

What and why PipeCD for Hybrid and Multi-Cloud Applications Deployments



KubeCon



CloudNativeCon

North America 2023

khanhtc1202 @ KubeCon NA 2023

About me



[@khanhtc1202](#)



[@khanhtc1202](#)



Software Engineer

Khanh Tran

A new dad who is experiencing a lack of sleep



Let's take a break, we have some questions :)

Why progressive delivery / deployment?

Why progressive delivery / deployment?



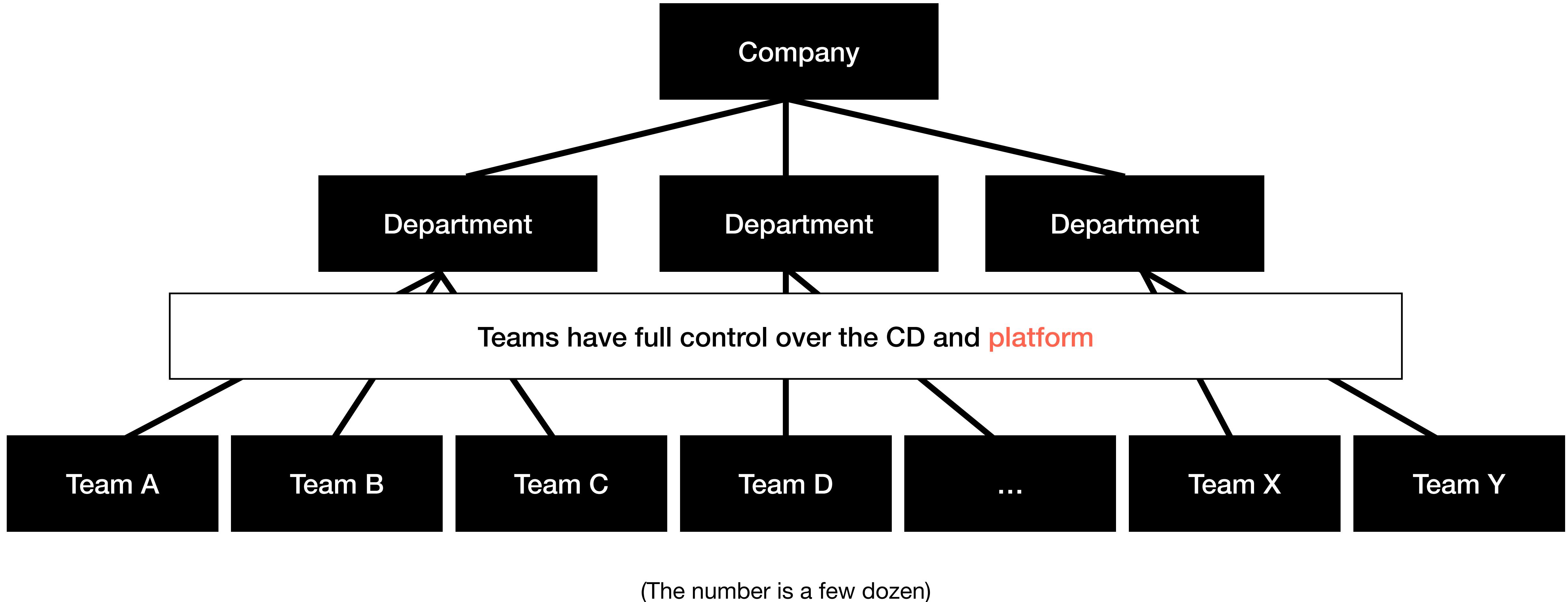
To avoid this end :)

Why multi-cloud environment?

Why multi-cloud environment?

- Diversity teams / background
- Service requires
- Legacy!

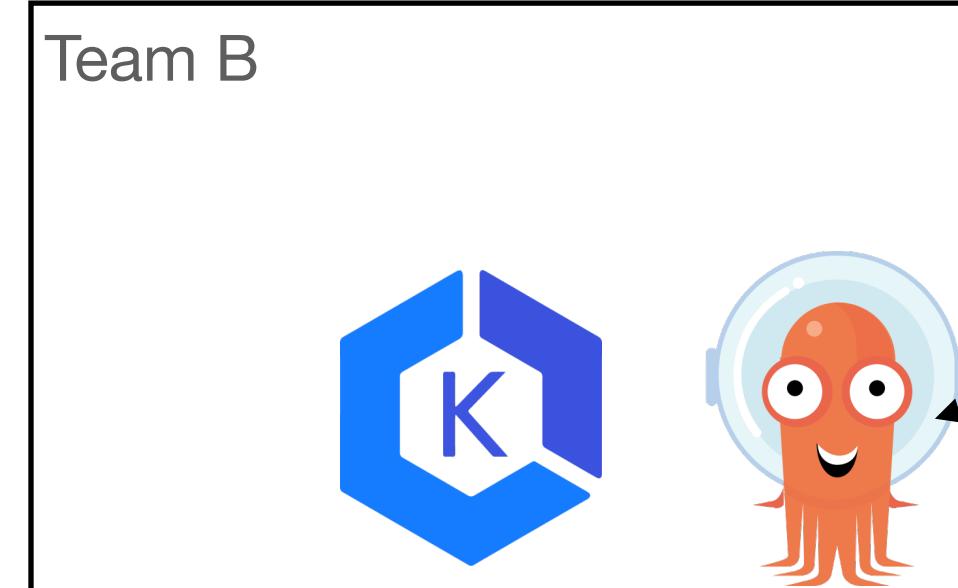
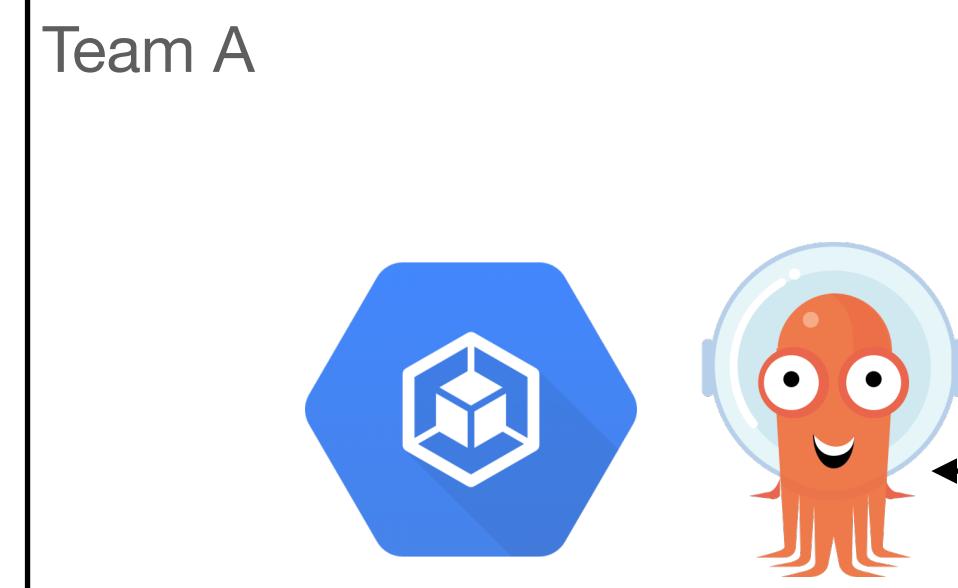
Why multi-cloud environment?



Any problem with that environment?

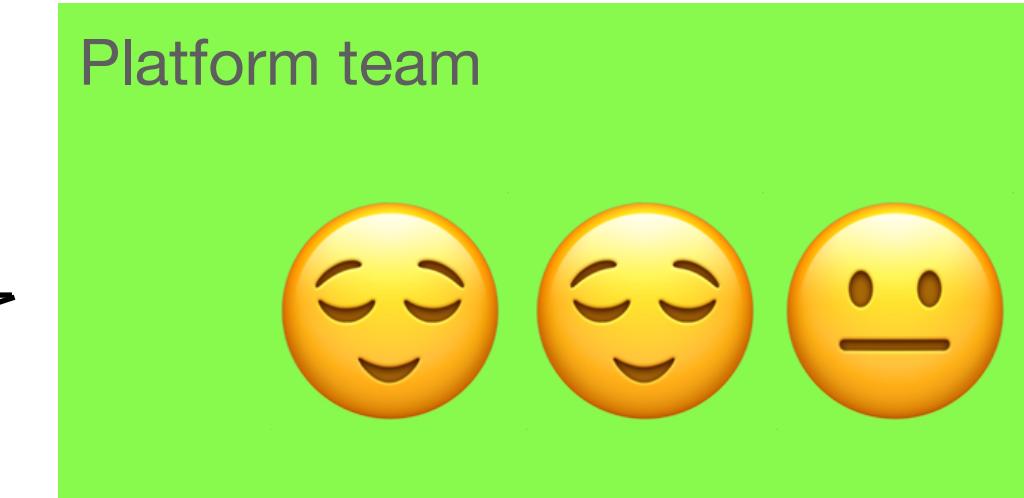
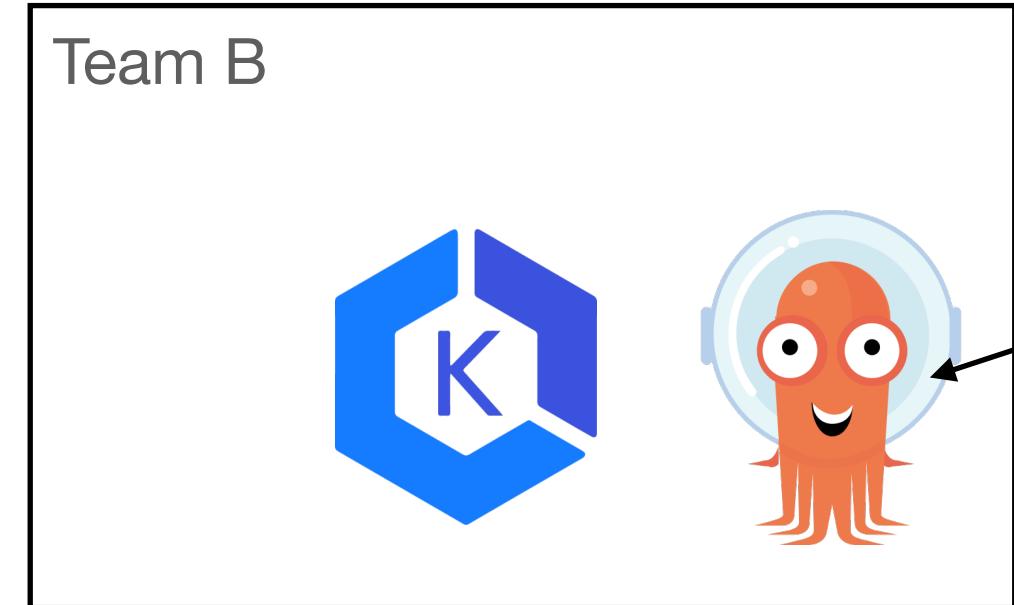
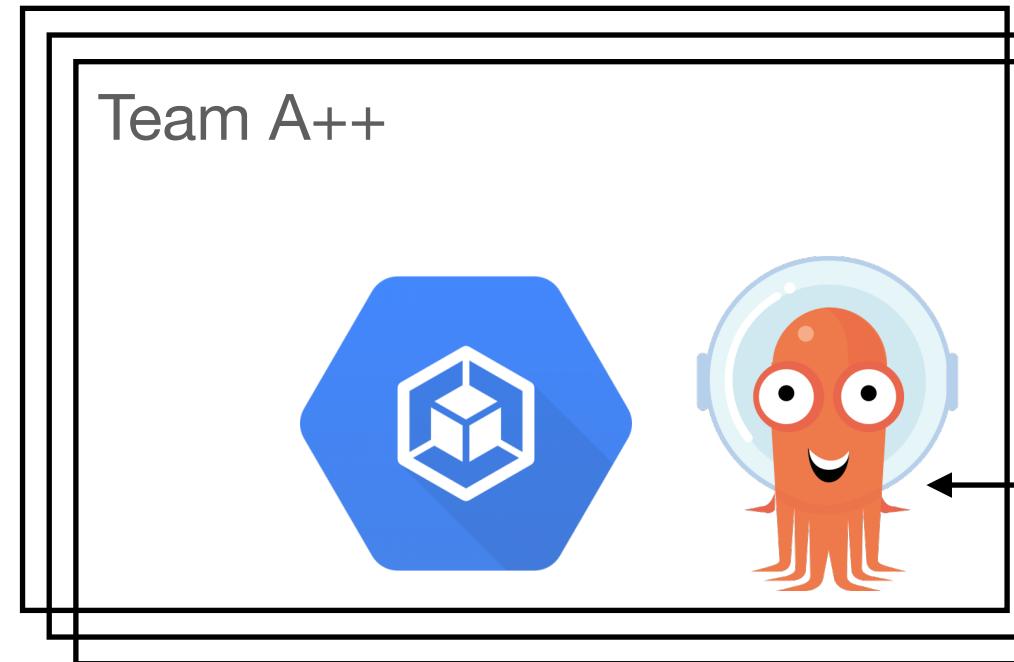
Any problem with that environment?

