JON BONSO AND KENNETH SAMONTE

AWS CERTIFIED

DEVOPS ENGINEER PROFESSIONAL

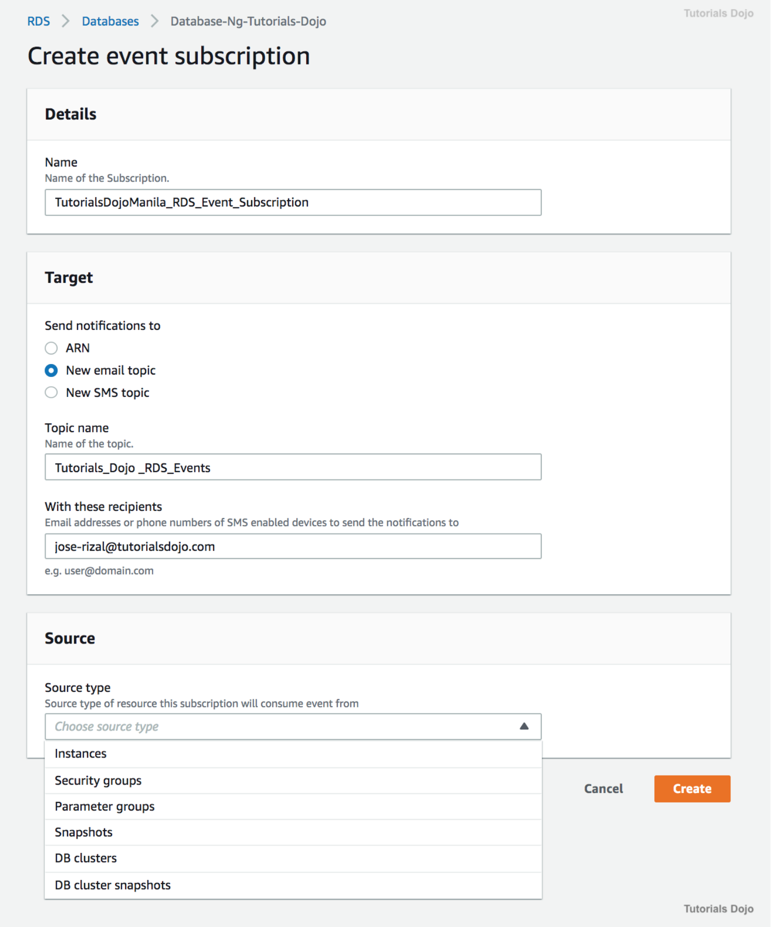
Tutorials Dojo

Study Guide and Cheat Sheets

Using default IAM policies for AWSCodeCommitPowerUser but must be limited to a specific repository only	Attach additional policy with Deny rule and custom condition if it does not match the specific repository or branch
You need to secure an S3 bucket by ensuring that only HTTPS requests are allowed for compliance purposes.	Create an S3 bucket policy that Deny if checks for condition aws:SecureTransport is **false**
Need to store a secret, database password, or variable, in the most cost-effective solution	Store the variable on SSM Parameter Store and enable encryption
Need to generate a secret password and have it rotated automatically at regular intervals	Store the secret on AWS Secrets Manager and enable key rotation.
Several team members, with designated roles, need to be granted permission to use AWS resources	Assign AWS managed policies on the IAM accounts such as, ReadOnlyAccess, AdministratorAccess, PowerUserAccess
Apply latest patches on EC2 and automatically create an AMI	Use Systems Manager automation to execute an Automation Document that installs OS patches and creates a new AMI.
Need to have a secure SSH connection to EC2 instances and have a record of all commands executed during the session	Install SSM Agent on EC2 and use SSM Session Manager for the SSH access. Send the session logs to S3 bucket or CloudWatch Logs for auditing and review.
Ensure that the managed EC2 instances have the correct application version and patches installed.	Use SSM Inventory to have a visibility of your managed instances and identify their current configurations.
Apply custom patch baseline from a custom repository and schedule patches to managed instances	Use SSM Patch Manager to define a custom patch baseline and schedule the application patches using SSM Maintenance Windows
Incident and Event Response	
Need to get a notification if somebody deletes files in your S3 bucket	Setup Amazon S3 Event Notifications to get notifications based on specified S3 events on a particular bucket.
Need to be notified when an RDS Multi-AZ failover happens	Setup Amazon RDS Event Notifications to detect specific events on RDS.

