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

July 2017

Notices

Contents

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.

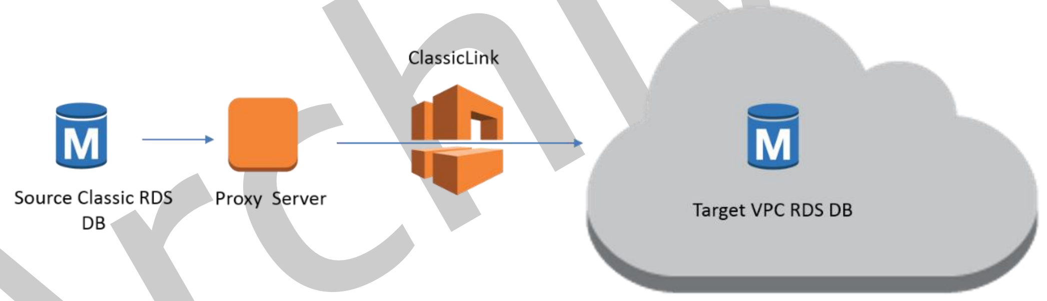


Figure 1: Topology for migration in the same account

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.

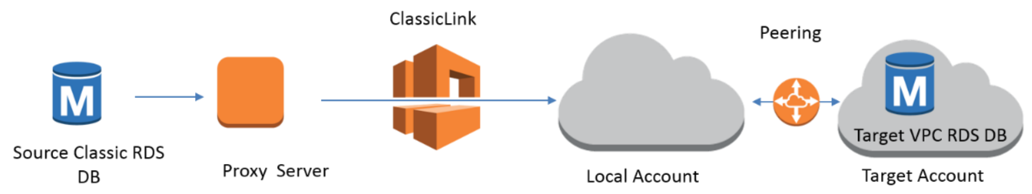


Figure 2: Topology for migration to a different account

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

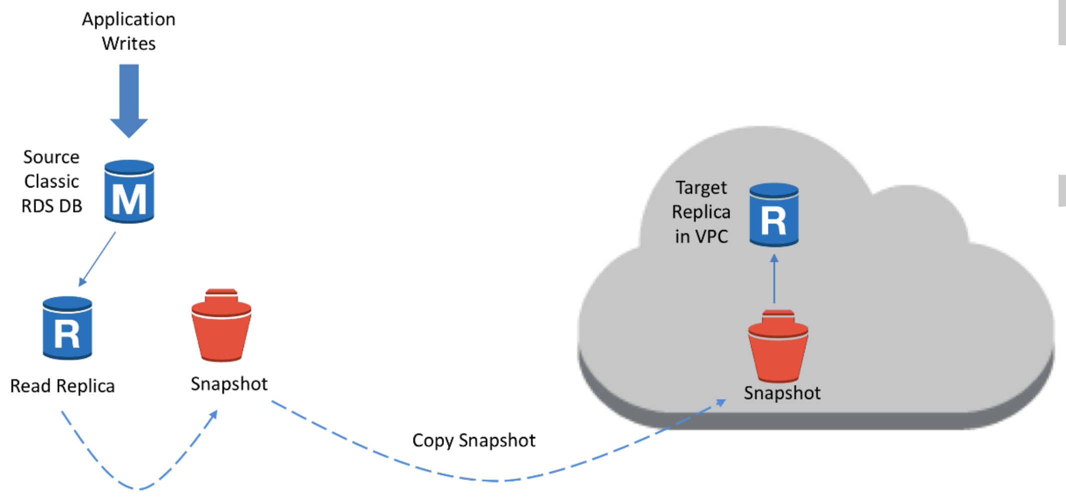


Figure 3: Creating a read replica snapshot and restoring in VPC

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

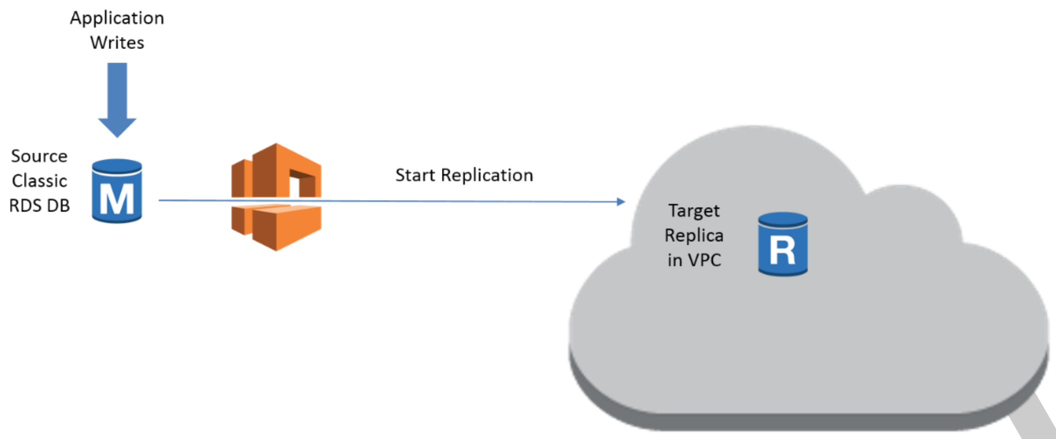


Figure 4: Setting up replication between the Classic and VPC read replica

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

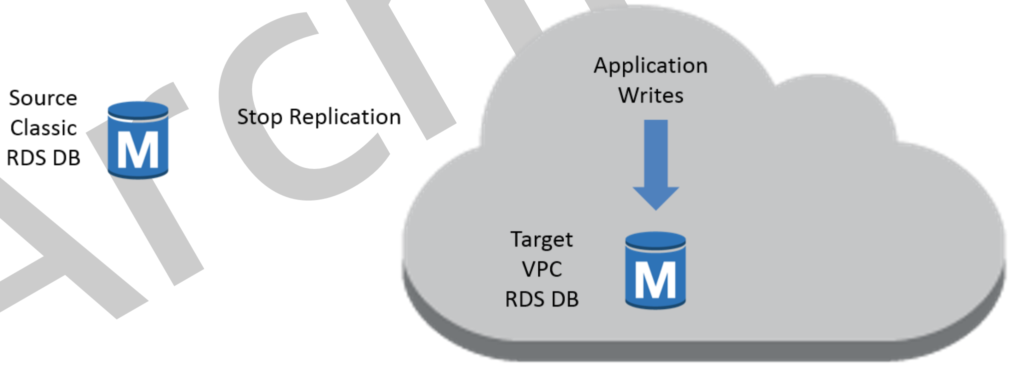


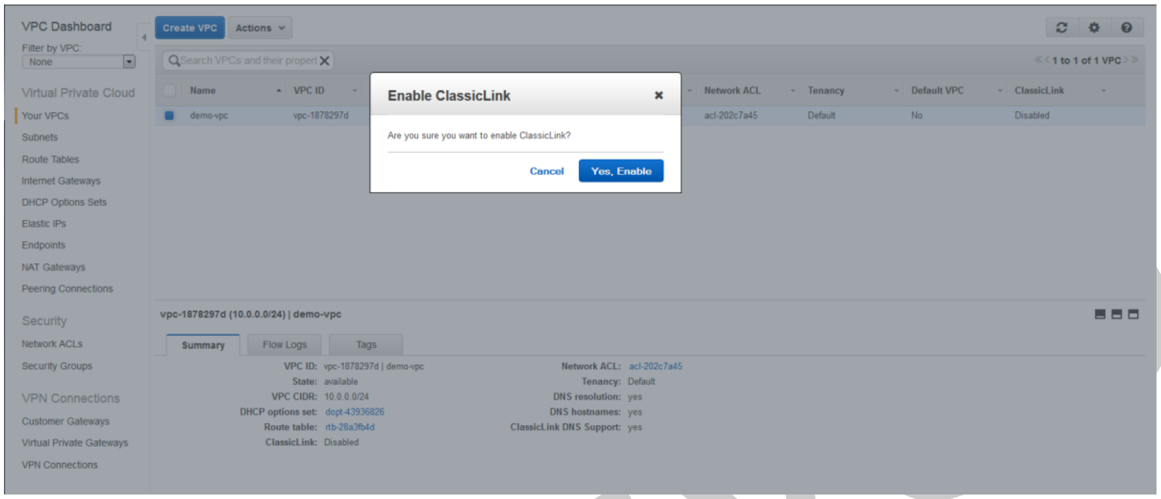
Figure 5: Cutting over application to the VPC RDS DB instance

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:



Enable ClassicLink

Are you sure you want to enable ClassicLink?

