workato

Automation Accelerators

CI/CD Automation with MS Teams

Admin Guide

Version 2.0

Table of Contents

Overview

Continuous Integration and Continuous Delivery or CI/CD is a practice to deliver quality automations reliably and securely.

The goal of CI/CD accelerator is to automate the way teams build, test, and package their automations. It helps automate the delivery to multiple environments such as test and production. Automating release processes makes it easier for teams to coordinate deployments, run quality controls on release, and reduce errors.

The CI/CD automation accelerator drives consistent release practices across multiple teams in an organization. Once the release processes are automated, teams can build better solutions and reduce change defect rates.

Using a standardized and automated approach, organizations can:

- Introduce new services faster.
- Increase the business value through improved quality of service.
- Enhance production environment stability by reducing post-implementation incident spikes.

Note: Workato Accelerators are not formally supported beyond the standard platform support provided as part of your existing Workato agreement. Accelerators are intended as a starting point for your solution development and can be modified/customized to your organization's needs. You may reach out to your Workato customer success team to provide feedback on this Accelerator or email accelerators.feedback@workato.com.

Dependencies

1. CI/CD Automation version 2 is built for use with Workato Environments feature. If you use multiple workspaces as environments, please request CI/CD Automation version 1.

1. Once installed, this accelerator includes 5 active recipes. This is in addition to any recipes you may create to customize the accelerator.

2. A GitHub account and repositories.

3. Microsoft Teams workspace with access to Workbot for Microsoft Teams from Workato.

Installing The Accelerator

The following high-level steps are required to install this accelerator. Details for each step are covered in the following sections.

1. Install the package.
2. Connect to Workato environments.
3. Setup GitHub repositories.
4. Setup Microsoft Teams workspace.
5. Start the recipes.

CI/CD Automation Accelerator Assets

The accelerator is provided in a zip package which contains the following assets:

1. Two connector SDK
    a. CICD for Workato
    b. CICD for GitHub
2. Nine recipes
    a. 4 Workbot for Microsoft Teams recipe
    b. 5 recipe functions
    c. 1 standard recipe
3. Two lookup tables
    a. Project configuration
    b. Audit logs
4. Three connections
    a. Workato accounts
    b. GitHub account
    c. Workbot for Microsoft Teams
5. One account property
    a. Group ID

Installing the package

1. Create a project/folder to import the package. This guide assumes a project named "**CICD Automation**" is used for importing the package.
2. Import the package **CICD-Automation-MSTeams-x.y.z.zip** (where x.y.z is the version number) using the Recipe Lifecycle Management tool.
    a. For detailed steps please refer to the docs.
    b. Lookup Table Data import - select ignore data.

Connect to Workato environments

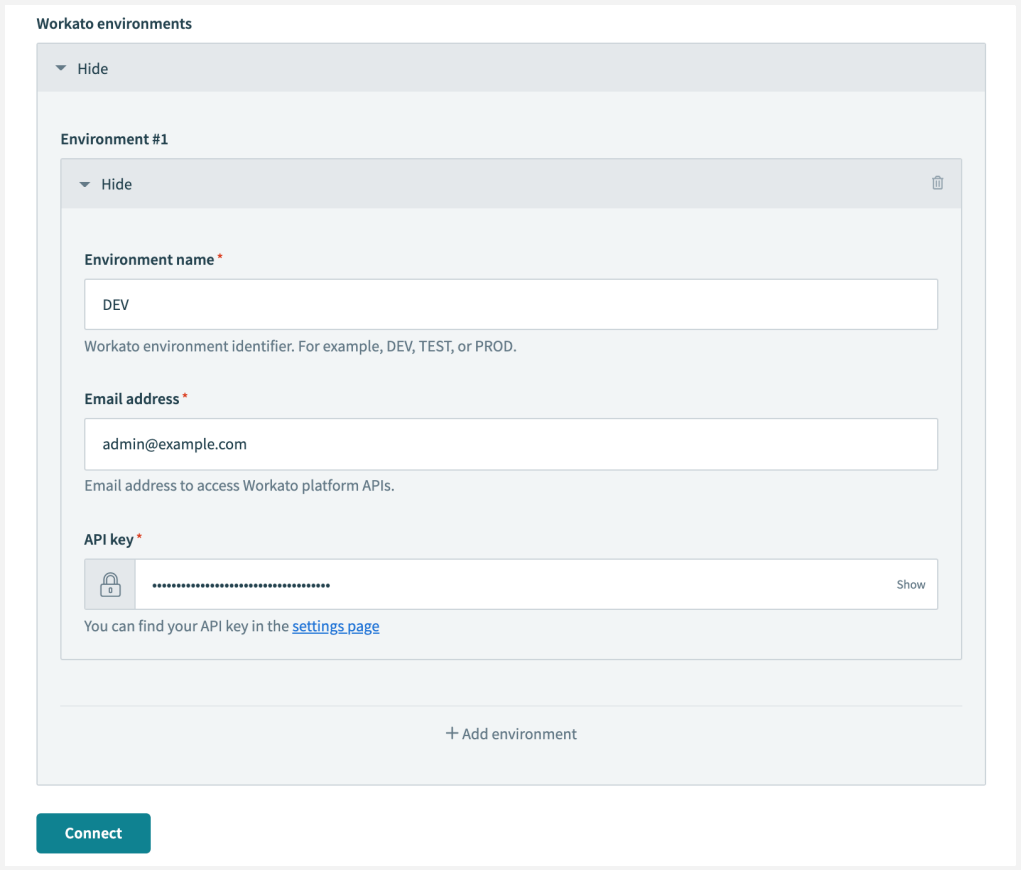
CICD for Workato custom SDK connection uses Workato developer APIs (docs) to perform various actions for package export and import with added advantage of long actions to handle asynchronous requests.

It also supports configuring multiple Workato workspaces in the same connection for ease of use and provides a secure connectivity to all Workato environments.

Navigate to the "Connections" folder under imported package to update the connection placeholder to configure your Workato environments.

Asset name	CICD | CONN | Workato
Asset type	Connection
Asset path	CICD Automation/Microsoft Teams/ Connections

Provide an environment name along with corresponding environment email and API key.



You can find your personal API key in the respective environment workspace settings page.

Open Account Settings from the left navigation bar in your workspace. Navigate to the API key section. Here, you can retrieve your email and Workato API key and refresh the API key if necessary.

