



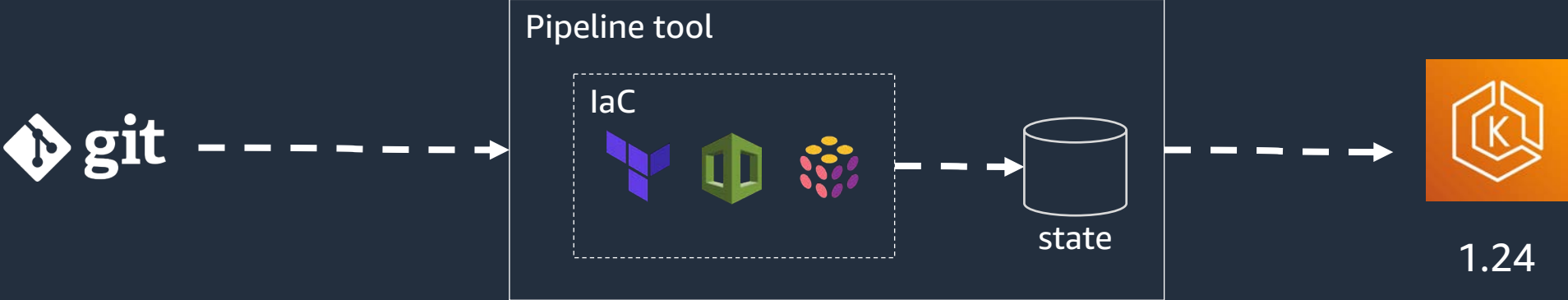
AMAZON EKS MULTI-CLUSTER TOPOLOGIES

# Scaling production grade Kubernetes Multi-Cluster environments using GitOps

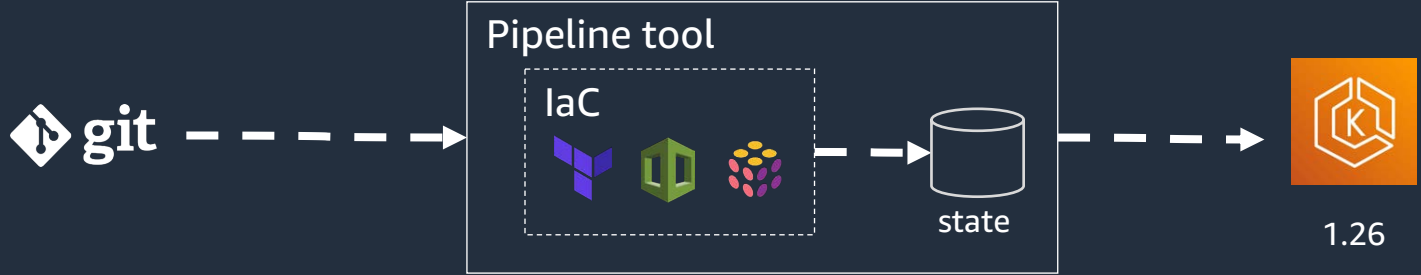
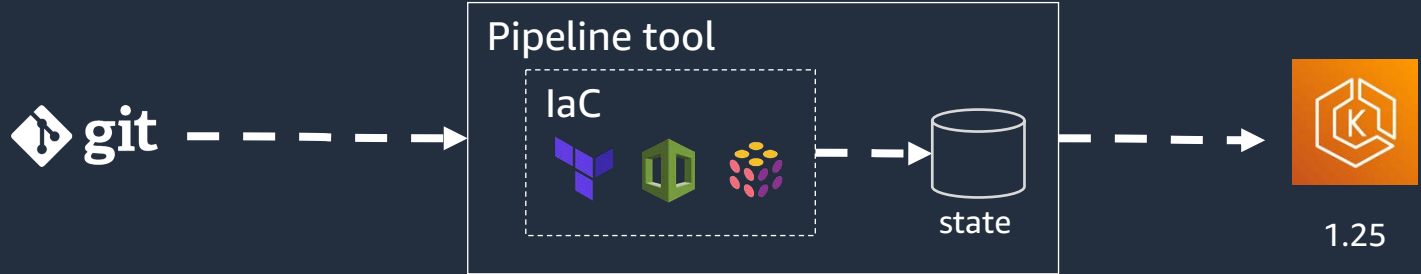
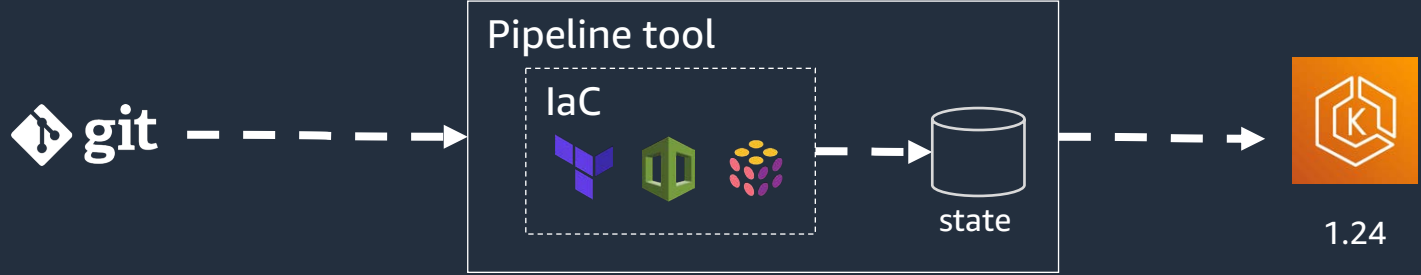
**Yuriy Bezsonov**

Senior Solutions Architect,  
AWS

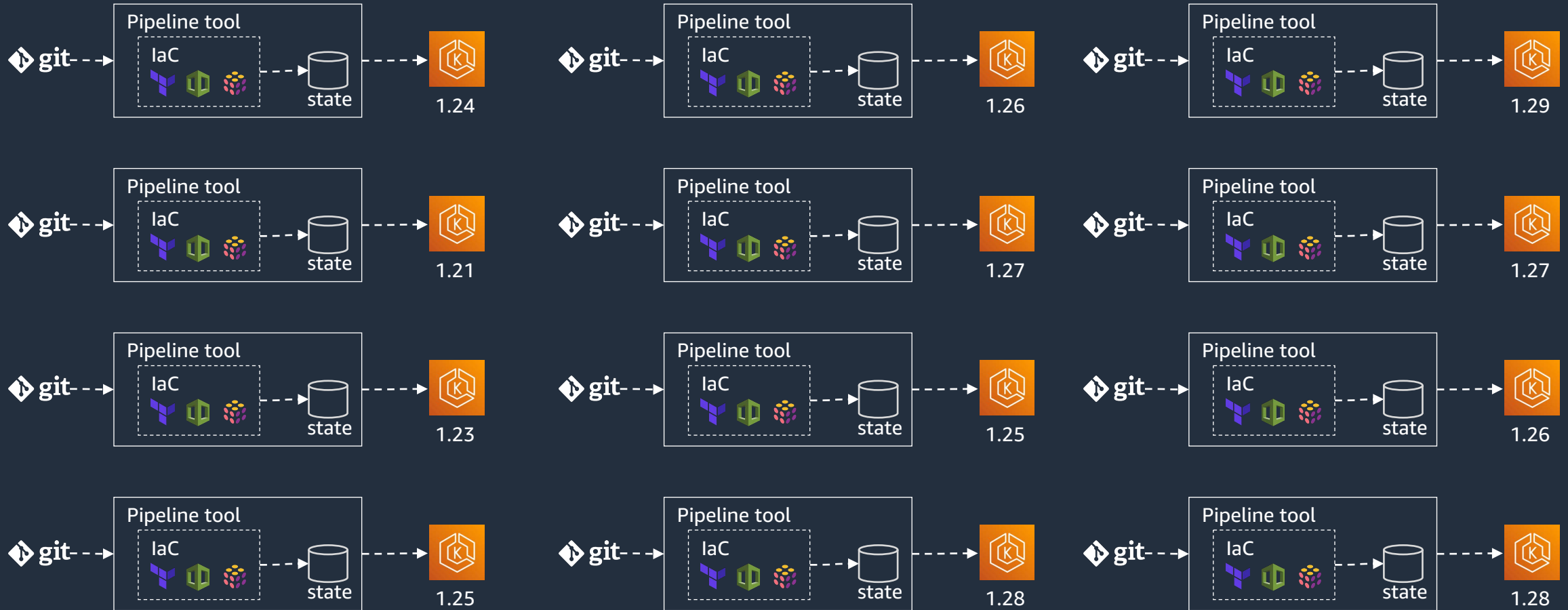
# Simple Beginnings



# Manageable Growth



# Un-Manageable Growth



# Why So Many Clusters!?



## Backends

- Apps and services
- Mobile
- IoT



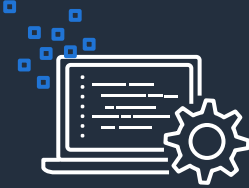
## Web applications

- Static websites
- Complex web apps



## Stateful Workloads

- Databases
- Streaming
- MapReduce
- Batch



## Legacy app modernization

- .NET apps
- Legacy homegrown Linux apps
- Monoliths



## AI/ML

- Autonomous vehicles
- Robotics
- Training & Inference

# Key challenges

Cluster management



Enforce security standards and best practices across clusters to automate deployments

