



≡ Configuration and operations > Configuration Examples

Configuration and operations

Configuration Examples

Examples below are aligned with the current config schema. For the exhaustive reference and per-field notes, see [Configuration](#).

Quick start

Absolute minimum

```
{
  agent: { workspace: "~/.openclaw/workspace" },
  channels: { whatsapp: { allowFrom: ["+15555550123"] } },
}
```

Save to `~/.openclaw/openclaw.json` and you can DM the bot from that number.

Recommended starter



```
identity: {  
  name: "Clawd",  
  theme: "helpful assistant",  
  emoji: "🦊",  
},  
agent: {  
  workspace: "~/.openclaw/workspace",  
  model: { primary: "anthropic/claude-sonnet-4-5" },  
},  
channels: {  
  whatsapp: {  
    allowFrom: ["+15555550123"],  
    groups: { "*": { requireMention: true } },  
  },  
},  
}
```

Expanded example (major options)

JSON5 lets you use comments and trailing commas. Regular JSON works too.



```
// Environment + shell
```

```
env: {
```

```
  OPENROUTER_API_KEY: "sk-or-...",
```

```
  vars: {
```

```
    GROQ_API_KEY: "gsk-...",
```

```
  },
```

```
  shellEnv: {
```

```
    enabled: true,
```

```
    timeoutMs: 15000,
```

```
  },
```

```
},
```

```
// Auth profile metadata (secrets live in auth-profiles.json)
```

```
auth: {
```

```
  profiles: {
```

```
    "anthropic:me@example.com": {
```

```
      provider: "anthropic",
```

```
      mode: "oauth",
```

```
      email: "me@example.com",
```

```
    },
```

```
    "anthropic:work": { provider: "anthropic", mode: "api_key" },
```

```
    "openai:default": { provider: "openai", mode: "api_key" },
```

```
    "openai-codex:default": { provider: "openai-codex", mode: "oauth" },
```

```
  },
```

```
  order: {
```

```
    anthropic: ["anthropic:me@example.com", "anthropic:work"],
```

```
    openai: ["openai:default"],
```

```
    "openai-codex": ["openai-codex:default"],
```

```
  },
```

```
},
```

```
// Identity
```

```
identity: {
```

```
  name: "Samantha",
```

```
  theme: "helpful sloth",
```

```
  emoji: "🦥",
```

```
},
```

```
// Logging
```



```
logging: {
  level: "info",
  file: "/tmp/openclaw/openclaw.log",
  consoleLevel: "info",
  consoleStyle: "pretty",
  redactSensitive: "tools",
},

// Message formatting
messages: {
  messagePrefix: "[openclaw]",
  responsePrefix: ">",
  ackReaction: "👁👁",
  ackReactionScope: "group-mentions",
},

// Routing + queue
routing: {
  groupChat: {
    mentionPatterns: ["@openclaw", "openclaw"],
    historyLimit: 50,
  },
  queue: {
    mode: "collect",
    debounceMs: 1000,
    cap: 20,
    drop: "summarize",
    byChannel: {
      whatsapp: "collect",
      telegram: "collect",
      discord: "collect",
      slack: "collect",
      signal: "collect",
      imessage: "collect",
      webchat: "collect",
    },
  },
},

// Tooling
```



```

tools: {
  media: {
    audio: {
      enabled: true,
      maxBytes: 20971520,
      models: [
        { provider: "openai", model: "gpt-4o-mini-transcribe" },
        // Optional CLI fallback (Whisper binary):
        // { type: "cli", command: "whisper", args: ["--model", "base", "{{Media
      ],
      timeoutSeconds: 120,
    },
    video: {
      enabled: true,
      maxBytes: 52428800,
      models: [{ provider: "google", model: "gemini-3-flash-preview" }],
    },
  },
},

// Session behavior
session: {
  scope: "per-sender",
  reset: {
    mode: "daily",
    atHour: 4,
    idleMinutes: 60,
  },
  resetByChannel: {
    discord: { mode: "idle", idleMinutes: 10080 },
  },
  resetTriggers: ["/new", "/reset"],
  store: "~/.openclaw/agents/default/sessions/sessions.json",
  maintenance: {
    mode: "warn",
    pruneAfter: "30d",
    maxEntries: 500,
    rotateBytes: "10mb",
  },
  typingIntervalSeconds: 5,
}

