

# CQRS 워크샵 가이드 (배포용)

## 1. AWS 계정 환경설정

### 클라우드 네트워크 준비

AWS Console > VPC > Create VPC

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with options like 'VPC dashboard', 'EC2 Global View', 'Filter by VPC', and a dropdown for 'Select a VPC'. Below that is a section for 'Virtual private cloud' with links to 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways', and 'Egress-only Internet gateways'. In the main area, there are sections for 'Resources by Region' and 'Amazon VPC resources'. The 'Create VPC' button is highlighted with a red box. Below it, a note says 'Note: Your Instances will launch in the Asia Pacific region.' The 'Resources by Region' section shows counts for VPCs (2), Subnets (8), Route Tables (6), NAT Gateways (2), VPC Peering Connections (0), and Network ACLs (2). Each resource has a 'See all regions' link.

왼쪽의 메뉴 항목을 그림과 같이 설정하고 Create VPC 버튼 클릭

The screenshot shows the 'Create VPC' configuration wizard. The left panel contains several tabs: 'VPC settings' (selected), 'Preview', 'NAT gateways (1)', 'VPC endpoints', and 'DNS options'. The 'VPC settings' tab includes fields for 'Name tag auto-generation' (set to 'project'), 'IPv4 CIDR block' (set to '10.0.0.0/16'), 'Tenancy' (set to 'Default'), and 'Number of Availability Zones (AZs)' (set to 2). The 'Preview' tab shows a network diagram with four subnets ('ap-northeast-2a', 'ap-northeast-2b', 'ap-northeast-2c', 'ap-northeast-2d') connected to three route tables ('rtb-public', 'rtb-private1', 'rtb-private2') and four network connections ('project-lpe', 'project-rtb-public', 'project-rtb-private1', 'project-rtb-private2'). The 'Create VPC' button is at the bottom right of the wizard.

아래처럼 VPC 와 Subnet 4개가 생성된 것을 확인

VPC > Your VPCs > vpc-0cec5bfc0fbfa07cf / project-vpc

**Details**

VPC ID	vpc-0cec5bfc0fbfa07cf	State	Available
Tenancy	Default	DHCP option set	dopt-6d53f5fe4de0c4d9
Default VPC	No	IPv4 CIDR	10.0.0.0/16
Network mapping unit metrics	Disabled	Route 53 Resolver DNS Firewall rule groups	-
Owner ID	494059048196	DNS hostnames	Enabled
Main route table	rtb-0fb3c757e40ccf9ea	Main network ACL	ad-deb198ec6bb293654
IPv6 pool	-	IPv6 CIDR (Network border group)	-

**CIDRs**

Address type	CIDR	Network Border Group	Pool
IPv4	10.0.0.0/16	-	-

**Subnets (4) Info**

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
project-subnet-private1-ap-northeast-2a	subnet-09e3b252f6d398bef	Available	vpc-0cec5bfc0fbfa07cf   proj...	10.0.128.0/20	-
project-subnet-public1-ap-northeast-2a	subnet-011312b5936929d8d	Available	vpc-0cec5bfc0fbfa07cf   proj...	10.0.0.0/20	-
project-subnet-public2-ap-northeast-2b	subnet-0a1eeff7649bc0816c	Available	vpc-0cec5bfc0fbfa07cf   proj...	10.0.16.0/20	-
project-subnet-private2-ap-northeast-2b	subnet-055fb6a045c53dedf	Available	vpc-0cec5bfc0fbfa07cf   proj...	10.0.144.0/20	-

## CLOUD9 생성

AWS Console > Cloud9 > Create environment

Developer Tools

# AWS Cloud9

A cloud IDE for writing, running, and debugging code

AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.

New AWS Cloud9 environment

Create environment

Getting started

Environment의 이름 입력 후 Configure settings에서 아래 그림처럼 설정.

- Instance Type: m5. large
- Network settings (advanced) > Network(VPC) : 드롭다운에서 위에서 생성한 project-vpc 선택
- Network settings (advanced) > Subnet : 드롭다운에서 project-subnet-public1 또는 project-subnet-public2 선택

Step 1  
[Name environment](#)

Step 2  
**Configure settings**

Step 3  
[Review](#)

## Configure settings

### Environment settings

**Environment type** [Info](#)  
Run your environment in a new EC2 instance or an existing server. With EC2 instances, you can connect directly through Secure Shell (SSH) or connect via AWS Systems Manager (without opening inbound ports).

Create a new EC2 instance for environment (direct access)  
Launch a new instance in this region that your environment can access directly via SSH.

Create a new no-ingress EC2 instance for environment (access via Systems Manager)  
Launch a new instance in this region that your environment can access through Systems Manager.

Create and run in remote server (SSH connection)  
Configure the secure connection to the remote server for your environment.

**Instance type**

t2.micro (1 GiB RAM + 1 vCPU)  
Free-tier eligible. Ideal for educational users and exploration.

t3.small (2 GiB RAM + 2 vCPU)  
Recommended for small-sized web projects.

m5.large (8 GiB RAM + 2 vCPU)  
Recommended for production and general-purpose development.

Other instance type  
Select an instance type.  
t3.nano

**Platform**

Amazon Linux 2 (recommended)

Amazon Linux AMI

Ubuntu Server 18.04 LTS

**Cost-saving setting**  
Choose a predetermined amount of time to auto-hibernate your environment and prevent unnecessary charges. We recommend a hibernation settings of half an hour of no activity to maximize savings.

After 30 minutes (default)

**IAM role**  
AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. [Learn more](#)

AWSServiceRoleForAWSCloud9

**Network settings (advanced)**

**Network (VPC)**  
Launch your EC2 instance into an existing Amazon Virtual Private Cloud (VPC) or create a new one. To allow the AWS Cloud9 environment to connect to its EC2 instance, attach an internet gateway (IGW) to your new VPC.

project-vpc | vpc-Ocec5bfc0fbfa07cf

**Subnet**  
Select a public subnet in which the EC2 instance is created. (For a private subnet, you must create an environment that connects to its instance via Systems Manager.)

project-subnet-public1-ap-northeast-2a | subnet-0113...

No tags associated with the resource.

Add new tag

You can add 50 more tags.

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

이후 Cloud9에 자동으로 접속됨. 이제부터 코딩 및 터미널 사용은 개인 노트북이 아닌 Cloud9에서 진행.

