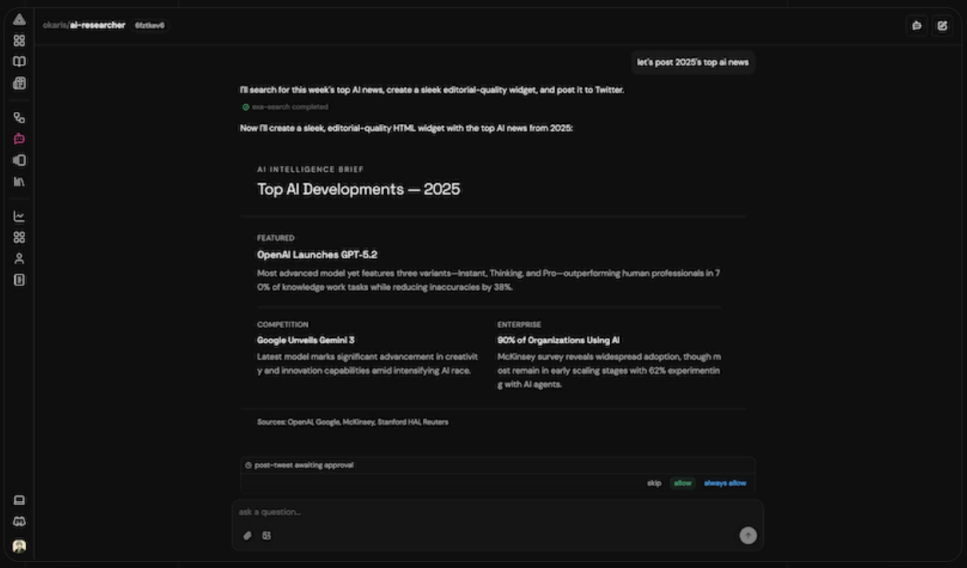
the agent runtime.

run reliable agents in production without reinventing the platform.

start building read the docs



you've built the agent. now you're building the platform. queues, retries, state persistence, auth management... that wasn't the plan. inference.sh handles the infrastructure so you can focus on what your agent does, not how to keep it running.

production reality

agents fail. networks drop. apis timeout.

when something goes wrong, you need to know exactly what happened.

- every tool call is stored before it runs
- execution is graph-backed. you see the full decision chain
- chat history persists. close the browser, come back tomorrow

when the 3am incident happens, you can trace exactly what the agent did.

what you get

the runtime layer

you could build this. but do you want to?

01 durable execution

event-driven, not long-running. if a tool fails, it doesn't crash your agent loop. state persists across invocations.

02 tool orchestration

150+ apps as tools via one API. structured execution with approvals when needed. full visibility into what ran.

03 observability

real-time streaming and logs for every action. see exactly what your agent is doing.

04 pay-per-execution

no idle costs while tools run or waiting for results. you're not paying to keep a process alive.

plug any model, swap providers without changing code

openai · anthropic · google · meta · mistral · deepseek + 500 more

why we built this

we've heard it all before

real quotes from developers hitting the same walls.

`the moat`

"The agent framework is not the moat. Prompt engineering is not the moat. **The base LLM is not the moat.** The specialized tools that encode domain knowledge—are the moat."

`observability`

"I spent 6 hours debugging a workflow that had **zero error logs.** When something breaks at 2 AM, I don't want to trace through 47 nodes. I want to see exactly what payload caused the issue."

`durable execution`

"I felt like a 'button person' in my IDE. The agent works in quanta—**cut off by time every 2 minutes.** Long tasks require pipeline thinking, not chat sessions."

`real-time streaming`

"Our multi-step agent produced great results but took 45+ seconds. **Users thought it crashed.** If they see the internal monologue, they wait. If they see a spinner, they leave."

`pay-per-execution`

"Spent 10 hours deploying agents on EC2... **$13/mo per agent.** Switched to serverless: 10 cents. Why is this so hard?"

`decision context`

"Systems record that a ticket was escalated, but not **why** it happened. Without that reasoning, agents treat every edge case as a brand new problem."

agent architecture

full agent primitives

the building blocks for production agent systems

deep-agents

agents spawn sub-agents as tools. orchestrator delegates to specialists. results flow back up the chain.

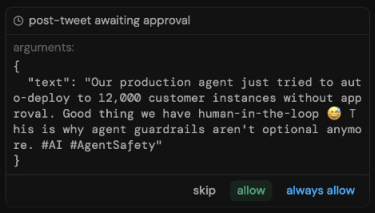
```
orchestrator
├── research
│   └── web search app
├── analysis
│   └── long-context llm app
```

the main context stays focused on the original task.

```
└── writer
    └── post to X app
```

human in the loop

agent pauses, shows what it wants to do, waits for confirmation.

