**EX280-Red Hat Certified Specialist in OpenShift Administration Exam-Version\_4.14** **Duration**: 3 Hrs

**Max.Mark**:300 **Pass Mark:** 210

***Initial Setup***

|  |  |  |
| --- | --- | --- |
| **S.NO** | ***DOMAIN NAME*** | ***IP ADDRESS*** |
| 1 | Workbench.lab.example.com | 172.25.250.11 |
| 2 | master01.lab.example.com | 172.25.250.12 |
| 3 | Utility.lab.example.com | 172.25.250.14 |
| 4 | API Server URL | https://api.ocp4.example.com:6443 |

1. Wild-card domain for the cluster: **apps.ocp4.example.com**
2. Documentation about openshift can be accessed at the following url:

<https://access.redhat.com/documentation/en-us/openshift_container_platform/4.14>

1. Kubeadmin password will be available in the location as /home/student/kubeadmin-password in the workbench VM itself.
2. User ocpadm user will be given.
3. ocpadm password for login in to workbench VM will be provided in the exam itself.

***Question Outline***

1. Configure the Identity Provider for the OpenShift
2. Configure Cluster permissions
3. Configure Project permissions
4. Create Groups and configure permissions
5. Configure and deploy a secure route
6. Scale the Application manually
7. Configure Auto-scaling for an application
8. Configure a Secret
9. Use the Secret value for Application Deployment
10. Configure a Service Account
11. Deploy an application
12. Deploy an application
13. Network policy
14. Persistent storage
15. Install Operator
16. Cronjob
17. Helm
18. Bootstrap Project template
19. Monitoring and health-check
20. Liveness probe

***Detailed Questions***

1. **Configure the Identity Provider for the Openshift**

* Create an Htpass Identity Provider with the name: htpass-ex280
* Create the secret for Identity provider users: htpass-idp-ex280
* Create the user account jobs with password deluges
* Create the user account wozniak with password grannies
* Create the user account collins with password culverins
* Create the user account adlerin with password artiste
* Create the user account armstrong with password spacesuits

ANS:

Unzip the given zip file

Login to the cluster using admin user here but in global exam should use kubeadmin as a user name.

# **oc login -u admin -p redhatocp https://api.ocp4.example.com:6443**

Execute the project.sh script and start the exam practice   
# vim ~/.vimrc

Set ai ts=2 et cuc 🡪 save the file

Step:1

Install the httpd-tools package in the workbench machine

#sudo yum install httpd-tools (only for exam)

Then create a directory and save the htpasswd users credentials under it

#mkdir mypass

#htpasswd –c –B –b mypass/users jobs deluges

#htpasswd –B –b mypass/users wozniak grannies

#htpasswd –B –b mypass/users collins culverins

#htpasswd –B –b mypass/users adlerin artiste

#htpasswd –B –b mypass/users armstrong spacesuits

#cat mypass/users

**STEP-2:**

Import the htpasswd users file in to the openshift as the secret in to the project called openshift-config

#oc create secret generic htpass-idp-ex280 --from-file htpasswd=mypass/users –n openshift-config

# oc get secret htpass-idp-ex280 –n openshift-config

**STEP-3:**

Configure the Oauth server to use HTPass Identity Provider.

Use the provided Documentation to get the file, copy the file and change the appropriate value as highlighted

### Authentication and authorization 🡪configuring identity provider🡪 Configuring an htpasswd identity provider 🡪 7.1.5 Sample htpasswd CR

#cat > oauth.yaml

apiVersion: config.openshift.io/v1

kind: OAuth

metadata:

name: cluster

spec:

identityProviders:

- name: htpass-ex280

mappingMethod: claim

type: HTPasswd

htpasswd:

fileData:

name: htpass-idp-ex280

while you practice use the below steps for oauth file creations.

# oc get oauth cluster –o yaml > oauth.yaml

Edit and update the oauth info in existing oauth file

Refer the do280 v-4.14 book p.no: 101



#oc replace –f oauth.yaml

#oc get oauth cluster –o yaml

# watch oc get pods –n openshift-authentication

**STEP-4:**

Login as all the users and verify. You should be able to login successfully

#oc login –u <user\_name> -p <password\_of\_user>

1. **Configure Cluster permissions**

* User jobs is able to modify the cluster.
* wozniak is able to create projects.
* armstrong cannot create projects.
* wozniak cannot modify the cluster.
* Remove the kubeadmin user from the cluster.

