

Force **Multiplying** Our People

Welcome to the Agentic Age. No question is bad. No topic is off limits. We are here for you — to help every engineer ship faster, think bigger, and build with confidence.



OPEN DOOR

Every question welcome,
every skill level valued



ACCELERATE

10x your impact with agentic
workflows



LEARN TOGETHER

Share patterns, avoid pitfalls,
level up as a team



SHIP IT

Real results, real code, real
production impact

What's New with Claude Code



New Commands

/goal: Claude works across turns until a condition is met

/scroll-speed: tune scroll rate with live preview

/mcp reconnect picks up **.mcp.json** edits without restart



MCP & Hooks

alwaysLoad config keeps MCP tools always available

Stdio servers receive

CLAUDE_PROJECT_DIR in env

continueOnBlock: PostToolUse feeds rejection back to Claude



Bedrock Fixes

ANTHROPIC_BEDROCK_SERVICE_TIER:

flex/priority support

Fixed Bedrock 400 with

ENABLE_PROMPT_CACHING_1H

Fixed multi-GB memory growth from image-heavy sessions

Tame Permission Prompts with **settings.json**

Claude asking if it's okay too much?

Ask it to help you with your `settings.json`



What You Control

permissions.allow/deny — what Claude does without asking

defaultMode — default, acceptEdits, or plan

env — inject AWS region, Node env, proxy



Three Layers

`~/.claude/settings.json` — your global defaults

`.claude/settings.json` — per-project rules (commit it)

`.claude/settings.local.json` — personal overrides



Quick Wins

/update-config — describe what you want in plain English

/fewer-permission-prompts — Claude reads your history and suggests the right allow rules

New Claude Code Teams Hub

As Claude Code moves to GA, we're consolidating support into a dedicated **Claude Code** team on Microsoft Teams. The POC chat will be shutting down — please join the new channels below. Search "**Claude Code**" in Teams to find the team.



Claude Code CoE

One stop shop for all things Ally Claude Code. Links to resources, process guides, and support channels.

Main channel



Marketplace

For MCP integrations, skills, and agents. Plugin questions, installation help, and contribution guidance.

Recommended



MCP

Information and discussion specific to MCP servers — building, hosting, and connecting them.



Platform Support

Installation support, troubleshooting questions, and incident tickets. Your first stop for issues.



Training Tips

Techniques, prompts, and patterns to get the most out of Claude Code. Learn from each other.

Recommended



Watercooler

General chat — migrating the Claude Code Lobby from Slack to our official community. All topics welcome.

Recommended



Champions Channel

Private sub-channel reserved for Claude Code Champions program members.

Private • By invitation

Observability Terrace



Topic One

Description of monitoring patterns, dashboards, or alerting strategies.



Topic Two

Description of logging best practices, tracing, or SLO updates.



Topic Three

Description of debugging workflows, incident analysis, or tooling tips.



Your Toolkit

allyai

Checks that your graph actually compiles, your edges go somewhere real, and your model integration won't blow up in production. Scaffolds new tools with tests so you don't ship untested code. Includes 12 labs if you want to learn LangGraph properly.

[/allyai:learn](#) • [/allyai:scaffold](#) • [/allyai:validate](#) • [/allyai:visualize](#) • [/allyai:deploy](#)

GenAI Platform & Enablement — Adric Saxon • v0.3.0

allyai + forge

Everything above, plus forge handles the rest of your workflow. You describe a feature, it plans, implements, then throws 14 reviewers at your diff—none of whom have seen your code before. They find the stuff you stopped noticing three commits ago.

[/explore](#) → [/plan](#) → [/implement](#) → [/review](#) → [/ship](#)

```
claude plugin marketplace add ally-claude-plugins-official
claude plugin install allyai — optionally add forge
```

Experimental — break things — tell us what sucks: [#genai-platform-enablement](#)

From Claude Code Skill to **LangGraph Agent**

You built a skill that works great in Claude Code — now what? Adric Saxon walks us through the full journey of taking an interactive skill and turning it into a fully autonomous, productionized agent on the Ally AI Platform.



Start with a Skill

Claude Code skills are great for **prototyping logic** — they give you fast iteration, tool access, and a human in the loop. But they require a developer at the keyboard.



Extract & Formalize

Identify the **decision graph** inside your skill. Map prompts to nodes, tool calls to edges, and human checkpoints to conditional branches — this is your LangGraph blueprint.



Deploy & Scale

Package the agent for the **Ally AI Platform**. Add observability, error recovery, and scheduling — now it runs autonomously without anyone at the keyboard.

Presented by Adric Saxon — GenAI Platform & Enablement

— WHAT'S NEXT

Next Session



Upcoming Topic

Brief description of what the next COE session will cover.



Requests Welcome

Have a topic you'd like covered? A tool you want demonstrated? Let us know.

The COE is yours. Bring questions, bring challenges, bring ideas. We build together.