

Learn about Claude Code Routines

Claude Code Routines launched on April 14, 2026 - and within just 24 hours, half the community had encountered the limit on the number of jobs they could complete by noon.

Mastering Claude Code Routines

Claude Code Routines launched on April 14, 2026 – and within just 24 hours, half the community had encountered a limit on the number of tasks that could be completed by noon. The other half had found the discipline to differentiate scalable processes from those that became overloaded overnight.

This course will teach you that discipline in just two hours, with 3 practical procedures delivered to your personal account upon completion.

What you will actually build

1. A scheduled backlog process runs on weekdays at 9 AM, reads Linear, and generates a 300-word morning report—using only the Linear MCP connector, without loading anything else.
2. A PR review process is triggered when a pull_request event occurs on GitHub, filtered by your main branch and label `needs-review`, maintaining one session per PR so that subsequent commits build upon the previous response.
3. A Stripe webhook process ? a process for sending draft emails with user instructions, connected via a secure proxy, with Gmail having minimal privileges to only create drafts (never send).

What makes this course different from other tutorials?

Anthropic's launch guide explains the functionality of Routines. This course explains what everyone learned in the first week about using it without exceeding limits, leaking scope, or silently deploying a faulty product on day 7.

Each lesson concludes with common mistakes that real users have made in the first 24 hours – unfiltered triggers, lengthy prompts that consume 56% of tokens for narration, the 'all connectors are enabled by default' trap, OAuth tokens silently expiring after two weeks. You will learn how to fix these common errors.

Who is this course for?

Engineers who have used Claude Code at least a few times and want to move from 'AI pair programming' to 'AI cloud of agents performing real-world tasks'. You don't need in-depth DevOps experience – just a working Claude Code installation, a Pro or Max plan, and a real workflow you want to automate.

Ultimately, you'll be submitting scheduled articles within the same week that the Routines feature launches. That's a very short timeframe, and it's happening right now.

What you will learn

1. Explain the differences between Routines and Interactive Claude Code, Cowork, and /loop.
2. Build scheduled processes, GitHub events, and API triggers with appropriate filtering discipline.
3. Configure MCP connectors (Notion, Slack, Gmail, Linear, Postgres) with minimum OAuth permissions.
4. Apply cost-effectiveness techniques to stay within the limits of Pro 5/day, Max 15/day, or Team 25/day.
5. Evaluate when Routines is the right tool compared to GitHub Actions, Zapier, n8n, or cron.
6. Implement three manufacturing quality processes onto your own account with a trial-and-error procedure.

After this course, you will be able to

1. Design and deploy production-quality Claude Code Routines running on Anthropic's cloud 24/7.
2. Configure MCP connectors (Notion, Slack, Gmail, Linear, Postgres) with minimum OAuth permissions to pass security assessments.
3. Reduce your token routine costs by 30-50% with prompt discipline, a feat most accomplished users have experienced and realized through hard work.
4. Choose the right automation tool for each task—routines, GitHub Actions, Zapier, n8n, or cron—with a clear decision framework.
5. Add the 'Claude Code Routines Expert' experience to your resume – a skill that didn't exist before April 14, 2026, and is now highly sought after.

What you will build

Document alteration detection process

A scheduled process scans your codebase every Monday and posts a short report on Slack about where the code has changed but the documentation hasn't. Runs in your own account with real Slack integration.

Routine bot evaluates PR

A GitHub-event routine evaluates pull requests on your own repository. Filtered by your main branch, limited to the `needs-review` label, it posts structured review comments that pass the code review check.

Mastering Claude Code Routines

Demonstrate that you can design, implement, and operate Claude Code Routines in a production environment with appropriate filtering discipline, connectivity range, and engineering cost.

Prerequisites

1. Basic knowledge of Claude Code is required (having run at least a few interactive sessions).
2. There are Claude Pro, Max, Team, or Enterprise plans (Routines are not included in the Free plan).
3. Proficient in using the terminal, Git, and basic JSON.

Your laptop is no longer the bottleneck.

This course shifts your mindset from interactive Claude Code to automated cloud agents. What are routines, who are they for, and what will you be building in this course?

One thing changed on April 14, 2026.

If you've been using Claude Code for the past six months, you'll know its rhythm. Open your laptop. Type Claude. Tell it what you need. Wait. Review. Repeat.

On April 14th, Anthropic launched Routines - and that rhythm was broken.

Routines are Claude Code sessions that run without your presence. They are triggered on a schedule. They are triggered when someone opens a pull request. They are triggered when your Stripe webhook sends a signal to a URL. They run on Anthropic's cloud. Your laptop can be closed. You can be asleep.

This is how Anthropic described itself in its launch blog post:

Your computer doesn't need to be turned on.

That's the only sentence that matters.

Why did this change everything?

Before Routines came along, Claude Code was an incredibly fast pair programmer – but it was tied to you. You started the sessions. You ended the sessions. You reviewed all the output.

That model scales poorly. You can only control a certain number of Claudes at a time before context switching exhausts you. And half of the useful AI work – monitoring, classification, tracking deviations – is precisely the kind you don't want to do manually.

Routines solve this problem. They allow you to describe a task once and then let Claude run it on a schedule, by event, or on demand. You only need to be there to review, not to execute.

