

Search

# Introduction to OpenShift Applications

## Tell us what you think!

Congratulations on completing at least 25% of the course. We would appreciate feedback on your learning experience. Click below to fill out a brief survey.

[TAKE ME TO THE SURVEY \(HTTPS://SURVEY.OLE.REDHAT.COM/SURVEY?COURSE=DO101&MODALITY=COURSE&OFFERING=101\)](https://survey.ole.redhat.com/survey?course=DO101&modality=course&offering=101)

NOT RIGHT NOW

☐ Don't ask again

TRANSLATIONS ▾

CERTIFICATE OF ATTENDANCE



P (/rol/app/courses/do101-4.2/pages/pr01) (/rol/app/courses/do101-4.2/pages/pr01s02) 1 (/rol/app/courses/do101-4.2/pages/ch01) (/rol/app/courses/do101-4.2/pages/ch01s02) (/rol/app/courses/do101-4.2/pages/ch01s03) (/rol/app/courses/do101-4.2/pages/ch01s04) 4.2/pages/pr01) (/rol/app/courses/do101-4.2/pages/ch01s05) (/rol/app/courses/do101-4.2/pages/ch01s06) (/rol/app/courses/do101-4.2/pages/ch01s07) 2 (/rol/app/courses/do101-4.2/pages/ch02) (/rol/app/courses/do101-4.2/pages/ch02s02) (/rol/app/courses/do101-4.2/pages/ch02s03) (/rol/app/courses/do101-4.2/pages/ch03) (/rol/app/courses/do101-4.2/pages/ch03s02) (/rol/app/courses/do101-4.2/pages/ch03s03) (/rol/app/courses/do101-4.2/pages/ch03s04) (/rol/app/courses/do101-4.2/pages/ch03s05) (/rol/app/courses/do101-4.2/pages/ch03s06) (/rol/app/courses/do101-4.2/pages/ch03s07) 4 (/rol/app/courses/do101-4.2/pages/ch04) (/rol/app/courses/do101-4.2/pages/ch04s02) (/rol/app/courses/do101-4.2/pages/ch04s03) (/rol/app/courses/do101-4.2/pages/ch05) (/rol/app/courses/do101-4.2/pages/ch05s02) (/rol/app/courses/do101-4.2/pages/ch05s03) (/rol/app/courses/do101-4.2/pages/ch05)

← PREVIOUS (/ROL/APP/COURSES/DO101-4.2/PAGES/CH03)

→ NEXT (/ROL/APP/COURSES/DO101-4.2/PAGES/CH03S03)

## Guided Exercise: Updating an Application



In this exercise, you will update the source code for a Node.js application deployed on OpenShift.

### Outcomes

You should be able to:

- Create a new OpenShift application using the `oc new-app` command.
- Trigger a new build manually from the OpenShift web console, after updating the source code of an application.
- Set up webhooks in GitHub to automatically trigger new builds when there are new commits to the Git repository.

To perform this exercise, ensure that you have access to:

- A running Red Hat OpenShift Container Platform cluster.
- The source code for the `version` application in the `D0101-apps` Git repository on your local system.

### Procedure 3.1. Steps

1. Install the OpenShift command line interface (CLI).
  - 1.1. Download the OpenShift CLI binary for your platform from <https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest/> (<https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest/>).
  - 1.2. Download the compressed files of the form `openshift-client-  
<platform>-<version>.*`.

**WARNING**

Do not download the openshift-install-\* files.

- 1.3. Unzip the compressed archive file, and then copy the `oc` binary to a directory of your choice. Ensure that this directory is in the `PATH` variable for your system. On macOS and Linux, copy the `oc` binary to `/usr/local/bin`:

```
$ sudo cp oc /usr/local/bin/  
$ sudo chmod +x /usr/local/bin/oc
```

For Windows 10 systems, decompress the downloaded archive, and then follow the instructions at <https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/> (<https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/>) to edit the `PATH` environment variable and add the full path to the directory where the `oc` binary is located.

- 1.4. Verify that the `oc` binary works for your platform. Open a new command line terminal and run the following:

```
$ oc version  
Client Version: openshift-clients-4.2.0-201910041700
```

Your output may be slightly different based on the version of the OpenShift client that you downloaded.

**NOTE**

To use the `oc` command from within the VS Code embedded terminal, restart the IDE after installing it.

