

# How to build an internal developer platform on Kubernetes with Backstage, IaC, and GitOps



# Hello!

---

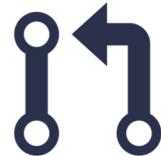
I am **Engin Diri**,

- Working at Pulumi as Customer Experience Architect
- Cloud Transformation and Enablement
- Continuous everything...





# Kingdon Barrett



OSS Engineer @ Weaveworks



Flux Maintainer @ [github.com/fluxcd](https://github.com/fluxcd)



[Kingdon @ weave.works](mailto:Kingdon@weave.works)

for support

email [↑](#) [←](#)  [❤️](#) CNCF slack @ KingdonB



**@yebyen@hachyderm.io** @yebyen

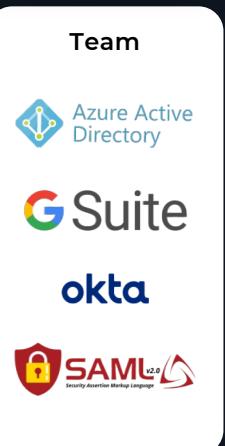
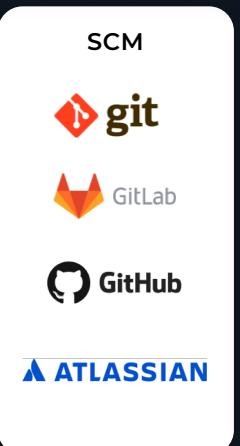
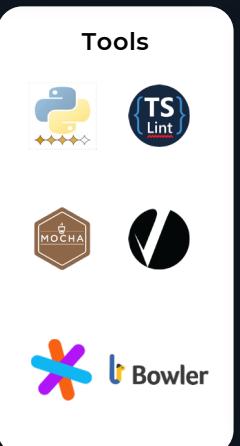
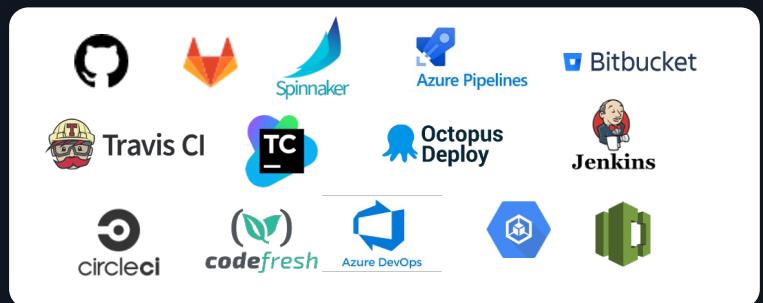
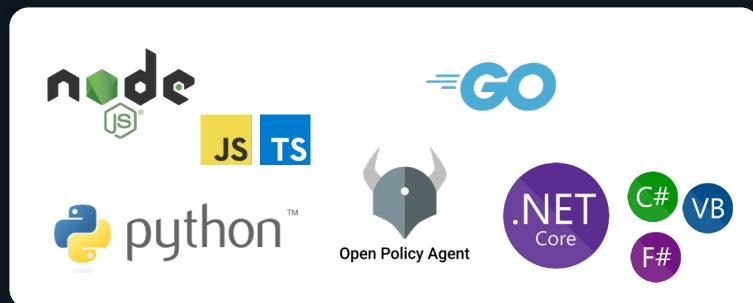
All computers, all the time

📍 41.6970, -86.2375 📅 Joined October 2007

1,225 Following 370 Followers

# Introduction

# Your Cloud, Your Language, Your Way



Build

Deploy

Manage

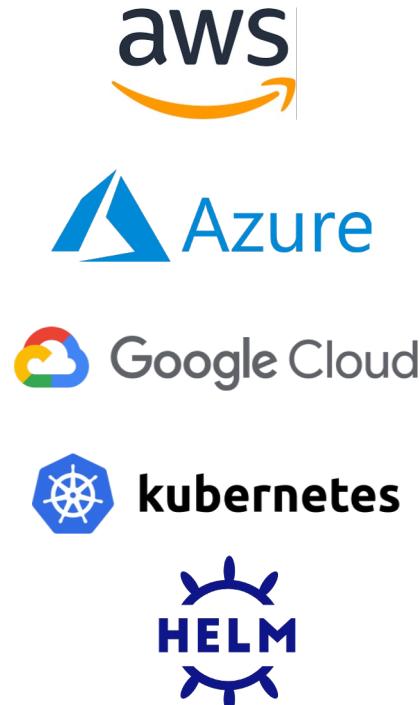
# Multi-Cloud Infrastructure as Code

BUILD, DEPLOY, AND MANAGE ANYWHERE WITH A STANDARD WORKFLOW

## Core Features

- Any Cloud, Any Language
- State Management
- Secrets Management
- Guaranteed Preview Plans
- CI/CD Integrations
- Webhooks
- REST API
- Automation API
- Dashboards and Reports

