



# Red Hat OpenShift Service on AWS 4

## 安全性与合规性

在 AWS 集群上配置安全性上下文约束



在 AWS 集群上配置安全性上下文约束

## 法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

本文档提供有关配置安全性上下文约束的说明。

---

## 目录

<b>第 1 章 查看审计日志</b> .....	<b>3</b>
1.1. 关于 API 审计日志	3
1.2. 查看审计日志	4
1.3. 过滤审计日志	8
1.4. 收集审计日志	8
1.5. 其他资源	9
<b>第 2 章 为基于 IP 的 AWS 角色假设添加额外的限制</b> .....	<b>10</b>
2.1. 创建基于身份的 IAM 策略	10
2.2. 附加基于身份的 IAM 策略	11
2.3. 其他资源	11



# 第 1 章 查看审计日志

Red Hat OpenShift Service on AWS 审计提供一组安全相关的按时间排序的记录，记录各个用户、管理员或其他系统组件影响系统的一系列活动。

## 1.1. 关于 API 审计日志

审计在 API 服务器级别运作，记录所有传入到服务器的请求。每个审计日志包含以下信息：

表 1.1. 审计日志字段

字段	描述
<b>level</b>	生成事件的审计级别。
<b>auditID</b>	为每个请求生成的唯一审计 ID。
<b>stage</b>	生成此事件实例时请求处理的阶段。
<b>requestURI</b>	客户端向服务器发送的请求 URI。
<b>verb</b>	与请求相关联的 Kubernetes 操作动词。对于非资源请求，这是小写 HTTP 方法。
<b>user</b>	经过身份验证的用户信息。
<b>impersonatedUser</b>	可选。如果请求模拟了另一个用户，则为被模拟的用户信息。
<b>sourceIPs</b>	可选。源 IP，请求发起的源和任何中间代理。
<b>userAgent</b>	可选。客户端报告的用户代理字符串。请注意，用户代理由客户端提供，且必须不可信任。
<b>objectRef</b>	可选。这个请求的目标对象引用。这不适用于 <b>List</b> 类型请求，或者非资源请求。
<b>responseStatus</b>	可选。响应的状态，即使 <b>ResponseObject</b> 不是 <b>Status</b> 类型也会生成。对于成功的响应，这只会包括代码。对于非状态类型错误响应，这将自动生成出错信息。
<b>requestObject</b>	可选。请求中的 API 对象，采用 JSON 格式。在进行 version conversion、defaulting、admission 或 merging 之前，在请求中的 <b>RequestObject</b> 记录（可能会被转换为 JSON 格式）。这是一个外部版本化的对象类型，可能自身并不是一个有效的对象。对于非资源请求，这会被忽略，且只在 Request 级别或更高级别中被记录。
<b>responseObject</b>	可选。响应中返回的 API 对象，使用 JSON 格式。在转换为外部类型后， <b>ResponseObject</b> 被记录，并被序列化为 JSON 数据。在非资源请求中会省略它，且仅在 Response 级别中记录。

字段	描述
<b>requestReceivedTimestamp</b>	请求到达 API 服务器的时间。
<b>stageTimestamp</b>	请求到达当前审计阶段的时间。
<b>annotations</b>	可选。一个无结构的键值映射，它存储在一个审计事件中，可以通过在请求服务链中调用的插件来设置它，包括认证、授权和准入插件。请注意，这些注解用于审计事件，且与所提交对象的 <b>metadata.annotations</b> 没有关联。标识信息组件的键应该是唯一的以避免名称冲突，例如 <b>podsecuritypolicy.admission.k8s.io/policy</b> 。值应该较短。注解包含在 Metadata 级别中。

