Advanced Live Training:

Scripted Actions
Lab Guide

Revision 1

June 20, 2024

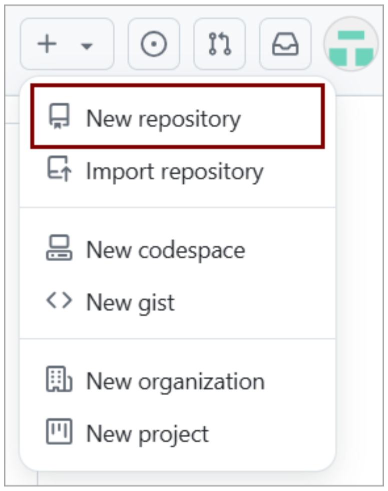
Contents

1. Set up GitHub

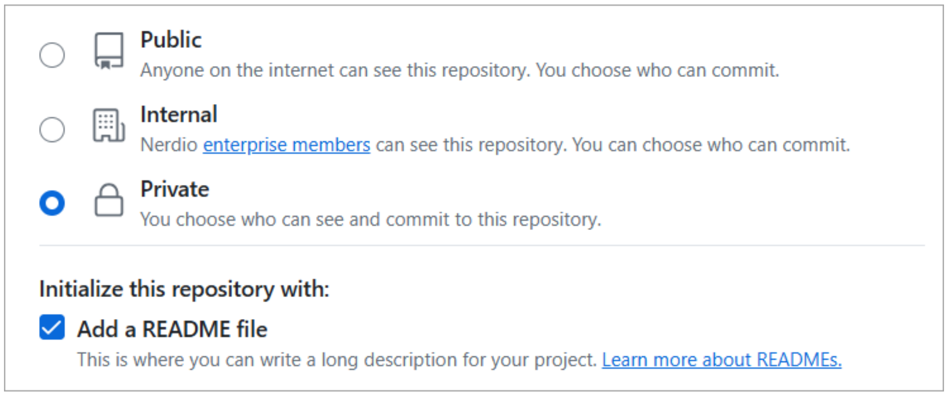
The first step is to create a repository in GitHub. The repository should be private and include a `Readme.md` file.

To create a new GitHub repository:

1. Sign in to your GitHub account.
2. In the upper-right corner of any page, select **Create new** `+ ▾`, and then select **New repository**.



3. In the **Create a new repository** dialog box, provide the required details.
   - Set the repository privacy to **Private**.
   - Select the **Add a README file** option.



Public
Anyone on the internet can see this repository. You choose who can commit.

Internal
Nerdio enterprise members can see this repository. You can choose who can commit.

[✓ Private]
You choose who can see and commit to this repository.

Initialize this repository with:

✓ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

4. Select **Create repository**.

For more details, see Creating a new repository.

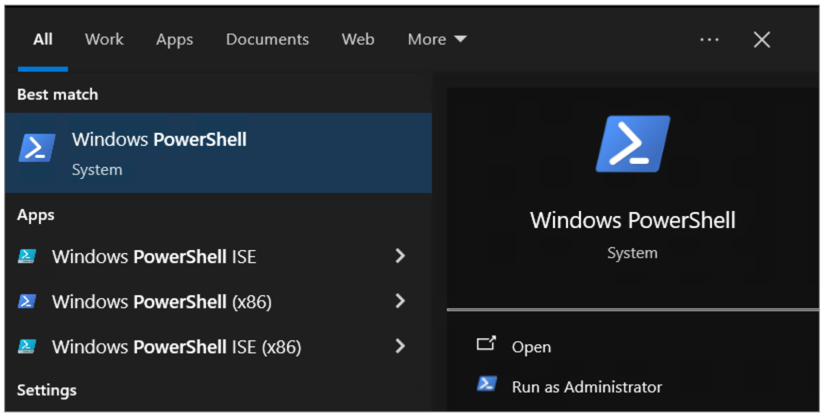
2. Set up Visual Studio Code

The next step is to set up integration for Visual Studio Code and GitHub.

Important! To complete this step, you need to have the **GitHub Pull Requests** extension installed. To install the extension, go to Visual Studio Marketplace.

To set up the integration:

1. In Windows Search, enter `powershell`, and then select the app.



Note: Select **Install** if the app has not been installed yet.

