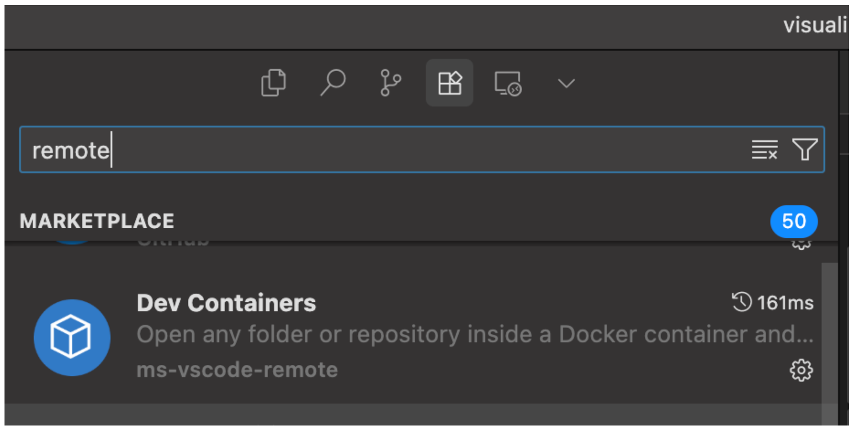
RISC-V Compiler and Simulator Setup Tutorial

Prerequisites

- Computer running Windows 10/11, macOS, or Linux
- Internet connection
- Administrative privileges

Step 1: Install Cursor

1. Download Cursor: https://cursor.com/downloads
2. Install and launch Cursor.
3. Install the following extensions inside Cursor by clicking "extension" icon as shown below. This will open marketplace and just type following:
   - "Remote" and install "Dev Containers" package

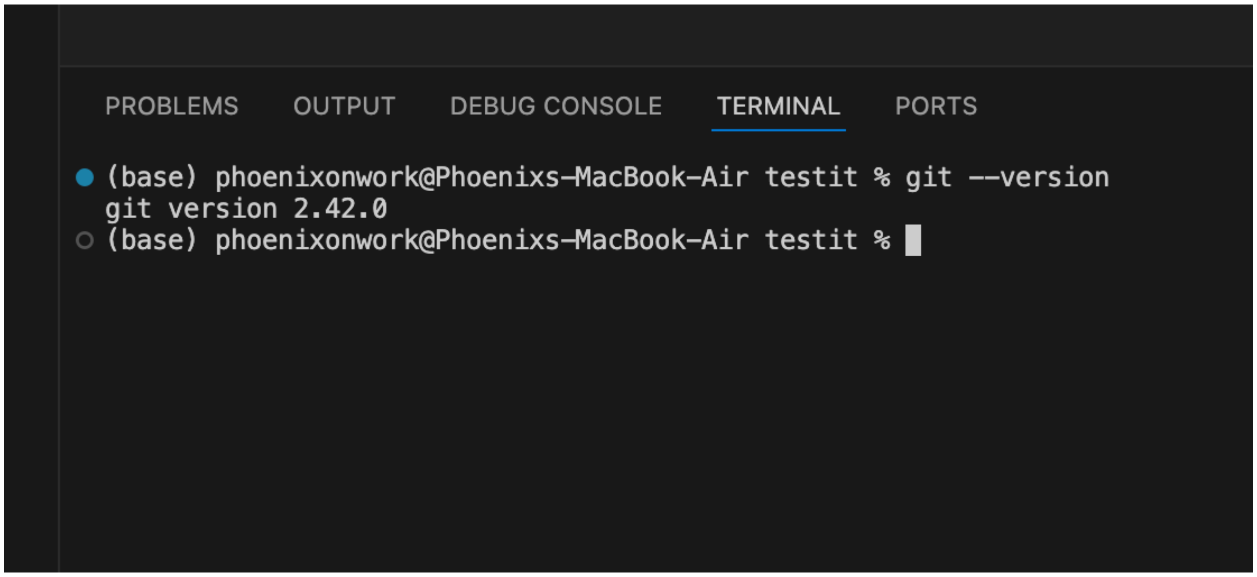


Dev Containers
Open any folder or repository inside a Docker container and...
ms-vscode-remote

Step 2: Install Git and Clone the Repository

1. Download Git: https://git-scm.com/downloads
2. Install Git following the instructions for your OS.
3. Open terminal in Cursor and Verify installation:

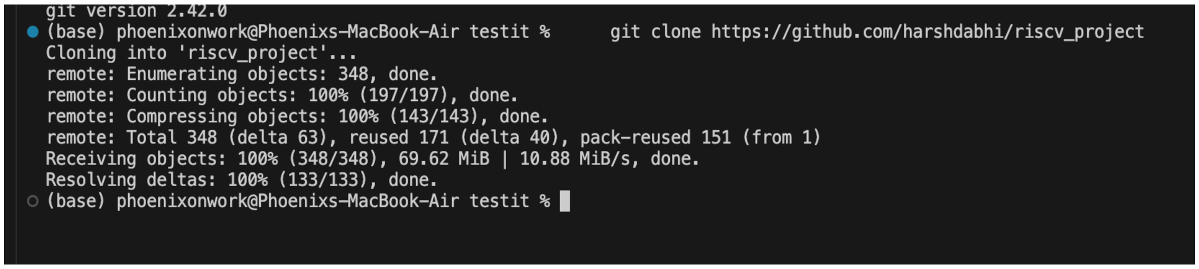
```
git --version
```



```
● (base) phoenixonwork@Phoenixs-MacBook-Air testit % git --version
  git version 2.42.0
○ (base) phoenixonwork@Phoenixs-MacBook-Air testit %
```

4. Clone the project repository:

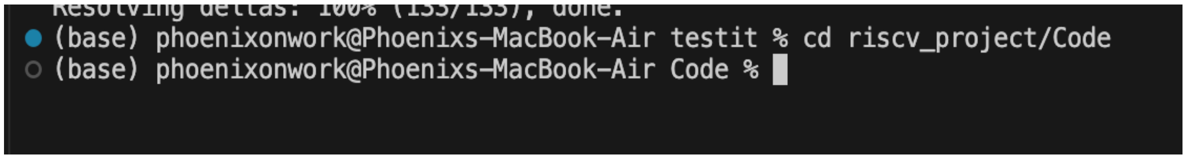
   ```
   git clone https://github.com/harshdabhi/riscv_project
   ```



```
  git version 2.42.0
● (base) phoenixonwork@Phoenixs-MacBook-Air testit %     git clone https://github.com/harshdabhi/riscv_project
  Cloning into 'riscv_project'...
  remote: Enumerating objects: 348, done.
  remote: Counting objects: 100% (197/197), done.
  remote: Compressing objects: 100% (143/143), done.
  remote: Total 348 (delta 63), reused 171 (delta 40), pack-reused 151 (from 1)
  Receiving objects: 100% (348/348), 69.62 MiB | 10.88 MiB/s, done.
  Resolving deltas: 100% (133/133), done.
○ (base) phoenixonwork@Phoenixs-MacBook-Air testit %
```

5. Navigate to the code folder:

   ```
   cd riscv_project/code
   ```



```
  Resolving deltas: 100% (133/133), done.
● (base) phoenixonwork@Phoenixs-MacBook-Air testit % cd riscv_project/Code
○ (base) phoenixonwork@Phoenixs-MacBook-Air Code %
```

Step 3: Install Docker

1. Download Docker Desktop: https://www.docker.com/products/docker-desktop
2. Install Docker and ensure it is running.

3. Verify installation:

```
docker -version
```

Step 4: Set Up the RISC-V Environment using Docker

Option 1: Build Docker Image (~1 hour)

1. Open terminal in Cursor.
2. Navigate to folder containing Dockerfile.
3. Build the Docker image:

```
docker build -t riscv_simulator:latest .
```

Option 2: Pull Pre-Built Docker Image

1. Pull the image:

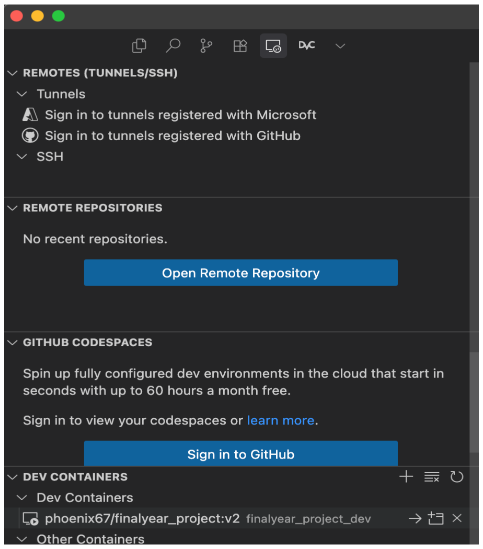
```
docker pull phoenix67/finalyear_project:v2
```

2. Run the container and mount your code folder :

```
docker run -it -v $(pwd):/app phoenix67/finalyear_project:v2
```

Step 5: Use Cursor Remote Development

1. Open Cursor.
2. On left hand side click on the icon shown in image and toggle to dev containers
3. Click on the arrow button pointing on right side. This will open overall simulator in present folder (opening will take 2-3 min approx.)



