

► Guided Exercise

Creating a Multicontainer Template

In this exercise, you will create an OpenShift template that deploys a multicontainer application. The template includes an HTTP API, a database, and persistent storage.

Outcomes

You should be able to:

- Create a template from a running application using the **oc get** command.
- Clean the template to remove runtime information.
- Add parameters to the template.
- Create a new application from the template using the OpenShift command-line tool (**oc**).

Before You Begin

To perform this exercise, ensure you have access to:

- A running OpenShift cluster.
- The PHP 7.4 S2I builder image, and the MySQL 8.0 database image required by the template.
- The sample application (quotes) in the Git repository.

Run the following command on the **workstation** VM to validate the prerequisites, provision persistent storage, deploy the multicontainer application to the **youruser quotes-dev** project, and download the required files to complete this exercise:

```
[student@workstation ~]$ mkdir -p ~/D0288/{labs,solutions}/create-template; \
    cd ~/D0288/labs/create-template;tar xvf T6-EjExtra.tar.xz; \
    cd ~/D0288/solutions/create-template; \
    tar xvf T6EjExtra-SOL.tar.xz
```

To review how the application is deployed, refer to the **new-app-db.sh**, **add-vol.sh**, **new-app-php.sh**, and **add-route.sh** scripts in the **~/D0288/labs/create-template** folder.

► 1. Deploy the database and php apps.

1.1. Log in to OpenShift using your developer user account:

```
[student@workstation ~]$ oc login -u developer -p developer \
> https://api.ocp4.example.com:6443
Login successful.
...output omitted...
```

1.2. Create **quotes-dev** project:

```
[student@workstation ~]$ oc new-project quotes-dev
```

1.3. Deploy the database and application with scripts provided:

```
[student@workstation ~]$ cd ~/DO288/labs/create-template; ./new-app-db.sh
[student@workstation ~]$ ./add-vol.sh; ./new-app-php.sh; ./add-route.sh
```

```
[student@workstation ~]$ oc status
In project youruser-quotes-dev on server
https://api.cluster.domain.example.com:6443

http://quotesapi-quotes-dev.apps.ocp4.example.com to pod port
  8080-tcp (svc/quotesapi)
dc/quotesapi deploys istag/quotesapi:latest <-
  bc/quotesapi source builds https://github.com/redhattraining/DO288-apps on
openshift/php:latest
  deployment #1 deployed 28 seconds ago - 1 pod

svc/quotesdb - 172.30.239.22:3306
dc/quotesdb deploys openshift/mysql:8.0
  deployment #2 deployed about a minute ago - 1 pod
  deployment #1 failed about a minute ago: newer deployment was found running
...output omitted...
```

The previous output shows that the database is deployed from a container image, and the application is deployed from source code.

1.4. Verify that the project includes a persistent volume claim:

```
[student@workstation ~]$ oc get pvc
NAME          STATUS  VOLUME      CAPACITY  ACCESS MODES  ...
quotesdb-claim  Bound   pvc-df1c...  1Gi       RWO        ...
```

1.5. Verify that the project includes a route:

```
[student@workstation ~]$ oc get route/quotesapi -o jsonpath='{.spec.host}{"\n"}'
quotesapi-quotes-dev.apps.ocp4.example.com
```

Do not try to test the application. It will fail because the database is not initialized.
The only use of the **quotes-dev** project is to export its resources to a template.

- 2. Create the template definition file. Start with a basic template definition file and then export the required resources one by one, clean out the runtime information, and then copy the cleaned resource configuration to the template.

Create a new file called **quotes-template-clean.yaml** in the **/home/student** directory. Add the following YAML snippet to declare this resource definition file as an OpenShift template:

```
apiVersion: template.openshift.io/v1
kind: Template
metadata:
  name: quotes
  annotations:
    openshift.io/display-name: Quotes Application
    description: The Quotes application provides an HTTP API that returns a
      random, funny quote.
    iconClass: icon-php
    tags: php,mysql
objects:
```

You can also copy the YAML snippet from the [~/D0288/solutions/create-template/new-template.yaml](#) file.



