Hewlett Packard Enterprise

HPC Security Dashboard

VIT, Vellore

INVESTIGATION REPORT – CIS Compliance and Vulnerabilities

HPE Mentor : Rao, Yarlagadda Srinivasa (Vasu)

VIT Mentor : Ruby D

Team Members : Arka Pramanik, Nishanth VM, Alok N, Swayam, Keshav

Submitted by

Arka Pramanik

20BCE0447

HPE CTY Intern

Overview

Docker and Containerized Applications

Docker is an open-source platform widely used for building, shipping, and running distributed applications within containers. Containers offer a lightweight method of packaging software, including all the necessary dependencies, system tools, and settings required for the application to run. By encapsulating applications within containers, developers can ensure consistent and portable deployment across different environments, from development to production and across various infrastructure types.

Containerized applications are those packaged and executed within containers. This approach enables developers to create self-contained units that can be easily deployed on any infrastructure without modification. Containerization streamlines application management, simplifies deployment processes, enhances resource utilization, and improves flexibility and portability by facilitating easy movement of containers between different hosts or cloud providers.

Kubernetes:

Kubernetes is an open-source container orchestration platform initially developed by Google and now maintained by the Cloud Native Computing Foundation (CNCF). It automates the deployment, scaling, and management of containerized applications across clusters of computers. Kubernetes provides a robust framework for efficiently managing containers at scale by automating various tasks, such as scaling, load balancing, and networking. This enables seamless application deployment, management, and scaling across different environments, whether on public clouds, private clouds, or on-premise data centers.

High Performance Computing (HPC):

High Performance Computing refers to the use of parallel processing and powerful computing resources, such as supercomputers or clusters of computing nodes, to perform complex computational tasks at high speeds. HPC systems are employed in diverse scientific and engineering domains,

including weather forecasting, molecular modeling, financial modeling, and more. As the volume of data generated by research and the demand for computational power have increased, HPC has become crucial in accelerating scientific discoveries and solving complex problems.

Aim

The aim of my project is to improve security in Kubernetes clusters by focusing on compliance and vulnerability management. It involves identifying and providing remediation for compliance issues and vulnerabilities within the cluster, ensuring a secure and resilient environment for containerized applications.

Tool Selection and Rationale

After conducting an exhaustive analysis of multiple open-source security tools, we have arrived at the conclusion that Aqua Security offers industry leading tools.
Kube-bench is widely recognized as a leading tool for assessing Kubernetes clusters against CIS benchmarks. Its comprehensive checks and adherence to industry best practices make it an ideal choice for ensuring compliance in the Kubernetes environment.
Trivy, on the other hand is a powerful vulnerability scanner for container images, provides extensive coverage for identifying vulnerabilities in dependencies and helps in maintaining a secure container ecosystem.
Furthermore, both tools are backed by Aqua Security, a prominent industry leader renowned for its expertise in container security. Leveraging the experience and support of Aqua Security further enhances the credibility and reliability of the chosen tools.
By combining kube-bench and Trivy, this project ensures robust compliance management and vulnerability assessment, enabling administrators to proactively mitigate security risks and maintain a secure Kubernetes cluster environment.

While both Kube-Bench and Trivy are powerful tools for assessing security in Kubernetes clusters and container images, they lack native visualization capabilities. This project bridges that gap by integrating Trivy and Kube-Bench

with Grafana, enabling the visualization of their metrics in a consolidated and easily understandable dashboard.

Key Features of the HPC Security Dashboard

The HPC Security Dashboard provides the following features:

- **Automated CIS Benchmark Assessment:** Kube-Bench & Trivy is utilized to perform automated CIS benchmark assessments on the Kubernetes cluster, evaluating its security configuration against industry best practices.

- **Container Image Vulnerability Scanning:** Trivy scans container images for known vulnerabilities and provides a detailed report on any security issues found.

- **Kube Automate Solution:** Trivy and Kube-Bench metrics are collected and exposed by Prometheus, a monitoring and alerting toolkit, and a Node Server pod which serves the metrics report enabling efficient data collection for visualization.

- **Visualization in Grafana:** The collected metrics from Trivy and Kube-Bench are visualized in a Grafana dashboard, providing a consolidated view of the security posture of the Kubernetes cluster. This allows for easy identification of vulnerabilities and misconfigurations, aiding in proactive security management.

With this enhanced HPC Security Dashboard, administrators and security teams can gain valuable insights into the security of their Kubernetes clusters, making informed decisions and taking proactive measures to strengthen the overall security posture.

Enhanced Logging and Visualization

The HPC Security Dashboard project provides a centralized platform for logging and visualization of security metrics. It collects data from Kube Bench and Trivy, stores it in Prometheus, and then visualizes the metrics through Grafana. This approach offers a range of benefits, including:

- **Comprehensive Security Insights:** The integration of Kube Bench and Trivy metrics allows users to obtain a holistic view of their Kubernetes cluster's security. They can identify vulnerabilities, misconfigurations, and container-level risks, all in one place.

- **Real-time Monitoring:** The dashboard provides real-time monitoring of security metrics, enabling users to detect and respond to security incidents promptly.

- **Intuitive Visualization:** Grafana's rich visualization capabilities allow users to create custom dashboards, charts, and graphs to better understand the security metrics. This visual representation enhances the interpretability and analysis of the data.

