



Release Owl

Technical Documentation

Version 1.04



Contents

Core Capabilities	3
Build Pipelines.....	3
Release Pipelines.....	3
Application Overview	4
Dashboard.....	4
Build Pipeline	6
Creating a new Build Pipeline:	6
Release Pipelines.....	15
Creating a New Release Pipeline	15
Environments.....	27
Credential Manager	31
Inbox	33



Release Owl

Release Owl is a native DevOps platform designed to accomplish continuous delivery for SAP® cloud. Release Owl is a product suite that has a rich set of features to achieve seamless continuous integration and well-orchestrated release management for the applications built on SAP® cloud platform.

Core Capabilities

Build Pipelines

Build Pipelines can be created and run with Release Owl, which can do the following to achieve continuous Integration for SAP cloud applications. It fetches the source from GIT Repository and package the Application as MTA artefact.

Release Pipelines

Release Pipelines can be created and run with ReleaseOwl that aim to the do the following:

- A. Automated Deployment of MTA files in multiple SAP cloud environments (called **Stages**) like QA, UAT, Stage etc.
- B. The ability to set pre-deployment / post-deployment actions for each stage which includes
 - i. The ability to send an approval task for a lead / manager to approve the deployment.
 - ii. The ability to assign a manual task to a team member in case any configuration changes to be done and tracked for each stage.
- C. Run automated tests built using Selenium on every cloud environment and report the test results.



Application Overview

Once a user logs in to the ReleaseOwl application, the Home Page will look as follows:

A screenshot of the Release Owl application dashboard. The left-hand pane contains a menu with options: Dashboard, Build pipelines, Release pipelines, Environments, Credential Manager, and Inbox. The right-hand pane displays the "Project >> Dashboard" view. It features two main sections: "Recent Build Cycles" and "Recent Release Cycles". The "Recent Build Cycles" section has a table with columns: Build Pipeline, Build #, Repo, Branch, and Result. It lists five build cycles, with the first three marked "Success" and the last two marked "Failure". The "Recent Release Cycles" section has a table with columns: Release Pipeline, Environment, Build Pipeline Ref, and Status. It lists one release cycle with a status of "Completed".

Build Pipeline	Build #	Repo	Branch	Result
Demo1	1	demo.git	master	Success
Demo	1	demo.git	master	Success
ROdemo	1	demo.git	master	Success
ROpipeline	2	demo.git	master	Failure
ROpipeline	1	demo.git	master	Success

Release Pipeline	Environment	Build Pipeline Ref	Status
00001	Dev	Demo1	Completed

The Menu options are available on the left-hand pane of the Home page.

On navigating to each menu option on the left-hand pane, the corresponding details are displayed on the right-hand pane.

- [Dashboard](#)
- [Build Pipelines](#)
- [Release Pipelines](#)
- [Environments](#)
- [Credentials](#)
- [Inbox](#)

Dashboard

By default, the Dashboard page is shown.



Release Owl for SAP® Cloud

Dashboard → Project » Dashboard

Build pipelines

Release pipelines

Environments

Credential Manager

Inbox

Recent Build Cycles Refresh

Build Pipeline	Build #	Repo	Branch	Result
ROpipeline	2	demo.git	master	Failure
ROpipeline	1	demo.git	master	Success
FincanceCloud	1	demo.git	master	Success

On the right-hand pane, the results of recently executed 5 build pipeline cycles and 5 release pipeline cycles are displayed.

Release Owl for SAP® Cloud

Dashboard → Project » Dashboard

Build pipelines

Release pipelines

Environments

Credential Manager

Inbox

Recent Build Cycles Refresh

Build Pipeline	Build #	Repo	Branch	Result
ROpipeline	2	demo.git	master	Failure
ROpipeline	1	demo.git	master	Success
FincanceCloud	1	demo.git	master	Success

Release Owl for SAP® Cloud

Dashboard → Project » Dashboard

Build pipelines

Release pipelines

Environments

Credential Manager

Inbox

Recent Release Cycles Refresh

Release Pipeline	Environment	Build Pipeline Ref	Status
00001	PaIQAEEnv	Demo1	Completed
demo01	PaIQAEEnv	ROpipeline	Completed
demo01	PaIQAEEnv	ROpipeline	Stopped
Demo01	PaIQAEEnv	ROpipeline	Waiting
Demo1	PaIQAEEnv	ROpipeline	Completed



Build Pipeline

The pre-requisite for creating build pipeline is to setup the credentials for connecting to the GIT repository. We can register the credentials in the [Credential Manager](#) section on the left-hand pane.

Build Pipeline allows to connect to the GIT Repository and fetches the changes. It then packages them in to MTA (Multi Target Application) format which is the standard for SAP cloud deployments and then publishes in to an artefact repository.

To navigate to the Build Pipeline menu, click on the menu option Build Pipelines on the left-hand side.

The screenshot shows the Release Owl interface for SAP Cloud. The left-hand menu has 'Build pipelines' highlighted. The main area shows a breadcrumb 'Project » Dashboard » Build pipelines' and a 'Create New Build Pipeline' button. Below is a table of build pipelines.

Build pipelines	Modified on	Repo	Branch	Build now	Results	Delete
BuildPipeline1	09/04/2020 15:26	paldemobb.git	*/master			
DemoBP1	09/04/2020 15:19	demo.git	*/master			

Details related to the specified build like its Name, Modified Date, Created By, Modified By are seen in this grid.

We can trigger the job by clicking on the Run button, view the results by clicking on the Results button, delete the specific job by clicking on the Delete button against the specific build entry under Build Pipeline Cycle grid.

Creating a new Build Pipeline:

We can create a new Build Pipeline by clicking on Create New Build Pipeline link available on the extreme right above the Build Pipeline Cycle grid.

This screenshot is identical to the previous one, but with a blue arrow pointing to the 'Create New Build Pipeline' button in the top right corner of the main area.



On clicking the Create New Pipeline link, the user will be prompted to enter the credentials, select Repo Type, Repo, Branch and Scheduler. The Select Credential screen will look as follows:

This screenshot shows the '1. Select Credential' step of the pipeline creation process. The left sidebar contains navigation links: Dashboard, Build pipelines (highlighted), Release pipelines, Environments, Credential Manager, and Inbox. The main content area has a progress bar with six steps: 1. Select Credential (active), 2. Select RepoType, 3. Select Repo, 4. Select Branch, 5. Builder, and 6. Below the progress bar, the '1. Select Credential' section contains two input fields: 'Build Pipeline Name: *' with a placeholder 'only alphanumerics allowed' and 'SCM Credentials:' with a dropdown arrow.

Release Owl has a Credential Manager wherein you can setup credentials and map them to various systems during the configuration.

You specify the Build Pipeline name that you are going to create against the Job Name field and the current logged in username gets auto populated against the SCM Credentials field.

You get to see the Step2 button only if the JobName is greater than 5 characters.

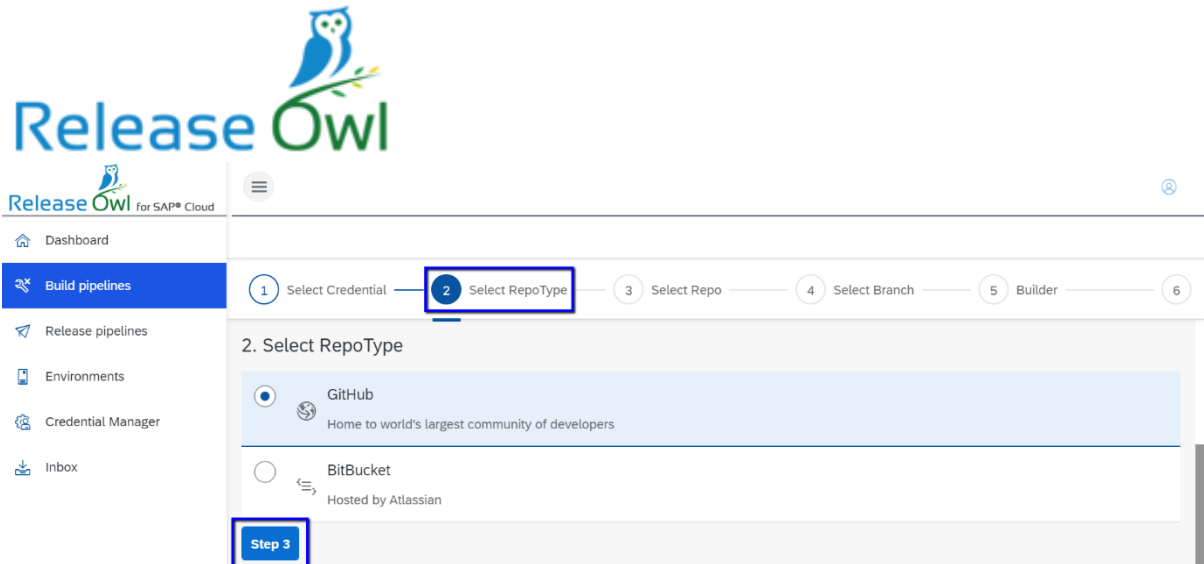
Upon entering the JobName, you can go to the next screen by clicking on Step2 button.

This screenshot shows the '1. Select Credential' screen after some data has been entered. The 'Build Pipeline Name: *' field now contains 'ReleaseOwlDemo1' and the 'SCM Credentials:' dropdown is set to 'SCMGithub'. A blue 'Step 2' button is now visible at the bottom left of the main content area, indicating that the job name is long enough to proceed to the next step.

The next screen is Select Repo Type. Currently GitHub and BitBucket are supported.

On selecting GitHub in this case, the credentials supplied in the first step are used to connect to Git Hub.

