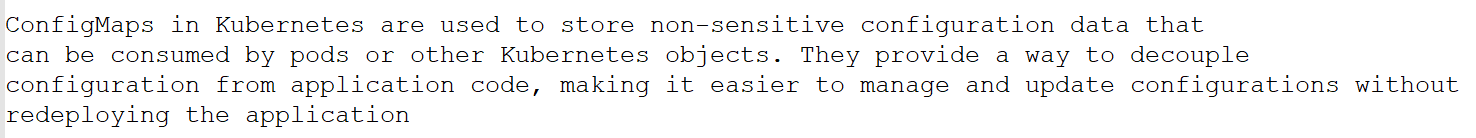
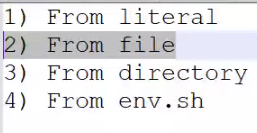
**Kubernetes-06**

**Config Maps:**

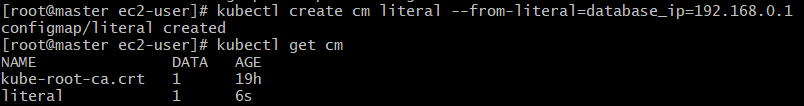
**We can create our configmap by different ways:**

****

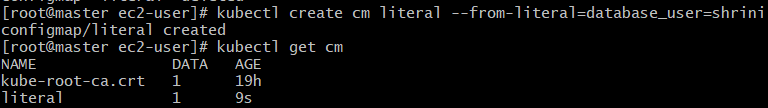
1. **How to create configmap using imperative command:**

**By using “--from-literal” we can save our data in configmap**

**Kubectl create cm <Config\_map\_name> --from-literal=database\_ip=192.168.0.1**

****

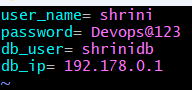
****

****

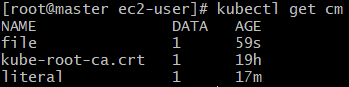
1. **Creating config-map by file:**

**First we are creating application-properties file for passing env- variables:**

**VI application. Properties:**

****

****

****

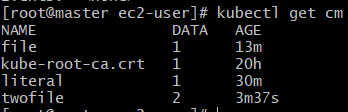
**But if we want to create two files in configmap then,**

**Vi test.properties:**

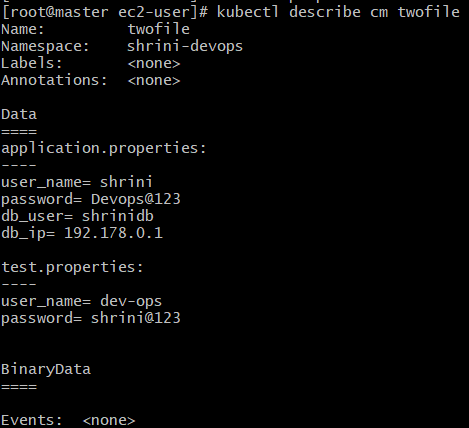
****

**Now we will use two properties in the yaml file:**

****

****

**And in above it is showing “data=2” then it means in a configmap we are using 2 properties and if we use10 properties then it will show “data=10”;**

****

**In the above pic we are using two properties in the configmap, by using “--from-file”;**

1. **Creating configmap by using --from-file but by using dir;**

**First we will try to create a dir and moves all the properties in that directory:**

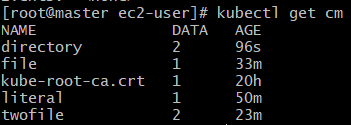
****

**Here, we are moving the properties file into the directory folder:**

**Then we are creating configmap with that folder:**

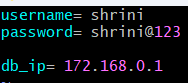
****

**Configmap is created with the two properties:**

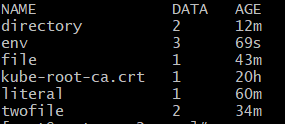
****

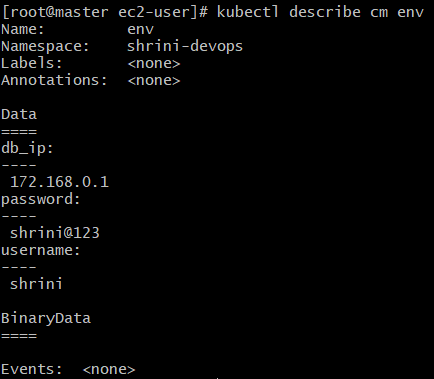
1. **Creating config-map by using env.sh:**

