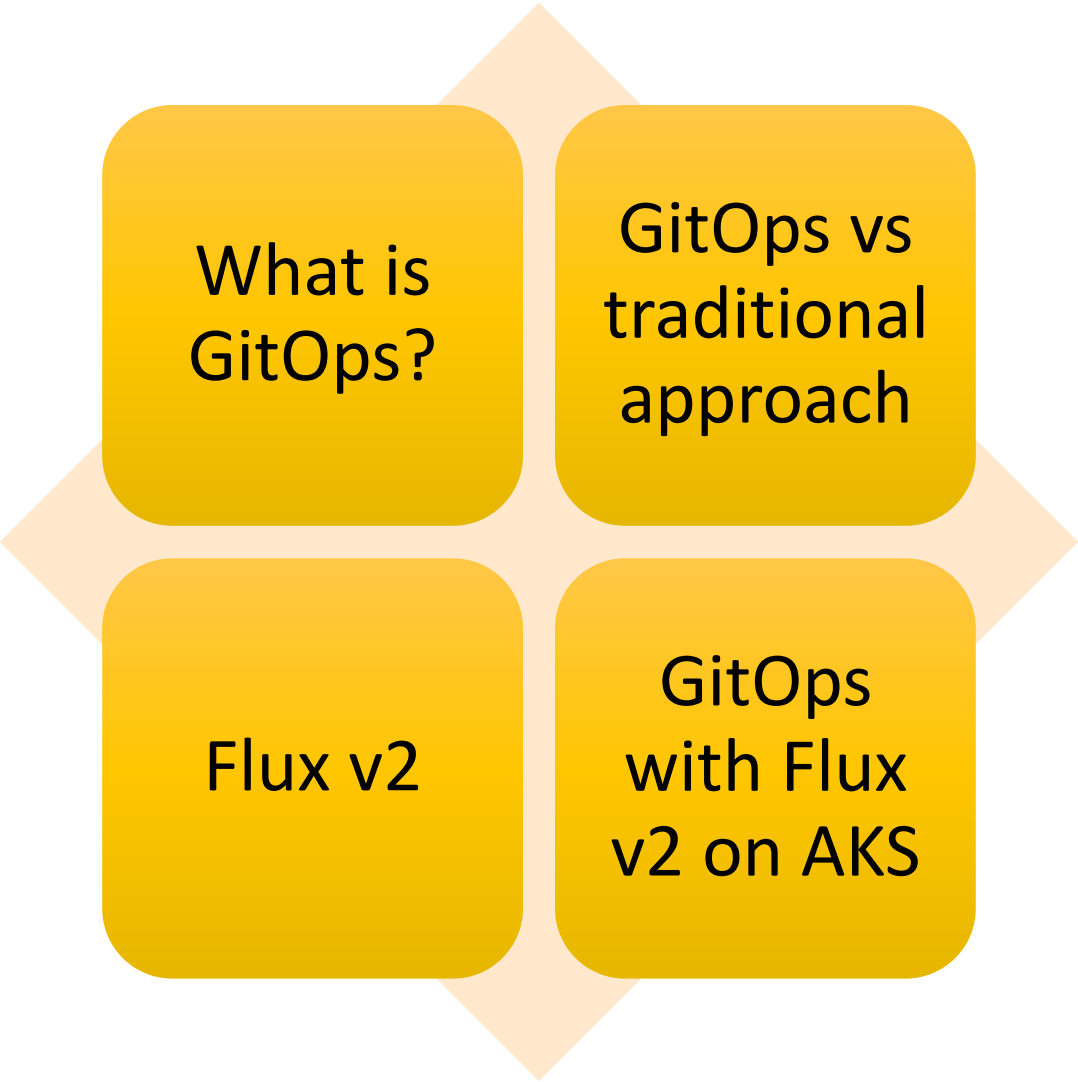
Supercharge your AKS Deployments

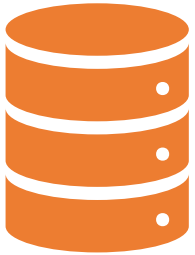
with GitOps and Flux V2

Who Am I?

- Geert Baeke
- Twitter: @geertbaeke
- Blog: https://blog.baeke.info
- GitHub: https://github.com/gbaeke
- YouTube: https://youtube.com/geertbaeke

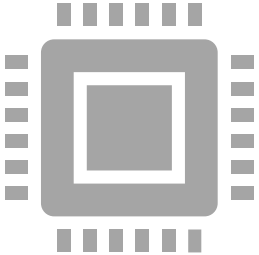

What is GitOps?
GitOps vs traditional approach
Flux v2
GitOps with Flux v2 on AKS

