

EX 280 v12

Exam Instructions:

You will have 1 VMs in the exam

3 Hours Exam Cluster is already configured, and all required packages are installed.

In case you need any package, you can install with preconfigured repositories.

Docs are available in html and pdf format

Cluster Web Console and API url will be given in the

exam:

<https://console.domain3.example.com:6443>

<https://api.domain3.example.com:6443>

You have one user opsadm in workbecnch.example.com is given in exarm password of opsadm is given in instructions

{ use ssh to access the workbench }

**Password of kubeadm is stored in file in workbench “
/home/opsadm/kubeadminpassword.txt”**

All the configuration tools to run OpenShift cluster already available in workbench.

All Docs are available in html and pdf format

**After entering in exam open a terminal and do ssh by root on workstation by given password.
one common password will be given in the exam, we can use same password to access
workstation VM as root.**

Q1) Manage Identity Providers:

- configure the Oauth to use HTPasswd as the identity provider
- secret name should be ex280-secure and Identity provider name should be ex280-idp-secure
- Create user bob with password indionce
- Create user qwerty with password catalog
- Create user john with password john123
- Create user armstrong with password natasha123
- create user natasha with password arthstrong
- Create user harry with password harry123
- Create user susan with password susuan123

SOLUTION

```
student@workstation:~$ File Edit View Search Terminal Help
[student@workstation ~]$ # adding users
[student@workstation ~]$ htpasswd -B -b /tmp/htpasswd susan redhat
Updating password for user susan
[student@workstation ~]$ 
[student@workstation ~]$ cat /tmp/htpasswd
bob:$2y$05$ttEniueyDkiy30TjATP4Z01PqBoIHcYnrA.CQEh7e5XyL2B8GKTjq
qwerty:$2y$05$MkIpRB.eBnugvmLwUYfKOSED/gMg4.DtGbfZ2lyd1hv.rClPzby.
john:$2y$05$EZzoKe3xsNKhkxIX5KfL5.EknYVWiNZ0TEyg.xrunUPkT1.Oh0IjG
armstrong:$2y$05$g9EEB/poQJv01IqsZRxab09bA9Kpf.iGfd62GCLDv6XE Fot.owejW
natasha:$2y$05$imcZydRrPNMKJVU1QVEm0QIaEppy/UwLI Pa09fdgjcIdCibkbbFG
harry:$2y$05$4YNu9uzk2uCYJm7C.ubglu7ZE2w1PJQR70n8A6fbZ8iIDkU7ETu2a
susan:$2y$05$WhNzbsQVaV.mB/hLywGume4qzAtq0PYyTycnPBnvM5EYdpNNxDB2
[student@workstation ~]$
```

```
[student@workstation ~]$ # create secret
[student@workstation ~]$ 
[student@workstation ~]$ oc create secret generic ex280-secure --from-file htpasswd=/tmp/htpasswd -n openshift-config
```

>>>> oc edit oauth cluster

```
resourceVersion: "70802"
uid: 5177eecf-0fcf-48c0-b0cb-e2e3e7bd5f64
spec:
  identityProviders:
    - htpasswd:
        fileData:
          name: ex280-secure
        mappingMethod: claim
        name: ex280-idp-secure
        type: HTPasswd
```

need to remember full spec entry

new pods created

```
File Edit View Search Terminal Help
[student@workstation ~]$ oc edit oauth cluster
Edit cancelled, no changes made.
[student@workstation ~]$ oc get pods -n openshift-authentication
NAME          READY   STATUS    RESTARTS   AGE
oauth-openshift-656cbddb57-nqhj4  1/1     Terminating   1      356d
oauth-openshift-696db68646-56qm4  1/1     Running     0      40s
oauth-openshift-696db68646-dfzms  1/1     Running     0      71s
oauth-openshift-696db68646-tpjn7  0/1     Pending      0      10s
[student@workstation ~]$
```

login with every user

```
File Edit View Search Terminal Help
[student@workstation ~]$ oc login -u bob -p redhat
Login successful.

You don't have any projects. You can try to create a new project, by running
  oc new-project <projectname>

[student@workstation ~]$ oc whoami
bob
[student@workstation ~]$
```

back to kubeadmin user

```
Using project "default".
[student@workstation ~]$ oc get users
NAME          UID          FULL NAME  IDENTITIES
admin         835bde25-34d7-47a8-a9d5-0875a846ab23
armstrong    415abead-f4c5-44be-ae9f-c7215b3516cd
bob          4f72a9f0-f982-4818-bfe0-bd7e663553bc
harry         46e75607-40b2-495e-ac5d-45e312581d3f
john          430a2497-ba6f-467b-a1b1-65e5dedaefcd
natasha       98b1de65-65f5-458e-9532-154d605e2dc9
qwerty        56007198-554c-4b07-b288-bfe1ea9c8819
susan          c7a7baef-df40-48c8-b180-475fb7c46571
[student@workstation ~]$
```

Q2) Manage Cluster Project and Permissions:

- Create project with named apollo,test,demo
- bob user should have cluster administrator rights
- john user can create projects
- qwerty user can not create projects-
- natasha user can only view the resources of apollo and test project
- armstrong user should have admin access to apollo project.
- kubeadmin user should not exist (remove kubeadmin user)

```
[student@workstation ~]$ oc new-project apollo
Now using project "apollo" on server "https://api.ocp4.example.com:6443".
You can add applications to this project with the 'new-app' command. For example, try:
  oc new-app rails-postgresql-example
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
  kubectl create deployment hello-node --image=k8s.gcr.io/e2e-test-images/agnhost:2.33 -- /agnhost serve-hostname
```

```
[student@workstation ~]$ oc new-project test
Now using project "test" on server "https://api.ocp4.example.com:6443".
You can add applications to this project with the 'new-app' command. For example, try:
  oc new-app rails-postgresql-example
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
  kubectl create deployment hello-node --image=k8s.gcr.io/e2e-test-images/agnhost:2.33 -- /agnhost serve-hostname
```

```
[student@workstation ~]$ oc new-project demo
Now using project "demo" on server "https://api.ocp4.example.com:6443".
You can add applications to this project with the 'new-app' command. For example, try:
  oc new-app rails-postgresql-example
to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:
  kubectl create deployment hello-node --image=k8s.gcr.io/e2e-test-images/agnhost:2.33 -- /agnhost serve-hostname
```

```
[student@workstation ~]$ oc adm policy add-cluster-role-to-user cluster-admin
bob
clusterrole.rbac.authorization.k8s.io/cluster-admin added: "bob"
[student@workstation ~]$ █
```

```
[student@workstation ~]$ # to check bob user gets rights
[student@workstation ~]$ 
[student@workstation ~]$ 
[student@workstation ~]$ oc login -u bob -p redhat
Login successful.
```

```
You have access to 69 projects, the list has been suppressed. You can list all projects with 'oc projects'
```

```
Using project "demo".
```

```
[student@workstation ~]$ oc get nodes
NAME      STATUS    ROLES      AGE      VERSION
master01   Ready     master,worker  354d    v1.23.3+e419edf
master02   Ready     master,worker  354d    v1.23.3+e419edf
master03   Ready     master,worker  354d    v1.23.3+e419edf
[student@workstation ~]$
```

```
[student@workstation ~]$ oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth
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
clusterrole.rbac.authorization.k8s.io/self-provisioner removed: "system:authenticated:oauth"
[student@workstation ~]$ oc login -u qwerty -p redhat
Login successful.
```

```
You don't have any projects. Contact your system administrator to request a project.
```

```
[student@workstation ~]$ oc new-project deepak
Error from server (Forbidden): You may not request a new project via this API
[student@workstation ~]$ █
```

```
[student@workstation ~]$ # jhon can create projects
[student@workstation ~]$
[student@workstation ~]$
[student@workstation ~]$ oc login -u bob -p redhat
Login successful.

You have access to 69 projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project "default".
[student@workstation ~]$ oc adm policy add-cluster-role-to-user self-provisioner jhon
Warning: User 'jhon' not found
clusterrole.rbac.authorization.k8s.io/self-provisioner added: "jhon"
[student@workstation ~]$ oc adm policy add-cluster-role-to-user self-provisioner john
clusterrole.rbac.authorization.k8s.io/self-provisioner added: "john"
[student@workstation ~]$ oc login -u jhon -p redhat
Login failed (401 Unauthorized)
Verify you have provided correct credentials.
[student@workstation ~]$ oc login -u john -p redhat
Login successful.

You don't have any projects. You can try to create a new project, by running
  oc new-project <projectname>
```

```
[student@workstation ~]$ # natashs can view resources of appolo , test
[student@workstation ~]$
[student@workstation ~]$
[student@workstation ~]$ oc login -u bob -p redhat
Login successful.

You have access to 69 projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project "default".
[student@workstation ~]$ oc policy add-role-to-user view natasha -n appolo
Error from server (NotFound): namespaces "appolo" not found
[student@workstation ~]$ oc policy add-role-to-user view natasha -n apollo
clusterrole.rbac.authorization.k8s.io/view added: "natasha"
[student@workstation ~]$ oc policy add-role-to-user view natasha -n test
clusterrole.rbac.authorization.k8s.io/view added: "natasha"
[student@workstation ~]$ ■
```

```
[student@workstation ~]$ # armstrong is admin of apollo
[student@workstation ~]$
[student@workstation ~]$
[student@workstation ~]$
[student@workstation ~]$ oc policy add-role-to-user admin armstrong apollo
Warning: User 'apollo' not found
clusterrole.rbac.authorization.k8s.io/admin added: ["armstrong" "apollo"]
[student@workstation ~]$ oc policy add-role-to-user admin armstrong -n apollo
clusterrole.rbac.authorization.k8s.io/admin added: "armstrong"
[student@workstation ~]$ ■
```

```
[student@workstation ~]$ # delete kubeadmin user
[student@workstation ~]$
[student@workstation ~]$ oc delete secrets kubeadmin -n kube-system
secret "kubeadmin" deleted
[student@workstation ~]$ ■
```

Q3) Managing Groups:

- Create groups with name site-users and guest-users
- add qwerty user in guest-users group
- add harry and susan users in site-users
- Give edit permission to sites-users groups on test project
- Give view permission to guest-users groups on demo project

```
[student@workstation ~]$ # create groups
[student@workstation ~]$
[student@workstation ~]$ oc adm groups new site-users
group.user.openshift.io/site-users created
[student@workstation ~]$ oc adm groups new guest-users
group.user.openshift.io/guest-users created
[student@workstation ~]$
```

```
[student@workstation ~]$ oc adm groups add-users guest-users qwerty
group.user.openshift.io/guest-users added: "qwerty"
[student@workstation ~]$ oc adm groups add-users site-users harry susan
group.user.openshift.io/site-users added: ["harry" "susan"]
[student@workstation ~]$ oc get group
NAME      USERS
guest-users  qwerty
site-users   harry, susan
[student@workstation ~]$
```

```
student@workstation:~ [student@workstation ~]$ oc policy add-role-to-group edit site-users -n test
clusterrole.rbac.authorization.k8s.io/edit added: "site-users"
[student@workstation ~]$ oc policy add-role-to-group view guest-users -n demo
clusterrole.rbac.authorization.k8s.io/view added: "guest-users"
[student@workstation ~]$
```

```
student@workstation:~ [student@workstation ~]$ oc get rolebinding -n test
NAME      ROLE          AGE
admin     ClusterRole/admin  54m
edit      ClusterRole/edit  5m17s
system:deployers ClusterRole/system:deployer  54m
system:image-builders ClusterRole/system:image-builder  54m
system:image-pullers ClusterRole/system:image-puller  54m
view      ClusterRole/view  22m
[student@workstation ~]$ oc get rolebinding -n demo
NAME      ROLE          AGE
admin     ClusterRole/admin  54m
system:deployers ClusterRole/system:deployer  54m
system:image-builders ClusterRole/system:image-builder  54m
system:image-pullers ClusterRole/system:image-puller  54m
view      ClusterRole/view  4m49s
[student@workstation ~]$
```

Q4) Create resource quota for project rocky:

- pods = 3
- cpu = 2
- services = 6
- memory = 1Gi
- secrets = 6
- replication controllers = 6

```
student@workstation:~  
[student@workstation ~]$ oc project rocky  
Already on project "rocky" on server "https://api.ocp4.example.com:6443".  
[student@workstation ~]$  
[student@workstation ~]$  
[student@workstation ~]$ oc create quota my-quota --hard(cpu=2,memory=1G,pods  
=3,services=6,replicationcontrollers=6,secrets=6  
resourcequota/my-quota created  
[student@workstation ~]$  
[student@workstation ~]$  
[student@workstation ~]$ oc get quota  
NAME      AGE      REQUEST           LIMIT  
my-quota  11s      cpu: 0/2, memory: 0/1G, pods: 0/3, replicationcontrollers: 0  
/6, secrets: 9/6, services: 0/6  
[student@workstation ~]$
```

Q5) Create LimitRange for project darpa:

- Set the pod memory limit between 5Mi and 300Mi
- Set the container memory limit between 5Mi and 300Mi and container default request limit for memory is 100Mi
- Set the pod cpu limit between 5m and 300m
- Set the container cpu limit between 5m and 300m and container default request limit for cpu is 100m

```
student@workstation ~]$ vim limit.yaml
[student@workstation ~]$ vim limit.yaml
[student@workstation ~]$ oc create -f limit.yaml
limitrange/dev-limit created
[student@workstation ~]$ oc get limitranges
NAME      CREATED AT
dev-limit  2023-07-18T09:09:28Z
[student@workstation ~]$ oc describe limitranges
Name:      dev-limit
Namespace: darpa
Type       Resource  Min   Max     Default Request  Default Limit  Max Limit/R
equest Ratio
-----
Pod        cpu      5m    300m    -           -           -
Pod        memory   5Mi   300Mi   -           -           -
Container  cpu      5m    300m   100m       100m       -
Container  memory   5Mi   300Mi   100Mi     100Mi     -
[student@workstation ~]$
```

first go to project darpa

> oc project darpa
the create file limits.yaml

vim limits.yaml {{ need to write full file }}

```
student@workstation:~$ vim limits.yaml
apiVersion: v1
kind: LimitRange
metadata:
  name: dev-limit
  namespace: darpa
spec:
  limits:
    - type: Pod
      max:
        cpu: 300m
        memory: 300Mi
      min:
        cpu: 5m
        memory: 5Mi
    - type: Container
      max:
        cpu: 300m
        memory: 300Mi
      min:
        cpu: 5m
        memory: 5Mi
  default:
    cpu: 100m
    memory: 100Mi
```

Q6) Scale Application Manually:

- scale the single-pod replicas to 5 under the project world and make sure all pods should run successfully.

```
student@workstation:~$ oc get deployment
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
httpd    1/1     1           1          18m
[student@workstation ~]$ oc scale deployment httpd --replicas=5
deployment.apps/httpd scaled
[student@workstation ~]$ oc get pods
NAME      READY  STATUS      RESTARTS  AGE
httpd-568dd846c7-2nbwc  0/1   ContainerCreating  0         9s
httpd-568dd846c7-2v8rz  1/1   Running       0         18m
httpd-568dd846c7-8kgjw  0/1   ContainerCreating  0         9s
httpd-568dd846c7-qhgs2  1/1   Running       0         9s
httpd-568dd846c7-qw8hv  0/1   ContainerCreating  0         9s
[student@workstation ~]$ oc get deployment
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
httpd    5/5     5           5          19m
[student@workstation ~]$
```

Q7) Autoscale of Pods in scaling project:

- minimum replicas=2, maximum replicas=9 and cpu percentage=60%
- Default request for container memory should 100Mi and cpu 50m

for practice create oc new-project scaling

> oc project scaling

```
student@workstation:~$ oc get deployment
NAME      READY  UP-TO-DATE  AVAILABLE  AGE
httpd    1/1     1           1          18s
[student@workstation ~]$
```

```
student@workstation:~$ oc set resources deployment httpd --requests=cpu=50m
,memory=100Mi
deployment.apps/httpd resource requirements updated
[student@workstation ~]$ oc get pods
NAME      READY  STATUS      RESTARTS  AGE
httpd-6fb69fb454-pnn2f  1/1   Running     0         11s
[student@workstation ~]$ oc autoscale deployment httpd --min=2 --max=9 --cpu-
percentage=60
error: unknown flag: --cpu-percentage
See 'oc autoscale --help' for usage.
[student@workstation ~]$ oc autoscale deployment httpd --min=2 --max=9 --cpu-
percent=60
horizontalpodautoscaler.autoscaling/httpd autoscaled
[student@workstation ~]$ oc get hpa
NAME      REFERENCE      TARGETS      MINPODS  MAXPODS  REPLICAS  AGE
httpd    Deployment/httpd  <unknown>/60%  2        9         1         21s
[student@workstation ~]$
```

Q8) Create secret with named magic in math project

- The key name should be dirlong
- The value of key should be asdf234234=

<<< this question is random >>>>

example question

Q8) Create secret with named magic in monday project

- The key name should be MYSQL_ROOT_PASSWORD
- The value of key should be redhat

solution:-

```
oc create secret generic monday --from-literal MYSQL_ROOT_PASSWORD=redhat -n monday  
oc get secret -n monday
```

solution ---

Go to project monday

The screenshot shows a terminal window titled "student@workstation:~". The terminal has a dark background and light-colored text. It displays the following commands and output:

```
student@workstation:~$ oc create secret generic monday --from-literal MYSQL_ROOT_PASSWORD=redhat -n monday  
secret/monday created  
[student@workstation ~]$ oc get secrets -n monday  
NAME          TYPE           DATA  AGE  
builder-dockercfg-79vgz  kubernetes.io/dockercfg  1     26m  
builder-token-fkdqq    kubernetes.io/service-account-token  4     26m  
builder-token-qwzd6    kubernetes.io/service-account-token  4     26m  
default-dockercfg-95ssx kubernetes.io/dockercfg  1     26m  
default-token-7l94w     kubernetes.io/service-account-token  4     26m  
default-token-mn8jd    kubernetes.io/service-account-token  4     26m  
deployer-dockercfg-28fd4 kubernetes.io/dockercfg  1     26m  
deployer-token-gxkph   kubernetes.io/service-account-token  4     26m  
deployer-token-p8gpx   kubernetes.io/service-account-token  4     26m  
monday                Opaque          1     15s  
[student@workstation ~]$
```

same with 9th question

Q9) Use Secret in monday project

- There is one pod already exists.
- it should use magic secret previously created
- Application Should Produce Output.