Amazon RDS Event Notifications

A database is a critical component of any enterprise system; therefore, it should be appropriately monitored for any potential issues. Like Amazon S3, there is also an event notification feature in Amazon RDS that notifies you of occurrences in your database resources. You can configure RDS to send you updates if an RDS DB instance has been created, restarted, or deleted. The RDS Event notifications can also detect low storage, configuration changes, Multi-AZ failover events, and many more.



RDS > Databases > Database-Ng-Tutorials-Dojo

Create event subscription

Details

Name
Name of the Subscription.
TutorialsDojoManila_RDS_Event_Subscription

Target

Send notifications to
○ ARN
● New email topic
○ New SMS topic

Topic name
Name of the topic.
Tutorials_Dojo _RDS_Events

With these recipients
Email addresses or phone numbers of SMS enabled devices to send the notifications to
jose-rizal@tutorialsdojo.com
e.g. user@domain.com

Source

Source type
Source type of resource this subscription will consume event from
Choose source type

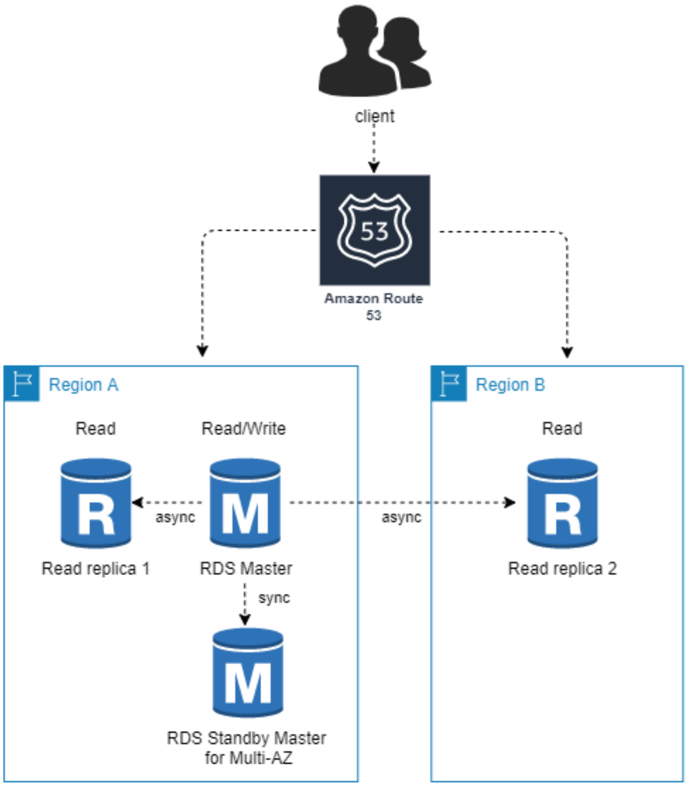
Amazon RDS produces numerous events in specific categories that you can subscribe to using various tools such as the AWS CLI, Amazon RDS Console, or the RDS API. Each event category can refer to the parameter group, snapshot, or security group of your DB instance. Moreover, you can automatically process your RDS event notifications by using an AWS Lambda function or set an alarm threshold that tracks specific metrics by creating a CloudWatch Alarm.

