FINAL REPORT

ADVANCED COMPUTER NETWORK SECURITY

Research and Implementation of Security Features in Microservices - Cloud Workload Protection with Kubescape

Presented By Group 9

MEMBERS

NGỌC THƠ

THU HIỀN

NGỌC NHUNG

PROCESS

01 Context

02 Achieved Results

03 Approach Methods

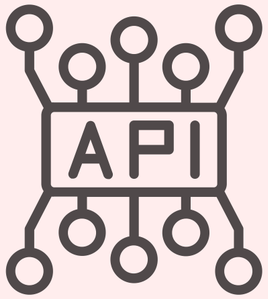
04 Feature and use case

05 Architecture

06 Demo Scenarios

1. CONTEXT

Microservice



A method of software development where applications are built as a collection of **small, independent** services communicating through clear APIs.

Easily **upgradable** and **scalable** applications.

K8S **provides an ideal environment** for deploying and managing these microservices

Kuberscape = K8S + Security

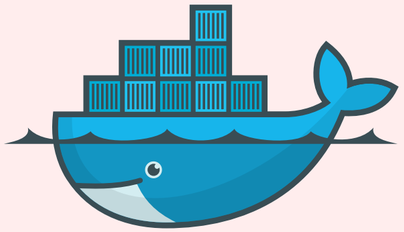
Utilizing a set of predefined rules to:

- Risk Analysis
- Security Compliance
- Misconfiguration Scanning

Helps K8s users and administrators can save time, effort, and resources.

In summary, using microservices with Kubernetes and Kuberscape helps us build applications that are easier to manage, upgrade, and scale, while also being secure.

2. ACHIEVED RESULTS



Create, manage, and run containers.

Create a local Kubernetes environment on the computer. (Minikube)

An automated security tool that helps detect vulnerabilities and risks in our Kubernetes cluster configurations, allowing continuous security monitoring and improvement.

Results

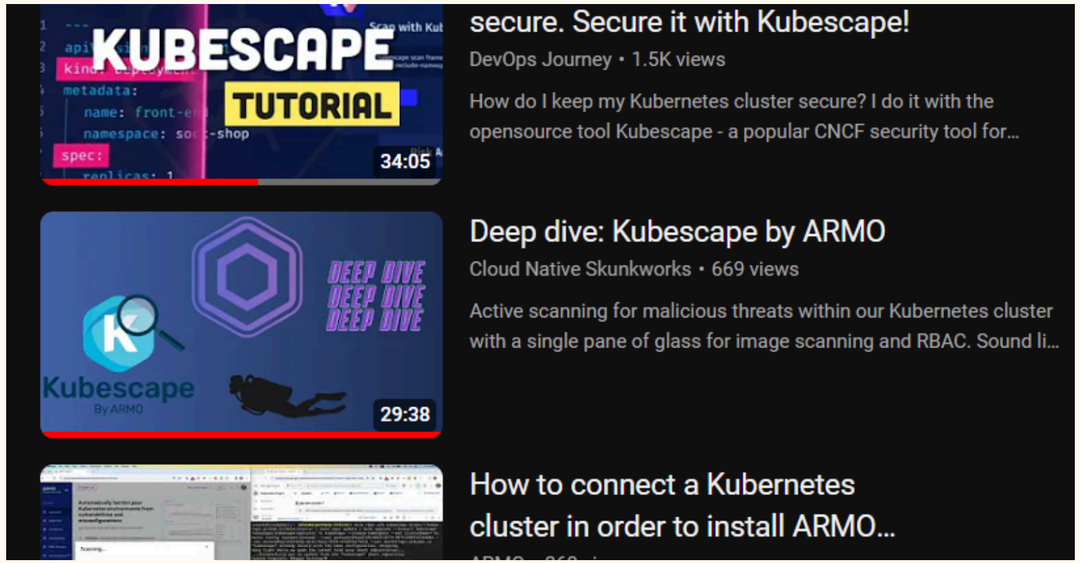
1. Successfully installed and deployed Minikube.
2. Applied Kubescape to scan for vulnerabilities in the K8s cluster.
3. Completed 5/5 test scenarios for the tool.

By linking these steps together, we've built a comprehensive system where we can develop, test, and secure our microservices efficiently.

3. APPROACH METHODS

Some Useful Documents

- Official websites of K8S, Minikube, Kubescape, Docker.
- Provided course materials.
- Relevant blogs/GitHub repositories.
- YouTube videos.