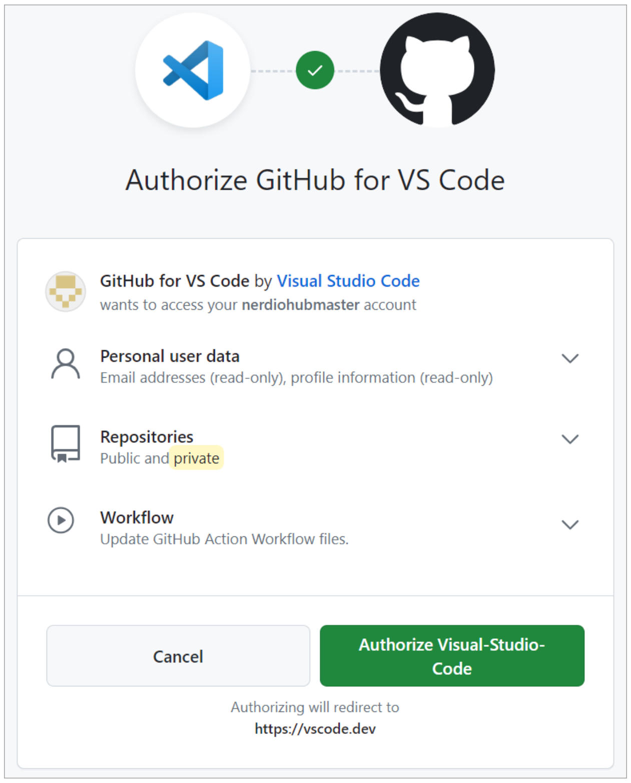
2. Open Visual Studio Code, and then sign in to GitHub to allow VS Code to authenticate to your remote repository in GitHub:

   a. In the VS Code sidebar, in the lower-left corner, select **Accounts** .

   b. Provide your GitHub account credentials, and then select **Confirm**.

   Important! The GitHub account username is not your email address.

c. Select **Authorize Visual-Studio-Code**.


Authorize GitHub for VS Code

GitHub for VS Code by Visual Studio Code
wants to access your nerdiohubmaster account

Personal user data
Email addresses (read-only), profile information (read-only)

Repositories
Public and private

Workflow
Update GitHub Action Workflow files.

Authorizing will redirect to
https://vscode.dev

Tip: If the authentication fails, run the following commands in the terminal:

- `git config --global user.name "Your Name"`
- `git config --global user.email yourname@example.com`

3. In your local environment, use Windows File Explorer to create a directory that you can use as the root for your local repository.

Example: Create a folder named `Repo` in the root of OneDrive, or the `Documents` folder.

4. Once the repository is created and Git-authorized, in Visual Studio Code, clone that repository:

   a. Press `Ctrl + Shift + P` to open the `Show-Command` prompt at the top of Visual Studio Code.

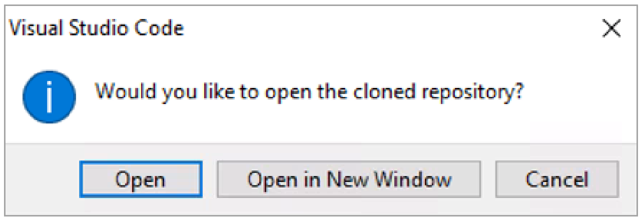
   b. Enter `git`, and then select `Git:Clone`.

   c. Select the `Clone from GitHub` prompt.

   d. Select the repository that you previously created.

Note: The selected folder is where the repository should be cloned. Ensure it is the folder that you previously created via the File Explorer. Please see the above example for the original folder creation.

e. Once you get a prompt to open the repository, select **Open**.


Visual Studio Code

Would you like to open the cloned repository?

If the operation completes successfully, a new folder is displayed in the File Explorer. It should be of the same name as the repository you created in GitHub, and must contain a `readme.md` file.

5. With the current project opened in Visual Studio Code, right-click in the **Explorer** view, and then select **Create new folder**. Name the folder `windows`.
6. Repeat the same steps to create the folder named `runbooks`.

Note: In the **Sync** view, the new folders should be displayed as uncommitted items in the queue.

7. Commit the folders: In the message text box, enter `Initial commit`, and then expand the drop-down menu (white arrow) and select **Commit & Sync**.
8. Once the **Commit & Sync** operation is completed, go back to the browser, and refresh the GitHub page.

The new folders should now be displayed in GitHub, and you can verify that the **Commit & Sync** operation is performing as expected.

