**Secrets**

Imagine that we are deploying some containerized apps, configuration of this app contains some sensitive data such as usernames and passwords. Because of this sensitive nature, it is not recommended to use as plain text format in manifests files.

How do you manage **sensitive data in kubernetes**?

**Secrets**



**What is meant by secrets?**

**How do we use in kubernetes?**

Objectives:



**Secret:**

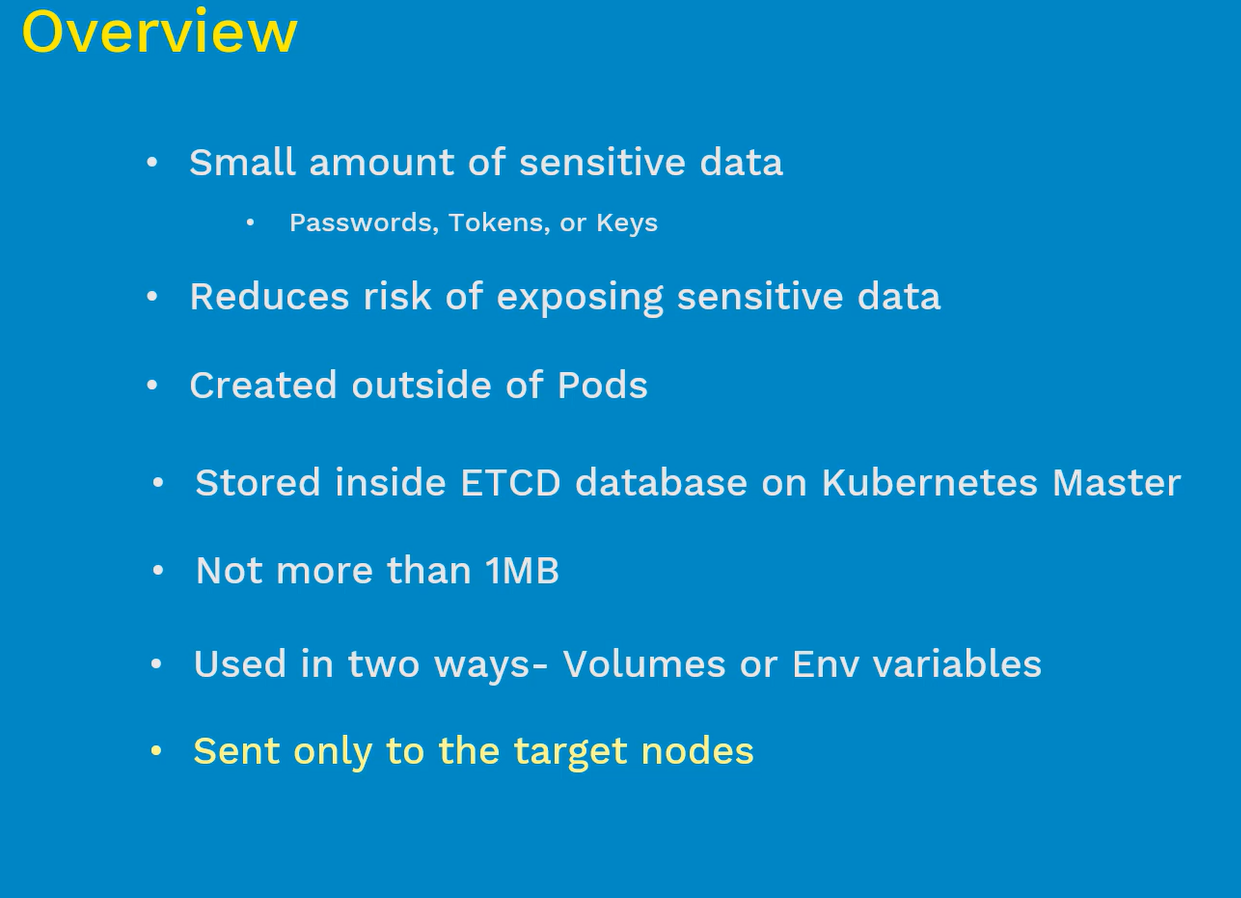
Kubernetes object used to handle small amount of sensitive data by using secrets which include passwords, token or key

