

# Tag 3: GitOps, Docker in der Entwicklung und Deployment-Strategien

19.06.2024, Daniel Krämer & Malte Fischer

© Copyright 2024 anderScore GmbH

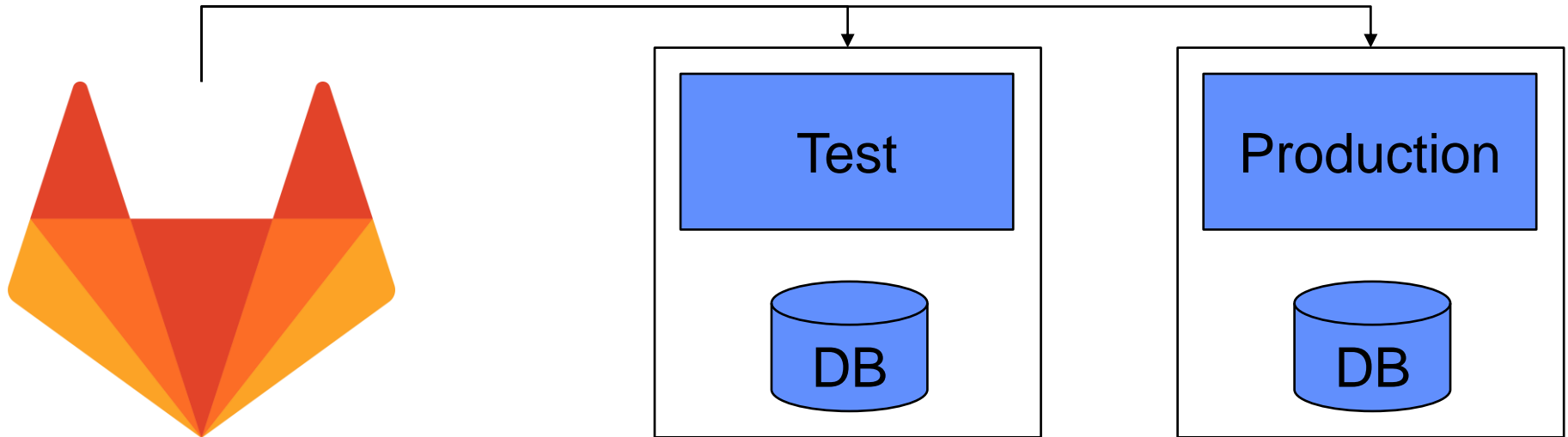
- **Tag 1 – Einführung in Git und GitLab, Git-Workflow im Team**
  - Einführung & Kursüberblick
  - Grundlagen von Git
  - Git Rebase und Merge-Strategien
  - Git Remote
  - Grundlagen von GitLab
  - Git-Workflow im Team
- **Tag 2 – Vertiefung Git-Workflow, CI/CD & GitLab CI**
  - Gitflow-Workflow
  - Tags, Releases & deren Verwaltung
  - GitLab-Runner
  - Einführung in GitLab CI/CD & gitlab.yml
- **Tag 3 – GitOps, Docker in der Entwicklung und Deployment-Strategien**
  - GitOps Grundlagen
  - Lokale Entwicklung mit Docker
  - Container/Docker-Registry
  - Erstellen von Release- und Tagged-Images
  - Möglichkeiten des Deployments & Verwaltung von Konfiguration
  - Abschlussübung & Diskussion

- **Tag 1 – Einführung in Git und GitLab, Git-Workflow im Team**
  - Einführung & Kursüberblick
  - Grundlagen von Git
  - Git Rebase und Merge-Strategien
  - Git Remote
  - Grundlagen von GitLab
  - Git-Workflow im Team
- **Tag 2 – Vertiefung Git-Workflow, CI/CD & GitLab CI**
  - Gitflow-Workflow
  - Tags, Releases & deren Verwaltung
  - GitLab-Runner
  - Einführung in GitLab CI/CD & gitlab.yml
- **Tag 3 – GitOps, Docker in der Entwicklung und Deployment-Strategien**
  - GitOps Grundlagen
  - Lokale Entwicklung mit Docker
  - Container/Docker-Registry
  - Erstellen von Release- und Tagged-Images
  - Möglichkeiten des Deployments & Verwaltung von Konfiguration
  - Abschlussübung & Diskussion

# Möglichkeiten des **DEPLOYMENT**

## Deployment über GitLab

- Sehr flexibel konfigurierbar
- Viele Strategien möglich
- Fast jeder Workflow abbildbar

