



HI-TECH INSTITUTION

CORPORATE CAREER ENHANCEMENT TRAININGS

OUR ROOT LEVEL
TRAINING WILL
GIVE YOU BETTER
GROWTH





ABOUT US

Our Vision:

To provide better training by full filling the requirements of our trainee.

Our Mission:

We always ensure to give practical based training. And we make the candidates to get good hands-on experience on any platform.

Philosophy:

Our Root Level Training Will give you Better Growth.

We successfully survived around 5 years in the IT field. Started this is as small Training room. But now we are having 5 branches across India.

Certified Trainers taking the session on various domain with any level of doubts clarification.

For More Details: www.hitechins.in

Write feedback to operations@hitechins.in

Amazon Relational Database Service (Amazon RDS)

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Overview of Amazon RDS

Why do you want a managed relational database service? Because Amazon RDS takes over many of the difficult or tedious management tasks of a relational database:

- When you buy a server, you get CPU, memory, storage, and IOPS, all bundled together. With Amazon RDS, these are split apart so that you can scale them independently. If you need more CPU, less IOPS, or more storage, you can easily allocate them.
- Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
- To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances, and it restricts access to certain system procedures and tables that require advanced privileges.
- You can have automated backups performed when you need them, or manually create your own backup snapshot. You can use these backups to restore a database. The Amazon RDS restore process works reliably and efficiently.
- You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use MySQL, MariaDB, or PostgreSQL Read Replicas to increase read scaling.
- You can use the database products you are already familiar with: MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server, and the new, MySQL-compatible Amazon Aurora DB engine.
- In addition to the security in your database package, you can help control who can access your RDS databases by using AWS Identity and Access Management (IAM) to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud.

DB Instances

The basic building block of Amazon RDS is the *DB instance*. A DB instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a stand-alone database instance. You can create and modify a DB instance by using the AWS Command Line Interface, the Amazon RDS API, or the AWS Management Console.

Each DB instance runs a *DB engine*. Amazon RDS currently supports the **MySQL, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server DB engines**. Each DB engine has its own supported features, and each version of a DB engine may include specific features. Additionally, each DB

engine has a set of parameters in a DB parameter group that control the behavior of the databases that it manages.

How You Are Charged for Amazon RDS

When you use Amazon RDS, you pay only for what you use, and there are no minimum or setup fees. You are billed according to the following criteria.

- **Instance class** – Pricing is based on the class (for example, micro, small, large, xlarge) of the DB instance consumed.
- **Running time** – You are billed by the instance-hour, which is equivalent to a single instance running for an hour. For example, both a single instance running for two hours and two instances running for one hour consume two instance-hours. If a DB instance runs for only part of an hour, you are charged for a full instance-hour.
- **Storage** – The storage capacity that you have provisioned to your DB instance is billed per GiB per month. If you scale your provisioned storage capacity within the month, your bill is pro-rated.
- **I/O requests per month** – Total number of storage I/O requests that you have made in a billing cycle.
- **Backup storage** – Backup storage is the storage that is associated with automated database backups and any active database snapshots that you have taken. Increasing your backup retention period or taking additional database snapshots increases the backup storage consumed by your database. Amazon RDS provides backup storage up to 100% of your provisioned database storage at no additional charge. For example, if you have 10 GiB-months of provisioned database storage, we provide up to 10 GiB-months of backup storage at no additional charge. Most databases require less raw storage for a backup than for the primary dataset, so if you don't keep multiple backups, you never pay for backup storage. Backup storage is free only for active DB instances.
- **Data transfer** – Internet data transfer in and out of your DB instance.

Database Engine-Specific Topics

- Amazon Aurora on Amazon RDS
- MariaDB on Amazon RDS
- Microsoft SQL Server on Amazon RDS
- MySQL on Amazon RDS
- Oracle on Amazon RDS
- PostgreSQL on Amazon RDS







To create a DB instance running the Microsoft SQL Server DB engine

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the top right corner of the Amazon RDS console, choose the region in which you want to create the DB instance.
3. In the navigation pane, choose **Instances**.
4. Choose **Launch DB Instance**.

Select the Database Engine which we required

Select Engine

To get started, choose a DB Engine below and click Select.

	SQL Server Express Microsoft SQL Server Express Edition Select
	Microsoft SQL Server Express Edition is an affordable database management system that supports database sizes up to 10 GB. Refer to Microsoft's web site for more details.
	SQL Server Web Microsoft SQL Server Web Edition Select
	Microsoft SQL Server Web Edition is an efficient and affordable database management system. In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services. Refer to the AWS Service Terms for more details.
	