Cloud9 화면의 하단 터미널에서 AWS Credential 설정 확인.

```
bash - "ip-172-31-15-79.x" ✘ Immediate ✘ +  
TeamRole:~/environment $ aws sts get-caller-identity  
{  
  "Account": "██████████",  
  "UserId": "AROAXGCBWZUCIW47KKUDG:MasterKey",  
  "Arn": "arn:aws:sts::██████████:assumed-role/TeamRole/MasterKey"  
}
```

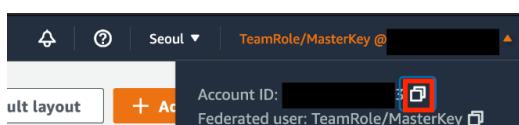
npm 업데이트 및 타입스크립트 설치

```
npm install -g npm@8.19.2 && npm install -g typescript
```

CDK 설치

```
npm install -g cdk  
cdk --version # 2.47.X
```

CDK 를 사용하기 위한 계정 설정 ("CDK bootstrap") (↓↓↓)



```
export ACCOUNT_ID="위의 그림 참고하여 Account ID 여기에 입력"  
cdk bootstrap aws://$ACCOUNT_ID/ap-northeast-2
```

CDK 코드 가져오기

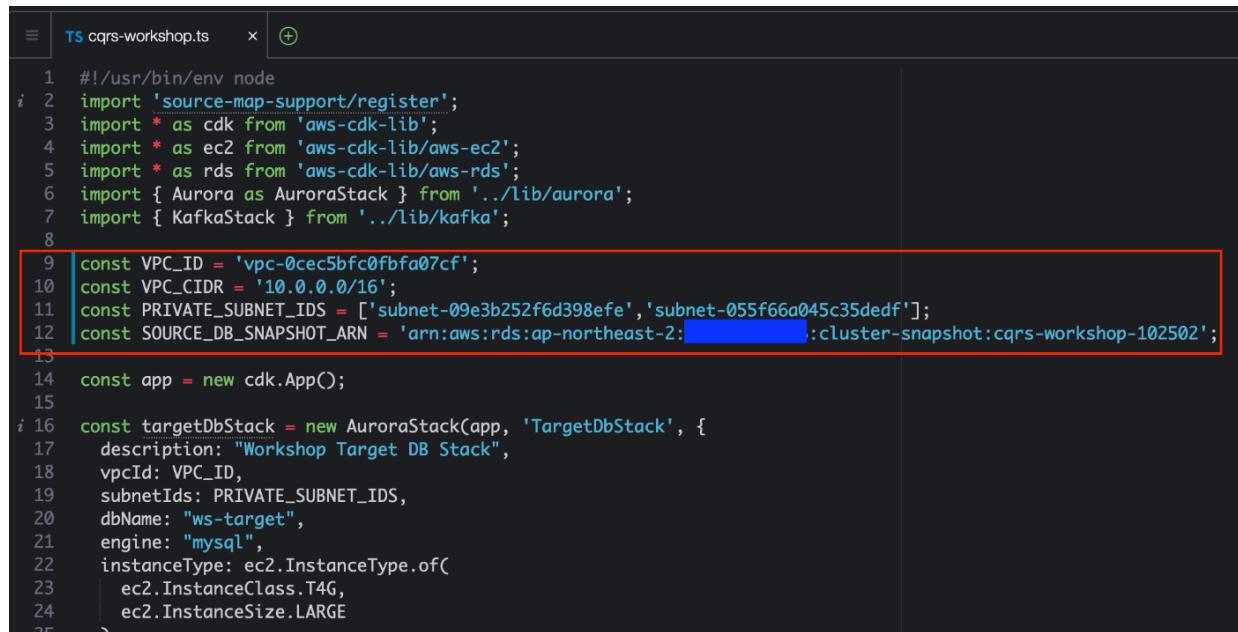
```
git clone https://github.com/eshluke/cqrs-dms-workshop.git  
cd cqrs-dms-workshop/cdk/
```

```
| npm install
```

## SOURCE & TARGET DB (AURORA MYSQL) 설치

(중요\*\*\*) cqrss-dms-workshop/cdk/bin/cqrss-workshop.ts 상단에 4가지 상수 값을 넣어야 함  
아래의 값은 예시로, 각자 계정의 값을 AWS Console에서 찾아서 입력해야 함 (그림 참고)

- VPC\_ID: 위에서 생성한 VPC ID
- VPC\_CIDR: 위에서 생성한 VPC의 CIDR 범위
- PRIVATE\_SUBNET\_IDS: 위에서 생성한 VPC의 Private Subnet 2개의 ID
- SOURCE\_DB\_SNAPSHOT\_ARN: 나에게 공유된 DB Snapshot ARN



```
 1 #!/usr/bin/env node
 2 import 'source-map-support/register';
 3 import * as cdk from 'aws-cdk-lib';
 4 import * as ec2 from 'aws-cdk-lib/aws-ec2';
 5 import * as rds from 'aws-cdk-lib/aws-rds';
 6 import { Aurora as AuroraStack } from '../lib/aurora';
 7 import { KafkaStack } from '../lib/kafka';
 8
 9 const VPC_ID = 'vpc-0cec5bfc0fbfa07cf';
10 const VPC_CIDR = '10.0.0.0/16';
11 const PRIVATE_SUBNET_IDS = ['subnet-09e3b252f6d398efe', 'subnet-055f66a045c35dedf'];
12 const SOURCE_DB_SNAPSHOT_ARN = 'arn:aws:rds:ap-northeast-2:[REDACTED]:cluster-snapshot:cqrss-workshop-102502';
13
14 const app = new cdk.App();
15
16 const targetDbStack = new AuroraStack(app, 'TargetDbStack', {
17   description: "Workshop Target DB Stack",
18   vpcId: VPC_ID,
19   subnetIds: PRIVATE_SUBNET_IDS,
20   dbName: "ws-target",
21   engine: "mysql",
22   instanceType: ec2.InstanceType.of(
23     ec2.InstanceClass.T4G,
24     ec2.InstanceSize.LARGE
25   )
26 })
```

**Your VPCs (1/2) Info**

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> project-vpc	vpc-0cec5bfc0fbfa07cf	Available	10.0.0.0/16	-
<input type="checkbox"/>	vpc-06a3b212dce72d871	Available	172.31.0.0/16	-

**vpc-0cec5bfc0fbfa07cf / project-vpc**

**Details** | CIDs | Flow logs | Tags

**Details**

VPC ID <input checked="" type="text"/> vpc-0cec5bfc0fbfa07cf	State Available	DNS hostnames Enabled
Tenancy Default	DHCP option set dopt-0d65f3fe4de0c4d9	Main route table rtb-0f83c757e40ccf9ea
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
Network mapping unit metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID <input checked="" type="text"/> 494059048196

