EXPLORING AMAZON RDS MYSQL SECOND TIER READ REPLICA

Exploring Amazon RDS MySQL Second Tier Read Replica

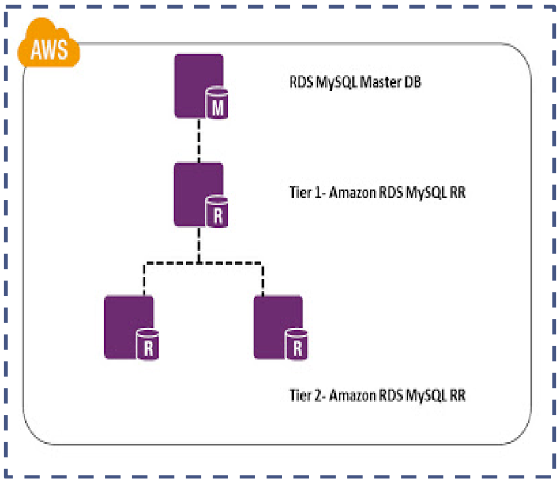
Content covered in this white paper

- Steps to configure Multi-Tiered Amazon RDS MySQL Read replicas
- Second Tier Read Replica Deployment architectures

Amazon RDS MySQL version 5.6.12 on AWS WEST region was used.

Steps to configure Multi- Tiered Amazon RDS MySQL Read Replicas

Configuration steps for the following architecture is given

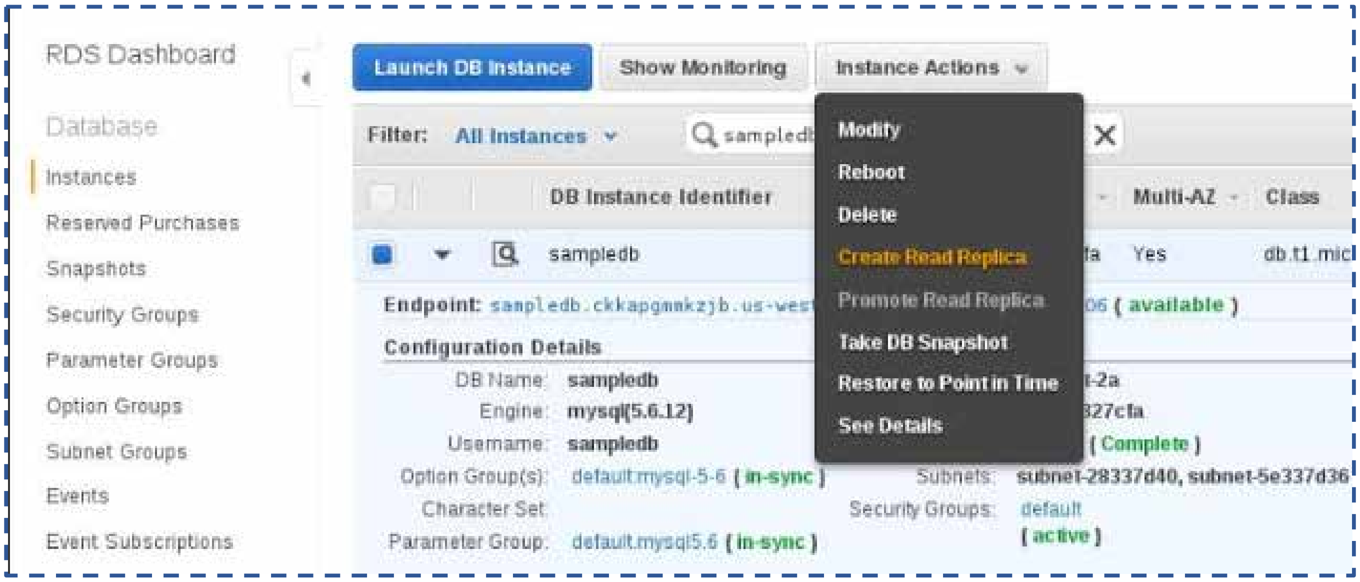


Introduction to Amazon RDS

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time consuming database administration tasks, freeing we up to focus on were applications and business. Amazon RDS provides we six familiar database engines to choose from, including Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB

Step 1: Creating Read Replica from Master and Place at Tier 1:

To create RDS MySQL Read replica navigate to the dashboard of Amazon RDS, select the Amazon RDS MySQL Master(named"sampledb") and use the option of "Create Read Replica".



