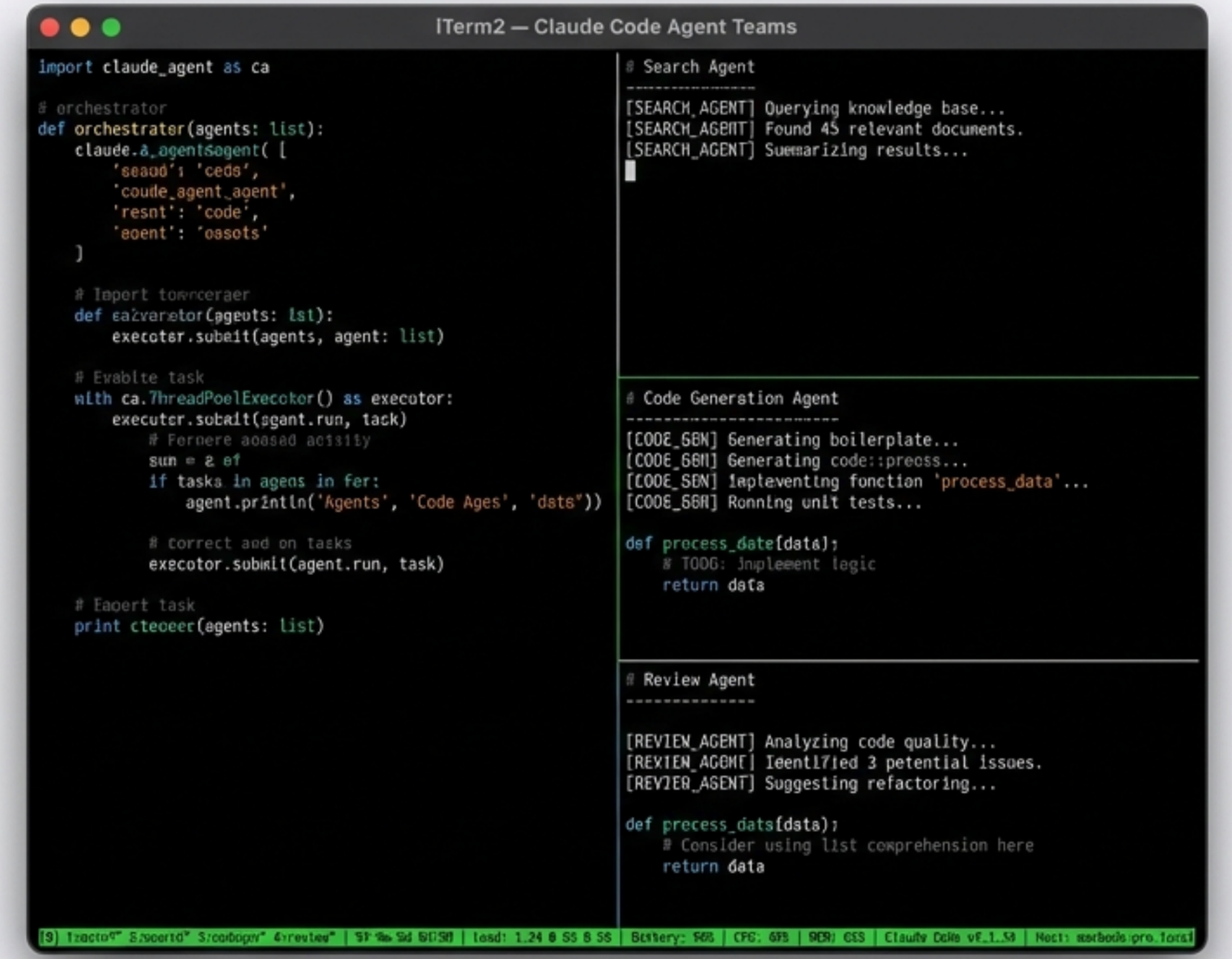
Claude Code Agent Teams

The Orchestration Guide: From Single-Threaded Coding to Multi-Agent Parallelism

OS REQUIREMENT:	macOS (Sequoia recommended)
CORE STACK:	iTerm2 + tmux + Claude Code (v2.1.34+)
BASED ON:	Agent Teams Setup Guide by Charlie (charlieautomates)



