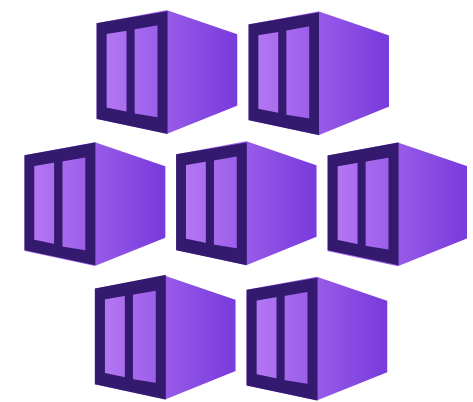


To Push or to Pull: Managing AKS clusters



+



=



Agenda

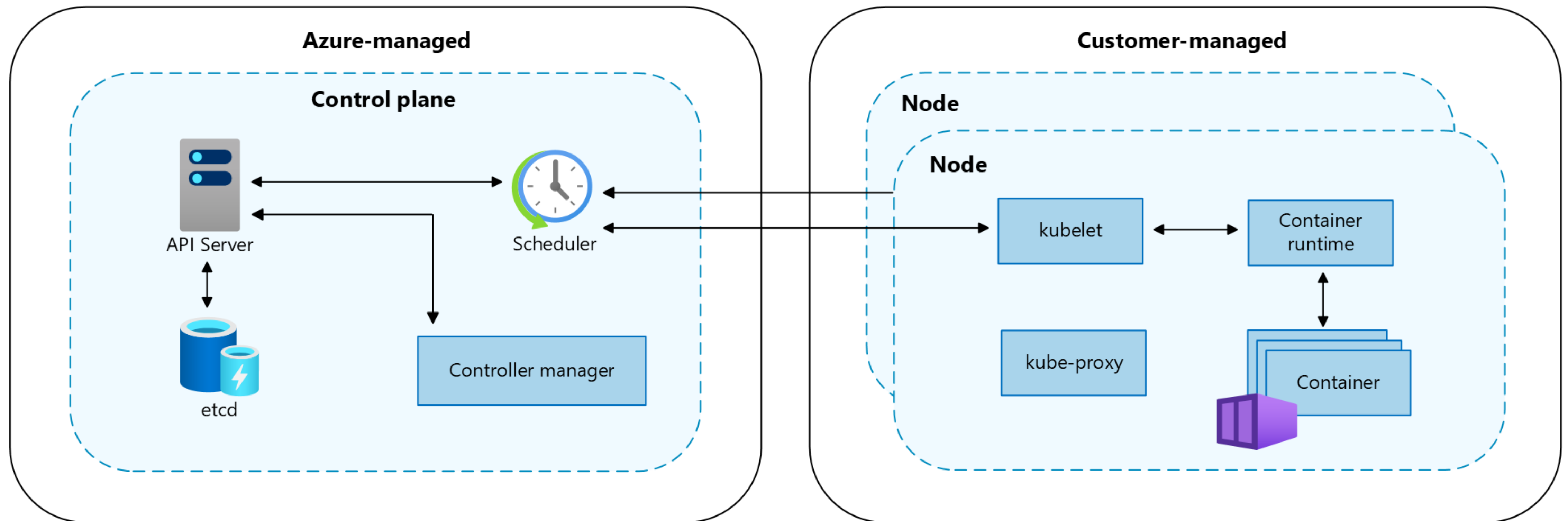
- AKS - Azure Kubernetes Service
- GitOps
- Push-based model
- Pull-based model

AKS - Azure Kubernetes Service

- AKS is managed container orchestration service based on Kubernetes
 - Azure manages the control plane (Kubernetes Api server, etcd , scheduler)
- It simplifies deploying, managing, and operating Kubernetes cluster by offloading the responsibility to Azure
 - Kubernetes version upgrades
 - Node patching and upgrades
 - Node provisioning / scaling
 - Microsoft Entra ID integration
- AKS provides the tools but user is responsible for creating node pools, implementing the application deployment, choose network model, configuring policy and monitoring etc.
- With AKS, Microsoft is Leader in the 2024 Gartner® Magic Quadrant™ for Container Management