Source:
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Events.html

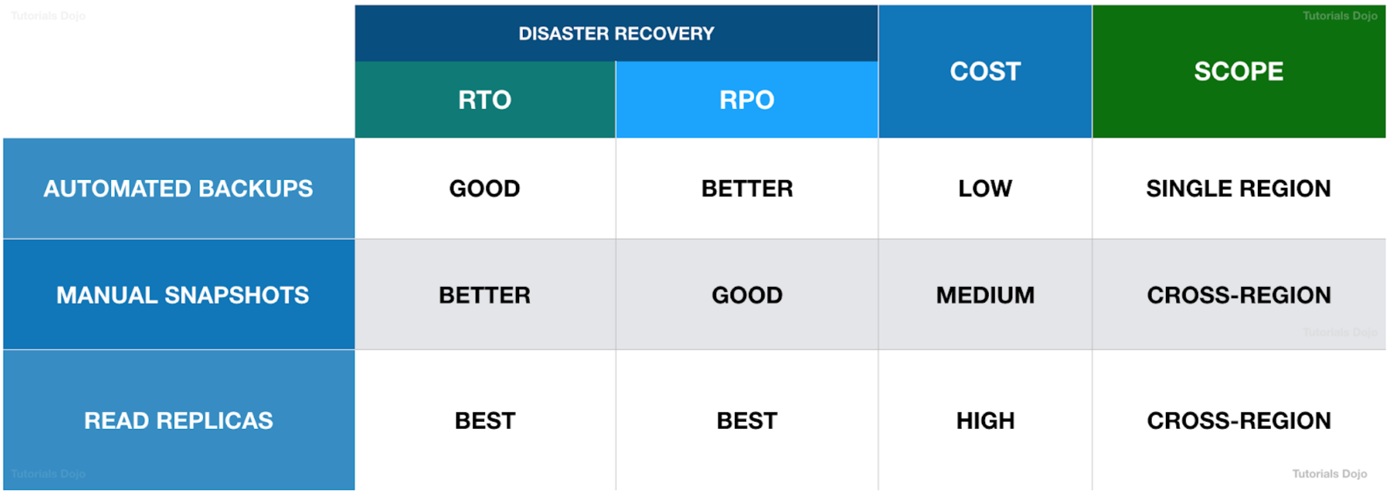
Amazon RDS Disaster Recovery Strategies

AWS provides a plethora of database services and database engines that you can use for your applications. If you need a relational database type, you can use Amazon Aurora, Amazon RDS, or Amazon Redshift. For key-value data store, you can use an Amazon DynamoDB table that you can upgrade to DynamoDB Global to support your users worldwide. Amazon ElastiCache is perfect for in-memory data stores where you can use a Redis or Memcached engine. You can choose many other database services for document, wide column, graph, time series, and ledger type databases.

Amazon Aurora is a fully managed MySQL and PostgreSQL-compatible relational database that provides high performance, availability, and scalability to your applications. Since it is a fully managed service, Amazon handles all of the underlying resources in your Aurora database and ensures that your cluster is highly available to meet your disaster recovery objectives and achieves fault tolerance. Aurora is excellent, but it has certain limitations, which compels companies to choose Amazon RDS as their database tier. Aurora does not use a native MySQL or PostgreSQL engine like RDS and can't directly run Oracle and Microsoft SQL Server databases unless you migrate them using the AWS Database Migration Service (AWS DMS) and AWS Schema Conversion Tool (AWS SCT). These constraints are the reasons why thousands of companies are still using Amazon RDS in their cloud architecture.



Amazon RDS is a managed database service. Unlike its "fully managed" counterparts, AWS does not entirely 'manage' or control all of the components of an Amazon RDS database compared to what it does for Amazon Aurora. If you launched an RDS database, you are responsible for making it highly scalable and highly available by deploying Read Replicas or using Multi-AZ Deployments configurations. You can also improve the data durability of your database-tier by taking automated or manual snapshots in RDS. For disaster recovery planning, you can set up a disaster recovery (DR) site to another AWS Region if the primary region becomes unavailable.



