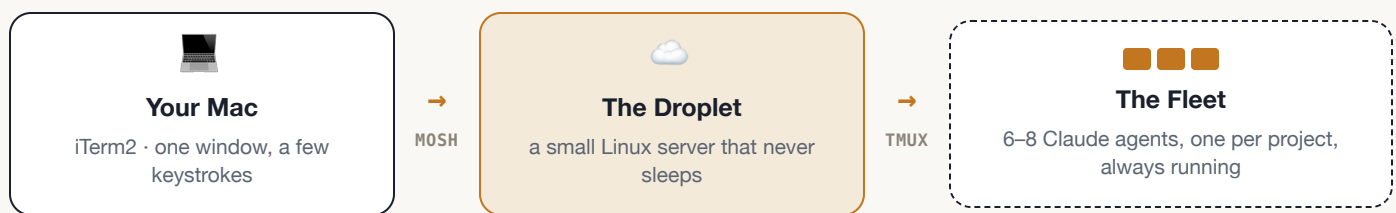


# Running a Fleet of AI Agents From One Terminal Window

How to set up **Claude Code** on your Mac, park an always-on **server** in the cloud, and drive a whole team of AI coding agents from a single window. The exact setup I use every day.



*Your laptop is just the remote control. The work happens on the server, and it keeps happening after you close the lid.*

Hey, my friend Mark asked me to write this up for you. He runs his whole operation this way and thought you'd get a lot out of it. It's meant to be read start to finish once, then kept as a reference. Nothing here needs deep technical background; every command is spelled out.

00

## The 30-second version

what this actually is

Three moving pieces, and that's the whole trick:

- ▶ **Claude Code** is an AI coding agent that lives in your terminal. You talk to it in plain English ("build me a screening tool," "fix this bug"), and it reads files, writes code, runs commands, and tests its own work. A genuine collaborator, not autocomplete.
- ▶ **The Droplet** is a cheap Linux computer you rent in the cloud that is *always on*. Instead of running the agents on your laptop, you run them there. Close your laptop, lose your wifi, get on a plane, and the agents keep working.
- ▶ **The Fleet** is the payoff: because the server never sleeps, you can keep *many* agents alive at once (one per project) and reconnect to all of them instantly with a single command from anywhere.

The rest of this guide sets those up in order: first Claude Code on your Mac, then the server, then how to tie it all together so it's one keystroke to boot the whole thing.

## 01 iTerm2 + Claude Code on your Mac

the local setup · ~15 min

Start here even if you never touch a server. This alone is worth it. Two things to install, then you're running.

### First, a better terminal

macOS ships with a terminal app, but **iTerm2** is the free upgrade everyone uses: better tabs, colors, search, and split panes. Download it from [iterm2.com](https://iterm2.com) and drag it to Applications. That's the whole install.

### Then, the tools Claude Code needs

Claude Code runs on **Node.js** (a common piece of developer plumbing). The cleanest way to get it is **Homebrew**, the standard Mac package manager. Paste these into iTerm2 one at a time:

```
iTerm2 · your Mac

# 1. install Homebrew (skip if you already have it)
$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# 2. install Node.js
$ brew install node

# 3. install Claude Code itself
$ npm install -g @anthropic-ai/claude-code
```

### Run it

Go into any project folder and type one word:

```
iTerm2 · your Mac

$ cd ~/my-project
$ claude
# the first time, it opens a browser to log in with your
# Claude subscription, then drops you into a chat. That's it.
> build me a simple webpage that shows today's weather
```

#### THE ONE MENTAL MODEL THAT MATTERS

Claude Code isn't a chatbot you copy-paste out of. It's *in* your folder. It can see your files, change them, run your code, and check whether it worked, all with your permission. You describe the goal; it does the mechanical work.

## 02 The Droplet: why a cloud server changes everything

the always-on machine

"Droplet" is just DigitalOcean's cute name for a small cloud server. Any provider works (DigitalOcean, Hetzner, AWS Lightsail). It's a Linux computer you rent by

the month and reach over the internet. Here's why bothering with one is a genuine unlock.

### The problem it solves

- ▶ **Your laptop is a bad host for long work.** Close the lid and the agent stops. Walk to a meeting and it's paused. Lose wifi and it disconnects. Anything that takes an hour is at the mercy of your day.
- ▶ **A server doesn't have a lid.** It runs 24/7 in a data center. An agent you start there keeps building whether you're watching or not, whether your Mac is open or shut, awake or asleep.
- ▶ **It's the same everywhere.** Your Mac, a borrowed laptop, your phone on the train all connect to the *same* server and see the *same* running agents. The work lives in one place; your devices are just windows onto it.

### What I actually do with mine

My server holds all my projects and runs a small army of Claude agents against them around the clock. I kick off a long build ("screen these 400 land parcels and score them"), close my laptop, and check back later from my phone. The server did the work while I was gone. For anything bigger than a quick edit, that's the whole game.