Knowledge Required

- Learn the architecture and components of K8S.
- How to deploy Minikube.
- Install and use Docker.
- Install and use Kubescape.
- Understand scan results and find ways to fix vulnerabilities.

Severity	Control name	Failed r
Critical	Disable anonymous access to Kubelet service	0
Critical	Enforce Kubelet client TLS authentication	0
High	Ensure CPU limits are set	7
High	Ensure memory limits are set	7
Medium	Prevent containers from allowing command execution	2
Medium	Non-root containers	1
Medium	Allow privilege escalation	8
Medium	Ingress and Egress blocked	1
Medium	Automatic mapping of service account	1
Medium	Administrative Roles	2
Medium	Container hostPort	1
Medium	Cluster-internal networking	1

4. FEATURES AND USE CASE

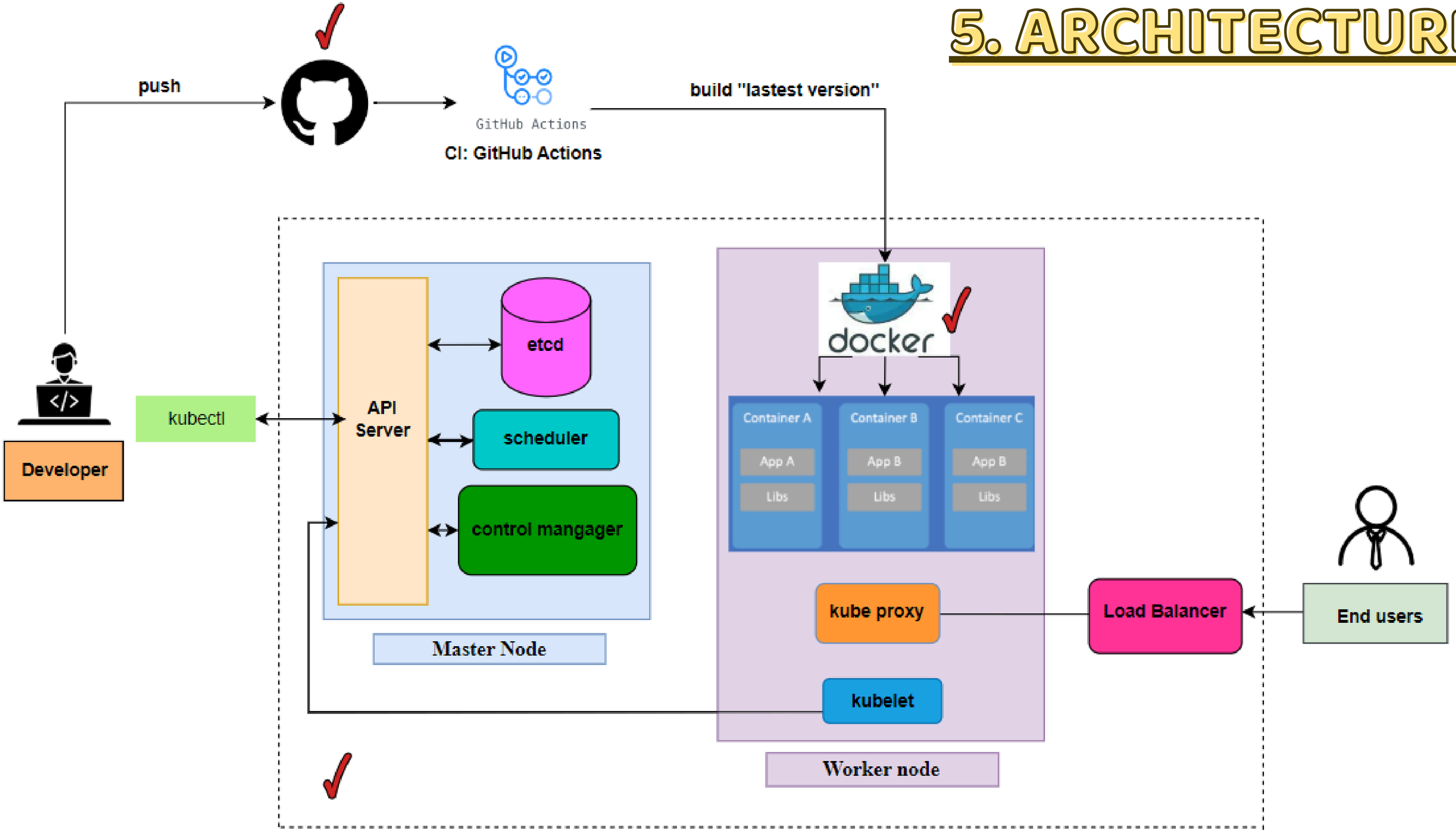
Features

1. Scan and detect security vulnerabilities.
2. Analyze risks and compliance levels with security standards such as CIS Benchmark, MITRE ATT&CK, and NSA-CISA frameworks.
3. User-friendly command-line interface.
4. Flexible output formats: JSON, XML, HTML, or PDF.
5. Scan Kubernetes objects and Helm Charts.