The Paradigm Shift: Lead vs. Worker

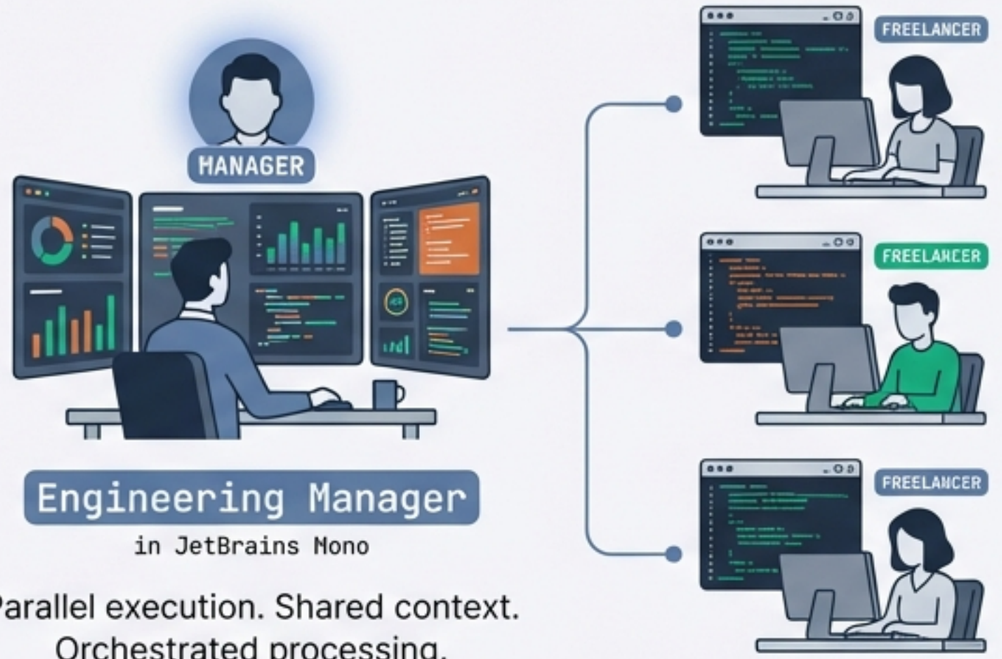
Old Model

Solo Coder

in JetBrains Mono

Linear execution. Single context.
Serial processing.

New Model



Engineering Manager

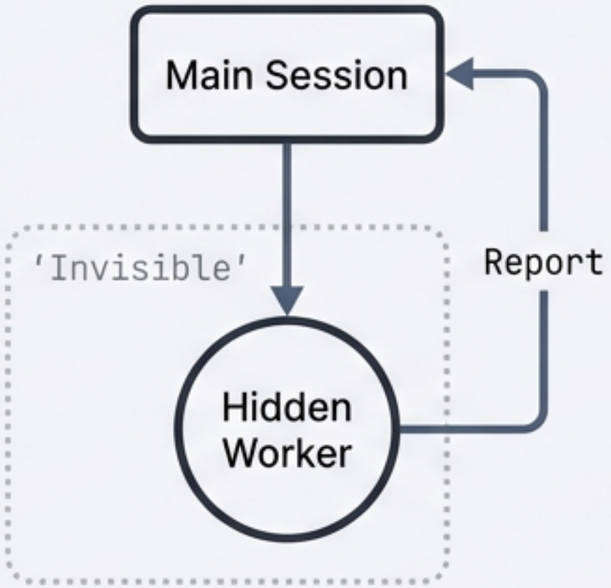
in JetBrains Mono

Parallel execution. Shared context.
Orchestrated processing.

