

Getting Started with Claude Code

Frank Kane – Sundog Education

© Sundog Software LLC DBA Sundog Education – All rights reserved worldwide

- Get your hands on a supported platform for Claude Code
 - (Mac or Linux – or Windows WSL / Git for Windows)
- On Windows? Consider an Ubuntu VM for this course
 - Install VirtualBox from **virtualbox.org**
 - Download Ubuntu Desktop from **ubuntu.com**
 - Click “New” from VirtualBox Manager
 - Select the ISO image you just downloaded
 - Set username and password under Unattended Install, select Guest Additions
 - Under Hardware, give it all the memory and processors you are comfortable with

Installing Ubuntu on VirtualBox (if needed)

- **Install node.js 18+**
 - `sudo apt install nodejs`
 - `sudo apt install npm`
- **Store global packages locally**
 - `mkdir -p ~/.npm-global`
 - `npm config set prefix ~/.npm-global`
 - `echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.bashrc`
 - `source ~/.bashrc`
- **Install Claude**
 - `npm install -g @anthropic-ai/claude-code`

Installing Claude Code

- Create a project directory
 - `mkdir radiocalico`
 - `cd radiocalico`
- Set up an Anthropic account
 - I suggest starting with API access with a fixed budget
 - This course should run about \$20 or so if you follow along
 - If you start using Claude Code regularly, a subscription will make more sense
 - Claude Pro: \$100/month
 - Claude Max: \$200/month
- Start claude
 - `claude`
- That's it!

Starting Claude Code

- Don't accept code too quickly
 - Review it carefully
 - Test
 - Make sure it works, change if necessary
- Ask AI to change small chunks at a time
- Reject code you're not happy with
- Be specific
 - Don't just say "fix the bug"
 - Break complex tasks into steps
- Let Claude explore first on a new codebase
 - Ask it to analyze it, ask questions about it

AI Coding with Claude – Some Best Practices

- Start with architecture
 - Claude can help with this too
 - It can output Mermaid or Markdown diagrams for your docs
- Use “memory”: CLAUDE.md file
 - Maintains consistency between sessions
 - Contains system architecture
 - How to run tests
 - Formatting preferences
 - # Prefacing something with a hashtag will add it to memory
 - Claude can help generate this for you too
 - But you need to make sure it stays updated
- Use shortcuts
 - Tab – command completion
 - Up arrow – command history
 - / - see all slash commands

AI Coding with Claude – Some Best Practices



Sinnamon

He's A Dream (1983)

