Using Cyral with Security Groups to control access to AWS database repositories

Introduction

This document introduces the concept of Security Groups in Amazon Web Services (AWS) and provides detailed instructions on how to leverage it in conjunction with Cyral to configure and build out data layer security for all your cloud data endpoints.

What are AWS Security Groups?

AWS Security Groups (SG) provide security at the protocol and port access level. Each security group — working much the same way as a firewall — contains a set of rules that filter traffic coming into and out of an instance. Unlike network access control lists (NACLs), there are no “Deny” rules. If there is no rule that explicitly permits a particular data packet, it will be dropped.

What is Cyral?

Cyral helps companies observe, protect and control their data in the cloud. Cyral intercepts every request to all data endpoints and captures its application context without impacting latency or scalability. Companies can now (1) detect and react to threats, (2) log accesses for audit and compliance, and (3) centrally manage authorization, all while working in conjunction with their existing investments in orchestration, identity management, SIEM and monitoring. Cyral adopts an API-first model and the platform seamlessly integrates with a DevOps and "shift left" approach without requiring any app or deployment changes.

Why use SG with Cyral?

Enterprises are moving database and warehousing operations to the cloud at an accelerating pace. Unfortunately, this comes with a new set of challenges. The complexity of public cloud offerings makes it difficult to know whether the data layer is secure. Cyral solves this by enabling observability, protection and control over all cloud data endpoints consistently and at any scale.

SG enforces who can interact with an endpoint. By leveraging it in conjunction with Cyral, one can make sure all interactions with a cloud data endpoint happen only through Cyral. This can be done with a few easy steps, either in the AWS UI or via CLI. The result is strict enforcement of data layer security and data

activity monitoring as shown in Figure 1. Users and services no longer directly access the RDS endpoints. Instead they only interact via Cyral.

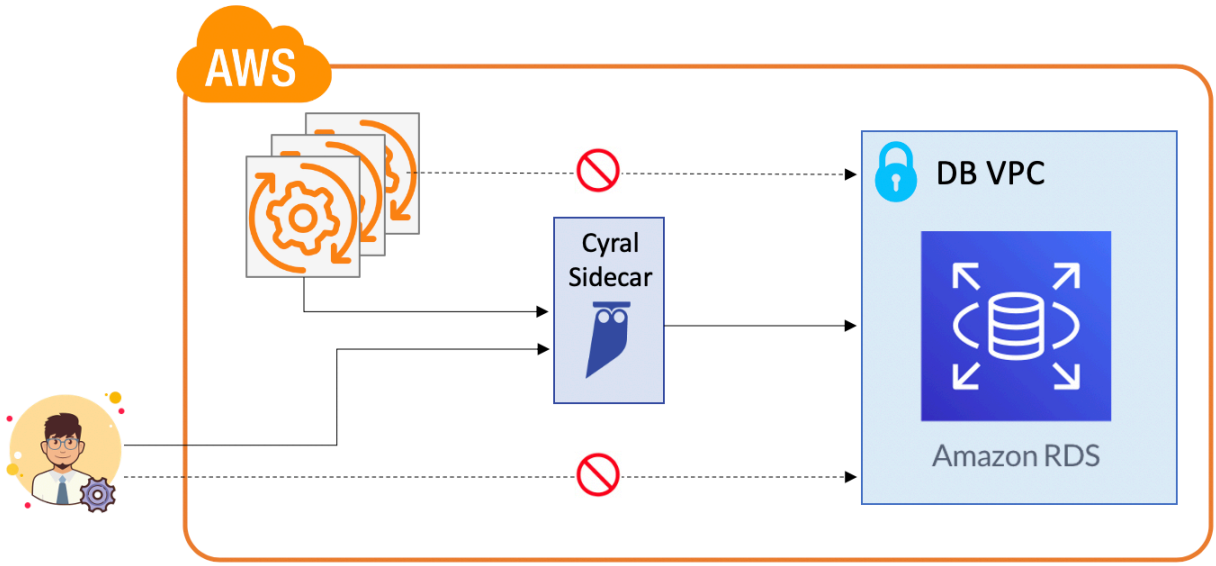


Figure 1

For the rest of this document, we will use RDS as an example of the cloud data endpoint that needs to be secured. The process of securing other data endpoints follows the same steps. If you have any specific questions about your needs, please contact our team at product@cyral.com.