Warning

Ensure that the **description** attribute value is in a single continuous line with no line breaks. You will get errors for improperly formatted YAML files when the template is parsed later in the exercise.

- 3. Export the resources in the project one by one, starting with the image stream resources.

- 3.1. Export the image stream:

```
[student@workstation ~]$ oc get -o yaml is > /tmp/is.yaml
```

- 3.2. Make a copy of the exported file and clean the runtime attributes from it:

```
[student@workstation ~]$ cp /tmp/is.yaml /tmp/is-clean.yaml
```

- 3.3. Clean the runtime attributes from the [/tmp/is-clean.yaml](#) file.

A copy of the cleaned file is available at [~/D0288/labs/create-template/is-clean.yaml](#). Compare your cleaned file against this version and make the necessary edits to make them the same.

Remove the first two lines from the file:

```
apiVersion: v1
items:
```

- 3.4. There are one image stream resource in the exported file for the quoteapi application quotesapi application.

- 3.5. The following steps involve cleaning the image stream resource for the quotesapi application only.

Remove the **openshift.io/generated-by**, **creationTimestamp**, **generation**, **namespace**, **resourceVersion**, **selfLink**, and **uid** attributes.

Remove the **managedFields**, and **status** attributes including their child attributes.

Tema 6 – Ejercicio Extra

- 4. Export and clean the build configuration.

- 4.1. Export the build configuration:

```
[student@workstation ~]$ oc get -o yaml bc > /tmp/bc.yaml
```

- 4.2. Make a copy of the exported file and clean the runtime attributes from it:

```
[student@workstation ~]$ cp /tmp/bc.yaml /tmp/bc-clean.yaml
```

Clean the runtime attributes from the `/tmp/bc-clean.yaml` file.

A copy of the cleaned file is available at `~/DO288/labs/create-template/bc-clean.yaml`. Compare your cleaned file against this version and make the necessary edits to make them the same.

Ignore differences in the `secret` and `uri` attributes. These values will be different for you.

- 4.3. Remove the first two lines of the file:

```
apiVersion: v1
items:
```

- 4.4. Remove the `openshift.io/generated-by`, `creationTimestamp`, `generation`, `namespace`, `resourceVersion`, `selfLink`, and `uid` attributes.
- 4.5. Remove the `managedFields` attribute and all its child attributes.
- 4.6. Remove the `namespace` attribute that refers to the `youruser-quotes-dev` project. Do not remove the `namespace: openshift` reference under the `sourceStrategy` attribute lower down in the file.
- 4.7. Remove the `lastTriggeredImageID` attribute under `imageChange`.
- 4.8. Remove the `status` attribute and all its child attributes at the bottom of the file.
- 4.9. Remove the `kind: List` attribute at the bottom of the file, and all other attributes under it.

- 5. Export and clean the deployment configuration for the quotesapi application and the quotesdb database.

- 5.1. Export the deployment configuration:

```
[student@workstation ~]$ oc get -o yaml dc > /tmp/dc.yaml
```

- 5.2. Make a copy of the exported file and clean the runtime attributes from it:

```
[student@workstation ~]$ cp /tmp/dc.yaml /tmp/dc-clean.yaml
```

- 5.3. Clean the runtime attributes from the `/tmp/dc-clean.yaml` file.

A copy of the cleaned file is available at `~/DO288/labs/create-template/dc-clean.yaml`. Compare your cleaned file against this version and make the necessary edits to make them the same.

- 5.4. Remove the first two lines of the file:

```
apiVersion: v1
items:
```

- 5.5. Remove all references to **openshift.io/generated-by**, **creationTimestamp**, **generation**, **resourceVersion**, **selfLink**, and **uid** attributes in the file.
- 5.6. Remove the **namespace** attribute that refers to the **youruser-quotes-dev** project. Do not remove the **namespace: openshift** reference under the **imageChangeParams** attribute lower down in the file.
- 5.7. Remove the **managedFields** attribute and all its child attributes.
- 5.8. Remove all references to **image** and **lastTriggeredImage** attributes in the file.
- 5.9. Remove the **status** attribute and all its child attributes for both deployment configuration resources.
- 5.10. Remove the **kind: List** attribute at the bottom of the file, and all other attributes under it.
- 6. Export and clean the service configuration for the quotesapi application and the quotesdb database.

