A black background with blue and yellow text

Description automatically generated

Advanced Live Training:

Scripted Actions

Lab Guide

Revision 1

**June 20, 2024**

Contents

[1. Set up GitHub 2](#_Toc169734699)

[2. Set up Visual Studio Code 3](#_Toc169734700)

[3. Set up Nerdio Manager 6](#_Toc169734701)

[3.1 Secure the Repository Token 6](#_Toc169734702)

[3.2 Enable Azure Runbooks Scripted Actions 11](#_Toc169734703)

# 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**.

A screenshot of a computer

Description automatically generated

1. 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.

A screenshot of a computer

Description automatically generated

1. Select **Create repository**.

For more details, see [Creating a new repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository).

# 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** extensioninstalled. To install the extension, go to [Visual Studio Marketplace](https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github). |

To set up the integration:

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

A screenshot of a computer

Description automatically generated

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

1. Open Visual Studio Code, and then sign in to GitHub to allow VS Code to authenticate to your remote repository in GitHub:
2. In the VS Code sidebar, in the lower-left corner, select **Accounts** .
3. Provide your GitHub account credentials, and then select **Confirm**.

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

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

A screenshot of a computer

Description automatically generated

|  |
| --- |
| **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](mailto:yourname@example.com) |

1. 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. |

1. Once the repository is created and Git-authorized, in Visual Studio Code, clone that repository:
2. Press Ctrl + Shift + P to open the Show-Command prompt at the top of Visual Studio Code.
3. Enter git, and then select Git:Clone.
4. Select the Clone from GitHub prompt.
5. 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. |

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

A screenshot of a computer

Description automatically generated

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.

1. 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**.**
2. 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. |

1. Commit the folders: In the message text box, enter Initial commit, and then expand the drop-down menu (white arrow) and select **Commit & Sync**.
2. 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.

# Set up Nerdio Manager

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

* [Secure the repository token](#_Secure_the_Repo)
* [Enable Azure Runbooks scripted actions](#_Enable_Azure_Runbooks)

## 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**.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. 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.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. Select **Generate token**.
2. 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. |

1. In Nerdio Manager, at the MSP level, go to **Settings > Integrations**.
2. In the **GitHub repositories** section, select **Add token**.

A screenshot of a computer

Description automatically generated

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

A screenshot of a login page

Description automatically generated

1. Above **Tokens**, select **Link repository**.
2. 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**.

A screenshot of a computer

Description automatically generated

## 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:
2. Go to **Settings** > **Integrations**, and in the **Azure runbooks scripted actions** section, ensure the **Current status** field is set to **Enabled**.
3. Select **connect** to connect your subscription.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. Upload the certificate to the Azure Portal:
2. Open the [Azure Portal](https://portal.azure.com/#home) where Nerdio Manager is installed, and navigate to **Microsoft** **Entra ID**.
3. In the side pane, go to **Manage** > **App registrations**.
4. On the **All applications** tab, in the **Display name** section, select the Nerdio Manager web app.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

1. 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.
2. Optionally, add the description, and then select **Add**.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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