

How to use central vault in Smart Systems Hub Manufacturing-X Learn & Explore Environment

1. Accept GitHub Invitation

- Customer will get email invitation on provided GitHub user email address.
- From their user have to accept the invitation.
- Then customer's GitHub user name will be added to Smart Systems Hub GitHub user account.

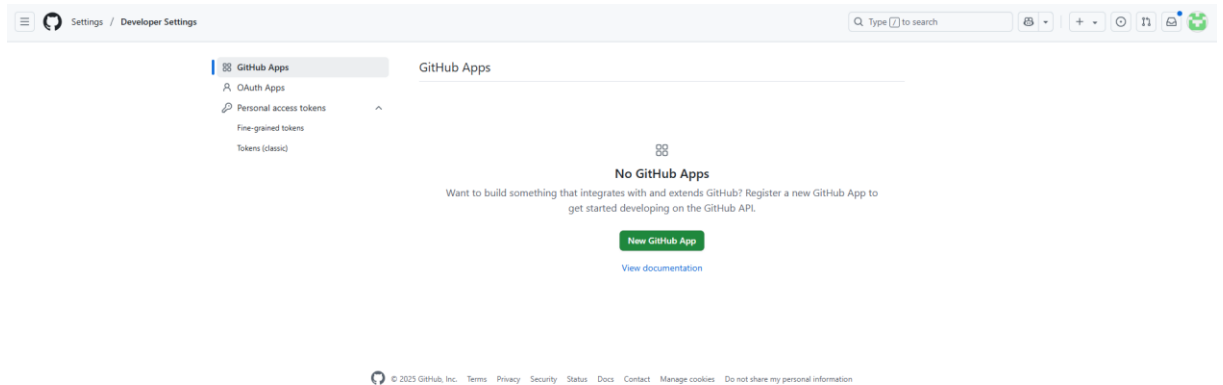
2. Generate token in GitHub

- Firstly, user have to go to his/her GitHub user settings

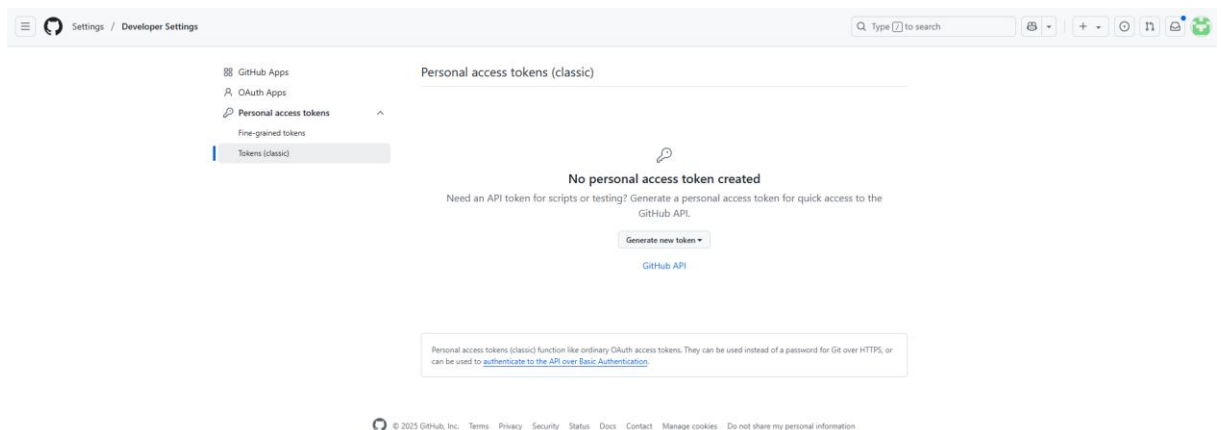
<https://github.com/settings/profile>

The screenshot displays the GitHub user settings page. On the left, a sidebar lists various settings categories: Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys, Organizations, Enterprises, Moderation, Code, planning, and automation, Repositories, Codespaces, Packages, Copilot, Pages, Saved replies, Security, Code security, Integrations, Applications, Scheduled reminders, Archives, Security log, Sponsorship log, and Developer settings. The main content area shows the profile settings, including a profile picture (a green cross on a white background), Name, Public email, Bio, Pronouns, URL, ORCID ID, Social accounts, Company, and Location. The ORCID ID section includes a 'Connect your ORCID ID' button. The Social accounts section has four links to social profiles. The Company section has a text input field. The Location section has a text input field. At the bottom, there is a checkbox for 'Display current local time' and a note about time differences. A footer note states: 'All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to show this data wherever your profile appears. (Learn more about our privacy statement)'.

- Now user have to go to Developer settings. Then inside developer settings user have to go to Personal access tokens.



- Then user have to select Tokens (classic), and then click on generate new token.



