1. <u>Launching the DB mySQL via yaml code in KUBERNETES.</u>

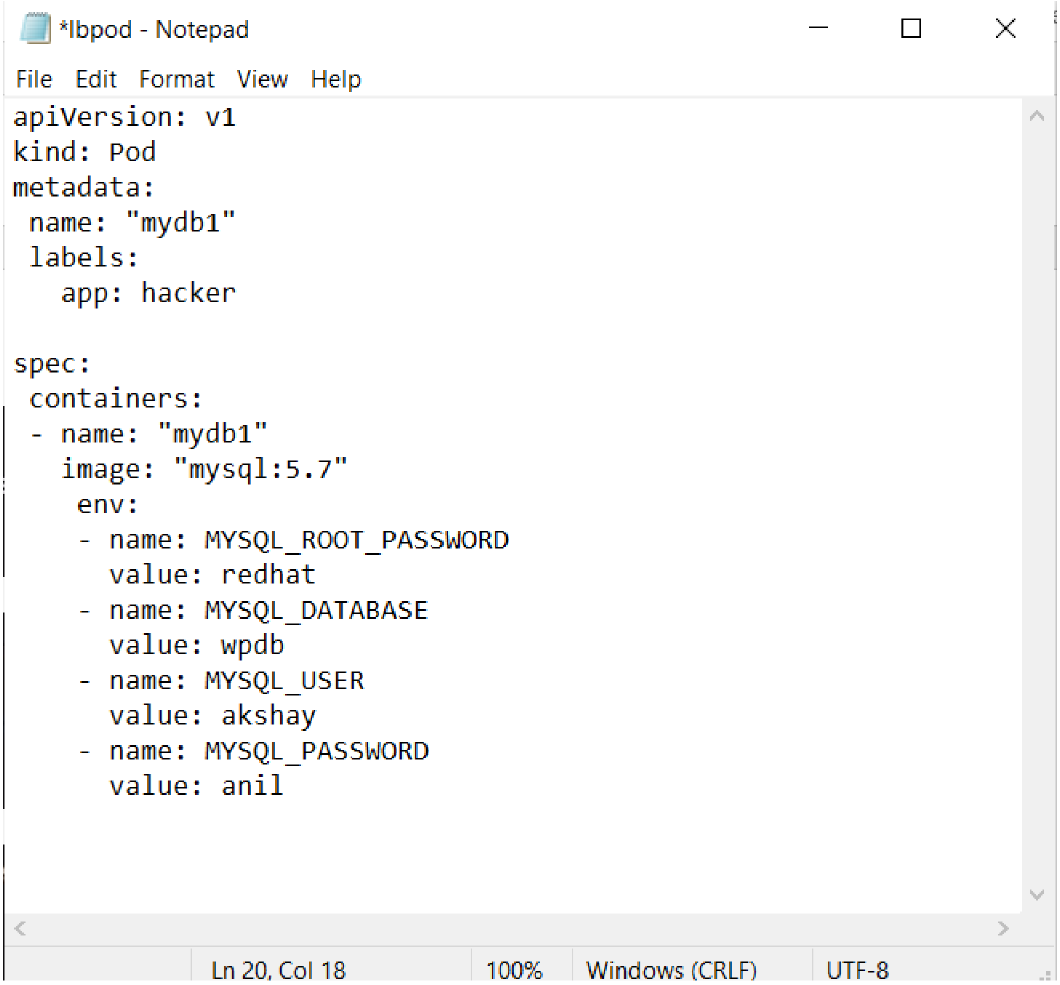
2. <u>SECRET SERVICE OF K8-</u>

   Encoding the confidential Info of DB, like root_passwrod and username, as to use mysql image we have pass four info (MYSQL_ROOT_PASSWORD, MYSQL_DATABASE, MYSQL_USER, MYSQL_PASSWORD) as Environment Variables.

3. <u>Extracting the encoded password via CLI –</u>
   K8's master server only understands YAML, so pod launched via CLI even first get converted to YAML then passed in to K8 for executing.
   So, exposing this file will get me the encoded password too and can decode it by base64 conversion.

4. <u>Create a secret key via CLI</u>
   Exploring the help command to get the usage of SECERET in K8 via CLI.

