

A D B 2 0 6

What's new in Amazon RDS

Kevin Jernigan
Principal Product Manager, RDS and Aurora
Amazon Web Services

Agenda

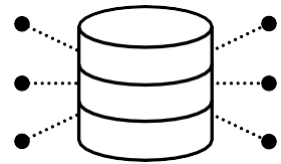
- Configuring your database instance in Amazon Relational Database Service (Amazon RDS)
- Managing high availability, read replicas, and backups in Amazon RDS
- Monitoring and troubleshooting your Amazon RDS database
- Optimizing costs in Amazon RDS
- Spotlighting new features **NEW!**

Amazon RDS: Managed relational database service with a choice of popular database engines



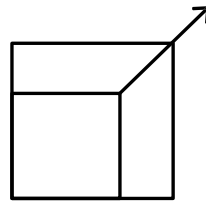
Microsoft SQL Server

ORACLE®



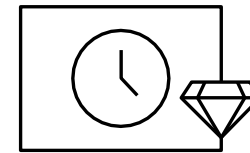
Easy to administer

Easy deployment and maintenance of hardware, the OS, and DB software; built-in monitoring



Performant & scalable

Scalable compute and storage with a few clicks; minimal downtime for your application



Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, and failover



Secure & compliant

Data encryption at rest and in transit; industry compliance and assurance programs







Configuring your database instance in Amazon RDS

Amazon RDS database engines

Commercial

Open source

Cloud native

<div>ORACLE®</div> <div>Microsoft SQL Server</div>	<div> MariaDB®</div> <div> PostgreSQL</div> <div> MySQL®</div>	<div> Amazon Aurora</div> <div>MySQL-compatible PostgreSQL-compatible</div>
Storage based on Amazon Elastic Block Store (Amazon EBS)		Aurora storage system

Which database engine version should you use?

MySQL: ~~5.5~~, 5.6, 5.7, 8.0

MySQL 8.0: Windows functions, common table expressions, JSON functions, spatial support, improved performance, crash-safe DDL operations, security roles, TLS 1.2

MariaDB: 10.0, 10.1, 10.2, **10.3**

MariaDB 10.3: Oracle compatibility (PL/SQL parser), sequences, INTERSECT and EXCEPT, new ROW type and TYPE OF stored functions, invisible columns, temporal versioned tables, user-defined aggregates, instant ADD COLUMN operations

PostgreSQL: ~~9.3~~, 9.4, 9.5, 9.6, 10, **11** **NEW!**

PostgreSQL 10: Native table partitioning, improved parallelism in query execution, ICU collation support, column group statistics, enhanced FDW and PLV8 extensions, huge pages enabled by default

PostgreSQL 11: Embedded transactions within a stored procedure, improvements to partitioning, improvements to parallelism, faster addition of columns with a non-null column default

Oracle: 11.2, 12.1, 12.2

Oracle 12.2: New engine versioning scheme—release update (.ru) and release update revision (.rur)

SQL Server: ~~2008 R2~~, 2012, 2014, 2016, 2017

Which instance type should you choose?

T family

- Burstable instances
- 1 vCPU, 1 GiB RAM - 8 vCPUs, 32 GiB RAM
- Moderate networking performance
- Is good for small or variable workloads
- You can monitor CPU credit metrics in Amazon CloudWatch
- T2.micro is eligible for the AWS Free Tier
- T3 enables unlimited mode—can burst above baseline for an extracharge **NEW!**

M family

- General purpose instances
- 2 vCPUs, 8 GiB RAM - 64 vCPUs, 256 GiB RAM
- High-performance networking
- Is good for running CPU-intensive workloads (such as WordPress)
- M5 offers up to 96 vCPUs **NEW!**

R family

- Memory-optimized instances
- 2 vCPUs, 16 GiB RAM - 96 vCPUs, 768 GiB RAM
- High-performance networking
- Is good for query-intensive workloads or high connection counts
- R5 offers up to vCPU, 768 GiB RAM **NEW!**
- Oracle support is available for X1 and X1e instances and for CPU license optimization

Which storage type should you choose?

General purpose (gp2)

- SSD storage
- Maximum of 64 TiB for Oracle **NEW!**
- Maximum of 32 TiB for PostgreSQL, MySQL, and MariaDB
- Maximum of 16 TiB for SQL Server
- Leveraging of Amazon EBS elastic volumes
- IOPS that is determined by volume size
- Minimum of 100 IOPS (below 33.33 GiB)
- Bursts of up to 3,000 IOPS (applicable below 1.3 TiB)
- Baseline of 16,000 IOPS per volume (at 5.3 TiB and above)
- Affordable performance

Provisioned IOPS (io1)

- SSD storage
- Maximum of 64 TiB for Oracle **NEW!**
- Maximum of 32 TiB for PostgreSQL, MySQL, and MariaDB
- Leveraging of Amazon EBS elastic volumes
- Maximum of 80,000 IOPS (64,000 on **NEW!** SQL Server)
- Delivery within 10 percent of the IOPS performance 99.9 percent of the time
- High performance and consistency