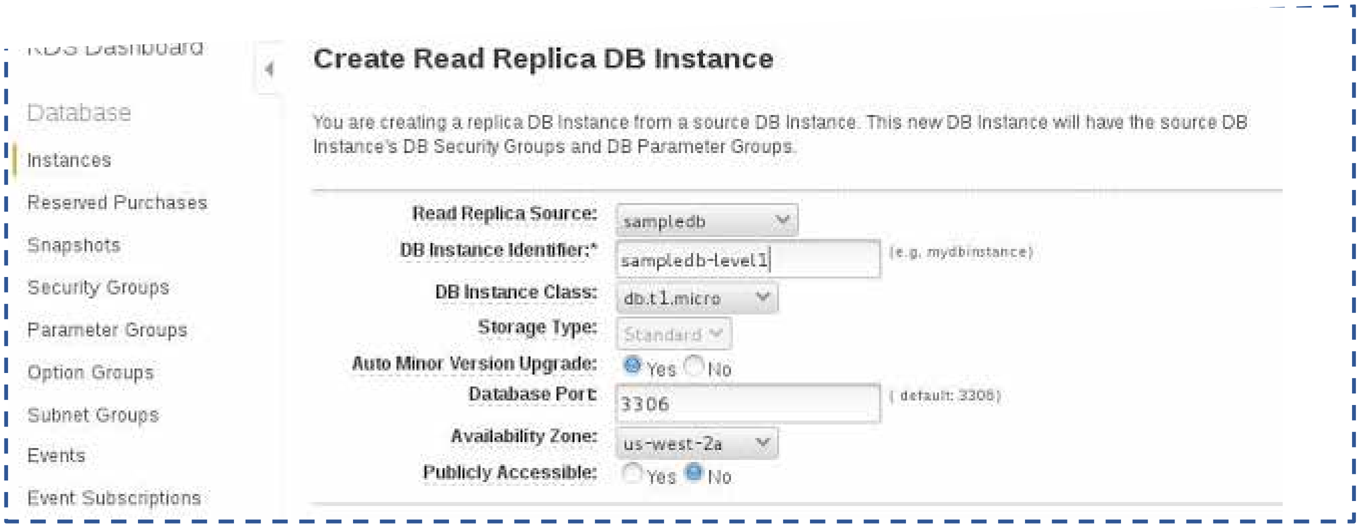
sampledb Yes db.t1.mic

Endpoint: sampledb.ckkapgmmkzjb.us-west ...06 (available)

Configuration Details

DB Name: sampledb
Engine: mysql(5.6.12)
Username: sampledb
Option Group(s): default:mysql-5-6 (in-sync)
Character Set:
Parameter Group: default.mysql5.6 (in-sync)
Subnets: subnet-28337d40, subnet-5e337d36
Security Groups: default (active)

Name the newly created Amazon RDS Read replica as "**sampledb-level1**" and place it in Tier 1. The Tier 1 Amazon RDS MySQL read Replica can be created in same AZ of Master or in a different AZ for High Availability. When the Tier replica is placed in Different AZ , we should factor few extra milliseconds of latency during replication.

