

P. 考试说明

L. 练习要求

1. HTPasswd
2. add-role-to-user
3. user account
4. use quotas
5. LimitRange
6. scale
7. Scale an application automatically
8. route
9. Deploy an application
10. Configure a secret
11. Configure an application to use a secret
12. Configure a service account
13. uses the service account
14. request memory
15. Configure application data
16. network policy
17. template
18. operator
19. cron job
20. Collect the default support information
21. A storage class has been configured to provide NFS storage
22. liveness probe

O. OBJECTIVE

P. 考试说明

1. 大多数题可以使用网页做
2. 网页登录，使用的地址、帐号、密码考题中都会提供
3. 命令行，需要先 ssh 至 workbech。主机名、帐号、密码考题中都会提供

L. 练习要求

1. HTPasswd

Configure your OpenShift cluster to use an HTPasswd identity provider with the following requirements:

- ☐ The name of the identity provider is: `ex280-htpasswd`

- ☐ The name of the secret is: `ex280-idp-secret`
- ☐ The user account `armstrong` is present and can log in with password `indionce`
- ☐ The user account `collins` is present and can log in with password `veraster`
- ☐ The user account `aldrin` is present and can log in with password `roonkere`
- ☐ The user account `jobs` is present and can log in with password `sestiver`
- ☐ The user account `wozniak` is present and can log in with password `glegunge` ,
Configure `wozniak` to create a project
- ☐ Delete virtual user

[opsadm@workbench]

```
1  $ htpasswd
2  `htpasswd` -b[cmBdpsDv] [-C cost] `passwordfile` `username` `password`
3  -b Use the password from the command line rather than prompting for it.
4  ...
5
6  *$ htpasswd -bBc htpasswd armstrong indionce
7  *$ htpasswd -b htpasswd collins veraster
8  *$ htpasswd -b htpasswd aldrin roonkere
9  *$ htpasswd -b htpasswd jobs sestiver
10 *$ htpasswd -b htpasswd wozniak glegunge
11
12 *$ oc -n openshift-config \
13     create secret generic ex280-idp-secret \
14     --from-file htpasswd=htpasswd
15
16 $ oc explain OAuth.spec
17 $ oc explain OAuth.spec.identityProviders
18 $ oc explain OAuth.spec.identityProviders.htpasswd
19 $ oc explain OAuth.spec.identityProviders.htpasswd.fileData
20
21 *$ oc edit oauth cluster
```

```
1  ...
2  spec:
3    identityProviders:
4      - ldap:
5        ...
6      # 增加 6 行
7      - htpasswd:
8        fileData:
9          name: ex280-idp-secret
10        mappingMethod: claim
11        name: ex280-htpasswd
12        type: HTTPasswd
```

!!! 注意：几分钟后，将开始重新布署后生效

```

1  *$ oc adm policy add-cluster-role-to-user cluster-admin jobs
2
3  *$ oc get clusterrolebinding -o wide | egrep 'NAME|self'
4  NAME                                ROLE                                AGE   `GROUPS`                                ...
5  `self-provisioners` ClusterRole/self-provisioner 119d `system:authenticated:oauth` ...
6
7  *$ oc adm policy add-cluster-role-to-user self-provisioner wozniak
8
9  *$ oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth
10 Warning: Your changes may get lost whenever a master is restarted, unless you prevent
    reconciliation of this rolebinding using the following command: "oc annotate
    clusterrolebinding.rbac self-provisioners
    'rbac.authorization.kubernetes.io/autoupdate=false' --overwrite"
11 ...
12
13 *$ oc annotate clusterrolebinding.rbac self-provisioner
    'rbac.authorization.kubernetes.io/autoupdate=false' --overwrite

```

```

1  *$ oc -n kube-system delete secrets kubeadmin
2

```

2. add-role-to-user

Configure your OpenShift cluster to meet the following requirements:

The following projects exist:

- ☐ apollo
- ☐ manhattan
- ☐ gemini
- ☐ bluebook
- ☐ titan

The user account **armstrong** is an **administrator** for project **apollo** and project **gemini**

The user account **wozniak** can **view** project **titan** but not administer or delete it

[opsadm@workbench]

```

1  *$ oc -n apollo adm policy add-role-to-user admin armstrong
2  *$ ^apollo^gemini
3
4  *$ oc -n titan adm policy add-role-to-user view wozniak
5

