# EX288 v4.2 Feedback

Multiple candidates (Christopher Herbert, Robert Loesch, Deepak Kumbhar, Garrett Rabian) have reached 100% on the exam and the exam questions are well known.

**Initial Setup:**

ssh devop@workbench

#login using user “devop” and use the provided password

oc login -u ocpdevop -p <password> <https://api.master.domainX.example.com>

#see below on how to get this command

oc patch configs.imageregistry.operator.openshift.io/cluster --type merge -p '{"spec":{"defaultRoute":true}}'

sudo podman login -u regdevop -p <password> registry.domainX.example.com

#You will need to memories the following command

oc create secret generic registry /

--from-file .dockerconfigjson=${XDG\_RUNTIME\_DIR}/containers/auth.json /

--type kubernetes.io/dockerconfigjson

Command is available in **Build -> Creating Build Inputs -> Creating secrets** documentation.

Here is how to get to the “oc patch” command in the product documentation:

1. Click the link to the product documentation on the page with "Additional critical information"
2. You will see a list of product documentation that is organized like this: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.2/>
3. Find the entry for “Registry”
4. There will be html and pdf documentation. I used the html version
5. In the documentation for Registry, in Firefox search for "oc patch" or go directly to the section **2.2 Enable the Image Registry default route with the Custom Resource Definition**
6. Copy and paste the command to patch the Registry Operator CRD:

$ oc patch configs.imageregistry.operator.openshift.io/cluster --type merge -p '{"spec":{"defaultRoute":true}}'

1. After executing this command, you can then run oc get route -n openshift-image-registry and should get the route for the OpenShift internal registry.

**7 Questions**

1. Applications Deployment - Deploy nodejs application

**Solution Notes:**

You need to do a git clone add a comma in package.json and make sure to commit and push this in the

git repository. The comma is missing in the description line, but you can use “python -m json.tool

package.json” and it will show line number in vim you can do escape :<line number> to get to the line. Also

be familiar on how to use the --build-env with npm\_config\_registry as the application is a nodejs application.

The value for npm\_config\_registry is provided in the question.

Once the application is up and running, you are asked to add an entry in the browser and save the entry by

clicking the save icon on the application.

**Solution:**

oc new-project pastebin

git clone <http://git.domainX.example.com/pastebin.git>

cd xyz

python -m json.tool package.json

vim package.json

*add comma at the end of line 5 of package.json*

python -m json.tool package.json

git add .

git commit -a -m “Fixed package.json”

git push

oc new-app --name abc --build-env npm\_config\_registry=<provided npm URL> <http://git.domainX.example.com/pastebin.git>

oc expose svc abc --hostname <provided route>

oc get route

go to the URL in browser

type the provided text in the browser

save changes by clicking the save icon at the bottom

1. Update assemble script with date and message for info.html and deploy application using the rhscl/httpd-24-rhel7 builder image.  
     
   **Solution Notes:**

You need to do a git clone and you need to make sure the assemble script is correct and that all the files in .s2i/bin are executable. There is a customization section in the provided assemble script and you need to add a copy statement and generate an info.html page with the current date in yyyy-mm-dd format and text that is provided. Use DATE=$(date "+ %Y-%m-%d") to set the date.

The builder image is in the external registry. You need to import-image to import the builder image to the project. This requires the secret for the external registry and linking the secret to the **builder service account** on the project. You need to make sure you are logged into as sudo podman with oc id and token and then you can import-image from external repo. Then create a registry secret in the namespace and link it with builder service account otherwise you cannot use the httpd-24-rhel as a builder image

**Solution:**

oc new-project <project name>

git clone <http://git.domainX.example.com/oxy.git>

cd oxy

ls -a

cd .s2i/bin

chmod +x -Rf .s2i/bin

vim assemble

*cp /src/\*.html ./*

*Date=$(date “+%Y-%m-%d”)*

*echo “${Date}” >> info.html*

*echo “type the text provided” >> info.html*

git add .

git commit -a -m “Updated assemble script”

git push

oc create secret generic registry --from-file .dockerconfigjson=${XDG\_RUNTIME\_DIR}/containers/auth.json /

--type kubernetes.io/dockerconfigjson

Run scorpio if you can’t remember the Runtime\_Dir

[If you encounter the following error

error reading /run/user/1000/containers/auth.json cannot find file or directory

You might encounter this error if you have executed question 7 before question 2/3 which means if you have logged on to podman with another registry

Two ways to resolve

1. oc podman logout –all and rerun the above create secret command
2. oc create secret docker-registry registry --docker-username=regdevop

