Hosting and deployment

# Fly.io

Deploy OpenClaw on Fly.io

**Goal:** OpenClaw Gateway running on a **Fly.io** machine with persistent storage, automatic HTTPS, and Discord/channel access.

## What you need

- **flyctl CLI** installed
- Fly.io account (free tier works)
- Model auth: Anthropic API key (or other provider keys)
- Channel credentials: Discord bot token, Telegram token, etc.

## Beginner quick path

1. Clone repo → customize `fly.toml`
2. Create app + volume → set secrets
3. Deploy with `fly deploy`
4. SSH in to create config or use Control UI

## 1) Create the Fly app

```
# Clone the repo
git clone https://github.com/openclaw/openclaw.git
cd openclaw

# Create a new Fly app (pick your own name)
fly apps create my-openclaw

# Create a persistent volume (1GB is usually enough)
fly volumes create openclaw_data --size 1 --region iad
```

**Tip:** Choose a region close to you. Common options: `lhr` (London), `iad` (Virginia), `sjc` (San Jose).

## 2) Configure fly.toml

Edit `fly.toml` to match your app name and requirements.

**Security note:** The default config exposes a public URL. For a hardened deployment with no public IP, see or use `fly.private.toml`.

```
app = "my-openclaw"  # Your app name
primary_region = "iad"

[build]
  dockerfile = "Dockerfile"

[env]
  NODE_ENV = "production"
  OPENCLAW_PREFER_PNPM = "1"
  OPENCLAW_STATE_DIR = "/data"
  NODE_OPTIONS = "--max-old-space-size=1536"

[processes]
  app = "node dist/index.js gateway --allow-unconfigured --port 3000 --bind lan"

[http_service]
  internal_port = 3000
  force_https = true
  auto_stop_machines = false
  auto_start_machines = true
  min_machines_running = 1
  processes = ["app"]

[[vm]]
  size = "shared-cpu-2x"
  memory = "2048mb"

[mounts]
  source = "openclaw_data"
  destination = "/data"
```

**Key settings:**

| Setting | Why |
| --- | --- |
| `--bind lan` | Binds to `0.0.0.0` so Fly's proxy can reach the gateway |
| `--allow-unconfigured` | Starts without a config file (you'll create one after) |

| Setting | Why |
| --- | --- |
| `internal_port = 3000` | Must match `--port 3000` (or `OPENCLAW_GATEWAY_PORT`) for Fly health checks |
| `memory = "2048mb"` | 512MB is too small; 2GB recommended |
| `OPENCLAW_STATE_DIR = "/data"` | Persists state on the volume |

## 3) Set secrets

```
# Required: Gateway token (for non-loopback binding)
fly secrets set OPENCLAW_GATEWAY_TOKEN=$(openssl rand -hex 32)

# Model provider API keys
fly secrets set ANTHROPIC_API_KEY=sk-ant-...

# Optional: Other providers
fly secrets set OPENAI_API_KEY=sk-...
fly secrets set GOOGLE_API_KEY=...

# Channel tokens
fly secrets set DISCORD_BOT_TOKEN=MTQ...
```

**Notes:**

- Non-loopback binds (`--bind lan`) require `OPENCLAW_GATEWAY_TOKEN` for security.
- Treat these tokens like passwords.
- **Prefer env vars over config file** for all API keys and tokens. This keeps secrets out of `openclaw.json` where they could be accidentally exposed or logged.

## 4) Deploy

```
fly deploy
```

First deploy builds the Docker image (~2-3 minutes). Subsequent deploys are faster.