This is one solution, kubernetes use the passwords or keys in the pod manifest file. The main of the secrets is to reduce the risk of accidental exposure of confidential information. Secrets are very high level concept on kubernetes cluster

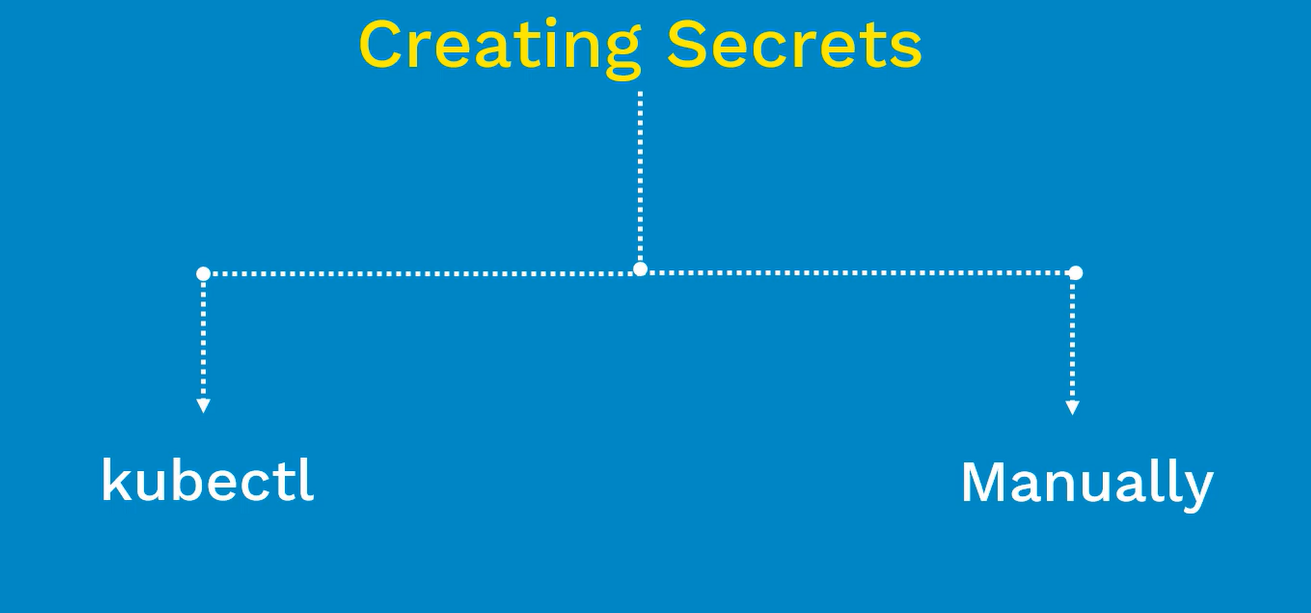
* Maximum size of the secret is 1MB
* Secrets are stored in the etcd database on kubernetes master
* Secrets are created outside the pods and we can use as many times on the pods
* Used in two ways – Volumes or Environment variables

(we also have a similar kind of secrets concept on CMF tools like salt, ansible, puppet. These CMF tools will distribute secrets all nodes without respect the node requirement)

* Each secret is stored in tmpfs volume, so that it will restrict access application in the node( application run in tmpfs and container run in different format, this is very selective to pods and container)