AKS - Azure Kubernetes Service

Architecture



<https://learn.microsoft.com/en-us/azure/aks/core-aks-concepts#what-is-aks>

GitOps

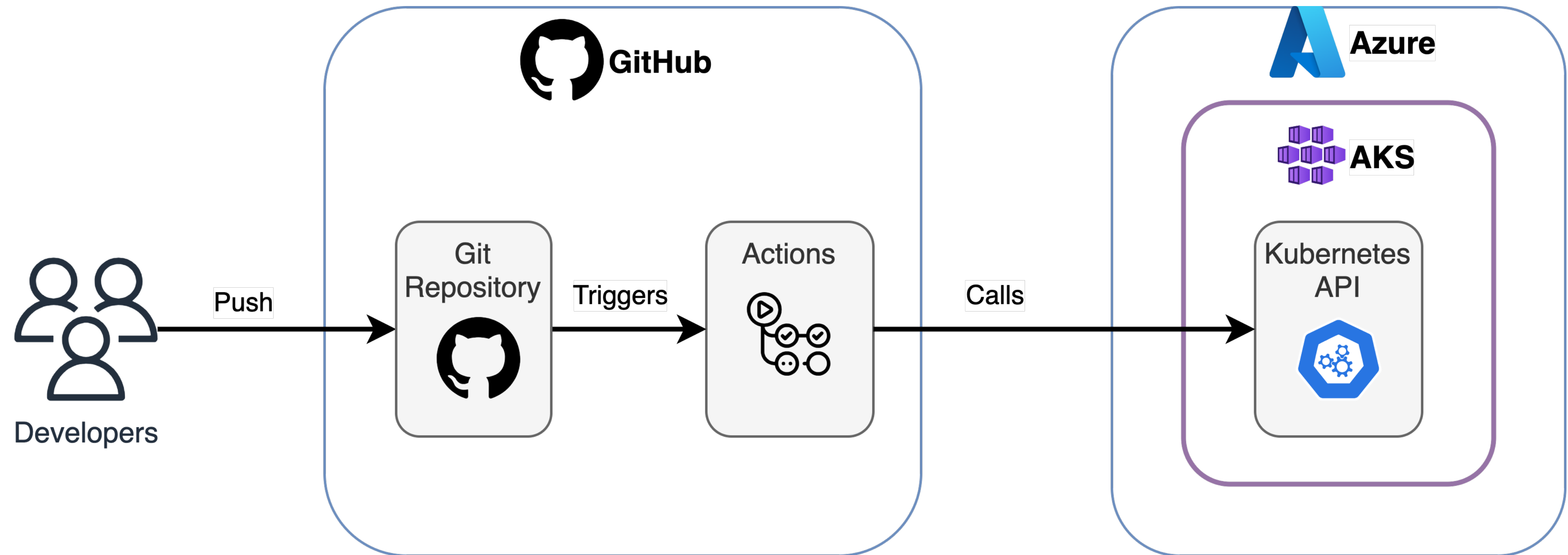
- GitOps is a DevOps practice for managing infrastructure and application configuration using declarative programming
 - Kubernetes deployment YAML files
- Git repositories are used to manage your AKS cluster infrastructure and application deployment
 - “pull request to manage your cluster”
- You can leverage your existing git practices.
 - Code reviews
 - Branching and merge strategies
 - Commit message guidelines
- Together with GitOps and CI/CD, you can automate your Kubernetes deployments.
- GitOps can be implemented using two main approach: push and pull

Push-based approach

- Push-based approach is usually implemented with CI/CD pipelines
 - Azure DevOps
 - Github Actions
- Deploys changes using the **kubectl** command
- It can be a simpler setup because it does not require extra components in the Kubernetes cluster
 - However, you need to plan about how to implement manage the pipelines
 - How to deploy applications across multiple Kubernetes clusters?
- Fast deployment speed because the pipeline can be triggered immediately on code change
- Security considerations
 - Access control
 - CI/CD tool requires privileged access for the Kubernetes cluster
 - Credentials are stored in the CI/CD tool
 - Network
 - Kubernetes API should be accessible to the CI/CD tool (INBOUND connection)

