

# Practice Set for Ex280

## REMEMBER

1. Believe in yourself, you can easily score 100%
2. Before solving any question, first clear your concepts related to that question.
3. Always try to solve yourself first then check doc solution (while attempting questions 1<sup>st</sup> time).
4. Practice a lot and try to solve all questions

Instructions:

1. Password for all the users in workbench is <given password>
2. You can access the openshift using the console <web UI console link>
3. <I.P. of workbench> workbench.example.com workbench
4. Official online documents < document links > #Suggested to open pdf.

## | Q1 Create users: [ Hint: do on cli and web console both ]

- A. Create users such that Identity name should be ex280-htpasswd
- B. Secret used to create the identity should be ex280-idp-secret
- C. Create user armstrong with the password setiyet
- D. Create user collins with the password rdxterr
- E. Create user aldrin with the password kusterus
- F. Create user jobs with the password demost
- G. Create user wozniak with the password nitosh

Additional scenario (means files or pre-given environment):

1. kubeadmin password will be given at workbench at location= /root/kube-passwd
2. After doing “ssh root@workbench” you will be logged in as the system:admin with cluster-admin rights.
3. httpd-tools will be pre-installed [ better run always “yum install httpd-tools” ]

# Ans 1:

**Step 1: open terminal of the Host OS (The landing OS after logging in the environment).**

**Step 2: open the view Exam app from the activities section in the Host OS (RHEL 8).**

**Step 3: ----- Solution starts from now-----**

**Note: “root@workbench]#” means terminal identifier**

```
root@host_os]# ssh root@workbench
```

```
root@workbench]# oc whoami
```

**output > system:admin**

```
root@workbench]# yum install httpd-tools -y
```

**output > [ installing... httpd-tools.....]**

```
$ yum install httpd-tools -y
Centos-8 - AppStream                               16 kB/s | 4.3 kB   00:00
Centos-8 - AppStream                               15 MB/s | 6.3 MB   00:00
Centos-8 - Base                                    12 kB/s | 3.9 kB   00:00
Centos-8 - Base                                    7.1 MB/s | 2.3 MB   00:00
Centos-8 - Extras                                 4.3 kB/s | 1.5 kB   00:00
Centos-8 - Extras                                 32 kB/s | 9.2 kB   00:00
Extra Packages for Enterprise Linux Modular 8 - x86_64    33 kB/s | 17 kB   00:00
Extra Packages for Enterprise Linux Modular 8 - x86_64    363 kB/s | 557 kB   00:01
Extra Packages for Enterprise Linux 8 - x86_64       35 kB/s | 18 kB   00:00
Extra Packages for Enterprise Linux 8 - x86_64       7.7 MB/s | 9.1 MB   00:01
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
<hr/>				
Installing:				
httpd-tools	x86_64	2.4.37-30.module_el8.3.0+561+97fdbbcc	AppStream	104 k
Installing dependencies:				
apr	x86_64	1.6.3-11.el8	AppStream	125 k
apr-util	x86_64	1.6.1-6.el8	AppStream	105 k
Installing weak dependencies:				
apr-util-bdb	x86_64	1.6.1-6.el8	AppStream	25 k
apr-util-openssl	x86_64	1.6.1-6.el8	AppStream	27 k

```
root@workbench]# htpasswd -c -b -B ex280-htpasswd armstrong setiyet
```

**output > Adding password for user armstrong**

```
root@workbench]# htpasswd -b -B ex280-htpasswd collins rdxterr
```

**output > Adding password for user collins**

```
root@workbench]# htpasswd -b -B ex280-htpasswd aldrin kusterus
```

**output > Adding password for user aldrin**

```
root@workbench]# htpasswd -b -B ex280-htpasswd jobs demost
```

**output > Adding password for user jobs**

```
root@workbench]# htpasswd -b -B ex280-htpasswd wozniak nitosh
```

**output > Adding password for user wozniak**

```
root@workbench]# ls
```

**output > kube-passwd ex280-htpasswd**

```
root@workbench]# cat ex280-htpasswd
```

**output >**

```
$ htpasswd -c -b -B ex280-htpasswd armstrong setiyet
Adding password for user armstrong
$ htpasswd -b -B ex280-htpasswd collins rdxterr
Adding password for user collins
$ htpasswd -b -B ex280-htpasswd aldrin kusterus
Adding password for user aldrin
$ htpasswd -b -B ex280-htpasswd jobs demost
Adding password for user jobs
$ htpasswd -b -B ex280-htpasswd wozniak nitosh
Adding password for user wozniak
$ ls
ex280-htpasswd  kube-passwd
$ cat ex280-htpasswd
armstrong:$2y$05$Si3G56tw14WNQm0xe80sETOqGmBop.hSpmSmqha7qjrIM4qZQ.wAPy
collins:$2y$05$1DMqIEfNyS2FKnrUQodEE.KYuTVC0LVN1bzzdUk21PW2QKfwa.p4S
aldrin:$2y$05$O1RU2ryG7rD6/H44H5g9me3k86/OU5.KIRc5GaU0aobUAq7StkMAM
jobs:$2y$05$XQoJvkC8Zv7vnLuU6DWq0.mqlir3BK29v6MiP4.eRjCtUwkj3td8y
wozniak:$2y$05$3rd8KdNoy8b0vDXhWIUoGeQlAT.o1Z4VAwiMkNkAbr2CaMi4yOMhG
```

```
root@workbench]# cat ex280-htpasswd
```

**output >**

```
armstrong:$2y$05$TMdbbnuAp2u/xHbiVOm5oOcrRMiKGQ7JuUyB/Yu9mUywYiwzCRNP.
```

```
collins:$2y$05$jTP5wnM0Q2vEu1bYA1FBG.X1hUMoksl0iLnUqJ0LLuYGZ7JUSdYy
```

```
aldrin:$2y$05$tQ3qzKDGSITL0zou9APa/ODkOwxQzUJm5pLcBkSzbQJ/qR3YYuSvO
```

```
jobs:$2y$05$ukNRyBS/3LqUQqSOxngYVufU8dwTA/84YKQk7yl/Q8/kt3E5m6VZO
```

```
wozniak:$2y$05$UMHrPv9KeGAN1yg33/xos.CcaeJKjycgT7zfj/z228bObeuTjwDV.
```

```
root@workbench]# cat kube-passwd
```

```
output > $2y$05$AhVQ3w5mKwjXKXGBQqU2=
```

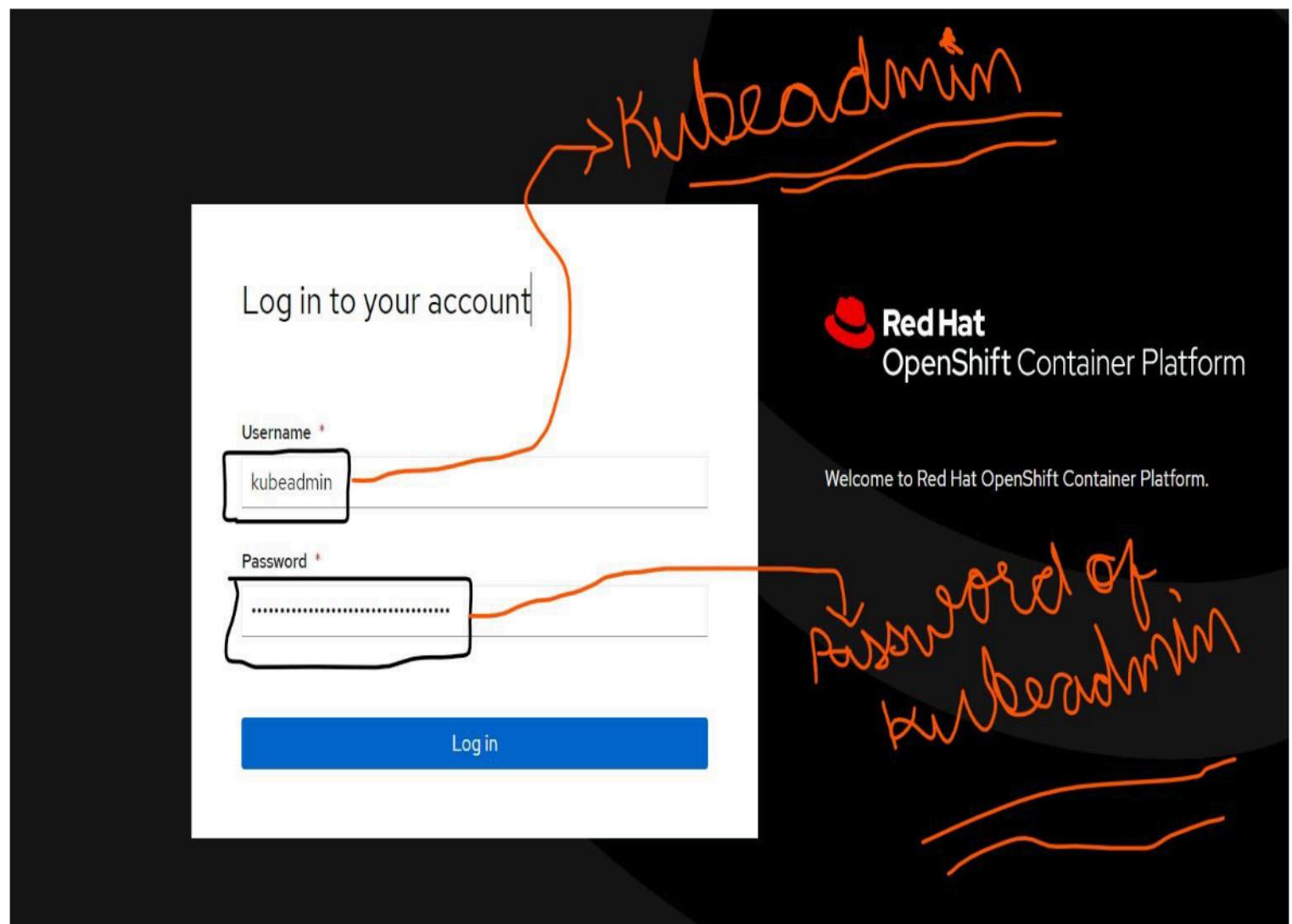
```
$ cat kube-passwd  
$2y$05$AhVQ3w5mKwjXKXGBQqU2=
```

```
root@workbench]# oc create secret generic ex280-idp-secret --from-file=htpasswd=ex280-htpasswd -n openshift-config
```

