RDS Aurora Connectivity

This document describes how to configure access to RDS Aurora instances from on-premises equipment and EC2-Classic instances. It also describes the recommended configuration which hides the RDS Aurora instances from the Internet.

Getting Started

Establishing connectivity to RDS Aurora instances is mostly about configuring a Virtual Private Cloud (VPC) to suit your requirements. All RDS Aurora database instances reside in a VPC associated with your AWS account. If you already have a VPC and would like to create Aurora instances in it, these instructions will provide examples of how to configure your VPC to allow access to your RDS Aurora instances. If you don't have a VPC or would like to create a new VPC for your RDS Aurora instances, the RDS Launch Wizard will create and configure a VPC for you.

Public vs. Private Accessibility

One of the choices you'll make when creating RDS Aurora instances is whether or not the instance will be publicly accessible. If you will be accessing your RDS Aurora instances exclusively from EC2 instances or devices in the same VPC as the RDS Aurora instances, answer "No" in the Publicly Accessible field (see example below). When the RDS Aurora instance is created, it will have a private IP address, but no public (Internet routable) IP address.

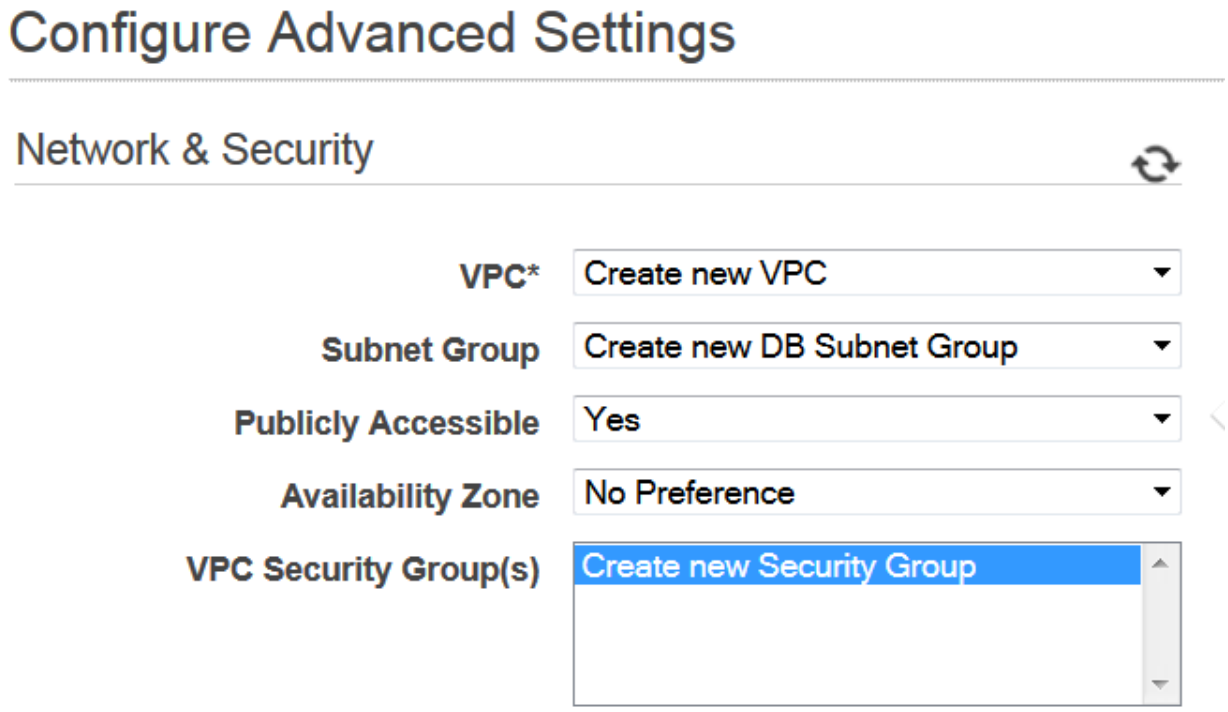
If you plan to access your RDS Aurora instances from outside the VPC, such as from your on-premises equipment or from AWS EC2 instances in other AWS Regions, answering "Yes" in the Publicly Accessible field will provide the RDS Aurora instance with a public (Internet routable) IP address as well as a private (non-Internet routable) one. Note that there may be additional steps required to configure your VPC to allow access to the RDS Aurora instance from outside the VPC, such as configuring VPC Route Tables, Network ACLs, and Security Groups. Examples are provided in the sections below.

ClassicLink

If you plan to access your RDS Aurora instances from EC2 instances residing in the same region, but not in a VPC (commonly known as EC2-Classic), you can enable ClassicLink on the VPC where your RDS Aurora instances reside. Enabling ClassicLink allows your EC2-Classic instances to communicate with your RDS Aurora instances using their Private IP address. Doing so allows you to take advantage of the higher throughput and lower latency connectivity available for inter-instance communication within AWS, avoid network bandwidth charges associated with communicating over the Internet, and may improve security.

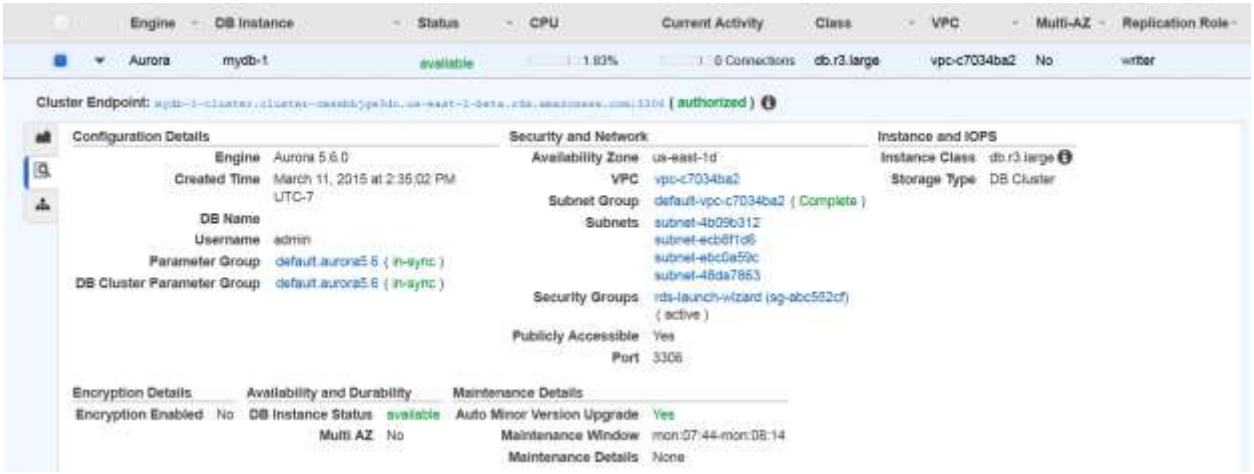
Starting from Scratch (No existing VPC or Aurora Instances)

The simplest way to create a VPC for your Aurora instance is to let the RDS Launch Wizard do it for you. It will create and configure the VPC and create a new RDS Aurora instance in it. The figure below shows an example of using the RDS Launch Wizard to create a new VPC and make the RDS Aurora instance publicly accessible.



RDS Aurora Instance

Here's the RDS Aurora instance that the RDS Launch Wizard created for us. Notice that it is publicly accessible and that it has been assigned to a Security Group and four Subnet Groups - one for each Availability Zone in the AWS Region.



VPC

Let's take a look at the VPC the RDS Launch Wizard created for us. Notice that DNS resolution and DNS hostnames VPC attributes have been enabled since we specified that the RDS Aurora instance will be publicly accessible.

vpc-c7034ba2 (172.30.0.0/16)

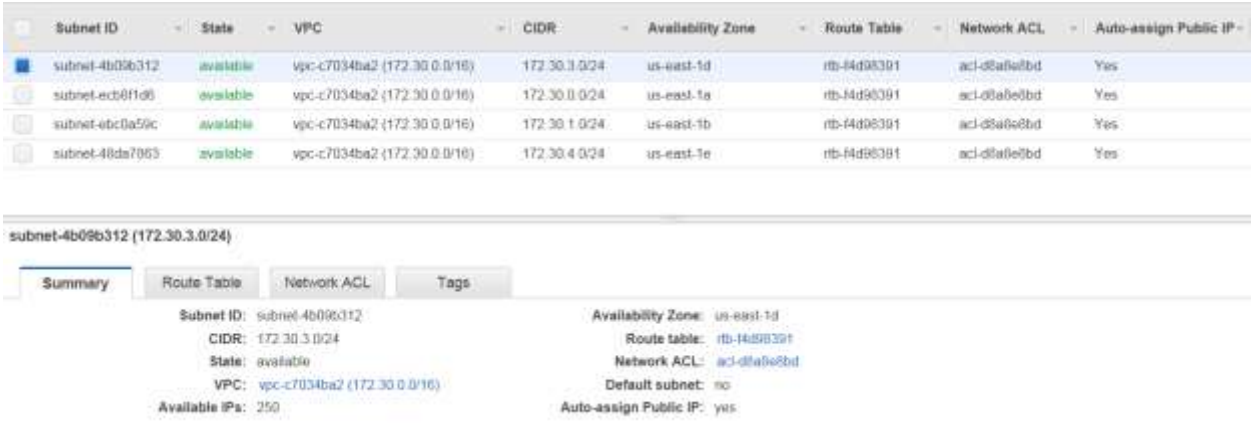
Summary | Tags

VPC ID:	vpc-c7034ba2	**Network ACL:**	acl-d8a8e8bd
State:	available	**Tenancy:**	Default
VPC CIDR:	172.30.0.0/16	**DNS resolution:**	yes
DHCP options set:	dopt-24acf449	**DNS hostnames:**	yes
Route table:	rtb-f4d98391		
ClassicLink:	Disabled		

Subnets

The RDS Launch Wizard has created a subnet in our VPC for each Availability Zone (AZ) in the AWS Region. Although an RDS Aurora instance resides in a single AZ at any given time, it's necessary to have a subnet for at least three AZs to achieve high availability. If the AZ containing your RDS Aurora instance becomes unavailable, RDS will automatically provision a new instance in an available AZ that has a VPC subnet. If there is no VPC subnet for an AZ, RDS won't provision Aurora instances in it.