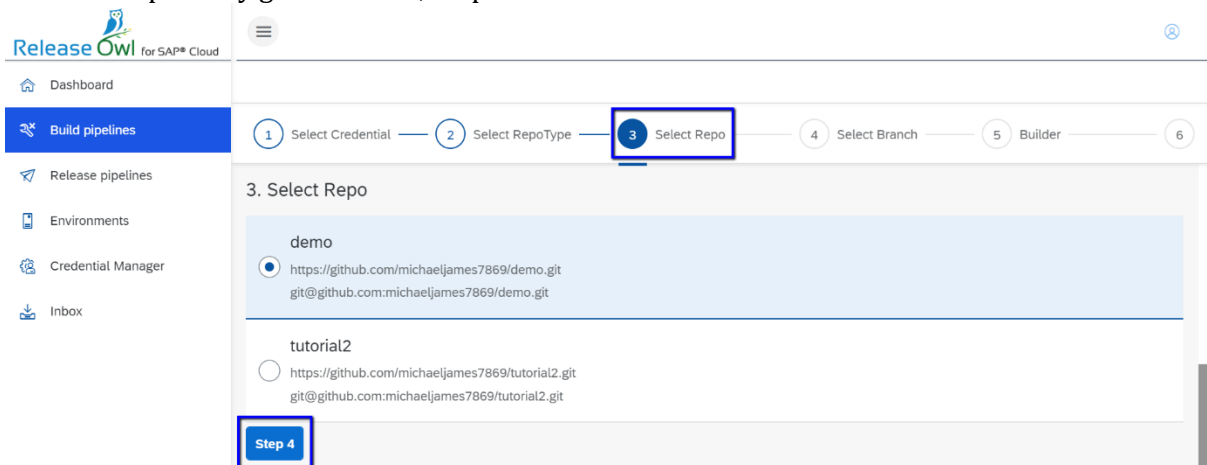
Also, you get to see Step 3 button on selecting one of the available options.



Click on Step3 button. The user now sees the Select Repo screen.

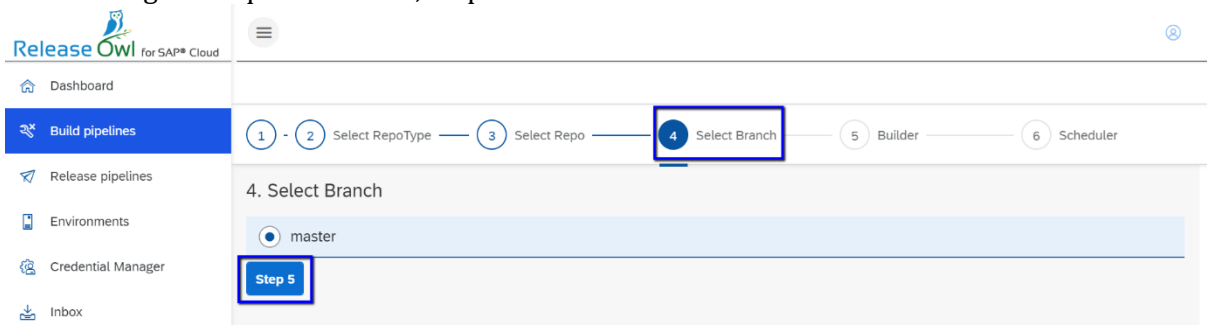
All the repositories associated to the user in the selected version control system are displayed in the Select Repo screen pertaining to the user. Select the required repository.

Once the repository gets selected, Step4 button is seen.



Upon clicking Step4 button, all the branches associated with the repository will be fetched to the user in Select Branch screen.

On selecting the required branch, Step5 button is seen.



Upon clicking Step5 button, the user is shown the kind of Builder. Currently only MTA is supported. The framework is flexible enough to add additional builders as required.



As MTA is the only option available, you see Step6 button.

If more Builder options are available, then only upon selecting one of the available options, you get to see Step6 button.

The screenshot shows the Release Owl interface for SAP Cloud. The left sidebar contains links to Dashboard, Build pipelines, Release pipelines, Environments, Credential Manager, and Inbox. The main area displays a progress bar with steps 1 through 6. Step 5, 'Builder', is the current step, and a 'Step 6' button is visible below the 'MTA' dropdown menu.

On clicking Step6 button, the user is taken to the Scheduler screen. The core capability of CI is scheduling. Currently, three types of schedules are available in Release Owl:

i) Manual - You can build anytime.

The screenshot shows the Release Owl interface for SAP Cloud. The left sidebar contains links to Dashboard, Build pipelines, Release pipelines, Environments, Credential Manager, and Inbox. The main area displays a progress bar with steps 1 through 6. Step 6, 'Scheduler', is the current step, and the 'Manual' option is selected. A 'Review' button is visible at the bottom.

ii) Schedule - You can set a particular time when you want to build.

The screenshot shows the Release Owl interface for SAP Cloud. The left sidebar contains links to Dashboard, Build pipelines, Release pipelines, Environments, Credential Manager, and Inbox. The main area displays a progress bar with steps 1 through 6. Step 6, 'Scheduler', is the current step, and the 'Schedule' option is selected. A 'Review' button is visible at the bottom. The 'Enter Time' field is highlighted, showing a 'Select time' dropdown and a 'HH:mm' input field.



iii) Web Hook - On selecting this option, you get to see the Web Hook URL which you can copy to the Git repository and on commit you can have the pipelines triggered.

There is a Review button at the bottom of this screen.

This screenshot shows the '6. Scheduler' step in a multi-step process. The steps are numbered 1 to 6: 1. Select RepoType, 2. Select Repo, 3. Select Branch, 4. Builder, 5. Scheduler, and 6. Scheduler. The 'WebHook' option is selected with a radio button. Below it, the 'WebHook URL' is displayed as 'https://na2.saparte.com/saparte/webhook/receive'. A 'Review' button is located at the bottom left of the scheduler section.

On clicking the Review button, you can check the details entered in the previous screens. Submit and Cancel buttons are available at the bottom of the page.

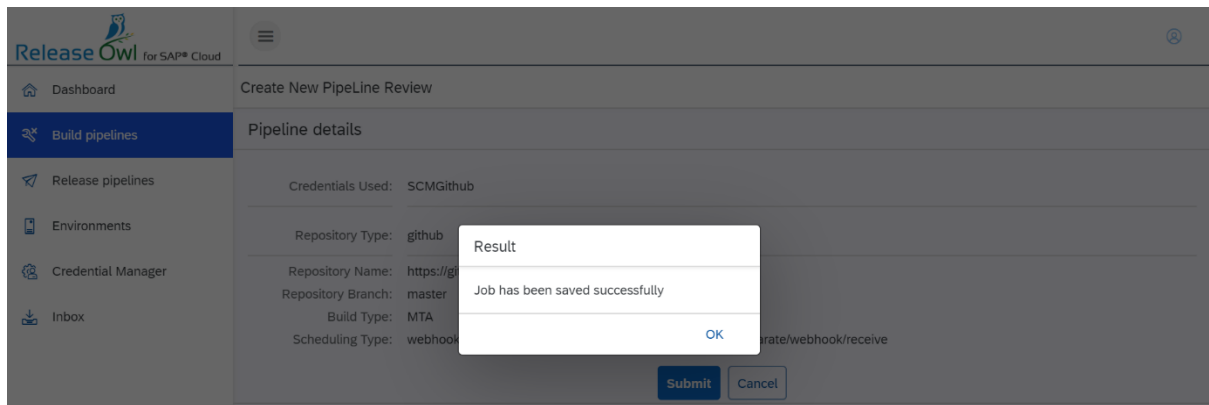
This screenshot shows the 'Create New Pipeline Review' screen. It displays the pipeline details: Credentials Used: SCMGithub, Repository Type: github, Repository Name: https://github.com/michaeljames7869/demo.git, Repository Branch: master, Build Type: MTA, and Scheduling Type: webhook--web hook with URL https://na2.saparte.com/saparte/webhook/receive. At the bottom, there are 'Submit' and 'Cancel' buttons.

Click Submit button. You get a confirmation pop up asking - Are you sure you want to create a New Pipeline? with Yes and No buttons.

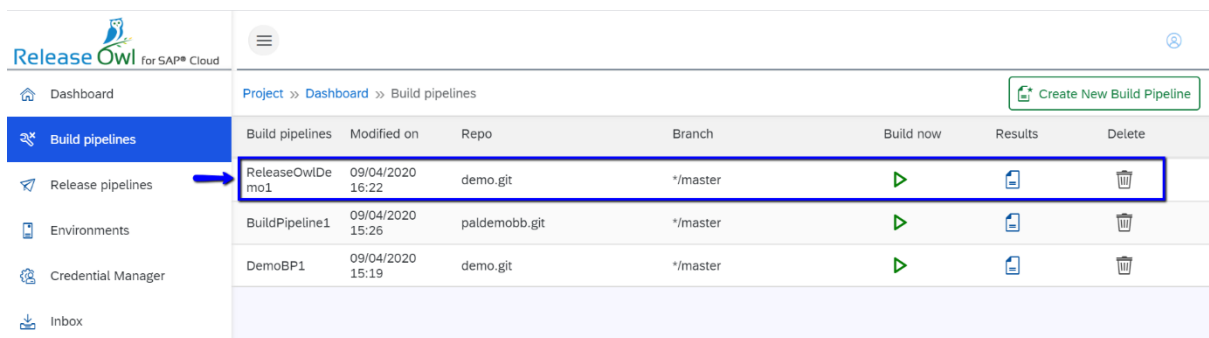
This screenshot shows the same 'Create New Pipeline Review' screen as before, but with a confirmation pop-up dialog box in the center. The dialog box has a title 'Confirmation' and a message 'Are you sure you want to create a new Build Pipeline?'. It has two buttons: 'Yes' and 'No'.




