apiVersion: fn.kpt.dev/v1alpha1

kind: EnsureNameSubstring

metadata:

name: my-config

substring: -taxirides

editMode: append

additionalNameFields:

- version: v1

kind: Service

path: spec/selector/app

- group: networking.k8s.io

version: v1

kind: Ingress

path: spec/defaultBackend/service/name

kind: ConfigMap

apiVersion: v1

metadata:

name: topic-config-taxirides

labels:

app: dyson

namespace: dyson

data:

topic: projects/pubsub-public-data/topics/taxirides-realtime # kpt-set: ${topic}

---

kind: ConfigMap

apiVersion: v1

metadata:

name: file-config-taxirides

labels:

app: dyson

namespace: dyson

data:

file: /data/output.json

apiVersion: apps/v1

kind: Deployment

metadata:

name: dyson-taxirides # kpt-set: dyson-${name}

labels:

app: dyson-taxirides # kpt-set: dyson-${name}

namespace: dyson

spec:

replicas: 1

selector:

matchLabels:

app: dyson-taxirides # kpt-set: dyson-${name}

template:

metadata:

labels:

app: dyson-taxirides # kpt-set: dyson-${name}

spec:

serviceAccountName: dyson-taxirides

securityContext:

runAsNonRoot: true

runAsUser: 1000

runAsGroup: 1000

fsGroup: 1000

containers:

- name: cron

image: us-docker.pkg.dev/my-project-id/repo-name/cron:latest # kpt-set: ${image-repo}cron:latest

env:

- name: STREAM

valueFrom:

configMapKeyRef:

name: file-config-taxirides

key: file

resources:

limits:

cpu: 300m

memory: 400Mi

requests:

cpu: 200m

memory: 200Mi

imagePullPolicy: Always

volumeMounts:

- name: data

mountPath: /data

- name: pulltop

securityContext:

allowPrivilegeEscalation: false

image: us-docker.pkg.dev/my-project-id/repo-name/dyson:latest # kpt-set: ${image-repo}dyson:latest

env:

- name: SYMBOL

valueFrom:

configMapKeyRef:

name: topic-config-taxirides

key: topic

- name: STREAM

valueFrom:

configMapKeyRef:

name: file-config-taxirides

key: file

resources:

limits:

cpu: 300m

memory: 400Mi

requests:

cpu: 200m

memory: 200Mi

imagePullPolicy: Always

volumeMounts:

- name: data

mountPath: /data

- name: websocketd

securityContext:

allowPrivilegeEscalation: false

image: us-docker.pkg.dev/my-project-id/repo-name/websocketd:latest # kpt-set: ${image-repo}websocketd:latest

env:

- name: STREAM

valueFrom:

configMapKeyRef:

name: file-config-taxirides

key: file

ports:

- protocol: TCP

containerPort: 8080

resources:

limits:

cpu: 500m

memory: 600Mi

requests:

cpu: 300m

memory: 312Mi

volumeMounts:

- name: data

mountPath: /data

volumes:

- name: data

emptyDir:

medium: Memory

apiVersion: apps/v1

kind: Deployment

metadata:

name: dyson-taxirides # kpt-set: dyson-${name}

labels:

app: dyson-taxirides # kpt-set: dyson-${name}

namespace: dyson

spec:

replicas: 1

selector:

matchLabels:

app: dyson-taxirides # kpt-set: dyson-${name}

template:

metadata:

labels:

app: dyson-taxirides # kpt-set: dyson-${name}

spec:

serviceAccountName: dyson-taxirides

securityContext:

runAsNonRoot: true

runAsUser: 1000

runAsGroup: 1000

fsGroup: 1000

containers:

- name: cron

image: us-docker.pkg.dev/my-project-id/repo-name/cron:latest # kpt-set: ${image-repo}cron:latest

env:

- name: STREAM

valueFrom:

configMapKeyRef:

name: file-config-taxirides

key: file

resources:

limits:

cpu: 300m

memory: 400Mi

requests:

cpu: 200m

memory: 200Mi

imagePullPolicy: Always

volumeMounts:

- name: data

mountPath: /data

- name: pulltop

securityContext:

allowPrivilegeEscalation: false

image: us-docker.pkg.dev/my-project-id/repo-name/dyson:latest # kpt-set: ${image-repo}dyson:latest

env:

- name: SYMBOL

valueFrom:

configMapKeyRef:

name: topic-config-taxirides

key: topic

- name: STREAM

valueFrom:

configMapKeyRef:

name: file-config-taxirides

key: file

resources:

limits:

cpu: 300m

memory: 400Mi

requests:

cpu: 200m

memory: 200Mi

imagePullPolicy: Always

volumeMounts:

- name: data

mountPath: /data

- name: websocketd

securityContext:

allowPrivilegeEscalation: false

image: us-docker.pkg.dev/my-project-id/repo-name/websocketd:latest # kpt-set: ${image-repo}websocketd:latest

env:

- name: STREAM

valueFrom:

configMapKeyRef:

name: file-config-taxirides

key: file

ports:

- protocol: TCP

containerPort: 8080

resources:

limits:

cpu: 500m

memory: 600Mi

requests:

cpu: 300m

memory: 312Mi

volumeMounts:

- name: data

mountPath: /data

volumes:

- name: data

emptyDir:

medium: Memory

apiVersion: v1

kind: Namespace

metadata:

name: dyson # kpt-set: ${namespace}

labels:

app: dyson

apiVersion: v1

kind: ServiceAccount

metadata:

name: dyson-taxirides

annotations:

iam.gke.io/gcp-service-account: dyson-sa@my-project-id.iam.gserviceaccount.com # kpt-set: ${service-account}@${project-id}.iam.gserviceaccount.com

namespace: dyson

labels:

app: dyson

apiVersion: v1

kind: Service

metadata:

name: dyson-taxirides

annotations:

cloud.google.com/neg: '{"ingress": true}'

labels:

app: dyson

namespace: dyson

spec:

ports:

- port: 80

protocol: TCP

targetPort: 8080

selector:

app: dyson-taxirides

type: ClusterIP

FROM node:18-alpine

WORKDIR /app

ARG NODE\_ENV=production

ENV NODE\_ENV $NODE\_ENV

RUN npm install pulltop --omit=dev

RUN apk add dumb-init

USER 1000

CMD ["sh", "-c", "dumb-init node /app/node\_modules/pulltop/pulltop.js $SYMBOL > $STREAM"]

FROM debian:bullseye as build

RUN apt-get -y update \

&& apt-get -y install git \

&& apt-get -y install build-essential \

&& apt-get -y install curl \

&& apt-get install -y dumb-init

WORKDIR /go/src/app

RUN git clone https://github.com/joewalnes/websocketd.git

WORKDIR /go/src/app/websocketd

RUN make

FROM debian:bullseye-slim

COPY --from=build /go/src/app/websocketd/websocketd /app/websocketd

COPY --from=build /usr/bin/dumb-init /app/dumb-init

USER 1000

CMD ["/bin/sh", "-c", "/app/dumb-init /app/websocketd --port=8080 --devconsole tail -F $STREAM -n 10"]

FROM debian:bullseye

ARG USERNAME=cron-user

ARG USER\_UID=1000

ARG USER\_GID=1000

RUN groupadd --gid $USER\_GID $USERNAME \

&& useradd --uid $USER\_UID --gid $USER\_GID -m $USERNAME

RUN apt-get update && apt-get -y install cron \

&& echo "\*/5 \* \* \* \* cron-user > /data/output.json" >> /etc/cron.d/rotate-output \

&& crontab -u cron-user /etc/cron.d/rotate-output \

&& chmod u+s /usr/sbin/cron

USER cron-user

CMD ["cron", "-f"]