Dev Containers

The Dev Containers lets you use a container as a full-featured development environment inside Visual Studio Code. You can open any folder inside (or mounted into) a container and take advantage of VSCode's full feature set. A devcontainer.json file in your project tells VS Code how to access (or create) a development container with a well-defined tool and runtime stack. In short, a Dev Container allows you to set up a development environment and tooling within a docker container and interact with that container via VS Code.

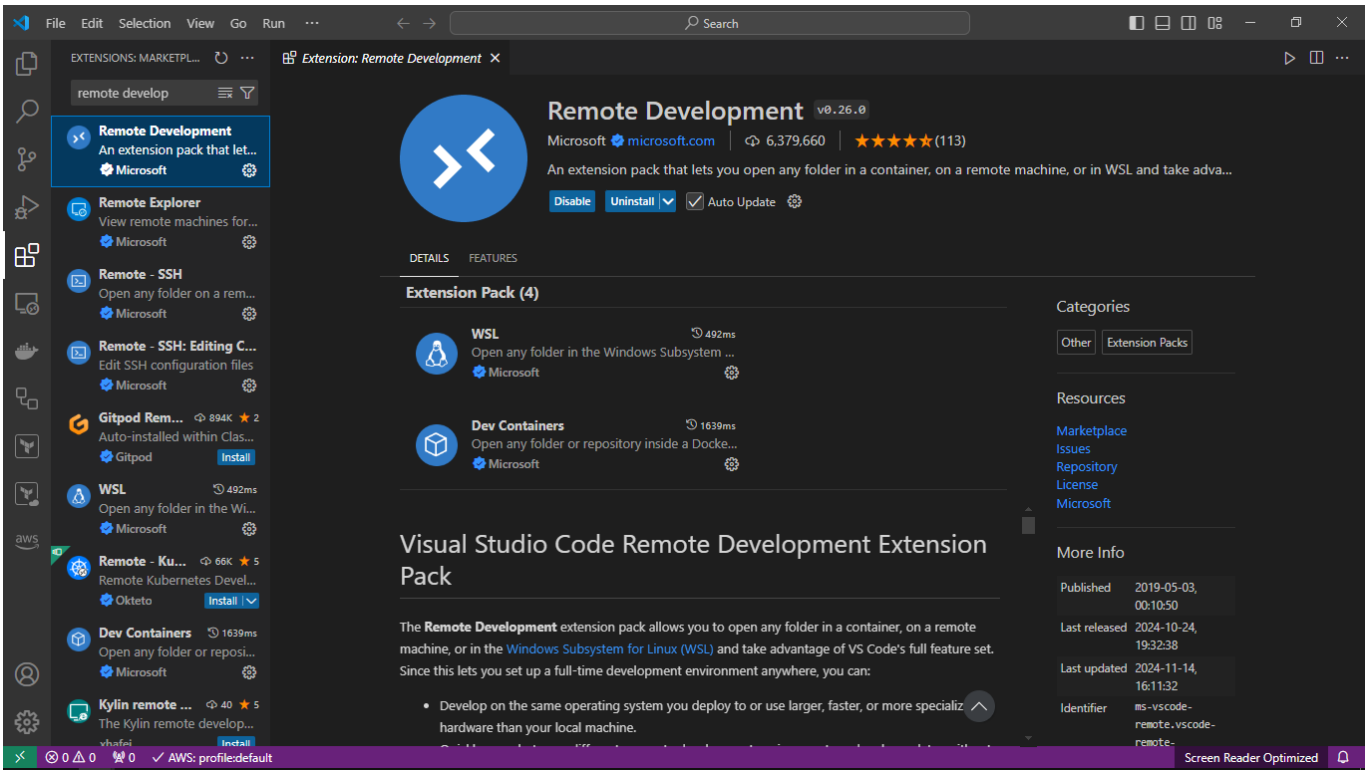
Pre-requisites

1. Install & start Docker Desktop.
2. VSCode installed

> Make sure **WSL 2** is enabled in WSL Integration inside Resources which is present in Docker Desktop settings.

Install Remote Development Extension

1. Open VSCode.
2. Open *Extensions* tab.
3. Type `remote development` or paste this `ms-vscode-remote.vscode-remote-extensionpack` extension id.
4. Install this extension.



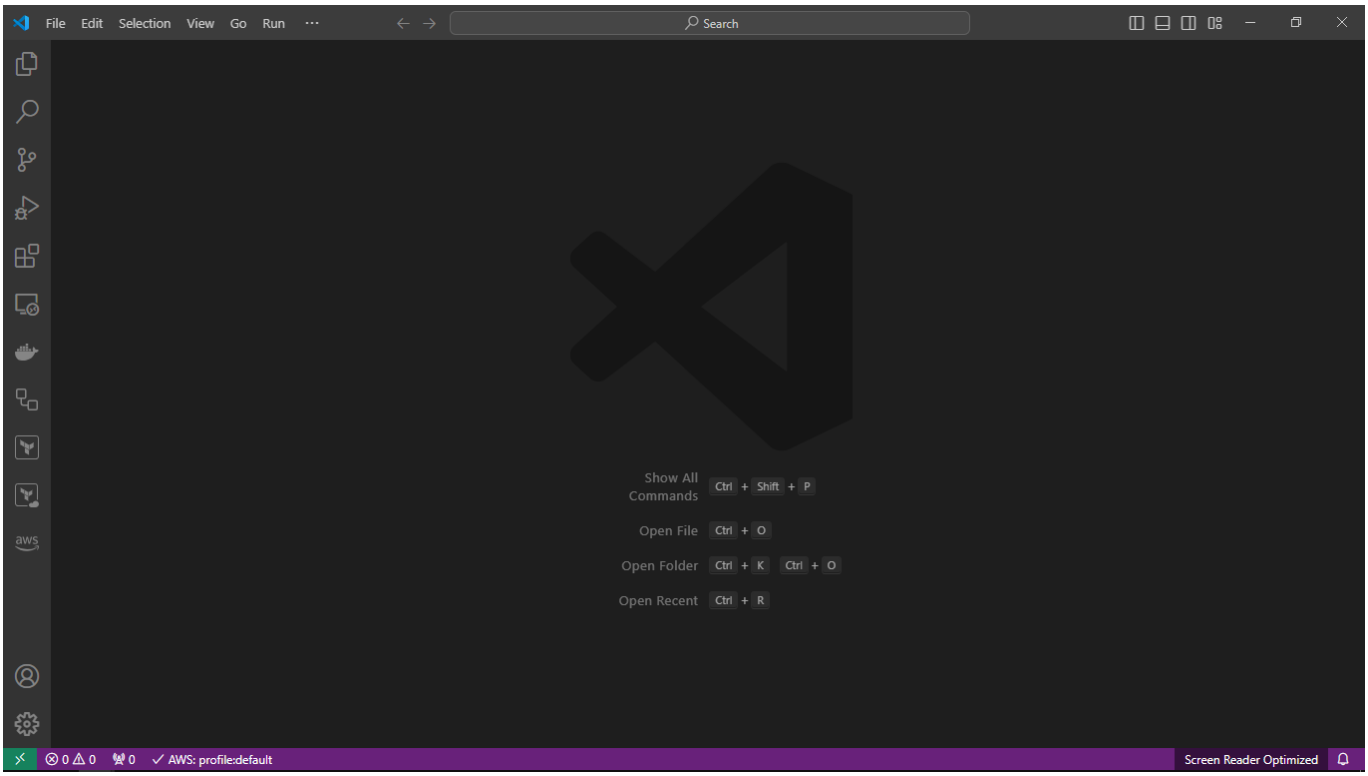
5. Once you install the remote development extension, a new symbol named **Remote Host** and labeled **Open a Remote Window** will be added to the VSCode status bar.

There are multiple ways to start the *dev-container*, choose according to your setup requirements

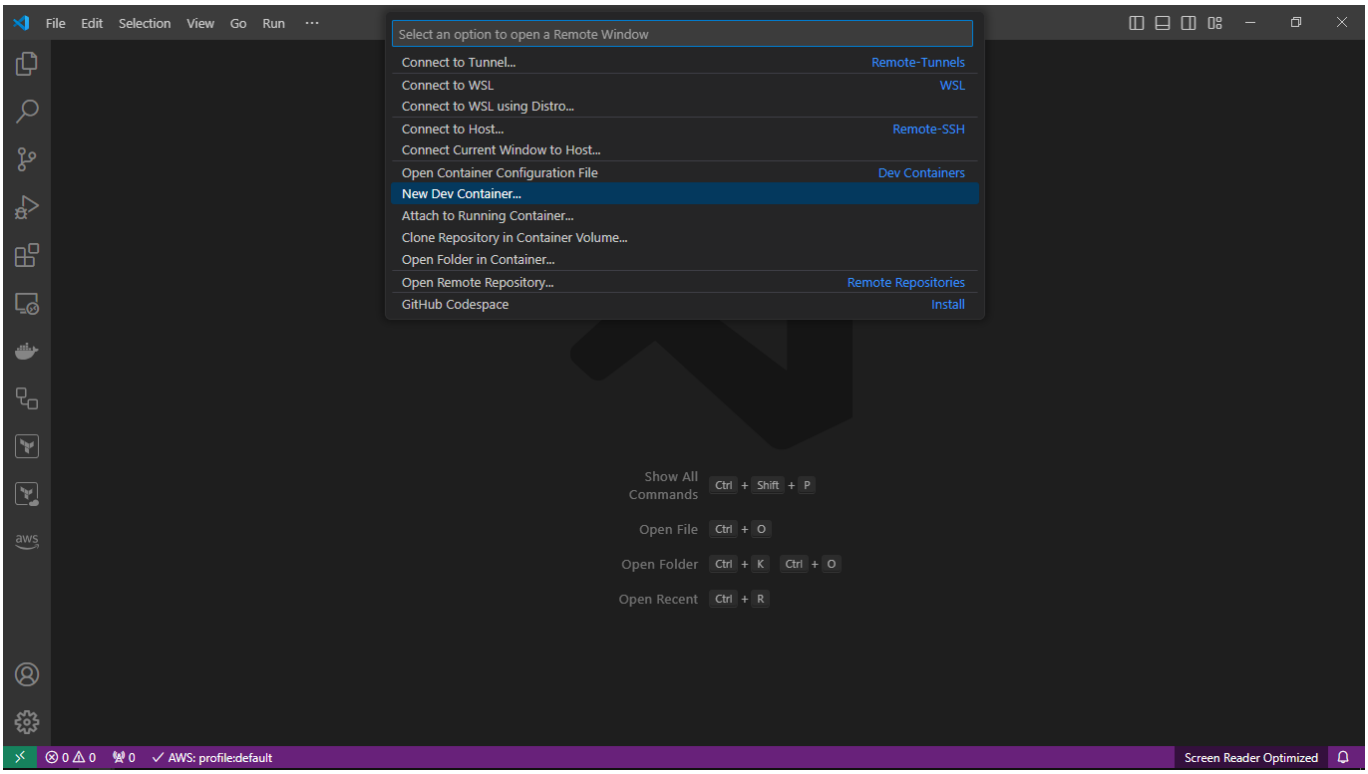
1. Start new *dev-container* and start coding.
2. Open existing project directory inside a *dev-container*.
3. Start new *dev-container* with cloning remote Git repository inside it.

Start new *dev-container*

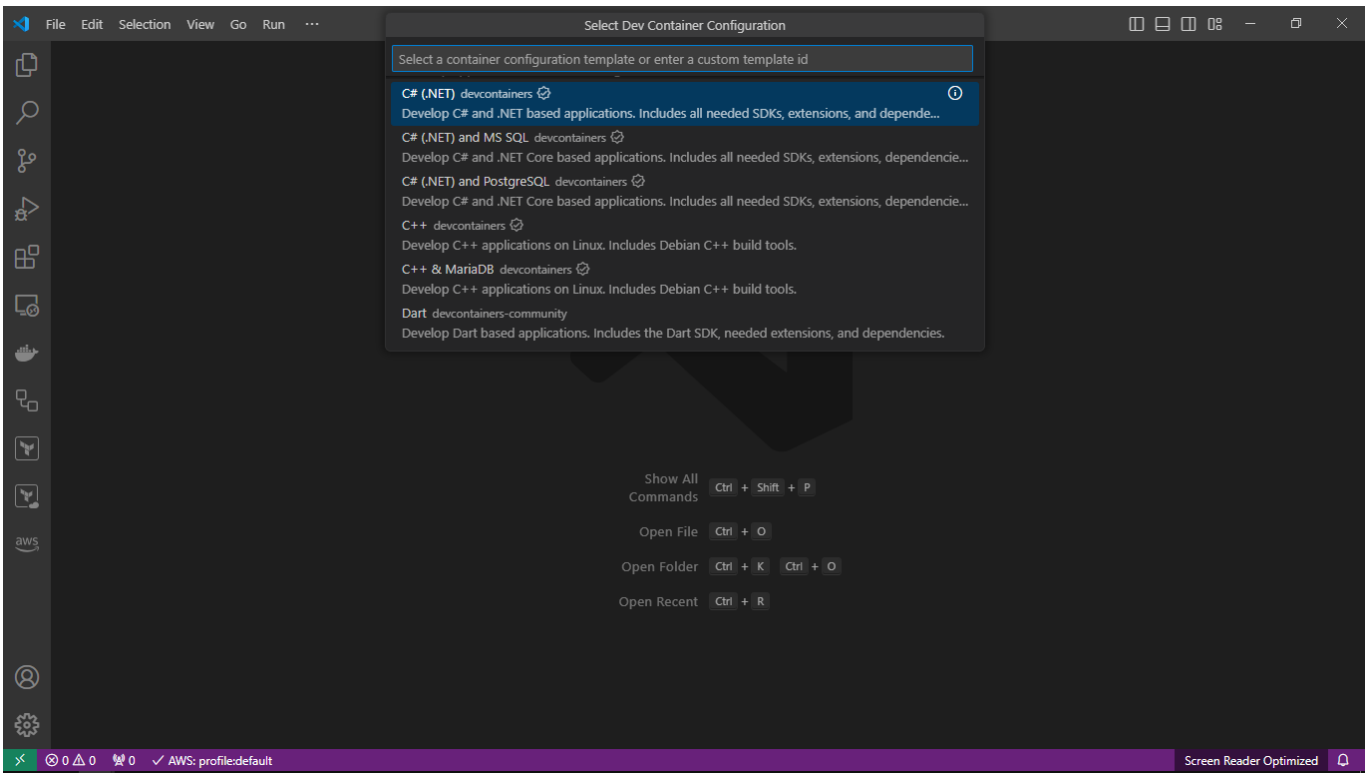
1. Click on **Remote Host** in VSCode status bar.



