

欢迎学习模块 5:添加数据库层。

模块概览



小节目录

- 1. 架构需求
- 2. 数据库层考虑因素
- 3. Amazon RDS
- 4. Amazon DynamoDB
- 5. 数据库安全控制
- 6. 将数据迁移到 AWS 数据库

演示

• Amazon RDS 自动备份和只读副本

实验

- 指导实验: 创建 Amazon RDS 数据库
- 挑战实验:将数据库迁移到 Amazon RDS



知识测验

本模块包含以下章节:

- 1. 架构需求
- 2. 数据库层考虑因素
- 3. Amazon RDS
- 4. Amazon DynamoDB
- 5. 数据库安全控制
- 6. 将数据迁移到 AWS 数据库

本模块还包括:

- Amazon Relational Database Service (Amazon RDS) 自动备份和只读副本配置的演示。
- 一个指导实验,您可以创建一个 Amazon RDS 数据库,并使用一个简单的 Web 应用程序连接到该数据库。
- 一个挑战实验,挑战您将数据从在 Amazon Elastic Compute Cloud (Amazon EC2) 实例上运行的数据库迁移到 Amazon RDS 数据库。

最后,您需要完成一个知识测验,以测试您对本模块中涵盖的关键概念的理解程度。

模块目标



学完本模块后, 您应该能够:

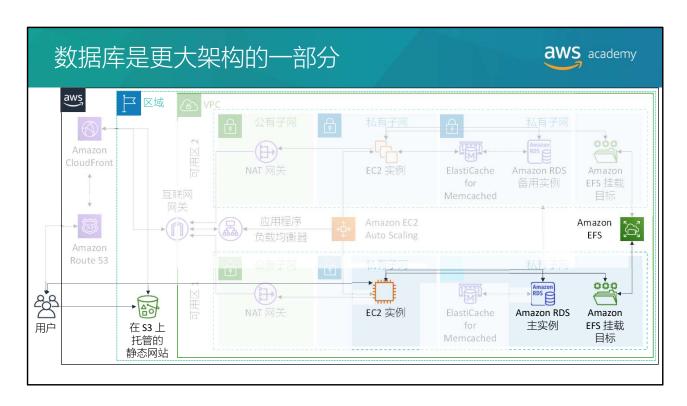
- 比较数据库类型
- 区分托管服务和非托管服务
- 说明何时使用 Amazon Relational Database Service (Amazon RDS)
- 说明何时使用 Amazon DynamoDB
- 描述可用的数据库安全控制
- 描述如何将数据迁移到 Amazon Web Services (AWS) 数据库
- 部署数据库服务器

学完本模块后, 您应该能够:

- 比较数据库类型
- 区分托管服务和非托管服务
- 说明何时使用 Amazon Relational Database Service (Amazon RDS)
- 说明何时使用 Amazon DynamoDB
- 描述可用的数据库安全控制
- 描述如何将数据迁移到 Amazon Web Services (AWS) 数据库
- 部署数据库服务器

模块 5:添加数据库层 第 1 节:架构需求

介绍第1节:架构需求。



在每个模块介绍新功能时,将逐一讨论这张图表中的各个部分。

在本模块中,您将学习如何创建使用 AWS 数据库服务的架构。其中讨论了不同的关系数据库和非关系数据库服务选项,包括 Amazon RDS。在了解到各种选项的基本优势后,您将能够针对数据库服务做出更明智的决策,选择最能满足您的特定架构需求的服务。

咖啡馆业务要求



该咖啡馆需要一种更易于维护的数据库解决方案,并应能够提供持久性、可扩展性和高性能等基本功能。





自从该咖啡馆在其网站上增加网上订餐的功能后,咖啡馆员工就注意到业务量增加了。此外,他们还发现,存储在数据库(该数据库安装在运行 Web 服务器的同一 EC2 实例上)中的订单历史记录提供了有价值的业务信息。Martha 用它来记账,Frank 偶尔会查看它,以便了解每一种甜点应该烘焙多少。

不过,Sofía 有一些担心。数据库需要升级和修补,而她不一定有时间定期做这些工作。 此外,管理数据库是一项专业技能,而她并不想花时间培训其他人来完成这项工作。 同时,她还担心咖啡馆没有按应有的频率进行数据备份。

咖啡馆员工希望减少技术学习投入相关的人力成本,而这些是他们自己管理数据库所必需的。因此,他们决定,使用托管数据库解决方案是有必要的。理想情况下,他们将找到一种提供持久性、可扩展性和高性能等基本功能的解决方案。

在本模块中,您将详细了解 AWS 提供的不同数据库服务,以及这些服务所提供的功能。 在了解了这些可用选项后,您应该能够选择一种能够成功满足这些新业务要求的数据 库解决方案。 模块 5: 添加数据库层

第2节:数据库层考虑因素



介绍第2节:数据库层考虑因素。

数据库考虑因素:可扩展性





可扩展性

需要多大的吞吐量?



总存储要求

所选择的解决方案能够在以后根据需 要扩展吗?



对象大小和类型





持久性

作为一名架构师, 当您考虑哪种类型最适合处理特定的工作负载时, 通常需要在不同的数据库类型之间做出选择。在选择数据库之前, 有一些重要的考虑因素可为您的决策过程提供参考。

首先,考虑可扩展性的重要性。使用传统的本地数据库,即使对于经验丰富的数据库管理员来说,扩展容量也是十分艰难的任务。这可能需要几个小时、几天或几周的时间。在扩展过程中,对数据库性能的影响是不可预测的,而且可能需要停机。然而,适当扩展数据库的重要性怎么强调也不为过。如果数据库预置不足,应用程序可能会停止工作。但是,如果过度预置数据库,就会因采购了不需要的资源而增加前期成本,这违背了AWS架构完善的框架的成本优化原则。

理想的情况是,您可以选择一种数据库解决方案,该解决方案具备处理启动时所需*否吐量*的资源,并且以后能随您的吞吐量需求增加而轻松*扩展*。

另一个值得拥有的功能是以后能随您的吞吐量需求或数据库负载减少而缩减预置的数据库容量。这样,您通过降低预置容量便可立即实现成本节省。自动扩展解决方案可帮助降低成本和人力开销。

数据库考虑因素:存储需求





可扩展性



总存储要求



对象大小和类型



持久性



它是否需要存储 GB、TB 或 PB 级数据?



第二,当您必须选择一个数据库来处理特定类型的工作负载时,应考虑工作负载的存储需求。

需要多大的数据库来满足您的数据需求?它是否需要存储 GB 级、TB 级、PB 级数据?

