# NewgenONE

## Deployment Admin

## User Guide

Version: 2024.2

# Disclaimer

This document contains information proprietary to Newgen Software Technologies Ltd. User may not disclose or use any proprietary information or use any part of this document without written permission from Newgen Software Technologies Ltd.

Newgen Software Technologies Ltd. makes no representations or warranties regarding any software or to the contents or use of this guide. It also specifically disclaims any express or implied warranties of merchantability, title, or fitness for any particular purpose. Even though Newgen Software Technologies Ltd. has tested the hardware and software and reviewed the documentation, it does not guarantee or imply that this document is error free or accurate regarding any particular specification. As a result, this product is sold as it is and user, the purchaser, is assuming the entire risk as to its quality and performance. Further, Newgen Software Technologies Ltd. reserves the right to revise this publication and make changes in its content without any obligation to notify any person, of such revisions or changes. Newgen Software Technologies Ltd. authorizes no Newgen agent, dealer or employee to make any modification, extension, or addition to the above statements.

Newgen Software Technologies Ltd. has attempted to supply trademark information about company names, products, and services mentioned in this document. Trademarks indicated below were derived from various sources.



**Newgen Corporate Office, New Delhi**
E-44/13
Okhla Phase- II
New Delhi - 110020
INDIA
Phone: +91 1146 533 200
info@newgensoft.com

# Contents

# Preface

This chapter provides information about the purpose of this guide, details on the intended audience, revision history, and related documents for NewgenONE Deployment Admin.

# Revision history

| Revision date | Description |
|---|---|
| November 2024 | Initial publication |

# About this guide

This guide describes how to create, execute, monitor, and manage pipelines for the deployment of NewgenONE and iBPS artifacts using Deployment Admin. This guide also explains how to associate cabinets and configure environments for deployment, manage various server-agent configurations, create and manage vaults, tokens, users, and roles.

To ensure you are referring to the latest and most recent revision of this guide, download it from one of the following locations:

- Newgen Internal Doc Portal, if you are a Newgen employee.
- Newgen Partner Portal, if you are a Newgen partner.

# Intended audience

This guide is intended for DevOps engineer, IT administrators, and release managers. The reader must have a technical background with an understanding of NewgenONE Deployment Admin. The user must have the necessary rights to access and work

with Deployment Admin for the deployment of artifacts created in NewgenONE or iBPS.

You must have a working knowledge of the following:

- The fundamentals and standard practices of your business area
- Basic understanding of NewgenONE or iBPS artifacts such as portal, process, business rule, and business reports.

# Related documents

The following documents are related to NewgenONE Deployment Admin:

- NewgenONE Overview Guide
- NewgenONE Admin Installation Guide

# Documentation feedback

To provide feedback or any improvement suggestions on technical documentation, you can write an email to docs.feedback@newgensoft.com.

To help capture your feedback effectively, share the following information in your email:

- Document name
- Version
- Chapter, topic, or section
- Feedback or suggestions

# Introducing Deployment Admin

Deployment Admin is a dedicated module of NewgenONE to configure and run continuous integration and deployment of the following artifacts to production, test, user acceptance testing (UAT), or system integration testing (SIT) environment:

- Portals
- Processes
- Reports
- Rules

Deployment Admin offers an easy-to-use user interface for the NewgenONE and iBPS users for creating pipelines to deploy the artifacts across different environments. With Deployment Admin you can also configure and deploy the following post-deployment services:

- Criteria management
- Process queue user group mapping
- Advanced search configuration
- Audit log configuration
- Process specific services:
    - Process server
    - Mailing agent
    - Message agent
    - Export utility
    - File Upload utility
    - Initiation agent

Deployment Admin provides you with pre-defined tasks to create a pipeline. You can also configure these tasks as per the requirement of the particular pipeline. The built-in user management feature of the Deployment Admin customizes the user interface according to the role and rights assigned to a user. This allows administrators to easily

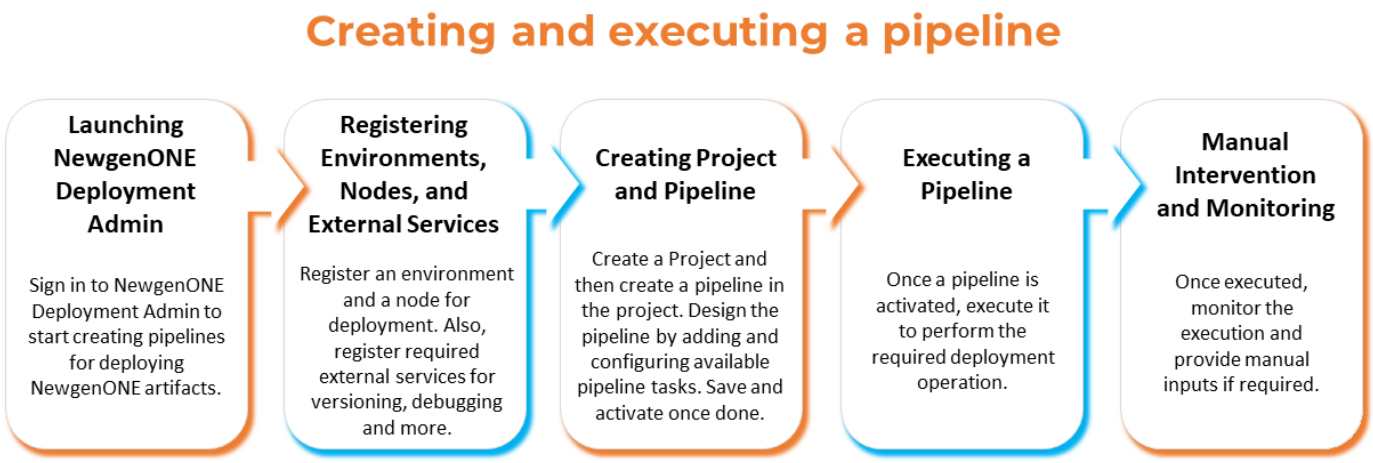
manage the level of user access through the rights and roles options giving the administrator fine control.

Following are the key features of the Deployment Admin:

- Supports usage of multiple nodes in the same environment. This allows simultaneous testing and deployment of multiple nodes running in the same environment through a single pipeline.
- Supports manual approval process. This allows the pipeline execution to halt for manual approval upon executing a pre-defined task.
- Supports rights management for granular control over user rights and roles.
- Vault functionality allows secure usage of sensitive information.
- Supports audit logs feature for tracking user activity.

Furthermore, you can navigate through the elements of the deployment manager user interface using the Tab key.

The following workflow diagram displays a typical workflow to create and execute pipelines to meet the deployment requirements for NewgenONE and iBPS artifacts.



# Getting started

To start using the Deployment Admin, you need to sign in to the Deployment Admin portal and then associate the required cabinets. This chapter describes the procedure for:

- Accessing Deployment Admin
- Associating a cabinet
- Configuring email account

# Accessing Deployment Admin

This topic describes the prerequisites and the procedure for signing in to the Deployment Admin.

## Prerequisites

To sign in to the Deployment Admin, you must have the following prerequisites:

- A valid username and password

  The default credential is:
  - Username: dmadmin
  - Password: admin123#

  The administrator can change the password for the default user and also create one more administrative user if required.

- Working URL of Deployment Admin
  The format of a typical URL:
  *https://<IP address>:<Port>/deploymentweb/*
  Where, <IP address> specifies the internet protocol (IP) address of the machine where the server is running, and <Port> specifies the port number of the server machine. For example: *https://127.0.0.1:6060/deploymentweb/*

- The appropriate role assigned to you
- The appropriate cabinet information

# Signing in to Deployment Admin

To sign in to the Deployment Admin, perform the following steps:

1. Open the Deployment Admin URL in a web browser. The sign-in page appears.

   To log in as a cabinet user, the user must be part of the supervisor group.

2. (Optional) To change the locale of Deployment Admin, click the locale dropdown and select the required option from the list. You can set the locale either to English or Arabic based on your business requirements. By default, the locale is set to English.
3. Enter the following details:

| Field | Operation |
|---|---|
| User type | Select the required user type from this dropdown.<br>• **Admin** — Select this option to sign in as an administrator.<br>• **Cabinet** — Select this option to sign in as a cabinet user. Selecting this option displays the Cabinet field. |
| Username | Enter your Deployment Admin username. |
| Password | Enter the password for the entered username. |
| Cabinet | Select the appropriate cabinet from the dropdown list if you are a cabinet user. |
| Captcha | Enter the captcha given. |

   Captcha verification can be enabled or disabled through the system settings by the admin.

4. (Optional) Select the **Remember Me** checkbox to allow the browser to remember your username.
5. Click **Login**. Upon successful verification of the credentials, you are signed in to the Deployment Admin.

Only an administrator can sign in to the Deployment Admin for the first time with the default credentials. To know more about the default credentials check the Prerequisites section. After

signing in for the first time the administrator must associate the required cabinets in order to import cabinet users from OmniDocs. Check Associating Cabinets to know more.

# Associating a cabinet

After signing in to the Deployment Admin for the first time, the administrator must associate the required cabinets. Once associated, the Deployment Admin can use those cabinets for user information and other operations.

To associate a cabinet, perform the following steps:

1. Sign in to Deployment Admin as an administrator. The Cabinet Configuration page appears.
2. Click **Associate Cabinets**. The Associate Cabinet dialog appears.
3. Enter the required details or select the required options in the following fields:

| Field | Operation |
|---|---|
| Secure | Select this checkbox to associate a cabinet over a secure connection. |
| Docker Environment | Select this checkbox to associate a cabinet from an environment available on Docker. |
| Machine Host Name | Select this option to get cabinets using the hostname of the source environment.<br>Enter the following details:<br>• **Host Name** — Enter the hostname of the NewgenONE environment where the cabinet is available<br>• **App Server Type** — Select the type of application server that is hosting the cabinet<br>• **AppServer Name/IP (EJB)** — This field is applicable for Docker environments. Enter the name or IP address of the application server that is hosting the cabinet.<br>• **Port** — Enter the port number for the environment where the cabinet is available<br>• **Product Version** — Select the required product version. The supported versions are NewgenONE 2023.1, NewgenONE 2023.2, and iBPS 5.0 SP3 Patch1. |

| Field | Operation |
|---|---|
| Domain Name | Select this option to get cabinets using the domain name of the source environment.<br>Enter the following details:<br>• **Domain Name** — Enter the appropriate NewgenONE domain name where the required cabinet is present.<br>• **App Server Type** — Select the type of application server that is hosting the cabinet<br>• **AppServer Name/IP (EJB)** — This field is applicable for Docker environments. Enter the name or IP address of the application server that is hosting the cabinet.<br>• **Product Version** — Select the required product version. The supported versions are NewgenONE 2023.1, NewgenONE 2023.2 and iBPS 5.0 SP3 Patch1. |

4. Click **Get Cabinets**. The list of available cabinets appears under the Available Cabinets column.
5. Select the required cabinets. The selected cabinets move under the Selected Cabinets column.
6. (Optional)Enable the toggle to add Oauth configuration to the selected cabinets.
7. Click **Associate**. The cabinets are now associated with the Deployment Admin.

## Related topic(s)

- Signing in to Deployment Admin
- Configuring email account

# Configuring an email account

The administrator must configure an email account for Deployment Admin to use for sending email notifications. You can sign in to Deployment Admin as an administrator and configure an email account for this purpose.

You must have administrator rights to configure an email account.

To configure an email account, perform the following steps:

1. Sign in to Deployment Admin as an administrator. The Cabinets page appears.
2. Click **Email Configuration,** the Email Configuration page appears.
3. Click Edit to make changes to the given fields.
4. Enter the required details in the following fields:

| Field | Operation |
|---|---|
| SMTP Host | The SMTP Host of the email service. |
| SMTP Port | TheSMTP port number for the email service. |
| Username | The username for signing in to the account. |
| Password | The password for account authentication. |

5. Click **Modify**. The email account gets configured.

## Related topic(s)

- Signing in to Deployment Admin
- Associating Cabinets

# Configuring system settings

The administrator can enable or disable captcha in the system settings.

You must have administrator rights to configure system settings.

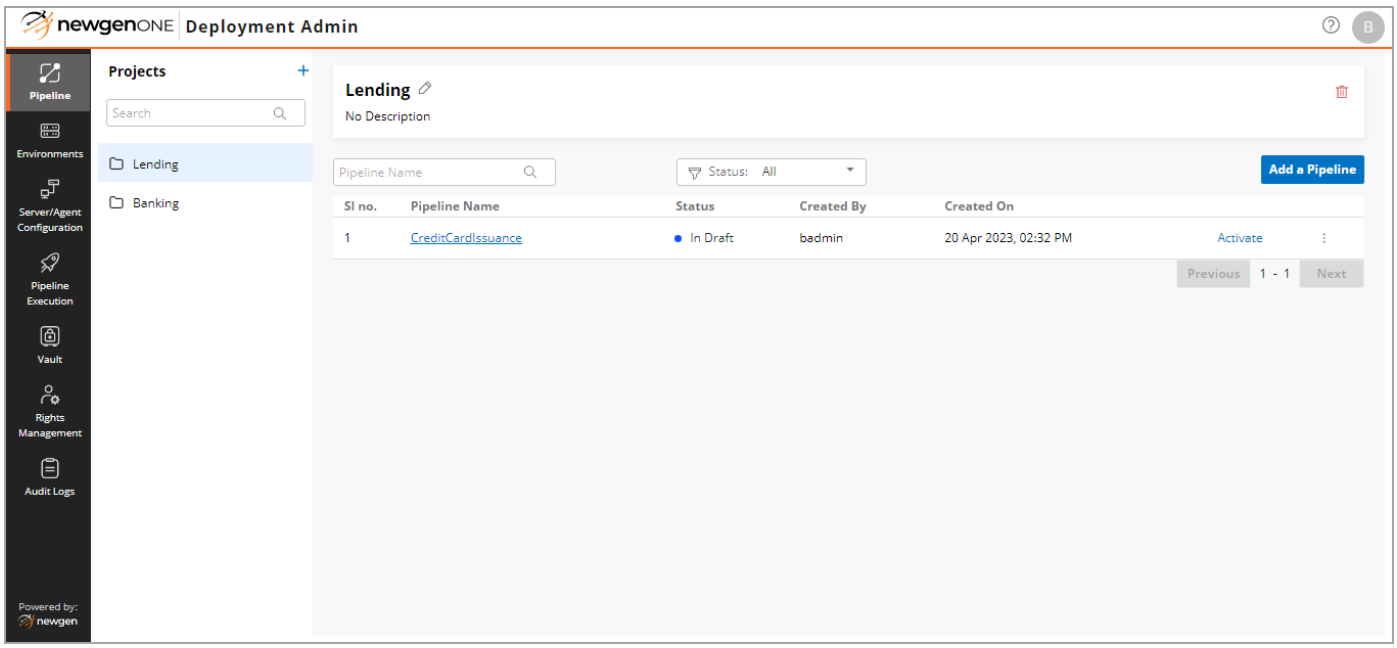
To configure system settings, perform the following steps:

1. Sign in to Deployment Admin as an administrator. The Cabinets page appears.
2. Click **System Settings,** the System Settings page appears.
3. Enable the toggle to allow captcha verification. The captcha verification is enabled for both cabinet user and admin user.

