



Persistent UNITE for GitHub

User Guide
Version 2.0.2

Statement of confidentiality

© Copyright Persistent Systems 2020

This document is provided with the explicit understanding that the contents will not be divulged to any third party without prior written consent from Persistent Systems. The contents of this document are proprietary and confidential information of Persistent Systems. All registered trademarks are acknowledged. All other company logos or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

Contents

Statement of confidentiality.....	2
Revision history	4
Overview	5
Prerequisites	6
Adding a branch from GitHub server to a GC stream.....	8
Adding a tag from GitHub server to a GC stream	11
Replacing a branch from GitHub server in a GC stream	14
Replacing a tag from GitHub server in a GC stream	16
Creating a branch from a tag	18
Creating a tag from a branch	21
Known issues and workarounds	24

Revision history

Changes to this document are summarized in the following table in chronological order.

Version	Date	Short description
2.0.2	31 August 2020	<p>Includes the following details:</p> <ul style="list-style-type: none">• Add a GitHub branch in Global Configuration (GC) stream• Add a GitHub tag in GC stream• Replace a GitHub branch in GC stream• Replace a GitHub tag in GC stream• Create a tag from the branch• Create a branch from the tag

Overview

Persistent UNITE™ for GitHub integrates GitHub with the Global Configuration (GC) application in the IBM Engineering Lifecycle Management (ELM) solution. This solution was previously known as the Internet of Things Continuous Engineering Solution (CE) and the Rational solution for Collaborative Lifecycle Management (CLM). For more information about name change, see [Renaming the IBM Continuous Engineering Portfolio](#).

When you work in an integrated environment, it is important to understand the terminology used by the applications. GitHub and ELM use slightly different terms for the same elements.

In GitHub, a *branch* is an independent line of development in a project. In ELM, it is called as *stream*. You can add branches in GC streams.

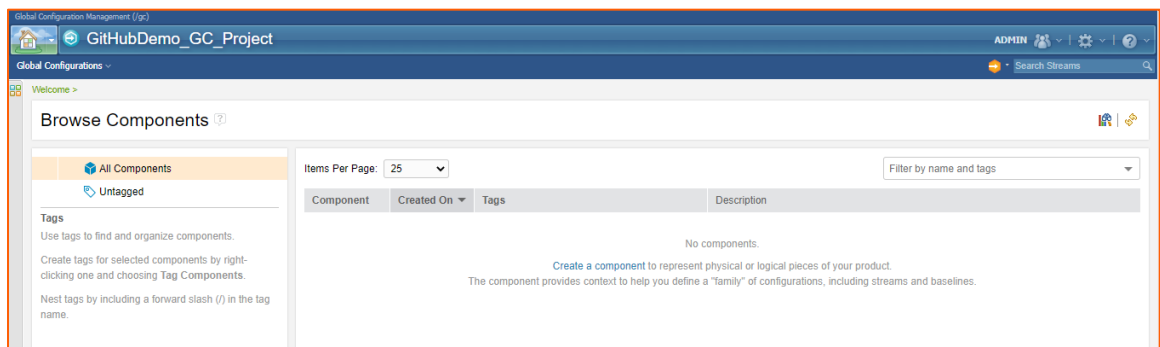
A *tag* in GitHub is a development branch that does not change. It is a pointer to the specific commit. In ELM, it is called as *baseline*. In ELM, you can add tags to GC streams. It helps you to find them easily in a search or to group them in a way that makes sense for your team.

Prerequisites

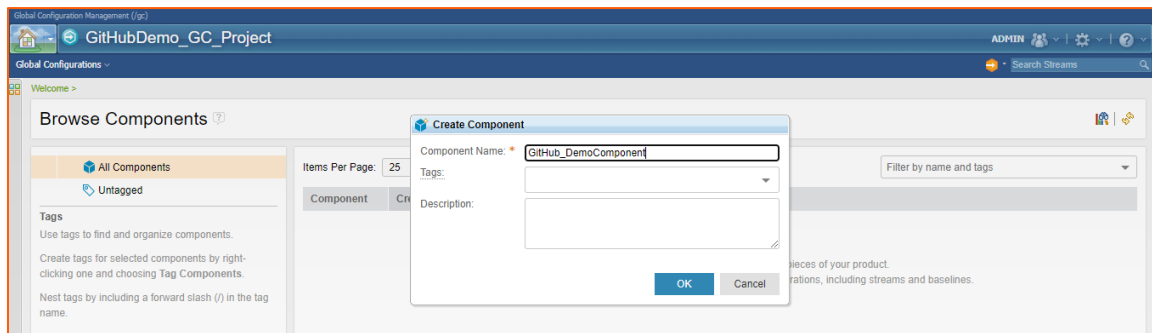
- *Persistent UNITE for GitHub* version 2.0.2 must be installed and configured.
See '*Persistent UNITE for GitHub Installation and Configuration Guide*' version 2.0.2 for the details.
 - A project area is created in the Global Configuration Management (GCM) application.
See [Creating GCM project area](#) for the details.
 - A component is created in GCM project area by following these steps:
Also, you can see [Creating GCM components](#) for the details.
1. Open the GCM project area and click **Browse and create components**.



