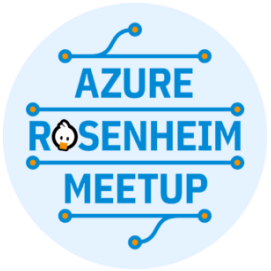
white duck

GitOps on Azure

Azure Rosenheim Meetup, 17.03.2022



Microsoft Partner

Microsoft

Gold Cloud Platform
Gold DevOps
Silver Application Development
Silver Security
Silver Application Integration

GitHub

Who we are?

Martin Brandl (white duck GmbH, Cloud Solution Architect & Azure MVP)
Twitter: @martin_jib
LinkedIn: linkedin.com/in/mbrandl/

Nico Meisenzahl (Senior Cloud & DevOps Consultant & Cloud & Data Management MVP)
Twitter: @nmeisenzahl
LinkedIn: https://www.linkedin.com/in/nicomeisenzahl/

Philip Welz (Senior Kubernetes & DevOps Engineer & GitLab Hero & CKA, CKAD & CKS)
Twitter: @philip_welz
LinkedIn: https://www.linkedin.com/in/philip-welz

twitter.com/whiteduck_gmbh

twitter.com/AzureMeetup

twitter.com/CloudNative_Ro

Housekeeping

- this meetup will be streamed on YouTube!
- want to participate?
  - join our meetup to get access to the Zoom meeting
    - https://www.meetup.com/Azure-Meetup-Rosenheim
  - we do also monitor the comments on YouTube

