

# 12 Factor Streaming Data Apps on Kubernetes



## Presenters



Andrew Stevenson  
CTO, Lenses.io



Fran Perez  
Backend Engineer, Lenses.io

# Open-Source tooling for Apache Kafka & Kubernetes



**>1 Million**  
Downloads



**>5000**  
Github Stars

## Commercial offering for Kafka & Kubernetes



DAIMLER



JCDecaux

Sainsbury's

PostFinance

CISION

RBS

telenet

ubeeqo

workable

Aduno Gruppe  
the smart way to pay

PickMe

SCHULER

nuvo

AGCO  
Your Agriculture Company

Quiup

VORTEXA

Safe  
Motos

ARTICLE.

Autoliv

CREDIMI

O.C.TANNER

# We Support Strategic Real-time Data Projects



## BABYLON HEALTH

Delivering affordable healthcare to everyone on earth through AI-driven chatbots

### Affordable Healthcare



## VORTEXA

Tracks the movement of over \$7 trillion worth of seaborne oil-based products in real-time

### Analytics



## ADUNO GRUPPE

Increasing customer engagement in their marketing campaigns

### Customer Engagement

# Lenses.io Mission

Reduce the **pain, cost & complexity** of  
modern data platforms.



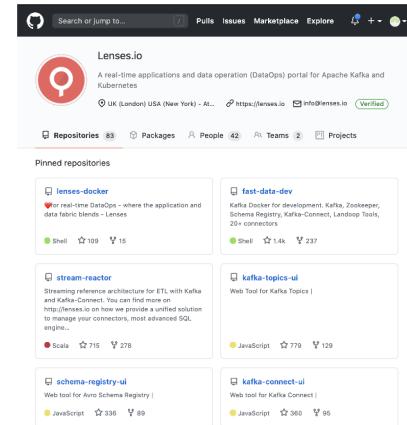
## [lenses.io/start](https://lenses.io/start)

All-in-one Kafka + Lenses.io developer Docker box



## [github.com/lensesio](https://github.com/lensesio)

Open-Source tooling & connectors

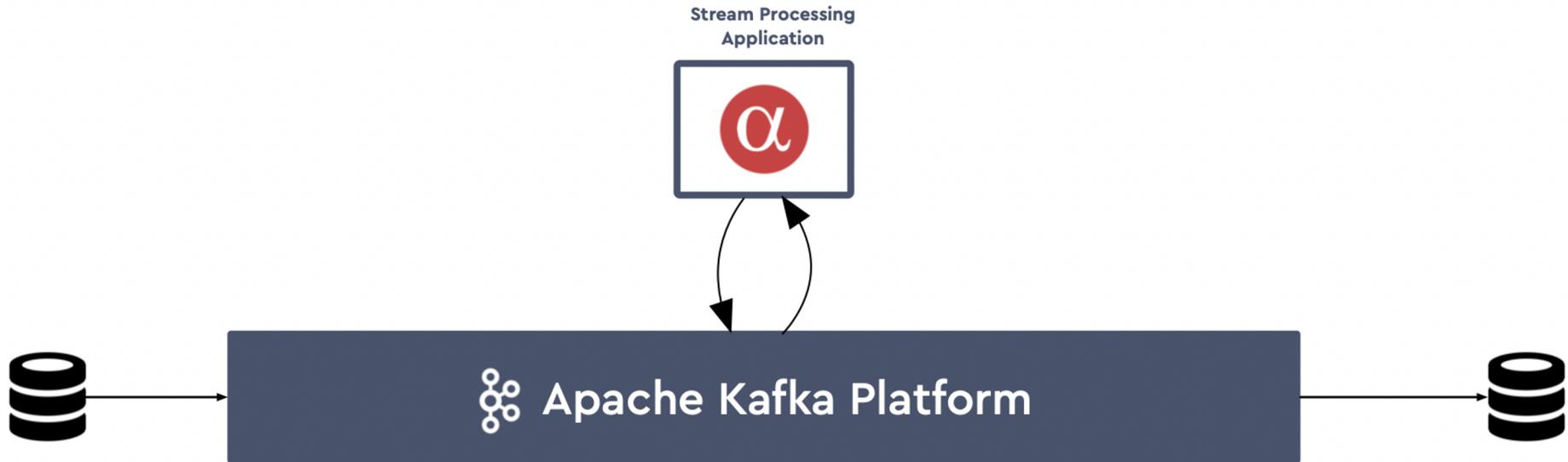


[launchpass.com/lensesio](https://launchpass.com/lensesio)

# Agenda

- ✓ Overview of a DataOps streaming platform
- ✓ Demo our deployment framework that takes the Ops out of DataOps
- ✓ How we built it and how it scales to any type of application

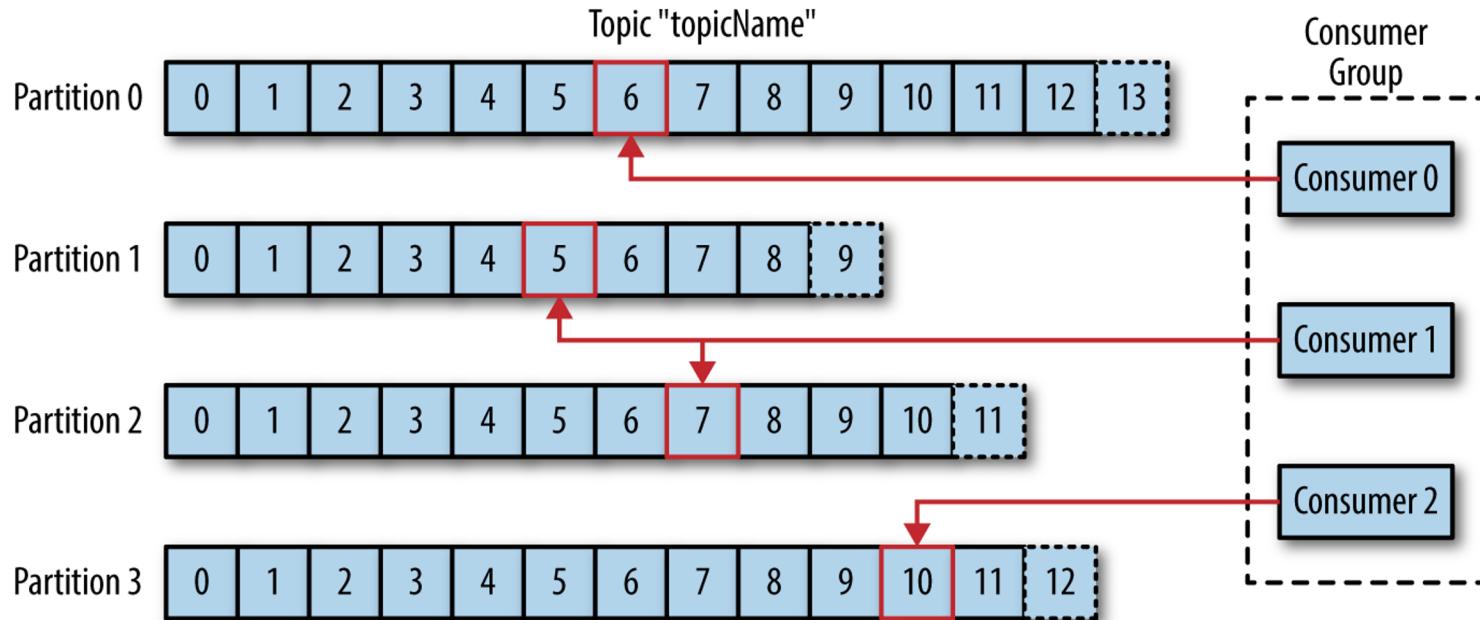
# Data platform



What is Kafka?