3. Set up Nerdio Manager

Finally, you need to set up integration in Nerdio Manager. For this, complete the following steps:

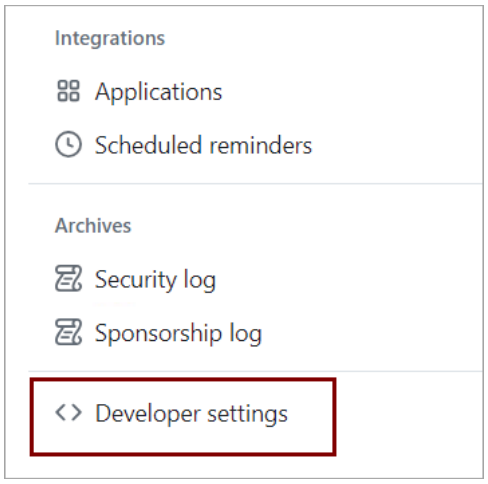
- Secure the repository token
- Enable Azure Runbooks scripted actions

3.1 Secure the Repository Token

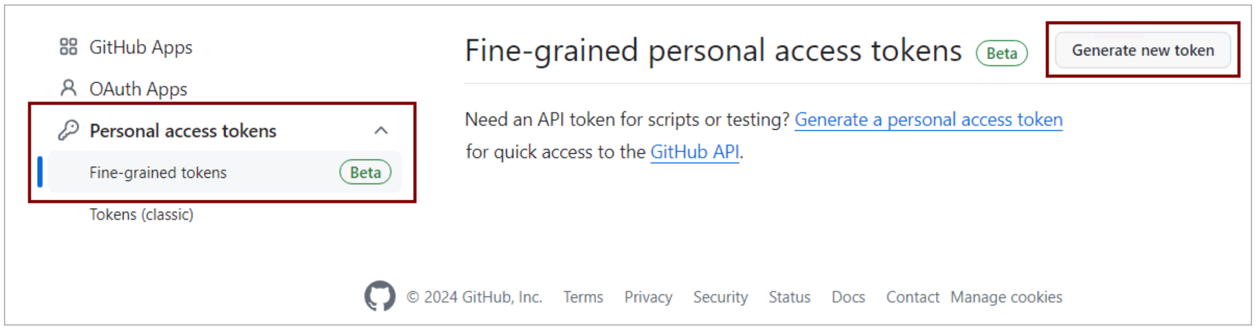
Personal access tokens are an alternative to using passwords for authentication to GitHub when using the GitHub API or the command line. They are intended to access GitHub resources on your behalf.

To secure a token:

1. In GitHub, select your **profile icon** > **Settings**.
2. In the left sidebar, select **Developer Settings**.

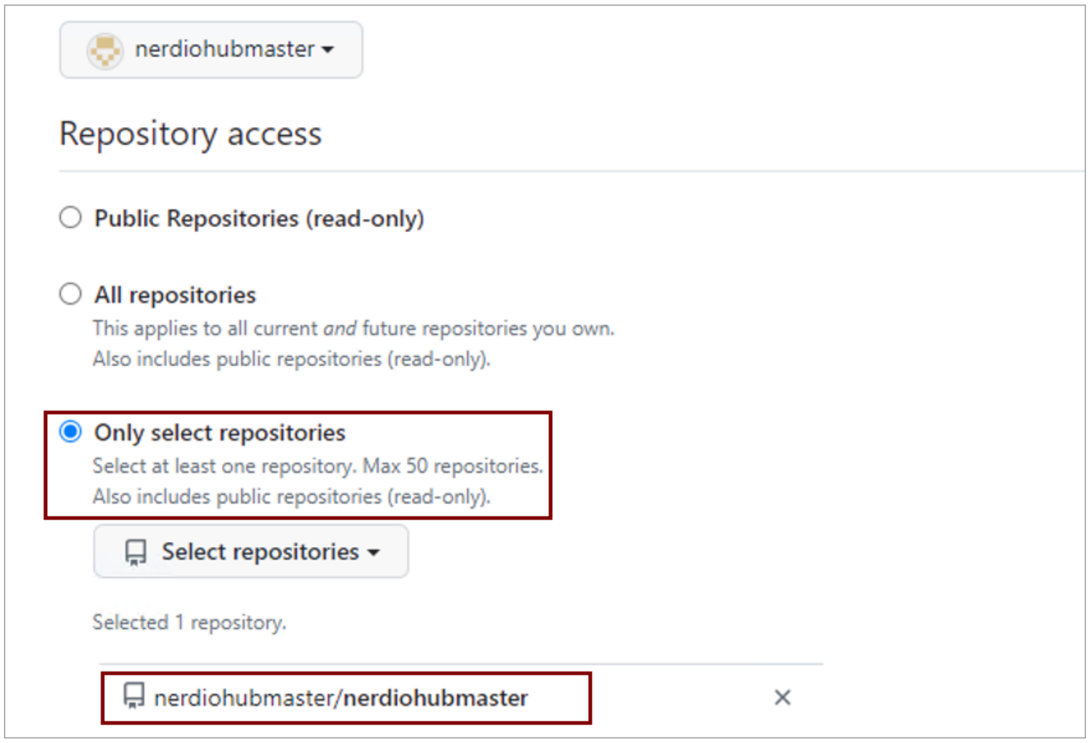


3. Select **Personal access tokens** > **Fine-grained tokens** > **Generate new token**.



Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

4. In the new dialog box, provide the required details.
   - **Repository access**: Select **Only select repositories**.
     - **Select repositories**: Select the repository that you want the token to access.



nerdiohubmaster

Repository access

Public Repositories (read-only)

All repositories
This applies to all current *and* future repositories you own.
Also includes public repositories (read-only).

