DAT302

AWS re:INVENT

Deep Dive on Amazon Relational Database Service (RDS)

Brian Welcker

Principal Product Manager, Amazon RDS

November 28, 2017

Amazon RDS enables customers to launch an optimally configured, secure and highly available database with just a few clicks. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business. Amazon RDS provides you six database engines to choose from, including Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB. In this session, we take a closer look at the capabilities of the RDS service and review the latest features available. We do a deep dive into how RDS works and the best practices to achieve the optimal performance, flexibility, and cost saving for your databases.

What is Amazon RDS?

- Managed relational database service in the AWS cloud
- Multi-engine support: Amazon Aurora, MySQL, MariaDB, PostgreSQL, Oracle, SQL Server
- Automated provisioning, patching, scaling, replicas, backup/restore
- Easily scales to handle growth
- High availability with Multi-AZ and Amazon Aurora

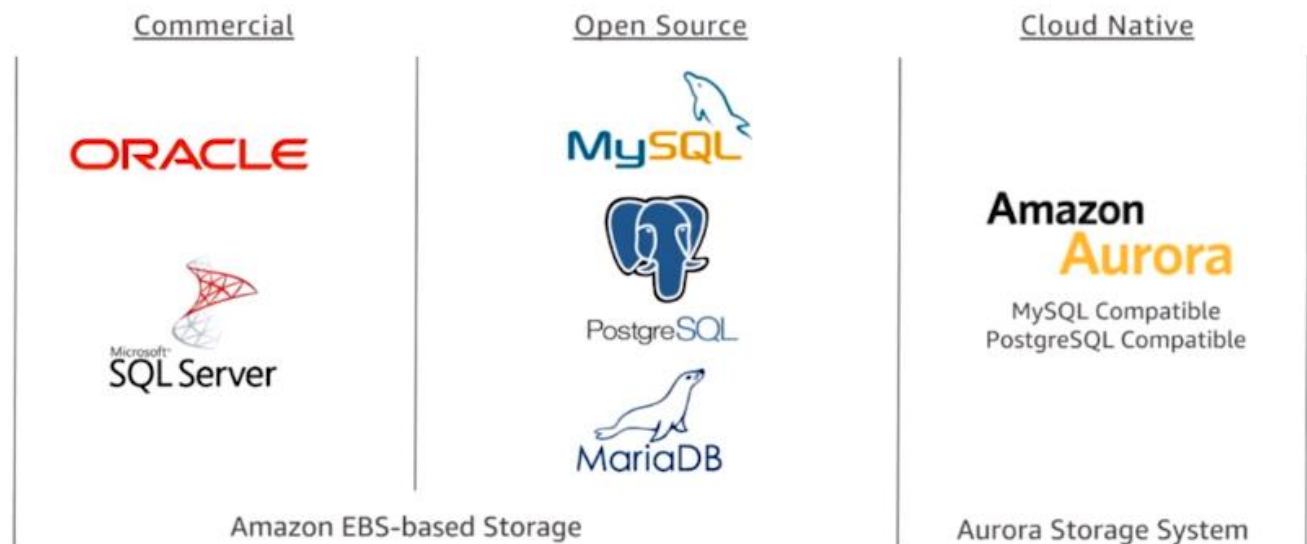
Amazon RDS

Why use Amazon RDS?

- Lower TCO because we manage "the muck"
  - Get more leverage from your teams
  - Focus on the things that differentiate you
- Built-in high availability and cross-region replication across multiple data centers
- Even a small startup can leverage multiple data centers to design highly available apps with over 99.95% availability

Configuring your database instance in Amazon RDS

The first thing you do when starting out with RDS is to pick your database engine, your storage size and the database instance that you want to run

Which RDS engine should I use?

Commercial	Open Source	Cloud Native
ORACLE	MySQL	Amazon Aurora
Microsoft SQL Server	PostgreSQL	MySQL Compatible PostgreSQL Compatible
	MariaDB	
Amazon EBS-based Storage		Aurora Storage System

Which instance type should I choose?

