Python Meetup

上海 · 北京 · 深圳 · 杭州 · 成都

打造 Python 项目的云原生 GitOps 工具链

张晋涛

- 张晋涛
- Apache APISIX committer
- Kubernetes ingress-nginx reviewer
- 『K8S 生态周报』发起人
- GitHub: tao12345666333

- 什么是云原生 GitOps
- 如何对 Python 项目进行云原生交付
- 如何用 GitOps 的方式交付 Python 项目
- 落地经验分享

一、什么是云原生 GitOps

什么是云原生

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.

云原生是一种构建和运行应用程序的方法，它利用了云计算交付模型的优势。

什么是 IaC

- Infrastructure as code
- 通过代码管理和配置基础设施
- 需要版本控制

- 声明式 vs 命令式
- Agent(Pull) vs Agentless(Push)

IaC 的收益

- 降本增效
- 减少错误
- 提高基础设施一致性
- 赋能 DevOps

什么是 GitOps

- XaC 泛指 Infrastructure as code 等
- GitOps == XaC + PR + CI/CD
- 同等对待应用程序代码和 XaC
- 通过 CI/CD pipeline 进行持续集成，交付和部署

云原生 GitOps 带来的收益

- 声明式配置
- 易于管理和追溯
- 消除信息孤岛
- 自动化程度得到提升

二、 Python 项目云原生交付

- 容器化
- 服务编排（可选）
- 部署 manifest

构建镜像

- 问题 1： 用哪个基础镜像
- 问题 2： Dockerfile 怎么写
- 问题 3： 怎么保证安全性（供应链攻击）

```
FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install -r requirements.txt

COPY main.py .

CMD ["python", "main.py"]

EXPOSE 8888
```

Kubernetes 中基于 DNS 的服务发现

两种规定：

- < 名称 >.< 命名空间 >.svc.cluster.<clusterDomain>
- <Pod-IP-Addr>.< 命名空间 >.pod.cluster.<clusterDomain>

- 问题 1： 选择什么方式部署， pod vs deploy vs daemonset
- 问题 2： 是否要写 operator 进行部署
- 问题 3： manifest 如何组织？ Helm？