How to transparently force all existing applications to use Cyral

You have applications interacting with your RDS instance, which you want to secure without disrupting or modifying these applications. This section explains how to transparently route applications through Cyral in order to accomplish this. We consider the two different ways in which applications may be communicating with the RDS instance.

Scenario 1: Using CNAME

In the example below, “invoices.hhiu.cyral.com” is a CNAME for the RDS endpoint and apps use the “invoices.hhiu.cyral.com” URL to get to the actual RDS instance. You can find the mapping of CNAMEs to actual endpoints in Route 53.

invoices.hhiu.cyral.com.	CNAME	invoices.cazko2wnefdm.us-west-1.rds.amazonaws.com	-	300

Also using Route 53, change the CNAME to now map to the Cyral sidecar (running on IP address 54.151.86.198) as shown below.

Edit Record Set

Name: invoices .hhiu.cyral.com.

Type: CNAME – Canonical name

Alias: Yes No

TTL (Seconds): 300 1m 5m 1h 1d

Value: 54.151.86.198

The domain name that you want to resolve to instead of the value in the Name field.
Example:
www.example.com

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record. Learn More

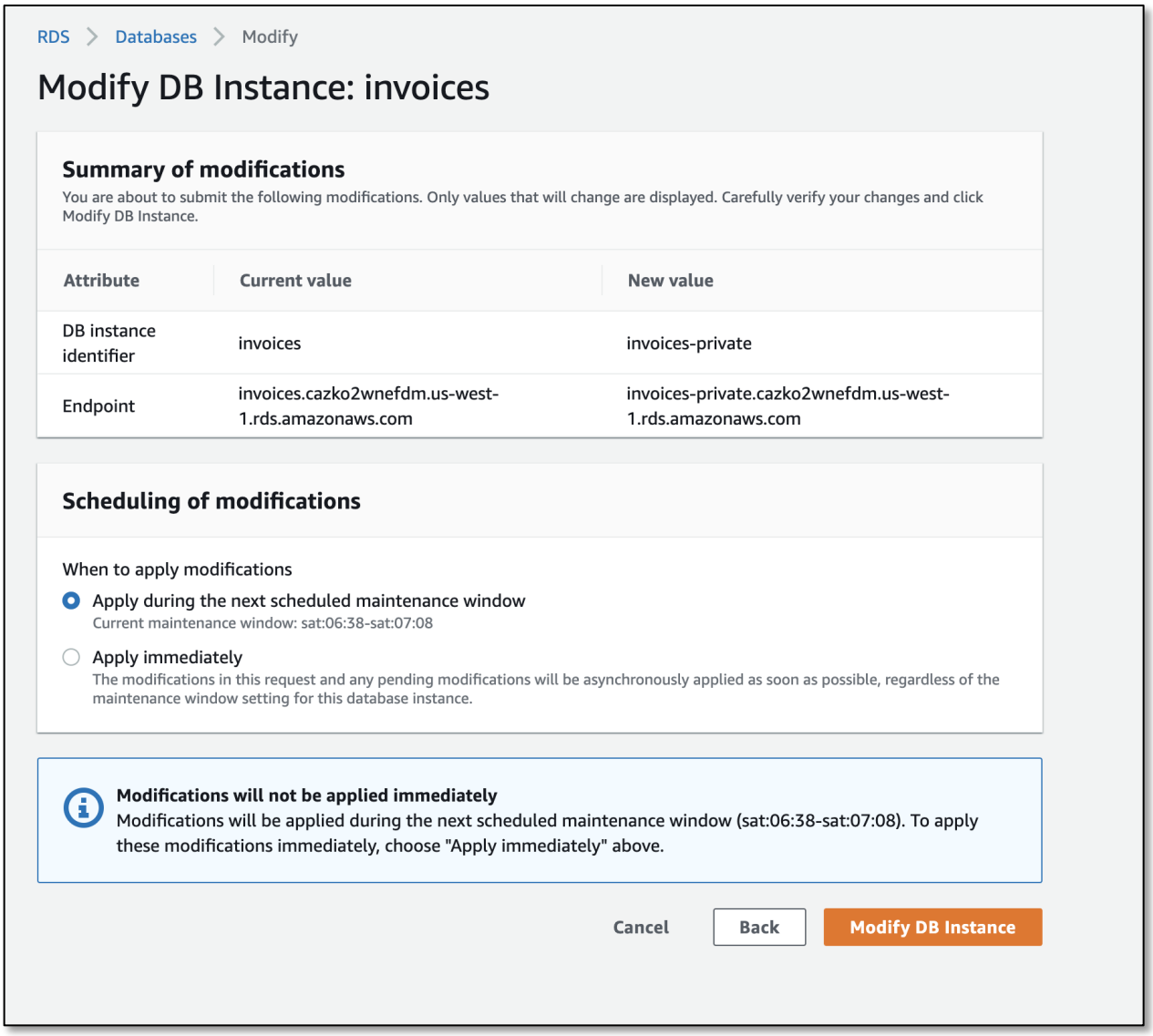
Note how the alias for the CNAME has changed.

