ProjectX

WORKFLOWS & AGENTS · DATASHEET

Automate the response, and keep every step accountable.

Workflows react to a video alert, a risky sign-in, a cloud anomaly, a data condition, a schedule or a webhook, and act across the platform: notify, open a case or a ticket, run SQL or a pipeline, revoke access, lock a door. Agents do the same under their own identity and budget. Every action is authorized as whoever started it, can be simulated first, and is recorded.

Why teams choose it

01
See what it would do first
A dry run plans and authorizes every action exactly as a real run would, and takes none of them.

02
Never more than the person behind it
Each workflow action is authorized as the triggering principal, and an agent acts within the narrower of its own and its user's permissions.

03
Destructive steps wait for a person
Revoking, deprovisioning and other destructive or costly actions stop for human approval, with a timeout and escalation.

04
Stopped within seconds
A kill switch per agent, per tenant or globally halts an agent before its next step, within 5 seconds.

In numbers

< 5 s
for a kill switch to stop an agent

6
reference agents, from security analyst to offboarding

0
side effects from a dry run

What it does

Triggers from every pillar

Schedule, data condition, video alert, identity risk, cloud anomaly, webhook or manual, with conditions written over any pillar's data.

A workflow starts from the event that matters and checks its conditions before it does anything, so a high-risk sign-in from a finance account can be handled differently from the rest.

Actions across the platform and beyond

Notify, create a case, open a ticket, call an API, run SQL or a pipeline, provision and deprovision, revoke access, disable an account, lock a door.

Connectors reach email, SMS, push, Slack, Teams and webhooks; PagerDuty and Opsgenie; ServiceNow, Jira Service Management, Zendesk and Freshservice; door control, PA, radio and SNMP; and generic HTTP, MQTT and message queues.

Build visually or in code, versioned

Build a workflow in the console — a live diagram of trigger, conditions and actions, reordered by drag or by keyboard — or keep the same definition in code, with versions, rollback and a run history recording inputs and outputs.

The builder and the API write one structure, so a workflow made in either is the same workflow. Publishing a change makes a new version; rolling back republishes an earlier one as the next version, so the history is never rewritten.

Approvals and error handling

Human approval gates with timeout and escalation, and per-action retry, compensate or escalate when something fails.

Each action says what happens if it fails. A refused or failed step is recorded as such in the run, and the run reports what was and was not done.

Agents with their own identity

An agent is a real principal with scopes, credentials and a budget, a registry of tools each declared with a schema and a permission, and memory.

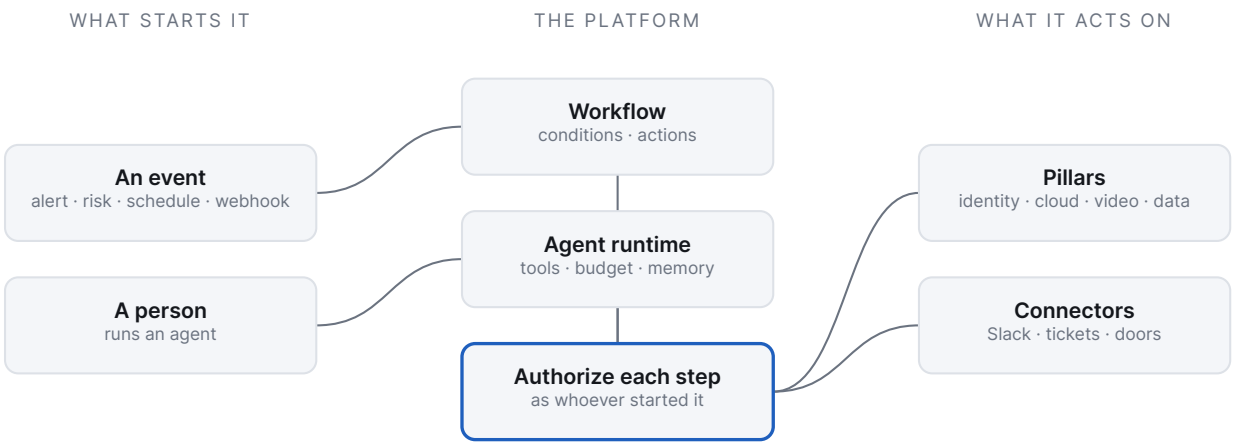
The runtime plans, calls tools, keeps short-term context and long-term memory in the lakehouse, retries and times out. Agents are versioned with staged rollout and rollback, and six reference agents install from a catalogue.

