

# Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

*July 2017*



© 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

# Contents

Introduction	1
Solution Overview	1
ClassicLink and EC2-Classic	2
RDS Read Replicas	2
RDS Snapshots	2
Migration Topology	3
Migration Steps	5
Step 1: Enable ClassicLink for the Target VPC	6
Step 2: Set up a Proxy Server on an EC2-Classic Instance	6
Step 3: Use ClassicLink Between the Proxy Server and Target VPC	7
Step 4: Configure the DB Instance (EC2-Classic)	8
Step 5: Create a User on the DB Instance (EC2-Classic)	9
Step 6: Create a Temporary Read Replica (EC2-Classic)	9
Step 7: Enable Backups on the Read Replica (EC2-Classic)	10
Step 8: Stop Replication on Read Replica (EC2-Classic)	11
Step 9: Create Snapshot from the Read Replica (EC2-Classic)	12
Step 10: Share the Snapshot (Optional)	13
Step 11: Restore the Snapshot in the Target VPC	15
Step 12: Enable Backups on VPC RDS DB Instance	17
Step 13: Set up Replication Between VPC and EC2-Classic DB Instances	18
Step 14: Switch to the VPC RDS DB Instance	19
Step 15: Take a Snapshot of the VPC RDS DB Instance	20
Step 16: Change the VPC DB Instance to be ‘Privately’ Accessible (Optional)	20
Step 17: Move the VPC DB Instance into Private Subnets (Optional)	21
Alternative Approaches	22
AWS Database Migration Service (DMS)	22

Changing the VPC Subnet for a DB Instance	23
Conclusion	24
Contributors	24
Further Reading	25
Appendix A: Set Up Proxy Server in Classic	25

# Abstract

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

If your Amazon Web Services (AWS) account was created before 2013, chances are you might be running Amazon RDS MySQL in an Amazon Elastic Compute Cloud (EC2)-Classic environment, and you are looking to migrate Amazon RDS into an Amazon EC2-Amazon Virtual Private Cloud (VPC) environment. This whitepaper outlines the requirements and detailed steps needed to migrate Amazon RDS MySQL databases from EC2-Classic to EC2-VPC with minimal downtime using RDS MySQL Read Replicas and ClassicLink.

## Introduction

There are two Amazon Elastic Compute Cloud (EC2) platforms that host Amazon Relational Database Service (RDS) database (DB) instances, *EC2-VPC* and *EC2-Classic*. On the EC2-Classic platform, your instances run in a single, flat network that you share with other customers. On the EC2-VPC platform, your instances run in a virtual private cloud (VPC) that's logically isolated to your AWS account. This logical network isolation closely resembles a traditional network you might operate in your own data center, plus it has the benefits of the AWS scalable infrastructure.

If you're running RDS DB instances in an EC2-Classic environment, you might be considering migrating your databases to Amazon VPC to take advantage of its features and capabilities. However, migrating databases across environments can involve complex backup and restore operations with longer downtimes that you might not be able to tolerate in your production environment. This whitepaper focuses on how to use RDS read replica and snapshot capabilities to migrate an RDS MySQL DB instance in EC2-Classic to a VPC over ClassicLink. By leveraging RDS MySQL replication with ClassicLink you can migrate your databases easily and securely with minimal downtime. Alternative methods are also discussed.

## Solution Overview

This solution uses EC2 ClassicLink to enable an RDS DB instance in EC2-Classic (that is, outside a VPC) to communicate to a VPC. First, a read replica of the DB instance in EC2-Classic is created. Then, a snapshot of the read replica (called the source DB instance) is taken and used to set up a read replica in the VPC. A ClassicLink proxy server enables communication between the source DB instance in EC2-Classic and the target read replica in the VPC. Once the target read replica in the VPC has caught up with the source DB instance in EC2-Classic, updates against the source are stopped and the target read replica is promoted. At this point, the connection details in any application that is reading or writing to the database are updated. The source database remains fully operational during the migration, minimizing downtime to applications.

Each of these components is explained in further detail as follows.

## ClassicLink and EC2-Classic

EC2 ClassicLink allows you to connect EC2-Classic instances to a VPC within the same AWS Region. This allows you to associate VPC security groups with the EC2-Classic instances, enabling communication between EC2-Classic instances and VPC instances using private IP addresses. The association between VPC security groups and the EC2-Classic instance removes the need to use public IP addresses or Elastic IP addresses to enable communication between these platforms.

ClassicLink is available to all users with accounts that support the EC2-Classic platform and can be used with any EC2-Classic instance. Using ClassicLink and private IP address space for migration ensures all communication and data migration happens within the AWS network without requiring a public IP address for your DB instance or an Internet Gateway (IGW) to be set up for the VPC.

## RDS Read Replicas

You can create one or more read replicas of a given source RDS MySQL DB instance and serve high-volume application read traffic from multiple copies of your data. Amazon RDS uses the MySQL engine's native asynchronous replication to update the read replica whenever there is a change to the source DB instance. The read replica operates as a DB instance that allows only read-only connections; applications can connect to a read replica just as they would to any DB instance. Amazon RDS replicates all databases in the source DB instance. Read replicas can also be promoted, so that they become standalone DB instances.

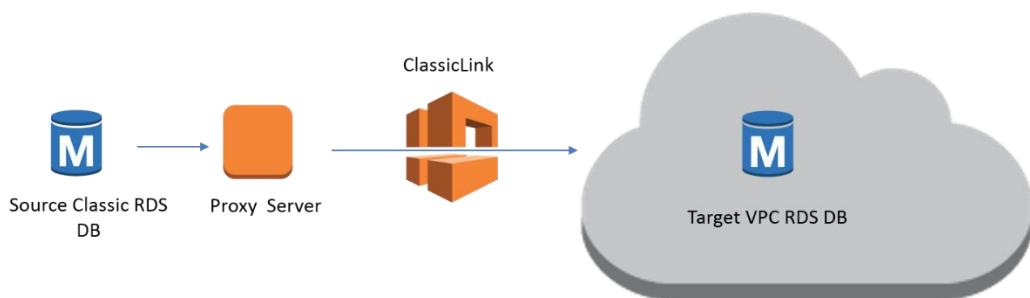
## RDS Snapshots

The ClassicLink solution relies on Amazon RDS snapshots to initially create the target MySQL DB instance in your VPC. Amazon RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. When you create a DB snapshot, you need to identify which DB instance you are going to back up, and then give your DB snapshot a name so you can restore from it later. Creating this DB snapshot on a single-Availability Zone (AZ) DB instance results in a brief I/O suspension that typically lasts no more than a few minutes. Multi-AZ DB instances are not

affected by this I/O suspension since the backup is taken on the standby instance.

## Migration Topology

ClassicLink allows you to link your EC2-Classic DB instance to a VPC in your account within the same Region. After you've linked an EC2-Classic DB instance, it can communicate with instances in your VPC using their private IP addresses. However, instances in the VPC cannot directly access the AWS services provisioned by the EC2-Classic platform using ClassicLink. So to migrate an RDS database from EC2-Classic to VPC you must set up a proxy server. The proxy server uses ClassicLink to link the source DB instance in EC2-Classic to the VPC. Port forwarding on the proxy server allows communication between the source DB instance in EC2-Classic and the target DB instance in the VPC. This topology is illustrated in Figure 1.