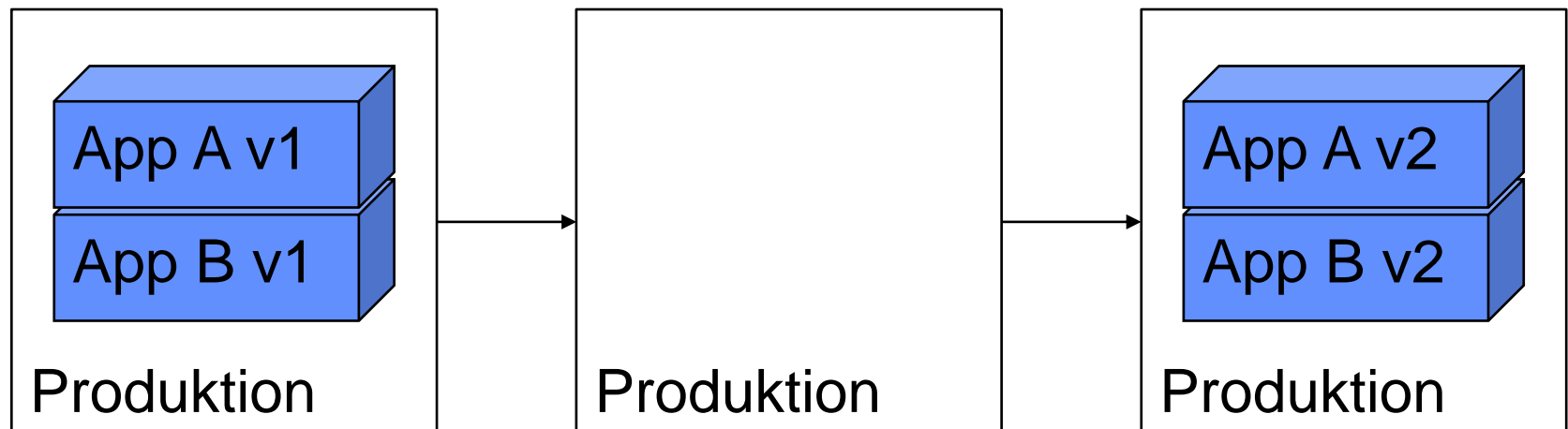


## Verschiedene Deployment Strategien

- Recreate Deployment
- Blue-Green Deployment
- Canary Deployment
- Shadow Deployment

