DialedPPC Deployment Walkthrough

From Downloaded Project to Live Website

Starting point: You have `dialedppc-astro-project.tar.gz` downloaded. **End point:** Site live on the internet at a URL you can share.

This walkthrough assumes nothing. Every step, every screen, every command.

PHASE 1: Install the Tools (One-Time Setup — ~15 minutes)

These tools only need to be installed once. After this, you never repeat Phase 1.

Step 1: Install Node.js

Node.js is what runs Astro. You need it to build the site.

1. Open your browser and go to: **https://nodejs.org**
2. You'll see two big green download buttons. Click the one labeled **"LTS"** (left side)
    - LTS = Long Term Support. It's the stable version. That's what you want.
3. A `.pkg` file downloads. Open it.
4. Click through the installer (Next → Next → Agree → Install). Default settings are fine.
5. When it says "Installation Complete," you're done. Close the installer.

Verify it worked:

1. Open **Terminal** (press `Cmd + Space`, type "Terminal", press Enter)
2. Type this and press Enter:

```
node --version
```

3. You should see something like `v22.x.x`. The exact number doesn't matter — if you see a version number, it worked.

Leave Terminal open. You'll use it for the rest of this walkthrough.

Step 2: Check if Git is installed

Git is how your code gets from your computer to GitHub. Macs sometimes have it pre-installed.

1. In Terminal, type:

```
git --version
```

2. **If you see a version number** (like `git version 2.x.x`) → Git is installed. Skip to Step 3.
3. **If you see a popup asking to install "Command Line Developer Tools"** → Click **"Install"**. This takes a few minutes. When it's done, run `git --version` again to confirm.
4. **If neither happens**, install Git manually:
   - Go to: **https://git-scm.com/download/mac**
   - Download and run the installer
   - Default settings are fine

Step 3: Create a GitHub account

GitHub is where your code lives online. Netlify pulls from GitHub to build your site.

1. Go to: **https://github.com**
2. Click **"Sign up"**
3. Enter your email, create a password, choose a username
   - Username suggestion: `jasonruiz` or `dialedppc` or whatever you prefer
   - This username will be visible but doesn't affect your site URL
4. Complete the verification steps (puzzle, email confirmation)
5. When asked about preferences, you can skip the personalization — click "Skip this step" or similar
6. You now have a GitHub account. Stay logged in.

Step 4: Configure Git on your Mac

Tell Git who you are (this labels your changes — required before you can push code).

In Terminal, run these two commands one at a time (replace with your actual name and email):

```
git config --global user.name "Jason Ruiz"
```

```
git config --global user.email "your-email@example.com"
```

Use the same email you signed up to GitHub with.

No output means it worked. That's normal for config commands.

PHASE 2: Unpack and Test the Project (~5 minutes)

Step 5: Unpack the project

1. Find `dialedppc-astro-project.tar.gz` in your Downloads folder
2. Double-click it. macOS will extract it into a folder called `dialedppc`
3. Move this folder somewhere permanent. Suggestion: your home folder or a `Projects` folder
    - Example: drag it to your Desktop for now, or create `~/Projects/` and put it there

In Terminal, navigate to the project folder. If you put it on your Desktop:

```
cd ~/Desktop/dialedppc
```

Or if you put it somewhere else, adjust the path. The key is: you need to be INSIDE the `dialedppc` folder in Terminal.

Verify you're in the right place:

```
ls
```

You should see files like `package.json`, `astro.config.mjs`, `src/`, `netlify.toml`, etc. If you see those files, you're in the right spot.

Step 6: Install project dependencies

This downloads all the libraries Astro needs. Only needed once (and again if you ever delete the `node_modules` folder).

```
npm install
```

You'll see some output scrolling by. It might take 30-60 seconds. When you see your Terminal prompt again (the `$` or `%` symbol), it's done.

You may see some yellow "warn" messages. That's normal. As long as there's no red "error" at the end, you're fine.

Step 7: Test the site locally

```
npm run dev
```

You'll see output like:

```
🚀  astro  v5.x.x ready in XXXms

 ┃ Local    http://localhost:4321/
 ┃ Network  use --host to expose
```

1. Open your browser and go to: **http://localhost:4321/**
2. You should see the DialedPPC homepage — dark background, green accents, mixing board aesthetic
3. Click "Services" in the nav — services page should load
4. Click "FAQ" — FAQ page should load, accordions should open/close
5. Forms won't actually submit locally (that's a Netlify feature), but they should display correctly

If everything looks right: Go back to Terminal and press `Ctrl + C` to stop the local server.

If something looks wrong: Stop here and tell me what you see. We'll fix it before deploying.

PHASE 3: Push to GitHub (~10 minutes)

Step 8: Create a GitHub repository

1. Go to **https://github.com** (make sure you're logged in)
2. Click the "**+**" icon in the top-right corner → "**New repository**"
3. Fill in:
   - **Repository name:** `dialedppc`
   - **Description:** (optional) `DialedPPC website`
   - **Visibility: Private** (your code doesn't need to be public. Netlify can access private repos.)