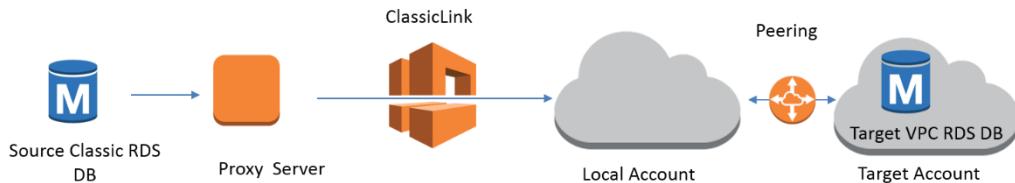


**Figure 1: Topology for migration in the same account**

If you're moving your RDS database to a different account you will need to set up a peering connection between the local VPC and the target VPC in the remote account. This topology is illustrated in Figure 2.

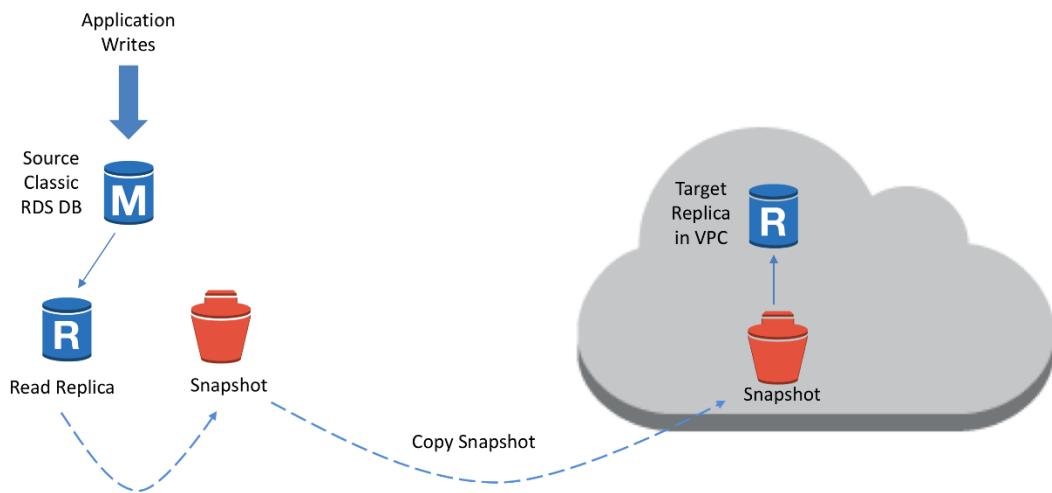
Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

---



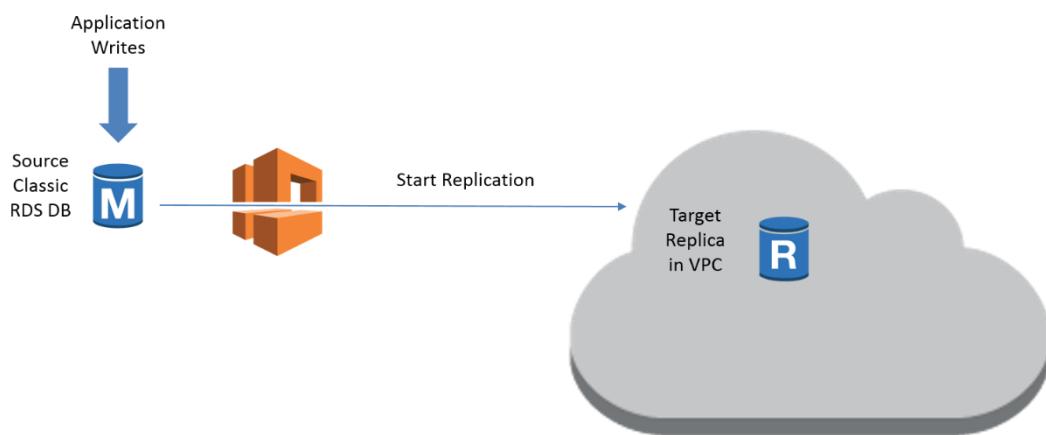
**Figure 2: Topology for migration to a different account**

Figure 3 illustrates how the snapshot of the DB instance is used to set up a read replica in the target VPC.



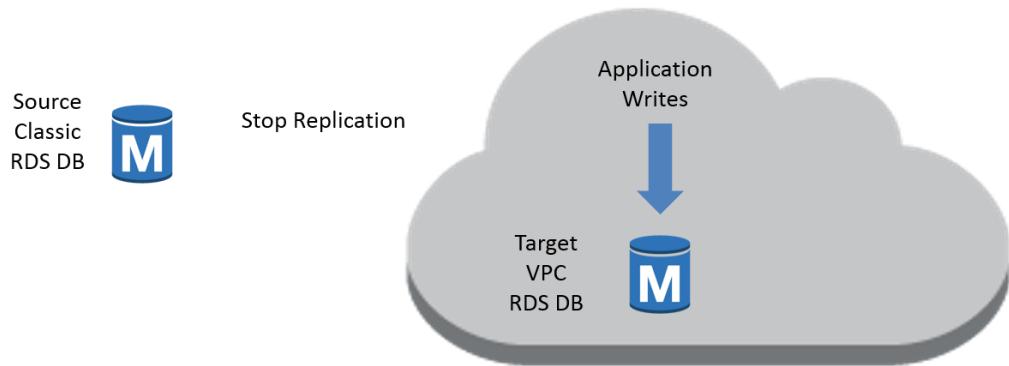
**Figure 3: Creating a read replica snapshot and restoring in VPC**

A ClassicLink proxy enables communication between the source RDS DB instance in EC2-Classic and the target VPC replica, as illustrated in Figure 4.



**Figure 4: Setting up replication between the Classic and VPC read replica**

Figure 5 illustrates how updates against the source DB instance are stopped and the VPC replica is promoted to master status.



**Figure 5: Cutting over application to the VPC RDS DB instance**

## Migration Steps

This section lists the steps you need to perform to migrate your RDS DB instance from EC2-Classic to VPC using ClassicLink.

## Step 1: Enable ClassicLink for the Target VPC

In the Amazon VPC console, from the VPC Dashboard, select the VPC for which you want to enable ClassicLink, select **Actions** in the drop-down list, and select **Enable ClassicLink**. Then choose **Yes, Enable** as shown below:

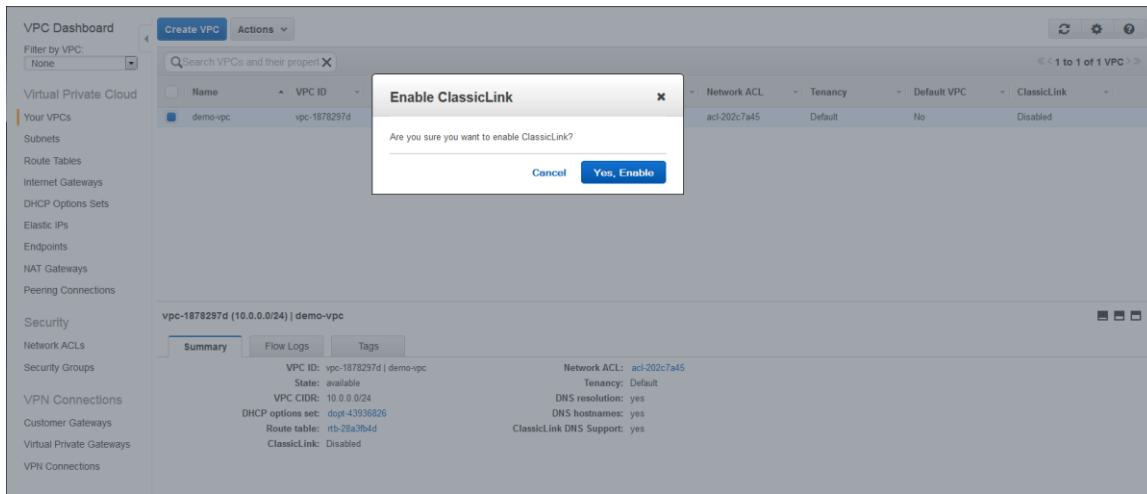


Figure 6: Enabling ClassicLink

## Step 2: Set up a Proxy Server on an EC2-Classic Instance

Install a proxy server on an EC2-Classic instance. The proxy server forwards traffic to and from the RDS instance in EC2-Classic. You can use an open-source package such as NGINX for port forwarding. For detailed information on setting up NGINX, see [Appendix A](#).

