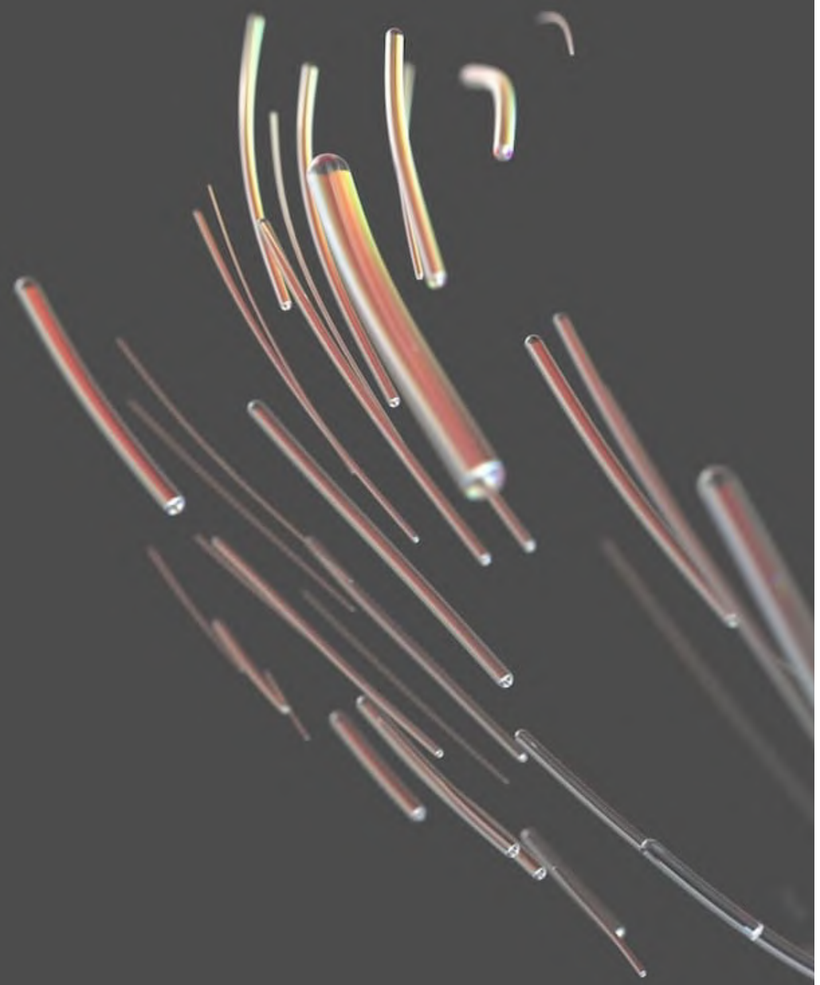


Code Days 2022 / 01.02.2022

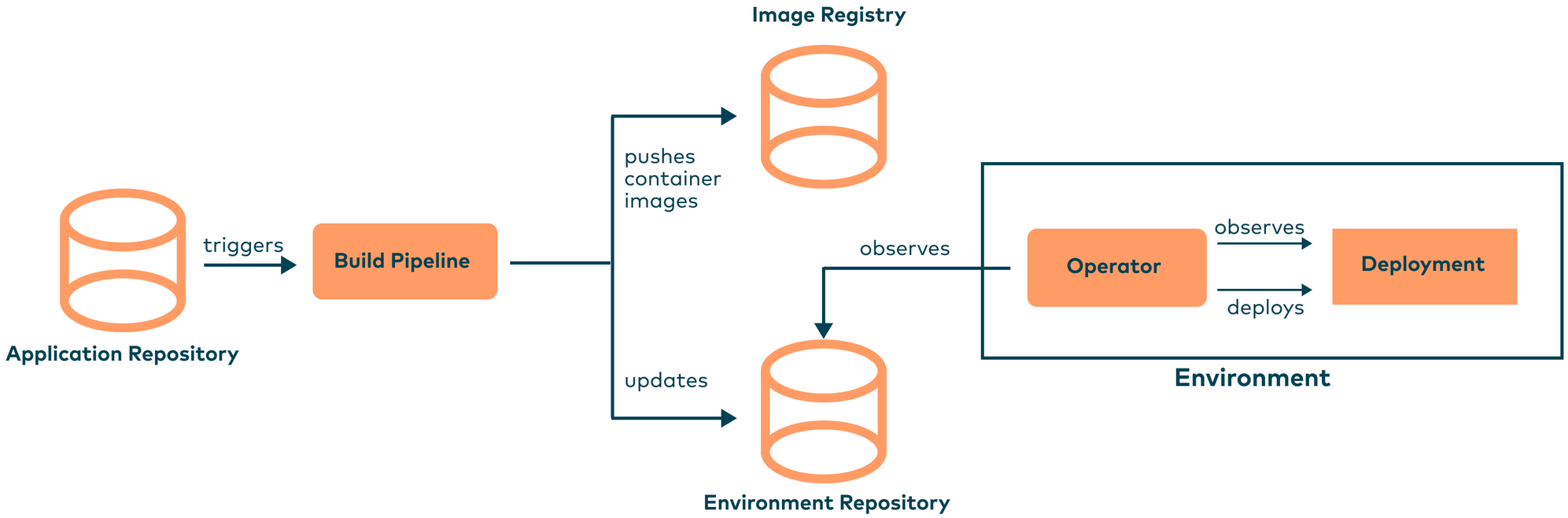
GitOps

Häufige Missverständnisse und übliche Fallstricke

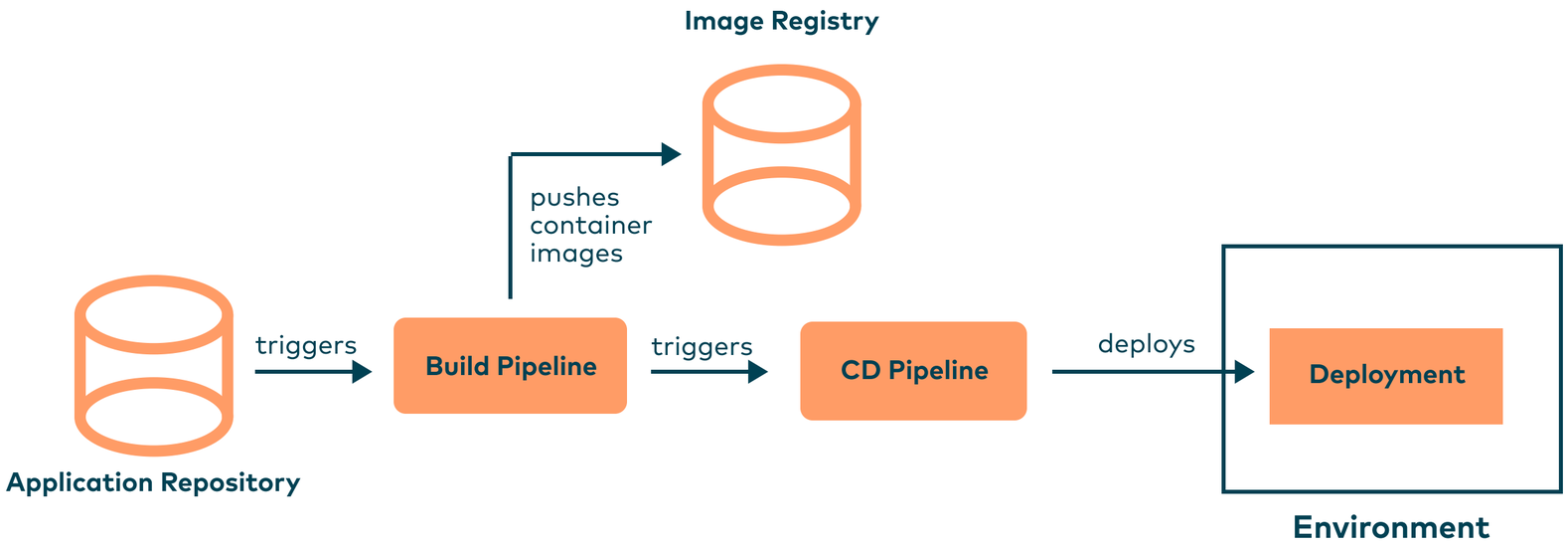
INNOQ

ANJA KAMMER
SENIOR CONSULTANT

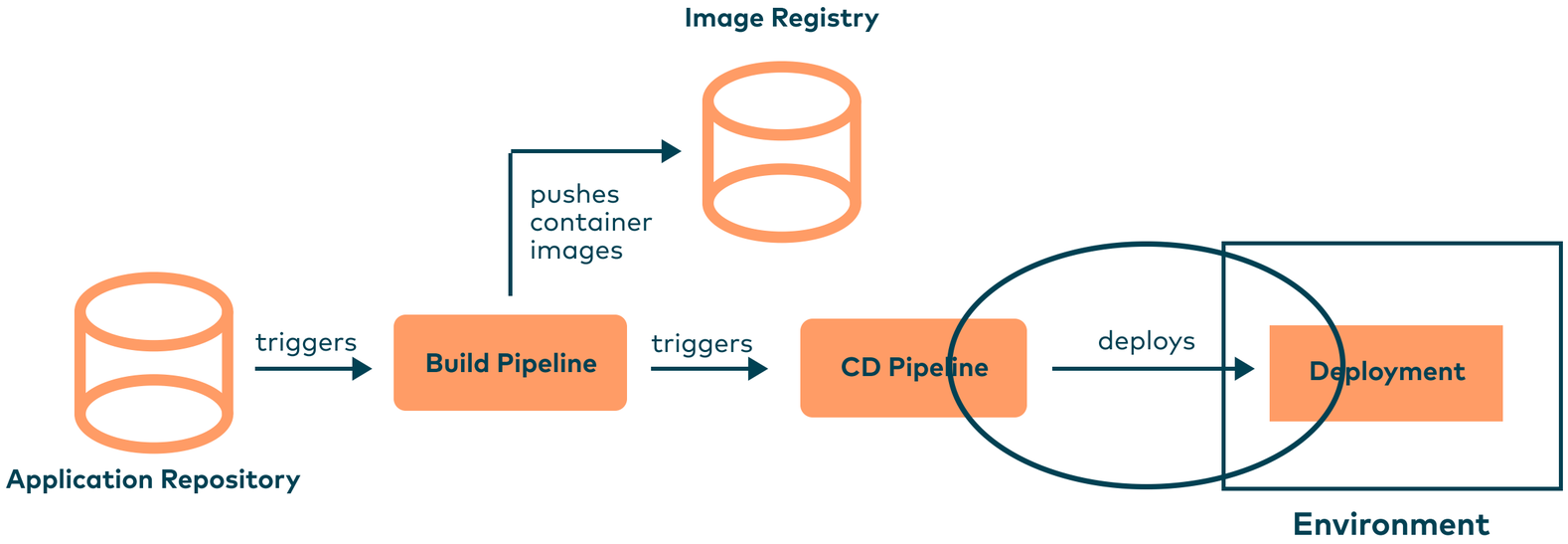
GitOps Workflow



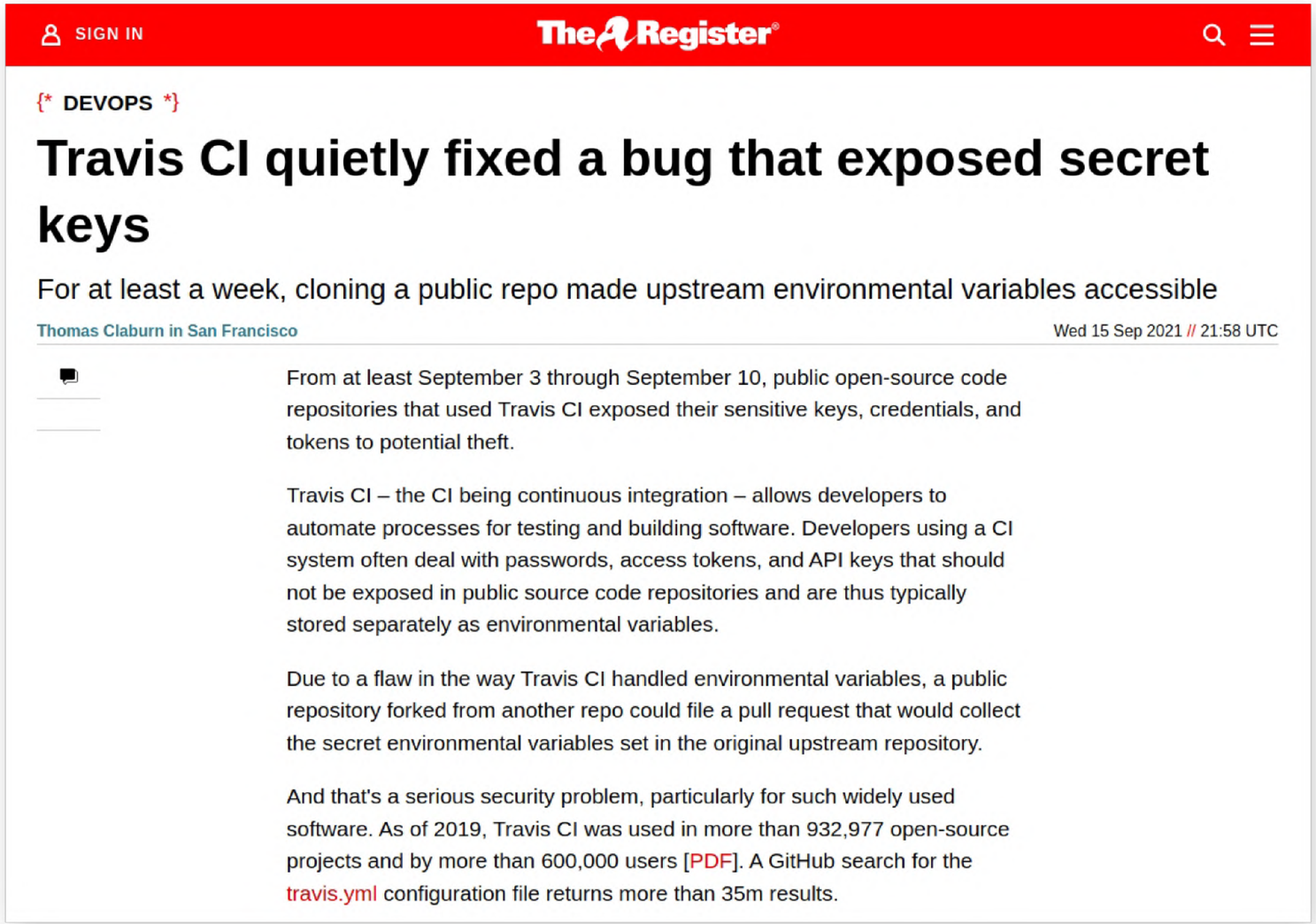
typische CI/CD Workflows



Push-basiertes Deployment



Travis CI Leak



SIGN IN

The Register

{* DEVOPS *}

Travis CI quietly fixed a bug that exposed secret keys

For at least a week, cloning a public repo made upstream environmental variables accessible

Thomas Claburn in San Francisco

Wed 15 Sep 2021 // 21:58 UTC

From at least September 3 through September 10, public open-source code repositories that used Travis CI exposed their sensitive keys, credentials, and tokens to potential theft.

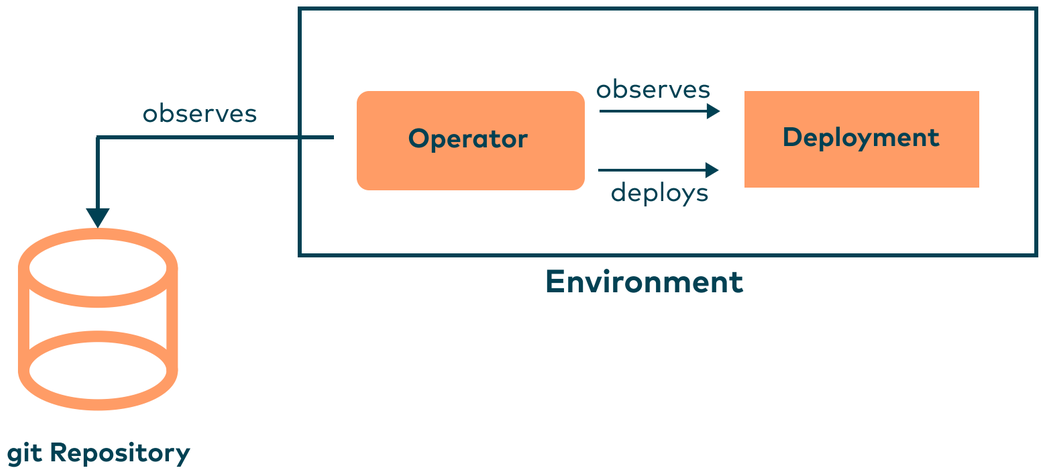
