Fundamentals

Agent Workspace

The workspace is the agent's home. It is the only working directory used for file tools and for workspace context. Keep it private and treat it as memory.

This is separate from `~/.openclaw/`, which stores config, credentials, and sessions.

Important: the workspace is the **default cwd**, not a hard sandbox. Tools resolve relative paths against the workspace, but absolute paths can still reach elsewhere on the host unless sandboxing is enabled. If you need isolation, use `agents.defaults.sandbox` (and/or per-agent sandbox config). When sandboxing is enabled and `workspaceAccess` is not `"rw"`, tools operate inside a sandbox workspace under `~/.openclaw/sandboxes`, not your host workspace.

Default location

Default: `~/.openclaw/workspace`

If `OPENCLAW_PROFILE` is set and not `"default"`, the default becomes `~/.openclaw/workspace-<profile>`.

Override in `~/.openclaw/openclaw.json`:

```
{
  agent: {
    workspace: "~/.openclaw/workspace",
  },
}
```

`openclaw onboard`, `openclaw configure`, or `openclaw setup` will create the workspace and seed the bootstrap files if they are missing.

If you already manage the workspace files yourself, you can disable bootstrap file creation:

```
{ agent: { skipBootstrap: true } }
```

Extra workspace folders

Older installs may have created `~/openclaw`. Keeping multiple workspace directories around can cause confusing auth or state drift, because only one workspace is active at a time.

Recommendation: keep a single active workspace. If you no longer use the extra folders, archive or move them to Trash (for example `trash ~/openclaw`). If you intentionally keep multiple workspaces, make sure `agents.defaults.workspace` points to the active one.

`openclaw doctor` warns when it detects extra workspace directories.

Workspace file map (what each file means)

These are the standard files OpenClaw expects inside the workspace:

- `AGENTS.md`

Operating instructions for the agent and how it should use memory.

Loaded at the start of every session.

Good place for rules, priorities, and “how to behave” details.

`SOUL.md`

Persona, tone, and boundaries.

Loaded every session.

`USER.md`

Who the user is and how to address them.

Loaded every session.

`IDENTITY.md`

The agent’s name, vibe, and emoji.

Created/updated during the bootstrap ritual.

`TOOLS.md`

Notes about your local tools and conventions.

Does not control tool availability; it is only guidance.

`HEARTBEAT.md`

Optional tiny checklist for heartbeat runs.

Keep it short to avoid token burn.

`BOOT.md`

Optional startup checklist executed on gateway restart when internal hooks are enabled.

Keep it short; use the message tool for outbound sends.

`BOOTSTRAP.md`

One-time first-run ritual.

Only created for a brand-new workspace.

Delete it after the ritual is complete.

`memory/YYYY-MM-DD.md`

Daily memory log (one file per day).

Recommended to read today + yesterday on session start.

`MEMORY.md` (optional)

Curated long-term memory.

Only load in the main, private session (not shared/group contexts).

See **Memory** for the workflow and automatic memory flush.

`skills/` (optional)

Workspace-specific skills.

Overrides managed/bundled skills when names collide.

`canvas/` (optional)

Canvas UI files for node displays (for example `canvas/index.html`).

If any bootstrap file is missing, OpenClaw injects a "missing file" marker into the session and continues. Large bootstrap files are truncated when injected; adjust limits with `agents.defaults.bootstrapMaxChars` (default: 20000) and `agents.defaults.bootstrapTotalMaxChars` (default: 150000). `openclaw setup` can recreate missing defaults without overwriting existing files.

What is NOT in the workspace

These live under `~/.openclaw/` and should NOT be committed to the workspace repo:

`~/.openclaw/openclaw.json` (config)

```
~/.openclaw/credentials/  (OAuth tokens, API keys)
~/.openclaw/agents/<agentId>/sessions/  (session transcripts + metadata)
~/.openclaw/skills/  (managed skills)
```

If you need to migrate sessions or config, copy them separately and keep them out of version control.

Git backup (recommended, private)

Treat the workspace as private memory. Put it in a **private** git repo so it is backed up and recoverable.

Run these steps on the machine where the Gateway runs (that is where the workspace lives).

1) Initialize the repo

If git is installed, brand-new workspaces are initialized automatically. If this workspace is not already a repo, run:

```
cd ~/.openclaw/workspace
git init
git add AGENTS.md SOUL.md TOOLS.md IDENTITY.md USER.md HEARTBEAT.md memory/
git commit -m "Add agent workspace"
```

2) Add a private remote (beginner-friendly options)

Option A: GitHub web UI

1. Create a new **private** repository on GitHub.
2. Do not initialize with a README (avoids merge conflicts).
3. Copy the HTTPS remote URL.
4. Add the remote and push:

```
git branch -M main
git remote add origin <https-url>
git push -u origin main
```

Option B: GitHub CLI (`gh`)

```
gh auth login
gh repo create openclaw-workspace --private --source . --remote origin --push
```

Option C: GitLab web UI

1. Create a new **private** repository on GitLab.
2. Do not initialize with a README (avoids merge conflicts).
3. Copy the HTTPS remote URL.
4. Add the remote and push:

```
git branch -M main
git remote add origin <https-url>
git push -u origin main
```

3) Ongoing updates

```
git status
git add .
git commit -m "Update memory"
git push
```

Do not commit secrets

Even in a private repo, avoid storing secrets in the workspace:

API keys, OAuth tokens, passwords, or private credentials.

Anything under `~/.openclaw/`.

Raw dumps of chats or sensitive attachments.

If you must store sensitive references, use placeholders and keep the real secret elsewhere (password manager, environment variables, or `~/.openclaw/`).

Suggested `.gitignore` starter:

```
.DS_Store
.env
**/*.key
**/*.pem
**/secrets*
```

Moving the workspace to a new machine

1. Clone the repo to the desired path (default `~/.openclaw/workspace`).
2. Set `agents.defaults.workspace` to that path in `~/.openclaw/openclaw.json`.
3. Run `openclaw setup --workspace <path>` to seed any missing files.
4. If you need sessions, copy `~/.openclaw/agents/<agentId>/sessions/` from the old machine separately.

Advanced notes

Multi-agent routing can use different workspaces per agent. See for routing configuration.

If `agents.defaults.sandbox` is enabled, non-main sessions can use per-session sandbox workspaces under `agents.defaults.sandbox.workspaceRoot`.

‹ Context OAuth ›