An Aurora database can have more than one instance. The set of instances that belong to the same Aurora database is called an Aurora cluster. An Aurora cluster can have one writer node and multiple reader nodes. Another reason for defining subnets for multiple AZs is to allow RDS Aurora instances in the same Aurora cluster to reside in different AZs. Each of those instances can reside in any AZ in the same region as long as VPC subnets are defined for the AZ. This provides you with options for load balancing database access over all available AZs in the region and to limit the impact to your business should one of the AZs become temporarily unavailable.

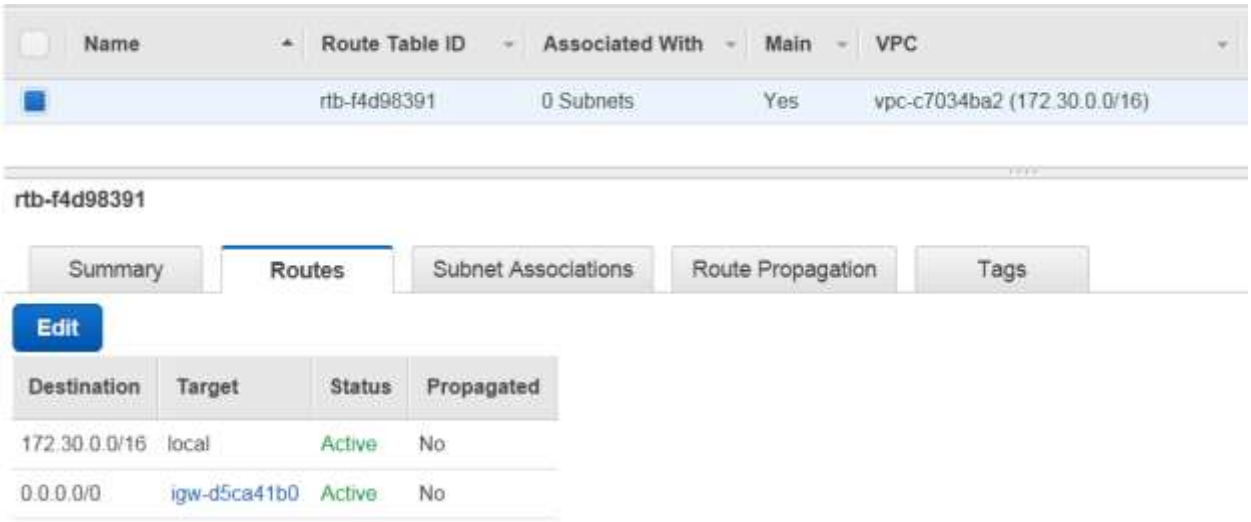


Internet Gateway

Because we specified that we wanted the RDS Aurora instance to be publicly accessible, the RDS Launch Wizard provisioned an Internet Gateway for our VPC. An AWS VPC Internet Gateway is horizontally-scaled, redundant, and highly-available and imposes no bandwidth constraints.

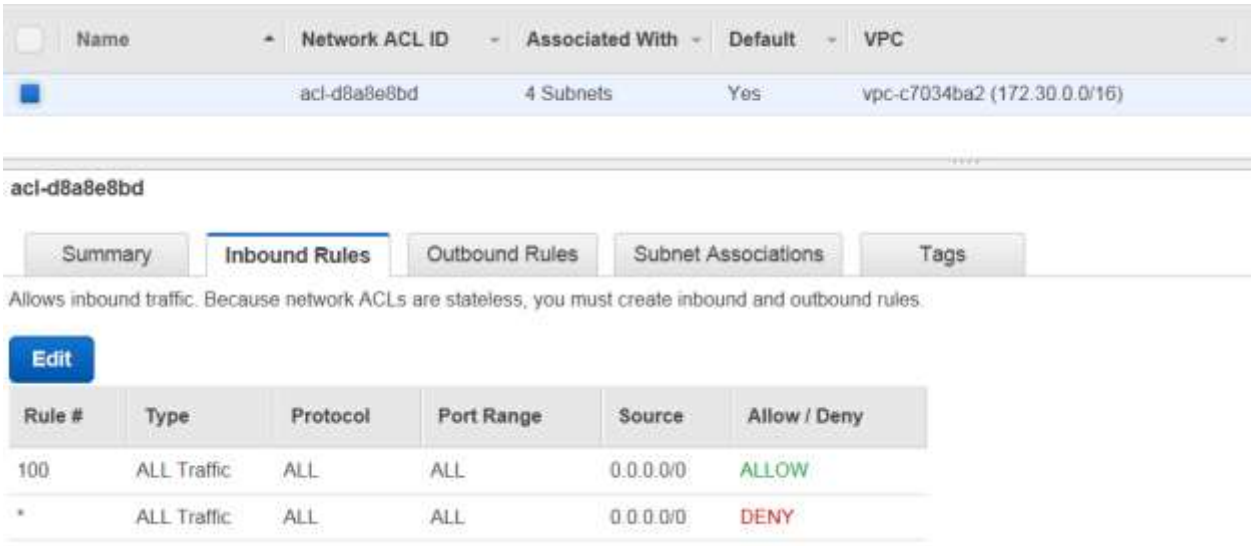
Route Table

The RDS Launch Wizard also created a Route Table in our VPC and configured it to route non-local network traffic to the Internet Gateway.



Network ACL

AWS VPC Network ACLs (Access Control Lists) allow you to specify what traffic is allowed and disallowed entering and exiting a subnet. The RDS Launch Wizard has created a Network ACL and associated it with all of the subnets in the VPC. The default is to allow all inbound and outbound traffic, but you can modify the rules to suit your requirements.



Name	Network ACL ID	Associated With	Default	VPC
	acl-d8a8e8bd	4 Subnets	Yes	vpc-c7034ba2 (172.30.0.0/16)

acl-d8a8e8bd

Summary | Inbound Rules | Outbound Rules | Subnet Associations | Tags

Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Edit

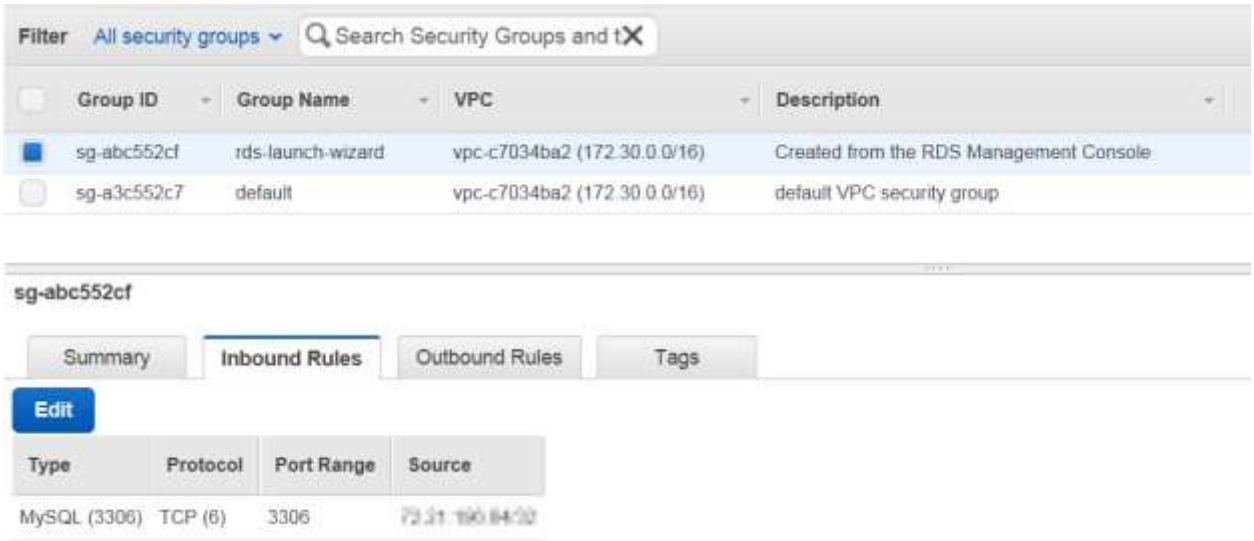
Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Security Groups

AWS VPC Security Groups specify which network traffic is allowed and disallowed at the instance level. The RDS Launch Wizard creates a Security Group that allows incoming traffic on the MySQL port (3306) for traffic originating from the system you accessed the AWS Console from. If you specified a port other than 3306 for the Aurora instance, that port will be used instead. Note that VPC Security Groups are managed from the VPC console rather than the RDS console.

