



RDS

O Relational Database Service é o serviço de banco de dados da AWS, é um serviço bem eficiente que possui integração com diversos SGBD, como:

- SQL Server
- Oracle
- MySQL
- PostgreSQL
- Aurora (da amazon)
- MariaDB

Features

O RDS possui dois pontos:

▼ Multi-AZ (Para recuperação de desastres)

- Onde existe uma versão primária e uma secundária do seu banco, suponhamos que a versão primária vá pro saco, daí o DNS (connection string) irá trocar o apontamento para versão secundária automaticamente.

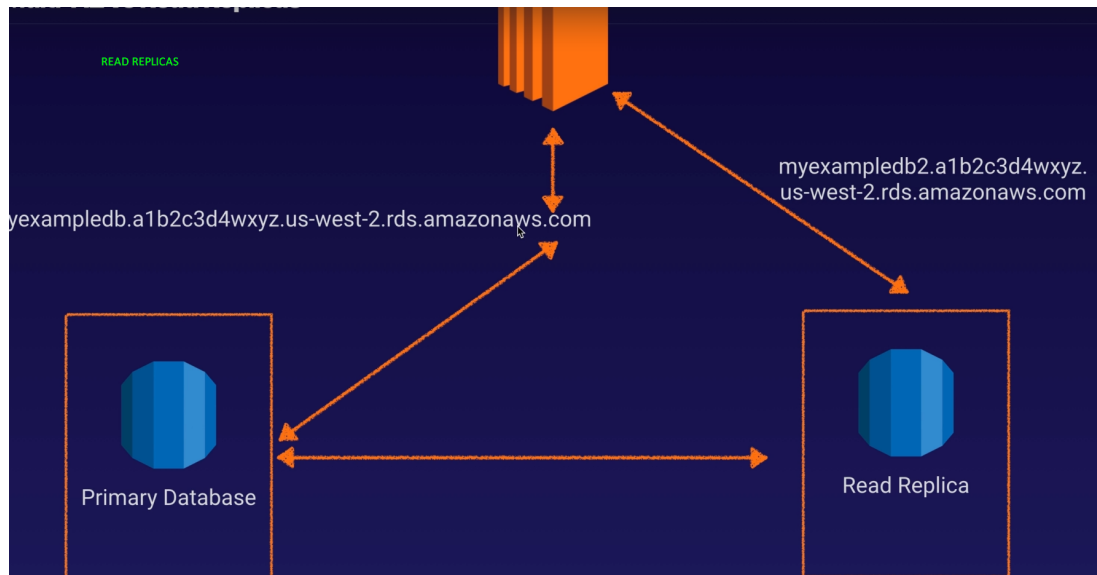


▼ Read Replica

- No modo de Read Replica, toda vez que você fizer uma ação de Write no seu banco principal, essa ação também será realizada no banco secundário, que fica em outra AZ. O DNS no entanto não trocará apontamento sozinho, como é no multi-AZ. Você precisará alterar tudo manualmente.

Com essa descrição, você deve ter pensado "Poxa, multi-az é muito melhor então", mas depende do contexto, a Read Replica é bem importante no caso de fazer um balanceamento de carga, para distribuir as requisições enviadas ao banco. Imagine por exemplo um servidor web de wordpress onde muitas pessoas leem todos os dias, nesse caso, distribuir a carga com o read replica seria excelente.

OBS: PODEMOS TER ATÉ 5 READ REPLICAS



Multi-AZ foca mais na da redundância, Read Replica foca no desempenho.

Processo Transacional x Analítico

OLTP (Online Transaction Processing):

- **Função:** OLTP é usado para gerenciar transações diárias e operacionais de uma organização. Ele é otimizado para processar um grande número de pequenas transações, como inserir, atualizar ou deletar dados.
- **Exemplo:** Sistemas de bancos, e-commerce e gerenciamento de inventário, onde as operações precisam ser rápidas e eficientes.
- **Características:** Alta velocidade, integridade dos dados, e suporte a muitas transações simultâneas.