**Subnets (1/4) Info**

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> project-subnet-private1-ap-northeast-2a	subnet-09e3b252f6d398efe	Available	vpc-0cec5bfc0fbfa07cf   project-vpc	10.0.128.0/20	-
<input type="checkbox"/> project-subnet-public1-ap-northeast-2a	subnet-011312b5936929d8d	Available	vpc-0cec5bfc0fbfa07cf   project-vpc	10.0.0.0/20	-
<input type="checkbox"/> project-subnet-public2-ap-northeast-2b	subnet-0a1ef7649bc0816c	Available	vpc-0cec5bfc0fbfa07cf   project-vpc	10.0.16.0/20	-
<input type="checkbox"/> project-subnet-private2-ap-northeast-2b	subnet-05f66a045c35def	Available	vpc-0cec5bfc0fbfa07cf   project-vpc	10.0.144.0/20	-

**subnet-09e3b252f6d398efe / project-subnet-private1-ap-northeast-2a**

**Details** | Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

**Details**

Subnet ID <input checked="" type="text"/> subnet-09e3b252f6d398efe	Subnet ARN <input checked="" type="text"/> arn:aws:ec2:ap-northeast-2:494059048196:subnet/subnet-09e3b252f6d398efe	State Available	IPv4 CIDR <input checked="" type="text"/> 10.0.128.0/20
Available IPv4 addresses 4088	IPv6 CIDR -	Availability Zone <input checked="" type="text"/> ap-northeast-2a	Availability Zone ID <input checked="" type="text"/> apne2-az1
Network border group ap-northeast-2	VPC vpc-0cec5bfc0fbfa07cf   project-vpc	Route table rtb-04629b83e5e3c29ac   project-rtb-private1-ap-northeast-2a	Network ACL acl-0eb198ecccbb2!
Default subnet No	Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign custom No

Amazon RDS > Shared snapshots (1)

**Shared snapshots (1)**

Snapshot name: arnawsrdsap-northeast-2... :cluster-snapshot:cqrs-workshop-102502

DB instance or cluster: cqrs-workshop

Snapshot creation time: October 25, 2022, 13:45 (UTC+09:00)

DB Instance created time: October 25, 2022, 13:11 (UTC+09:00)

Amazon RDS > Shared snapshots > arnawsrdsap-northeast-2... :cluster-snapshot:cqrs-workshop-102502

**Details**

ARN: arnawsrdsap-northeast-2... :cluster-snapshot:cqrs-workshop-102502

Instance/Cluster Name: N/A

Snapshot Creation Time: October 25, 2022, 13:45 (UTC+09:00)

Instance/Cluster Creation: October 25, 2022, 13:11 (UTC+09:00)

Cloud9에서 아래를 실행하여 Target DB 와 Source DB 생성

```
# cd cqrs-dms-workshop/cdk/
cdk deploy TargetDbStack # 15분
cdk deploy SourceDbStack # 15분
```

AWS Console > RDS > Databases 메뉴에서 DB 생성 확인

Amazon RDS > Databases

**Databases**

DB identifier: sourcedbstack-auroradatabase5475d328-3m8xtpwz15h

Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Main
Regional cluster	Aurora MySQL	ap-northeast-2	1 instance	Available	-	-	none
Writer instance	Aurora MySQL	ap-northeast-2b	db.r6g.large	Available	7.92%	2 Selects/Sec	none
Regional cluster	Aurora MySQL	ap-northeast-2	1 instance	Available	-	-	none
Writer instance	Aurora MySQL	ap-northeast-2a	db.r6g.large	Available	7.96%	2 Selects/Sec	none

## AMAZON MSK 클러스터 설치

Cloud9에서 아래를 실행하여 MSK 클러스터 생성

```
# cd cqrs-dms-workshop/cdk/
cdk deploy KafkaStack # 30분
```

## 2. DMS Replication Task 설정

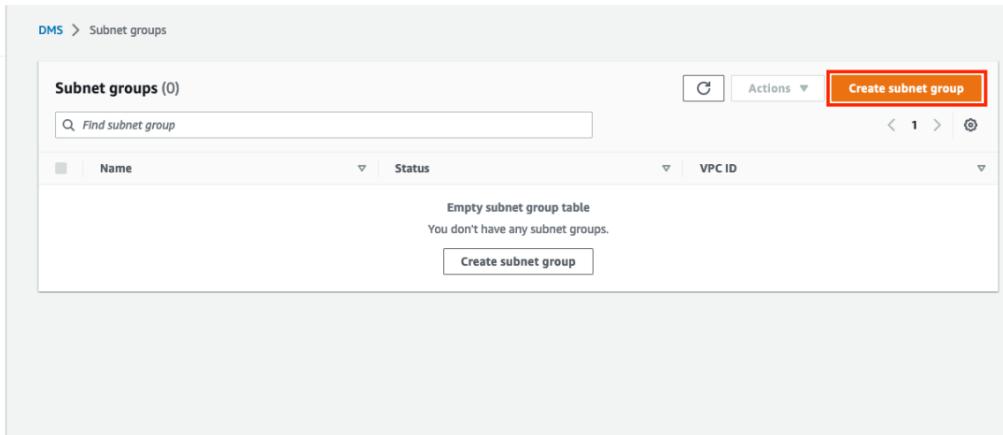
### DMS TASK 1 설정 (SOURCE DB → TARGET DB)

Source DB에서 Target DB로 데이터 동기화를 설정합니다 (Full Load + CDC).

## 1) DMS Subnet Group 생성

먼저 Replication Instance가 위치할 DMS Subnet Group을 생성합니다.

AWS Console에 접속하여 DMS > Subnet groups로 이동합니다. Create subnet group 버튼을 클릭합니다.



The screenshot shows the AWS DMS console with the 'Subnet groups' page open. On the left, there's a sidebar with options like 'Dashboard', 'Database migration tasks', 'Replication instances', 'Endpoints', 'Certificates', and 'Subnet groups'. The 'Subnet groups' option is highlighted with a red box. The main area shows a table titled 'Subnet groups (0)' with a single row labeled 'Empty subnet group table' and 'You don't have any subnet groups.' At the bottom right of the table area, there's a prominent 'Create subnet group' button, also highlighted with a red box.

상세 정보를 그림과 같이 입력한 뒤 하단의 Create subnet group 버튼을 클릭합니다.