Set up appropriate security groups so the proxy server can communicate with the RDS instance in EC2-Classic. In the following example, the proxy server and the RDS instance in EC2-Classic are members of the same security group that allows traffic within the security group.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

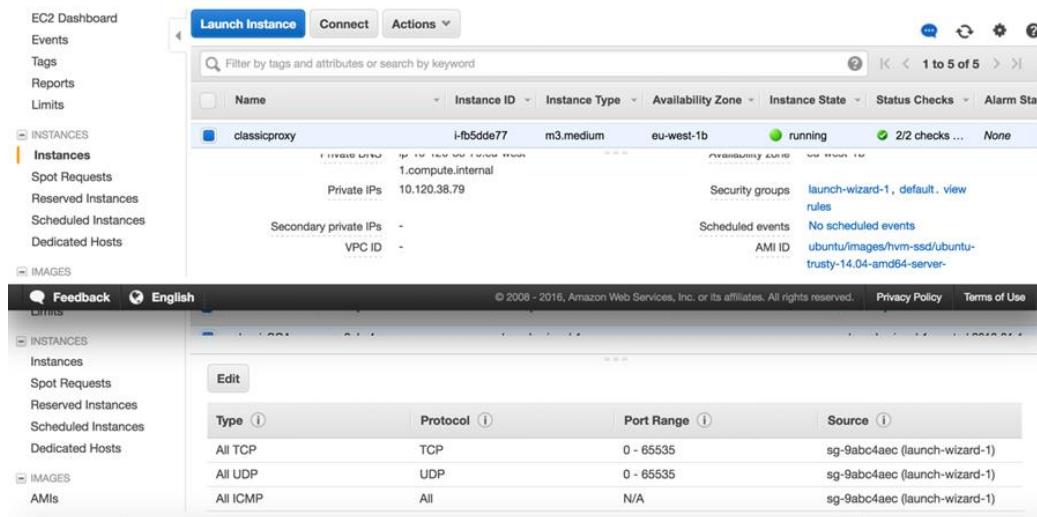


Figure 7: Security group setup

### Step 3: Use ClassicLink Between the Proxy Server and Target VPC

In the Amazon EC2 console, from the EC2 Instances Dashboard, select the EC2-Classic instance running the proxy server and choose **ClassicLink** on the Actions drop-down list to create a ClassicLink connection with the target VPC. Select the appropriate security group so that the proxy server can communicate with the RDS DB instance in your VPC. In the example in Figure 8, SG A1 is selected. Next, choose **Link to VPC**

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

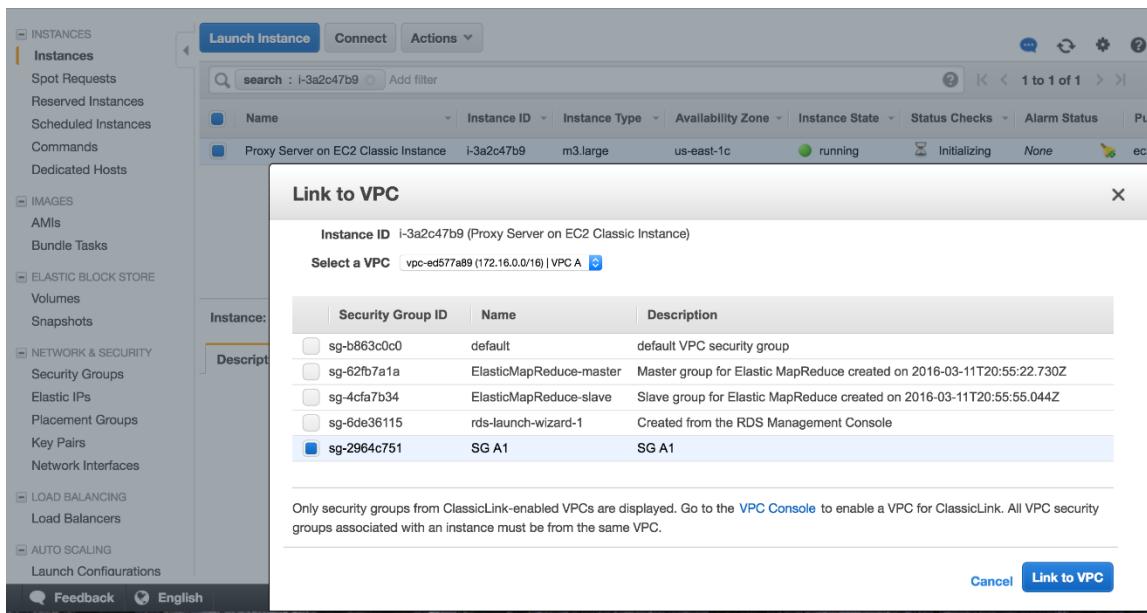


Figure 8: ClassicLink connection to VPC security group

## Step 4: Configure the DB Instance (EC2-Classic)

In the Amazon RDS console, from the RDS Dashboard, under **Parameter Groups**, select the parameter group associated with the RDS DB instance and use **Edit Parameters** to ensure the `innodb_flush_log_at_trx_commit` parameter is set to 1 (the default). This ensures ACID compliance. For more information see <http://tinyurl.com/innodb-flush-log-at-trx-commit>. This step is necessary only if the value has been changed from the default of 1.

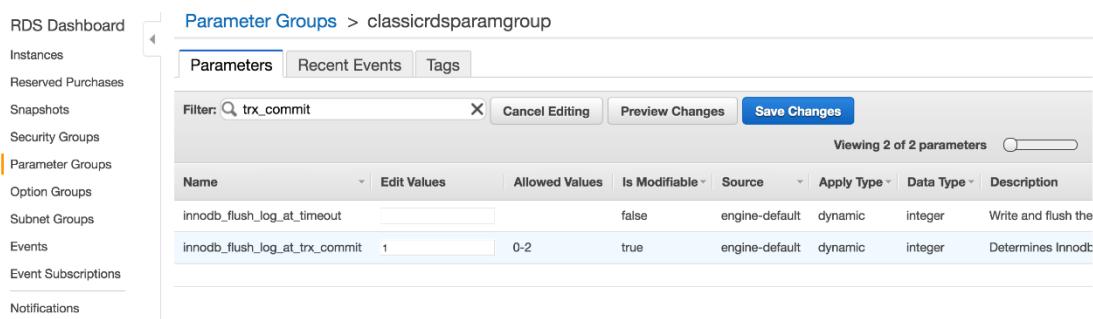


Figure 9: Parameter group values on a Classic DB instance

## Step 5: Create a User on the DB Instance (EC2-Classic)

Connect to the RDS DB instance running in EC2-Classic via `mysql` client to create a user and grant permissions to replicate data.

```
Prompt> mysql -h classicrdsinstance.123456789012.us-east-1.rds.amazonaws.com -P 3306 -u hhar -p
MySQL [(none)]> create user replicationuser identified by
'classictoVPC123';
Query OK, 0 rows affected (0.01 sec)
MySQL [(none)]> grant replication slave on *.* to
replicationuser;
Query OK, 0 rows affected (0.01 sec)
```

## Step 6: Create a Temporary Read Replica (EC2-Classic)

Use a temporary read replica to create a snapshot and ensure that you have the correct information to set up replication on the new VPC DB instance. In the Amazon RDS console, from the RDS Dashboard, under **Instances**, select the EC2-Classic DB instance, and select **Create Read Replica DB Instance**. Specify your replication instance information.

