POWERLOOM :: HOME EDITION

Powerloom on your laptop.
In under two minutes.

A self-contained single-user control plane. Alfred provisioned on first boot. Claude Code wired to drive it. No AWS, no multi-tenant auth, no invite codes -- just docker, an Anthropic key, and a terminal.

This guide walks the clone-to-first-turn path. Five steps. The webpage version lives at powerloom.org/docs/home-setup.

What you'll get

One `docker compose up` brings these into existence on your laptop:

- Postgres 16 + pgvector -- state store and vector memory.
- Powerloom API -- control plane. Creates org, user, and Alfred on first boot. Mints a 365-day Personal Access Token to a host-visible file.
- Alfred -- Powerloom's meta-agent, running on claude-opus-4-6. Drives provisioning via natural language.
- Docker-local MCP spawner -- Alfred can deploy MCP server containers (echo, files, postgres, slack) on your machine without Terraform or AWS.

Prerequisites

- Docker Desktop (or Linux docker + compose v2).
- Python 3.11+ on the host -- for scratch/home_init.py. Stdlib only.
- Claude Code -- anthropic.com/claude-code.
- An Anthropic API key -- console.anthropic.com. Alfred runs on claude-opus-4-6; expect cents per conversation.
- pip install mcp httpx in whichever Python Claude Code invokes.

Step 1 -- Clone and configure

Clone the repo, copy the example env file.

```
git clone https://github.com/shanerlevy-debug/Powerloom.git
cd Powerloom
cp .env.home.example .env.home
```

Edit .env.home. Set these two lines:

```
POWERLOOM_HOME_EMAIL=you@example.com
ANTHROPIC_API_KEY=sk-ant-api03-...
```

The compose file refuses to start without POWERLOOM_HOME_EMAIL; a typo surfaces immediately rather than silently provisioning the wrong user.

Step 2 -- Start the stack

```
docker compose -f docker-compose.home.yml \
    --env-file .env.home \
    up -d --build
```

Two containers come up: postgres and api. No UI, no pgadmin, no multiplexer -- the home stack is lean on purpose.

Tail the bootstrap log to watch first-boot provisioning:

```
docker compose -f docker-compose.home.yml \
    --env-file .env.home logs -f api \
    | grep -E 'home_bootstrap|alfred'
```

Expected lines, in order:

```
home_bootstrap.tenant_created org=... user=... email=you@example.com
alfred.bootstrap_for_org organization=... agent_created=True
home_bootstrap.artifacts_written pat_path=/app/home_data/pat.txt
home_bootstrap.done
```

Ctrl-C out. Your stack is up.

Step 3 -- Generate the Claude Code config

```
python scratch/home_init.py --write
```

Materializes two files under ./home_data/:

- .mcp.json -- the MCP client config block Claude Code reads to connect.
- cc-prompt.md -- first-message prompt with your user/org/Alfred IDs baked in.

Stash the Anthropic key into the runtime_credentials store so every CMA agent shares it:

```
python scratch/home_init.py --byok-anthropic sk-ant-api03-...
```

Step 4 -- Wire Claude Code

Copy the contents of ./home_data/.mcp.json into your Claude Code MCP config. Project-scoped: <project>/.mcp.json. User-wide: ~/.claude/mcp.json.

```
{
  "mcpServers": {
    "powerloom-home": {
      "command": "python",
      "args": ["/abs/path/Powerloom/mcp_clients/powerloom_mcp_server.py"],
      "env": {
        "POWERLOOM_PAT": "pat_...",
        "POWERLOOM_API_BASE_URL": "http://localhost:8000"
      }
    }
  }
}
```

If CC can't find the server, run it manually: python <path>/powerloom_mcp_server.py. ModuleNotFound errors -> pip install mcp httpx.

Step 5 -- Meet Alfred

Open a new Claude Code session. First message is the content of ./home_data/cc-prompt.md, edited to replace the placeholder in 'What I want to do:' with your actual goal.

Examples that work well as a first turn:

. "Deploy an echo MCP and invoke it to confirm the loop works."
. "Deploy a files MCP pointed at /data/mcp_files and summarize whatever I dropped there."
. "Create a project called 'ops-weekly' with threads for {a, b, c}. Hold them at priority=high."

CC calls powerloom_whoami and powerloom_list_agents first, then hands off to Alfred via tool calls. Alfred's voice is direct and terse -- no chattiness, no emoji, no filler. It's a feature.

MCP templates Alfred can deploy

Template	What it does	Config you provide
echo	Reference / sanity-check	none
files	Read/search files under a host-mounted directory	local_path: "/data/mcp_files"
postgres	Query a Postgres instance	dsn: "postgresql://..."
slack	Read/write Slack via a bot token	bot_token: "xoxb-..."

SaaS templates (github, notion, jira, linear, google_drive, etc.) aren't wired to local images yet. Use the SaaS edition at api.powerloom.org or watch the roadmap -- SaaS image packaging is an opportunistic follow-up.

Everyday commands

```
# Status
docker compose -f docker-compose.home.yml --env-file .env.home ps

# Tail logs
docker compose -f docker-compose.home.yml --env-file .env.home logs -f api

# psql shell
docker compose -f docker-compose.home.yml --env-file .env.home \
    exec postgres psql -U powerloom -d powerloom

# Rotate PAT
python scratch/home_init.py --rotate --write

# Stop (data persists)
docker compose -f docker-compose.home.yml --env-file .env.home down

# Start fresh (wipes state)
rm -rf ./home_data
docker compose -f docker-compose.home.yml --env-file .env.home up -d --build
```

Security notes

- Localhost-only by default. Postgres and the API port-map only to localhost. Exposing on a LAN requires rotating the internal service token and generating a real credential master key.
- Treat ./home_data/ like ~/.aws/credentials. The PAT is reveal-once in the DB but plain-text on disk.
- The docker socket mount is root-equivalent on the host -- acceptable for single-user localhost, explicitly NOT done in prod.
- Anthropic key lives in .env.home. Every Alfred invocation spends against it.

Troubleshooting

Symptom	Fix
pg_isready hangs	Port 6433 busy on host. Check `lsof -i :6433`.
home_bootstrap.failed in logs	POWERLOOM_HOME_EMAIL missing/malformed in .env.home.
curl /me returns 401	Stale PAT. Re-run `python scratch/home_init.py` or `--rotate`.
Alfred 'no runtime credential'	Run `python scratch/home_init.py --byok-anthropic sk-ant-...`
CC can't see powerloom-home tools	.mcp.json path must be absolute and exist. pip install mcp httpx.
DB corrupted after power loss	rm -rf ./home_data/postgres -- stack re-bootstraps on next up.

Related

- Webpage version of this guide: https://powerloom.org/docs/home-setup
- Full documentation: https://powerloom.org/docs
- Repo: https://github.com/shanerlevy-debug/Powerloom