Amazon RDS Relational Database Service

March 2018

What is Amazon RDS..?

- Managed Relational Database Service
- Amazon RDS manages
  - Time consuming installation & configuration
  - Backups & Recovery
  - software patching
  - Automatic failure detection
- Access to full capabilities of the engine
- Existing Code , Applications and tools work seamlessly
- Amazon RDS does not provide shell access
- *Cannot be in STOPPED state ☹ - sometime back* but now it can be in STOPPED state

Why to use Amazon RDS?

- Supported DBMS
  - Oracle
  - Microsoft SQL Server
  - MySQL
  - PostgreSQL
  - Aurora
- Provide Automatic Backups
- Allows to Increase Capacity
  - Exception - Microsoft SQL Server
- DB instance storage
  - Magnetic
  - General Purpose (SSD)
  - Provisioned IOPS (SSD)

DB Subnet...

- A DB subnet group is a collection of subnets (typically private) that you create for a VPC and Designated for your DB instances.
- If you use the Amazon RDS console, you can just select the VPC and subnets you want to use.
- Each DB subnet group must have at least one subnet in at least two Availability Zones in the region.
- Note : For a DB instance to be publicly accessible, the subnets in the DB subnet group must have an internet gateway.
- When you create a DB instance in a VPC, you must select a DB subnet group. Amazon RDS then uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet.

...DB Subnet

- Amazon RDS creates and associates an Elastic Network Interface to your DB instance with that IP address

- For Multi-AZ deployments, defining a subnet for two or more Availability Zones in a region allows Amazon RDS to create a new standby in another Availability Zone should the need arise.

- You need to do this even for Single-AZ deployments, just in case you want to convert them to Multi-AZ deployments at some point.

Parameters Group and Options Groups

Parameter Group	Options Group
• DB Engine configuration Parameters • Container for engine configuration values • applied to one or more db instances	Additional features provided by underlying DB 1) SQL Server - mirroring 2) MySQL - Memcached and Redis engines
• Dynamic and Static parameters • Dynamic does not require REBOOT • Static requires REBOOT (read_only = 1 for READ_REPLICA)	-

Maintenance Window

- What can cause maintenance ?
  - Updating DB instance OS or cluster OS
    - Security patch
- Scaling Storage
- Scaling Compute
- Weekly maintenance window
- **Impact**
  - DB instance made offline for a short time
  - DB instances are **not backed up before updating so take a backup before applying**
  - **to minimize downtime use Multi-AZ**
    - update happens to standby
    - standby is promoted to be primary
    - update happens to old-primary
- Does not limit the total execution time of these operations