Team management



Define boundaries between multiple teams

Add-on management



Install add-ons and their dependencies

Workload management



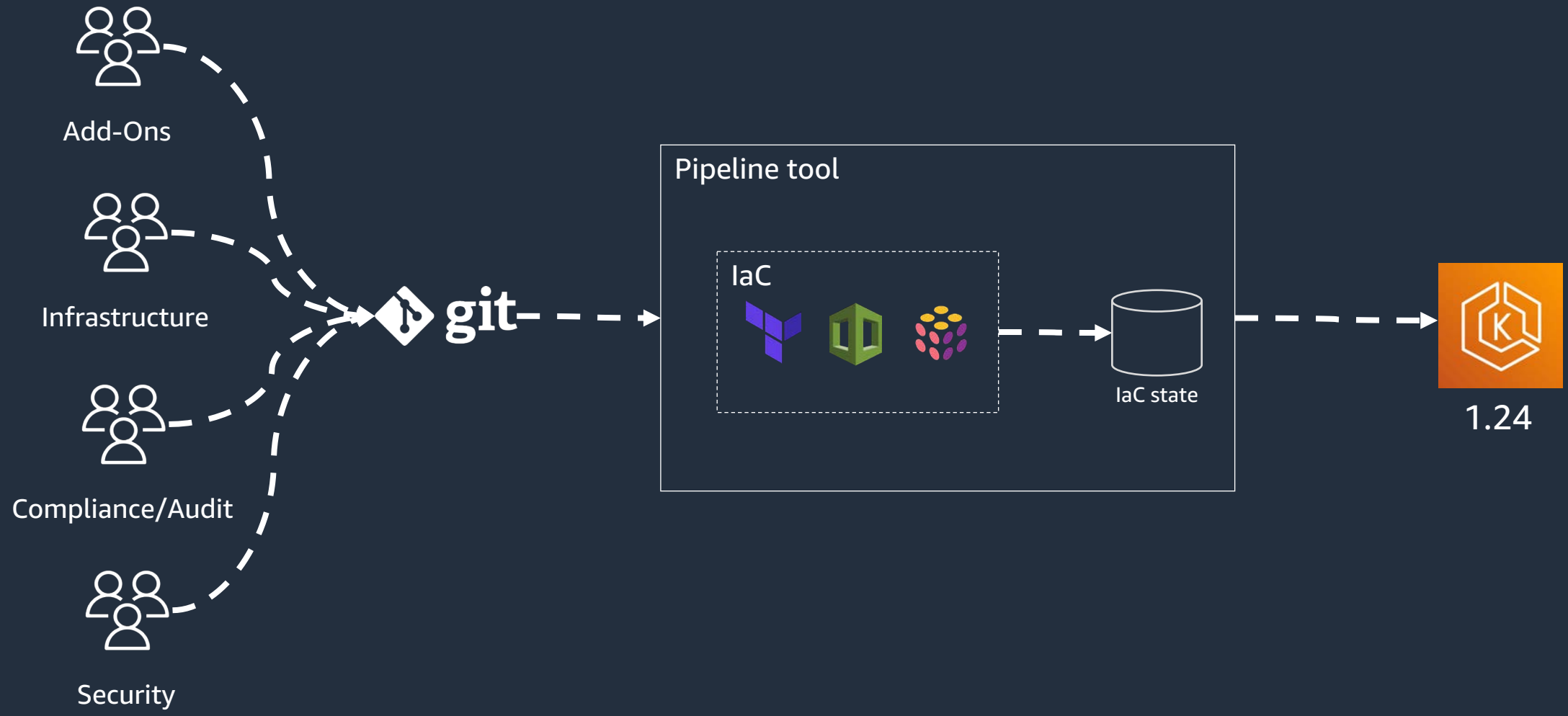
Provision multiple workloads at scale

Configuration management

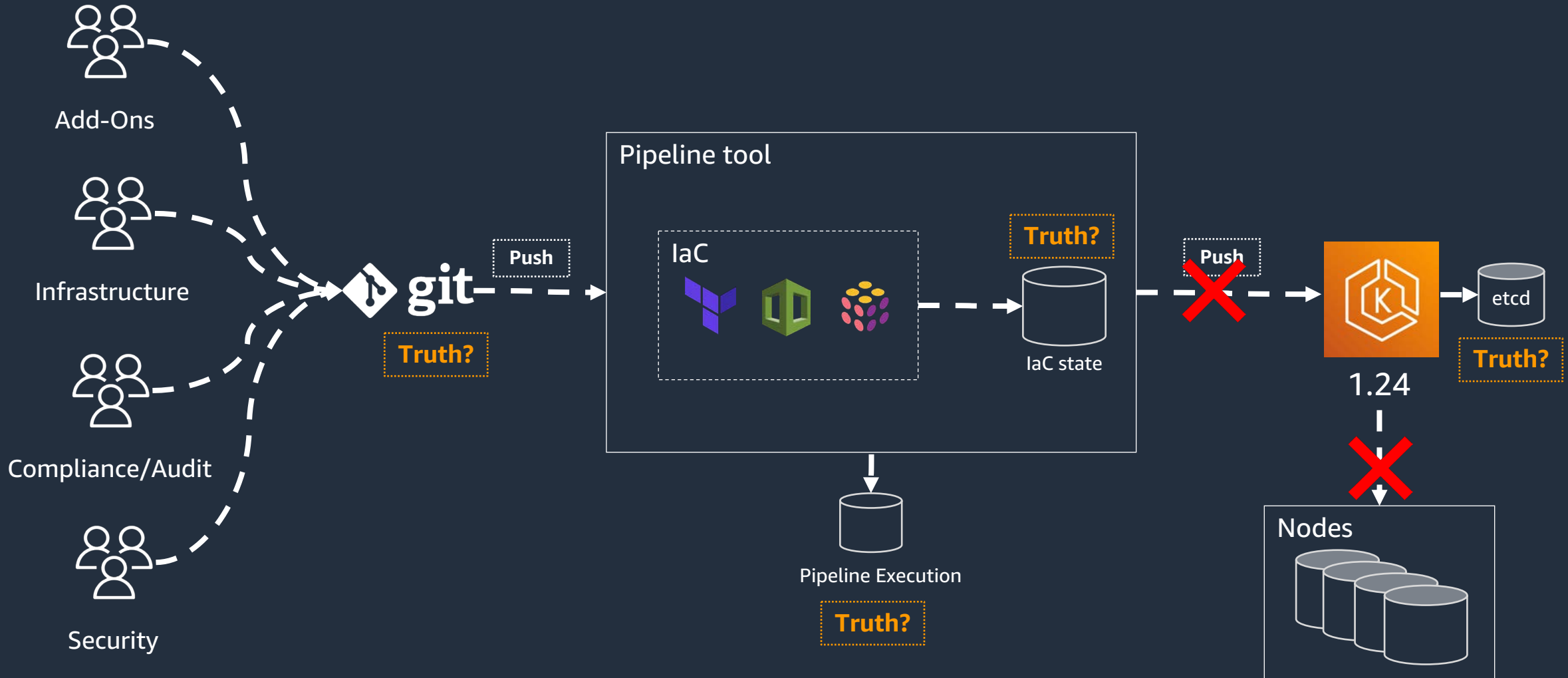


Automate configuration and upgrade lifecycle from a single source of truth

# Fleet Management Challenges

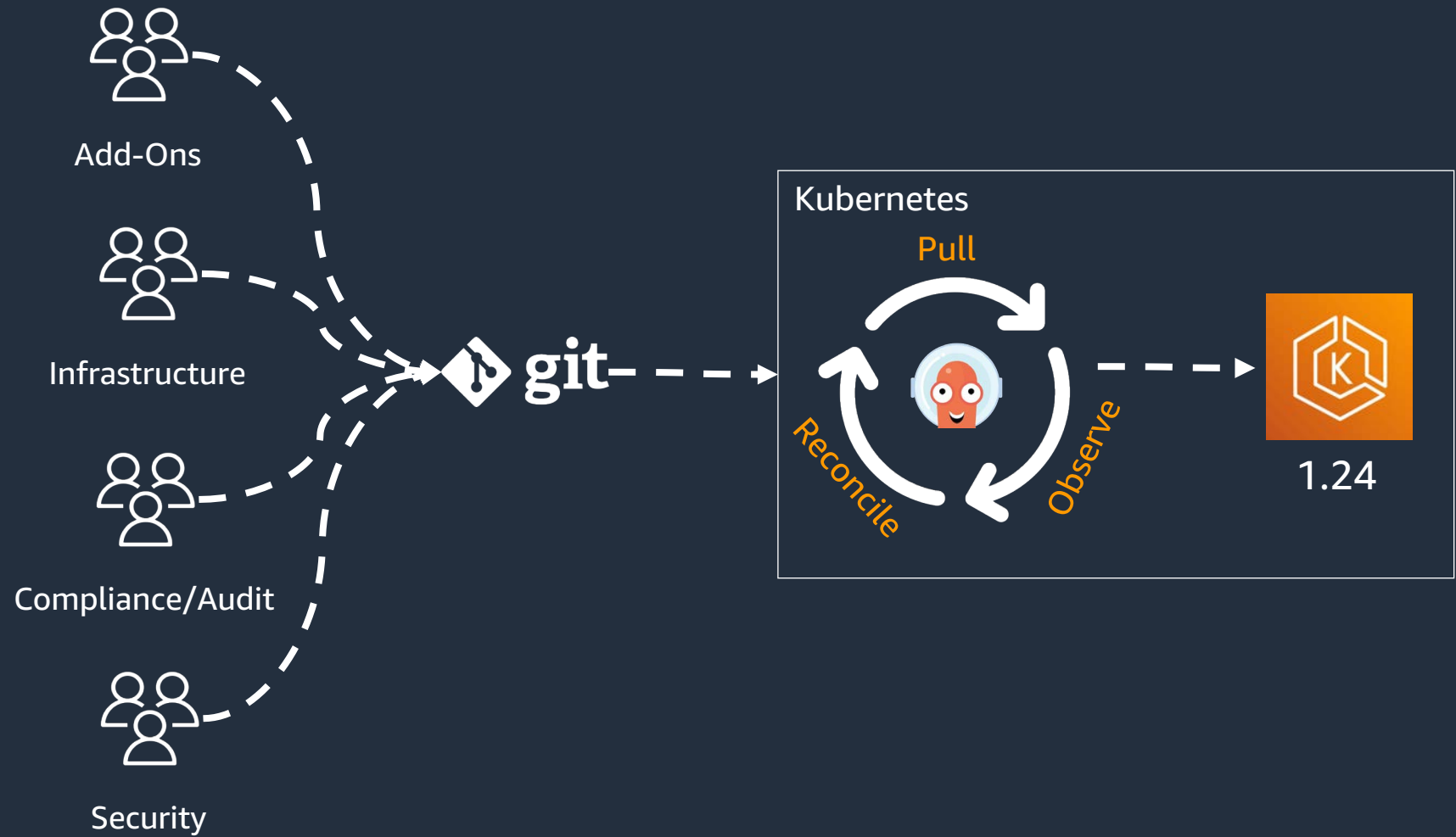


# Fleet Management Challenges



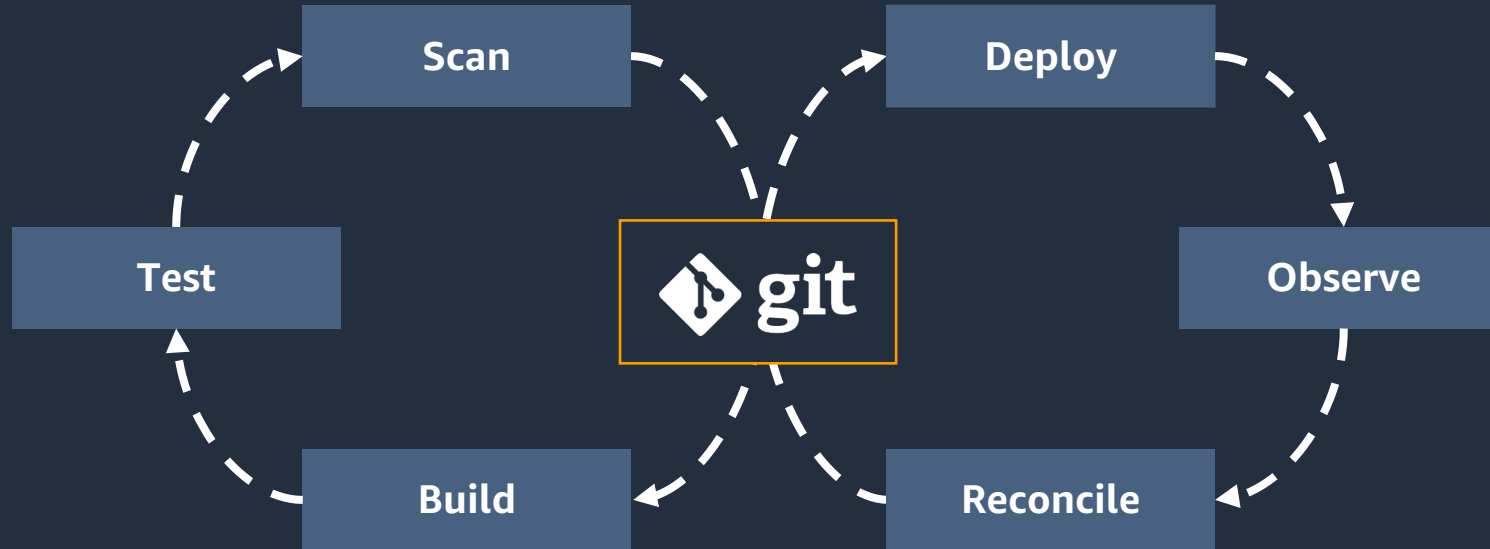


# Fleet Management With GitOps



# GitOps as the solution

GITOPS



Git is the **single source of truth** for the desired state, enabling reproducible automated deployments, cluster management, and monitoring.

