

Advanced Kubernetes Workshop

Fairwinds





Wifi: GoogleGuest



Chrome Browser



Incognito Mode

explore.qwiklabs.com

Agenda

What you build

Lab 0: Workshop Setup

Kubernetes Primer

Lab 1: Build with Skaffold

Lab 2: Deploy with Cloud Build and Spinnaker

Lunch

Lab 3: Control and secure with Istio

Lab 4: Monitor and troubleshoot with Istio

01

Lab 0: Setup

In Progress



SCHEDULED COURSE

Advanced Kubernetes Workshop

GKE 1.10, Istio 1.0.1 and Spinnaker 1.8

Sep 7, 2018 12:00PM EDT
Sunnyvale



← Advanced Kubernetes Workshop



Home

Catalog

My Learning

Help

Labs

○ Advanced Kubernetes Workshop 091018

Overview

GKE 1.10, Istio 1.0.1 and Spinnaker 1.8

Date

Friday, September 7, 2018 9:00AM PDT –
Saturday, September 15, 2018 9:00AM PDT



2160m access · 1440m completion

[Rate Lab](#) [Lab Details](#)**Connection Details**[Open Google Console](#)

Username

.....

Advanced Kubernetes Workshop

Password

.....



GCP Project ID

.....

[Start Lab](#)**1 days**Google Cloud Platform Customer
Engineering Team[Overview](#)[Setup](#)[Lab 0: Setup \(30 mins\)](#)[END OF LAB 0 - Please STOP and wait
for instructions before proceeding!](#)[Lab 1: Build \(45 mins\)](#)[END OF LAB 1 - Please STOP and wait
for instructions before proceeding!](#)



2160m access · 1440m completion



[Rate Lab](#) [Lab Details](#)



Connection Details



[Open Google Console](#)



Username

googlece9955_student@qwiklabs.net



Password

KgJ4mr7RC6



GCP Project ID

qwiklabs-gcp-ff55a5c11c7b193e



Lab Running

[End Lab](#)

1 days

Advanced Kubernetes Workshop

Google Cloud Platform Customer
Engineering Team

Overview

Setup

Lab 0: Setup (30 mins)

END OF LAB 0 - Please STOP and wait
for instructions before proceeding!

Cloning the repo

Get all files associated with this workshop by cloning the following repo.

```
git clone https://github.com/FairwindsOps/advanced-kubernetes-workshop
```

Running install scripts

Run the following command to kick off the script to install all resources for this workshop.

```
source ~/advanced-kubernetes-workshop/setup/setup.sh
```

Note: This step takes about 20 - 25 minutes to complete. Please ensure you do not close Cloud Shell during the setup.

00

What you build



LAB 0: SETUP

GKE-SPINNAKER

GKE-CENTRAL

GKE-WEST

LAB 0: SETUP

GKE-SPINNAKER

SPINNAKER 1.8

ISTIO-SYSTEM

ISTIO 1.2.5

ISTIO-SYSTEM

ISTIO 1.2.5

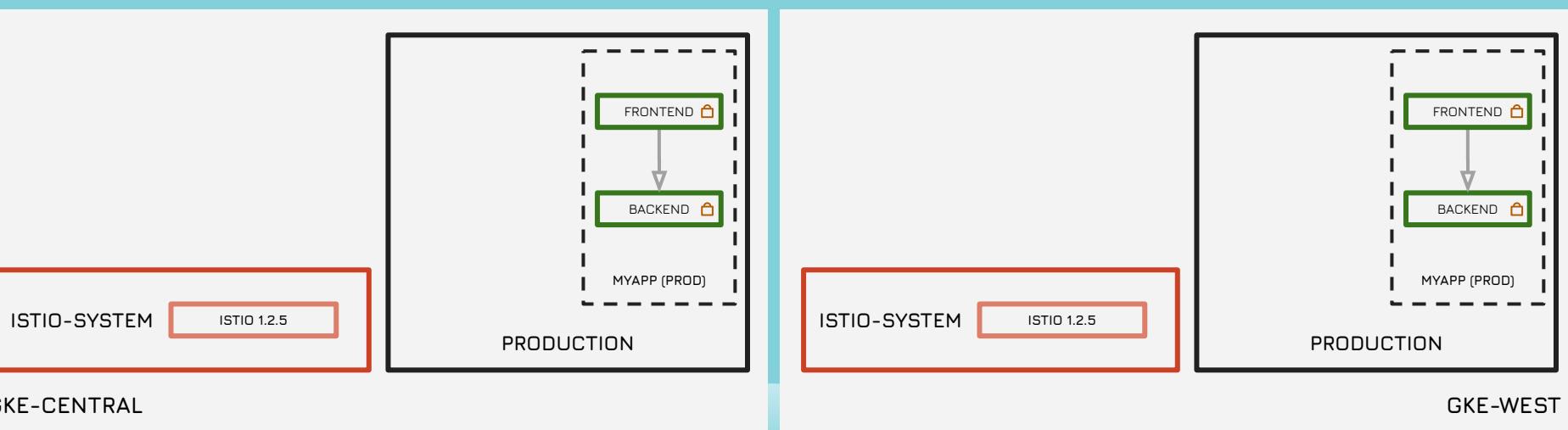
GKE-CENTRAL

GKE-WEST

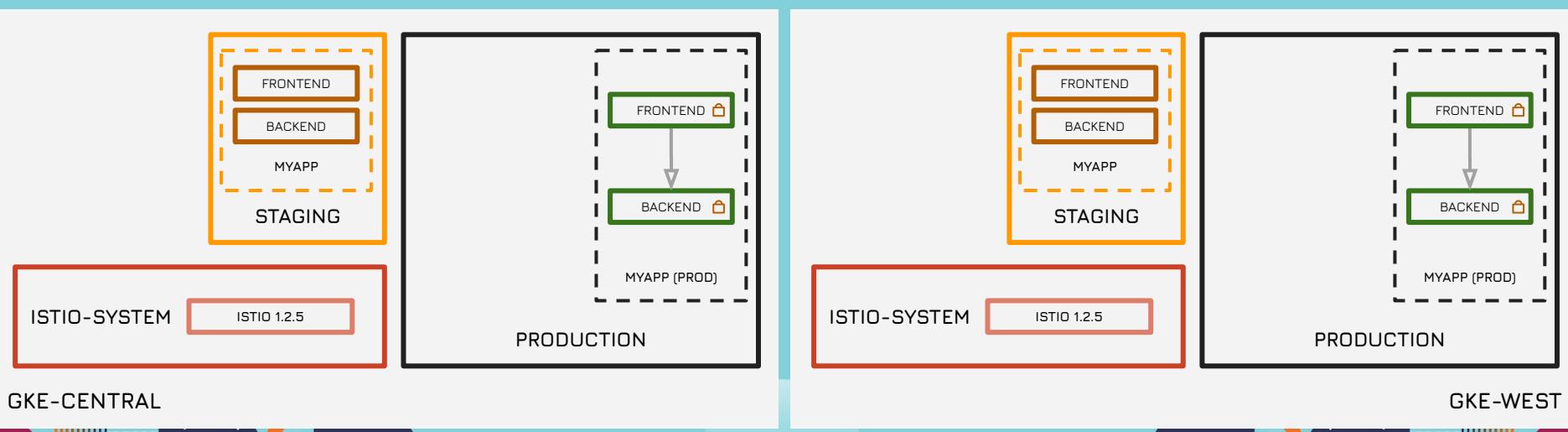
LAB 0: SETUP

GKE-SPINNAKER

SPINNAKER 1.8



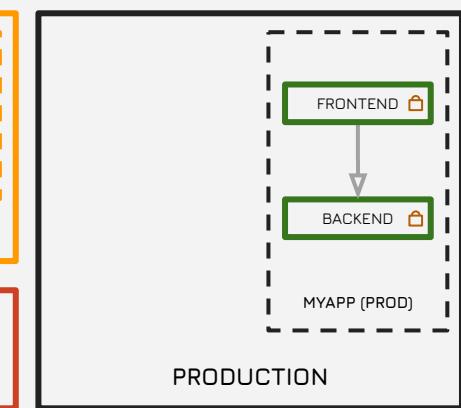
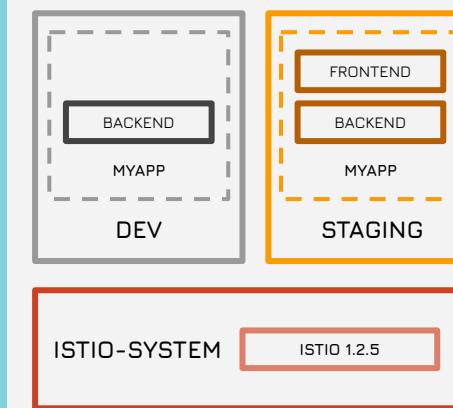
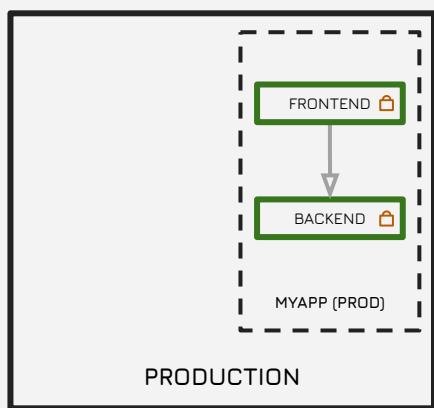
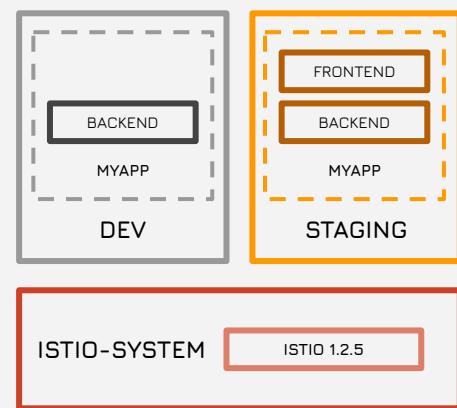
LAB 0: SETUP



LAB 0: SETUP

GKE-SPINNAKER

SPINNAKER 1.8



GKE-CENTRAL

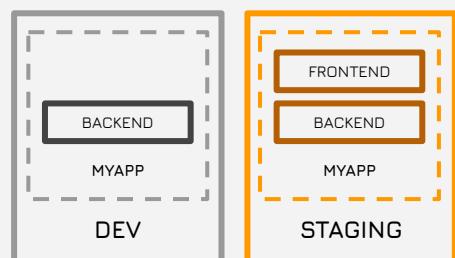
GKE-WEST

LAB 0: SETUP

GKE-SPINNAKER

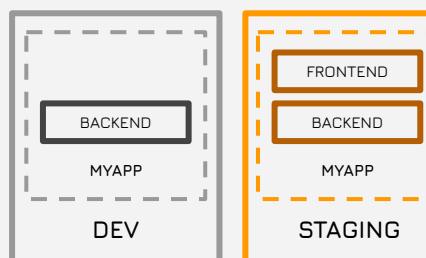
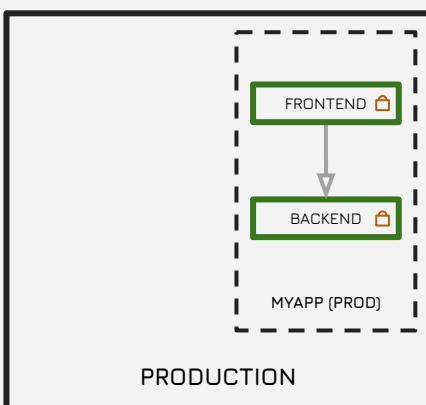
SPINNAKER 1.8

CONTAINER REGISTRY
FRONTEND
BACKEND
FRONTEND-DEV
BACKEND-DEV



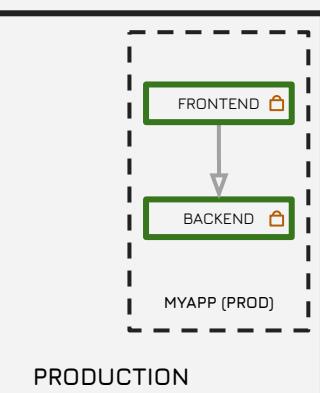
ISTIO-SYSTEM

ISTIO 1.2.5



ISTIO-SYSTEM

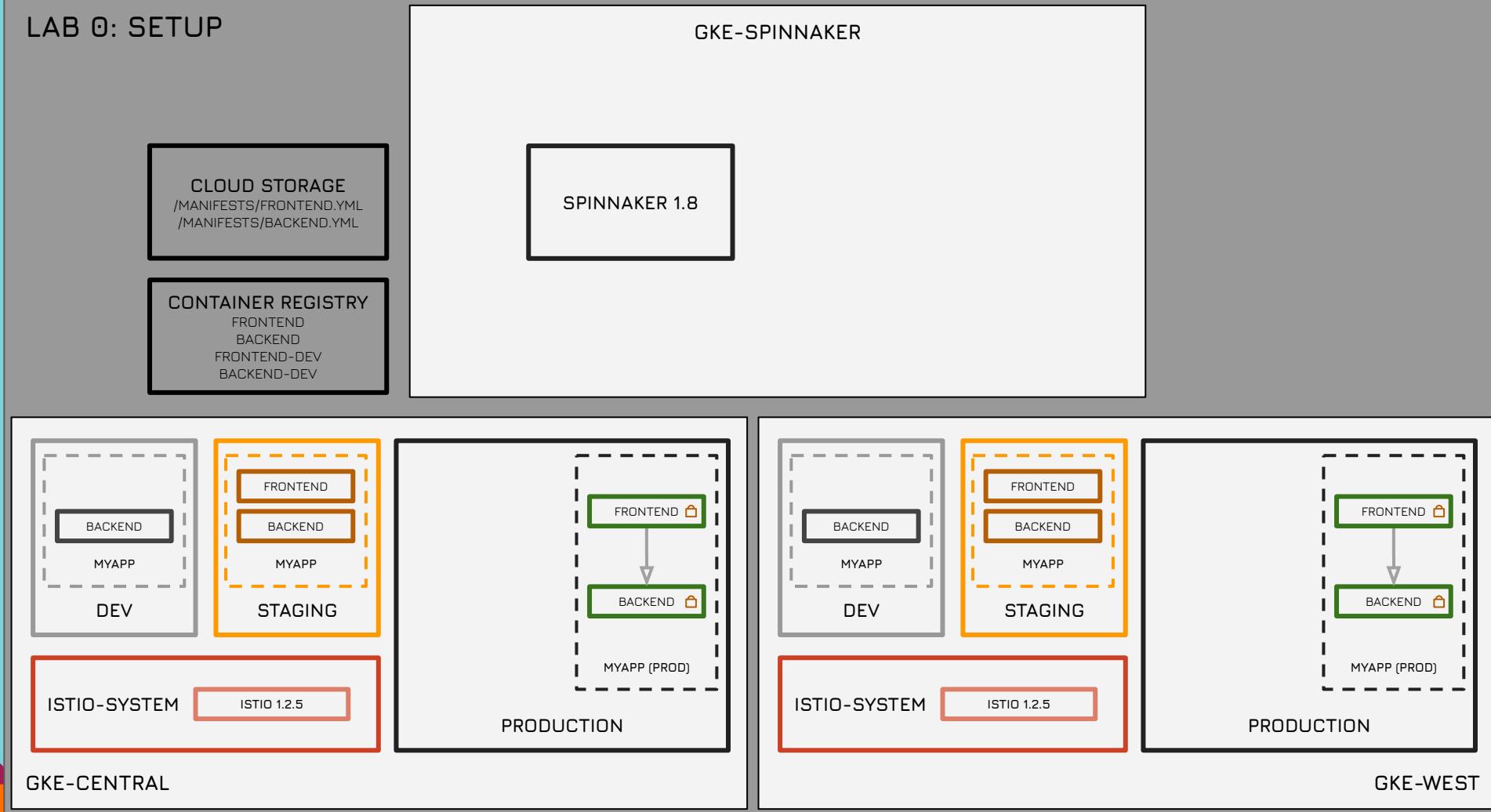
ISTIO 1.2.5



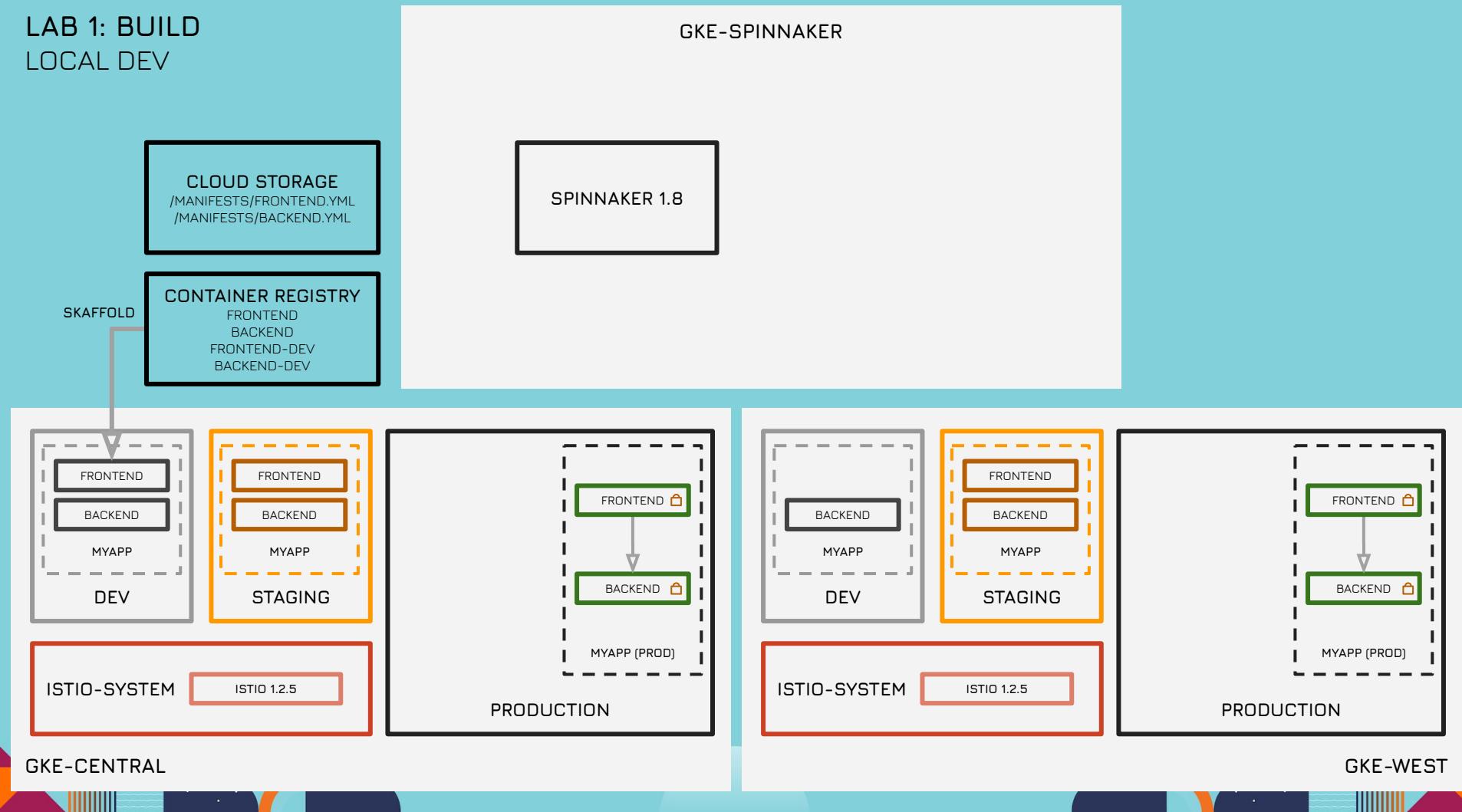
GKE-CENTRAL

GKE-WEST

LAB 0: SETUP



LAB 1: BUILD LOCAL DEV



LAB 1: BUILD

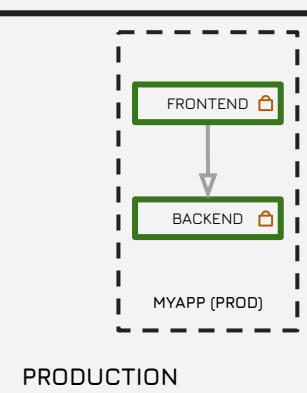
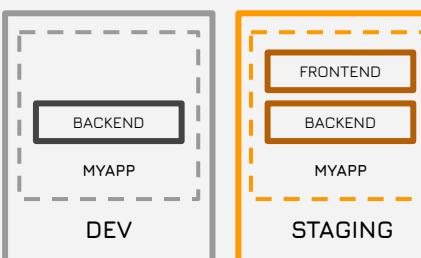
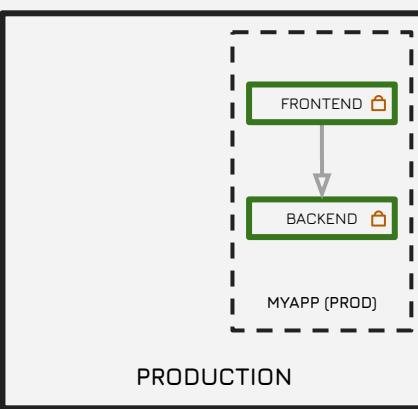
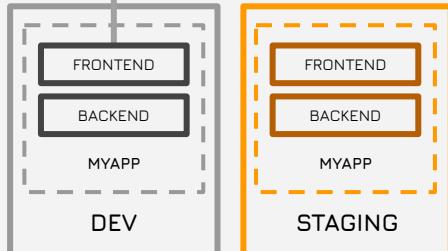
SKAFFOLD & CLOUD BUILD