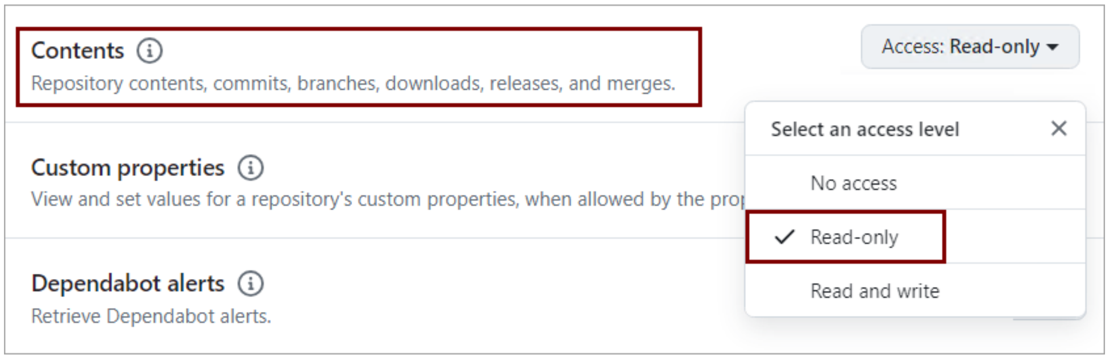
[✓ Only select repositories]
Select at least one repository. Max 50 repositories.
Also includes public repositories (read-only).

Select repositories

Selected 1 repository.

nerdiohubmaster/nerdiohubmaster

   - **Permissions**: Expand the **Repository permissions** menu, and then define the following:
     - **Contents**: Expand the **Access** menu and select **Read-only**.



Contents
Repository contents, commits, branches, downloads, releases, and merges.

Access: Read-only

Select an access level

No access

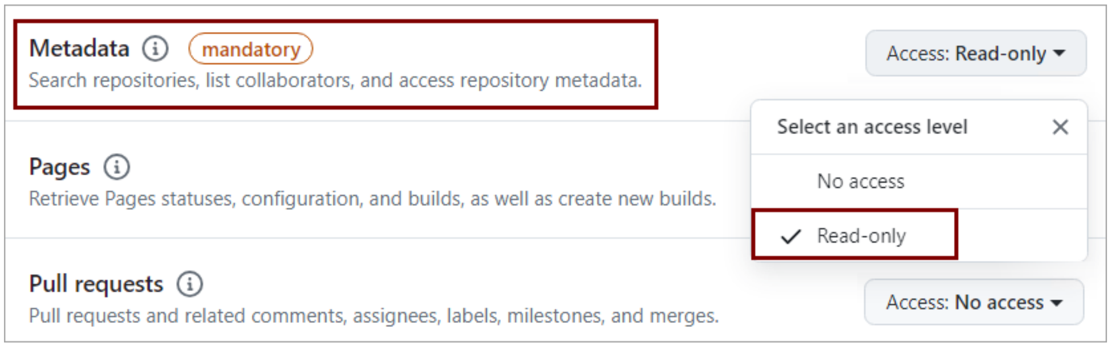
[✓ Read-only]

Read and write

Custom properties
View and set values for a repository's custom properties, when allowed by the prop

Dependabot alerts
Retrieve Dependabot alerts.

- **Metadata**: Expand the **Access** menu and select **Read-only** (this option is required).



Metadata ⓘ mandatory
Search repositories, list collaborators, and access repository metadata.

Access: Read-only

Select an access level
No access
✓ Read-only

Pages ⓘ
Retrieve Pages statuses, configuration, and builds, as well as create new builds.

Pull requests ⓘ
Pull requests and related comments, assignees, labels, milestones, and merges.

