
CODCON
May 18–20, 2026 | Minneapolis, Minnesota

GitOps Gone Wild: Hardening Delivery Pipelines for the AI era

Julien Semaan, Kubex
Corey Mcgalliard, Akamai Cloud

May 18–20, 2026 | Minneapolis, Minnesota

AI Agents B0rking things

Mashable Follow

An AI agent allegedly deleted a startup's production database, causing a huge outage

Alex Perry

Updated Mon, April 27, 2026 at 2:58 PM EDT

Entertainment > Streaming

AWS suffered ‘at least two outages’ caused by AI tools, and now I’m convinced we’re living inside a ‘Silicon Valley’ episode

Opinion By Jason England published February 20, 2026

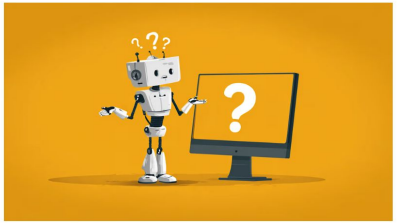
Amazon just speed-ran a season of 'Silicon Valley'

HBO / Getty Images)

Intent-based chaos testing is designed for when AI behaves confidently — and wrongly

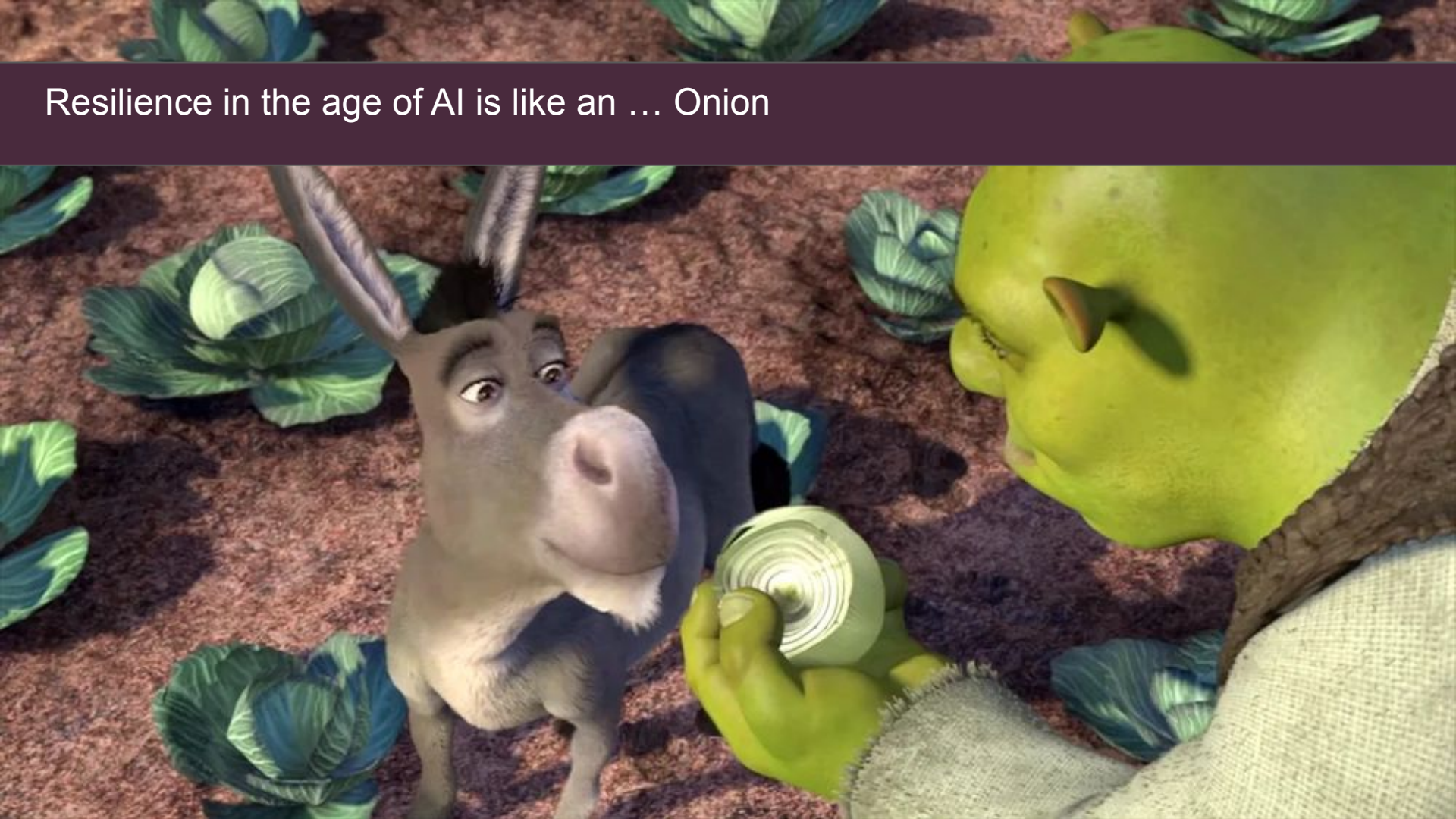
Sayali Patil

9:00 am, PT, May 9, 2026



Cleo® made with Midjourney

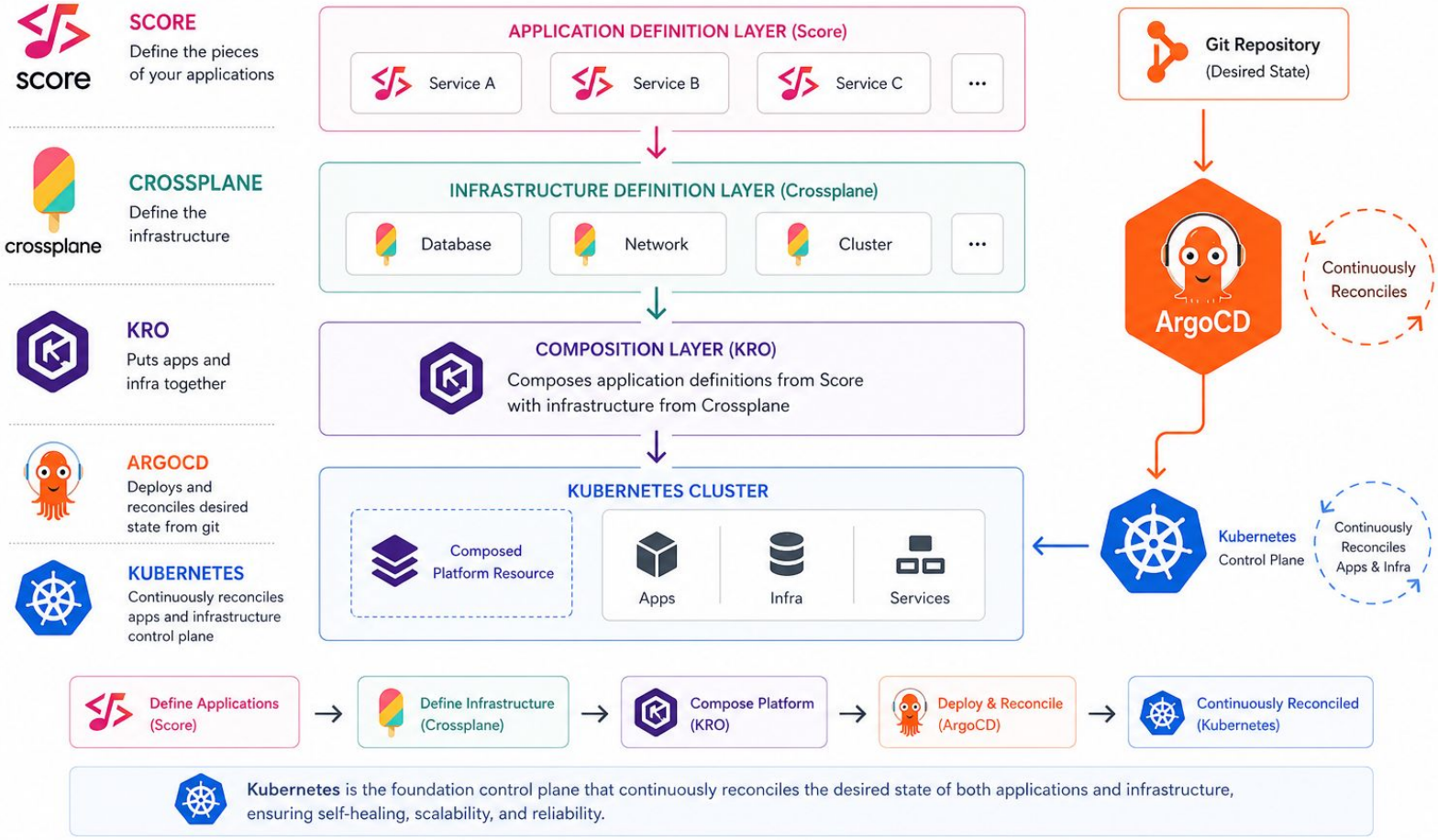
Resilience in the age of AI is like an … Onion