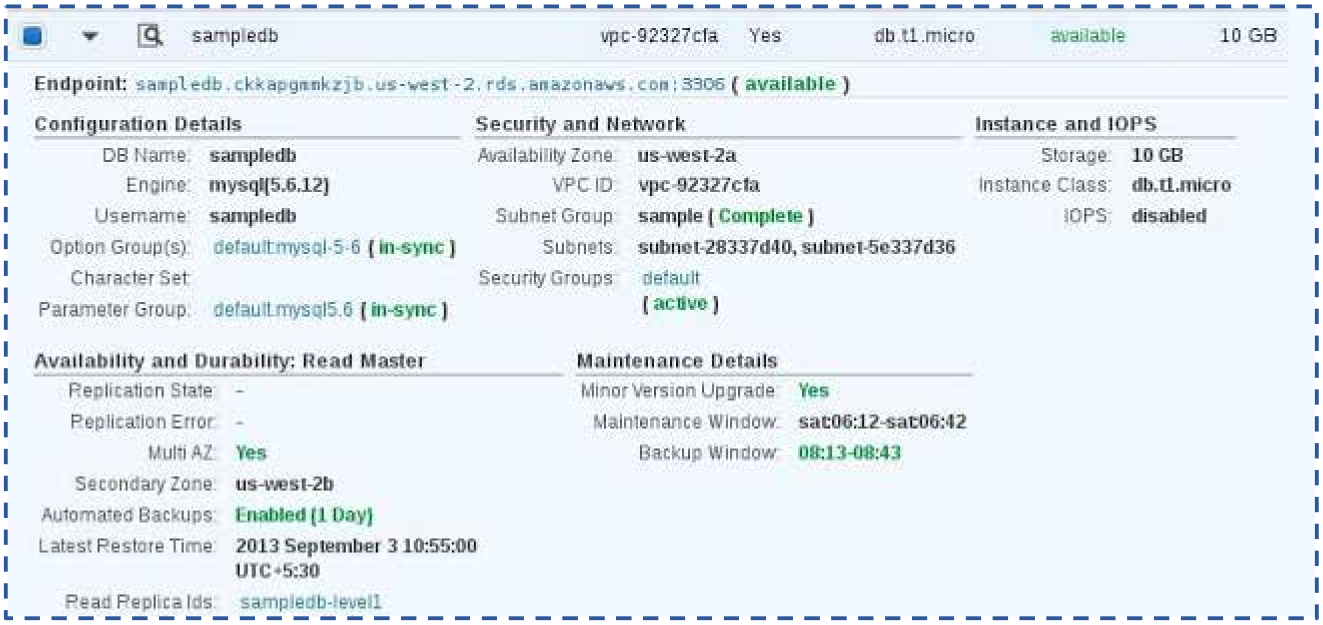


Create Read Replica DB Instance

You are creating a replica DB Instance from a source DB Instance. This new DB Instance will have the source DB Instance's DB Security Groups and DB Parameter Groups.

Read Replica Source:	sampledb
DB Instance Identifier:*	sampledb-level1 (e.g. mydbinstance)
DB Instance Class:	db.t1.micro
Storage Type:	Standard
Auto Minor Version Upgrade:	[✓ Yes] No
Database Port:	3306 (default: 3306)
Availability Zone:	us-west-2a
Publicly Accessible:	Yes [✓ No]

Explore the status from Master DB: Post successful creation of the Tier 1 Read Replica, We can see the Read Replica Id's from exploring the details of the Master DB. Illustrated in Below Screen shot :



sampledb vpc-92327cfa Yes db.t1.micro available 10 GB

Endpoint: sampledb.ckkapgmmkzjb.us-west-2.rds.amazonaws.com:3306 (available)

Configuration Details

DB Name:	sampledb
Engine:	mysql(5.6.12)
Username:	sampledb
Option Group(s):	default:mysql-5-6 (in-sync)
Character Set:	
Parameter Group:	default.mysql5.6 (in-sync)

Security and Network

Availability Zone:	us-west-2a
VPC ID:	vpc-92327cfa
Subnet Group:	sample (Complete)
Subnets:	subnet-28337d40, subnet-5e337d36
Security Groups:	default (active)

Instance and IOPS

Storage:	10 GB
Instance Class:	db.t1.micro
IOPS:	disabled

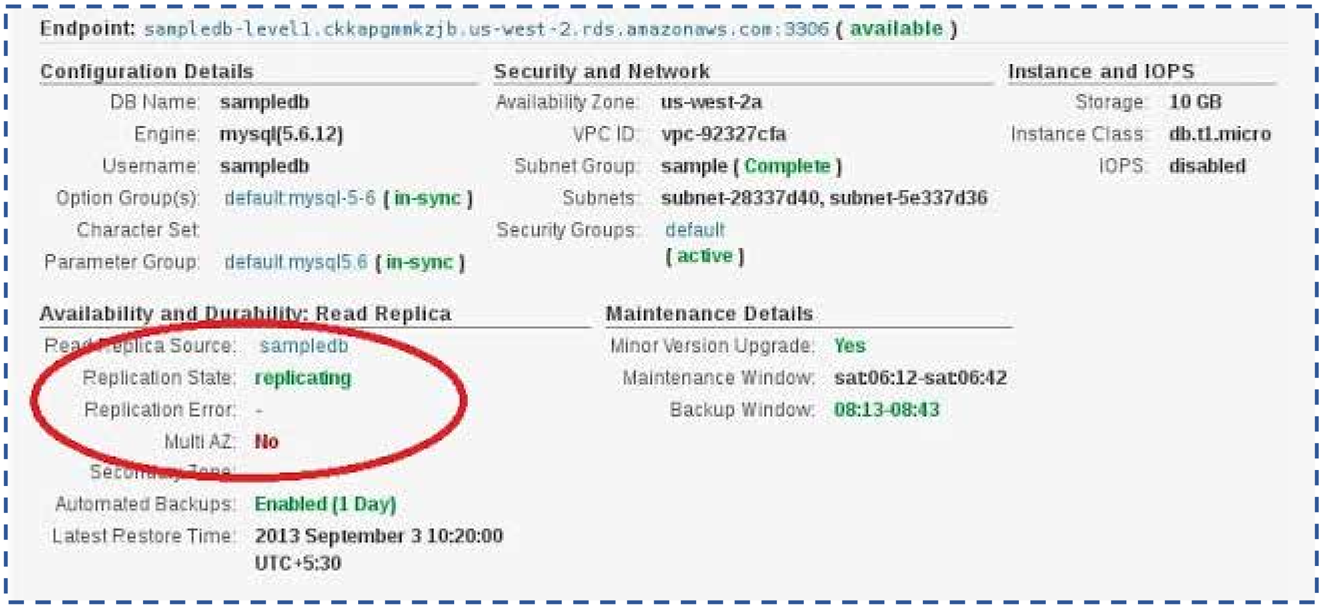
Availability and Durability: Read Master