# GitOps

## PRINCIPLES



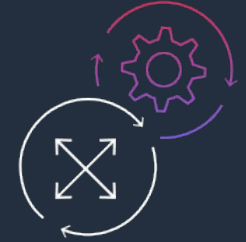
Desired state is expressed  
**declaratively**



Persisted in an **immutable**  
& **versioned** store



Agents automatically **pull**  
desired state



Agents continuously  
**observe** and **reconcile**



Reduces **Complexity**



Enhances **Auditability**



Boosts **Security**



Enforces **Consistency**

# Cluster Deployment



# Amazon EKS Blueprints

An open-source framework that allows you to deploy production-ready EKS clusters



Infrastructure as Code  
with Terraform and CDK



Based on AWS best  
practices and  
recommendations



Integrated with popular  
K8s tools and services



Fully extensible and  
customizable

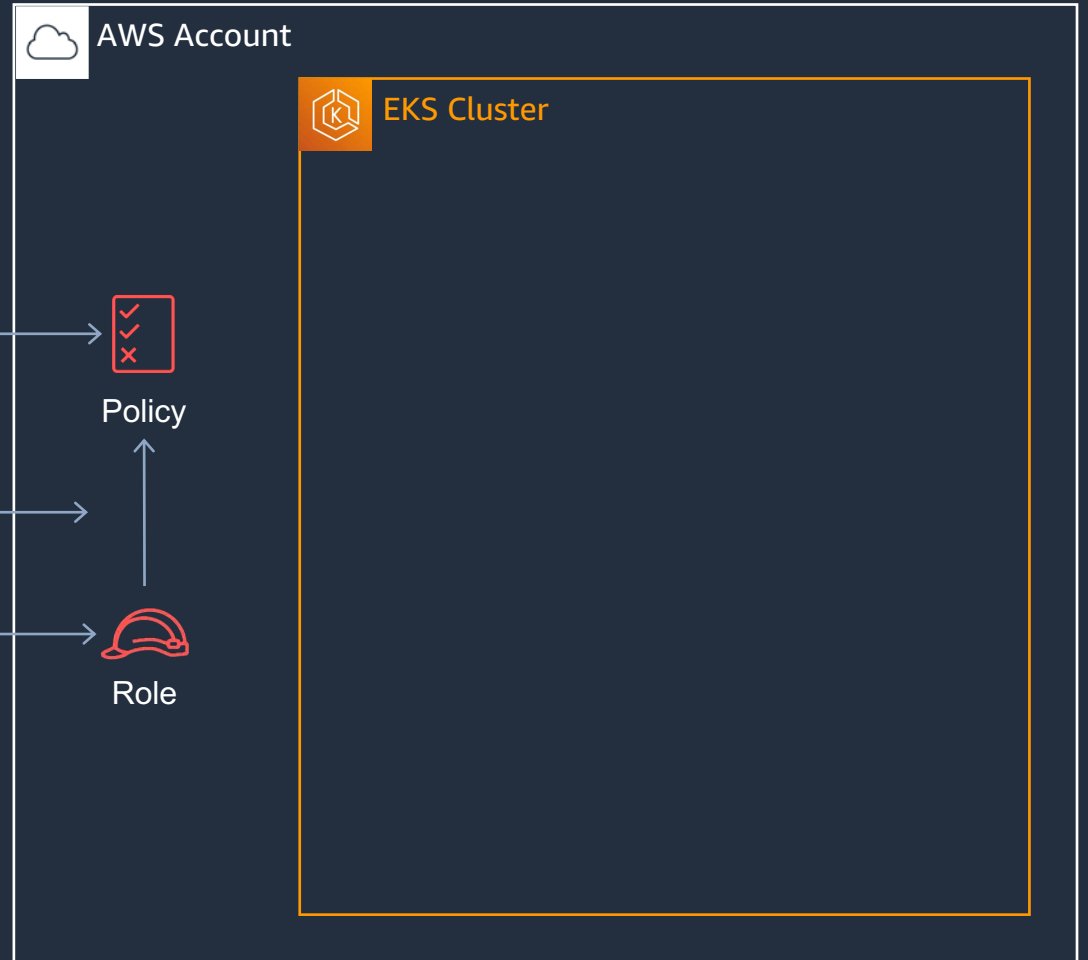
# Addons



## AWS-LOAD-BALANCER-CONTROLLER



```
resource "aws_iam_policy" "albc" {  
  ...  
}  
  
resource "aws_iam_role_policy_attachment" "albc" {  
  role      = aws_iam_role.albc.name  
  policy_arn = aws_iam_policy.albc.arn  
}  
  
resource "aws_iam_role" "albc" {  
  ...  
}
```

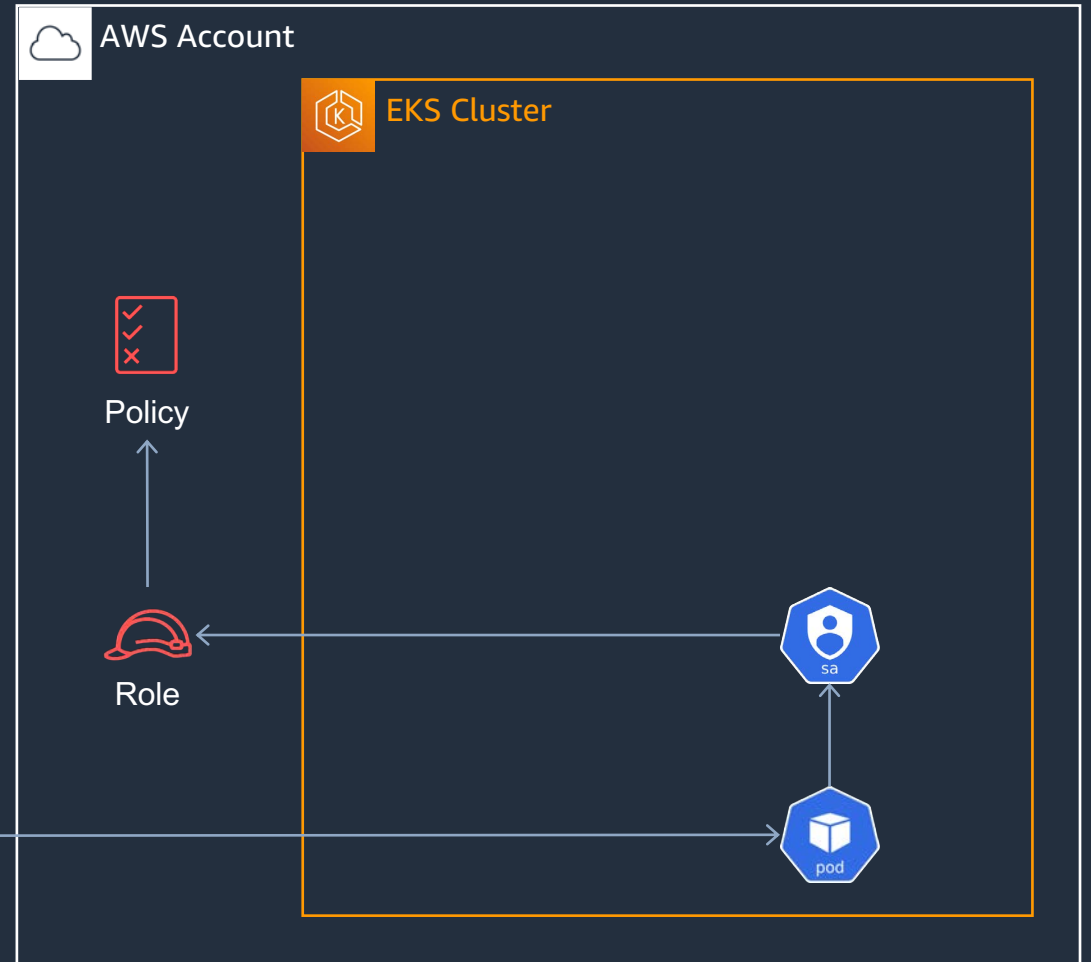


# Addons



## AWS-LOAD-BALANCER-CONTROLLER

```
resource "aws_iam_policy" "albc" {  
  ...  
}  
  
resource "aws_iam_role_policy_attachment" "albc" {  
  role      = aws_iam_role.albc.name  
  policy_arn = aws_iam_policy.albc.arn  
}  
  
resource "aws_iam_role" "albc" {  
  ...  
}  
  
resource "helm_release" "albc" {  
  ...  
  values = [  
    serviceAccount = {  
      create = true  
      annotations = {  
        eks.amazonaws.com/role-arn = aws_iam_role.albc.arn  
      }  
    }  
  ]  
  ...  
}
```







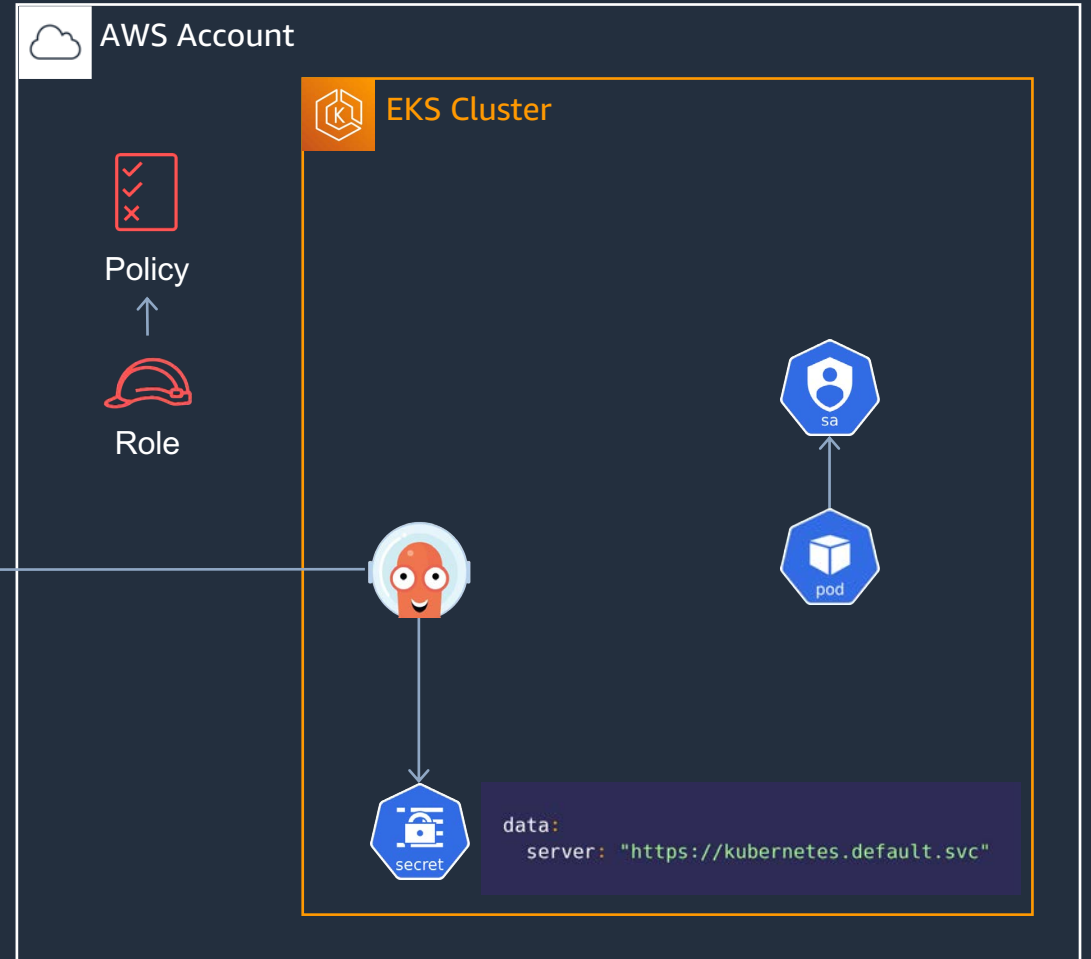
# Addons



## AWS-LOAD-BALANCER-CONTROLLER



```
resource "aws_iam_policy" "albc" {  
  ...  
}  
  
resource "aws_iam_role_policy_attachment" "albc" {  
  role      = aws_iam_role.albc.name  
  policy_arn = aws_iam_policy.albc.arn  
}  
  
resource "aws_iam_role" "albc" {  
  ...  
}
```