Avoid AI putting a decimal point in the wrong place

PLATFORM OVERVIEW

From application definition to continuously reconciled platform in your Kubernetes cluster



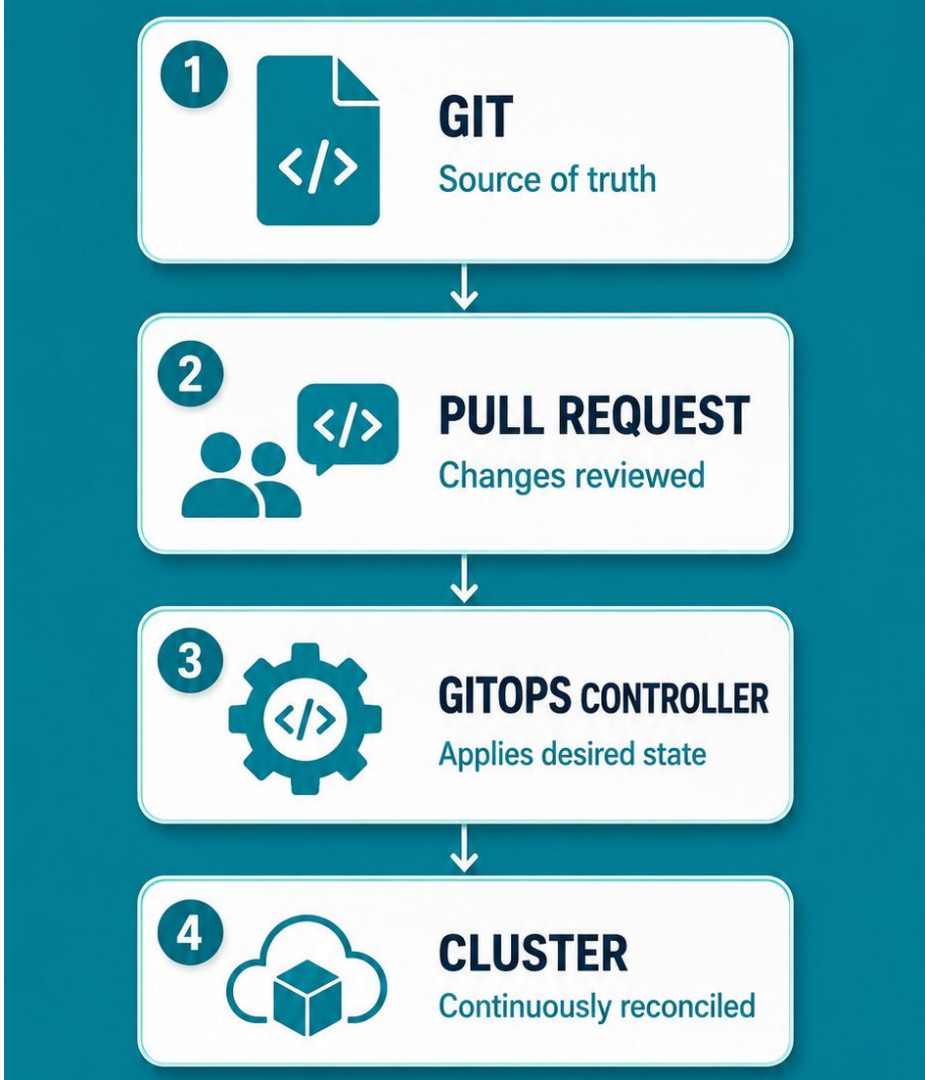
Application Development and Integration Layer