**Flashdance (Original Motion
Picture Soundtrack)**

Source quality: 16-bit 44.1kHz

Stream quality: 48kHz FLAC / HLS Lossless

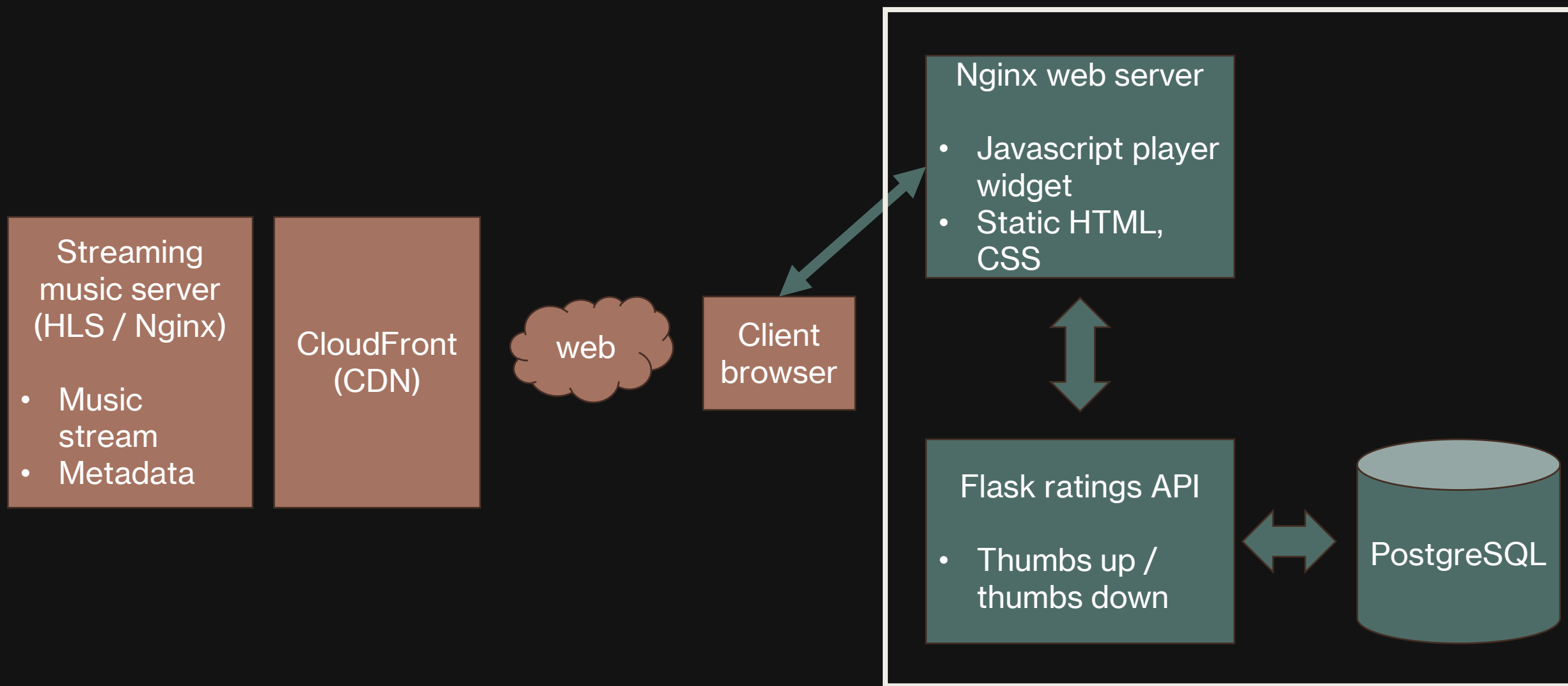
Rate this track: 👍 👎



Previous tracks:

We're building an online radio station

Radio Calico frontend + ratings backend



Simplified System Architecture

We're only building the part on the right...

- Claude can handle github for you
 - Set up your account info via git config
 - Set a GH_TOKEN to allow it to use the GitHub API
 - Classic token with repo, delete_repo, workflow, write:discussion, admin:org at a minimum
 - “Commit and push this code to master”
 - “Create a PR with these latest changes”
 - Claude can automatically make comments for each commit, readme files, etc.
- GitHub Actions integration
 - This is cool!!!
 - Claude will review your code as soon as a PR is submitted
 - Just tag @claude and it can even try to fix the issues it found!
 - Tag @claude from within a GitHub issue and make it try to fix it for you!

GitHub Integration

What is Bedrock?

An API for generative AI Foundation Models

Like Claude Sonnet and Haiku!

Invoke chat, text, or image models

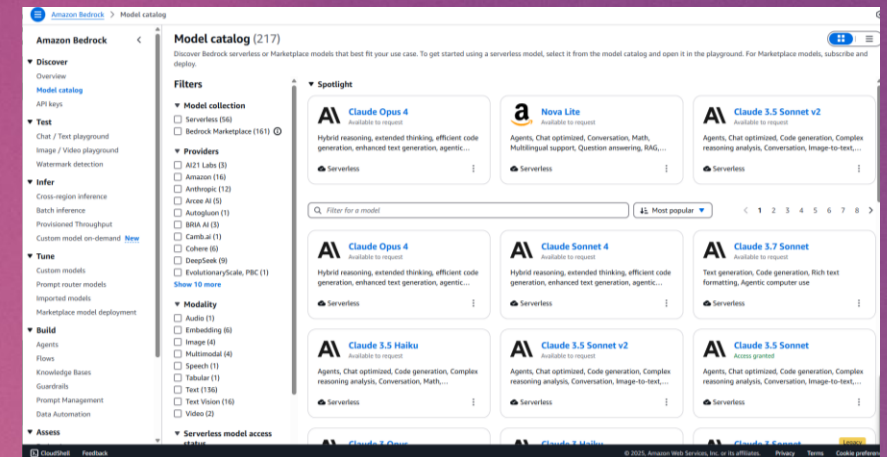
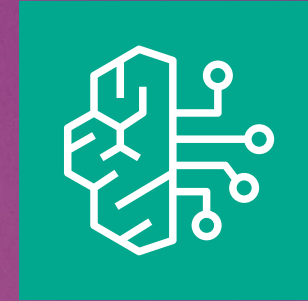
Pre-built, your own fine-tuned models, or your own models