Push-based approach

Architecture

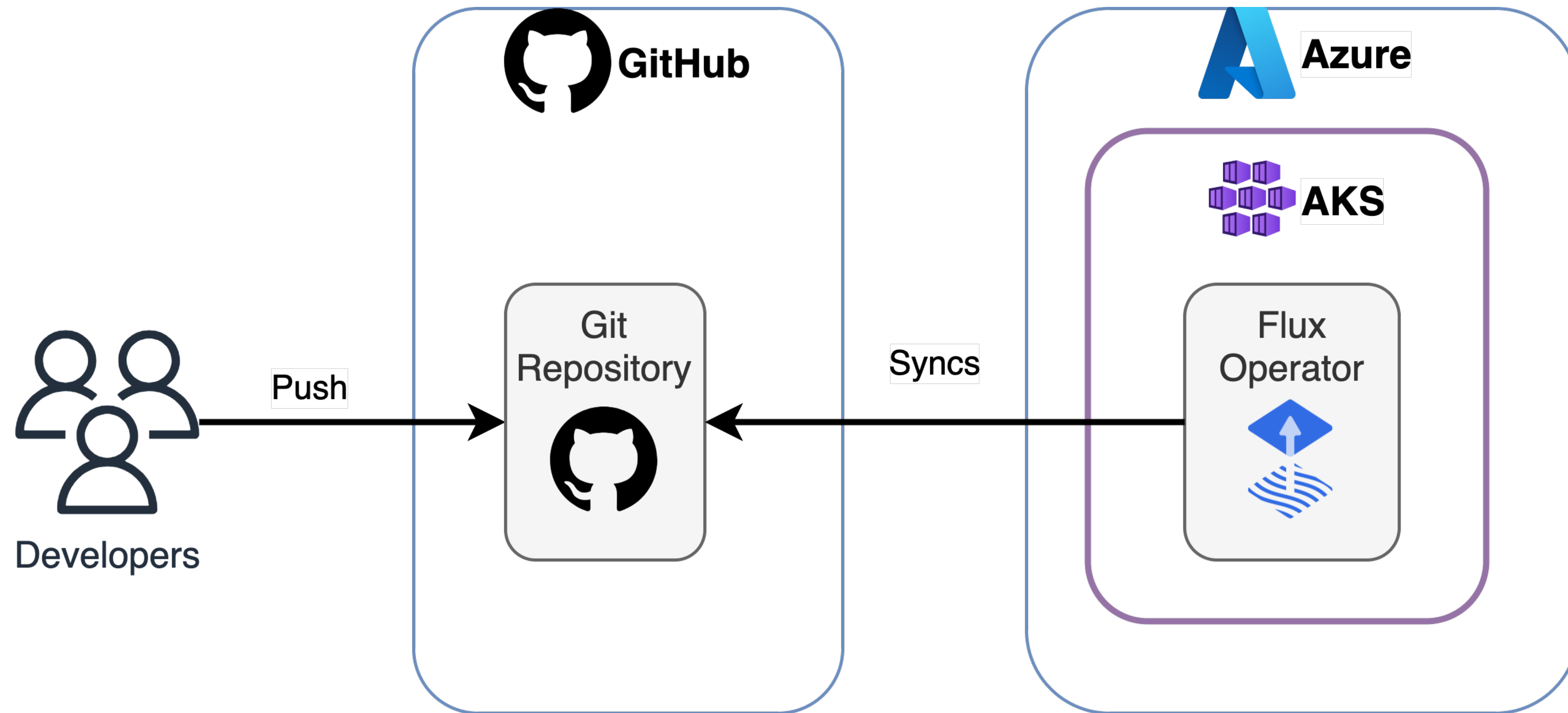


Pull-based model

- A pull-based approach is implemented with a Kubernetes operator running within the Kubernetes cluster
 - Flux CD
 - Argo CD
- More complex setup because it requires extra components in the Kubernetes cluster
 - Fortunately, AKS helps with this because GitOps is ‘built-in’ with the Flux V2 extension
- Flux Operator syncs the with Git repository and deploy Kubernetes resources
- Slower deployment speed because of polling intervals
- Security considerations
 - Access control
 - The Flux Operator requires read access to the Git repository
 - Git credentials are stored as a Kubernetes secret
 - Network
 - Kubernetes clusters require network access to the Git repository (OUTBOUND connection)

Pull-based model

Architecture



Questions?

Thank you!