Gitops, what's that?

- Manage applications and infrastructure manifests (YAML) in git
- Gets applied automatically on the system when merging pull-requests
- Inherits ways of working with code for managing operations

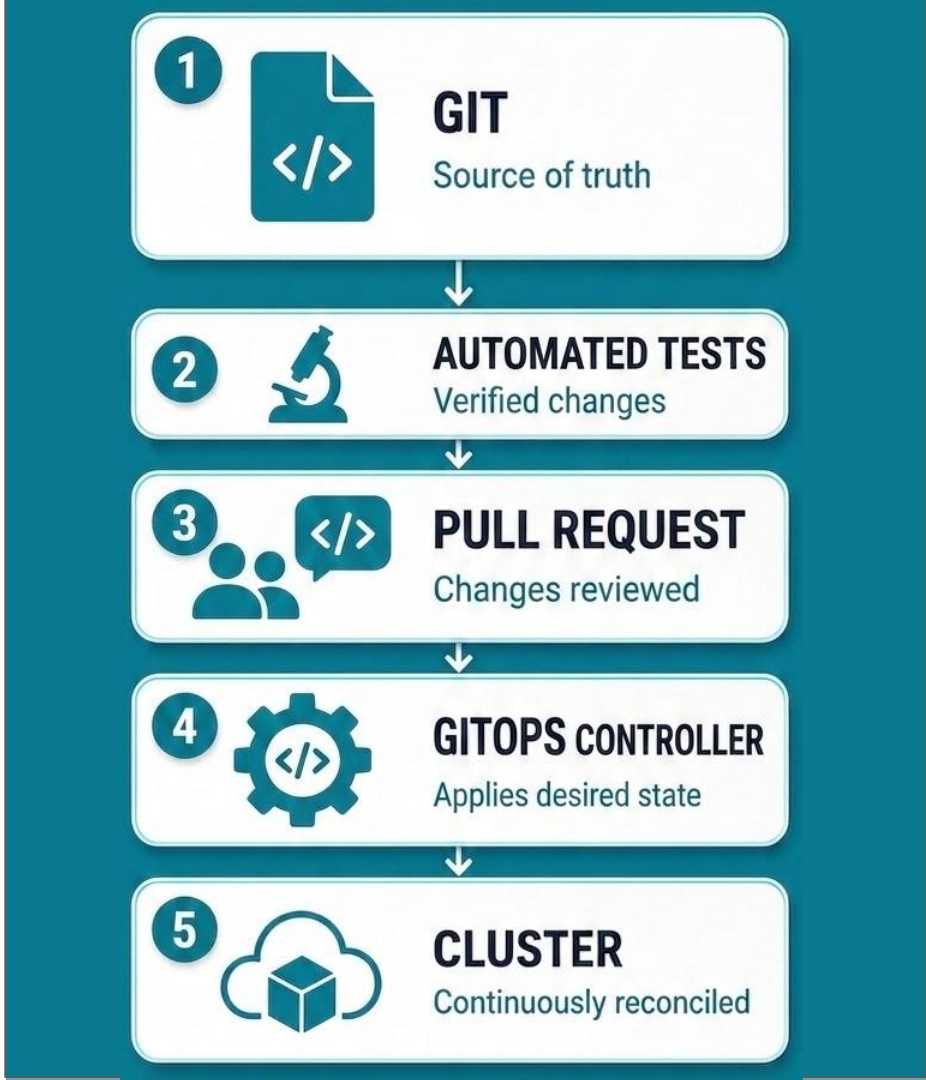
But it's not all good….