Kubernetes & Git



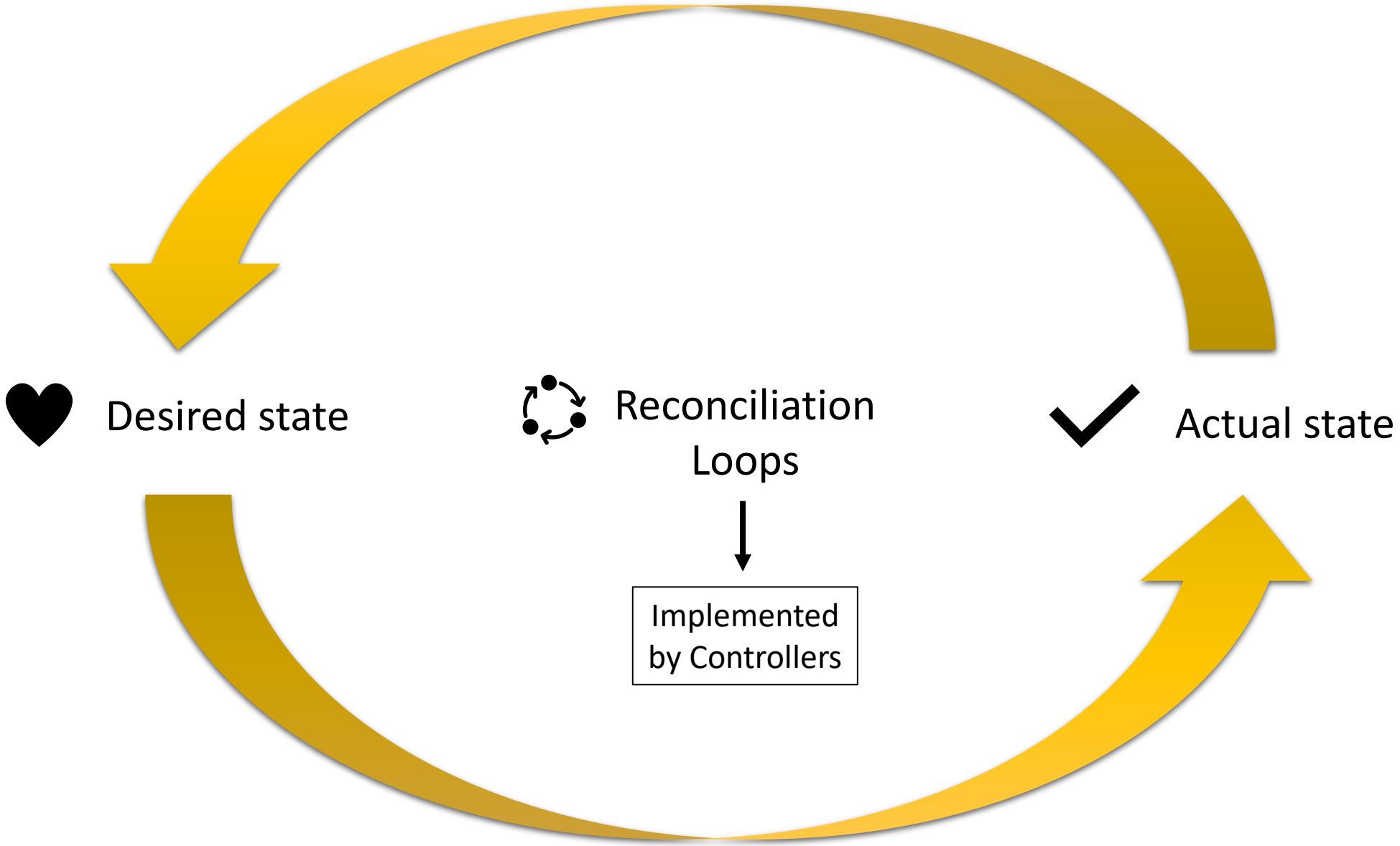
Kubernetes

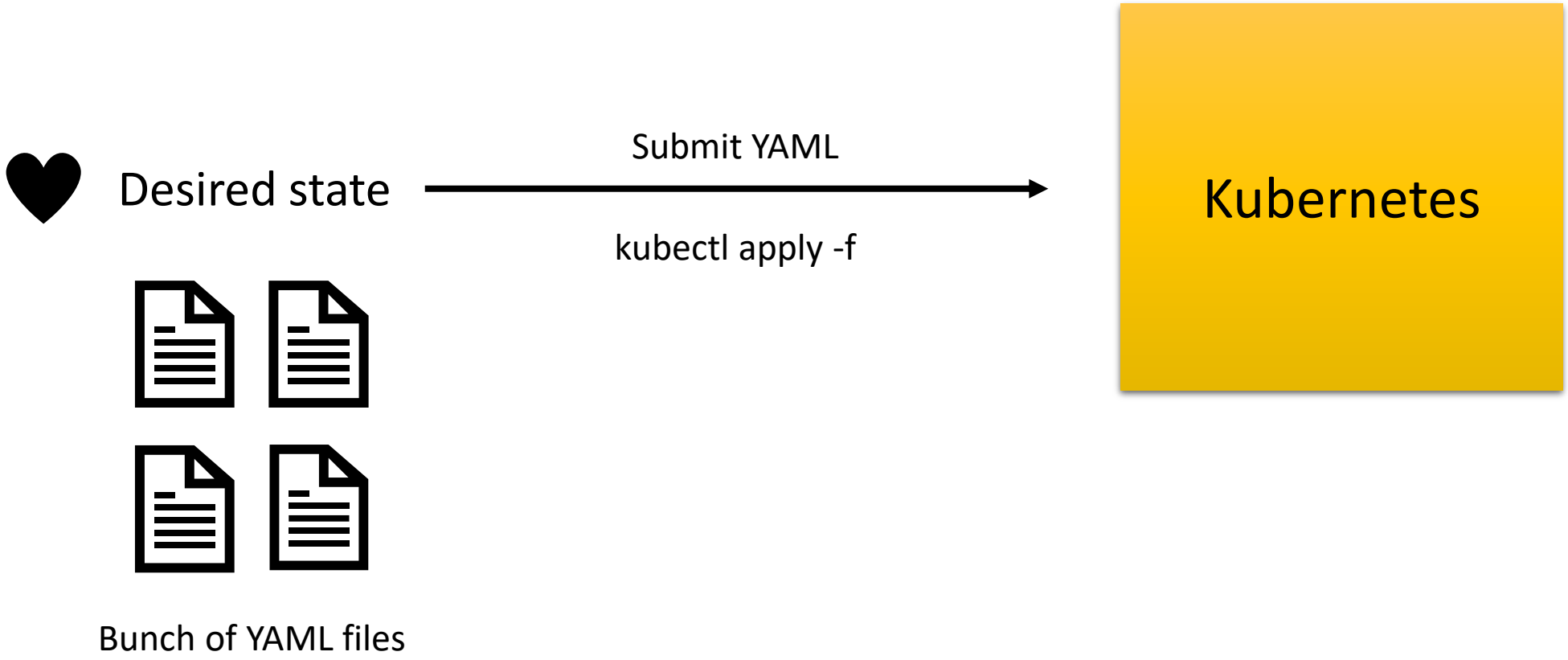
Open-source platform to orchestrate container operations

