Tanzu Cloud Native webinars

Kubernetes as a service to your users

Robert Jensen

Lead Systems Engineer @Vmware

@rhjensen / jensenr@vmware.com

Viktor Van Der Berg

Lead Systems Engineer @Vmware

@viktoriousss / vvandenberg@vmware.com

Agenda

- Multi cloud deployments
- Policies / Guardrails
- Scale out
- Upgrades
- Backup
- Complete automated delivery with CD on top

The purpose of this Webinar

Is to give an introduction to Cloud Native concepts and technologies.

We start with the basics, and try to take it to the next level.

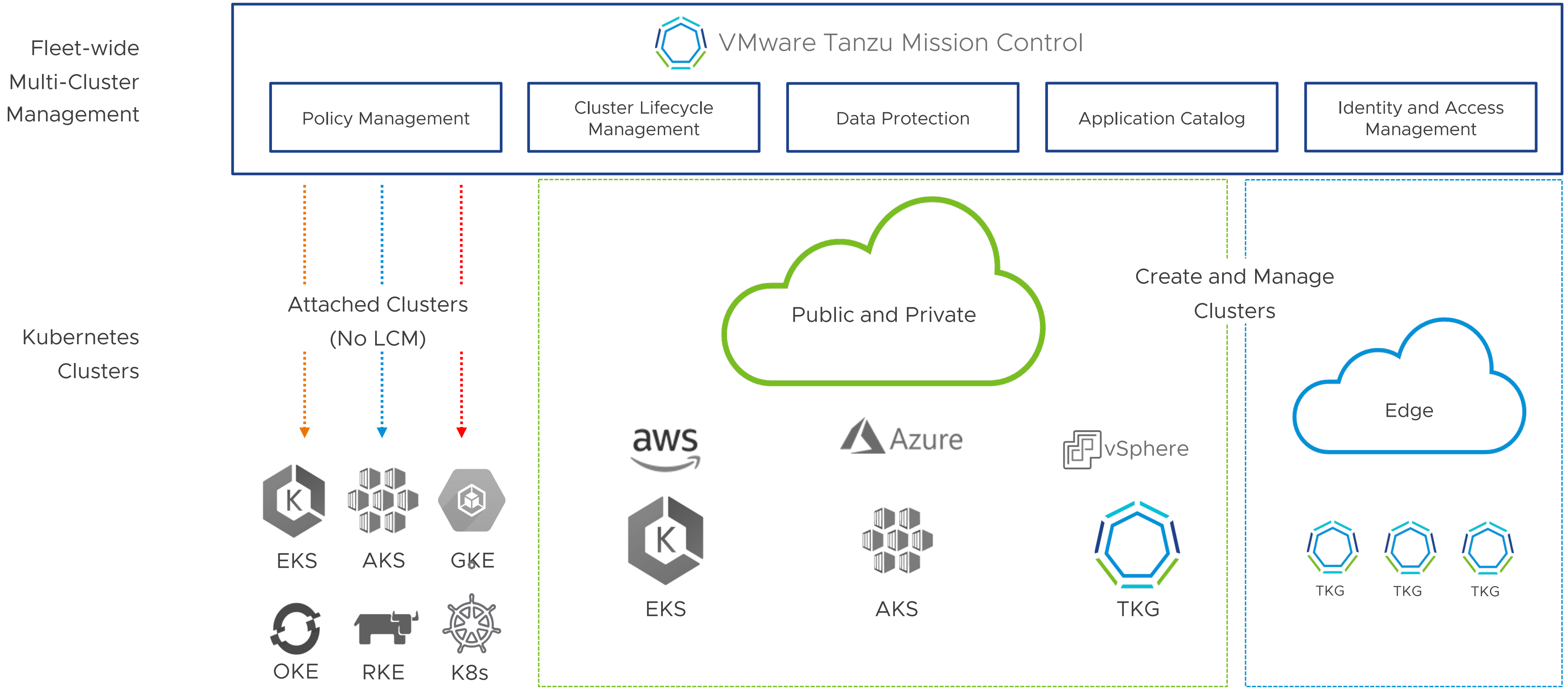
Questions : Please use the Q/A function. We will look at them In the end.

Tanzu Mission Control

Automated Kubernetes to your users

VMware Tanzu® Mission Control™

Centralized management hub with a robust policy engine that simplifies fleet Kubernetes management.



Comparing Registered vs Attach

VMware Tanzu Mission Control

Tanzu Kubernetes

EKS

AKS

	Tanzu Kubernetes / EKS		CNCF-Conformant (Attach)
✓	Lifecycle Management	✗	Lifecycle Management
✓	Apply policies	✓	Apply policies
✓	Manage Namespaces	✓	Manage Namespaces
✓	Deploy Packages	✓	Deploy Packages
✓	Data protection	✓	Data protection
✓	Inspections	✓	Inspections
✓	Continuous Delivery for clusters	✓	Continuous Delivery for clusters
✓	Enable integrations	✓	Enable integrations

EKS

AKS

GKE

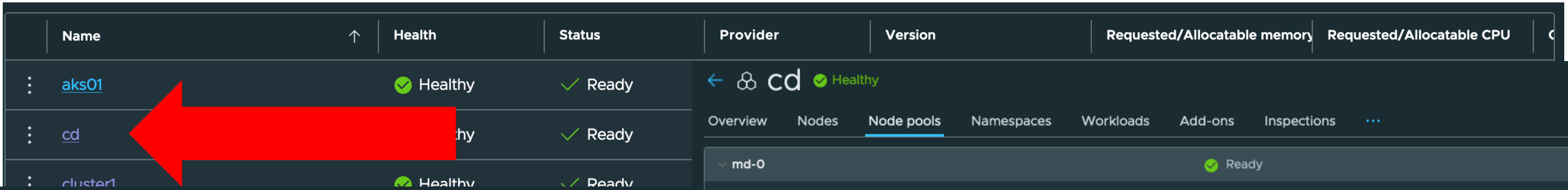
K8s

RKE

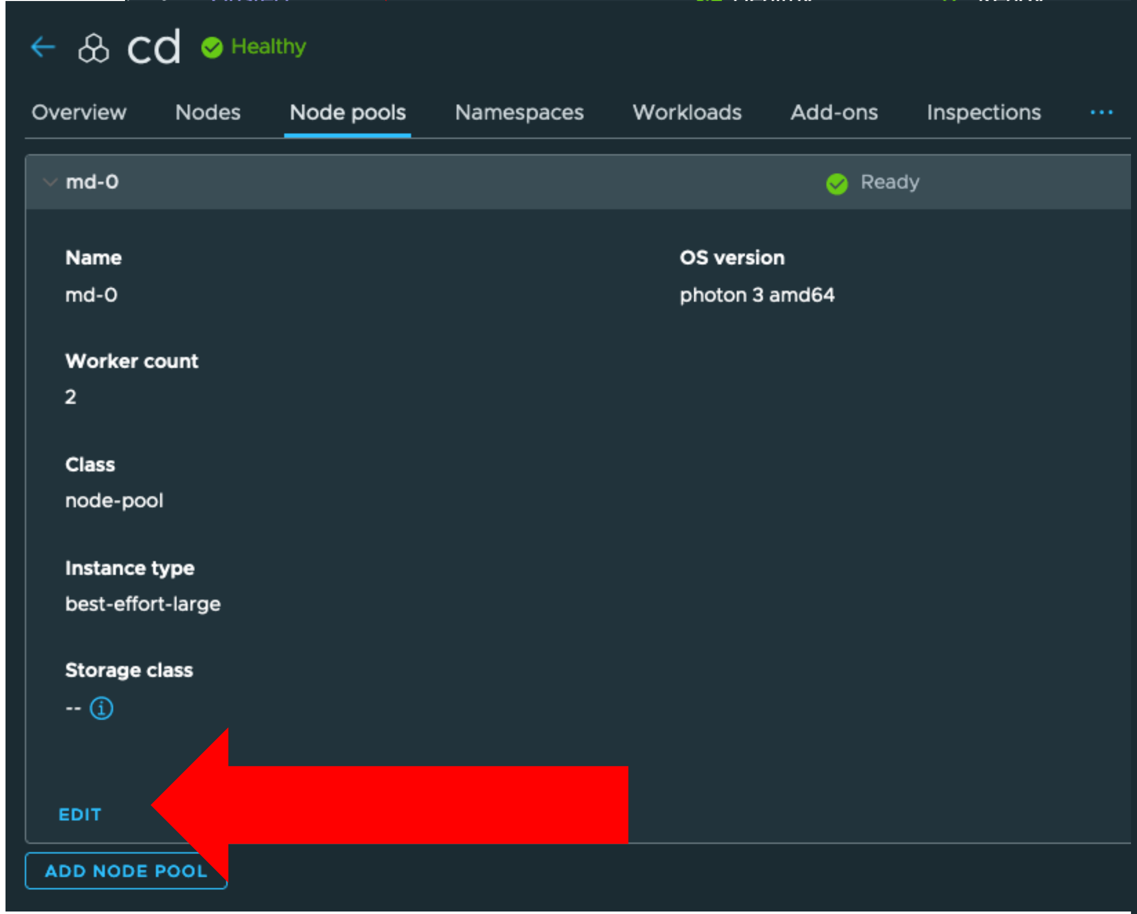
OKE

Scale Out

Scale Out



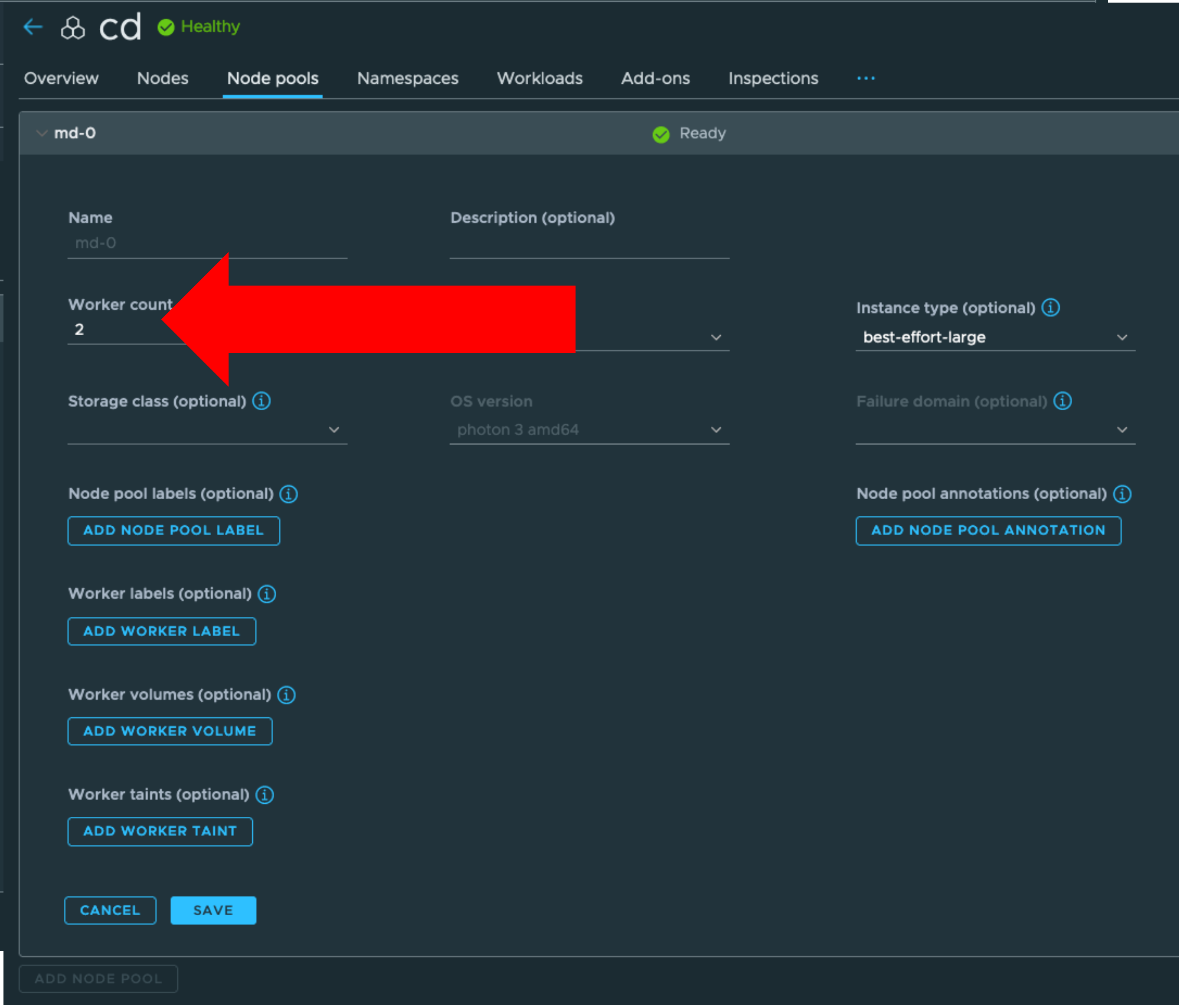
	Name	Health	Status	Provider	Version	Requested/Allocatable memory	Requested/Allocatable CPU
⋮	aks01	Healthy	Ready				
⋮	cd	…hy	Ready				
⋮	cluster1	Healthy	Ready				



cd Healthy

md-0 Ready

Name	OS version
md-0	photon 3 amd64
Worker count	
2	
Class	
node-pool	
Instance type	
best-effort-large	
Storage class	
--	



cd Healthy

md-0 Ready

Name: md-0

Description (optional):

Worker count: 2

Instance type (optional): best-effort-large

Storage class (optional):

OS version: photon 3 amd64

Failure domain (optional):

Node pool labels (optional)

Node pool annotations (optional)

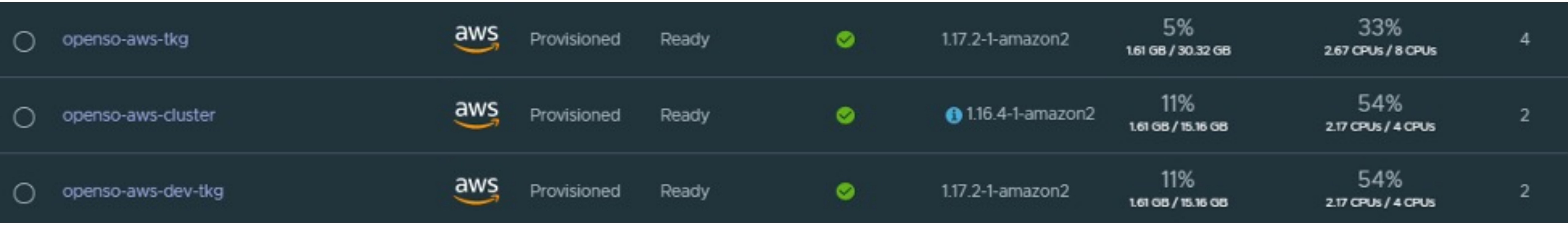
Worker labels (optional)

Worker volumes (optional)

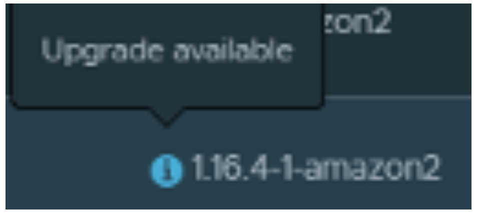
Worker taints (optional)