ANSWER:

# oc get clusterrolebindings –o wide | grep jobs 🡪 check the permission

* User jobs is able to modify the cluster

#oc adm policy add-cluster-role-to-user cluster-admin jobs

# oc get clusterrolebindings –o wide | grep jobs 🡪 check the permission

* wozniak is able to create project

#oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth 🡪 remove the permission for all the user

#oc adm policy add-cluster-role-to-user self-provisioner wozniak 🡪 only to give the permission for wozniak user

# oc describe clusterrolebindings self-provisioner

* amstrong cannot create projects

Login as that user and verify that he is not able to create project. No need to make any configuration change.

* wozniak cannot modify the cluster

Login as that user and verify that he is not able to execute cluster level commands like “oc get nodes”. No need to make any configuration change.

* Remove the kubeadmin user from the cluster

Don’t practice this command in the lab and only do it in the exam

# oc get secret kubeadmin –n kube-system 🡪check the kube user secret info

#oc delete secret kubeadmin –n kube-system

1. **Configure Project permissions**
   1. Create following projects
      1. apollo
      2. titan
      3. gemini
      4. bluebook
      5. apache
   2. User armstrong is admin for the apollo and titan project.
   3. User collins is able to view the apollo project.

ANSWER:

# **oc login –u jobs –p deluges**

1. #oc new-project apollo

#oc new-project titan

#oc new-project gemini

#oc new-project bluebook

#oc new-project manhattan

# oc get project

1. #oc policy add-role-to-user admin armstrong –n apollo

#oc policy add-role-to-user admin armstrong –n titan

1. #oc policy add-role-to-user view collins –n apollo

#oc get rolebinding –o wide -n apollo

#oc get rolebinding –o wide –n titan

1. **Create Groups and configure permissions**
   1. Create a group called commander and user wozniak is the member of this group.
   2. Create a group called pilot and user adlerin is the member of this group.
   3. The commander group members are able to edit the apollo and titan project.
   4. The pilot group members are able to view apollo project but not edit it.

ANSWER:

1. #oc adm groups new commander

#oc adm groups add-users commander wozniak

1. #oc adm groups new pilot

#oc adm groups add-users pilot adlerin

#oc get groups 🡪 check the group info

1. #oc adm policy add-role-to-group edit commander –n apollo

#oc adm policy add-role-to-group edit commander –n titan

1. #oc adm policy add-role-to-group view pilot –n apollo

#oc get rolebinding –o wide –n apollo/titan 🡪 check the all projects

1. **i, Configure Quotas for the Project**

**Note: in global exam it will come only one question either quota or limit range.**

**Create Resource Quota in manhattan project named ex280-quota**

* 1. **The amount of memory consumed across all containers may not exceed 1Gi**
  2. **The amount of CPU across all containers may not exceed 2 full cores.**
  3. **The maximum number of replication controllers does not exceed 3**
  4. **The maximum number of pods does not exceed 3**
  5. **The maximum number of services does not exceed 6**

ANSWER:

# oc project manhattan

#oc create quota ex280-quota --hard limits.memory=1Gi,limits.cpu=2,replicationcontrollers=3,pods=3,services=6

#oc describe resourcequota ex280-quota 🡪 Verify

# oc delete resourcequota ex280-quota **<if you want delete and recreate mean use it>**

**5.ii, Configure Limits for the Project**

**Create a Limit Range in the bluebook project name ex280-limits**

* 1. **The amount of memory consumed by a single pod is between 100Mi and 300Mi**
  2. **The amount of cpu consumed by a single pod is between 10m and 500m**
  3. **The amount of cpu consumed by a single container is between 10m and 500m with a default request value of 100m**
  4. **The amount of memory consumed by a single container is between 100Mi and 300Mi with a default request value of 100Mi**