- 6.1. Export the service configuration:

```
[student@workstation ~]$ oc get -o yaml svc > /tmp/svc.yaml
```

- 6.2. Make a copy of the exported file and clean the runtime attributes from it:

```
[student@workstation ~]$ cp /tmp/svc.yaml /tmp/svc-clean.yaml
```

- 6.3. Clean the runtime attributes from the **/tmp/svc-clean.yaml** file.
A copy of the cleaned file is available at **~/DO288/labs/create-template/svc-clean.yaml**. Compare your cleaned file against this version and make the necessary edits to make them the same.

- 6.4. Remove the first two lines of the file:

```
apiVersion: v1
items:
```

- 6.5. Remove all references to **openshift.io/generated-by**, **creationTimestamp**, **namespace**, **resourceVersion**, **selfLink**, and **uid** attributes in the file.
- 6.6. Remove the **managedFields** attribute and all its child attributes.
- 6.7. Remove all references to the **clusterIP** attribute under the **spec** attribute in the file.

Tema 6 – Ejercicio Extra

- 6.8. Remove the **status** attribute and all its child attributes for both service resources.
 - 6.9. Remove the **kind: List** attribute at the bottom of the file, and all other attributes under it.
- ▶ 7. Export and clean the route configuration for the quotesapi application.
- 7.1. Export the route configuration:

```
[student@workstation ~]$ oc get -o yaml route > /tmp/route.yaml
```

- 7.2. Make a copy of the exported file and clean the runtime attributes from it:

```
[student@workstation ~]$ cp /tmp/route.yaml /tmp/route-clean.yaml
```

- 7.3. Clean the runtime attributes from the **/tmp/route-clean.yaml** file.
A copy of the cleaned file is available at **~/DO288/labs/create-template/route-clean.yaml**. Compare your cleaned file against this version and make the necessary edits to make them the same.
- 7.4. Remove the first two lines of the file:

```
apiVersion: v1
items:
```

- 7.5. Remove all references to **openshift.io/generated-by**, **creationTimestamp**, **namespace**, **resourceVersion**, **selfLink**, and **uid** attributes in the file.
- 7.6. Remove the **managedFields** attribute and all its child attributes.
- 7.7. Remove the **host** and **subdomain** attributes under the **spec** attribute.
- 7.8. Remove the **status** attribute and all its child attributes.
- 7.9. Remove the **kind: List** attribute at the bottom of the file, and all other attributes under it.

- ▶ 8. Export and clean the persistent volume configuration (PVC) for the quotesdb database.

- 8.1. Export the persistent volume configuration:

```
[student@workstation ~]$ oc get -o yaml pvc > /tmp/pvc.yaml
```

- 8.2. Make a copy of the exported file and clean the runtime attributes from it:

```
[student@workstation ~]$ cp /tmp/pvc.yaml /tmp/pvc-clean.yaml
```

- 8.3. Clean the runtime attributes from the **/tmp/pvc-clean.yaml** file.
A copy of the cleaned file is available at **~/DO288/labs/create-template/pvc-clean.yaml**. Compare your cleaned file against this version and make the necessary edits to make them the same.

