Microsoft Azure intel CLOUD NATIVE COMPUTING FOUNDATION

探索 GitOps 平台的更多可能

郭旭东

极狐(GitLab)资深云原生架构师

2023.04.08

目录

CATALOGUE

ABOUT ME

郭旭东 极狐(GitLab)资深云原生架构师，Linux Foundation APAC 开源布道者、Education SIG Chair，NextArch Foundation TOC 成员，CCF 开源发展委员会执委委员，开放原子基金会开源大使，阿里云 MVP，云原生社区管理委员会成员，是 CNCF 项目 KubeVela / ChaosBlade 的 Maintainer。

Chapter I

什么是 GitOps 平台

GitOps 的定义与原则

GitOps = XaC + MR or PR + CI/CD

原则

整个系统以声明的方式描述

#1. The entire system described declaratively.

需要的系统状态在 Git 中进行版本管理

#2. The canonical desired system state versioned in Git.

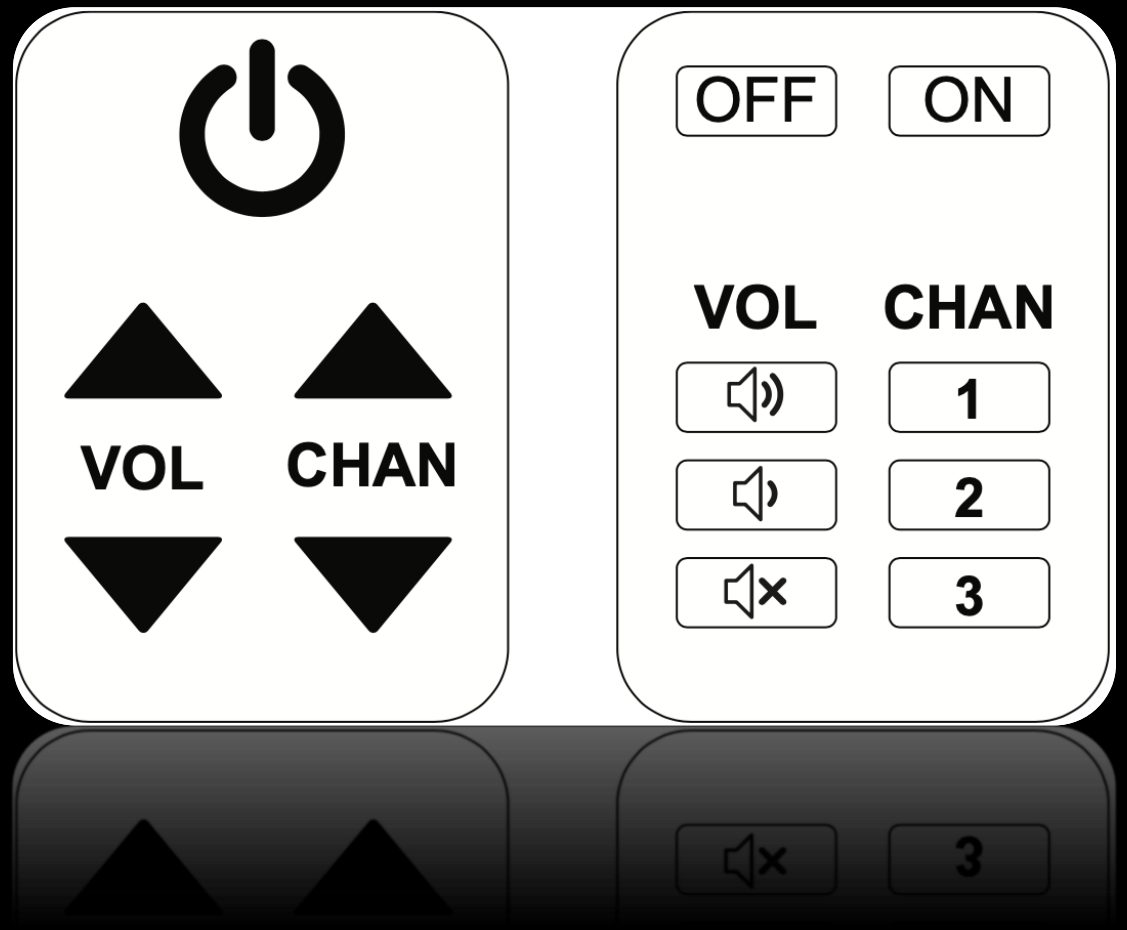
可自动应用于系统的已批准更改

#3. Approved changes that can be automatically applied to the system.

