



# Red Hat Managed Integration 2

## Creating Solution Patterns for Red Hat Managed Integration 2

For Red Hat Managed Integration 2



# Red Hat Managed Integration 2 Creating Solution Patterns for Red Hat Managed Integration 2

---

For Red Hat Managed Integration 2

## Legal Notice

Copyright © 2020 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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

The OpenStack<sup>®</sup> 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 provides instructions for creating Solution Patterns in Red Hat Managed Integration 2.

---

## Table of Contents

CHAPTER 1. INTRODUCING SOLUTION PATTERNS .....	3
CHAPTER 2. DOWNLOADING THE SOLUTION PATTERN TEMPLATE .....	4
CHAPTER 3. SOLUTION PATTERN FILE STRUCTURE .....	5
CHAPTER 4. COMMITTING YOUR SOLUTION PATTERN TO A REMOTE REPOSITORY .....	6
CHAPTER 5. ADDING SOLUTION PATTERNS TO YOUR MANAGED INTEGRATION CLUSTER .....	7
CHAPTER 6. EDITING THE OVERVIEW SECTION OF A RED HAT MANAGED INTEGRATION SOLUTION PATTERN .....	9
CHAPTER 7. EDITING TASK SECTIONS OF A RED HAT MANAGED INTEGRATION SOLUTION PATTERN	10
CHAPTER 8. EDITING PROCEDURES OF A RED HAT MANAGED INTEGRATION SOLUTION PATTERN ...	12
CHAPTER 9. EDITING RESOURCES FOR A RED HAT MANAGED INTEGRATION SOLUTION PATTERN ..	13
CHAPTER 10. EDITING TASK RESOURCES FOR A RED HAT MANAGED INTEGRATION SOLUTION PATTERN	14
CHAPTER 11. ADDING VERIFICATIONS TO A RED HAT MANAGED INTEGRATION SOLUTION PATTERN .	15
CHAPTER 12. ADDING ATTRIBUTES TO A RED HAT MANAGED INTEGRATION SOLUTION PATTERN ...	16
CHAPTER 13. DEFINE METADATA FOR YOUR SOLUTION PATTERN REPOSITORY .....	17



# CHAPTER 1. INTRODUCING SOLUTION PATTERNS

Solution Patterns are templates that shorten development time and encourage best practices by providing opinionated instructions about how to develop cloud-native, integrated applications on your Red Hat Managed Integration cluster. You can create a Solution Pattern to communicate technical concepts and provide guidance for users in your organization. Your Solution Pattern should document a clear user story that addresses a goal.

To write a Solution Pattern you require a knowledge of:

- AsciiDoc: A lightweight markup language that is similar to Markdown.
- JSON: A lightweight format for storing and transporting data.
- Git: An open source distributed version control system. You must push your AsciiDoc and JSON files to a Git repository accessible from your cluster, for example, GitHub or GitLab.
- OpenShift: A platform for developing and running containerized applications. You can add Solution Patterns using the OpenShift console.

You must create and edit an AsciiDoc file to construct a Solution Pattern. The AsciiDoc file should resemble the following example:

```
= Demonstrating features 1  
  
A short description for the feature. 2  
  
== Doing a task 3  
  
. Do this first: 4  
. Do this second:
```

- 1** Solution Pattern title. Only one level 1 heading per file is allowed.
- 2** The Solution Pattern description is displayed on the Solution Explorer home page. Do not use AsciiDoc attributes in this line because they are not displayed correctly on the homepage.
- 3** Optional subtasks describe how to complete the task.
- 4** A numbered list of steps the user must perform.

## Additional resources

- For more information on writing in AsciiDoc, see the [AsciiDoc Homepage](#)

## CHAPTER 2. DOWNLOADING THE SOLUTION PATTERN TEMPLATE

To start developing your Solution Pattern, download the template repository. The template contains:

- The directory structure
- An example Solution Pattern
- An example manifest file