- **DO NOT** check "Add a README file"
- **DO NOT** check "Add .gitignore"
- **DO NOT** choose a license
- Leave everything else as default

4. Click **"Create repository"**

You'll see a page with setup instructions. **Don't close this page** — you'll need the URL shown. It will look something like:

```
https://github.com/yourusername/dialedppc.git
```

Step 9: Connect your local project to GitHub

Back in Terminal (make sure you're still in the `dialedppc` folder), run these commands **one at a time**, in order:

Initialize Git in your project:

```
git init
```

Add all your files:

```
git add .
```

Create your first commit (a "save point"):

```
git commit -m "Initial commit: DialedPPC Astro site"
```

Set the branch name:

```
git branch -M main
```

Connect to your GitHub repository (replace `yourusername` with your actual GitHub username):

```
git remote add origin https://github.com/yourusername/dialedppc.git
```

Push your code to GitHub:

```
git push -u origin main
```

First time pushing: GitHub will ask you to authenticate. You'll get one of these:

- A browser popup asking you to authorize Git — click "Authorize"
- A prompt for username/password in Terminal — enter your GitHub username, and for password, you'll need a **Personal Access Token** (see Troubleshooting below)

Once the push completes, you'll see output mentioning `main -> main`.

Verify: Go back to your GitHub repo page in the browser and refresh. You should see all your project files listed.

PHASE 4: Deploy on Netlify (~5 minutes)

Step 10: Create a Netlify account

1. Go to: **https://app.netlify.com/signup**
2. Click **"Sign up with GitHub"** (this is the easiest path — links the accounts)
3. Authorize Netlify to access your GitHub account when prompted
4. You're now logged into Netlify

Step 11: Deploy the site

1. In Netlify, click **"Add new site"** → **"Import an existing project"**
2. Click **"Deploy with GitHub"**
3. You may be asked to authorize Netlify again — click through
4. You'll see a list of your repositories. Click **"dialedppc"**
5. Configure the build settings (most should auto-detect from `netlify.toml`):
   - **Branch to deploy:** `main`
   - **Build command:** `npm run build`
   - **Publish directory:** `dist`
6. Click **"Deploy dialedppc"**

Netlify will now build your site. You'll see a build log scrolling. This takes 1-2 minutes.

When it's done, you'll see a green "Published" status and a URL like:

```
https://random-name-12345.netlify.app
```

Click that URL. Your site is live on the internet.

Step 12: Set up your custom domain (later)

The random Netlify URL works but isn't what you want long-term. When you're ready to point `dialedppc.com` at this site:

1. In Netlify: **Site settings → Domain management → Add custom domain**
2. Enter `dialedppc.com`
3. Netlify will give you DNS instructions (nameservers to point your domain registrar to)
4. Update your domain registrar settings (wherever you bought `dialedppc.com`)

We can walk through this separately when you're ready. The site works on the Netlify URL in the meantime.

TROUBLESHOOTING

"Authentication failed" when pushing to GitHub

GitHub no longer accepts passwords for Git operations. You need a Personal Access Token:

1. Go to: **https://github.com/settings/tokens**
2. Click **"Generate new token" → "Generate new token (classic)"**
3. Give it a name like "My Mac"
4. Set expiration to **90 days** (you can regenerate later)
5. Check the box next to **"repo"** (full control of private repositories)
6. Click **"Generate token"**
7. **Copy the token immediately** — you won't see it again
8. When Terminal asks for your password, **paste the token** instead of your actual password

"Command not found: node" after installing

Close Terminal completely and reopen it. The PATH needs to refresh.

Site deploys but pages look broken

If the homepage works but Services/FAQ show 404:

- Check that your Netlify build log shows "3 page(s) built"
- If not, the build might have errored — copy the build log and bring it back here

Accordions don't open/close on FAQ page

The JavaScript is inline in the page. If it's not working, it's likely a build issue. Check the browser console (right-click → Inspect → Console tab) for red errors and bring those back here.

MAKING CHANGES LATER

When you want to update the site after it's deployed:

1. Open Terminal, navigate to the project folder:

```
cd ~/Desktop/dialedppc
```

(or wherever you put it)

2. Make your changes (edit files, or bring updated files from Claude)
3. Push the changes:

```
git add .
git commit -m "Description of what changed"
git push
```

4. Netlify automatically detects the push and rebuilds. New version live in ~1-2 minutes.

That's it. Three commands to update the live site.

WHAT YOU JUST SET UP

- **Node.js** — runs Astro on your Mac
- **Git** — tracks changes and pushes code
- **GitHub** — stores your code online (private)
- **Netlify** — builds and hosts your site (free tier)

- **Auto-deploy** — push code → site updates automatically

This is the same stack used by most modern web projects. The initial setup is the hard part. From here forward, updating the site is three commands.