distributed commit log

# Distributed commit log?





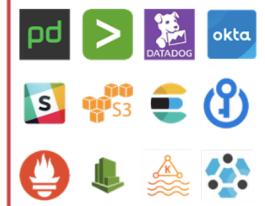
Self-Service  
Administration

Data Discovery  
& Exploration

Data Processing  
& Integration

Monitoring  
& Observability

Security  
& Governance



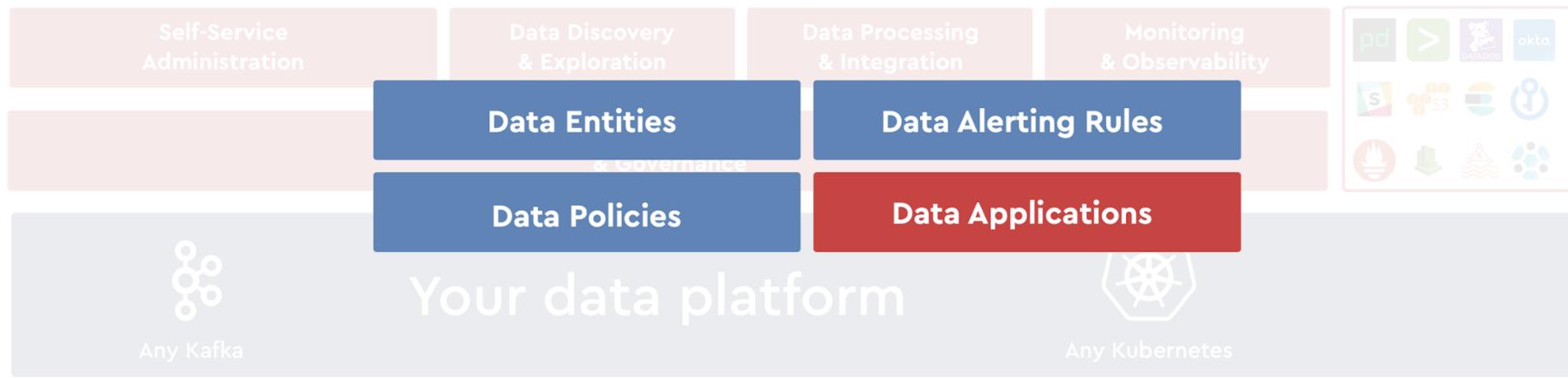
Any Kafka

Your data platform

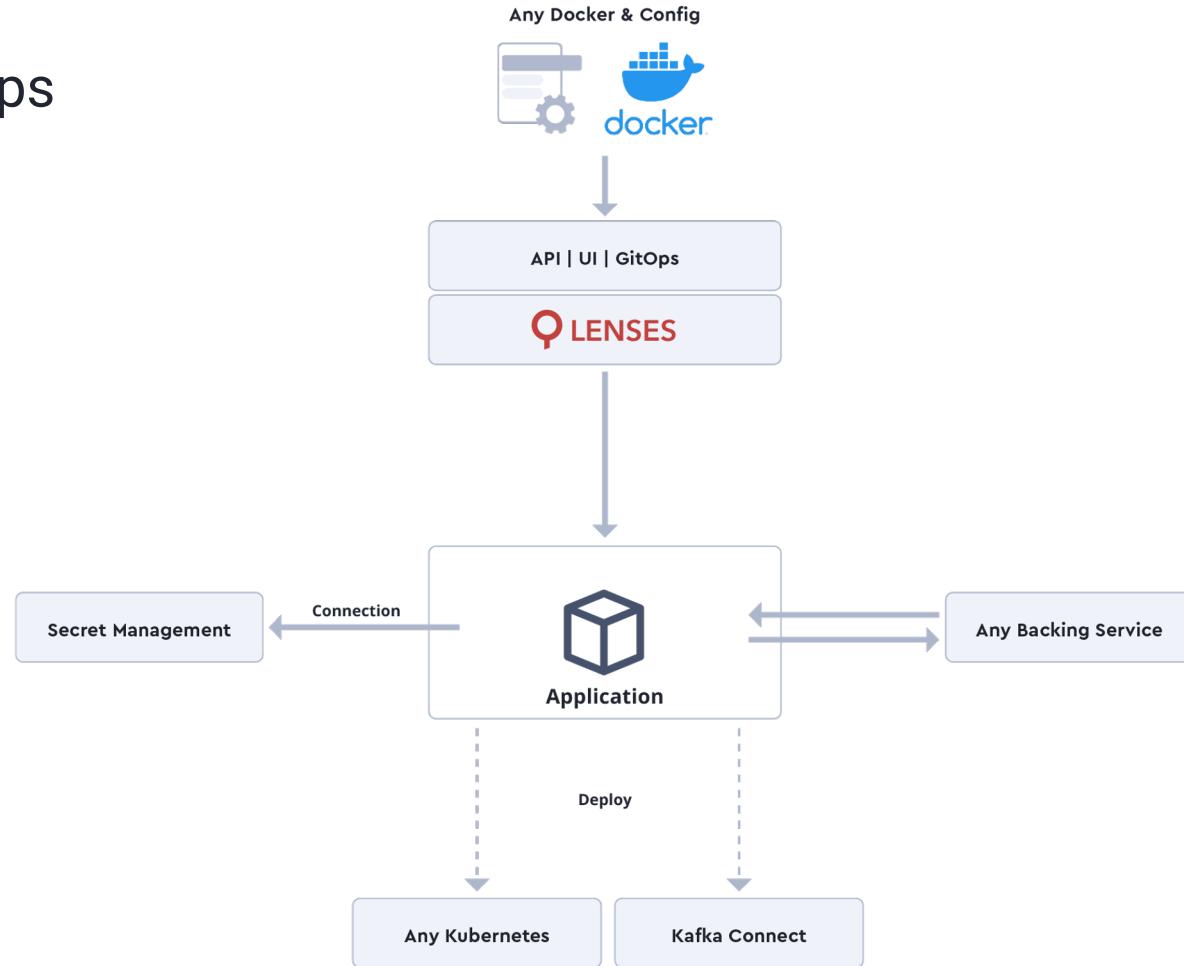


Any Kubernetes

# If not kafka, what are we deploying?

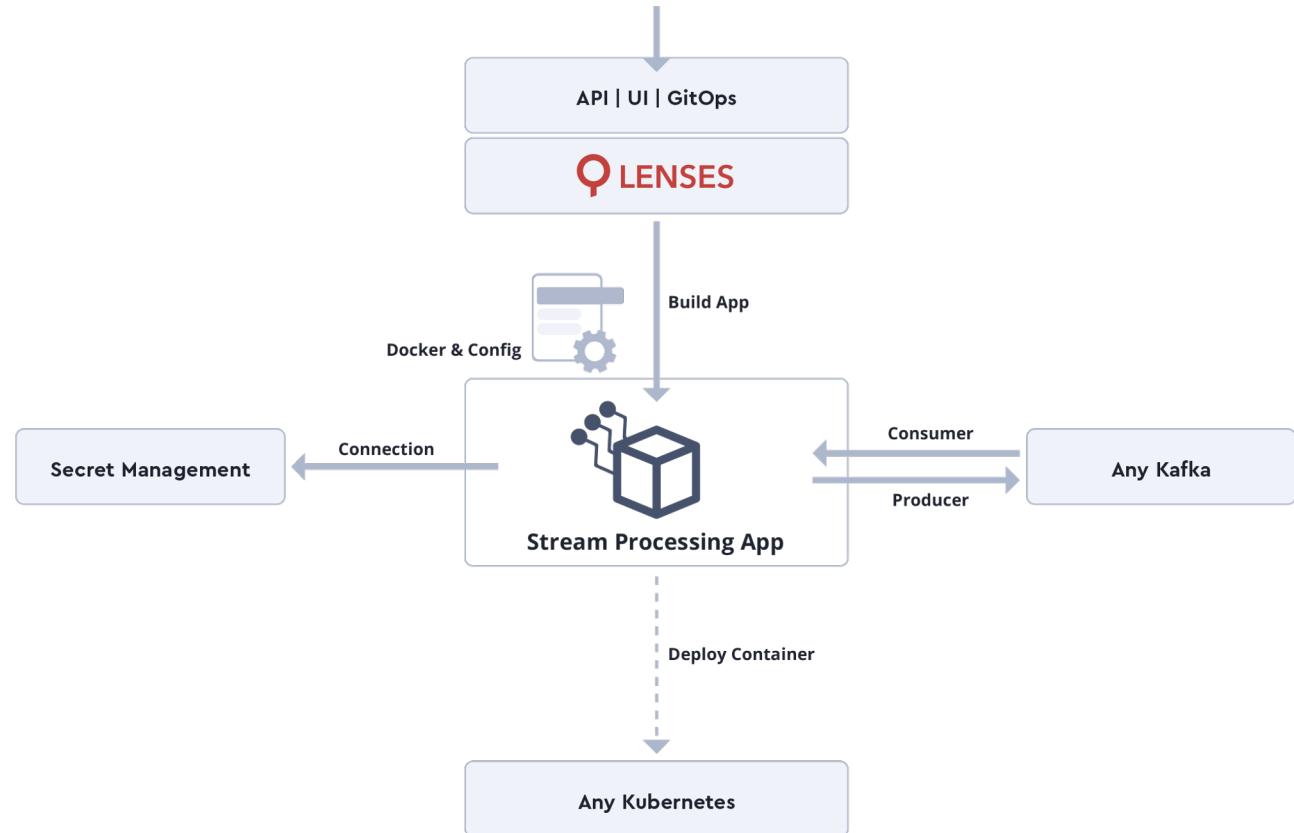


# Data intensive apps with Lenses



# Streaming SQL for Kubernetes

```
INSERT INTO electricity_events_avg
SELECT STREAM customer_id ,
       AVG (KW) AS KW
FROM electricity_events
WINDOW BY HOP 10m, 5m
GROUP BY customer_id
```