```

grade

```

1 $ oc -n apollo policy who-can delete deploy | grep -w armstrong
2     armstrong
3 $ oc -n gemini policy who-can delete deploy | grep -w armstrong
4     armstrong
5
6 $ oc -n titan policy who-can delete pod | grep -w wozniak
7 $ oc -n titan policy who-can get pod | grep -w wozniak
8     wozniak
9

```

3. user account

Configure your OpenShift cluster to meet the following requirements:

- ☐ The user account `armstrong` is a member of the `commander` group
- ☐ The user account `collins` is a member of the `pilot` group
- ☐ The user account `aldrin` is a member of the `pilot` group
- ☐ Members of the `commander` group have `edit` permission in the `apollo` project
- ☐ Members of the `pilot` group have `view` permission in the `apollo` project

[opsadm@workbench]

```

1 *$ oc adm groups new commander
2 *$ oc adm groups new pilot
3
4 *$ oc adm groups add-users commander armstrong
5 *$ oc adm groups add-users pilot collins
6 *$ oc adm groups add-users pilot aldrin
7
8 *$ oc -n apollo adm policy add-role-to-group edit commander
9 *$ oc -n apollo adm policy add-role-to-group view pilot
10

```

grade

```

1 $ oc -n apollo policy who-can patch pod | grep -w commander
2 Groups: `commander`
3
4 $ oc -n apollo policy who-can patch pod | grep -w pilot
5 $ oc -n apollo policy who-can get pod | grep -w pilot

```

4. use quotas

Configure your OpenShift cluster to use quotas in the `manhattan` project with the following requirements:

- ☐ The name of the quota is: `ex280-quota`

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

[opsadm@workbench]

```

1  *$ oc project manhattan
2
3  *$ oc create quota ex280-quota \
4      --hard=cpu=2,memory=1Gi,pods=3,services=6,replicationcontrollers=3
5

```

grade

```

1  $ oc -n manhattan describe quota ex280-quota
2  Name: ex280-quota
3  Namespace: manhattan
4  Resource          Used Hard
5  -----
6  cpu                0    2
7  memory             0    1Gi
8  pods               0    3
9  replicationcontrollers 0    3
10 services           0    6

```

5. LimitRange

Configure your OpenShift cluster to use limits in the **bluebook** project with the following requirements:

- ☐ The name of the limit range is: **ex280-limits**
- ☐ The amount of memory consumed by a single pod is between **5Mi** and **300Mi**
- ☐ The amount of memory consumed by a single container is between **5Mi** and **300Mi** with a default request of **100Mi**
- ☐ The amount of CPU consumed by a single pod is between **10m** and **500m**
- ☐ The amount of CPU consumed by a single container is between **10m** and **500m** with a default request of **100m**

[opsadm@workbench]

```

1  *$ oc project bluebook
2
3  $ oc api-resources | egrep 'NAME|limit'
4  NAME             SHORTNAMES  APIVERSION  NAMESPACE  KIND
5  limitranges      limits    `v1`        true        `LimitRange`
6  $ oc explain limitrange
7  $ oc explain limitrange.spec
8  $ oc explain limitrange.spec.limits
9
10 $ echo set nu ts=2 et sw=2 cuc paste > ~/.vimrc
11
12 *$ vim limitrange.yml

```

```

1  apiVersion: v1
2  kind: LimitRange
3  metadata:
4    name: ex280-limits
5    namespace: bluebook
6  spec:
7    limits:
8      - type: Pod
9        min:
10         memory: 5Mi
11         cpu: 10m
12        max:
13         memory: 300Mi
14         cpu: 500m
15      - type: Container
16        min:
17         memory: 5Mi
18         cpu: 10m
19        max:
20         memory: 300Mi
21         cpu: 500m
22        defaultRequest:
23         memory: 100Mi
24         cpu: 100m

```

```

1  *$ oc apply -f limitrange.yml

```

grade

```

1  $ oc -n bluebook describe limitranges
2  Name:          ex280-limits
3  Namespace:     bluebook
4  Type           Resource  Min  Max    Default Request  Default Limit  Max Limit/Request Ratio
5  ----           -
6  Pod            memory    5Mi  300Mi  -              -              -
7  Pod            cpu        10m  500m   -              -              -
8  Container      cpu        10m  500m   100m           500m           -
9  Container      memory    5Mi  300Mi  100Mi          300Mi          -

```

6. scale

Ensure that there are exactly **5** replicas of the minion application in the **gru** project