# Addons



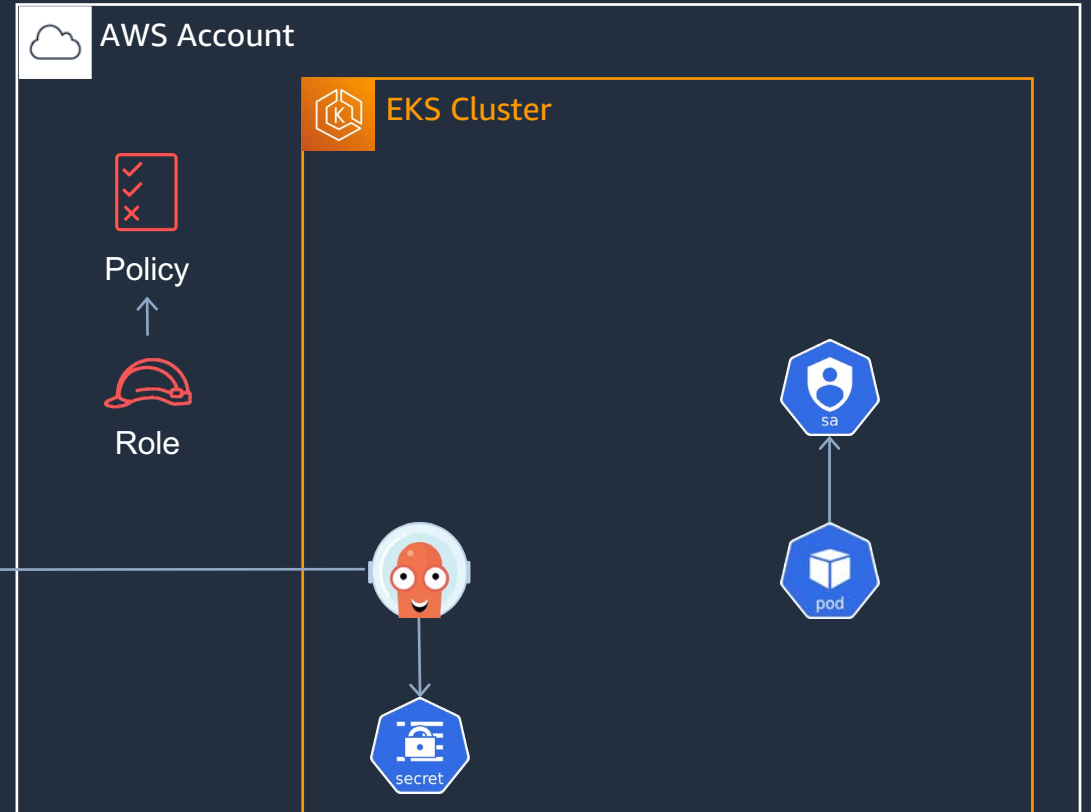
## AWS-LOAD-BALANCER-CONTROLLER

```
resource "aws_iam_policy" "albc" {
  ...
}

resource "aws_iam_role_policy_attachment" "albc" {
  role      = aws_iam_role.albc.name
  policy_arn = aws_iam_policy.albc.arn
}

resource "aws_iam_role" "albc" {
  ...
}

resource "kubernetes_secret_v1" "cluster" {
  ...
  metadata {
    annotations = [
      "albc_iam_role_arn: ${aws_iam_role.albc.arn}"
    ]
  }
  ...
}
```



```
data:
  server: "https://kubernetes.default.svc"
metadata:
  annotations:
    albc_iam_role_arn: arn:aws:iam::111122223333:role/albc-XXXX
```



# Addons



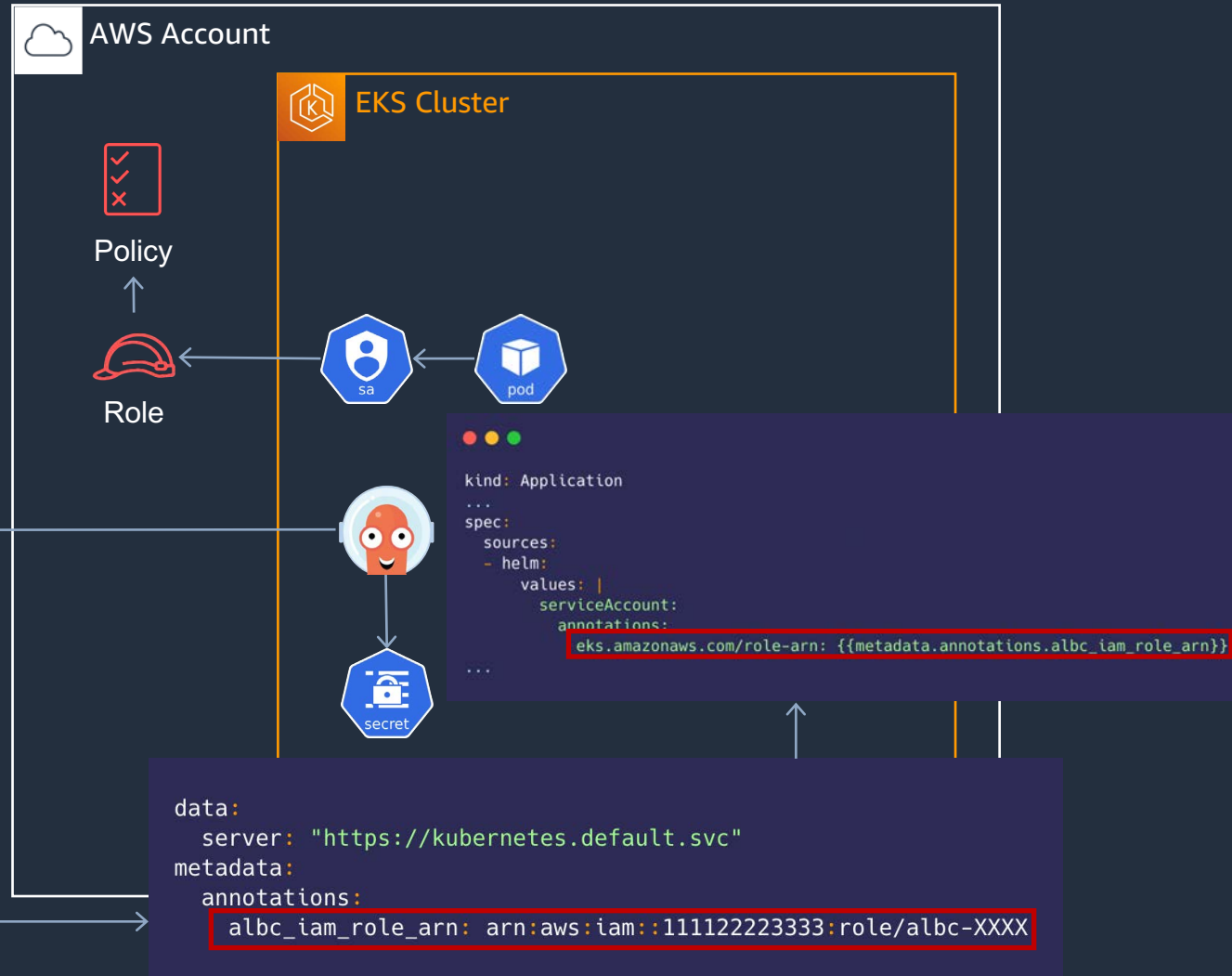
## AWS-LOAD-BALANCER-CONTROLLER

```
resource "aws_iam_policy" "albc" {
  ...
}

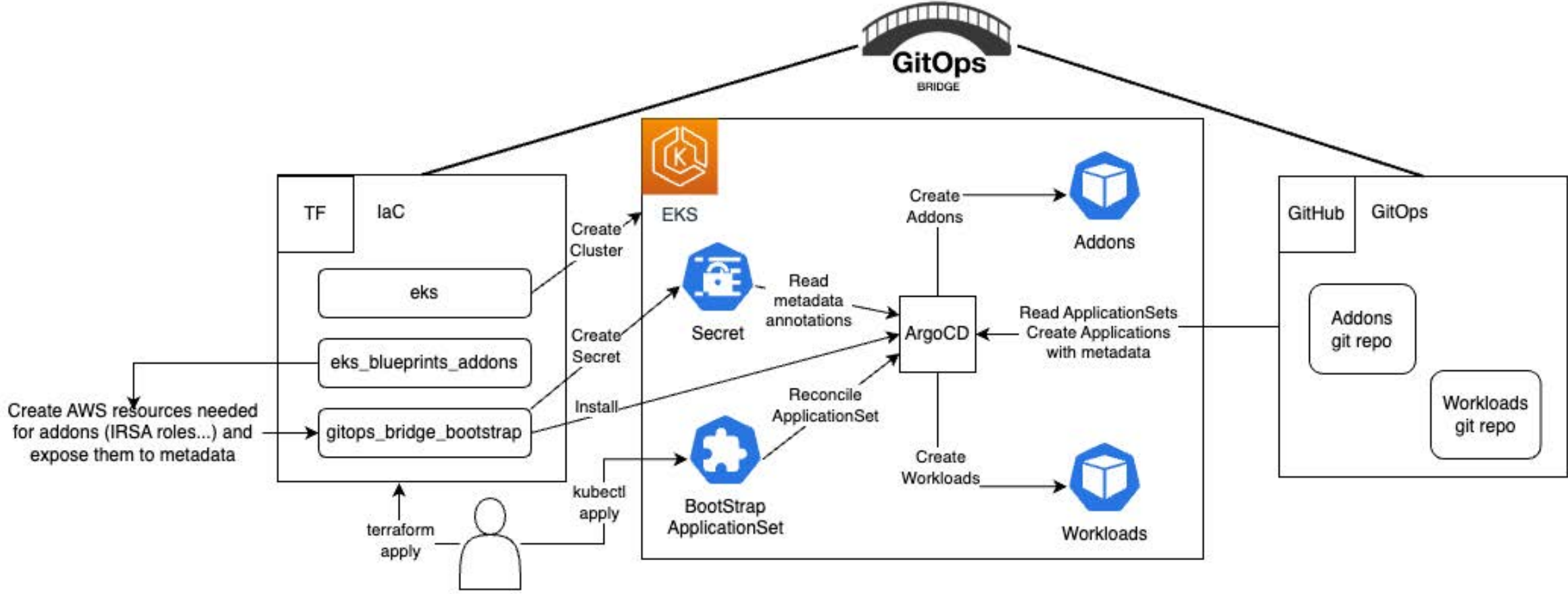
resource "aws_iam_role_policy_attachment" "albc" {
  role      = aws_iam_role.albc.name
  policy_arn = aws_iam_policy.albc.arn
}

resource "aws_iam_role" "albc" {
  ...
}

resource "kubernetes_secret_v1" "cluster" {
  ...
  metadata {
    annotations = [
      "albc_iam_role_arn: ${aws_iam_role.albc.arn}"
    ]
  }
  ...
}
```