软件代理来确保差异的修正和告警

#4. Software agents to ensure correctness and alert on divergence.

为什么声明式更合适?



声明式描述目标的性质，让计算机明白目标，而非流程。它不用告诉计算机问题领域，从而避免随之而来的副作用。

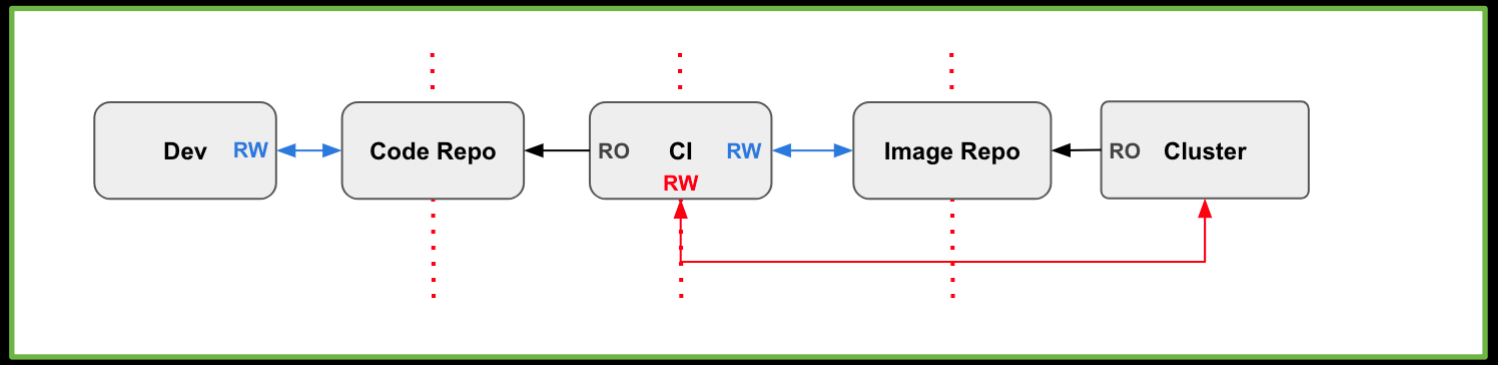
幂等性就是用户对于同一操作发起的一次请求或者多次请求的结果是一致的，不会因为多次点击而产生了副作用。

为什么一定是 GitOps?

