**Goal of this lecture:**

To implement an End to End GitOps pipeline using Terraform.

**Steps involved:**

1. Make code changes to change supermario game control to work with “**S**” key on keyboard. Replace webapp folder with the changed code in your repo.
2. Add **e2e-gitops.yaml** file under workflows directory with below code:

name: "Run SAST, Build and push supermario image, scan image, Update deployment and version files"

on:

push:

branches:

- main

env:

VERSION: $(( $(cat version.txt) + 1 ))

jobs:

sonarqube\_sast\_scan:

runs-on: ubuntu-latest

steps:

- name: Checkout Repository

uses: actions/checkout@v3

with:

fetch-depth: 0 # Shallow clones should be disabled for better analysis relevance

- name: SonarQube Scan

uses: sonarsource/sonarqube-scan-action@master

env:

SONAR\_HOST\_URL: ${{ secrets.SONAR\_HOST\_URL }}

SONAR\_TOKEN: ${{ secrets.SONAR\_TOKEN }}

# If you wish to fail your job when the Quality Gate is red, uncomment the

# following lines. This would typically be used to fail a deployment.

# - name: SonarQube Quality Gate Check

# uses: sonarsource/sonarqube-quality-gate-action@master

# timeout-minutes: 5

# env:

# SONAR\_TOKEN: ${{ secrets.SONAR\_TOKEN }}

build\_push\_supermario\_docker\_image:

runs-on: ubuntu-latest

needs: sonarqube\_sast\_scan

steps:

- name: Checkout Repository

uses: actions/checkout@v3

- name: Login to Docker Hub

run: echo "${{ secrets.DOCKERHUB\_TOKEN }}" | docker login -u "${{ secrets.DOCKERHUB\_USERNAME }}" --password-stdin

- name: Build and Push Docker Image

run: |

docker build -t docker.io/raghuthesecurityexpert/supermariogitopsproject:${{ env.VERSION }} .

docker push docker.io/raghuthesecurityexpert/supermariogitopsproject:${{ env.VERSION }}

run\_container\_image\_scan\_on\_supermario\_docker\_image:

runs-on: ubuntu-latest

needs: build\_push\_supermario\_docker\_image

steps:

- name: Checkout Repository

uses: actions/checkout@v3

- name: Login to Docker Hub

run: echo "${{ secrets.DOCKERHUB\_TOKEN }}" | docker login -u "${{ secrets.DOCKERHUB\_USERNAME }}" --password-stdin

- name: Get Docker Image from Docker Hub

run: |

docker pull docker.io/raghuthesecurityexpert/supermariogitopsproject:${{ env.VERSION }}

docker save -o supermariolatestdockerimage.tar docker.io/raghuthesecurityexpert/supermariogitopsproject:${{ env.VERSION }}

- name: Run Trivy vulnerability scanner in tarball mode

uses: aquasecurity/trivy-action@master

with:

input: /github/workspace/supermariolatestdockerimage.tar

exit-code: '0'

severity: 'CRITICAL,HIGH'

update\_k8s\_yaml\_version\_file\_with\_latest\_image\_tag:

runs-on: ubuntu-latest

needs: run\_container\_image\_scan\_on\_supermario\_docker\_image

steps:

- name: Checkout Repository

uses: actions/checkout@v3

- name: Set Git Config

run: |

git config --global user.email "${{ secrets.GIT\_EMAIL}}"

git config --global user.name "${{ secrets.GIT\_USERNAME}}"

- name: Update Deployment YAML

run: |

git pull

sed -i "s|image: raghuthesecurityexpert/supermariogitopsproject:.\*$|image: raghuthesecurityexpert/supermariogitopsproject:${{ env.VERSION }}|" deployment.yaml

echo ${{ env.VERSION }} > version.txt

CURRENT\_VERSION=$(cat version.txt)

git add deployment.yaml version.txt

git commit -m "Updated deployment yaml and version txt file with supermario image tag to ${CURRENT\_VERSION}"

git push

1. Make sure **SONAR\_HOST\_URL**, **SONAR\_TOKEN**, **DOCKERHUB\_TOKEN**, **DOCKERHUB\_USERNAME**, **GIT\_EMAIL**, **GIT\_USERNAME** secrets are defined as repository secrets.
2. Push your changes to remote GitHub Repo.
3. Make sure new tag is assigned to SuperMario Docker image in Dockerhub and same tag is stored in **version.txt** file and **deployment.yaml** file
4. Wait for atleast 3 minutes as ArgoCd default refresh interval is set to **3 minutes.**
5. Verify that supermario docker image with new tag is deployed on Azure Kubernetes Service. Also, new code changes are visible. Make sure to clear cache before checking the new code changes in SuperMario game.