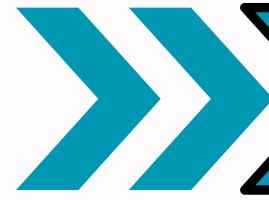




ASEAN CYBER SHIELD



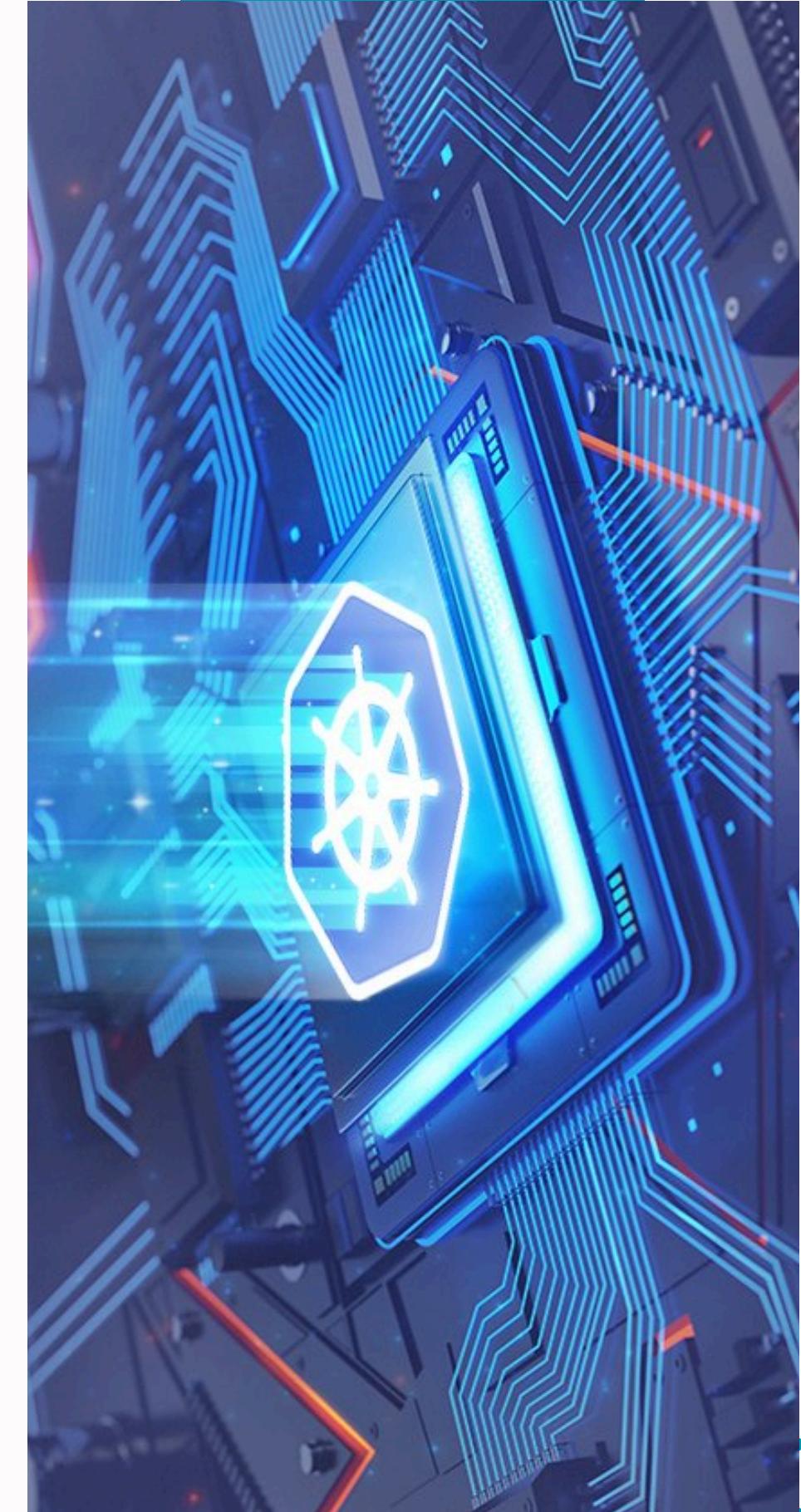
BUILDING A KUBERNETES CLOUD SECURITY INFRASTRUCTURE



Present by: AnonyMeow



August, 30 2024



Content

01

Project Progress

02