Kubernetes API 服务器的输出示例：

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Metadata",
  "auditID": "ad209ce1-fec7-4130-8192-c4cc63f1d8cd",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/namespaces/openshift-kube-controller-manager/configmaps/cert-recovery-controller-lock?timeout=35s",
  "verb": "update",
  "user": {
    "username": "system:serviceaccount:openshift-kube-controller-manager:localhost-recovery-client",
    "uid": "dd4997e3-d565-4e37-80f8-7fc122ccd785",
    "groups": [
      "system:serviceaccounts",
      "system:serviceaccounts:openshift-kube-controller-manager",
      "system:authenticated"
    ],
    "sourceIPs": ["::1"],
    "userAgent": "cluster-kube-controller-manager-operator/v0.0.0 (linux/amd64) kubernetes/$Format",
    "objectRef": {
      "resource": "configmaps",
      "namespace": "openshift-kube-controller-manager",
      "name": "cert-recovery-controller-lock",
      "uid": "5c57190b-6993-425d-8101-8337e48c7548",
      "apiVersion": "v1",
      "resourceVersion": "574307"
    },
    "responseStatus": {
      "metadata": {},
      "code": 200
    },
    "requestReceivedTimestamp": "2020-04-02T08:27:20.200962Z",
    "stageTimestamp": "2020-04-02T08:27:20.206710Z",
    "annotations": {
      "authorization.k8s.io/decision": "allow",
      "authorization.k8s.io/reason": "RBAC: allowed by ClusterRoleBinding \"system:openshift:operator:kube-controller-manager-recovery\" of ClusterRole \"cluster-admin\" to ServiceAccount \"localhost-recovery-client/openshift-kube-controller-manager\""
    }
  }
}
```

## 1.2. 查看审计日志

您可以查看每个 control plane 节点的 OpenShift API 服务器、Kubernetes API 服务器、OpenShift OAuth API 服务器和 OpenShift OAuth 服务器的日志。

### 流程

查看审计日志：

- 查看 OpenShift API 服务器审计日志：
  - a. 列出每个 control plane 节点可用的 OpenShift API 服务器审计日志：

```
$ oc adm node-logs --role=master --path=openshift-apiserver/
```

### 输出示例

```
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit-2021-03-09T00-12-19.834.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit-2021-03-09T00-11-49.835.log
```



```
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit-2021-03-09T00-13-00.128.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit.log
```

- b. 通过提供节点名称和日志名称来查看特定的 OpenShift API 服务器审计日志：

```
$ oc adm node-logs <node_name> --path=openshift-apiserver/<log_name>
```

例如：

```
$ oc adm node-logs ci-ln-m0wpfjb-f76d1-vnb5x-master-0 --path=openshift-apiserver/audit-2021-03-09T00-12-19.834.log
```

### 输出示例

```
{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Metadata","auditID":"381acf6d-5f30-4c7d-8175-c9c317ae5893","stage":"ResponseComplete","requestURI":"/metrics","verb":"get","user":{"username":"system:serviceaccount:openshift-monitoring:prometheus-k8s","uid":"825b60a0-3976-4861-a342-3b2b561e8f82","groups":["system:serviceaccounts","system:serviceaccounts:openshift-monitoring","system:authenticated"]},"sourceIPs":["10.129.2.6"],"userAgent":"Prometheus/2.23.0","responseStatus":{"metadata":{},"code":200},"requestReceivedTimestamp":"2021-03-08T18:02:04.086545Z","stageTimestamp":"2021-03-08T18:02:04.107102Z","annotations":{"authorization.k8s.io/decision":"allow","authorization.k8s.io/reason":"RBAC: allowed by ClusterRoleBinding \"prometheus-k8s\" of ClusterRole \"prometheus-k8s\" to ServiceAccount \"prometheus-k8s/openshift-monitoring\"\"}}
```

- 查看 Kubernetes API 服务器审计日志：

- a. 列出每个 control plane 节点可用的 Kubernetes API 服务器审计日志：

```
$ oc adm node-logs --role=master --path=kube-apiserver/
```

### 输出示例

```
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit-2021-03-09T14-07-27.129.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit-2021-03-09T19-24-22.620.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit-2021-03-09T18-37-07.511.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit.log
```

- b. 通过提供节点名称和日志名称来查看特定的 Kubernetes API 服务器审计日志：

