Database DevOps: CD for Stateful Applications

Stephen Atwell and Chris Crow

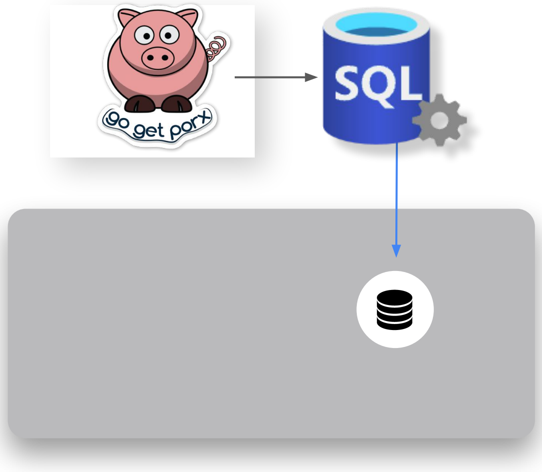
Data Integrity Tests in CI/CD Pipelines

Overview

Data aware CI/CD can mitigate downtime risk. By leveraging a single CI/CD pipelines for all changes (application, Database infra version, and Database schema version) you can ensure all changes work reliable before they reach production. By leveraging production data in pre production environments you can identify any data-specific edge cases to echanges before they affect production.

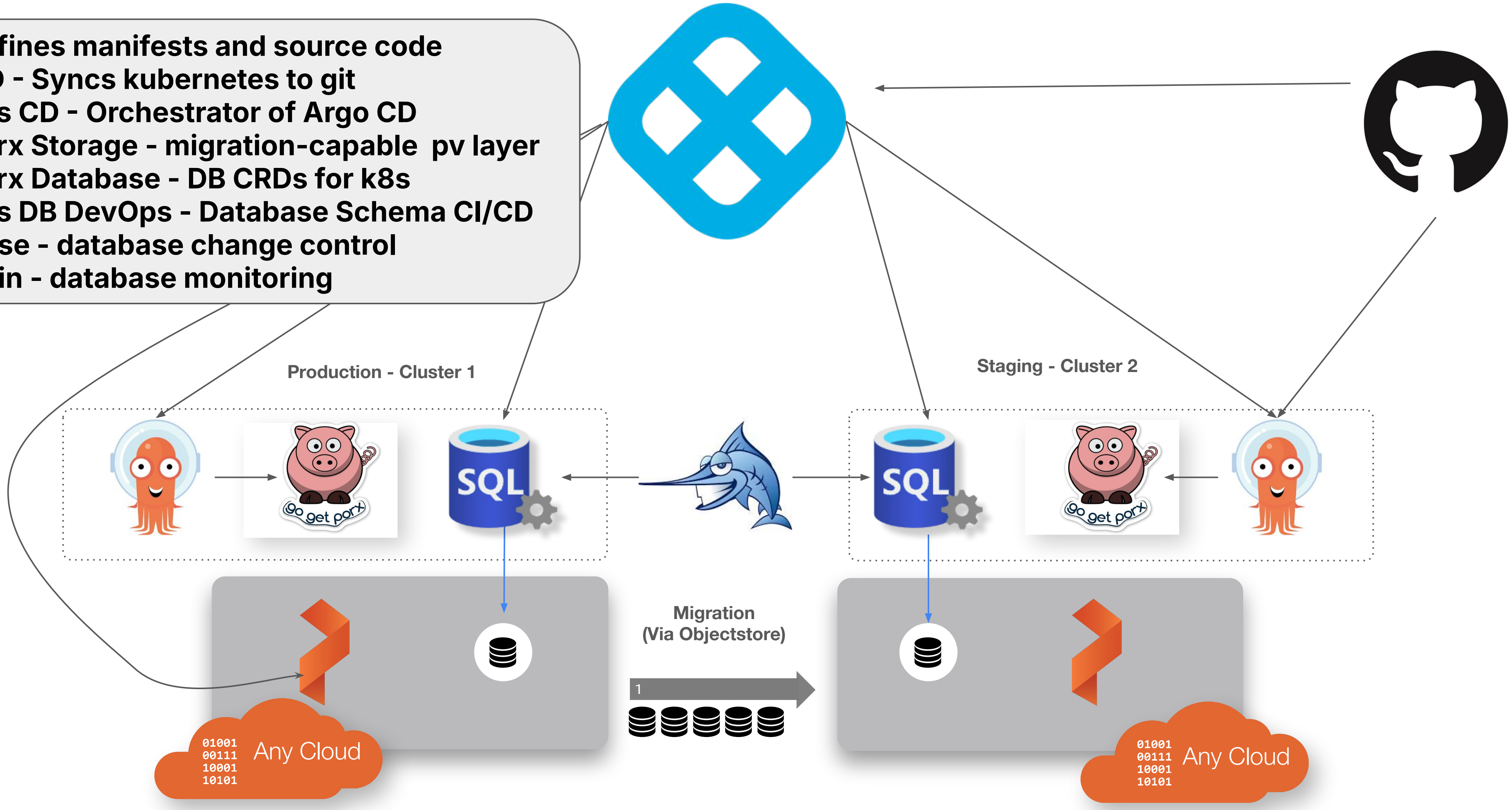
Our Application - A Nice Simple Architecture