Name	VPC ID	Network ACL	Tenancy	Default VPC	ClassicLink
demo-vpc	vpc-1878297d	acl-202c7a45	Default	No	Disabled

vpc-1878297d (10.0.0.0/24) | demo-vpc

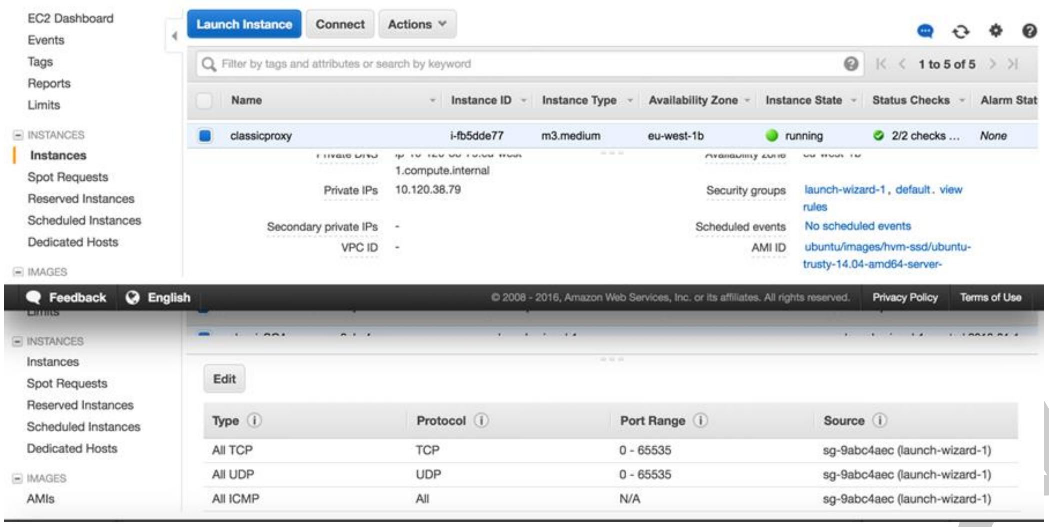
VPC ID:	vpc-1878297d | demo-vpc	Network ACL:	acl-202c7a45
State:	available	Tenancy:	Default
VPC CIDR:	10.0.0.0/24	DNS resolution:	yes
DHCP options set:	dopt-43936826	DNS hostnames:	yes
Route table:	rtb-28a3fb4d	ClassicLink DNS Support:	yes
ClassicLink:	Disabled		

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.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Stat
classicproxy	i-fb5dde77	m3.medium	eu-west-1b	running	2/2 checks ...	None

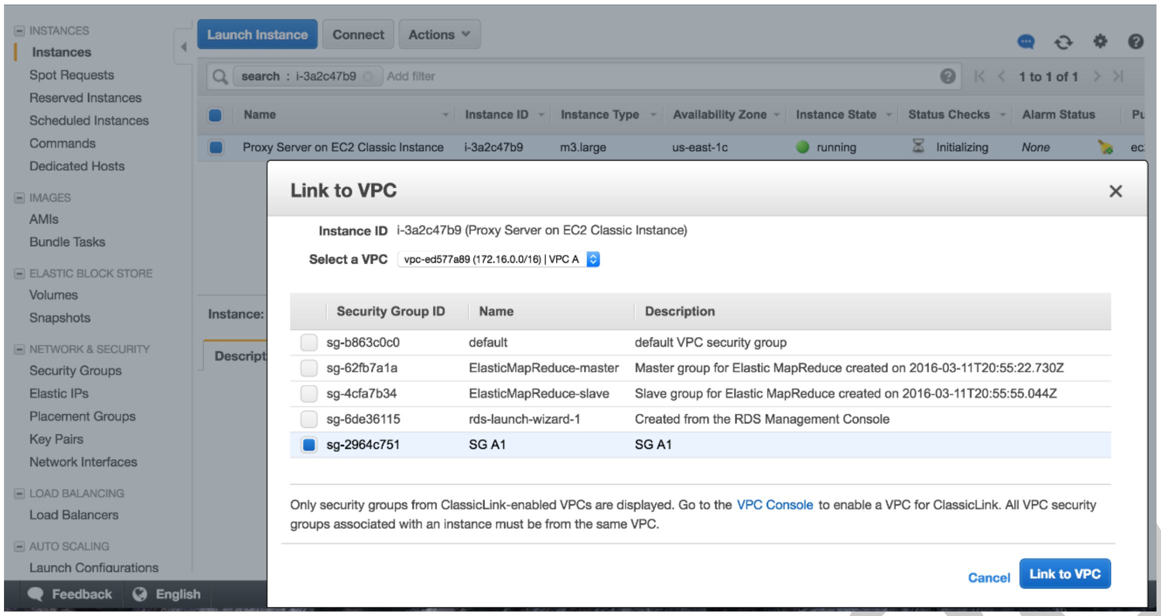
1.compute.internal
Private IPs 10.120.38.79
Secondary private IPs -
VPC ID -
Security groups launch-wizard-1, default. view rules
Scheduled events No scheduled events
AMI ID ubuntu/images/hvm-ssd/ubuntu-trusty-14.04-amd64-server-

Type	Protocol	Port Range	Source
All TCP	TCP	0 - 65535	sg-9abc4aec (launch-wizard-1)
All UDP	UDP	0 - 65535	sg-9abc4aec (launch-wizard-1)
All ICMP	All	N/A	sg-9abc4aec (launch-wizard-1)

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**



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Pu
Proxy Server on EC2 Classic Instance	i-3a2c47b9	m3.large	us-east-1c	running	Initializing	None	ec

Link to VPC

Instance ID i-3a2c47b9 (Proxy Server on EC2 Classic Instance)