If you plan to access your RDS Aurora instances from devices with different IP addresses, you will need to add rules to the Security Group to allow the inbound traffic.

If you're accessing RDS Aurora from a corporate network, you may need to create Inbound Rules for each of the IP address ranges used by your corporate network for Internet traffic. Engage your corporate network support team to determine which IP address ranges you should use.



Filter: All security groups | Search Security Groups and t

Group ID	Group Name	VPC	Description
sg-abc552cf	rds-launch-wizard	vpc-c7034ba2 (172.30.0.0/16)	Created from the RDS Management Console
sg-a3c552c7	default	vpc-c7034ba2 (172.30.0.0/16)	default VPC security group

sg-abc552cf

Summary | Inbound Rules | Outbound Rules | Tags

Edit

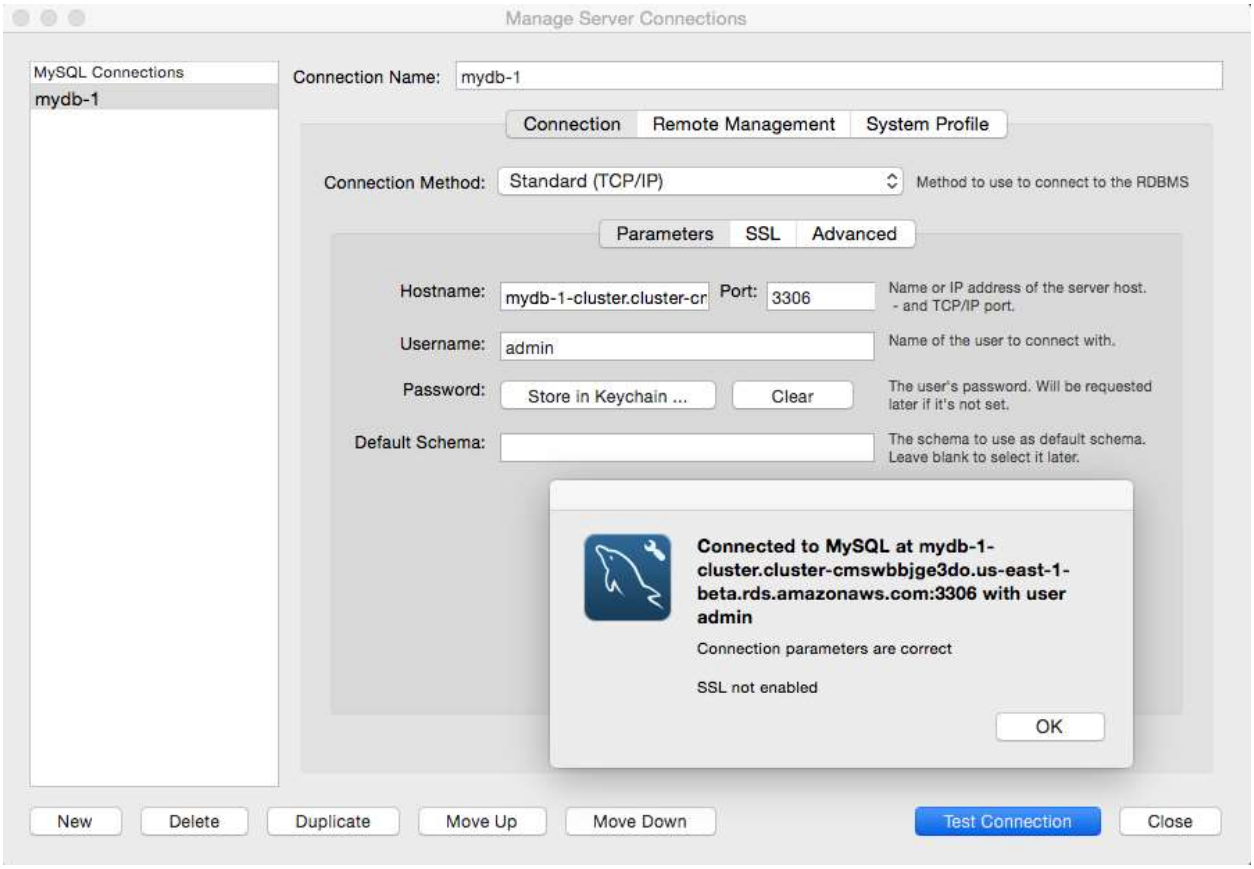
Type	Protocol	Port Range	Source
MySQL (3306)	TCP (6)	3306	[redacted]