T2 Family
- Burstable instances
- 1 vCPU/1 GB RAM > 8 vCPU 32 GB RAM
- Moderate networking performance
- Good for smaller or variable workloads
- Monitor CPU credit metrics in Amazon CloudWatch
- T2.micro is eligible for free tier

M3/M4 Family
- General-purpose instances
- 2 vCPU/8 GiB RAM > 64 vCPU 256 GiB RAM
- High-performance networking
- Good for running CPU intensive workloads (e.g., WordPress)

R3/R4 Family
- Memory-optimized instances
- 2 vCPU/16 GiB RAM > 64 vCPU 488 GiB RAM
- High-performance networking
- Good for query-intensive workloads or high connection counts

aws

Once you pick your database engine to use in RDS, you then have to pick your instance types from the 27 EC2 instance types that breakdown into the 3 groups above. T2's are good for your variable and intermittent workloads and are free to try things out, you need to look at your workloads and the credits they use up. The M family are also good for high performance for production workloads like a big WordPress site.

Which storage type should I choose?

General purpose (GP2)
- SSD storage
- Maximum of 16 TB!
- Leverages Amazon EBS Elastic Volumes
- IOPS determined by volume size
- Minimum of 100 IOPS (below 33.33GiB)
- Bursts to 3,000 IOPS (applicable below 1.3 TB)
- Baseline of 10,000 IOPS (at 3.3 TB and above)
- Affordable performance

Provisioned IOPS (IO1)
- SSD storage
- Maximum of 16 TB!
- Leverages Amazon EBS Elastic Volumes
- Maximum of 40 K IOPS (20 K on SQL Server)
- Delivers within 10% of the IOPS performance 99.9% of the time
- High performance and consistency

Magnetic
- Magnetic storage
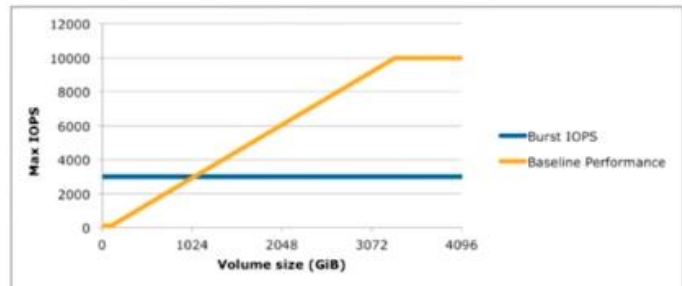- Maximum of 1 TB
- Supported for legacy databases

aws

Then you have to choose your storage type, SSD or magnetic. Always try out the GP2 storage type first to see if it fits your needs, you need to also monitor your burst credits being used in the CloudWatch logs.

How do I decide between GP2 and IO1?
Why am I not seeing 30 K IOPS?

- GP2 is a great choice, but be aware of burst credits on volumes < 1 TB
  - Hitting credit-depletion results in IOPS drop—latency and queue depth metrics will spike until credits are replenished
  - Monitor BurstBalance to see percent of burst-bucket I/O credits available
  - Monitor read/write IOPS to see if average IOPS is greater than the baseline
- Think of GP2 burst rate and PIOPS stated rate as maximum I/O rates

Chart: Max IOPS vs Volume size (GiB). Burst IOPS (flat line ~3000), Baseline Performance (rising from 0 to ~10000).

How do I scale my database instance?
Will there be downtime?