```



```
sendPolicy: {  
  default: "allow",  
  rules: [{ action: "deny", match: { channel: "discord", chatType: "group" } }],  
},  
>  
,
```

```
// Channels
```

```
channels: {  
  whatsapp: {  
    dmPolicy: "pairing",  
    allowFrom: ["+15555550123"],  
    groupPolicy: "allowlist",  
    groupAllowFrom: ["+15555550123"],  
    groups: { "*": { requireMention: true } },  
  },  
  
  telegram: {  
    enabled: true,  
    botToken: "YOUR_TELEGRAM_BOT_TOKEN",  
    allowFrom: ["123456789"],  
    groupPolicy: "allowlist",  
    groupAllowFrom: ["123456789"],  
    groups: { "*": { requireMention: true } },  
  },  
  
  discord: {  
    enabled: true,  
    token: "YOUR_DISCORD_BOT_TOKEN",  
    dm: { enabled: true, allowFrom: ["steipete"] },  
    guilds: {  
      "123456789012345678": {  
        slug: "friends-of-openclaw",  
        requireMention: false,  
        channels: {  
          general: { allow: true },  
          help: { allow: true, requireMention: true },  
        },  
      },  
    },  
  },  
},
```



```
slack: {
  enabled: true,
  botToken: "xoxb-REPLACE_ME",
  appToken: "xapp-REPLACE_ME",
  channels: {
    "#general": { allow: true, requireMention: true },
  },
  dm: { enabled: true, allowFrom: ["U123"] },
  slashCommand: {
    enabled: true,
    name: "openclaw",
    sessionPrefix: "slack:slash",
    ephemeral: true,
  },
},
},
},

// Agent runtime
agents: {
  defaults: {
    workspace: "~/.openclaw/workspace",
    userTimezone: "America/Chicago",
    model: {
      primary: "anthropic/claude-sonnet-4-5",
      fallbacks: ["anthropic/claude-opus-4-6", "openai/gpt-5.2"],
    },
    imageModel: {
      primary: "openrouter/anthropic/claude-sonnet-4-5",
    },
    models: {
      "anthropic/claude-opus-4-6": { alias: "opus" },
      "anthropic/claude-sonnet-4-5": { alias: "sonnet" },
      "openai/gpt-5.2": { alias: "gpt" },
    },
    thinkingDefault: "low",
    verboseDefault: "off",
    elevatedDefault: "on",
    blockStreamingDefault: "off",
    blockStreamingBreak: "text_end",
  },
}
```



```
blockStreamingChunk: {
  minChars: 800,
  maxChars: 1200,
  breakPreference: "paragraph",
},
blockStreamingCoalesce: {
  idleMs: 1000,
},
humanDelay: {
  mode: "natural",
},
timeoutSeconds: 600,
mediaMaxMb: 5,
typingIntervalSeconds: 5,
maxConcurrent: 3,
heartbeat: {
  every: "30m",
  model: "anthropic/claude-sonnet-4-5",
  target: "last",
  to: "+15555550123",
  prompt: "HEARTBEAT",
  ackMaxChars: 300,
},
memorySearch: {
  provider: "gemini",
  model: "gemini-embedding-001",
  remote: {
    apiKey: "${GEMINI_API_KEY}",
  },
  extraPaths: ["../team-docs", "/srv/shared-notes"],
},
sandbox: {
  mode: "non-main",
  perSession: true,
  workspaceRoot: "~/.openclaw/sandboxes",
  docker: {
