



Usage instructions:

1. Launch the product via 1-click. **Please wait until** the instance passes all status checks and is running. You can connect using your Amazon private key and 'ubuntu' login via your SSH client.

To update software, use: **sudo apt update && sudo apt -y upgrade**

2. Next change into the LibreChat directory and run:

cd ~/LibreChat

cp -n .env.example .env *# creates .env if missing, does not overwrite*

Generate strong secrets and write them into .env

SECRET="\$(openssl rand -hex 32)"

REFRESH="\$(openssl rand -hex 32)"

sed -i '/^JWT_SECRET=/d;/^JWT_REFRESH_SECRET=/d' .env

printf "JWT_SECRET=%s\nJWT_REFRESH_SECRET=%s\n" "\$SECRET" "\$REFRESH" >> .env

(Optional) Verify without revealing the values

awk -F= '/^JWT_SECRET|^JWT_REFRESH_SECRET/ {printf "%s length=%d\n",\$1,length(\$2)}' .env

Then open **.env** to add your provider keys:

sudo nano .env

3. Add at least one AI model provider API key*

OPENAI_API_KEY=your_key_here

or

ANTHROPIC_API_KEY=your_key_here

or

OPENROUTER_API_KEY=your_key_here

or

GROQ_API_KEY=your_key_here

- **Exit & Save**

Start the app

```
sudo docker compose pull
```

```
sudo docker compose up -d
```

5) Verify it's running

```
sudo docker ps --format '{{.Names}}\t{{.Status}}'
```

(Look for: LibreChat, chat-mongodb, chat-meilisearch)

Create your login credentials

```
docker exec -it LibreChat sh -lc 'npm run create-user'
```

You'll be prompted for name/email/password (and sometimes admin). When it finishes, you can log in on the web UI immediately.

6) Log in

Open in your browser:

http://Your_Instance_Public__IP

Example: **http://203.0.113.10**

- Click **Sign up** to create your first account (becomes admin).
- Inside the UI, you can add more providers in **Settings** → **Providers**.

Common tasks

Stop the stack

```
cd ~/LibreChat && sudo docker compose down
```

Start the stack

```
cd ~/LibreChat && sudo docker compose up -d
```

Update to the latest images

```
cd ~/LibreChat
```

```
sudo docker compose pull
sudo docker compose up -d --remove-orphans
```

View logs

```
sudo docker logs -f LibreChat
```

Where your data lives (for backup/restore)

- MongoDB: ~/LibreChat/data-node/
- Meilisearch: ~/LibreChat/meili_data_v1.6/
- File uploads: ~/LibreChat/uploads/
- Logs: ~/LibreChat/logs/

To use LibreChat, you'll need at least one model provider API key. Pick a provider (e.g., OpenAI, Anthropic, OpenRouter, Groq), create an account, then go to the provider's dashboard and generate a new **API key (sometimes called an access token). Copy the key and paste it into ~/LibreChat/.env under the matching variable (e.g., OPENAI_API_KEY=...). Treat it like a password: don't share it, don't commit it to Git, and consider setting spend/rate limits or rotating it periodically. For exact steps, see the "API Keys" or "Access" section of your chosen provider's documentation.*

Extra Information: (Optional)

Allocate Elastic IP

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>