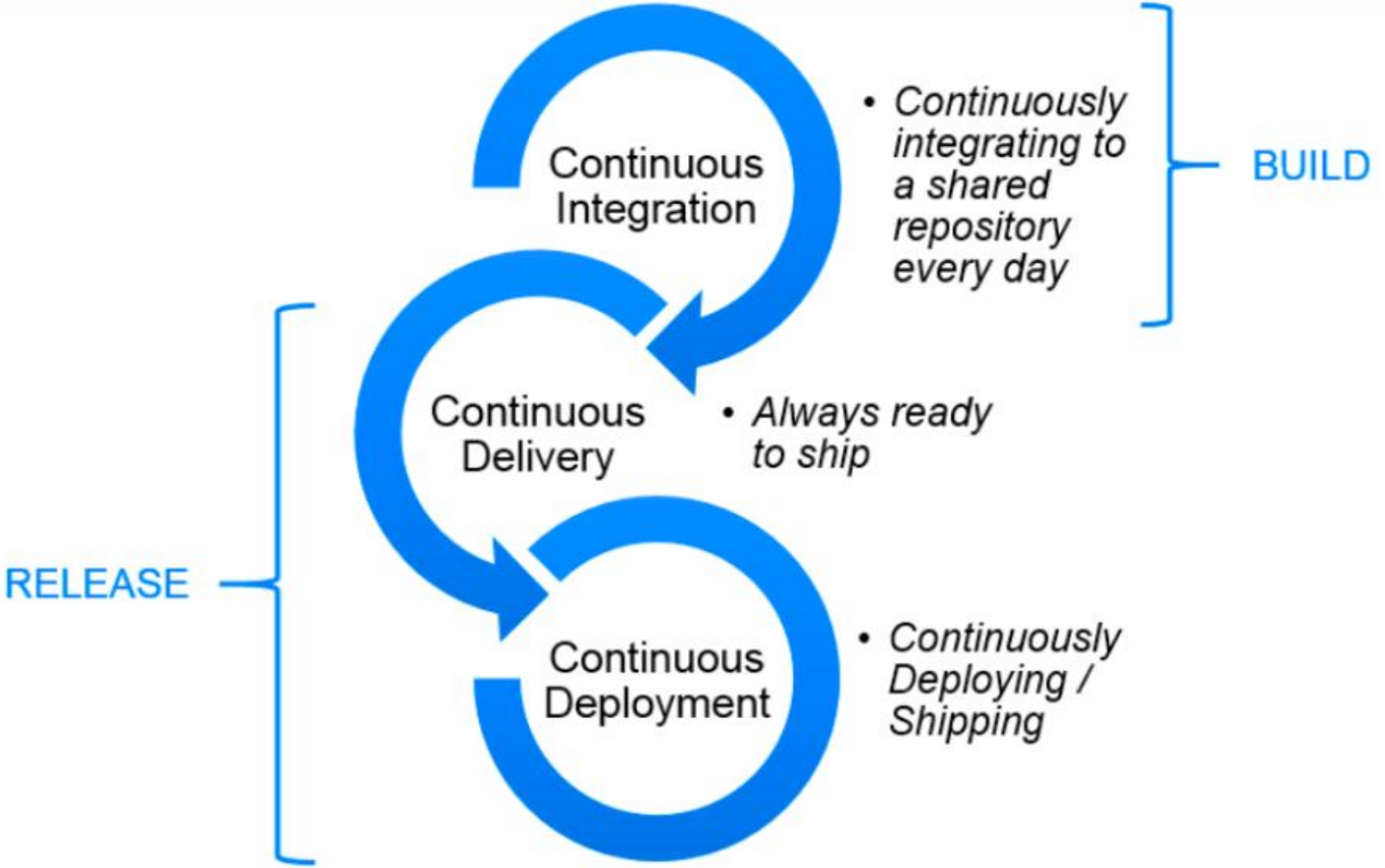
Agenda

- Introduction to GitOps
- Overview of Flux
- Flux on Azure
- Demo
- FAQ / Networking