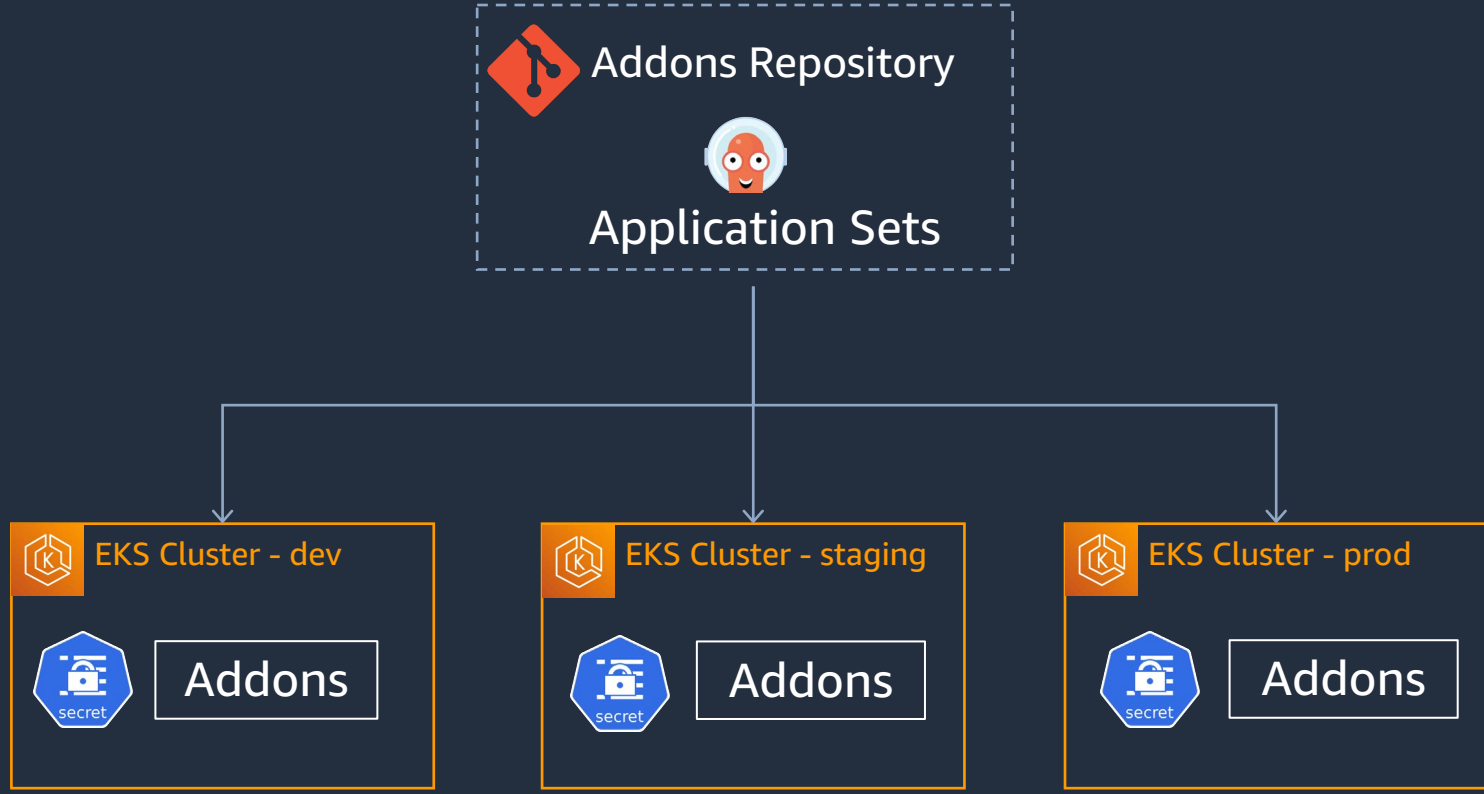


# The Gitops Bridge



# Centralized Cluster Environments


GITOPS REPOSITORY



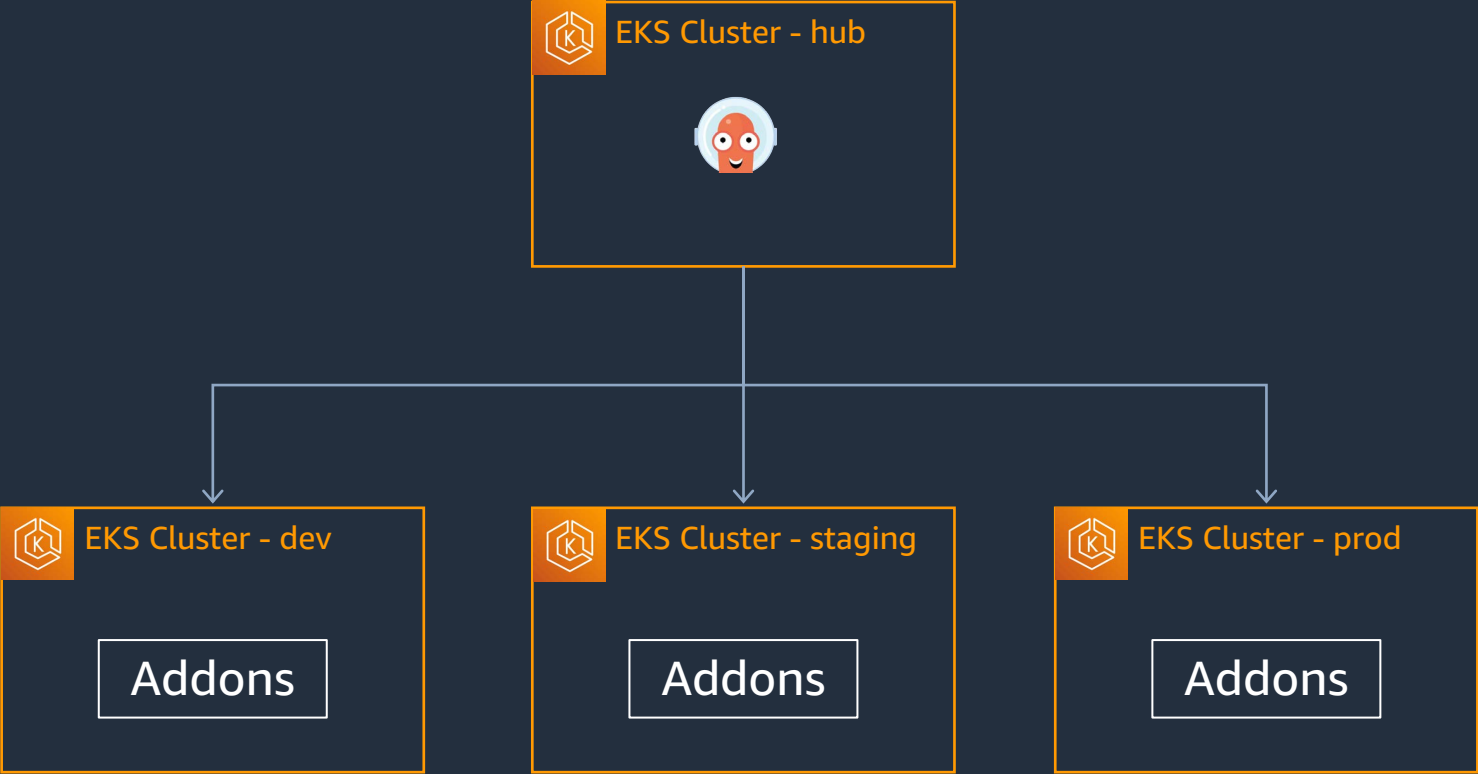
# Multi-Cluster centralized

## HUB-SPOKE TOPOLOGY

```
├── hub  
│   └── main.tf  
└── spokes  
    ├── main.tf  
    └── workspaces  
        ├── dev.tfvars  
        ├── prod.tfvars  
        └── staging.tfvars
```

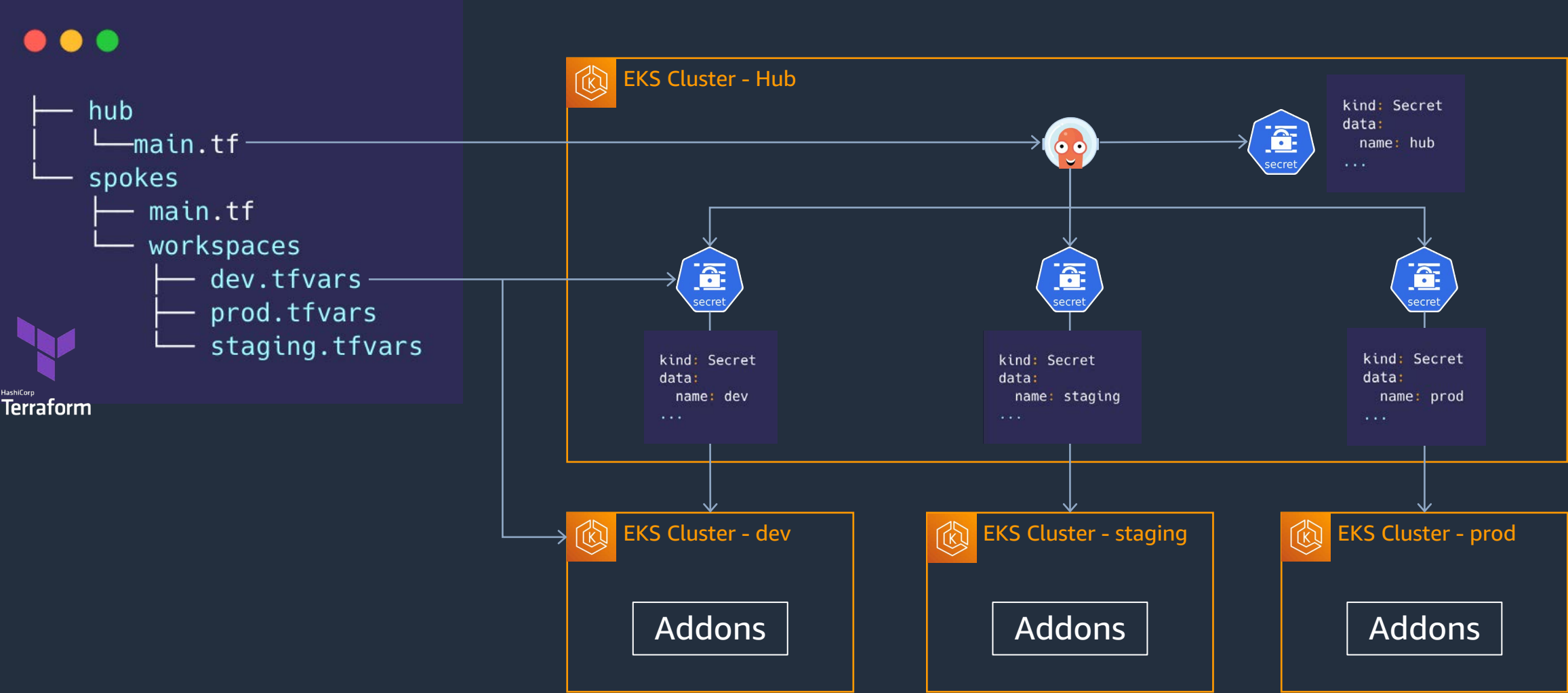


HashiCorp  
**Terraform**



# Multi-Cluster centralized

## HUB-SPOKE TOPOLOGY



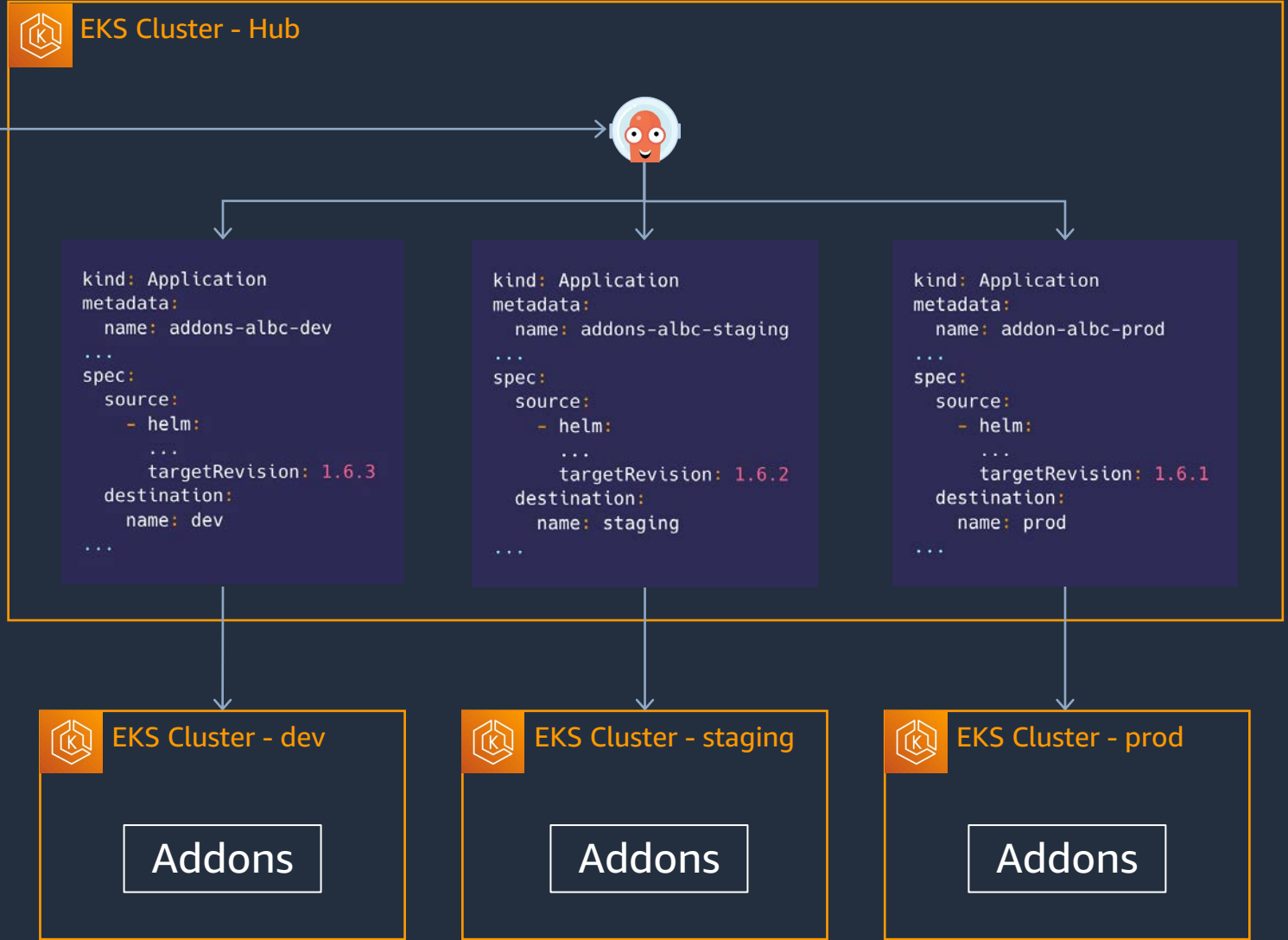
# Multi-Cluster centralized

## HUB-SPOKE TOPOLOGY



```
kind: ApplicationSet
metadata:
  name: addons-albc
...
spec:
  generators:
    - clusters:
        selector:
          matchLabels: dev
        values:
          addonCharVersion: 1.6.3
    - clusters:
        selector:
          matchLabels: staging
        values:
          addonCharVersion: 1.6.2
    - clusters:
        selector:
          matchLabels: dev
        values:
          addonCharVersion: 1.6.1
...