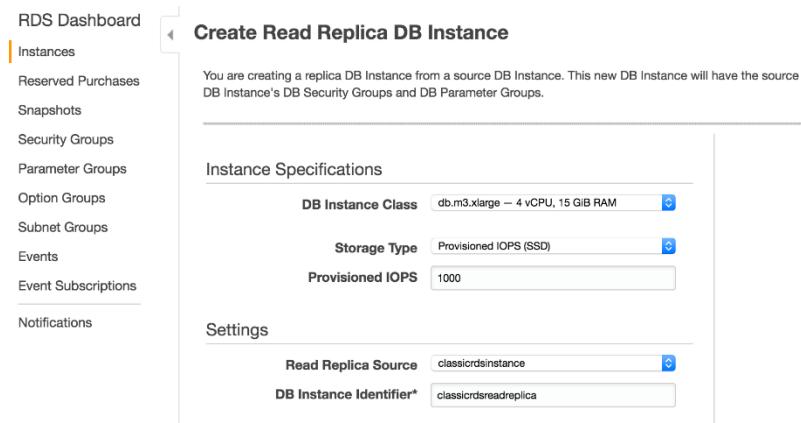


Figure 10: Classic read replica instance properties

You then need to specify the network and security properties for the replica.

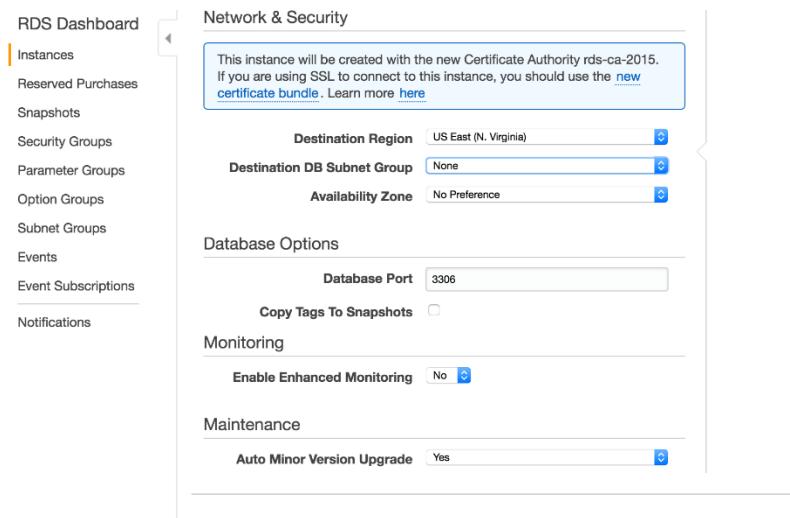


Figure 11: Classic read replica network and security properties

## Step 7: Enable Backups on the Read Replica (EC2-Classic)

From the RDS Dashboard, under **Instances**, select the Read Replica in EC2-Classic, and use **Modify DB Instances** to set the **Backup Retention Period** to a non-zero number of days. Setting this parameter to a positive number enables automated backups.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

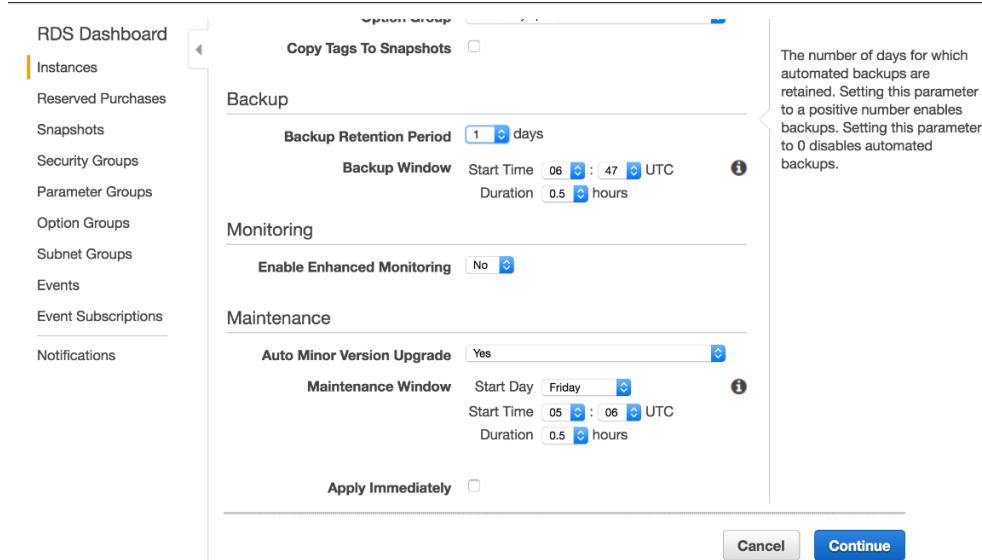


Figure 12: Enabling backups

## Step 8: Stop Replication on Read Replica (EC2-Classic)

When you are ready to switch over, connect to the RDS replica in EC2-Classic via a mysql client and issue the `mysql.rds_stop_replication` command.

```
Prompt> mysql -h classicrdsreadreplica1.chd3laahf8xl.us-east-1.rds.amazonaws.com -P 3306 -u hhar -p
MySQL [(none)]> call mysql.rds_stop_replication;
+-----+
| Message                                |
+-----+
| Slave is down or disabled               |
+-----+
1 row in set (1.02 sec)
Query OK, 0 rows affected (1.02 sec)
MySQL [(none)]>
```

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

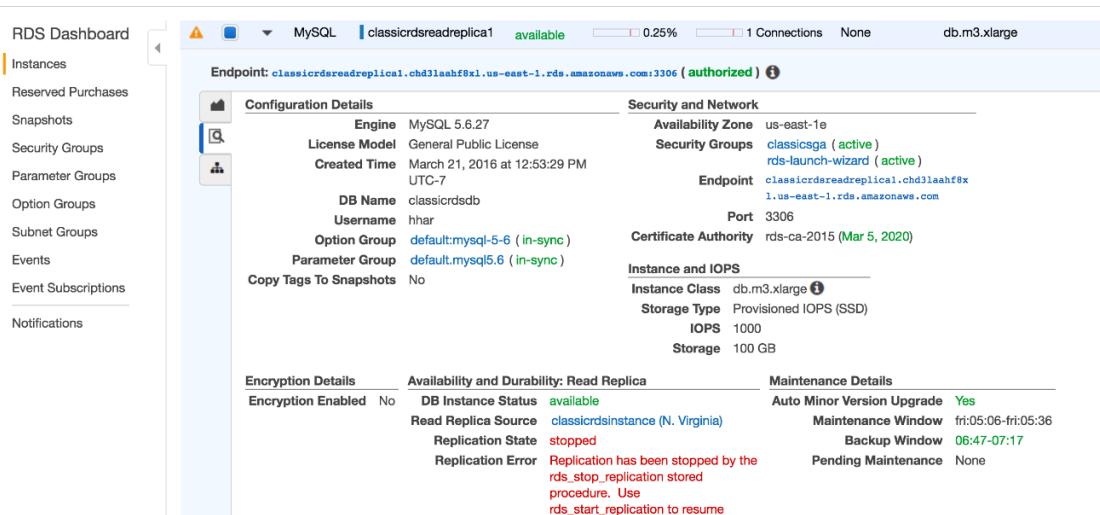


Figure13: Confirmation of replica status on the console

Using the following `show slave status` command, save the replication status data in a local file. You will need it later when setting up replication on the DB instance in VPC.

```
Prompt> mysql -h classicrdsreadreplica1.chd3laahf8x1.us-east-1.rds.amazonaws.com -P 3306 -u hhar -p -e "show slave status \G"
> readreplicastatus.txt
```