Travis CI – the CI being continuous integration – allows developers to automate processes for testing and building software. Developers using a CI system often deal with passwords, access tokens, and API keys that should not be exposed in public source code repositories and are thus typically stored separately as environmental variables.

Due to a flaw in the way Travis CI handled environmental variables, a public repository forked from another repo could file a pull request that would collect the secret environmental variables set in the original upstream repository.

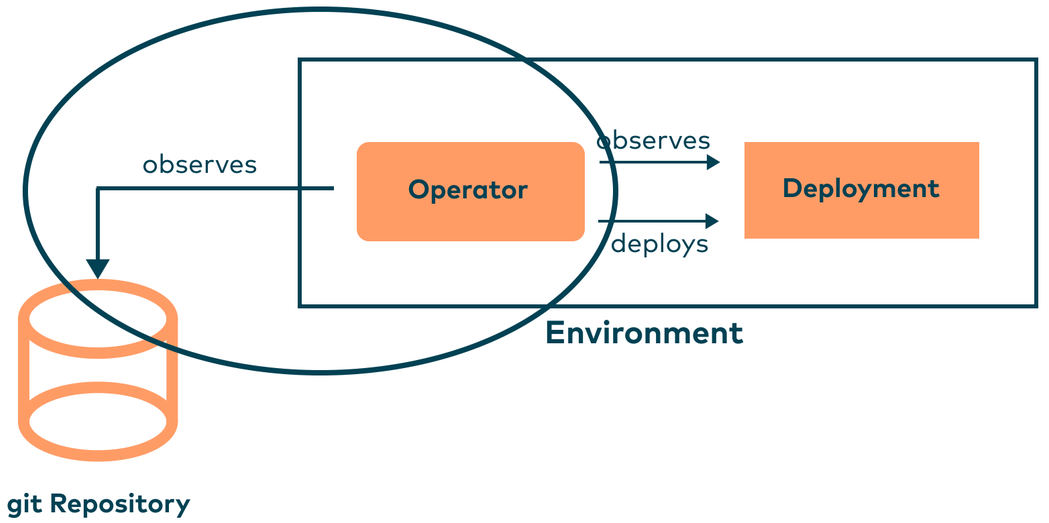
And that's a serious security problem, particularly for such widely used software. As of 2019, Travis CI was used in more than 932,977 open-source projects and by more than 600,000 users [PDF]. A GitHub search for the travis.yml configuration file returns more than 35m results.