REMOTES (TUNNELS/SSH)
Tunnels
Sign in to tunnels registered with Microsoft
Sign in to tunnels registered with GitHub
SSH

REMOTE REPOSITORIES
No recent repositories.
Open Remote Repository

GITHUB CODESPACES
Spin up fully configured dev environments in the cloud that start in seconds with up to 60 hours a month free.
Sign in to view your codespaces or learn more.
Sign in to GitHub

DEV CONTAINERS
Dev Containers
phoenix67/finalyear_project:v2 finalyear_project_dev
Other Containers

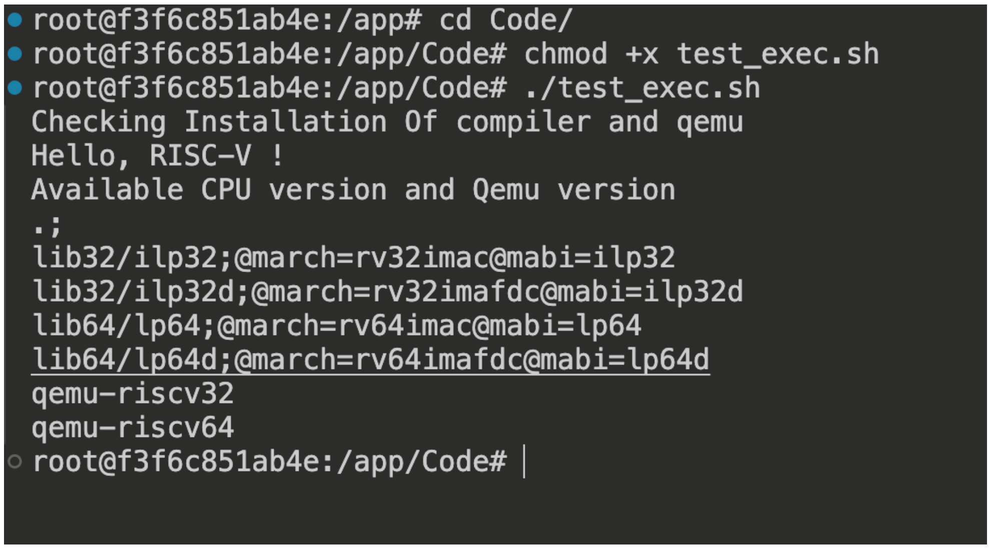
Step 6: Validate Installation

1. Make the validation script executable:

```
chmod +x test_exec.sh
```

2. Run the script:

```
./test_exec.sh
```



```
root@f3f6c851ab4e:/app# cd Code/
root@f3f6c851ab4e:/app/Code# chmod +x test_exec.sh
root@f3f6c851ab4e:/app/Code# ./test_exec.sh
Checking Installation Of compiler and qemu
Hello, RISC-V !
Available CPU version and Qemu version
.;
lib32/ilp32;@march=rv32imac@mabi=ilp32
lib32/ilp32d;@march=rv32imafdc@mabi=ilp32d
lib64/lp64;@march=rv64imac@mabi=lp64
lib64/lp64d;@march=rv64imafdc@mabi=lp64d
qemu-riscv32
qemu-riscv64
root@f3f6c851ab4e:/app/Code# 
```

If you see this message everything has been installed perfectly and running. It also show available cpu and qemu version.

3. Script checks:
   - RISC-V toolchain installation
   - QEMU simulator installation
   - Available tool versions

   Success message confirms proper installation.

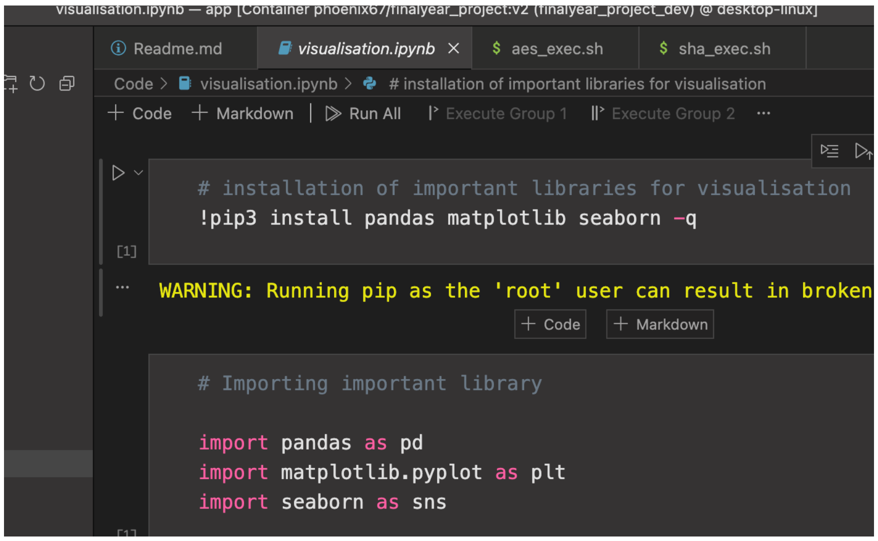
4. Similarly you can produce overall project results by following this steps
   - Make the validation script executable:

     ```
     chmod +x aes_exec.sh
     chmod +x sha_exec.sh
     ```

   - Run the script:

     ```
     ./aes_exec.sh
     ./sha_exec.sh
     ```

5. For visualization just click on visualization.ipynb notebook
   - Click on "Run all" button


visualisation.ipynb — app [Container phoenix67/finalyear_project:v2 (finalyear_project_dev) @ desktop-linux]
Readme.md
visualisation.ipynb
aes_exec.sh
sha_exec.sh
Code > visualisation.ipynb > # installation of important libraries for visualisation
+ Code
+ Markdown
Run All
Execute Group 1
Execute Group 2

```
# installation of important libraries for visualisation
!pip3 install pandas matplotlib seaborn -q
```

```
WARNING: Running pip as the 'root' user can result in broken
```

```
# Importing important library

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

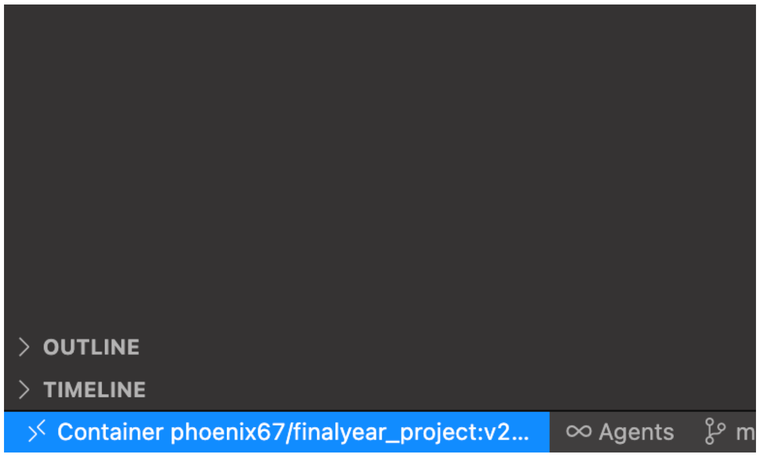
Step 7: Start Developing

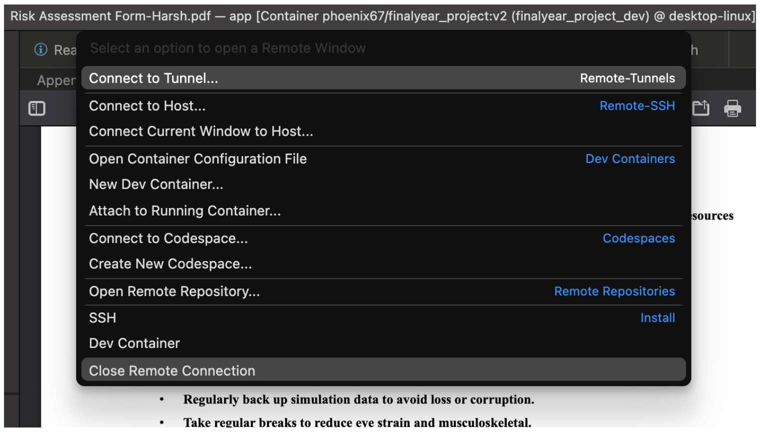
- Write RISC-V assembly or C programs
- Compile with `riscv64-unknown-elf-gcc ( support both 32 and 64 bit )`
- Simulate using QEMU

- Debug using Cursor

Step 8: Terminate session

1. Locate and click on blue color bar in bottom left of cursor terminal
2. Click on it to get the menu option
3. Click on last option to terminate the session. This will shutdown the docker container and its session.




Risk Assessment Form-Harsh.pdf — app [Container phoenix67/finalyear_project:v2 (finalyear_project_dev) @ desktop-linux]

Select an option to open a Remote Window
Connect to Tunnel... Remote-Tunnels
Connect to Host... Remote-SSH
Connect Current Window to Host...
Open Container Configuration File Dev Containers
New Dev Container...
Attach to Running Container...
Connect to Codespace... Codespaces
Create New Codespace...
Open Remote Repository... Remote Repositories
SSH Install
Dev Container
Close Remote Connection

- Regularly back up simulation data to avoid loss or corruption.
- Take regular breaks to reduce eye strain and musculoskeletal.