- **Centralized Platform:** By consolidating security metrics from Kube Bench and Trivy in a single dashboard, the HPC Security Dashboard simplifies the monitoring and management of Kubernetes cluster security.

Project Components

The HPC Security Dashboard project consists of the following components:

Trivy Operator:

Trivy Operator is a tool used for vulnerability scanning in containerized environments. It analyzes container images and identifies known vulnerabilities in their dependencies. Trivy Operator is integrated into the project to provide container image vulnerability scanning capabilities.

Kube Bench:
Kube Bench is a tool that evaluates the security configuration of a Kubernetes cluster by running the CIS Kubernetes Benchmark tests. It checks various aspects of the cluster's security and generates a detailed report highlighting any vulnerabilities or misconfigurations.

Prometheus:
Prometheus is a monitoring and alerting toolkit used to collect and store metrics from various sources. In the HPC Security Dashboard project, Prometheus is utilized to collect and store the metrics generated by Trivy and Kube Bench.

Kube Automate - custom built solution:
A Kubernetes pod is created to serve as a Node.js server. This pod accepts the metrics report from Kube Bench and Trivy and makes it available through designated output ports. It ensures efficient retrieval and updating of metrics reports.

Grafana:
Grafana is a popular open-source platform used for data visualization and monitoring. In this project, Grafana is employed to create a consolidated dashboard that visualizes the metrics collected from Trivy and Kube Bench. It provides a user-friendly interface for administrators and security teams to analyze the security metrics of the Kubernetes cluster.

Installation
The installation process for the HPC Security Dashboard project involves the following steps:

Cloning the Project Repository:
Start by cloning the project repository to your local machine using the command:

```
git clone
https://github.com/AP-XD/HPE-CTY-HPC-Security-Dashboard
```

MicroK8s Installation:

MicroK8s is installed as the Kubernetes environment for this project.

1. Install MicroK8s on your machine using the following commands:

```
sudo zypper addrepo --refresh \
https://download.opensuse.org/repositories/system:/snappy/open
SUSE_Tumbleweed\snappy
sudo zypper --gpg-auto-import-keys refresh
sudo zypper dup --from snappy
sudo zypper install snapd
sudo systemctl enable --now snapd
sudo systemctl enable --now snapd.apparmor
sudo snap install microk8s --classic --channel=1.27
sudo usermod -a -G microk8s $USER
sudo chown -f -R $USER ~/.kube
su - $USER
alias k="microk8s kubectl"
```

2. Start and enable MicroK8s using the following commands:

```
sudo -i
microk8s reset
exit
sudo iptables -P FORWARD ACCEPT
microk8s start
microk8s enable dns
k get ns
```

3. (Optional)Install k9s in OpenSUSE for monitoring and easy access of pods status

```
sudo zypper in k9s
```

Configuration:

Configure the `prom-values.yaml` and `trivy-values.yaml` files according to the provided content or modify them as per your requirements.

prom-values.yaml

```
prometheus:
prometheusSpec:
serviceMonitorSelectorNilUsesHelmValues: false
serviceMonitorSelector: {}
serviceMonitorNamespaceSelector: {}
```

trivy-values.yaml

```
serviceMonitor:
# enabled determines whether a serviceMonitor should be deployed
enabled: true
trivy:
# ignoreUnfixed is the flag to show only fixed vulnerabilities in
# vulnerabilities reported by Trivy. Set to true to enable it.
ignoreUnfixed: true
operator:
# metricsVulnIdEnabled is the flag to enable metrics about cve vulns id
# be aware of metrics cardinality is significantly increased with this
feature enabled.
metricsVulnIdEnabled: true
compliance:
# cron: this flag controls the cron interval for compliance report
generation
cron: 0 */1 * * *
# reportType: this flag controls the type of report generated (summary
or all)
reportType: all
```

Executing the Scripts:

1. Make the scripts executable using the commands:

```
chmod +x ./scripts/update_deploy.sh
chmod +x ./metrics_update.sh
```

2. Execute the `update_deploy.sh` script to install Trivy Operator and Kube Prometheus Stack Helm Chart automatically. Use the `metrics_update.sh` script to set up a CRON job for hourly metrics update.

3. Setup CRON JOB for auto updating the metrics hourly using the metrics_update.sh script and save logs in a file

```
{ crontab -l; echo "0 * * * * $(pwd)/metrics_update.sh
<namespace-name> >> $(pwd)/cron-log.txt "; } | crontab -
```

