Configure for AWS Secrets Manager

Contents:

Secrets Manager is a secure AWS service that enables you to store and manage Personal Access Tokens, which are used by the Trifacta® platform to access AWS resources. When a user requests access to an applicable AWS resource, the Trifacta platform queries the Secrets Manager for the user's Personal Access Token, which is then used to access the resource. Individual users do not have access to the Secrets Manager.

For storing Personal Access Tokens, the Trifacta platform uses the following attributes to distinguish platform secrets from secrets stored by other applications or users in the customer's AWS account:

- Application name
- Environment

Permissions to use secrets can be controlled based on custom Tags or Paths.

For more information, see *https://docs.databricks.com/dev-tools/api/latest/authentication.html*.

Limitations

> **NOTE**: You can have only one AWS account per Secrets Manager and per region. See below for configuring region-related settings.

> **NOTE**: Customer-managed encryption keys are not supported for use in AWS Secrets Manager by the Trifacta platform.

Supported AWS resources

AWS Secrets Manager is supported for use in accessing the Personal Access Tokens for platform users for the following AWS resources:

- AWS Databricks

Configure

Please complete the following configuration steps to enable integration with AWS Secrets Manager.

Configure region and metadata settings

Configure the following region-related settings.

You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.

Property	Description
`aws.region`	This value should already be defined for your AWS integration.

Configure roles

Through the AWS console, you can define and manage the policies for the IAM roles including those associated with your EC2 instance.

Attach policy to EC2 instance role

When you are running the Trifacta platform on an EC2 instance, you can leverage your enterprise IAM roles to manage permissions on the EC2 instance. When this type of authentication is enabled, Trifacta platform administrators can apply a role to the EC2 instance where the platform is running. The instance profile that is attached to the EC2 instances must have have the following secrets permission so that Trifacta platform can read and store secrets in Secrets Manager.

For more information on IAM roles for EC2, see *http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html*.

Steps:

The following permissions should be set in the EC2 instance:

1. Log in to the AWS console.
2. Add the following policy to the instance profile corresponding to the EC2 instance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:PutSecretValue",
        "secretsmanager:CreateSecret",
        "secretsmanager:DeleteSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:TagResource"
      ],
      "Resource": "arn:aws:secretsmanager:<aws.region>:<your_aws_account_id>:secret:Trifacta/"
<metadata.environment>/*"
    }
  ]
}
```

Resource value	Description
`<aws.region>`	The region for which the access is provided. Verify that this value is set to `aws.`
`<metadata.environment>`	Differentiates the secrets between a test and production environment. You can set this value as per your requirements.

3. Save the IAM role definition.

NOTE: Trifacta platform recommends to avoid deleting secrets from the AWS console, as they cannot be restored for a period of 7 days. During this period, you cannot create secrets using the same name. As a result, existing jobs may fail without secrets available for a specific user, and Trifacta platform cannot create a new secret with the same name. However, you can restore secrets through AWS CLI.

For more information on the AWS CLI for Secrets Manager, see *https://awscli.amazonaws.com/v2/documentation/api/latest/reference/secretsmanager/index.html*.