Use case

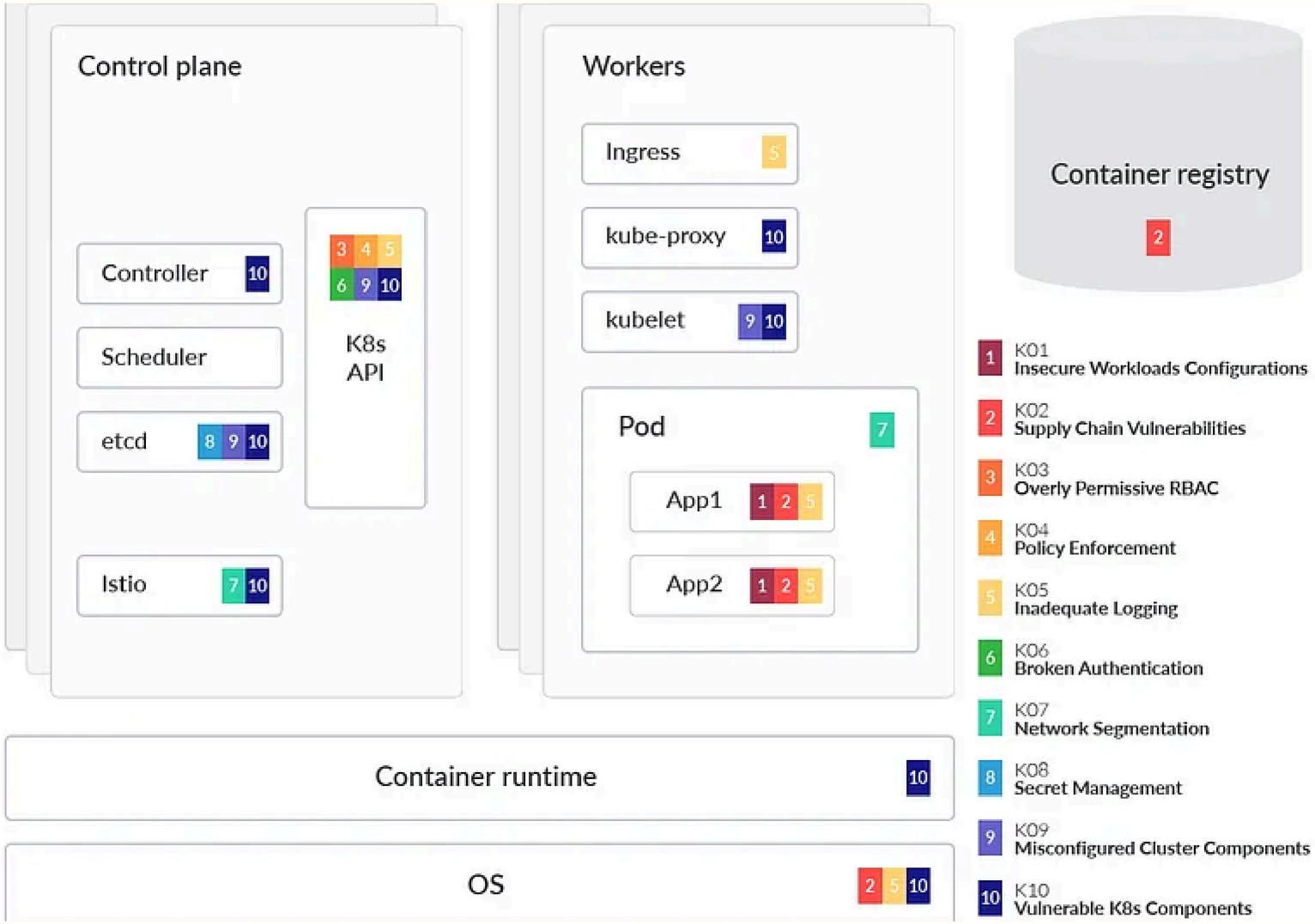
1. Detect security vulnerabilities.
2. Comply with security standards and regulations.
3. Check the configuration of Helm Charts before deployment to K8s.
4. Inspect RBAC (Role-Based Access Control) access rights.
5. Evaluate security risks: Identify vulnerable points.
6. Automate security processes: Integrate K8s into CI/CD pipelines or automation workflows.