Company context



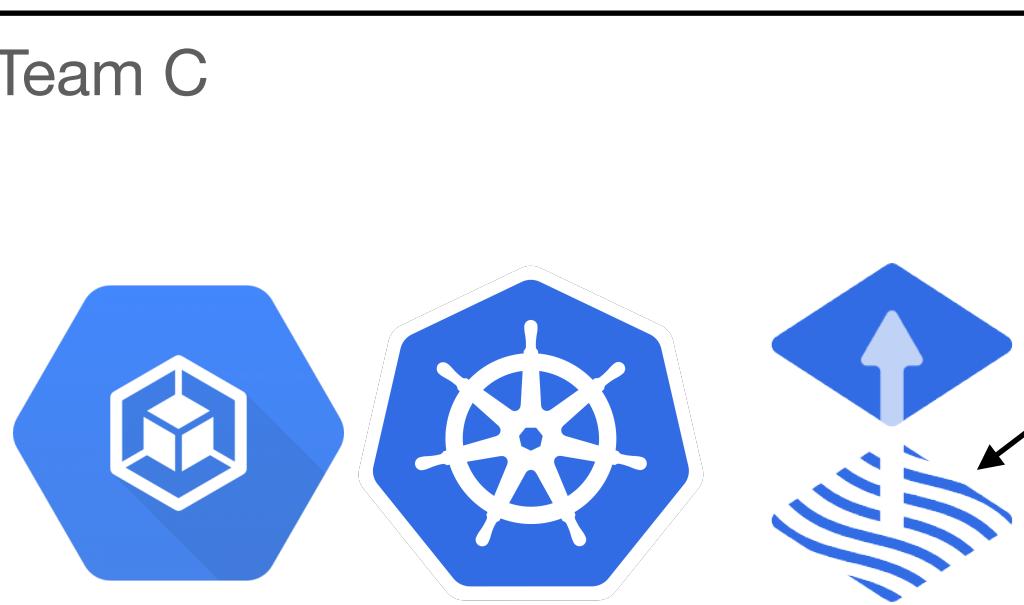
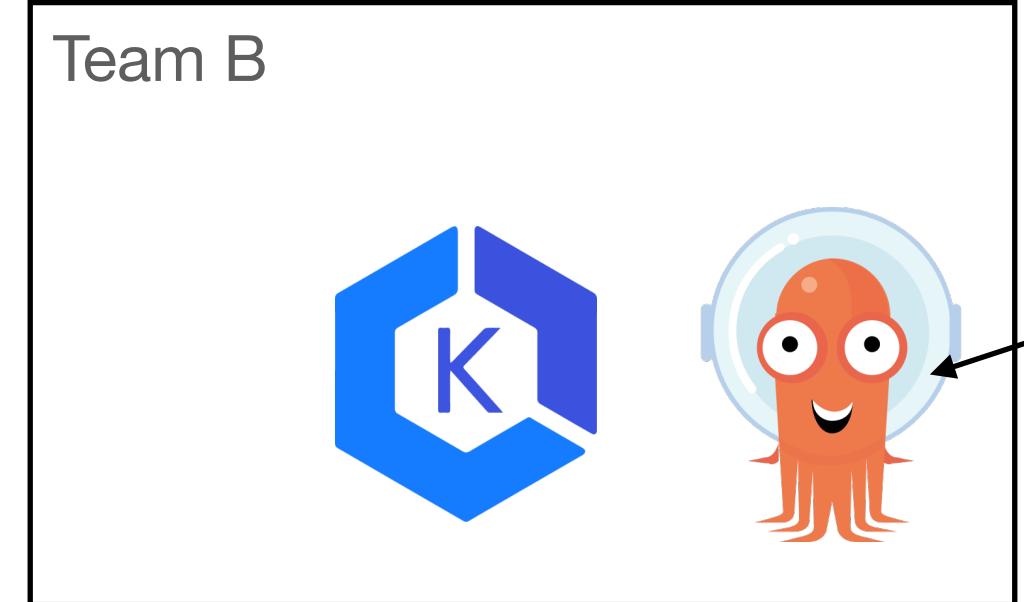
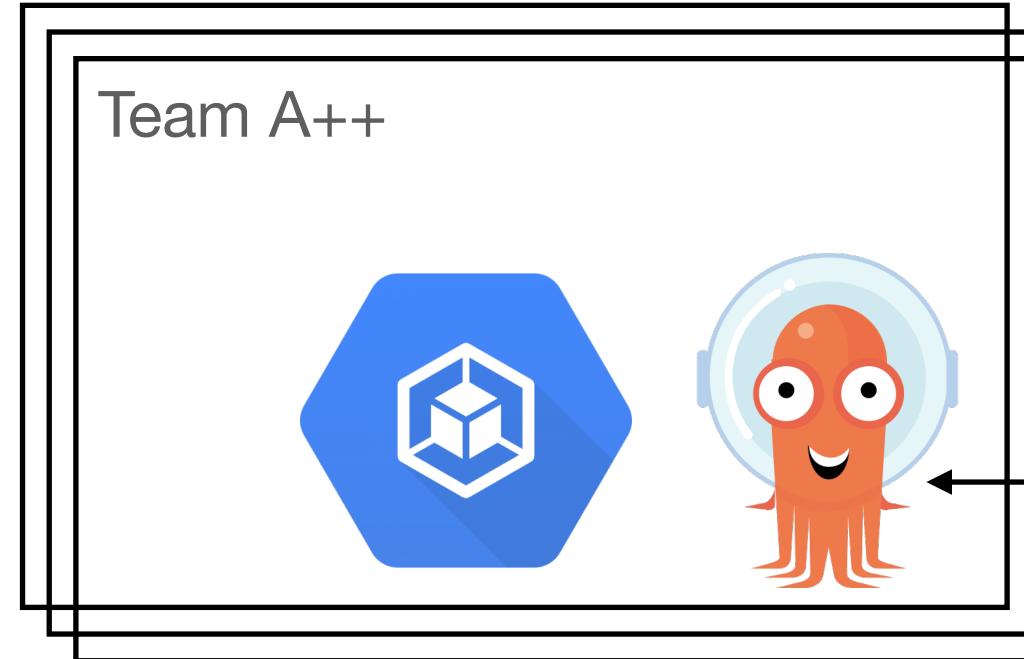
Any problem with that environment?

Company context



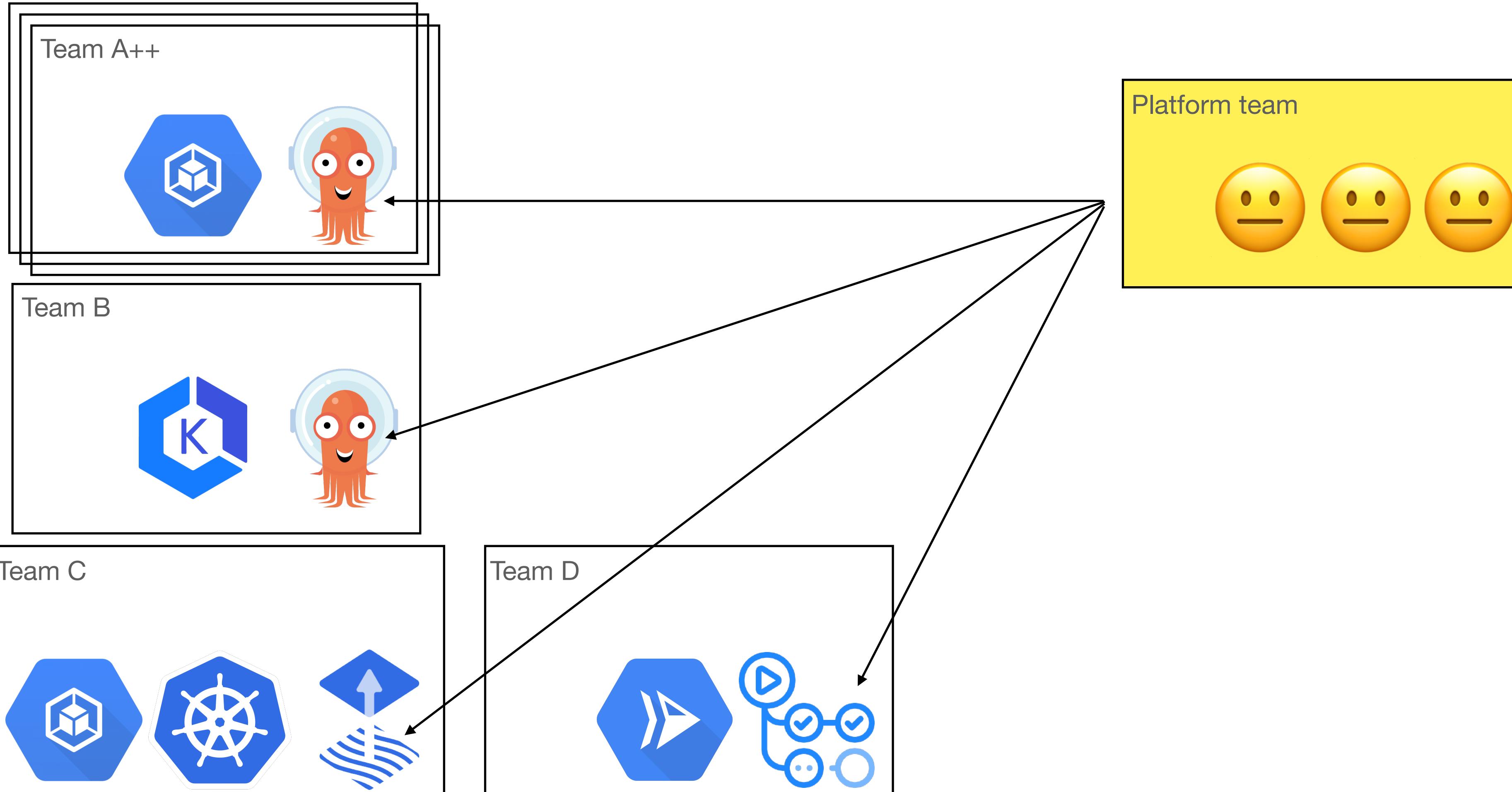
Any problem with that environment?

Company context



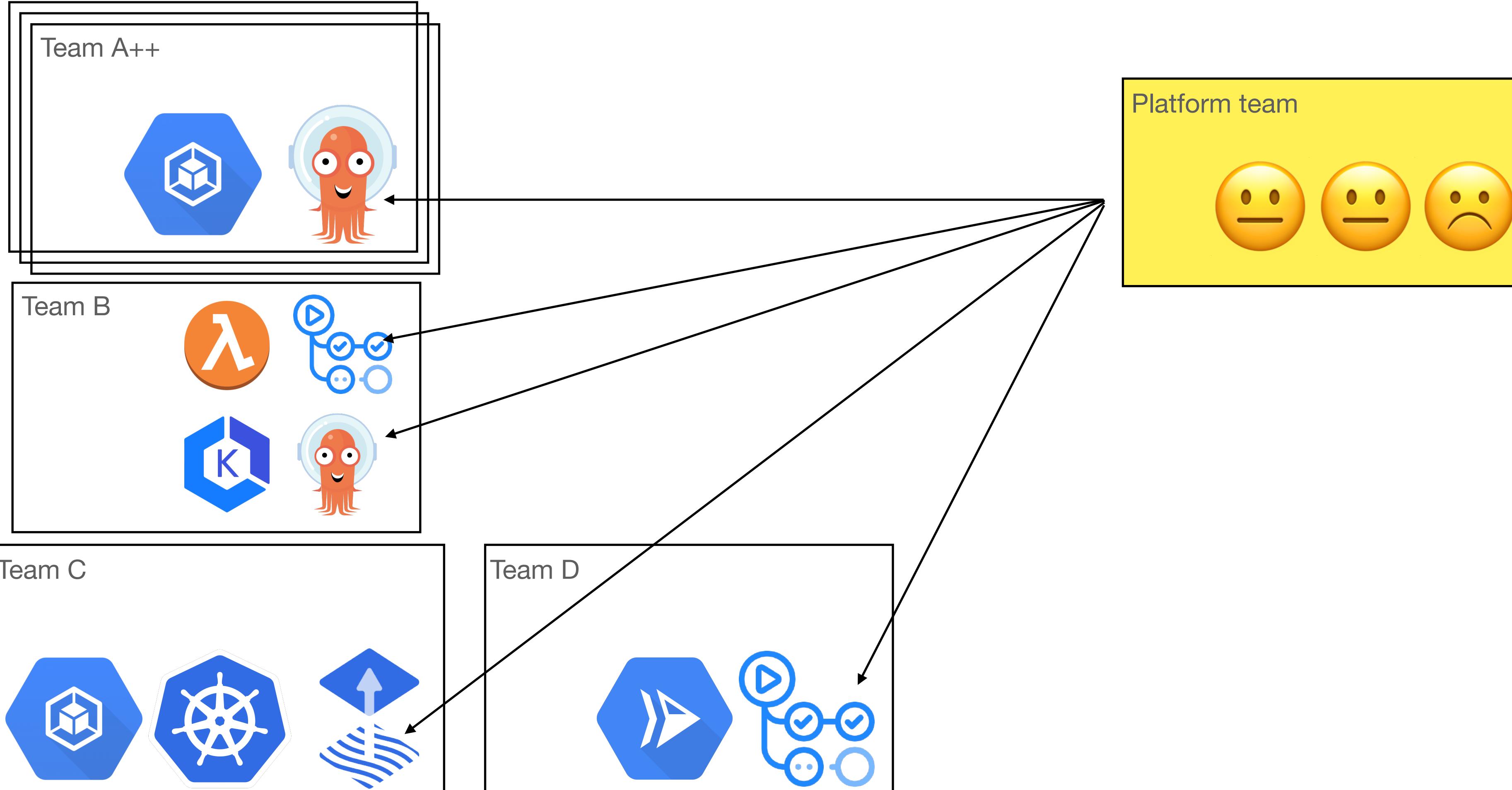
Any problem with that environment?