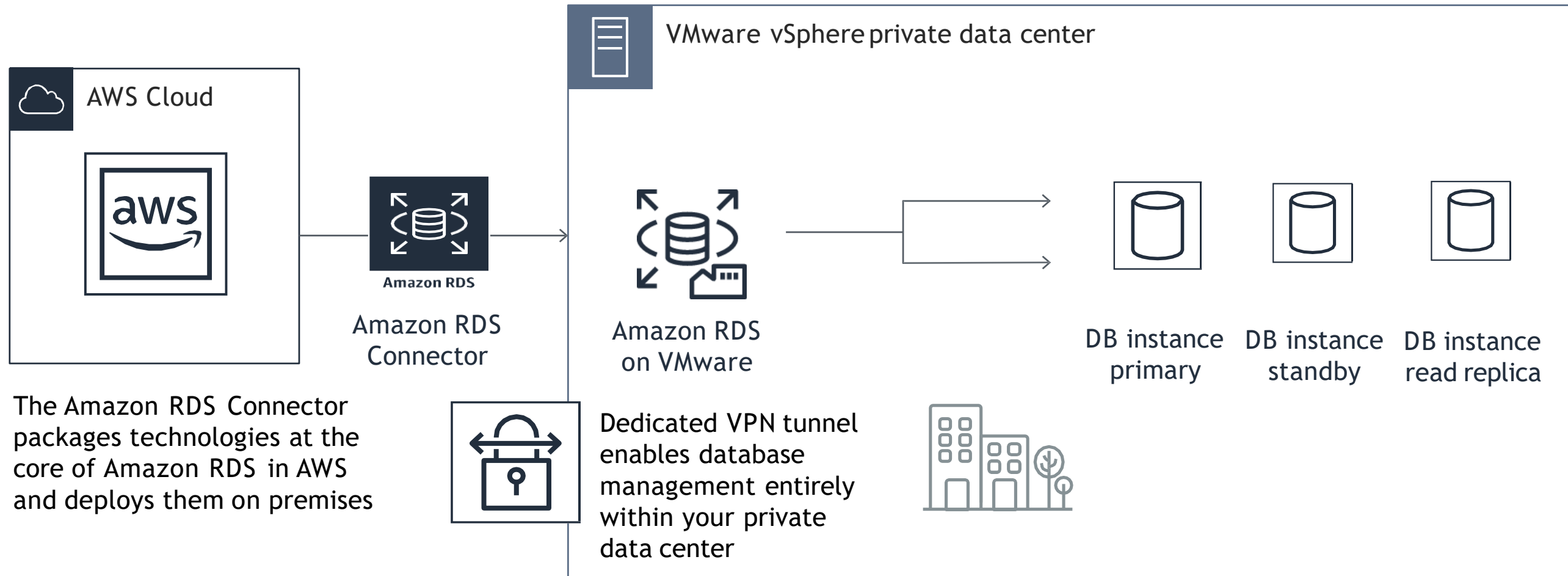
Magnetic

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

Can you run Amazon RDS in your data center?^{NEW!}

- Amazon RDS instances running in vSphere clusters (in preview)
- Use cases
 - Automated management of on-premises databases
 - Hybrid cloud backups and scaling
 - Migration of databases to AWS
- Features
 - Automated management, including provisioning and patching
 - Unified management console
 - Storage and compute scaling
 - Read replicas on premises or in AWS
 - Performance monitoring with vSphere and CloudWatch
 - Simple backup and restore on premises or in AWS
- Supported engines: MySQL, MariaDB, PostgreSQL, SQL Server, Oracle

