

Example 5: Running a release pipeline

4 minute read

Developer productivity

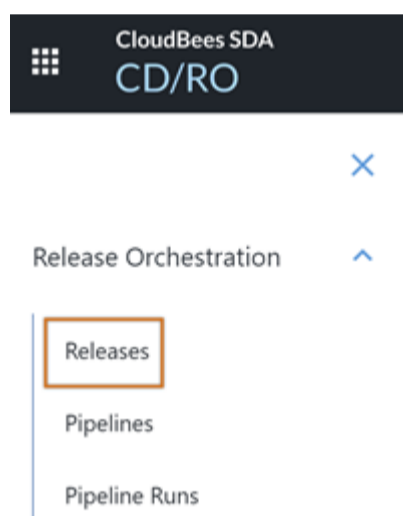
Releases are used to manage the tasks associated with releasing multiple applications or components together.

Perform these steps to create and run a CloudBees CD/RO release pipeline that does the following:

- Invokes the procedure
- Invokes the Tomcat and Helm Chart application models you created in earlier examples
- Implements manual and automated gates

Create a new release

1. From the main menu, navigate to **Release Orchestration > Releases**.



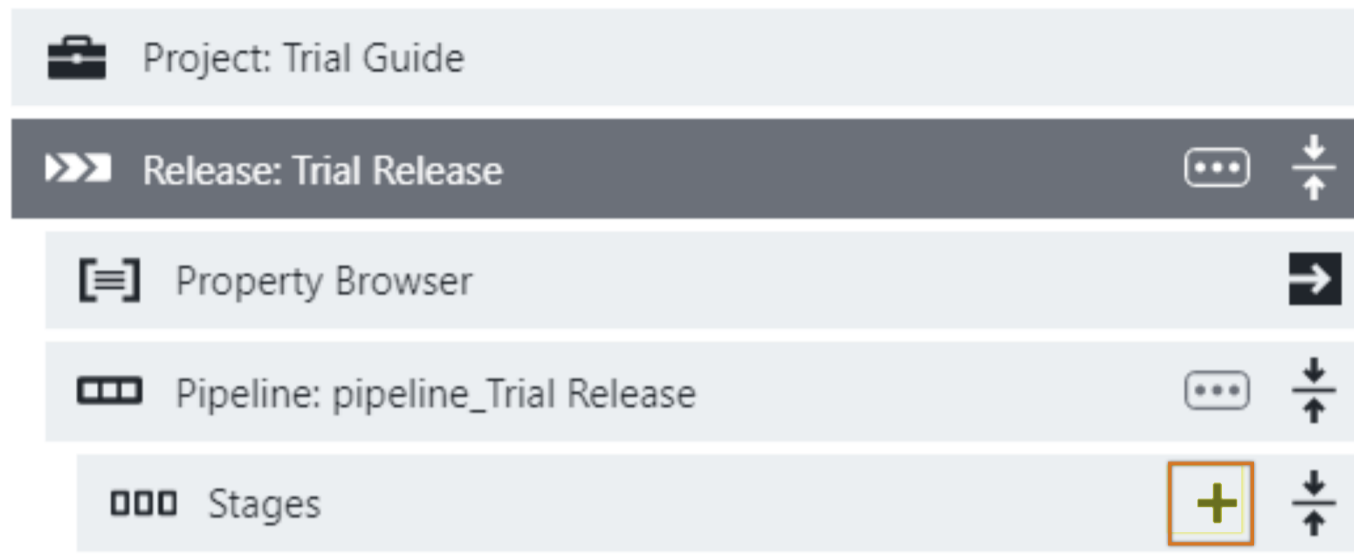
2. Select **There are no releases. Add one +** or **New +**. The New Release dialog appears.
3. Provide a release name such as *Trial Release*, and select a project (for example, *Trial Guide*).
4. Select **OK** to create the new release definition.

Add stage details

1. Rename the stage from **Stage 1** to QA.
 - a. Select the stage menu on the right and then select **Stage Details**.
 - b. Enter *QA* for the name of the stage, then select **OK** to save the change.
2. Create a stage called *UAT*.
 - a. From the Hierarchy Menu on the left, select the plus (+) next to Stages.

