ZIA Threat Hunting Lab

Step-by-Step: Accessing the GitHub Codespace Lab Environment

This guide walks you through accessing the lab environment using GitHub Codespaces and validating that your Python and Jupyter setup is working correctly.

Even if you have never used Codespaces before, follow these steps carefully and you will be up and running.

This video provides a step-by-step walkthrough of this guide.

Step 1 — Log in to GitHub

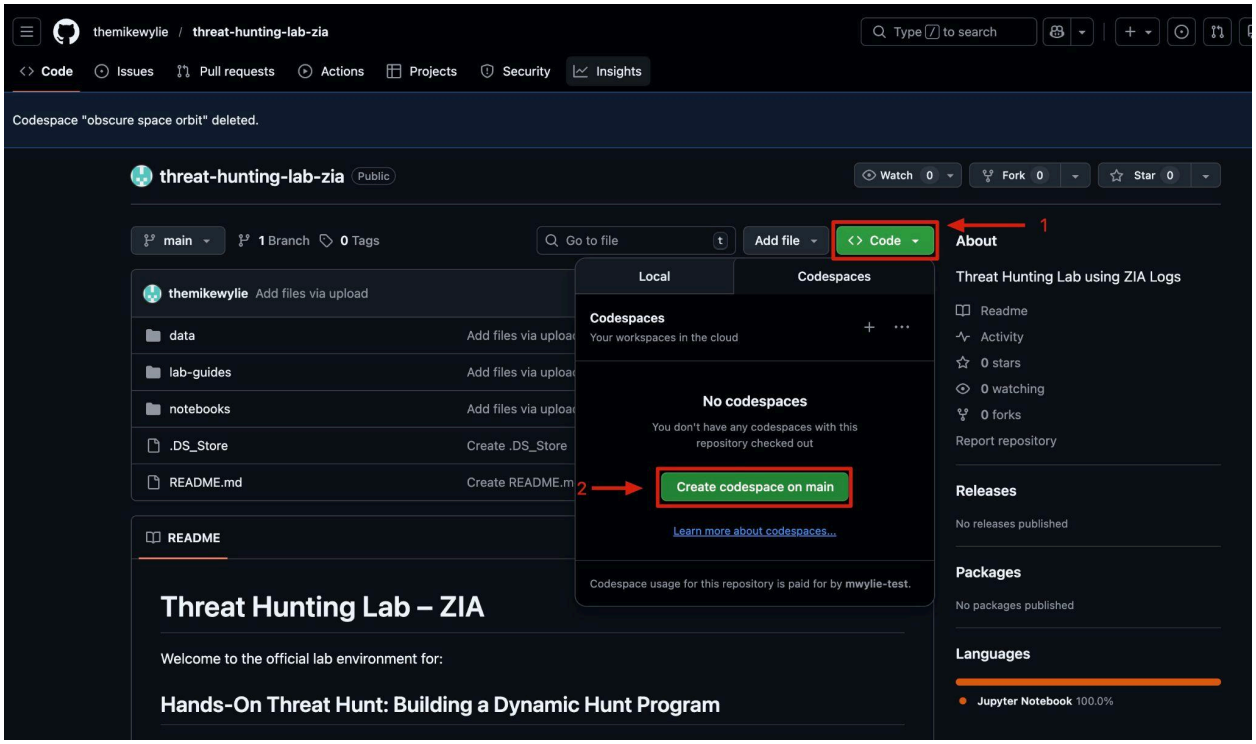
1. Navigate to: https://github.com
2. Log in using your GitHub credentials.

If you do not have a GitHub account, create one before continuing.

Step 2 — Open the Repository

1. Navigate to the lab repository URL: https://github.com/themikewylie/threat-hunting-lab-zia
2. Click the green **"Code"** button near the top-right of the repository page.
3. Select the **"Codespaces"** tab.
4. Click **"Create codespace on main"**

This will launch a new browser tab and begin building your cloud development environment.



Step 3 — Wait for the Codespace to Build

The first time you create a Codespace, it may take a few minutes.

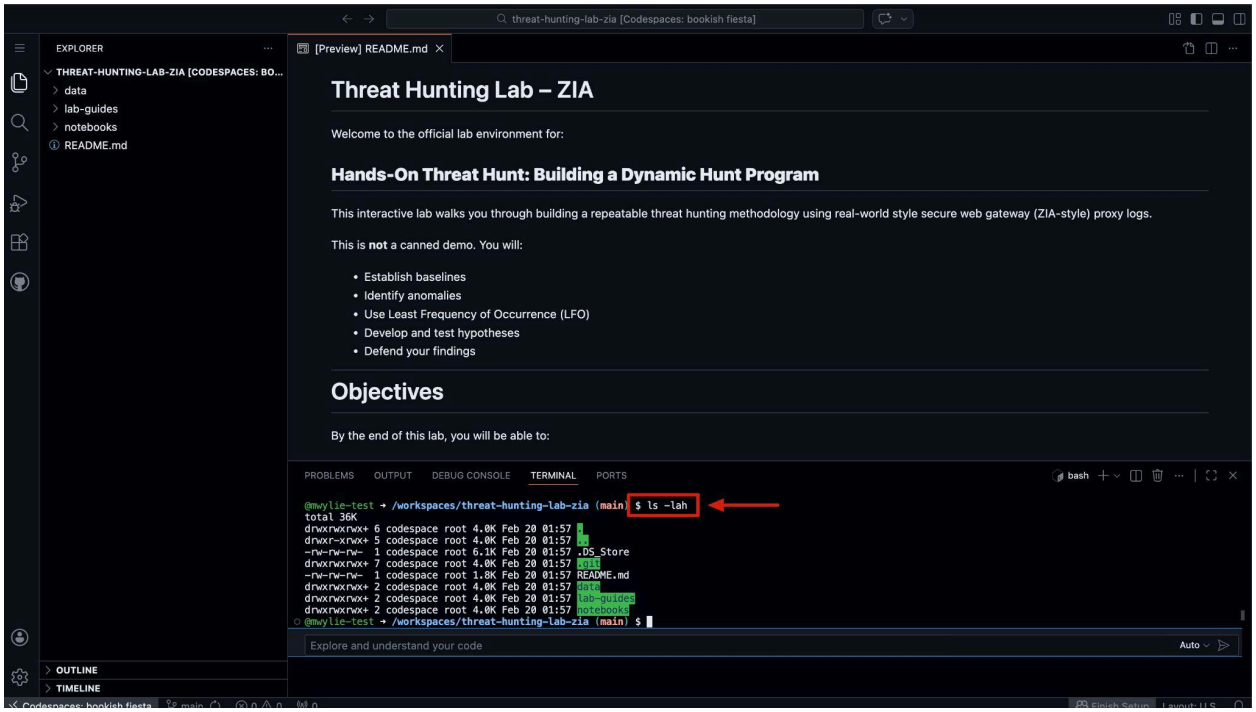
Behind the scenes, GitHub is:

- Creating a Linux container
- Installing dependencies
- Configuring Python
- Preparing your lab environment

Be patient. Do not refresh the page.

Once complete, you will see:

- A VS Code-style interface in your browser
- A README.md file open in the editor
- A terminal panel at the bottom



Step 4 — Verify Terminal Access

In the terminal panel at the bottom of the screen, type:

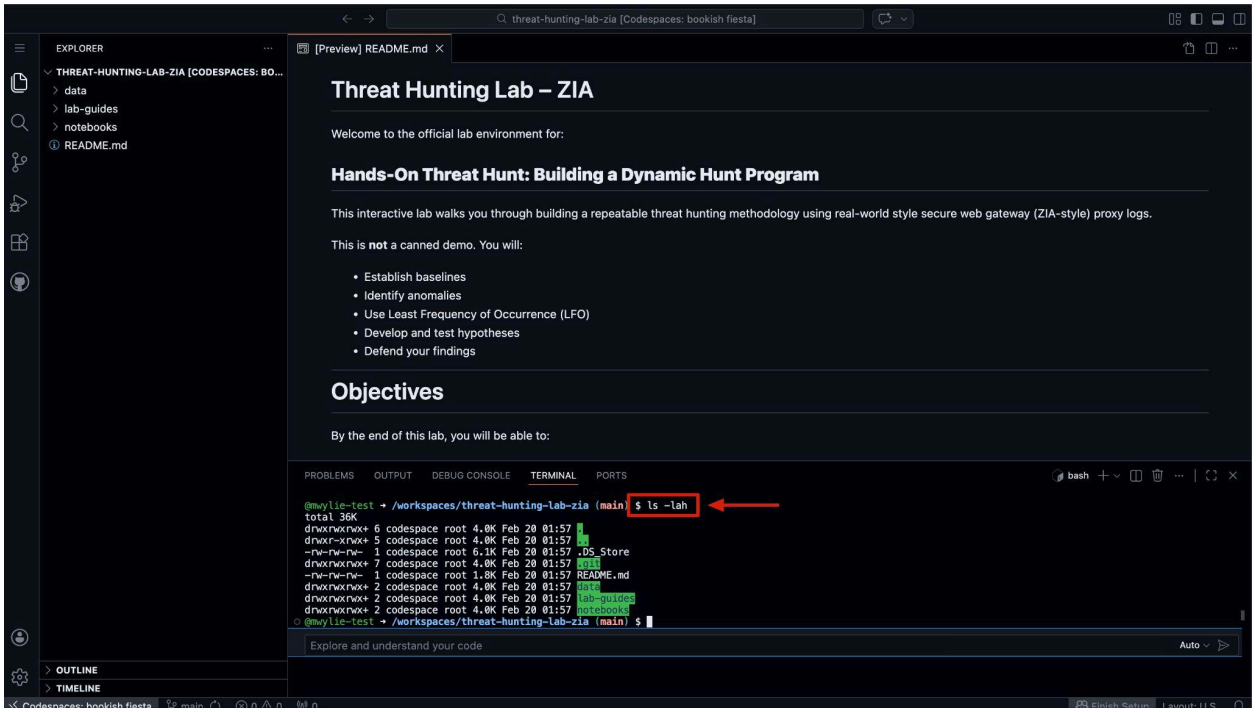
```shell
ls -lah
```

Press **Enter**.

You should see a list of files and directories in the repository.

This confirms:

- The container is working
- You have terminal access
- The repository cloned successfully



Step 5 — Verify Required Lab Directories

On the left side of the screen, you will see the **Explorer panel**.