- **Name:** 서브넷 그룹 이름
- **Description:** 서브넷 그룹 설명 (용도, 관리조직, etc)
- **VPC:** 위에서 생성한 project-vpc를 선택
- **Add subnets:** project-subnet-private1과 project-subnet-private2를 선택

DMS > Subnet groups > Create subnet group

### Create replication subnet group

**Subnet group configuration**

Name  
A regionally scoped unique identifier you will use to identify your Replication Subnet Group  
**cqrs-dms-subnet-group**

Description  
Free form text to describe your Replication Subnet Group  
**for cqrs-dms-workshop**

VPC  
**vpc-0cec5bfc0fbfa07cf - project-vpc**

**Add subnets**

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

subnet-09e3b252f6d398efe - project-subnet-private1-ap-northeast-2a ap-northeast-2a 10.0.128.0/20 Private	X
subnet-055f66a045c55dedf - project-subnet-private2-ap-northeast-2b ap-northeast-2b 10.0.144.0/20 Private	X

▶ Tags

Cancel **Create subnet group**

서브넷 그룹 생성을 확인합니다.

⌚ cqrs-dms-subnet-group created successfully. X

DMS > Subnet groups

**Subnet groups (1)**

<input type="checkbox"/>	Name	Status	VPC ID
<input type="checkbox"/>	<b>cqrs-dms-subnet-group</b>	Complete	vpc-0cec5bfc0fbfa07cf

## 2) DMS Replication Instance 생성

AWS Console에 접속하여 DMS > Replication Instances 메뉴로 이동한 후 Create replication instance 버튼을 클릭합니다.

The screenshot shows the AWS DMS console with the 'Replication instances' page open. The left sidebar includes options like 'Dashboard', 'Database migration tasks', 'Replication instances' (which is selected and highlighted with a red box), 'Endpoints', 'Certificates', 'Subnet groups', 'Events', and 'Event subscriptions'. The main area displays a table header for 'Replication instances (0)' with columns for Name, Status, VPC, Class, Engine version, Availability zone, Public, Public IP address, and Private IP address. Below the table, a message says 'Empty replication instance table' and 'You don't have any replication instances.' A large 'Create replication instance' button is located at the bottom right of the table area.

아래 그림처럼 입력합니다. 하단의 Create 버튼을 누릅니다.

- **Name:** Replication Instance 이름
- **Description:** 인스턴스의 사용목적, 관리자 등
- **Instance class:** 인스턴스 스펙 (실습용도의 스펙)
- **VPC:** 위에서 생성한 project-vpc 선택
- **Multi AZ:** 인스턴스 이중화 여부 (실습용으로는 이중화 하지 않음)
- **Publicly Accessible:** Private Subnet 내에서 진행하므로 체크 해제
- **Replication subnet group:** 위에서 생성한 cqrs-dms-subnet-group

The screenshot shows the 'Create replication instance' wizard. The current step is 'Advanced security and network configuration'. It includes fields for 'Replication subnet group' (set to 'cqrs-dms-subnet-group'), 'Availability zone' (set to 'No Preference'), 'VPC security group(s)' (set to 'Use default'), 'KMS key' (set to 'aws/dms'), and 'Maintenance' and 'Tags' sections. At the bottom, there are 'Cancel' and 'Create' buttons. On the left, the previous steps ('Replication instance configuration', 'Engine version', 'Allocated storage (GiB)', 'VPC', 'Multi AZ', and 'Publicly accessible') are visible, each with its own set of configuration options.

Replication Instance 가 생성된 것을 확인합니다.

Replication instances (1)											
	Name	Status	VPC	Class	Engine version	Availability zone	Public	Public IP address	Private IP address	Multi AZ	
	cqrss-dms-ri	Available	vpc-0cec5bfc0fbfa07cf	dms.t3.large	3.4.7	ap-northeast-2a	No	-	10.0.141.157	No	

### 3) DMS Source Endpoint 생성

Source DB 에 연결할 DMS Endpoint 를 생성합니다.

AWS Console > DMS > Endpoints 로 이동합니다. Create endpoint 버튼을 누릅니다.

The screenshot shows the AWS DMS console with the 'Endpoints' section selected. On the left sidebar, 'Endpoints' is highlighted with a red box. In the main content area, the 'Endpoints (0)' table has a single row: 'Empty endpoint table. You don't have any endpoints.' Below the table is a 'Create endpoint' button, which is also highlighted with a red box.

아래 그림과 같이 입력합니다.

- **Source endpoint 선택**
- **Select RDS DB instance 선택**
- **RDS Instance:** 드롭다운에서 ws-source1 선택
- **Endpoint identifier:** cqrss-task1-source
- **Access to endpoint database**에서 Provide access information manually 선택
- 아래에 DB 정보가 자동으로 입력된 것을 확인할 수 있습니다. (Password 의 값은 세션중 공유)

DMS > Endpoints > Create endpoint

## Create endpoint

**Endpoint type** [Info](#)

**Source endpoint**  
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

**Target endpoint**  
A target endpoint allows AWS DMS to write data to a database, or to other data source.

**Select RDS DB instance**

RDS Instance  
Instances available only for current user and region

**Endpoint configuration**

**Endpoint identifier** [Info](#)  
A label for the endpoint to help you identify it.

Descriptive Amazon Resource Name (ARN) - optional  
A friendly name to override the default DMS ARN. You cannot modify it after creation.

Source engine  
The type of database engine this endpoint is connected to. [Learn more](#)

Access to endpoint database

AWS Secrets Manager

Provide access information manually

Server name  
The name of the data server for the data provider.

Port  
The port the database runs on for this endpoint.

User name [Info](#)  Password [Info](#)

Secure Socket Layer (SSL) mode  
The type of Secure Socket Layer enforcement

**Endpoint settings**

위의 내용대로 작성한 뒤 하단의 Create endpoint 버튼을 누릅니다.

Endpoints 목록 화면에서 생성한 cqrs-task1-source 엔드포인트를 체크하고, Actions > Test connection 을 클릭합니다.

DMS > Endpoints

**Endpoints (2)**

Name	Type	Status	Engine	Server name	Port	Action
<input checked="" type="checkbox"/> cqrs-task1-source	Source	<span>Active</span>	Amazon Aurora MySQL	ws-source1.ctfsvqlzovzd.ap-northeast-2.rds.amazonaws.com	3306	<a href="#">Actions</a> <a href="#">Create endpoint</a>
<input type="checkbox"/> cqrs-task1-target	Target	<span>Active</span>	Amazon Aurora MySQL	AWS Secrets Manager credentials	-	<a href="#">Actions</a> <a href="#">Create endpoint</a>

**Actions** [Create endpoint](#) [Modify](#) [Delete](#) [Amazon Hub Mapping](#)

[Test connection](#)

인스턴스에서 엔드포인트까지 연결되는 것을 확인합니다.