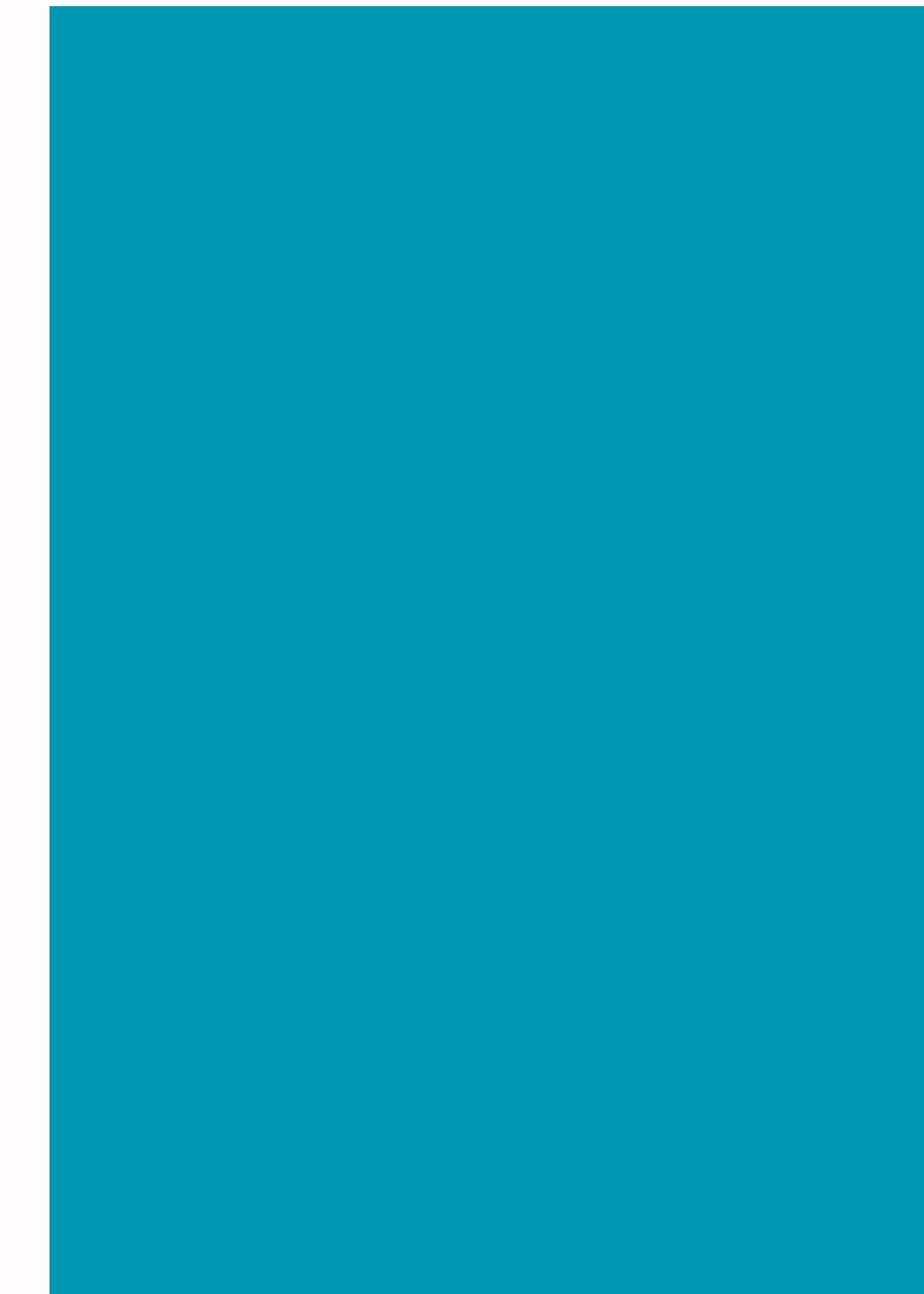
Security Implementation

03

Project Results

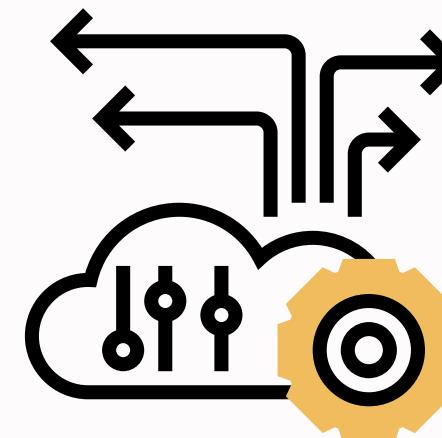
04

Comment for Project



Mid-project Progress

Application Deployment

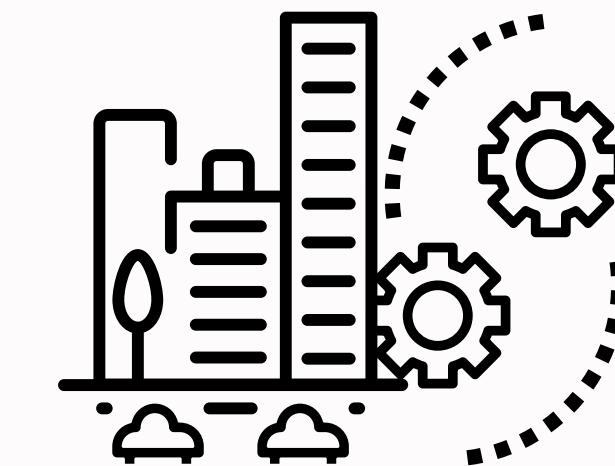


Sample application has been successfully deployed to the Kubernetes cluster.

Kubernetes Infrastructure

Kubernetes infrastructure have been installed and setup

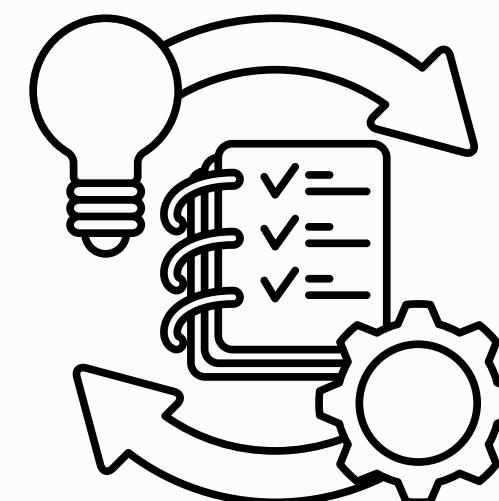
- Helm
- Cert-Manager
- Load Balancer
- NGINX Ingress Controller