# Creating and managing pipelines

The Pipeline menu allows you to create, edit, and delete projects and pipelines.
The Pipeline page of the Deployment Admin displays the following information and options:

| Field | Operation |
|---|---|
| Search projects | The search box provides the option to search a project by name. |
| Project list | Shows the projects created by all users in the same cabinet. |
| Create project | This icon allows you to create a project. |
| Search pipeline | The search box provides the option to search a pipeline in the selected project by name. |
| Filter pipeline | Allows you to filter the pipelines in the selected project by status. |
| Add Pipeline | This button allows you to create a new pipeline. |
| Pipeline menu | Clicking the pipeline menu icon ⋮ provides you with the following options:<br>• **Edit** — Selecting this option opens the pipeline in edit mode.<br>• **Delete** — Selecting this option allows you to delete the pipeline |
| Pipeline list | Displays the list of pipelines in the selected project. |
| Pipeline name | Clicking the pipeline name opens the pipeline in edit mode. Click here to know more about Editing a pipeline. |
| Activate or Deactivate | Clicking Activate or Deactivate on a pipeline allows you to activate or deactivate the pipeline. |




This chapter describes:

- Creating a project
- Managing projects

# Creating a project

Deployment Admin allows you to create projects that can store pipelines. Projects help you in categorizing pipelines as per their relevance. One project can have multiple pipelines associated with it.

> A Release project is created by default to design the release pipelines. It means you can create a release pipeline only under the Release project.

To create a project, perform the following steps:

1. Navigate to the Pipeline tab.
2. Click the **Add** icon + in the Projects section. The Add Project dialog appears.
3. In the Add Project dialog, enter the Project Name and Description of the project in their respective fields.

> The description is an optional field. Adding a description to a project can help in understanding the nature of the project.

4. Click **Add**. The project now appears in the Projects section and a popup message confirms the creation of the project.

### Related topic(s)

- Creating a pipeline
- Working with pipeline tasks
- Managing projects

# Creating a pipeline

A pipeline is a combination of sequential building and deployment tasks to deploy specific packages into the testing or production environment. You can create two types of pipelines using Deployment Admin — integration and release pipelines.

- **Integration pipeline** — It is a continuous integration pipeline for deploying artifacts into the target environment. This is a continuous cycle of building, testing, and deploying artifacts. For example, deploying the portal from the Development environment to the System Integration Testing (SIT) environment and then to the User Acceptance Testing (UAT) environment. For more information, see Creating an integration pipeline.
- **Release pipeline** — A release pipeline uses the package generated by an integration pipeline. A release pipeline is a continuous delivery pipeline for deploying the package into the live or production environment. For example, use a release pipeline to deploy a package on the production environment. For more information, see Creating a release pipeline.

You can create multiple pipelines within a project.

### Related topic(s)

- Creating an integration pipeline
- Creating a release pipeline

# Creating an integration pipeline

An integration pipeline is used to deploy the artifacts into the target environment. A new package is generated each time you execute the pipeline.

To create an integration pipeline in a project, perform the following steps:

1. Navigate to the **Pipeline** tab. On the left pane, a list of existing projects appears.
2. In the Projects section, select the required project where you want to store the pipeline.
3. Click the **Add a Pipeline** button. The Choose an Integration Template section appears.
4. Click the **Create from Scratch** button to create a pipeline from scratch or choose any of the following templates using the **Select this template** button to create a pipeline:
    - **Deployment Pipeline** — Use this template to design a pipeline that fetches the artifacts from the source environment, creates and uploads the artifacts package with the given name in the selected artifactory, and then deploys those artifacts on the given environment or node.
    - **Deployment pipeline with manual approval** — Use this template to design a pipeline that gets the artifacts from the source environment, creates and uploads the artifacts package with the given name in the selected artifactory, and then deploys those artifacts on the given environment or node. It also includes a manual approval task for securing deployments to the production environment.
    - **Deployment pipeline with the email notification task** — Use this template to design a pipeline that gets the artifacts from the source environment, creates and uploads the artifacts package with the given name in the selected artifactory, and then deploys those artifacts on the given environment or node. It also includes a manual approval task for securing deployments to the production environment and a notification trigger task for sending email notifications to the release managers.

    The Add Pipeline dialog appears.
5. Enter the following details in the dialog:
    - **Pipeline Name** — Enter the name of the integration pipeline.
    - (Optional) **Description** — Enter the description for the integration pipeline.
6. Click **Add**. The Pipeline Editor page appears.

7. Click **+Add Task** to add a pre-defined task from the All, Build, Deploy, Approval, Conditional, or Jenkins categories under the required stage. The following tasks are available under the different categories:
    - GetSource
    - Create Package
    - Deployment
    - Post Deployment Configuration
    - Manual Approval
    - Notification Trigger
    - Conditional Task
    - Jenkins Task
    - Upload Package
    - Download Package
8. Once added, double-click the task to configure its related properties. In case you used a pre-defined template for creating the pipeline, then tasks are added by default. However, you can add and modify the tasks. For more information, see Working with pipeline tasks.
   Additionally, you can click the add icon + next to the stage label to insert another stage, and later add the required tasks to it. To define parameters and checklists to use in the pipeline, click the **Parameters** and **Checklist** icons on the page's upper-right section. For more information, see Creating a parameter and Creating a checklist.
9. Click **Save** to confirm the changes. The pipeline now appears in the Release Pipelines list. After creation, you can also edit a pipeline. For more information, see Editing a pipeline.
10. Click **Activate**. A pop-up appears for configuring the execution settings either manually or by scheduling.
11. By default **Manual** is selected. To schedule pipeline execution select **Schedule.**
12. Enter the following fields:

| Field | Operation |
|---|---|
| Recurrence | Select the frequency you want to set, from this dropdown.<br>• Daily - Allows to run the pipeline daily<br>• Weekly - Allows to run the pipeline on the days you have scheduled it to.<br>• Custom - |

| Field | Operation |
|---|---|
| Days | If you set the frequency to weekly, you can choose any day from Monday to Sunday as the execution day(s). |
| At | Select the time at which you want the pipeline to be executed. |
| Cron | Specify when the job should run using a cron expression. This expression defines the time and date parameters for the job. |
| Username | Enter the username of the account that will be used to invoke the pipeline. |
| Password | Enter the password for the entered username. |

13. Click **Save & Next** to save the execution settings and move to the next step.
14. The flexible parameters defined for the pipeline will appear under **Pipeline Parameters.** To define the parameters, see Creating a parameter.
15. You can click the vault icon next to the flexible parameter field to enter its value from a vault.
16. Click the ellipsis icon ••• icon. A pop-up appears for selecting a vault with the following options:

| Field | Operation |
|---|---|
| Vault | Select a vault from a list of created vaults. To create a vault, see Adding vaults. |
| Token | Select a vault from a list of created tokens. To create a token, see Adding tokens. |
| Parameter | Select a parameter from a list of created parameter in a token. For more information, see Adding tokens. |

17. After selecting the vault, token and parameter, click on **Proceed**. The value of the parameter will be encrypted.
18. Click Save & Next to see the execution summary and pipeline parameters.
19. Click Activate. The pipeline is now scheduled for execution. You can see the activated pipelines in the Pipeline Execution tab. For more information, see Executing a pipeline.

During the execution stage, pipelines are non-editable.

### Related topic(s)

- Editing pipelines
- Working with pipeline tasks
- Managing pipelines

# Creating a release pipeline

A release pipeline is used to deploy the final package generated from the integration pipeline into the live or production environment. It allows the Release Manager to select the package generated by the integration pipeline to deploy on the target environment, thereby significantly reducing the risk of code crashes and disruptions within the production or target environment.

Furthermore, these release pipelines implement security measures including maker-checker protocols and multi-factor authentication mechanisms. Through the integration of these security features, potential errors, mistakes, and unauthorized activities are proactively prevented from compromising the integrity of the target environment.

You can create a release pipeline within the Release project only. The Release project is the default repository for storing the release pipelines.

To create a release pipeline, perform the following steps:

The first essential task within a release pipeline is the Deployment Approval. This task is integral to all pre-defined templates and is highly recommended to include when creating a pipeline from scratch. This is because it allows you to select the integration pipeline data for deploying the artifacts into the live or production environment.

1. Go to the **Pipeline** tab. On the left pane, a list of existing projects appears. By default, a Release project appears.
2. In the Projects section, select the **Release** project. Here, you can create and store the release pipelines.

3. Click **Add a Pipeline**. The Choose a Release Pipeline Template section appears.
4. Click the **Create from Scratch** button to create a pipeline from scratch or choose any of the following templates using the **Select this template** button to create a pipeline:
    - **Release Pipeline** — Use this template to design a pipeline that fetches the artifacts from the source environment, creates and uploads the artifacts package with the given name in the selected artifactory, and then deploys those artifacts on the given environment or node.
    - **Release pipeline with manual approval** — Use this template to design a pipeline that gets the artifacts from the source environment, creates and uploads the artifacts package with the given name in the selected artifactory, and then deploys those artifacts on the given environment or node. It also includes a manual approval task for securing deployments to the production environment.
    - **Release pipeline with the email notification task** — Use this template to design a pipeline that gets the artifacts from the source environment, creates and uploads the artifacts package with the given name in the selected artifactory, and then deploys those artifacts on the given environment or node. It also includes a manual approval task for securing deployments to the production environment and a notification trigger task for sending email notifications to the release managers.

    The Add Pipeline dialog appears.
5. Enter the following information in the dialog:
    - **Pipeline Name** — Enter the name of the release pipeline.
    - (Optional) **Description** — Enter the description for the release pipeline.
6. Click **Add**. The Pipeline Editor page appears.
7. Click **+Add Task** to add a pre-defined task from the All, Build, Deploy, Approval, Conditional, or Jenkins categories under the required stage. The following tasks are available under the different categories::
    - Deployment Approval
    - Deployment
    - Post Deployment Configuration
    - Manual Approval
    - Notification Trigger
    - Conditional Task
    - Jenkins Task
    - Upload Package
    - Download Package

8. Once added, double-click the task to configure its related properties. In case you used a pre-defined template for creating the pipeline, then tasks are added by default. However, you can add and modify the tasks. For more information, see Working with pipeline tasks.

   Additionally, you can click the add icon + next to the stage label to insert another stage, and later add the required tasks to it. To define parameter and checklists to use in the pipeline, click the **Parameter** and **Checklist** icons on the page's upper-right section. For more information, see Creating a parameter and Creating a checklist.
9. Click **Save** to confirm the changes. The pipeline now appears in the Release Pipelines list. After creation, you can also edit a pipeline. For more information, see Editing a pipeline.
10. Click **Activate**. The pipeline is now ready for execution. You can see the activated pipelines in the Pipeline Execution tab. For more information, see Executing a pipeline.

> During the execution stage, pipelines are non-editable.

### Related topic(s)

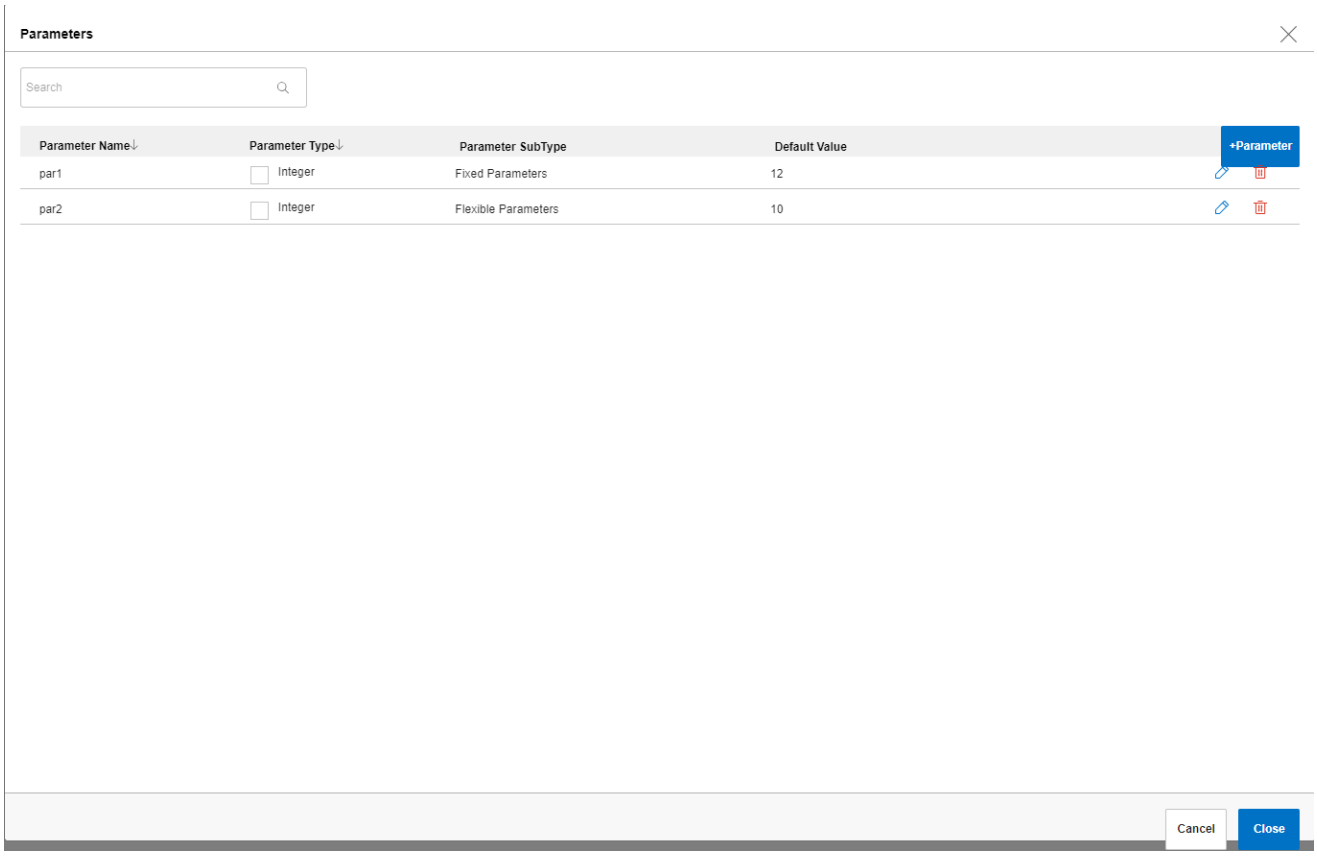
- Editing a pipeline
- Working with pipeline tasks
- Managing pipelines

# Creating a parameter

The Deployment Admin allows you to create and use parameters in a pipeline. You can create different types of parameters for specific data types.

