

## ▶ Lab

# Creating Applications from OpenShift Templates

### Performance Checklist

In this lab, you will deploy the To Do List application from a template that you will complete for reuse.

### Outcomes

You should be able to complete a template that deploys a database pod and a web application pod, use the template to deploy the application, and verify that the application works.

### Before You Begin

To perform this exercise, ensure you have access to:

- A running OpenShift cluster.
- The sample application (todo-single) in the Git repository.
- The npm dependencies required by the application (**restify**, **sequelize**, and **mysql**).
- The Node.js 16 S2I builder image, and the MySQL 8.0 database image required by the template.

Run the following command on the **workstation** VM to validate the prerequisites and to download the starter project files and solution files:

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

You can obtain these files using commands:

```
wget https://github.com/MMGInstructor/FichsEjercicios/raw/refs/heads/main/T6-LabExtra.tar
wget https://github.com/MMGInstructor/FichsEjercicios/raw/refs/heads/main/T6-LabExtra-SOL.tar
```

### Requirements

The application is a To Do List application written in JavaScript. It consists of a web front end, based on the AngularJS web framework, and an HTTP API back end, based on Node.js. The back end uses the Restify and Sequelize frameworks.

Both the application front end and back end run on the same container. The application is deployed from source code stored in a Git repository. A MySQL database is used as the data store. The application initializes the database on startup.

The template takes the following parameters:

- **APP\_GIT\_URL**: The Git URL where the source code for the To Do List application is stored.
- **HOSTNAME**: The host name used to access the To Do List application.
- **NPM\_PROXY**: The URL of the NPM repository server.
- **SECRET**: The secret for OpenShift webhooks.

## Tema 6 - Lab Extra

- **PASSWORD:** The database connection password. The user name is fixed in the template.
- **CLEAN\_DATABASE:** A flag that indicates whether the application initializes the database on startup.

The template parameters have the following restrictions:

- The **APP\_GIT\_URL**, **HOSTNAME**, **NPM\_PROXY**, and **CLEAN\_DATABASE** parameters are required.
- The **SECRET** and **PASSWORD** parameters have random default values.
- The **CLEAN\_DATABASE** parameter has the default value of **true**.

A starter template file called **todo-template.yaml** is provided in the **~/D0288/labs/review-template** folder. It contains the resources required to deploy the application. The resources are in the correct order, and already cleaned up. You are required to add the missing parameters and add references to the parameters in the resource list.

Deploy the application according to the following requirements:

- The application project is called **review-template**.
- Use the **~/D0288/labs/review-template/oc-new-app.sh** script to deploy the application. Edit the script as needed to pass any required parameters.
- The application initializes the database on startup.
- The password to access the database is **mypass**.
- Npm modules required to build the application are available from:

**http://nexus-infra.apps.ocp4.example.com/repository/nodejs**

- If you prefer to test the application with a web browser, use the web front end to add some entries to the To Do List application interactively. The application user interface is accessible from the following URL:

**http://todo.apps.ocp4.example.com/todo/index.html**

- If you prefer to test the application using the command line, use the following URL to submit an HTTP GET request to the application back end:

**http://todo.apps.ocp4.example.com/todo/api/items-count**

The reply includes a **count** attribute, even if there are no entries in the application. An error reply means the database was not initialized correctly.

## Steps

1. Review the **todo-template.yaml** starter template file in the **~/D0288/labs/review-template** folder. Make a copy of this file in the **/home/student/** directory, and use this copy to make your changes throughout the lab. Identify and add any missing parameters at the end of the file. Identify any parameter references missing from the template resource list and add them as needed. Use the existing parameter definitions and parameter references as a guide to add the missing ones.

## Tema 6 - Lab Extra

You are not required to, but you can inspect the **todo-template-clean.yaml** file in the same folder. This file contains the application resources exported by the **oc get** command, with the runtime attributes already removed. Do not make any changes to this file.

You are not required to, but you can also inspect the **new-app-db.sh**, **new-app-node.sh**, and **add-route.sh** files in the same folder that contain the commands used to generate the resources that were exported to the **todo-template-clean.yaml** file. Do not make any changes to these files and do not run any of them.