GKE-SPINNAKER



SPINNAKER 1.8

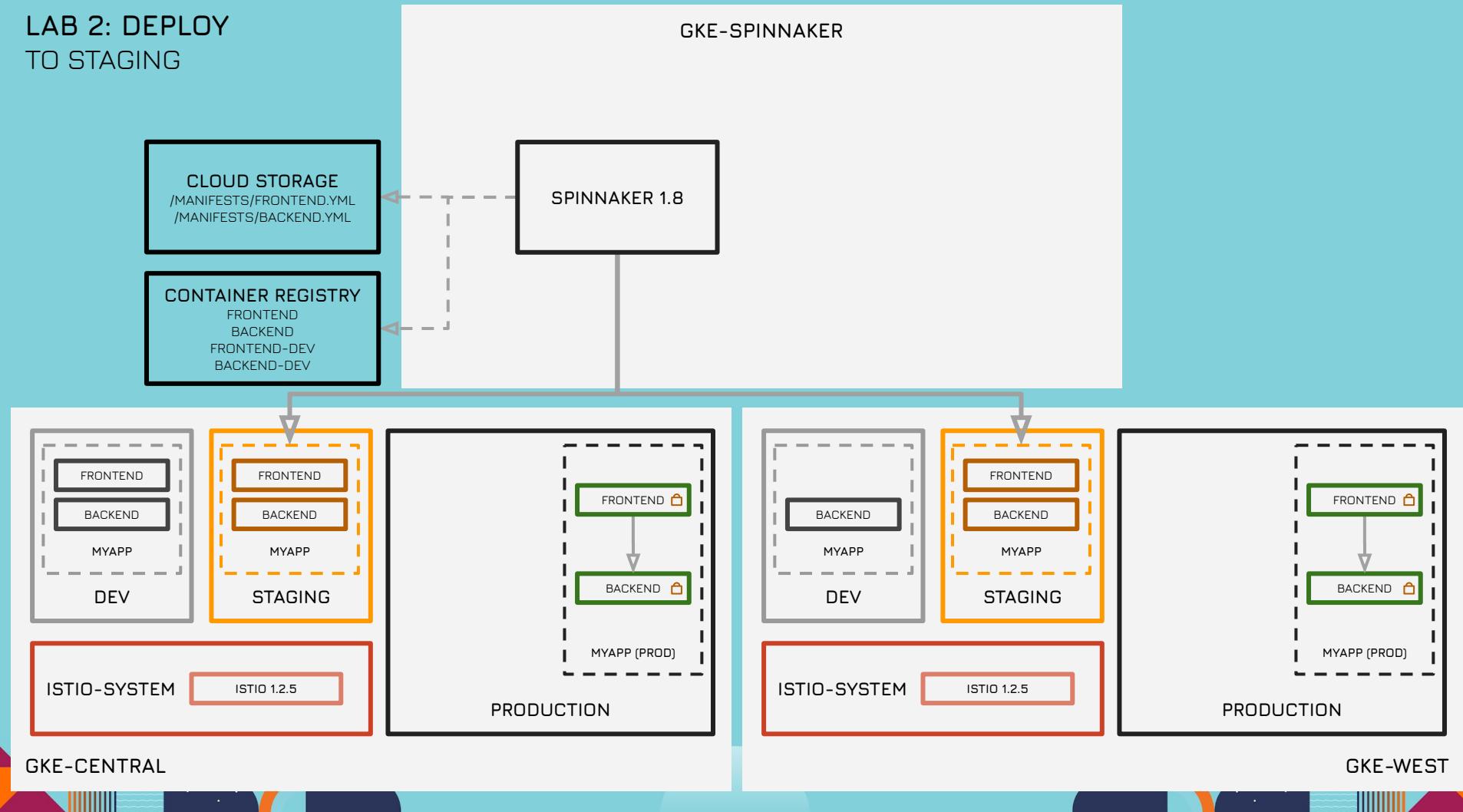
CLOUD BUILD



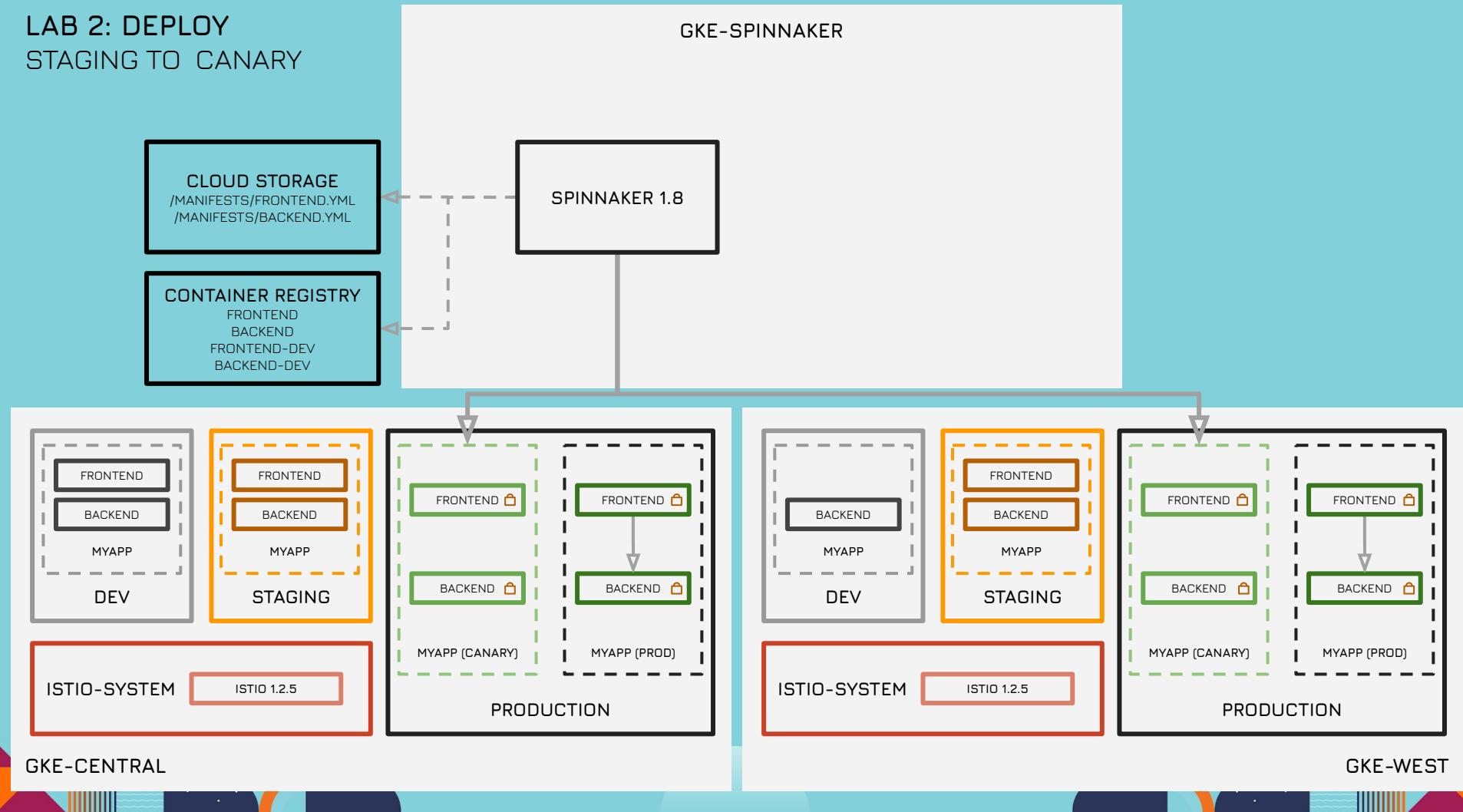
GKE-WEST

GKE-CENTRAL

LAB 2: DEPLOY TO STAGING



LAB 2: DEPLOY STAGING TO CANARY

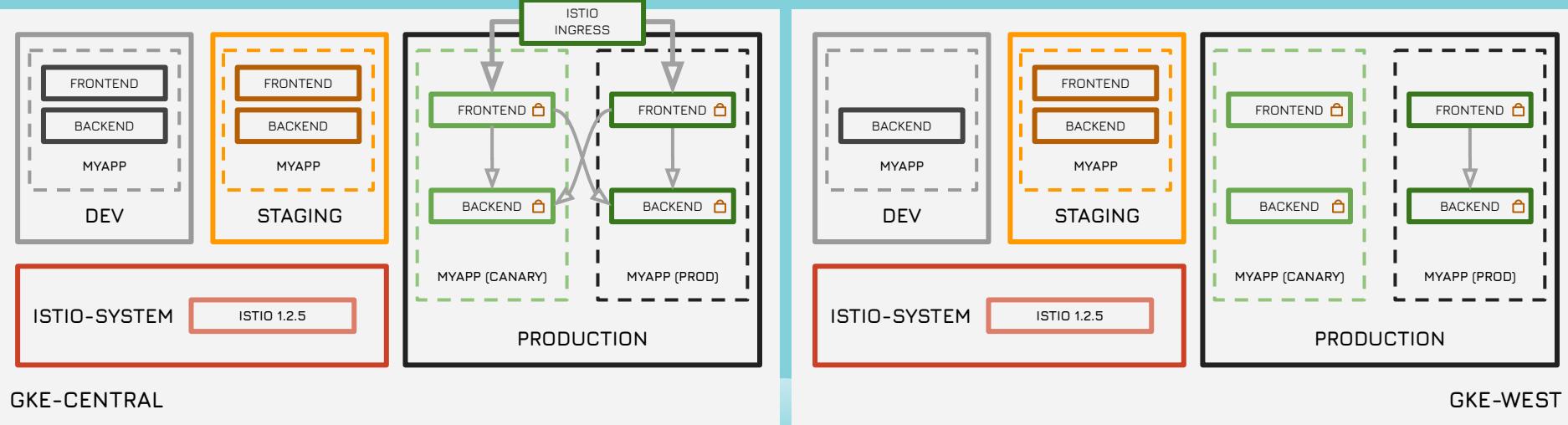


LAB 3: CONTROL ISTIO INGRESS

GKE-SPINNAKER



SPINNAKER 1.8



GKE-CENTRAL

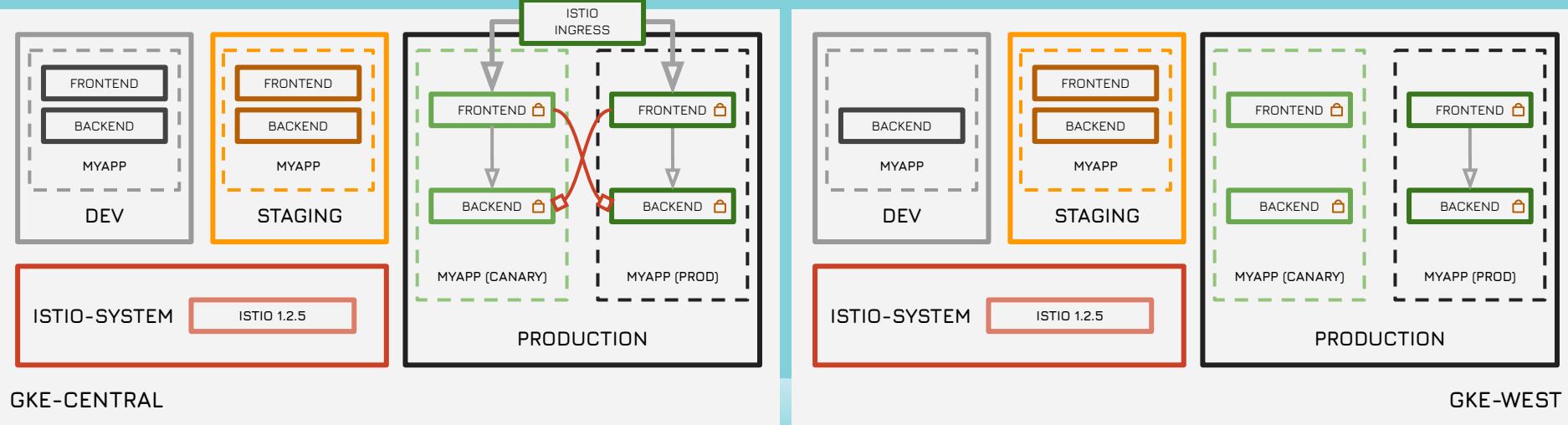
GKE-WEST

LAB 3: CONTROL TRAFFIC MGT

GKE-SPINNAKER



SPINNAKER 1.8



GKE-CENTRAL

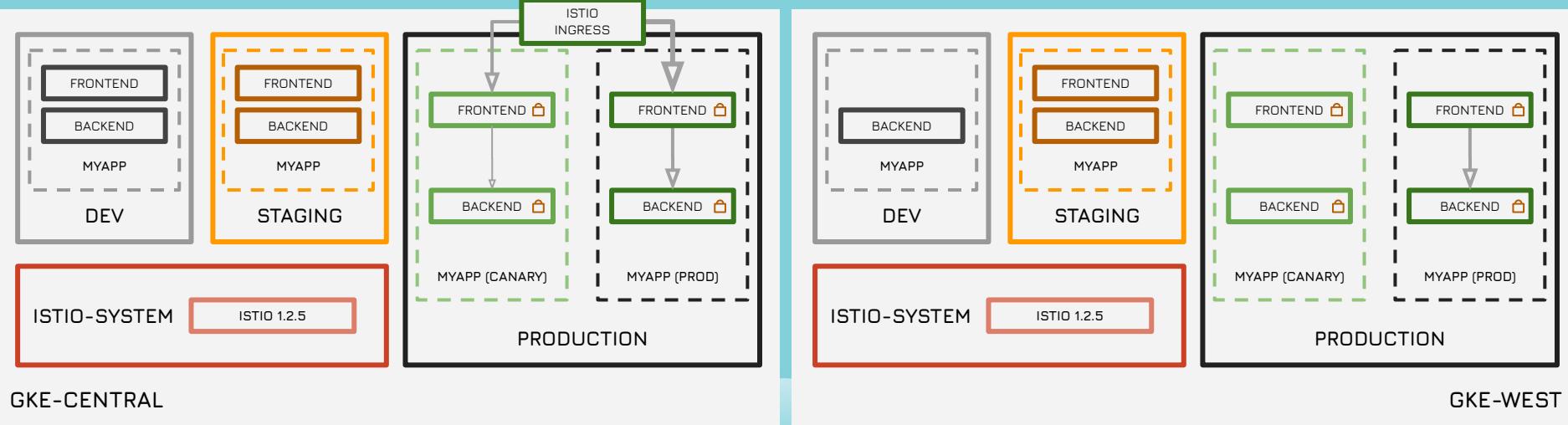
GKE-WEST

LAB 3: CONTROL RATE LIMITING

GKE-SPINNAKER



SPINNAKER 1.8



GKE-CENTRAL

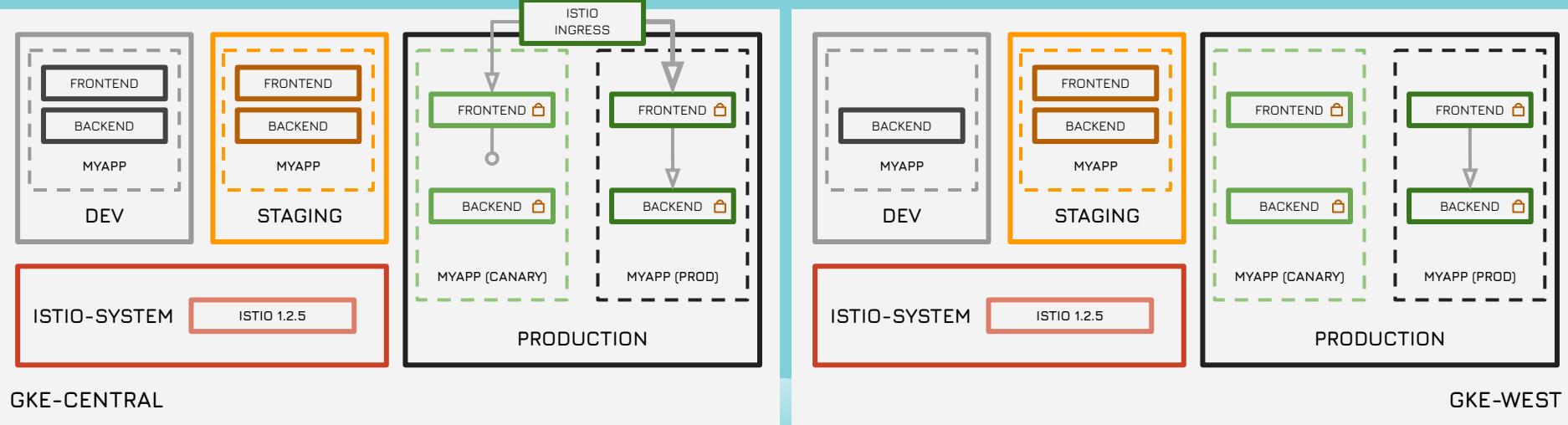
GKE-WEST

LAB 3: CONTROL CIRCUIT BREAKING

GKE-SPINNAKER



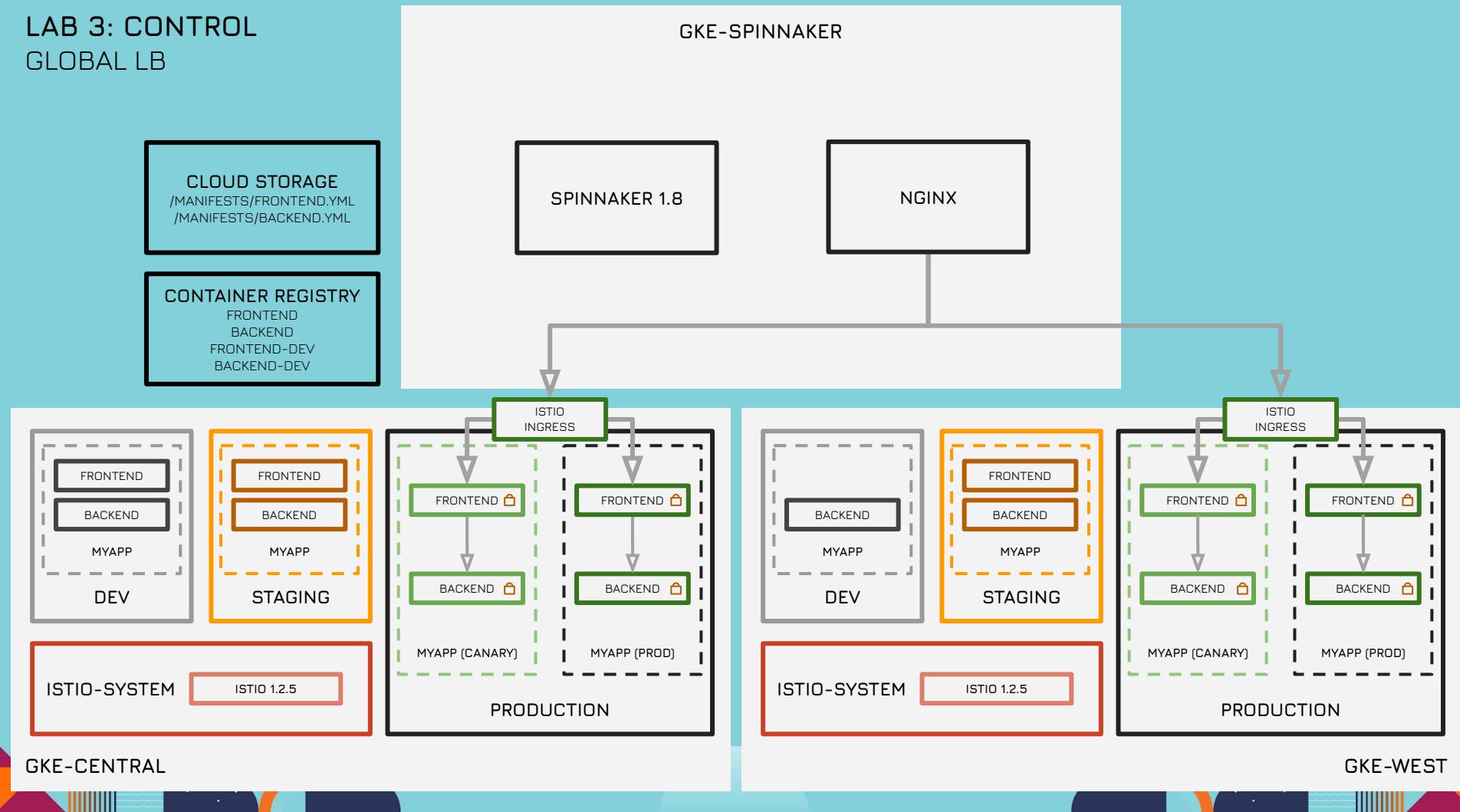
SPINNAKER 1.8



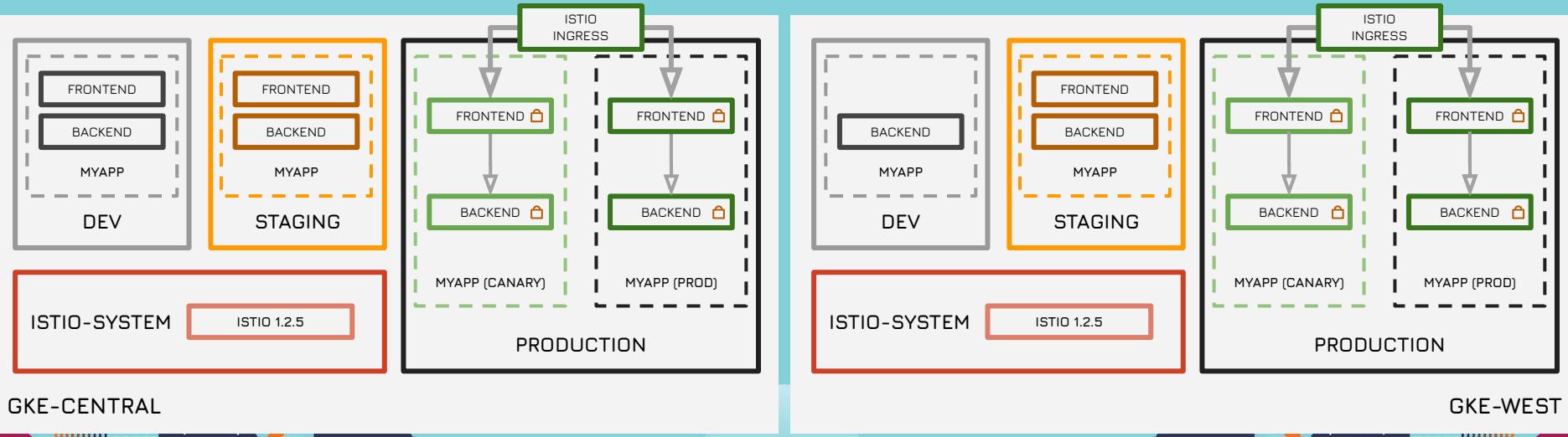
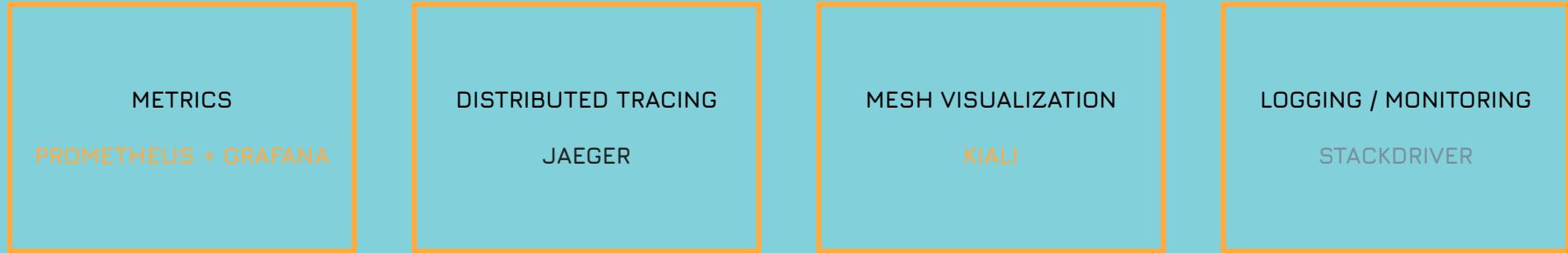
GKE-CENTRAL