## Recreate Deployment

- Altsystem wird offline genommen
- Neusystem wird danach deployed und gestartet



## Vorteile

- Einfache Implementierung
- Günstig

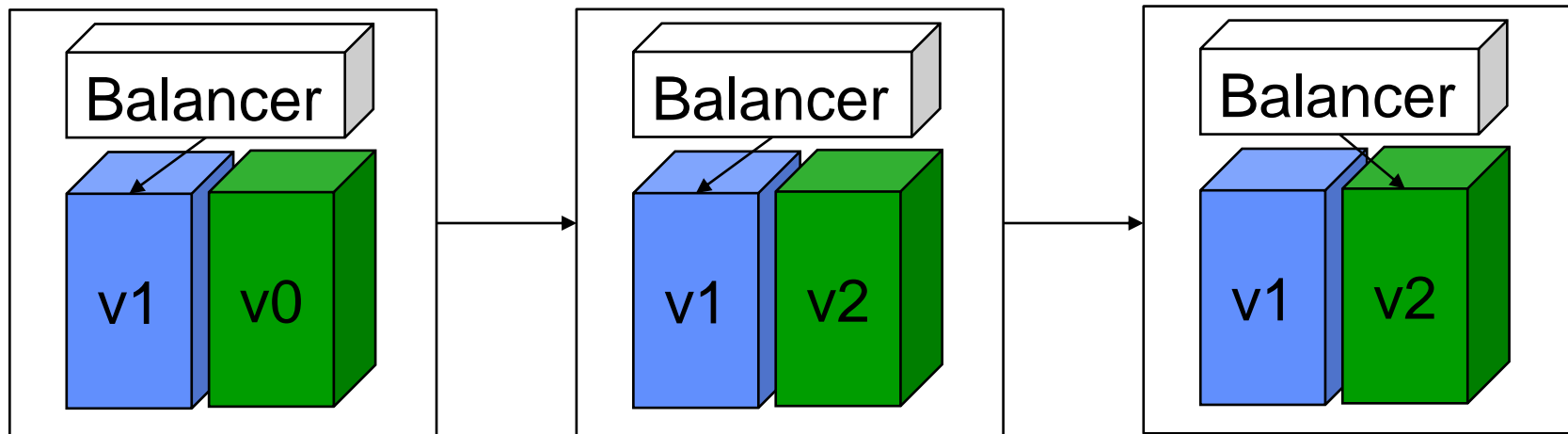
## Nachteile

- Downtime
- Aufwändiges Rollback
- Kein partielles Deployment möglich



## Blue-Green Deployment

- Alt und Neu werden parallel gehostet
- Anfragen gehen erst nur teilweise aufs neue System
- Alt System für Rollback und weitere Updates nutzen



## Vorteile

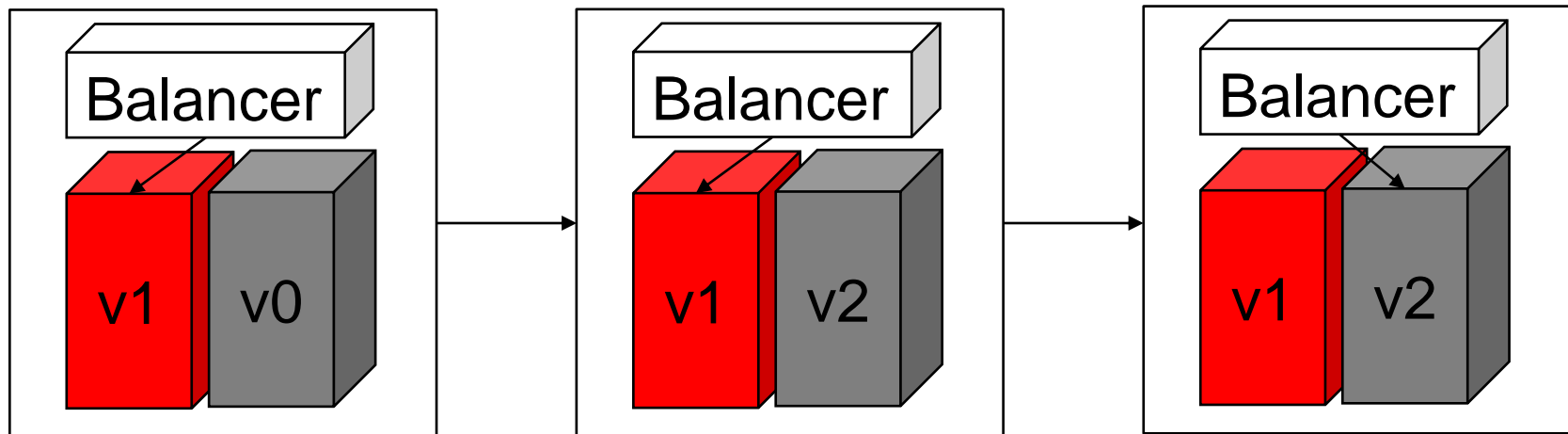
- Keine Downtime
- Einfaches Rollback
- Partielles Deployment
- Rolling Updates