Replication State:	-
Replication Error:	-
Multi AZ:	Yes
Secondary Zone:	us-west-2b
Automated Backups:	Enabled (1 Day)
Latest Restore Time:	2013 September 3 10:55:00 UTC+5:30
Read Replica Ids:	sampledb-level1

Maintenance Details

Minor Version Upgrade:	Yes
Maintenance Window:	sat:06:12-sat:06:42
Backup Window:	08:13-08:43

Explore the status from Tier 1 Read Replica : When we explore the Tier 1 Read Replica details, we will find it is pointing to the to Master DB. Illustrated in Below Screen Shot :



Endpoint: sampledb-level1.ckkapgmmkzjb.us-west-2.rds.amazonaws.com:3306 (available)

Configuration Details

DB Name:	sampledb
Engine:	mysql(5.6.12)
Username:	sampledb
Option Group(s):	default:mysql-5-6 (in-sync)
Character Set:	
Parameter Group:	default.mysql5.6 (in-sync)

Security and Network

Availability Zone:	us-west-2a
VPC ID:	vpc-92327cfa
Subnet Group:	sample (Complete)
Subnets:	subnet-28337d40, subnet-5e337d36
Security Groups:	default (active)

Instance and IOPS

Storage:	10 GB
Instance Class:	db.t1.micro
IOPS:	disabled

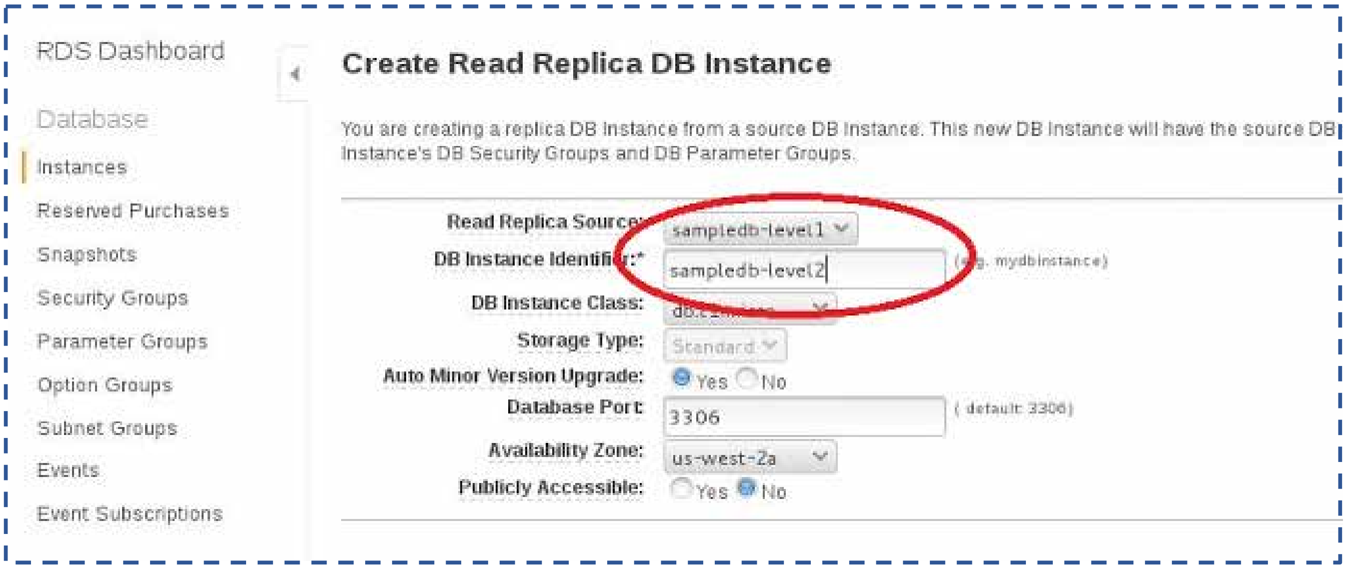
Availability and Durability: Read Replica

Read Replica Source:	sampledb
Replication State:	replicating
Replication Error:	-
Multi AZ:	No
Secondary Zone:	
Automated Backups:	Enabled (1 Day)
Latest Restore Time:	2013 September 3 10:20:00 UTC+5:30

Maintenance Details

Minor Version Upgrade:	Yes
Maintenance Window:	sat:06:12-sat:06:42
Backup Window:	08:13-08:43

Step 2: Creating Second Tier Read Replica from Tier 1 Read replica : To create Second Tier Read replica navigate to the dashboard of Amazon RDS, select the Amazon RDS Tier 1 Read Replica (**sampledb-leve1**) as the **source** and use the option of "Create Read Replica".