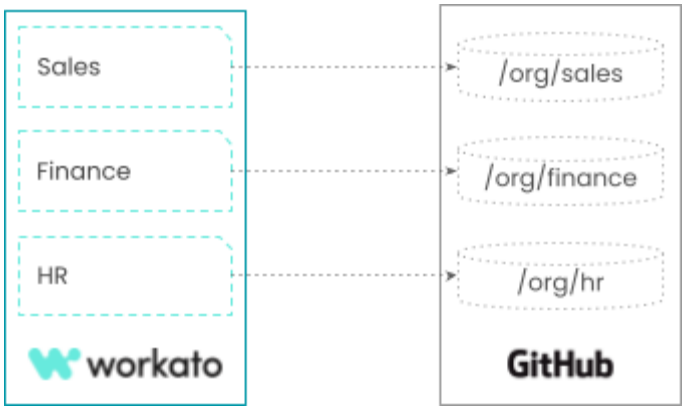
Repeat above steps for configuring test and production environments.

> *Note: Account settings are accessible using the workspace owner account only.*

Setup GitHub repositories

CI/CD Automation accelerator is developed to work with GitHub as a version control system out-of-the box.

The accelerator assumes that Workato top-level projects are mapped directly to the GitHub repositories. For example, if an organization contains three projects across Sales, Finance, and HR, you can map them to respective repositories in the organization's GitHub account as separate repositories.



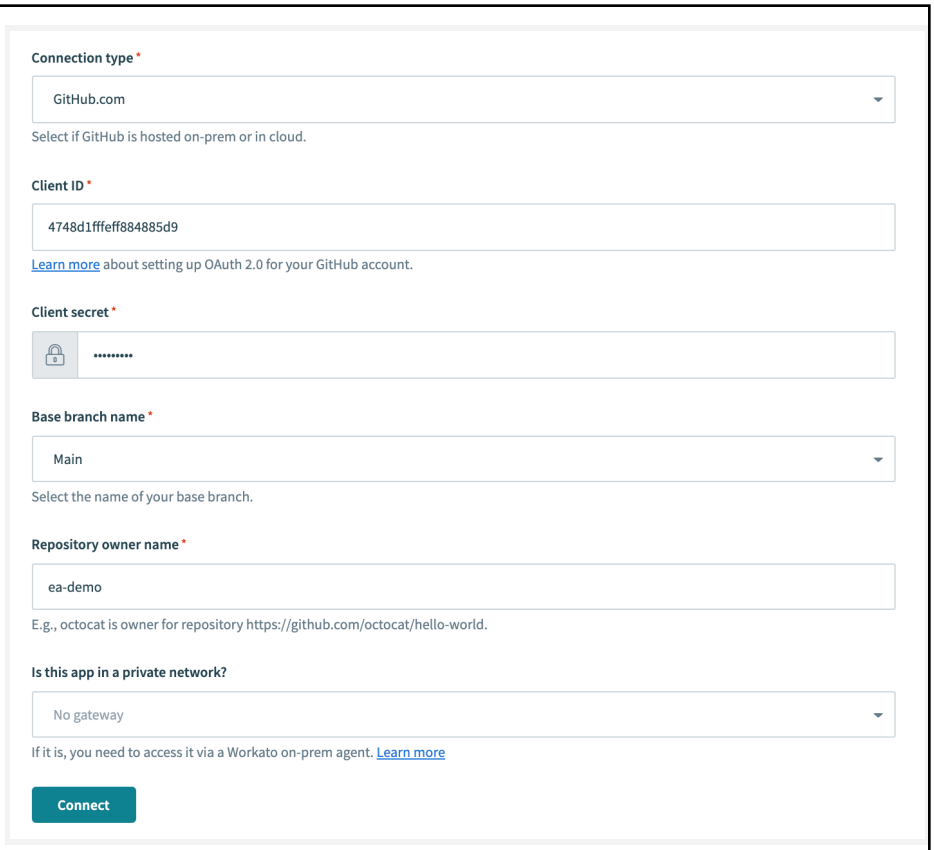
This approach has several benefits. By dividing the repositories by projects, you can continue to enforce access control policies outside the Workato platform. Also, this approach provides a solid foundation to manage release versions by projects. Please note that it is not recommended to modify files outside the Workato platform.

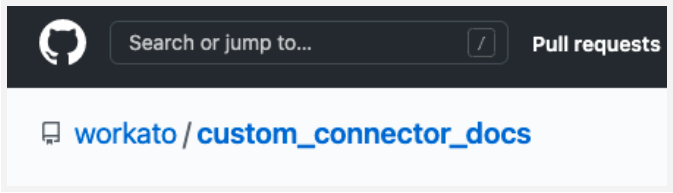
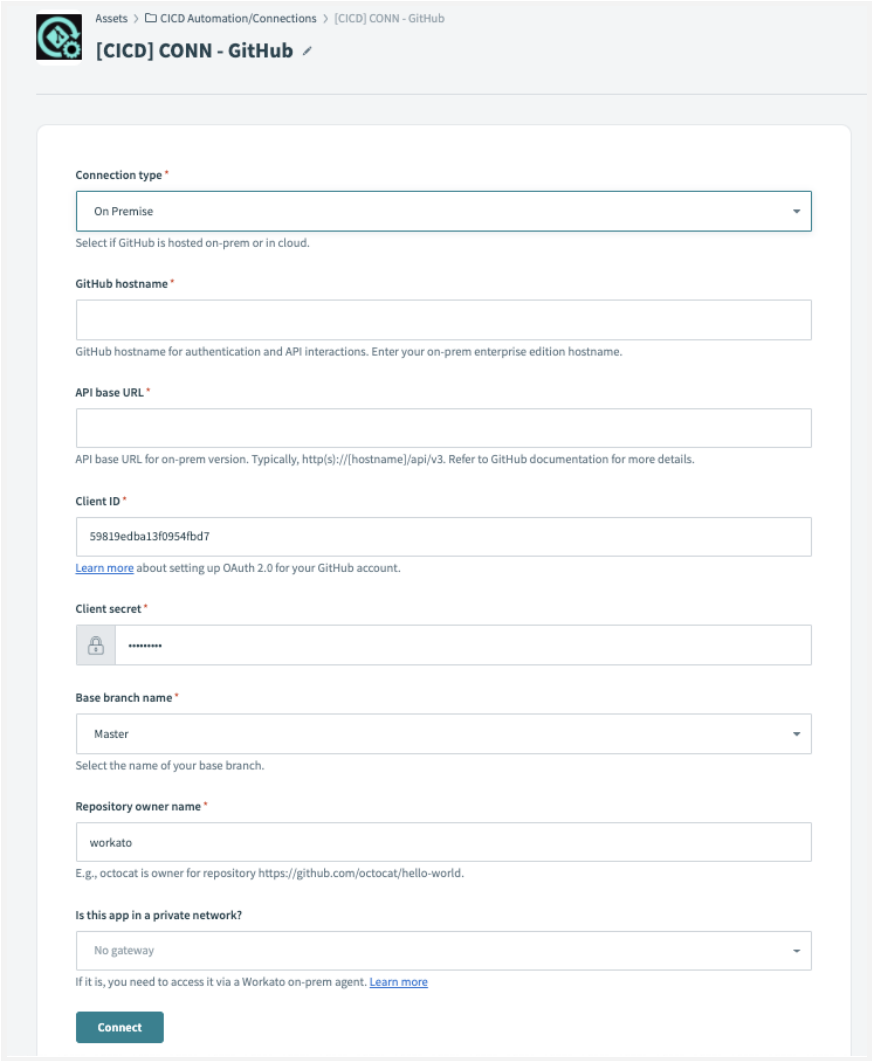
> Note: Please initialize the GitHub repository before using with CI/CD automation accelerator. You can use a sample readme file offered by GitHub while creating the repository for this purpose.

