Claude Code
Features Guide

Part 7 of the Claude Code Series

Claude Code Background Tasks: How AI Agents Work While You Sleep

Learn how Claude Code background tasks let AI agents run builds, audits, and research autonomously. Discover how we use async agents to manage multiple clients simultaneously.

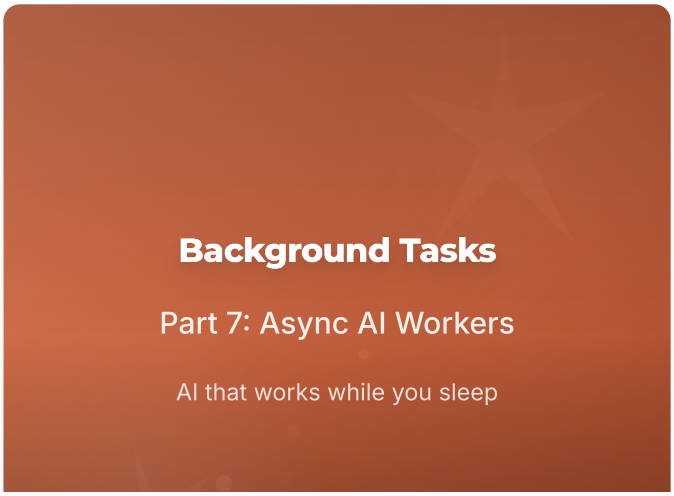
9 February 2026 · 11 min read

What if your AI development assistant could keep working after you walked away from your desk?

That is exactly what background tasks in Claude Code deliver. Instead of waiting for every operation to finish before moving on, you can fire off long-running processes, continue with other work, and get notified when results are ready.

For agencies managing multiple clients, this changes the game entirely. We can run a full site audit for one client while actively building out content for another, and have a third client's deployment monitored in the background. No idle time. No bottlenecks.

This guide explains how background tasks work, how we use them at our agency, and why they represent a shift towards truly autonomous AI workflows.



What Are Background Tasks in Claude Code?

Background tasks are long-running processes that execute independently of your current conversation with Claude Code. Think of them as instructions you hand off to an assistant who goes and completes the work in another room, then comes back to tell you the results.

In practical terms, this means operations like:

- **Running test suites** across large codebases
- **Building and deploying** applications
- **Monitoring dev servers** and log output
- **Executing complex data processing** scripts
- **Running comprehensive site audits**

Previously, each of these operations would block your workflow. You would start a build, wait for it to finish, review the output, then move to the next task. Background tasks eliminate that waiting.

How They Work

The mechanism is straightforward:

1. **You initiate a task:** Tell Claude Code to run a process in the background
2. **Claude fires it off:** The process starts executing independently
3. **You continue working:** Switch to other tasks, other projects, other conversations
4. **You get notified:** When the background task completes (or fails), Claude surfaces the results
5. **You review and act:** Pick up where the background task left off

Claude Code manages the process lifecycle for you. It monitors stdout and stderr, watches for completion or failure signals, and can even search through logs to diagnose issues when something breaks.

The 24-Hour Work Cycle

Real-Time Log Monitoring and Error Recovery

One of the most practical features of background tasks is real-time log monitoring. When you have a dev server or build process running in the background, Claude Code does not just start it and forget about it.

Watching for Problems

Claude actively monitors the output streams. If a build fails with an error, Claude can:

- **Identify the error** in the log output
- **Search through related files** to understand the root cause
- **Suggest or apply a fix** based on the error context
- **Re-run the process** to verify the fix worked

This is especially valuable for deployment pipelines. Instead of babysitting a deployment and manually scanning logs for issues, you can let Claude watch the process and alert you only when intervention is needed.

Practical Example

Imagine you are deploying a Next.js application. You kick off the build in the background and switch to writing documentation. Five minutes later, Claude surfaces a message: the build failed because a dependency version conflict was introduced in the last commit. Claude has already identified the conflicting packages and suggests pinning the version. You approve the fix, Claude applies it, and restarts the build, all while you never left your documentation work.

