

Your day: one card per ticket

- 1 Catch up /kotharo:work-resume
- 2 Start /kotharo:start-feature-worktree <KEY>
- 3 Run /team <KEY>
- 4 Answer the end card
- 5 Next /kotharo:next-ticket <NEXT>
- 6 Save /kotharo:work-save

Big or risky changes ask first. One yes on the end card, **Ship**, commits, pushes and opens the PR; you merge, and close-out moves the session to **main**. After **/kotharo:next-ticket**, run **/clear** and send what it suggests.

The band above the prompt

1 > ABC-12

2 Merge PR #7?

3

merge

4 #7 ✓

Docs 5

2 need you

5 61%

6 Merge

Details

ABC-14 Approve the plan? 12m

Approve

Details

ABC-15 Ship ABC-15 into main? 3m

Ship

Details

7 jenkins named by the repo's files here, and deploy is unbound. Bind it?

Bind

Not here

- > working > needs you > trouble
- > held, not joined or merged
- 1 **Mark**: its diamond shows what the ticket needs, in the colors on the left
 - 2 **Question**: the stop waiting on you, or a status such as "2 helpers working"
 - 3 **Stage dots**: done in green, the current one in the diamond's color; • unfolds them with names
 - 4 **Chips**: the open PR, linked (✓ ready to merge), **Docs** for the ticket's documents and PRs, and the tickets that need you
 - 5 **Context ring**: amber from 60%, red from 85%, when **Save notes** runs **/kotharo:work-save**
 - 6 **Buttons**: the team's default answer, and **Details** for the whole stop card
 - 7 **Hint**: a tool to bind, asked once, only when no stop waits

The need-you list and narrow windows

- A row's button answers that stop in its own session; **Details** opens its card here.
- Held tickets go last, marked queued. A **LOCAL-n** key was made while the tracker was down.
- A row whose session isn't running offers **Copy resume**.
- Rows clear once the PR merges, with a toast such as **Cleared ABC-12: PR #7 merged**.
- A row with no session, worktree or open PR turns **stale** and offers **Dismiss**, which works like **/kotharo:leave**.
- On a narrow window the band drops the chip words, then the percent, the stage name, Docs and the PR chip.

Start work

Any work	/team <KEY>
No ticket yet	/team "<what you want>"
Already started	/kotharo:join <KEY>
Stop early	--until design or scope, plan
Lighter run	--mode lean or thorough
Already decided	--decided criteria
How or why	/kotharo:trace-behavior
Brainstorm	plan mode, then /team "<idea>"
No team	ask Claude, then /ship
New repo	/kotharo:onboard-repo

What each kind of ticket gets

THE TICKET IS	YOU GET
A bug or tweak	A direct run and one end card

THE TICKET IS	YOU GET
A story from a known epic	Mostly direct, one story at a time
A big feature, no settled criteria	PRD and design at one scope stop, then the end card
A request for a design	One design stop; Publish to <tool> with a docs tool bound
Design now, build later	--until design ; close-out offers to commit or post it
A request for a PRD	The PM writes it; one scope stop, no code
A prod incident	Hotfix lane, your go for prod, then a blameless review
A pipeline or infra change	The devops role, a deploy stop, then deploy-verify

Lanes and how a run goes

LANE	FITS	STOPS
small-fix	Up to 3 files	End card
clear-ticket	Clear criteria	End card
new-feature	Large, needs a PRD	Scope, end card
hotfix	Prod is broken	End card, prod go

Every code ticket starts in **direct mode**: the main session changes the code, runs the tests, lint and typecheck, runs **/cr** once, fixes any Major and shows one end card.

It asks first, with a reason, when the change touches more than 8 files or 2 projects, or a migration, auth, tenant data or a public contract. Skipping the plan still gets a review before the merge stop.

Make it work first, then refine. The smallest change that meets the criteria, one test per criterion, the rest on Follow-ups.