Submit Feedback

Hierarchy Menu



The New Stage Dialog appears.

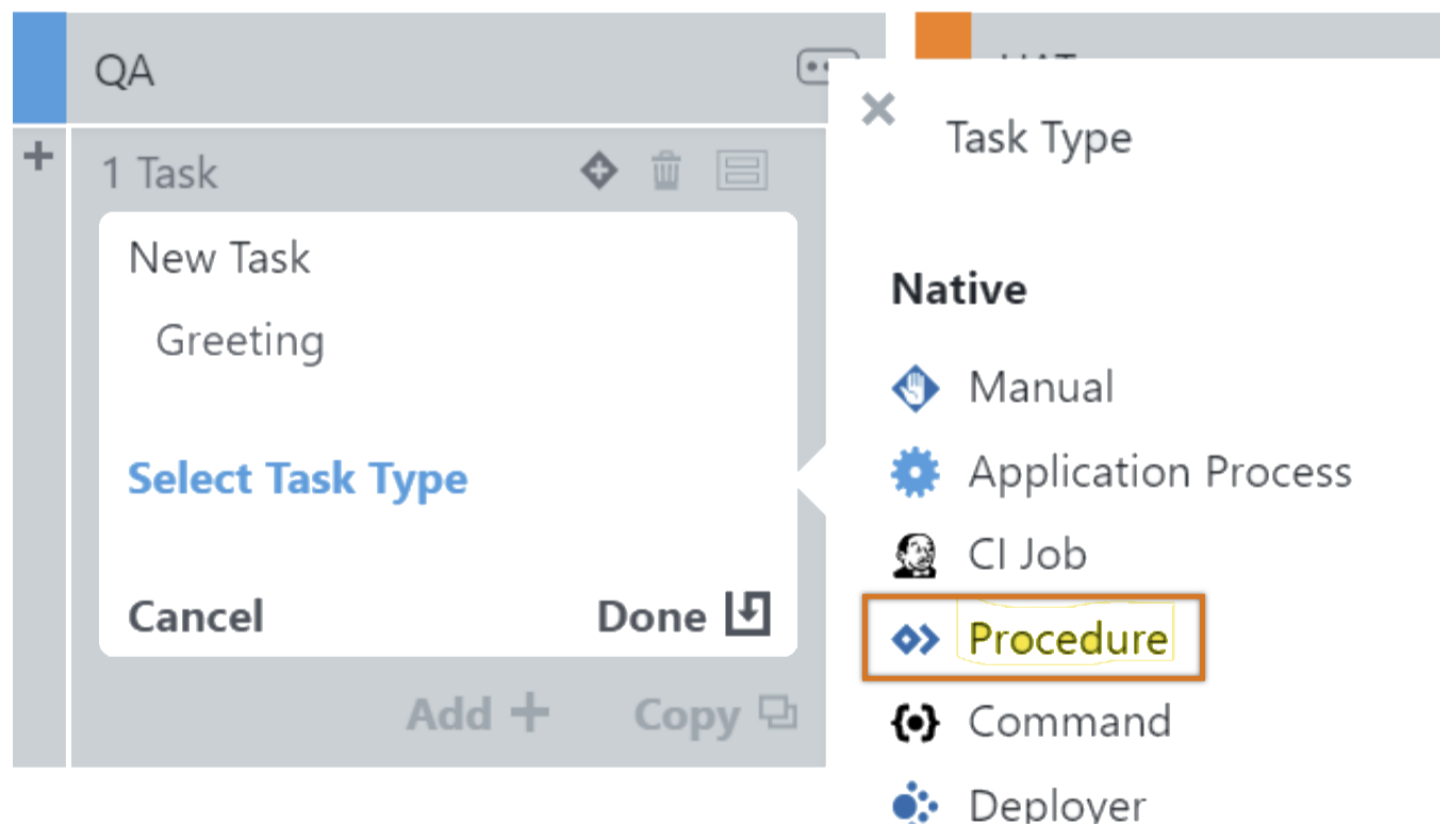
- b. In the New Stage dialog, enter name *UAT* and then select **OK** to create the new stage.

Add a procedure task

Add the procedure created earlier to the QA stage.