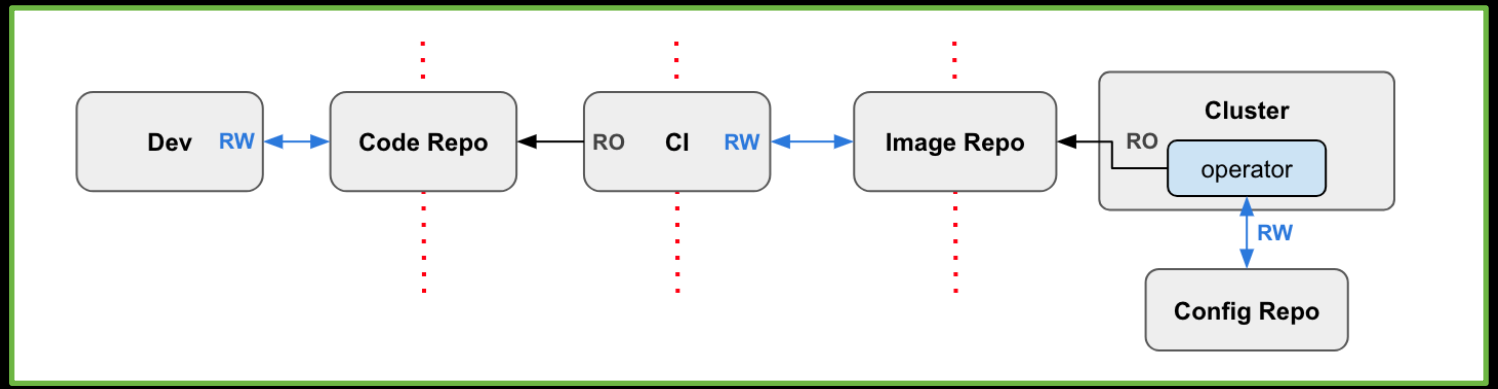
- 版本管理
- 分支策略
- 代码审查
- 历史记录
- 用户体验

Push or Pull

Push Based Deployment



Pull Based Deployment



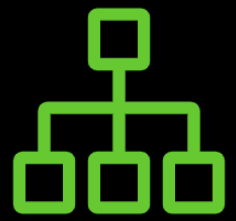
什么是 GitOps 平台?

