

AWS DMS Monitoring Runbook

Runbook –

<https://github.com/aws-samples/aws-dms-monitoring-runbook>

This runbook deploys following monitoring -

1. Centralized Amazon CloudWatch dashboard to review resource consumption (e.g., CPU, Memory, Storage utilizations or Capacity utilization for DMS Serverless etc.) by all AWS DMS Classic Instances.
2. Centralized Amazon CloudWatch dashboard to review CDC (Change Data Capture) Metrics like Source Latency, Target Latency etc. from all DMS tasks.
3. Setup AWS DMS event notifications (Including change of state like stop, start, fail etc. for all DMS Instances & tasks) for all AWS DMS classic Instances and tasks.
4. Setup hourly Amazon CloudWatch alerts for Errors & Warnings in all AWS DMS migrations including AWS DMS Classic, homogenous migration or DMS Serverless. Users can customize the notification frequency.
5. Setup alerts for AWS DMS Instances to notify when breach thresholds.
6. Setup other alerts for DMS Instances and DMS task. For example, script will alert if there are DMS instances with public access enabled or unused DMS Classic Instances or DMS task with debug logging enabled.

All monitoring placed by the solution are fully customizable where users can choose which monitoring to implement from above list. Also, individual monitoring can be customized to specify filtering for special logs, events.

Prerequisites

1. DMS Instances, tasks are already deployed.
2. Amazon Simple Notification Service (SNS) topic already setup for sending notification to users.
3. Amazon Simple Storage Service (S3) bucket to store deployment scripts.
4. AWS Identity and Access Management (IAM) permissions to create new role & deploy solution. As part of this solution, two new Amazon IAM roles (prefixed with Amazon CloudFormation stack name) will be created:

- %-**LambdaExecutionRole**

Permissions:

- "dms:Describe*",
- "logs:DescribeLogGroups",
- "logs:DescribeLogStreams",
- "logs:FilterLogEvents",
- "cloudwatch:PutMetricAlarm",
- "cloudwatch:PutDashboard",
- "sns:Publish"

- %-**SchedulerExecutionRole**

Permissions:

- "lambda:InvokeFunction"

Implementation

1. Download scripts folder including python zip files and Amazon CloudFormation template Upload downloaded files to your Amazon S3 bucket:

Destination
s3://dms-

Succeeded
✔ 6 files, 25.4 KB (100.00%)

Failed
⚠ 0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (6 Total, 25.4 KB)

< 1 >

Name	Folder	Type	Size	Status	Error
dms_cdc_latency_dashboa...	-	application/zip	3.2 KB	✔ Succeeded	-
dms_error_notifications.py...	-	application/zip	3.2 KB	✔ Succeeded	-
dms_instance_alarms.py.zi...	-	application/zip	3.4 KB	✔ Succeeded	-
dms_instance_dashboard.p...	-	application/zip	3.0 KB	✔ Succeeded	-
dms_monitoring.yaml	-	-	9.1 KB	✔ Succeeded	-
dms_status_checks.py.zip	-	application/zip	3.7 KB	✔ Succeeded	-

2. Create Amazon CloudFormation stack to deploy the monitoring. This template accept parameter input to select which script to deploy. You also require to provide Amazon S3 bucket name where the scripts are uploaded. Also provide Amazon Simple Notification Service (SNS) topic ARN. This ARN is set on all Amazon Lambda functions deployed by this template as 'SNS_TOPIC' environment variable.

- Step 1
Create stack
- Step 2
Specify stack details
- Step 3
Configure stack options
- Step 4
Review and create

Create stack

Prerequisite - Prepare template

You can also create a template by scanning your existing resources in the [IaC generator](#).

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ **Choose an existing template**

Upload or choose an existing template.

☐ **Use a sample template**

Choose from our sample template library.

☐ **Build from Application Composer**

Create a template using a visual builder.