- javascript front end, connecting to go backend
- BBQ Restaurant inventory tracking
  - inventory stored in Database
    - database stores in Kubernetes Persistent Volume
- Entire app (frontend, backend, DB) running on Kubernetes



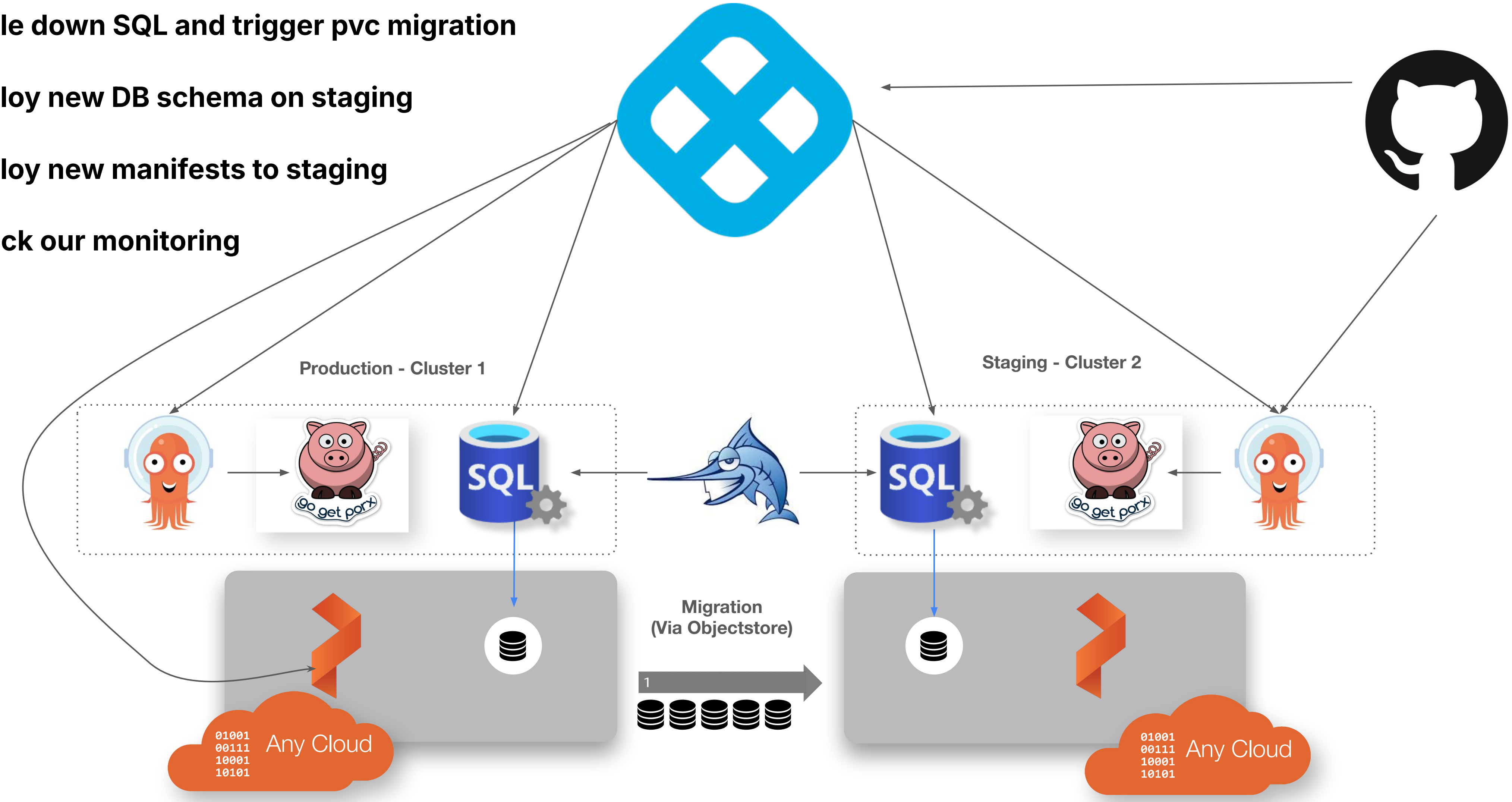
Our Application - Operational Ecosystem

1. **Git - defines manifests and source code**
