+++ title = “Authentication” description = “AWS authentication” keywords = [“grafana”, “aws”, “authentication”] aliases = [“/docs/grafana/latest/datasources/cloudwatch”] weight = 5 +++

AWS authentication

Requests from a Grafana plugin to AWS are made on behalf of an IAM role or an IAM user. The IAM user or IAM role must have the associated policies to perform certain API actions. Since these policies are specific to each data source, refer to the data source documentation for details.

All requests to AWS APIs are performed on the server side by the Grafana backend using the official AWS SDK.

This topic has the following sections:

- Authentication methods
- Assuming a role
- Endpoint
- AWS credentials file
- EKS IAM roles for service accounts

Authentication methods

You can use one of the following authentication methods. Currently, `AWS SDK Default`, `Credentials file` and `Access and secret key` are enabled by default in open source Grafana. You can enable/disable them if necessary if you have server configuration access. For more information, refer to [allowed_auth_providers]({{< relref “../../administration/configuration.md#allowed_auth_providers” >}}) documentation.

- `AWS SDK Default` performs no custom configuration and instead uses the default provider as specified by the AWS SDK for Go. It requires you to configure your AWS credentials separately, such as if you’ve configured the CLI, if you’re running on an EC2 instance, in an ECS task, or for a Service Account in a Kubernetes cluster.

- `Credentials file` corresponds directly to the SharedCredentialsProvider provider in the Go SDK. It reads the AWS shared credentials file to find a given profile. While `AWS SDK Default` will also find the shared credentials file, this option allows you to specify which profile to use without using environment variables. This option doesn’t have any implicit fallbacks to other credential providers, and it fails if the credentials provided from the file aren’t correct.

- `Access and secret key` corresponds to the StaticProvider and uses the given access key ID and secret key to authenticate. This method doesn’t have any fallbacks, and will fail if the provided key pair doesn’t work.

- `Workspace IAM role` corresponds to the EC2RoleProvider. The EC2RoleProvider pulls credentials for a role attached to the EC2 instance that Grafana runs on. You can also achieve this by using the authentication method AWS SDK Default, but this option is different as it doesn't have any fallbacks. This option is currently only enabled by default in Amazon Managed Grafana.

Assuming a role

The `Assume Role ARN` field allows you to specify which IAM role to assume. When left blank, the provided credentials are used directly and the associated role or user should have the required permissions. If this field is non-blank, on the other hand, the provided credentials are used to perform an sts:AssumeRole call.

You can disable this feature in the Grafana configuration. For more information, refer to [assume_role_enabled]({{< relref "../../administration/configuration.md#assume_role_enabled" >}}) documentation.

External ID

If you are assuming a role in another account that was created with an external ID, then specify the external ID in this field. For more information, refer to the AWS documentation on external ID.

Endpoint

The `Endpoint` field allows you to specify a custom endpoint URL that overrides the default generated endpoint for the AWS service API. Leave this field blank if you want to use the default generated endpoint. For more information on why and how to use Service endpoints, refer to the AWS service endpoints documentation.

AWS credentials file

Create a file at `~/.aws/credentials`. That is the `HOME` path for user running grafana-server.

> **Note:** If you think you have the credentials file in the right place and it is still not working, you might try moving your .aws file to '/usr/share/grafana/' and make sure your credentials file has at most 0644 permissions.

Example content:

```
[default]
aws_access_key_id = asdsadasdasdasd
```

```
aws_secret_access_key = dasdasdsadasdasdasdsa
region = us-west-2
```

EKS IAM roles for service accounts

The Grafana process in the container runs as user 472 (called “grafana”). When Kubernetes mounts your projected credentials, they will by default only be available to the root user. To allow user 472 to access the credentials (and avoid falling back to the IAM role attached to the EC2 instance), you need to provide a security context for your pod.

```
securityContext:
  fsGroup: 472
  runAsUser: 472
  runAsGroup: 472
```