Asset name	CICD | CONN | GitHub
Asset type	Connection
Asset path	CICD Automation/Microsoft Teams/Connections

Update the connection placeholder in the imported package to configure your GitHub account.

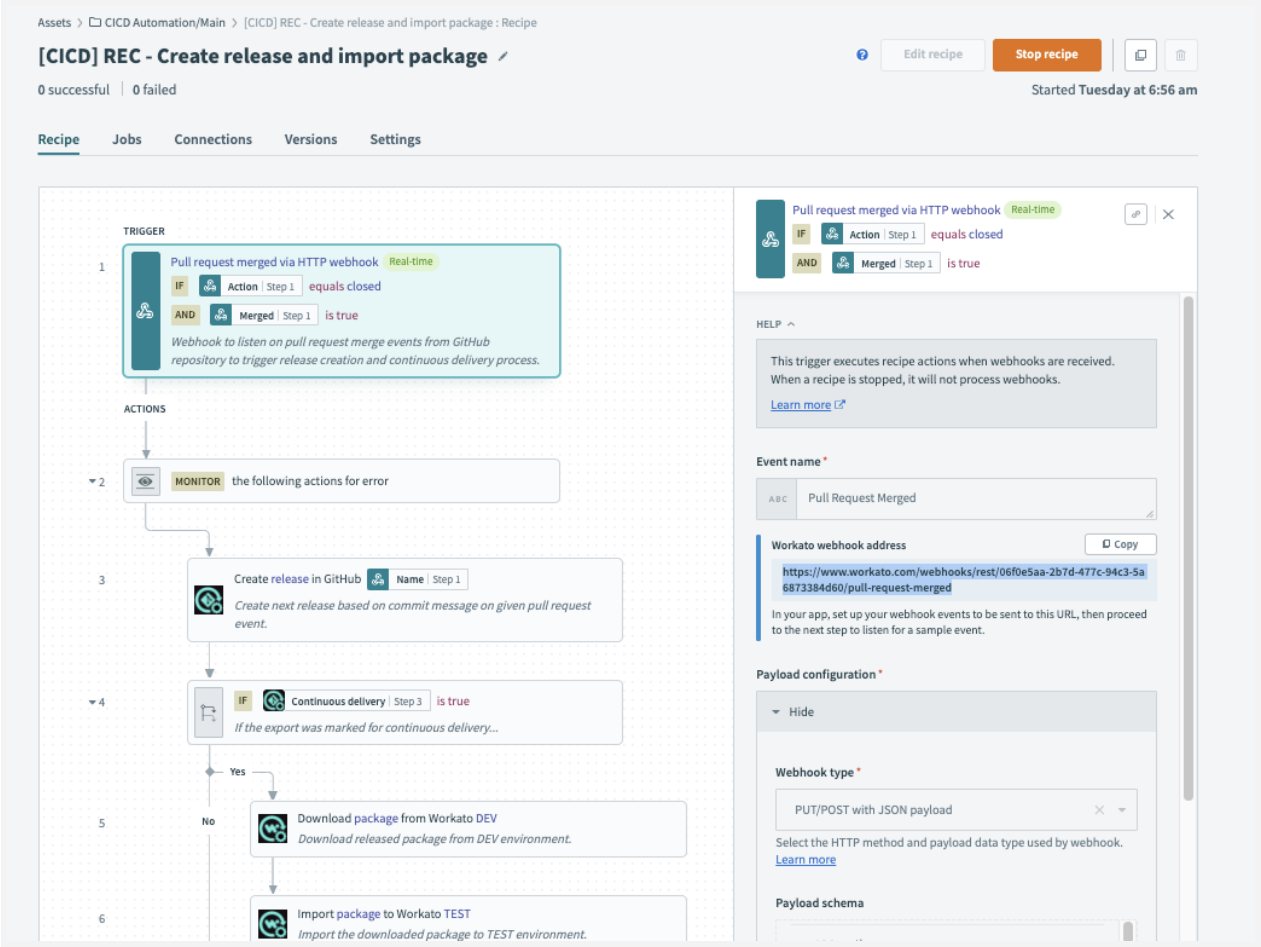
GitHub connection options

Connection type	 Select if GitHub is hosted on-prem or in the cloud.
OAuth Client ID and secret	Create a GitHub OAuth app for use with the accelerator. Use this document for a step-by-step guide. Use the following values for setting up OAuth app: **Application name**: Workato CICD Automation **Homepage URL**: https://www.workato.com **Application description**: Workato CICD Automation **Authorization callback URL**: https://www.workato.com/oauth/callback
Base branch name	Select the name of the repository base branch. For older setup it may be referred as “master”.
Repository owner name	Provide a repository owner name, it should be the same for all repositories to be used with the accelerator. In the example below, the owner name is “workato” for the repository “custom_connector_docs”.