Select a VPC vpc-ed577a89 (172.16.0.0/16) | VPC A

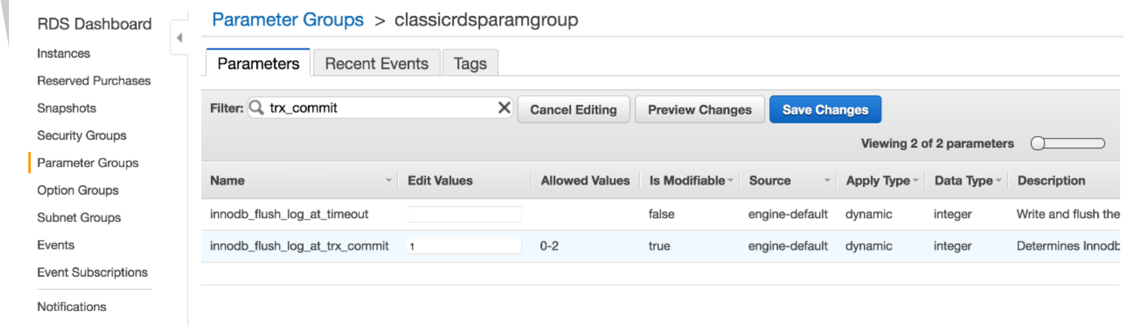
	Security Group ID	Name	Description
☐	sg-b863c0c0	default	default VPC security group
☐	sg-62fb7a1a	ElasticMapReduce-master	Master group for Elastic MapReduce created on 2016-03-11T20:55:22.730Z
☐	sg-4cfa7b34	ElasticMapReduce-slave	Slave group for Elastic MapReduce created on 2016-03-11T20:55:55.044Z
☐	sg-6de36115	rds-launch-wizard-1	Created from the RDS Management Console
☑	sg-2964c751	SG A1	SG A1

Only security groups from ClassicLink-enabled VPCs are displayed. Go to the VPC Console to enable a VPC for ClassicLink. All VPC security groups associated with an instance must be from the same VPC.

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.



Filter: trx_commit

Viewing 2 of 2 parameters

Name	Edit Values	Allowed Values	Is Modifiable	Source	Apply Type	Data Type	Description
innodb_flush_log_at_timeout			false	engine-default	dynamic	integer	Write and flush the
innodb_flush_log_at_trx_commit	1	0-2	true	engine-default	dynamic	integer	Determines Innodb

Figure 9: Parameter group values on a Classic DB instance

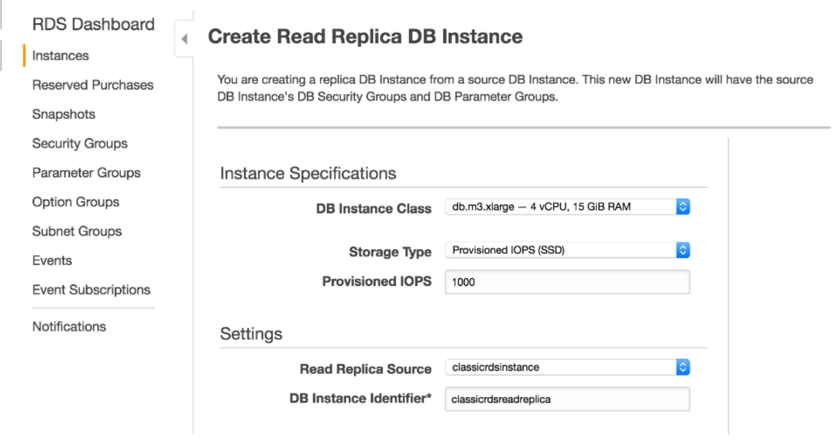
Step 5: Create a User on the DB Instance (EC2-Classic)

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.



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.

Instance Specifications

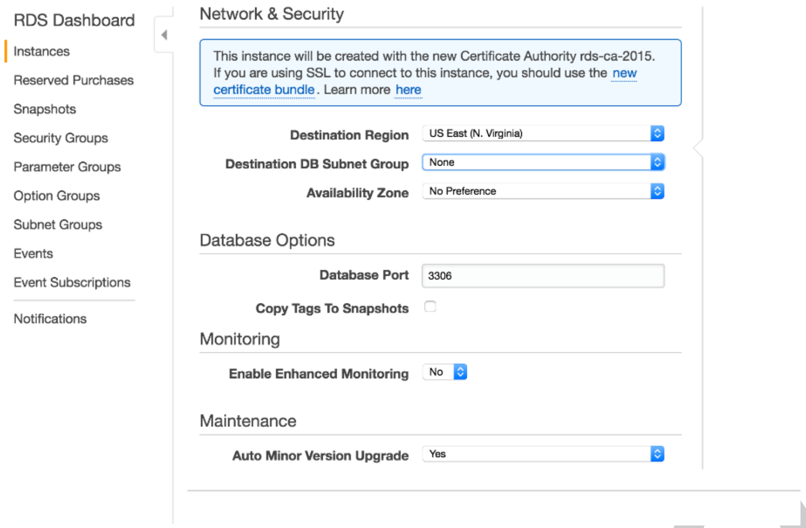
DB Instance Class	db.m3.xlarge — 4 vCPU, 15 GiB RAM
Storage Type	Provisioned IOPS (SSD)
Provisioned IOPS	1000

Settings

Read Replica Source	classicrdsinstance
DB Instance Identifier*	classicrdsreadreplica

Figure 10: Classic read replica instance properties

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



Network & Security

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

Destination Region	US East (N. Virginia)
Destination DB Subnet Group	None
Availability Zone	No Preference

Database Options

Database Port	3306
Copy Tags To Snapshots	☐

Monitoring

Enable Enhanced Monitoring	No

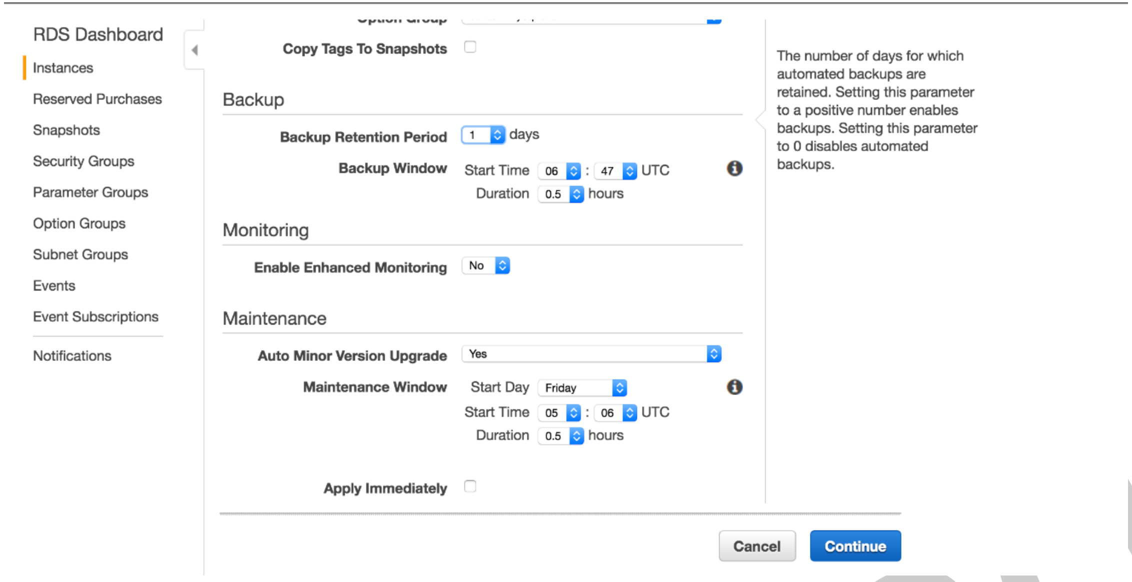
Maintenance

Auto Minor Version Upgrade	Yes

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.