**Vi env.sh**

****

****

****

****

**So in the above pic we are seeing each data storing value and it will remove the spaces,**

**Differences:**

**Literal:**   
**If suppose we have 50 properties then it will be difficult to add them using --from-literal,  
on that case we have to use;**

**From file:**

**If suppose we have 50 files then it will be difficult to add them using --from-literal,--from-file,  
on that case we have to use;**

**From directory:**

**We can use multiple files with the help of directory and in dir we will mention all the files which we have all the properties**

**Environment variables:**

**Difference between --from-file and --from-env-file:**

**In env-file the data count will be number of variables we have passed and also it will not consider spaces and filename.** **In env file the variable name should not start with any numeric letter.**

**“AND”**

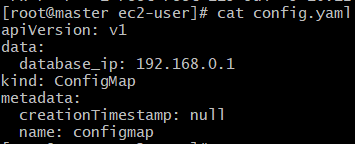
**In from-file we can see the data count as 1 only because it copy the complete content of the file.**

**ConfigMap from YAML:**

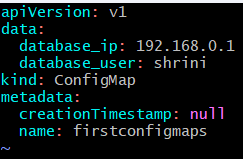
**Quickly write any file by using -o yaml.**

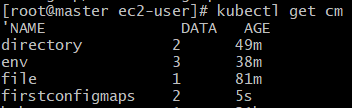
**[kubectl create cm configmap --from- literal=database\_ip=192.168.0.1 --dry-run=client -o yaml > config.yaml ]**

**It will create a config. Yaml file and then we can modify as we want:**

****

**Now I will modify by adding one more data and also changing the name;**

****

****

**So for best practices to create configmaps we can use by “ –o yaml”:**

**kubectl create cm configmap --from-literal=database\_ip=192.168.0.1 --dry-run=client -o yaml > config.yaml**

**kubectl create cm configmap --from-file=application.properties --dry-run=client -o yaml > config.yaml**

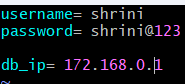
**kubectl create cm configmap --from-file=properties --dry-run=client -o yaml > config.yaml**

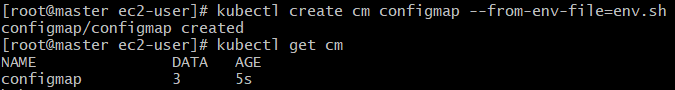
**kubectl create cm configmap --from-env-file=env.sh --dry-run=client -o yaml > config.yaml**

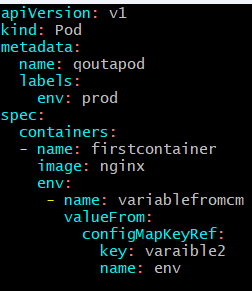
**Inject config-maps inside pods:**

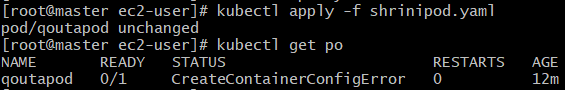
* **Inject as ENV**
* **Inject as file**

**Inject as ENV:**

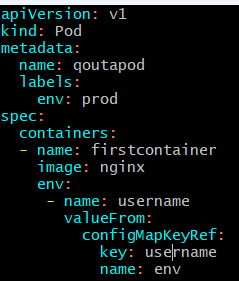
**VI env.sh:  
**

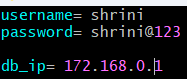
**Creating one configmap by env.sh file:**

**Creating pod by giving the env variable as “key” :- **

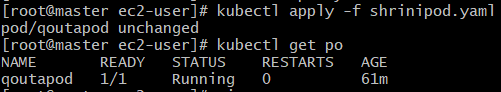
**And when we try to create pod by using configmap then,  
**

**It gives error because it is not correctly mapped with the configmap and there is no such variable:**

****

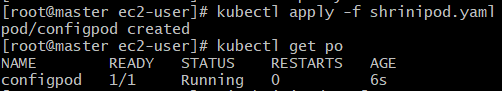
**“And”  
In env.sh file values should be matched**