```
output > secret/ex280-idp-secret created
```

```
$ oc create secret generic ex280-idp-secret --from-file=htpasswd=ex280-htpasswd -n openshift-config  
secret/ex280-idp-secret created
```

#### Step 4: open web-console of openshift in Firefox



Red Hat  
OpenShift  
Container Platform

- Home
- Overview
- Projects
- Search
- Explore
- Events
- Operators
- Workloads
- Networking
- Storage
- Builds
- Monitoring
- Compute
- User Management
- Administration**

Provider  
Libvirt  
OpenShift Version  
4.5.4  
Update Channel  
stable-4.5

Cluster Inventory

click 1

Compute >

User Management >

Administration ▾

- Cluster Settings** → 2
- Namespaces
- Resource Quotas
- Limit Ranges
- Custom Resource Definitions

The screenshot shows the left sidebar of the OpenShift web interface. The 'Administration' section is expanded, showing the 'Cluster Settings' option, which is highlighted with a blue underline. Other options like 'Events', 'Operators', 'Workloads', 'Networking', 'Storage', 'Builds', 'Monitoring', 'Compute', and 'User Management' are also listed.

The screenshot shows the 'Cluster Settings' page. At the top, there are three tabs: 'Details' (underlined), 'Cluster Operators', and 'Global Configuration'. A large red box labeled 'click' with a red arrow points to the 'Global Configuration' tab. Below the tabs, there's a section for 'Channel' set to 'stable-4.5' with a pencil icon, and 'Last Completed Ver: 4.5.4'. A red box labeled '3' with a red arrow points to the channel selection area. The main content area includes sections for 'Cluster ID', 'Desired Release Image' (set to 'quay.io/openshift-release-dev/ocp-release@sha256:02dfcae8f6a67e71538'), and 'Cluster Version Configuration' (with a 'CV version' button). A red box labeled '4' with a red arrow points to the 'CV version' button.

The screenshot shows the 'Global Configuration' page under the 'Cluster Settings' section. At the top, there are three tabs: 'Details', 'Cluster Operators', and 'Global Configuration' (highlighted with a blue underline). A large red box labeled 'click' with a red arrow points to the 'Global Configuration' tab. Below the tabs, there's a section for 'Edit the following resources to manage the configuration of your cluster.' It lists 'Authentication' and 'OAuth'. A red box labeled '5' with a red arrow points to the 'Authentication' link. A red box labeled '6' with a red arrow points to the 'OAuth' link. The main content area shows a table with columns 'Configuration Resource' and 'Description'. The first row is for 'Authentication', and the second row is for 'OAuth'. Each row has a 'More' button at the end.

**OA** cluster
Details

YAML

## OAuth Details

## Name

cluster

## Labels

No labels

**OA** cluster
DetailsYAML

```

35   selfLink: /apis/config.openshift.io/v1/oauths/cluster
36   uid: 6df6b890-5521-45a7-8844-5d9da61ba1e2
37 spec:
38   identityProviders:
39     - name: ex280-htpasswd
40       type: HTPasswd
41       mappingMethod: claim
42       htpasswd:
43         fileData:
44           name: ex280-idp-secret
45
46

```

**← identity**

**← secret**

Save

Reload

Cancel

**click 6**

## OA cluster

[Details](#) [YAML](#)

### OAuth Details

Name  
cluster

Labels  
No labels

Annotations  
2 Annotations 

Created At  
 Jul 29, 2020 10:16 am

Owner  
No owner

### Identity Providers

Identity providers determine how users log into the cluster.

[Add](#) ▾

Name	Type	Mapping Method
ex280-htpasswd	HTPasswd	claim

8  
Identity

**Important Note: if like above image it doesn't show don't worry move to next step that will make it happen.**

### Step 5: Activating the identity

**Note: switch to terminal from here**

```
$ oc login -u user
Authentication required for https://openshift:6443 (openshift)
Username: user
Password:
```

**Here replace user with the below list and also give respective password:**

Username	Password
armstrong	setiyet
collins	rdxterr
aldrin	kusterus
jobs	demost
wozniak	nitosh

----- You should be able to login and your identity would be activated-----

----- Question one ends here-----

## Q2 Create Project and configure permissions

[Hint: do on cli]

**a. Make sure the projects should be available:**

[ Note: project names would be different from here ]

- manhattan
- havannah
- drac
- apollo
- pending

**b. User armstrong should be able to administrate project manhattan and havannah.**

**C. user wozniak should be able to edit the project apollo.**

|-----|

## Ans 2:

### Step 1: Create projects

```
root@workbench]# oc new-project manhattan
```

output > now using and switched to project manhattan

```
root@workbench]# oc new-project havannah
```

output > now using and switched to project havannah

```
root@workbench]# oc new-project drac
```

output > now using and switched to project drac

```
root@workbench]# oc new-project apollo
```

output > now using and switched to project apollo

```
root@workbench]# oc new-project pending
```

output > now using and switched to project pending

### Step 2: assign rolebindings to the users

- a. root@workbench]# oc project < the project where you want user to have access or bind to user>
- b. root@workbench]# oc policy add-role-to-user < role > < user > -n < respective project >

**Note: do follow the approach a->b->a->b and so on... as to reduce errors first a to switch to project you want to bind user with then apply b to apply rolebindings to user on project**

< role > -> edit , admin , view ( edit for edit permission, admin for admin access and view for view access ).

< user > -> put the user depends on the project and permission.

< respective project > -> put the project on which you want user to have role on

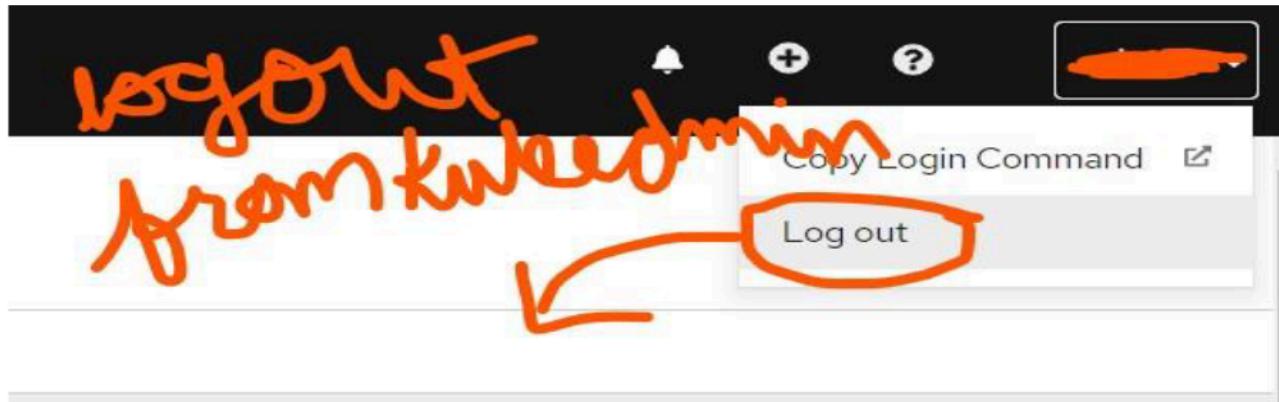
**Q3. Configure Cluster Permission:** [ hint: do on cli ]

- a. User jobs should be able to do administrate the cluster.
  - b. User woznaik should not have any administration power
  - c. No user should be able to provision project
  - d. collins user would be able to provision project
  - e. kubeadmin user should be deleted

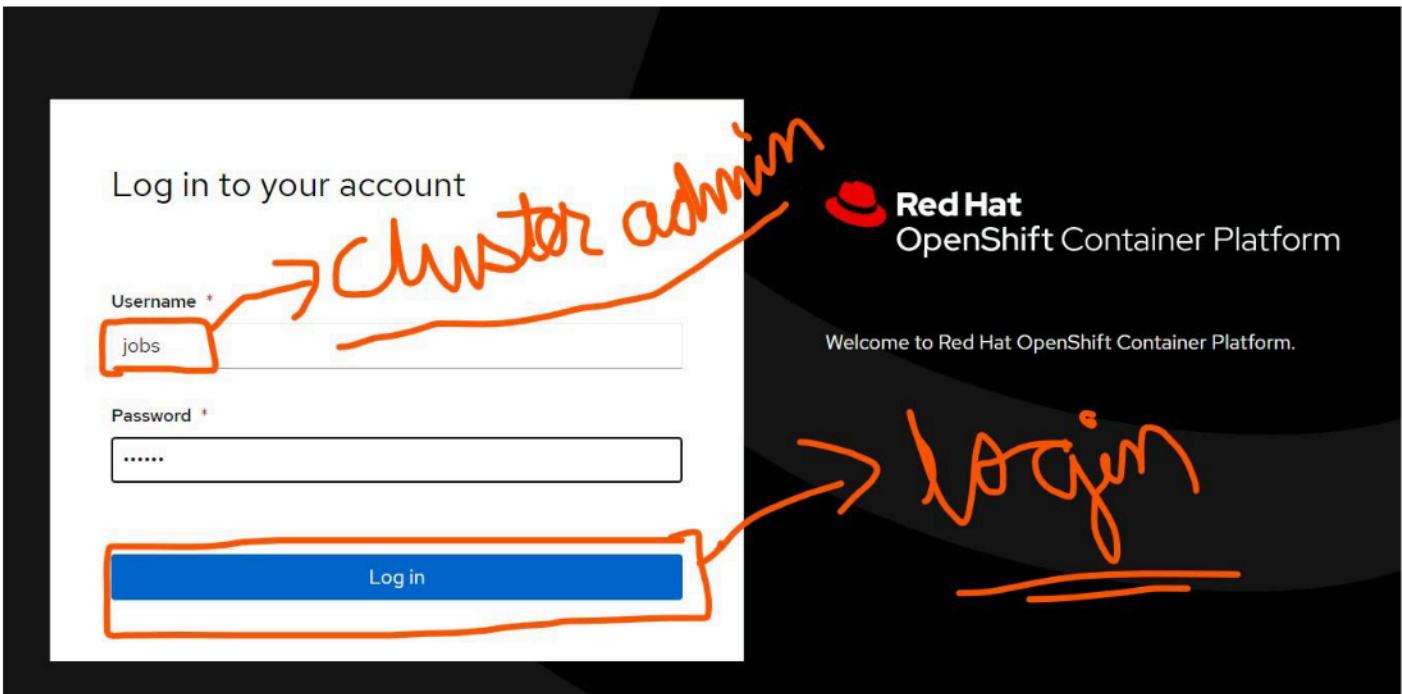
**Ans 3:**

```
# oc adm policy add-cluster-role-to-user cluster-admin jobs  
  
# oc adm policy remove-cluster-role-from-user cluster-admin woznaik  
  
# oc adm policy remove-cluster-role-from-group self-provisioner system:authenticated:oauth  
  
# oc adm policy add-cluster-role-to-user self-provisioner collins
```

Important point



**--- Imp note: log out from kubeadmin in the web console  
and log in as jobs user there as jobs user is cluster admin---**



[Important point]

```
# oc delete secrets kubeadmin -n kube-system
```

```
|-----|
```

```
|-----|
```

## Q4. Create and configure Groups [ hint: do on cli ]

```
|-----|
```

- a. Create a group named commander in the cluster.
- b. The user armstrong should be part of commander group.
- c. Create a group named pilot in the cluster.
- d. The users collins and aldrin should be part of pilot group.
- e. commander group should be able to edit the project drac
- f. pilot group should be able to view the project manhattan

## Ans 4:

```
# oc adm groups new commander armstrong  
# oc adm groups new pilot collins aldrin  
# oc project drac  
# oc policy add-role-to-group edit commander -n drac  
# oc project manhattan  
# oc policy add-role-to-group view pilot -n manhattan
```

## Q5. Create quota [ hint: do on terminal cli ]

- a. Create a resource quota ex280-quota on the project natasha
- b. The pods quota should be 3
- c. The service quota should be 6
- d. The memory quota should be 1G
- e. The cpu quota should be 2
- f. The replication controller quota should be 3

## Ans 5:

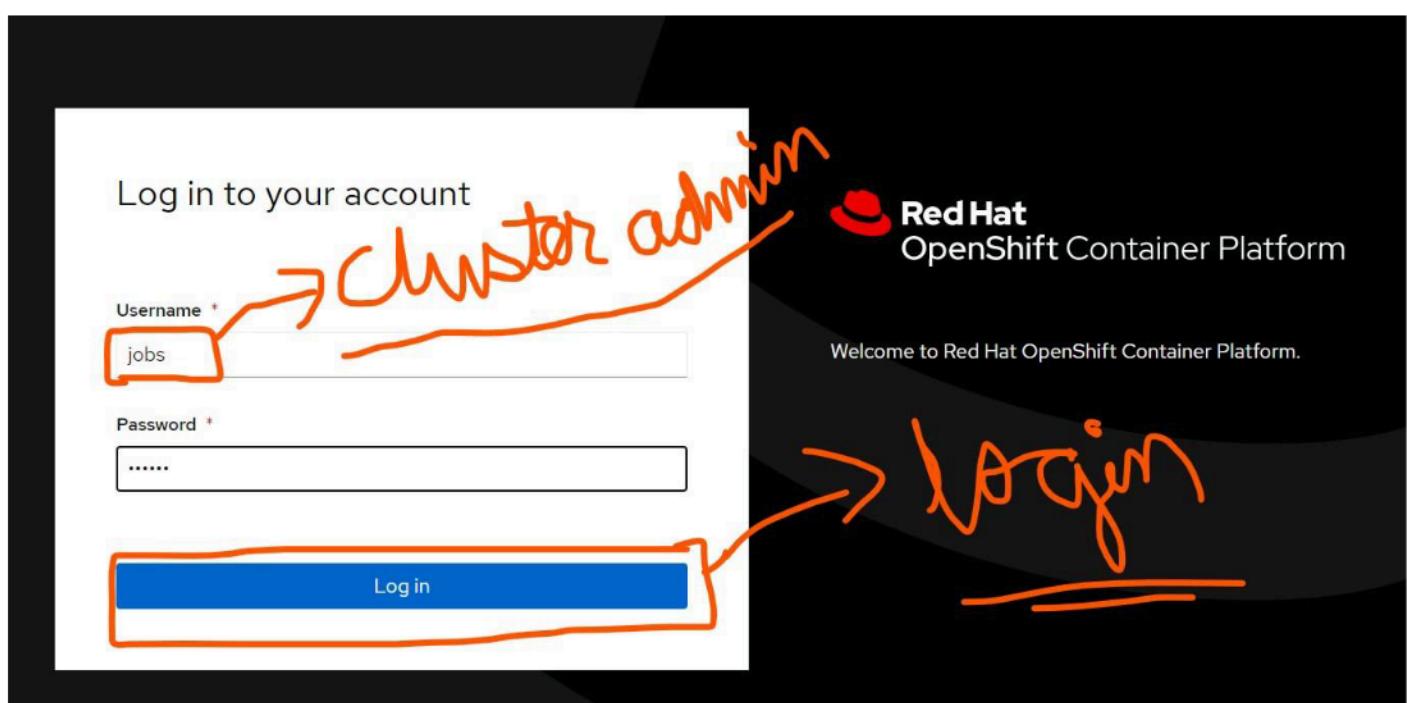
```
# oc project natasha  
# oc create quota ex280-quota --hard(cpu=2,memory=1G,pods=3,services=6,replicationcontrollers=3)  
# oc describe resourcequota ex280-quota -n natasha
```

## Q6. Create Limit [hint: should do on web console]

- a. Create a limit ex280-limits on the project salty
- b. The container memory should be 5Mi to 300Mi
- c. The pod memory should be 5Mi to 300Mi
- d. The container cpu should be 10m to 500m
- e. The pod cpu should be 10m to 500m
- f. The default request for container on memory and cpu should be 100Mi and 100m respectively

Ans 6:

Step 1: Go to web console and login as jobs user



Step 2: Go to limit range and select project where you want to create limit range.



Events

Operators

Workloads

Networking

Storage

Builds

Monitoring

Compute

User Management

Administration

Cluster Settings

Namespaces

Resource Quotas

Limit Ranges

Custom Resource Definitions

Project: all projects

## Limit Ranges

select  
Project  
where  
we  
want  
to  
create  
limit

Project: all projects

salty

Create Project

+ salty

1

project  
select

## Step 3: create the limitrange:



## Step 4: create limit:

Project: salty

### Create Limit Range

Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

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

View shortcuts

limit name  
project  
for container  
for pod

Create Cancel Download

## Step 5: Check Limits:

The screenshot shows a "Limit Range Details" page for a project named "salty". The page includes sections for "Details" and "YAML". Handwritten annotations include: "limit name" pointing to the "Name" field; "project name" pointing to the "Namespace" field; and "details" pointing to the "Limits" table.

Project: salty

Limit Ranges > Limit Range Details

LR ex280-limits

Details YAML

Limit Range Details

Name ex280-limits

Namespace salty

Labels No labels

Annotations 0 Annotations

Created At a few seconds ago

Owner No owner

Limits

Type	Resource	Min	Max	Default Request	Default Limit	Max Limit/Request Ratio
Container	cpu	10m	500m	100m	500m	-
Container	memory	5Mi	300Mi	100Mi	300Mi	-
Pod	cpu	10m	500m	-	-	-
Pod	memory	5Mi	300Mi	-	-	-

- If all details right then good to go.

## Q7. Deploy The application

[hint: should do on web console and cli]

|-----|

- a. An application rocky is running on project broker make sure the application is running
- b. The application should produce output on route rocky and should be accessible with the url <https://rocky-broker.ocp4.apps.example.com>

|-----|

## Ans 7:

**Step 1: go to web-console and go to Compute to find the kind of taint to gather data to apply toleration on the dc:**

[ Hint { very Imp } : Don't remove taint, apply tolerations ]

Red Hat  
OpenShift  
Container Platform

Horizontal Pod Autoscalers

Networking

Storage

Builds

Build Configs

Builds

Image Streams

Monitoring

**Compute**

1

2

Nodes

3

click

Name ↑ Status ↑ Role ↑ Pods ↑

Name	Status	Role	Pods
crc-fd5nx-master-0	Ready	worker	-

must be  
a worker

## **Step 2:**

**Go to worker node and then go to details of it then to taints as shown in image**

The screenshot shows the OpenShift web interface. On the left, there is a sidebar with the following menu items:

- Networking
- Storage
- Builds
  - Build Configs
  - Builds
- Image Streams
- Monitoring
  - Compute
  - Nodes
- Machines
- Machine Sets
- Machine Autoscalers
- Machine Health Checks
- Machine Configs
- Machine Config Pools
- User Management

Red annotations with numbers 1, 2, 3, and 4 are overlaid on the interface:

- An arrow points from the 'Compute' item to the 'Nodes' item.
- An arrow points from the 'Nodes' item to the 'crc-fd5nx-master-0' node in the main content area.
- A red box highlights the 'Details' tab in the top navigation bar of the node details page.
- A red box highlights the 'Taints' section on the right side of the node details page, with the number 4 and the word 'click' written next to it.

The main content area shows the 'Node Details' for the node 'crc-fd5nx-master-0'. It includes sections for Node Name, Status, External ID, Node Addresses, and Node Labels. The 'Taints' section is highlighted with a red box and the number 4, with the word 'click' written next to it.

**Step 3: check what kind of taint is there**

The screenshot shows the OpenShift web interface with the 'Compute' menu selected in the sidebar. The main content area displays the 'Edit Taints' dialog and the node's labels and taints.

The 'Edit Taints' dialog has the following fields:

KEY	VALUE	EFFECT
node	worker	NoSchedule

Below the dialog, the node's labels and taints are listed:

- beta.kubernetes.io/arch=amd64
- beta.kubernetes.io/os=linux
- kubernetes.io/arch=amd64
- kubernetes.io/hostname=crc-fd5nx-master-0
- kubernetes.io/os=linux
- node-role.kubernetes.io/master
- node-role.kubernetes.io/worker
- node.openshift.io/os\_id=rhcos

Yellow annotations with numbers 1, 2, 3, and 4 are overlaid on the interface:

- An arrow points from the 'Compute' item in the sidebar to the 'Nodes' item.
- An arrow points from the 'Nodes' item in the sidebar to the 'Taints' section in the main content area.
- A yellow box highlights the 'Taints' section in the main content area, with the number 3 written next to it.
- A yellow box highlights the 'Save' button in the 'Edit Taints' dialog, with the number 4 written next to it.

# Click cancel then go to deploymentconfigs of the project

The screenshot shows the Red Hat OpenShift Container Platform interface. On the left, the navigation sidebar is visible with the following structure:

- Home
- Operators
- Workloads** (highlighted with a red box)
  - Pods
  - Deployments
  - Deployment Configs** (highlighted with a red box)
  - Stateful Sets
  - Secrets
  - Config Maps

A red arrow labeled '1' points from the 'Workloads' section to the 'Deployment Configs' item. A red arrow labeled '2' points from the 'Deployment Configs' item to the 'Deployment Configs' link in the main content area. A red box highlights the 'Project: broker' dropdown in the top right. A red arrow labeled '3' points from the 'Project: broker' dropdown to the 'Deployment Configs' heading. A red box highlights the 'DC rocky' entry in the list. A red arrow labeled '4' points from the 'DC rocky' entry to the bottom right corner of the page.

**Deployment Configs**

Name	Namespace	Status
DC rocky	NS broker	1 of 1 pods

The screenshot shows the Red Hat OpenShift Container Platform interface, specifically the 'Deployment Config Details' page for 'DC rocky'. The navigation sidebar on the left is identical to the one in the previous screenshot, with the 'Deployment Configs' item highlighted by a red box and labeled '1'. A red arrow labeled '2' points from the 'Deployment Configs' item to the 'Deployment Config Details' link in the breadcrumb trail. A red box highlights the 'Project: broker' dropdown in the top right. A red arrow labeled '3' points from the 'Project: broker' dropdown to the 'Deployment Config Details' heading. A red box highlights the 'DC rocky' link in the breadcrumb trail. A red arrow labeled '4' points from the 'DC rocky' link to the bottom right corner of the page.

**Deployment Config Details**

Latest Version
2

**Message**  
config change

**Update Strategy**  
Rolling

**Min Ready Seconds**  
Not Configured

**Triggers**  
ConfigChange

Project: broker

0 scaling to 1

Name: rocky

Namespace: NS broker

Labels: No labels

Pod Selector: app=rocky

Node Selector: No selector

Tolerations: 0 Tolerations

Edit Tolerations

No Tolerations Found

Add More

Cancel Save

Min Ready Seconds: Not Configured

Triggers: ConfigChange

Edit Tolerations

KEY	OPERATOR	VALUE	EFFECT
node	Equal	worker	NoSched...

+ Add More

1 2 3 4

put

Cancel Save

**DC** rocky
[Details](#) [YAML](#) [Replication Controllers](#) [Pods](#) [Environment](#) [Events](#)

## Deployment Config Details



*rolling update*

Name	rocky	Latest Version	3
Namespace	NS broker	Message	config change
Labels	No labels	Update Strategy	Rolling
Pod Selector	app=rocky	Min Ready Seconds	Not Configured
Node Selector	No selector	Triggers	ConfigChange
Tolerations	[Toleration]		

**DC** rocky
[Details](#) [YAML](#) [Replication Controllers](#) [Pods](#) [Environment](#) [Events](#)

## Deployment Config Details



Name  
rocky

Namespace  
NS broker

Labels  
No labels

Pod Selector  
app=rocky

Node Selector  
No selector

Tolerations  
[Toleration]

## **Step 4: delete precreated route**