Copy Tags To Snapshots

Backup

Backup Retention Period 1 days

Backup Window Start Time 06 : 47 UTC
Duration 0.5 hours

Monitoring

Enable Enhanced Monitoring No

Maintenance

Auto Minor Version Upgrade Yes

Maintenance Window Start Day Friday
Start Time 05 : 06 UTC
Duration 0.5 hours

Apply Immediately

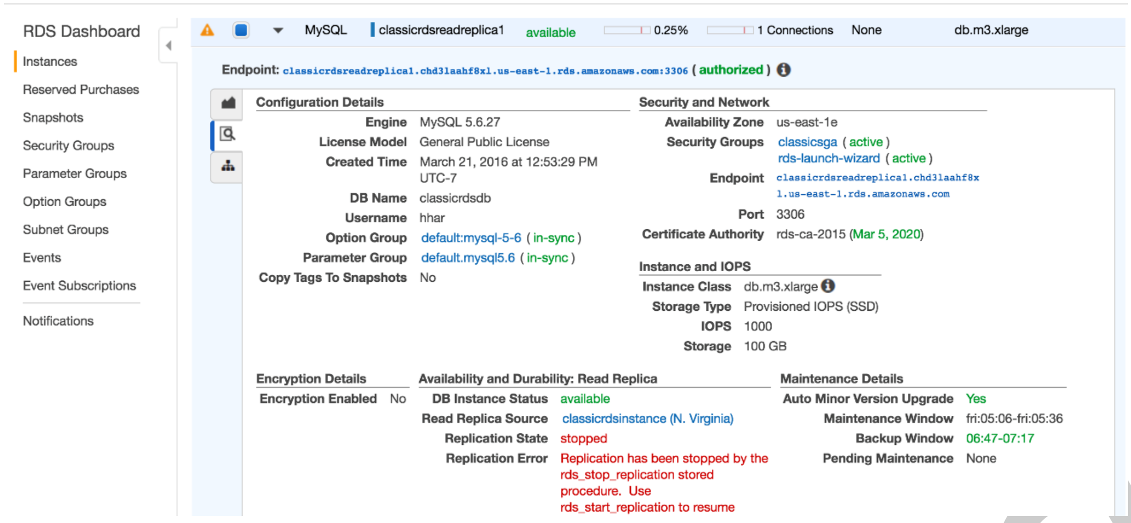
The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