## Clouds



## Infra Providers



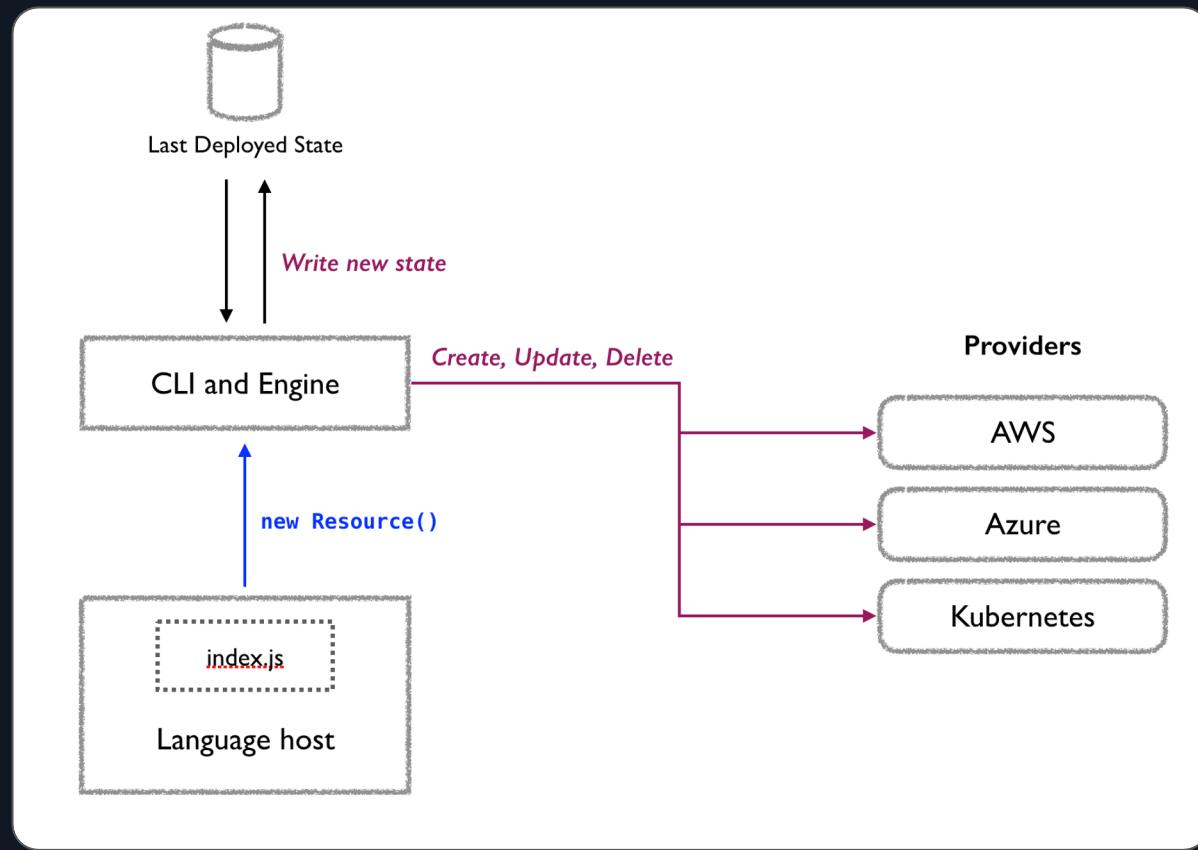
## Cloud Native



# Pulumi High Level View

# Pulumi - High Level View

## Pulumi Architecture

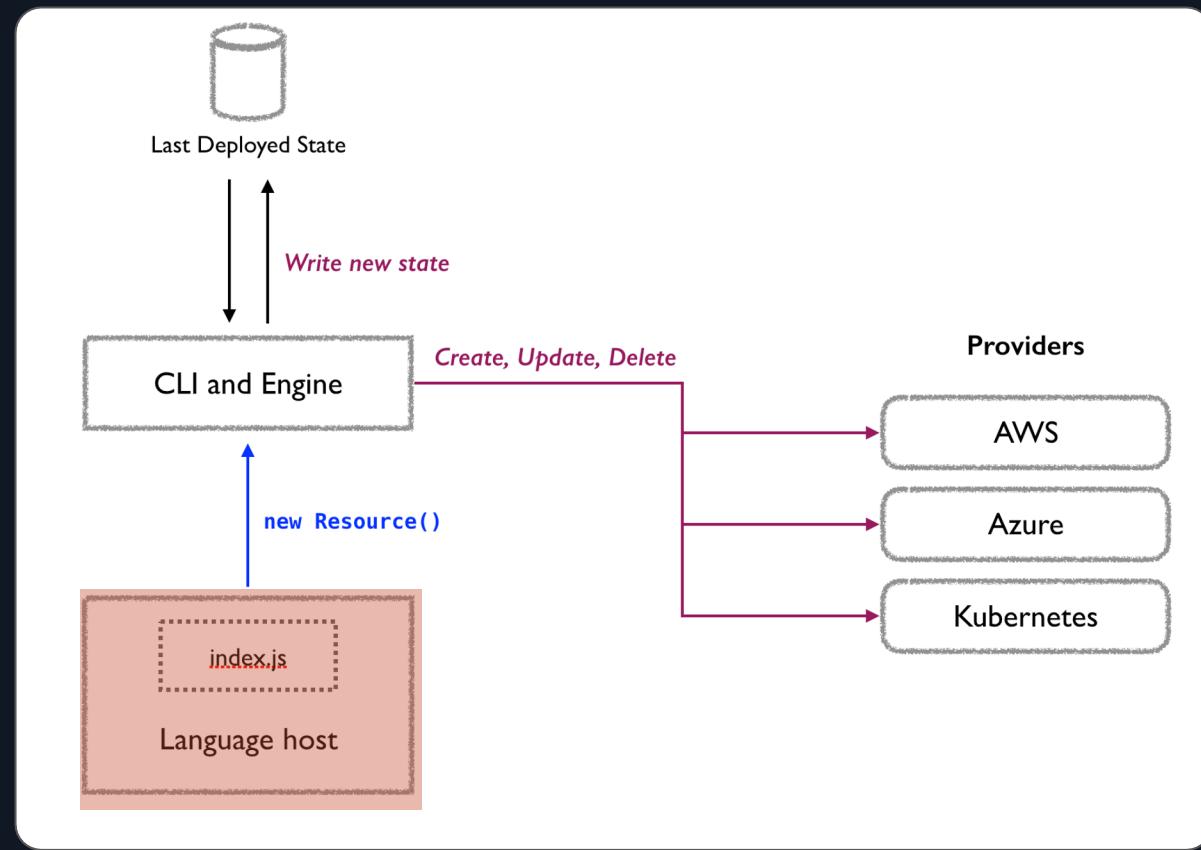


# Pulumi - High Level View

---

## Pulumi Architecture

- Language Host

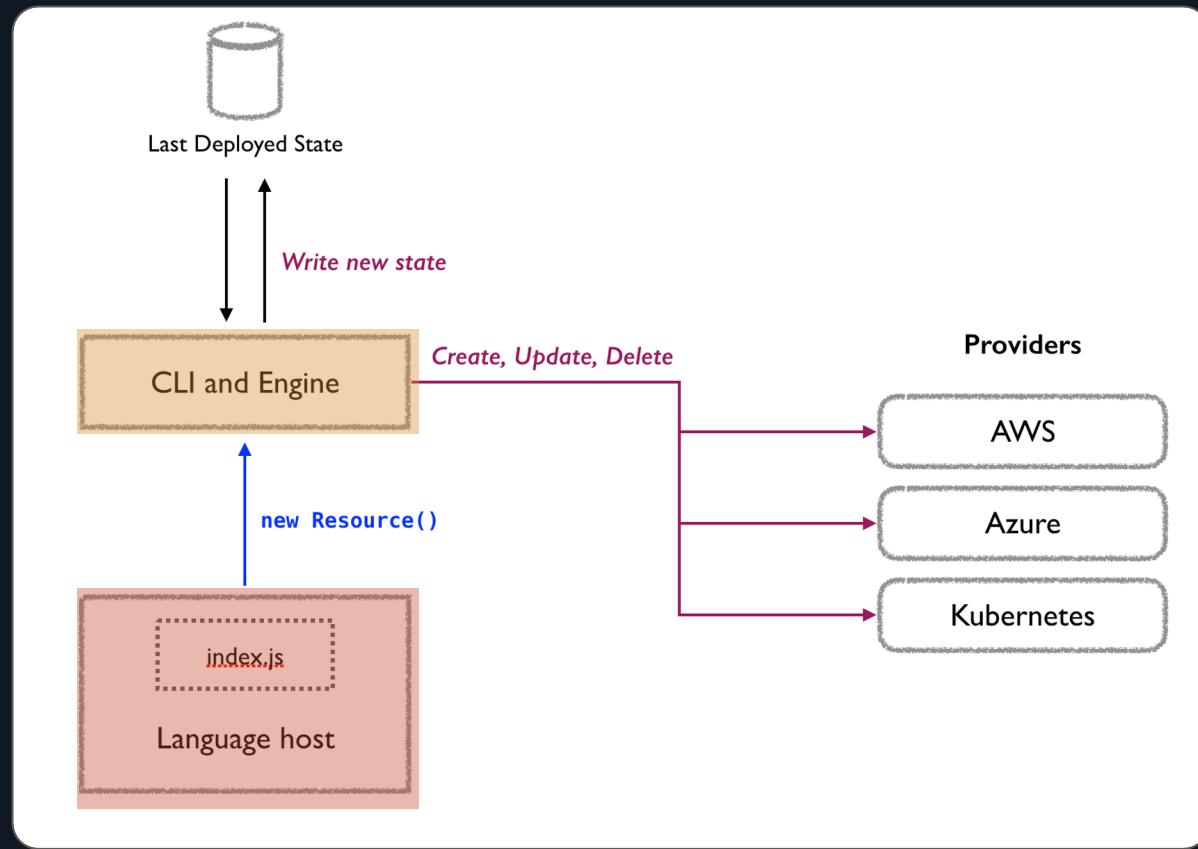


# Pulumi - High Level View

---

## Pulumi Architecture

- Language Host
- CLI & Engine
- Providers
- State Backend
  - Pulumi Service
  - Cloud Storage (S3, ...)

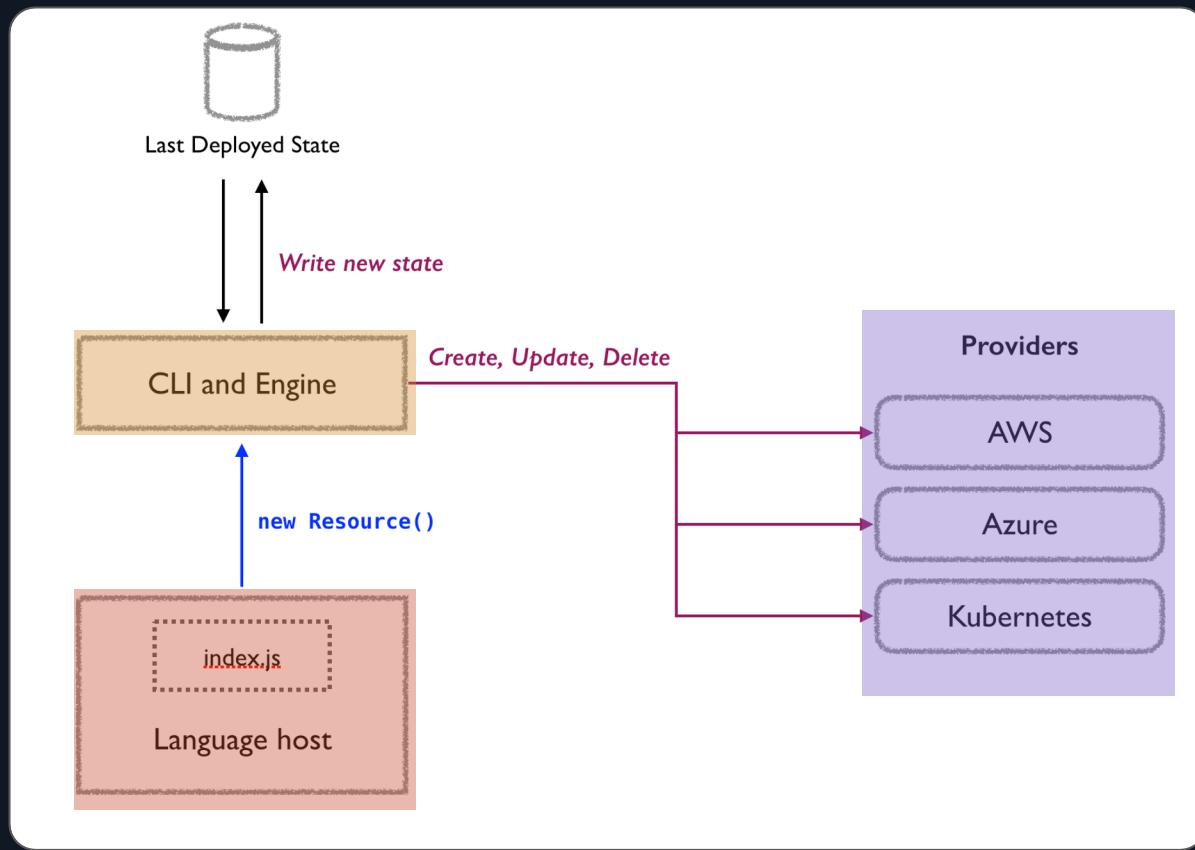


# Pulumi - High Level View

---

## Pulumi Architecture

- Language Host
- CLI & Engine
- Providers
- State Backend
  - Pulumi Service
  - Cloud Storage (S3, ...)

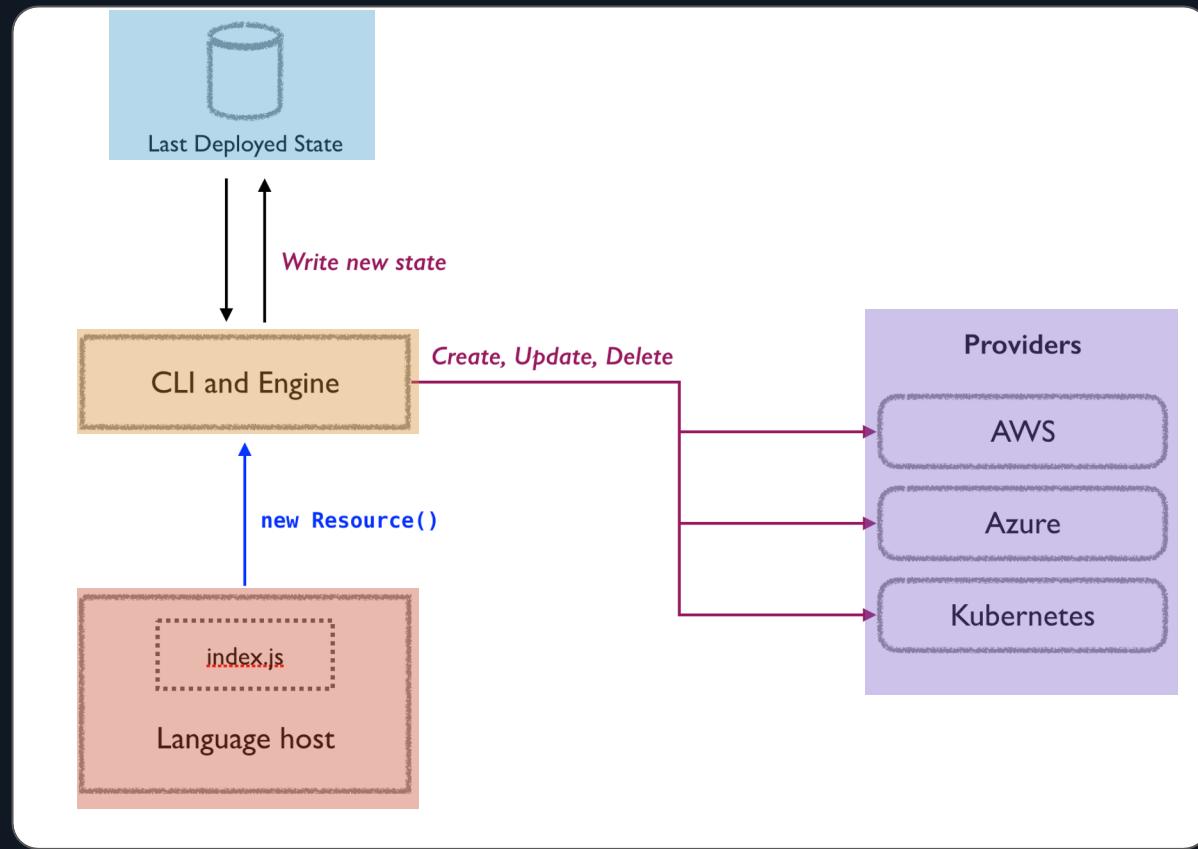


# Pulumi - High Level View

---

## Pulumi Architecture

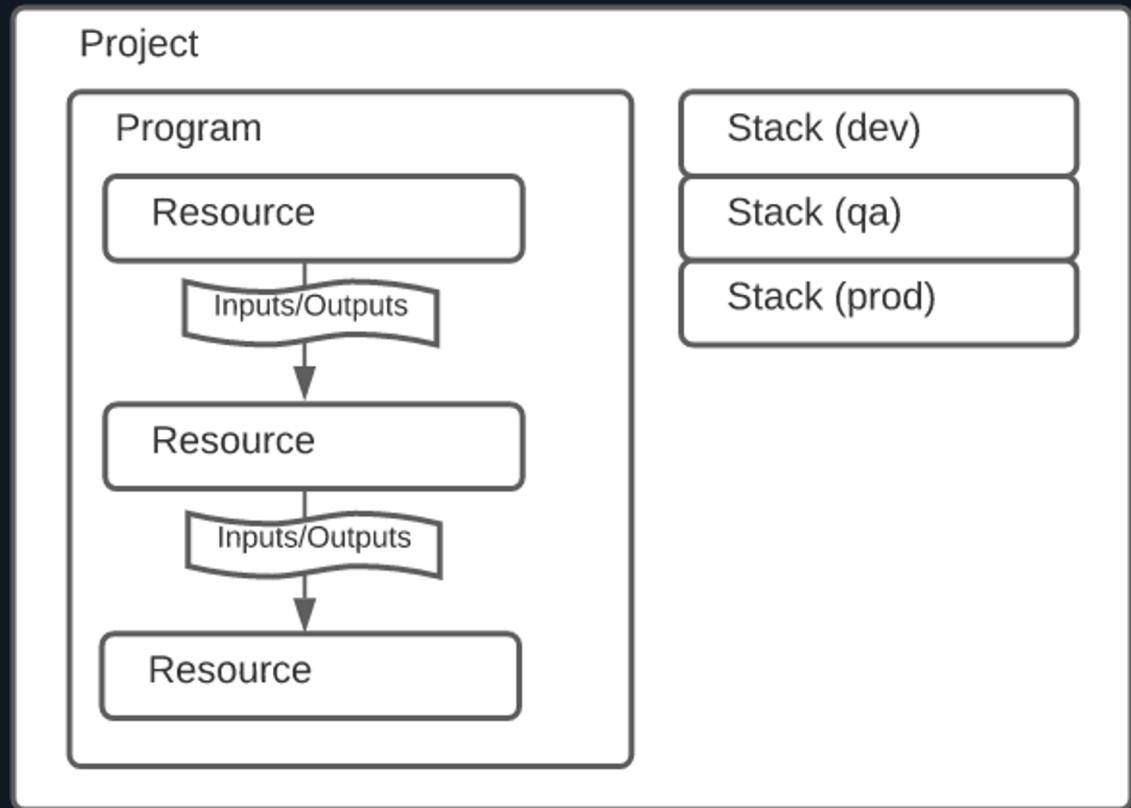
- Language Host
- CLI & Engine
- Providers
- State Backend
  - Pulumi Service
  - Cloud Storage (S3, ...)