- Now user will get the token which he/she have to use to log in into central vault. And then he/she have to go to central vault url.

3. Login to central vault

- URL for central vault: <https://vault.c-27d7c36.kyma.ondemand.com/>



Sign in to Vault

Method

GitHub

GitHub token

More options

Sign in

Contact your administrator for login credentials.

4. See your secrets in central vault

- After logging in to the central vault, the user will be able to see the following



Vault v1.18.1

Secrets engines

cubbyhole/
cubbyhole_9f89b59e
per-token private secret storage

secret/
kv_3fea353c

[Details](#)[View](#)[View](#)

Quick actions

Secrets engines
Supported engines include databases, KV version 2, and PKI.

No mount selected












Select a mount above to get started.

Learn more

Explore the features of Vault and learn advance practices with the following tutorials and documentation.

[Secrets Management](#)[Monitor & Troubleshooting](#)[Build your own Certificate Authority \(CA\)](#)

- Then user have to click on secret/ and then he/she can able to see this:

Secrets Configuration		
<input type="text" value="Search secret path"/>		<input type="button" value="Search"/> <input type="button" value="Create secret +"/>
 FaheemBhatti/		<input type="button" value="..."/>
 aero-tier-1-new/		<input type="button" value="..."/>
 aero-tier-2-new/		<input type="button" value="..."/>
 ax-testbed1/		<input type="button" value="..."/>
 bdrs-secret/		<input type="button" value="..."/>
 initial-check-with-helm-chart-bdrs/		<input type="button" value="..."/>
 prostep-bos/		<input type="button" value="..."/>
 prostep-hadiplast/		<input type="button" value="..."/>
 prostep-sonova/		<input type="button" value="..."/>
 scode-cpu/		<input type="button" value="..."/>
 ssh-internal-test-entity-1/		<input type="button" value="..."/>

- From here user can only able to see his/her assigned folder. If he/she will open any other folder then he/she will able to see

Secrets / secret / FaheemBhatti


 **secret** version 2

[Secrets](#) [Configuration](#)

View secret

Type the path of the secret you want to view.
Include a trailing slash to navigate to the list view.

View list

 You do not have the required permissions or the directory does not exist.

- When user will click on his/her folder then he/she can able to see:

Secrets / secret / scode-cpu

 **secret** version 2

[Secrets](#) [Configuration](#)

 Search

Create secret +

 [apikey](#)

...

 [edc-client-secret](#)

...

 [private-key-alias](#)

...

 [public-key-alias](#)

...

1-4 of 4 < 1 >

- Then from here he/she can see any secrets. Example for apikey:

Secrets / secret / scode-cpu / apikey

scode-cpu/apikey

[Overview](#) [Secret](#) [Metadata](#) [Paths](#) [Version History](#)

Current version

The current version of this secret.

1

[Create new](#) +

Secret age


Current secret version age. Last updated on Mar 20 2025, 11:49:08 AM.


2 months

[View metadata](#) →

Paths

The paths to use when referring to this secret in API or CLI.

API path  /v1/secret/data/scode-cpu/apikey

CLI path  -mount="secret" "scode-cpu/apikey"

- Then he/she have to click on Secret and then he/she can able to see:




Secrets / secret / scode-cpu / apikey

scode-cpu/apikey

[Overview](#) [Secret](#) [Metadata](#) [Paths](#) [Version History](#)

☐ JSON

[Delete](#)
[Destroy](#)
[Copy](#)
[Version 1](#)
[Create new version](#) +

Key	Value	Version 1 created Mar 20, 2025 11:49 AM
content	   ■■■■■■■■■■	

- From here he/she have to click on eye icon then he/she can able to see secrets

5. Add secret to central vault

- User can add his/her secret inside his/her assigned folder

- To add secret user have to click on Create secret:

Secrets / secret / scode-cpu

secret version 2

Secrets Configuration

scode-cpu/ Search Create secret +

apikey	...
edc-client-secret	...
private-key-alias	...
public-key-alias	...

1-4 of 4 < 1 >

- Then user can able to see this:

Secrets / secret / scode-cpu / Create

Create Secret

☐ JSON

Path for this secret
Names with forward slashes define hierarchical path structures.

scode-cpu/

Secret data

key		👁	Add
-----	--	---	-----

✓ Show secret metadata

Save Cancel

- Here user have to write his/her secret name and then Secret data with key and value. And then user have to click on save button.

- User can also turn on JSON-button to create the secret:

Secrets / secret / scode-cpu / Create

Create Secret

☒ JSON

Path for this secret

Names with forward slashes define hierarchical path structures.

scode-cpu/

Secret data



```
1 {  
2   "": ""  
3 }
```

▼ [Show secret metadata](#)

Save

Cancel