```



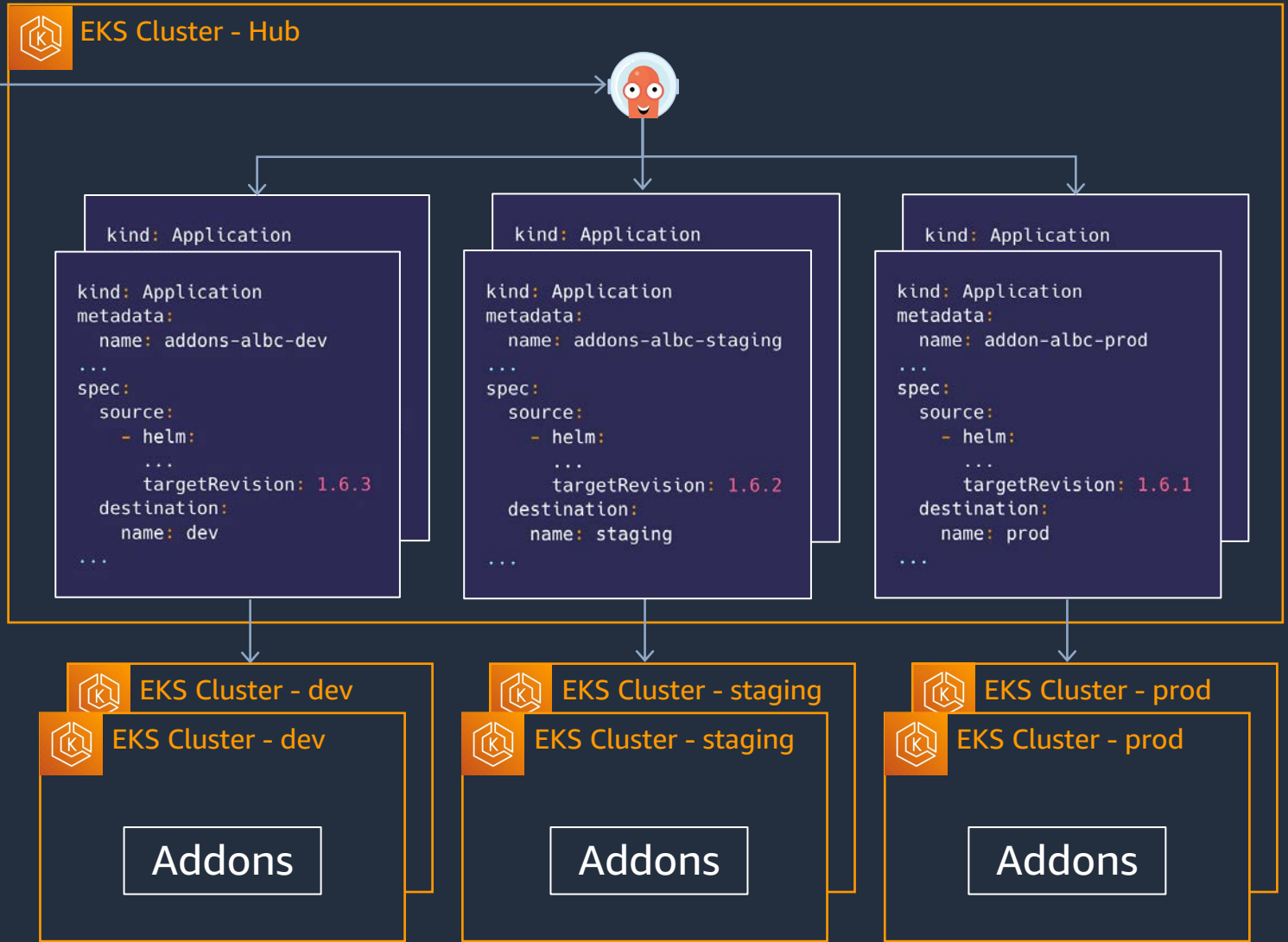


# Multi-Cluster centralized

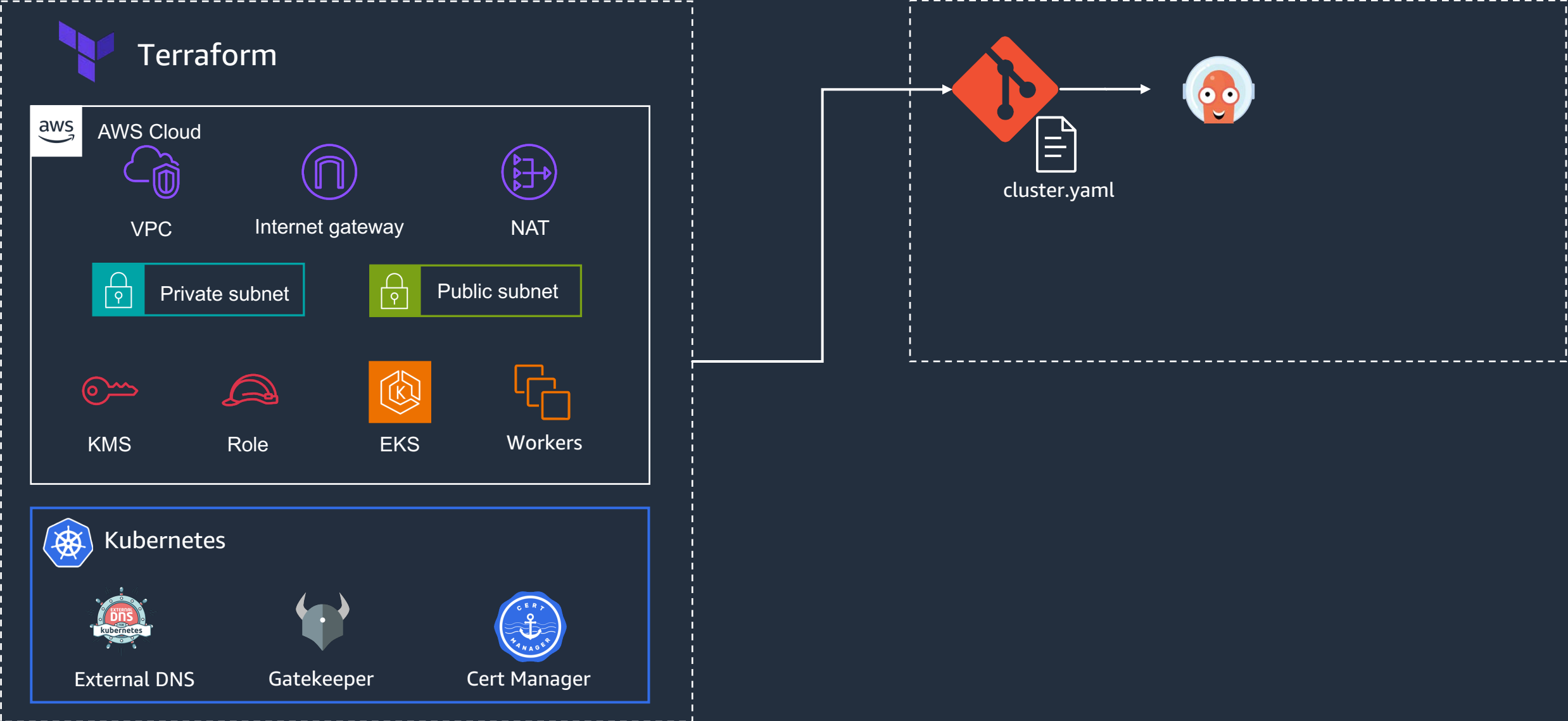
## HUB-SPOKE TOPOLOGY



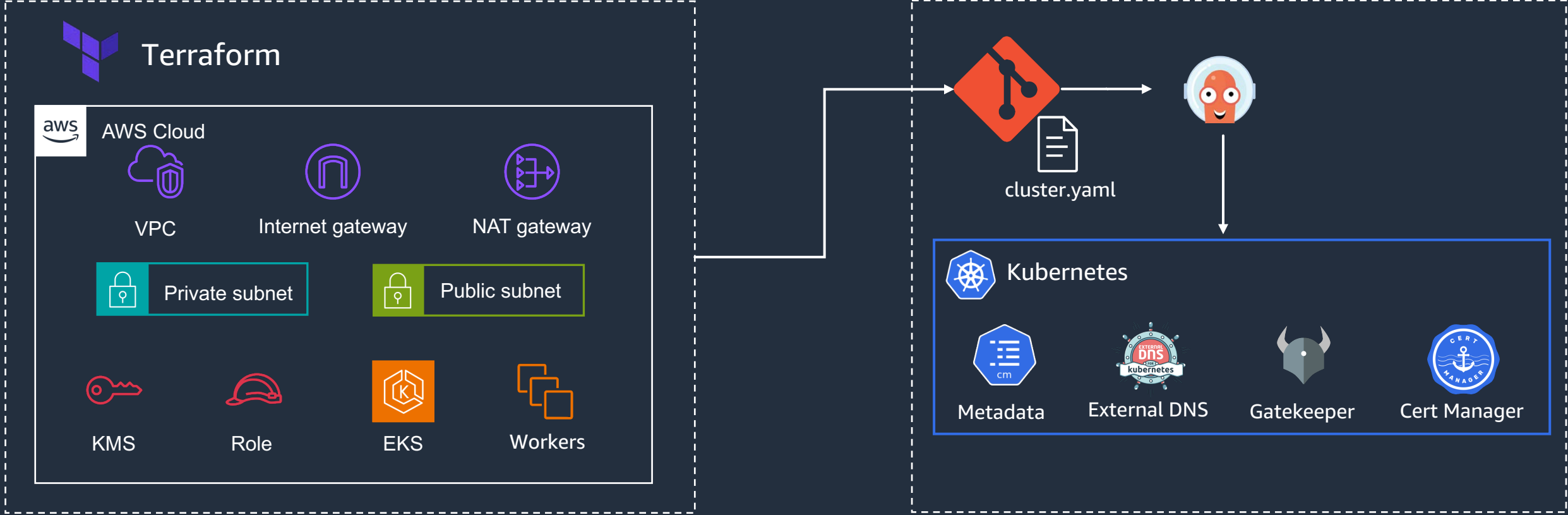
```
kind: ApplicationSet
metadata:
  name: addons-albc
...
spec:
  generators:
    - clusters:
        selector:
          matchLabels: dev
        values:
          addonCharVersion: 1.6.3
    - clusters:
        selector:
          matchLabels: staging
        values:
          addonCharVersion: 1.6.2
    - clusters:
        selector:
          matchLabels: dev
        values:
          addonCharVersion: 1.6.1
  ...
```