**Now if we try to create pod it will get created by env.sh file;**

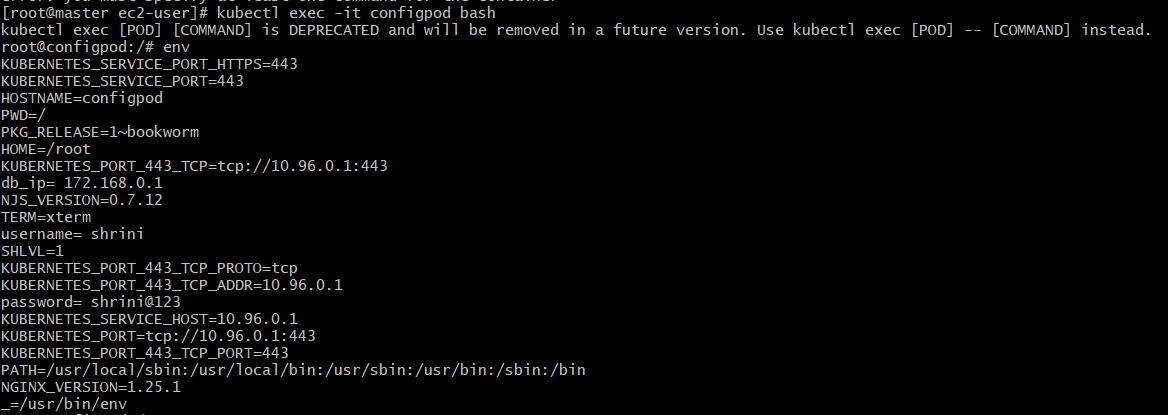
****

**So it means,**

**[env.sh=pod/env/configmapkeyref/key] :- should be same  
How to inject all values of cm into pod?  
adding the  in pod yaml file.**

****

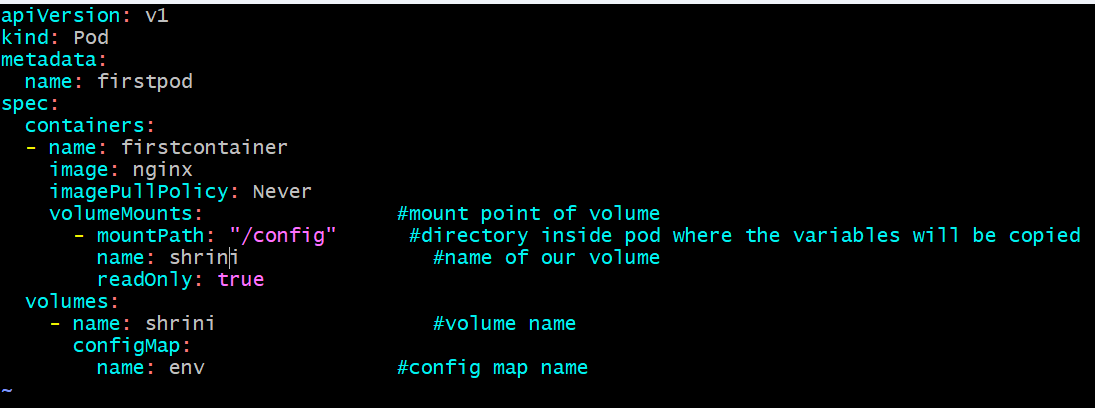
**And pod also gets created then if try to execute into the pod:**

**Our env variable we can saw there.**

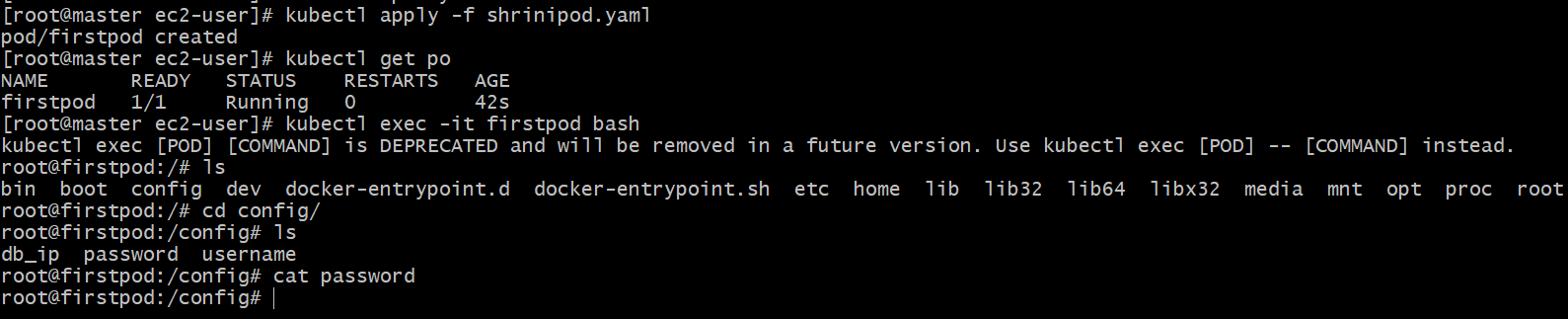
**Inject config maps as files:**

**If suppose we need to copy the variables inside a pod as a file then we can use config maps as files.  
This to be copy in terms of volumes.**