Guardrails on every run

Tool allow-lists, spend caps, rate limits, blast-radius limits on destructive actions, hard-stopped budgets and a full replayable trace.

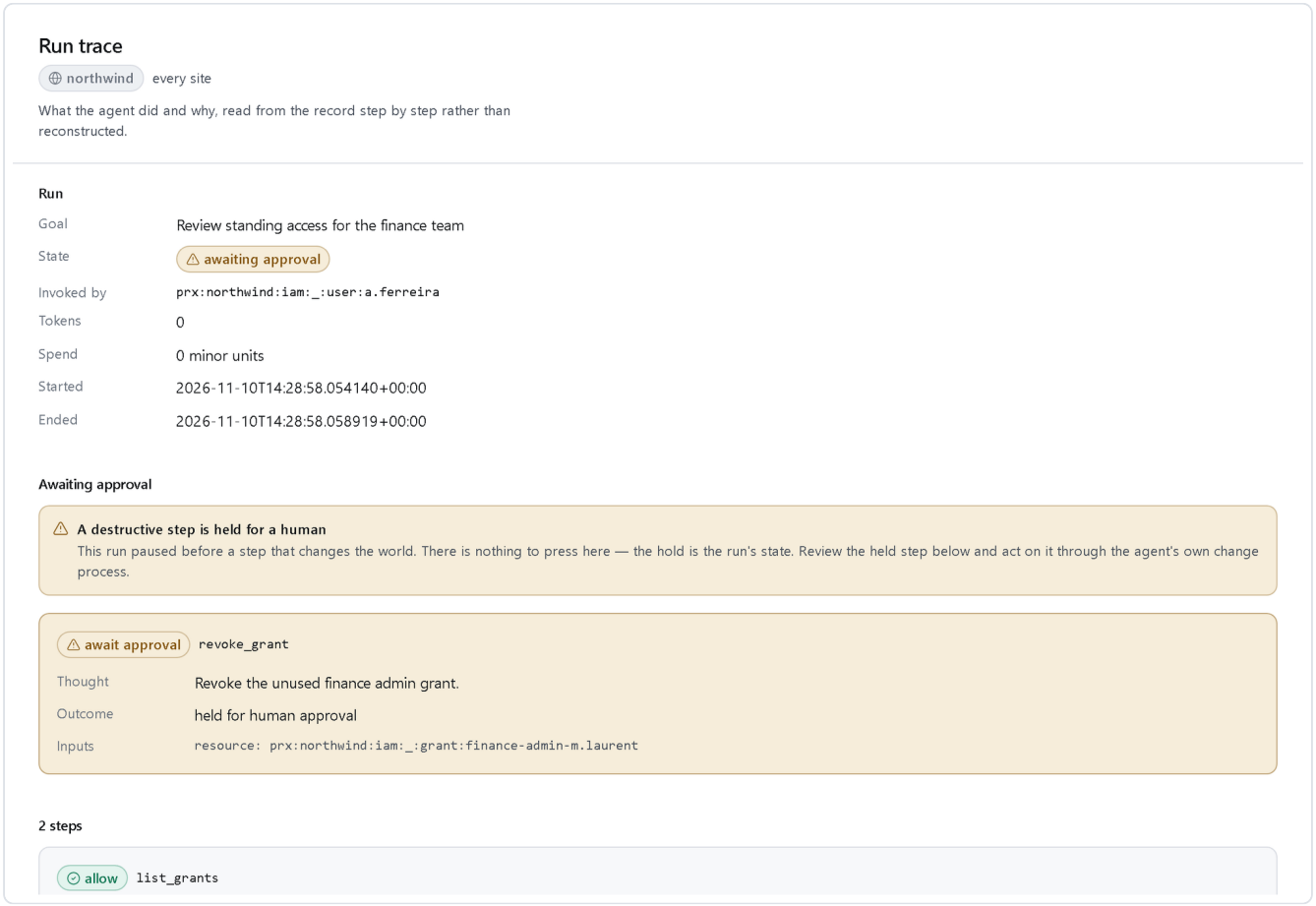
Every reasoning step and tool call is recorded with its decision, so a run can be replayed and reviewed. A budget in tokens or currency stops the run when it is spent.