1. Select **Add +**.
2. Enter a task name such as *Greeting*, then select **Select Task Type**.
3. Select **Procedure** as the task type.



4. Select **Define** in the New Task box.
5. Select the project and procedure for the procedure definition you created earlier. Then select the right arrow beside **Input Parameters** to enter the parameters.

Edit Task: Greeting
In Stage: QA

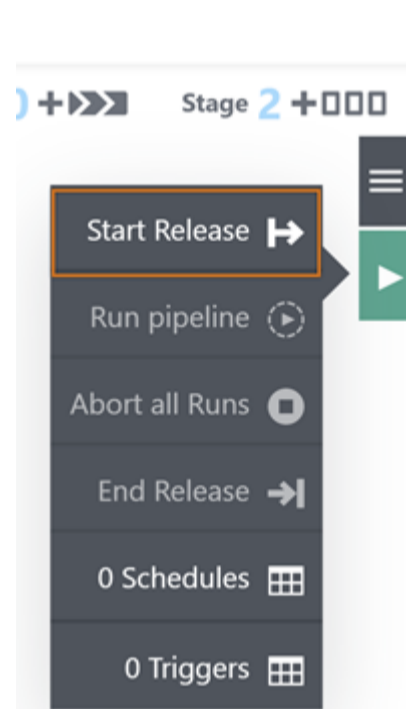
- Application Process
- Microservice Process
- CI Job
- Deployer
- Procedure**
- Manual

- Trial Guide
- Trial Procedure
- 2 Input Parameters ⚠️ 2 Required
- 1 Output Parameter
- Assign a Resource or Resource Pool

- Provide parameter values:
 - In the **Greet Name** field, enter **DevOpsGuru**.
 - In the CD User field, select one of the users from the list.
- Select **OK**.
- Select the right arrow beside the **Output Parameter**. Check the **Show in Stage Summary as** box, and then select **OK** to save the task definition.

Start the release and run the release pipeline

- Select the **Run** button, and then select **Start Release**.



- Select **Run**. Note the stages running.
- Once complete, select the **Summary** link on the **QA** stage and note how the procedure output was propagated up.
- Select the task name to view the job details, and then close the dialog to continue.

Add a pipeline parameter

- Use the breadcrumbs to return to the release definition editor.

Trial Guide / Releases / **Trial Release** / Runs

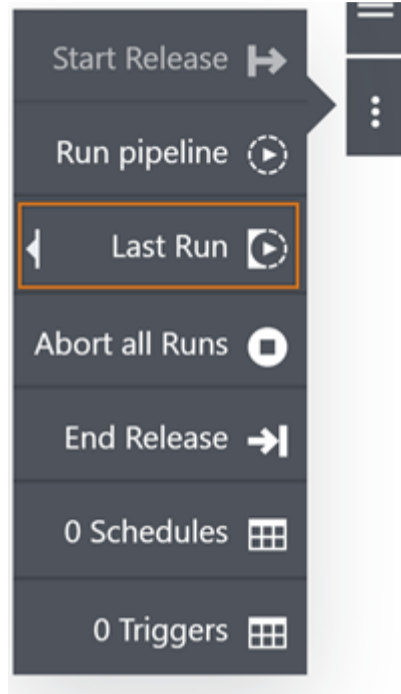
- From the Hierarchy Menu, select the context menu beside the **Pipeline: pipeline_Trial Release**.
- Select **Parameters** and add a parameter, *helloUser*.
- Select **OK** to save, and then exit the Parameters dialog.
- Modify the task to use the parameter value.

Tip Select the task name.