```
$ oc get routes
NAME      HOST/PORT          PATH  SERVICES  PORT  TERMINATION
WILDCARD
rocky    rocky-broker.2886795298-80-host19nc.environments.katacoda.com  rocky     8080-tcp
None
$
```

**# oc delete routes.route.openshift.io rocky**

**Step 5: create new route using the given hostname**

**# oc expose svc rocky --hostname rocky-broker.ocp4.apps.example.com**

**Step 6: click on the link in the question and see if getting output if yes then great and 100% correct.**

## **Q8. Manually scale the application**

**[hint: should do on web console and cli, apply tolerations]**

- 
- a. An application minion in running on project gru make sure the application is running
  - b. The application should produce output
- 

**Ans 8:**

**Step 1: go to web-console and go to dc and apply the toleration**

Red Hat  
OpenShift  
Container Platform

Administrator

Home

Operators

Workloads

1 Pods

1 Deployments

Deployment Configs

Stateful Sets

Secrets

Config Maps

Cron Jobs

Jobs

Daemon Sets

Replica Sets

Replication Controllers

Horizontal Pod Autoscalers

Networking

Storage

Builds

Deployment Configs > Deployment Config Details

DC

Details YAML Replication Controllers Pods Environment Events

Deployment Config Details

0 scaling to 1

Name: rocky

Namespace: NS ← project

Labels: No labels

Pod Selector: app=rocky

Node Selector:

Tolerations: 0 Tolerations ↗ click

Latest Version: 2

Message: config change

Update Strategy: Rolling

Min Ready Seconds: Not Configured

Triggers: ConfigChange

Project: broker

0 scaling to 1

Name: rocky

Namespace: NS broker

Labels: No labels

Pod Selector: app=rocky

Node Selector: No selector

Tolerations: 0 Tolerations ↗

Edit Tolerations

+ Add More

No Tolerations Found

Cancel Save

Min Ready Seconds: Not Configured

Triggers: ConfigChange

## Edit Tolerations

KEY

OPERATOR

VALUE

EFFECT

node

Equal

worker

NoSched...



+ Add More

put



Cancel

Save

Deployment Configs > Deployment Config Details

DC

Details

YAML

Replication Controllers

Pods

Environment

Events

### Deployment Config Details



2.

rolling update

Name  
rocky

Latest Version  
3

Namespace  
NS

Message  
config change

Labels  
No labels

Update Strategy  
Rolling

Pod Selector  
app

Min Ready Seconds  
Not Configured

Node Selector  
No selector

Triggers  
ConfigChange

Tolerations

[Toleration](#)



After applying toleration



Details   YAML   Replication Controllers   Pods   Environment   Events

### Deployment Config Details



Name



L

3

Namespace

NS

N

C

Labels

No labels

L

E

Pod Selector



app=

N

N

Node Selector

No selector

T

C

Tolerations



T

C

# go back to terminal now

# oc project gru

# oc scale deploymentconfigs/minion --replicas=5 -n gru

# oc get pods

Output > if showing 5 pods running then all correct

# oc get routes

> copy the route and open in firefox for final testing.

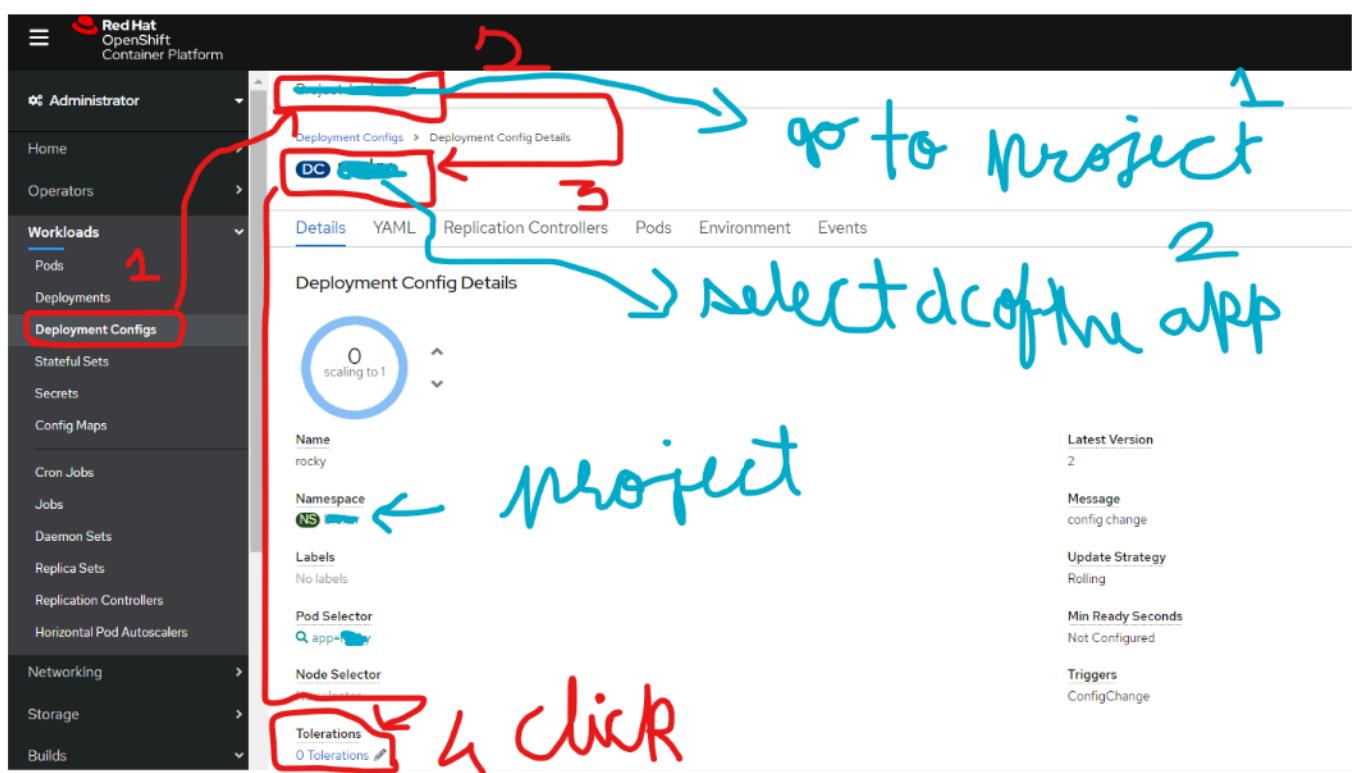
# Q9. Autoscale the application

[hint: should do on web console and cli, apply tolerations]

- a. An application crab in running on project seawater make sure the application is running
- b. The application should be auto scaled with 6 pods as minimum and should have at max 9 pods limit and cpu percentage should be 60 %.
- c. The application should have the limit of 100m and request 10m for the cpu
- d. Application should be up and running and produces and output.

## Ans 9:

Step 1: go to web-console and go to dc and apply the toleration



### Edit Toleration

No Toleration Found

Add More

Name: rocky

Namespace: NS

Labels: No labels

Pod Selector: app=

Node Selector: No selector

Tolerations: 0 Toleration

Min Ready Seconds: Not Configured

Triggers: ConfigChange

Cancel Save

A large blue arrow points from the 'Add More' button at the top right towards the 'Tolerations' section at the bottom left.

### Edit Toleration

KEY	OPERATOR	VALUE	EFFECT
node	Equal	worker	NoSched...

+ Add More

1 2 3 4

put

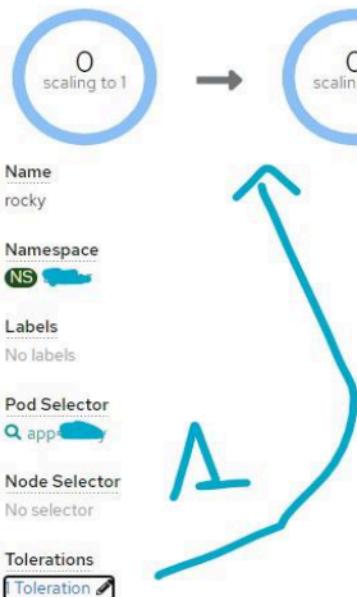
Cancel Save

A large blue arrow points from the 'Save' button at the bottom right towards the 'Put' label at the bottom left.



Details   YAML   Replication Controllers   Pods   Environment   Events

## Deployment Config Details



Name

rocky

Namespace

NS

Labels

No labels

Pod Selector

app=

Node Selector

No selector

Tolerations

1 Toleration

rolling update  
After applying toleration

Latest Version

3

Message

config change

Update Strategy

Rolling

Min Ready Seconds

Not Configured

Triggers

ConfigChange



Details   YAML   Replication Controllers   Pods   Environment   Events

## Deployment Config Details



Name

Namespace

NS

Labels

No labels

Pod Selector

app=

Node Selector

No selector

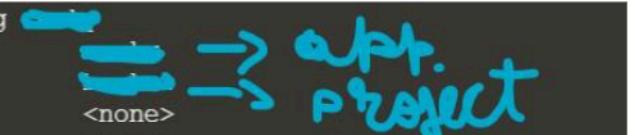
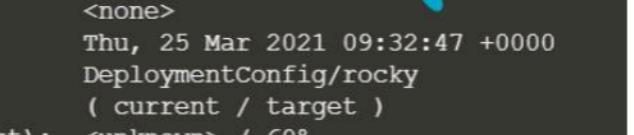
Tolerations

1 Toleration

```
# go back to terminal now  
# oc project seawater  
# oc autoscale deploymentconfigs/crab --min=6 --max=9 --cpu-percent=60 -n seawater  
# oc set resources deploymentconfigs/crab --limits(cpu)=100m --requests(cpu)=10m  
# oc get pods
```

**Output > if showing 6 pods running then all correct**

```
# oc describe horizontalpodautoscalers.autoscaling < press tab here two times then enter >
```

```
$ oc describe horizontalpodautoscalers.autoscaling   
Name:   
Namespace:  
Labels:  
Annotations:  
CreationTimestamp: Thu, 25 Mar 2021 09:32:47 +0000  
Reference: DeploymentConfig/rocky  
Metrics:  
  resource cpu on pods (as a percentage of request): <unknown> / 60%  
Min replicas: 6  
Max replicas: 9  
DeploymentConfig pods: 0 current / 0 desired  
Events: <none>  
Pods:
```