	SQL Server SE Microsoft SQL Server Standard Edition Select
	Microsoft SQL Server Standard Edition includes core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
	SQL Server EE Microsoft SQL Server Enterprise Edition Select
	Microsoft SQL Server Enterprise Edition delivers comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.

[Cancel](#)

- Choose the SQL Server icon, and then choose **Select** for the **SQL Server Express** edition.

6. The **Specify DB Details** page appears.

Specify DB Details

Free Tier

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

The database engine or edition you selected is not eligible for RDS Free Tier.

Instance Specifications

DB Engine	sqlserver-se
License Model	license-included
DB Engine Version	12.00.4422.0.v1
DB Instance Class	db.m4.large — 2 vCPU, 8 GiB RAM
Time Zone (Optional)	Pacific Standard Time
Multi-AZ Deployment	No
Storage Type	General Purpose (SSD)
Allocated Storage*	200 GB

Scaling storage after launching a DB Instance is currently not supported for SQL Server. You may want to provision storage based on anticipated future storage growth.

Settings

DB Instance Identifier*	
Master Username*	
Master Password*	
Confirm Password*	

* Required

Cancel Previous **Next Step**


7. On the **Specify DB Details** page, provide the information for your DB instance as shown in the following table:

For This Parameter	Do This
License Model	Choose license-included to use the general license agreement for Microsoft SQL Server.
DB Engine Version	Choose the most recent version of SQL Server available in the list.
DB Instance Class	Choose db.t2.micro . This instance class is appropriate for testing.
Time Zone	Do not choose a time zone. If you don't choose a time zone, your DB instance uses the default time zone.
Storage Type	Choose the storage type General Purpose (SSD) .
Allocated Storage	Type 20 to allocate 20 GiB of storage for your database. There is a warning that you should consider allocating more storage, but since this is a sample DB instance, 20 GiB is sufficient.
DB Instance Identifier	Type sample-instance .
Master Username	Type a name that you will use as the master user name to log on to your DB Instance with all database privileges. The master user name is a SQL Server Authentication login.
Master Password and Confirm Password	Type a password for your master user password. It must contain between 8 and 128 printable ASCII characters (excluding /, ", and @).

8. Choose **Next** to continue.
9. The **Configure Advanced Settings** page appears.

Configure Advanced Settings

Network & Security

VPC* Default VPC (vpc) 

Subnet Group default

Publicly Accessible Yes

Availability Zone No Preference

VPC Security Group(s) Create new Security Group
default (VPC)

Database Options

Database Port 1433

DB Parameter Group default:sqlserver-se-12.0

Option Group default:sqlserver-se-12-00

Copy Tags To Snapshots ☐

Enable Encryption No

Backup

Backup Retention Period 7 days

Backup Window No Preference

Monitoring

Enable Enhanced Monitoring No

Maintenance

Auto Minor Version Upgrade Yes

Maintenance Window No Preference

* Required

[Cancel](#) [Previous](#) [Launch DB Instance](#)

On the **Configure Advanced Settings** page, provide the information for your DB instance as shown in the following table:

For This Parameter

VPC
Subnet Group
Publicly Accessible
Availability Zone
VPC Security Group
Database Port

Do This

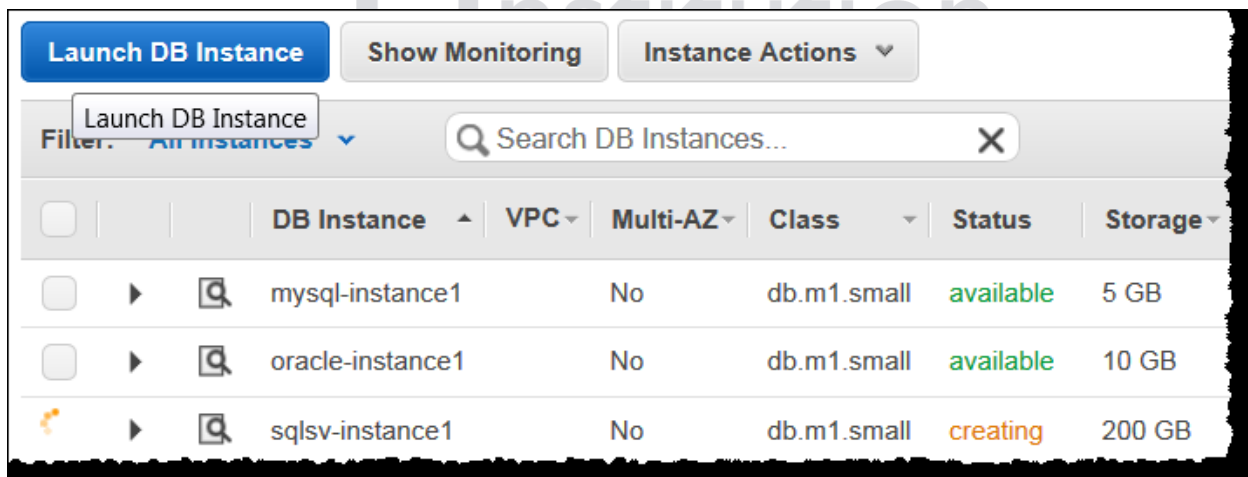
Choose **Create new VPC**.
 Choose **Create new DB Subnet Group**.
 Choose **Yes**.
 Choose **No Preference**.
 Choose **Create new Security Group**.
 Leave the default value of **1433** unless you have a specific port you want to access the database through. SQL Server installations default to port 1433, but in some cases a firewall might block this port. If in doubt, ask your network administrator what port you should use.
 Leave the default value.
 Leave the default value.
 Leave this setting unselected.
 Choose 7.
 Choose **No Preference**.
 Choose **No**.
 Choose **Yes**.
 Choose **No Preference**.

