JOHN BRYCE
Leading in IT Education
matrix company

8363

RHCA K8S WORKSHOP EXAM

Please make sure to write the commands you are writing as a separate document and send all of your solutions to a repository that you own.
Please send me the repo - make sure it’s public.

1. Deploy a pod named nginx-pod using the nginx:alpine image.
   Name: nginx-pod-yourname
   Image: nginx:alpine
2. Deploy a messaging pod using the redis:alpine image with the labels set to tier=msg.
   Pod Name: messaging
   Image: redis:alpine
   Labels: tier=msg
3. Create a namespace named apx-x998-yourname
4. Get the list of nodes in JSON format and store it in a file at /tmp/nodes-yourname
5. Create a service messaging-service to expose the messaging application within the cluster on port 6379.
   a. Use imperative commands - kubectl
   b. Service: messaging-service
   c. Port: 6379
   d. Type: ClusterIp
   e. Use the right labels
6. Create a service messaging-service to expose the messaging application within the cluster on port 6379.
   a. Service: messaging-service
   b. Port: 6379
   c. Type: ClusterIp
   d. Use the right labels
7. Create a deployment named hr-web-app using the image kodekloud/webapp-color with 2 replicas
   a. Name: hr-web-app
   b. Image: kodekloud/webapp-color
   c. Replicas: 2
8. Create a static pod named static-busybox on the master node that uses the busybox image and the command sleep 1000
   a. Name: static-busybox
   b. Image: busybox
9. Create a POD in the finance-yourname namespace named temp-bus with the image redis:alpine
   a. Name: temp-bus

© All rights reserved to John Bryce Training LTD

    b. Image Name: redis:alpine

10. Create a Persistent Volume with the given specification
    a. Volume Name: pv-analytics
    b. Storage: 100Mi
    c. Access modes: ReadWriteMany
    d. Host Path: /pv/data-analytics
11. Create a Pod called redis-storage-yourname with image: redis:alpine with a Volume of type emptyDir that lasts for the life of the Pod. specs:.
    a. Pod named 'redis-storage-yourname'
    b. Pod 'redis-storage-yourname' uses Volume type of emptyDir
    c. Pod 'redis-storage-yourname' uses volumeMount with mountPath = /data/redis
12. Create this pod and attached it a persistent volume called pv-1
    a. Make sure the PV mountPath is hostbase : /data

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: use-pv
  name: use-pvspec-yourname
  containers:
  - image: nginx
    name: use-pv
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

13. Create a new deployment called nginx-deploy, with image nginx:1.16 and 1 replica. Record the version. Next upgrade the deployment to version 1.17 using rolling update. Make sure that the version upgrade is recorded in the resource annotation.
    a. Deployment : nginx-deploy. Image: nginx:1.16
    b. Image: nginx:1.16
    c. Task: Upgrade the version of the deployment to 1:17
    d. Task: Record the changes for the image upgrade
14. Create an nginx pod called nginx-resolver using image nginx, expose it internally with a service called nginx-resolver-service. Test that you are able to look up the

© All rights reserved to John Bryce Training LTD

service and pod names from within the cluster. Use the image: busybox:1.28 for dns lookup. Record results in /root/nginx-yourname.svc and /root/nginx-yourname.pod

15. Create a static pod on node01 called nginx-critical with image nginx. Create this pod on node01 and make sure that it is recreated/restarted automatically in case of a failure.

16. Create a pod called multi-pod with two containers.
    Container 1, name: alpha, image: nginx
    Container 2: beta, image: busybox, command sleep 4800.
    a. Environment Variables:
        i. container 1:
        ii. name: alpha

        iii. Container 2:
        iv. name: beta

Pod Design Questions:

- Understand how to use Labels, Selectors and Annotations
- Understand Deployments and how to perform rolling updates
- Understand Deployments and how to perform rollbacks
- Understand Jobs and CronJobs

1. Type the command for:

   Get pods with label information

2. Create 5 nginx pods in which two of them is labeled env=prod and three of them is

   labeled env=dev

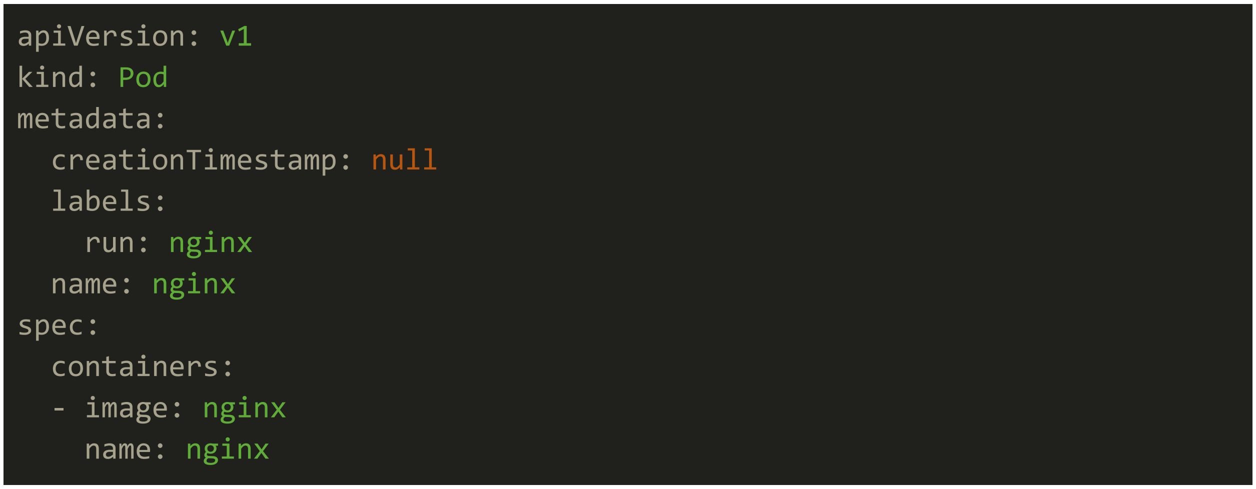
3. Verify all the pods are created with correct labels