2. **ArgoCD - Syncs kubernetes to git**
3. **Harness CD - Orchestrator of Argo CD**
4. **Portworx Storage - migration-capable pv layer**
5. **Portworx Database - DB CRDs for k8s**
6. **Harness DB DevOps - Database Schema CI/CD**
7. **Liquibase - database change control**
8. **DBMarlin - database monitoring**



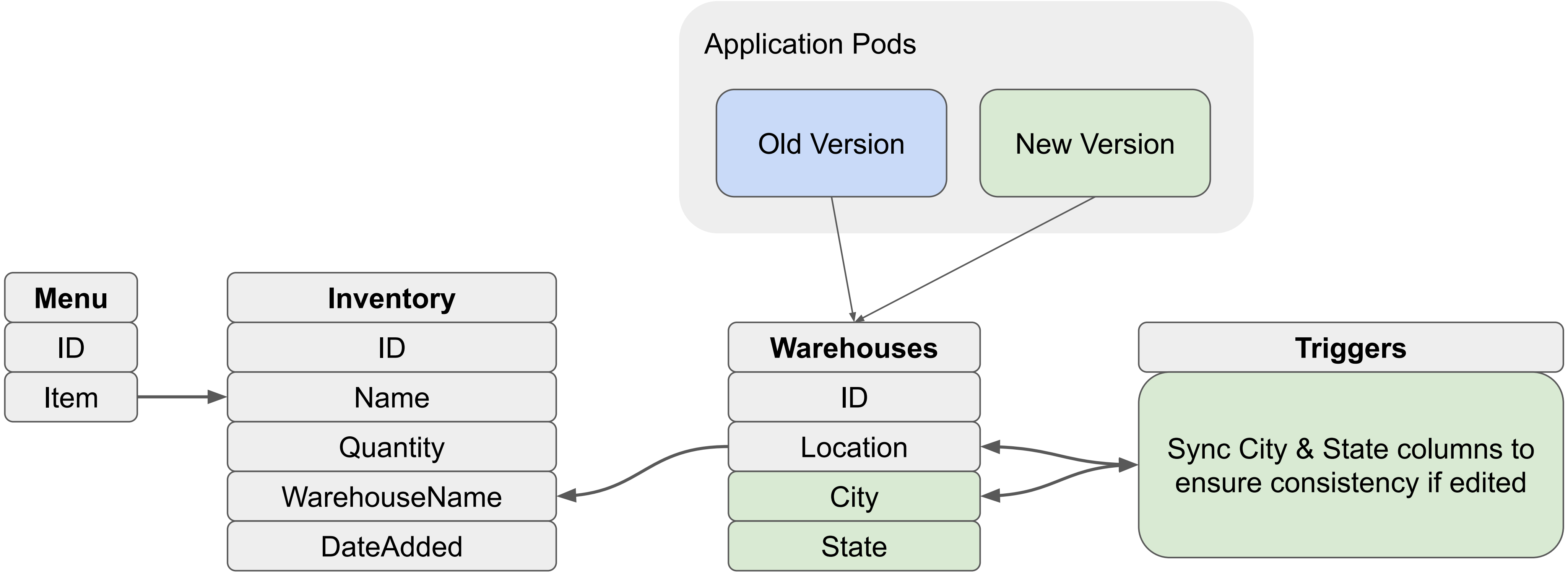
Overview - Testing with Data in CD Pipelines

