ProjectX

SERVERS · DATASHEET

Virtual and dedicated servers from one catalog.

Launch a VPS in minutes or order dedicated hardware, both through the same lifecycle and the same API. Choose an image, template the first boot, and get the whole order-to-running record audited — on capacity we run and partner capacity we contract.

Why teams choose it

01
Virtual or dedicated, one way of working
A VPS in minutes or single-tenant hardware — the same lifecycle verbs, the same API and the same audit record either way, so your tooling never forks.

02
The state you see is the state that runs
The platform reconciles its record against what the provider is actually running, so drift surfaces on your dashboard instead of hiding until an incident.

03
No secrets baked into images
First-boot configuration uses a fixed, allow-listed templating syntax, so a golden image stays clean and reusable across every server you launch.

04
Your images stay yours
Custom and golden images are placed per tenant, per region — an image you build is never visible to another customer.

In numbers

< 90 s
order to running, at the 95th percentile

< 5 min
from sign-up to a running server, with no person involved

1
API, audit record and invoice for virtual and dedicated alike

What it does

Full VPS lifecycle

Create, resize, start, stop, reboot, rebuild and destroy, each a single call with the state the platform holds reconciled against what the provider is actually running.

Every verb works the same from the console, the API, the SDKs and the CLI. Resize keeps the disk, rebuild starts from a fresh image while the server's record and history stay put, and the reconciler keeps the record honest against the provider.

Dedicated servers

Order, provision, reinstall and decommission single-tenant hardware through the same resource model as a VPS, with the places the two genuinely differ — a machine that cannot be held stopped — surfaced rather than hidden.

Order from the published catalog and follow the provisioning run step by step in the console. Reinstall when you need a clean machine, decommission when you are done — same verbs, same audit record as everything virtual.

Image catalog

Marketplace, custom and golden images, placed per tenant so an image built in one region is yours and not the whole platform's.

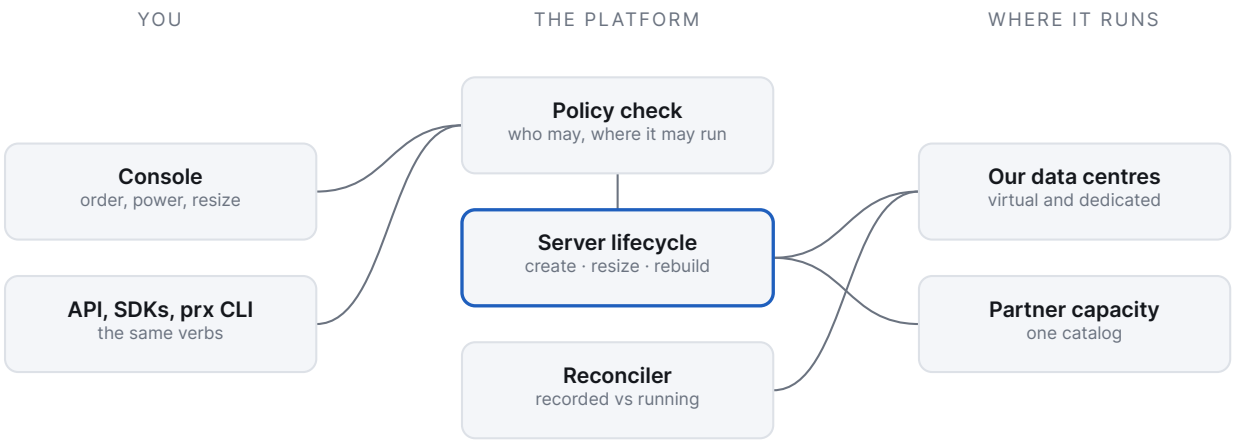
Start from a marketplace image, register your own, or promote a tested build to a golden image the whole team launches from. Placement is per tenant and per region, so your images never leak into another customer's catalog.

cloud-init templating

A fixed, allow-listed templating syntax for first-boot configuration, so a server comes up configured without a secret being baked into an image.