GitOps 平台是一种将 GitOps 工具与其他工具和服务集成在一起的平台，旨在提供更完整的 DevOps 自动化解决方案。

Chapter II

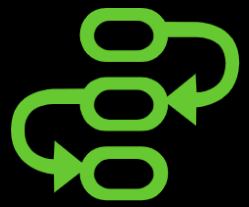
构建高质量的 GitOps 平台

GitOps 平台通常提供以下能力



GitOps 工具管理

将不同的 GitOps工具集成在一起，可以自动管理应用程序的部署、配置和更新。



自动化流水线

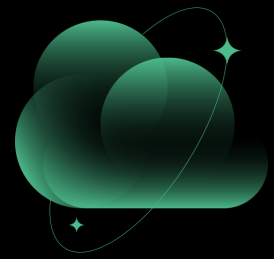
提供自动化管道，从代码提交到应用程序部署和监控全过程自动化。

代码审查

由于代码会直接影响到云等基础设置，每次修改都需要充分的 Review。

安全工具

提供安全性的功能，如自动化的漏洞扫描、访问控制和加密。



根据需求选择工具

DIY (Do It Yourself)
Toolchain

OR

AIO (All In One)
Platform

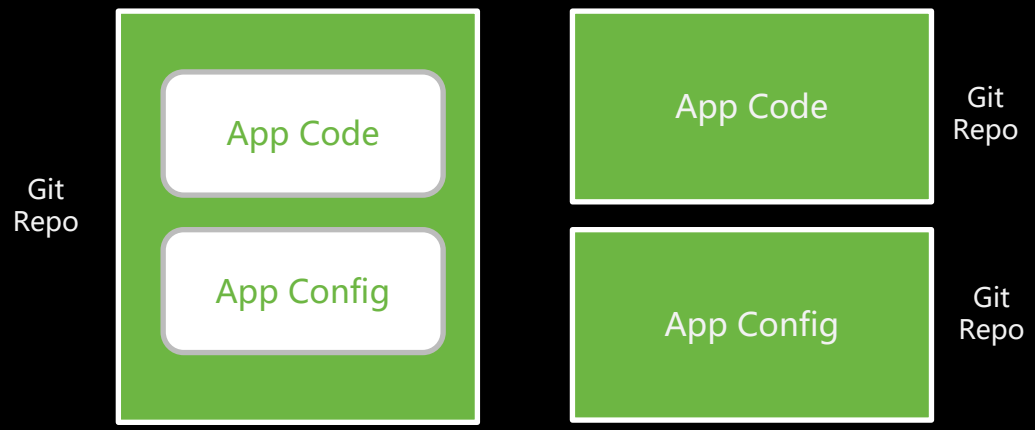
成本

效率

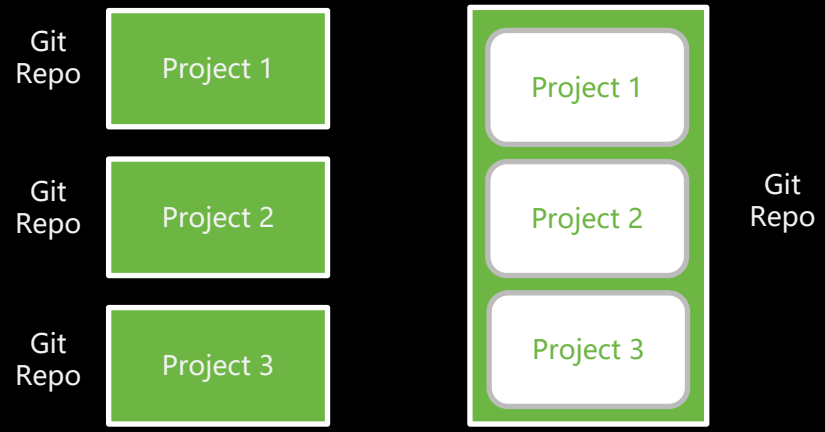
体验

仓库设计

Together or Separated

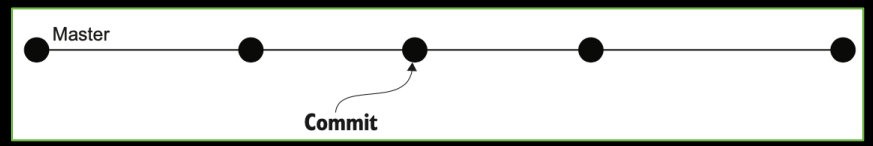