# Pulumi - High Level View

---

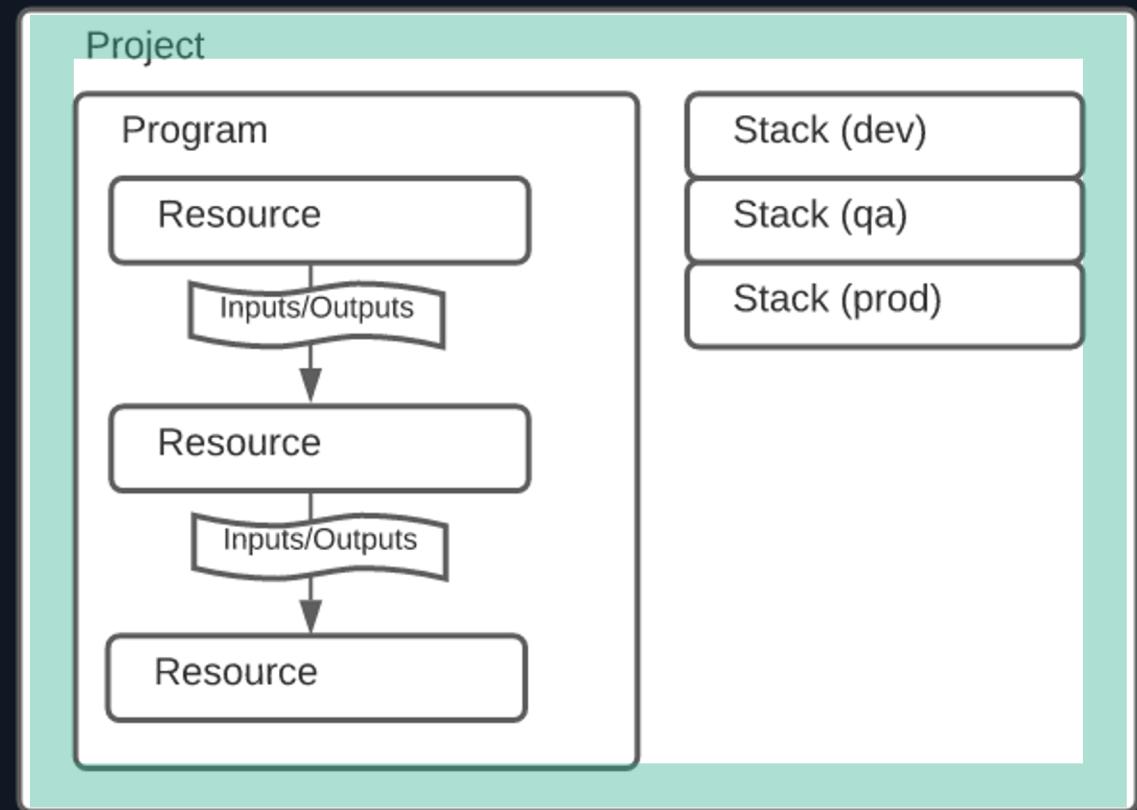
- Automate your infrastructure setup with one or more Pulumi projects



# Pulumi - High Level View

---

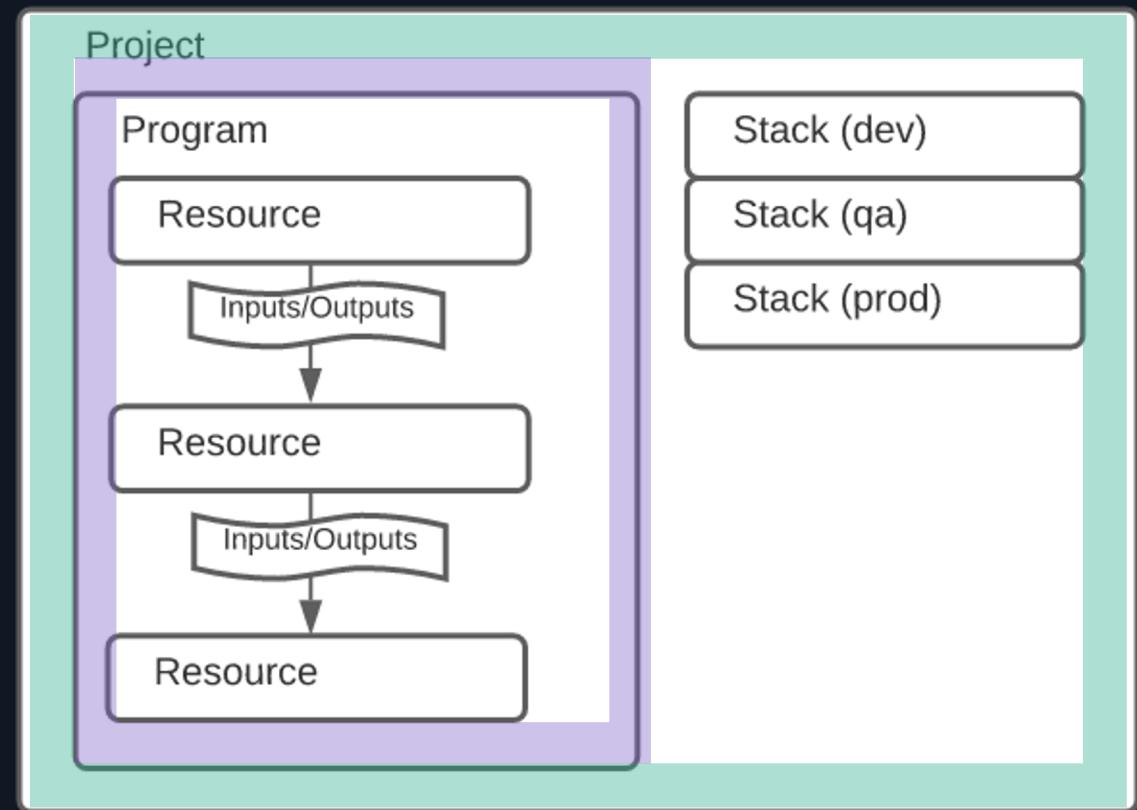
- Automate your infrastructure setup with one or more Pulumi projects
- A **project** contains:



# Pulumi - High Level View

---

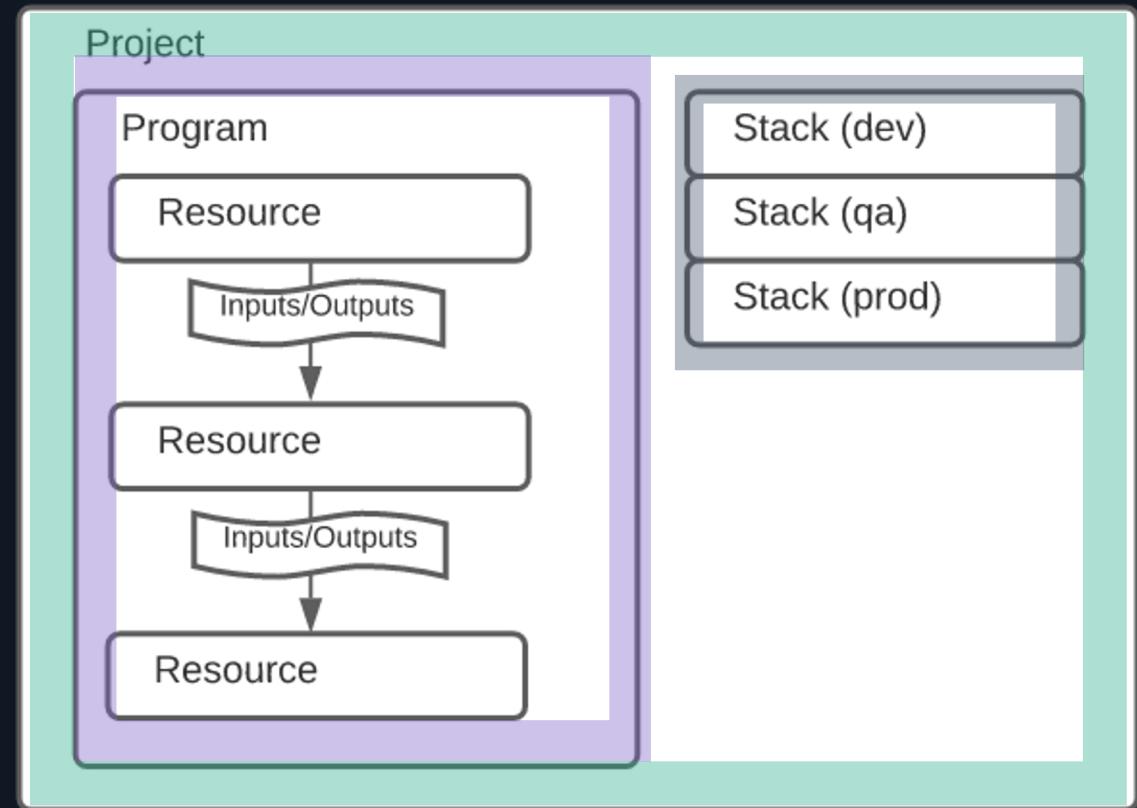
- Automate your infrastructure setup with one or more Pulumi projects
- A **project** contains:
  - a **program**: this is your “template” from which you can create stacks