- **Shared Context**: Agents talk to each other.
- **Visibility**: Watch every keystroke in real-time.
- **Intervention**: Redirect teammates mid-task.

Architecture: Subagents vs. Agent Teams

Subagents (The Invisible Workers)



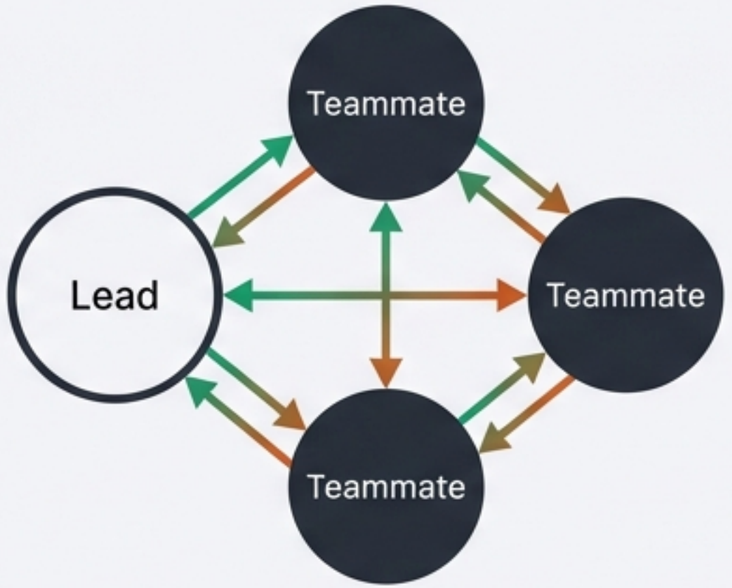
Behavior: Sequential, hidden execution.

Communication: Report-only. No peer-to-peer.

Cost: ~1.5x single session.

Best For: Simple retrieval, file search, single-file reading.

Agent Teams (The Collaborative Crew)



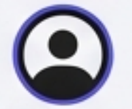
Behavior: Parallel, visible execution (split panes).

Communication: Full mesh (Lead ↔ Team, Team ↔ Team).

Cost: ~3-4x single session per teammate.