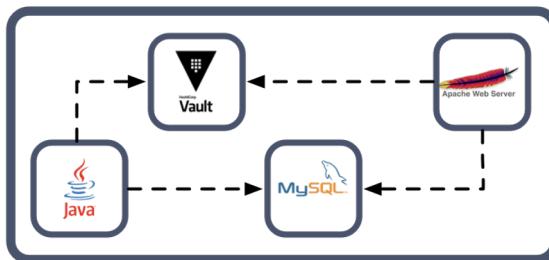




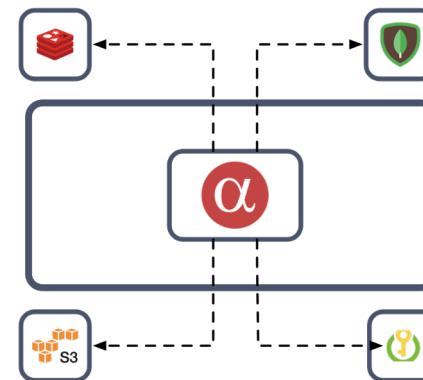
3 minutes demo of deploying a streaming  
app

# What's special about real-time apps?

- ✓ Unbounded data
- ✓ Many sources
- ✓ React to data
- ✓ State
- ✓ Inputs/Outputs
- ✓ Data access
- ✓ Complex security models
- ✓ Connection management



Traditional Self-Contained Microservice



Stream Processing Platform

# Platform as a Service



What does it mean

What should we care about

New challenges

12 Factor apps manifest

The Twelve-Factor App

Adam Wiggins

In the modern era, software is commonly delivered as a service, called web apps, or *software-as-a-service*. The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use declarative formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a clean connect with the underlying operating system, offering maximum portability between execution environments;
- Are suitable for deployment on modern cloud platforms, obviating the need for servers and systems administration;
- Minimize divergence between development and production, enabling continuous deployment for maximum agility;
- And can scale up without significant changes to tooling, architecture, or development

Introduction

The Twelve-Factor App

Page 1

Page 2

2 pages left in this chapter

I. Codebase

One codebase tracked in revision control, many deploys

A twelve-factor app is always tracked in a version control system, such as Git, Mercurial, or Subversion. A copy of the revision tracking database is known as a *code repository*, often shortened to *code repo* or just *repo*.

A codebase is any single repo (in a centralized revision control system like Subversion), or any set of repos who share a root commit (in a decentralized revision control system like Git).

Codebase Deploy

One codebase maps to many deploys

There is always a one-to-one correlation between the codebase and the app:

- If there are multiple codebases, it's not an app – it's a distributed system. Each component in a distributed system is an app, and each can individually comply with twelve-factor.
- Multiple apps sharing the same code is a violation of the twelve-factor app principle.

The Twelve-Factor App

Page 5

Page 6

1 page left in this chapter

Scale (running processes)

Workload diversity (process types)

This does not exclude individual processes from handing their own internal multiplexing, via threads inside the runtime VM, or the async/evented model found in tools such as EventMachine, Twisted, or Node.js. But an individual VM can only grow so large (vertical scale), so the application must also be able to span multiple processes running on multiple physical machines.

This process model truly shines when it comes time to scale out. The share-nothing, horizontally partitionable nature of twelve-factor app processes means that adding more concurrency is a simple and reliable operation. The array of process types and number of processes of each type is known as the *process formation*.

The process model truly shines when it comes time to scale out. The share-nothing, horizontally partitionable nature of twelve-factor app processes means that adding more concurrency is a simple and reliable operation. The array of process types and number of processes of each type is known as the *process formation*.