5. ARCHITECTURE



6. DEMO SCENARIOS

OWASP Kubernetes Top 10



6. DEMO SCENARIOS

1. Use ARMO Platform to fix CPU and memory limits vulnerability

Ensuring that memory limits are set for containers in Kubernetes is crucial because it helps: Manage resources, mitigate resource contention, and impact the performance of other containers.

Severity	Control name	Failed resources	All Resources	Compliance score
Critical	Disable anonymous access to Kubelet service	0	0	Action Required *
Critical	Enforce Kubelet client TLS authentication	0	0	Action Required *
High	Applications credentials in configuration files	1	52	98%
High	Ensure CPU limits are set	10	25	60%
High	Ensure memory limits are set	11	25	56%
Medium	Prevent containers from allowing command execution	2	92	98%
Medium	Non-root containers	10	25	60%

Select Administrator: Windows PowerShell

```
PS C:\Windows\system32> kubescape scan framework mitre
⚠  current version 'v3.0.8' is not updated to the latest release: 'v3.0.9'
☑ Initialized scanner
☑ Loaded policies
☑ Loaded exceptions
☑ Loaded account configurations
☑ Accessed Kubernetes objects
☑ Collected RBAC resources
Control: C-0012 100% |████████████████████████████████████████| (26/26, 22 it/s)
☑ Done scanning. Cluster: minikube
☑ Done aggregating results
```

Framework scanned: MITRE

Controls	26
Passed	10
Failed	14
Action Required	2

Failed resources by severity:

Critical	0
High	7
Medium	27
Low	3

Run with '--verbose'/'-v' to see control failures for each resource.

Severity	Control name	Failed resources	All Resources	Compliance score
Critical	Disable anonymous access to Kubelet service	0	0	Action Required *
Critical	Enforce Kubelet client TLS authentication	0	0	Action Required *
High	Applications credentials in configuration files	1	52	98%
High	List Kubernetes secrets	6	92	93%
Medium	Prevent containers from allowing command execution	2	92	98%
Medium	Roles with delete capabilities	4	92	96%
Medium	Delete Kubernetes events	2	92	98%
Medium	Administrative Roles	2	92	98%
Medium	CoreDNS poisoning	4	92	96%
Medium	Access container service account	9	66	86%
Medium	Cluster internal networking	2	7	71%

8:06 PM
5/10/2024

6. DEMO SCENARIOS

2. Configure Security Context to prevent Escalating Privilege:

Check whether the ***allowPrivilegeEscalation*** field in the container's ***securityContext*** is set to ***False***.

Severity	Control name	Failed resources	All Resources	Compliance score
Critical	Disable anonymous access to Kubelet service	0	0	Action Required *
Critical	Enforce Kubelet client TLS authentication	0	0	Action Required *
High	Ensure CPU limits are set	7	25	72%
High	Ensure memory limits are set	7	25	72%
Medium	Prevent containers from allowing command execution	2	92	98%
Medium	Non-root containers	10	25	60%
Medium	Allow privilege escalation	8	25	68%
Medium	Ingress and Egress blocked	12	25	52%
Medium	Automatic mapping of service account	14	79	82%
Medium	Administrative Roles	2	92	98%

ARMO Platform | C-0016 - Allow privilege escal

cloud.armosec.io/failed-resource/view_v2?controlIds=C-0016&frameworkName=NSA&reportGuid=115fd8e0-7816-4942-a53f-7d7f2e7bc255

Ẩn danh

You have **20 days** left in your trial | Upgrade now | Finish setup

ARMO
POWERED BY KUBESCAPE

New customer 98fe11ab24ad

Failed Resources:

Search

- default
  - balanced (Deployment)
  - bar-app (Pod)
  - foo-app (Pod)
  - mongo-depl (Deployment)
  - nginx-depl (Deployment)
- ingress-nginx
  - ingress-nginx-controller (Deployment)
- kubescape
  - node-agent (DaemonSet)
  - service-discovery (Job)

Cluster: minikube | **Resource:** balanced ()

Create ticket

C-0016

```
36          resources:
37            limits:
38              cpu: 500m
39              memory: 200Mi
40          terminationMessagePath: /dev/termination-log
41          terminationMessagePolicy: File
42        dnsPolicy: ClusterFirst
43        restartPolicy: Always
44        schedulerName: default-schedul
45        securityContext: {}
46        terminationGracePeriodSeconds:
47
```

C-0016

```
36          resources:
37            limits:
38              cpu: 500m
39              memory: 200Mi
40          terminationMessagePath: /dev/terminatior
41          terminationMessagePolicy: File
42 +        securityContext:
43 +          allowPrivilegeEscalation: false
44 +          privileged: false
45        dnsPolicy: ClusterFirst
46        restartPolicy: Always
47        schedulerName: default-scheduler
48        securityContext: {}
49        terminationGracePeriodSeconds: 30
50
```

7:02 AM
5/11/2024

6. DEMO SCENARIOS

3. Schedule Kubescape for automatic scanning

Set up an automation workflow to use Kubescape to scan Kubernetes environments periodically. This ensures that security concerns are checked regularly and automatically, aiding in early detection and remediation of security vulnerabilities.

YAML JSON

```yaml
apiVersion: batch/v1
kind: CronJob
metadata:
  name: kubescape-scanv3
spec:
  schedule: "0 12 * * *"  # scan everyday at 12:00 AM
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: kubescape
            image: gcr.io/k8s-minikube/kicbase:v0.0.43  # Image Kubescape
            command: ["kubescape", "scan", "framework", "nsa"]
          restartPolicy: OnFailure

```

Search this site
- Documentation
- Getting started
- Concepts
  - Overview
  - Cluster Architecture
  - Containers
  - Workloads
    - Pods
    - Workload Management
      - Deployments
      - ReplicaSet
      - StatefulSets
      - DaemonSet
      - Jobs
      - Automatic Cleanup for Finished Jobs
      - **CronJob**

CronJob

FEATURE STATE: `Kubernetes v1.21 [stable]`

A *CronJob* creates Jobs on a repeating schedule.

CronJob is meant for performing regular scheduled actions such as backups, report generation, and so on. One CronJob object is like one line of a *crontab* (cron table) file on a Unix system. It runs a Job periodically on a given schedule, written in Cron format.

CronJobs have limitations and idiosyncrasies. For example, in certain circumstances, a single CronJob can create multiple concurrent Jobs. See the limitations below.

When the control plane creates new Jobs and (indirectly) Pods for a CronJob, the `.metadata.name` of the CronJob is part of the basis for naming those Pods. The name of a CronJob must be a valid DNS subdomain value, but this can produce unexpected results for the Pod hostnames. For best compatibility, the name should follow the more restrictive rules for a DNS label. Even when the name is a DNS subdomain, the name must be no longer than 52 characters. This is because the CronJob controller will automatically append 11 characters to the name you provide and there is a constraint that the length of a Job name is no more than 63 characters.

Example

CronJob API reference
Edit this page
Create child page
Create an issue
Print entire section

Example
Writing a CronJob spec
Schedule syntax
Job template
Deadline for delayed Job start
Concurrency policy
Schedule suspension
Jobs history limits
Time zones
CronJob limitations
Unsupported TimeZone specification
Modifying a CronJob
Job creation
What's next

6. DEMO SCENARIOS

4. Use CIS framework to scan Minikube

The CIS (center for information security) framework provides a comprehensive set of best practices for securing Kubernetes environments, helping to ensure that your cluster adheres to industry standards for security.

```
PS C:\Windows\system32> kubescape scan framework cis-v1.23-t1.0.1
 ⚠   current version 'v3.0.8' is not updated to the latest release: 'v3.0.9'
 ☑  Initialized scanner
 ☑  Loaded policies
 ☑  Loaded exceptions
 ☑  Loaded account configurations
 ☑  Accessed Kubernetes objects
 ☑  Collected RBAC resources
Control: C-0206 100% |
 ☑  Done scanning. Cluster: minikube
 ☑  Done aggregating results


─────────────────────────────────────────────────


Framework scanned: cis-v1.23-t1.0.1
```

