Take Back the Cloud

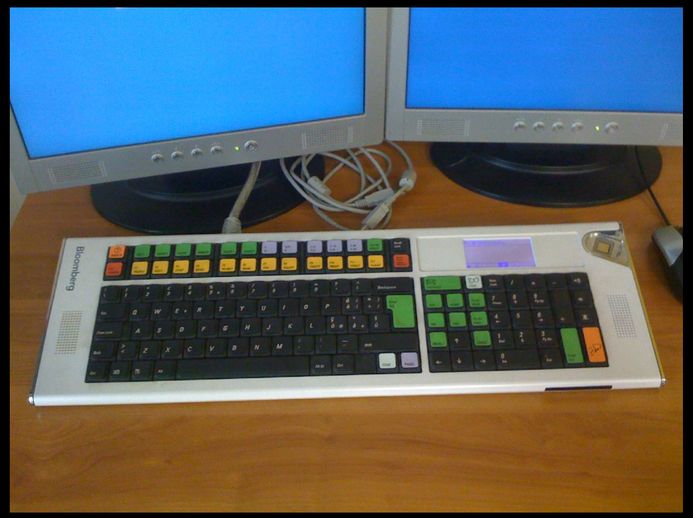
The End of Traditional SaaS

Steve Sklar

A Little About Me

- First “programming language” was Excel VBA
- Work for QuestDB (questdb.io)
- Blog semi-regularly at sklar.rocks
- Can be found on GitHub “@sklarsa”

QuestDB



Software Delivery Models

Self-Hosted

Self-Hosted

Deliver software as artifacts

Customer is responsible for operations

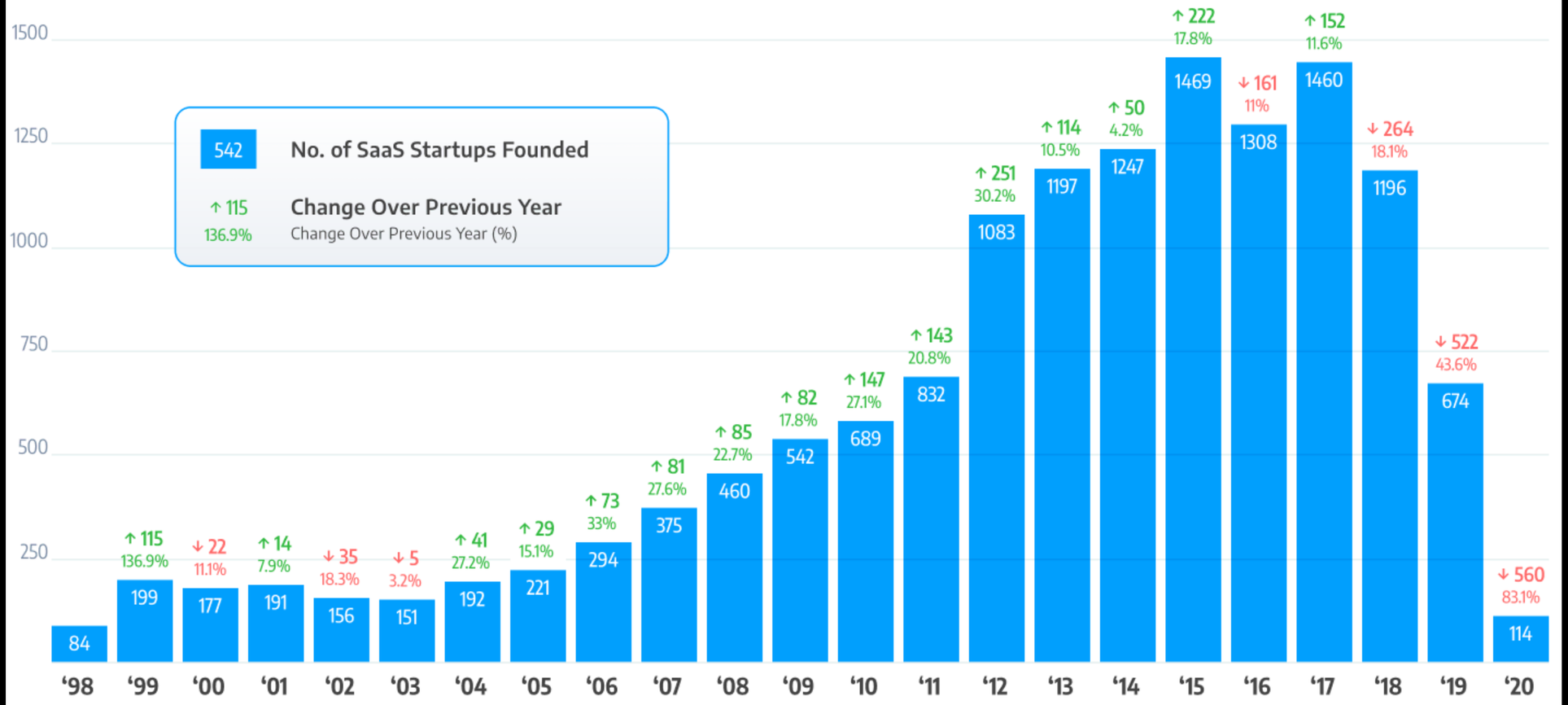
Support requires high-levels of coordination

Vendor has no control over execution environment

Customer owns the data


Self-Hosted
SaaS

New SaaS Startups Launched By Year



Source: The Latka Agency, SaaStr, Crunchbase, CardConnect

Ascendix

SaaS

Deliver software as a service

Vendor is responsible for operations

Vendor has complete control over execution environment

Vendors owns the data

Unpredictable pricing

Outage risk


Self-Hosted
BYOC
SaaS

What is BYOC?

What is BYOC?

