

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

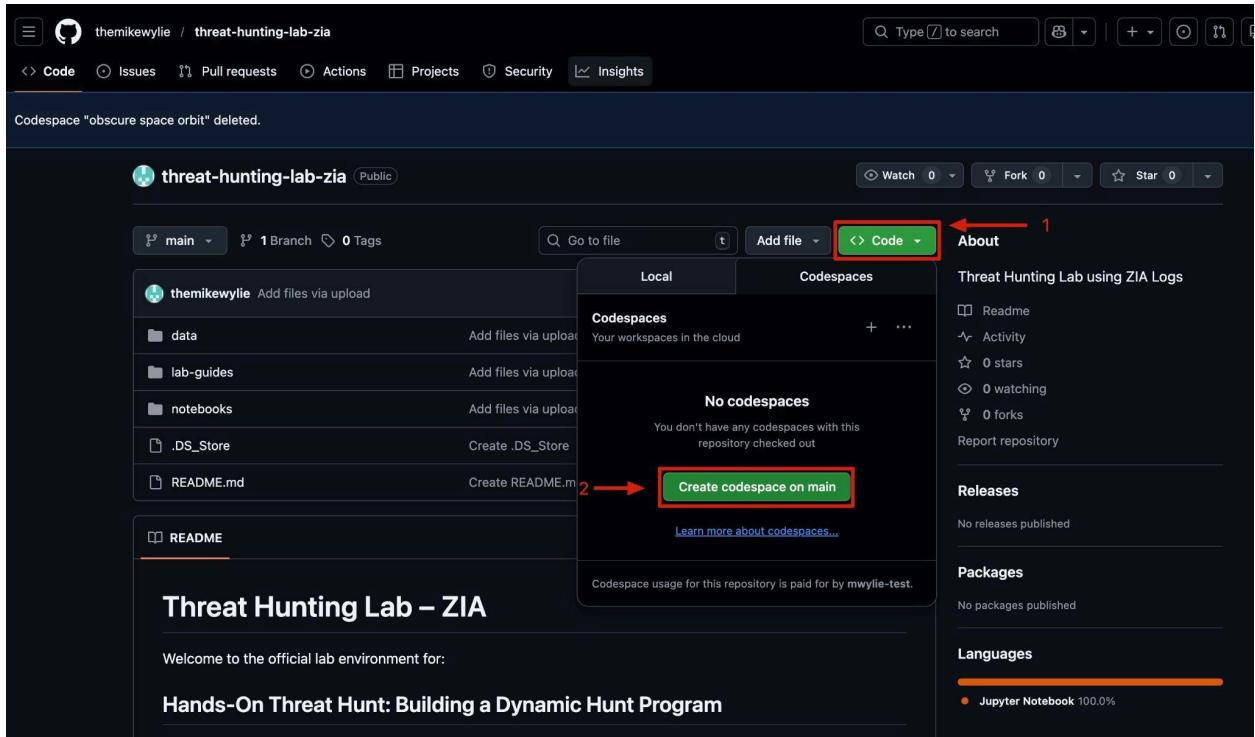
1. Navigate to: <https://github.com>
2. Log in using your GitHub credentials.

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.

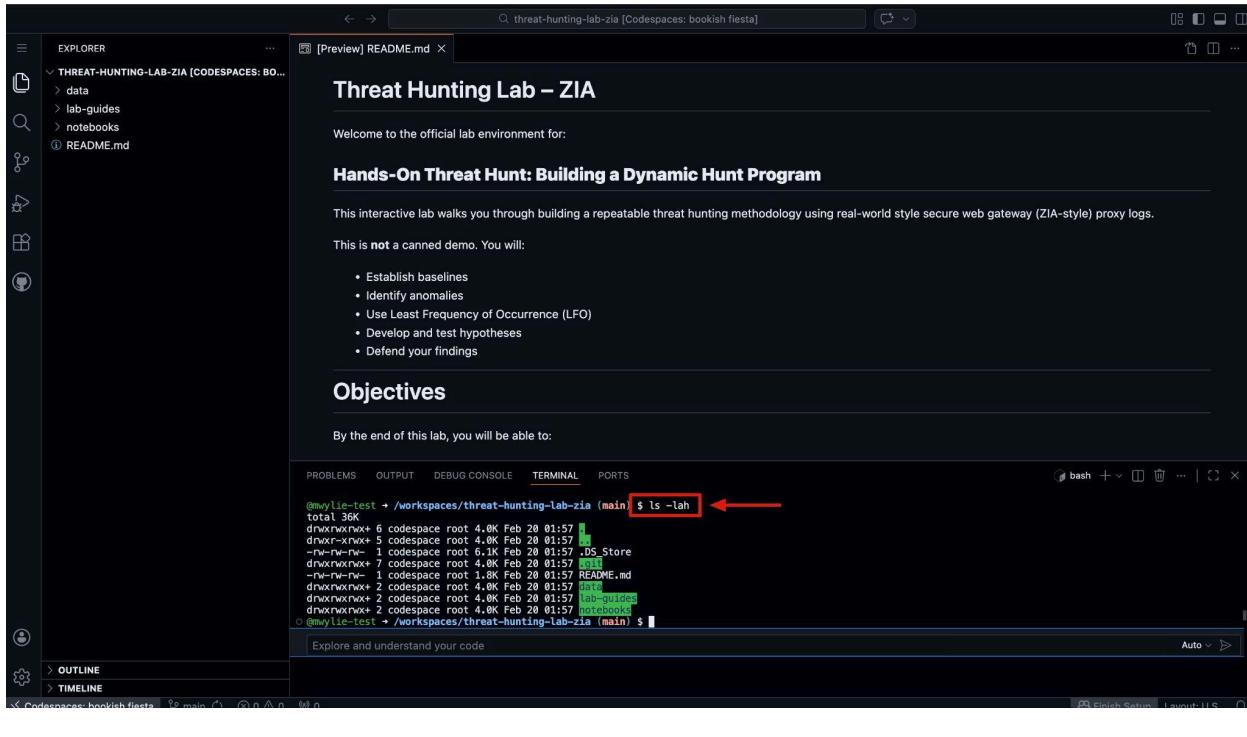
Behind the scenes, GitHub is:

- 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:

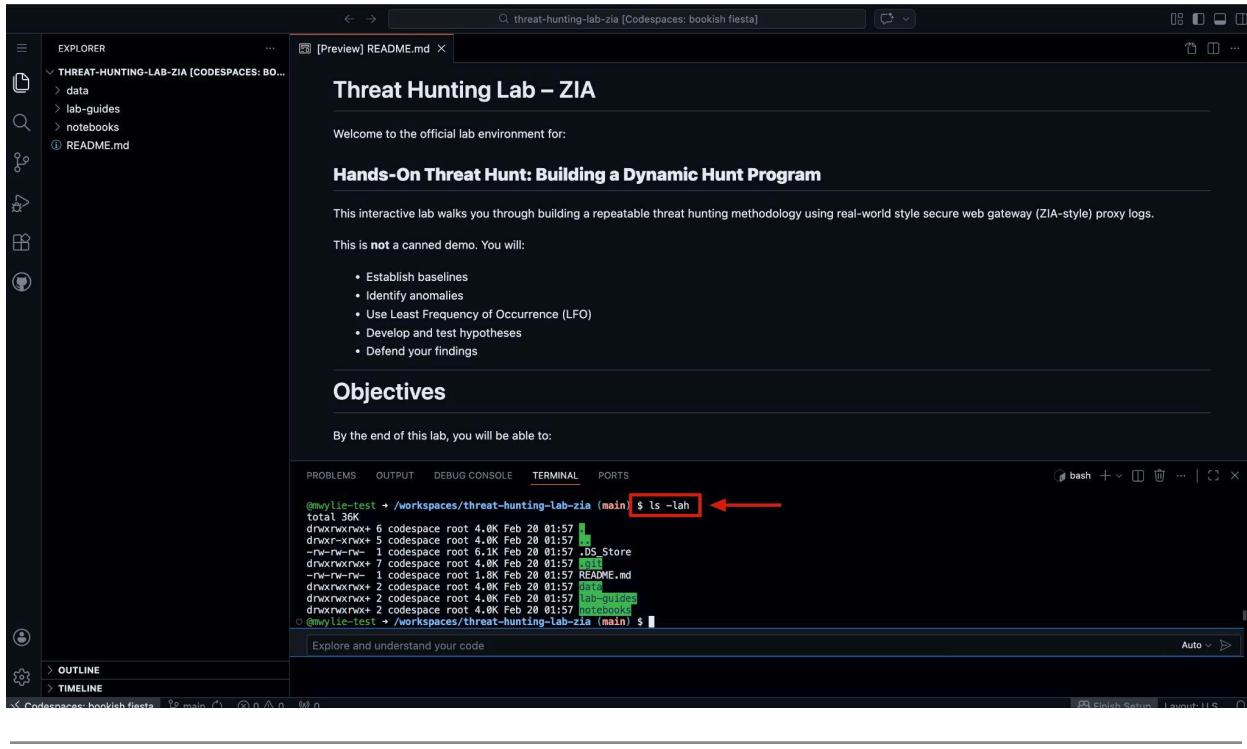
```
Shell  
ls -lah
```

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.