After deployment, verify:

```
fly status
fly logs
```

You should see:

```
[gateway] listening on ws://0.0.0.0:3000 (PID xxx)
[discord] logged in to discord as xxx
```

## 5) Create config file

SSH into the machine to create a proper config:

```
fly ssh console
```

Create the config directory and file:

```
mkdir -p /data
cat > /data/openclaw.json << 'EOF'
{
  "agents": {
    "defaults": {
      "model": {
        "primary": "anthropic/claude-opus-4-6",
        "fallbacks": ["anthropic/claude-sonnet-4-5", "openai/gpt-4o"]
      },
      "maxConcurrent": 4
    },
    "list": [
      {
        "id": "main",
        "default": true
      }
    ]
  },
  "auth": {
    "profiles": {
      "anthropic:default": { "mode": "token", "provider": "anthropic" },
      "openai:default": { "mode": "token", "provider": "openai" }
    }
  },
  "bindings": [
    {
      "agentId": "main",
      "match": { "channel": "discord" }
    }
  ],
  "channels": {
    "discord": {
      "enabled": true,
      "groupPolicy": "allowlist",
      "guilds": {
        "YOUR_GUILD_ID": {
          "channels": { "general": { "allow": true } },
          "requireMention": false
        }
      }
```

```
    }
  },
  "gateway": {
    "mode": "local",
    "bind": "auto"
  },
  "meta": {
    "lastTouchedVersion": "2026.1.29"
  }
}
EOF
```

**Note:** With `OPENCLAW_STATE_DIR=/data`, the config path is `/data/openclaw.json`.

**Note:** The Discord token can come from either:

- Environment variable: `DISCORD_BOT_TOKEN` (recommended for secrets)
- Config file: `channels.discord.token`

If using env var, no need to add token to config. The gateway reads `DISCORD_BOT_TOKEN` automatically.

Restart to apply:

```
exit
fly machine restart <machine-id>
```

## 6) Access the Gateway

### Control UI

Open in browser:

```
fly open
```

Or visit `https://my-openclaw.fly.dev/`

Paste your gateway token (the one from `OPENCLAW_GATEWAY_TOKEN`) to authenticate.

### Logs

```
fly logs              # Live logs
fly logs --no-tail    # Recent logs
```

### SSH Console

```
fly ssh console
```

## Troubleshooting

### "App is not listening on expected address"

The gateway is binding to `127.0.0.1` instead of `0.0.0.0`.

**Fix:** Add `--bind lan` to your process command in `fly.toml`.

### Health checks failing / connection refused

Fly can't reach the gateway on the configured port.

**Fix:** Ensure `internal_port` matches the gateway port (set `--port 3000` or `OPENCLAW_GATEWAY_PORT=3000`).

### OOM / Memory Issues

Container keeps restarting or getting killed. Signs: `SIGABRT`, `v8::internal::Runtime_AllocateInYoungGeneration`, or silent restarts.

**Fix:** Increase memory in `fly.toml`:

```
[[vm]]
  memory = "2048mb"
```

Or update an existing machine:

```
fly machine update <machine-id> --vm-memory 2048 -y
```

**Note:** 512MB is too small. 1GB may work but can OOM under load or with verbose logging. **2GB is recommended.**

## Gateway Lock Issues

Gateway refuses to start with "already running" errors.

This happens when the container restarts but the PID lock file persists on the volume.

**Fix:** Delete the lock file:

```
fly ssh console --command "rm -f /data/gateway.*.lock"
fly machine restart <machine-id>
```

The lock file is at `/data/gateway.*.lock` (not in a subdirectory).

## Config Not Being Read

If using `--allow-unconfigured`, the gateway creates a minimal config. Your custom config at `/data/openclaw.json` should be read on restart.

Verify the config exists:

```
fly ssh console --command "cat /data/openclaw.json"
```

›

### Writing Config via SSH

The `fly ssh console -C` command doesn't support shell redirection. To write a config file:

```
# Use echo + tee (pipe from local to remote)
echo '{"your":"config"}' | fly ssh console -C "tee /data/openclaw.json"

# Or use sftp
fly sftp shell
> put /local/path/config.json /data/openclaw.json
```

**Note:** `fly sftp` may fail if the file already exists. Delete first:

```
fly ssh console --command "rm /data/openclaw.json"
```

### State Not Persisting

If you lose credentials or sessions after a restart, the state dir is writing to the container filesystem.