-–docker-password=passwordprovidedinexam registry.domainX.example.com] instead of the above secret command

oc secret link **builder** registry

oc import-image httpd-24-rhel /

--from=registry.doaminX.example.com**:443**/rhscl/httpd-24-rhel7 /

--confirm --reference-policy local

oc new-app --name <app name> /

httpd-24-rhel~http://git.domainX.example.com/oxy.git

oc expose svc <app name> --hostname <provided route>

oc get route

Open the URL in browser or curl

http://<route>/info.html

1. Using OpenShift ConfigMaps and Secrets - Deploy an application and inject a configmap

**Solution Notes:**

You are asked to deploy the **openshift/hello-openshift** image which is available in the external registry.

By default, when the application is up, it will display "Hello OpenShift".

You are asked to create a configmap “sedicen” with a RESPONSE key and text "Soda pop. …" and set it in the environment.

This will trigger a redeploy and the text on the page changes to "Soda pop ….".

Since the openshift/hello-openshift is in external registry and a container image, you need the following steps to deploy the application first before you can create configmap and set env:

**Solution:**  
oc new-project xyz

sudo podman login -u regdevop -p <password> **registry.doaminX.example.com**

oc create secret generic registry --from-file .dockerconfigjson=${XDG\_RUNTIME\_DIR}/containers/auth.json /

--type kubernetes.io/dockerconfigjson

Run scorpio if you can’t remember the Runtime\_Dir

[If you encounter the following error while creating the secret

error reading /run/user/1000/containers/auth.json cannot find file or directory

You might encounter this error if you have executed question 7 before question 2/3 which means if you have logged on to podman with another registry

Two ways to resolve

1. oc podman logout –all and rerun the above create secret command
2. oc create secret docker-registry registry --docker-username=regdevop

-–docker-password=passwordprovidedinexam registry.domainX.example.com] instead of the above secret command

oc secret link **default** registry **--for pull**

**oc podman pull openshift/hello-openshift**

(if you want to check if the image is there)

oc import-image hello-openshift /

--from registry.doaminX.example.com:**443**/openshift/hello-openshift /

--confirm --reference-policy local

oc new-app –-name abc -i hello-openshift

oc expose svc abc –-hostname <provided route>

oc get route

curl http://<route>. => should display "Hello Openshift"

oc create configmap sedicen --from-literal RESPONSE=”Soda pop…”

oc describe configmap/sedicen  
oc set env dc/abc --from configmap/sedicen

curl http://<route>. => should display "Soda pop. …"

1. Set a post-commit build hook  
     
   **Solution Notes:**

Application oxy in project octane is already deployed for you. You are asked to change the build configuration of an application to invoke a build hook. The build hook is a python script mailer.py

You can use the oc set build-hook command to set the build hook.

Then you can trigger a rebuild with oc start-build and follow the build logs with oc logs -f bc/app to see that the mailer.py build hook is run.

There are also instructions to optionally verify that mail was sent via a mail command. Use sudo mail -u capnhook to see mail for user capnhook and verify that mail exists with subject "Mail from never never land …"

**Solution:**

oc project octane

oc set build-hook bc/oxy --post-commit --command -- python mailer.py

Note: there is space between -- and python

oc start-build bc/oxy

oc logs -f bc/appname

oc status

oc get pods (check if pods are running)

sudo mail -u capnhook

#last command should display the 1 email received

**Finding help:**

oc set build-hook --help | grep -A 26 Examples

oc set probe --help | grep -A 26 Examples

1. Set a liveness probe  
     
   **Solution Notes:**

Application oxy in project octane is already deployed for you. It is the same as in previous question. You need to set a liveness probe for tcp port 8080 with an initial delay of 10 seconds and a timeout of 30 seconds.

**Solution:**

--this question is continuation of question 4. We just need to execute one command without changing the project

oc set probe dc/oxy --liveness --open-tcp=8080 --initial-delay-seconds=10 --timeout-seconds=30

oc get pods to verify that pod is running

oc describe to verify

1. Modify a php-mysql template and create an application from the template  
     
   **Solution Notes:**

The Template is provided in json or yaml format and you need to use wget to download from the link provided.

