

Cloud Computing Lab

Experiment 7

Understanding the concept of cloud datastore.

Study and implement the various Database As a Service provided by AWS (Amazon RDS, Amazon Dynamo DB etc)

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

School of Computer Engineering

Amazon Elastic Compute Cloud (EC2)



Database on AWS

Amazon RDS, Dynamo DB












Database on Aws



- ❑ Purpose-built
- ❑ Performance at scale
- ❑ Fully managed
- ❑ Secure & highly available

Database on Aws

Database services

Database type	Use cases	AWS service
Relational	Traditional applications, ERP, CRM, e-commerce	 Amazon Aurora  Amazon RDS  Amazon Redshift
Key-value	High-traffic web apps, e-commerce systems, gaming applications	 Amazon DynamoDB
In-memory	Caching, session management, gaming leaderboards, geospatial applications	 Amazon ElastiCache for Memcached  Amazon ElastiCache for Redis
Document	Content management, catalogs, user profiles	 Amazon DocumentDB (with MongoDB compatibility)
Wide column	High scale industrial apps for equipment maintenance, fleet management, and route optimization	 Amazon Keyspaces (for Apache Cassandra)
Graph	Fraud detection, social networking, recommendation engines	 Amazon Neptune
Time series	IoT applications, DevOps, industrial telemetry	 Amazon Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	 Amazon QLDB

Amazon Relational Database Service (RDS)



- Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

Amazon DynamoDB



- Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-active, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

Case study



- Present a Comparative study on Amazon RDS and Amazon Dynamo DB based on usability, performance, scale and cost.

Create Amazon RDS instance

Amazon RDS is available on several database instance types - optimized for memory, performance or I/O - and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server.

Our experiment focuses on MySQL.

Create and Connect to a MySQL Database

Step : Create a MySQL DB Instance

- a. In the top right corner of the Amazon RDS console, select the Region in which you want to create the DB instance. (us-east 1)
- b. Click on Create database in the Create database section.
- c. You now have options to select your engine. For this tutorial, click the MySQL icon, leave the default value of edition and engine version, and select the Free Tier template.
- d. You will now configure your DB instance. The list below shows the example settings you can use for this tutorial:

Create Amazon RDS instance

Settings:

- **DB instance identifier:** Type a name for the DB instance that is unique for your account in the Region that you selected. For this tutorial, we will name it `rds-mysql-10minTutorial`.
- **Master username:** Type a username that you will use to log in to your DB instance. We will use `masterUsername` in this example.
- **Master password:** Type a password that contains from 8 to 41 printable ASCII characters (excluding `/`, `"`, and `@`) for your master user password.
- **Confirm password:** Retype your password

Instance specifications:

- **DB instance class:** Select *db.t2.micro --- 1vCPU, 1 GIB RAM*. This equates to 1 GB memory and 1 vCPU. To see a list of supported instance classes, see [Amazon RDS Pricing](#).
- **Storage type:** Select *General Purpose (SSD)*. For more information about storage, see [Storage for Amazon RDS](#).
- **Allocated storage:** Select the default of 20 to allocate 20 GB of storage for your database. You can scale up to a maximum of 64 TB with Amazon RDS for MySQL.
- **Enable storage autoscaling:** If your workload is cyclical or unpredictable, you would enable storage autoscaling to enable RDS to automatically scale up your storage when needed. This option does not apply to this tutorial.
- **Multi-AZ deployment:** Note that you will have to pay for Multi-AZ deployment. Using a Multi-AZ deployment will automatically provision and maintain a synchronous standby replica in a different Availability Zone. For more information, see [High Availability Deployment](#).

Create Amazon RDS instance

e. You are now on the Connectivity section where you can provide information that RDS needs to launch your MySQL DB instance. The list below shows settings for our example DB instance.

Connectivity

- **Virtual Private Cloud (VPC):** Select *Default VPC*. For more information about VPC, see [Amazon RDS and Amazon Virtual Private Cloud \(VPC\)](#).

Additional connectivity configurations

- **Subnet group:** Choose the *default* subnet group. For more information about subnet groups, see [Working with DB Subnet Groups](#).
- **Public accessibility:** Choose *Yes*. This will allocate an IP address for your database instance so that you can directly connect to the database from your own device.
- **VPC security groups:** Select *Create new VPC security group*. This will create a security group that will allow connection from the IP address of the device that you are currently using to the database created.
- **Availability zone:** Choose *No preference*. See [Regions and Availability Zones](#) for more details.
- **Port:** Leave the default value of 3306.

Create Amazon RDS instance

In the **Additional configurations** section:

Database options

- **Database name:** Type a database name that is 1 to 64 alpha-numeric characters. If you do not provide a name, Amazon RDS will not automatically create a database on the DB instance you are creating.
- **DB parameter group:** Leave the default value. For more information, see [Working with DB Parameter Groups](#).
- **Option group:** Leave the default value. Amazon RDS uses option groups to enable and configure additional features. For more information, see [Working with Option Groups](#).

Encryption

This option is not available in the free tier. For more information, see [Encrypting Amazon RDS Resources](#).

Backup

- **Backup retention period:** You can choose the number of days to retain the backup you take. For this tutorial, set this value to *1 day*.
- **Backup window:** Use the default of *No preference*.

Monitoring

- **Enhanced Monitoring:** Select *Disable enhanced monitoring* to stay within the free tier. Enabling enhanced monitoring will give you metrics in real time for the operating system (OS) that your DB instance runs on. For more information, see [Viewing DB Instance Metrics](#).