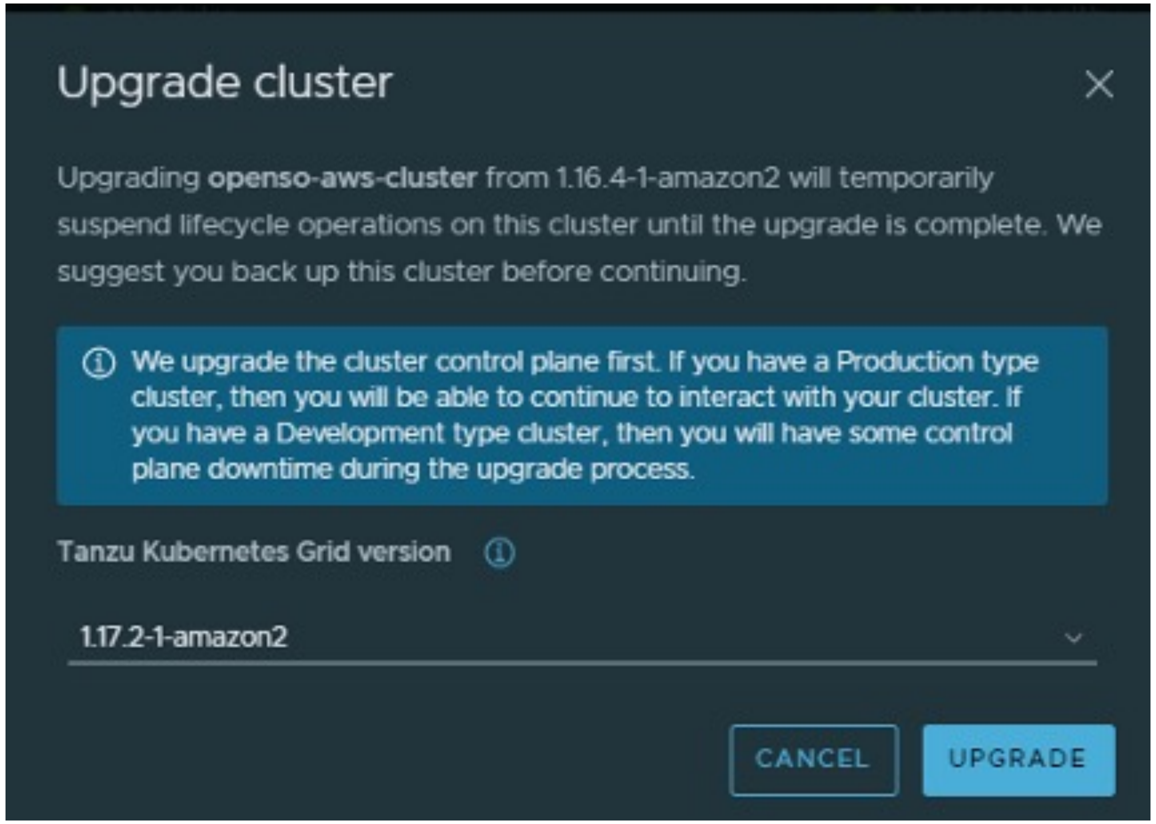
Upgrades

Upgrades



○	openso-aws-tkg	aws	Provisioned	Ready	✓	1.17.2-1-amazon2	5% 1.61 GB / 30.32 GB	33% 2.67 CPUs / 8 CPUs	4
○	openso-aws-cluster	aws	Provisioned	Ready	✓	ℹ 1.16.4-1-amazon2	11% 1.61 GB / 15.16 GB	54% 2.17 CPUs / 4 CPUs	2
○	openso-aws-dev-tkg	aws	Provisioned	Ready	✓	1.17.2-1-amazon2	11% 1.61 GB / 15.16 GB	54% 2.17 CPUs / 4 CPUs	2





Upgrade cluster

Upgrading **openso-aws-cluster** from 1.16.4-1-amazon2 will temporarily suspend lifecycle operations on this cluster until the upgrade is complete. We suggest you back up this cluster before continuing.

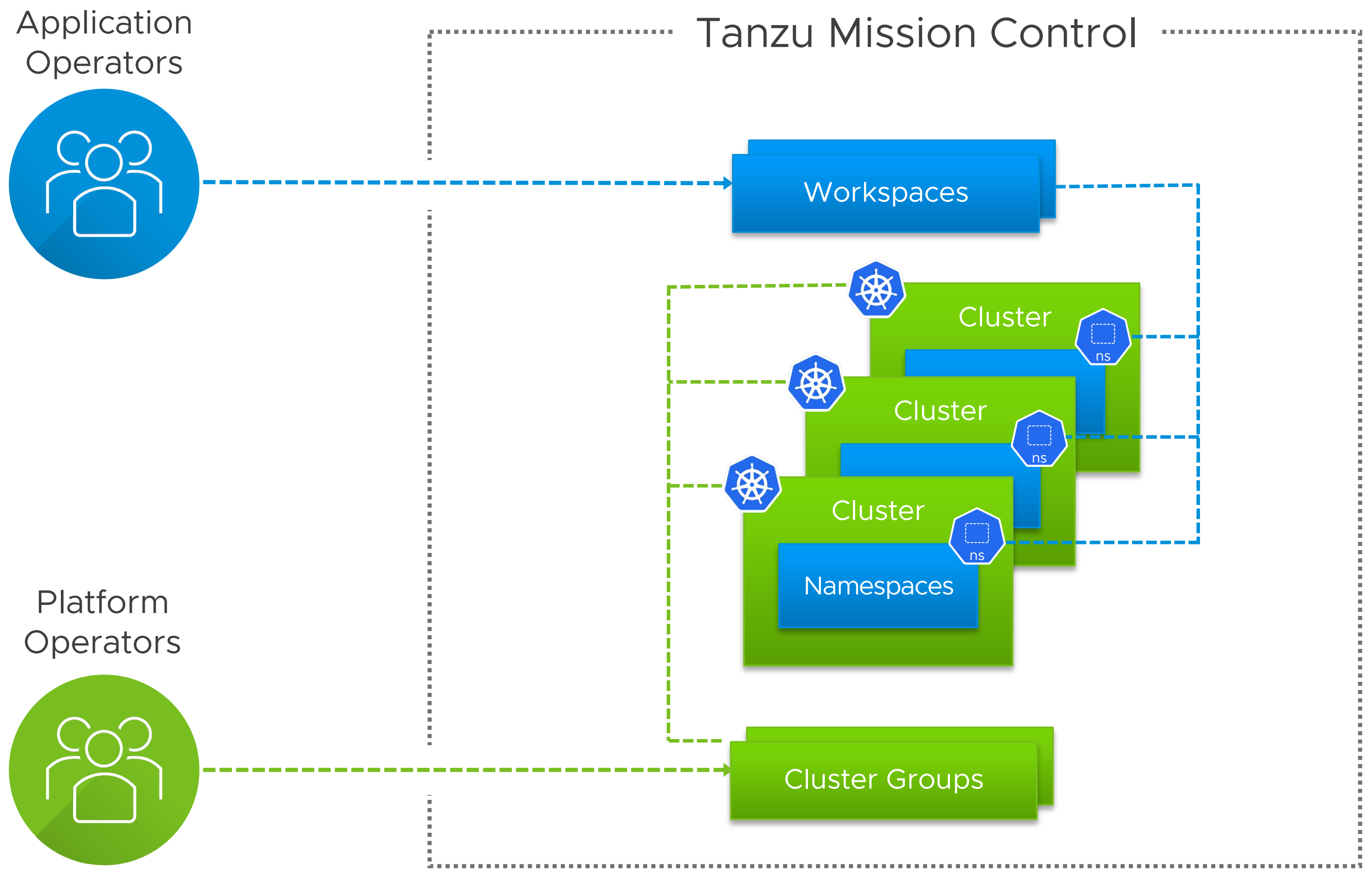
ⓘ We upgrade the cluster control plane first. If you have a Production type cluster, then you will be able to continue to interact with your cluster. If you have a Development type cluster, then you will have some control plane downtime during the upgrade process.

Tanzu Kubernetes Grid version: 1.17.2-1-amazon2

Policy Management

Build a Robust Security Framework

With the global policy engine



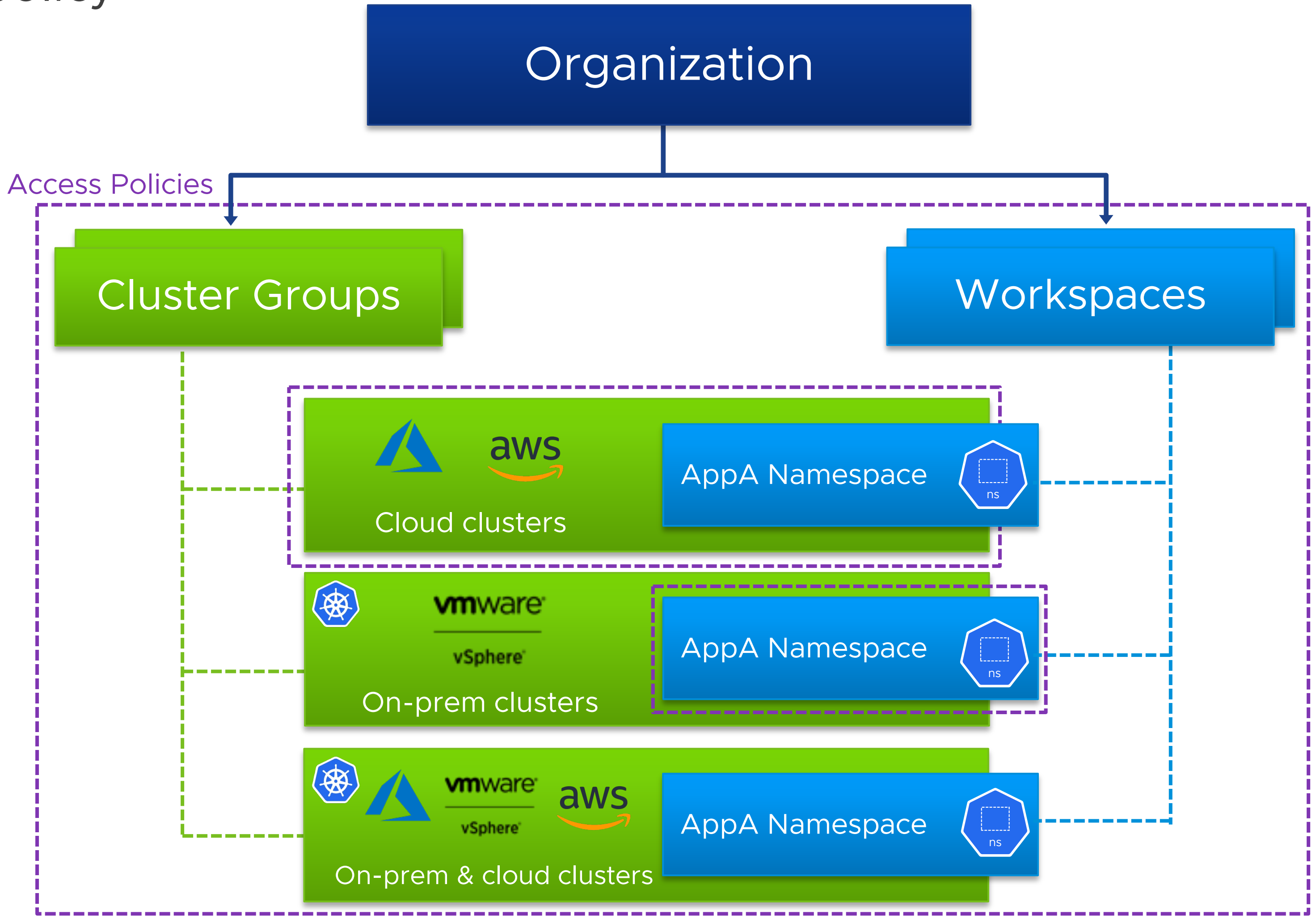
Separate logical groups for Platform and Application teams

Consistent policy application on all attached clusters

Minimal involvement of Help Desk and Security teams

Group Resources for Consistent Guard Rails

With unified identity and access policy

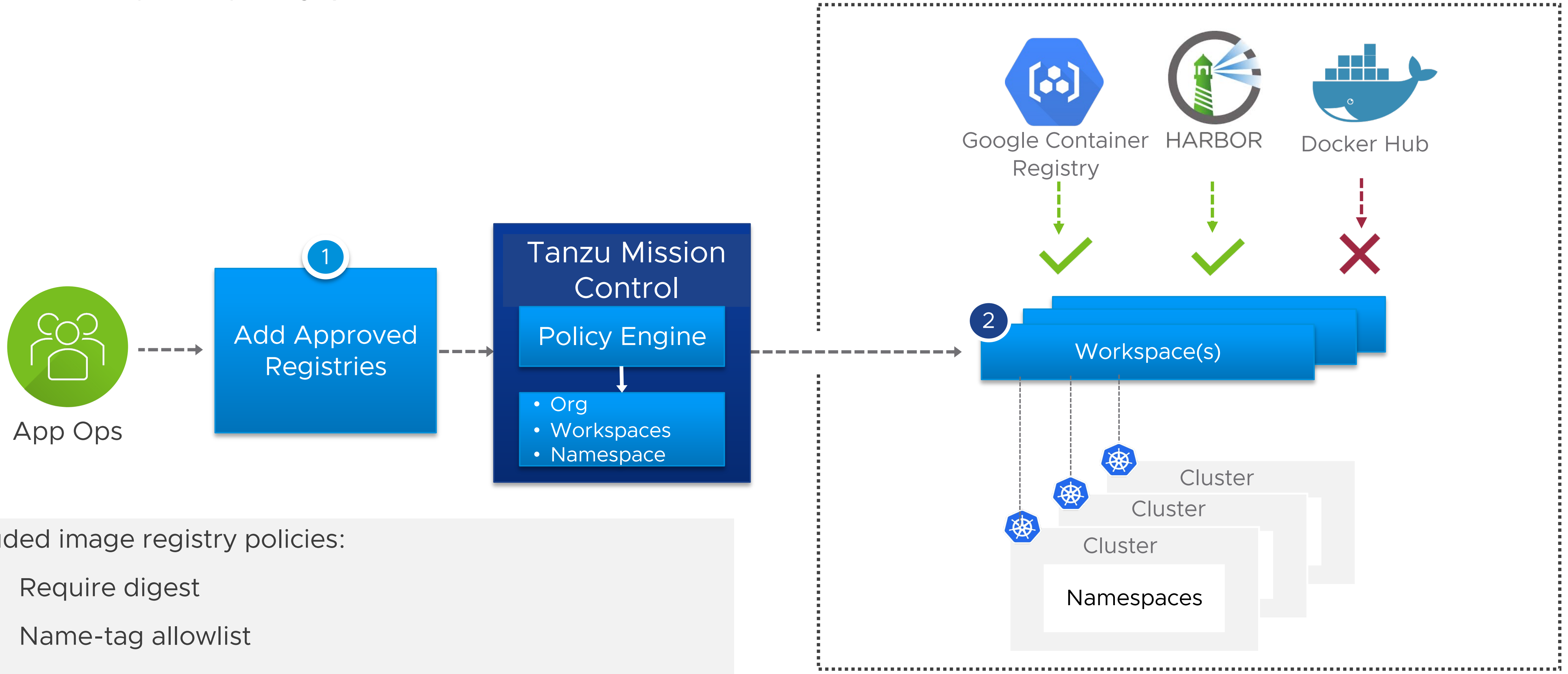


Controlled access to:

- Workspaces
- Namespaces
- Clusters
- Cluster groups

Restrict Access to Image Registries

With image registry policies

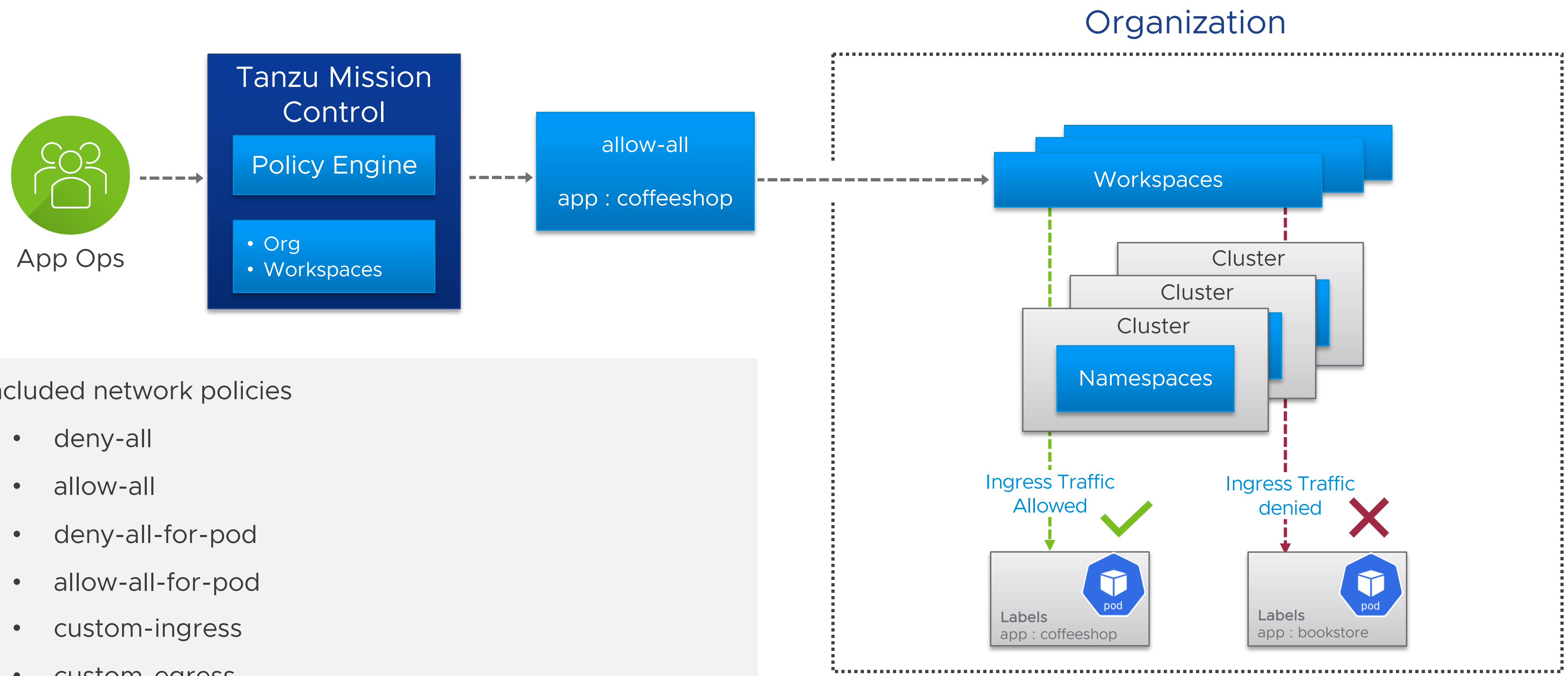


Included image registry policies:

- Require digest
- Name-tag allowlist
- Block latest tag
- Custom

Restrict Pod Network Access

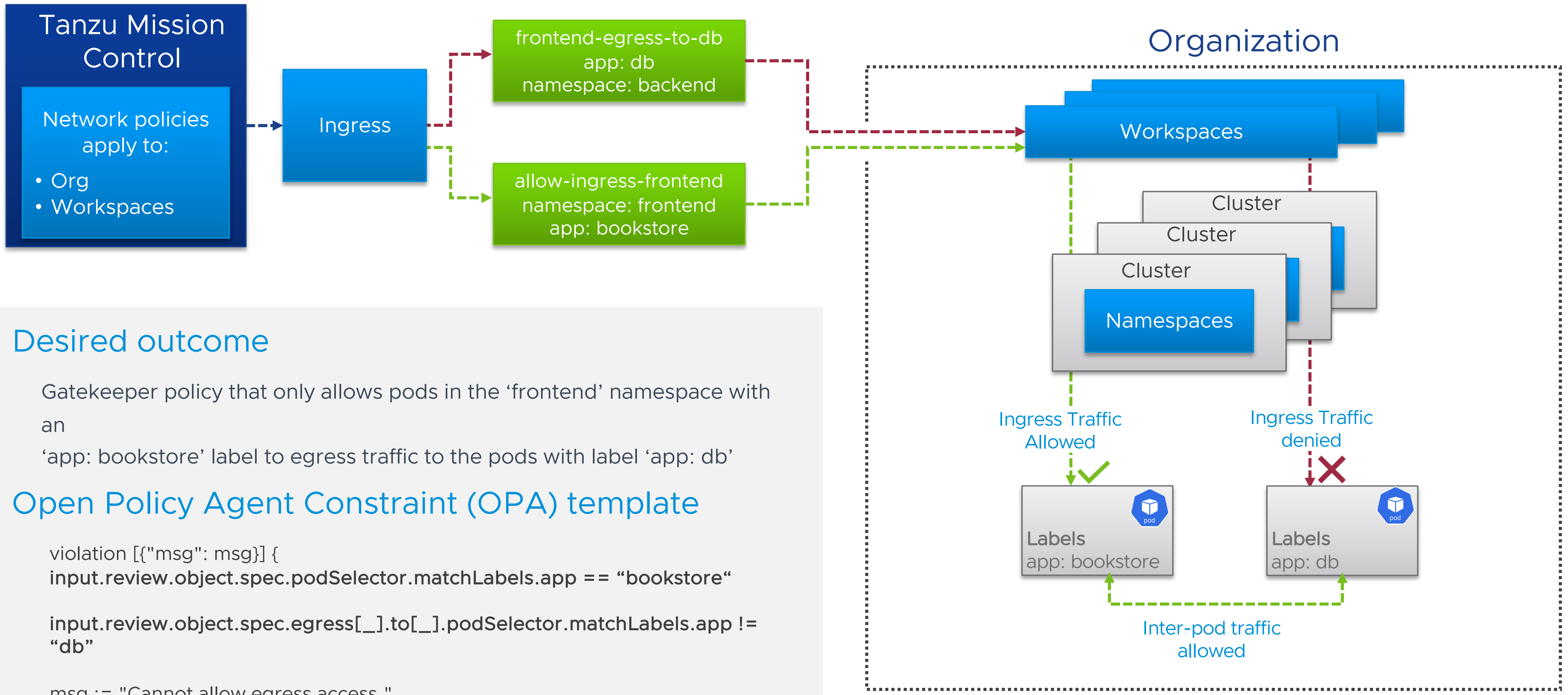
With network policies



Included network policies

- deny-all
- allow-all
- deny-all-for-pod
- allow-all-for-pod
- custom-ingress
- custom-egress

Example of OPA Constraint for a Two-Tier Web App



Desired outcome

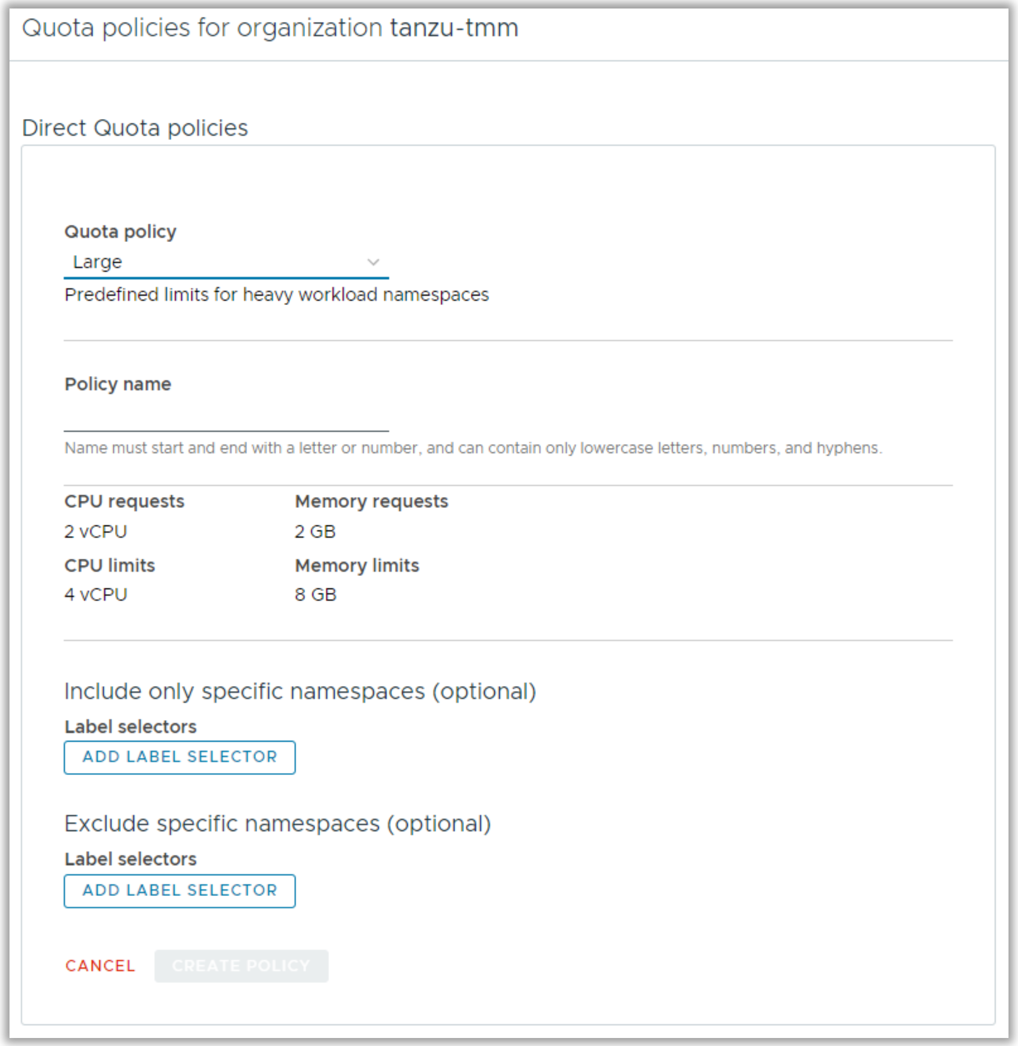
Gatekeeper policy that only allows pods in the 'frontend' namespace with an
'app: bookstore' label to egress traffic to the pods with label 'app: db'

Open Policy Agent Constraint (OPA) template

```
violation [{"msg": msg}] {
input.review.object.spec.podSelector.matchLabels.app == "bookstore"

input.review.object.spec.egress[_].to[_].podSelector.matchLabels.app !=
"db"

msg := "Cannot allow egress access."
```

Control CPU and Memory Quotas for Your Organization

Using Namespace Quota policies



Quota policies for organization tanzu-tmm

Direct Quota policies

Quota policy
Large
Predefined limits for heavy workload namespaces

Policy name

Name must start and end with a letter or number, and can contain only lowercase letters, numbers, and hyphens.

CPU requests	**Memory requests**
2 vCPU	2 GB
CPU limits	**Memory limits**
4 vCPU	8 GB

Include only specific namespaces (optional)

Label selectors

Exclude specific namespaces (optional)

Label selectors

Out-of-the-box templates

Custom quotas for custom needs

CPU and memory limits/requests

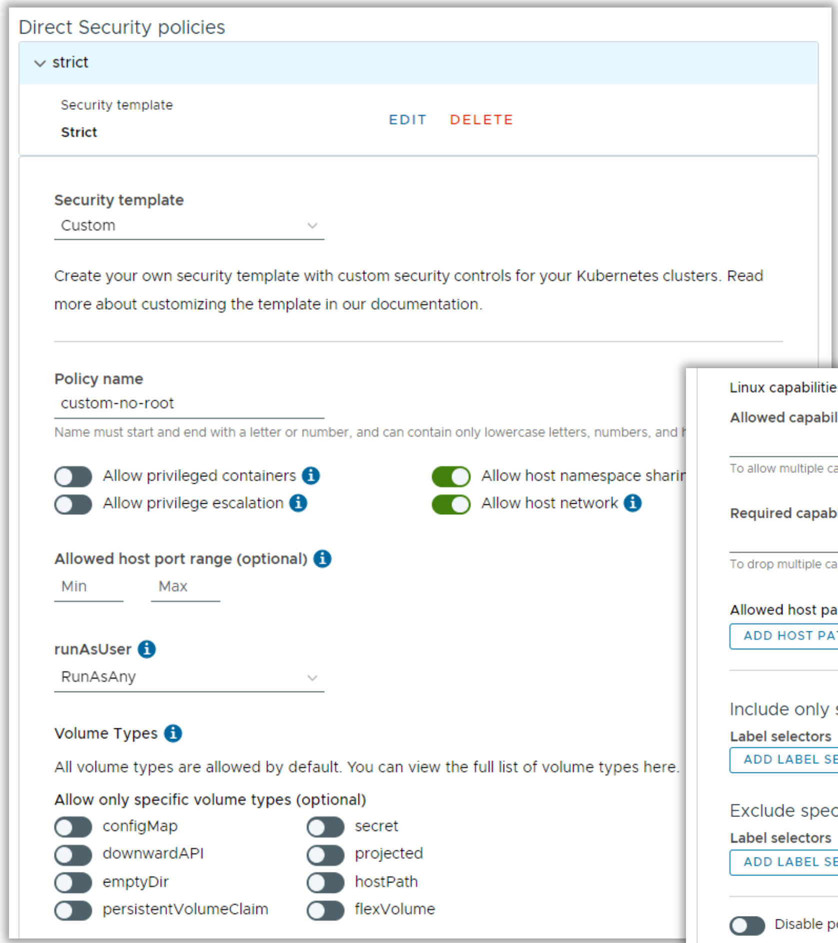
- Ratio-based policies with custom templates

Define and enforce once

Powered by OPA Gatekeeper

Security Policies That Meet Your Needs

From none to very strict


Direct Security policies

strict

Security template
Strict

EDIT DELETE

Security template
Custom

Create your own security template with custom security controls for your Kubernetes clusters. Read more about customizing the template in our documentation.

Policy name
custom-no-root
Name must start and end with a letter or number, and can contain only lowercase letters, numbers, and

Allow privileged containers
Allow privilege escalation
[✓ Allow host namespace sharir]
[✓ Allow host network]

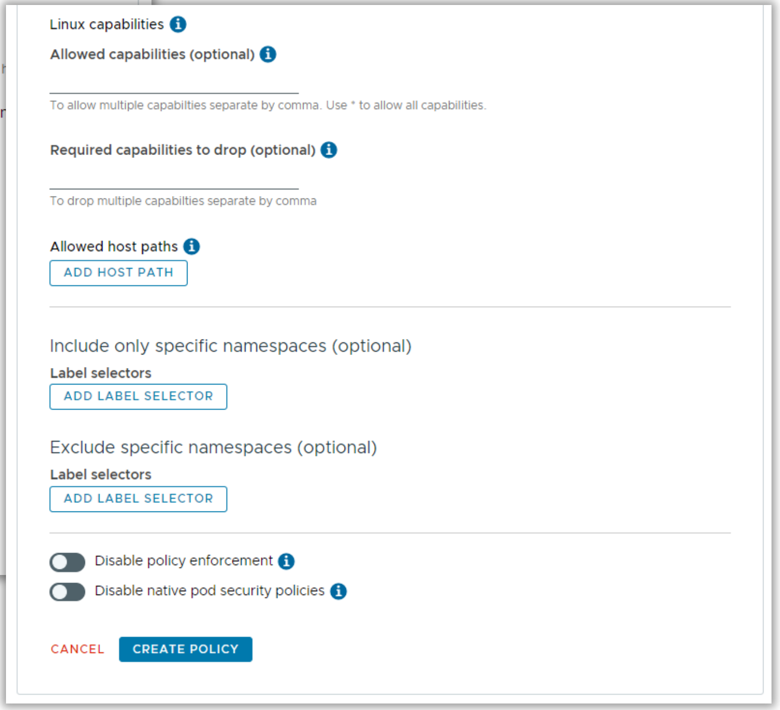
Allowed host port range (optional)
Min Max

runAsUser
RunAsAny

Volume Types
All volume types are allowed by default. You can view the full list of volume types here.

Allow only specific volume types (optional)

configMap	secret
downwardAPI	projected
emptyDir	hostPath
persistentVolumeClaim	flexVolume


Linux capabilities
Allowed capabilities (optional)
To allow multiple capabilties separate by comma. Use * to allow all capabilities.