6. Modify the procedure task Greet Name to be `#[helloUser]`.

Run the release pipeline

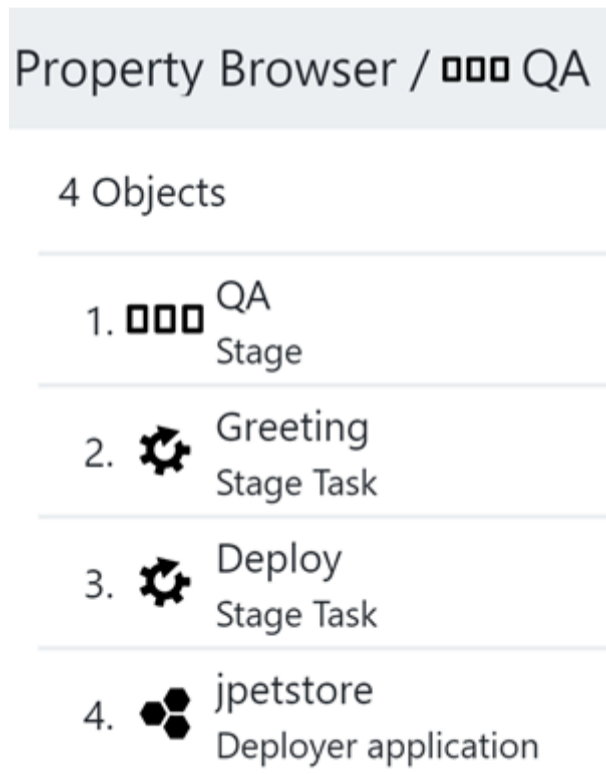
1. From the release options menu on the right, select **Last Run**, and then select **Run**.



2. To see where this value was applied, look at the job results.

Tip Display the job details by selecting the task name.

3. Open the log associated with the **Hello <greetName>** step.
4. Examine the pipeline runtime properties and parameters.
 - a. Select **Pipeline Runs** at the top of the screen.
 - b. Select the Actions menu to view the properties and parameters.
 - c. Select Expand to view the tasks. Select the task menu button to view the task properties.

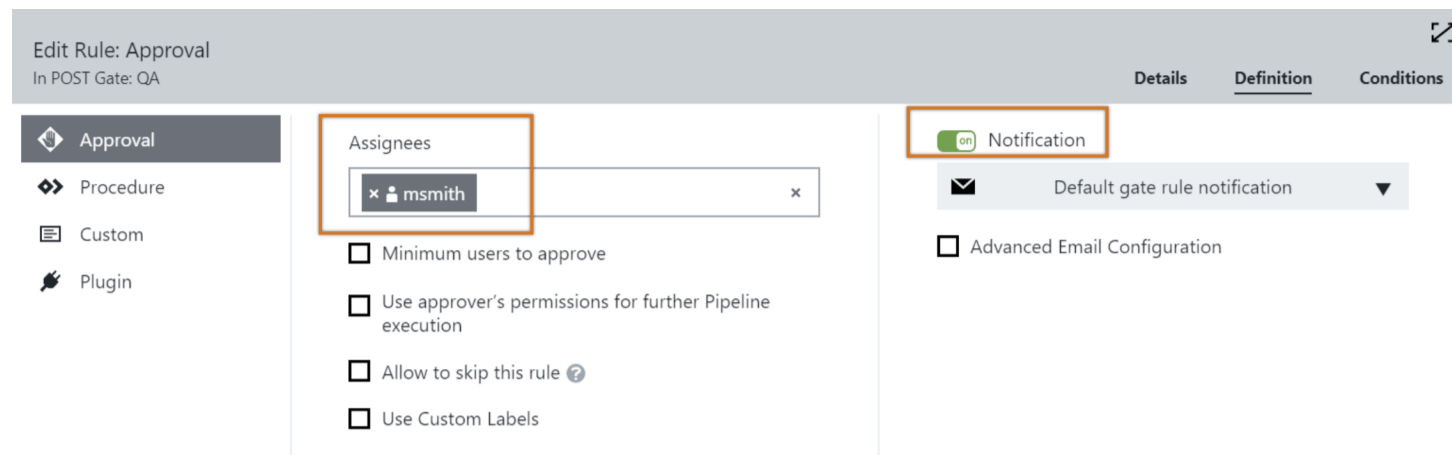


- d. Open the Jobs Details to view the job properties and parameters.

Add a manual gate

1. Use the breadcrumbs to navigate back to the release definition editor.
2. Select the + on the upper right side of a task to open the gates view. Select **Add +** to create a new rule for the gate.

3. Provide a name for the new rule, such as *Approve*. Select **Select Rule type** and **Approval** as the type.
4. Select **Define**.
5. Enter your username under **Assignees**, and then select **OK** to complete the gate definition.