invoices.mbhat.cyral.com.	CNAME	54.151.86.198	-	-

This now enforces that all application traffic to be routed through Cyral.

Scenario 2: Using RDS URL

During a maintenance window rename the RDS URL to a private one only known to Cyral. You can do this by choosing your database from the RDS console and modifying the RDS instance identifier. A new endpoint is automatically generated. You can choose when you want the change to take effect. Below we've chosen to do this at the next maintenance window. Use Route53 to map the old URL to Cyral.



The steps above are necessary but not sufficient to ensure that all traffic goes through the data security layer. A user with database credentials may continue to directly go to the RDS instance. To prevent this and to enforce all accesses go through Cyral we will now configure security groups.

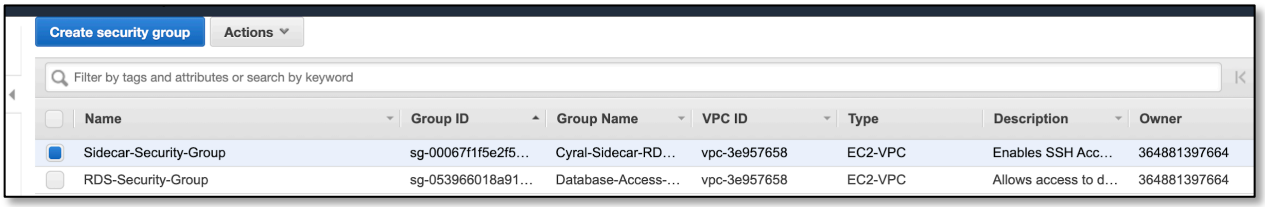
How to easily configure SG with Cyral?

This section will cover how to configure SG with Cyral in a few easy steps as follows:

- Create a security group for RDS
- Add the RDS instance to above security group
- Create a security group for Cyral
- Add an inbound rule and an outbound rule for the RDS security group that only allows traffic to and from the Cyral security group

Let's imagine a Cyral deployment with a sidecar named Cyral-Sidecar-RDS that is protecting invoices.hhiu.cyral.com

We begin by creating a security group for Cyral called Sidecar-Security-Group and a security group for RDS called RDS-Security-Group as shown below. To create a security group Open the Amazon VPC console at https://console.aws.amazon.com/vpc/ and choose Security Groups



Add the RDS instance to the RDS-Security-Group as shown below

RDS > Databases > Modify

Modify DB Instance: invoices

Summary of modifications

You are about to submit the following modifications. Only values that will change are displayed. Carefully verify your changes and click Modify DB Instance.

Attribute	Current value	New value
Security group	default	RDS-Security-Group

Scheduling of modifications

When to apply modifications

- (●) Apply during the next scheduled maintenance window
  Current maintenance window: sat:06:38-sat:07:08
- ( ) Apply immediately
  The modifications in this request and any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this database instance.