Required capabilities to drop (optional)
To drop multiple capabilties separate by comma

Allowed host paths
ADD HOST PATH

Include only specific namespaces (optional)
Label selectors
ADD LABEL SELECTOR

Exclude specific namespaces (optional)
Label selectors
ADD LABEL SELECTOR

Disable policy enforcement
Disable native pod security policies

CANCEL CREATE POLICY

Baseline

Minimally restrictive policy while preventing known privilege escalations

Strict

Heavily restricted policy, following current pod hardening best practices

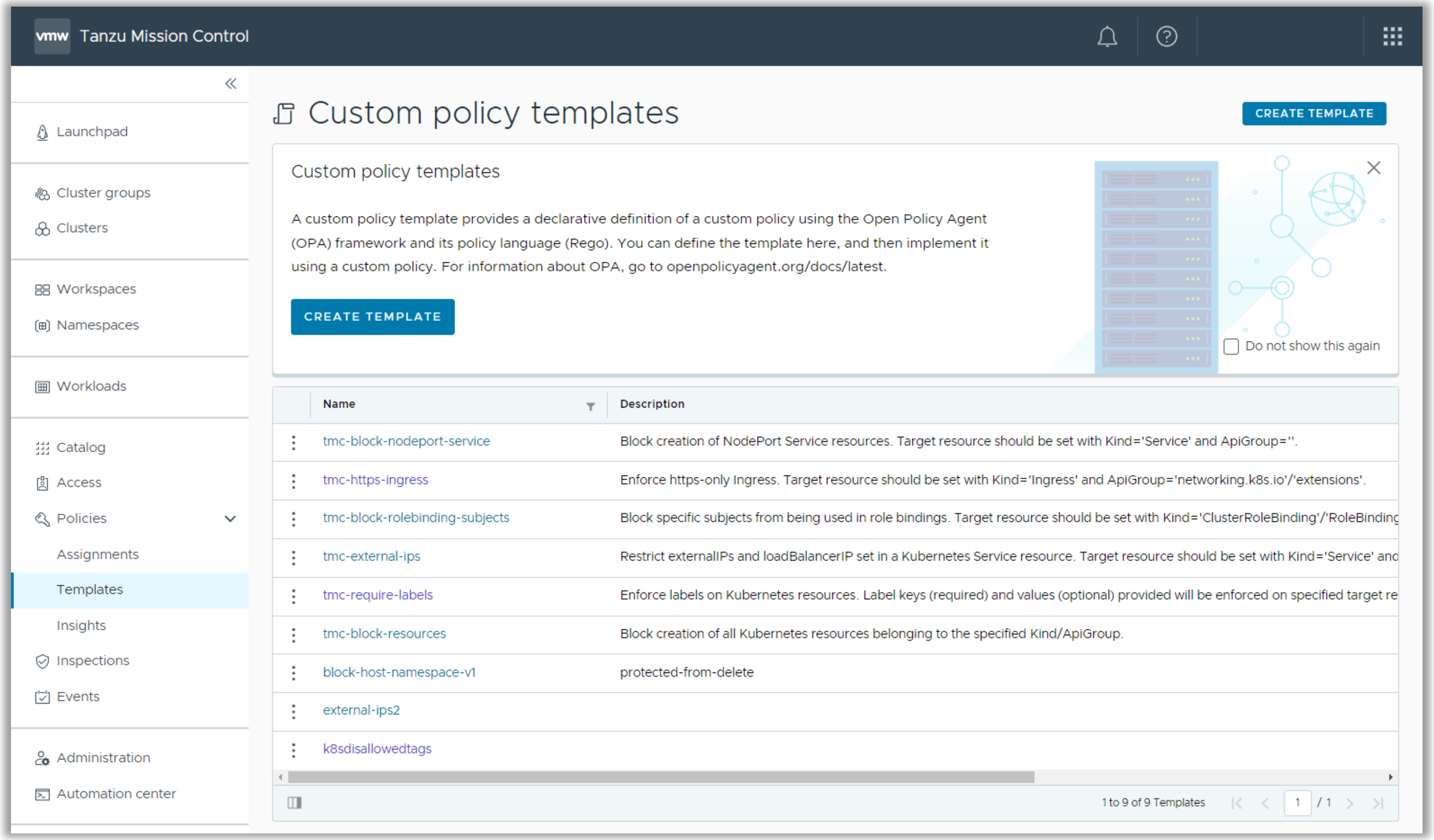
Custom

Create your own

Powered by OPA Gatekeeper

Enforce Policy-as-Code on Kubernetes Resources

Building your own custom policies



Custom policy templates

Custom policy templates

A custom policy template provides a declarative definition of a custom policy using the Open Policy Agent (OPA) framework and its policy language (Rego). You can define the template here, and then implement it using a custom policy. For information about OPA, go to openpolicyagent.org/docs/latest.

CREATE TEMPLATE

Do not show this again

Name	Description
tmc-block-nodeport-service	Block creation of NodePort Service resources. Target resource should be set with Kind='Service' and ApiGroup=''.
tmc-https-ingress	Enforce https-only Ingress. Target resource should be set with Kind='Ingress' and ApiGroup='networking.k8s.io'/'extensions'.
tmc-block-rolebinding-subjects	Block specific subjects from being used in role bindings. Target resource should be set with Kind='ClusterRoleBinding'/'RoleBinding
tmc-external-ips	Restrict externalIPs and loadBalancerIP set in a Kubernetes Service resource. Target resource should be set with Kind='Service' and
tmc-require-labels	Enforce labels on Kubernetes resources. Label keys (required) and values (optional) provided will be enforced on specified target re
tmc-block-resources	Block creation of all Kubernetes resources belonging to the specified Kind/ApiGroup.
block-host-namespace-v1	protected-from-delete
external-ips2	
k8sdisallowedtags	

1 to 9 of 9 Templates 1 / 1

Create your own policy template

Based on OPA constraints

Supports namespace selectors

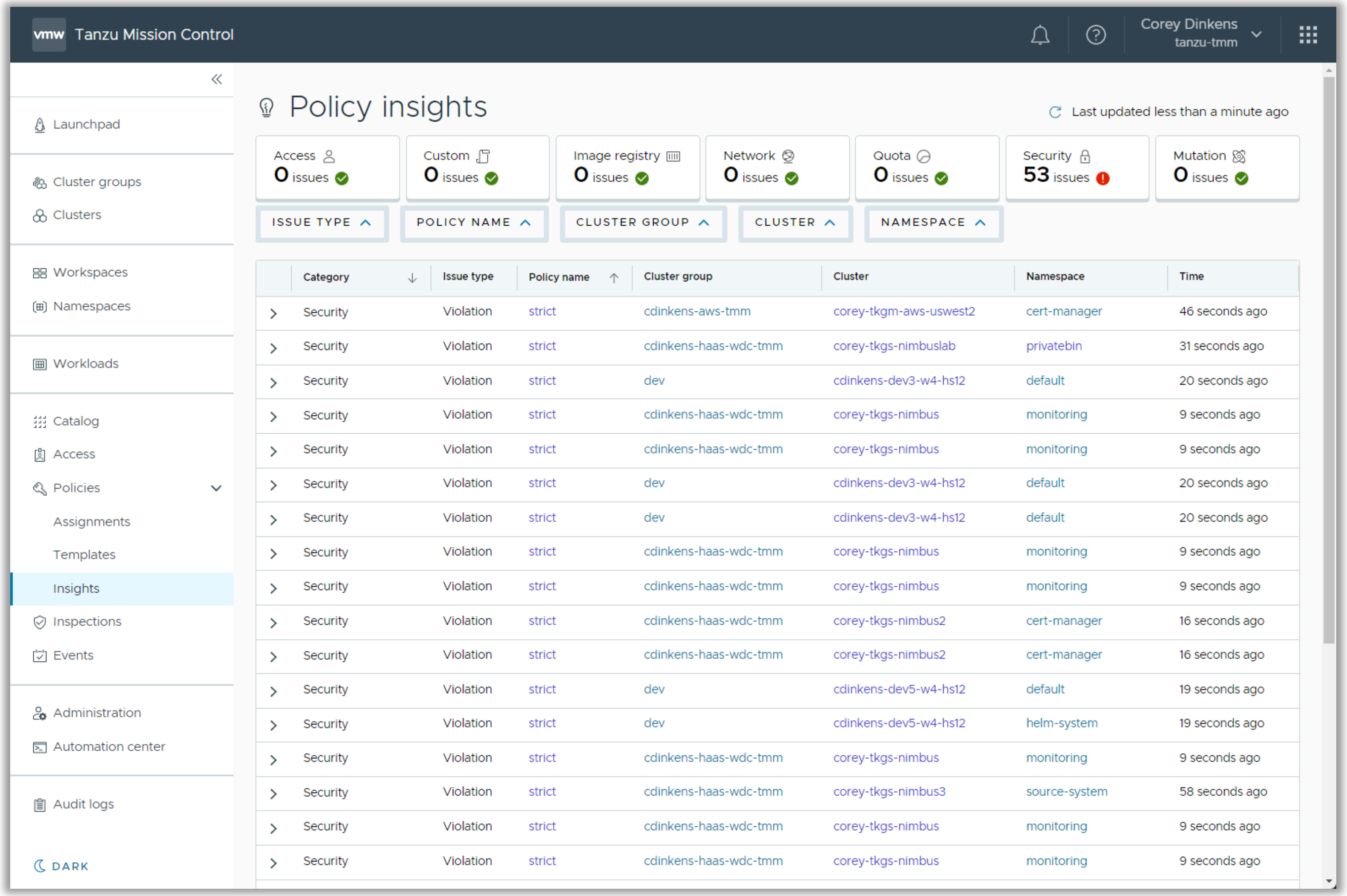
Comes with prebuilt templates

Future-proofed for deprecation of PSP

Powered by OPA Gatekeeper

View Detailed Policy Analysis in your Organization

With Policy insights



vmw Tanzu Mission Control

Corey Dinkens
tanzu-tmm

Launchpad
Cluster groups
Clusters
Workspaces
Namespaces
Workloads
Catalog
Access
Policies
Assignments
Templates
Insights
Inspections
Events
Administration
Automation center
Audit logs
DARK

Policy insights

Last updated less than a minute ago

Access	Custom	Image registry	Network	Quota	Security	Mutation
0 issues ✓	0 issues ✓	0 issues ✓	0 issues ✓	0 issues ✓	53 issues ❗	0 issues ✓

ISSUE TYPE · POLICY NAME · CLUSTER GROUP · CLUSTER · NAMESPACE

Category	Issue type	Policy name	Cluster group	Cluster	Namespace	Time
Security	Violation	strict	cdinkens-aws-tmm	corey-tkgm-aws-uswest2	cert-manager	46 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbuslab	privatebin	31 seconds ago
Security	Violation	strict	dev	cdinkens-dev3-w4-hs12	default	20 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago
Security	Violation	strict	dev	cdinkens-dev3-w4-hs12	default	20 seconds ago
Security	Violation	strict	dev	cdinkens-dev3-w4-hs12	default	20 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus2	cert-manager	16 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus2	cert-manager	16 seconds ago
Security	Violation	strict	dev	cdinkens-dev5-w4-hs12	default	19 seconds ago
Security	Violation	strict	dev	cdinkens-dev5-w4-hs12	helm-system	19 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus3	source-system	58 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago
Security	Violation	strict	cdinkens-haas-wdc-tmm	corey-tkgs-nimbus	monitoring	9 seconds ago

Provides overall policy status of your organization

Emits policy events

Provides aggregate and detailed views

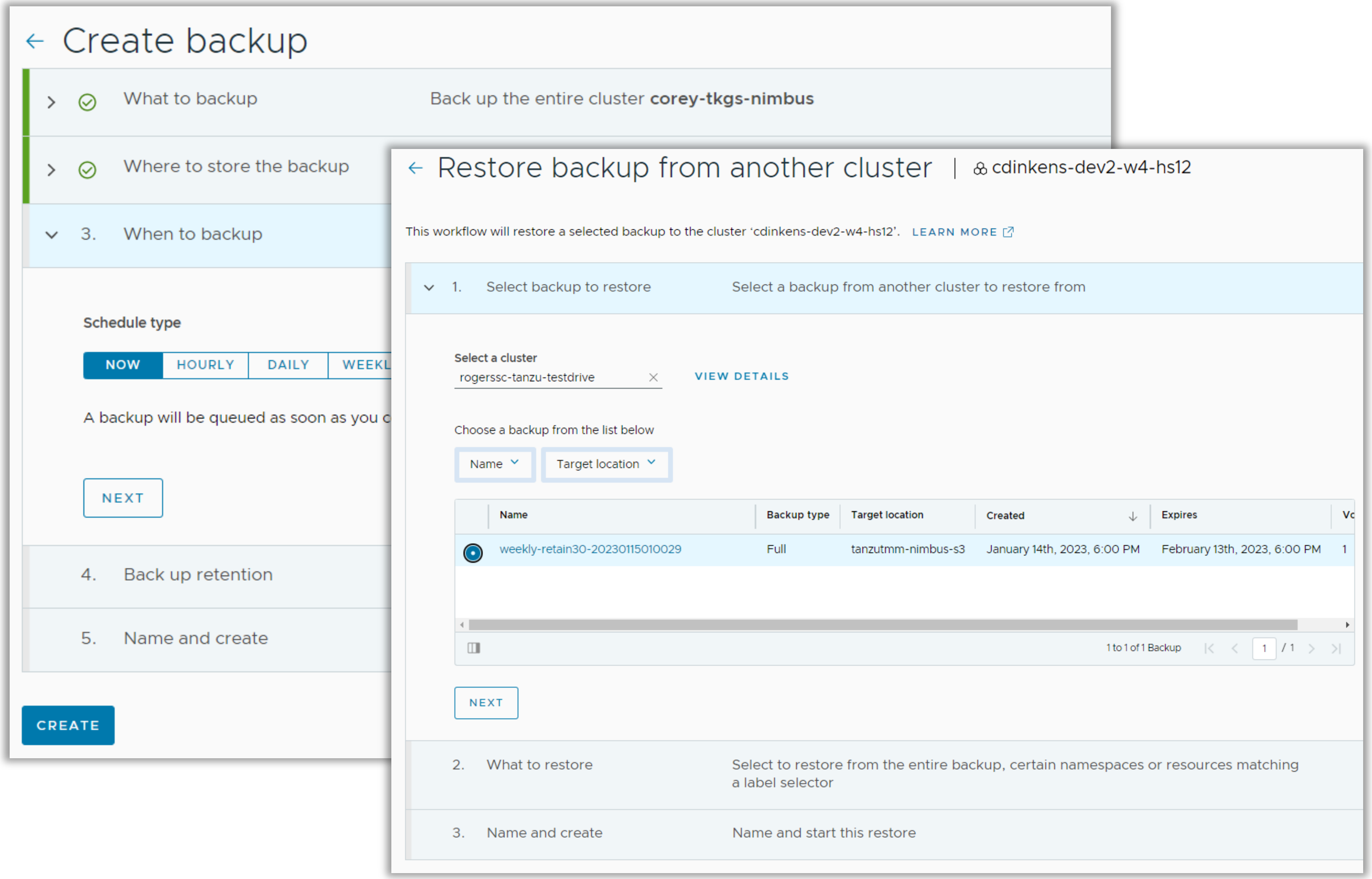
Policy debugging capabilities

One portal for viewing policy violations

Data Protection

Complete Data Protection

Powered by Velero



Create backup

✓	What to backup	Back up the entire cluster **corey-tkgs-nimbus**
✓	Where to store the backup	
3.	When to backup	

Schedule type

NOW | HOURLY | DAILY | WEEKL

A backup will be queued as soon as you c

NEXT

4. Back up retention
5. Name and create

CREATE

Restore backup from another cluster | cdinkens-dev2-w4-hs12

This workflow will restore a selected backup to the cluster 'cdinkens-dev2-w4-hs12'. LEARN MORE