	DISASTER RECOVERY		COST	SCOPE
	RTO	RPO		
AUTOMATED BACKUPS	GOOD	BETTER	LOW	SINGLE REGION
MANUAL SNAPSHOTS	BETTER	GOOD	MEDIUM	CROSS-REGION
READ REPLICAS	BEST	BEST	HIGH	CROSS-REGION

An RDS Read Replica is mainly used to vertically scale your application by offloading the read requests from your primary DB instance. But do you know that it is tremendously useful for disaster recovery too? It uses asynchronous replication to mirror all the changes from the primary instance to the replica, located on the same or a different AWS Region. In contrast, the Multi-AZ Deployments configuration uses synchronous replication to keep its standby instance up-to-date. As its name implies, the standby instance is just on *standby*, meaning it neither accepts read nor write requests. This standby instance can only run on the same AWS Region, unlike a Read Replica with a cross-region capability. These unique attributes enable the Read Replica to provide the best RTO and RPO for your disaster recovery plan. You can deploy a Read Replica of your RDS database to another AWS Region to expedite the application failover if the primary region becomes unavailable without having to wait for hours to migrate and launch the automated/manual RDS snapshots to the other region.

You should also know the difference between automated backups, manual snapshots, and Read Replicas for your Business Continuity Plan (BCP). Amazon RDS has a built-in automated backups feature that regularly takes snapshots of your database and stores it on an Amazon S3 bucket that is owned and managed by AWS. The retention period of these backups vary between 0 and 35 days. It provides a low-cost DR solution for your database-tier but is only limited to a single AWS Region. Manual snapshots are the ones that you manually take yourself, hence the name. In contrast with the automated backups, the S3 bucket where the snapshots are stored is owned by you, which means that you can control its retention period and deploy cross-region snapshots. Since you manage your own RDS snapshots, you can move these across AWS Regions using a shell script or a Lambda function run by CloudWatch Events regularly.

The advantage of using Read Replicas over automated backups and manual snapshots is its near real-time synchronous replication. To put it into perspective, the replication time of the primary DB instance to the replica instance is less than a second! Compare that to the time required to move an RDS snapshot across another region and waiting for it to start up. Hence, Read Replicas provide the fastest RTO and the best RPO for your architecture. The only setback is its high cost since you have to run your replica continuously.

Sources:
https://aws.amazon.com/blogs/database/implementing-a-disaster-recovery-strategy-with-amazon-rds/
https://d0.awsstatic.com/whitepapers/Backup_and_Recovery_Approaches_Using_AWS.pdf
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CopySnapshot.html

AWS Database Services

Amazon Aurora

- A fully managed relational database engine that's compatible with MySQL and PostgreSQL.
- With some workloads, Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL.
- Aurora includes a high-performance storage subsystem. The underlying storage grows automatically as needed, up to 64 terabytes. The minimum storage is 10GB.
- **DB Clusters**
    - An Aurora DB cluster consists of one or more DB instances and a cluster volume that manages the data for those DB instances.
    - An Aurora cluster volume is a virtual database storage volume that spans multiple AZs, with each AZ having a copy of the DB cluster data.
    - Cluster Types:
        - Primary DB instance – Supports read and write operations, and performs all of the data modifications to the cluster volume. Each Aurora DB cluster has one primary DB instance.
        - Aurora Replica – Connects to the same storage volume as the primary DB instance and supports only read operations. Each Aurora DB cluster can have up to 15 Aurora Replicas in addition to the primary DB instance. Aurora automatically fails over to an Aurora Replica in case the primary DB instance becomes unavailable. You can specify the failover priority for Aurora Replicas. Aurora Replicas can also offload read workloads from the primary DB instance.
- **Aurora Multi Master**
    - The feature is available on Aurora MySQL 5.6
    - Allows you to create multiple read-write instances of your Aurora database across multiple Availability Zones, which enables uptime-sensitive applications to achieve continuous write availability through instance failure.
    - In the event of instance or Availability Zone failures, Aurora Multi-Master enables the Aurora database to maintain read and write availability with zero application downtime. There is no need for database failovers to resume write operations.
- **Monitoring**
    - Subscribe to Amazon RDS events to be notified when changes occur with a DB instance, DB cluster, DB cluster snapshot, DB parameter group, or DB security group.
    - Database log files
    - RDS Enhanced Monitoring — Look at metrics in real time for the operating system.
    - RDS Performance Insights monitors your Amazon RDS DB instance load so that you can analyze and troubleshoot your database performance.
    - Use CloudWatch Metrics, Alarms and Logs