Run a new release pipeline

1. In the Actions menu, select the **Last Run** option to reuse the parameter value you selected earlier, and then select **Run**.

The pipeline halts because it is waiting for the manual approval. (If configured, an email will also be sent with options to approve).

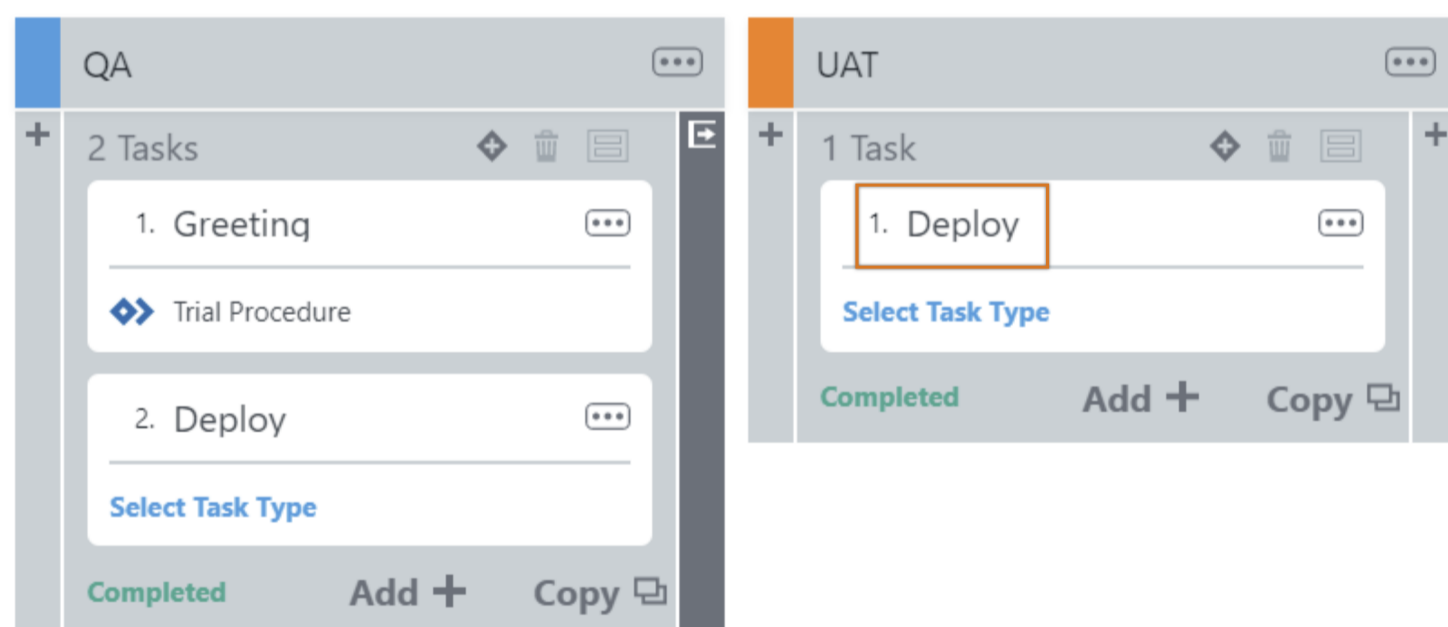
2. Select the **Response required** pull-down to select the user designated to approve.
3. Select **Approve** and provide a comment, and then select **OK**.

Examine the audit report

Use the pipeline runtime Actions menu to access the audit reports. Select **Audit Reports**.

