GitLab CI/CD (Deploying Java App to AKS Cluster)

Pre-requisites:

1. JAVA code
2. Dockerfile
3. Deployment.yml
4. Service.yml
5. AKS cluster

Step1: Create AKS cluster on Azure

- Once the AKS cluster is created, take the kube config and use that as a variable on GitLab.
- Command: **cat config | base64 -w 0**
- CI/CD variable name: **KUBECONFIG_B64**

Step2: Now add the Source code files on gitlab repositories.

- In Gitlab repo add java and pom.xml files
- And add deployment related files (deployment.yml, service.yml, docker file)
- Finally add .gitlab-ci.yml

Step3: Create a namespace and image pull secret on AKS cluster

- **kubectl create ns development**
- **kubectl create secret docker-registry gitlab-registry-secret **
  **--docker-email=<> **
  **--docker-username=<> **
  **--docker-password=<> **
  **--docker-server=registry.gitlab.com **
  -n development

Step4: Now add remaining files

Deployment.yml.template

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: java-deployment
  namespace: development
  labels:
    app: java
spec:
  replicas: 1
  selector:
    matchLabels:
      app: java
  template:
    metadata:
```

```
    labels:
      app: java
  spec:
    imagePullSecrets:
      - name: gitlab-registry-secret
    containers:
    - name: javaapp1
      image: registry.gitlab.com/akhildurga-group/myfirstrepo:${CI_COMMIT_SHORT_SHA}
      ports:
      - containerPort: 8080
```

Service.yml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: java-service
  namespace: development
spec:
  type: LoadBalancer
  selector:
    app: java
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```

Dockerfile

```
FROM openjdk:17-jdk-slim

COPY target/java-cicd-demo-1.0-SNAPSHOT.jar /app.jar

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "/app.jar"]
```

.gitlab-ci.yml

```yaml
include:
  - template: Security/SAST.gitlab-ci.yml
  - template: Security/Dependency-Scanning.gitlab-ci.yml
```

```yaml
  - template: Security/Secret-Detection.gitlab-ci.yml
  - template: Security/Container-Scanning.gitlab-ci.yml

stages:
  - test
  - build
  - testing
  - dockerize
  - container-security
  - deploy

variables:
  MAVEN_OPTS: "-Dmaven.repo.local=.m2/repository"
  IMAGE_NAME: registry.gitlab.com/$CI_PROJECT_PATH
  CS_IMAGE: "$IMAGE_NAME:$CI_COMMIT_SHORT_SHA"
  SECURE_LOG_LEVEL: debug

build:
  stage: build
  image: maven:3.9.5-eclipse-temurin-17
  needs:
    - job: sast
      optional: true
    - job: dependency_scanning
      optional: true
    - job: secret_detection
      optional: true
  script:
    - mvn package -DskipTests
  artifacts:
    paths:
      - target/java-cicd-demo-1.0-SNAPSHOT.jar

test:
  stage: testing
  image: maven:3.9.5-eclipse-temurin-17
  needs:
    - job: build
  script:
    - mvn test
```

```
build_docker:
  stage: dockerize
  image: docker:24.0.5
  services:
    - docker:dind
  needs:
    - job: build
    - job: test
  script:
    - docker build -t $CS_IMAGE .  #$IMAGE_NAME:$CI_COMMIT_SHORT_SHA .
    - docker login -u "$CI_REGISTRY_USER" -p "$CI_REGISTRY_PASSWORD"
registry.gitlab.com
    - docker push $CS_IMAGE  #$IMAGE_NAME:$CI_COMMIT_SHORT_SHA

container_scanning:
  stage: container-security
  needs:
    - job: build_docker
  variables:
    CI_APPLICATION_REPOSITORY: $IMAGE_NAME
    CI_APPLICATION_TAG: $CI_COMMIT_SHORT_SHA

aks-connect:
  image: google/cloud-sdk:alpine
  stage: deploy
  needs:
    - job: container_scanning
  before_script:
    - echo "$KUBECONFIG_B64" | base64 -d > kubeconfig
    - export KUBECONFIG=$PWD/kubeconfig
    - apk add --no-cache curl gettext
    - curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
    - install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl

  script:
    - kubectl get nodes
    - envsubst < deployment.yml.template > deployment.yml
    - kubectl apply -f deployment.yml
```

```
    - kubectl apply -f service.yml
```

Step5: Add Java files:

HelloWorldApp.java

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
public class HelloWorldApp {
    public static void main(String[] args) {
        SpringApplication.run(HelloWorldApp.class, args);
    }

    @RestController
    class HelloController {
        @GetMapping("/")
        public String greet() {
            return "Hello from GitLab CI/CD running in AKS!";
        }
    }
}
```

HelloWorldTest.java

```
package com.example;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.web.client.TestRestTemplate;
import org.springframework.http.ResponseEntity;

import static org.assertj.core.api.Assertions.assertThat;
```

```
@SpringBootTest(webEnvironment =
SpringBootTest.WebEnvironment.RANDOM_PORT)
public class HelloWorldTest {

    @Autowired
    private TestRestTemplate restTemplate;

    @Test
    public void testGreetEndpoint() {
        ResponseEntity<String> response =
                restTemplate.getForEntity("/", String.class);
        assertThat(response.getBody())
                .isEqualTo("Hello from GitLab CI/CD running in AKS!");
    }
}
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>java-cicd-demo</artifactId>
  <version>1.0-SNAPSHOT</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.3</version>
    <relativePath/>
  </parent>

  <properties>
    <maven.compiler.source>11</maven.compiler.source>
    <maven.compiler.target>11</maven.compiler.target>
  </properties>
```

```
<dependencies>
 <dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-web</artifactId>
 </dependency>
 <dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-test</artifactId>
   <scope>test</scope>
 </dependency>
 <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>4.13.2</version>
  <scope>test</scope>
 </dependency>
 <dependency>
   <groupId>commons-collections</groupId>
   <artifactId>commons-collections</artifactId>
   <version>3.2.2</version>
 </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <mainClass>com.example.HelloWorldApp</mainClass>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>3.2.2</version>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>true</addClasspath>
            <mainClass>com.example.HelloWorldApp</mainClass>
```

```
            </manifest>
          </archive>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Step6: Run the pipeline and validate form the AKS cluster

```
durga [ ~ ]$ kubectl get all -n development
NAME                                    READY   STATUS    RESTARTS   AGE
pod/java-deployment-64877896bc-jpqf5    1/1     Running   0          20m

NAME                   TYPE           CLUSTER-IP    EXTERNAL-IP     PORT(S)        AGE
service/java-service   LoadBalancer   10.0.85.31    20.185.210.78   80:31370/TCP   65m

NAME                              READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/java-deployment   1/1     1            1           65m

NAME                                         DESIRED   CURRENT   READY   AGE
replicaset.apps/java-deployment-596ff95cfd   0         0         0       53m
replicaset.apps/java-deployment-64877896bc   1         1         1       20m
```

Verify the results from the browser as well.