OLAP (Online Analytical Processing):

- **Função:** OLAP é usado para realizar análises complexas e consultas sobre grandes volumes de dados históricos. Ele é otimizado para consultas que envolvem a agregação e análise de dados, como relatórios, dashboards e análise de tendências.
- **Exemplo:** Sistemas de Business Intelligence (BI) e Data Warehouses, que analisam dados de vendas ao longo do tempo para entender padrões e tomar decisões estratégicas.

- **Características:** Foco em leitura e consulta de dados, com menos ênfase em transações frequentes.

Backups

Além do Multi-AZ e Read Replica, temos outras opções de backup também, são elas:

▼ Automated Backups

Permite recuperar o banco de dados para qualquer momento em um “período de retenção”. O período de retenção pode ser entre 1 e 35 dias. Os Automated Backups fazem um snapshot completo diariamente, e também irão armazenar logs transacionais ao longo do dia. Quando você realizar um recovery, a AWS irá restaurar o snapshot mais recente e também aplicar os logs de inserção daquele dia.

▼ Database Snapshots

São feitos manualmente e ficarão armazenados mesmo depois da deleção da instância de RDS, diferentemente da versão automatizada

OBS: Independente do tipo de backup, um novo banco de dados será criado a partir do qual você fez o backup, portanto se trata de um novo endpoint, que você terá de corrigir manualmente

Elasticache

O Elasticache é o serviço que fornece a opção de armazenar em cache as queries mais comuns e mais requisitadas, dessa forma salvando o tempo de processamento para uma consulta real em um banco de dados em disco. Neste serviço temos duas opções de sistema: **Redis** e **Memcached**.

DynamoDB

O DynamoDB é um serviço de banco de dados NoSQL projetado para oferecer um desempenho rápido e rapidamente escalável. Ele é o ideal para aplicações

que demandam latência mínima.

Modelo de Dados:

- **Orientado a Tabelas:** O DynamoDB armazena dados em tabelas, com cada tabela contendo itens (registros). Cada item pode ter atributos (colunas), e o Schema da tabela é flexível, permitindo diferentes itens na mesma tabela com diferentes atributos.
- **Chave Primária:** Para identificar itens únicos, o DynamoDB usa uma chave primária, que pode ser uma chave simples (partition key) ou composta (partition key + sort key).

Performance Escalável:

- **Leitura e Escrita com Baixa Latência:** O DynamoDB oferece latência de milissegundos em leituras e escritas, independentemente da escala.
- **Escalabilidade Automática:** Com o Auto Scaling, o DynamoDB pode ajustar automaticamente a capacidade de leitura e escrita da tabela com base na carga de trabalho.

Capacidade Provisionada vs. Capacidade Sob Demanda:

- **Provisionada:** Você configura a capacidade de leitura e escrita, e o DynamoDB aloca os recursos necessários para suportar essa carga.
- **Sob Demanda:** Você paga apenas pelo que utiliza, o que é ideal para cargas de trabalho imprevisíveis.

Modelos de Consistência:

- **Leitura Consistente Eventualmente:** Retorna os dados mais recentes, mas sem garantia absoluta de que seja o mais recente.
- **Leitura Fortemente Consistente:** Garante que a leitura sempre retorna os dados mais recentes.

DAX (DynamoDB Accelerator):

O DynamoDB Accelerator (DAX) é um cache na memória altamente disponível e gerenciado que pode acelerar as leituras do DynamoDB em até 10x, reduzindo a latência para microssegundos, mesmo sob carga elevada.