GKE-WEST

LAB 3: CONTROL GLOBAL LB



LAB 4: MONITOR



GKE-CENTRAL

GKE-WEST

LAB 4: MONITOR

METRICS, TRACING, VISUALIZATION, LOGGING

METRICS

PROMETHEUS + GRAFANA

DISTRIBUTED TRACING

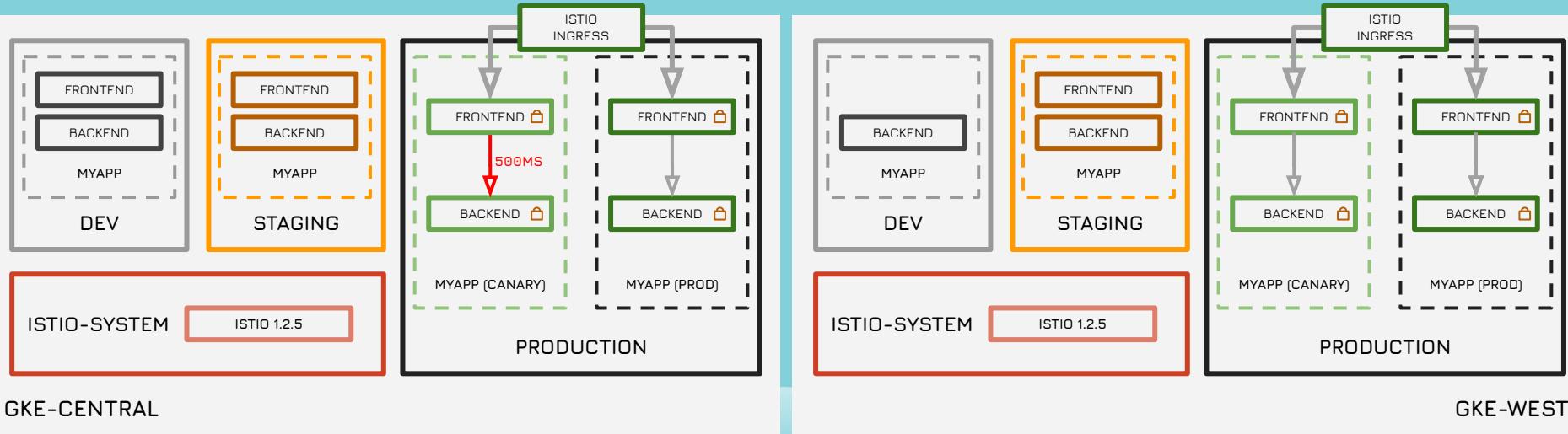
JAEGER

MESH VISUALIZATION

KIALI

LOGGING / MONITORING

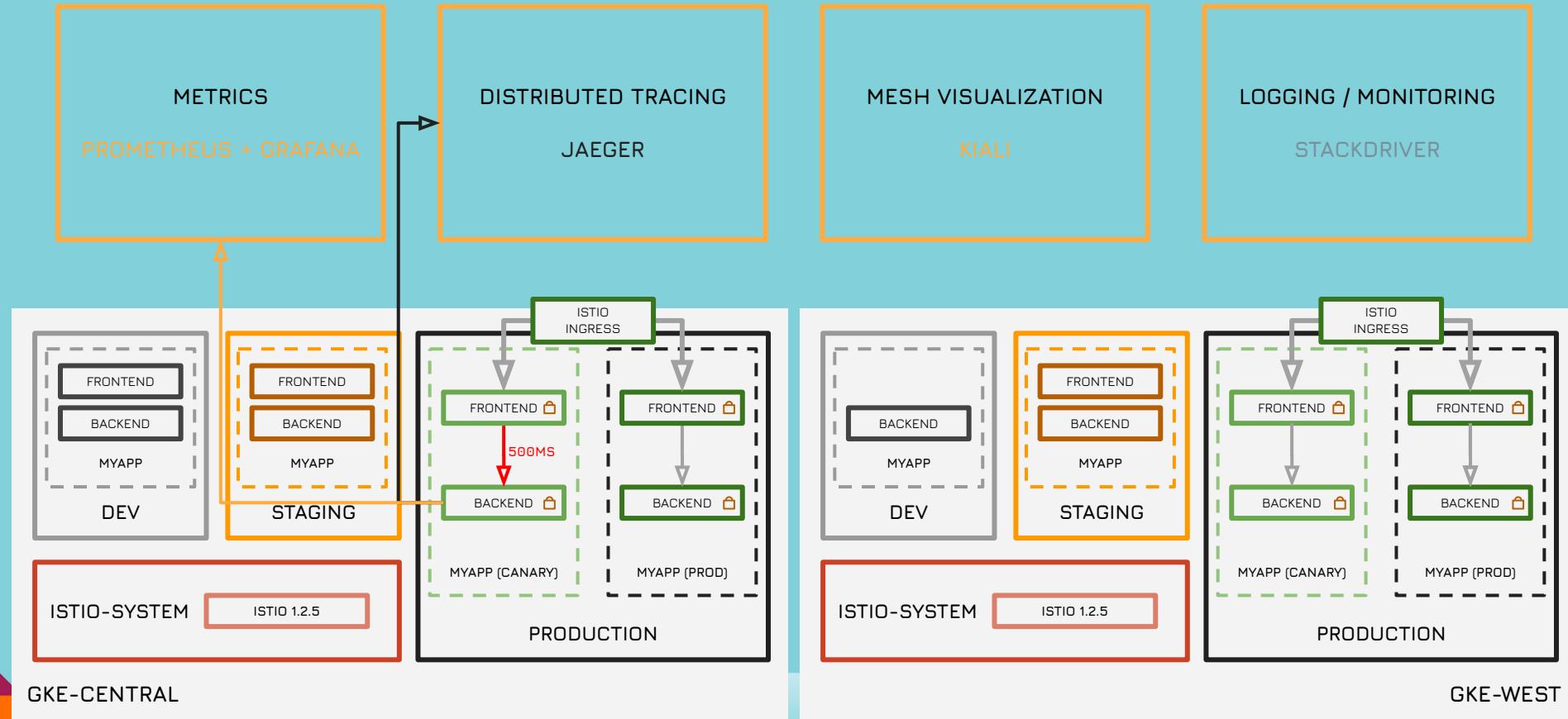
STACKDRIVER



GKE-CENTRAL

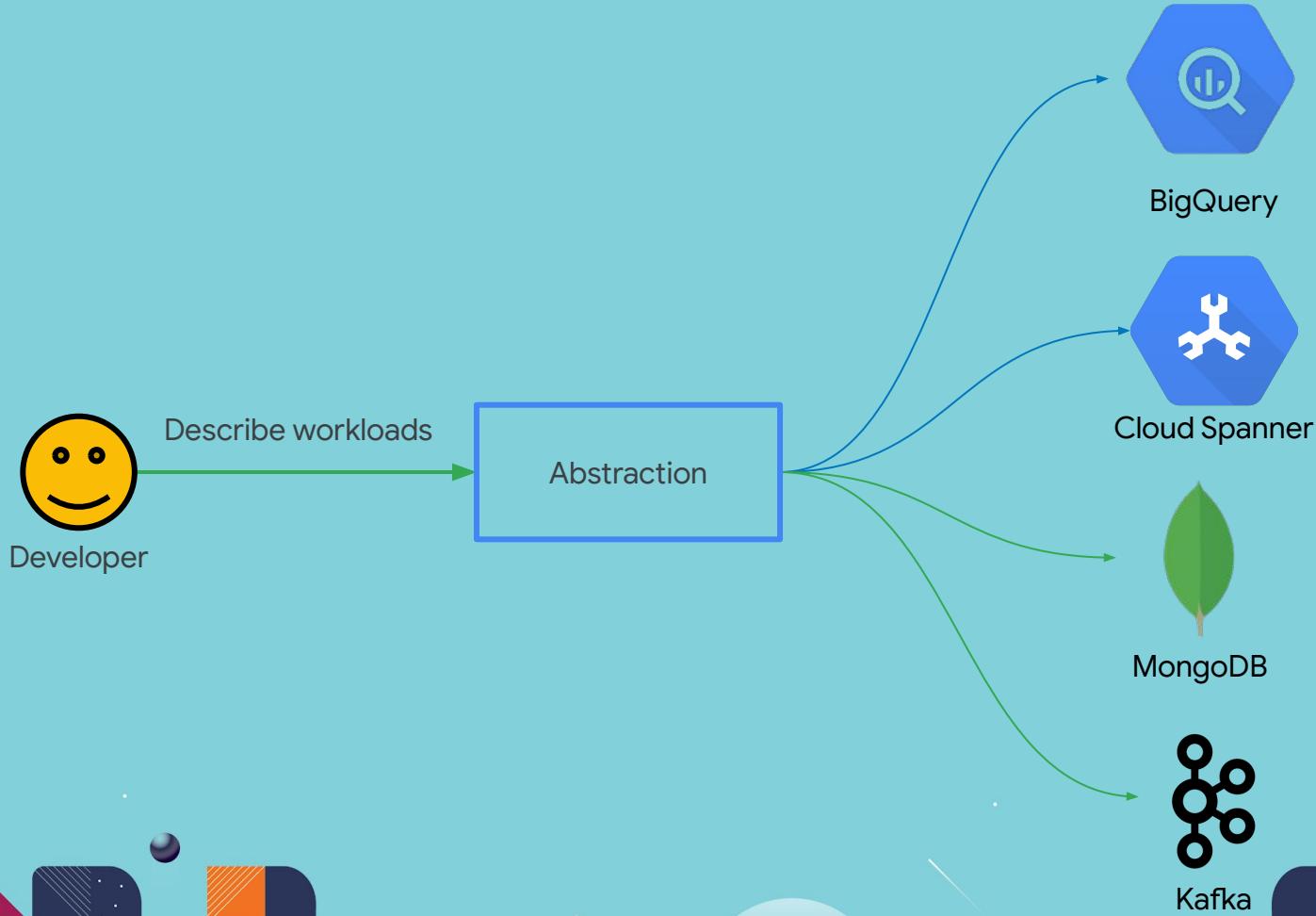
GKE-WEST

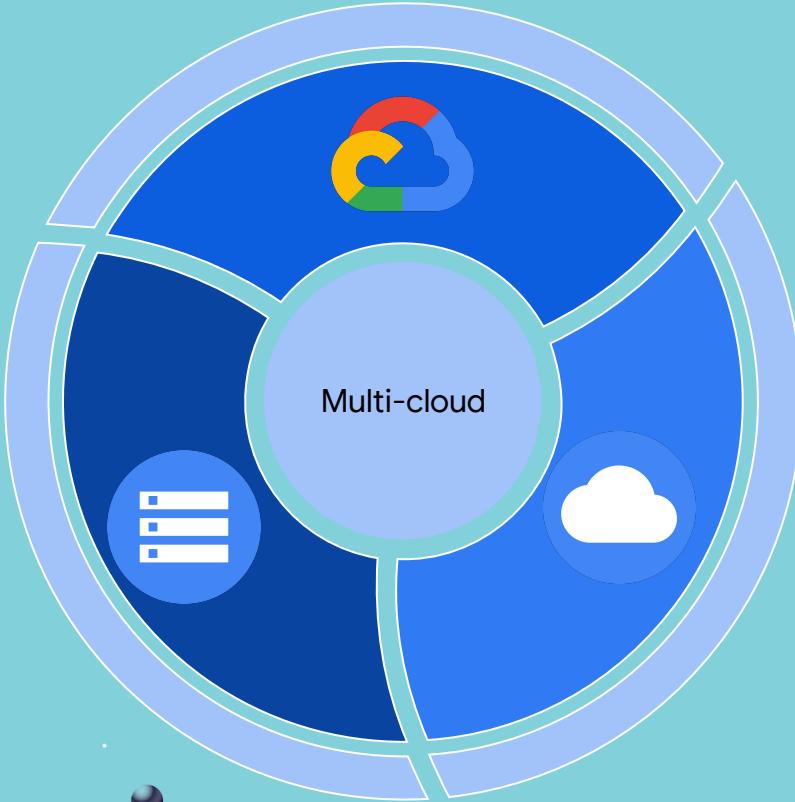
LAB 4: MONITOR TROUBLESHOOTING



02

Distributed Applications in Multi-cloud





Zero vendor lock-in

Cost & Control

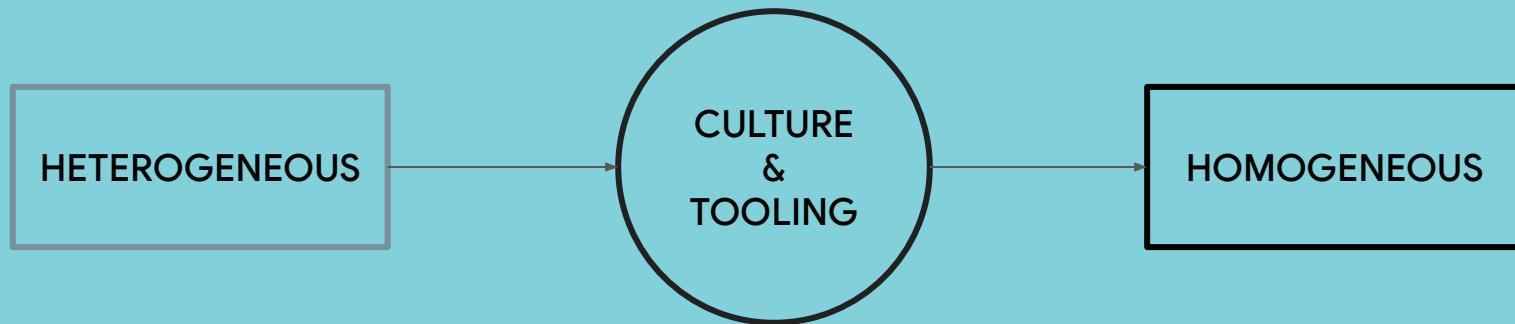
Resiliency & Availability

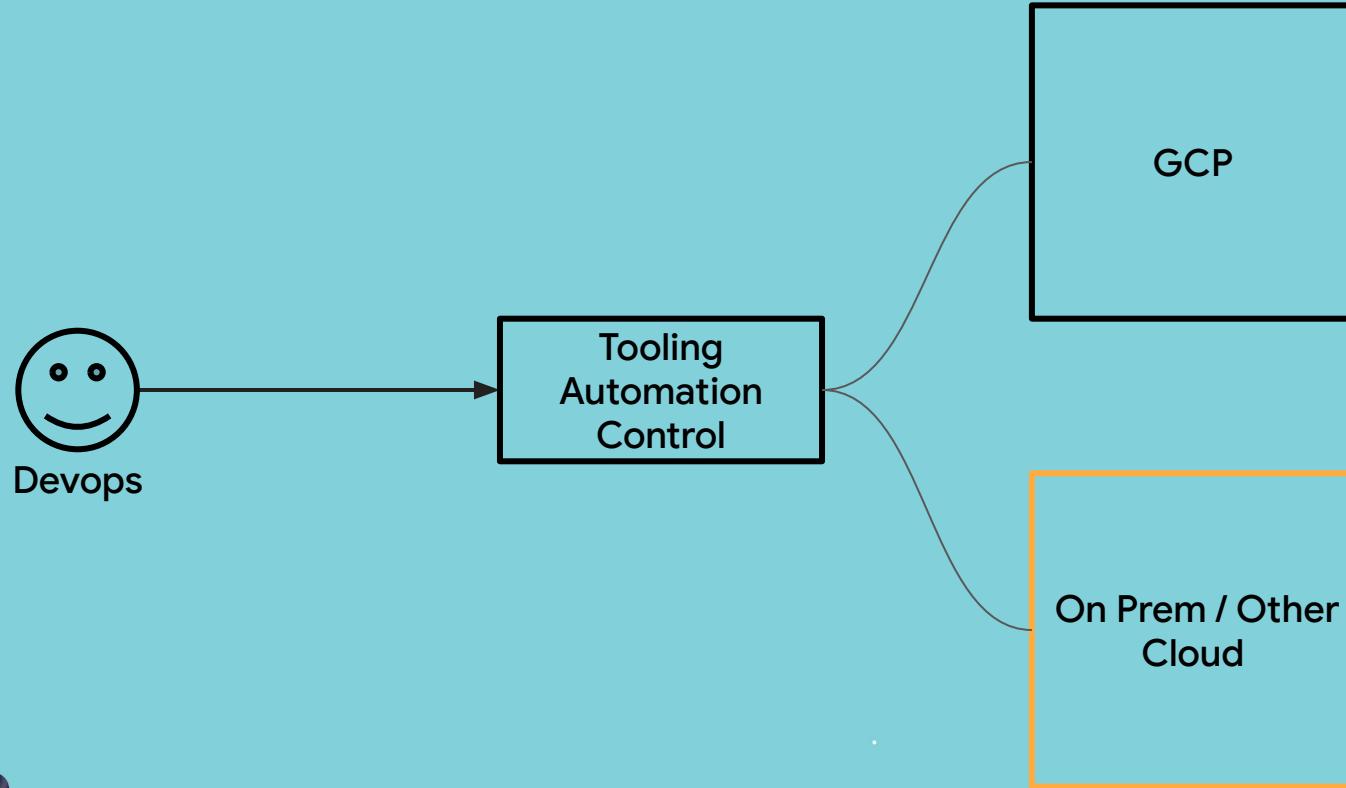
Services & Innovation

Technology / Business Partnerships

Opinionated Systems vs Freedom of Choice







SERVICE MESH

Connect and secure applications

CI/CD

Manage applications

ORCHESTRATION

Run applications

CONTAINER

Package applications

ISTIO

Connect and secure applications

SPINNAKER

Manage applications

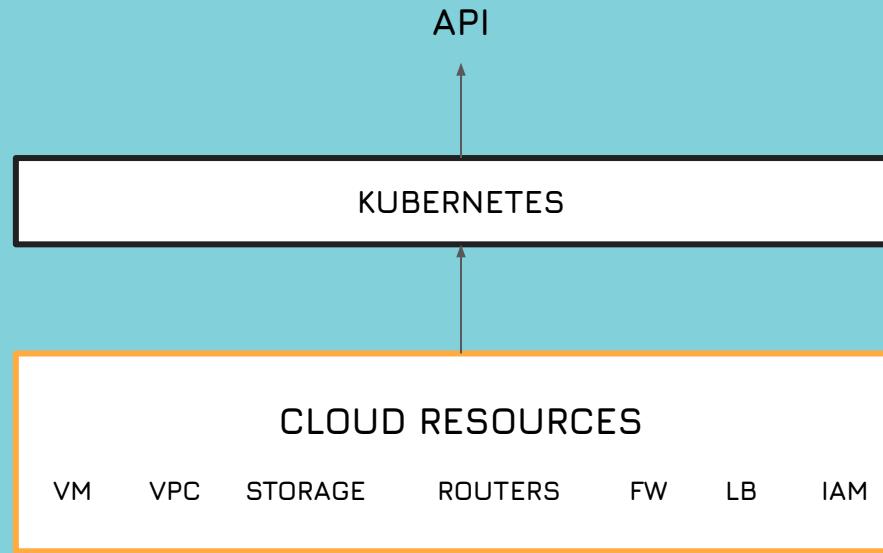
KUBERNETES

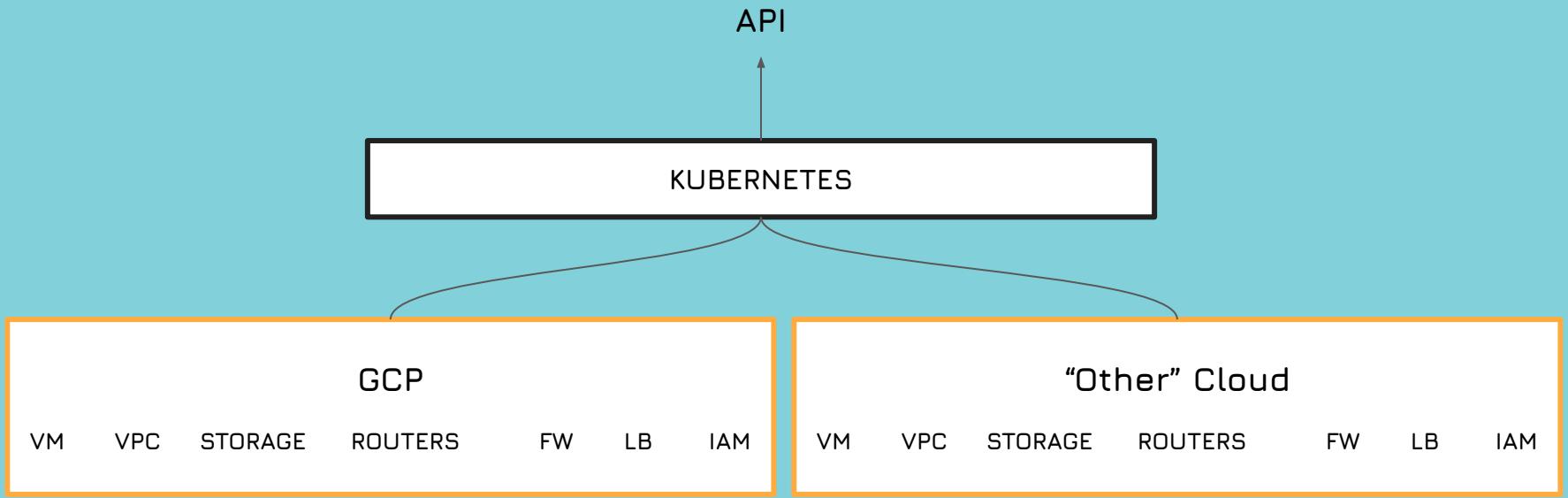
Run applications

DOCKER

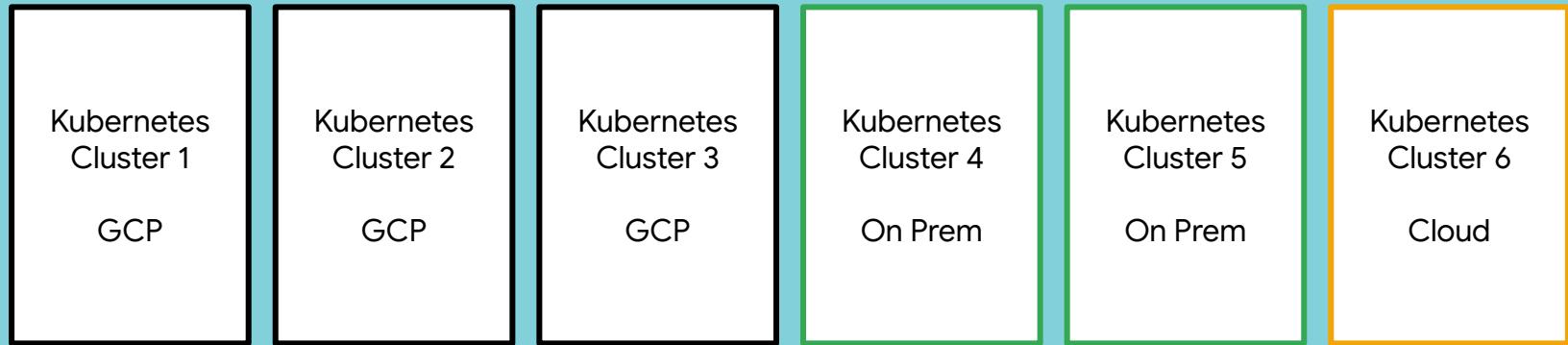
Package applications

Kubernetes is a **declarative** way to **describe** your applications

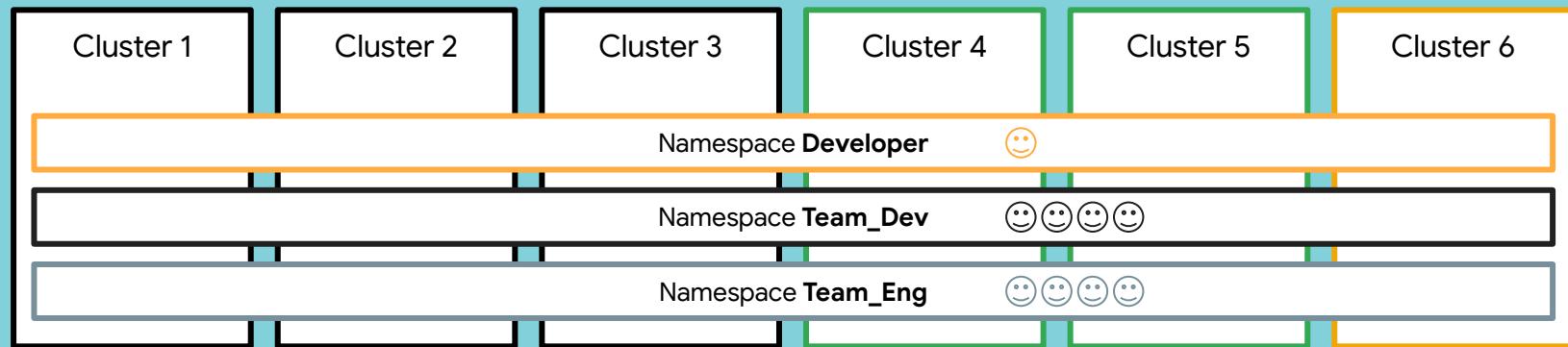


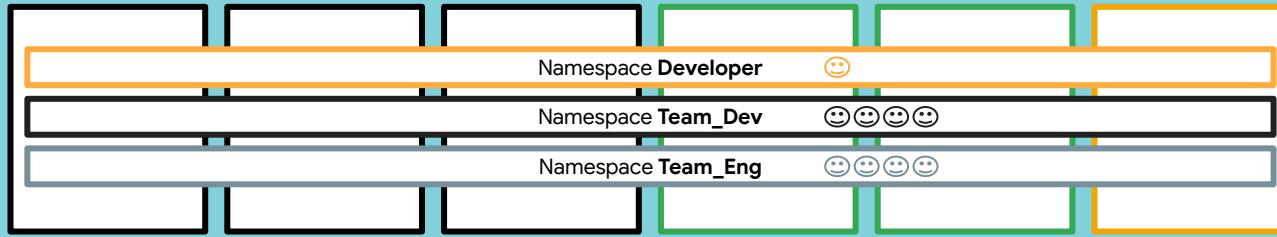


Cluster Architecture & Management

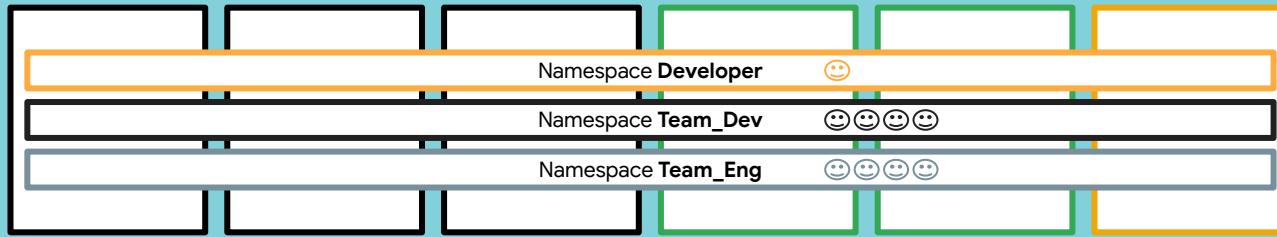


Cluster Consistency

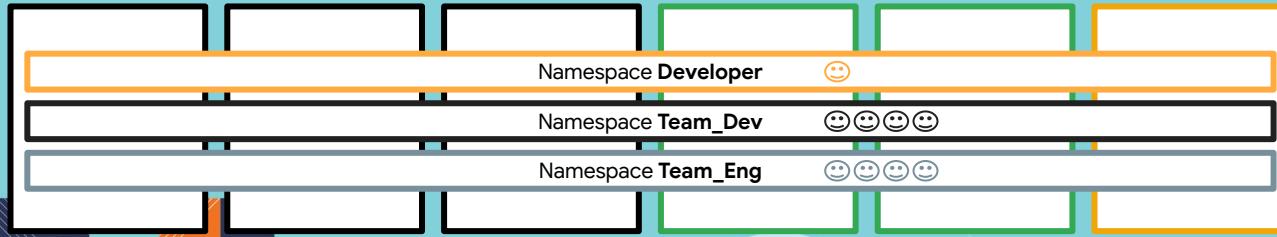




Prod



Stage



Dev



Cluster Management

Application Management

Service Discovery



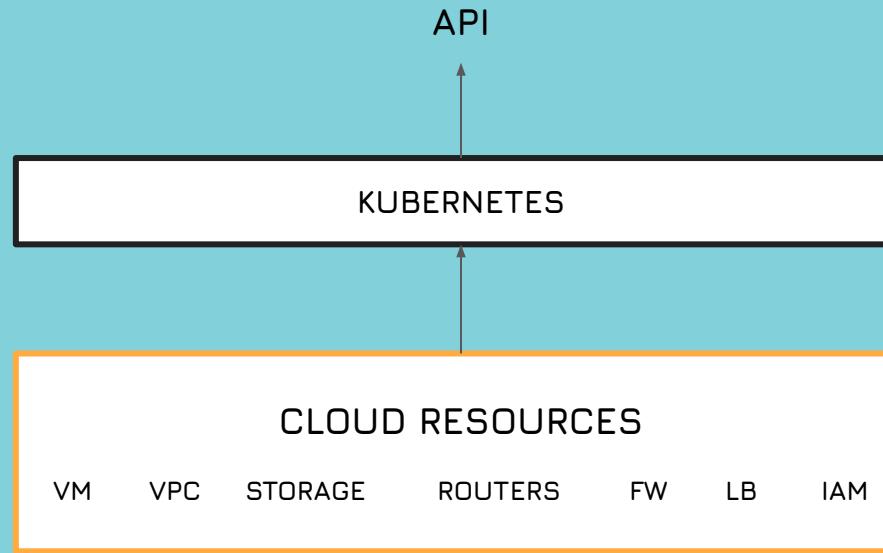
Federation vs Cluster Independence



03

Kubernetes

Kubernetes is a **declarative** way to **describe** your applications

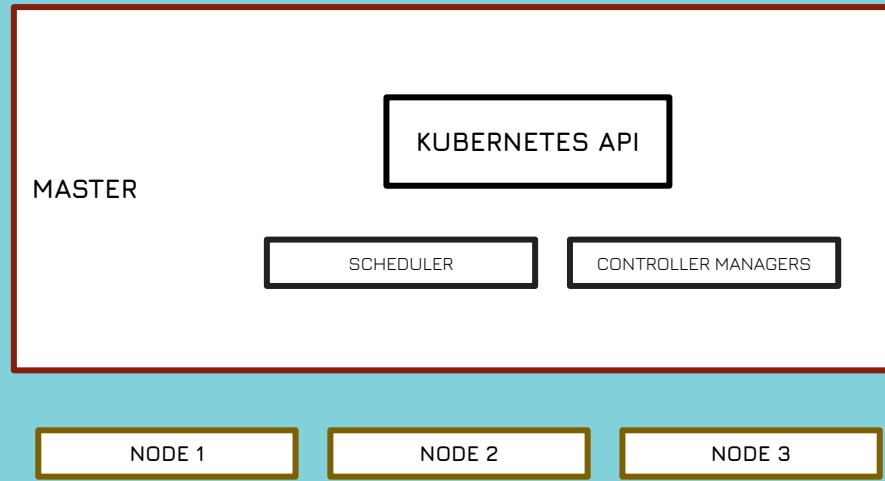


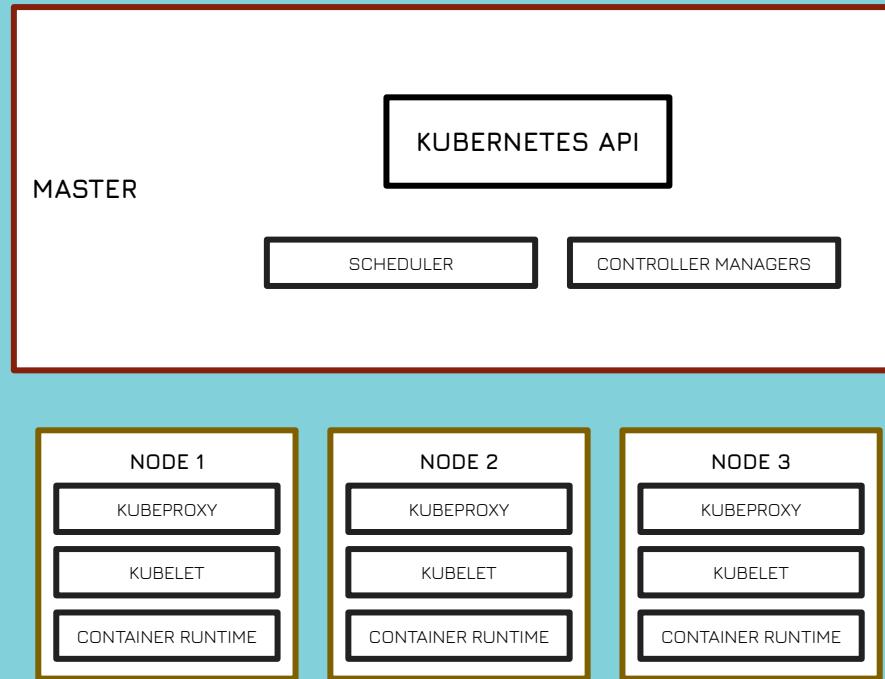
MASTER

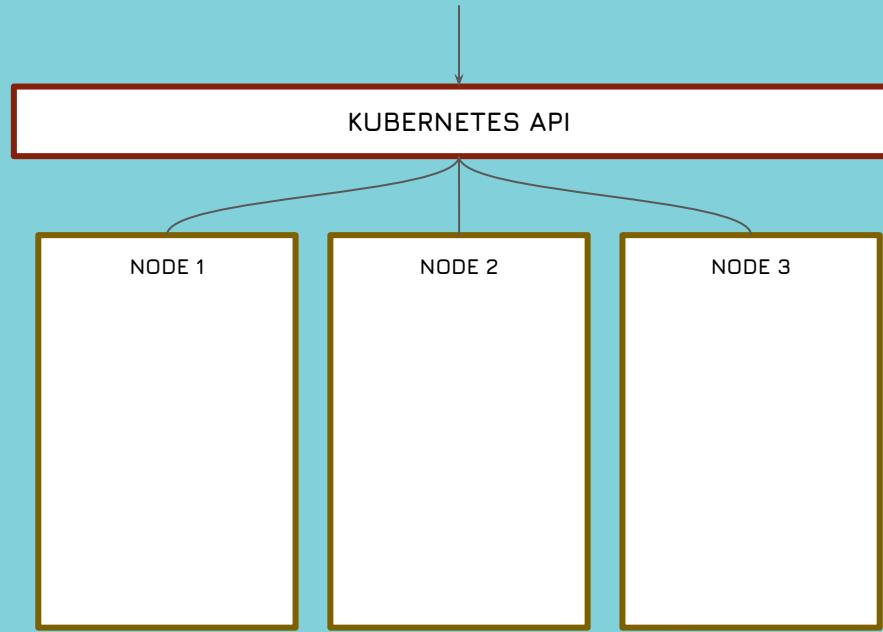
NODE 1

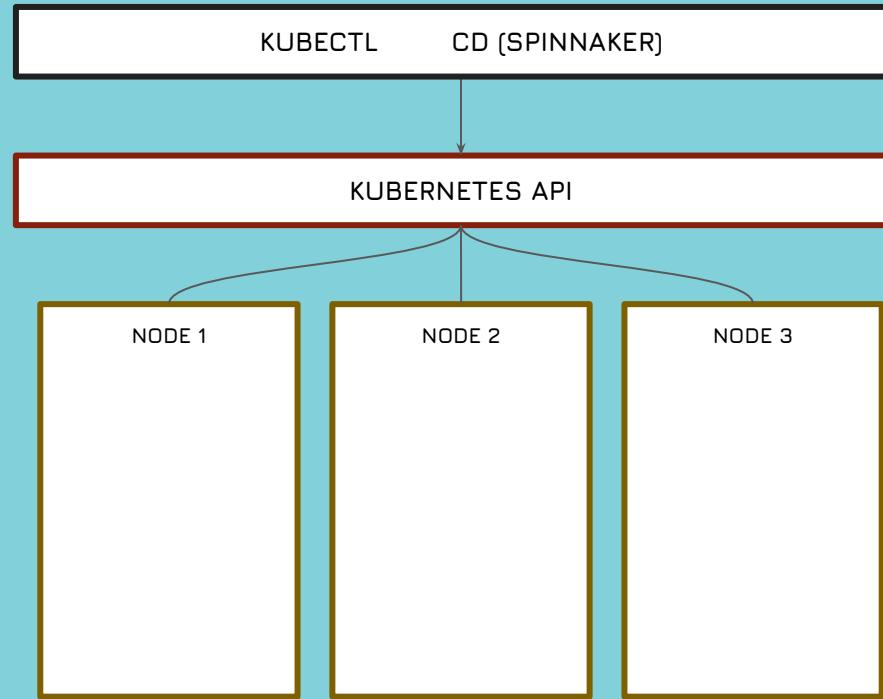
NODE 2

NODE 3











Kube C.T.L

KUBECUTTLE

kubec-tl





kubectl
(pronounced kube control)





Everything in Kubernetes is a Resource





Resources are endpoints in Kubernetes API

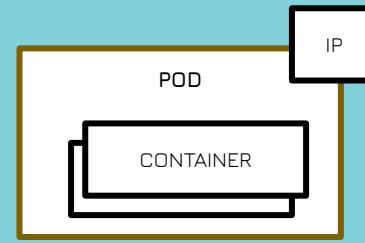




Labels to keep track of Resources



Pod Deployment Service Ingress Secret Configmap



"app": "frontend"

DEPLOYMENT - FRONTEND

POD

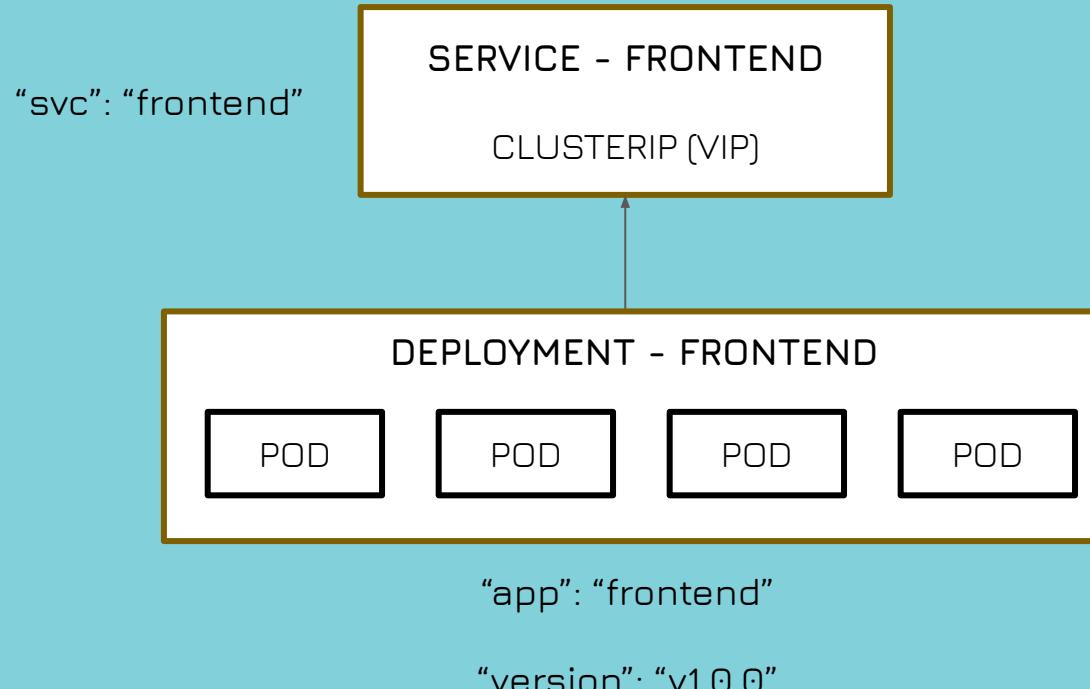
POD

POD

POD

“app”: “frontend”

“version”: “v1.0.0”



“svc”: “frontend”

SERVICE - FRONTEND
CLUSTERIP (VIP)

DEPLOYMENT

“app”: “frontend”

“version”: “v1.0.0”

DEPLOYMENT

“app”: “frontend”

“version”: “v1.0.1”

“svc”:
“frontend”

SERVICE - FRONTEND
CLUSTERIP (VIP)

DEPLOYMENT

“app”: “frontend”
“version”: “v1.0.0”

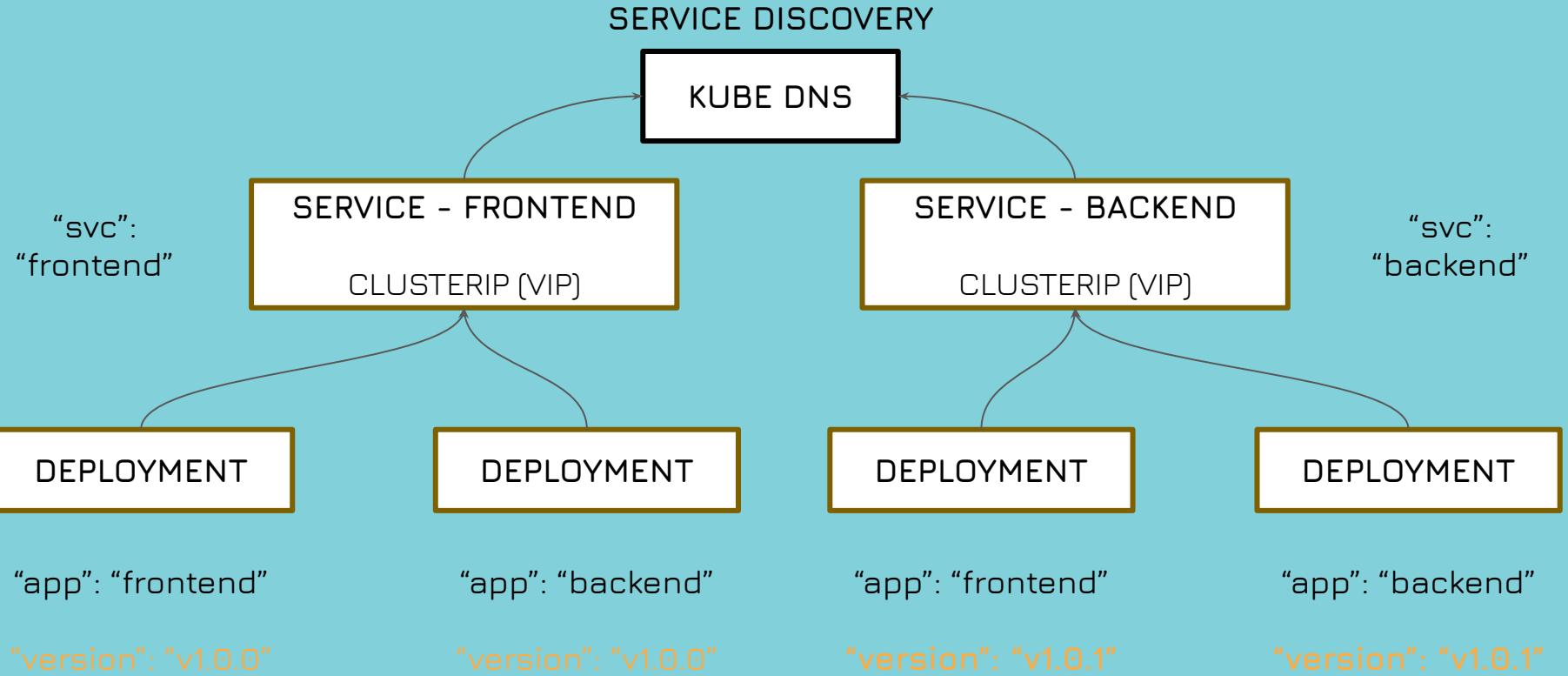
SERVICE - BACKEND
CLUSTERIP (VIP)

DEPLOYMENT

“svc”:
“backend”

DEPLOYMENT

“app”: “backend”
“version”: “v1.0.1”



“ingress”:
“frontend”

INGRESS

RESOURCE &
CONTROLLER

“svc”:
“frontend”

SERVICE - FRONTEND

CLUSTERIP (VIP)

DEPLOYMENT

DEPLOYMENT

“app”: “frontend”

“version”: “v1.0.0”

“app”: “backend”

“version”: “v1.0.0”

INGRESS

RESOURCE &
CONTROLLER

“ingress”:
“backend”