Security Implementation

Kubernetes infrastructure have been installed and setup

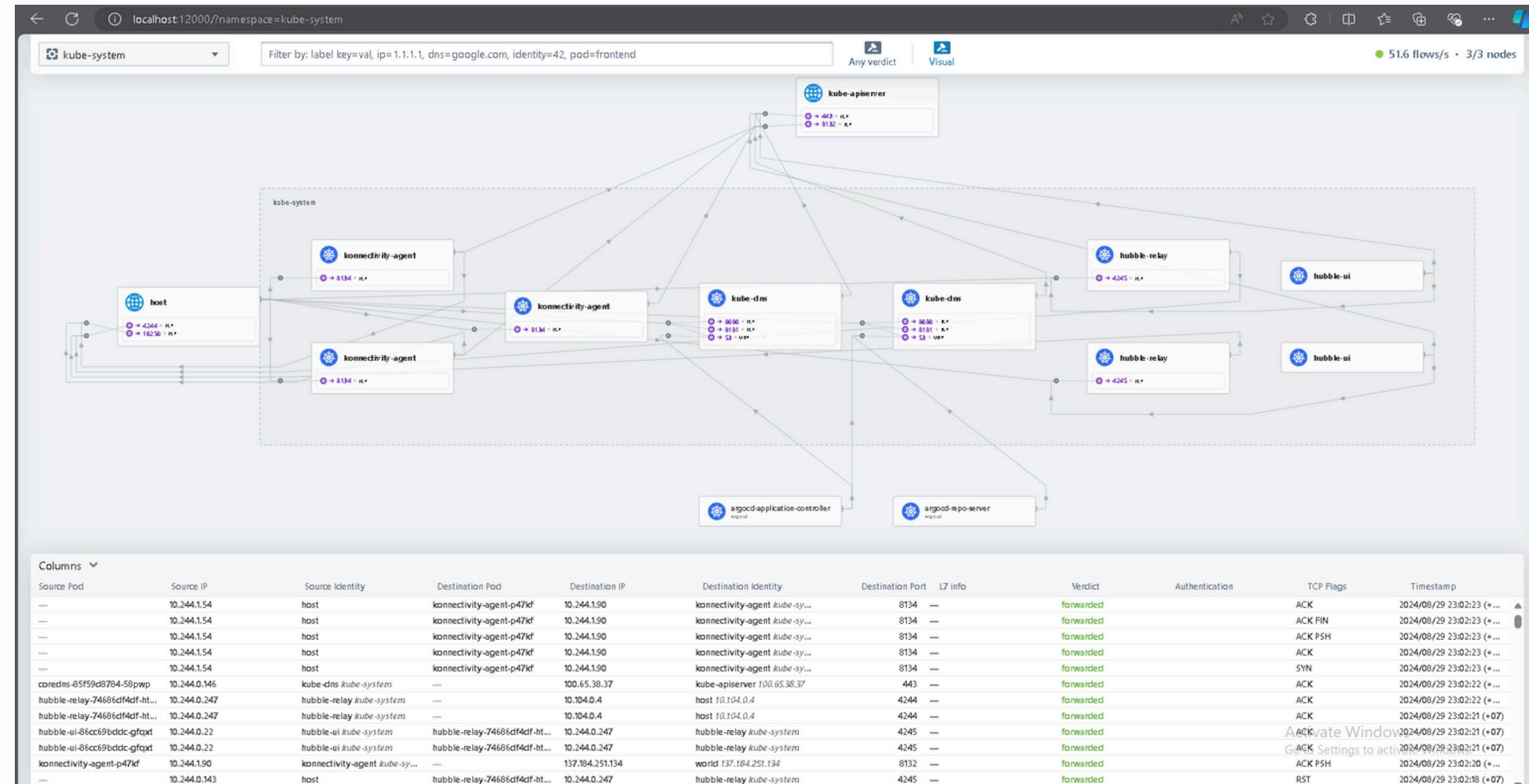
- Container Runtime Security
- Encryption in Transit
- NGINX Ingress Configuration
- Kubernetes Secrets



Security Implementation

Securing Network

- Used **VPC** on DigitalOcean to create an **isolated** network environment
- Implemented **Cilium** and **Hubble** to enhance network security and provide deep observability
- **Cilium** enhances the security posture of our Kubernetes cluster, ensuring that only authorized services can communicate with each other
- **Hubble** allows us to monitor and observe network traffic in real time, helping us to identify potential security issues and optimize network performance