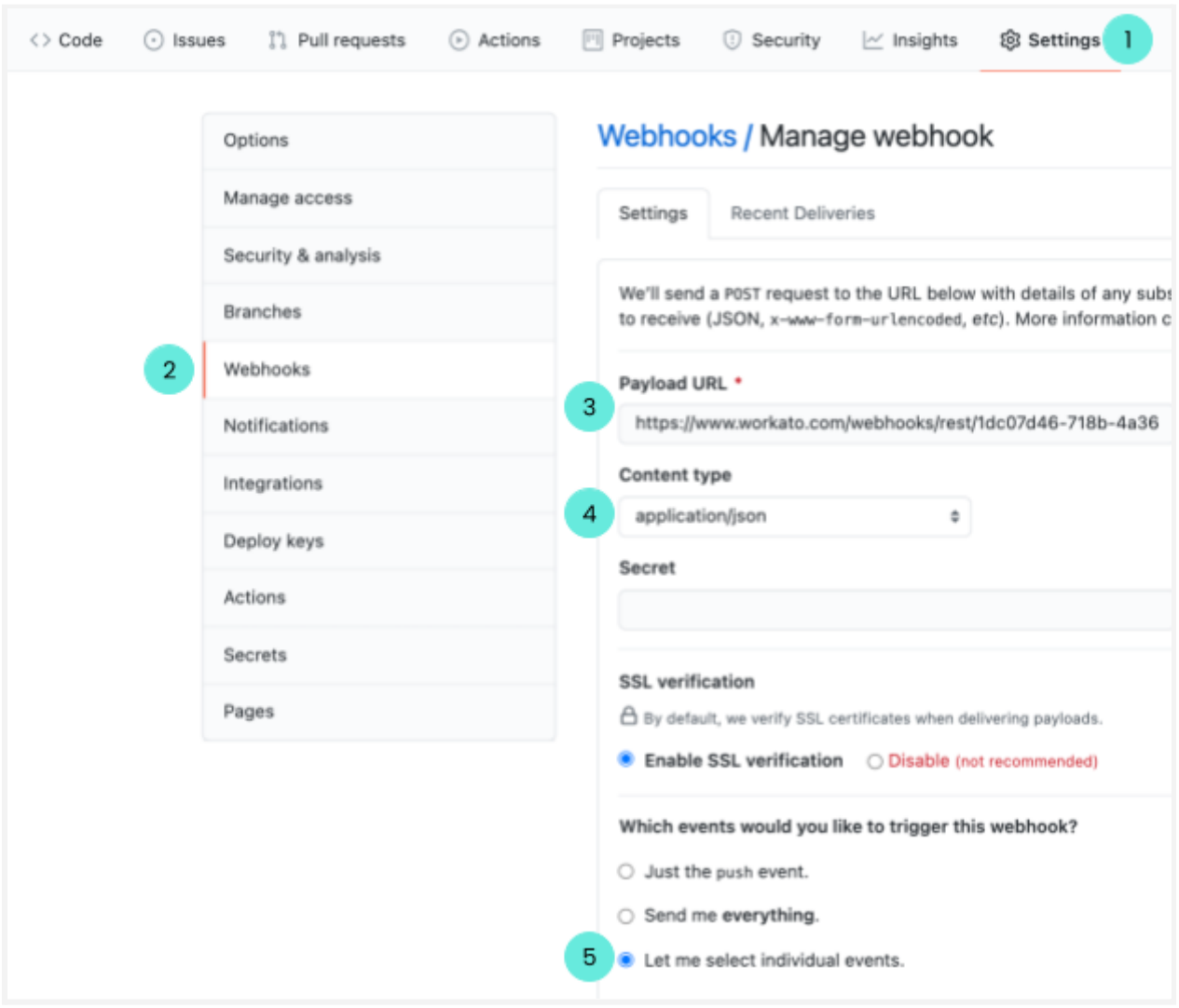
	
Additional configuration for on-prem edition	
GitHub hostname	 Provide hostname for on-premise GitHub server.
Is this app in a private network?	You need to access the on-prem GitHub edition via a Workato on-prem agent (OPA). Learn more about how to configure the OPA. Accelerator uses HTTP profile for making the connection.

GitHub webhook configuration

1. For each GitHub repository, set up webhook events by going to the “Settings” page of the repository.
2. Select the Webhooks option.
3. Update Payload URL to match imported from “CICD Automation/Microsoft Teams/Recipes/CICDT | REC-001 | Create release and import package with MS Teams” recipe’s webhook endpoint. You can find it in Trigger (step 1) of the recipe.



4. Change the content type to “application/json”.
5. Select “Let me select individual events” option and select only the “Pull requests” checkbox.
6. Click “Add webhook” to save.


Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
1
Options
Manage access
Security & analysis
Branches
2
Webhooks
Notifications
Integrations
Deploy keys
Actions
Secrets
Pages
Webhooks / Manage webhook
Settings
Recent Deliveries
We'll send a POST request to the URL below with details of any subs
to receive (JSON, x-www-form-urlencoded, etc). More information c
3
Payload URL *
https://www.workato.com/webhooks/rest/1dc07d46-718b-4a36
4
Content type
application/json
Secret
SSL verification
By default, we verify SSL certificates when delivering payloads.
Enable SSL verification
Disable (not recommended)
Which events would you like to trigger this webhook?
Just the push event.
Send me everything.
5
Let me select individual events.

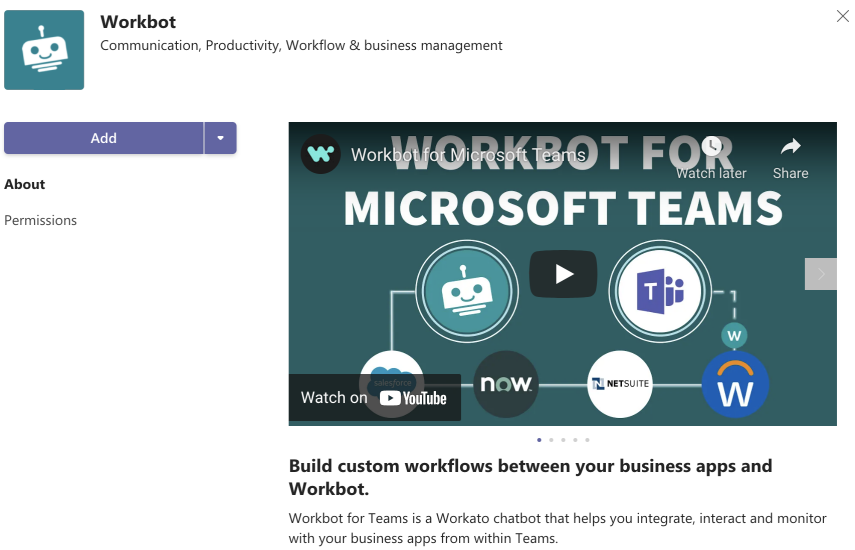
Which events would you like to trigger this webhook?

- ○ Just the `push` event.
- ○ Send me **everything**.
- ◉ Let me select individual events.