don't run this command in exam it is needed to create a exam env on your labs == (oc new-app --name monday --docker-image registry.access.redhat.com/rhscl/mysql-57-rhel7:5.7-47)

solution:-

oc project monday

oc get pods

oc get deployment (there can be deployment also not dc)

oc get dc (there can be dc also not deployment)

oc logs <podname>

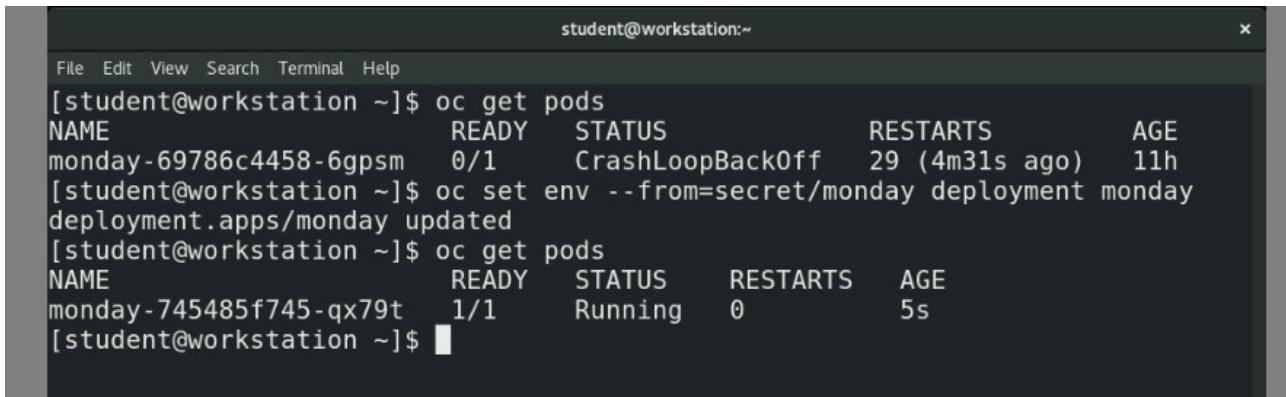
```
oc get events
```

due to env variable the application will not be running. if you have set the variable afn after that also if application is not working than you can see the logs and events for further reasons. may be on worker node taint is applied. if it is than remove the taint from worker node only.

HOW TO SET THE ENV IN DEPLOYMENT WITH SECRET.

```
oc set env --help | less
```

```
oc set env --from=secret/monday deployment monday
```



```
student@workstation:~  
File Edit View Search Terminal Help  
[student@workstation ~]$ oc get pods  
NAME READY STATUS RESTARTS AGE  
monday-69786c4458-6gpsm 0/1 CrashLoopBackOff 29 (4m31s ago) 11h  
[student@workstation ~]$ oc set env --from=secret/monday deployment monday  
deployment.apps/monday updated  
[student@workstation ~]$ oc get pods  
NAME READY STATUS RESTARTS AGE  
monday-745485f745-qx79t 1/1 Running 0 5s  
[student@workstation ~]$ █
```

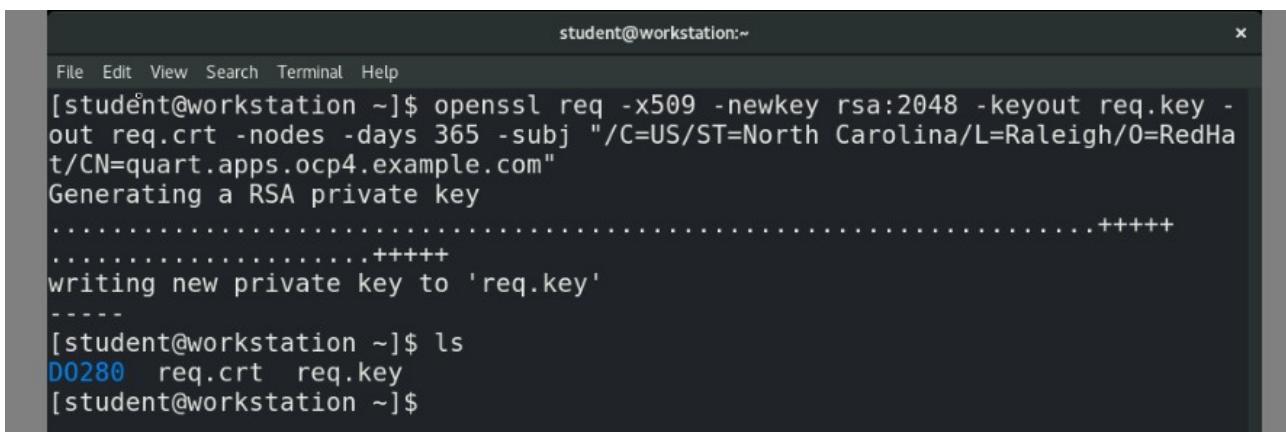
A terminal window titled "student@workstation:~". The window shows a sequence of commands being run in a Bash shell. It starts with "oc get pods", followed by "oc set env --from=secret/monday deployment monday", which updates the deployment. Finally, it runs "oc get pods" again to show the updated state of the pods.

Q10) Create Secure Route in quart project

- One application is already running named with hello based on http and route htquart
- It should run on https with self-signed certificate.
- Subject is "/C=US/ST=North Carolina/L=Raleigh/O=RedHat/CN=quart.apps.ocp4.example.com"
- It should run on https with following url: https://quart.apps.domain3.example.com
- Application should produce output.

