**Enabling and Disabling IAM Database Authentication**

By default, IAM database authentication is disabled on DB instances and DB clusters. You can enable IAM database authentication (or disable it again) using the AWS Management Console, AWS CLI, or the Amazon RDS API.

**Topics**

* [AWS Management Console](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.Enabling.html#UsingWithRDS.IAMDBAuth.Enabling.Console)
* [AWS CLI](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.Enabling.html#UsingWithRDS.IAMDBAuth.Enabling.CLI)
* [Amazon RDS API](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.Enabling.html#UsingWithRDS.IAMDBAuth.Enabling.API)

**AWS Management Console**

To create a new DB instance or DB cluster with IAM authentication by using the console, see the following workflows:

* For Amazon RDS for MySQL, see [Creating a DB Instance Running the MySQL Database Engine](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateInstance.html).
* For Aurora MySQL, see [Creating an Amazon Aurora DB Cluster](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.CreateInstance.html).

Each of these creation workflows has a **Configure Advanced Settings** page, where you can enable IAM DB authentication. In that page's **Database Options** section, choose **Yes** for **Enable IAM DB Authentication**.

**To enable or disable IAM authentication for an existing DB instance or cluster**

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose either **Instances** or **Clusters**.
3. Choose the DB instance or DB cluster that you want to modify, and then complete one of the following actions:
   * For a DB instance, choose **Instance actions**, and then choose **Modify**.
   * For a DB cluster, choose **Cluster actions**, and then choose **Modify cluster**.
4. In the **Database options** section, for **IAM DB authentication**, choose **Enable IAM DB authentication** or **Disable**, and then choose **Continue**.
5. To apply the changes immediately, choose **Apply immediately**.
6. Choose **Modify DB instance** or **Modify cluster** as appropriate.

**To restore a DB instance or cluster**

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Choose the snapshot you want to restore, and then choose **Restore Snapshot** from **Snapshot Actions**.
4. In the **Settings** section, type an identifier for the DB instance in **DB Instance Identifier**.
5. In the **Database options** section, for **IAM DB authentication**, choose **Enable IAM DB authentication** or **Disable**.
6. Choose **Restore DB Instance**.

**AWS CLI**

To create a new DB instance or DB cluster with IAM authentication by using the AWS CLI, use one of the following commands:

* [create-db-instance](http://docs.aws.amazon.com/cli/latest/reference/rds/create-db-instance.html) for Amazon RDS MySQL
* [create-db-cluster](http://docs.aws.amazon.com/cli/latest/reference/rds/create-db-cluster.html) for Aurora MySQL

Specify the --enable-iam-database-authentication option, as shown in the following example.

aws rds create-db-instance \

--db-instance-identifier *mydbinstance* \

--db-instance-class *db.m3.medium* \

--engine *MySQL* \

--allocated-storage *20* \

--master-username *masterawsuser* \

--master-user-password *masteruserpassword* \

**--enable-iam-database-authentication**

For an existing DB instance or DB cluster, use one of the following AWS CLI commands:

* [modify-db-instance](http://docs.aws.amazon.com/cli/latest/reference/rds/modify-db-instance.html) for Amazon RDS MySQL
* [modify-db-cluster](http://docs.aws.amazon.com/cli/latest/reference/rds/modify-db-cluster.html) for Aurora MySQL

Specify either the --enable-iam-database-authentication or --no-enable-iam-database-authentication option, as appropriate.

By default, Amazon RDS modifies the DB instance during the next maintenance window. If you want to override this and enable IAM DB authentication as soon as possible, use the --apply-immediatelyparameter.

The following example shows how to immediately enable IAM authentication for an existing DB instance.

aws rds modify-db-instance \

--db-instance-identifier *mydbinstance* \

**--apply-immediately** \

**--enable-iam-database-authentication**

If you are restoring a DB instance or DB cluster, use one of the following AWS CLI commands:

* aws rds [restore-db-instance-to-point-in-time](http://docs.aws.amazon.com/cli/latest/reference/rds/restore-db-instance-to-point-in-time.html)
* aws rds [restore-db-instance-from-db-snapshot](http://docs.aws.amazon.com/cli/latest/reference/rds/restore-db-instance-from-db-snapshot.html)

The IAM database authentication setting defaults to that of the source snapshot. To change this setting, set the --enable-iam-database-authentication or --no-enable-iam-database-authenticationoption, as appropriate.

**Amazon RDS API**

For a new DB instance or DB cluster, use one of the following API actions:

* [CreateDBInstance](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_CreateDBInstance.html) for Amazon RDS MySQL
* [CreateDBCluster](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_CreateDBCluster.html) for Aurora MySQL

Set the EnableIAMDatabaseAuthentication parameter to true.

For an existing DB instance or DB cluster, use one of the following API actions:

* [ModifyDBInstance](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_ModifyDBInstance.html) for Amazon RDS MySQL
* [ModifyDBCluster](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_ModifyDBCluster.html) for Aurora MySQL

Set the EnableIAMDatabaseAuthentication to true to enable IAM authentication, or false to disable it.

If you are restoring a DB instance or DB cluster, use one of the following API actions:

* [RestoreDBInstanceToPointInTime](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_RestoreDBInstanceToPointInTime.html)
* [RestoreDBInstanceFromDBSnapshot](http://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_RestoreDBInstanceFromDBSnapshot.html)

The IAM database authentication setting defaults to that of the source snapshot. To change this setting, set the EnableIAMDatabaseAuthentication to true to enable IAM authentication, or false to disable it.

**IAM Database Authentication for MySQL and Amazon Aurora**

With Amazon RDS for MySQL or Aurora with MySQL compatibility, you can authenticate to your DB instance or DB cluster using AWS Identity and Access Management (IAM) database authentication. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

An *authentication token* is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.

IAM database authentication provides the following benefits:

* Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
* You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance or DB cluster.
* For applications running on Amazon EC2, you can use EC2 instance profile credentials to access the database instead of a password, for greater security.

**Availability for IAM Database Authentication**

IAM database authentication is available for the following database engines and instance classes:

* MySQL 5.6, minor version 5.6.34 or higher. All instance classes are supported, except for db.m1.small.
* MySQL 5.7, minor version 5.7.16 or higher. All instance classes are supported, except for db.m1.small.
* Aurora with MySQL compatibility, version 1.10 or higher. All instance classes are supported, except for db.t2.small.

**Limitations for IAM Database Authentication**

With IAM database authentication, you are limited to a maximum of 20 new connections per second. If you are using a db.t2.micro instance class, the limit is 10 connections per second.

The Amazon RDS for MySQL and Aurora MySQL database engines do not impose any limits on authentication attempts per second. However, when you use IAM database authentication, your application must generate an authentication token. Your application then uses that token to connect to the DB instance or cluster. If you exceed the maximum new-connection-per-second limit, then the extra overhead of IAM database authentication can cause connection throttling. The extra overhead can even cause existing connections to drop.

We recommend the following:

* Use IAM database authentication as a mechanism for temporary, personal access to databases.
* Don't use IAM database authentication if your application requires more than 20 new connections per second.
* Use IAM database authentication only for workloads that can be easily retried.

**Note**

For information about the maximum total connections for MySQL, see see [Maximum MySQL connections](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html#USER_ConnectToInstance.max_connections). For information about the maximum total connections for Aurora MySQL, see[Maximum Connections to an Aurora MySQL DB Instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/AuroraMySQL.Managing.Performance.html#AuroraMySQL.Managing.MaxConnections).

**Summary:**

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.

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 that allows 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 the database engines and instance types described before.