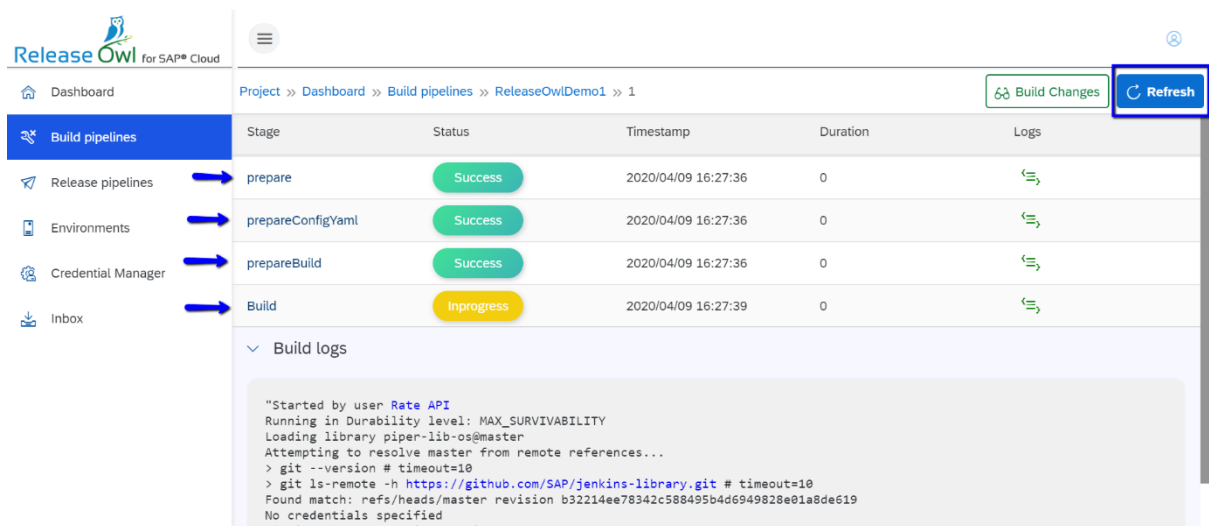
Click Yes button. You get a pop-up window stating Job has been saved successfully with an Ok button.



Click OK button and new build pipeline gets created. The new entry is seen in the Build Pipelines grid.



To run a new pipeline, click on the Run icon  next to that entry. The build pipeline cycle gets triggered and the user will be shown details at different stages of preparing the build configuration file and packaging.






After manual refresh by clicking Refresh button on the top right corner, the status of the individual stages gets updated.

The screenshot shows the Release Owl dashboard for SAP Cloud. The left sidebar contains navigation links: Dashboard, Build pipelines, Release pipelines, Environments, Credential Manager, and Inbox. The main content area displays a table of build pipeline stages for 'ReleaseOwlDemo1'. The stages are: prepare, prepareConfigYaml, prepareBuild, and Build. All stages show a 'Success' status. Below the table, there is a 'Build logs' section with a dropdown arrow. The logs are displayed in a text area, showing the output of the 'prepare' stage.

Stage	Status	Timestamp	Duration	Logs
prepare	Success	2020/04/09 16:27:36	0	⋮
prepareConfigYaml	Success	2020/04/09 16:27:36	0	⋮
prepareBuild	Success	2020/04/09 16:27:36	0	⋮
Build	Success	2020/04/09 16:27:39	1	⋮

Build logs:

```
"Started by user Rate API
Running in Durability level: MAX_SURVIVABILITY
Loading library piper-lib-os@master
Attempting to resolve master from remote references...
> git --version # timeout=10
> git ls-remote -h https://github.com/SAP/jenkins-library.git # timeout=10
Found match: refs/heads/master revision b32214ee78342c588495b4d6949828e01a8de619
No credentials specified
```

The real time logs are seen under Build logs at the bottom. The individual stage logs can be viewed by clicking the log icon  against any of the individual stages. The corresponding log gets displayed.

For e.g., on selecting prepare, you can see the prepare logs as follows:

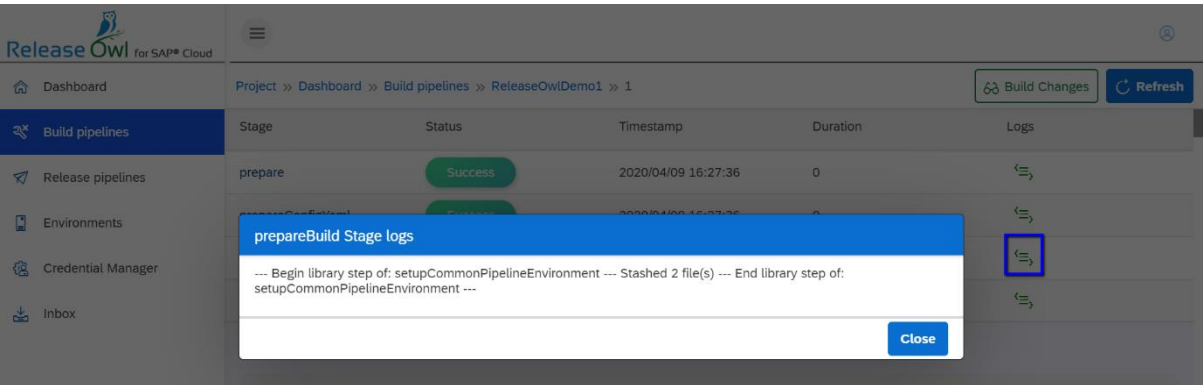
The screenshot shows the Release Owl dashboard with the 'prepare' stage logs displayed. The logs are shown in a text area, detailing the git operations and the successful completion of the stage. The 'prepare Stage logs' section is highlighted with a blue box.

prepare Stage logs

```
using credential 6e13b3da-7959-4955-9852-869741c86b82 Cloning the remote Git repository Cloning repository
https://github.com/michaeljames7869/demo.git > git init /var/jenkins_home/workspace/ReleaseOwlDemo1 # timeout=10
Fetching upstream changes from https://github.com/michaeljames7869/demo.git > git --version # timeout=10 using
GIT_ASKPASS to set credentials SCMGithub1 > git fetch --tags --force --progress
https://github.com/michaeljames7869/demo.git +refs/heads/*:refs/remotes/origin/* > git config remote.origin.url
https://github.com/michaeljames7869/demo.git # timeout=10 > git config --add remote.origin.fetch
+refs/heads/*:refs/remotes/origin/* # timeout=10 > git config remote.origin.url
https://github.com/michaeljames7869/demo.git # timeout=10 Fetching upstream changes from
https://github.com/michaeljames7869/demo.git using GIT_ASKPASS to set credentials SCMGithub1 > git fetch --tags --
force --progress https://github.com/michaeljames7869/demo.git +refs/heads/*:refs/remotes/origin/* > git rev-parse
refs/remotes/origin/master^(commit) # timeout=10 > git rev-parse refs/remotes/origin/master^(commit) #
timeout=10 Checking out Revision d659e1f71d3828181cfa4b65417a54b3fd84ac2a (refs/remotes/origin/master) > git
config core.sparsecheckout # timeout=10 > git checkout -f d659e1f71d3828181cfa4b65417a54b3fd84ac2a Commit
message: "Update README.md" First time build. Skipping changelog.
```

Similarly, on selecting prepareconfigyaml, you can see prepareconfigyaml logs.

On selecting prepareBuild, you can see prepareBuild logs.





Release Owl

for SAP® Cloud

Dashboard

Build pipelines

Release pipelines

Environments

Credential Manager

Inbox

Build Stage logs

```

--- Begin library step of: mtaBuild --- --- Begin library step of: dockerExecute --- + docker ps -q + docker pull
devxvc/mbtci:1.0.10 1.0.10.10: Pulling from devxvc/mbtci Digest:
sha256:a18be0ae27065102ee23b1bf34d89a729af88c5ca299ce9c437c70af6a08f07 Status: Image is up to date for
devxvc/mbtci:1.0.10 docker.io/devxvc/mbtci:1.0.10 + docker inspect -f . devxvc/mbtci:1.0.10 . [INFO] 'mta.yaml' file
found in project sources. + date +%Y%m%d%H%M%S + sed -le $S$(timestamp)/20200409105743/g mta.yaml [INFO]
Executing mta build call: 'mta build --mtar tutorial.mtar --platform cf --target /'. ules/has-unicode: npm WARN enoent
SKIPPING OPTIONAL DEPENDENCY: ENOENT: no such file or directory, rename
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/has-unicode' ->
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/has-unicode.DELETE'
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: inherits@2.0.4
(node_modules/fsevents/node_modules/inherits): npm WARN enoent SKIPPING OPTIONAL DEPENDENCY: ENOENT:
no such file or directory, rename
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/inherits' ->
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/inherits.DELETE' npm
WARN optional SKIPPING OPTIONAL DEPENDENCY: ini@1.3.5 (node_modules/fsevents/node_modules/ini): npm
WARN enoent SKIPPING OPTIONAL DEPENDENCY: ENOENT: no such file or directory, rename
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/ini' ->
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/ini.DELETE' npm WARN
optional SKIPPING OPTIONAL DEPENDENCY: isarray@1.0.0 (node_modules/fsevents/node_modules/isarray): npm
WARN enoent SKIPPING OPTIONAL DEPENDENCY: ENOENT: no such file or directory, rename
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/isarray' ->
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/isarray.DELETE' npm
WARN optional SKIPPING OPTIONAL DEPENDENCY: minimist@1.2.5
(node_modules/fsevents/node_modules/minimist): npm WARN enoent SKIPPING OPTIONAL DEPENDENCY:
ENOENT: no such file or directory, rename
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/minimist' ->
'/var/jenkins_home/workspace/ReleaseOwlDemo1/ui/node_modules/fsevents/node_modules/minimist.DELETE' npm
WARN optional SKIPPING OPTIONAL DEPENDENCY: ms@2.1.2 (node_modules/fsevents/node_modules/ms): npm
WARN enoent SKIPPING OPTIONAL DEPENDENCY: ENOENT: no such file or directory, rename


```

Close

Build Changes

Refresh

Logs


Release Owl for SAP® Cloud

Dashboard

Build pipelines

Release pipelines

Environments

Credential Manager

Inbox

Project » Dashboard » Build pipelines » ReleaseOwlDemo1 » 1

[Build Changes](#)
[Refresh](#)

Stage	Status	Timestamp	Duration	Logs
prepare	Success	2020/04/09 16:27:36	0	Logs
prepareConfigYaml	Success	2020/04/09 16:27:36	0	Logs
prepareBuild	Success	2020/04/09 16:27:36	0	Logs
Build	Success	2020/04/09 16:27:39	1	Logs

One can view the recent changes in the build by clicking the Build Changes button available on the top right corner above the individual build stages.