- Scale compute/memory vertically up or down
  - Handle higher load to grow over time
  - Lower usage to control costs
  - New host is attached to existing storage with minimal downtime
- Scale up Amazon EBS storage (now up to 16 TB!)
  - Amazon EBS engines now support Elastic Volumes for fast scaling (now including SQL Server)
  - No downtime for storage scaling
  - Initial scaling operation may take longer, because storage is reconfigured on older instances
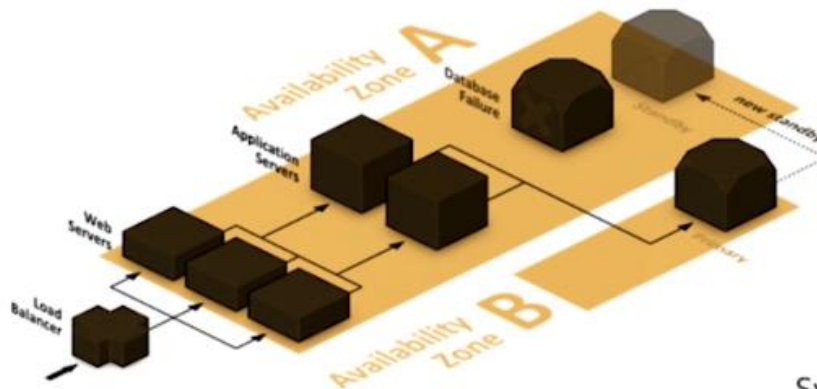 - Can re-provision IOPS on the fly

Managing high availability, read replicas, and backups in Amazon RDS

How do I ensure database high availability?

Multi-AZ provides enterprise-grade fault-tolerance
solution for production
databases

Automatic failover
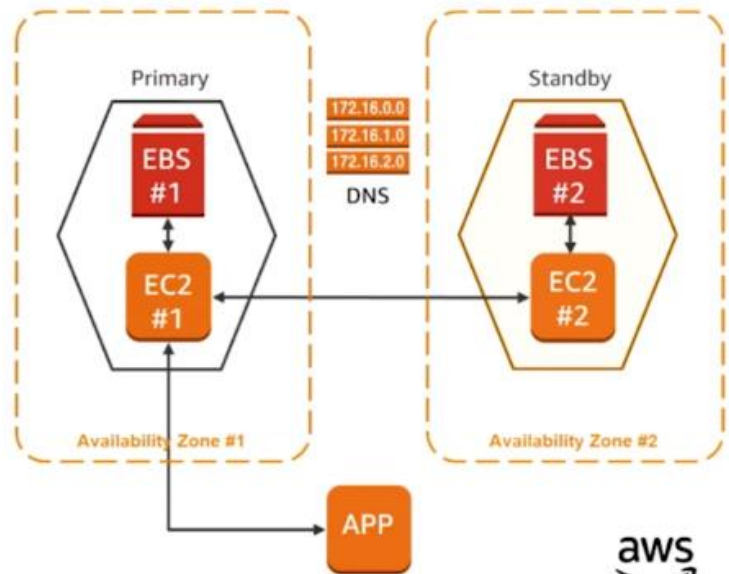Synchronous replication
Inexpensive and enabled with one click

aws

What happens during a Multi-AZ failover?
How long does it take?

- Each host manages set of Amazon EBS volumes with a full copy of the data
- Instances are monitored by an external observer to maintain consensus over quorum
- Failover initiated by automation or through the Amazon RDS API
- Redirection to the new primary instance is provided through DNS

aws

What happens during a Multi-AZ failover?

How long does it take?

- Each host manages set of Amazon EBS volumes with a full copy of the data
- Instances are monitored by an external observer to maintain consensus over quorum
- Failover initiated by automation or through the Amazon RDS API
- Redirection to the new primary instance is provided through DNS

This process takes about 1 - 2 minutes to failover to the secondary host as above. We then start to create a new standby instance on the other AZ where the failure occurred

Why would I use Read Replicas?