## Nachteile

- Aufwendige Implementierung
- Zufällige Zuweisung zu neuem System
- Teuer

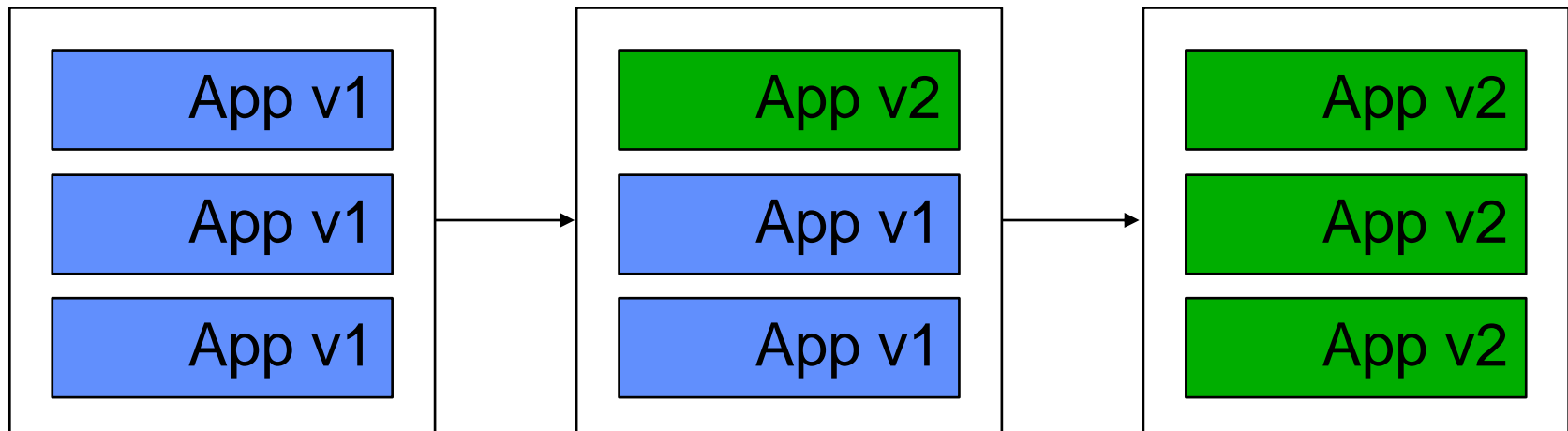
## Red-Black Deployment

- Sehr ähnlich zu Blue-Green
- Kompletter Wechsel zu Neusystem



## Canary Deployment

- Neue Version nur für bestimmten Teil der User released
- Test der Version mit Testgruppe



## Vorteile

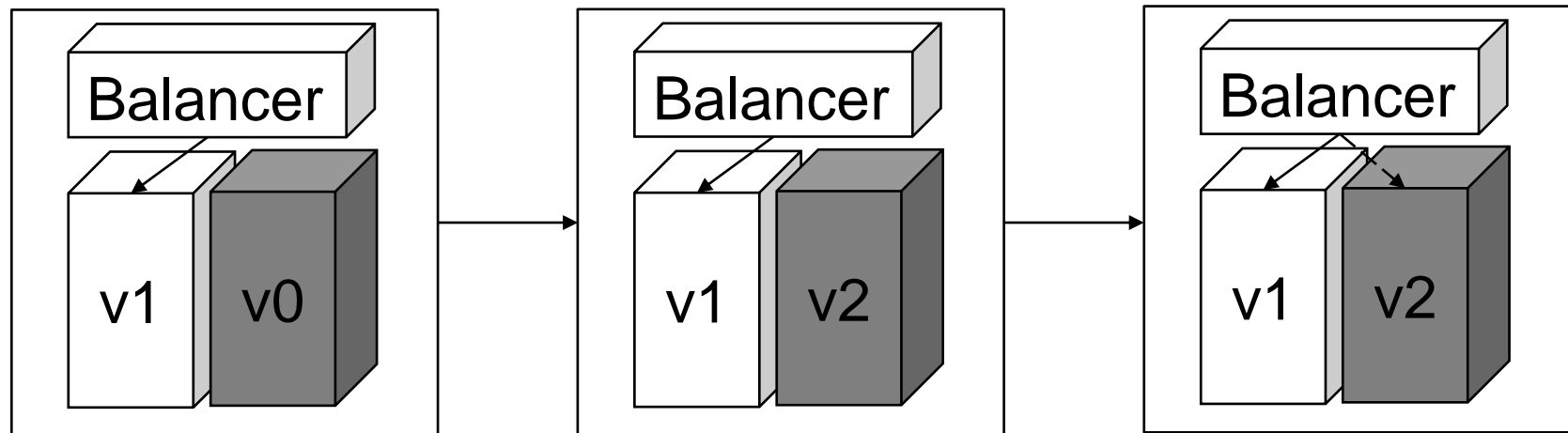
- Keine Downtime
- Partielles Deployment
- Gezielte Testgruppe

## Nachteile

- Aufwendige Implementierung
- Teuer
- Testgruppe nötig

## Shadow Deployment

- Neusystem wird parallel zu Altsystem gehosted
- Anfragen werden vom Altsystem bearbeitet
- Anfragen werden zum Neusystem gespiegelt