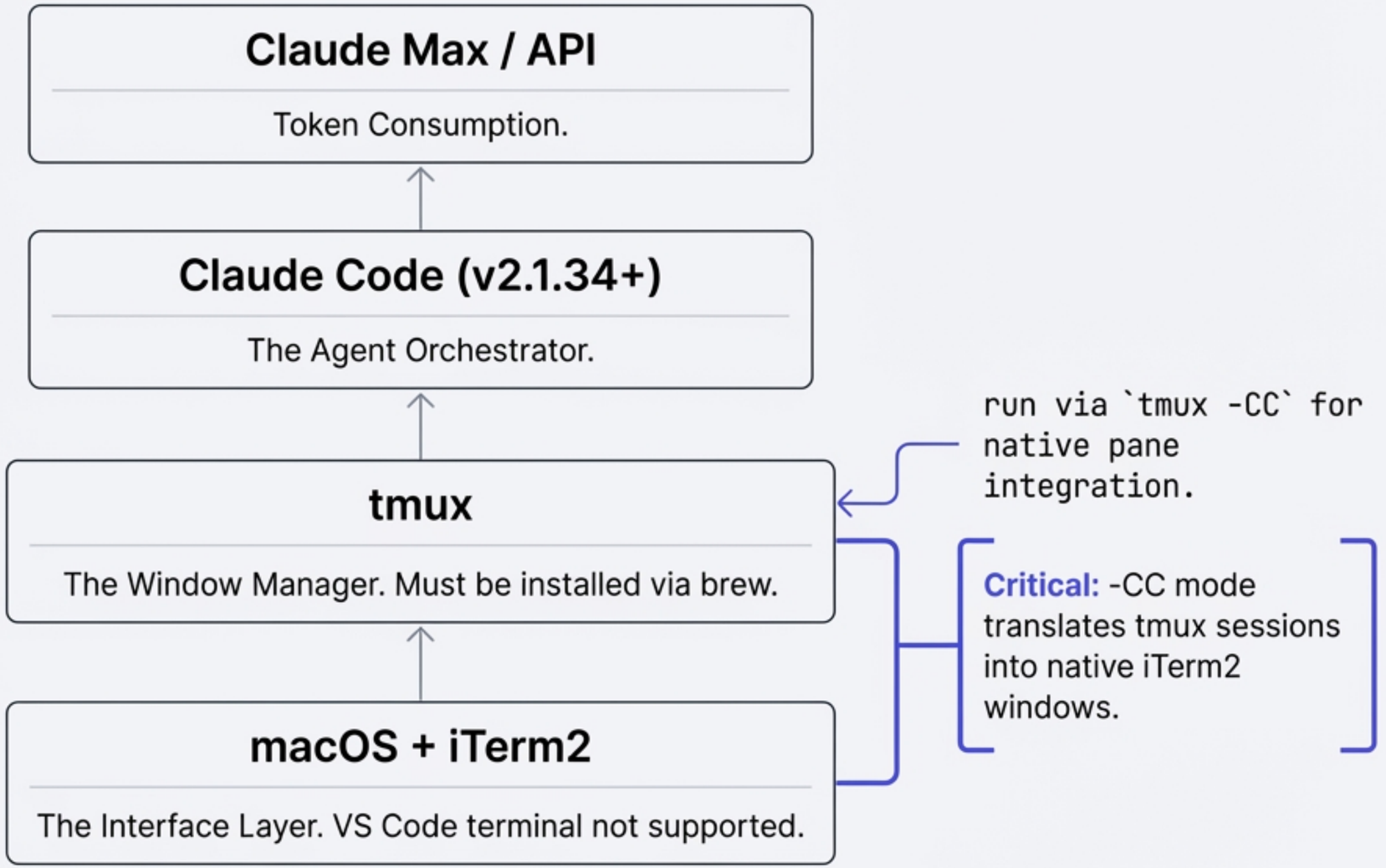
Best For: Complex coordination, debugging, multi-file builds.

Decision Matrix: When to Scale Up

Scenario	Result & Architecture
Just search these files and tell me the answer. Inter Regular	→ **Use Subagent**  Subagent
Build 3 features at once, and let me watch the code.	→ **Use Agent Team**  Agent Team
Quick research from multiple angles. Inter Regular	→ **Use Subagent**  Subagent
Review PR: Security + Performance + Tests. Inter Regular	→ **Use Agent Team**  Agent Team
Debug a bug with competing theories. Inter Regular	→ **Use Agent Team**  Agent Team

Key Takeaway: If the *process* matters as much as the *answer*, or if tasks interact, use **Agent Teams**.

The Technical Stack



Step 1: The Feature Flag

Context

The ‘Teammate Tool’ is experimental. It must be explicitly enabled in the environment *before* the process starts.

⚠

If this variable is missing, Claude **silently**, Claude **silently** falls back to **Subagents**.

Code

Crucial for process initialization. →

~/.zshrc

```
export CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1
```

```
> source ~/.zshrc
```

Step 2: Project Configuration

.claude/settings.local.json

```
{
  "env": { "CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS": "1" },
  "teammateMode": "tmux"
}
```

Teammate Modes Explained

- **“auto”**: Default. Risky if tmux isn't detected.
- **“in-process”**: Stacks agents in one window. Hard to navigate.
- "**tmux**": **Recommended.** Forces native split-panes.

Pro Tip Box

If schema validation fails, use a Python script to inject the key (see notes).

Step 3 & 4: The Launch Script & Alias

The Script (~/.claude/launch-team.sh)

~/.claude/launch-team.sh

```
#!/bin/zsh
export CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1
exec claude --dangerously-skip-permissions --chrome
```

← Ensures environment inheritance.