SERVICE - BACKEND

CLUSTERIP (VIP)

DEPLOYMENT

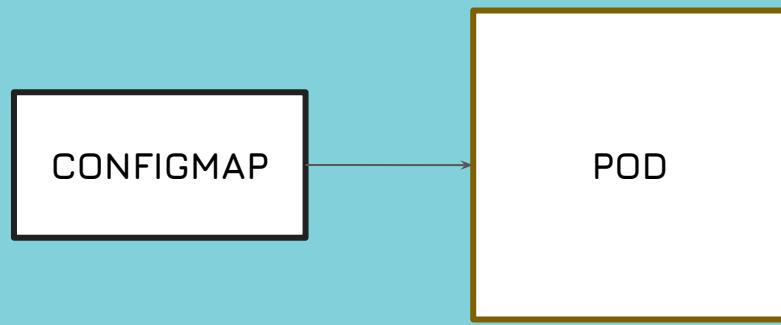
DEPLOYMENT

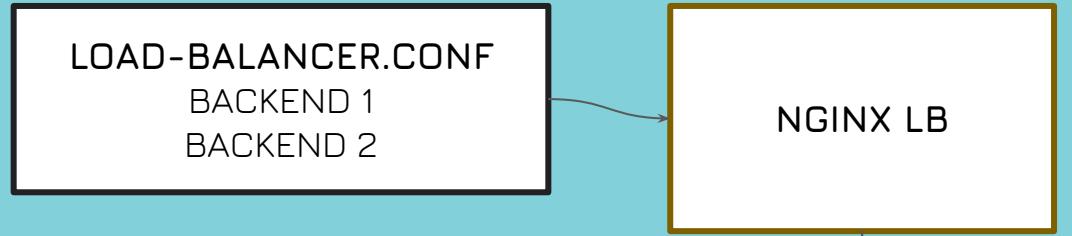
“app”: “frontend”

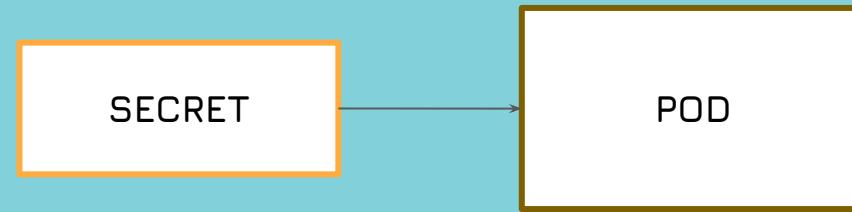
“version”: “v1.0.1”

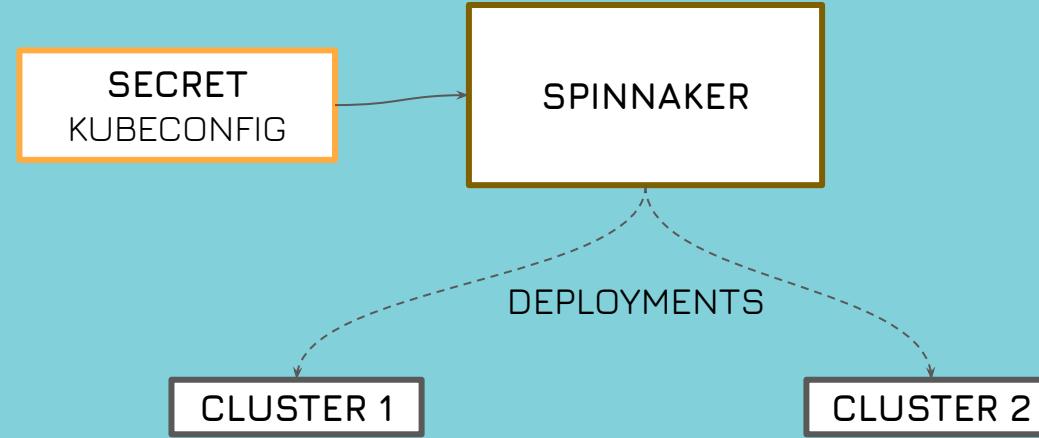
“app”: “backend”

“version”: “v1.0.1”









Custom Resources



ReactiveOps RBAC Manager

☰ README.md

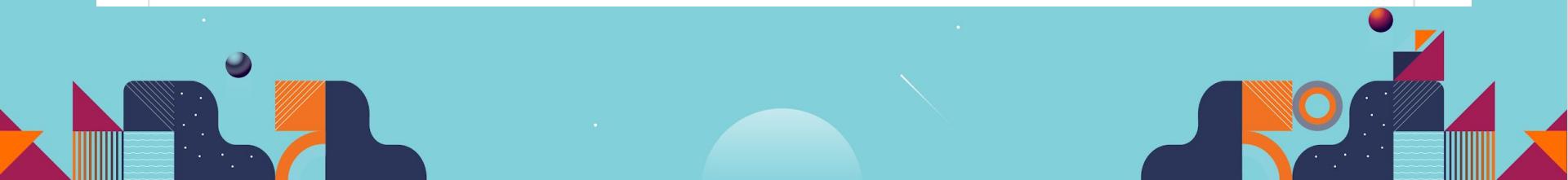
RBAC Manager

[go report](#) A+  PASSED

RBAC Manager was designed to simplify authorization in Kubernetes. This is an operator that supports declarative configuration for RBAC with new custom resources. Instead of managing role bindings or service accounts directly, you can specify a desired state and RBAC Manager will make the necessary changes to achieve that state.

This project has three main goals:

1. Provide a declarative approach to RBAC that is more approachable and scalable.
2. Reduce the amount of configuration required for great auth.
3. Enable automation of RBAC configuration updates with CI/CD.