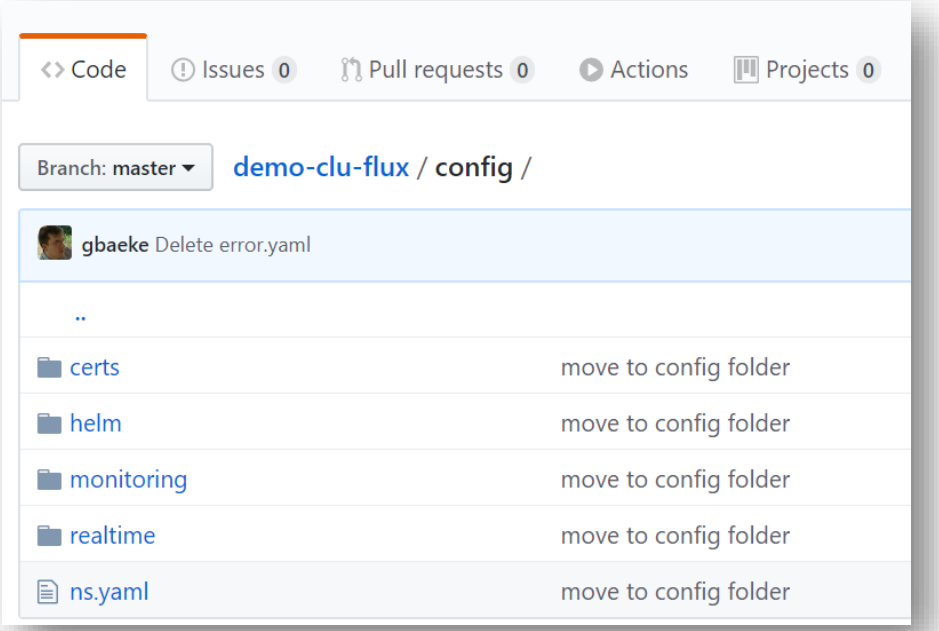


Git

Version control system originally developed by Linus Torvalds


Desired state
Reconciliation Loops
Implemented by Controllers
Actual state


Desired state
Submit YAML
kubectl apply -f
Kubernetes
Bunch of YAML files



Desired state in git

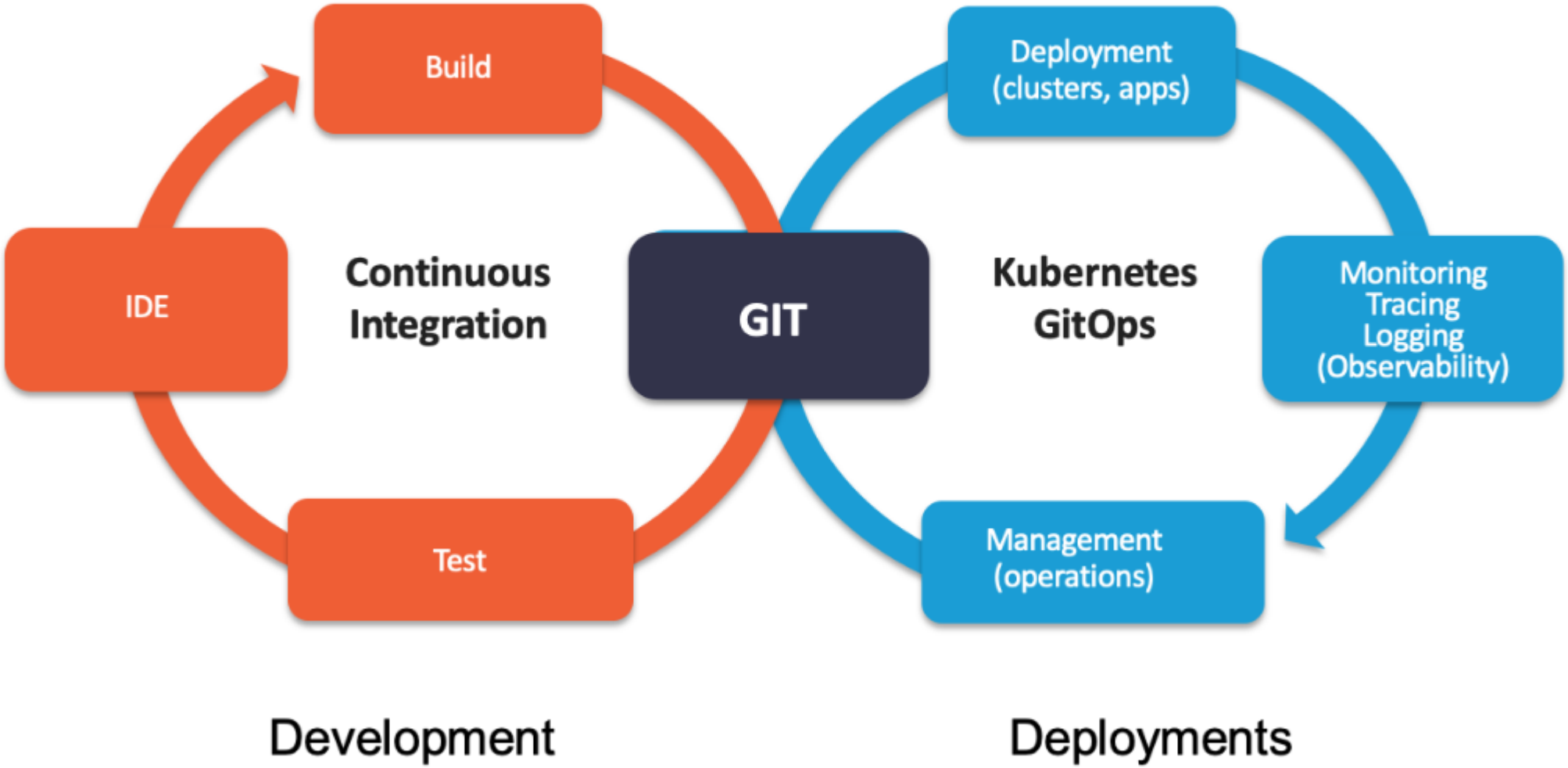
Revisions
Audit trail
Change control

Pull desired state
and apply (CD)

GitOps
Agent

Kubernetes

No need to provide cluster credentials
to external systems like GitHub or Azure DevOps



Source: https://www.weave.works/blog/gitops-is-cloud-native

Traditional approach

Pipeline driven by a CI/CD system (such as Azure DevOps)

