CLAUDE CODE

Setup Reference Card

Anthropic's agentic command-line coding tool • Docs: docs.claude.com/claude-code

Claude Code is a terminal-based agent that reads and edits files, runs commands, and works directly inside your repository. It also ships IDE plugins for VS Code and JetBrains, and a desktop app if you prefer to skip the terminal.

1 · Install

The **native installer** is the recommended method on every platform. It does not require a Node.js installation, and it auto-updates in the background. Pick *one* method per machine — do not mix installers.

Platform	Command	Updates
macOS / Linux / WSL	curl -fsSL https://claude.ai/install.sh | bash	Automatic
Windows (PowerShell)	irm https://claude.ai/install.ps1 | iex	Automatic
macOS (Homebrew)	brew install --cask claude-code	brew upgrade
Windows (WinGet)	winget install Anthropic.ClaudeCode	winget upgrade

Legacy npm method (deprecated, requires Node.js 18+; the native installer is preferred):

```
npm install -g @anthropic-ai/claude-code
```

Verify the install — open a new terminal in your project, then:

```
claude --version      # confirm the binary is on your PATH
claude doctor         # diagnose install type, auth, PATH, MCP
claude                # start an interactive session
```

> **PATH tip:** the native installer places the binary at ~/.local/bin/claude. If claude isn't found, reopen your terminal or add that directory to your PATH.

2 · Connect a Model Provider —Model Access Gateway (MAG)

Step 1 – Get an American Science Cloud (AmSC) Account

Follow the instructions listed here: https://amsc-docs-d762d2.gitlab.io/GM-getting-started#access--login to create an AmSC account. Once the account is approved, proceed to the next step.

Step 2 — Request MAG access and generate a Personal Access Token (PAT)

Follow the instructions listed here: https://amsc-docs-d762d2.gitlab.io/model-access-gateway#quick-start

Step 3 — Set environment variables

Add these to your shell profile (~/.bash_profile, ~/.zshrc, or PowerShell $PROFILE). Match the model names to what your key provides.

```
# Core connection
export ANTHROPIC_AUTH_TOKEN=<your-MAG-PAT>
export ANTHROPIC_BASE_URL=https://i2-api.genesis.american-science-cloud.org/
export ANTHROPIC_MODEL=claude-opus-4-8
```

```
# Per-tier models  (for: claude --model <opus|sonnet|haiku>, plan mode)
export ANTHROPIC_DEFAULT_OPUS_MODEL=claude-opus-4-8
export ANTHROPIC_DEFAULT_SONNET_MODEL=claude-sonnet-4-6
export ANTHROPIC_DEFAULT_HAIKU_MODEL=claude-haiku-4-5

# Optional
export DISABLE_TELEMETRY=1
#export ANTHROPIC_LOG=debug      # uncomment for verbose logging
```

Step 3 — Run Claude `claude` in your project directory. Confirm the active model and provider ("API Usage Billing") in the welcome banner and via the `/status` slash command.

Other providers: Claude Code also supports Amazon Bedrock. See the AmSC documentation and Claude Code docs for Bedrock setup (enable models, configure AWS CLI, set provider env vars).

3 · Per-Project Settings

Drop a `.claude/settings.local.json` for machine-local overrides, and do not commit the file containing the secret token to a version-control system:

```
{
  "env": {
    "ANTHROPIC_BASE_URL": "https://i2-api.genesis.american-science-cloud.org/",
    "ANTHROPIC_AUTH_TOKEN": "<your-MAG-PAT>",
    "ANTHROPIC_MODEL": "claude-opus-4-8"
  }
}
```

4 · CLAUDE.md — Project Context

A `CLAUDE.md` file gives Claude Code standing context. It reads the file in the current directory *plus* every `CLAUDE.md` up the parent tree, and a global `~/.claude/CLAUDE.md`. Use it for stack notes, code-style rules, and common commands.

```
# ./CLAUDE.md  (project root)
## Technologies
- Web framework: FastAPI + Ariadne (GraphQL)
- Database: PostgreSQL + SQLAlchemy 2.0 with Alembic migrations

## Code Style
- Never use local imports.
- Type annotations required on all functions; 4-space indent, PEP 8.

## Testing
- Use model_utils.py for test data; aim for 80%+ coverage.
- Avoid excessive mocking; every test has a docstring.
```

Keep secrets out of CLAUDE.md. CLAUDE.md is context that often gets committed to version control. Put API keys and tokens in environment variables or `.claude/settings.local.json` (git-ignored) instead — never paste a live key into CLAUDE.md.

5 · Everyday Commands

Command	What it does
`claude`	Start an interactive session in the current directory
`claude "..."`	Run a one-off prompt, e.g. `claude "explain the auth flow"`

Command	What it does
claude --model <tier>	Pick a tier: opus / sonnet / haiku
claude mcp list	List configured MCP servers
claude update	Apply a pending update immediately
claude doctor	Diagnostics: install type, version, PATH, auth, MCP

Useful slash commands in the TUI:

/init	Generate a starter CLAUDE.md for the project (in-session)
/help	List slash commands; /status shows auth & limits