GEBO.AI SETUP GUIDE · STEP 1 OF 7

Install Gebo.ai with Docker

Download the reference Docker Compose file and start Gebo.ai with the four databases it uses. Two commands, then a few minutes of downloading.

TIME	About 15 minutes, mostly the first download of the images (a few GB).
YOU NEED	A server with Docker Engine and Docker Compose v2 (Linux), or Docker Desktop on Windows 10/11 or macOS. About 10 GB of memory available to Docker.
NETWORK	Internet access to GitHub and Docker Hub. Port 12999 reachable from your users' PCs.

1 Prepare the server

Install Docker following docs.docker.com/engine/install (Linux) or Docker Desktop (Windows, macOS). Then check that Docker Compose v2 is there:

```
docker compose version
```

Docker Compose v2

The command is `docker compose`, with a space. The old `docker-compose` 1.x program is retired and does not read this file.

2 Download the compose file and start Gebo.ai

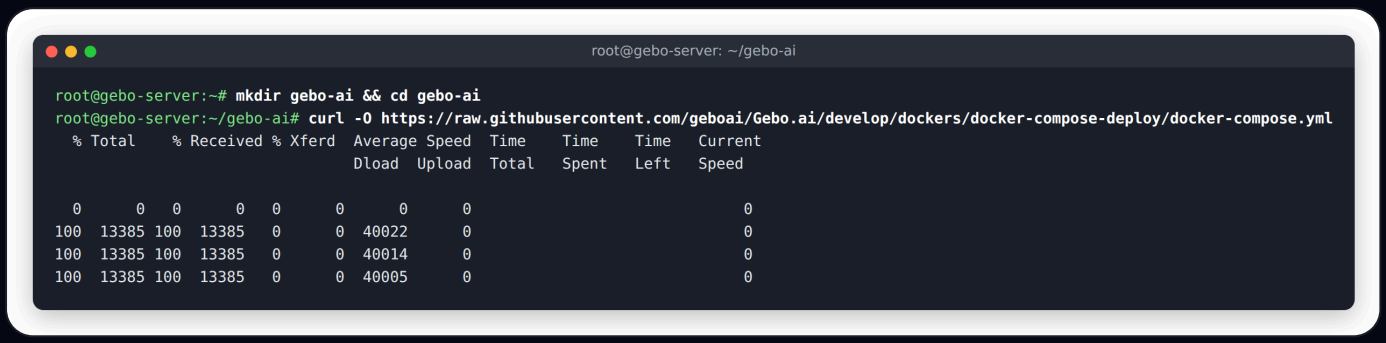
Create a folder for Gebo.ai, download the file into it and start it:

```
mkdir gebo-ai && cd gebo-ai
curl -O https://raw.githubusercontent.com/geboai/Gebo.ai/develop/dockers/docker-
```

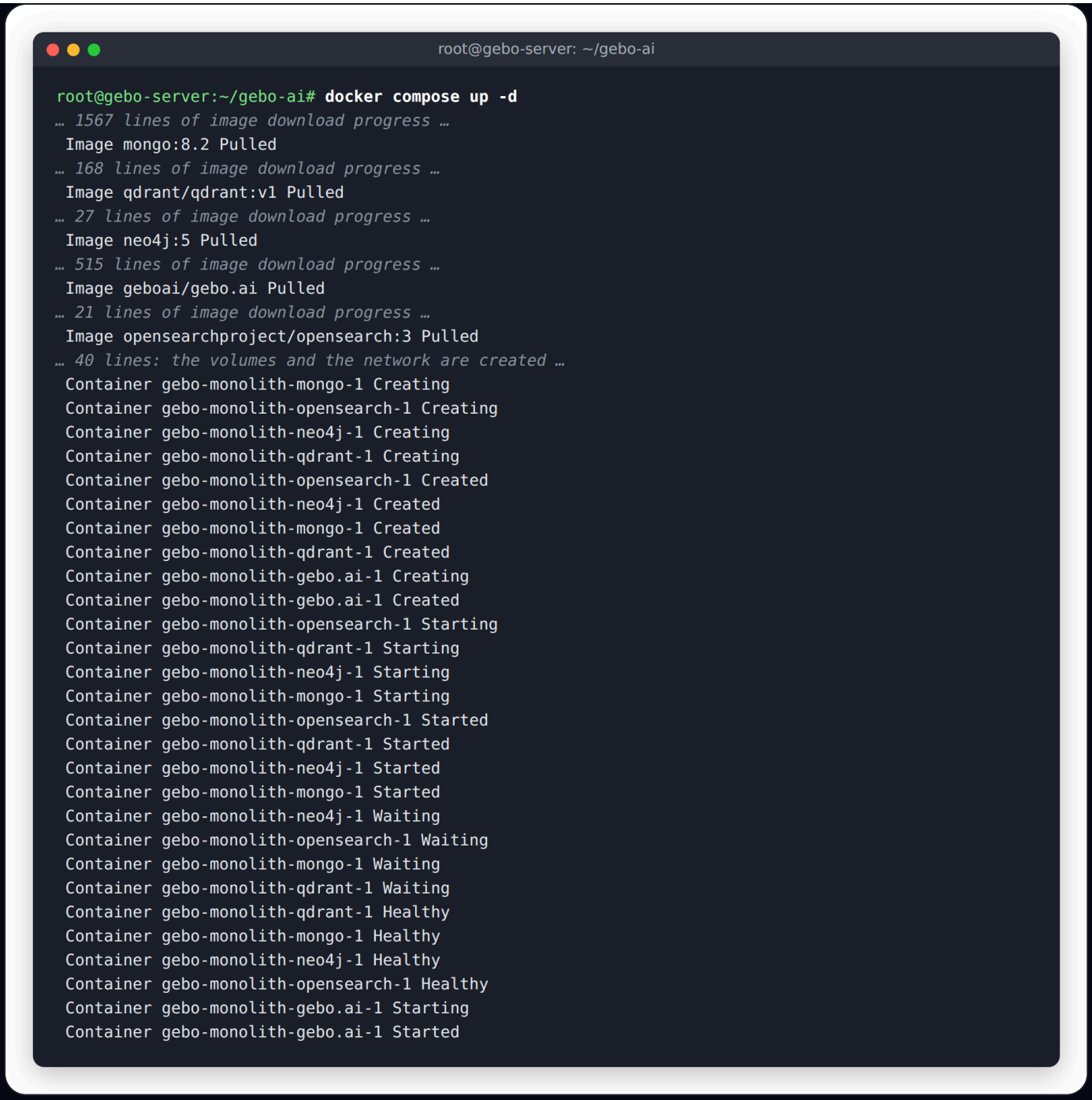
```
compose-deploy/docker-compose.yml
docker compose up -d
```