2. Click **Create a component**.



3. Enter the **Component Name**. For example, `GitHub_DemoComponent`.



4. Click **OK**. The component is created.

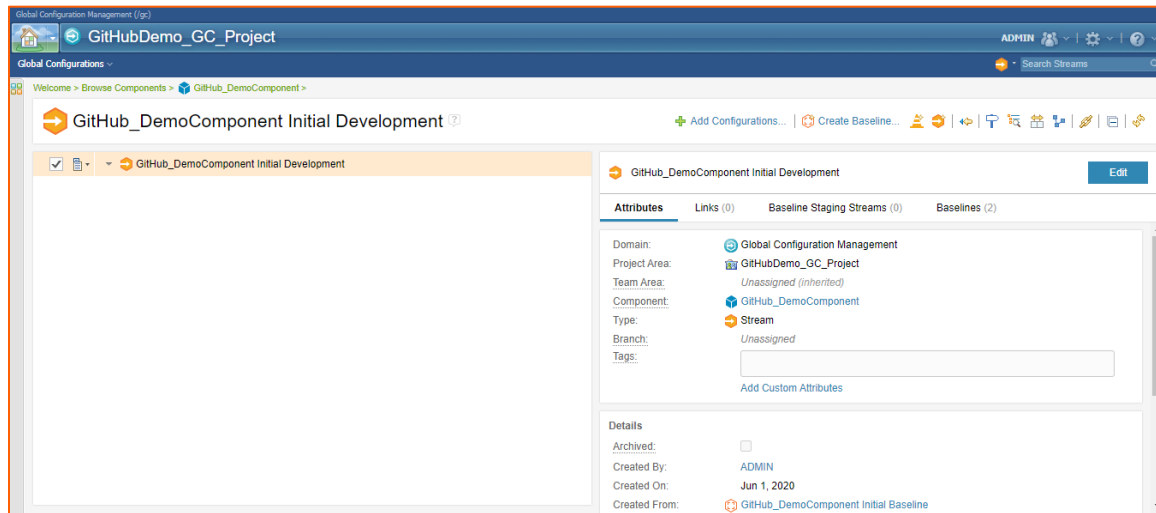


5. From the list of components, click the newly created component to see the details.

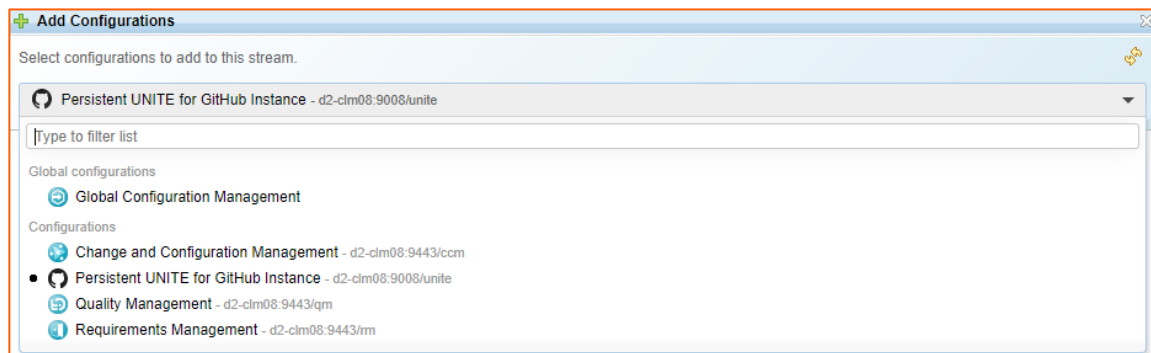
Adding a branch from GitHub server to a GC stream

1. Open the GC stream in GCM application.
2. Select the GC stream and click **Add Configurations**.

Add Configurations window opens.

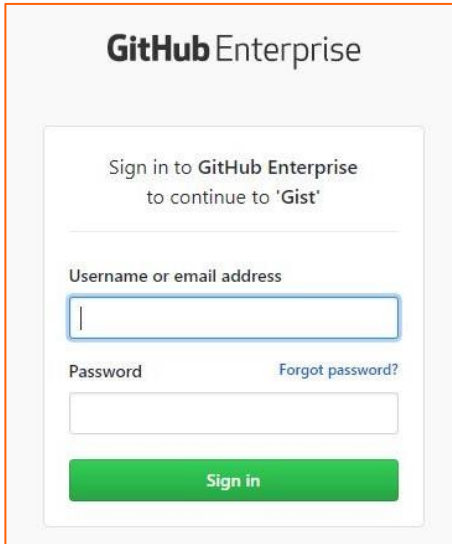


3. Select Persistent UNITE for GitHub Instance from **Configurations** list.
It is the connector instance name that is assigned during connector instance configuration.



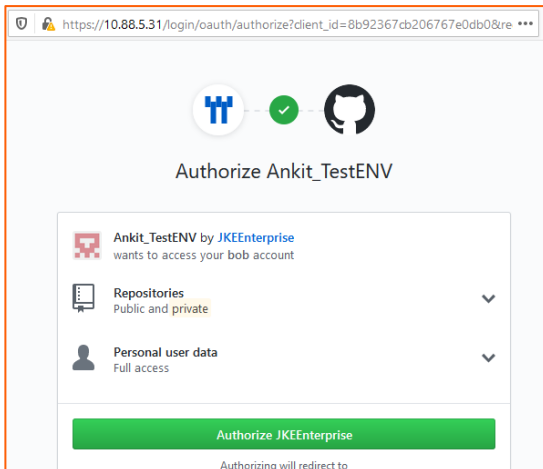
4. If you see the following page, enter **Username or email address** and **Password** of your GitHub account and click **Sign in**.

Note: If you added an artifact to GC stream previously, the login page does not show and it logs in automatically.



The image shows the GitHub Enterprise login page. At the top, it says "GitHub Enterprise". Below that, it says "Sign in to GitHub Enterprise to continue to 'Gist'". There are two input fields: "Username or email address" and "Password". A link "Forgot password?" is next to the password field. A green "Sign in" button is at the bottom.

5. If you are logging in for the first time, an authorization page is shown. Authorize the access to go to Add Configurations page.



The image shows a browser window with the URL "https://10.88.5.31/login/oauth/authorize?client_id=8b92367cb206767e0db08&re...". The page title is "Authorize Ankit_TestENV". It shows a connection between a Jenkins icon, a green checkmark, and a GitHub icon. Below this, it says "Ankit_TestENV by JKEEnterprise wants to access your bob account". There are two sections: "Repositories" with "Public and private" access and "Personal user data" with "Full access". A green "Authorize JKEEnterprise" button is at the bottom. A small note at the bottom says "Authorizing will redirect to".

6. In Add Configurations page, select the following fields from the lists:

- **Repository**
- **Artifact Type** - Select Branch.

The following table is updated with the relevant records.

Select configurations to add to this stream.

Persistent UNITE for github_demo - d2-clm08.9008/unite

Repository: JKEEnterpriseJupiter

Artifact Type: Branch

No.	Name
1	JKEEnterprise_jupiter_testStreamDemoOne
2	master
3	testBranchOne
4	testBranchThree
5	testBranchTwenty
6	testBranchTwo

1 records selected

Prev | Showing 1 - 6 of 6 | Next

Persistent UNITE™ powered by OSLC

Submit Cancel

7. Select the branch and click **Submit**.

Note: Alternatively, you can search the branch name in search text box.

The added branch is reflected as a stream and listed under the GC stream.