## Step 9: Create Snapshot from the Read Replica (EC2-Classic)

From the RDS Dashboard, under **Instances**, select the Read Replica that you just stopped, and use **Take Snapshot** to create a DB snapshot.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

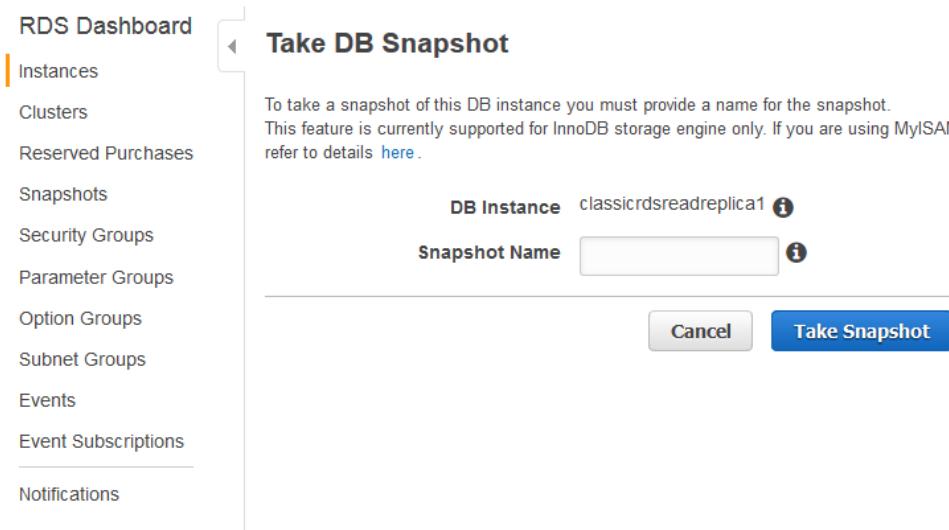


Figure 1: Taking a Snapshot of the read replica

## Step 10: Share the Snapshot (Optional)

If you are migrating across accounts, you need to share the snapshot. From the Amazon RDS console, under **Snapshots**, select the recently created read replica, and use **Share Snapshot** to make the snapshot available across accounts. This step is not required if the target VPC is in same account. After sharing the snapshot, log in to the new account after this step is finished.

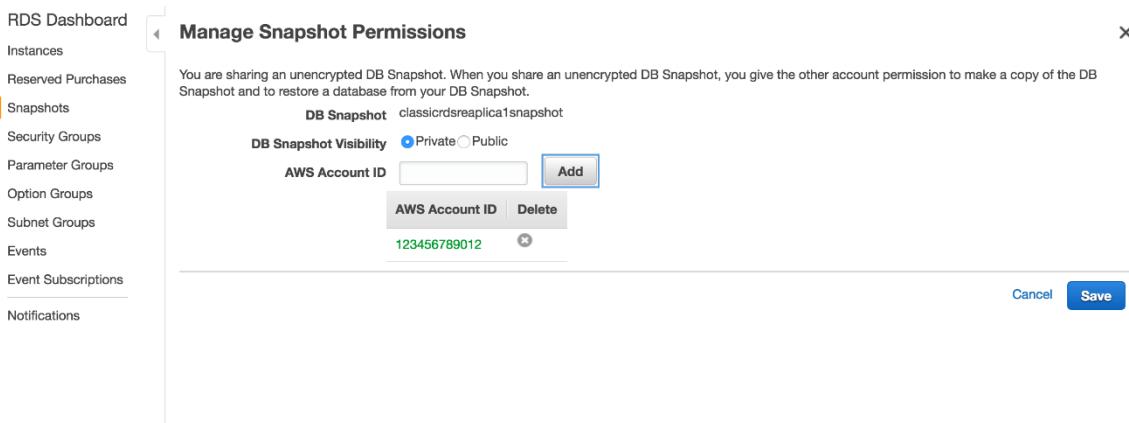


Figure 2: Sharing a snapshot between accounts

If you are migrating to a different account you need to set up a peering connection between the local VPC and the target VPC in the remote account.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

You will have to allow access to the security group that you used when you enabled the ClassicLink between the proxy server and VPC.