GitOps to Azure

INTRODUCTION TO GITOPS

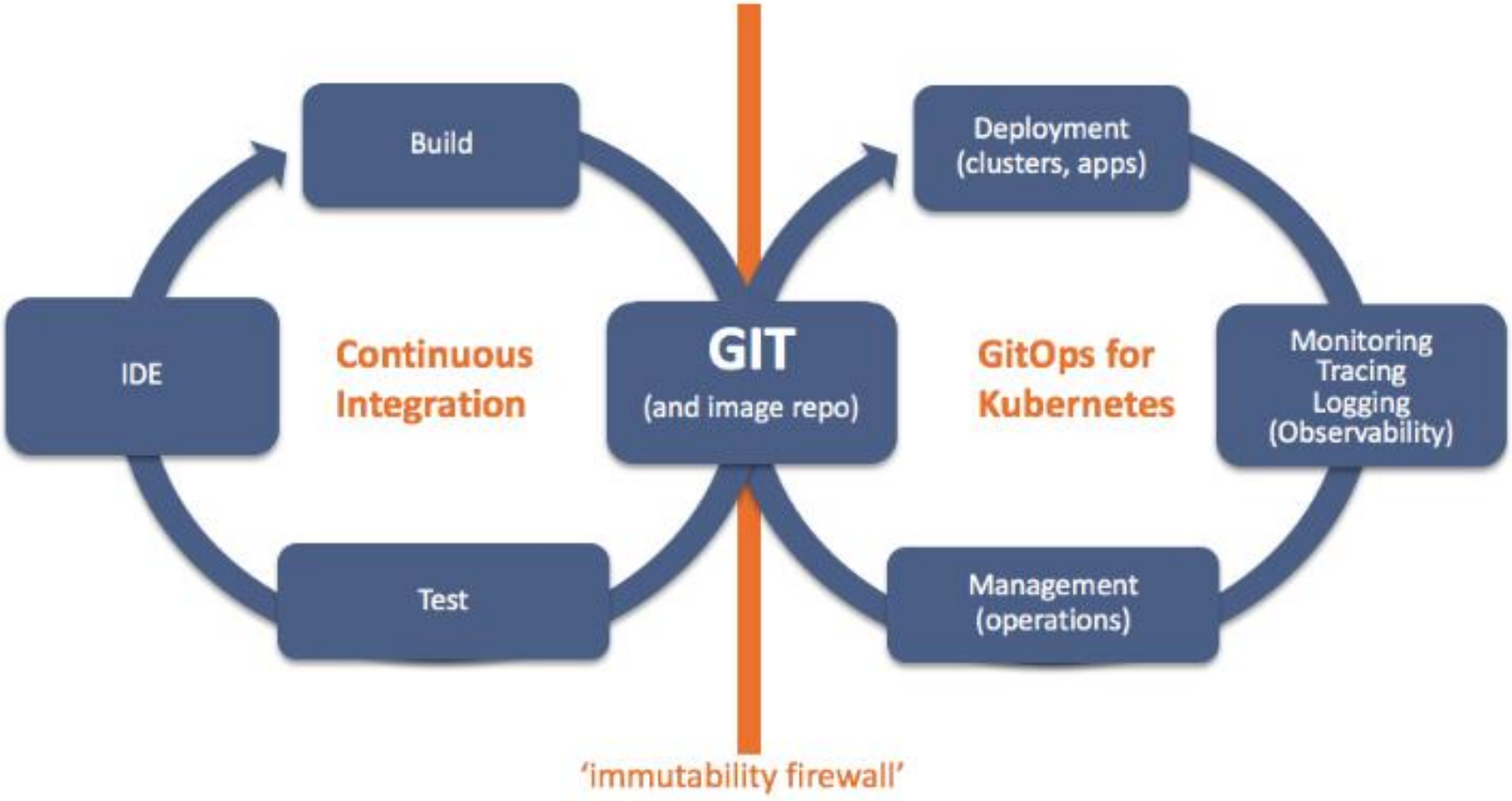
Continuous…



GitOps

Is a way to do Kubernetes cluster management and application delivery. GitOps works by using Git as a single source of truth for declarative infrastructure and applications.

GitOps



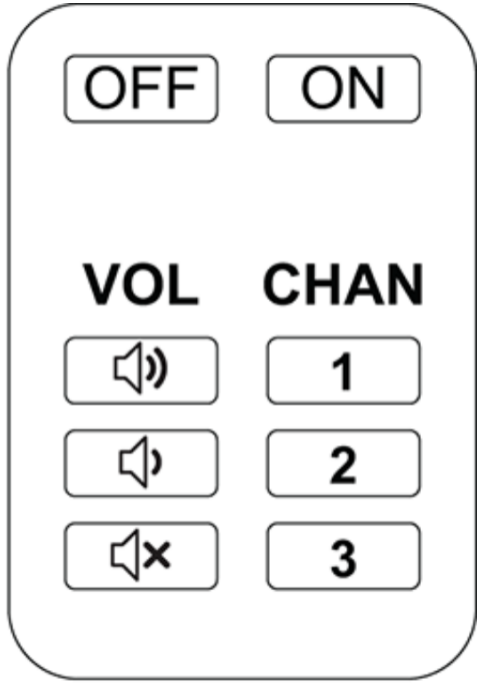
GitOps principles

- a system managed by GitOps must have its desired state expressed declaratively
- desired state is stored in a way that enforces immutability, versioning and retains a complete version history
- software agents automatically pull the desired state declarations from the source
- software agents continuously observe actual system state and attempt to apply the desired state
- OpenGitOps (https://opengitops.dev)

“desired state expressed declaratively”