How it fits together

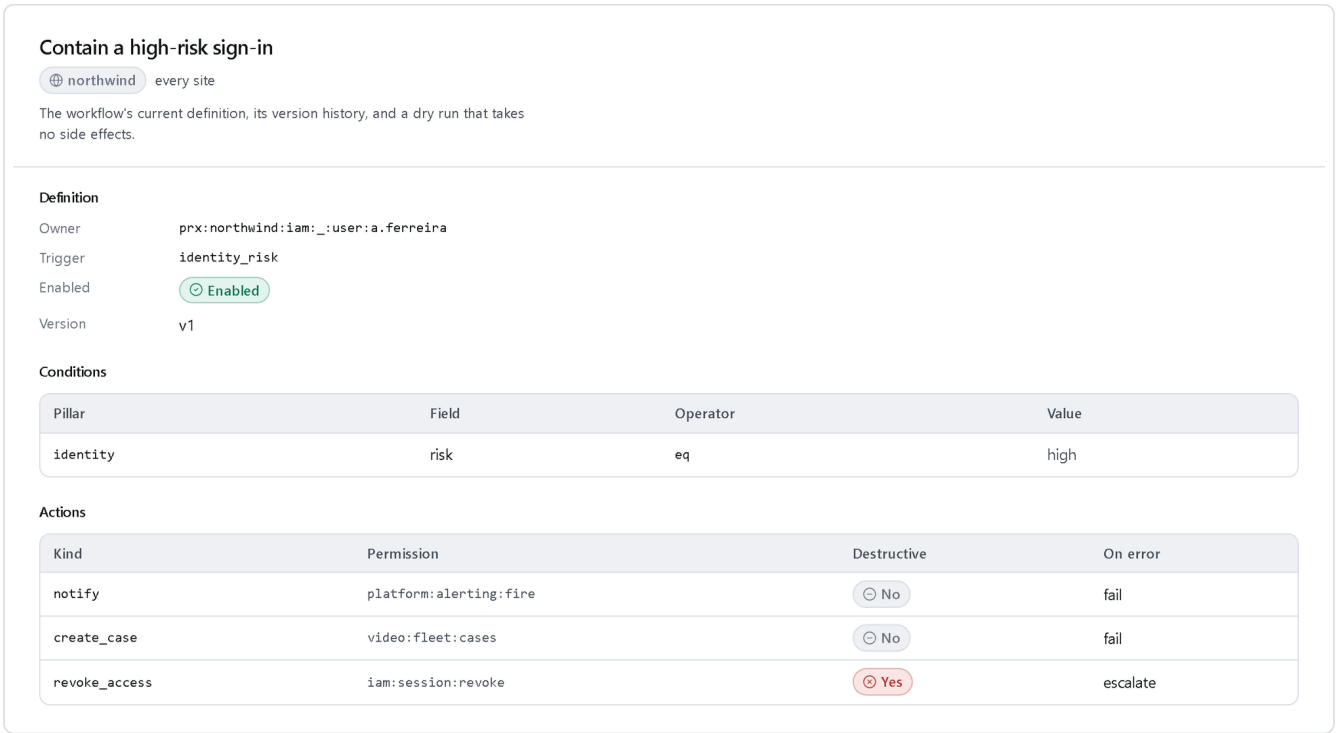


An event from any pillar starts a workflow, or a person starts an agent. Each step is planned, checked against the guardrails and authorized against the policy decision point as whoever started it. Allowed steps act through the pillar or the connector that owns them; destructive ones wait for a person; every step is recorded.

