MANAGING SECRETS IN KUBERNETES

The Nautilus DevOps team is working to deploy some tools in Kubernetes cluster. Some of the tools are licence based so that licence information needs to be stored securely within Kubernetes cluster. Therefore, the team wants to utilize Kubernetes secrets to store those secrets. Below you can find more details about the requirements:

1. We already have a secret key file official.txt under /opt location on jump host. Create a generic secret named official, it should contain the password/license-number present in official.txt file.
2. Also create a pod named secret-nautilus.
3. Configure pod's spec as container name should be secret-container-nautilus, image should be debian preferably with latest tag (remember to mention the tag with image). Use sleep command for container so that it remains in running state. Consume the created secret and mount it under /opt/demo within the container.
4. To verify you can exec into the container secret-container-nautilus, to check the secret key under the mounted path /opt/demo. Before hitting the Check button please make sure pod/pods are in running state, also validation can take some time to complete so keep patience.

Note: The kubectl utility on jump\_host has been configured to work with the kubernetes cluster

SOLUTION

-**kubectl create secret generic beta --from-file=/opt/beta.txt**

**-vi /tmp/secret.yml**

**-kubectl get pods**

**-kubectl exec secret-devops -- cat /opt/demo/beta.txt**

apiVersion: v1

kind: Pod

metadata:

name: secret-devops

labels:

name: myapp

spec:

volumes:

- name: secret-volume-devops

secret:

secretName: beta

containers:

- name: secret-container-devops

image: ubuntu:latest

command: ["/bin/bash", "-c", "sleep 10000"]

volumeMounts:

- name: secret-volume-devops

mountPath: /opt/apps

readOnly: true