不同的数据库架构支持的最大数据容量不同。有些数据库设计非常适合传统应用,而有些则非常适合缓存或会话管理。还有一些数据库非常适合物联网 (IoT) 或大数据应用。

了解您的总存储需求对于选择数据库至关重要。



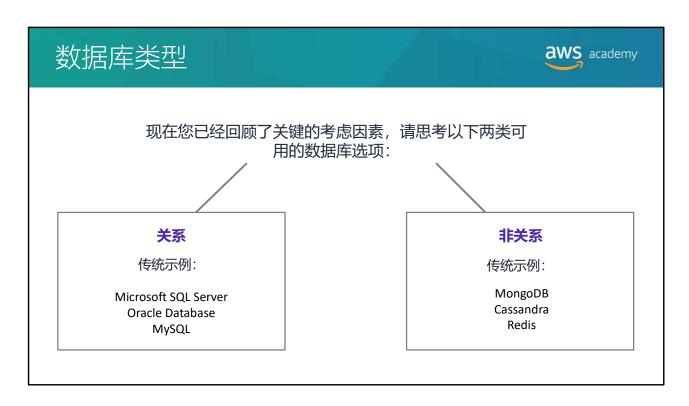
第三,当您为特定工作负载选择数据库时,应考虑必须存储的对象的大小和类型。 您需要存储简单数据结构、大型数据对象,还是两者都需要?



最后,作为您选择数据库的第四个考虑因素,应考虑要存储的数据的持久性要求。

数据持久性是指保证您的数据不会丢失,数据可用性是指在需要时访问数据的能力。您需要什么级别的数据持久性和数据可用性?如果您要存储的数据对您的业务至关重要,您应该选择一种数据库解决方案,将这些数据的多份冗余副本存储在多个地理上分离的物理位置。这种解决方案通常会导致成本增加,因此平衡业务需求和成本考虑因素很重要。

另一个重要的考虑因素是了解数据驻留要求或*监管义务*是否适用于您的数据。例如, 是否有需要遵守的区域数据隐私法律?如果有,应选择能够支持合规性的数据库解决 方案。



有多种类型的数据库可供选择,其中许多是专用的数据库。不过,数据库类型通常分为两大类:关系数据库和非关系数据库。

*关系数据库系统*是大多数人最熟悉的数据库类型。传统示例包括 Microsoft SQL Server、Oracle 数据库和 MySQL。

非关系数据库是最近发展起来的,但已经有几十年的历史。这类数据库在现代计算领域发挥着至关重要的作用。示例包括 MongoDB、Cassandra 和 Redis。

关系数据库类型



优势:

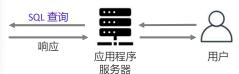
- 易用性
- 数据完整性
- 减少了数据存储
- 通用语言(结构化查询语言或 SQL)



关系数据库管理系统 (RDBMS)

在以下情况下,关系数据库是理想之选:

- 需要遵守严格的 schema 规则、ACID 合规性和数据质量强制执行规则。
- 不需要极大的读/写容量
- 不需要极高的性能
 - RDBMS 可能是最适合、最省事的解决方案



关系数据库有时称为*关系数据库管理系统* (RDBMS)。这类数据库仍然是世界上最受欢迎和最常用的数据库类别。根据 2019 年的 <u>DB-Engines 排名研究</u>,关系数据库在所有类别中人气指数占 75% 以上。排名前四位的分别是 Oracle、MySQL、Microsoft SQL Server 和 PostgreSQL,它们都是关系数据库。

关系数据库一直如此受欢迎,原因有很多。其中包括*易用性*、数据完整性控制、在降低整体数据存储方面表现出色,以及对结构化查询语言 (SQL) 的支持。SQL 是一种用于查询或与存储在 RDBMS 中的结构化数据交互的流行语言,受到大多数供应商的支持。

如果您的使用案例适合*严格的 schema 规则*,且 schema 结构定义明确,不需要经常改变,那么关系数据库是不错的选择。此外,关系数据库事务满足 *ACID* 合规性,也就是说,这类数据库通过提供遵循*原子性、一致性、隔离性、持久性*的事务来确保数据完整性。但是,如果您的应用程序需要极高的读/写容量,则关系数据库可能不是合适的选择。最后,将关系数据库与其他选项进行比较时,极高性能是关系数据库的一个特征。但是,对于许多使用案例来说,关系数据库可能是最省事的解决方案。

非关系数据库类型



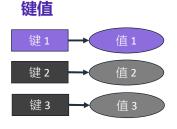
优势

- 灵活性
- 可扩展性
- 高性能
- 功能强大的 API

在以下情况下,非关系数据库是理想之选:

- 数据库必须横向扩展以便处理庞大的数据量
- 数据不适合传统 schema
- 传统 RDBMS 无法以经济高效的方式支持您的读/ 写速率

示例模型





非关系数据库有时称为 *NoSQL 数据库。*这类数据库不同于关系数据库,因为它们专用于特定的数据模型,并且具有灵活的 schema 来构建现代应用程序。非关系数据库因其*灵活性、可扩展性、高性能和功能强大的 API* 而广受赞誉。

这类数据库使用各种数据模型,其中包括键值、图形、文档、内存中和搜索。NoSQL schema 是动态的。例如,行不需要在每列中都包含数据。此外,NoSQL 数据库可以通过增加其运行的服务器数量来*横向*扩展。

非关系数据库的 schema 灵活,其中每个对象可以具有不同的结构。它们可以存储结构化数据 (如关系数据库记录)、半结构化数据 (如 JavaScript 对象表示法 (JSON) 文档),以及非结构化数据(如照片文件或电子邮件)。当数据可能存在结构上的不一致性时,非关系数据库可以很好地发挥作用。

最后,非关系数据库针对特定的数据模型和访问模式进行了优化,可实现*更高的性能*, 而不是试图实现关系数据库的功能。



您已经了解了关系数据库和非关系数据库之间的区别,现在可以思考一下有哪些 Amazon 数据库产品符合这两种类别。

较常用的 AWS 关系数据库产品包括 Amazon RDS、Amazon Redshift 和 Amazon Aurora.

类似地,较常用的*非关系*数据库包括 Amazon DynamoDB、Amazon ElastiCache 和 Amazon Neptune。

在本模块中,您将更深入地了解 Amazon RDS 和 Amazon DynamoDB。有关 AWS 上提供的数据库服务的完整列表及其最常见的使用案例,请参见 AWS 数据库 – 概览页面。





- 在选择数据库时,应考虑可扩展性、 存储需求、要存储的对象的类型和大小, 以及持久性要求
- 关系数据库具有严格的 schema 规则,可确保数据完整性,并支持 SQL
- 非关系数据库可横向扩展,提供更高的可扩展性和灵活性,并且很适合半结构化和非结构化数据

本模块中这节内容的要点包括:

- 在选择数据库时,应考虑可扩展性、存储需求、要存储的对象的类型和大小,以及持久性要求
- 关系数据库具有严格的 schema 规则,可确保数据完整性,并支持 SQL
- 非关系数据库可横向扩展,提供更高的可扩展性和灵活性,并且很适合半结构化和 非结构化数据

模块 5: 添加数据库层 第 3 节: Amazon RDS

介绍第3节: Amazon RDS。



Amazon RDS 是一项完全托管的关系数据库服务,可在云中创建和运行关系数据库。在了解更多关于 Amazon RDS 的详细信息之前,您将首先了解 Amazon RDS 作为*托管*数据库服务的优势。



现在,您将了解使用托管的 AWS 数据库服务的一些优势。

在左侧的用例中,您将数据库托管在*本地*数据中心。在这种情况下,一切事情都由您负责。您必须为物理服务器的机架供电,维护运行数据库的服务器的操作系统,并执行数据库软件的安装和打补丁工作。您还必须执行备份,配置具备高可用性和扩展性的解决方案,并优化使用数据库的应用程序。

