Hosting and deployment

GCP

Goal

Run a persistent OpenClaw Gateway on a GCP Compute Engine VM using Docker, with durable state, baked-in binaries, and safe restart behavior.

If you want “OpenClaw 24/7 for ~$5-12/mo”, this is a reliable setup on Google Cloud. Pricing varies by machine type and region; pick the smallest VM that fits your workload and scale up if you hit OOMs.

What are we doing (simple terms)?

- Create a GCP project and enable billing
- Create a Compute Engine VM
- Install Docker (isolated app runtime)
- Start the OpenClaw Gateway in Docker
- Persist `~/.openclaw` + `~/.openclaw/workspace` on the host (survives restarts/rebuilds)
- Access the Control UI from your laptop via an SSH tunnel

The Gateway can be accessed via:

- SSH port forwarding from your laptop
- Direct port exposure if you manage firewalling and tokens yourself

This guide uses Debian on GCP Compute Engine. Ubuntu also works; map packages accordingly. For the generic Docker flow, see Docker.

›

Quick path (experienced operators)

1. Create GCP project + enable Compute Engine API
2. Create Compute Engine VM (e2-small, Debian 12, 20GB)
3. SSH into the VM
4. Install Docker
5. Clone OpenClaw repository
6. Create persistent host directories
7. Configure `.env` and `docker-compose.yml`
8. Bake required binaries, build, and launch

What you need

- GCP account (free tier eligible for e2-micro)
- gcloud CLI installed (or use Cloud Console)
- SSH access from your laptop
- Basic comfort with SSH + copy/paste
- ~20-30 minutes
- Docker and Docker Compose
- Model auth credentials
- Optional provider credentials
  - WhatsApp QR

Telegram bot token

Gmail OAuth

›

1) Install gcloud CLI (or use Console)

Option A: gcloud CLI (recommended for automation)

Install from **https://cloud.google.com/sdk/docs/install**

Initialize and authenticate:

```
gcloud init
gcloud auth login
```

Option B: Cloud Console

All steps can be done via the web UI at

2) Create a GCP project

CLI:

```
gcloud projects create my-openclaw-project --name="OpenClaw Gateway"
gcloud config set project my-openclaw-project
```

Enable billing at (required for Compute Engine).

Enable the Compute Engine API:

```
gcloud services enable compute.googleapis.com
```

Console:

1. Go to IAM & Admin > Create Project
2. Name it and create
3. Enable billing for the project
4. Navigate to APIs & Services > Enable APIs > search “Compute Engine API” > Enable

3) Create the VM

Machine types:

Type	Specs	Cost	Notes
e2-small	2 vCPU, 2GB RAM	~$12/mo	Recommended
e2-micro	2 vCPU (shared), 1GB RAM	Free tier eligible	May OOM under load

CLI:

```
gcloud compute instances create openclaw-gateway \
  --zone=us-central1-a \
  --machine-type=e2-small \
  --boot-disk-size=20GB \
  --image-family=debian-12 \
  --image-project=debian-cloud
```

Console:

1. Go to Compute Engine > VM instances > Create instance
2. Name: `openclaw-gateway`
3. Region: `us-central1` , Zone: `us-central1-a`
4. Machine type: `e2-small`
5. Boot disk: Debian 12, 20GB
6. Create

4) SSH into the VM

CLI:

```
gcloud compute ssh openclaw-gateway --zone=us-central1-a
```

Console:

Click the “SSH” button next to your VM in the Compute Engine dashboard.

Note: SSH key propagation can take 1-2 minutes after VM creation. If connection is refused, wait and retry.

5) Install Docker (on the VM)

```
sudo apt-get update
sudo apt-get install -y git curl ca-certificates
curl -fsSL https://get.docker.com | sudo sh
sudo usermod -aG docker $USER
```

Log out and back in for the group change to take effect:

```
exit
```

Then SSH back in:

```
gcloud compute ssh openclaw-gateway --zone=us-central1-a
```

Verify:

```
docker --version
docker compose version
```

6) Clone the OpenClaw repository

```
git clone https://github.com/openclaw/openclaw.git
cd openclaw
```

This guide assumes you will build a custom image to guarantee binary persistence.