1.	Select backup to restore	Select a backup from another cluster to restore from

Select a cluster: rogerssc-tanzu-testdrive — VIEW DETAILS

Choose a backup from the list below

Name | Target location

	Name	Backup type	Target location	Created	Expires	Vc
●	weekly-retain30-20230115010029	Full	tanzutmm-nimbus-s3	January 14th, 2023, 6:00 PM	February 13th, 2023, 6:00 PM	1

1 to 1 of 1 Backup — 1 / 1

NEXT

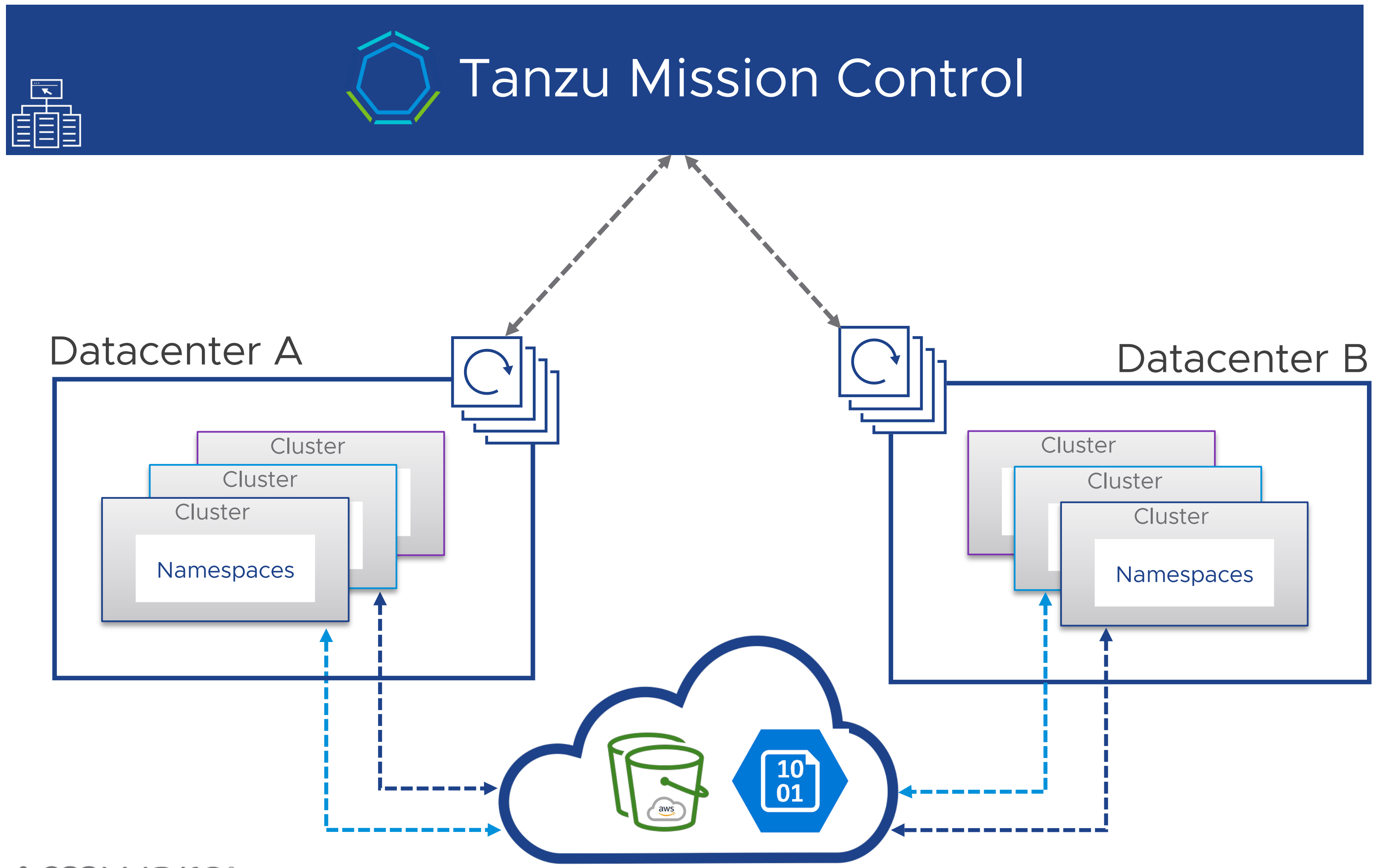
2.	What to restore	Select to restore from the entire backup, certain namespaces or resources matching a label selector
3.	Name and create	Name and start this restore

- Back up clusters, namespaces, or objects using labels
- Scheduled backups
- Back up to S3-compatible or Azure Blob targets
- Configure backup targets for cluster groups and clusters
- Cross-cluster restore
- Restic support for PV backups
- Custom CA support for storage targets

Powered by VELERO

Multi-Cloud Kubernetes data protection

With cross-cluster restoration



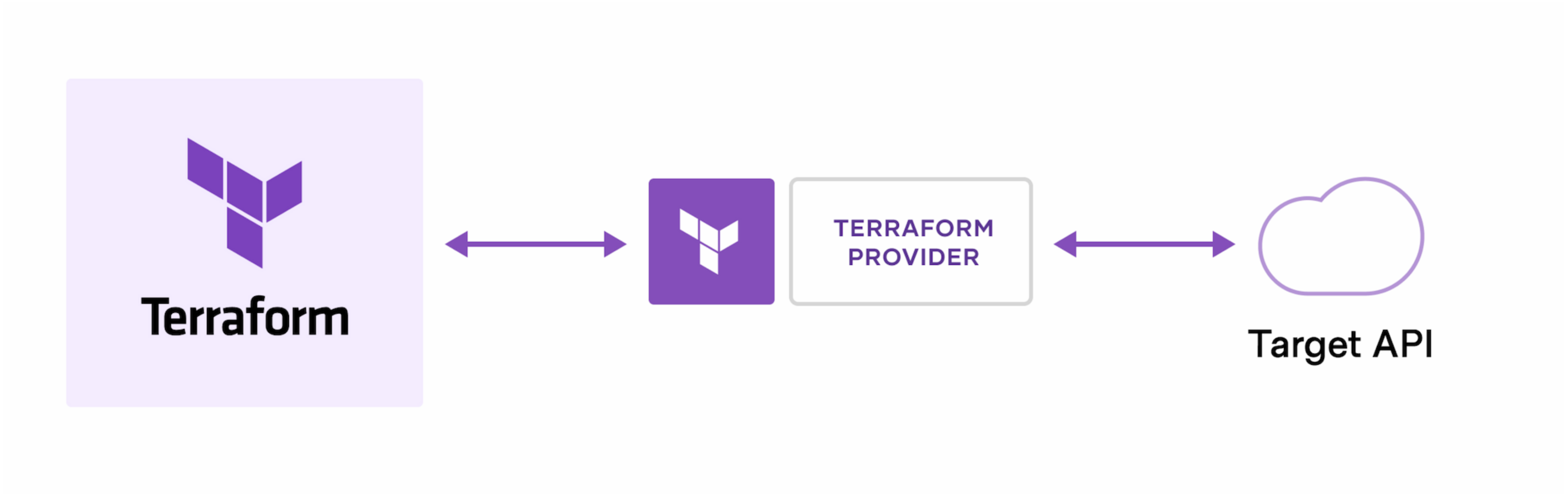
- Recover applications quickly
- Replicate production environments for staging or testing
- Backup target flexibility and choice
- Move applications between any CNCF-conformant cluster

Powered by VELERO

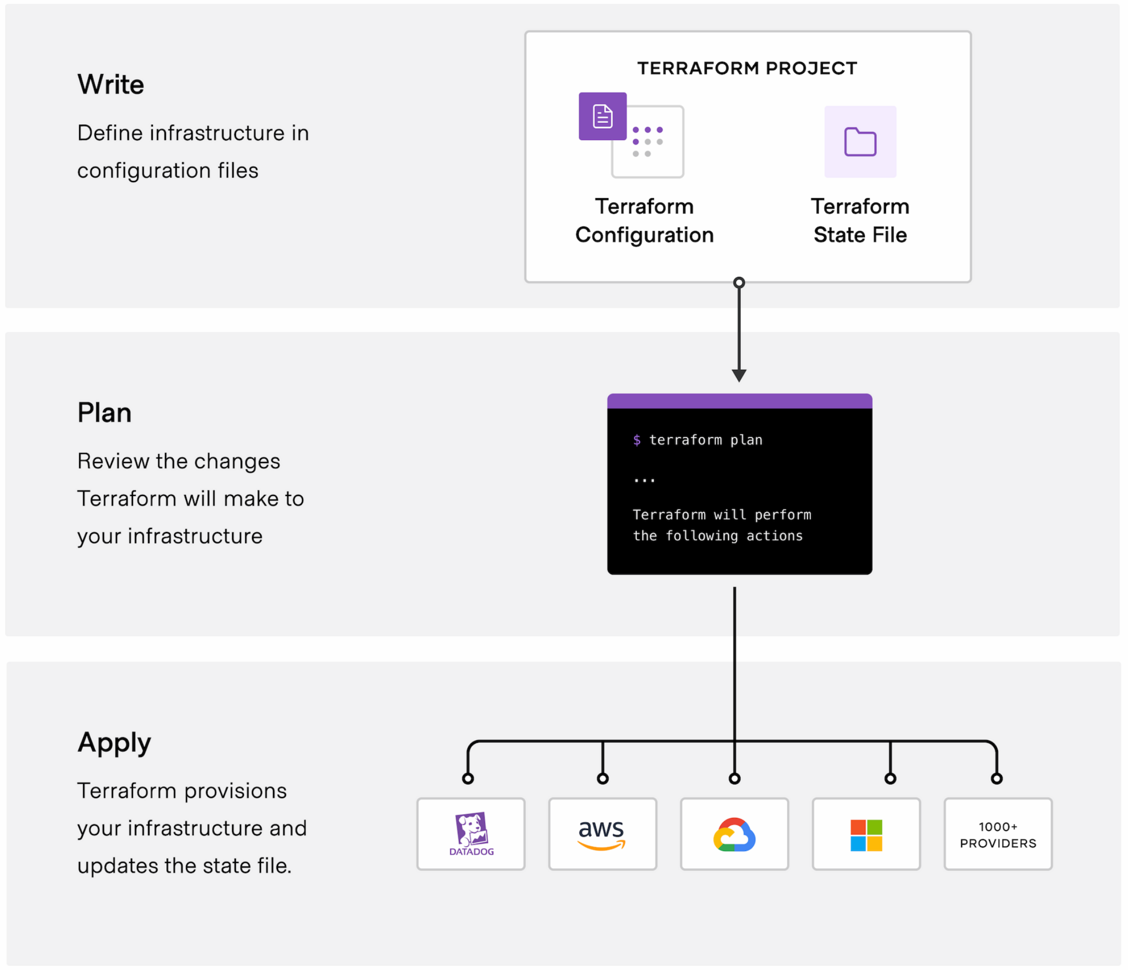
IAC with the Terraform Provider for TMC

What is Terraform?

“Terraform is an infrastructure as code (IAC) tool that lets you build, change and version cloud and on-premises resources safely and efficient.”



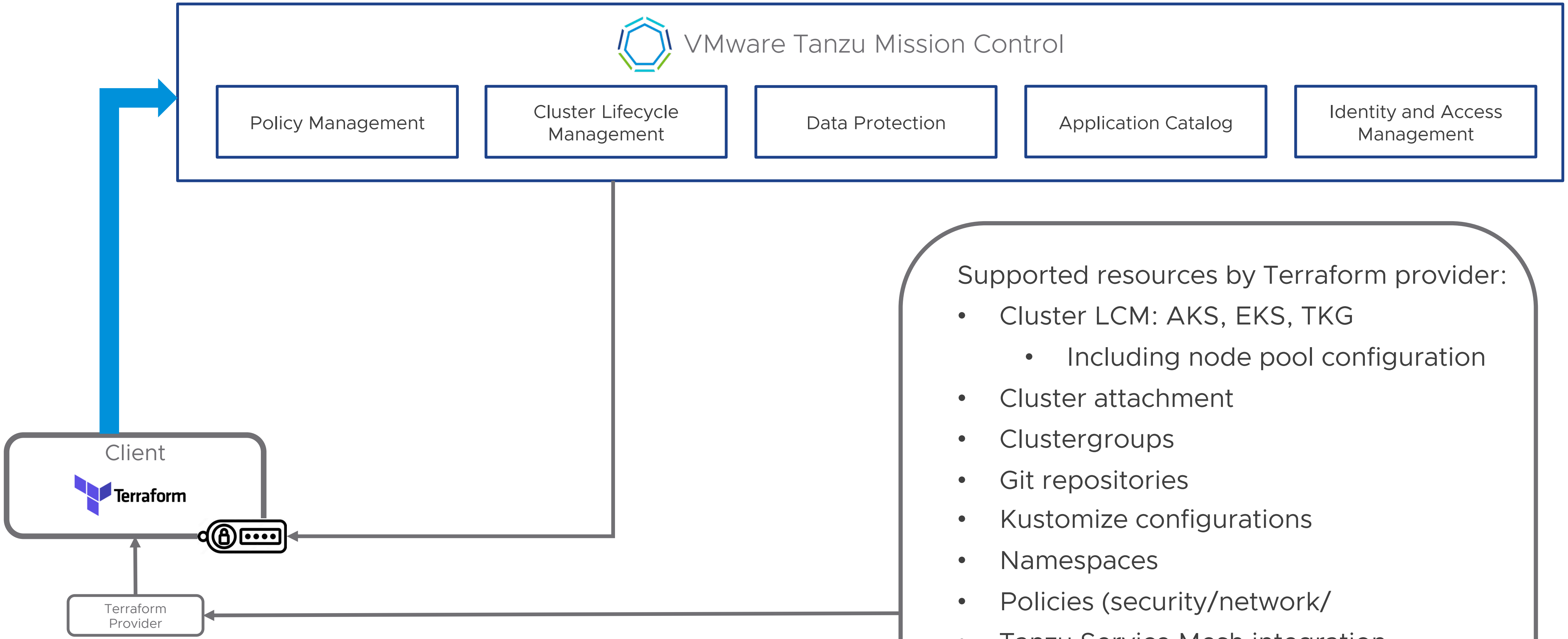
Terraform 101



Write
Define infrastructure in configuration files
TERRAFORM PROJECT
Terraform Configuration
Terraform State File
Plan
Review the changes Terraform will make to your infrastructure
Apply
Terraform provisions your infrastructure and updates the state file.
DATADOG
aws
1000+ PROVIDERS

```
$ terraform plan
...
Terraform will perform
the following actions
```