**Fix:** Ensure `OPENCLAW_STATE_DIR=/data` is set in `fly.toml` and redeploy.

## Updates

```
# Pull latest changes
git pull

# Redeploy
fly deploy

# Check health
fly status
fly logs
```

### Updating Machine Command

If you need to change the startup command without a full redeploy:

```
# Get machine ID
fly machines list

# Update command
fly machine update <machine-id> --command "node dist/index.js gateway --port 3000

# Or with memory increase
fly machine update <machine-id> --vm-memory 2048 --command "node dist/index.js ga
```

**Note:** After `fly deploy`, the machine command may reset to what's in `fly.toml`. If you made manual changes, re-apply them after deploy.

## Private Deployment (Hardened)

By default, Fly allocates public IPs, making your gateway accessible at `https://your-app.fly.dev`. This is convenient but means your deployment is discoverable by internet scanners (Shodan, Censys, etc.).

For a hardened deployment with **no public exposure**, use the private template.

## When to use private deployment

- You only make **outbound** calls/messages (no inbound webhooks)
- You use **ngrok or Tailscale** tunnels for any webhook callbacks
- You access the gateway via **SSH, proxy, or WireGuard** instead of browser
- You want the deployment **hidden from internet scanners**

## Setup

Use `fly.private.toml` instead of the standard config:

```
# Deploy with private config
fly deploy -c fly.private.toml
```

Or convert an existing deployment:

```
# List current IPs
fly ips list -a my-openclaw

# Release public IPs
fly ips release <public-ipv4> -a my-openclaw
fly ips release <public-ipv6> -a my-openclaw

# Switch to private config so future deploys don't re-allocate public IPs
# (remove [http_service] or deploy with the private template)
fly deploy -c fly.private.toml

# Allocate private-only IPv6
fly ips allocate-v6 --private -a my-openclaw
```

After this, `fly ips list` should show only a `private` type IP:

```
VERSION  IP                   TYPE             REGION
v6       fdaa:x:x:x:x::x      private          global
```

## Accessing a private deployment

Since there's no public URL, use one of these methods:

### Option 1: Local proxy (simplest)

```
# Forward local port 3000 to the app
fly proxy 3000:3000 -a my-openclaw

# Then open http://localhost:3000 in browser
```

### Option 2: WireGuard VPN

```
# Create WireGuard config (one-time)
fly wireguard create

# Import to WireGuard client, then access via internal IPv6
# Example: http://[fdaa:x:x:x:x::x]:3000
```

### Option 3: SSH only

```
fly ssh console -a my-openclaw
```

## Webhooks with private deployment

If you need webhook callbacks (Twilio, Telnyx, etc.) without public exposure:

1. **ngrok tunnel** - Run ngrok inside the container or as a sidecar
2. **Tailscale Funnel** - Expose specific paths via Tailscale
3. **Outbound-only** - Some providers (Twilio) work fine for outbound calls without webhooks

Example voice-call config with ngrok:

```
{
  "plugins": {
    "entries": {
      "voice-call": {
        "enabled": true,
        "config": {
          "provider": "twilio",
          "tunnel": { "provider": "ngrok" },
          "webhookSecurity": {
            "allowedHosts": ["example.ngrok.app"]
          }
        }
      }
    }
  }
}
```

The ngrok tunnel runs inside the container and provides a public webhook URL without exposing the Fly app itself. Set `webhookSecurity.allowedHosts` to the public tunnel hostname so forwarded host headers are accepted.

## Security benefits

| Aspect | Public | Private |
|---|---|---|
| Internet scanners | Discoverable | Hidden |
| Direct attacks | Possible | Blocked |

| Aspect | Public | Private |
| --- | --- | --- |
| Control UI access | Browser | Proxy/VPN |
| Webhook delivery | Direct | Via tunnel |

## Notes

- Fly.io uses **x86 architecture** (not ARM)
- The Dockerfile is compatible with both architectures
- For WhatsApp/Telegram onboarding, use `fly ssh console`
- Persistent data lives on the volume at `/data`
- Signal requires Java + signal-cli; use a custom image and keep memory at 2GB+.

## Cost

With the recommended config ( `shared-cpu-2x` , 2GB RAM):

- ~$10-15/month depending on usage
- Free tier includes some allowance

See **Fly.io pricing** for details.

‹ Uninstall Hetzner ›