7) Create persistent host directories

Docker containers are ephemeral. All long-lived state must live on the host.

```
mkdir -p ~/.openclaw
mkdir -p ~/.openclaw/workspace
```

›

8) Configure environment variables

Create `.env` in the repository root.

```
OPENCLAW_IMAGE=openclaw:latest
OPENCLAW_GATEWAY_TOKEN=change-me-now
OPENCLAW_GATEWAY_BIND=lan
OPENCLAW_GATEWAY_PORT=18789

OPENCLAW_CONFIG_DIR=/home/$USER/.openclaw
OPENCLAW_WORKSPACE_DIR=/home/$USER/.openclaw/workspace

GOG_KEYRING_PASSWORD=change-me-now
XDG_CONFIG_HOME=/home/node/.openclaw
```

Generate strong secrets:

```
openssl rand -hex 32
```

Do not commit this file.

9) Docker Compose configuration

Create or update `docker-compose.yml` .

```
services:
  openclaw-gateway:
    image: ${OPENCLAW_IMAGE}
    build: .
    restart: unless-stopped
    env_file:
      - .env
    environment:
      - HOME=/home/node
      - NODE_ENV=production
      - TERM=xterm-256color
      - OPENCLAW_GATEWAY_BIND=${OPENCLAW_GATEWAY_BIND}
      - OPENCLAW_GATEWAY_PORT=${OPENCLAW_GATEWAY_PORT}
      - OPENCLAW_GATEWAY_TOKEN=${OPENCLAW_GATEWAY_TOKEN}
      - GOG_KEYRING_PASSWORD=${GOG_KEYRING_PASSWORD}
      - XDG_CONFIG_HOME=${XDG_CONFIG_HOME}
      - PATH=/home/linuxbrew/.linuxbrew/bin:/usr/local/sbin:/usr/local/bin:/usr/sl
    volumes:
      - ${OPENCLAW_CONFIG_DIR}:/home/node/.openclaw
      - ${OPENCLAW_WORKSPACE_DIR}:/home/node/.openclaw/workspace
    ports:
      # Recommended: keep the Gateway loopback-only on the VM; access via SSH tunn
      # To expose it publicly, remove the `127.0.0.1:` prefix and firewall accordi
      - "127.0.0.1:${OPENCLAW_GATEWAY_PORT}:18789"
    command:
      [
        "node",
        "dist/index.js",
        "gateway",
        "--bind",
        "${OPENCLAW_GATEWAY_BIND}",
        "--port",
        "${OPENCLAW_GATEWAY_PORT}",
      ]
```

10) Bake required binaries into the image (critical)

Installing binaries inside a running container is a trap. Anything installed at runtime will be lost on restart.

All external binaries required by skills must be installed at image build time.

The examples below show three common binaries only:

- `gog` for Gmail access
- `goplaces` for Google Places
- `wacli` for WhatsApp

These are examples, not a complete list. You may install as many binaries as needed using the same pattern.

If you add new skills later that depend on additional binaries, you must:

1. Update the Dockerfile
2. Rebuild the image
3. Restart the containers

Example Dockerfile

```
FROM node:22-bookworm

RUN apt-get update && apt-get install -y socat && rm -rf /var/lib/apt/lists/*

# Example binary 1: Gmail CLI
RUN curl -L https://github.com/steipete/gog/releases/latest/download/gog_Linux_x8
  | tar -xz -C /usr/local/bin && chmod +x /usr/local/bin/gog

# Example binary 2: Google Places CLI
RUN curl -L https://github.com/steipete/goplaces/releases/latest/download/goplace
  | tar -xz -C /usr/local/bin && chmod +x /usr/local/bin/goplaces

# Example binary 3: WhatsApp CLI
RUN curl -L https://github.com/steipete/wacli/releases/latest/download/wacli_Linu
  | tar -xz -C /usr/local/bin && chmod +x /usr/local/bin/wacli

# Add more binaries below using the same pattern

WORKDIR /app
COPY package.json pnpm-lock.yaml pnpm-workspace.yaml .npmrc ./
COPY ui/package.json ./ui/package.json
COPY scripts ./scripts

RUN corepack enable
RUN pnpm install --frozen-lockfile

COPY . .
RUN pnpm build
RUN pnpm ui:install
RUN pnpm ui:build

ENV NODE_ENV=production

CMD ["node","dist/index.js"]
```

