
AWS Database Migration Services NoSQL Lab Runbook

MongoDB to DynamoDB Migration

Revised: 2017.10.26

© 2017 Amazon Web Services



This document is © 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Contents

1 Introduction	6
1.1 Objective	6
1.1.1 Lab Setup	6
1.1.2 Lab Steps	6
1.1.3 Lab Teardown	6
1.2 About AWS Database Migration Service (AWS DMS)	7
2 Lab Setup	8
2.1 Setup EC2 Key Pair	8
2.2 Setup AWS CloudFormation Stack	9
2.2.1 AWS CloudFormation Stack Selection	10
2.2.2 AWS CloudFormation Stack Settings	11
2.2.3 AWS CloudFormation Stack Options	12
2.2.4 AWS CloudFormation Stack Review	13
2.2.5 AWS CloudFormation Stack: List Stacks	13
2.2.6 AWS CloudFormation Stack: Outputs	14
3 Lab Steps	15
3.1 Create AWS DMS Role	15
3.2 AWS Database Migration Services	21
3.2.1 AWS DMS Replication Instances	22
3.2.2 AWS DMS Endpoints	24
3.2.3 AWS DMS Tasks	31
4 Lab Teardown	36
4.1 Teardown AWS DMS Resources	36
4.1.1 Teardown AWS DMS Resources: Tasks	36
4.1.2 Teardown AWS DMS Resources: Endpoints	36
4.1.3 Teardown AWS DMS Resources: Replication Instances	38
4.2 Teardown AWS CloudFormation Stack	38
4.2.1 Teardown CloudFormation Stack: Identify Workshop Stack	38
4.2.2 Teardown CloudFormation Stack: Delete Workshop Stack	39
4.2.3 Teardown CloudFormation Stack: Confirm Deletion	39
4.2.4 Teardown CloudFormation Stack Status: Delete Failed	39
4.3 Teardown EC2 Key Pair	41
4.4 Teardown DynamoDB	42
5 Conclusion	45

List of Tables

List of Figures

1	Key Pairs: List All Keys	8
2	Key Pair: Create Dialog	9
3	Key Pair: Created	9
4	AWS CloudFormation Stack Selection: Template	10
5	AWS CloudFormation Stack Selection: Confirmation	11
6	AWS CloudFormation: Settings Overview	11
7	AWS CloudFormation Stack Parameters: Confirmation	11
8	AWS CloudFormation Stack Options: All	12
9	AWS CloudFormation Stack Options: Confirmation	12
10	AWS CloudFormation Stack: Review	13
11	AWS CloudFormation Stack Review: Creation	13
12	AWS CloudFormation Stack: List Stacks	14
13	AWS CloudFormation Stack: Outputs	14
14	AWS DMS Role - IAM Console	15
15	AWS DMS Role - Create IAM Role	16
16	AWS DMS Role - Select 'AWS Service'	17
17	AWS DMS Role - Select 'DMS'	17
18	AWS DMS Role - Attach Permissions	18
19	AWS DMS Role - Filter for DynamoDB Permissions	19
20	AWS DMS Role - Add Role Name	20
21	AWS DMS Role - Filter Roles to find new role	21
22	AWS DMS Role - View Details, Note the ARN for later	21
23	AWS DMS Replication Instances - List All	22
24	AWS DMS Replication Instances - Create New	23
25	AWS DMS Replication Instances - List New Creating	24
26	AWS DMS Replication Instances - List New Available	24
27	AWS DMS Endpoints - List All	25
28	AWS DMS Endpoints - Create Endpoint	26
29	AWS DMS Endpoints - Create Source Endpoint (Details)	27
30	AWS DMS Endpoints - Create Source Endpoint (Test)	28
31	AWS DMS Endpoints - Create Source Endpoint (Test Success)	29
32	AWS DMS Endpoints - List All (Updated)	29
33	AWS DMS Endpoints - Create Target Endpoint (Details)	30
34	AWS DMS Endpoints - Create Target Endpoint (Test)	30
35	AWS DMS Endpoints - Create Target Endpoint (Test Success)	31

36	AWS DMS Endpoints - List Endpoints (Source and Target)	31
37	AWS DMS Tasks - List Tasks	32
38	AWS DMS Tasks - Create Task (Detail)	32
39	AWS DMS Tasks - Create Task (Continued)	33
40	AWS DMS Tasks - Add Selection Criteria	34
41	AWS DMS Tasks - Create Task (Final)	35
42	AWS DMS Tasks - List Tasks (Updated)	35
43	Teardown AWS DMS - List All Endpoints	37
44	Teardown AWS DMS - Endpoints Deleted	37
45	Teardown CloudFormation Stack: Delete Success	39
46	Teardown CloudFormation Stack: Delete Failed	39
47	Teardown CloudFormation Stack: Confirm Delete Resources	40
48	Teardown EC2 Key Pair - List Keys	41
49	Teardown EC2 Key Pair - Delete Key	42
50	Teardown DynamoDB: DynamoDB Console	43
51	Teardown DynamoDB: DynamoDB Console	43
52	Teardown DynamoDB: DynamoDB Console	44
53	Teardown DynamoDB: DynamoDB Console	44
54	Teardown DynamoDB: DynamoDB Console	45

1 Introduction

1.1 Objective

In this lab, you will be performing a database migration from a MongoDB source to a Amazon DynamoDB target using the AWS Databases Migration Service (AWS DMS).

1.1.1 Lab Setup

- Create EC2 Key Pair
- Launch AWS CloudFormation Stack

1.1.2 Lab Steps

- Create AWS Database Migration Service Resources
- Create Source Endpoint in AWS DMS
- Create Target Endpoint in AWS DMS
- Create a Migration Task in AWS DMS
- Start the migration

1.1.3 Lab Teardown

- Delete AWS CloudFormation Stack
- Delete AWS Database Migration Service Resources
- Delete EC2 Key Pair

1.2 About AWS Database Migration Service (AWS DMS)

The AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases.

The service supports homogenous migrations such as Oracle to Oracle, as well as heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL. It also allows you to stream data to Amazon Redshift, Amazon DynamoDB, and Amazon S3 from any of the supported sources including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, SAP ASE, SQL Server and MongoDB, enabling consolidation and easy analysis of data in the petabyte-scale data warehouse. AWS Database Migration Service can also be used for continuous data replication with high-availability.

To find out more about AWS DMS, see the product page at <https://aws.amazon.com/dms/>

2 Lab Setup

2.1 Setup EC2 Key Pair

In this step, you will generate an EC2 key pair for use in the Database Migration Workshop labs.

Make sure you have selected the **Asia Pacific (Tokyo)** region by visiting the following link:

<http://amzn.to/aws-tokyo-keypairs> (=> <https://ap-northeast-1.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-1#KeyPairs:sort=keyName>)

If no EC2 key pairs have been created in this region yet, you will see the following screen:

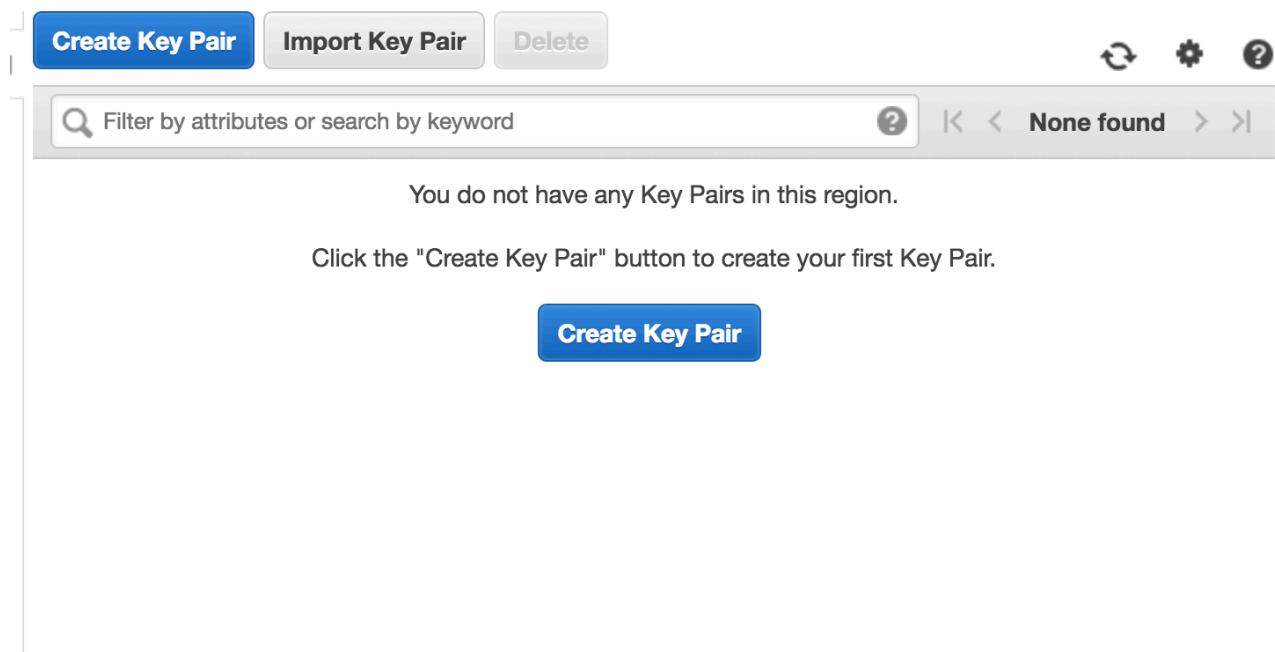


Figure 1: Key Pairs: List All Keys

For this workshop, we recommend the creation of a new EC2 key pair to be used associated with workshop resources. We will later delete this key pair along with the workshop resources.

You should click the **Create Key Pair** button and enter **workshop** as the name for the new key pair as shown below, clicking the **Create** button to complete the creation of the new key pair.