Specify template [Info](#)

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☒ **Amazon S3 URL**

Provide an Amazon S3 URL to your template.

☐ **Upload a template file**

Upload your template directly to the console.

☐ **Sync from Git - new**

Sync a template from your Git repository.

Amazon S3 URL

Amazon S3 template URL

S3 URL: [https://\[REDACTED\].s3.us-east-2.amazonaws.com/dms_monitoring.yaml](https://[REDACTED].s3.us-east-2.amazonaws.com/dms_monitoring.yaml)

[View in Application Composer](#)

[Cancel](#)

[Next](#)

- Step 1
Create stack
- Step 2
Specify stack details
- Step 3
Configure stack options
- Step 4
Review and create

Specify stack details

Provide a stack name

Stack name

dms-monitoring

Stack name must be 1 to 128 characters, start with a letter, and only contain alphanumeric characters. Character count: 14/128.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AlarmOnDMSResources

Setup CloudWatch Alarm to receive alerts on AWS DMS Instance resource utilizations

Yes

AlarmOnErrorWarningsInTask

Setup CloudWatch Alarm to receive alerts on AWS DMS Task Errors & Warnings

Yes

CreateNewIAMRole

Create a new IAM role with full permissions on DMS, CloudWatch, and CloudWatch Logs

Yes

DMSInstanceDashboard

Setup consolidated CloudWatch Dashboard to monitor all AWS DMS Instance Resource consumptions

Yes

DMSNotifications

Setup notification for AWS DMS Instance best practices (Public access/unused instances etc.)

Yes

DeployCDCDashboard

Setup CloudWatch Dashboard to monitor AWS DMS CDC Latency

Yes

S3BucketName

Name of Amazon S3 bucket in the Region you are deploying the monitoring

c[REDACTED]o

SnsTopicArn

The ARN of the SNS topic to which the Lambda functions will publish notifications

arn:aws:sns:us-east-2:[REDACTED]:dms-sns-topic

Cancel

Previous

Next

3. List of AWS resources deployed after successful completion of Amazon CloudFormation template.

Resources (14)

<input type="text"/> Search resources				
Logical ID	Physical ID	Type	Status	
DMSInstanceEventSubscription	dms-monitoring-DMSInstancesSubscription	AWS::DMS::EventSubscription	CREATE_COMPLETE	
DMSTaskEventSubscription	dms-monitoring-DMSTasksSubscription	AWS::DMS::EventSubscription	CREATE_COMPLETE	
LambdaExecutionRole	dms-monitoring-LambdaExecutionRole	AWS::IAM::Role	CREATE_COMPLETE	
SchedulerExecutionRole	dms-monitoring-SchedulerExecutionRole	AWS::IAM::Role	CREATE_COMPLETE	
LambdaFunction1	dms-monitoring-CDC-Dashboard	AWS::Lambda::Function	CREATE_COMPLETE	
LambdaFunction2	dms-monitoring-ErrorNotifications	AWS::Lambda::Function	CREATE_COMPLETE	
LambdaFunction3	dms-monitoring-DMS-Instance-Alarms	AWS::Lambda::Function	CREATE_COMPLETE	
LambdaFunction4	dms-monitoring-DMS-Instances-Dashboard	AWS::Lambda::Function	CREATE_COMPLETE	
LambdaFunction5	dms-monitoring-Misc-Alerts	AWS::Lambda::Function	CREATE_COMPLETE	
LambdaFunction1Scheduler	dms-monitoring-Create-CDC-Dashboard	AWS::Scheduler::Schedule	CREATE_COMPLETE	
LambdaFunction2Scheduler	dms-monitoring-Schedule-ErrorNotifications	AWS::Scheduler::Schedule	CREATE_COMPLETE	
LambdaFunction3Scheduler	dms-monitoring-Schedule-DMS-Instance-Alarms	AWS::Scheduler::Schedule	CREATE_COMPLETE	
LambdaFunction4Scheduler	dms-monitoring-Create-DMS-Instances-Dashboard	AWS::Scheduler::Schedule	CREATE_COMPLETE	
LambdaFunction5Scheduler	dms-monitoring-Schedule-Misc-Alerts	AWS::Scheduler::Schedule	CREATE_COMPLETE	