Funcionamento:

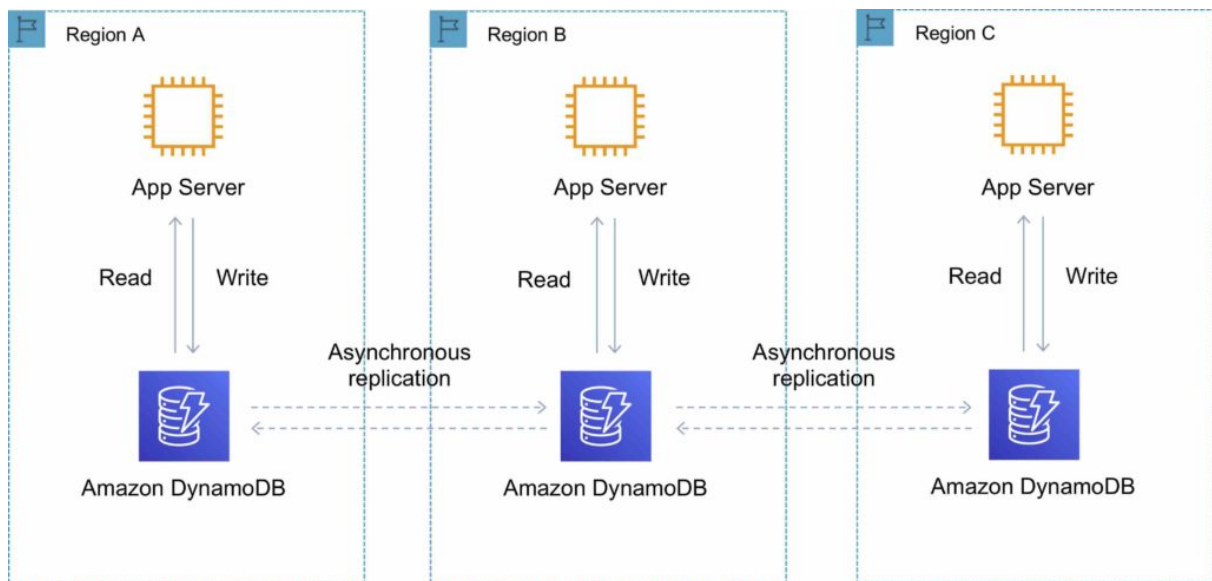
- O **DAX armazena em cache os resultados de consultas**, leituras de chave-valor e operações de leitura de tabelas do DynamoDB. Quando uma aplicação faz uma leitura de dados, o DAX verifica primeiro o cache para ver se os dados estão disponíveis. Se os dados não estiverem no cache, o DAX os busca no DynamoDB, armazena no cache, e retorna para a aplicação.

Consistência:

- **Cache de Leitura Consistente:** Mesmo com o DAX, você pode escolher entre leituras consistentes eventualmente ou fortemente consistentes, dependendo de suas necessidades.
- **Gerenciamento de Cache:** O DAX cuida automaticamente da invalidação de cache e da atualização dos dados, garantindo que as leituras sejam rápidas e confiáveis.

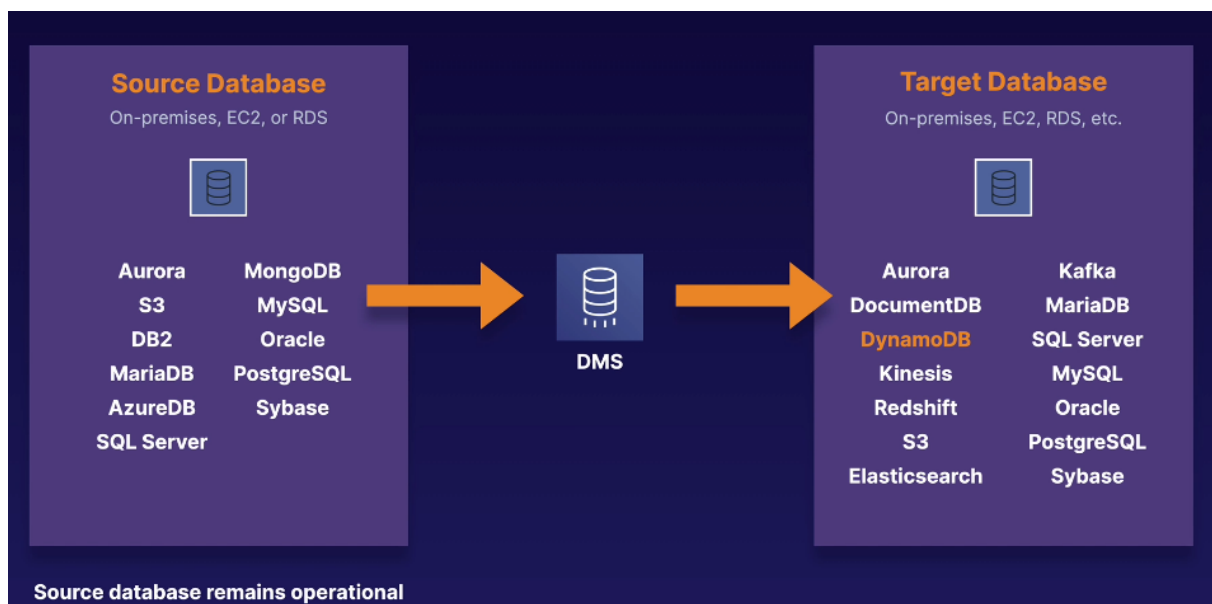
Cross Region Replication (Com o Global Tables)