三、以 GitOps 方式进行交付

- 编写代码
- 提交代码
- 构建镜像
- 发布镜像
- Manifest 更新
- 应用到 Kubernetes

注：此处未包含安全和测试等内容

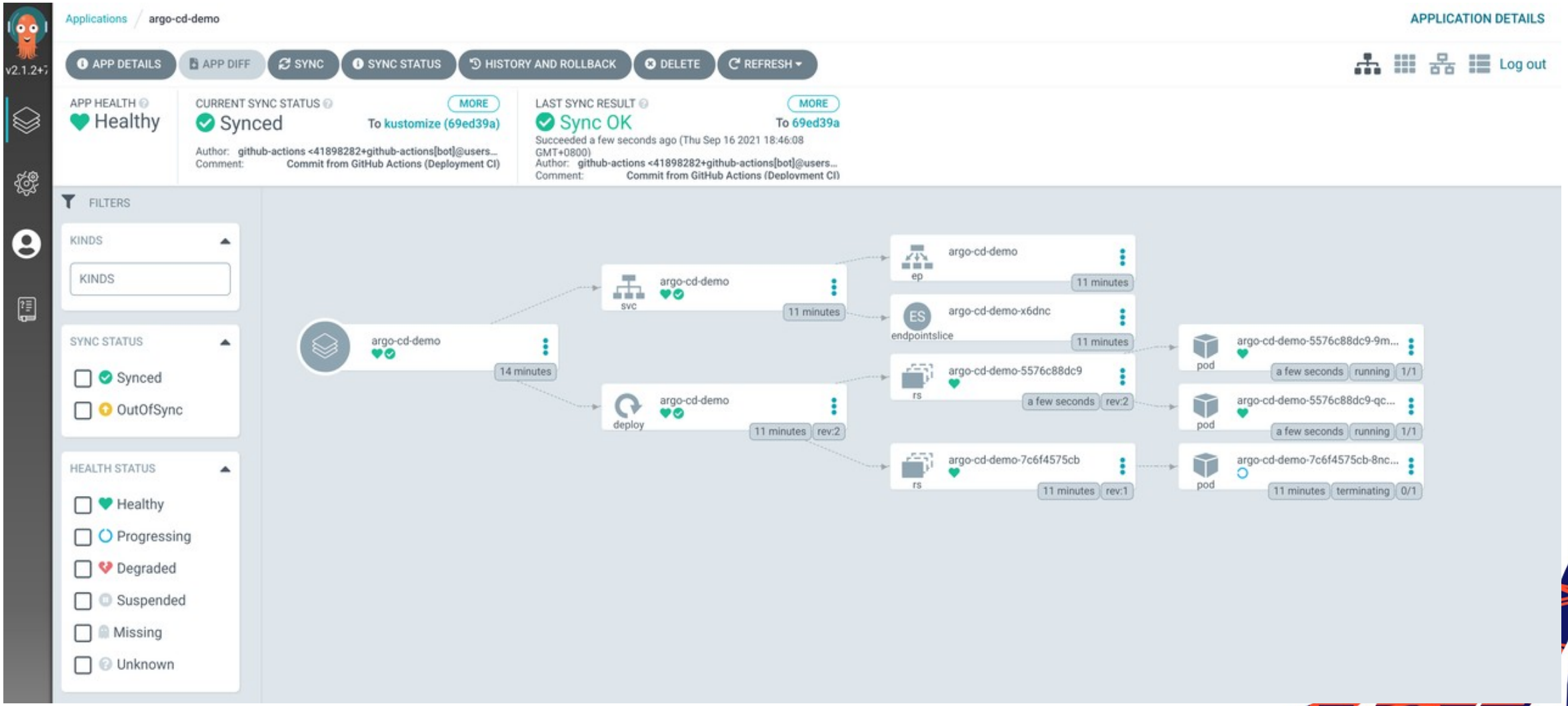
示例项目： https://github.com/tao12345666333/argo-cd-python-demo

```yaml
build:
  name: Push Docker image to GitHub Packages
  runs-on: ubuntu-latest
  permissions:
    packages: write
    contents: read
  steps:
    - name: Check out the repo
      uses: actions/checkout@v2
    - name: Log in to GitHub Docker Registry
      uses: docker/login-action@v1
      with:
        registry: ghcr.io
        username: ${{ github.actor }}
        password: ${{ secrets.GITHUB_TOKEN }}
    - name: Build container image
      uses: docker/build-push-action@v2
      with:
        push: true
        tags: |
          ghcr.io/${{ github.repository }}/argo-cd-demo:${{ github.sha }}

    - name: Image digest
      run: echo ${{ steps.docker_build.outputs.digest }}
```

如何做


```yaml
deploy:
  name: Deploy
  runs-on: ubuntu-latest
  continue-on-error: true
  needs: build

  steps:
    - name: Check out code
      uses: actions/checkout@v2

    - name: Setup Kustomize
      uses: imranismail/setup-kustomize@v1
      with:
        kustomize-version: "4.3.0"

    - name: Update Kubernetes resources
      env:
        DOCKER_USERNAME: ${{ secrets.DOCKER_USERNAME }}
      run: |-
        cd manifests
        kustomize edit set image ghcr.io/${{ github.repository }}/argo-cd-demo:${{ github.sha }}
        cat kustomization.yaml
        kustomize build ./ > ../kustomization/manifests.yaml
        cat ../kustomization/manifests.yaml

    - uses: EndBug/add-and-commit@v7
      with:
        default_author: github_actions
        branch: kustomize
```