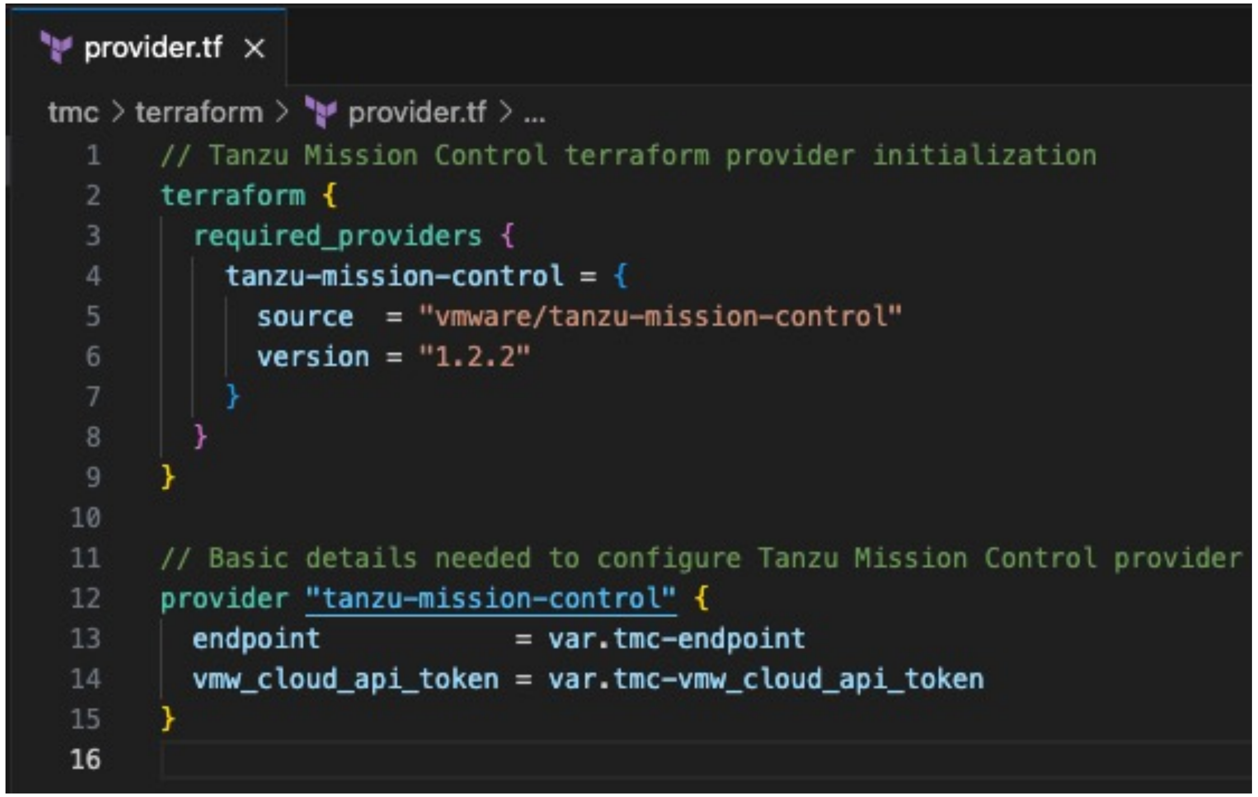
TMC + Terraform Basics


VMware Tanzu Mission Control
Policy Management
Cluster Lifecycle Management
Data Protection
Application Catalog
Identity and Access Management
Client
Terraform
Terraform Provider

Supported resources by Terraform provider:

- Cluster LCM: AKS, EKS, TKG
    - Including node pool configuration
- Cluster attachment
- Clustergroups
- Git repositories
- Kustomize configurations
- Namespaces
- Policies (security/network/
- Tanzu Service Mesh integration
- Workspaces

```
terraform init

terraform plan

terraform apply

terraform destroy
```

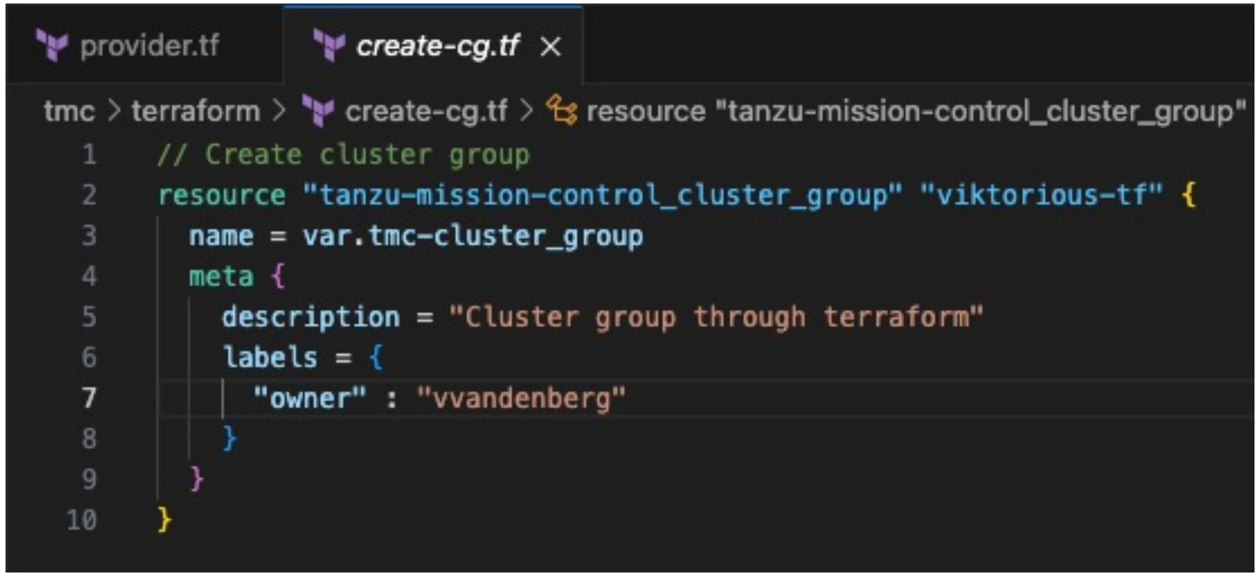


provider.tf ×

tmc > terraform > provider.tf > ...

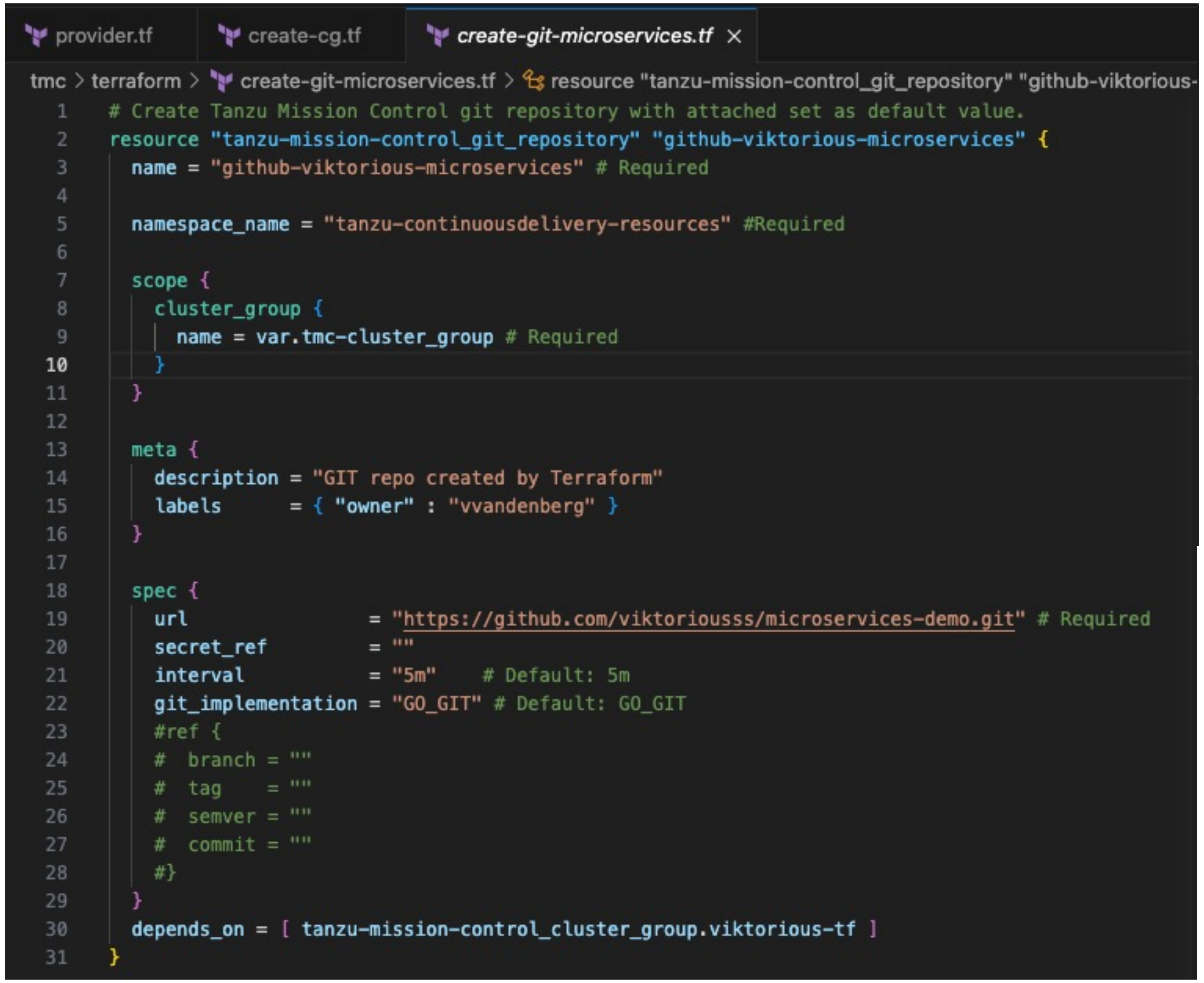
```
// Tanzu Mission Control terraform provider initialization
terraform {
  required_providers {
    tanzu-mission-control = {
      source  = "vmware/tanzu-mission-control"
      version = "1.2.2"
    }
  }
}

// Basic details needed to configure Tanzu Mission Control provider
provider "tanzu-mission-control" {
  endpoint            = var.tmc-endpoint
  vmw_cloud_api_token = var.tmc-vmw_cloud_api_token
}

```

```
terraform init

terraform plan

terraform apply

terraform destroy
```


provider.tf
create-cg.tf
tmc > terraform > create-cg.tf > resource "tanzu-mission-control_cluster_group"

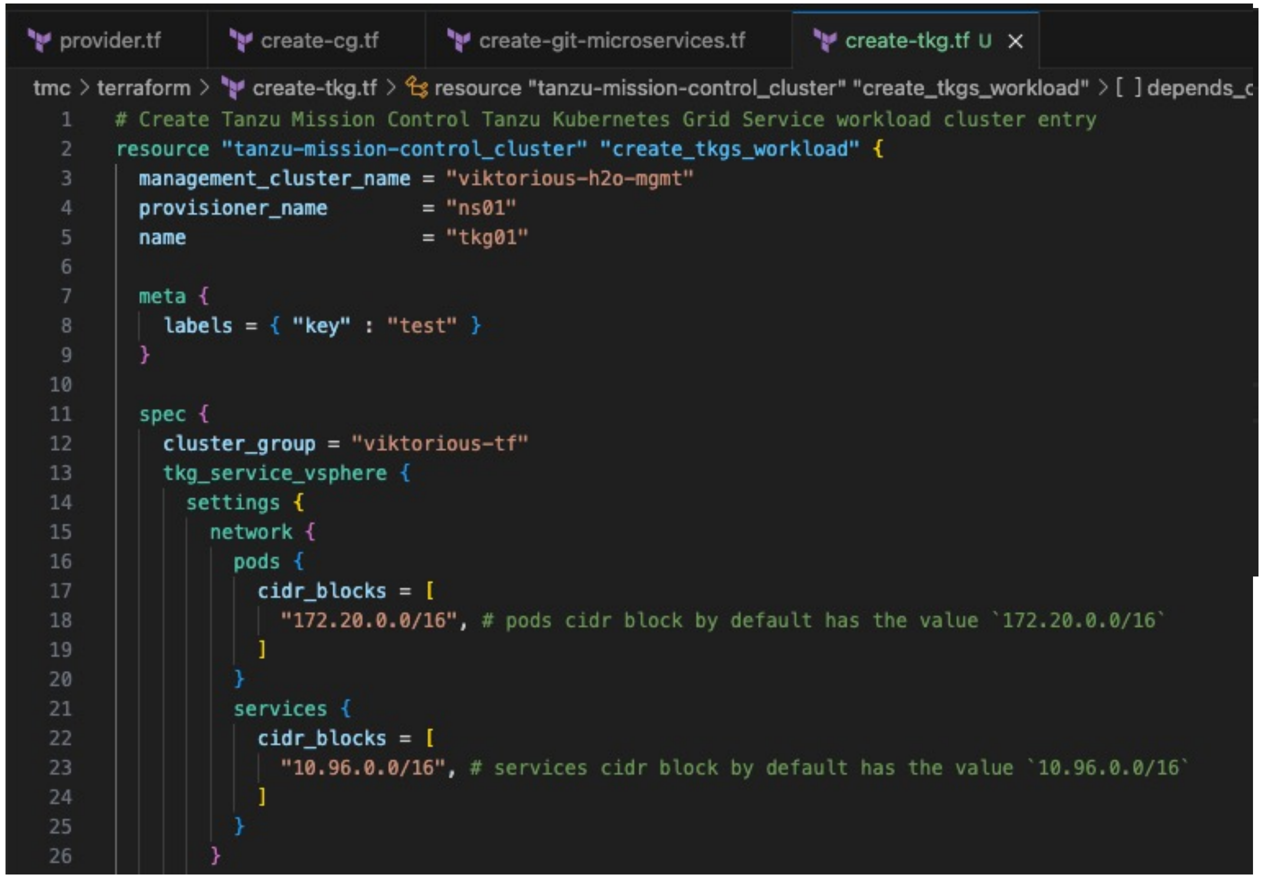
```
// Create cluster group
resource "tanzu-mission-control_cluster_group" "viktorious-tf" {
  name = var.tmc-cluster_group
  meta {
    description = "Cluster group through terraform"
    labels = {
      "owner" : "vvandenberg"
    }
  }
}
```

```
terraform init

terraform plan

terraform apply

terraform destroy
```


provider.tf
create-cg.tf
create-git-microservices.tf
tmc > terraform > create-git-microservices.tf > resource "tanzu-mission-control_git_repository" "github-viktorious-

```
# Create Tanzu Mission Control git repository with attached set as default value.
resource "tanzu-mission-control_git_repository" "github-viktorious-microservices" {
  name = "github-viktorious-microservices" # Required

  namespace_name = "tanzu-continuousdelivery-resources" #Required

  scope {
    cluster_group {
      name = var.tmc-cluster_group # Required
    }
  }

  meta {
    description = "GIT repo created by Terraform"
    labels      = { "owner" : "vvandenberg" }
  }

  spec {
    url                = "https://github.com/viktoriousss/microservices-demo.git" # Required
    secret_ref         = ""
    interval           = "5m"    # Default: 5m
    git_implementation = "GO_GIT" # Default: GO_GIT
    #ref {
    #  branch = ""
    #  tag    = ""
    #  semver = ""
    #  commit = ""
    #}
  }
  depends_on = [ tanzu-mission-control_cluster_group.viktorious-tf ]
}
```

```
terraform init

terraform plan

terraform apply

terraform destroy
```


provider.tf
create-cg.tf
create-git-microservices.tf
create-tkg.tf U
tmc > terraform > create-tkg.tf > resource "tanzu-mission-control_cluster" "create_tkgs_workload" > [ ] depends_c

```
# Create Tanzu Mission Control Tanzu Kubernetes Grid Service workload cluster entry
resource "tanzu-mission-control_cluster" "create_tkgs_workload" {
  management_cluster_name = "viktorious-h2o-mgmt"
  provisioner_name        = "ns01"
  name                    = "tkg01"

  meta {
    labels = { "key" : "test" }
  }

  spec {
    cluster_group = "viktorious-tf"
    tkg_service_vsphere {
      settings {
        network {
          pods {
            cidr_blocks = [
              "172.20.0.0/16", # pods cidr block by default has the value `172.20.0.0/16`
            ]
          }
          services {
            cidr_blocks = [
              "10.96.0.0/16", # services cidr block by default has the value `10.96.0.0/16`
            ]
          }
        }
```