[opsadm@workbench]

```
1  *$ oc project gru
2
3  *$ oc status
4  `dc/minion` deploys registry.ocp4.example.com:8443/redhattraining/hello-world-nginx:latest
5  ...
6
7  *$ oc scale dc/minion --replicas 5
8
```

grade

```
1  $ oc get po
2  NAME                READY   STATUS    RESTARTS   AGE
3  minion-1-deploy    0/1     Completed  0           55s
4  minion-1-g6rkm     1/1     `Running`  0           54s
5  minion-1-4w7xb     1/1     `Running`  0           6s
6  minion-1-684lg     1/1     `Running`  0           6s
7  minion-1-b8k4h     1/1     `Running`  0           6s
8  minion-1-s97tp     1/1     `Running`  0           6s
```

7. Scale an application automatically

Automatically scale the hydra deployment in the **lerna** project with the following requirements:

- ☐ Minimum number of pods: **6**
- ☐ Maximum number of pods: **9**
- ☐ Target average CPU utilization per pod: **60** percent
- ☐ The pods require **25m** CPU time to operate
- ☐ The pods must not consume more than **100m** CPU time

[opsadm@workbench]

```
1  *$ oc project lerna
2
3  *$ oc status
4  `dc/hydra` deploys registry.ocp4.example.com:8443/redhattraining/hello-world-nginx:latest
5  ...
6
7  $ oc set resources -h
8  *$ oc set resources dc/hydra \
9  --limits=cpu=100m \
```

```

10     --requests=cpu=25m
11
12     $ oc autoscale -h
13 *$ oc autoscale deploymentconfig/hydra \
14     --min 6 \
15     --max 9 \
16     --cpu-percent=60
17

```

grade

```

1     $ oc get po
2     NAME                READY   STATUS    RESTARTS   AGE
3     hydra-1-deploy        0/1     Completed 0           3m31s
4     hydra-2-deploy        0/1     Completed 0           2m49s
5     hydra-2-vpwj9         1/1     `Running` 0           109s
6     hydra-2-cjp8s         1/1     `Running` 0           77s
7     hydra-2-g2jhn         1/1     `Running` 0           77s
8     hydra-2-gm2sc         1/1     `Running` 0           77s
9     hydra-2-lmmcr         1/1     `Running` 0           77s
10    hydra-2-xrvq6          1/1     `Running` 0           77s

```

8. route

Configure the `oxcart` application in the `area51` project with the following requirements:

- ☐ The application uses a secure route called `oxcart`
 - ☐ Traffic between the client and the router is `encrypted`
 - ☐ Traffic between the router and the service is `unencrypted`
 - ☐ The route uses a CA signed certificate with the following subject fields:
`/C=CN/ST=BJ/L=beijing/O=RedHat/OU=RHT/CN=classified.apps.ocp4.example.com`
 - ☐ The application is reachable only at the following address:
`https://classified.apps.ocp4.example.com`
 - ☐ The application produces output
- A utility script called `newcert` has been provided on the workbench system to create the CA signed certificate
- You may enter the certificate parameters manually or pass the subject as a parameter.

Your certificate signing request will be uploaded to the CA where it will be immediately signed and then downloaded to your current directory.

[opsadm@workbench]

```

1     *$ oc project area51
2
3     *$ newcert /C=CN/ST=BJ/L=beijing/O=RedHat/OU=RHT/CN=classified.apps.ocp4.example.com
4
5     *$ oc get route
6     NAME          HOST/PORT          PATH    SERVICES    PORT    ...

```



```

7   `oxcart` classified.apps.ocp4.example.com      `oxcart`      8080-tcp    ...
8
9   *$ oc delete route oxcart
10
11  *$ oc create route edge \
12      --service oxcart \
13      --hostname classified.apps.ocp4.example.com \
14      --key classified.apps.ocp4.example.com.key \
15      --cert classified.apps.ocp4.example.com.crt
16

```

grade

```

1   $ curl -vI https://classified.apps.ocp4.example.com
2   ...
3   * Server certificate:
4   *   subject: C=CN; ST=BJ; L=beijing; O=RedHat; OU=RHT; CN=classified.apps.ocp4.example.com
5   *   start date: Jul  2 12:23:25 2024 GMT
6   *   expire date: Jul  1 12:23:25 2029 GMT
7   *   subjectAltName: host "classified.apps.ocp4.example.com" matched cert's
   *   "*.apps.ocp4.example.com"
8   *   issuer: C=US; ST=North Carolina; L=Raleigh; O=Red Hat; CN=ocp4.example.com
9   *   `SSL certificate verify ok.`
10  ...