Access: No access

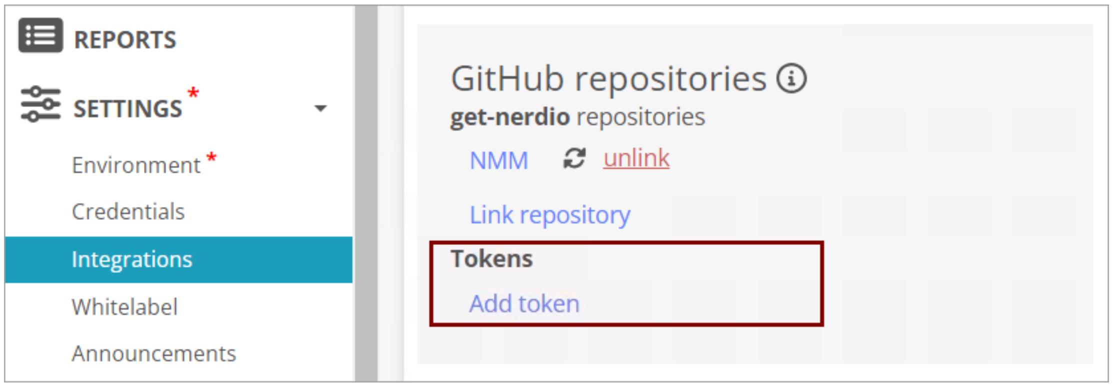
5. Select **Generate token**.

6. Save the token to Notepad, and then next to the token secret, select **Copy**.

Warning: Do not close the token window until the token is saved in Nerdio Manager.

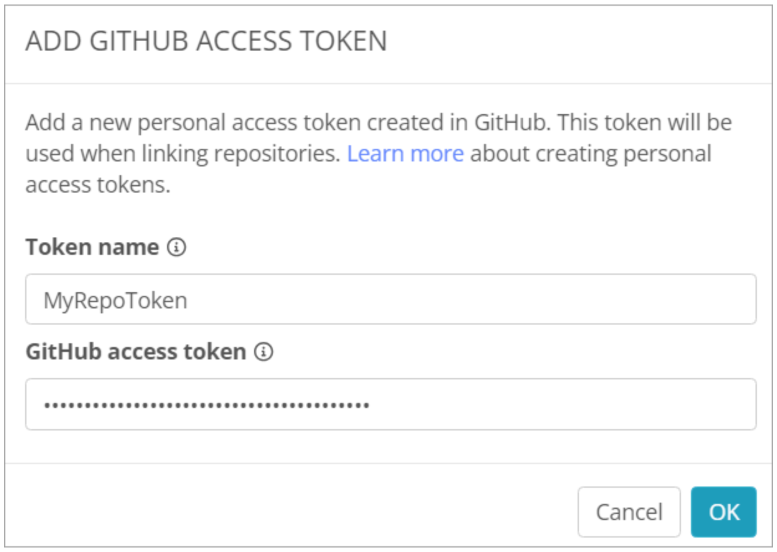
7. In Nerdio Manager, at the MSP level, go to **Settings > Integrations**.

8. In the **GitHub repositories** section, select **Add token**.



GitHub repositories ⓘ
get-nerdio repositories
NMM unlink
Link repository
Tokens
Add token

9. In the new dialog box, provide the token name and paste the GitHub token that you previously saved to Notepad. Select **OK**.



ADD GITHUB ACCESS TOKEN

Add a new personal access token created in GitHub. This token will be used when linking repositories. Learn more about creating personal access tokens.

Token name: MyRepoToken

GitHub access token: ••

10. Above **Tokens**, select **Link repository**.

11. In the new dialog box, provide the following details:

   - **GitHub Account**: Enter your username for the GitHub account.

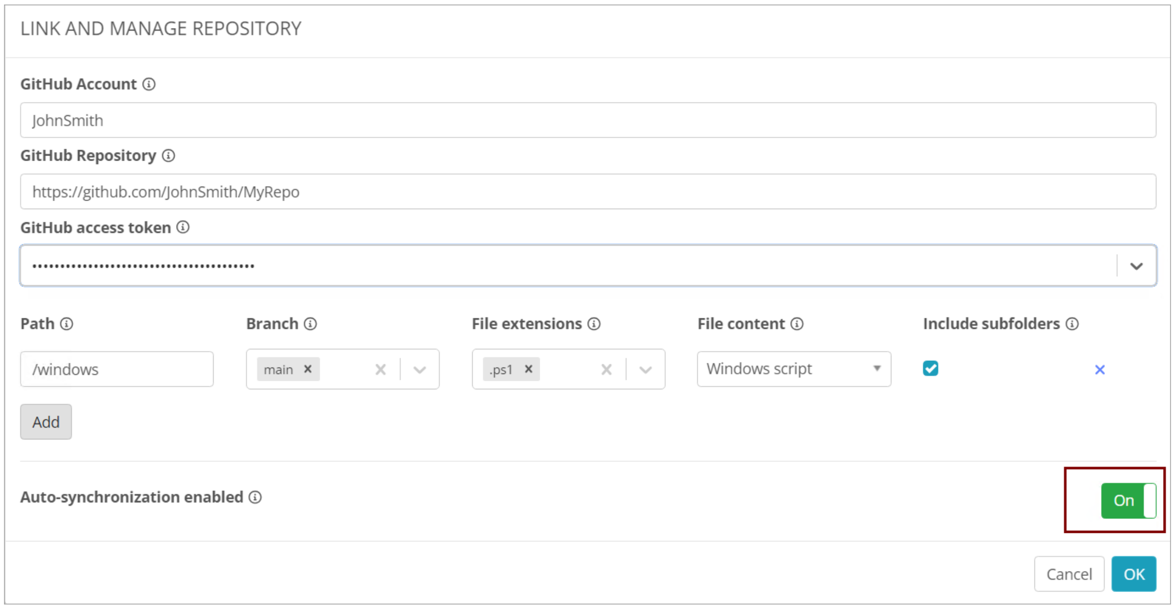
     Important! The GitHub account username is not your email address.