- 8.4. Remove the first two lines of the file:

```
apiVersion: v1
items:
```

- 8.5. Remove all attributes under the **metadata.annotations** attribute.
- 8.6. Remove all references to **creationTimestamp**, the **finalizers** attribute and its children, as well as the **namespace**, **resourceVersion**, **selfLink**, and **uid** attributes in the file.
- 8.7. Remove the **managedFields** attribute and all its child attributes.
- 8.8. Remove the **dataSource** attribute under the **spec** attribute.
- 8.9. Remove the **storageClassName**, **volumeMode**, and **volumeName** attributes under the **spec** attribute.
- 8.10. Remove the **status** attribute and all its child attributes.
- 8.11. Remove the **kind: List** attribute at the bottom of the file, and all other attributes under it.
- ▶ 9. Copy the individual resource configuration YAML snippets from the cleaned files into the **/home/student/quotes-template-clean.yaml** file under the **objects** attribute in the following order:

```
[student@workstation ~]$ cat /tmp/is-clean.yaml >> ~/quotes-template-clean.yaml
[student@workstation ~]$ cat /tmp/bc-clean.yaml >> ~/quotes-template-clean.yaml
[student@workstation ~]$ cat /tmp/dc-clean.yaml >> ~/quotes-template-clean.yaml
[student@workstation ~]$ cat /tmp/svc-clean.yaml >> ~/quotes-template-clean.yaml
[student@workstation ~]$ cat /tmp/route-clean.yaml >> ~/quotes-template-clean.yaml
[student@workstation ~]$ cat /tmp/pvc-clean.yaml >> ~/quotes-template-clean.yaml
```

A copy of the final cleaned template is available at **~/DO288/solutions/create-template/quotes-template-clean.yaml**. Compare your final cleaned template against this file.

Ignore differences in the **secret** and **uri** attributes. These values will be different for you. You will parameterize these values in the next step.

- ▶ 10. Add parameters to make the template reusable.

To keep this guided exercise short, you will add only three parameters: one for the Git URL where the application source code is stored, and two for passwords and secrets.

- 10.1. Inspect the file that contains the parameter definition. Notice that only the **APP_GIT_URL** parameter is required. The **PASSWORD** and **SECRET** parameters have random default values:

Tema 6 – Ejercicio Extra

```
[student@workstation ~]$ cat ~/DO288/labs/create-template/parameters.yaml
parameters:
- name: APP_GIT_URL
  displayName: Application Source Git URL
  description: The Git URL of the application source code
  required: true
- name: PASSWORD
  displayName: Database Password
  description: Password to access the database
  generate: expression
  from: '[a-zA-Z0-9]{16}'
- name: SECRET
  displayName: Webhook Secret
  description: Secret for webhooks
  generate: expression
  from: '[a-zA-Z0-9]{40}'
```

10.2. Make a copy of the final cleaned template YAML file before adding the parameters:

```
[student@workstation ~]$ cp ~/quotes-template-clean.yaml ~/quotes-template.yaml
```

10.3. Open the **quotes-template.yaml** file in a text editor. Copy the contents of the **parameters.yaml** file to the end of the **quotes-template.yaml** file.

10.4. Change the **secret**, **password**, and **uri** attributes to reference template parameters.

Use the following listing as a guide to make the changes:

```
...output omitted...
  kind: BuildConfig
  ...output omitted...
  name: quotesapi
  ...output omitted...
  source:
    contextDir: quotes
    git:
      uri: ${APP_GIT_URL}
    type: Git
    ...output omitted...
  triggers:
    - github:
        secret: ${SECRET}
    type: GitHub
    - generic:
        secret: ${SECRET}
  ...output omitted...
```

Tema 6 – Ejercicio Extra

```
kind: DeploymentConfig
...output omitted...
name: quotesapi
...output omitted...
- name: DATABASE_PASSWORD
  value: ${PASSWORD}
...output omitted...
kind: DeploymentConfig
...output omitted...
name: quotesdb
...output omitted...
- name: MYSQL_PASSWORD
  value: ${PASSWORD}
...output omitted...
```

- 10.5. To make the MySQL database container persistent, you need to add a **volumeMounts** attribute to the database deployment configuration, and reference the persistent volume claim resource declared in the template.

Add the following lines to the quotesdb deployment configuration as follows:

```
...output omitted...
terminationMessagePath: /dev/termination-log
terminationMessagePolicy: File
volumeMounts:
- mountPath: /var/lib/mysql/data
  name: quotesdb-volume-1
dnsPolicy: ClusterFirst
restartPolicy: Always
...output omitted...
```

- 10.6. The template file is now complete. Save your edits.

