Code - OSS Development Container

Remote - Containers Open

This repository includes configuration for a development container for working with Code - OSS in a local container or using GitHub Codespaces.

> ***Tip:*** *The default VNC password is* `vscode`*. The VNC server runs on port* `5901` *and a web client is available on port* `6080`*.*

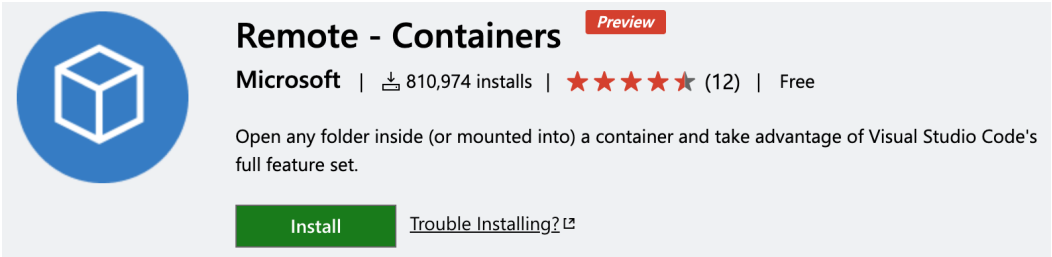
Quick start - local

If you already have VS Code and Docker installed, you can click the badge above or here to get started. Clicking these links will cause VS Code to automatically install the Remote - Containers extension if needed, clone the source code into a container volume, and spin up a dev container for use.

1. Install Docker Desktop or Docker for Linux on your local machine. (See docs for additional details.)
2. **Important**: Docker needs at least **4 Cores and 8 GB of RAM** to run a full build. If you are on macOS, or are using the old Hyper-V engine for Windows, update these values for Docker Desktop by right-clicking on the Docker status bar item and going to **Preferences/Settings > Resources > Advanced**.

   > ***Note:*** *The Resource Monitor extension is included in the container so you can keep an eye on CPU/Memory in the status bar.*

3. Install Visual Studio Code Stable or Insiders and the Remote - Containers extension.



   > ***Note:*** *The Remote - Containers extension requires the Visual Studio Code distribution of Code - OSS. See the FAQ for details.*

4. Press `Ctrl/Cmd` + `Shift` + `P` or `F1` and select **Remote-Containers: Clone Repository in Container Volume...**.

   > ***Tip:*** *While you can use your local source tree instead, operations like* `yarn install` *can be slow on macOS or when using the Hyper-V engine on Windows. We recommend the "clone repository in container" approach instead since it uses "named volume" rather than the local filesystem.*

5. Type `https://github.com/microsoft/vscode` (or a branch or PR URL) in the input box and press `Enter`.
6. After the container is running, open a web browser and go to http://localhost:6080, or use a VNC Viewer to connect to `localhost:5901` and enter `vscode` as the password.

Anything you start in VS Code, or the integrated terminal, will appear here.

Next: **Try it out!**

Quick start - GitHub Codespaces

1. From the microsoft/vscode GitHub repository, click on the **Code** dropdown, select **Open with Codespaces**, and then click on **New codespace**. If prompted, select the **Standard** machine size (which is also the default).

   > ***Note:*** *You will not see these options within GitHub if you are not in the Codespaces beta.*

2. After the codespace is up and running in your browser, press `Ctrl/Cmd` + `Shift` + `P` or `F1` and select **Ports: Focus on Ports View**.

3. You should see **VNC web client (6080)** under in the list of ports. Select the line and click on the globe icon to open it in a browser tab.

   > ***Tip:*** *If you do not see the port, `Ctrl/Cmd` + `Shift` + `P` or `F1`, select **Forward a Port** and enter port `6080`.*

4. In the new tab, you should see noVNC. Click **Connect** and enter `vscode` as the password.

Anything you start in VS Code, or the integrated terminal, will appear here.

Next: **Try it out!**

Using VS Code with GitHub Codespaces

You may see improved VNC responsiveness when accessing a codespace from VS Code client since you can use a VNC Viewer. Here's how to do it.

1. Install Visual Studio Code Stable or Insiders and the the GitHub Codespaces extension.

   > ***Note:*** *The GitHub Codespaces extension requires the Visual Studio Code distribution of Code - OSS.*

2. After the VS Code is up and running, press `Ctrl/Cmd` + `Shift` + `P` or `F1`, choose **Codespaces: Create New Codespace**, and use the following settings:

   - `microsoft/vscode` for the repository.
   - Select any branch (e.g. **main**) - you can select a different one later.
   - Choose **Standard** (4-core, 8GB) as the size.

3. After you have connected to the codespace, you can use a VNC Viewer to connect to `localhost:5901` and enter `vscode` as the password.

   > ***Tip:*** *You may also need change your VNC client's **Picture Quality** setting to **High** to get a full color desktop.*

4. Anything you start in VS Code, or the integrated terminal, will appear here.

Next: **Try it out!**

Try it!

This container uses the Fluxbox window manager to keep things lean. **Right-click on the desktop** to see menu options. It works with GNOME and GTK applications, so other tools can be installed if needed.

> ***Note:*** *You can also set the resolution from the command line by typing* `set-resolution`.

To start working with Code - OSS, follow these steps:

1. In your local VS Code client, open a terminal (`Ctrl/Cmd` + `Shift` + `` ` ``) and type the following commands:

   ```
   yarn install
   bash scripts/code.sh
   ```

2. After the build is complete, open a web browser or a VNC Viewer to connect to the desktop environment as described in the quick start and enter `vscode` as the password.

3. You should now see Code - OSS!

Next, let's try debugging.

1. Shut down Code - OSS by clicking the box in the upper right corner of the Code - OSS window through your browser or VNC viewer.

2. Go to your local VS Code client, and use the **Run / Debug** view to launch the **VS Code** configuration. (Typically the default, so you can likely just press `F5`).

   > ***Note:*** *If launching times out, you can increase the value of* `timeout` *in the "VS Code", "Attach Main Process", "Attach Extension Host", and "Attach to Shared Process" configurations in launch.json. However, running* `scripts/code.sh` *first will set up Electron which will usually solve timeout issues.*

3. After a bit, Code - OSS will appear with the debugger attached!

Enjoy!