Travis CI quietly fixed a bug that exposed secret keys. (2021). https://www.theregister.com/2021/09/15/travis_ci_leak/

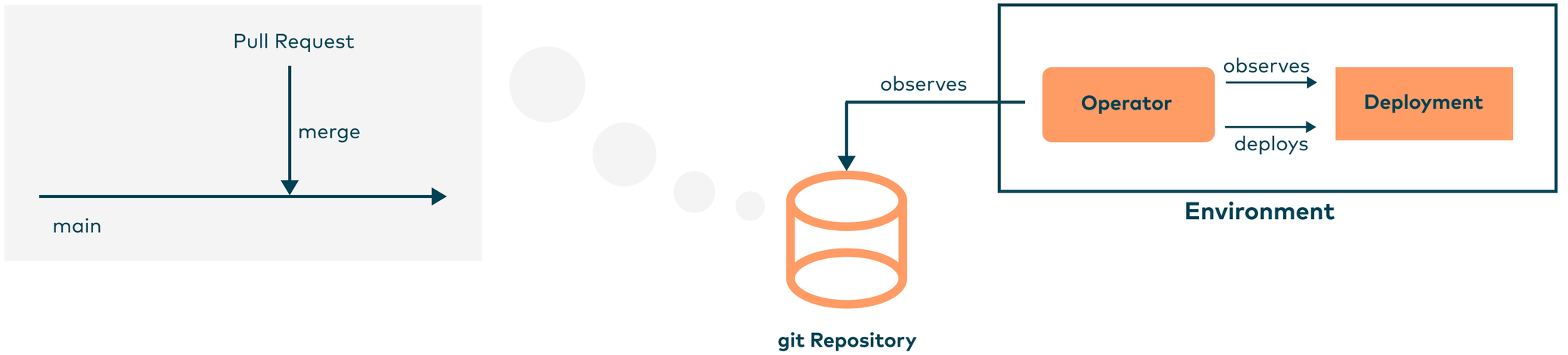
Pull-basiertes Deployment



Pull-basiertes Deployment



Pull Requests

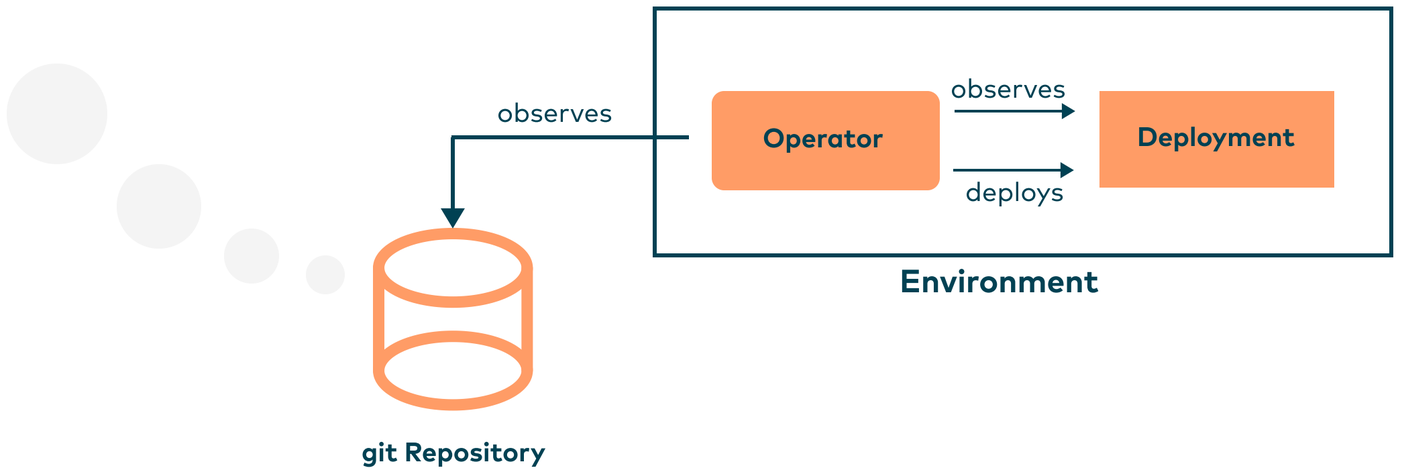


Git Historie

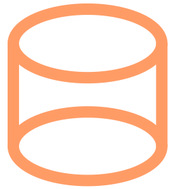
```
$ git log
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    Change version number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700
```

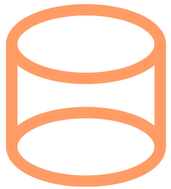


Repositories



Application Repository

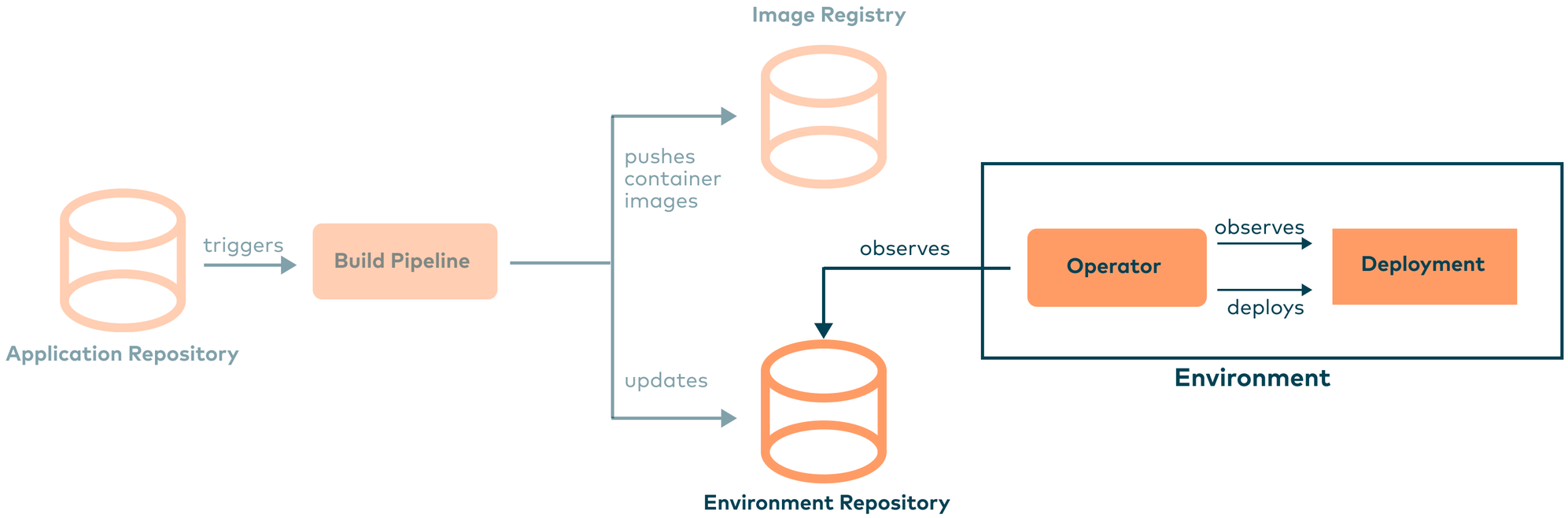
- Source Code
- CI Pipeline Konfigurationen
- Container Konfiguration



Environment Repository

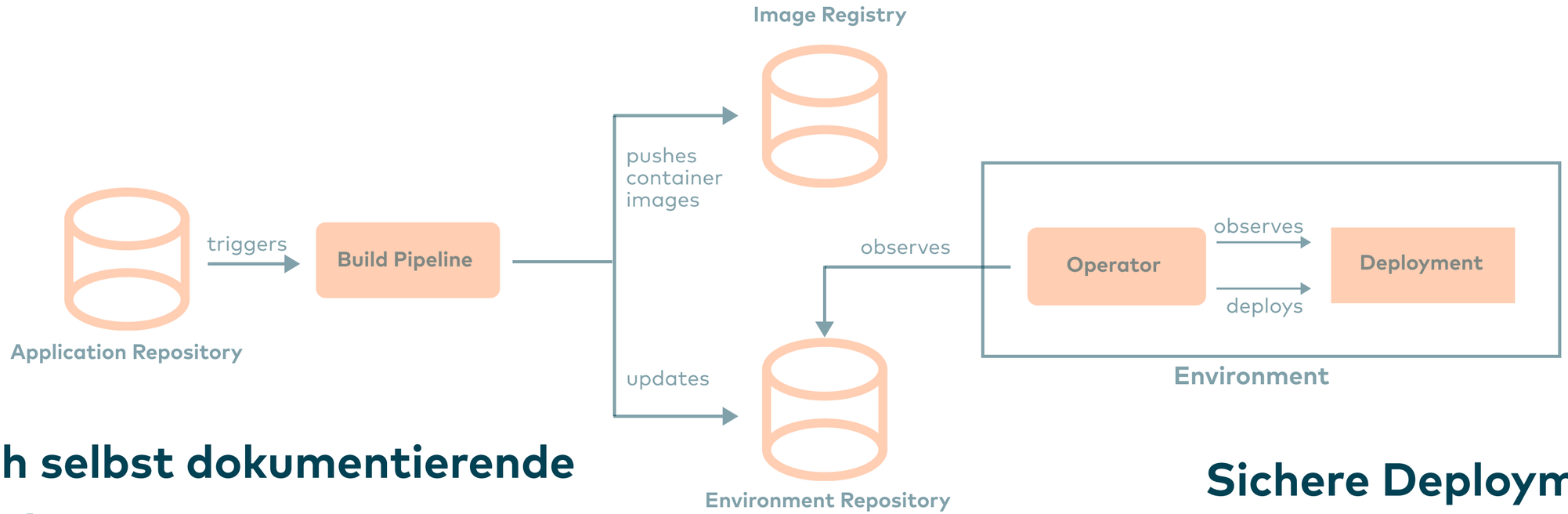
- Infrastruktur Konfiguration
- CI Pipeline Konfigurationen

Environment Repository



GitOps Workflow

Leichteres Staffing und Schnelleres Onboarding



Sich selbst dokumentierende Deployments

Sichere Deployments

Einfache und schnelle Fehlerbehebung

Missverständnisse

“GitOps ist auch nur ein neues Wort für DevOps”

GitOps

Techn. Implementierung

DevOps

Gründe für CI/CD

GitOps

Techn. Implementierung
Konkreter Prozess

DevOps

Gründe für CI/CD
Kultur und Methoden

"GitOps ist auch nur Infrastructure-as-Code"

**“A key part of GitOps is the idea of *environments-as-code*:
describing your deployments declaratively using files
(for example, Kubernetes manifests)
stored in a Git repository. ”**

https://cloud.google.com/kubernetes-engine/docs/tutorials/gitops-cloud-build

Environment as Code

Ausführung durch Operator

Infrastructure as Code

autom. Ausführung

"Es sind GitOps Tools (für Kubernetes) nötig"

Plattform X

selbstgebauter Operator

Kubernetes

k8s-native GitOps Tools

"GitOps ist auch nur eine bessere Deployment-Pipeline"