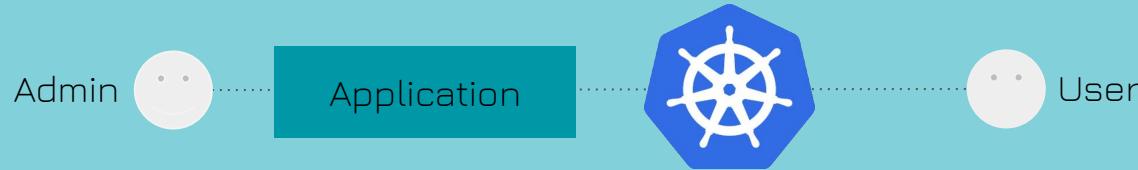


Istio



Bringing it all together

An Application Deployment to User Story



Admin



Admin



runs



Deployment

Admin



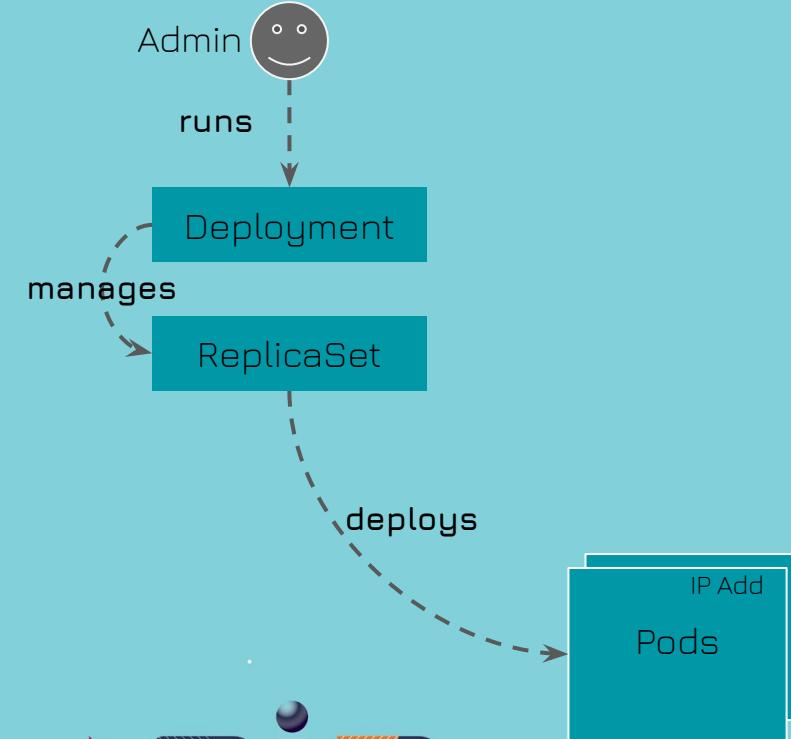
runs

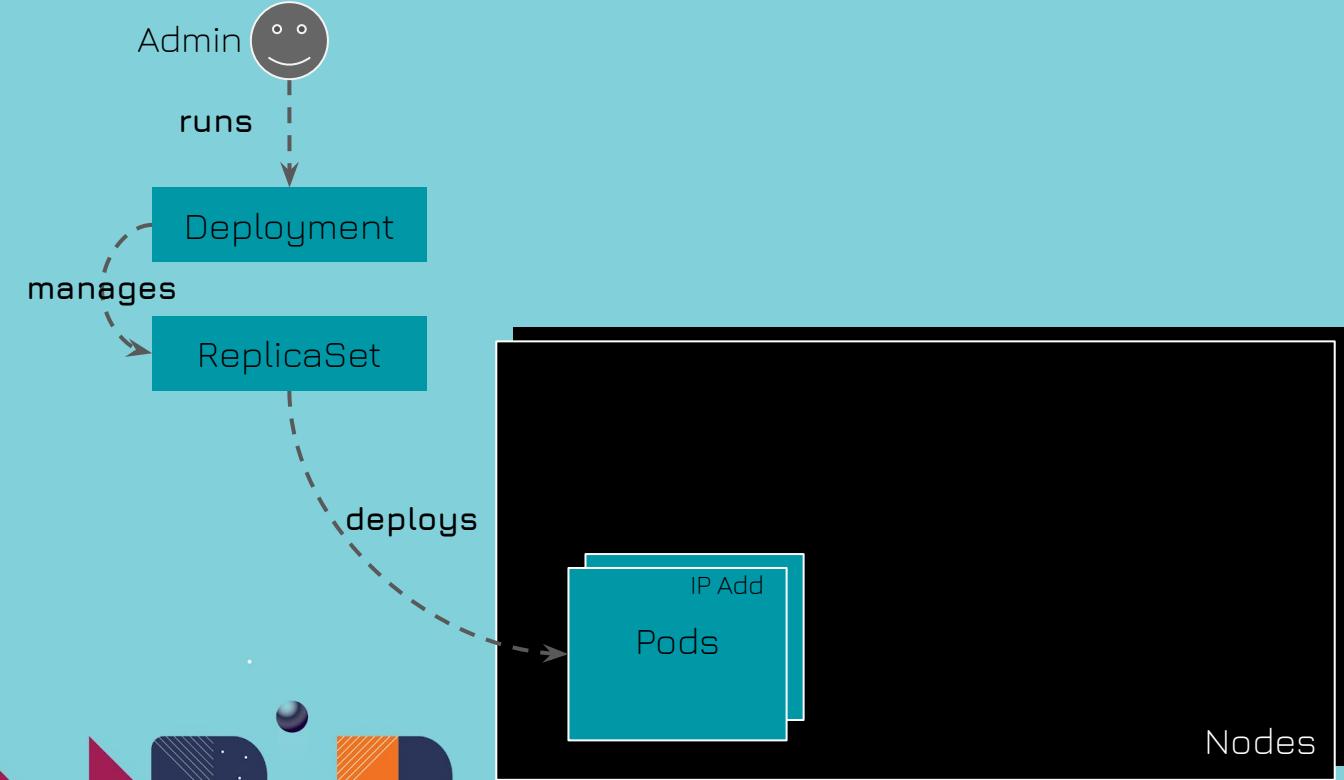
Deployment

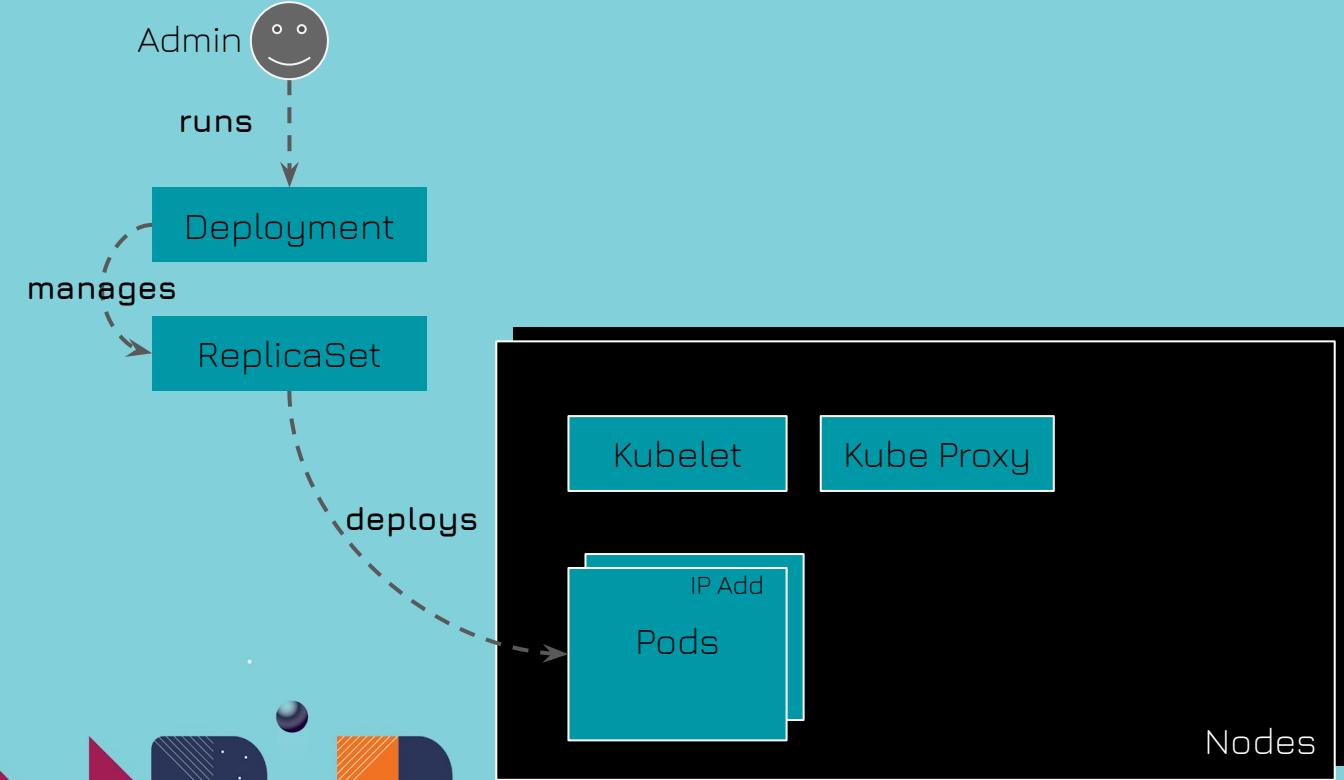
ReplicaSet

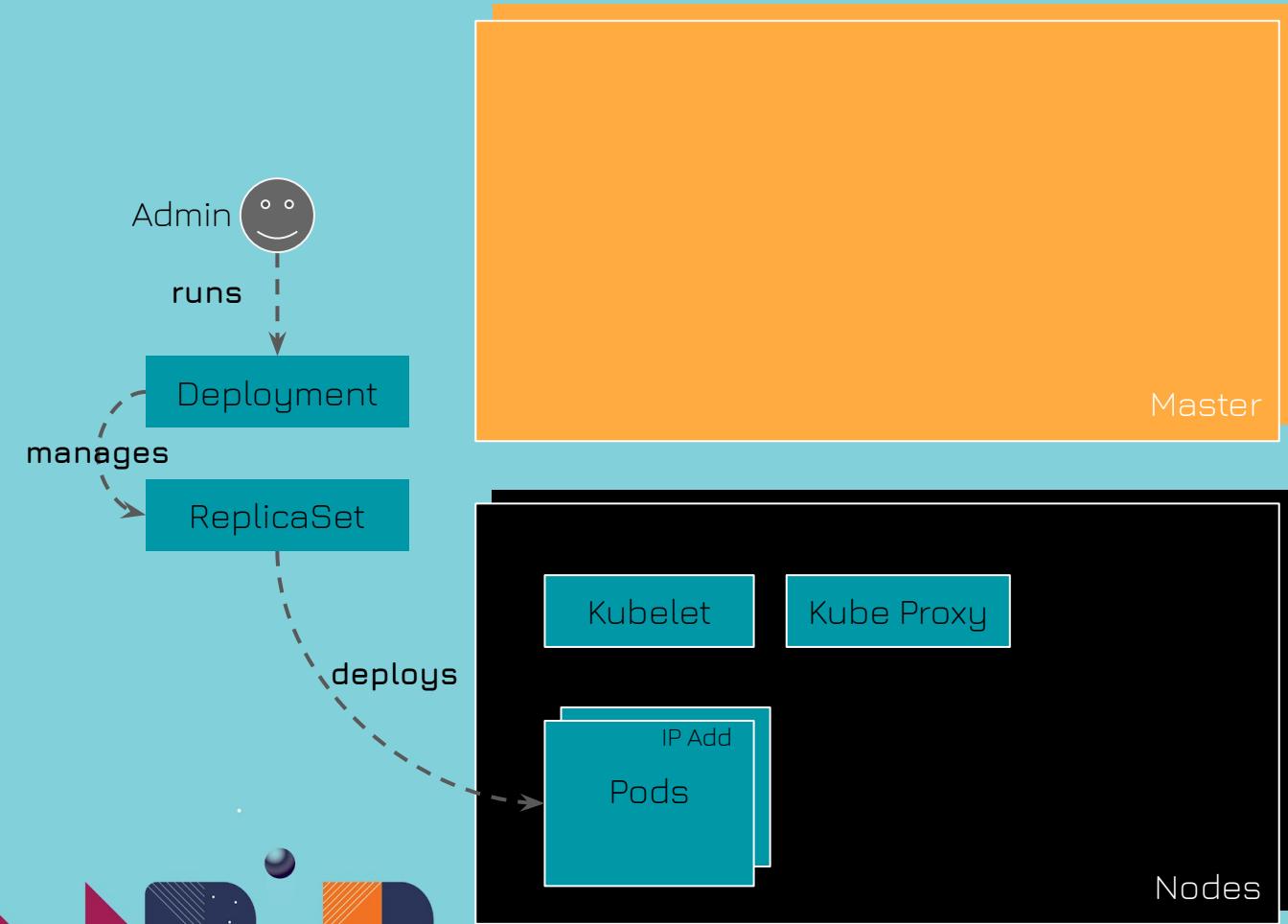
manages

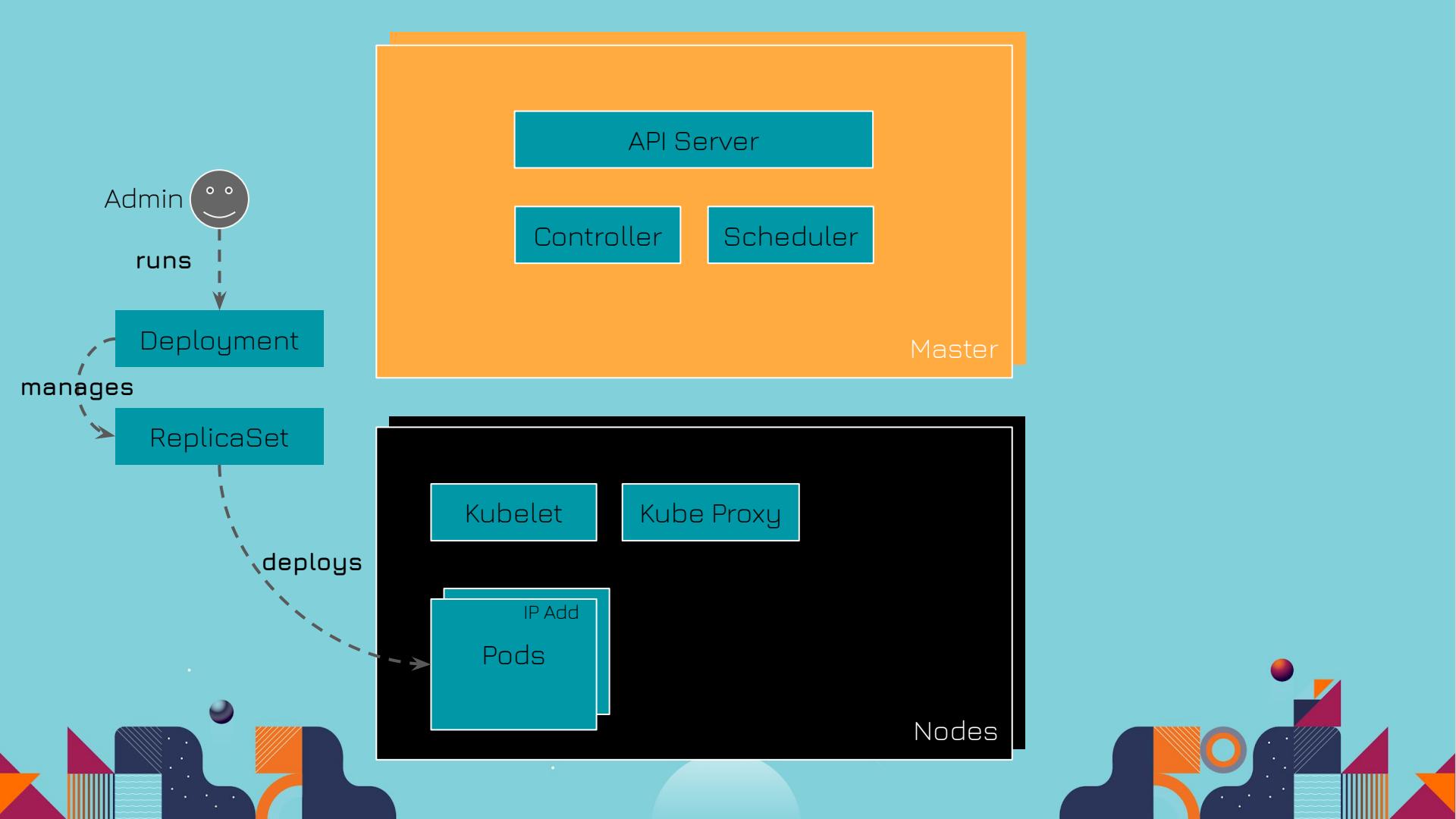


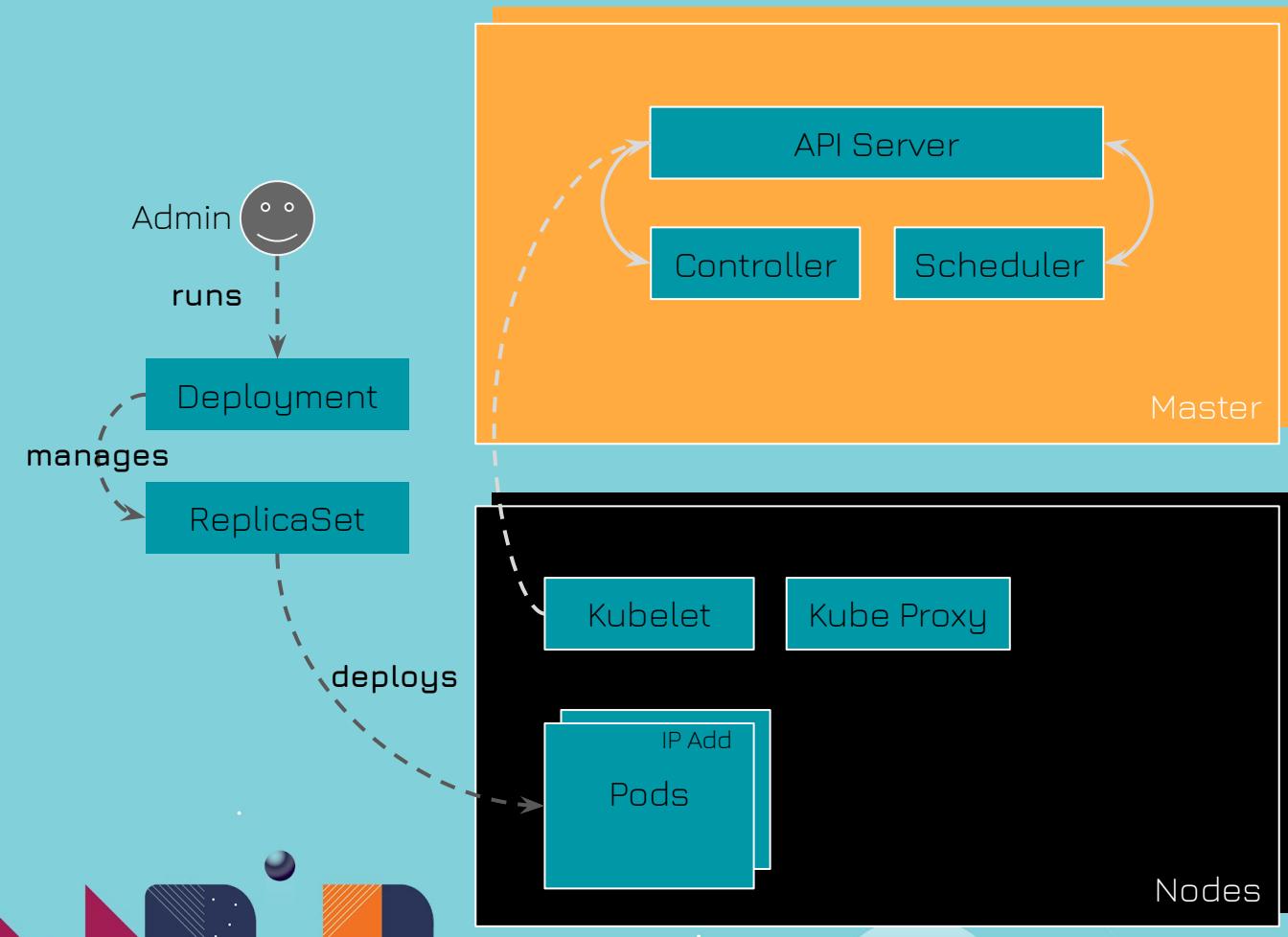


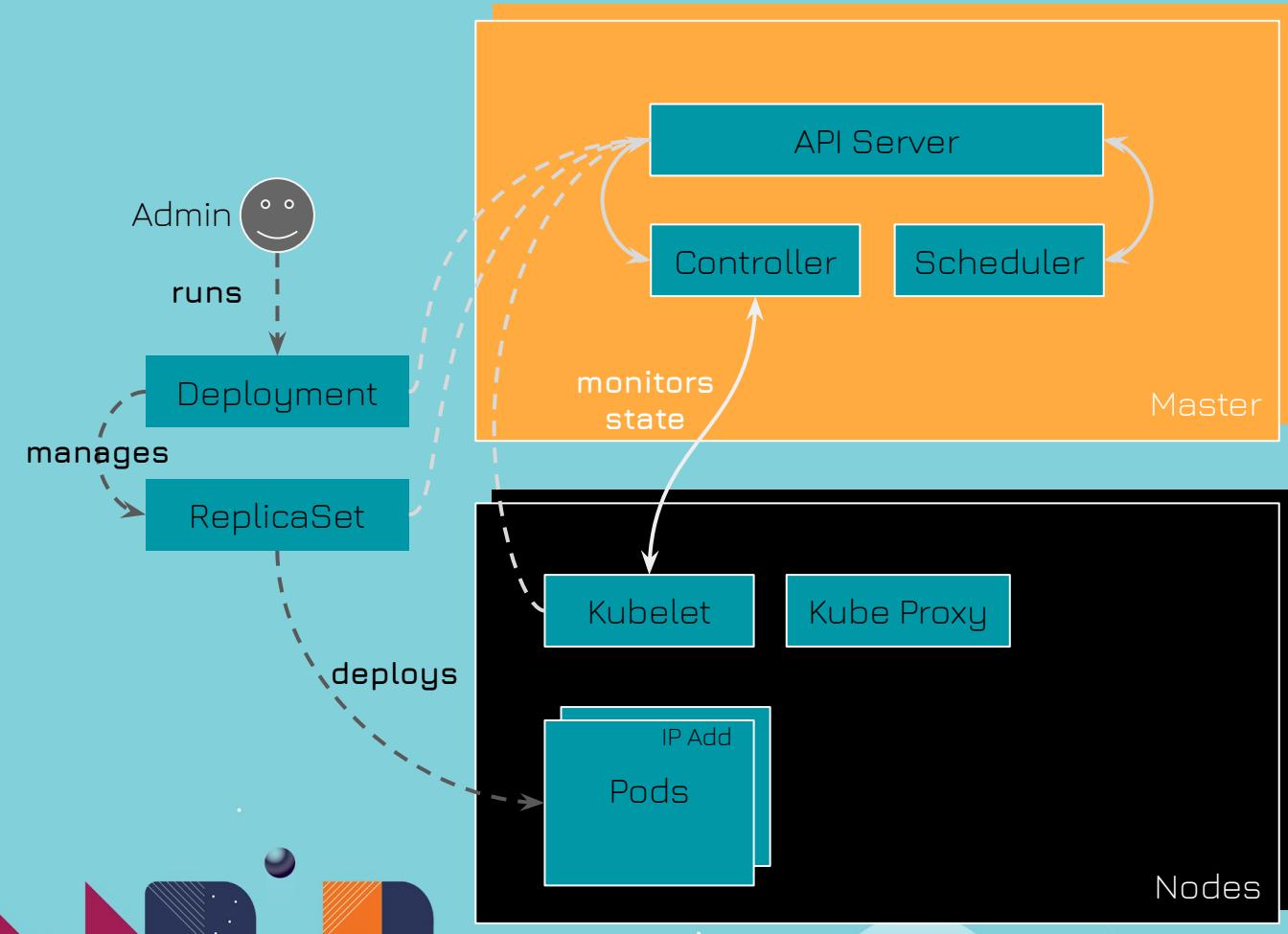


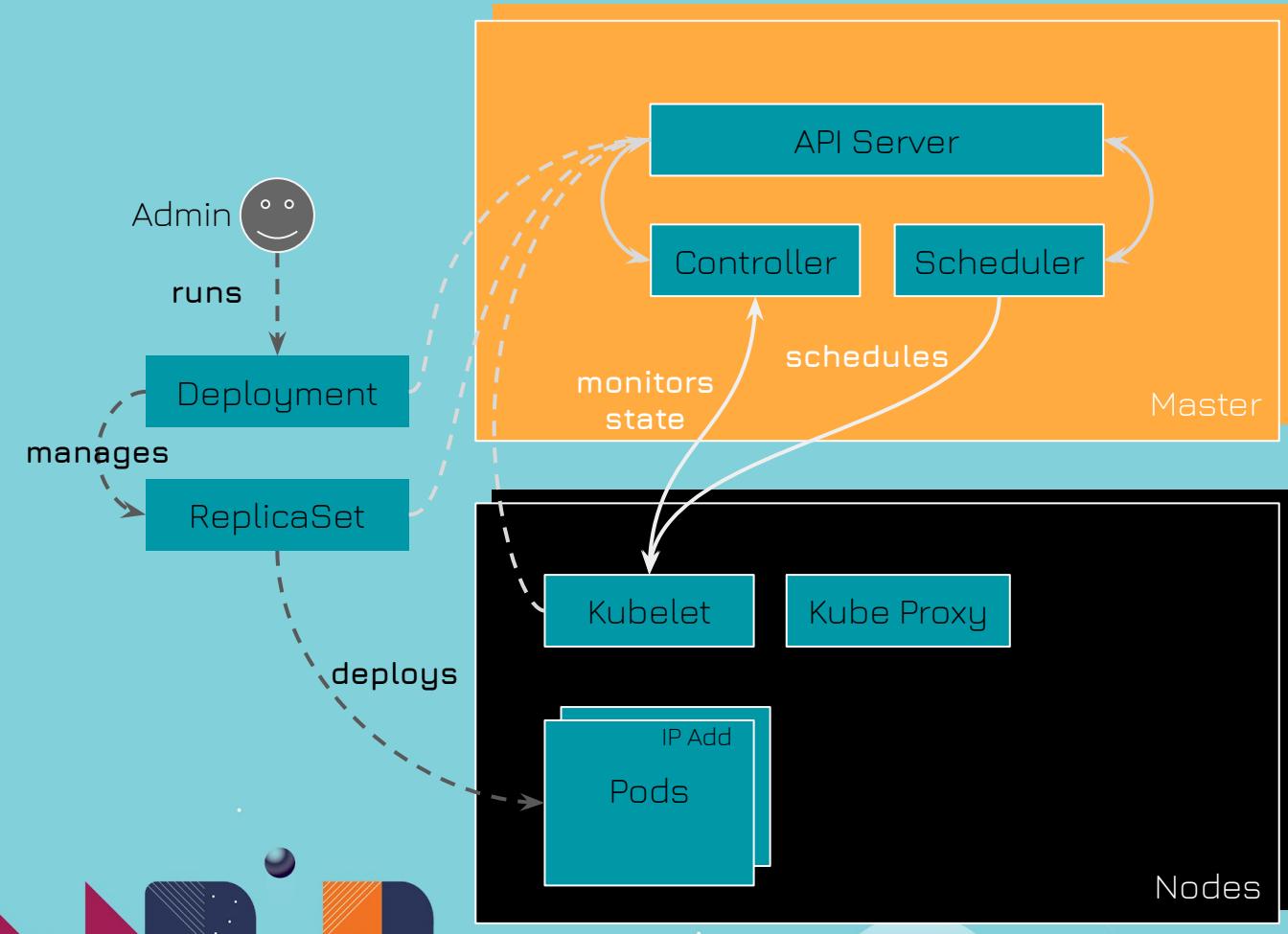


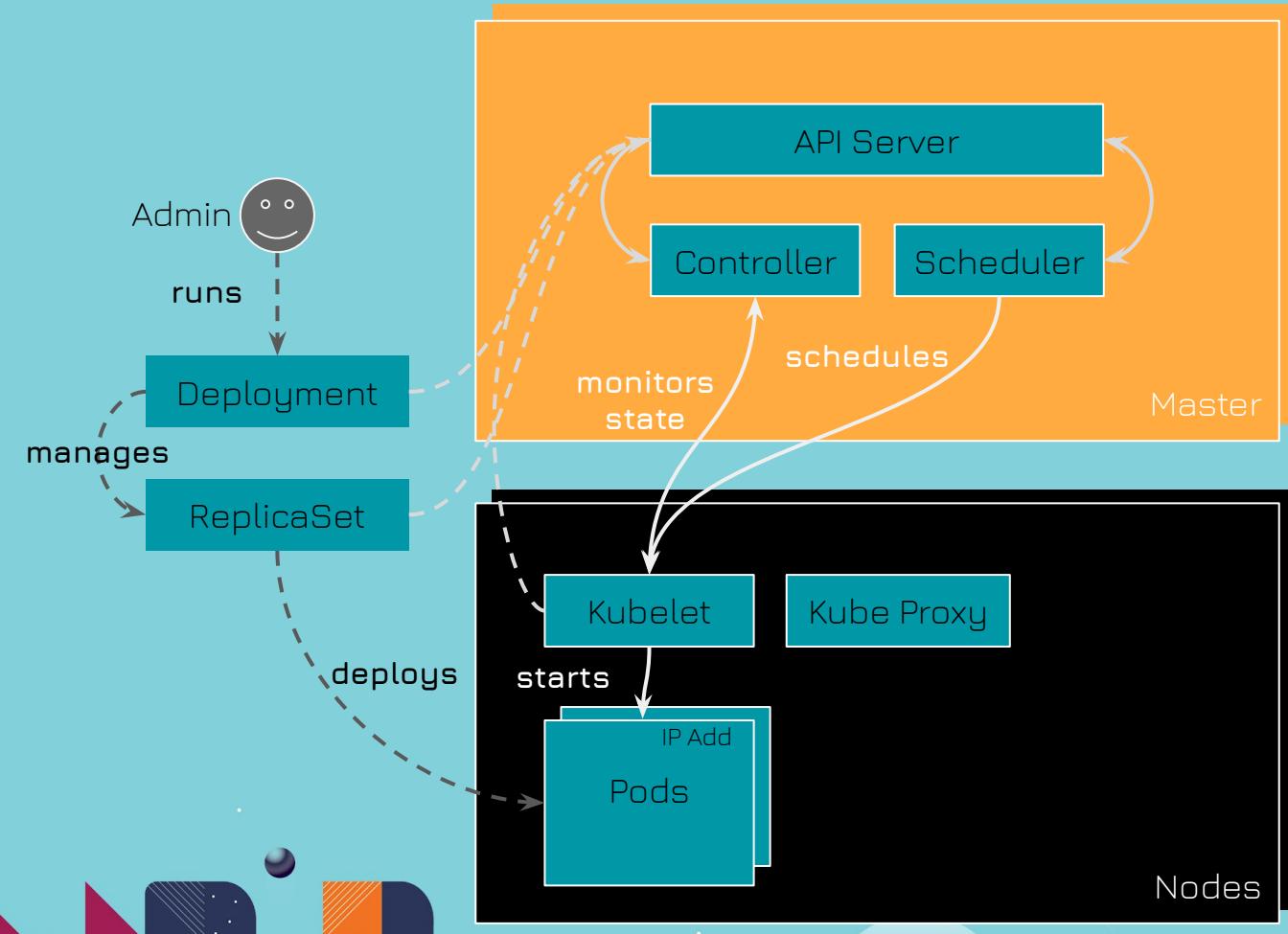


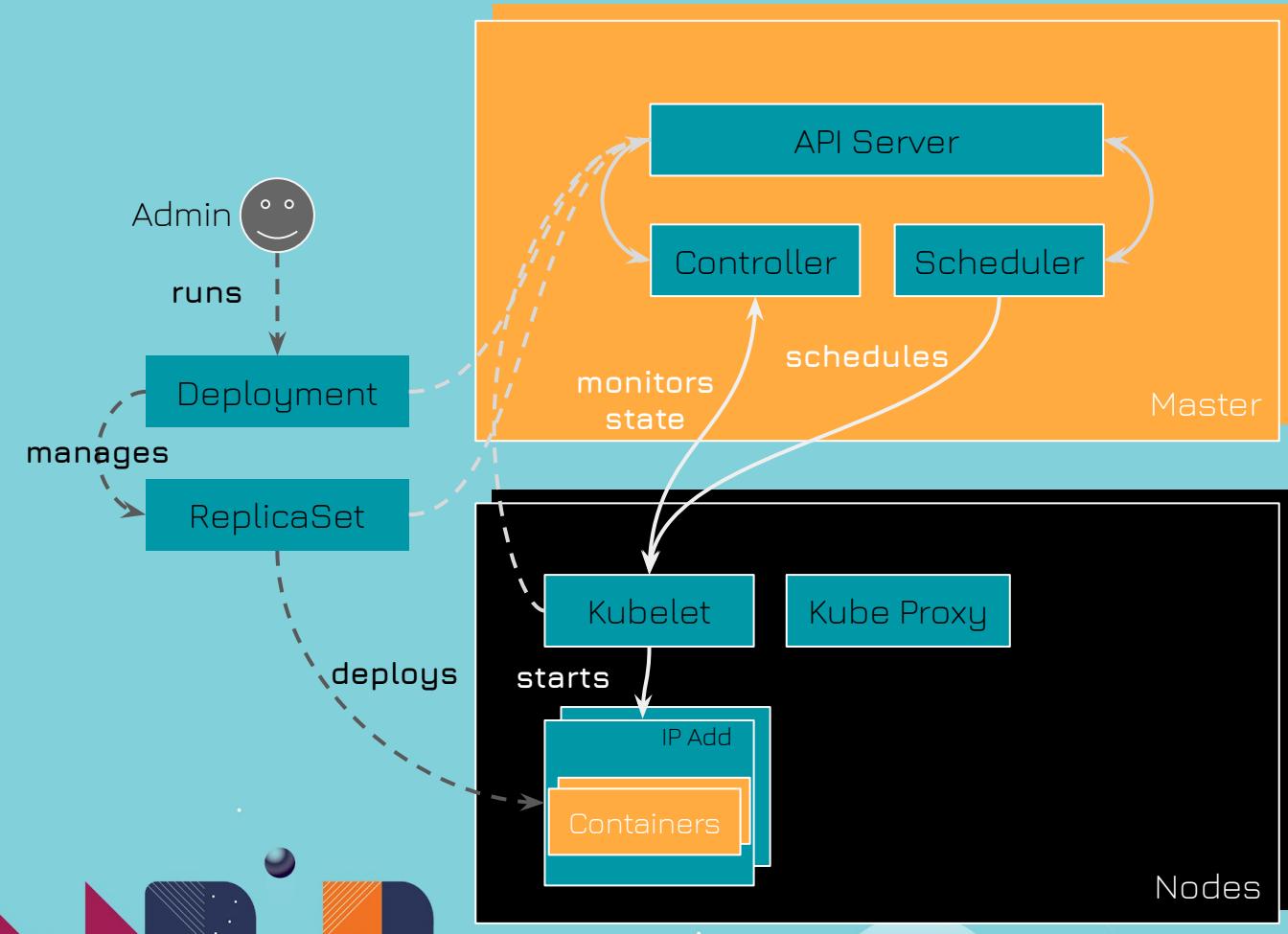


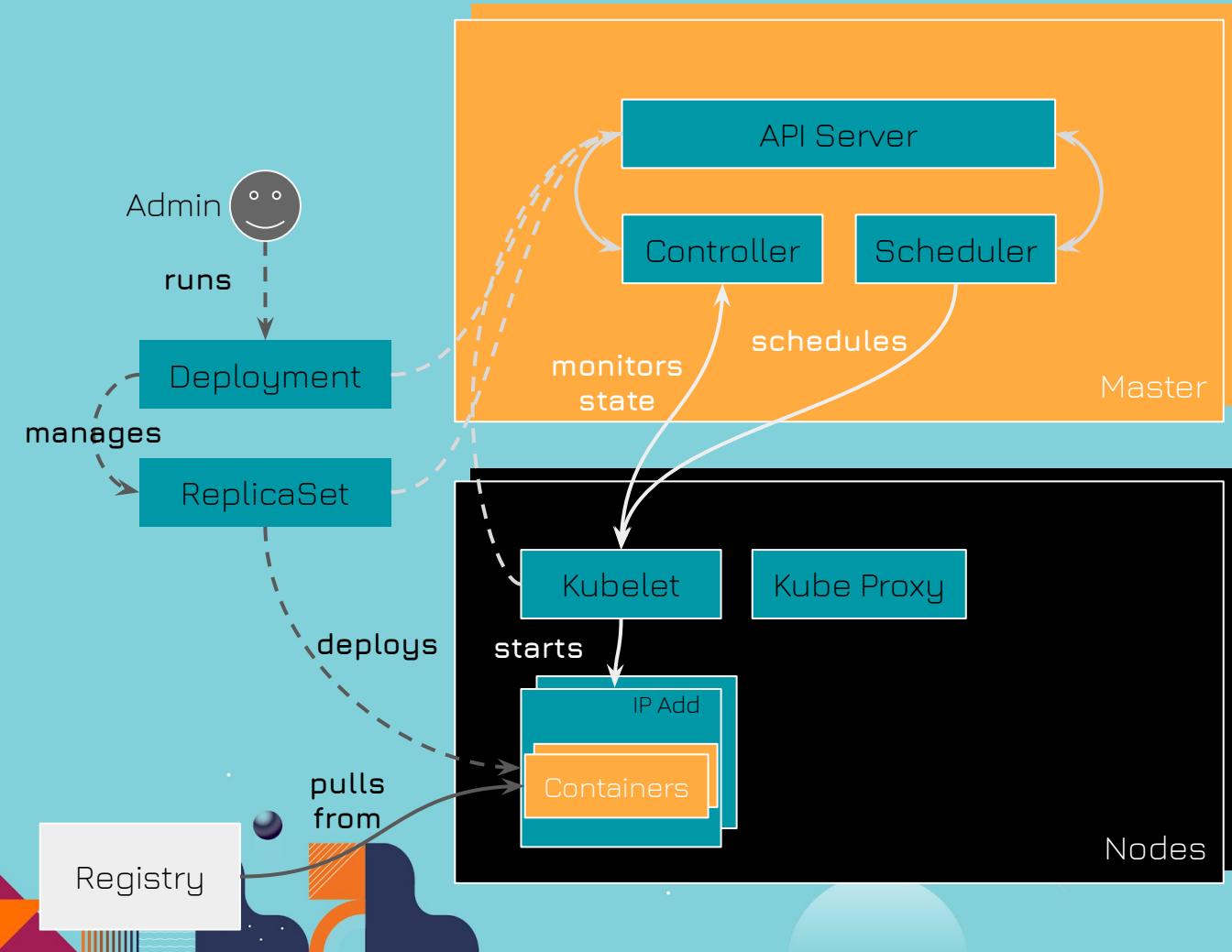


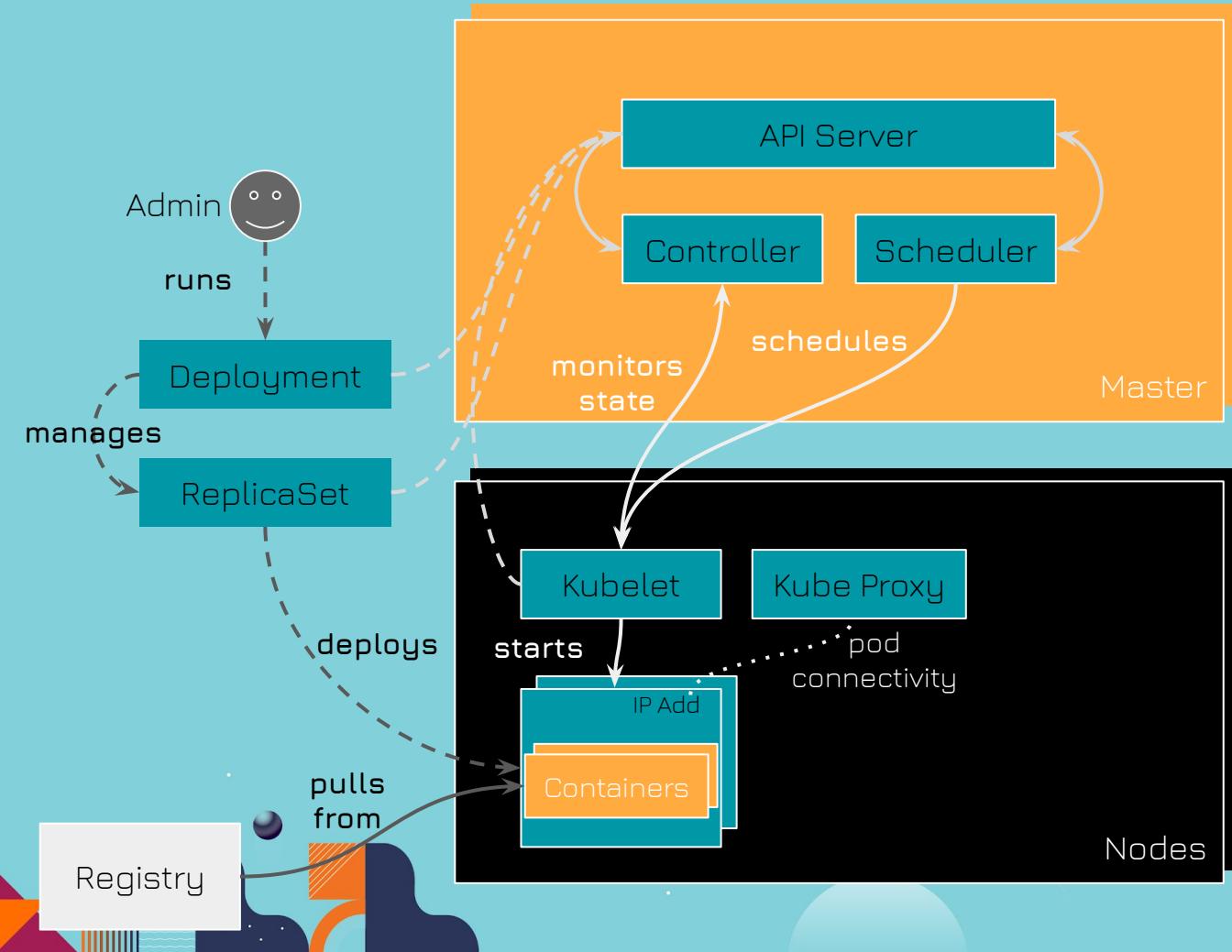


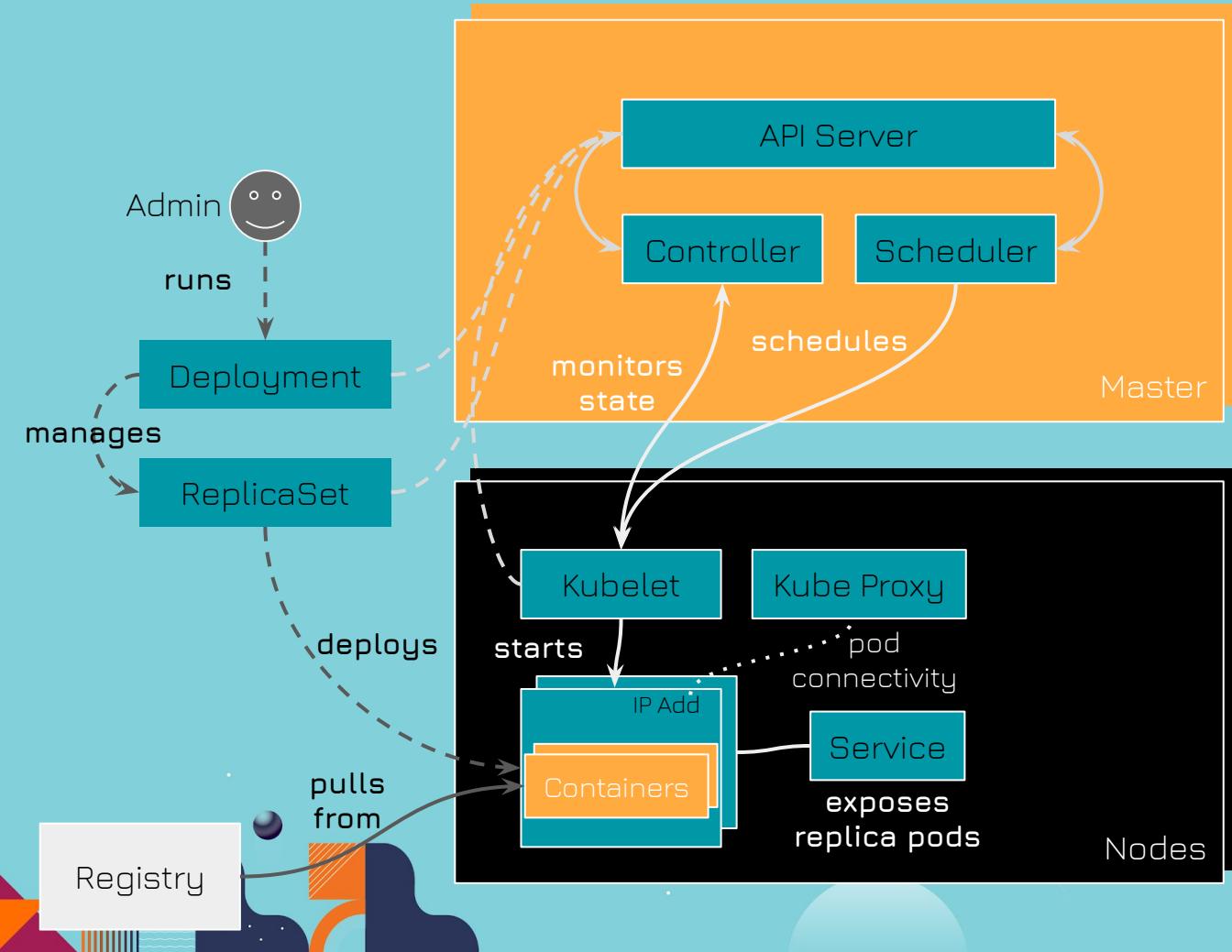


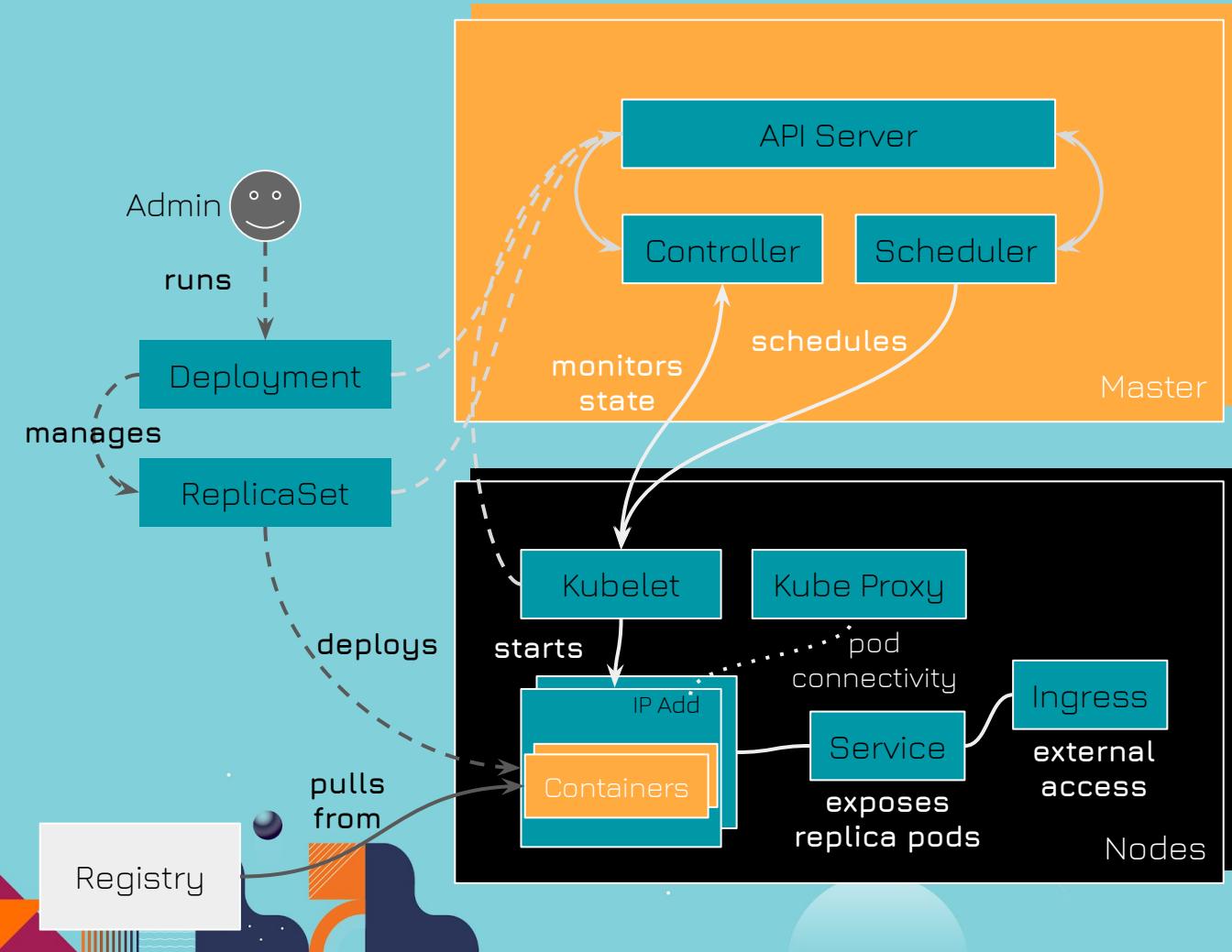


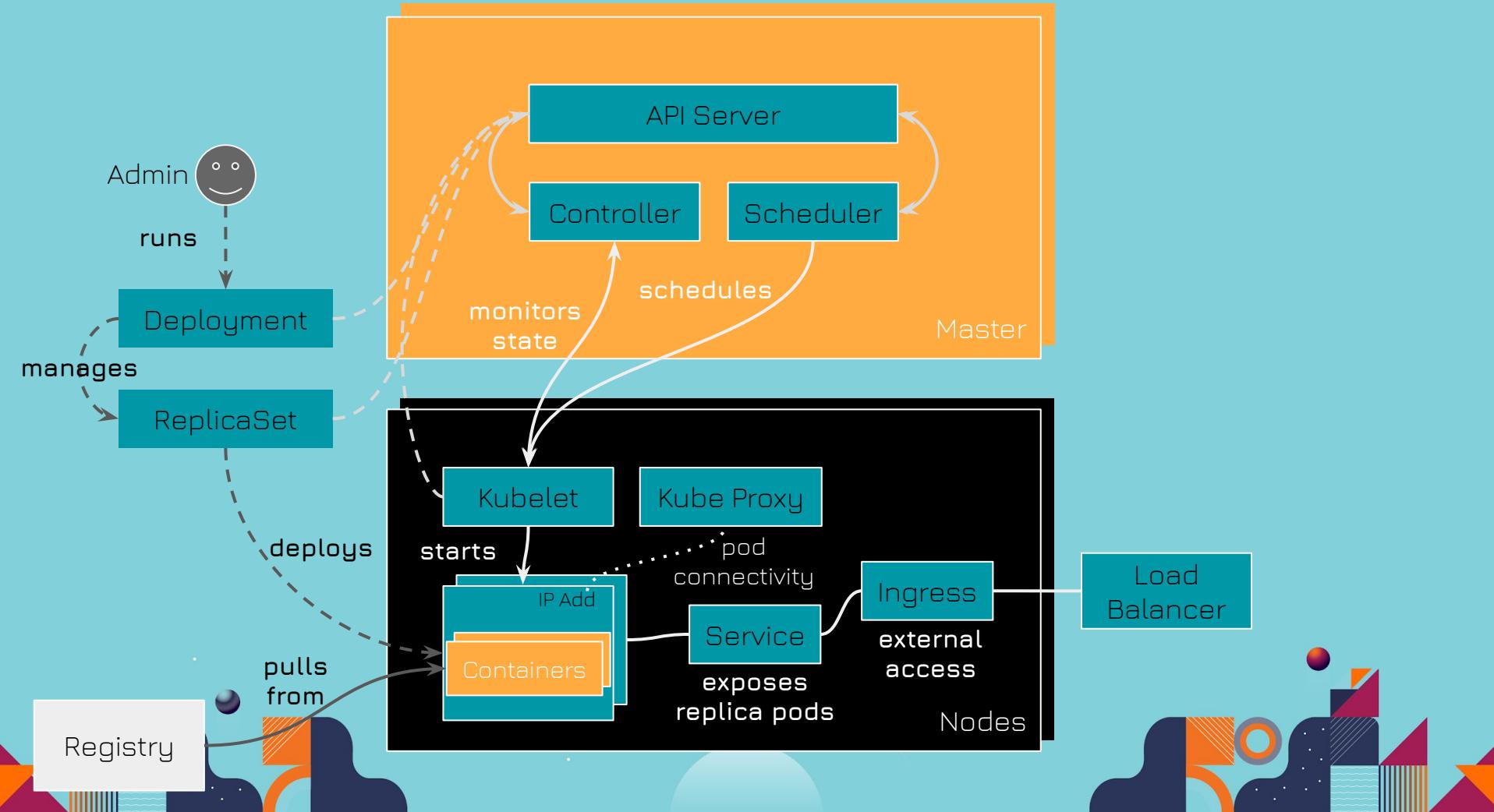


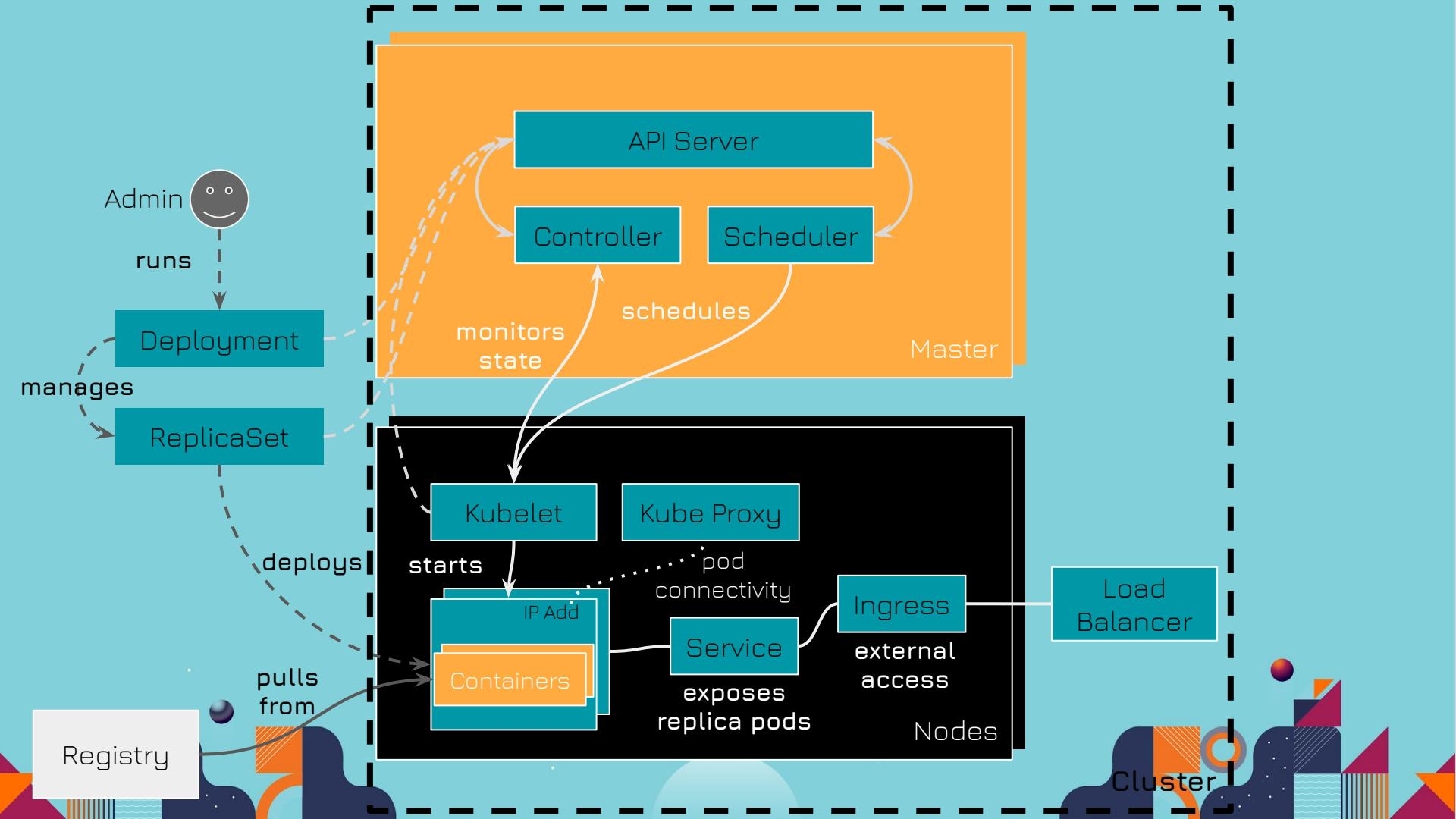


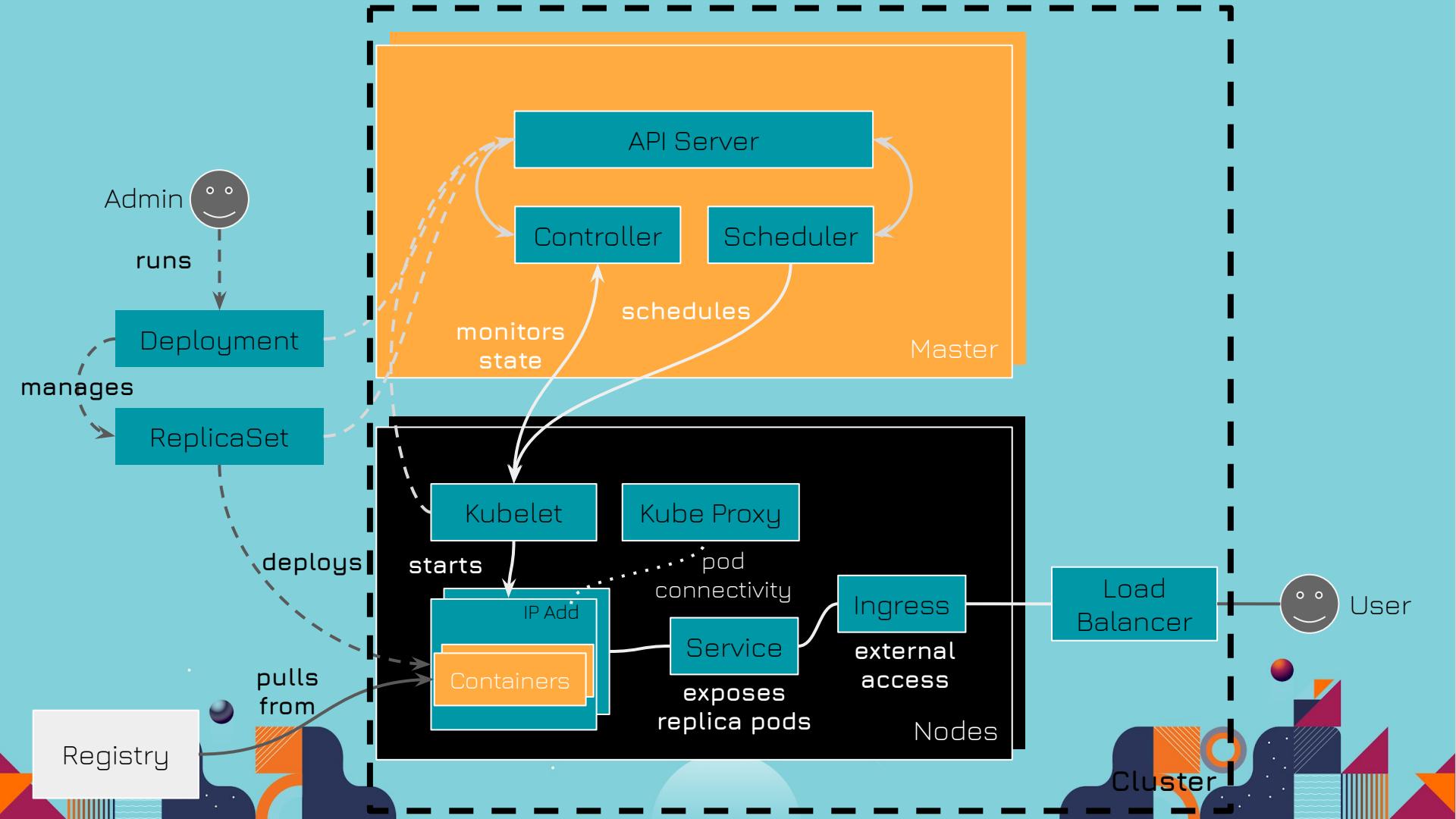












Lab 1: Build with Skaffold

Check on progress of setup script



Know before you go



Cloud Console

The screenshot shows the Google Cloud Platform Dashboard for project "qwiklabs-gcp-f55a5c11c7b193e". The dashboard includes sections for Project info, API usage, Resources, and Error Reporting.

Project info:

- Project name: qwiklabs-gcp-f55a5c11c7b193e
- Project ID: qwiklabs-gcp-f55a5c11c7b193e
- Project number: 395502397178

APIs & Services: Requests (requests/sec) over time (6 PM, 6:15, 6:30, 6:45). One entry: api/request_count.consumed_api:REDUCE_SUM(qwiklabs-gcp-f55a5c11c7b193e): 0.067

Resources: This project has no resources.

Trace: No trace data from the past 7 days.

Error Reporting: No sign of any errors. Have you set up Error Reporting?

Cloud Shell Code Editor (BETA)

The screenshot shows the Cloud Shell Code Editor interface. It features a file browser on the left and a code editor on the right.

File Browser:

- Root (8085-dot-440327-dot-5024-dot-devshell...):
 - backend-vs-can-to-can-prod-to-prod.yml
 - backend.yml
 - frontend-vs-50prod-50canary.yml
 - frontend-vs-all-to-canary.yml
 - frontend-vs-all-to-prod.yml
 - frontend.yml
 - myapp-vs-base.yml
 - namespaces.yml
 - seeding.yml
 - update-backend.sh
 - update-frontend.sh
 - README.md
 - spinnaker
- ...abs-gcp-f55a5c11c7b193e... (active tab):

```
googlece9955_student@cloudshell:~ (qwiklabs-gcp-f55a5c11c7b193e)$
```

Code Editor:

```
seedingsyml
1  kind: ConfigMap
2  metadata:
3    name: frontend-config
4    namespace: staging
5    labels:
6      app: frontend
7      moniker: spinnaker.io/application: "myapp"
8    data:
9      BACKEND_ENDPOINT: "http://backend.staging"
10     annotations:
11       appVersion: v1
12       kind: Deployment
13       metadata:
14         labels:
15           app: frontend-primary
16           namespace: staging
17           tier: staging
18           annotations:
19             app: frontend
20             version: production
21             tier: staging
22             spinnaker:
23               annotations:
24                 moniker: spinnaker.io/application: "myapp"
```

Don't hit END LAB unless you're done/sure!

Lab Running

END LAB

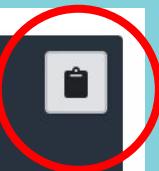
07:30:57

Always know your context!

```
~ (gke-spinnaker) $ kubectx  
gke-central  
gke-spinnaker  
gke-west  
~ (gke-spinnaker) $ █
```

Use copy/paste as much as possible

```
cd ~  
curl -L https://git.io/getLatestIstio | sh -  
cd istio-0.8.0
```



Variables

```