Bring Your Own Cloud

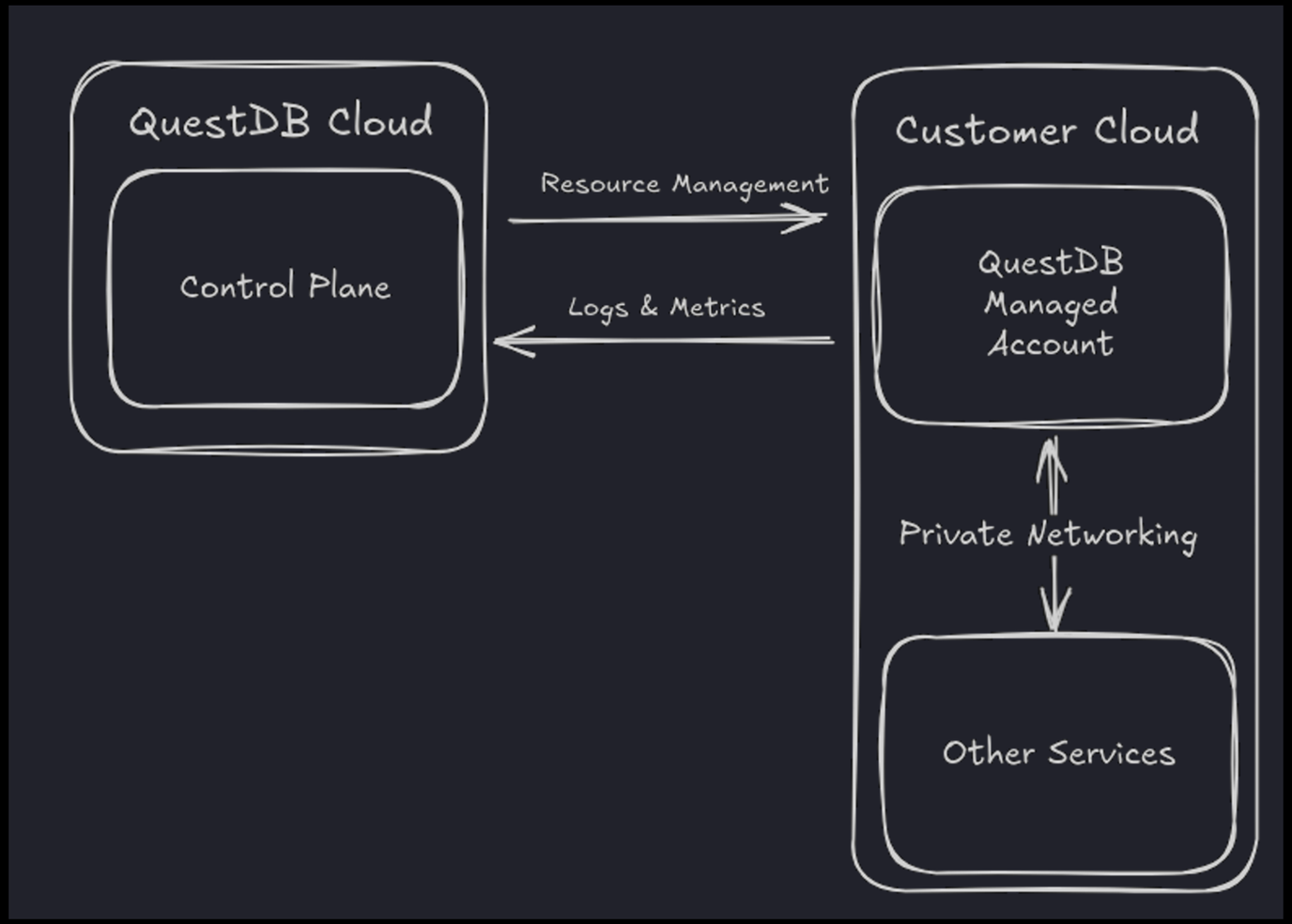
Deliver software as artifacts

Vendor is responsible for operations

Most support can be done by the vendor

Vendor has control over the execution environment

Customer owns the data

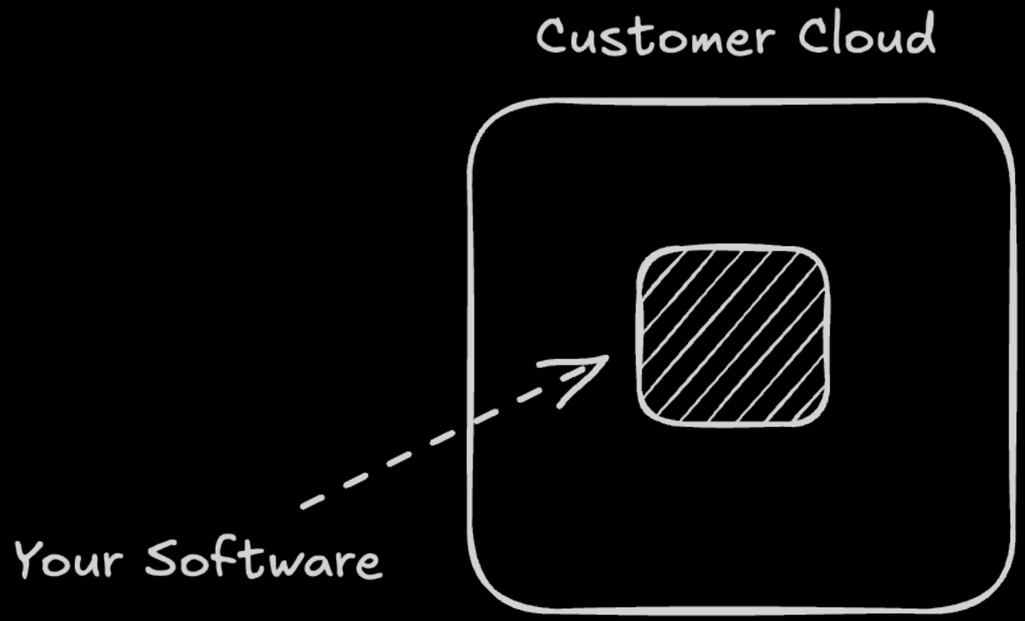


	Self Hosted	BYOC	SaaS
Infrastructure Admin	Customer	Vendor	Vendor
Data Owner	Customer	Customer	Vendor
Who Pays Infrastructure Costs?	Customer	Customer	Vendor
Networking	Private	Private	Public
Initial Time to Launch	Months/Weeks	Weeks/Days	Minutes/Seconds