When you edit the template, only the following changes need to be made to the body of the template:

* Project name is indy
* Set the **namespace: indy**
* Set the **name: ex288-php-mysql**
* Change the version of php (**7.0**, 7.1, 7.2, or latest)
* Change the version of mysql (**5.7** or latest?) - note this might already be set to latest so no changes would be required

from:

kind: ImageStreamTag

name: **mysql:5.7**

namespace: openshift

* Add the parameter for HELLO\_AUDIENCE to the list of env variables. You can search for HELLO\_MESSAGE in the template and will find a list of env variables such as DATABASE\_USER, DATABASE\_PASSWORD, etc. and HELLO\_MESSAGE. Then you simply need to copy the two lines for HELLO\_MESSAGE and add them to the end of the list and change them from HELLO\_MESSAGE to HELLO\_AUDIENCE. Here is an example

…

* env

…

- name: HELLO\_MESSAGE

value: ${HELLO\_MESSAGE}

**- name: HELLO\_AUDIENCE**

**value: ${HELLO\_AUDIENCE}**

* You need to make the following changes to the parameter list:
* One of the first parameters NAME(?) is required but doesn't have a default value. I simply **removed the required: true line**.
* **GIT URL** - Replace the value of the git repository with the one that is provided in the question
* Add the HELLO\_AUDIENCE parameter. This is simply a copy of the existing HELLO\_MESSAGE parameter and change the name to HELLO\_AUDIENCE. The value for HELLO\_AUDIENCE is **Engineers**
* Change the HELLO\_MESSAGE value from Ahoy to **Bonjour**
* The parameter for the domain (APPLICATION\_DOMAIN?) is the only required parameter that does not have a default value so that the application does not start without it. – **Remove default value for APPLICATION\_DOMAIN**

Additional Feedback

Both versions of php and mysql in the original template are not available in exam environment and required to change. You should use the following command to check available istag of these image stream in openshift before edit the template:  
...  
oc get istag -n openshift -o name | grep php  
oc get istag -n openshift -o name | grep mysql  
...  
These commands will show what versions of these images are available and you then edit your template accordingly

**Solution:**

oc new-project indy

wget http://..../php-app.yaml

vim php-app.yaml

**Notes on editing php-app.yaml**

1. namespace = **indy**
2. name = **ex288-php-mysql**
3. name: mysql:5.6toname: mysql:**5.7**name: php:6.0 to name:php:**7.0**
4. env  
   …

name: HELLO\_MESSAGE

value: ${HELLO\_MESSAGE}

**name: HELLO\_AUDIENCE**

**value: ${HELLO\_AUDIENCE}**

1. Remove required= true for NAME parameter
2. Update value for parameter “Source Repository URL” = <provided Git URL>
3. Add new parameter  
     
   - name: HELLO\_AUDIENCE  
   Description: HELLO AUDIENCE  
   value: **Engineers**
4. Update HELLO\_MESSAGE parameter value  
     
   - name: HELLO\_MESSAGE  
   Description: HELLO MESSAGE  
   value: **Bonjour**
5. Remove value attribute for APPLICATION\_DOMAIN parameter

oc create -f php-app.yaml

oc new-app --name abc indy/ex288-php-mysql -p APPLICATION\_DOMAIN=<domain URL provided>

open in browser, should display text 🡪 Bonjour Engineers!

oc delete all -l template=**ex288-php-mysql**

oc new-app --name abc indy/ex288-php-mysql /  
--p APPLICATION\_DOMAIN=<domain URL provided> /

--p HELLO\_AUDIENCE = Architects

--p HELLO\_MESSAGE = Hello

open in browser, should display text 🡪 Hello Architects!

1. Log into OpenShift internal registry and pull an image to the workbench  
     
   **Solution Notes:**

You need to login to OpenShift with the ocpdevop user and the provided password. The route for the OpenShift internal registry is not exposed by default and therefore you need to run an oc patch command to expose the route. You can look up the command to use for the patch from the documentation. They give you a link to the documentation in the exam.

Here is how to get to the command in the product documentation:

1. Click the link to the product documentation on the page with "Additional critical information"
2. You will see a list of product documentation that is organized like this: <https://access.redhat.com/documentation/en-us/openshift_container_platform/4.2/>
3. Find the entry for “Registry”
4. There will be html and pdf documentation. I used the html version
5. In the documentation for Registry, in Firefox search for "oc patch" or go directly to the section **2.2 Enable the Image Registry default route with the Custom Resource Definition**
6. Copy and paste the command to patch the Registry Operator CRD:
   * 1. $ oc patch configs.imageregistry.operator.openshift.io/cluster --type merge -p '{"spec":{"defaultRoute":true}}'
7. After executing this command, you can then run oc get route -n openshift-image-registry and should get the route for the OpenShift internal registry.

Next get the token and use sudo podman login -u ocpdevop -p $TOKEN <route>. with the route provided in step 7 to log on to the OpenShift internal registry. You should see a message Login Succeeded!