The Twelve-Factor App

Page 29

Page 30

1 page left in this chapter



# How Lenses approaches it

**Connections**

**Secret Providers**

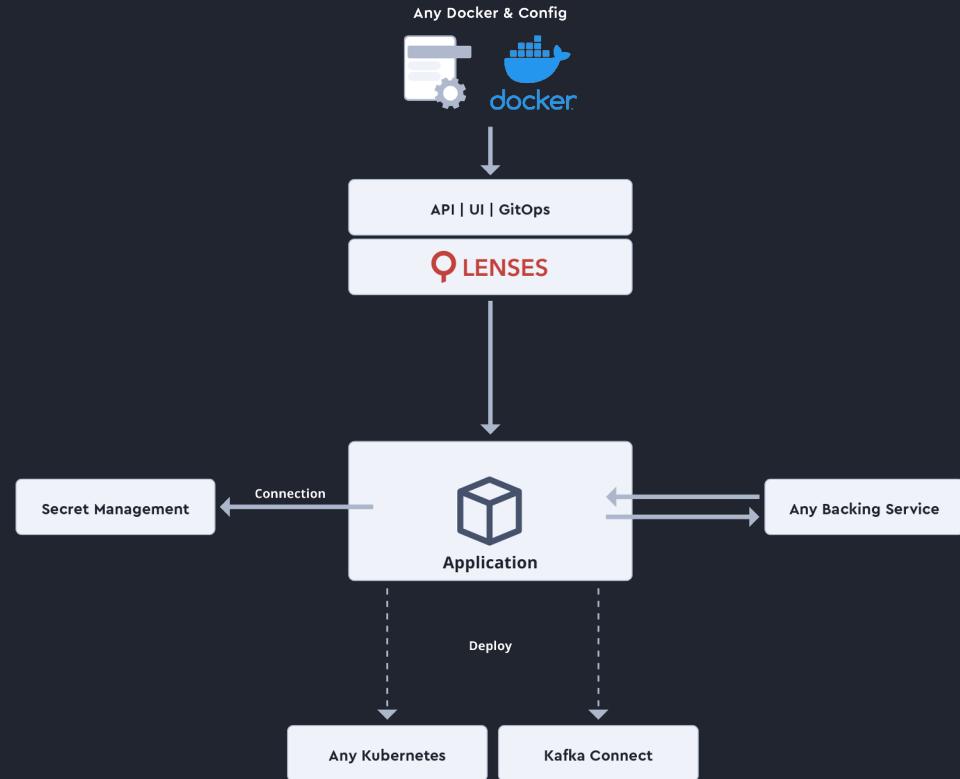
**Applications**

**Deployment  
Targets**

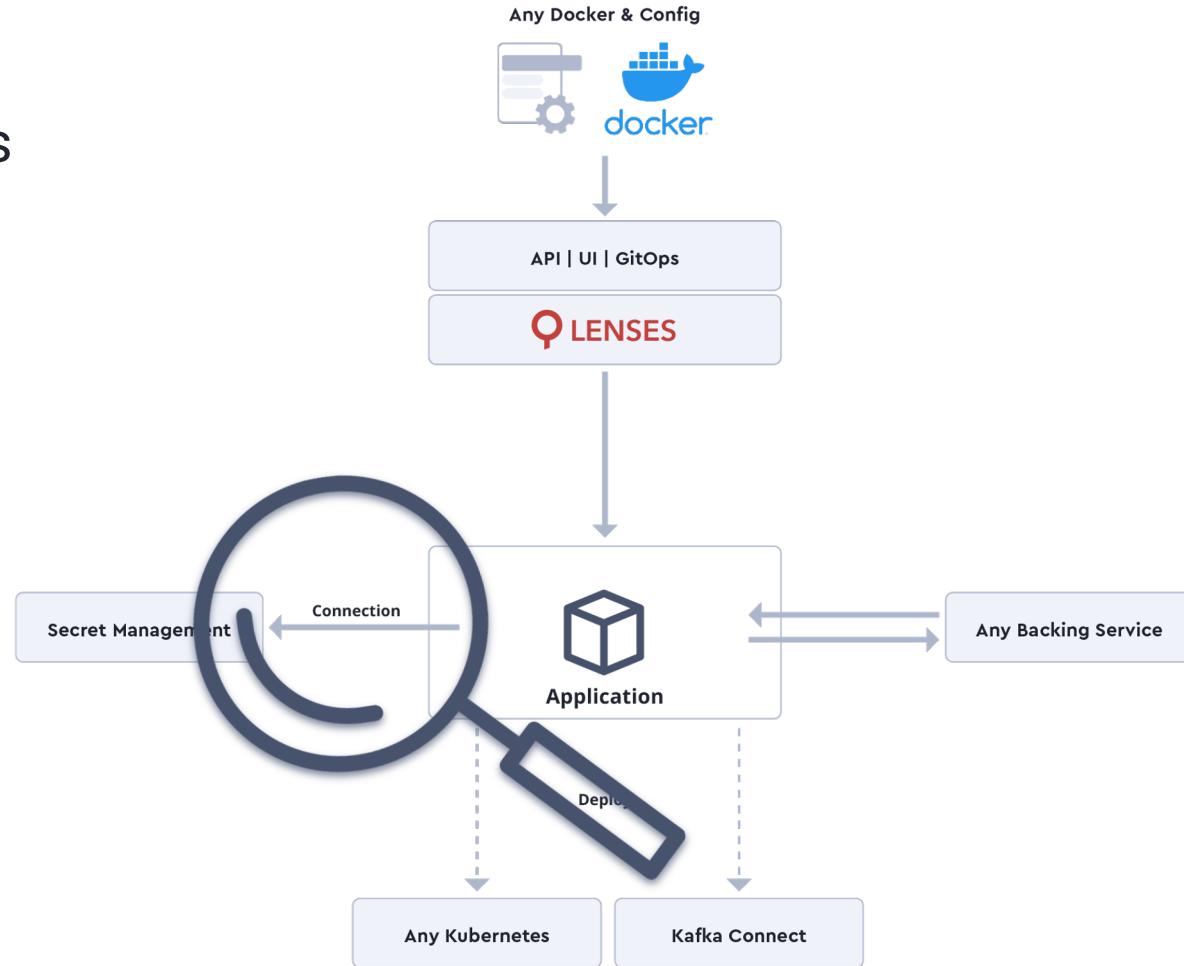
# Templating systems <> Streaming apps

Common concerns for any app:

- ✓ Input params
- ✓ Data sources
- ✓ Data sinks
- ✓ Security
- ✓ App image to run
- ✓ Scale
- ✓ Liveness probe
- ...



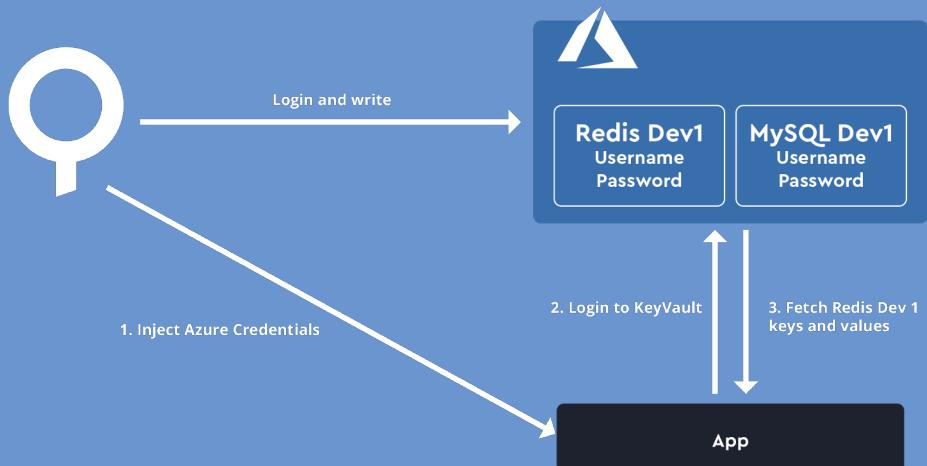
# Secret Providers



## Secret Providers

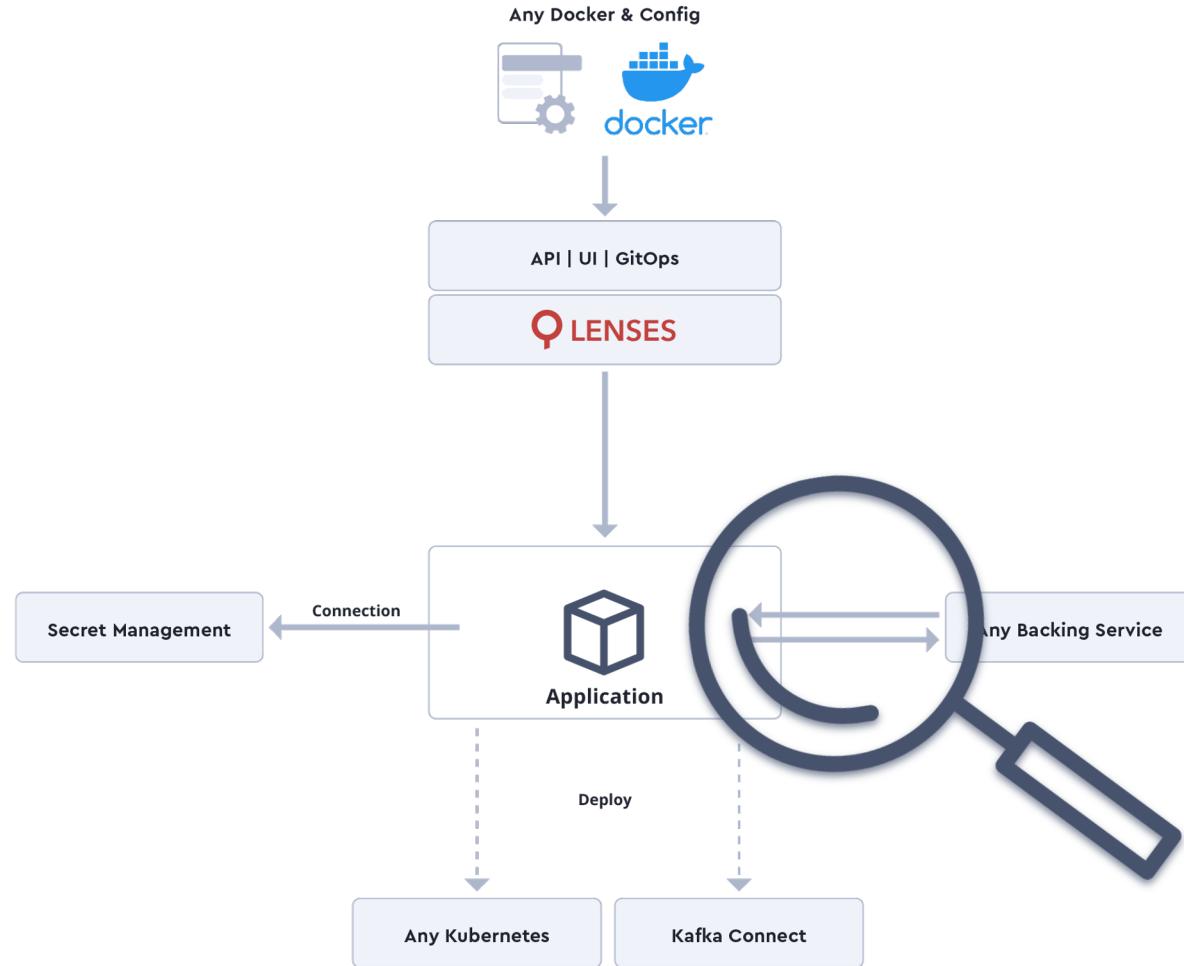
Secret providers. Who (where in the end) is going to keep secrets e.g.:

- ✓ Lenses itself
- ✓ AzureKeyVault
- ✓ HashicorpVault
- ✓ ...