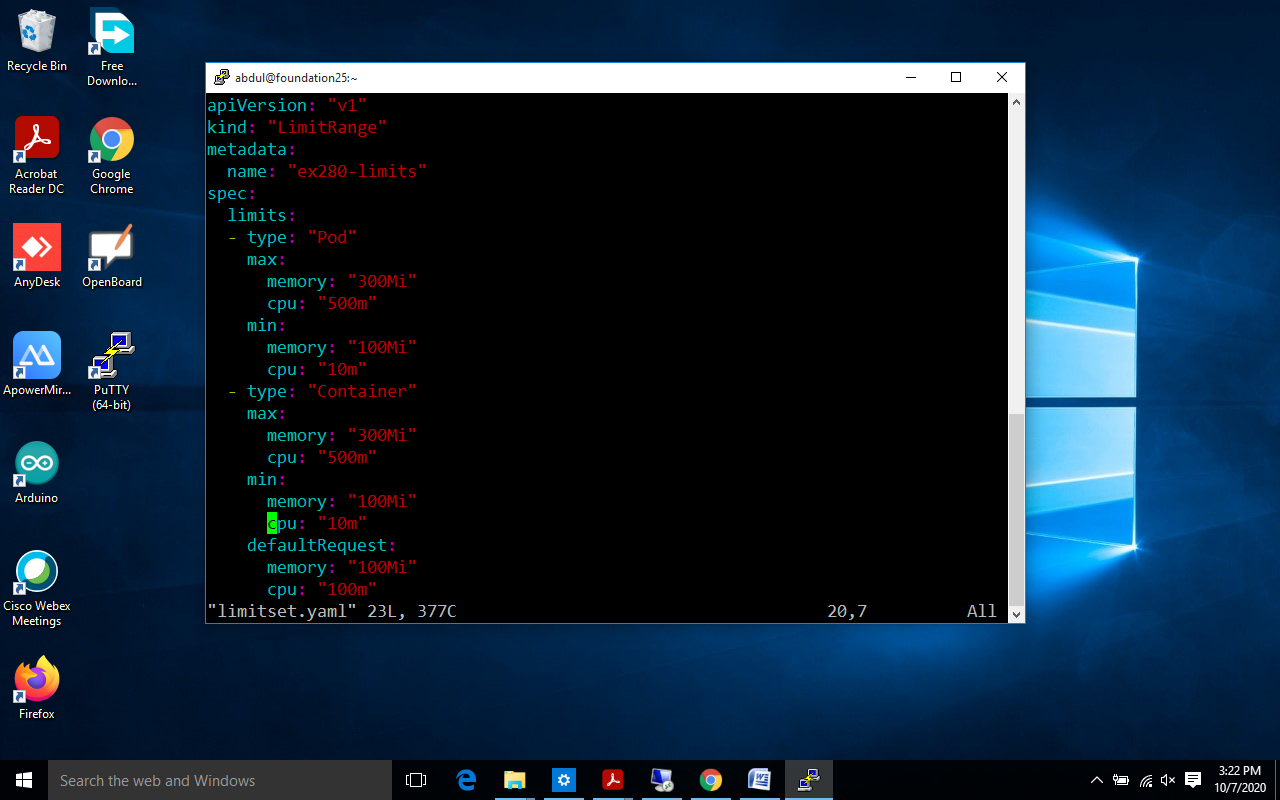
ANSWER:

#oc project bluebook

#vim limits.yaml

The content of the file should be as in the screen-shot with same indentation

Develop ->nodes->**working with cluster🡪 Restrict resource consumption with limit range🡪** Creating a Limit Range(8.3.2)



#oc create –f limits.yaml

#oc describe limitranges ex280-limits 🡪 verify

1. **Configure and deploy a secure route**

Deploy an application called oxcart securely in the project called area51

* 1. The application has self-signed certificate available at

"/C=US/ST=NC/L=Raleigh/O=RedHat/OU=RHT/CN=oxcart.apps.ocp4.example.com"

Use **newcert** command to generate required tls objects, pass the subject as argument value.

* 1. The application should be reachable at the following url

https://oxcart.apps.ocp4.example.com

* 1. Application produces a valid Output.

ANSWER:

Execute the startscenario.sh script file.

#oc project area51

#oc get pods

#oc get route

#oc delete route oxcart 🡪Delete the route because it is insecure

#oc get route 🡪Verify it

To use the newcert command to generate CA

# sudo cp newcert /bin

# sudo chmod a+x /bin/newcert

# newcert "/C=US/ST=NC/L=Raleigh/O=RedHat/OU=RHT/CN=oxcart.apps.ocp4.example.com"

Create the Edge Route

#oc get service

#oc create route edge --service=oxcart --cert=apps-crc.testing.crt --key=apps-crc.testing.key –hostname=oxcart.apps.ocp4.example.com

#oc get route

verify the URL using firefox

https://oxcart.apps.ocp4.example.com

1. **Scale the Application manually**

Scale an application called hydra in the project called lerna

The hydra application should be scaled to five times.