BYOC Architecture

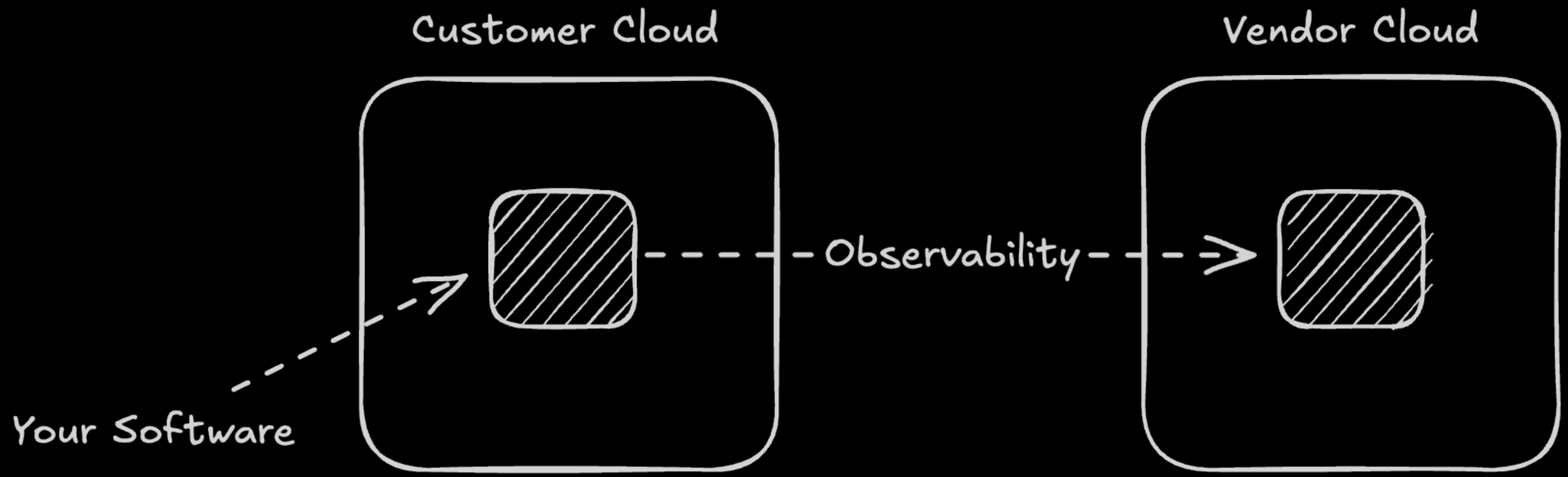
BYOC Architecture

Level 0: Self-Hosted



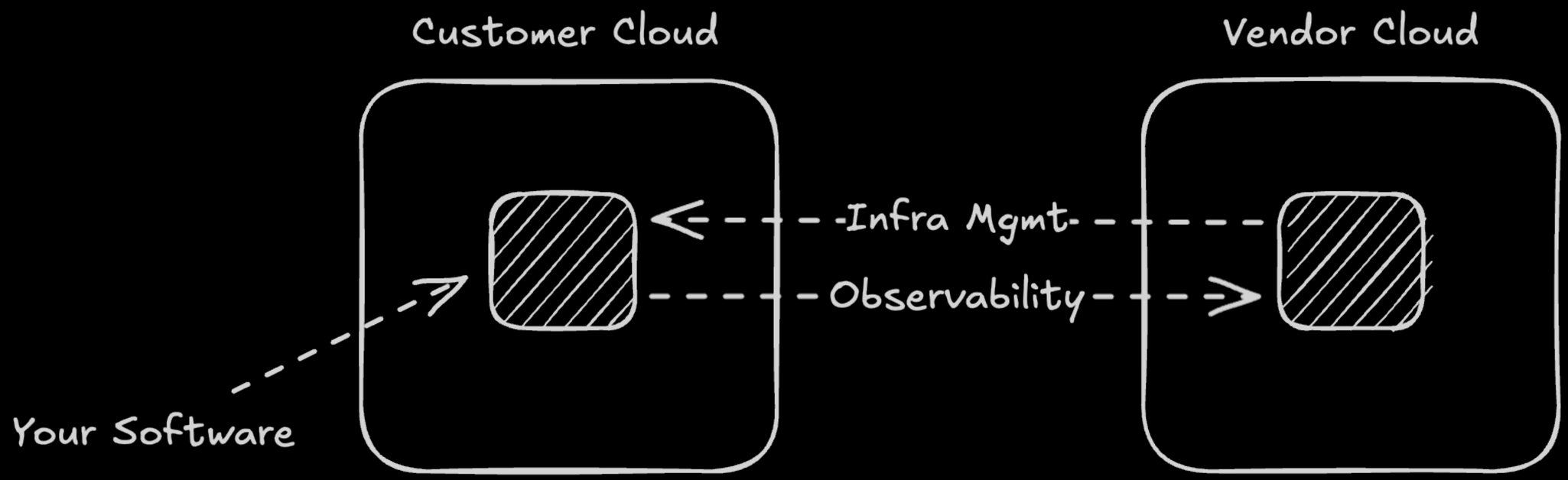
BYOC Architecture

Level 1: Observability



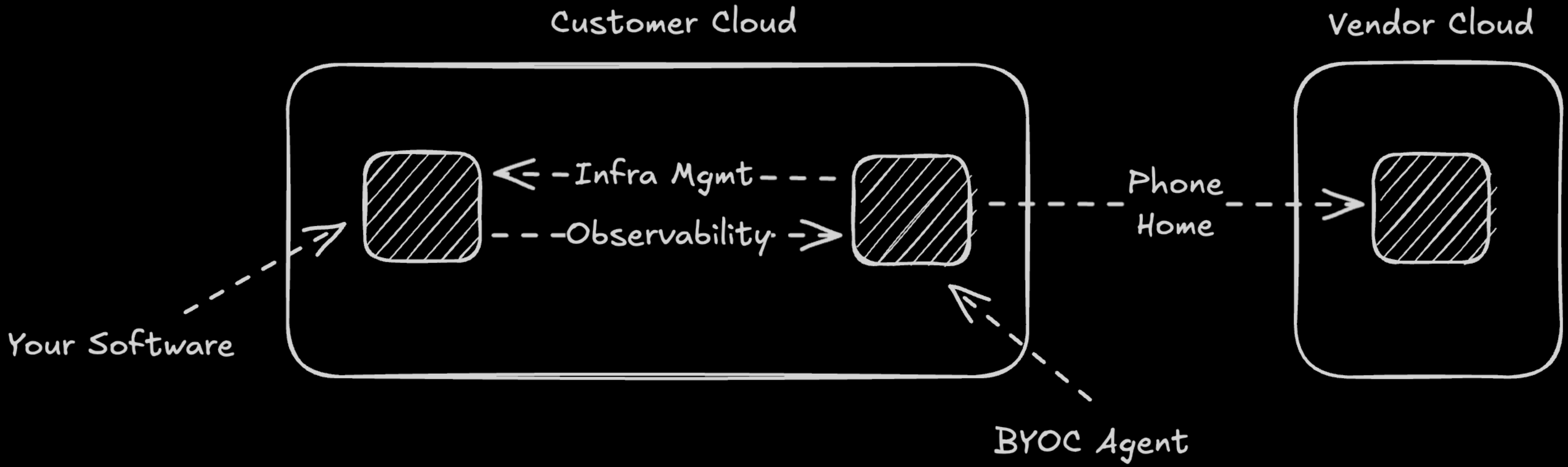
BYOC Architecture

Level 2: Delegation of Responsibility



BYOC Architecture

Level 3: Agent-Based

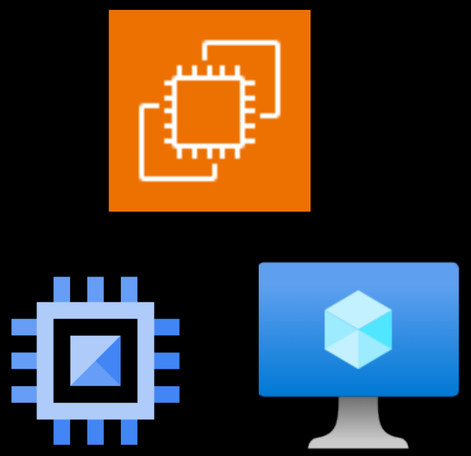


BYOC Technology

BYOC Technology

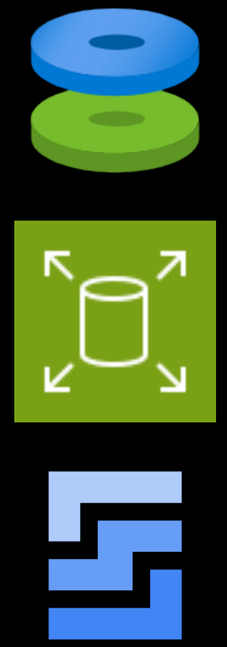
Cloud Building Blocks

Compute



Networking

Block Storage

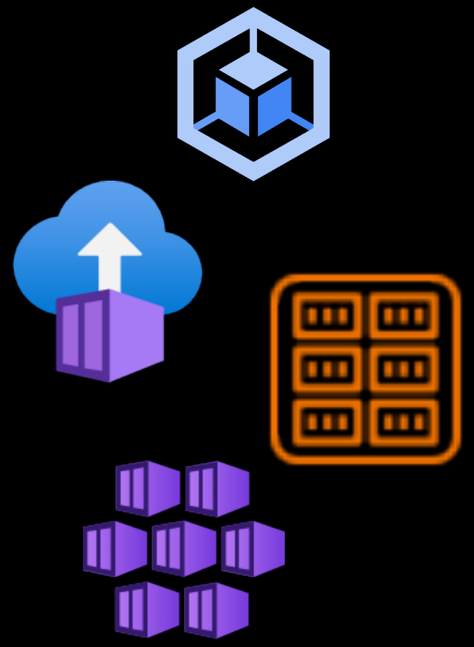


Blob Storage

BYOC Technology

Cloud Managed Services

Containers



Databases

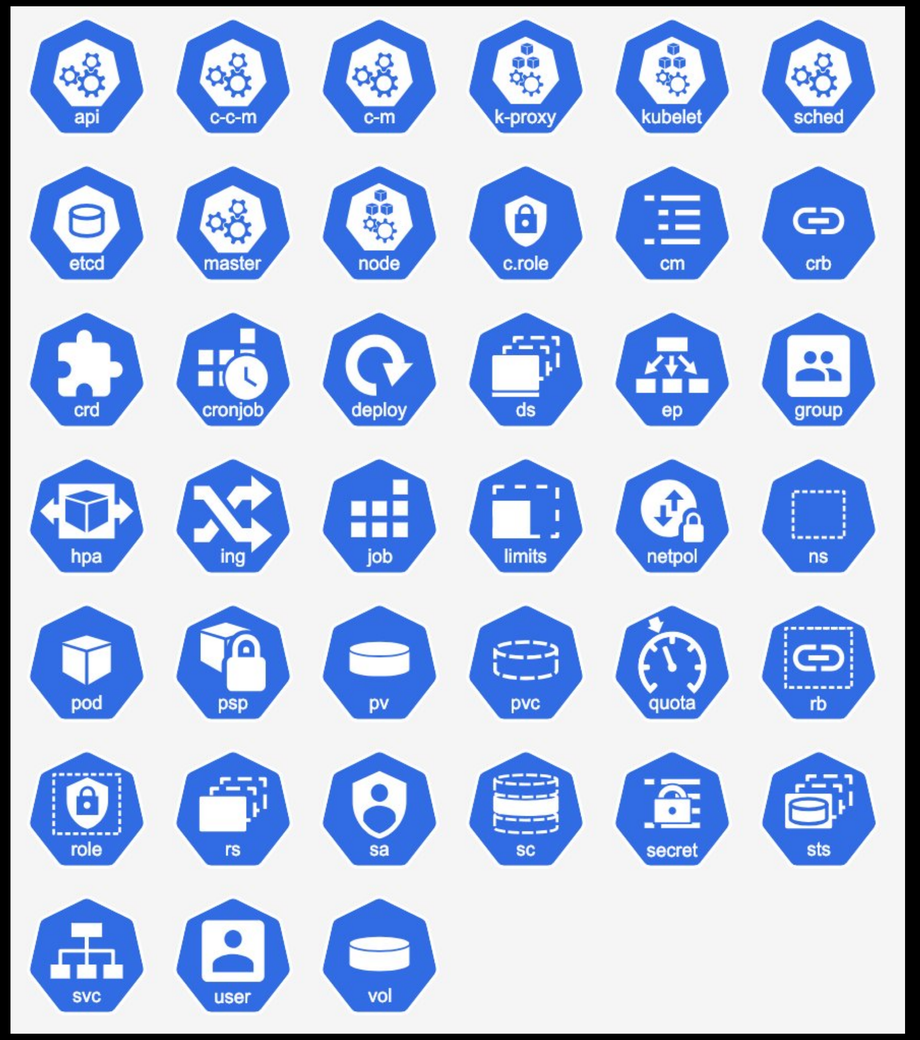
PaaS

Serverless

BYOC Technology

Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: k8s-deployment-nginx-v2
  labels:
    app: nginx
spec:
  replicas: 4
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.23.1
        ports:
        - containerPort: 80
```



BYOC Technology

Infrastructure as Code

Reproduceability across environments and customers

Increased levels of automation — no “click-ops”

Unlocks the power of GitOps

Versioned infrastructure

Integration (and sometimes unit) testable

HashiCorp Terraform

aws CDK

Pulumi

BYOC Technology

Observability

Open vs Closed Source Solutions

Balance configurability with ease of use

Pillars of Observability (v1)

Telemetry (v2)

Tune your alerts!

OpenTelemetry

BYOC Challenges