4. Get the pods with label env=dev

5. Get the pods with label env=dev and also output the labels

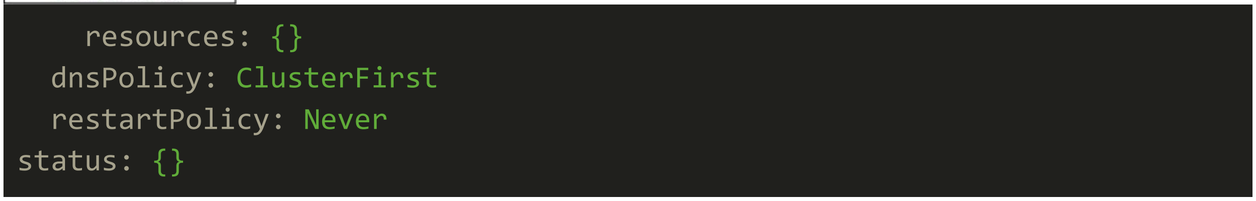
6. Get the pods with label env=prod
7. Get the pods with label env=prod and also output the labels
8. Get the pods with label env
9. Get the pods with labels env=dev and env=prod
10. Get the pods with labels env=dev and env=prod and output the labels as well
11. Change the label for one of the pod to env=uat and list all the pods to verify
12. Remove the labels for the pods that we created now and verify all the labels are removed
13. Let’s add the label app=nginx for all the pods and verify (using kubectl)
14. Get all the nodes with labels (if using minikube you would get only master node)
15. Label the worker node nodeName=nginxnode
16. Create a Pod that will be deployed on the worker node with the label nodeName=nginxnode

Add the **nodeSelector** to the below and create the pod



```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
```

© All rights reserved to John Bryce Training LTD



```
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Never
status: {}
```

17. Verify the pod that it is scheduled with the node selector on the right node… fix it if it’s not behind scheduled.
18. Verify the pod nginx that we just created has this label

Deployments:

1. Create a deployment called webapp with image nginx with 5 replicas
    a. Use the below command to create a yaml file.
        i. kubectl create deploy webapp --image=nginx --dry-run -o yaml > webapp.yaml
        ii. Edit it and add 5 replica’s
2. Get the deployment rollout status
3. Get the replicaset that created with this deployment
4. EXPORT the yaml of the replicaset and pods of this deployment
5. Delete the deployment you just created and watch all the pods are also being deleted

© All rights reserved to John Bryce Training LTD

6. Create a deployment of webapp with image nginx:1.17.1 with container port 80 and verify the image version
   a. kubectl create deploy webapp --image=nginx:1.17.1 --dry-run -o yaml > webapp.yaml
   b. add the port section (80) and create the deployment
7. Update the deployment with the image version 1.17.4 and verify
8. Check the rollout history and make sure everything is ok after the update
9. Undo the deployment to the previous version 1.17.1 and verify Image has the previous version
10. Update the deployment with the wrong image version 1.100 and verify something is wrong with the deployment
    a. Expect: kubectl get pods (ImagePullErr)
    b. Undo the deployment with the previous version and verify everything is Ok
    c. kubectl rollout history deploy webapp --revision=7
    d. Check the history of the specific revision of that deployment
    e. update the deployment with the image version latest and check the history and verify nothing is going on
11. Apply the autoscaling to this deployment with minimum 10 and maximum 20 replicas and target CPU of 85% and verify hpa is created and replicas are increased to 10 from 1
12.
13. Clean the cluster by deleting deployment and hpa you just created

© All rights reserved to John Bryce Training LTD

14. Create a job and make it run 10 times one after one (run > exit > run >exit ..) using the following configuration:

    kubectl create job hello-job --image=busybox --dry-run -o yaml -- echo "Hello I am from job" > hello-job.yaml"

    a. Add to the above job **completions: 10 inside the yaml**

© All rights reserved to John Bryce Training LTD

CONFIG MAP:

1. Create a file called config.txt with two values key1=value1 and key2=value2 and verify the file

```
cat >> config.txt << EOF
key1=value1
key2=value2
EOF
cat config.txt
```

2. Create a configmap named keyvalcfgmap and read data from the file config.txt and verify that configmap is created correctly
3. Create an nginx pod and load environment values from the above configmap keyvalcfgmap and exec into the pod and verify the environment variables and delete the pod

   // first run this command to save the pod yml

   kubectl run nginx --image=nginx --restart=Never --dry-run -o yaml > nginx-pod.yml

© All rights reserved to John Bryce Training LTD