在中间的用例中,您在一个或多个Amazon EC2 示例上安装数据库。在这种情况下,AWS 负责物理数据中心环境的维护,并且操作系统预安装在您启动的 EC2 实例上。但是,您仍然要负责操作系统安装之上的每一个配置层,这意味着您必须手动管理很多资源。

右侧的使用案例使用的是 AWS 托管数据库产品。这些解决方案提供高可用性、可扩展性和数据库备份作为您可以配置的内置选项。AWS 负责处理常见的、重复性的数据库管理任务。您只负责优化您的应用程序,并确保数据库层有效地适用于您的应用程序。

Amazon RDS 特性











访问模式 事务性轻量级分析

数据大小 低 TB 范围

性能 中高吞吐量、低延迟

业务使用案例 事务性 OLAP

Amazon Relational Database Service (Amazon RDS) 是一项完全托管的关系数据库服务。

Amazon RDS 通常用于是事务性访问模式的使用案例,或用于轻量级分析。

作为关系数据库,理想的数据大小范围可达到低 TB 范围。随着存储需求的增长,您可以预置额外的存储。Amazon Aurora 引擎会随着您数据库存储需求的增长而自动增加您的数据库容量大小。它最多可扩展至 64TB 或您定义的最大容量。MySQL、MariaDB、Oracle 和 PostgreSQL 引擎使您能够扩展到 32TB 的存储空间。Microsoft SQL Server 最高支持 16TB。存储扩展是动态的,无停机时间。

在性能方面,Amazon RDS 提供了两种选项。其中一个是通用固态硬盘 (SSD) 支持的存储选项。SSD 选项提供了一致的基准,即预置的每 GB 有 3 IOPS,并且可以突增至高出基准达 3000 IOPS。这种存储类型适用于广泛的数据库工作负载。另一个是预置 IOPS SSD 存储选项,非常适用于 I/O 密集型事务性数据库工作负载。

Amazon RDS: 用途和数据库类型





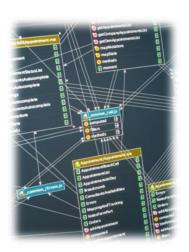
Amazon RDS

非常适合以下应用:

- 具有更复杂的数据
- 需要合并和联接数据集
- 需要强制实施语法规则

支持的六种数据库类型:

- Microsoft SQL Server
- Oracle
- MySQL
- PostgreSQL
- Aurora
- MariaDB



作为关系数据库产品,Amazon RDS 对于具有复杂、结构化良好数据的应用程序来说是不错的选择。如果您的工作负载必须经常合并和联接数据集,并且必须有严格执行的语法规则,那么 Amazon RDS 是不错的选择。例如,Amazon RDS 经常用于支持传统应用、企业资源规划 (ERP)、客户关系管理 (CRM) 和电子商务应用。

Amazon RDS 有六个数据库引擎可供选择,包括 Microsoft SQL Server、Oracle、MySQL、PostgreSQL、Amazon Aurora 和 MariaDB。

数据库实例大小调整 **aws** academy T系列 M 系列 R 系列 突增型 通用型 内存优化型 类型 实例 实例 实例 1 个 vCPU/1GB RAM 2 ↑ vCPU/8GB RAM 2 ↑ vCPU/16GB RAM 调整大小 至 96 个 vCPU/384GB RAM 至 96 个 vCPU/768GB RAM 至8个vCPU/32GB RAM 联网 中等性能 高性能 高性能 适合的工 查询密集型、 小型或可变 CPU 密集型 作负载 连接次数高 R5 提供多达 T3 可突增到基准以上, M5 提供多达 96 个 vCPU 亮点 96 个 vCPU、768GiB RAM 需额外付费

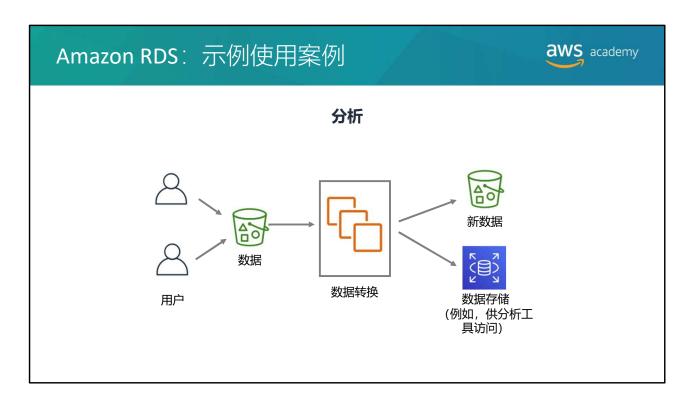
所有类型的 Amazon RDS 数据库都运行在服务上,但 Aurora 是个例外,它可以作为无服务器选项运行。Amazon RDS 可用于多种数据库实例类型,这些实例类型针对不同类型的工作负载进行了优化。

T 系列实例类型提供基准水平的 CPU 性能,并且只要需要,随时能够突增 CPU 使用率。例如,T3 实例可均衡计算、内存和网络资源,非常适用于 CPU 使用率中等但使用量可能临时激增的数据库工作负载。

M 系列实例是另一种通用选项,但 M 系列为 CPU 密集型工作负载提供了更多的选择。 对于开源或企业应用中的中小型数据库,M 实例是不错的选择。

最后,R 系列实例针对内存密集型数据库工作负载进行了优化。

部分数据库引擎支持其他的数据库实例类。有关更多信息,请参阅 Amazon RDS 数据库实例,特别是参阅 AWS 文档中的选择数据库实例类页面,了解实例类的详细信息。



试想这样一个示例使用案例,其中 Amazon RDS 用作部署的业务挑战解决方案的一部分。

在本例中,一家公司希望从他们的一些业务数据中获得有意义的见解。用户将数据上传到 Amazon Simple Storage Service (Amazon S3) 存储桶。然后,使用 Amazon EC2 实例集群来转换这些数据。这可能以批处理的形式定期进行。转换后的数据将存储在另一个 S3 存储桶中。但是,部分转换后的数据会被插入到一个 RDS 实例中,该实例标记为数据存储。之后可以使用分析软件或 AWS 分析服务查询此 RDS 实例,这些分析软件或服务可通过 SQL 查询来访问数据,获得业务见解。



您现在将完成模块 5 – 指导实验: 创建 Amazon RDS 数据库。



在本指导实验中, 您将完成以下任务:

- 1. 创建 Amazon RDS 数据库
- 2. 配置 Web 应用程序与数据库实例的通信



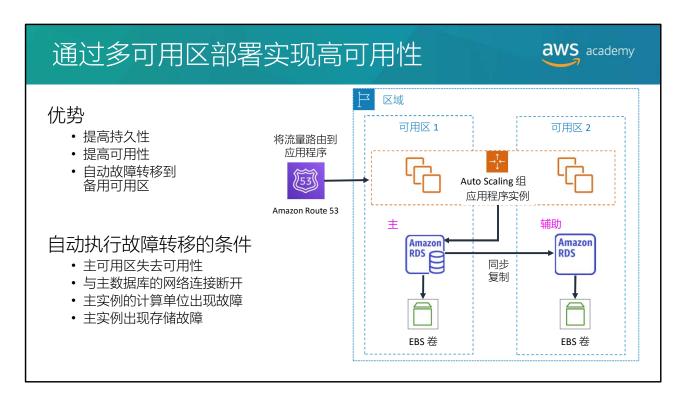
现在可以开始指导实验了。



指导实验总结: 要点



完成这个指导实验之后,您的讲师可能会带您讨论此指导实验的要点。



Amazon RDS 通过实施多可用区方法提供高可用性。Amazon RDS 会自动在不同可用区中预置和维护同步备用实例。主数据库实例将跨可用区同步复制到备用实例,以提供数据冗余、消除 I/O 冻结,并在系统备份期间将延迟峰值降至最小。