```

9. Deploy an application

- ☐ Deploy the chart named `ascii-movie` in the project `redhat-movie` from the repository `http://helm.ocp4.example.com/charts` name `ex280-repo`
- ☐ You may use the `telnet` or `nc` commands to validate the deployment

[opsadm@workbench]

```

1   *$ oc project redhat-movie
2
3   *$ helm repo add ex280-repo http://helm.ocp4.example.com/charts
4
5   *$ helm search repo
6   NAME                  CHART VERSION  APP VERSION  DESCRIPTION
7   ex280-repo/ascii-movie 0.16.1         1.9.3        Star Wars movie SSH and Telnet
   server
8   ...
9
10  *$ helm install redhat-movie ex280-repo/ascii-movie
11

```

grade

```

1  $ oc get all
2  NAME                                READY   STATUS    RESTARTS   AGE
3  pod/redhat-movie-ascii-movie-5b8f6548f-bpfcf  1/1     Running   0           74s
4
5  NAME                                TYPE                      CLUSTER-IP      EXTERNAL-IP
6  service/redhat-movie-ascii-movie  LoadBalancer            172.30.232.20    `192.168.50.20`
7  service/redhat-movie-ascii-movie-api ClusterIP                  172.30.240.198   <none>
8  1977/TCP                            74s
9
10 NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
11 deployment.apps/redhat-movie-ascii-movie  1/1     1             1           74s
12
13 NAME                                DESIRED   CURRENT   READY   AGE
14 replicaset.apps/redhat-movie-ascii-movie-5b8f6548f  1         1         1       74s

```

```

1  $ nc 192.168.50.20 23
2  <Ctrl-C>

```

10. Configure a secret

Configure a secret in the `math` project with the following requirements:

- ☐ The name of the secret is: `magic`
- ☐ The secret defines a key with name: `decoder_ring`
- ☐ The secret defines the key with value: `6YWN572u5q2j56GuCg==`

[opsadm@workbench]

```

1  *$ oc project math
2
3  $ oc create secret generic -h
4  *$ oc create secret generic magic \
5  --from-literal decoder_ring=6YWN572u5q2j56GuCg==
6

```

grade

```

1  $ oc extract secret/magic --to=-
2  # decoder_ring
3  6YWN572u5q2j56GuCg==

```

11. Configure an application to use a secret

Configure the application called `qed` in the `math` project with the following requirements:

- ☐ The application uses the secret previously created called: `magic`

- ☐ The secret defines an environment variable with name: `DECODER_RING`
- ☐ The application output no longer displays: `Sorry, application is not configured correctly.`

[opsadm@workbench ~]

```
1 $ oc project math
2
3 *$ oc status
4 `dc/qed` deploys registry.ocp4.example.com:8443/redhattraining/hello-world-nginx
5 ...
6
7 $ oc set env -h
8 *$ oc set env dc/qed --from=secret/magic
9
```

grade

```
1 $ oc rsh dc/qed env | grep DECODER_RING
2 DECODER_RING=XpWy9KdcP3Tr9FFHGQgZgVRCKukQdrQsbc10c2ZYhDk=
```

12. Configure a service account

Configure a service account in the `apples` project to meet the following requirements:

- ☐ The name of the service account is `ex280sa`
- ☐ The service account allows pods to be run as any available user

[opsadm@workbench ~]

```
1 *$ oc project apples
2
3 *$ oc create serviceaccount ex280sa
4 $ oc get sa
5
6 $ oc get scc
7 NAME PRIV CAPS ...
8 `anyuid` false <no value> ...
9 ...
10 *$ oc adm policy add-scc-to-user anyuid -z ex280sa
11
12 $ oc get clusterrole | grep cluster.*admin
13 `cluster-admin` YYYY-MM-DDThh:mm:ssZ
14 ...
15 *$ oc adm policy add-cluster-role-to-user cluster-admin -z ex280sa
16
```

13. uses the service account

Deploy the application called `oranges` in the `apples` project so that the following conditions are true:

- ☐ The application uses the `ex280sa` service account
- ☐ No configuration components have been added or removed
- ☐ The application produces output