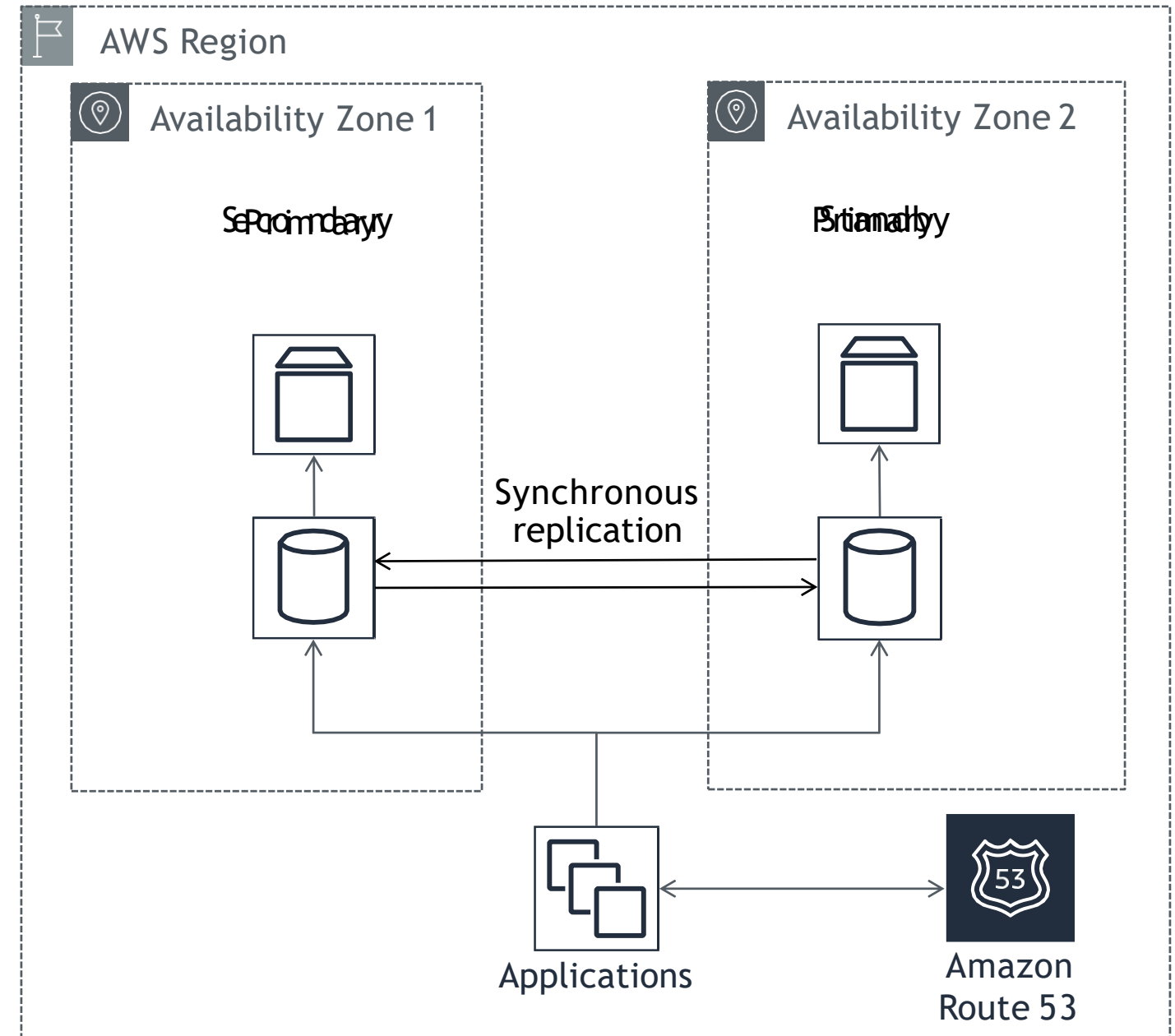
Amazon RDS on VMware **NEW!**



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

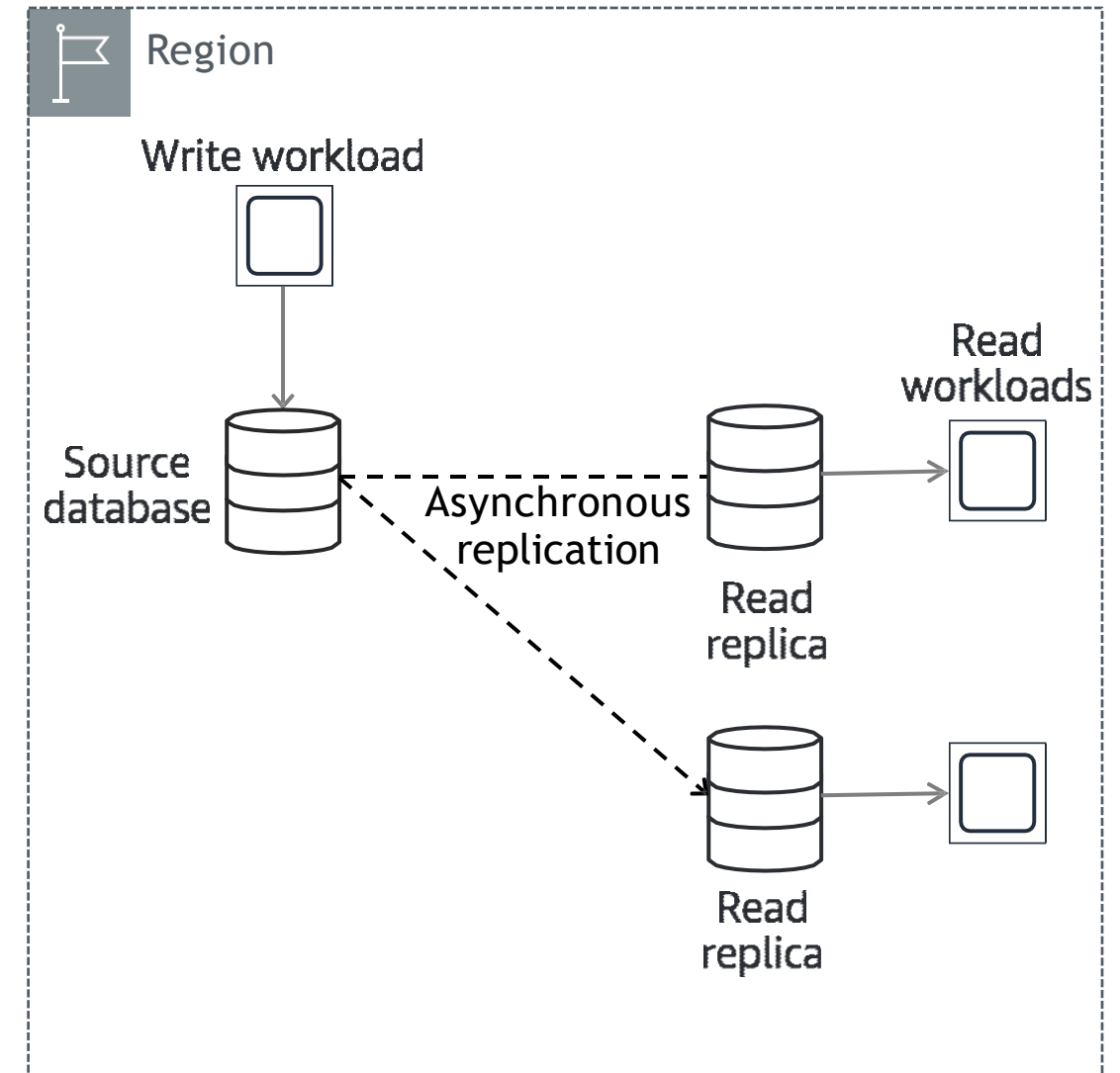
How do you ensure high availability for your database?

- Multi-AZ configurations provide enterprise-grade fault tolerance solution for production databases
- Each database host manages a 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 is initiated by automation or through the Amazon RDS API
- Redirection to the new primary instance is provided through DNS
- This detects infrastructure issues, not database engine problems



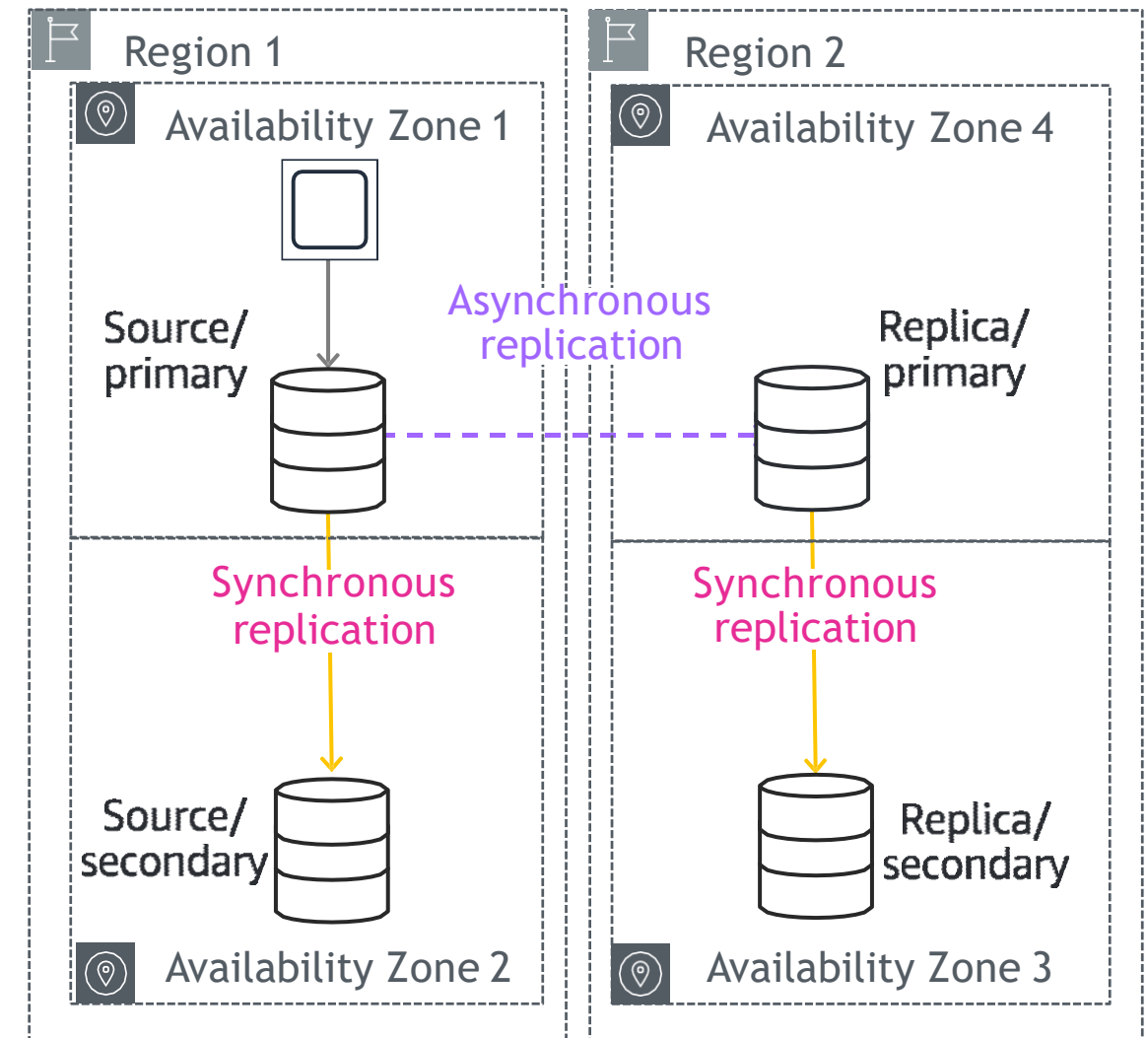
How can you gain read scalability?