如何做

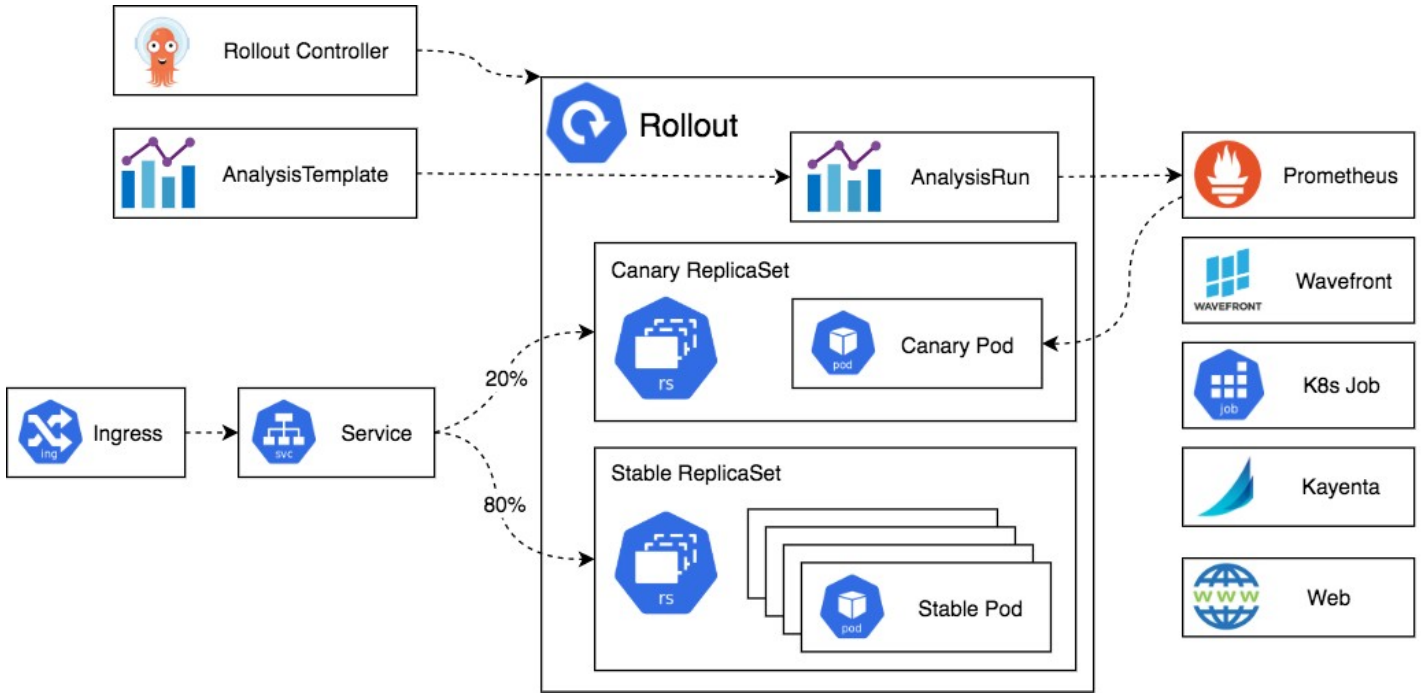

Applications / argo-cd-demo
APPLICATION DETAILS
APP DETAILS
APP DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
APP HEALTH
Healthy
CURRENT SYNC STATUS
Synced
MORE
To kustomize (69ed39a)
Author: github-actions <41898282+github-actions[bot]@users...
Comment: Commit from GitHub Actions (Deployment CI)
LAST SYNC RESULT
Sync OK
MORE
To 69ed39a
Succeeded a few seconds ago (Thu Sep 16 2021 18:46:08 GMT+0800)
Author: github-actions <41898282+github-actions[bot]@users...
Comment: Commit from GitHub Actions (Deployment CI)
FILTERS
KINDS
SYNC STATUS
Synced
OutOfSync
HEALTH STATUS
Healthy
Progressing
Degraded
Suspended
Missing
Unknown
argo-cd-demo
14 minutes
argo-cd-demo
svc
11 minutes
argo-cd-demo
deploy
11 minutes
rev:2
argo-cd-demo
ep
11 minutes
argo-cd-demo-x6dnc
endpointslice
11 minutes
argo-cd-demo-5576c88dc9
rs
a few seconds
rev:2
argo-cd-demo-7c6f4575cb
rs
11 minutes
rev:1
argo-cd-demo-5576c88dc9-9m...
pod
a few seconds
running
1/1
argo-cd-demo-5576c88dc9-qc...
pod
a few seconds
running
1/1
argo-cd-demo-7c6f4575cb-8nc...
pod
11 minutes
terminating
0/1

- 合理的选择工具
- kustomize 可自定义 Kubernetes YAML 的配置
- Argo CD 采用 Pull 模型

四、落地经验分享

- 因地制宜，没有银弹，选择匹配度比较高的
- 统一流程（优先）
- GitHub Action vs GitLab CI

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Rollout
metadata:
  name: example-rollout
spec:
  replicas: 10
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.15.4
        ports:
        - containerPort: 80
  minReadySeconds: 30
  revisionHistoryLimit: 3
  strategy:
    canary:
      maxSurge: "25%"
      maxUnavailable: 0
      steps:
      - setWeight: 10
      - pause:
          duration: 1h # 1 hour
      - setWeight: 20
      - pause: {} # pause indefinitely
```



Argo Rollout

- 镜像构建
- 漏洞扫描
- test case
- 可观测性

Thanks!

感 谢 您 的 观 看