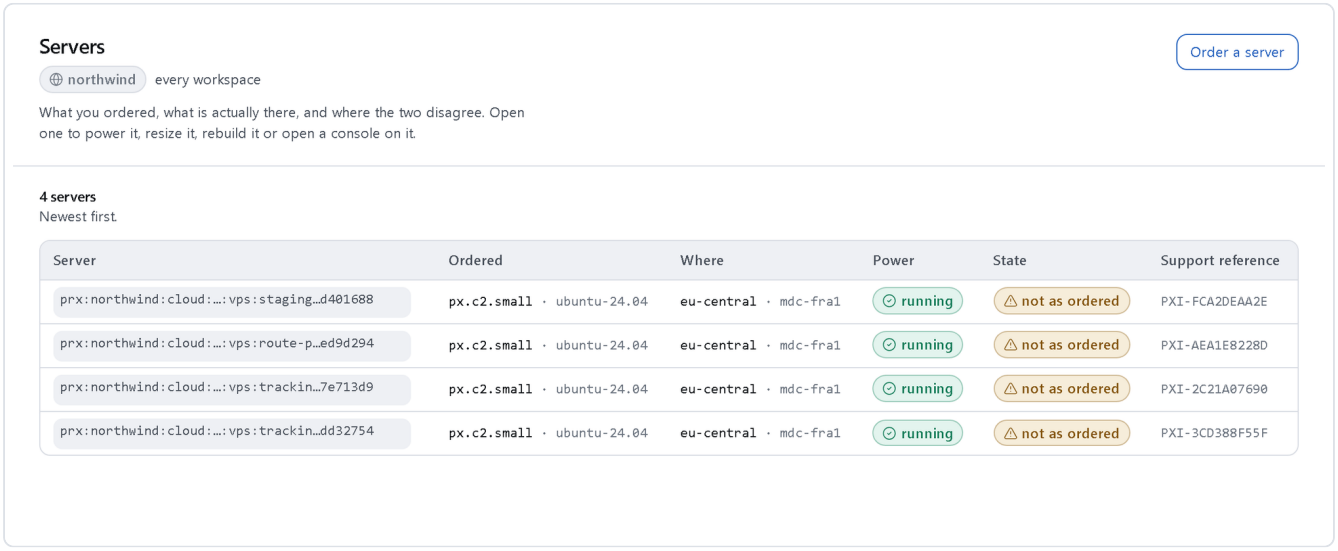
Hostname, users, packages and files are filled at first boot from a template whose syntax is fixed and allow-listed. The golden image stays generic and reusable, and there is no credential sitting inside it to forget about.