- Relieve pressure on your source database with additional read capacity
- Bring data close to your applications in different regions
- Promote a Read Replica to a master for faster recovery in the event of disaster
- Upgrade a Read Replica to a new engine version
- Supported for MySQL, MariaDB, and PostgreSQL
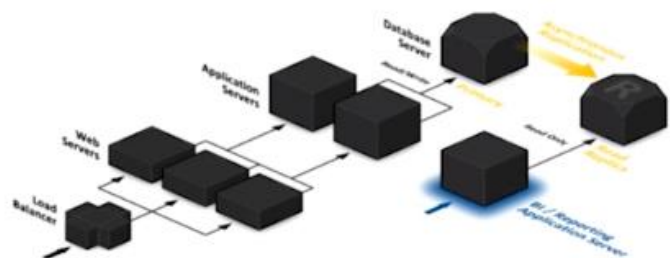
When should I use Multi-AZ as opposed to Read Replicas?

Multi-AZ

- Synchronous replication—highly durable
- Only primary instance is active at any point in time
- Backups can be taken from secondary
- Always in two Availability Zones within a Region
- Database engine version upgrades happen on primary
- Automatic failover when a problem is detected

Read Replicas

- Asynchronous replication—highly scalable
- All replicas are active and can be used for read scaling
- No backups configured by default
- Can be within an Availability Zone, cross-AZ, or cross-region
- Database engine version upgrades independently from source instance
- Can be manually promoted to a standalone database

aws

How does Amazon RDS manage backups?

- Two options – automated backups and manual snapshots
- Amazon RDS backups leverage Amazon EBS snapshots stored in Amazon S3
- Transaction logs are stored every 5 minutes in Amazon S3 to support point-in-time recovery (PITR)
- No performance penalty for backups
- Snapshots can be copied across regions or shared with other accounts
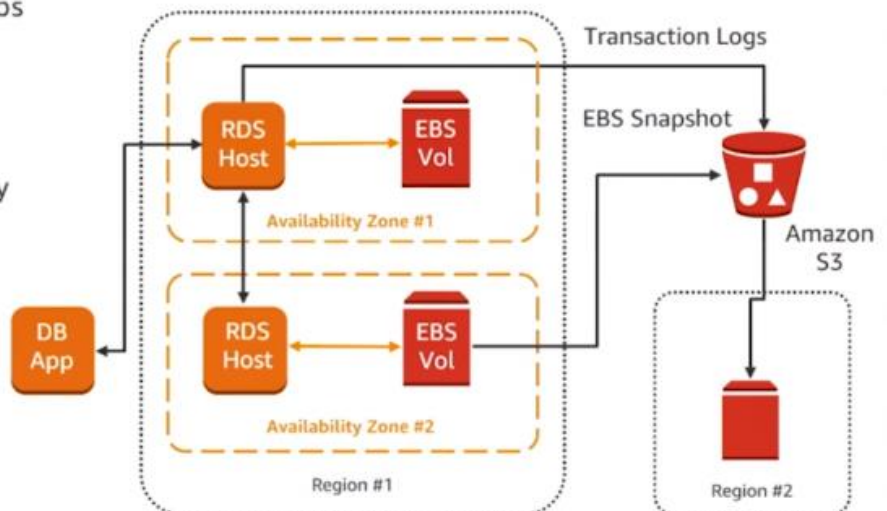
aws

When should I use automated backups as opposed to snapshots?

Automated backups

- Specify backup retention window per instance (7-day default)
- Kept until outside of window (35-day maximum) or instance is deleted
- Supports PITR
- Good for disaster recovery

Manual snapshots

- Manually created through AWS console, AWS CLI, or Amazon RDS API
- Kept until you delete them

- Restores to saved snapshot
- Use for checkpoint before making large changes, non-production/test environments, final copy before deleting a database

How do I restore a backup?
Why does it take so long?

- Restoring creates an entirely new database instance
  - Define the instance configuration just like a new instance
  - Will get the default parameter, security, and option groups
- New volumes are hydrated from Amazon S3
  - While the volume is usable immediately, full performance requires the volume to warm up until fully instantiated
  - Migrate to a DB instance class with high I/O capacity
  - Maximize I/O during restore process

Launch DB Instance

Restore time

Restore DB Instance

Instance specifications

Restoring a database will create a new database instance from the current one running, it takes a while to get the new restored large database for all the blocks to get hydrated and ready quickly.