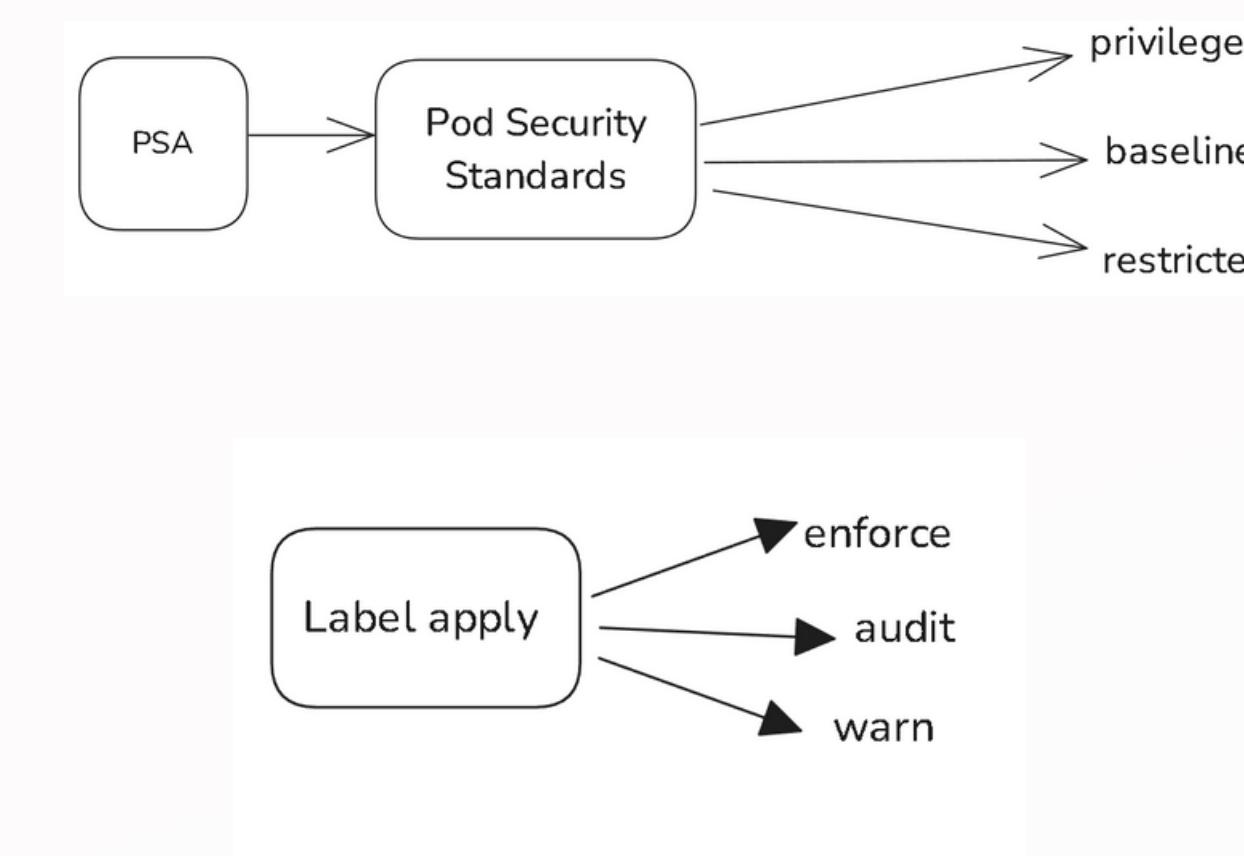
Security Implementation

Prevent Pod Privilege

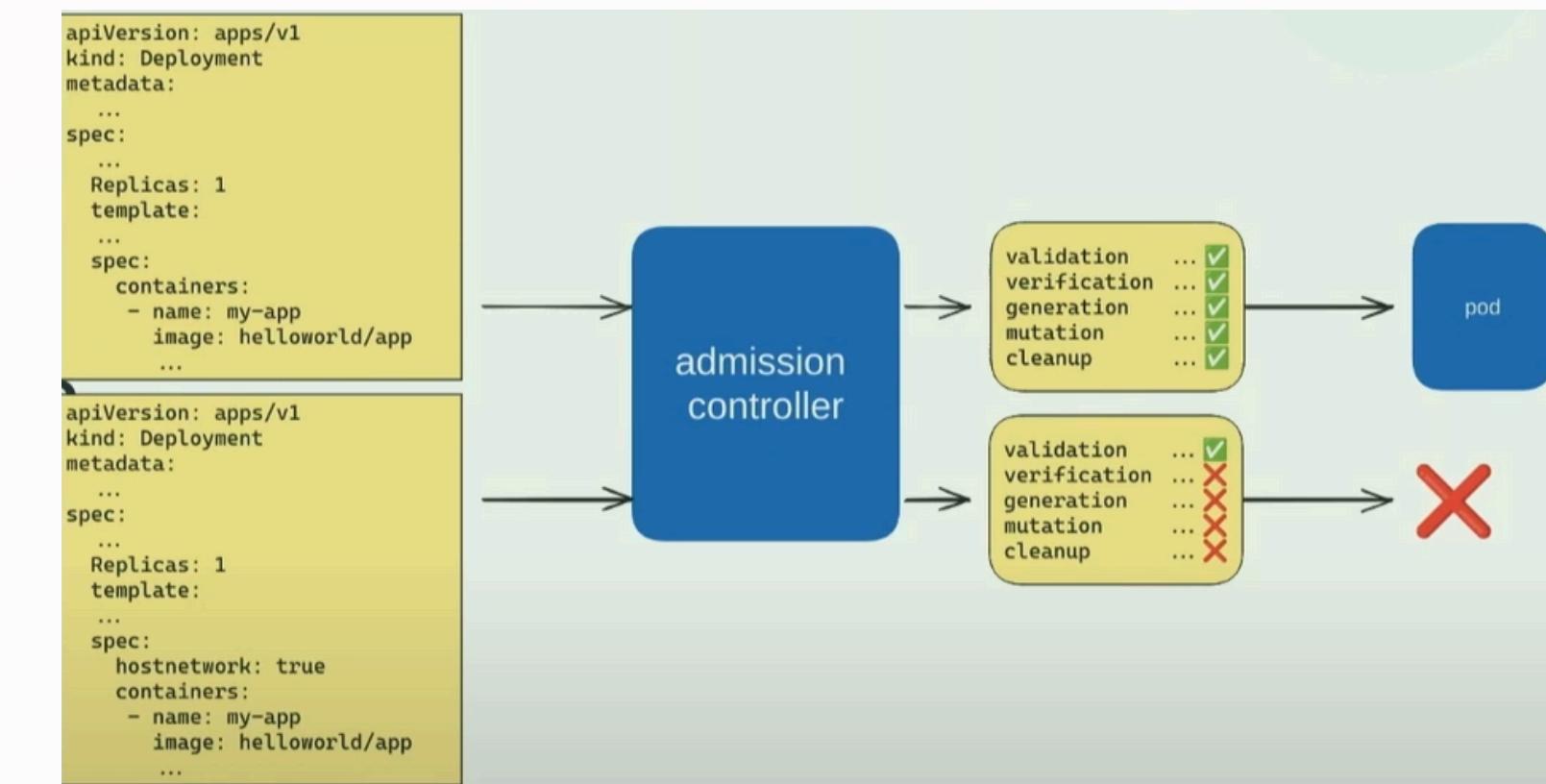
```
securityContext:  
    runAsNonRoot: true  # Ensure non-root user  
    runAsUser: 1000  
    runAsGroup: 3000  
    fsGroup: 2000  
    allowPrivilegeEscalation: false  
    readOnlyRootFilesystem: true  
    seccompProfile:          # Adding seccompProfile  
        type: RuntimeDefault  
    capabilities:  
        drop:  
        - ALL
```

Security Implementation

Prevent Pod Privilege (Cont.)