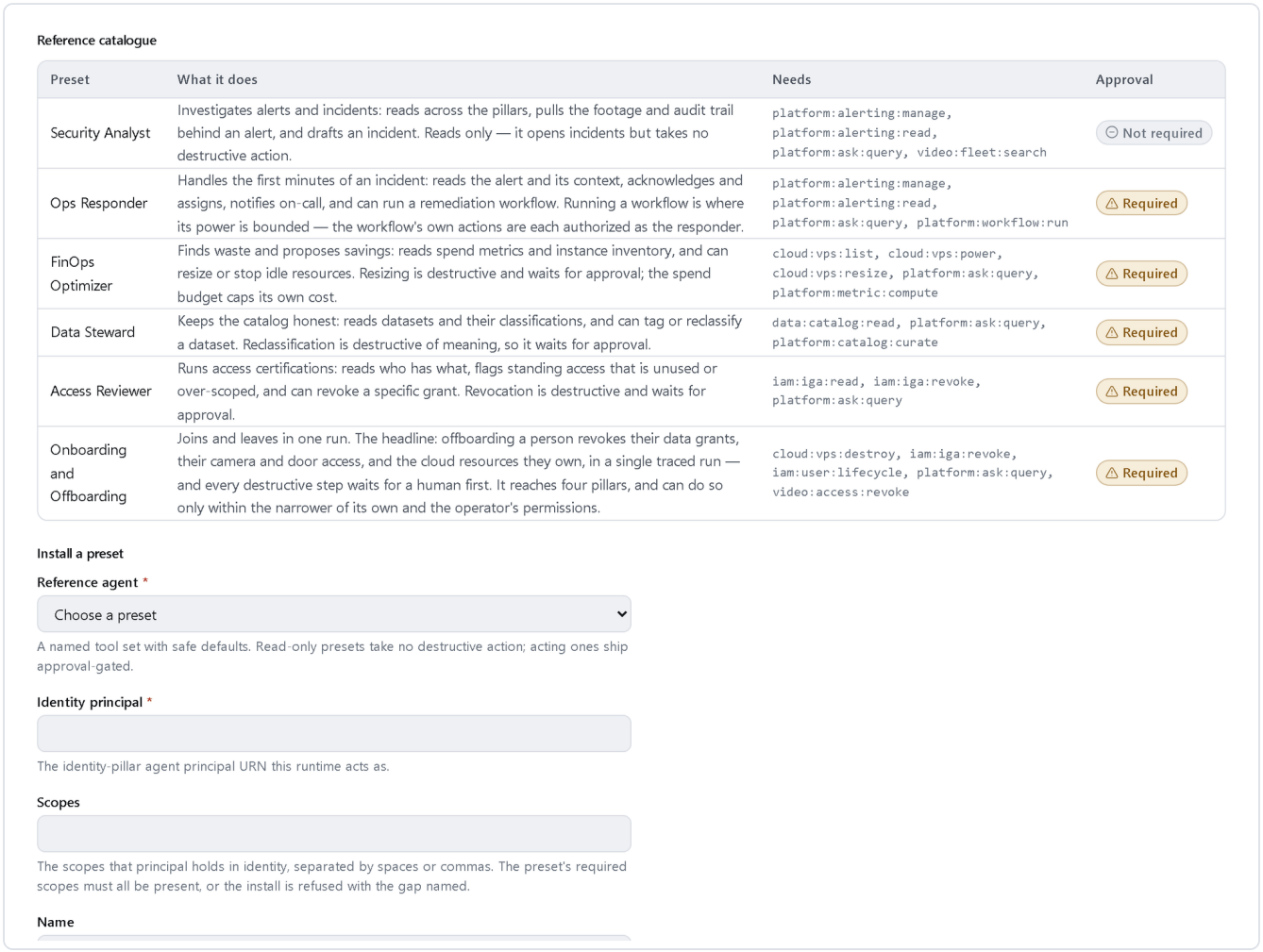
In the console



Every step recorded; the destructive one waits for a person. Screens shown with demo data.



Trigger, condition and actions, each with the permission it needs. Screens shown with demo data.