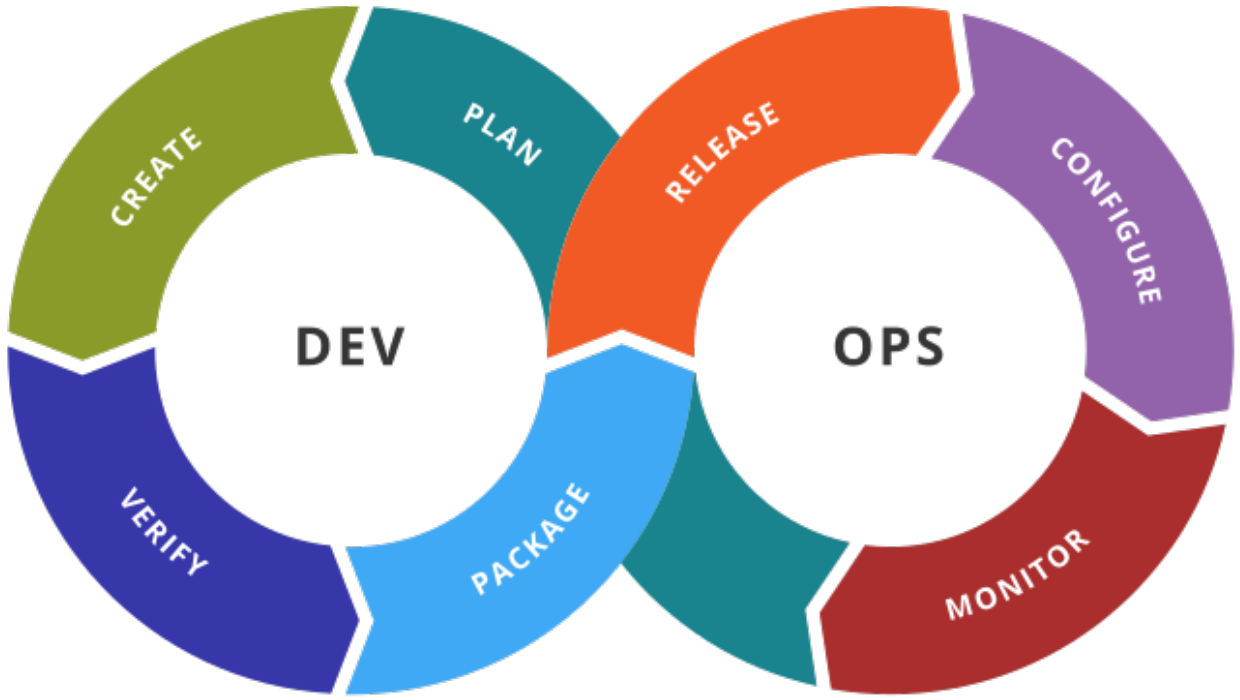
Continuous Deliver with TMC

GitOps

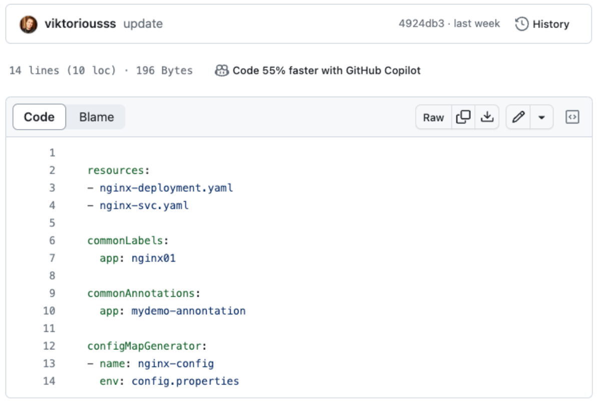
GitOps evolved from DevOps. The specific state of deployment configuration is version-controlled. Because the most popular version-control is Git, GitOps' approach has been named after Git.

Changes to configuration can be managed using code review practices, and can be rolled back using version-controlling. Essentially, all of the changes to a code are tracked, bookmarked, and making any updates to the history can be made easier

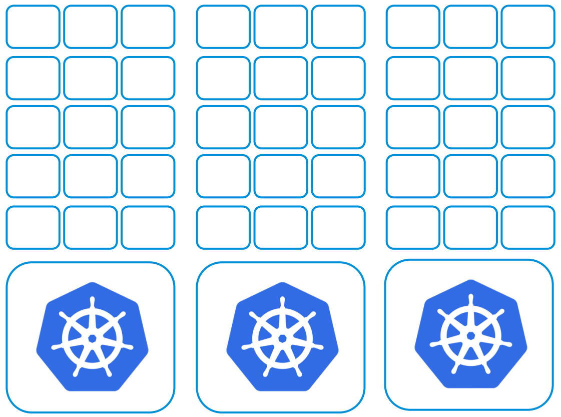
Our focus today is on GitOps



Basic GitOps


viktoriousss update
4924db3 · last week
History
14 lines (10 loc) · 196 Bytes
Code 55% faster with GitHub Copilot
Code
Blame
Raw

```

resources:
- nginx-deployment.yaml
- nginx-svc.yaml

commonLabels:
  app: nginx01

commonAnnotations:
  app: mydemo-annontation

configMapGenerator:
- name: nginx-config
  env: config.properties
```



Tooling that can help you doing the magic

What is Kustomize?

Provides a declarative and layered approach for Kubernetes cluster configuration and deployment + configuration of apps running on Kubernetes.

Natively built into kubectl.

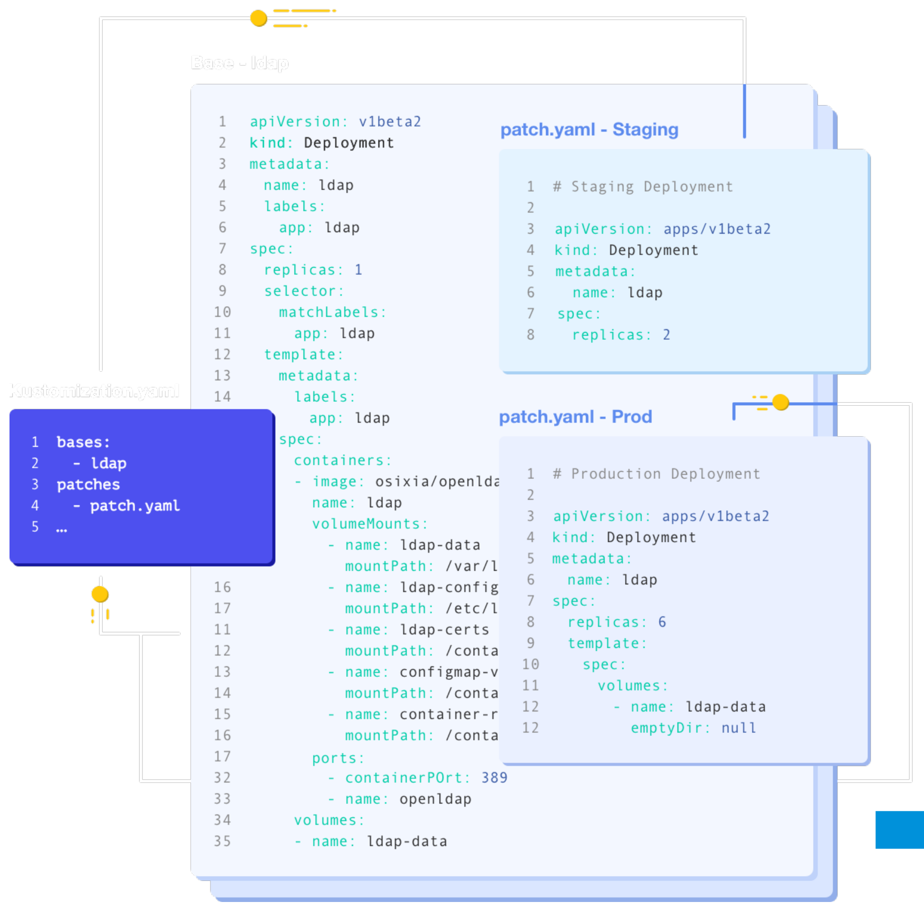
What is Helm?

Helm is a package manager that automates the creation, packaging and configuration and deployment of Kubernetes applications.

What is Flux?

Flux is a tool for keeping Kubernetes clusters in sync with sources of configuration (like Git repos) and automates updates to configuration when there is new code to deploy.

Basics of Kustomize



```yaml
bases:
  - ldap
patches
  - patch.yaml
…
```

```yaml
apiVersion: v1beta2
kind: Deployment
metadata:
  name: ldap
  labels:
    app: ldap
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ldap
  template:
    metadata:
      labels:
        app: ldap
      spec:
        containers:
        - image: osixia/openlda
          name: ldap
          volumeMounts:
            - name: ldap-data
              mountPath: /var/l
            - name: ldap-config
              mountPath: /etc/l
            - name: ldap-certs
              mountPath: /conta
            - name: configmap-v
              mountPath: /conta
            - name: container-r
              mountPath: /conta
          ports:
            - containerPOrt: 389
            - name: openldap
        volumes:
        - name: ldap-data
```

```yaml
# Staging Deployment

apiVersion: apps/v1beta2
kind: Deployment
metadata:
   name: ldap
spec:
   replicas: 2
```

```yaml
# Production Deployment

apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: ldap
spec:
  replicas: 6
  template:
    spec:
      volumes:
        - name: ldap-data
          emptyDir: null
```

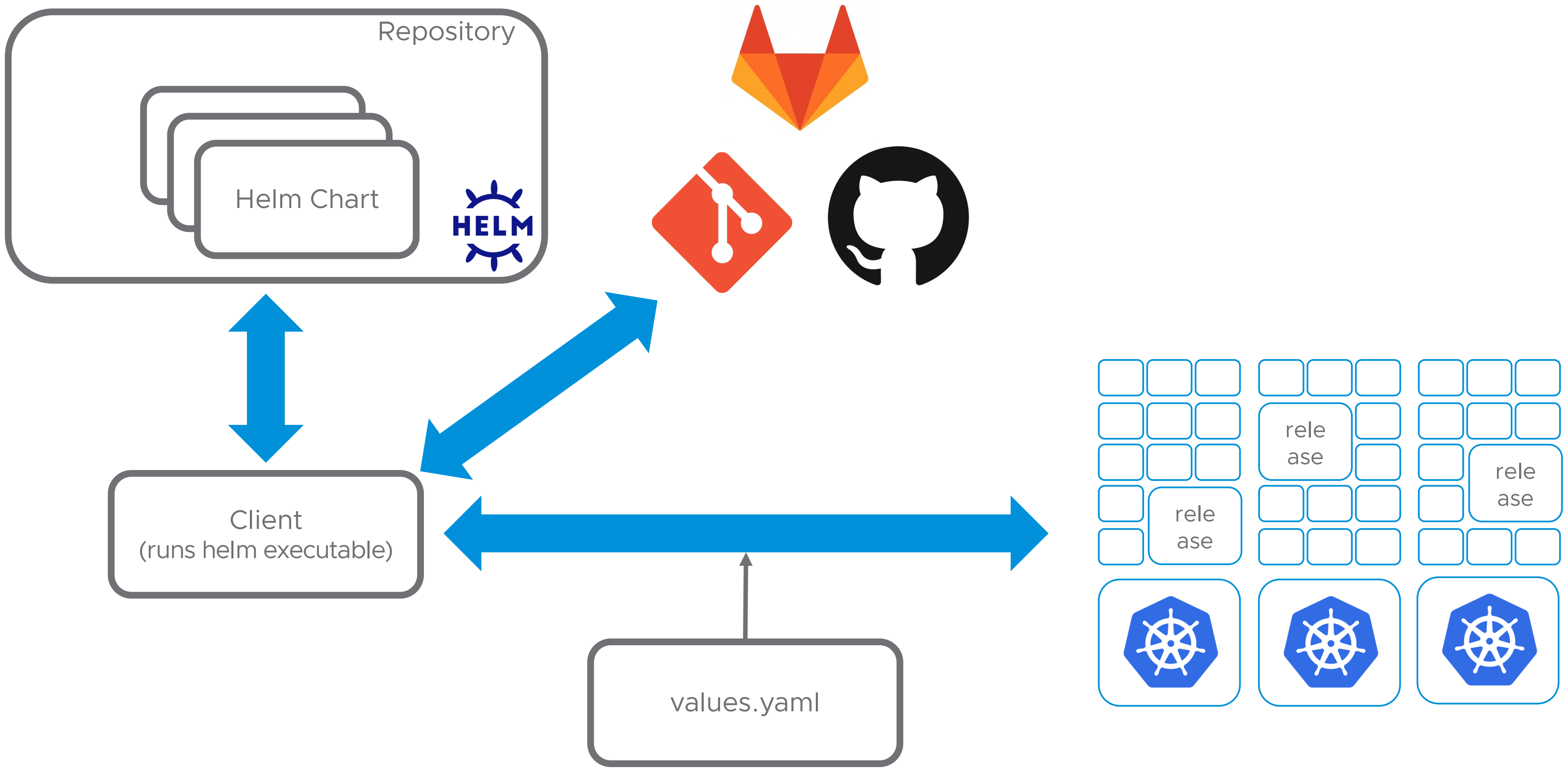
Overlay: staging

Overlay: prod

Base

```
kubectl -k ./overlay/prod
```

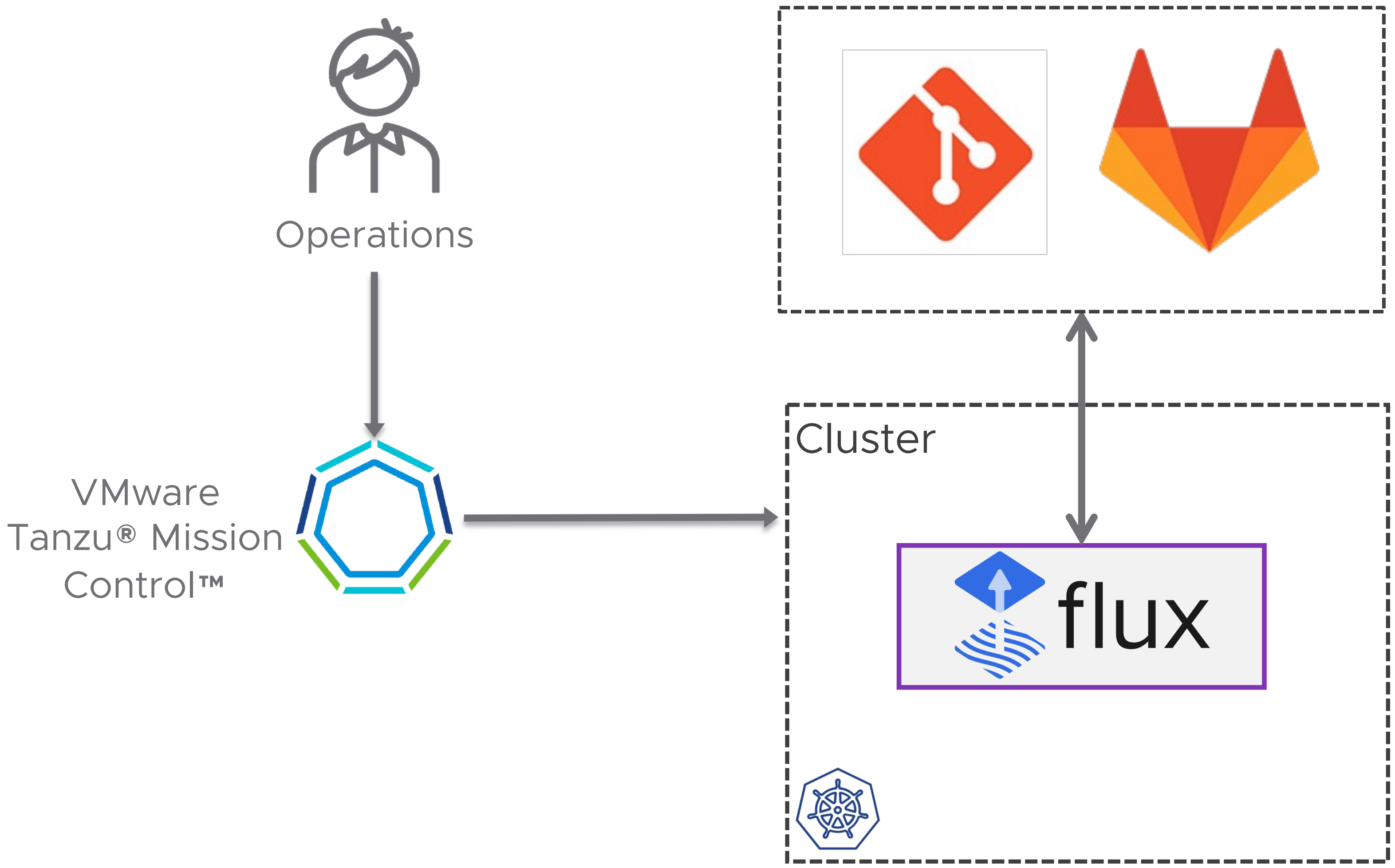
Basics of Helm



```
helm install bitnami/rabbitmq -name rbq-prod -namespace -brq -f values.yaml
```