RDS – Automatic Failover

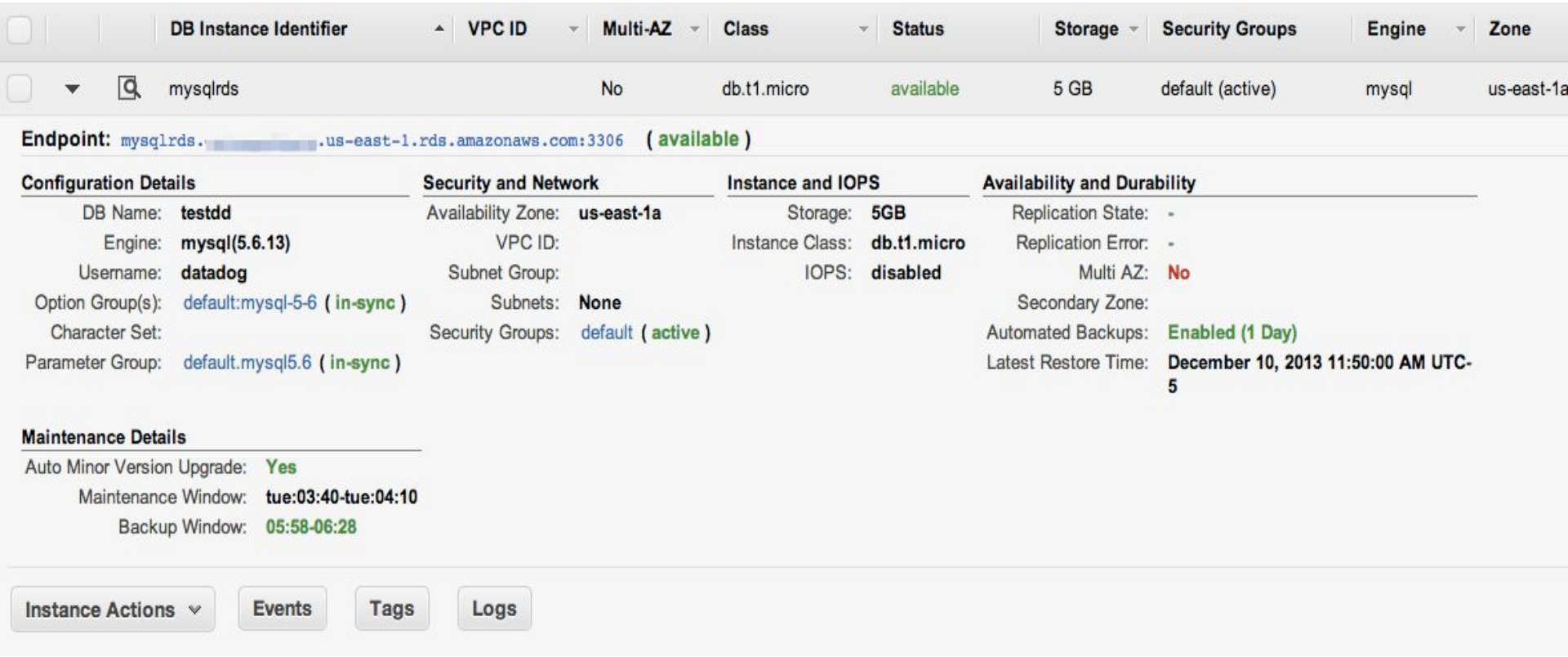
- Multi AZ deployment makes it possible to automatically failover to another AZ in case
  - AZ outage
  - Primary fails
  - Instance server type is changed
  - Manual failover initiated
  - Updating software version
  - Backups are taken against the stand by so as to reduce I/O freezes and slow down

RDS Aurora

- Drop-in replacement for MySQL
- Fully managed, MySQL-compatible, Relational DB engine
- Cost-effective Open-source DB
- Supports up to 15 Aurora Replicas
- Highly durable with less possibility of data loss
- Maximum size of 64 terabytes (TB)

Understanding RDS Instance Details

- MySQL DB Engine
  - Configuration Details
  - Security and Network
  - Instance and IOPS
  - Availability and Durability
  - Maintenance Details