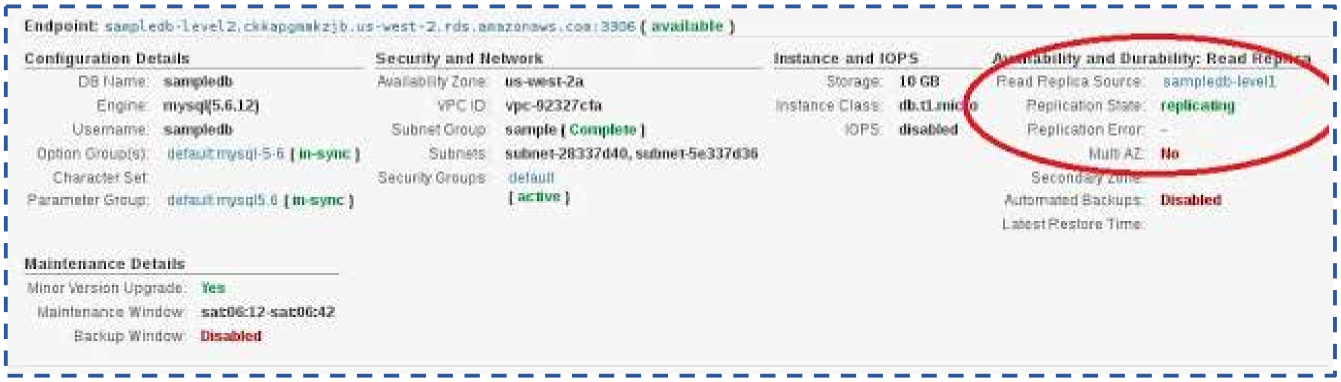
Create Read Replica DB Instance

You are creating a replica DB Instance from a source DB Instance. This new DB Instance will have the source DB Instance's DB Security Groups and DB Parameter Groups.

Read Replica Source:	sampledb-level1	
DB Instance Identifier:*	sampledb-level2	(e.g. mydbinstance)
DB Instance Class:	db.t1.micro	
Storage Type:	Standard	
Auto Minor Version Upgrade:	[✓ Yes] No	
Database Port:	3306	(default: 3306)
Availability Zone:	us-west-2a	
Publicly Accessible:	Yes [✓ No]	

Name the newly created Amazon RDS Read replica as "**sampledb-level2**" and place it in Second Tier 2. The Second Tier -2 Amazon RDS MySQL read Replica can be created in same AZ of Tier 1/Master or in a different AZ for High Availability.

Explore the status from Tier 2 Read Replica : When we explore the Tier 2 Read Replica details, we will find it is pointing to the to Tier 1 - sampledb-level1 as replication source. Illustrated in Below Screen Shot :



Endpoint: sampledb-level2.ckkapgmakzjb.us-west-2.rds.amazonaws.com:3306 (available)

Configuration Details

DB Name:	sampledb
Engine:	mysql(5.6.12)
Username:	sampledb
Option Group(s):	default:mysql-5-6 (in-sync)
Character Set:	
Parameter Group:	default.mysql5.6 (in-sync)

Security and Network

Availability Zone:	us-west-2a
VPC ID:	vpc-92327cfa
Subnet Group:	sample (Complete)
Subnets:	subnet-28337d40, subnet-5e337d36
Security Groups:	default (active)

Instance and IOPS

Storage:	10 GB
Instance Class:	db.t1.micro
IOPS:	disabled

Availability and Durability: Read Replica

Read Replica Source:	sampledb-level1
Replication State:	replicating
Replication Error:	-
Multi AZ:	No
Secondary Zone:	
Automated Backups:	Disabled
Latest Restore Time:	

Maintenance Details

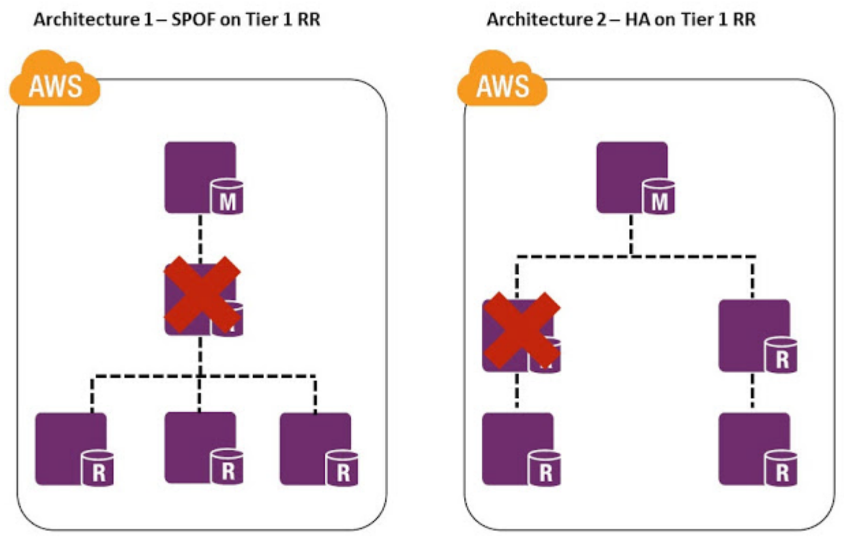
Minor Version Upgrade:	Yes
Maintenance Window:	sat:06:12-sat:06:42
Backup Window:	Disabled

