

Kubernetes (k8s) via Rsyslog

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Общее

Настройка логирования Kubernetes (k8s) выполняется путем модификации kube-apiserver. Подробное описание механизма аудита k8s приведено на официальном [сайте](#). Данная инструкция предназначена для настройки аудита k8s для последующей передачи логов в KUMA.

Настройка k8s

1. Необходимо подключиться к ноде k8s с ролью control plane
2. На ноде создаем директорию, куда будет помещена политика аудита

```
sudo mkdir /etc/kubernetes/audit/
```

3. В созданной директории создаем файл с политикой аудита `/etc/kubernetes/audit/audit-policy.yaml` любым удобным способом. Содержимое файла может варьироваться от целей логирования, ниже приведен пример политики с официального сайта.

Будьте внимательны, конфигурации в k8s как правило задаются в виде файлов YAML, которые чувствительны к отступам. Валидируйте файлы перед их применением во избежание ошибок.

Пример политики аудита k8s

```
apiVersion: audit.k8s.io/v1 # This is required.
kind: Policy
# Don't generate audit events for all requests in RequestReceived stage.
omitStages:
```

- "RequestReceived"

rules:

Log pod changes at RequestResponse level

- level: RequestResponse

resources:

- group: ""

Resource "pods" doesn't match requests to any subresource of pods,

which is consistent with the RBAC policy.

resources: ["pods"]

Log "pods/log", "pods/status" at Metadata level

- level: Metadata

resources:

- group: ""

resources: ["pods/log", "pods/status"]

Don't log requests to a configmap called "controller-leader"

- level: None

resources:

- group: ""

resources: ["configmaps"]

resourceNames: ["controller-leader"]

Don't log watch requests by the "system:kube-proxy" on endpoints or services

- level: None

users: ["system:kube-proxy"]

verbs: ["watch"]

resources:

- group: "" # core API group

resources: ["endpoints", "services"]

Don't log authenticated requests to certain non-resource URL paths.

- level: None

userGroups: ["system:authenticated"]

nonResourceURLs:

- "/api*" # Wildcard matching.

- "/version"

Log the request body of configmap changes in kube-system.

- level: Request

```

resources:
- group: "" # core API group
  resources: ["configmaps"]
# This rule only applies to resources in the "kube-system" namespace.
# The empty string "" can be used to select non-namespaced resources.
namespaces: ["kube-system"]

# Log configmap and secret changes in all other namespaces at the Metadata level.
- level: Metadata
  resources:
  - group: "" # core API group
    resources: ["secrets", "configmaps"]

# Log all other resources in core and extensions at the Request level.
- level: Request
  resources:
  - group: "" # core API group
  - group: "extensions" # Version of group should NOT be included.

# A catch-all rule to log all other requests at the Metadata level.
- level: Metadata
  # Long-running requests like watches that fall under this rule will not
  # generate an audit event in RequestReceived.
  omitStages:
  - "RequestReceived"

```

4. Далее создаем директорию, в которую будут записаны логи аудита k8s

```
sudo mkdir -p /var/log/kubernetes/audit/
```

5. Далее необходимо будет внести изменения в конфигурацию пода kube-apiserver. Перед этим настоятельно рекомендуется сделать резервную копию конфигурации, например, следующей командой из вашей рабочей директории:

```
sudo cp /etc/kubernetes/manifests/kube-apiserver.yaml .
```

6. Вносим изменение в kube-apiserver с помощью команды:

```
sudo vi /etc/kubernetes/manifests/kube-apiserver.yaml
```

7. В секции `spec.containers.command` указываем следующие флаги, соблюдая отступы:

```
- --audit-policy-file=/etc/kubernetes/audit/audit-policy.yaml
- --audit-log-path=/var/log/kubernetes/audit/audit.log
```

Где `/etc/kubernetes/audit/audit-policy.yaml` - путь к политике аудита, а `/var/log/kubernetes/audit/audit.log` - путь к файлу для записи логов.