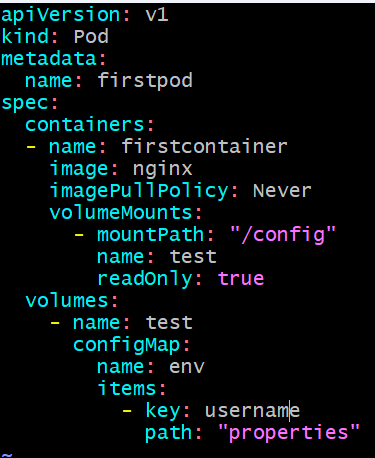
**Vi shrinipod.yaml**

****

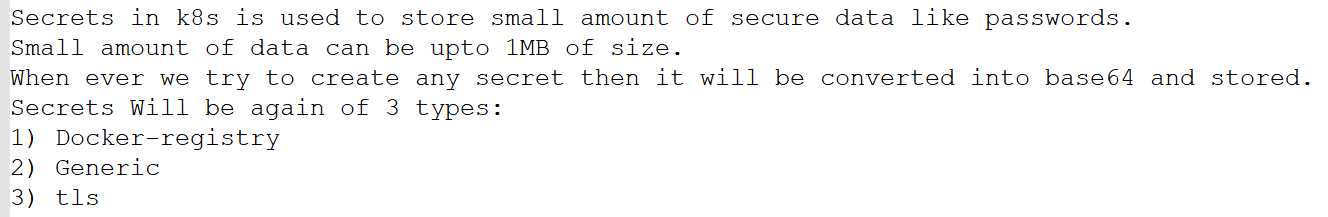
**[In the above pic we are mentioning in the containers “volumeMounts” so by which when container get created it deploy our configmap content to the “mounthpath” location also it taking from the volume of “existing configmap” name:env ]**

****

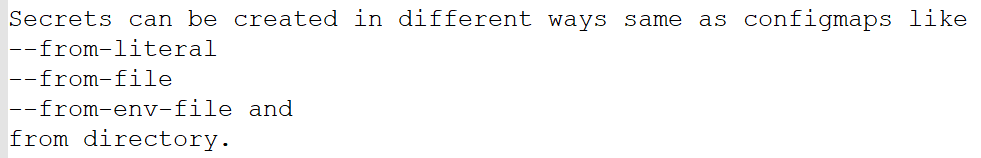
**If you want to inject specific variable as files in pod:**