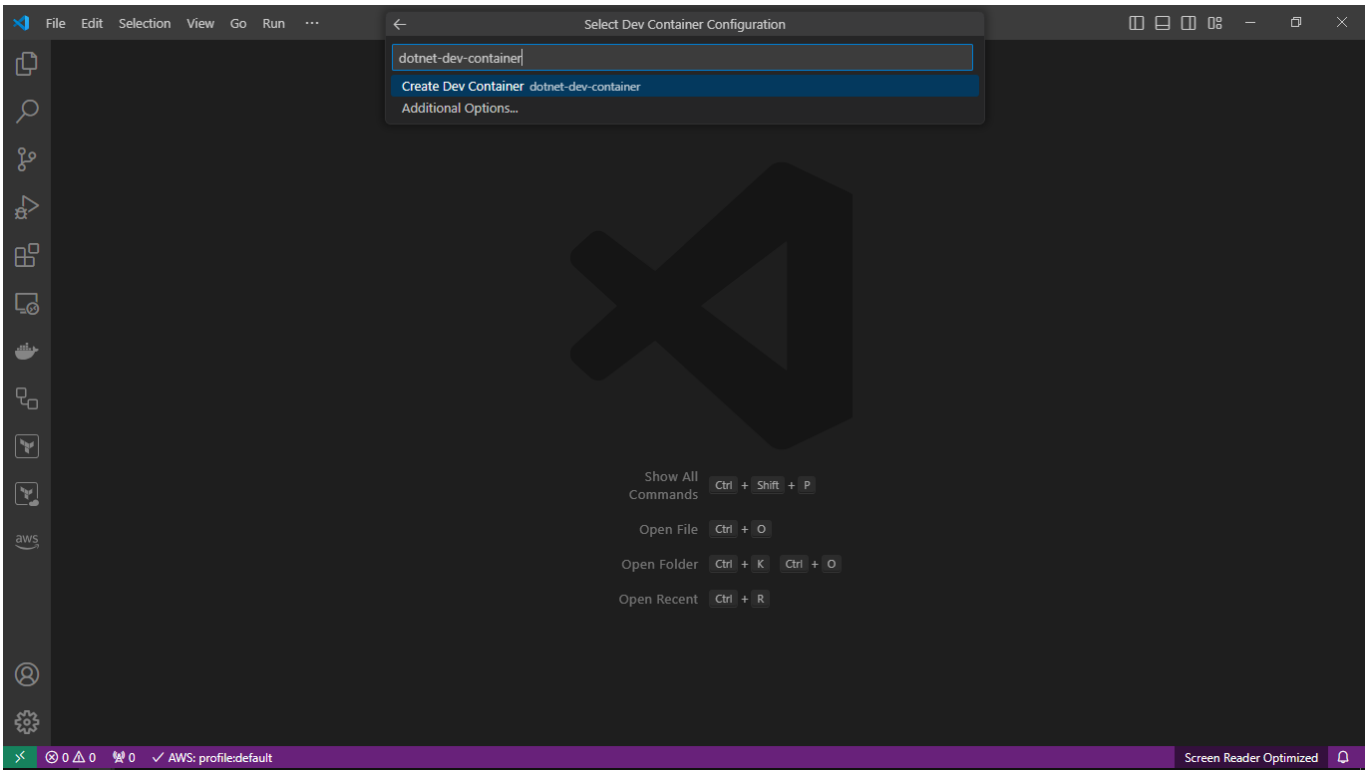
2. The Command Palette will be dropped down. From that, select `New Dev Container`.



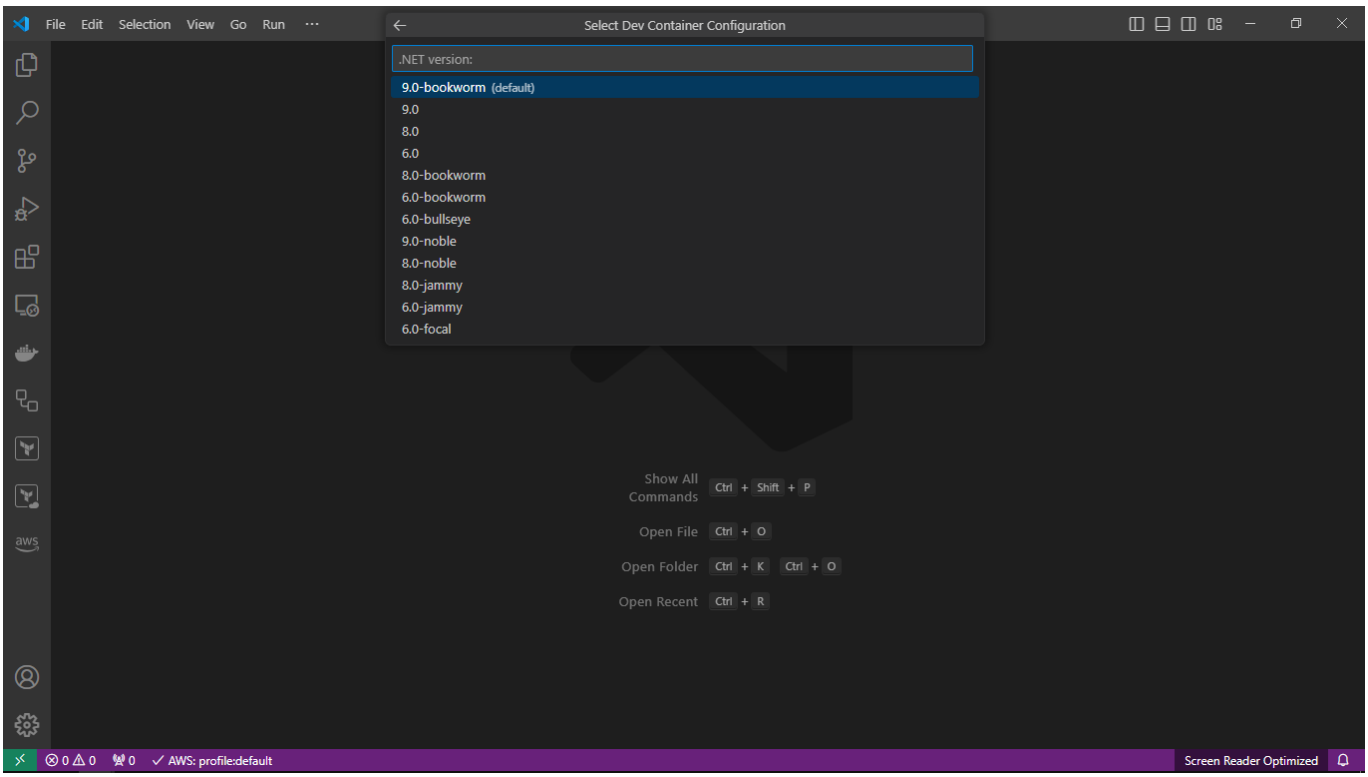
3. Now select the container configuration template from the drop-down. For documentation purposes, we will select the `C# (.NET)` template.



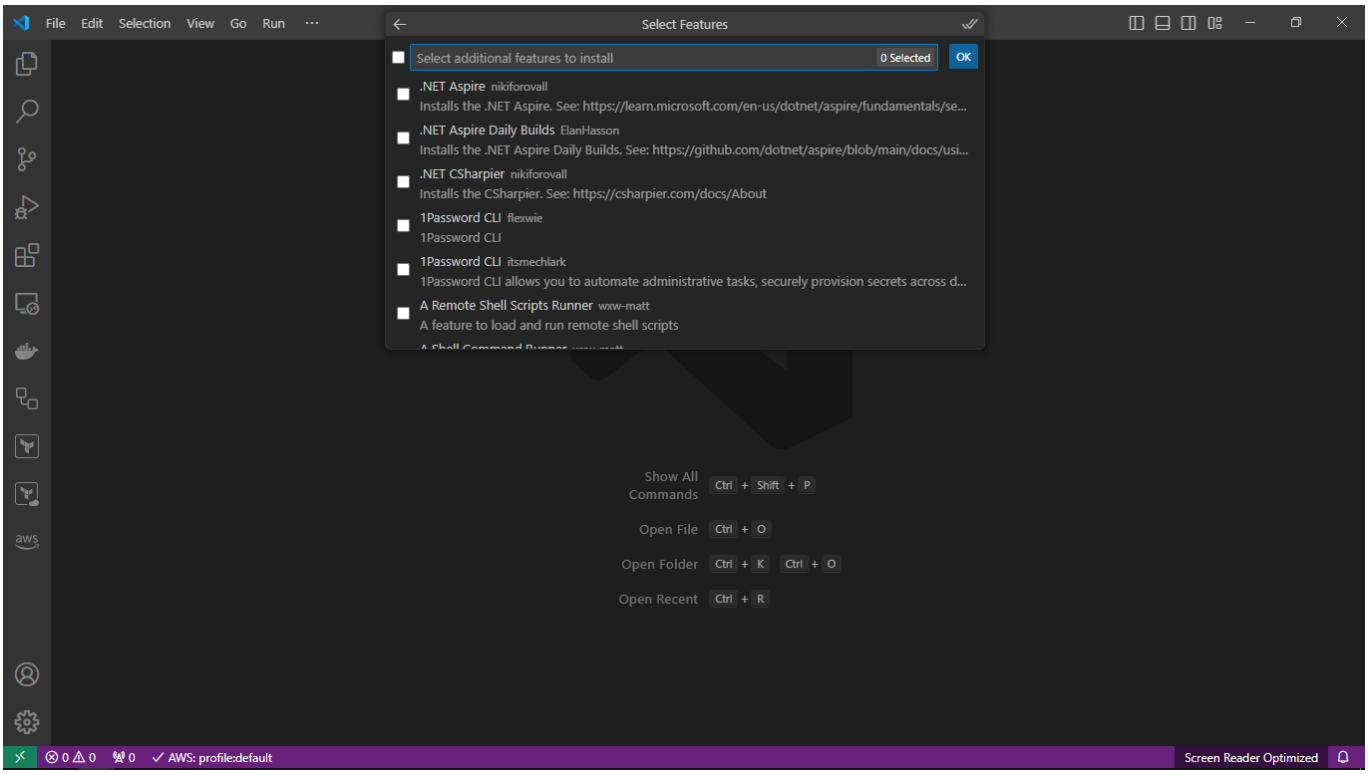
4. Now, enter the custom dev container name, & we can either **Create Dev Container** or configure **Additional Options**.



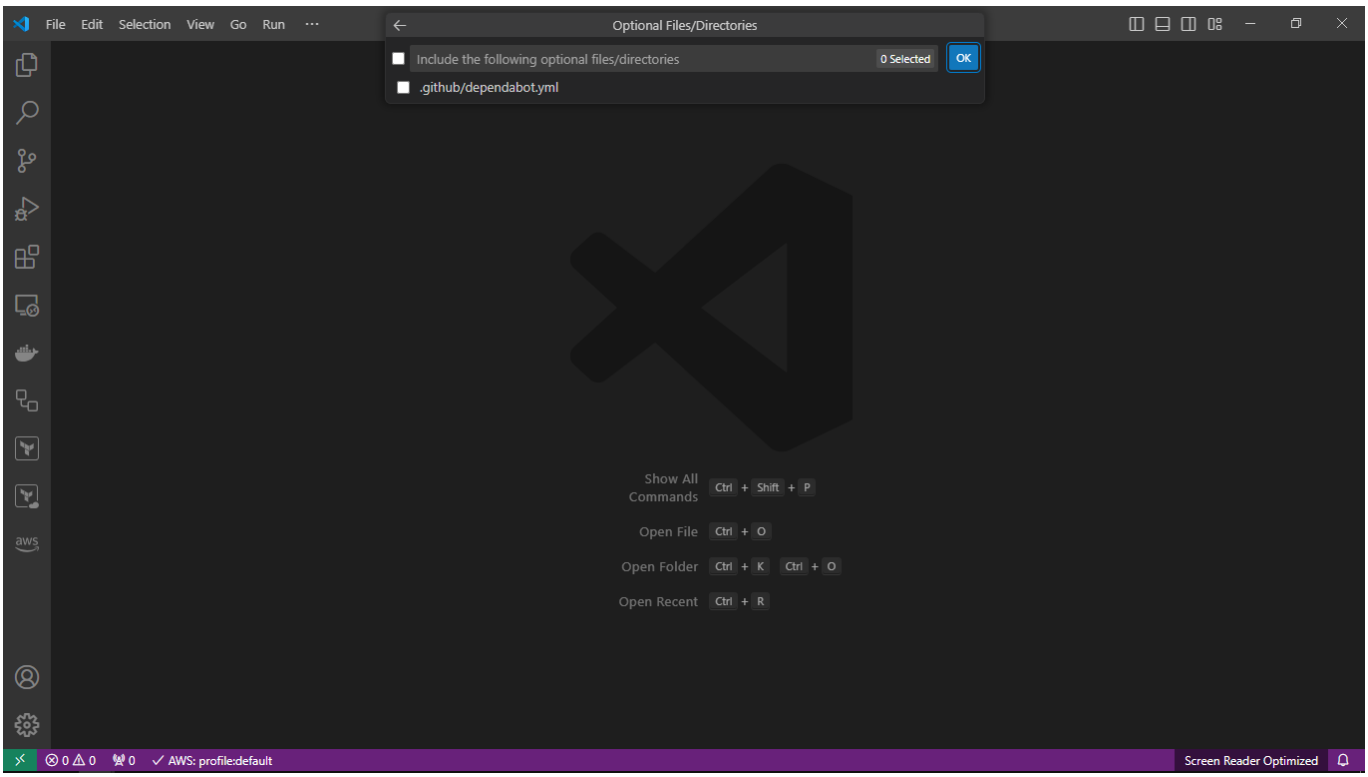
5. Inside **Additional Options**, we can configure the container configuration template version. Select the version according to your requirements.