Company context



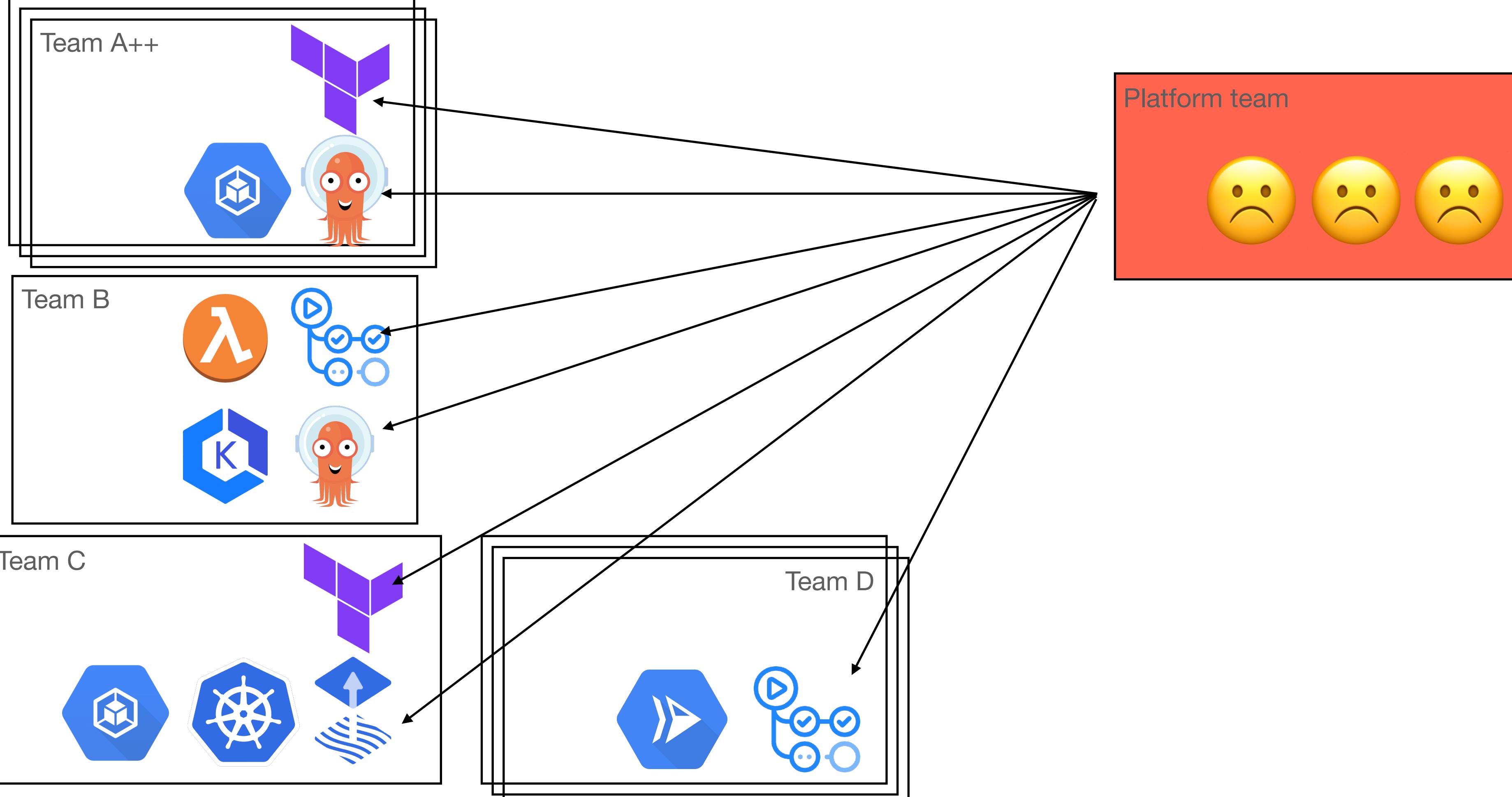
Any problem with that environment?

Company context



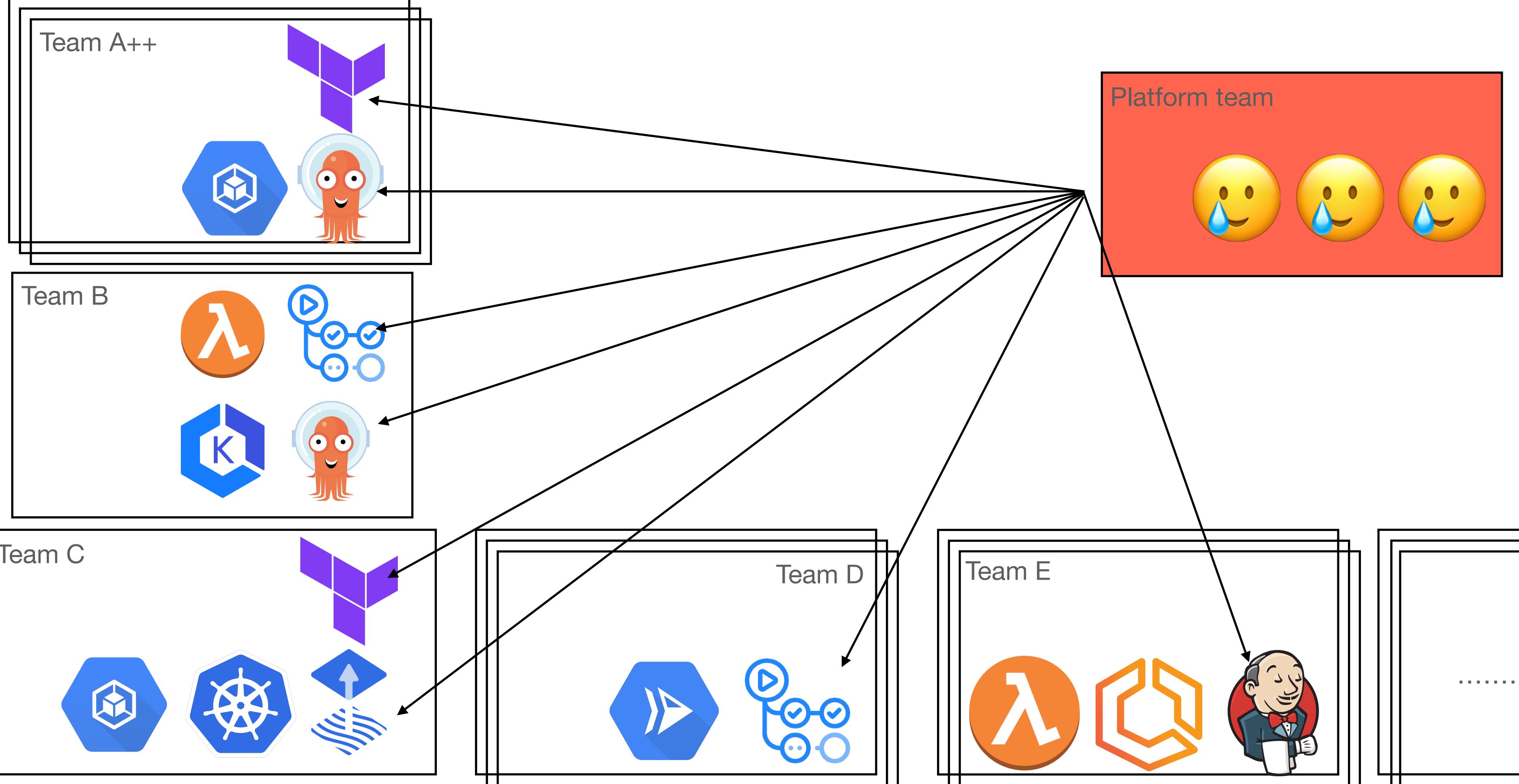
Any problem with that environment?

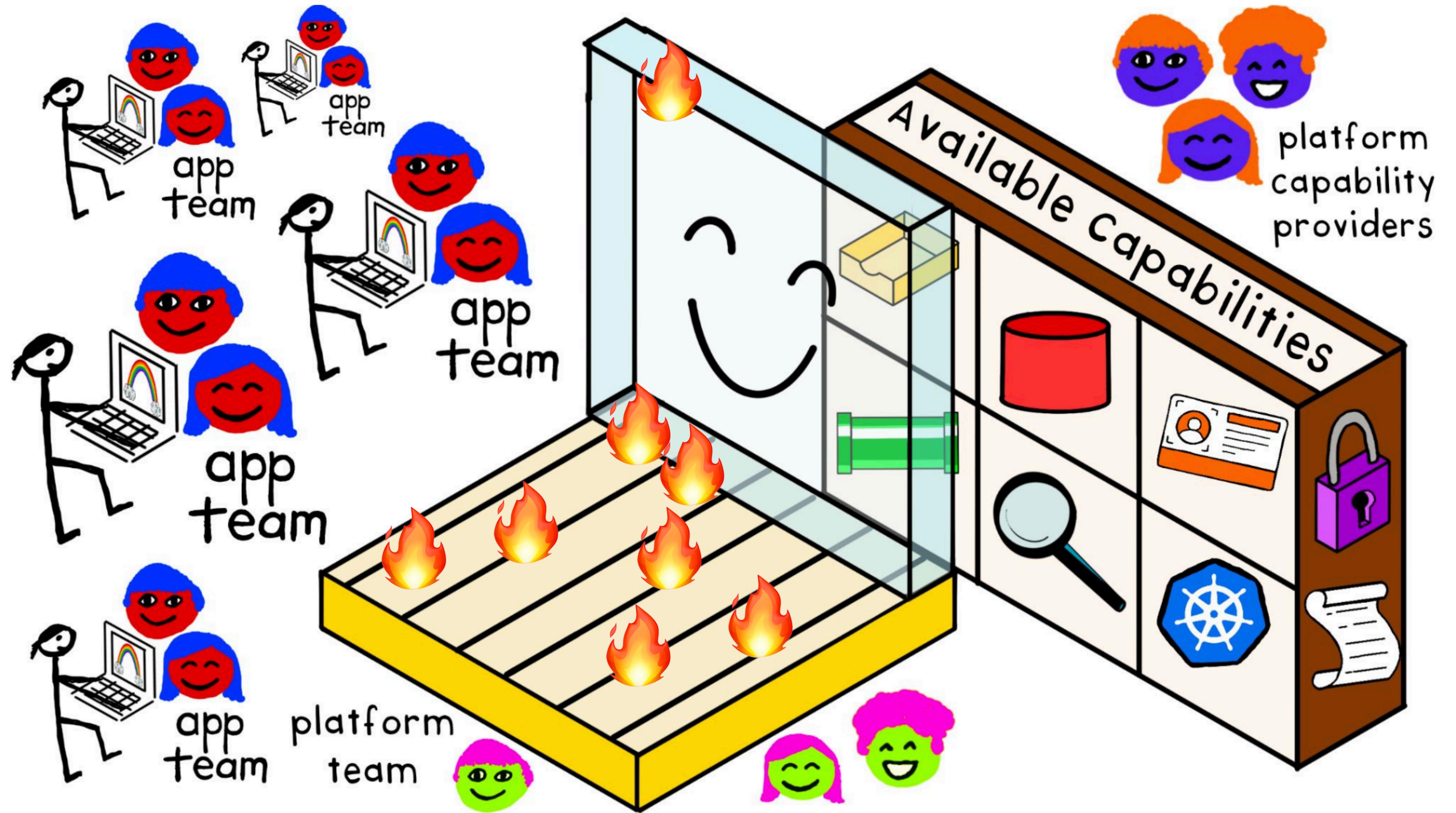
Company context



Any problem with that environment?

Company context





Picture by @wiggitywhitney, edited by me

Let's others do our jobs!

Let's others do our jobs!

In business world, we call it “**task delegation**”

What do we do?

- We need a way to implement “task delegation” idea
- And the solution must have a consistent and easy enough interface (so that the delegated target can handle it easily)

Platform team should treat the platform as their product!

