

Build a proactive agent workflow with Claude Code

21 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=ESP7PLTXNY8](https://www.youtube.com/watch?v=ESP7PLTXNY8)

<https://www.youtube.com/watch?v=eSP7PLTXNy8>

SUMMARY

Maya from Anthropic discusses the new "routines" feature in Claude Code, designed to transform proactive agent workflows by automating tasks without requiring constant user input. The feature aims to alleviate common challenges faced by developers, such as infrastructure management and session monitoring, enabling agents to operate more independently and efficiently.

- Routines automate remote Claude Code sessions by defining prompts, repositories, connectors, and triggers.*
- Key challenges in building proactive agents include managing hosting, triggering sessions, and maintaining human oversight.*
- Routines run on Claude's managed infrastructure, eliminating the need for local machine dependency.*
- Users can set customizable triggers for routines, including time-based schedules and event-based actions.*
- Routines allow for interactive sessions, enabling real-time monitoring and steering of agent actions.*
- A practical example includes automating documentation updates based on changes in source code or GitHub events.*
- The feature encourages developers to focus on domain expertise rather than infrastructure maintenance.*
- Maya invites users to start using routines with a simple command in Claude Code.*

Hello everyone. How are you? Good. Okay. Amazing. Welcome to the last workshop session of the day. I hope you have all enjoyed the very first day of code with Claude. Uh my name is Maya. I'm a member of our applied AI team here at Anthropic. Uh what that means is I spend about half my time developing our own firstparty products and features and the other half of my time helping customers develop their very own products, features, agents on top of our models.

Today I'm here to talk to you about how to build a proactive agent workflow with cloud code. Um, can I get a show of hands? Who has used our routines feature inside of Cloud Code? All right, some folks over here. Awesome. Awesome. Um, that's what I'm going to be talking about today. Okay, so first off, a question for the group here. Who has tried to run clawed code on a cron? Can I get a show of hands? Folks, put your hands up high. Awesome. Awesome. Um, now keep your hands up if you've enjoyed building all of that infra and maintaining that job. All right, we have one guy back there. We have one guy.

Thank you. Thank you for your effort. Thank you for your work. Um, we felt similar pain internally at Anthropic as we tried to develop uh proactive agents uh that run on cloud code. Uh, and we decided to do something about it. So, we believe that coding agents shouldn't wait for you to press enter to get started. Right now, Claude code is a really powerful coding tool, but we want to take Claude code and turn it into a really

powerful coding teammate.

A teammate notices when something breaks and does something about it. Right now, a tool waits for you to enter your prompt and actually press enter. So the goal of today's presentation is to talk about how we have created this feature called routines to take clawed code from a tool today um into the teammate of tomorrow. So today we'll be talking about four things. I'll go through some of the challenges that you folks have felt uh building proactive agents today. I'll go through this new feature inside of cloud code called routines.

We'll go through a real example about how we use routines internally at Anthropic to automate documentation creation and then finally we'll talk about applying routines to your own workflows. So I want to talk first through the challenges with building proactive agents today. We all know it's doable. Um but I want to talk about what what's a little bit cumbersome with this. The first thing that's a little bit difficult with building proactive agents today is deciding where these agents should run. You probably don't want them running on your local machine because if you close your laptop or your laptop dies, your agent session is done. What that means is you'll need to manage things like hosting, data persistence, and authentication.

Basically, you'll need to build a whole infrastruct outside of your prompts, which is doable, but it's a lot of work and there's a lot of boilerplate code there. The next thing you'll need to do is figure out when to actually kick off these sessions and trigger these agents. Again, you can build things and build on top of cron or you can do things like uh post to endpoints that you have to spin up. Um, but again, there's there's a lot of infra that you need to build yourself here.

Finally, the challenge with building proactive agents today is sometimes you want to be a human in the loop, but other times you want to be a human out of the loop with these agents. Um, right now when you kick off a headless clawed code session, it's often hard to figure out what your agent is actually doing in real time. There's no way to watch, steer, bound, or even resume your agent session. It's it's difficult to do that.