The Alias (~/.zshrc)

~/.zshrc

```
alias cddi='tmux kill-server 2>/dev/null; tmux -CC new-session
   -s claude-team "$HOME/.claude/launch-team.sh"'
```

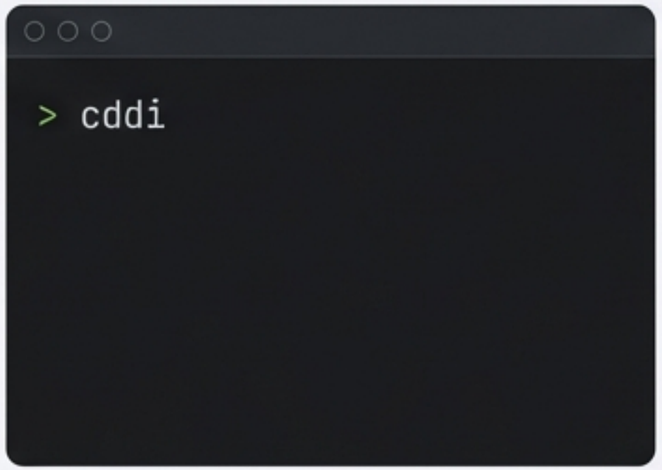
Prevents zombie sessions.

Triggers native iTerm2 integration.

Starts fresh environment.