GitOps

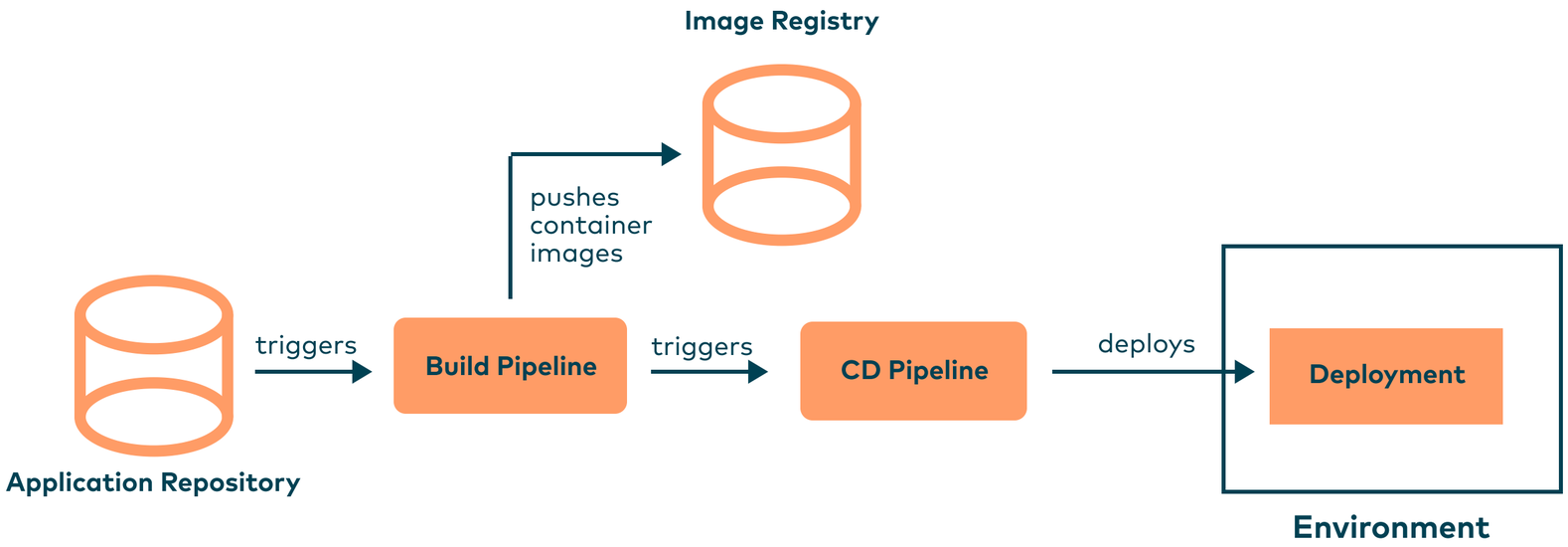
git Tool

CI/CD

separates Tool

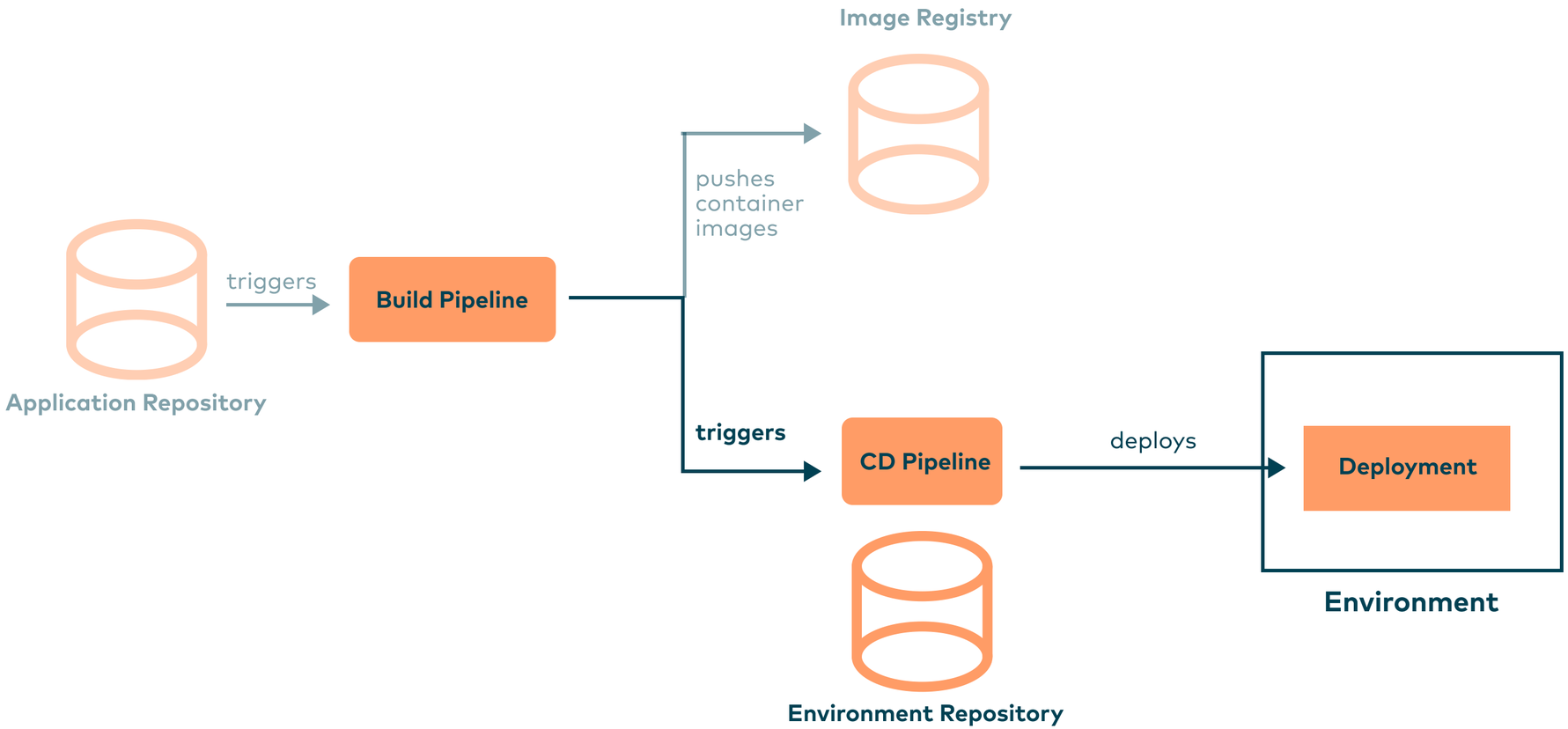
GitOps Transformation

typische CI/CD Workflows



Schritt 1
CI/CD Pipeline aufteilen

Schritt 1 - CI/CD Pipeline aufteilen

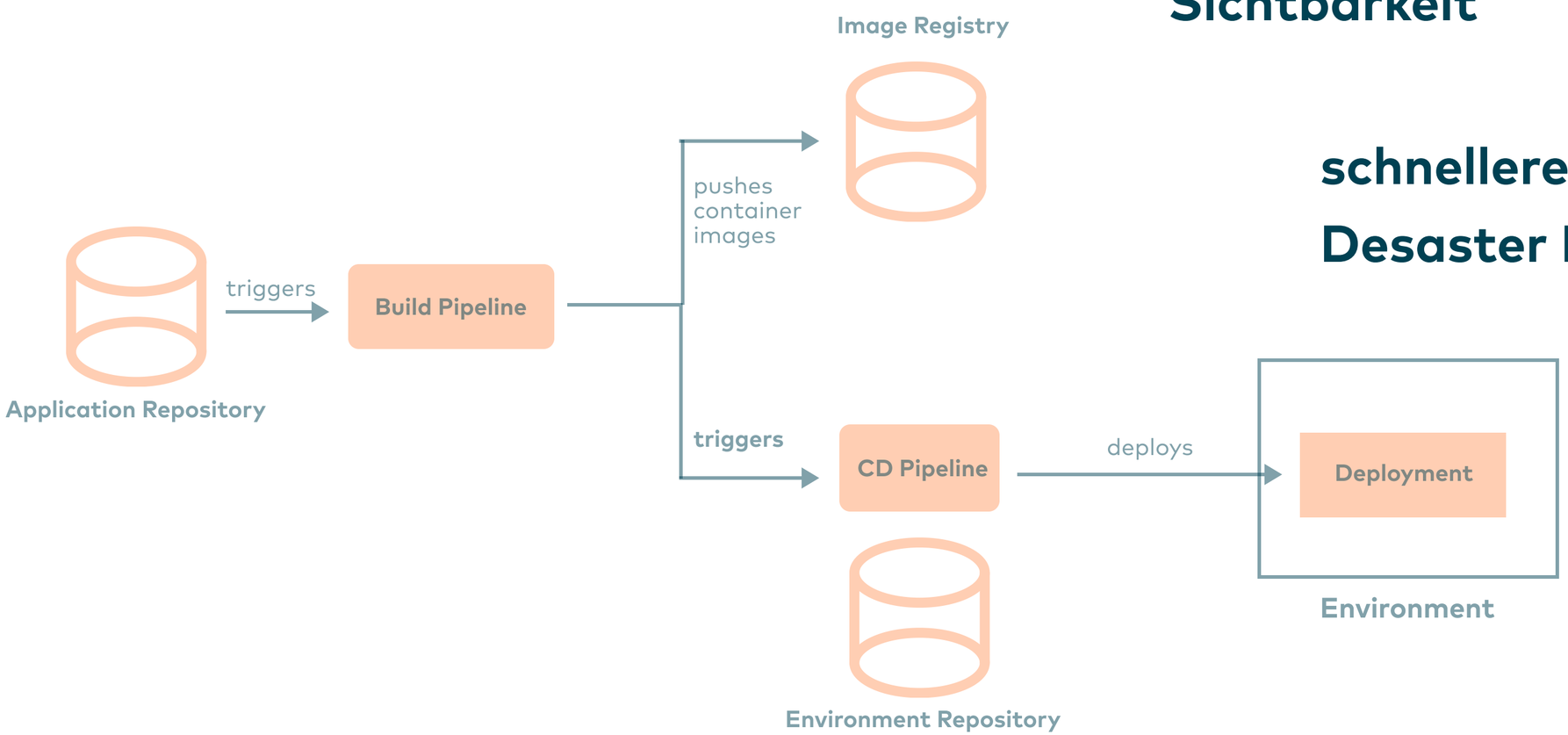


Beispiel-Setup:

- https://github.com/gitops-tech/example-application
- https://github.com/gitops-tech/example-environment

Workflow Dispatch:
https://docs.github.com/en/actions/using-workflows/events-that-trigger-workflows#workflow_dispatch