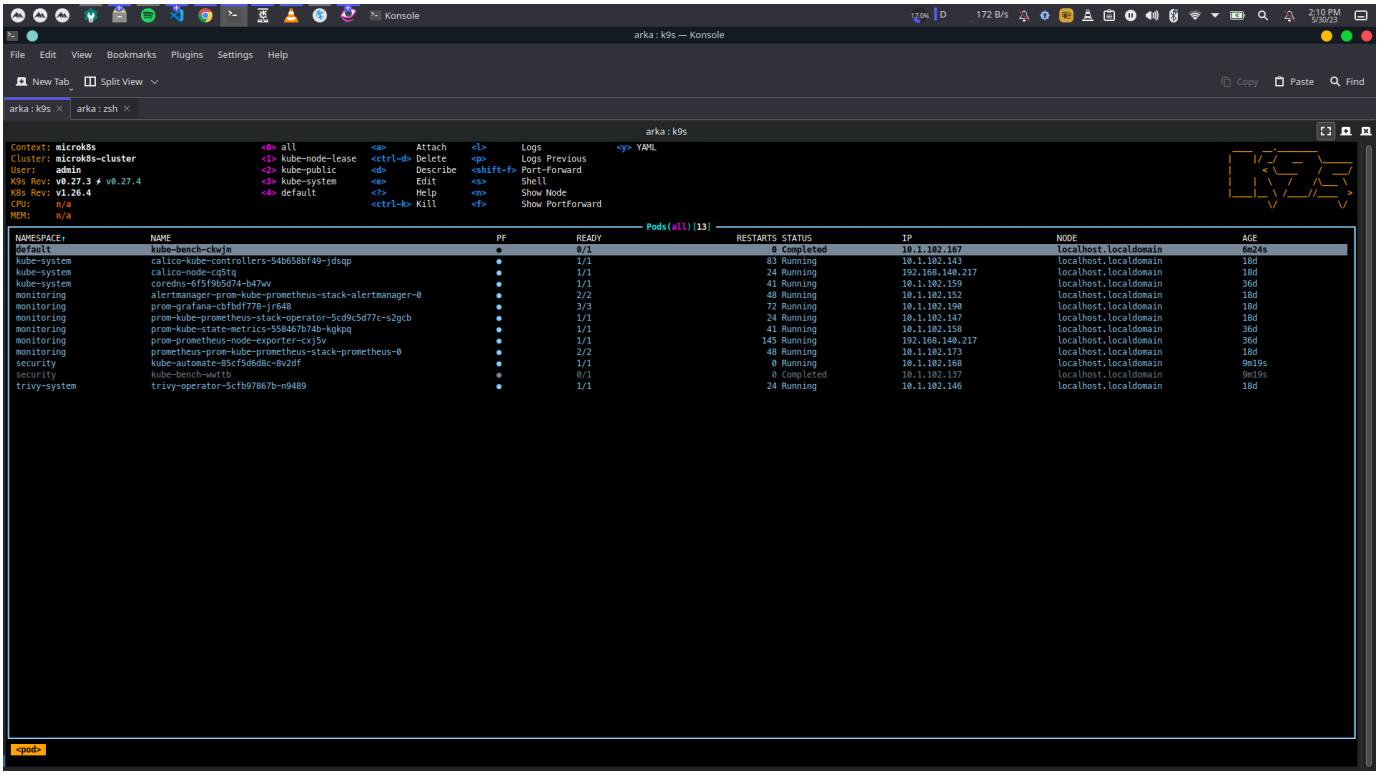
If you don't want to save logs of the cronjob and only view them via mail, use the following command:

```
{ crontab -l; echo "0 * * * * $(pwd)/metrics_update.sh
<namespace-name>"; } | crontab -
mail # Press ENTER to view mail from cronjob
```

Setup Port Forwarding:

1. Run k9s

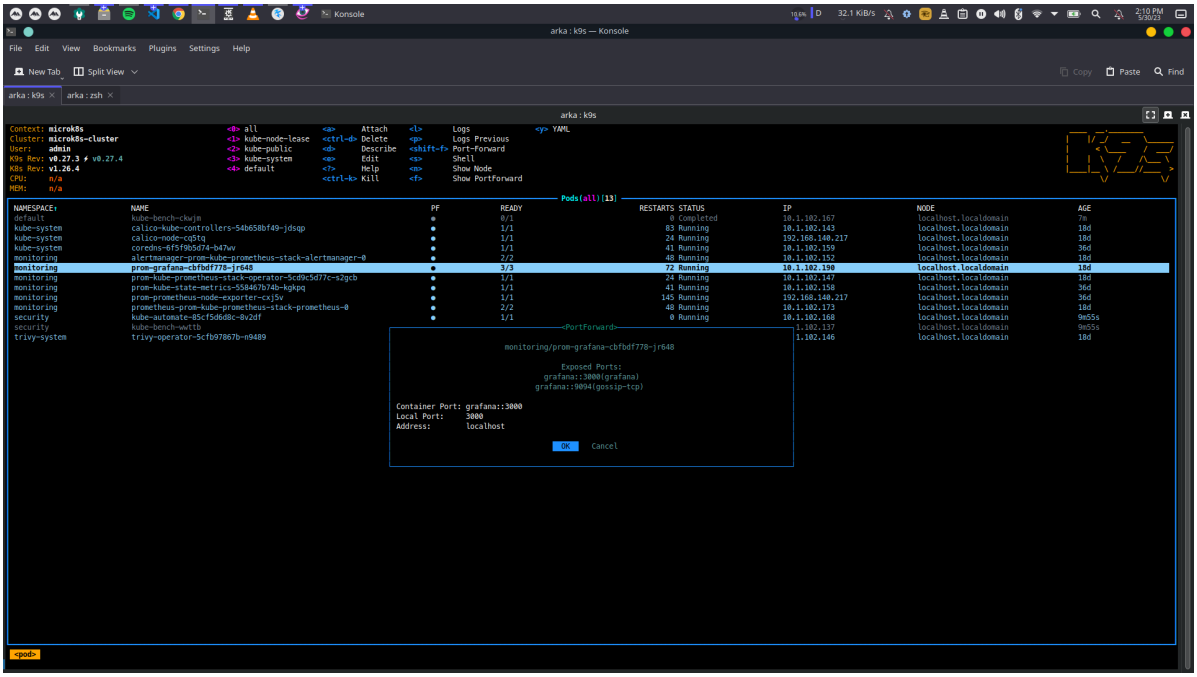
```
./k9s
```



2. Enable port forwarding for the following pods by pressing Shift+F and Ok on each pod to confirm port forwarding

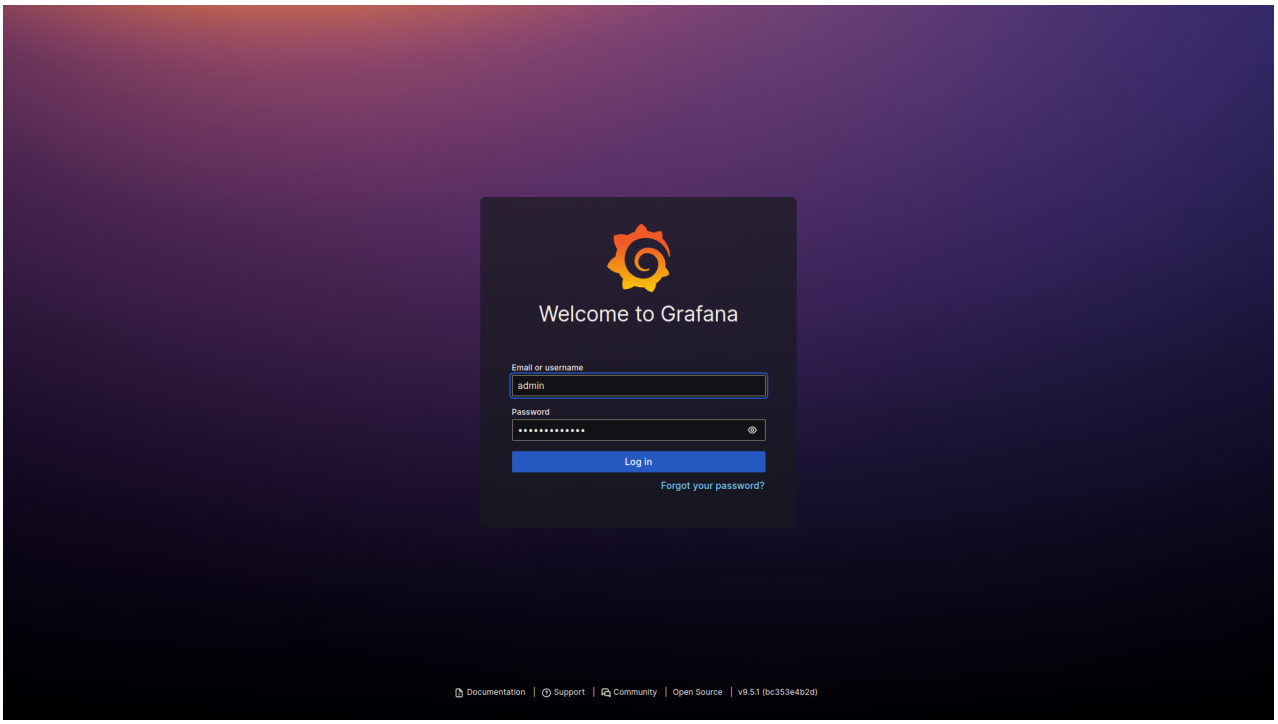
- `prom-grafana pod`
- `prometheus-prom-kube-prometheus-stack-prometheus pod`
- `trivy-operator pod`

Sample:

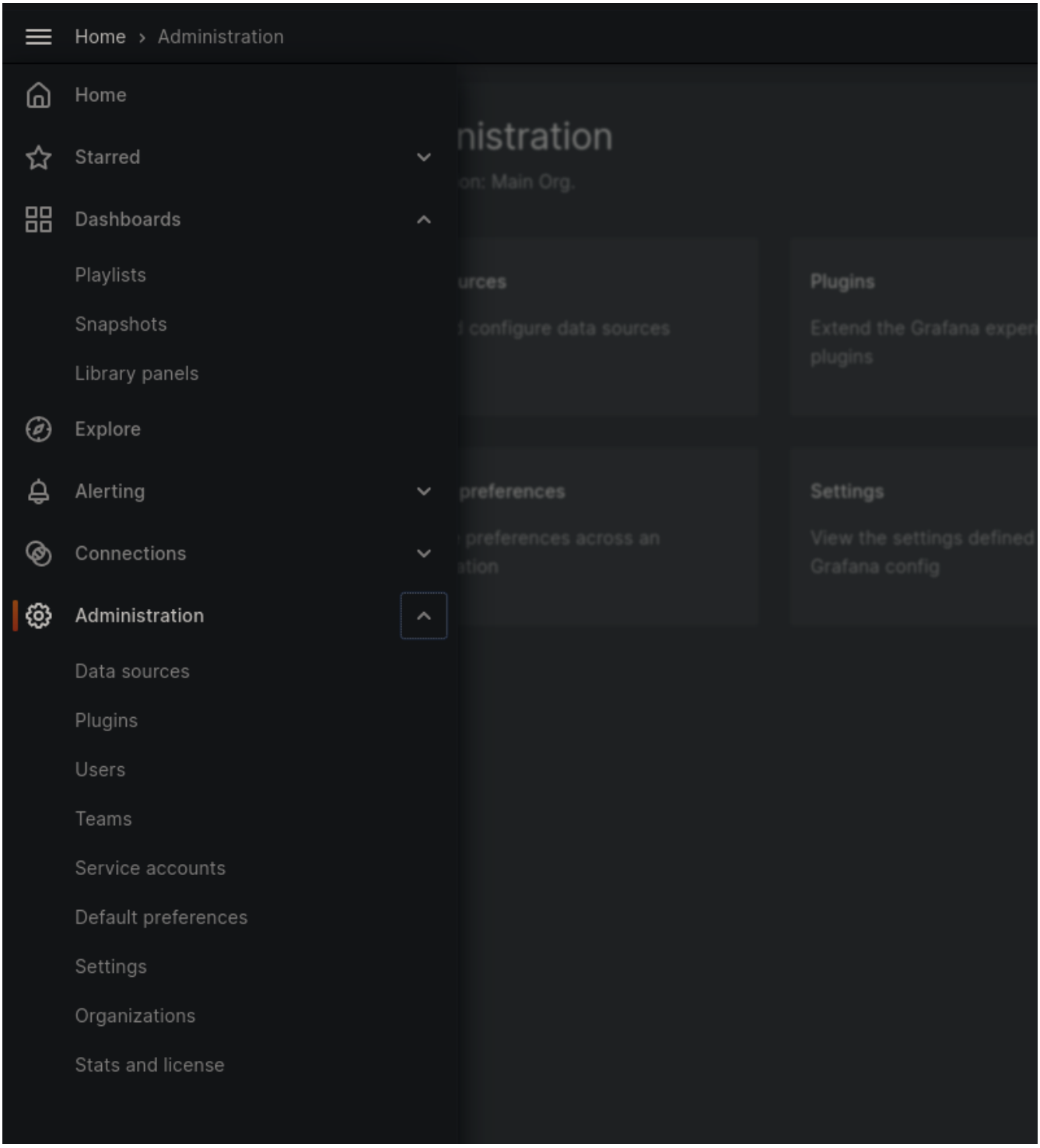


Configure Datasources in Grafana:

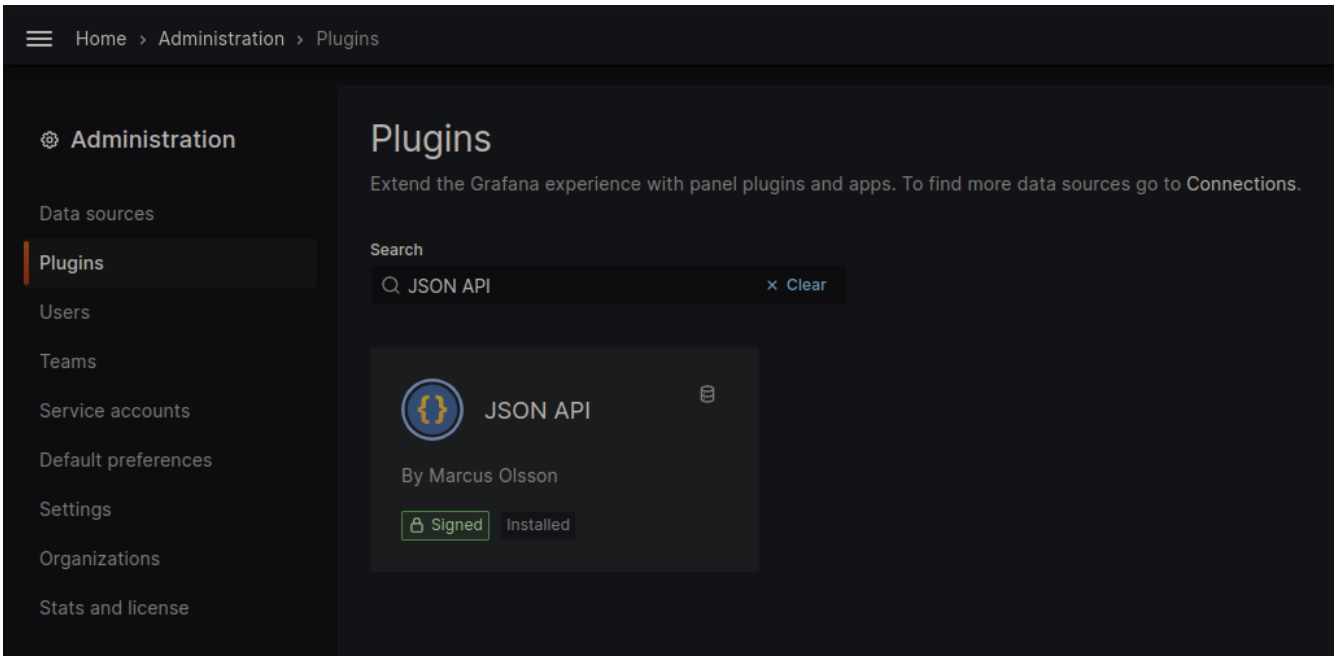
1. Access Grafana web UI on http://localhost:3000/ and enter admin credentials