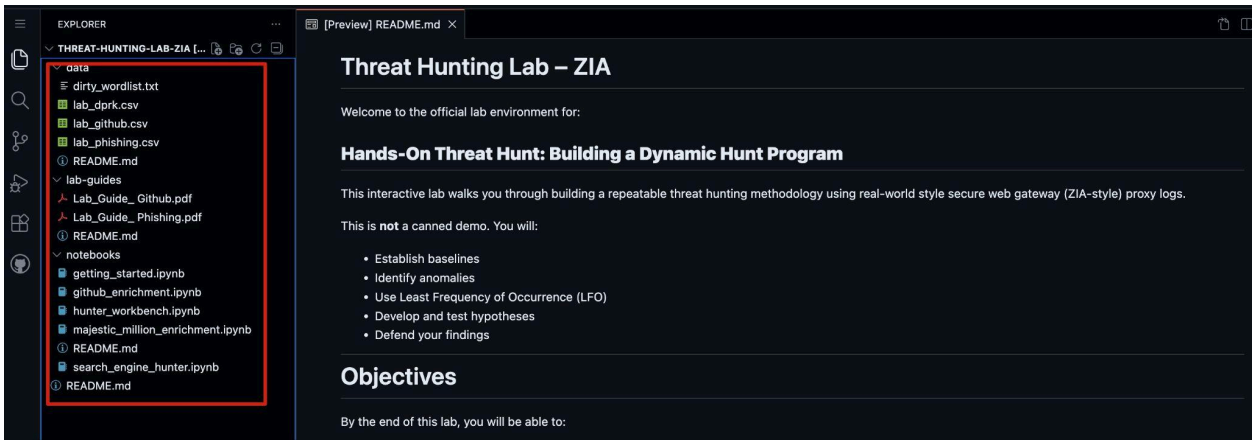
Confirm you can see the following directories:

- `/data`
- `/lab-guides`
- `/notebooks`

If you see these folders, your environment is loaded correctly.

If not:

- Click the Explorer icon (top-left file icon)
- Expand the repository folder



Step 6 — Open the Jupyter Notebook

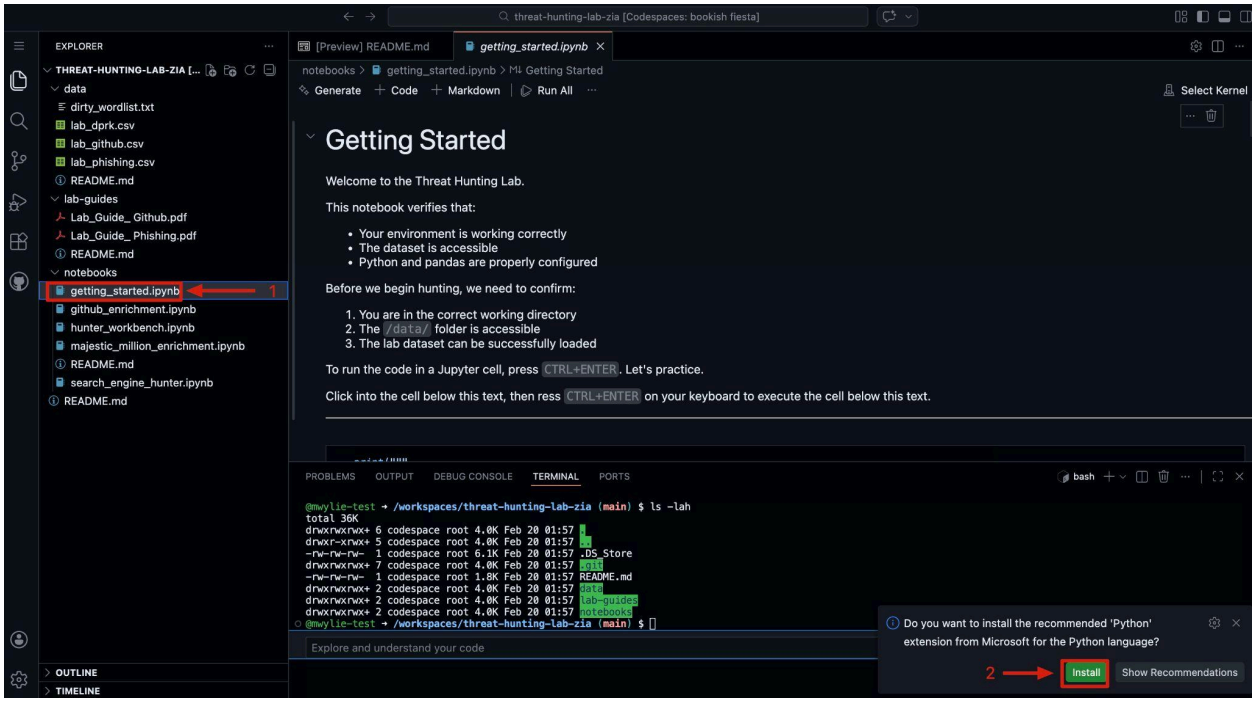
1. Navigate to:

```
None

/notebooks
```

2. Open:

```
None

getting_started.ipynb
```

⚠️ Note: The file extension is `.ipynb` (Jupyter Notebook file).