# Pulumi - High Level View

---

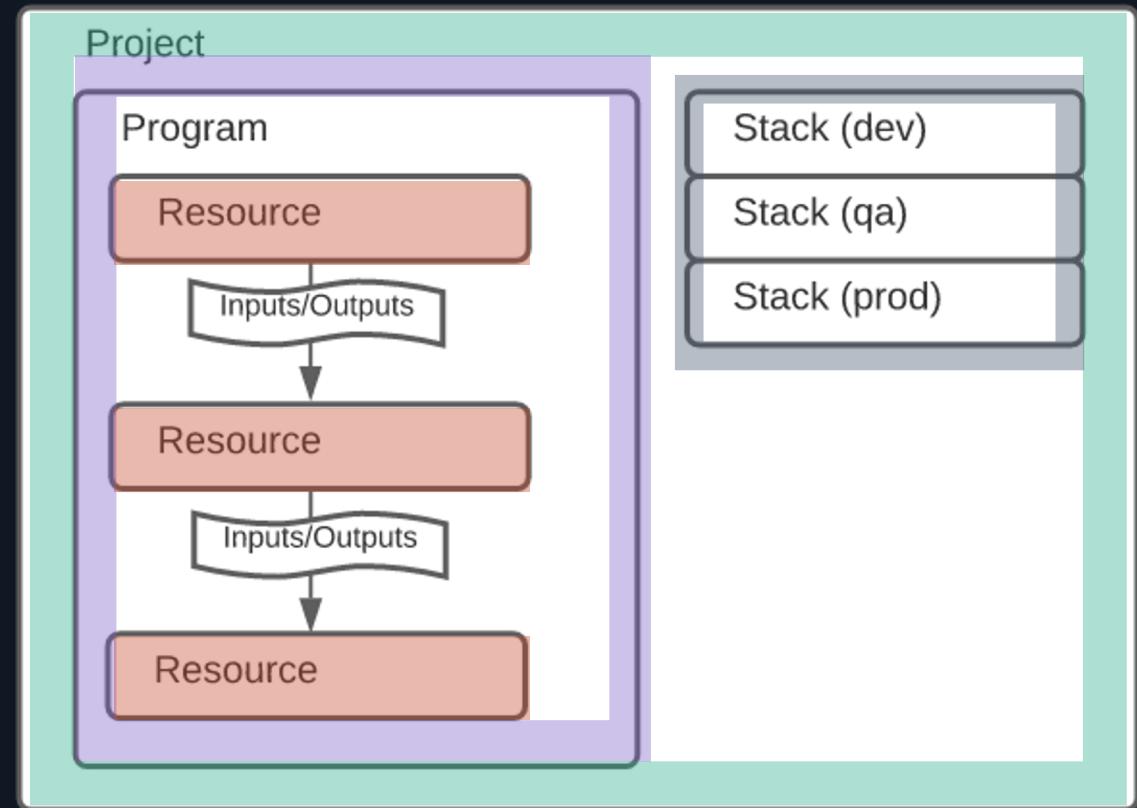
- Automate your infrastructure setup with one or more Pulumi projects
- A **project** contains:
  - a **program**: this is your “template” from which you can create stacks
  - the **stacks**: each stack is an “instance” of the program



# Pulumi - High Level View

---

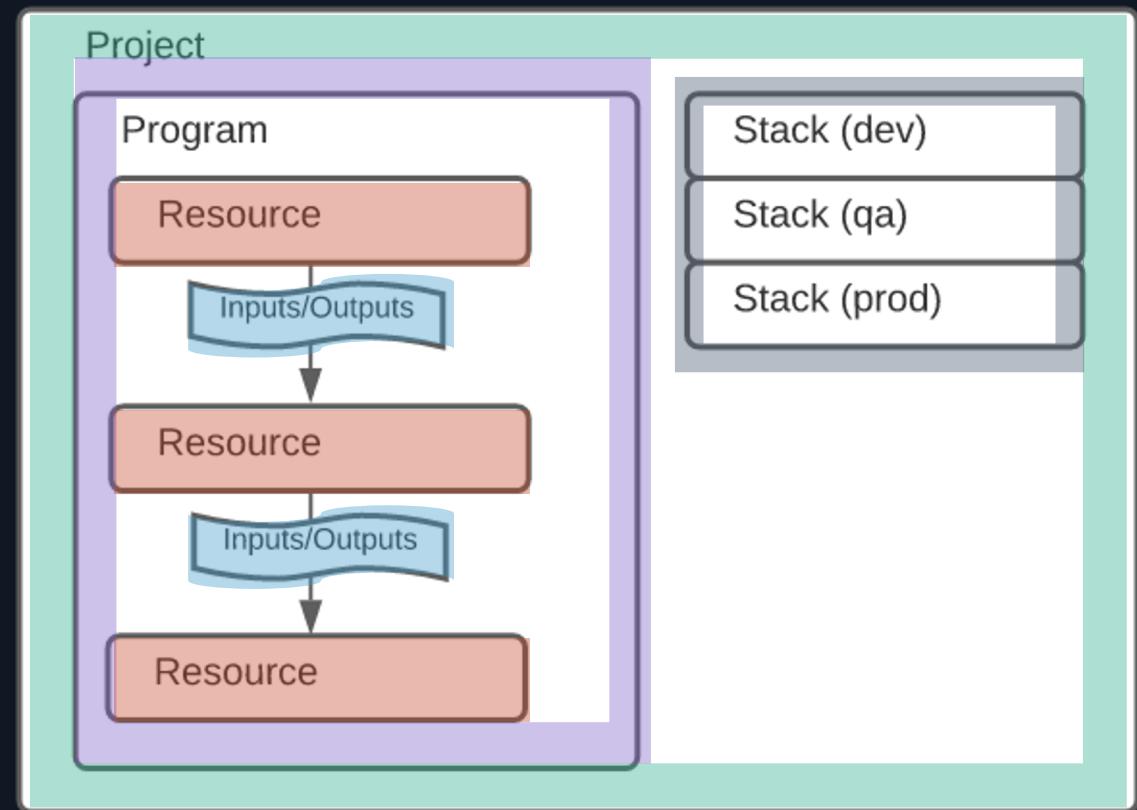
- Automate your infrastructure setup with one or more Pulumi projects
- A **project** contains:
  - a **program**: this is your “template” from which you can create stacks
  - the **stacks**: each stack is an “instance” of the program
  - from **resources**: smallest building block



# Pulumi - High Level View

---

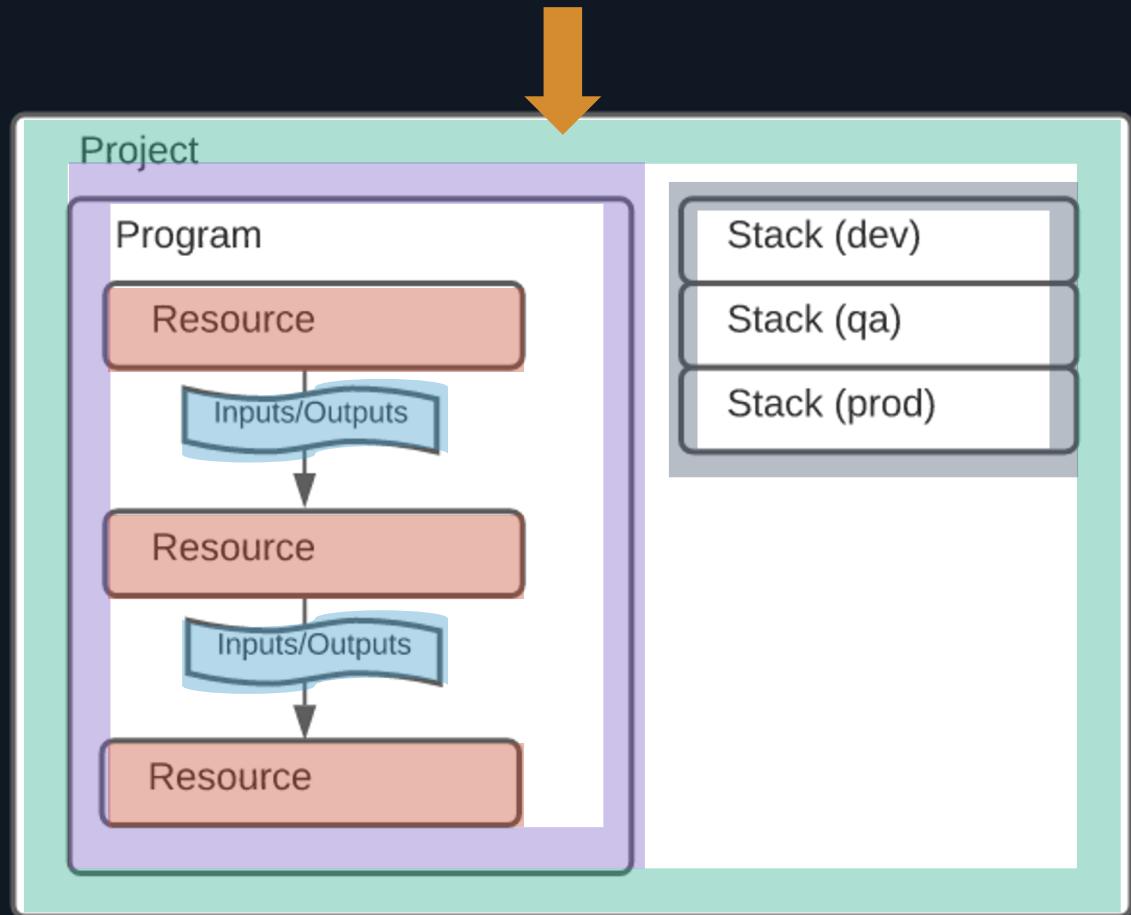
- Automate your infrastructure setup with one or more Pulumi projects
- A **project** contains:
  - a **program**: this is your “template” from which you can create stacks
  - the **stacks**: each stack is an “instance” of the program
  - from **resources**: smallest building block
  - have **inputs/outputs**: read or set values for your **resource**



# Pulumi - High Level View

---

- Automate your infrastructure setup with one or more Pulumi projects
- A **project** contains:
  - a **program**: this is your “template” from which you can create stacks
  - the **stacks**: each stack is an “instance” of the program
  - from **resources**: smallest building block
  - have **inputs/outputs**: read or set values for your **resource**
- One project can use **info** from another project as input





flux



# Flux in 2 Minutes



# GitOps

- For Beginners: (ELI5)
- Developer just pushes changes
- (**new desired state**) to Git
- They go automatically to **Kubernetes**  
(User Experience should be simple)



# GitOps

- For Beginners: (ELI5)
- **Developer** just pushes changes
- (**new desired state**) to **Git**
- ...notify us if everything went alright  
(User Experience should be simple)



# GitOps

- Developer just pushes changes to **Git**
- ...in case anything changes, <-> **GitOps** reconciling (**Flux**) will automatically put it back  
This is called **Drift Correction**

Dist. from regular imperative CI/CD



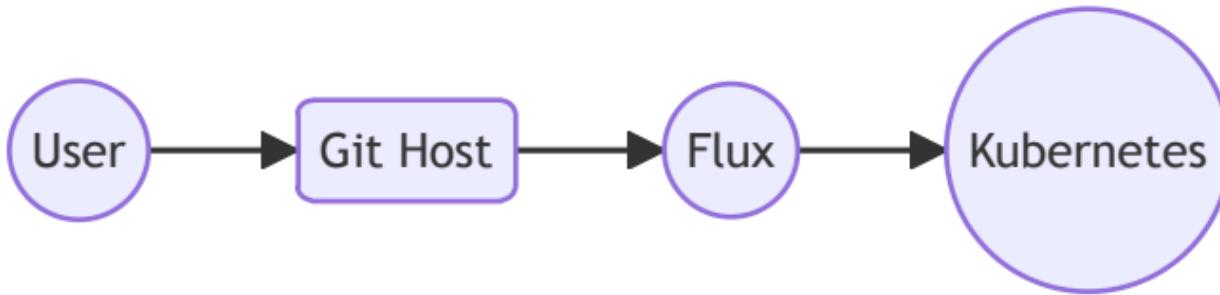
# 4 Principles of GitOps

1. **Declarative** Configuration
2. **Versioned** Storage
3. Pulled **automatically**
4. Reconciled **continuously**