To create a parameter, perform the following steps:

1. On the pipeline edit page, click the icon. The parameters dialog appears.
2. Click **+Parameter**. A new parameter row appears.
3. Enter the required details and select the required options in the following fields:




- **Parameter Name** — Enter the name of the parameter.
- **Type** — Select the data type for the parameter from this dropdown list. You can select from integer, text, and boolean types.
- **Parameter SubType** — Select the sub type of the parameter. It can be any of the following:
    - **Fixed Parameter** — It is a parameter that has a fixed value that does not change during run-time. A fixed parameter is only used while designing the pipeline. For example, you can use fixed parameters in conditional tasks.
    - **Flexible Parameter** — It is a parameter that can change or update its values during runtime. Such parameters are only used while designing the pipeline and their value remains dynamic throughout. For example, you can use dynamic parameters in conditional tasks.
    - **Runtime Parameter** — It is a parameter that can change or update its values during the pipeline execution. Such parameters are only used while executing the pipeline and their value remains dynamic throughout. For example, when a user executes a pipeline, he gets a

prompt to enter the value of the parameter based on which execution is further continued.

- **Default Value** — Enter a value that the parameter holds by default.

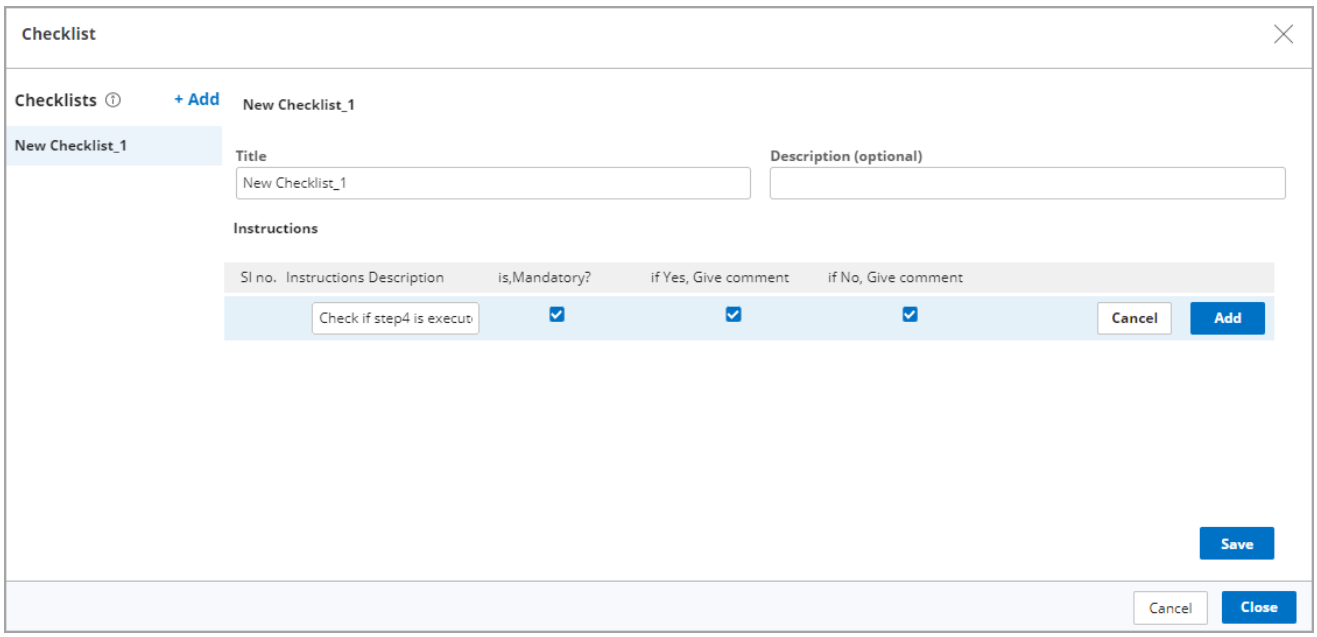
4. Click **Add**. The parameter is now added and available for use in the pipeline.
5. Repeat steps 3 to 5 to add multiple parameters.
6. Once done, click **Close**.

# Creating a checklist

The Deployment Admin allows you to create and use checklists in a pipeline where manual intervention is required during execution. While executing the pipeline, the user can follow the instructions available on the checklist to determine and perform the correct manual task.

To create a checklist, perform the following steps:

1. On the pipeline edit page, click the icon. The Checklist dialog appears.
2. Click **+Add**. A new checklist appears.
3. Enter the required details in the following fields:
   - **Title** — Enter the name of the checklist
   - **Description (optional)** — Enter a description of the checklist if required.
4. Click **+Instructions**. A new instruction row appears.
5. Enter the required details and select the required options in the following fields:
   - **Instructions Description** — Enter the name of the checklist
   - **is,Mandatory?** — Select if this instruction is mandatory to follow.
   - **if Yes, Give comment** — Selecting this requires the user to provide a comment if the instruction is true.
   - **if No, Give comment** — Selecting this requires the user to provide a comment if the instruction is false.
6. Click **Add** to save the instruction details.




7. Repeat steps 5 to 6 to add multiple instructions.
8. Once done, click **Save**.
9. Repeat steps 3 to 9 to add multiple checklists.
10. Once done, click **Close**. The checklists are now available in the pipeline.

# Editing a pipeline

The pipeline editor GUI allows you to build or edit a pipeline using the provided pre-defined tasks. To build or edit a pipeline, perform the following steps:

1. Open the required pipeline from the project. The pipeline editor page appears.
2. Follow steps 8 to 15 to make the required modification.
3. Once all the modifications are done, click **Save** on the pipeline title section to save the pipeline with the updated details.
4. Click **Activate** to enable the pipeline for execution.

Once activated, a pipeline appears on the *Pipeline Execution* page. At this stage the pipeline becomes non-editable.

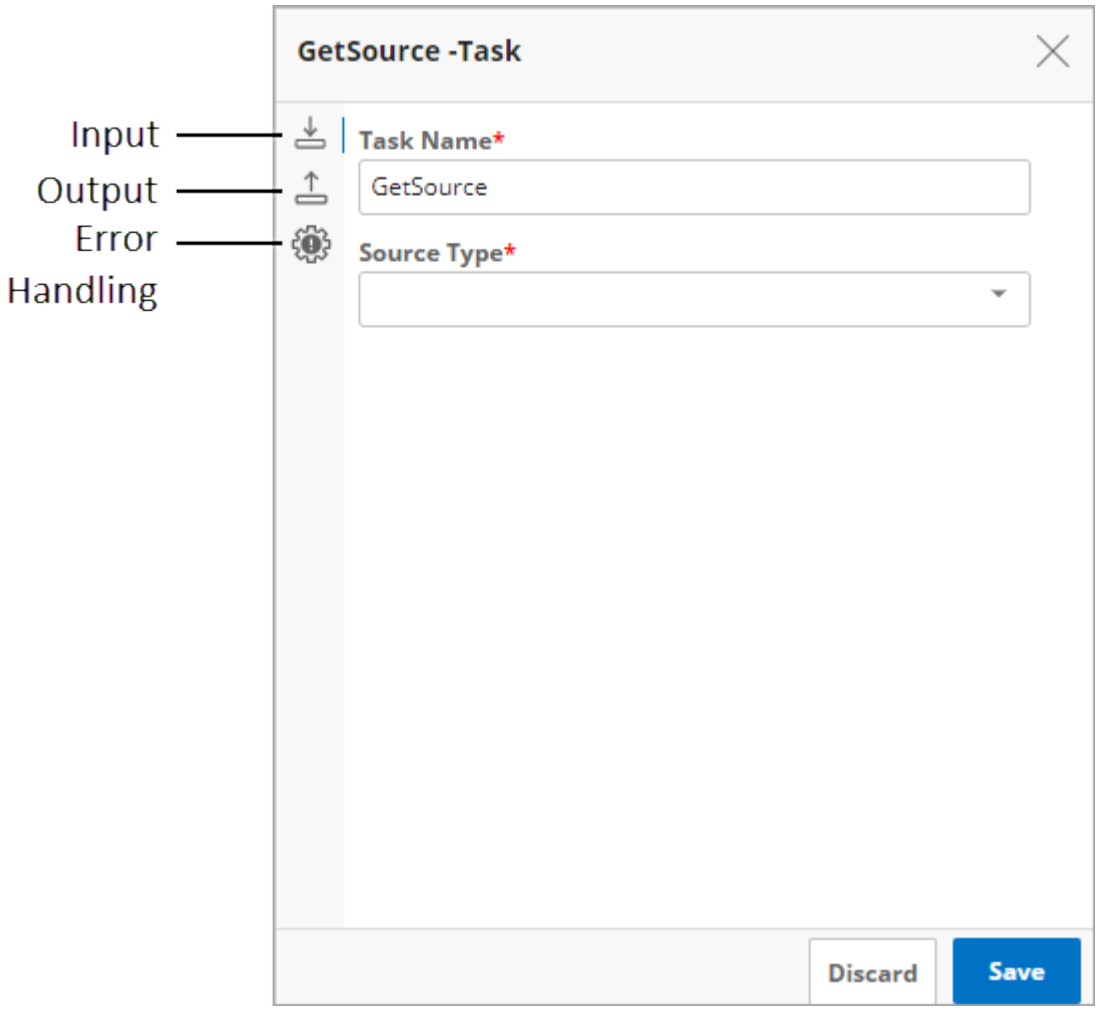
## Related topic(s)

1. Creating a pipeline

# Working with pipeline tasks

Tasks are pre-defined instruction sets that perform specific operations in a pipeline. The Deployment Admin allows you to configure the parameters of each task to fulfill your requirement.

Double-clicking each task opens the task parameter panel. The task parameter panel is comprised of three tabs. These are input parameters, output parameters, and error handling parameters. each tab displays the respective fields.




This topic describes the function and parameter definition of the following tasks:

- GetSource
- Create Package
- Deployment
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Conditional Task
- Jenkins Task
- Upload Package
- Download Package

# GetSource

The GetSource task fetches the source code of a portal, process, business rule, or business report from the defined location of the logged-in cabinet. The GetSource task provides you with the following parameters:

The GetSource task can only be accessed in integration pipelines.

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter the customized task name as per requirement. |
| Source Type | Select one of the following types of source from this dropdown list:<br>• Portal<br>• Process<br>• BRMS Rule<br>• Business Reports(BAM)<br>Parameters specific to each source type are listed below. |
| Portal Name | Select the portal name from this dropdown list if the source type is selected as Portal. |
| Process Name | Select the process name from this dropdown list if the source type is selected as Process. |

| Field | Operation |
|---|---|
| Process Version | Select the version of the selected process from this dropdown list if the source type is selected as Process. |
| Package Name | Select the package name from this dropdown list if the source type is selected as BRMS Rule. |
| Rule Name | Select the rule name from this dropdown list if the source type is selected as BRMS Rule. |
| Reports | Select the report name from this dropdown list if the source type is selected as Business Reports (BAM). |
| Custom Source Code | Select one of the options to deploy custom code:<br>• **SVN** — Select to deploy custom code from an SVN repository and enter the following details:<br>◦ **SVN Server** — Select the registered SVN server from this dropdown list.<br>◦ **SVN Repository Path** — Enter the repository path to deploy the custom code.<br>◦ **Custom Source Code Type** - Select the code type that is used.<br>• **GitHub** — Select to deploy custom code from a GitHub repository and enter the following details:<br>◦ **GitHub Server** — Select the registered GitHub server to deploy the custom code.<br>◦ **Custom Source Code Type** - Select the code type that is used.<br>• **GitLab** — Select to deploy custom code from a GitLab repository and enter the following details:<br>◦ **GitLab Server** — Select the registered GitLab server to deploy the custom code.<br>◦ **Custom Source Code Type** - Select the code type that is used.<br>• **No Custom Code** — Select if no custom code is required. |
| Portal Migration | Select from the following options:<br>• Yes - Select to allow portal migration even if data objects exist in the target environment.<br>• No - Select to not allow portal migration if data objects exist in the target environment. |

**Output parameters:**

| Field | Operation |
|---|---|
| Get Source Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

## Related topic(s)

- Create Package
- Deployment
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Conditional Task
- Jenkins Task

# Create Package

The Create Package task generates deployment artifacts using the source code fetched by the GetSource task, creates a package of the artifacts, and saves it in an artifactory server for deployment. The Create Package task provides you with the following parameters:

The Create Package task can only be accessed in integration pipelines.

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter the customized task name as per requirement. |
| Package Name | Enter the required name for the package. |
| Select Artifactory | Select an artifactory from this dropdown list to save the package. |

**Output parameters:**

| Field | Operation |
|---|---|
| PackagingOutput (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable**

option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.

- **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.

- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

---

### Related topic(s)

- Deployment
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Conditional Task
- Jenkins Task

---

# Deployment

The Deployment task deploys the created package from the artifactory server to the target environment. The Deployment task provides you with the following parameters:

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter the customized task name as per requirement. |
| Environment Name | Select the required target environment from this dropdown list. |
| Node Name | Select the required node in the selected environment from this dropdown list. |
| SMS Server Details - Enter the following to connect to an SMS server. | |
| Secure | Select this checkbox secure connection. |
| IP DomainName | Enter the IP domain name of the SMS server. |

| Field | Operation |
|---|---|
| Server Type | Select the server type from this dropdown list. |
| Port | Enter the port number. |
| Connect | Click to connect to the SMS server. A popup message confirms successful connection. |
| Cabinet Name | Select the required cabinet on the connected SMS server from this dropdown list. |
| Volume Name | Select the required volume available in the selected cabinet from this dropdown list. |
| Site Name | This field is auto-generated with the site name defined in the selected volume. |
| MDM Server Details - Enter the following to connect to the MDM server. | |
| Same as sms server details | Select this checkbox if the MDM server is running on the same environment as the defined SMS server. |
| Secure | Select this checkbox to secure connection. |
| IP DomainName | Enter the IP domain name of the MDM server. |
| Server Type | Select the server type from this dropdown list. |
| Port | Enter the port number. |
| Connect | Click to connect to the MDM server. A popup message confirms successful connection. |
| Cabinet Name | Select the required cabinet on the connected MDM server from this dropdown list. |
| Micro Service Enabled | Select this option if you want to enable OD microservices with the deployment. |
| Gateway URI | Enter the gateway URI of the micro UI. |
| OD Authentication URI | Enter the URI for OD authentication. |
| OD Image Service URI | Enter the URI for OD image service. |

**Output parameters:**

| Field | Operation |
|---|---|
| Deployment Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

## Related topic(s)

- Create Package
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Conditional Task
- Jenkins Task

# Post Deployment Configuration

While deploying a process, the supported services of the same must also be deployed in the target environment. The Post Deployment Configuration task allows you to configure the services that are necessary for the deployment. The Post Deployment Configuration task provides you with the following parameters:

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter the customized task name as per requirement. |
| Services | Select the service to deploy from this dropdown list:<br>• **Criteria Management** — Deploys criteria from the source environment to the target environment.<br>• **Process queue user group mapping** — Deploys the user group associated with the selected process. This helps in managing rights as users added to the imported user group get the applicable rights.<br>• **Advanced Search Configuration** — Deploys all advanced search configurations from the source environment to the target environment.<br>• **Audit log configuration** — Deploys all audit log configurations from the source environment to the target environment.<br>• **Process Specific Services** — Deploys services specific to the selected process. |
| Criteria Name | Enter the name of the criteria to deploy if Criteria Management is selected in the Services field. |
| Process | Enter the name of the process to deploy the user groups associated with the process queue if Process queue user group mapping is selected in the Services field. |
| Service Type | Select the required service type for deployment if Process Specific Services is selected in the Services field:<br>• Process Server<br>• Mailing Agent<br>• Message Agent<br>• Export Utility<br>• File Upload Utility<br>• Initiation Agent |
| Service Name | This field lists all the available services of the selected service type. Select the required service from this dropdown list. |

| Field | Operation |
|---|---|
| Mailing Utility Configuration — Enter these fields for mailing utility-type services. | |
| Web Service Address | Specify the Web Server address. You may also specify the port if required. |
| Mailing Agent Configuration — Enter these fields for mailing agent type services. | |
| Mail Server Address | Specify the IP address of the mail server to send e-mails. |
| Mail Read Receipt | Select Yes to send an acknowledgment mail to the recipient. |
| PortNo | Specify the port of the above-mentioned mail server. |
| User Email Id | Enter the email ID to use for sending e-mails. |
| Account Name | Enter the email account name. |
| Account Password | Enter the email account password. |
| Security Mode | Select the required security mode. The options are TLS, SSL, and NONE. |
| Message Agent Configuration — Enter these fields for messaging agent type services. | |
| Object Count | Specify the number of messages to be locked at a time by the Message Agent.<br>The optimal value is 50 |
| Export Utility Configuration — Enter these fields for export utility type services. | |
| Export Location | Enter the location to save the CSV file. |
| File Upload Utility Configuration — Enter these fields for file upload utility type services. | |
| Upload Directory | Enter the location of the files that need to be uploaded. |
| File Name | Enter the name of the file to upload from the upload directory. |
| Initiaition Agent Configuration — Enter these fields for initiation agent type services. | |
| Mail Protocol | Enter the email protocol to use. For example, SMTP, POP, and more. |
| Mail Password | Enter the password of the account. |
| Import Path | Enter the path to store the fetched emails temporarily. |
| Destination Folder | Enter the destination path to store the processed document.<br>After the processing, if the operation is successful, the document gets stored in the Successful folder. Else, it gets stored in the Failure folder. |
| Source Folder | Enter the path of the Mail Server to fetch the emails. |

| Field | Operation |
|---|---|
| Trigger Failure Notification To | Enter the email IDs to send notification in case of a failure. |
| Target Environment Configuration — Enter these fields for the target environment. | |
| Environment Name | Select the required target environment from this dropdown list. |
| Node Name | Select the required node in the selected environment from this dropdown list. |
| Cabinet Configuration — Enter these fields for any type of services. | |
| Get Cabinets | Click **Get Cabinets** to fetch all the available cabinets in the selected environment. |
| Cabinet Name | Select the required target cabinet from this dropdown list. |
| Username | Enter the username for the selected service. You can also select a parameter from a vault using the vault icon next to this field. |
| Password | Enter the password for the provided username. You can also select a parameter from a vault using the vault icon next to this field. |

**Output parameters:**

| Field | Operation |
|---|---|
| Post Deployment Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable**

option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.

- **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.

- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

### Related topic(s)

- Deployment
- Manual Approval
- Notification Trigger
- Conditional Task
- Jenkins Task

# Manual Approval

The Manual Approval task stops the pipeline execution for manual approval. This task sends a notification to the selected users. It can also present the users with a checklist to follow before giving approval. The Manual Approval task provides you with the following parameters:

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter the customized task name as per requirement. |
| Notify To | Enter the email address of the user you want to notify for manual approval. You can also add multiple email addresses using commas as separators. |
| Checklists | This dropdown list displays all the available checklists for this pipeline. Select one or more checklists from this dropdown list for the user to follow before granting approval. To know more about checklists and how to create one, check Creating a checklist. |

**Output parameters:**

| Field | Operation |
|---|---|
| Manual Approval Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

## Related topic(s)

- Create Package
- Deployment
- Post Deployment Configuration
- Notification Trigger
- Conditional Task
- Jenkins Task

# Notification Trigger

The Notification Trigger task sends an email to the selected users with the pre-defined mail subject and body. The Notification Trigger task provides you with the following parameters:

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter the customized task name as per requirement. |
| Notify To | Enter the email address of the user you want to notify. You can also add multiple email addresses using commas as separators. |
| Email Subject | Enter the required email subject for the notification. |
| Email Body | Enter the required email body for the notification. |

**Output parameters:**

| Field | Operation |
|---|---|
| Email Notification Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.

- **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.

- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

### Related topic(s)

- Deployment
- Post Deployment Configuration
- Manual Approval
- Conditional Task
- Jenkins Task

# Conditional Task

Conditional Task allows you to execute another task upon satisfying a pre-defined criterion. Conditional Task provides you with the following parameters:

**Input parameters:**

| Field | Operation |
|---|---|
| Variable(Text) | Select the required variable to create the If condition. |
| Operator | Select the required operator from this dropdown list. |
| Value | Enter the expected value of the selected variable. |
| Add More^ | Click **Add More** and then select the required function to add another condition. |

**Output parameters:**

| Field | Operation |
|---|---|
| Conditional Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

## Related topic(s)

- Create Package
- Deployment
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Jenkins Task

# Jenkins Task

Jenkins Task allows you to execute a job from a registered Jenkins server. Jenkins Task provides you with the following parameters:

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter Name for the task. |
| Jenkins Server | Select the registered Jenkins server from this dropdown list. |
| Folder Name | This dropdown list displays all the folders available on the selected Jenkins server. Select the required folder. |
| Job Name | This dropdown list displays all the available jobs in the selected folder. Select the required job to invoke. |

**Output parameters:**

| Field | Operation |
|---|---|
| Conditional Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.

- **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.

- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

### Related topic(s)

- Deployment
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Deployment Approval

# Upload package

Upload package allows you to upload your package to a JFrog or FTP server.

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter Name of the task. |
| Package Name | Enter the package name |
| Server Type | Select any one server type from the following to upload the package<br>• FTP<br>• JFROG<br>Parameters specific to each source type are listed below. |
| FTP server | This drop down list displays all the FTP servers that is configured in the repository under server/agents configuration. |
| FTP Directory | This is the path, where the package will be stored. |
| Notify to | Entered email id(s) will receive the notification with the upload package details and the key for downloading it. |
| Select Artifactory | This drop down list displays all the JFROG servers that is configured in the artifactory under server/agents configuration. |

| Field | Operation |
|---|---|
| Notify to | Entered email id(s) will receive the notification with the package details and the key for accessing it. |

**Output parameters:**

| Field | Operation |
|---|---|
| Upload Package Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

# Download package

Download package allows you to download your package from a JFrog or FTP server.

**Input parameters:**

| Field | Operation |
|---|---|
| Task Name | Enter Name of the task. |
| Server Type | Select any one server type to download the package from<br>• FTP<br>• JFrog<br>Parameters specific to each source type are listed below. |
| FTP server | This drop down list displays all the FTP servers that is configured in the repository under server/agents configuration. Select the one you want to download from. |
| FTP Package | Select the package you want to download from the selected server. |
| Key | Enter the key to the selected package. |
| Source Type | This is the artifact type which will be populataed on the basis of the package you have selected. |
| Select Artifactory | This drop down list displays all the artifactories that is configured in the artifacts under server/agents configuration. Select the one you want to download from. |
| Select Artifact | Select the artifact you want to download from the selected artifactory. |
| Key | Enter the key to the selected artifact. |
| Source Type | This is the artifact type which will be populataed on the basis of the package you have selected. |

**Output parameters:**

| Field | Operation |
|---|---|
| Download Package Output (only Boolean) | Select a boolean type variable in this field. When the task executes successfully, this variable is set to 1 otherwise 0. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

# Deployment Approval

The initial task within any release pipeline is Deployment Approval. This task is integral to all pre-defined templates and is highly recommended to include when creating a pipeline from scratch. The main purpose of this task is to facilitate the selection of the integration pipeline package in the release pipeline. While executing the release pipeline, you have the option to select the necessary integration pipeline and its associated package that you want to deploy into the target environment.

In addition, the Deployment Approval task allows you to enable multi-factor authentication to add an additional layer of security. Using this feature, the Release Manager enters the authentication code received on the registered email address

before rejecting or approving the package request. The Deployment Approval task provides you with the following parameters:

The Deployment Approval task can only be accessed in release pipelines.

**Input parameters:**

| Field | Description |
|---|---|
| Task Name | Enter the name of the task. |
| Select Integration Pipeline | From the dropdown list, select the required integration pipeline and its associated package for the release pipeline. This list only shows the successfully executed integration pipelines. |
| Pipeline Type (Read Only) | This field is automatically prefilled with value based on the chosen integration pipeline. The type of pipeline can be a portal, process, BRMS rule, or business report (BAM). |
| Enable Multi-Factor Authentication | Allows you to control the multi-factor authentication functionality in the Deployment Approval task.<br>• Enable the toggle to activate the multi-factor authentication. By enabling it, the Release Manager is prompted to enter the authentication code, received on his registered email address, before accepting or rejecting the generated request for package approval.<br>• Disable the toggle to deactivate the multi-factor authentication. |
| Allow Add Comments | Allows you to add comments while generating the request for package approval.<br>• Enable the toggle to allow users to input the comments when generating the package approval request. By enabling the toggle, it becomes necessary for the users to add a comment.<br>• Disable the toggle to restrict the users from adding comments when generating the package approval request. |

**Error handling parameters:**

Here, you can specify the type of error-handling action you want the system to perform when an error occurs during the pipeline execution:

- **Handle Error** — Select this option to specify the type of action you want the system to perform if an error occurs during the pipeline execution. Click **+Actions** and then select the required error-handling action:
  - **Retry** — Select this action to retry the execution of the GetSource task upon failure and specify the retry limit for it. The minimum and maximum limit for retry is 0 and 3, respectively. On selection, a Retry No. dropdown appears from where you can select the required retry limit. Based on the chosen retry limit, the system tries to execute the GetSource task. You can also assign a value to a variable by clicking **+Actions** and then selecting the **Set Variable** option. For more details, see Set Variable. Upon exceeding the specified retry count, the pipeline execution gets terminated and the specified value gets assigned to the selected variable.
  - **Set Variable** — Select this action to assign a value to a variable if an error occurs during pipeline execution. Two fields namely, Set Variable and As appear. Select the required variable from the Set Variable dropdown list, and then enter the required value in the As field. You can add multiple variables here.
- **Continue & Exit** — Select this option to continue and exit the pipeline execution if an error occurs.

## Related topic(s)

- Deployment
- Post Deployment Configuration
- Manual Approval
- Notification Trigger
- Conditional Task
- Jenkins Task

# Managing pipelines

The Pipelines Designs page provides you with various options to edit, activate, deactivate, or delete available pipelines.

To edit a pipeline, perform the following steps:

1. Open the specific project.
2. In the pipeline list area click the pipeline name. You can also click the options icon ⋮ on the required pipeline and then select Edit. The pipeline now opens in the pipeline editor.
3. Follow the instructions provided in Editing pipelines to modify the pipeline as required.
4. Click **Save** on the pipeline title section to save the pipeline with the updated details.

To activate or deactivate a pipeline, perform the following steps:

1. Open the specific project.
2. In the pipeline list area click **Activate** or **Deactivate** on the required pipeline to activate or deactivate the pipeline. The Activate Pipeline or Deactivate Pipeline dialog appears based on your selection.
3. Click **Activate** on the Activate Pipeline dialog to confirm the activation of the pipeline. The pipeline is now activated.
   Click **Deactivate** on the Deactivate Pipeline dialog to confirm the deactivation of the pipeline. The pipeline is now deactivated.

> Only the activated pipelines become available for execution.

To delete a pipeline, perform the following steps:

1. Open the specific project.
2. In the pipeline list area click the options icon ⋮ on the required pipeline and then select **Delete**. The Delete Pipeline dialog appears.
3. Click **Delete**.

# Managing projects

The Pipelines Designs page provides you with various options to edit or delete available projects.

To edit a projects, perform the following steps:

1. Open the specific project.
2. Click the edit icon next to the project name. The Edit Project dialog appears.
3. Enter the new Project Name and Description as required.
4. Click **Edit**. The changes are now saved.

To delete a project, perform the following steps:

1. Open the specific project.
2. In the projects title area click the delete icon. The Delete Project dialog appears.
3. Click **Delete**. The project is now deleted.

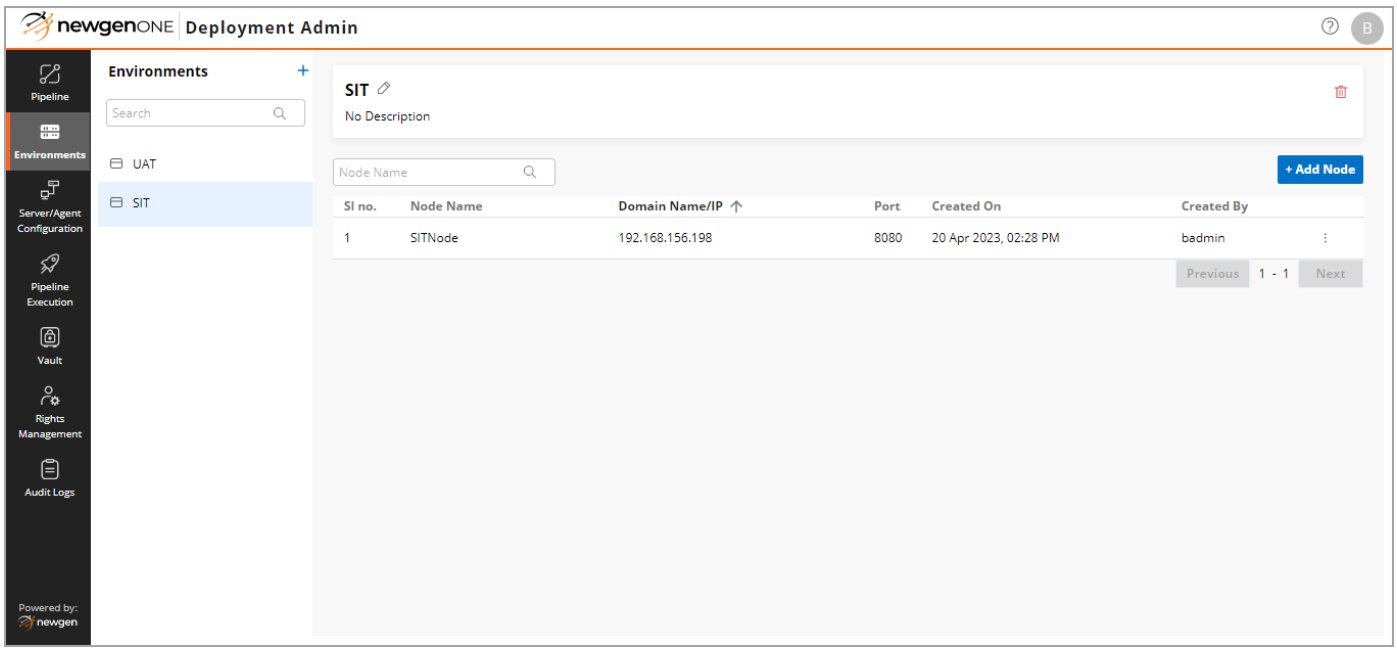
## Related topic(s)

- Creating a project
- Creating a pipeline
- Working with pipeline tasks

# Adding and managing environments

The Environments menu allows you to add, edit, and delete environments and nodes to deploy artifacts. The Environments page of the Deployment Admin displays the following information and options:

| Field | Operation |
|---|---|
| Search environment | The search box provides the option to search an environment by name. |
| Environment list | Shows the environments added by all users in the same cabinet. |
| Add environment | This icon allows you to add an environment. |
| Edit environment | Clicking the edit icon next to the environment name in the title section allows you to edit the environment. To know more, see Managing environments. |
| Delete environment | Clicking the delete icon next to the environment name in the title section allows you to delete the environment. To know more, see Managing environments. |
| Search node | The search box provides the option to search a node in the selected environment by name. |
| Add Node | This button allows you to add a new node in the selected environment. |
| Node menu | Clicking the node menu icon ⋮ provides you with the following options:<br>• **Edit** - Selecting this option opens the Edit Node dialog to edit the node. To know more visit Managing nodes.<br>• **Delete** - Selecting this option allows you to delete the node. To know more, see Managing environments. |
| Node list | Displays the list of nodes in the selected environment. |




This chapter describes:

- Adding an environment
- Managing environments
- Adding a node
- Managing nodes

# Adding an environment

Deployment Admin allows you to add environments to host one or more nodes in order to deploy an artifact. To add an environment, perform the following steps:

1. Navigate to the Environments page and then click the create new environment icon + on the Environments section. The New Environment dialog appears.
2. On the New Environment dialog, enter the Environment Name and a Description of the environment in their respective fields.