Schritt 1 - CI/CD Pipeline aufteilen



Sichtbarkeit

schnelleres und vollständiges Desaster Recovery

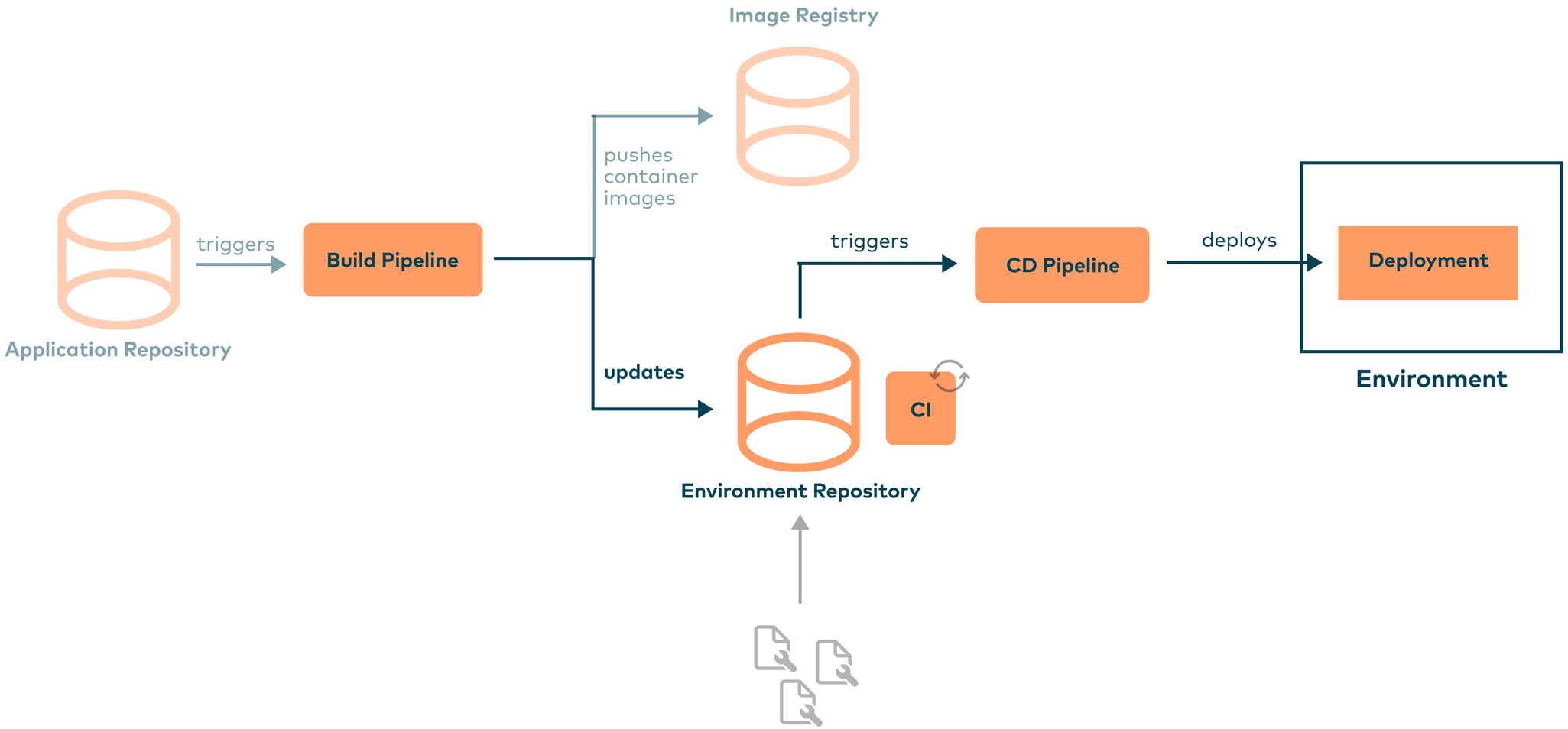
Beispiel-Setup:

- https://github.com/gitops-tech/example-application
- https://github.com/gitops-tech/example-environment

Workflow Dispatch:
https://docs.github.com/en/actions/using-workflows/events-that-trigger-workflows#workflow_dispatch

Schritt 2
Konfiguration zentralisieren

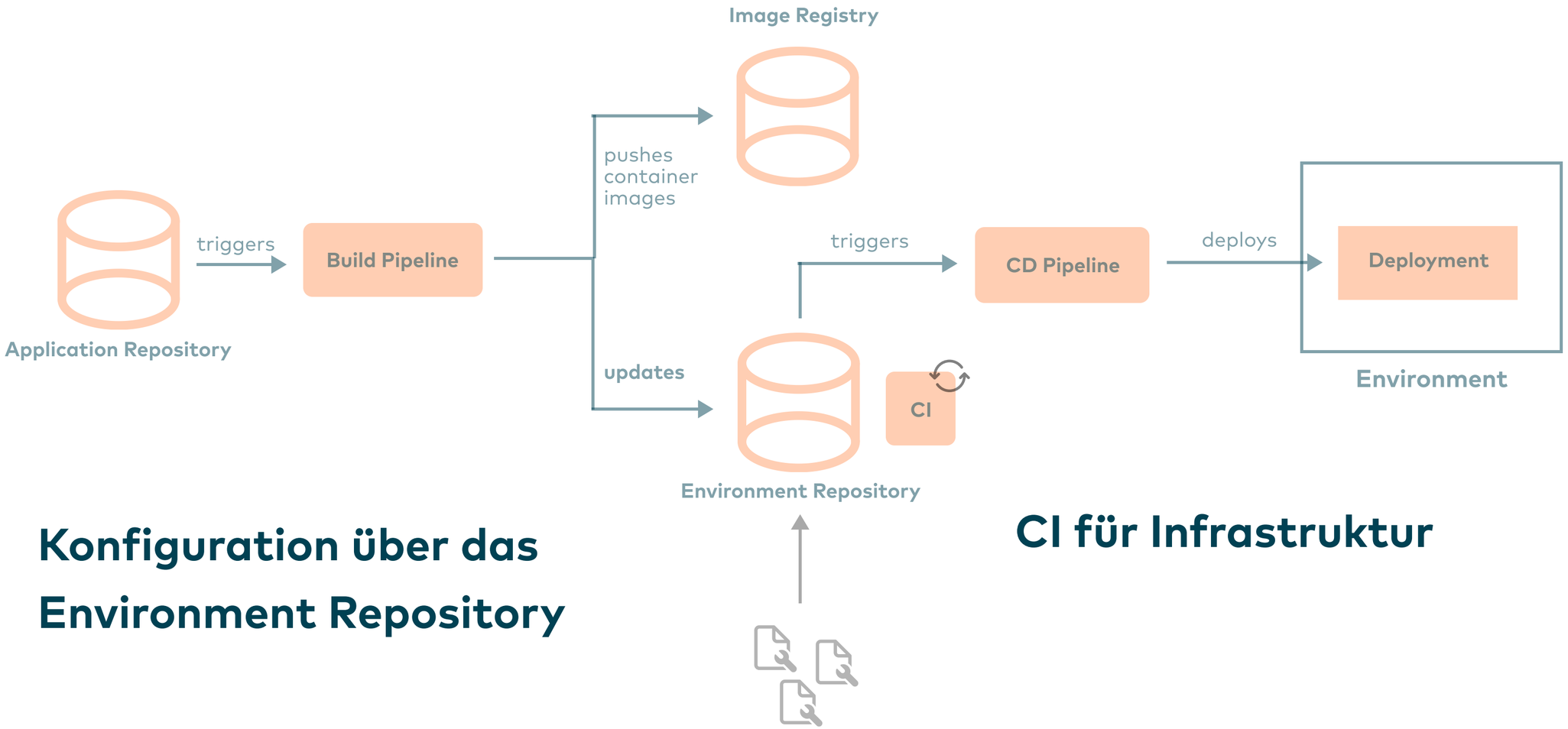
Schritt 2 - Konfiguration zentralisieren



Infra-CI Tooling:

- https://www.kubestack.com/framework
- https://kind.sigs.k8s.io/

Schritt 2 - Konfiguration zentralisieren



Konfiguration über das Environment Repository

CI für Infrastruktur

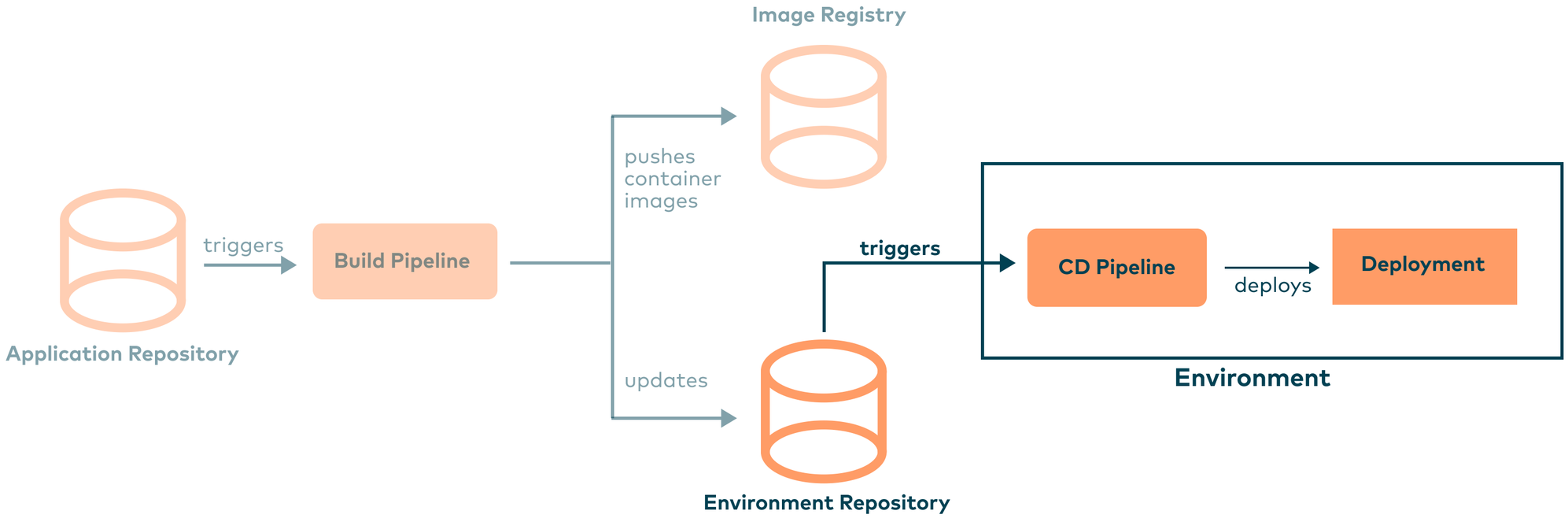
Infra-CI Tooling:

- https://www.kubestack.com/framework
- https://kind.sigs.k8s.io/

Schritt 3

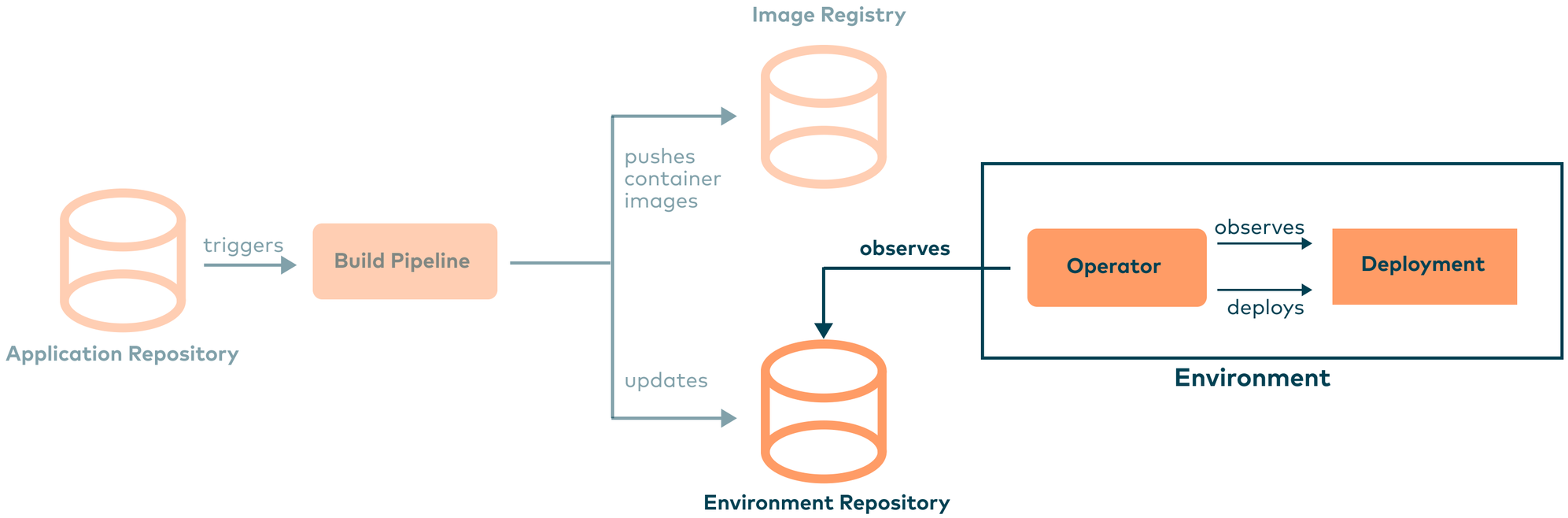
CD Pipeline in die Zielumgebung heben

Schritt 3 - CD Pipeline in die Zielumgebung heben Zwischenlösung

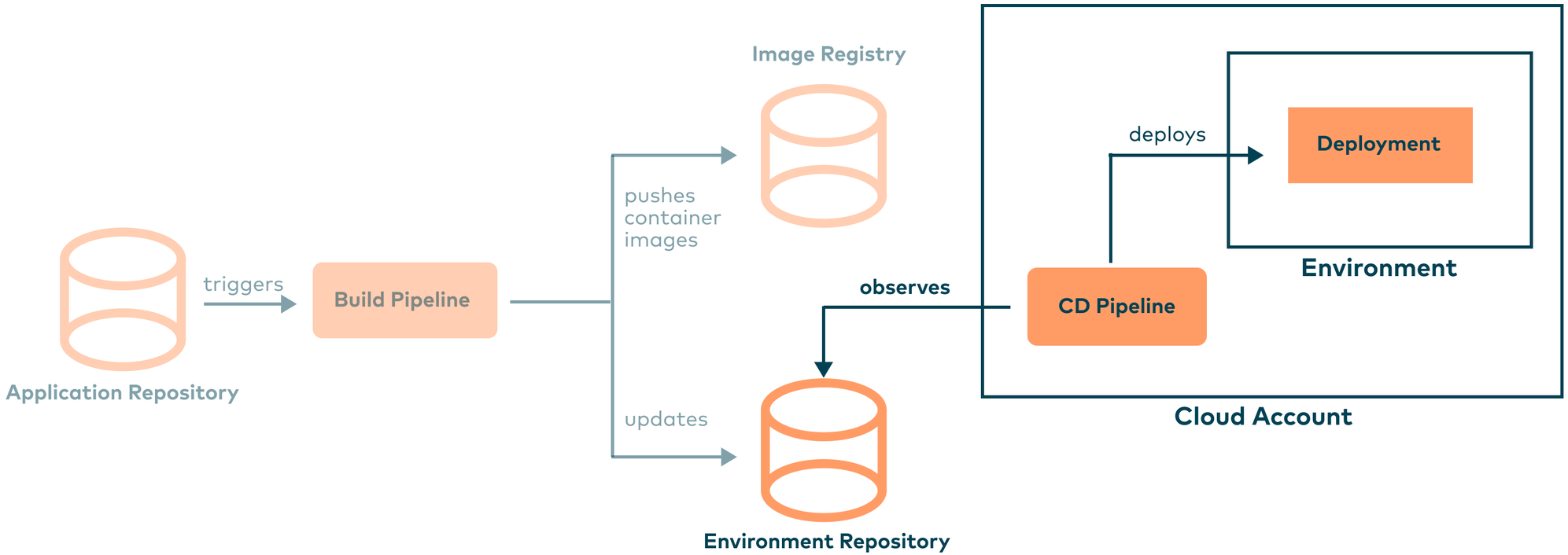


Schritt 3 - CD Pipeline in die Zielumgebung heben

GitOps Operator

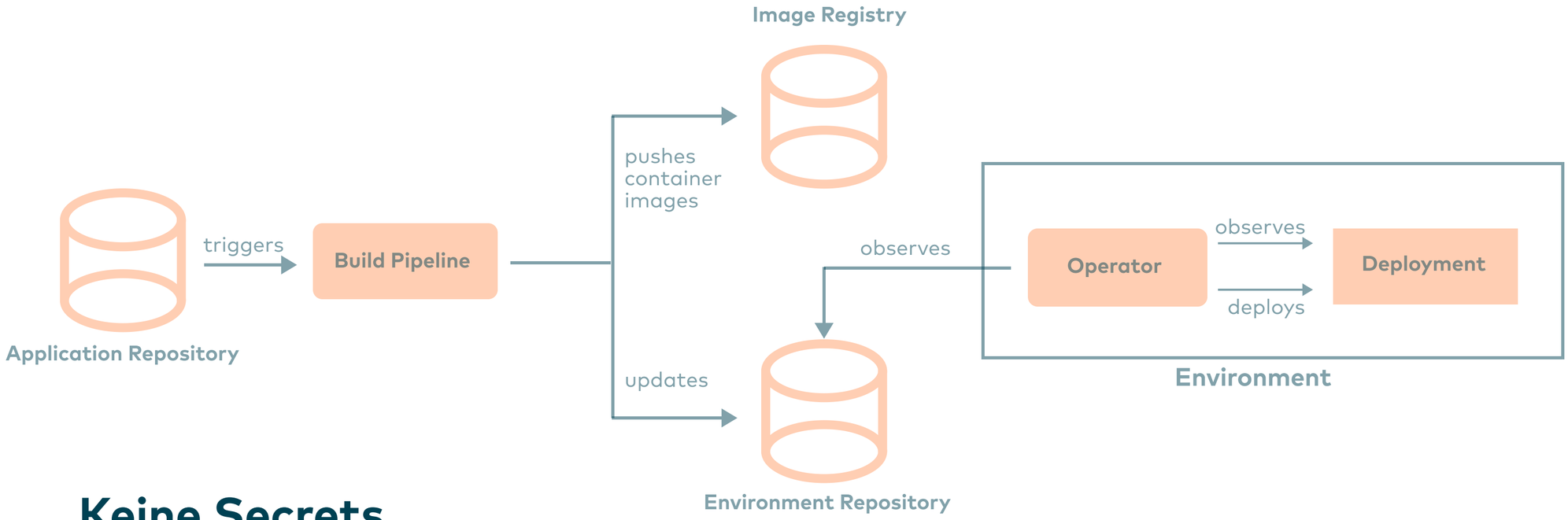


Schritt 3 - CD Pipeline in die Zielumgebung heben Alternative



Google Cloud Build Tutorial:
https://cloud.google.com/kubernetes-engine/docs/tutorials/gitops-cloud-build