   - **GitHub Repository**: Provide the path to your GitHub repository.
   - **GitHub access token**: From the drop-down list, select the token that you previously added.
   - Add paths for the following scripted actions to mirror two folders that were previously created:

Windows script	• **Path**: `/windows` • **Branch**: The `main` branch is predefined and can remain as default. • **File extension**: The `ps1` extension is predefined and can remain as default. • **File content**: `Windows script` is predefined and can remain as default. • Select **Add**.

Azure runbook	• **Path**: `/runbooks` • **Branch**: The `main` branch is predefined and can remain as default. • **File extension**: The `ps1` extension is predefined and can remain as default. • **File content**: `Azure runbook` is predefined and can remain as default. • Select **Add**.

- Ensure the **Auto-synchronization enabled** option is **ON**, and then select **OK**.



LINK AND MANAGE REPOSITORY

GitHub Account: JohnSmith

GitHub Repository: https://github.com/JohnSmith/MyRepo

GitHub access token: ••

Path	Branch	File extensions	File content	Include subfolders
/windows	main	.ps1	Windows script	✓

3.2 Enable Azure Runbooks Scripted Actions

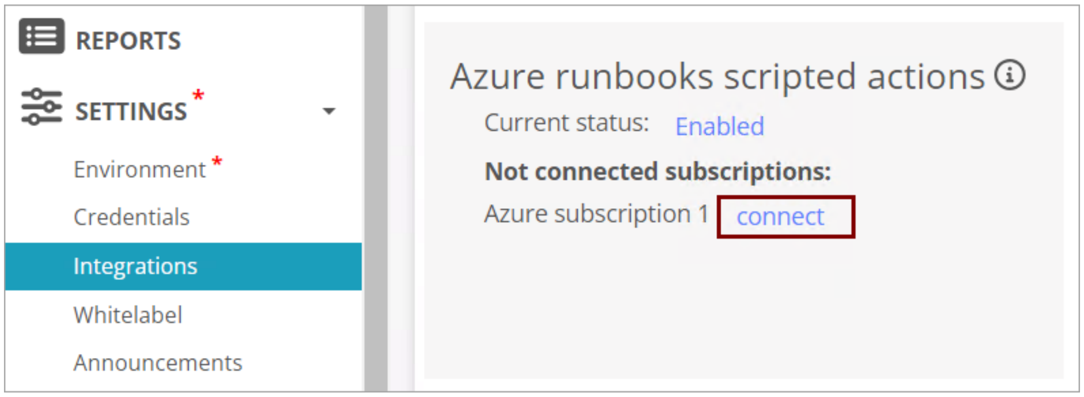
The next step is to enable Azure runbooks scripted actions in Nerdio Manager.

To enable Azure runbooks:

1. Download the certificate from Nerdio Manager:

    a. Go to **Settings** > **Integrations**, and in the **Azure runbooks scripted actions** section, ensure the **Current status** field is set to **Enabled**.

    b. Select **connect** to connect your subscription.