   Description is an optional field. Adding a description to an environment can help in understanding more about the environment.

3. Click **Add Environment**. The environment now appears in the Environments section and a popup message appears to confirm the creation of the environment .

### Related topic(s)

- Managing environments
- Adding a node
- Managing nodes

# Managing environments

The Environments page provides you with various options to edit or delete available environments.

To edit an environment, perform the following steps:

1. Go to the Environments menu and select the required environment in the Environments section. The environment details appear on the right.
2. Click the edit icon next to the environment name in the title section. The Update Environment dialog appears.
3. Make necessary changes to the Update Environment dialog.
4. Click **Update Environment**. The environment is now updated with the modified information.

To delete an environment, perform the following steps:

1. Go to the Environments menu and select the required environment in the Environments section. The environment details appear on the right.
2. Click the delete icon next to the environment name in the title section. The Delete Environment dialog appears.
3. Click **Delete**. The environment is now deleted from the Environments section.

### Related topic(s)

- Adding an environment
- Adding a node
- Managing nodes

# Adding a node

A node is an application server configured with NewgenONE. Nodes are added to the Deployment Admin to deploy artifacts on them. To add a node to an environment in Deployment Admin, perform the following steps:

1. Open the Environments menu.
2. Select the required environment from the Environments pane. The environment details appear on the right with the existing node list.
3. Click **Add Node**. The Add Node dialog appears.
4. Enter the following fields in the Add Node dialog:

| Field | Description |
|---|---|
| Node Name | Enter the name of the node. |
| Secure (http enabled) | Select this checkbox if the application server uses a secure connection. |
| Docker Environment | Select this checkbox to associate the addition of node available on Docker. On selecting, the **App Server Name/IP (EJB)** field appears. In this field, enter the name or IP of the application server. |
| Machine Host Name | Select this option to add a node using the hostname of the target environment. On selecting, the following fields appear:<br>• **Machine Host Name** — Enter the hostname of the target environment where the node is available.<br>• **Server Type** — Select the type of application server of the node. JBoss EAP, WebLogic, and WebSphere are the different types of application server types supported.<br>• **Communication Port** — Enter the port number of the environment where the node is available. |

| Field | Description |
|---|---|
| Domain Name | Select this option to add a node using the domain name of the target environment. On selecting, the following fields appear:<br>• **Domain Name** — Enter the name of the domain name of the target environment where the node is available.<br>• **Server Type** — Select the type of application server of the node. JBoss EAP, WebLogic, and WebSphere are the different types of application servers supported. |
| Deployment through DM Agent | Select this option to establish a connection between the source and target environment using the DM Agent. For more information, see Configuring deployment agent. On selecting this option, the following field appear:<br>• **Server Home Directory** — Enter the home directory path on the target environment where you want to transfer the artifacts using the DM agent. |

| Field | Description |
|---|---|
| Deployment through SSH Protocol | Select this option to establish a secure connection between the source and target environment using the Secure Socket Shell (SSH) protocol. On selecting, the following fields appear:<br>• **Deployment Directory** — Specify the home directory path on the target environment where you want to transfer the artifacts using the SSH protocol.<br>• **SSH Port** — Enter the port number on which the SSH server runs in the target environment.<br>Ensure that the SSH service is enabled in the target environment.<br>• **Username** — Enter the SSH username. To use a token parameter as the username value for security purposes, click the corresponding vault icon and then click the ellipsis icon ⋯. The Selecting Vault dialog appears where you can enter the vault, token, and parameter to use as the SSH username. For more information, see Adding a vault.<br>• **Password** — Enter the SSH password associated with the username. To use a token parameter as the password value for security purposes, click the corresponding vault icon and then click the ellipsis icon ⋯. The Selecting Vault dialog appears where you can enter the vault, token, and parameter to use as the SSH password. For more information, see Adding a vault.<br>Currently, the **Deployment through SSH Protocol** option is only applicable for deploying portals on the target environment. |
| Description (Optional) | Enter additional information about the addition of a node. |

5. Click **Add**. The node is now added to the environment list and a popup message confirms the same.

### Related topic(s)

- Managing environments
- Managing nodes
- Adding an environment

# Managing nodes

The Environments page provides you with various options to edit or delete available nodes in an environment.

To edit a node, perform the following steps:

1. Go to the Environments menu and select the required environment in the Environments section. The environment details appear on the right.
2. Click the options icon ⋮ next to the required node and then select Edit. The Edit Node page appears.
3. Make necessary changes in the Edit Node dialog.
4. Click **Edit**. The node is now updated with the modified information.

To delete a node, perform the following steps:

1. Go to the Environments menu and select the required environment in the Environments section. The environment details appear on the right.
2. Click the options icon ⋮ next to the required node and then select **Delete**. The Delete Node dialog appears.
3. Click **Delete**. The environment is now deleted from the Environments section.

### Related topic(s)

- Managing environments
- Adding nodes
- Managing nodes

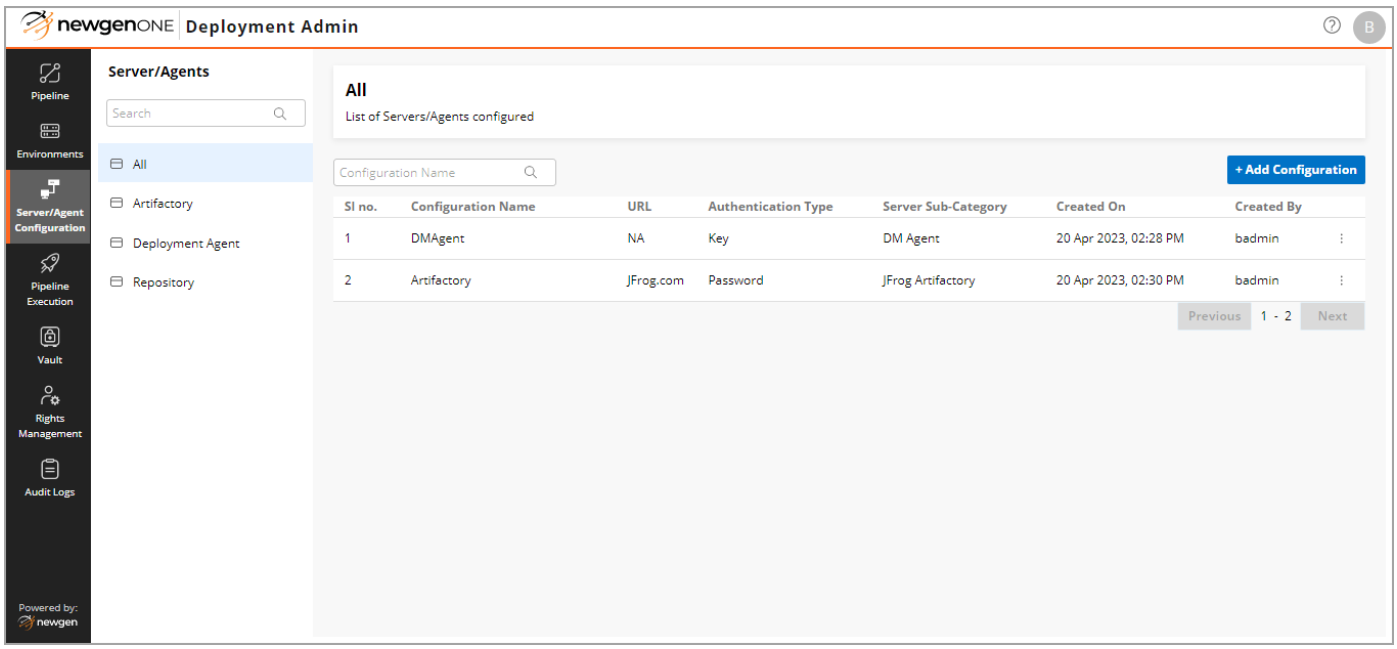
# Managing server-agent configuration

The Server/Agent Configuration menu allows you to configure external servers for different services. The Deployment Admin allows you to configure several pre-defined external servers and agent types. The following topics describe the procedure for registering each type of external server and agent:

- Configuring services
- Configuring artifactory
- Configuring deployment agent
- Configuring repository
- Configuring Jenkins server

The Server/Agent Configuration page of the Deployment Admin displays the following information and options:

| Field | Operation |
|---|---|
| Search Server/Agents Categories | The search box provides the option to search a server/agents category by name. |
| Server/Agents Categories list | Shows the permitted server/agents categories. |
| Search Configuration | The search box provides the option to search a configuration in the selected server/agents category by name. |
| Add Configuration | This icon allows you to add an external server configuration. |
| Configuration menu | Clicking the configuration menu icon ⋮ provides you with the following options:<br>• **Edit** — Selecting this option opens the configuration in edit mode.<br>• **Delete** — Selecting this option allows you to delete the configuration. |
| Configuration list | Displays the list of configurations in the selected server/agents category. |




# Configuring services

The Deployment Admin allows you to configure servers that provide specific services.

To configure a server, perform the following steps:

1. Navigate to the Server/Agent Configuration menu and then select **All**. The details of all the registered servers and agents appear on the right.
2. Click **+ Add Configuration**. The New Configuration dialog appears.
3. On the New Configuration dialog fill in the following fields:

| Field | Operation |
|---|---|
| Configuration Name | Enter a name for the configuration. |
| Server Category | Select the required category of the server to configure. The available options are Artifactory, Deployment Agent, Repository, and Jenkins. |

| Field | Operation |
|---|---|
| Server Sub-Category | This dropdown list provides you with the available sub-categories for the selected server category.<br>Select a sub-category for the artifactory server category:<br>• JFrog Artifactory<br>• OmniDocs<br>Select a sub-category for the deployment agent server category:<br>• DM Agent<br>Select a sub-category for the repository server category:<br>• GitHub<br>• GitLab<br>• SVN<br>• FTP Server<br>Select a sub-category for the Jenkins server category:<br>• Jenkins Server |
| URL | Enter the URL of the server. |
| Authentication type | Select an authentication type. |
| Username | For Password type authentication, enter the Username for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a username. |
| Password | For Password type authentication, enter the Password for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a password. |
| Token | For Token type authentication, enter the token for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a token. |
| Key | For Key type authentication, enter the key for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a key. |
| Test Connection | Click to check the connection status with the server. |

4. Click **Add**. The server gets configured.

### Related topic(s)

- Configuring artifactory
- Configuring deployment agent
- Configuring repository
- Configuring Jenkins server

# Configuring artifactory

Artifactory servers can store the artifacts that the Deployment Admin generates and perform versioning automatically. To configure an Artifactory server, perform the following steps:

1. Navigate to the Server/Agent Configuration menu and then select Artifactory. The details of the Artifactory category appear on the right.
2. Click **Add Configuration**. The New Configuration dialog appears.
3. On the New Configuration dialog fill in the following fields:

| Field | Operation |
|---|---|
| Configuration Name | Enter a name for the configuration. |
| Server Category | This field is non-editable with Artifactory as the pre-selected option. |
| Server Sub-Category | Select one of the following sub-categories from this dropdown list:<br>• JFrog Artifactory<br>• OmniDocs |
| URL | Enter the URL of the server. |
| Authentication type | Select an authentication type. |
| Username | For Password type authentication, enter the Username for authentication. You can also click the vault icon next to this field to use a parameter from a vault as an username. |
| Password | For Password type authentication, enter the Password for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a password. |

| Field | Operation |
|---|---|
| Token | For Token type authentication, enter the token for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a token. |
| Key | For Key type authentication, enter the key for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a key. |
| Repository Name | For JFrog Artifactory server, enter the name of the required repository. |
| Folder Name | For OmniDocs Artifactory server, enter the name of the required folder. |
| Test Connection | Click to check the connection status with the artifactory server. |

4. Click **Add**. The artifactory server gets configured.

### Related topic(s)

- Configuring artifactory
- Configuring deployment agent
- Configuring repository
- Configuring Jenkins server

# Configuring deployment agent

Deployment agents are used for secure deployment communication between the Deployment Admin and the target environment. To configure a deployment agent, perform the following steps:

1. Navigate to the Server/Agent Configuration menu and then select Deployment Agent. The details of the Deployment Agent category appear on the right.
2. Click **Add Configuration**. The New Configuration dialog appears.
3. On the New Configuration dialog fill in the following fields:

| Field | Operation |
|---|---|
| Configuration Name | Enter a name for the configuration. |
| Server Category | This field is non-editable with Deployment Agent as the pre-selected option. |
| Server Sub-Category | Select DM Agent from this dropdown list. |
| Authentication type | A key appears upon selecting the Server Sub-Category. You can copy the key using the copy icon adjacent to the key. |
| Select Environment | Select the environment name from this dropdown list. |
| Select Node | Select the required node in the selected environment from this dropdown list. |

4. Click **View Agent**. The command to run the deployment agent on the target environment appears. You can copy the command using the copy icon adjacent to the command. To know more, see Starting deployment agent on target environment.
5. Click **Add**. The configuration is now added.

## Related topic(s)

- Configuring artifactory
- Configuring deployment agent
- Configuring repository
- Configuring Jenkins server

# Starting deployment agent on target environment

To deploy any artifact to a target environment, you must run the deployment agent on the target environment first.

**Prerequisites**

- You must have sufficient rights to access the install directory of the Deployment Admin.
- You must have the command generated while configuring the deployment agent in the Deployment Admin.

To run the deployment agent on the target environment, perform the following steps:

1. Open the **deployment-agent** folder in the Deployment Admin install directory.
2. Copy the **dmagent.jar** to the target environment and then launch the command prompt from the same location.
3. Run the command generated while configuring the deployment agent in the Deployment Admin. On successful execution of the command, the command prompt window displays "Running".