Challenges
Networking

Coordination between Customer and Service Provider

Can get expensive quickly, if you’re not careful

- Egress and cross-AZ charges

Who manages DNS entries? What about the resolver?

- Private networking can make this difficult

Who owns the root certs? How are certs renewed?

Challenges

Security

Internal RBAC and best practices

- SSO, multifactor auth, up-to-date group membership

Cloud security scans

Certifications

- SOC2 vs ISO 27001

Challenges

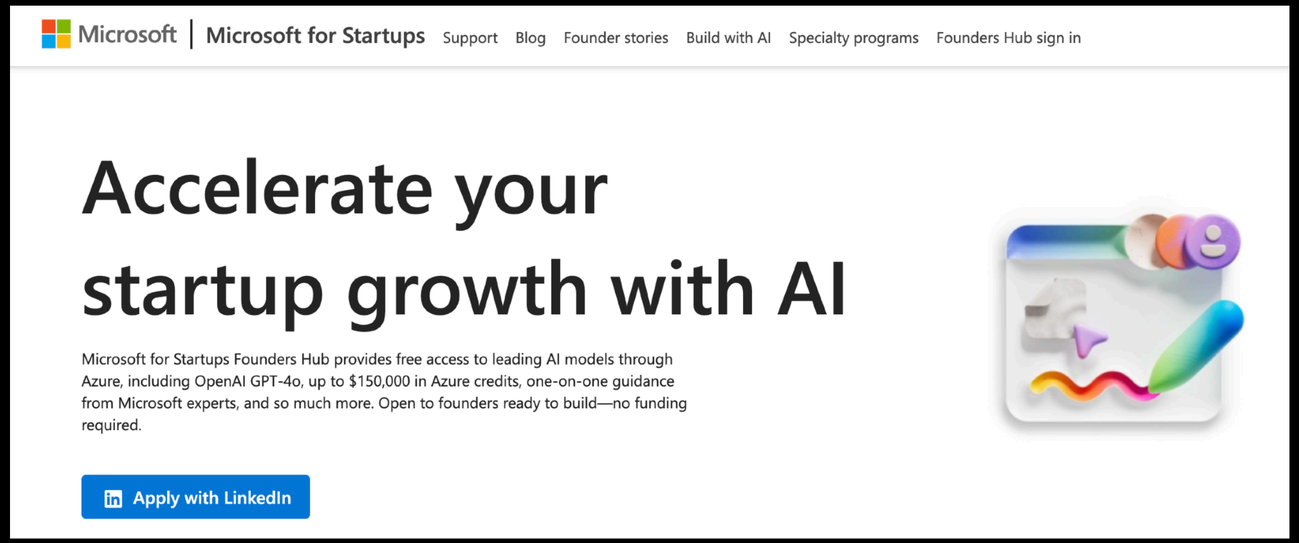
Pricing

Customer pays for infra

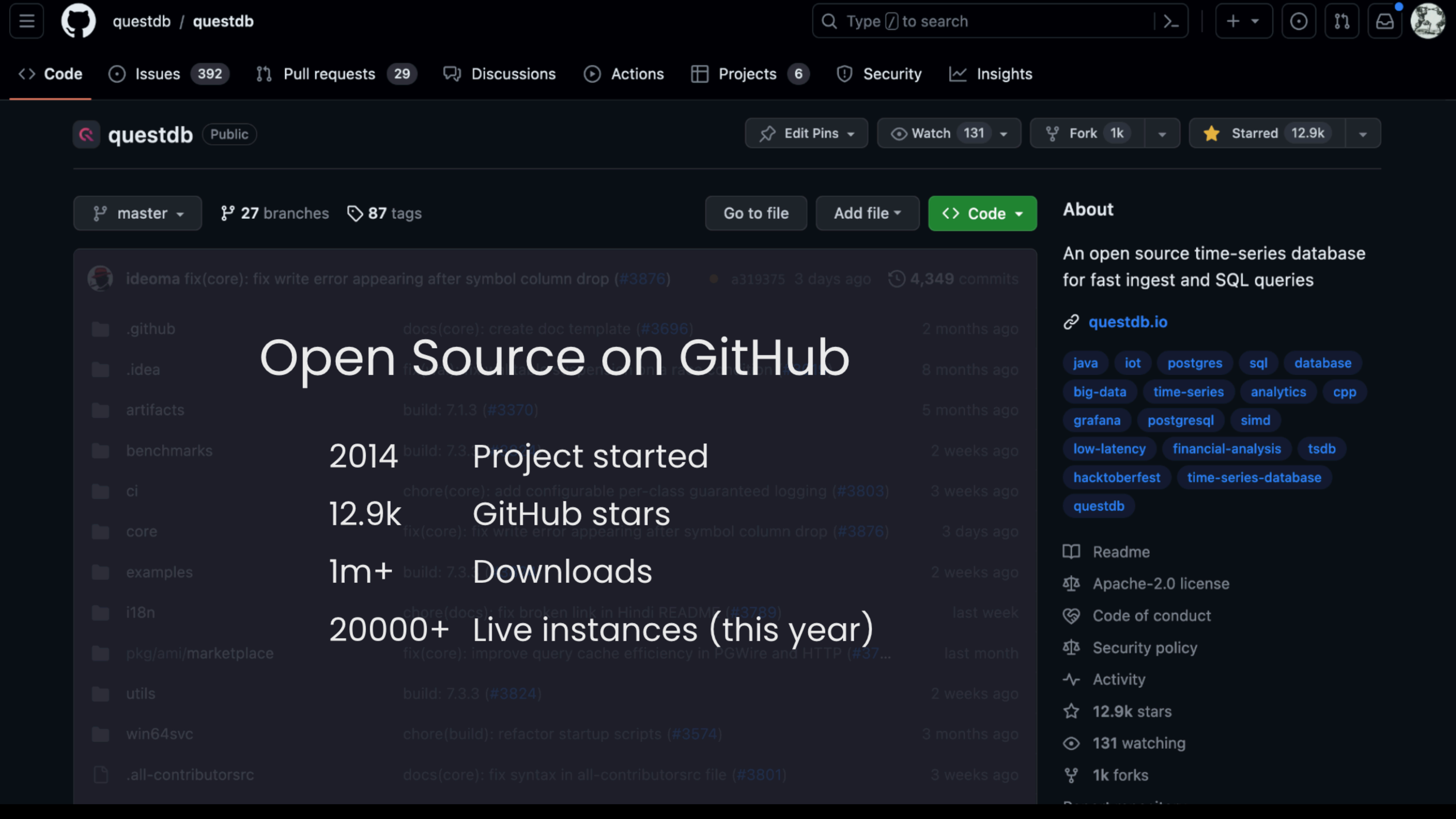
More difficult to calculate price

Can use cloud credit programs

Margin capture over SaaS



BYOC @ QuestDB


questdb / questdb
Type / to search
Code
Issues 392
Pull requests 29
Discussions
Actions
Projects 6
Security
Insights
questdb Public
Edit Pins
Watch 131
Fork 1k
Starred 12.9k
master
27 branches
87 tags
Go to file
Add file
Code
ideoma fix(core): fix write error appearing after symbol column drop (#3876)
a319375 3 days ago
4,349 commits
.github
.idea
artifacts
benchmarks
ci
core
examples
i18n
pkg/ami/marketplace
utils
win64svc
.all-contributorsrc
Open Source on GitHub
2014 Project started
12.9k GitHub stars
1m+ Downloads
20000+ Live instances (this year)
About
An open source time-series database for fast ingest and SQL queries
questdb.io
java
iot
postgres
sql
database
big-data
time-series
analytics
