



Red Hat Decision Manager 7.0

Deploying the employee roster starter application for Red Hat Business Optimizer on Red Hat OpenShift Container Platform

Red Hat Decision Manager 7.0 Deploying the employee roster starter application for Red Hat Business Optimizer on Red Hat OpenShift Container Platform

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes how to run and use the OptaShift Employee Rostering example included as an add-on in Red Hat Decision Manager 7.0.

Table of Contents

PREFACE	3
CHAPTER 1. INSTALLING AND RUNNING THE EMPLOYEE ROSTERING DEMONSTRATION ON OPENSIFT	4
1.1. DEPLOYING USING THE SOURCE TO IMAGE (S2I) TEMPLATE	4
1.2. DEPLOYING WITH PRE-BUILT BINARY	6
CHAPTER 2. OVERVIEW OF THE EMPLOYEE ROSTERING STARTER APPLICATION	9
2.1. ENTERING SOURCE INFORMATION FOR EMPLOYEE ROSTERING	9
2.2. CREATING AN OPTIMAL EMPLOYEE ROSTER	12
2.3. SETTING EMPLOYEE AVAILABILITY	12
2.4. CONSTRAINTS	13
APPENDIX A. VERSIONING INFORMATION	14

PREFACE

As a business rules developer, you can test and interact with the Red Hat Business Optimizer functionality by quickly deploying the **employee-rostering** starter project included in the Red Hat Decision Manager distribution to OpenShift.

Prerequisites

- You have downloaded the Red Hat Decision Manager add-ons distribution, which includes the **employee-rostering** starter application
- You have access to a deployed OpenShift environment. For details, see the documentation for the OpenShift product that you use.

CHAPTER 1. INSTALLING AND RUNNING THE EMPLOYEE ROSTERING DEMONSTRATION ON OPENSIFT

You can deploy the Red Hat Business Optimizer Employee Rostering starter application to an OpenShift instance.

Preparing deployment files

1. Unzip the add-ons distribution (`rhdm-7.0.0-add-ons.zip`).
2. Unzip the employee-rostering zip file that is extracted from the add-ons archive.



NOTE

When you unzip the employee-rostering zip file, the `optashift-employee-rostering-7.5.0.Final-redhat-4` folder is created. This folder is the base folder in subsequent steps. File and folder names may have higher version numbers than specifically noted in this document.

There are two options for quickly deploying this application to OpenShift:

1. Using the Source to Image (S2I) template. This template pulls the source for the application from a Git repository and builds it during the deployment.
2. Using the pre-built binary application archive that is provided in the distribution. In this case, a template prepares the environment but does not rebuild the application.