- Easy to spread mistakes at scale
- Failures are more often "the wrong thing happened everywhere" instead of "nothing happened"
- Control loops make it difficult to make out-of-band adjustments



Test Driven Software Process

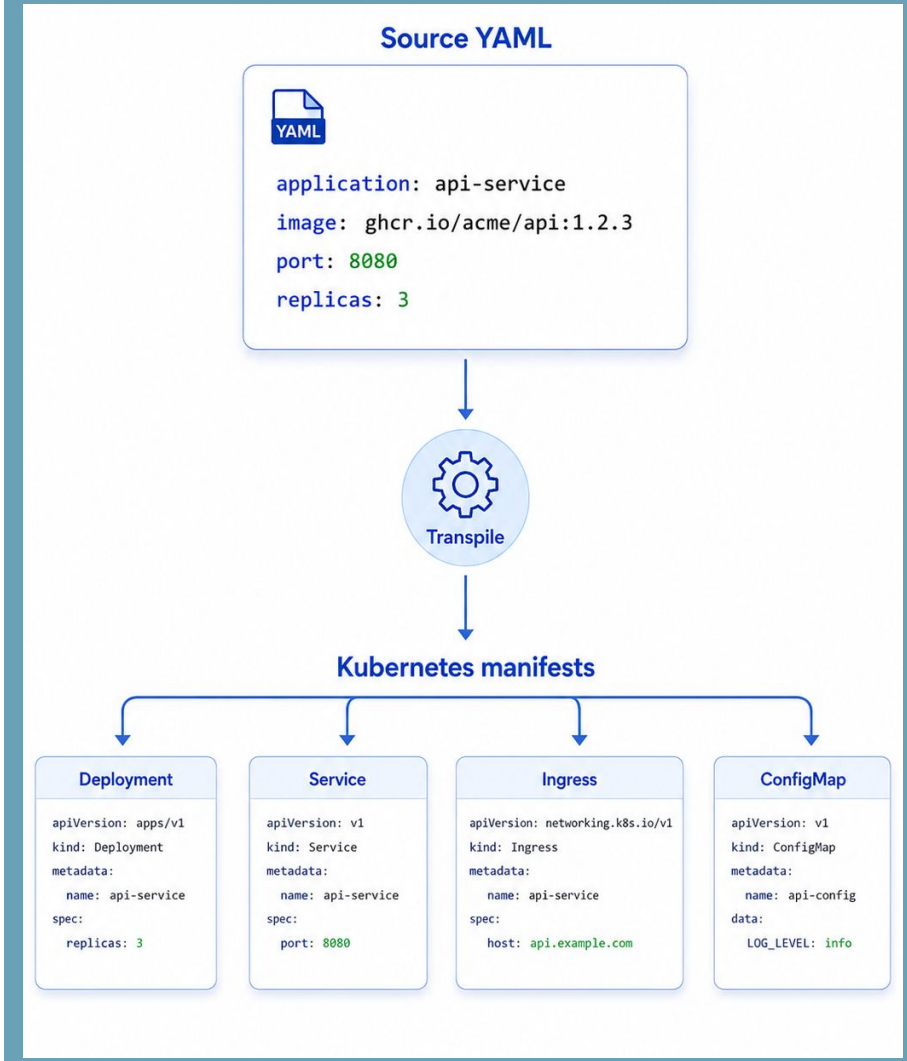
- **Test-Driven Development (TDD):** provides a structured framework that acts as a quality control layer for AI-driven development
- **End to End Tests:** Provides the ability to validate changes meet the requirements of the organization and validate the scope of the change.
- **Canary and Non-Production Environments:** Provides production like resources to validate changes before prompting them into production.