SECRET\_KEY\_NAME=[PROVIDER]:[VAULT\_URL]:[LOOKUP\_KEY]:mounted:type

# Connections

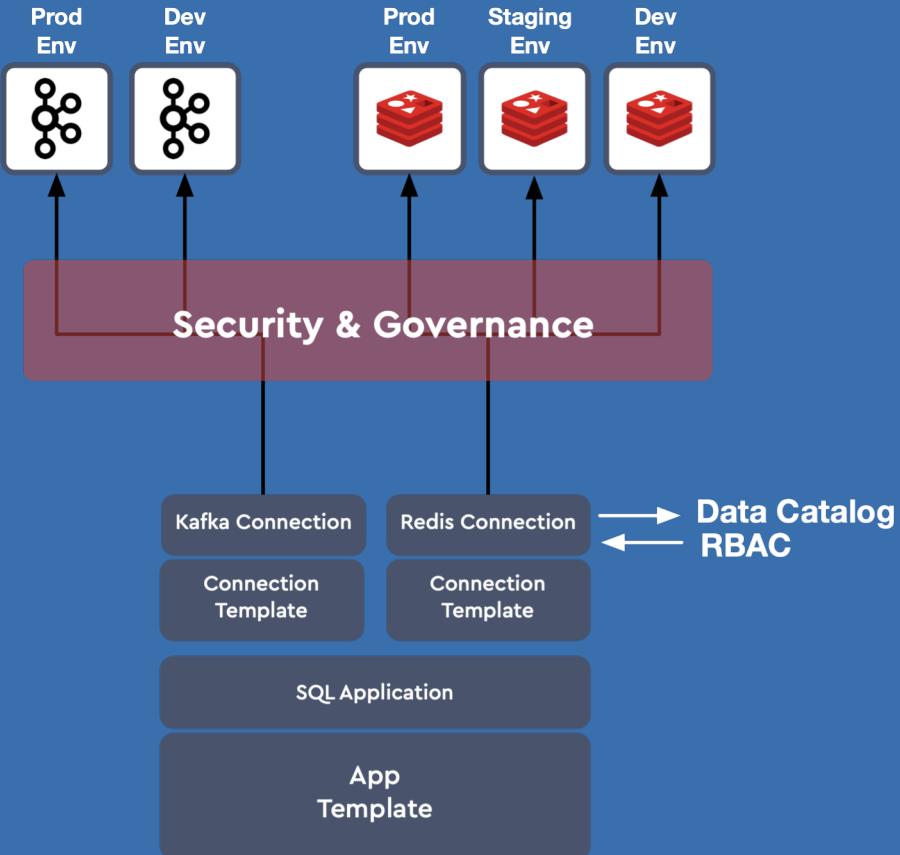


## Connections

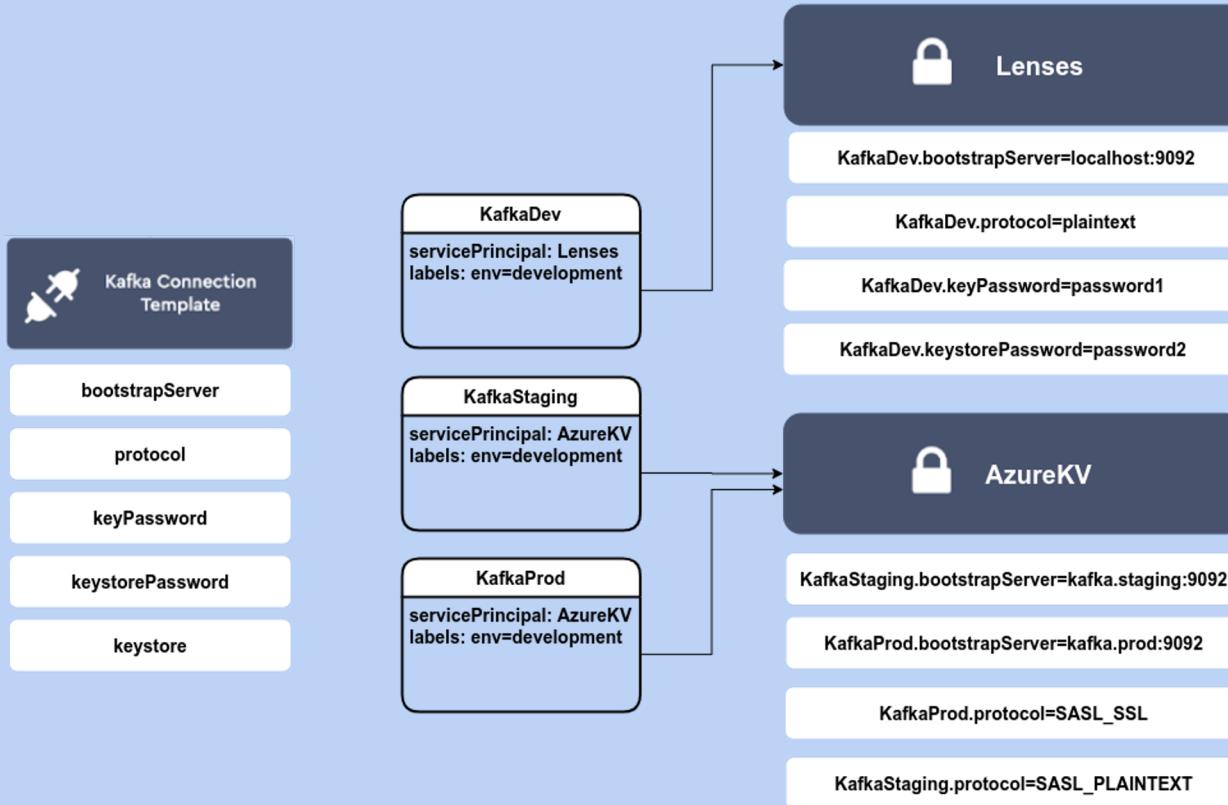
Connection descriptor. Properties needed to connect to a system e.g.:

- ✓ Kafka brokers
- ✓ DB connection string

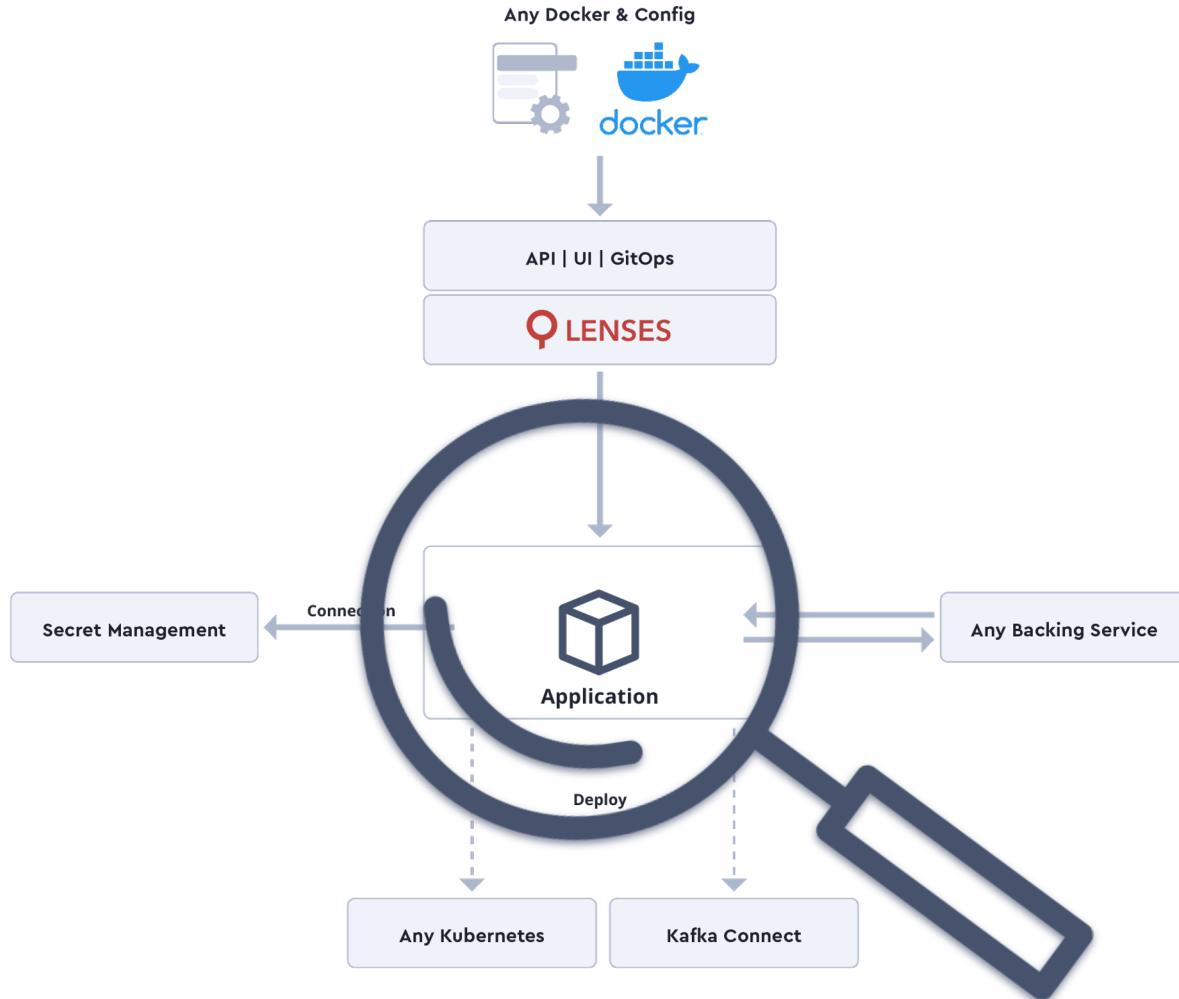
\*Connections are referenced by Names



# Storing connection details



# Applications



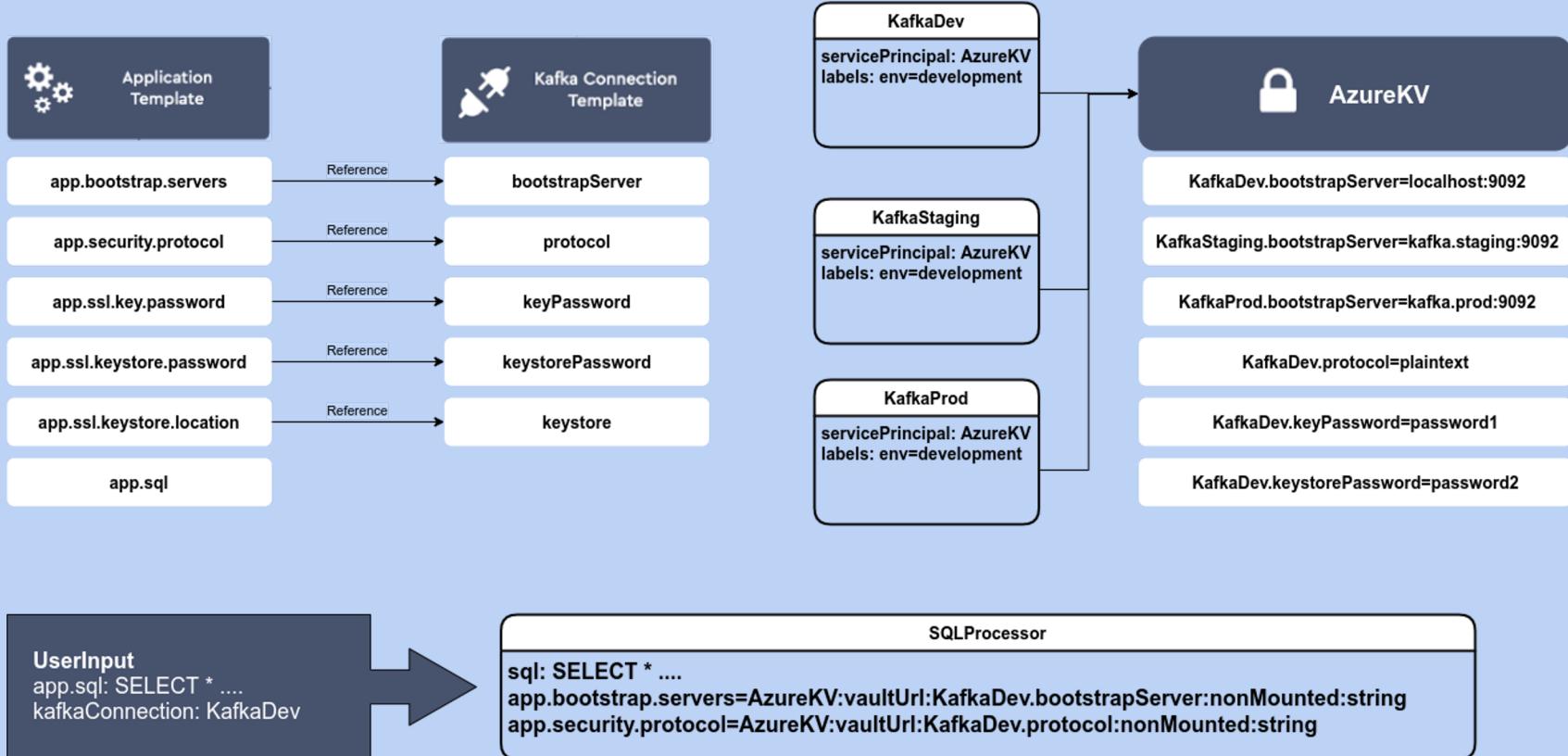
# Applications

Application descriptor. Properties needed by an Application to be executed, including references to Connection instances based on existing templates e.g.:

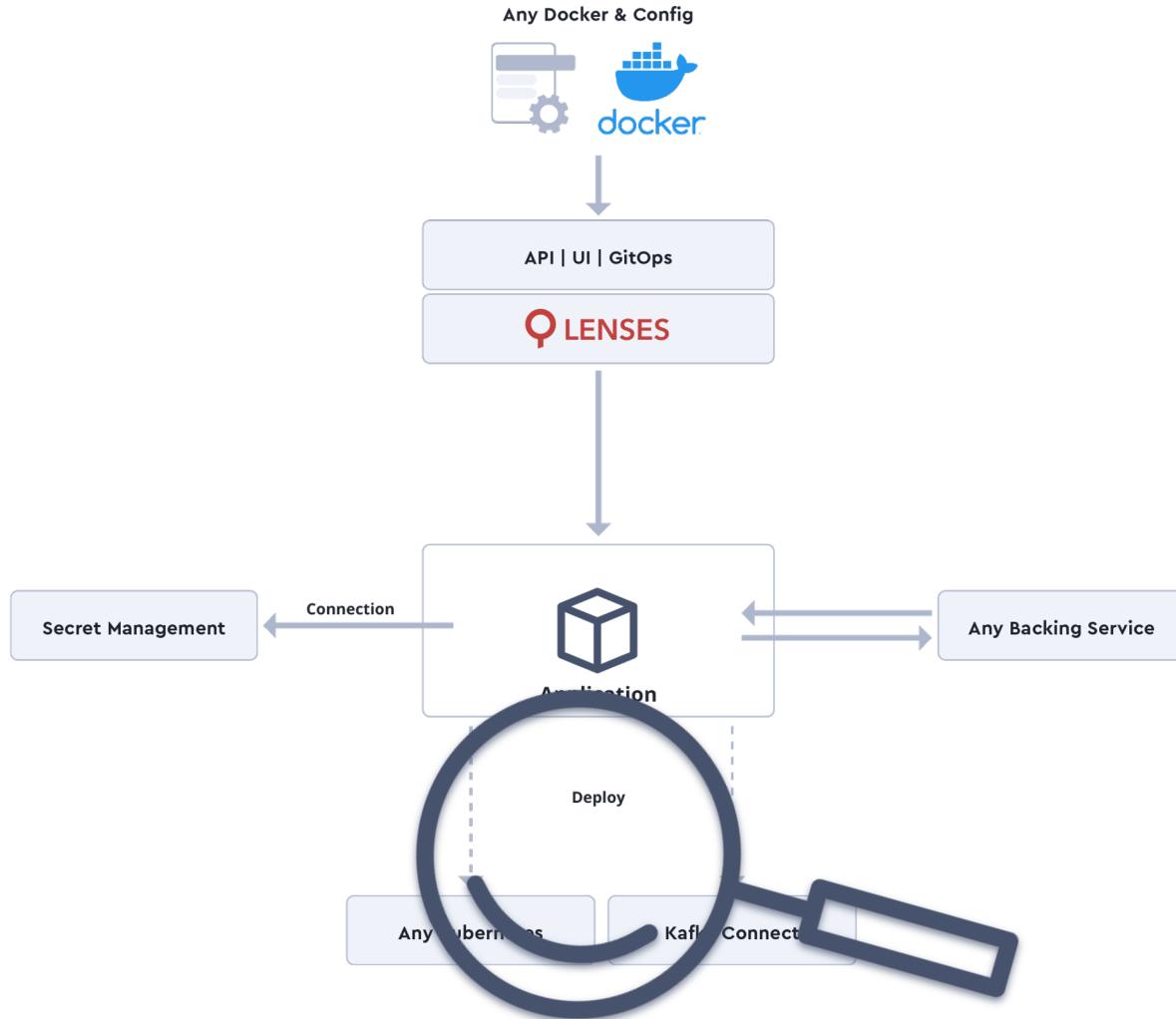
- ✓ SQL query for SQL Processors
- ✓ KCQL query for Kafka Connector
- ✓ Twitter bot: hashtag to listen to
- ✓ Twitter bot: twitter connection as Connection reference

```
name: SQLProcessor
author: Lenses.io
appLanguage: Scala
enabled: true
description: SQL Processor.
templateVersion: 1
createdBy: lenses
category: Application
configurationKeys:
- key: kafka.bootstrap.servers
  mounted: false
  required: true
  connectionReference:
    templateName: Kafka
    propertyKey: kafkaBootstrapServers
  category: Reference
  createdBy: lenses
  displayName: Kafka Bootstrap Servers
  placeholder: PLAINTEXT://host1:port1,SSL://host2:port2
  description: Comma separated list of protocol://host:port to use for initial connection to Kafka.
```