```
# oc get routes
```

> copy the route and open in firefox for final testing.

---

## Q10. Create secret

[hint: can do on web console or cli]

---

---

a. On project math create secret magic

b. Secret should have:

**Key:** decoder\_ring

**Value:** 07UMHrPv9KeGAN1yg33CcaeJKJycgT7zfjz=

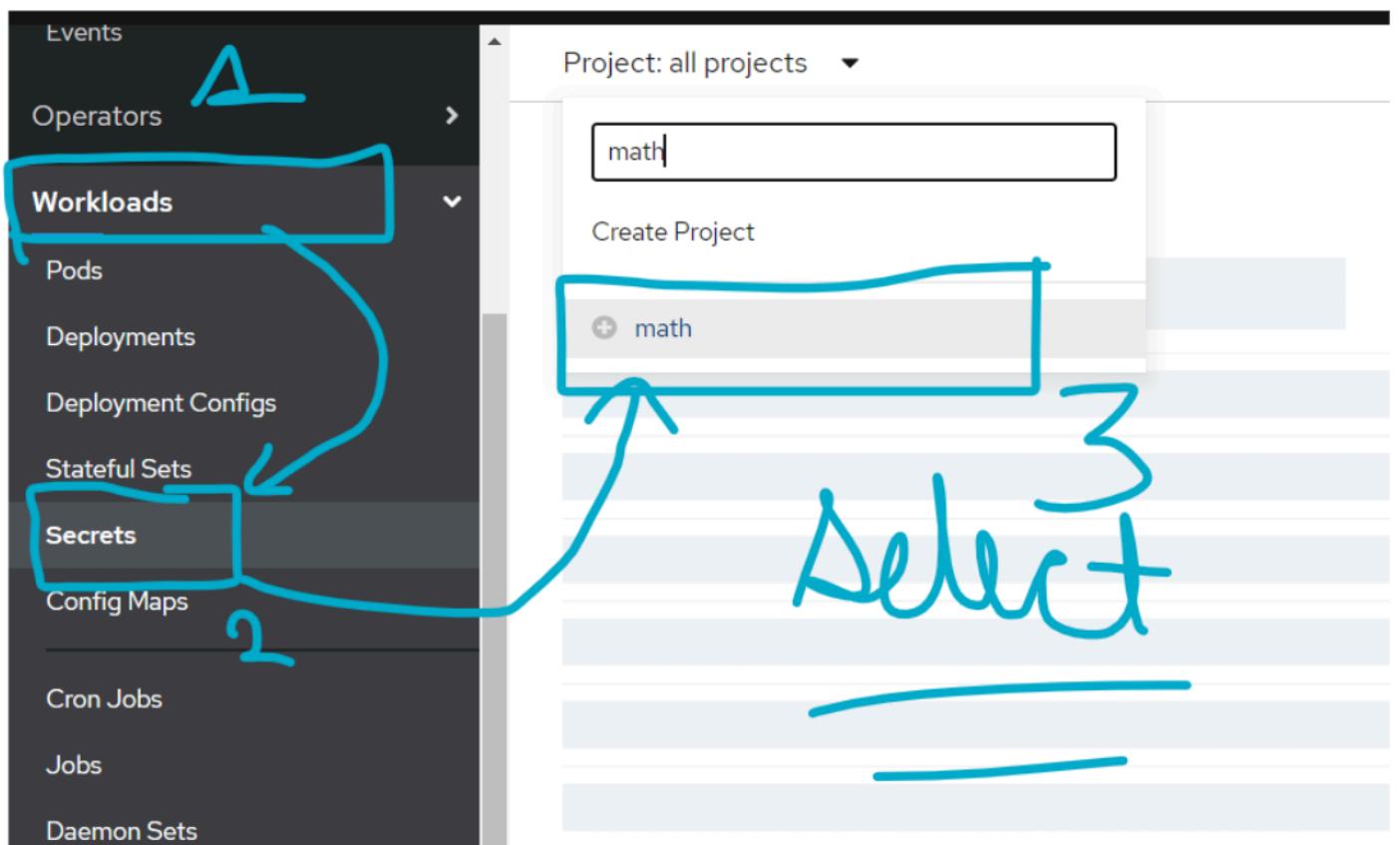
---

---

# Ans 10:

## GUI way:

### Step 1: Go to secrets and select math project



### Step 2: Create secret magic



Project: math

## Create Key/Value Secret

Key/value secrets let you inject sensitive data into your application as files or environment variables.

Secret Name \*

magic

Unique name of the new secret.

Key \*

decoder\_ring

Value

Drag and drop file with your value here or browse to upload it.

Browse...

07UMHrPv9KeGANIyg33CcaeJKJycgT7zfjz=

+ Add Key/Value

Create

Cancel

### Step 3: Check if secret is there in cli

# oc project math

# oc get secrets | grep magic

```
$ oc get secrets | grep magic
magic                         Opaque          1      42s
$
```

## CLI way:

```
# oc project math
```

```
# oc create secret generic magic --from-literal=decoder_ring=07UMHrPv9KeGAN1yg33CcaeJKJycgT7zfjz= -n math
```

```
# oc get secrets | grep magic
```

```
$ oc project math
Now using project "math" on server "https://openshift:6443".
$ oc create secret generic magic --from-literal=decoder_ring=07UMHrPv9KeGAN1yg33CcaeJKJycgT7zfjz= -n math
secret/magic created
$ oc get secrets | grep magic
magic          Opaque      1        11s
$
```

## Q11. Set application environment

[hint: should do on cli]

- |-----|
- a. On project math there's application running qed set the environment variable using the secret magic
  - b. Environmental name should be DECODER\_RING.
  - c. Application should no longer display "not properly configured"
- |-----|

Ans 11:

Step 1: Apply toleration:

# Step 1: go to web-console and go to dc and apply the toleration

The screenshot shows the Red Hat OpenShift web console interface. On the left, a sidebar menu under 'Workloads' has 'Deployment Configs' highlighted with a red box and labeled '1'. A blue arrow points from the sidebar to the main content area. In the main area, a 'Deployment Configs > Deployment Config Details' path is shown. A red box highlights the 'DC' tab in the navigation bar, with a blue arrow pointing to it and labeled '2'. Handwritten blue text says 'go to project' above the sidebar and 'select dc of the app' next to the 'DC' tab. Below the tabs, the 'Deployment Config Details' section shows a summary: '0 scaling to 1', 'Name: rocky', 'Namespace: NS', 'Labels: No labels', 'Pod Selector: app=rocky', 'Node Selector: No selector', 'Tolerations: 0 Tolerations'. Handwritten blue text says 'project' next to the namespace and 'click' next to the 'Tolerations' section.

The screenshot shows the 'Edit Tolerations' dialog box. At the top, it says 'Project: broker'. The main area displays 'No Tolerations Found'. A large blue arrow points from the 'Tolerations' section in the background to the 'Add More' button in the dialog. Handwritten blue text says 'click' next to the 'Add More' button. At the bottom right of the dialog are 'Cancel' and 'Save' buttons. The background shows the deployment config details with the 'Tolerations' section highlighted.

## Edit Tolerations

KEY

OPERATOR

VALUE

EFFECT

node

Equal

worker

NoSched...



+ Add More

put



Cancel

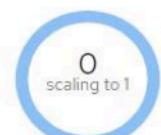
Save

Deployment Configs > Deployment Config Details

DC

Details   YAML   Replication Controllers   Pods   Environment   Events

### Deployment Config Details



0 scaling to 1



0 scaling to 1

rolling update

Name

rocky

Namespace

NS

Labels

No labels

Pod Selector

app

Node Selector

No selector

Tolerations

[ Tolerations ]



After applying toleration

Latest Version

3

Message

config change

Update Strategy

Rolling

Min Ready Seconds

Not Configured

Triggers

ConfigChange



Details   YAML   Replication Controllers   Pods   Environment   Events

### Deployment Config Details



Name

Namespace

Labels

No labels

Pod Selector

app=

Node Selector

No selector

Tolerations

Toleration

## Step 2: add environmental variable

# oc project math

# oc set env deploymentconfigs/qed --from=secrets/magic

# Q12. Deploy the application with secured route

[hint: should do on gui and cli]

a. On project area51 there's app running oxcart

b. Make sure app is up and running.

c. Create secured route oxcart and with details:

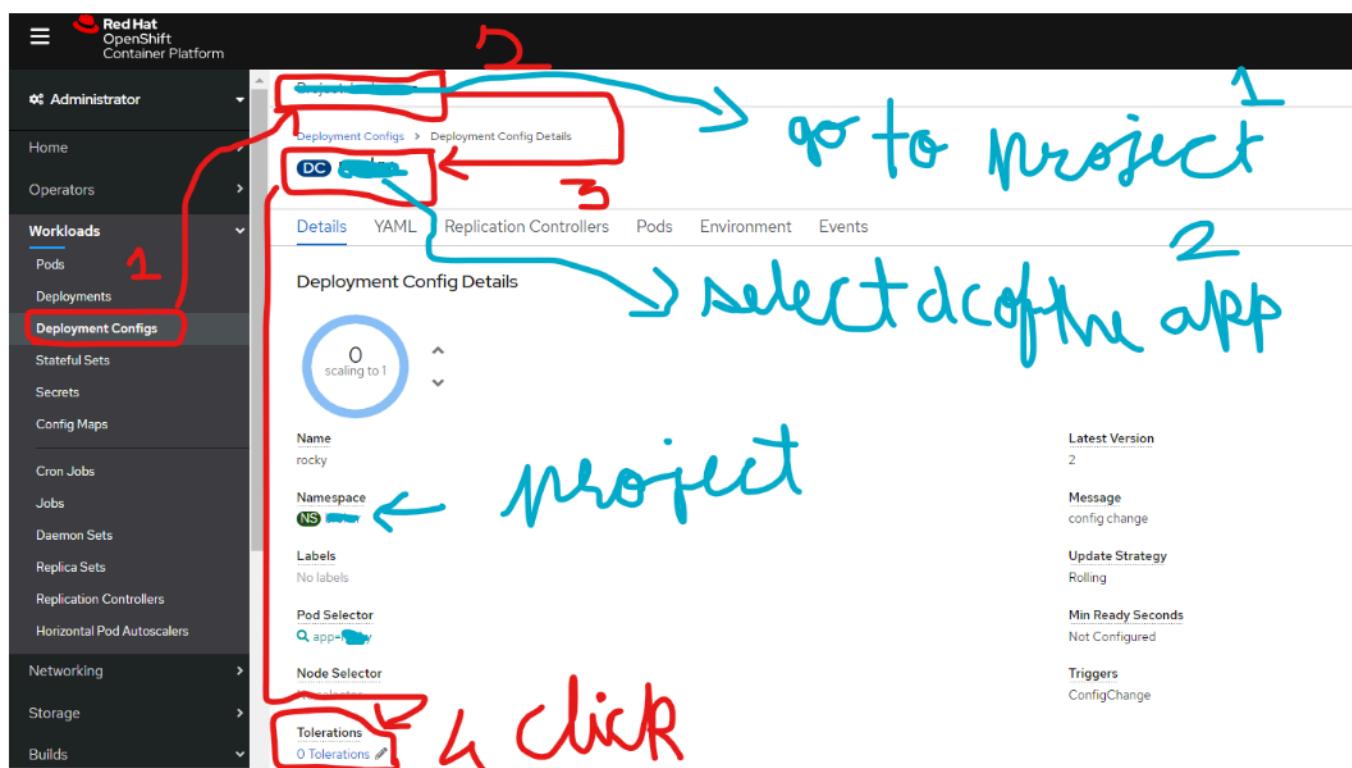
/C=SI/ST=Ljubljana/L=Ljubljana/O=Security/OU=IT Department/CN=secured.apps.ocp4.example.com

d. Application should be accessible, up and running and should get output using the link <https://secured.apps.ocp4.example.com>

Ans 12:

Step 1: Apply toleration

Step 1: go to web-console and go to dc and apply the toleration



Project: broker

0 scaling to 1

Name: rocky

Namespace: NS

Labels: No labels

Pod Selector: app=rocky

Node Selector: No selector

Tolerations: 0 Tolerations

No Tolerations Found

Add More

Cancel Save

Min Ready Seconds: Not Configured

Triggers: ConfigChange

## Edit Tolerations

KEY	OPERATOR	VALUE	EFFECT
node	Equal	worker	NoSched...

+ Add More

1 2 3 4

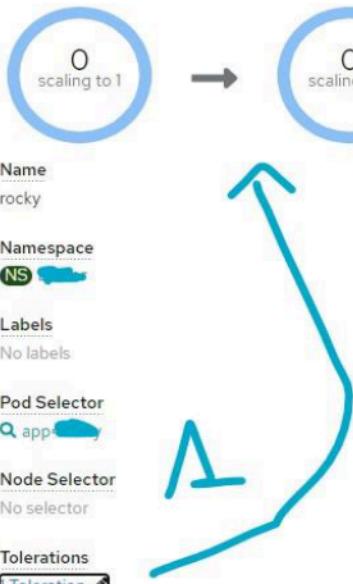
put

Cancel Save



Details YAML Replication Controllers Pods Environment Events

## Deployment Config Details

Name  
rockyNamespace  
NSLabels  
No labelsPod Selector  
app=Node Selector  
No selectorTolerations  
[Toleration]

rolling update  
After applying toleration

Latest Version  
3Message  
config changeUpdate Strategy  
RollingMin Ready Seconds  
Not ConfiguredTriggers  
ConfigChange

Details YAML Replication Controllers Pods Environment Events

## Deployment Config Details

Name  
[redacted]Namespace  
NSLabels  
No labelsPod Selector  
app=Node Selector  
No selectorTolerations  
[Toleration]L  
3N  
CL  
EN  
HT  
C

## **Step 2: Now proceed with the certificate creation**