[opsadm@workbench ~]

```
1  *$ oc project apples
2
3  *$ oc status
4  `dc/oranges` deploys registry.ocp4.example.com:8443/ubi9/httpd-24:latest
5  ...
6
7  *$ oc set sa dc/oranges ex280sa
8
9  $ oc get svc
10 NAME          TYPE          CLUSTER-IP          EXTERNAL-IP    PORT(S)          AGE
11 oranges       `NodePort`    `172.30.131.196`    <none>         `8080`:`31449`/TCP 28h
12
13 $ oc get no -o wide
14 NAME          STATUS  ROLES                                AGE  VERSION          INTERNAL-IP  ...
15 `master01`    Ready   control-plane,master,worker  282d  v1.25.4+77bec7a  `192.168.50.10` ...
16 $ curl 192.168.50.10:31449
17 curl: (7) Failed to connect to master01 port 30756: Connection refused
18
19 $ oc get po -o wide
20 ...
21 oranges-2-vgjkv 1/1      Running      0          8m35s    `10.8.0.220`  master01  ...
22 $ oc rsh dc/oranges curl 10.8.0.220:8080 && echo ok
23 $ oc rsh dc/oranges curl 172.30.131.196:8080 || echo no
24
25 $ oc get po --show-labels
26 NAME          READY  STATUS      RESTARTS  AGE  LABELS
27 ...
28 oranges-2-vgjkv 1/1    Running     0          12m    `deployment-
config.name=oranges`,deployment=oranges-2,deploymentconfig=oranges
29 *$ oc edit svc/oranges
```

```
1  ...
2  selector:
3    #deployment-config.name: orange
4    deployment-config.name: oranges
5  ...
```

grade

```
1 $ curl 192.168.50.10:31449
2 <html>
3   <body>
4     <h1>Hello, world from nginx!</h1>
5   </body>
6 </html>
```

14. request memory

Deploy the application called `atlas` in the `mercury` project so that the following conditions are true:

- ☐ No configuration components have been added or removed
- ☐ The application produces output

[opsadm@workbench ~]

```
1 *$ oc project mercury
2
3 *$ oc status
4 http://atlas-mercury.apps.ocp4.example.com to pod port 8080-tcp (svc/atlas)
5 deployment/atlas deploys istag/atlas:latest
6 ...
7
8 $ oc set resources -h
9 *$ oc edit deployment/atlas
```

```
1 ...
2     resources:
3     requests:
4       #memory: 10000Mi
5       memory: 128Mi
6       ...
```

grade

```
1 $ oc status
2
3 $ curl http://atlas-mercury.apps.ocp4.example.com
```

15. Configure application data

Deploy an application using the `registry.ocp4.example.com:8443/redhattraining/hello-openshift` image that meets the following requirements:

- ☐ The application is part of a project named: `acid`
- ☐ The application is named: `phosphoric`
- ☐ The application uses a key named `RESPONSE` in a configuration map named `sedicen`

- ☐ The application is running and available at <http://phosphoric-acid.apps.ocp4.example.com> and displays the following initial text:
`Soda pop won't stop can't stop`
- ☐ Re-deploying the application after making changes to the configuration map results in a corresponding change to the displayed text

[opsadm@workbench ~]

```
1  *$ oc project acid
2
3  *$ oc create configmap sedicen \
4      --from-literal RESPONSE="Soda pop won't stop can't stop"
5
6  *$ oc new-app \
7      --name phosphoric \
8      --image registry.ocp4.example.com:8443/redhattraining/hello-openshift
9
10 *$ oc set env deployment/phosphoric --from=configmap/sedicen
11
12 *$ oc expose svc/phosphoric \
13     --hostname=phosphoric-acid.apps.ocp4.example.com
```

grade

```
1  $ curl phosphoric-acid.apps.ocp4.example.com
2  Soda pop won't stop can't stop
```

16. network policy

Configure a network policy using the `database` and `checker` projects with the following requirements:

- ☐ The `database` project has network policy with the name `db-allow-mysql-conn` based on pod selector label `network.openshift.io/policy-group`
- ☐ Connections to the `database` project are restricted to deployments from the `checker` project
- ☐ The network policy is filtered by project `selector` using the `team=devsecops` label and pod selector using the `deployment=web-mysql` label
- ☐ The application can establish a connection to port `3306/TCP`