****

**SECRETS:**

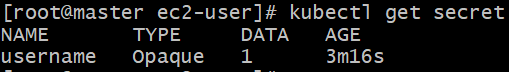
****

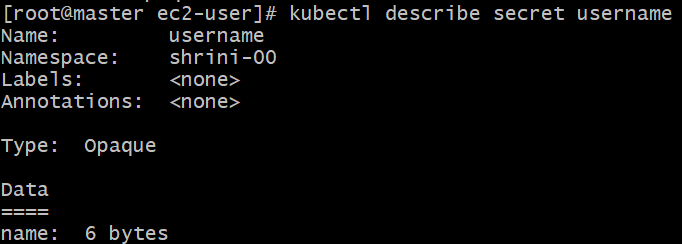
**Most commonly used will be generic.**

****

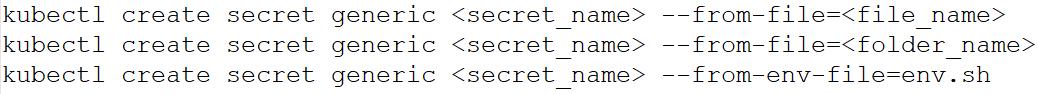
**How to create a secret ?**

**From literal:**

****

****

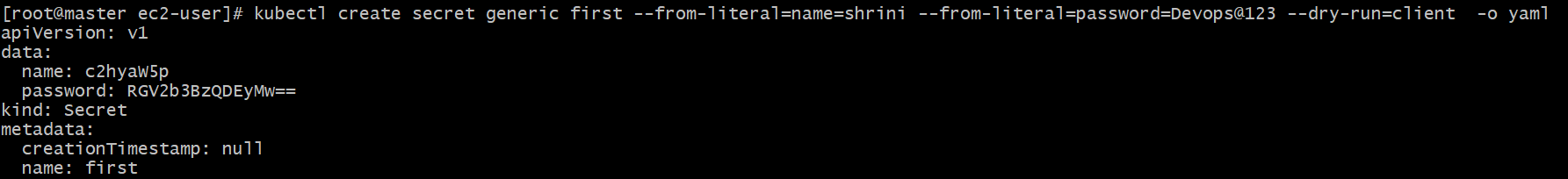
**kubectl get secret <secret\_name> -o yaml --> To see a basic yaml file related to secret.**