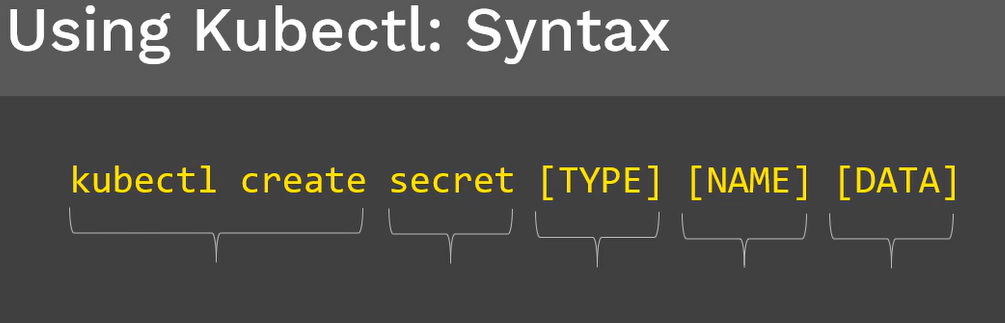
# Creating Secrets:



## Kubectl:

Once we can create secrets, we can use in one or more pods

Syntax:



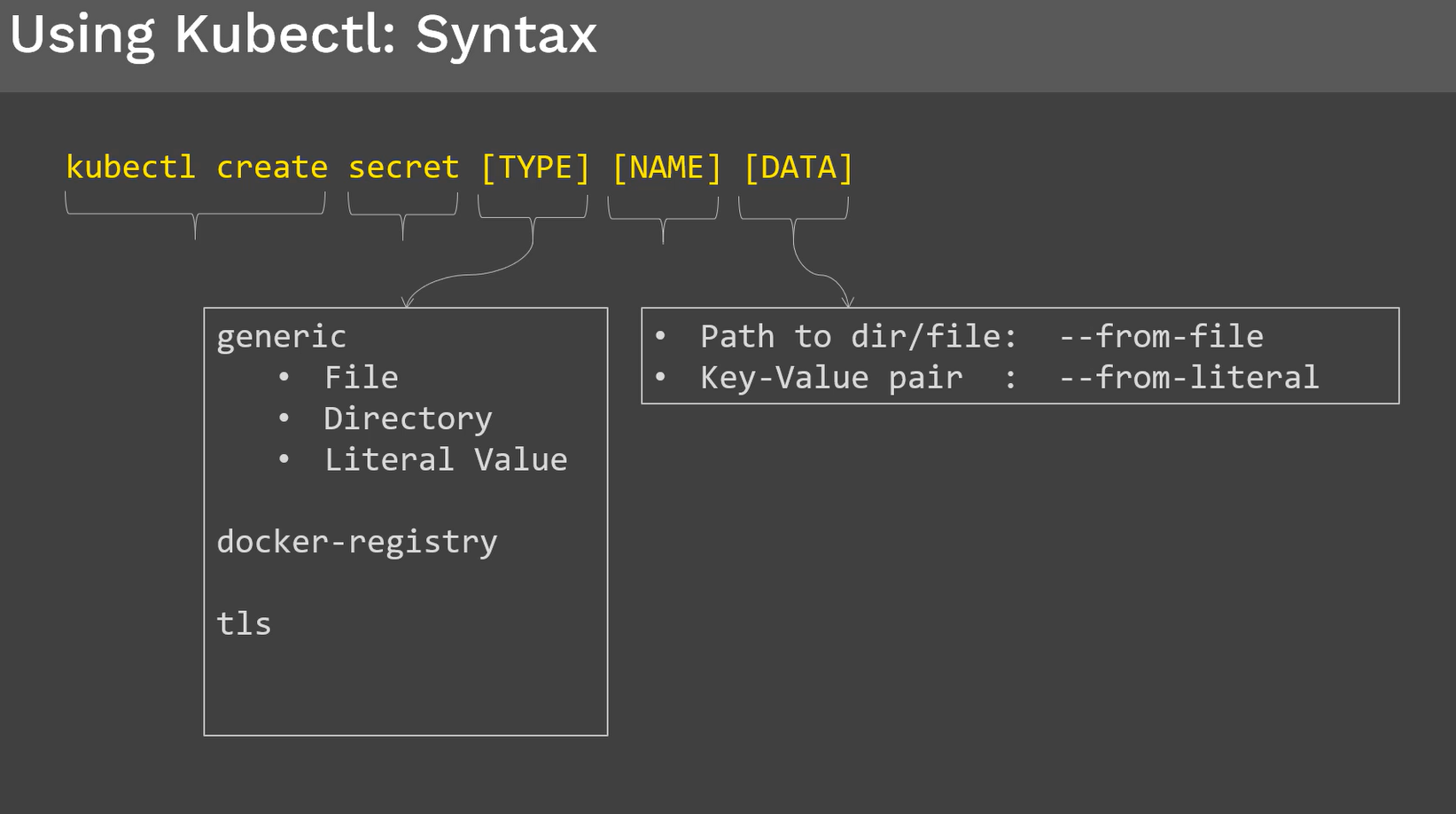
Kubectl create follows that **object(**secret) followed by **TYPE**