Um, so we wanted to address each of these three issues um, and build routines. Routines is a brand new feature inside of Cloud Code. It's an automation where you can kick off a remote Claude code session by only defining the prompt, what repos you want to connect it to, what connectors it has available to work with, and a trigger. Claude Code handles the rest. So there were three kind of main things we were thinking about as we went ahead and developed this routines feature inside of cloud code.

The first thing is that we wanted these agents to be always available. These agents these routines run on claude codes managed infrastructure. And what what's nice with that is that we deal with the hosting the session state and the connector off for you. nothing depends on your laptop being opened and we deal uh with all of the cloud stuff for you which I think is quite nice. The next thing is we want these agents to be able to work proactively with customizable triggers. You might want to kick them off on a timebased schedule or you might want to work uh event based.

uh we have the ability to work natively with GitHub events as well as your own custom events that you can post

to um web hooks and endpoints with the event payload as context. Finally, and the last point that I think is really nice is that these clawed code sessions that get get launched with these routines are interactive and steerable as if you were launching claude code in the terminal. Every routine is really just a claude code session under the hood that you can open, you can watch, follow up on, steer, and resume um from web CLI and desktop.

And so I want to walk you guys through a real use case that uh we use here at Entropic internally. So the question for us and for a member of our engineering team is how can we automate docs creation with routines? So, just to add a little bit of data behind this, weekly PRs for Claude Code have gone up 200% since the beginning of the new year. This has been super awesome for our Claude Code engineering team. Um, their productivity is insane.

This has been really awesome for you folks because you get new features inside of Claude Code very, very quickly. The one person that this has not been so awesome for is the one engineer responsible for maintaining our documentation across cloud code and the agent SDK. And so when routines launched, she was a super big fan and early adopter. And I want to walk you through how she set up a couple routines to help automate documentation creation for cloud code and our cloud agent SDK.

So on my side here, I have the terminal open. And I encourage you all to open your terminal and launch cloud code. And inside of the terminal here, I'm able to type what we see on the screen here, slash schedule, and actually type in something that Sarah, our documentation uh queen, uh has done to actually set up this routine. So she went and typed in once a week uh please review all the new changes merged to main against our documentation repo and create a PR to update docs if you see any changes.

I encourage you folks to think right now what are some tasks that you do every day that would help if they could run on a schedule or if cla could actually initiate these sessions for you. Um I encourage you to think about that. After kicking this off inside of Claude code, Claude comes back and prompts me with a couple questions. It might ask, "Hey Maya, at what time every week do you want me to actually kick this off?"

Or once I create a PR, do you want me to notify you in any way? Maybe ping you on Slack. And once I answer these questions, Claude actually goes ahead and creates a routine um that we'll view in a little bit inside of Claude Code on the web. But first, I want to walk through the three main decisions you'll need to make as you create any routine. The first decision is you'll need to figure out when your routine should trigger.

What's actually the event or is there a certain cadence that you want this to run? The second decision is what context or what information does Claude need to have to actually be successful here? Do you need access to certain docs or does Claude need access to certain tools to ping you? Finally, the last thing to think about is how do you actually steer Claude in the session to keep it honest? How do you guide claude to the output that you want? And so we'll dive into each one of these and I'll talk through how Sarah, a member uh of our team, actually does this to automate documentation creation.

So the first one is the trigger. When uh should this event actually run? So inside of routines, there's basically two ways to do this. You can have things kick off on a schedule on a timebased trigger. For that earlier example I showed you, uh this is how we do a weekly review of differences between our source code for cloud code as well as our documentation repo. You can also have routines kick off uh on an event-based cadence. So maybe every time a release is cut, uh you can diff the release branch against the docs and see if there's any new features that you'll need to spin up PRs for in our documentation repo.