RDS Lifecycle

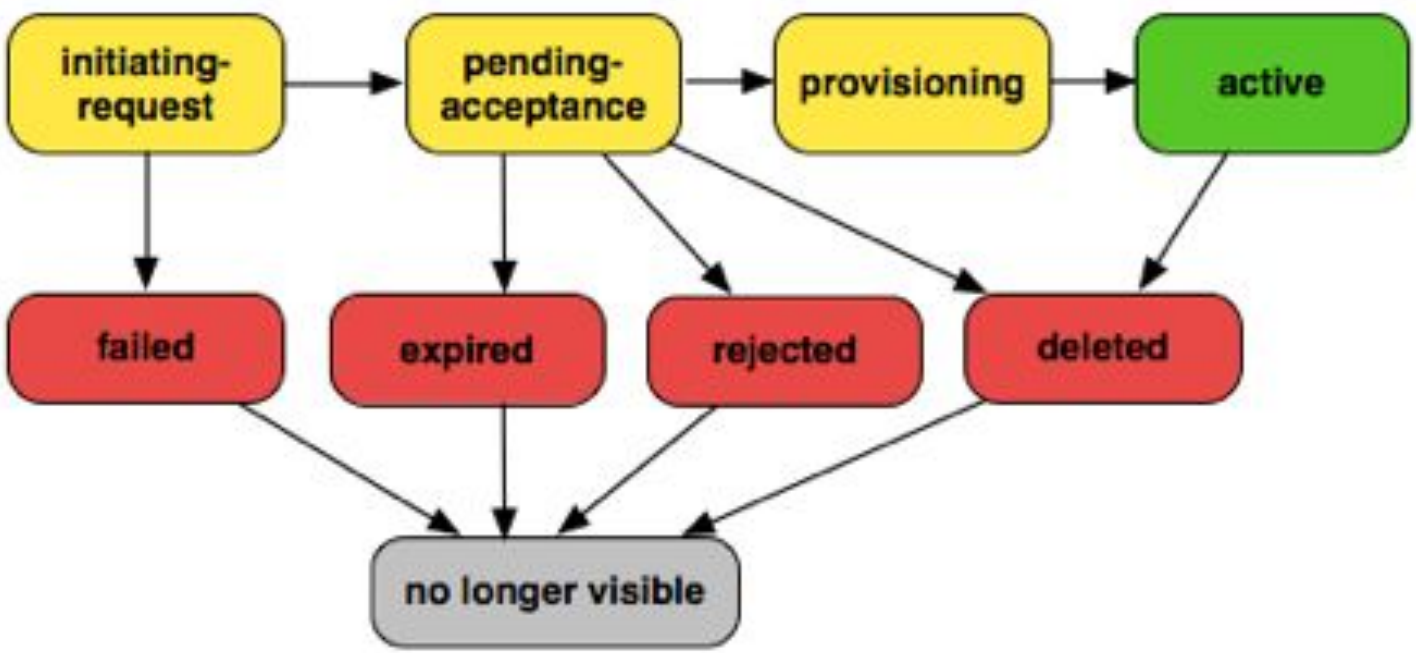


Image Source: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/images/peering-lifecycle-diagram.png

Before you jump with RDS

Check out the limitations with each database

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SQLServer.html

SQL Server does not support following plan

- Database mail
- Service Broker
- Windows Authentication
- Replication
- Distribution Transaction Coordinator (MSDTC)
- FILESTREAM support
- ….

Best Practices

- Scaling Amazon RDS
  - Read Replica
  - Instance type upgradation
- Securing the System
  - Security Group
- Backup and Recovery
  - Regular Snapshots
  - Multi-AZ RDS
  - Enable Automatic Backups
- DB Instance Recommendations
  - CPU utilization under 80%
  - Enough memory space

Read Replicas

Read Replica

- Strategy used to improve the performance of the solution as the read load is distributed across the read replicas
- Side effect and benefit is that it provides High Availability for RO operations
- Replica is Read Only so you cannot write to it
- Replica is kept in sync by RDS engine
  - for ‘MySQL’ , ‘PostgreSQL’ it uses underlying DB engine’s asynchronous replication technique
  - Aurora uses proprietary technique
  - Read Replica Aurora uses the same SHARED VIRTUAL STORAGE as master so there is no need to copy data to synchronize
  - Aurora results in faster and cheaper solution compared to MySQL

- ‘Read Replica’ always lags behind from master in a given point in time (< 10 ms)
- Read Replica is NOT SUPPORTED BY SQL SERVER RDS

Benefits

- Enhanced Performance
- Improved Availability

Read Replicas and Multi AZ Deployment Comparison

Parameter	Multi AZ Deployment	Read Replicas
Replication to other instance	Synchronous	Asynchronous

What can we do with Snapshots ?

1. Restore - RDS instance from SNAPSHOT
2. Migrate - Migrate to Aurora Database
3. Copy - Copy to any region (including the one where it is stored)
4. Delete - Obvious !