GitOps Workflow



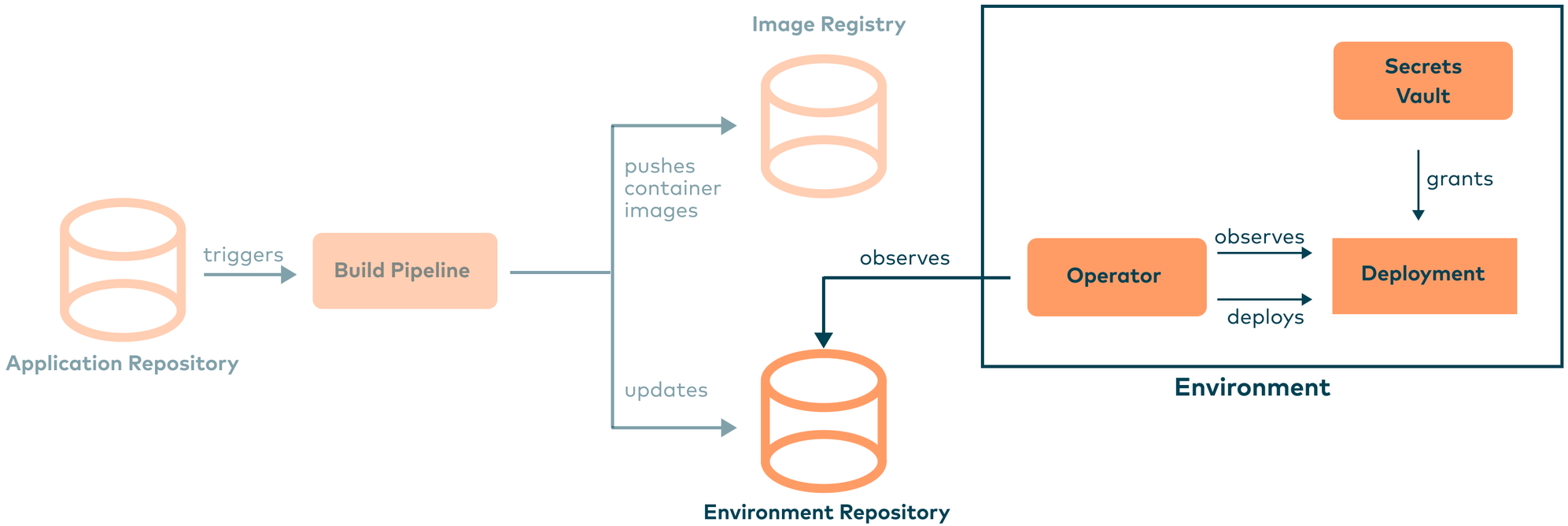
Keine Secrets
in der Pipeline

Nur externe
CI Pipelines

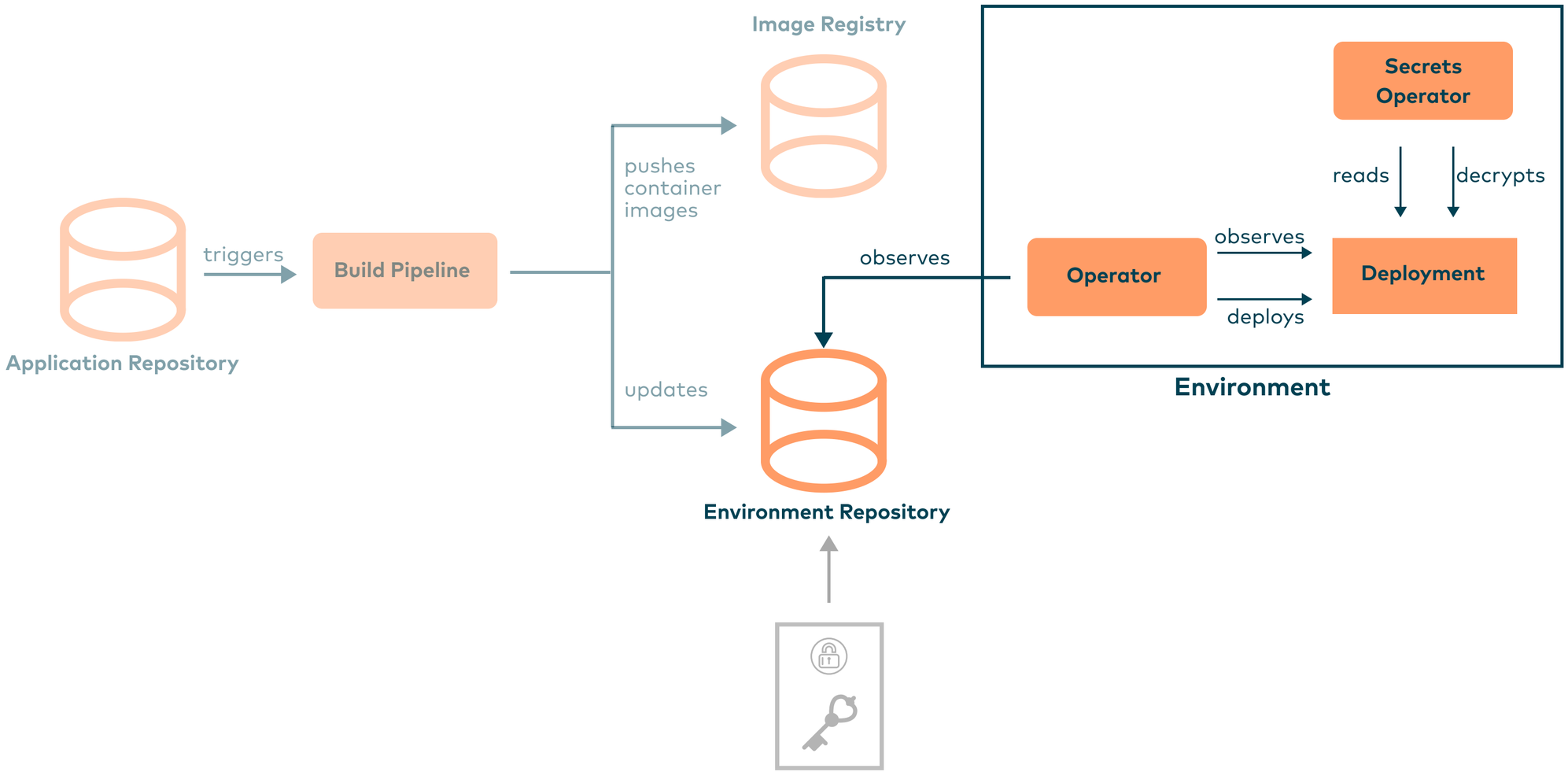
Fallstricke

Secrets Management ist nicht enthalten

Secrets Vault

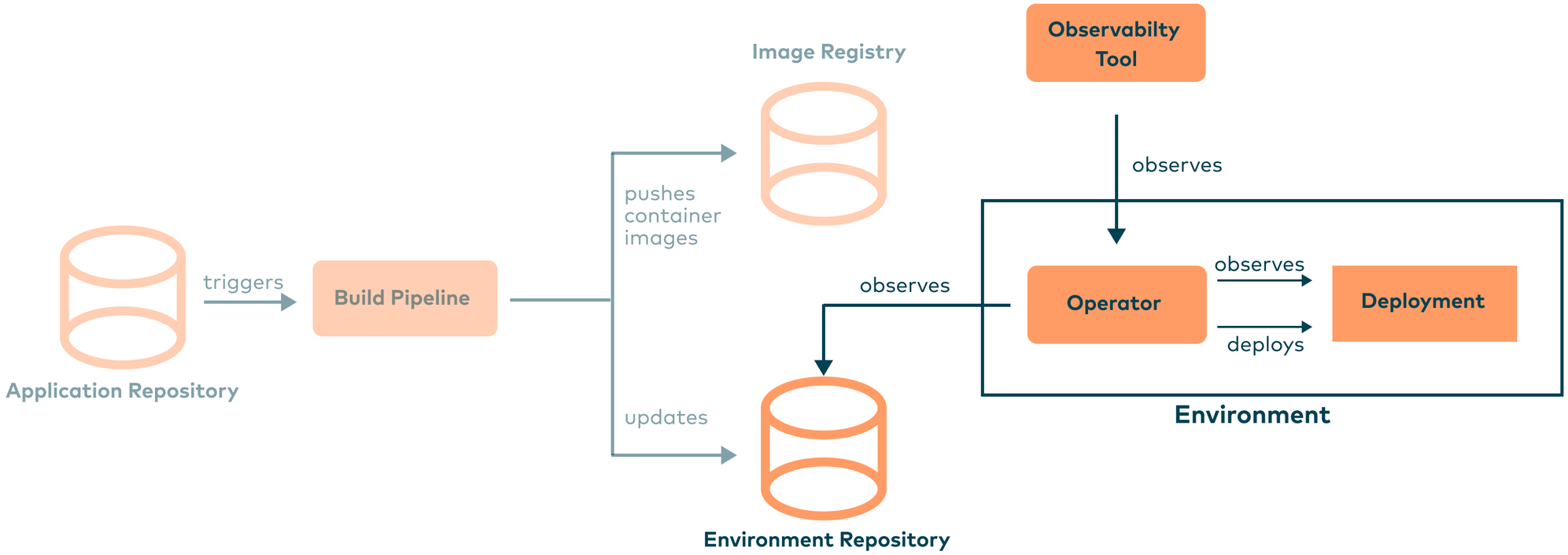


Secrets Operator



Die Pipeline ist weniger rot

Observabilty

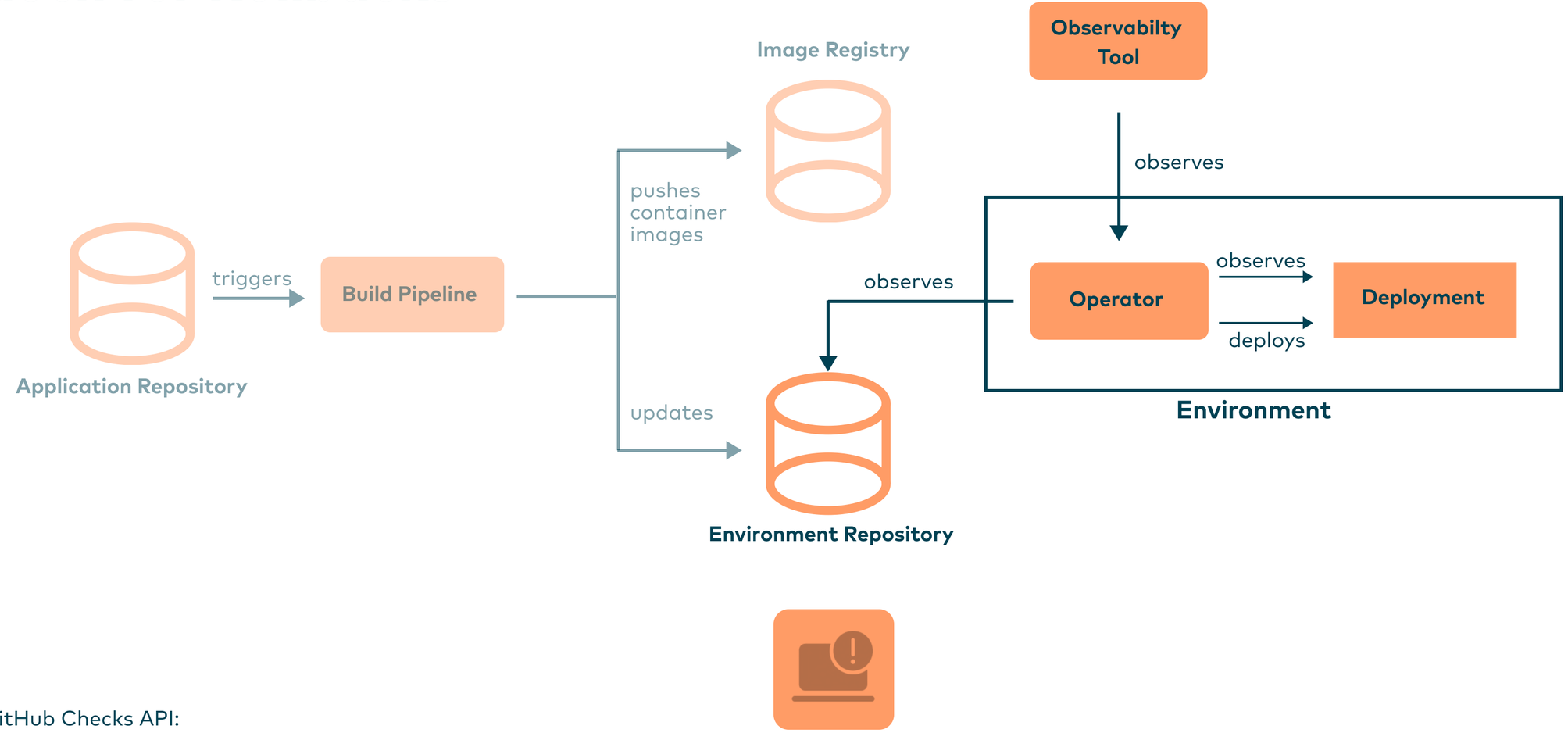


GitHub Checks API:
https://docs.github.com/en/rest/reference/checks

Bi-direktionales Rollback

Observabilty

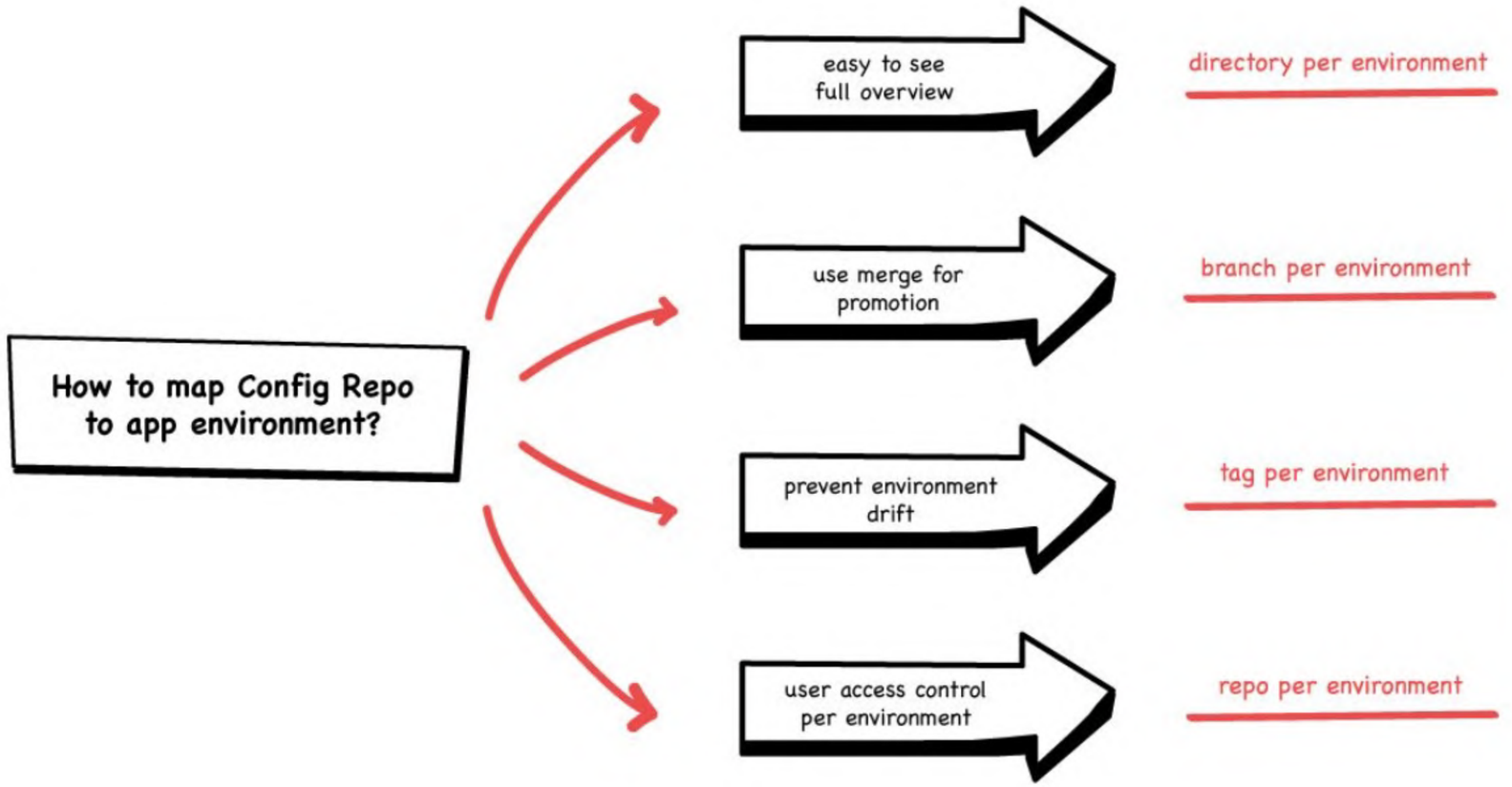
auch für Rollbacks



GitHub Checks API:
https://docs.github.com/en/rest/reference/checks

Abbildung multipler Stages

Promotion



Miell, Ian. (2021). GitOps: What You Need to Know Now.
https://info.container-solutions.com/what-is-gitops-ebook

Wohin mit den Deployment Manifesten?

Referenzen