### Procedure

1. Clone the template repository

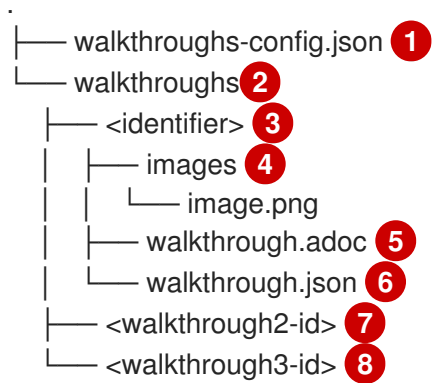
```
git clone https://github.com/integr8ly/walkthrough-template.git
```

2. Change to the **walkthrough-template** directory:

```
cd walkthrough-template
```



## CHAPTER 3. SOLUTION PATTERN FILE STRUCTURE



- 1 This is an optional file that defines metadata for the set of Solution Patterns in the repository.
- 2 This is a required directory. All Solution Patterns are defined under **walkthroughs**.
- 3 7 8 The directory name of a Solution Pattern. This is also displayed in the URL of the Solution Explorer.
- 4 All of the images for the Solution Pattern.
- 5 **walkthrough.adoc** is where the content of the Solution Pattern is written in AsciiDoc format.
- 6 **walkthrough.json** is the manifest file of the Solution Pattern where you can define dependencies.

The default **walkthrough.json** does not define any dependent services:

```

{
  "dependencies": {
    "repos": [],
    "serviceInstances": []
  }
}

```

## CHAPTER 4. COMMITTING YOUR SOLUTION PATTERN TO A REMOTE REPOSITORY

Push your repository to GitHub or GitLab.

### Procedure

1. Create a new remote repository and take note of the remote URL.
2. Add the remote repository as you upstream repository.

```
git remote add upstream <remote URL>
```

3. Commit the changes that you added to your local repository.

```
git commit -m "Add commit message"
```

4. Push the changes in your local repository to GitHub.

```
git push -u upstream master
```

## CHAPTER 5. ADDING SOLUTION PATTERNS TO YOUR MANAGED INTEGRATION CLUSTER

The home page of the Solution Explorer lists the Solution Patterns from all the Git repositories you are subscribed to. Any developer can add Solution Patterns to your cluster.

### Procedure

1. Navigate to the **Solution Explorer**.
2. Click the gear icon in the top right to display the **Application settings** screen.
3. Enter the URLs of the Solution Pattern Git repositories you want to add to your cluster using the following syntax:

```
https://github.com/<org>/<repo>.git
```

where **<org>** is the name of your GitHub organization and **<repo>** is the name of your repository.



### NOTE

- List URLs in the order you want them to appear in the Solution Explorer.
- Enter one URL per line.
- To include a specific branch, append **#<branch-name>** to the url. For example:

```
https://github.com/<org>/<repo>.git#version-one
```

4. Click **Save**.  
This triggers an automatic refresh of the Solution Explorer.
5. When the deployment is complete, refresh your browser.  
You should now see new Solution Patterns available from the dashboard.
6. If the Git repository is updated with new content, the Solution Explorer is not automatically updated.  
Refresh the Solution Explorer to view the changes:
  - a. Click the gear icon in the top right to display the **Application settings** screen.
  - b. Click Save to trigger a refresh of the Solution Explorer app.
  - c. When the Solution Explorer refresh is complete, refresh your browser.
  - d. Navigate to the **Solution Patterns** tab to see the updated content.



### NOTE

You can access the Git repository that contains the Solution Pattern source code by clicking the **Repository** link located in the upper right corner of each group of Solution Patterns on the **All Solutions Patterns** tab in the Solution Explorer.

### Additional resources

- To learn more about creating solution patterns, see the [Creating Solution Patterns documentation](#).

## CHAPTER 6. EDITING THE OVERVIEW SECTION OF A RED HAT MANAGED INTEGRATION SOLUTION PATTERN

The first section of the **walkthrough.adoc** describes your Solution Pattern. The overview section outlines the purpose of the Solution Pattern. The overview is displayed when a user clicks on the Solution Pattern in the Solution Explorer.

- The overview is defined by the content before the first task.
- The title of the overview is the name of the Solution Pattern.
- The overview heading is followed by a short introduction to the Solution Pattern.

The overview description is displayed in the Solution Explorer dashboard for the Solution Pattern.

### Prerequisites

- You have created a repository using the Solution Pattern template.
- You have added the Solution Pattern repository to your cluster as described in [Chapter 5, Adding Solution Patterns to your Managed Integration cluster](#).