~ (gke-spinnaker) $ source ~/advanced-kubernetes-workshop/setup/env.sh
Your active configuration is: [cloudshell-11479]
~ (gke-spinnaker) $ █
```

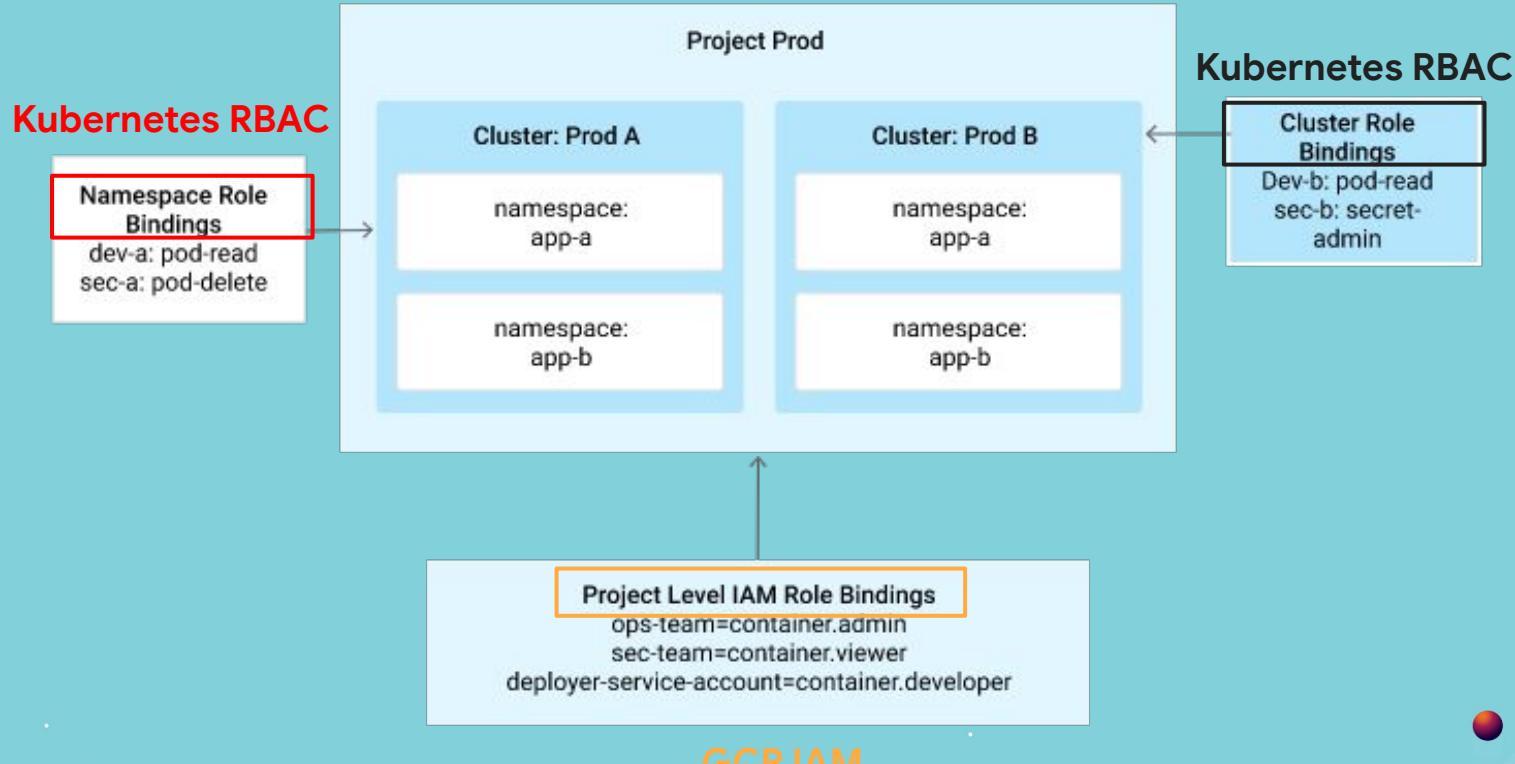
Workshop Helper Commands

```
~ (gke-spinnaker) $ workshop_
workshop_connect          workshop_fortio           workshop_hal-config
workshop_create-pipelines   workshop_get-ingress   workshop_terraform-apply
~ (gke-spinnaker) $ workshop_█
```

Helm

Package manager for Kubernetes

Client & Tiller



Stop after lab 1 and wait for further instructions

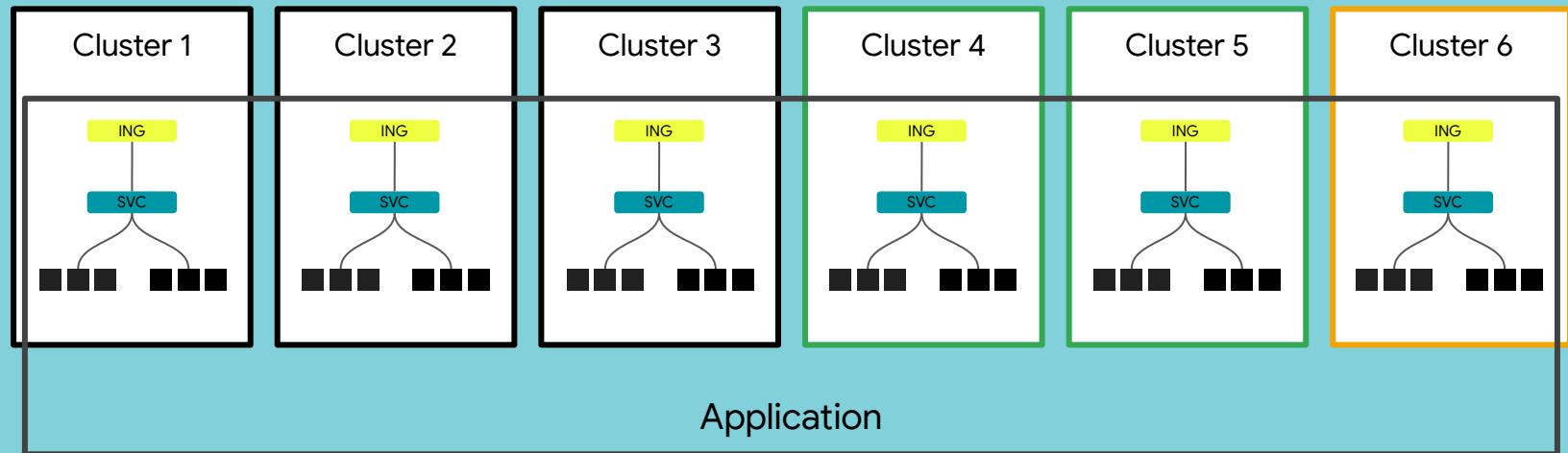
04

Continuous Delivery with Spinnaker

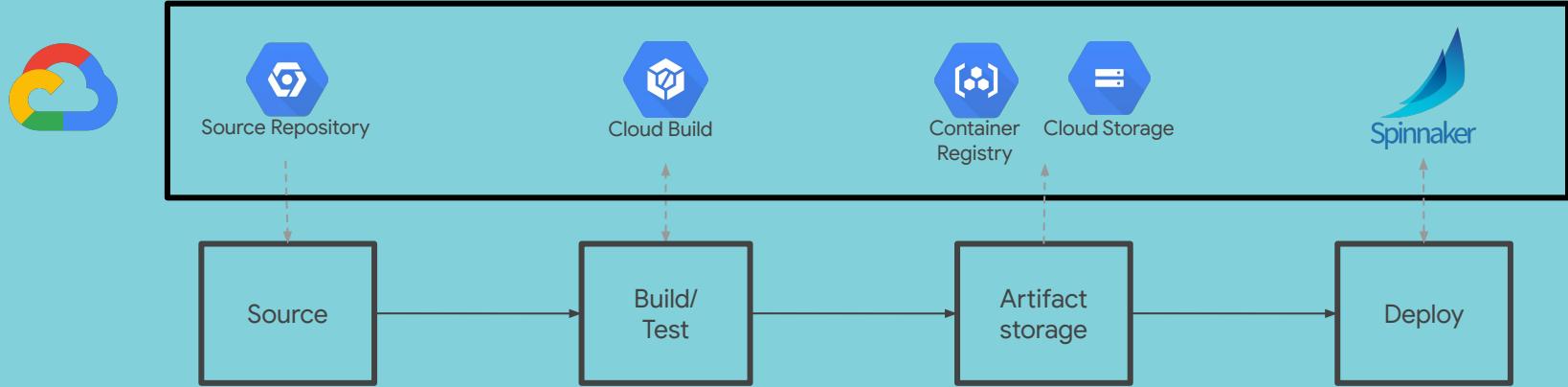
An Application in Kubernetes



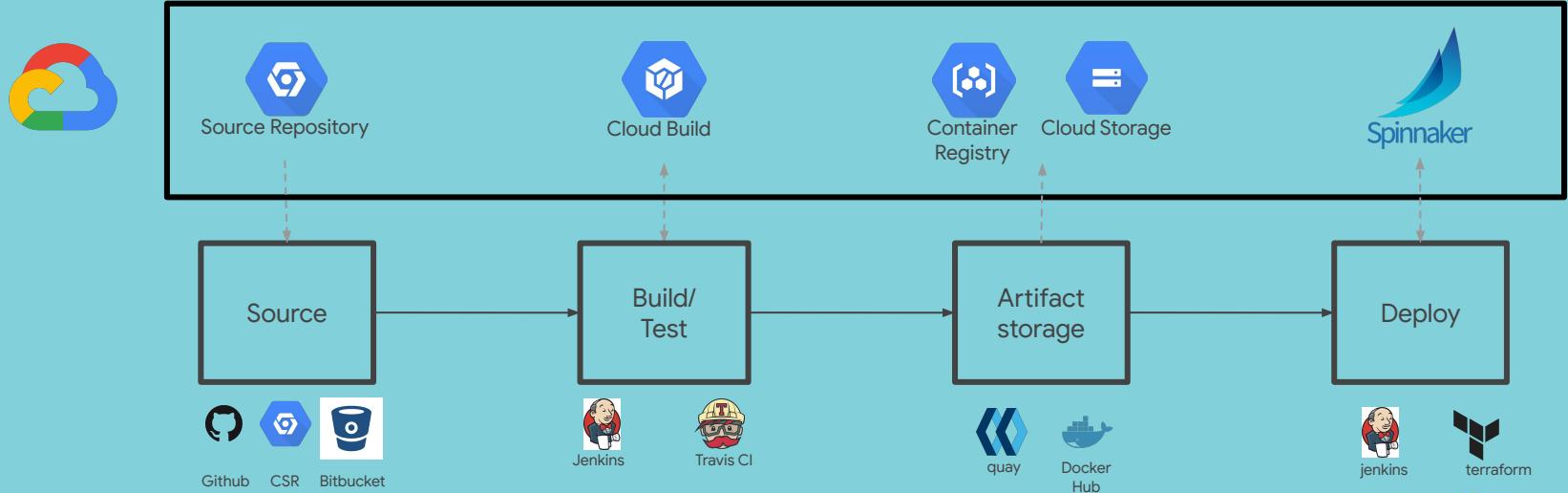
An Application in Kubernetes



CI/CD on Google Cloud



CI/CD on Google Cloud



Spinnaker

Spinnaker is an open-source, multi-cloud, ***continuous delivery*** platform

Application management
Workload management



2.5k

applications
deploying with
Spinnaker

9.5k

pipelines



4k

deployments per
day

1

Spinnaker
instance

Application Centric Management

Clusters
Namespaces
Services
Load Balancers

The screenshot shows the Spinnaker interface for managing the **mdservice** application. The top navigation bar includes links for SPINNAKER, Search, Projects, Applications, and a user dropdown for stevenkim@spinnaker-test.net. The main tabs are PIPELINES, CLUSTERS, and TASKS, with CLUSTERS selected.

The left sidebar contains search and filter options for ACCOUNT (my-kubernetes-account), REGION (default), STACK ((none), prod, stage), STATUS (Healthy, Unhealthy, Disabled, Starting, Out of Service, Unknown), AVAILABILITY ZONES (default), INSTANCE TYPES, and INSTANCE COUNT (Min: [] Max: []).

The central area displays two clusters under the **MY-KUBERNETES-ACCOUNT** section:

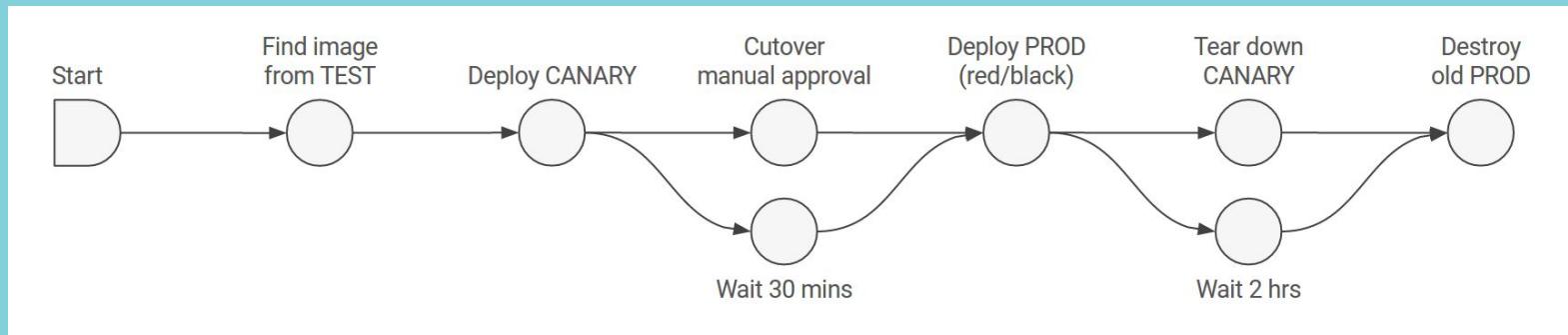
- mdservice-prod**: 128 instances at 100% health. A horizontal bar chart shows 128 green segments, each representing an instance.
- mdservice-stage**: 2 instances at 100% health. A horizontal bar chart shows 2 green segments.

A detailed view for the **mdservice-prod** cluster is shown on the right, titled **mdservice-prod-v059**. It includes sections for Server Group Actions (Rollback, Resize, Disable, Destroy, Clone), INFORMATION (Last Update: 6/14/24 18 PDT, Status: ACTIVE), HEALTH (Pods: 128 at 100%), DEPLOYMENT (No deployment found), REPLICAS (Min: 128, Max: 128, Current: 128), and HORIZONTAL POD AUTOSCALING (Desired: 128, Target CPU: 40%, Current CPU: 0%, Latest Rescale: a minute ago). A "Kube UI" link is also present.

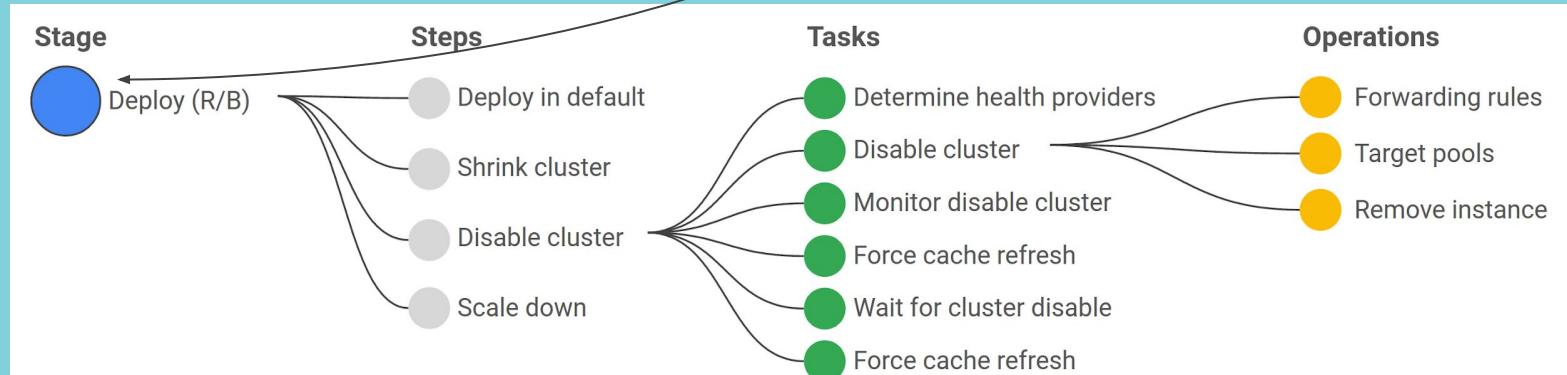
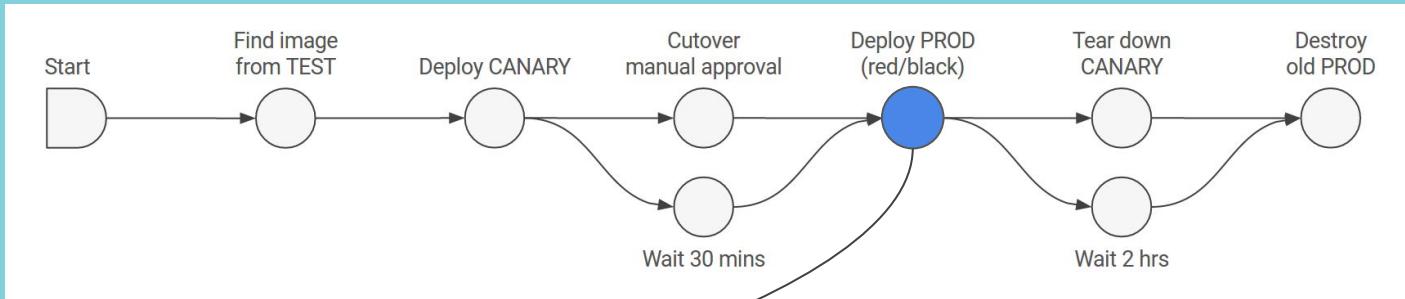
Deployment Management

Pipelines

Stages



Abstracts and Orchestrates APIs



Deployment Strategies



Kayenta: Automated Canary Deployments

The image shows two side-by-side screenshots of the Kayenta web interface.

Left Screenshot: A "Run Kayenta Canary" button is at the top. Below it is a "STAGE DETAILS: RUN KAYENTA CANARY" section with a duration of 00:06. A table shows one step: "Run Kayenta Canary" started at 2018-03-23 13:54:41 PDT, duration 00:06, and status TERMINAL.

Right Screenshot: An "RUN KAYENTA CANARY" section with a red warning icon. It shows a "Canary Summary" card with a score of 85. The card includes "Canary Result", "Duration", and "Last Updated". A message box states: "Exceptions (Aggregate Canary Results) Final canary score 85.0 is below the pass score threshold."

