

RDS vs. EC2

NOTE: This content will be available as a video course soon! We are making it available as a Learning Path resource in the meantime to help you prepare for the newly updated AWS Certified Solutions Architect - Associate (SAA-C03) exam.

Introduction

Hello, and welcome to this course, which will provide a framework to help you evaluate appropriate database deployment solutions on AWS using either the Amazon Relational Database Service, or RDS, or Amazon Elastic Compute Cloud, or EC2 instances.

Before we get started, I would like to introduce myself. My name is Danny Jessee, and I am one of the trainers here at Cloud Academy, specializing in AWS–Amazon Web Services–and AWS certifications. Feel free to contact me with any questions using the details shown on the screen, or you can always get in touch with us here at Cloud Academy by sending an email to support@cloudacademy.com, where one of our Cloud experts will reply to your question.

Who should attend this course?

This course is for anyone who is responsible for defining and implementing relational database solution architectures in AWS, or anyone who is looking to migrate an existing application that uses a relational database backend to the AWS Cloud.

Learning Objectives

By the end of this course, you'll be able to choose between using Amazon RDS or EC2 instances for your relational database deployments. In many cases, using an AWS-managed RDS database will be the fastest, simplest, and most effective approach. However, depending on your specific requirements, there are scenarios where it makes more sense to install a database on Amazon EC2 instances instead.

Prerequisites

To get the most out of this course, you should have a basic understanding of relational databases, Amazon RDS, and Amazon EC2. For more information about these services, I encourage you to check out these lectures:

Compute Fundamentals For AWS: EC2 - Elastic Compute Cloud

<https://cloudacademy.com/course/compute-fundamentals-for-aws/ec2-elastic-compute-cloud/>

Database Fundamentals for AWS - Part 1 of 2: Amazon Relational Database Service

<https://cloudacademy.com/course/database-fundamentals-part-one-1064/amazon-relational-database-service/>

Feedback

Here at Cloud Academy, we strive to keep our content current to provide the best training available. If you have any feedback, positive or negative, or if you notice anything that needs to be updated or corrected for the next release cycle, please reach out to us at support@cloudacademy.com. Thank you!

RDS vs. EC2

Hello, and welcome to this lecture, where I will discuss how to choose between using the Amazon Relational Database Service, or RDS, or Elastic Compute Cloud, or EC2 instances for your relational database deployments in the AWS Cloud. Both RDS and EC2 instances allow you to host databases securely within your own Virtual Private Cloud, or VPC. Both give you the ability to scale your databases to meet varying levels of storage, throughput, and performance demand, and both can be configured to operate within the AWS free tier. But that's where the similarities end. In order to decide which option is best for your database, it helps to have a better understanding of the various benefits each approach has to offer. So let's begin by discussing the advantages and disadvantages of both approaches, beginning with RDS.

Amazon RDS

Amazon RDS is an AWS-managed service that allows you to automatically install and provision relational databases in the AWS Cloud using popular open-source and commercial database engines including MySQL, MariaDB, PostgreSQL, and even Oracle and Microsoft SQL Server. RDS also offers Amazon Aurora, which is a MySQL and PostgreSQL-compatible database with up to 5 times the throughput of MySQL, up to 64 TB of auto-scaling SSD storage, and 6-way replication of your data across 3 availability zones, all at a fraction of the cost of a traditional database deployment.

Now what makes RDS appealing for so many use cases is just how simple it is to quickly spin up a new database without having to install or configure any operating systems or database software. And since AWS is fully responsible for the management of your RDS instance, you never have to worry about any of the administrative tasks typically associated with managing a database like configuring backups, security, or high availability. You also never need to worry about patching or hardening the underlying operating system of the servers that host your databases. This makes RDS ideally suited for smaller development shops or any organization that can't afford the expense of a dedicated database administrator, or DBA function. And although you don't have

access to the server that hosts your RDS database, you can still use most standard database administration tools to connect to your database and perform any necessary actions.

