Argo CD 搭配 Kustomize 實作 GitOps 部署

周育緯

About me

- III DevOps contributor
- 8 年 System, DevOps 與 Backend 開發與維護經驗
- 資展國際、工研院: Kubernetes, DevOps 講師

GitOps

- 雲原生 Continuous Deployment
- 所有部署都使用聲明式(declarative)腳本
- 所有部署環境yaml皆存放至Git repository

GitOps

- 所有對環境的調整，皆應由調整Git repostiroy來進行觸發
- 需確保環境與Git上的腳本為一致

GitOps優點

- 可以經由git commit history來紀錄環境從部署至今所有狀態

你有遇到過

- 一個App deployment yaml檔該如何部署出多套環境給不同情境(e.g. dev, staging, testing, pre-prod, prod)嗎？
- 如果各個環境有需要設定不同的參數呢？
  - e.g. deployment名稱不同 (dev-api, staging-api, testing-api, pre-prod-api, Prod-api)

解決辦法?

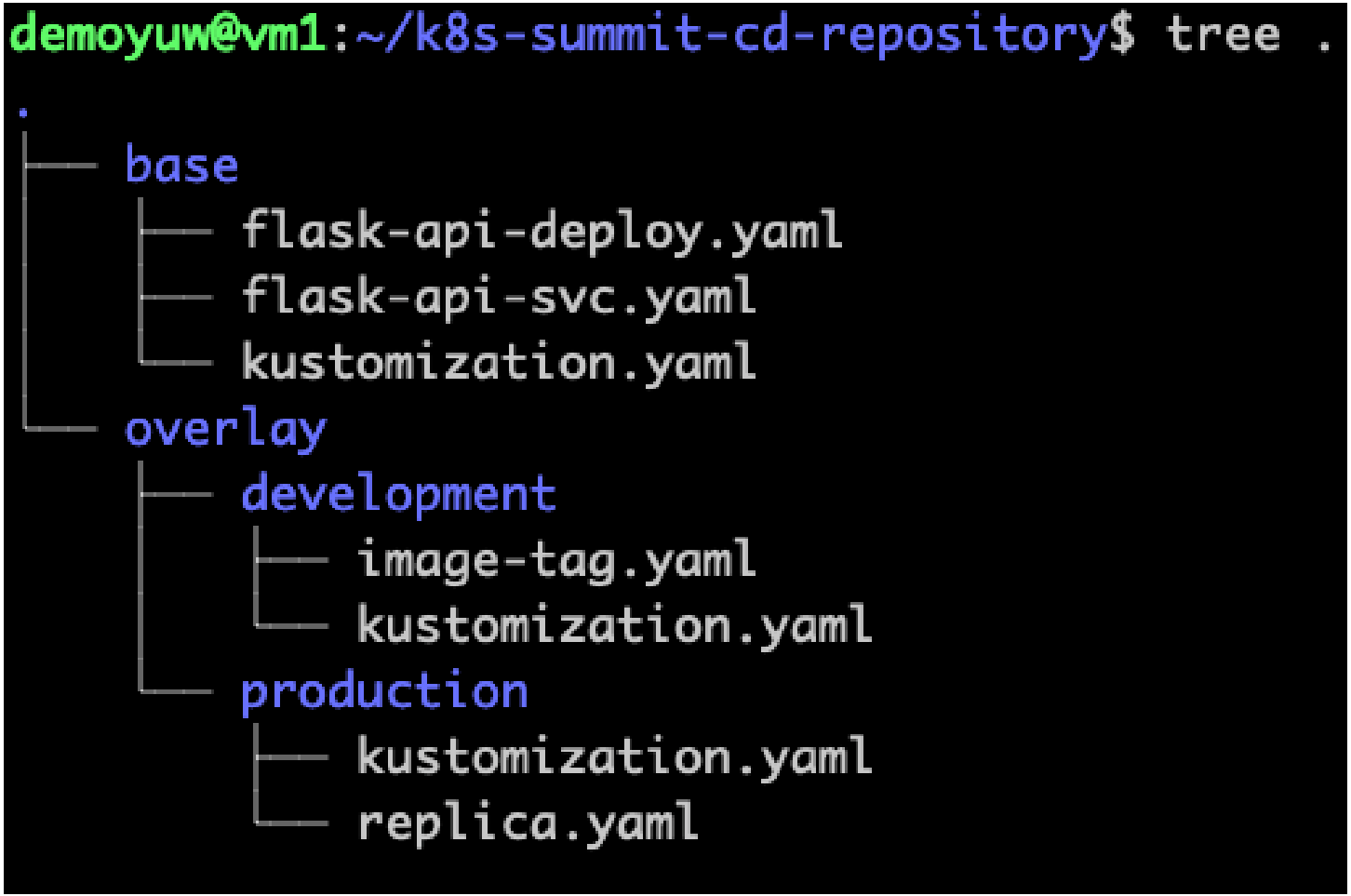
- 寫5份deployment 的yaml 嗎？
  - 需要維護5份yaml files
- 一份deployment yaml在用sed replace嗎?
  - 容易出錯
- 把他打包成Helm chart嗎？在帶入variable嗎？
  - 複雜度過高