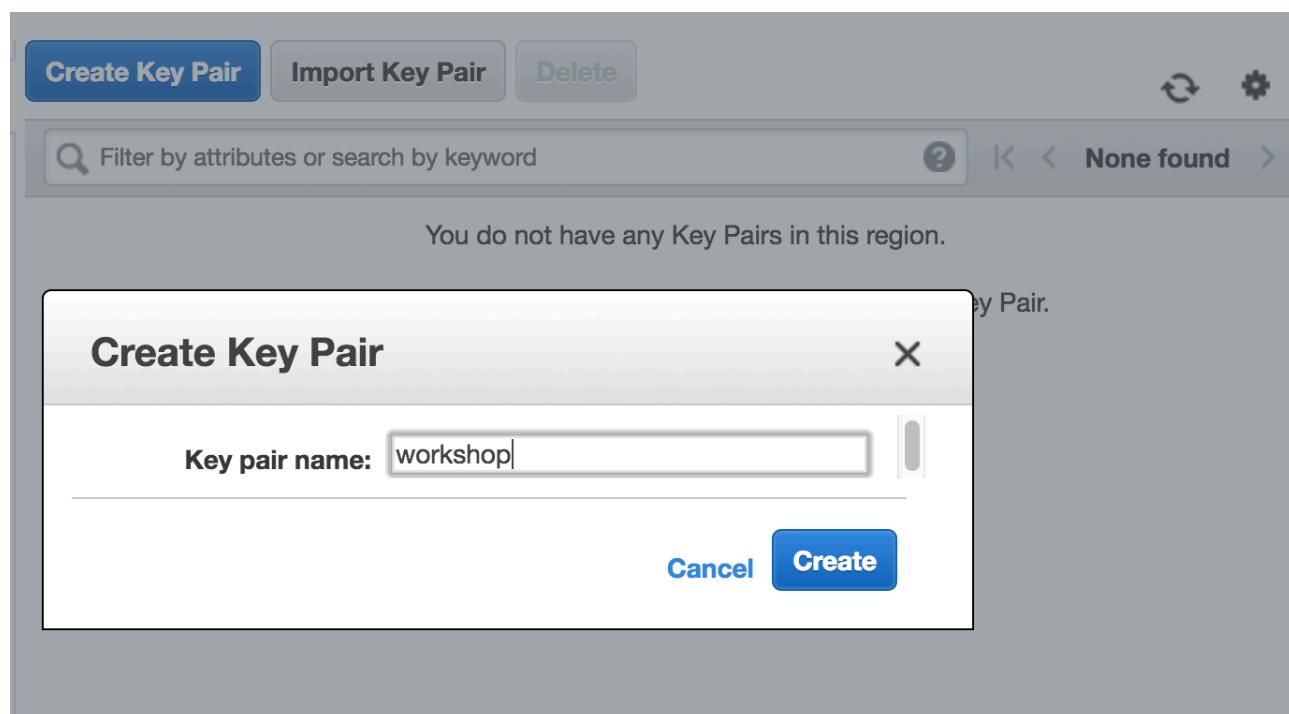


Figure 2: Key Pair: Create Dialog

You should now see a new EC2 key pair labeled **workshop**:

		Create Key Pair	Import Key Pair	Delete	Refresh	Settings	Help
		Filter by attributes or search by keyword		? < < 1 to 1 of 1 > >			
		Key pair name	Fingerprint				
<input checked="" type="checkbox"/>	workshop		cc:e3:5d:70:b2:47:56:ae:17:a0:44:5d:c6:30:d3:ce:cb:02:6a:a3				

Figure 3: Key Pair: Created

The private key for this EC2 key pair will have automatically downloaded to your browser's default directory, and this file will be called **workshop.pem**.

2.2 Setup AWS CloudFormation Stack

In this step, you will launch a AWS CloudFormation template that will setup the following resources needed for this lab:

- Source Database: Amazon RDS Oracle (this database will be pre-populated with sample database installed from <https://github.com/awslabs/aws-database-migration-samples>)
- Target Database: Amazon RDS PostgreSQL

CAUTION: The resources created automatically with this CloudFormation template will continue to run until the CloudFormation stack is deleted or the individual resources are shutdown -- the steps for teardown are located in this document at [Teardown: AWS Cloudformation Stack](#)

To launch this template, use the following link:

<http://amzn.to/aws-dms-workshop-nosql> (=> <https://ap-northeast-1.console.aws.amazon.com/cloudformation/home?region=ap-northeast-1#/stacks/new?stackName=workshop-cfn-nosql&templateURL=https://s3-ap-northeast-1.amazonaws.com/aws-dms-workshop/workshop-lab-nosql-mongodb-dynamodb.yaml>)

2.2.1 AWS CloudFormation Stack Selection

You should now see the following:

Create stack

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The left sidebar has tabs for 'Select Template', 'Specify Details', 'Options', and 'Review'. The main area is titled 'Select Template' with a sub-instruction: 'Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.' Below this are two sections: 'Design a template' (with a 'Design template' button) and 'Choose a template' (with three options: 'Select a sample template' (radio button), 'Upload a template to Amazon S3' (button 'Choose File' with 'No file chosen'), and 'Specify an Amazon S3 template URL' (radio button selected, input field containing 'https://s3-ap-northeast-1.amazonaws.com/aws-dms-works' with a 'View/Edit template in Designer' link). At the bottom right are 'Cancel' and 'Next' buttons.

Figure 4: AWS CloudFormation Stack Selection: Template

The default settings are recommended, and no changes are required to this page. Click the Next button on the bottom of the page to continue.

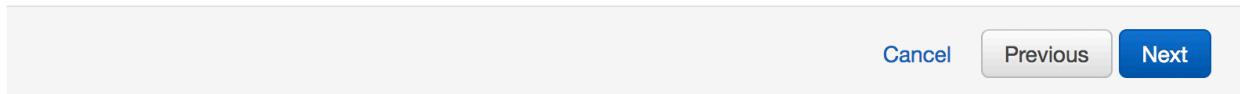


Figure 5: AWS CloudFormation Stack Selection: Confirmation

2.2.2 AWS CloudFormation Stack Settings

This page displays the settings and parameters for the CloudFormation stack.

- For **Stack name**, you can use the default of **workshop-cfn-nosql**
- For **KeyName**, set to the name of the Key Pair you created earlier (default is **workshop**)
- For **KeyPair**, you will need to confirm your Key Pair created earlier (in our example, it is labeled **workshop**)

The screenshot shows the "Create stack" wizard in the AWS CloudFormation console. The current step is "Specify Details".

- Stack name:** workshop-cfn-nosql
- Parameters:** Key Pair
- Key Pair:** KeyName: workshop
- Source MongoDB Database Configuration:** MongoDBInstanceType: t2.medium

At the bottom right of the main form area, there is a confirmation bar with three buttons: "Cancel" (blue), "Previous" (gray), and "Next" (blue).

Figure 6: AWS CloudFormation: Settings Overview

Once you have confirmed the settings, click the **Next** button on the bottom of the page to continue.

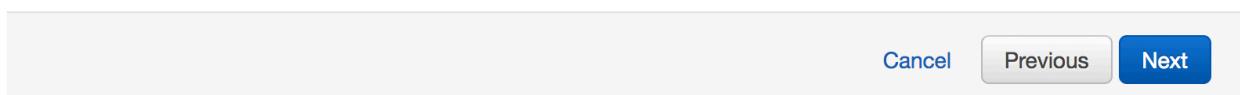


Figure 7: AWS CloudFormation Stack Parameters: Confirmation

2.2.3 AWS CloudFormation Stack Options

You will now see the Options for this CloudFormation template. **The default settings are recommended, and no changes are required to this page.**

Create stack

Select Template

Specify Details

Options

Review

Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. [Learn more](#).

	Key (127 characters maximum)	Value (255 characters maximum)	
1	<input type="text"/>	<input type="text"/>	+

Permissions

You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. [Learn more](#).

IAM Role [Choose a role \(optional\)](#)

Enter role arn

► Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#).

[Cancel](#) [Previous](#) **Next**

Figure 8: AWS CloudFormation Stack Options: All

Click the **Next** button on the bottom of the page to continue.

[Cancel](#) [Previous](#) **Next**

Figure 9: AWS CloudFormation Stack Options: Confirmation

2.2.4 AWS CloudFormation Stack Review

Create stack

Review

Template

Template URL: <https://s3-ap-northeast-1.amazonaws.com/aws-dms-workshop/workshop-lab-nosql-mongodb-dynamodb.yaml>

Description: This CloudFormation template is used during the AWS DMS Workshop and provisions the following - a new VPC (TODO) - a new EC2 instance running MongoDB (source database) - creates an Amazon DynamoDB table (target database) CAUTION -- You will be billed for the AWS resources used if you create a stack from this template, and are responsible for any/all charges made while running these services in your AWS account

Estimate cost: Cost

Details

Stack name: workshop-cfn-nosql

Key Pair

KeyName: workshop

Source MongoDB Database Configuration

MongoDBInstanceType: t2.medium

Options

Tags

No tags provided

Advanced

Notification

Termination Protection	Disable
Timeout	none
Rollback on failure	Yes

Cancel Previous Create

Figure 10: AWS CloudFormation Stack: Review

Click the **Create** button on the bottom of the page to continue.

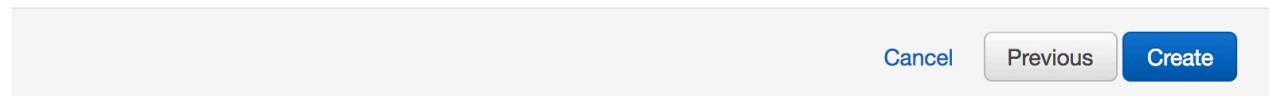


Figure 11: AWS CloudFormation Stack Review: Creation

AWS resources associated with this workshop lab will now be automatically created immediately, and billing will proceed until this CloudFormation stack is deleted or those resources are individually stopped/terminated -- the steps for teardown are located in this document at [Teardown: AWS Cloudformation Stack](#)

2.2.5 AWS CloudFormation Stack: List Stacks

You should now see a list of any existing CloudFormation templates for this region in your account, and the new CloudFormation stack called **workshop** should now appear in your console.

The screenshot shows the AWS CloudFormation 'List Stacks' interface. At the top, there are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter bar is set to 'Active' and 'By Stack Name'. The main table has columns for Stack Name, Created Time, Status, and Description. One row is visible for the stack 'workshop', which was created on 2017-10-22 at 12:12:32 UTC+0800 and is in the 'CREATE_COMPLETE' status. The description notes it's a CloudFormation template.

	Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/>	workshop	2017-10-22 12:12:32 UTC+0800	CREATE_COMPLETE	This CloudFormation te...

Figure 12: AWS CloudFormation Stack: List Stacks

If the stack does not immediately appear, you may need to click the **Refresh** button above the list panel.

2.2.6 AWS CloudFormation Stack: Outputs

When the stack and its underlying resource creations are completed, the status will be displayed as **CREATE_COMPLETE**. Once this has occurred, we can gather Outputs from the CloudFormation stack.

The screenshot shows the 'Outputs' tab for the stack 'workshop-cfn-nosql'. At the top, there's a message about StackSets. The main table lists four outputs: 'MongoDBServerIP' with value '10.0.0.20', 'Regionname' with value 'ap-northeast-1', 'VPC' with value 'vpc-55263831', and 'StackName' with value 'workshop-cfn-nosql'. The table has columns for Key, Value, Description, and Export Name.