Also by @wiggitywhitney

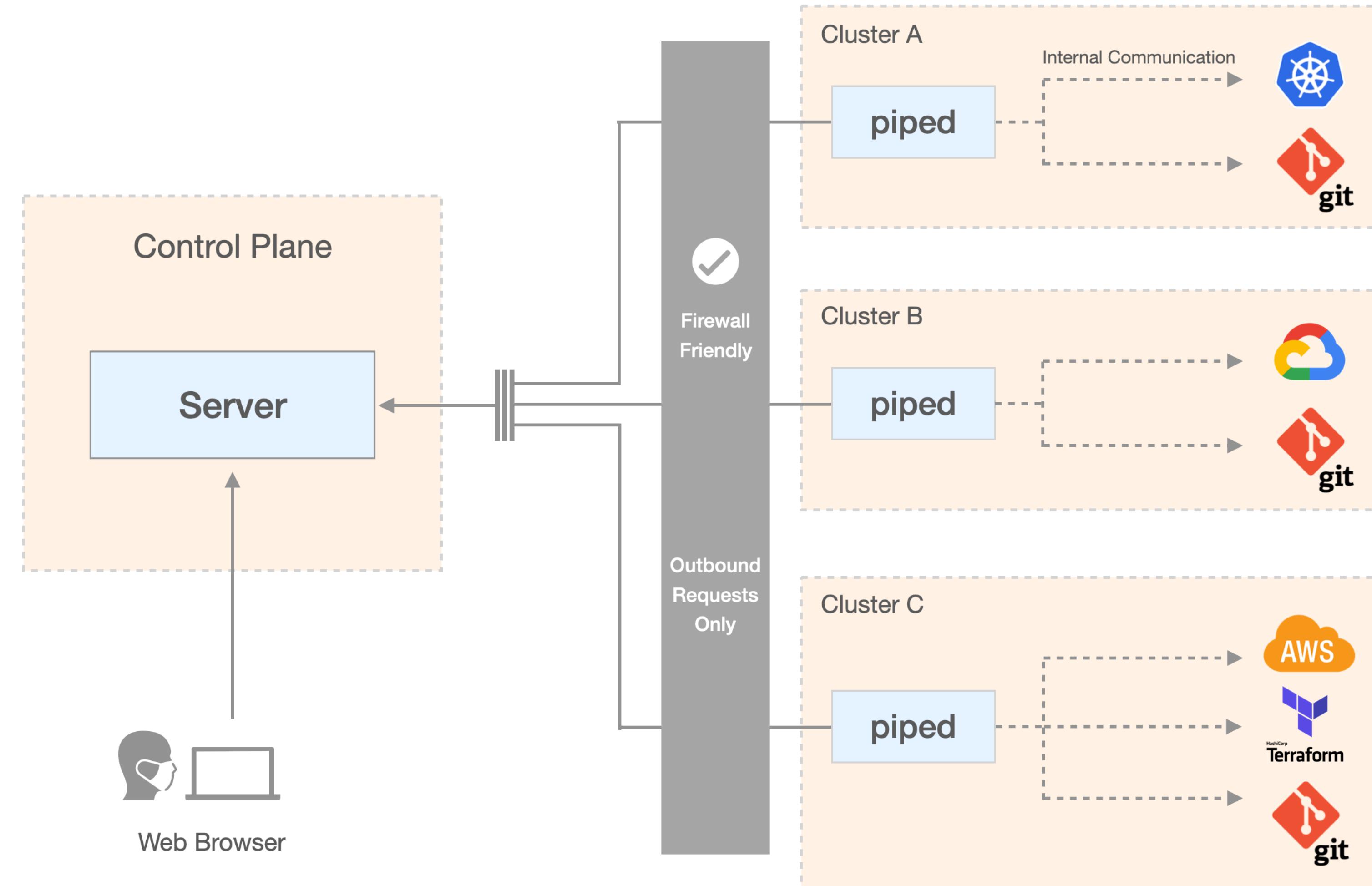


PipeCD

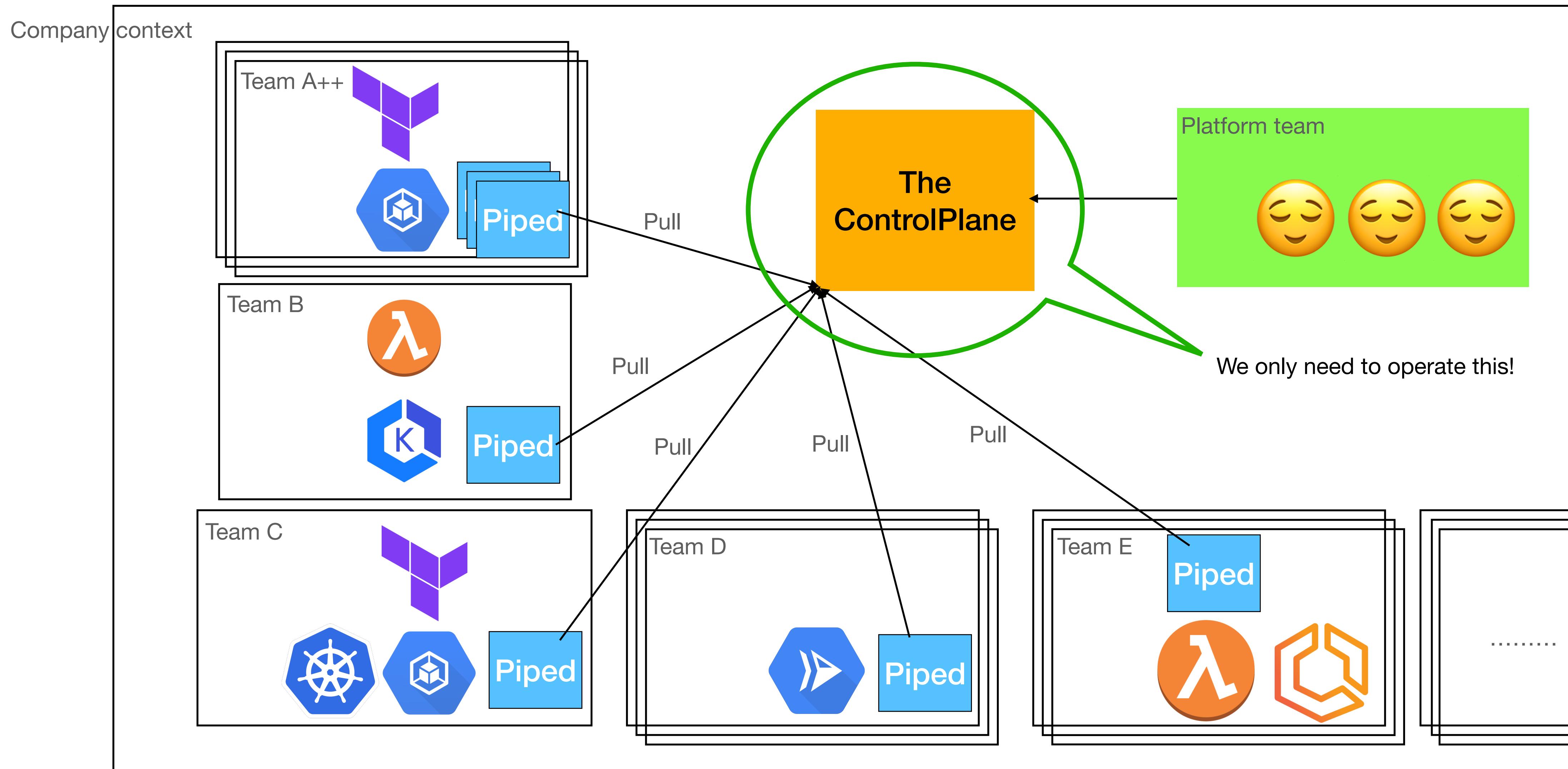
What do we do?

- We need a way to implement “task delegation” idea
- And the solution must have a consistent and easy enough interface (so that the delegated target can handle it easily)

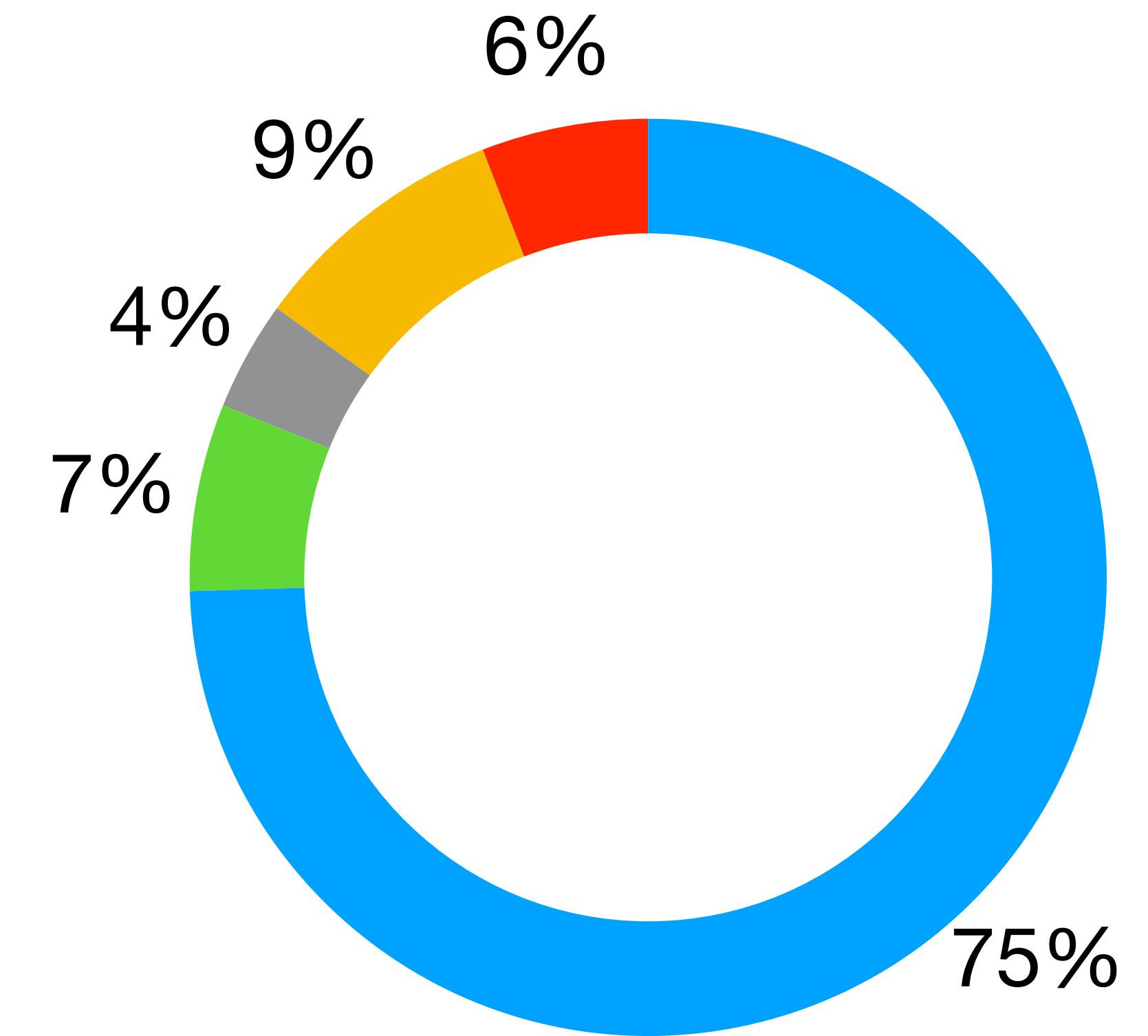
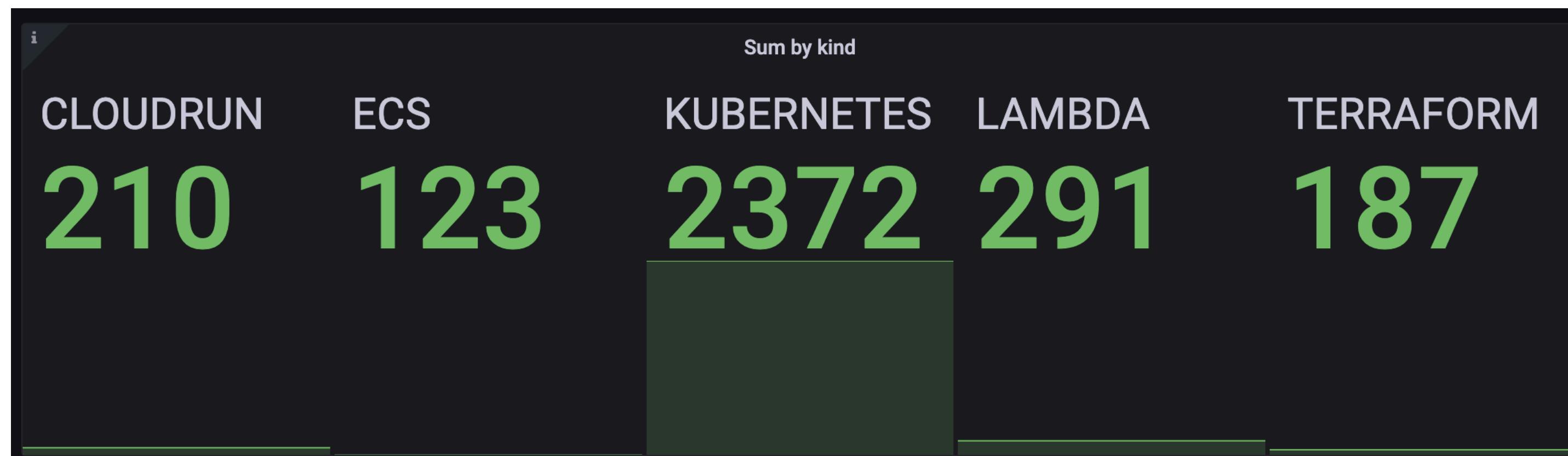
Our ideally CD architecture solution



Our ideally CD architecture solution



Our ideally CD architecture solution



1 PipeCD control plane well serves over 3200 apps!

What do we do?