# Securing your Amazon RDS database instance

How do I secure my Amazon RDS database?

- Amazon RDS is designed to be secure by default
- Network isolation with Amazon Virtual Private Cloud (Amazon VPC)
- AWS Identity and Access Management (IAM)-based resource-level permission controls
- Encryption at rest using AWS KMS (all engines) or Oracle/Microsoft TDE
- Use SSL protection for data in transit

What does Amazon VPC provide?

- Places your instance in a private subnet, making it secure from public routes on the Internet
- Database instance IP firewall protection lets you securely control network configuration
- Turn off *Public Accessibility* in DB instance settings to restrict access outside Amazon VPC
- Use ClassicLink to network with non-VPC resources

VPC 10.1.0.0/16

10.1.1.0/24

M

Availability Zone

AWS Region
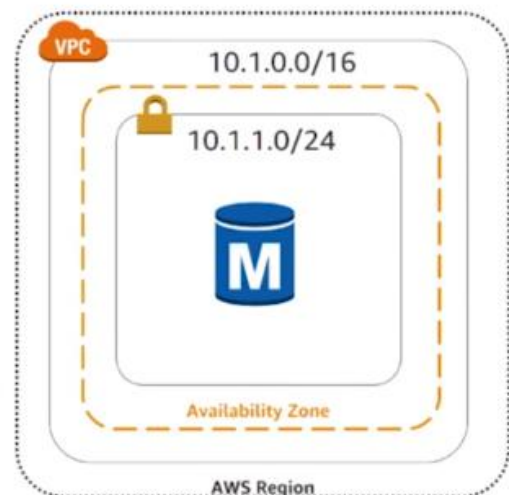
Routing rules

AWS Direct Connect

VPN connection

VPC peering

Internet gateway

aws

How do I grant access to my database?

- Use IAM to control who can perform actions on RDS resources
- Do not use AWS root credentials to manage Amazon RDS resources—you should create an IAM user for everyone, including yourself
- Can use AWS Multi-Factor Authentication (MFA) to provide extra level of protection

Controlled with IAM

DBA and Ops

Amazon RDS
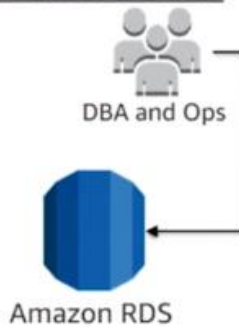
Controlled with database grants

Applications   Users and DBA

Your database

How do I encrypt my database?

- Use AWS KMS-based encryption in the AWS console
- No performance penalty for encrypting data
- Centralized access and audit of key activity
- Best practices
    - Encryption cannot be removed from DB instances
    - If source is encrypted, Read Replicas must be encrypted
    - Add encryption to an unencrypted DB instance by encrypting a snapshot copy
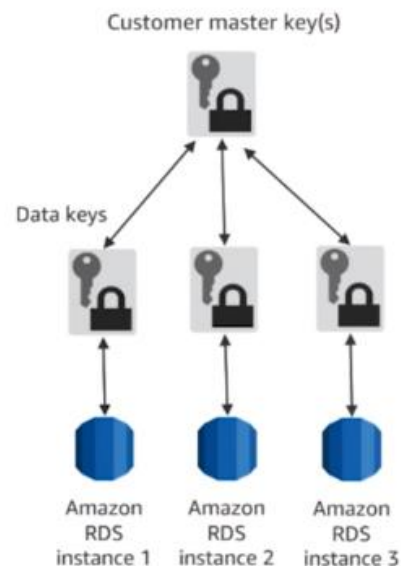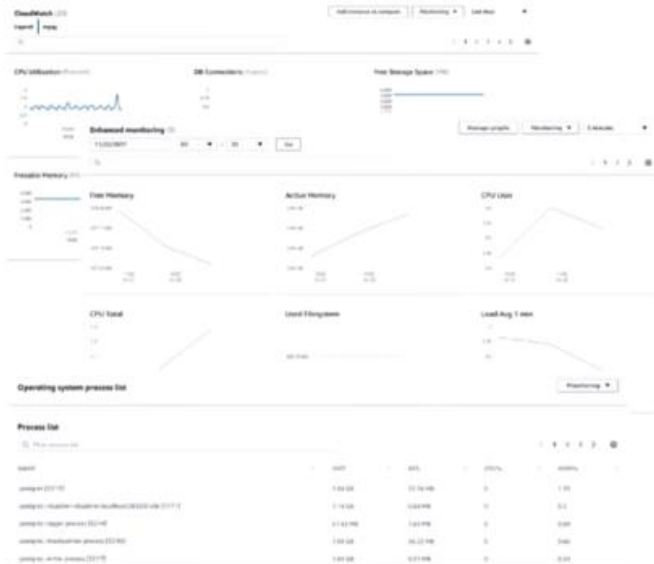
Customer master key(s)