Safe Deployments

Execution Windows

Restrict execution to specific time windows

Days of the Week (No days selected implies execution on any day if triggered)

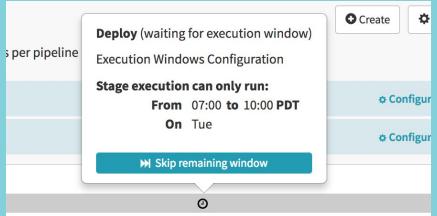
Sun	Mon	Tue	Wed	Thu	Fri	Sat	All	None	Weekdays
-----	-----	-----	-----	-----	-----	-----	-----	------	----------

Time of Day

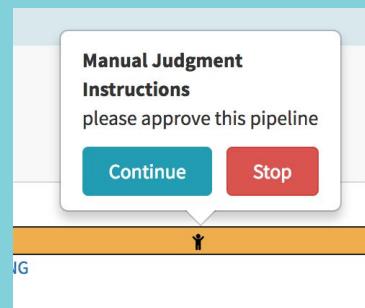
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

This stage will only run within the following windows (all times in PDT):

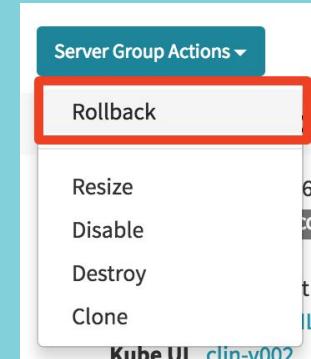
From 07 : 00 to 10 : 00 PDT



Manual Judgements

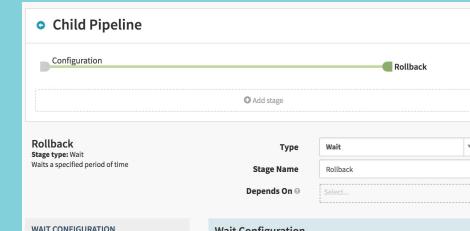


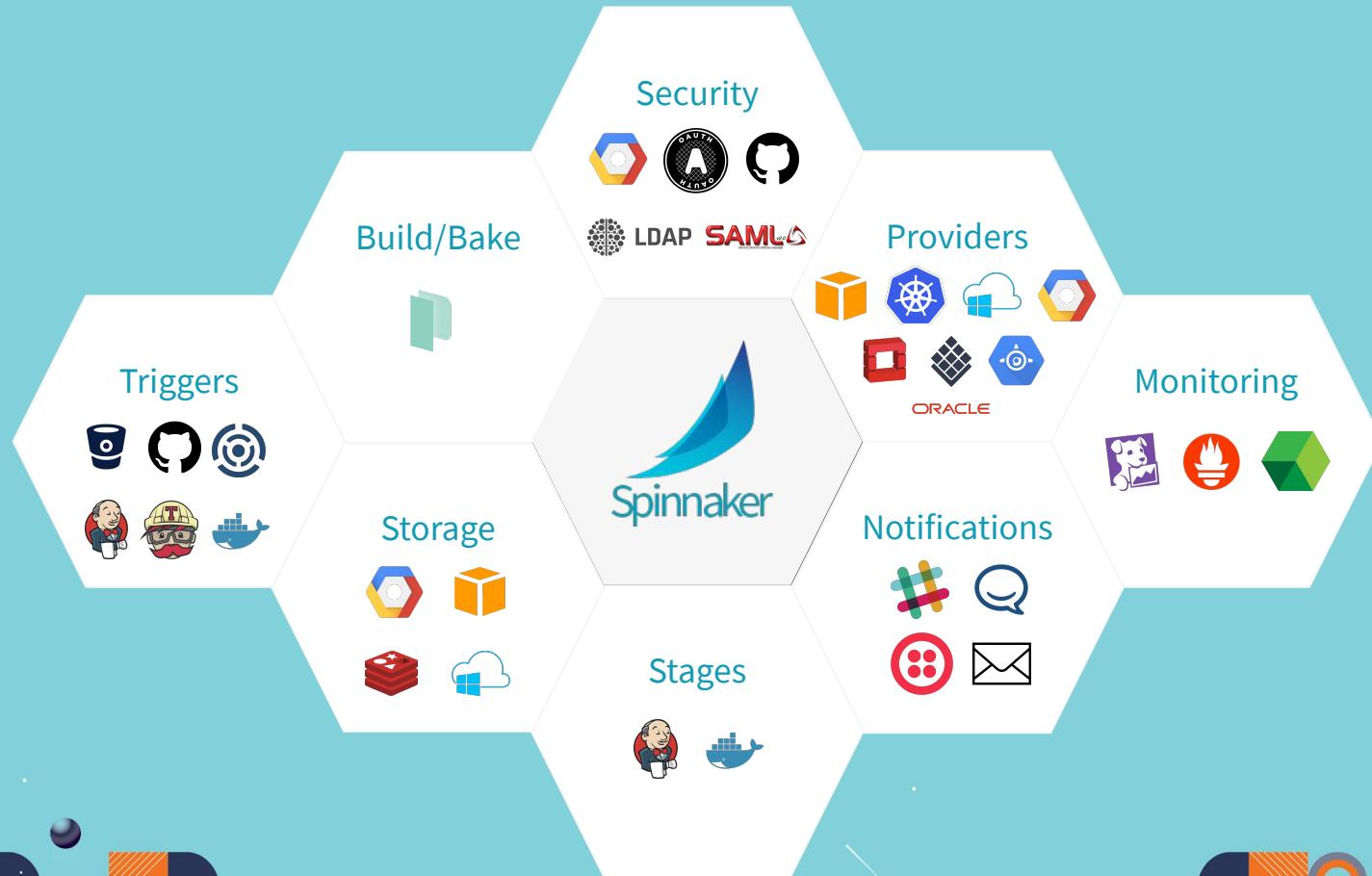
Manual Rollbacks



Automated Rollbacks

Trigger a pipeline that does a rollback on a failed deployment





Lab 2: Deploy with Spinnaker

05

Istio & Service Mesh

Everybody got all fired up about kubernetes and microservices and then were like ‘Oh, s--t, what’s going on?’ Istio gives us a view of our entire system and lets us find trouble spots.

– An early adopter who will remain nameless

Service Mesh

Transparently automate application network functions.



Separating applications from network functions



Secure, Monitor, Manage

Intelligent Routing

- Dynamic route configuration
- A/B tests
- Canaries
- Gradually upgrade versions

Resilience

- Timeouts
- Retries
- Health checks
- Circuit breakers

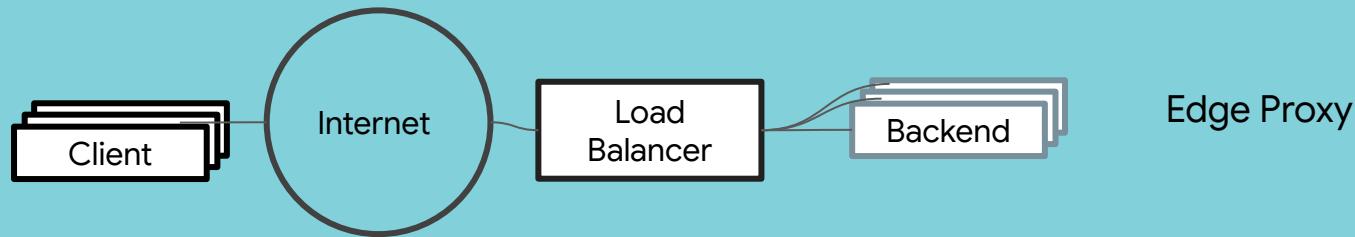
Security & Policy

- Mutual TLS
- Organizational policy
- Access policies
- Rate Limiting

Telemetry

- Service Dependencies
- Traffic Flow
- Distributed Tracing





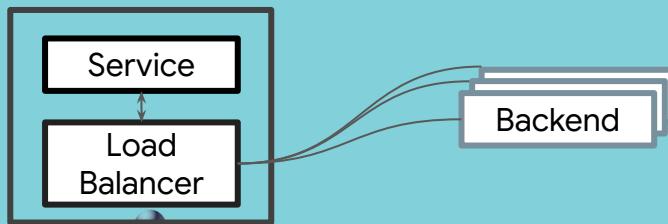
Edge Proxy



Middle Proxy



Embedded Client Library



Sidecar Proxy

Source :

<https://blog.envoyproxy.io/introduction-to-modern-network-load-balancing-and-proxying-a57f6ff80236>

Envoy

Lightweight L4/L7 Proxy

Resilient and Scalable

HTTP/2 and gRPC

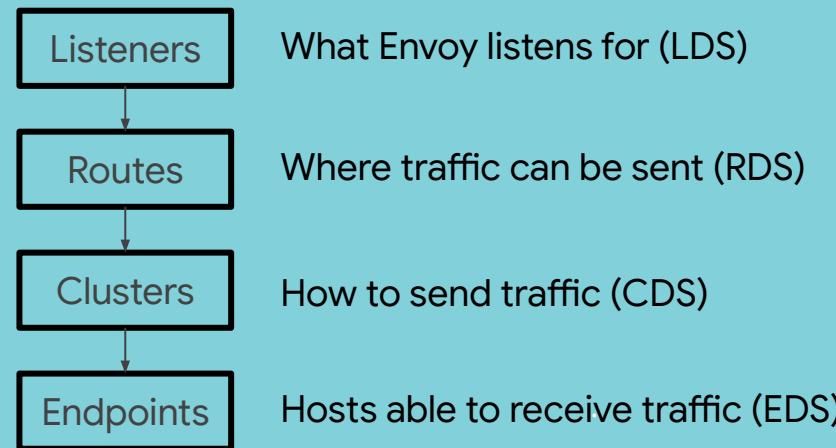
Health checks, circuit breakers, timeouts, retry budgets

No hot reloads - API driven config updates



Envoy Configuration

Configured with **Aggregated Discovery Services, ADS**



"You now work on 'Istio'"



Istio

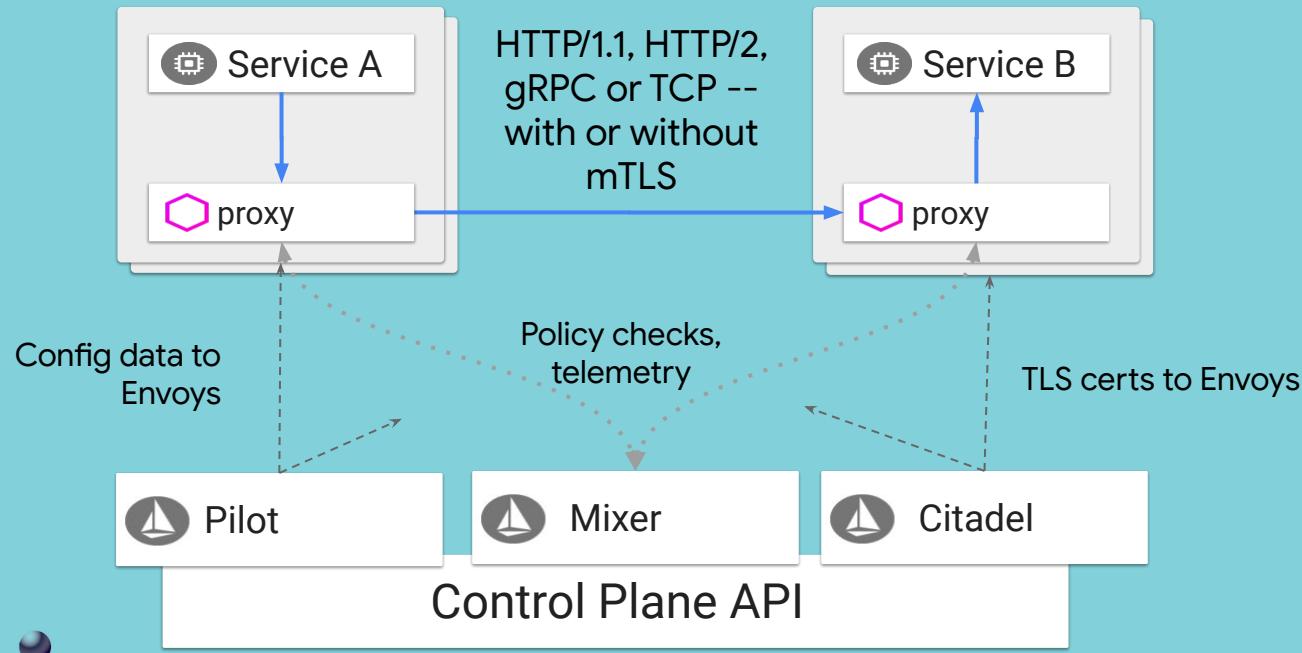
Security

Observability

Traffic Management

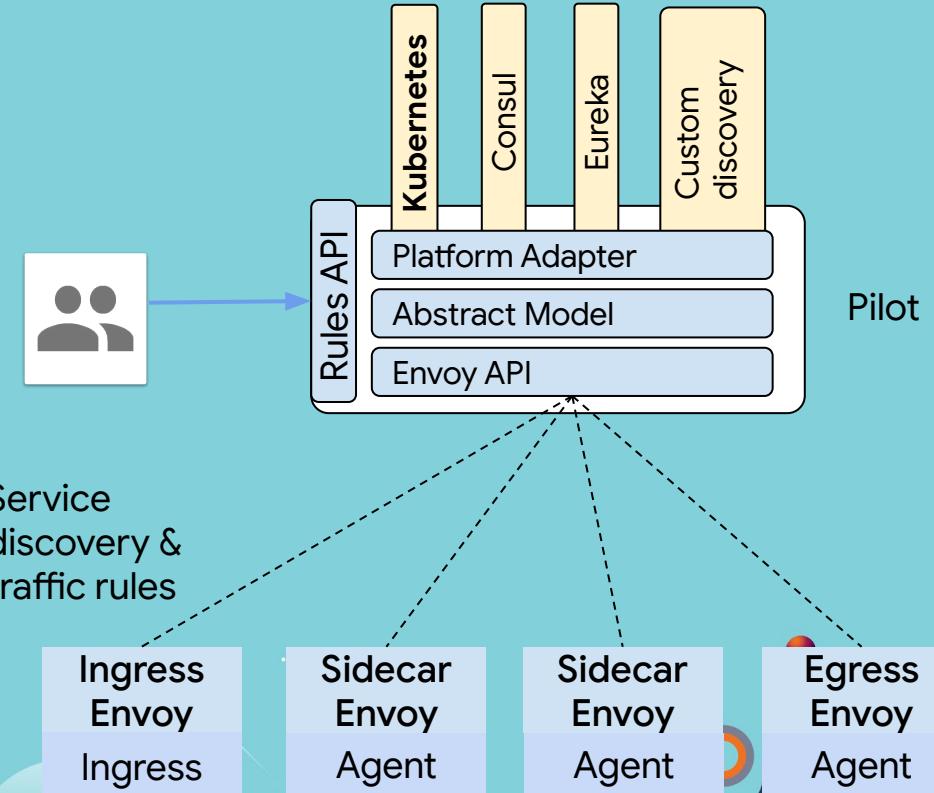


Istio Architecture



Pilot configures the data plane

- Service Topology
- Routing Rules
- Sidecar / Envoy Configuration

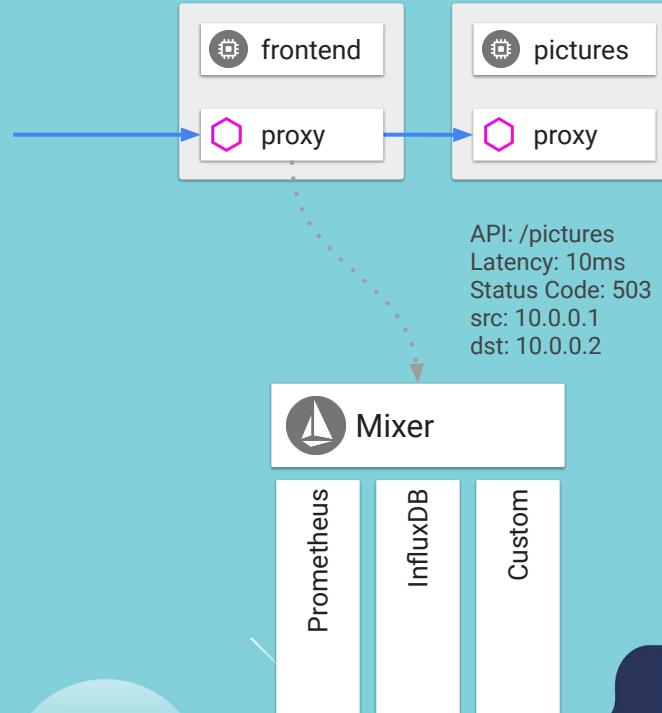


Mixer provides policy and telemetry

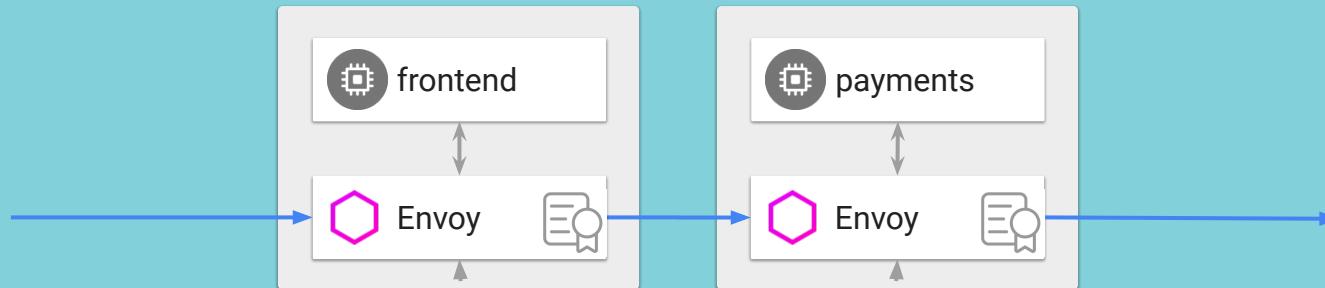
Policy / Quota Enforcement

Telemetry

Extensible via Adapters



Citadel provides identity and encryption



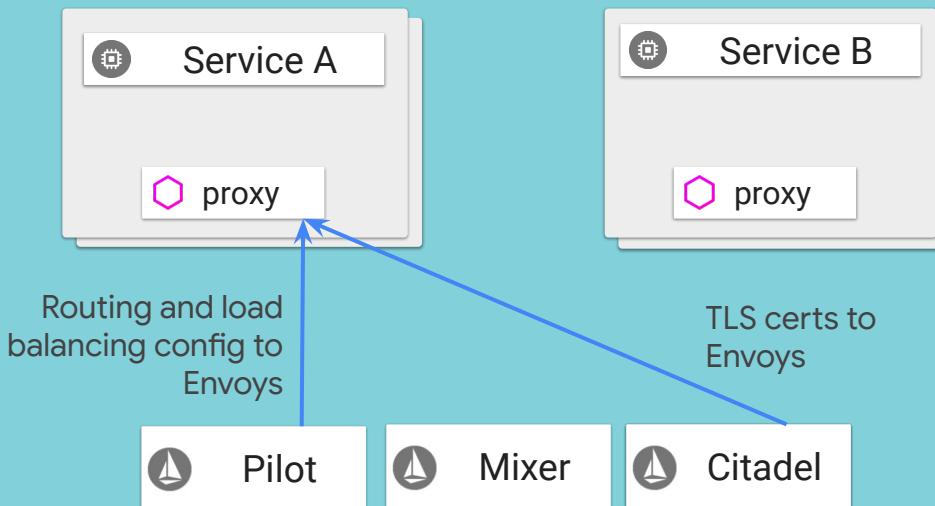
SAN:
"spiffe://example.local/ns/prod/sa/foo"
Namespace: **prod**
Service account: **foo**

SAN:
"spiffe://example.local/ns/prod/sa/bar"
Namespace: **prod**
Service account: **bar**



Citadel

Life of a request in the mesh

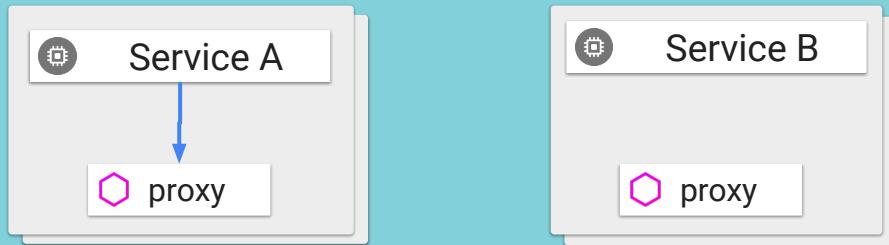


Service A comes up.

Envoy is deployed with it and fetches service information, routing and configuration policy from Pilot.

If Istio Auth is being used, TLS certs are securely distributed as well.

Life of a request in the mesh



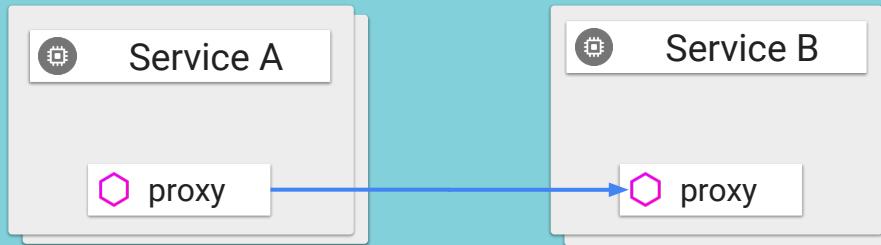
Service A places a call to service B.

Client-side Envoy intercepts the call.

Envoy consults config to know how/where to route call to service B.



Life of a request in the mesh



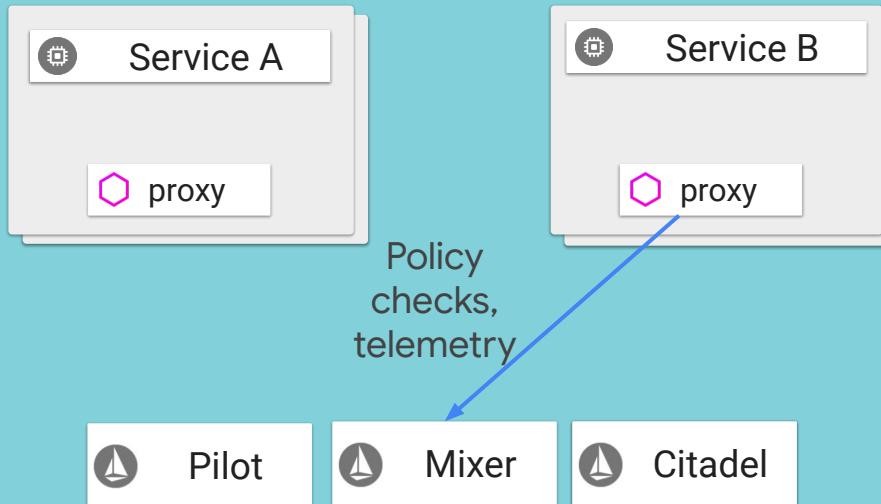
HTTP/1.1, HTTP/2, gRPC or TCP -- with or without mTLS

Envoy forwards request to appropriate instance of service B.

There, the Envoy proxy deployed with the service intercepts the call.

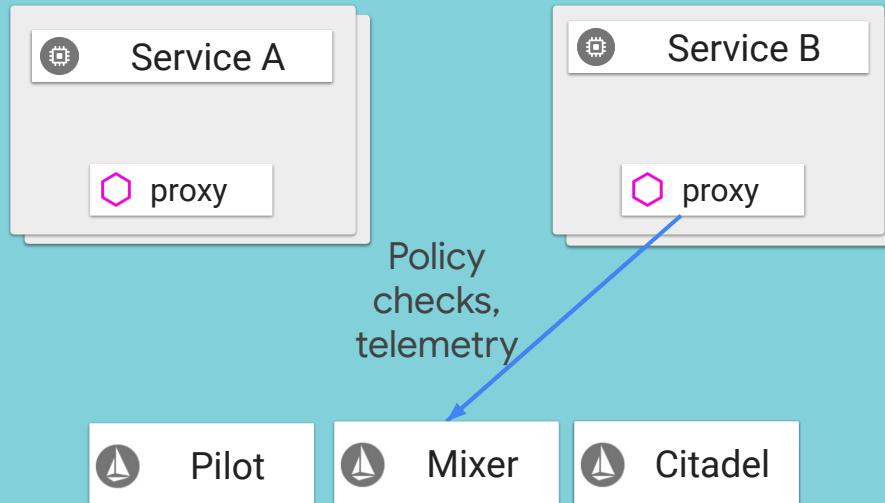


Life of a request in the mesh



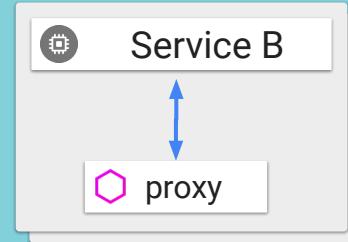
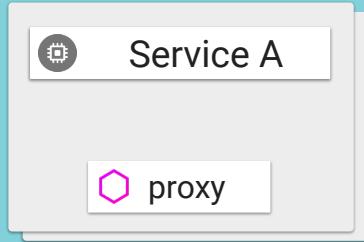
Server-side Envoy checks with Mixer to validate that call should be allowed (ACL check, quota check, etc).

Life of a request in the mesh

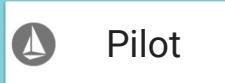


Mixer checks with appropriate adaptors (policy engine, quota adaptor) to verify that the call can proceed and returns True/False to Envoy

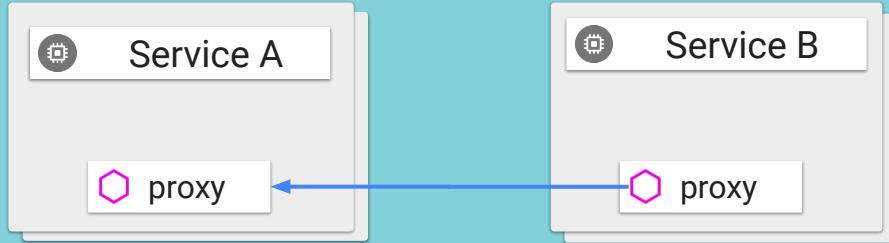
Life of a request in the mesh



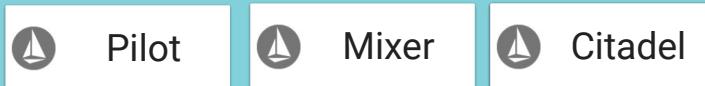
Server-side Envoy forwards request to service B, which processes request and returns response



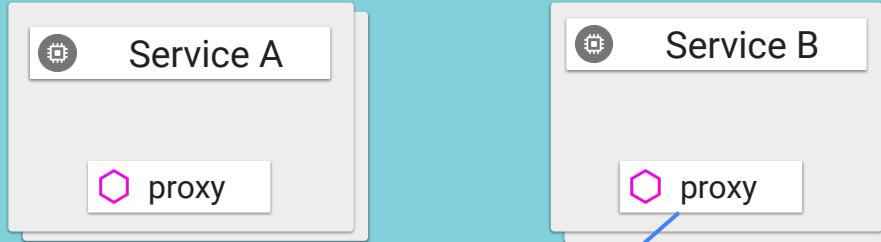
Life of a request in the mesh



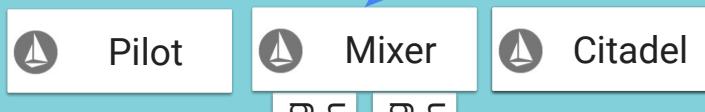
Envoy forwards response to the original caller, where response is intercepted by Envoy on the caller side.



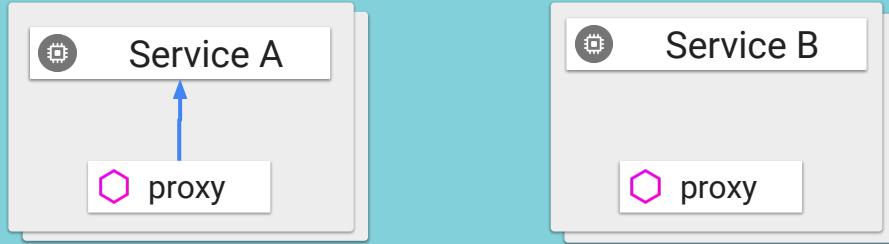
Life of a request in the mesh



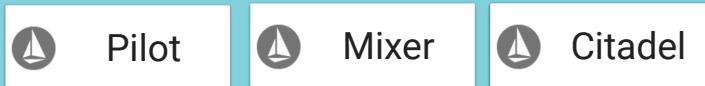
Envoy reports telemetry to Mixer, which in turn notifies appropriate plugins



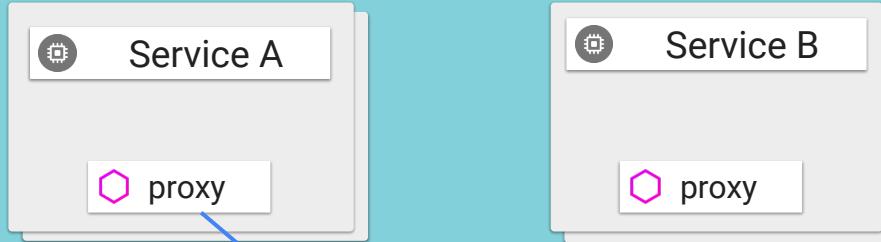
Life of a request in the mesh



Client-side Envoy forwards response to original caller.