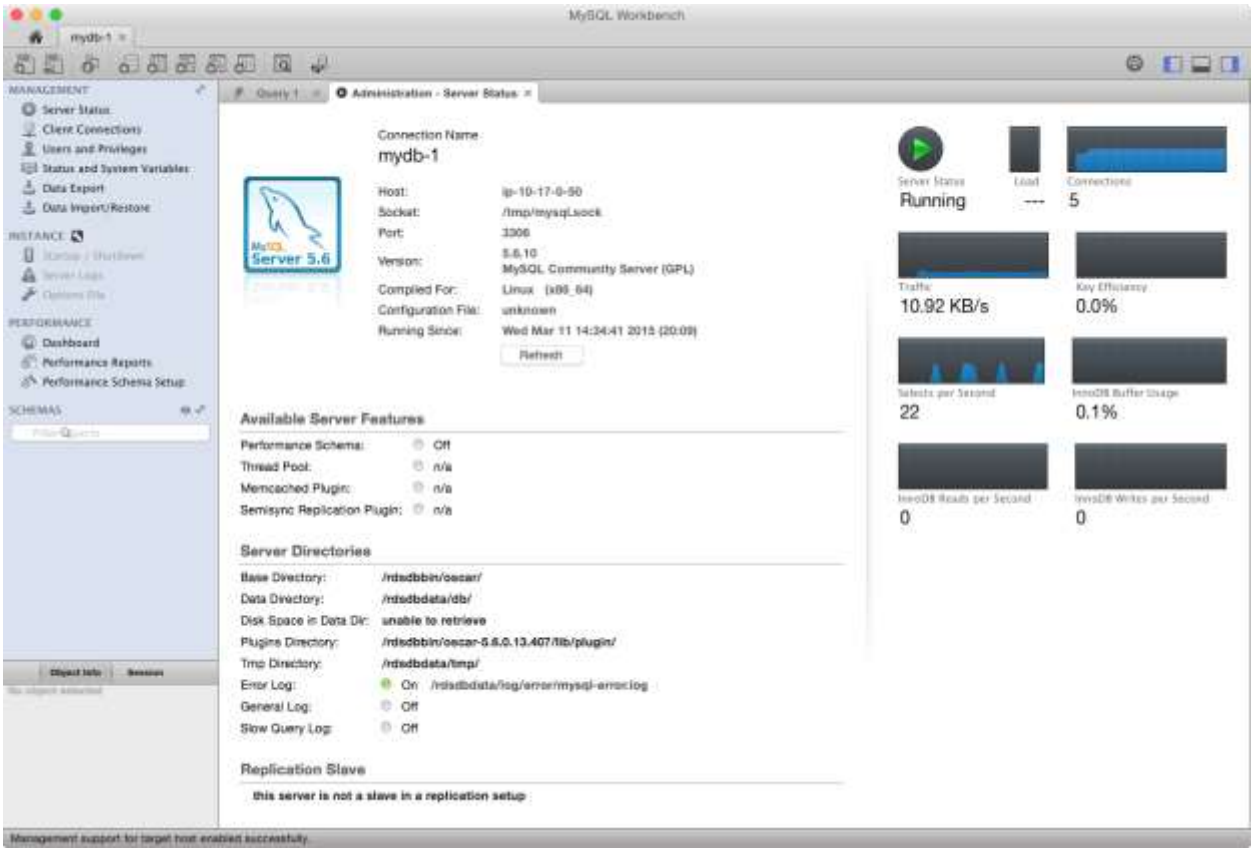
Getting Connected

Connecting Over the Internet

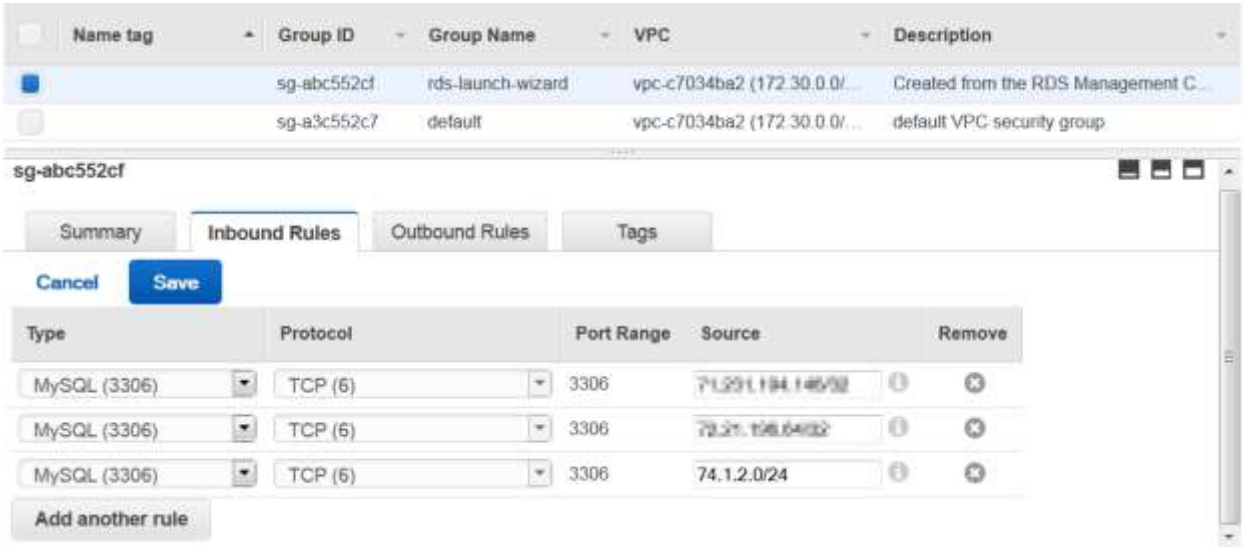
The previous sections describe a simple VPC configuration that enables connectivity to RDS Aurora instances over the Internet. If you were following along and used the RDS Launch Wizard to create the VPC as well as the RDS Aurora instance, the VPC will already be configured to accept incoming connections from the IP address used to run the RDS Launch Wizard. The endpoint (DNS name and port) that you will use to connect to the instance can be found on the Instances section of the RDS console as shown in the figure below.

The following figures show an example of connecting to an RDS Aurora instance using the MySQL Workbench utility.





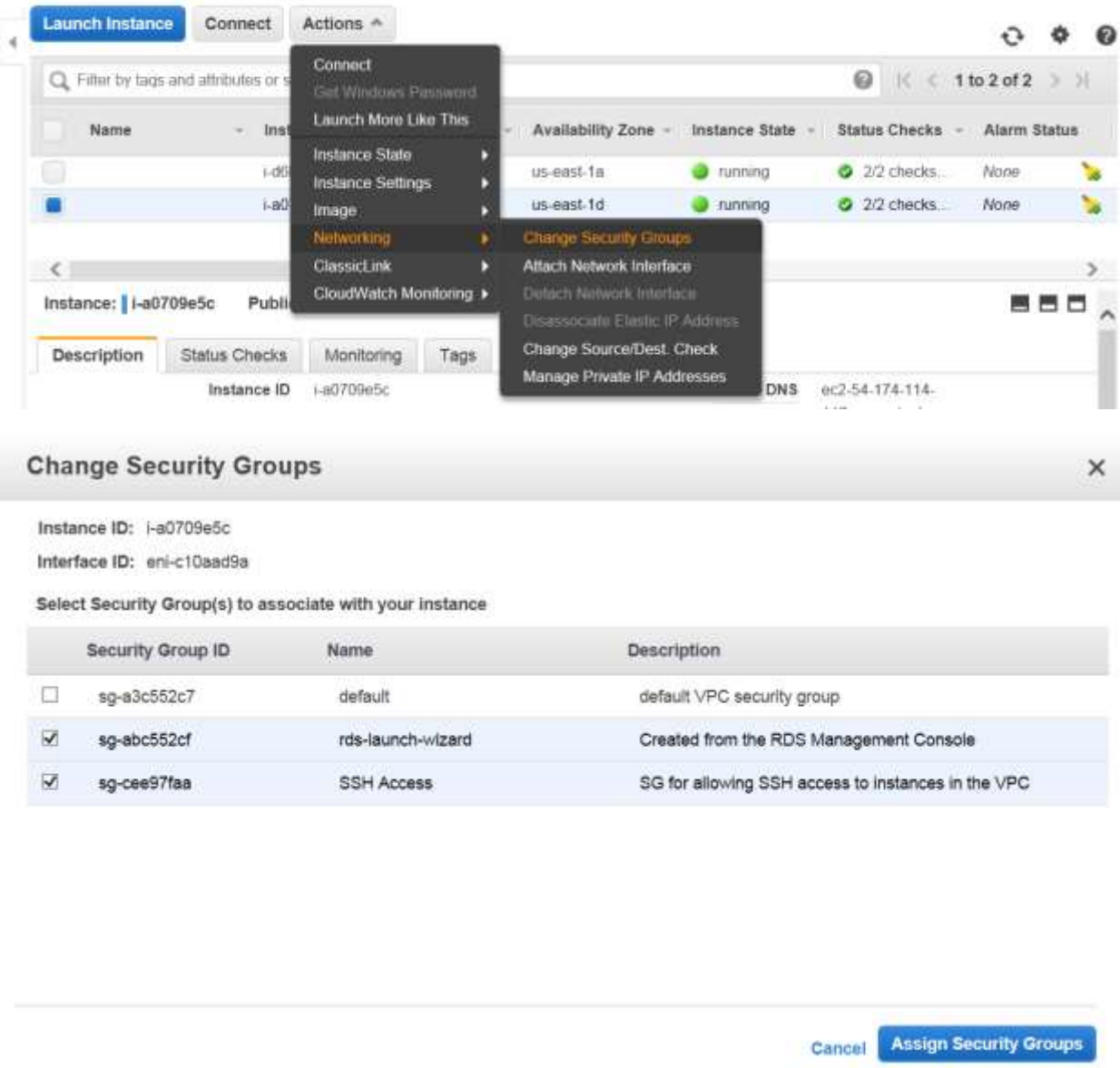
If you need to connect to RDS Aurora from other devices, you'll need to add their IP addresses or IP address ranges to the VPC Security Group.



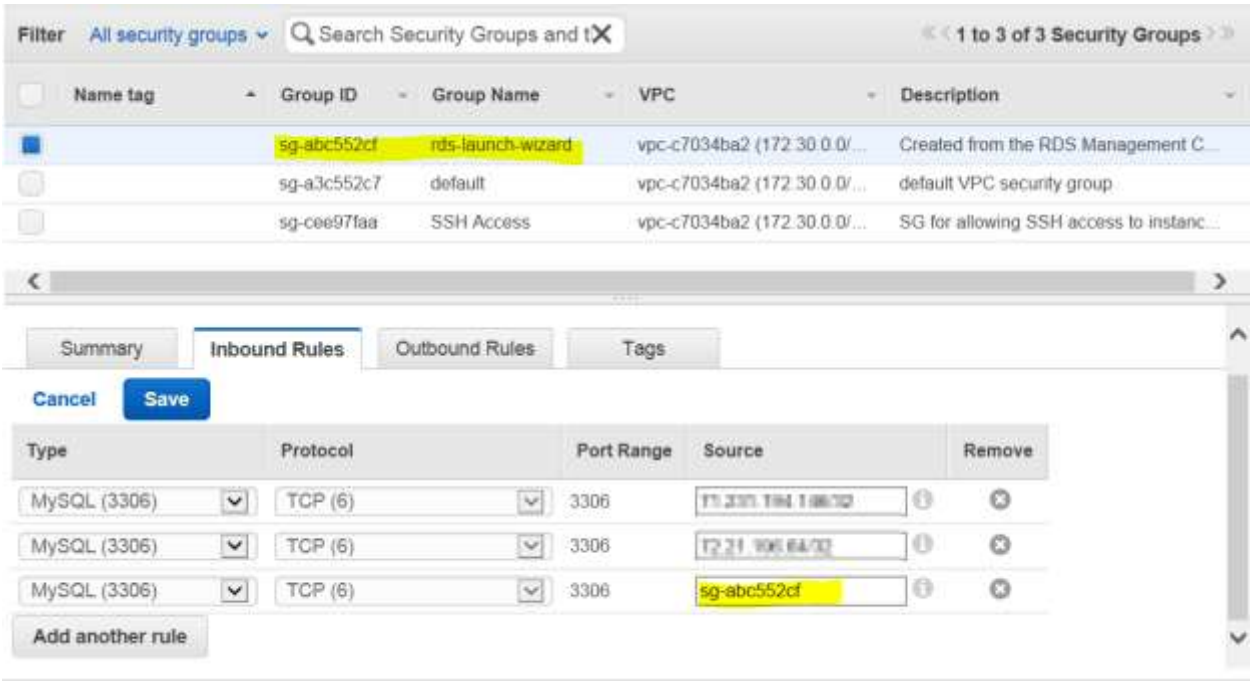
If you used a port other than the default (3306), use "Custom TCP Rule" for the Type and specify the appropriate port in the Port Range field.