ANSWER:

#oc project lerna

#oc get deployment

#oc scale --replicas=5 deployment.apps/hydra

#oc get pods

1. **Configure Autoscaling for an Application**

Configure an autoscaling for the scala application in the project gru with following specification.

* 1. Minimum number of replicas: 6
  2. Maximum number of replicas: 40
  3. Threshold CPU-Percentage: 60
  4. Application resource of CPU Request: 25m
  5. Application limits of CPU Limits: 100m

ANSWER:

#oc project gru

#oc get pods

#oc get deployment

**Set Application Resources:**

#oc set resources --requests cpu=25m --limits cpu=100m deployment.apps/scala

#oc describe deployment.apps/scala | grep –A4 Limits

**Set Autoscaling:**

#oc autoscale --min=6 --max=40 --cpu-percent=60 deployment.apps/scala

#oc get hpa

1. **Configure a Secret**

Configure a secret in the math project and the name of secret should be magic.

The secret should have following key value pairs

Decoder\_Ring: ASDA142hfh-gfrhhueo-erfdk345v

ANSWER:

#oc project math

#oc create secret generic magic --from-literal Decoder\_Ring=ASDA142hfh-gfrhhueo-erfdk345v

#oc get secret magic -o yaml

1. **Use the Secret value for Application Deployment**

* Configure the environmental variable for the application called qed in the math project so that it uses the secret “magic”
* After configuring the environmental value for the application, it should stop producing the following output

“Hello Openshift”

ANSWER

# oc get pods

# oc get route

# curl 🡪 pre-check

# oc describe pod qed-7766d4df7-5l9wx | grep –A1 Environment

# oc get all | grep deploy

# oc set env --from secret/magic deployment.apps/qed

# oc describe pod qed-5ddb5b5cfc-bjqkw | grep -A1 Environment

# oc get route

# curl 🡪 you will get diff o/p in global exam

1. **Configure a Service Account**

Create a service account called ex-280-sa in the project called apples.

This service account should able to run application with anyuserid.

ANSWER

#oc project apples

#oc create serviceaccount ex-280-sa

#oc get sa ex-280-sa

#oc adm policy add-scc-to-user anyuid -z ex-280-sa

#oc get clusterrolebinding -o wide | grep ex-280-sa

1. **Deploy an application**

Deploy an application called oranges in the project called apples

* 1. This application should use the service account ex-280-sa
  2. The Application should produce a valid output.

ANSWER:

**Step:1 update the SA in deployment**

#oc get pods 🡪 **precheck**

#oc logs oranges-bc578f98d-mm8d6

#oc get all | grep deploy

#oc set serviceaccount deployment.apps/oranges ex-280-sa

#oc get pods 🡪 **post check**

**Step:2 Check the change the label name**

#oc describe pod oranges-7849dcbd68-wqhhc | grep Labels

#oc get service

#oc describe service oranges | grep Selector 🡪 pre-check

#oc describe service oranges | grep Endpoint 🡪 pre-check

#oc edit service oranges

Spec🡪Selector 🡪 deployment:orange to **deployment: oranges**

#oc describe service oranges | grep Selector 🡪 post-check

#oc describe service oranges | grep Endpoint 🡪 post-check

#oc get route

#curl oranges.apps.ocp4.example.com

1. **Deploy an application called mercury in the project atlas**
   1. Don’t add any new configuration.
   2. Application should produce a valid output.

ANSWER:

# oc project atlas

#oc get pod

#oc get events

# oc get all | grep deploy

# oc set resources --requests memory=256Mi deployment.apps/mercury

# oc get pod

# oc get route

# curl 🡪 verify the o/p

1. **Network Policy**

Deploy an application called mercury in the project atlas

* 1. Don’t add any new configuration
  2. Application should produce a valid output
  3. In the atlas project, create a network policy to allow traffic to the mercury pod in the atlas project from the rocky pod in the bluewills project via **TCP** on port **3102**. Here 8080 only.
  4. Ingress from  
     project label **kubernetes.io/metadata.name: bluewills**  
     pod label **deployment: rocky**
  5. create a policy in the name (**networkpolicy-ex280**)
  6. The application pod should be running

Answer:

# oc project atlas

# oc describe pod mercury-57bf74444-vzs4j |grep Label

## Networking -> networkpolicy -> 21.2. Creating a network policy --> we have to refer two files from the documents

[https://docs.redhat.com/en/documentation/openshift\_container\_platform/4.12/html-single/networking/index#nw-networkpolicy-object\_creating-network-policy](EX280V414K_OL.docx) 🡪 OCP document exact link. from this document we have to take 2 declarative file 1st one and last one.