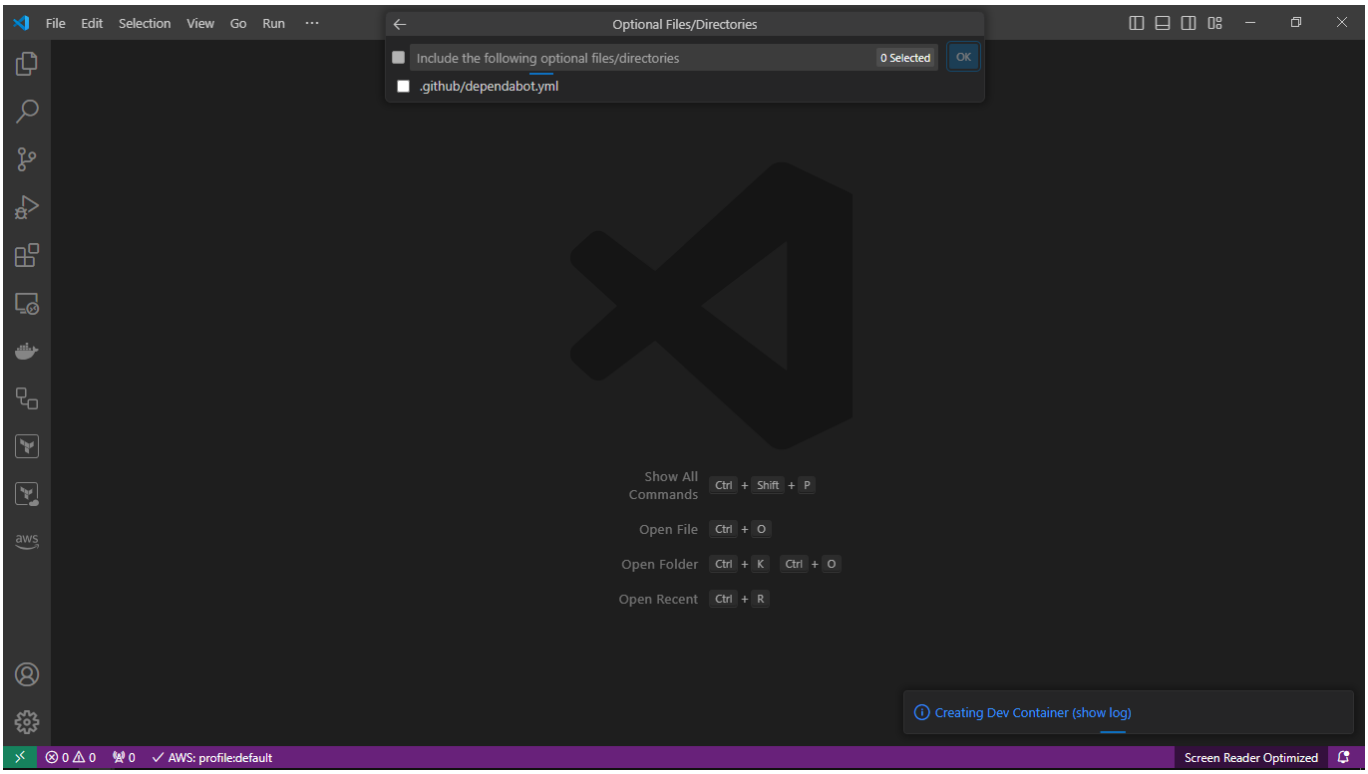
6. We can also select the additional features to install in our *devcontainer*. For now, we will click `OK` as we do not need any additional features.



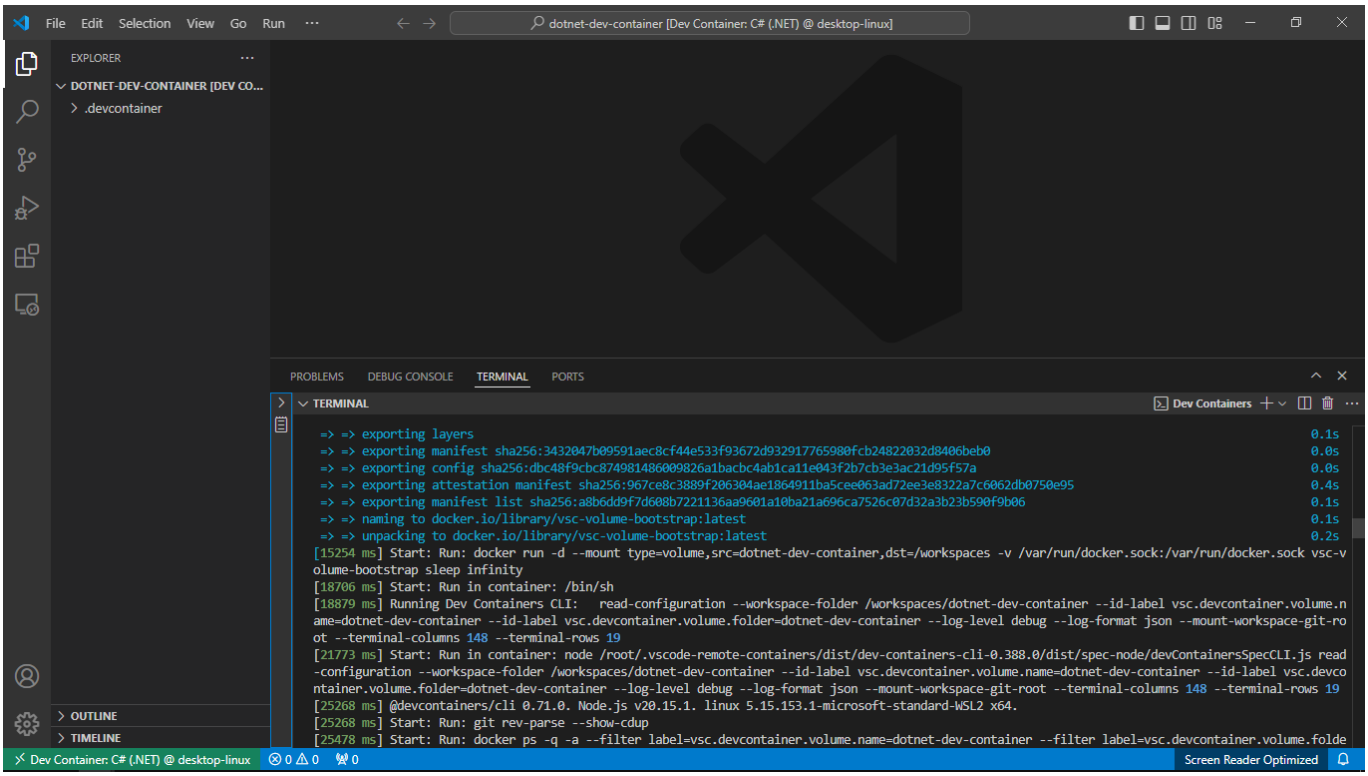
7. We can include the *.github/dependabot.yml* file for checking package updates. For now, we will click **OK** as we do not need it.



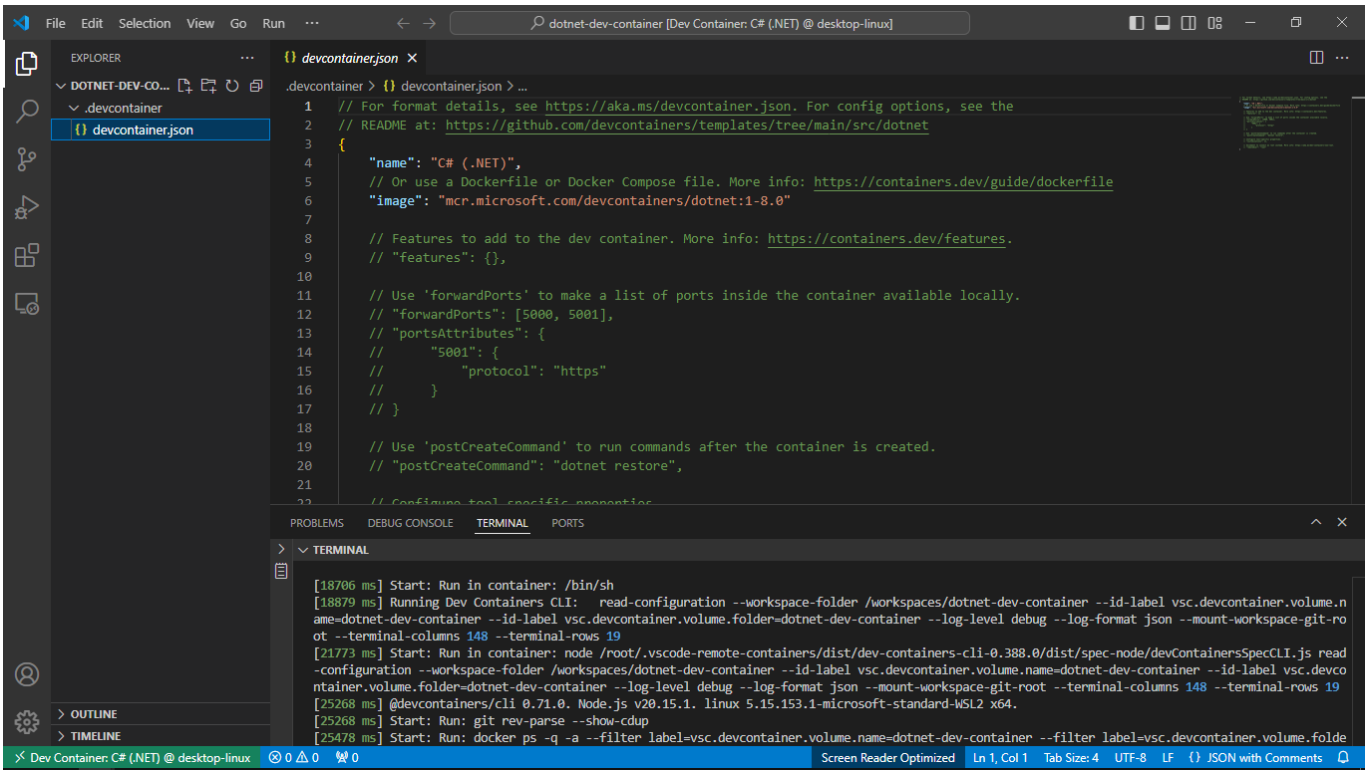
8. The creation of *dev-container* will be started. Click on show logs.