- We need a way to implement “task delegation” idea
- And the solution must have a consistent and easy enough interface (so that the delegated target can handle it easily)

Consistent and easy interface

```
1  apiVersion: pipecd.dev/v1beta1
2  kind: CloudRunApp
3  spec:
4    name: wait-approval
5    pipeline:
6      stages:
7        - name: CLOUDRUN_PROMOTE
8          with:
9            percent: 10
10         - name: WAIT_APPROVAL
11         - name: CLOUDRUN_PROMOTE
12           with:
13             percent: 100
```

```
1  apiVersion: pipecd.dev/v1beta1
2  kind: KubernetesApp
3  spec:
4    name: wait-approval
5    pipeline:
6      stages:
7        - name: K8S_CANARY_ROLLOUT
8          with:
9            replicas: 10%
10         - name: WAIT_APPROVAL
11         - name: K8S_PRIMARY_ROLLOUT
12         - name: K8S_CANARY_CLEAN
13
```

SUCCESS 4 months ago

env: example

team: abc

The deployment was completed successfully

Application wait-approval

Piped piped-examples-cloudrun

Summary Sync with the specified pipeline because piped received a command from user via web console or pipecd

Commit

Sync with pipe-cd/pipecd repository at
f503cb62d695023557c79a3ceea7322f15f89dc1 (b292dfa
[🔗](#))

Triggered by khanhtc1202

 CLODRUN_PROMOTE

10% Promoted

 WAIT_APPROVAL

Approved by khanhtc1202

 CLODRUN_PROMOTE

100% Promoted

SUCCESS a year ago

env: example

team: product

The deployment was completed successfully

Application wait-approval

Piped piped-examples-gcp

Summary Sync progressively because of updating image helloworld from v0.7.0 to v0.30.0

Commit

Update to use newest artifacts (#111) (e690800 [🔗](#))

Triggered by Le Van Nghia

 K8S_CANARY_ROLLOUT

 WAIT_APPROVAL

Approved by nghialv

 K8S_PRIMARY_ROLLOUT

 K8S_CANARY_CLEAN

Why PipeCD?

- Simple, unified and easy to use but powerful pipeline definition to construct your deployment
- Same deployment interface to deploy applications of any platform, including Kubernetes, Terraform, GCP Cloud Run, AWS Lambda, AWS ECS
- No CRD or applications' manifest changes are required; Only need a pipeline definition along with your application manifests
- No deployment credentials are exposed or required outside the application cluster
- Built-in deployment analysis as part of the deployment pipeline to measure impact based on metrics, logs, emitted requests
- Easy to interact with any CI; The CI tests and builds artifacts, PipeCD takes the rest
- Insights show metrics like lead time, deployment frequency, MTTR and change failure rate to measure delivery performance
- Designed to manage thousands of cross-platform applications in multi-cloud for company scale but also work well for small projects

Why PipeCD?

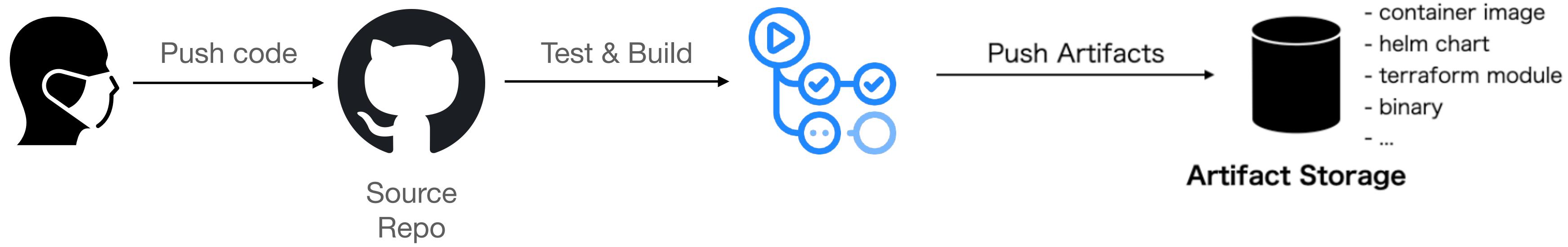
- Simple, unified and easy to use but powerful pipeline
- Same deployment interface to deploy applications of any platform,
- No CRD or applications' manifest changes are required; with no extra costs
- No deployment credentials are exposed
- Built-in deployment analysis as part of the deployment pipeline based on static code analysis
- Easy to interact with any CI; including Jenkins, GitHub Actions, CircleCI, and GitLab CI
- Insights show metrics of your deployment including success rate, failure rate, and enhancement
- Designed to manage thousands of cross-platform applications in multi-cloud for company scale but also work well for small projects

Why PipeCD?

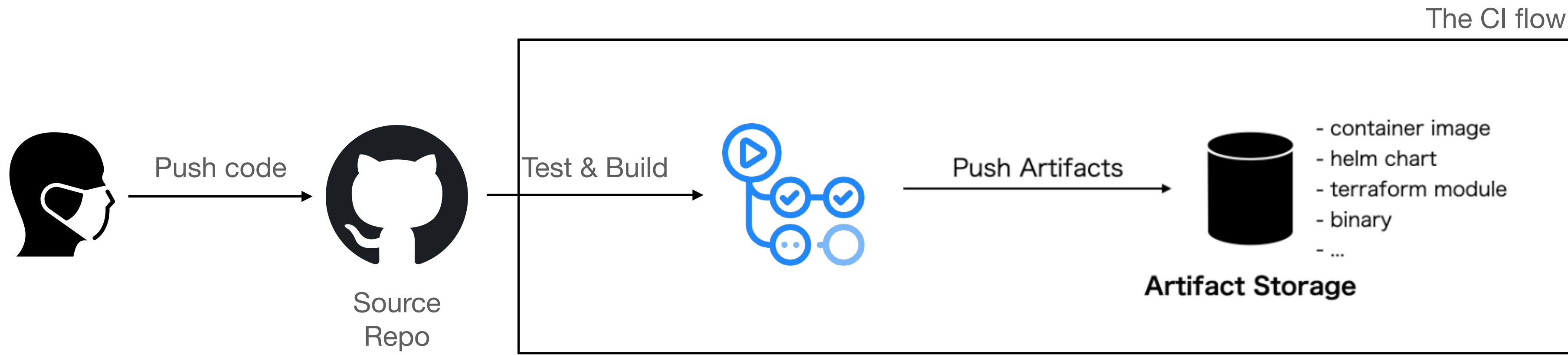
- Sit back and let PipeCD handle the deployment
 - Same deployment interface to deploy applications of any platform
 - No CI/CD configuration required. Just focus on your application code.
 - No dependency on external tools or services.
 - Built-in support for Kubernetes, AWS Lambda, and AWS Step Functions.
 - Easy to use and integrates well with popular CI/CD tools like GitHub Actions and Jenkins.
 - Insights and monitoring features help you understand the status of your deployments.
 - Design focus is to manage thousands of cross-platform applications in a large enterprise company scale but also work well for small projects
- Sorry for the wall of text 😂