运行具有高可用性的数据库实例可提高计划内系统维护期间的可用性。Amazon RDS 可帮助保护数据库免受数据库实例故障和可用区中断的影响。它还能提高数据库实例的可用性和持久性,使其成为生产数据库工作负载的不错选择。

如果基础设施出现故障,Amazon RDS 会将故障自动转移到备用实例。故障转移完成后,您可以恢复数据库操作。故障转移后数据库实例的终端节点维持不变,因此您的应用程序可在没有手动管理干预的情况下恢复数据库操作。

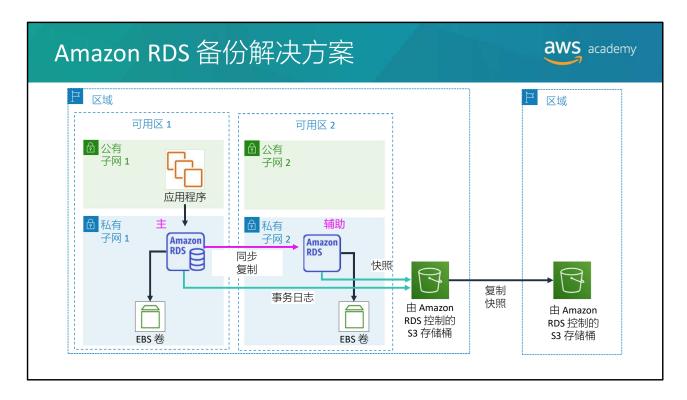
故障转移条件包括主可用区失去可用性、与主数据库的网络连接断开、主实例上的计算单位故障或存储故障。

使用只读副本提高性能 **aws** academy 区域 优势 可用区1 • 提高性能 • 提高可用性 公有子网 • 专为安全性设计 受以下数据库支持 MySQL 应用程序 MariaDB PostgreSQL 读取/ 私有子网 写入 • Oracle 限制 Ama. RDS 只读 Amazon • 每个数据库最多可以有五个只读副本 读取 副本 • 为了实现严格的先写后读一致性, 需从主数据 库实例读取 异步复制

Amazon RDS 使您能够使用源数据库实例创建一种称为*只读副本*的特殊类型的数据库实例。对源数据库实例的更新将异步复制到只读副本。

使用只读副本可以提高*性能。*例如,您可以通过将只读查询从应用程序路由到只读副本来减少源数据库实例的负载。只读副本还可以提高*可用性。*对于读取量较大的数据库工作负载,您可以进行弹性扩展,超越单个数据库实例的容量限制。如有需要,您也可以手动将只读副本提升为独立的数据库实例。

Amazon RDS for MySQL、MariaDB、PostgreSQL 和 Oracle 均支持只读副本。这些数据库引擎中的每一个支持每个主数据库最多有五个只读副本。如果您需要严格的先写后读一致性(也就是说,*您读取的内容始终是刚写入的内容*,即使是立即读取),则应从主数据库实例中读取。否则,您可以分配负载,并从只读副本中读取。



现在,试想这样一个基于 Amazon RDS 的备份解决方案的示例架构。

在本例中,RDS 数据库的快照和事务日志存储在由 Amazon RDS 控制的 S3 存储桶中。借助 Amazon RDS,您还可以将其指定为将自动或手动数据库快照复制到另一个 AWS 区域。您也可以将快照复制到单独的 AWS 账户。

有关更多信息,请参阅 AWS 文档中的复制快照。



现在,讲师可以选择使用 AWS 管理控制台来演示在 Amazon RDS 上配置自动备份和只读副本。

Amazon Aurora





Amazon Aurora 是完全托管且与 MySQL 和 PostgreSQL 兼容的关系数据库引擎。

- 用于联机事务处理 (OLTP)
- 最多可将 MvSQL 吞吐量增加至五倍*
- 最多可将 PostgreSQL 吞吐量增加至三倍*
- 跨三个可用区以六条路径复制数据
- 只需对您现有的应用程序进行少量更改

*以下数据库提供详细的基准测试信息: MySQL 和PostgreSQL。

Amazon Aurora 是 Amazon RDS 的一种数据库引擎选项。

Aurora 是专为云构建与 MySQL 和 PostgreSQL 兼容的关系数据库。它既具有传统企业级数据库的性能和可用性,又具有开源数据库的简单性和成本效益。

使用 Aurora 的速度最高可达标准 MySQL 数据库的五倍,标准 PostgreSQL 数据库的三倍。 Aurora 可以实现商用数据库的安全性、可用性和可靠性,而成本只有商用数据库的 1/10。有关此基准的详细说明以及如何自行复制此基准,请参阅 Amazon Aurora MySQL 性能基准指南和 Amazon Aurora PostgreSQL 性能基准指南。

Aurora 采用一种具有容错能力并能自我修复的分布式存储系统,这一系统可以把每个数据库实例自动扩展到最高 64TB。它具备高性能和高可用性,支持最多 15 个低延迟只读副本、时间点恢复、持续备份到 Amazon S3,还支持跨三个可用区复制。

Aurora 常用于*联机事务处理* (OLTP)。OLTP 系统必须能够处理大量并发用户的操作,并能够运行插入和更新请求。一个常见的 OLTP 示例是订单录入系统。OLTP 通常与*联机分析处理* (OLAP) 形成对比,后者的特点是更复杂的查询量较少。

Amazon Redshift





Amazon Redshift 是一项数据仓库服务。

- •用于联机分析处理 (OLAP)
- 存储非常大的数据集
 - 在 Amazon Redshift 中存储高度结构化、频繁访问的数据
 - 此外,可以在 Amazon S3 中存储 EB 级结构化、半结构化和非结构化数据。

Amazon Redshift 是另一款 AWS 关系数据库产品,它不在 Amazon RDS 上运行。

Amazon Redshift 提供 PB 级数据仓库和数据湖分析,Amazon Redshift 通常用于存储频繁访问、高度结构化的数据。Amazon Redshift 还可以直接访问 Amazon S3 中的数据,因此您可以在 Amazon S3 中维护 EB 级的结构化、半结构化和非结构化数据。

即使有数千个并发查询,Amazon Redshift 也能提供始终如一的快速性能。无论您要查询 Amazon Redshift 数据仓库中的数据,还是查询 Amazon S3 数据湖中的数据,都可获得一致的性能。您可以查询 Parquet、JSON、Avro、CSV 等开放文件格式,也可以直接使用 SQL 查询 Amazon S3。