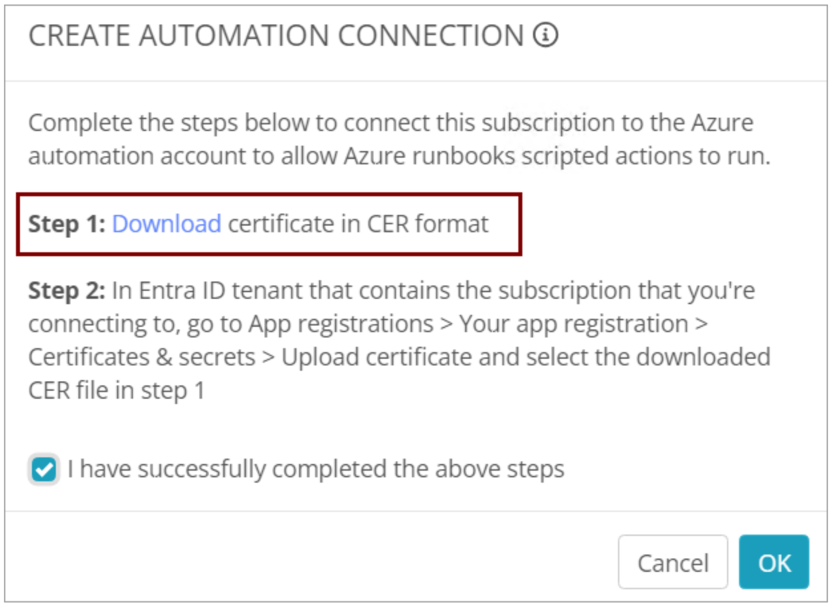
Azure runbooks scripted actions

Current status: Enabled

Not connected subscriptions:

Azure subscription 1 connect

    c. In the **Create automation connection** dialog box, select **Download**, and then save the certificate to your downloads location.



CREATE AUTOMATION CONNECTION

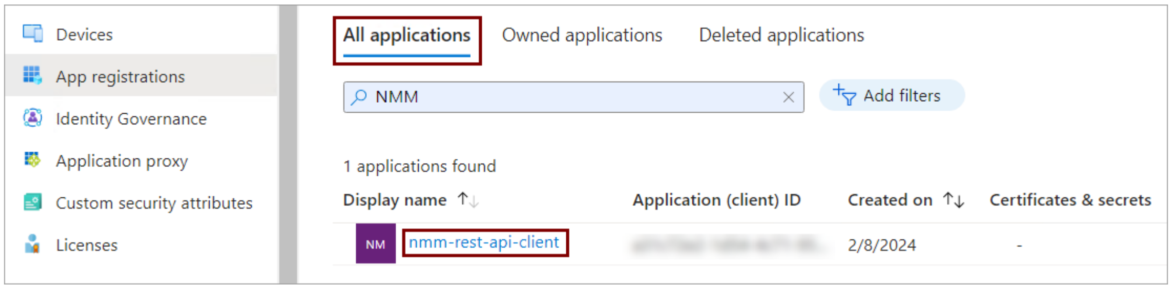
Complete the steps below to connect this subscription to the Azure automation account to allow Azure runbooks scripted actions to run.

Step 1: Download certificate in CER format

Step 2: In Entra ID tenant that contains the subscription that you're connecting to, go to App registrations > Your app registration > Certificates & secrets > Upload certificate and select the downloaded CER file in step 1

✓ I have successfully completed the above steps

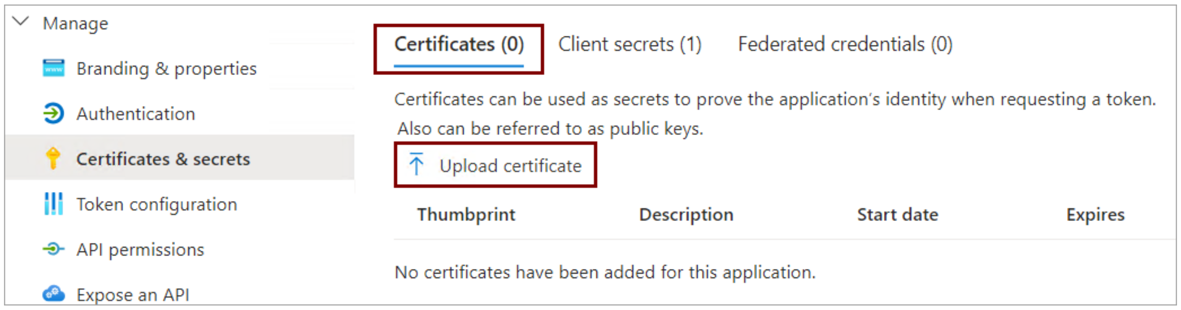
2. Upload the certificate to the Azure Portal:
    a. Open the Azure Portal where Nerdio Manager is installed, and navigate to **Microsoft Entra ID**.
    b. In the side pane, go to **Manage** > **App registrations**.
    c. On the **All applications** tab, in the **Display name** section, select the Nerdio Manager web app.



1 applications found

Display name	Application (client) ID	Created on	Certificates & secrets
NM nmm-rest-api-client	[redacted]	2/8/2024	-

    d. On the Nerdio Manager app service page, go to **Manage** > **Certificates & secrets**.
    e. On the **Certificates** tab, select **Upload certificate**.