Detail: [opengitops.dev](https://opengitops.dev) from GitOps WG



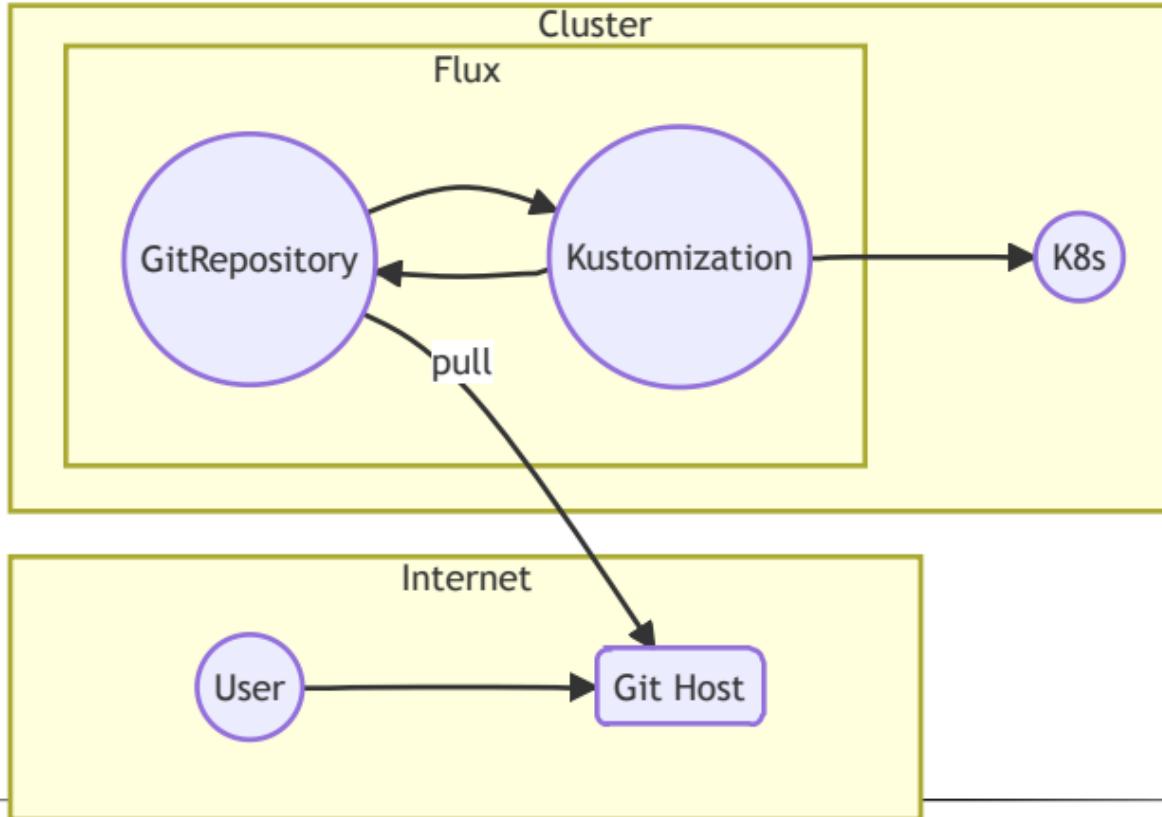
# Diagram



## Easy Version

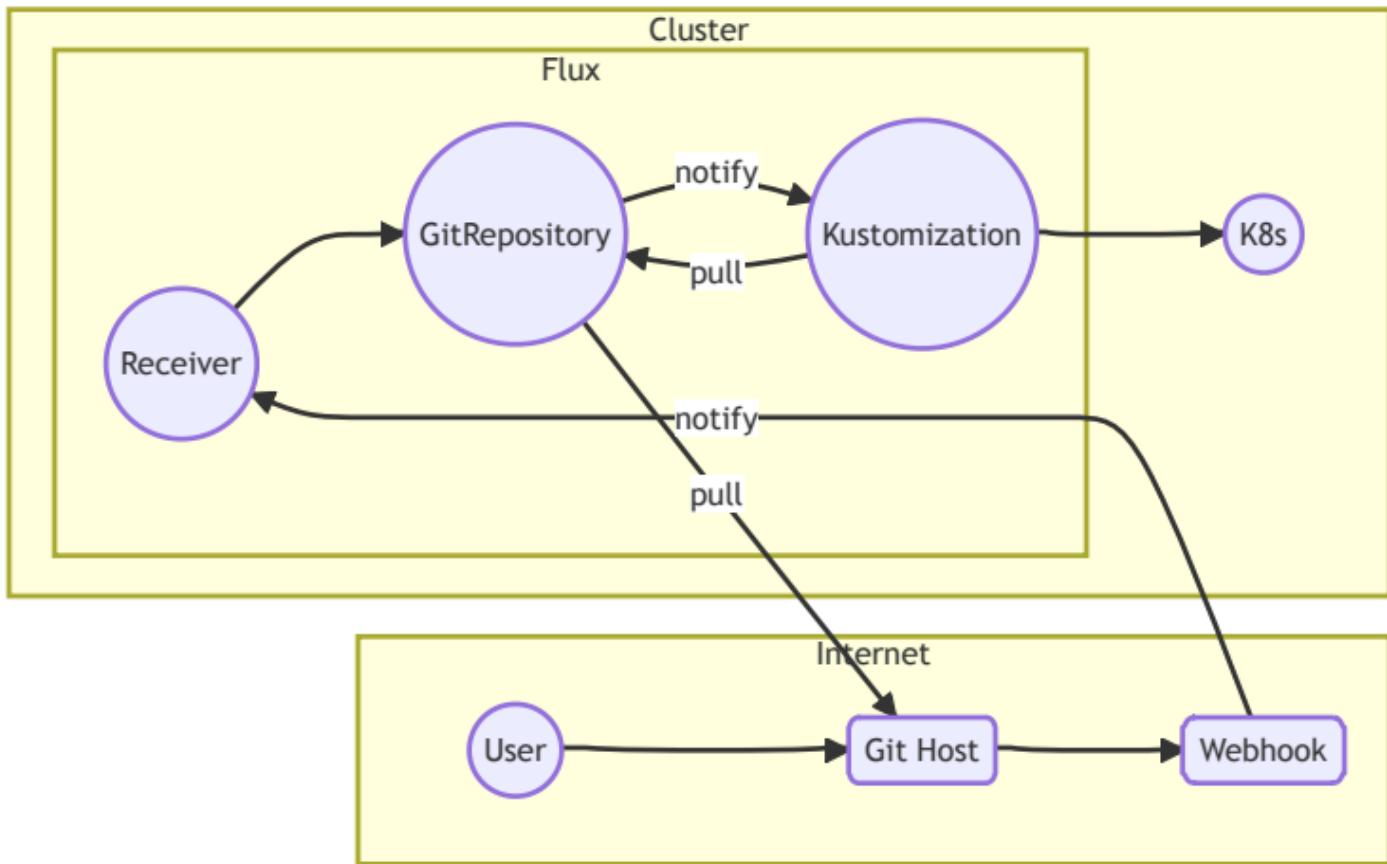


# How Flux does GitOps





# **Flux is hosted on a K8s cluster**





# **Subscriptions in K8s are automatic**



# Subscriptions in K8s are negotiated automatically

- On the other hand, across the internet (GitHub, GitLab, Gitea, CodeCommit, ...)
- Webhooks typically need configured



# Flux is Community

- Flux Developed in open under CNCF
- **Weekly Dev Meetings** - Public (Zoom)
- Transparent RFCs, Public Roadmap

[fluxcd.io/#calendar](https://fluxcd.io/#calendar)



# Flux is Supported



👋 Kingdon Barrett



OSS Engineer @ Weaveworks



Flux Maintainer @ [github.com/fluxcd](https://github.com/fluxcd)



[Kingdon @ weave.works](https://weave.works) for support

email CNCF slack @ KingdonB



@yebyen@hachyderm.io

All computers, all the time

⌚ 41.6970, -86.2375 📅 Joined October 2007

1,225 Following 370 Followers



# Flux Manages Itself



# GitOps

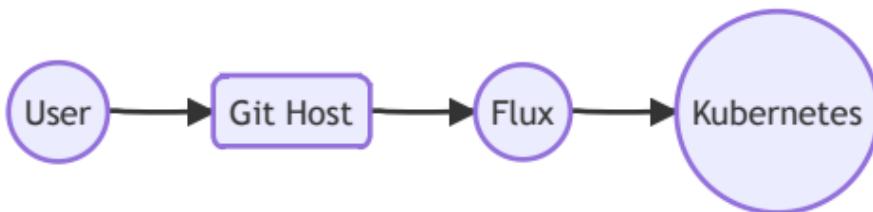
If this is GitOps...





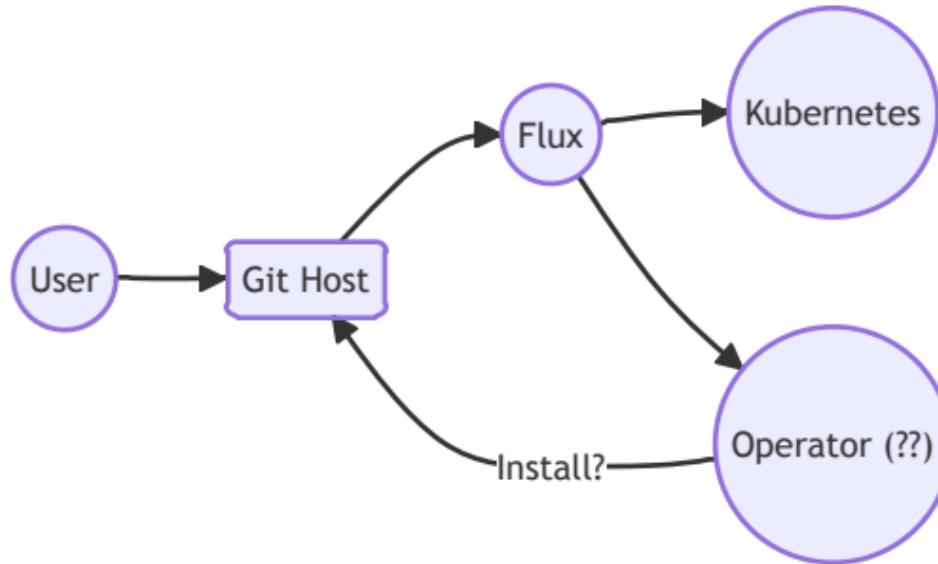
# Flux does GitOps

And Flux does GitOps...





# How does Flux get installed?





# How does Flux get installed?

??? That's up to you

- ?: copy Flux into Git
- ^- Typically: `flux bootstrap` does this



# What is Bootstrap?



# Bootstrap Decomposed

- clusters/my-cluster/
  - flux-system/
    - gotk-components - Flux itself (controllers)
    - **gotk-sync** - Business end of Flux (**config**)

K8s CRs GitRepo and Kustomization represent a running Flux installation



# Bootstrap Decomposed

- Bootstrap also sets up some secrets
- Repository secrets: usually SSH key
- Read-only by default, probably private
- (but needs write access to bootstrap)



# How does Flux get installed?

- ?: copy Flux controllers into **Git? no**
- Other possibility: Flux is **managed by** GitOps only some other way ... **how?**

Another **Operator** can install Flux!



# Flux as OCI Artifact

You do not need to copy Flux into Git

- (controllers **released** as OCI artifact)

```
oci://ghcr.io/fluxcd/  
  flux-manifests:v2.1.0
```

- OCIRepository is another Flux **source kind** - more about that later



# Demo



Backstage

# Backstage

---

Backstage is an open platform for building developer portals.