README

F. Kubescape

F.1. Tổng quan về Kubescape

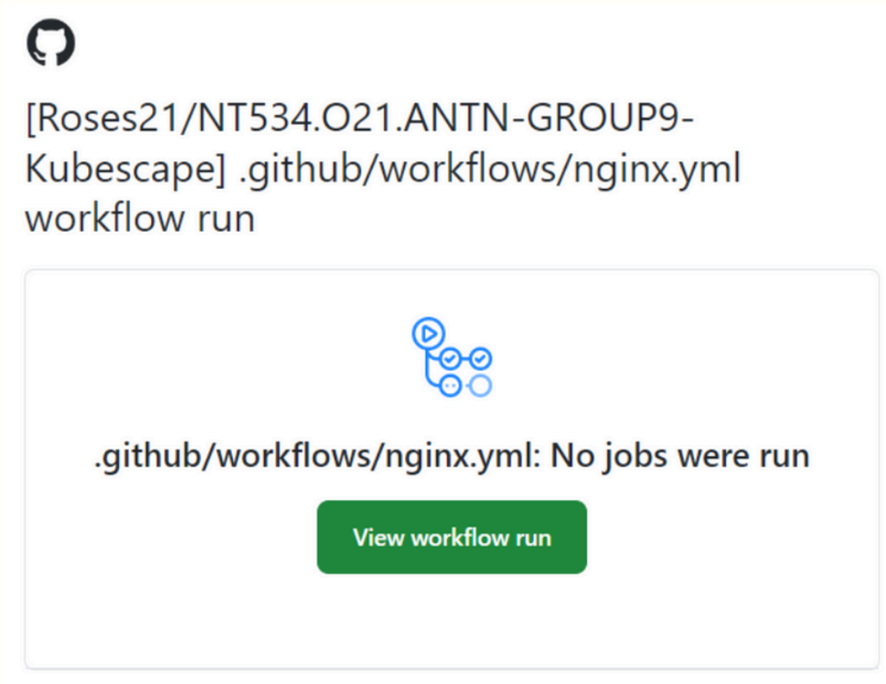
- Kubescape là một nền tảng bảo mật Kubernetes mã nguồn mở. Nhắm mục tiêu vào người thực hành DevSecOps hoặc kỹ sư nền tảng, nó cung cấp giao diện CLI dễ sử dụng, định dạng đầu ra linh hoạt và khả năng quét tự động. Nó giúp người dùng và quản trị viên Kubernetes tiết kiệm thời gian, công sức và tài nguyên quý giá.
- Kubescape có thể **chạy dưới dạng một bộ các microservices bên trong cluster K8s**. Điều này cho phép bạn liên tục theo dõi trạng thái của một cụm.
- Kubescape scans:
    - Cluster.
    - Manifest files (YAML files, Helm Chart).
    - Code repositories.
    - Container registries.
    - Images.

      => Phát hiện: cấu hình sai thông qua 3 khung NSA, MITRE, CIS; lỗ hổng phần mềm và tính toán được mức điểm của risks.
- Chi tiết 3 khung mà Kubescape sử dụng:
    - NSA-CISA (National Security Agency - Cybersecurity and Infrastructure Security Agency): gồm 24 controls, đề cập những vấn đề về: cấu hình cluster, quyền truy cập và xác thực, pod và container, bảo mật mạng, quản

6. DEMO SCENARIOS

5. Registry Scanning & Repository Scanning

This scenario discusses how GitHub Actions can enhance the security of the CI/CD pipeline by automating security-related tasks and providing integration capabilities with security tools, version control, access control, and testing.



https://github.com/Roses21/NT534.O21.ANTN-GROUP9-Kubescape

Roses21/NT534.O21.ANTN-GF | Báo cáo Cuối kỳ ATMMTNC - E

canva.com/design/DAGDhcdW5po/PQQw515jgiGwx5SdkpvGrA/edit

Tệp | Đổi cỡ & Chuyển đổi Magic | Báo cáo Cuối kỳ ATMMTNC | Dùng thử bản Pro trong 30 ngày | Thuyết trình | Chia sẻ

Thiết kế | Thành phần | Văn bản | Thương hiệu | Tải lên | Vẽ | Dự án | Ứng dụng

Tìm kiếm hình ảnh

Tải lên tệp

Ghi hình

Hình ảnh | Video | Âm thanh

Vị trí

FINAL REPORT

ADVANCED COMPUTER NETWORK SECURITY

Research and Implementation of Security Features in Microservices - Cloud Workload Protection with Kubescape

Presented By Group 9

HH

1

Dashboard

Ghi chú | Thời lượng | Đếm ngược | Trang 1 / 20 | 40%

8:48 AM 6/8/2024

Thank You Very Much