Type can be one of the following:

* Generic

1. File
2. Directory
3. Literal Value

* Docker-registry
* Tls

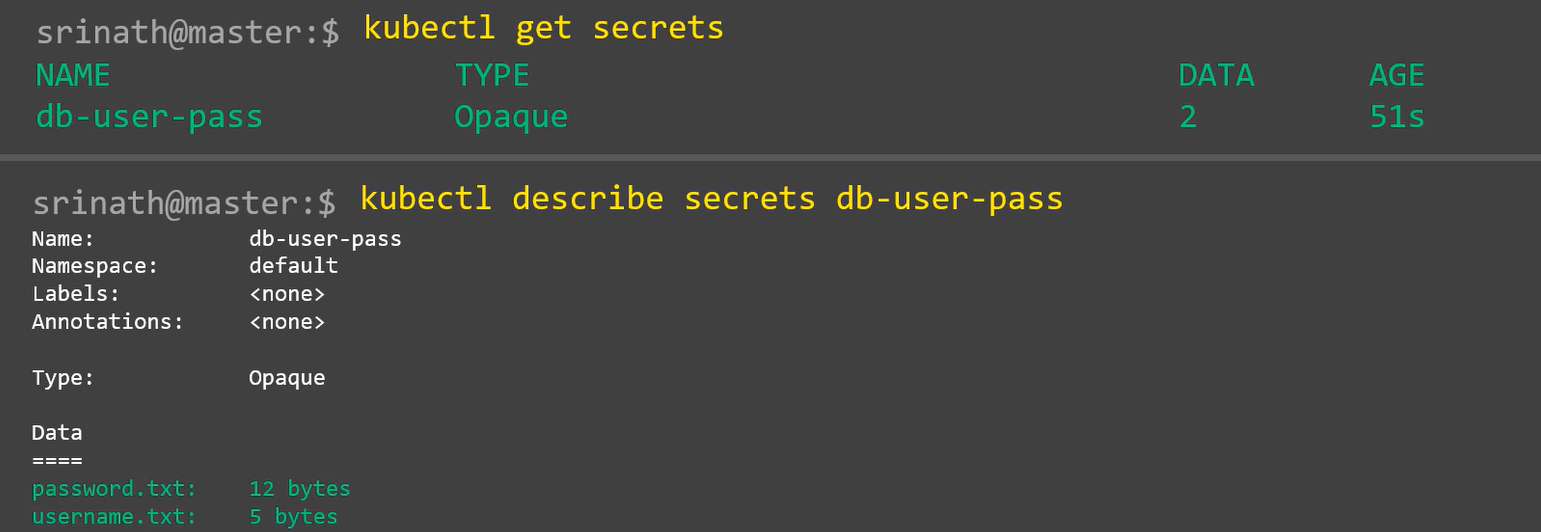


**Creating secret using kubectl:**

Using kubectl

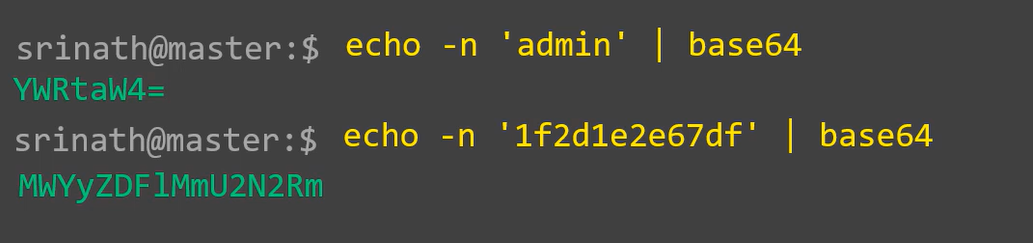


Get and Describe

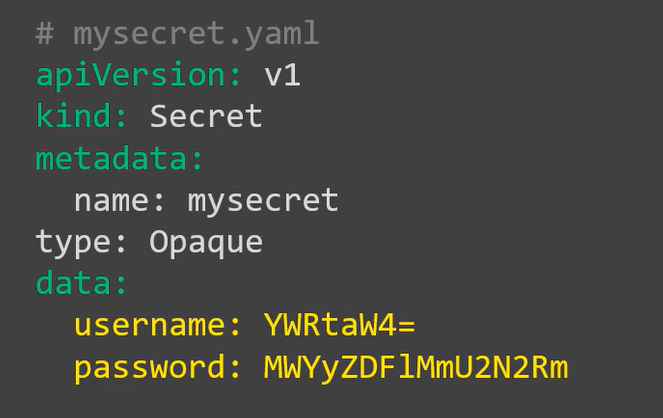


## Manually:

To create the secrets manually, first we need to encrypt the data. Generally we encrypt the data by using base64

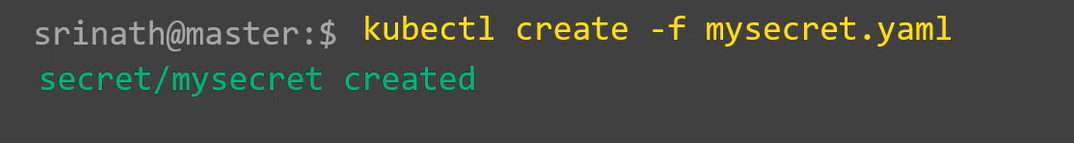


We encrypt the username and passwords by using above commands by base64. Once the data is encrypt we use those values and create manifest file



First provide the apiVersion and **kind** is Secret and come to **data** section: write username and password

Data field is map which consists of key-value pair. Username and password is keys and encoded data is named as Values. We are manually enter the base64 encoded values in the Data field. Once we have spec file is ready, then we have to create secret manifest file using kubectl create command



Kubectl get and describe to display the secret file that we create.