Steps that execute tasks

CI/CD systems need credentials to your clusters

Push-based

Flux v2

Open-source GitOps solution

Microsoft

D2IQ

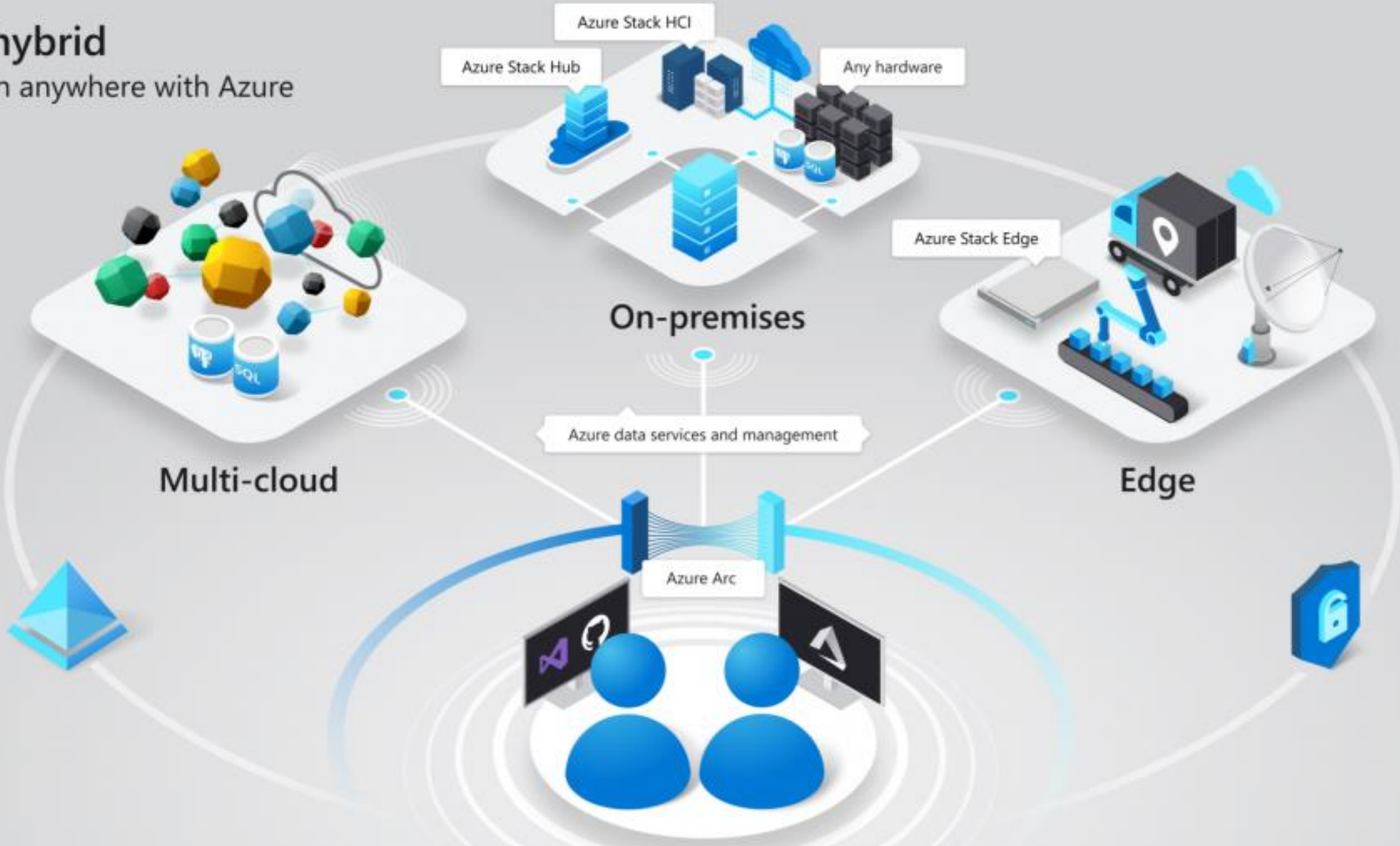
OPENSHIFT

vmware®

amazon

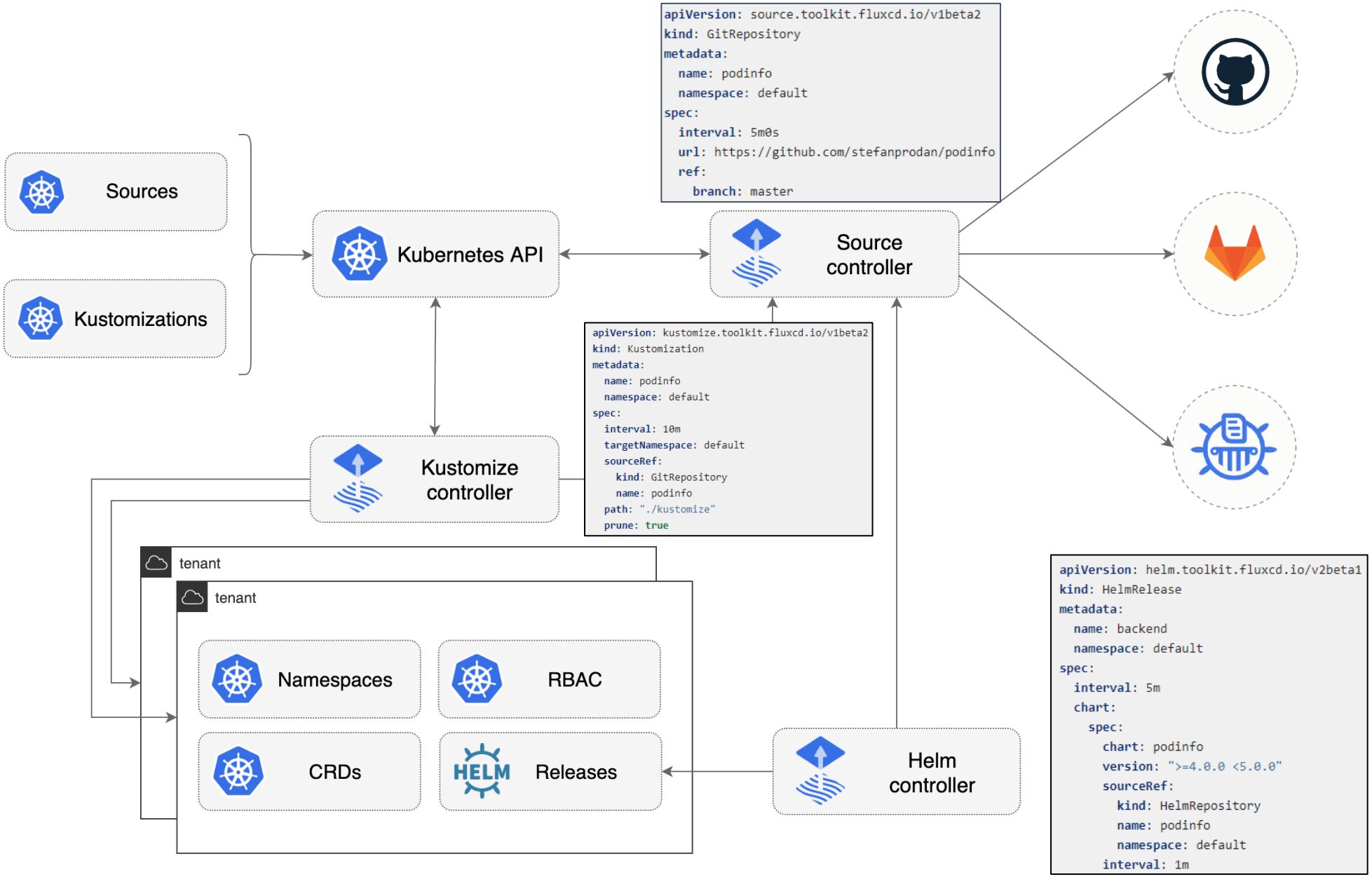
Azure hybrid

Innovation anywhere with Azure



Microsoft Azure

azure.com/hybrid



Source: [Flux Documentation | Flux (fluxcd.io)]

- debug-pod.yaml
- deployment.yaml
- hpa.yaml
- kustomization.yaml
- loadgen.yaml
- namespace.yaml
- service.yaml
- virtual-node-deployment.yaml

```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
namespace: loadgen

resources:
  - namespace.yaml
  - deployment.yaml
  - service.yaml
  - hpa.yaml
  - debug-pod.yaml
```

kustomization.yaml

Run "**kubectl kustomize .**" in folder containing these files

original service.yaml without namespace

```yaml
apiVersion: v1
kind: Service
metadata:
  name: superapi
```

```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: loadgen
---
apiVersion: v1
kind: Service
metadata:
  name: go-template-load
  namespace: loadgen
spec:
  ports:
  - name: http
    port: 80
    targetPort: 8080
  selector:
    app: go-template-load
  type: ClusterIP
```

result is a composition of resources in one YAML manifest

/base
deployment.yaml
service.yaml
kustomization.yaml

```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

commonLabels:
  app: superapi

resources:
  - deployment.yaml
  - service.yaml
```

/base
/overlays
/dev
kustomization.yaml
namespace.yaml
/prd
kustomization.yaml
namespace.yaml