Created at



Donated to



CLOUD NATIVE  
COMPUTING FOUNDATION

# Backstage

---

**A developer portal = one frontend for your entire infrastructure**

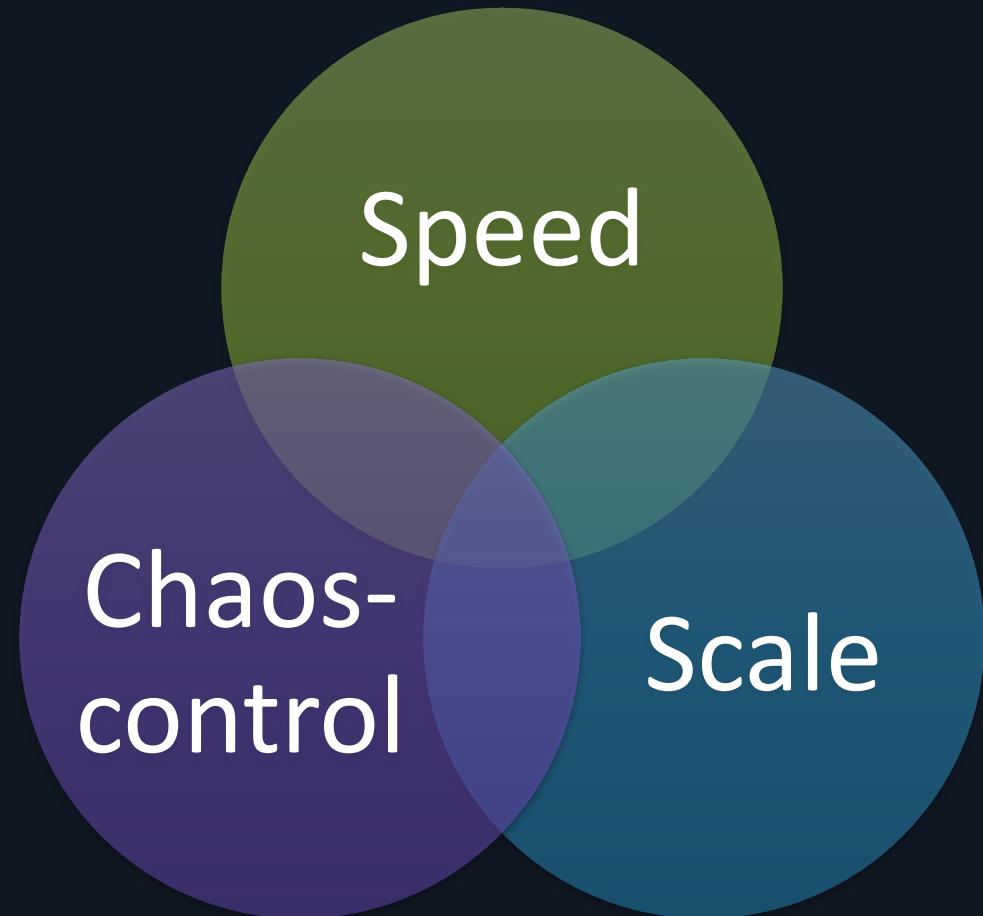
Unifies all your tooling, services, apps, data, and docs with a single, consistent UI

Makes sense of everything in your ecosystem, regardless of how and where individual components are running

Let developers focus on what they do best (leading to much less activity in #aaargh Slack channel)

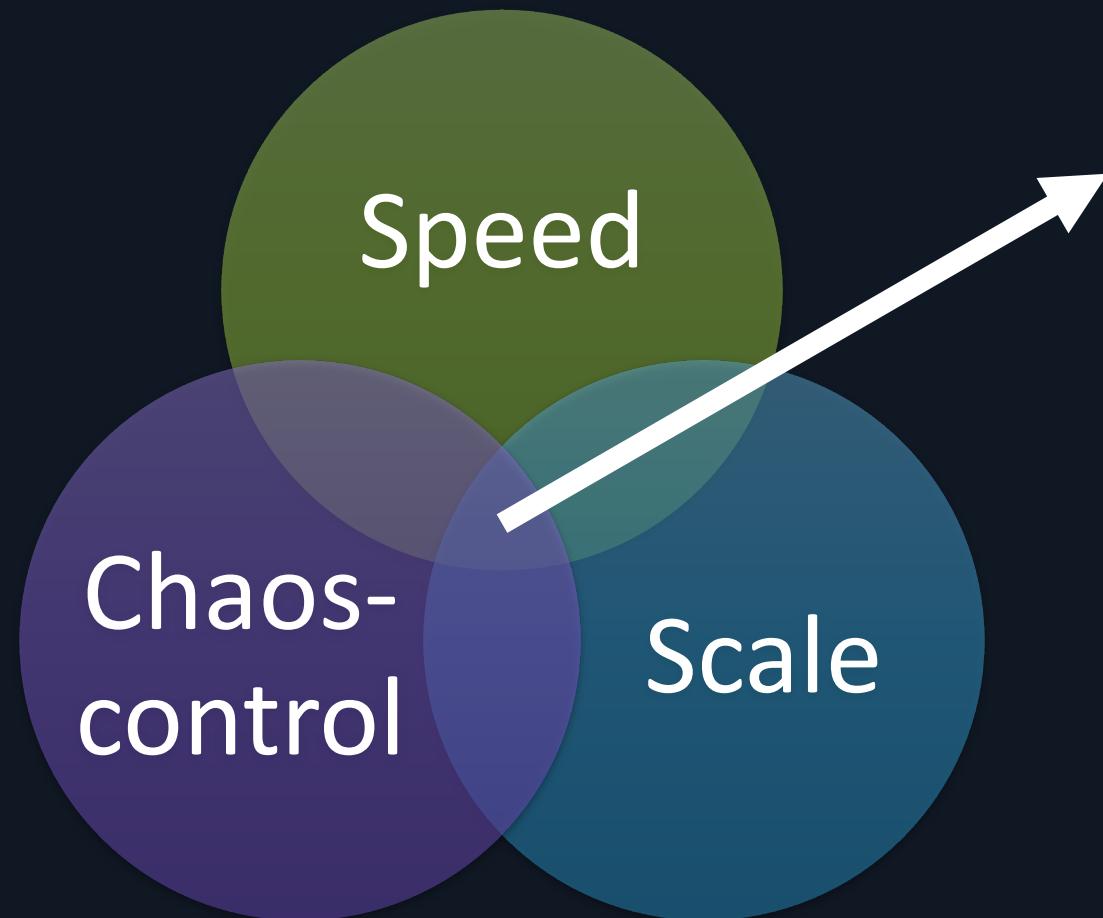
# Backstage + Pulumi + Flux

---



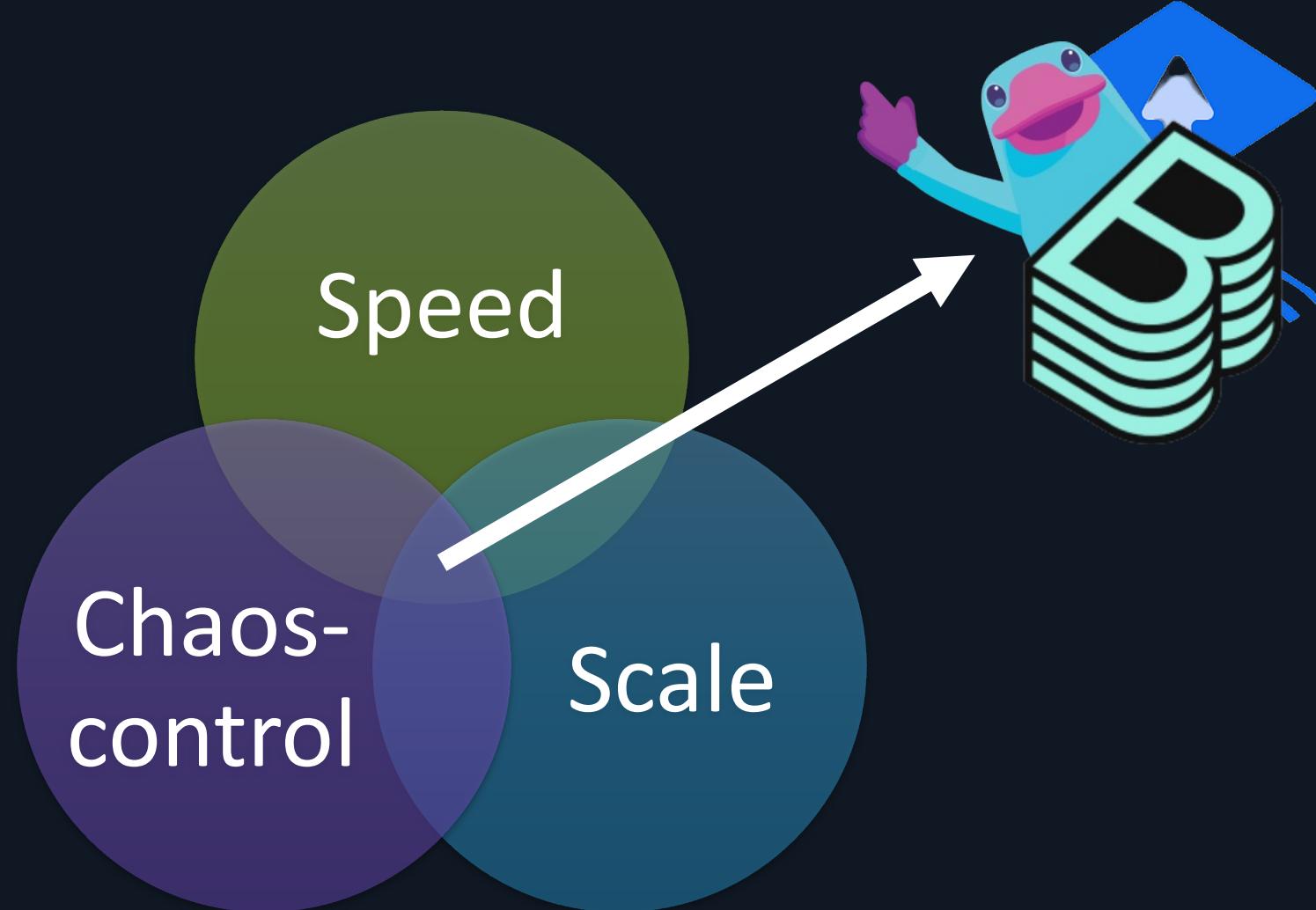
# Backstage + Pulumi + Flux

---



# Backstage + Pulumi + Flux

---



# Backstage

---

Backstage lets any developer:

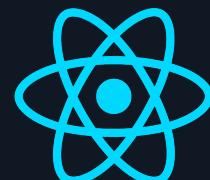
- **Create** new software in seconds, aligned to your best practices
- **Manage** all the software they own in one centralized location
- **Explore** the entire software ecosystem, enabling collaboration across your org

# Backstage

---

Backstage has a customizable, extensible plugin architecture

- Built with modern technologies and common frameworks.
- Makes it easy to develop for and contribute to your dev portal.
- Cloud-agnostic and vendor-neutral.



Demo time!