   > You must not close the command prompt.

While deploying artifacts on multiple target environments, you must run the deployment agent on all the target environments.

# Configuring repository

Repository services are used to store artifacts that the Deployment Admin generates. Deployment Admin also allows you to check out stored artifacts from the repository. To configure a repository server, perform the following steps:

1. Navigate to the **Server/Agent Configuration** tab and select **Repository**. The details of the Repository category appear on the right.
2. Click **Add Configuration**. The New Configuration dialog appears.

3. On the New Configuration dialog fill in the following fields:

| Field | Operation |
|---|---|
| Configuration Name | Enter a name for the configuration. |
| Server Category | This field is non-editable with Repository as the pre-selected option. |
| Server Sub-Category | Select one of the following sub-categories from this dropdown list:<br>• GitHub<br>• GitLab<br>• SVN<br>• FTP Server |
| URL | Enter the URL of the server. |
| Authentication type | Select an authentication type. |
| Username | For Password type authentication, enter the Username for authentication. You can also click the vault icon next to this field to use a parameter from a vault as an username. |
| Password | For Password type authentication, enter the Password for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a password. |
| FTP Server Address | Enter the FTP Server Address here, such as `ftp.newgen.co.in`, to establish a secure connection for file transfers |
| FTP Server Port | Enter the FTP Server Port here, to establish a secure connection for file transfers |
| FTP Directory | Enter the FTP Directory here, to specify the folder for file transfers. |
| Token | For Token type authentication, enter the token for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a token. |
| Branch Name | Enter the branch name of the GitHub or GitLab server. |
| Test Connection | Click to check the connection status with the repository. |

4. Click **Add**. The repository gets configured.

## Related topic(s)

# Configuring Jenkins server

Deployment Admin allows you to register Jenkins servers. You can register a Jenkins server and then execute a job from the registered Jenkins server using the Jenkins Task in your pipeline. To configure a Jenkins server, perform the following steps:

1. Navigate to the **Server/Agent Configuration** tab and then select **Jenkins**. The details of the Jenkins category appear on the right.
2. Click **+ Add Configuration**. The New Configuration dialog appears.
3. On the New Configuration dialog fill in the following fields:

| Field | Operation |
|---|---|
| Configuration Name | Enter a name for the configuration. |
| Server Category | This field is non-editable with Jenkins as the pre-selected option. |
| Server Sub-Category | Select Jenkins Server from this dropdown list. |
| Jenkins URL | Enter the URL of the Jenkins server. |
| Authentication type | Select an authentication type. |
| Username | For Password type authentication, enter the Username for authentication. You can also click the vault icon next to this field to use a parameter from a vault as an username. |
| Password | For Password type authentication, enter the Password for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a password. |
| Token | For Token type authentication, enter the token for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a token. |
| Key | For Key type authentication, enter the key for authentication. You can also click the vault icon next to this field to use a parameter from a vault as a key. |

| Field | Operation |
| --- | --- |
| Test Connection | Click to check the connection status with the Jenkins server. |

4. Click **Add**. The Jenkins server gets configured.

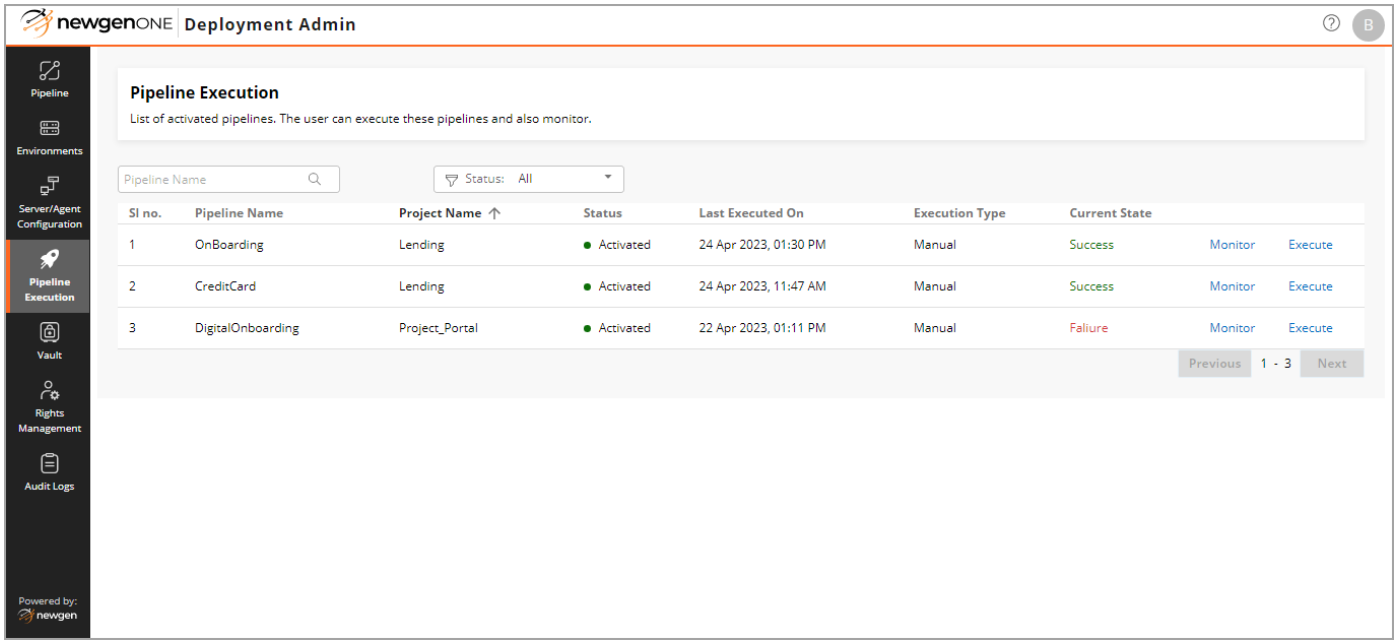
## Related topic(s)

- Configuring artifactory
- Configuring deployment agent
- Configuring repository
- Configuring Jenkins server

# Executing and monitoring pipeline

The Pipeline Execution page displays all the activated and deactivated pipelines and allows you to execute and monitor the activated pipelines. The Pipeline Execution page also provides you with the history of pipeline execution. The Pipeline Execution page of the Deployment Admin displays the following information and options:

| Field | Operation |
|---|---|
| Search pipeline | The search box provides the option to search an environment by name. |
| Filter pipeline | Allows you to filter the pipelines by the activation status. |
| Pipeline list | Displays the list of activated and deactivated pipelines. |
| Execute | This option allows you to execute an activated pipeline. |
| Monitor | This option allows you to monitor the execution stages of an executed pipeline. |
| History | This option allows you to check the history of an activated pipeline. |



# Executing and monitoring an integration pipeline

The Pipeline Execution tab allows you to execute activated pipelines. You can also monitor the executed or in-execution pipelines to check the progress of each stage (task) in the pipeline. While monitoring, you also get an option to rollback. The primary function of rollback is to revert to the last successful deployment in the event of unexpected issues or failures in a new deployment. This ensures that the system can quickly return to a stable and known state, minimizing downtime and reducing the impact of problems that may arise during the deployment process.

You can view the history to check the attempts and failure details of the deactivated pipelines.

To execute a pipeline, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **Execute** next to the required pipeline. The Execute Confirmation dialog appears.
3. Click **Execute** to confirm. The Pipeline Monitoring page appears displaying the pipeline execution numbers and the progress state of each stage in the pipeline. Upon successful execution, the status of the pipeline changes to Success.

To monitor a pipeline, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **Monitor** next to the required pipeline. The Pipeline Monitoring page appears displaying the pipeline execution numbers and the progress state of each stage in the pipeline. Apart from this, you can also see the pipeline execution date and executioner details.

To rollback to the last successful pipeline execution, perform the following steps:

> The Rollback option is only applicable for portals.

1. Go to the **Pipeline Execution** tab.
2. Click **Monitor** next to the required release pipeline. A new page appears displaying the pipeline execution numbers and the progress state of each stage in the release pipeline.
3. Click the **Rollback** link. This link only appears next to the last successful pipeline execution.
   A Rollback Confirmation dialog appears.
4. Click **Rollback** to confirm.
   It reverts all the current execution changes to the previous successful execution changes.

To view the history of a pipeline, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **History** next to the required pipeline. The Pipeline Monitoring page appears displaying the pipeline execution numbers and the progress state of each stage in the pipeline. Apart from this, you can also see the pipeline execution date and executioner details.
   Here, you can click the required link under the failed stage to view possible errors.

# Executing and monitoring a release pipeline

When you execute a release pipeline, you can select the required package of the integration pipeline to deploy into the live or production environment. Once done, a package approval request is generated and sent to the Release Manager.

> You can only execute an activated release pipeline.

A Release Manager can further monitor the pipeline to execute and approve or reject the package approval request, and check the progress of each stage (task) in an executed or in-execution pipeline. While monitoring, you also get an option to rollback. The primary function of rollback is to revert to the last successful deployment in the

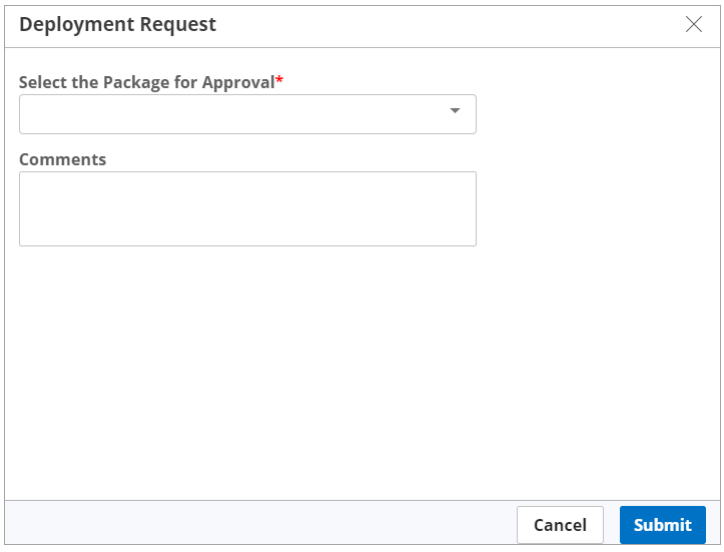
event of unexpected issues or failures in a new deployment. This ensures that the system can quickly return to a stable and known state, minimizing downtime and reducing the impact of problems that may arise during the deployment process.

You can view the history to check the attempts and failure details of the deactivated pipeline. A Release Manager accepts or rejects the generated package approval request before deploying the package into the production or target environment. In case multi-factor authentication is enabled, the Release Manager needs to enter the authentication code received on the registered email address before rejecting or approving the package request.

Once the Release Manager grants the approval, the next task in the pipeline gets executed.

To execute a release pipeline, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **Execute** next to the required release pipeline. The Execute Confirmation dialog appears.
3. Click **Execute** to confirm. The Pipeline Monitoring page appears displaying the pipeline execution numbers and the progress of each stage in the release pipeline. Typically, Stage 1 of the release pipeline is Deployment Approval.
4. Click **Generate Request for Package Approval** link under Stage 1. The Deployment Request dialog appears.




5. Select the required integration pipeline from the dropdown list to use in the release pipeline.

On selection, the package details appear that include the integration pipeline name, pipeline type, created by, and the creation date.

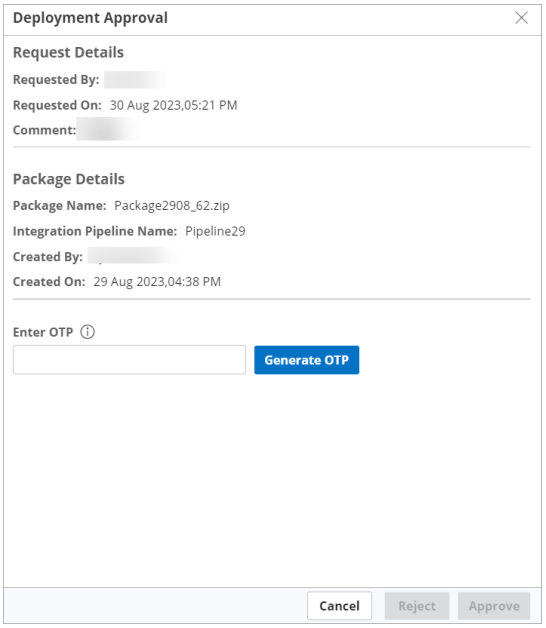
6. Enter a comment.

> You are prompted to enter a comment only if the Allow Add Comments toggle is enabled while configuring the properties of the Deployment Approval task. For more information, see Deployment Approval.

7. Click **Submit**. The package approval request is submitted to the Release Manager. Once the Release Manager approves the approval, the next task in the pipeline gets executed. Upon successful execution, the status of the pipeline changes to Success.

To accept or reject the package approval request, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **Monitor** next to the required release pipeline. A new page appears displaying the pipeline execution numbers and the progress of each stage in the release pipeline.
3. Click **Approval Required for Selected Package** link under Stage 1. The Deployment Approval dialog appears. It displays the package and request details including the integration pipeline name, pipeline type, created by, creation date, requested by, request date, and comment.



> No comment appears in case the Allow Add Comments toggle is disabled while configuring the properties of the Deployment Approval task. For more information, see Deployment Approval.

4. Click **Generate OTP**. An OTP is sent to your registered email address.
5. Enter the OTP and then click **Verify OTP**.
6. On successful verification, click **Approve**.
   On approval, the next task gets executed based on the release pipeline flow. Upon successful execution, the status of the pipeline changes to Success.

To monitor a pipeline to check the progress of each stage, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **Monitor** next to the required pipeline. The Pipeline Monitoring page appears displaying the pipeline execution numbers and the progress of each stage in the pipeline. Apart from this, you can also see the pipeline execution date and executioner details.

To rollback to the last successful pipeline execution, perform the following steps:

> The Rollback option is only applicable for portals.

1. Go to the **Pipeline Execution** tab.
2. Click **Monitor** next to the required release pipeline. A new page appears displaying the pipeline execution numbers and the progress of each stage in the release pipeline.
3. Click the **Rollback** link. This link only appears next to the last successful pipeline execution. A Rollback Confirmation dialog appears.
4. Click **Rollback** to confirm.
   It reverts all the current execution changes to the previous successful execution changes.

To view the history of a pipeline, perform the following steps:

1. Go to the **Pipeline Execution** tab.
2. Click **History** next to the required pipeline. The Pipeline Monitoring page appears displaying the pipeline execution numbers and the progress of each stage in the pipeline. Apart from this, you can also see the pipeline execution date and executioner details. Here, you can click the required link under the failed stage to look for possible errors.

# Creating and managing vault and token

A vault is a secure digital store that stores tokens. Tokens are secure digital objects that can store credentials, critical assets, information (for example credit card details, passwords, and so on), or dynamic data required during the execution of deployment pipelines.