    image: "openclaw-sandbox:bookworm-slim",
    workdir: "/workspace",
    readOnlyRoot: true,
    tmpfs: ["/tmp", "/var/tmp", "/run"],
  },
}
```




```
    network: "none",
    user: "1000:1000",
  },
  browser: {
    enabled: false,
  },
},
},
},

tools: {
  allow: ["exec", "process", "read", "write", "edit", "apply_patch"],
  deny: ["browser", "canvas"],
  exec: {
    backgroundMs: 10000,
    timeoutSec: 1800,
    cleanupMs: 1800000,
  },
  elevated: {
    enabled: true,
    allowFrom: {
      whatsapp: ["+15555550123"],
      telegram: ["123456789"],
      discord: ["steipete"],
      slack: ["U123"],
      signal: ["+15555550123"],
      imessage: ["user@example.com"],
      webchat: ["session:demo"],
    },
  },
},

// Custom model providers
models: {
  mode: "merge",
  providers: {
    "custom-proxy": {
      baseUrl: "http://localhost:4000/v1",
      apiKey: "LITELLM_KEY",
      api: "openai-responses",
    },
  },
}
```



```
authHeader: true,
headers: { "X-Proxy-Region": "us-west" },
models: [
  {
    id: "llama-3.1-8b",
    name: "Llama 3.1 8B",
    api: "openai-responses",
    reasoning: false,
    input: ["text"],
    cost: { input: 0, output: 0, cacheRead: 0, cacheWrite: 0 },
    contextWindow: 128000,
    maxTokens: 32000,
  },
],
},
},
},

// Cron jobs
cron: {
  enabled: true,
  store: "~/.openclaw/cron/cron.json",
  maxConcurrentRuns: 2,
  sessionRetention: "24h",
},

// Webhooks
hooks: {
  enabled: true,
  path: "/hooks",
  token: "shared-secret",
  presets: ["gmail"],
  transformsDir: "~/.openclaw/hooks/transforms",
  mappings: [
    {
      id: "gmail-hook",
      match: { path: "gmail" },
      action: "agent",
      wakeMode: "now",
      name: "Gmail",
    }
  ]
}
```



```

    sessionKey: "hook:gmail:{{messages[0].id}}",
    messageTemplate: "From: {{messages[0].from}}\nSubject: {{messages[0].sub,
    textTemplate: "{{messages[0].snippet}}",
    deliver: true,
    channel: "last",
    to: "+15555550123",
    thinking: "low",
    timeoutSeconds: 300,
    transform: {
      module: "gmail.js",
      export: "transformGmail",
    },
  },
],
gmail: {
  account: "openclaw@gmail.com",
  label: "INBOX",
  topic: "projects/<project-id>/topics/gog-gmail-watch",
  subscription: "gog-gmail-watch-push",
  pushToken: "shared-push-token",
  hookUrl: "http://127.0.0.1:18789/hooks/gmail",
  includeBody: true,
  maxBytes: 20000,
  renewEveryMinutes: 720,
  serve: { bind: "127.0.0.1", port: 8788, path: "/" },
  tailscale: { mode: "funnel", path: "/gmail-pubsub" },
},
},

// Gateway + networking
gateway: {
  mode: "local",
  port: 18789,
  bind: "loopback",
  controlUi: { enabled: true, basePath: "/openclaw" },
  auth: {
    mode: "token",
    token: "gateway-token",
    allowTailscale: true,
  },
},