DB Parameter Group
Option Group
Copy Tags To Snapshots
Backup Retention Period

Backup Window
Enable Enhanced Monitoring
Auto Minor Version Upgrade
Maintenance Window

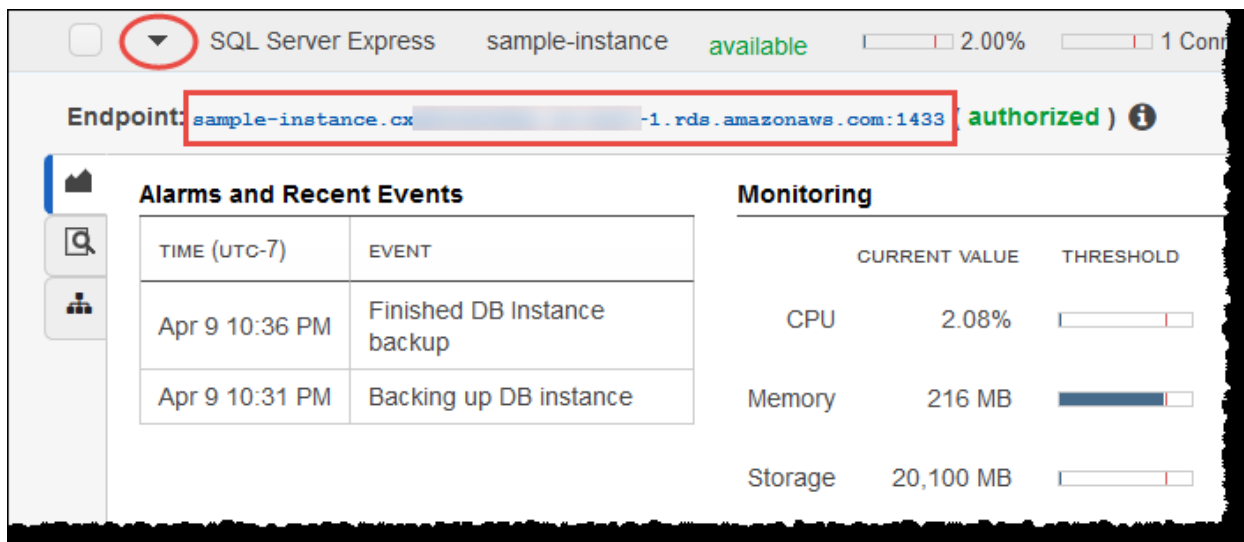
10. Choose **Launch DB Instance**.
11. Choose **View Your DB Instances**.

On the RDS console, the new DB instance appears in the list of DB instances. The DB instance has a status of **creating** until the DB instance is ready to use. When the state changes to **available**, you can connect to the DB instance. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.



To connect to a DB Instance using SSMS

1. Find the DNS name and port number for your DB Instance.
 - Open the RDS console and then choose **Instances** to display a list of your DB instances.
 - Choose the row for your SQL Server DB instance to display the summary information for the instance.



- Copy the endpoint. The **Endpoint** field has two parts separated by a colon (:). The part before the colon is the DNS name for the instance, the part following the colon is the port number. Copy both parts.
2. Start SQL Server Management Studio.
- The **Connect to Server** dialog box appears.