8. Select the branch to see **Overview** and **Details** in the right pane.

Global Configuration Management (Gp3)

GitHubDemo_GC_Project

ADMIN

Global Configurations

Welcome > Browse Components > GitHub_DemoComponent >

GitHub_DemoComponent Initial Development

Create Baseline... Replace... + -

master

Overview

Connector: GitHub

Repository: JKEEnterpriseJupiter

Details

Artifact Type: Branch

Branch Name: master

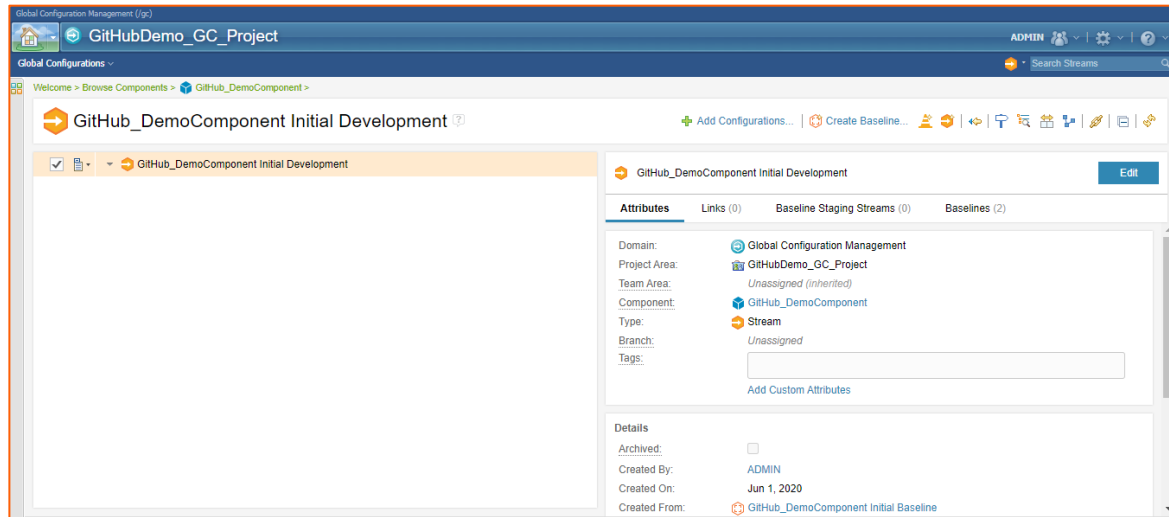
About: https://d2-clm08.9008/unite/services/1.0/628/config/github/4A4B45456E7465720726973652F6A757069746572/resources/stream/6D6173746572

Component: https://d2-clm08.9008/unite/services/1.0/628/config/github/4A4B45456E7465720726973652F6A757069746572/resources/component

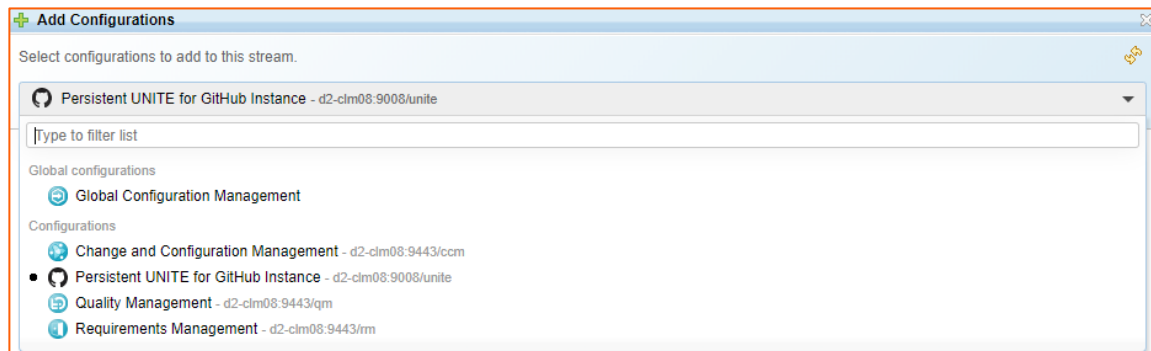
Adding a tag from GitHub server to a GC stream

1. Open the GC stream in GCM application.
2. Select the GC stream and click **Add Configurations**.

Add Configurations window opens.

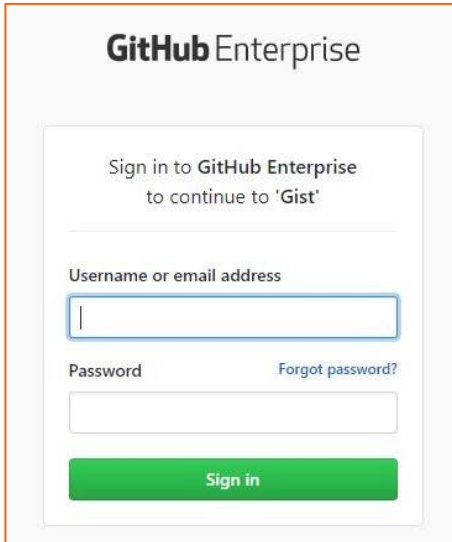


3. Select Persistent UNITE for GitHub Instance from **Configurations** list.
It is the connector instance name that is assigned during connector instance configuration.



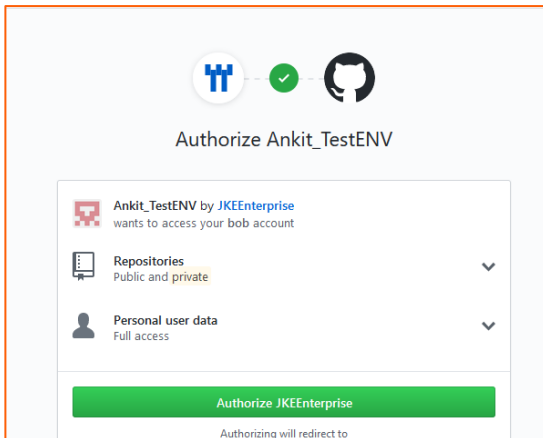
4. If you see the following page, enter **Username or email address** and **Password** of your GitHub account and click **Sign in**.

Note: If you added an artifact to GC stream previously, the login page does not show and it logs in automatically.



The image shows the GitHub Enterprise login page. At the top, it says "GitHub Enterprise". Below that, it says "Sign in to GitHub Enterprise to continue to 'Gist'". There are two input fields: "Username or email address" and "Password". A link "Forgot password?" is next to the password field. A green "Sign in" button is at the bottom.

5. If you are logging in for the first time, an authorization page is shown. Authorize the access to go to Add Configurations page.



The image shows the GitHub authorization page. At the top, it says "Authorize Ankit_TestENV". Below that, it shows a list of permissions: "Repositories" (Public and private) and "Personal user data" (Full access). A green "Authorize JKEEnterprise" button is at the bottom. Below the button, it says "Authorizing will redirect to".

6. In Add Configurations page, select following fields from the lists:

- **Repository**
- **Artifact Type** - Select Tag.