9. Terminal will be opened and the logs of dev container creation will be displayed.



10. The .devcontainer/devcontainer.json file will be created with the devcontainer configuration.

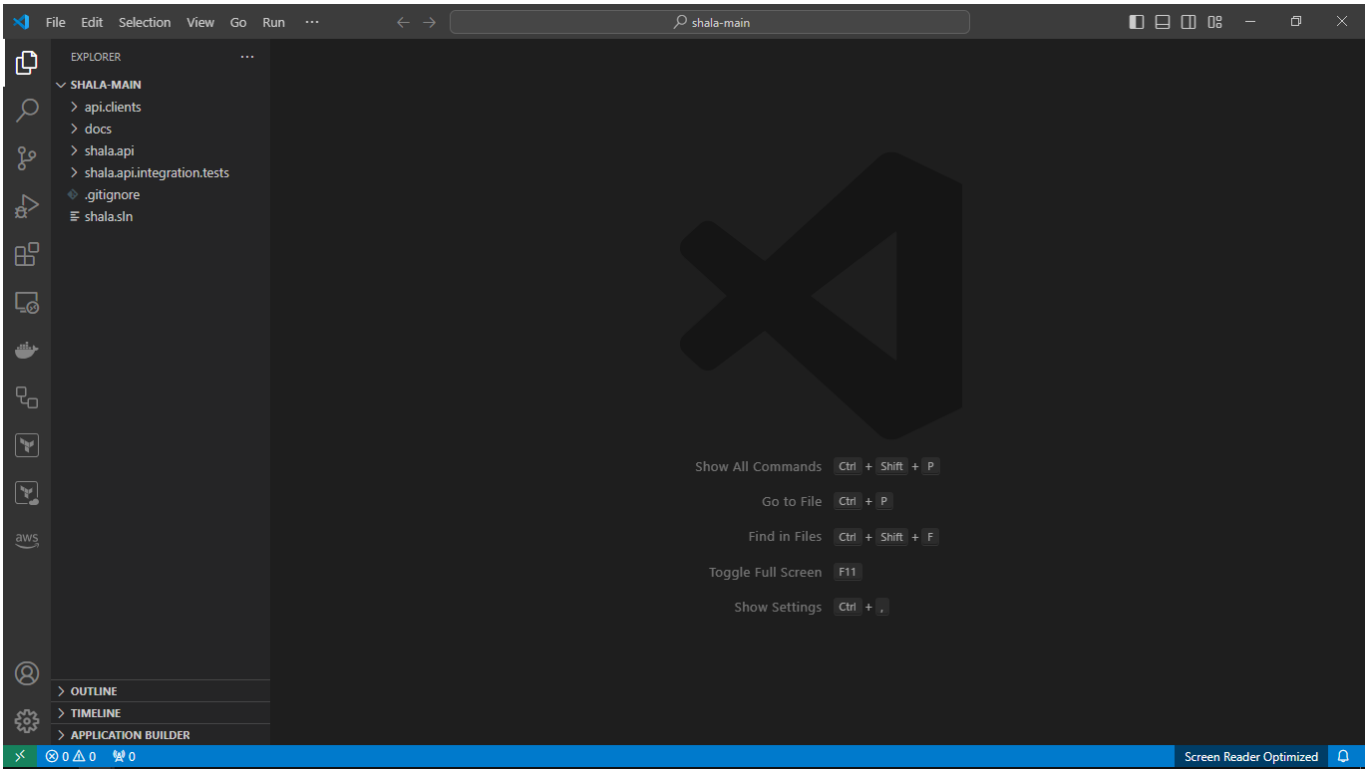


11. The dev container setup is completed and now you can initialize your project.

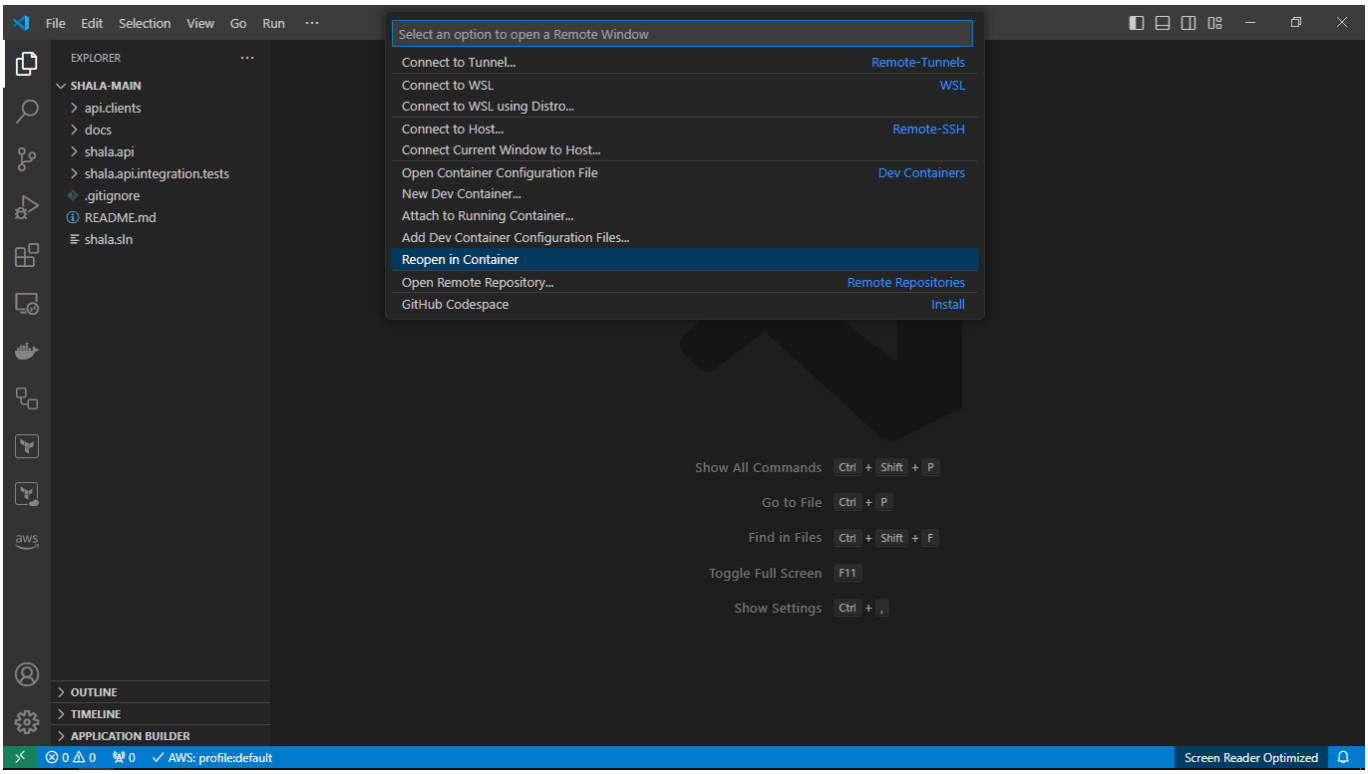
> Git is also installed with the template, so you can initialize & configure Git to clone the existing repository or push to the new repository.

Open the Existing Project Directory inside a Dev Container

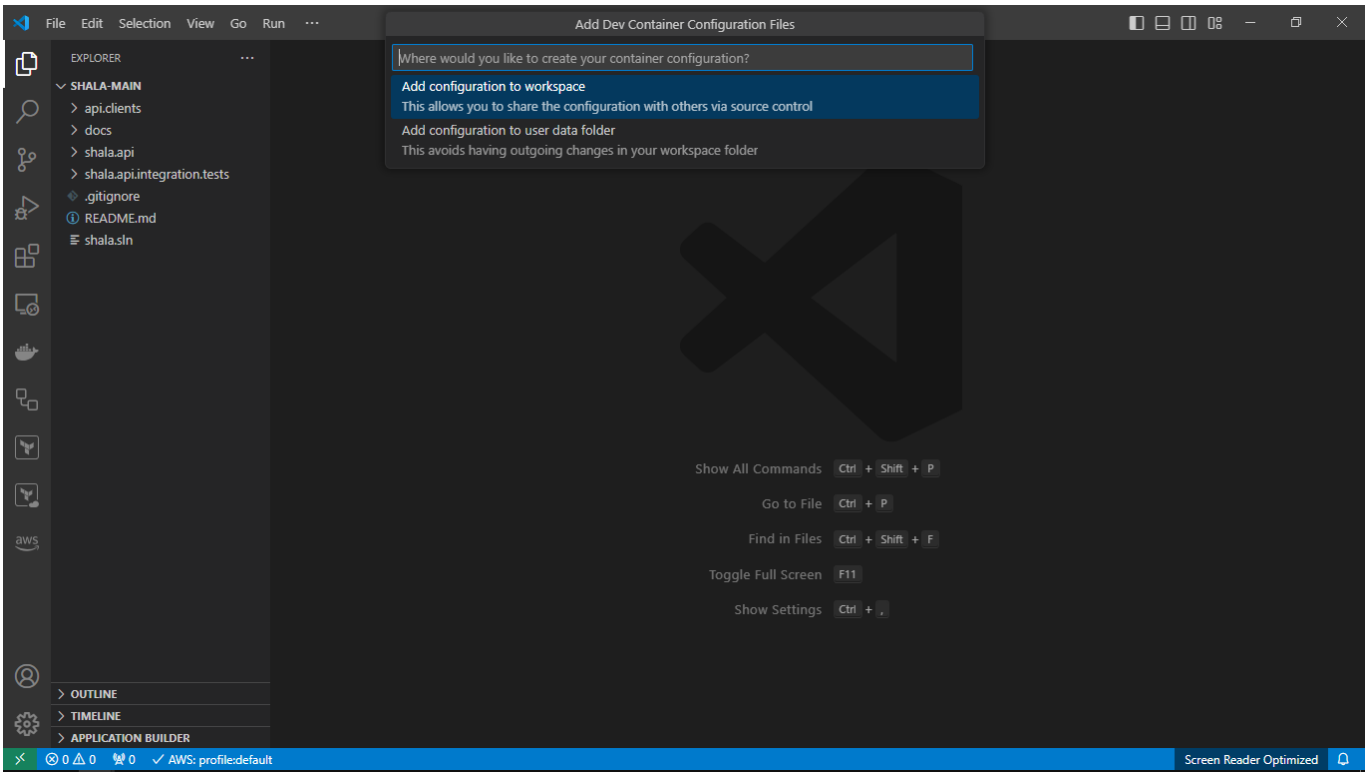
1. Open the project directory containing project files inside VSCode.
2. Click on **Remote Host** in VSCode status bar.



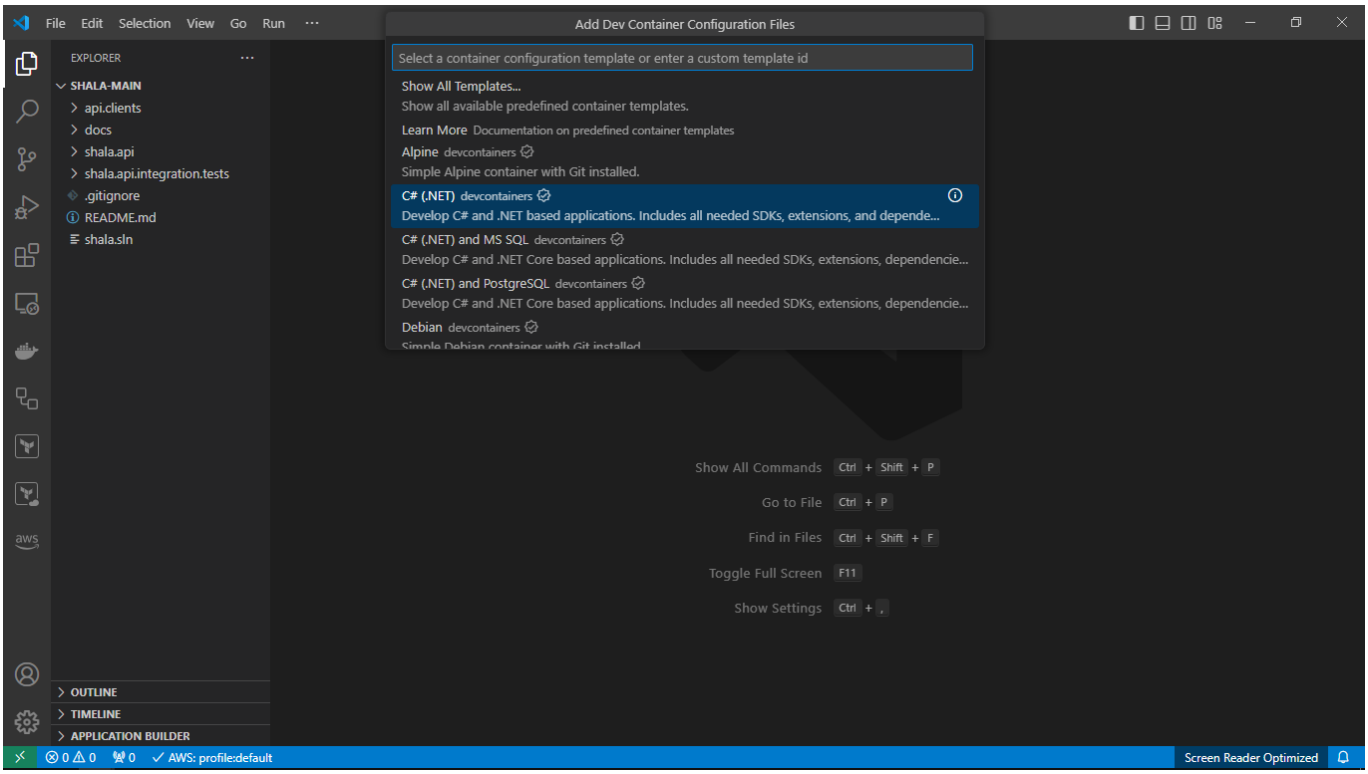
3. The Command Palette will be dropped down. From that, select `Reopen in Container`.