The question continues to pull an image from a project in OpenShift to the workbench. You would use a command as follows: sudo podman pull <route>/<project>/<image>

You should see how the image is being pulled layer by layer to the workbench.

**Solution:**

oc login -u ocpdevop -p <password> <https://api.master.domainX.example.com>

oc patch 🡪 refer above on “how to get this command from documentation”

oc get route -n openshift-image-registry

TOKEN=$(oc whoami -t)

podman login -u ocpdevop -p $TOKEN <route to openshift internal registry from above command>

sudo podman pull <route>/<project>/<image> **--creds ocpdevop:$TOKEN**

**Final Notes:**

At the end of the exam, you should have modified the following resources on workbench:

* You should have done a git clone for question 1 of the project pastebin and modified package.json and committed and pushed the changes back to git.
* You should have done a git clone for question 2 and modified the assemble script and committed and pushed the changes back to git.
* You should have downloaded the template and modified it to meet the requirements.

There are no other resources required to be modified outside of OpenShift.

**Exam Setup:**

* User for ssh to workbench: devop or root
* 1 user id for OpenShift: ocpdevop
* no admin user
* 1 user id for external registry: regdevop
* external registry as registry.domainx.example.com
* no ssh access other than workbench
* git repository at git.domainx.example.com
* Exam instructions show various IPs and server hostnames for workbench, load balancer, 3 master nodes, 2 worker nodes, etc. but those are not really needed. The URL to use for login to OpenShift is the API URL.
* Link to product documentation is provided

**Key Takeaways:**

* In the 4.1 course material, the major changes are to use quay as the external registry and GitHub as the source code repository. In the exam they are back to registry.lab.example.com for the registry and services.lab.example.com for source code.
* Some questions require setting up a secret for the registry and linking this secret to the builder so that images can be pulled from the external registry.

**MPW Feedback:**

1 For The internal registry, the service is there but it needs to be exposed using oc expose to create the route. Then you can login. if this were OpenShift 3 it would be simple you ensure you are using admin account and login and then expose the service, but it is OpenShift 4 and the internal registry is an operator and has a custom resource definition (CRD) that must be patched.

In order to change the internal image repository, you must do the following

you can Enabling Image Registry Operator by patching CRD

## Enable the Image Registry default route with the Custom Resource Definition

In OpenShift Container Platform, the Registry Operator controls the registry feature. The Operator is defined by the configs.imageregistry.operator.openshift.io Custom Resource Definition (CRD).

If you need to automatically enable the Image Registry default route, patch the Image Registry Operator CRD.

Procedure

* Patch the Image Registry Operator CRD:
* $ oc patch configs.imageregistry.operator.openshift.io/cluster --type merge -p '{"spec":{"defaultRoute":true}}'

[https://docs.openshift.com/container-platform/4.2/registry/configuring-registry-operator.html#registry-operator-default-crd\_configuring-registry-operator](https://docs.openshift.com/container-platform/4.1/registry/configuring-registry-operator.html#registry-operator-default-crd_configuring-registry-operator)

In order to patch the CRD developer account has sufficient privileges

2 for pull secrets and import images sometimes you need to be logged into the registry with sudo and other times you don’t need sudo depending on the type of image you are pulling

 3 finally for linking pull secrets sometimes pull secret need to be linked to builder deployer or default service account depending on image

*Procedure (from OpenShift 4.1 documentation)*

You can expose the route by using **DefaultRoute** parameter in the **configs.imageregistry.operator.openshift.io** resource or by using custom routes.

To expose the registry using **DefaultRoute**:

Set **DefaultRoute** to **True**:

$ oc patch configs.imageregistry.operator.openshift.io/cluster --patch '{"spec":{"defaultRoute":true}}' --type=merge

Log in with Podman:

$ HOST=$(oc get route default-route -n openshift-image-registry --template='{{ .spec.host }}')$

podman login -u $(oc whoami) -p $(oc whoami -t) --tls-verify=false $HOST

**--tls-verify=false** is needed if the cluster’s default certificate for routes is untrusted. You can set a custom, trusted certificate as the default certificate with the [Ingress Operator](https://docs.openshift.com/container-platform/4.1/networking/ingress-operator.html#configuring-ingress).

only gotcha is having to remember to remove the "--tls-verify=false"

Also, a lot of the information regarding secrets that are mentioned in the feedback document are solved by using the flag "--reference-policy local" while doing oc import-image

I didn’t have to link a secret to a builder or deployer

I might have had to link a secret to default once, probably for the template question