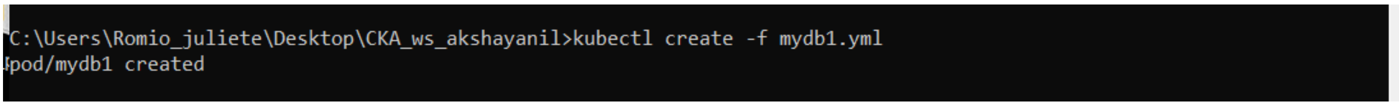
1st : Code to launch the mysql image:



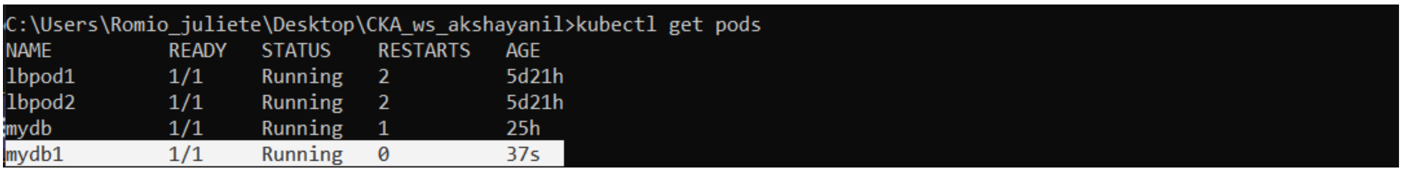
```
apiVersion: v1
kind: Pod
metadata:
 name: "mydb1"
 labels:
   app: hacker

spec:
 containers:
 - name: "mydb1"
   image: "mysql:5.7"
    env:
    - name: MYSQL_ROOT_PASSWORD
      value: redhat
    - name: MYSQL_DATABASE
      value: wpdb
    - name: MYSQL_USER
      value: akshay
    - name: MYSQL_PASSWORD
      value: anil
```

kubectl create -f mydbl1.yml

kubectl get pods

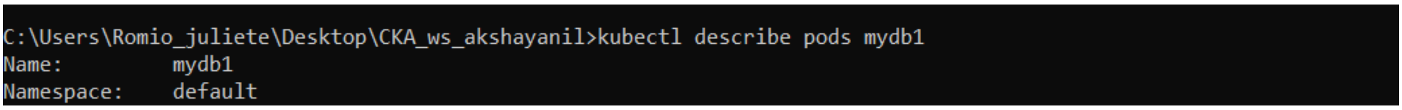


```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl create -f mydb1.yml
pod/mydb1 created
```

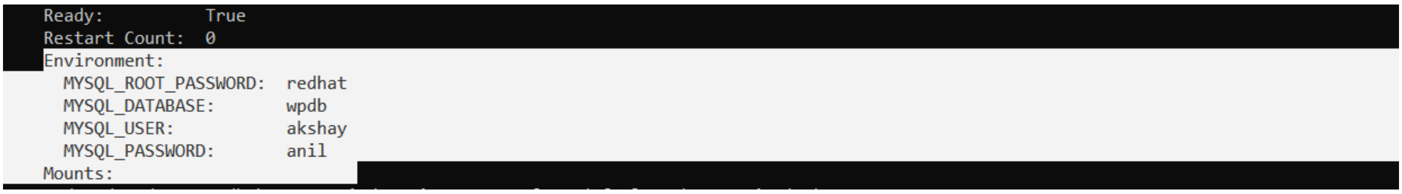


```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
lbpod1       1/1     Running   2          5d21h
lbpod2       1/1     Running   2          5d21h
mydb         1/1     Running   1          25h
mydb1        1/1     Running   0          37s
```

Environment variables - kubectl describe pods mydb1



```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl describe pods mydb1
Name:         mydb1
Namespace:    default
```



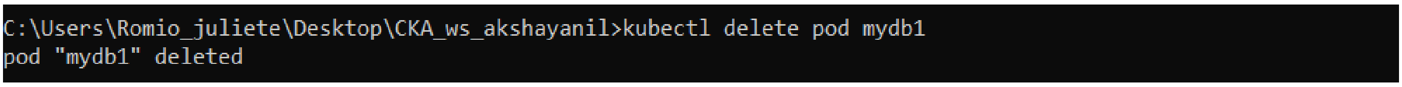
```
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  redhat
      MYSQL_DATABASE:       wpdb
      MYSQL_USER:           akshay
      MYSQL_PASSWORD:       anil
    Mounts:
```

Successfully launched via code….

Another challenge: encode the password

K8 provides a service of SECRET. u have to write one external yml file for this.

1st: deleting the pod



```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl delete pod mydb1
pod "mydb1" deleted
```

