Workshop flow

Linux basics, GitHub, Next.js and a live restaurant website.

1. Useful Linux commands

These commands are used in the Cloud Labs terminal. Type the command in the left column and press **Enter**. A directory means a folder.

COMMAND	WHAT IT DOES
`ls`	List the contents of the current folder.
`pwd`	Print the current working directory: the folder you are in.
`clear`	Clear the terminal screen. Your files remain unchanged.
`mkdir folder`	Create a new folder named **folder** in the current directory.
`cd folder`	Change directory and go inside **folder**.
`cd ..`	Go back up one level to the parent directory.

You will use these after opening the terminal in Cloud Labs. The word **folder** is an example name; you can replace it with the folder name you want.

2. Create a GitHub account

Open github.com/signup and create an account. You can choose **Continue with Google** for sign-up.

Use the email address associated with your CookieSensei registration. Complete GitHub's email verification and keep your GitHub username handy.

WHAT WE WILL BUILD

First, create and deploy a basic Next.js project. Then use ChatGPT to turn its home page into a restaurant page for **cookie**, push the change to GitHub and see the live website update.

Student handout • Two-hour workshop • Updated 16 October 2026

CLOUD LABS + SSH

Open your lab and add an SSH key

Run the terminal commands inside Cloud Labs.

3. Cloud Labs login

Open cookiesensei.com and click **Open Labs** at the top right, or go directly to code.cookiesensei.com.

Log in using **GitHub**. Use the same email/account you registered with CookieSensei. Create a new workspace and select the template shown by the trainer, such as the **Next.js template** for web development.

4. Open the terminal

Open the browser editor in your lab. Click the warning/error indicator at the bottom left to open the lower panel, then select **Terminal**. You can also use **Terminal → New Terminal**.

You should now see a command prompt. Try **pwd** and **ls** to see where you are and what is in the folder.

5. Generate an SSH key for GitHub

Open the GitHub SSH key generation guide and choose Linux. Use this command:

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Replace the email inside the quotes with the email used for CookieSensei registration. Press **Enter** to accept the default key location, then choose and confirm a passphrase. If a key already exists, ask the trainer before overwriting it.

6. Add the public key to GitHub

```
cat ~/.ssh/id_ed25519.pub
```

Copy the entire public-key line, starting with **ssh-ed25519** and ending with your email. Open github.com/settings/keys and click **New SSH key**. Give it any recognizable name, choose **Authentication Key**, paste the line and add the key.

COPY THE PUBLIC KEY

Copy only the file ending in **.pub**. Keep the private key private.

Test the connection and create the app

Use the same Cloud Labs terminal for the following steps.

7. Test your connection

```
ssh -T git@github.com
```

On the first connection, you may see a message like this:

```
The authenticity of host 'github.com (IP ADDRESS)'
can't be established.
ED25519 key fingerprint is
SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
Are you sure you want to continue connecting (yes/no)?
```

Compare the fingerprint with GitHub's official fingerprints. If it matches, type **yes** and press Enter. Enter your key passphrase if requested.

A successful connection looks like this:

```
Hi USERNAME! You've successfully authenticated, but GitHub does not
provide shell access.
```

The username should be yours. The message about shell access is normal.

8. Create a Next.js project

Open the Next.js installation guide, select **npm** and copy the three commands below. Start in a lab folder that does not already contain a my-app folder or another Git project.

```
npx create-next-app@latest my-app --yes
cd my-app
npm run dev
```

Run one command at a time. If npm asks to install create-next-app, type **y**. Wait for project creation to finish before running the next command.

Open the **port 3000 preview** in Cloud Labs, as shown by the trainer. You should see the starter website. Keep the project folder **my-app** open in the editor.

The recommended defaults create a TypeScript app with app/page.tsx. If saved preferences give a different setup, ask the trainer to select the recommended defaults.

GITHUB REPOSITORY + FIRST PUSH

Send your Next.js project to GitHub

Run all Git commands from inside my-app, where package.json is located.

9. Create a new GitHub repository

Open github.com/new. Choose your own account as owner and enter any repository name. This guide uses **new_git_repo**. Choose its visibility, leave README, .gitignore and license initialization unselected, then click **Create repository**.