Maintenance

- **Auto minor version upgrade:** Select *Enable auto minor version upgrade* to receive automatic updates when they become available.
- **Maintenance Window:** Select *No preference*.

Deletion protection

Clear *Enable deletion protection* for this tutorial. When this option is enabled, you're prevented from accidentally deleting the database.

Create Amazon RDS instance



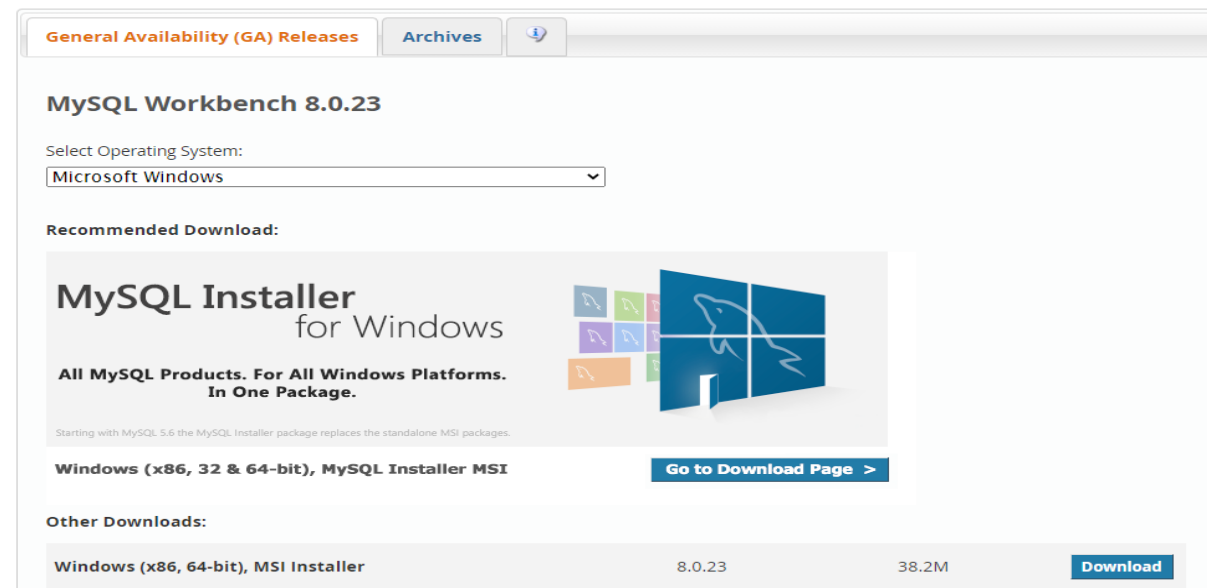
- Click Create database.

Your DB Instance is now being created. Click View Your DB Instances.

Connect with Amazon RDS instance

- Connect with your DB instance.
 - ▣ Download SQL Client
 - Go to the [Download MySQL Workbench](https://dev.mysql.com/downloads/workbench/) page to download and install MySQL Workbench.

(<https://dev.mysql.com/downloads/workbench/>)



Connect with Amazon RDS instance

□ Connect with MySQL Database.

a. Launch the MySQL Workbench application and go to Database > Connect to Database (Ctrl+U) from the menu bar.

b. A dialog box appears. Enter the following:

Hostname: You can find your hostname on the Amazon RDS console.

Port: The default value should be 3306.

Username: Type in the username you created for the Amazon RDS database. In this tutorial, it is '*masterUsername*'.

Password: Click *Store in Vault* and enter the password that you used when creating the Amazon RDS database.

c. Click OK

Connect with Amazon RDS instance

The screenshot displays the AWS Management Console interface for an Amazon RDS instance named `rds-mysql-10mintutorial`. The left sidebar shows the navigation menu with options like Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Recommendations. The main content area shows the instance details under the 'Summary' tab, including the DB identifier, CPU, Role, and Instance. The 'Connectivity & security' tab is selected, showing the 'Endpoint & port' section. The endpoint is `rds-mysql-10mintutorial.cbhj9h0miwe5.us-east-1.rds.amazonaws.com` and the port is `3306`. A red box highlights the endpoint, and a yellow box highlights the port. A 'Connect to Database' dialog box from MySQL Workbench is overlaid on the console. The dialog shows the 'Parameters' tab with fields for Hostname, Username, Password, and Default Schema. The Hostname field is populated with the endpoint from the console, and the Port field is populated with `3306`. A red arrow points from the endpoint in the console to the Hostname field in the dialog, and a yellow dashed arrow points from the port in the console to the Port field in the dialog. The dialog also has tabs for 'SSL' and 'Advanced', and buttons for 'Cancel' and 'OK'.

Amazon RDS `rds-mysql-10mintutorial`

Summary

DB identifier	CPU
<code>rds-mysql-10mintutorial</code>	1.4

Role	Current act
Instance	0 C

Connectivity & security

Endpoint & port

Endpoint: `rds-mysql-10mintutorial.cbhj9h0miwe5.us-east-1.rds.amazonaws.com`

Port: `3306`

Connect to Database

Stored Connection:

Connection Method:

Parameters | SSL | Advanced

Hostname: `rds-mysql-10mintutorial.cbhj9h0miwe5.us-east-1.rds.amazonaws.com` Port: `3306`

Username: `masterUsername`

Password: Store in Keychain... Clear

Default Schema:

Cancel OK

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

MySQL Workbench

Reference

16

<https://docs.aws.amazon.com/whitepapers/latest/aws-overview/database.html>

<https://aws.amazon.com/getting-started/hands-on/create-mysql-db/>

Thank you