So let's quickly run through some of the other advantages of using RDS for your database deployments in the AWS Cloud. When you choose RDS, AWS will automatically provision your database instance along with its associated storage, configure automatic backups, and even automatic minor version upgrades if you choose.

RDS also takes care of encryption for your databases, both in transit and at rest. And when we talk about encryption at rest, this extends beyond just your primary instance storage and also includes your database backups and snapshots as well.

You even have the option to choose between preconfigured templates for production or dev/test deployments. So again, when you don't have the time or resources for a dedicated DBA, or if you just want to get up and running as quickly as possible with your new database, it's really great to let RDS handle all of these things for you.

High availability and scalability

And beyond all of this, many of the supported database engines in RDS will also allow you to configure a highly available multi-AZ deployment with just a single click. And with a multi-AZ deployment, you get a 99.95% uptime SLA. So in a multi-AZ deployment, RDS will automatically create a standby, synchronous replica instance in a different availability zone than your primary instance. RDS will then automatically fail over to that standby instance in the event an infrastructure failure is ever detected on your primary instance. And these failures could include anything from an outage within an availability zone, to a loss of network connectivity, or even a compute or storage failure. And best of all, when a failure does happen, AWS will automatically update the URL associated with your database to point to your standby instance during a failover, then back to your primary instance once it's up and running again. And all of this happens without requiring any manual intervention from an administrator whatsoever.

Now when it comes to scalability, RDS makes it easy to scale out your database deployments by adding asynchronous read replica instances, which can help offload read demand from your primary instance. And you can imagine how much time and effort would be involved for a DBA to have to spin up additional EC2 instances and then configure a solution like database mirroring or a failover cluster, so RDS definitely offers some advantages when it comes to configuring high availability and scalability. To learn more about when to use Multi-AZ deployments and read replicas with RDS, please check out this course:

When to use RDS Multi-AZ & Read Replicas

<https://cloudacademy.com/course/using-rds-multi-az-read-replicas/rds-multi-az/>

And to create your first RDS database, check out this hands-on lab:

Create Your First Amazon RDS Database

<https://cloudacademy.com/lab/create-your-first-amazon-rds-database/>

Amazon EC2

So as you've seen, RDS offers a lot of inherent advantages. It's extremely easy to set up, it's cost effective, and it offloads the responsibility for your more time and labor-intensive database administration tasks to AWS, allowing you to focus on your own business requirements. But there are still scenarios where it makes more sense to leverage EC2 instances to host your relational databases instead of RDS. So now, let's talk about some of the reasons why you may want to choose EC2 instances over RDS.

Amazon EC2 allows you to provision virtual servers on demand that can run a variety of different operating system types and versions, whereas RDS abstracts away the underlying operating system of the database instance. So if your application requires a specific OS type or version for your database, or if you need to run a database that isn't currently supported by RDS such as IBM DB2 or SAP HANA, you'll need to provision your own EC2 instances. You'll also need to use EC2 instances if your application requires you to configure any operating system-specific settings, or install any additional software directly on your database server.

It might also make sense for you to leverage EC2 instances if your organization already has full-time DBAs on staff who need to have access to advanced features and configuration settings within the database. For instance, RDS doesn't support some of the more advanced features in Microsoft SQL Server such as persistent memory devices or its Resource Governor. If your application requires any of these features, you'll have no choice but to provision SQL Server on an EC2 instance.

So as you've seen, most of the reasons you'd want to choose EC2 instances over RDS involve needing to have a greater level of control over your database or the underlying server operating system than what RDS provides. And this could involve anything from controlling exactly when or how you apply patches and perform maintenance on your database, to which ports and protocols you want your database to use. Perhaps you need to configure multiple versions or multiple instances of a particular database engine on a single server. You may even need to set up an advanced RAID and striping configuration for your database's storage using EBS volumes. While all of these would be non-starters with RDS, they're entirely possible when using EC2 instances.