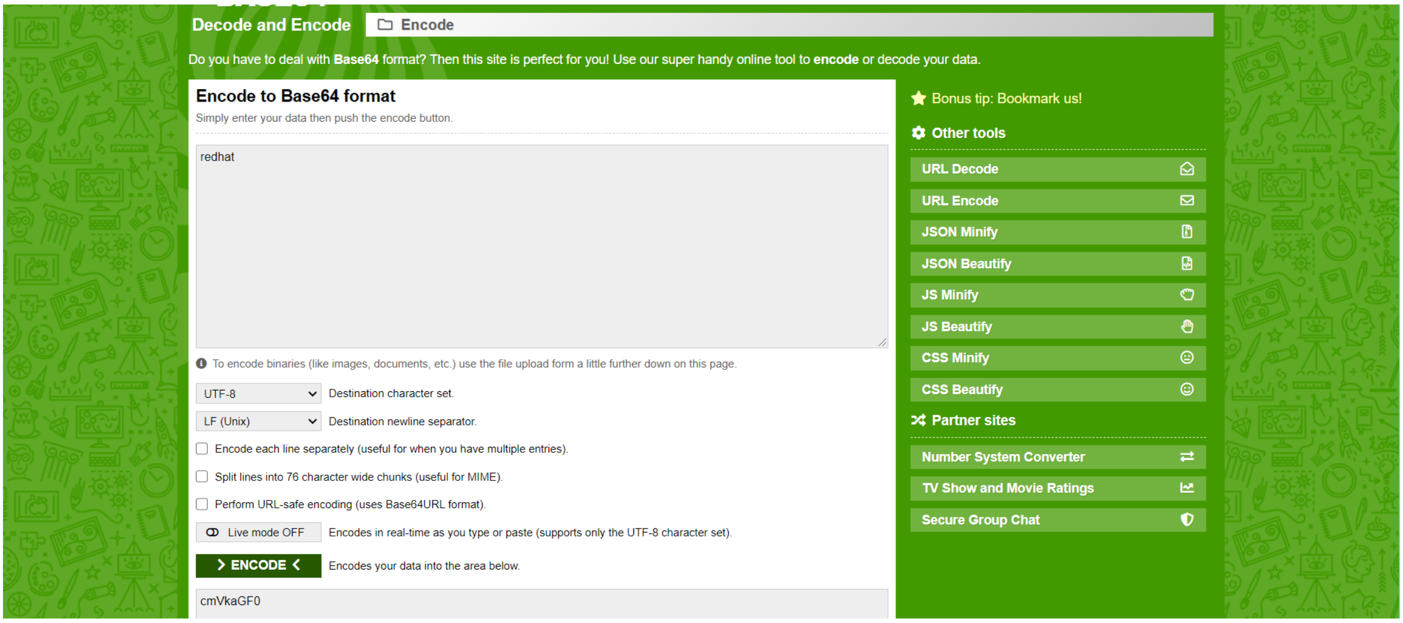
2nd: using base64 encoding to encode the password.

base64 encode

base64 - Google Search

base64 **decode**

base64 **encode**



Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to **encode** or decode your data.

Encode to Base64 format

Simply enter your data then push the encode button.

```
redhat
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 — Destination character set.

LF (Unix) — Destination newline separator.

□ Encode each line separately (useful for when you have multiple entries).

□ Split lines into 76 character wide chunks (useful for MIME).

□ Perform URL-safe encoding (uses Base64URL format).

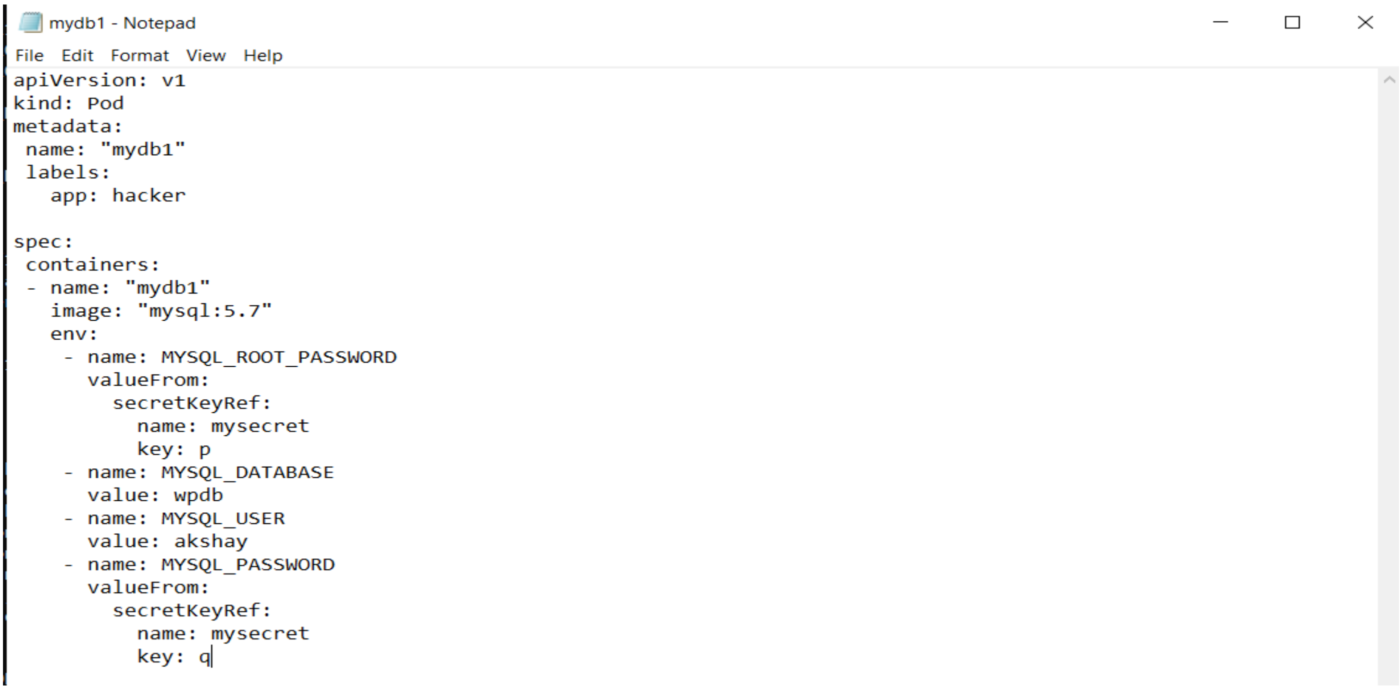
Live mode OFF — Encodes in real-time as you type or paste (supports only the UTF-8 character set).

> ENCODE < — Encodes your data into the area below.