Release Owl for SAP® Cloud

Dashboard » Build pipelines » ReleaseOwlDemo1 » 1

Build Changes Refresh

Stage	Status	Timestamp	Duration	Logs
prepare	Success	2020/04/09 16:27:36	0	⋮
prepareConfigYaml	Success	2020/04/09 16:27:36	0	⋮
prepareBuild	Success	2020/04/09 16:27:36	0	⋮
Build	Success	2020/04/09 16:27:39	1	⋮

Build logs

```
"Started by user Rote API
Running in Durability level: MAX_SURVIVABILITY
Loading library piper-lib-os@master"
```

Click on the Build Changes button. No change is shown for the very first build of the build pipeline. That is considered the reference for the subsequent builds of the corresponding Build Pipeline.

Release Owl for SAP® Cloud

Project » Dashboard » Build pipelines » ReleaseOwlDemo1 » 1 » Build Changes for ReleaseOwlDemo1, Build 1

No Build Changes

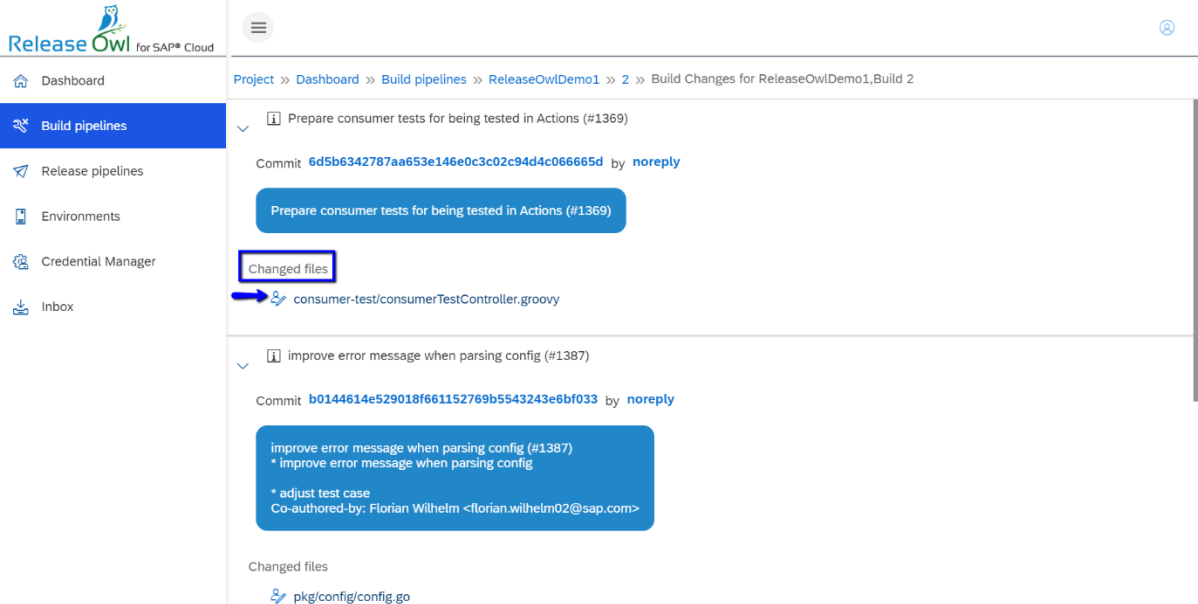
For the subsequent build with any changes made and the corresponding build pipeline is run, the Build Changes screen would show the details as follows:

Release Owl for SAP® Cloud

Project » Dashboard » Build pipelines » ReleaseOwlDemo1 » 2 » Build Changes for ReleaseOwlDemo1, Build 2

- > [i] Prepare consumer tests for being tested in Actions (#1369)
- > [i] improve error message when parsing config (#1387)
- > [i] Hotfix for ServiceKey config parameter (#1377)

Upon expanding the respective trees, we get to see further details about which files are changed and who changed the files etc.



Project » Dashboard » Build pipelines » ReleaseOwlDemo1 » 2 » Build Changes for ReleaseOwlDemo1, Build 2

Prepare consumer tests for being tested in Actions (#1369)

Commit 6d5b6342787aa653e146e0c3c02c94d4c066665d by noreply

Prepare consumer tests for being tested in Actions (#1369)

Changed files

consumer-test/consumerTestController.groovy

improve error message when parsing config (#1387)

Commit b0144614e529018f661152769b5543243e6bf033 by noreply

improve error message when parsing config (#1387)
 * improve error message when parsing config
 * adjust test case
 Co-authored-by: Florian Wilhelm <florian.wilhelm02@sap.com>


Changed files

pkg/config/config.go

Release Pipelines

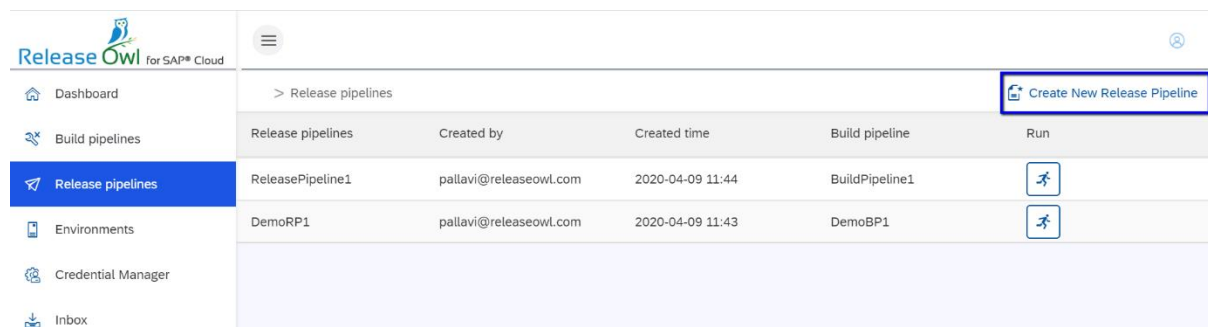
The pre-requisite for creating release pipeline is to setup the cloud environments. You can register the environments in the [Environments](#) section on the left-hand pane.



Click on the Release Pipeline link from the Menu options available on the left-hand pane.

The user gets to see all the details pertaining to the release pipelines that got executed till date on the right-hand pane. Details such as created by, respective build pipeline, and execution option  (Run icon) can be seen.

Creating a New Release Pipeline

On the right-hand corner, you can see Create New Release Pipeline link through which you can create a new release pipeline.



Release pipelines	Created by	Created time	Build pipeline	Run
ReleasePipeline1	pallavi@releaseowl.com	2020-04-09 11:44	BuildPipeline1	
DemoRP1	pallavi@releaseowl.com	2020-04-09 11:43	DemoBP1	

Click on Create New Release Pipeline link and you get to see the following:



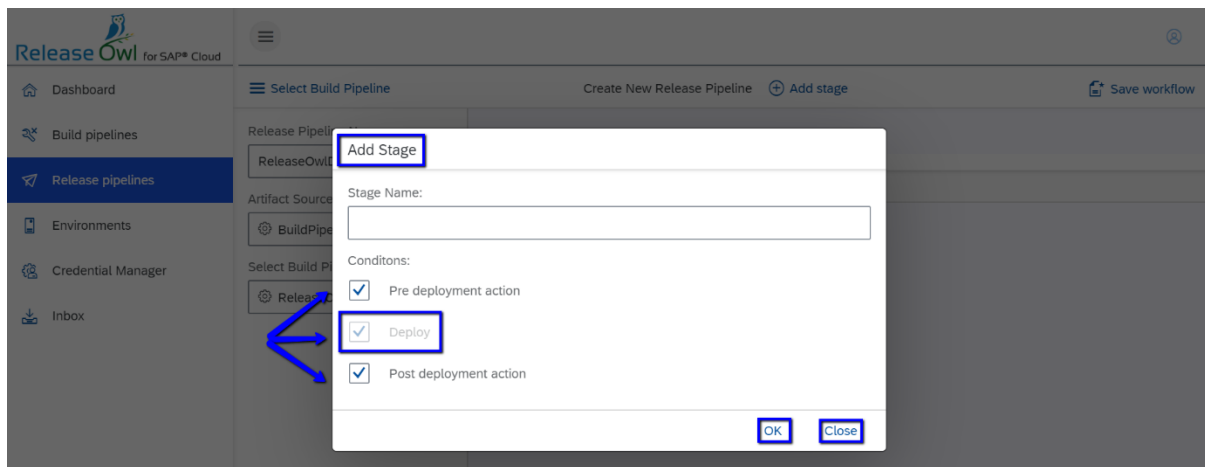
Enter the new Release Pipeline Name that is to be created against Release Pipeline Name field. Currently the artefact source is Build Pipeline. You can include few more options like GIT or Bit Bucket in the near future. You can select the reference Build Pipeline for creating the new Release Pipeline from Select Build Pipeline list box.

Then click on Add stage link available.

You get to see Add Stage screen. Enter the stage on (environment) such as UAT or QA and check the options – Pre-Deployment action or Post-Deployment action based on the need such as any approvals or configuration changes to be done before or after the deployment respectively.