- Use Amazon RDS read replicas to relieve pressure on your source database with additional read capacity
- Create up to five replicas per source database
- Monitor replication lag in CloudWatch or Amazon RDS console
- Read replicas are supported for MySQL, MariaDB, and PostgreSQL
- Single-region read replicas for Oracle **NEW!**
- This is coming soon for SQL Server **NEW!**



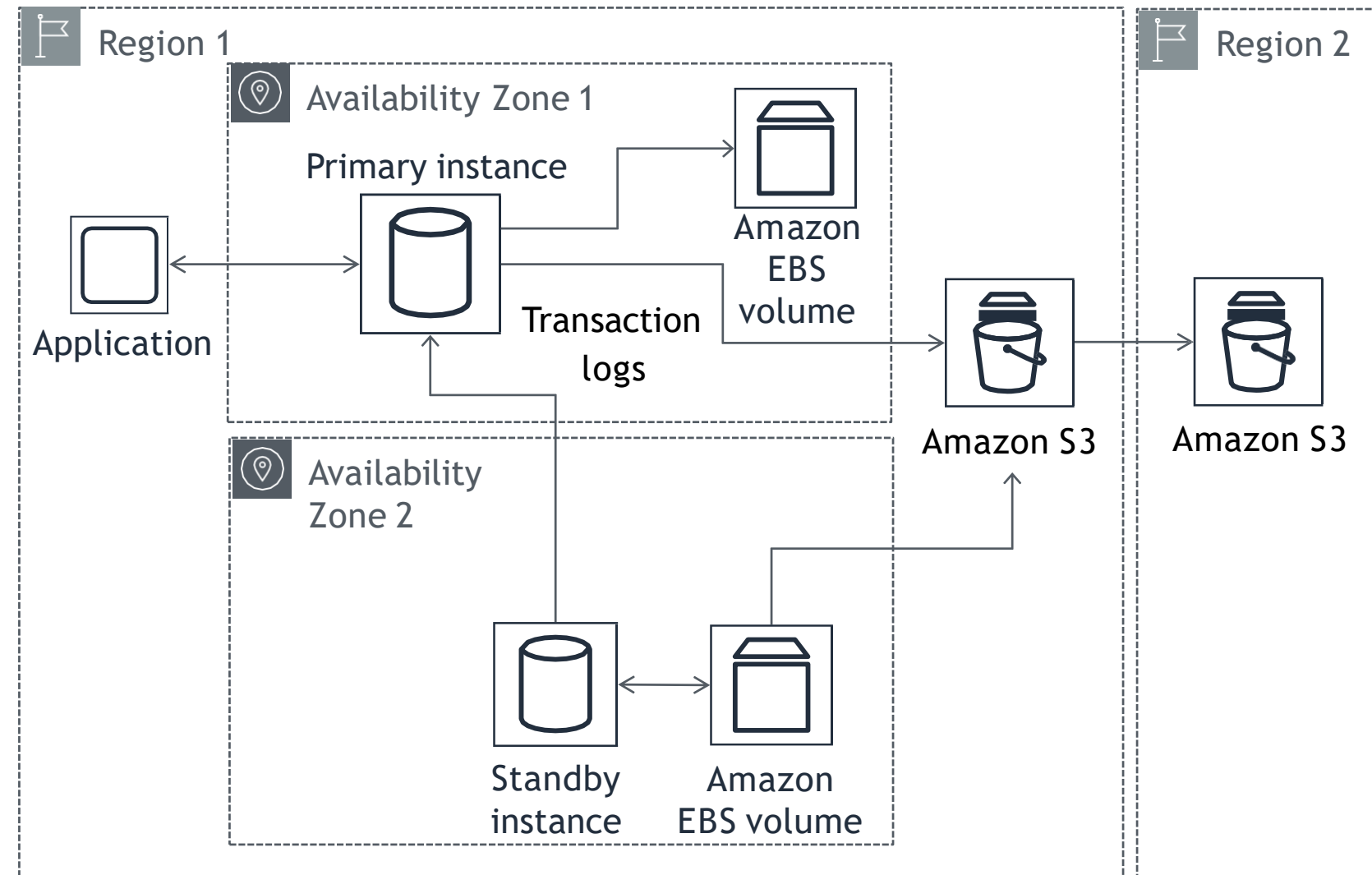
How can you plan for disaster recovery?

- Use a cross-region read replica as a standby database for recovery in the event of a disaster
- Configure read replicas for Multi-AZ to reduce recovery time
- Use delayed replication for MySQL to protect from self-inflicted disasters
- Feature is supported for MySQL, MariaDB, and PostgreSQL
- For Oracle and SQL Server, use cross-region backup copies



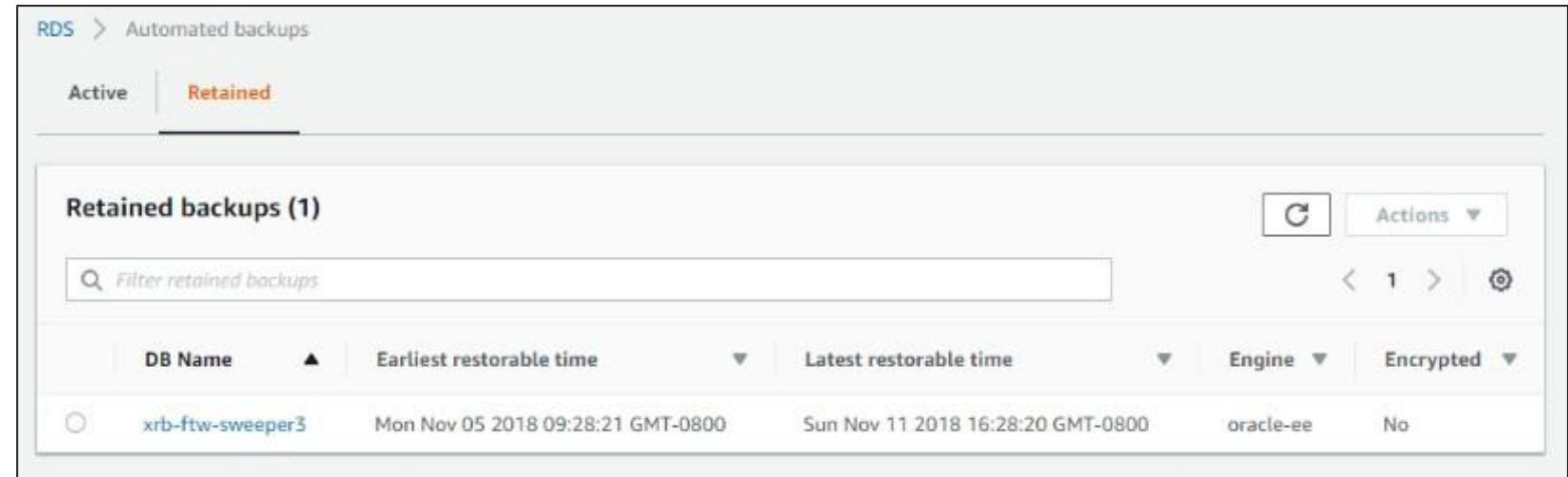
How does Amazon RDS manage backups?