```



```
tailscale: { mode: "serve", resetOnExit: false },
remote: { url: "ws://gateway.tailnet:18789", token: "remote-token" },
reload: { mode: "hybrid", debounceMs: 300 },
},
```

```
skills: {
  allowBundled: ["gemini", "peekaboo"],
  load: {
    extraDirs: ["~/Projects/agent-scripts/skills"],
  },
  install: {
    preferBrew: true,
    nodeManager: "npm",
  },
  entries: {
    "nano-banana-pro": {
      enabled: true,
      apiKey: "GEMINI_KEY_HERE",
      env: { GEMINI_API_KEY: "GEMINI_KEY_HERE" },
    },
    peekaboo: { enabled: true },
  },
},
}
```

Common patterns

Multi-platform setup



```
agent: { workspace: "~/.openclaw/workspace" },
channels: {
  whatsapp: { allowFrom: ["+15555550123"] },
  telegram: {
    enabled: true,
    botToken: "YOUR_TOKEN",
    allowFrom: ["123456789"],
  },
  discord: {
    enabled: true,
    token: "YOUR_TOKEN",
    dm: { allowFrom: ["yourname"] },
  },
},
}
```

Secure DM mode (shared inbox / multi-user DMs)

If more than one person can DM your bot (multiple entries in `allowFrom`, pairing approvals for multiple people, or `dmPolicy: "open"`), enable **secure DM mode** so DMs from different senders don't share one context by default:



```
// Secure DM mode (recommended for multi-user or sensitive DM agents)
session: { dmScope: "per-channel-peer" },
    >

channels: {
  // Example: WhatsApp multi-user inbox
  whatsapp: {
    dmPolicy: "allowList",
    allowFrom: ["+15555550123", "+15555550124"],
  },

  // Example: Discord multi-user inbox
  discord: {
    enabled: true,
    token: "YOUR_DISCORD_BOT_TOKEN",
    dm: { enabled: true, allowFrom: ["alice", "bob"] },
  },
},
}
```

OAuth with API key failover



```
auth: {  
  profiles: {  
    "anthropic:subscription": {  
      provider: "anthropic",  
      mode: "oauth",  
      email: "me@example.com",  
    },  
    "anthropic:api": {  
      provider: "anthropic",  
      mode: "api_key",  
    },  
  },  
  order: {  
    anthropic: ["anthropic:subscription", "anthropic:api"],  
  },  
},  
agent: {  
  workspace: "~/.openclaw/workspace",  
  model: {  
    primary: "anthropic/claude-sonnet-4-5",  
    fallbacks: ["anthropic/claude-opus-4-6"],  
  },  
},  
}
```

Anthropic subscription + API key, MiniMax fallback



```
auth: {
  profiles: {
    "anthropic:subscription": {
      provider: "anthropic",
      mode: "oauth",
      email: "user@example.com",
    },
    "anthropic:api": {
      provider: "anthropic",
      mode: "api_key",
    },
  },
  order: {
    anthropic: ["anthropic:subscription", "anthropic:api"],
  },
},
models: {
  providers: {
    minimax: {
      baseUrl: "https://api.minimax.io/anthropic",
      api: "anthropic-messages",
      apiKey: "${MINIMAX_API_KEY}",
    },
  },
},
agent: {
  workspace: "~/.openclaw/workspace",
  model: {
    primary: "anthropic/claude-opus-4-6",
    fallbacks: ["minimax/MiniMax-M2.1"],
  },
},
}
```

Work bot (restricted access)



```
identity: {  
  name: "WorkBot",  
  theme: "professional assistant",  
},  
agent: {  
  workspace: "~/work-openclaw",  
  elevated: { enabled: false },  
},  
channels: {  
  slack: {  
    enabled: true,  
    botToken: "xoxb-...",  
    channels: {  
      "#engineering": { allow: true, requireMention: true },  
      "#general": { allow: true, requireMention: true },  
    },  
  },  
},  
}
```

Local models only



```
agent: {
  workspace: "~/.openclaw/workspace",
  model: { primary: "lmstudio/minimax-m2.1-gs32" },
},
models: {
  mode: "merge",
  providers: {
    lmstudio: {
      baseUrl: "http://127.0.0.1:1234/v1",
      apiKey: "lmstudio",
      api: "openai-responses",
      models: [
        {
          id: "minimax-m2.1-gs32",
          name: "MiniMax M2.1 GS32",
          reasoning: false,
          input: ["text"],
          cost: { input: 0, output: 0, cacheRead: 0, cacheWrite: 0 },
          contextWindow: 196608,
          maxTokens: 8192,
        },
      ],
    },
  },
},
},
}
```

Tips

If you set `dmPolicy: "open"` , the matching `allowFrom` list must include `"*"` .

Provider IDs differ (phone numbers, user IDs, channel IDs). Use the provider docs to confirm the format.

Optional sections to add later: `web` , `browser` , `ui` , `discovery` , `canvasHost` , `talk` , `signal` , `imessage` .



See [Providers](#) and [Troubleshooting](#) for deeper setup notes.

[< Configuration Reference](#)

[>](#)

[Authentication >](#)

Powered by [mintlify](#)