Data keys

Amazon RDS instance 1   Amazon RDS instance 2   Amazon RDS instance 3

Monitoring your Amazon RDS database instance

How do I monitor my Amazon RDS database?

Amazon CloudWatch metrics
- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag

Amazon CloudWatch Alarms
- Similar to on-premises monitoring tools

Enhanced monitoring for Amazon RDS
- Access to over 50 CPU, memory, file system, and disk I/O metrics
- Low as 1-second intervals

Integration with third-party monitoring tools

How do I improve database performance?

- Introducing Amazon RDS Performance Insights
- Measures DB Load: Average Active Sessions (AAS)
- Identifies database bottlenecks (Top SQL)
  - Easy
  - Powerful
- Identifies source of bottlenecks
- Enables problem discovery
- Adjustable time frame
  - Hour, day, week, and longer
- Coming soon for Amazon EBS-based Amazon RDS engines

Can I know when service events happen?

- Amazon RDS uses Amazon SNS to receive notification when an event occurs
- Notifications can be in any form supported by Amazon SNS (email, a text message, or a call to an HTTP endpoint)
- Six different source types (DB instance, DB parameter group, DB security group, DB snapshot, DB cluster, DB cluster snapshot)
- 17 different event categories (availability, backup, deletion, configuration change, etc.)

Target

Send notifications to
- ARN
- New email topic

ARN

notifications ▼

Source

Source type
Instances ▼

Instances to include
- All instances
- Select specific instances

Specific instances
select instances ▼

mypg ✕   mysql5619a ✕

Event categories to include
- All event categories
- Select specific event categories

Specific event
select event categories ▼

backup ✕   maintenance ✕

aws

You can create subscriptions for events that happen in your environment so that you get notifications

Maintenance and billing for Amazon RDS

How do you maintain my database?
How can I plan for it?

- Any maintenance that causes downtime (typically only a few times per year) will be scheduled in your maintenance window
- Operating system or Amazon RDS software patches are usually performed without restarting databases
- Database engine upgrades require downtime
  - Minor version upgrades—automatic or manually applied
  - Major version upgrades—manually applied
  - Version deprecations—three- to six-month notification before scheduled upgrades
- View upcoming maintenance events in your AWS Personal Health Dashboard

Event log   ⟳ ❓

▼ Service: Relational Database Service ❶

Event	Status	Regi... ❶	Start time	Last update t...	Affected ...	Event ca...
RDS operationa...	Closed	us-east-1	October 13, 2...	October 13, 2...	-	🔧 Issue
RDS operationa...	-	-	September 13...	September 13...	1 entity	📢 Notific...
● RDS operationa...	-	us-east-1	August 20, 20...	September 13...	1 entity	📢 Notific...

aws

How am I charged for Amazon RDS?