```
cmVkaGF0
```

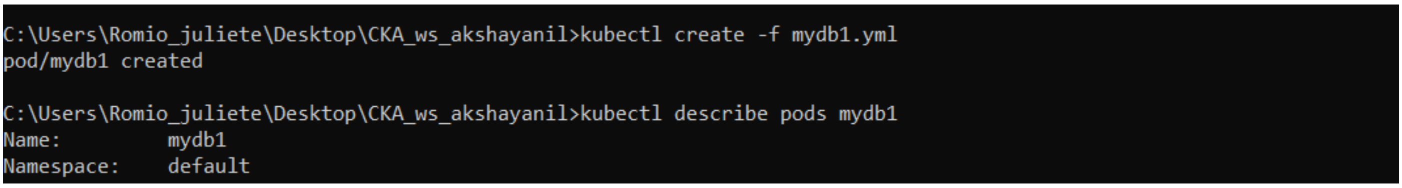
redhat - > base64 encode -> cmVkaGF0

3rd write the code for mydb and for secret key



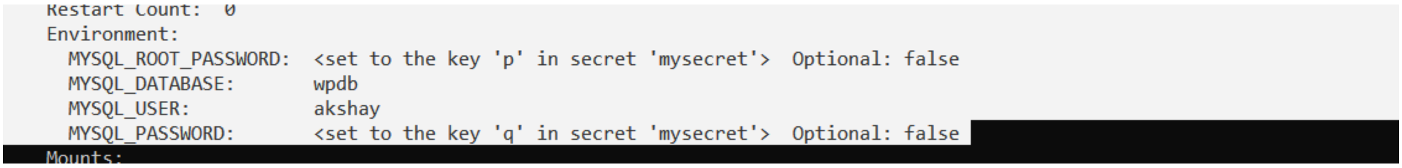
```
apiVersion: v1
kind: Pod
metadata:
 name: "mydb1"
 labels:
   app: hacker

spec:
 containers:
 - name: "mydb1"
   image: "mysql:5.7"
   env:
    - name: MYSQL_ROOT_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mysecret
          key: p
    - name: MYSQL_DATABASE
      value: wpdb
    - name: MYSQL_USER
      value: akshay
    - name: MYSQL_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mysecret
          key: q
```

create mydb1: kubectl create -f mydb1.yml



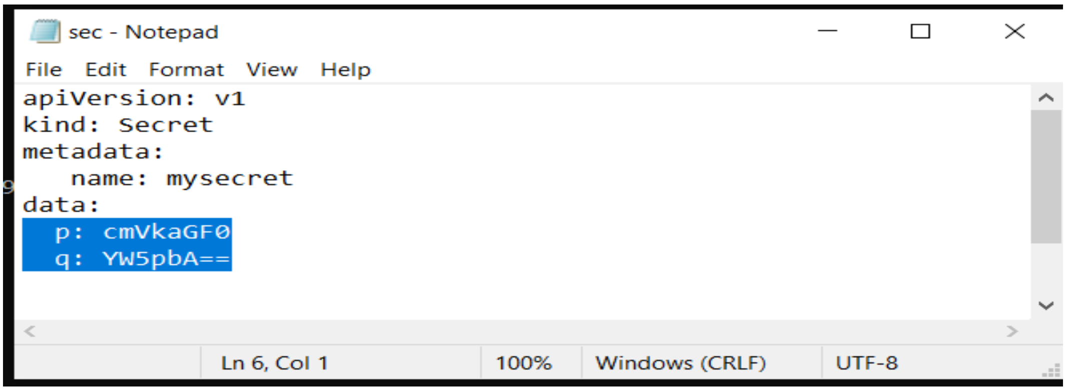
```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl create -f mydb1.yml
pod/mydb1 created

C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl describe pods mydb1
Name:         mydb1
Namespace:    default
```

View environment variable now, it will not show: kubectl describe pods mydb1