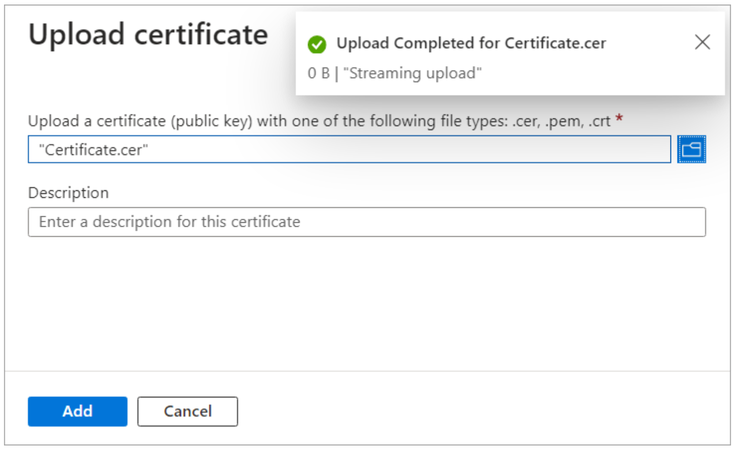
Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Description	Start date	Expires

No certificates have been added for this application.

    f. In the **Upload certificate** pane that opens on the right, click **Select a file**, and then browse to your downloads location where you previously saved the certificate. Select the certificate to upload it to the portal.

g. Optionally, add the description, and then select **Add**.



Upload certificate

Upload Completed for Certificate.cer

0 B | "Streaming upload"

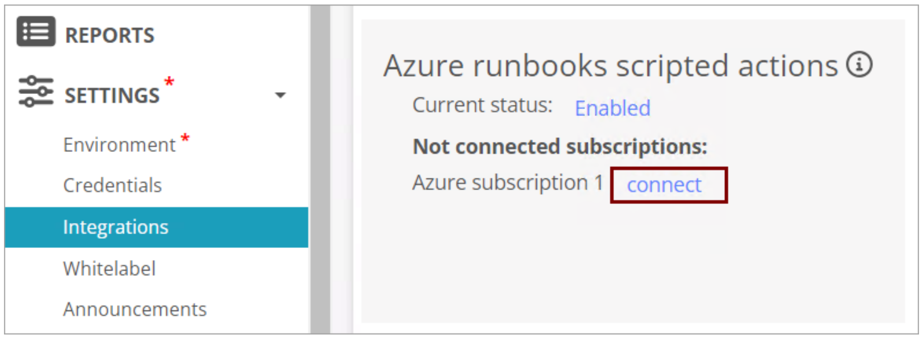
Upload a certificate (public key) with one of the following file types: .cer, .pem, .crt *

"Certificate.cer"

Description

Enter a description for this certificate

3. Go back to **Nerdio Manager** > **Settings** > **Integrations**.
4. In the **Azure runbooks scripted actions** section, select **connect** again.



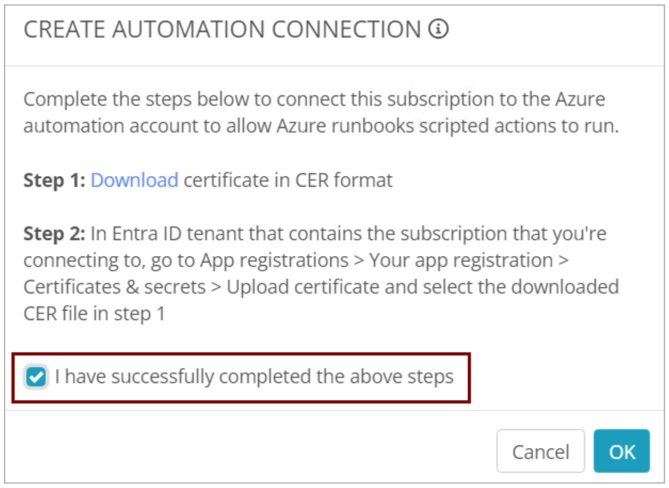
Azure runbooks scripted actions

Current status: Enabled

Not connected subscriptions:

Azure subscription 1 connect

5. In the **Create automation connection** dialog box, select the checkbox to confirm that you have completed the above steps. Select **OK**.



CREATE AUTOMATION CONNECTION

Complete the steps below to connect this subscription to the Azure automation account to allow Azure runbooks scripted actions to run.

Step 1: Download certificate in CER format

Step 2: In Entra ID tenant that contains the subscription that you're connecting to, go to App registrations > Your app registration > Certificates & secrets > Upload certificate and select the downloaded CER file in step 1

☑ I have successfully completed the above steps

Both GitHub and Azure runbooks are now enabled in your Nerdio Manager **Integrations** module.