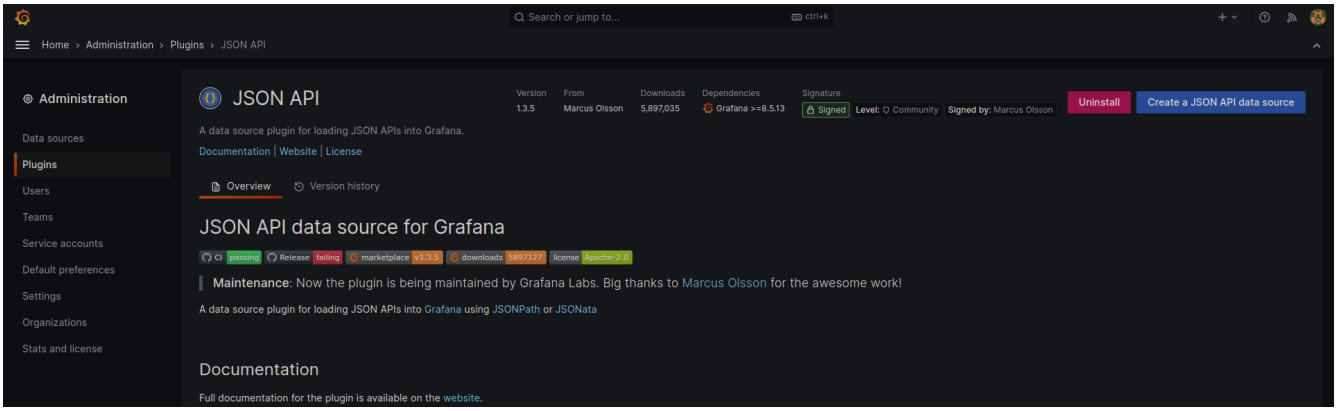
2. Plugin activation On the top left corner, click on hamburger icon and navigate to Administration > Plugins



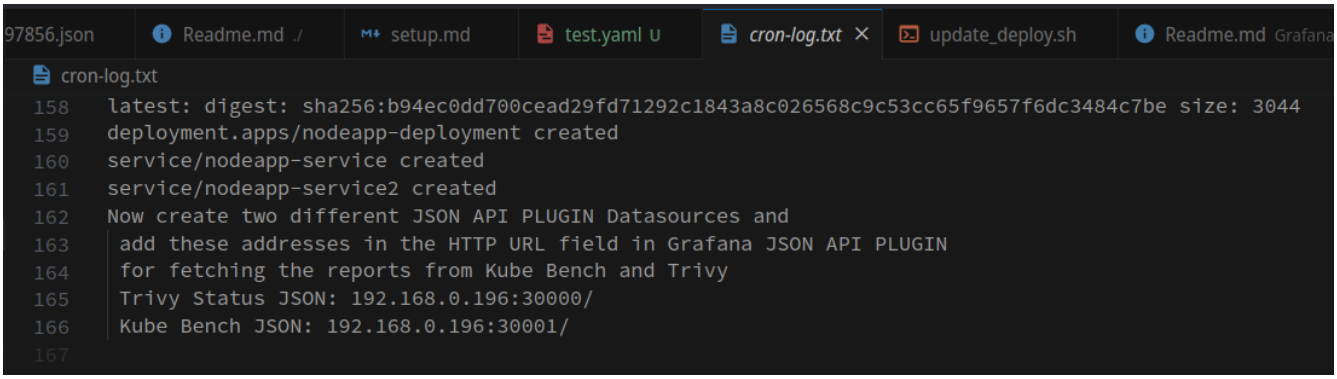
Now in the search bar, enter json and click 'JSON API' and install it.



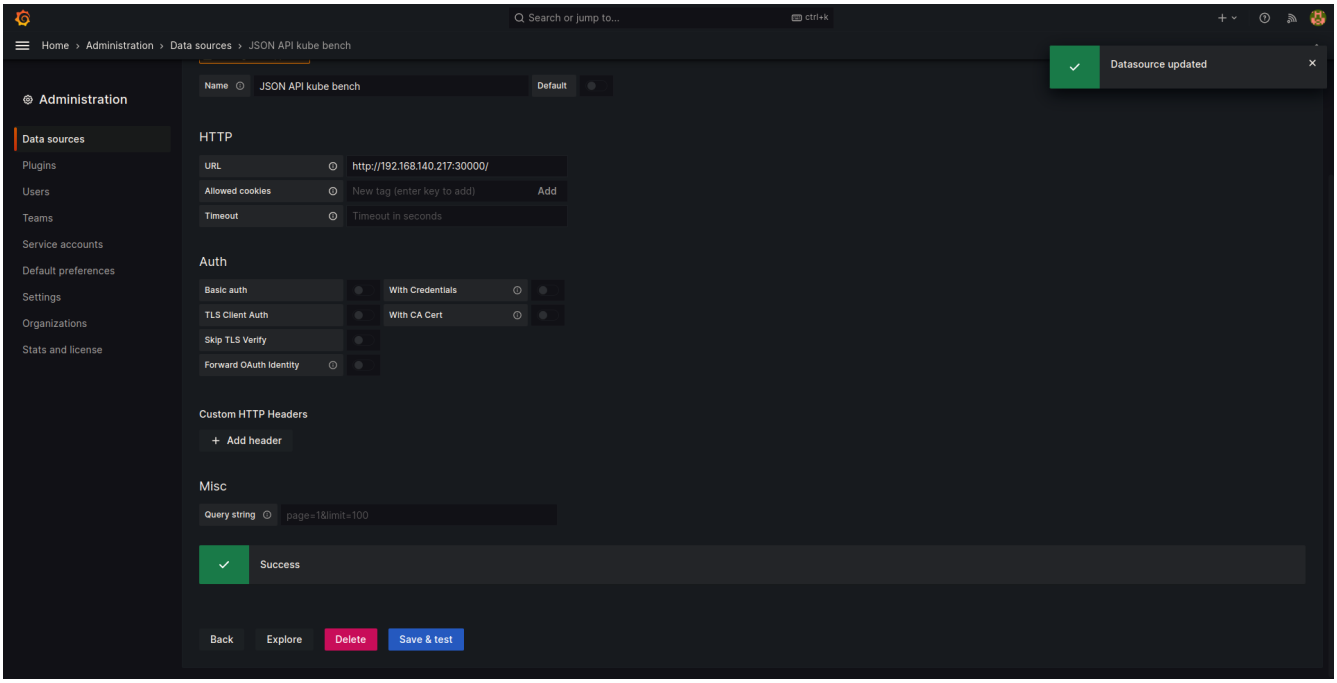
Click 'Create a JSON API Data Source'.