### Procedure

1. Edit the **walkthroughs/1-template-walkthrough/walkthrough.adoc** file.
2. Change the title of the Solution Pattern, for example:

```
┆ = My first Solution Pattern
```

3. Add content for the overview section.
4. Commit your changes.

```
┆ git commit -am "add overview"
```

5. Push your changes to the remote repository.

```
┆ git push upstream master
```

To see your changes in Solution Explorer, refresh Solution explorer by clicking the gear icon followed by **Save**.

## CHAPTER 7. EDITING TASK SECTIONS OF A RED HAT MANAGED INTEGRATION SOLUTION PATTERN

Tasks are procedures in the **walkthrough.adoc** file that guide a user through a Solution Pattern.



### NOTE

You can create another level of subtasks by creating a third level heading (**===**). Subtasks are displayed in the Solution Explorer on the parent task page.

### Prerequisites

- You have created a repository using the Solution Pattern template.
- You have added the Solution Pattern repository to your cluster as described in [Adding Solution Patterns](#).

### Procedure

1. Edit the **walkthroughs/1-template-walkthrough/walkthrough.adoc** file.
2. Change the task title, for example:

```
== Doing the first task
```

3. Write a introduction about the task.
4. Change the task timing estimate. Each task has an attribute to communicate the approximate time that the task should take the user to perform.  
This is displayed in the Solution Pattern overview as part of the task list.

For example to change the estimated time for the first task to 12 minutes:

```
[time=12]
== Doing the first task
```

5. Edit the numbered list:

```
. First step
. Second step
. Third step
```

6. Save, commit, and push your changes.



### NOTE

The content of the Solution Explorer does not change automatically after a change in the subscribed repository. You must redeploy the Solution Explorer for the change to be visible.

7. Redeploy the Solution Explorer:
  - a. Click the gear icon in the top right to display the **Application settings** screen.

- b. Click Save to trigger a redeployment of the Solution Explorer app.
- c. When the deployment is complete, refresh your browser.
- d. Navigate to the **Solution Patterns** tab to see the updated content.

## CHAPTER 8. EDITING PROCEDURES OF A RED HAT MANAGED INTEGRATION SOLUTION PATTERN


A procedure is a set of steps that you perform to complete a task or subtask.

Write procedures using AsciiDoc numbered lists. Prepend each step with a dot (.) to create a simple numbered list.

### Prerequisites

- You have created a repository using the Solution Pattern template.
- You have added the Solution Pattern repository to your cluster as described in [Chapter 5, Adding Solution Patterns to your Managed Integration cluster](#).

### Procedure

1. Edit the **walkthroughs/1-template-walkthrough/walkthrough.adoc** file in the template repository.
2. Edit the numbered list:  
 . First step  
  . Second step  
  . Third step
3. Save, commit, and push your changes.

To see your changes in Solution Explorer, refresh Solution explorer by clicking the gear icon followed by **Save**.



## CHAPTER 9. EDITING RESOURCES FOR A RED HAT MANAGED INTEGRATION SOLUTION PATTERN

The Solution Explorer displays information and links on the right side pane in Solution Explorer. You can add Solution Pattern resources by using block attributes. You can only define Solution Pattern resources in the overview section.



### NOTE

- Setting **serviceName** is optional.
- If **serviceName** is set to the name of a Red Hat Middleware service, an icon indicating the service status is displayed next to the resource.
- For a list of default services, see [the value for DEFAULT\\_SERVICES](#).
- Setting the **window="\_blank"** parameter for a Solution Pattern resource link is also optional, but ensures that the target of the link displays in a separate browser tab.

### Prerequisites

- You have created a repository using the Solution Patterns template.
- You have added the Solution Pattern repository to your cluster as described in [Chapter 5, Adding Solution Patterns to your Managed Integration cluster](#).

### Procedure

1. Edit the **walkthroughs/1-template-walkthrough/walkthrough.adoc** file.
2. Add Solution Pattern resources using block attributes to the end of the overview section, for example:

```
[type=walkthroughResource,serviceName=openshift]
.OpenShift
****
* link:{openshift-host}[Openshift Console, window="_blank"]
****
```

3. Save, commit, and push your changes.

To see your changes in Solution Explorer, refresh Solution explorer by clicking the gear icon followed by **Save**.