Key	Value	Description	Export Name
MongoDBServerIP	10.0.0.20	MongoDB Server private IPv4 address for DMS	
Regionname	ap-northeast-1		
VPC	vpc-55263831	Newly created VPC for this workshop	
StackName	workshop-cfn-nosql		

Figure 13: AWS CloudFormation Stack: Outputs

3 Lab Steps

3.1 Create AWS DMS Role

To access Amazon DynamoDB from AWS DMS, we need to create an IAM service role.

Search IAM

Dashboard

- Groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Credential report

Encryption keys

Welcome to Identity and Access Management

IAM users sign-in link:
<https://shirkeys.signin.aws.amazon.com/console>

Customize | Copy Link

IAM Resources

Users: 7 Roles: 76
Groups: 4 Identity Providers: 0
Customer Managed Policies: 15

Security Status

3 out of 5 complete.

⚠	Activate MFA on your root account	▼
✓	Create individual IAM users	▼
✓	Use groups to assign permissions	▼
⚠	Apply an IAM password policy	▼
✓	Rotate your access keys	▼

Figure 14: AWS DMS Role - IAM Console

The screenshot shows the AWS IAM Roles page. On the left, a sidebar menu includes options like Dashboard, Groups, Users, Roles (which is selected and highlighted in orange), Policies, Identity providers, Account settings, Credential report, and Encryption keys. A search bar labeled "Search IAM" is at the top. The main content area has a title "Roles" and a sub-section "What are IAM roles?". It explains that IAM roles are a secure way to grant permissions to entities and lists several examples. Below this is a section titled "Additional resources:" with links to IAM Roles FAQ, Documentation, a tutorial, and common scenarios. At the bottom, there are two buttons: "Create role" (in blue) and "Delete role". Below these buttons is a search bar with a magnifying glass icon and the placeholder "Search". A table follows, with columns "Role name" and "Description". A single row is shown, with the "Role name" column containing "Admin" and the "Description" column being empty. There is also a small checkbox next to the "Admin" entry.

Figure 15: AWS DMS Role - Create IAM Role

Create role



Select type of trusted entity



Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

API Gateway	Data Pipeline	IoT	Service Catalog
Auto Scaling	Directory Service	Lambda	Storage Gateway
Batch	DynamoDB	Lex	
CloudFormation	EC2	Machine Learning	
CloudHSM	EC2 Container Service	OpsWorks	
CloudWatch Events	EMR	RDS	
CodeBuild	Elastic Beanstalk	Redshift	
CodeDeploy	Elastic Transcoder	SMS	
Config	Glue	SNS	
DMS	Greengrass	SWF	

* Required

[Cancel](#)

[Next: Permissions](#)

Figure 16: AWS DMS Role - Select 'AWS Service'

CodeBuild	Elastic Beanstalk	Redshift
CodeDeploy	Elastic Transcoder	SMS
Config	Glue	SNS
DMS	Greengrass	SWF

Select your use case

DMS

Allows Database Migration Service to call AWS services on your behalf.

* Required

[Cancel](#)

[Next: Permissions](#)

Figure 17: AWS DMS Role - Select 'DMS'

Create role



Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#)[Refresh](#)

		Policy name	Attachments	Description
<input type="checkbox"/>	AdministratorAccess	5	Provides full access to AWS services and resources.	
<input type="checkbox"/>	allow-all-ssm	1	Permit access by EC2 instances to register with Directory ...	
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	0	Provides full access to create/edit/delete APIs in Amazon ...	
<input type="checkbox"/>	AmazonAPIGatewayDynamicDNS	0	Allows API Gateway to push logs to user's account.	
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	0	Provides full access to invoke APIs in Amazon API Gateway.	
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	0	Allows API Gateway to push logs to user's account.	
<input type="checkbox"/>	AmazonAppStreamFullAccess	1	Provides full access to Amazon AppStream via the AWS ...	
<input type="checkbox"/>	AmazonAppStreamReadOnlyAccess	0	Provides read only access to Amazon AppStream via the ...	
<input type="checkbox"/>	AmazonAppStreamServiceAccess	1	Default policy for Amazon AppStream service role.	
<input type="checkbox"/>	AmazonAthenaFullAccess	0	Provide full access to Amazon Athena and scoped access...	
<input type="checkbox"/>	AmazonCloudDirectoryFullAccess	0	Provides full access to Amazon Cloud Directory Service.	
<input type="checkbox"/>	AmazonCloudDirectoryReadOnlyAccess	0	Provides read only access to Amazon Cloud Directory Ser...	
<input type="checkbox"/>	AmazonCognitoDeveloperAuthenticatedIdentit...	0	Provides access to Amazon Cognito APIs to support deve...	
<input type="checkbox"/>	AmazonCognitoPowerUser	0	Provides administrative access to existing Amazon Cognit...	
<input type="checkbox"/>	AmazonCognitoReadOnly	0	Provides read only access to Amazon Cognito resources.	

*** Required**[Cancel](#)[Previous](#)[Next: Review](#)

Figure 18: AWS DMS Role - Attach Permissions

Create role



Attach permissions policies

Choose one or more policies to attach to your new role.

[Create policy](#)

[Refresh](#)

Filter: Policy type ▾		Showing 7 results
	Policy name ▾	Attachments ▾
<input checked="" type="checkbox"/>	▶ AmazonDynamoDBFullAccess	2 Provides full access to Amazon DynamoDB via the AWS Ma...
<input type="checkbox"/>	▶ AmazonDynamoDBFullAccesswithDataPipeline	0 Provides full access to Amazon DynamoDB including Export...
<input type="checkbox"/>	▶ AmazonDynamoDBReadOnlyAccess	0 Provides read only access to Amazon DynamoDB via the A...
<input type="checkbox"/>	▶ AWSApplicationAutoscalingDynamoDBTableP...	0 Policy granting permissions to Application Auto Scaling to a...
<input type="checkbox"/>	▶ AWSLambdaDynamoDBExecutionRole	0 Provides list and read access to DynamoDB streams and wri...
<input type="checkbox"/>	▶ AWSLambdaDynamoDBExecutionRole-f4d5cf...	0
<input type="checkbox"/>	▶ AWSLambdaInvocation-DynamoDB	0 Provides read access to DynamoDB Streams.

* Required

[Cancel](#)

[Previous](#)

[Next: Review](#)

Figure 19: AWS DMS Role - Filter for DynamoDB Permissions

Create role



Review

Provide the required information below and review this role before you create it.

Role name* Maximum 64 characters. Use alphanumeric and '+,-,@-' characters.

Role description Maximum 1000 characters. Use alphanumeric and '+,-,@-' characters.

Trusted entities AWS service: dms.amazonaws.com

Policies  [AmazonDynamoDBFullAccess](#) 

* Required Cancel Previous Create role

Figure 20: AWS DMS Role - Add Role Name

Specify the role name **dms-workshop-dynamodb-role** and click **Create role**

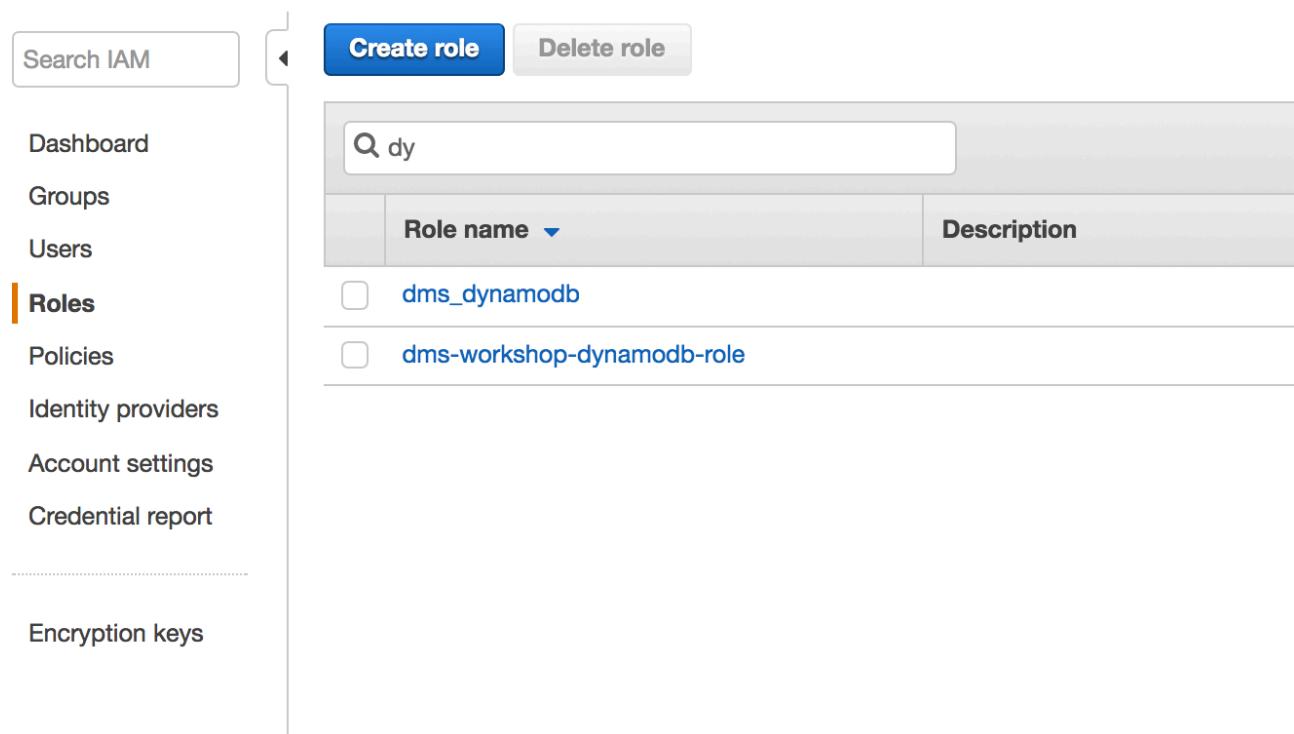


Figure 21: AWS DMS Role - Filter Roles to find new role

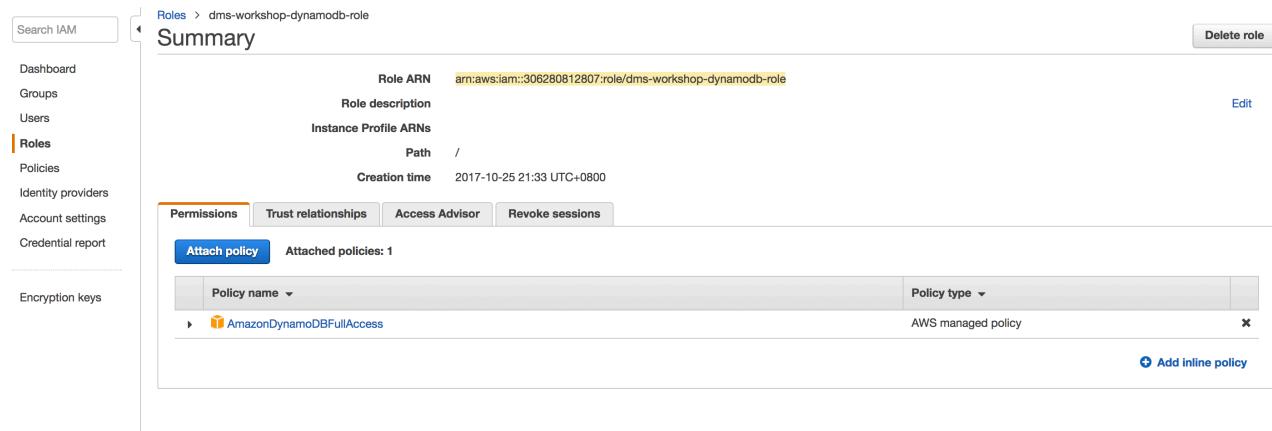


Figure 22: AWS DMS Role - View Details, Note the ARN for later

3.2 AWS Database Migration Services

The AWS Database Migration Service (AWS DMS) consist of three main components:

- **Replication Instances** for performing the movement of data
- **Endpoints** for specifying the source and target databases
- **Tasks** for controlling the execution of migrations on the replication instances

We will provide instructions on how to set up each of these three components.

3.2.1 AWS DMS Replication Instances

AWS DMS Replication Instances - List All To create our new AWS DMS replication instances, we will first view the console for replication instances by visiting the following link:

<http://amzn.to/aws-tokyo-dms-instances> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#replication-instances:>)

You should now see the following screen:

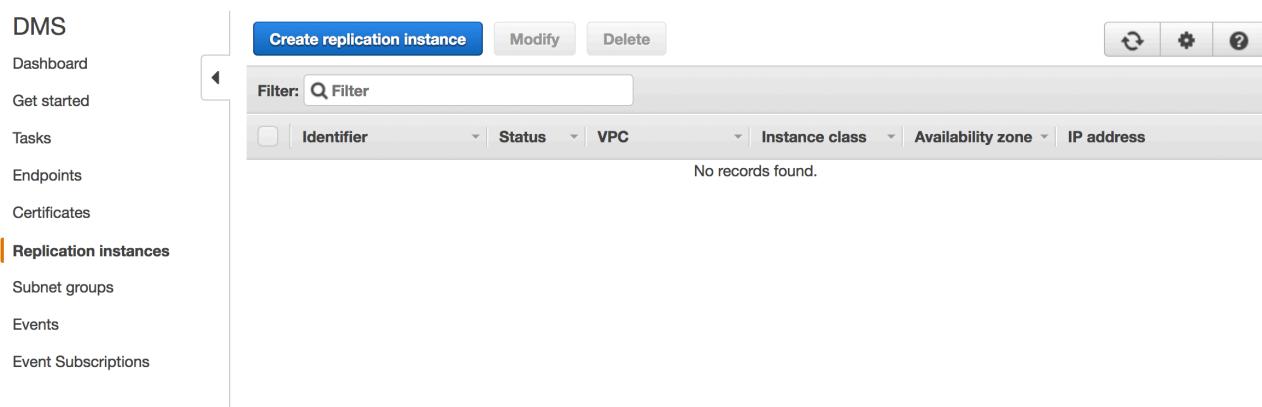


Figure 23: AWS DMS Replication Instances - List All

AWS DMS Replication Instances - Create New Next, click on **Create Replication Instance** button, and populate the values on this page:

Create replication instance

A replication instance initiates the connection between the source and target databases, transfers the data, and caches any changes that occur on the source database during the initial data load. Use the fields below to configure the parameters of your new replication instance including network and security information, encryption details, and performance characteristics. We suggest you shut down the replication instance once your migration is complete to prevent further usage charges.

The screenshot shows the 'Create New' configuration page for an AWS DMS Replication Instance. The form fields are as follows:

Name*	workshop-nosql	<small>i</small>
Description*	AWS DMS Workshop for NoSQL	<small>i</small>
Instance class*	dms.t2.medium	<small>i</small>
Replication engine version*	2.3.0	<small>i</small>
VPC*	vpc-55263831 - workshop-cfn-nosql	<small>i</small>
Multi-AZ	No	<small>i</small>
Publicly accessible	<input checked="" type="checkbox"/>	<small>i</small>

Below the form, there is a link labeled 'Advanced'.

Figure 24: AWS DMS Replication Instances - Create New

using the following values:

- **Name:** workshop-nosql
- **Description:** AWS DMS Workshop for NoSQL
- **Instance class:** dms.t2.medium
- **Replication engine version:** (leave as default)
- **VPC:** (select VPC created by CloudFormation stack, which should be named **workshop-cfn-nosql**)

CAUTION: The replication instance created here will continue to run until specifically deleted -- the steps for teardown are located in this document at [Teardown: AWS DMS Resources](#)

AWS DMS Replication Instances - Confirm After creation, the replication instance(s) will be provisioned and display status of **Creating**:

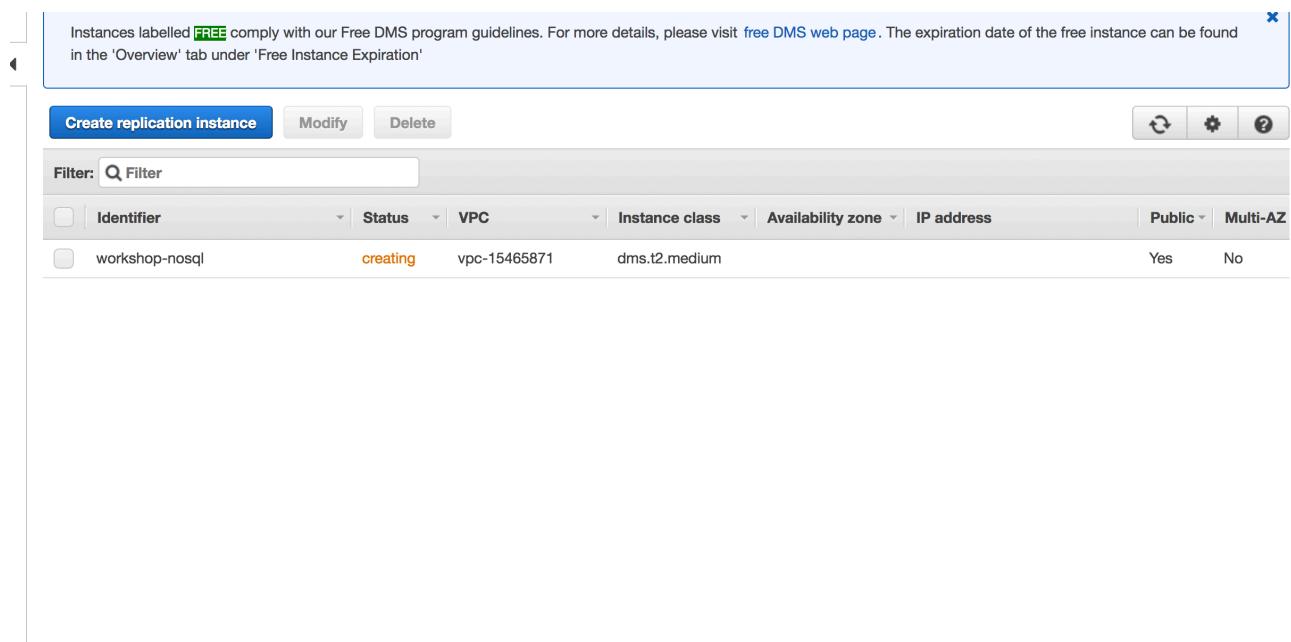


Figure 25: AWS DMS Replication Instances - List New Creating

...before changing to Available:

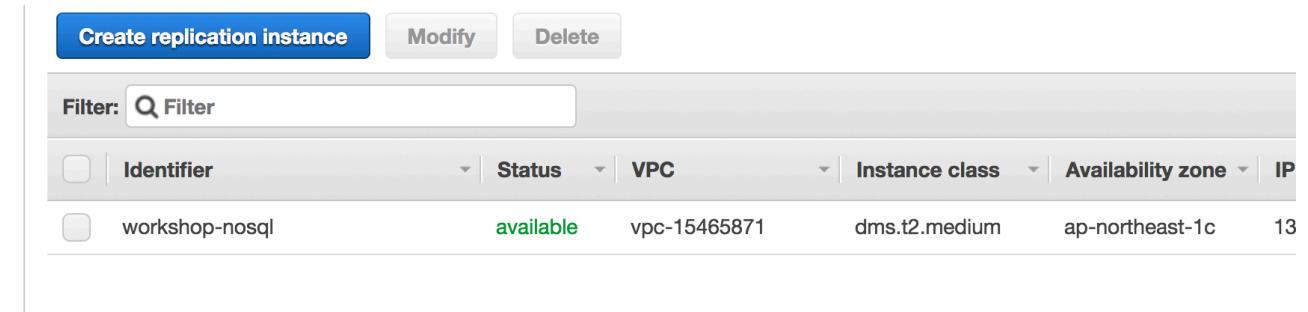


Figure 26: AWS DMS Replication Instances - List New Available

Now that the **DMS Replication Instance** server is ready and available to perform database migrations, we will next create **DMS Endpoints** to connect the Replication Instance to the Source and Target databases.

3.2.2 AWS DMS Endpoints

AWS DMS Endpoints - List All To create our new AWS DMS endpoints, we will first view the console for endpoints by visiting the following link:

<http://amzn.to/aws-tokyo-dms-endpoints> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#endpoints:>)

You should now see the following screen:

The screenshot shows the AWS DMS Endpoints management interface. On the left, a sidebar lists various DMS components: Dashboard, Get started, Tasks, Certificates, Replication instances, Subnet groups, Events, and Event Subscriptions. The 'Endpoints' option is selected and highlighted with an orange border. The main content area is titled 'AWS DMS Endpoints - List All'. It features a 'Create endpoint' button at the top left. Below it is a search/filter bar with fields for 'Identifier', 'Type', 'Status', 'Engine', 'Server name', 'Port', 'Migration Hub Mapping', and 'ARN', along with a 'Filter' dropdown and a 'Q' icon. The main table has no data, with the message 'No records found.' displayed below its header.