4. List of Amazon Lambda functions deployed:

<input type="checkbox"/>	Function name	Description	Package type	Runtime
<input type="checkbox"/>	dms-monitoring-Misc-Alerts	-	Zip	Python 3.12
<input type="checkbox"/>	dms-monitoring-DMS-Instance-Alarms	-	Zip	Python 3.12
<input type="checkbox"/>	dms-monitoring-CDC-Dashboard	-	Zip	Python 3.12
<input type="checkbox"/>	dms-monitoring-DMS-Instances-Dashboard	-	Zip	Python 3.12
<input type="checkbox"/>	dms-monitoring-ErrorNotifications	-	Zip	Python 3.12

5. Each Lambda function has various environment variables to configure. Example:

- dms-monitoring-ErrorNotifications

Configurable parameters:

SNS_TOPIC_ARN

Set by default with Amazon SNS ARN specified as Amazon CloudFormation stack parameter.

LOG_GROUP_PATTERN

#Specify pattern for AWS DMS Log pattern('dms-*'). Do NOT change this pattern.

EXCLUDED_LOG_STREAMS

#Specify pattern for AWS DMS task log stream to exclude from the list.

filter_pattern

Filter pattern for errors and warnings in selected LogGroups.
Default set to '?ERROR ?WARNING ?error ?warning' .

- dms-monitoring-misc-alerts

Configurable parameters:

SNS_TOPIC_ARN

Set by default with Amazon SNS ARN specified as Amazon CloudFormation stack parameter.

EXCLUDE_INSTANCES

Specify comma separated list of DMS Instances you want to exclude from the monitoring alert.

EXCLUDE_TASKS

Specify comma separated list of DMS tasks you want to exclude from the monitoring alert.

- dms-monitoring-dms-instance-alarms

Configurable parameters:

CPU_THRESHOLD

Set by default to alert when CPU Utilization cross 80% threshold.

MEMORY_THRESHOLD

Set by default to alert when Freeable Memory cross below 25% threshold.

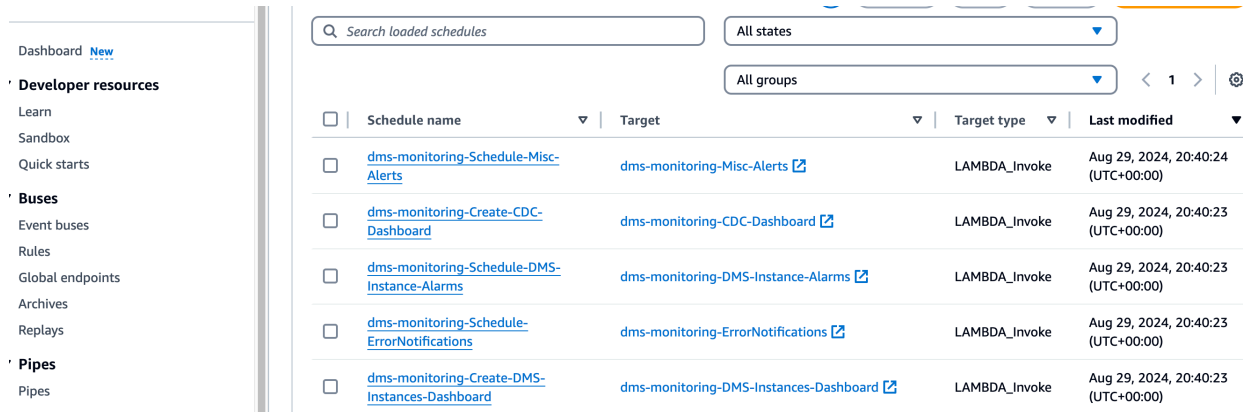
SWAP_THRESHOLD

Set by default to alert when SWAP Utilization cross 25% threshold.