11) Build and launch

```
docker compose build
docker compose up -d openclaw-gateway
```

Verify binaries:

```
docker compose exec openclaw-gateway which gog
docker compose exec openclaw-gateway which goplaces
docker compose exec openclaw-gateway which wacli
```

Expected output:

```
/usr/local/bin/gog
/usr/local/bin/goplaces
/usr/local/bin/wacli
```

12) Verify Gateway

```
docker compose logs -f openclaw-gateway
```

Success:

```
[gateway] listening on ws://0.0.0.0:18789
```

13) Access from your laptop

Create an SSH tunnel to forward the Gateway port:

```
gcloud compute ssh openclaw-gateway --zone=us-central1-a -- -L 18789:1
```

Open in your browser:

`http://127.0.0.1:18789/`

Paste your gateway token.

What persists where (source of truth)

OpenClaw runs in Docker, but Docker is not the source of truth. All long-lived state must survive restarts, rebuilds, and reboots.

Component	Location	Persistence mechanism	Notes
Gateway config	`/home/node/.openclaw/`	Host volume mount	Includes `openclaw.json` , tokens
Model auth profiles	`/home/node/.openclaw/`	Host volume mount	OAuth tokens, API keys
Skill configs	`/home/node/.openclaw/skills/`	Host volume mount	Skill-level state
Agent workspace	`/home/node/.openclaw/workspace/`	Host volume mount	Code and agent artifacts
WhatsApp session	`/home/node/.openclaw/`	Host volume mount	Preserves QR login
Gmail keyring	`/home/node/.openclaw/`	Host volume + password	Requires `GOG_KEYRING_PASSWORD`
External binaries	`/usr/local/bin/`	Docker image	Must be baked at build time

mponent	Location	Persistence mechanism	Notes
Node runtime	Container filesystem	Docker image	Rebuilt every image build
OS packages	Container filesystem	Docker image	Do not install at runtime
Docker container	Ephemeral	Restartable	Safe to destroy

Updates

To update OpenClaw on the VM:

```
cd ~/openclaw
git pull
docker compose build
docker compose up -d
```

Troubleshooting

SSH connection refused

SSH key propagation can take 1-2 minutes after VM creation. Wait and retry.

OS Login issues

Check your OS Login profile:

```
gcloud compute os-login describe-profile
```

Ensure your account has the required IAM permissions (Compute OS Login or Compute OS Admin Login).

Out of memory (OOM)

If using e2-micro and hitting OOM, upgrade to e2-small or e2-medium:

```
# Stop the VM first
gcloud compute instances stop openclaw-gateway --zone=us-central1-a

# Change machine type
gcloud compute instances set-machine-type openclaw-gateway \
  --zone=us-central1-a \
  --machine-type=e2-small

# Start the VM
gcloud compute instances start openclaw-gateway --zone=us-central1-a
```

Service accounts (security best practice)

For personal use, your default user account works fine.

For automation or CI/CD pipelines, create a dedicated service account with minimal permissions:

1. Create a service account:

   ```
   gcloud iam service-accounts create openclaw-deploy \
     --display-name="OpenClaw Deployment"
   ```

2. Grant Compute Instance Admin role (or narrower custom role):

```
gcloud projects add-iam-policy-binding my-openclaw-project \
  --member="serviceAccount:openclaw-deploy@my-openclaw-project.iam.gserviceac
  --role="roles/compute.instanceAdmin.v1"
```

Avoid using the Owner role for automation. Use the principle of least privilege.

See for IAM role details.

Next steps

- Set up messaging channels:
- Pair local devices as nodes:
- Configure the Gateway:

< Hetzner macOS VMs >

Powered by mintlify