Computer Science & Information Systems

**DevOps for Cloud - Lab Sheet 6 - Module 7**

**(M7: Continuous Deployment)**

This lab sheet needs to be administered along with Module 7: Continuous Deployment

**Notation used in the document**

* ‘>’ represents the terminal, where we type the commands.
* The text mentioned within ‘[‘ and ‘]’ brackets provides additional documentation for the step.

1. Objectives:
2. To demonstrate different deployment strategies using Argo CD Rollouts
3. Pre-requisite:
4. Minikube is installed and running [“minikube start” command]
5. Kubectl is installed
6. Argo CD is installed and running
7. Lab Exercise:

[Ref URL - https://argo-rollouts.readthedocs.io/en/stable/]

**Task 1: Install Argo Rollouts**

1. Open terminal (command prompt) and type the following commands

> minikube start

> kubectl create namespace argo-rollouts

> kubectl apply -n argo-rollouts -f https://github.com/argoproj/argo-rollouts/releases/latest/download/install.yaml

**Task 2: Implement RollingUpdate (Ramped)**

Nginx-deployment.yaml file

# File: nginx-deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

  labels:

    app: nginx

  namespace: default

spec:

  replicas: 3

  selector:

    matchLabels:

      app: nginx

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:latest

        ports:

        - containerPort: 80

---

# Nodeport service to expose the application

apiVersion: v1

kind: Service

metadata:

  name: nginx-service

  namespace: default

spec:

  selector:

    app: nginx

  ports:

    - protocol: TCP

      port: 80

      targetPort: 80

  type: NodePort

**Task 3: Implement Blue-Green Deployment**

Nginx-deployment.yaml file

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

  labels:

    app: nginx

  namespace: default

spec:

  replicas: 1

  selector:

    matchLabels:

      app: nginx

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:stable-alpine3.17-slim

        ports:

        - containerPort: 80

---

apiVersion: argoproj.io/v1alpha1

kind: Rollout

metadata:

  name: nginx-rollout

  namespace: default

spec:

  replicas: 1

  selector:

    matchLabels:

      app: nginx

  strategy:

    blueGreen:

      activeService: nginx-service

      previewService: nginx-preview-service

      autoPromotionEnabled: true

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:latest

        ports:

        - containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

  name: nginx-service

  namespace: default

spec:

  ports:

  - protocol: TCP

    port: 80

    targetPort: 80

  selector:

    app: nginx

---

apiVersion: v1

kind: Service

metadata:

  name: nginx-preview-service

  namespace: default

spec:

  ports:

  - protocol: TCP

    port: 80

    targetPort: 80

  selector:

    app: nginx

[In Argo CD Rollouts, Live (Blue) Environment (pods are original pods) and Preview (Green) environment where pods are marked with labels such as “rollout”. Two images are rolled out here, one for each environment.]

**Task 4: Implement Canary Deployment**

Nginx-deployment.yaml file

apiVersion: argoproj.io/v1alpha1

kind: Rollout

metadata:

  name: nginx-canary

  namespace: default

spec:

  replicas: 2

  selector:

    matchLabels:

      app: nginx

  strategy:

    canary:

      steps:

      - setWeight: 20

      - pause:

          duration: 10m

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:stable-alpine3.17-slim

        ports:

        - containerPort: 80

---

apiVersion: v1

kind: Service

metadata:

  name: nginx-service

  namespace: default

spec:

  ports:

  - protocol: TCP

    port: 80

    targetPort: 80

  selector:

    app: nginx

[The setWeight step specifies that 20% of the traffic should be routed to the canary version. The pause step pauses the rollout for 10 minutes (duration: 10m) to observe the behavior of the canary version before proceeding further.

First, run this code with image set to “nginx:stable-alpine3.17-slim”. Next, replace the image as “nginx:latest” and rerun / sync the manifest with Argo CD]

4. Outputs/Results:

Students are expected to perform the tasks provided in the lab capsule, and thereby gain a practical understanding of the concept of rolling update strategy implementation.

References:

<https://argo-cd.readthedocs.io/en/stable/getting_started/>

<https://argo-rollouts.readthedocs.io/en/stable/>