- There are two options—automated backups and manual snapshots
- Amazon RDS leverages Amazon EBS snapshots stored in Amazon Simple Storage Service (Amazon S3)
- For automated backups, transaction logs are stored every five minutes in Amazon S3 to support point-in-time recovery
- There is no performance penalty for backups, and there is a brief pause for Single-AZ configurations
- Snapshots can be copied across regions or shared with other accounts



New Amazon RDS backup features

- Retain automated backups **NEW!**
 - You can optionally keep automated backups and transaction logs upon instance deletion
 - You can perform a point-in-time restore to any point during the retained period
 - Automated backups are retained for the original retention period for the instance
- Specify parameter group value on restore **NEW!**
- Copy incremental encrypted snapshots **NEW!**



RDS > Automated backups

Active Retained

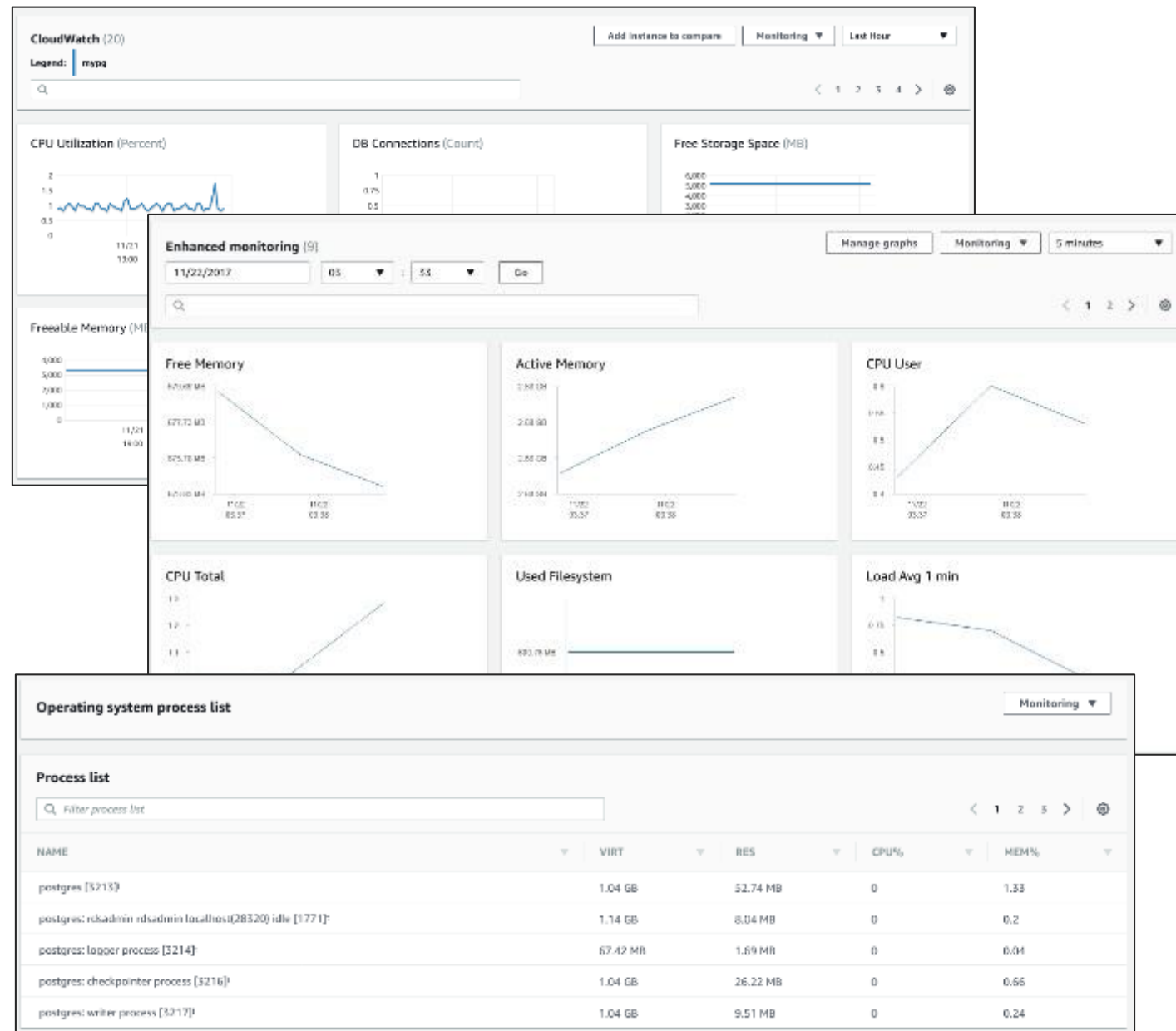
Retained backups (1)

Filter retained backups

	DB Name ▲	Earliest restorable time ▼	Latest restorable time ▼	Engine ▼	Encrypted ▼
<input type="radio"/>	xrb-ftp-sweeper3	Mon Nov 05 2018 09:28:21 GMT-0800	Sun Nov 11 2018 16:28:20 GMT-0800	oracle-ee	No

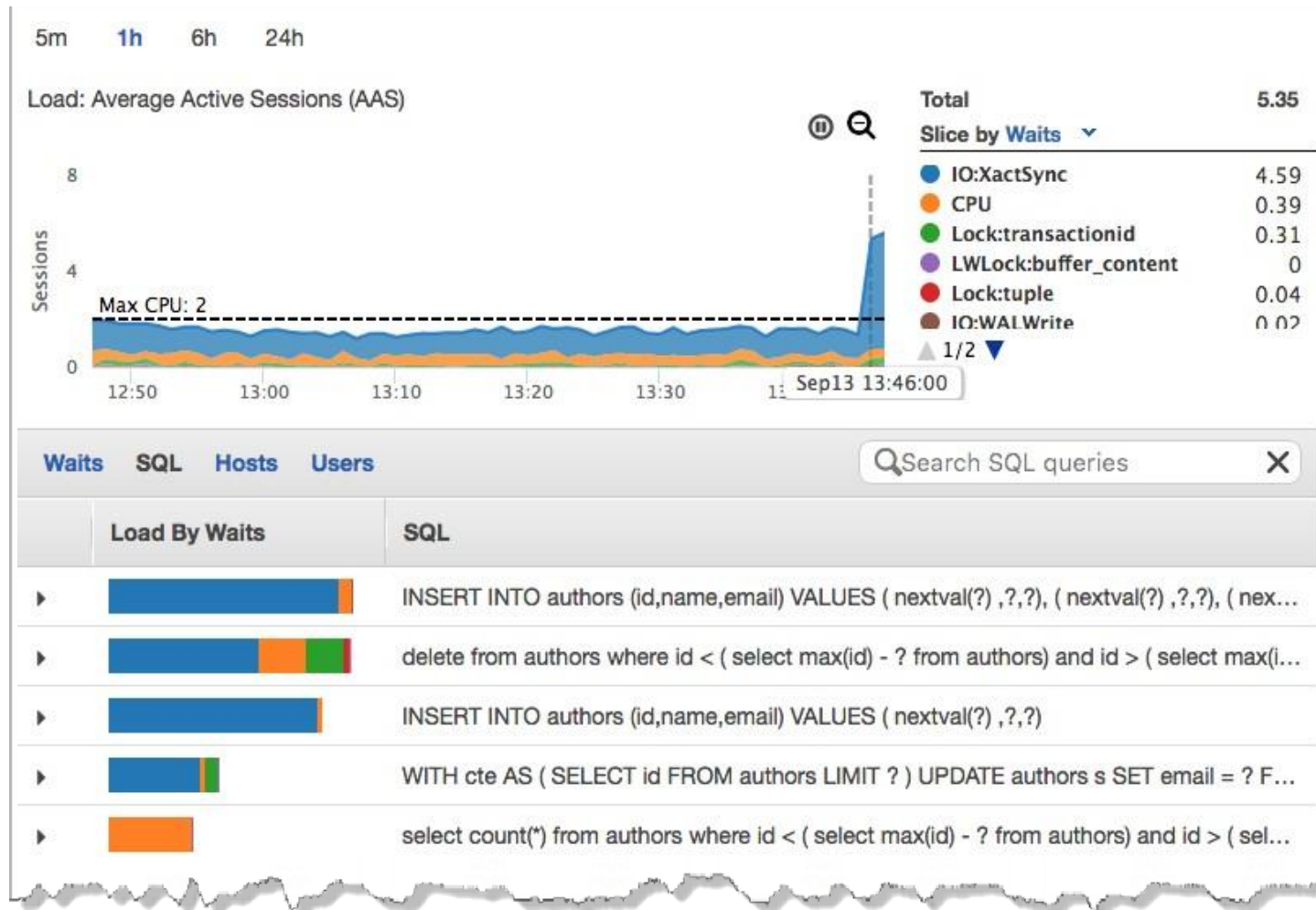
Monitoring and troubleshooting your Amazon RDS database