****

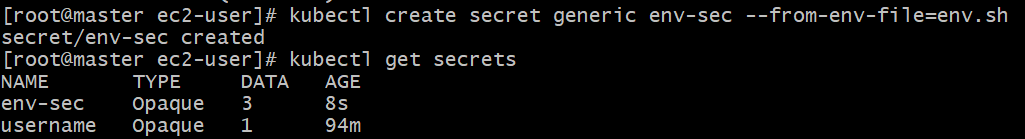
**How to get the encoded value ?**

****

**How to create secrete using yaml by using --from-literal?**

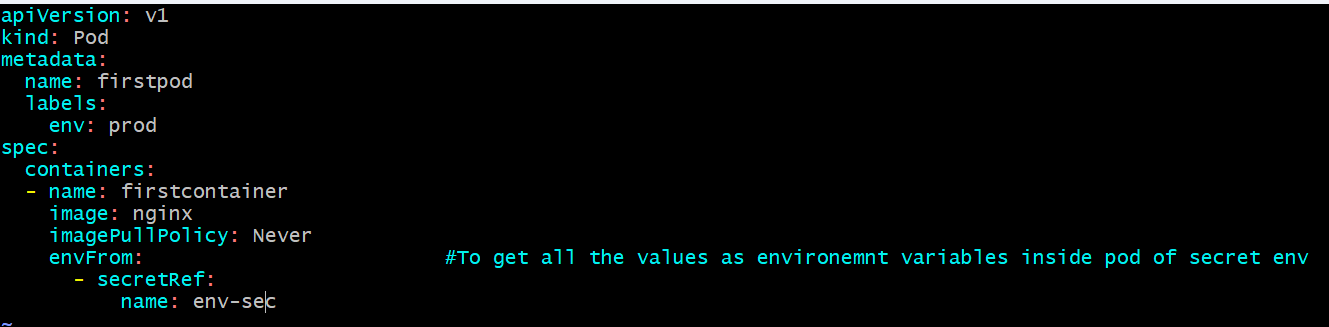
****

**How to create secrete using yaml by using --from-env-file?**

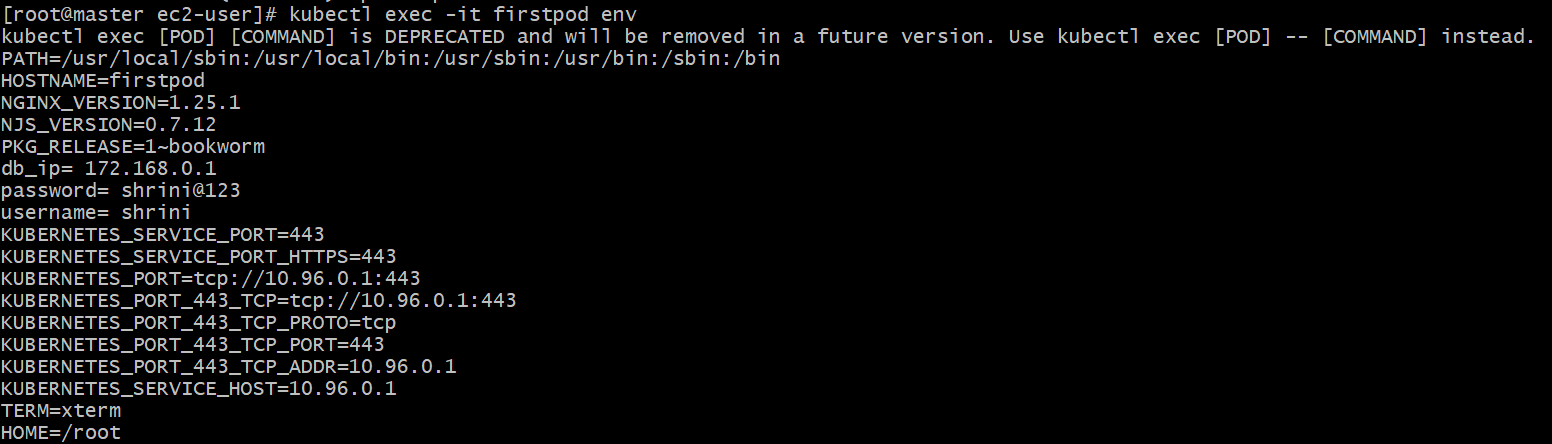
****

**1. How to inject all values of secrets into pod by env.sh?**

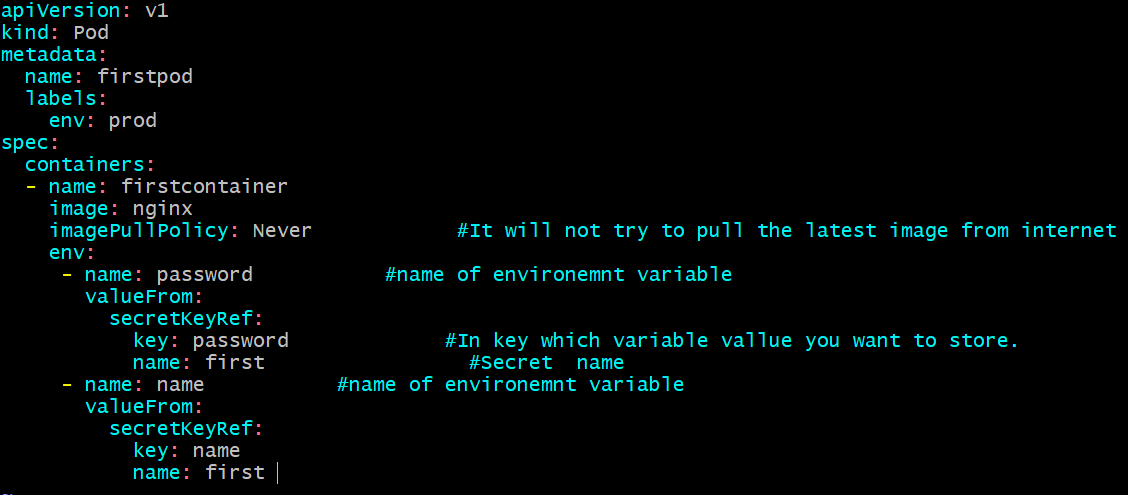
**Vi shrinipod.yaml**

****

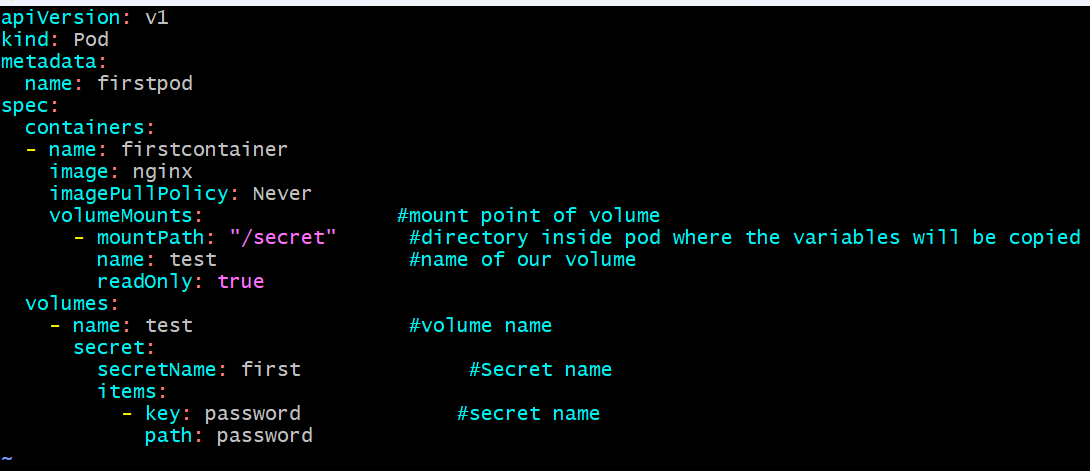
**And checking the values of secrets in the pods:**

****

**2. If we want to inject multiple values?**

****

**3. If you want to inject specific variable as files in pod:**

****

**---------------------------------SHRINIVAS-------------------------------------**