- Database instance (instance hours)
  - Combination of Region + instance type + database engine + license (optional)
- Database storage (GB-mo)
  - Can be either provisioned (Amazon EBS) or consumed (Amazon Aurora)
  - Provisioned IOPS (IOPS-Mo) for IO1 storage type
  - Database I/O requests (IOs) for Amazon Aurora and Amazon EBS magnetic-storage types
- Backup storage (GB-mo)
  - Size of backups and snapshots stored in Amazon S3
  - No charge for backup storage up to 100% of total database storage
- Data transfer (GB-mo)
  - Uses AWS regional data-transfer pricing

Amazon RDS Free Tier

DATABASE
Amazon RDS

750 Hours

per month of db.t2.micro database usage (applicable DB engines)

Managed Relational Database Service for MySQL, PostgreSQL, MariaDB, Oracle BYOL, or SQL Server

Learn more about Amazon RDS »

EXPAND DETAILS

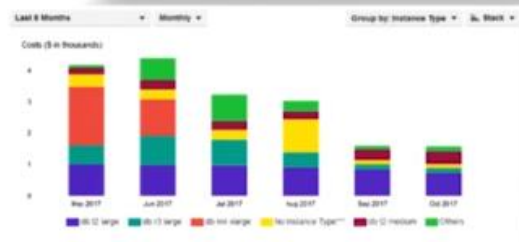
How do I understand my bill?

- Amazon RDS charges are grouped by Region
- Instances are grouped by engine
- Storage and backup charges are cross-engine
- Use AWS Cost Explorer for graphical comparison
- Use the AWS Cost & Usage Report for billing details
  - Must be enabled for account
  - Stored in your Amazon S3 bucket

How can I save money on my database?

- Amazon RDS Reserved Instances (RIs) provide a discount over on-demand prices
- Region, instance family, and engine of on-demand usage must match to apply benefit
- Amazon RDS RIs offer size flexibility for open-source and Oracle BYOL engines
- By default, RIs are shared between all accounts in consolidated billing
- Use the RI utilization report to determine how your RIs are being used
 - Support for RI coverage report coming soon

Can I stop my database when it's not in use?

- Stop and start a running database instance from the console or AWS CLI
- Available for single-AZ DB instances
- While instance is stopped, you only pay for storage
- Backup retention window is maintained while stopped
- Instances are restarted after 7 days
  - Pending maintenance operations are applied
  - Instances can be stopped again if desired
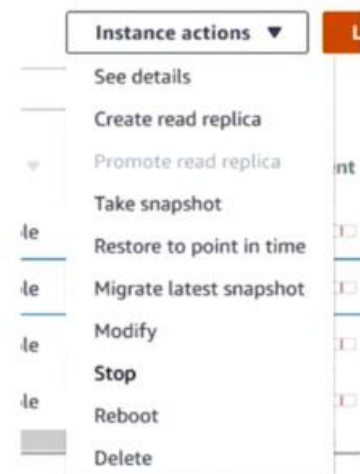
Related Sessions at re:Invent

DAT311	Technology Trends in Data Processing	Wednesday, Nov 29, 11:30 AM Venetian, Level 5, Palazzo N
DAT312	Migrating Your SQL Server Databases to Amazon RDS	Wednesday, Nov 29, 4:00 PM Venetian, Level 3, Lido 3006
DAT203	Running MySQL Databases on AWS	Thursday, Nov 30, 11:30 Venetian, Level 1, Casanova 502
DAT201	AWS Database and Analytics State of the Union - 2017	Thursday, Nov 30, 12:15 PM Venetian, Level 2, Venetian Theatre
DAT313	Running Oracle Databases on Amazon RDS	Thursday, Nov 30, 4:45 PM Venetian, Level 4, Lando 4202
DAT314	Best Practices for Running PostgreSQL on AWS	Friday, Dec 1, 9:15 AM Venetian, Level 4, Lando 4202

AWS
re:Invent

Thank you!

aws