Cost, PR size and slices

Estimate	The first stop prices the run
Budget stop	Past limits.budget_stop_multiplier times the estimate: Continue or Cut
Size stop	A PR over the limit: Split here , Accept or Raise the limit
PR size	review.max_pr_lines , review.max_pr_files , off by default
Slices	A big ticket ships as several PRs; the band reads PR 2 of 3
Prices	kotharo estimate price

During a run

Answer	Press the default, or a digit
Label ends in ...	Fills the prompt for you to finish
Accept answers	Approve with these answers
Where it stands	/kotharo status
Skip a step	/kotharo skip <step> <reason>
Model for a role	/kotharo tier <role> <tier>
The plan	/team-plan
Ticket docs	/kotharo docs
Before a PR	/kotharo:pr-prep

Review a PR	/kotharo:cr
Held stops, helpers and prompts	
Held stop	With 3 waiting, new stops wait quietly and read Plan stop: 3 ahead
Several PRs	Merge 2 merges the ready ones in order
Helpers	Press 2 helpers working for one row each; silent 5m means no tool for 5 minutes
Manual mode	Outside auto mode, after 10 quiet minutes the band says it may be waiting on a permission prompt

What the end card shows

- The **Intent** and **Riskiest area**, Follow-ups, then gates, tests, **Reviewed by**, such as **cr (0 Major, 2 Minor)**, and the diff.
- Ship** is one yes to commit, push and open the PR, reviewing again only if the code changed. You merge it, and the team reads the PR's state and goes on.
- Follow-ups** come Major first: the top 3 and a count of the rest. Answer **all** or **all but 2**.
- Blockers** each get their own card: **Fix it**, **Accept with a reason** or **Defer to a ticket**.
- Up to 2 changes the retro suggests, such as a setting or a CLAUDE.md line, each with its evidence.
- Flaky tests** passed when run again alone. They're named and never disabled.
- PR review** names the bot review to come; the merge waits for it, or offers **Wait for the review**.
- When a review tool's limit is used up, a budget stop offers **Wait** or **Skip the review**.

When else it stops you

- A login or VPN it can't do for you.
- A question the code can't answer.
- A fix loop past its round limit: 2, or 1 in lean and 3 in thorough.

Fix and tune

Overkill	/kotharo:leave keeps your work
Lost track	/kotharo:work-resume <KEY>
Bad deploy	/kotharo:rollback
Tracker down	/kotharo:sync-tickets
Plain branch	/ship --push skips the ask
Pull main in	/kotharo:sync-main
Old worktrees	/kotharo:prune-worktrees
Settings	/kotharo:config
Missing a tool	/kotharo:setup

Modes, autonomy and teaching

Direct	ask by default; always never plans, never always plans
Mode	balanced , lean (1 fix round) or thorough (3)
Autonomy	guided adds plan and design stops; autonomous pushes and deploys to dev

Teaching	explain says why each step matters; coach asks you to guess first
Turn it on	/kotharo:config set learning.level coach --user
Your progress	kotharo teach progress

Your CLAUDE.md always wins.

Without auto mode

/kotharo:setup proposes allow rules for **.claude/settings.json** and writes them on your OK. It never sets a mode; set acceptEdits in your own **~/claude/settings.json**.

To stop the script's prompts, allow **"Bash(kotharo *)"** in **~/claude/settings.json**.

After the end card's yes, Claude Code confirms the push and the PR once each. **/team** says how many prompts to expect; **kotharo help** lists the script's commands.

More commands

Search all repos	/kotharo:estate-scan
Sprint status	/kotharo:sprint standup
Release	/kotharo:release-cut
Release steps	Run sheet, watch, morning check
Stats	/metrics
The design	/diagram
Hours logged	/kotharo:session-hours

Tools it connects

Tickets: Jira, Linear, GitHub Issues, GitLab Issues, Azure Boards. Writes wait and go later if the tracker is down

Source host: GitHub, GitLab, Azure Repos. Opens the PR only after your OK

Deploy: Jenkins, GitHub Actions, Vercel, Netlify, Cloudflare Workers, Azure Pipelines. Deploys only on your go, then checks it is live

CI: GitLab CI. On GitHub the team uses **gh run**

Errors and logs: Sentry, New Relic, Honeycomb, ClickStack, App Insights, Datadog, CloudWatch. Bounded queries only

PR review bot: CodeRabbit, Claude Code Review, Codex, Copilot code review, Gemini Code Assist, Greptile, Qodo Merge. Its open findings come in by severity

Local review: CodeRabbit CLI, Codex CLI, Greptile CLI. Reviews before the push

Static analysis: SonarQube, Semgrep

Browser testing: Playwright. Local hosts and test accounts only

Docs: Confluence, Notion, Google Drive, repo docs/. Close-out offers **Publish to <tool>**

Chat: Slack, Google Chat, Teams

Design source: Figma

IaC: Terraform

Cloud sign-in: AWS SSO. Gives you the **aws sso login** line to run

Secrets: AWS SSM Parameter Store, 1Password, HashiCorp Vault, Doppler. Says a secret exists, never reads its value