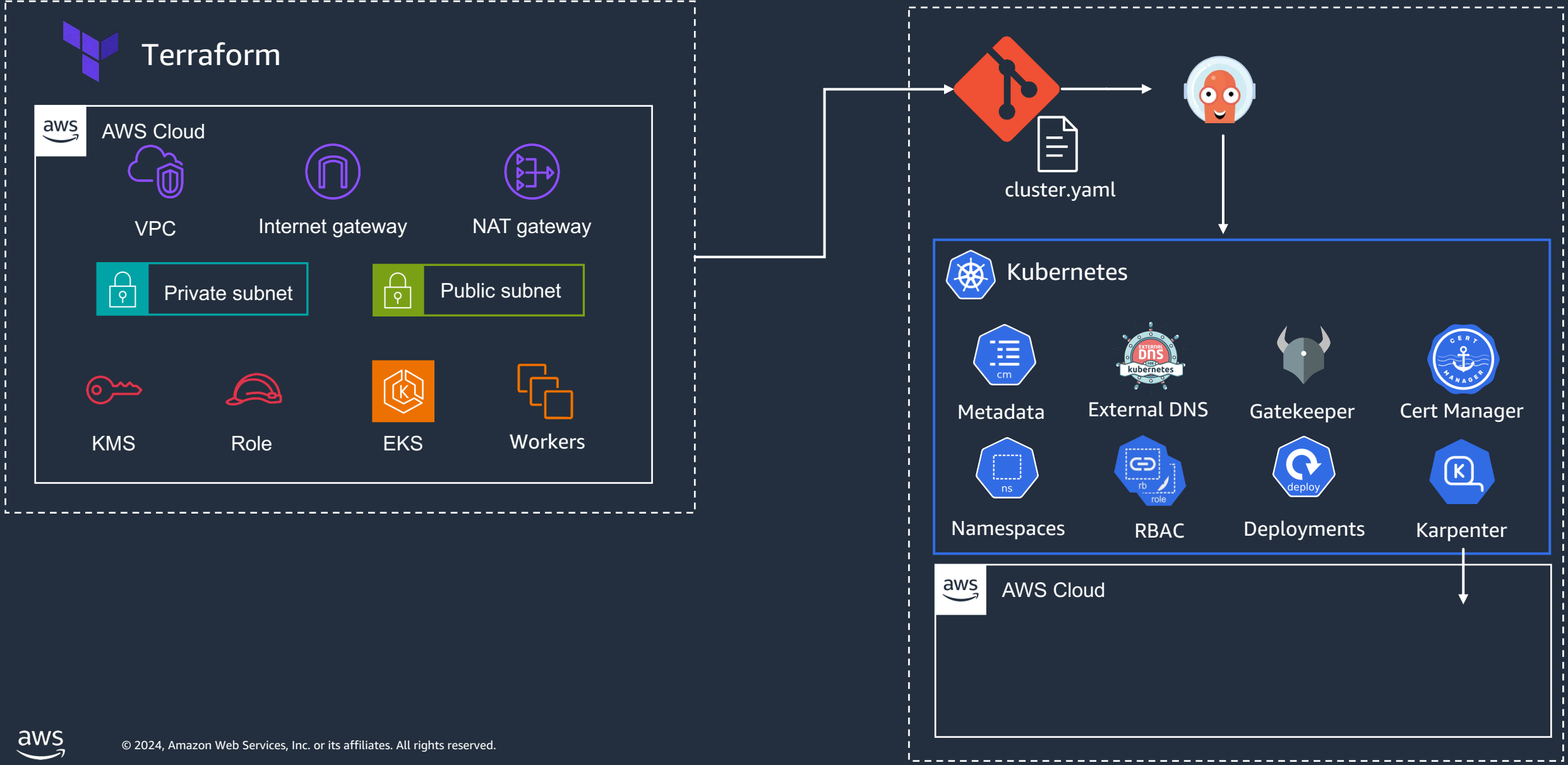
# GitOps Bridge: From IaC to GitOps



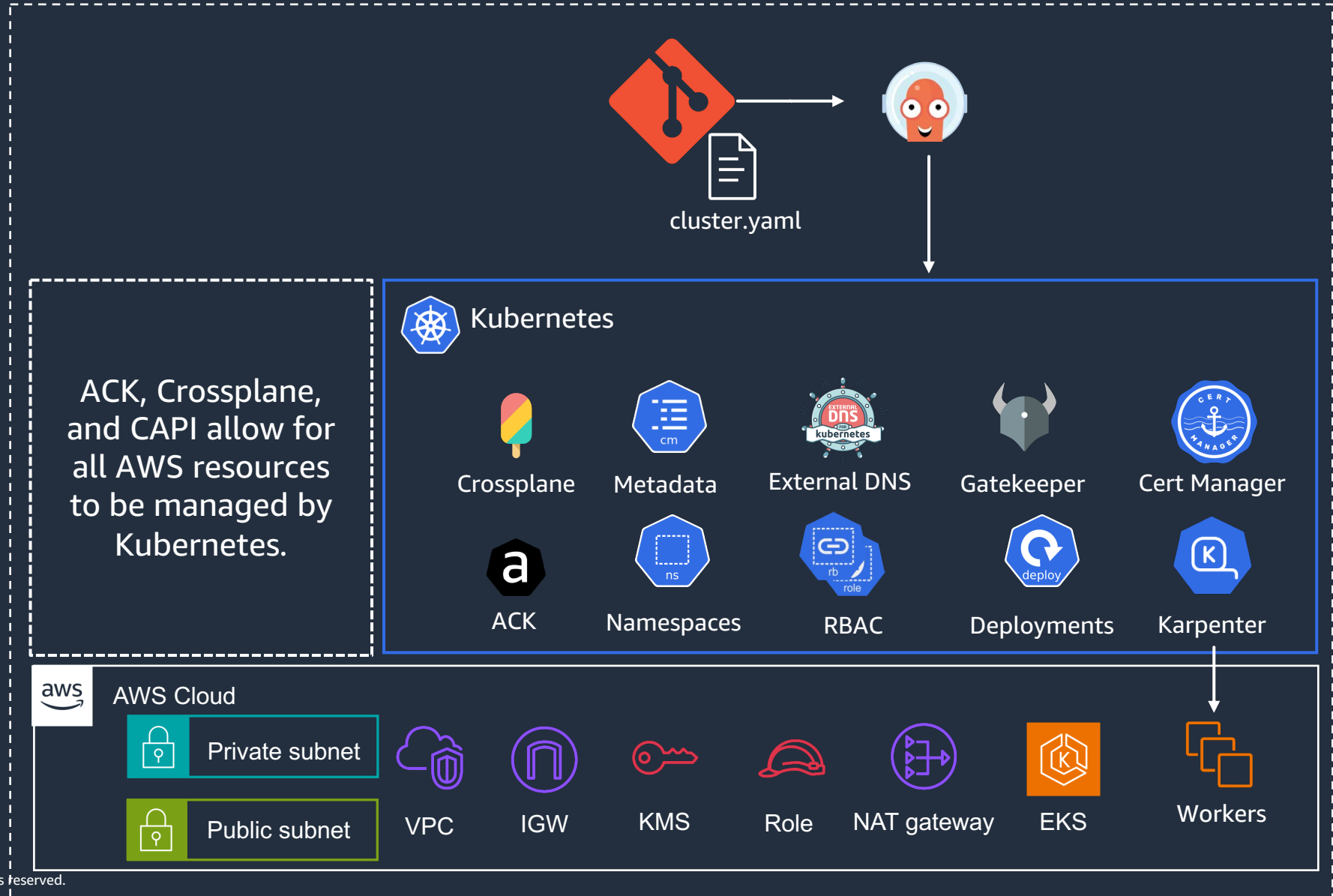
# GitOps Bridge: From IaC to GitOps



# GitOps Bridge: From IaC to GitOps



# Kubernetes Native Fleet Management

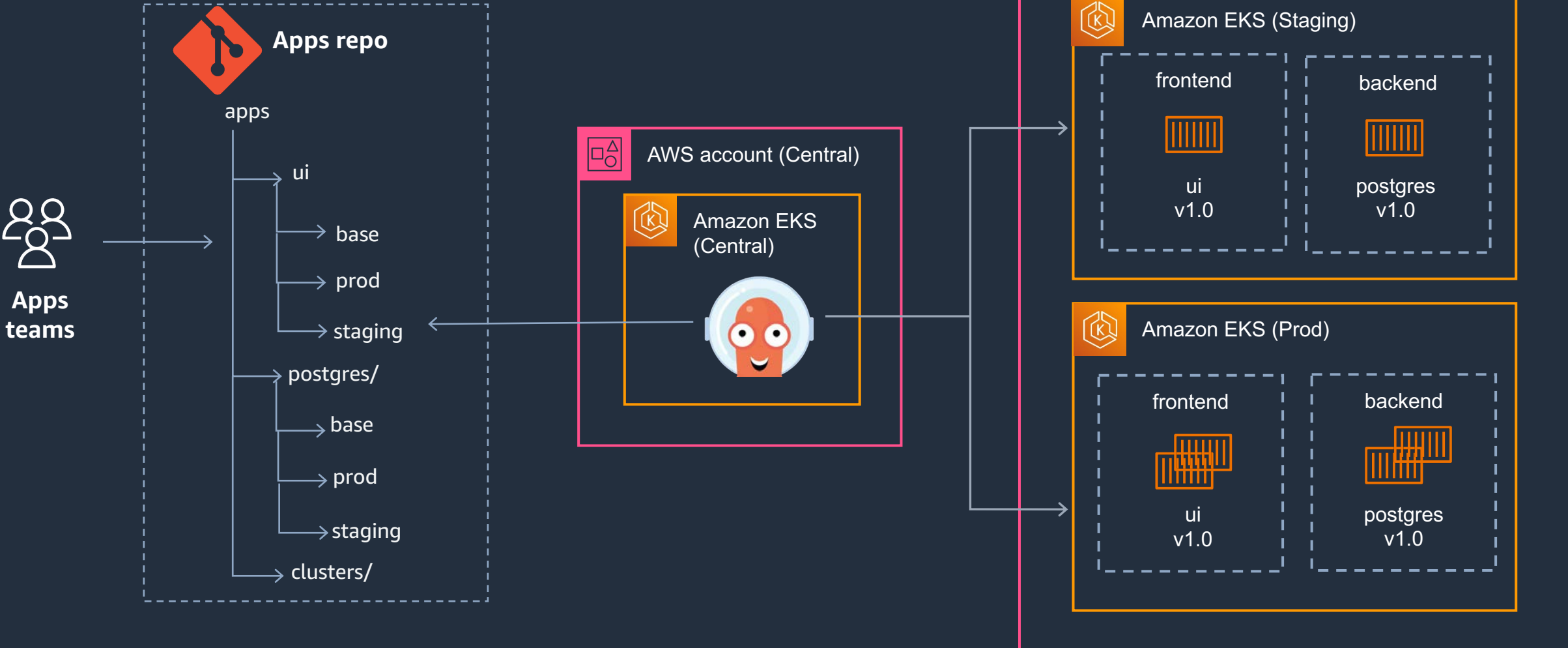


# Configuration Management



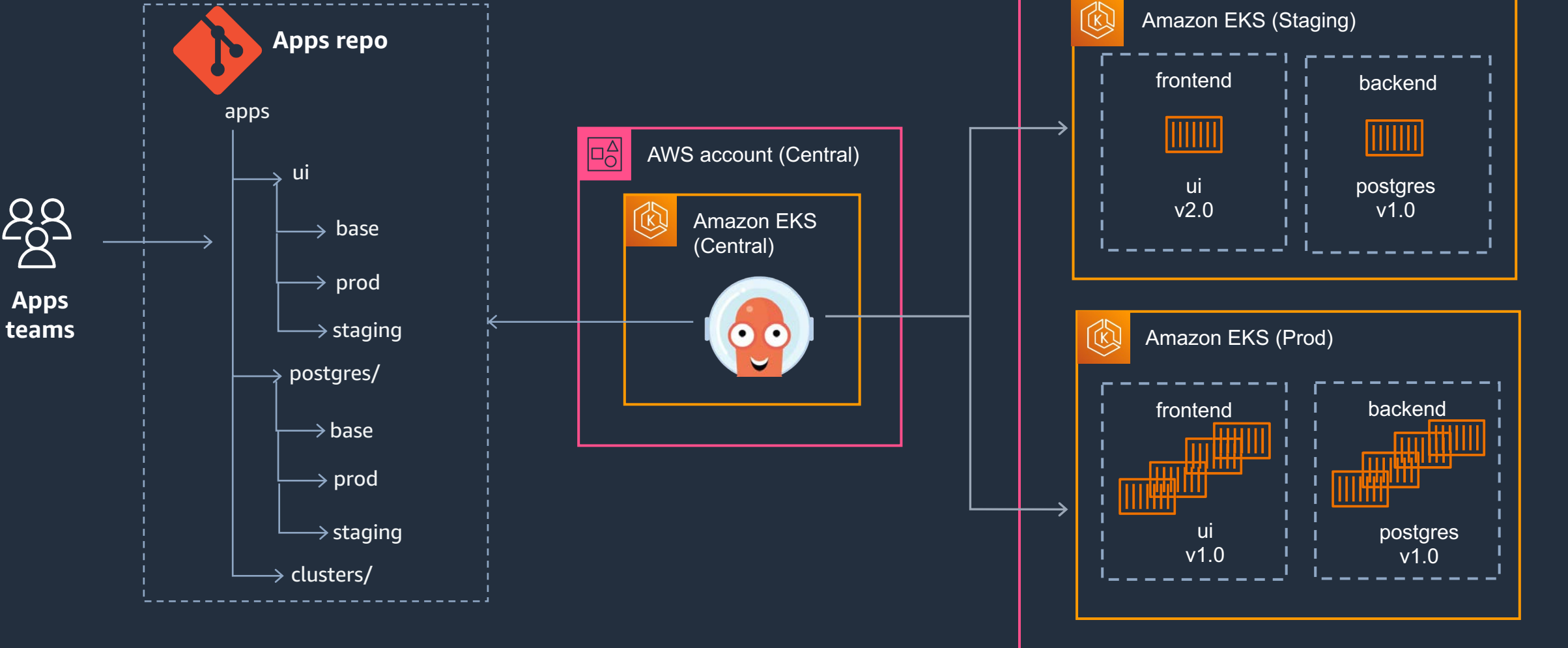
# Configuration management

## ENVIRONMENTS (STAGING VS. PROD)



# Configuration management

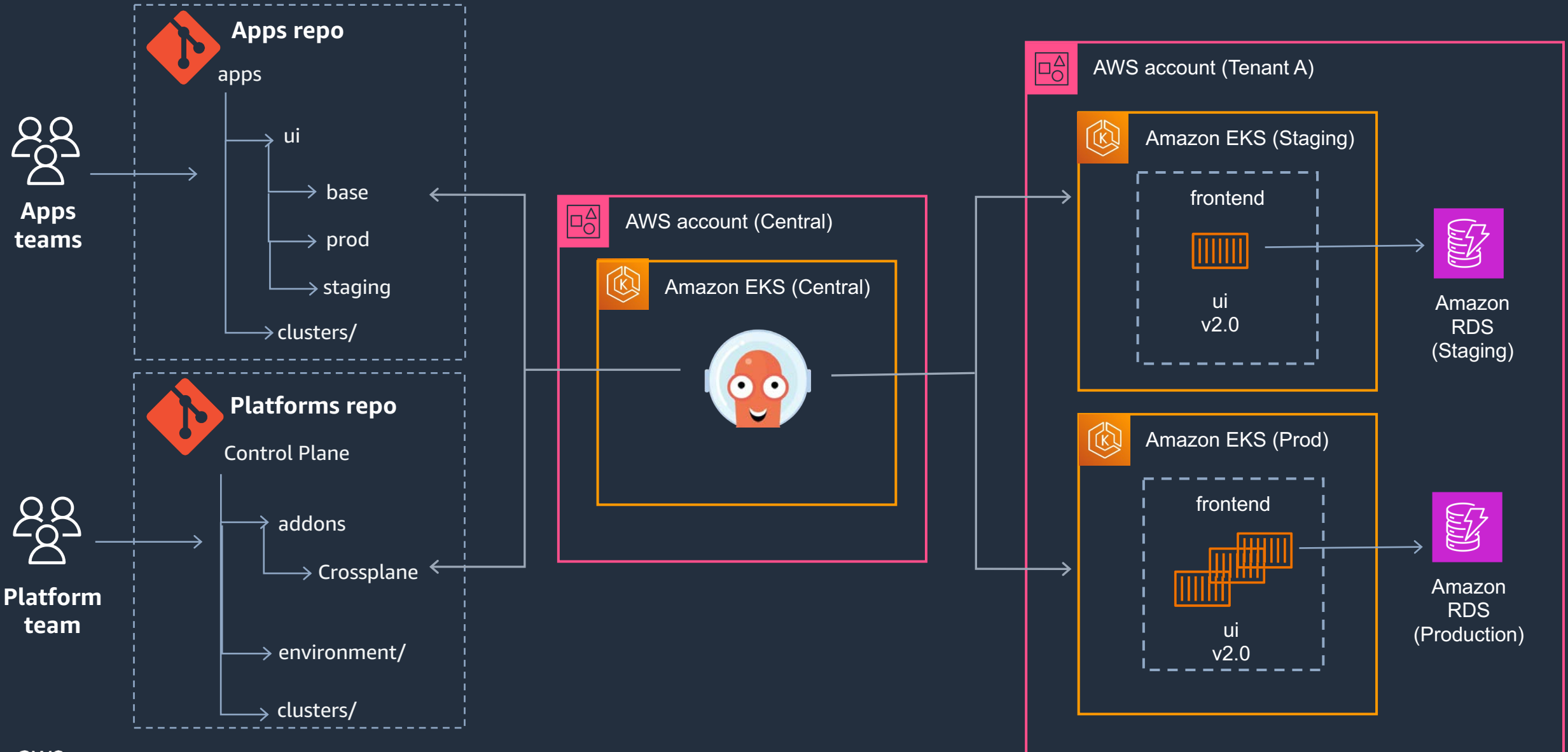
## DEPLOY AND SCALE





# Configuration management

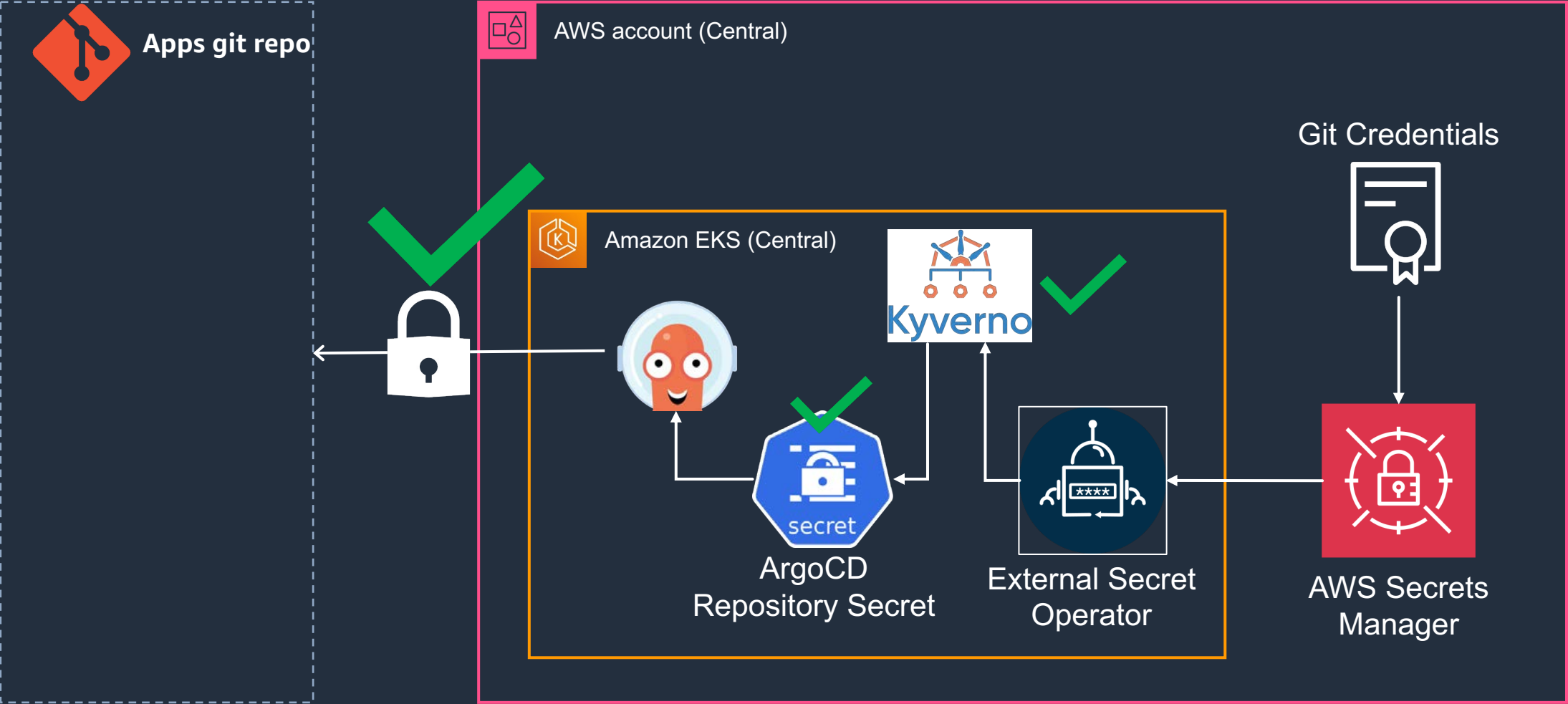