```
$ oc adm node-logs <node_name> --path=kube-apiserver/<log_name>
```

例如：

```
$ oc adm node-logs ci-ln-m0wpfjb-f76d1-vnb5x-master-0 --path=kube-apiserver/audit-2021-03-09T14-07-27.129.log
```

## 输出示例

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Metadata",
  "auditID": "cfce8a0b-b5f5-4365-8c9f-79c1227d10f9",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/namespaces/openshift-kube-scheduler/serviceaccounts/openshift-kube-scheduler-sa",
  "verb": "get",
  "user": {
    "username": "system:serviceaccount:openshift-kube-scheduler-operator:openshift-kube-scheduler-operator",
    "uid": "2574b041-f3c8-44e6-a057-baef7aa81516",
    "groups": [
      "system:serviceaccounts",
      "system:serviceaccounts:openshift-kube-scheduler-operator",
      "system:authenticated"
    ],
    "sourceIPs": ["10.128.0.8"],
    "userAgent": "cluster-kube-scheduler-operator/v0.0.0 (linux/amd64) kubernetes/$Format",
    "objectRef": {
      "resource": "serviceaccounts",
      "namespace": "openshift-kube-scheduler",
      "name": "openshift-kube-scheduler-sa",
      "apiVersion": "v1"
    },
    "responseStatus": {
      "metadata": {},
      "code": 200,
      "requestReceivedTimestamp": "2021-03-08T18:06:42.512619Z",
      "stageTimestamp": "2021-03-08T18:06:42.516145Z",
      "annotations": {
        "authentication.k8s.io/legacy-token": "system:serviceaccount:openshift-kube-scheduler-operator:openshift-kube-scheduler-operator",
        "authorization.k8s.io/decision": "allow",
        "authorization.k8s.io/reason": "RBAC: allowed by ClusterRoleBinding \"system:openshift:operator:cluster-kube-scheduler-operator\" of ClusterRole \"cluster-admin\" to ServiceAccount \"openshift-kube-scheduler-operator/openshift-kube-scheduler-operator\""
      }
    }
  }
}
```

- 查看 OpenShift OAuth API 服务器审计日志 :
  - a. 列出每个 control plane 节点可用的 OpenShift OAuth API 服务器审计日志 :

```
$ oc adm node-logs --role=master --path=oauth-apiserver/
```

## 输出示例

```
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit-2021-03-09T13-06-26.128.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit-2021-03-09T18-23-21.619.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit-2021-03-09T17-36-06.510.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit.log
```

- b. 通过提供节点名称和日志名称来查看特定的 OpenShift OAuth API 服务器审计日志 :

```
$ oc adm node-logs <node_name> --path=oauth-apiserver/<log_name>
```

例如 :

```
$ oc adm node-logs ci-ln-m0wpfjb-f76d1-vnb5x-master-0 --path=oauth-apiserver/audit-2021-03-09T13-06-26.128.log
```

## 输出示例

```
{
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Metadata",
  "auditID": "dd4c44e2-3ea1-4830-9ab7-c91a5f1388d6",
  "stage": "ResponseComplete",
  "requestURI": "/apis/user.openshift.io/v1/users/~",
  "verb": "get",
  "user": {
    "username": "system:serviceaccount:openshift-"
  }
}
```

```
monitoring:prometheus-k8s", "groups":
["system:serviceaccounts", "system:serviceaccounts:openshift-
monitoring", "system:authenticated"]], "sourceIPs":
["10.0.32.4", "10.128.0.1"], "userAgent": "dockerregistry/v0.0.0 (linux/amd64)
kubernetes/$Format", "objectRef":
{"resource": "users", "name": "~", "apiGroup": "user.openshift.io", "apiVersion": "v1"}, "response
Status": {"metadata": {}, "code": 200}, "requestReceivedTimestamp": "2021-03-
08T17:47:43.653187Z", "stageTimestamp": "2021-03-
08T17:47:43.660187Z", "annotations":
{"authorization.k8s.io/decision": "allow", "authorization.k8s.io/reason": "RBAC: allowed by
ClusterRoleBinding \"basic-users\" of ClusterRole \"basic-user\" to Group
\"system:authenticated\""}}}
```

- 查看 OpenShift OAuth 服务器审计日志：

- a. 列出每个 control plane 节点可用的 OpenShift OAuth 服务器审计日志：

```
$ oc adm node-logs --role=master --path=oauth-server/
```

### 输出示例

```
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit-2022-05-11T18-57-32.395.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-0 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit-2022-05-11T19-07-07.021.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-1 audit.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit-2022-05-11T19-06-51.844.log
ci-ln-m0wpfjb-f76d1-vnb5x-master-2 audit.log
```

- b. 通过提供节点名称和日志名称来查看特定的 OpenShift OAuth 服务器审计日志：

```
$ oc adm node-logs <node_name> --path=oauth-server/<log_name>
```

例如：

```
$ oc adm node-logs ci-ln-m0wpfjb-f76d1-vnb5x-master-0 --path=oauth-server/audit-
2022-05-11T18-57-32.395.log
```

### 输出示例

```
{"kind": "Event", "apiVersion": "audit.k8s.io/v1", "level": "Metadata", "auditID": "13c20345-
f33b-4b7d-b3b6-
e7793f805621", "stage": "ResponseComplete", "requestURI": "/login", "verb": "post", "user":
{"username": "system:anonymous", "groups": ["system:unauthenticated"]}, "sourceIPs":
["10.128.2.6"], "userAgent": "Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101
Firefox/91.0", "responseStatus": {"metadata":
{}}, "code": 302, "requestReceivedTimestamp": "2022-05-
11T17:31:16.280155Z", "stageTimestamp": "2022-05-
11T17:31:16.297083Z", "annotations":
{"authentication.openshift.io/decision": "error", "authentication.openshift.io/username": "kubea
dmin", "authorization.k8s.io/decision": "allow", "authorization.k8s.io/reason": ""}}
```

**authentication.openshift.io/decision** 注解的可能值 **allow**、**deny** 或 **error**。

### 1.3. 过滤审计日志

您可以使用 **jq** 或另一个 JSON 解析工具来过滤 API 服务器审计日志。



#### 注意

日志记录到 API 服务器审计日志的信息量是由设置的审计日志策略控制的。

以下流程提供了使用 **jq** 在 control plane 节点 **node-1.example.com** 上过滤审计日志的示例。有关使用 **jq** 的详情，请参考 **jq 手册**。

#### 前提条件

- 您可以使用具有 **dedicated-admin** 角色的用户访问集群。
- 您已安装了 **jq**。

#### 流程

- 根据用户过滤 OpenShift API 服务器审计日志：

```
$ oc adm node-logs node-1.example.com \
  --path=openshift-apiserver/audit.log \
  | jq 'select(.user.username == "myusername")'
```

- 根据用户代理过滤 OpenShift API 服务器审计日志：

```
$ oc adm node-logs node-1.example.com \
  --path=openshift-apiserver/audit.log \
  | jq 'select(.userAgent == "cluster-version-operator/v0.0.0 (linux/amd64)
  kubernetes/$Format")'
```

- 通过特定 API 版本过滤 Kubernetes API 服务器审计日志，仅输出用户代理：

```
$ oc adm node-logs node-1.example.com \
  --path=kube-apiserver/audit.log \
  | jq 'select(.requestURI | startswith("/apis/apiextensions.k8s.io/v1beta1")) | .userAgent'
```

- 通过排除动词来过滤 OpenShift OAuth API 服务器审计日志：

```
$ oc adm node-logs node-1.example.com \
  --path=oauth-apiserver/audit.log \
  | jq 'select(.verb != "get")'
```

- 根据标识用户名和失败并显示错误的事件过滤 OpenShift OAuth 服务器审计日志：

```
$ oc adm node-logs node-1.example.com \
  --path=oauth-server/audit.log \
  | jq 'select(.annotations["authentication.openshift.io/username"] != null and
  .annotations["authentication.openshift.io/decision"] == "error")'
```

### 1.4. 收集审计日志

您可以使用 `must-gather` 工具来收集审计日志以调试集群，您可以检查或发送到红帽支持。

## 流程

1. 使用 `-- /usr/bin/gather_audit_logs` 运行 `oc adm must-gather` 命令：

```
$ oc adm must-gather -- /usr/bin/gather_audit_logs
```

2. 从工作目录中刚刚创建的 `must-gather` 目录创建一个压缩文件。例如，在使用 Linux 操作系统的计算机上运行以下命令：

```
$ tar cvaf must-gather.tar.gz must-gather.local.472290403699006248 1
```

- 1** 将 `must-gather-local.472290403699006248` 替换为实际目录名称。

3. 在 [红帽客户门户网站的客户支持页面](#)中，将压缩文件附加到支持问题单中。

## 1.5. 其他资源

- [must-gather 工具](#)
- [关于日志转发](#)

## 第 2 章 为基于 IP 的 AWS 角色假设添加额外的限制

您可以在 AWS 帐户中实施额外的安全层，以防止角色假设非允许 IP 地址。

### 2.1. 创建基于身份的 IAM 策略

您可以创建一个基于身份的 Identity and Access Management (IAM)策略，当请求来自红帽提供的 IP 以外的 IP 地址时，拒绝访问所有 AWS 操作。

#### 前提条件

- 您可以使用创建和修改 IAM 策略所需的权限访问 [see AWS 管理控制台](#)。

#### 流程

1. 使用您的 AWS 帐户凭证登录到 AWS 管理控制台。
2. 进入 IAM 服务。
3. 在 IAM 控制台中，从左侧导航菜单中选择 **Policies**。
4. 点击 **Create policy**。
5. 选择 **JSON** 选项卡，以使用 JSON 格式定义策略。
6. 要获取您需要输入到 JSON 策略文档中的 IP 地址，请运行以下命令：

```
$ ocm get /api/clusters_mgmt/v1/trusted_ip_addresses
```



#### 注意

这些 IP 地址不是永久的，可能随时更改。您必须持续查看 API 输出，并在 JSON 策略文档中进行必要的更新。

7. 将以下 **policy\_document.json** 文件复制并粘贴到编辑器中：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": []
        },
        "Bool": {
          "aws:ViaAWSService": "false"
        }
      }
    }
  ]
}
```

8. 将在第 6 步中获得的所有 IP 地址复制并粘贴到 `policy_document.json` 文件中的 `"aws:SourceIp": []` 数组中。
9. 点 **Review and create**。
10. 为策略提供名称和描述，并查看准确性的详细信息。
11. 点 **Create policy** 保存该策略。



### 注意

条件键 `aws:ViaAWSService` 必须设置为 `false`，以便根据初始调用启用后续调用成功。例如，如果您对 `aws ec2 describe-instances` 进行了初始调用，则在 AWS API 服务器中进行的所有后续调用，以检索附加到 ec2 实例的 EBS 卷的信息，如果条件键 `aws:ViaAWSService` 没有设置为 `false`，则失败。后续调用会失败，因为它们会源自 AWS IP 地址，这些地址不包含在 AllowList 中。

## 2.2. 附加基于身份的 IAM 策略

创建基于身份的 IAM 策略后，将其附加到 AWS 帐户中的相关 IAM 用户、组或角色，以防止这些实体的基于 IP 的角色假设。

### 流程

1. 进入 AWS Management Console 中的 IAM 控制台。
2. 选择您要为其附加策略的默认 IAM `ManagedOpenShift-Support-Role` 角色。



### 注意

您可以更改默认的 IAM `ManagedOpenShift-Support-Role` 角色。有关角色的更多信息，请参阅 [红帽支持访问](#)。

3. 在 **Permissions** 选项卡中，从 **Add Permissions** 下拉列表中选择 **Add Permissions** 或 **Create inline policy**。
4. 搜索您之前创建的策略：
  - a. 输入策略名称。
  - b. 根据适当的类别进行过滤。
5. 选择策略，再单击 **Attach policy**。



### 重要

为确保基于 IP 的角色假设阻止，您必须保持允许列表的 IP 保持最新状态。如果不这样做，可能会导致红帽站点可靠性工程(SRE)无法访问您的帐户并影响您的 SLA。如果您有任何疑问或需要帮助，请联系我们的支持团队。

## 2.3. 其他资源

- 有关基于源 IP 拒绝访问的更多信息，请参阅 [AWS : 根据 AWS 文档中的源 IP 访问 AWS](#)。