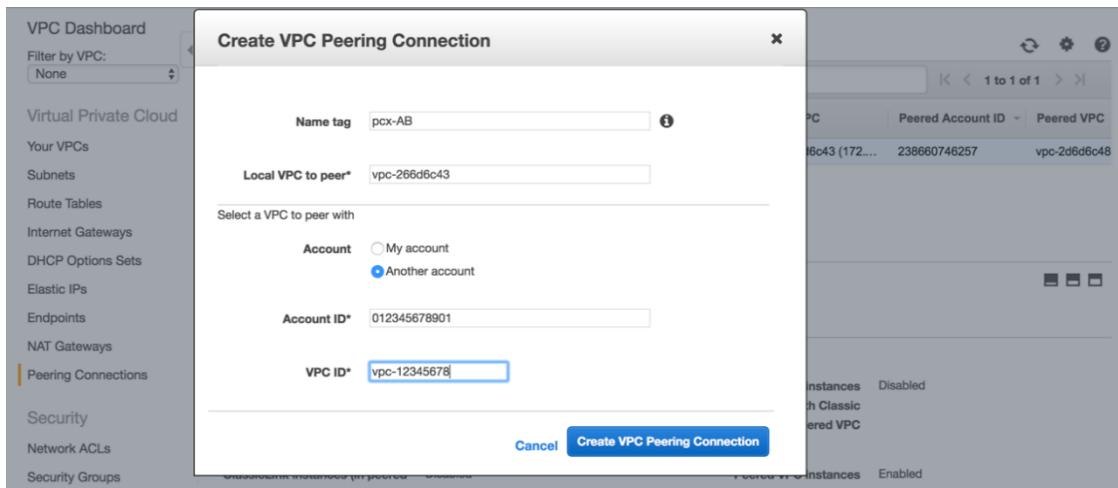


Figure 16: Creating a VPC peering connection

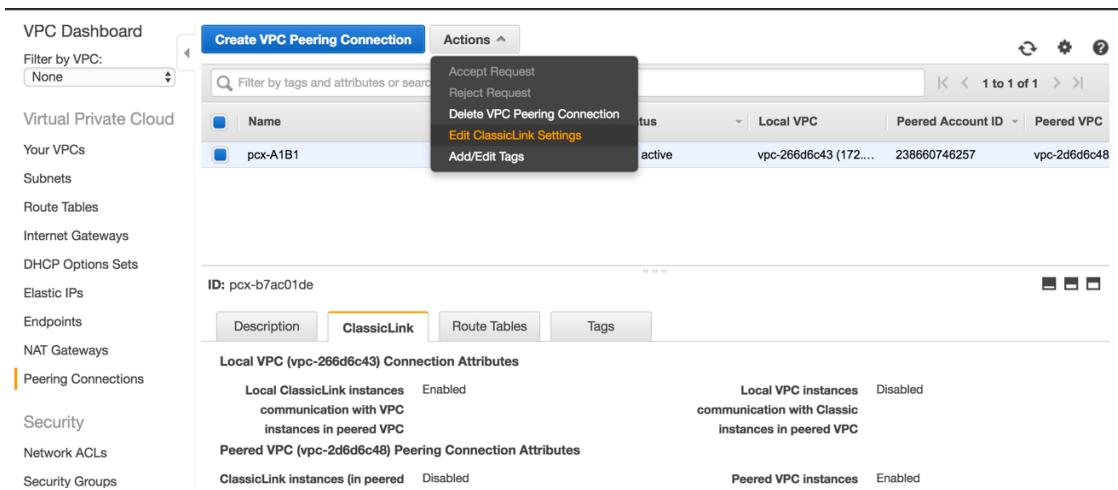


Figure 17: Enabling ClassicLink over a peering connection

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

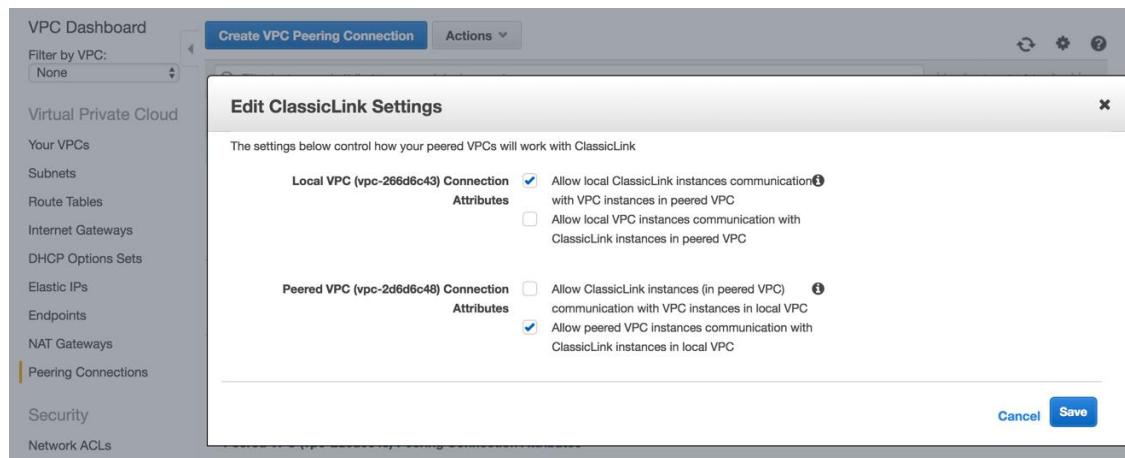


Figure 18: ClassicLink settings for peering

## Step 11: Restore the Snapshot in the Target VPC

From the Amazon RDS console, under **Snapshots**, select the Classic Read Replica, and use **Restore Snapshot** to restore the Read Replica snapshot. You should also select **Multi-AZ Deployment** at this time.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

The screenshot shows the AWS RDS Dashboard with the 'Restore DB Instance' wizard open. The left sidebar lists various RDS management options like Instances, Clusters, Reserved Purchases, Snapshots, Security Groups, Parameter Groups, Option Groups, Subnet Groups, Events, Event Subscriptions, and Notifications. The main panel has three tabs: 'Instance Specifications', 'Settings', and 'Network & Security'. Under 'Instance Specifications', the DB Engine is set to 'mysql', License Model to 'general-public-license', DB Instance Class to 'db.t2.micro — 1 vCPU, 1 GiB RAM', and Multi-AZ Deployment to 'Yes'. Under 'Settings', the DB Snapshot ID is 'rds:mysqldevelopment-2016-06-20-22-52' and the DB Instance Identifier is empty. Under 'Network & Security', the VPC is set to 'Default VPC (vpc-3bd2205)', Subnet Group to 'default', Publicly Accessible to 'Yes', and Availability Zone to 'No Preference'.

**Figure 19:** Restoring snapshot in target VPC

**Note:** We highly recommend that you enable the Multi-AZ Deployment option during initial creation of the new VPC DB instance. If you bypass this step and convert to Multi-AZ after switching your application over to the VPC DB instance, you can experience a significant performance impact, especially for write-intensive database workloads.

Under **Networking & Security**, set **Publicly Accessible** to **Yes**. Next, select the target VPC and appropriate subnet groups to ensure connectivity from the VPC RDS DB instance to the Classic Proxy Server.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

This instance will be created with the new Certificate Authority rds-ca-2015. If you are using SSL to connect to this instance, you should use the [new certificate bundle](#). Learn more [here](#).

VPC: VPC A (vpc-ed577a89)  
Subnet Group: default-vpc-ed577a89  
Publicly Accessible: Yes  
Availability Zone: No Preference

Database Options:  
Database Port: 3306  
Option Group: default:mysql-5-6  
Copy Tags To Snapshots:

Maintenance:  
Auto Minor Version Upgrade: Yes

Figure 20: Setting VPC and subnet group on VPC DB instance

RDS Dashboard

Instances

Clusters

Reserved Purchases

Snapshots

Security Groups

Parameter Groups

Option Groups

Subnet Groups

Events

Event Subscriptions

Settings

DB Instance Identifier: vpdrdsdiffaccount

New Master Password: [redacted]

Network & Security

Security Group: SG B (sg-f93f029d) (vpc-2d6d6c48)  
default (sg-353c0151) (vpc-2d6d6c48)

Certificate Authority: rds-ca-2015

Publicly Accessible:  Yes  No

Figure 3: Security group settings for cross-account migration

## Step 12: Enable Backups on VPC RDS DB Instance

By default, backups are not enabled on read replicas. From the Amazon RDS console, under **Instances**, select the VPC RDS DB instance and use **Modify DB Instances** to enable backups.



## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

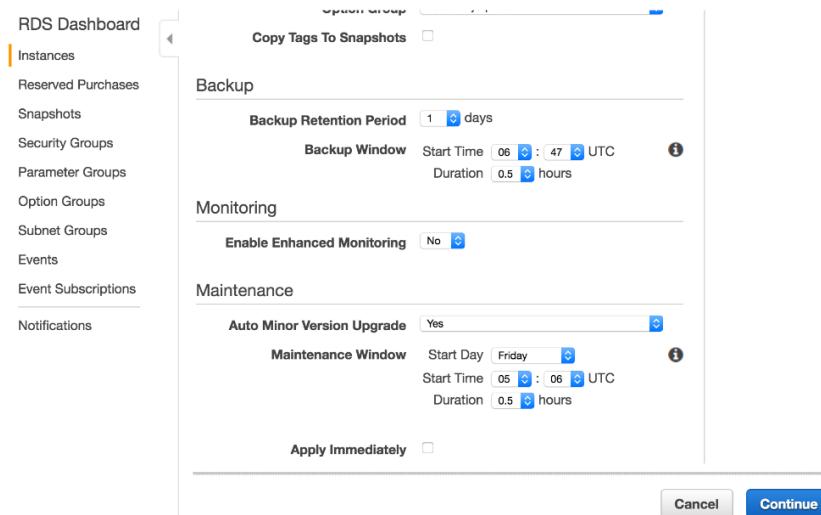


Figure 422: Setting backup retention

## Step 13: Set up Replication Between VPC and EC2-Classic DB Instances

Retrieve the log file name and log position number from information saved in the previous step.

```
Prompt> cat readreplicastatus.txt | grep Master_Log_File
      Master_Log_File: mysql-bin-changelog.001993
Prompt> cat readreplicastatus.txt | grep Exec_Master_Log_Pos
      Exec_Master_Log_Pos: 120
```

Connect to the VPC RDS DB instance via a `mysql` client through the ClassicLink proxy and set the EC2-Classic RDS DB instance as the replication master by issuing the `rds_start_replication` command. Use the private IP address of the EC2-Classic proxy server as well as the log position from the output above.

```
MySQL [(none)]> call mysql.rds_set_external_master('<private-ip-address-of-proxy', 3306, 'replicationuser', 'classictoVPC123', 'mysql-bin-changelog.001993', 120, 0);
Query OK, 0 rows affected (0.12 sec)
MySQL [(none)]> call mysql.rds_start_replication;
+-----+
| Message |
+-----+
```

```
+-----+
| Slave running normally. |
+-----+
1 row in set (1.03 sec)
Query OK, 0 rows affected (1.03 sec)
```