To verify the changes you made during this step, review the **quotes-template.yaml** file in the **~/DO288/solutions/create-template** folder. If you are uncertain about your edits, you can copy the solution file and continue to the next step.

- 11. Create a new application from the template.

- 11.1. Create the **youruser-myquotes** project:

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

- 12. Create a new instance of the quotes application from the template.

- 12.1. Use the **oc new-app** command to create a new instance of the quotes application from the template:

```
[student@workstation ~]$ oc new-app --file=quotes-template.yaml \
> -p APP_GIT_URL=https://github.com/redhattraining/DO288-apps \
> -p PASSWORD=mypass
```

Tema 6 – Ejercicio Extra

```
--> Deploying template "myquotes/quotes" to project myquotes

Quotes Application
-----
The Quotes application provides an HTTP API that returns a random, funny quote.

* With parameters:
* Application Source Git URL=https://github.com/redhattraining/D0288-apps
* Database Password=mypass
* Webhook Secret=D6xlih2F3KIyYwsIXs3nLysGHJbi6JLhtaul6I10 # generated

--> Creating resources ...
imagestream.image.openshift.io "quotesapi" created
buildconfig.build.openshift.io "quotesapi" created
deploymentconfig.apps.openshift.io "quotesapi" created
deploymentconfig.apps.openshift.io "quotesdb" created
service "quotesapi" created
service "quotesdb" created
route.route.openshift.io "quotesapi" created
persistentvolumeclaim "quotesdb-claim" created
--> Success
...output omitted...
```



Note

If you see an error at this step, ensure that there are no extra line breaks, and that your template YAML file is properly indented. OpenShift displays the line number in the template where processing failed. Use this information to identify and fix the issues.

- 12.2. Wait for the build to complete and for the quotesdb and quotesapi applications to have one pod each ready and running.

```
[student@workstation ~]$ oc get pods
NAME          READY   STATUS    RESTARTS   AGE
...output omitted...
quotesapi-1-tphzw   1/1     Running      0          3m12s
quotesdb-1-rk8mw   1/1     Running      0          4m28s
```

- 13. Populate the database and test the application.

- 13.1. Inspect the script that populates the database.

The **populate-db.sh** script in the **~/D0288/labs/create-template** directory uses a port-forwarding tunnel to connect to the database pod and a local MySQL client to run an SQL script:

```
[student@workstation ~]$ cat ~/D0288/labs/create-template/populate-db.sh
...output omitted...
echo 'Creating tunnel...'
pod=$(oc get pod -l deploymentconfig=quotesdb -o name)
oc port-forward $(basename ${pod}) 30306:3306 &
tunnel=$!
sleep 3
```

Tema 6 – Ejercicio Extra

```
echo 'Initializing the database...'
mysql -h127.0.0.1 -P30306 -uquoteapp -pmypass quotesdb \
< ~/DO288/labs/create-template/quote.sql
sleep 3
echo 'Terminating tunnel...'
kill ${tunnel}
```

13.2. Run the **populate-db.sh** script:

```
[student@workstation ~]$ ~/DO288/labs/create-template/populate-db.sh
Creating tunnel...
Forwarding from 127.0.0.1:30306 -> 3306
Forwarding from [::1]:30306 -> 3306
Initializing the database...
Handling connection for 30306
Terminating tunnel...
```

13.3. Get the route host name for the application:

```
[student@workstation ~]$ oc get route/quotesapi -o jsonpath='{.spec.host}{"\n"}'
quotesapi-youruser-quotes-dev.apps.cluster.domain.example.com
```

13.4. Use the **curl** command and the host name from the previous step to access the application. Your output might be different from this example:

```
[student@workstation ~]$ curl \
> quotesapi-myquotes.apps.ocp4.example.com/get.php
Veni, vidi, vici...
```

- 14. Clean up. Delete the projects from OpenShift and release the persistent storage allocated for this exercise.

Open a terminal window on the **workstation** VM and run the following command to perform the cleanup tasks:

```
[student@workstation ~]$ oc delete project myquotes
```

```
[student@workstation ~]$ oc delete project quotes-dev
```

This concludes the guided exercise.