Deployment Layer

How yaml-to-yaml can help?

- Simplifies surface area to review for both humans and AI agents.
- Delegate complexity to the transpiler
  - It is its own product (in-house or 3rd party)
  - Quality controlled
  - Deterministic
- Reviews can focus on the intent of the change instead of all the underlying overly detailed YAML
- Prevents AI agents (and humans) from "improvising" too much in the manifests.
  - When it's not part of the simplified spec, consider it as a new feature for the transpiler



Platform & Infrastructure yaml-to-yaml using KRO

- Infrastructure configuration is flexible by design and that's our problem
  - Node type
  - Region
  - Node count
  - Kubernetes version
  - Network configuration
  - Firewall rules
  - Load balancer settings
  - Storage class
  - Disk size and type
  - Availability zone strategy
  - Backup configuration
  - **And the list goes on and on and on**
- Define the internal standards to use and expose a simple interface with just the right knobs to configure for your specific needs

https://github.com/kubernetes-sigs/kro

```yaml
apiVersion: api.platform.example.com/v1
kind: PlatformCluster
metadata:
  name: demo-cluster
  namespace: default
  labels:
    environment: dev
    team: platform-team
    purpose: development
spec:
  protected: true
  clusterSpec:
    clusterName: demo-cluster
    region: us-east
    kubernetesVersion: "1.35"
    nodePool:
      count: 3
      type: g6-standard-4
    labels:
      environment: dev
      team: platform-team
  components:
    gatewayFabric:
      enabled: true
    userApps:
      enabled:
        - demo-app
```

App yaml-to-yaml using Score

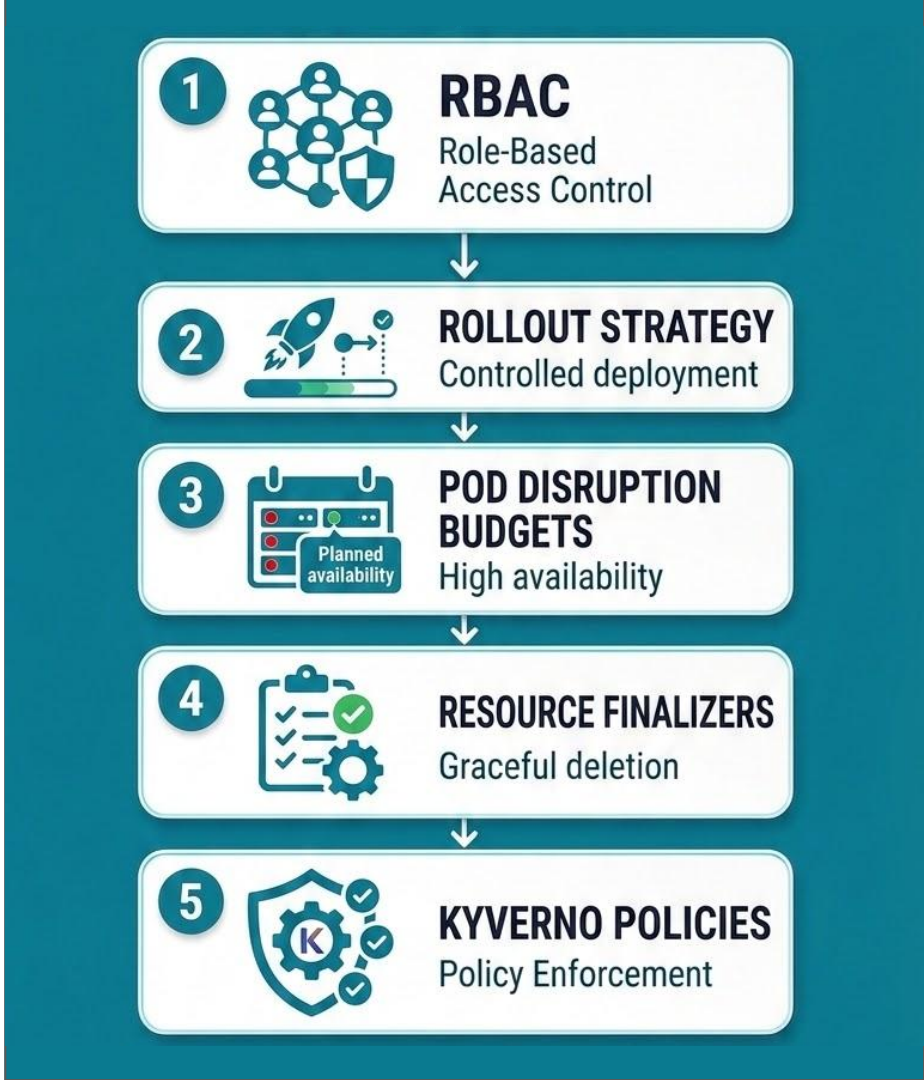
- Lightens up the declaration of apps and their dependencies
- Services are glued together via the transpiler and rendering layer.
- Complexity lives in Score, it knows how to stitch components together, users only expresses a simple intent