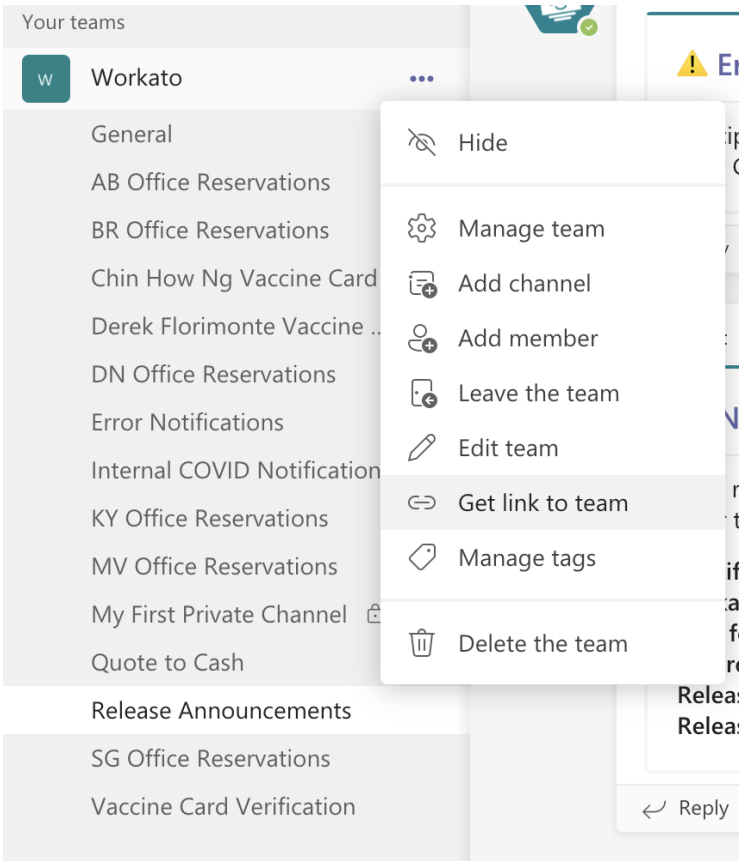
- ☐ **Branch or tag creation**
  Branch or tag created.
- ☐ **Branch or tag deletion**
  Branch or tag deleted.
- ☐ **Check runs**
  Check run is created, requested, rerequested, or completed.
- ☐ **Check suites**
  Check suite is requested, rerequested, or completed.
- ☐ **Code scanning alerts**
  Code Scanning alert created, fixed in branch, or closed
- ☐ **Collaborator add, remove, or changed**
  Collaborator added to, removed from, or has changed permissions for a repository.
- ☐ **Commit comments**
  Commit or diff commented on.
- ☐ **Deploy keys**
  A deploy key is created or deleted from a repository.
- ☐ **Deployment statuses**
  Deployment status updated from the API.
- ☐ **Deployments**
  Repository was deployed or a deployment was deleted.
- ☐ **Discussion comments** 🔒
  Discussion comment created, edited, or deleted.
- ☐ **Discussions** 🔒
  Discussion created, edited, pinned, unpinned, locked, unlocked, transferred, answered, unanswered, labeled, unlabeled, had its category changed, or was deleted.
- ☐ **Forks**
  Repository forked.
- ☐ **Issue comments**
  Issue comment created, edited, or deleted.
- ☐ **Issues**
  Issue opened, edited, deleted, transferred, pinned, unpinned, closed, reopened, assigned, unassigned, labeled, unlabeled, milestoned, demilestoned, locked, or unlocked.
- ☐ **Labels**
  Label created, edited or deleted.
- ☐ **Meta**
  This particular hook is deleted.
- ☐ **Milestones**
  Milestone created, closed, opened, edited, or deleted.
- ☐ **Packages**
  GitHub Packages published or updated in a repository.
- ☐ **Page builds**
  Pages site built.
- ☐ **Project cards**
  Project card created, updated, or deleted.
- ☐ **Project columns**
  Project column created, updated, moved or deleted.
- ☐ **Projects**
  Project created, updated, or deleted.
- ☐ **Pull request review comments**
  Pull request diff comment created, edited, or deleted.
- ☐ **Pull request review threads**
  A pull request review thread was resolved or unresolved.
- ☐ **Pull request reviews**
  Pull request review submitted, edited, or dismissed.
- ☑ **Pull requests**
  Pull request opened, closed, reopened, edited, assigned, unassigned, review requested, review request removed, labeled, unlabeled, synchronized, ready for review, converted to draft, locked, unlocked, auto merge enabled,
- ☐ **Pushes**
  Git push to a repository.

Setup Microsoft Teams workspace

Install Workbot from the Microsoft Teams Apps Listing for yourself.



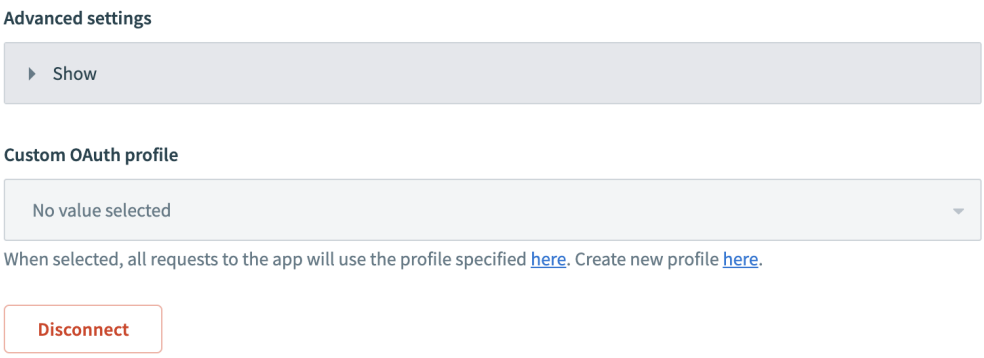
From the Teams in your workspace, copy the link to the Team that has all the possible reviewers of the projects in Workato.



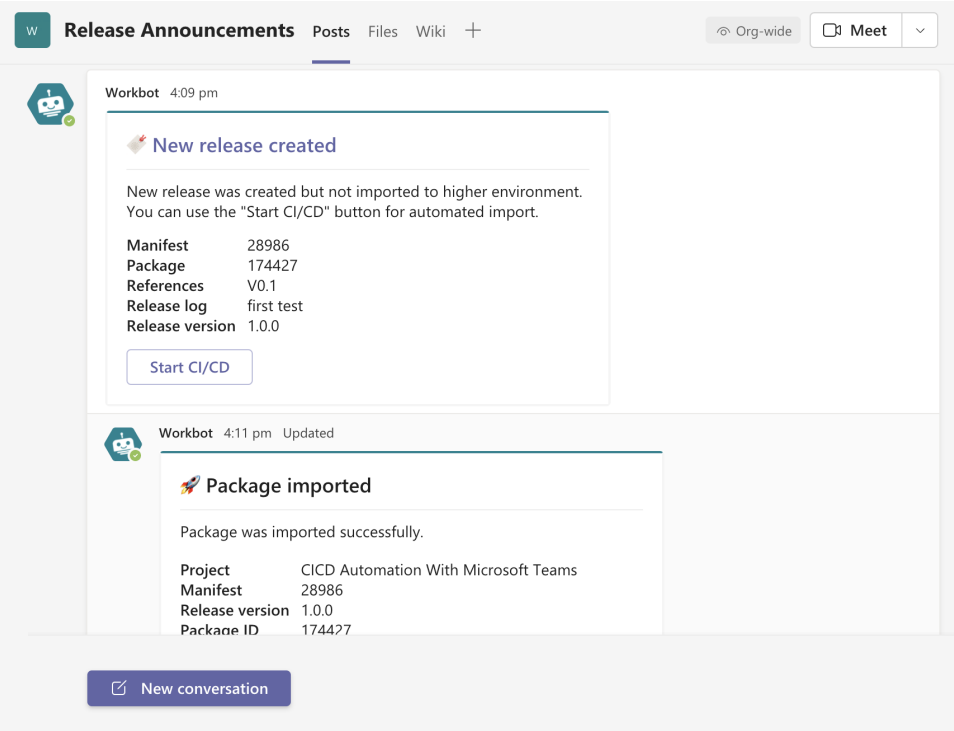
Within the link, copy the “group ID”, i.e. “https://teams.microsoft.com/l/team/19%3a99bc6d45c22b4cbc97c7be88f1cc0ff8%40thread.tacv2/conversations?groupId={{**b7a56b3d-5a86-4284-99e3-d64e0b0e016e**}}&tenantId=a1aa4e15-f702-40ed-8632-fce16e362275”. Navigate to Account Properties in your Workato account and look for the “CICD_GroupID” property. Paste the group ID into the value.