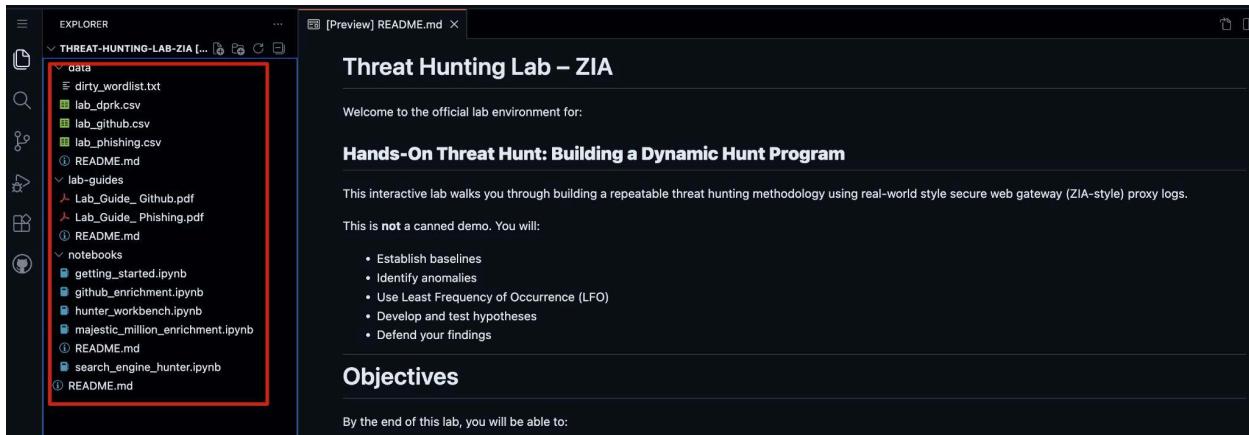
Confirm you can see the following directories:

- `/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

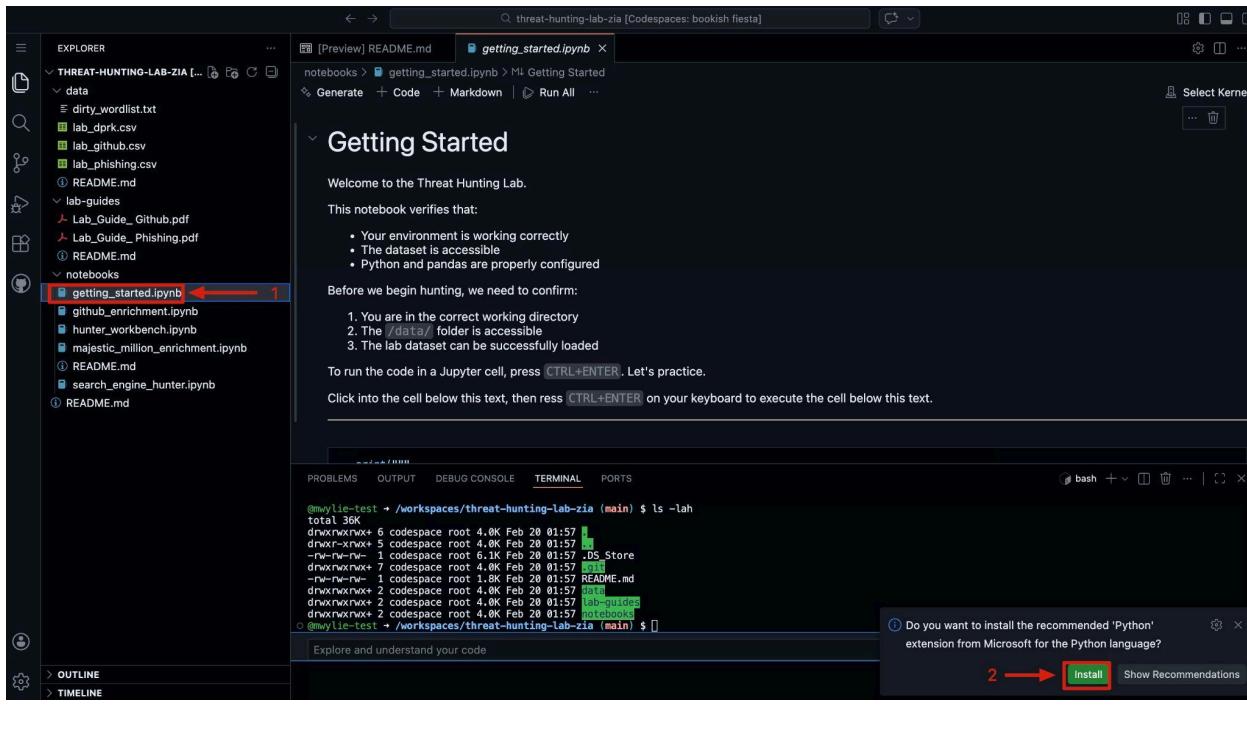
`/notebooks`

2. Open:

None

`getting_started.ipynb`

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



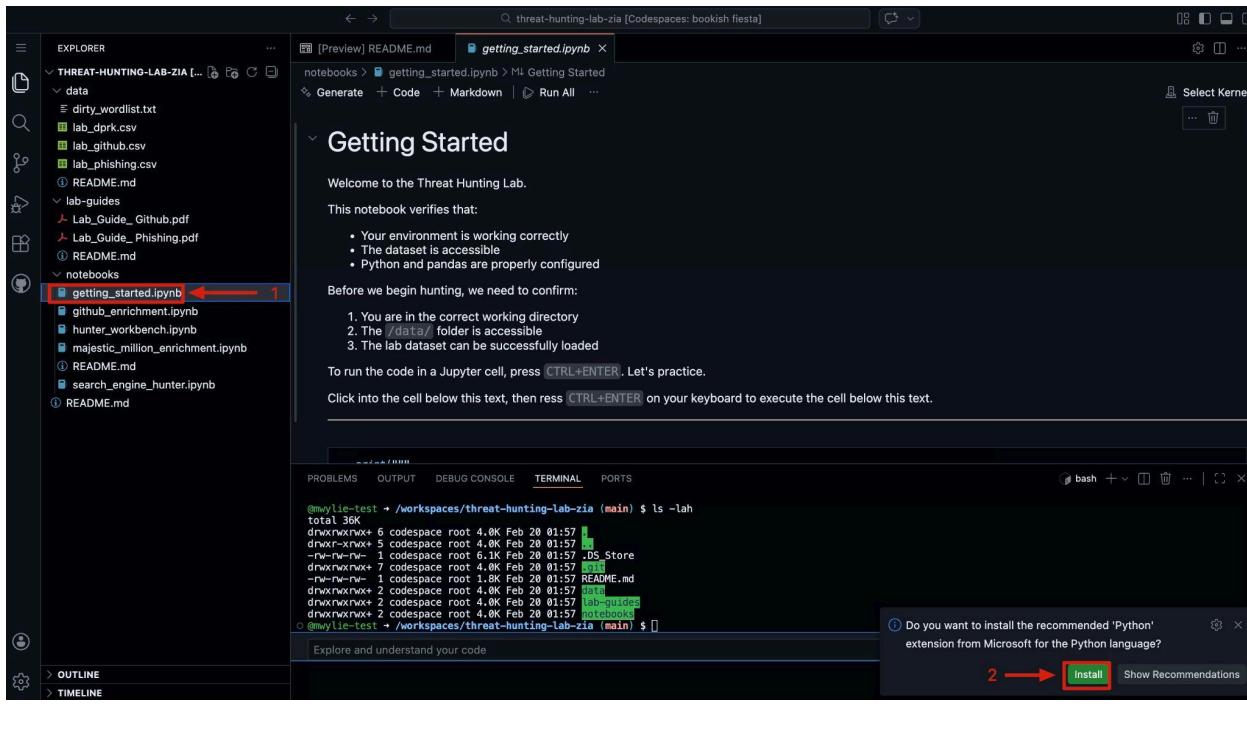
Step 7 — Install Required Extensions (First-Time Setup)

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.

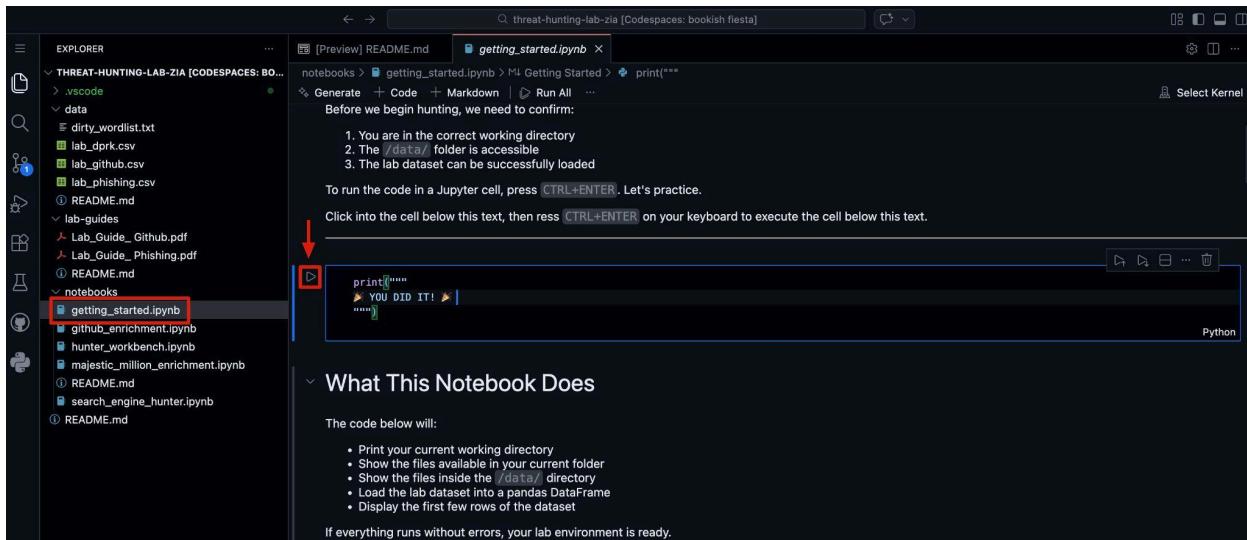
Then either:

- 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



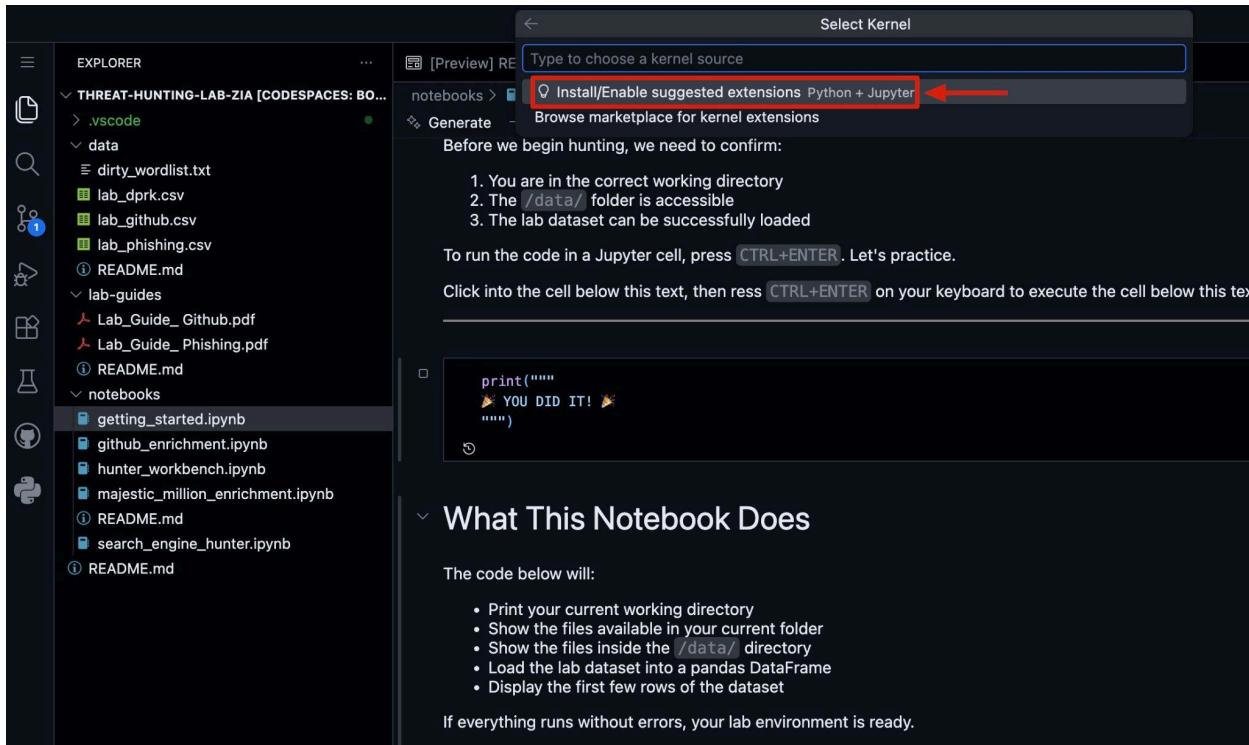
Step 9 — Select the Correct Python Interpreter

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

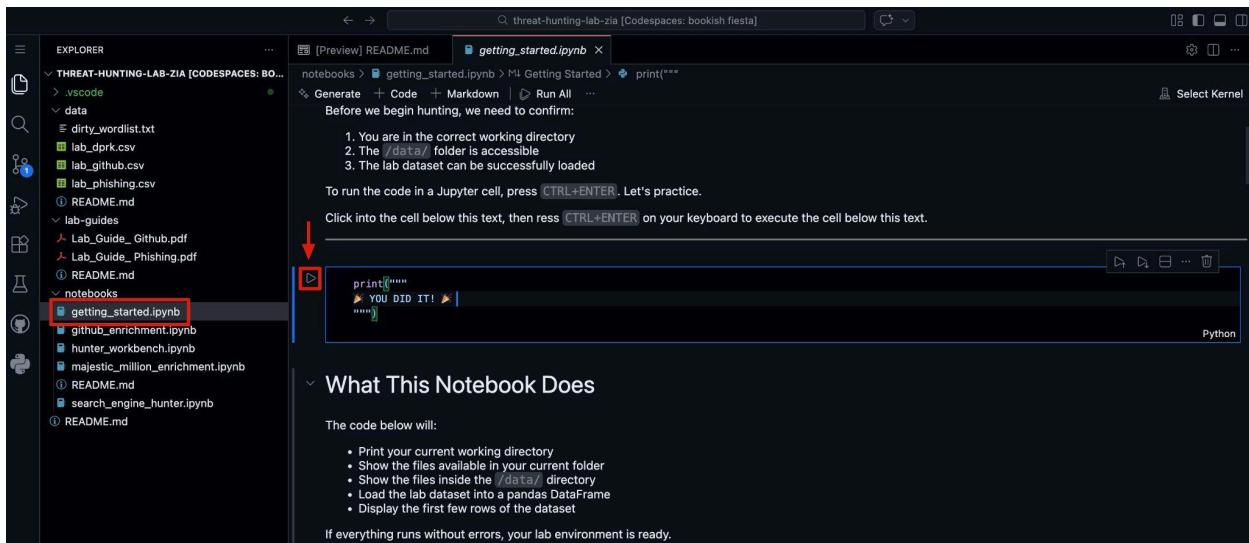
At the top of the screen:

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:

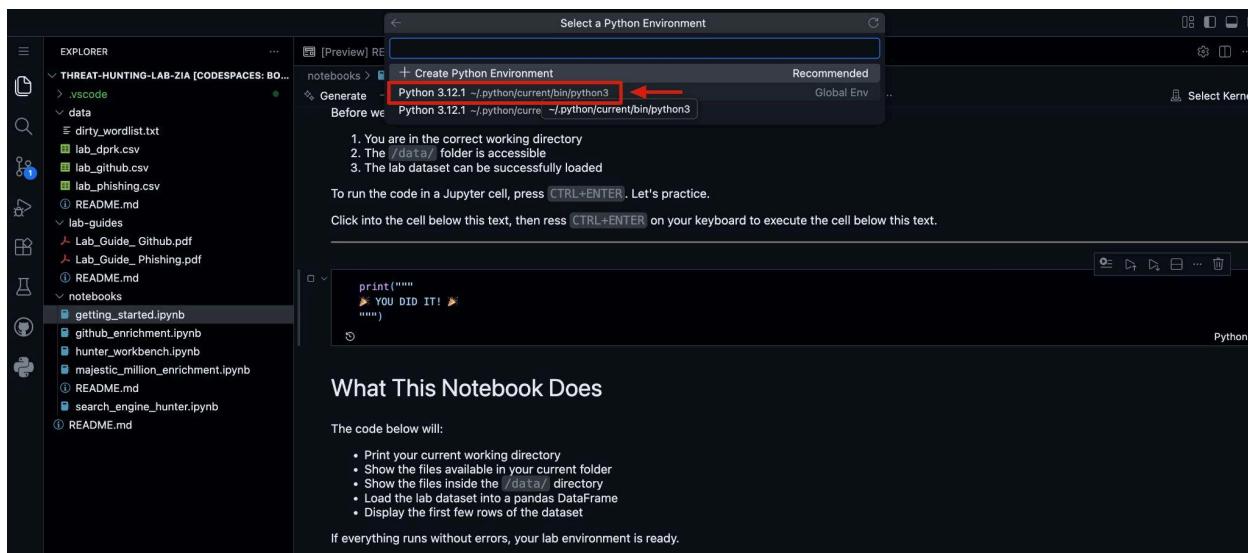
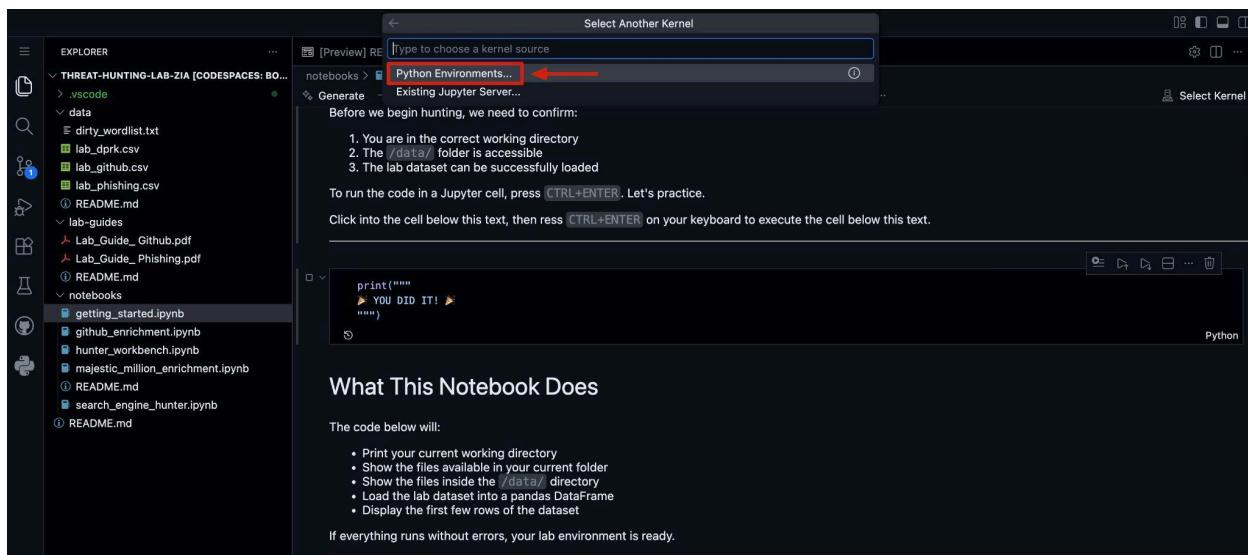
None

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.