Let's walk through some use-cases

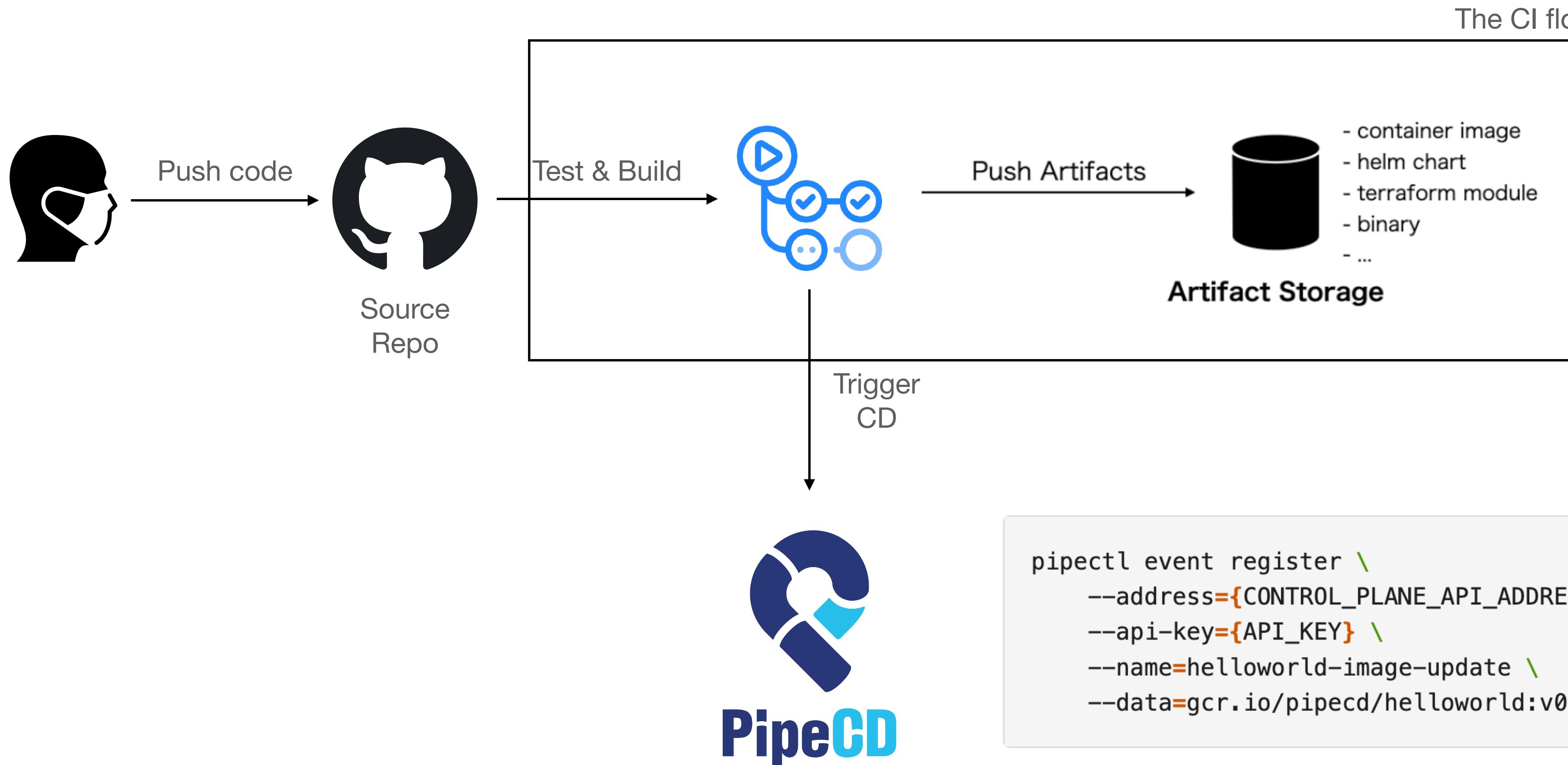
Common flow



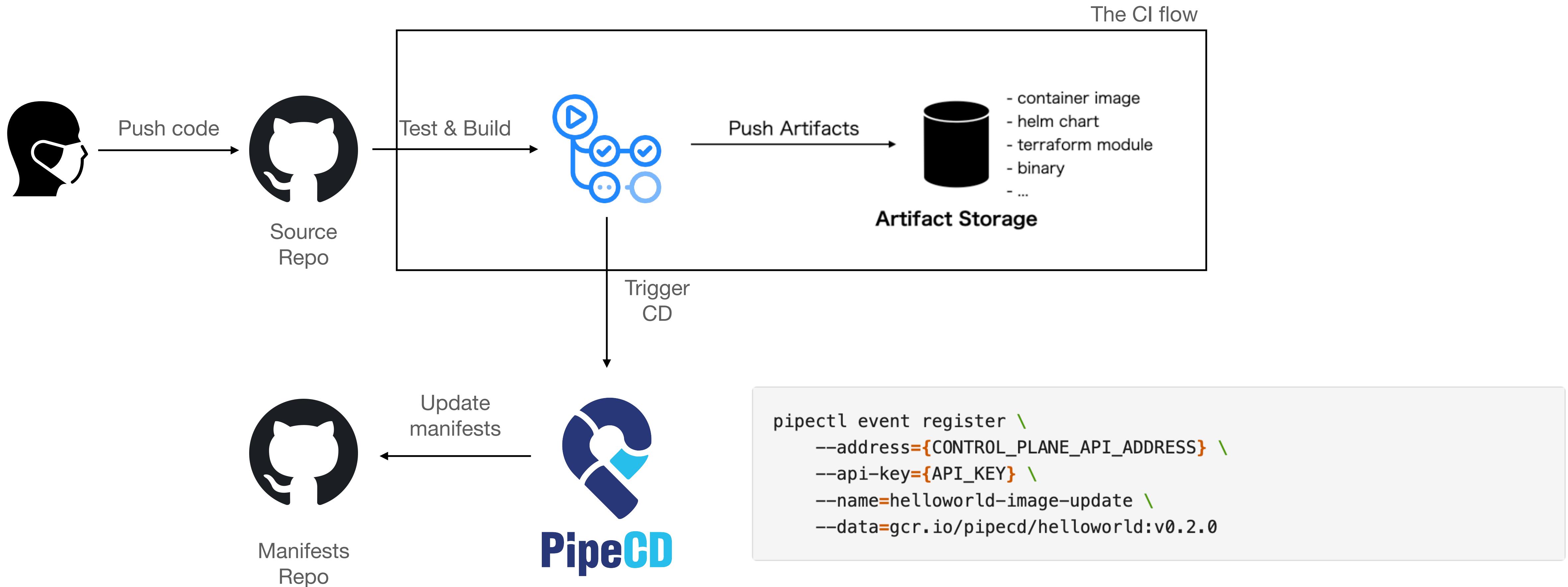
Common flow



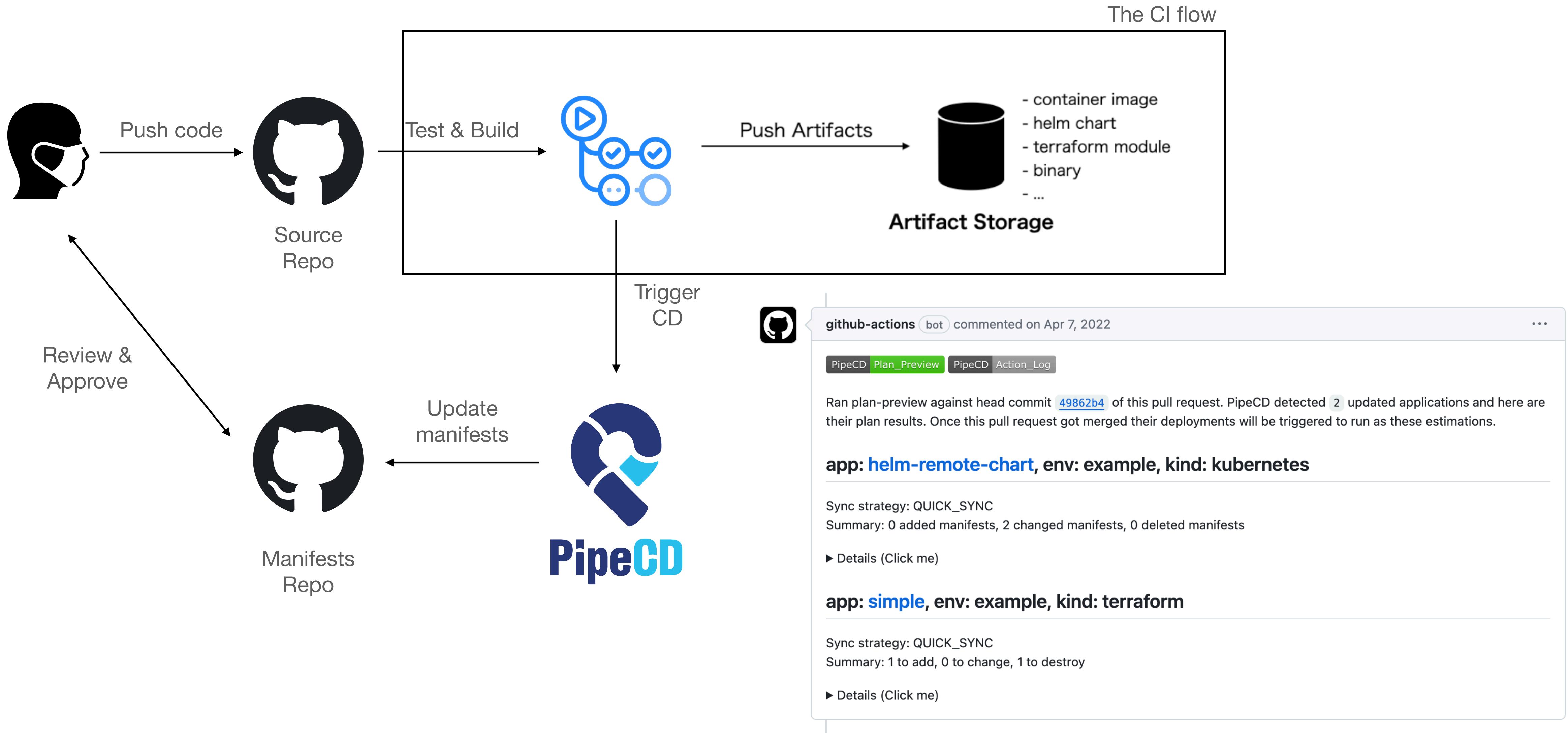
Common flow - Event watcher



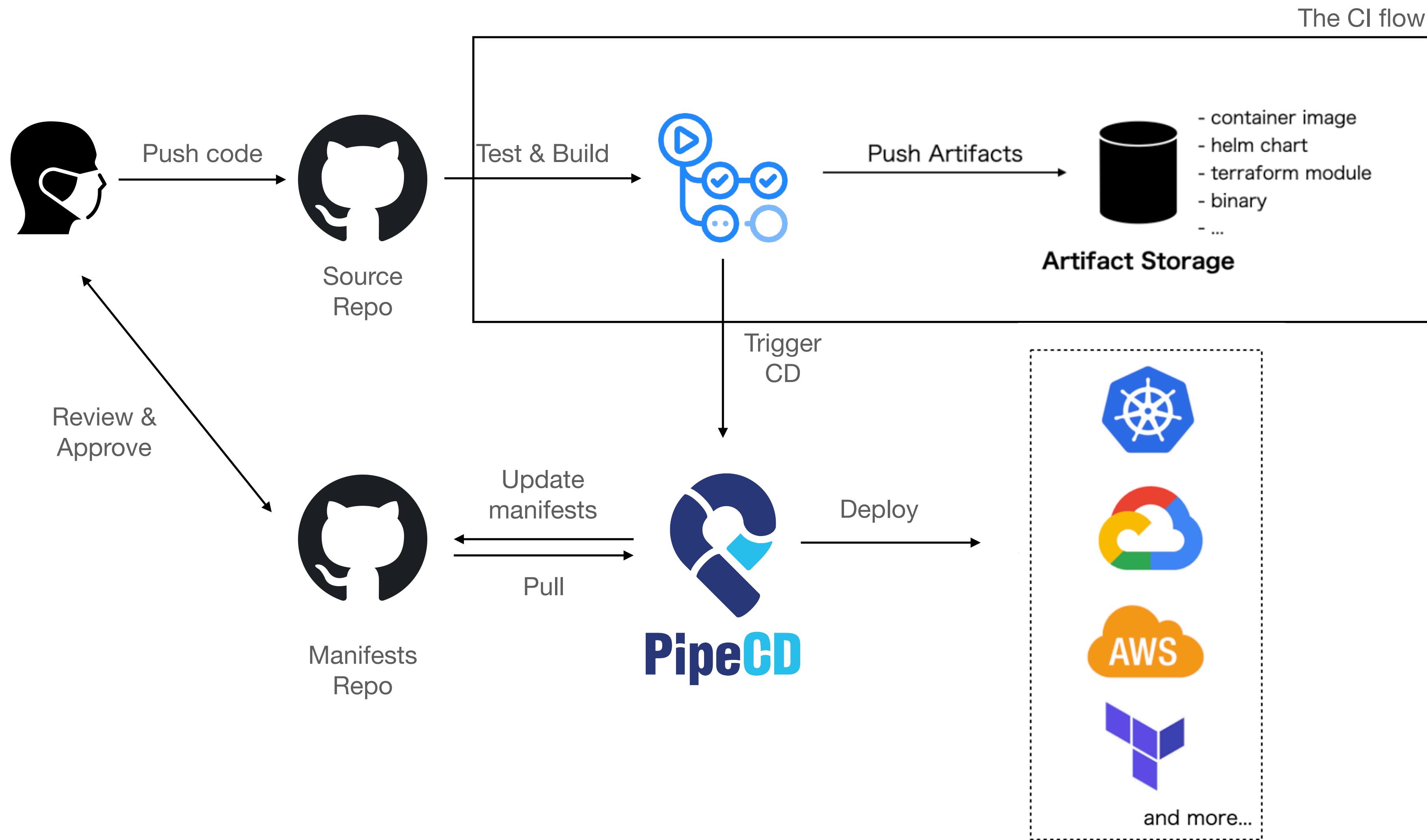
Common flow - Plan preview



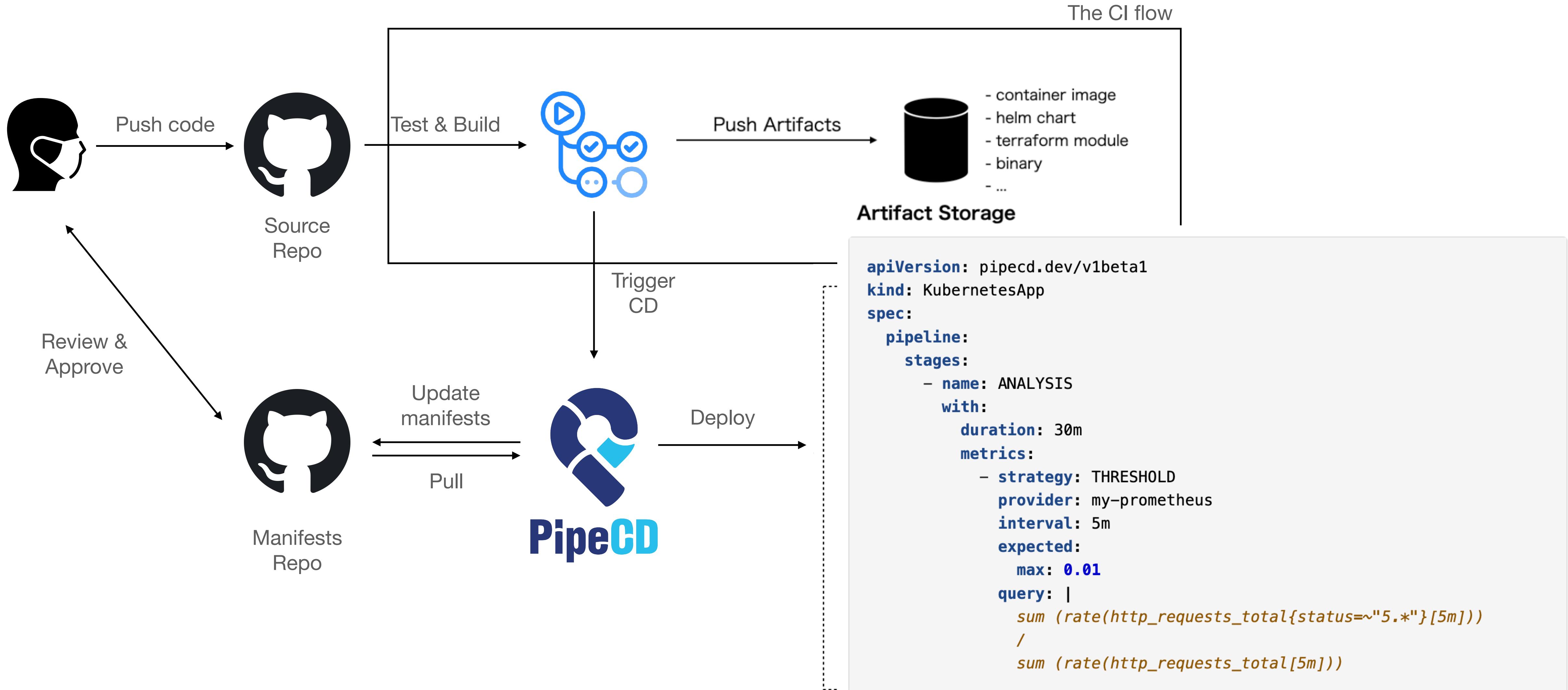
Common flow - Plan preview



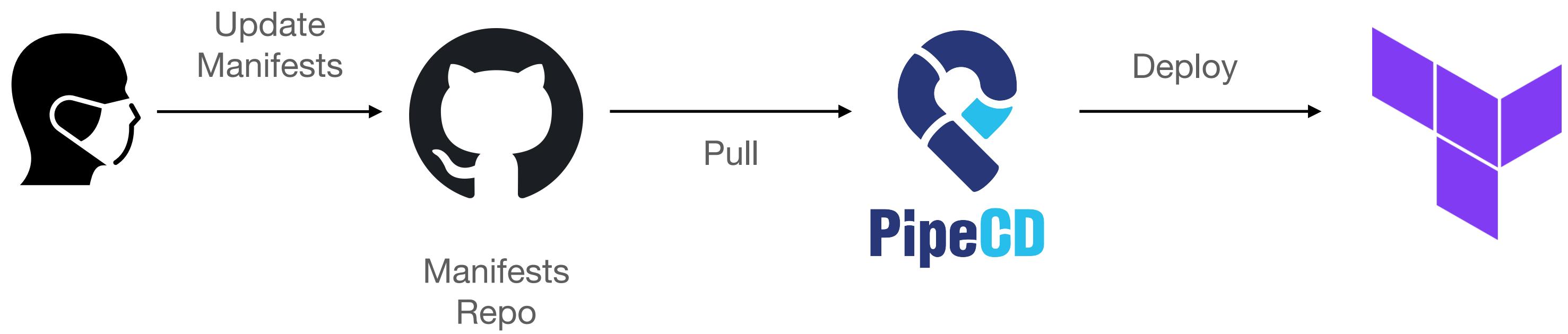
Common flow



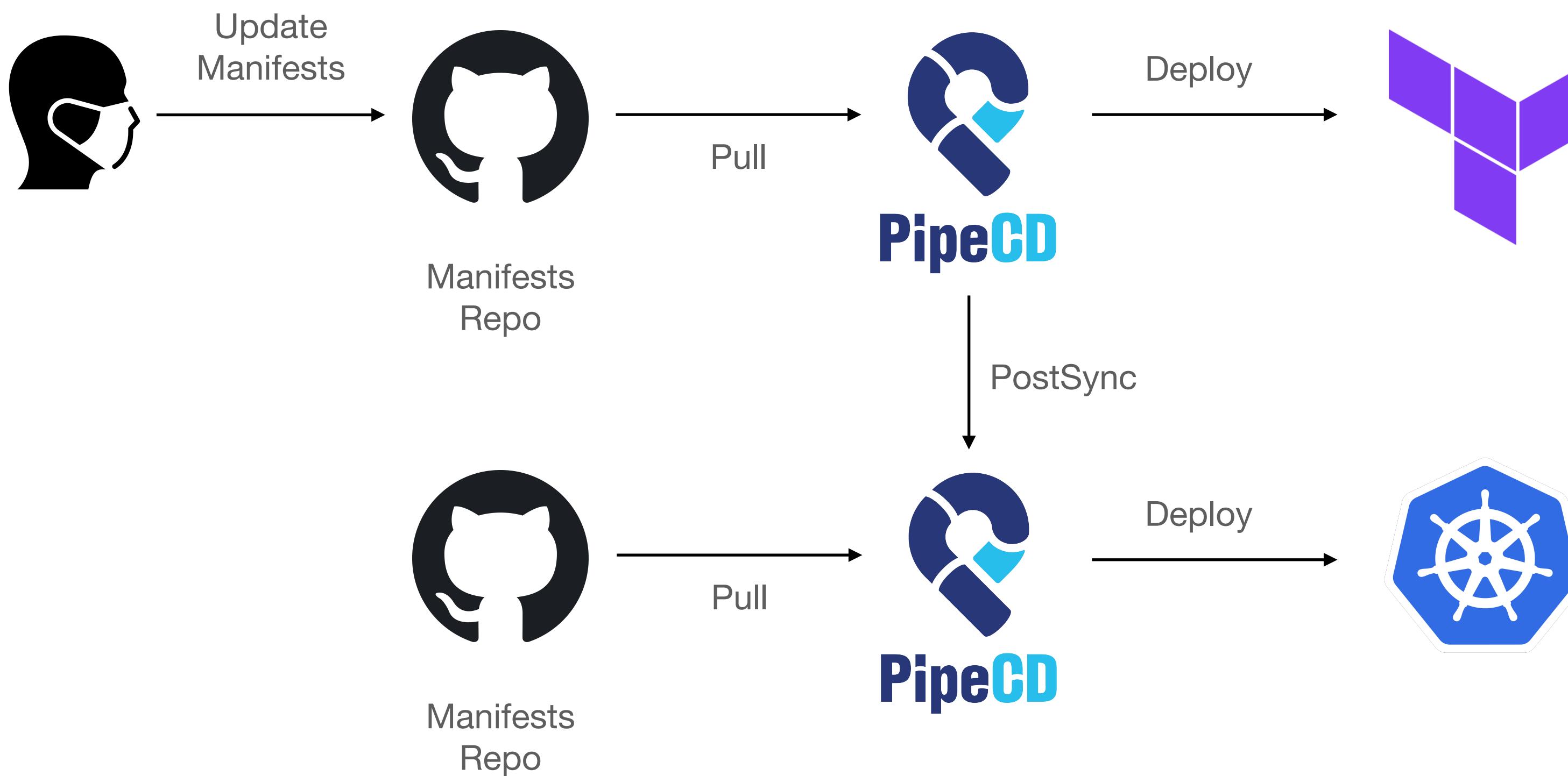
Common flow - Progressive with ADA