Navigate to the “Connections” folder under imported package to connect the Microsoft Teams connection placeholder. It uses standard Microsoft Teams OAuth for authorization. Select the appropriate workspace during the authentication to create the connection.

Asset name	CICDT | CONN | Workbot for Microsoft Teams
Asset type	Connection
Asset path	CICD Automation/Microsoft Teams/Connections



Optionally, create a channel named “Release Announcements” to keep all collaborators informed as the releases are created.



Update the channel name in the recipe “CICD Automation/Microsoft Teams/Recipes/CICDT | REC | Create release and deploy package” steps #7, #9, and #13.

Connected Workbot should be added to the selected channel to allow notifications.

Start the recipes

Once you have successfully set up the connections, please start the following recipes. You can click on the dropdown menu on each recipe to start the recipe.

Recipe name	Path under CICD Automation
CICDT | BOT | Display command options	Microsoft Teams / Bots
CICDT | BOT | Configure project	Microsoft Teams / Bots
CICDT | BOT | Submit release request	Microsoft Teams / Bots
CICDT | BOT | Submit deployment request	Microsoft Teams / Bots
CICDT | REC | Create release and deplou package	Microsoft Teams / Recipes
CICD | RF | Create folders list	Functions
CICD | RF | Create configured projects list	Functions
CICDT | RF | Build and create pull request	Microsoft Teams / Recipes
CICDT | RF | Import package from MS Teams	Microsoft Teams / Recipes
CICDT | RF | Get User Channel ID	Microsoft Teams / Recipes

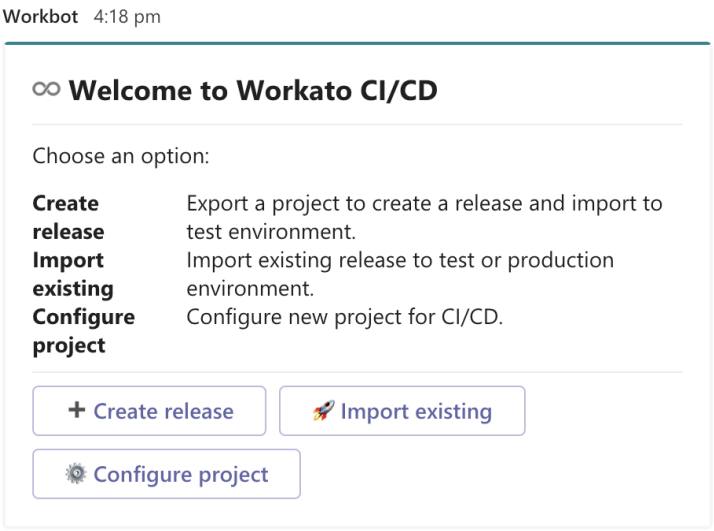
Using The Accelerator

Start the automation

Start the CI/CD automation by using Workbot for Microsoft Teams command given below.

```
workato_cicd_start
```

The above command will present different options to the user, select an appropriate option to continue.



Configure projects

You will need to configure the project for use with CI/CD automation before you can use automated export and import capabilities. You can configure the project by selecting the “Configure project” option. It will open a modal wizard to configure a new project with the following input fields.

There are two different approaches to configure projects

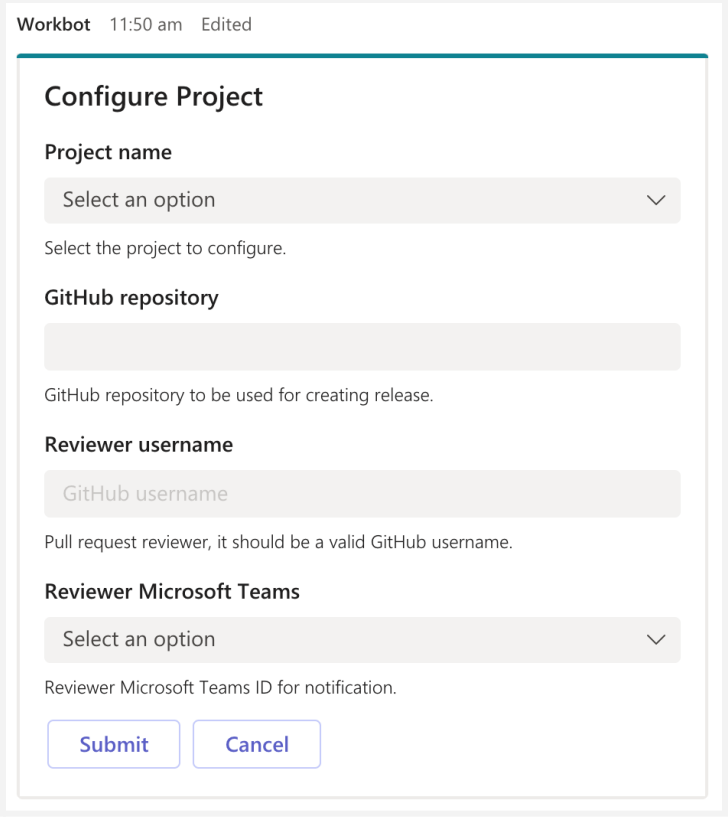
1. **Configure project for full release**: Full release option uses the new Projects API for the streamlined deployment workflow. Full release presents automatically created list of top-level project for configuration with GitHub repository.

   Configuring full release will deploy all assets in the project with each release/deployment request. In addition, it will ensure that all folder structures

are in-sync across environments and you will not be required to pre-create folders across environments before configuration/deployment.

The projects deployed using the full release mode will also log complete history of different build and deploy actions under the Deployment tab for each project.