Or maybe your engineer is actually deploying uh changes and creating PRs might tag their changes. Maybe this is like a new feature. They could tag it with a label that says need docs. And you could actually kick off cloned code sessions anytime one of these uh labeled PRs get merged. The next thing you'll need to think about is context. What does your agent actually need to know to be successful? Likely you'll need to give it access to either one or more codebase repos. So for this docs example, uh we need to give Claude access to not only our Claude code source code to figure out what new changes exist there, but also our docs repo, right? For Claude to actually create uh new PRs there.

Next, you might want to provide additional context to these sessions. Uh maybe for this one example, I want Claude to have access to all of our existing marketing briefs. Maybe I want Claude to use similar language and verbiage that we use uh in other marketing materials externally at Anthropic. So maybe all of this lives inside of Google Drive and I'll want to give Claude access to these files during the session. So I'll hook up the drive connector.

Or maybe anytime Claude creates a PR, um I'll actually want it to ping me on Slack. Uh so I'll give it access to the Slack connector. It's important to think about this as you're setting up the routine because whatever context Claude has, that's uh the ceiling of how successful Claude will be. Finally, the last thing to think about here is steerability. How do we actually ensure the quality um of Claude's outputs?

There are a couple ways to do this. Um, one thing that I think is quite interesting is to actually invest in agent onagent review. If folks have actually designed multi- agent systems here and have heard of the generator uh critique pattern, uh this is something that we've borrowed here. You can actually set up one routine to go ahead and create uh docs PRs and you can set up another routine that maybe triggers on that PR's creation um to actually go ahead and leave comments on the PR before a human actually gets to it.

Another option we had emphasized it's nice to have a human out of the loop, but sometimes you actually do need to monitor these cloud code sessions and maybe nudge Claude in a different direction. What's really nice and what I'll show you in a second is that you can actually open Claude Code on the web and you can view what's happening inside of a live session as if you would working with Claude in the terminal. You can ask it questions midsession. You can push it in another another direction.

You you can also resume a past routine or a past session and continue the conversation. Finally, the last thing we do and what I think is quite obvious is that we verify Claude's outputs. For this documentation example, we actually render the page that or the documentation page that Claude has changed and created and we confirm

those outputs are what we expect. And so now I'll jump back into that initial routine that we've kicked off here in this demo. Um, and we'll jump to the demo slides. Awesome. And so we can see right now I'm in cloud.ai AI and I can go over to this left side panel and actually click on this uh code button and I can jump into routines on this left hand side and I can actually click on this routine that I had created earlier.

On the left hand side here you can see that it's connected to two repositories. our uh mocked up Cloud Code source code as well as our Cloud Code documentation. You can see that this runs every Monday at 10 a.m. and it's connected to GitHub as well as Slack. These instructions here on the right hand side Claude generated for me based on the initial prompt that I pasted in and the questions that I answered. We can see here that uh this is a weekly documentation sync uh for our Claude code fork.

And I can actually go ahead and click on a session here. And I can see that these initial instructions are what uh gets pasted in for the very beginning of this Claude code session. We can see that Claude has read these instructions and started by looking at the source code repository to see uh any changes, any recent PRs that have been merged. Looked at our change log and compared that to what's inside of the documentation repo.

We can see Claude's actually found some changes here and gone ahead and opened a PR for me. This is one example where you can kick off a routine on a schedule. Now I want to show you another example where we can actually kick off a routine uh based on a GitHub event. So here I've actually created a new routine and I've already filled some of this in. Um I want to make uh another documentation automation uh routine and but this time I want this to actually trigger every time I create a new GitHub issue.

So I pasted in some instructions here basically to investigate the issue that this session triggers on. Figure out if it's related to a documentation gap and then if it is and if you believe that this is a gap um go ahead open a PR and actually ping me in this channel. So I've gone ahead and connected our cloud code documentation repo as well as our source code again. And I want to show you how I actually set up this trigger.