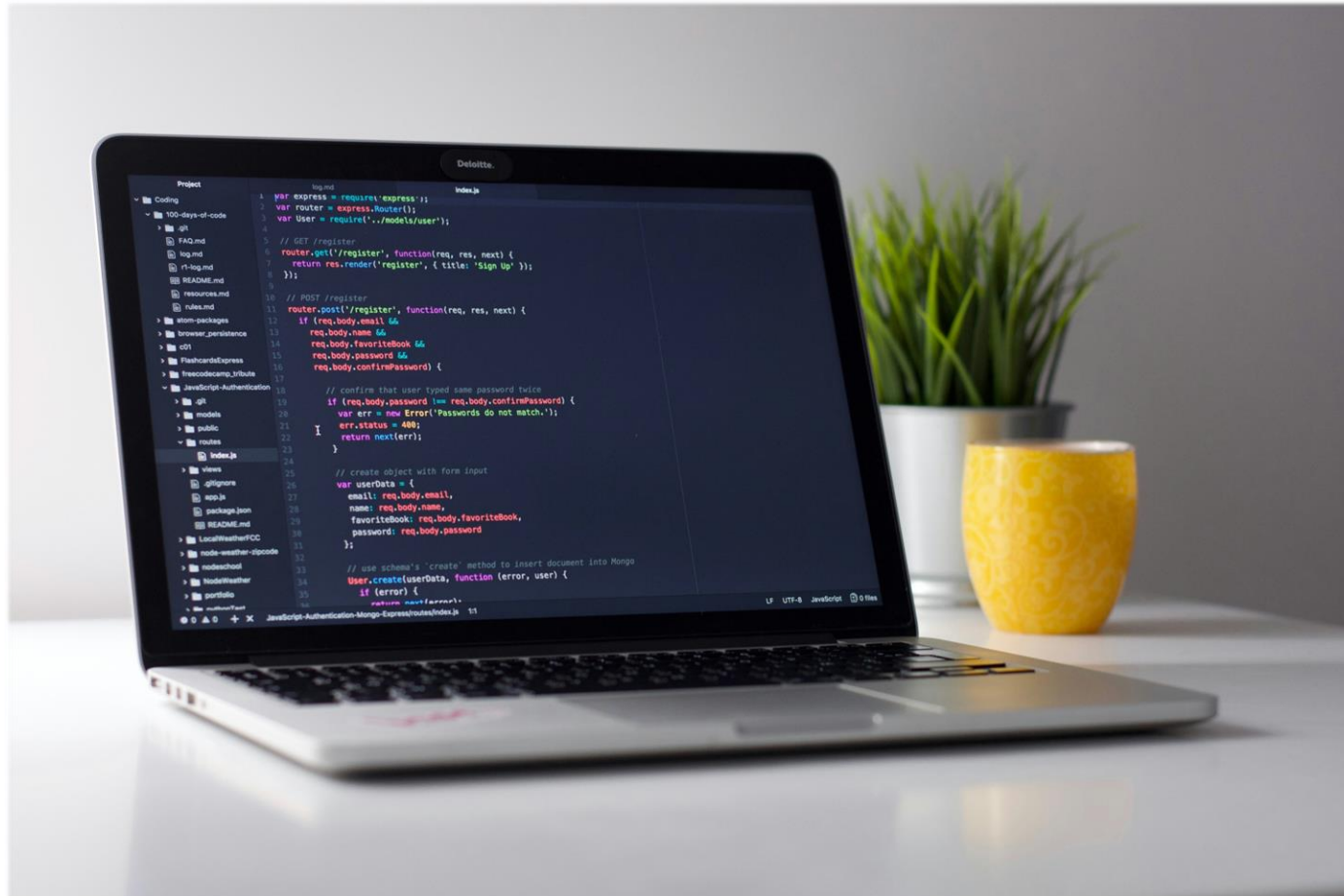
## Vorteile

- Keine Downtime
- Kein Rollback nötig
- Test unter Volllast möglich

## Nachteile

- Aufwendige Implementierung
- Teuer

## Variablen und Secrets





## Variablen und Secrets

- Abweichungen in Konfiguration zwischen Umgebungen
- Umgebungsvariablen
- Adressen/URLs
- Zugangsdaten
- Konfiguration von Docker Containern

## GitLab Variablen

4

4

4

Q Search or go to...

ISSUES

Merge requests 0

Manage >

Plan >

Code >

Build >

Secure >

Deploy >

Operate >

Monitor >

Analyze >

Settings >

General

Integrations

Webhooks

Access Tokens

Repository

Merge requests

CI/CD

Packages and registries

Monitor

Usage Quotas

Q Search page

General pipelines

Expand

Customize your pipeline configuration.