https://score.dev/

```yaml
apiVersion: score.dev/v1b1
metadata:
  name: hello-world
  annotations:
    tags: "nodejs,http,website,javascript,postgres"
containers:
  hello-world:
    image: scorespec/sample-score-app:latest
    variables:
      PORT: "3000"
      MESSAGE: "Hello, World!"
      DB_DATABASE: ${resources.db.name}
      DB_USER: ${resources.db.username}
      DB_PASSWORD: ${resources.db.password}
      DB_HOST: ${resources.db.host}
      DB_PORT: ${resources.db.port}
resources:
  db:
    type: postgres
  dns:
    type: dns
  route:
    type: route
    params:
      host: demo.oss.baby
      path: /
      port: 8080
service:
  ports:
    www:
      port: 8080
      targetPort: 3000
```

Kubernetes

- **RBAC:** Role Based Access Control allows us to enforce that users, and agents have appropriately scoped access to resources.
- **Rollout Strategies:** these ensure updates to the applications running in the cluster honor policies around updating their configuration and software
- **Pod Disruption Budgets:** Enforce that software is running and pass health checks before adjusting scheduling and de-scheduling of resources.
- **Resource Finalizers:** Enable us to ensure that appropriate steps are taken before deleting resources in our cluster
- **(Kyverno) Policies:** Guarantees that changes meet the requirements of the system before passed to internal system controllers.



Infrastructure Layer

Cloud Provider Resource Protection

OSS-ers Assemble

Platform level protection

- **PlatformCluster** spec contains a `**protected**` flag
- When enabled:
  - Prevents Infrastructure Resource Deletion
  - Prevents modifying the region
  - Configures Cloud Provider Deletion Protection
  - Pauses Crossplane on critical resources
  - Prevents Removing an application
  - Custom Finalizers on all KRO resources
  - Enforces Kyverno Policies to stop mutations/deletions of critical objects

```yaml
apiVersion: api.platform.example.com/v1
kind: PlatformCluster
metadata:
  name: demo-cluster
  namespace: default
  labels:
    environment: dev
    team: platform-team
    purpose: development
spec:
  protected: true
  clusterName: demo-cluster
  region: us-east
  kubernetesVersion: "1.35"
  nodePool:
    count: 3
    type: g6-standard-4
  labels:
    environment: dev
    team: platform-team
components:
  gatewayFabric:
    enabled: true
  userApps:
    enabled:
      - demo-app
```

Join Us for Our Next Talk!

Next Day

From Apps To Infrastructure: A Cloud Native First Approach

- Julien Semaan, Kubex & Corey McGalliard, Akamai

Thursday May 21, 2026
11:55am – 12:35pm CDT

200F (Level Two)

Hope to see you there!