Pod security admission



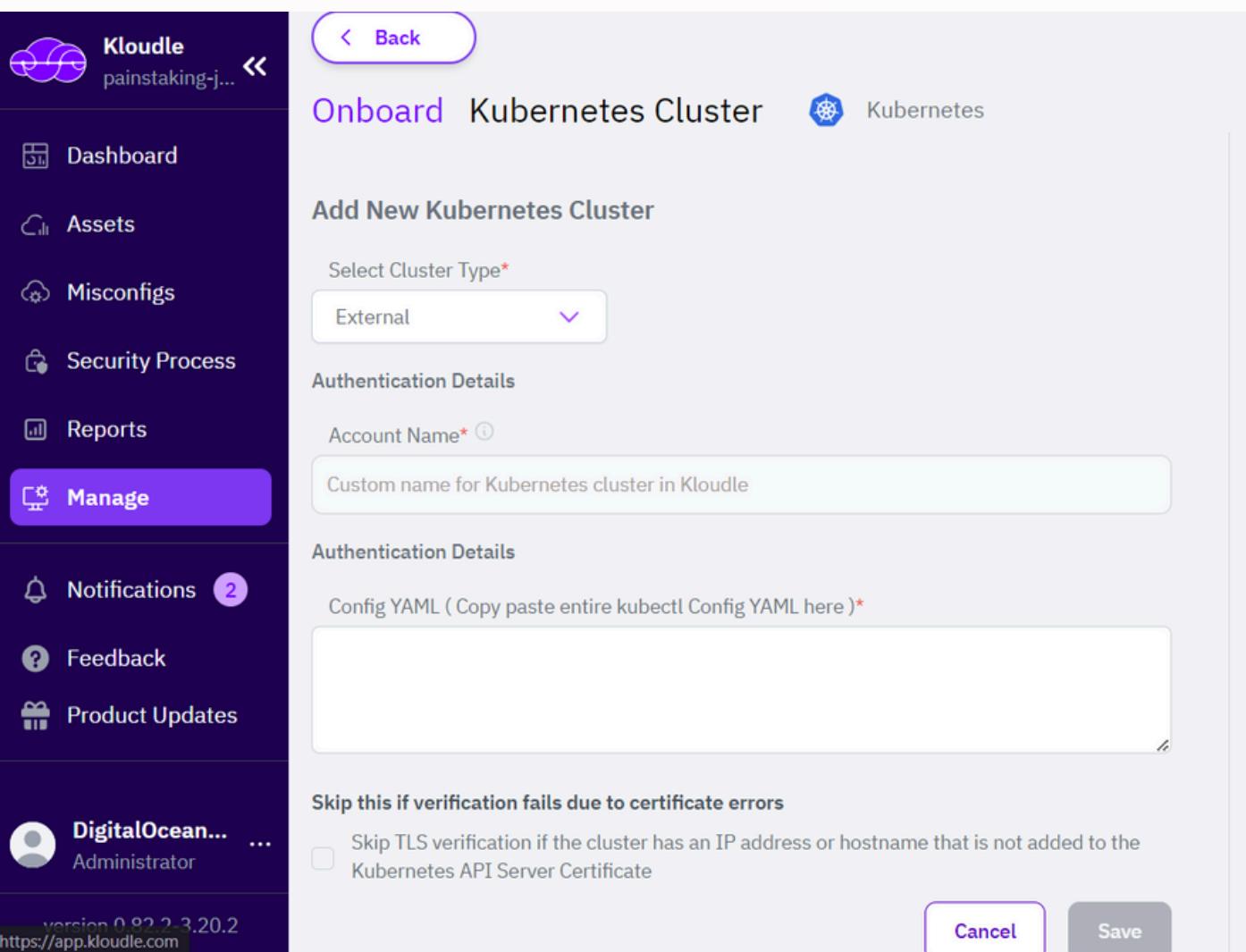
```
kubectl label --overwrite ns argocd pod-security.kubernetes.io/enforce=restricted
kubectl label --overwrite ns cert-manager pod-security.kubernetes.io/enforce=restricted
kubectl label --overwrite ns flask-app pod-security.kubernetes.io/enforce=restricted
kubectl label --overwrite ns ingress-nginx pod-security.kubernetes.io/enforce=restricted
```

Security Implementation

Cloud Security Scanning(Kloudle)

Kloudle is a cloud security tool designed to help organizations identify and manage security risks across their cloud environments.

To ensure our Kubernetes infrastructure, we integrated Kloudle as part of our security testing process. Kloudle is a cloud-native security tool that helps in identifying and mitigating security risks across cloud environments (DigitalOcean), including Kubernetes.



Follow these easy steps to onboard Kubernetes Cloud account

Prerequisites

1. A kubernetes administrator or user with the ability to create resources at cluster level, is required to run the shell script as it invokes `kubectl` with your saved user credentials.
2. Ensure your `kubeconfig` cluster context is set correctly since the script creates resources in the current context. You can verify this using `kubectl cluster-info`.

Instructions

These instructions are for an **external** cluster with a public IP address.