Add a deployer to each stage

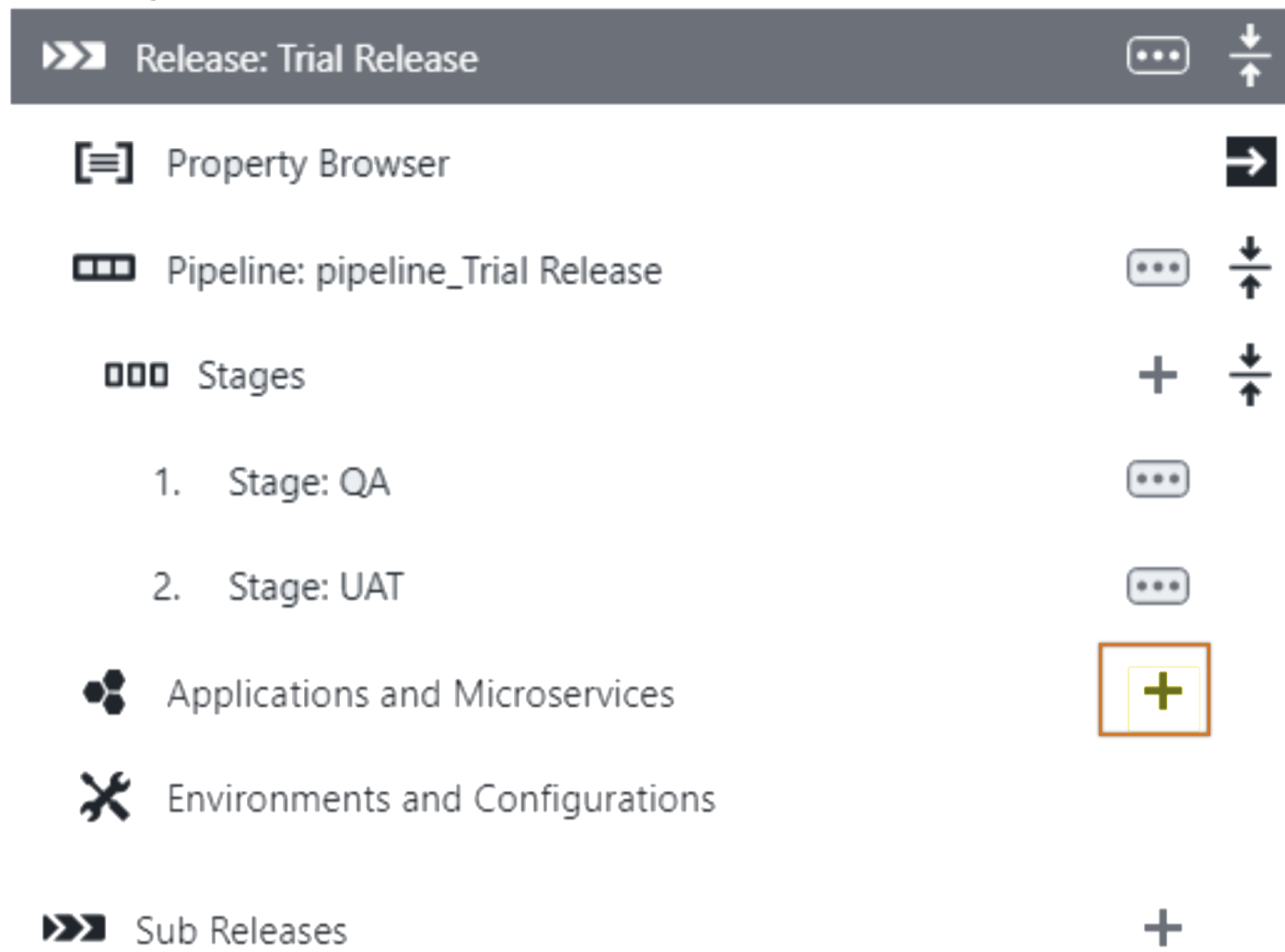
1. Use the breadcrumb to return to the release definition editor. Add a new task called *Deploy* in each stage.



2. Select **Select Task Type** and choose **Deployer** for both Deploy tasks.
3. Select **Define**.
4. Select **Deploy Serially**.
5. Select **OK** to complete the task definition.