```
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  <set to the key 'p' in secret 'mysecret'>  Optional: false
      MYSQL_DATABASE:       wpdb
      MYSQL_USER:           akshay
      MYSQL_PASSWORD:       <set to the key 'q' in secret 'mysecret'>  Optional: false
    Mounts:
```

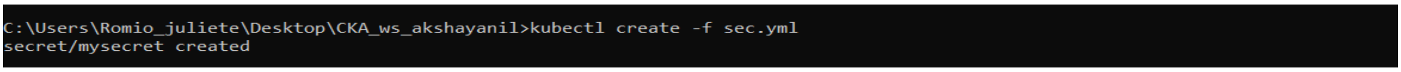
Code for SECRET KEY



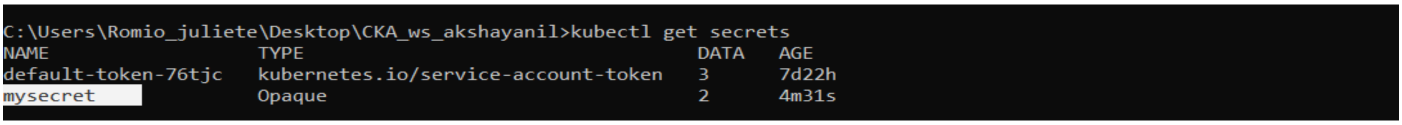
```
apiVersion: v1
kind: Secret
metadata:
    name: mysecret
data:
  p: cmVkaGF0
  q: YW5pbA==
```

kubectl create -f sec.yml - run the yml file

kubectl get secrets - check the secret file



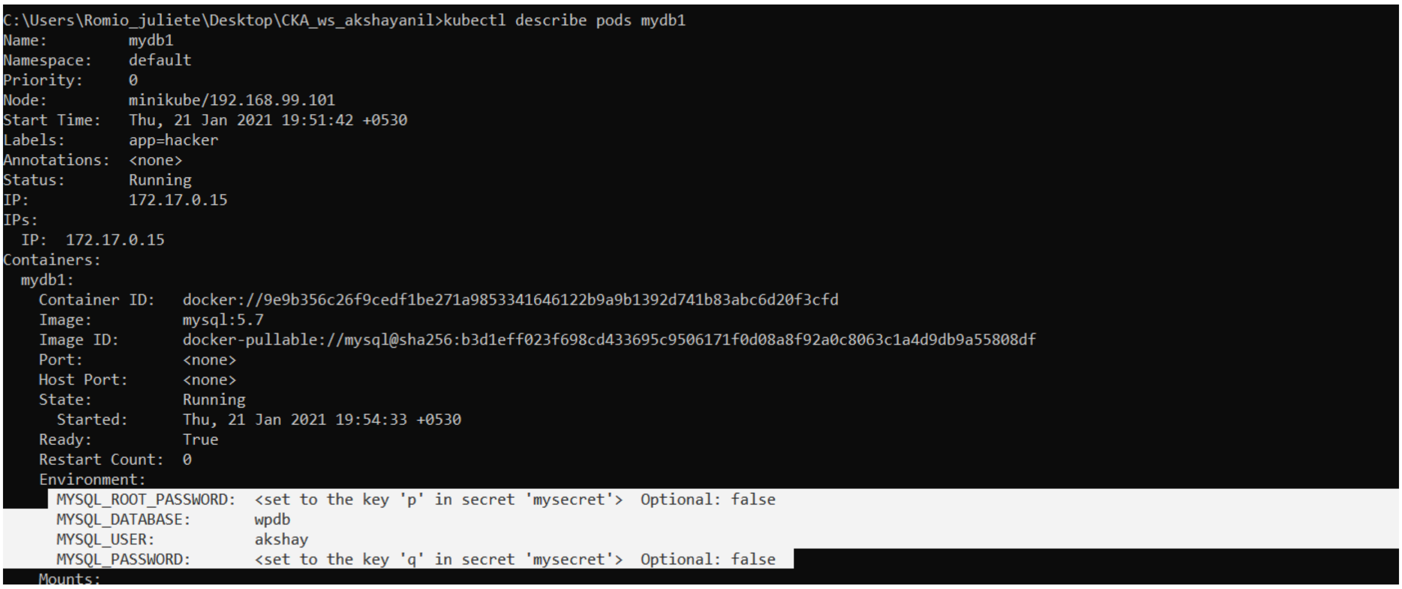
```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl create -f sec.yml
secret/mysecret created
```



```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl get secrets
NAME                  TYPE                                  DATA   AGE
default-token-76tjc   kubernetes.io/service-account-token   3      7d22h
mysecret              Opaque                                2      4m31s
```

4th: VERIFICATION

kubectl describe pods mydb1