Background Tasks: Async AI Workers

Launch a task, pre-authorise the tools it needs, and walk away. Claude runs autonomously and reports back when it's done — hours later if needed.

Background Agents: Autonomous AI Workers

Background tasks become truly powerful when combined with Claude Code's agent capabilities. A background agent is not just a process runner; it is a fully autonomous AI worker that can reason, make decisions, and complete multi-step tasks independently.

What Background Agents Can Do

When you fire off a background agent, it operates with the same capabilities as an interactive Claude Code session:

- **Read and write files** across the codebase
- **Run shell commands** and interpret output
- **Make multi-step decisions** based on intermediate results
- **Create and modify code** following project conventions
- **Search through documentation** and codebases for context

The difference is that all of this happens without requiring your attention. You set the objective, the agent works towards it, and you review the deliverable when it is done.

Enhanced Task Management

Background agents support persistent task tracking that survives across sessions:

- **Task persistence:** A task started on Monday afternoon can still be tracked on Tuesday morning
- **Dependencies:** Define which tasks must complete before others can begin
- **Cross-session coordination:** Multiple agents can work on related tasks with awareness of each other's progress

This is not theoretical. Our agency runs exactly this kind of workflow daily, which brings us to how we actually use this in practice.

24/7

Autonomous Execution

Background tasks run independently of your terminal session. Codebase health checks, content audits, and data analysis — all happening while you do other work.

How We Use Background Tasks at Our Agency

At Jordan James Media, background tasks have fundamentally changed how we handle multi-client workloads. Here is what a typical day looks like.

Multi-Client Workflow

We manage digital marketing for multiple clients simultaneously. Before background tasks, this meant context-switching between projects constantly, losing focus, and creating bottlenecks where one client's work had to wait while another's was actively being handled.

Now, the workflow looks like this:

Morning:

- Fire off a comprehensive SEO audit for Client A as a background task
- Start actively building out a new landing page for Client B
- Queue a competitor analysis report for Client C in the background

While working on Client B's landing page:

- Client A's SEO audit completes - results ready for review
- Client C's competitor analysis is 60% through data collection
- We continue focused work on Client B without interruption

After lunch:

- Review Client A's audit findings, create action items
- Client C's analysis is complete - review and format for presentation
- Start background deployment of Client B's landing page
- Begin active work on Client A's priority fixes

This is not multitasking in the human sense, where everything gets diluted. Each background agent gives its task full attention. We are simply orchestrating when we review and direct the work.

Case Study: The 20-Microsite Audit

We recently needed to audit over 20 location-specific microsites for a local plumbing franchise. Each site needed technical SEO checks, content quality assessment, schema validation, and performance benchmarking.

Without background tasks, this would have been days of repetitive work: open site, run checks, document findings,

move to next site.

Instead, we structured it as a team operation. The team lead agent coordinated four specialist agents, each running as a background task:

- **Agent 1:** Technical SEO crawl across all 20+ sites
- **Agent 2:** Content quality and keyword gap analysis
- **Agent 3:** Schema markup validation and local SEO checks
- **Agent 4:** Core Web Vitals and performance benchmarking

The team lead monitored progress while the four agents worked independently. Each agent reported back with structured findings. The entire audit, which would have taken days of manual work, was completed in a fraction of the time.

This is a real example of how Claude Code agent teams and background tasks work together. The agents were not sitting in a queue waiting for their turn. They were all working simultaneously, in the background, while the orchestrating agent managed the overall workflow.

Running Audits While Working on Content

Here is a pattern we use regularly:

1. **Start an SEO audit** as a background task against a client's site
2. **While it runs,** write blog content or build out landing pages for the same client
3. **When the audit finishes,** review findings and immediately apply fixes to the content you are already working on

The audit informs the content work in near real-time. No waiting. No separate "audit day" versus "content day." Everything flows together.