- O Global Tables do DynamoDB nos dá a opção de replicar um banco em outra região, garantindo maior disponibilidade.
- Quando se cria uma réplica, deve se especificar qual região você deseja criá-la.
- Essa função é bem interessante para que se possa oferecer uma latência baixa para os usuários, independente de qual local do mundo ele está acessando, uma vez que podemos distribuir carga de trabalho entre as réplicas.



Database Migration Service (DMS)

Este serviço oferece uma migração entre SGBDs de maneira altamente controlada e consistente, toda a migração é feita enquanto o banco de destino continua operacional, o que faz com que os seus negócios não sejam afetados!'



Como referenciado na imagem acima, deve se escolher um banco de "destino" e outro banco "alvo", o DynamoDB, por sua vez, só possui a opção como banco alvo (não é possível fazer um egresso de dados do dynamo utilizando DMS)

Segurança

- Criptografado usando KMS (Encryption at rest)
- Permite utilização de VPN site-to-site
- Suporta políticas e roles do IAM
- Opção de acesso granulado (permitir somente determinadas funções para cada usuário)
- Monitorável com CloudWatch e CloudTrail

Redshift

O Amazon Redshift é o serviço de data warehouse da Amazon, projetado para a armazenagem de Petabytes de dados de diversas fontes, a extração de dados para relatórios de BI, por exemplo, ocorre aqui, já que fazer queries de extração em um banco de produção seria um workload bem grande, não?

Características

- **Armazenamento em Colunas:**
 - Redshift utiliza um armazenamento em colunas, o que significa que dados de cada coluna são armazenados juntos, permitindo uma compressão eficiente e aumentando a velocidade de consultas analíticas que acessam colunas específicas em grandes conjuntos de dados.
- **Consultas Massivamente Paralelas (MPP):**
 - Redshift distribui as operações de consulta entre múltiplos nós de computação, permitindo a execução simultânea de diversas consultas com alta performance.
- **Integração com AWS:**
 - Redshift se integra facilmente com outros serviços da AWS, como Amazon S3, Amazon RDS, Amazon EMR, e Amazon DynamoDB, permitindo a ingestão de dados de várias fontes e simplificando a criação de pipelines de dados.
- **Redshift Spectrum:**

- Permite executar consultas diretamente em dados armazenados no S3 sem a necessidade de carregá-los para o cluster Redshift, proporcionando maior flexibilidade e economia de custos.
- Um detalhe importante é que o Redshift é disponibilizado em apenas 2 Availability Zones

Aurora

Aurora é um serviço de banco de dados relacional da AWS que combina a alta performance e disponibilidade de bancos de dados comerciais com a simplicidade e o custo-efetivo dos bancos de dados open-source. o Aurora é compatível com PostgreSQL e MySQL.

Principais Características:

1. Alta Performance:

- Aurora é projetado para ser até cinco vezes mais rápido que o MySQL e três vezes mais rápido que o PostgreSQL padrão, oferecendo baixa latência e alto throughput.