cpp
grafana
postgresql
simd
low-latency
financial-analysis
tsdb
hacktoberfest
time-series-database
questdb
Readme
Apache-2.0 license
Code of conduct
Security policy
Activity
12.9k stars
131 watching
1k forks

My 5 Tenants of BYOC

BYOC @ QuestDB

5 Tenants of BYOC

🔐 Security

💸 Cost Controls

👌 Simplicity

💻 Iterative Development

☁️ Multi-Cloud Support

My 5 Tenants of BYOC

Security 🔐

Account isolation

Private networking by default

Cloud IAM/RBAC

- Assume roles across tenants
- Principle of least-privilege

Cloud native bastion services

My 5 Tenants of BYOC

Cost Controls 💸

Minimize managed service usage…

But still use managed services when required

Contain networking fees

- Service Endpoints
- Vnet Peering

Start small, scale later

My 5 Tenants of BYOC

Simplicity 👌

Extreme Automation

Empower Support Engineers

- Minimize click-to-insights
- Enable safe service updates
- Developers as SREs

Minimal Architecture

Single source of truth

My 5 Tenants of BYOC

Iterative Development 💻

SSH as an escape hatch

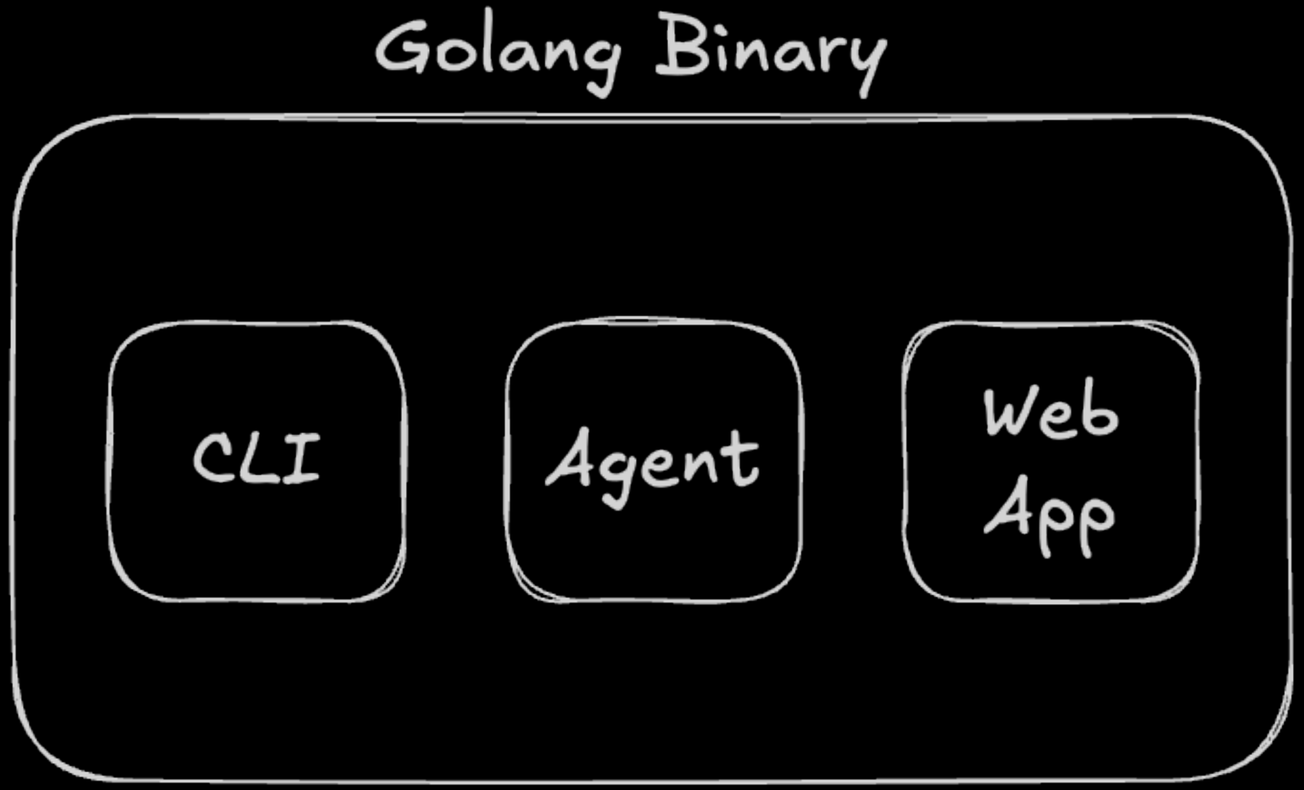
Component-based architecture

Start with familiar tools, but don’t be afraid to swap

Write everything in Go

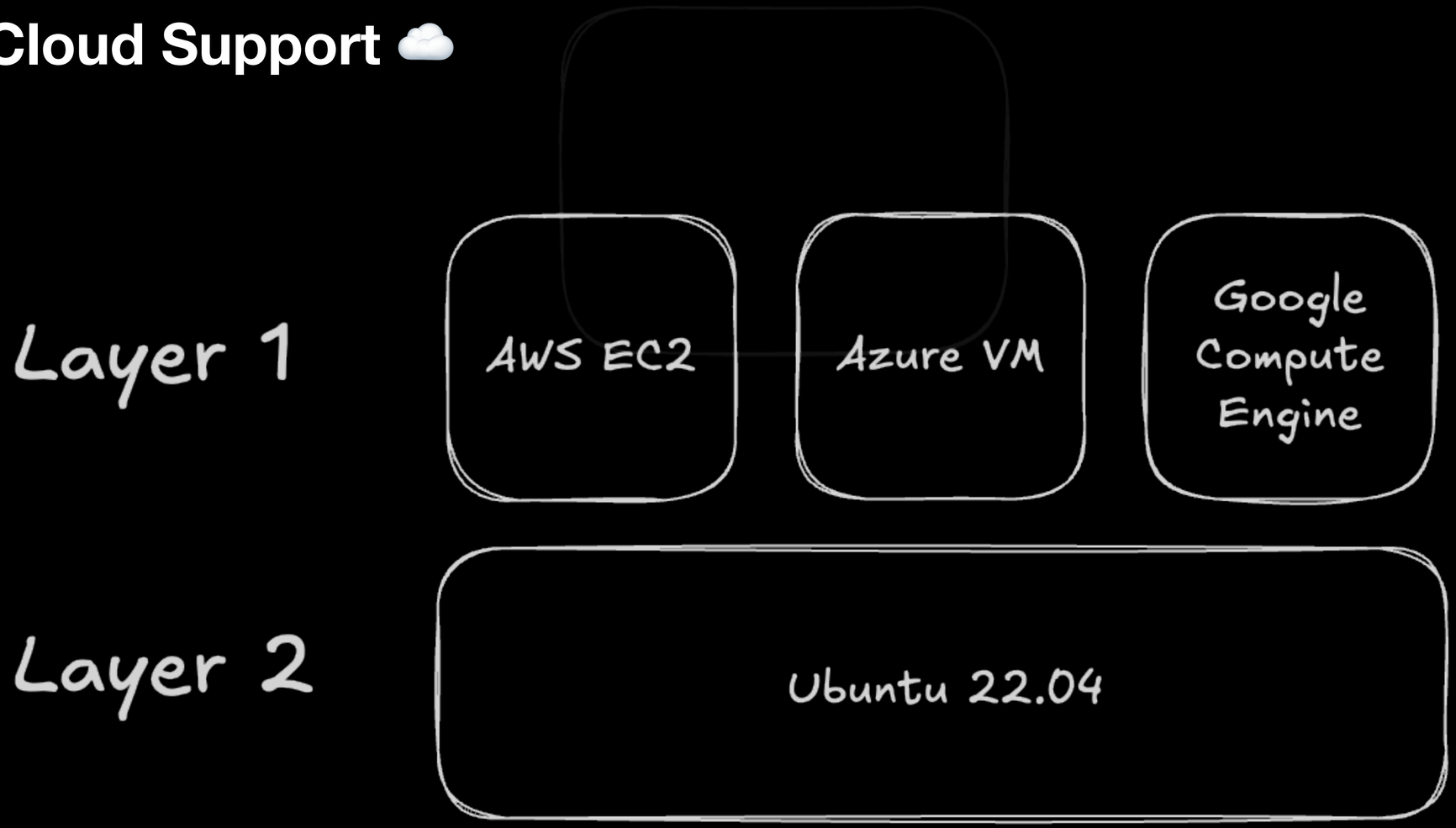
My 5 Tenants of BYOC

Iterative Development 💻



My 5 Tenants of BYOC

Multi-Cloud Support ☁️



BYOC Deep Dive

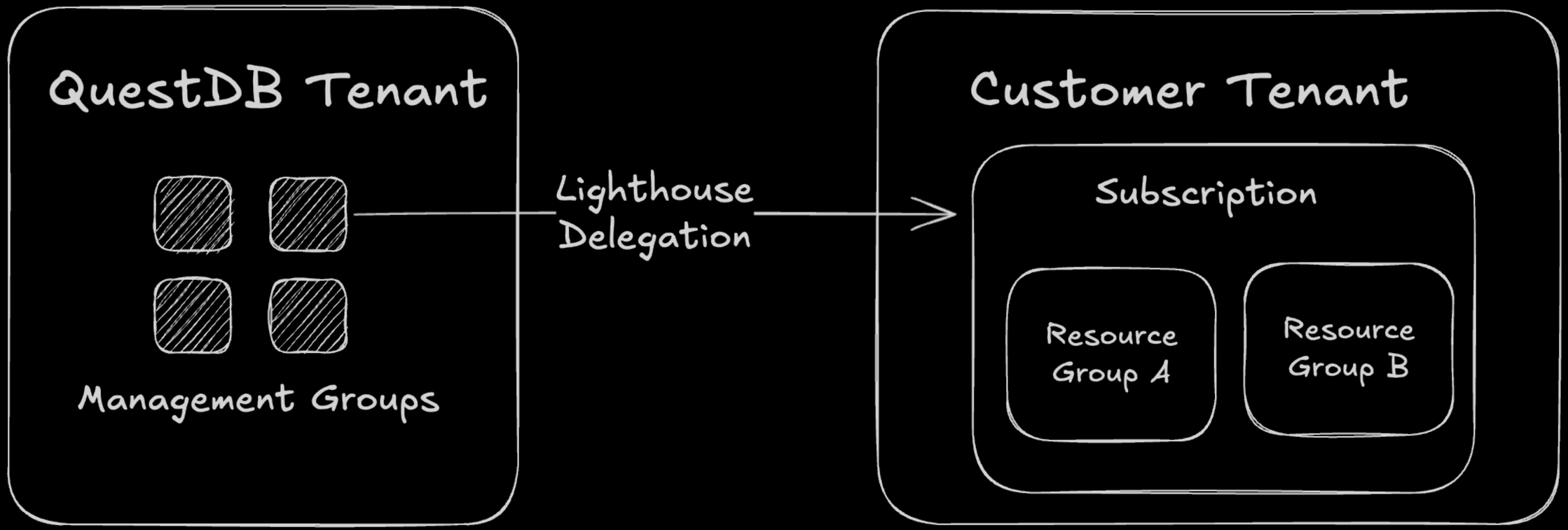
BYOC @ QuestDB

Configuration