Note: we have to take the starting content from last file then will take the last contents from 1 st file.

N/W POLICY FILE

kind: NetworkPolicy

apiVersion: networking.k8s.io/v1

metadata:

name: networkpolicy-ex280 🡪 POLICY NAME take it from q/p

spec:

podSelector:

matchLabels:

deployment: mercury 🡪 DEST APP LABEL NAME take it oc describe cmnd

ingress:

- from:

- namespaceSelector:

matchLabels:

kubernetes.io/metadata.name: bluewills 🡪 SOURCE PROJECT LBELNAME take it from q/p

podSelector:

matchLabels:

deployment: rocky 🡪 SOURCE APP LABEL NAME take it from q/p

ports:

- protocol: TCP

port: 8080

# oc create –f networkpolicy.yml

# oc get networkpolicy

# oc get pod –o wide –n atlas 🡪 pod ipaddress

# oc project bluewills

# oc rsh rocky-7fcffcd-dm6cv

🡪 # Curl 10.8.0.85:8080 check the o/p

1. **Persistent storage**

Deploy Application in the project

 Create application named ‘**gamma**’

 Application should be on project **space**

Number of pods should be running 3

 Application uses the image- quay.io/redhattraining/hello-world-nginx

 Provision storage for the file system /srv

 Application should be reachable at the following URL

<https://space.apps.ocp4.example.com>

 Storage instance should be named ‘gamma-pv’

 PV should be of size 1Gi

 Reclaim policy should ‘Delete’ [should be same as NFS Storageclass]

 NFS Server is available at 192.168.50.254 (get it from oc get storageclass)

 NFS Share Directory: /exports-ocp4

 Claim mode should be ‘ReadWriteMany’

**Note:** In exam NFS Server Details may not be given directly, you need to check NFS storageclass, get the details

Create a PV for the application

ing the command

#oc get storageclass

#oc get storageclass <name> -o yaml <Use this in exam to get the inputs>

Create a PVC for the application

 Create a PVC named ‘gamma-pvc’ for space project

 PVC should of size 1Gi

 Claim mode should be ‘ReadWriteMany’

 PVC should bind to ‘gamma-pv’

Answer:

**Step:1 🡪 application deployment**

# oc new-project space

# oc new-app --name=gamma --image=quay.io/redhattraining/hello-world-nginx

# oc scale --replicas 3 deployment.apps/gamma

# oc get svc

# oc create route edge --service=gamma --hostname= space.apps.ocp4.example.com

# oc rsh gamma-85fd5d9f69-4tg2k

# df –hT 🡪 precheck

**Step:2-> storage creation**

* Storage -> understanding persistent storage -> persistent volumes -> 3.3.4phase -> 3.3.4.1-> mount options

From the file we have to remove the particular 2 lines

i,mount options ii,-nfsvers=4.1

then add the values as per the q/p

PV creation file

apiVersion: v1

kind: PersistentVolume

metadata:

name: gamma-pv 🡪 take it Q/P

spec:

capacity:

storage: 1Gi 🡪 take it Q/P

accessModes:

- ReadWriteMany 🡪 take it Q/P

persistentVolumeReclaimPolicy: Delete 🡪 take it SC

nfs:

path: /exports-ocp4

server: 192.168.50.254 🡪 take it SC

claimRef:

name: gamma-pvc 🡪 take it Q/P

namespace: space 🡪 take it Q/P

# oc create –f pv.yaml

# oc get pv gamma-pv

PVC CREATION

# oc set volumes --type pvc --claim-name gamma-pvc --claim-size=1Gi --claim-mode ReadWriteMany --add --mount-path /srv deployment.apps/gamma

# oc get pv gamma-pv

# oc get pod

# oc rsh gamma-85fd5d9f69-4tg2k

#df –hT 🡪 post check

1. **Install Operator**

 Install **File Integrity operator** in the **openshift-file-integrity** project.

 Operator should have **stable** channel.