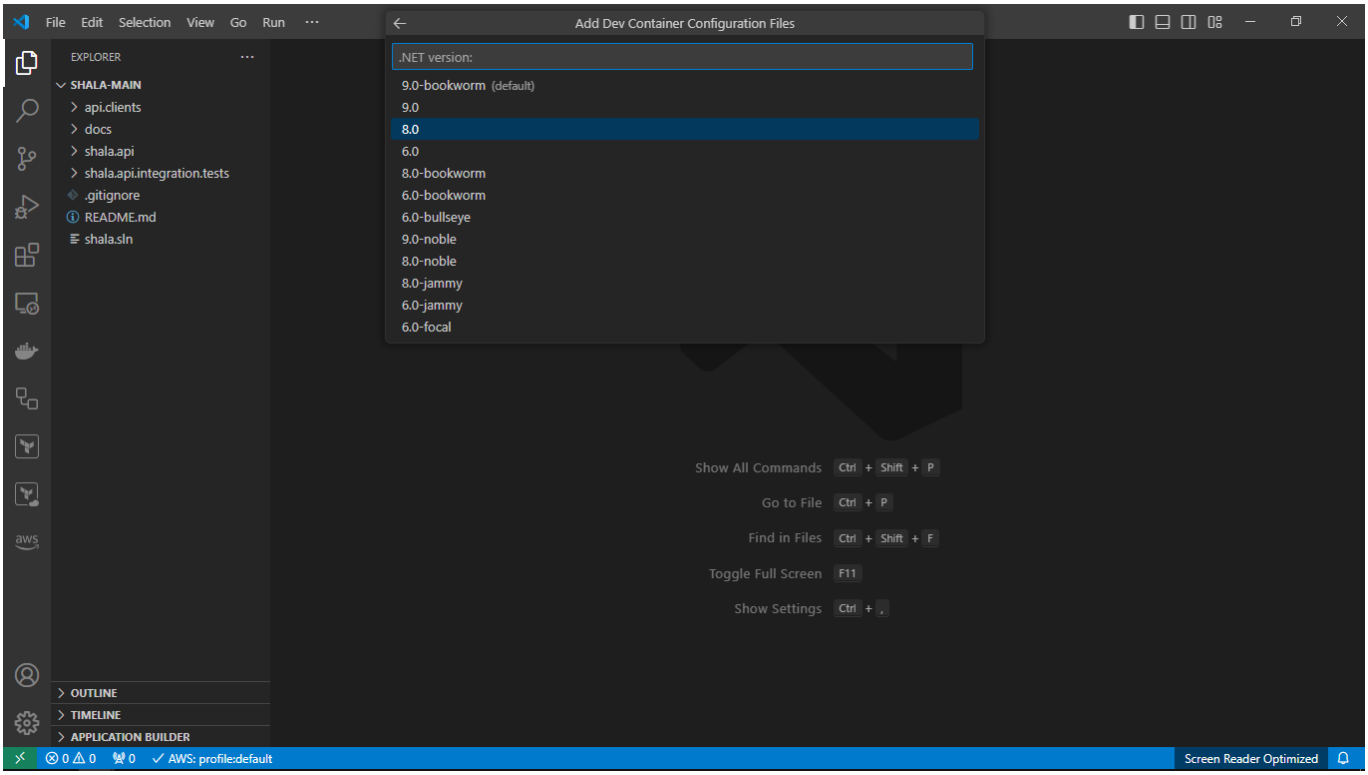
4. Select `Add configuration to workspace`.



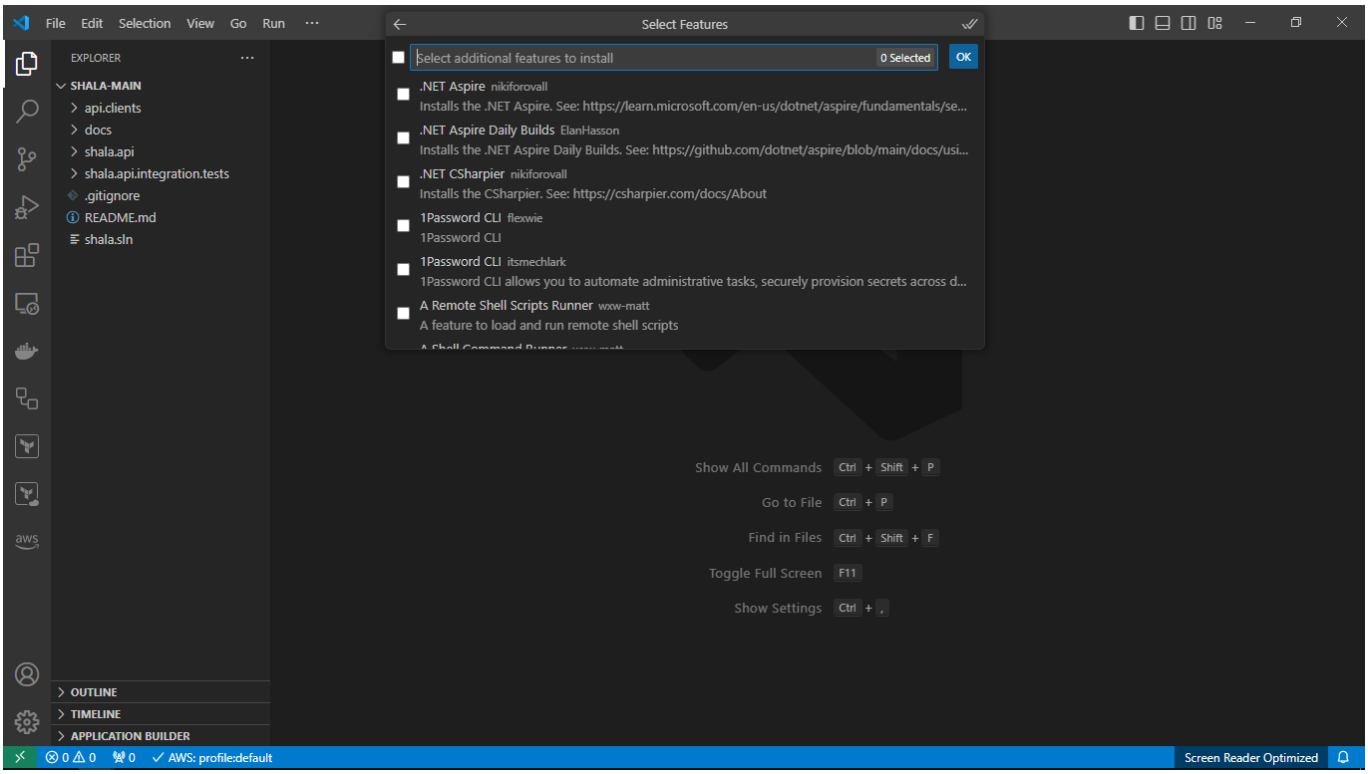
5. Now select the container configuration template from the drop-down. For documentation purposes, we will select the `C# (.NET)` template.



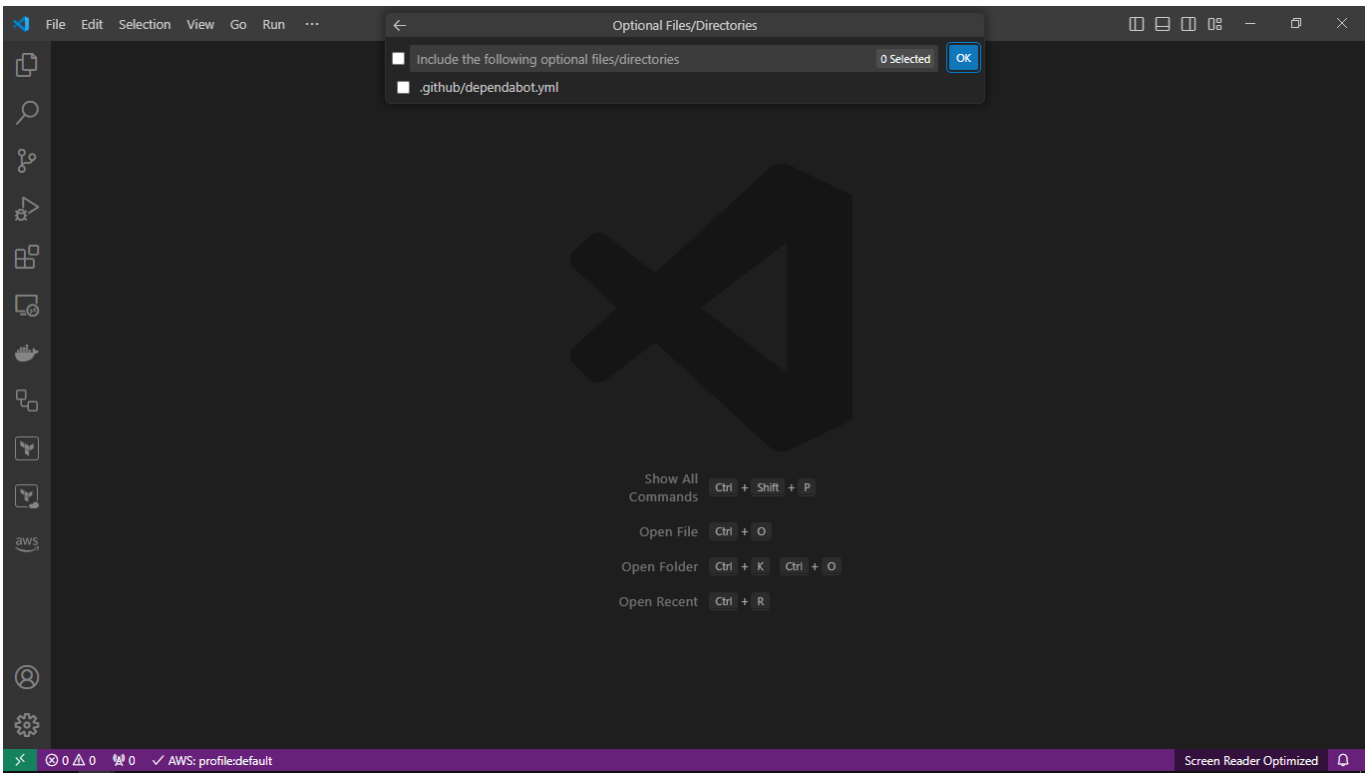
6. Select the container configuration template version from the drop-down according to your requirements.



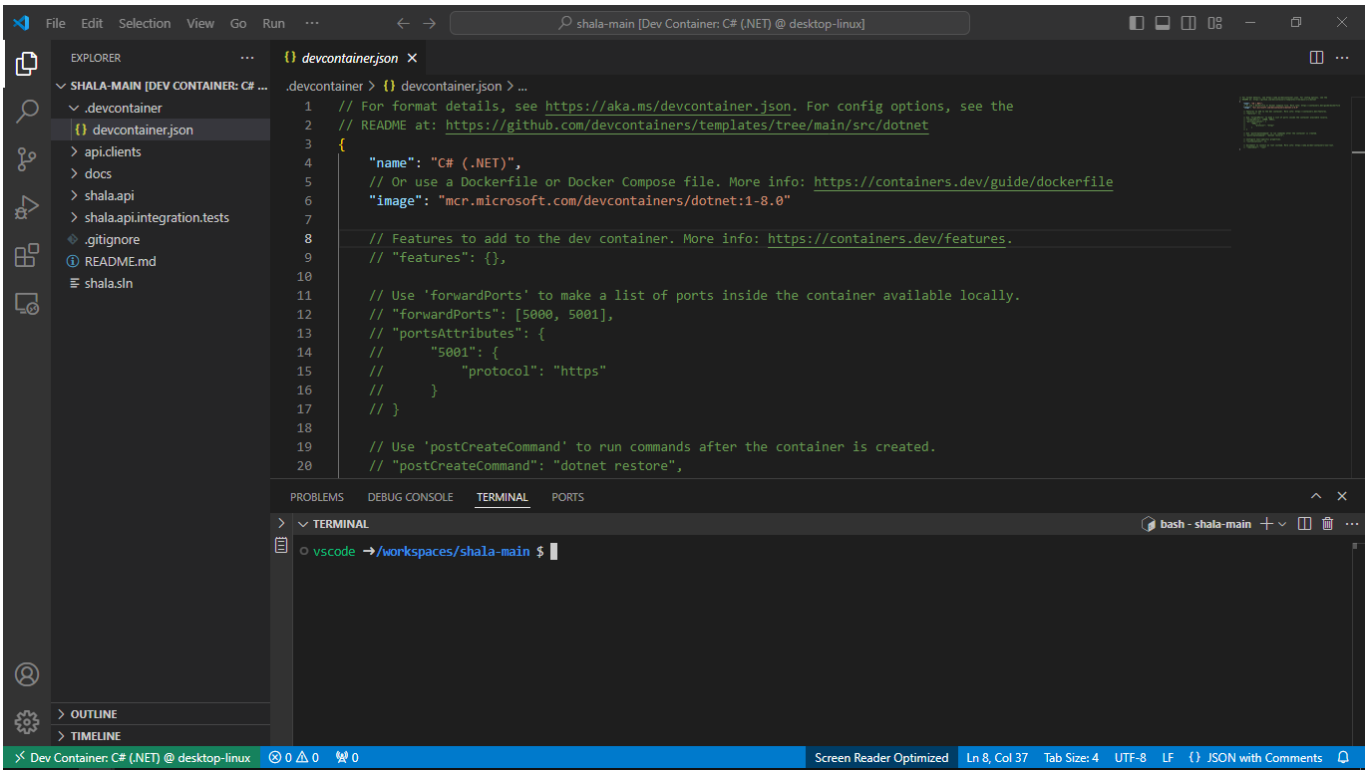
7. We can also select the additional features to install in our *devcontainer*. For now, we will click OK as we do not need any additional features.



8. We can include the *.github/dependabot.yml* file for checking package updates. For now, we will click **OK** as we do not need it.



9. The creation of *dev-container* will be started.
10. The .devcontainer/devcontainer.json file will be created with dev container configuration inside your local folder.