```
# oc project area51
# yum install openssl -y
# openssl genrsa -out a.key 2048
# openssl req -new -key a.key -out a.csr
# openssl x509 -req -in a.csr -out a.crt -days 365 -signkey a.key
```

```
$ openssl genrsa -out a.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
$ openssl req -new -key a.key -out a.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:SI
State or Province Name (full name) []:Ljubljana
Locality Name (eg, city) [Default City]:Ljubljana
Organization Name (eg, company) [Default Company Ltd]:Security
Organizational Unit Name (eg, section) []:IT Department
Common Name (eg, your name or your server's hostname) []:secured.apps.ocp4.example.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
$
```

```
$ openssl x509 -req -in a.csr -out a.crt -days 365 -signkey a.key
Signature ok
subject=C = SI, ST = Ljubljana, L = Ljubljana, O = Security, OU = IT Department, CN = secured.apps.ocp4.example.com
Getting Private key
$
```

```
# oc create secret tls tls-secret --key a.key --cert a.crt
```

```
$ oc create secret tls tls-secret --key a.key --cert a.crt
secret/tls-secret created
$
```

```
##### delete pre-created route #####
```

```
# oc get routes
```

\$ oc get routes		NAME	HOST/PORT	PATH	SERVICES	PORT
ERMINATION	WILDCARD					
oxcart	oxcart-area51.2886795285-80-host04nc.environments.katacoda.com				oxcart	8080-tcp
	None					

```
# oc delete routes.route.openshift.io oxcart  
# oc create route edge oxcart --service oxcart --hostname  
secured.apps.ocp4.example.com --key a.key --cert a.crt
```

**Step 3: check the link it should be accessible now**

## Q13. Create Service Account and configure it

[hint: should do on cli]

|-----|

- a. On project apples create service account ex280-sa
- b. Any user should be able to access the service account

|-----|

### Ans 13:

#### Step 1: switch to project apples

```
# oc project apples
```

```
# oc create sa ex280-sa
```

```
# oc adm policy add-scc-to-user anyuid -z ex280-sa
```

# Q14. Deploy the application

[hint: should do it on gui and cli]

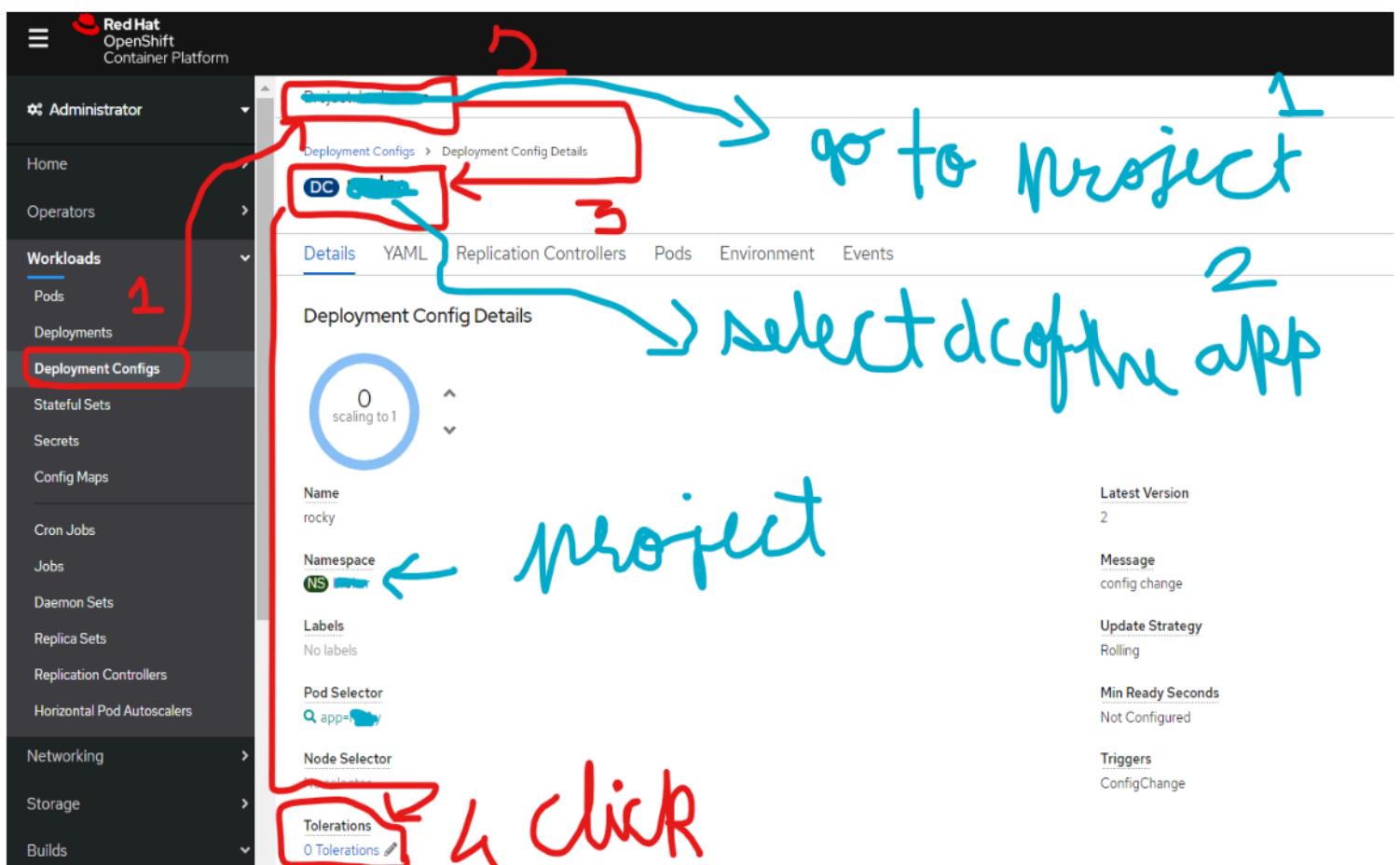
- a. There's application oranges running on the apples project
- b. Add the service account

Ans 14:

Step 1: switch to project apples

# oc project apples

**Step 1: go to web-console and go to dc and apply the toleration**



### Edit Tolerations

No Tolerations Found

+ Add More

Name: rocky

Namespace: NS

Labels: No labels

Pod Selector: app=

Node Selector: No selector

Tolerations: 0 Tolerations

Min Ready Seconds: Not Configured

Triggers: ConfigChange

Cancel Save

A screenshot of a Kubernetes UI showing the 'Edit Tolerations' dialog. The dialog has a header 'Edit Tolerations' and a message 'No Tolerations Found'. It contains a button '+ Add More' with a blue arrow pointing to it. Below are fields for 'Name' (set to 'rocky'), 'Namespace' (set to 'NS'), 'Labels' (empty), 'Pod Selector' (with a search bar containing 'app='), 'Node Selector' (empty), and 'Tolerations' (showing '0 Tolerations'). On the right, there are sections for 'Min Ready Seconds' (set to 'Not Configured') and 'Triggers' (set to 'ConfigChange'). At the bottom are 'Cancel' and 'Save' buttons.

### Edit Tolerations

KEY	OPERATOR	VALUE	EFFECT
node	Equal	worker	NoSched...

+ Add More

put 1 2 3 4

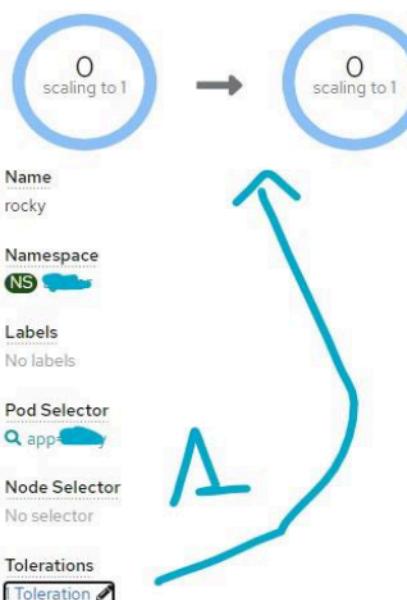
Cancel Save

A screenshot of the same 'Edit Tolerations' dialog, but now containing four tolerance entries. Each entry consists of four fields: KEY (set to 'node'), OPERATOR (set to 'Equal'), VALUE (set to 'worker'), and EFFECT (set to 'NoSched...'). Above these entries, the text '+ Add More' is visible. To the left of the entries, the word 'put' is written in large blue letters, with arrows pointing from each letter to the corresponding row. The bottom right corner shows 'Cancel' and 'Save' buttons.



Details   YAML   Replication Controllers   Pods   Environment   Events

### Deployment Config Details



After applying toleration

rolling update

D.

Latest Version: 3

Message: config change

Update Strategy: Rolling

Min Ready Seconds: Not Configured

Triggers: ConfigChange



Details   YAML   Replication Controllers   Pods   Environment   Events

### Deployment Config Details



Name



L

3

Namespace



M

C

Labels

No labels

U

F

Pod Selector



N

N

Node Selector

No selector

T

C

Tolerations



## **Step 2:**

**# oc project apples**

**# oc set sa deploymentconfigs/oranges ex280-sa**

**# oc edit deploymentconfigs/oranges**

**-> change orange to oranges in deploymentconfigs**

**# oc edit svc/oranges**

**-> change orange to oranges in svc file**

**# oc get routes**

**-> open the url in browser you should get output “There is no comparision”**

## **Q15. Deploy the application**

**[hint: should do it on gui and cli]**

- |-----|
  - a. There's application running mars on project pathfinder**
  - b. Make sure application produce output and is up and running and no additional resources are added or removed**

|-----|

**Ans 15:**

**Step 1:**

**Step 1: go to web-console and go to dc and apply the toleration**

1. Go to project

2. Select dc of the app

3. Click

4. Click

5. Click

The screenshot shows the Red Hat OpenShift Container Platform interface. On the left, the navigation sidebar is open, showing 'Workloads' selected. Under 'Deployment Configs', the 'Deployment Configs' tab is highlighted with a red box and labeled '1'. The main content area shows a 'Deployment Config Details' page for a deployment config named 'rocky'. The 'Tolerations' section is highlighted with a red box and labeled '4 click'. A large blue arrow points from the sidebar to the main content area, labeled '5 click'. Handwritten annotations include 'go to project' with a red arrow pointing to the top right, 'select dc of the app' with a red arrow pointing to the 'Deployment Configs' tab, and 'project' with a red arrow pointing to the 'Namespace' field.

1. Add More

2. Cancel

3. Save

The screenshot shows the 'Edit Tolerations' dialog box. At the top, it says 'Edit Tolerations'. Below that, it says 'No Tolerations Found'. In the center, there is a button labeled '+ Add More' with a blue box around it and a blue arrow pointing to it. At the bottom right, there are two buttons: 'Cancel' and 'Save'.

## Edit Tolerations

KEY	OPERATOR	VALUE	EFFECT
node	Equal	worker	NoSched...

+ Add More

put [ 1 2↑ 3↓ 4↓ ]

Cancel Save

Deployment Configs > Deployment Config Details



Details YAML Replication Controllers Pods Environment Events

### Deployment Config Details



Name  
rocky

Namespace  
NS

Labels  
No labels

Pod Selector  
app

Node Selector  
No selector

Tolerations

[ Tolerations ]

rolling update  
after applying toleration

Latest Version  
3

Message  
config change

Update Strategy  
Rolling

Min Ready Seconds  
Not Configured

Triggers  
ConfigChange

DC

[Details](#) [YAML](#) [Replication Controllers](#) [Pods](#) [Environment](#) [Events](#)
**Deployment Config Details****Name**

L

3

**Namespace**

N

C

**Labels**

No labels

L

F

**Pod Selector**

P

N

**Node Selector**

No selector

T

C

**Tolerations**

T

C

**Step 2: go to dc of the application in gui**

The screenshot shows the OpenShift web interface with a dark theme. On the left, there is a sidebar with the following navigation:

- Administrator** (dropdown)
- Workloads** (selected)
- Pods**
- Deployments**
- Deployment Configs** (selected)
- Stateful Sets**
- Secrets**
- Config Maps**
- Events**
- Operators**

The main content area shows a deployment config named 'mars' in the 'pathfinder' project. The 'Deployment Config Details' section includes the following information:

- Scaling:** 0 scaling to 1
- Name:** mars
- Namespace:** NS pathfinder
- Labels:** No labels
- Pod Selector:** app=mars
- Node Selector:** star=trek
- Latest Version:** 1
- Message:** config change
- Update Strategy:** Rolling
- Min Ready Seconds:** Not Configured
- Triggers:** ConfigChange

Annotations on the screenshot highlight specific steps:

- An orange box surrounds the 'Workloads' section in the sidebar.
- An orange box surrounds the 'Deployment Configs' section in the sidebar.
- A red box highlights the 'Project: pathfinder' dropdown in the top right.
- A large red box highlights the entire 'Deployment Config Details' section.
- An orange arrow points from the 'Deployment Configs' button in the sidebar to the 'Deployment Configs' section in the main content.
- An orange arrow points from the 'Deployment Configs' section in the main content to the 'Node Selector' field.

### Step 3:

Go back to cli :-