How can you monitor your Amazon RDS database?



- CloudWatch metrics
 - CPU, storage, memory
 - Swap usage
 - I/O (read and write)
 - Latency (read and write)
 - Throughput (read and write)
 - Replica lag
- CloudWatch alarms—similar to on-premises monitoring tools
- Enhanced monitoring
 - Access to additional CPU, memory, file system, and disk I/O metrics
 - As low as one-second intervals
- Integration with third-party monitoring tools

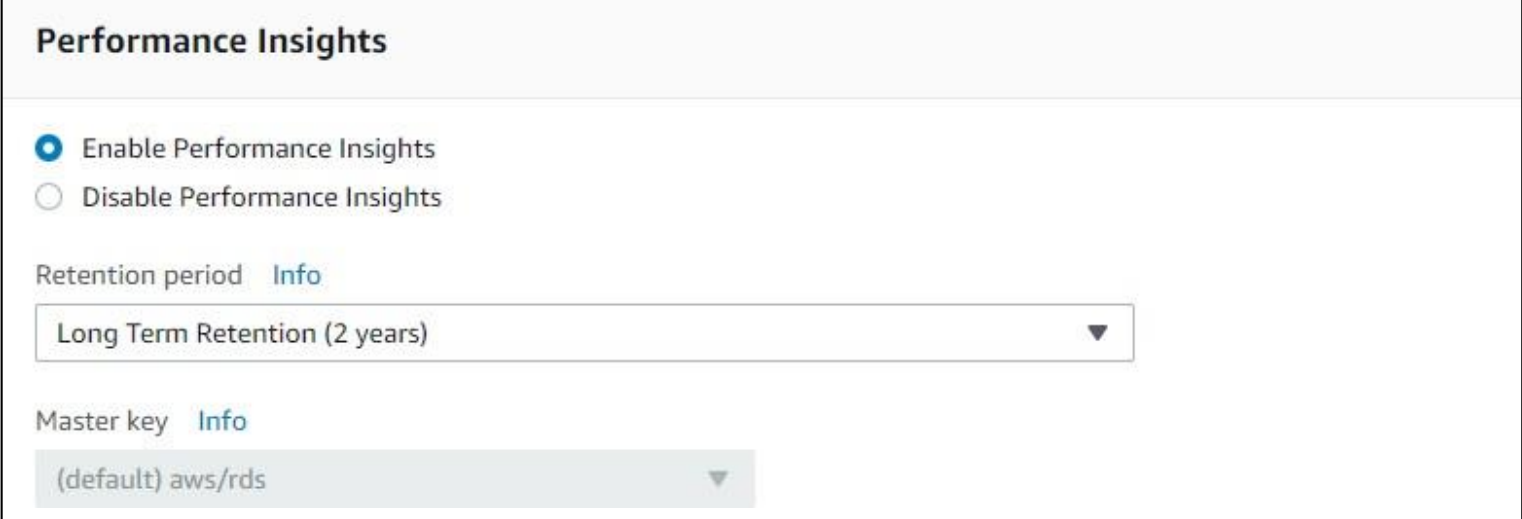
How do you troubleshoot performance problems?



- Identify database bottlenecks with Amazon RDS Performance Insights
- Measure database load—average active sessions measured through lightweight sampling
- View load by database wait states (CPU, I/O, locks)
- Slice database load by top SQL statements
- Use adjustable time frame—hour, day, week, and longer

New features in Performance Insights **NEW!**

- Engine support—all engines
- Extended data retention **NEW!**
 - Retain up to two years of performance data
 - Trend performance over time, analyze month-over-month activity, and compare end-of-quarter or end-of-year performance with earlier performance
- Load metrics in CloudWatch **NEW!**
 - DBLoad
 - DBLoadCPU
 - DBLoadNonCPU
- AWS CloudFormation support



The screenshot shows the 'Performance Insights' configuration page in the AWS console. It includes a title bar 'Performance Insights', two radio buttons for 'Enable Performance Insights' (selected) and 'Disable Performance Insights', a 'Retention period' dropdown menu set to 'Long Term Retention (2 years)', and a 'Master key' dropdown menu set to '(default) aws/rds'. Each dropdown menu has an 'Info' link next to it.

Performance Insights	
<input checked="" type="radio"/>	Enable Performance Insights
<input type="radio"/>	Disable Performance Insights
Retention period	Info
Long Term Retention (2 years) ▼	
Master key	Info
(default) aws/rds ▼	