Under **Quick setup - if you have done this kind of thing before**, select **SSH**. Your URL will look like:

```
git@github.com:USERNAME/new_git_repo.git
```

For the workshop owner, the example is **git@github.com:cookieSensei/new_git_repo.git**. Students must use their own repository URL.

10. Run the repository commands

If npm run dev is still running, click the terminal and press **Ctrl+C** to return to the prompt. Stay inside **my-app**. Under GitHub's command-line setup instructions, use this sequence, replacing the name, email and USERNAME:

```
echo "# new_git_repo" >> README.md
git init
git config user.name "Your Name"
git config user.email "your_email@example.com"
git add README.md
git add .
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:USERNAME/new_git_repo.git
git push -u origin main
```

git add . includes the Next.js files as well as the README, so Vercel receives the complete website. Keep the generated .gitignore file; it keeps node_modules and build files out of Git. The name and email identify your commits.

FIRST PUSH COMPLETE

Refresh the GitHub repository page. You should see your app files, including **package.json**, the **app** folder and README.md. Next, deploy this repository.

FIRST DEPLOYMENT + PAGE UPDATE

Deploy, then design the restaurant page

Your first deployment shows the Next.js starter. The next push will update it.

11. Vercel deployment

Open vercel.com and log in using GitHub. Choose **Add New → Project** or **New Project**. Allow access to your repository if prompted, then import **new_git_repo**.

Check **Framework Preset: Next.js**. Keep **Root Directory** at the repository root (./), where package.json is located. Leave the default build settings and click **Deploy**.

Wait until the deployment is ready, then open the provided **vercel.app** domain. Your first website is live. Keep this production website link for checking later updates.

12. Update page.tsx using ChatGPT

Open ChatGPT and send this prompt:

PROMPT TO COPY

I have created a new Next.js project in VS Code in CookieSensei Cloud Labs. I want to update its page.tsx to reflect some frontend changes. I want the website to show a restaurant page for a restaurant named cookie. Add relevant details like an order menu on your own for now. Return the complete code for page.tsx, using the packages already installed, so I can paste it into my project.

Copy the page.tsx code from the response. In Cloud Labs, open **my-app/app/page.tsx**, replace the existing contents and save the file. If your project uses a src folder, the file is **my-app/src/app/page.tsx**. Copy the code itself, without the surrounding Markdown backticks.

Run **npm run dev** from my-app to preview the restaurant page. If an error appears, share the error with ChatGPT or ask the trainer, then save the corrected code before continuing.

Stop the dev server and clear the terminal

Click the terminal, hold **Ctrl** and press **C** (**Ctrl+C**). Then run:

```
clear
```

You are still inside my-app. Clearing the terminal does not delete or save any files.

UPDATE GITHUB + LIVE WEBSITE

Push your restaurant page

Save the file, stage it, commit it, then push it to GitHub.

13. Update the GitHub repository

In the terminal, from inside **my-app**, run these commands one by one:

```
git add .
git commit -m "Update restaurant page"
git push -u origin main
```

git add . stages the saved changes. **git commit** records them. **git push** sends the new commit to GitHub. Running only git push will not send an edit that has not been committed.

Refresh the GitHub repository and confirm that **Update restaurant page** appears as the latest commit and page.tsx contains the new code.

14. See the updated live website

Vercel normally starts a new deployment when you push to **main**. Wait for the build to finish; this can take a few minutes. Open or refresh the production domain you saved earlier. The restaurant page should appear.

If the changes do not appear

1. Open your Vercel project and select **Deployments**. Check whether a deployment for the latest GitHub commit is building, ready or failed.

2. If no deployment started, choose **Create Deployment**, select or enter the **main** branch and deploy its latest commit.

3. If the latest deployment failed, open its build logs and ask the trainer to help fix the error. Save the fix, then repeat the add, commit and push commands.

4. If you need to retry an existing deployment of the latest commit, open its menu and choose **Redeploy**. Check that it uses the new commit; redeploying an older commit will keep the old page.

WORKSHOP COMPLETE

You used Linux commands, connected GitHub through SSH, created a Next.js project, deployed it with Vercel and published a restaurant-page update. Keep your GitHub repository link and your live website link.

Help: Vercel Git deployments and managing deployments.