2. Create a new project and deploy the To Do List application using the template definition file you completed during the previous step.

Review the **oc-new-app.sh** starter script file in the **~/DO288/labs/review-template** folder. Make a copy of this file in the **/home/student/** directory and use this copy to make your changes throughout the lab. Identify and add any missing parameters to the **oc new-app** command line. Run the finalized **oc-new-app.sh** script to deploy the application.

Do not perform any manual steps to initialize the database. The To Do List application creates the required database tables if you pass the correct set of parameters to the template. The database tables do not need an initial data set.

3. Test the To Do List application using either a web browser or the command line.
4. Clean up. Delete the project from OpenShift.

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

```
[student@workstation ~]$ oc delete project review-template
```

This concludes the lab.

# Creating Applications from OpenShift Templates

## Performance Checklist

In this lab, you will deploy the To Do List application from a template that you will complete for reuse.

## Outcomes

You should be able to complete a template that deploys a database pod and a web application pod, use the template to deploy the application, and verify that the application works.

## Before You Begin

To perform this exercise, ensure you have access to:

- A running OpenShift cluster.
- The sample application (`todo-single`) in the Git repository.
- The npm dependencies required by the application (`restify`, `sequelize`, and `mysql`).
- The Node.js 16 S2I builder image, and the MySQL 8.0 database image required by the template.

Run the following command on the **workstation** VM to validate the prerequisites and to download the starter project files and solution files:

```
[student@workstation ~]$ mkdir -p ~/DO288/{labs,solutions}/review-template; \
    cd ~/DO288/labs/review-template; tar xvf T6-LabExtra.tar.xz; \
    cd ~/DO288/solutions/review-template; \
    tar xvf T6-LabExtra-SOL.tar.xz
```

## Requirements

The application is a To Do List application written in JavaScript. It consists of a web front end, based on the AngularJS web framework, and an HTTP API back end, based on Node.js. The back end uses the Restify and Sequelize frameworks.

Both the application front end and back end run on the same container. The application is deployed from source code stored in a Git repository. A MySQL database is used as the data store. The application initializes the database on startup.

The template takes the following parameters:

- **APP\_GIT\_URL**: The Git URL where the source code for the To Do List application is stored.
- **HOSTNAME**: The host name used to access the To Do List application.
- **NPM\_PROXY**: The URL of the NPM repository server.
- **SECRET**: The secret for OpenShift webhooks.

## Tema 6 - Lab Extra

- **PASSWORD:** The database connection password. The user name is fixed in the template.
- **CLEAN\_DATABASE:** A flag that indicates whether the application initializes the database on startup.

The template parameters have the following restrictions:

- The **APP\_GIT\_URL**, **HOSTNAME**, **NPM\_PROXY**, and **CLEAN\_DATABASE** parameters are required.
- The **SECRET** and **PASSWORD** parameters have random default values.
- The **CLEAN\_DATABASE** parameter has the default value of **true**.

A starter template file called **todo-template.yaml** is provided in the **~/DO288/labs/review-template** folder. It contains the resources required to deploy the application. The resources are in the correct order, and already cleaned up. You are required to add the missing parameters and add references to the parameters in the resource list.

Deploy the application according to the following requirements:

- The application project is called **review-template**
- Use the **~/DO288/labs/review-template/oc-new-app.sh** script to deploy the application. Edit the script as needed to pass any required parameters.
- The application initializes the database on startup.
- The password to access the database is **mypass**.
- Npm modules required to build the application are available from:  
**http://nexus-infra.apps.ocp4.example.com/repository/nodejs**
- If you prefer to test the application with a web browser, use the web front end to add some entries to the To Do List application interactively. The application user interface is accessible from the following URL:  
**http://todo.apps.ocpc4.example.com/todo/index.html**
- If you prefer to test the application using the command line, use the following URL to submit an HTTP GET request to the application back end:  
**http://todo.apps.ocp4.example.com/todo/api/items-count**

The reply includes a **count** attribute, even if there are no entries in the application. An error reply means the database was not initialized correctly.