- declarative!
  - you describe what you want to be achieved
- easy to validate

Imperative remote control



Declarative remote control

white duck

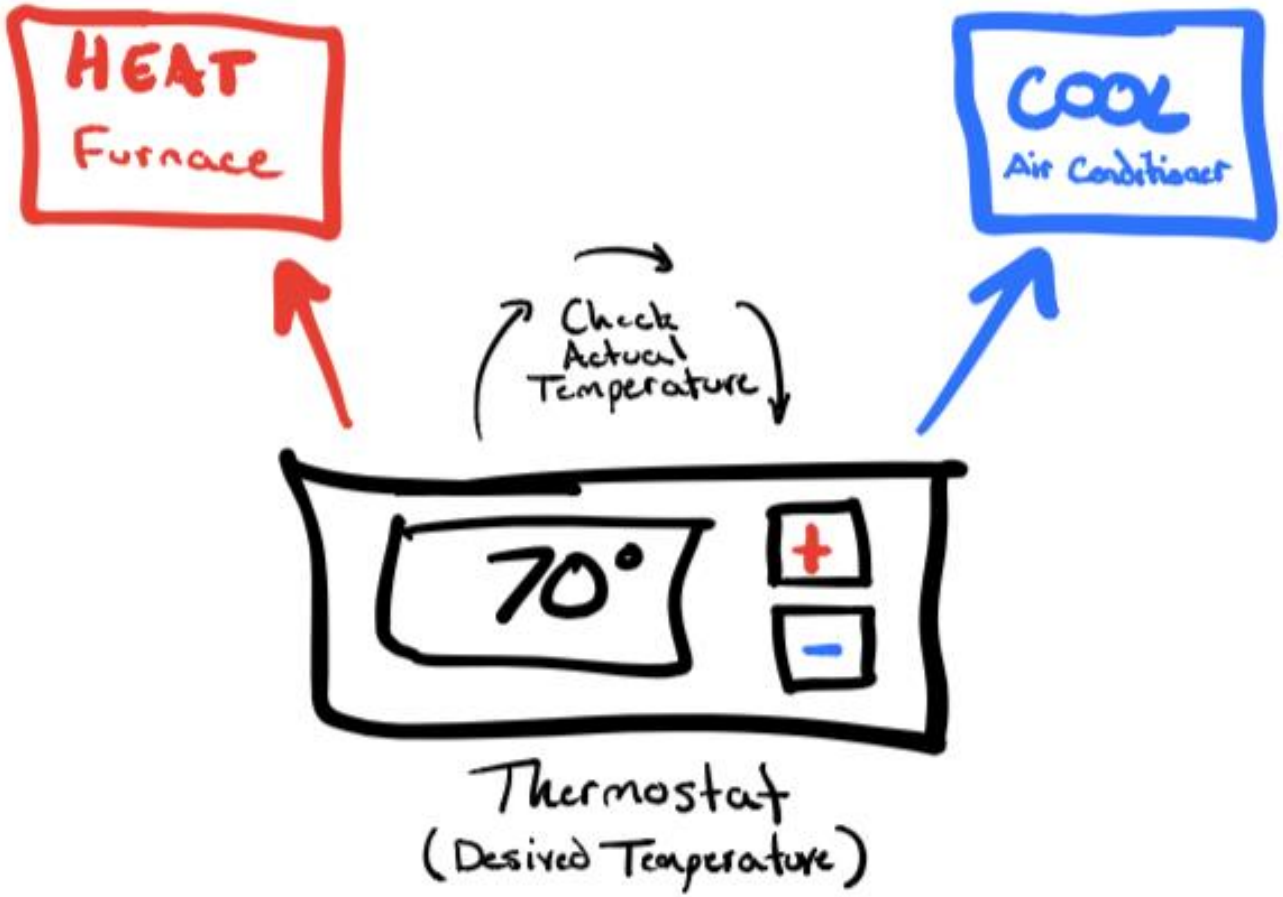
“desired state is stored immutable & versioned”

- Git is the single source of truth
- Git is immutable
- Git is versioned
- Git history allows easy rollbacks

“agents automatically pull the desired state”

- pull instead of push
- changes are automatically applied
- Operator pattern
- privileged operators don’t cross security boundaries
- separates the what and the how

Desired state



“agents continuously observe actual state”

- continuously checking that desired state is met
  - no configuration drift!
- recovers from errors without intervention
- control loop for your operations

GitOps workflow (a sample)

Development

- Developer writes and tests code & merges code changes
- Pipeline builds the image and updates the deployment manifest (via PR)