Enable cluster wide monitoring

 Operator should **automatically** update itself through OLM.

Answer:

Refer the p.no 295 from DO280 V-4.14 Redhat book

1. **Cron job**

Deploy a cronjob

 Cronjob should be part of marathon project

 Cronjob named as the application image name.

 Use the image quay.io/redhattraining/scaling

 The Job should be executed at 4.05 every 2nd day of the month.

 Use Service account called ex280-ocpsa.

 Successful Job History Limit should be 13.

Answer:

# oc new-project marathon

# oc create cronjob scaling --image quay.io/redhattraining/scaling --schedule '5 4 2 \* \*'

# oc get cronjob

# oc create sa ex280-ocpsa

# oc edit cronjob scaling 🡪 add the service account name in last spec and above of containers like serviceAccountName: ex280-ocpsa and edit succesfulljobhistorylimit from 3 to 13.

# oc describe cronjob scaling

1. **Helm Charts:**

Deploy an application and its dependencies from a Helm chart.

* Helm chart path: <http://helm.ocp4.example.com/charts>
* Give repo name on your own
* Install the etherpad chart to the charts-development project.
* Application Should be available at the following URL

https://etherpad-charts-development.apps.ocp4.example.com

Verify the output using nc and telnet

Answer:

# helm version

# oc new-project charts-development

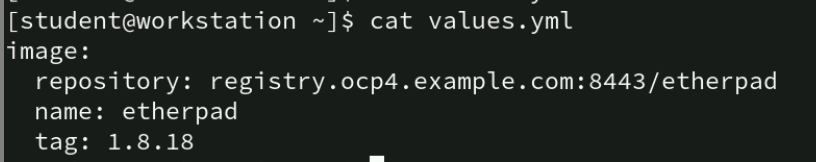
# helm repo add myrepo <http://helm.ocp4.example.com/charts>

# helm repo list

# helm search repo myrepo

# helm show values myrepo/etherpad

# cat > values.yml



# helm install etherpad myrepo/etherpad -f values.yml

# oc get pod

# oc get svc

# nc (svc ip address)

# telnet (diff ip) 🡪 check from another terminal using given ipaddress

1. **Bootstrap Project template**

As the administrator, update the OpenShift cluster to use a new project template.

* 1. The project template must automatically create the limit ranges for new projects. For example, a project named test has the limit range as test-limitrange respectively.
  2. Project should have default **LimitRange** as below

• Each container requests 30 MiB of memory.

• Each container is limited to 100 MiB of memory.

. Each container is set to min 28Mi of memory

. Each container is set to max 102Mi of memory

Answer:

# oc adm create-bootstrap-project-template –o yaml > proj-temp.yaml

# vim proj-temp.yaml 🡪 add the below contents in the file above of parameters

- apiVersion: "v1"

kind: "LimitRange"

metadata:

name: "${PROJECT\_NAME}-limitrange"

spec:

limits:

- type: "Container"

memory:

min: "28Mi"

memory:

max: "102Mi"

default:

memory: "100Mi"

defaultRequest:

memory: "30Mi"

# oc create –f proj-temp.yaml –n openshift-config

# oc get projects.config.openshift.io

# oc edit projects.config.openshift.io cluster

* Add the below content in spec part like and remove the curly braces

spec:

projectRequestTemplate:

name: project-request

# oc get pod –n openshift-apiserver

# oc new-project test

# oc get limitranges

1. **Monitoring and health-check:**

* Use the must-gather tool for collecting log files and diagnostic information about your cluster.
* Store the result at /home/student/ex280-<clusterID>.tar.gz
* Upload the data in <https://app.com/upload>

Answer:

# oc get clusterversions –o yaml | grep –i clusterID

# mkdir ex280-<cluster-id>

# oc adm must-gather –dest-dir ex280-<cluster-id>

# ls ex280-<cluster-id>

# tar –zcvf ex280-<cluster-id>.tar.gz ex280-<cluster-id>

21. **Activate liveness probes for the application in space project.**

 It should utilize the tcp socket port:8080

 Probe parameters, initial-delay-seconds=3 and timeout- seconds=10.

Answer:

# oc project space

# oc get deployment

# oc describe deployment gamma | grep –i live

# oc set probe --liveness --open-tcp 8080 --initial-delay-seconds 3 --timeout-seconds 10 deployment.apps/gamma

# oc describe deployment gamma | grep –i live