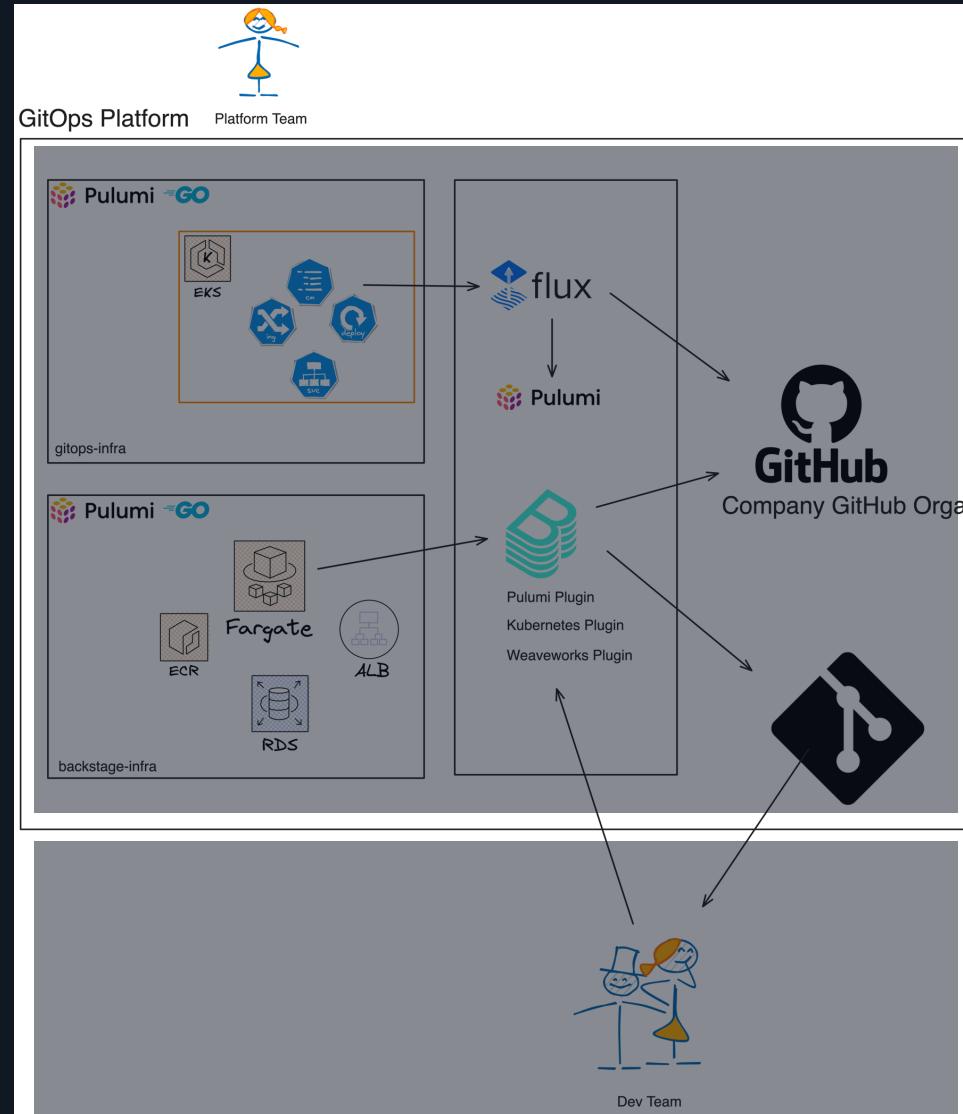
# Demo Time!