AWS SERVICES WITH GITOPS



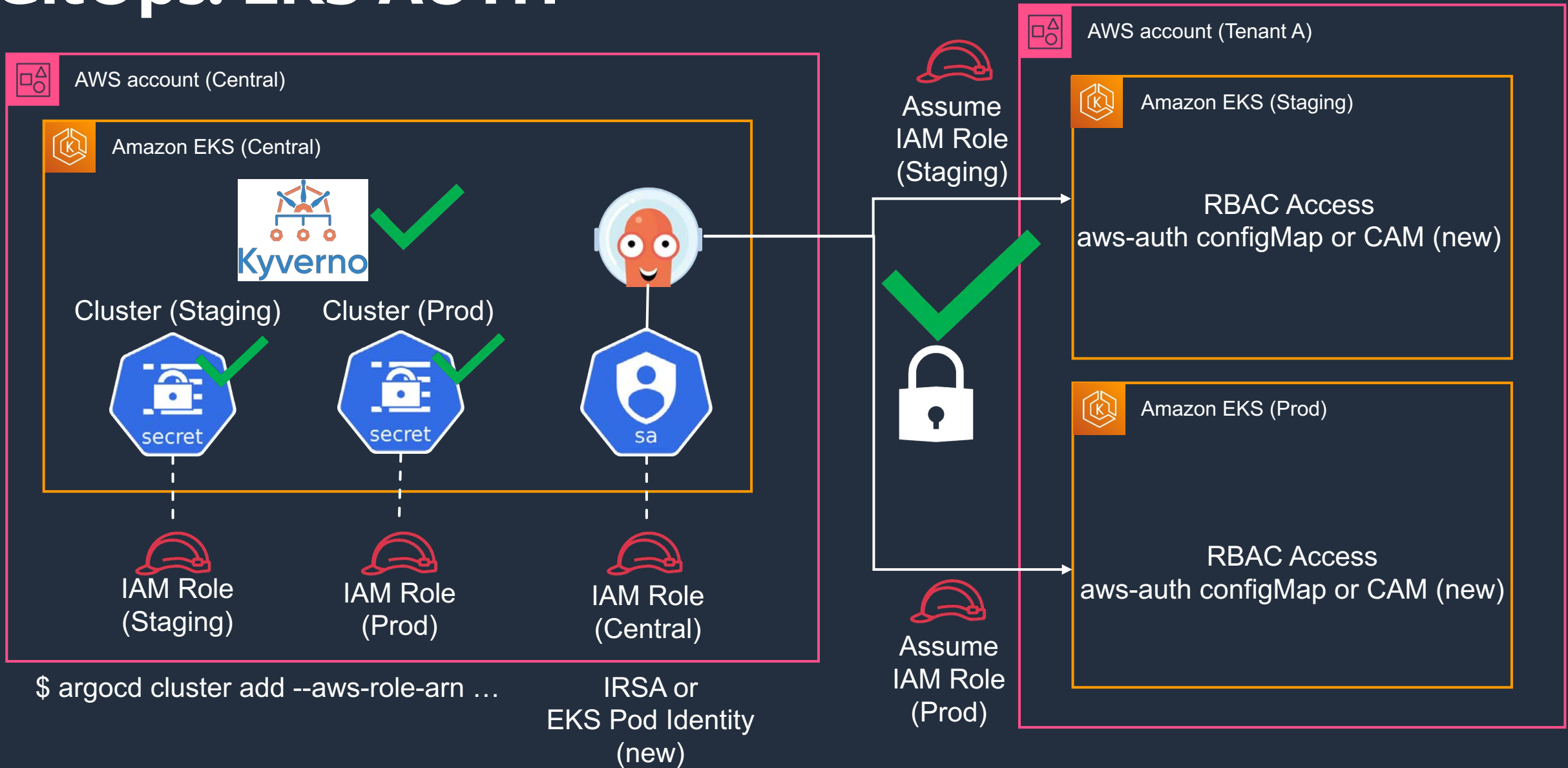
# Secured Gitops Best Practices



# GitOps: Git Credentials



# GitOps: EKS AUTH



# GitOps EKS Blueprints



<https://aws-ia.github.io/terraform-aws-eks-blueprints/patterns/gitops-getting-started-argocd/>

# ArgoCD on EKS Workshop



<https://catalog.us-east-1.prod.workshops.aws/workshops/e36277ba-4094-4df4-b62f-d1e655800123>



# Thank you!

**Yuriy Bezsonov**

<https://www.linkedin.com/in/yuriybezsonov/>

Please respond to 1-minute survey



<https://pulse.aws/survey/BFRATGVV>