Modifications will not be applied immediately
Modifications will be applied during the next scheduled maintenance window (sat:06:38-sat:07:08). To apply these modifications immediately, choose "Apply immediately" above.

Cancel | Back | Modify DB Instance

Similarly, add the Cyral sidecar to Sidecar-Security-Group.

Next create an inbound rule for the RDS-Security-Group. Choose Type to be “MySQL/Aurora” (Note: Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL.) This will default the port to 3306. Add the Group ID for the Sidecar-Security-Group as the Source. Finally, add a detailed description and click on “Save Rules” as shown below.

Security Groups > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Type	Protocol	Port Range	Source		Description
MYSQL/Aurora	TCP	3306	Custom	sg-00067f1f5e2f59174	Only Sidecar can interact with RDS

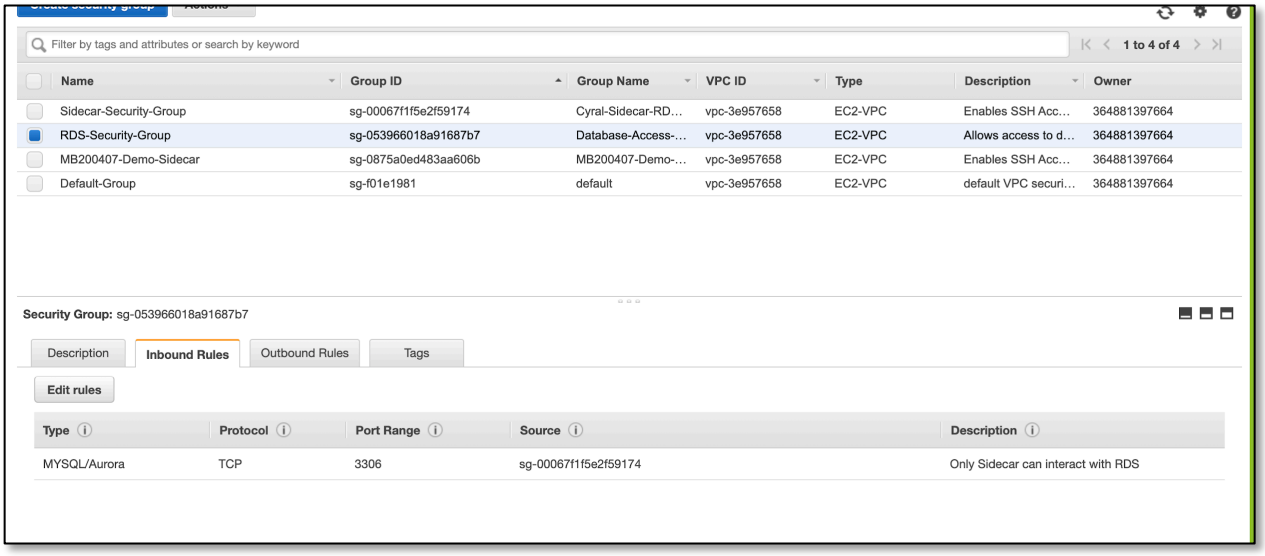
Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