The analogy in traditional software: It's the difference between a developer who only writes code and a developer who also provides a service. You're not just writing – you're running something.

You don't need to be a senior engineer. You don't need extensive DevOps experience. If you can write a Claude Code prompt and understand what a webhook is, you're ready.

What you will build

Upon completing this course, you will have:

1. A scheduled process is in place to sort through your team's backlog each weekday at 9:00 AM.
2. A GitHub routine event to review pull requests without consuming your daily limit.
3. A routine API trigger to respond to the Stripe webhook with a drafted instruction email.
4. A personal "routine library" - three production processes that you will actually continue to run.

This is not a simple course. The goal is for you to complete it and have a real automation system implemented on your own account.

Three types of triggers

The processes will be triggered by one of the following three types of triggers:

Trigger	Use when	For example
Schedule	You need to perform regular maintenance.	Daily report generator, weekly security scan, nightly document consistency check.
GitHub event	You need to react promptly to changes in the code.	Pull request reviewer, issue classifier, move directory inspector
API call	You need to react to whatever else	Stripe webhook, Sentry alerts, incident logs, customer support tickets

The first habit most people develop is scheduling. The most valuable habit most people develop is a GitHub event or an API trigger.

We will build each type in lessons 2, 3, and 4.

What are routines not?

A few things need clarification, as the community is still learning about this:

Routines are not a replacement for interactive Claude Code. You still open Claude to program in pairs for a feature. Routines handle the things you would normally do manually between features—classification, review, scheduled reporting.

Routines are not workflow-building tools like Zapier or n8n. Those tools are built on defined data paths (moving data from A to B). Routines are built on AI that reads, infers, and writes. We will compare them in detail in Lesson 7.

Routines are not a replacement for cron for arbitrary scripts. If your job is simple `rsync /data /backup`, you don't need Claude in a loop. Routines are most effective when the job requires AI-powered reasoning – reading code, interpreting errors, writing text.

Routines are not unlimited. Each service package has a daily limit on the number of times a process can be run. The Pro package allows 5 runs/day, Max 15 runs, and Team/Enterprise 25 runs. We will discuss cost optimization techniques in Lesson 6.

Change your core mindset.

If you only remember one thing from this lesson, it's this:

Stop thinking, "I'll ask Claude to do that." Start thinking, "I'll configure Claude to do that whenever this happens."

That shift—from requiring synchronous to asynchronous configuration—is the whole value of Routines. Each lesson in this course is actually teaching you how to implement that effectively.

Note regarding the research preview

Routines will be released as a research preview on April 14, 2026. A few things are still under development:

1. Currently, Routines belongs to individual users, not organizations. Shared group processes are not yet supported.
2. The only webhook source supported at launch is GitHub. More sources will be added in the future.
3. The user interface is still under development - please expect changes during the first 30 days.

This course is based on launch materials and first-day community experience. Anything still under development will be noted so you can adjust as Anthropic releases updates.

Before proceeding, please ensure you have:

1. A Claude Code installation works well (the command `claude --version` will return the version number).
2. A Pro, Max, Team, or Enterprise subscription plan.
3. A terminal you are familiar with
4. A real-world workflow that you want to automate.

If any of those are missing, pause here and set them up. Lesson 2 will begin immediately.

Key points to remember

1. Routines transform Claude Code from a collaborative programmer into a cloud-based agent that runs automatically without your intervention.
2. Three types of triggers: Scheduling, GitHub events, API calls.
3. Shift your mindset from "I'll ask Claude" to "I'll configure Claude to run when X happens."
4. This course builds three practical processes for product quality.
5. This is a research preview - some things will change.

1. Question 1:

What is the core shift in thinking when moving from interactive Claude Code to Routines?

1. A. Change from "I direct Claude" to "Claude operates independently while I sleep"
2. B. Stop telling Claude to do everything and let him decide for himself.
3. C. Switching from the command-line interface (CLI) to the web interface

4. D. Use shorter prompts to save tokens.

EXPLAIN:

The shift is from synchronous direction to asynchronous delegation. You configure a process once, and then it runs automatically—on a schedule, when an event occurs on GitHub, or when your API triggers it.

2. Question 2:

Which of the following is NOT a valid trigger for a Routine?

1. A. Cron schedule (every weekday at 9 AM)
2. B. The pull_request event on GitHub
3. C. Call the API from your own webhook.
4. D. A voice command from your phone

EXPLAIN:

Currently, Routines supports three triggers: Schedules, GitHub events, and API calls. Voice, which is not a trigger, will be supported in the April 2026 research preview.

3. Question 3:

What are the main differences between interactive Claude Code sessions and Routines?

1. A. Routines uses a different AI model than interactive sessions.
2. B. Routines can only be run once a day.
3. C. Routines run automatically on Anthropic's infrastructure, without needing your laptop to be turned on.
4. D. Routines require separate registration.

EXPLAIN:

Routines run on a cloud infrastructure managed by Anthropic. Your machine doesn't need to be powered on. The model is the same – the difference lies in the execution environment and the triggering model.

Submit your work

Training results

You have completed **0** questions.

-- / --

[Review the lesson](#)

You finished reading the article "**Learn about Claude Code Routines**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