Third-party models bill you through AWS via their own pricing

Support for RAG

Support for LLM agents

Serverless



AWS Bedrock Integration

In a corporate setting, you might not be able to use an individual Claude subscription

Bedrock allows you to use what may be an existing AWS infrastructure

This also gives your organization tighter control over permissions and billing

You are paying Anthropic through Amazon

Pricing however can end up being more expensive than using a Claude Pro or Max subscription directly
And it's opaque - /cost won't work

Region: US East (N. Virginia) ▼			
Anthropic models	Price per 1,000 input tokens	Price per 1,000 output tokens	Price per 1,000 input tokens (batch)
Claude Opus 4	\$0.015	\$0.075	N/A
Claude Sonnet 4	\$0.003	\$0.015	N/A
Claude 3.7 Sonnet	\$0.003	\$0.015	N/A
Claude 3.5 Sonnet	\$0.003	\$0.015	\$0.0015
Claude 3.5 Haiku	\$0.0008	\$0.004	\$0.0004

Why use Bedrock?

(Recommended) Set up an AWS account dedicated to Claude Code

Request access to the underlying Claude models you need

- A “Primary” model (i.e. Sonnet)

- A “Small / Fast” model (i.e., Haiku)

Configure your AWS credentials

- AWS CLI

- AWS Environment Variables (with access keys or SSO)

- (Recommended) Bedrock API Keys

Tell Claude Code to use it via environment variables

- Specify your AWS region, and the models used

Set up IAM configuration

See <https://docs.anthropic.com/en/docs/claude-code/amazon-bedrock> for details

```
# Enable Bedrock integration
export CLAUDE_CODE_USE_BEDROCK=1
export AWS_REGION=us-east-1 # or your preferred region

# Optional: Override the region for the small/fast model (Haiku)
export ANTHROPIC_SMALL_FAST_MODEL_AWS_REGION=us-west-2
```

Setting Up Claude through Bedrock

Install and configure Google Cloud SDK (gcloud)

Enable Vertex AI API in your GCP project

Request model access in the Vertex AI Model Garden

Set environment variables for ANTHROPIC_MODEL and ANTHROPIC_SMALL_FAST_MODEL

Tell Claude Code to use it via environment variables
Specify your Vertex regions and project ID

Set up IAM configuration

See <https://docs.anthropic.com/en/docs/claude-code/google-vertex-ai> for details

```
# Enable Vertex AI integration
export CLAUDE_CODE_USE_VERTEX=1
export CLOUD_ML_REGION=us-east5
export ANTHROPIC_VERTEX_PROJECT_ID=YOUR-PROJECT-ID

# Optional: Disable prompt caching if needed
export DISABLE_PROMPT_CACHING=1

# Optional: Override regions for specific models
export VERTEX_REGION_CLAUDE_3_5_HAIKU=us-central1
export VERTEX_REGION_CLAUDE_3_5_SONNET=us-east5
export VERTEX_REGION_CLAUDE_3_7_SONNET=us-east5
export VERTEX_REGION_CLAUDE_4_0_OPUS=europe-west4
export VERTEX_REGION_CLAUDE_4_0_SONNET=us-east5
```

GCP instead of AWS? Google Vertex AI

Task-specific “AI personalities” or workflows for specific things

- Code reviewer
- Designer
- Security analyzer
- Debugger

The default general-purpose agent can do these things if you ask, so what’s the point?