NOTE

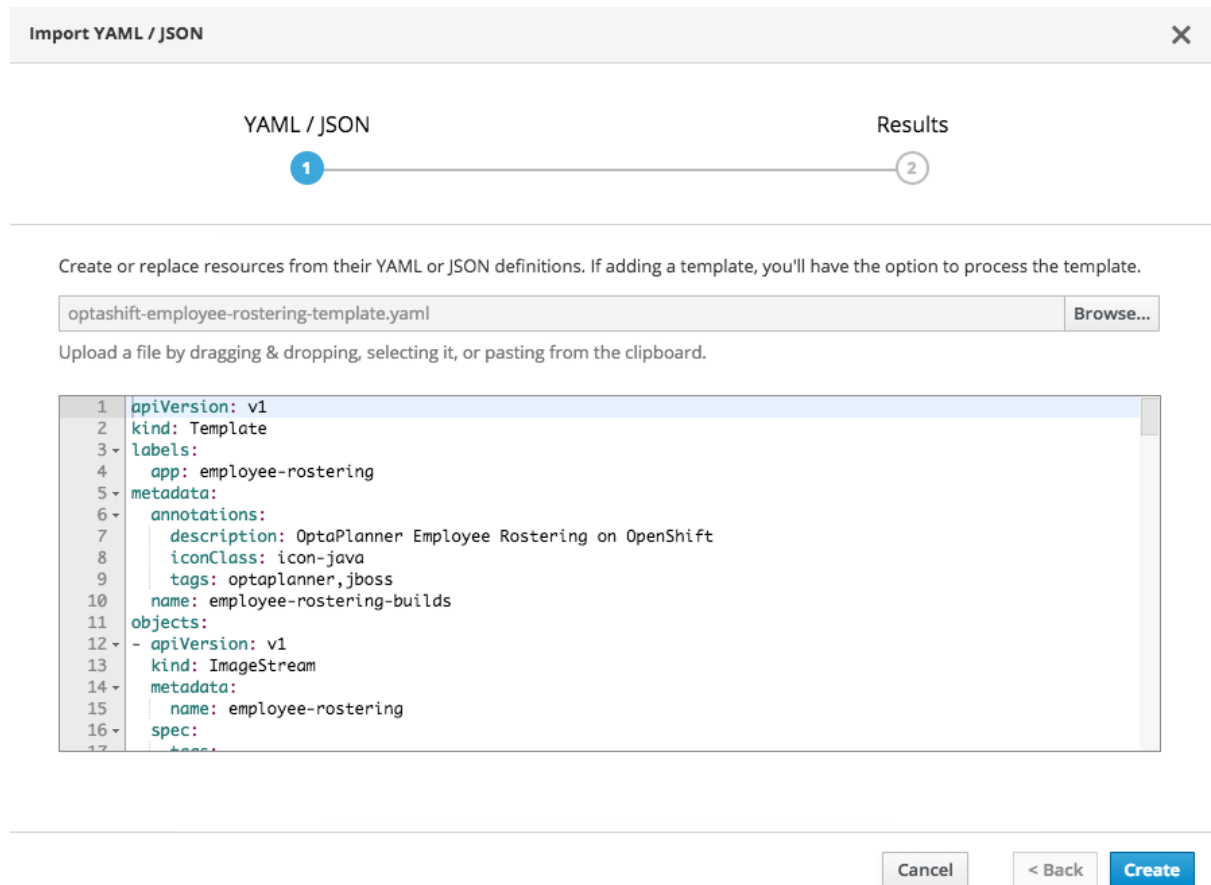
The S2I approach requires significantly more resources to complete than the binary deployment. If your OpenShift environment has less than 2GB of memory available, for example, when using the OpenShift Online Starter tier, you must follow the binary deployment approach.

1.1. DEPLOYING USING THE SOURCE TO IMAGE (S2I) TEMPLATE

Procedure

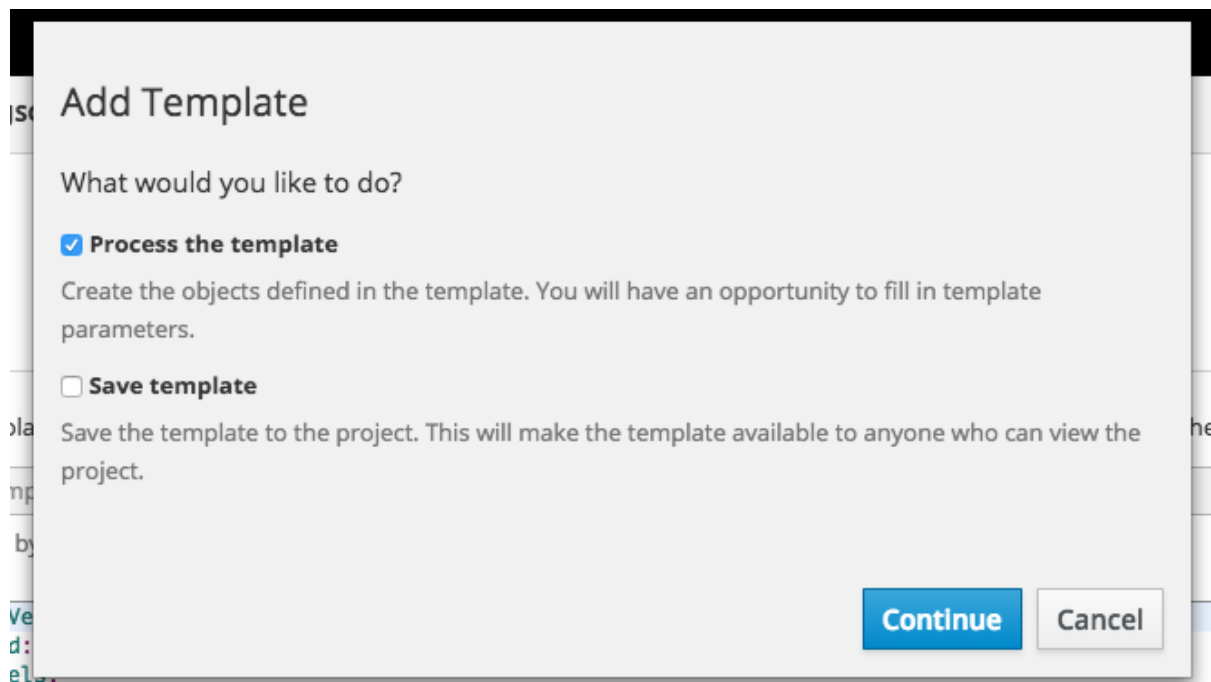
1. Log in to the OpenShift web console and click Import YAML/JSON in the upper right corner.
2. Select the project to add to from the Add to Project drop down menu, or select Create Project to create a new one. If creating a new project, enter a new project name, display name, and description.
3. Click the **Browse** button and navigate to the `optashift-employee-rostering-template.yaml` file. Starting from the `optashift-employee-rostering-7.5.0.Final-redhat-4` folder, access the file by clicking `sources > openshift > templates`
4. Click **Create**.