## Steps

1. Review the **todo-template.yaml** starter template file in the **~/DO288/labs/review-template** folder. Make a copy of this file in the **/home/student/** directory, and use this copy to make your changes throughout the lab. Identify and add any missing parameters at the end of the file. Identify any parameter references missing from the template resource list and add them as needed. Use the existing parameter definitions and parameter references as a guide to add the missing ones.

You are not required to, but you can inspect the **todo-template-clean.yaml** file in the same folder. This file contains the application resources exported by the **oc get** command, with the runtime attributes already removed. Do not make any changes to this file.

## Tema 6 - Lab Extra

You are not required to, but you can also inspect the **new-app-db.sh**, **new-app-node.sh**, and **add-route.sh** files in the same folder that contain the commands used to generate the resources that were exported to the **todo-template-clean.yaml** file. Do not make any changes to these files and do not run any of them.

- 1.1. Create a copy of the starter template to make edits:

```
[student@workstation ~]$ cp ~/D0288/labs/review-template/todo-template.yaml \
> ~/todo-template.yaml
```

- 1.2. Identify any missing parameters in the template file.

Open the **~/todo-template.yaml** file with a text editor.

Scroll down to the end of the file. The file contains four parameters called **APP\_GIT\_URL**, **HOSTNAME**, **NPM\_PROXY**, and **SECRET**. These parameter definitions are complete and do not need any changes.

The **PASSWORD** and **CLEAN\_DATABASE** parameters are missing and need to be added.

- 1.3. Add the missing parameters to the **~/todo-template.yaml** template file.

Append the following lines to the file. You can copy these lines from the **add-parameters.yaml** file in the **~/D0288/solutions/review-template** folder:

```
- name: PASSWORD
  displayName: Database Password
  description: Password to access the database
  generate: expression
  from: '[a-zA-Z0-9]{16}'
- name: CLEAN_DATABASE
  displayName: Initialize the database
  description: If 'true', the database is cleaned when the application starts.
  required: true
  value: "true"
```

- 1.4. Identify the missing references to parameter values in the template resource list.

Some of the resources already reference the parameters correctly. The following listing highlights the parameter references that are already in the template file. You do not need to change any of them:

```
...output omitted...
kind: BuildConfig
...output omitted...
  name: todoapp
  ...output omitted...
  triggers:
  - github:
      secret: ${SECRET}
    type: GitHub
  - generic:
      secret: ${SECRET}
  ...output omitted...
kind: DeploymentConfig
  ...output omitted...
  name: todoapp
```

## Tema 6 - Lab Extra

```
...output omitted...
- env:
  - name: DATABASE_NAME
    value: tododb
  - name: DATABASE_PASSWORD
    value: ${PASSWORD}
  - name: DATABASE_SVC
    value: tododb
  - name: DATABASE_USER
    value: todoapp
  - name: DATABASE_INIT
    value: ${CLEAN_DATABASE}
...output omitted...
```

A few resources are missing references to parameters. The following listing highlights the attributes that are set to hard-coded values and need to be replaced by parameter references:

```
...output omitted...
kind: BuildConfig
...output omitted...
name: todoapp
...output omitted...
strategy:
  sourceStrategy:
    env:
      - name: npm_config_registry
        value: http://INVALIDHOST.NODOMAIN.NUL
...output omitted...
kind: DeploymentConfig
...output omitted...
name: tododb
...output omitted...
  - env:
    - name: MYSQL_DATABASE
      value: tododb
    - name: MYSQL_PASSWORD
      value: FIXEDPASSWD
...output omitted...
kind: Route
...output omitted...
name: todoapp
...output omitted...
spec:
  host: http://INVALIDHOST.NODOMAIN.NUL
...output omitted...
```

15. Add the missing references to parameters to the `~/todo-template.yaml` file. The following listing shows the changes you need to make:

```
...output omitted...
kind: BuildConfig
...output omitted...
name: todoapp
```

## Tema 6 - Lab Extra

```
...output omitted...
strategy:
  sourceStrategy:
    env:
      - name: npm_config_registry
        value: ${NPM_PROXY}
...output omitted...
kind: DeploymentConfig
...output omitted...
  name: tododb
  ...output omitted...
    - env:
      - name: MYSQL_DATABASE
        value: tododb
      - name: MYSQL_PASSWORD
        value: ${PASSWORD}
...output omitted...
  kind: Route
  ...output omitted...
  name: todoapp
  ...output omitted...
spec:
  host: ${HOSTNAME}
...output omitted...
```

- 1.6. Review your edits and save the changes to the `~/todo-template.yaml` template file. You can compare your edits with the solution file in the `~/D0288/solutions/review-template` folder.
2. Create a new project and deploy the To Do List application using the template definition file you completed during the previous step.