Name	Description	Optional
Project name	Workato project name to configure. E.g. Salesforce Case Sync.	No
GitHub repository	GitHub repository to be used for creating pull requests and managing project release versions. It is recommended to use one-to-one mapping between project name and GitHub repositories for error-free release tagging.	No
Reviewer username	Pull request reviewer, it should be a valid GitHub username.	Yes
Reviewer Slack	Slack ID of review user for notification during pull request assignment.	Yes

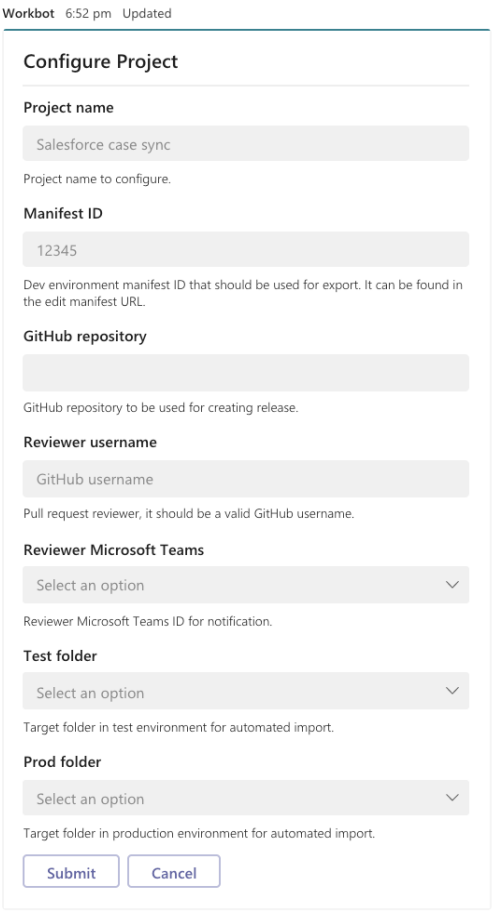


2. **Configure project for partial release**: Partial release option will use the manifest and the Recipe Lifecycle Management APIs for export and import of packages across environments.

   Manifests allow you to deploy only the assets that you want to manage as a part of the current release. This gives better flexibility in managing developed assets with other under-development assets separately while allowing them to be part of the same top-level project.

   To use this option, you will need to create a manifest using the platform tools and also ensure that the target top-level project folders already exist in the test and production environments.

Name	Description	Optional
Project name	Workato project name to configure. E.g. Salesforce Case Sync.	No
Manifest ID	Dev environment manifest ID that should be used for export. It can be found in the edit manifest URL. It should be unique across all configured projects. However, you can use different manifest IDs for a given Project name and GitHub repository combination.	No
GitHub repository	GitHub repository to be used for creating pull requests and managing project release versions. It is recommended to use one-to-one mapping between project name and GitHub repositories for error-free release tagging.	No
Reviewer username	Pull request reviewer, it should be a valid GitHub username.	Yes
Reviewer Microsoft Teams	Microsoft User ID of review user for notification during pull request assignment.	Yes
Test folder	Target folder in the test environment for automated import.	No
Prod folder	Target folder in the production environment for automated import.	No



The provided details will be configured in the “Project configuration” lookup table in Workato.

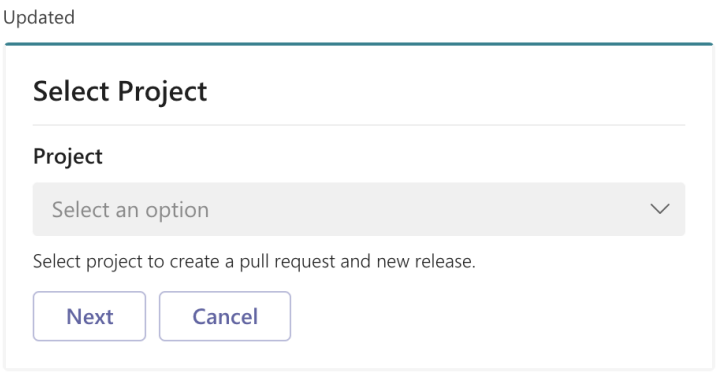
Create new release

You can create new releases for any configured projects for continuous integration and continuous delivery.

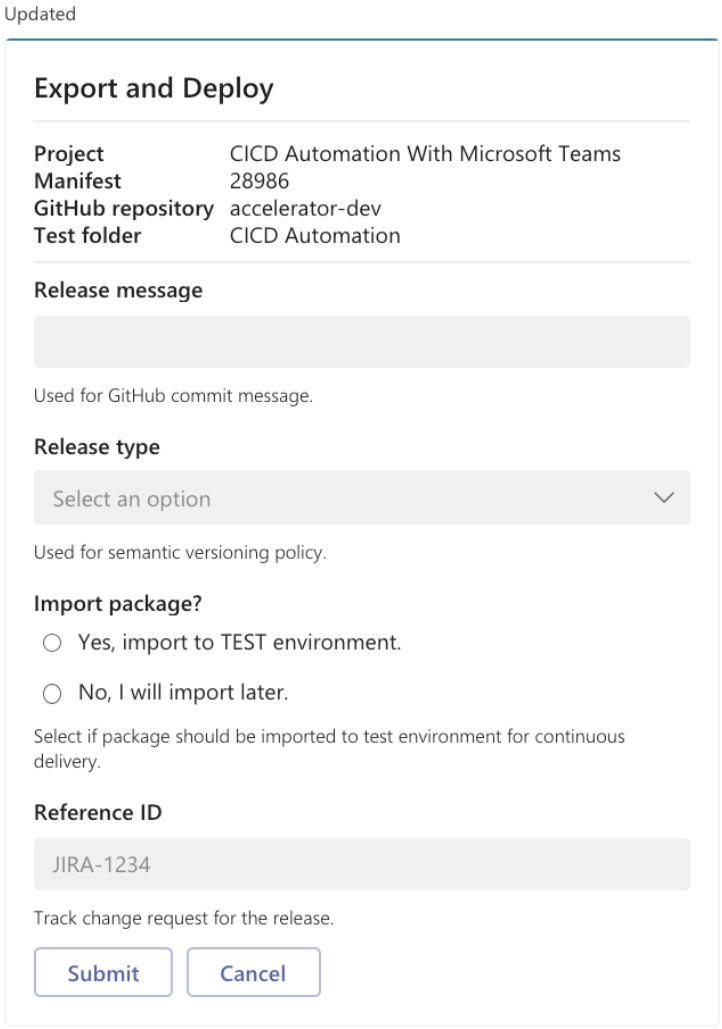
Creating a release consists of the following automated steps:

1. Build project/export the manifest from the Workato development environment.
2. Download, unzip, and commit the exported package to a new feature branch in the GitHub repository.
3. Create a pull request and assign it to the configured reviewer.
4. Notify pull request details to the requester and the reviewer on Microsoft Teams.
5. Monitor pull request merge event and create new release based on commit message details such as release type and last release version.
6. deploy the newly created to the Workato test environment (if selected).