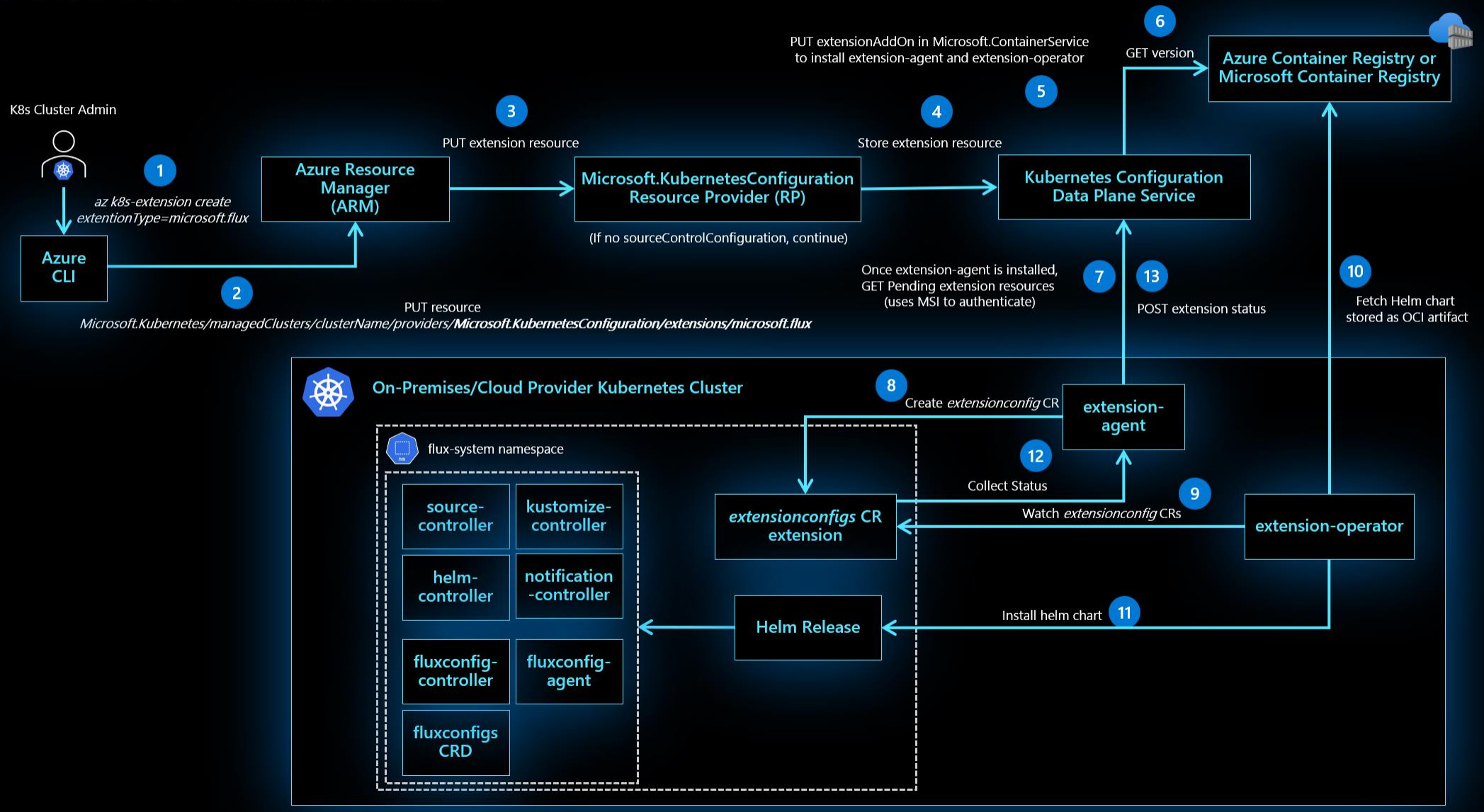
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization

namespace: superapi-dev

commonLabels:
  environment: dev

namePrefix: dev-

resources:
- ../../base
- namespace.yaml

replicas:
- count: 2
  name: superapi
```

GitOps on AKS

with the microsoft.flux extension

Azure Kubernetes Service (AKS)

Cluster extension – microsoft.flux



Steps

Extensions	Enable extension manager on subscription
Providers	Register required service providers
Azure CLI	Enable Azure CLI extensions
Deploy	Deploy one or more Flux configurations (Microsoft.flux extension auto installs)

[Tutorial: Use GitOps with Flux v2 in Azure Arc-enabled Kubernetes or Azure Kubernetes Service (AKS)]

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

main quick-guides / fluxv2 / README.md

gbaeke typo

1 contributor

302 lines (239 sloc) | 11.5 KB

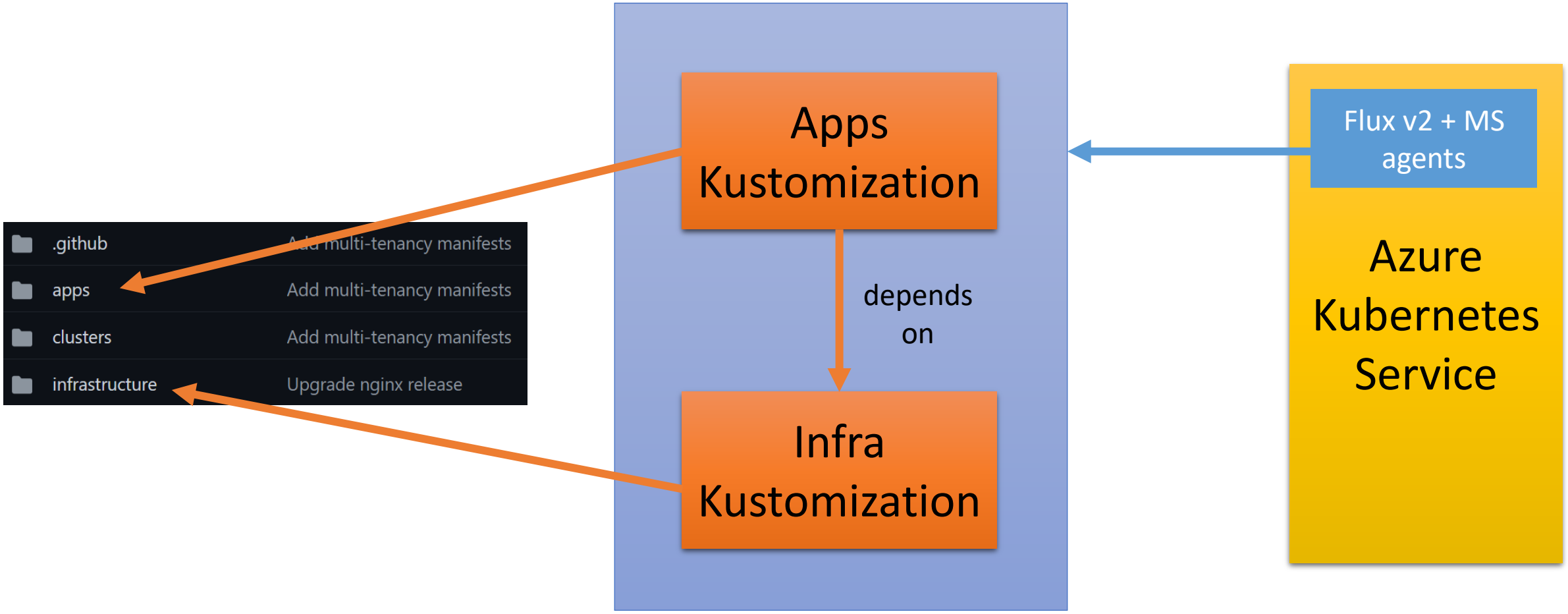
Flux v2 on AKS

Requirements

You need the following to run the commands:

- An Azure subscription with a deployed AKS cluster; a single node will do
- Azure CLI and logged in to the subscription with owner access
- All commands run in bash, in my case in WSL 2.0 on Windows 11
- kubectl and a working kube config (use az aks get-credentials)

https://bit.ly/3yLXow1


.github
Add multi-tenancy manifests
apps
Add multi-tenancy manifests
clusters
Add multi-tenancy manifests
infrastructure
Upgrade nginx release
Apps Kustomization
depends on
Infra Kustomization
Flux v2 + MS agents
Azure Kubernetes Service

Flux Config

gbaeke / gitops-flux2-quick-guide Public

forked from Azure/gitops-flux2-kustomize-helm-mt

Code | Pull requests | Actions | Projects | Wiki | Se

main | 2 branches | 0 tags

This branch is 9 commits behind Azure:main.

csand-msft Merge pull request Azure#2 from Azure/join... ... c4e3272 O

.github	Add multi-tenancy manifests
apps	Add multi-tenancy manifests
clusters	Add multi-tenancy manifests
infrastructure	Upgrade nginx release

https://bit. y/3RItPUP |

```
az k8s-configuration flux create -g $RG -c $CLUSTER \
  -n cluster-config --namespace cluster-config -t managedClusters \