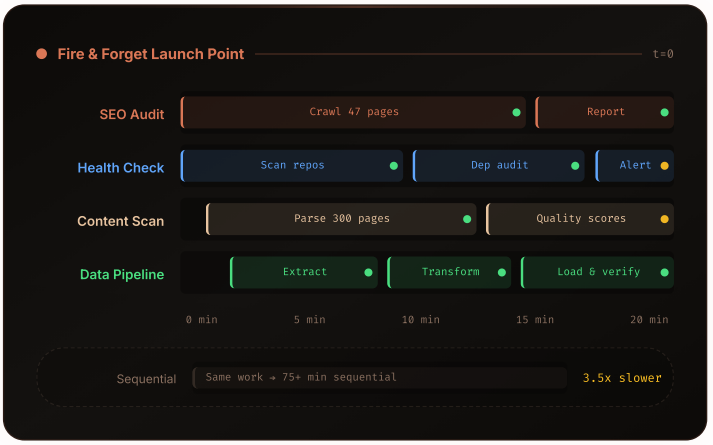
Deploy and Monitor

Deployments are another natural fit:

1. Push changes to staging
2. Fire off a background task to build, deploy, and run smoke tests
3. Continue working on the next feature or client project
4. Get notified when deployment succeeds (or fails with specific error details)

We have caught deployment issues within minutes that previously would have sat unnoticed until someone manually checked the staging environment.

Parallel Task Execution



The "While You Sleep" Workflow

This is where background tasks start to feel genuinely transformative.

End-of-Day Setup

Before wrapping up for the day, we set up background agents with clear objectives:

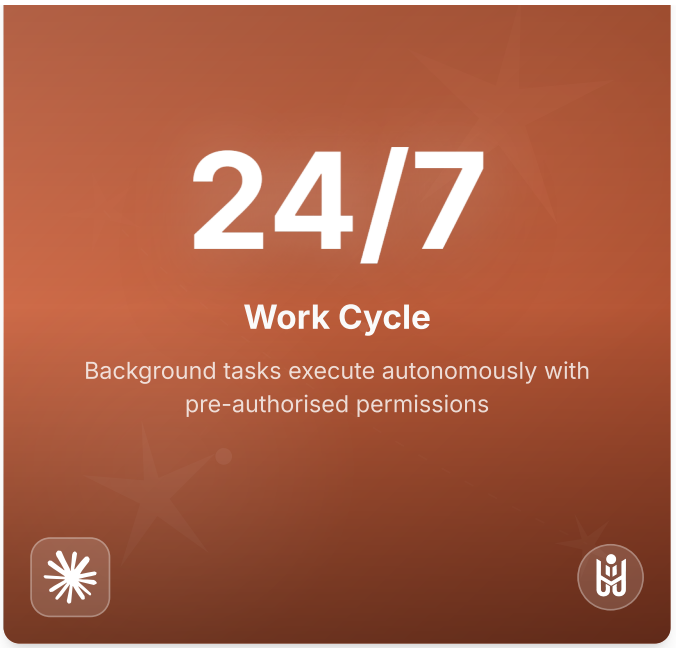
- **Research tasks:** "Analyse the top 10 ranking pages for [keyword], document their content structure, word count, and key topics covered"
- **Data processing:** "Process this month's analytics data across all client sites and flag any significant traffic changes"
- **Content drafts:** "Generate first drafts for next week's blog posts based on the content calendar"
- **Monitoring:** "Watch the deployment pipeline overnight and log any errors"

Morning Review

The next morning, the deliverables are waiting:

- Competitor research is structured and ready to inform content strategy
- Analytics summaries highlight which clients need attention
- Content drafts are ready for human review and refinement
- Overnight deployments have been monitored and any issues are documented

This is not a theoretical future. This is how we operate today using Claude Code's background task capabilities. The agents do the heavy lifting while the team sleeps, and we review and refine their output during business hours.