How do you adhere to best practices? **NEW!**

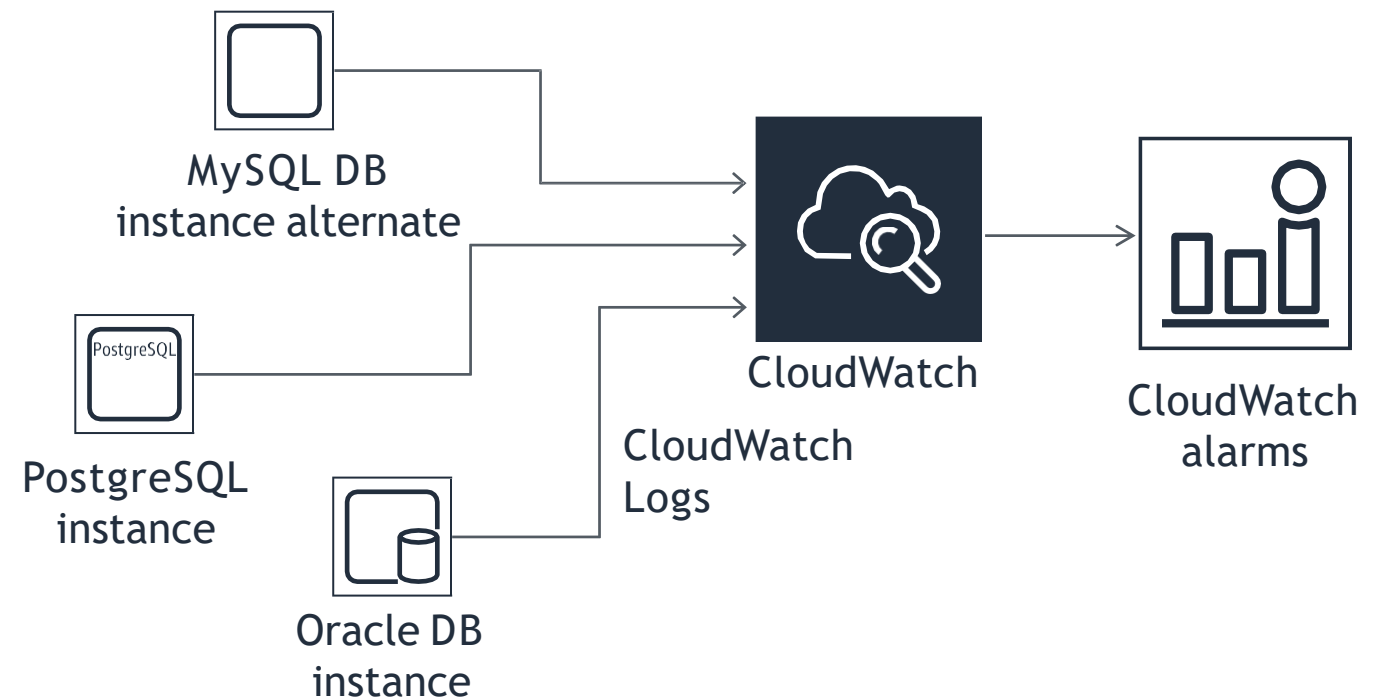
- Amazon RDS recommendations provide individualized best practice guidance by analyzing your resources
- It offers an initial set of configuration-based recommendations
- Future launches will include usage, parameters, and performance recommendations
- Results are presented in the AWS Management Console to apply immediately or schedule for the next maintenance window

The screenshot displays the 'Recommendations' section in the AWS Management Console. It features tabs for 'Active (2)', 'Dismissed (0)', 'Applied (0)', and 'Scheduled (0)'. The 'Active' tab is selected, showing a recommendation titled 'Engine version outdated (1)'. Below this, a table lists the affected DB instances. One instance, 'mysql56test', is shown with a recommendation to upgrade from MySQL 5.6.29 to the latest minor version. The recommendation includes a description, a recommendation time, and the next maintenance window. Action buttons for 'Dismiss', 'Schedule for the next maintenance window', and 'Apply now' are available. Below the table, another recommendation for 'Enhanced monitoring disabled (1)' is partially visible.

DB instances	Dismiss	Schedule for the next maintenance window	Apply now
<input checked="" type="checkbox"/> mysql56test			
Your DB instance is running mysql version 5.6.29. We recommend that you upgrade to the latest minor version because it contains the latest security fixes and other improvements.			
Sun Jul 22 20:59:17 GMT-700 2018		Fri Jul 27 02:27:00 GMT-700 2018	

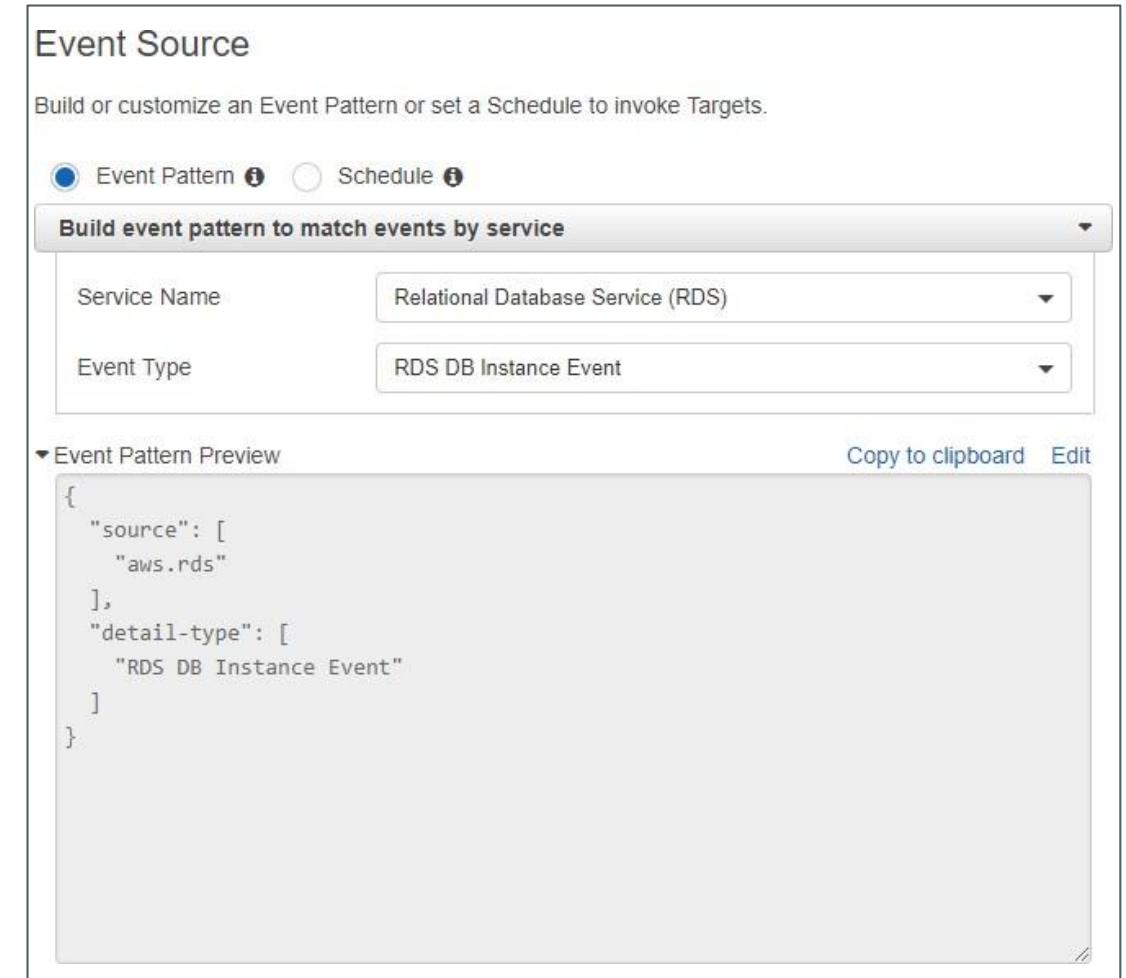
How do you manage database log files?

- You can view database engine logs in the Amazon RDS console or download them via the Amazon RDS API
- You can now publish logs in real time to Amazon CloudWatch Logs **NEW!**
 - You can set CloudWatch alarms based on text search patterns
 - Supported engines are Aurora MySQL, Amazon RDS MySQL, MariaDB, Oracle, and Amazon RDS PostgreSQL
 - This feature is coming soon for Aurora PostgreSQL



How can you track events with Amazon RDS resources?