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)]>
```



MySQL | classicrdsreadreplica1 | available | 0.25% | 1 Connections | None | db.m3.xlarge

Endpoint: classicrdsreadreplica1.chd3laahf8xl.us-east-1.rds.amazonaws.com:3306 (authorized)

Configuration Details

Engine	MySQL 5.6.27
License Model	General Public License
Created Time	March 21, 2016 at 12:53:29 PM UTC-7
DB Name	classicrdsdb
Username	hhar
Option Group	default:mysql-5-6 (in-sync)
Parameter Group	default.mysql5.6 (in-sync)
Copy Tags To Snapshots	No

Security and Network

Availability Zone	us-east-1e
Security Groups	classicsga (active) rds-launch-wizard (active)
Endpoint	classicrdsreadreplica1.chd3laahf8x1.us-east-1.rds.amazonaws.com
Port	3306
Certificate Authority	rds-ca-2015 (Mar 5, 2020)

Instance and IOPS

Instance Class	db.m3.xlarge
Storage Type	Provisioned IOPS (SSD)
IOPS	1000
Storage	100 GB

Encryption Details

Encryption Enabled	No

Availability and Durability: Read Replica

DB Instance Status	available
Read Replica Source	classicrdsinstance (N. Virginia)
Replication State	stopped
Replication Error	Replication has been stopped by the rds_stop_replication stored procedure. Use rds_start_replication to resume

Maintenance Details

Auto Minor Version Upgrade	Yes
Maintenance Window	fri:05:06-fri:05:36
Backup Window	06:47-07:17
Pending Maintenance	None

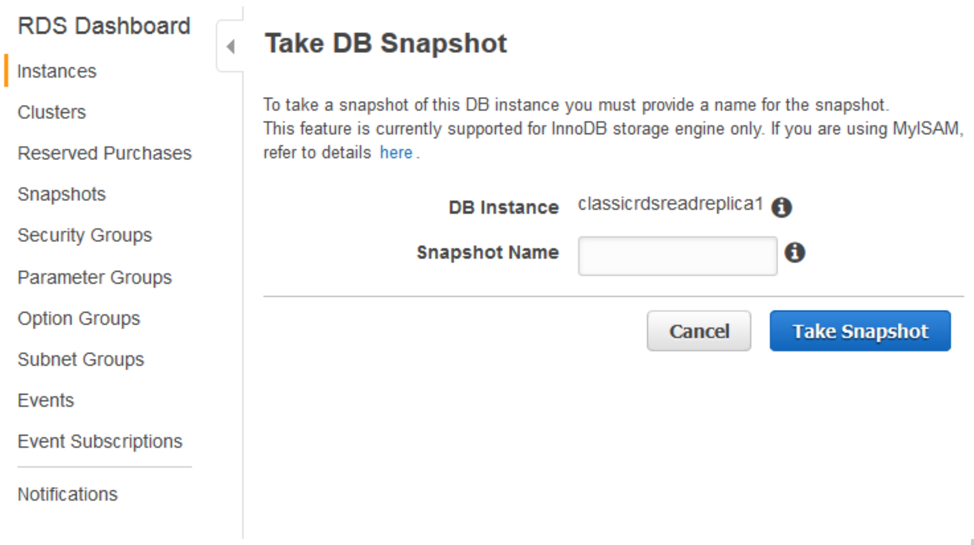
Figure13: Confirmation of replica status on the console

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.chd3laahf8xl.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.



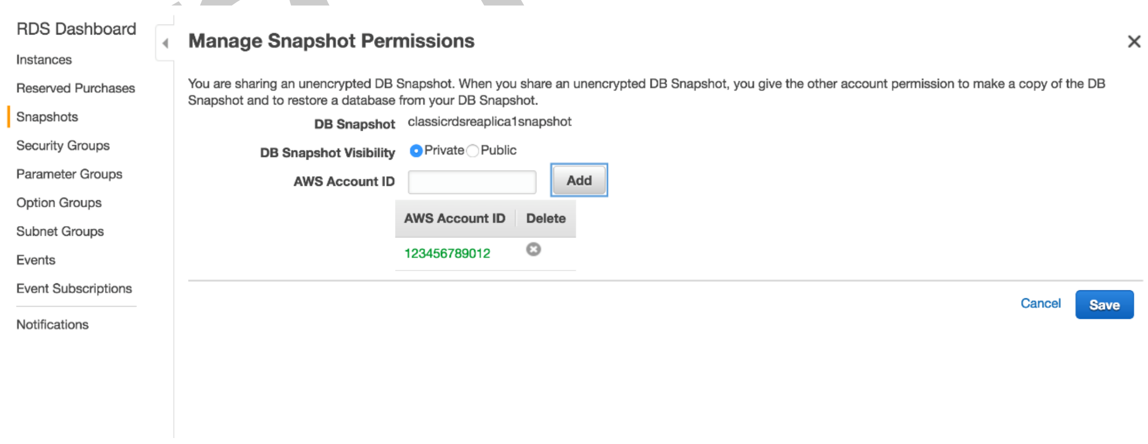
To take a snapshot of this DB instance you must provide a name for the snapshot.
This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.

DB Instance: classicrdsreadreplica1
Snapshot Name:

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.