To Load Balance AWS RDS Read Replica's Refer this article : http://harish11g.blogspot.com/2013/08/Load-balancing-Amazon-RDSMySQL-read-replica-slaves-using-HAProxy.html

Second Tier Read Replica Deployment Architectures

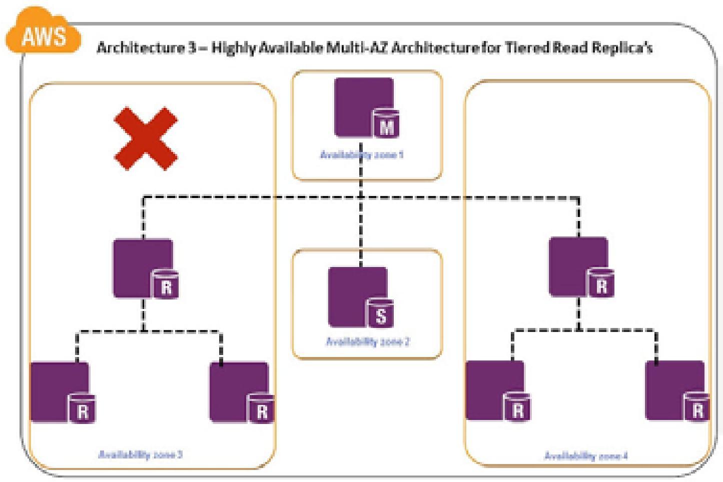
One of the main complexity behind Multi-Level replication is that, if Tier 1 Read Replica if not properly architected/placed, it can be a single point of failure. Imagine a case where we need 4

Read Replicas for were Master DB. We can take following approaches as illustrated below while designing were infrastructure for this requirement.


Architecture 1 – SPOF on Tier 1 RR
Architecture 2 – HA on Tier 1 RR
AWS
AWS
M
R
R
R
M
R
R
R

In Architecture-1, Tier-1 Read replica is a Single Point of Failure. Instead if we split Tier-1 itself into two separate fleets it offers better availability than architecture-1. Since both Tier-1 RR put replication load on Master DB , we can by pass this using Hot Standby instance. Benefits of this approach is explained below in the best practice architecture.

Best Practice High Availability Architecture for Second Tier Read Replica:



Condition 1 - Master DB Failure : Hot Standby becomes new master and Tier-1 Read Replicas points automatically to new source

Condition 2- Tier-1 Read Replica Failure : Entire Tier-1 and Associated Tier-2 has to be recreated. The Alternate Active Fleet of Tier-1+Tier-2 will serve the requests for high availability.

Condition 3 - Tier-2 Read Replica Failure : Only the non performing Tier-2 Read Replica instance has to be recreated.

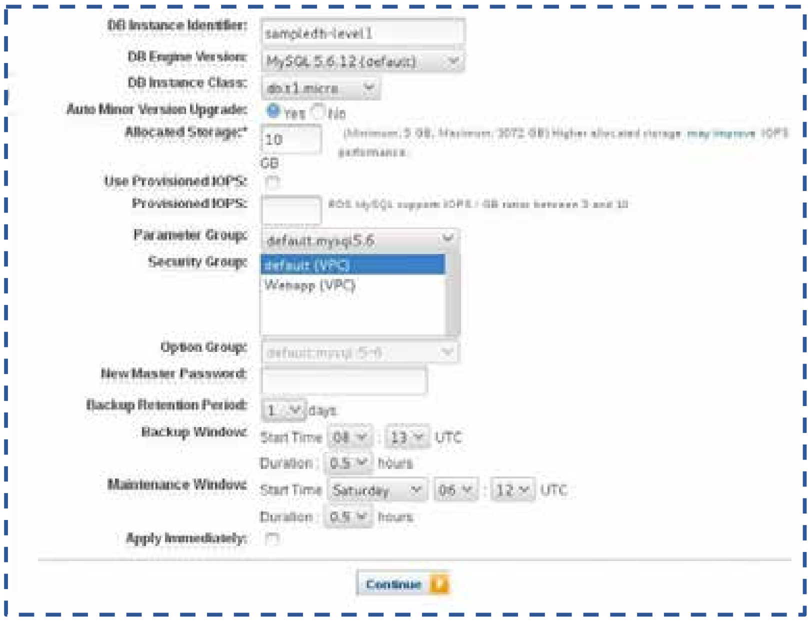
***Condition 4- AZ NW problem*:** In event AZ-3 is failed, Requests are served by alternate fleet of Tier-1+Tier-2 in another AZ(AZ-4).