Step 1. Ensure you are in the right target cluster by running the following on terminal

```
kubectl cluster-info
```

Step 2. Run the following command to execute the shell script

```
curl -sS https://raw.githubusercontent.com/Kloudle/kloudle-kubernetes-onboarding/master/kubernetes-readonly-admin-creator.sh | bash
```

The script creates the following:

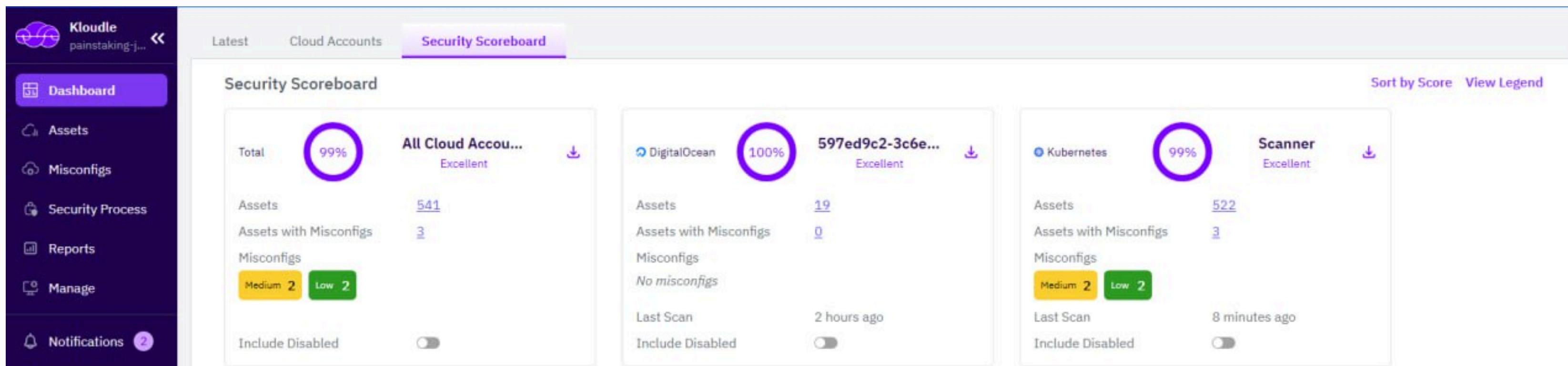
- 2.1. **Read Only ClusterRole**
- 2.2. **ClusterRoleBinding** for the ClusterRole
- 2.3. **Service Account**
- 2.4. **Secret Token** for the Service Account

Step 3. Add the printed kubeconfig to Kloudle from the bash prompt

Project Result

Security Overview for Kubernetes:

- Score: The Kubernetes cluster received a security score of 99%.
- Misconfigurations: The scan detected 3 assets:
 - Medium Severity: 2 misconfigurations.
 - Low Severity: 2 misconfigurations



What have we Learned from this project?

- Solid understanding of Kubernetes architecture and core components.
- Focused on container orchestration, resource management, and security.
- Explored RBAC, network policies, and cloud security in Kubernetes.
- Emphasized container security and policy enforcement tools.
- Learned automation through CI/CD pipelines and IaC.
- Recognized the importance of monitoring and auditing for compliance.

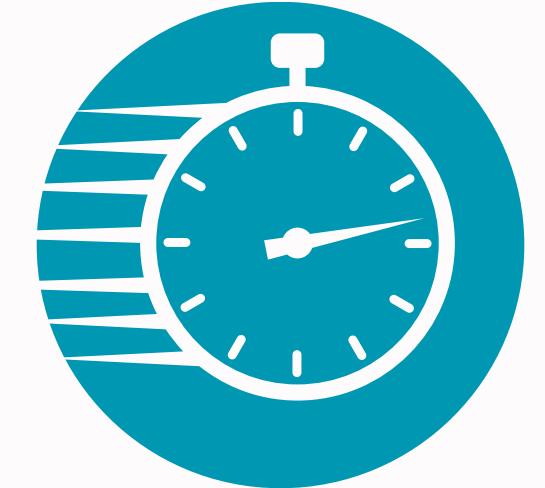
What skills did we experience in this project?

- Improved Kubernetes administration by deploying, managing, and scaling clusters.
- Configured security settings and networking policies.
- Implemented cloud security best practices and secured Kubernetes components.
- Enhanced Docker and container security knowledge.
- Gained hands-on experience with CI/CD and security automation.
- Strengthened knowledge of Kubernetes security.

Difficulty in Understanding the Concept and Implementation



Complex Technology Integration





**THANK
YOU**