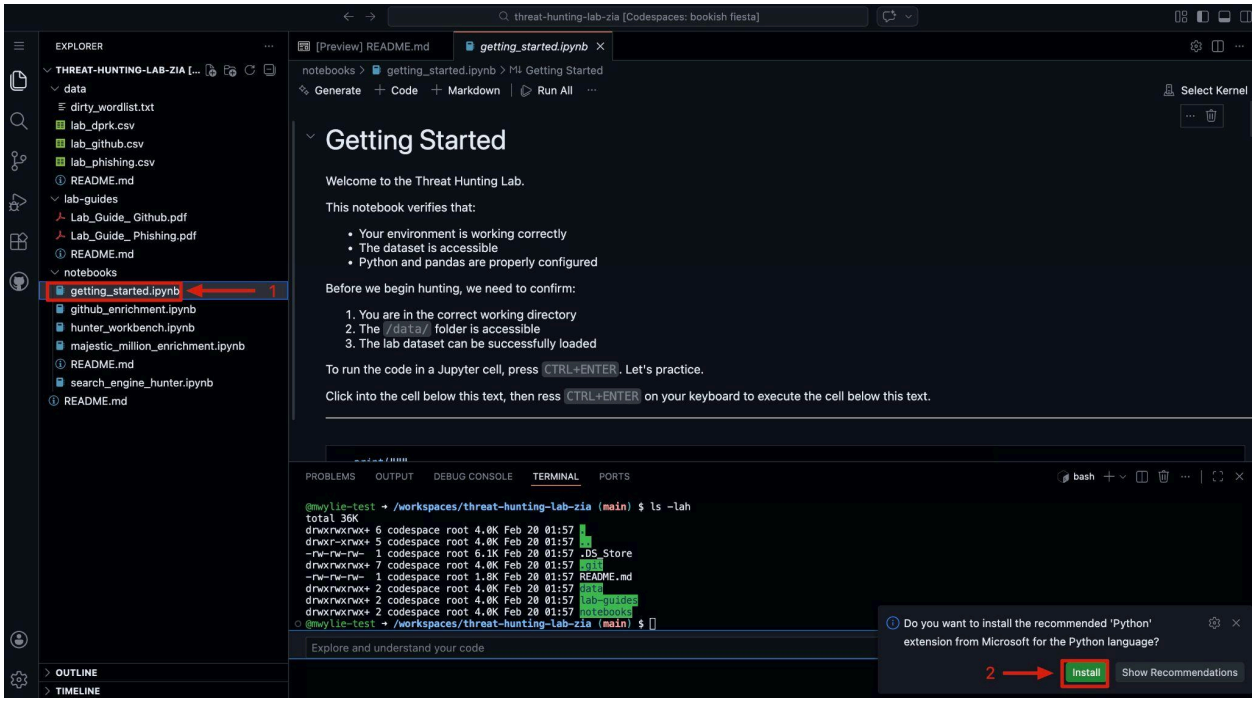
Step 7 — Install Required Extensions (First-Time Setup)

The first time you open a Jupyter notebook in Codespaces, you will likely see a prompt asking to install:

- Python extension; and/or
- Jupyter extension

These are required.

At the bottom of the screen, click **Install**.



Step 8 — Run the First Python Test Cell

Inside `getting_started.ipynb`, you will see a simple `print()` statement.

Click inside the first code cell.

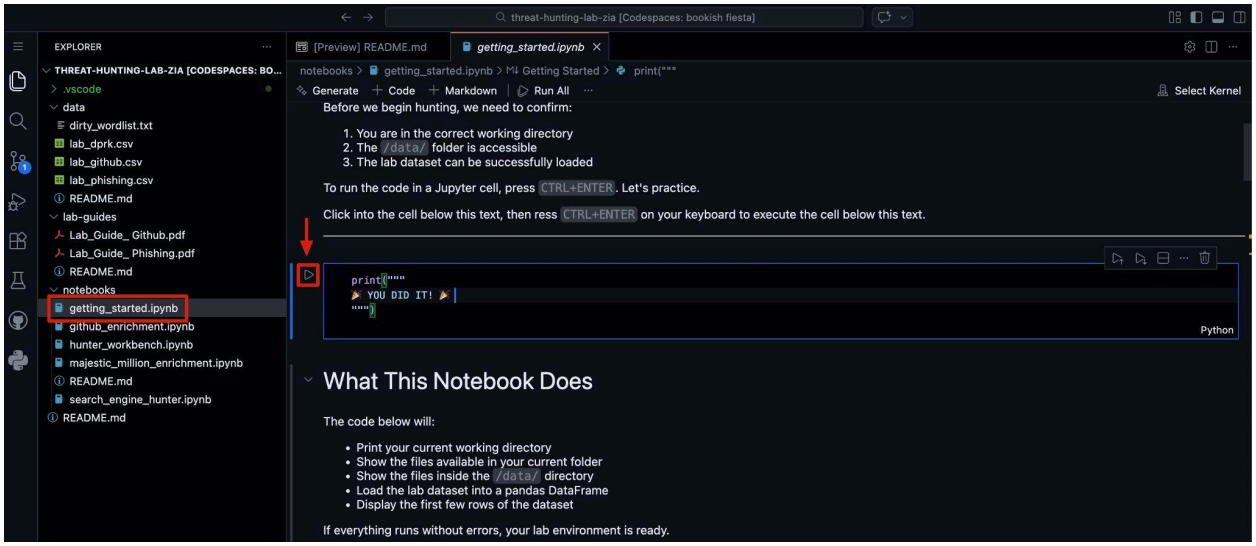
Then either:

- Press **Shift + Enter**
- Or click the ▶ Run button

If everything is configured properly, you will see output appear below the cell.

This confirms:

- Python is working
- Jupyter is connected
- The kernel is active



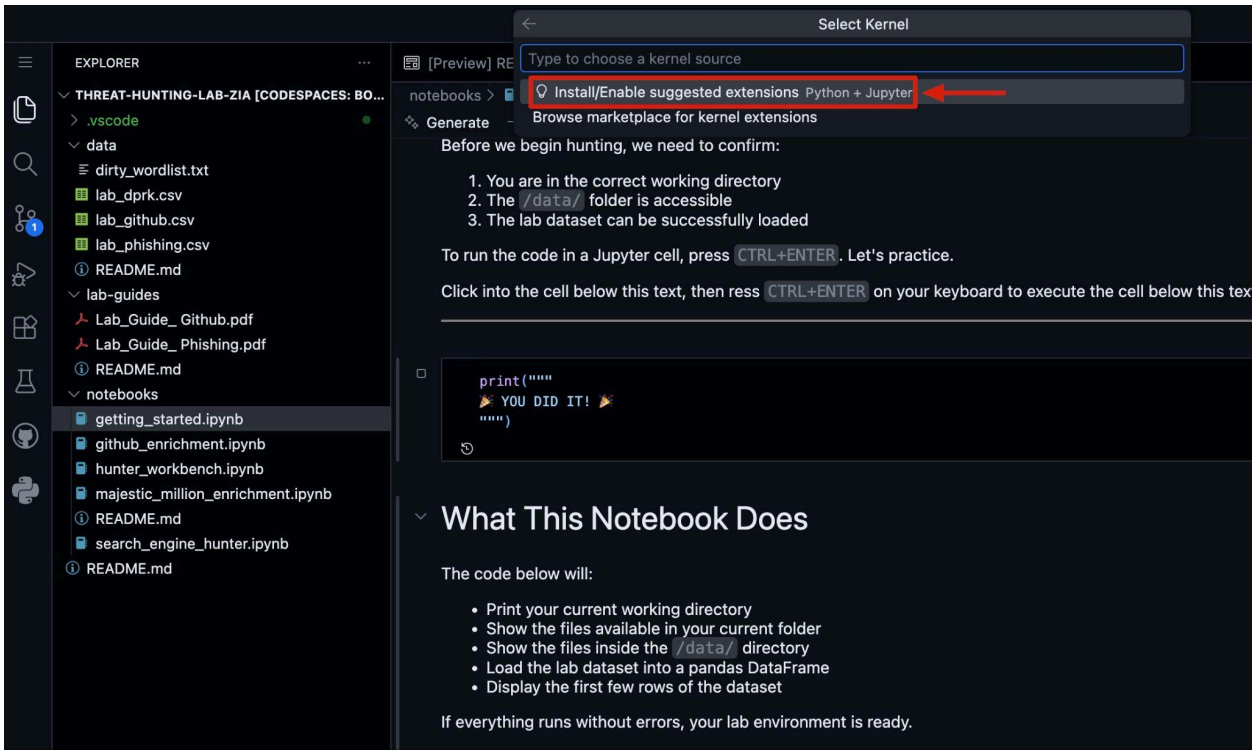
Step 9 — Select the Correct Python Interpreter

When you try to run the first code cell, you will likely be prompted to Install/Enable suggested extensions “Python + Jupyter”.

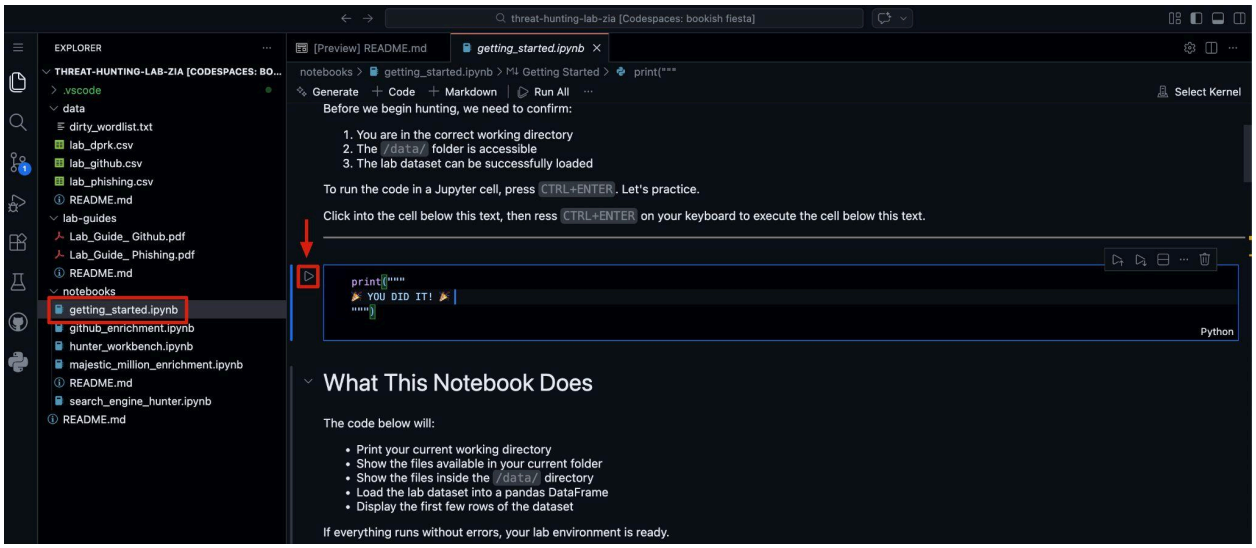
At the top of the screen:

1. Click **Install** or **Enable** for the suggested extensions.
2. Wait for installation to complete.

This only needs to be done once per Codespace.



When you try to run the first code cell again, you will likely be prompted to select a Python environment.



Click:

“Select Kernel” → “Python Environments…”

