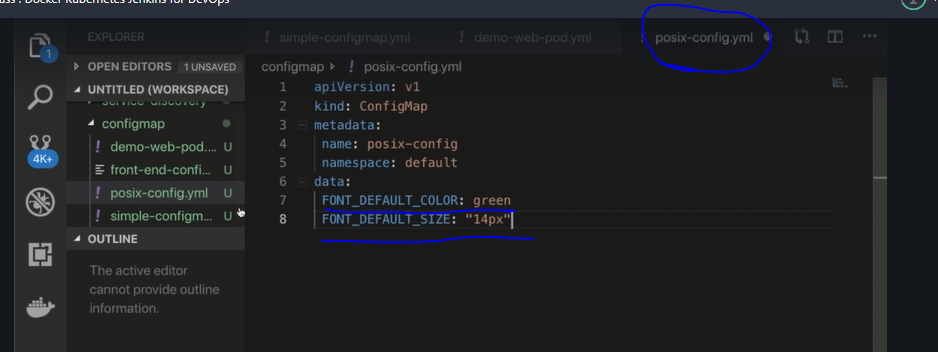
Open the container in the exec mode and verify if the config values are present:

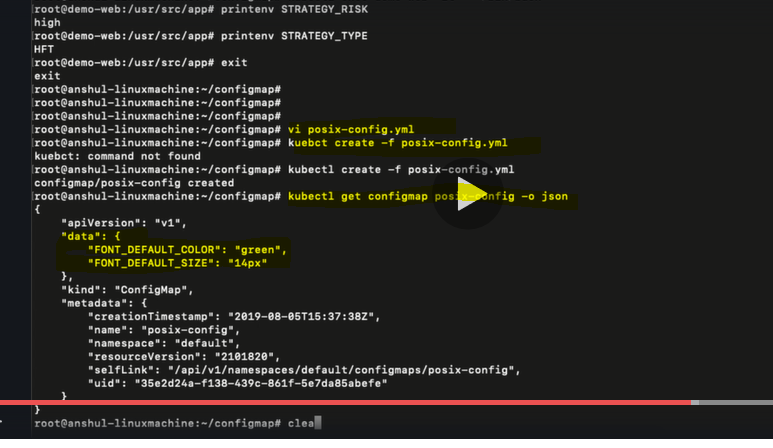




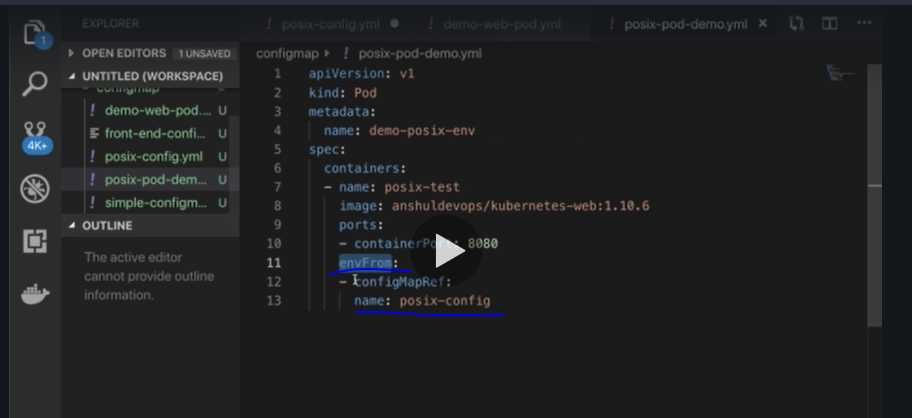
This is required when we have multiple env variables to use.