Multi-AZ Hot Standby Implementation is a recommended best practice when it comes to Multi-tiered Read replica implementation

- It is recommended to create Tier 1 Read Replicas from Multi-AZ DB instance to offload read queries from the source master DB instance for high traffic sites. If the source instance of a Multi-AZ deployment fails over to the secondary, any associated read replicas will be switched to use the secondary as their replication source automatically. This model guarantees high availability.
- Also when we initiate the creation of a Tier 1 read replica, Amazon RDS takes the DB Snapshot of were Standby DB instance (instead of Source DB) and begins replication. This model saves I/O suspension on were source DB during the snapshot process **Other Points to Note:**

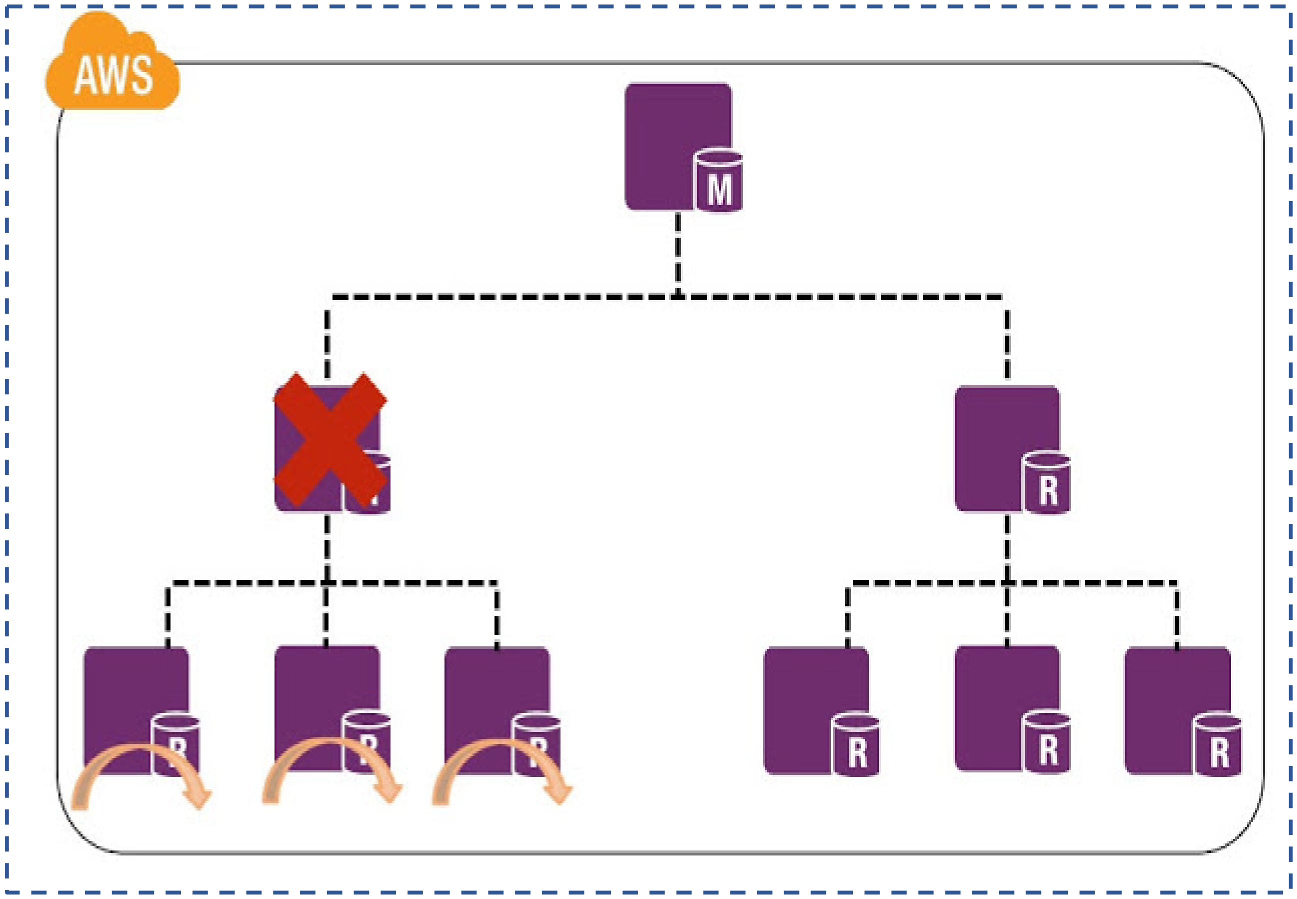
Note :
To Load Balance AWS RDS Read Replica's Refer this article
: **http://harish11g.blogspot.com/2013/08/Load-balancing-Amazon-RDS-MySQL-read-replica-slaves-using-HAProxy.html**