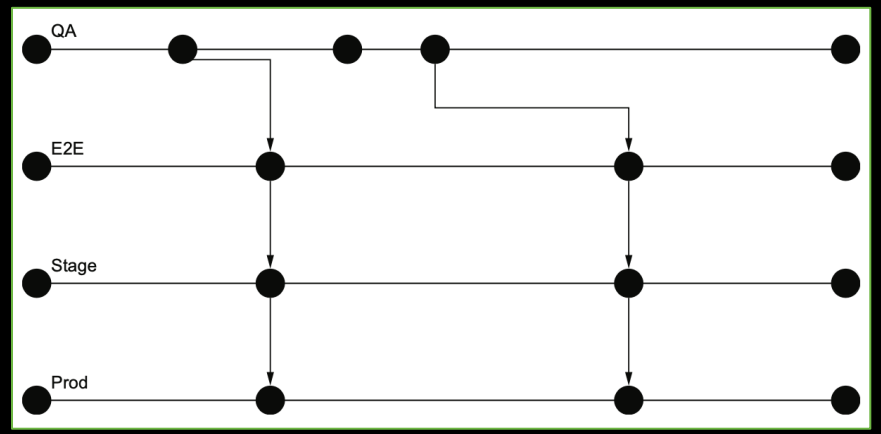
Multi-Repo or Mono-Repo



分支策略

Single-Branch or Multi-Branch





Base - ldap

```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
```

Kustomization.yaml

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

patch.yaml - Staging

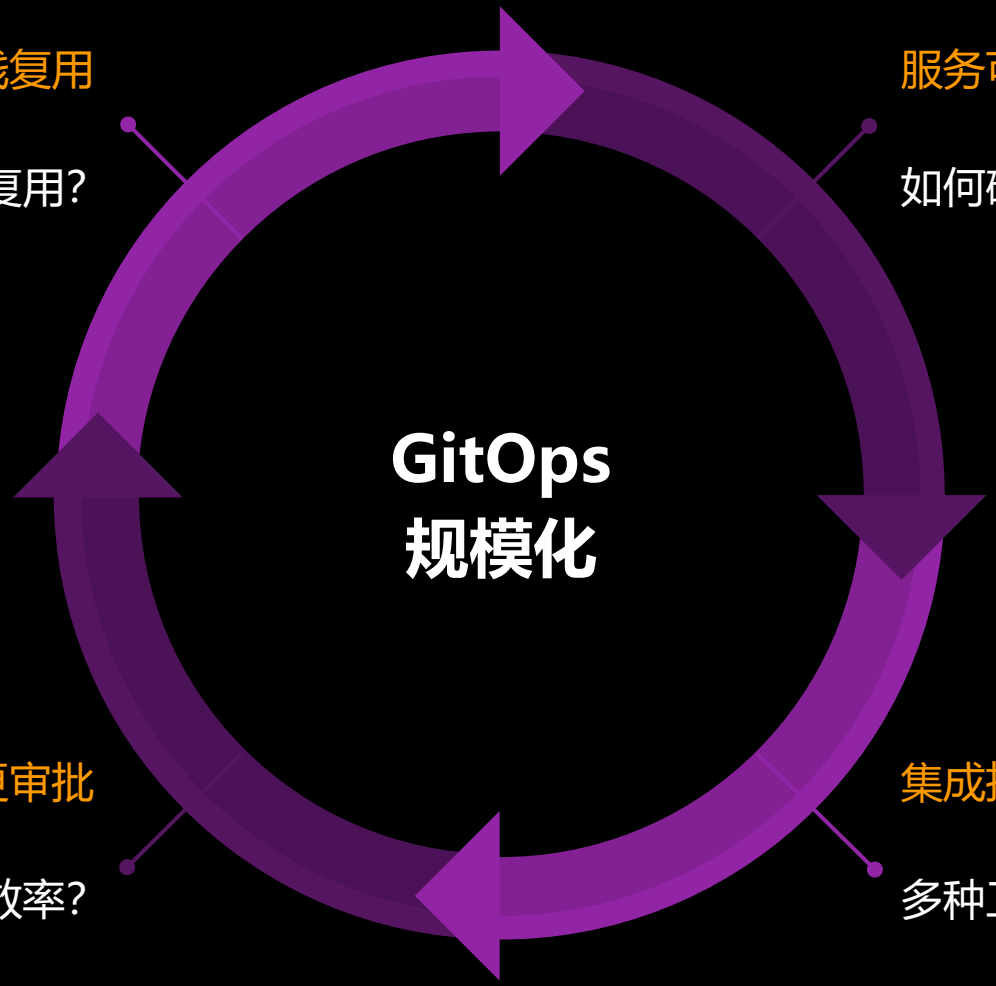
```yaml
# Staging Deployment

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

patch.yaml - Prod

```yaml
# Production Deployment

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

Kustomize

实现规模化的核心关注点



实践复用

如何提高实践的可复用?

服务可用

如何确保服务可用性?

变更审批

如何提升代码评审效率?

