How To Set Up Your Codespace

For the second half of the course, you will work in **repos made from this course template, each with a codespace**. Each lab gets its own repo, made this same way. A codespace is a Linux machine in the cloud, with every tool this course needs already installed. You can use VS Code in your browser to work in it.

It's important to build before class because the first build takes a few minutes. We want everyone to come in ready to jump right in on the activity.

Watch the setup video first, https://youtu.be/6Gy53XLUmv0, then use this sheet as your checklist.

1. Make your repo from the template

1. Signed in to GitHub, open the course template: **https://github.com/joshbrake/e190bo-devcontainer-template**
2. Click **Use this template → Create a new repository**, and fill it in:
   - **Owner: your own account**, not an organization. Codespaces bills the repo's owner, so this is what keeps your hours free.
   - **Repository name:** your choice. `e190bo-lab-05` is a good one, since Lab 5 lives here too.
   - **Visibility: Private.**
3. Click **Create repository**.

Do not click Fork. A fork of the template is stuck public, and its history isn't yours. Your git history is part of your grade from Lab 5 on, so it needs to start at your own first commit.

2. Create the codespace

1. In **your new repo** (check the name at the top-left is yours, not the template), click **Code → Codespaces → Create codespace on main**.
2. A browser tab opens with VS Code and starts building. **Leave it open.** The first build takes a few minutes. After that, it resumes in seconds.
3. You're done waiting when the terminal prints this banner:

```
E190BO environment ready.
```

3. Check that it works

In the terminal, run:

```
bash demos/check_tools.sh
```

Every line should have a ☑, and the last line should read **Everything is installed.** Then:

```
python demos/hello_web.py
```

Open the **PORTS** tab (next to TERMINAL), find port **8080**, and click the globe icon. A page saying **"Port forwarding works"** should open in a new tab. Go back to the terminal and press **Ctrl-C** to stop the server.

4. Coming back to it

- **Always reopen the same codespace.** Go to https://github.com/codespaces and click your existing codepsace. Don't create a new one each time.
- **It stops itself** after 30 minutes of inactivity. To stop it sooner, go to https://github.com/codespaces, open the **three-dot menu** next to it, and choose **Stop codespace**.
- **Your hours.** GitHub gives you 120 core-hours of Codespaces a month for free, or 180 if you have the GitHub Student Developer Pack. This codespace uses 2 cores, so that's 60 to 90 hours of actual work a month. That's plenty, as long as you don't leave it running.

If something goes wrong

What you see	What to do
No **Use this template** button	Make sure you're signed in to GitHub.
The codespace opened on the *template*, not your repo	Delete it at https://github.com/codespaces, then redo step 2 from your own repo.
A ✗ in `check_tools.sh`	Screenshot the output and post it on Zulip.
"Running in recovery mode"	The build failed. Screenshot the message and post it on Zulip.
The 8080 page doesn't load	Check that `hello_web.py` is still running in the terminal, then click the globe icon again.

Zulip first. If you're stuck on setup, someone else is too.