How it fits together

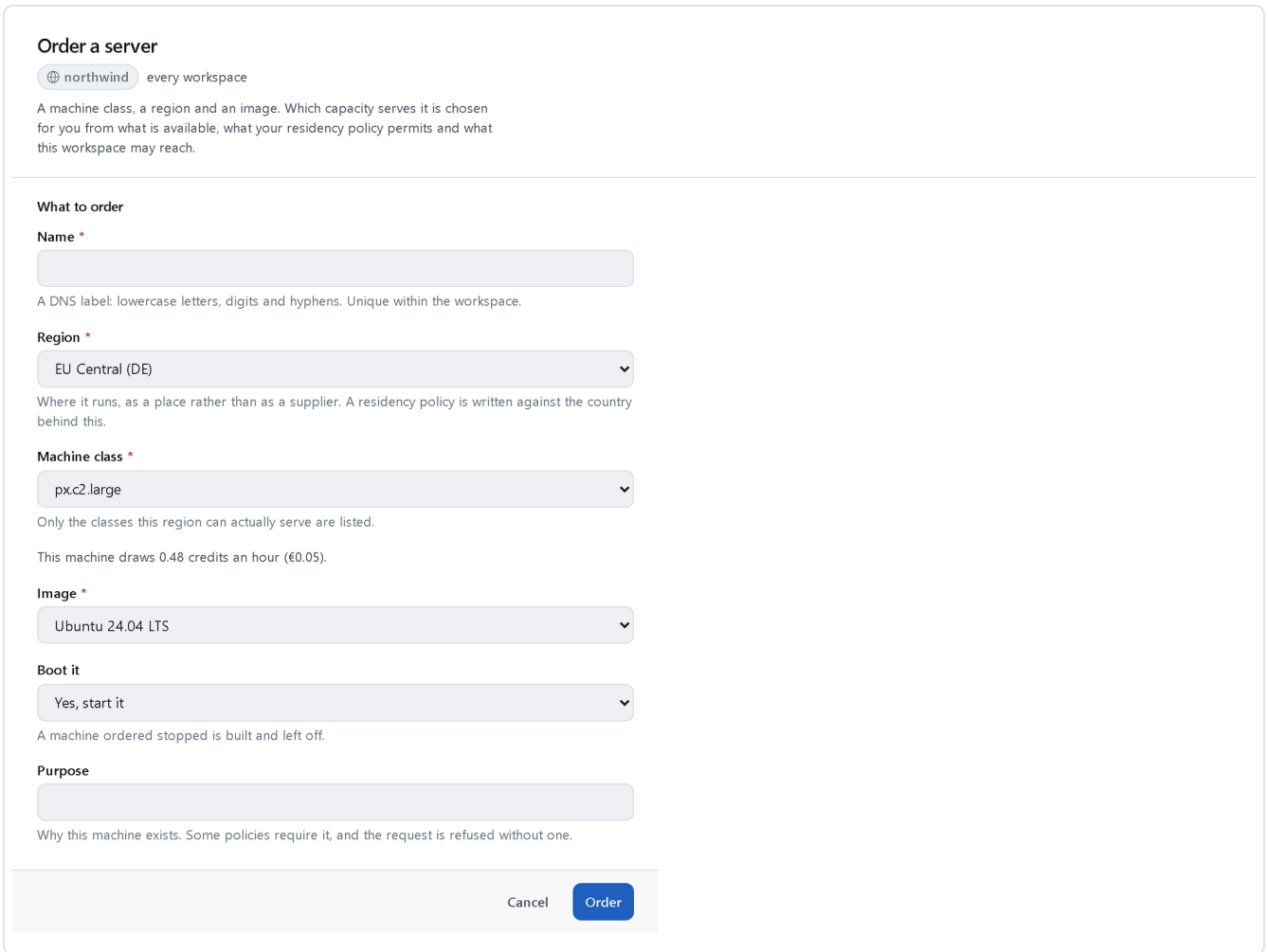


You order a server in the platform's own vocabulary from the console, the API, the SDKs or the CLI. The platform checks the order against policy, places it on a substrate that satisfies your residency rules, and keeps reconciling what it recorded against what is actually running, with every step on the audit chain.

In the console



Every server in one list, with its state reconciled against what is actually running. Screens shown with demo data.



Pick a size and a region; the price is on the page before you order. Screens shown with demo data.

Key ideas

Machine class
A size in the platform's own vocabulary, such as px.c2.small. You order the class and the region; which capacity serves it is the platform's decision, so your scripts never name a provider.

Desired and observed state
What you ordered and what is actually running are recorded separately. The reconciler compares them continuously, and a difference shows on the server rather than staying hidden.

Golden image
A tested image your team promotes and launches from. It stays generic, because per-server settings arrive at first boot from a template rather than being baked in.

Substrate
Where a server physically runs: our data centres, contracted partner capacity, or hardware you own and register yourself. Residency rules decide which substrates an order may use.

In depth

Security model
Every lifecycle call is a permission question answered by the platform's one policy engine, and every answer — allowed or refused — is written to the append-only audit chain. On hardware you own, the platform holds no standing access.

Placement and residency
An order names a class and a region. Residency obligations filter the possible capacity before one is chosen, so a placement your policy forbids is never made — rather than made and flagged afterwards.

Operations
Power, resize, rebuild, rescue, snapshot and clone are verbs on the server, each one call from the console, the API, the SDKs or the CLI. Start and stop schedules keep development machines from billing overnight.

How it works

01 **Pick a size and a region** · Console
Choose a compute class and where it runs — the price is on the page before you order.

02 **Configure the first boot** · cloud-init
Add an image and a first-boot template; the server comes up configured, with no secrets baked in.

03 **Launch with one call** · API
The console, the SDKs and the CLI all issue the same create — the server is typically running within minutes.

04 **Operate the whole lifecycle** · prx CLI
Start, stop, resize, rebuild and destroy from one console or one API call, each action audited.

Who it is for

SaaS teams

You ship an application and need predictable compute you can script end to end.

Servers you create, resize, rebuild and destroy from one API, with every action audited.

Agencies and hosters

You run many small sites and services and want one bill across all of them.

Per-hour classes at published rates, every server on the same monthly invoice.

Platform engineers

You need single-tenant hardware for licensing, isolation or performance.

Dedicated servers ordered, provisioned and reinstalled through the same resource model as a VPS.

Works with

Console · API · TypeScript SDK · Python SDK · prx CLI · cloud-init

Questions buyers ask

How quickly can I get a server?
A VPS is created with one call and is typically running within minutes. Dedicated hardware goes through a provisioning run you can follow step by step in the console.

Which operating systems can I run?
The image catalog carries marketplace images, and you can register your own custom and golden images per tenant, placed in the region you choose.

Can I automate everything?
Yes. Every lifecycle verb is an API call, and the TypeScript SDK, the Python SDK and the prx CLI cover the same surface as the console.

How is it billed?
By the hour at the published rate for the class and region, rolled into one monthly invoice with everything else you use. The full rate card is on the pricing page.

Pricing

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