#Init Containers based on the services

apiVersion: v1

kind: Pod

metadata:

name: myapp-pod

labels:

app: myapp

spec:

containers:

- name: myapp-container

image: sreeharshav/rollingupdate:v5

initContainers:

- name: init-myservice

image: busybox:1.28

command: ['sh', '-c', "until nslookup testsvc1.$(cat /var/run/secrets/kubernetes.io/serviceaccount/namespace).svc.cluster.local; do echo waiting for myservice; sleep 2; done"]

- name: init-mydb

image: busybox:1.28

command: ['sh', '-c', "until nslookup testsvc2.$(cat /var/run/secrets/kubernetes.io/serviceaccount/namespace).svc.cluster.local; do echo waiting for mydb; sleep 2; done"]

==============================================================================

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: nginx01

name: nginx01

spec:

replicas: 1

selector:

matchLabels:

run: nginx01

template:

metadata:

labels:

run: nginx01

spec:

containers:

- image: sreeharshav/rollingupdate:v5

name: nginx01

volumeMounts:

- mountPath: /scripts

name: scripts

initContainers:

- image: busybox:latest

name: init-busybox-1

command: ["/bin/sh"]

args: ["-c", "sleep 60"]

- image: busybox:latest

name: init-busybox-2

volumeMounts:

- mountPath: /scripts

name: scripts

command:

- wget

- "-O"

- "/scripts/script.sh"

- https://sree-kubernetes-data.s3.amazonaws.com/script.sh

#args:

# - "wget https://sree-kubernetes-data.s3.amazonaws.com/script.sh"

restartPolicy: Always

volumes:

- name: scripts

emptyDir: {}

**MySQL DB Seeding with Init Containers:**

<https://www.magalix.com/blog/kubernetes-patterns-the-init-container-pattern#:~:text=In%20Kubernetes%2C%20an%20init%20container,database%20schemas%20and%20so%20on>.

apiVersion: v1

kind: Pod

metadata:

name: mydb

labels:

app: db

spec:

initContainers:

- name: fetch

image: mwendler/wget

command: ["wget","--no-check-certificate","https://sample-videos.com/sql/Sample-SQL-File-1000rows.sql","-O","/docker-entrypoint-initdb.d/dump.sql"]

volumeMounts:

- mountPath: /docker-entrypoint-initdb.d

name: dump

containers:

- name: mysql

image: mysql

env:

- name: MYSQL\_ROOT\_PASSWORD

value: "example"

volumeMounts:

- mountPath: /docker-entrypoint-initdb.d

name: dump

volumes:

- emptyDir: {}

name: dump

----ALSO SEEDING ENABLED ON ADMINIER AS WELL---

apiVersion: v1

kind: Pod

metadata:

name: mydb

labels:

app: db

spec:

initContainers:

- name: init-myservice #Not Needed. Kept only for testing.

image: busybox:1.28

command: ['sh', '-c', 'sleep 30']

- name: fetch

image: mwendler/wget

command: ["wget","--no-check-certificate","https://sreeterraformbucketdev.s3.amazonaws.com/Sample-SQL-File-1000rows.sql","-O","/docker-entrypoint-initdb.d/dump.sql"]

volumeMounts:

- mountPath: /docker-entrypoint-initdb.d

name: dump

containers:

- name: mysql

image: mysql

env:

- name: MYSQL\_ROOT\_PASSWORD

value: "India123456"

volumeMounts:

- mountPath: /docker-entrypoint-initdb.d

name: dump

volumes:

- emptyDir: {}

name: dump

---

apiVersion: v1

kind: Service

metadata:

labels:

app: db

name: mydb

spec:

ports:

- port: 3306

protocol: TCP

targetPort: 3306

selector:

app: db

---

apiVersion: v1

kind: Pod

metadata:

name: adminer

labels:

app: adminer

spec:

initContainers:

- name: check-adminer-is-up-1

image: yauritux/busybox-curl

command: ['sh', '-c', 'sleep 45']

- name: check-adminer-is-up-2

image: yauritux/busybox-curl

command: ['sh', '-c', 'curl http://mydb:3306']

containers:

- name: adminer

image: adminer

ports:

- containerPort: 8080

---

apiVersion: v1

kind: Service

metadata:

labels:

app: adminer

name: adminer

spec:

ports:

- port: 8080

protocol: TCP

targetPort: 8080

nodePort: 31944

selector:

app: adminer

type: NodePort