You are sharing an unencrypted DB Snapshot. When you share an unencrypted DB Snapshot, you give the other account permission to make a copy of the DB Snapshot and to restore a database from your DB Snapshot.

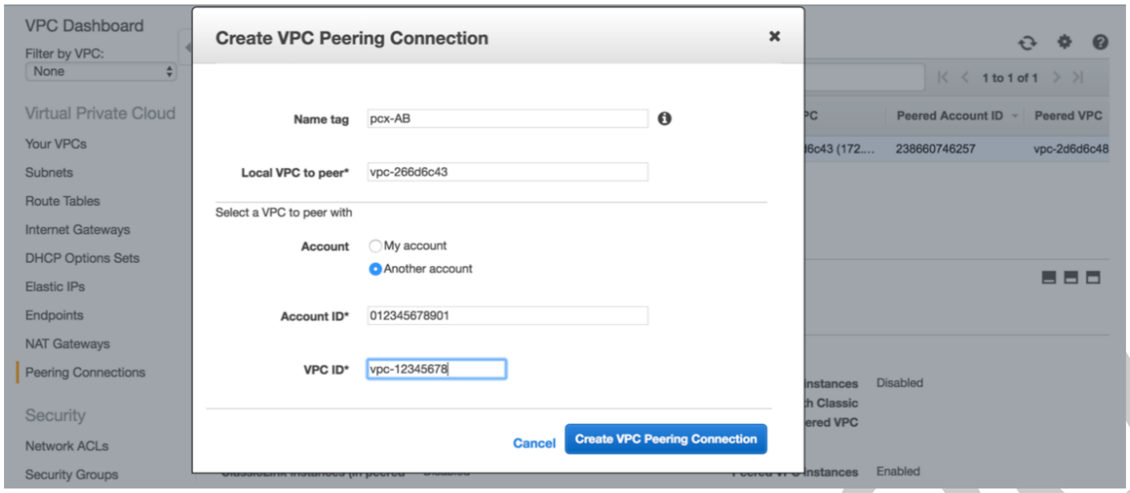
DB Snapshot: classicrdsreaplica1snapshot
DB Snapshot Visibility: [✓ Private] Public
AWS Account ID:

AWS Account ID	Delete
123456789012	⊗

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.

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

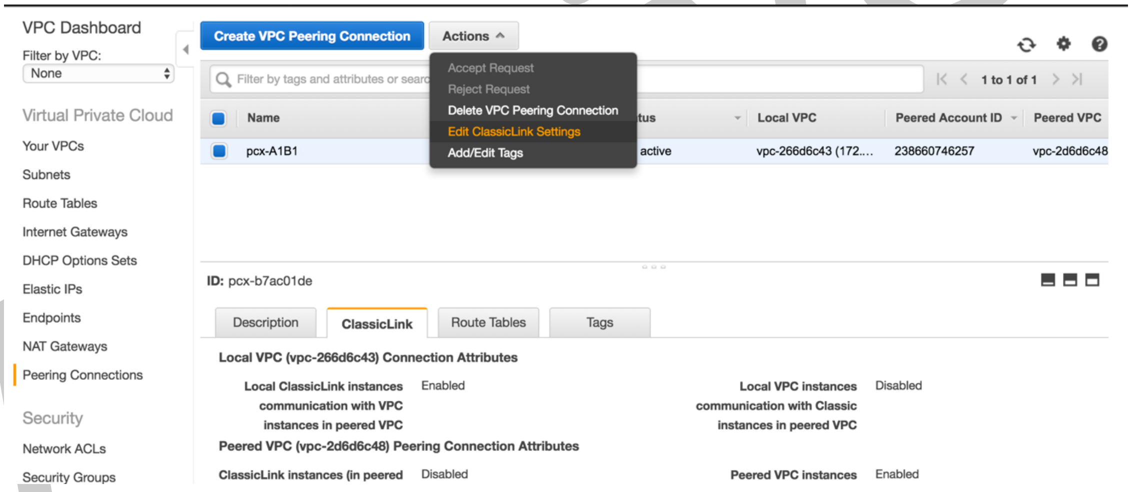


Name tag	pcx-AB
Local VPC to peer*	vpc-266d6c43

Select a VPC to peer with

Account	My account [✓ Another account]
Account ID*	012345678901
VPC ID*	vpc-12345678

Figure 16: Creating a VPC peering connection



Name	Status	Local VPC	Peered Account ID	Peered VPC
pcx-A1B1	active	vpc-266d6c43 (172....	238660746257	vpc-2d6d6c48

ID: pcx-b7ac01de

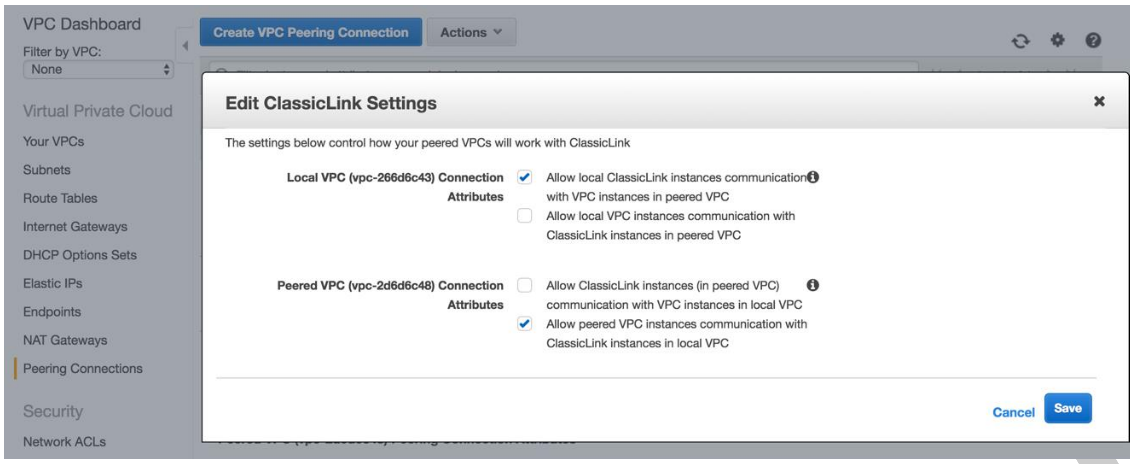
Local VPC (vpc-266d6c43) Connection Attributes

Local ClassicLink instances communication with VPC instances in peered VPC	Enabled	Local VPC instances communication with Classic instances in peered VPC	Disabled

Peered VPC (vpc-2d6d6c48) Peering Connection Attributes

ClassicLink instances (in peered	Disabled	Peered VPC instances	Enabled

Figure 17: Enabling ClassicLink over a peering connection



Edit ClassicLink Settings

The settings below control how your peered VPCs will work with ClassicLink

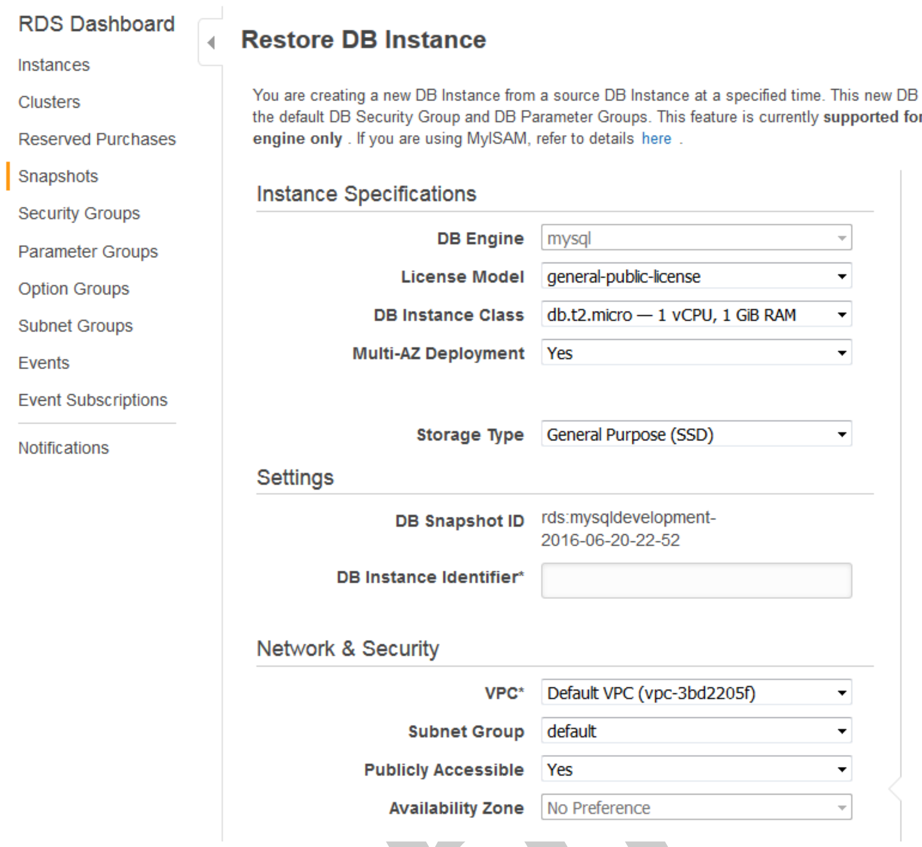
Local VPC (vpc-266d6c43) Connection Attributes
- [✓] Allow local ClassicLink instances communication with VPC instances in peered VPC
- [ ] Allow local VPC instances communication with ClassicLink instances in peered VPC

Peered VPC (vpc-2d6d6c48) Connection Attributes
- [ ] Allow ClassicLink instances (in peered VPC) communication with VPC instances in local VPC
- [✓] Allow peered VPC instances communication with ClassicLink instances in local VPC