The following table is updated with the relevant records.

Select configurations to add to this stream.

Persistent UNITE for github_demo - d2-cim08.9008/unite

Repository: JKEEnterprise/jupiter

Artifact Type: Tag

No.	Name
1	v5
2	v4
3	v3
4	v2
5	v1
6	testBranchOne_testBaselineDemoOne

1 records selected

Prev | Showing 1 - 8 of 8 | Next

Persistent UNITE™ powered by OSLC

Submit Cancel

7. Select the tag and click **Submit**.

Note: Alternatively, you can search the tag name in search text box.

The added tag is reflected as a baseline and listed under the GC stream.

Global Configuration Management (jgc)

GitHubDemo_GC_Project

Global Configurations

Welcome > Browse Components > GitHub_DemoComponent >

GitHub_DemoComponent Initial Development

Create Stream... | Replace... | [Icons]

GitHub_DemoComponent Initial Development
v5

Overview

Connector: GitHub

Repository: JKEEnterprise/jupiter

Details

Artifact Type: Tag

Tag Name: v5

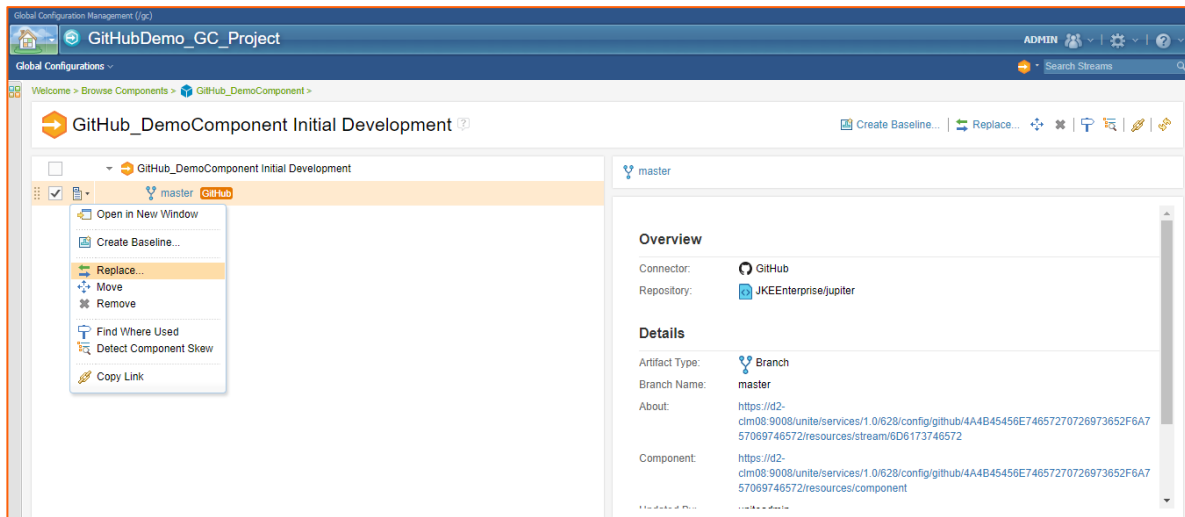
About: https://d2-cim08.9008/unite/services/1.0/625/config/github/4A4B45456E74657270726973652F6A757069746572/resources/baseline/7635

Component: https://d2-cim08.9008/unite/services/1.0/625/config/github/4A4B45456E74657270726973652F6A757069746572/resources/component

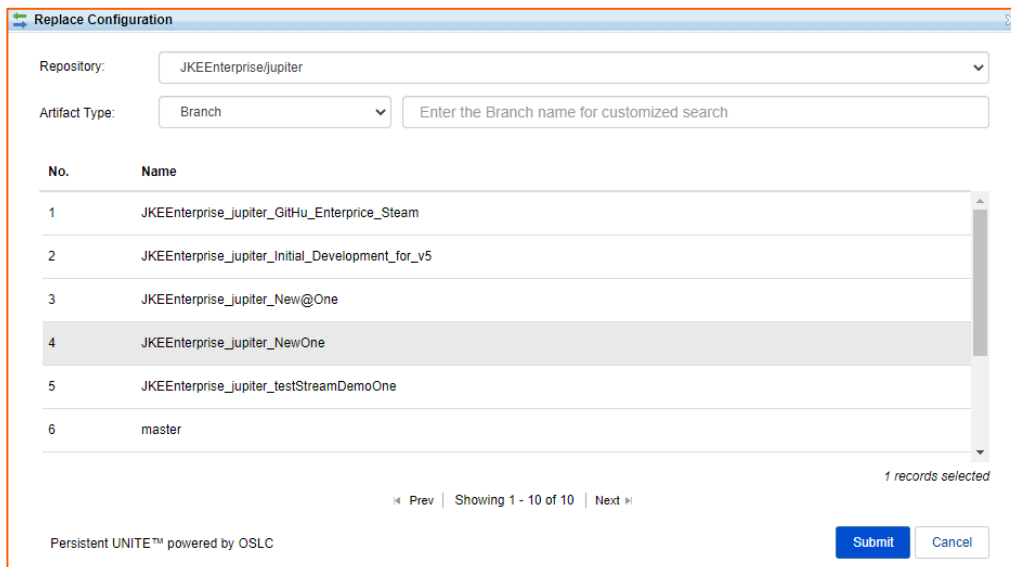
8. Select the tag to see **Overview** and **Details** in the right pane.

Replacing a branch from GitHub server in a GC stream

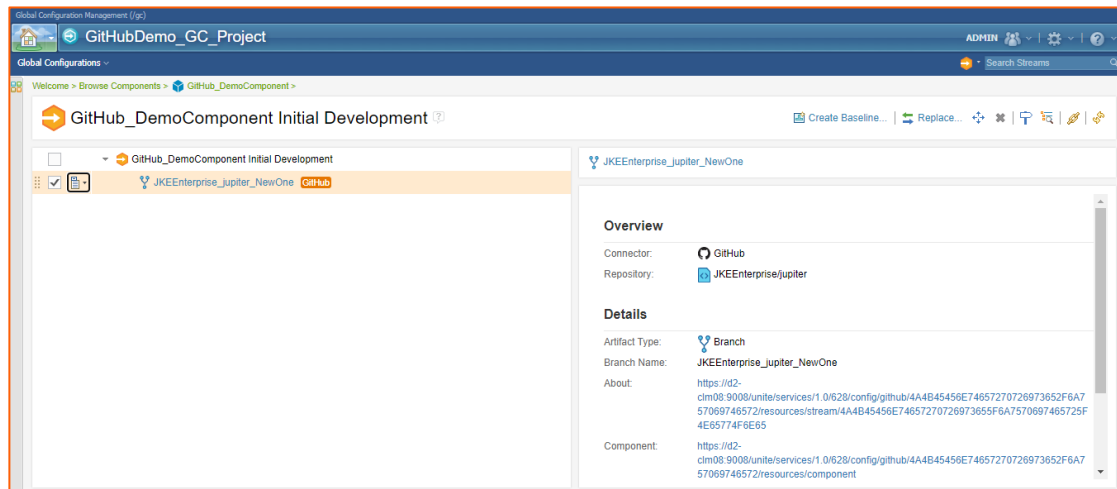
1. Open the GC stream that includes the branch to be replaced.
Note: A branch in GitHub is represented as a stream in GCM application.
2. Select the branch and go to **Options > Replace**.
 Replace Configuration page opens.



