Messaging platforms

IRC

Connect OpenClaw to IRC channels and direct messages.

Use IRC when you want OpenClaw in classic channels (`#room`) and direct messages. IRC ships as an extension plugin, but it is configured in the main config under `channels.irc` .

Quick start

1. Enable IRC config in `~/.openclaw/openclaw.json` .
2. Set at least:

```
{
  "channels": {
    "irc": {
      "enabled": true,
      "host": "irc.libera.chat",
      "port": 6697,
      "tls": true,
      "nick": "openclaw-bot",
      "channels": ["#openclaw"]
    }
  }
}
```

3. Start/restart gateway:

```
openclaw gateway run
```

Security defaults

- `channels.irc.dmPolicy` defaults to `"pairing"`.
- `channels.irc.groupPolicy` defaults to `"allowlist"`.

With `groupPolicy="allowlist"`, set `channels.irc.groups` to define allowed channels.

Use TLS (`channels.irc.tls=true`) unless you intentionally accept plaintext transport.

Access control

There are two separate “gates” for IRC channels:

1. **Channel access** (`groupPolicy` + `groups`): whether the bot accepts messages from a channel at all.
2. **Sender access** (`groupAllowFrom` / per-channel `groups["#channel"].allowFrom`): who is allowed to trigger the bot inside that channel.

Config keys:

DM allowlist (DM sender access): `channels.irc.allowFrom`

Group sender allowlist (channel sender access): `channels.irc.groupAllowFrom`

Per-channel controls (channel + sender + mention rules): `channels.irc.groups["#channel"]`

`channels.irc.groupPolicy="open"` allows unconfigured channels (**still mention-gated by default**)

Allowlist entries can use nick or `nick!user@host` forms.

Common gotcha: `allowFrom` is for DMs, not channels

If you see logs like:

```
irc: drop group sender alice!ident@host (policy=allowlist)
```

…it means the sender wasn't allowed for **group/channel** messages. Fix it by either:

setting `channels.irc.groupAllowFrom` (global for all channels), or

setting per-channel sender allowlists: `channels.irc.groups["#channel"].allowFrom`

Example (allow anyone in `#tuirc-dev` to talk to the bot):

```
{
  channels: {
    irc: {
      groupPolicy: "allowlist",
      groups: {
        "#tuirc-dev": { allowFrom: ["*"] },
      },
    },
  },
}
```

Reply triggering (mentions)

Even if a channel is allowed (via `groupPolicy` + `groups`) and the sender is allowed, OpenClaw defaults to **mention-gating** in group contexts.

That means you may see logs like `drop channel … (missing-mention)` unless the message includes a mention pattern that matches the bot.

To make the bot reply in an IRC channel **without needing a mention**, disable mention gating for that channel:

```
{
  channels: {
    irc: {
      groupPolicy: "allowlist",
      groups: {
        "#tuirc-dev": {
          requireMention: false,
          allowFrom: ["*"],
        },
      },
    },
  },
}
```

Or to allow **all** IRC channels (no per-channel allowlist) and still reply without mentions:

```
{
  channels: {
    irc: {
      groupPolicy: "open",
      groups: {
        "*": { requireMention: false, allowFrom: ["*"] },
      },
    },
  },
}
```

Security note (recommended for public channels)

If you allow `allowFrom: ["*"]` in a public channel, anyone can prompt the bot. To reduce risk, restrict tools for that channel.

Same tools for everyone in the channel

```
{
  channels: {
    irc: {
      groups: {
        "#tuirc-dev": {
          allowFrom: ["*"],
          tools: {
            deny: ["group:runtime", "group:fs", "gateway", "nodes", "cron", "brows
          },
        },
      },
    },
  },
}
```

Different tools per sender (owner gets more power)

Use `toolsBySender` to apply a stricter policy to `"*"` and a looser one to your nick:

```
{
  channels: {
    irc: {
      groups: {
        "#tuirc-dev": {
          allowFrom: ["*"],
          toolsBySender: {
            "*": {
              deny: ["group:runtime", "group:fs", "gateway", "nodes", "cron", "br
            },
            eigen: {
              deny: ["gateway", "nodes", "cron"],
            },
          },
        },
      },
    },
  },
}
```

Notes:

- `toolsBySender` keys can be a nick (e.g. `"eigen"`) or a full hostmask (`"eigen!~eigen@174.127.248.171"`) for stronger identity matching.
  The first matching sender policy wins; `"*"` is the wildcard fallback.

For more on group access vs mention-gating (and how they interact), see: .

NickServ

To identify with NickServ after connect:

```
{
  "channels": {
    "irc": {
      "nickserv": {
        "enabled": true,
        "service": "NickServ",
        "password": "your-nickserv-password"
      }
    }
  }
}
```

Optional one-time registration on connect:

```
{
  "channels": {
    "irc": {
      "nickserv": {
        "register": true,
        "registerEmail": "bot@example.com"
      }
    }
  }
}
```

Disable `register` after the nick is registered to avoid repeated REGISTER attempts.

Environment variables

Default account supports:

- `IRC_HOST`
- `IRC_PORT`
- `IRC_TLS`

```
IRC_NICK
IRC_USERNAME
IRC_REALNAME
IRC_PASSWORD
IRC_CHANNELS  (comma-separated)
IRC_NICKSERV_PASSWORD
IRC_NICKSERV_REGISTER_EMAIL
```

Troubleshooting

If the bot connects but never replies in channels, verify `channels.irc.groups` and whether mention-gating is dropping messages (`missing-mention`). If you want it to reply without pings, set `requireMention:false` for the channel.

If login fails, verify nick availability and server password.

If TLS fails on a custom network, verify host/port and certificate setup.

‹ Discord Slack ›