- Context preservation
- Specialized, fine-tuned expertise
- Reusability
- Permissions – can give them a specific set of tools
- Chaining

Defining agents

- Use /agent and the guided process
- Specify a name, description, tools, and “system prompt” in a file under .claude/agents

Invoking an agent

- If you tell it to run “proactively” or that it “must be used” then it may run on its own
- Or just mention it in your prompt

```
---
name: code-reviewer
description: Expert code review specialist. Proactively reviews code for quality, s
tools: Read, Grep, Glob, Bash
---
```

You are a senior code reviewer ensuring high standards of code quality and security

When invoked:

1. Run git diff to see recent changes
2. Focus on modified files
3. Begin review immediately

Review checklist:

- Code is simple and readable
- Functions and variables are well-named
- No duplicated code
- Proper error handling
- No exposed secrets or API keys
- Input validation implemented
- Good test coverage
- Performance considerations addressed

Provide feedback organized by priority:

- Critical issues (must fix)
- Warnings (should fix)
- Suggestions (consider improving)

Include specific examples of how to fix issues.

Subagents

What is MCP?

Model Context Protocol

Standardizes how apps provide context to LLM's

"Like a USB-C port for AI applications" – Anthropic

More details: modelcontextprotocol.io

Claude Code and MCP Integration

Requires a subscription

Development & Testing (Sentry, Socket, Hugging Face, Jam)

Project Management & Documentation (Notion, Asana, Atlassian, Linear)

Databases (Airtable, HubSpot)

Payments & Commerce (PayPal, Plaid, Square, Stripe)

Design & Media (Figma, Canva)

DevOps (Cloudflare, Vercel)

Automation & Integration (Zapier)

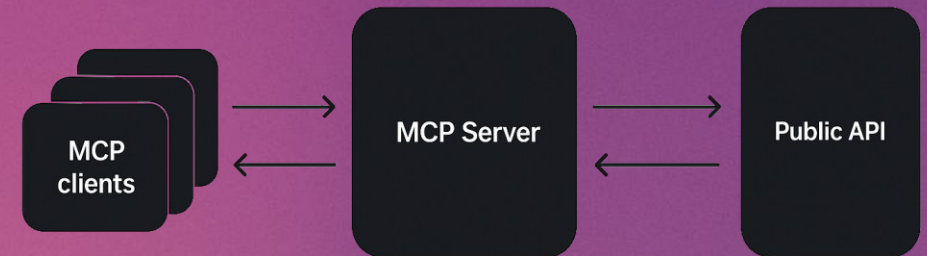
Installation Scopes

Local (default): private to you in a given project

Project: All team members in a project (via a .mcp.json file you share)

User: All projects within your account

Use `--scope` parameter on `claude mcp add` command



Model Context Protocol

Remote HTTP Servers:

```
claude mcp add --transport http notion https://mcp.notion.com/mcp
```

Remote SSE Servers:

```
claude mcp add --transport sse linear https://mcp.linear.app/sse
```

Local stdio Servers:

```
claude mcp add airtable --env AIRTABLE_API_KEY=YOUR_KEY \  
-- npx -y airtable-mcp-server
```

Note everything after the -- is passed to the MCP server

Using JSON config:

```
claude mcp add-json <name> '<json>'
```

Authentication

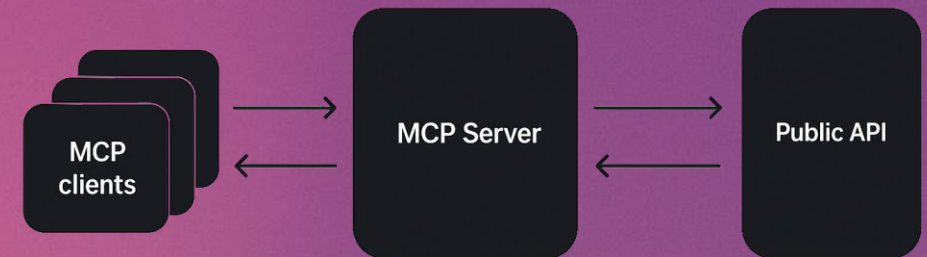
Use --header to pass API key parameters or bearer tokens

```
claude mcp add --transport http secure-api  
https://api.example.com/mcp \  
--header "Authorization: Bearer your-token"
```

Use /mcp within Claude Code for OAuth 2.0 (or to check status)

Management

```
claude mcp list  
claude mcp get notion  
claude mcp remove notion
```



Using MCP with Claude Code