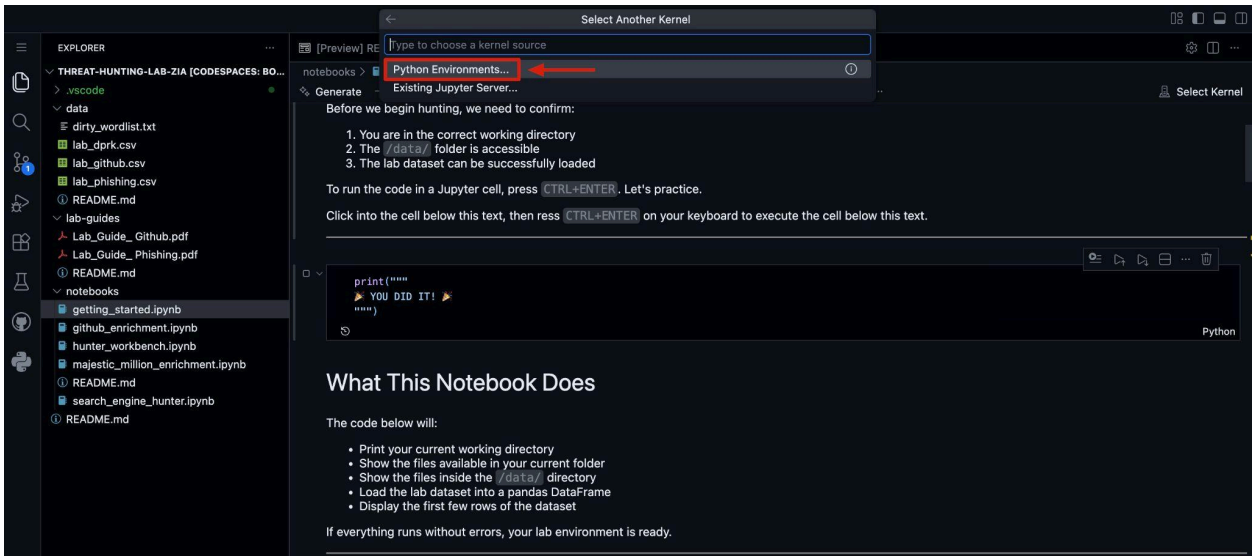
Then choose:

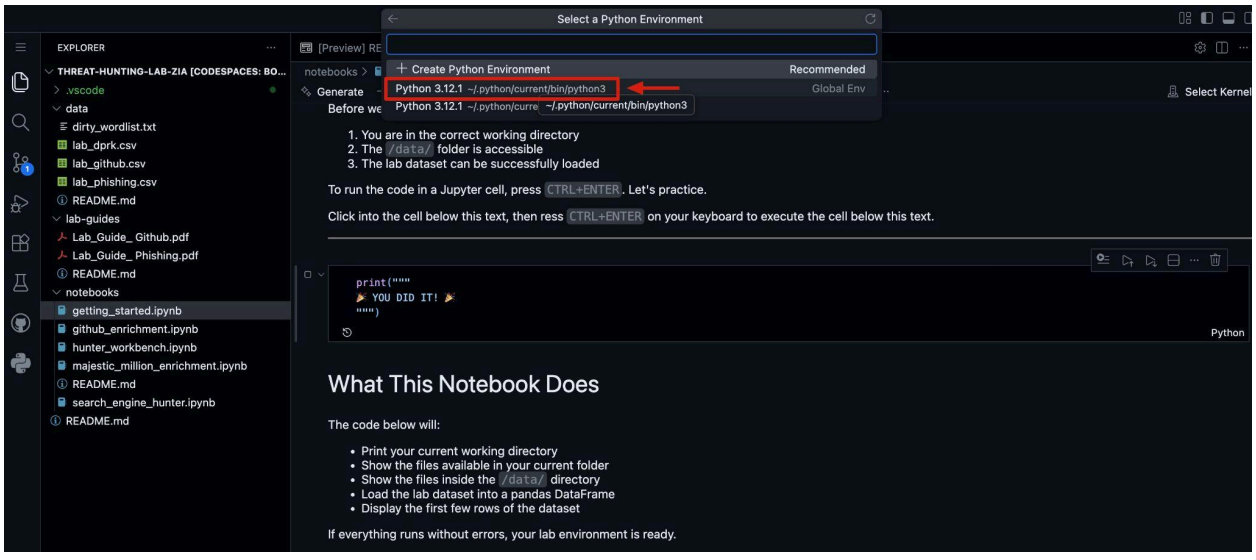
```
None
Python 3.12.1 ~/python/current/bin/python3
```

This ensures:

- You are using the correct interpreter
- All lab dependencies will install properly
- Your notebook can execute Python code

If you do not select the correct environment, cells may fail to run.





Step 10 — Run the First Python Test Cell

Inside `getting_started.ipynb`, you will see a simple `print()` statement.

Click inside the first code cell.

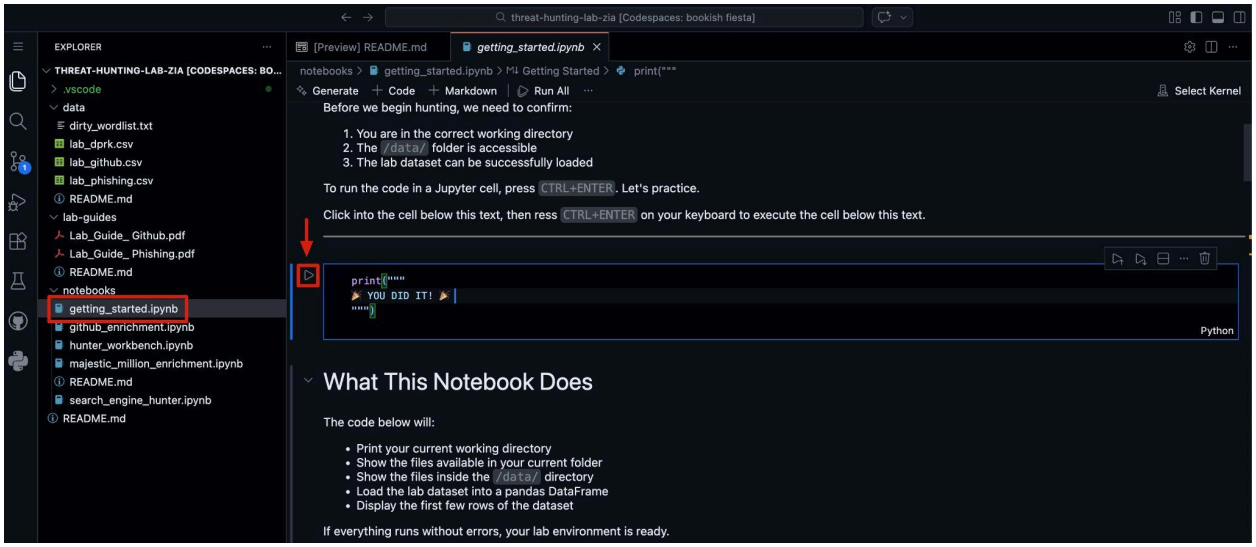
Then either:

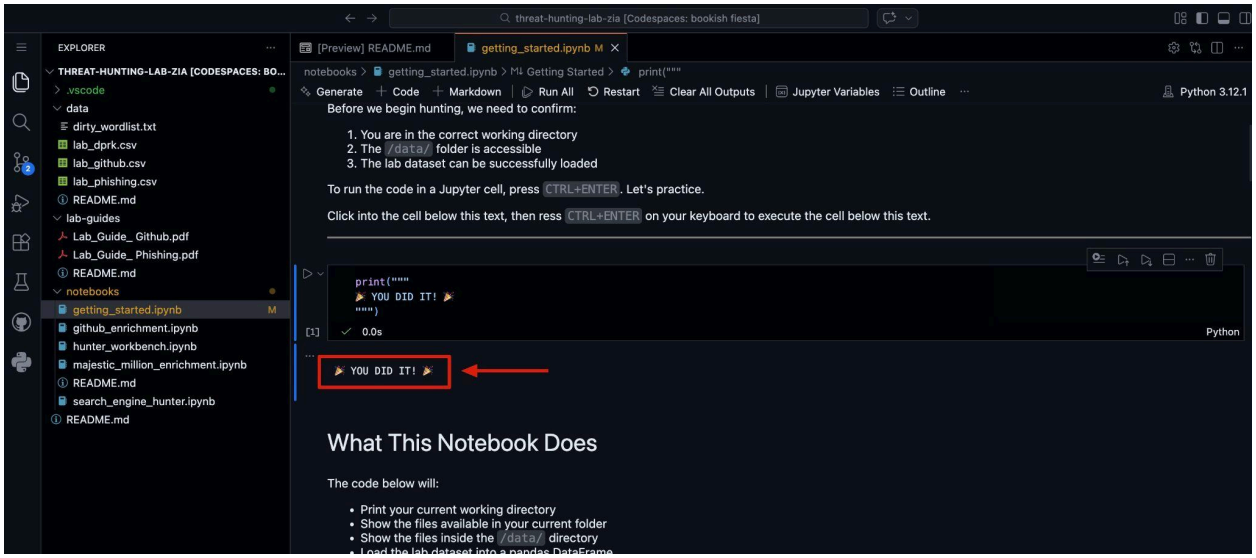
- Press **Shift + Enter**
- Or click the ▶ Run button

If everything is configured properly, you will see output appear below the cell.

This confirms:

- Python is working
- Jupyter is connected
- The kernel is active





Step 10 — Run the Notebook Cells

After confirming Python works:

You have two options:

Option A — Run One Cell at a Time (Recommended for Learning)

Press **Shift + Enter** on each cell and read the output as you go.

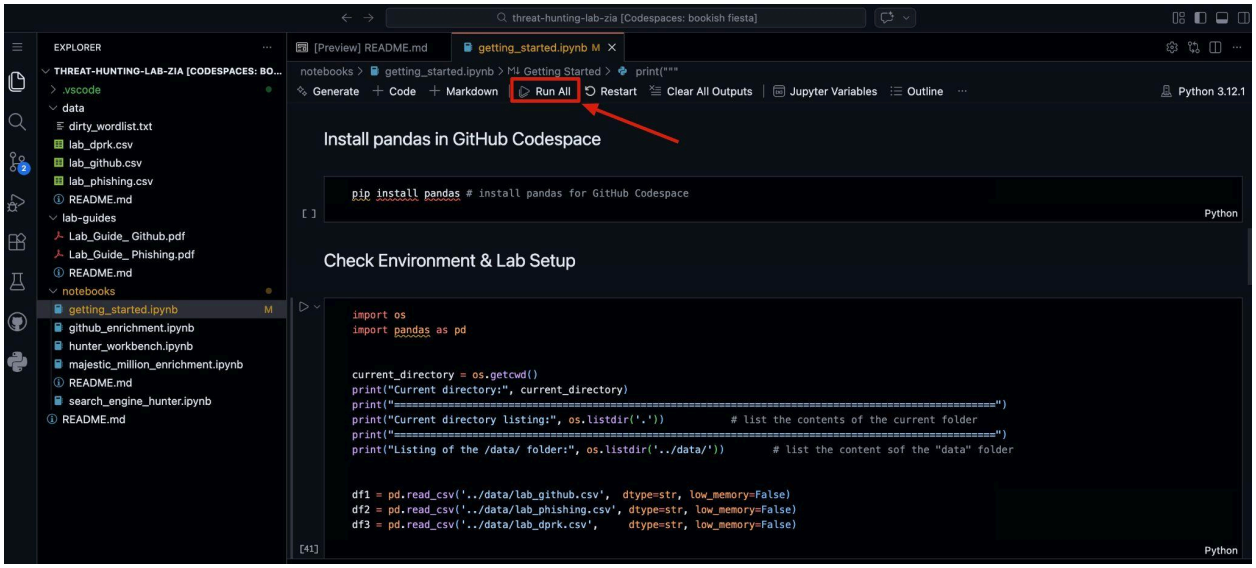
This is best for understanding what is happening.

Option B — Run All (Recommended for Quick Setup)

Click **“Run All”** at the top of the notebook.