You can check your work by examining the logs in the `checker` project

[opsadm@workbench ~]

```

1  *$ oc project database
2
3  *$ oc get networkpolicies
4  NAME          POD-SELECTOR  AGE
5  db-deny-all   <none>       115s
6
7  *$ vim db-networkpolicy.yaml

```

```

1  apiVersion: networking.k8s.io/v1
2  kind: NetworkPolicy
3  metadata:
4    name: db-allow-mysql-conn
5    namespace: database
6  spec:
7    podSelector:
8      matchLabels:
9        network.openshift.io/policy-group: database
10   policyTypes:
11     - Ingress
12   ingress:
13     - from:
14       - podSelector:
15         matchLabels:
16           deployment: web-mysql
17       namespaceSelector:
18         matchLabels:
19           team: devsecops
20   ports:
21     - protocol: TCP
22       port: 3306

```

```

1  *$ oc apply -f db-networkpolicy.yaml
2

```

grade

```

1  $ POD_IP=$(oc get pod -n database -o jsonpath='{.items[0].status.podIP}')
2
3  $ oc -n checker rsh deployments/test \
4    mysql -h ${POD_IP} -uroot -predhat -e "show databases;"
5  +-----+
6  | Database          |
7  +-----+
8  | information_schema |
9  | mysql             |
10 | performance_schema |
11 | sys                |
12 +-----+

```

17. template

Configure your OpenShift cluster so that new projects are created with limits using the following requirements:

- ☐ The name of the limit range is: `PROJECT_NAME-limits` where `PROJECT_NAME` is the name of the project created using `oc new-project`
- ☐ The amount of memory consumed by a single container is between `128Mi` and `1Gi` with a default of `512Mi` and a default request of `256Mi`

[opsadm@workbench ~]

```
1 $ oc adm create-bootstrap-project-template -h
2 *$ oc adm create-bootstrap-project-template -o yaml > 17.yaml
3
4 $ oc get limitranges -A
5 $ oc get limitranges -A -o yaml | grep -v cpu
6
7 *$ vim 17.yaml
```

```
1  apiVersion: template.openshift.io/v1
2  kind: Template
3  metadata:
4    name: project-request
5    # 增加 1 行
6    namespace: openshift-config
7  objects:
8  - apiVersion: project.openshift.io/v1
9    kind: Project
10   metadata:
11     annotations:
12       openshift.io/description: ${PROJECT_DESCRIPTION}
13       openshift.io/display-name: ${PROJECT_DISPLAYNAME}
14       openshift.io/requester: ${PROJECT_REQUESTING_USER}
15     name: ${PROJECT_NAME}
16   spec: {}
17 - apiVersion: rbac.authorization.k8s.io/v1
18   kind: RoleBinding
19   metadata:
20     name: admin
21     namespace: ${PROJECT_NAME}
22   roleRef:
23     apiGroup: rbac.authorization.k8s.io
24     kind: ClusterRole
25     name: admin
26   subjects:
27   - apiGroup: rbac.authorization.k8s.io
28     kind: User
29     name: ${PROJECT_ADMIN_USER}
30 # >>>> 添加 BEGIN
31 - apiVersion: v1
32   kind: LimitRange
```



```

33     metadata:
34         # The name of the limit range
35         name: ${PROJECT_NAME}-limits
36         namespace: ${PROJECT_NAME}
37     spec:
38         limits:
39         # a single container
40         - type: Container
41           min:
42             memory: 128Mi
43           max:
44             memory: 1Gi
45           default:
46             memory: 512Mi
47           defaultRequest:
48             memory: 256Mi
49     # <<<< 添加 END
50     parameters:
51     - name: PROJECT_NAME
52     - name: PROJECT_DISPLAYNAME
53     - name: PROJECT_DESCRIPTION
54     - name: PROJECT_ADMIN_USER
55     - name: PROJECT_REQUESTING_USER

```

```

1  *$ oc apply -f 17.yml
2
3  *$ oc api-resources | egrep -iw 'name|project'
4  $ oc explain -h
5  *$ oc explain --api-version=config.openshift.io/v1 project
6  *$ oc explain --api-version=config.openshift.io/v1 project.spec
7  *$ oc explain --api-version=config.openshift.io/v1 project.spec.projectRequestTemplate
8
9  *$ oc edit projects.config.openshift.io cluster

```

```

1  ...
2  #spec: {}
3  spec:
4      projectRequestTemplate:
5          name: project-request