```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl describe pods mydb1
Name:         mydb1
Namespace:    default
Priority:     0
Node:         minikube/192.168.99.101
Start Time:   Thu, 21 Jan 2021 19:51:42 +0530
Labels:       app=hacker
Annotations:  <none>
Status:       Running
IP:           172.17.0.15
IPs:
  IP:  172.17.0.15
Containers:
  mydb1:
    Container ID:   docker://9e9b356c26f9cedf1be271a9853341646122b9a9b1392d741b83abc6d20f3cfd
    Image:          mysql:5.7
    Image ID:       docker-pullable://mysql@sha256:b3d1eff023f698cd433695c9506171f0d08a8f92a0c8063c1a4d9db9a55808df
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 21 Jan 2021 19:54:33 +0530
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  <set to the key 'p' in secret 'mysecret'>  Optional: false
      MYSQL_DATABASE:       wpdb
      MYSQL_USER:           akshay
      MYSQL_PASSWORD:       <set to the key 'q' in secret 'mysecret'>  Optional: false
    Mounts:
```

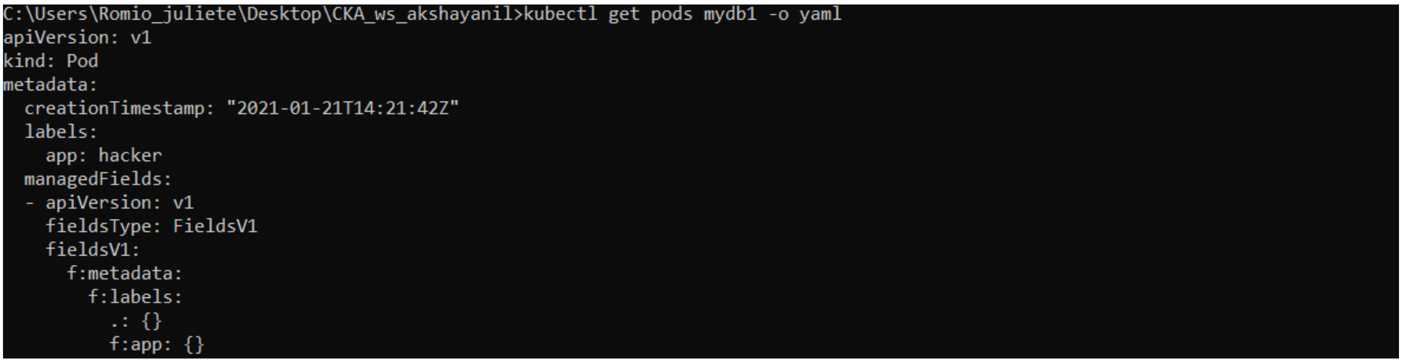
YES…… SUCCESSFULLY ENCODED THE PASSWORD…………

Another Challenge: is there a way to get the password after encoding too?

- K8 master only understand YAML.
- so, even when we used CLI to launch the pod , behind the scenes it first get converted to YAML code the K8 does the work.

Is there any way to see this code? Yes

kubectl get pods mydb1 -o yaml.

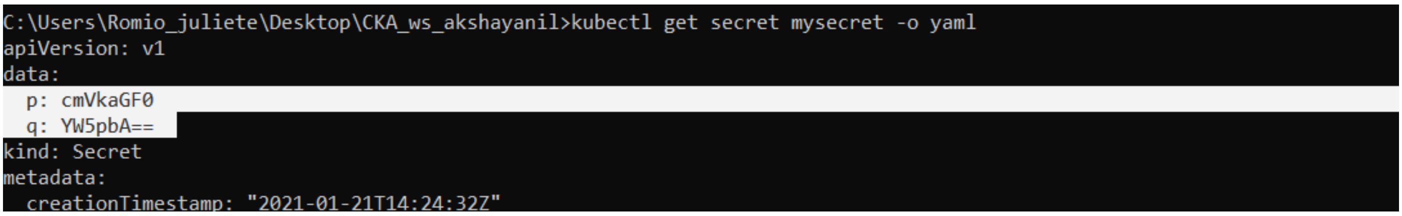