Solutions

> create certificates



```
student@workstation:~  
File Edit View Search Terminal Help  
[student@workstation ~]$ openssl req -x509 -newkey rsa:2048 -keyout req.key -  
out req.crt -nodes -days 365 -subj "/C=US/ST=North Carolina/L=Raleigh/O=RedHa  
t/CN=quart.apps.ocp4.example.com"  
Generating a RSA private key  
.....+  
.....++  
writing new private key to 'req.key'  
----  
[student@workstation ~]$ ls  
D0280 req.crt req.key  
[student@workstation ~]$
```

A terminal window titled "student@workstation:~". The window shows the execution of the "openssl req" command to generate a self-signed certificate. It specifies an RSA key of 2048 bits, sets the subject to "/C=US/ST=North Carolina/L=Raleigh/O=RedHat/CN=quart.apps.ocp4.example.com", and creates files "req.key" and "req.crt". The "ls" command at the end lists these files.

>> now create route

```
student@workstation:~  
File Edit View Search Terminal Help  
[student@workstation ~]$ oc get svc  
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE  
httpd    ClusterIP  172.30.207.110  <none>        8080/TCP,8443/TCP  23m  
[student@workstation ~]$ oc create route edge htquart --service=httpd --hostname=quart.apps.ocp4.example.com --cert ./req.crt --key ./req.key  
route.route.openshift.io/htquart created  
[student@workstation ~]$ oc get route  
NAME      HOST/PORT      PATH      SERVICES      PORT      TEMPLATES  
RMINATION  WILDCARD  
htquart   quart.apps.ocp4.example.com      httpd      8080-tcp      ed  
ge        None  
httpd     httpd-quart.apps.ocp4.example.com      httpd      8080-tcp  
          None  
[student@workstation ~]$
```

Q11) Create Service Account(user) in qed:

- Service Account (user) Should be project1-sa
- Service Account should be associated with anyuid SCC

Solution :-

```
oc project qed  
oc create serviceaccount --help | less  
oc create serviceaccount project1-sa -n qed  
oc adm policy --help | less  
oc adm policy add-scc-to-user --help | less  
oc adm policy add-scc-to-user anyuid -z project1-sa
```

```
student@workstation:~  
File Edit View Search Terminal Help  
[student@workstation ~]$ oc project qed  
Already on project "qed" on server "https://api.ocp4.example.com:6443".  
[student@workstation ~]$ oc create serviceaccount project1-sa  
serviceaccount/project1-sa created  
[student@workstation ~]$ oc adm policy add-scc-to-user anyuid -z project1-sa  
clusterrole.rbac.authorization.k8s.io/system:scc:anyuid added: "project1-sa"  
[student@workstation ~]$ █
```

Q12) Deploy application in the project qed:

- There is one pod already running
- Modify the application as is should run with any user as provided by application
- Application should produce output

Dont run this command in exam it is needed to create a exam env on your labs

(oc new-app --name gitlab --docker-image gitlab/gitlab-ce:8.4.3-ce.0)

(oc expose service gitlab --port 80)

(oc get pods) - it will give you error now can try to resolve it.

Dont run this command in exam it is needed to create a exam env on your labs

Solution:-

oc project qed

oc get pods

oc get deployment (there can be deployment also not dc)

oc get dc (there can be dc also not deployment)

oc logs <podname>

oc get events

due to no permission the application will not be running it means you need to set the service account so that container can run with any user.

oc set serviceaccount –help|less

The screenshot shows a terminal window titled "student@workstation:~". The terminal history is as follows:

```
[student@workstation ~]$ oc get pod
NAME           READY   STATUS      RESTARTS   AGE
gitlab-c5fdbcc9b-9v79q   0/1     CrashLoopBackOff   6 (27s ago)   7m25s
[student@workstation ~]$ oc set serviceaccount deployment/gitlab-c5fdbcc9b-9v79q project1-sa
Error from server (NotFound): deployments.apps "gitlab-c5fdbcc9b-9v79q" not found
[student@workstation ~]$ oc get pod
NAME           READY   STATUS      RESTARTS   AGE
gitlab-c5fdbcc9b-9v79q   0/1     CrashLoopBackOff   7 (52s ago)   12m
[student@workstation ~]$ oc set serviceaccount deployment/gitlab project1-sa
deployment.apps/gitlab serviceaccount updated
[student@workstation ~]$ oc get pods
NAME           READY   STATUS    RESTARTS   AGE
gitlab-b85664fbe-qpqjh   1/1     Running   0          8s
[student@workstation ~]$
```

13) Install an helm chart etherpad from repository <http://helm.ocp4.example.com/charts>

```
student@workstation ~]$ helm repo add do280-repo http://helm.ocp4.example.com/charts
"do280-repo" has been added to your repositories
[student@workstation ~]$ helm search repo --versions
NAME          CHART VERSION   APP VERSION   DESCRIPTION
do280-repo/etherpad    0.0.7        latest       A Helm chart for etherpad lite
do280-repo/etherpad    0.0.6        latest       A Helm chart for etherpad lite
do280-repo/examplechart 0.1.0        latest       A Helm chart for Kubernetes
do280-repo/influxdb      4.12.1      1.8.10      Scalable datastore for metrics, events, a
nd rea...
do280-repo/mysql-persistent 0.0.2      0.0.2       This content is experimental, do not use i
t in p...
[student@workstation ~]$ helm install example-app do280-repo/etherpad
W0928 09:38:04.093363    6211 warnings.go:70] would violate PodSecurity "restricted:v1.24": allowPrivilegedEscalation != false (container "etherpad" must set securityContext.allowPrivilegeEscalation=false), unrestricted capabilities (container "etherpad" must set securityContext.capabilities.drop=["ALL"]), runAsNonRoot != true (pod or container "etherpad" must set securityContext.runAsNonRoot=true), seccompProfile (pod or container "etherpad" must set securityContext.seccompProfile.type to "RuntimeDefault" or "Localhost")
)
NAME: example-app
LAST DEPLOYED: Thu Sep 28 09:38:03 2023
NAMESPACE: helm-pro
STATUS: deployed
REVISION: 1
TEST SUITE: None

[student@workstation ~]$ oc get all
NAME                                         READY   STATUS    RESTARTS   AGE
pod/example-app-etherpad-65d78f5dd8-x66kn   0/1     Running   0          5s

NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/example-app-etherpad   ClusterIP  172.30.85.92 <none>        9001/TCP   5s

NAME                           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/example-app-etherpad   0/1     1           0          5s

NAME                           DESIRED   CURRENT   READY   AGE
replicaset.apps/example-app-etherpad-65d78f5dd8   1         1         0         5s

NAME          SERVICES   PORT   TERMINATION   HOST/PORT   WILDCARD   PAT
route.route.openshift.io/example-app-etherpad   example-app-etherpad-helm-pro.apps.ocp4.example.com
example-app-etherpad   http   edge/Redirect   None
```