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.



RDS Dashboard
Instances
Clusters
Reserved Purchases
Snapshots
Security Groups
Parameter Groups
Option Groups
Subnet Groups
Events
Event Subscriptions
Notifications

Restore DB Instance

You are creating a new DB Instance from a source DB Instance at a specified time. This new DB Instance will have the default DB Security Group and DB Parameter Groups. This feature is currently **supported for InnoDB storage engine only** . If you are using MyISAM, refer to details here .

Instance Specifications

DB Engine	mysql
License Model	general-public-license
DB Instance Class	db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ Deployment	Yes
Storage Type	General Purpose (SSD)

Settings

DB Snapshot ID	rds:mysqldevelopment-2016-06-20-22-52
DB Instance Identifier*	

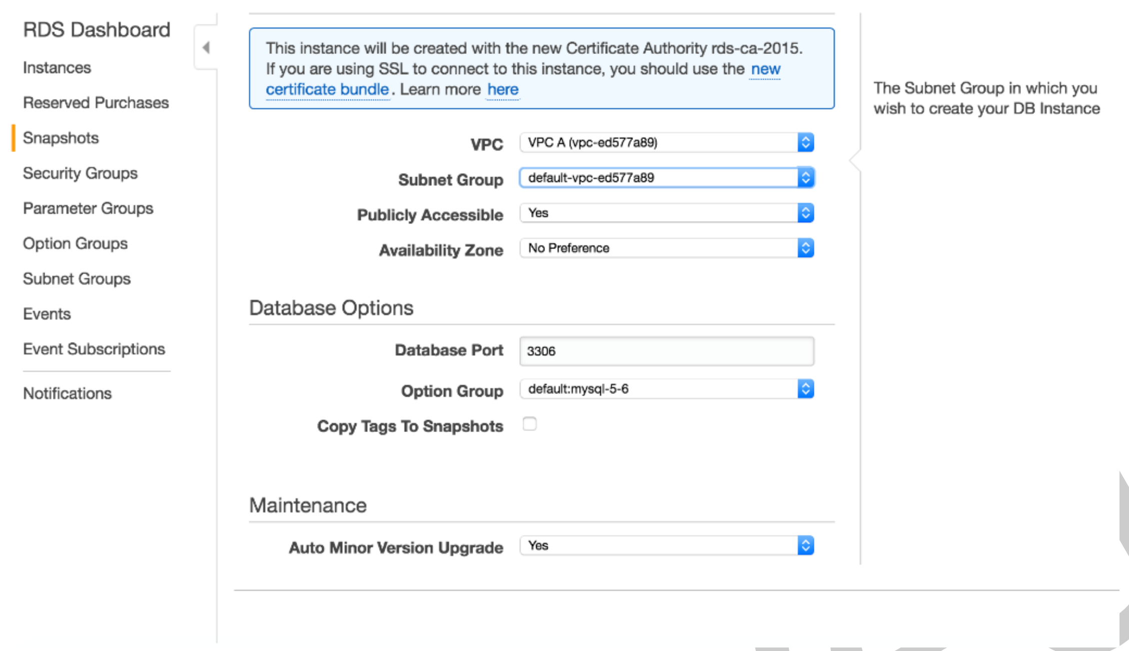
Network & Security

VPC*	Default VPC (vpc-3bd2205f)
Subnet Group	default
Publicly Accessible	Yes
Availability Zone	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.



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

The Subnet Group in which you wish to create your DB Instance

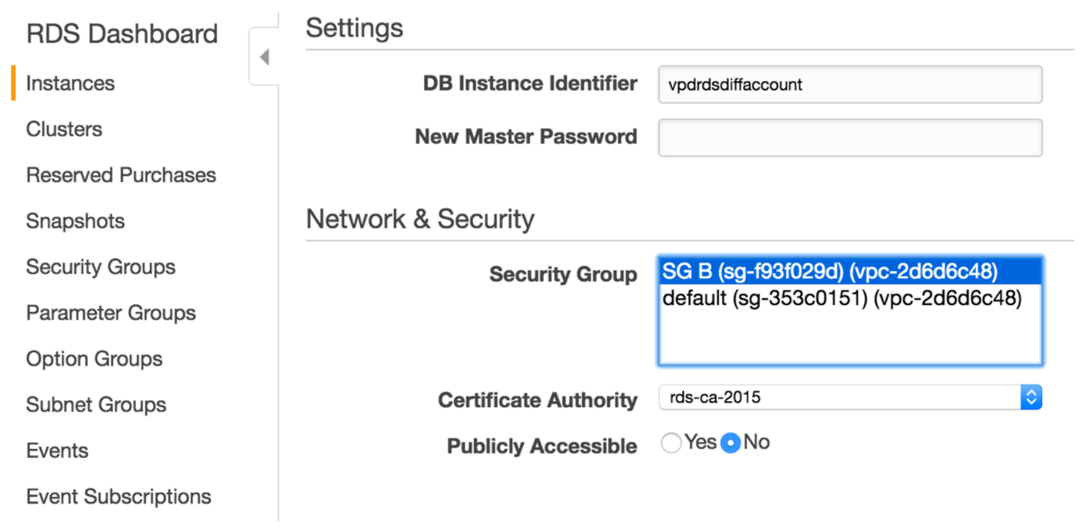
Database Options

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



Settings

DB Instance Identifier	vpdrdsdiffaccount
New Master Password	

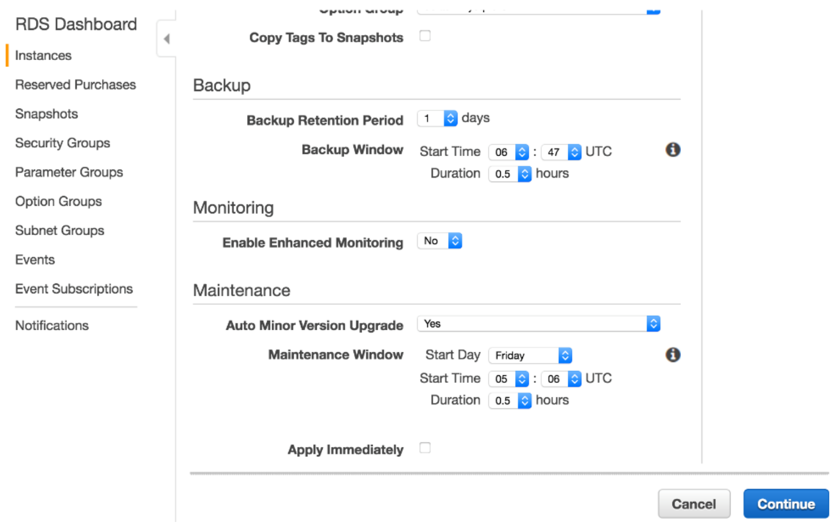
Network & Security

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.



Copy Tags To Snapshots

Backup

Backup Retention Period: 1 days

Backup Window: Start Time 06 : 47 UTC, Duration 0.5 hours

Monitoring

Enable Enhanced Monitoring: No

Maintenance

Auto Minor Version Upgrade: Yes

Maintenance Window: Start Day Friday, Start Time 05 : 06 UTC, Duration 0.5 hours

Apply Immediately

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;
```

