Keep Claude Code working without you

Here's how to set up what I showed in my video. Claude Code has three levels of autopilot above typing "keep going": a job on a timer, a goal with a finish line, and a task in the cloud that runs with your laptop closed. You won't write any code.

What you need

- A paid Claude plan. Claude Code comes with Pro, Max and Ultra (and with Team and Enterprise). It isn't in the free plan.
- The Claude desktop app. All of this runs in **Claude Code**, which is the **Code** tab of the app, not the regular chat.
- For level three, a Pro, Max, Ultra, Team or Enterprise plan. Cloud scheduled tasks are in research preview, so what they can do may change.

The three levels

Level	Command	Claude keeps going until	Needs your laptop on
0	You type “keep going”	It decides it’s done	Yes
1	`/loop`	You stop it, or Claude decides the work is done	Yes, with the session open
2	`/goal`	A finish line it can prove is met	Yes
3	`/schedule`	It runs again each time its schedule comes round	No

Most people are at zero. Claude works, then stops when it thinks it’s done or needs you, and you type “keep going” or click allow, until it stops again.

Step 1: open Claude Code and turn on auto mode

1. Download the Claude desktop app from claude.ai/download, install it and sign in.
2. Click **Code** at the top of the sidebar.

3 Make a new folder to work in, for example “Autopilot” in your Documents.

4 In the row under the message box, choose **Local**, then click the folder control, choose **Open folder…** and pick that folder.

5 In the same row, choose **Auto**.

Auto mode is what removes the allow clicks. In auto mode, a second model reviews each action instead of you. In a terminal, `Shift+Tab` cycles the modes, and the status bar shows `auto mode on`.

> Keep an eye on it. In Auto, Claude can install programs and change files without asking first. If you’d rather approve each step, choose **Manual**.

Auto mode doesn’t start a new turn. Claude still stops whenever it judges the work done. The next three levels fix that.

Level 1: /loop, a job on a timer

Type `/loop`, then a time and a job. For example:

```
/loop check if my site is up every 30 minutes
```

Claude turns the time into a schedule, sets it up and tells you how often it will run. The time can go first, like `30m`, or at the end, like `every 2 hours`. You can use seconds, minutes, hours and days.

What to know before you rely on it:

- **Claude just repeats.** `/loop` has no finish line. It runs the job every time the clock comes round.
- **Your laptop has to stay on,** and the Claude Code session has to stay open. A loop only runs while Claude Code is running and idle.
- **It stops after seven days.** A recurring loop runs one last time and deletes itself. To go longer, set it up again.
- **It can start a little late.** A job set for every 30 minutes can start up to 15 minutes after the time.
- **To stop it,** type: `cancel the site check loop`.

If you leave the time off, Claude picks one itself each time, between one minute and one hour.

Level 2: /goal, work until the finish line

Type `/goal`, then a finish line. Claude starts working straight away, and after every turn a separate model checks whether the finish line is met. If it isn't, Claude takes another turn. You don't type anything between turns.

```
/goal my website scores 90 or higher on
Google's speed test
```

While a goal is active, you'll see `◎ /goal active` with how long it has been running.

The finish line decides whether this works, so write it with three things:

- **One measurable end state.** A test result, a file count, an empty list, a score.
- **How Claude should prove it.** The checking model doesn't run anything itself. It only reads what Claude has shown in the conversation, so ask Claude to show the number. If Claude can't run the speed test from your computer, pick a

finish line it can check, like “every link on my site works”.

- **What must not change on the way.** For example, “no other file is modified”.

Add a limit, so a goal can’t run forever: put `or stop after 20 turns` at the end of the finish line. Claude reports against it each turn.

Claude stops in four cases: the finish line is met, the checking model judges it impossible, a problem only you can fix (a sign-in failure or no credit left, for example) clears the goal, or you stop it. So it doesn’t always reach the finish line.

- Type `/goal` on its own to see the finish line, the turns so far and the latest reason.
- Type `/goal clear` to stop it.

For a goal to run without you, run it in auto mode (Step 1). In Manual mode, Claude still asks before anything your settings don’t already allow. And your laptop has to stay on.

Level 3: /schedule, a task in the cloud

Type `/schedule` and describe the task. Claude asks you follow-up questions, then saves it. For example:

```
/schedule every weekday at 7am, [the job
you want done], and leave the result for me
to review
```

Claude asks about the schedule, the repositories and the prompt before it saves anything. You can see and change your tasks at claude.ai/code/routines, and click **Run now** to try one straight away.

What to know:

- **It runs on Anthropic's servers,** so it keeps working when your laptop is closed.
- **It starts in auto mode.** You don't pick a permission mode. A run still stops and waits for you if an action matches an ask rule, or after repeated blocks.
- **The prompt has to stand on its own.** The task runs by itself, so say what to do and what done

looks like.

- **It starts from a fresh copy of a GitHub repository,** not the files on your computer. What else it can reach depends on the repositories you pick, the environment's network access and the connectors you include. Remove any connector the task doesn't need, because a run can write through the ones you include.
- **Minimum interval: one hour.**
- **It acts as you.** Commits, pull requests and messages through your connected accounts carry your name.
- **Its runs count against your usage,** like any other session.

To see what a run did, open it from the task's **History** list.

Which level to start with

Pick one job you keep checking on yourself.

- It should repeat on the clock, and you only need to know the result: level 1.

- It has an end, and you can say how Claude proves it's done: level 2.
- It should run while you're away, from a repository, and wait for your review: level 3.

Written on 10 November 2026 from Claude Code's own documentation pages on `/goal`, scheduled tasks, scheduled tasks in the cloud and permission modes. I haven't run these steps myself while writing this guide. Claude changes fast: if a menu has moved, look for the same words in the app.