You can create a new release by selecting the “Create release” option. It will open an adaptive card with the following input fields.

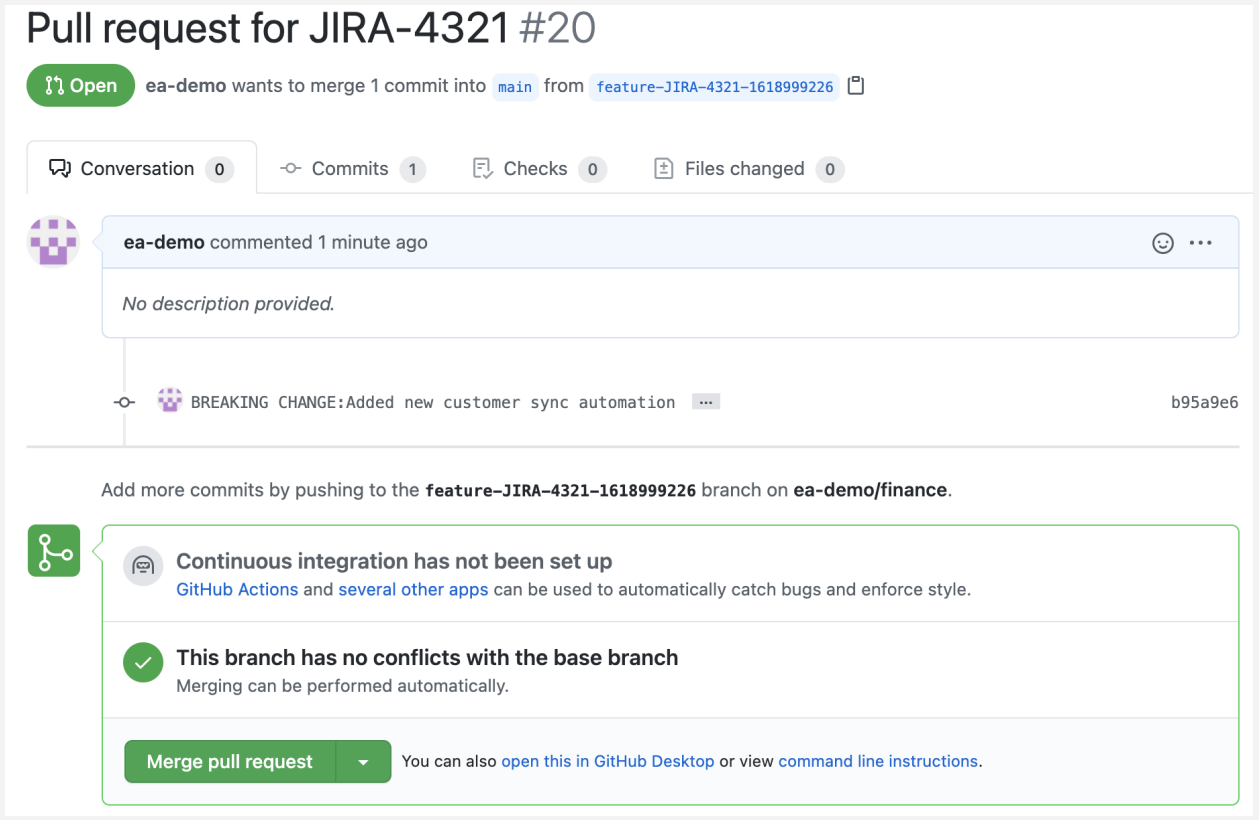
Name	Description	Optional
Select project	 Select the project from the menu to create a new release.	No
Release message	Release comments used as GitHub commit messages.	No
Release type	Select from the Major, Minor, Bug-fix options for automatic release tagging based on semantic versioning policy.	No
Deploy package?	Select Yes to import to the test environment for continuous delivery or No to import later.	No

Reference ID	Used to track any change request number like the JIRA issue with the release. It is optional input but recommended.	Yes



Upon successful build and commit of the package to GitHub repository, a pull request will be created. It will need to be reviewed and managed by the assigned reviewer/user to continue with the rest of the CI/CD process of creating and importing the project release to the test environment.

Reviewers and requesters will be notified of pull request creation events in the Microsoft Teams message. Reviewers can use the attached pull request link to open it in GitHub. The reviewers can review pull request changed files for adherence to defined guidelines, provide review comments, and request fixes in the dev environment. Or approve the change for promotion to the test environment by merging the pull request.



Reviewers should also ensure that the merge does not have any conflict with the main branch due to any changes/removal of asset files from the project and resolve such issues before merging the pull request.

> Tip: For partial release, create manifest with source folder as “All projects” and then select only the assets that you want to be included in the release. This approach will maintain the folder hierarchy and avoid any merge conflicts.

Successful merge events will then be captured by the CI/CD automation accelerator to automatically create a new release and import the package to the test environment, if requested.

Workbot 14:37

🚀 New release imported

New release was created and imported to test environment. Please refer to link above for release details.

Manifest	29401
Package	176274
Test folder	526110
References	JIRA-1324
Release log	releasing from teams
Release version	0.3.0

Deploy release

Finally, you can deploy any existing release using the “Deploy release” option. This option will be useful to promote configured projects to the production environment upon successful completion of the testing phase. But you can also use it to deploy changes to other environments such as testing to rollback changes to previous release versions.

The import existing option requires the following input fields.

Name	Description	Optional
Select project to import	Select the project from the menu to import.	No
Target environment	Select the target environment to import the package.	No
Release version	Select the release version to import. Release versions also display package ID and reference ID, if any, to help select the right version for importing to a higher environment.	No
Comments	Any comments to log with the import process.	Yes

Troubleshooting

Error while connecting Microsoft Teams

One of the following errors while connecting to Microsoft Teams workspace:

`Bot already installed to the team`

`Teams team accelerators-dev already has Workbot installed.`

We do not support multiple Workbot connections to the same team. Try one of these solutions:

- If you connected to an incorrect Microsoft Teams team by mistake, try connecting to a different Microsoft Teams team.
- If you think you should own Workbot, ask the above user to remove it from accelerators-dev. Once they remove it, you will be able to install Workbot. Learn how to manage apps in Microsoft Teams.
- You don't really have to use Workbot, you can also create a shiny new bot for your Microsoft Teams team. It's easy and you will be able to do exactly the same things.

Feedback and Questions

If you have any feedback about this accelerator or any of the assets or documentation included with it, please contact the Accelerators team at accelerators.feedback@workato.com.