1. **Scale down SQL and trigger pvc migration**
2. **Deploy new DB schema on staging**
3. **Deploy new manifests to staging**
4. **Check our monitoring**



Today’s Change? A database Schema update

1. **3 tables: manu, inventory, warehouses**
2. **Goal: Update Warehouses to have a ‘city’ and ‘state’ column**
3. **Take no downtime, ensure no data loss, maintain data integrity**

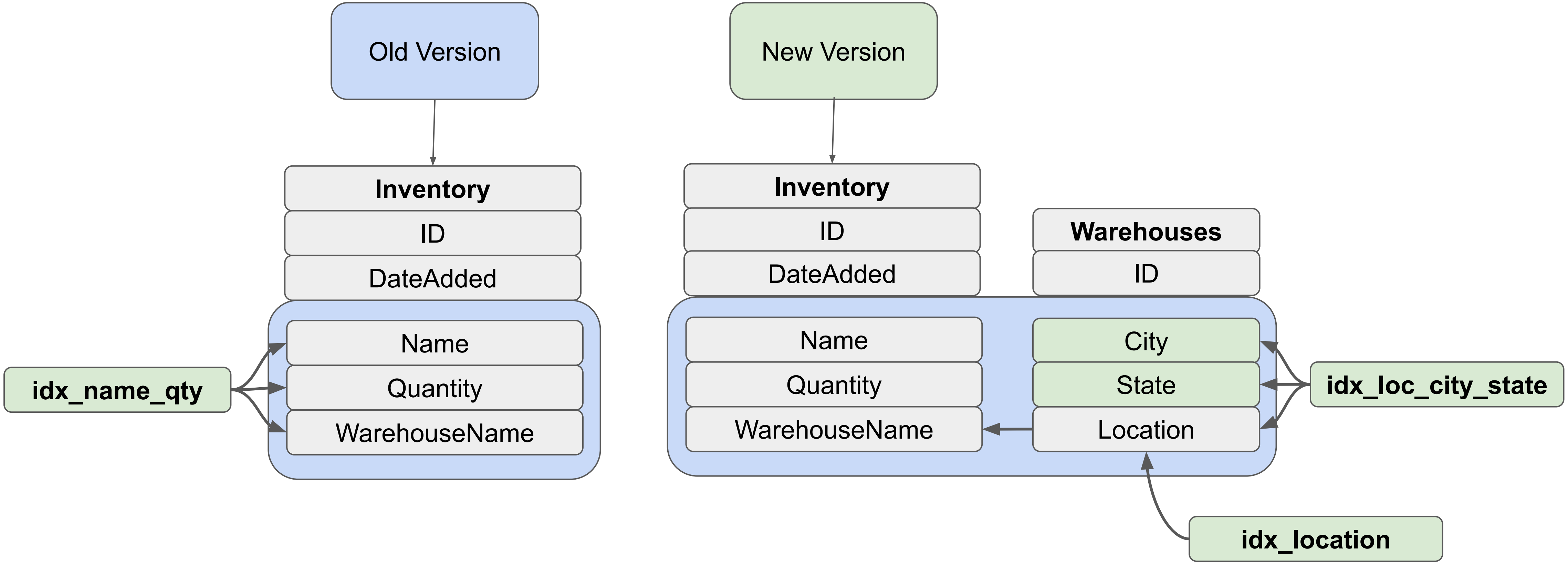


Demo Time!

and hopefully we’ll remember to edit data first this time....

Why We have a Performance Problem?

1. **Need indexes to make us able to efficiently join the 2 tables**



Questions?