Add the applications to the release

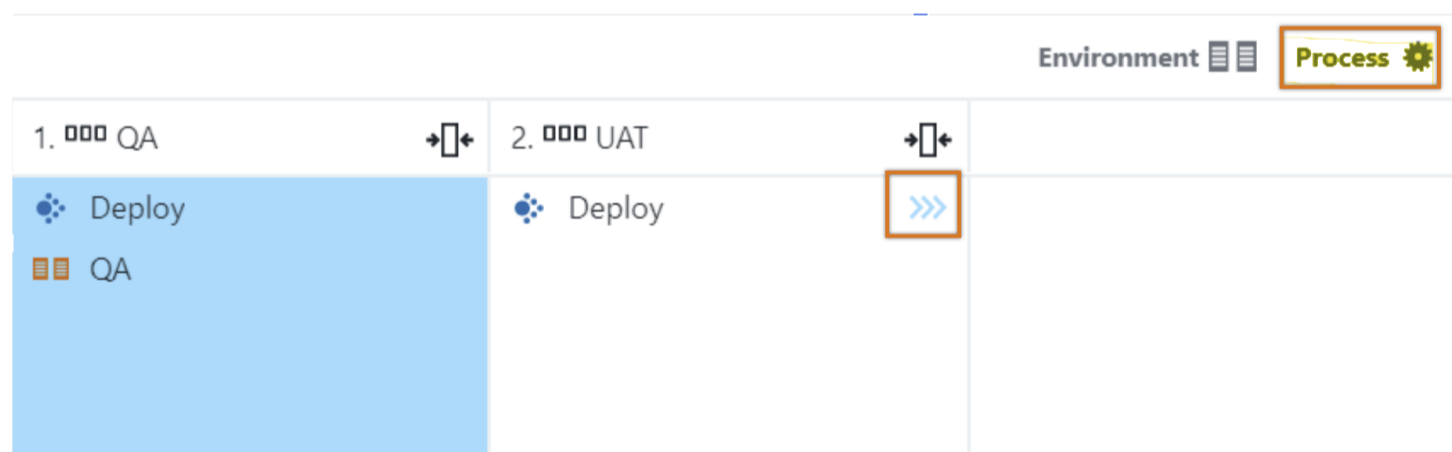
1. In the Hierarchy Menu on the left, select the plus (+) next to **Applications and Microservices**.



2. Select both applications, and select the blue double-arrow button (>>) to add the applications. Then select the back button (<) to close the selection view.
3. Select the snapshot for the jpetstore application that you created earlier.
4. Select **OK**.

Configure the application to run in QA and UAT environments

1. Select the right arrow next to **Environments and Configurations** in the Hierarchy Menu on the left.
2. Select the cell under the QA column for **jpetstore** and then select the **Environment** tab.
3. Select the QA environment from the Trial Guide project, and then select **OK**.
4. Select the **Process** tab, and then select **Deploy**.



5. Do the same for the jpetstore application under **UAT**. But this time, select the **Trial Guide UAT** environment.
6. Similarly, set up the RSS application to point to the Helm Deploy QA and UAT environments.

Q Find...			
	2 Applications	1. QA	2. UAT
1.	jpetstore	Deploy QA Deploy	Deploy UAT Deploy
2.	RSS	Deploy RSS - QA Deploy Application	Deploy RSS - UAT Deploy Application

7. Select **OK** to save the environment configuration.

Run the final release pipeline

1. Select the Actions menu, and then select **Last Run**.
2. Once the Deploy task has started, select **Summary** in the QA stage.

The screenshot displays the trial release pipeline interface. At the top, a header bar shows the pipeline name "Trial Release_pipeline_Trial Release_12_202" with ID "10302170452", the current stage "QA", and the date "Mar 02, 2021". Below this, the "QA" stage is selected, showing a summary card with the following details:

- Last Run: Mar 02, 2021 - 12:05
- Duration: 16:03
- 1st Run
- 2 Tasks, 2 Completed
- Task 1: Greeting (1st), Trial Procedure (100%)
- Task 2: Deploy (1st), Deployer (100%)

An orange arrow points to the "Summary" link in the summary card. To the right, a "Stage / QA" window is open, showing a list of tasks with checkmarks:

- RSS - Deploy Application on RSS - QA ✓
- jpetstore - Deploy on QA ✓

The user "whoRanMe admin" is listed at the bottom of the window.

3. Allow the release pipeline to complete.
4. Provide the required response to continue the pipeline. Note the new path to production view.
5. Select the Actions menu, and then select **Audit Reports** to view the Audit Report.

[← Example 4: Running a Helm chart deployment](#)

[Example 6: Exploring DSL →](#)