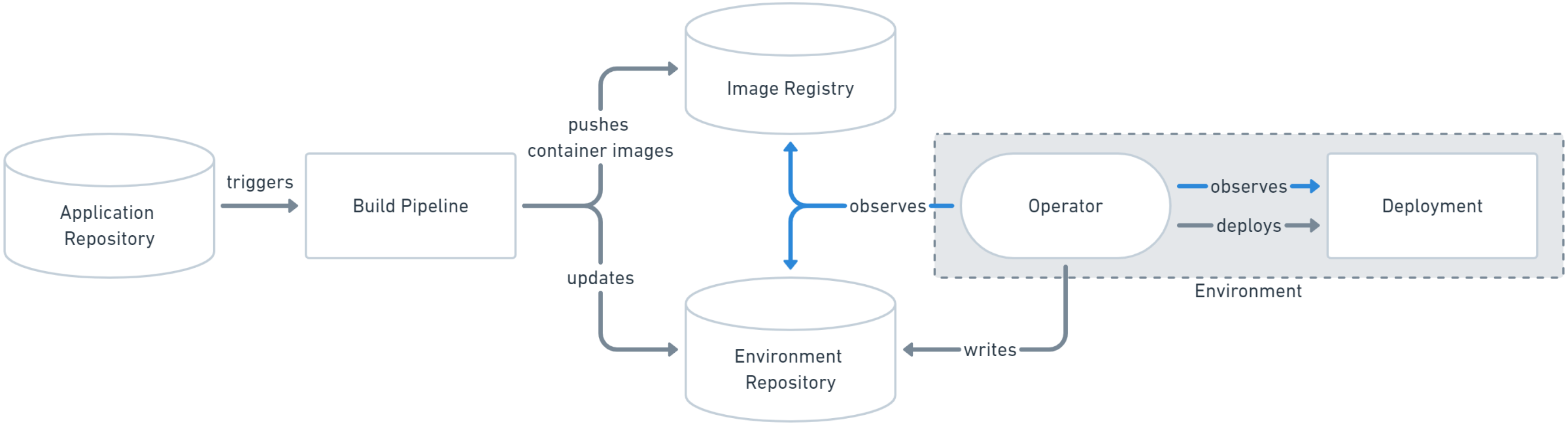
Staging

- After testing the developer merges the deployment manifest PR into the staging branch
- The GitOps agent reconciles the state in the staging cluster

Production

- After validation the developer merges the changes (via PR) into the production branch
- The GitOps agent reconciles the state in the production cluster

Big picture



GitOps benefits

- increased productivity
- enhanced developer experience
- improved stability
- higher reliability
- consistency and standardization
- stronger security guarantees

GitOps: the bad and the ugly

- can be only used with Kubernetes (for now)
- prevent concurrent push on the same repository to not cause conflicts
- git management can become overwhelming
- doesn’t solve centralized secret management
- validation pipelines are a must