Amazon RDS

- Supports **Aurora**, **MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server**.
- You can get high availability with a primary instance and a synchronous secondary instance that you can fail over to when problems occur. You can also use MySQL, MariaDB, or PostgreSQL Read Replicas to increase read scaling.
- You can select the computation and memory capacity of a DB instance, determined by its **DB instance class**. If your needs change over time, you can change DB instances.
- Each DB instance has minimum and maximum storage requirements depending on the storage type and the database engine it supports.
- You can run your DB instance in several AZs, an option called a **Multi-AZ deployment**. Amazon automatically provisions and maintains a secondary standby DB instance in a different AZ. Your primary DB instance is synchronously replicated across AZs to the secondary instance to provide data redundancy, failover support, eliminate I/O freezes, and minimize latency spikes during system backups.

Security

- Security Groups
  - **DB Security Groups** - controls access to a DB instance that is not in a VPC. By default, network access is turned off to a DB instance. This SG is for the EC2-Classic platform.
  - **VPC Security Groups** - controls access to a DB instance inside a VPC. This SG is for the EC2-VPC platform.
  - **EC2 Security Groups** - controls access to an EC2 instance and can be used with a DB instance.
- A *resource owner* is the AWS account that created a resource. That is, the resource owner is the AWS account of the *principal entity* (the root account, an IAM user, or an IAM role) that authenticates the request that creates the resource.
- A *permissions policy* describes who has access to what. Policies attached to an IAM identity are *identity-based policies* (IAM policies) and policies attached to a resource are *resource-based policies*. Amazon RDS supports only identity-based policies (IAM policies).
- MySQL and PostgreSQL both support **IAM database authentication**.

High Availability using Multi-AZ

- Multi-AZ deployments for **Oracle, PostgreSQL, MySQL, and MariaDB** DB instances use **Amazon's failover technology**. **SQL Server DB** instances use **SQL Server Mirroring**.
- The primary DB instance switches over automatically to the standby replica if any of the following conditions occur:
  - An Availability Zone outage
  - The primary DB instance fails
  - The DB instance's server type is changed

  - The operating system of the DB instance is undergoing software patching
  - A manual failover of the DB instance was initiated using **Reboot with failover**

Read Replicas

- Updates made to the source DB instance are asynchronously copied to the Read Replica.
- You can reduce the load on your source DB instance by routing read queries from your applications to the Read Replica.
- You can elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.
- You can create a Read Replica that has a different storage type from the source DB instance.

Backups and Restores

- Your DB instance must be in the **ACTIVE state** for automated backups to occur. Automated backups and automated snapshots don't occur while a copy is executing in the same region for the same DB instance.
- The first snapshot of a DB instance contains the data for the full DB instance. Subsequent snapshots of the same DB instance are incremental.
- The default backup retention period is one day if you create the DB instance using the RDS API or the AWS CLI, or seven days if you used the AWS Console.
- Manual snapshot limits are limited to 100 per region.
- You can copy a snapshot within the same AWS Region, you can copy a snapshot across AWS Regions, and you can copy a snapshot across AWS accounts.
- When you restore a DB instance to a point in time, the default DB parameter and default DB security group is applied to the new DB instance.