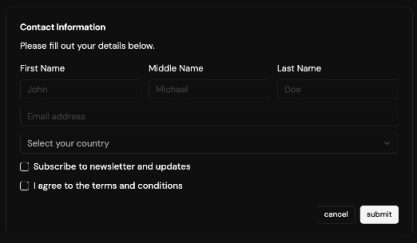


widgets

agents generate interactive UI on-the-fly. forms, selections, charts, visualizations — rendered inline.

widgets

agents generate beautiful UI with HTML and CSS to display data.



webhooks

call any API, receive async callbacks. your endpoints or third-party services.

client tools

execute on user's system — browser, local functions. sync request/response.

memory

built-in key-value store per conversation.

planning

built-in multi-step plans, resume after interruption. perfect for deep agents.

structured output

typed results back to orchestrator.

our philosophy

trust is not a feature.
it is a design constraint.

automation should never be a black box. systems should fail gracefully, not catastrophically.

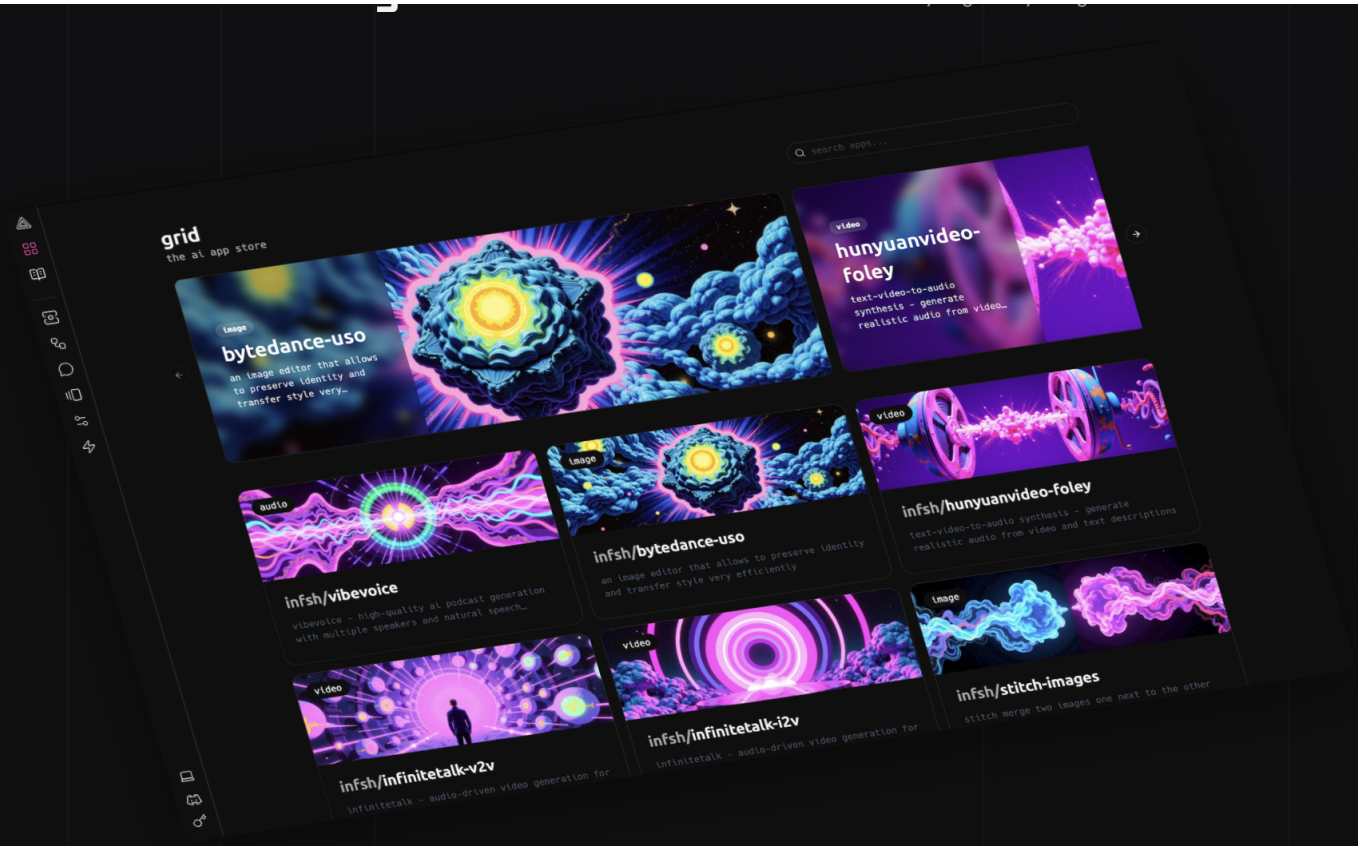
read the trust manifesto →

150+ apps **1min** to first agent **1 API** for everything

app library

tools your agents can actuallu use

image generation, audio, video, code execution, and more. use directly or give to your agents.