white duck

GitOps on Azure

FLUX

Flux

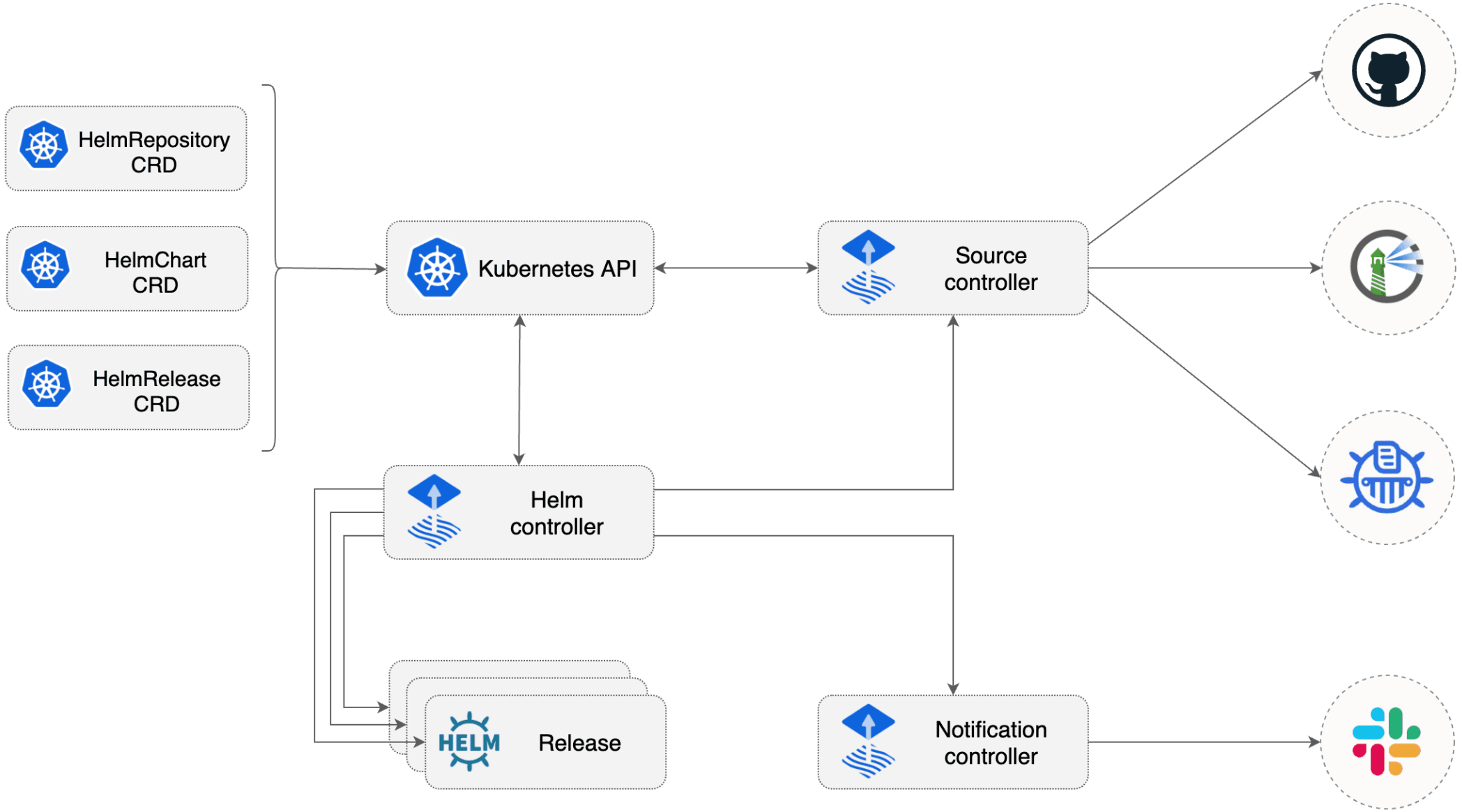
- “is a collection of tools for keeping Kubernetes clusters in sync with sources of configuration”
- contains of
  - Operators keeping Kubernetes clusters in sync
  - CRDs to store configuration
  - CLI
- focuses on the basics
- easy to deploy
- There is Flux v1 and v2!

Flux in detail

- supports multiple sources like Git and Helm Repositories and S3 Buckets
- allows to patch resources with Kustomize
- supports Helm Releases
- can do health assessment
- understands dependency mapping
- can automatically act on new container images
- allows feedback loops (notifications, integrations)

white duck

Flux lifecycle (with Helm and Notification)