Figure 27: AWS DMS Endpoints - List All

Click the **Create Endpoint** button to proceed to creating the first endpoint.

AWS DMS Endpoints - Create Source You should now see the following page:

Create endpoint

AWS DMS accesses your data sources and targets using endpoints. A source endpoint allows AWS DMS to read data from a database (on-premise or in the cloud), or from a non-database source such as Amazon S3. A target endpoint allows AWS DMS to write data to a database, or to a non-database target.

We recommend that you choose "Run test" on this page, to verify that your endpoint is valid before using it in an AWS DMS task.

Endpoint type* Source Target [?](#)

Endpoint identifier* [?](#)

Source engine* [?](#)

Server name*

Port* [?](#)

SSL mode* [?](#)

User name* [?](#)

Password* [?](#)

[Advanced](#)

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC* [?](#)

Replication instance* [?](#)

Refresh schemas after successful connection test [?](#)

[Cancel](#)

Figure 28: AWS DMS Endpoints - Create Endpoint

Create endpoint

AWS DMS accesses your data sources and targets using endpoints. A source endpoint allows AWS DMS to read data from a database (on-premise or in the cloud), or from a non-database source such as Amazon S3. A target endpoint allows AWS DMS to write data to a database, or to a non-database target.

We recommend that you choose "Run test" on this page, to verify that your endpoint is valid before using it in an AWS DMS task.

The screenshot shows the 'Create Source Endpoint' configuration page. The 'Endpoint type*' field is set to 'Source'. The 'Endpoint identifier*' field contains 'dms-workshop-mongodb'. The 'Source engine*' dropdown is set to 'mongodb'. The 'Server name*' field is '10.0.0.11'. The 'Port*' field is '27017'. The 'SSL mode*' dropdown is 'none'. The 'Authentication mode*' dropdown is 'password'. The 'User name*' field is 'dms_user'. The 'Password*' field is obscured by dots. The 'Authentication source*' field is 'dms_sample'. The 'Database name*' field is 'dms_sample'. The 'Authentication mechanism*' dropdown is 'scram_sha-1'. The 'Metadata mode*' dropdown is 'document'. A checkbox for '_id as a separate column' is checked.

Figure 29: AWS DMS Endpoints - Create Source Endpoint (Details)

DMS Endpoint - Enter Data for Source Endpoint Enter the fields as follows:

- **Endpoint type:** (select default of Source)
- **Endpoint identifier:** dms-workshop-mongodb
- **Source engine:** mongodb
- **Server name:** (the MongoDBServerIP generated by the CloudFormation stack)
- **Port:** 27017
- **SSL mode:** none
- **Authentication mode:** password
- **User name:** dms_user
- **Password:** dms_user
- **Authentication source:** dms_sample

- **Database name:** dms_sample
- **Authentication mechanism:** scram_sha-1
- **Metadata mode:** document
- ****_id as a separate column**:** (selected/checked)

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC* vpc-55263831 - workshop-cfn-nosql

Replication instance* workshop-nosql - vpc-55263831

Refresh schemas after successful connection test

Run test

Create endpoint

Figure 30: AWS DMS Endpoints - Create Source Endpoint (Test)

For the test portion, fill the following details:

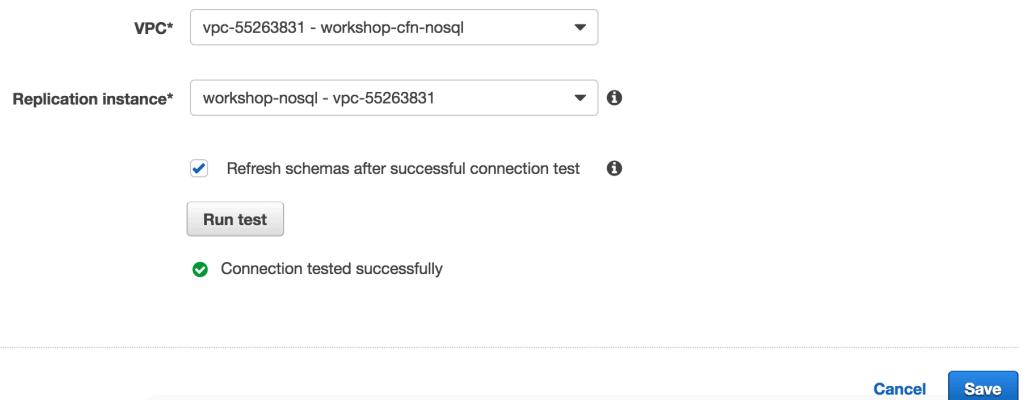
- **VPC:** (select the VPC marked with **workshop**)
- **Replication Instance:** (select the DMS Replication Instance created earlier)
- **Refresh schemas...:** (leave default of checked)

DMS Endpoint - Test Source Endpoint Selecting the **workshop** VPC and the **DMS Replication Instance** that lives in that VPC that you created earlier, you can now test the connection from that Replication Instance instance to the Source database via the **DMS Endpoint** settings above. Click the **Run Test** button to proceed. A successful test will display as Connection Successfully Tested as shown below:

▶ Advanced

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.



The screenshot shows the 'Create Source Endpoint' page in the AWS DMS console. The 'VPC*' dropdown is set to 'vpc-55263831 - workshop-cfn-nosql'. The 'Replication instance*' dropdown is set to 'workshop-nosql - vpc-55263831'. A checked checkbox labeled 'Refresh schemas after successful connection test' has an information icon next to it. Below these are two buttons: 'Run test' and 'Save'. The 'Run test' button is greyed out. The 'Save' button is blue. Below the buttons, a green checkmark indicates 'Connection tested successfully'.

Figure 31: AWS DMS Endpoints - Create Source Endpoint (Test Success)

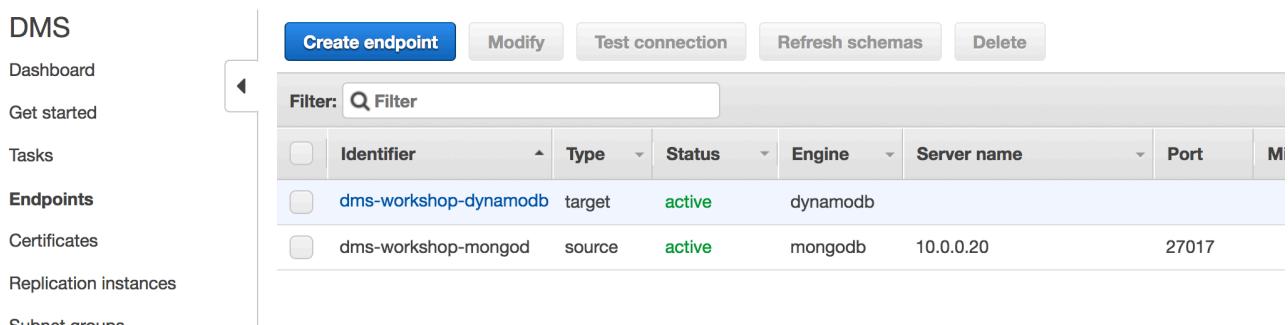
You can now click the **Save** button to save this endpoint.

Congratulations! You have successfully set up and tested the Source endpoint. We will now repeat the process for the Target endpoint.

AWS DMS Endpoints - List All (Updated) We will return to the DMS Endpoints view in the console by visiting the following link:

<http://amzn.to/aws-tokyo-dms-endpoints> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#endpoints:>)

You should now see the following screen:



The screenshot shows the 'List All' screen for AWS DMS Endpoints. On the left is a navigation sidebar with links: Dashboard, Get started, Tasks, Endpoints (which is selected), Certificates, Replication instances, and Subnet groups. The main area has a 'Create endpoint' button and several other buttons: Modify, Test connection, Refresh schemas, and Delete. There is a 'Filter' input field. A table lists two endpoints:

	Identifier	Type	Status	Engine	Server name	Port	
<input type="checkbox"/>	dms-workshop-dynamodb	target	active	dynamodb			
<input type="checkbox"/>	dms-workshop-mongod	source	active	mongodb	10.0.0.20	27017	

Figure 32: AWS DMS Endpoints - List All (Updated)

Again, we will click to **Create Endpoint**, seeing the Create Endpoint page as before.

Create endpoint

AWS DMS accesses your data sources and targets using endpoints. A source endpoint allows AWS DMS to read data from a database (on-premise or in the cloud), or from a non-database source such as Amazon S3. A target endpoint allows AWS DMS to write data to a database, or to a non-database target.

We recommend that you choose "Run test" on this page, to verify that your endpoint is valid before using it in an AWS DMS task.

The screenshot shows the 'Create endpoint' configuration page. The 'Endpoint type' is set to 'Target'. The 'Endpoint identifier' field contains 'dms-workshop-dynamodb'. The 'Target engine' dropdown is set to 'dynamodb'. The 'Service Access Role ARN' field contains 'arn:aws:iam::306280812807:role/dms-workshop-dyna'. Below these fields is a collapsed 'Advanced' section.

Figure 33: AWS DMS Endpoints - Create Target Endpoint (Details)

Enter the fields as follows:

- **Endpoint type:** Target
- **Endpoint identifier:** dms-workshop-dynamodb
- **Source engine:** dynamodb
- **Service Access Role ARN:** (this is the Service Role ARN you created earlier)

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

The screenshot shows the 'Test endpoint connection' section. The 'VPC' dropdown is set to 'vpc-15465871 - workshop'. The 'Replication instance' dropdown is set to 'workshop-nosql - vpc-15465871'. Below these is a 'Run test' button. At the bottom right are 'Cancel' and 'Create endpoint' buttons.

Figure 34: AWS DMS Endpoints - Create Target Endpoint (Test)

For the test portion, fill the following details:

- **VPC:** (select the VPC marked with **workshop**)
- **Replication Instance:** (select the DMS Replication Instance created earlier)
- **Refresh schemas...:** (leave default of checked)

DMS Endpoint - Test Target Endpoint Selecting the **workshop** VPC and the **DMS Replication Instance** that lives in that VPC that you created earlier, you can now test the connection from that Replication Instance instance to the Target database via the **DMS Endpoint** settings above. Click the **Run Test** button to proceed. A successful test will display as Connection Successfully Tested as shown below:

▼ Test endpoint connection (optional)

Test your endpoint connection by selecting a replication instance within your desired VPC. After clicking "Run test", an endpoint will be created with the details provided and attempt to connect to the instance. If the connection fails, you can edit and test it again. Endpoints that aren't saved will be deleted.