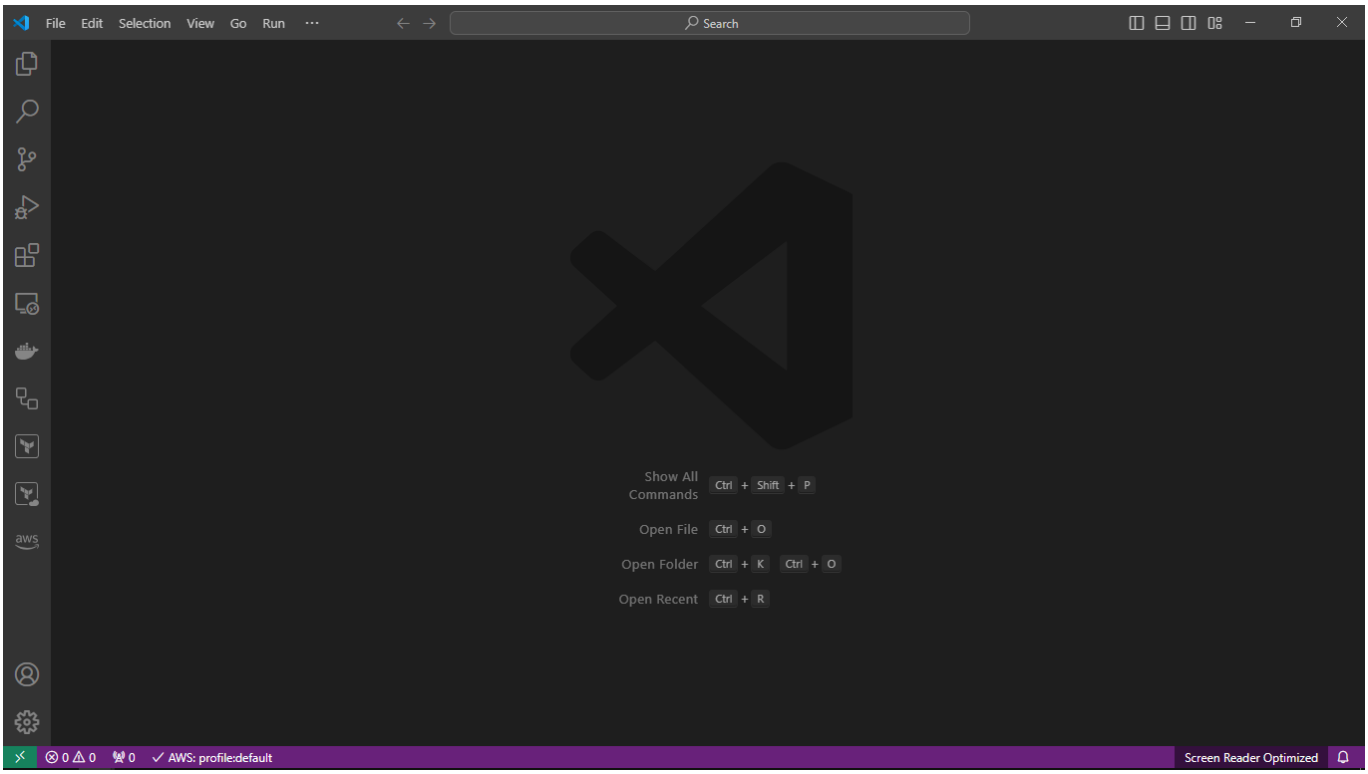


11. The dev container setup is completed and now you can initialize your project.

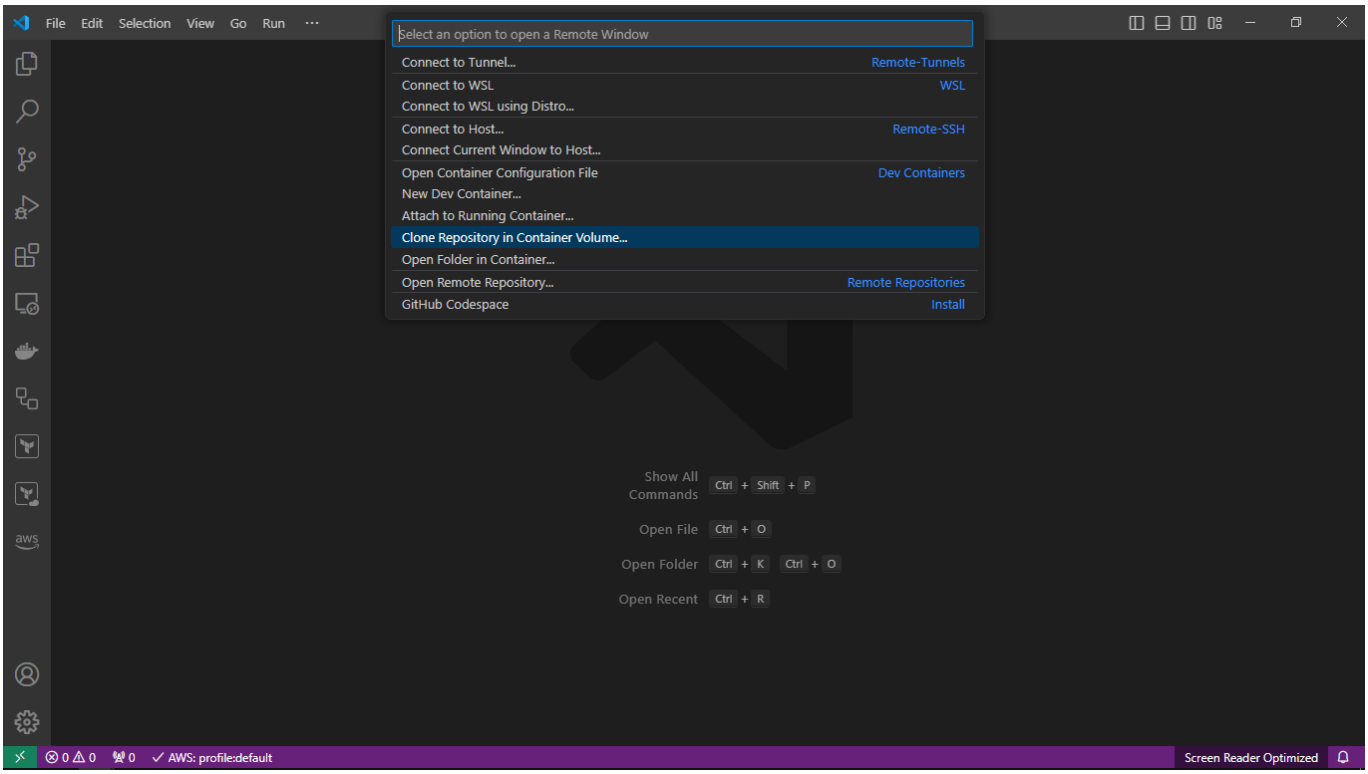
> Git is also installed with the template, so you can initialize & configure git to push to new repository.

Start new *dev-container* with cloning remote Git repository inside it

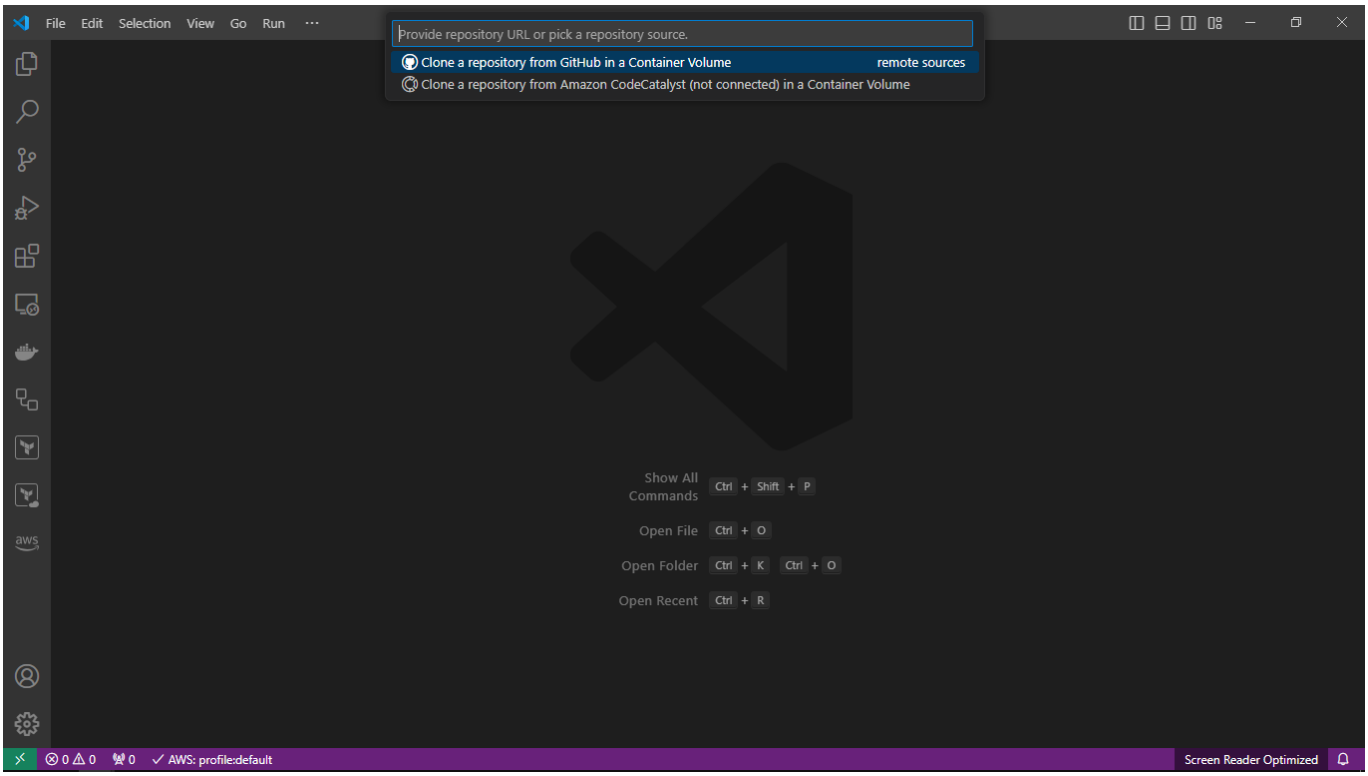
1. Click on **Remote Host** in VSCode status bar.



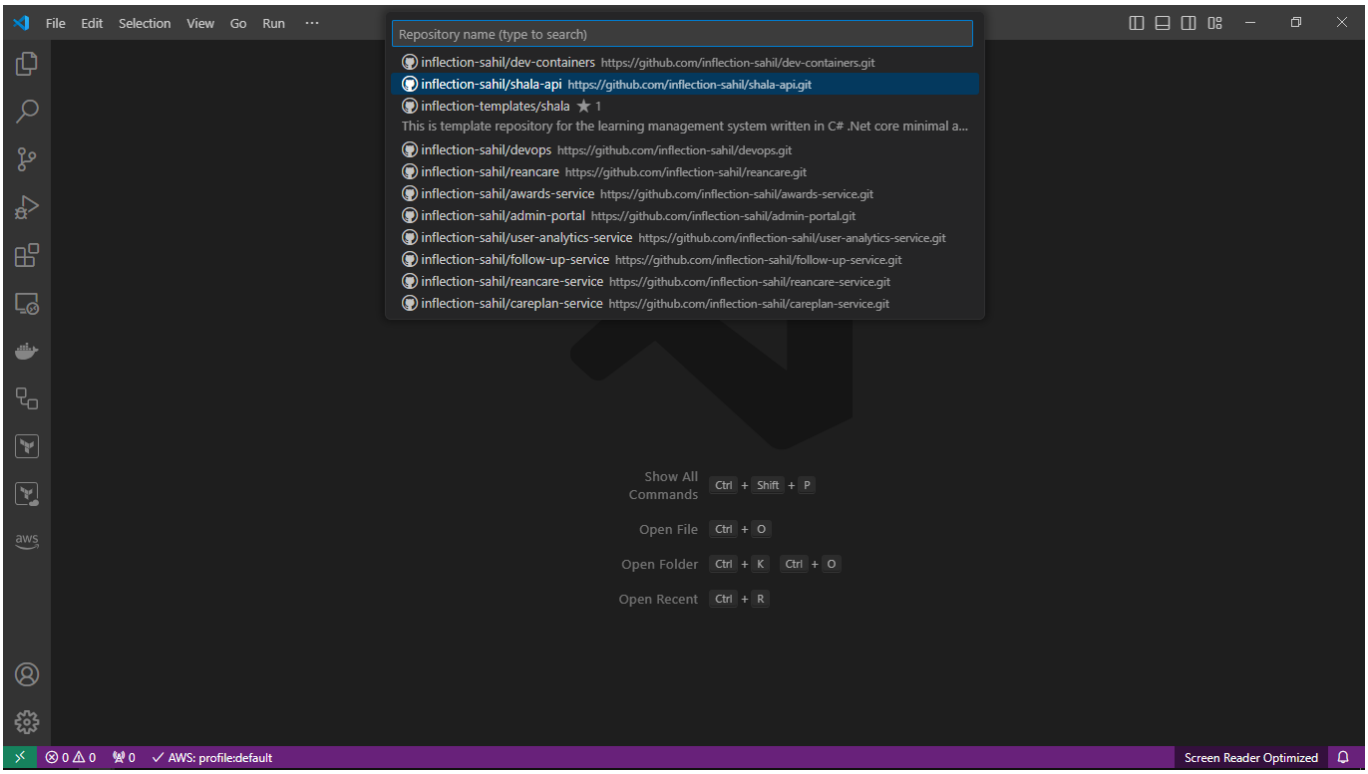
3. Command Palette will be dopped down. From that, select `Clone Repository in a Container Volume`.



