

Kubernetes (k8s) via webhook

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

Данный способ является экспериментальным. Рекомендуемый способ приведен в статье [k8s via rsyslog](#)

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. В этой же директории создаем файл `/etc/kubernetes/audit/audit-webhook.yaml` любым удобным способом. Пример содержимого файла приведен ниже:

```
apiVersion: v1
```

```
kind: Config
```

```
preferences: {}
```

```
clusters:
```

```
- name: kube-auditing
cluster:
  server: http://10.10.10.10:7777/input
users: []
contexts:
- name: default-context
context:
  cluster: kube-auditing
  user: ""
current-context: default-context
```

Где 10.10.10.10 - адрес коллектора KUMA, а 7777 - порт коллектора. Все прочие параметры из примера можно оставлять без изменений.5. Далее создаем директорию, в которую будут записаны логи аудита k8s

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-webhook-config-file=/etc/kubernetes/audit/audit-webhook.yaml
- --audit-webhook-mode=batch
- --audit-webhook-batch-max-size=1
```

Где `/etc/kubernetes/audit/audit-policy.yaml` - путь к политике аудита.

Настройка секции `spec.containers.command`

```

spec:
  containers:
  - command:
    - kube-apiserver
    - --advertise-address=10.68.85.163
    - --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/audit-policy.yaml
    - --audit-webhook-config-file=/etc/kubernetes/audit/audit-webhook.yaml
    - --audit-webhook-mode=batch
    - --audit-webhook-batch-max-size=1
  image: registry.k8s.io/kube-apiserver:v1.31.1

```

9. Далее в том же файле создаем соответствующий том и точку монтирования для политик

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

```

- hostPath:
  path: /etc/kubernetes/audit/
  type: DirectoryOrCreate
  name: k8s-audit

```

Здесь `/etc/kubernetes/audit/` - путь к директории с политикой аудита и настройками webhook

Настройка секции volumes

```
type: DirectoryOrCreate
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/
    type: DirectoryOrCreate
    name: k8s-audit
status: {}
```

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

```
- mountPath: /etc/kubernetes/audit/
  name: k8s-audit
  readOnly: true
```

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/
  name: k8s-audit
  readOnly: true
```

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/`

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* (`audit-webhook.yaml`).

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

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 #6

Created 10 October 2024 10:29:58 by Koala

Updated 24 October 2024 10:08:26 by Koala