Figure 1.1. Upload template window



5. In the Add Template dialog box, click **Continue** to accept the default settings.

Figure 1.2. Add template window



6. In the Import YAML/JSON dialog box, click **Create** to accept the default settings.

Figure 1.3. Configure template window

The screenshot shows the 'Configure template window' for 'employee-rostering-builds'. The window has a title bar 'Import YAML / JSON' and a close button 'X'. Below the title bar is a progress indicator with three steps: 'YAML / JSON' (1), 'Template Configuration' (2, highlighted), and 'Results' (3). The main content area is titled 'employee-rostering-builds' and includes the following fields:

- Git source repository:** A text input field containing 'https://github.com/ge0ffrey/optashift-employee-rostering'.
- Git source URI for application:** A label below the first field.
- Git branch/tag reference:** A text input field containing 'master'.
- Git branch/tag reference:** A label below the second field.
- Labels:** A section with a sub-label 'Labels' and a link 'About Labels'. Below it, a text input field contains 'app' and another contains 'employee-rostering'. There is an 'Add Label' link and a close button 'X'.

At the bottom right, there are three buttons: 'Cancel', '< Back', and 'Create'.

- In the Success dialog box, click **Close**.
- Wait for the build and deployment to complete. This might take several minutes. When the deployment completes, click the link in the upper right corner and above the shaded bar to open the application.

Figure 1.4. Completed deployment and resulting link

The screenshot shows the 'Completed deployment and resulting link' page. The page title is 'APPLICATION employee-rostering'. In the top right corner, there is a link 'http://employee-rostering-optashifts2i.192.168.99.100.nip.io' highlighted with a red circle. Below the title, there is a 'DEPLOYMENT' section for 'employee-rostering, #1'. Underneath, there is a 'CONTAINER: EMPLOYEE-ROSTERING' section with details: 'Image: optashifts2i/employee-rostering c7fd3bf 698.9 MiB', 'Build: employee-rostering, #1', 'Source: Timer now only start if no error occurs 6e8de4b', and 'Ports: 8080/TCP'. To the right of this section is a circular progress indicator showing '1 pod'. Below the container section, there is a 'Networking' section with 'SERVICE Internal Traffic' (employee-rostering, 8080/TCP (8080-tcp) → 8080) and 'ROUTES External Traffic' (http://employee-rostering-optashifts2i.192.168.99.100.nip.io, Route employee-rostering, target port 8080). At the bottom, there is a 'Builds' section showing 'employee-rostering' and a status 'Build #1 is complete created an hour ago'.



NOTE

Perform a hard refresh of your browser page if the web app does not open after clicking the link.

1.2. DEPLOYING WITH PRE-BUILT BINARY

Procedure

1. Prepare the binary deployment.



NOTE

You must conduct binary deployments using the oc command line tools. See the OpenShift documentation for installation steps.

- a. Working from the **optashift-employee-rostering-7.5.0.Final-redhat-4** distribution folder, open the **binaries** folder.
 - b. Copy the **optashift-employee-rostering-webapp-7.5.0.Final-redhat-4.war** to the **sources** folder.
 - c. Rename the file to **ROOT.war**. The updated name instructs the application server to deploy the file to the root context of the server.
2. Using the command line, change to the **sources** folder.
 3. Create a project using the following command:

```
oc new-project optashift-demo --display-name="Employee Rostering Demo"
```



NOTE

Alternatively, you can use an existing project. In this case, replace all instances of **optashift-demo** in this procedure with your project name.