Verify the replication status on VPC Read Replica using the show slave status command.

```
MySQL [(none)]> show slave status \G;
```

## Step 14: Switch to the VPC RDS DB Instance

After ensuring that the data in the VPC read replica has caught up to the EC2-Classic master, configure your application to stop writing data to the RDS DB instance in EC2-Classic. After the replication lag has caught up, connect to the VPC RDS DB instance via a mysql client and issue the rds\_stop\_replication command.

```
MySQL [(none)]> call mysql.rds_stop_replication;
```

At this point, the VPC will stop replicating data from the master. You can now promote the replica by connecting to the VPC RDS DB instance via a mysql client and issuing the mysql.rds\_reset\_external\_master command.

```
MySQL [(none)]> call mysql.rds_reset_external_master;
+-----+
| Message          |
+-----+
| Slave is down or disabled |
+-----+
1 row in set (1.04 sec)
+-----+
| message          |
+-----+
| Slave has been reset |
```



```
+-----+
1 row in set (3.12 sec)
Query OK, 0 rows affected (3.12 sec)
```

You can now change the endpoint in your application to write to the VPC RDS DB instance.

## Step 15: Take a Snapshot of the VPC RDS DB Instance

From the Amazon RDS console, under **Instances**, select the VPC RDS DB instance and use **Take Snapshot** to capture a user snapshot for recovery purposes.

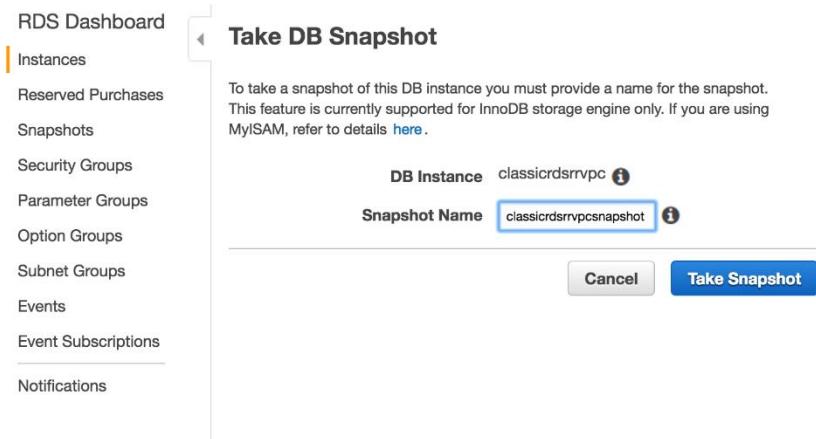


Figure 23: Taking a snapshot of the DB instance in VPC

## Step 16: Change the VPC DB Instance to be ‘Privately’ Accessible (Optional)

After the migration to the new VPC RDS DB instance is complete, you can make it be privately (not publicly) accessible. From the Amazon RDS console, under **Instances**, select the DB instance and click **Modify**. Under **Network & Security**, set **Publicly Accessible** to **No**.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas

The screenshot shows the AWS RDS Dashboard with the 'Instances' tab selected. On the left, a sidebar lists options like Instances, Reserved Purchases, Snapshots, Security Groups, Parameter Groups, Option Groups, Subnet Groups, Events, Event Subscriptions, and Notifications. The main panel displays instance configuration fields: DB Instance Class (db.m3.xlarge), Multi-AZ Deployment (Yes), Storage Type (Provisioned IOPS (SSD)), Allocated Storage\* (100 GB), Provisioned IOPS (1000). Under 'Settings', the DB Instance Identifier is set to 'classicrdsrvpc'. In the 'Network & Security' section, the Security Group dropdown is expanded to show 'ElasticMapReduce-slave (sg-4cfa7b34) (SG A1 (sg-2964c751) (vpc-ed577a89))', 'default (sg-b863c0c0) (vpc-ed577a89)', and 'rds-launch-wizard-1 (sg-6de36115) (vpc-ed577a89)'. The 'Certificate Authority' is set to 'rds-ca-2015'. The 'Publicly Accessible' checkbox is checked (indicated by a blue outline).

Figure 24: Setting instance to not be publicly accessible

## Step 17: Move the VPC DB Instance into Private Subnets (Optional)

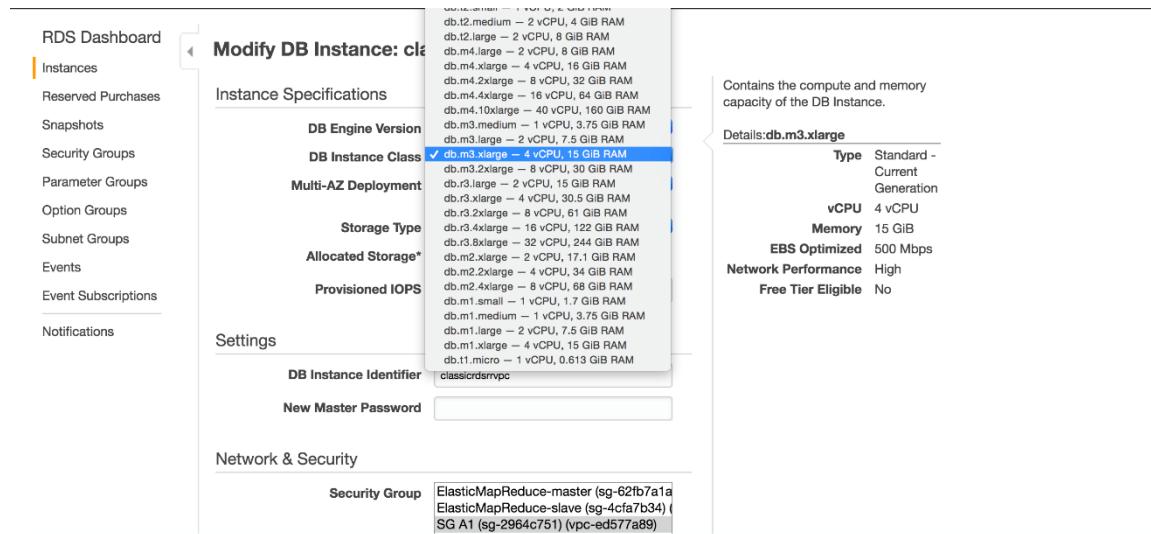
You can edit the **DB Subnet Groups** membership for your VPC RDS DB instance to move the VPC RDS DB instance to a private subnet. In the following example, the subnets 172.16.2.0/24 and 172.16.3.0/24 are private subnets.

The screenshot shows the AWS RDS Dashboard with the 'Subnet Groups' tab selected. The main panel displays the 'Edit DB Subnet Group' interface for 'default-vpc-ed577a89'. It shows the 'VPC ID' as 'VPC A (vpc-ed577a89)' and a 'Description' of 'Created from the RDS M...'. Below, it says 'Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or [add all the subnets](#) related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.' An 'Availability Zone' dropdown is set to 'us-east-1a'. A 'Subnet ID' dropdown is set to '- Select One -' with an 'Add' button next to it. A table lists subnets: 'us-east-1e' with 'subnet-5252216f' and '172.16.3.0/24', and 'us-east-1a' with 'subnet-d0b90a88' and '172.16.2.0/24'. At the bottom are 'Cancel' and 'Save' buttons.

Figure 25: Configuring subnet groups

To change the private IP address of the RDS DB instance in the VPC you have to perform a scale-up or scale-down operation. For example, you could choose a larger instance size. After the IP address changes, you can scale again to the original instance size.

