# **Issue**

I want to connect to an Amazon Relational Database Service (Amazon RDS) DB instance using AWS Identity and Access Management (IAM) credentials instead of native authentication methods. How can I do that?

# **Short Description**

Amazon RDS users can connect to an RDS DB instance or cluster with IAM user or role credentials and an authentication token. IAM database authentication is more secure than native authentication methods in the following ways:

* IAM database authentication tokens are generated using your AWS access keys. You don't need to store database user credentials.
* Authentication tokens have a lifespan of 15 minutes, so you don't need to enforce password resets.
* IAM database authentication requires an SSL connection, so all data transmitted to and from your RDS DB instance is encrypted.
* If your application is running on Amazon Elastic Compute Cloud (Amazon EC2), you can use EC2 instance profile credentials to access the database. You don't need to store database passwords on your instance.

**Overview:**

To set up IAM database authentication using IAM roles, follow these steps:

1. Enable IAM DB authentication on the RDS DB instance.
2. Connect to an EC2 instance and install the MySQL server package.
3. Create a database user account that uses an AWS authentication token.
4. Create an IAM role allowing Amazon RDS access.
5. Add an IAM policy that maps the database user to the IAM role.
6. Attach the IAM role to the EC2 instance.
7. Generate an AWS authentication token to identify the IAM role.
8. Download the SSL root certificate file or certificate bundle file.
9. Connect to the RDS DB instance using IAM role credentials and the authentication token.

**Note:** IAM database authentication is available only for certain database engines and instance types. For the list of supported engines and instances, see [Availability for IAM Database Authentication](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html#UsingWithRDS.IAMDBAuth.Availability).

# **Prerequisites**

Before you begin this procedure, be sure you have launched the following:

* An RDS DB instance that [supports IAM database authentication](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html#UsingWithRDS.IAMDBAuth.Availability).
* An EC2 instance to connect to the database.

**Enable IAM DB authentication on the RDS DB instance**

To enable IAM database authentication, you can use the AWS Management Console, AWS Command Line Interface (AWS CLI), or the Amazon RDS API.

For instructions, see [Enabling and Disabling IAM Database Authentication](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.Enabling.html).

**Note:**On the **Modify DB Instance** page, under **Maintenance**, you can select **Apply Immediately**to enable IAM database authentication immediately. Depending on other pending modifications, choosing **Apply Immediately** might cause downtime.

**Connect to an EC2 instance and install the MySQL server package**

[Connect to your EC2 instance](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html). If your instance is using an Ubuntu or Amazon Linux Amazon Machine Image (AMI), follow these steps to install the MySQL server package:

**Ubuntu AMI**  
1.    Run this command to install the MySQL package:

$ sudo apt-get install mysql-server

2.    Run this command to set up a root password and to remove the insecure features from your installation:

$ sudo mysql\_secure\_installation

3.    Run this command to start MySQL server at every boot:

$ sudo chkconfig mysqld on

4.    Run this command to start the MySQL server:

$ sudo service mysqld start

**Amazon Linux AMI**  
1.    Run this command to install the MySQL package:

$ sudo yum install mysql-server -y

2.    Run this command to set up a root password and to remove the insecure features from your installation:

$ sudo mysql\_secure\_installation

3.    Run this command to start MySQL server at every boot:

$ sudo chkconfig mysqld on

4.    Run this command to start the MySQL server:

$ sudo service mysqld start

**Create a database user account that uses an AWS authentication token**

1.    From your EC2 instance, connect to the RDS DB instance by running this command. Be sure to enter the master password to log in.

$ mysql -h {Instance endpoint} -u -u$DBUSER -p"$DBPASS"

2.    Run this command to create a database user account that will use an AWS authentication token instead of a password:

CREATE USER {dbusername} IDENTIFIED WITH AWSAuthenticationPlugin as 'RDS';

3.    Optionally, run this command to require the user to connect to the database using SSL:

GRANT USAGE ON \*.\* TO '{dbusername}'@'%'REQUIRE SSL;

4.  Run, the below command to verify the user plugin and authentication:

SELECT host, user, plugin, authentication\_string FROM mysql.user;

5. Run the “exit” command to close MySQL. Then, log off from the instance.

**Create an IAM role allowing Amazon RDS access**

1. Open the[IAM console.](https://console.aws.amazon.com/iam/) Then, choose**Roles** from the navigation pane.
2. Choose **Create role**.
3. Choose **AWS service**, and then choose **EC2**.
4. For **Select your use case**, choose **EC2**, and then choose **Next: Permissions**.
5. In the search bar, type “RDS”. Then, select **AmazonRDSFullAccess** or a custom RDS IAM policy that grants fewer privileges.
6. Choose **Next: Review.**
7. For **Role Name**, type a name for this IAM role.
8. Choose **Create Role**.

**Add an IAM policy that maps the database user to the IAM role**

1. From the IAM role list, open your newly created IAM role.
2. Choose **Add inline policy**.
3. Enter the policy from [Creating and Using an IAM Policy for IAM Database Access](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.IAMPolicy.html).   
   **Note:** Be sure to edit the "Resource" value with the details of your database resources, such as your DB instance identifier and database user name.
4. Choose **Review policy**.
5. For **Name**, type a policy name.
6. Choose **Create policy**.

**Attach the IAM role to the EC2 instance**

1. Open the[Amazon EC2 console](https://console.aws.amazon.com/ec2/).
2. Choose the EC2 instance you will use to connect to Amazon RDS.
3. [Attach your newly created IAM role](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html#attach-iam-role)to the EC2 instance.
4. Reconnect to your EC2 instance using [SSH](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html).

**Generate an AWS authentication token to identify the IAM role**

After you connect to your EC2 instance, run the following AWS CLI command to [generate an authentication token](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.Connecting.AWSCLI.html#UsingWithRDS.IAMDBAuth.Connecting.AWSCLI.AuthToken). Copy and store the authentication token for later use.

**Note:**This token expires within 15 minutes of creation.

$ aws rds generate-db-auth-token --hostname {db or cluster endpoint} --port 3306 --username {IAM user or instance profile}

**Download the SSL root certificate file or certificate bundle file**

Run this command to download the root certificate that works for all regions:

$ wget https://s3.amazonaws.com/rds-downloads/rds-ca-2015-root.pem

If your application does not accept certificate chains, run the following command to download the certificate bundle that includes both the old and new root certificates:

$ wget https://s3.amazonaws.com/rds-downloads/rds-combined-ca-bundle.pem

**Note:**For Windows platform applications that need a PKCS7 file, see [Using SSL to Encrypt a Connection to a DB Instance](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.SSL.html) to download the appropriate certificate.

**Connect to the RDS DB instance using IAM role credentials and the authentication token**

After you download the certificate file, run the following command to connect to the RDS DB instance with SSL using the MySQL utility.

**Note:** If you're connecting to instances in an Amazon Aurora DB cluster, you can connect to one of these endpoints: the cluster endpoint, the reader endpoint, or the instance endpoint.

$ mysql -h {db or cluser endpoint} --ssl-ca={certificate file name with complete path} --ssl-verify-server-cert -u {dbusername2} -p"{authenticationtoken}" --enable-cleartext-plugin

# **Related Information**

[IAM Database Authentication for MySQL and Amazon Aurora](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html)