3. Select the new branch to replace the existing branch and click **Submit**.
 The existing branch is replaced.
 The new branch is represented as a stream under GC stream.

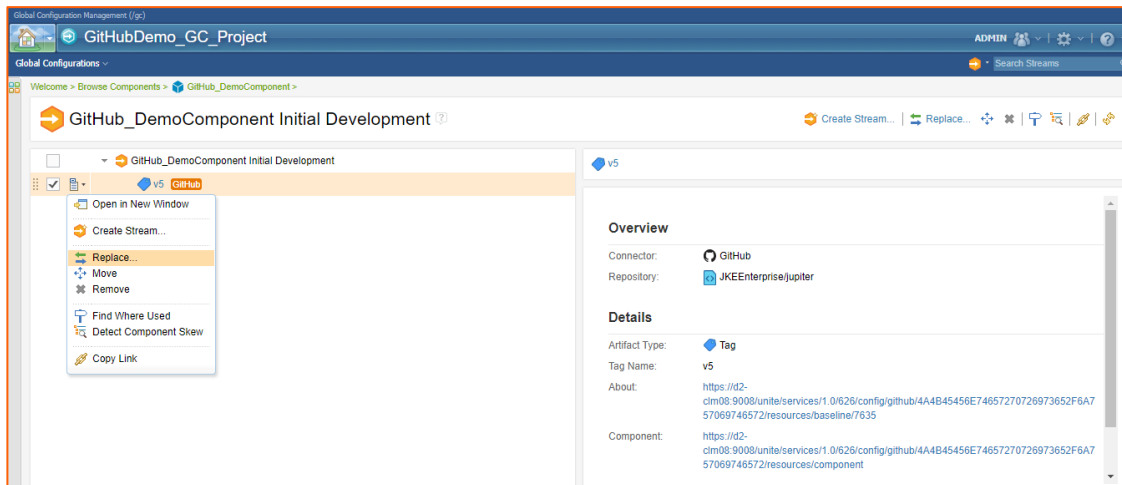


4. Select the branch to see **Overview** and **Details** in the right pane.

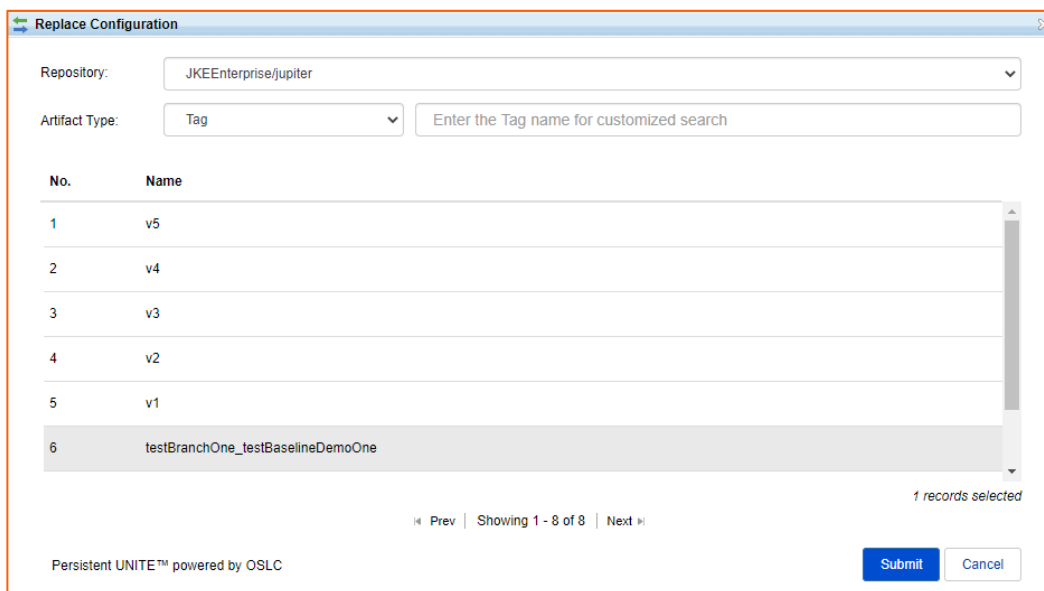


Replacing a tag from GitHub server in a GC stream

1. Open the GC stream that includes the tag to be replaced.
Note: A tag in GitHub is represented as a baseline in GCM application.
2. Select the tag and click **Options > Replace**.
 Replace Configuration page opens.



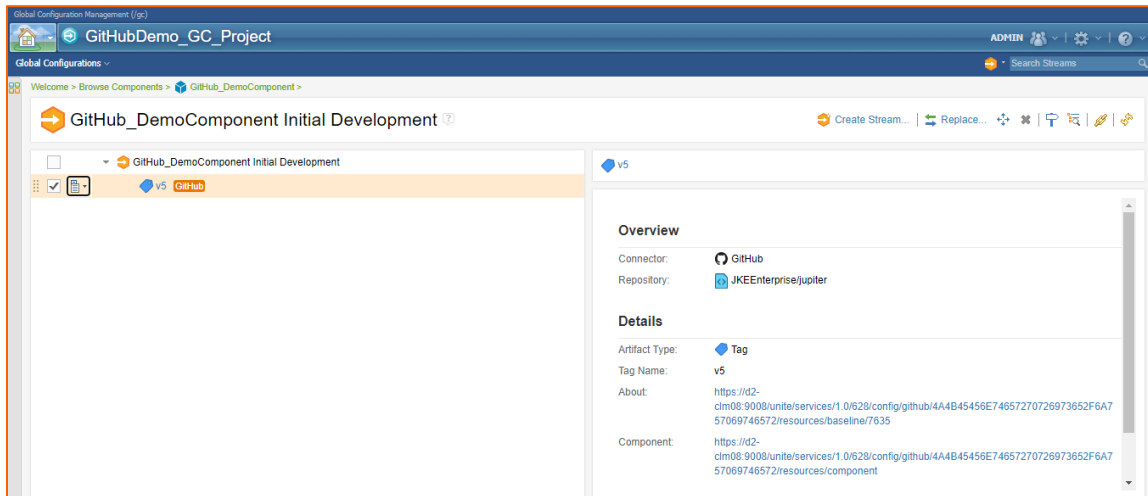
3. Select the new tag to replace the existing tag and click **Submit**.
 The existing tag is replaced.
 The new tag is represented as a baseline under GC stream.