- 托管的 AWS 数据库服务负责处理管理任务,让您可以专注于自己的应用程序
- Amazon RDS 支持 Microsoft SQL Server、Oracle、 MySQL、PostgreSQL、Aurora 和 MariaDB
- Amazon RDS 多可用区部署通过自动故障转移提供 高可用性
- 每个主数据库最多可以有五个只读副本,从而提高 Amazon RDS 的性能
- Amazon Aurora 是一种完全托管且与 MySQL 和 PostgreSQL 兼容的关系数据库引擎
- Amazon Redshift 是一款数据仓库关系数据库产品

本模块中这节内容的要点包括:

- 托管的 AWS 数据库服务负责处理管理任务, 让您可以专注于自己的应用程序
- Amazon RDS 支持 Microsoft SQL Server、Oracle、MySQL、PostgreSQL、Aurora 和 MariaDB
- · Amazon RDS 多可用区部署通过自动故障转移提供高可用性
- 每个主数据库最多可以有五个只读副本,从而提高性能
- Amazon Aurora 是一种完全托管且与 MySQL 和 PostgreSQL 兼容的关系数据库引擎
- Amazon Redshift 是一款数据仓库关系数据库产品

模块 5: 添加数据库层 第 4 节: Amazon DynamoDB

介绍第4节: Amazon DynamoDB。

Amazon DynamoDB





一项完全托管的非关系键值和文档数据库服务。







儿似力的



适合企业使用

极强的横向扩 展能力 事件驱动型编程 (无服务器计算) 加密、访问 控制、备份

Amazon DynamoDB 是一项完全托管的非关系 NoSQL 数据库服务。它凭借无缝可扩展性提供快速可预测的性能。

主要优势如下:

- 大规模性能
 - 提供只有几毫秒的一致响应时间
 - 您可以构建吞吐量几乎无限的应用程序
- 无服务器
 - 无需预置、修补或管理服务器
 - 无需安装、维护或操作软件
- 适合企业使用
 - 支持 ACID 事务
 - 默认情况下静态加密所有数据
 - 提供多区域复制 (全局表)
 - 提供精细的身份与访问控制
 - 您可以执行完整的数据备份,而不会影响性能

Amazon DynamoDB 特性





非常适合以下应用:

- 具有简单的大容量数据 (高 TB 范围)
- 必须快速扩展
- 不需要复杂的联接
- 需要超高吞吐量和低延迟

主要功能

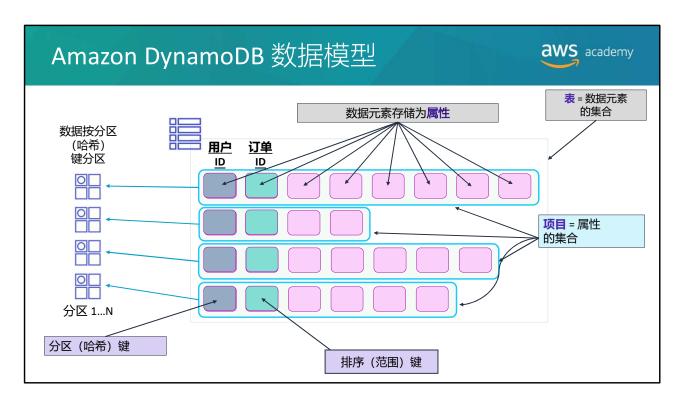
- NoSQL表
- 项目可以拥有不同属性
- 内存中的缓存
- 支持每秒超过 2000 万个请求的峰值



DynamoDB 非常适合要处理大量数据且必须快速扩展的应用程序。它还提供超高的吞吐量和较低的延迟,对于游戏、广告技术、移动应用程序以及有这些要求的其他应用程序来说,是一个不错的选择。但是,如果您的工作负载需要复杂的联接,关系数据库可能是更好的选择。

DynamoDB 表没有固定的 schema,每一个项目可以拥有不同数量的属性。

由于 DynamoDB 提供内存中的缓存,每秒可以处理超过 2000 万个请求,因此经常用于必须维持会话状态的应用程序、无服务器 Web 应用程序和微服务。



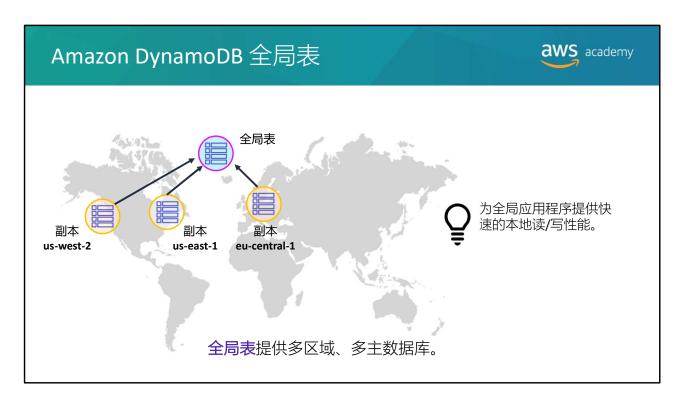
现在来看一下 Amazon DynamoDB 数据模型。由于 DynamoDB 是 NoSQL 数据库,因此与关系数据库不同,它不强制执行严格的 schema。

DynamoDB 表包含项目。一个项目的属性数量不固定。每个属性由一个键值对组成。

DynamoDB 支持使用用户定义的主键(分区键,也称为哈希键)来执行键值 GET/PUT 操作。主键是表中项目必须具备的唯一属性,可以通过它唯一地标识每个项目。您在创建表时指定主键。此外,DynamoDB 允许使用全局二级索引和本地二级索引对非主键属性进行查询,从而提供灵活的查询。

当数据集的大小增加以及随着预置的容量增加,会发生自动分区。

考虑这样一个例子,您创建了一个 Amazon DynamoDB 表来保存订单信息。在本例中,您的表条目将始终具有用户ID和订单ID。但是,您只能收集部分订单的其他详细信息(如用户的电子邮件地址),不能收集另外一些订单的这些信息。随着更多的项目添加到表中,您可以继续收集新的属性。这些属性可以包括您在开始创建表时没有预期到的属性。由于具有这种灵活性,DynamoDB 非常适合快速变化的非结构化、半结构化和结构化数据。

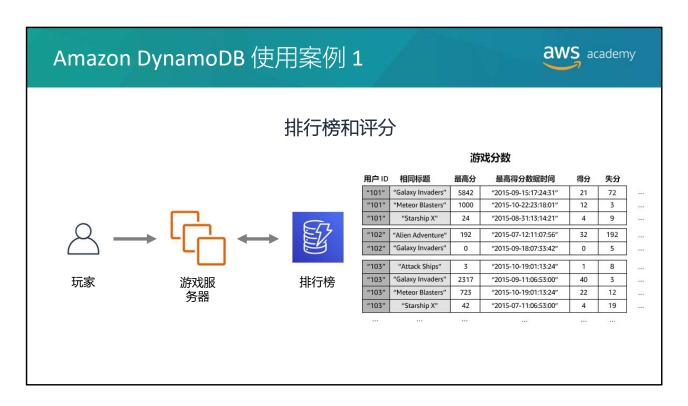


默认情况下,Amazon DynamoDB 会在一个区域中跨多个可用区复制您的数据。但在某些情况下,您可能希望跨多个区域复制数据。

Amazon DynamoDB 全局表为部署多区域、多主数据库提供了完全托管的解决方案。您不需要构建和维护自己的复制解决方案。在创建全局表时,指定要在其中提供表的 AWS 区域。DynamoDB 执行所有必要的任务,以便在这些区域中创建相同的表。然后,DynamoDB 会将持续的数据更改传播到全部的表。