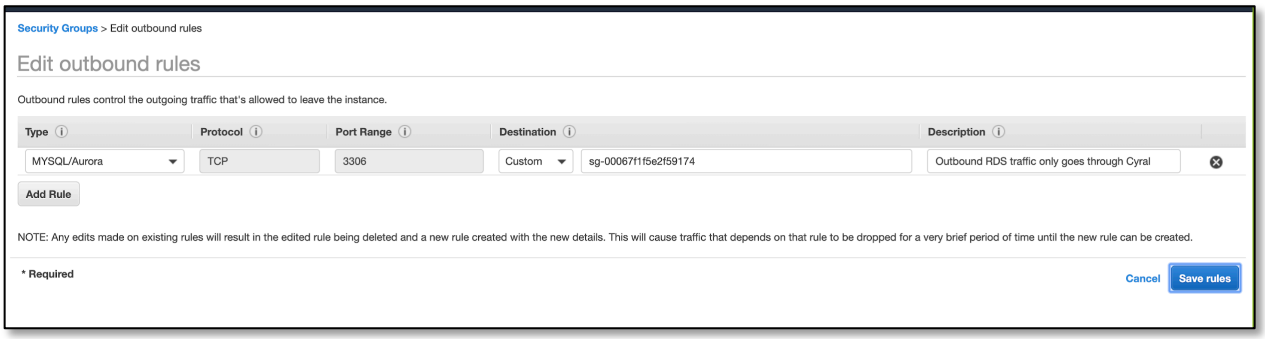
* Required

Cancel | Save rules

This enforces that all inbound traffic to the MySQL RDS instance comes only through Cyral.



Do the same for outbound rules thereby enforcing that all outbound traffic from the MySQL RDS instance only goes through Cyral.



Create security group | Actions

Filter by tags and attributes or search by keyword — 1 to 4 of 4

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
Sidecar-Security-Group	sg-00067f1f5e2f59174	Cyral-Sidecar-RD…	vpc-3e957658	EC2-VPC	Enables SSH Acc…	364881397664
RDS-Security-Group	sg-053966018a91687b7	Database-Access-…	vpc-3e957658	EC2-VPC	Allows access to d…	364881397664
MB200407-Demo-Sidecar	sg-0875a0ed483aa606b	MB200407-Demo-…	vpc-3e957658	EC2-VPC	Enables SSH Acc…	364881397664
Default-Group	sg-f01e1981	default	vpc-3e957658	EC2-VPC	default VPC securi…	364881397664

Security Group: sg-053966018a91687b7

Description | Inbound Rules | Outbound Rules | Tags

Edit rules

Type	Protocol	Port Range	Destination	Description
MYSQL/Aurora	TCP	3306	sg-00067f1f5e2f59174	Outbound RDS traffic only goes through Cyral

We want anyone to be able to access the Cyral sidecar and so we set an inbound rule allowing all traffic as shown below.

Create security group | Actions

Filter by tags and attributes or search by keyword — 1 to 4 of 4

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
Sidecar-Security-Group	sg-00067f1f5e2f59174	Cyral-Sidecar-RD…	vpc-3e957658	EC2-VPC	Enables SSH Acc…	364881397664
RDS-Security-Group	sg-053966018a91687b7	Database-Access-…	vpc-3e957658	EC2-VPC	Allows access to d…	364881397664
MB200407-Demo-Sidecar	sg-0875a0ed483aa606b	MB200407-Demo-…	vpc-3e957658	EC2-VPC	Enables SSH Acc…	364881397664
Default-Group	sg-f01e1981	default	vpc-3e957658	EC2-VPC	default VPC securi…	364881397664

Security Group: sg-00067f1f5e2f59174

Description | Inbound Rules | Outbound Rules | Tags

Edit rules

Type	Protocol	Port Range	Source	Description
Custom TCP Rule	TCP	27000 - 27009	0.0.0.0/0	
Custom TCP Rule	TCP	3300 - 3309	0.0.0.0/0	
Custom TCP Rule	TCP	5430 - 5439	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	
Custom TCP Rule	TCP	440 - 449	0.0.0.0/0	
All ICMP - IPv4	All	N/A	0.0.0.0/0	

And, we are done! This enforces all traffic in and out of the MySQL RDS instance to go through Cyral and enables you to observe, protect and control your cloud data layer.

If one attempts to bypass the data security layer and tries to connect to the RDS instance directly the operation will error out as shown below

```
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~
➜  ~ mysql -h invoices.hhiu.cyral.com -P 3306 -u admin -p
Enter password:
ERROR 2003 (HY000): Can't connect to MySQL server on 'invoices.hhiu.cyral.com' (60)
➜  ~
```

About Cyral:

Cyral enables companies to guard against data exfiltration without requiring any agenda or app modifications. The Cyral service is easy to use and plugs seamlessly into an infrastructure-as-code framework. The Cyral founding team has deep expertise in building databases, compilers, and proxy-based distributed services, and is backed by Redpoint, A.Capital and Costanoa.