CKA Lab Part 3 - Application Lifecycle Management

Lab 1 - Perform rolling updates on a deployment

Apply the following yaml file https://raw.githubusercontent.com/David-VTUK/CKAExampleYaml/master/nginx-svc-and-deployment.yaml

Update this deployment to leverage the nginx container version 1.7.11. Ensure that --record=true has been used.

```
wget https://raw.githubusercontent.com/David-VTUK/CKAExampleYaml/master/nginx-svc-and-deployment.yamlhttps://raw.githubusercontent.com/David-VTUK/CKAExampleYaml/master/nginx-svc-and-deployment.yaml

Nano nginx-svc-and-deployment.yaml


    spec:
   containers:
   - name: nginx
     image: nginx:1.7.11

Kubectl apply -f nginx-svc-and-deployment.yaml --record=true
```

Lab 2 - Change the update strategy for a deployment

Using the YAML file from Lab 1, amend it so that:

- Strategy is “Rolling Update”
- Max Surge is “1”
- Max Unavailable is “1”

```
Nano nginx-svc-and-deployment.yaml

spec:
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
```

Lab 3 - Perform a rollback on a deployment

Rollback the changes that were implemented from Lab 1.

```
kubectl rollout undo deployment nginx-deployment
```

Lab 4 - Scale a deployment

Scale the deployment from the first lab exercise to leverage 6 pods.

```
kubectl scale --replicas=6 deployment nginx-deployment
```

Lab 5 - Create and run a Job

Spec and execute a job that:

- Leverages the “perl” image
- Calculates pi to 2000 places

Note, use the command command: ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"] in the pod manifest

The command above will output to stdout on the container, therefore inspect the output

```
apiVersion: batch/v1
kind: Job
metadata:
  name: pi
spec:
```

```
  template:
    spec:
      containers:
      - name: pi
        image: perl
        command: ["perl",  "-Mbignum=bpi", "-wle", "print bpi(2000)"]
      restartPolicy: Never
  backoffLimit: 4

kubectl logs pi-vblhk
3.14159265358979…….
```

Lab 6 - Create and use a Config Map

Create two texts files in /tmp/

db_h.txt with the contents “database_host”

db_p.txt with the contents “database_port”

Create a configmap called “db-connection” from the above two files.

Create a nginx pod which leverages these values as environment variables “db_h” and “db_p”

```
echo "database_host" > /tmp/db_h.txt
echo "database_port" > /tmp/db_p.txt

kubectl create configmap dbconnection --from-file=dbh=/tmp/db_h.txt --from-file=dbp=/tmp/db_p.txt

apiVersion: v1
kind: Pod
metadata:
  name: nginx-web
  labels:
 role: web
```

```
spec:
  containers:
  - name: nginx
 image: nginx
 command: [ "/bin/sh", "-c", "env" ]
 env:
   - name: DB_HOST
     valueFrom:
       configMapKeyRef:
         name: db-connection
         key: db_h
   - name: DB_PORT
     valueFrom:
       configMapKeyRef:
         name: db-connection
         key: db_p
```

Lab 7 - Create and use Secrets

Create a secret called “db-credentials” directly from the CLI with the following key:value pair.

db-username : dbuser

db-password : dbpassword

```
kubectl create secret generic db-credentials --from-literal db-username=dbuser --from-literal db-password=dbpassword
```

Create a pod to leverage these as environment variables.

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-web-secret
  labels:
```

```
    role: web
spec:
  containers:
  - name: nginx
    image: nginx
    env:
      - name: db_username
        valueFrom:
          secretKeyRef:
           name: db-credentials
           key: db-username
      - name: db_password
        valueFrom:
          secretKeyRef:
            name: db-credentials
            key: db-password
```

Lab 8 - Configure a pod with specific environment variables

Create a pod that has two environment variables configured:

Variable1 = somevalue

Variable2 = someothervalue

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-web-env
  labels:
    role: web
spec:
  containers:
  - name: nginx
    image: nginx
    command: [ "/bin/sh", "-c", "env" ]
    env:
- name : variable1
  value : somevalue
- name : variable2
  value: someothervalue
```