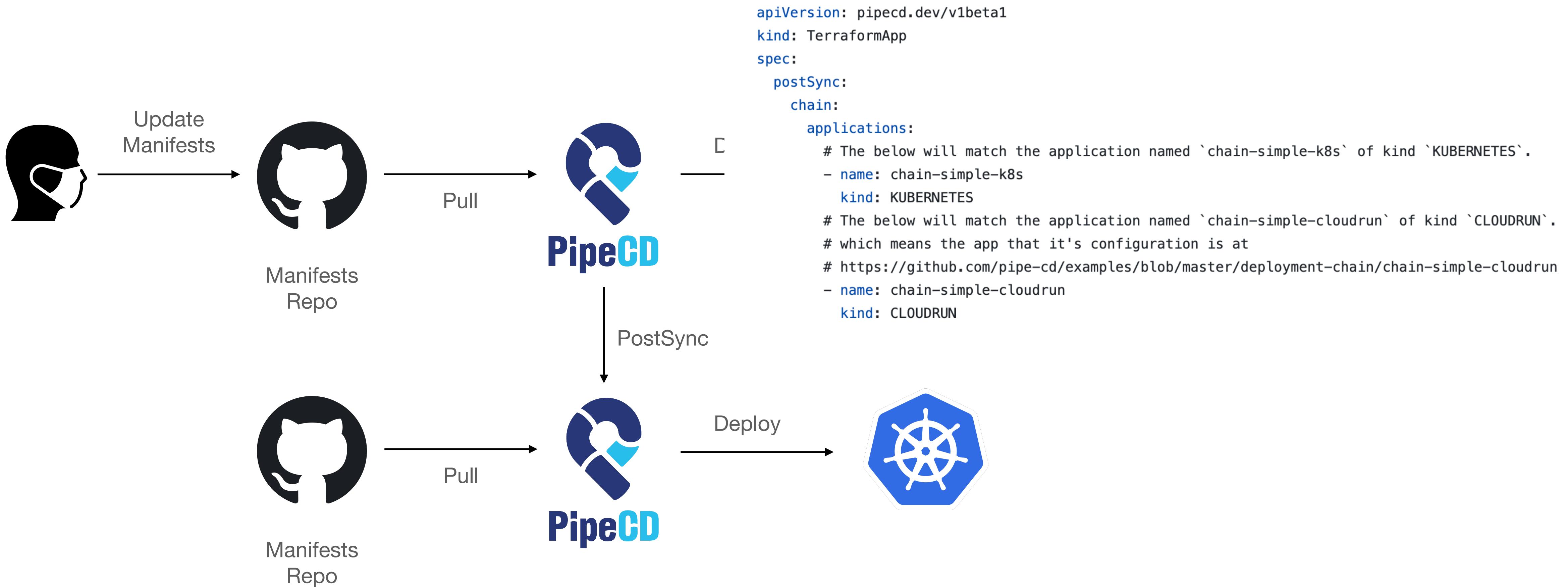
Uncommon flow



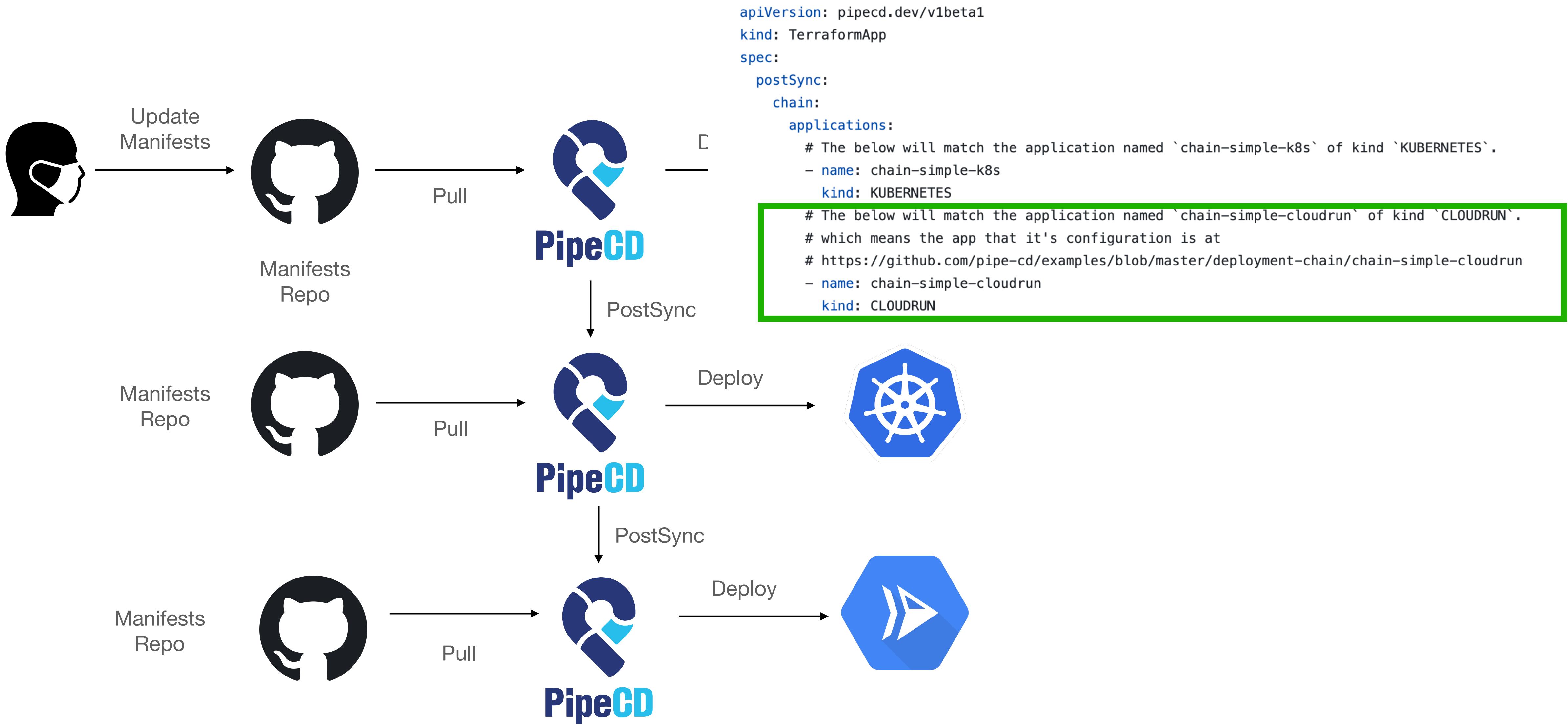
Uncommon flow - deployment chain



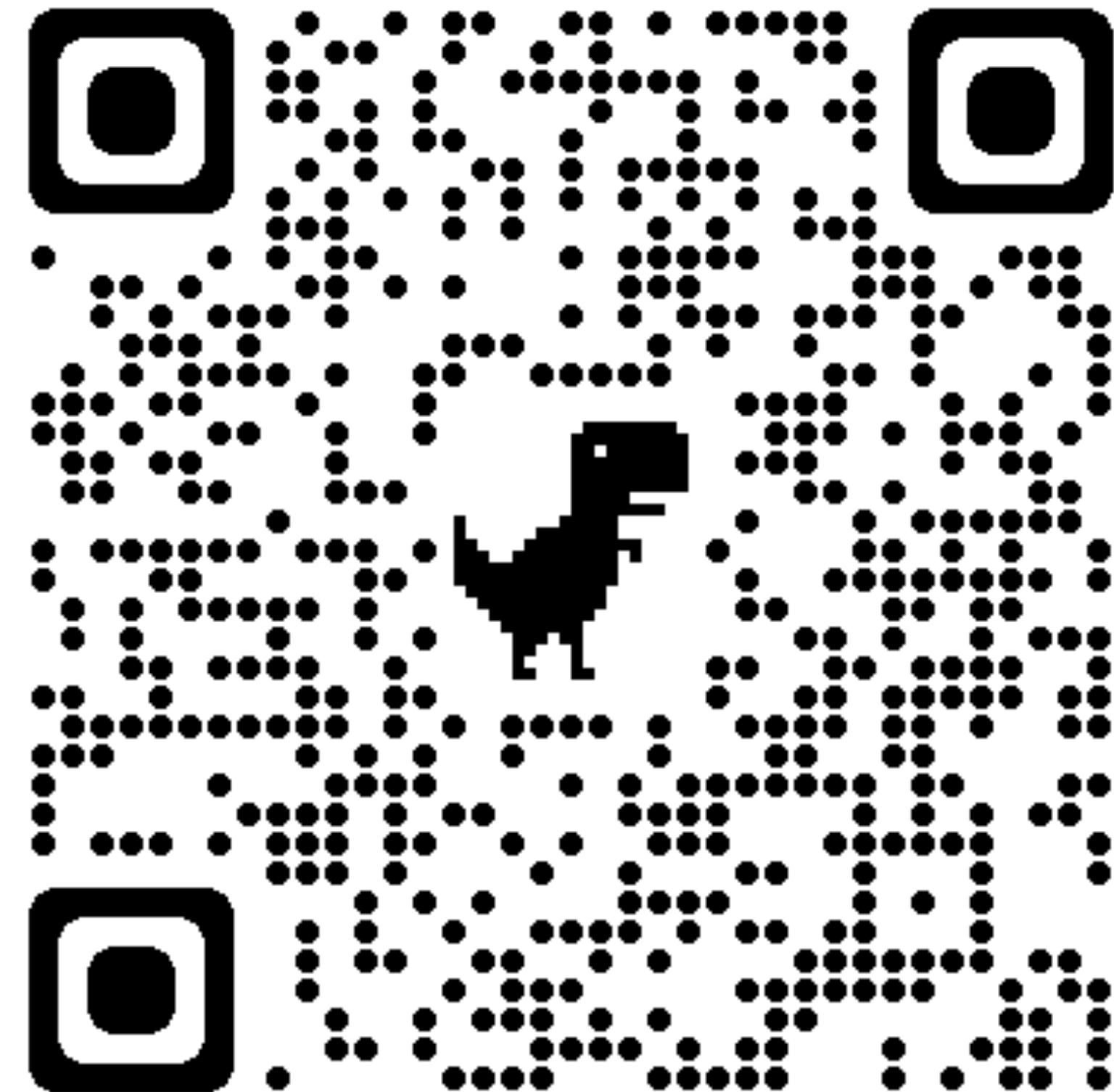
Uncommon flow - deployment chain



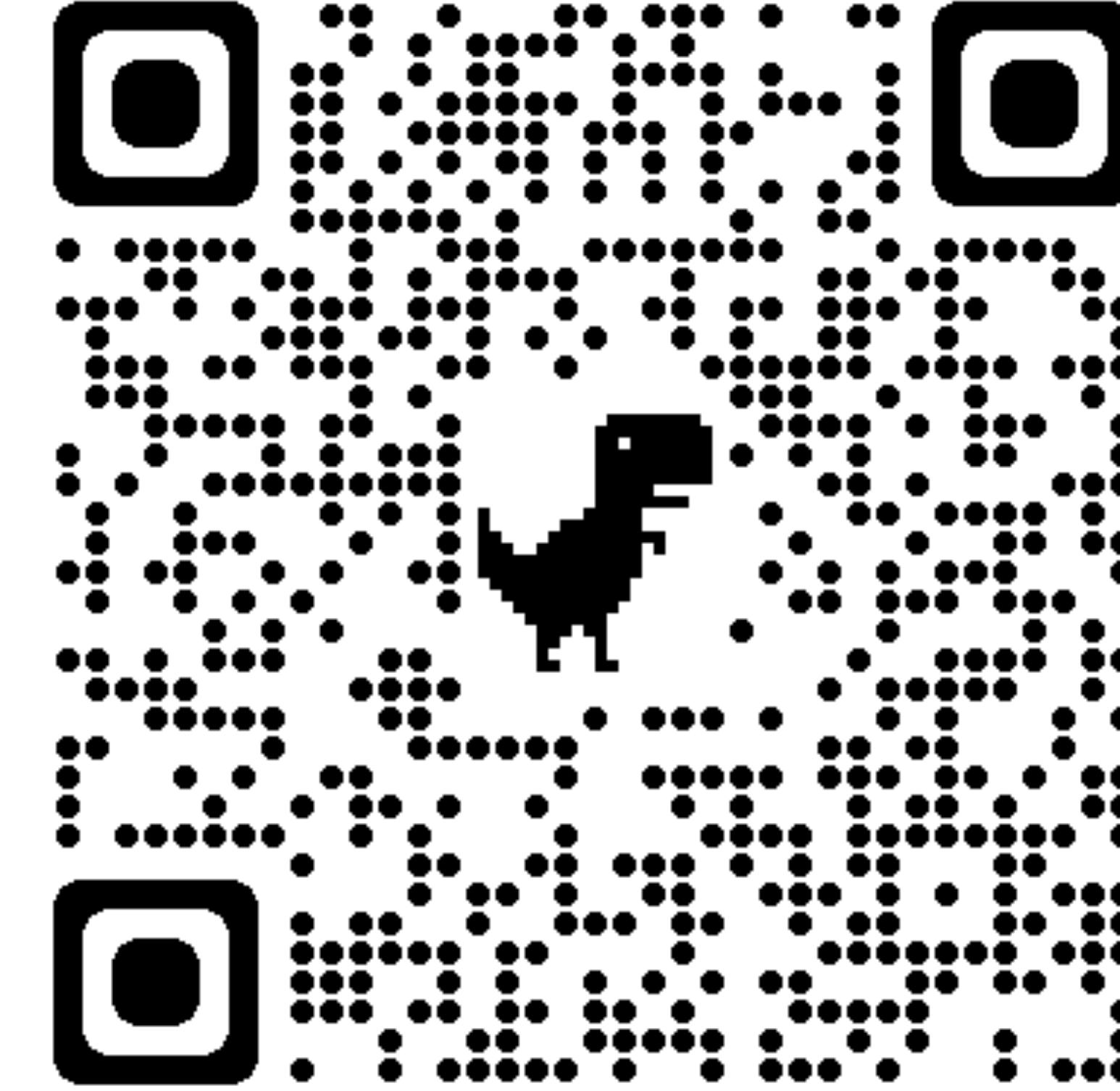
Uncommon flow - deployment chain



And that's not ALL



<https://pipecd.dev>



<https://play.pipecd.dev>

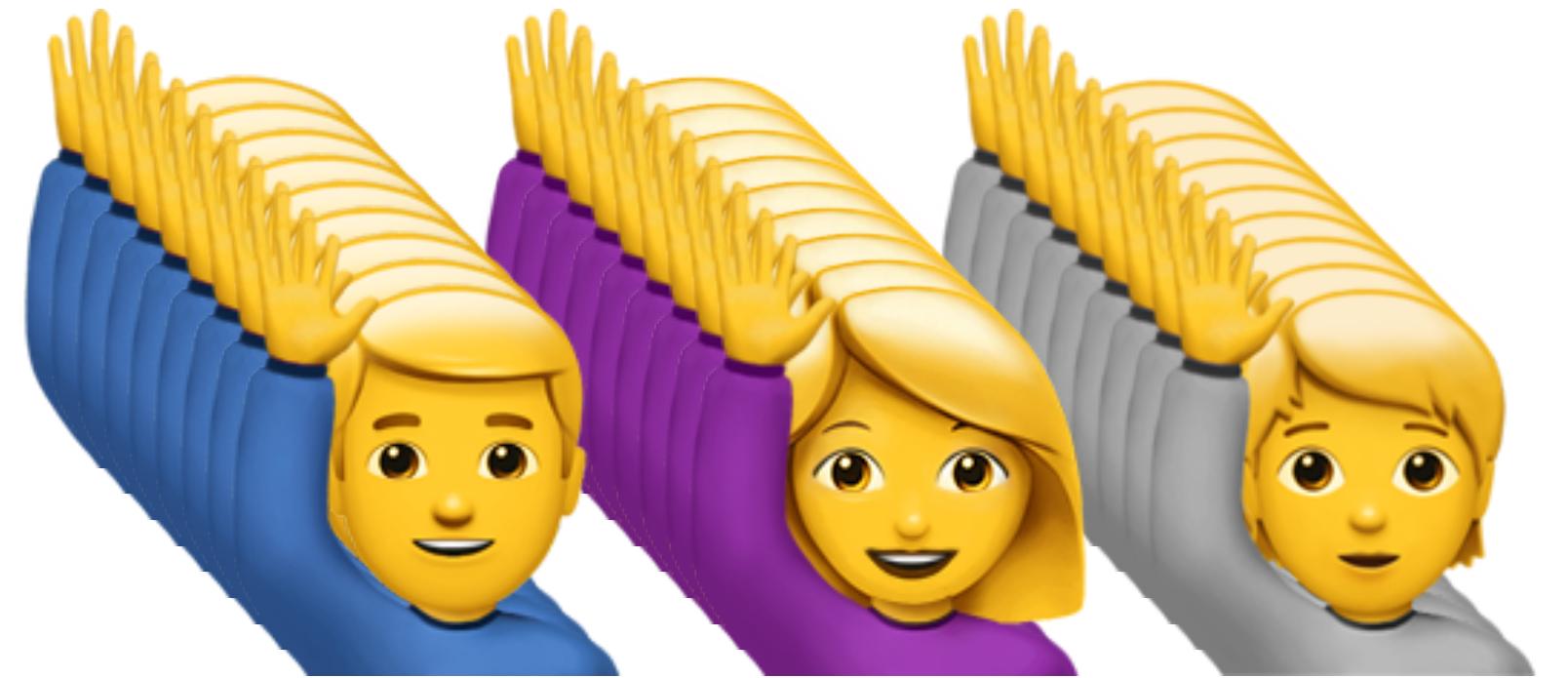
PipeCD current status

- Latest release version is v0.45.3 (production used)
- CNCF Sandbox joined from May 2023 🎉