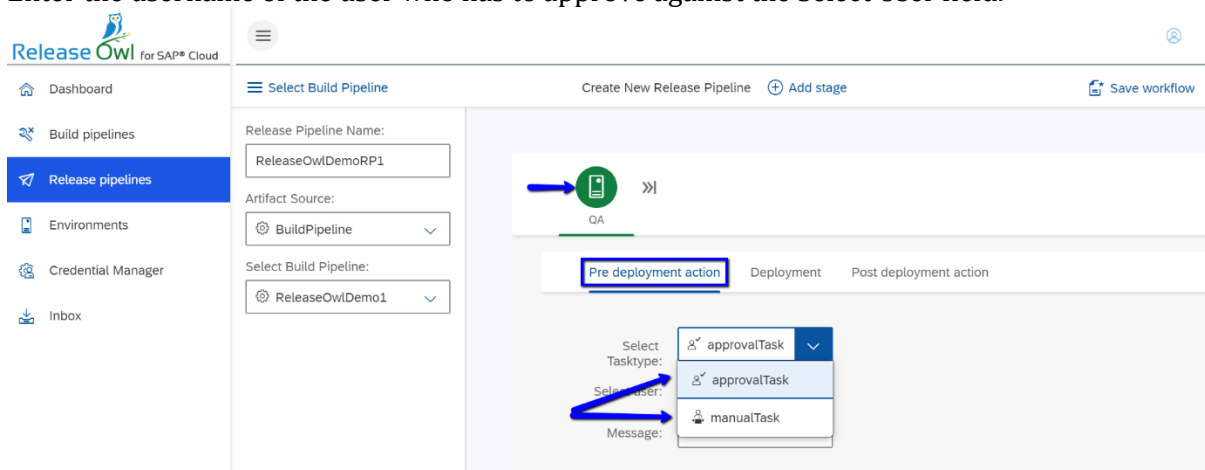
All the three – Pre-Deployment Action, Deploy, Post-Deployment action options will be checked by default.

Deploy option will appear disabled and checked by default in order to complete the deployment process as it is for this purpose that you create a new release pipeline.



On clicking OK, the user is taken to the Pre-Deployment action screen where one can select the pre-deployment task that is needed - here either approval task for any approval by any manager or manual task in case any configuration setting change needed to be done manually by the assignee. Here let us select Approve Task option.

Enter the username of the user who has to approve against the Select User field.



Select approval task from the Select Task Type screen and enter the task assignee username against Select User field.

Enter details regarding what action is to be performed by the assignee against Message field.



The screenshot shows the Release Owl interface for configuring a release pipeline. The left sidebar contains navigation links: Dashboard, Build pipelines, Release pipelines (highlighted), Environments, Credential Manager, and Inbox. The main area is titled 'Select Build Pipeline' and includes fields for 'Release Pipeline Name' (ReleaseOwlDemoRP1), 'Artifact Source' (BuildPipeline), and 'Select Build Pipeline' (ReleaseOwlDemo1). The 'Pre deployment action' tab is selected, showing configuration for a task named 'QA'. The task type is 'approvalTask', the user is 'userqatest11@g...', and the message is 'Submission for Approvz'. Blue arrows point to the task type, user, and message fields.

Then go to Deployment screen by clicking on Deployment link. The following screen is displayed:

The screenshot shows the Release Owl interface with the 'Deployment' tab selected. The configuration for the 'QA' task is shown, with the 'Select environment' dropdown set to 'QAEnv'. A blue arrow points to the 'Select environment' field.

You can add any number of environments as and when required. The Deployment option here is an automated task wherein the .MTA file gets deployed.

Click on Post Deployment action link available. The Post Deployment Action screen is displayed as follows:

The screenshot shows the Release Owl interface with the 'Post deployment action' tab selected. The configuration for the 'QA' task is shown, with the task type set to 'manualTask'. The 'Select user' dropdown is open, showing 'approvalTask' and 'manualTask' options. Blue arrows point to the 'manualTask' option in the dropdown.



The user can select the post-deployment task that is needed - here either approval task for any approval by any manager post-deployment or any manual task requiring any change in the settings needed to be done manually by the assignee.

Select manual task from the Select Task Type screen and enter the task assignee username against Select User field.

Enter the username of the user who has to perform the manual task post deployment against the Select User field.

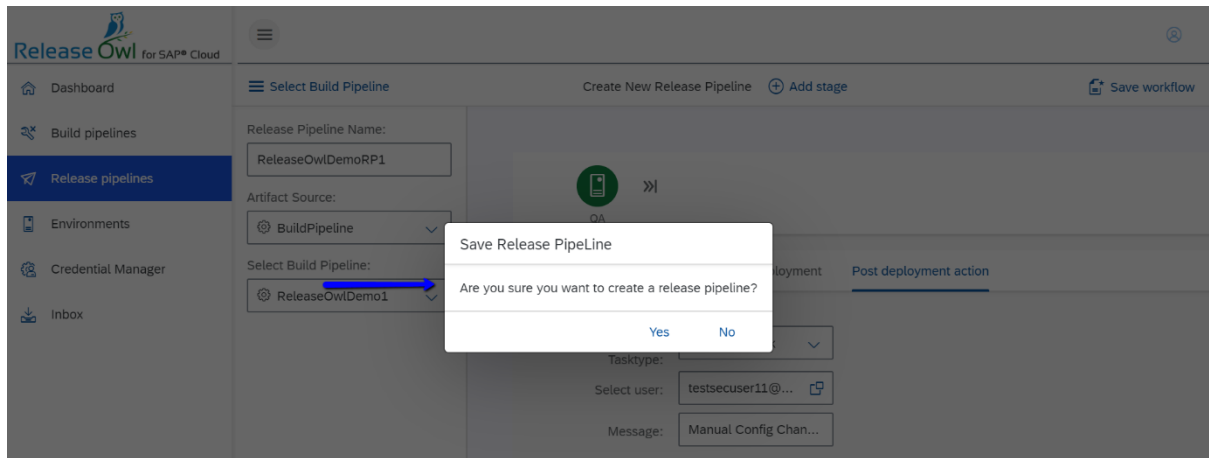
Enter details regarding what action is to be performed by the assignee against Message field.

This screenshot shows the 'Select Build Pipeline' configuration page in the Release Owl interface. The left sidebar contains navigation links: Dashboard, Build pipelines, Release pipelines (highlighted), Environments, Credential Manager, and Inbox. The main content area is titled 'Select Build Pipeline' and includes fields for 'Release Pipeline Name' (ReleaseOwlDemoRP1), 'Artifact Source' (BuildPipeline), and 'Select Build Pipeline' (ReleaseOwlDemo1). On the right, there's a 'QA' stage configuration section with tabs for 'Pre deployment action', 'Deployment', and 'Post deployment action' (which is selected and highlighted with a blue box). Below the tabs, there are three fields: 'Select Tasktype:' with a dropdown menu showing 'manualTask', 'Select user:' with a text field containing 'testsecuser11@...', and 'Message:' with a text field containing 'Manual Config Change'. A blue double-headed arrow points from the 'manualTask' dropdown to the 'Select user:' field.

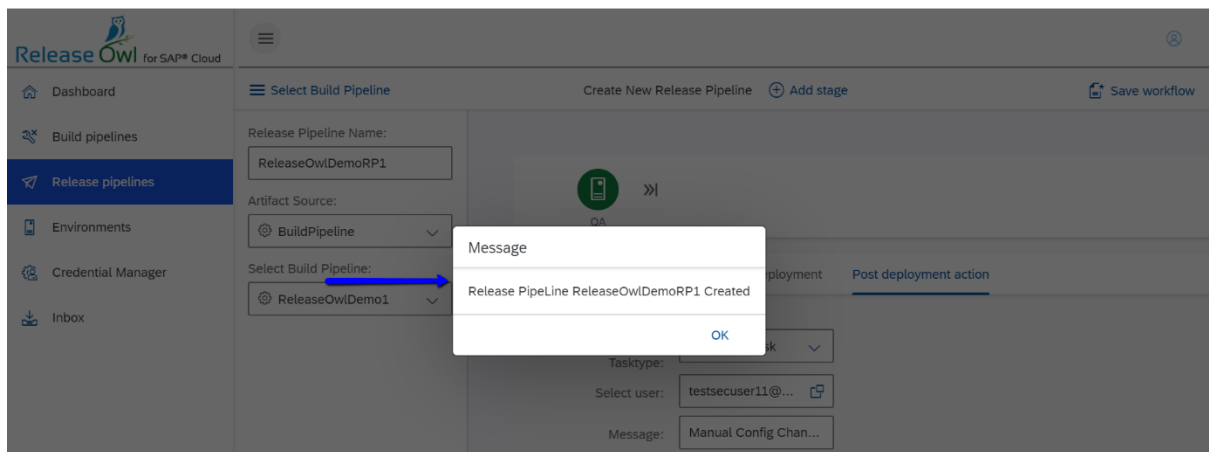
Click on Save Workflow link.

This screenshot shows the same 'Select Build Pipeline' configuration page as the previous one, but with the 'Save workflow' button in the top right corner highlighted with a blue box. The 'Post deployment action' tab is still selected, and the configuration fields remain the same.

You get a confirmation popup window as follows:



Click Yes.



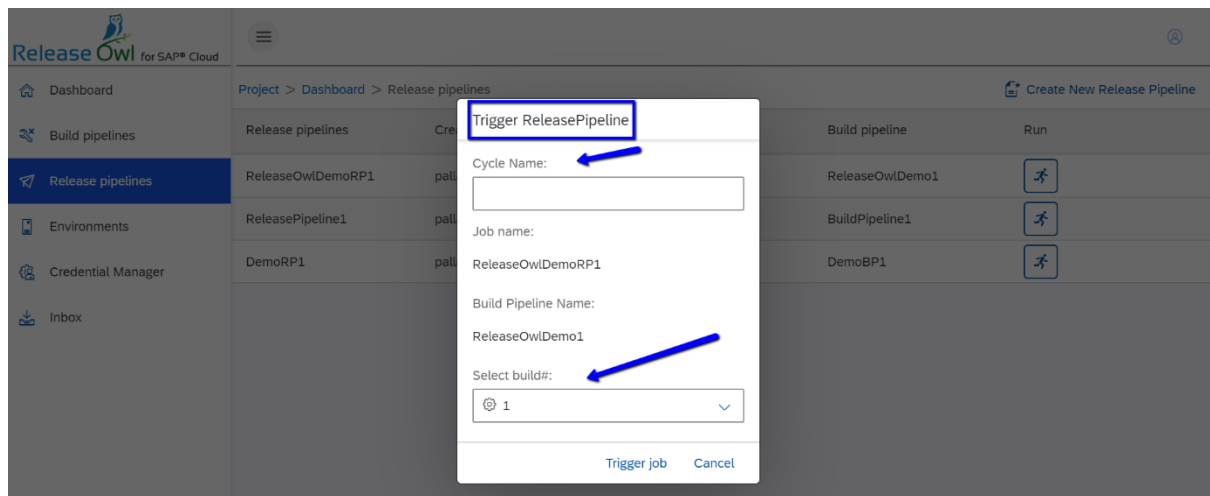
Click OK.