## Amazon Web Services – Move Amazon RDS MySQL Databases to Amazon VPC using Amazon EC2 ClassicLink and Read Replicas



**Figure 26: Forcing a scale optimization**

Note: Alternatively, you can open an AWS support request (<https://aws.amazon.com/contact-us/>) and the RDS Operations team will move the migrated VPC RDS instance to the private subnet.

## Alternative Approaches

There are other ways to approach migrating your Amazon RDS MySQL databases from EC2-Classic to EC2-VPC. We cover two alternatives here. One approach is to use AWS Database Migration Service and another is to specify a new VPC subnet for a DB instance using the AWS Management Console.

## AWS Database Migration Service (DMS)

An alternative approach to migration is to use AWS Database Migration Service (DMS). AWS DMS can migrate your data to and from the most widely used commercial and open-source databases. The service supports homogenous migrations such as Amazon RDS to Amazon RDS, as well as heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database.

Although AWS DMS can provide comprehensive ongoing replication of data, it replicates only a limited amount of data definition language (DDL). AWS DMS doesn't propagate items such as indexes, users, privileges, stored procedures, and other database changes not directly related to table data. In addition, AWS DMS does not automatically leverage RDS snapshots for the initial instance creation, which can increase migration time.

## Changing the VPC Subnet for a DB Instance

Amazon RDS provides a feature that allows you to easily move an RDS DB instance in EC2-Classic to a VPC. You specify a new VPC subnet for an existing DB instance in the Amazon RDS console, the Amazon RDS API, or the AWS command-line tools.

To specify a new subnet group, in the Amazon RDS console under **Network & Security, Subnet Group**, expand the drop-down list and select the subnet group that you want from the list. You can choose to apply this change immediately or during the next scheduled maintenance window.

However, there are a few limitations with this approach:

- The DB instance isn't available during the move. The move could take between 5 to 10 minutes.
- Moving Multi-AZ instances to a VPC isn't currently supported.
- Moving an instance with read replicas to a VPC isn't currently supported.

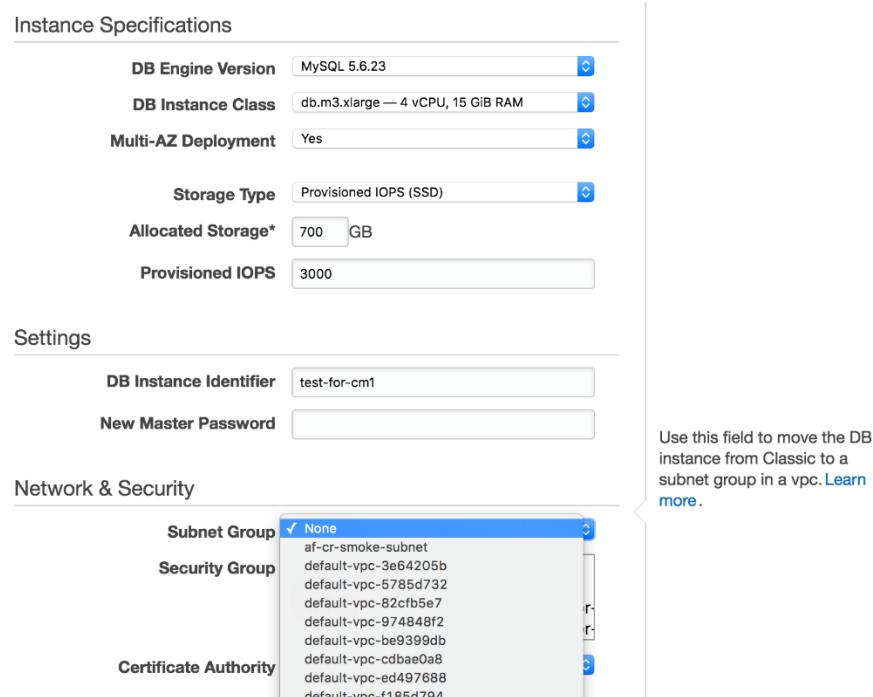


Figure 27: Specifying a new subnet group (in a VPC) for a database instance

If these limitations are acceptable for your DB instances, we recommend that you test this feature by restoring a snapshot of your database in EC2-Classic and then moving it to your VPC. If these limitations are not acceptable, then the ClassicLink approach presented in this whitepaper will enable you to minimize downtime during the migration to your VPC.

## Conclusion

This paper highlights the key steps for migrating RDS MySQL instances from EC2-Classic to EC2-VPC environments using ClassicLink and RDS read replicas. This approach enables minimal down-time for production environments.

## Contributors

The following individuals and organizations contributed to this document:

- Harshal Pimpalkhute, Sr. Product Manager, Amazon EC2 Networking
- Jaime Lichauco, Database Administrator, Amazon RDS

- Korey Knote, Database Administrator, Amazon RDS
- Brian Welcker, Product Manager, Amazon RDS
- Prahlad Rao, Solutions Architect, Amazon Web Services

## Further Reading

For additional help, please consult the following sources:

- [http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_VPC.html](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.html)
- [http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_VPC.WorkingWithRDSDInstanceinaVPC.html](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.WorkingWithRDSDInstanceinaVPC.html)
- [http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_MySQL.html](http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MySQL.html)
- [http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC\\_Networking.html](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Networking.html)
- <http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-classiclink.html>

## Appendix A: Set Up Proxy Server in Classic

Use an Amazon Machine Image (AMI) of your choice to launch an EC2 Classic instance. The following example is based on the AMI Ubuntu Server 14.04 LTS (HVM).

Connect to the EC2-Classic instance and install NGINX:

```
Prompt> sudo apt-get update
Prompt> sudo wget http://nginx.org/download/nginx-1.9.12.tar.gz
Prompt> sudo tar -xvzf nginx-1.9.12.tar.gz
Prompt> cd nginx-1.9.12
Prompt> sudo apt-get install build-essential
Prompt> sudo apt-get install libpcre3 libpcre3-dev
Prompt> sudo apt-get install zlib1g-dev
Prompt> sudo ./configure --with-stream
```

```
Prompt> sudo make  
Prompt> sudo make install
```

Edit the NGINX daemon file /etc/init/nginx.conf:

```
# /etc/init/nginx.conf - Upstart file  
  
description "nginx http daemon"  
author "email"  
  
start on (filesystem and net-device-up IFACE=lo)  
stop on runlevel [!2345]  
  
env DAEMON=/usr/local/nginx/sbin/nginx  
env PID=/usr/local/nginx/logs/nginx.pid  
  
expect fork  
respawn  
respawn limit 10 5  
  
pre-start script  
    $DAEMON -t  
    if [ $? -ne 0 ]  
        then exit $?  
    fi  
end script  
  
exec $DAEMON
```

Edit the NGINX configuration file /usr/local/nginx/conf/nginx.conf:

```
# /usr/local/nginx/conf/nginx.conf - NGINX configuration file  
  
worker_processes 1;  
events {  
    worker_connections 1024;  
}  
  
stream {
```

```
server {  
    listen 3306;  
    proxy_pass classicrdsinstance.123456789012.us-east-  
1.rds.amazonaws.com:3306;  
}  
}
```

From the command line start NGINX:

```
Prompt> sudo initctl reload-configuration  
Prompt> sudo initctl list | grep nginx  
Prompt> sudo initctl start nginx
```

Configure NGINX port forwarding:

```
# /usr/local/nginx/conf/nginx.conf - NGINX configuration file  
worker_processes 1;  
events {  
    worker_connections 1024;  
}  
stream {  
    server {  
        listen 3306;  
        proxy_pass classicrdsinstance.123456789012.us-east-  
1.rds.amazonaws.com:3306;  
    }  
}
```