Appendix

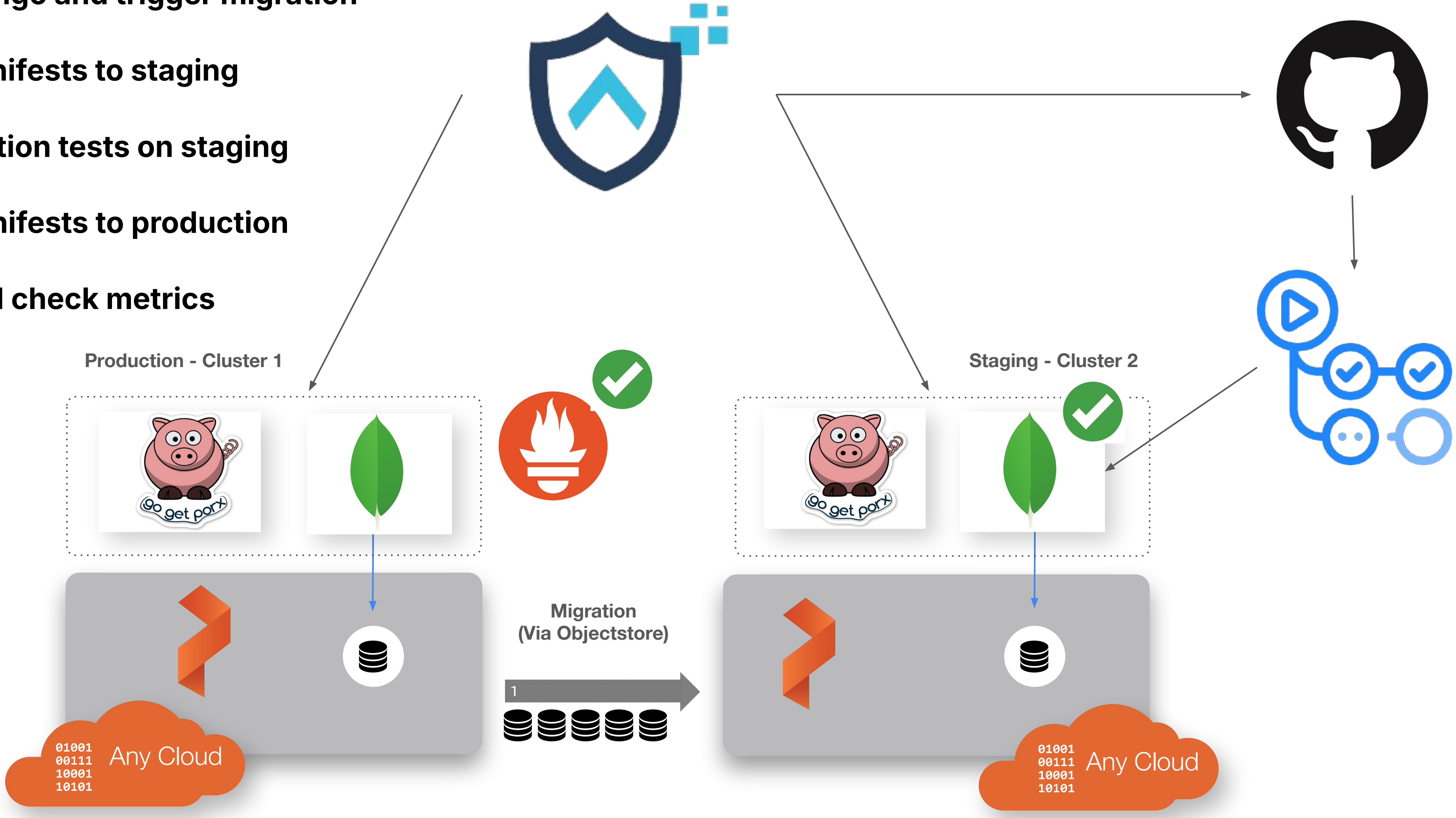
Data Integrity Tests in CI/CD Pipelines

Overview

Data integrity tests can pose a significant challenge to organizations. By managing data as part of a CI/CD pipeline, we can ensure that any application change also includes tests a copy of production data.

