Gigabits Ops Agent — Usage Guide

Gigabits Ops Agent is a Linux operations dashboard you open in a web browser. It lets you monitor a server, manage its applications and services, install and update software, read system logs, and get AI-assisted explanations — all from one screen, without memorizing terminal commands.

This guide covers **how to use the dashboard** once it's installed and running on your server. It's written so you can start on your own, even if you are new to Linux. Follow the steps in order.

Table of contents

Before you start

You only need a few things to use the dashboard.

Requirement	Why	How to check
A web browser on your computer	To open the dashboard	Any modern browser (Chrome, Firefox, Edge, Safari)
Network access to the server on port `8787`	The dashboard is served on this port	You'll confirm this when the page loads in Step 1
Administrator (root) access to the server	Needed once, to get an access token	Run `sudo whoami` on the server — it should print `root`

> **New to Linux?** A few terms in this guide (root, sudo, token, service) are explained in plain language in the Glossary. You do not need to understand them to follow the steps.

Quick start (2 steps)

If you just want to get going, here is the whole process. Each step is explained in detail below.

1. **Open the dashboard** in your browser (see Step 1).
    - Get the server's Public IPv4 address (run `curl ifconfig.me`, or copy it from the EC2 console).
    - Go to `http://<Public IPv4 address>:8787`.
2. **Get an access token** so you can use the management features (see Step 2).
    - In the dashboard, pick a token duration and click **Get Token Command**.
    - Run the command it shows you in a terminal on the server.
    - Paste the token back into the dashboard and click **Apply Token**.

That's it. You're ready to use every feature.

What the product does

Gigabits Ops Agent gives you one browser view of a Linux server. Underneath, it uses the same standard tools an administrator would type by hand (`apt`, `dnf`, `yum`, `systemctl`, `journalctl`), so you can always verify what it did.

You can:

- **See system status** — hostname, Linux version, CPU, memory, disk, load, and uptime.
- **See running applications** — it automatically detects common software like Nginx, Apache, MySQL/MariaDB, PostgreSQL, Redis, Docker, MongoDB, Node.js, and more, and shows whether each is running.
- **Manage services** — start, stop, restart, or reload services.
- **Manage packages** — search for, install, upgrade, and uninstall software, and see how many updates (including security updates) are available.
- **Review package history** — see what was recently installed or changed.
- **Read logs** — view system, kernel, authentication, service, and package logs for a chosen time range.
- **Get AI help** — ask read-only questions or have the AI explain a log excerpt in plain language (optional feature).

What it is not: It is an administration tool, not a replacement for your normal tools or your organization's change-management process. It never runs commands on its own from the AI; you stay in control.

Step 1: Open the dashboard

1. **Get the server's IP address.** Use either method:
   - Run this command in a terminal on the server:

     ```
     curl ifconfig.me
     ```

   - Or open the **Instance summary** page in the EC2 console and copy the **Public IPv4 address**.
2. **Open your browser** and append `:8787` to that address, in this format:

   ```
   http://<Public IPv4 address>:8787
   ```

For example, if the address is `203.0.113.25`, go to `http://203.0.113.25:8787`.

Expected result: The dashboard loads and shows the system status page (hostname, CPU, memory, and so on).

> The dashboard uses `http`, not `https`. Only use it on a trusted internal network. See Security and safe use.

If the page does not load, see Troubleshooting.

Step 2: Get an access token

Viewing status is open, but any **management action** (installing packages, restarting services, reading logs, using AI) requires a temporary access token. This is a safety feature: it proves that whoever is clicking buttons in the browser also has administrator access on the server itself.

Follow these steps the first time:

1. In the dashboard, find the token section and **select a duration** (for example, 30 minutes). Tokens last up to 24 hours.
2. Click **Get Token Command**. The dashboard shows you a command to copy.
3. **Paste and run that command** in a terminal on the server, logged in as root (or using `sudo`). It prints a token — a long string of characters.
4. **Copy the token**, paste it into the dashboard's token box, and click **Apply Token**.
5. Done. You can now use every feature until the token expires.

Why the round trip? The dashboard cannot grant itself access. By making you run a command on the server, it confirms you are a real administrator of that machine, not just someone who found the web page.

When the token expires, simply repeat these steps to get a new one.

> Tokens are temporary and are kept only in the agent's memory. Your browser also remembers the token for convenience so you don't have to paste it every time.

Using the features

Once your token is applied, these sections are available in the dashboard.