This will:

- Install required Python packages (via pip)
- Load CSV files
- Validate data access



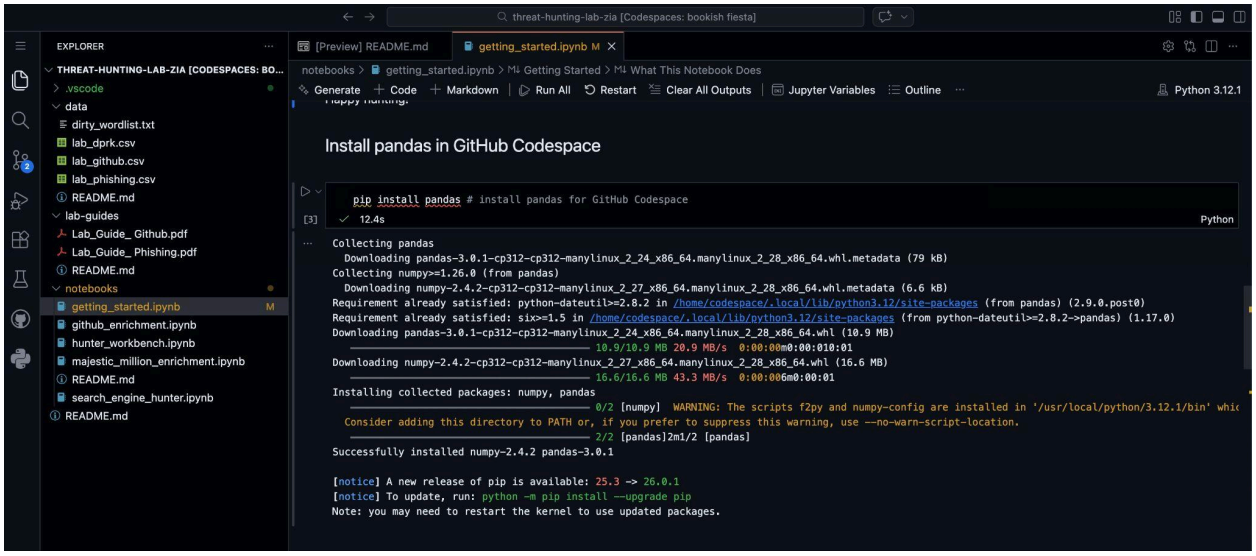
Step 11 — Monitor Package Installation

If this is your first run, you may see pip installation output.

This is normal.

Scroll through the output and look for:

- Successful installs
- No red error messages
- Confirmation that dependencies were installed



Step 12 — Verify CSV Data Loads Correctly

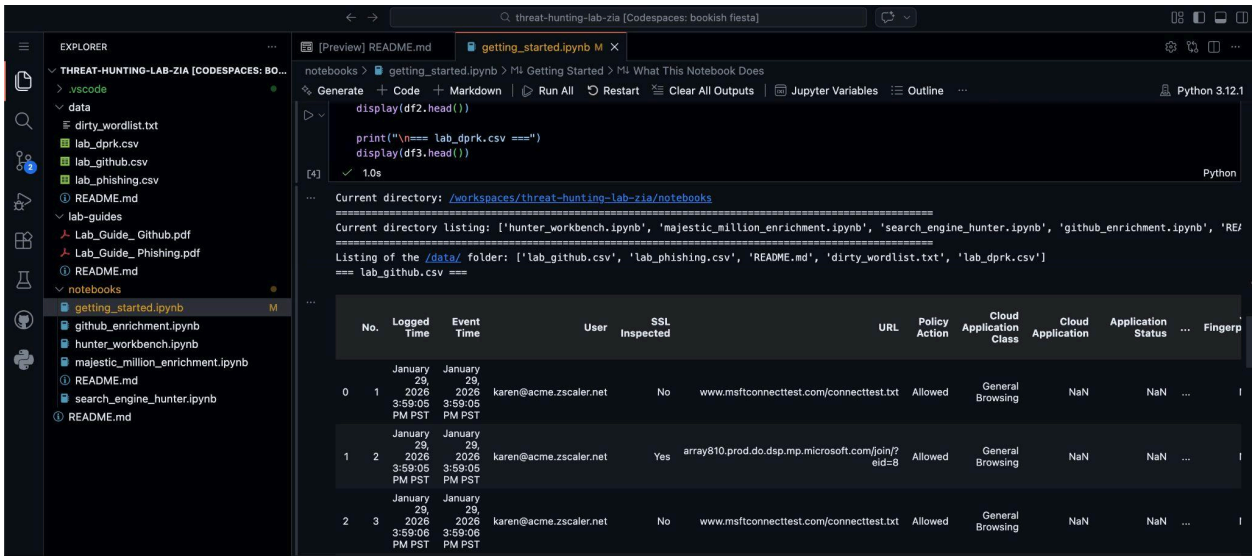
The notebook will read lab datasets from:

```
None
../data/
```

You should see confirmation that:

- `Lab_github.csv, lab_phishing.csv, lab_dprk.csv` loads successfully
- A dataframe preview appears
- No file path errors occur

If you see a dataframe preview (rows and columns displayed), your environment is fully operational.



Troubleshooting Tips

If something does not work:

Notebook will not run?

- Ensure Python + Jupyter extensions are installed.
- Confirm correct interpreter selected (Python 3.12.1).

CSV not found?

- Confirm you are running from the `/notebooks` directory.
- Confirm `/data` directory exists.

Terminal not visible?

- Click **View** → **Terminal** to reopen it.

All Else Fails?

- Restart the kernel 

What You Have Now

If all steps succeeded, you now have:

- A fully functional cloud-based lab environment
- Python 3.12 configured
- Jupyter Notebook operational
- Access to all lab datasets
- Terminal access for bash-based hunting

You are ready to begin threat hunting. Happy hunting!