**Kubernetes Cluster Scanner Tool**

**Introduction**

The Kubernetes Cluster Scanner Tool is a utility designed to perform security scans on Kubernetes deployments using various security tools. This tool automates the scanning process across different namespaces, identifying vulnerabilities and security risks in your Kubernetes environment.

**Components**

**Docker Image (mariamosama/cluster-scanner:v1)**

The Docker image serves as the foundation for the scanning tool. It includes all the required tools and scripts for conducting security scans on Kubernetes deployments.

**scan.sh Script**

The scan.sh shell script is the core of the tool. It orchestrates the scanning process by performing the following steps:

1. Checks for the presence of kubectl and installs it if not found.
2. Iterates through all namespaces in the cluster.
3. For each deployment within a namespace:

* Utilizes trivy to scan the Docker image for vulnerabilities and saves the results to a JSON file.
* Executes kube-bench to assess Kubernetes security benchmarks and saves the results.
* Runs kube-hunter to identify potential security risks and saves the results.

The results are organized in a directory structure that includes the namespace, deployment name, timestamp, and the tool's name.

1. Logs the progress and completion of each scan.

**CronJob (cluster-scanner-cronjob.yaml)**

The Kubernetes CronJob resource automates the execution of the scanning tool at scheduled intervals, ensuring regular security scans are performed. It employs the Docker image and the scan.sh script to execute the scanning tasks.

**Persistent Storage for Scan Results**

The Kubernetes Cluster Scanner Tool supports persistent storage for scan results, allowing you to access and retain scan data beyond the lifecycle of individual scanning pods. This feature utilizes Longhorn and Persistent Volume Claims (PVCs) and can be accessed through a result access pod.

**Usage**

To use the Kubernetes Cluster Scanner Tool, follow these steps:

1. Deploy the CronJob resource using the provided cluster-scanner-cronjob.yaml file. This schedules the scanning tool to run at specified intervals.
2. Verify that the Docker image (mariamosama/cluster-scanner:v1) contains the required tools (trivy, kube-bench, kube-hunter) and they are properly configured.
3. Monitor the logs of the CronJob or individual pods to observe the progress and completion of the scanning tasks.
4. The scan results are stored in the /app/scan-results directory within the container. The directory structure includes the namespace, deployment name, timestamp, and the name of the scanning tool. This could be accessed by the “result-access-pod”.

**Conclusion**

The Kubernetes Cluster Scanner Tool is a valuable addition to your Kubernetes security toolkit, assisting in the identification of vulnerabilities and security risks in your deployments. By automating the scanning process, this tool contributes to maintaining the security posture of your Kubernetes environment.