# Wiring Apps with real properties



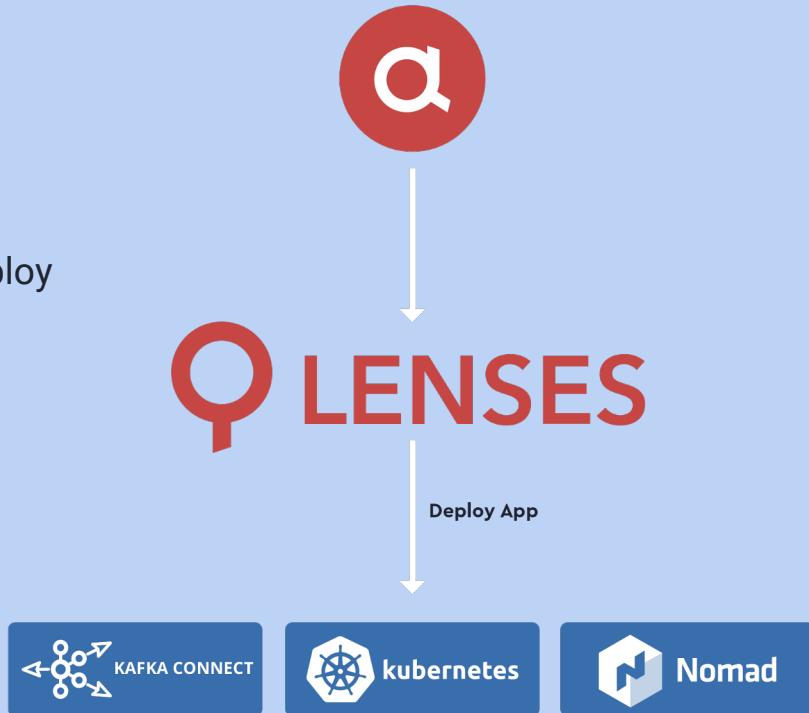
# Deployments



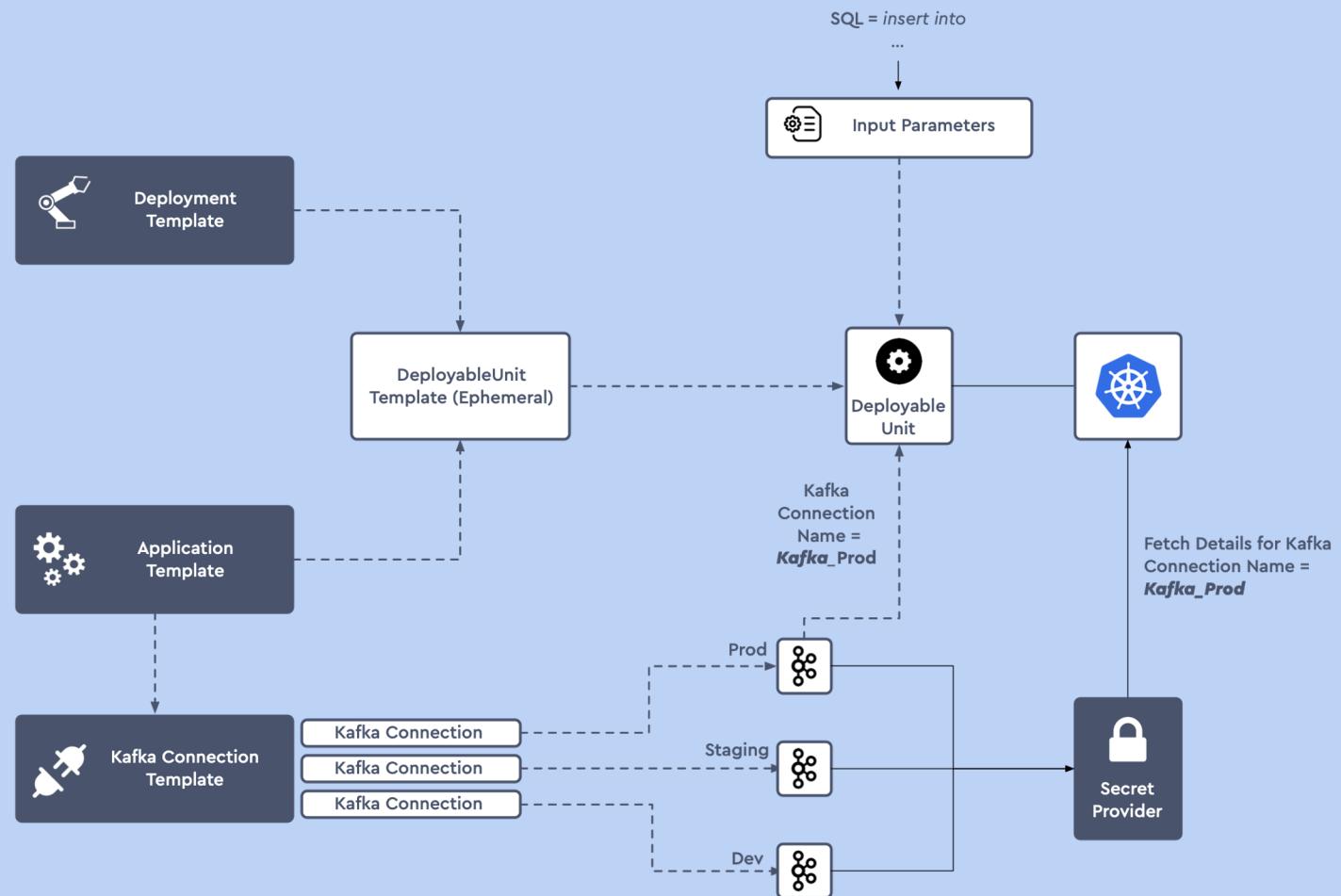
## Deployments

Deployment descriptor. Properties needed to deploy an application e.g.:

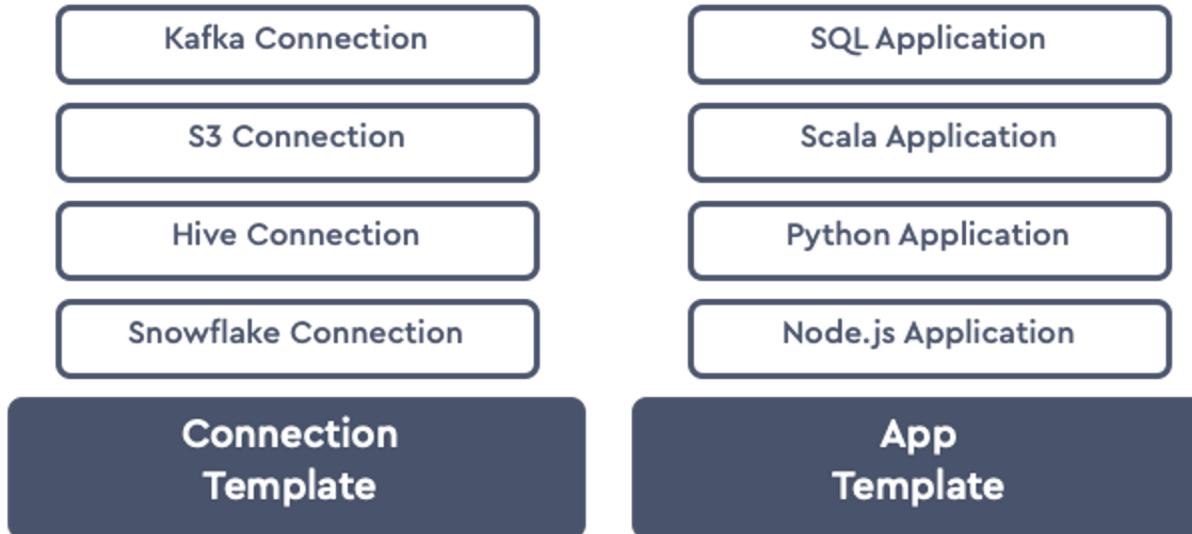
- ✓ Docker Image
- ✓ Number of runners
- ✓ Namespace (kubernetes)
- ✓ Shutdown hook
- ✓ Liveness probe