System overview

See hostname, distribution, kernel, CPU count, uptime, and live CPU, memory, load, disk, and network usage.

Applications

Automatically detects common software (Nginx, Apache, MySQL/MariaDB, PostgreSQL, Redis, Docker, PHP-FPM, MongoDB, Node.js, RabbitMQ, and others) and shows whether each is running or stopped, its version, and its ports.

Services

Lists services and their state (active, failed, stopped, enabled, disabled, masked). You can start, stop, restart, and reload services where supported.

Packages

Search enabled software repositories, install, upgrade, and uninstall packages, and see counts of available updates including security updates. A history view shows recent package changes.

OS Diagnostics (logs)

Read system logs safely. You choose:

- **A log source** — for example all system events, only errors, kernel messages, authentication events, package history, or a specific service's log.
- **A time range** — 5 minutes, 15 minutes, 1 hour, 6 hours, 24 hours, since boot, or a custom start/end.
- **A line limit** — 25, 50, 100, 200, 500, or custom.

Results are limited in size so the page stays fast. If a log source has no entries, the dashboard tells you clearly instead of showing something unrelated.

The AI assistant (optional)

The AI assistant is **optional** — the product works fully without it. When enabled, it gives read-only guidance:

- **Ask general questions** about the server in the main AI Assistant section.
- **Explain a log** — inside OS Diagnostics, click **Explain with AI** to get a plain-language explanation of only the log excerpt you're looking at. It does not see your other data.

The AI never runs commands. It may suggest commands, but you review and run them yourself. It uses a service called OpenRouter, and its settings are stored on the server at `/etc/gigabits-ops-agent/ai.env`.

> **Do not** paste passwords, API keys, private keys, tokens, or other secrets into AI questions or into logs you send to the AI.

To use AI, an OpenRouter API key must be configured. If AI features report an error, see the AI section in Troubleshooting.

Security and safe use

- Use the dashboard only on a **trusted administration network**.
- **Do not expose port `8787` to the public internet** without a firewall and additional network protection.
- All management actions require a valid, time-limited token.
- Reading logs requires a token, because logs can contain sensitive information.
- Log sources are limited to a safe, pre-approved list — arbitrary file paths are not allowed.
- Review logs before sending them to the AI, since AI uses an external provider.
- The OpenRouter API key is stored with restricted permissions and is not shown again after you save it.

Troubleshooting

Work through the section that matches your problem.

The dashboard won't load in the browser

1. Confirm the service is running on the server: `systemctl is-active gigabits-ops-agent` should print `active`.
2. Confirm it's listening on the port: `sudo ss -tulpn | grep 8787` should show a line for port 8787.
3. Confirm you used the right address (`localhost` on the server, or the correct `<server-ip>` from another computer).
4. Confirm a firewall isn't blocking port 8787 between your computer and the server.

Package search or install fails

Refresh the package lists first, then try again:

- Debian/Ubuntu: `sudo apt-get update`
- RHEL/Rocky/Alma with DNF: `sudo dnf makecache`
- Systems using YUM: `sudo yum makecache`

Repository configuration is managed by your operating system; if a package can't be found, it may not be in your enabled repositories.

Diagnostics (logs) show nothing or an error

- Confirm your token is still active (get a new one if it expired).
- Some systems store logs in the system journal and others in `/var/log` files — try a different log source.
- The account running the service may need permission to read certain logs (for example, kernel messages).
- Try a wider time range.

AI features report an error

- Confirm an OpenRouter API key has been configured.
- Confirm your token is still active.

- Confirm the server can reach the internet (OpenRouter).
- For the exact error, check the service log on the server:

```
sudo journalctl -u gigabits-ops-agent -n 100 --no-pager
```

Glossary

Plain-language definitions for terms used in this guide.

- **Root / sudo** — the administrator account on Linux. `sudo` runs a single command as the administrator. Reading system logs and getting a token require it.
- **Service** — a program that runs in the background (for example, a web server). Managed with `systemctl`.
- **systemd / systemctl** — the standard system that starts, stops, and monitors services on modern Linux.
- **Port** — a numbered channel a program listens on. This dashboard uses port `8787`.
- **Token** — a temporary password the dashboard needs before it will perform management actions. It expires automatically.
- **Repository** — an online source your system downloads software from.
- **Log / journal** — records of what the system and its programs have done, used for diagnosing problems.
- `<server-ip>` — a placeholder. Replace it with your server's real IP address (find it with `hostname -I`).
