draupnir — Idiot Guide

version 0.1.1

generated 2026-07-11

---

rendered by **nornir** · typst engine

Contents

draupnir — for idiots

The five-minute version. No Norse degree required.

What is it?

Draupnir is a small Rust library that **turns things on**. You hand it a short recipe — a `BootSpec` — and it starts the machine that recipe describes:

- a **virtual machine** (KVM), or
- a **container** (like a redis server), or
- a **real physical server** in a datacenter (bare metal).

That's it. It starts the instance and lets you turn it on, off, and check whether it's running.

The name

Draupnir is Odin's magic gold ring. Every ninth night it drips **eight identical copies** of itself. So the name fits a tool that can stamp out **a whole fleet of identical machines** from one image — but don't overthink it: mostly Draupnir just boots one thing at a time.

The three ways it boots something

1. **KVM** — a virtual machine. Draupnir doesn't build the VM machinery itself; it asks its friend **tunnr** to do the actual QEMU/KVM boot.
2. **Container** — an OCI container (e.g. redis). Draupnir asks a container runtime to pull the image and start it.
3. **Redfish** — this is the cool one. Every serious server has a tiny second computer inside it (the **BMC**) that's on even when the server is "off". Draupnir talks to that little computer over the network and says: *"here's an installation CD (an ISO), pretend it's in the drive, boot from it once, now power on."* No operating system on the server needed — you can install one from scratch, remotely.

How you'd use it

```
use draupnir::{BootSpec, Boot, container::ContainerBoot};

let spec = BootSpec::container("cache", "docker.io/library/redis:7");
let machine = ContainerBoot::new().boot(&spec)?;   // redis is now running
```

Swap `ContainerBoot` for `KvmBoot` or `RedfishBoot` and change the recipe — same shape every time.

Who uses Draupnir?

Two bigger tools lean on it so they don't each rewrite "how to boot a machine":

- **jera** — decides *what* to run as a job. When the job is a VM or container, it asks Draupnir to boot it.
- **Skidbladnir** — installs and runs software as a service across an airgap. When it needs an instance, it asks Draupnir.

Draupnir itself is dumb-simple on purpose: it just boots things. Everything clever (jobs, services, orchestration) lives above it.

The honesty rule

The scaffold is real where it says it is (the recipe types, the validation, the fleet fan-out — all tested) and **honestly unfinished** everywhere else: the not-yet-written backend guts say `todo!("draupnir: …")` out loud instead of faking success. Read `guide.md` next for the real API.