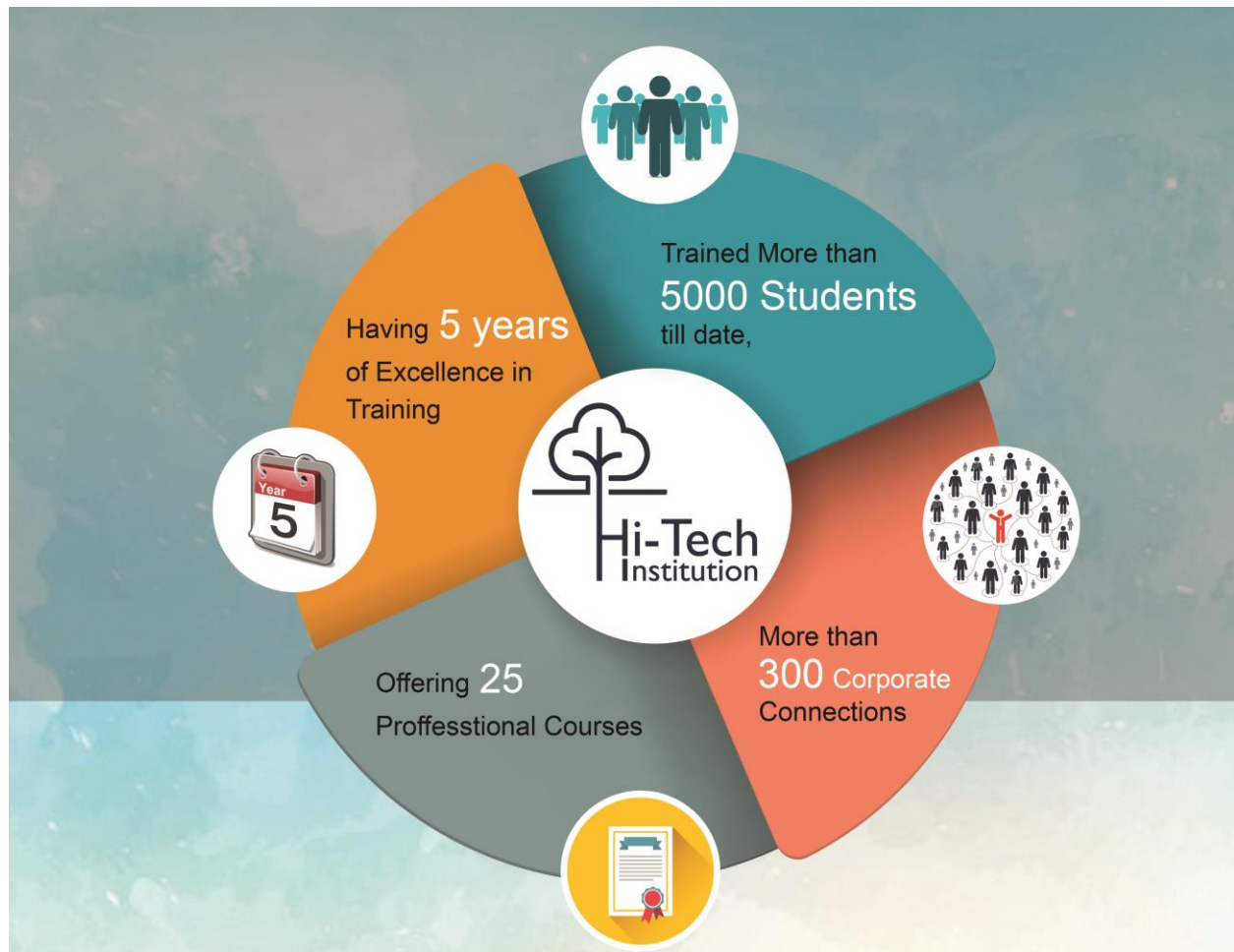
3. Provide the information for your sample DB instance.

- For **Server type**, choose **Database Engine**.
- For **Server name**, type or paste the DNS name and port number of your sample DB Instance, separated by a comma.

Important

- Change the colon between the DNS name and port number to a comma.
- For example, your server name should look like the following:
- For **Authentication**, choose **SQL Server Authentication**.
- For **Login**, type the master user name you chose earlier for your sample DB instance.
- For **Password**, type the password you chose earlier for your sample DB instance.

4. Choose **Connect**.



TOP RECRUITERS





50%

offer for School or College students

30%

offer for IT Employees

Above offer applicable only technical courses. Terms and conditions apply



operations@hitechins.in



www.hitechins.in



CONTACT US

7092 90 91 92 / 82 20 21 7640

PONDICHERRY

No.32, 100 feet road,
Ellaipillaichavady,
Pondicherry – 605 005,
Nearby Rajiv Gandhi Hospital

TAMBARAM

No.24, Chithi Vinayagar Kovil street,
KamarajNagar, Tambaram Sanatorium,
Chennai – 600 047,
Nearby Sanatorium Railway Station

VELACHERRY

No: 21, Officer Colony,
100 feet road, VijayaNagar,
Velacherry – 600 042,
Nearby Sathya Home Appliances

Locations

Chennai & Pondicherry