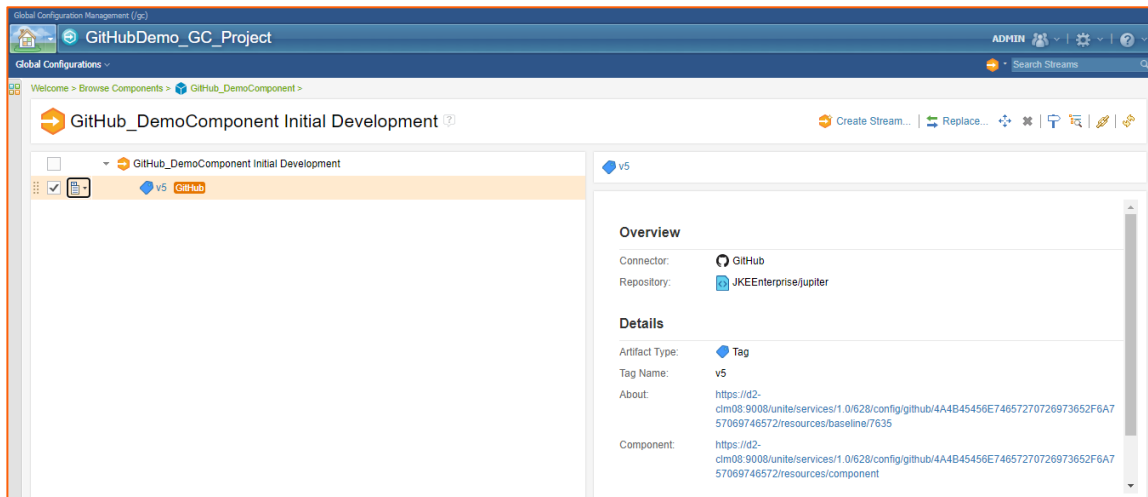
Creating a branch from a tag

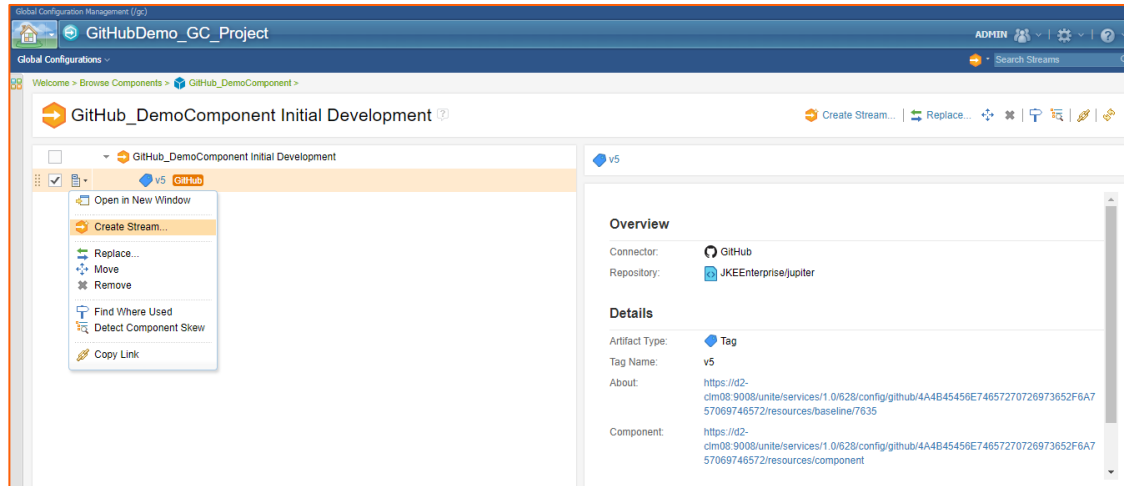
1. Open the GC stream in GCM application.
2. Select the required baseline to create the stream from.

Note: A tag in GitHub is represented as a baseline and a branch in GitHub is represented as a stream in GCM application.

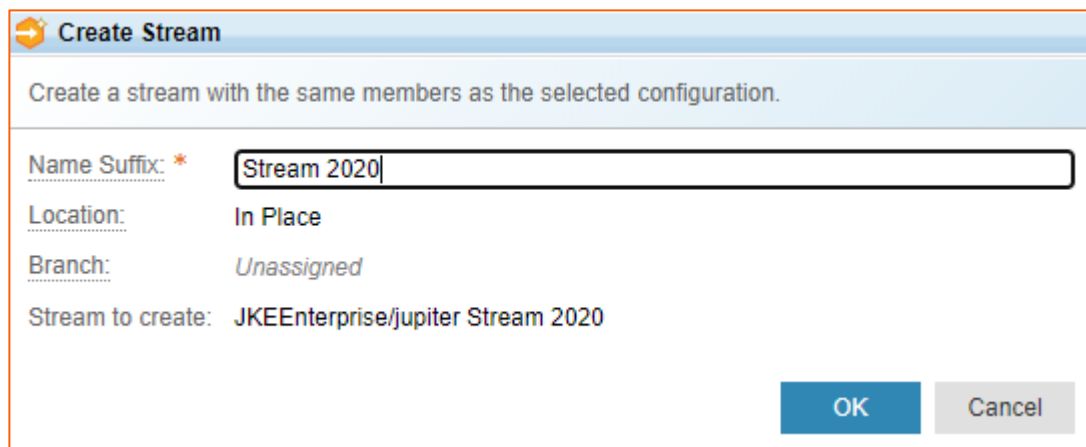


3. Click **Create Stream** or **Options > Create Stream**.
Create Stream window opens.





4. **Name Suffix** populates the auto-generated information (for example, Initial Development). Update the **Name Suffix** with a preferred name.