The user will be taken to the Release Pipelines screen and the newly created release pipeline will be seen as the very first entry on the right hand pane as it is the most recent one that is created.

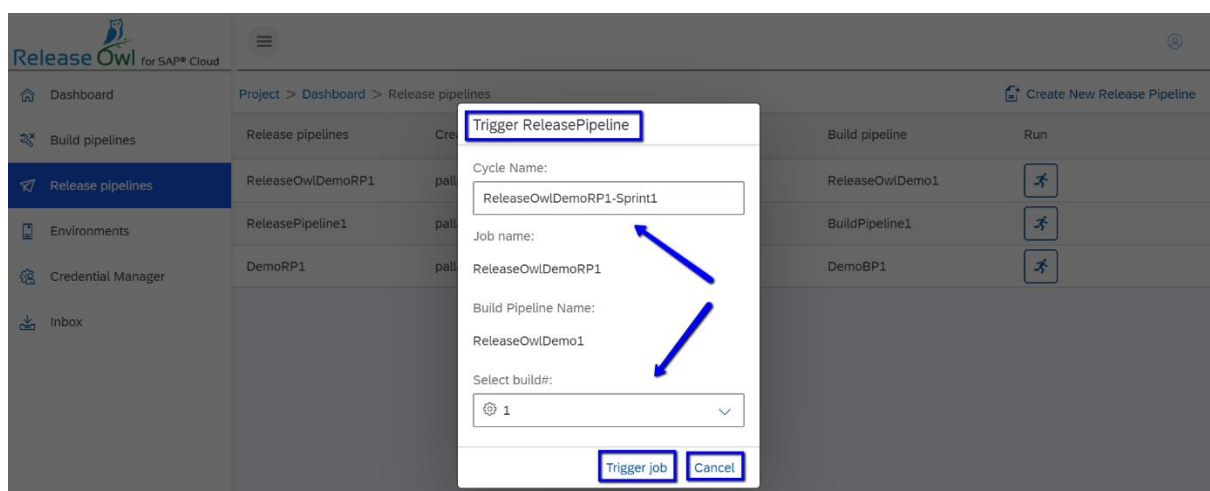
Release pipelines	Created by	Created time	Build pipeline	Run
ReleaseOwlDemoRP1	pallavi@releaseowl.com	2020-04-09 12:12	ReleaseOwlDemo1	
ReleasePipeline1	pallavi@releaseowl.com	2020-04-09 11:44	BuildPipeline1	
DemoRP1	pallavi@releaseowl.com	2020-04-09 11:43	DemoBP1	



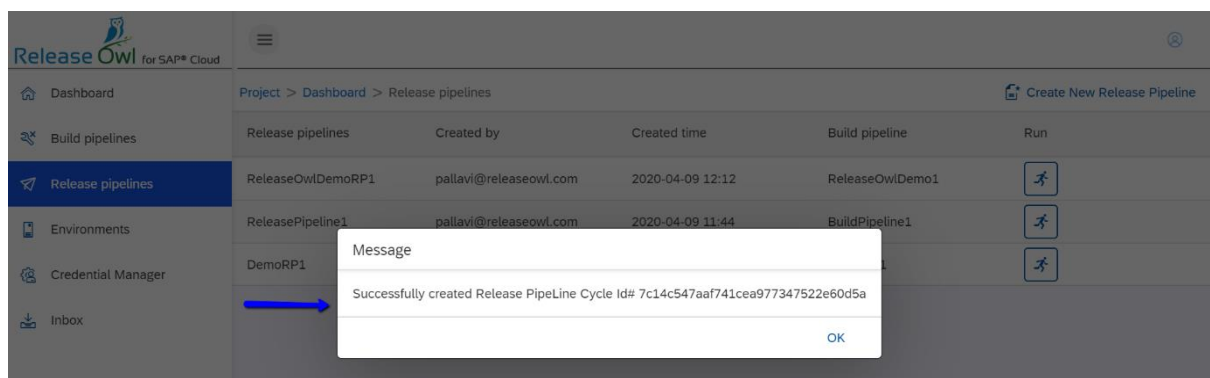
Click on the Run icon against the newly release pipeline entry to execute the corresponding release pipeline. Trigger Release Pipeline window gets displayed.



Enter relevant Cycle Name and select the appropriate build number the artefacts of which are to be released to the specified environment. By default, the latest build number is picked up by the system.



Click on the Trigger Job button. We get a popup window upon successful creation of the Release Pipeline Cycle.





Click OK. The user would be shown the Pre-Condition Deployment task that is to be done for the deployment to take place.

Release Owl for SAP® Cloud

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1 > 7c14c547aaf741cea977347522e60d5a

QA

Pre Condition - QA

Assigned To: userqatest11@gmail.com

Task Type: Approval task

Task Details: Submission for Approval

Status: WAITING

Approve

Reject

StartTime: 2020-04-09 12:22

EndTime: 2020-04-09 12:22

This pending task of approval is to be completed by the assignee seen in the above screen. Now when the assigned user (userqatest11@gmail.com) logs in to the application and checks his inbox, he can see the tasks that are pending for his approval.

Release Owl for SAP® Cloud

Inbox

Release Pipeline Name	Cycle name	Stage name	Label	Task type	Task Description	Action
ReleaseOwlDemoRP1	ReleaseOwlDemoRP1-Sprint1	QA	Pre Condition - QA	Approval task	Submission for Approval	Approve / Reject

On clicking Reject, the release pipeline cycle gets aborted.

On clicking Accept, we get a pop up as follows:

Release Owl for SAP® Cloud

Inbox

Release Pipeline Name	Cycle name	Stage name	Label	Task type	Task Description	Action
ReleaseOwlDemoRP1	ReleaseOwlDemoRP1-Sprint1	QA	Pre Condition - QA	Approval task	Submission for Approval	Approve / Reject

Message

Release PipeLine Stage ReleaseOwlDemoRP1 Approved !

OK

On clicking OK, the pending task that was seen earlier in the inbox earlier, now gets cleared.



Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

Inbox

Release Pipeline Name	Cycle name	Stage name	Label	Task type	Task Description	Action
No data						

Now log in to the app as the user who created the release pipeline. Go to Release Pipeline section and double click the Release Pipeline that is being run. Deployment status will be seen as Waiting which means it is in progress.

Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1

Release Pipeline	Environment	Build Pipeline Ref	Status
ReleaseOwlDemoRP1-Sprint1	PalQA	ReleaseOwlDemo1	Waiting

Double click the entry shown above. Pre-Deployment Condition is met and is now Complete.

The next stage is Deployment. The status of Deployment would now be Started.

Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1 > 0465aaba8e142d5a3c4eb661a61d327

QA

Pre Condition - QA

Assigned To: userqatest11@gmail.com
Task Type: Approval task
Task Details: Submission for Approval
Status: COMPLETED

Start Time: 2020-04-10 09:40
End Time: 2020-04-10 09:40

Deploy Task - QA

buildJobName: ReleaseOwlDemo1
buildJobNumber: 1
Status: STARTED
Logs

Start Time: 2020-04-10 09:40
End Time:

Upon manual refresh, the status gets updated to In Progress.



Release Owl for SAP® Cloud

Dashboard | Build pipelines | **Release pipelines** | Environments | Credential Manager | Inbox

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1 > 0465aaba8e142d5a3c4eb661a61d327

QA

Pre Condition - QA

Assigned To: userqatest11@gmail.com
Task Type: Approval task
Task Details: Submission for Approval
Status: COMPLETED

Deploy Task - QA

buildJobName: ReleaseOwlDemo1
buildJobNumber: 1
Status: IN PROGRESS
Logs

StartTime: 2020-04-10 09:40
EndTime: 2020-04-10 09:40

The user can view the deployment logs at any point by clicking the Logs link available.

Stage Logs

Deploy task Build Logs

```
"Started by user Rate API
Running in Durability level: MAX_SURVIVABILITY
Loading library piper-lib-os@master
Attempting to resolve master from remote references...
> git --version # timeout=10
> git ls-remote -h https://github.com/SAP/jenkins-library.git # timeout=10
Found match: refs/heads/master revision 66636d68f9f3759409b001dc09a0eb215be54eaa
No credentials specified
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/SAP/jenkins-library.git
> git init /var/jenkins_home/workspace/ReleaseOwlDemoRP1_QA_10062_11039@libs/piper-lib-os # timeout=10
Fetching upstream changes from https://github.com/SAP/jenkins-library.git
> git --version # timeout=10
> git fetch --no-tags --force --progress https://github.com/SAP/jenkins-library.git +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url https://github.com/SAP/jenkins-library.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/SAP/jenkins-library.git # timeout=10
Fetching without tags
Fetching upstream changes from https://github.com/SAP/jenkins-library.git
> git fetch --no-tags --force --progress https://github.com/SAP/jenkins-library.git +refs/heads/*:refs/remotes/origin/*
Checking out Revision 66636d68f9f3759409b001dc09a0eb215be54eaa (master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 66636d68f9f3759409b001dc09a0eb215be54eaa
Commit message: "\"Hotfix for ServiceKey config parameter (#1377)\""
```

OK

After a couple of minutes, upon refresh, the status of Deploy Task changes to Completed and Post-Deployment task awaits action from the assignee.