```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl get pods mydb1 -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: "2021-01-21T14:21:42Z"
  labels:
    app: hacker
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:metadata:
        f:labels:
          .: {}
          f:app: {}
```

It will be a very long code.

So, we can view the password here by:

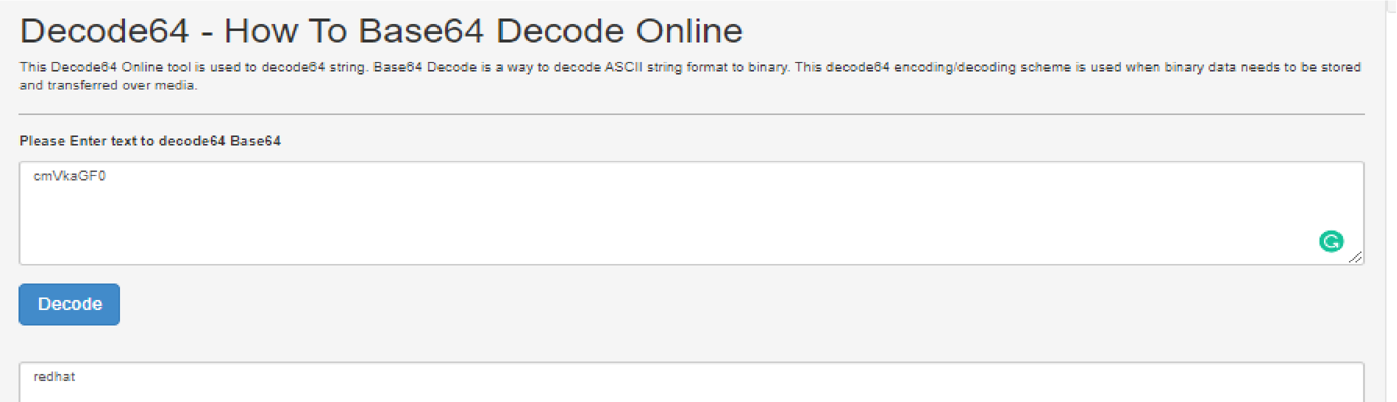
kubectl get secret mysecret -o yaml



```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl get secret mysecret -o yaml
apiVersion: v1
data:
  p: cmVkaGF0
  q: YW5pbA==
kind: Secret
metadata:
  creationTimestamp: "2021-01-21T14:24:32Z"
```

since its not encryption,

we can use decode64 from google or by CLI too to decode it.



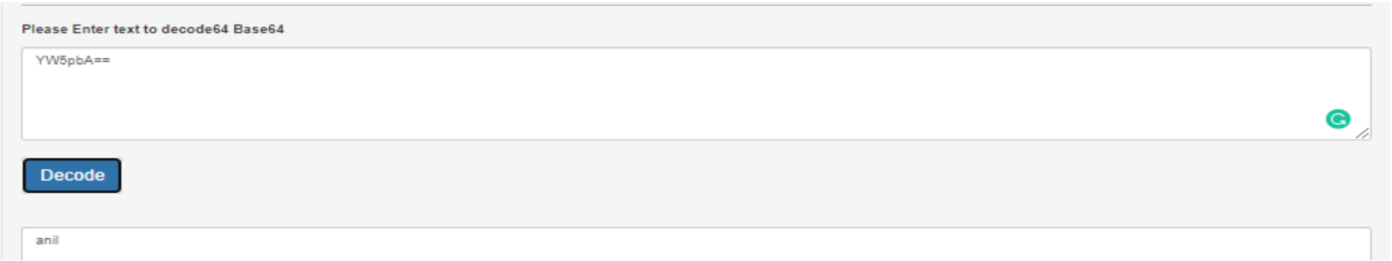
Decode64 - How To Base64 Decode Online

This Decode64 Online tool is used to decode64 string. Base64 Decode is a way to decode ASCII string format to binary. This decode64 encoding/decoding scheme is used when binary data needs to be stored and transferred over media.

Please Enter text to decode64 Base64

cmVkaGF0

redhat



Please Enter text to decode64 Base64

YW5pbA==

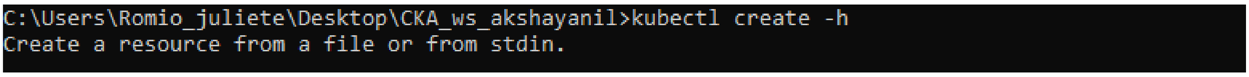
anil

Now, can we create a secret in K8 via CLI?

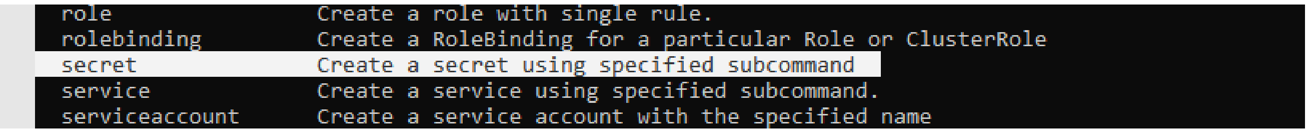
YES. lets see how.

1st ask k8 for help in creation

kubectl create -h



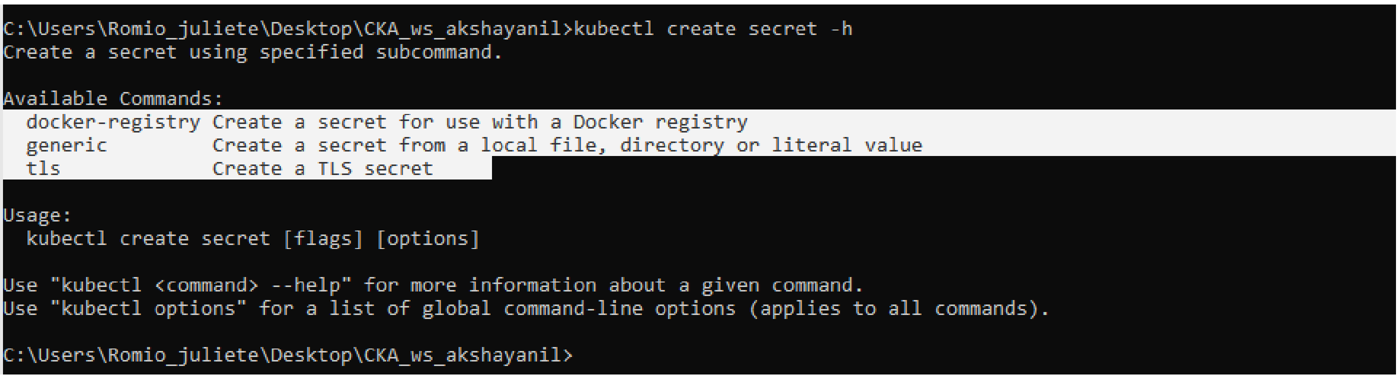
```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl create -h
Create a resource from a file or from stdin.
```