---

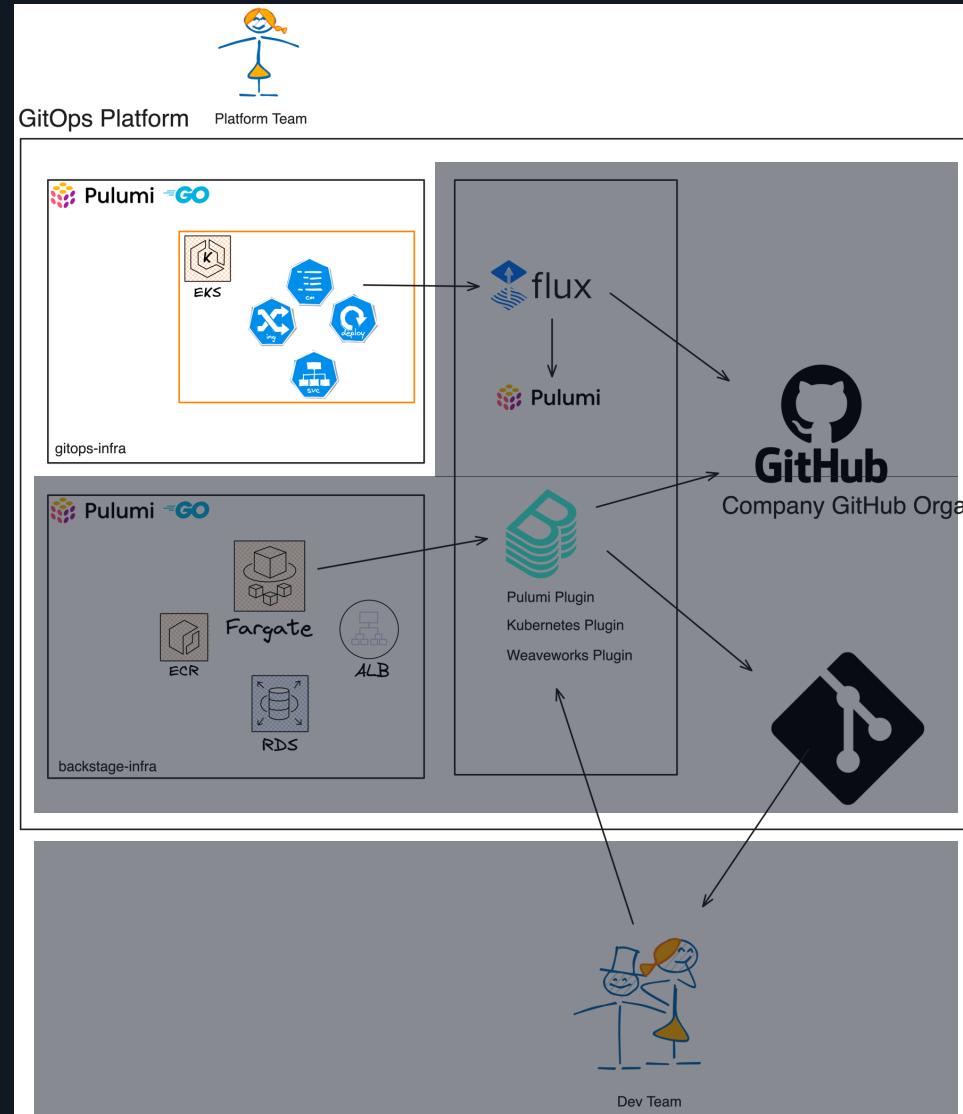


**Go + AWS + EKS**

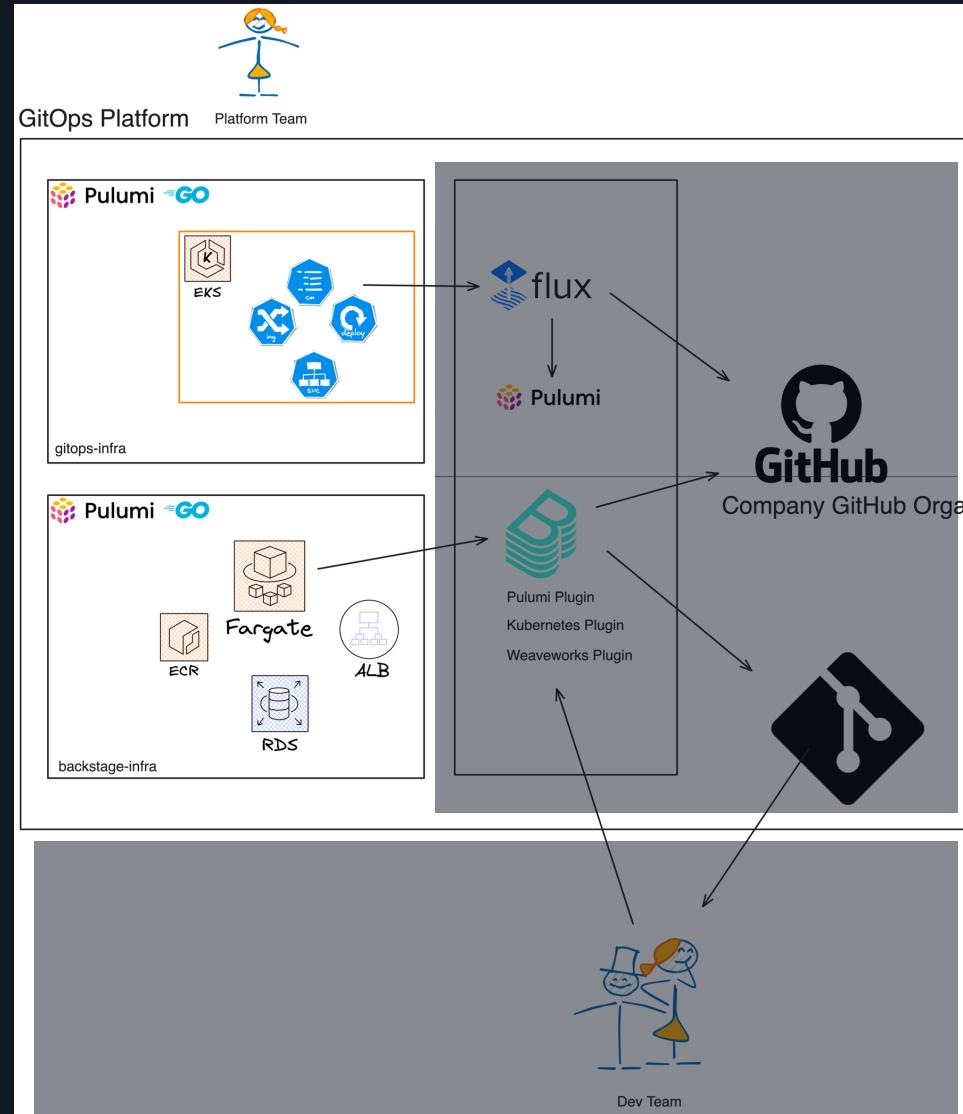
# Demo Architecture



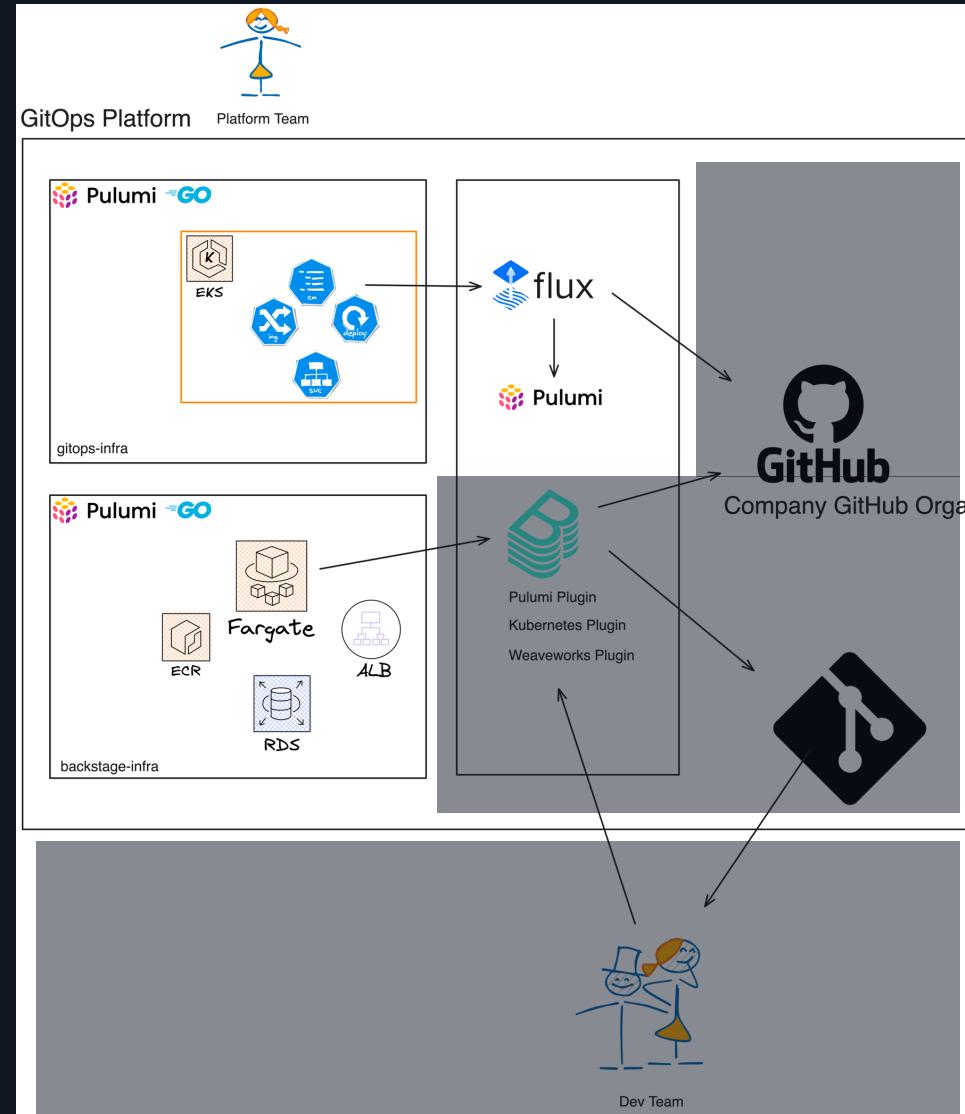
# Demo Architecture



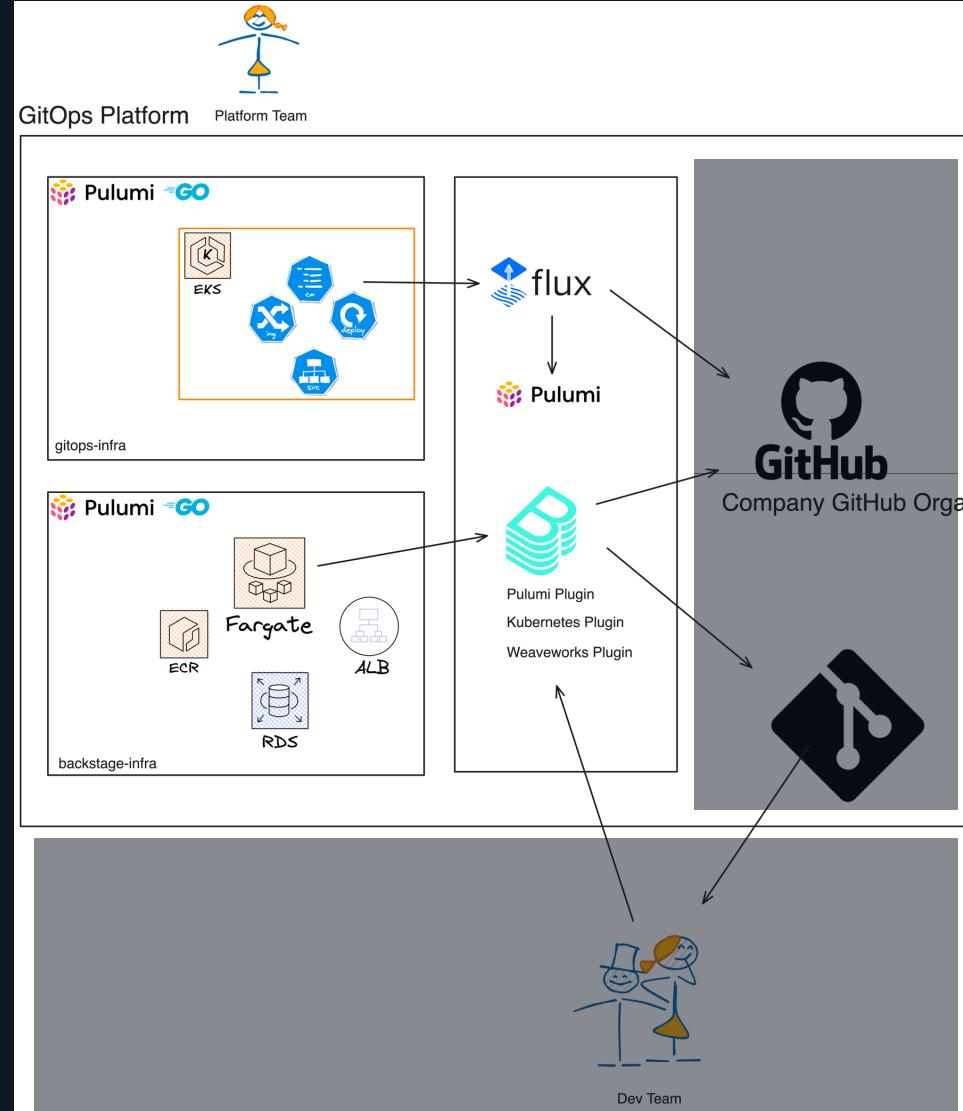
# Demo Architecture



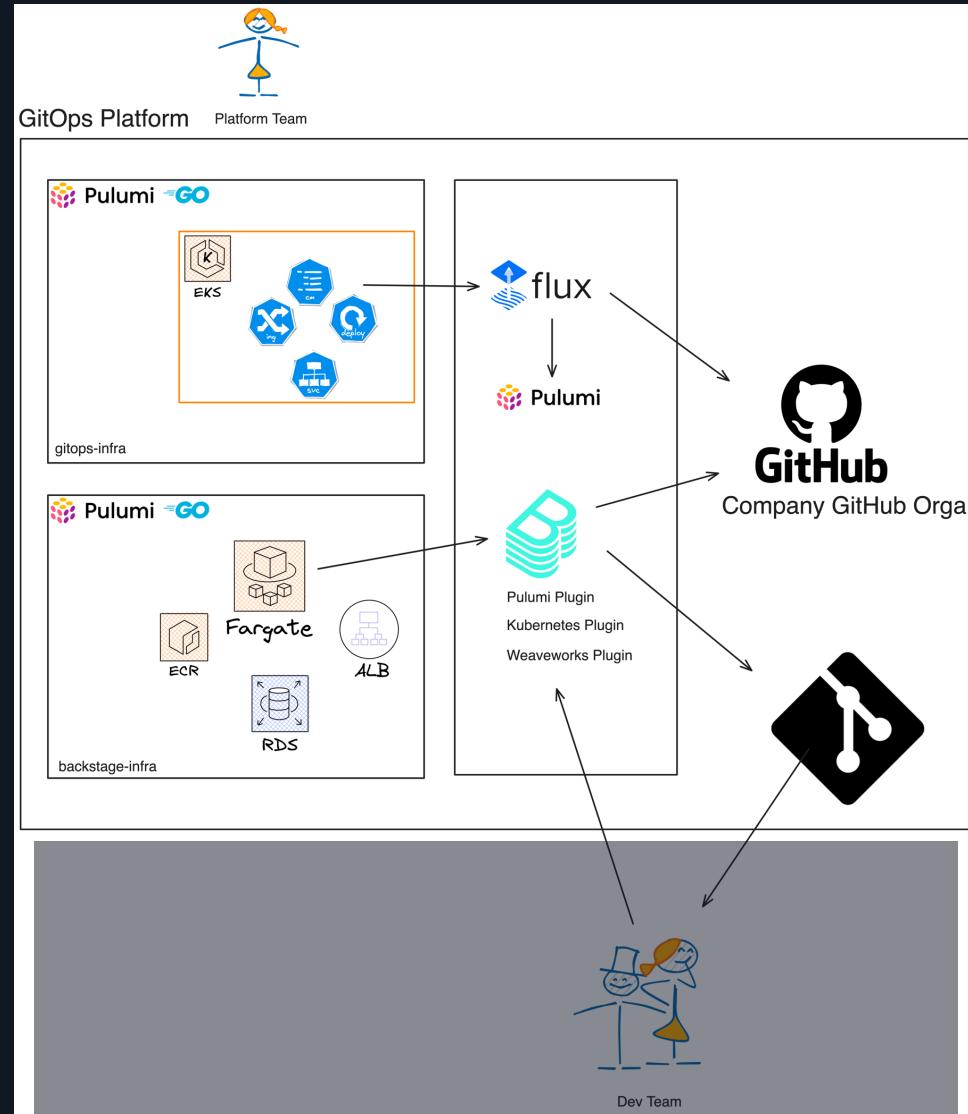
# Demo Architecture



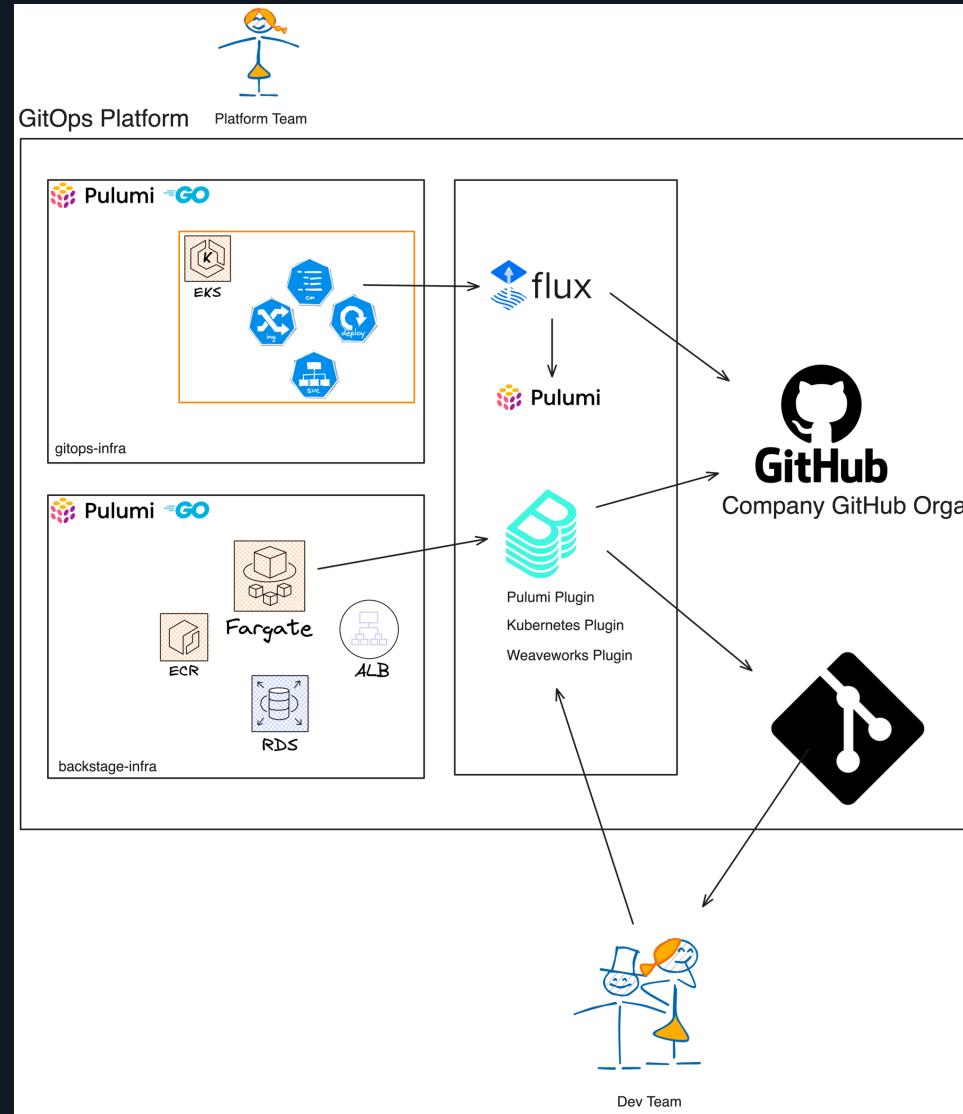
# Demo Architecture



# Demo Architecture



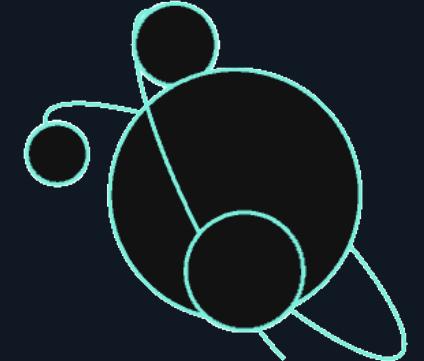
# Demo Architecture



# Demo!

# Follow up

---



- <https://backstage.io/>
- <https://www.weave.works/>
- <https://www.pulumi.com/>
- <https://fluxcd.io/>
- <https://github.com/my-backstage-demo/pulumi-backstage-flux-gitops-aws>