Reference catalogue

Preset	What it does	Needs	Approval
Security Analyst	Investigates alerts and incidents: reads across the pillars, pulls the footage and audit trail behind an alert, and drafts an incident. Reads only — it opens incidents but takes no destructive action.	`platform:alerting:manage`, `platform:alerting:read`, `platform:ask:query`, `video:fleet:search`	Not required
Ops Responder	Handles the first minutes of an incident: reads the alert and its context, acknowledges and assigns, notifies on-call, and can run a remediation workflow. Running a workflow is where its power is bounded — the workflow's own actions are each authorized as the responder.	`platform:alerting:manage`, `platform:alerting:read`, `platform:ask:query`, `platform:workflow:run`	Required
FinOps Optimizer	Finds waste and proposes savings: reads spend metrics and instance inventory, and can resize or stop idle resources. Resizing is destructive and waits for approval; the spend budget caps its own cost.	`cloud:vps:list`, `cloud:vps:power`, `cloud:vps:resize`, `platform:ask:query`, `platform:metric:compute`	Required
Data Steward	Keeps the catalog honest: reads datasets and their classifications, and can tag or reclassify a dataset. Reclassification is destructive of meaning, so it waits for approval.	`data:catalog:read`, `platform:ask:query`, `platform:catalog:curate`	Required
Access Reviewer	Runs access certifications: reads who has what, flags standing access that is unused or over-scoped, and can revoke a specific grant. Revocation is destructive and waits for approval.	`iam:iga:read`, `iam:iga:revoke`, `platform:ask:query`	Required
Onboarding and Offboarding	Joins and leaves in one run. The headline: offboarding a person revokes their data grants, their camera and door access, and the cloud resources they own, in a single traced run — and every destructive step waits for a human first. It reaches four pillars, and can do so only within the narrower of its own and the operator's permissions.	`cloud:vps:destroy`, `iam:iga:revoke`, `iam:user:lifecycle`, `platform:ask:query`, `video:access:revoke`	Required

Install a preset

Reference agent *

Choose a preset

A named tool set with safe defaults. Read-only presets take no destructive action; acting ones ship approval-gated.

Identity principal *

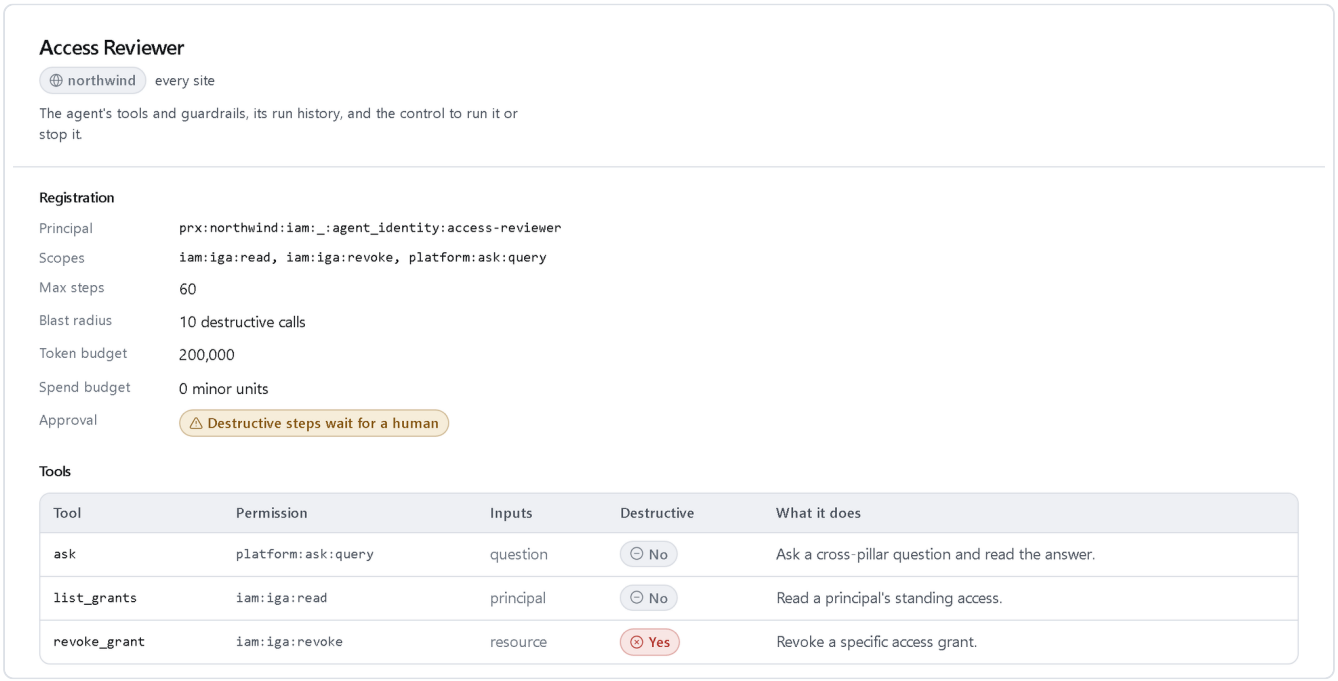
The identity-pillar agent principal URN this runtime acts as.