Connecting from Within the Same VPC

In order to connect to your RDS Aurora instances from EC2 instances in the same VPC, you'll need to associate the EC2 instances with a VPC Security Group that allows access to the RDS Aurora Instances.



You may also need to add a rule to the VPC Security Group allowing traffic from instances associated with that group.



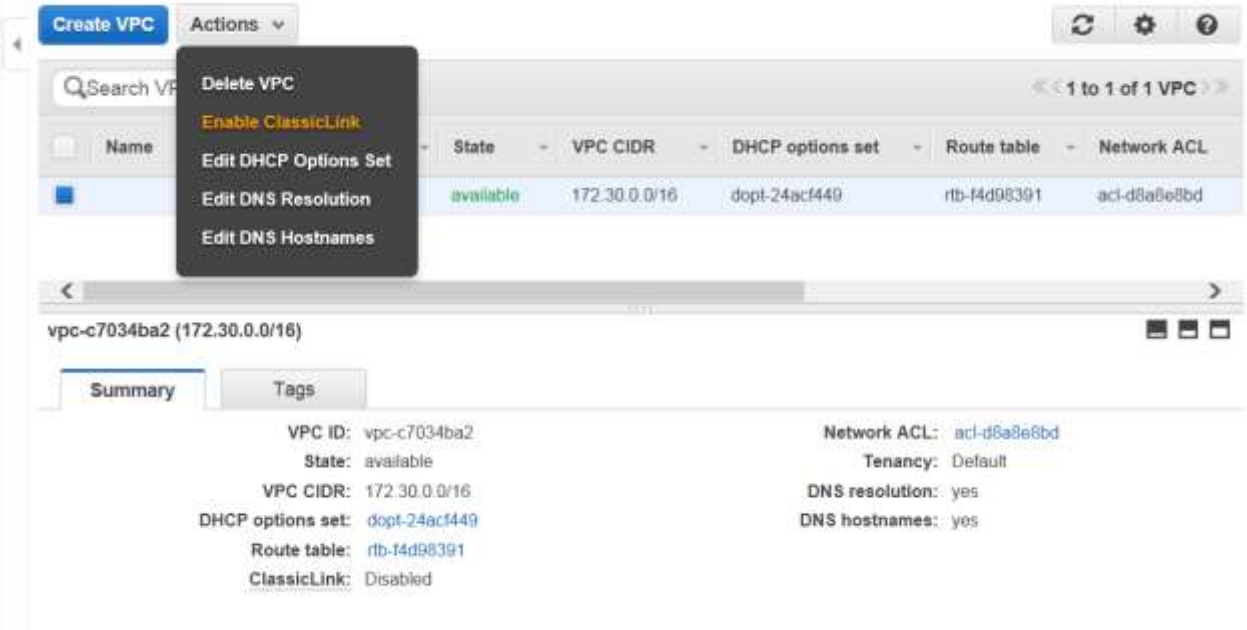
Once these changes are made, we can connect to the RDS Aurora instance:

```
mysql --user=admin -p --host=mydb-1-cluster.cluster-cmswbbjge3do.us-east-1-beta.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15881
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

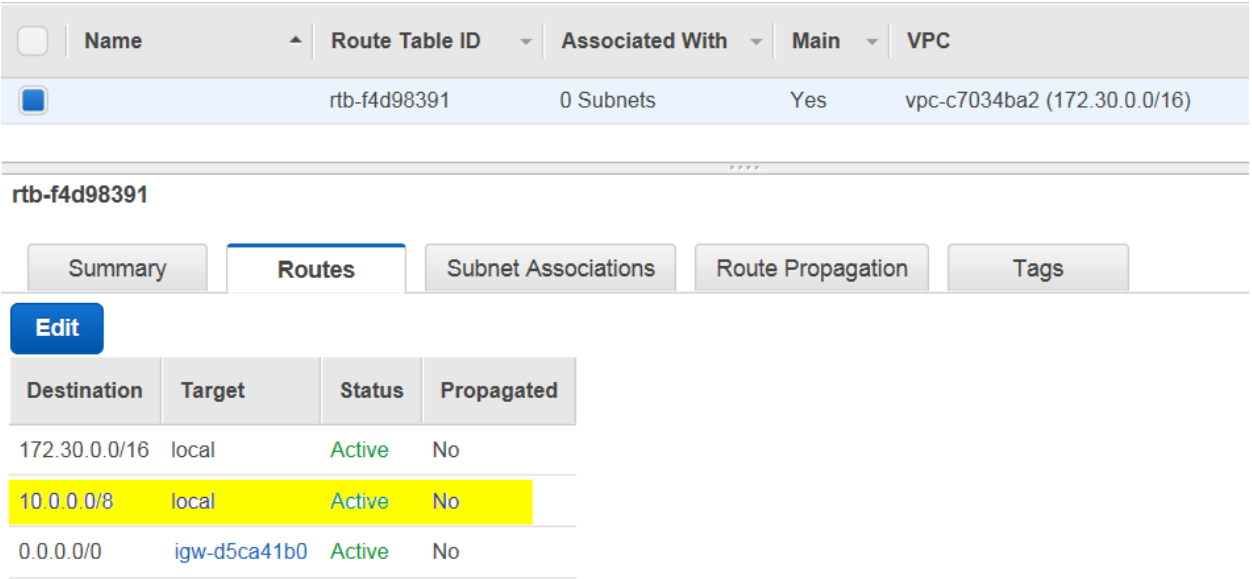
Note that the DNS hostname for the RDS Aurora instance resolves to its internal IP address when used within EC2 instances in the same VPC. This allows communication over the AWS inter-instance network, providing high bandwidth and low latency without incurring network bandwidth charges associated with communicating over the Internet.

Connecting from EC2-Classic

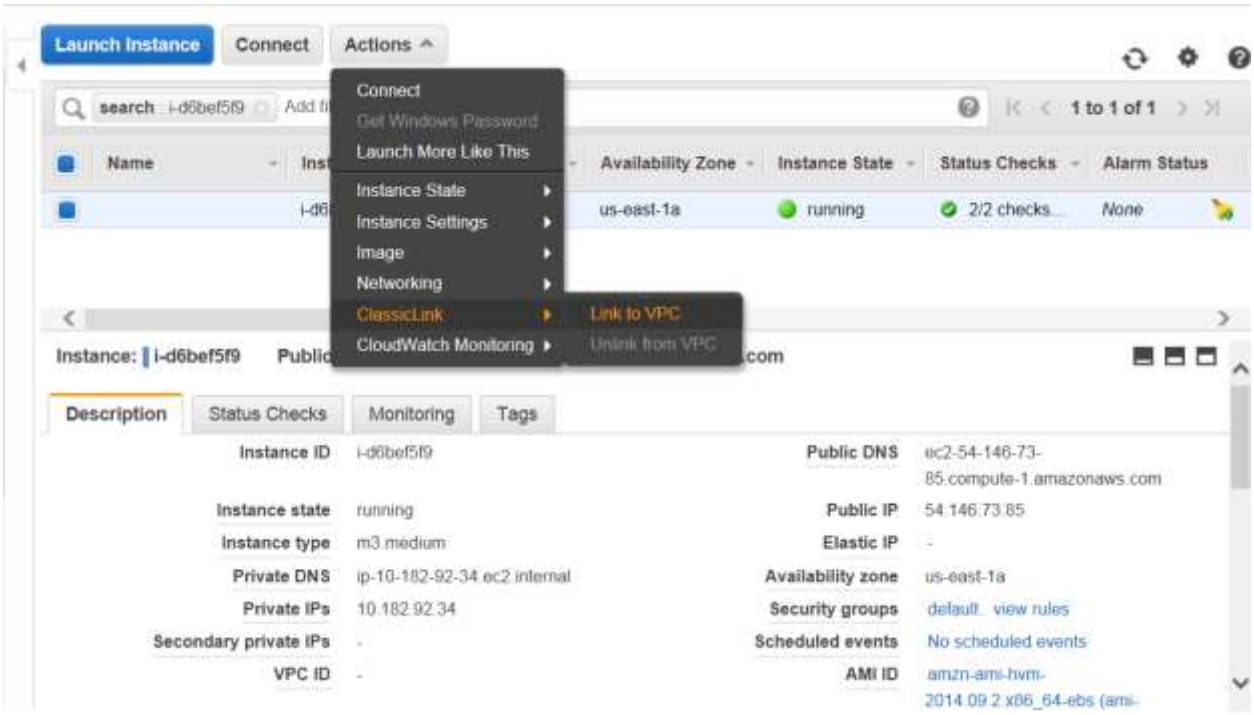
If you need to connect to RDS Aurora from EC2-Classic instances (instances that are not in a VPC) in the same region as the RDS Aurora instance, you can enable ClassicLink in the VPC and manage access using VPC Security Groups.



Enabling ClassicLink adds a new entry to the VPC Route Table to allow network traffic to the AWS inter-instance network.



After enabling ClassicLink on the VPC, we can now add EC2-Classic instances to the VPC Security Group that provides access to your RDS Aurora instances.



We can now connect to the RDS Aurora instance via its private IP address. Note that we cannot use the DNS name since that resolves to the public IP address from EC2-Classic instances and we haven't defined rules to allow communication with the public IP address from EC2-Classic. We could add rules to allow communication via the public IP address, but then we wouldn't be taking advantage of the benefits ClassicLink provides.

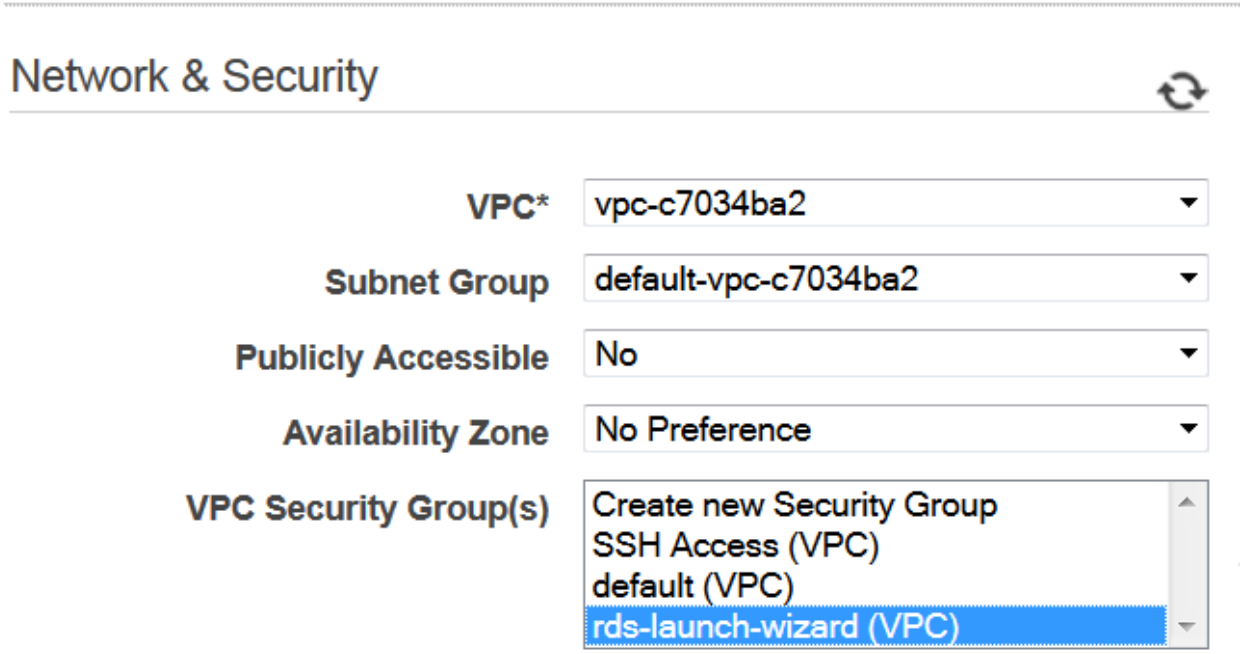
```
$ mysql --user=admin -p --host=172.30.3.168
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18162
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
```