16) Install an operator called file-integrity-manager

- Plan Automatic
- Using project openshift-file-integrity

Red Hat OpenShift

Administrator

Home

Operators

- OperatorHub
- Installed Operators

Workloads

Networking

Storage

Builds

Observe

Compute

Project: All Projects

All Items

Monitoring

Networking

OpenShift Optional

Security

Storage

Source

- do280 Operator Catalog Red Hat (5)
- Provider
- Red Hat (5)

Install state

- Installed (1)
- Not Installed (4)

Capability level

- Basic Install (2)
- Seamless Upgrades (2)

All Items

Filter by keyword...

5 items

do280 Operator Catalog Red Hat

Compliance Operator

An operator which runs OpenSCAP and allows you to keep your cluster compliant with...

do280 Operator Catalog Red Hat

File Integrity Operator

An operator that manages file integrity checks on nodes.

do280 Operator Catalog Red Hat

LVM Storage

An operator for deploying...

do280 Operator Catalog Red Hat

OpenShift Virtualization

Creates and maintains an OpenShift Virtualization

Red Hat OpenShift

Administrator

Home

Operators

- OperatorHub
- Installed Operators

Workloads

Networking

Storage

Builds

Observe

Compute

Project: All Projects

All Items

Monitoring

Networking

OpenShift Optional

Security

Storage

Source

- do280 Operator Catalog Red Hat (5)
- Provider
- Red Hat (5)

Install state

- Installed (1)
- Not Installed (4)

Capability level

- Basic Install (2)
- Seamless Upgrades (2)

All Items

Filter by keyword...

File Integrity Operator

1.2.1 provided by Red Hat

Install

Latest version

An operator that manages file integrity checks on nodes.

1.2.1

Capability level

- Basic Install
- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

Source

do280 Operator Catalog Red Hat

Provider

Red Hat

Red Hat OpenShift

Administrator

Home

Operators

- OperatorHub
- Installed Operators

Workloads

Networking

Storage

Builds

Observe

Compute

User Management

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

- stable

Installation mode *

- All namespaces on the cluster (default)
- A specific namespace on the cluster

Installed Namespace *

- openshift-file-integrity (Operator recommended)

Namespace creation

Namespace `openshift-file-integrity` does not exist and will be created.

Enable Operator recommended cluster monitoring on this Namespace

Namespace monitoring

File Integrity Operator

provided by Red Hat

Provided APIs

File Integrity

FileIntegrity is the Schema for the fileIntegrity API

FINS FileIntegrityNodeStatus

FileIntegrityNodeStatus defines the status of a specific node

Red Hat OpenShift

Administrator

Home

Operators

OperatorHub

Installed Operators

Workloads

Networking

Storage

Builds

Observe

Compute

User Management

Operator will be available in a single Namespace only.

Installed Namespace *

openshift-file-integrity (Operator recommended)

Namespace creation
Namespace `openshift-file-integrity` does not exist and will be created.

Enable Operator recommended cluster monitoring on this Namespace

Namespace monitoring
Please note that installing non-Red Hat operators into OpenShift namespaces and enabling monitoring voids user support. Enabling cluster monitoring for non-Red Hat operators can lead to malicious metrics data overriding existing cluster metrics. For more information, see the cluster monitoring documentation [\[?\]](#)

Update approval * [\[?\]](#)

Automatic

Manual

Install **Cancel**

Red Hat OpenShift

Administrator

Home

Operators

OperatorHub

Installed Operators

Workloads

Networking

Storage

Builds

Observe

Compute

User Management

File Integrity Operator
1.2.1 provided by Red Hat

Installed operator - ready for use

View Operator **View installed Operators in Namespace openshift-file-integrity**

Red Hat OpenShift

Administrator

Home

Overview

Projects

Search

API Explorer

Events

Operators

OperatorHub

Installed Operators

Workloads

Networking

Storage

Project: All Projects

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name	Namespace	Managed Namespaces	Status	Last updated	Provided APIs
File Integrity Operator 1.2.1 provided by Red Hat	openshift-file-integrity	All Namespaces	Succeeded Up to date	Sep 28, 2023, 8:38 AM	File Integrity FileIntegrityNodeStatus
LVM Storage 4.12.0 provided by Red Hat	openshift-storage	openshift-storage	Succeeded Up to date	Sep 28, 2023, 8:24 AM	LVMCluster
MetallB Operator 4.12.0-202307051630 provided by Red Hat	metallb-system	All Namespaces	Succeeded Up to date	Sep 28, 2023, 8:24 AM	BGPPeer AddressPool BFDProfile BGPAdvertisement View 4 more...
OpenShift Metrics 4.12.0-202307051630 provided by Red Hat	openshift-metrics	All Namespaces	Succeeded	Sep 28, 2023, 8:26 AM	OpenShiftMetrics

14) Create a cronjob test-cron

- 04:05 time
- Every 2 day and every month
- Use image registry.io/imagename
- The service account and service account name is project1-sa
- Successful job history limit 14
- Project name should be cron-test