```

grade

```

1  $ watch oc get pod -n openshift-apiserver
2  NAME                                READY   STATUS    RESTARTS   AGE
3  apiserver-5774cb6f8-j2ndh           2/2    Running   0          19m
4  <Ctrl+C>
5
6  $ oc new-project test
7
8  $ oc get limitranges
9  NAME              CREATED AT
10 test-limits       YYYY-MM-DDThh:mm:ssZ

```

18. operator

Install the file-integrity operator with the following requirements:

- ☐ The operator is installed in the `openshift-file-integrity` project
- ☐ The approval strategy is `Automatic`
- ☐ Cluster monitoring is enabled for the openshift-file-integrity project

[opsadm@workbench ~]

```
1 $ oc whoami --show-console
2 https://console-openshift-console.apps.ocp4.example.com
```

 Firefox <https://console-openshift-console.apps.ocp4.example.com>

Operators / operatorHub /

 file Enter /



File Integrity Operator / Install /

Installation mode *

- ☐ A specific namespace on the cluster

Installed Namespace *

- ☐ Operator recommended Namespace: Project: openshift-file-integrity
- ☒ Enable Operator recommended cluster monitoring on this Namespace

Update approval *

- ☐ Automatic

19. cron job

Create a cron job using the image at

`registry.ocp4.example.com:8443/redhattraining/hello-world-nginx` with the following requirements:

- ☐ The cron job name is `job-runner`
- ☐ The cron job runs at `04:05` on the `2nd` day of `every month`
- ☐ The successful job history limit is `14`
- ☐ The service account and service account name is `magna`
- ☐ The cron job runs in the project called `elementum`

[opsadm@workbench ~]

```
1 *$ oc new-project elementum
2
```

```

3  *$ oc create sa magna
4
5  $ oc explain cronjob.spec | grep -i succ
6      successfulJobsHistoryLimit <integer>
7      ...
8  *$ oc create cronjob job-runner \
9      --image=registry.ocp4.example.com:8443/redhattraining/hello-world-nginx \
10     --schedule="5 4 2 * *" \
11     --dry-run=client \
12     -o yaml > 19.yaml
13
14  *$ vim 19.yaml

```

```

1  apiVersion: batch/v1
2  kind: CronJob
3  metadata:
4      name: job-runner
5      # 增加 1 行 [可选]
6      namespace: elementum
7  spec:
8      # 增加 1 行
9      successfulJobsHistoryLimit: 14
10     jobTemplate:
11         metadata:
12             name: job-runner
13         spec:
14             template:
15                 spec:
16                     serviceAccountName: magna
17                     containers:
18                         - image: registry.ocp4.example.com:8443/redhattraining/hello-world-nginx
19                           name: job-runner
20                           resources: {}
21                     restartPolicy: OnFailure
22     schedule: 5 4 2 * *
23     status: {}

```

```

1  *$ oc create -f 19.yaml
2
3  *$ oc set sa cronjob/job-runner magna
4

```

grade

```

1  $ oc get all
2  NAME                                SCHEDULE    SUSPEND    ACTIVE    LAST SCHEDULE    AGE
3  cronjob.batch/job-runner            5 4 2 * *   False      0          <none>       58s

```

20. Collect the default support information

Collect the default support information for your OpenShift cluster with the following requirements:

- ☐ The data is stored as a compressed tar archive using: `tar cvaf`
- ☐ The name of the compressed tar archive is: `ex280-ocp-clusterID.tar.gz`

where clusterID is the unique identifier of your OpenShift cluster

The archive has been uploaded for grading

A utility script has been provided for you to upload the archive as follows:

```
/usr/local/bin/upload-cluster-data ex280-ocp-clusterID.tar.gz
```

You may upload the archive as many times as necessary. Each uploaded archive will overwrite any previously uploaded archive.

[opsadm@workbench ~]

```
1  *$ oc adm must-gather
2  ...
3  ClusterID: `b1d661ca-7fb3-42e2-a62a-968b80672189`
4  ClusterVersion: Stable at "4.14.0"
5  ClusterOperators:
6    All healthy and stable
7
8  *$ tar cvaf ex280-ocp-b1d661ca-7fb3-42e2-a62a-968b80672189.tar.gz must<Tab>
9
10 *$ /usr/local/bin/upload-cluster-data ex280-ocp-b1d661ca-7fb3-42e2-a62a-968b80672189.tar.gz
11
```

21. A storage class has been configured to provide NFS storage

Using information from that storage class, configure a persistent volume with the following requirements:

- ☐ Name: `landing-pv`
- ☐ Access mode: `ReadOnlyMany`
- ☐ Size: `1Gi`
- ☐ The reclaim policy matches the storage class

Configure a persistent volume claim with the following requirements:

- ☐ Name: `landing-pvc`
- ☐ The access mode is the same as the persistent volume
- ☐ The size is the same as the persistent volume

Deploy the application with the following requirements:

- ☐ The application exists in a project called `page`

- ☐ The application uses a deployment called `landing`
- ☐ The application uses the image hosted at `registry.ocp4.example.com:8443/redhattraining/hello-world-nginx`
- ☐ The nginx mountpoint is `/usr/share/nginx/html`
- ☐ The application uses `3` pods
- ☐ The application is accessible at `https://landing-page.apps.ocp4.example.com`

PS: 建议使用  Firefox 网页 完成

[opsadm@workbench ~]

```
1  $ oc get storageclasses
2  *$ oc get storageclasses nfs-storage -o yaml
3  reclaimPolicy: `Delete`
4  ...
5
6  $ oc get po -A | grep nfs
7  $ oc -n nfs-client-provisioner get all
8  *$ oc -n nfs-client-provisioner get deployment/nfs-client-provisioner -o yaml
9  ...
10         nfs:
11             path: `/exports-ocp4`
12             server: `192.168.50.254`
13
14  *$ vim nfs-pv.yaml
```

```
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4      name: landing-pv
5  spec:
6      accessModes:
7          - ReadOnlyMany
8      capacity:
9          storage: 1Gi
10     storageClassName: nfs-storage
11     #storageClassName: nfs-client
12     persistentVolumeReclaimPolicy: Delete
13     nfs:
14         path: /exports-ocp4
15         server: 192.168.50.254
16     #path: /nfsshare
17     #server: workstation.ocp4.example.com
```

```
1  *$ oc apply -f nfs-pv.yaml
2
3  *$ oc project page
4
5  *$ vim nfs-pvc.yaml
```

```

1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: landing-pvc
5  spec:
6    accessModes:
7      - ReadOnlyMany
8    resources:
9      requests:
10       storage: 1Gi
11    storageClassName: nfs-storage
12    volumeName: landing-pv

```

```

1  *$ oc apply -f nfs-pvc.yml
2
3  *$ oc new-app \
4    --name landing \
5    --image registry.ocp4.example.com:8443/redhattraining/hello-world-nginx
6
7  *$ oc set volumes deployment/landing \
8    --add \
9    --name web-volume \
10   --type pvc \
11   --claim-name landing-pvc \
12   -m /usr/share/nginx/html
13
14  *$ oc scale deployment/landing --replicas 3
15
16  *$ rm -r classified*
17  *$ newcert /C=CN/ST=BJ/L=beijing/O=RedHat/OU=RHT/CN=landing-page.apps.ocp4.example.com
18
19  *$ oc create route edge landing \
20    --service=landing \
21    --hostname=landing-page.apps.ocp4.example.com \
22    --key classified.apps.ocp4.example.com.key \
23    --cert classified.apps.ocp4.example.com.crt
24

```

grade

```
1  $ curl -vI https://landing-page.apps.ocp4.example.com
```

22. liveness probe

An application named **atlas** has been deployed with a single container in the **mercury** project

Implement a **liveness** probe for this container that meets the following requirements:

- ☐ The probe monitors liveness by performing a **TCP** socket check on port **8080**
- ☐ The probe has an initial delay of **10** seconds and a timeout of **30** seconds

☐ Your changes can survive a rebuild

PS: 建议使用  Firefox 网页 完成

[opsadm@workbench ~]

```
1  *$ oc project mercury
2
3  *$ oc status
4    `deployment/atlas` deploys istag/atlas:latest
5    ...
6
7  $ oc set probe --help
8  *$ oc set probe deployment/atlas \
9    --liveness \
10   --open-tcp=8080 \
11   --initial-delay-seconds=10 \
12   --timeout-seconds=30
13
```

0. OBJECTIVE

SCORE

- ☐ Manage OpenShift Container Platform
- ☐ Deploy applications
- ☐ Manage storage for application configuration and data
- ☐ Configure applications for reliability
- ☐ Manage authentication and authorization
- ☐ Configure network security
- ☐ Enable developer self-service
- ☐ Manage OpenShift operators
- ☐ Configure application security