  --scope cluster -u https://github.com/gbaeke/gitops-flux2-quick-guide \
  --branch main  \
  --kustomization name=infra path=./infrastructure prune=true \
  --kustomization name=apps path=./apps/staging prune=true dependsOn=["infra"]
```

The above commands results in one source (git repository) and two "kustomizations"
The Source Controller and Kustomization controllers take action when they see these resources in the "cluster-config" namespace.

Home > Kubernetes services > clu-fluxms >

clu-fluxms/cluster-config

GitOps configuration

Search (Ctrl+/)

- Overview
- Configuration objects
- Source
- Kustomizations

Delete Refresh

Status

Compliance state	Non-compliant
Configuration objects	8 objects
Installation status	Succeeded
Source last sync commit	main/c4e327286379495f8b6edc2c9c208095b6cc53ab
Source last updated	7/19/2022, 12:52:41 PM
Status last updated	7/19/2022, 2:25:27 PM

Properties

Namespace	cluster-config
Scope	cluster
Type	Flux v2
Kustomizations	2 Kustomizations

Home > Kubernetes services > clu-fluxms > clu-fluxms/cluster-config

clu-fluxms/cluster-config | Configuration objects

GitOps configuration

Search (Ctrl+/)

- Overview
- Configuration objects
- Source
- Kustomizations

Refresh

Object	Kind	Compliance state
cluster-config	GitRepository	Compliant
cluster-config-apps	Kustomization	Non-compliant
cluster-config-infra	Kustomization	Pending
bitnami	HelmRepository	Compliant
podinfo	HelmRepository	Compliant
nginx	HelmRelease	Compliant
redis	HelmRelease	Non-compliant
test-chart	HelmChart	Non-compliant

Repository structure

- Monorepo
- Repo per environment
- Repo per team
- Repo per app

Deploying new container images

- Flux can scan container registries and retrieve image tags
- Select tag based on policy
- Replace tag in manifests
- Checkout, commit and push changes to git repository

In summary

- GitOps for deployments of infrastructure and apps
- Keep on using CI as you see fit
- Fully declarative based on manifests in git
- Pull vs push
- Fully integrated with AKS