```
affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Hiding RDS Aurora Instances from the Internet

For many use cases, allowing direct access to databases from the Internet is undesirable. In this section, we'll describe how to configure the VPC so that your RDS Aurora instances are not visible to the Internet and can be accessed only by EC2 instances or devices in the same VPC as the RDS Aurora instances. A common use case is public-facing web application and an RDS Aurora instance that is not publicly accessible.

The simplest way to hide RDS Aurora instances from the Internet is to simply specify "No" in the Publicly Accessible field when creating the instance. The instance will be created with a private IP address, but no public IP address. The only way to communicate with the instance is from within the VPC or EC2-Classic instances that are ClassicLinked to the VPC. In the example below, the RDS Aurora instance is being created in the same VPC we've been using previously and uses the VPC Security Group that we configured for public access. However, since the RDS Aurora instance has no public IP address, it cannot be reached from the Internet even though the VPC Security Group allows incoming traffic from the Internet.



Notice that we can use the DNS name for this RDS Aurora instance (within AWS in the same region) since it resolves only to the private IP address.

```
$ mysql --user=admin -p --host=my-private-db-2-cluster.cluster-cmswbbjge3do.us-east-1-
beta.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 187
Server version: 5.6.10 MySQL Community Server (GPL)
```

```
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Using an Existing VPC

If you already have an AWS VPC, you can provision RDS Aurora instances in it as well. RDS Aurora requires a minimum of three AWS Availability Zones for high availability, so you'll need at least three VPC subnets – one for each AZ. You'll also need to create an RDS DB Subnet Group so that RDS knows which subnets to use for your RDS Aurora instances.

Scenario

We will host a Web-facing app that accesses an RDS Aurora Cluster with one Writer and two Reader instances. The database instances should be accessible only by the Web app.

AWS VPC Configuration

In order to prevent access to the RDS Aurora instances from outside the AWS VPC, we will use both VPC Subnet Groups and AWS Network ACLs to limit access to the databases.

The VPC is in a region with four Availability Zones, so the it has been configured with four private and four public subnets – one private and one public for each of the four Availability Zones. The databases will reside in the private subnets while the Web app resides in the public subnets.

Name	Subnet ID	State	CIDR	Availability Zone	Route Table	Auto-assign Public
priv-1a	subnet-93551da9	available	172.30.1.0/24	us-east-1a	rtb-6de3c608 | p...	No
priv-1b	subnet-6b5a3e1c	available	172.30.2.0/24	us-east-1b	rtb-6de3c608 | p...	No
priv-1d	subnet-1ab33643	available	172.30.3.0/24	us-east-1d	rtb-6de3c608 | p...	No
priv-1e	subnet-ab2b8980	available	172.30.4.0/24	us-east-1e	rtb-6de3c608 | p...	No
pub-1a	subnet-09561e33	available	172.30.5.0/24	us-east-1a	rtb-2fefca4a | pu...	Yes
pub-1b	subnet-0d5a3e7a	available	172.30.6.0/24	us-east-1b	rtb-2fefca4a | pu...	Yes
pub-1d	subnet-dfb33686	available	172.30.7.0/24	us-east-1d	rtb-2fefca4a | pu...	Yes
pub-1e	subnet-f22b89d9	available	172.30.8.0/24	us-east-1e	rtb-2fefca4a | pu...	Yes

There are two Route Tables – one has a route to an AWS VPC Internet Gateway for the public subnets and one that has no external routing for the private subnets. You can see the Route Table associations for each subnet in the figure above.