Launching & Spawning

Step 1
Trigger



Step 2
Spawn Prompt

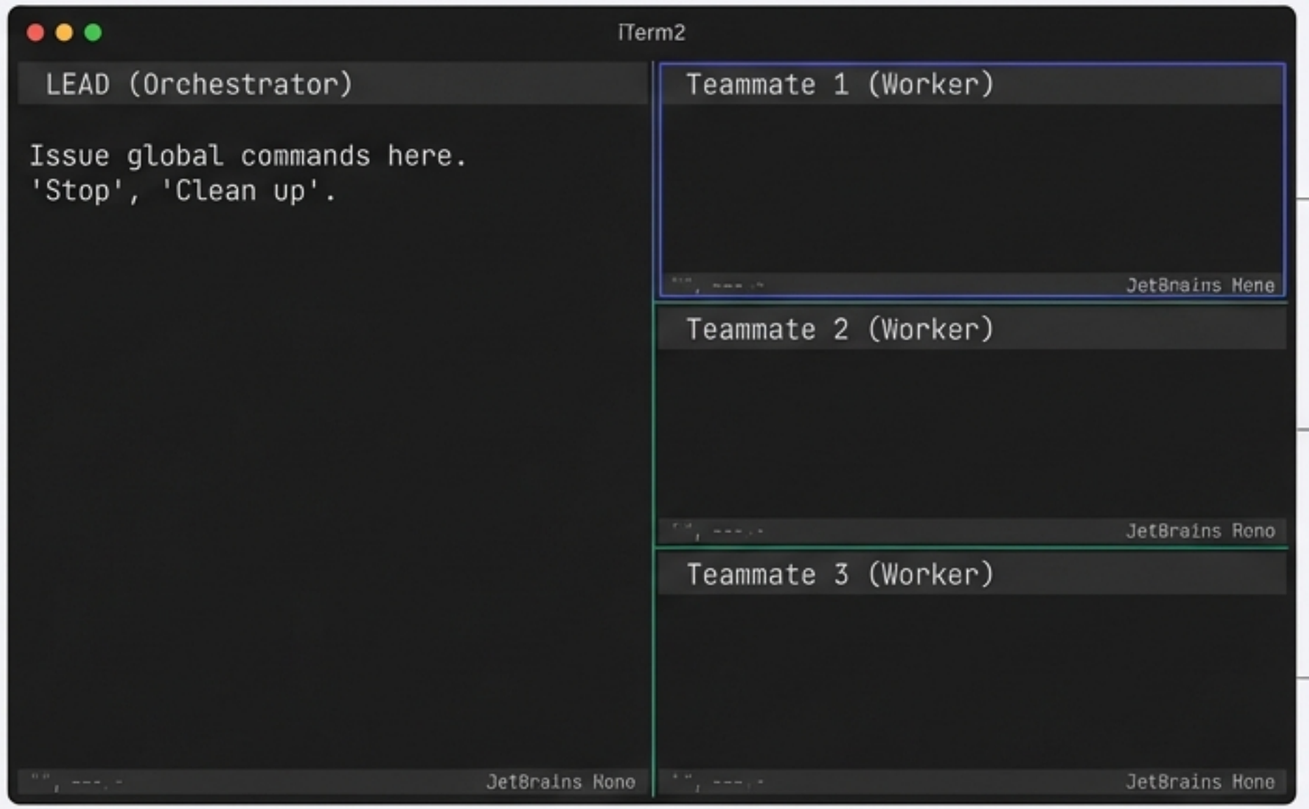
Claude

Create a team called 'my-team' with 3 teammates:
One for frontend review, one for backend API, one for tests.

Step 3
Automation Sequence

- Lead creates team structure.
- Shared task list generated.
- Teammates spawn in new panes.
- Tasks assigned.
- Coordination begins.

The Command Center Interface



Click any pane to focus and message directly.

Fallback Navigation (In-Process Mode)

Shift+Down	Enter	Escape
Next Teammate	View Session	Interrupt Turn

Operational Modes & Control

Delegate Mode

Shift+Tab

Prevents the Lead agent from coding. Restricts Lead to pure coordination and delegation.

Shutdown Protocol

Always clean up through the Lead.

```
> Shut down all teammates
  and clean up.
```

Monitoring

Ctrl+T

Toggle visibility of the shared task list (in-process mode).

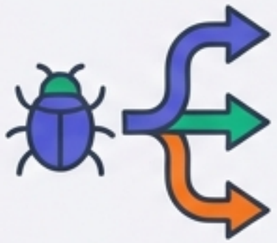
Real-World Workflow Patterns

The Parallel Review

Assign specific lenses to a single PR.

- Security
- Performance
- Test Coverage

Competing Theories



Isolate root causes faster.

- Agent A: `Check DB pooling`
- Agent B: `Check logs`
- Agent C: `Load test`

Agents debate findings.

The Assembly Line

Building independent features.

- `Phase 2 execution`
- `Phase 3 execution`

Must: Work on different files/branches.

Best Practices for Stability

	1. Task Sizing	Sweet spot: 5-6 tasks per teammate. • Too small = overhead. Too large = drift.
	2. Conflict Avoidance	Assign by directory path to prevent overwrites. Example: Frontend (`src/components`) vs Backend (`src/api`).
	3. Context Injection	Teammates have amnesia. • Must explicitly include context in spawn prompt. Example: Mention JWT usage, specific libraries, or architectural patterns.
	4. Start Simple	Begin with read-only tasks (audits/research) before code generation.

Troubleshooting Common Blockers

I don’t have a Teammate tool / Claude uses Subagents

Cause:
Env var missing at startup.

Fix:
Check `~/.zshrc`, restart iTerm2, verify `echo $CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS`.

Duplicate session: claude-team

Cause:
Old tmux session lingering.

Fix:
Run `tmux kill-server`.

Split Panes Not Appearing

Cause:
Running in-process mode.

Fix:
Verify `teammateMode: 'tmux'` in settings.
Check `echo $TMUX`.

Quick Reference Card

One-Time Setup

1. Export Feature Flag (`.zshrc`)
2. Set `teammateMode' (`settings.local.json`)
3. Create Launch Script & Alias

→ Daily Routine

`cddi` → “Create team...” → Work → “`Clean up`”

Shortcuts

`Shift+Up/Down`	Cycle Teammates
`Shift+Tab`	Toggle Delegate Mode

⚠ Emergency / Nuclear

`tmux kill-server`

Kills everything if sessions get stuck.