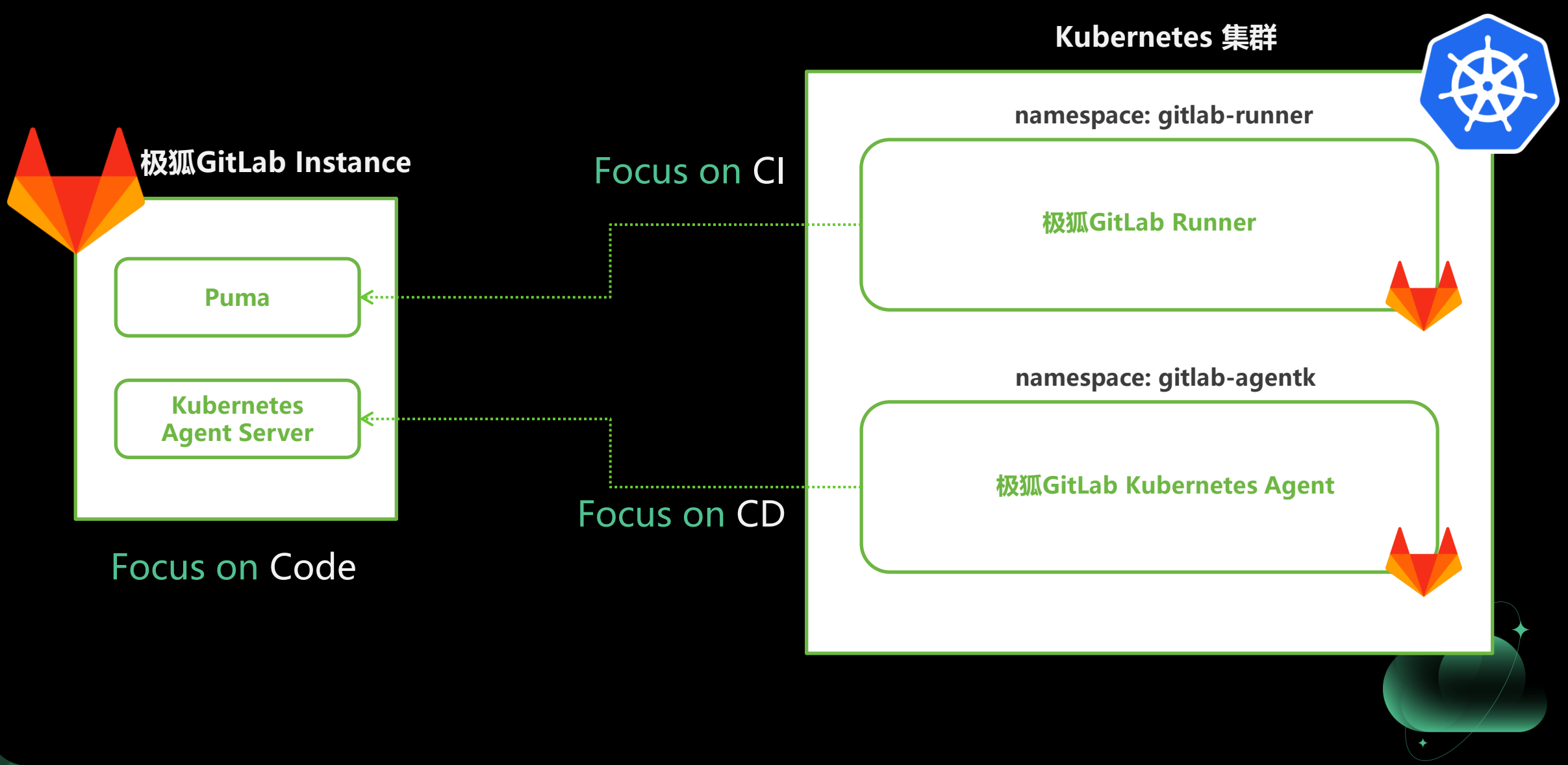
集成扩展

多种工具如何集成扩展?