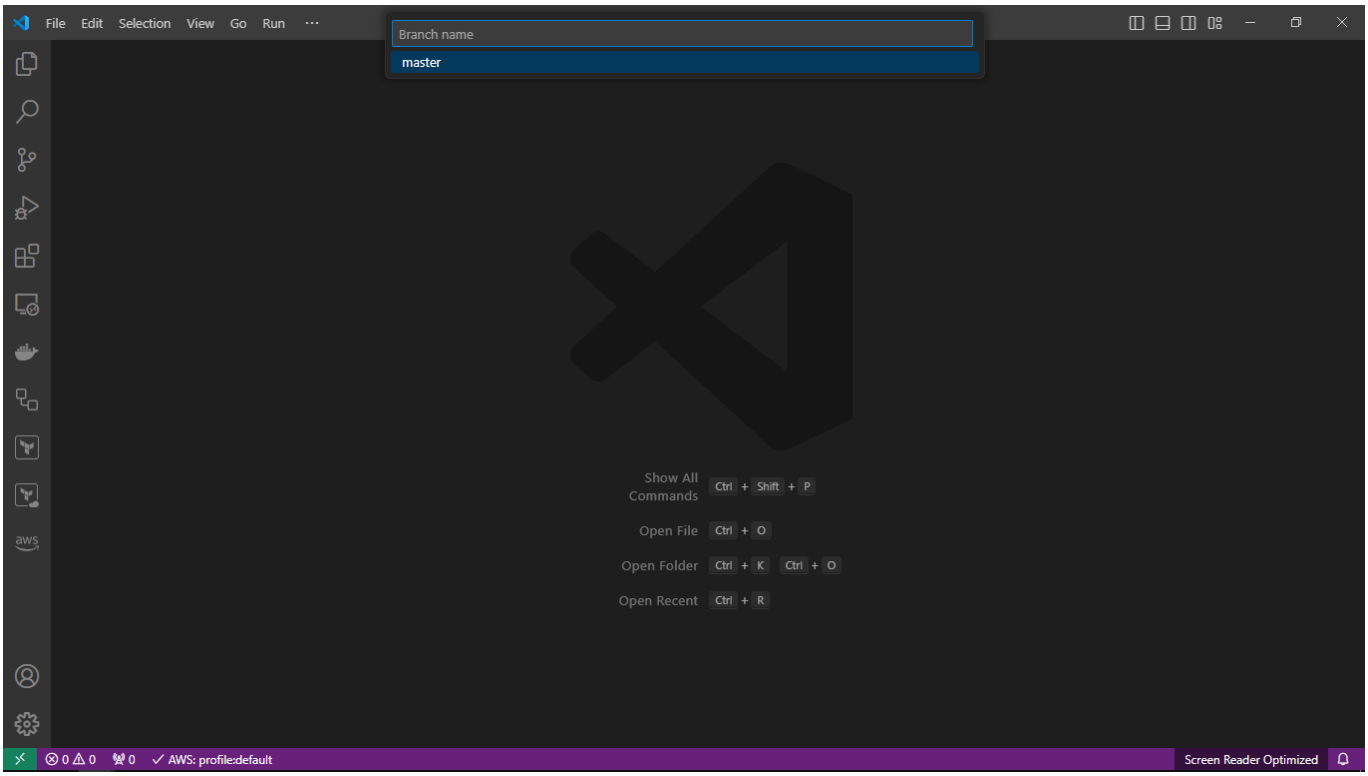
4. Select `Clone a repository from GitHub in Container Volume`.



5. Select the repository you want to clone.

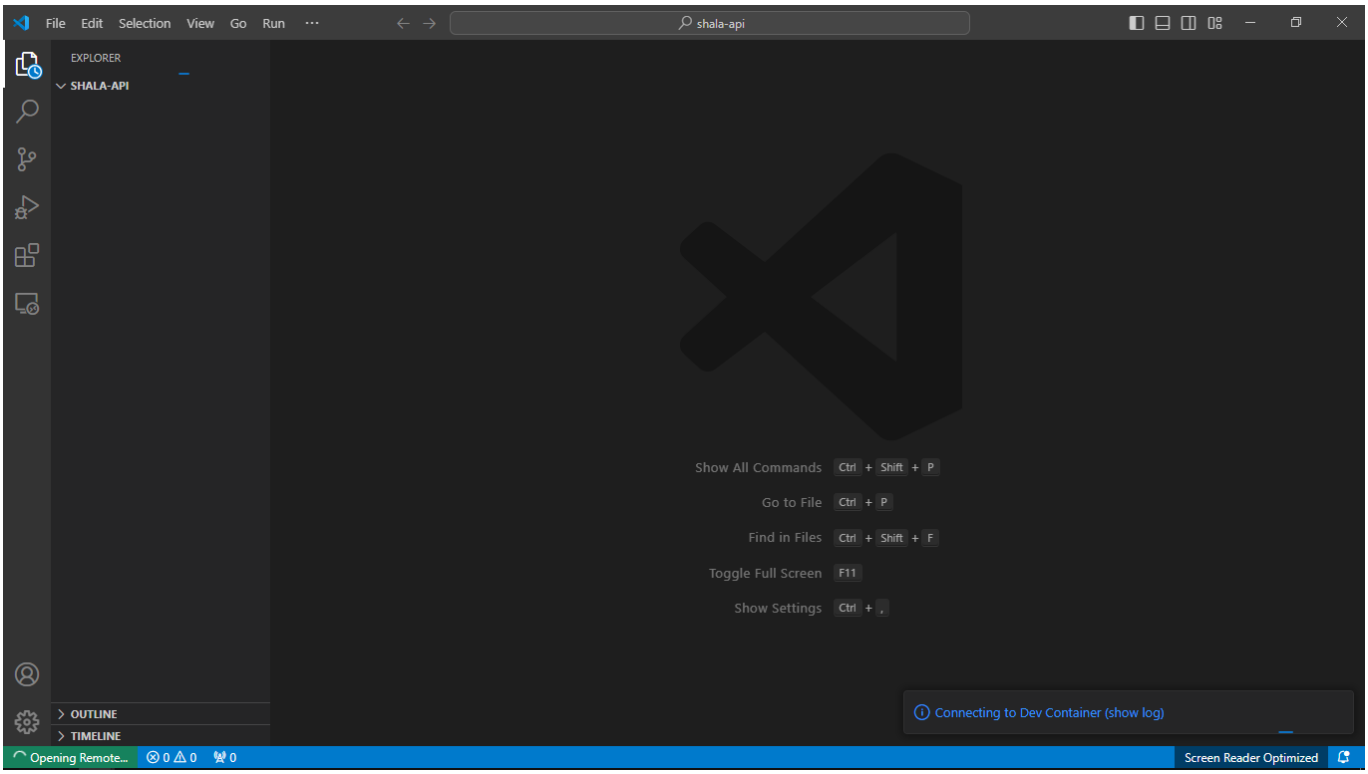


6. Select the branch name you want to clone.

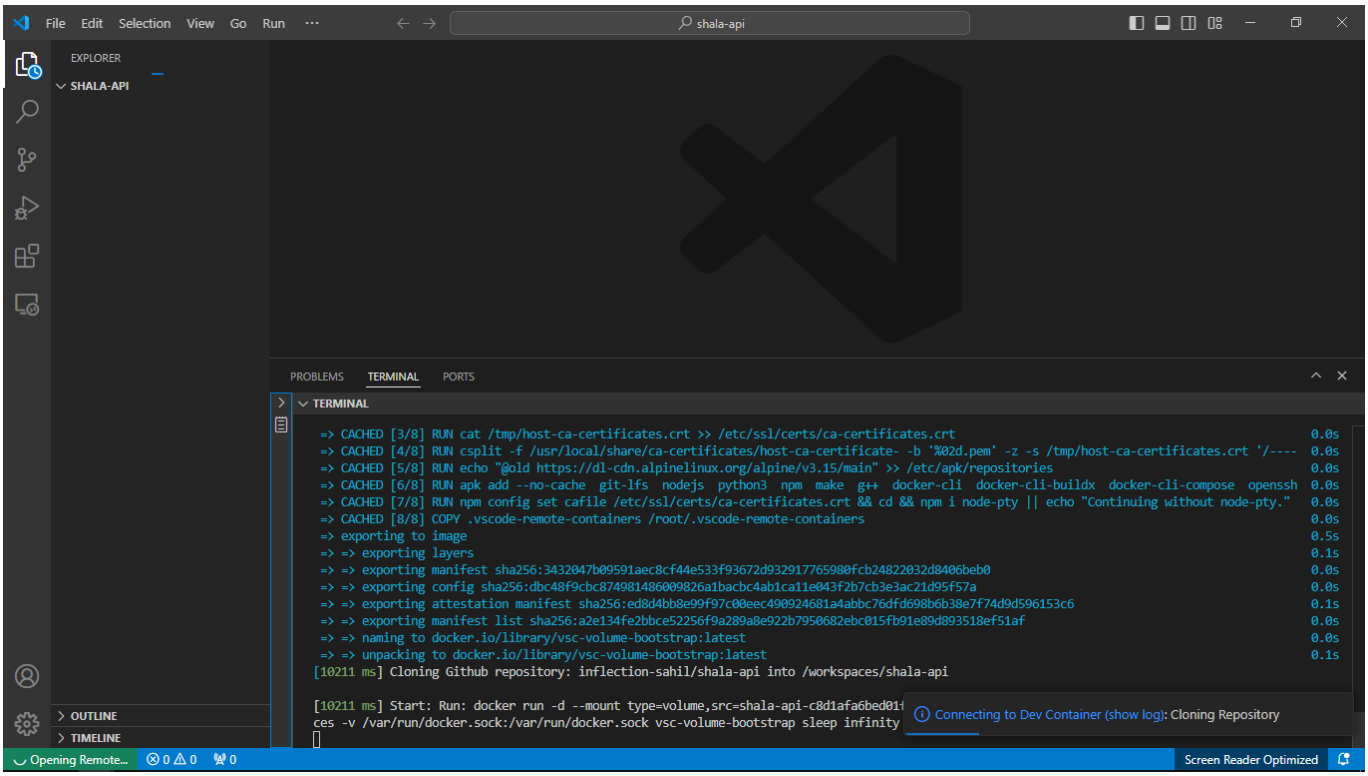


7. It will ask you to log into GitHub.

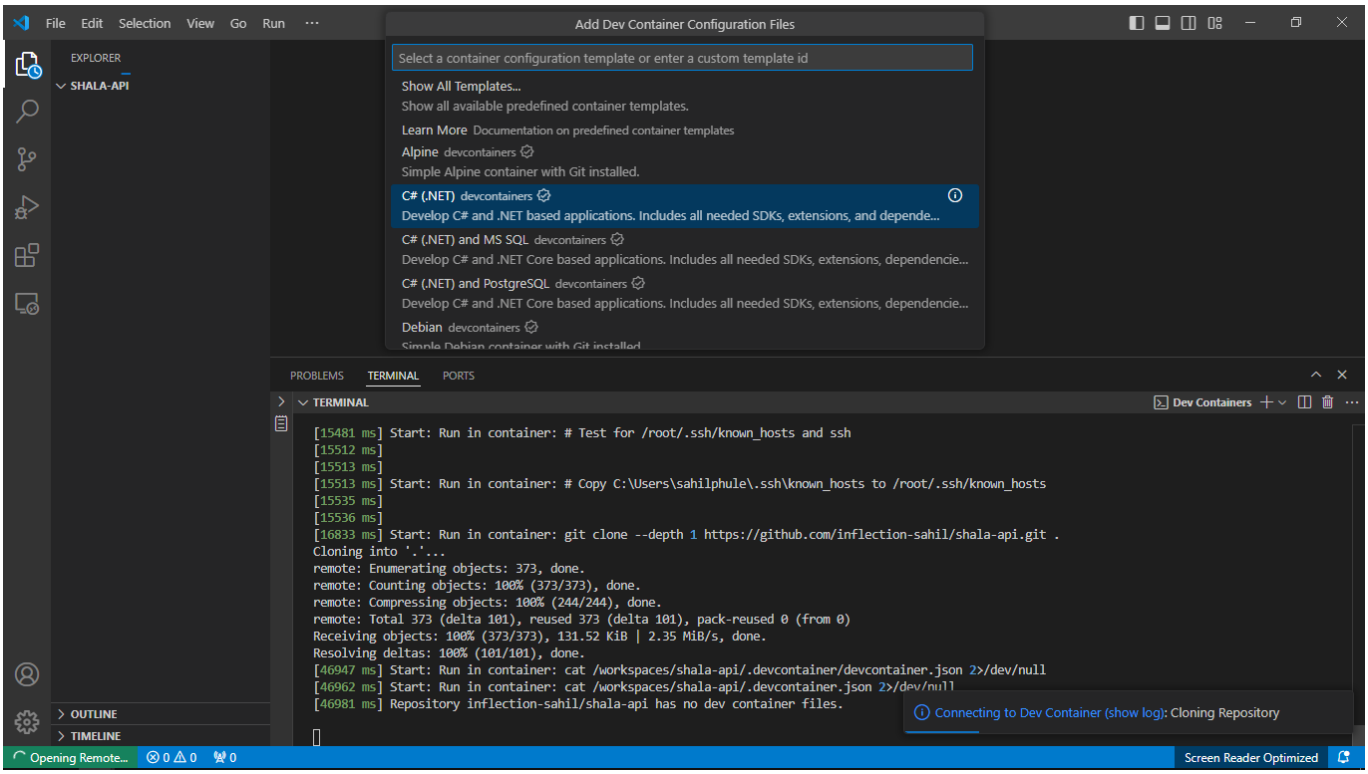
8. Once logged in it will start creating the **Dev Container**. Click on `show logs`.