CAPACITY_THRESHOLD

Set by default to alert when Capacity Utilization for AWS DMS Serverless replication cross 75% threshold.

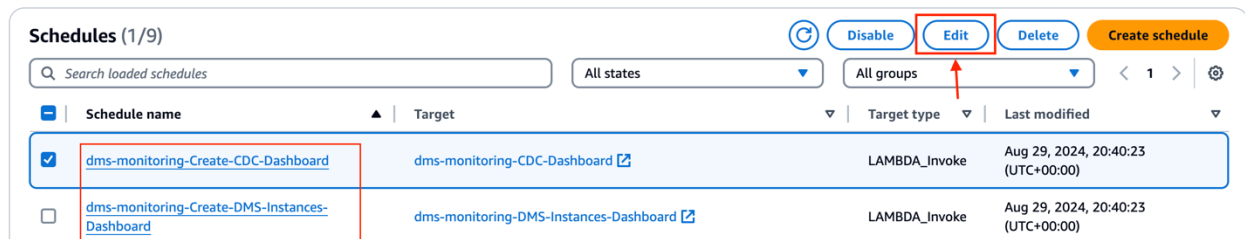
6. List of AWS Event Bridge Schedules deployed.



The screenshot shows the AWS Event Bridge console interface. On the left is a sidebar with navigation links: Dashboard, Developer resources, Buses, and Pipes. The main area displays a table of loaded schedules. At the top, there is a search bar and two dropdown menus for 'All states' and 'All groups'. The table has columns for 'Schedule name', 'Target', 'Target type', and 'Last modified'. Five schedules are listed, each with a checkbox in the first column.

<input type="checkbox"/>	Schedule name	Target	Target type	Last modified
<input type="checkbox"/>	dms-monitoring-Schedule-Misc-Alerts	dms-monitoring-Misc-Alerts	LAMBDA_Invoke	Aug 29, 2024, 20:40:24 (UTC+00:00)
<input type="checkbox"/>	dms-monitoring-Schedule-CDC-Dashboard	dms-monitoring-CDC-Dashboard	LAMBDA_Invoke	Aug 29, 2024, 20:40:23 (UTC+00:00)
<input type="checkbox"/>	dms-monitoring-Schedule-DMS-Instance-Alarms	dms-monitoring-DMS-Instance-Alarms	LAMBDA_Invoke	Aug 29, 2024, 20:40:23 (UTC+00:00)
<input type="checkbox"/>	dms-monitoring-Schedule-ErrorNotifications	dms-monitoring-ErrorNotifications	LAMBDA_Invoke	Aug 29, 2024, 20:40:23 (UTC+00:00)
<input type="checkbox"/>	dms-monitoring-Schedule-DMS-Instances-Dashboard	dms-monitoring-DMS-Instances-Dashboard	LAMBDA_Invoke	Aug 29, 2024, 20:40:23 (UTC+00:00)

7. Scheduler also have two schedules for creating Amazon CloudWatch dashboard. These schedules has Amazon Lambda functions as target and one time execution date is set to 2050/01/01.



The screenshot shows a closer view of the AWS Event Bridge console. At the top, there are buttons for 'Disable', 'Edit' (highlighted with a red box and an arrow), and 'Delete'. Below these is a table with two schedules. The first schedule, 'dms-monitoring-Schedule-CDC-Dashboard', is selected with a checkbox. The second schedule, 'dms-monitoring-Schedule-DMS-Instances-Dashboard', is not selected. Both schedules have 'LAMBDA_Invoke' as the target type and a last modified date of 'Aug 29, 2024, 20:40:23 (UTC+00:00)'.