Get the URL from the cron-log.txt and now in the HTTP > URL field, enter node-ip:port. Configure the datasource for both Kube-Bench and Trivy to fetch data from <node-ip>:30000 and <node-ip>:30001.

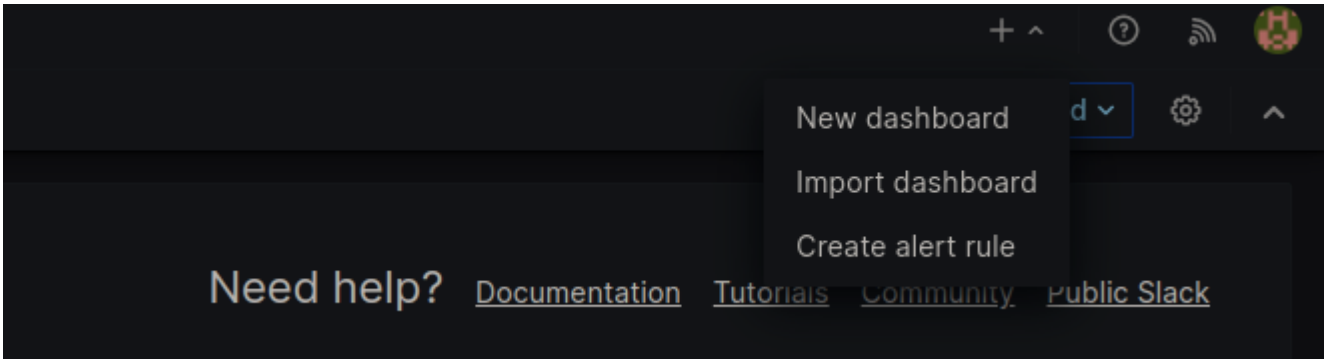


Click save and test. If you get a success message then the plugin is configured properly.

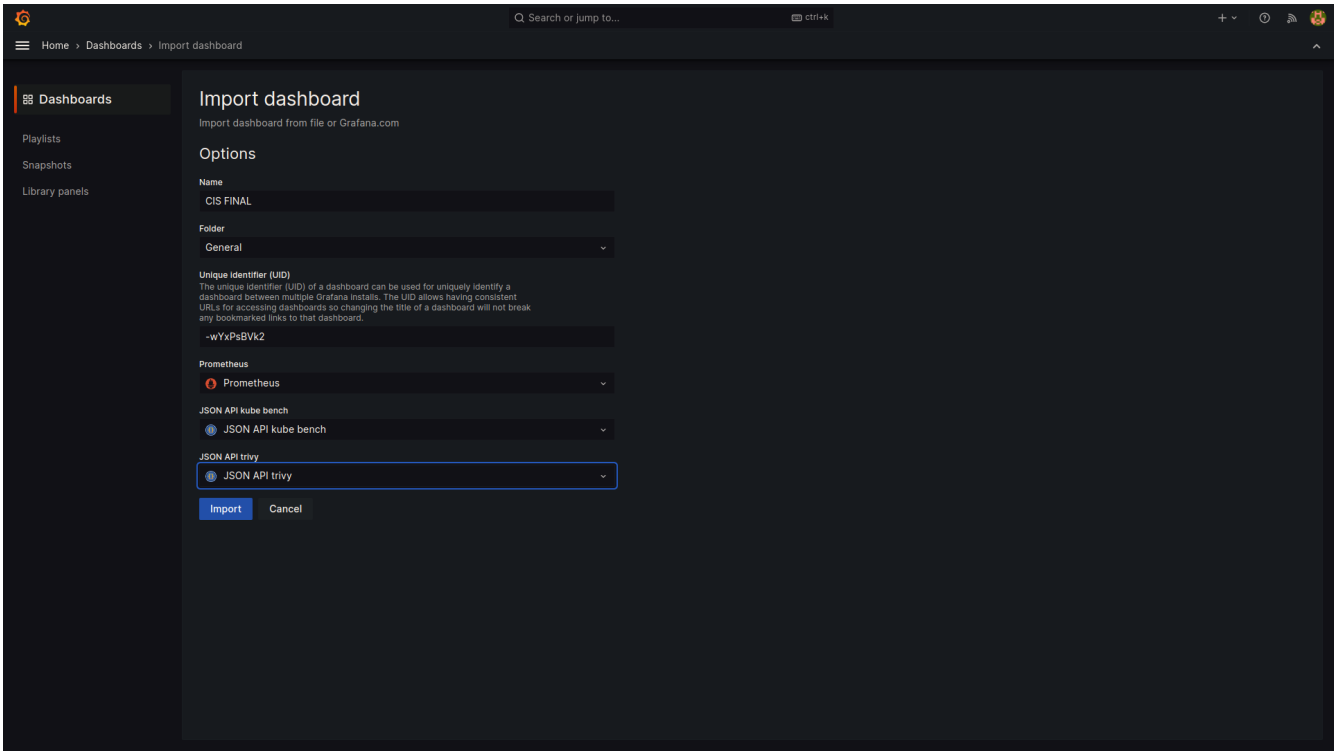


Import the Dashboard

Click on the + symbol on the upper right corner and select Import dashboard from there



Now a window will appear like this, prompting you to import a json file. Import this file and select the datasources appropriately and click import.