Kustomize

- 讓無模板的yaml可以支援多種用途
- CNCF special interest groups (SIGs) 贊助
- Kubernetes 1.14版開始支援

Sample Repoitory

- https://github.com/demoyuw/k8s-summit-cd-repository.git
- Git clone https://github.com/demoyuw/k8s-summit-cd-repository.git
- cd k8s-summit-cd-repository


demoyuw@vm1:~/k8s-summit-cd-repository$ tree .
.
base
flask-api-deploy.yaml
flask-api-svc.yaml
kustomization.yaml
overlay
development
image-tag.yaml
kustomization.yaml
production
kustomization.yaml
replica.yaml

kustomization.yaml 定義四個類別

- resources: 現有資源
- generators: 創建新資源
- meta: 可以同步調整resources, generators 內容
  - vars, namespace, apiVersion, kind

kustomization.yaml 定義四個類別

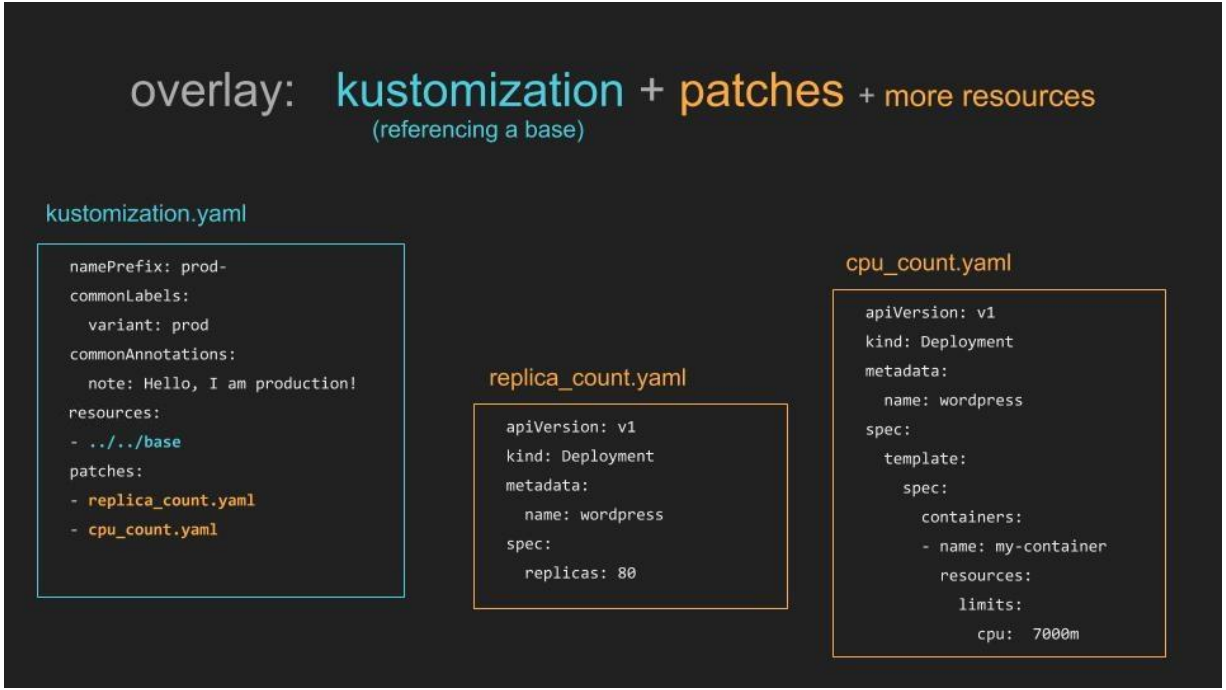
- Transformers: 變形
  - namePrefix
  - nameSuffix
  - Images
  - commonLabels
  - commonAnnotations

base kustomization.yaml

5 lines (5 sloc) | 120 Bytes

```
1  apiVersion: kustomize.config.k8s.io/v1beta1
2  kind: Kustomization
3  resources:
4  - flask-api-deploy.yaml
5  - flask-api-svc.yaml
```

加上Patches


overlay: kustomization + patches + more resources
(referencing a base)
kustomization.yaml
namePrefix: prod-
commonLabels:
variant: prod
commonAnnotations:
note: Hello, I am production!
resources:
- ../../base
patches:
- replica_count.yaml
- cpu_count.yaml
replica_count.yaml
apiVersion: v1
kind: Deployment
metadata:
name: wordpress
spec:
replicas: 80
cpu_count.yaml
apiVersion: v1
kind: Deployment
metadata:
name: wordpress
spec:
template:
spec:
containers:
- name: my-container
resources:
limits:
cpu: 7000m

Overlay deployment kustomization.yaml

```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
resources:
- ../../base
namePrefix: dev-
commonLabels:
  environment: dev
commonAnnotations:
  created_by: yuweichou
patches:
- image-tag.yaml

```

用patch替換dev 使用的image tag

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask
spec:
  template:
    spec:
      containers:
      - name: flask
        image: demoyuw/flask-api:develop
```

Kustomize Comand

- kubectl kustomize {放置 kustomization.yaml 的資料夾名稱}
- 生成替換掉或增加參數的yaml
  - kubectl kustomize overlay/development