Create a config.yml file in POSIX format:

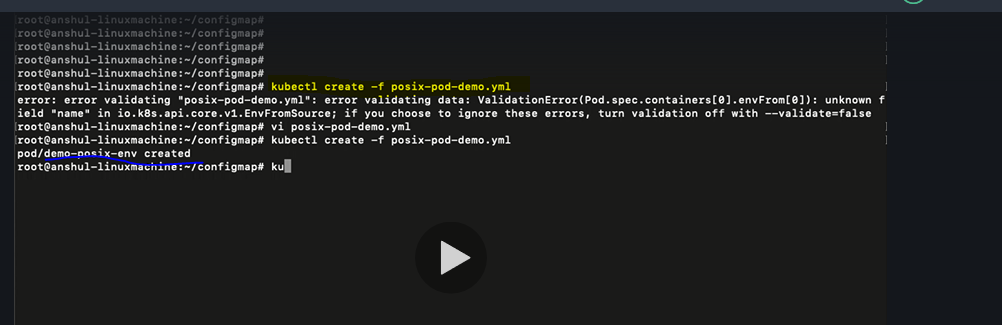




Create apod manifest file which references the posix-config.yml

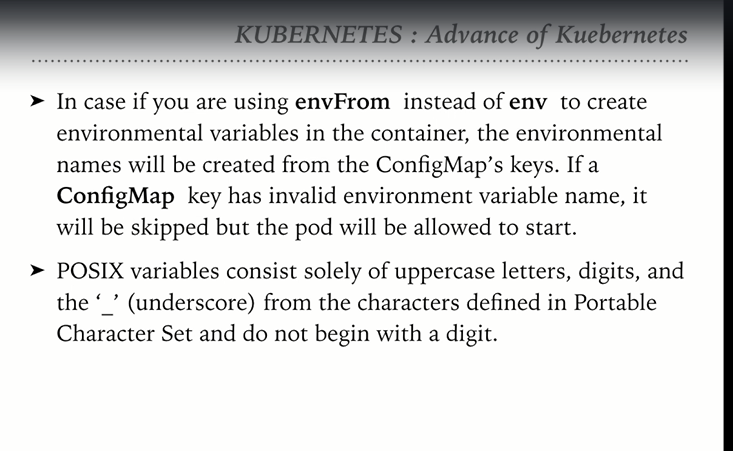


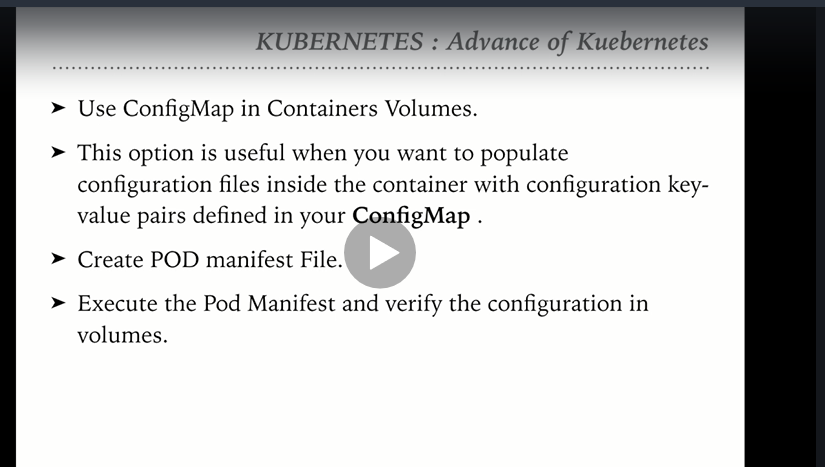
Create a pod with the above manifest

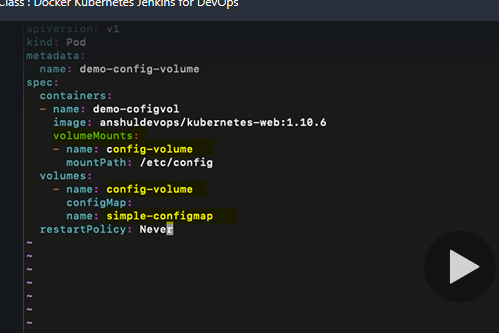


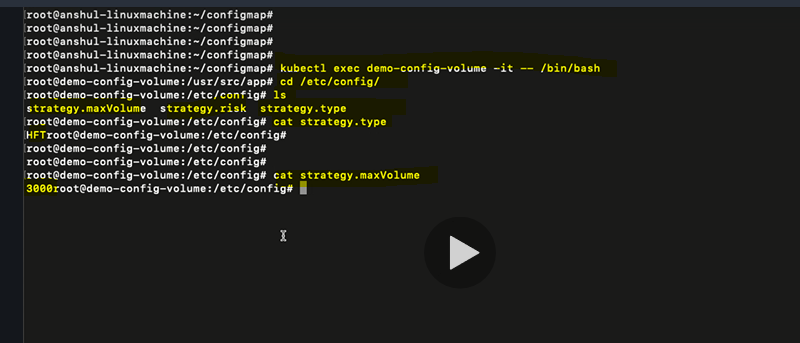
Get inside the container and verify that the env variables which are part of POSIX-configare present here:



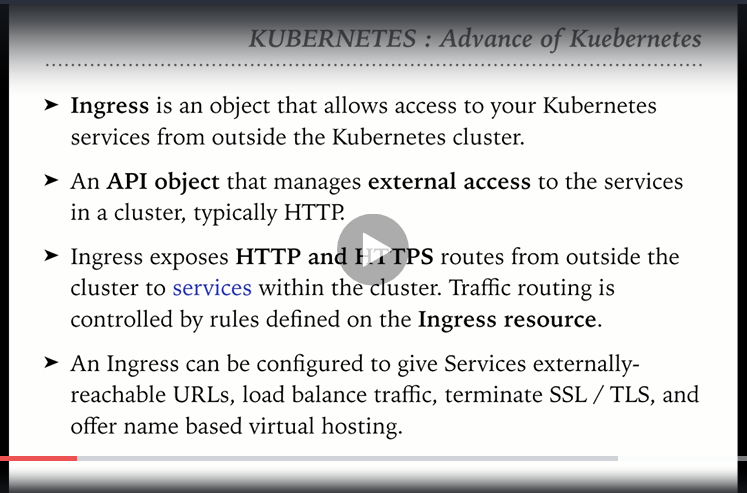


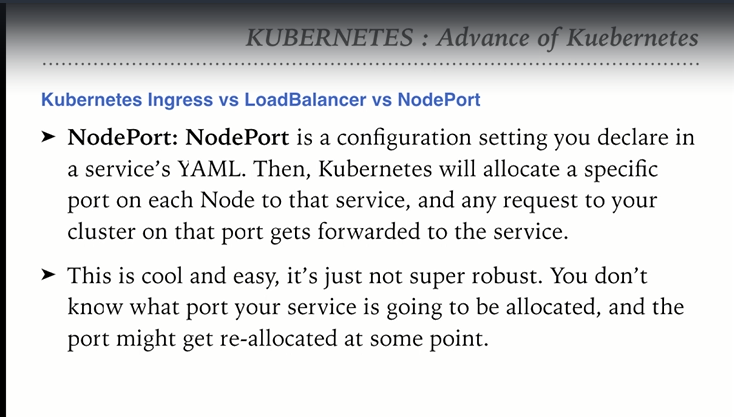


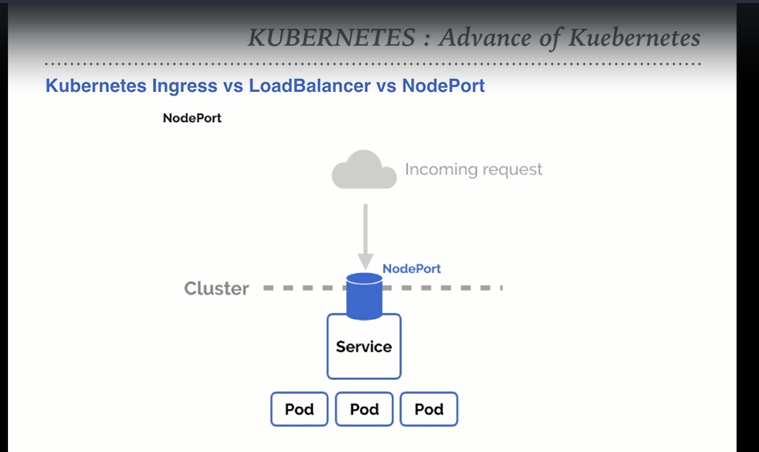


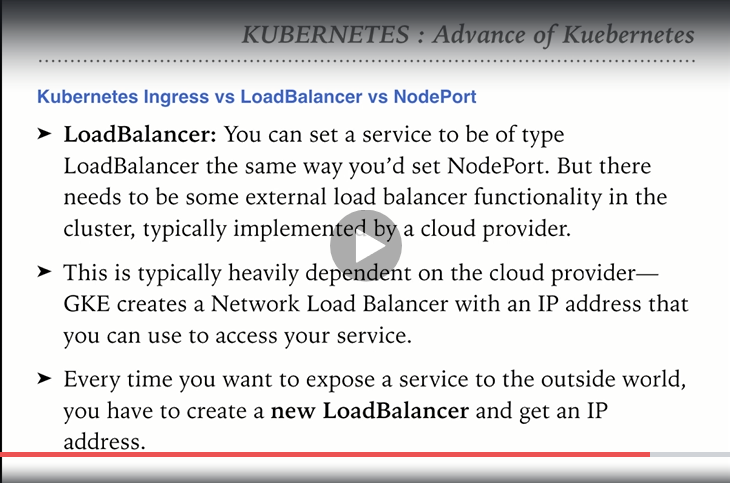