DMS > Endpoints > cqrstask1-source

cqrstask1-source

Connections

Endpoint identifier	Replication instance	Status	Message
cqrstask1-source	cqrstask1-ri	successful	<button>Test connections</button>

#### 4) DMS Target Endpoint 생성

Target DB 에 연결할 DMS Endpoint 를 생성합니다.

AWS Console > DMS > Endpoints 로 이동합니다. Create endpoint 버튼을 누릅니다.

AWS DMS

DMS > Endpoints

Endpoints (0)

Name	Type	Status	Engine	Server name	Port	Migration Hub Mapping	ARN	Certificate ARN
Empty endpoint table You don't have any endpoints.								

Create endpoint

아래 그림과 같이 입력합니다.

- **Target endpoint** 선택
- **Select RDS DB instance** 선택
- **RDS Instance:** 드롭다운에서 ws-target1 선택
- **Endpoint identifier:** cqrstask1-target
- **Access to endpoint database**에서 Provide access information manually 선택
- 아래에 DB 정보가 자동으로 입력된 것을 확인할 수 있습니다. (Password 의 값은 세션중 공유)

위의 내용대로 작성한 뒤 하단의 Create endpoint 버튼을 누릅니다.

DMS > Endpoints > Create endpoint

## Create endpoint

**Endpoint type** [Info](#)

Source endpoint  
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

Target endpoint  
A target endpoint allows AWS DMS to write data to a database, or to other data source.

Select RDS DB instance

RDS Instance  
Instances available only for current user and region  
ws-target1

**Endpoint configuration**

**Endpoint identifier** [Info](#)  
A label for the endpoint to help you identify it.  
cqrss-task1-target

Descriptive Amazon Resource Name (ARN) - optional  
A friendly name to override the default DMS ARN. You cannot modify it after creation.  
Friendly-ARN-name

Target engine  
The type of database engine this endpoint is connected to. [Learn more](#)  
Amazon Aurora MySQL

Access to endpoint database  
 AWS Secrets Manager  
 Provide access information manually

Server name  
The name of the data server for the data provider.  
ws-target1.ctfsvqlzovzd.ap-northeast-2.rds.amazonaws.com

Port  
The port the database runs on for this endpoint.  
3306

User name [Info](#)  
admin

Password [Info](#)

Secure Socket Layer (SSL) mode  
The type of Secure Socket Layer enforcement  
none

**Endpoint settings**

Endpoints 목록 화면에서 생성한 cqrss-task1-target 엔드포인트를 체크하고, Actions > Test connection 을 클릭합니다.

DMS > Endpoints

**Endpoints (2)**

Name	Type	Status	Engine	Server name	Port	Migration Hub Mapping	ARN
cqrss-task1-target	Target	Active	Amazon Aurora MySQL	AWS Secrets Manager credentials	-		arn:aws:dms:ap-northeast-2:123456789012:target-ws-target1

**Actions** [Create endpoint](#)

- [Modify](#)
- Test connection**
- [Delete](#)

드롭다운에서 cqrss-dms-ri 인스턴스를 선택한 뒤 Run test 버튼을 클릭합니다.

DMS > Endpoints > ws-source1 > Test endpoint connection

### Test endpoint connection

Replication instance  
A replication instance performs the database migration

Endpoint identifier	Replication instance	Status	Message
	No records found		

[Back](#)

인스턴스에서 엔드포인트에 정상적으로 연결된 경우 아래와 같이 화면에 표시됩니다.

DMS > Endpoints > cqrss-task1-target > Test endpoint connection

### Test endpoint connection

Replication instance  
A replication instance performs the database migration

Endpoint identifier	Replication instance	Status	Message
cqrss-task1-target	cqrss-dms-ri	successful	

[Back](#)

## 5) DMS Task 생성

AWS Console > Database migration tasks로 이동하고 Create task 버튼을 클릭합니다.

AWS DMS

DMS > Database migration tasks

Database migration tasks (0)

Create task

Empty replication task table  
You don't have any replication tasks.

Create database migration task

Task configuration에 아래 내용을 입력합니다.

- Task-identifier:** cqrss-task1
- Replication instance:** 위에서 생성한 인스턴스 선택 (cqrss-dms-ri)
- Source database endpoint:** cqrss-task1-source

- **Target database endpoint:** cqrs-task1-target
- **Migration type:** Migrate existing data and replicate ongoing changes

DMS > Database migration tasks > Create database migration task

### Create database migration task

**Task configuration**

Task identifier  
cqrs-task1

Descriptive Amazon Resource Name (ARN) - optional  
A friendly name to override the default DMS ARN. You cannot modify it after creation.  
Friendly-ARN-name

Replication instance  
cqrs-dms-ri - vpc-Ocec5bfc0fbfa07cf

**i Upgrades to versions 3.4.7 and higher**  
You have 1 instance that uses AWS DMS version 3.4.7. Upgrades to AWS DMS versions 3.4.7 and higher require that you configure AWS DMS to use VPC endpoints or use public routes. This requirement applies to source and target endpoints for these data stores: S3, Kinesis, Secrets Manager, DynamoDB, Amazon Redshift, and OpenSearch Service. [Learn more](#)

View endpoints

Source database endpoint  
cqrs-task1-source

Target database endpoint  
cqrs-task1-target

Migration type [Info](#)  
Migrate existing data and replicate ongoing changes

Task settings에서 아래의 그림처럼 설정합니다.

- **Target table preparation mode:** Truncate
- 나머지는 디폴트 값 그대로 사용

**Task settings**

**Editing mode** [Info](#)

**Wizard**  
You can enter only a subset of the available task settings.

**JSON editor**  
You can enter all available task settings directly in JSON format.

**Target table preparation mode** [Info](#)

**Do nothing**  
 **Drop tables on target**  
 **Truncate**

**⚠ Possible data loss on target database**  
Truncate is the table preparation mode that you chose. Before migrating your source data, DMS leaves existing target tables and their metadata in place, but deletes all existing data from these tables before starting the migration.

**Stop task after full load completes** [Info](#)

**Don't stop**  
 **Stop before applying cached changes**  
 **Stop after applying cached changes**

**Include LOB columns in replication** [Info](#)

**Don't include LOB columns**  
 **Full LOB mode**  
 **Limited LOB mode**

**Maximum LOB size (KB)** [Info](#)

32

**Turn on validation**  
Choose this setting if you want AWS DMS to compare the data at the source and the target immediately after it performs a full data load. Validation ensures that your data was migrated accurately, but it requires additional time to complete.