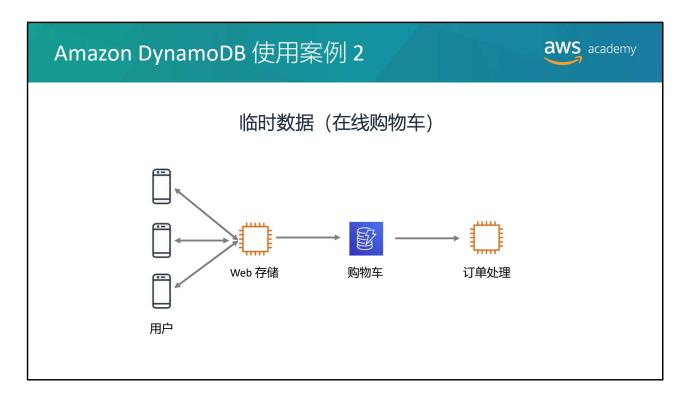
为说明全局表的一个使用案例,假设您有一个大型客户群跨越三个地理区域:美国东海岸、美国西海岸和西欧。客户在使用您的应用程序时必须更新其个人资料信息。

为了满足这一业务要求,您可以创建由三个特定于区域的 CustomerProfiles 表构成的全局表。然后,DynamoDB 会自动在这些表之间复制数据更改。在一个区域中对 CustomerProfiles 数据所做的更改将无缝传播到其他区域。此外,如果其中一个 AWS 区域暂时不可用,您的客户仍然可以访问其他区域中的相同 CustomerProfiles 数据。



游戏公司将 Amazon DynamoDB 应用于游戏平台的各个部分,包括游戏状态、玩家数据、会话历史记录和排行榜。您可以可靠地扩展到数百万并发用户和请求,同时确保始终保持低延迟。

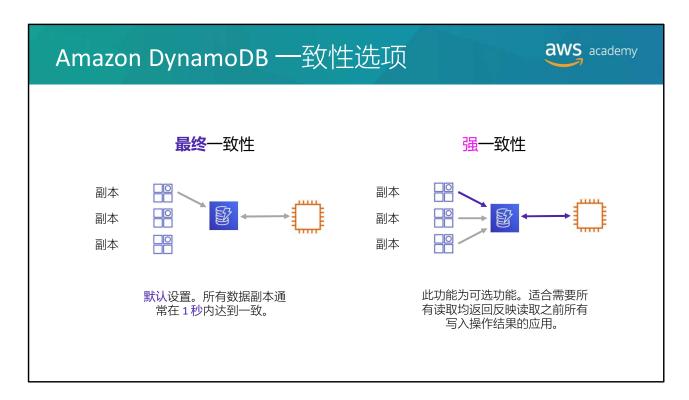
在此示例使用案例中,在 Amazon DynamoDB 中维护一个游戏得分排行榜。主动连接到游戏服务器的玩家可能会通过游戏用户界面 (UI) 看到其中的一些信息。当新的高分出现时,GameScores 排行榜会不断更新,每个高分玩家的胜负记录也会发生变化。以Electronic Arts (EA) 为例,这是一家大型视频游戏公司,在全球拥有超过 3 亿注册玩家。对于 EA 来说,高并发意味着每秒 10 万次以上的请求和数百万日活跃用户。有关详细信息,请参阅 DynamoDB 游戏使用案例。



之所以开发 Amazon DynamoDB,是因为客户需要一个可扩展且高度可靠的键值数据库来支持 Amazon 电子商务的基本操作,如购物车。

2012 年,Amazon 首席技术官 Werner Vogels 在宣布发布 Amazon DynamoDB 时,在一篇 博文中说明了 Amazon 购物车的业务要求。"这种非关系或 NoSQL 数据库针对购物车等使用案例打造,任何停机时间或性能下降…都会对收益产生直接影响,它们对数据系统的容错能力和性能要求非常严格。这些服务还需要能够逐步扩展基础设施,以适应请求速率或数据集大小的增长。另一个重要的要求…是可预测性。"

DynamoDB 功能集不断发展。但是,激励开发这项服务的核心目标如今仍然适用于希望实施网店购物车订单处理解决方案的企业或组织。



*读取一致性*表示的是某个数据项目的成功写入或更新*如何*以及*何时*反映在该数据项的后续读取操作中。Amazon DynamoDB 提供的逻辑使您能够针对应用程序中的每个读取请求,指定您需要的一致性特征。

最终一致性读取

当您从 DynamoDB 表读取数据时,响应反映的可能不是刚刚完成的写入操作的结果。响应可能包含某些陈旧数据。如果您在短时间后重复读取请求,响应将返回最新的数据。

强一致性读取

当您请求强一致性读取时, DynamoDB 会返回具有最新数据的响应。这些响应反映了所有之前成功写入操作的更新。如果网络延迟或中断,可能会无法执行强一致性读取。

除非您指定其他读取方式,否则 DynamoDB 将使用*最终一致性读取*。读取操作(如 GetItem、Query 和 Scan)提供了一个 ConsistentRead 参数。如果您将此参数设置为 *true* , DynamoDB 将在操作过程中使用强一致性读取。



现在您已经了解了 Amazon RDS 和 Amazon DynamoDB 的基本关键功能,请思考在这两种数据库服务中,哪一种更适合咖啡馆的使用案例。

在本课程前面的模块中,您完成了一项挑战实验,您在其中部署了在 EC2 实例上运行的 MySQL 数据库。该数据库托管一个表,其中包含有关咖啡馆菜单的详细信息。它还提供数据存储层,用来存储客户在线下的订单。

如果咖啡馆团队决定他们想迁移数据库层,以便改为使用 Amazon 托管的数据库服务,那么这两个数据库选项中的哪一个更合适?

讲师将带班级学员讨论这个主题。我们鼓励您参与讨论,并解释您给出的答案的原因。





- Amazon DynamoDB 是一项完全托管的非 关系键值和文档 NoSQL 数据库服务。
- DynamoDB 是无服务器服务,可提供极高的横向扩展和低延迟性能。
- DynamoDB 全局表可确保数据复制到多个区域。
- DynamoDB 默认提供最终一致性(一般情况下,在写入后的 1 秒钟内的读取完全一致)。强一致性是另一种选项。

本模块中这节内容的要点包括:

- Amazon DynamoDB 是一项完全托管的非关系键值和文档 NoSQL 数据库服务。
- DynamoDB 是无服务器服务,可提供极高的横向扩展和低延迟性能。
- DynamoDB 全局表可确保数据复制到多个区域。
- DynamoDB 默认提供最终一致性(一般情况下,在写入后的 1 秒钟内的读取完全一致)。强一致性是另一种选项。