Like I had mentioned before, there's two different types of triggers. Um, schedulebased and event based. And within event-based triggers, we have native git github events supported here as well as the ability to trigger from your own code by sending a post request. So here I'll create a uh GitHub event trigger and trigger on issue opens anytime I open an issue inside of this cloud code documentation repo. And I want this connected to Slack so I can send me a ping anytime I make a PR um as well as our GitHub MCP.

So I will go ahead and create this here. And now let's make sure that this is working. So I have a new issue open here that I'm about to create inside of our Cloud Code Docs repo. Um so I happen to know that there are a few few tools missing from docs in this new version. So I'm going to go ahead and actually create this So I can see here that I've gone ahead and created that. And now let's refresh this page.

And we can see actually that a new run has gotten picked up here. We can see that these initial instructions are the very first prompt. Um and we can see that this additional context from this issuer passed in as well. I happen to know that I actually already have another PR open for this. So, let me just guide Claude to stop this session.

I've already made these changes and we can see here the ability to actually steer Claude in real time after routine gets kicked off.

Awesome. Um, now let's come back and talk about the different ways uh that we can use routines to actually automate your challenges. As uh developers, I want to talk through um a couple ways that we could turn common software engineering or developer challenges into routines with quad code. The first one I want to talk through is this deploy verifier. Um maybe you have recently uh just deployed changes to an to a service and you want to make sure that this service is healthy and you shouldn't roll back these changes.

I want to think about this uh in three ways. First, what should my trigger be here? Second, what context is important to provide to Claude in this routine? And lastly, how do I plan to actually interact or steer this routine to keep Claude honest? I happen to know that my CD pipeline can post after every deploy. So, this seems like a pretty good trigger to actually kick off this routine on. I can actually post to this uh web hook that we support inside of routines. And this can actually kick off my Claude code session.

In terms of what context I think is important to provide Claude here, I can think of a couple things. one, it's probably nice to give Claude access to the source code for the service that we reach recently deployed on. Next, it's probably important to give Claude access to monitoring tools. Maybe that's data dog, maybe that's Graphana, I don't know what you folks use, but um likely helpful to give Claude access to these uh monitoring tools. And maybe if something goes down, I want Claude to alert me. I want Claude to ping me on Slack or send me an email um or maybe maybe even send me a text using Twilio or something like that. These are some of the connectors or the tools that I would give Claude access to inside of this session.

In terms of keeping Claude honest or actually steering my session, maybe I'd start by having Claude run an investigation for me and giving me an eventual no go or no-go decision to actually roll back this change. I could jump into Claude code on the web and actually view and read Claude's analysis for this session. Then maybe I can continue to work with Claude if I actually think that this should be rolled back and actually use Claude to help me roll back a change.

Maybe eventually as I watch Claude work more and more and trust its decisions, I can let Claude roll back the change itself if based on the monitoring monitoring uh data it has access to. Uh if I deem that that's the right decision. There are other challenges like maybe you want to build an on call investigator or maybe you're actually a PM and your job is to go through uh a lot a lot of issues inside of your backlog. Maybe that's GitHub issues.

Maybe that's posts inside of a Slack channel and maybe you want to kick off a weekly job that actually reads through all of these issues. Um maybe it's kicked off on a timebased trigger. You give it access to GitHub and Slack and wherever your issues live. Um, and you use Claude to actually help you prioritize and maybe open PRs for the most important issues. Okay, my final takeaways here, uh, proactive agents, uh, beat reactive agents. We want Claw to go from a tool to a teammate. Um, you can move from an agent that is waiting for you to actually press enter and create a PR to an agent that reacts to problems um, and opens a PR itself.

We built routines so you don't have to focus on maintaining all of this infra, but instead you can actually concentrate on your domain and process expertise. This is what routines handle for you. And finally, I encourage you to get started with routines today. Uh you're a single slashcchedule command uh inside of Claude Code uh away from creating your very first routine. Awesome. Thanks so much,