4. Create a build and deployment configuration. Use the **optashift-employee-rostering-template-binary.yaml** template file to set the OpenShift configuration, such as the route to expose the application to make it accessible to browsers.

```
oc process -f openshift/templates/optashift-employee-rostering-template-binary.yaml -n optashift-demo | oc create -f - -n optashift-demo
```

5. Upload the binary file to begin a deployment.

```
oc start-build employee-rostering --from-file=ROOT.war
```

6. Use the OpenShift web UI to view the details for the deployed application. Click the link in the Routes section to open the starter application.

Figure 1.5. OpenShift web console with deployed Employee Rostering starter application

Employee Rostering Demo Add to Project


Name List by Application

APPLICATION <http://employee-rostering-optashift-demo.192.168.99.100.nip.io>

employee-rostering

DEPLOYMENT

employee-rostering, #1



CONTAINER: EMPLOYEE-ROSTERING

- Image: [optashift-demo/employee-rostering 7d3fdb8](#) 526.7 MiB
- Build: [employee-rostering, #1](#)
- Source: Binary
- Ports: 8080/TCP

Networking

<p>SERVICE Internal Traffic</p> <p>employee-rostering</p> <p>8080/TCP (8080-tcp) → 8080</p>	<p>ROUTES External Traffic</p> <p>http://employee-rostering-optashift-demo.192.168.99.100.nip.io</p> <p>Route employee-rostering, target port 8080</p>
---	---

Builds

[employee-rostering](#)

✓ **Build #1 is complete** created 2 minutes ago [View Full Log](#)

```

Pushed 7/12 layers, 60% complete
Pushed 8/12 layers, 70% complete
Pushed 9/12 layers, 88% complete
Pushed 10/12 layers, 96% complete
Pushed 11/12 layers, 99% complete
Pushed 12/12 layers, 100% complete
Push successful
                    
```

CHAPTER 2. OVERVIEW OF THE EMPLOYEE ROSTERING STARTER APPLICATION

The employee rostering starter application assigns employees to shifts on various positions in an organization. For example, you can use the application to distribute shifts in a hospital between nurses, shifts in a diner between its employees, or shifts on an assembly line between workers.

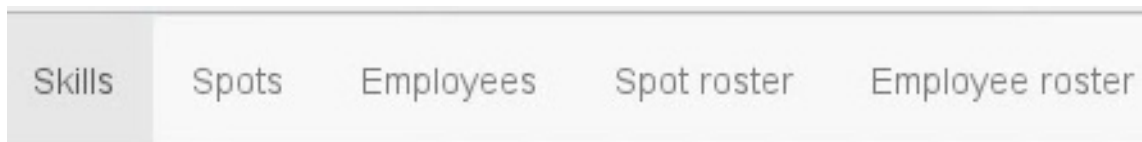
Optimal employee rostering must take a number of variables into account. For example, different skills can be necessary for shifts in different positions. Also, some employees might be unavailable for some time slots or might prefer a particular time slot. You can specify inputs for these variables using the tabs along the top of the application window.



IMPORTANT

In the current release, the employee rostering starter application fills a fixed amount of shifts in predefined dates. You can not change the dates or shifts. Therefore, it is not possible to use this application for production tasks. You can, however, update any skill, spot, or employee value after you have added them to the system.

Figure 2.1. Employee rostering toolbar



2.1. ENTERING SOURCE INFORMATION FOR EMPLOYEE ROSTERING

Tenants

In the **TENANT** field at the top right corner of the application window, you can select one of the predefined tenants. Each tenant represents an independent set of data; any change in the data for one tenant does not affect any other tenants.

You can switch between tenants to use several independent data sets, for example, to prepare employee rosters for different locations.

Skills

In the skills section, you can set all skills that can be necessary in any position within the roster. For example, a 24-hour diner would require cooking, serving, bussing, and hosting skills, in addition to skills such as general human resources and restaurant operations.

To enter these skills, click the **Skills** tab.

For each skill:

1. Type the name of the skill individually into the **Skill Name** field.
2. Click **Add**.

Skill name

Driver's license

+ Add

Spots

Spots refer to the various positions at the business. For a diner, spots include the bar, the bussing stations, the front counter, the various kitchen stations, the serving areas, and the office. For each spot, you can select a required skill from the list that you entered in the Skills tab. The following table shows a sample list of spots and skills for every spot:

Table 2.1. Spots Example

Name	Required Skill
Bar	Serving
Bussing Station 1	Bussing
Bussing Station 2	Bussing
Front Counter	Hosting, Cash register
Grill	Cooking
Prep	Food prep
Soups	Cooking
Serving area 1	Serving
Serving area 2	Serving
Office	HR

For each spot:

1. Enter the spot name into the **Spot Name** field at the bottom of the screen.
2. Select a required skill from the **Required Skill** drop-down menu.
3. Click **Add**.