## CHAPTER 10. EDITING TASK RESOURCES FOR A RED HAT MANAGED INTEGRATION SOLUTION PATTERN

The Solution Explorer displays helpful information and links on the right side panel that are specific to a task. Solution Pattern resources are displayed for every task. Task resources typically change as you navigate the Solution Pattern.

### Prerequisites

- You have created a repository using the Solution Pattern template.
- You have added the Solution Pattern repository to your cluster as described in [Chapter 5, Adding Solution Patterns to your Managed Integration cluster](#).

### Procedure

1. Edit the **walkthroughs/1-template-walkthrough/walkthrough.adoc** file.
2. At the end of any task section, add the following:

```
[type=taskResource]
.Useful links
****
* link:https://google.com[Task related link]
****
```

3. Save, commit, and push your changes.

To see your changes in Solution Explorer, refresh Solution explorer by clicking the gear icon followed by **Save**.

## CHAPTER 11. ADDING VERIFICATIONS TO A RED HAT MANAGED INTEGRATION SOLUTION PATTERN

Verifications are interactive elements in a Solution Pattern that confirm whether a user has successfully completed a procedure. They are questions presented to the user, which can be answered with either a **Yes** or **No** response.

### Prerequisites

- You have created a repository using the Solution Pattern template.
- You have added the Solution Pattern repository to your cluster as described in [Chapter 5, Adding Solution Patterns to your Managed Integration cluster](#).

### Procedure

1. Edit the **walkthroughs/1-template-walkthrough/walkthrough.adoc** file.
2. At the end of the procedure section following the numbered list, add:

```
[type=verification]
Check that the dashboard of service X reports no errors.
```

3. Add the following block for users that answer the question with **No**:

```
[type=verificationFail]
Try turning it off and on again.
```

4. Save, commit, and push your changes.

To see your changes in Solution Explorer, refresh Solution explorer by clicking the gear icon followed by **Save**.

## CHAPTER 12. ADDING ATTRIBUTES TO A RED HAT MANAGED INTEGRATION SOLUTION PATTERN

Attributes enable you to use variables in your Solution Pattern. The syntax for using attributes is `{<attribute name>}`. Solution Patterns have access to predefined attributes:

- Access a Middleware service defined route: `{route-<route name>-host}`
- AMQ Online Broker URL: `{enmasse-broker-url}`
- AMQ Online Credential Username: `{enmasse-credentials-username}`
- AMQ Online Credential Password: `{enmasse-credentials-password}`
- AMQ Online URL: `{enmasse-url}`.
- CodeReady Workspaces URL: `{che-url}`.
- API Management URL: `{api-management-url}`
- Fuse URL: `{fuse-url}`
- Launcher URL: `{launcher-url}`
- OpenShift console: `{openshift-host}`
- OpenShift app host: `{openshift-app-host}`

### Prerequisites

- You have created a repository using the Solution Pattern template.
- You have added the Solution Pattern repository to your cluster as described in [Chapter 5, Adding Solution Patterns to your Managed Integration cluster](#).

### Procedure

1. Edit the `walkthroughs/1-template-walkthrough/walkthrough.adoc` file.
2. Add another step and include the attribute value:  

```
█ . Log in to the link:{openshift-url}[Red Hat OpenShift Console].
```
3. Save, commit, and push your changes.
4. Redeploy the Solution Explorer:
  - a. Click the gear icon in the top right to display the **Application settings** screen.
  - b. Click Save to trigger a redeployment of the Solution Explorer app.
  - c. When the deployment is complete, refresh your browser.
  - d. Navigate to the **Solution Patterns** tab to see the updated content.

## CHAPTER 13. DEFINE METADATA FOR YOUR SOLUTION PATTERN REPOSITORY

By default, the Solution Patterns from a repository are displayed in the Solution Explorer with a title corresponding to your repository name. You can add a **walkthroughs-config.json** file with the following content if, for example, you want to title the group **My Solution Patterns**:

```
{  
  "prettyName": "My Solution Patterns"  
}
```

*Revised on 2020-09-17 09:11:54 UTC*