Pricing of Read Replicas

- MySQL RDS - USD 75
  - db.m3.medium
  - Single AZ Deployment
  - Storage : General Purpose : 100 GB
- With 1 Read Replica - USD 75 + USD 75 = 150
- Same as standard instance

HIGH AVAILABILITY AND DR OPTIONS

Current Limitations

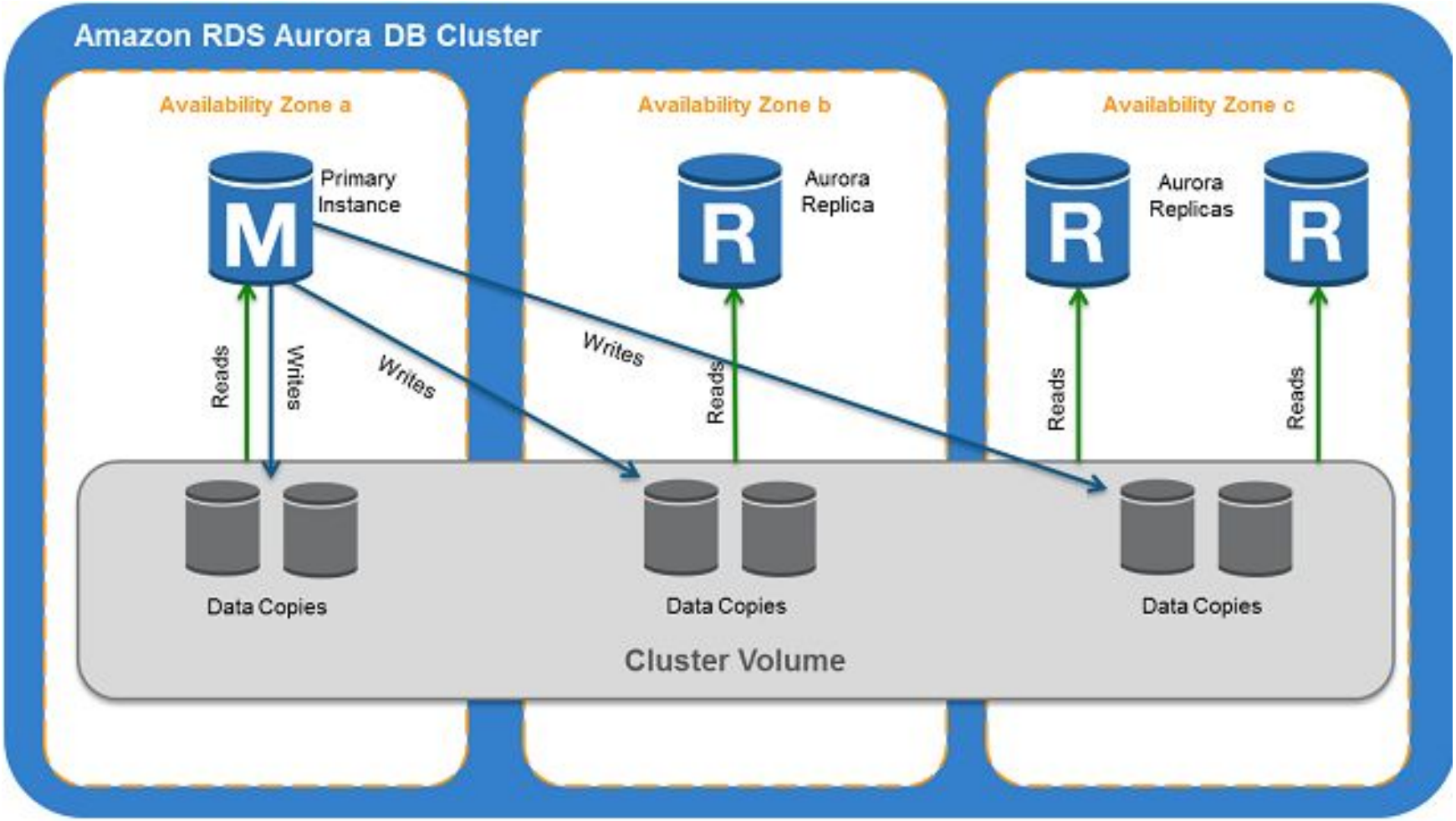
	MySQL	Oracle	SQL Server
Read Replica	Yes	No	Not Supported
Replication from on-premise to RDS and vice versa	Yes	No	Not supported

Aurora

What is Aurora DB ?