2. Deploy the version application to OpenShift using the `oc` tool.

- 2.1. Log in to your OpenShift account using the API end point URL that was provided to you:

You will be prompted for a user name and password. Use the same user name and password that you used to log in to the OpenShift web console.

**WARNING**

The API end point URL and the OpenShift web console URL are **not** the same. The API end point URL is used mainly to interact with OpenShift, using command line tools like `oc`.

**SHOW SOLUTION**

- 2.2. Create a new project called `youruser-version`.

**SHOW SOLUTION**

- 2.3. Launch the Visual Studio Code (VS Code) editor. In the Explorer view (**View** → **Explorer**), open the `D0101-apps` folder in the `My Projects` workspace. The source code for the version application is in the `version` directory.
- 2.4. In the source control view in VS Code (**View** → **SCM**), ensure that the `D0101-apps` entry under `SOURCE CONTROL PROVIDERS` shows the `master` branch. If you were working with another branch for a different exercise, click on the current branch and then select `master` in the `Select a ref to checkout` window to switch to the `master` branch.

**WARNING**

Each exercise uses a unique branch. Always create a new branch using `master` as the base.

- 2.5. Click **master** in the `D0101-apps` entry under `SOURCE CONTROL PROVIDERS`.

Select `Create new branch...` from the list of options.

- 2.6. At the prompt, enter `update-app` for the branch name. The Source Control view updates the `D0101-apps` entry with

the new branch name.

- 2.7. Push the `update-app` branch to your `DO101-apps` GitHub repository.

Click the Publish Changes cloud icon next to the `update-app` branch to push your local branch to your remote Git repository. If prompted, then provide your GitHub user name and password.

- 2.8. Deploy the `version` application from your `DO101-apps` Git repository.

Switch to a command line terminal. Enter the command below in a single line with no line breaks. Do not type the `\` character at the end of the lines. The command is broken into multiple lines in this example for clarity and formatting purposes only:

#### NOTE

The `#update-app` indicates that OpenShift should use the `update-app` Git branch, which was created in the previous step. The `--context-dir` parameter indicates the directory under the `DO101-apps` repository where the application source code is stored.

SHOW SOLUTION

3. Test the application.

- 3.1. Open a web browser and access the OpenShift web console. Log in to the OpenShift web console using your developer account.
- 3.2. Switch to the Developer perspective.
- 3.3. Click Topology in the navigation pane. From the Project: list, select the `yourname-version` project.
- 3.4. Click the `version` deployment config, and then click the Overview tab. Verify that a single pod is running. You may have to wait for a minute or two for the application to be fully deployed.

Figure 3.3: Application deployment complete

- 3.5. Applications created using the `oc new-app` command do not create a route resource by default. Run the following command to create a route and allow access to the application from the internet:

```
$ oc expose svc/version
route.route.openshift.io/version exposed
```

#### NOTE

You can also create a route using the OpenShift web console; select the Administrator perspective, and then click on **Networking** → **Routes**.

- 3.6. View the Topology page again, and observe that the `version` deployment now displays an icon to open a URL. Click Open URL to view the application.

Figure 3.4: Route URL

- 3.7. The application opens in a new browser tab. You should see version 1 of the application as output.

Figure 3.5: Application version 1

Close the browser tab.

4. Update the `version` application.

- 4.1. In the VS Code explorer view (**View** → **Explorer**), click the `app.js` file in the `version` application folder of the `D0101-apps` folder. VS Code opens an editor tab for the `app.js` file.

- 4.2. Change the response variable to `version 2` as follows:

SHOW SOLUTION

- 4.3. Save the changes to the file.

- 4.4. Commit your changes locally, and then push the new commit to your GitHub repository.

Access the Source Control view (**View** → **SCM**). Locate the entry for the `app.js` file under the **CHANGES** heading for the `D0101-apps` directory.

Click the plus (+) button to add the `app.js` file changes to your next commit.

- 4.5. Add a commit message of `updated app to v2` in the message prompt, and then click the check mark button to commit the staged changes.

- 4.6. Locate the `D0101-apps` entry in the **SOURCE CONTROL PROVIDERS** heading.

Click the **Synchronize Changes** icon to publish the changes to the remote repository.

If prompted that this action will push and pull commits to and from the origin, then click **OK** to continue. If prompted, then provide your GitHub user name and password.