```
# oc project pathfinder
```

```
# oc get nodes -L star
```

```
$ oc get nodes -L star
NAME           STATUS   ROLES      AGE    VERSION
crc-fd5nx-star-0   Ready    worker   239d   v1.18.3+012b3ec
$
```

STAR Trek  
↑  
worker Node ↑

```
# go back to dc and change trek to Trek in dc
```

Project: pathfinder → 1

Deployment Configs > Deployment Config Details

DC mars ← DC 2

Details YAML Replication Controllers Pods Environment Events

```
119     app: mars
120   spec:
121     containers:
122       - name: mars
123         image: 'image-registry.openshift-image-registry.svc:5000/openshift/httpd:2.4'
124         ports:
125           - containerPort: 8080
126             protocol: TCP
127         resources: {}
128         terminationMessagePath: /dev/termination-log
129         terminationMessagePolicy: File
130         imagePullPolicy: IfNotPresent
131         restartPolicy: Always
132         terminationGracePeriodSeconds: 30
133         dnsPolicy: ClusterFirst
134         nodeSelector:
135           star: Trek
136         securityContext: {}
137         schedulerName: default-scheduler
138       status:
139         latestVersion: 1
140         observedGeneration: 1
141         replicas: 0
142         updatedReplicas: 0
143         availableReplicas: 0
144         unavailableReplicas: 0
145         details:
146           message: config change
```

→ 3 changed to Trek from trek

Save Reload Cancel → 4 → 5

DC mars

Details   YAML   Replication Controllers   Pods   Environment   Events

Deployment Config Details

1 pod

Name: mars

Namespace: pathfinder

Labels: No labels

Pod Selector: app=mars

Node Selector: star=Trek

*Greeting Pod*

*changed selector*

#### Step 4: get the output of the application

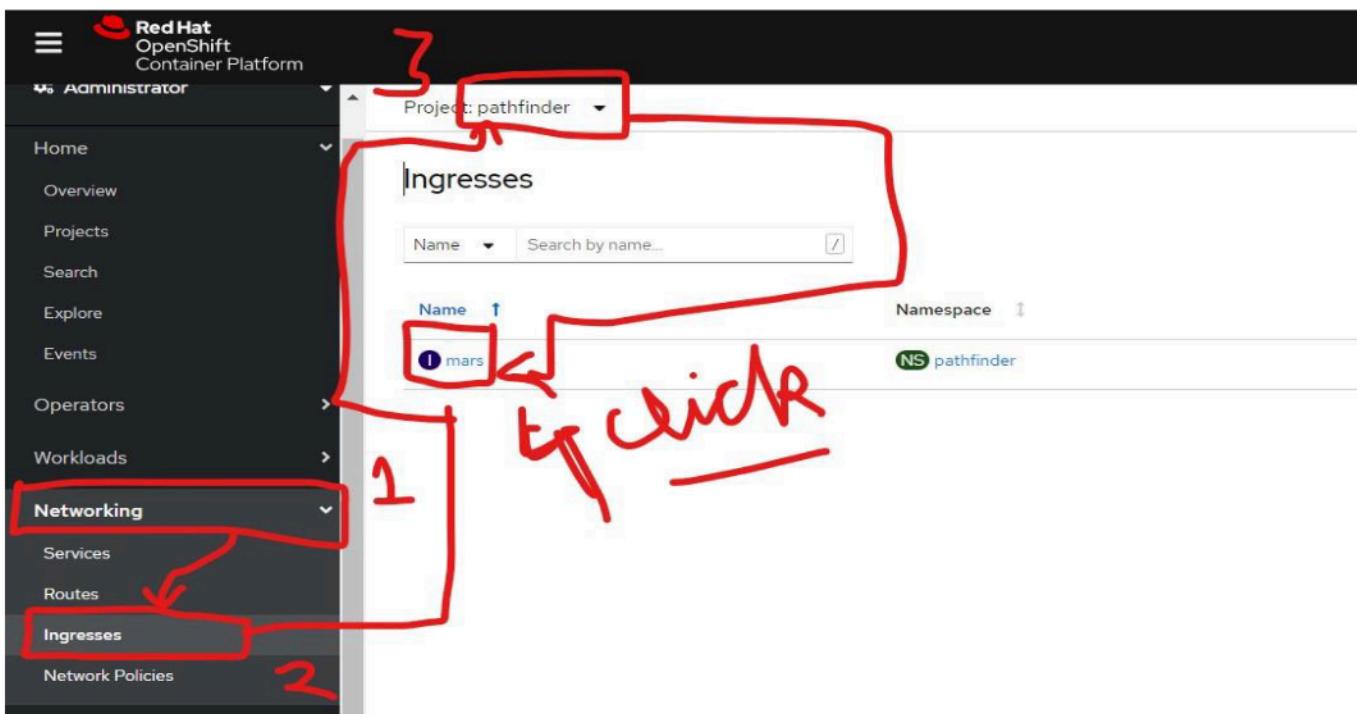
```
# oc get routes
```

Output >

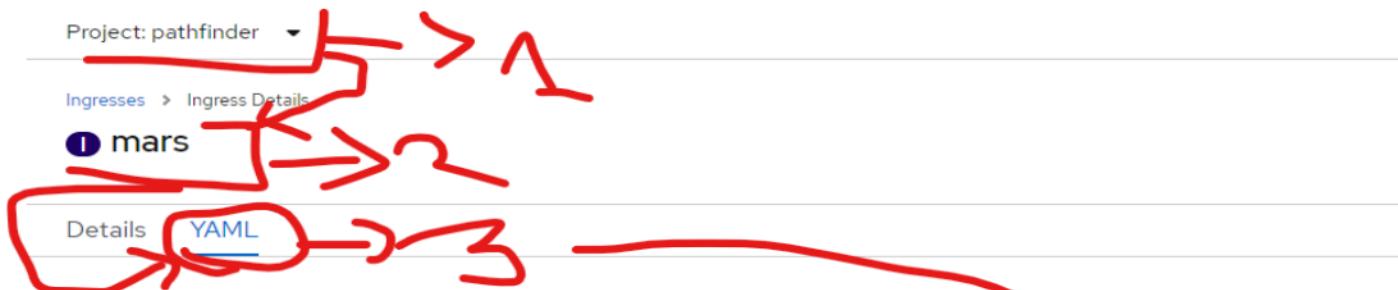
NAME	HOST/PORT	PATH	SERVICES	PORT
mars	mars-pathfinder.ocp4.app.example.com		mars	8080-tcp

## **Step 5: go to gui in network ingress file of the project pathfinder**

## **Change app to apps in host variable**



--->>>>>Change the network ingress yml file



-----→>>>>>>To below content

Project: pathfinder ▾

Ingresses > Ingress Details

1 mars

Details YAML

```
1 kind: Ingress
2 apiVersion: networking.k8s.io/v1beta1
3 metadata:
4   name: mars
5   namespace: pathfinder
6   selfLink: /apis/networking.k8s.io/v1beta1/namespaces/pathfinder/ingresses/mars
7   uid: 47bb4ae6-a9e5-49f4-86cd-b407f395e025
8   resourceVersion: '300300'
9   generation: 2
10  creationTimestamp: '2021-03-25T11:42:44Z'
11  managedFields:
12    - manager: Mozilla
13      operation: Update
14      apiVersion: networking.k8s.io/v1beta1
15      time: '2021-03-25T11:45:49Z'
16      fieldsType: FieldsV1
17      fieldsV1:
18        'f:spec':
19          'f:rules': {}
20  spec:
21    rules:
22      - host: mars-pathfinder.ocp4.apps.example.com
23        http:
24          paths:
```

binal Yer  
Ingress

**Step 6: wait for 5-6 mins and then:**

**# oc get routes**

**Output >**

NAME	HOST/PORT	PATH	SERVICES	PORT
mars	mars-pathfinder.ocp4.apps.example.com		mars	8080-tcp

**Step 7: open the <https://mars-pathfinder.ocp4.apps.example.com>**

**In browser and should get output like “ we are trying to get you back home ”**

---

|-----|

## **Q16. Deploy the application**

**[hint: should do it on gui and cli]**

|-----|

- a. There's application running helm on project working**
- b. Make sure application produce output and is up and running and no additional resources are added or removed**

|-----|

### **Ans 16:**

**>>> in cli**

**# oc project working**

**>>> in gui**

**Apply tolerations**

## Step 1: go to web-console and go to dc and apply the toleration

The screenshot shows the Red Hat OpenShift Container Platform web interface. On the left, the navigation sidebar is open, showing various workloads like Pods, Deployments, and Deployment Configs. A red box labeled '1' highlights the 'Deployment Configs' option. The main content area shows a 'Deployment Config Details' page for a deployment named 'rocky'. A blue arrow labeled '2' points from the top of the page to the 'Deployment Configs' link in the breadcrumb. Another blue arrow labeled '3' points from the breadcrumb to the 'DC' button. Handwritten blue text on the right says 'go to project' with a circled '1' above it, and 'select dc of the app' with a circled '2' below it. A blue arrow labeled 'project' points from the handwritten text to the 'Namespace' field ('NS'). Handwritten blue text 'click' with a circled '4' points to the 'Tolerations' section at the bottom of the page.

This screenshot shows the 'Edit Tolerations' dialog box overlaid on the main deployment configuration page. The dialog has a title 'Edit Tolerations' and a message 'No Tolerations Found'. It features a button '+ Add More' highlighted with a blue box and a circled '1'. In the bottom right corner of the dialog are 'Cancel' and 'Save' buttons. The background of the main page shows the deployment configuration details for 'rocky' with a 'Tolerations' section containing '0 Tolerations'.

## Edit Tolerations

KEY	OPERATOR	VALUE	EFFECT
node	Equal	worker	NoSched...

+ Add More

put 1↑ 2↑ 3↑ 4↑

Cancel Save

Deployment Configs > Deployment Config Details



Details YAML Replication Controllers Pods Environment Events

### Deployment Config Details



Name  
rocky

Namespace  
NS

Labels  
No labels

Pod Selector  
app

Node Selector  
No selector

Tolerations  
Toleration

rolling update  
A after applying toleration

Latest Version  
3

Message  
config change

Update Strategy  
Rolling

Min Ready Seconds  
Not Configured

Triggers  
ConfigChange



Details   YAML   Replication Controllers   Pods   Environment   Events

## Deployment Config Details

**1 pod**

**Name** [redacted] L 3

**Namespace** NS M C

**Labels** No labels U F

**Pod Selector** app=M N H

**Node Selector** No selector T C

**Tolerations** 1 Toleration E

## Step 2: Go to DC of the application helm in project working

The screenshot shows the Red Hat OpenShift Container Platform interface. A red box labeled '1' highlights the 'Workloads' menu item in the sidebar. A red box labeled '2' highlights the 'Deployment Configs' sub-item under 'Workloads'. A red box labeled '3' highlights the 'Project: working' dropdown in the top header bar. The main content area displays the 'Deployment Config Details' for the 'helm' deployment config, which has '1 pod'.

**Red Hat OpenShift Container Platform**

**Administrator**

- Home
- Operators
- Workloads**
  - Pods
  - Deployments
  - Deployment Configs**
- Stateful Sets
- Secrets
- Config Maps

**Cron Jobs**

**Jobs**

**Daemon Sets**

**Replica Sets**

**Replication Controllers**

**Horizontal Pod Autoscalers**

**Networking**

**Storage**

**Deployment Configs** > Deployment Config Details

**DC helm**

**Details**   **YAML**   Replication Controllers   Pods   Environment   Events

**Deployment Config Details**

**1 pod**

**Name** helm Lates 1

**Namespace** NS working Message config

**Labels** No labels Update Rolling

**Pod Selector** app=helm Min R Not C

**Node Selector** Trigger Config

**DC helm**

Details **YAML** Replication Controllers Pods Environment Events