**Topic: Ingress**

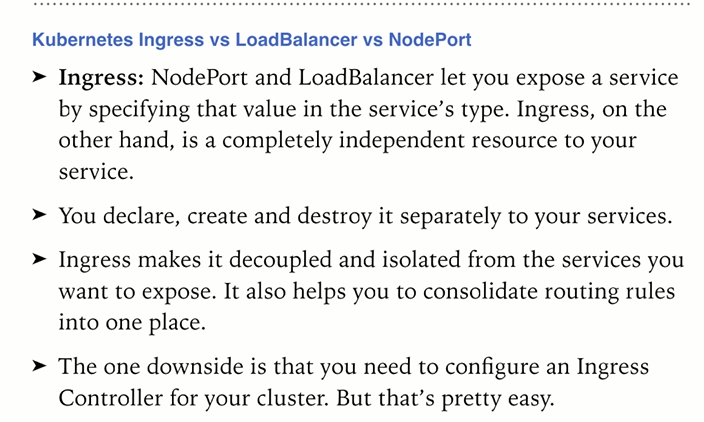


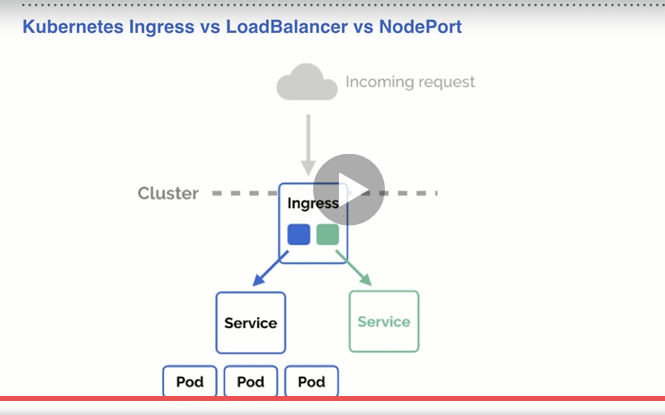


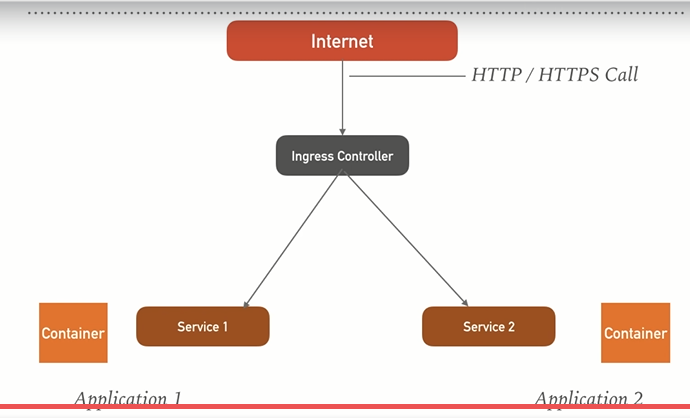


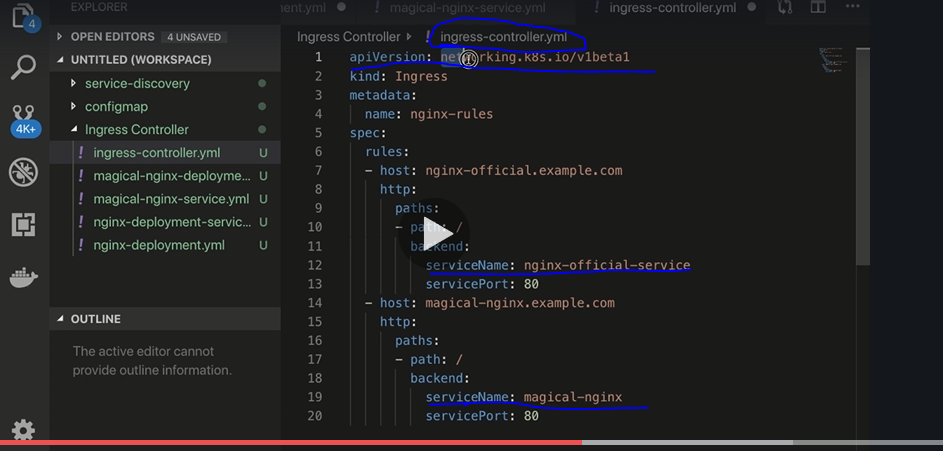


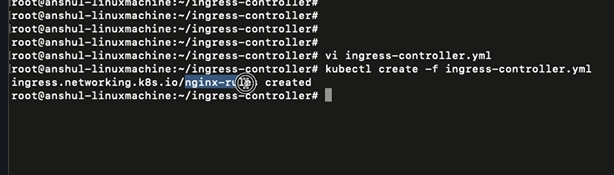


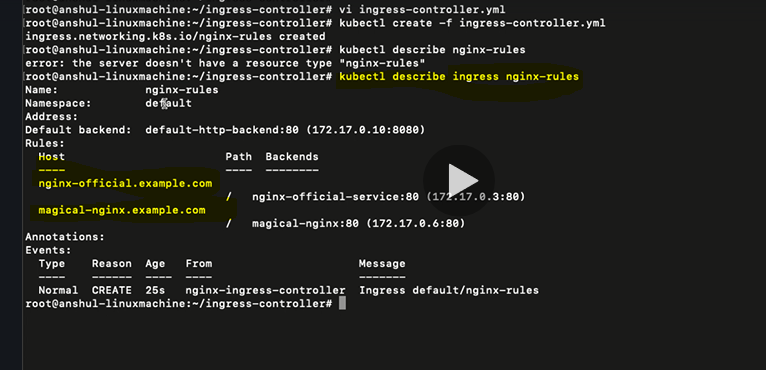






Following is the Ingress yml file, this is to specify the DNS for the services which we need to access.





How to access the services:

