Rancher on AWS, Logging to CloudWatch with IRSA for Authentication

Date:	2023-07-05
modified:	2023-07-12
tags:	AWS, EKS, IRSA, IAM, Kubernetes, Amazon, Rancher, Logging, CloudWatch
description:	Rancher Logging to AWS CloudWatch with IRSA
category:	Container
slug:	rancher-on-aws-logging-to-cloudwatch-with-irsa-for-authentication
Author:	Dominik Wombacher
lang:	en
transid:	rancher-on-aws-logging-to-cloudwatch-with-irsa-for-authentication
Status:	published

This is the third Article of the Series **Integrate Rancher with AWS services**, I'm focusing on Logging to CloudWatch from Rancher by using IAM Roles for Service Accounts (IRSA) to authenticate to avoid long-term credentials.

Update: The recording of my talk Rancher integration with AWS services: possibilities, challenges, outlook (abstract and slide-deck) at openSUSE Conference 23 is online and covers parts of this article as well.

- media.ccc.de (includes options to download video and audio)
- youtube.com

Terminology

I assume you have a basic level of understanding about *Kubernetes objects* and *annotations* as well as *Helm charts*, *repositories*, *releases*. If you want to brush up your knowledge, links to resources about those topics are part of the second article of this series: Rancher on AWS, Backup to S3 with IRSA for Authentication

To learn more about AWS IAM Roles and IRSA, I recommend to checkout the first Article of this series: What is IAM Roles for Service Accounts (IRSA) and Amazon EKS Pod Identity Webhook?

Rancher Logging

Overview

Rancher provides the rancher-logging Helm chart, which is based on the kube-logging operator, it's using Fluent Bit to collect and Fluentd to forward the logs. One of the supported targets is Amazon CloudWatch.

With enhanced cloud provider logging, logs from Amazon EKS will be collected and pushed to CloudWatch as well.

IRSA is technically supported but the necessary *serviceAccount annotation* need to be added after the installation in a separate step.

The official documentation about Rancher Integration with Logging Services provides further information about the functionality and general installation.

IAM Policy

You need a IAM Policy to create later the IAM Role linked to a Kubernetes service account. An example how such a policy could look like to push logs to CloudWatch, based on the out_cloudwatch_logs example:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:PutRetentionPolicy",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

IRSA Configuration

The *kube-logging* operator already supports IRSA but it can't be configured as part of the Helm installation with the *rancher-logging* Helm chart yet. The annotation need to be added afterwards to the Logging Resources, this is different compared to Rancher Backup, where this is possible directly as part of the installation.

I created a GitHub Pull Request to include the *serviceAccount* annotation in the Helm chart.

In the meantime, you have to edit the Logging Resources `rancher-logging-root` and (if Rancher is running on Amazon EKS with enabled enhanced cloud logging) `rancher-logging-eks` in namespace `cattle-logging-system` manually.

The configuration is not covered in the official Rancher Documentation yet. Six lines need to be added per Logging Resource, each with it's own `role-arn`:

```
spec:
  fluentd:
    serviceAccount:
      metadata:
        annotations:
          eks.amazonaws.com/role-arn: arn:aws:iam::1234567890:role/my-rancher-logging-role
```

Besides the *serviceAccount annotation*, the initial IRSA setup for the cluster and the creation of the IAM Role, with a link to the service account, used by the *kube-logging* operator, is required.

The names of the service accounts *rancher-logging* will create are `rancher-logging-root-fluentd` and (in case of EKS and enhanced logging) `rancher-logging-root-fluentd`, also in namespace `cattle-logging-system`. You have to create a IAM Role and link it to those accounts by using `eksctl` as shown in the first article What is IAM Roles for Service Accounts (IRSA) and Amazon EKS Pod Identity Webhook?.

ClusterOutput

To push logs to the actual target, Fluentd is using so called *Outputs*. There are two types, **Output** (namespace) and **ClusterOutput** (cluster-wide), further reading.

Following an example how a basic *ClusterOutput* could look like, I named it `testcloudwatchoutput` and it will push all available logs to the group `rancher-demo-cluster-log-group` and stream `rancher-demo-cluster-log-stream` in Amazon CloudWatch of AWS Region `us-east-1`.

```yaml
apiVersion: logging.banzaicloud.io/v1beta1
kind: ClusterOutput
metadata:
  creationTimestamp: '2023-05-17T14:03:16Z'
  generation: 4
  managedFields:
    - apiVersion: logging.banzaicloud.io/v1beta1
      fieldsType: FieldsV1
      fieldsV1:
        f:status:
          .: {}
          f:active: {}
      manager: manager
      operation: Update
      subresource: status
      time: '2023-05-17T14:03:16Z'
    - apiVersion: logging.banzaicloud.io/v1beta1
      fieldsType: FieldsV1
      fieldsV1:
        f:spec:
          .: {}
          f:cloudwatch:
            .: {}
            f:auto_create_stream: {}
            f:log_group_name: {}
            f:log_stream_name: {}
            f:region: {}
      manager: rancher
      operation: Update
      time: '2023-05-18T10:14:12Z'
  name: testcloudwatchout
  namespace: cattle-logging-system
  resourceVersion: '16031275'
  uid: a4bd1852-eca8-487b-bdc8-47d9966e6da2
spec:
  cloudwatch:
    auto_create_stream: true
    log_group_name: rancher-demo-cluster-log-group
    log_stream_name: rancher-demo-cluster-log-stream
    region: us-east-1
status:
  active: true
```

Conclusion

Logging is a complex field and very dependent on the individual requirements and use-case. I recommend to invest some time to learn about *kube-logging*, to write down the goals you want to achieve with your log setup and work backwards from there to perform the actual configuration.

This is way beyond the scope of this Article, my main intention was to show how to bring IRSA into the mix.

Compared to Rancher Backup, it's a little more effort to get IRSA working, mainly because of the missing support in the Helm chart, but it's worth it and not too complicated.

I hope my contribution to the *rancher-logging* Helm chart will help to improve the user experience. As soon a pull request that allows the *serviceAccount* annotation, was merged, I plan to also submit a PR to get the Rancher Documentation updated accordingly. Configuring such a feature, based on security best practices, should be as easy as possible to achieve broad adoption :)

In the next article of this series, I take a break from IRSA and will talk about login to Rancher via SAML Authentication by using AWS IAM Identity Center as SAML identity provider.

Article series **Integrate Rancher with AWS services**:

1. What is IAM Roles for Service Accounts (IRSA) and Amazon EKS Pod Identity Webhook?
2. Rancher on AWS, Backup to S3 with IRSA for Authentication
3. **Rancher on AWS, Logging to CloudWatch with IRSA for Authentication**
4. Rancher on AWS, SAML Authentication with AWS IAM Identity Center as SAML IdP (coming soon)
5. Rancher on AWS, GitOps with Fleet and AWS CodeCommit (coming soon)