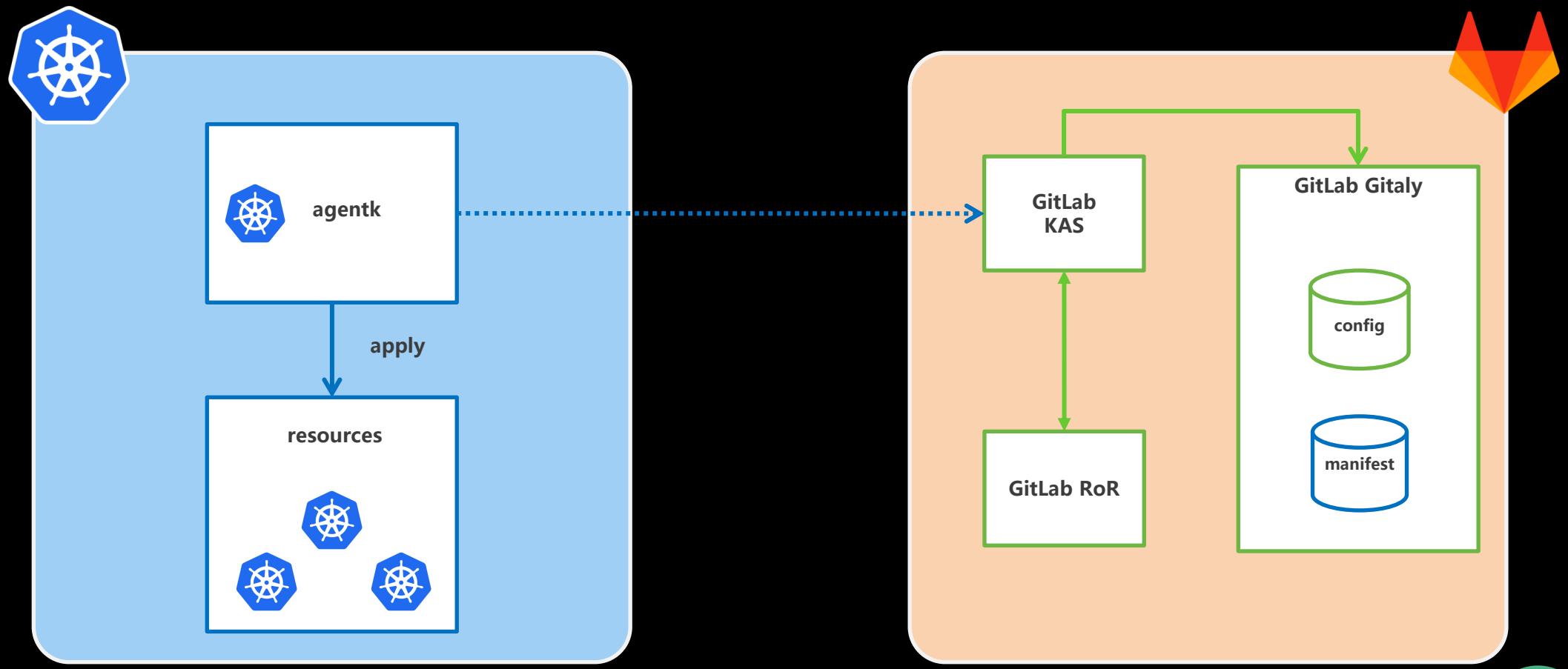
Chapter Ⅲ

极狐(GitLab)的最佳实践

极狐GitLab GitOps 组件概览



极狐GitLab GitOps工作流程



简单配置：低学习成本，低心智负担

配置

.gitlab/agents/agent-name/config.yaml

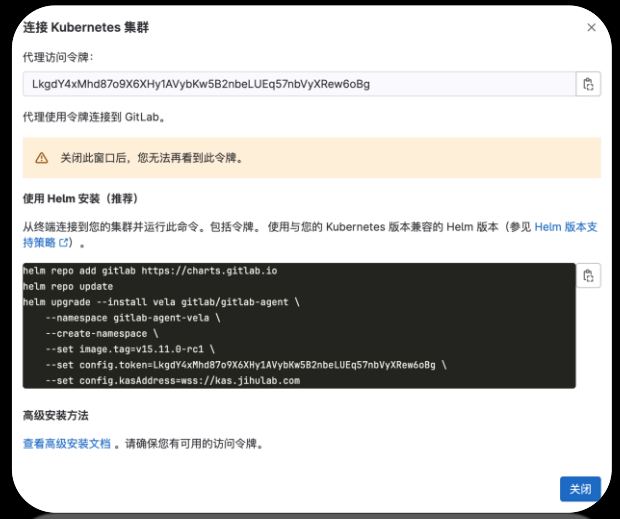
连接集群

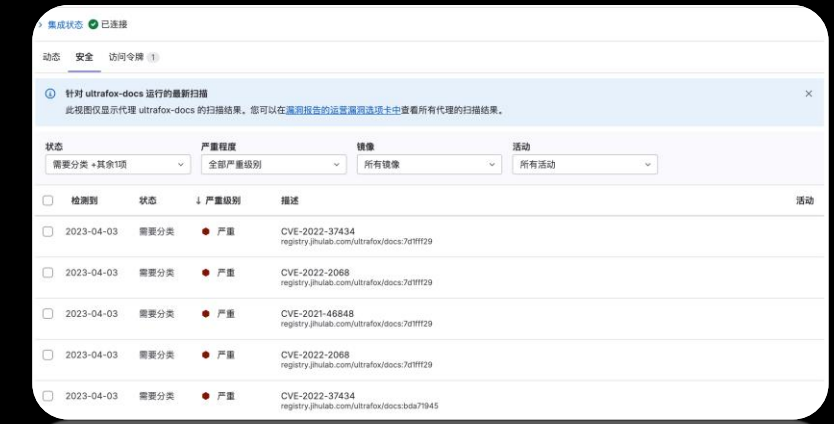
Helm upgrade

自动同步

拉取、部署、扫描

```yaml
gitops:
  manifest_projects:
  - id: cloud-native/gitops/k8s-agent
    default_namespace: default
    paths:
      # Read all YAML files from this directory.
    - glob: '/native-resource/*.yaml'
    - glob: '/kubevela/*.yaml'
    reconcile_timeout: 3600s
    dry_run_strategy: none
    prune: true
    prune_timeout: 3600s
    prune_propagation_policy: foreground
    inventory_policy: must_match
container_scanning:
  cadence: '05 11 * * *'
  vulnerability_report:
    namespaces:
      - default
```