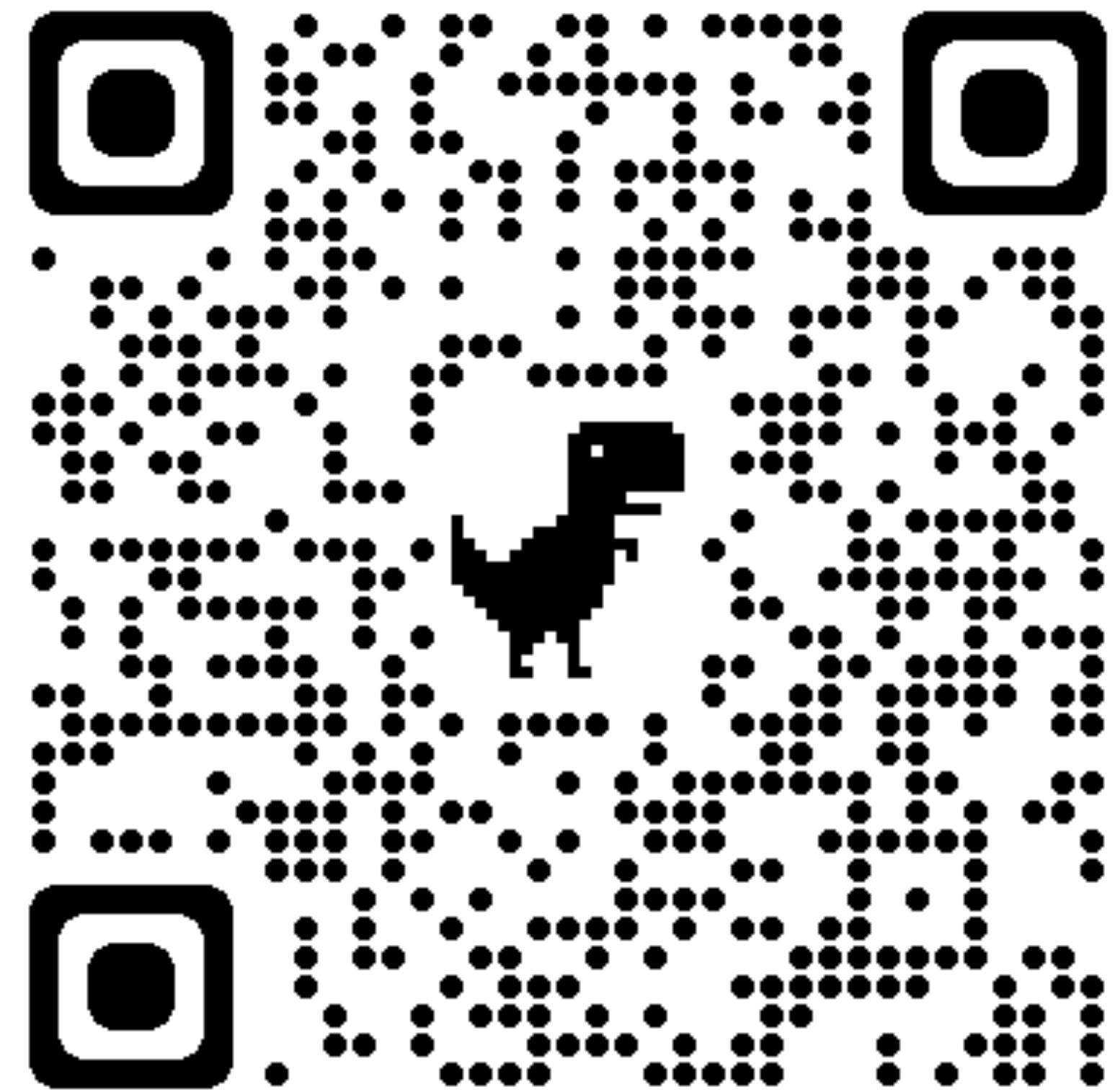
Roadmap

- We're on the road to version v1.0 with several big updates
 - Adopt pluggable architecture for the piped (the agent) to enable users self create plugins
 - Find a way to interact with ArgoRollout / FluxCD / OpenKruise Rollout / ...
 - More features on pre/post deployment
- It's fresh and has a lot of room for growth



Let's keep in touch

- CNCF slack #pipecd
- Github pipe-cd/pipecd
- Official site pipecd.dev



Thanks for listening 😊

<https://pipecd.dev>