#### ONE HONEST CATCH: SIZE IT RIGHT

A too-small server *freezes* when you pile many heavy agents onto it (I learned this the hard way on a 1-CPU box with no swap memory). If you're going to run a real fleet, get at least 2 CPUs and 8 GB of memory, and add "swap" (a safety valve for memory) on day one. Budget roughly **\$12–48/month** depending on size. A small starter box is ~\$6/mo.

## 03 Would this be useful for you?

an honest read

Claude Code on your Mac (Part 1): almost certainly yes, if you build anything. The *server* + *fleet* layer is worth it under specific conditions. Here's the honest split so you don't over-build.

#### ● The server is worth it if...

- ✓ You run **long jobs** (builds, scrapes, data crunching) that outlast a single sitting.
- ✓ You juggle **several projects** and want each one's agent alive and ready, not restarted every time.
- ✓ You want to start something and **walk away**: close the laptop, check from your phone.
- ✓ You hop between **devices/locations** and want one consistent workspace.
- ✓ You like the idea of **agents working while you sleep**.

#### ● Skip it (for now) if...

- ✗ You mostly do **quick, one-off edits** that finish in a few minutes.
- ✗ You work on **one project at a time** and always at your desk.
- ✗ You'd rather not manage a server, pay a monthly fee, or learn two new tools yet.
- ✗ Your work is **sensitive** in a way that makes a rented cloud box a non-starter.

Reasonable path: **start with Part 1 only**. Live on Claude Code locally for a couple of weeks. The day you think "ugh, I wish this had kept running while I was out" is your signal to add the server. Don't build the fleet before you feel the pain it solves.

This is the piece people don't expect. Two small tools, **mosh** and **tmux**, turn "log into a server" into "press one key and all my agents are back." Here's how each works, in plain terms.

### mosh

A tougher replacement for the standard **ssh** connection. Regular ssh drops the moment your wifi hiccups or your laptop sleeps. **mosh survives it**: change networks, shut the lid, reopen hours later, and your session is simply *there*, no reconnecting.

### tmux

Keeps your work *alive on the server* even when nothing is connected. It gives you **tabs that persist**: one tab per project, each running its own agent. Disconnect entirely and every tab keeps running, right where you left it.

Put them together and you get what I call the **fleet**: a set of persistent tabs on the server ( **pigeon** , **dra** , **outreach** , one per project), each with its own Claude agent that auto-resumes its last conversation. I don't rebuild it each morning. I wrap the whole thing in a single shortcut:

```
iTerm2 · one command boots everything

$ csf
# ↑ my alias. behind the scenes it just runs:
#   mosh your-user@your-droplet-ip -- tmux attach -t fleet

# and instantly I'm looking at all my tabs, every agent
# still running exactly where I left it:
[ pigeon ] [ dra ] [ outreach ] [ ca-consumer ] [ misc ]
```

## How the pieces line up

- 1 You type one short command**  
on your Mac ( **csf** , or whatever you name it).
- 2 mosh connects**  
to the server, and stays connected through sleep, wifi changes, and travel.
- 3 tmux hands you the fleet**  
every project's tab, every agent, exactly as you left it.
- 4 You jump between tabs**  
with a keystroke, giving one agent a new task while three others keep grinding on their own.
- 5 You close the laptop.**  
Nothing stops. Reopen tonight, type **csf** , and you're right back in.

### THE FLEET IS JUST A LIST YOU CONTROL

Which tabs exist is defined by a plain text file (mine is a one-line-per-project list). Add a project, remove one, rename them: it's editing a text file, then rebuilding the session. Your friendly local Claude Code can set this whole thing up for you; you don't have to hand-write it.

**1 Do Part 1.**

Get iTerm2 + Claude Code running on your Mac. Use it for real work for a week or two. This is 90% of the value on its own.

**2 Rent a small server**

when you feel the "wish it kept running" itch. DigitalOcean is the friendliest start; pick at least 2 CPU / 8 GB and turn on backups. Add swap immediately.

**3 Install mosh + tmux + Claude Code on the server**

literally ask your local Claude Code to walk you through it, or to write the setup for you. It's a handful of commands.

**4 Set up your one-key launcher**

(the `csf`-style alias) and a small project list for the fleet. Again: hand this to Claude Code and let it build the scripts.

**5 Live in it.**

After a week you'll wonder how you worked any other way.

**YOU DON'T HAVE TO DO THE PLUMBING YOURSELF**

The genuinely nice irony: the AI agent *is* the sysadmin. Once you have Claude Code running locally, you can tell it "help me set up a DigitalOcean server and a persistent tmux fleet I can reconnect to," and it will hand you each step. That's exactly how this setup gets built and maintained.