```yaml
deployments:
  - name: byoc-test-az-2
    pulumiConfig:
      azure:
        tenantId: 12345678-aaaa-bbbb-cccc-12345678
        subscriptionId: 12345678-aaaa-bbbb-cccc-12345678
        location: eastus
    net:
      enablePublicInternetAccess: true
      azure:
        enableBastion: true
      ingress:
        http:
          port: 443
    clusters:
    - name: dev
      baseImageId: /subscriptions/12345678-aaaa-bbbb-cccc-12345678/resourceG
      diskSizeInGb: 16
      instanceType: Standard_F2s_v2
      replicaCount: 1
      questdb:
        config:
          shared.worker.count: "1"
          cairo.snapshot.instance.id: "dev"
        image: private-questdb-repo.questdb.com/questdb:2.1.2-enterprise
      snapshots:
        systemdCalendarSchedule: "*-*-* 01:00:00 UTC"
        snapshotsToRetain: 5
```

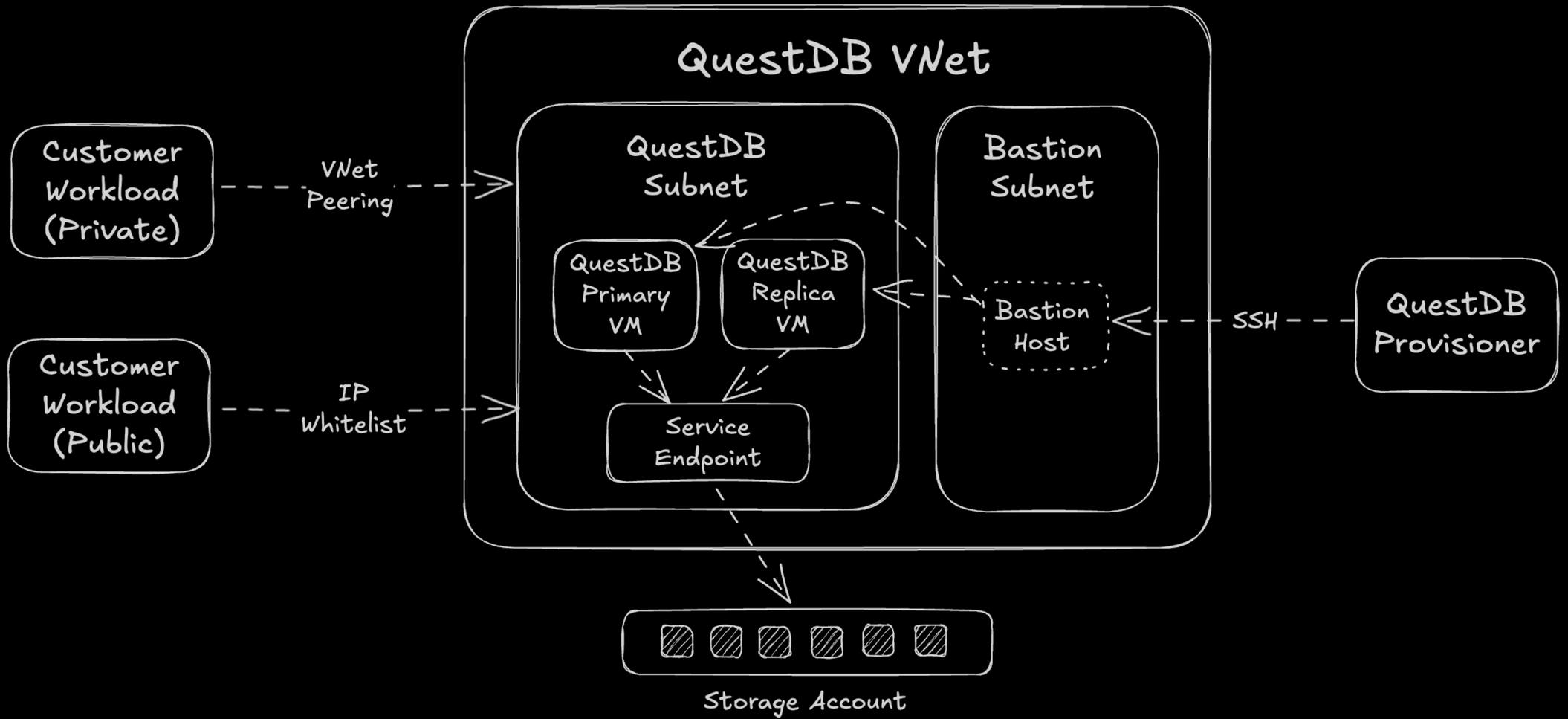
BYOC @ QuestDB

IAM & RBAC



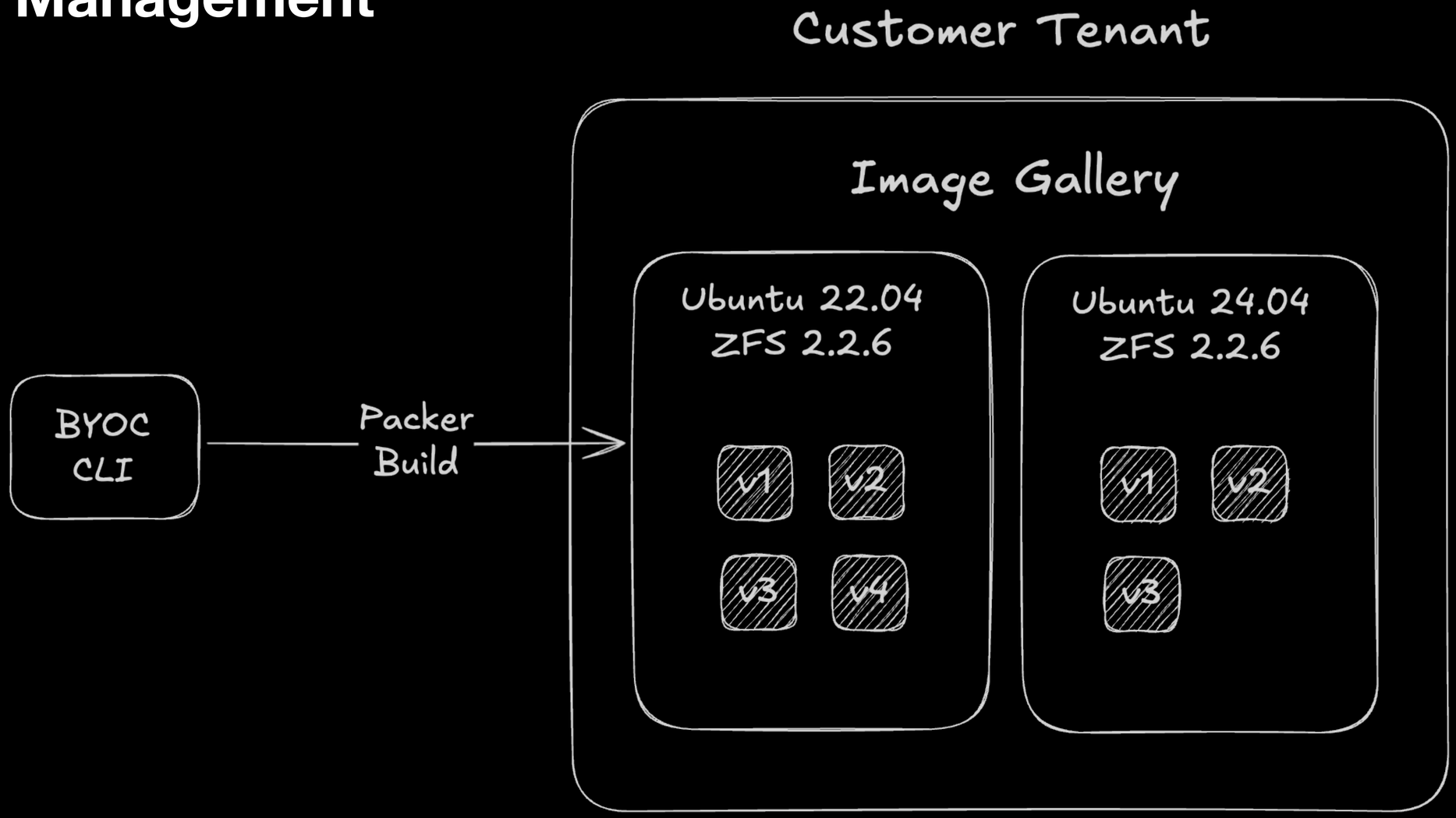
BYOC @ QuestDB

Networking



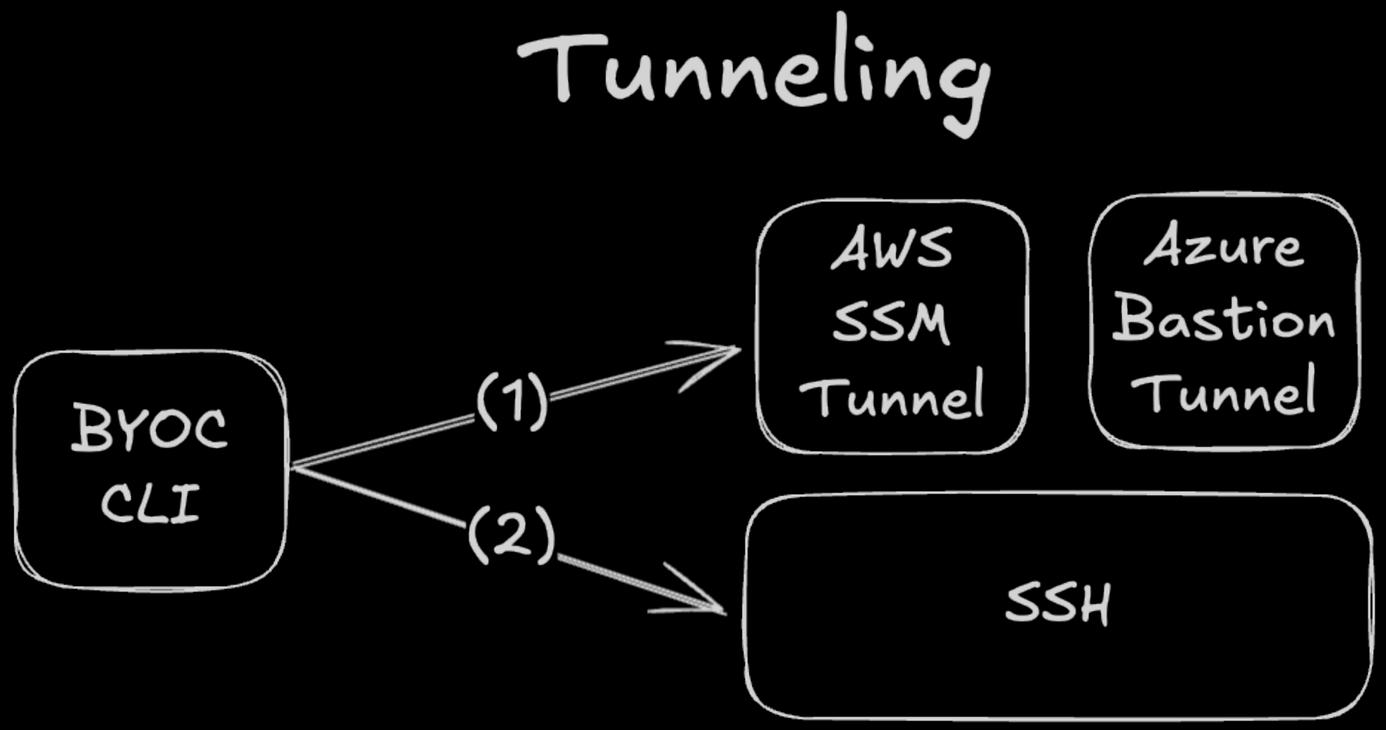
BYOC @ QuestDB

Image Management



BYOC @ QuestDB

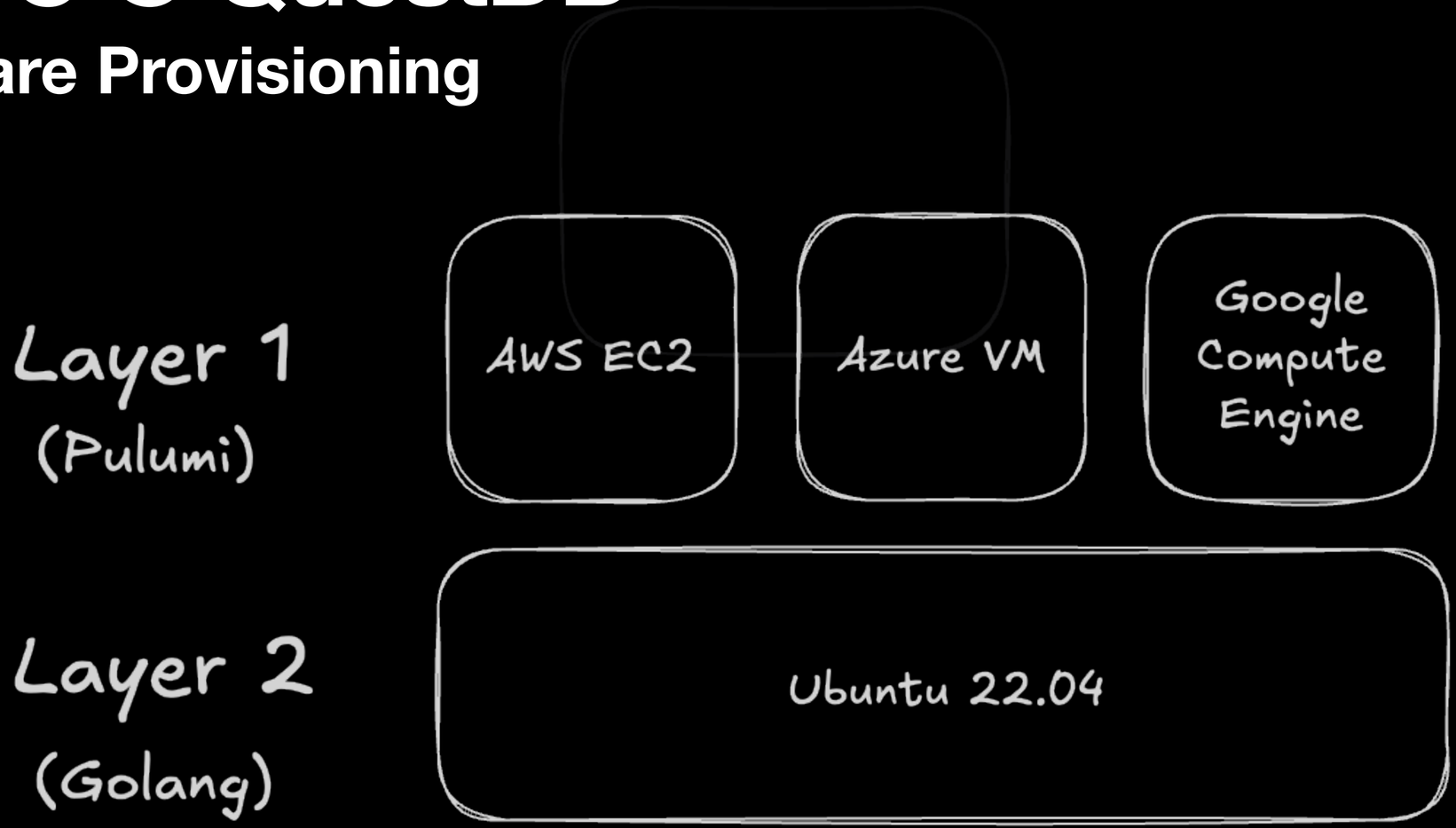
Tunneling



1. Open cloud-specific tunnel to port 22
2. Connect over SSH (key stored securely)

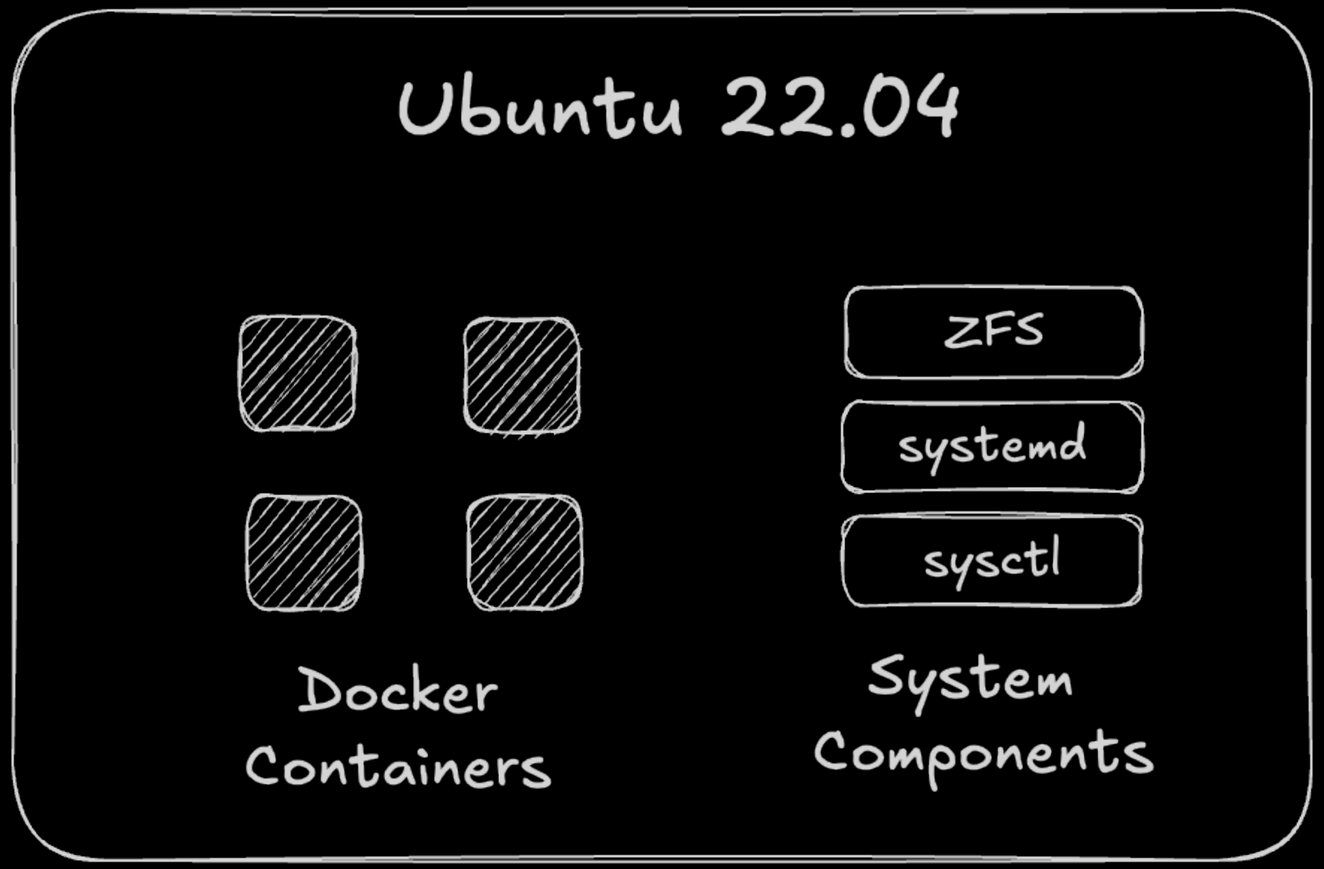
BYOC @ QuestDB

Software Provisioning