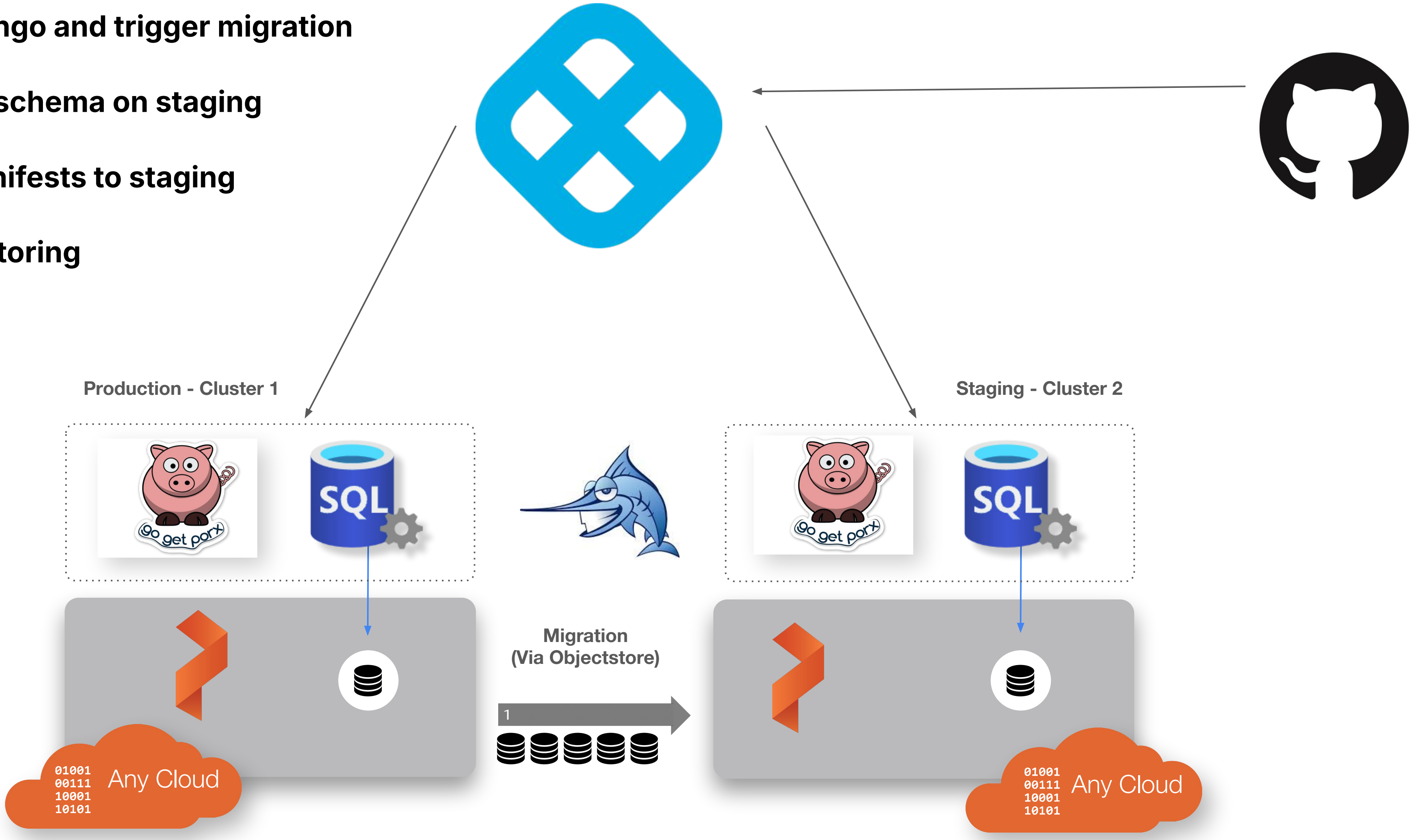
Overview - Testing with Data in CD Pipelines

1. **Scale down Mongo and trigger migration**
2. **Deploy new manifests to staging**
3. **Perform integration tests on staging**
4. **Deploy new manifests to production**
5. **Scale traffic and check metrics**



Overview - Testing with Data in CD Pipelines

1. **Scale down Mongo and trigger migration**
2. **Deploy new DB schema on staging**
3. **Deploy new manifests to staging**
4. **Check our monitoring**


Production - Cluster 1
go get porx
SQL
01001
00111
10001
10101
Any Cloud
Migration
(Via Objectstore)
1
Staging - Cluster 2
go get porx
SQL
01001
00111
10001
10101
Any Cloud