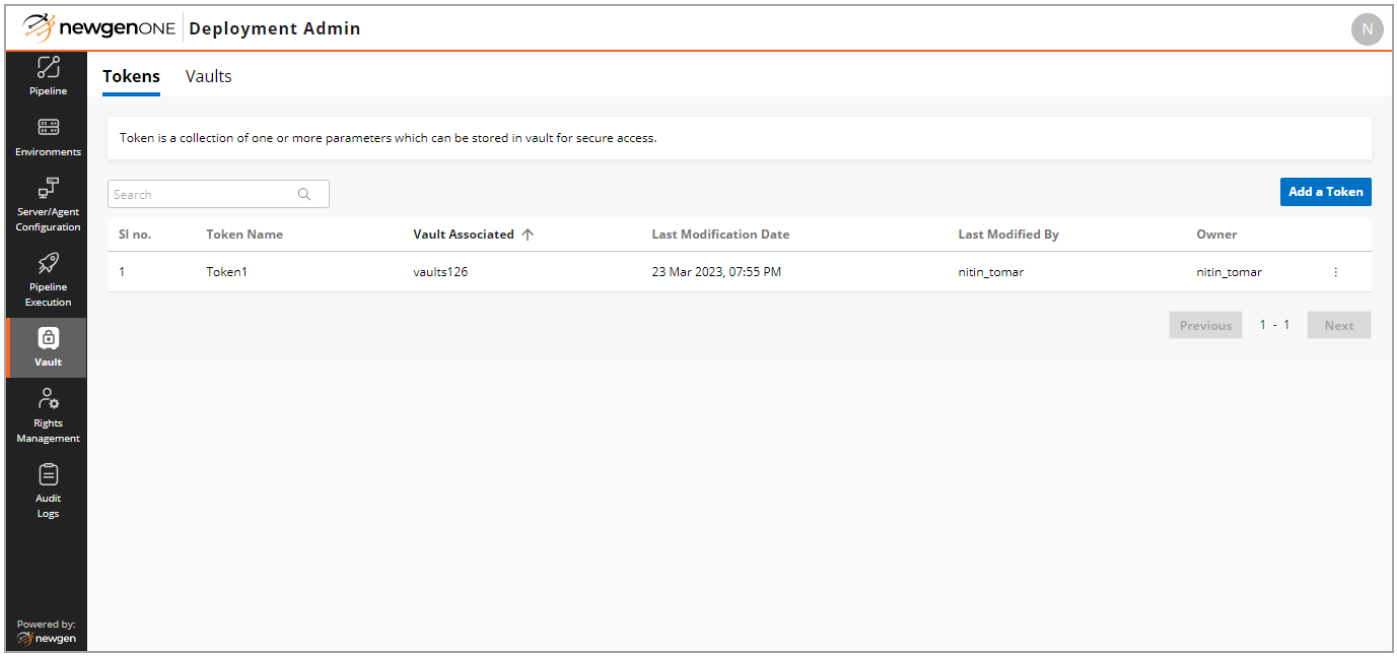
The Vault page allows you to add, edit, and delete tokens and vaults as per your requirement. It comprises the following tabs:

- Tokens
- Vaults

The Vault page is accessible to release managers and other custom roles with specific permission. By default, release managers can add, edit, and delete vaults and tokens.

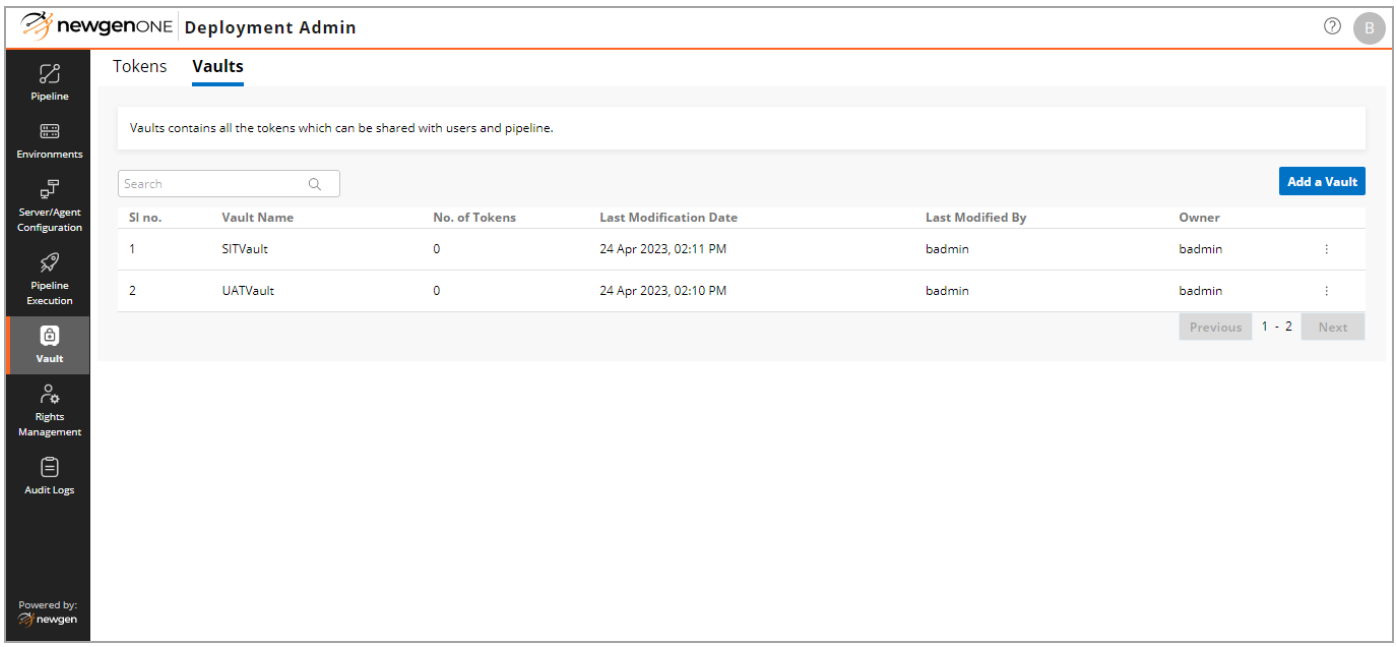
The Tokens tab provides you with the following options and information:

| Field | Operation |
|---|---|
| Search bar | This field allows you to search for an existing token. |
| Token details | This area displays all the available tokens and their details including the token name, associated vault name, last modification date, the user responsible for the last modification, and owner name. |
| Add a Token | This button allows you to add a new token as per your requirement. |
| Token action menu | This action item menu allows you to edit or remove a token. |




The Vaults tab provides you with the following options and information:

| Field | Operation |
|---|---|
| Search bar | This field allows you to search an existing vault. |
| Vault details | This area displays all the available vaults and their details including the vault name, number of tokens in each vault, last modification date, the user responsible for the last modification, and owner name. |
| Add a Vault | This button allows you to add a new token as per your requirement. |
| Vault action menu | This action item menu allows you to edit or remove a vault. |




This chapter describes:

- Adding a vault
- Adding a token

# Adding a vault

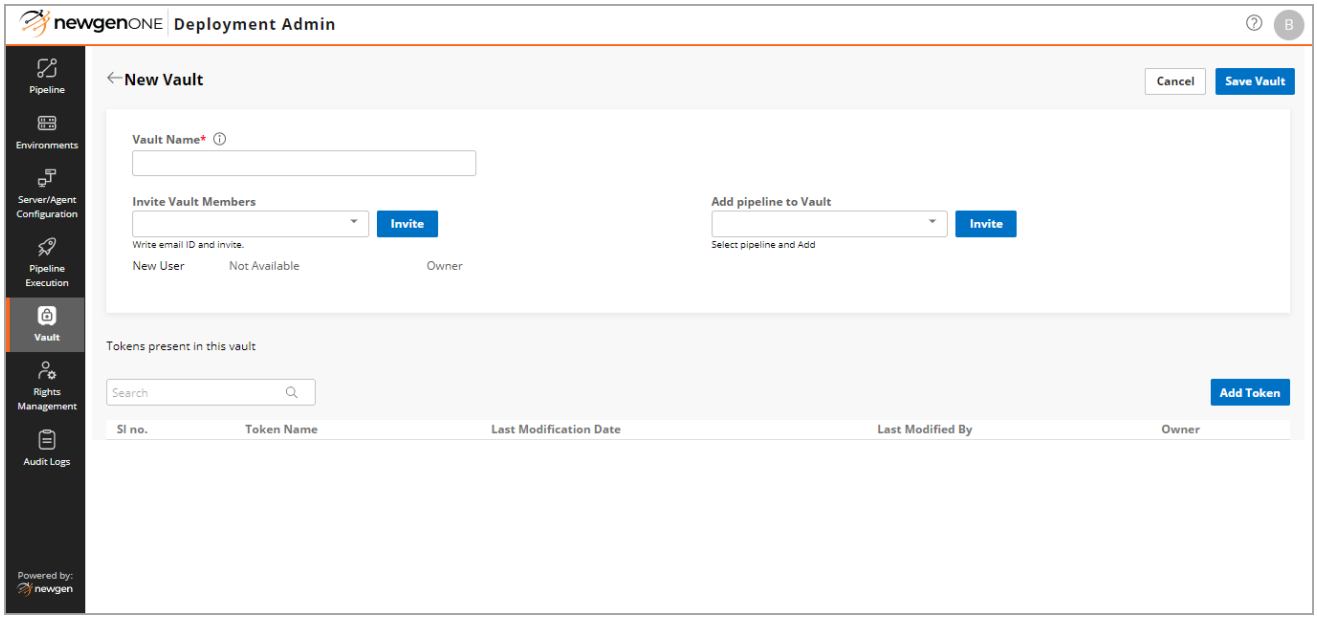
The Vaults tab under the Vault menu allows you to add, edit, and remove vaults.

To add a vault, perform the following steps:

1. Go to the **Vaults** tab.
2. Click **Vaults**.
3. Click **Add a Vault**. The New Vault page appears.
4. Enter the following details in the respective fields:

| Field | Operation |
|---|---|
| Vault Name | Enter the name of the vault. |
| Invite Vault Members | Select users from the dropdown list and click **Add** to add members to the vault. |
| Add pipeline to Vault | Select a deployment pipeline from the dropdown list and click **Add** to add the pipeline to the vault. |

| Field | Operation |
| --- | --- |
| Add Token | Click **Add Token**, select the required tokens on the Select Tokens dialog box, and then click **Done**. |




5. Click **Save Vault**. The vault now appears in the vault list under the Vaults tab.

## Related topic(s)

Adding a token

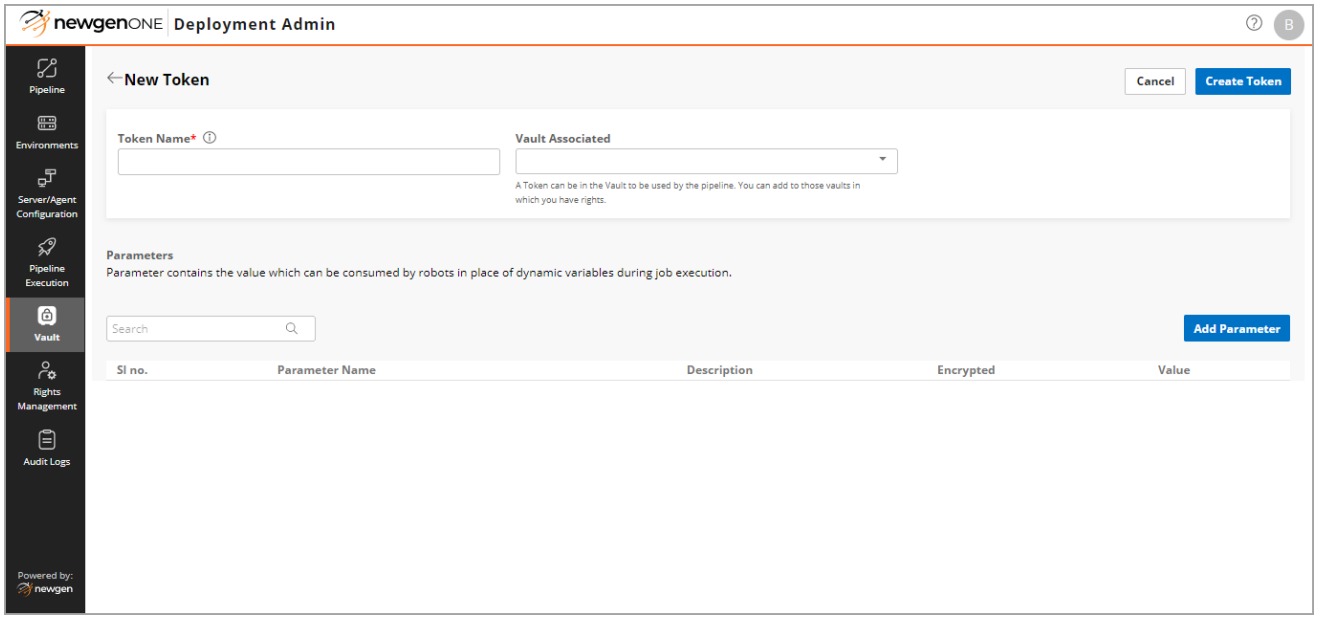
# Adding a token

The Tokens tab under the Vault menu allows you to add, edit, and remove tokens.

To add a token, perform the following steps:

1. Go to the **Vaults** tab.
2. Click **Tokens**.
3. Click **Add a Token**. The New Token page appears.
4. Enter the following details in the respective fields:
    - **Token Name** — Enter the name of the token.

- **Vault Associated** — Select the vault from the dropdown list to associate with this token.
- **Add Parameter** — Add necessary parameters with required values that a robot can use in place of dynamic variables during job execution. Perform the following steps to add parameters:
    a. Click **Add Parameter**. The parameter details fields appear.
    b. Enter the name of the parameter in the Parameter Name field.
    c. (Optional) Enter a description in the Description field.
    d. (Optional) Select the Encrypted checkbox to encrypt the value of the parameter.
    e. Enter the value of the parameter in the Value field.
    f. Click the check icon ✓ to add the parameter.