Review the `oc-new-app.sh` starter script file in the `~/D0288/labs/review-template` folder. Make a copy of this file in the `/home/student/` directory and use this copy to make your changes throughout the lab. Identify and add any missing parameters to the `oc new-app` command line. Run the finalized `oc-new-app.sh` script to deploy the application.

Do not perform any manual steps to initialize the database. The To Do List application creates the required database tables if you pass the correct set of parameters to the template. The database tables do not need an initial data set.

- 2.1. Create a copy of the starter script to make edits:

```
[student@workstation ~]$ cp ~/D0288/labs/review-template/oc-new-app.sh \
> ~/oc-new-app.sh
```

- 2.2. Identify any missing parameters in the `~/oc-new-app.sh` script file.

Open the `~/oc-new-app.sh` file with a text editor.

The script passes values for the `APP_GIT_URL`, `NPM_PROXY`, `PASSWORD`, and `CLEAN_DATABASE` parameters.

The values passed for `APP_GIT_URL`, `NPM_PROXY`, and `PASSWORD` are correct; you do not need to change them. You need to change the value for the `CLEAN_DATABASE` parameter to `true`.

You also need to add a command-line option to pass a value for the `HOSTNAME` parameter.

2.3. Make changes to the `~/oc-new-app.sh` script.

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

```
oc new-app --name todo --file ~/todo-template.yaml \
> -p APP_GIT_URL=https://github.com/redhattraining/D0288-apps \
> -p NPM_PROXY=http://nexus-infra.apps.ocp4.example.com/repository/nodejs \
> -p PASSWORD=mypass \
> -p CLEAN_DATABASE=true \
> -p HOSTNAME=todo.apps.ocp4.example.com
```

Do not add line breaks to any parameter value. The value for **NPM\_PROXY** needs to be on a single line. Recall that you do not need to change it from the starter file.

2.4. Review your edits and save the changes to the `~/oc-new-app.sh` file. You can compare your edits with the solution file in the `~/D0288/solutions/review-template` folder.

3. Test the To Do List application using either a web browser or the command line.

3.1. 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...
```

3.2. Create the **review-template** project:

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

3.3. Run the `oc-new-app.sh` script to create all of the resources from the `todo-template.yaml` template file:

```
[student@workstation ~]$ ~/oc-new-app.sh
...output omitted...
--> Creating resources...
imagestream "todoapp" created
buildconfig "todoapp" created
deploymentconfig "todoapp" created
deploymentconfig "tododb" created
service "todoapp" created
service "tododb" created
route "todoapp" created
--> Success
...output omitted...
```

## Tema 6 - Lab Extra

- 3.4. Wait for the To Do List application build to finish:

```
[student@workstation ~]$ oc logs bc/todoapp -f  
...output omitted...  
Push successful
```

- 3.5. Wait for the application and the database pod to be ready and running:

```
[student@workstation ~]$ oc get pods  
NAME READY STATUS RESTARTS AGE  
...output omitted...  
todoapp-1-nch6c 1/1 Running 0 1m  
tododb-1-lpk63 1/1 Running 0 2m
```

- 3.6. Get the route host name for the application:

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

- 3.7. Test the application using the route host name obtained from the previous step.

If you prefer to test the To Do List application using a web browser, access the following URL and add some entries to the list:

<http://todo.apps.ocp4.example.com/todo/index.html>

- 3.8. If you prefer to test the To Do List application using the command line, open a terminal window and run the following command:

```
[student@workstation ~]$ curl -siw "\n" http://todo.apps.ocp4.example.com \  
> /todo/api/items-count  
HTTP/1.1 200 OK  
...output omitted...  
{"count":0}
```

4. Clean up. Delete the project from OpenShift.

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

```
[student@workstation ~]$ oc delete project review-template
```