- Fully managed MySQL & PostgreSQL compatible Relational Database Engine
- Provides 5 times throughput of MySQL
- Provides 3 times throughput of PostgreSQL
- +RDS Benefits of managed service for
  - provisioning
  - patching
  - backup
  - recovery
  - failover detection & repair

Aurora Architecture



What is so special about Aurora DB ?

- DB Cluster
  - DB cluster consists of one or more instances
  - Primary Instance
    - supports read/write operations
- Aurora Replicas
    - Read Only
    - Can have upto 15 aurora replicas
    - Increased Scale
      - Read Replicas
    - Increased HA as replicas are also failover targets
- Cluster Volume
    - Single virtual volume
    - multiple copies of data are kept across AZ
    - automatically grows the volume upto 64 TB
    - volume determines the table size
- Storage auto-repair

Aurora Endpoints

- 3 endpoints
  - Cluster
  - Reader
  - Instance

`specified as domain name : port`

- Cluster Endpoint
  - Always points it to primary instance
  - If the primary instance fails, then the cluster endpoint will point to the new primary instance.
  - Read /Write
  - Primary Instance also has its own endpoint
    - If the primary instance is down then you need to manage the failover

`eg. mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com:3306`

Reader & Instance Endpoint

- **Reader Endpoint**
  - Connects to one of the aurora replicas
  - Provides load balancing (at connection level) for RO queries
  - Cannot use it for write operations

`mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com:3306`

- **Instance Endpoint**
  - end point for DB instance in an Aurora cluster
  - each DB instance has its own endpoint
  - provides direct control over connections to the DB cluster
    - e.g. for load balancing by read workload (based on queries) instead of connections
    - If you want to load-balance queries to distribute the read workload for a cluster, you will need to manage that in your application.

Instance Endpoint

```
e.g.
mydbcluster-primary.123456789012.us-east-1.rds.amazonaw
s.com:3306
```

How the HA is provided by Aurora ?

- Consider an Aurora DB cluster that has two Aurora Replicas in different Availability Zones from its primary instance.
- By connecting to the cluster endpoint, you can send both read and write traffic to the primary instance.
- You can also connect to the endpoint for each Aurora Replica and send queries directly to those DB instances.
- **In the unlikely event that the primary instance or the Availability Zone that contains the primary instance fails, then RDS will promote one of the Aurora Replicas to be the new primary instance and update the Domain Name Service (DNS) record for the cluster endpoint to point to the new primary instance.**
- Your application will continue to send read and write traffic to your Aurora DB cluster by using the cluster endpoint with minimal interruption in service.

Security

- Control RDS Management
  - Using IAM policies
- Running in VPC
  - Security groups
- To authenticate logins and permissions
  - standard mysql techniques
  - IAM database authentication

Cluster Parameter Group

- Cluster Parameter Group are applied for all instances
- This is only for Aurora Cluster

Fault Tolerance - Failover Priority

- During failover, Amazon RDS will promote the replica with the highest priority to primary
- If there is contention between 2 or more replicas in the same priority tier, then Amazon RDS will promote the replica that is the same size as the primary instance
- Priority tier logic
  a. Tier 0 > Tier 1 > … > Tier 15

Managing Cluster

- **Storage Scaling**
  - autoscaled
  - grows in 10 GB increment upto 64 TB
  - checked on hourly basis
- **Instance Scaling**
  - change the size

- **Read Scaling**
  - Upto 15 RR
  - lag < 100 ms
- **Max Connections**
  - determined by the instance size
  - controlled at `'max_connection'` parameter at instance level

Comparing Aurora with MySQL

http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Aurora.Overview.html#Aurora.Overview.CompareMySQL

End of Module

Q & A