Continuous Delivery with Tanzu Mission Control

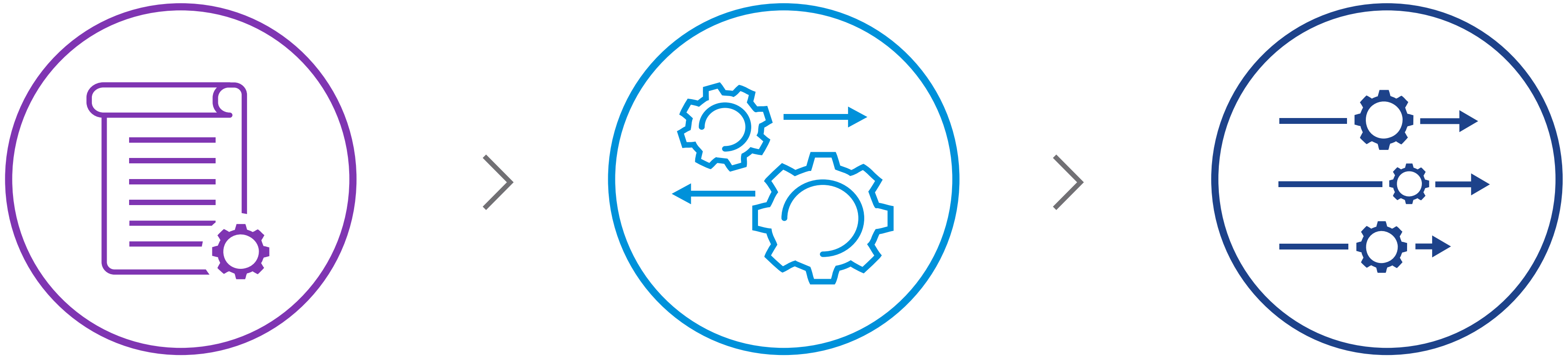
Manage Cluster Configuration and Helm deployments with Flux CD



- Define resources once as YAML and reuse across your cluster fleet
- Improves DevOps cluster handoffs
- Store credentials that only propagate when required by a repository
- Manage Helm deployments on clusters with public and private repositories

Continuous Delivery in Tanzu Mission Control

Sync YAML artifacts to your cluster(s) and cluster groups



Define cluster settings

Describe your cluster config as YAML and store in a git repo

Add to Tanzu Mission Control

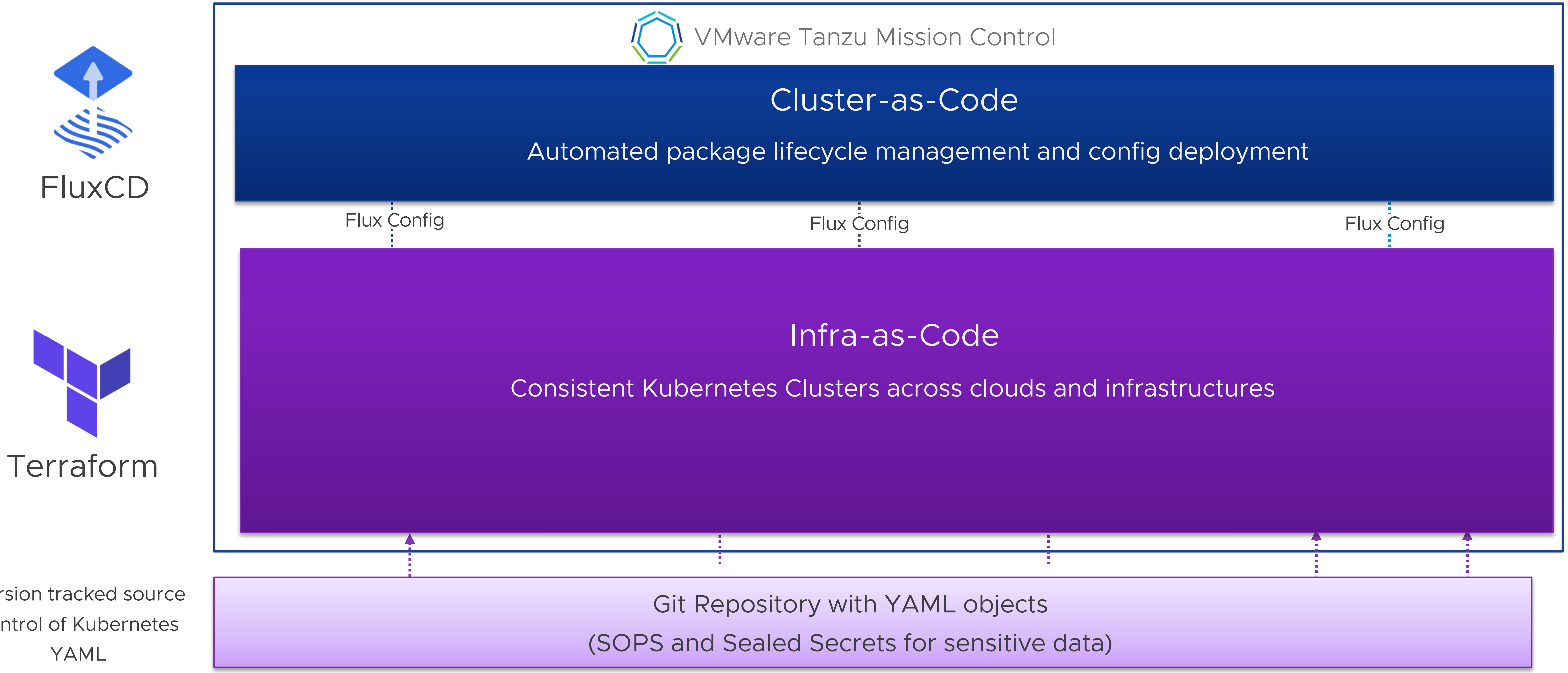
Attach the repo to Tanzu Mission Control cluster or cluster group and point it to the path containing YAML configs

Flux Controller syncs YAML

Syncs your YAML to the cluster and creates objects described in your YAML

Automated Cluster Operations with Continuous Delivery

Consistent configurations and deployments enforced across clouds and cluster types



FluxCD
Terraform
VMware Tanzu Mission Control
Cluster-as-Code
Automated package lifecycle management and config deployment
Flux Config
Flux Config
Flux Config
Infra-as-Code
Consistent Kubernetes Clusters across clouds and infrastructures
Version tracked source control of Kubernetes YAML
Git Repository with YAML objects
(SOPS and Sealed Secrets for sensitive data)

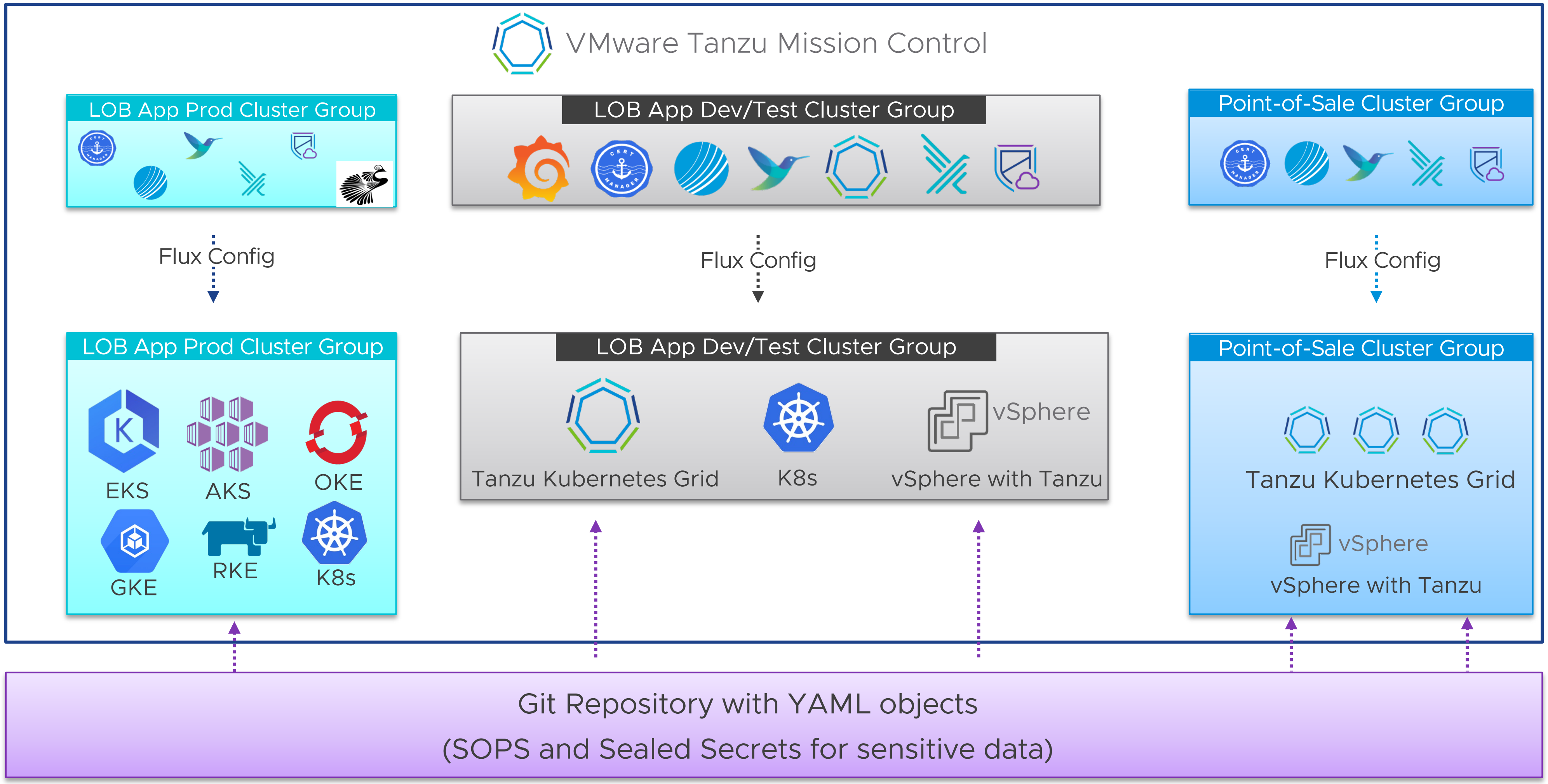
Automated Cluster Operations with Continuous Delivery

Consistent configurations and deployments enforced across clouds and cluster types

Automated package lifecycle management and config deployment

Consistent Kubernetes Clusters across clouds and infrastructures

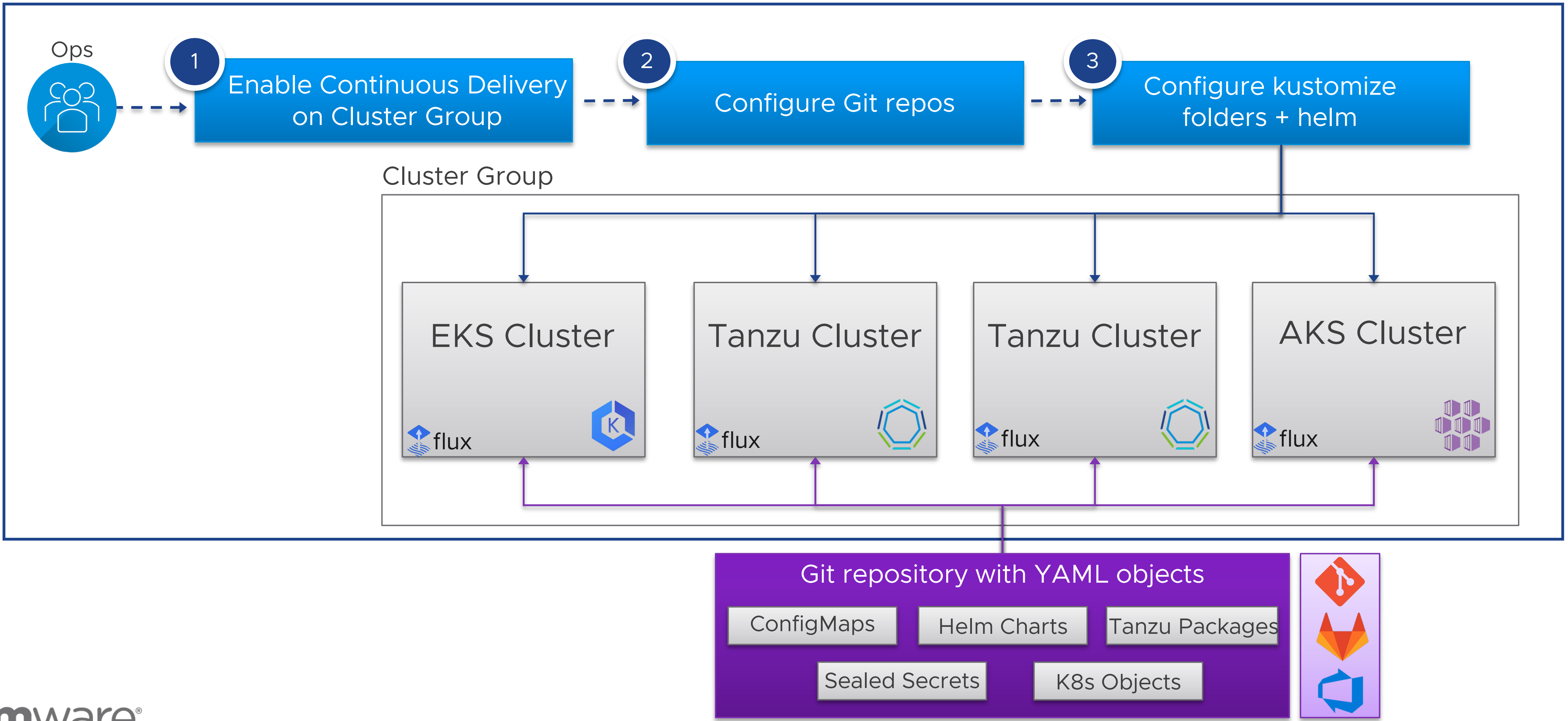
Version tracked source control of Kubernetes YAML


VMware Tanzu Mission Control
LOB App Prod Cluster Group
LOB App Dev/Test Cluster Group
Point-of-Sale Cluster Group
Flux Config
Flux Config
Flux Config
LOB App Prod Cluster Group
EKS
AKS
OKE
GKE
RKE
K8s
LOB App Dev/Test Cluster Group
Tanzu Kubernetes Grid
K8s
vSphere
vSphere with Tanzu
Point-of-Sale Cluster Group
Tanzu Kubernetes Grid
vSphere
vSphere with Tanzu
Git Repository with YAML objects
(SOPS and Sealed Secrets for sensitive data)

FluxCD Group Configuration

Three simple steps

VMware Tanzu® Mission Control™



Next times Agenda

Discover the magic of data services in Kubernetes with Mattias Soderberg

- Data services in K8s, the new abstraction of storage!
- Tanzu Data Solutions
- Demo of Tanzu and Postgres
- Demo of Tanzu and RabbitMQ

Until next time & Q&A

Look at the following

Viktor's Blog : https://www.viktorious.nl

Viktor's Github

- Terraform example: https://github.com/viktoriousss/tanzu-demo-essentials/tree/main/tmc/terraform
- Kustomize example: https://github.com/viktoriousss/tanzu-demo-essentials/tree/main/kustomize

Kubernetes : https://kubernetes.io

CNCF : https://www.cncf.io

Tanzu : https://tanzu.vmware.com

Register for next event on https://webinars.tanzu.dk

Recording / Slides will also be available there.

Robert Jensen

Lead Systems Engineer @Vmware

@rhjensen / jensenr@vmware.com

Viktor Van Der Berg

Lead Systems Engineer @Vmware

@viktoriousss / vvandenberg@vmware.com