Lab12 - Configuring and Using Audit Logging

The DevOps team you are working on decides to keep track of the events occurring in your Kubernetes cluster. You have been tasked to configure audit logging for specific events.

1. Create the audit policy file named `/etc/kubernetes/audit/rules/audit-policy.yaml`.

```
brahim@Training:~/cks-lab/labs/12-audit-logging$ vagrant ssh kube-control-plane
Last login: Wed Feb  7 21:54:06 2024 from 10.0.2.2
vagrant@kube-control-plane:~$ sudo mkdir -p /etc/kubernetes/audit/rules
vagrant@kube-control-plane:~$ sudo vim /etc/kubernetes/audit/rules/audit-policy.yaml
vagrant@kube-control-plane:~$ sudo cat /etc/kubernetes/audit/rules/audit-policy.yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
```

2. Configure the API server to consume the audit policy file by editing its configuration file.

```
vagrant@kube-control-plane:~$ sudo vim /etc/kubernetes/manifests/kube-apiserver.yaml
vagrant@kube-control-plane:~$ sudo grep -A5 audit /etc/kubernetes/manifests/kube-apiserver.yaml
    - --audit-policy-file=/etc/kubernetes/audit/rules/audit-policy.yaml
    - --advertise-address=192.168.56.10
    - --allow-privileged=true
    - --authorization-mode=Node,RBAC
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --enable-admission-plugins=NodeRestriction
--
    - mountPath: /etc/kubernetes/audit/rules/audit-policy.yaml
      name: audit
      readOnly: true
    - mountPath: /var/log/kubernetes/audit/logs/
      name: audit-log
      readOnly: false
  hostNetwork: true
  priority: 2000001000
  priorityClassName: system-node-critical
  securityContext:
--
  - name: audit
    hostPath:
      path: /etc/kubernetes/audit/rules/audit-policy.yaml
      type: File
  - name: audit-log
    hostPath:
      path: /var/log/kubernetes/audit/logs/
      type: DirectoryOrCreate
status: {}
vagrant@kube-control-plane:~$
```

3. Add a rule for logging `Metadata`-level events for Deployments, Pods, and Services in all namespaces.

```
vagrant@kube-control-plane:~$ sudo cat /etc/kubernetes/audit/rules/audit-policy.yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
- level: Metadata
  resources:
  - group: ""
    resources: ["pods", "deployments", "services"]
```

4. Add another rule for logging `RequestResponse`-level events for ConfigMaps and Secrets in the namespace `config-data`.

```
vagrant@kube-control-plane:~$ sudo cat /etc/kubernetes/audit/rules/audit-policy.yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
- level: Metadata
  resources:
  - group: ""
    resources: ["pods", "deployments", "services"]
- level: RequestResponse
  resources:
  - group: ""
    resources: ["configmaps", "secrets"]
  namespaces: ["config-data"]
```

5. Add a third rule for ignoring `Metadata`-level events for Pod `log` and `status` commands.

```
vagrant@kube-control-plane:~$ sudo cat /etc/kubernetes/audit/rules/audit-policy.yaml
apiVersion: audit.k8s.io/v1
kind: Policy
rules:
- level: Metadata
  resources:
  - group: ""
    resources: ["pods", "deployments", "services"]
- level: RequestResponse
  resources:
  - group: ""
    resources: ["configmaps", "secrets"]
  namespaces: ["config-data"]
- level: None
  resources:
  - group: ""
    resources: ["pods/log", "pods/status"]
```

6. Register the audit policy file with the API server. Write the log output to `/var/log/kubernetes/audit/logs/apiserver.log`. The maximum age of the log entries should not exceed 30 days.

```
vagrant@kube-control-plane:~$ sudo vim /etc/kubernetes/manifests/kube-apiserver.yaml
vagrant@kube-control-plane:~$ sudo grep -A5 audit-policy-file /etc/kubernetes/manifests/kube-apiserver.yaml
    - --audit-policy-file=/etc/kubernetes/audit/rules/audit-policy.yaml
    - --audit-log-path=/var/log/kubernetes/audit/logs/apiserver.log
    - --audit-log-maxage=30
    - --advertise-address=192.168.56.10
    - --allow-privileged=true
    - --authorization-mode=Node,RBAC
vagrant@kube-control-plane:~$
```

7. Create a Pod named `nginx` with the image `nginx:1.21.6`.

```
brahim@Training:~/cks-lab/labs/12-audit-logging$ kubectl run nginx --image=nginx:1.21.6
pod/nginx created
brahim@Training:~/cks-lab/labs/12-audit-logging$
brahim@Training:~/cks-lab/labs/12-audit-logging$ kubectl get pod
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          16s
brahim@Training:~/cks-lab/labs/12-audit-logging$
```

8. Find the relevant entry in the audit log.

```
brahim@Training:~/cks-lab/labs/12-audit-logging$ vagrant ssh kube-control-plane
Last login: Wed Feb  7 22:42:12 2024 from 10.0.2.2
vagrant@kube-control-plane:~$ sudo cat /var/log/kubernetes/audit/logs/apiserver.log | grep audit.k8s.io/v1
{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Metadata","auditID":"fc4c9410-c252-443b-a0a6-318b00ce9f40","stage":"RequestReceived","
requestURI":"/api/v1/namespaces/kube-system/pods/kube-apiserver-kube-control-plane","verb":"delete","user":{"username":"system:node:kube-contr
ol-plane","groups":["system:nodes","system:authenticated"]},"sourceIPs":["192.168.56.10"],"userAgent":"kubelet/v1.28.2 (linux/amd64) kubernete
s/89a4ea3","objectRef":{"resource":"pods","namespace":"kube-system","name":"kube-apiserver-kube-control-plane","apiVersion":"v1"},"requestRece
ivedTimestamp":"2024-02-07T23:03:31.645275Z","stageTimestamp":"2024-02-07T23:03:31.645275Z"}
{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Metadata","auditID":"afbc3710-8749-453f-aa0b-3eed21be89f7","stage":"RequestReceived","
requestURI":"/api/v1/namespaces/kube-system/pods/kube-controller-manager-kube-control-plane","verb":"get","user":{"username":"system:node:kube
-control-plane","groups":["system:nodes","system:authenticated"]},"sourceIPs":["192.168.56.10"],"userAgent":"kubelet/v1.28.2 (linux/amd64) kub
ernetes/89a4ea3","objectRef":{"resource":"pods","namespace":"kube-system","name":"kube-controller-manager-kube-control-plane","apiVersion":"v1
"},"requestReceivedTimestamp":"2024-02-07T23:03:31.666422Z","stageTimestamp":"2024-02-07T23:03:31.666422Z"}
```