VPC*

Replication instance* ⓘ

✔ Connection tested successfully

Cancel Save

Figure 35: AWS DMS Endpoints - Create Target Endpoint (Test Success)

You can now click the **Save** button to save this endpoint. You should now see the list of DMS Endpoints with two endpoints in an Active status as shown below:

Endpoints						
	Identifier	Type	Status	Engine	Server name	Port
<input checked="" type="checkbox"/>	dms-workshop-dynamodb	target	active	dynamodb		
<input checked="" type="checkbox"/>	dms-workshop-mongod	source	active	mongodb	10.0.0.20	27017

Figure 36: AWS DMS Endpoints - List Endpoints (Source and Target)

You have now successfully created both DMS Endpoints, Source and Target, and you can now move to the creation of a DMS Task.

3.2.3 AWS DMS Tasks

The following link will display all **DMS Tasks** in this region

<http://amzn.to/aws-tokyo-dms-tasks> (<=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#tasks:>)

AWS DMS Tasks -- List All You should now see the following:

Figure 37: AWS DMS Tasks - List Tasks

AWS DMS Tasks - Create New You will now create a new AWS DMS Task by clicking the **Create Task** button, which will display the following page:

Create task

A task can contain one or more table mappings which define what data is moved from the source to the target. If a table does not exist on the target, it can be created automatically.

Task name*	dms-workshop-task-mongodb2dynamodb	
Replication instance*	workshop-nosql - vpc-15465871	
Source endpoint*	dms-workshop-mongodb	
Target endpoint*	dms-workshop-dynamodb	
Migration type*	Migrate existing data	
<input checked="" type="checkbox"/> Start task on create		

Figure 38: AWS DMS Tasks - Create Task (Detail)

Within this page, you will enter the following data:

- **Task name:** dms-workshop-task-mongodb2dynamodb
- **Replication instance:** (defaults to **DMS Replication Instance** you created earlier)
- **Source endpoint:** (defaults to **DMS Endpoint** for the MongoDB source database that you created earlier)
- **Target endpoint:** (defaults to **DMS Endpoint** for the DynamoDB target database that you created earlier)
- **Migration type:** Migrate existing data
- **Start task on create:** (leave this checked)

▼ Task Settings

Target table preparation mode* Do nothing ⓘ
 Drop tables on target ⓘ
 Truncate

Include LOB columns in replication* Don't include LOB columns ⓘ
 Full LOB mode
 Limited LOB mode

Max LOB size (kb)* ⓘ

Enable logging

[Advanced Settings](#)

Figure 39: AWS DMS Tasks - Create Task (Continued)

Continue by entering the following data:

- **Target table preparation mode:** Drop tables on target
- **Include LOB columns in replication:** Limited LOB mode
- **Max LOB size (kb):** 32
- **Enable logging:** selected/checked

AWS DMS Tasks - Add Selection Criteria Add the following selection criteria, as shown below:

▼ Table mappings

Guided **JSON**

Selection rules ⓘ

At least one selection rule with an include action is required. Once you have one or more selection rules, you can add transformation rules.

Where ⓘ

Schema name is ▾

Table name is like ⓘ

Use % as a wildcard.

Action ⓘ

Filter ⓘ

[Add column filter](#)

Add selection rule

The screenshot shows the 'Selection rules' section of the AWS DMS Tasks interface. It includes fields for 'Schema name is' (set to 'dms_sample'), 'Table name is like' (set to '%'), and 'Action' (set to 'Include'). There is also a 'Filter' section with a link to 'Add column filter'. A prominent blue button at the bottom right says 'Add selection rule'.

Figure 40: AWS DMS Tasks - Add Selection Criteria

AWS DMS Tasks - Create Task (Final) Review the information and click the **Create Task** button to continue.



Figure 41: AWS DMS Tasks - Create Task (Final)

You will now see the new DMS Task listed with a status of Creating, then Starting, then Running, as shown below

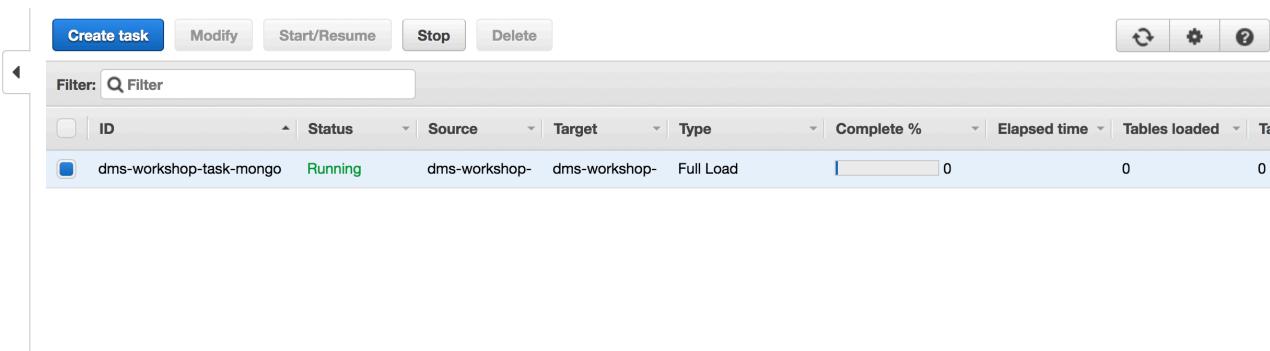


Figure 42: AWS DMS Tasks - List Tasks (Updated)

You have now successfully set up all the major components for a database migration with AWS DMS. Next, we will troubleshoot some errors that commonly occur during real-world migrations.

4 Lab Teardown

4.1 Teardown AWS DMS Resources

You will now destroy the previously created AWS DMS resources. Because there is an interdependency between these resources, we will terminate the resources in the reverse order of the original creation.

The order for destruction will be:

- Tasks
- Endpoints
- Replication Instances

You can find all of these resources under the AWS console for AWS Database Migration Services at the following link:

<http://amzn.to/aws-tokyo-dms-instances> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#replication-instances:>)

4.1.1 Teardown AWS DMS Resources: Tasks

First, you will delete the existing AWS DMS Task associated with this workshop. You can find this at the following link:

<http://amzn.to/aws-tokyo-dms-tasks> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#tasks:>)

You should see a list of all DMS Tasks in this region. Locate the DMS Task created earlier during this workshop. The default name for the Task created during this workshop was **dms-workshop-task-oracle2postgres**.

Select the task and, if it is running, click the **Stop** button.

Wait a few moments until the Task until the status is updated to Stop, then click the **Delete** button.

The process of deleting a DMS Task may take a minute or two. Continue to refresh until the DMS Task is no longer visible, then proceed to the next step.

4.1.2 Teardown AWS DMS Resources: Endpoints

Next, you can now delete the DMS Endpoints that were created earlier in this lab. You can find this at the following link:

<http://amzn.to/aws-tokyo-dms-endpoints> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#endpoints:>)

You should now see a list of all DMS Endpoints in this region. There will be two Endpoints to delete, one for the Source and one for the Target. The default names for this workshop are:

- Source: **dms-workshop-mongodb**
- Target: **dms-workshop-dynamodb**

If you used those defaults, your page should look like the following:

	Identifier	Type	Status	Engine	Server name	Port	Migration Hub Map
<input type="checkbox"/>	dms-workshop-dynamodb	target	active	dynamodb			
<input type="checkbox"/>	dms-workshop-mongod	source	active	mongodb	10.0.0.11	27017	

Figure 43: Teardown AWS DMS - List All Endpoints

Delete both Source and Target endpoints, confirming that you wish to delete. Finally, you will see the following page:

	Identifier	Type	Status	Engine	Server name	Port	Migration Hub Map
<input type="checkbox"/>	dms-workshop-dynamodb	target	deleting	dynamodb			
<input checked="" type="checkbox"/>	dms-workshop-mongod	source	deleting	mongodb	10.0.0.11	27017	

Figure 44: Teardown AWS DMS - Endpoints Deleted

The process of deleting DMS Endpoints may take a minute or two. Continue to refresh until the DMS Endpoints are no longer visible, then proceed to the next step.

4.1.3 Teardown AWS DMS Resources: Replication Instances

To delete the DMS replication instances for this workshop, we will first view the console for replication instances by visiting the following link:

<http://amzn.to/aws-tokyo-dms-instances> (=> <https://ap-northeast-1.console.aws.amazon.com/dms/home?region=ap-northeast-1#replication-instances:>)

Locate the DMS Replication Instance created earlier during this workshop. The default name for the Replication Instance created during this workshop was **dms-workshop-mongodb2dynamodb-repl**.

You will now select your workshop DMS Replication Instance, clicking the **Delete** button to proceed:

Click the **Delete** button again if you certain this is the correct DMS Replication Instance you set up earlier in the workshop.

The status of the Replication Instance will now show as **Deleted**

The process of deleting DMS Replication Instances may take a minute or two. Continue to refresh until the DMS Replication Instances are no longer visible, then proceed to the next step.

4.2 Teardown AWS CloudFormation Stack

For process of tearing down any resources created by a CloudFormation stack is a part of the CloudFormation lifecycle and can be performed.

CAUTION: This step must be performed after the teardown of any DMS resources -- see [Teardown AWS DMS Resources](#) for this process.