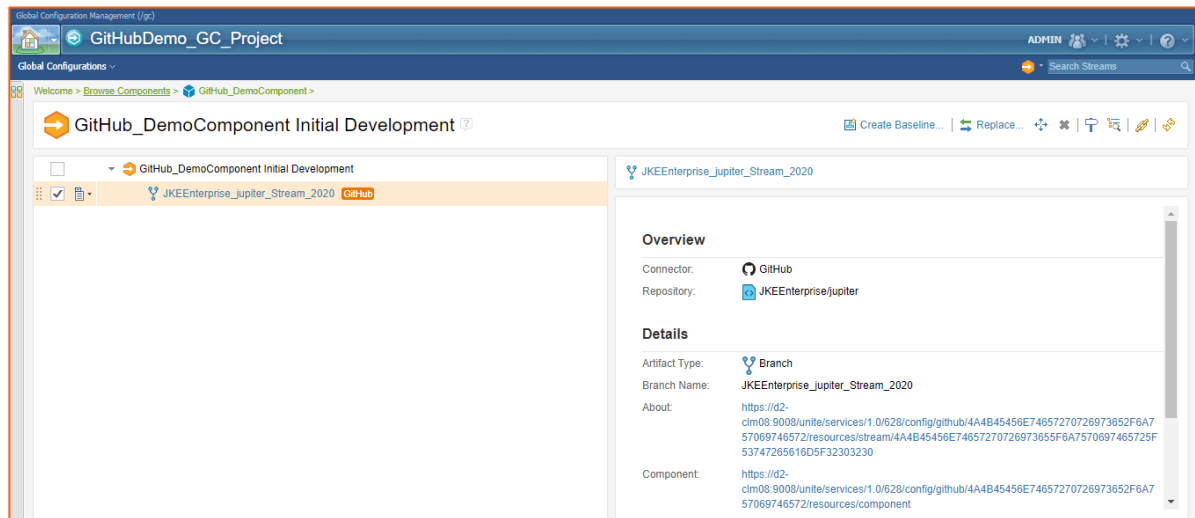


5. Click **OK**. The stream is created.

The new stream is represented as a branch in GitHub.

Note: It takes some time to create the stream. Progress is shown when the stream creation is in progress.

6. Select the branch to see **Overview** and **Details** in the right pane.

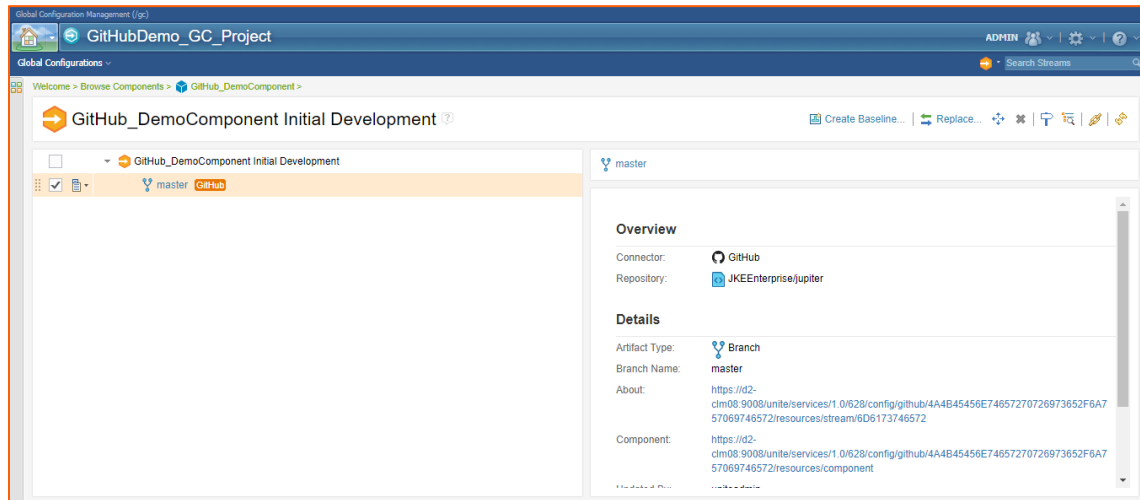


7. Click **About** link to go to the new GitHub branch from the GCM application.
8. Click **Component** link to go to the GitHub repository from the GCM application.