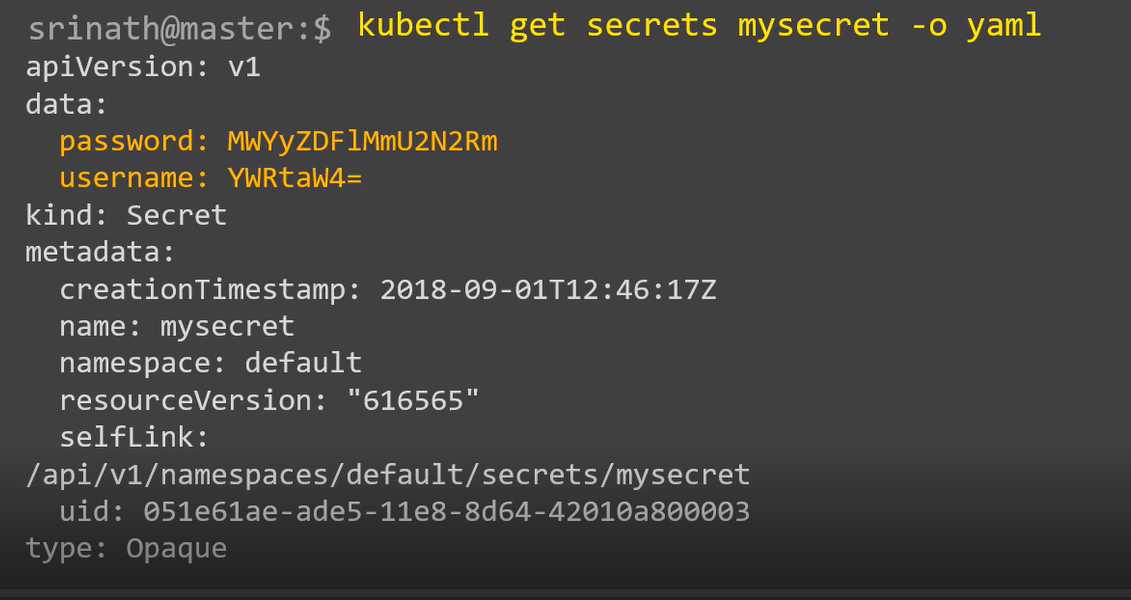
Kubectl get secret mysecret

Kubectl describe secret mysecret –o yaml

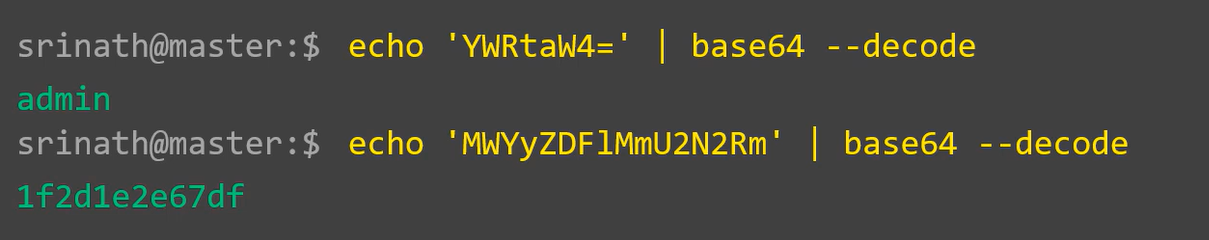
# Decoding Secrets:

We use same base64 method to decode secrets, let’s display the secrets that we create recently and formatted the output into YAML file

Command: kubectl get secrets mysecret –o yaml



Check the data field it has the username and password in encoded format, we can use the base64 to decode the values