# Let's cook

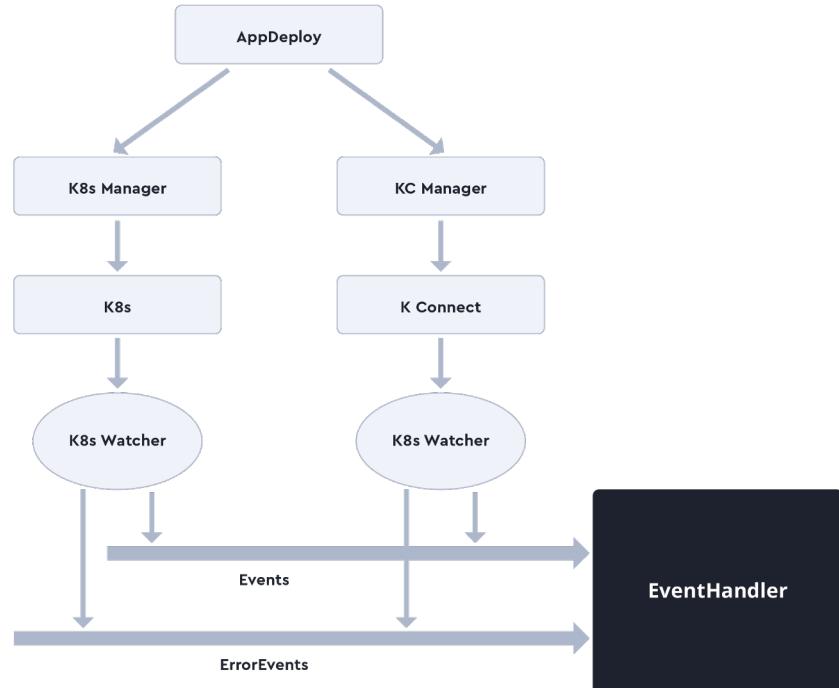


# Scalability - Can be for any app not just SQL processors

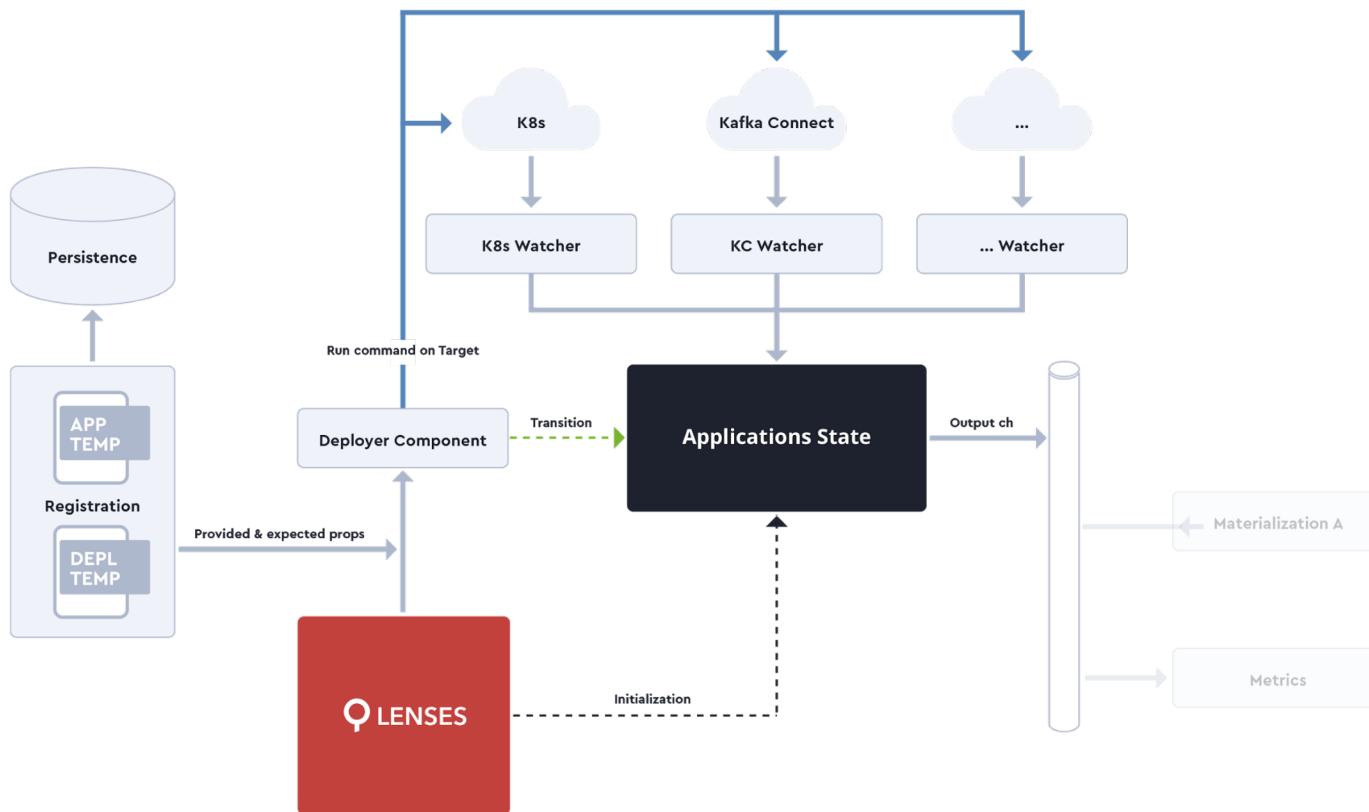


# Observability

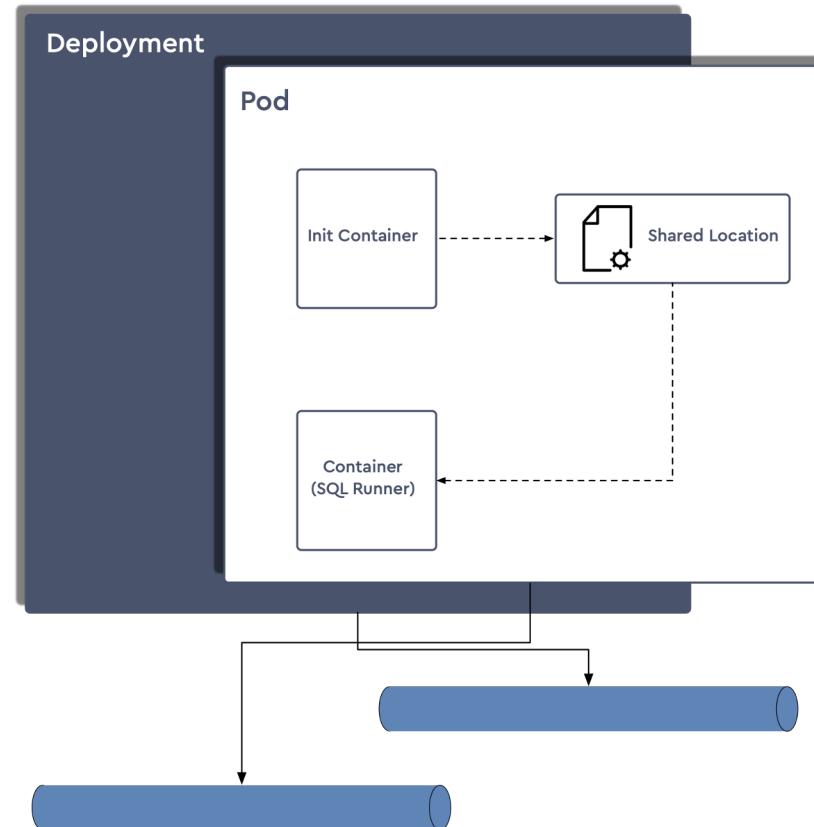
- ✓ We (Lenses) watch what's happening with the apps we deployed
- ✓ We can also expose through our templates ports where the application is already emitting more biz specific metrics



# Architecture Overview (draft, needs modifications)



# Kubernetes Write side (draft, needs modifications.)



# Dev Experience

- ✓ Import / Export applications  
(enable GitOps)
- ✓ Move apps without worrying  
about knowing backend service
- ✓ Promotion workflows eg. from  
Dev -> Prod
- ✓ Governance & security
- ✓ Streaming Apps marketplace
- ✓ Enable moving apps around  
environments by running  
Lenses-cli commands (or clicks  
if you are using our UI)

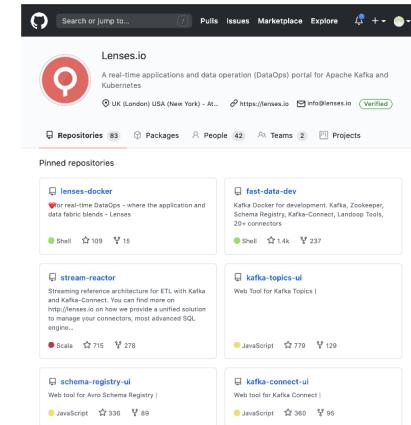
## [lenses.io/start](https://lenses.io/start)

All-in-one Kafka + Lenses.io developer Docker box



## [github.com/lensesio](https://github.com/lensesio)

Open-Source tooling & connectors



[launchpass.com/lensesio](https://launchpass.com/lensesio)



THANK  
YOU