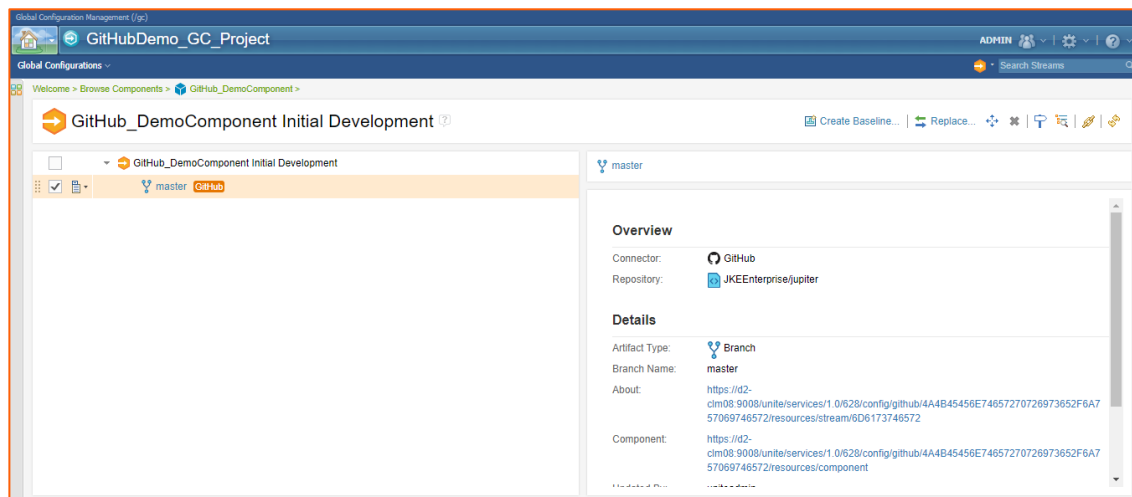
Creating a tag from a branch

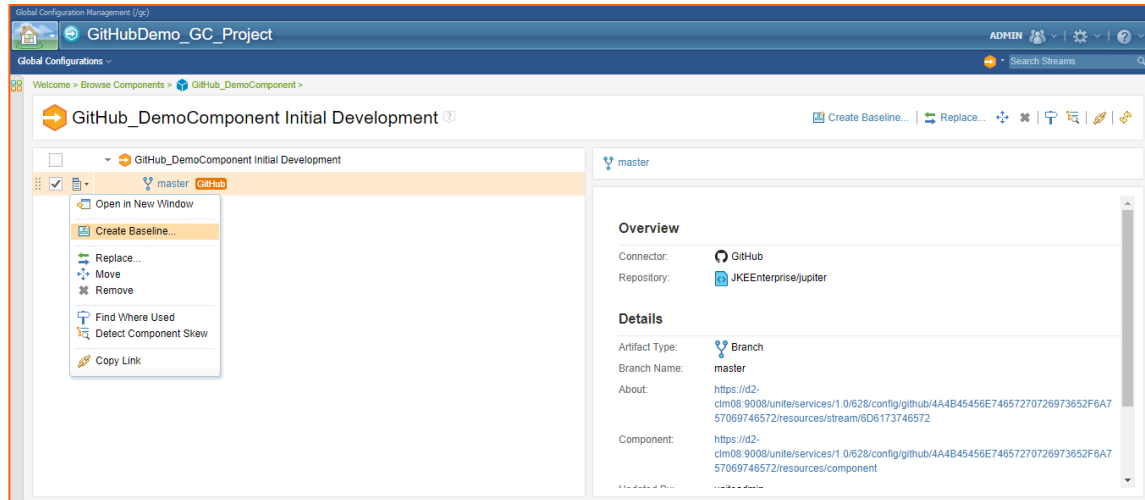
1. Open the GC stream in GCM application.
2. Select the required stream to create the baseline from.

Note: A branch in GitHub is represented as a stream and a tag in GitHub is represented as a baseline in GCM application.

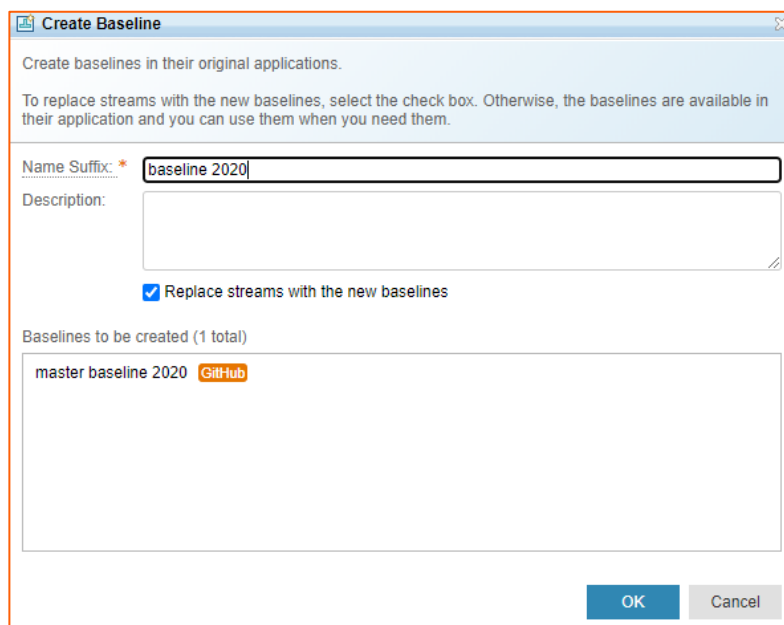


3. Click **Create Baseline** or go to **Options > Create Baseline**.
Create Baseline window opens.



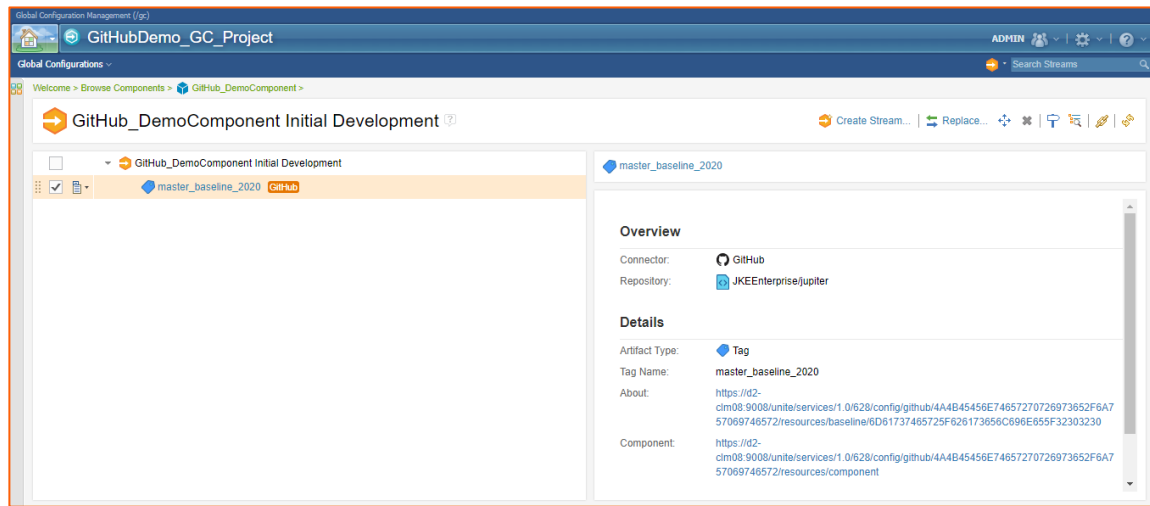


4. **Name Suffix** populates the auto-generated information (for example, 20200811-1454). Update the **Name Suffix** with a preferred name.



5. Click **OK**. The baseline is created.
The new baseline is represented as a tag in GitHub.
Note: It takes some time to create the stream. Progress is shown when the stream creation is in progress.

6. Select the tag to see **Overview** and **Details** in the right pane.



7. Click **About** link to go to the new GitHub tag from the GCM application.
8. Click **Component** link to go to the GitHub repository from the GCM application.

Known issues and workarounds

Branch or tag name convention

Issue: You cannot start the branch or tag name with the term `GitHub`. Also, GitHub does not allow the special characters when you create the branch and tag. *Persistent UNITE for GitHub* replaces “(“ or “)” characters with a space (“ ”) and replaces space (“ ”) or “/” character with an underscore character (“_”).

***Persistent UNITE for GitHub* does not allow multiple GitHub accounts for the ELM user**

Issue: The ELM user must use only one GitHub account to log in to GitHub server through *Persistent UNITE for GitHub* connector.

Workaround: If the ELM user needs to use more than one GitHub account, the user must log out of the ELM application. Also, delete the internet browser cache storage before the user logs in with another GitHub account.



Persistent

Persistent Systems (BSE & NSE: PERSISTENT) builds software that drives our customers' business; enterprises and software product companies with software at the core of their digital transformation.

www.persistent.com