**Task logs** [Info](#)

**Turn on CloudWatch logs**  
DMS task logging uses Amazon CloudWatch to log information during the migration process. You can change the component activities logged and the amount of information logged for each one.

**Batch-optimized apply** [Info](#)

**Turn on batch-optimized apply**  
Choose this setting if you want AWS DMS to efficiently group changes and commit them in batches. Without batch-optimized apply turned on, AWS DMS applies each change as an individual transaction. When migrating data to an Amazon Redshift target endpoint, AWS DMS sets batch-optimized apply by default because ongoing commits to an OLAP engine can become expensive.

**▶ Advanced task settings**

Table Mappings에서 Wizard를 선택하고 Add new selection rule 버튼을 클릭합니다.  
그리고 아래와 같이 입력합니다.

- **Schema:** Enter a schema
- **Source name:** `diner`
- **Source table name:** %
- **Action:** Include

**Table mappings**

Editing mode: [Info](#)

Wizard  
You can enter only a subset of the available table mappings.

JSON editor  
You can enter all available table mappings directly in JSON format.

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ Selection rules

Choose the schema and/or tables you want to include with, or exclude from, your migration task. [Info](#)

[Add new selection rule](#)

▼ where schema name is like 'diner' and Source table name is like '%', include

Schema  
[Enter a schema](#)

Source name  
Use the % character as a wildcard

Source table name  
Use the % character as a wildcard

Action  
Choose "Include" to migrate your selected objects, or "Exclude" to ignore them during the migration.  
 Include

Source filters [Info](#)      Add column filter

▶ Transformation rules

Migration task startup configuration 항목의 Start migration task 를 “Manually later” 로 지정합니다 (그림).  
하단 Create task 버튼을 클릭합니다.

**Migration task startup configuration**

Start migration task

Automatically on create  
Available only if the premigration assessment is not enabled.

Manually later

▶ Tags

[Cancel](#) [Create task](#)

Task 가 Creating 에서 Ready 상태로 바뀌는 것을 확인합니다.  
해당 태스크를 선택한 뒤 Actions > Restart/Resume 을 클릭합니다.

DMS > Database migration tasks

**Database migration tasks (1/1)**

[Find database migration tasks](#)

Identifier	Status	Progress	Type	Source	Target	Actions
cqrs-task1	Ready	0%	Full load, ongoing replication	cqrs-task1-source	cqrs-task1-target	<a href="#">Create premigration assessment</a> <a href="#">Modify</a> <a href="#">Move</a> <a href="#">Restart/Resume</a> <a href="#">Stop</a> <a href="#">Delete</a>

DMS > Database migration tasks

Database migration tasks (1/1)

Identifier	Status	Progress	Type	Source	Target	Replication instance
cqr-task1	Load complete, replication ongoing	100%	Full load, ongoing replication	cqr-task1-source	cqr-task1-target	cqr-dms-ri

## DMS TASK 2 설정 (TARGET DB → KAFKA)

이 단계에서는 Target DB에서 Kafka Topic으로 데이터 동기화하는 DMS Task 2를 생성합니다.  
실습 진행을 간소화하기 위해 DMS Subnet Group과 Replication Instance는 재사용 합니다.

### 1) Task 2 Source Endpoint 생성

Target DB를 Task 2의 Source Endpoint로 생성합니다.

동일한 과정이므로 자세한 설명은 생략합니다.

- **Source endpoint** 선택
- **Select RDS DB instance** 선택
- **RDS Instance:** 드롭다운에서 **ws-target1** 선택
- **Endpoint identifier:** **cqr-task2-source**
- **Access to endpoint database**에서 **Provide access info manually** 선택
- 아래에 DB 정보가 자동으로 입력된 것을 확인할 수 있습니다. (Password의 값은 세션중 공유)

DMS > Endpoints > Create endpoint

## Create endpoint

**Endpoint type** [Info](#)

**Source endpoint**  
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

**Target endpoint**  
A target endpoint allows AWS DMS to write data to a database, or to other data source.

**Select RDS DB instance**

RDS Instance  
Instances available only for current user and region

**Endpoint configuration**

**Endpoint identifier** [Info](#)  
A label for the endpoint to help you identify it.

Descriptive Amazon Resource Name (ARN) - optional  
A friendly name to override the default DMS ARN. You cannot modify it after creation.

Source engine  
The type of database engine this endpoint is connected to. [Learn more](#)

Access to endpoint database

AWS Secrets Manager

Provide access information manually

Server name  
The name of the data server for the data provider.

Port  
The port the database runs on for this endpoint.

User name [Info](#)  Password [Info](#)

Secure Socket Layer (SSL) mode  
The type of Secure Socket Layer enforcement

**Endpoint settings**

## 2) Task 2 Target Endpoint 생성

AWS Console > DMS > Endpoints로 이동합니다. Create endpoint를 클릭합니다.

아래와 같이 입력합니다.

- **Target endpoint** 선택
- **Endpoint identifier:** cqrss-task2-target
- **Target engine:** Kafka
- **Broker:** 이어서 설명
- **Topic (optional)** : diner

Broker 연결 정보 찾기:

DMS > Endpoints > Create endpoint

### Create endpoint

**Endpoint type** Info

Source endpoint  
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data sources such as Amazon S3.

Target endpoint  
A target endpoint allows AWS DMS to write data to a database, or to other data source.

Select RDS DB instance

**Endpoint configuration**

**Endpoint identifier** Info  
The label for the endpoint to be used later.  
**cqrs-task2-target**

**Descriptive Amazon Resource Name (ARN) - optional**  
A friendly name to override the default DMS ARN. You cannot modify it after creation.  
**Friendly-ARN-name**

**Target engine**  
The type of database engine this endpoint is connected to. [Learn more](#)  
**Kafka**

**Broker**  
Apache Kafka can partition messages in a topic among multiple brokers. Specify the host and port of each broker in your Kafka cluster as a comma-separated list of host and port values.  
**tcp://om:9092,b-1-projectkafkacluster.smf7w.c3.kafka.ap-northeast-2.amazonaws.com:9092**

**Topic - optional**  
**diner**

Max length is 255 letters and symbols. You can use period (.), underscore (\_), and minus (-). Topic names with a period (.) or underscore (\_) can collide in internal data structures. Use either one, but not both of these symbols in the topic name.

**Endpoint settings**

**Tags**

**Test endpoint connection (optional)**

**Create endpoint**

별도의 브라우저 탭에서 AWS Console > MSK로 이동합니다.

CDK로 생성했던 클러스터를 클릭합니다.