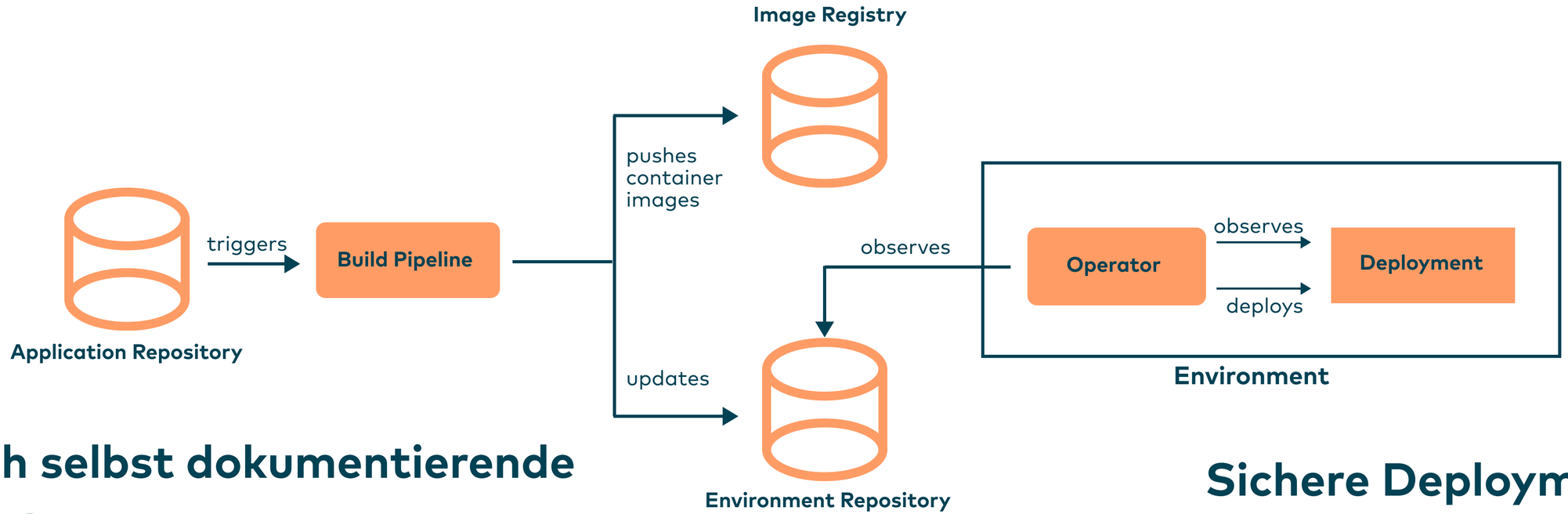
```yaml
apiVersion: source.toolkit.fluxcd.io/v1beta1
kind: GitRepository
metadata:
  name: podinfo
  namespace: default
spec:
  interval: 1m
  url: https://github.com/stefanprodan/podinfo
  ref:
    branch: master
```

Flux CD - Git Repository Resource:
https://fluxcd.io/docs/components/source/gitrepositories

Fazit

GitOps Workflow

Leichteres Staffing und Schnelleres Onboarding



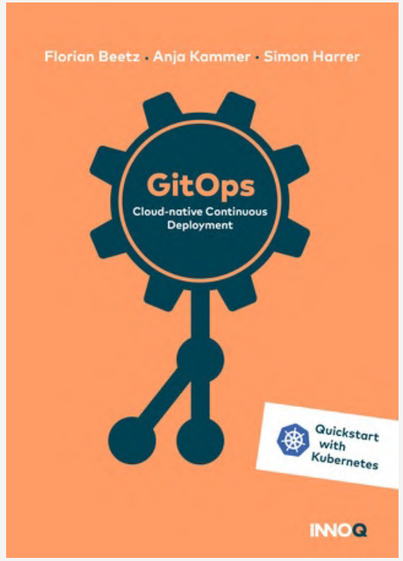
Sich selbst dokumentierende Deployments

Sichere Deployments

Einfache und schnelle Fehlerbehebung

Empfehlungen

www.innoq.com



gitops.tech

Bewerte diesen Vortrag

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 333660

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstr. 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorgraben 11
90402 Nürnberg