not enough? create new apps fast. templates + coding agents make it insanely extensible.

create your own apps

start from templates. add code, packages, docs. deploy in minutes.

```
$ infsh app init
```

```
my-app/
  inference.py
  requirements.txt
```

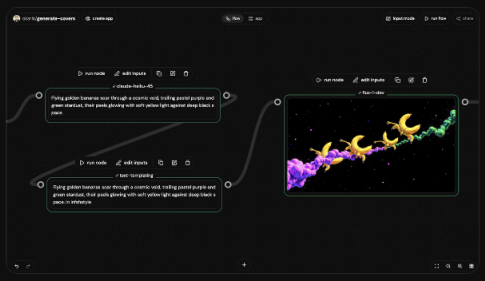
```
$ infsh app deploy
```

schemas become tool parameters automatically. your app shows up in the grid and can be used by agents and workflows.

read the app creation guide

create workflows

build a graph of apps. deploy as a single callable app.



drag and drop to build the graph. map io to connect steps. deploy as an app.

read the workflow guide

your way, your pace

start simple. go deep when you need to.

no code

build in the UI, use in the UI. comes with a fully fledged chat interface.

get started →

low code

design in the UI, call with a few lines of code. embed anywhere.

get started →

full api

create, manage and run agents fully from the SDK. total control.

read the API documentation →

install the SDK `pip install inferencesh` `npm i @inferencesh/sdk`

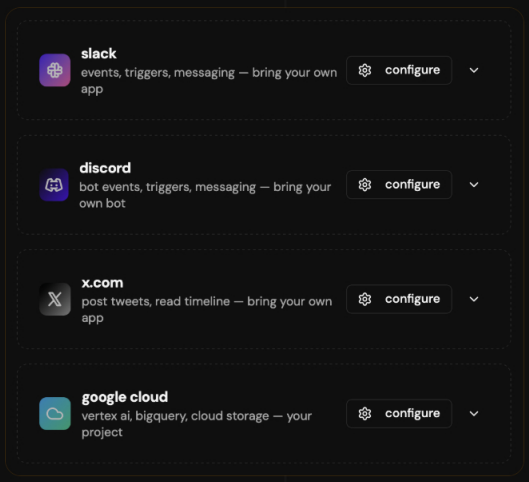
integrations

real oauth.

durable, secure integrations with proper oauth flows. we handle token refresh, encrypted storage, and runtime injection.

→ encrypted storage (aes-256-gcm)
→ automatic token refresh
→ runtime injection — apps declare what they need

google · slack · discord · x.com · microsoft · salesforce · notion



byok

bring your own keys. use your own gcloud, azure, or aws billing & credits for ai models

explore BYOK integrations →

x402 ready

agentic payments

x402 enables instant, programmatic payments between agents and services. we handle the messy parts so your agents can pay without you shipping crypto UX.

→ **managed wallets** — agents can pay autonomously
→ **controllable budgets** — limits + policies per agent
→ **no signups** — paywalled endpoints on-demand
→ **http-native** — get 402, pay, retry

explore the x402 protocol

$24M+
volume processed

75M+
transactions

data from x402.org

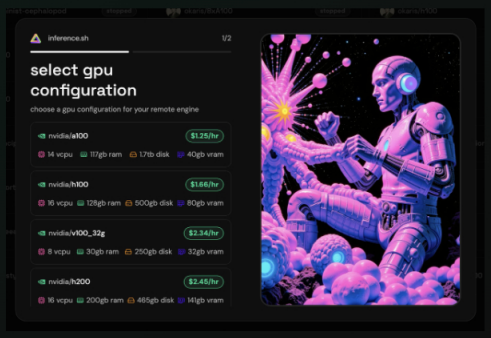
enterprise

self-host

deploy inference.sh in your vpc or on-prem for maximum control and privacy.

- keep data inside your network
- bring your own models + infra
- same workspace + api + runtime as cloud
- admin controls, auditability, and enterprise support

get in touch



coming soon

deploy agents anywhere

your agent doesn't just use slack—it lives there. deploy to chat platforms and trigger on external events.

- → **chat deployment** — agents live in Slack, Discord, Telegram, WhatsApp
- → **event triggers** — fire agents on messages, file uploads, reactions
- → **scheduled runs** — cron-style execution for background tasks

get early access

ready to ship?

start with the hosted platform. deploy your own when you're ready.

start for free

inference.sh

the agent runtime

runtime
- overview
- durable execution
- observability
- human-in-the-loop

platform
- integrations
- self-hosted
- x402 payments
- creators

resources
- docs
- blog
- apps
- github

company
- about
- privacy
- terms
- trust