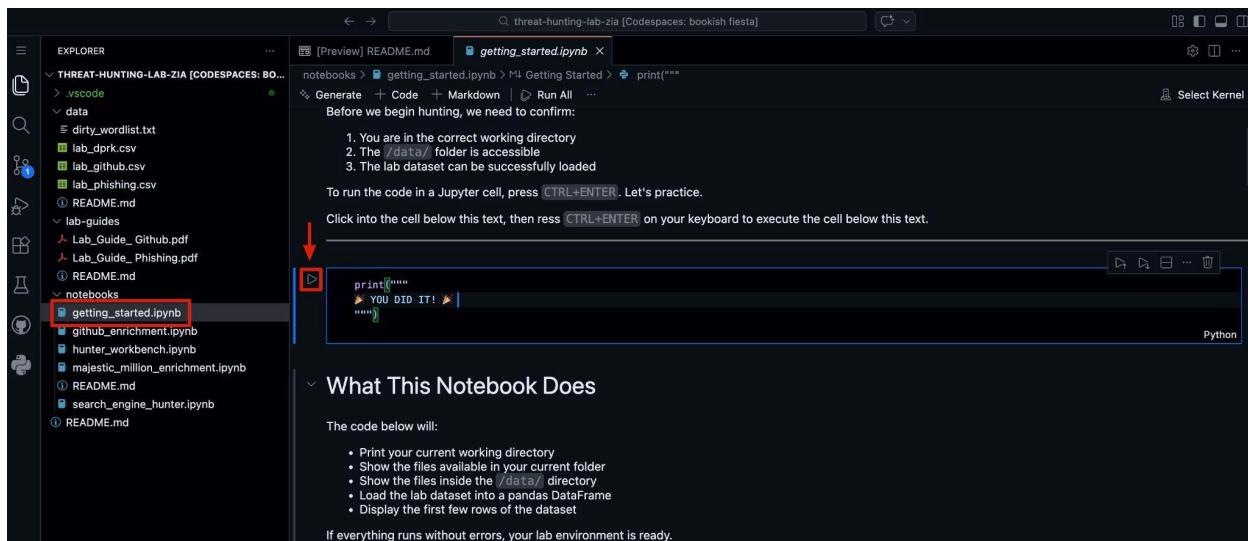
Then either:

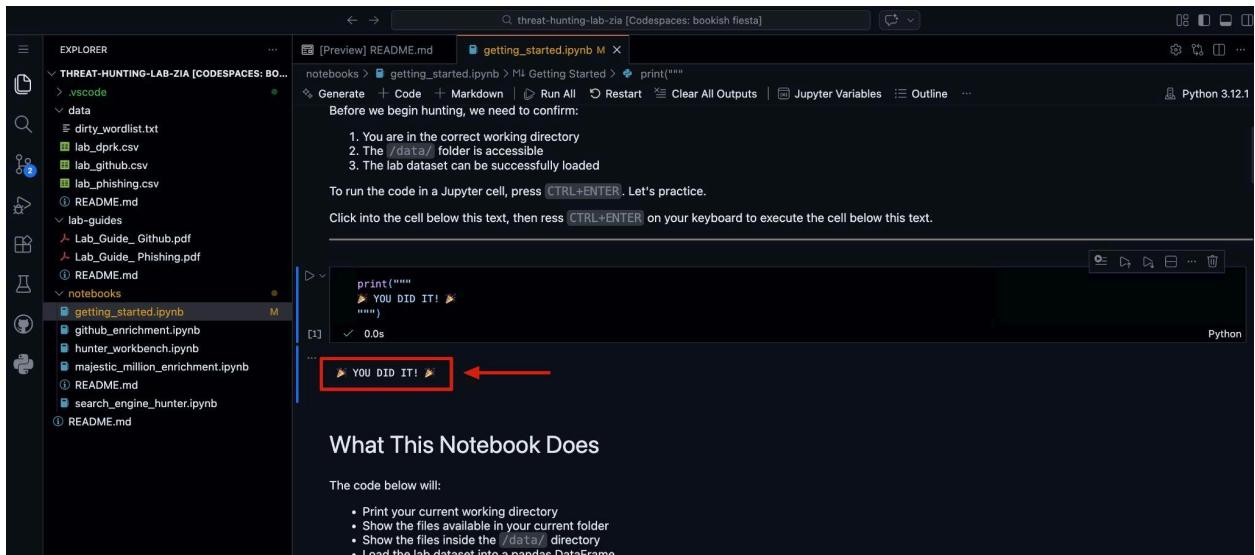
- Press **Shift + Enter**
- Or click the ► Run button

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

The screenshot shows a GitHub Codespace interface with a Jupyter Notebook open. The notebook is titled 'getting_started.ipynb'. In the top right corner of the notebook tab, there is a 'Run All' button, which is highlighted with a red arrow. The main content area of the notebook shows code for installing pandas and checking the environment.

```

pip install pandas # install pandas for GitHub Codespace

```

```

import os
import pandas as pd

current_directory = os.getcwd()
print("Current directory:", current_directory)
print("=====")
print("Current directory listing:", os.listdir('.'))
print("=====")
print("Listing of the /data/ folder:", os.listdir('../data/'))
print("=====")

df1 = pd.read_csv('../data/lab_github.csv', dtype=str, low_memory=False)
df2 = pd.read_csv('../data/lab_phishing.csv', dtype=str, low_memory=False)
df3 = pd.read_csv('../data/lab_dprk.csv', dtype=str, low_memory=False)