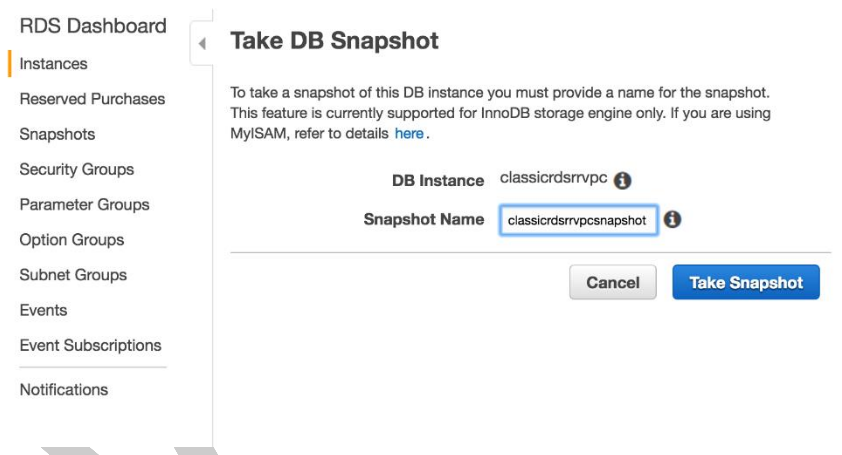
At this point, the VPC will stop replicating data from the master. You can now promote the replica by connecting to the VPC RDS DB instance via a `mysql` client and issuing the `mysql.rds_reset_external_master` command.

```
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.



To take a snapshot of this DB instance you must provide a name for the snapshot. This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.

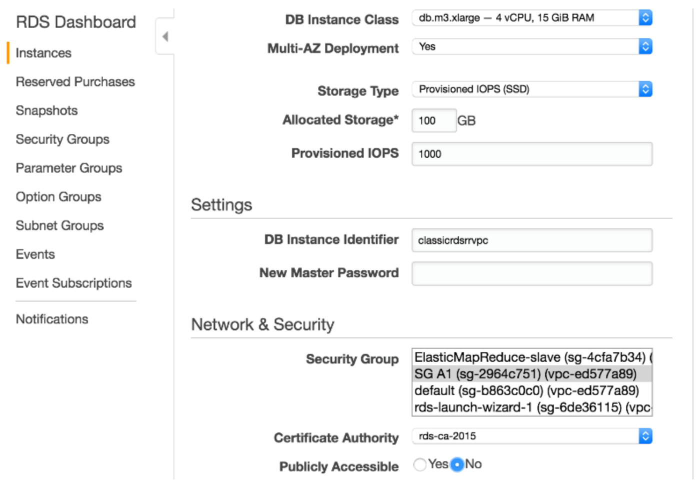
DB Instance classicrdsrrvpc

Snapshot Name classicrdsrrvpcsnapshot

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**.



DB Instance Class	db.m3.xlarge — 4 vCPU, 15 GiB RAM
Multi-AZ Deployment	Yes
Storage Type	Provisioned IOPS (SSD)
Allocated Storage*	100 GB
Provisioned IOPS	1000

Settings

DB Instance Identifier	classicrdsrrvpc
New Master Password	

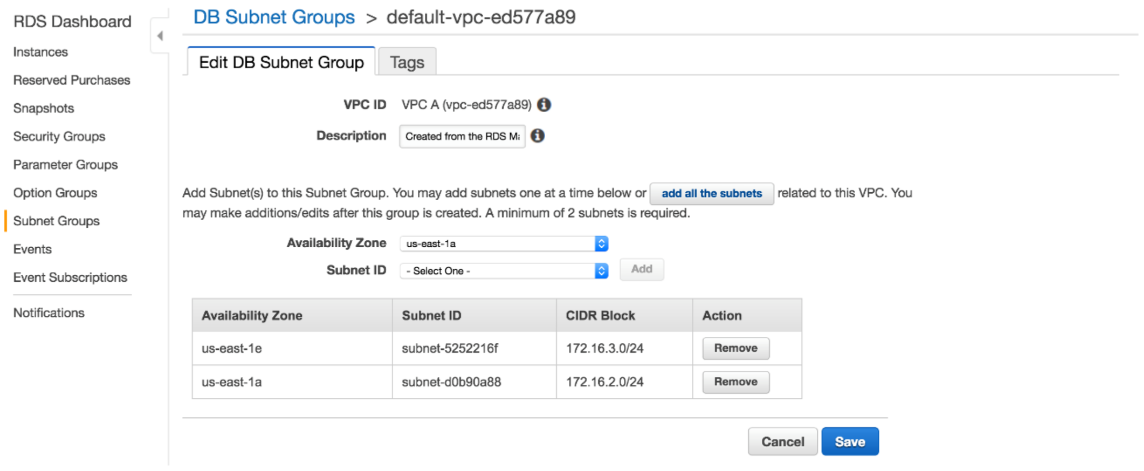
Network & Security

Security Group	ElasticMapReduce-slave (sg-4cfa7b34) (SG A1 (sg-2964c751) (vpc-ed577a89) default (sg-b863c0c0) (vpc-ed577a89) rds-launch-wizard-1 (sg-6de36115) (vpc
Certificate Authority	rds-ca-2015
Publicly Accessible	Yes [✓ No]

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.



VPC ID	VPC A (vpc-ed577a89)
Description	Created from the RDS M:

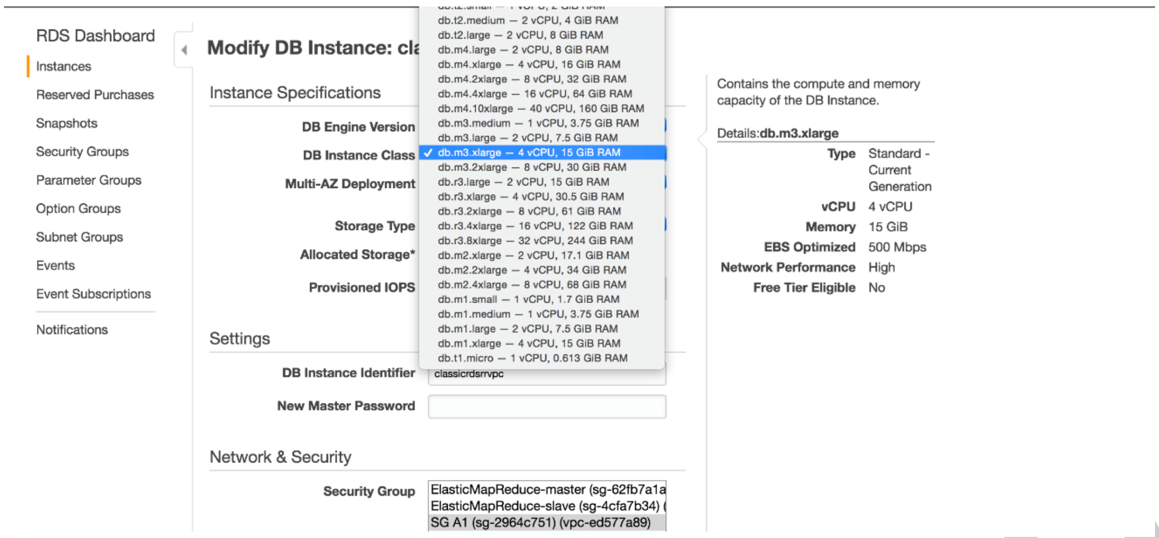
Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created. A minimum of 2 subnets is required.

Availability Zone	us-east-1a
Subnet ID	- Select One -

Availability Zone	Subnet ID	CIDR Block	Action
us-east-1e	subnet-5252216f	172.16.3.0/24	Remove
us-east-1a	subnet-d0b90a88	172.16.2.0/24	Remove

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.



Modify DB Instance: cla

Instance Specifications

DB Engine Version
DB Instance Class
Multi-AZ Deployment
Storage Type
Allocated Storage*
Provisioned IOPS

db.t2.medium – 2 vCPU, 4 GiB RAM
db.t2.large – 2 vCPU, 8 GiB RAM
db.m4.large – 2 vCPU, 8 GiB RAM
db.m4.xlarge – 4 vCPU, 16 GiB RAM
db.m4.2xlarge – 8 vCPU, 32 GiB RAM
db.m4.4xlarge – 16 vCPU, 64 GiB RAM
db.m4.10xlarge – 40 vCPU, 160 GiB RAM
db.m3.medium – 1 vCPU, 3.75 GiB RAM
db.m3.large – 2 vCPU, 7.5 GiB RAM
✓ db.m3.xlarge – 4 vCPU, 15 GiB RAM
db.m3.2xlarge – 8 vCPU, 30 GiB RAM
db.r3.large – 2 vCPU, 15 GiB RAM
db.r3.xlarge – 4 vCPU, 30.5 GiB RAM
db.r3.2xlarge – 8 vCPU, 61 GiB RAM
db.r3.4xlarge – 16 vCPU, 122 GiB RAM
db.r3.8xlarge – 32 vCPU, 244 GiB RAM
db.m2.xlarge – 2 vCPU, 17.1 GiB RAM
db.m2.2xlarge – 4 vCPU, 34 GiB RAM
db.m2.4xlarge – 8 vCPU, 68 GiB RAM
db.m1.small – 1 vCPU, 1.7 GiB RAM
db.m1.medium – 1 vCPU, 3.75 GiB RAM
db.m1.large – 2 vCPU, 7.5 GiB RAM
db.m1.xlarge – 4 vCPU, 15 GiB RAM
db.t1.micro – 1 vCPU, 0.613 GiB RAM

Settings

DB Instance Identifier: classicrdsrrvpc
New Master Password:

Network & Security

Security Group: ElasticMapReduce-master (sg-62fb7a1a
ElasticMapReduce-slave (sg-4cfa7b34) (
SG A1 (sg-2964c751) (vpc-ed577a89)

Contains the compute and memory capacity of the DB Instance.

Details:**db.m3.xlarge**

Type	Standard - Current Generation
vCPU	4 vCPU
Memory	15 GiB
EBS Optimized	500 Mbps
Network Performance	High
Free Tier Eligible	No

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.

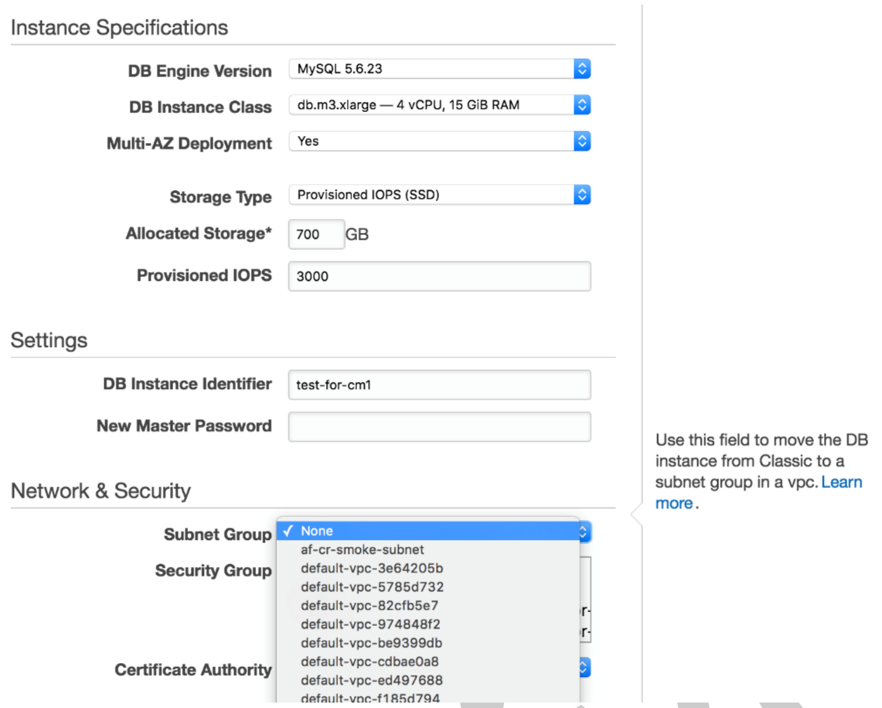
Changing the VPC Subnet for a DB Instance

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.



Instance Specifications

DB Engine Version	MySQL 5.6.23
DB Instance Class	db.m3.xlarge — 4 vCPU, 15 GiB RAM
Multi-AZ Deployment	Yes
Storage Type	Provisioned IOPS (SSD)
Allocated Storage*	700 GB
Provisioned IOPS	3000

Settings

DB Instance Identifier	test-for-cm1
New Master Password	

Network & Security

Subnet Group	✓ None
Security Group	
Certificate Authority	

af-cr-smoke-subnet
default-vpc-3e64205b
default-vpc-5785d732
default-vpc-82cfb5e7
default-vpc-974848f2
default-vpc-be9399db
default-vpc-cdbae0a8
default-vpc-ed497688
default-vpc-f185d794

Use this field to move the DB instance from Classic to a subnet group in a vpc. Learn more.

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.WorkingWithRDSInstanceinaVPC.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-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;
    }
}
```