<input checked="" type="checkbox"/>	Schedule name	Target	Target type	Last modified
<input checked="" type="checkbox"/>	dms-monitoring-Schedule-CDC-Dashboard	dms-monitoring-CDC-Dashboard	LAMBDA_Invoke	Aug 29, 2024, 20:40:23 (UTC+00:00)
<input type="checkbox"/>	dms-monitoring-Schedule-DMS-Instances-Dashboard	dms-monitoring-DMS-Instances-Dashboard	LAMBDA_Invoke	Aug 29, 2024, 20:40:23 (UTC+00:00)

8. Edit schedules for creating these dashboards and specify date and time as per your schedule to execute the target Lambda functions and deploy respective Amazon CloudWatch Dashboards.

Schedule pattern

Occurrence | [Info](#)
You can define an one-time or recurrent schedule.

☒ One-time schedule☐ Recurring schedule

Date and time
The date and time to invoke the target.

2050/01/01

00:00

UTC

YYYY/MM/DD

Use 24-hour format timestamp (hh:mm)

Time zone

Flexible time window
If you choose a flexible time window, Scheduler invokes your schedule within the time window you specify. For example, if you choose 15 minutes, your schedule runs within 15 minutes after the schedule start time.

Off

[Cancel](#)[Skip to Review and save schedule](#)[Next](#)

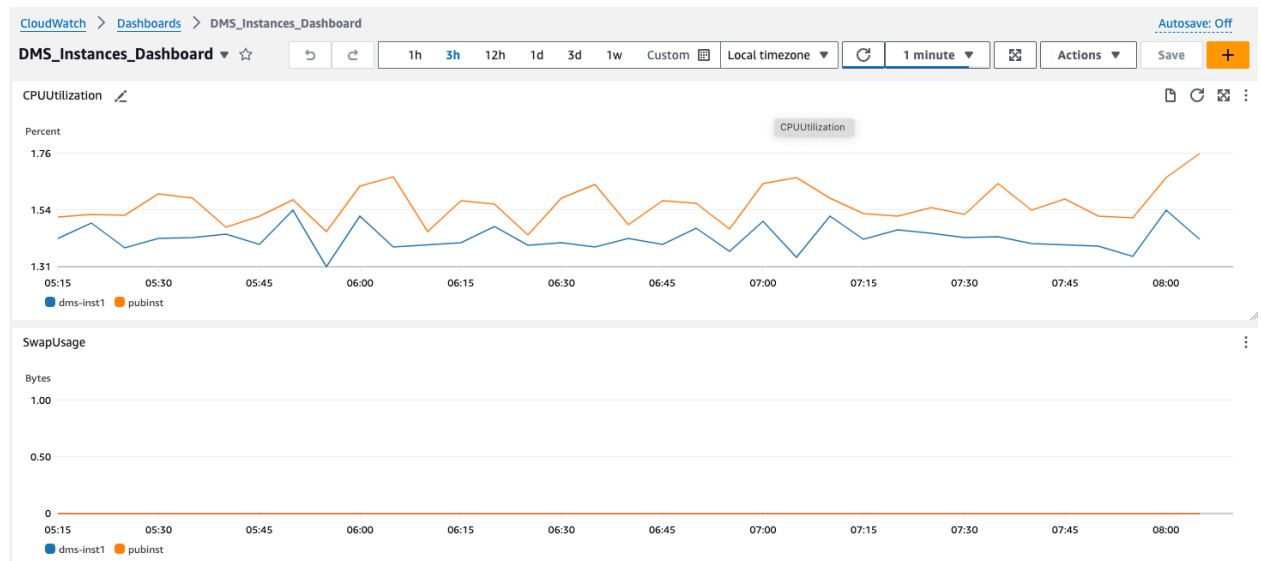
- Optionally, you can also verify and update schedule for remaining monitoring alerts set on Amazon Event Bridge scheduler.