Name	Route Table ID	Associated With	Main	VPC
priv-route	rtb-6de3c608	0 Subnets	Yes	vpc-d9a2eabc (172.30.0.0/16) | We...
pub-route	rtb-2fefca4a	4 Subnets	No	vpc-d9a2eabc (172.30.0.0/16) | We...

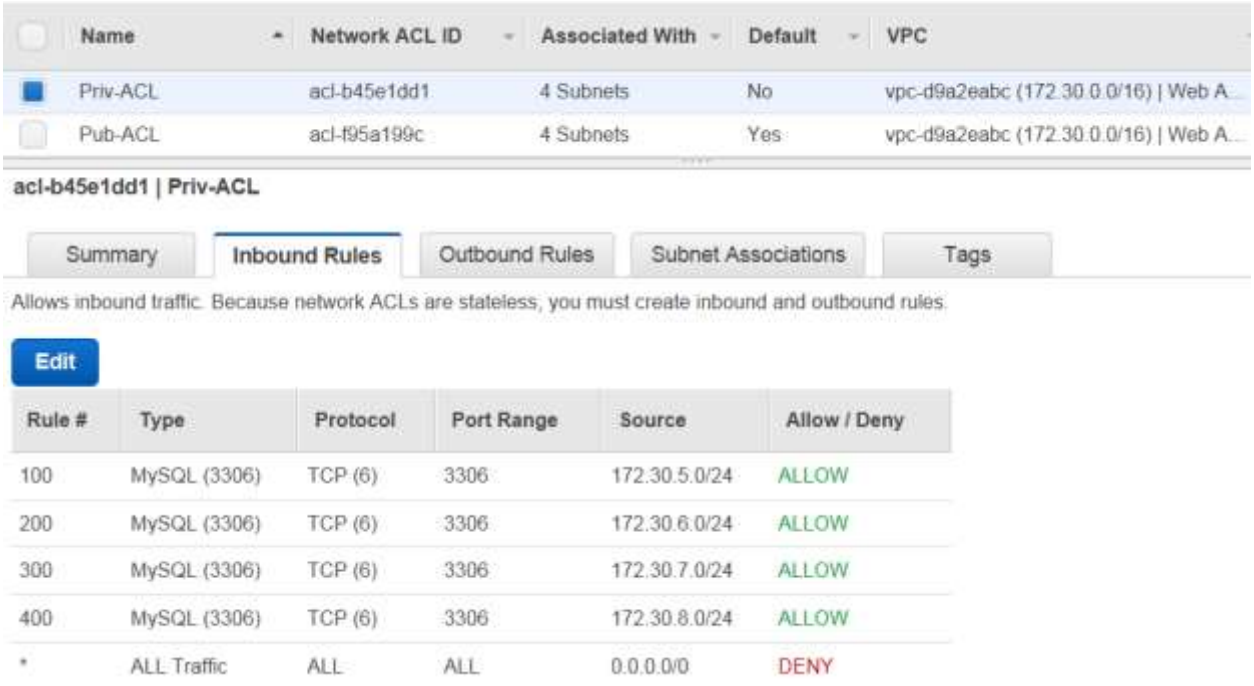
rtb-2fefca4a | pub-route

Summary | **Routes** | Subnet Associations | Route Propagation | Tags

Edit

Destination	Target	Status	Propagated
172.30.0.0/16	local	Active	No
0.0.0.0/0	igw-c248c3a7	Active	No

The Network ACLs for the private subnets are configured to allow incoming traffic only from the public subnets and only on the port used by the RDS Aurora instances.



Name	Network ACL ID	Associated With	Default	VPC
Priv-ACL	acl-b45e1dd1	4 Subnets	No	vpc-d9a2eabc (172.30.0.0/16) | Web A...
Pub-ACL	acl-f95a199c	4 Subnets	Yes	vpc-d9a2eabc (172.30.0.0/16) | Web A...

acl-b45e1dd1 | Priv-ACL

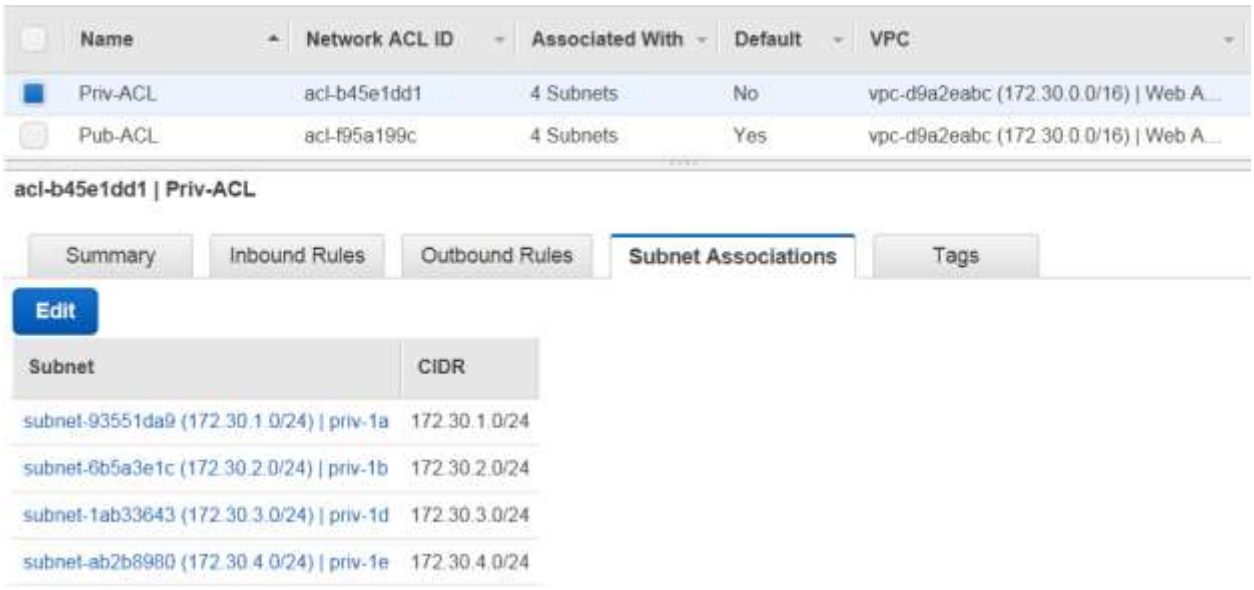
Summary | Inbound Rules | Outbound Rules | Subnet Associations | Tags

Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Edit

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	MySQL (3306)	TCP (6)	3306	172.30.5.0/24	ALLOW
200	MySQL (3306)	TCP (6)	3306	172.30.6.0/24	ALLOW
300	MySQL (3306)	TCP (6)	3306	172.30.7.0/24	ALLOW
400	MySQL (3306)	TCP (6)	3306	172.30.8.0/24	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Each of the four private subnets is associated with this Network ACL.



Name	Network ACL ID	Associated With	Default	VPC
Priv-ACL	acl-b45e1dd1	4 Subnets	No	vpc-d9a2eabc (172.30.0.0/16) | Web A...
Pub-ACL	acl-f95a199c	4 Subnets	Yes	vpc-d9a2eabc (172.30.0.0/16) | Web A...

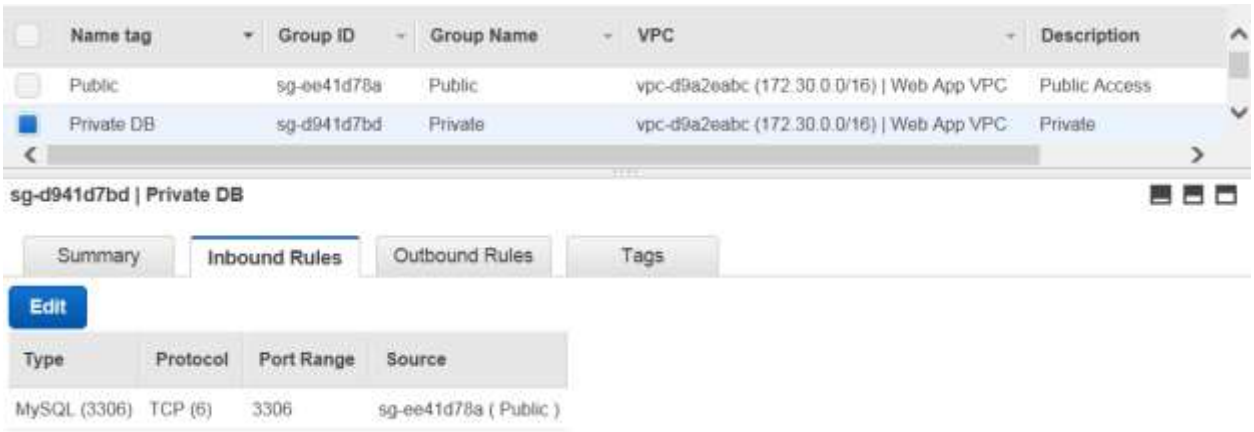
acl-b45e1dd1 | Priv-ACL

Summary | Inbound Rules | Outbound Rules | Subnet Associations | Tags

Edit

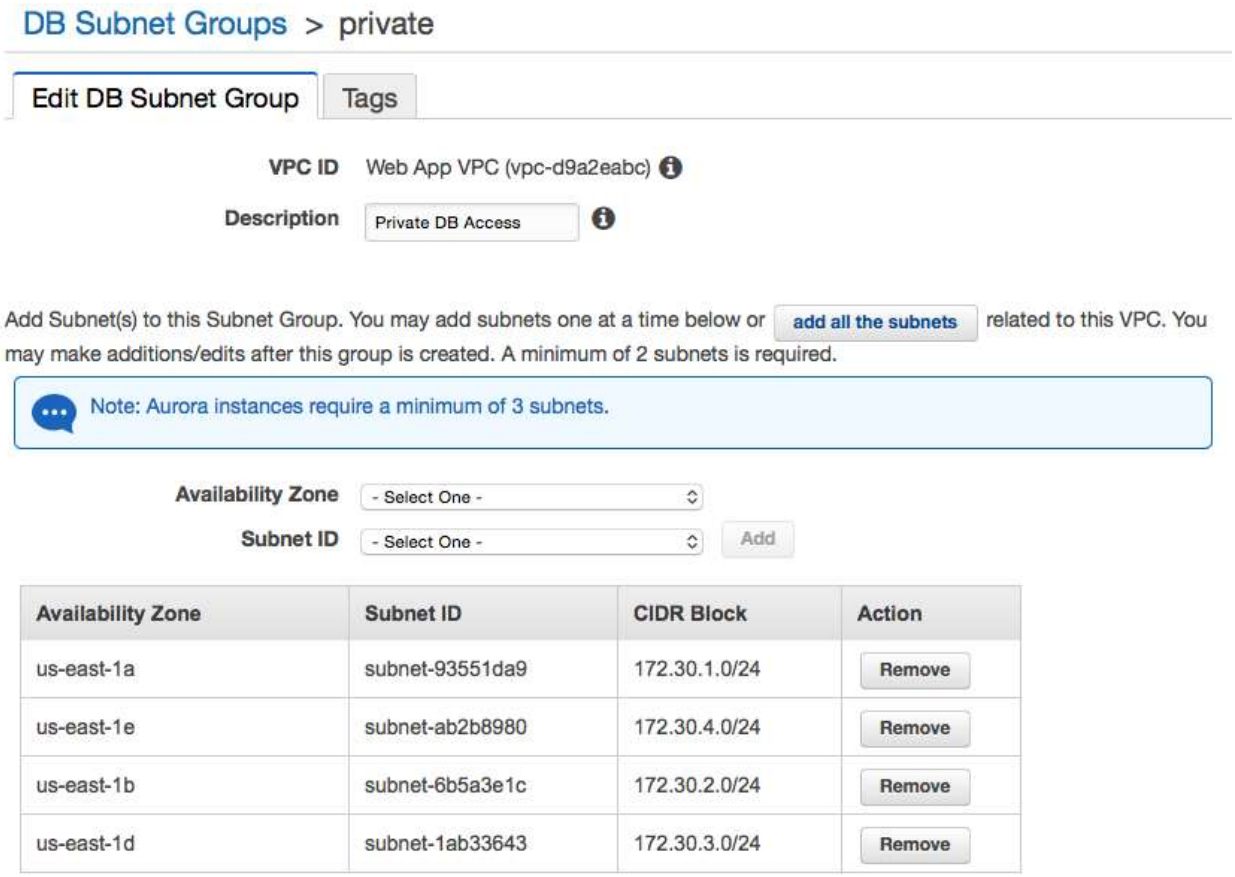
Subnet	CIDR
subnet-93551da9 (172.30.1.0/24) | priv-1a	172.30.1.0/24
subnet-6b5a3e1c (172.30.2.0/24) | priv-1b	172.30.2.0/24
subnet-1ab33643 (172.30.3.0/24) | priv-1d	172.30.3.0/24
subnet-ab2b8980 (172.30.4.0/24) | priv-1e	172.30.4.0/24

There are also two VPC Security Groups – one for database use and one for Web app use. The inbound rule for the private Security Group accepts traffic only on the database port and only from instances that are associated with the public VPC Security Group. The Web app instances are associated with the public VPC Security Group.



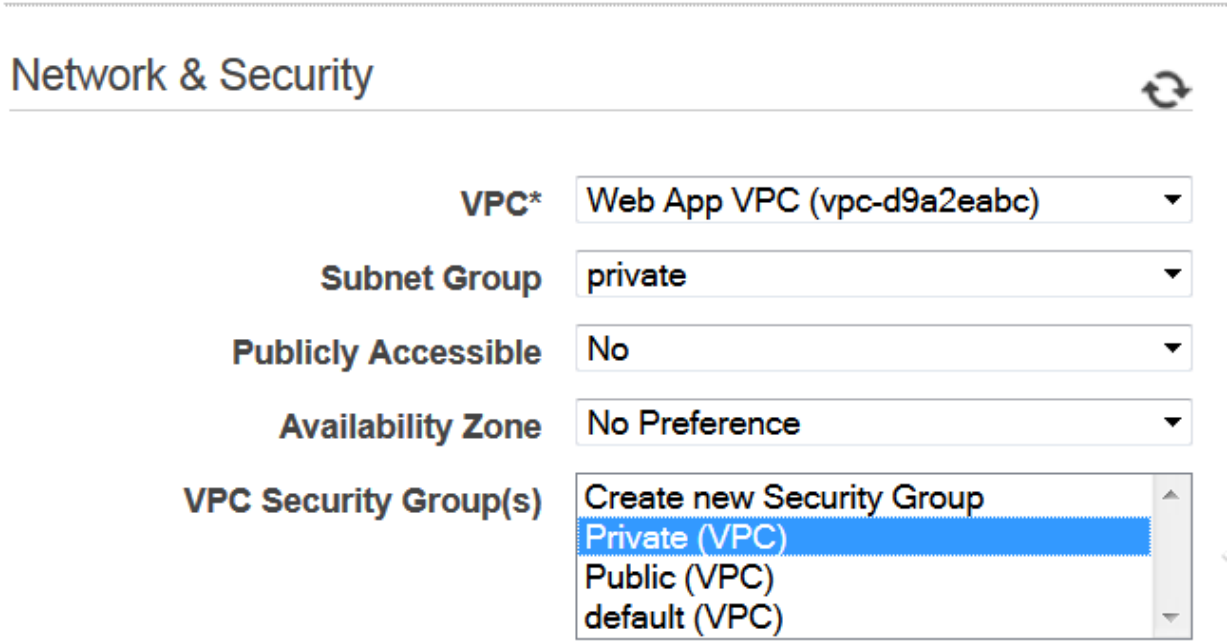
RDS DB Subnet Group Configuration

Before we can create RDS Aurora instances in this VPC, we need to tell RDS which VPC Subnets to use. In this example, an RDS DB Subnet Group named “private” was created that maps to each of the VPC private subnets where we want RDS to provision our RDS Aurora instances.

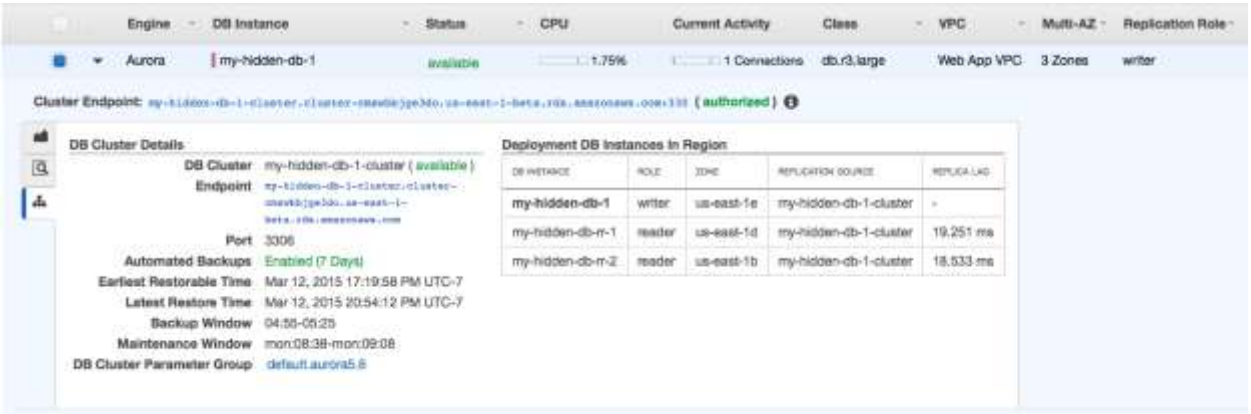


RDS Aurora Cluster Creation

Now we can create RDS Aurora instances in the VPC. The Subnet Group is set to use the "private" RDS DB Subnet Group, Publicly Accessible is set to "No" so that the RDS Aurora instance will have a private IP address only, and the VPC Security Group field is set to the "Private" VPC Subnet Group.



Here's the cluster after creating two Aurora Replicas. Notice that each instance is in a different AZ.



Accessing the RDS Aurora Cluster Instances

Using an AWS EC2 instance created in the same VPC in one of the public subnets, we can connect to each of the instances in the RDS Aurora Cluster, but they will not be accessible from anywhere outside the VPC.

First, we'll create a new database and table on the Writer instance.