5. Click **Create Token**. The token now appears in the token list under the Tokens tab.

## Related topic(s)

Adding a vault

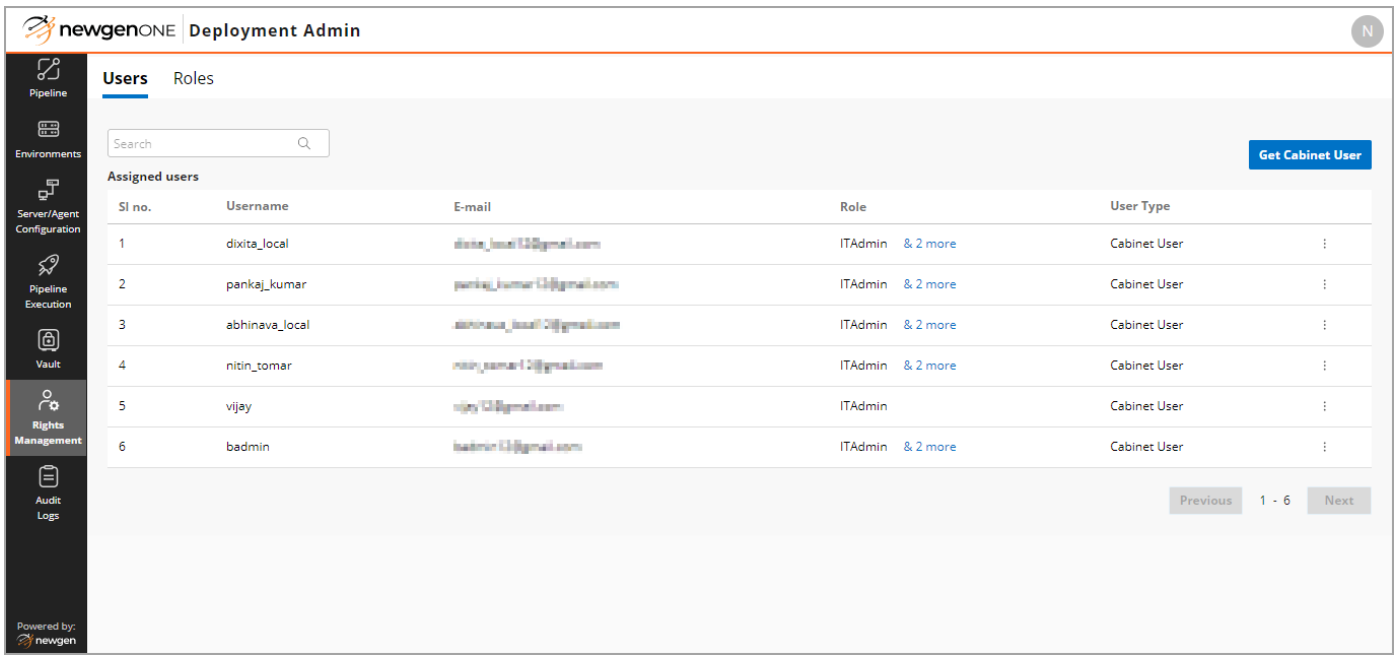
# Managing users and roles

The Rights Management page allows you to import and manage users. It also allows you to create, manage, and assign custom roles. This page comprises the following tabs:

- Users – The Users tab provides you with the option to:
  - Import OmniDocs users
- Roles – The Roles tab provides you with the options to:
  - Create roles
  - Edit roles
  - Assign users to roles

The Rights Management page displays the following options under the Users tab:

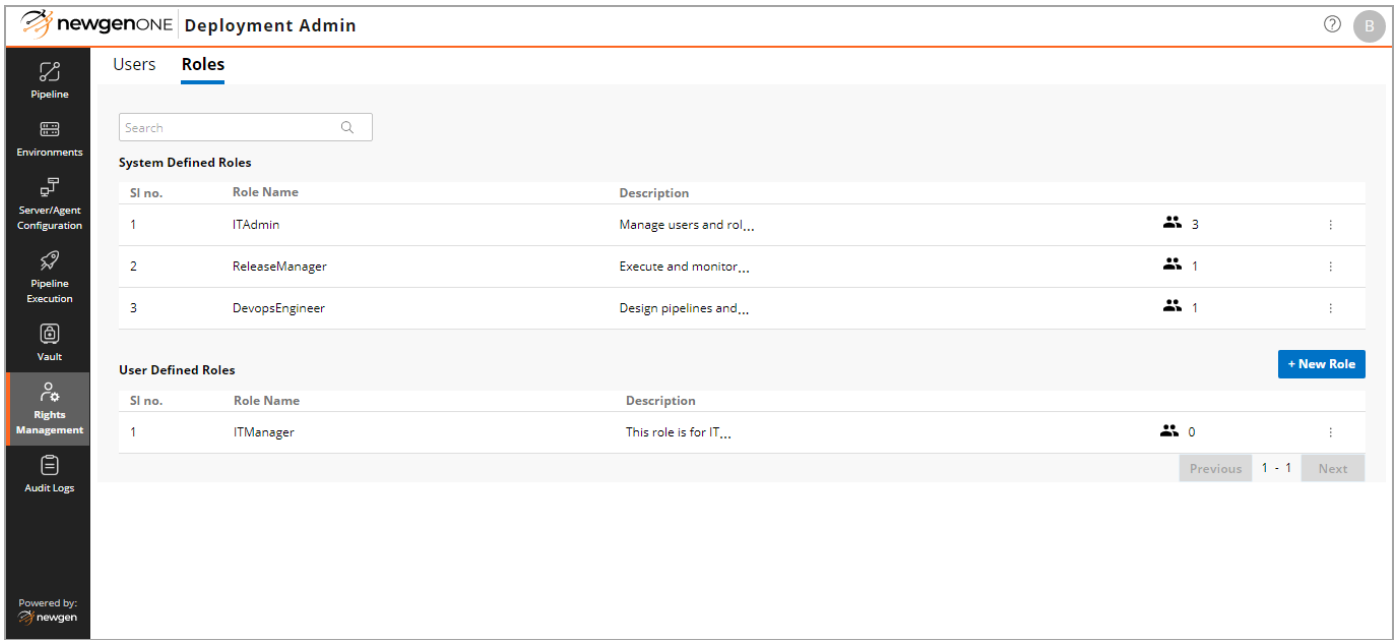
| Field | Operation |
|---|---|
| Search bar | This field allows you to search for an existing user. |
| User details | This area displays all the available users and their details including their email ID, role, and user type. |
| Get OmniDocs User | This button allows you to import existing users from Newgen OmniDocs. |
| User menu | The vertical ellipsis button on each listed user provides the following list of operations:<br>• **Edit User** – Clicking this option displays the Update User dialog box. On the Update User dialog, you can make the necessary changes and then click **Update User** to modify the user.<br>• **Remove** – Clicking this option displays the confirmation dialog box. Click **Yes** to delete the user. |




The Rights Management graphical user interface (GUI) displays the following options under the Roles tab:

| Field | Operation |
|---|---|
| Search bar | This field allows you to search for an existing role. |
| System Defined Roles | This section displays all the pre-defined roles with their description and the number of users assigned to the role. There are three system-defined roles in Deployment Admin:<br>• **ITAdmin** — The main role of an IT administrator is to manage the user rights and roles in Deployment Admin.<br>• **ReleaseManager** — The role of a release manager is to manage the user rights for servers and agents, pipelines, environments, roles, vault and token, audit logs, and release pipelines.<br>• **DeveopsEngineer** — The role of a development engineer is to manage the user rights for servers and agents, pipelines, environments, roles, vault and token, and release pipelines. |
| Role menu | This menu icon reveals the following context menus:<br>• **View** – Displays the permission details, assigned users, and basic details of the role.<br>• **Assign Users** – Allows you to assign users to the role. |
| Add a New Role | This button allows you to add a new user-defined role. |
| User Defined Roles | This section displays the roles you define. |

| Field | Operation |
|---|---|
| Edit | This icon allows you to edit permissions, add or remove users, and edit basic details of a particular user-defined role. |




The Rights Management tab is accessible to administrators and other custom roles with specific permission. By default, administrators can import, view, and edit users as well as create, edit, and assign roles. Only administrators can remove users and delete roles.

This chapter describes:

- Imporing cabinet users
- Creating a new role
- Editing roles

# Importing cabinet users

The Roles tab of the User Management GUI allows you to import OmniDocs users.

To import OmniDocs users, perform the following steps:

1. Click **User Directory** on the upper-right of the Users tab. The User Directory window appears.
2. Click **Newgen OmniDocs**.

3. Enter the OmniDocs server details.
4. Click **Test Connection**. A successful connection enables the Proceed button.
5. Click **Proceed**.
6. Click **Get OmniDocs User** on the upper-right of the Users tab. The Get OmniDocs User window appears.
7. Enter the following fields:

| Field | Description |
|---|---|
| Username | Select the required OmniDocs Username from this dropdown list to import. |
| Add | Click Add to import the details of the selected user. |
| Full Name (Optional) | Full name of the user is automatically populated. |
| Never Expire | Never Expire is non-editable for importing users. |
| Password expires in | Enter the password expiry date and time. |
| Email Id | Enter the email id of the user. |
| Fax number (Optional) | Enter the fax number of the user. |
| Mobile Number | Enter the mobile number of the user. |
| Assign RPA Role | Select the required role for the user from this dropdown list. |

8. Click **Create User**.

# Creating a new role

There are two different kinds of roles in the Deployment Admin:

- **SYSTEM DEFINED ROLES** – These roles are pre-defined in the Deployment Admin. Permissions for these roles are not editable.
- **USER DEFINED ROLES** – You can define these roles with a unique set of permissions as per your requirement.

By default, only administrators can add new roles.

The Roles tab of the Rights Management GUI provides you with the option to create user-defined roles. To create a user-defined role, perform the following steps:

1. Click the **+ New Role** button on the Roles tab of the Rights Management GUI. Add a new Role dialog appears.
2. Enter the name of the role in the **Role Name** field.
3. (Optional) Enter a description of the role in the **Role Description** field.
4. Click **Add Role**. The role now appears in the USER DEFINED ROLES section on the Roles tab.

Once created, you can edit the role permissions to meet your requirement.

# Editing a role

The Roles tab of the User Management GUI provides you with the option to edit a newly created role, as well as any existing role in the USER DEFINED ROLES section.

By default, only administrators can edit user-defined roles.

To edit a role, perform the following steps:

1. Navigate to the required role in the USER DEFINED ROLES section and then click the edit icon on that specific role. The edit page appears.
2. On the Permissions tab, click the required permission group on the left of the page. The related permissions appear on the right.
3. Select the toggle next to the permission group name on the right section. This activates all the permissions in this group.
4. Customize the permissions as per the requirement with the help of the toggles adjacent to each permission.
5. Repeat steps 2 to 4 to modify all the required permission groups.
6. Click **Save**. A confirmation pop-up appears on successfully saving the role with the modifications.

Once the permissions are set, you can assign users to a role.

### Related topic(s)

- Creating a new role
- Imporing cabinet users
- Assigning users to roles

# Assigning users to roles

Deployment Admin allows you to assign users to both system-defined roles and user-defined roles.

By default, only administrators can assign users to roles.

#### Assigning users to a system-defined role

To assign users to system-defined roles, perform the following steps:

1. Navigate to the Roles tab of the Rights Management user interface.
2. Click the vertical ellipsis icon ⋮ on the required role under the SYSTEM DEFINED ROLES section and then select **Assign Users**. The Assign Users page appears.
3. Navigate to the intended user in the Available Users section on the left.
4. Click the add icon ⊕ adjacent to the required user. The user now appears in the Selected Users section on the right. You may use the delete icon 🗑 adjacent to the users in the Selected Users section to remove users.
5. Repeat steps 4 to 5 to select all the required users.
6. Click **Save**. The selected users are now assigned to the role. A message appears confirming the update.

#### Assigning users to a user-defined role

To assign users to user-defined roles, perform the following steps:

1. Navigate to the Roles tab of the Rights Management user interface.

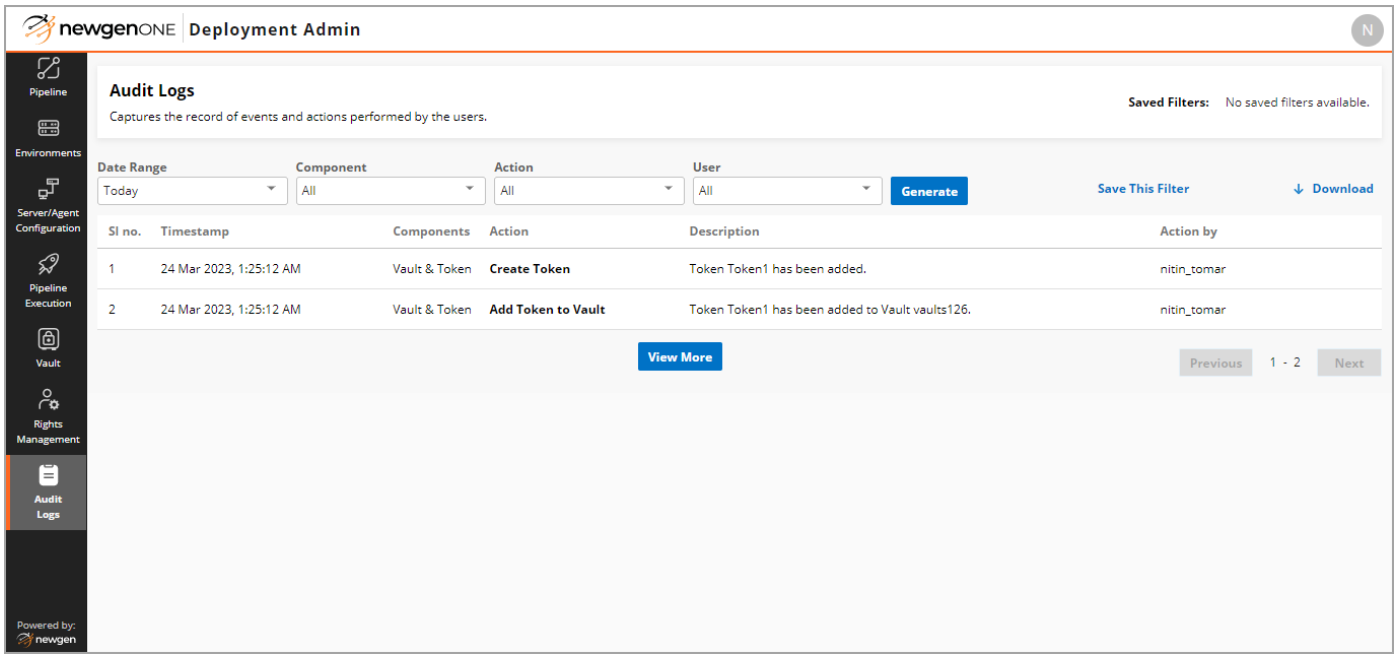
2. Click the edit icon on the required role under the USER DEFINED ROLES section. The edit page appears.
3. Click the **Assign Users** tab. The Assign Users page appears.
4. Navigate to the intended user in the **Available Users** section on the left.
5. Click the add icon adjacent to the required user. The user now appears in the Selected Users section on the right. You may use the delete icon adjacent to the users in the Selected Users section to remove them.
6. Repeat steps 4 to 5 to select all the required users.
7. Click **Save**. The selected users are now assigned to the role. A message appears confirming the update.

# Accessing logs

Deployment Admin allows you to monitor all the actions performed by each user through Audit Logs. With this functionality, you can customize the logs with the help of various filters and download the logs.

The Audit Logs GUI displays the following options and information fields:

| Field | Operation |
|---|---|
| Date Range | This field allows you to select a time frame from the dropdown list. |
| Component | This field allows you to select a specific component from the dropdown list. |
| Action | This field allows you to select an action from the dropdown list. |
| User | This field allows you to select a user from the dropdown list. |
| Generate | This button generates the logs based on the indicated criteria. |
| Save This Filter | This button allows you to save the selected filter criteria. |
| Download | This button downloads the generated log in .txt or .csv format. |
| Result area | This area displays the filtered log entries. |



To filter and download logs, perform the following steps:

1. On the Audit Logs page, select the **Date Range**, **Component**, **Action**, and **User** as per requirement from their respective dropdown list.
2. Click **Generate**. The filtered logs appear in the result area.
3. Click **Download** and then either select **Download in text** to download the filtered logs in .txt format or select **Download in csv** to download the filtered logs in .csv format.