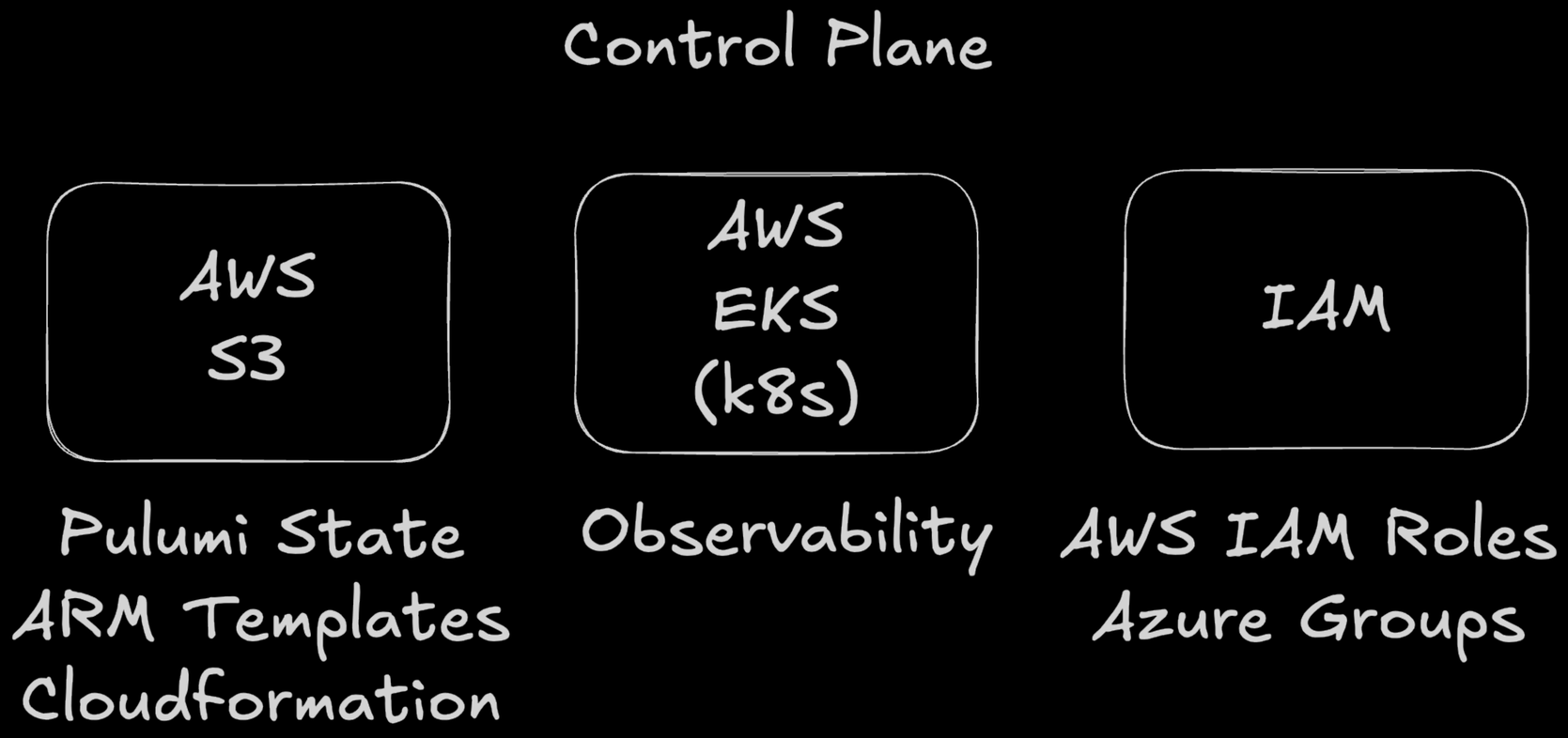
BYOC @ QuestDB

Software Provisioning



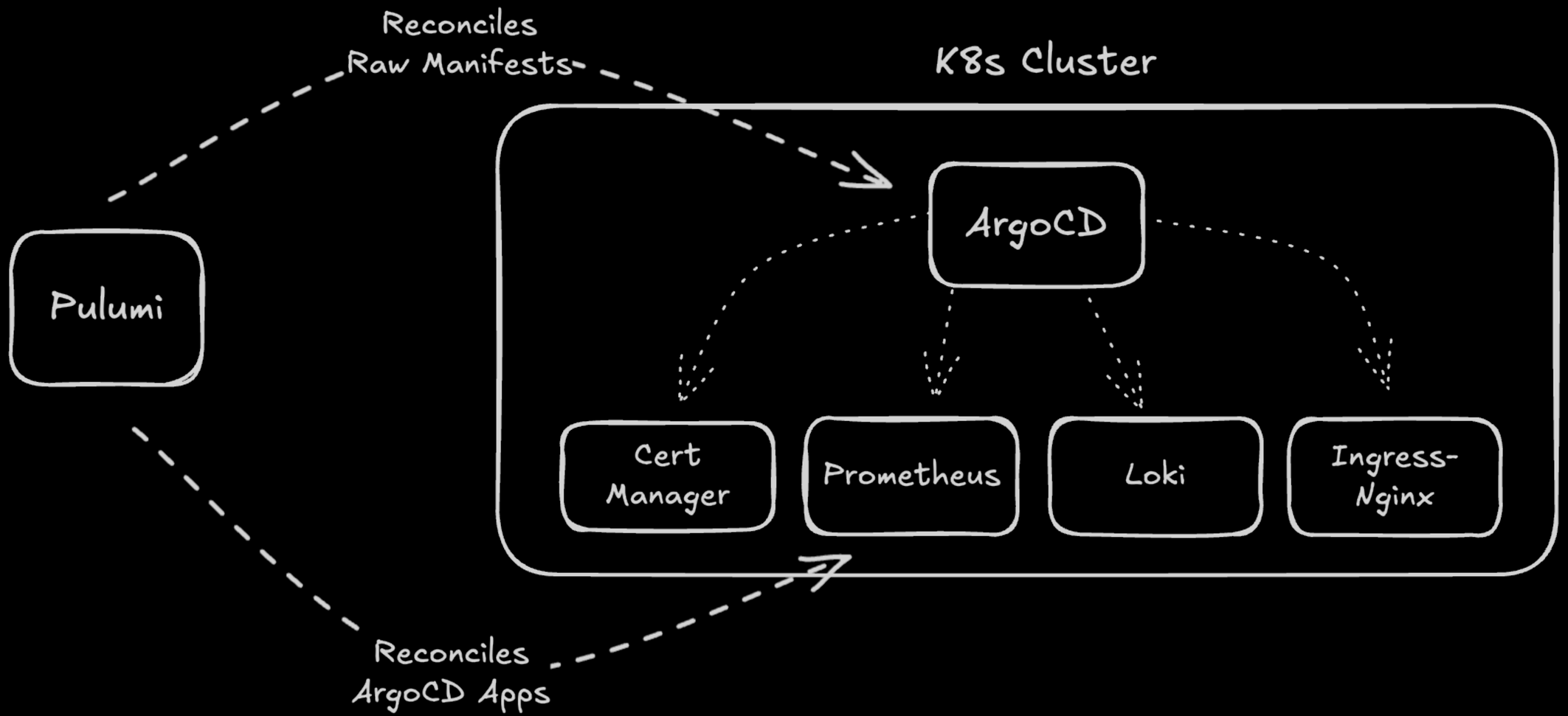
BYOC @ QuestDB

Control Plane



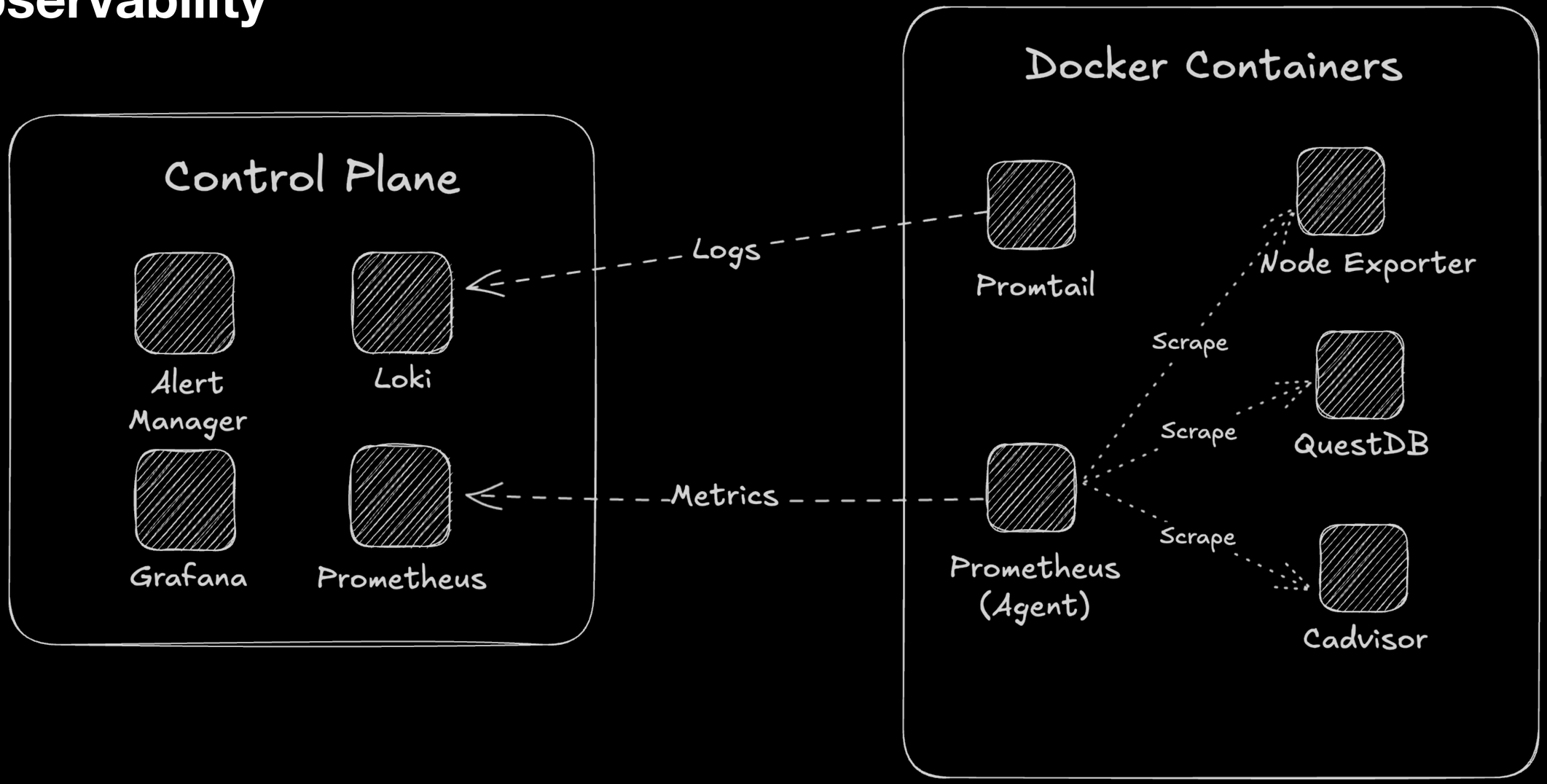
BYOC @ QuestDB

Control Plane: K8s with Pulumi



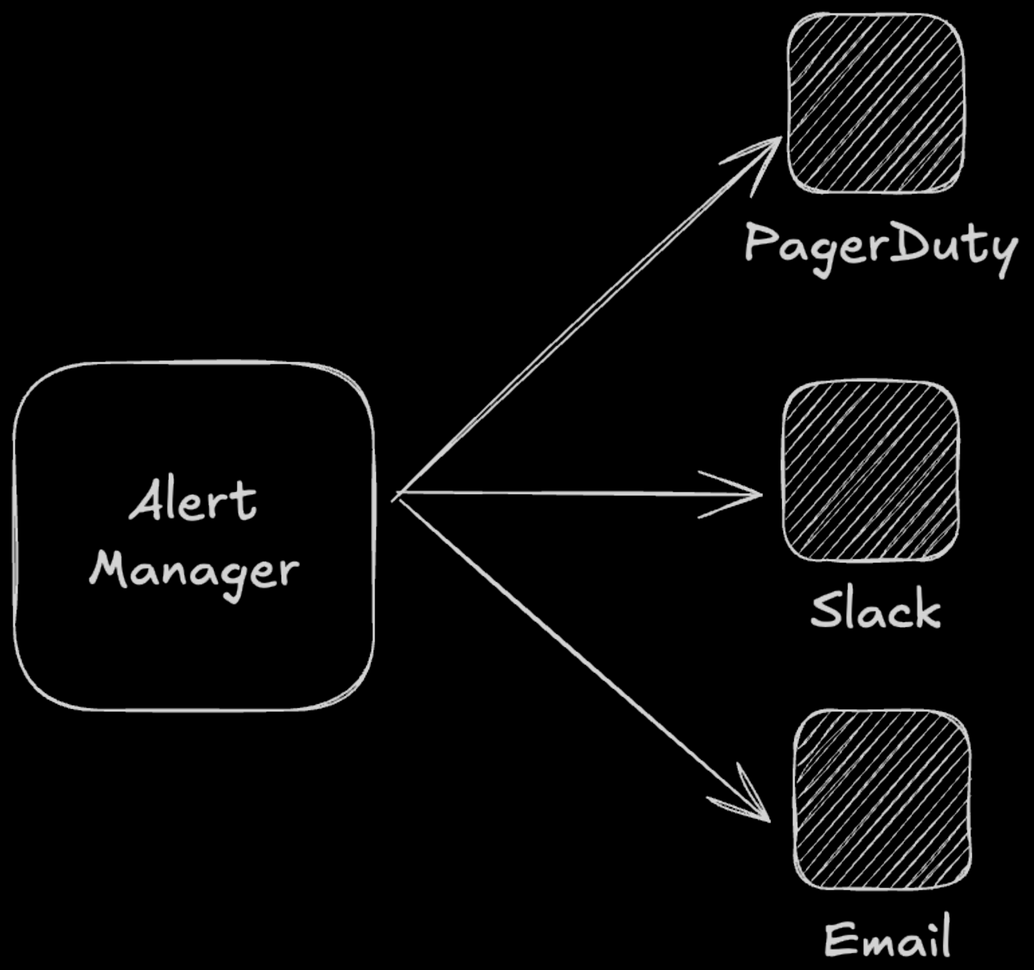
BYOC @ QuestDB

Observability



BYOC @ QuestDB

Observability



BYOC @ QuestDB

Challenges

Control Plane vs Deployment dependencies

Image Management — ZFS

“Advanced” IAM Features (Azure Lighthouse)

Internal distribution strategy

Customer data migrations (replication helps!)

No k8s

Questions?

www.questdb.com

demo.questdb.io