```
[ec2-user@ip-172-30-7-75 ~]$ mysql --user=admin -p --host=my-hidden-db-1-cluster.cluster-
cmswbbjge3do.us-east-1-beta.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2612
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database mydb;
Query OK, 1 row affected (0.02 sec)

mysql> create table mydb.myuser as select * from mysql.user;
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Now we'll read the table from one of the Reader instances.

```
mysql> [ec2-user@ip-172-30-7-75 ~]$ mysql --user=admin -p --host=my-hidden-db-rr-
1.cmswbbjge3do.us-east-1-beta.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 176
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select count(*) from mydb.myuser;
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.01 sec)
```

```
mysql> exit
Bye
```

And again from the other Reader instance.

```
[ec2-user@ip-172-30-7-75 ~]$ mysql --user=admin -p --host=my-hidden-db-rr-2.cmswbbjge3do.us-east-
1-beta.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 155
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select count(*) from mydb.myuser;
+----------+
| count(*) |
+----------+
|        3 |
+----------+
1 row in set (0.01 sec)

mysql>
```

Conclusion

In this document, you learned how to connect to RDS Aurora instances from on-premises equipment, EC2-Classic instances using AWS ClassicLink, and how to hide RDS Aurora instances from the Internet while allowing access from Web-facing apps.

Further Reading

AWS RDS Aurora: http://aws.amazon.com/rds/aurora/
AWS VPC: http://aws.amazon.com/vpc/