Create a cronjob using webconsole or take yaml from documentation

Both provided in instruction

```
[student@workstation ~]$ oc get all
NAME           SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
cronjob.batch/cron-test  05 04 2 * *  False       0          <none>    22s
[student@workstation ~]$
```

```
spec:
  containers:
    - args:
        - /bin/sh
        - -c
        - date; echo Hello from the Kubernetes cluster
      image: anyinge name given in exarm
      imagePullPolicy: IfNotPresent
      name: hello
      resources: {}
      terminationMessagePath: /dev/termination-log
      terminationMessagePolicy: File
    dnsPolicy: ClusterFirst
    restartPolicy: OnFailure
    schedulerName: default-scheduler
    securityContext: {}
    terminationGracePeriodSeconds: 30
  schedule: 05 04 2 * *
  successfulJobsHistoryLimit: 14
  suspend: false
status: {}
```

```
[student@workstation ~]$ # create service account
[student@workstation ~]$
[student@workstation ~]$ oc create sa jupiter
serviceaccount/jupiter created
[student@workstation ~]$ oc adm policy add-scc-to-user anyuid -z jupiter
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:anyuid added: "jupiter"
[student@workstation ~]$ oc adm policy add-cluster-role-to-user cluster-admin -z jupiter
clusterrole.rbac.authorization.k8s.io/cluster-admin added: "jupiter"
[student@workstation ~]$
[student@workstation ~]$ # set service account in cron job
[student@workstation ~]$
[student@workstation ~]$ oc set sa cronjob.batch/cron-test jupiter
cronjob.batch/cron-test serviceaccount updated
[student@workstation ~]$
```

20) create an network policy to connect two pod which are resides in different projects

- First pod is in database namespace and second is in checker project
- Pod in project checker should communicate with database pod
- Create an network policy with name mysql-db-conn
- Policy uses labels to specify namespace with team=devsecops and for pods selector
- Uses label deployment=web-mysq
- Use port 3306/tcp
- Check the logs of checker pod for verification

NETWORK POLICY -->

--> USE THE LABEL GIVEN IN THE QUESTION FOR DATABASE DEPLOYMENT FROM THE YAML.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: web-allow-prod
  namespace: default
spec:
  podSelector:
    matchLabels:
      netwroking.k8s.io/v1/network: database
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              team: devsecops
        podSelector:
          matchLabels:
            deployment: my-web-mysql
  ports:
    - protocol: TCP
      port: 3306
```

(CREATE IN DATABASE NAMESPACE)

18) Create a pv

- Name is landing-pv
- Size - 1Gi
- Policy - retain
- Mode - ReadOnlyMany

Create a pvc

- Name is landing-pvc
- Size same as pv
- Mount pvc to /usr/share/nginx/html
- Project name is page
- Mode same as pv

Create an Deployment with

- Name landing
- Image is registry.ocp4.example.com:8443/redhattraining/hello-world-nginx:v1.0
- Application uses this link to show output <http://contet.example.ocp4.com>
- After attaching storage it shows desired output
- Name of deployment is landing

Solution

Check details of storage class

```
[student@workstation ~]$ oc describe storageclasses.storage.k8s.io nfs2
Name:          nfs2
IsDefaultClass: No
Annotations:   <none>
Provisioner:   example.com/external-nfs
Parameters:    path=/open001,readOnly=false,server=192.168.50.254
AllowVolumeExpansion: <unset>
MountOptions:  <none>
ReclaimPolicy: Delete
VolumeBindingMode: Immediate
Events:        <none>
[student@workstation ~]$
```

Create pv

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: landing-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
  - ReadOnlyMany
  nfs:
    path: /open001
    server: 192.168.50.254
  persistentVolumeReclaimPolicy: Retain
~
~
```

Create pvc in project page

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: landing-pvc
spec:
  accessModes:
    - ReadOnlyMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: nfs2
```

Project: default ▾ Application: All applications ▾

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *
landing

Deployment strategy

Strategy type
Rolling Update

The rolling strategy will wait for pods to pass their readiness check, scale down old components and then scale up.

Maximum number of unavailable Pods
25%

The maximum number of pods that can be unavailable during the rolling deployment. This can be either a percentage (10%) or a whole number (1).

Maximum number of surge Pods

Images

Container:  container

Deploy image from an image stream tag

Image Name *

`registry.ocp4.example.com:8443/redhattraining/hello-world-nginx:v1.0`

Container image name

► Show advanced image options

Environment Variables

Container:  container

Name	Value
------	-------

Deployments > Deployment details

D landing

Actions ▾

Details Metrics YAML ReplicaSets Pods Environment Events

Deployment details

 3 Pods

Name landing	Update strategy RollingUpdate	Add storage
Namespace NS default	Max unavailable 25% of 3 pods	Edit update strategy
Labels No labels	Max surge 25% greater than 3 pods	Edit resource limits
		Edit labels
		Edit annotations
		Edit Deployment
		Delete Deployment

Add Storage to **D** landing

PersistentVolumeClaim *

Use existing claim
PVC landing-pvc

Create new claim

Mount path *

/usr/share/nginx/html

Mount path for the volume inside the container.

Mount as read-only

Subpath

Optional path within the volume from which it will be mounted into the container. Defaults to the root of the volume.

The volume will be mounted into all containers. You can [select specific containers](#) instead.

Save **Cancel**

Containers					
Name	Image	Resource limits	Ports		
container	registry.ocp4.example.com:8443/redhattraining/hello-world-nginxv1.0	-	8080/TCP		
Volumes					
Name	Mount path	SubPath	Type	Permiss...	Utilized by
landing-pvc	/usr/share/nginx/html	No subpath	PVC landing-pvc	Read/Write	container

Conditions

Expose deployment

Expose svc nginx –hostname contet.example.ocp4.com

Curl on route to check

15) Create an project template with limitrange with container

- minimum memory is 5Mi max is 1Gi defaultrequest 254 Mi defaultlimit is 512 Mi .
- make sure this template available as default request new-project template for users.

Create limit range using web console and move ..

