CIS Hardening and Sonobuoy Testing Integration Report

Skander Lahbaiel
LiJo Technologies

April 25, 2024

1 Introduction

As part of our efforts to enhance the security and compliance of our Kubernetes environment managed through Juju and MicroK8s, we have integrated CIS hardening and Sonobuoy conformance tests into our CI pipeline within GitLab. This document outlines the process and provides details of the integration and testing procedures.

2 CI Pipeline Configuration

The CI pipeline is configured in GitLab to trigger a series of commands that prepare and execute security and compliance tests on the MicroK8s cluster. The pipeline is structured as follows:

2.1 GitLab CI Configuration

```
test:
  stage: test
  script:
    - juju switch microk8s-aymen
    - juju ssh microk8s/0 "sudo whoami"   # Verifies sudo access and juju ssh
    - juju scp tests/run-microk8s-tests.sh microk8s/0:/home/ubuntu/run-microk8s-tests.sh   # Copy script to /home/ubuntu
    - juju ssh microk8s/0 "chmod +x /home/ubuntu/run-microk8s-tests.sh"   # Make the script executable
    - juju ssh microk8s/0 "/home/ubuntu/run-microk8s-tests.sh"   # Execute the script
    - echo "Tests completed successfully"
```

2.2 Key Activities

1. Environment Switching: The CI script starts by switching the Juju environment to target the specific MicroK8s cluster (‘microk8s-aymen‘).

2. Script Deployment: It then copies and executes the ‘run-microk8s-tests.sh‘ script on the targeted MicroK8s machine.

3. Test Execution: The script performs the CIS hardening and Sonobuoy tests, ensuring that the cluster meets the required security benchmarks and Kubernetes conformance standards.

3 Test Script Details

The `run-microk8s-tests.sh` bash script is central to the testing process, performing steps to ensure that the MicroK8s cluster is properly secured and compliant.

3.1 Script Contents

```
#!/bin/bash
set -e   # Exit on any error

# Check MicroK8s status
echo "Checking MicroK8s status..."
sudo microk8s.status --wait-ready

# Enable CIS hardening
echo "Enabling CIS hardening..."
sudo microk8s.enable cis-hardening

# Execute kube-bench
echo "Running kube-bench..."
sudo microk8s kube-bench
sudo microk8s kube-bench --check 1.2.3

# Prepare Sonobuoy
echo "Downloading and preparing Sonobuoy..."
wget https://github.com/vmware-tanzu/sonobuoy/releases/download/v0
   .57.1/sonobuoy_0.57.1_linux_amd64.tar.gz
sudo tar -xvf sonobuoy_0.57.1_linux_amd64.tar.gz -C /usr/local/bin/

# Configure Kubernetes client
echo "Configuring Kubernetes client..."
mkdir -p /home/ubuntu/.kube
sudo cp /var/snap/microk8s/current/credentials/client.config /home/
   ubuntu/.kube/config
sudo chown ubuntu:ubuntu /home/ubuntu/.kube/config
chmod 644 /home/ubuntu/.kube/config
export KUBECONFIG=/home/ubuntu/.kube/config
echo "KUBECONFIG set to: $KUBECONFIG"

# Run Sonobuoy tests
echo "Running Sonobuoy tests..."
sonobuoy run --wait
results=$(sonobuoy retrieve)
mkdir -p ~/sonobuoy_results
tar -xvf $results -C ~/sonobuoy_results
echo "Tests complete. Results stored in ~/sonobuoy_results"
```

4 Conclusion

The integration of CIS hardening and Sonobuoy testing into our CI pipeline ensures continuous compliance and security assessment of our Kubernetes environment. By automating these tests, we maintain high standards for security and operational reliability, while also enabling quick feedback and issue resolution.

This report serves as a comprehensive overview of the procedures and configurations employed to achieve this integration. The continuous integration setup not only automates the process but also embeds security and compliance into our development lifecycle, significantly reducing risk and enhancing system stability.

5 Appendix: Node Preparation

Before running the GitLab-CI script, certain preliminary steps were required on the node of the cluster to ensure the environment was correctly configured:

```
sudo visudo
# Add the following line:
ubuntu ALL=(ALL) NOPASSWD: ALL

sudo usermod -a -G microk8s ubuntu
# Change the ownership of the .kube directory
sudo chown -R ubuntu ~/.kube
newgrp microk8s
```

6 Screenshot from GitLab GUI

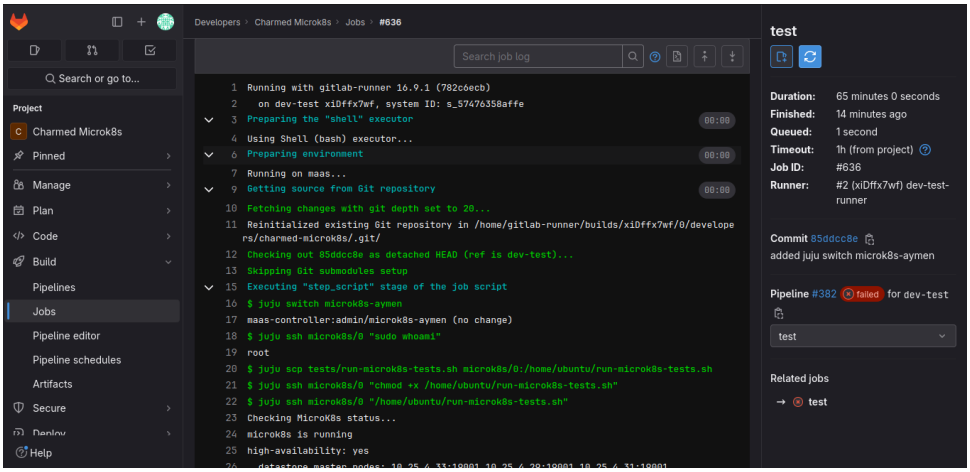

Figure 1: Execution of the scripts