8. Дополнительно можно определить другие параметры логирования, такие как размер файла логов и количество файлов (подробное описание параметров можно найти [тут](#)):

```
- --audit-log-maxsize=500
- --audit-log-maxbackup=3
```

Настройка секции `spec.containers.command`

```
spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=10.0.2.15
    - --allow-privileged=true
    - --authorization-mode=Node,RBAC
    - --client-ca-file=/etc/kubernetes/pki/ca.crt
    - --enable-admission-plugins=NodeRestriction
    - --enable-bootstrap-token-auth=true
    - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
    - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
    - --etcd-keyfile=/etc/kubernetes/pki/apiserver-etcd-client.key
    - --etcd-servers=https://127.0.0.1:2379
    - --kubelet-client-certificate=/etc/kubernetes/pki/apiserver-kubelet-client.crt
    - --kubelet-client-key=/etc/kubernetes/pki/apiserver-kubelet-client.key
    - --kubelet-preferred-address-types=InternalIP,ExternalIP,Hostname
    - --proxy-client-cert-file=/etc/kubernetes/pki/front-proxy-client.crt
    - --proxy-client-key-file=/etc/kubernetes/pki/front-proxy-client.key
    - --requestheader-allowed-names=front-proxy-client
    - --requestheader-client-ca-file=/etc/kubernetes/pki/front-proxy-ca.crt
    - --requestheader-extra-headers-prefix=X-Remote-Extra-
    - --requestheader-group-headers=X-Remote-Group
    - --requestheader-username-headers=X-Remote-User
    - --secure-port=6443
    - --service-account-issuer=https://kubernetes.default.svc.cluster.local
    - --service-account-key-file=/etc/kubernetes/pki/sa.pub
    - --service-account-signing-key-file=/etc/kubernetes/pki/sa.key
    - --service-cluster-ip-range=10.96.0.0/12
    - --tls-cert-file=/etc/kubernetes/pki/apiserver.crt
    - --tls-private-key-file=/etc/kubernetes/pki/apiserver.key
    - --audit-policy-file=/etc/kubernetes/audit/policy.yaml
    - --audit-log-path=/etc/kubernetes/audit/audit.log
    - --audit-log-maxsize=500
    - --audit-log-maxbackup=3
    image: registry.k8s.io/kube-apiserver:v1.30.2
    imagePullPolicy: IfNotPresent
    livenessProbe:
```

9. Далее в том же файле создаем соответствующие тома и точки монтирования для политики и директории для хранения логов

10. В секцию `volumes` добавляем следующее соблюдая отступы:

```
- hostPath:
  path: /etc/kubernetes/audit/audit-policy.yaml
  type: File
name: audit
- hostPath:
  path: /var/log/kubernetes/audit/
  type: DirectoryOrCreate
name: audit-log
```

Здесь `/etc/kubernetes/audit/audit-policy.yaml` - путь к политике аудита, а `/var/log/kubernetes/audit/audit.log` - путь к файлу для записи логов.

Настройка секции `volumes`

```

volumes:
- hostPath:
    path: /etc/ssl/certs
    type: DirectoryOrCreate
    name: ca-certs
- hostPath:
    path: /etc/ca-certificates
    type: DirectoryOrCreate
    name: etc-ca-certificates
- hostPath:
    path: /etc/kubernetes/pki
    type: DirectoryOrCreate
    name: k8s-certs
- hostPath:
    path: /usr/local/share/ca-certificates
    type: DirectoryOrCreate
    name: usr-local-share-ca-certificates
- hostPath:
    path: /usr/share/ca-certificates
    type: DirectoryOrCreate
    name: usr-share-ca-certificates
- hostPath:
    path: /etc/kubernetes/audit/audit-policy.yaml
    type: File
    name: audit
- hostPath:
    path: /var/log/kubernetes/audit/
    type: DirectoryOrCreate
    name: audit-log

```

11. В секцию `volumeMounts` добавляем следующее соблюдая отступы:

```

- mountPath: /etc/kubernetes/audit/audit-policy.yaml
  name: audit
  readOnly: true
- mountPath: /var/log/kubernetes/audit/
  name: audit-log
  readOnly: false

```

Настройка секции `volumeMounts`

```
volumeMounts:
- mountPath: /etc/ssl/certs
  name: ca-certs
  readOnly: true
- mountPath: /etc/ca-certificates
  name: etc-ca-certificates
  readOnly: true
- mountPath: /etc/kubernetes/pki
  name: k8s-certs
  readOnly: true
- mountPath: /usr/local/share/ca-certificates
  name: usr-local-share-ca-certificates
  readOnly: true
- mountPath: /usr/share/ca-certificates
  name: usr-share-ca-certificates
  readOnly: true
- mountPath: /etc/kubernetes/audit/audit-policy.yaml
  name: audit
  readOnly: true
- mountPath: /var/log/kubernetes/audit/
  name: audit-log
  readOnly: false
```

12. Сохраняем все внесенные в файл изменения.

Т.к. была изменена конфигурация kube-apiserver, то под будет пересоздан, что может потребовать примерно до 1 минуты времени. Если под не смог подняться, необходимо проверить все внесенные изменения на предмет ошибок и опечаток, а также изучить логи по пути `/var/log/pods/`

Если все было сделано правильно, то под kube-apiserver поднимется и в директории `/var/log/kubernetes/audit/` появится файл `audit.log` и начнет наполняться логами k8s.

Настройка Rsyslog

Настройки ниже приведены для deb-систем.

1. Установка rsyslog

```
apt install rsyslog
```

2. Включение и запуск службы rsyslog

```
systemctl enable rsyslog.service  
systemctl start rsyslog.service
```

3. Создание файла конфигурации для отправки через rsyslog файла лога k8s

```
nano /etc/rsyslog.d/k8s.conf
```

Пример содержимого файла с отправкой по TCP:

```
$ModLoad imfile  
$InputFileName /var/log/kubernetes/audit/audit.log  
$InputFileTag tag_k8s_log:  
$InputFileStateFile k8s_log  
$InputFileSeverity info  
$InputFileFacility local6  
$InputRunFileMonitor  
  
local6.* @@10.10.10.10:7777
```

Где 10.10.10.10 - адрес коллектора KUMA, 7777 - порт коллектора KUMA

Для отправки событий по протоколу UDP последнюю строчку следует заменить на:

```
local6.* @10.10.10.10:7777
```

4. После сохранения изменений в файле необходимо перезапустить сервис Rsyslog командой:

```
systemctl restart rsyslog.service
```

Настройка коллектора KUMA

После того как параметры передачи событий настроены, требуется создать коллектор в веб-интерфейсе KUMA для событий `k8s`.

1. На шаге **Транспорт** укажите тип и порт в соответствии с настройками на стороне *Kubernetes*.

2. На шаге **Парсинг** событий выберите нормализатор **k8s via syslog**.

3. На шаге **Маршрутизация** проверьте, что в набор ресурсов коллектора добавлены следующие точки назначения:

- **Хранилище**. Для отправки обработанных событий в хранилище.

- **Коррелятор**. Для отправки обработанных событий в коррелятор.

Если точки назначения **Хранилище** и **Коррелятор** не добавлены, создайте их.

4. На шаге **Проверка параметров** нажмите **Сохранить и создать сервис**.

5. Скопируйте появившуюся команду для установки коллектора KUMA.

Полезные ссылки

Пакет контента для k8s: https://github.com/KUMA-Community/kuma_content/tree/main/rules/app/kubernetes

Документация по настройке аудита k8s: <https://kubernetes.io/docs/tasks/debug/debug-cluster/audit/>

Revision #10

Created 9 October 2024 07:40:07 by Koala

Updated 24 October 2024 10:07:36 by Koala