Background Tasks and the Future of AI Agencies

Background tasks are one piece of a larger shift. Combined with agent teams for parallel coordination, plan mode for strategic thinking, and MCP integration for connecting to external tools, they paint a picture of where AI-powered agencies are heading.

Towards 24/7 Operations

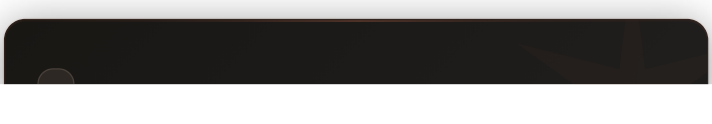
Traditional agencies operate during business hours. AI-augmented agencies can run critical processes around the clock:

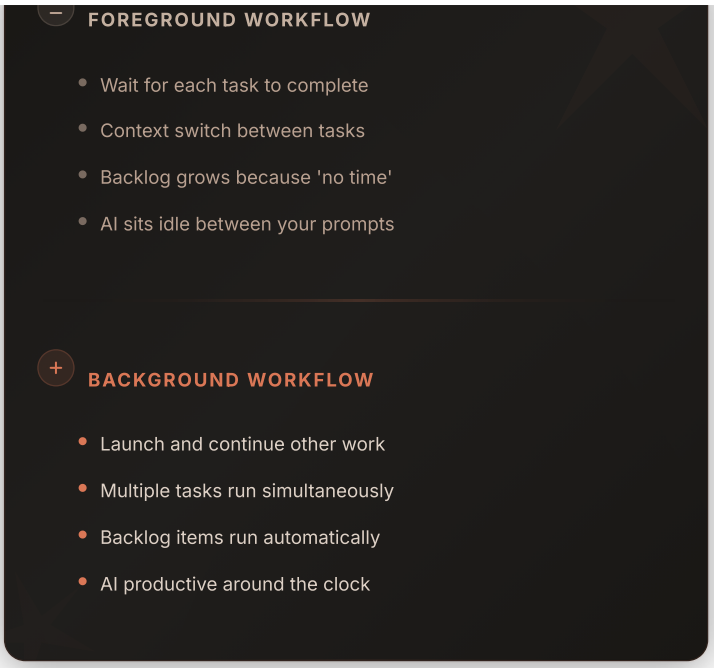
- **Continuous monitoring:** Site health, search rankings, ad performance
- **Overnight processing:** Large-scale data analysis, report generation
- **Proactive maintenance:** Catching and flagging issues before clients notice them
- **Global coverage:** Serving clients across time zones without staffing constraints

The Human-AI Balance

Background tasks do not remove humans from the equation. They remove humans from the *waiting* equation. Instead of sitting idle while a process runs, you direct your attention where it creates the most value: strategy, client relationships, creative direction, and quality review.

The AI handles execution at scale. Humans handle judgement and direction. Background tasks are the mechanism that makes this division of labour practical rather than theoretical.





Getting Started with Background Tasks

If you are using Claude Code and want to start leveraging background tasks:

Start Simple

- Run your test suite in the background while you write code
- Deploy to staging in the background while you update documentation
- Monitor build logs while working on the next feature

Scale Up

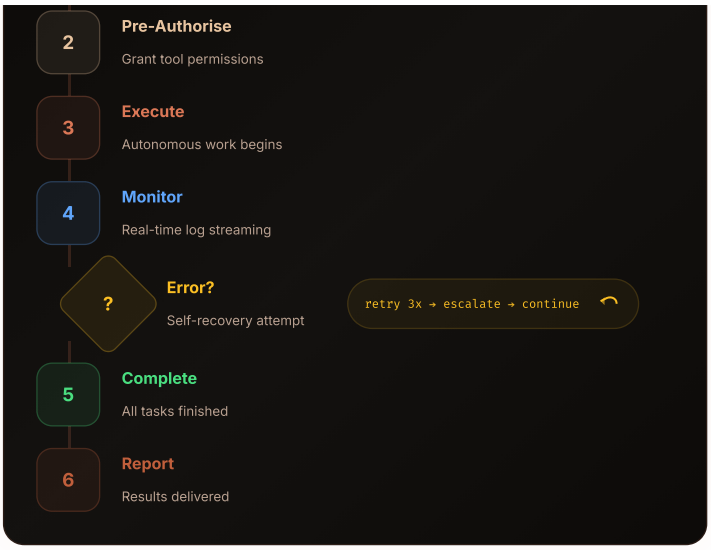
- Use background agents for research and analysis tasks
- Queue multiple independent tasks and review results as they complete
- Set up end-of-day background workflows for overnight processing