模块 5:添加数据库层第 5 节:数据库安全控制



介绍第5节:数据库安全控制。

保护 Amazon RDS 数据库



建议

- 在 Virtual Private Cloud (VPC) 中运行 RDS 实例
 - 提供服务隔离和 IP 防火墙保护
- 使用 AWS Identity and Access Management (IAM) 策略进行身份验证和访问
 - 权限决定谁可以管理 Amazon RDS 资源
- 使用安全组来控制哪些 IP 地址或 EC2 实例可以连接到您的数据库
 - 默认禁用网络访问
- 使用安全套接字层 (SSL) 加密传输中的数据
- 在数据库实例和快照上使用 Amazon RDS 加密,以保护静态数据
- 使用数据库引擎的安全功能控制可以登录数据库实例上的数据库的人员
- 配置事件通知,以便在发生重要的 Amazon RDS 事件时提醒您







安全性是您和 AWS 的共同责任。AWS 负责云的安全性,这意味着,AWS 保护运行 Amazon RDS 的基础设施,而客户负责云中的安全性。

Amazon RDS 的一项安全建议是在 *Virtual Private Cloud (VPC) 中运行您的 RDS 实例。*VPC 使您能够将实例放置在私有子网中,确保其不受互联网上的公有路由影响。VPC 还提供 IP 防火墙保护,使您能够安全地控制适用的网络配置。

此外, 我们建议您:

- 使用 AWS Identity and Access Management (IAM) 策略进行身份验证和控制访问。
- 配置*安全组*来限制连接。打开可访问数据库的 TCP 端口。但是,您可以限制这些连接的来源。
- 使用SSL 连接确保与数据库之间的所有通信都受到保护。
- 使用Amazon RDS 加密来加密您的数据和数据库快照。
- 使用数据库引擎的安全功能,实现强制实施密码复杂性等要求。

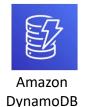
您可以*启用事件通知*,以便在 RDS 实例上发生重要事件时接收提醒。这些事件可能包括实例是否已关闭、备份是否启动、是否发生故障转移、安全组是否发生更改或者存储空间是否不足等。

保护 Amazon DynamoDB



建议

- 使用 IAM 角色对访问进行身份验证
- 使用 IAM 策略
 - 定义使用 DynamoDB API 的精细访问权限
 - 定义表、项目或属性级别的访问权限
 - 遵循授予最小权限的原则
- · 配置 VPC 终端节点
 - 防止连接流量遍历开放的互联网
 - VPC 终端节点策略让您能够控制和限制对 DynamoDB 表的 API 访问
- 考虑使用客户端加密
 - 在尽可能靠近数据源的位置加密数据



默认情况下提供的安全功能

- 对存储在表、索引、流和备份中的所有用户 数据进行**静态加密**
- 传输中加密 所有与 DynamoDB 和其他 AWS 资源之间的通信都使用 HTTPS

为了保护 Amazon DynamoDB 的安全,您应该用来保护 Amazon RDS 的许多相同的最佳实践也适用。

例如,使用 IAM **角色**来确保安全的身份验证,以及**使用 IAM 策略**来定义访问权限。

如果您只需要访问从 Virtual Private Cloud (VPC) 中的 DynamoDB,则应使用 **VPC 终端节点**来限制为仅从所需的 VPC 访问。这样做可以防止流量遍历开放的互联网,受到该环境的影响。

此外,如果您将敏感或机密数据存储在 DynamoDB 中,您可能希望在尽可能靠近数据源的位置加密数据,以便在该数据的整个生命周期内对其进行保护。

请注意,默认情况下,DynamoDB 提供特定的安全功能。例如,DynamoDB 可保护静态存储的用户数据、本地客户端与 DynamoDB 之间传输的数据,以及 DynamoDB 与同一AWS 区域中的其他 AWS 资源之间传输的数据。

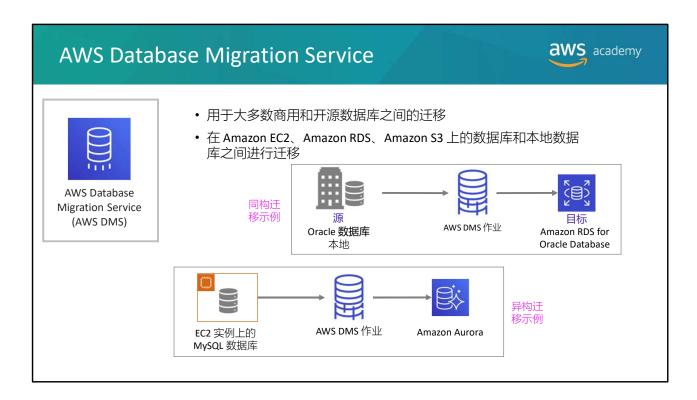
有关更多详细信息,请阅读 DynamoDB 安全性最佳实践文档。

模块 5: 添加数据库层

第6节:将数据迁移到 AWS 数据库



介绍第6节:将数据迁移到AWS数据库。



您可以使用 AWS Database Migration Service (AWS DMS) 将现有的数据库迁移或复制到 Amazon RDS。AWS DMS 支持在最广泛使用的数据库之间迁移。

支持的*源数据库*包括 Oracle、Microsoft SQL Server、MySQL、MariaDB、PostgreSQL、IBM Db2 LUW、SAP、MongoDB 和 Amazon Aurora。*目标数据库*引擎包括 Oracle、Microsoft SQL Server、PostgreSQL、MySQL、Amazon Redshift、SAP ASE、Amazon S3 和 Amazon DynamoDB。

AWS DMS 还支持同构(相同引擎)迁移和异构(不同引擎)迁移。

所示的第一个示例是同构转换,将本地 Oracle 数据库迁移到 Amazon RDS for Oracle Database。

第二个示例展示了异构迁移,将在 Amazon EC2 实例上运行的 MySQL 数据库迁移到 Amazon Aurora。

AWS DMS 的主要功能



- 执行一次性迁移
- 或完成连续数据复制
 - 示例: 配置从本地数据库到 RDS 实例的连续数据复制
- AWS Schema Conversion Tool (AWS SCT) 支持 在源和目标之间更改数据库引擎
- 典型的迁移主要步骤:
 - 1. 创建目标数据库
 - 2. 迁移数据库 schema
 - 3. 设置数据复制过程
 - 4. 启动数据传输,并确认完成
 - 5. 将生产切换到新数据库 (用于一次性迁移)



AWS DMS 可用于执行一次性迁移,也可用于完成两个数据库之间的连续数据复制。例如,您可以使用它来配置从本地数据库到 RDS 实例的连续数据复制。

当您想要执行从一个数据库引擎到另一个数据库引擎的异构迁移时,可能需要使用 AWS Schema Conversion Tool (AWS SCT)。

典型的数据库迁移涉及以下主要步骤:

- 1. 创建目标数据库
- 2. 迁移数据库 schema
- 3. 设置数据复制过程
- 4. 启动数据传输,并确认完成
- 5. 将生产切换到新数据库(用于一次性迁移)

使用 AWS Snowball Edge 和 AWS DMS

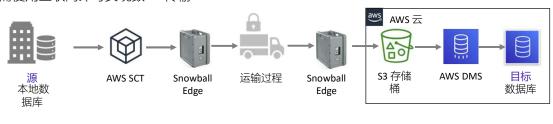


迁移数据不可行的情况:

- 数据库太大
- 连接太慢
- 您有隐私和安全方面的考虑

使用 AWS Snowball Edge

• 无需使用互联网即可实现数 TB 传输



较大的数据迁移可能包含许多 TB 的信息。由于网络带宽的限制或数据量的限制,这一过程会很困难。AWS Database Migration Service (AWS DMS) 可以使用 AWS Snowball Edge和 Amazon S3,比其他方法更快地迁移大型数据库。

AWS Snowball 是一项 AWS 服务,此服务提供的 Edge 设备可让您以比网络更快的速度将数据传输到云。Edge 设备是 AWS 拥有的设备,它提供了大量的内置存储空间。此外,Edge 设备使用 256 位加密和开放标准的受信任平台模块 (TPM),可以确保您的数据安全并形成完整的监管链。

在您使用 Edge 设备时,数据迁移过程包括以下阶段:

- 使用 AWS SCT 本地提取数据并将数据移至 Edge 设备
- 将设备寄回给 AWS
- 在 AWS 收到您的货件后,该设备会自动将其数据加载到 S3 存储桶中
- AWS DMS 提取相关文件并将数据迁移到目标数据存储中



您现在将完成模块 5 – 挑战实验:将数据库迁移到 Amazon RDS。

业务需求:托管数据库



对 Sofía 和 Nikhil 来说,在 EC2 实例上运行的数据库 变得难以维护。





最近 Olivia 访问咖啡馆时,向他们谈到了 Amazon RDS 的功能。

Sofía 和 Nikhil 决定将咖啡馆的数据库迁移到 Amazon RDS。

对 Sofía 和 Nikhil 来说,咖啡馆数据库的管理变得很困难。例如,他们必须在咖啡馆停止营业后加班加点进行每周的数据库备份。

另外,他们最近正在努力安装所需的补丁。Sofía 和 Nikhil 在周末下班后的时间几乎没有完成这项工作。Frank 和 Martha 现在意识到,这些维护工作增加了企业的人力成本,因为他们必须为 Sofía 和 Nikhil 的加班时间支付薪水。

AWS 解决方案架构师 Raquel 是咖啡馆的顾客,也是 Sofía 的朋友。Sofía 在最近的一次 谈话中向 Olivia 提到了数据库的主题。Olivia 建议他们将数据库迁移到 Amazon RDS。

该解决方案将减轻手动执行常见数据库维护任务的负担,比如备份、补丁安装和升级。 作为一项完全托管的服务,Amazon RDS 会自动执行这些任务。

在本活动中,您将扮演 Nikhil 和 Sofía,负责将咖啡馆数据库迁移到 Amazon RDS。

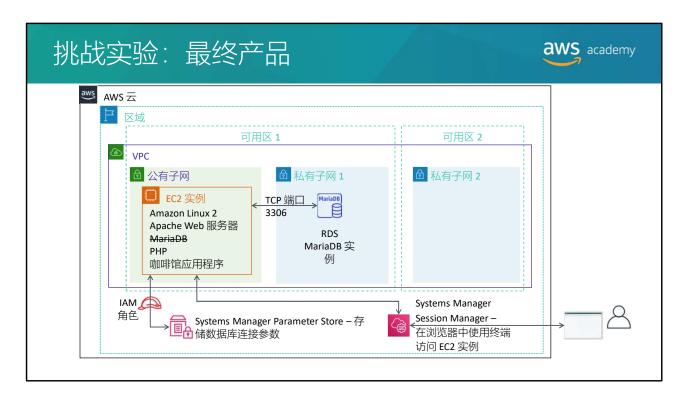
挑战实验: 任务



- 1. 创建 RDS 实例
- 2. 分析现有咖啡馆应用程序部署
- 3. 使用 EC2 实例中的数据库
- 4. 使用 RDS 数据库
- 5. 将数据导入 RDS 数据库实例
- 6. 将咖啡馆应用程序连接到新数据库

在本挑战实验中, 您将完成以下任务:

- 1. 创建 RDS 实例
- 2. 分析现有咖啡馆应用程序部署
- 3. 使用 EC2 实例中的数据库
- 4. 使用 RDS 数据库
- 5. 将数据导入 RDS 数据库实例
- 6. 将咖啡馆应用程序连接到新数据库



该图总结了您完成实验后将会构建的内容。

该咖啡馆的 Web 应用程序原先使用的是安装在 EC2 实例上的 MariaDB。在实验过程中,您将创建一个 RDS MariaDB 实例,将 EC2 实例上的数据从数据库中转储出来,并将这些数据迁移到 RDS 数据库中。

为了完成迁移,您可以使用 AWS Systems Manager Session Manager (通过 Web 浏览器提供终端访问)连接到 EC2 实例。在迁移过程中,您可以根据需要使用安装在 EC2 实例上的 MySQL 客户端连接到两个数据库。数据库凭证和数据库连接信息存储在 AWS Systems Manager Parameter Store 中。因此,在 EC2 实例中附加了一个 IAM 角色,以允许 Web 应用程序从参数仓库中读取数据。



现在可以开始挑战实验了。



挑战实验总结: 要点



完成这个挑战实验之后,您的讲师现在可能会带您讨论此挑战实验的要点。

模块 5: 添加数据库层 模块总结

现在来回顾下本模块,并对知识测验和对实践认证考试问题的讨论进行总结。

模块总结



总体来说,您在本模块中学习了如何:

- 比较数据库类型
- 区分托管服务和非托管服务
- 说明何时使用 Amazon Relational Database Service (Amazon RDS)
- 说明何时使用 Amazon DynamoDB
- 描述可用的数据库安全控制
- 描述如何将数据迁移到 Amazon Web Services (AWS) 数据库
- 部署数据库服务器

总体来说, 您在本模块中学习了如何:

- 比较数据库类型
- 区分托管服务和非托管服务
- 说明何时使用 Amazon Relational Database Service (Amazon RDS)
- 说明何时使用 Amazon DynamoDB
- 描述可用的数据库安全控制
- 描述如何将数据迁移到 Amazon Web Services (AWS) 数据库
- 部署数据库服务器



现在可以完成本模块的知识测验。

样题



一个应用程序需要一个高度可用的关系数据库,初始存储容量为 8TB。数据库将每天增长 8GB。为了支持预期的流量,至少需要八个只读副本来处理数据库读取。哪些选项将满足这些要求?

- A. DynamoDB
- B. Amazon S3
- C. Amazon Aurora
- D. Amazon Redshift

请查看答案选项,并根据之前突出显示的关键字排除错误选项。

正确答案是 C。Amazon Aurora 是一个关系数据库,可自动扩展以适应数据增长。 Amazon Redshift 不支持只读副本,也不会自动扩展。DynamoDB 是 NoSQL 服务,而非 关系数据库。Amazon S3 是对象存储,而非关系数据库。

其他资源



- AWS 数据库 资源页面
- Amazon RDS 入门指南
- Amazon RDS 最佳实践
- Amazon RDS 常见问题
- Amazon DynamoDB 开发人员指南
- Amazon DynamoDB 常见问题

如果您想了解有关本模块所涵盖主题的更多信息,下面这些其他资源可能会有所帮助:

- AWS 数据库 资源页面
- Amazon RDS 入门指南
- Amazon RDS 最佳实践
- Amazon RDS 常见问题
- Amazon DynamoDB 开发人员指南
- Amazon DynamoDB 常见问题



感谢您完成本模块的学习。