The screenshot shows the AWS MSK Clusters page. It lists a single cluster named 'ProjectKafkaCluster'. The cluster status is 'Active', authentication is 'Unauthenticated', and it has 2 brokers. A red box highlights the 'ProjectKafkaCluster' link under the Cluster name column.

우측 상단의 Client Information 버튼을 클릭합니다.

The screenshot shows the AWS MSK Cluster details page for 'ProjectKafkaCluster'. It displays cluster summary information including Apache Kafka version (2.8.1), ARN (arn:aws:kafka:ap-northeast-2:24540904196:cluster/ProjectKafkaCluster), and creation time (October 21, 2022 at 11:39 PM GMT+9). A red box highlights the 'View client information' button.

Bootstrap servers > Plaintext 타입의 Private endpoint를 복사합니다 (그림).

The screenshot shows the AWS MSK Client Information page for 'ProjectKafkaCluster'. It lists bootstrap servers for both 'Plaintext' and 'TLS'. Under 'Plaintext', a red box highlights the 'Private endpoint' field which contains the URL 'tcp://b-2-projectkafkacluster.smf7w.c3.kafka.ap-northeast-2.amazonaws.com:9092'. Below this, another 'Private endpoint' entry is shown with the URL 'tcp://b-3-projectkafkacluster.smf7w.c3.kafka.ap-northeast-2.amazonaws.com:9094'.

엔드포인트 생성 후 인스턴스로부터 연결할 수 있는지 테스트합니다.

DMS > Endpoints > **cqrs-task2-target** > Test endpoint connection

### Test endpoint connection

**Replication instance**  
A replication instance performs the database migration  
**cqrs-dms-ri**

**Run test**

Endpoint identifier	Replication instance	Status	Message
<b>cqrs-task2-target</b>	<b>cqrs-dms-ri</b>	<b>successful</b>	

**Back**

### 3) DMS Task 2 생성

AWS Console > DMS > Database migration tasks로 이동합니다. 우측의 Create task 버튼을 클릭합니다.

The screenshot shows the AWS DMS console with the 'Database migration tasks' page. On the left, there's a sidebar with options like 'Dashboard', 'Database migration tasks' (which is selected and highlighted with a red box), 'Replication instances', 'Endpoints', 'Certificates', 'Subnet groups', 'Events', and 'Event subscriptions'. Below that is a 'New feature announcements' section. The main area shows a table titled 'Database migration tasks (1)'. The table has columns for Identifier, Status, Progress, Type, Source, Target, and Replication instance. One row is listed: 'cqr-task1' with status 'Load complete, replication ongoing', progress '100%', type 'Full load, ongoing replication', source 'cqr-task1-source', target 'cqr-task1-target', and replication instance 'cqr-dms-ri'. At the top right of the table area, there's a 'Create task' button, which is also highlighted with a red box.

Task configuration 에 아래 내용을 입력합니다.

- **Task-identifier:** cqr-task2
- **Replication instance:** 위에서 생성한 인스턴스 선택 (cqr-dms-ri)
- **Source database endpoint:** cqr-task2-source
- **Target database endpoint:** cqr-task2-target
- **Migration type:** Migrate existing data and replicate ongoing changes

The screenshot shows the 'Create database migration task' dialog. It has a 'Task configuration' section. The 'Task identifier' field contains 'cqr-task2' (highlighted with a red box). The 'Replication instance' dropdown contains 'cqr-dms-ri - vpc-0cec5bfc0fbfa07cf' (highlighted with a red box). The 'Source database endpoint' dropdown contains 'cqr-task2-source' (highlighted with a red box). The 'Target database endpoint' dropdown contains 'cqr-task2-target' (highlighted with a red box). The 'Migration type' dropdown contains 'Migrate existing data and replicate ongoing changes' (highlighted with a red box). There's also a note about upgrading to version 3.4.7 and higher, which is enclosed in a blue box and has a 'View endpoints' button.

Task settings 에서 아래의 그림처럼 설정합니다.

- **Target table preparation mode:** Drop tables on target
- 나머지는 디폴트 값 그대로 사용

**Task settings**

Editing mode | [Info](#)

**Wizard**  
You can enter only a subset of the available task settings.

**JSON editor**  
You can enter all available task settings directly in JSON format.

Target table preparation mode | [Info](#)

Do nothing

**Drop tables on target**

Truncate

Include LOB columns in replication | [Info](#)

Don't include LOB columns

Full LOB mode

Limited LOB mode

Maximum LOB size (KB) | [Info](#)

32

Turn on validation  
Choose this setting if you want AWS DMS to compare the data at the source and the target immediately after it performs a full data load. Validation ensures that your data was migrated accurately, but it requires additional time to complete.

Task logs | [Info](#)

Turn on CloudWatch logs  
DMS task logging uses Amazon CloudWatch to log information during the migration process. You can change the component activities logged and the amount of information logged for each one.

**▶ Advanced task settings**

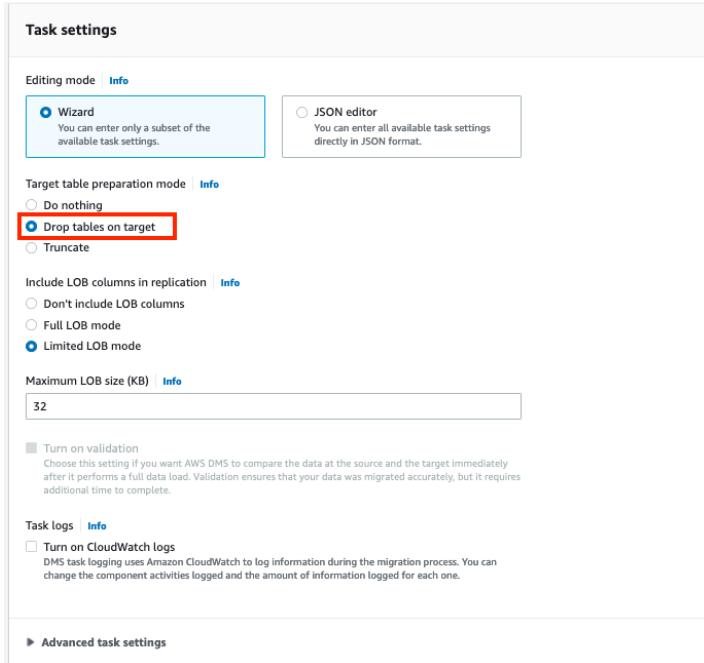


Table Mappings에서 Wizard를 선택하고 Add new selection rule 버튼을 클릭합니다.  
그리고 아래와 같이 입력합니다.

