



☰ Install & Updates > **Deploy on Render**

Install & Updates

Deploy on Render

Deploy OpenClaw on Render using Infrastructure as Code. The included `render.yaml` Blueprint defines your entire stack declaratively, service, disk, environment variables, so you can deploy with a single click and version your infrastructure alongside your code.

Prerequisites

A Render account (free tier available)

An API key from your preferred model provider

Deploy with a Render Blueprint


Deploy to Render

Clicking this link will:

1. Create a new Render service from the `render.yaml` Blueprint at the root of this repo.
2. Prompt you to set `SETUP_PASSWORD`
3. Build the Docker image and deploy

Once deployed, your service URL follows the pattern `https://<service-name>.onrender.com` .

Understanding the Blueprint


Render Blueprints are YAML files that define your infrastructure. The  `render.yaml` in this repository configures everything needed to run OpenClaw:

```
services:
  - type: web
    name: openclaw
    runtime: docker
    plan: starter
    healthCheckPath: /health
    envVars:
      - key: PORT
        value: "8080"
      - key: SETUP_PASSWORD
        sync: false # prompts during deploy
      - key: OPENCLAW_STATE_DIR
        value: /data/.openclaw
      - key: OPENCLAW_WORKSPACE_DIR
        value: /data/workspace
      - key: OPENCLAW_GATEWAY_TOKEN
        generateValue: true # auto-generates a secure token
    disk:
      name: openclaw-data
      mountPath: /data
      sizeGB: 1
```

Key Blueprint features used:

Feature	Purpose
<code>runtime: docker</code>	Builds from the repo's Dockerfile
<code>healthCheckPath</code>	Render monitors <code>/health</code> and restarts unhealthy instances
<code>sync: false</code>	Prompts for value during deploy (secrets)
<code>generateValue: true</code>	Auto-generates a cryptographically secure value
<code>disk</code>	Persistent storage that survives redeploys

Choosing a plan

 an	Spin-down	Disk	Best for
Free	After 15 min idle	Not available	Testing, demos
Starter	Never	1GB+	Personal use, small teams
Standard+	Never	1GB+	Production, multiple channels

The Blueprint defaults to `starter` . To use free tier, change `plan: free` in your fork's `render.yaml` (but note: no persistent disk means config resets on each deploy).

After deployment

Complete the setup wizard

1. Navigate to `https://<your-service>.onrender.com/setup`
2. Enter your `SETUP_PASSWORD`
3. Select a model provider and paste your API key
4. Optionally configure messaging channels (Telegram, Discord, Slack)
5. Click **Run setup**

Access the Control UI

The web dashboard is available at `https://<your-service>.onrender.com/openclaw` .

Render Dashboard features

Logs

View real-time logs in **Dashboard** → **your service** → **Logs** . Filter by:

Build logs (Docker image creation)

Deploy logs (service startup)

Runtime logs (application output)

Shell access



For debugging, open a shell session via **Dashboard** → **your service** → **Shell**. The persistent disk is mounted at `/data`.

Environment variables

Modify variables in **Dashboard** → **your service** → **Environment**. Changes trigger an automatic redeploy.

Auto-deploy

If you use the original OpenClaw repository, Render will not auto-deploy your OpenClaw. To update it, run a manual Blueprint sync from the dashboard.

Custom domain

1. Go to **Dashboard** → **your service** → **Settings** → **Custom Domains**
2. Add your domain
3. Configure DNS as instructed (CNAME to `*.onrender.com`)
4. Render provisions a TLS certificate automatically

Scaling

Render supports horizontal and vertical scaling:

Vertical: Change the plan to get more CPU/RAM

Horizontal: Increase instance count (Standard plan and above)

For OpenClaw, vertical scaling is usually sufficient. Horizontal scaling requires sticky sessions or external state management.

Backups and migration

Export your configuration and workspace at any time:

<https://<your-service>.onrender.com/setup/export>

This downloads a portable backup you can restore on any OpenClaw host.

Troubleshooting

Service won't start

Check the deploy logs in the Render Dashboard. Common issues:

Missing `SETUP_PASSWORD` – the Blueprint prompts for this, but verify it's set

Port mismatch – ensure `PORT=8080` matches the Dockerfile's exposed port

Slow cold starts (free tier)

Free tier services spin down after 15 minutes of inactivity. The first request after spin-down takes a few seconds while the container starts. Upgrade to Starter plan for always-on.

Data loss after redeploy

This happens on free tier (no persistent disk). Upgrade to a paid plan, or regularly export your config via `/setup/export`.

Health check failures

Render expects a 200 response from `/health` within 30 seconds. If builds succeed but deploys fail, the service may be taking too long to start. Check:

Build logs for errors

Whether the container runs locally with `docker build && docker run`

[◀ Deploy on Railway](#)

[Deploy on Northflank >](#)



Powered by mintlify

>