ProjectX

A SAAS ON PREDICTABLE COMPUTE · USE-CASE BRIEF

Run your product on servers, Kubernetes and databases with a bill you can predict.

Servers and clusters provisioned in seconds, managed databases with point-in-time recovery, schedules and autoscaling so nothing idles, and budgets and caps that stop spend before the invoice does.

The situation

The product runs on a hyperscaler whose bill changes every month for reasons that take a week to explain. Development environments run all weekend and a database restore is a support ticket.

How it works, step by step

01 **Run the application** · Kubernetes
Servers with a full lifecycle from create to destroy, or managed and self-managed Kubernetes clusters with node pools that autoscale.

02 **Keep the data safe** · Managed databases
Managed PostgreSQL, MySQL, Redis and more, with replicas, failover and automated backups with point-in-time recovery.

03 **Stop paying for idle** · Servers
Start and stop schedules for development servers, and autoscaling node pools for production.

04 **Know the price in advance** · Cloud prices
Hourly and monthly metering per resource at published rates, with proration on resize exact to the cent.

05 **Set the ceiling** · Pricing
Budgets, hard caps and prepaid-only accounts; an empty balance alerts first and refuses new provisioning.

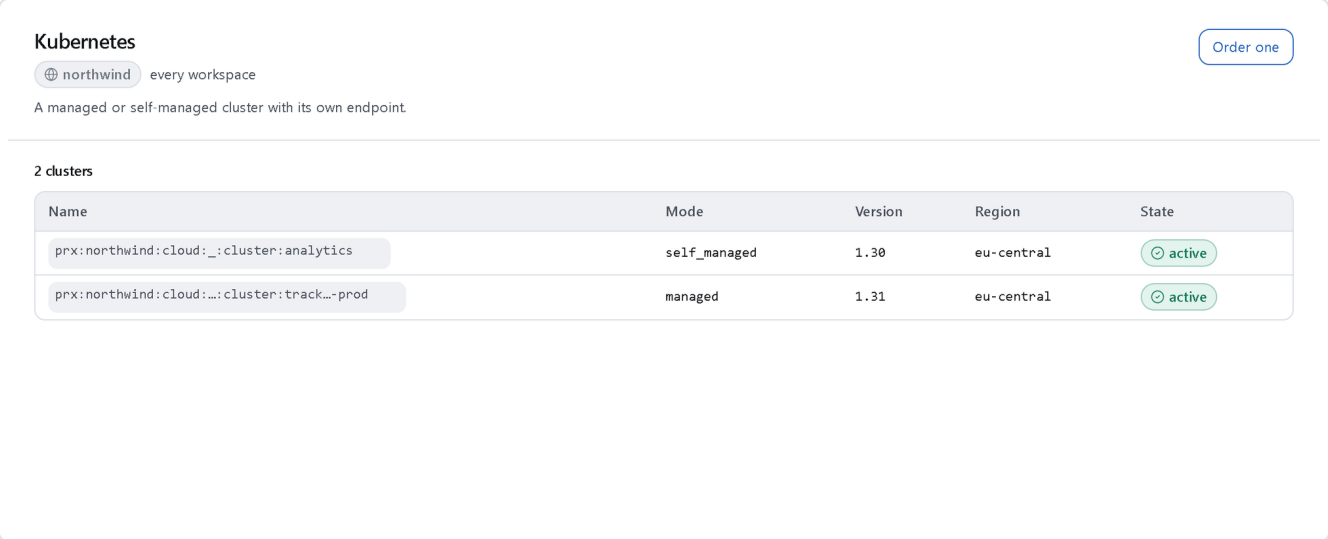
What you get

01
Running in minutes
From sign-up to a running server in under five minutes, with no human involved.

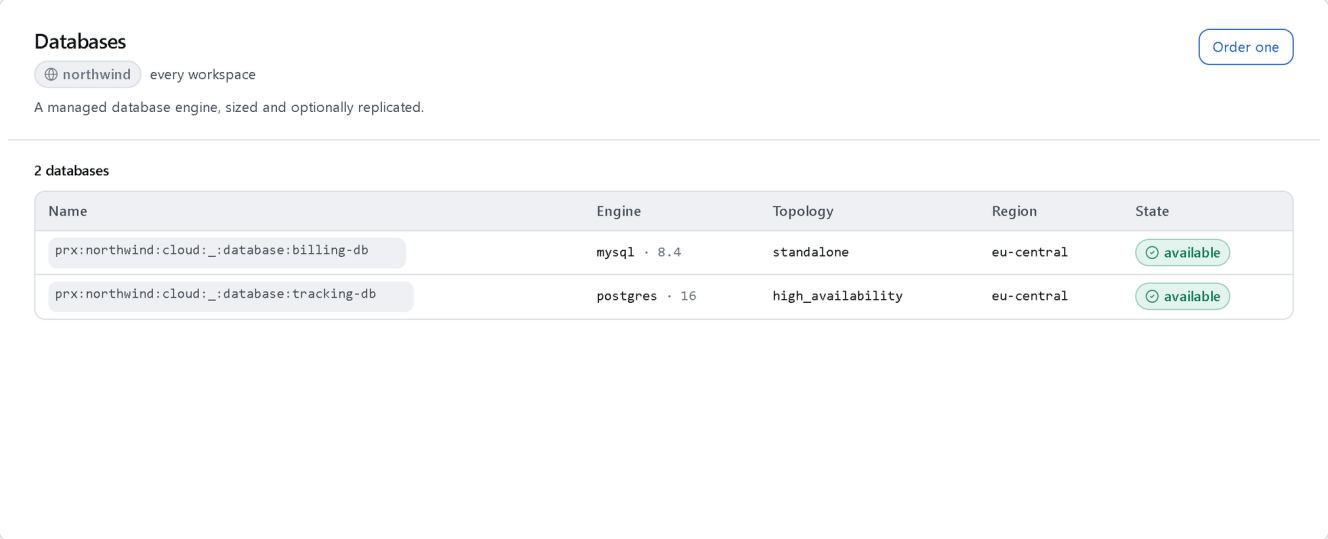
02
A bill that matches the plan
Published rates per resource, metered hourly and prorated exactly.

03
Restores without a ticket
Point-in-time recovery and failover drills are in the console.

In the console



Managed and self-managed clusters side by side, one resource each. Screens shown with demo data.



Every engine in one list, each with its topology and version. Screens shown with demo data.

Questions buyers ask

Can we use our existing cloud accounts?
Managed Kubernetes can be provisioned across GKE, AKS, EKS, LKE, DOKS and others, alongside servers on the platform's own substrate.

What happens when the balance runs out?
Low and empty alerts come first; then new provisioning is refused, and compute pauses after a grace period.