配置简单，快速上手，仅需几行代码，开箱即用的 GitOps

Chapter IV

探索更多的可能

Open Question

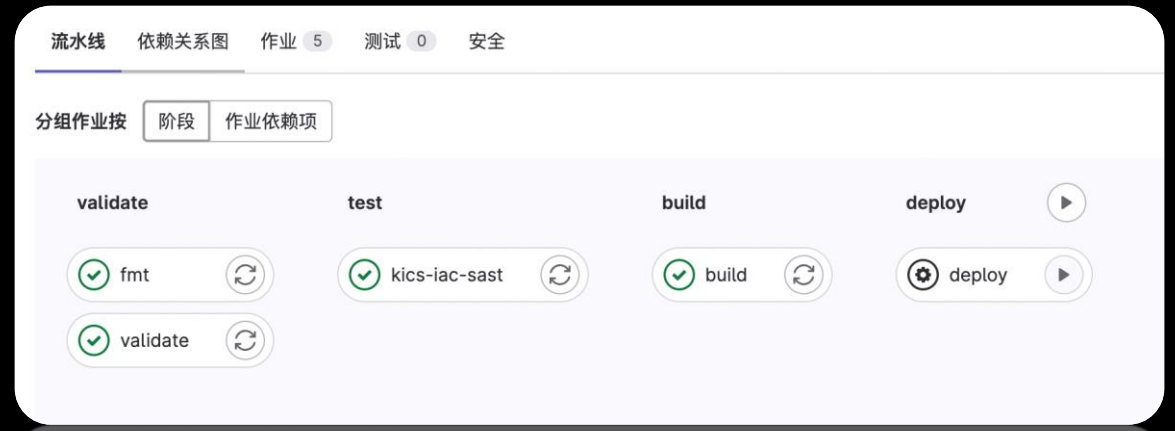
常用的有哪些 GitOps 工具?

一般用这些工具管理哪些资源?

K8S 集群? Cloud Resources? 有其他的吗?

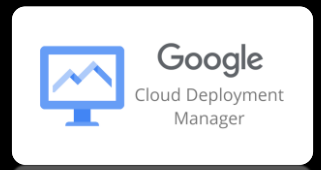
GitLab as Code/Everything as Code

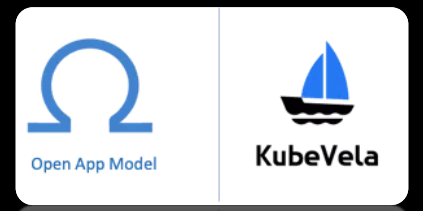
```hcl
terraform {
  required_providers {
    gitlab = {
      source  = "gitlabhq/gitlab"
      version = "15.8.0"
    }
  }
}

variable "gitlab_access_token" {
  type = string
}

provider "gitlab" {
  token = var.gitlab_access_token
}

data "gitlab_group" "cloud_native_group" {
  group_id = 617
}

# Add a project to the group - guoxudong.io/example-create-by-tf
resource "gitlab_project" "sample_group_project" {
  name             = "Example Create by TF"
  namespace_id     = data.gitlab_group.sample_group.group_id
  visibility_level = "public"
  description      = "Example project created by terraform"
  default_branch   = "main"
  emails_disabled  = true
  wiki_enabled     = false
  topics           = ["example"]

  push_rules {
    author_email_regex     = "@jihulab\\.com$"
    commit_committer_check = true
    member_check           = true
    prevent_secrets        = true
  }
}
```



多工具支持

除了极狐GitLab 提供的 KAS 功能，同样支持其他所有的 GitOps 工具，您可以选择任何您想使用的工具。





AI in GitOps

- 根据自然语言生成 k8s/tf resource
- 将 k8s/tf 以自然语言描述
- AI as a Workflow engine
- …

自组织，自适应，自感知，自编程

NOTHING IS IMPOSSIBLE !

Thanks

添加微信请备注姓名公司并说明来意

Microsoft Azure

intel

CLOUD NATIVE COMPUTING FOUNDATION