P1) Circular replication are not allowed in this tiered replica creation process.
P2) Third Tier cannot be created. Actually in production, very rarely we need third tier and it is not important feature
P3) When the "X" Tier replica is placed in Different AZ , we should factor few extra milliseconds of latency during replication.
P4) Before a DB instance (Master or Tier 1) can serve as a replication source, we must enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0. This requirement does not apply to second tier replica as they are not source DB instance for another read replica.So create "Read Replica" option will appear only when the Backup Retention is set to minimum of 1 day for any level Read Replica creation. When we "create Read Replica" by default RR is created with Backup Retention set to 0. Use the modify option Illustrated in below screenshot and set the Backup Retention Period. :



DB Instance Identifier:	sampledb-level1
DB Engine Version:	MySQL 5.6.12 (default)
DB Instance Class:	db.t1.micro
Auto Minor Version Upgrade:	[✓ Yes] No
Allocated Storage:*	10 GB (Minimum: 5 GB, Maximum: 3072 GB) Higher allocated storage may improve IOPS performance.
Use Provisioned IOPS:	☐
Provisioned IOPS:	RDS MySQL supports IOPS / GB ratios between 3 and 10
Parameter Group:	default.mysql5.6
Security Group:	[✓ default (VPC)] Webapp (VPC)
Option Group:	default:mysql-5-6
New Master Password:	
Backup Retention Period:	1 days
Backup Window:	Start Time 08 : 13 UTC Duration : 0.5 hours
Maintenance Window:	Start Time Saturday 06 : 12 UTC Duration : 0.5 hours
Apply Immediately:	☐

P5) We have used t1.micro as RDS DB for article explanatory purpose. For production use cases please use proper instance types after proper capacity planning.
P6) AWS has released Parallel replica creation process, where we can create multiple Tier 1 and Tier 2 replicas in parallel. Since we no longer need to wait for one replica creation before starting the next one, it becomes easy to create multiple RR quickly. Without this feature, it would take hours to create a large Multi-tiered Replica setup.
P7) If a replica lags too far behind for were environment, the normal practice is to scale up or consider deleting and recreating the read replica. Imagine we have architecture where there are two Tier 1 and three Second Tier RR as illustrated in below architecture:



Now since Tier -1 RR is lagging , we are planning to delete and recreate the same. When we delete the Tier-1 RR, All the Second Tier Read Replica's will now become Standalone , Single AZ DB's. Once we have recreated the Tier -1 RR again and managed to retain the original end point as well, we cannot re-point these Second Tier RR to Tier -1 again. We need to recreate all 3 second tier from the Tier -1 again. For a application with heavy DB dependency and read traffic, it means 1/2 of the fleet is now down. This can lead to uncomfortable performance situation. This is not a ideal condition and AWS RDS team can take this in their Road Map. For such cases, it is better to create a new Tier -1 and Second tier fleet first , update were LB/App configs and then delete the old lagging fleet

P8)Why it is better to have Multi-tiered Read Replica as the last resort in were architecture?

8.a) Currently 30 Read Replica can be created overall for a master in two tiers. 30 Read replica is more than sufficient and usually turns out to be costly architecture approach. Use this approach only for cases, which demand heavy read and when application code cannot accommodate changes and are highly DB dependent.

8.b) It is recommended in DB world to stay away from Multi-level replication as much possible. Were architecture will be much simpler with one master and "X" replica slaves, rather than having tiered replica's. As we observed in above deployment architectures, the second tier replica slave will be a trouble to manage in event of replication delay, crashes and network problems affecting the Tier-1 RR or the Master DB.

8.c) In case were application code can be redesigned, it is recommended to take following approaches before resorting to Second Tier replicas architecture

- Functional partition the RDS MySQL with Hot Stand By and Read Replica's
- Re-balance the DB load by using alternate data stores provided by AWS like Dynamo DB, ElastiCache, Cloud Search etc.

If the above methods does not work for we, take the Second tier Read replica approach.

About the Author- Harish Ganesan, is the Chief Technology Officer (CTO) and Co-Founder of SecureKloud. Harish Ganesan has more than 16+ years of experience in architecting and developing cloud computing, e-commerce and mobile application systems. He has also built large Internet banking solutions that catered to the needs of millions of users, where security and authentication were critical factors. He is responsible for the overall technology direction of the SecureKloud products and services in Cloud, Big Data and Mobility Space.