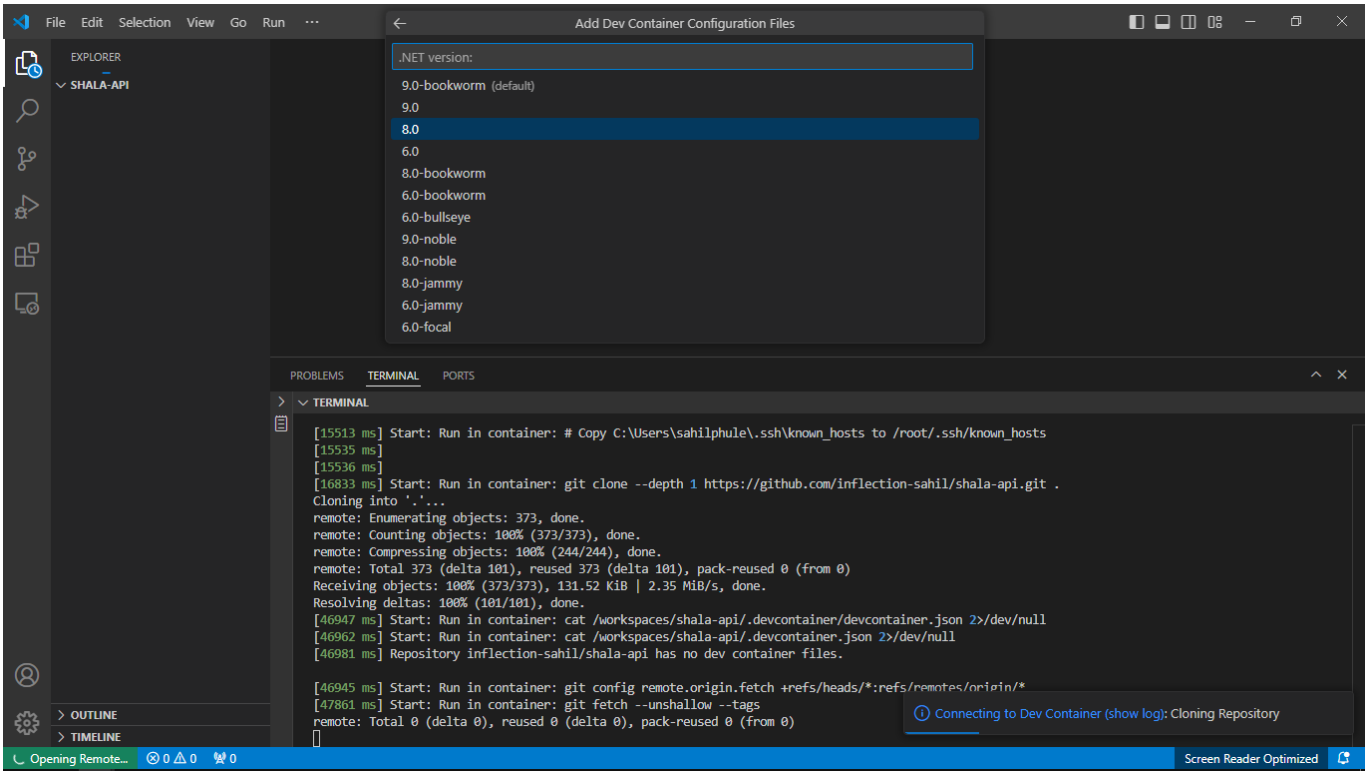
9. The terminal will be opened diplaying the dev container creation logs.



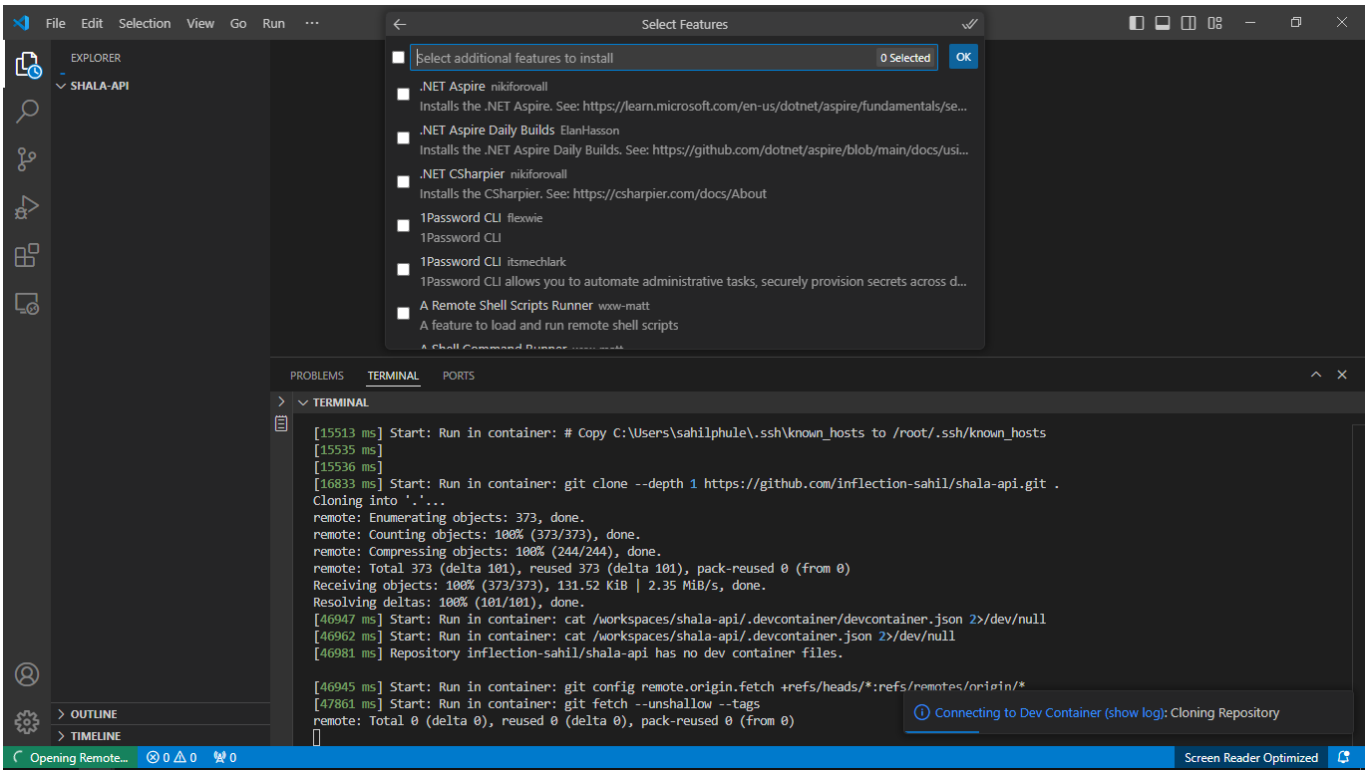
10. Now select the container configuration template from the drop down. For documentation purpose, we will select `C# (.NET)` template.



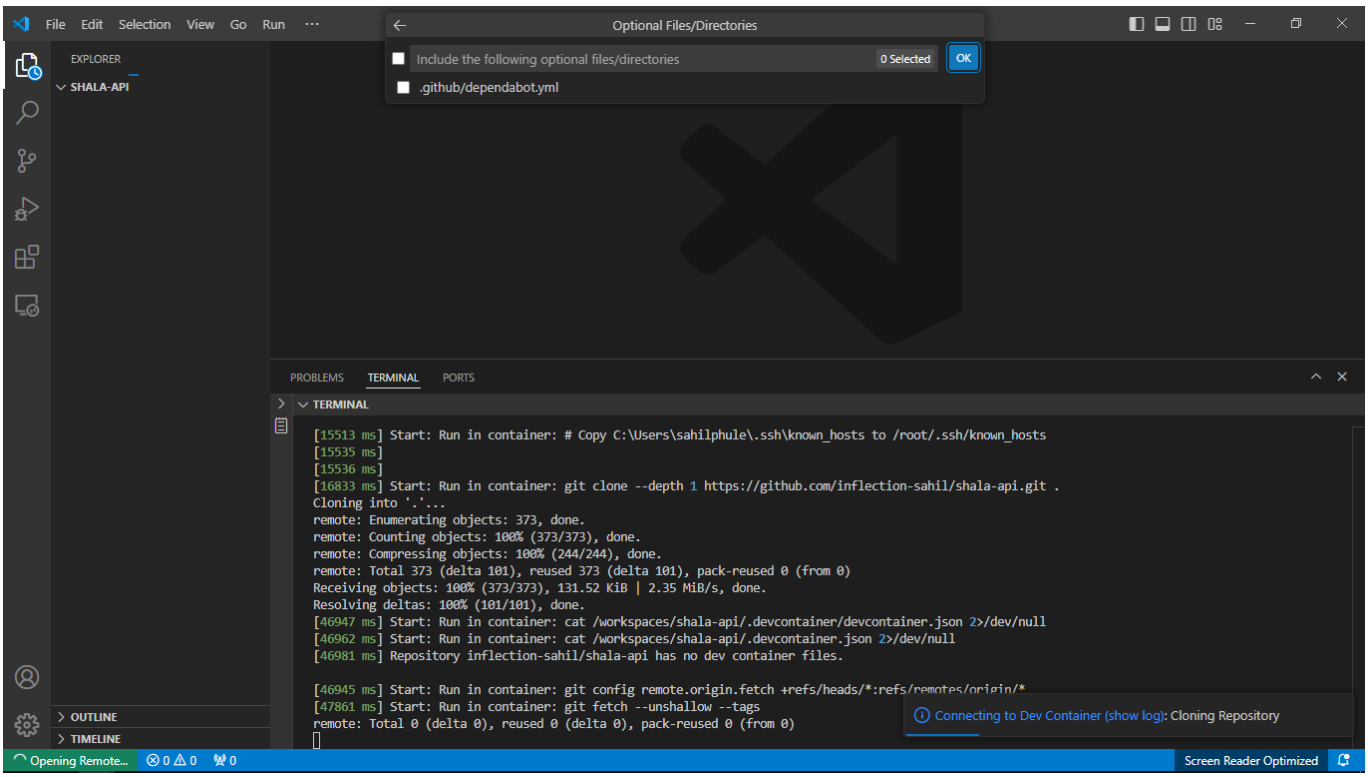
11. Select the container configuration template version from the drop down according to your requirements.



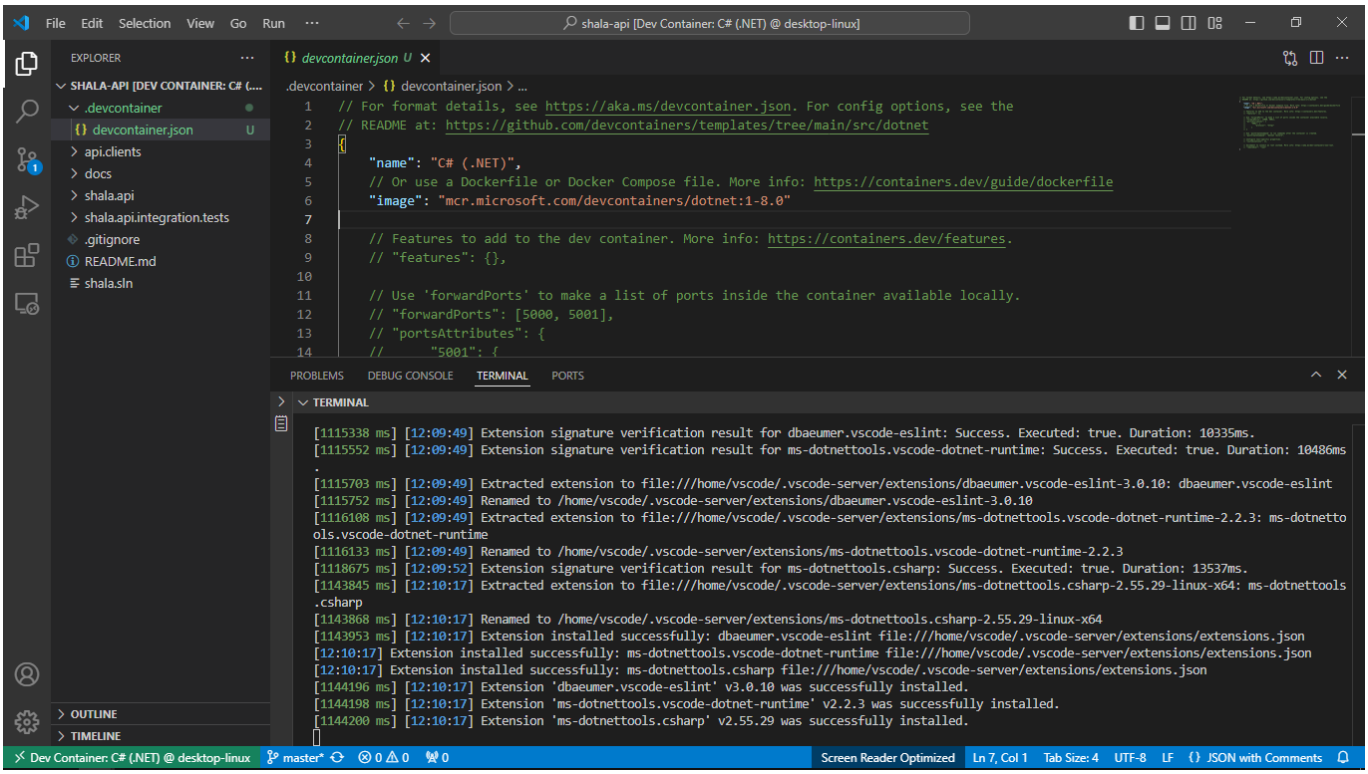
12. We can also select the additional features to install in our *devcontainer*. For now, we will click OK as we do not need any additional features.



13. We can include the *.github/dependabot.yml* file for checking package updates. For now, we will click **OK** as we do not need it.



14. The creation of the dev container with The .devcontainer/devcontainer.json file and cloning of the files has been done.



15. The dev container setup with an existing GitHub repository is completed and now you can edit your project.

> Git is also configured with the repository while cloning, so you can edit the project and push the changes to the repository.