5. Trigger a new build manually using the OpenShift web console.

- 5.1. Click on **Builds** in the navigation pane to view the **Build Configs** page.

Figure 3.6: Build Configs Page

- 5.2. Expand the menu to the right of the `version` build config, and then click **Start Build**.

Figure 3.7: Start Build

- 5.3. You should now see the details page for the build `version-2`.

Click on the **Logs** tab to see the progress of the build. A new container image with the updated source code is built and pushed to the OpenShift image registry. Wait until the application container image is built and deployed. You should see a **Push successful** message in the logs before proceeding to the next step.

- 5.4. Click on **Topology** in the navigation pane, and then click **Open URL** from the `version` deployment. Version 2 of the application is displayed.

Figure 3.8: Version 2 of the application

Close the browser tab.

6. Set up webhooks in GitHub to automatically trigger a new build when you commit changes to the repository.

- 6.1. Navigate to your `D0101-apps` repository (<https://github.com/yourgituser/DO101-apps>) on GitHub using a web browser. Log in to your GitHub account if prompted.

- 6.2. Click on the **Settings** tab to open the settings page for the repository.

Figure 3.9: GitHub repository settings

- 6.3. Click on **Webhooks** in the left menu to open the webhooks page.

Figure 3.10: Webhooks page

- 6.4. Click **Add webhook** to add a new webhook. You may be asked to reenter your GitHub password. After your password is confirmed, the `Add Webhook` page displays.
- 6.5. Next, you must provide a payload URL to the webhook. The webhook is triggered (sends a HTTP POST request to the URL) when you commit some changes to the Git repository.  
  
To get the payload URL, switch to the OpenShift web console in your browser, and then click on `Builds` in the navigation pane.
- 6.6. Click the version build config to bring up the `Build Config Overview` page.  
  
Scroll to the bottom of this page and locate the `Copy URL with Secret link` next to the `GitHub type`.  
  
Click on the notepad icon to copy the payload URL.

Figure 3.11: Payload URL

- 6.7. Switch back to the GitHub webhooks page.  
  
Paste the payload URL into the `Payload URL` field.  
  
Change the `Content type` to `application/json`.  
  
Do not change any other fields.  
  
Click **Add webhook** to add the webhook.  
  
GitHub sends a test ping to the payload URL to verify availability, and displays a green check mark for the webhook, if it was successful.

Figure 3.12: Webhook active

7. Push changes to the Git repository to automatically trigger a new build.
  - 7.1. Edit the `D0101-apps/version/app.js` file in VS Code to change the response variable to version 3, as follows:

**SHOW SOLUTION**

- 7.2. Save the changes to your file.
- 7.3. Stage the changes, commit, and then push the changes to your Git repository using steps similar those used for version 2 of the application.
- 7.4. After your changes are committed and pushed to the remote Git repository, a new build is automatically triggered.  
  
In the OpenShift web console, click on **Builds** → **version**, and then click the `Builds` tab to view the list of builds for the `version` application. Notice that a new build, `version-3` has started.

Figure 3.13: Build list

- 7.5. Wait for the build to complete successfully, and for the application to be redeployed.  
  
Click `Topology`, and then click `Open URL` to view the updated application. It should now display version 3 in the output.

Figure 3.14: Version 3 of app

Close the browser tab.

8. Clean up. Delete the `youruser-version` project and remove the webhooks from GitHub.
  - 8.1. Navigate to the `Settings` page of the `D0101-apps` repository in GitHub.  
  
Click on `Webhooks` to view the webhooks page.
  - 8.2. Click **Delete** next to the active webhook, and then confirm the deletion in the resulting confirmation window. You

might be prompted for your GitHub password.

- 8.3. In the OpenShift web console, click on **Advanced** → **Projects** to view the list of projects, and then delete the *youruser-version* project. When prompted, confirm the deletion.

Enter *youruser-version* in the confirmation window, and then delete the project.

Figure 3.15: Delete version app

This concludes the guided exercise.

← [PREVIOUS \(/ROL/APP/COURSES/DO101-4.2/PAGES/CH03\)](/rol/app/courses/do101-4.2/pages/ch03)

→ [NEXT \(/ROL/APP/COURSES/DO101-4.2/PAGES/CH03S03\)](/rol/app/courses/do101-4.2/pages/ch03s03)