Release Owl for SAP® Cloud

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1 > 0465aaba8e142d5a3c4eb661a61d327 Refresh

Assigned To: userqatest11@gmail.com
Task Type: Approval task
Task Details: Submission for Approval
Status: COMPLETED
StartTime: 2020-04-10 09:40
EndTime: 2020-04-10 09:40

Deploy Task - QA
buildJobName: ReleaseOwlDemo1
buildJobNumber: 1
Status: COMPLETED
Logs

Post Condition - QA
Assigned To: testsecuser11@gmail.com
Task Type: Manual task
Task Details: Manual Config Change Needed
Status: WAITING
Approve
Reject

Now when the assigned user (testsecuser11@gmail.com) logs in to the application and checks his inbox, he can see the tasks that are pending for his action.

Release Owl for SAP® Cloud

Inbox

Release Pipeline Name	Cycle name	Stage name	Label	Task type	Task Description	Action
ReleaseOwlDemoRP1	ReleaseOwlDemoRP1-Sprint1	QA	Post Condition - QA	Manual task	Manual Config Change Needed	Complete

The assigned user should make the necessary change manually and click Complete.

Release Owl for SAP® Cloud

Inbox

Release Pipeline Name	Cycle name	Stage name	Label	Task type	Task Description	Action
ReleaseOwlDemoRP1	ReleaseOwlDemoRP1-Sprint1	QA	Post Condition - QA	Manual task	Manual Config Change Needed	Complete

Message
Release PipeLine Stage ReleaseOwlDemoRP1 Completed !
OK



Click OK. The pending task that was seen earlier in the inbox earlier, now gets cleared.

Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

Inbox

Release Pipeline Name	Cycle name	Stage name	Label	Task type	Task Description	Action
No data						

Now log in to the app as the user who created the release pipeline. Go to Release Pipeline section and the status is now seen as Completed because the post deployment condition is fulfilled.

Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1

Release Pipeline	Environment	Build Pipeline Ref	Status
ReleaseOwlDemoRP1-Sprint1	PalQA	ReleaseOwlDemo1	Completed

Double click the Release Pipeline that is being run. Post Deployment Task status will now be seen as Completed.

Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

Project > Dashboard > Release pipelines > ReleaseOwlDemoRP1 > 0465aaba8e142d5a3c4eb661a61d327

Pre Condition - QA

Assigned To: userqatest11@gmail.com
Task Type: Approval task
Task Details: Submission for Approval
Status: COMPLETED

Deploy Task - QA

buildJobName: ReleaseOwlDemo1
buildJobNumber: 1
Status: COMPLETED
Logs

Post Condition - QA

Assigned To: testsecuser11@gmail.com
Task Type: Manual task
Task Details: Manual Config Change Needed
Status: COMPLETED

All the recent Build and Release Pipelines executed can be seen in the Dashboard.



Release Owl for SAP® Cloud

Project » Dashboard

Recent Build Cycles

Build Pipeline	Build #	Repo	Branch	Result
ReleaseOwlDemo1	1	demo.git	master	Success
DemoBP1	1	demo.git	master	Success

Recent Release Cycles

Release Pipeline	Environment	Build Pipeline Ref	Status
ReleaseOwlDemoRP1-Sprint1	PalQA	ReleaseOwlDemo1	Completed
ReleaseOwlDemoRP1-Sprint1	QAEV	ReleaseOwlDemo1	Failed

Environments

On clicking Environments menu option on the left-hand pane, we get to see the list of environments where we can deploy the SCM code war/mtar in Cloud Foundry. We can register new environment by clicking on the link Register SAP Cloud Environment available on the right-hand pane.

Release Owl for SAP® Cloud

List of Environments

Register SAP Cloud Environment

PalEnvforIncCFURL	https://cockpit.hanatrial.ondemand.com/cockpit/ https://api.cf.us10.hana.ondemand.com	pallavi@releaseowl.com 27/03/2020 11:46
QAEV	https://cockpit.hanatrial.ondemand.com/cockpit/ https://api.cf.us10.hana.ondemand.com	pallavi@releaseowl.com 10/03/2020 13:02
PalQA	https://cockpit.hanatrial.ondemand.com/cockpit/ https://api.cf.us10.hana.ondemand.com	pallavi@releaseowl.com 26/03/2020 12:54
PalEnvforIncCred	https://cockpit.hanatrial.ondemand.com/cockpit/ https://api.cf.us10.hana.ondemand.com	pallavi@releaseowl.com 27/03/2020 11:36

Click on Register SAP Cloud Environment link. We get to see the following window.

This screenshot shows the Release Owl web application interface. On the left is a sidebar with navigation links: Dashboard, Build pipelines, Release pipelines, Environments (highlighted), Credential Manager, and Inbox. The main area is titled "List of Environments" and contains a modal window titled "Edit/Create Environment". The modal has several input fields: "Name" with a placeholder "only alphanumerics allowed", "Cloudfoundry URL", "Space", "Org", "API Endpoint", and "Credential key" with a dropdown arrow. At the bottom of the modal are "Save" and "Cancel" buttons. On the right side of the main area, there is a "Register SAP Cloud Environment" button and a list of existing environments with columns for email, name, and timestamp.

Enter the values in the fields and click Save.

This screenshot shows the same Release Owl interface as the previous one, but with the "Edit/Create Environment" modal filled out with specific values. The "Name" field contains "PalQAEV", "Cloudfoundry URL" contains "https://cockpit.hanatrial.ondemand.com/cockpit/", "Space" contains "dev", "Org" contains "p2000454655trial", "API Endpoint" contains "https://api.cf.us10.hana.ondemand.com", and the "Credential key" dropdown is set to "PalQA". The "Save" and "Cancel" buttons are still at the bottom of the modal. The background shows the "List of Environments" section with a table containing one entry named "DemoEnv" with two URLs.

The values to be entered in the above screen can be seen in the cockpit as shown below. Also ensure to select your credential to login to the cockpit that you have registered in the Credential Manager.

cockpit.hanatrial.ondemand.com/cockpit#/globalaccount/55223e15-f2f3-43c0-8a74-8229be7f9971/subaccount/8684fa28-4fdb-409b-b48d-d363e5a6a...

SAP Cloud Platform Cockpit Join our user survey! 30 Pavani Pallavi

Subaccount: trial - Overview

[Delete Subaccount](#)

Subaccount Details

Subdomain: p2000454655trial

ID: 8684fa28-4fdb-409b-b48d-d363e5a6a0d4

Cloud Foundry

Org Name: p2000454655trial

Org ID: 3d9b8ac2-2a9b-4e03-9794-089ba8edb9f0

Members: 1

API Endpoint: https://api.cf.us10.hana.ondemand.com

[Disable Cloud Foundry](#)

Spaces (1) [Create Space](#)

Name	Applications	Service Instances
dev	0	1

Make sure the space (here dev) is assigned to a quota plan (lite).

If there is no plan assigned, create a plan by name lite by going to the Quota Plans.

SAP Cloud Platform Cockpit Join our user survey! 30 Pavani Pallavi

Subaccount: trial - Quota Plans

All: 1 plan, 1 resource

[New Plan](#)

Create Space Quota Plan

The org. quota limit is applicable for a resource if you do not enter a space quota limit. If the space quota limit for a resource exceeds the org. quota limit, the org. quota limit applies.

*Name:

*Memory (MB): Org. Quota: 4 GB

Routes: Org. Quota: 10

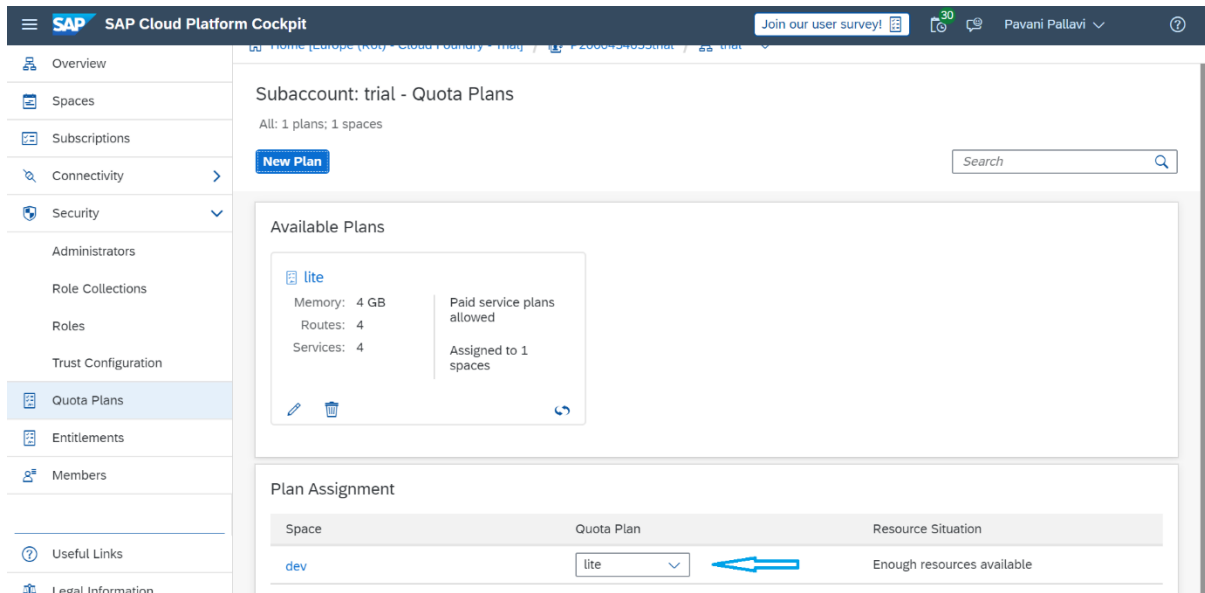
Services: Org. Quota: 40

Instance Memory (MB): Org. Quota: 8 GB

App Instances: Org. Quota: 20

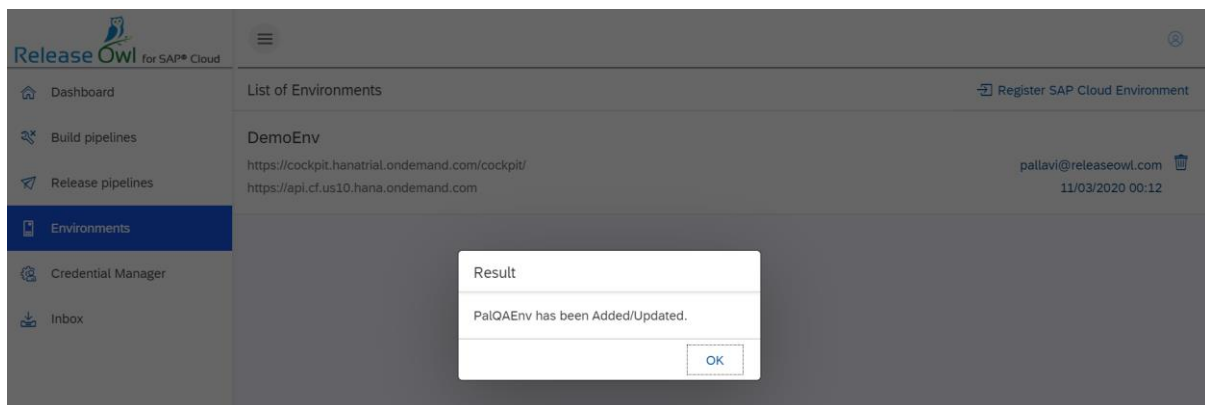
Allow Paid Services: ☒ Org. Quota allows paid plans.

[Save](#) [Cancel](#)



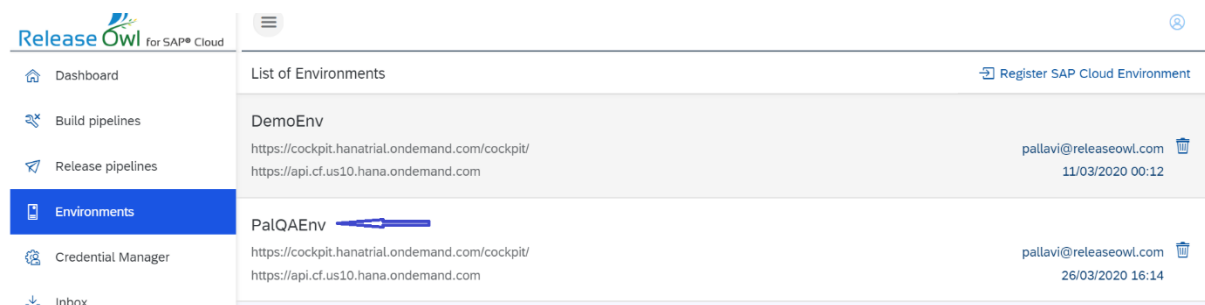
The screenshot shows the SAP Cloud Platform Cockpit interface. On the left is a navigation menu with options like Overview, Spaces, Subscriptions, Connectivity, Security, Administrators, Role Collections, Roles, Trust Configuration, Quota Plans (selected), Entitlements, and Members. The main area is titled 'Subaccount: trial - Quota Plans' and shows 'All: 1 plans; 1 spaces'. There is a 'New Plan' button and a search bar. Under 'Available Plans', a 'lite' plan is shown with details: Memory: 4 GB, Routes: 4, Services: 4, and 'Paid service plans allowed' and 'Assigned to 1 spaces'. Below this is a 'Plan Assignment' table with columns 'Space', 'Quota Plan', and 'Resource Situation'. The table has one row for space 'dev' with 'lite' selected in the 'Quota Plan' column and 'Enough resources available' in the 'Resource Situation' column. A blue arrow points to the 'lite' dropdown.

Click OK.



The screenshot shows the Release Owl for SAP Cloud interface. The left navigation menu includes Dashboard, Build pipelines, Release pipelines, Environments (selected), Credential Manager, and Inbox. The main area is titled 'List of Environments' and shows a table with one entry: 'DemoEnv' with two URLs and the user 'pallavi@releaseowl.com' with a timestamp '11/03/2020 00:12'. A modal popup titled 'Result' is displayed in the center, containing the message 'PalQAEEnv has been Added/Updated.' and an 'OK' button.

The created environment gets listed under the List of Environments as shown below.



The screenshot shows the Release Owl for SAP Cloud interface with the 'Environments' menu selected. The 'List of Environments' table now has two entries. The first is 'DemoEnv' with the same details as before. The second entry is 'PalQAEEnv' with a blue pencil icon for editing, the same two URLs, the user 'pallavi@releaseowl.com', and a timestamp '26/03/2020 16:14'.

Click Save. We get a popup as follows:

In order to edit the already existing environment, double click the corresponding environment under the list of environments available on the right-hand pane. We get to see the details pertaining to that environment which we can edit and save by clicking Save button.



Credential Manager

There are a number of third-party sites and applications that can interact with Release Owl, like artefact repositories, cloud-based storage systems, services etc. The admin can configure credentials in the application for use by Release Owl. Once the user adds/configures the credentials in Release Owl, the credentials can be used by Pipeline projects to interact with the 3rd party applications.

We need a SCM username/password to check out the source code to build. Credential can be of GitHub repository / Cloud Foundry / any source code management credential.

On clicking Credential Manager menu option on the left-hand pane, we get to see the list of credentials that are registered. We can register new credential by clicking on the link Register Credential available on the right-hand pane.

We can register new credential by clicking on the link Register Credential available on the right-hand pane.



Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

List of credentials

Register Credential

SCMGithub
PalSCMBB
PalCPCred
PalCockpit
QAEEnvCred

Scope: GLOBAL

Credential Name: PalCPCred

User Name: pallavi.pavani@gmail.com

password: *****

Save Cancel

Enter the Scope, Credential Name, Username and password and click Save.

Note: Global scope is the default scope. If we want credentials to be generally available to jobs, we should use Global scope

On clicking Save, we get the following pop up:

Release Owl for SAP® Cloud

Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

List of credentials

Register Credential

SCMGithub
PalSCMBB
PalCockpit
QAEEnvCred
PalCPCred

Result

PalCPCred Credential Created

OK

Click Ok. The new credential gets added and can be seen under List of Credentials.

Release Owl for SAP® Cloud

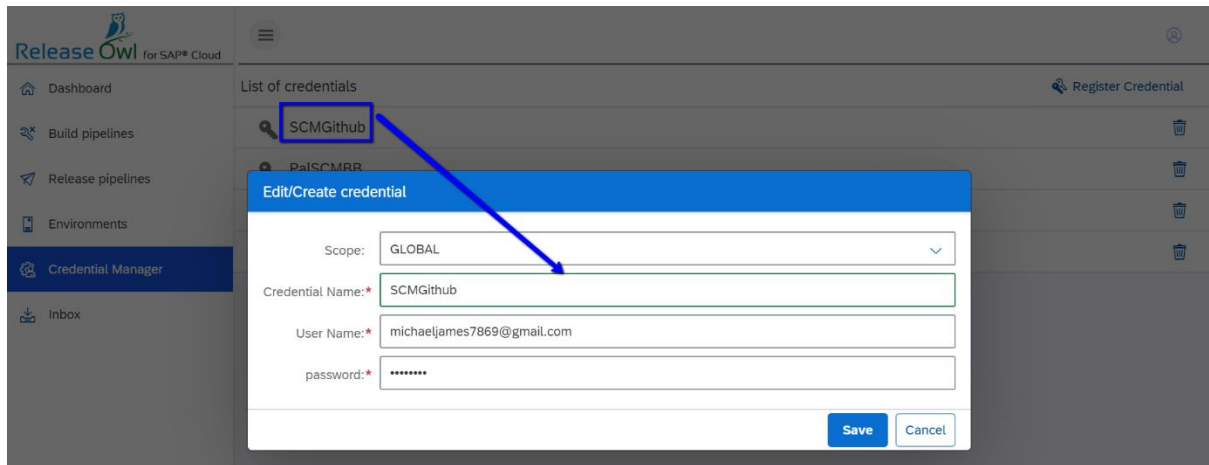
Dashboard
Build pipelines
Release pipelines
Environments
Credential Manager
Inbox

List of credentials

Register Credential

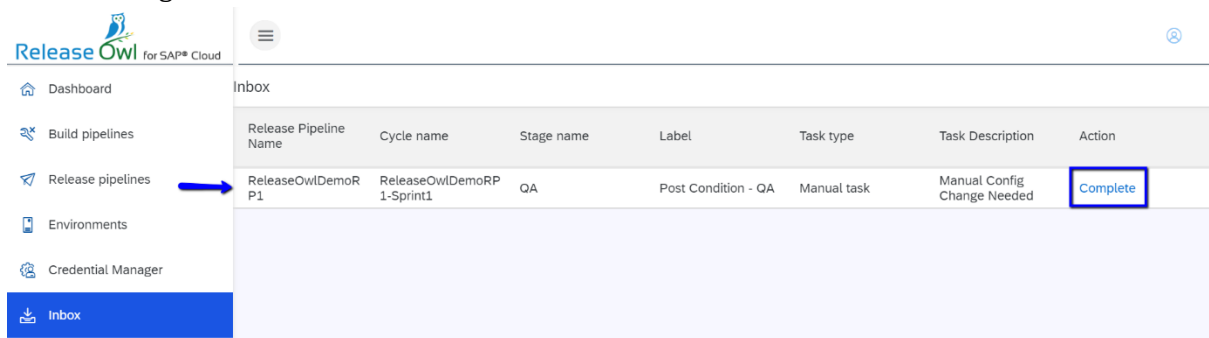
SCMGithub
PalSCMBB
QAEEnvCred
PalCPCred

Similarly, on double clicking any available credential, we can edit the details and save them.



Inbox

The user can check his inbox for any pending action items that are assigned to him say for e.g., approving the deployment or manually changing the configuration settings or any such tasks that are assigned to him.



Once the tasks are done, the user can no more see them in his inbox.