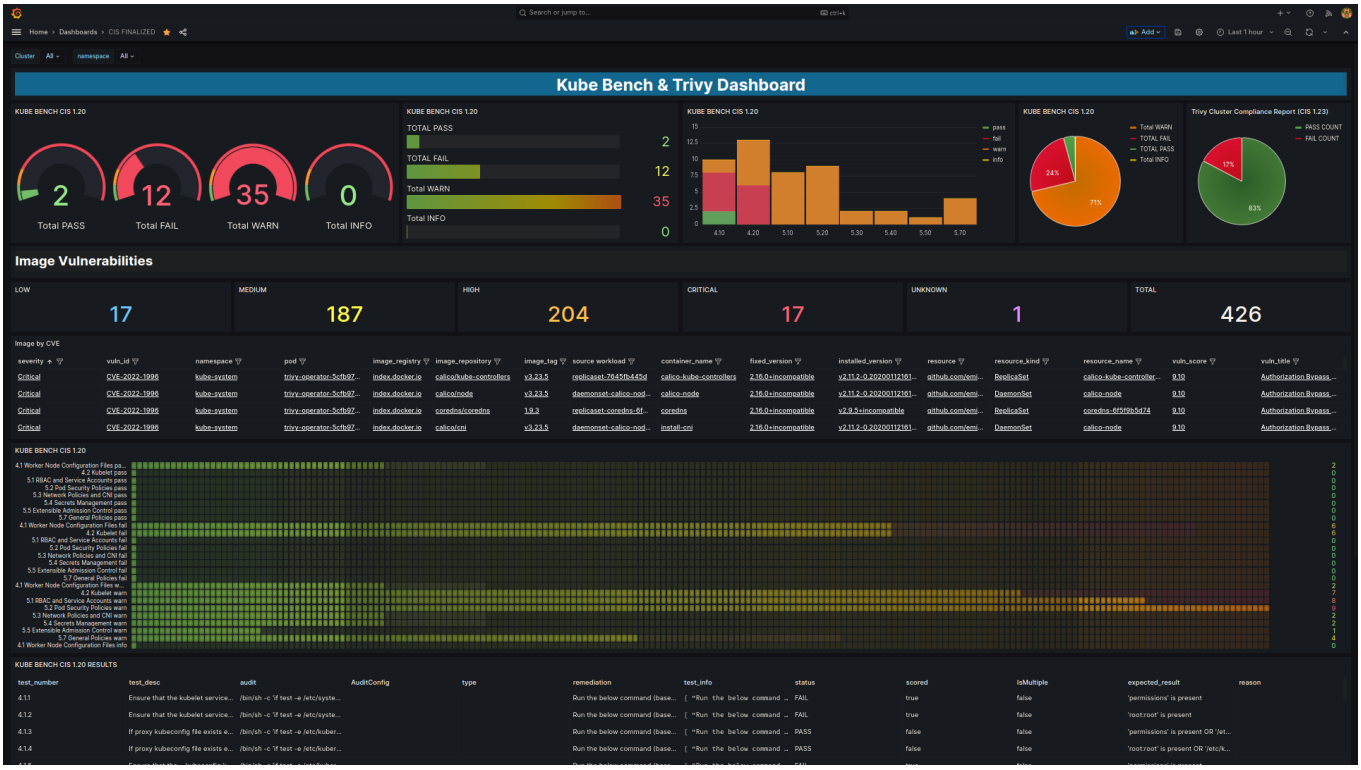
Enhanced Logging and Visualization:

Once the installation and configuration steps are completed, the HPC Security Dashboard will provide enhanced logging and visualization of Kubernetes security metrics. The consolidated Grafana dashboard will display various metrics, including vulnerability scan results from Trivy, security configuration checks from Kube Bench, and other relevant information.

Troubleshooting

If any issues are encountered during the installation or configuration process, refer to the troubleshooting steps provided in the project documentation. These steps include troubleshooting common problems related to firewall issues, port conflicts, and misconfigurations.

Final Result



Conclusion

The HPC Security Dashboard project successfully integrates Trivy Operator, Kube Bench, Prometheus, and Grafana to enhance the logging and visualization of Kubernetes security metrics. By following the provided installation steps and configuring the necessary components, administrators and security teams can access a consolidated Grafana dashboard that provides comprehensive monitoring and analysis of the security posture of their Kubernetes clusters.

We extend our heartfelt thanks to the mentors from VIT and HPE for their invaluable guidance and the wonderful opportunity they have provided us to work on this project. Their expertise and support have been instrumental in our success, and we are grateful for their mentorship and contributions throughout the project.