Clean up

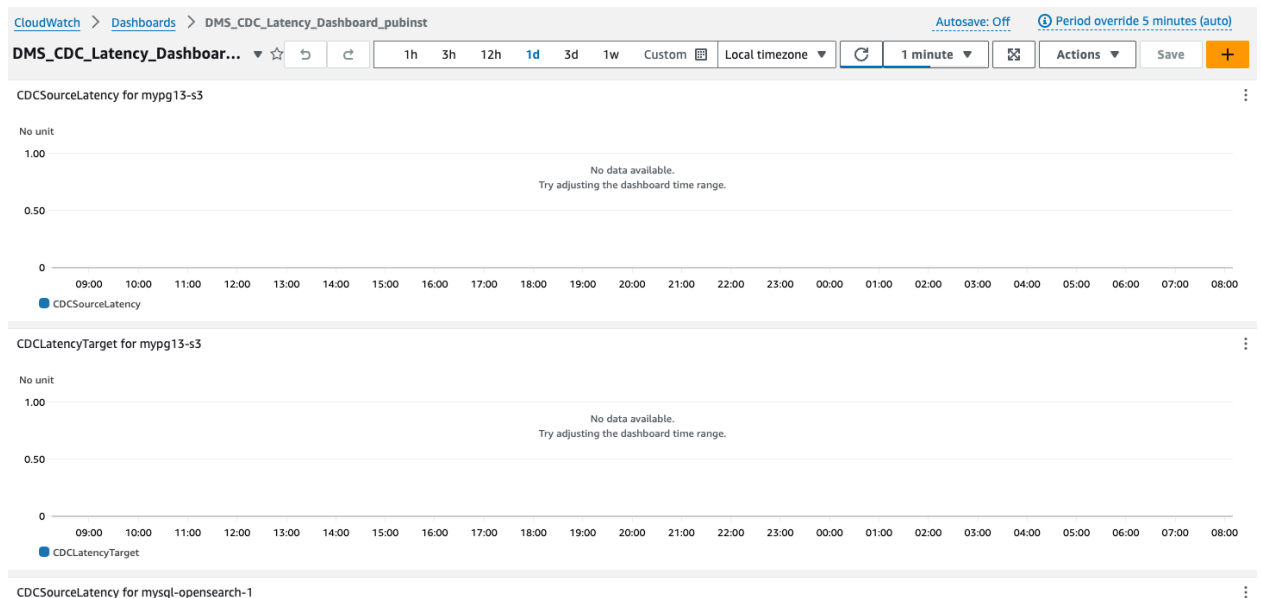
Delete Amazon CloudFormation stack to remove all resources deployed as part of this solution.

Appendix

1. Centralized Amazon CloudWatch dashboard for AWS DMS Replication Instances



2. Centralized Amazon CloudWatch dashboard for DMS task CDC Metrics




3. CloudFormation Template for DMS Instance and Task Event Subscription

Event subscriptions (2)				
<div>Find event subscription</div>				
<div>< 1 ></div>				
<input type="checkbox"/>	Name	Status	Source type	Enabled
<input type="checkbox"/>	DMSInstanceEventSubscription-6W3pJzZ5z44TXoCR	Active	Replication instance	Yes
<input type="checkbox"/>	DMSTaskEventSubscription-blkOYoPfieSnWvaX	Active	Replication task	Yes

4. CloudWatch Alerts for Errors and Warnings in DMS Tasks

Alert:dms-task-CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO

 AWS Notifications <no-reply@sns.amazonaws.com>
To: Mathur, Ravi


Matched events in log group: dms-tasks-pubinst, log stream: dms-task-CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO

Events:

2024-08-29T20:02:12 [TASK_MANAGER] JI: Task 'CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO' starting full load only in resume mode after recoverable error, retry #7 (replicationtask.c:1592)
2024-08-29T20:02:13 [METADATA_MANAGE] JE: RetCode: SQL_ERROR SqlState: 08001 NativeError: 101 Message: could not translate host name "mypg11-r1. [REDACTED]us-east-1.rds.amazonaws.com" to address not known [1022502] (ar_odbc_conn.c:599)
2024-08-29T20:02:43 [TASK_MANAGER] JI: Task - CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO is in ERROR state, updating starting status to AR_NOT_APPLICABLE (repository.c:5483)
2024-08-29T20:02:43 [TASK_MANAGER] JE: Task 'CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO' encountered a recoverable error, retry attempt # 7 (repository.c:5565)
2024-08-29T20:13:39 [TASK_MANAGER] JI: Task 'CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO' starting full load only in resume mode after recoverable error, retry #8 (replicationtask.c:1592)
2024-08-29T20:13:39 [METADATA_MANAGE] JE: RetCode: SQL_ERROR SqlState: 08001 NativeError: 101 Message: could not translate host name "mypg11-r1. [REDACTED]us-east-1.rds.amazonaws.com" to address not known [1022502] (ar_odbc_conn.c:599)
2024-08-29T20:14:09 [TASK_MANAGER] JI: Task - CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO is in ERROR state, updating starting status to AR_NOT_APPLICABLE (repository.c:5483)
2024-08-29T20:14:09 [TASK_MANAGER] JE: Task 'CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO' encountered a recoverable error, retry attempt # 8 (repository.c:5565)
2024-08-29T20:35:42 [TASK_MANAGER] JI: Task 'CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO' starting full load only in resume mode after recoverable error, retry #9 (replicationtask.c:1592)
2024-08-29T20:35:42 [METADATA_MANAGE] JE: RetCode: SQL_ERROR SqlState: 08001 NativeError: 101 Message: could not translate host name "mypg11-r1. [REDACTED]us-east-1.rds.amazonaws.com" to address not known [1022502] (ar_odbc_conn.c:599)
2024-08-29T20:36:12 [TASK_MANAGER] JI: Task - CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO is in ERROR state, updating starting status to AR_NOT_APPLICABLE (repository.c:5483)
2024-08-29T20:36:12 [TASK_MANAGER] JE: Task 'CXTAD4BCDDCAIK6HXASWG35QKNJPDFOBIA6WFO' encountered a recoverable error, retry attempt # 9 (repository.c:5565)

5. DMS Instances and DMS tasks best practice alerts

AWS DMS Alerts!

 AWS Notifications <no-reply@sns.amazonaws.com>
To: Mathur, Ravi

Yesterday at 3:31 PM

AWS DMS instance is running idle and NO active DMS task found!
-[dms-inst1] [arn:aws:dms:us-east-1:[REDACTED]:rep:L7OQQIL3LZBJTH5XK6W4DDIGNI]
Recommended Action: You may save cost by removing DMS Instances that are no longer required.

AWS DMS instance is running idle and NO active DMS task found!
-[pubinst] [arn:aws:dms:us-east-1:[REDACTED]:rep:LSK2HBCEAFWWH5YMPA3SLRGU3YD5HPV4HUX6A]
Recommended Action: You may save cost by removing DMS Instances that are no longer required.

AWS DMS instance has public access enabled!
-[pubinst] [arn:aws:dms:us-east-1:[REDACTED]:rep:LSK2HBCEAFWWH5YMPA3SLRGU3YD5HPV4HUX6A]
Recommended Action: It is security best practice to disable public access from all AWS DMS instances.

DMS tasks with debug logging enabled:
-sless-s3: arn:aws:dms:us-east-1:[REDACTED]:task:HAAW2DCTWRARTPVOVMAM4X2G3Q
Recommended Action: It is recommended to enable DMS task debug logging only for the short duration for troubleshooting. Keeping debug logging enabled for longer period of time may impact DMS replication instance performance. It may also fill up the storage quickly and impact all of the task running on the same DMS instance.