```

135 test: false
136 selector:
137   app: helm
138 template:
139   metadata:
140     creationTimestamp: null
141   labels:
142     app: helm
143 spec:
144   containers:
145     - name: helm
146       image: 'image-registry.openshift-image-registry.svc:5000/openshift/httpd:2.4'
147       ports:
148         - containerPort: 8080
149           protocol: TCP
150       resources:
151         limits:
152           cpu: 100m
153           memory: 100Gi
154         requests:
155           cpu: 10m
156           memory: 80Gi
157       terminationMessagePath: /dev/termination-log
158       terminationMessagePolicy: File
159       imagePullPolicy: IfNotPresent
160   restartPolicy: Always
161   terminationGracePeriodSeconds: 30
162   dnsPolicy: ClusterFirst
163   volumeMounts:

```

1

2

Change +8  
2 Gi

**DC helm**

Details **YAML** Replication Controllers Pods Environment Events

```

135 test: false
136 selector:
137   app: helm
138 template:
139   metadata:
140     creationTimestamp: null
141   labels:
142     app: helm
143 spec:
144   containers:
145     - name: helm
146       image: 'image-registry.openshift-image-registry.svc:5000/openshift/httpd:2.4'
147       ports:
148         - containerPort: 8080
149           protocol: TCP
150       resources:
151         limits:
152           cpu: 100m
153           memory: 100Gi
154         requests:
155           cpu: 10m
156           memory: 1Gi
157       terminationMessagePath: /dev/termination-log
158       terminationMessagePolicy: File
159       imagePullPolicy: IfNotPresent
160   restartPolicy: Always
161   terminationGracePeriodSeconds: 30
162   dnsPolicy: ClusterFirst
163   volumeMounts:

```

like this

**Save****Reload****Cancel**

### **Step 3: get to the cli**

**# oc get routes**

**Out put > <url of the application output>**

**Step 4: copy the url and paste in firefox and check if out put coming or not out put should be like “ mars is the project .....**

**If there's output then great all done.**

### **IMPORTANT points-**

- after deleting kubeadmin secret make sure to login as jobs user in the web console**
- apply toleration to all the questions**
- in applying sa make sure to set service account after toleration then do the rest like checking dc and svc do check them after applying toleration**
- always first switch in project before creating resources.**
- loggin to workbench from root user**

## **Q17. Deploy an application**

**Deploy the chart name redhat-cenima in the project ascii-hall from the repository  
<http://helm.domain.23.example.com/charts/>**

**You may sure telnet or nc command to validate the deployment**

**Ans 17.**

```
# oc project ascii-hall (If already exists)
# helm repo add do280-repo http://helm.domain.23.example.com/charts/
# helm search repo --versions
# helm install redhat-cenima do280-repo http://helm.domain.23.example.com/charts/
# oc get all
```

## **Q18. Inject configuration data:**

**Using the hello-openshift image, deploy and application that meets  
the following requirements**

- i. The application is part of a project named: czech
- j. The application is named: ernie
- k. The application looks for a key named: RESPONSE
- l. Configuration map named: ex280-cm
- m. Once deployed the application is running and available at  
<http://ernie.apps.ocp4.example.com>
- n. Re-deploying the application after making changes to the config map  
should result in a corresponding change to the displayed text!  
'six czech cricket critics'

**Ans 18 :**

```
# oc new-project czech
# oc new-app --name=ernie --image=quay.io/redhattraining/hello-openshift
// Registry server name information is given exam instruction link ,
# oc get pods
# oc get service
# oc expose service ernie --hostname=ernie.apps.ocp4.example.com

# oc get route
http://ernie.apps.ocp4.example.com
```

```
# oc create configmap ex280-cm --from-literal RESPONSE='six czech  
cricket critics'  
  
# oc get cm ex280-cm  
  
# oc describe cm ex280-cm  
  
# oc get pods  
# oc get cm ex280-cm  
  
# oc get deploy  
# oc describe pod ernie-96c76bc57-kllql | grep Env  
# oc set env --from configmap/ex280-cm deploy/ernie  
# oc get pod
```

#### **Q19. Configure a Network Policy**

**Configure a network policy using database and checker project with following requirements :**

**The database project has network policy with the name db-allow-mysql-comm based on pod selector label network.openshift.io/policy-group**

**Communications to the database project are restricted to deployments from the checker projects**

**The network policy is filtered by project selector using the team=devsecops label and pod checker using the deployment=web-mysql label**

**The application can establish a connections to port 3304/TCP**

**You can check your work by the logs in the checker project**

**Ans 19 :**

```
# oc project database  
  
# oc get pods  
  
# oc get route  
  
# oc describe pod mercury-58786b7869-66lw6 | grep Labels  
  
# oc describe pod mercury-58786b7869-66lw6 | grep Ports  
  
# oc describe pod rocky-74b5d6fd7d-cmhsw -n checker | grep Labels  
# oc describe project checker | grep Labels
```

```

# vim policy.yml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: db-allow-mysql-comm
spec:
  podSelector:
    matchLabels:
      network.openshift.io/policy-group: database
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              team: devsecops
        podSelector:
          matchLabels:
            deployment: web-mysql
        ports:
          - protocol: TCP
            port: 3304
    : wq!
# oc create -f policy.yml
# oc get networkpolicy
# oc describe networkpolicy db-allow-mysql-comm
# oc get pod -o wide
# oc project checker
# oc get pods

```

## **Q.20 . Persistent storage**

**Configure a persistent volume with following requirements**

**Name: landing-pv**  
**The access mode is ReadWriteMany**  
**Size: 1Gi**

**The reclaim policy matches the storage class**

**Configure a persistent volume claim with 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 following requirements:**

**The application exists in a project called page**

**The application uses a deployment called landing**

**The application uses the images located a**

**registry.domain12.example.com/nginxinc/nginx-unprivileged:latest**

**The nginx mountpoint in /usr/share/nginx/html**

**The application uses 3 pods**

**The application is accessible at**

**http://landing.page.apps.domain12.example.comDelete'**

**Ans 20:**

```
# oc new-project page
# oc new-app --name=landing --image=
registry.domain12.example.com/nginxinc/nginx-unprivileged:latestnx
# oc get pods
# oc get service
# oc expose service gamma --hostname=landing.page.apps.domain12.example.com
# oc get route
http://landing.page.apps.domain12.example.com
# oc get sc
// know the storage class name
# oc describe sc <storage class name>
// This command will help you to know the information like your shared nfs
information, reclaim policy information and more, Generally shared nfs information
available in Important.Configuration.Information
```

```
# vim pv.yaml
```

```
apiVersion: v1
```

```
kind: PersistentVolume
```

```
metadata:
```

```
  name: landing-pv
```

```
spec:
```

```
  capacity:
```

```
  storage: 1Gi
```

```
  accessModes:
```

```
- ReadWriteMany
persistentVolumeReclaimPolicy: Retain
nfs:
path: /exports-ocp4
server: 192.168.50.254
: wq!
# oc create -f pv.yaml
# oc get pv
# oc describe pv landing-pv
# vim pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
name: landing-pvc
spec:
accessModes:
- ReadWriteMany
resources:
requests:
storage: 1Gi
storageClassName: your-storage-class # Specify the storage class name here
# oc get pvc
# oc describe pvc landing-pvc
# oc get all |grep deploy
# oc edit deploy/landing
containers:
- name: nginx-container
image: registry.domain12.example.com/nginxinc/nginx-
unprivileged:latest
// ADD Following Parametters
volumeMounts:
- name: nginx-html
mountPath: /usr/share/nginx/html
volumes:
- name: nginx-html
persistentVolumeClaim:
claimName: landing-pvc
# oc get pods
# oc rsh landing-5c494cf5bf-d7jqt
```

## **Q.20 Install Operator**

**Install the file-integrity operator with following requirements:**

**The operator installed in the openshift-file-integrity project  
The approval strategy is Automatic  
Cluster monitoring is credit for the openshift-file-integrity  
Project**

### **Ans 20**

- 1.In the OpenShift Container Platform web console, navigate to Operators → OperatorHub.
- 2.Search for the File Integrity Operator, then click Install.
- 3.Keep the default selection of Installation mode and namespace to ensure that the Operator will be installed to the openshift-file-integrity namespace.
4. Click Install.

To confirm that the installation is successful:

Navigate to the Operators → Installed Operators page.

Check that the Operator is installed in the openshift-file-integrity namespace and its status is Succeeded.

## **Q21. Collect cluster Information for Red Hat support**

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

**The data is stored as 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  
This archive has been uploading for grading  
A utility script has been provided for you to upload the archive as follow:  
/usr/local/bin/upload-cluster-data ex280-ocp-clusterID.tar.gz**

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

**Ans 21 :**

```
# oc adm must-gather --dest-dir ex280-ocp
# oc get clusterversion // know clusterID
# tree ex280-clusterdata/
# tar -cvaf ex280-ocp.CLUSTER_ID.tar.gz ex280-ocp
# /usr/local/bin/upload-cluster-data ex280-ocp-clusterID.tar.gz

// This utility available in exam
```

## **Q22. Configure a health probe**

**An application named atlas has been deployed with single container is the mercury project.**

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

**The probe monitors liveness by performing a TCP socket check on port 8080**

**The probe has an initial delay of 10 second and a timeout of 30 second**

**Your changes can survive a rebuild**

Ans 22:

```
# oc project mercury
# oc get pods
# oc set probe --liveness --open-tcp 8080 --initial-delay-seconds 3 --timeout-
seconds 10 dc/atlas
# oc describe dc/atlas | grep Live
```