Go Full Orchestration

- Combine background tasks with agent teams for parallel work
- Use custom subagents specialised for specific background roles
- Integrate with external tools via MCP for comprehensive automation

---

Background Task Lifecycle



Key Takeaways

1. **Background tasks run independently** of your current Claude Code session, eliminating idle waiting time
2. **Real-time log monitoring** means Claude catches and can fix errors without your intervention
3. **Background agents are fully autonomous,** capable of reading, writing, reasoning, and executing multi-step tasks
4. **Multi-client workflows** become practical when each client's work runs in parallel background processes
5. **The "while you sleep" model** lets you set up agents before end of day and review deliverables in the morning
6. **Task persistence and dependencies** enable complex, multi-session workflows that survive restarts
7. **Combined with agent teams,** background tasks enable enterprise-scale operations from lean teams
8. **The future of AI agencies** is 24/7 autonomous operations with human direction and quality control

---

PRO TIP

Pre-Authorisation Is Key

Before launching a background task, Claude prompts for tool permissions upfront. You're not giving carte blanche — you're pre-authorising specific capabilities. Be explicit about what the task can and cannot do.

Ready to Work Smarter with AI?

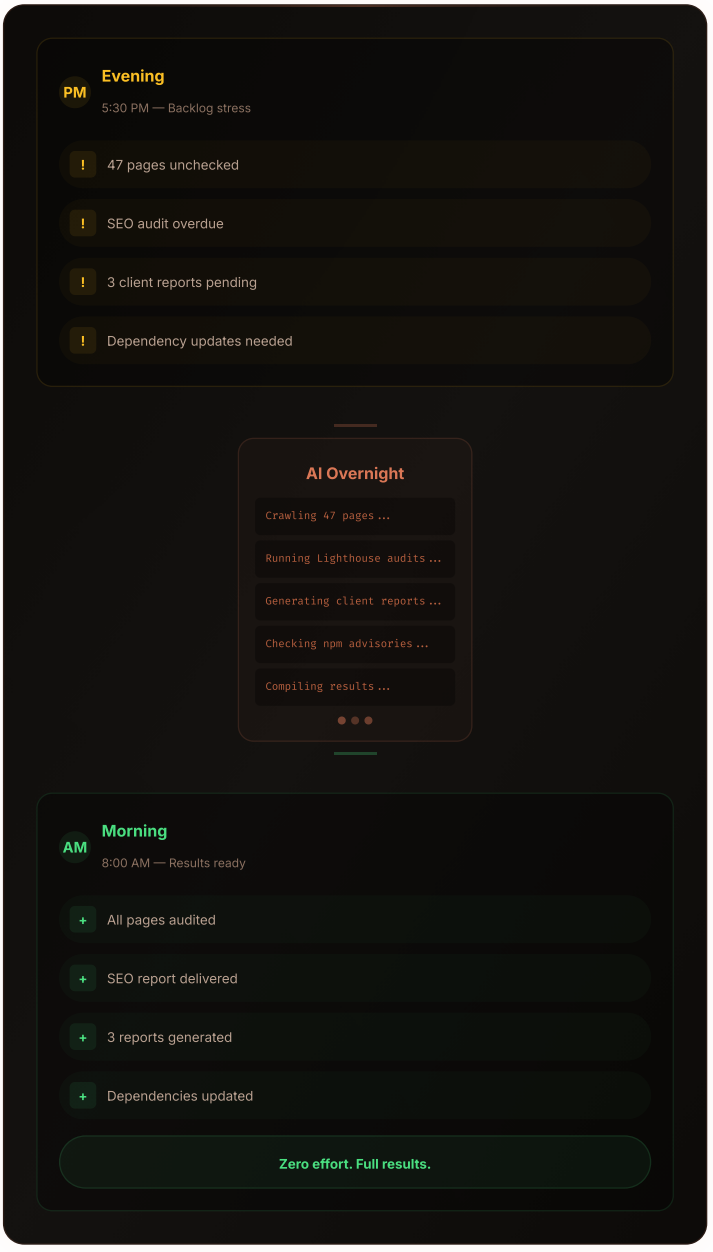
Jordan James Media uses Claude Code's background task capabilities to deliver more for our clients, faster. If you want an agency that leverages cutting-edge AI tooling to maximise results, we should talk.