# Consuming Secrets in Pods:

* Volumes
* Environment Variables

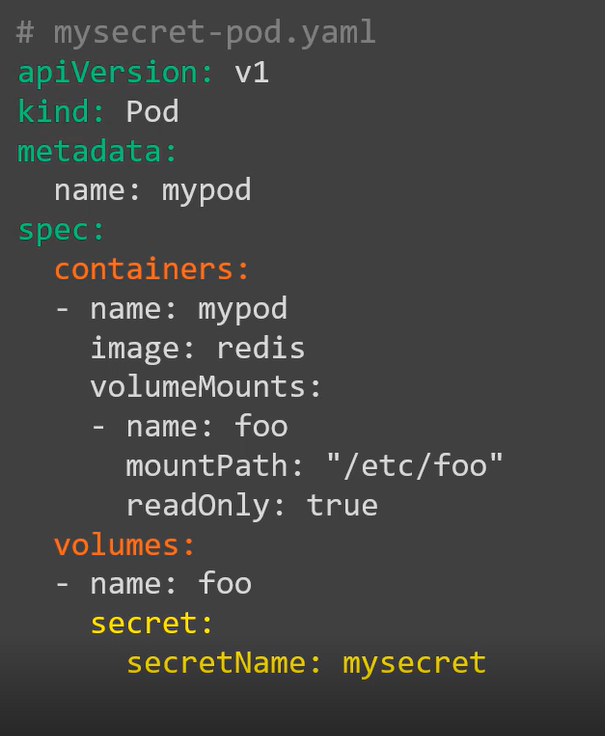


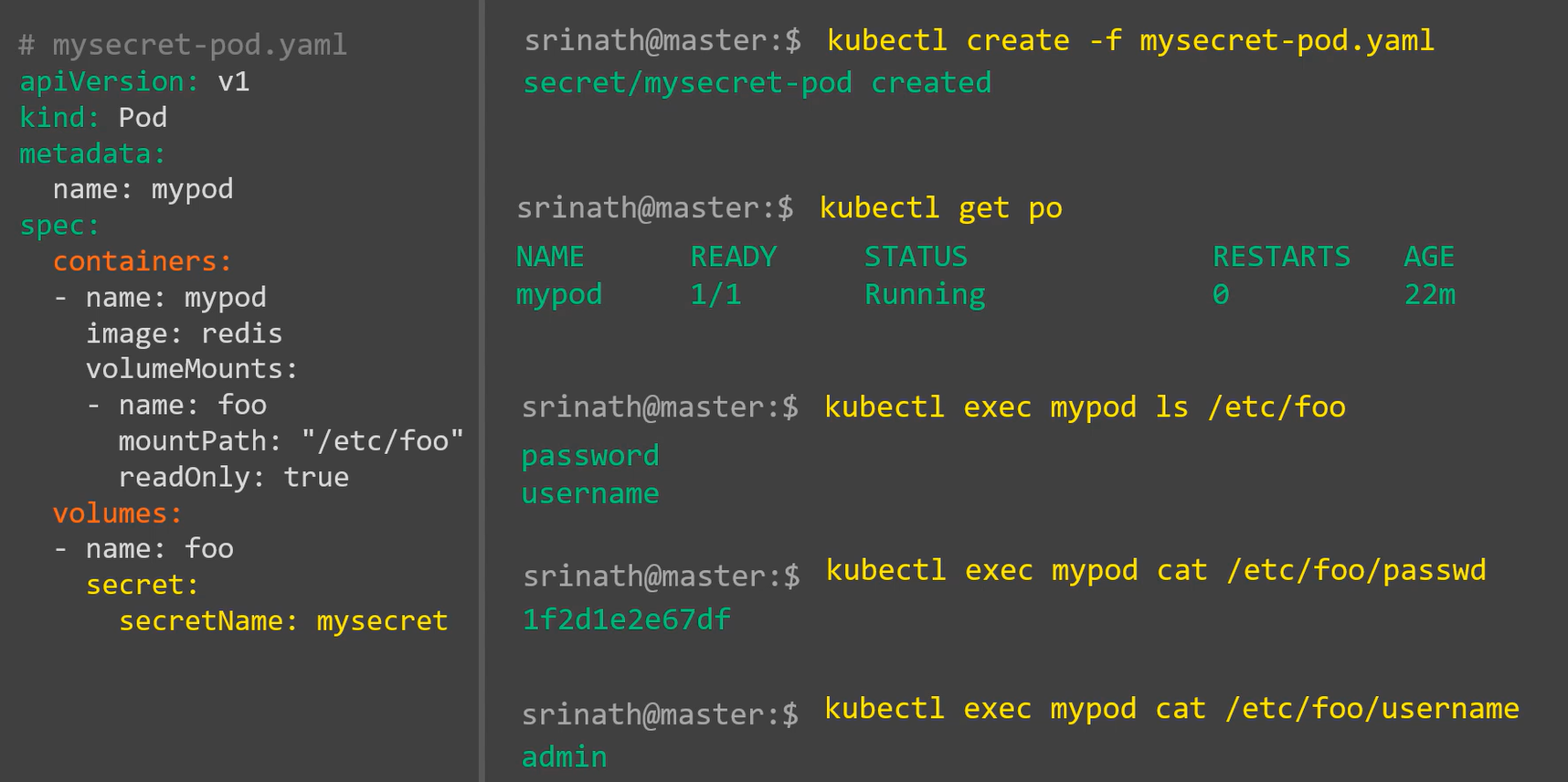
## Consuming secrets from “volume”