```
demoyuw@vm1:~/k8s-summit-cd-repository$ kubectl kustomize overlay/development
apiVersion: v1
kind: Service
metadata:
  annotations:
    created_by: yuweichou
  labels:
    app: flask
    environment: dev
  name: dev-flask-service
spec:
  ports:
  - name: apiport
    port: 10009
    protocol: TCP
  selector:
    app: flask
    environment: dev
  type: NodePort
---
```

```yaml
---
apiVersion: apps/v1
kind: Deployment
metadata:
  annotations:
    created_by: yuweichou
  labels:
    environment: dev
  name: dev-flask
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flask
      environment: dev
  strategy:
    type: Recreate
  template:
    metadata:
      annotations:
        created_by: yuweichou
      labels:
        app: flask
        environment: dev
    spec:
      containers:
      - image: demoyuw/flask-api:develop
        name: flask
```

Use generate YAML and apply on kubernetes

- Generate yaml and apply to k8s
  - kubectl kustomize overlay/development | kubectl apply -f -

```
demoyuw@vm1:~/k8s-summit-cd-repository$ kubectl kustomize overlay/development | kubectl apply -f -
service/dev-flask-service created
deployment.apps/dev-flask created
```

部署出dev deployment and service env

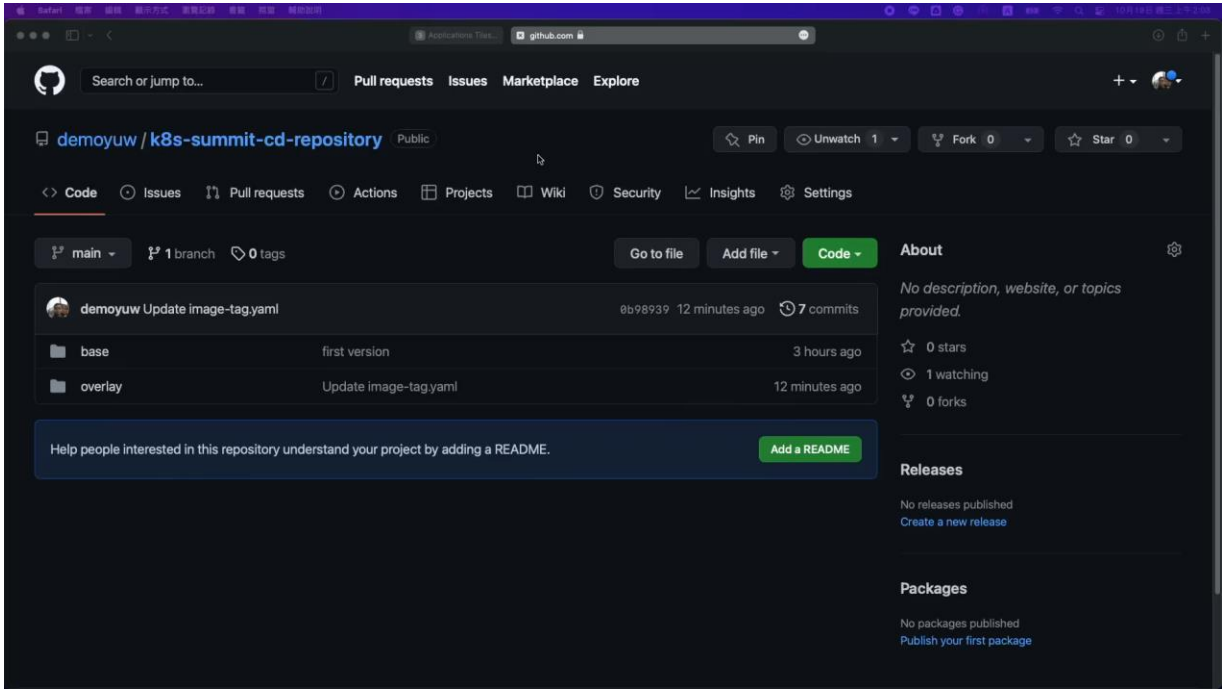
```
demoyuw@vm1:~/k8s-summit-cd-repository$ kubectl get all
NAME                              READY   STATUS    RESTARTS   AGE
pod/dev-flask-74f5c6576-srrhn     1/1     Running   0          30s

NAME                          TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)           AGE
service/dev-flask-service     NodePort    10.43.163.93   <none>        10009:32273/TCP   30s
service/kubernetes            ClusterIP   10.43.0.1      <none>        443/TCP           27h

NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/dev-flask     1/1     1            1           30s

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/dev-flask-74f5c6576     1         1         1       30s
```

Argo CD

- A declarative, GitOps continuous delivery tool for Kubernetes

argo

用ArgoCD 搭配kustomize來部署環境ㄌ



歡迎大家參考 III DevOps

https://www.iiidevops.org

更多詳細教學影音，請觀看我們的 YouTube 影音　影音連結

或訂閱我們的YouTube頻道，享受最新第一手教學資訊。