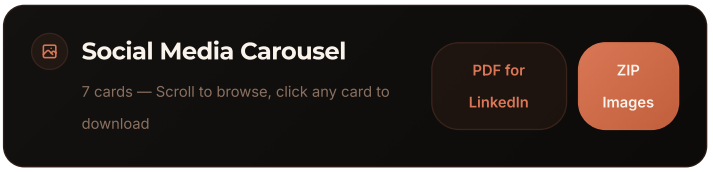
Talk to Us About AI-Powered Marketing

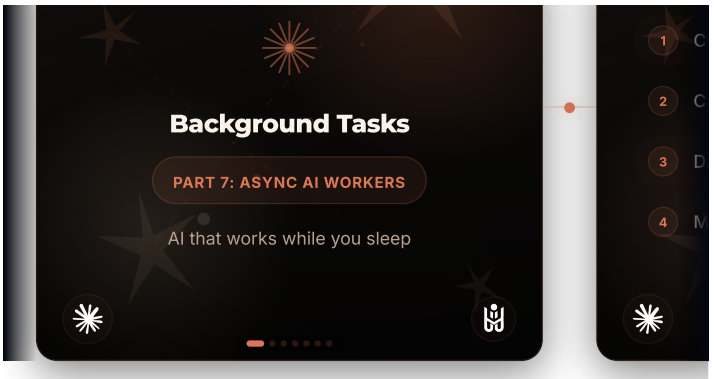
Related Reading:

- Claude Code's Biggest Update: 7 Features That Changed How We Build
- Claude Code Agent Teams: How AI Agents Coordinate Complex Projects
- Claude Code MCP Integration: Connecting AI to Your Entire Toolchain

While You Sleep







Share This Article
Spread the knowledge

Twitter LinkedIn Copy

Build With Claude Code

We use Claude Code to ship faster, smarter, and at scale. Let us build your next project.

Get Your Free AI Strategy → App Development

Claude Code
8 parts

View All

Quick Links

→ Series Home
→ App Development
→ Get In Touch

Claude Code Stats

Model	Opus 4.6
Features	7

Series Parts 8

Agency Tested
Real projects, real results

Ready to outperform your competition?

Get your free AI strategy consultation today.

Start Your Free Consultation →

Jordan James Media
GROWTH STUDIO

Australia's AI-first digital marketing agency. We build intelligent systems that drive exponential growth.

info@jordanjamesmedia.com

0475 897 737

Sydney, Australia

SERVICES

- SEO Services
- Google Ads
- Website Design
- Local SEO
- Content Marketing
- App Development

COMPANY

- About Us
- Blog
- Case Studies
- Pricing
- Tools
- Contact

LOCATIONS

Sydney

Melbourne

Brisbane

Perth

Adelaide

All Locations

All Locations →

Privacy Policy

Terms of Service

Portal Login

Built with Claude 4.6 Opus