```
  role                 Create a role with single rule.
  rolebinding          Create a RoleBinding for a particular Role or ClusterRole
  secret               Create a secret using specified subcommand
  service              Create a service using specified subcommand.
  serviceaccount       Create a service account with the specified name
```

2nd: asking help in secret

kubectl create secret -h

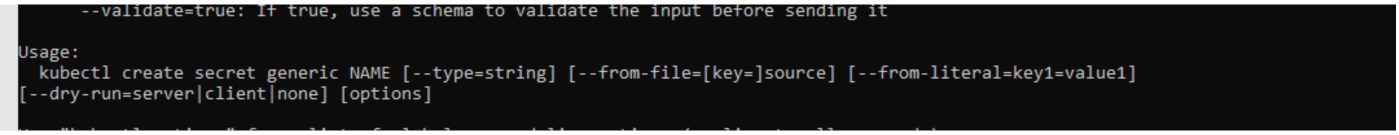


```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl create secret -h
Create a secret using specified subcommand.

Available Commands:
  docker-registry Create a secret for use with a Docker registry
  generic         Create a secret from a local file, directory or literal value
  tls             Create a TLS secret

Usage:
  kubectl create secret [flags] [options]

Use "kubectl <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).

C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>
```

till yet we created key-value pair i.e literal value for secret key i.e. generic

3rd: more help

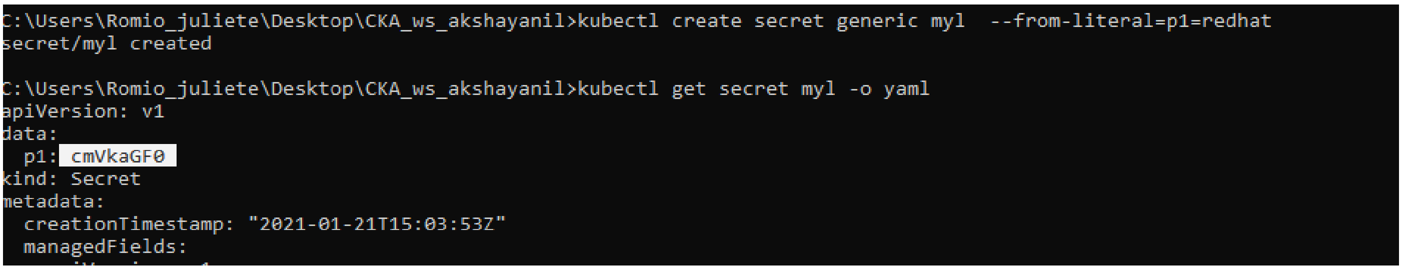
kubectl create secret generic -h



```
      --validate=true: If true, use a schema to validate the input before sending it

Usage:
  kubectl create secret generic NAME [--type=string] [--from-file=[key=]source] [--from-literal=key1=value1]
[--dry-run=server|client|none] [options]
```

4th: lets use now…

kubectl create secret generic mys --from-literal=p1=redhat



```
C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl create secret generic myl  --from-literal=p1=redhat
secret/myl created

C:\Users\Romio_juliete\Desktop\CKA_ws_akshayanil>kubectl get secret myl -o yaml
apiVersion: v1
data:
  p1: cmVkaGF0
kind: Secret
metadata:
  creationTimestamp: "2021-01-21T15:03:53Z"
  managedFields:
```

Also, it automatically encode your value to base64.
demonstrated successfully….

BY

AKSHAY ANIL

Track me down : https://akshayanil1080.github.io/mywebsite/

And much more to go……………….