```
apiVersion: v1
kind: LimitRange
metadata:
  creationTimestamp: "2023-09-29T09:19:32Z"
  name: mem-limit-range
  namespace: "6654"
  resourceVersion: "476088"
  uid: f6f5d12b-1209-47b5-bd8b-07eab7de4080
spec:
  limits:
    - default:
        memory: 512Mi
      defaultRequest:
        memory: 256Mi
    max:
      memory: 1Gi
    min:
      memory: 128Mi
    type: Container
"/tmp/oc-edit-970466099.yaml" 23L, 575C
```

```
[student@workstation ~]$ oc adm create-bootstrap-project-template -o yaml > template.yaml
[student@workstation ~]$
[student@workstation ~]$
[student@workstation ~]$ oc get limitranges mem-limit-range -o yaml > file
```

```
[student@workstation ~]$
[student@workstation ~]$ vim template.yaml
[student@workstation ~]$ # combine both files template+limitrange
[student@workstation ~]$
[student@workstation ~]$ █
```

```
  kind: ClusterRole
  name: admin
  subjects:
    - apiGroup: rbac.authorization.k8s.io
      kind: User
      name: ${PROJECT_ADMIN_USER}
  apiVersion: v1
  kind: LimitRange
  metadata:
    name: ${PROJECT_NAME}-limits
  spec:
    limits:
      - default:
          memory: 512Mi
        defaultRequest:
          memory: 256Mi
      max:
        memory: 1Gi
      min:
        memory: 128Mi
      type: Container
  parameters:
    - name: PROJECT_NAME
    - name: PROJECT_DISPLAYNAME
    - name: PROJECT_DESCRIPTION
    - name: PROJECT_ADMIN_USER
    - name: PROJECT_REQUESTING_USER
```

```
[student@workstation ~]$ oc create -f template.yaml -n openshift-config
template.template.openshift.io/project-request created
[student@workstation ~]$ █
```

```
[student@workstation ~]$
[student@workstation ~]$ oc edit projects.config.openshift.io cluster
```

```
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
  name: version
  uid: db3a68ee-9e5c-47cc-9301-b1ce0d707a0a
  resourceVersion: "1781"
  uid: 82190a26-2ff0-437c-8535-530d499f61f0
spec:
  projectRequestTemplate:
    name: project-request
```

```
[student@workstation ~]$ oc edit projects.config.openshift.io cluster
project.config.openshift.io/cluster edited
[student@workstation ~]$ oc get pods -n openshift-apiserver
NAME           READY   STATUS    RESTARTS   AGE
apiserver-7c949fbc9d-xbnlc  2/2     Running   12          148d
[student@workstation ~]$ oc get pods -n openshift-apiserver -w
NAME           READY   STATUS    RESTARTS   AGE
apiserver-7c949fbc9d-xbnlc  2/2     Terminating   12          148d
apiserver-f9674c5b4-g57mp   0/2     Pending    0            3s
apiserver-7c949fbc9d-xbnlc  1/2     Terminating   12          148d
```

```
[student@workstation ~]$ oc get pods -n openshift-apiserver
NAME           READY   STATUS    RESTARTS   AGE
apiserver-f9674c5b4-g57mp  2/2     Running   0            112s
[student@workstation ~]$
```

```
[student@workstation ~]$ oc new-project 9988
Now using project "9988" on server "https://api.ocp4.example.com:6443".
```

You can add applications to this project with the 'new-app' command. For example, try:

```
  oc new-app rails-postgresql-example
```

to build a new example application in Ruby. Or use kubectl to deploy a simple Kubernetes application:

```
  kubectl create deployment hello-node --image=k8s.gcr.io/e2e-test-images/agnhost:2.33 -- /agnhost serve-hostname
```

```
[student@workstation ~]$ oc get limitranges
NAME      CREATED AT
memtotal-mem  2023-09-29T10:07:34Z
[student@workstation ~]$ oc get limitranges -n 9988
NAME      CREATED AT
memtotal-mem  2023-09-29T10:07:34Z
[student@workstation ~]$ █
```

21) Start a Probe in project start

- Create a Liveliness Health Probe in project tuesday which has 1 pod running
- With port 8443
- Initial delay of 3 sec
- Time out for the probe is 10 sec
- Probe must survive atleast 3 crash

The screenshot shows the Red Hat OpenShift web interface. The left sidebar is titled 'Workloads' and includes options like OperatorHub, Installed Operators, Workloads (Pods, Deployments, DeploymentConfigs, StatefulSets, Secrets, ConfigMaps, CronJobs, Jobs, DaemonSets, ReplicaSets, ReplicationControllers, HorizontalPodAutoscalers, PodDisruptionBudgets), and a 'Create Deployment' button. The main content area is titled 'Deployments' and shows a table with one row for 'dep-new01'. The table columns are Name, Status, Labels, and Pod selector. The status is '3 of 3 pods' and labels are 'No labels'. The pod selector is 'app=dep-new01'. To the right of the table is a context menu with options: Edit Pod count, Add HorizontalPodAutoscaler, Add PodDisruptionBudget, Pause rollouts, Restart rollout, Add Health Checks (which is highlighted in yellow), Add storage, Edit update strategy, Edit resource limits, and Edit labels.

The screenshot shows the Red Hat OpenShift web interface. The left sidebar is identical to the previous screenshot. The main content area is titled 'Readiness probe' and contains the text: 'A readiness probe checks if the Container is ready to handle requests. A failed readiness probe means that a Container should not receive any traffic from a proxy, even if it's running.' Below this is a 'Add Readiness probe' button. The next section is 'Liveness probe' with the text: 'A liveness probe checks if the Container is still running. If the liveness probe fails the Container is killed.' Below this is a 'Add Liveness probe' button. The final section is 'Startup probe' with the text: 'A startup probe checks if the application within the Container is started. If the startup probe fails the Container is killed.' Below this is a 'Add Startup probe' button. At the bottom of the page are 'Add' and 'Cancel' buttons.

18) Collect Cluster information and create a tar file with name system10<cluster.id>.tar.gz and send it to redhat support.

- Use command tar cvaf
- One script has been provided to upload tar in redhat support
- /usr/bin/script system10<cluster.id>.tar.gz
- This script can be performed multiple times and it will overwrite the tar file every time

Solution:

```
[student@workstation ~]$ # collect all cluster information  
[student@workstation ~]$  
[student@workstation ~]$ oc adm must-gather
```

```
[student@workstation ~]$ tar cvaf system10-0ac9f411-9bc0-4cd4-9c63-1146da12ac83.tar.gz must-gather.local.3396062060357082571/
```

```
pvc.yml  
pv.yaml  
sc.yml  
system10-0ac9f411-9bc0-4cd4-9c63-1146da12ac83.tar.gz  
Templates  
template.yaml  
Videos  
[student@workstation ~]$ /usr/bin/script system10-0ac9f411-9bc0-4cd4-9c63-1146da12ac83.tar.gz
```