2. Alta Disponibilidade e Escalabilidade:

- Ele é replicado automaticamente em múltiplas AZs, garantindo alta disponibilidade.
- Pode escalar automaticamente tanto em termos de armazenamento (até 128 TB por instância) quanto de capacidade de leitura com réplicas de leitura.

3. Segurança:

- Aurora oferece criptografia em repouso e em trânsito, e integrações nativas com IAM para controle de acesso.

4. Automação e Gerenciamento:

- Inclui backups automáticos, recuperação point-in-time, patching automático e outras funcionalidades gerenciadas, liberando da manutenção rotineira.

▼ Tipos de Réplica

O Aurora oferece 3 tipos de réplicas: Aurora Replica, MySQL Read Replica, PostgreSQL Read Replica:

Aurora Replicas:

- **Definição:** As Aurora Replicas são réplicas de leitura criadas dentro do cluster de banco de dados Aurora. **Suportam failover automático.**
- **Características:**
 - **Permite até 15 réplicas dentro de uma região,** com latência de leitura inferior a 10 milissegundos, pois as réplicas compartilham o mesmo armazenamento subjacente com o banco de dados principal.
 - **Failover Automático:** Em caso de falha na instância primária, uma das Aurora Replicas pode ser promovida automaticamente para ser a nova instância principal, garantindo alta disponibilidade.
 - **Sincronização Rápida:** Devido ao armazenamento compartilhado, as réplicas estão sempre atualizadas com a instância principal, minimizando a latência de replicação.

MySQL/PostgreSQL Read Replicas:

- Você também pode criar **até 5 Read Replicas** MySQL ou PostgreSQL. Essas réplicas podem ser criadas tanto dentro do mesmo cluster quanto em outra região (cross-region), usando replicação assíncrona.
- **Características:**
 - **Cross-Region Replication:** Útil para casos onde você precisa de read replicas em diferentes regiões para minimizar a latência para usuários em diferentes partes do mundo.
 - **Independência:** Essas réplicas não compartilham o armazenamento com a instância principal e são replicadas de forma assíncrona, o que pode resultar em maior latência de replicação.
 - **Capacidade de Promoção:** Essas réplicas também podem ser promovidas a instâncias principais, mas **SEM suporte ao failover automático.**

Aurora Serverless

Essa é a configuração que permite que o Aurora escale automaticamente sua capacidade com base na demanda da aplicação, sem a necessidade de gerenciamento manual de instâncias de banco de dados. É eficiente para workloads imprevisíveis.

Elastic MapReduce

EMR é um serviço que facilita o processo de ETL de grandes volumes de dados, utilizando frameworks como Hadoop, Spark etc. Ele permite que você configure clusters de processamento de dados de forma rápida e fácil, escalando de acordo com a demanda, e pagando apenas pelo uso dos recursos.

Principais Características:

1. Processamento em Escala:

- Pode processar petabytes de dados de maneira eficiente, utilizando clusters de EC2 que podem ser dimensionados automaticamente conforme a carga de trabalho.

2. Integração com o Ecossistema AWS:

- Integrado nativamente com outros serviços, como o S3, DynamoDB, RDS, Redshift, e IAM, permitindo fluxos complexos e seguros.

3. Custo-Benefício:

- Você pode usar instâncias Spot para reduzir os custos do cluster, e ainda configurá-lo para redimensionar automaticamente com base no workload, otimizando os custos.

4. Gestão e Automação:

- O EMR automatiza muitas tarefas de configuração e gerenciamento do cluster, incluindo provisionamento, configuração de nodes, aplicação de patches de software e monitoramento de desempenho.

Anotações

- O RDS roda em uma EC2, no entanto você não consegue entrar no SO delas, somente no serviço de BD.
- RDS não é serverless, a menos que o SGBD seja um AURORA SERVERLESS
- É necessário habilitar o Automatic Backups para dar deploy em uma Read Replica.
- Read Replicas podem ter Multi-AZ habilitado, além de também poderem ser promovidas a banco de dados principal
- **Se você estiver utilizando uma RDS single-AZ e precisar fazer um backup ou snapshot, as operações de I/O serão suspensas, cuidado!!!**