Now just because you choose to use EC2 instances instead of RDS doesn't mean you can't still configure some of the native functionality that RDS provides such as high availability, automatic failover, and scheduled backups for your databases. Just know that setting all of this up with EC2

instances will require significantly more effort than if you just used RDS. But depending on your specific requirements, this may be a worthwhile tradeoff for you.

So that covers the reasons you might want to use EC2 instances for your database deployments instead of RDS. To create your first EC2 instance, please check out these hands-on labs:

Create Your First Amazon EC2 Instance (Linux)

<https://cloudacademy.com/lab/create-your-first-amazon-ec2-instance/>

Create Your First Amazon EC2 Instance (Windows)

<https://cloudacademy.com/lab/create-your-first-amazon-ec2-instance-windows/>

Final Thoughts

So to wrap things up here, you're probably wondering: with all the benefits that RDS has to offer, surely all of this added functionality comes at a much higher cost when compared to just running your database on EC2 instances. Surprisingly, though, costs for database deployments using RDS and EC2 may be more comparable than you might expect. While RDS deployments are typically more expensive than comparable EC2 instance deployments, the difference in cost will vary depending on the specific instance and storage types you choose for your deployments. It's also important to remember that the additional costs associated with those manual database administration tasks will become your responsibility if you choose to use EC2 instances instead of RDS.

So if you're interested in getting a better sense of what these various database deployment options might cost for your applications in the AWS Cloud, you can input your specific numbers into the AWS Pricing Calculator at <https://calculator.aws>, where you can configure pricing estimates for different types of deployments by searching for things like RDS, Oracle, or EC2.

And just a couple of final thoughts: if you already have a dedicated DBA on staff, you'll probably want to use EC2 instances for the cost savings you'll get compared to RDS. You'll also want to use EC2 instances if you need to provision a database type or version that isn't supported by RDS. But in many other cases, RDS will probably be your best option. By reducing the management overhead associated with running your relational databases in the AWS Cloud, RDS allows you to worry less about tedious, labor-intensive administrative tasks, enabling you to focus on the things that deliver value to your business instead.

Summary

Hello, and welcome to the final lecture in this course, where I want to quickly summarize what we've covered.

In this course, we discussed the various features and options you should consider when choosing between Amazon RDS or EC2 instances for your cloud database deployments. RDS makes it easy to quickly set up and operate relational databases in the AWS Cloud using popular database engines such as MySQL, PostgreSQL, Oracle, and Microsoft SQL Server. RDS automatically handles tedious, time-consuming administrative tasks like configuring high availability, backups, patching, and security while abstracting away the underlying server.

On the other hand, EC2 instances allow you to have total control over the underlying server, but require much more care and feeding when it comes to manual administration, both in terms of patching the operating system and database engine, as well as configuring features that are already provided by RDS like high availability, backups, and encryption.

While database deployments on EC2 instances do generally cost less than with RDS, many organizations will make up this cost difference in the labor saved by not needing to perform so many manual administrative tasks. That being said, EC2 instance deployments are still necessary in situations where you need to run a database that isn't supported by RDS. Aside from that, however, there are no hard and fast rules for when you must choose RDS or EC2 instances for a particular database deployment.

That brings me to the end of this lecture, and to the end of this course. You should now be able to describe the pros and cons of database deployments using RDS or EC2 instances and determine which is a more appropriate solution based on a given set of requirements.

Feedback on our courses here at Cloud Academy is valuable to both us as trainers and any students looking to take the same course in the future. If you have any feedback, positive or negative, it would be greatly appreciated if you could email support@cloudacademy.com.

Thank you so much for your time and best of luck with your continued learning!



Danny Jessee, AWS Certification Specialist

danny.jessee@cloudacademy.com | [@dannyjessee](https://twitter.com/dannyjessee) | [linkedin.com/in/dannyjessee](https://www.linkedin.com/in/dannyjessee)