First create secret manually



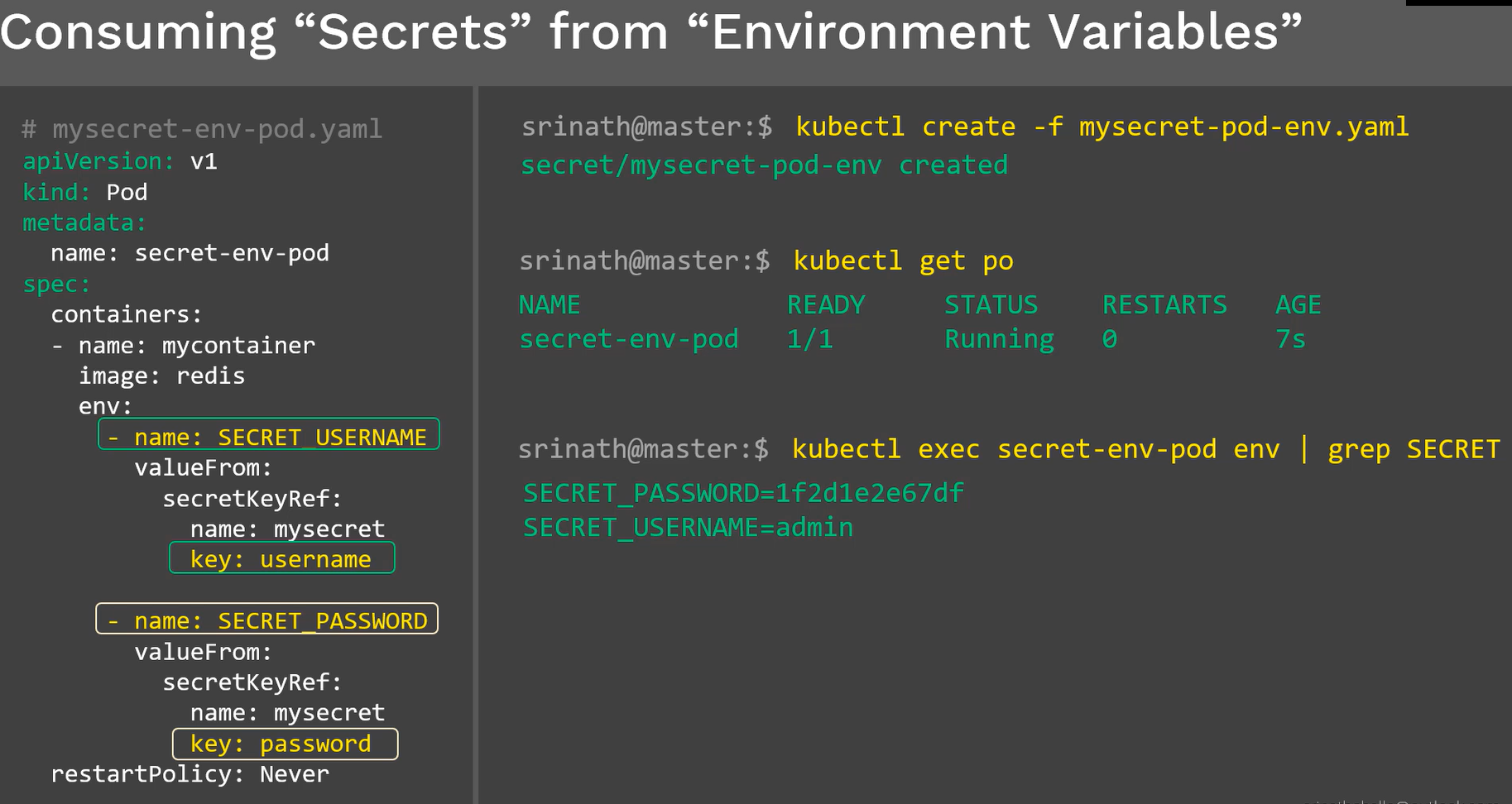
**Pod manifest file:**





## Using Environment Variables:





**Note: Refer demo file: secrets-demo.txt file**