On Windows PowerShell write `curl.exe -O` instead of `curl -O`.

Or download it by hand: open the compose file on GitHub, click **Download raw file** and save it as `docker-compose.yml` in a folder named `gebo-ai`. Then, in that folder, run `docker compose up -d`.



The compose file is downloaded into the new gebo-ai folder.

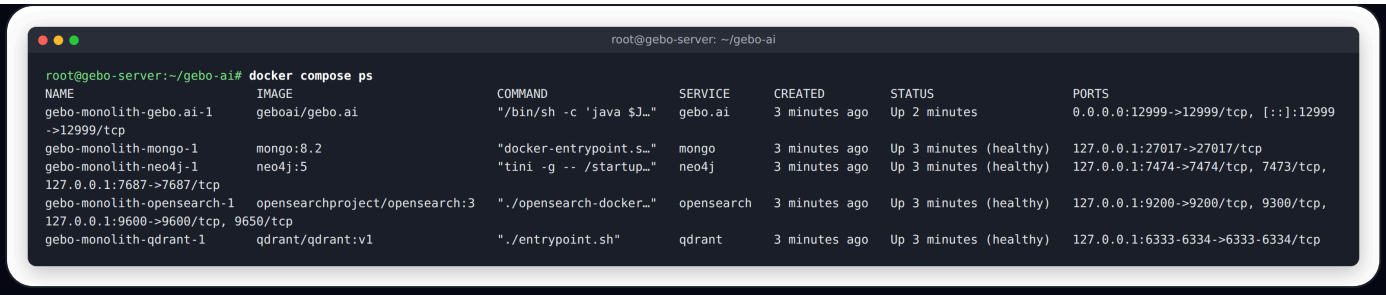


The first `docker compose up -d` downloads the images, creates the data volumes and starts the five containers. Gebo.ai starts last, when its four databases are healthy.

---

3 Check that everything runs

```
docker compose ps
```



Mongo, Qdrant, Neo4j and OpenSearch are healthy, gebo.ai is up.

The first start of Gebo.ai takes 1-3 minutes. Then open **http://<server address>:12999/** in a browser and continue with step 2.

---

4 Good to know

- **Data:** everything is kept in Docker volumes named `gebo-monolith_*`. They survive restarts and upgrades; only `docker compose down -v` deletes them. Back up `gebo-monolith_gebo-mongo` and `gebo-monolith_gebo-work` together.
- **Work directory:** the documents Gebo.ai mirrors, its caches, the uploads and the chat sessions live in `/opt/gebo.ai/work`, the `gebo-work` volume. **Setup → Platform → Gebo.ai work directory** shows it, already set. On a Linux package or Windows installation, choose it there once, before anything else: an empty folder the service can write to, such as `/var/gebo-ai/data`.
- **Upgrades:** in the gebo-ai folder, run `docker compose pull` and `docker compose up -d`.
- **Firewall:** the databases listen only on the server itself. Only Gebo.ai is reachable from the network, on port 12999: open it for your users, for example `sudo ufw allow 12999/tcp`.
- **HTTPS:** Gebo.ai answers in plain HTTP. To serve it over HTTPS follow Part A of the Gebo.ai Production Hardening Manual (`docs/GEBO-HARDENING-MANUAL.md` in the Gebo.ai repository).

---

5 Alternative: the ready-made virtual appliance

If you prefer a virtual machine to Docker, gebo.ai/downloads offers Ubuntu Server 24.04 with Gebo.ai already installed, as VMware/ESXi (VMDK), Hyper-V (VHDX) and Proxmox/KVM (qcow2) images.

1. Import the image in your hypervisor and start the virtual machine (Hyper-V: generation 1).
2. Change the passwords listed in the archive's `accounts.txt`, as the downloads page explains.
3. Open **http://<virtual machine address>:12999/** and continue with step 2.

Windows Server

Docker Desktop does not run on Windows Server: there, use the Hyper-V appliance, or a Linux virtual machine with Docker.

← Gebo.ai setup guide

Next: First access: create the administrator →