```

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

The screenshot shows the same GitHub Codespace interface as before, but the output pane now displays the actual pip installation logs. It shows the command 'pip install pandas', the download of numpy and pandas packages, and a warning message about scripts being installed in the Python bin directory.

```

pip install pandas # install pandas for GitHub Codespace

```

```

Collecting pandas
  Downloading pandas-3.0.1-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl.metadata (79 kB)
Collecting numpy>=1.26.0 (from pandas)
  Downloading numpy-2.4.2-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (6.6 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/codespace/.local/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /home/codespace/.local/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Collecting pandas-3.0.1-cp312-cp312-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl (10.9 kB)
  Downloading numpy-2.4.2-cp312-cp312-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (16.6 MB)
  10.9/10.9 MB 29.9 kB/s 0:00:00m0:00:01
  16.6/16.6 MB 43.3 kB/s 0:00:00m0:00:01
Installing collected packages: numpy, pandas
  0/2 [numpy] WARNING: The scripts f2py and numpy-config are installed in '/usr/local/python/3.12.1/bin' which
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  2/2 [pandas]2m1/2 [pandas]
Successfully installed numpy-2.4.2 pandas-3.0.1

[notice] A new release of pip is available: 25.3 -> 26.0.1
[notice] To update, run: python -m pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.

```

Step 12 — Verify CSV Data Loads Correctly

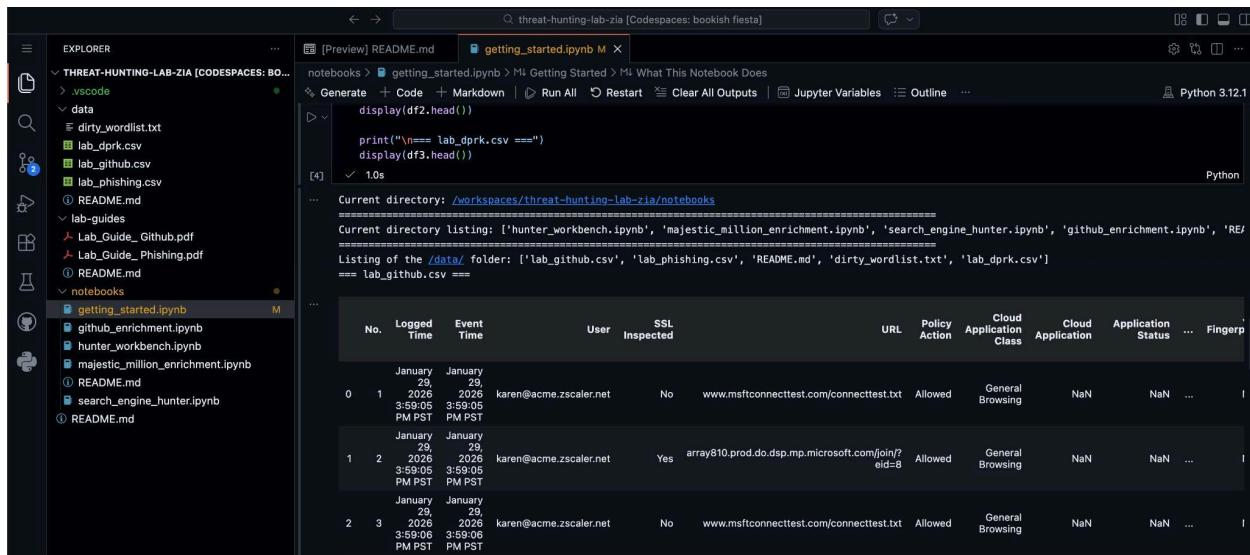
The notebook will read lab datasets from:

```
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.



The screenshot shows a Jupyter Notebook interface with the following details:

- EXPLORER:** Shows a tree view of the workspace, including a `data` folder containing `dirty_wordlist.txt`, `lab_dprk.csv`, `lab.github.csv`, `lab_phishing.csv`, `README.md`, `lab-guides` (with `Lab_Guide_Github.pdf` and `Lab_Guide_Phishing.pdf`), and `notebooks` (with `getting_started.ipynb`).
- CELLS:** One cell is active, showing Python code:

```
display(df2.head())
print("==== lab_dprk.csv ===")
display(df3.head())
```
- OUTPUT:** The output shows the first few rows of the `lab_dprk.csv` file:

```
[4]:    ✓ 1.0s
...
Current directory: /workspaces/threat-hunting-lab-zia/notebooks
=====
Current directory listing: ['hunter_workbench.ipynb', 'majestic_million_enrichment.ipynb', 'search_engine_hunter.ipynb', 'github_enrichment.ipynb', 'README.md', 'dirty_wordlist.txt', 'lab_dprk.csv']
==== lab.github.csv ===
```
- DATA TABLE:** A large data table is displayed, showing logs or events. The columns include: No., Logged Time, Event Time, User, SSL Inspected, URL, Policy Action, Cloud Application Class, Cloud Application, Application Status, and Fingerprint. The data shows three entries, all of which were allowed and categorized as General Browsing.

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  **Restart**
-

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!