Figure 2.2. Spot name example

Spot name

Required skill

+ Add

Employees

The Employees tab contains a list of all people employed by the business and the skills they possess. In this list, you can set multiple skills for an employee if necessary.

For each employee:

1. Enter the name in the **Employee Name** field at the bottom of the screen.
2. Specify a relevant skill by typing in the **Skills** field and selecting an item from the suggestion list that appears. To specify several skills, repeat this process before proceeding.
3. Click **Add**.

Figure 2.3. Employee name field

Employee name Skills + Add

Spot Roster

The Spot Roster is a table of all shifts (time slots) during which the business is open and all spots that must be manned during each shift. Each required spot is represented by a red rectangle.

Spots can support multiple employees per shift. If a spot is not required for a shift, you can remove the spot completely from the roster.

Click the blue + icon in any cell to add a new spot for the given shift. To remove a required spot completely from a shift, click the blue X icon in the corresponding cell of the grid.

Figure 2.4. Spot roster example

▶ Solve
↻ Refresh
































Spot	wednesday 2017-02-01 06:00-14:00	wednesday 2017-02-01 14:00-22:00	wednesday 2017-02-01 22:00-06:00	thursday 2017-02-02 06:00-14:00
Battery	Amy Cole    +	Unassigned  +	Unassigned  +	Unassigned  +
Bumper	Amy Fox    +	Unassigned  +	Unassigned  +	Unassigned  +
Chassis	Unassigned  +	Unassigned  +	Unassigned  +	Unassigned  +
Doors	Unassigned  +	Unassigned  +	Unassigned  +	Unassigned  +
Engine	Unassigned  +	Unassigned  +	Unassigned  +	Unassigned  +
Lights	Unassigned  +	Unassigned  +	Unassigned  +	Unassigned  +
Radiator	Unassigned  +	Unassigned  +	+	Unassigned  +

Figure 2.5. Spot icons detail (lock, edit, remove, add)



2.2. CREATING AN OPTIMAL EMPLOYEE ROSTER

Procedure

1. Populate your shifts and spots with required employees as necessary.
 - a. Click the **edit** icon of a spot to open the "Select employee" modal window.
 - b. Choose the employee from the list and click the **Assign** button.
 - c. The employee is now locked for the given spot and will not be changed by the solver.
 - d. Click the **lock** icon on the spot to allow the solver to change an assignment.
 - e. Click the **remove** icon to undo an employee assignment to a spot.
2. Click the Solve button to create an optimized solution.

The engine takes up to 30 seconds to create a fully optimal result.

When the optimal result is available, the Spot Roster window displays the name of the assigned employee for each spot and shift.

Employee Roster

The Employee Roster displays the same schedule columns as the Spot Roster, but the rows contain employee names and the spots each employee is scheduled for during a given shift.

Figure 2.6. Employee roster example

Solve
 Refresh
Click the *Solve* button to start solving.

Employee	Shift 1	Shift 2	Shift 3	Shift 4
Adam Poe			Office	
Afia Hall		Server 1		
Bob Baron		Counter		
Brian King	Bar			
Deka Allen		Bar		
Gerry Lin		Soups		
Lisa Garcia	Grill			Counter
Lynne Davis	Server 1			
Macall Short	Office			Prep
Raif Taylor		Grill		

2.3. SETTING EMPLOYEE AVAILABILITY

You can use the Employee Roster screen to set the shift preferences of employees. The engine attempts to accommodate the preferences when creating the optimal solution. You can click one of three buttons in a shift to declare preference for an employee.

Figure 2.7. Employee shift preferences



From left to right in the figure above, you can select the following options for an employee:

1. Unavailable
2. Undesired but available
3. Desired

Making no selection means the employee has no preference for the given shift.

After making changes to employee availability, click the **Solve** button to create a new schedule.



NOTE

The Unavailable option also applies for employees who call in sick or are on vacation. If an employee calls out just before a shift, a manager can indicate the scheduled employee's unavailability and then create a new schedule.

2.4. CONSTRAINTS

The Red Hat Business Optimizer rules for this starter application use both hard and soft constraints. During an optimization, the engine may not violate hard constraints, for example, if an employee is unavailable (out sick), or that an employee cannot work two spots in a single shift. The engine tries to adhere to soft constraints, such as an employee's preference to not work a specific shift, but can violate them if the optimal solution requires it.

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on: Monday, October 1, 2018.