4.2.1 Teardown CloudFormation Stack: Identify Workshop Stack

You can see a list of all active CloudFormation Stacks in this region by clicking the following link:

<http://amzn.to/aws-tokyo-cloudformation-stacks-active> (=> <https://ap-northeast-1.console.aws.amazon.com/cloudformation/home?region=ap-northeast-1#/stacks?filter=active>)

If you have chosen the default CloudFormation stack name during the setup, then you should now select the check-box on that specific CloudFormation stack and click the button **Delete Stack**

4.2.2 Teardown CloudFormation Stack: Delete Workshop Stack

4.2.3 Teardown CloudFormation Stack: Confirm Deletion

*If your stack is still displayed with a **Delete in Progress** status, then please wait a few minutes for the CloudFormation stack to complete deletion.*

If the CloudFormation stack has been successfully deleted, and no other stacks are active in this region, you should expect to see the following:

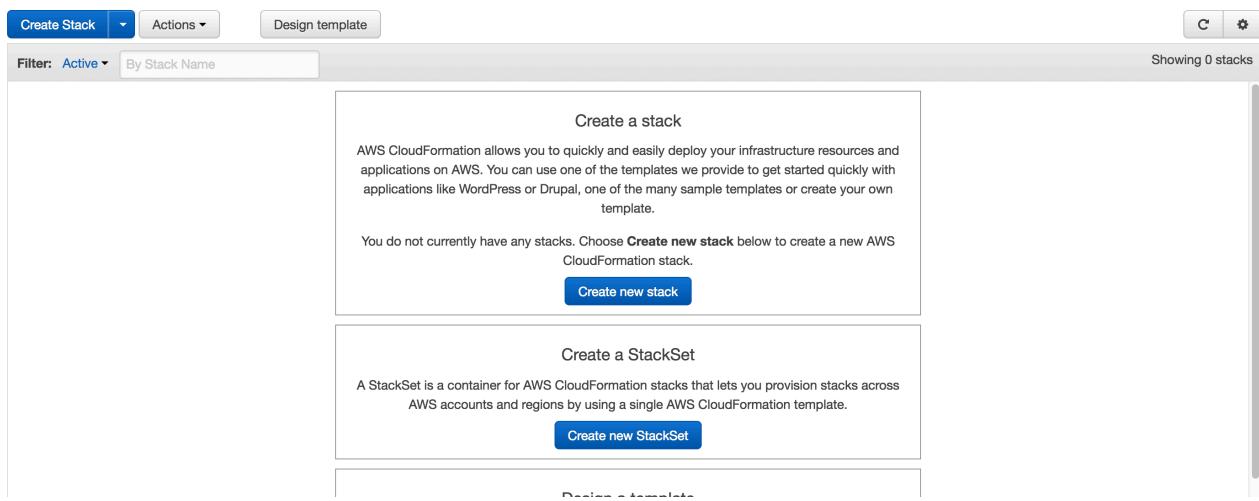


Figure 45: Teardown CloudFormation Stack: Delete Success

If the CloudFormation stack still exists, check the following remediations below based on the specific CloudFormation Stack Error:

- Status: Delete Failed

4.2.4 Teardown CloudFormation Stack Status: Delete Failed

If you should see the following message:

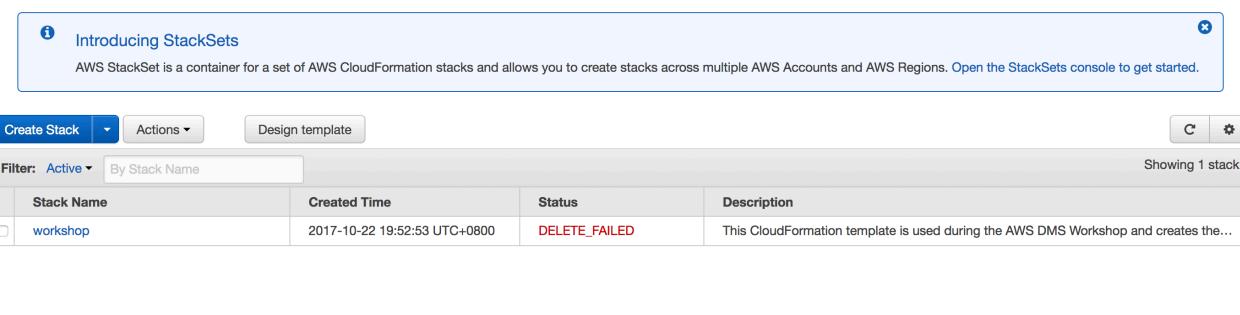


Figure 46: Teardown CloudFormation Stack: Delete Failed

First, confirm that the AWS DMS resources have been terminated -- see [Teardown AWS DMS Resources](#) for more details.

Second, you can retry the process to [Delete the Workshop Stack](#) above. You may be presented with a dialog similar to the following:

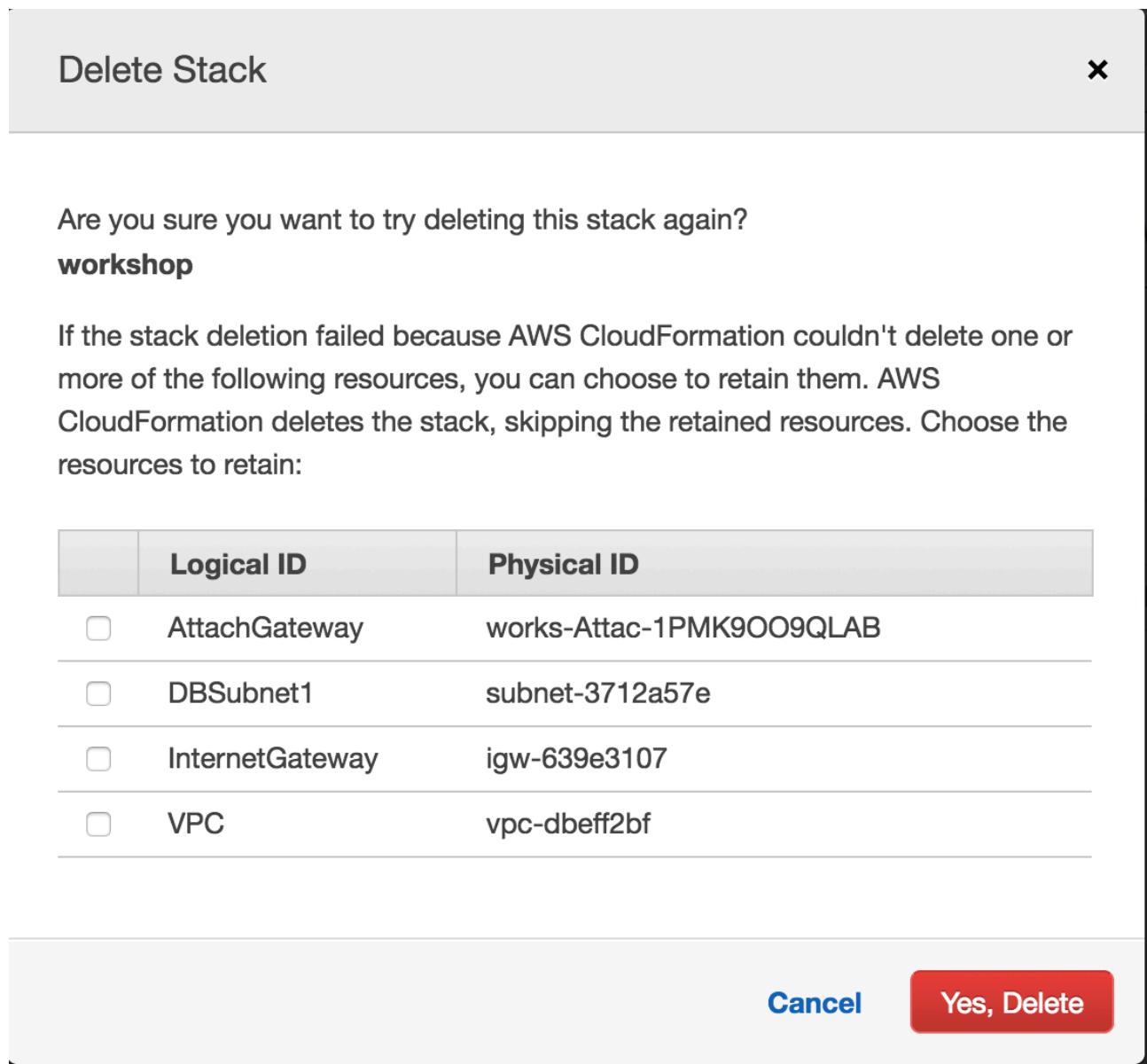


Figure 47: Teardown CloudFormation Stack: Confirm Delete Resources

If so, then individually confirm that the resources are no longer required, checking the box of each resource you wish to delete, then clicking the **Yes, Delete** button to complete the process.

4.3 Teardown EC2 Key Pair

If you had previously created an EC2 key pair for this workshop during [setup](#), it is recommended that you now delete it to ensure security.

Follow this link to access your list of EC2 key pairs in this region:

<http://amzn.to/aws-tokyo-keypairs> (=> <https://ap-northeast-1.console.aws.amazon.com/ec2/v2/home?region=ap-northeast-1#KeyPairs:sort=keyName>)

You will see a list of key pairs similar to the following:

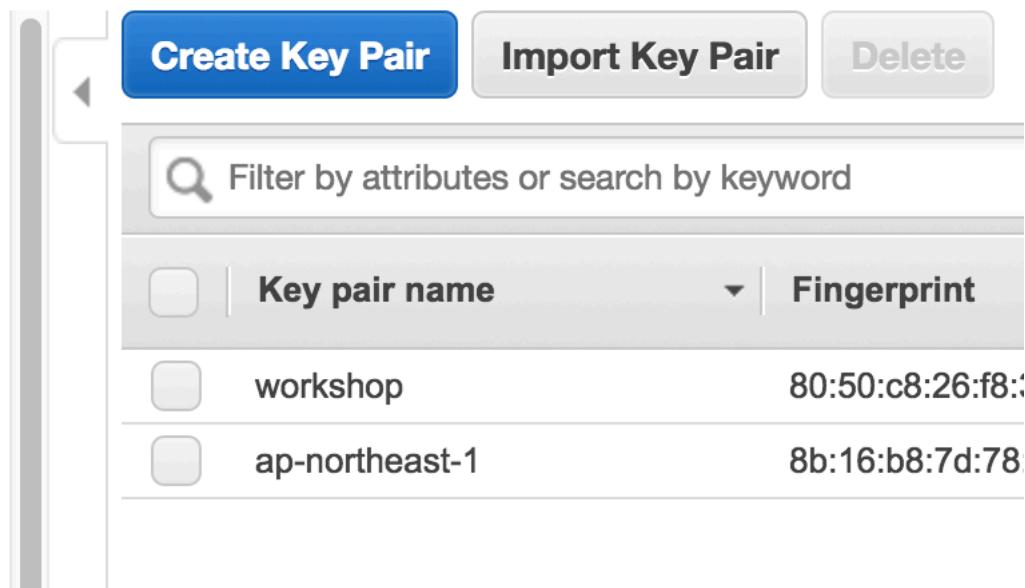


Figure 48: Teardown EC2 Key Pair - List Keys

Next, select the appropriate key pair (the default name for this workshop key pair was **workshop**), and click the **Delete** button, after which you will receive the following confirmation:

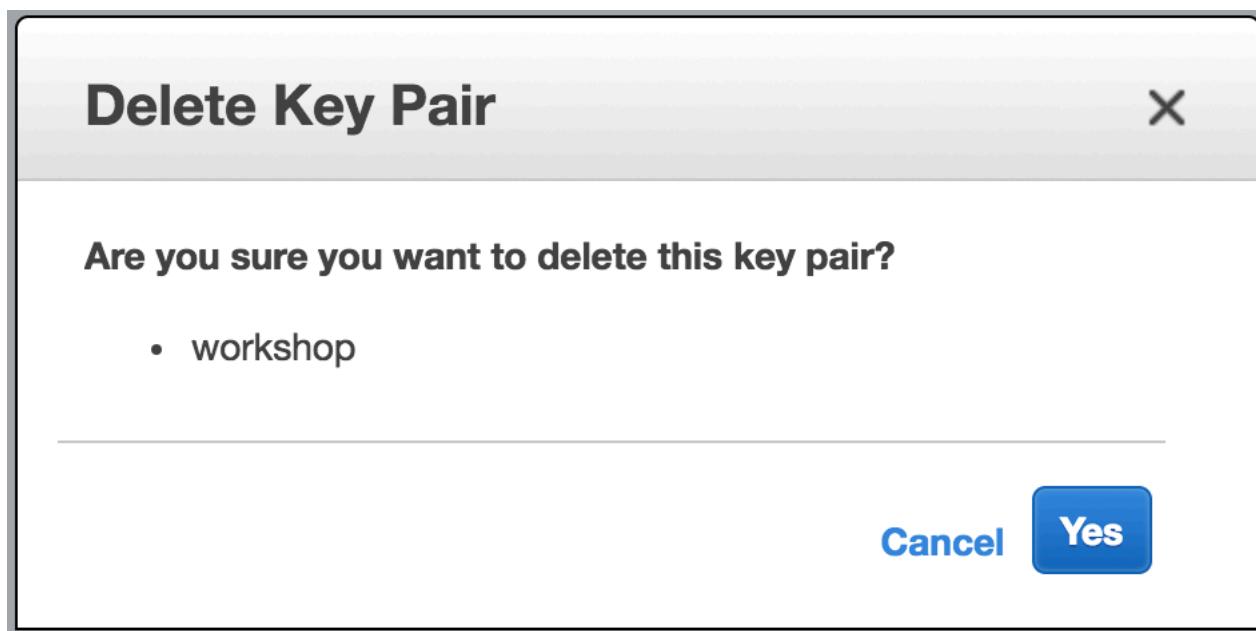


Figure 49: Teardown EC2 Key Pair - Delete Key

If you are certain this is the workshop key you created, then click **Yes** button to permanently delete this key.

CAUTION: the EC2 Key Pair will be permanently deleted and cannot be recovered, so any associated resources will no longer be administratively accessible.

4.4 Teardown DynamoDB

CAUTION: It is critical that you delete your DynamoDB tables to avoid incurring costs -- **the currently provisioned DynamoDB tables will cost approximately USD\$1400/month!**

The screenshot shows the AWS DynamoDB console dashboard. On the left, a navigation sidebar lists 'DynamoDB' (selected), 'Dashboard', 'Tables' (highlighted with an orange border), 'Reserved capacity', 'DAX', 'Dashboard', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main content area includes:

- Create table**: A button to start creating a new table.
- Recent alerts**: A section stating "No CloudWatch alarms have been triggered." with a link to "View all in CloudWatch".
- Total capacity for Asia Pacific (Tokyo)**: Capacity summary table:

Provisioned read capacity	2105	Reserved read capacity	0
Provisioned write capacity	2105	Reserved write capacity	0
- Service health**: A table showing the status of Amazon DynamoDB (Tokyo):

Current Status	Details
✓ Amazon DynamoDB (Tokyo)	Service is operating normally

 With a link to "View complete service health details".

Figure 50: Teardown DynamoDB: DynamoDB Console

Under Tables, you will see a number of tables that need to be deleted

The screenshot shows the AWS DynamoDB console 'Tables' page. On the left, a navigation sidebar lists 'DynamoDB' (selected), 'Dashboard', 'Tables' (highlighted with an orange border), 'Reserved capacity', 'DAX', 'Dashboard', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main content area displays a table of 12 tables:

Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity
awsdms_full_load_exceptions	Active	HashKey,RangeKey,TableName (-	0	100	100
mib_data	Active	_id (String)	-	0	200	200
name_data	Active	_id (String)	-	0	200	200
nfl_data	Active	_id (String)	-	0	200	200
nfl_stadium_data	Active	_id (String)	-	0	200	200
person	Active	_id (String)	-	0	200	200
sport	Active	_id (String)	-	0	200	200
sport_location	Active	_id (String)	-	0	200	200
sport_team	Active	_id (String)	-	0	200	200
sporting_event	Active	_id (String)	-	0	200	200
sporting_event_ticket	Active	_id (String)	-	0	200	200
test2	Active	id (String)	-	0	5	5

Figure 51: Teardown DynamoDB: DynamoDB Console

Select each table and click **Delete table**

The screenshot shows the AWS DynamoDB console interface. On the left, a sidebar lists 'DynamoDB', 'Tables' (selected), 'Reserved capacity', 'DAX', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main area displays a list of tables under 'Tables', including 'awsdms_full_load_exceptions' (selected), 'mlb_data', 'name_data', 'nfl_data', 'nfl_stadium_data', 'person', 'sport', 'sport_location', 'sport_team', 'sporting_event', 'sporting_event_ticket', and 'test2'. A context menu is open over the 'awsdms_full_load_exceptions' table, with options 'Import', 'Export', and 'Delete table' visible. The 'Delete table' option is highlighted with a yellow box. To the right, the details for the 'awsdms_full_load_exceptions' table are shown, including its stream details and table details.

Table Name	Primary partition key	Primary sort key
awsdms_full_load_exceptions	HashKey,RangeKey,TableName (String)	-

Stream details

- Stream enabled: No
- View type: -
- Latest stream ARN: -

Table details

Attribute	Value
Table name	awsdms_full_load_exceptions
Primary partition key	HashKey,RangeKey,TableName (String)
Primary sort key	-
Time to live attribute	DISABLED
Table status	Active
Creation date	October 25, 2017 at 10:12:13 PM UTC+8
Provisioned read capacity units	100 (Auto Scaling Disabled)
Provisioned write capacity units	100 (Auto Scaling Disabled)

Figure 52: Teardown DynamoDB: DynamoDB Console

Repeat for all tables that were created during the workshop

The screenshot shows the AWS DynamoDB console interface, similar to Figure 52. The sidebar and table list are identical. A context menu is open over the 'mlb_data' table, with 'Delete table' highlighted. To the right, the details for the 'mlb_data' table are shown.

Table Name	Primary partition key	Primary sort key
mlb_data	_id (String)	-

Stream details

- Stream enabled: No
- View type: -
- Latest stream ARN: -

Table details

Attribute	Value
Table name	mlb_data
Primary partition key	_id (String)
Primary sort key	-
Time to live attribute	DISABLED
Table status	Active
Creation date	October 25, 2017 at 10:12:24 PM UTC+8
Provisioned read capacity units	200 (Auto Scaling Disabled)
Provisioned write capacity units	200 (Auto Scaling Disabled)
Last decrease time	-
Last increase time	-
Storage size (in bytes)	0 bytes

Figure 53: Teardown DynamoDB: DynamoDB Console

Until there are no more tables remaining

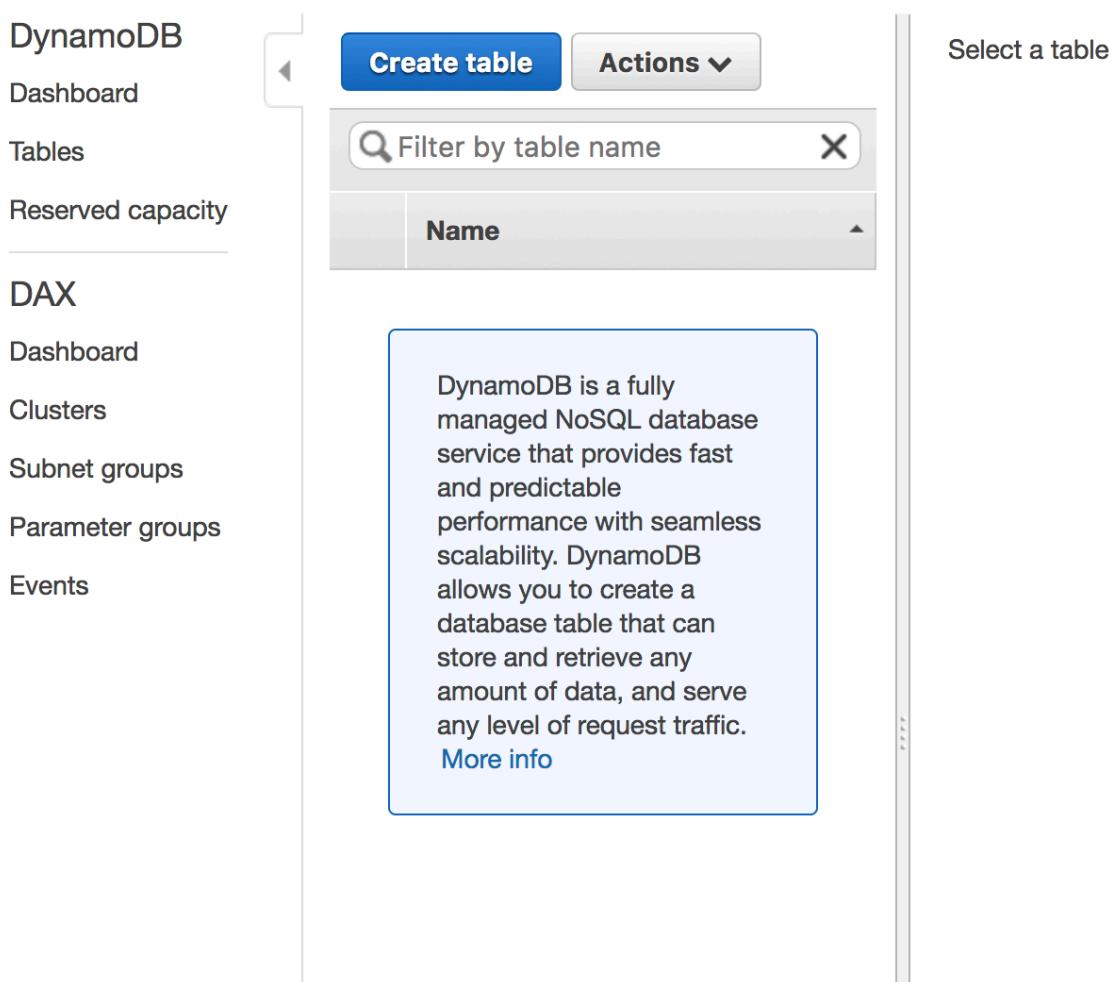


Figure 54: Teardown DynamoDB: DynamoDB Console

5 Conclusion

This concludes the workshop. We hope that you enjoyed it and learned from the experience. Please let us know what you enjoyed about the experience, and how we can improve this workshop, by filling out the survey.

Thank you for attending!