white duck

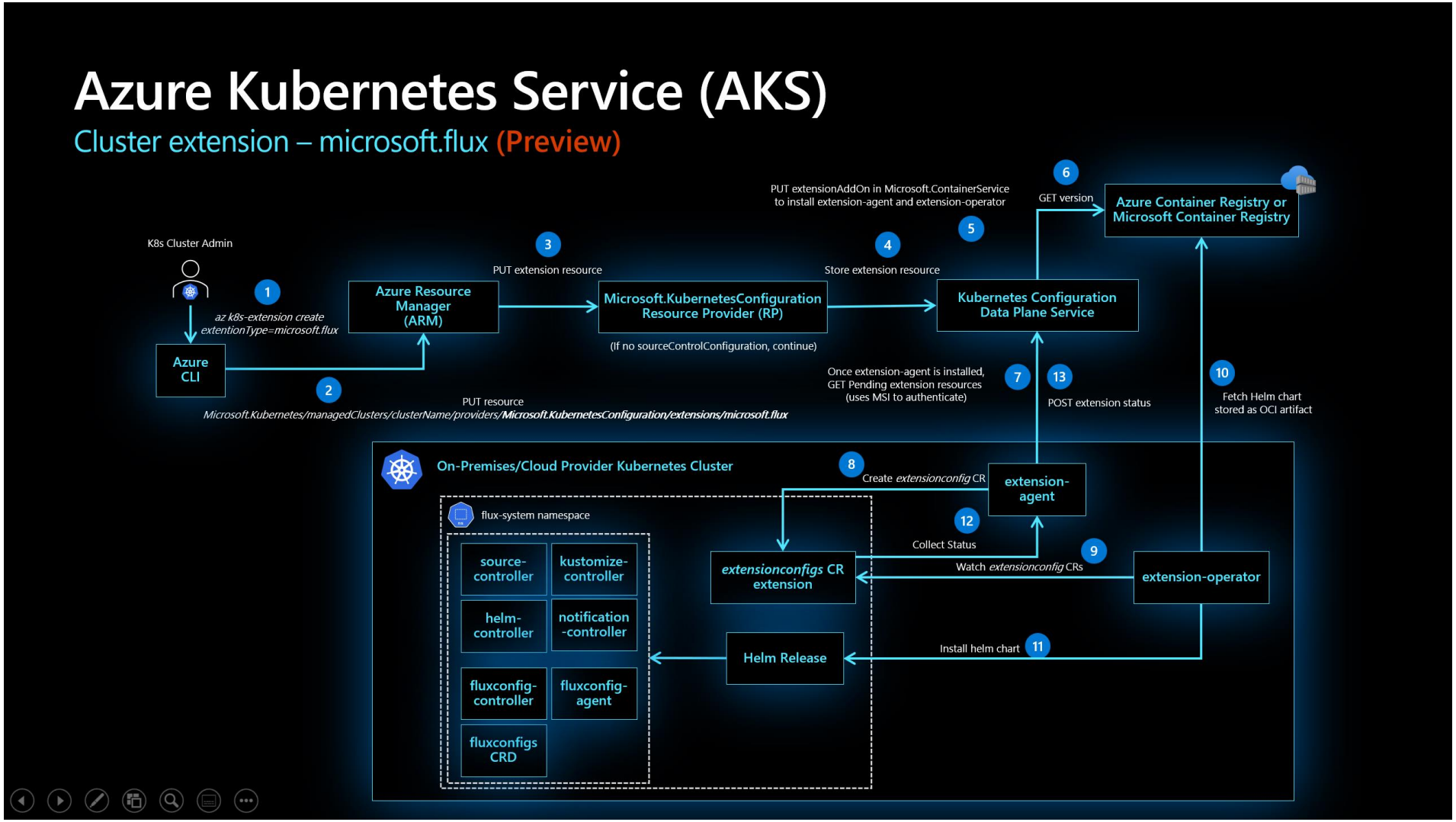
GitOps on Azure

FLUX ON AZURE

Flux on Azure

- Flux v1 is available as AKS addon
- Flux v2 was introduced in Azure Arc-enabled Kubernetes
- since Decembre 2021 public preview in AKS
- enabled as a cluster extension
- cluster extension vs. addon
- can be only used with the Azure Resource Manager
  - Terraform does not support cluster extensions as of now

Flux on Azure



GitOps on Azure

DEMO

Demo

- install & configure Flux extension on an existing AKS cluster
- App deployment with GitOps pattern

FAQ / NETWORKING

Questions?

Martin Brandl (white duck GmbH, Cloud Solution Architect & Azure MVP)
Twitter: @martin_jib
LinkedIn: linkedin.com/in/mbrandl/

Nico Meisenzahl (Senior Cloud & DevOps Consultant & Cloud & Data Management MVP)
Twitter: @nmeisenzahl
LinkedIn: https://www.linkedin.com/in/nicomeisenzahl/

Philip Welz (Senior Kubernetes & DevOps Engineer & GitLab Hero & CKA, CKAD & CKS)
Twitter: @philip_welz
LinkedIn: https://www.linkedin.com/in/philip-welz

twitter.com/whiteduck_gmbh

twitter.com/AzureMeetup

twitter.com/CloudNative_Ro