Scopes

The scopes that principal holds in identity, separated by spaces or commas. The preset's required scopes must all be present, or the install is refused with the gap named.

Name

Six reference agents, each with the permissions it needs. Screens shown with demo data.



Access Reviewer

northwind every site

The agent's tools and guardrails, its run history, and the control to run it or stop it.

Registration

Principal	`prx:northwind:iam:_:agent_identity:access-reviewer`
Scopes	`iam:iga:read, iam:iga:revoke, platform:ask:query`
Max steps	60
Blast radius	10 destructive calls
Token budget	200,000
Spend budget	0 minor units
Approval	Destructive steps wait for a human

Tools

Tool	Permission	Inputs	Destructive	What it does
`ask`	`platform:ask:query`	question	No	Ask a cross-pillar question and read the answer.
`list_grants`	`iam:iga:read`	principal	No	Read a principal's standing access.
`revoke_grant`	`iam:iga:revoke`	resource	Yes	Revoke a specific access grant.

An agent with its own identity, limits and tool list. Screens shown with demo data.

Key ideas

Trigger
What starts a workflow: a schedule, a data condition, a video alert, an identity risk, a cloud anomaly, a webhook or a person.

Dry run
A run that plans and authorizes every action as a real one would, and takes none of them.

Agent identity
The principal an agent acts as, with its own scopes, credentials and budget, separate from the person who runs it.

Blast radius
The most destructive actions an agent may take in one run. One more is refused.

In depth

Authorization per action
Approving a workflow is consent to the workflow, not permission for its actions. Each action is asked about separately, as the principal behind the run, and an agent can never exceed its own identity's permissions.

Budgets that stop
Spend caps and rate limits apply per agent, and a token or currency budget hard-stops a run when it is spent rather than warning after the fact.

A record of every step
Workflow runs keep their inputs and outputs, and agent runs keep a replayable trace of every reasoning step and tool call with its decision.

How it works

01 **Define it** · API
Write the trigger, conditions and actions as a definition, with what each action does on error.

02 **Simulate it** · Console
Run it dry and read what every step would do and whether it would be allowed.

03 **Install an agent** · Console
Pick a reference agent, give it its own principal and scopes, and set its budget.

04 **Review the trace** · Console
Read each run step by step, and see which steps are waiting for a person.

Who it is for

Security operations

A consistent first response to a risky sign-in or a camera going dark, at any hour.

Workflows that notify, open a case and contain access, simulated before they are switched on.

IT and HR

Leavers whose data, door and cloud access all end on the same day.

An offboarding agent that revokes across four pillars in one traced run, each destructive step approved by a person.

Platform and finance teams

Idle resources found and stopped without handing a bot the keys.

A FinOps agent with a spend cap that proposes resizes and stops, and waits for approval to make them.

Works with

Investigator console · API · prx CLI · TypeScript SDK · Python SDK · Slack, Teams, PagerDuty, ServiceNow and Jira

Questions buyers ask

How are workflows written?
As definitions through the API, CLI and SDKs, which you can keep in version control. The console shows each workflow with its definition, dry runs, run history and versions.

What can an agent do without asking?
Only what its own scopes and its user's permissions both allow. Destructive or high-cost actions wait for a person, and a blast-radius limit caps how many it may take in one run.

Does an agent need a language model?
No. You can give an agent its plan, or let a model-driven planner propose the tool calls; the proposals are held to the same guardrails and widen nothing.

How do we stop one?
A kill switch per agent, per tenant or globally takes effect within 5 seconds, before the agent's next step.

How is it priced?
Workflows are billed per 1,000 executions and agents per run, at published rates in Platform Credits. A dry run is free. The rates are on the pricing page.

Pricing

Current published rates, in your currency: https://adaydev.com/pricing/products/workflows