Life of a request in the mesh

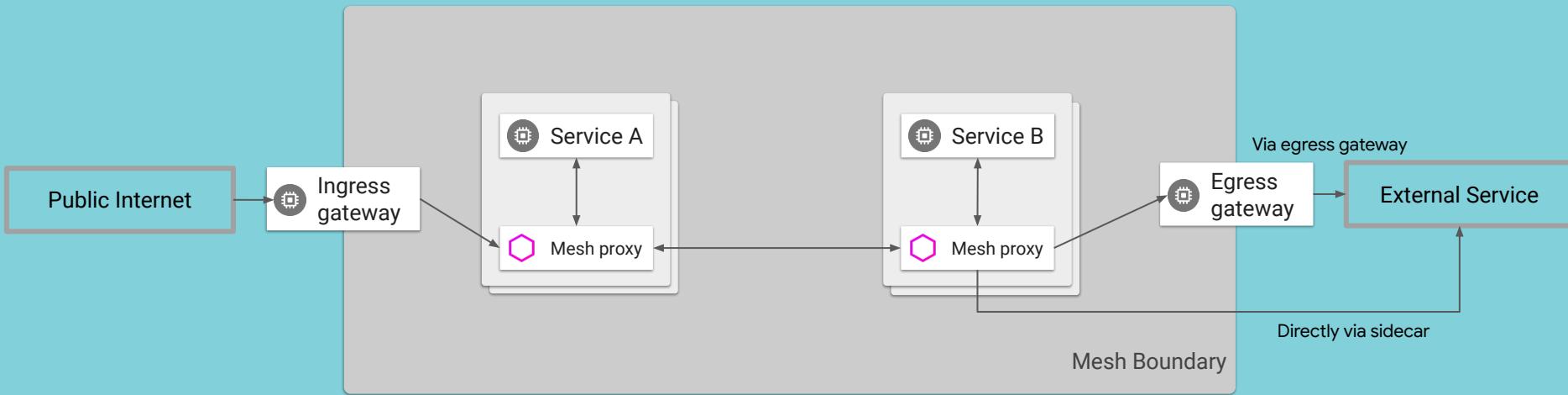


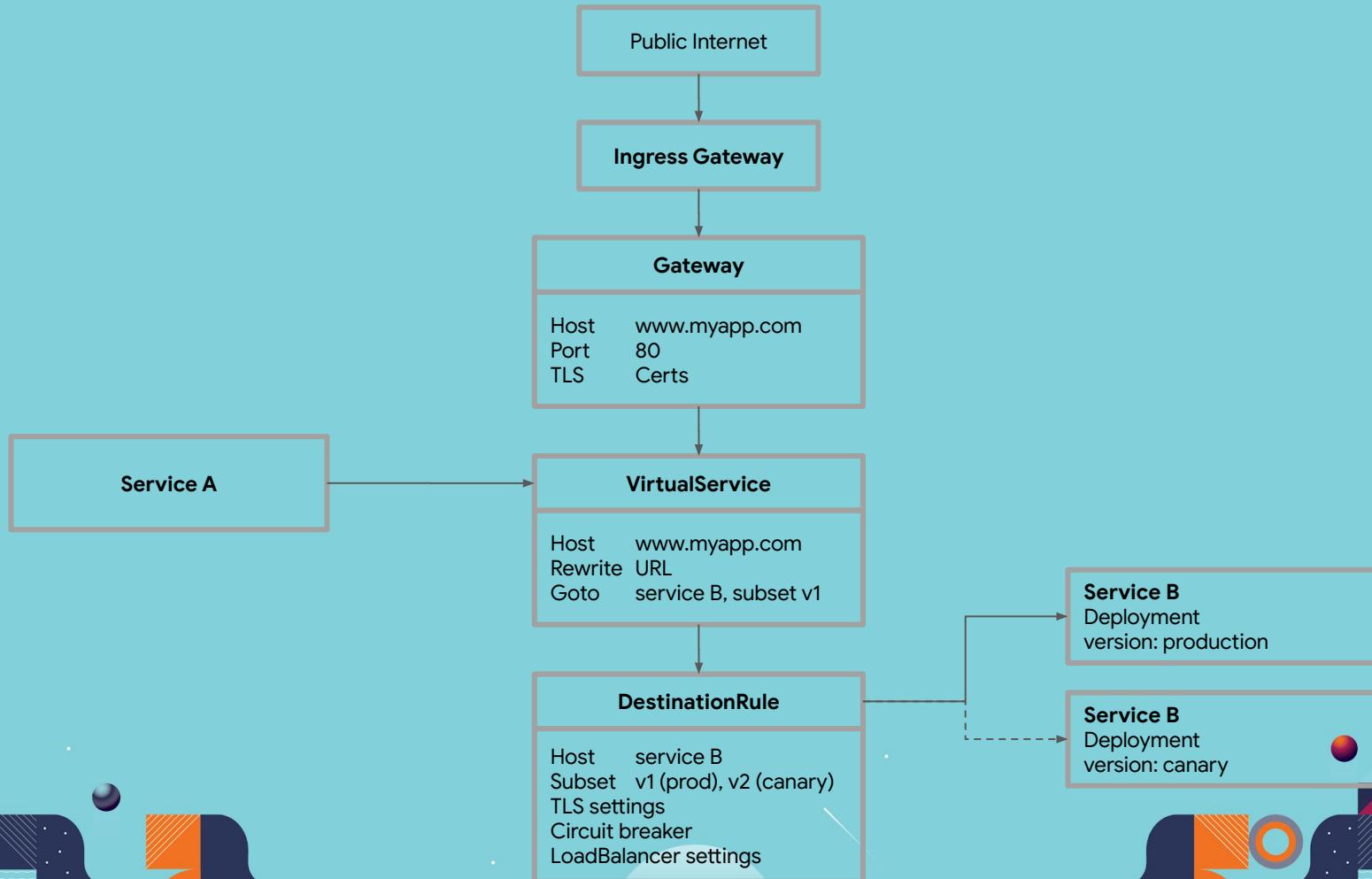
Client-side Envoy reports telemetry to Mixer (including client-perceived latency), which in turn notifies appropriate plugins



Traffic Management

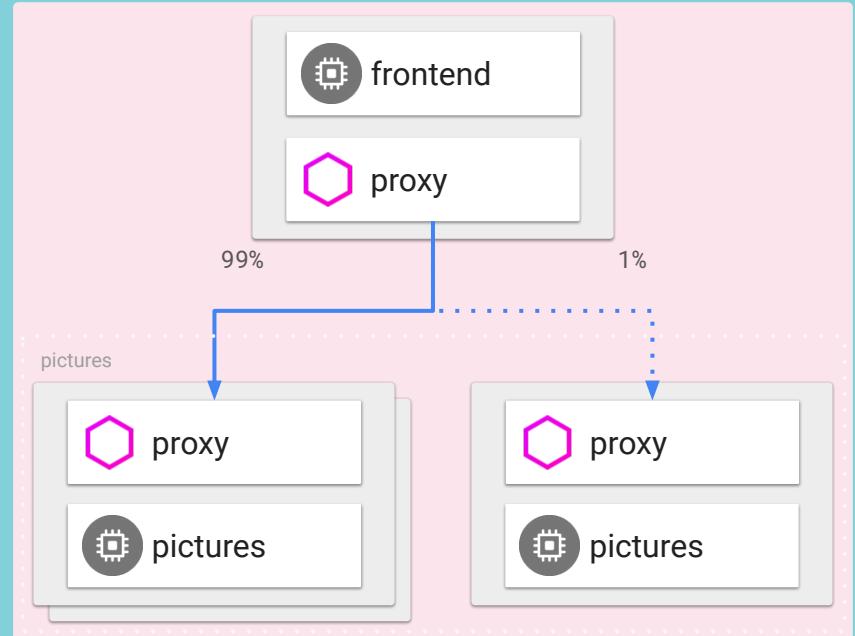






Application rollout

```
// A simple traffic splitting rule
destination: pictures.example.local
match:
  source: frontend.example.local
route:
- tags:
    version: v1.5
    env: prod
    weight: 99
- tags:
    version: v2.0-alpha
    env: staging
    weight: 1
```

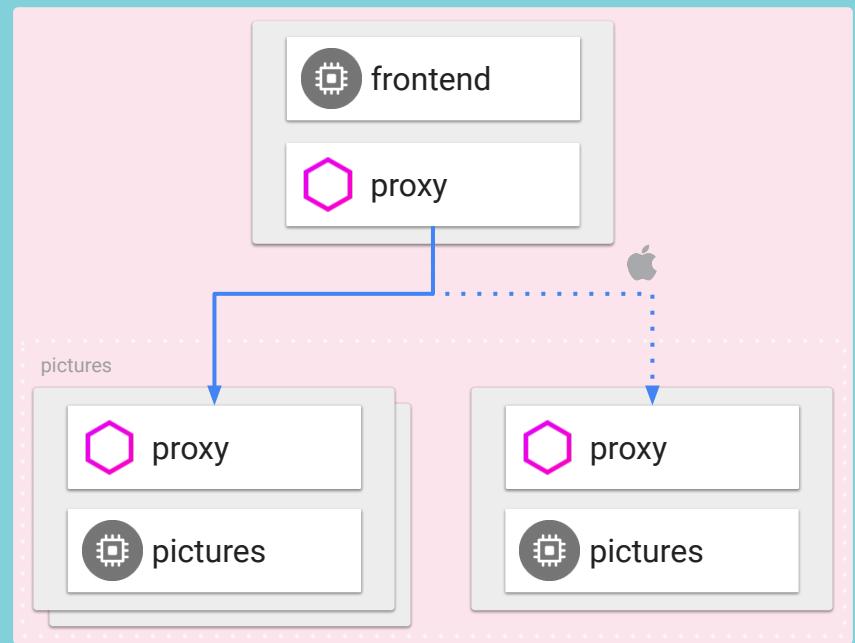


version: 1.5
env: prod

version: 2.0-alpha
env: staging

Traffic steering

```
// Content-based traffic steering rule
destination: pictures.example.local
match:
  httpHeaders:
    user-agent:
      regex: ^(.*)?(iPhone)(;.*?$
precedence: 2
route:
- tags:
  version: 2.0-alpha
  env: staging
```



version: 1.5
env: prod

version:
2.0-alpha
env:
staging

Resiliency

```
// Circuit breakers
destination: auth.cluster.local
policy:
- tags:
  version: v1
  circuitBreaker:
    simpleCb:
      httpConsecutiveErrors: 7
      sleepWindow: 5m
      httpDetectionInterval: 1m
```



Resiliency features

Timeouts

Retries with timeout budget

Circuit breakers

Health checks

AZ-aware load balancing w/ automatic failover

Control connection pool size and request load

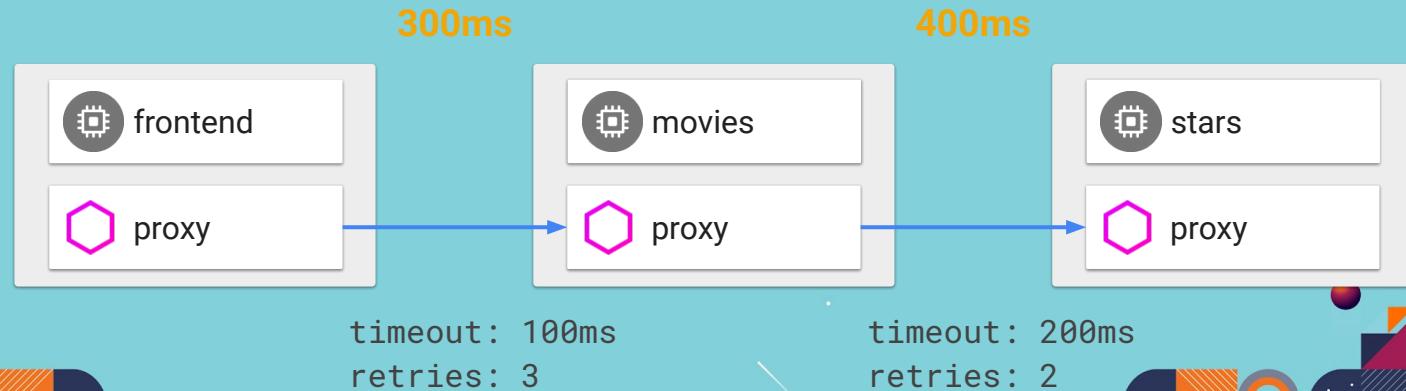
Systematic fault injection

Resiliency testing

Systematic fault injection to identify weaknesses in failure recovery policies

HTTP/gRPC error codes

Delay injection



Efficiency

L7 load balancing

- Passive/Active health checks, circuit breaks

- Backend subsets

- Affinity

TLS offload

- No more JSSE or stale SSL versions

HTTP/2 and gRPC proxying

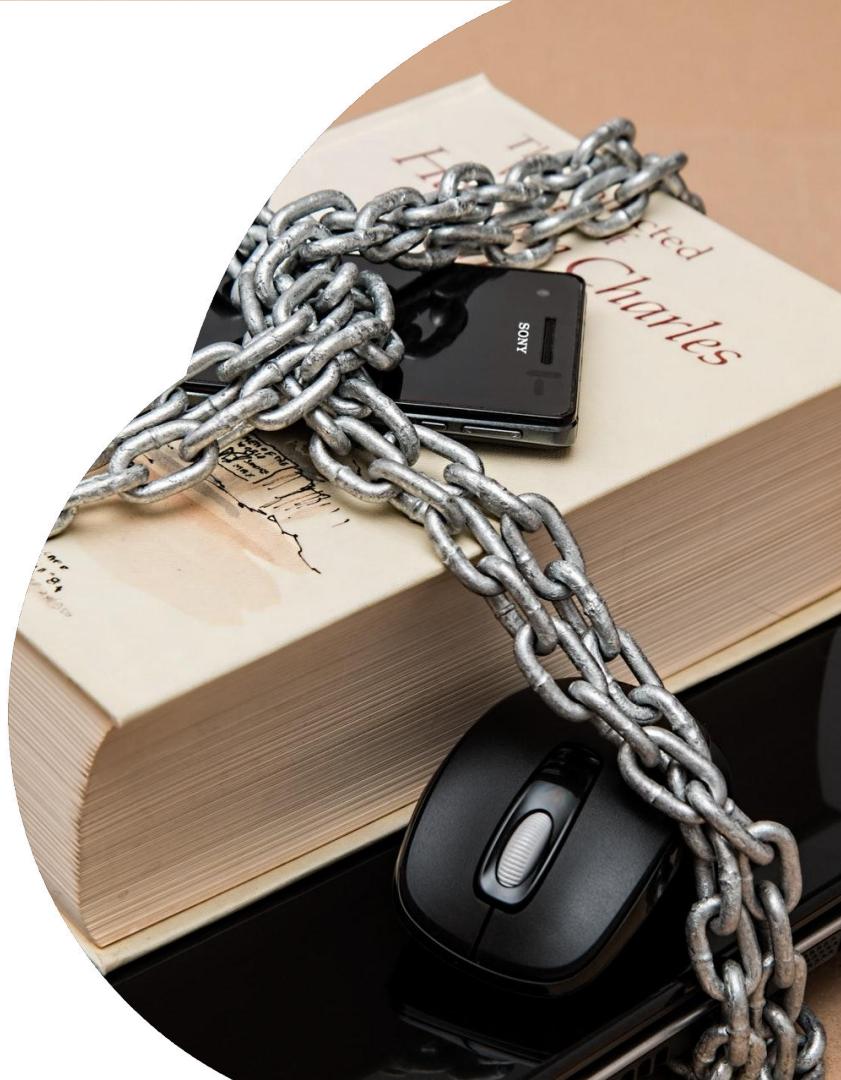


| Security



Security

- Verifiable identity
- Secure naming / addressing
- Traffic encryption
- Revocation



Lab 3: Control and secure with Istio

| Observability



Observability

Monitoring & tracing should not be an afterthought in the infrastructure

- Metrics without instrumenting apps
- Consistent metrics across fleet
- Trace flow of requests across services
- Portable across metric backend providers





Source

All

Source Version

All

HTTP Destination

All

TCP Destination

All

Destination Version

All

Zoom Out

Last 5 minutes

Refresh every 5s

Istio.io



Global Request Volume

35 ops

Global Success Rate (non-5xx responses)

84.9%

4xxs

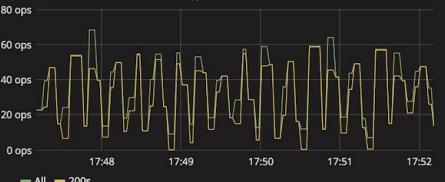
N/A

5xxs

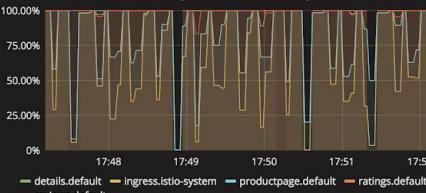
5.2 ops

Service Mesh

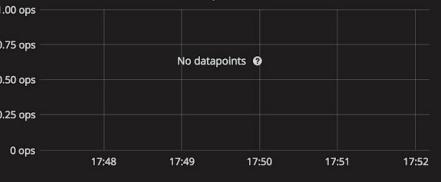
Request Volume



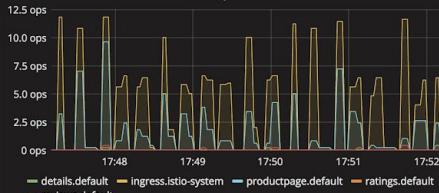
Success Rate by Service (non-5xx responses)



4xxs by Service



5xxs by Service



HTTP Services

▼ details.default.svc.cluster.local

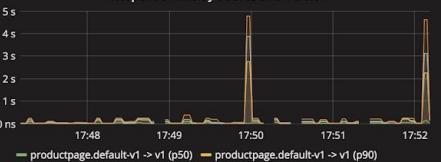
Requests by Source, Version, and Response Code



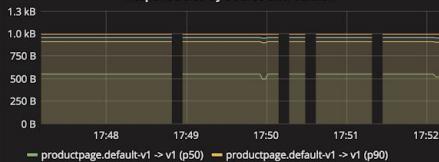
Success Rate by Source and Version (non-5xx responses)



Response Time by Source and Version



Response Size by Source and Version



▼ ingress.istio-system.svc.cluster.local

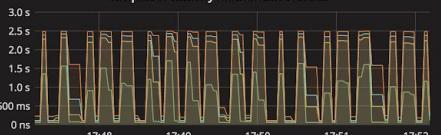
Requests by Source, Version, and Response Code



Success Rate by Source and Version (non-5xx responses)



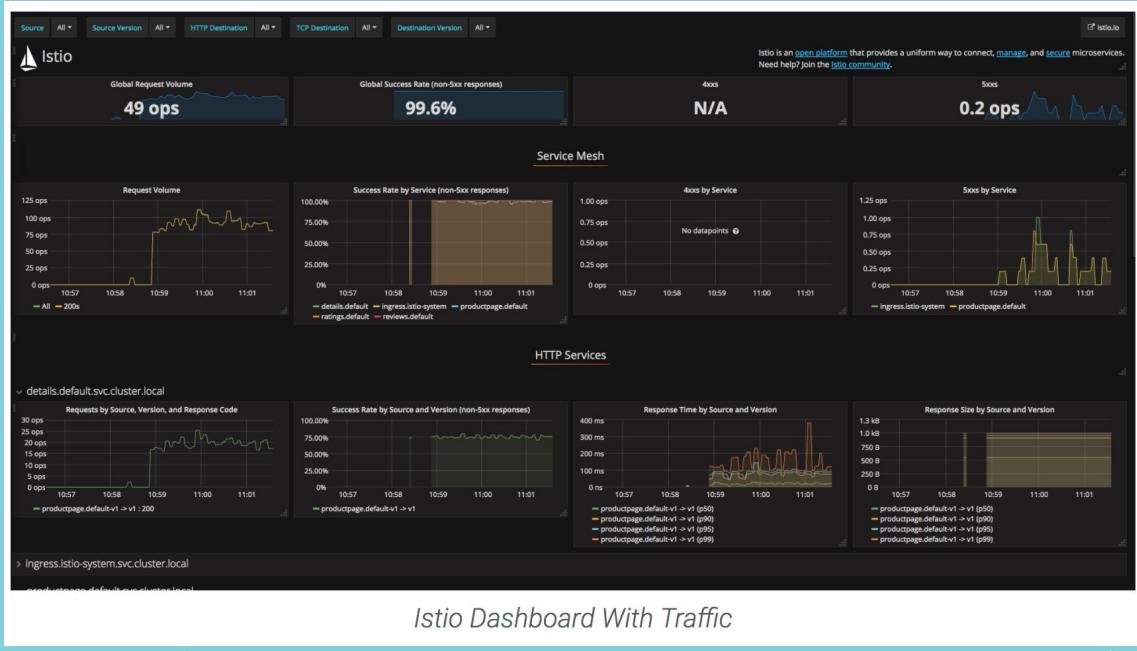
Response Time by Source and Version



Response Size by Source and Version



Traffic management and operational agility



1. Direct traffic away from starving instances
2. Scale by directing traffic to multiple versions
3. Roll out new versions without worrying about ops challenges
4. Apply access control, rate limiting policies to protect services from bad behavior

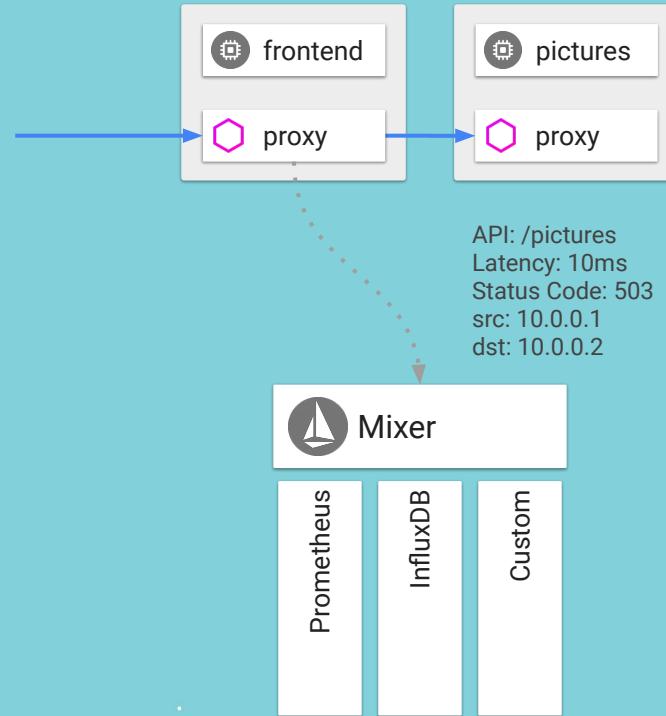


Metrics flow

Mixer collects metrics emitted by Envoy

Adapters in the Mixer normalize and forward to monitoring backends

Metrics backend can be swapped at runtime



Tracing

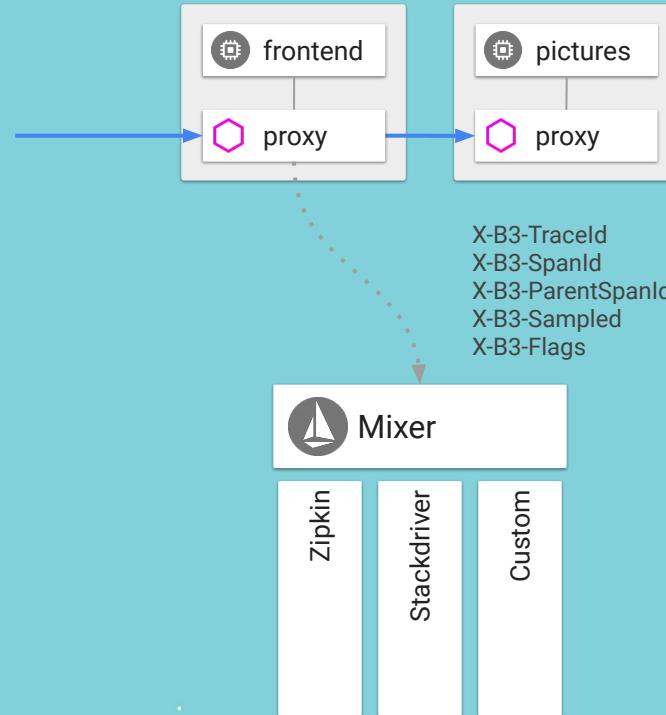
Applications do not have to deal with generating spans or correlating causality

Envoy generate spans

Applications need to forward context headers on outbound calls

Envoy send traces to Mixer

Adapters at Mixer send traces to respective backends





Lab 3: Control and secure with Istio
Lab 4: Monitor and troubleshoot with Istio







Cluster Ops Offerings Overview

**At Fairwinds, we help
engineering teams focus
on building great
products and services.**

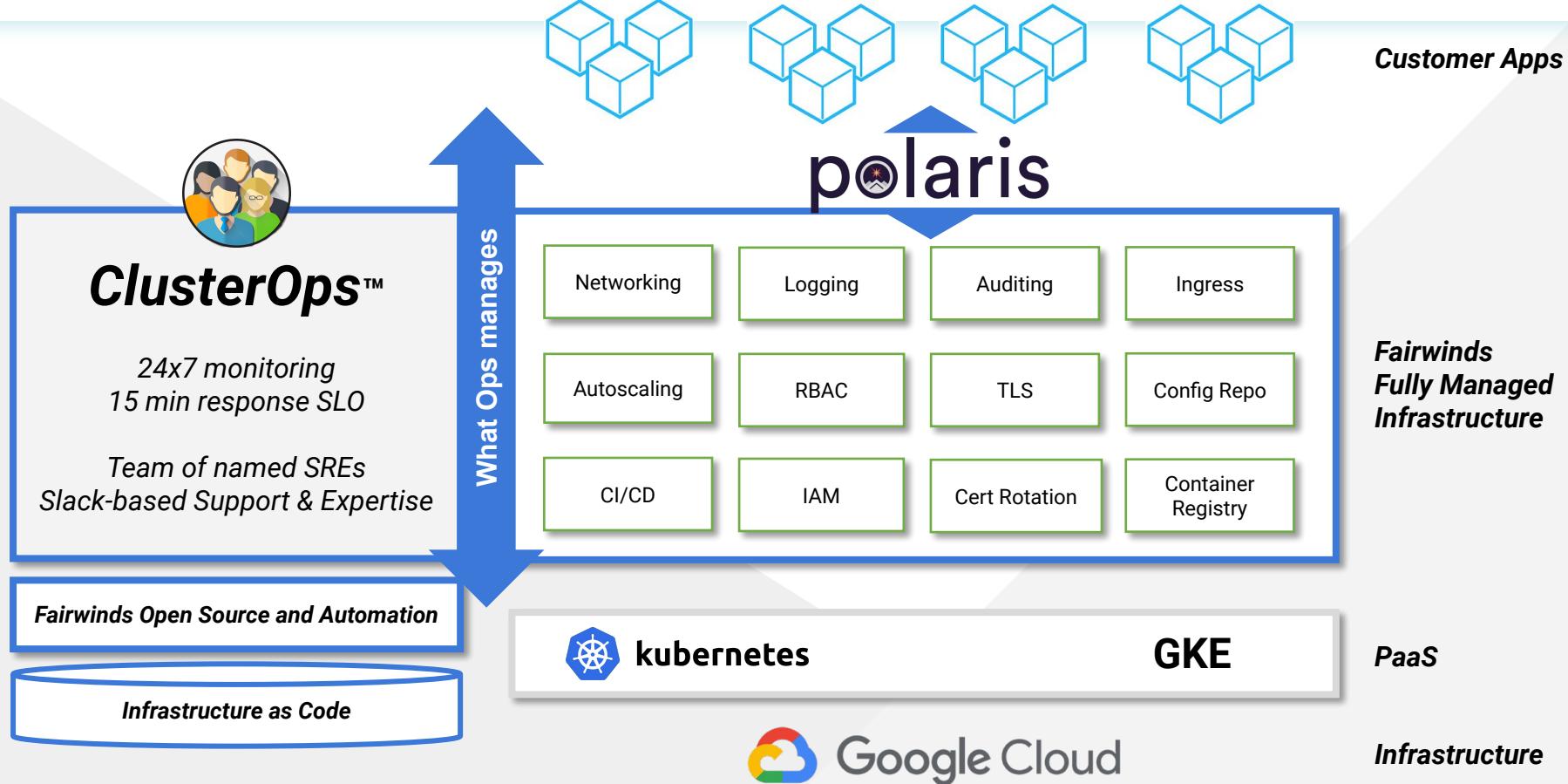
- Ship product faster
- Avoid risky trial-and-error
- Reduce operational cost

*Unleash the full benefits of
Cloud-native architecture*

- Performance
- Scalability
- High availability
- Security
- Cost predictability



Where ClusterOps fits



Subscription Offerings

	Architecture & Advisory	ClusterOps Express	ClusterOps Pro
Before Scenario	<i>Limited internal knowledge on K8s but a need to scale K8s across multiple teams</i>	<i>Containerized, maybe on K8s w/ challenges, need to get to K8s quickly and reliably</i>	<i>Partially containerized, some unique requirements, need to get to K8s quickly and reliably</i>
Typical Scope	<i>Multi-cluster</i>	<i>2 clusters</i>	<i>2+ clusters</i>
Base Architecture	<i>Delivered as Code</i>	<i>Opinionated</i>	<i>Opinionated</i>
Customization to Base Architecture	<i>Advisory</i>		<i>Yes, limited scope</i>
App Containerization	<i>Advisory</i>	<i>Advisory</i>	<i>1 app/service containerized</i>
Reliability Tuning	<i>Advisory</i>	X	X
Upgrades		X	X
Security Patching		X	X
Cluster Maintenance		X	X
24x7 Kubernetes Paging		X	X
Slack-based Support	X	X	X
Container Examples & Training	X		X
CI/CD Workflow Examples & Training	X		X
Misc. DevOps Projects	<i>Extra SOW</i>	<i>Extra SOW</i>	<i>Extra SOW</i>

What is Polaris?

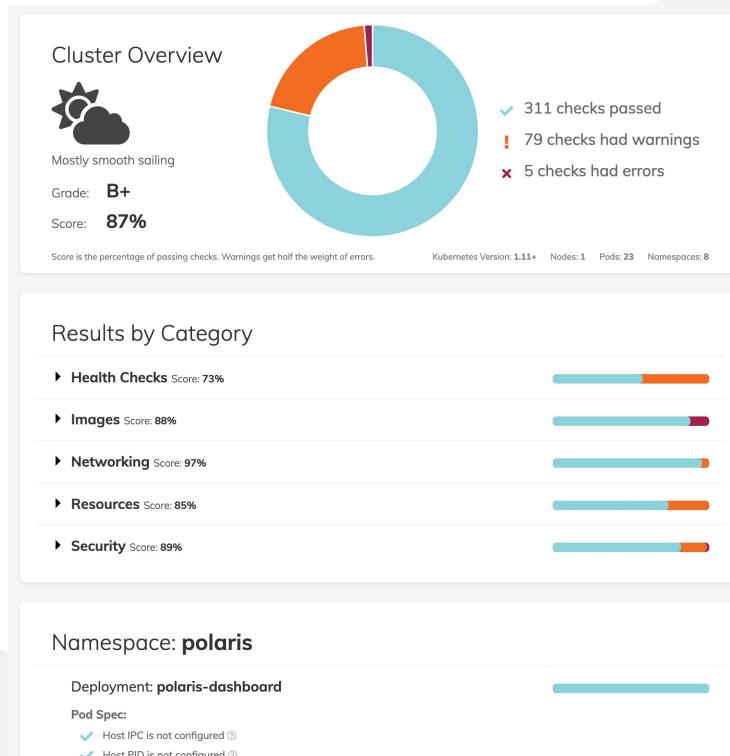
Fairwinds Polaris validates the handoff between development and production.

Fairwinds Polaris ensures your Kubernetes deployments, and overall cluster, aligns to Kubernetes best practices.

Polaris is Open Source and looks for common Kubernetes configuration issues across the following categories:

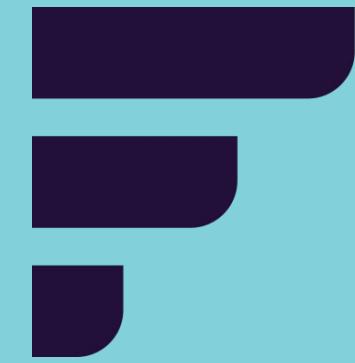
- Health Checks
- Images
- Networking
- Resources
- Security

polaris



**Thank you. Please take our survey to be entered
into the raffle.**

<http://bit.ly/advkubefeedback>

The logo consists of a stylized letter 'F' composed of two thick, dark blue horizontal bars. The top bar is slightly curved at the right end, and the bottom bar has a vertical cutout on its left side.

Fairwinds