Auto DevOps

Expand

Automate building, testing, and deploying your applications based on your continuous integration and delivery configuration. [How do I get started?](#)

Runners

Expand

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Artifacts

Expand

A job artifact is an archive of files and directories saved by a job when it finishes.

Variables

Collapse

Variables store information that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more.](#)

Variables can be accidentally exposed in a job log, or maliciously sent to a third party server. The masked variable feature can help reduce the risk of accidentally exposing variable values, but is not a guaranteed method to prevent malicious users from accessing variables. [How can I make my variables more secure?](#)

Variables can have several attributes. [Learn more.](#)

- Protected: Only exposed to protected branches or protected tags.
- Masked: Hidden in job logs. Must match masking requirements.
- Expanded: Variables with \$ will be treated as the start of a reference to another variable.

CI/CD Variables </> 0

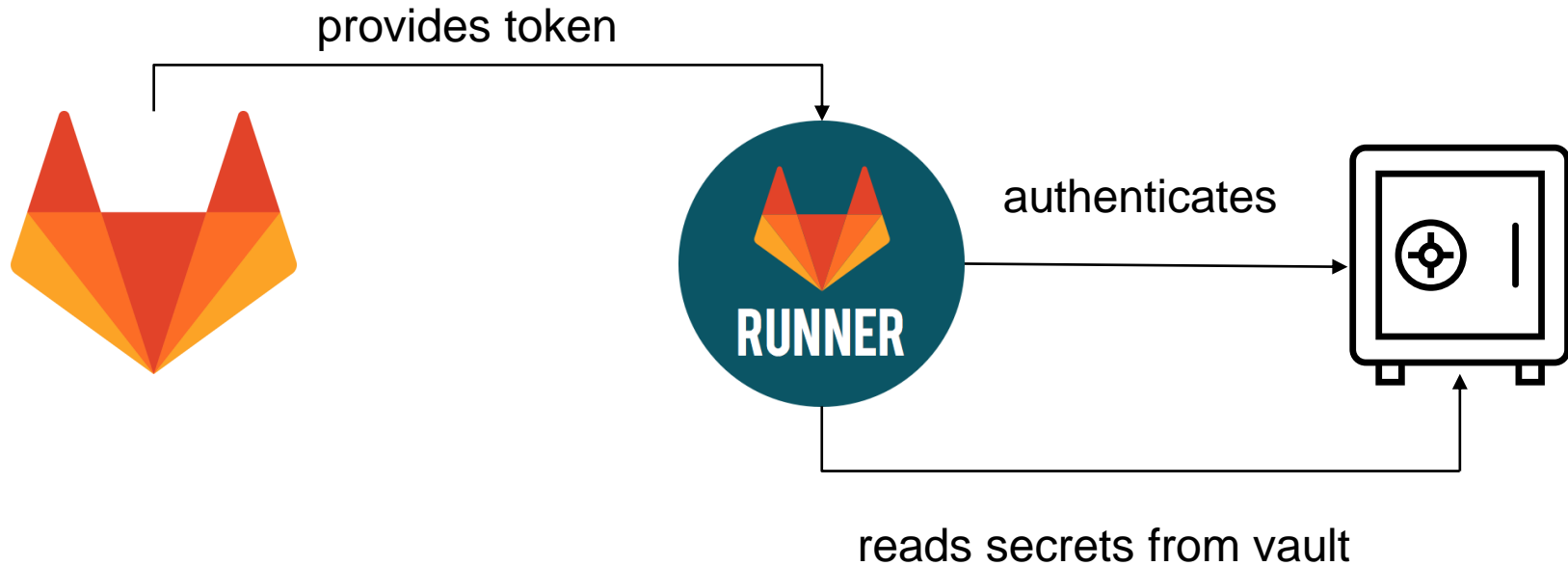
Add variable

Key ↑	Value	Environments	Actions
There are no variables yet.			

## GitLab Variablen

- Keine externen Tools
- Begrenzte Sicherheit
  - Jeder, der Projekteinstellungen bearbeiten kann, kann sie sehen
  - Können mit fehlerhafter Pipeline geloggt werden
- Müssen richtig konfiguriert sein
  - Visible oder masked
  - Protect Variable
- Können überschrieben werden

## Secret Vaults



## Secret Vaults

- Externe Provider
  - HashiCorp Vault, Azure Key Vault, Google Cloud Secret Manager
- Müssen eingebunden werden
- Können in mehreren Projekten verwendet werden
- Bieten mehr Features (Secret Rotation)
- Sicherer als GitLab Variablen