- Amazon RDS event notifications let you know when important things happen
- There are built-in notifications for Amazon Simple Notification Service (Amazon SNS)
- RDS now publishes events to Amazon CloudWatch Events **NEW!**
 - Lets you create rules to respond to changes in resources
 - Supports cross-account event delivery
- Amazon RDS includes 6 source types (DB instance, DB parameter group, DB security group, DB snapshot, DB cluster, DB cluster snapshot)
- Amazon RDS includes 17 event categories (availability, backup, deletion, configuration change, and more)



The screenshot shows the 'Event Source' configuration page in the AWS CloudWatch console. It includes a title 'Event Source', a subtitle 'Build or customize an Event Pattern or set a Schedule to invoke Targets.', and two radio buttons for 'Event Pattern' (selected) and 'Schedule'. Below these is a dropdown menu 'Build event pattern to match events by service'. Under this menu are two dropdowns: 'Service Name' set to 'Relational Database Service (RDS)' and 'Event Type' set to 'RDS DB Instance Event'. At the bottom is an 'Event Pattern Preview' section showing a JSON pattern, with 'Copy to clipboard' and 'Edit' links.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

☒ Event Pattern ⓘ ☐ Schedule ⓘ

Build event pattern to match events by service

Service Name Relational Database Service (RDS)

Event Type RDS DB Instance Event

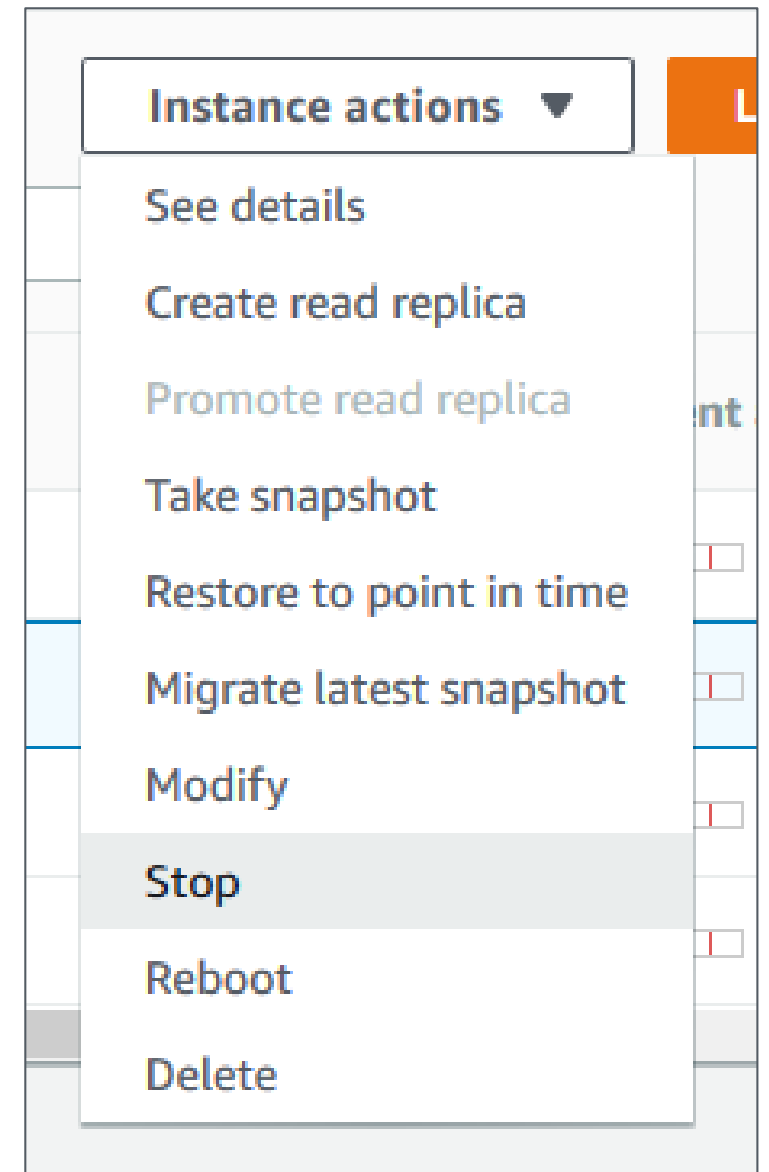
▼ Event Pattern Preview [Copy to clipboard](#) [Edit](#)

```
{
  "source": [
    "aws.rds"
  ],
  "detail-type": [
    "RDS DB Instance Event"
  ]
}
```

Optimizing costs in Amazon RDS

Can you stop and start your databases?

- This solution is for development and test environments
- You can stop and start a running DB instance from the console or AWS Command Line Interface (AWS CLI)
- It is now available for both Single-AZ and Multi-AZ DB instances and Aurora DB clusters **NEW!**
- While an instance is stopped, you only pay for storage
- Backup retention window is maintained while an instance is stopped
- Instances are restarted after seven days
 - Pending maintenance operations are applied
 - Instances can be stopped again, if needed



Can you save money with Reserved Instances (RIs)?

- Amazon RDS RIs provide a discount compared to on-demand prices
- You can match region, instance family, and engine of on-demand usage to apply benefit
- Amazon RDS RIs offer size flexibility for open-source and Oracle BYOL engines
- By default, RIs are shared among all accounts in consolidated billing
- RI utilization and coverage reports help you determine how your RIs are being used
- Amazon RDS RI recommendations report uses historical data to recommend which RIs to buy

NEW!

\$2,056	35%	8
Estimated Annual Savings*	Savings vs. On-Demand	Purchase Recommendations
Based on your past 30 days of RDS usage, we've identified 8 one-year, all-upfront RI purchase recommendations to save an estimated \$2,056 annually , representing a savings of 35% versus on-demand costs . You can take action on these recommendations in the RDS RI Purchase Console .		
		Sort by: Monthly Estimated Savings ▾ Download CSV
Purchase Recommendations (8)		Details
Buy 1 db.m4.large reserved instance <small>Size flexible**</small>		\$54.17 monthly savings
US East (Ohio) Database: MySQL License: No license required Single-AZ <i>Based on your past 30 days of on-demand usage, we recommend purchasing 1 db.m4.large reserved instance to cover 4 normalized units per hour of db.m4 family usage to maximize savings.</i>		Upfront Cost: \$883.00
View Associated RDS Usage		Recurring Monthly Cost: \$0.00
		Expected RI Utilization: 100%
Buy 4 db.t2.small reserved instances <small>Size flexible**</small>		\$40.39 monthly savings
US East (N. Virginia) Database: Aurora MySQL License: No license required Single-AZ <i>Based on your past 30 days of on-demand usage, we recommend purchasing 4 db.t2.small reserved instances to cover 4 normalized units per hour of db.t2 family usage to maximize savings.</i>		Upfront Cost: \$952.00
View Associated RDS Usage		Recurring Monthly Cost: \$0.00
		Expected RI Utilization: 100%

Thank you!

Kevin Jernigan
kmj@amazon.com