- **Schema:** Enter a schema
- **Source name:** `diner`
- **Source table name:** %
- **Action:** Include

**Table mappings**

Editing mode: [Info](#)

**Wizard**  
You can enter only a subset of the available table mappings.

**JSON editor**  
You can enter all available table mappings directly in JSON format.

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ Selection rules

Choose the schema and/or tables you want to include with, or exclude from, your migration task. [Info](#)

[Add new selection rule](#)

▼ where schema name is like 'diner' and Source table name is like '%', include

Schema  
[Enter a schema](#)

Source name  
Use the % character as a wildcard  
[diner](#)

Source table name  
Use the % character as a wildcard  
[%](#)

Action  
Choose "Include" to migrate your selected objects, or "Exclude" to ignore them during the migration.  
[Include](#)

Source filters [Info](#)      Add column filter

▶ Transformation rules

Migration task startup configuration 항목의 Start migration task 를 “Manually later” 로 지정합니다 (그림).  
하단 Create task 버튼을 클릭합니다.

**Migration task startup configuration**

Start migration task

Automatically on create  
Available only if the premigration assessment is not enabled.

**Manually later**

▶ Tags

[Cancel](#)      **Create task**

Task 가 Creating 에서 Ready 상태로 바뀌는 것을 확인합니다.  
해당 태스크를 선택한 뒤 Actions > Restart/Resume 을 클릭합니다.

DMS > Database migration tasks

Database migration tasks (1/2)							Actions	Quick view and compare	Create task
Identifier	Status	Progress	Type	Source	Target	Started			
cqrs-task1	Load complete, replication ongoing	100%	Full load, ongoing replication	cqrs-task1-source	cqrs-task1-target	October 26, 2022 at 01:4	<a href="#">Create premigration assessment</a> <a href="#">Modify</a> <a href="#">Move</a> <b>Restart/Resume</b> <a href="#">Stop</a> <a href="#">Delete</a>		
<input checked="" type="checkbox"/> cqrs-task2	Ready	0%	Full load, ongoing replication	cqrs-task2-source	cqrs-task2-target	-			

DMS > Database migration tasks

Database migration tasks (1/2)							Actions	Quick view and compare	Create task	
Identifier	Status	Progress	Type	Source	Target	Replication instance	Started			
cqrs-task1	Load complete, replication ongoing	100%	Full load, ongoing replication	cqrs-task1-source	cqrs-task1-target	cqrs-dms-ri	October 26, 2022 at 01:4			
<input checked="" type="checkbox"/> cqrs-task2	Load complete, replication ongoing	100%	Full load, ongoing replication	cqrs-task2-source	cqrs-task2-target	cqrs-dms-ri	October 26, 2022 at 02:4			

### 3. Migration 된 데이터 확인

#### TASK 1 마이그레이션 확인

Cloud9 에 접속합니다.

```
sudo yum update -y
sudo yum install -y mysql
```

(터미널을 하나 더 띄우는 것을 권장합니다.) 다음을 실행하여 Source DB / TargetDB 에 접속합니다.  
prompt 에 패스워드를 입력합니다 (세션 중 공유).

```
mysql -h sourcedbstack-xxxxx.cluster-xxxxx.ap-northeast-2.rds.amazonaws.com \
-P 3306 -u admin -p diner
```

아래의 SQL 명령어를 수행하여 Source DB 로부터 Target DB 에 데이터 마이그레이션이 잘 되었는지 확인합니다.

```
show tables;
select * from orders;
select * from delivery;
select * from kitchen;
```

```

Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 1026
Server version: 8.0.23 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [diner]> show tables;
+-----+
| Tables_in_diner |
+-----+
| delivery      |
| kitchen       |
| orders        |
+-----+
3 rows in set (0.00 sec)

MySQL [diner]> select * from orders;
+-----+
| order_id | status | kitchen_id | delivery_id | created_at           | updated_at          |
+-----+
| 1        | CLOSED | 1          | 1            | 2022-10-25 12:25:05 | 2022-10-25 12:26:09 |
| 2        | CLOSED | 2          | 2            | 2022-10-25 12:25:39 | 2022-10-25 12:30:40 |
| 3        | CLOSED | 3          | 3            | 2022-10-25 12:31:08 | 2022-10-25 12:32:18 |
+-----+
3 rows in set (0.00 sec)

MySQL [diner]> select * from delivery;
+-----+
| delivery_id | status | created_at           | updated_at          |
+-----+
| 1           | DELIVERED | 2022-10-25 12:25:19 | 2022-10-25 12:26:09 |
| 2           | DELIVERED | 2022-10-25 12:25:46 | 2022-10-25 12:30:40 |
| 3           | DELIVERED | 2022-10-25 12:31:53 | 2022-10-25 12:32:18 |
+-----+
3 rows in set (0.00 sec)

MySQL [diner]> select * from kitchen;
+-----+
| kitchen_id | status | created_at           | updated_at          |
+-----+
| 1           | DONE   | 2022-10-25 12:25:05 | 2022-10-25 12:25:42 |
| 2           | DONE   | 2022-10-25 12:25:39 | 2022-10-25 12:26:44 |
| 3           | DONE   | 2022-10-25 12:28:17 | 2022-10-25 12:31:28 |
+-----+
3 rows in set (0.00 sec)

```

## TASK 2 마이그레이션 확인

Cloud9에서 `./cqrss-dms-workshop/cdk/cqrss-workshop.ts`에 아래의 정보를 추가합니다.

- `KAFKA_BOOTSTRAP_SERVERS`
- `KAFKA_ZOOKEEPER_ADDRESSES`

```

1 #!/usr/bin/env node
2 import 'source-map-support/register';
3 import * as cdk from 'aws-cdk-lib';
4 import * as ec2 from 'aws-cdk-lib/aws-ec2';
5 import * as rds from 'aws-cdk-lib/aws-rds';
6 import { Aurora as AuroraStack } from '../lib/aurora';
7 import { KafkaStack } from '../lib/kafka';
8 import { DmsStack } from '../lib/dms';
9 import { FargateStack } from '../lib/fargate';
10
11 const VPC_ID = 'vpc-0cec5bfc0fbfa07cf';
12 const VPC_CIDR = '10.0.0.0/16';
13 const PRIVATE_SUBNET_IDS = ['subnet-09e3b252f6d398efe','sub
14 const SOURCE_DB_SNAPSHOT_ARN = 'arn:aws:rds:ap-northeast-2:
15 const KAFKA_BOOTSTRAP_SERVERS = 'b-2.projectkafkaclust.smtf
16 const KAFKA_ZOOKEEPER_ADDRESSES = 'z-1.projectkafkaclust.sm
17

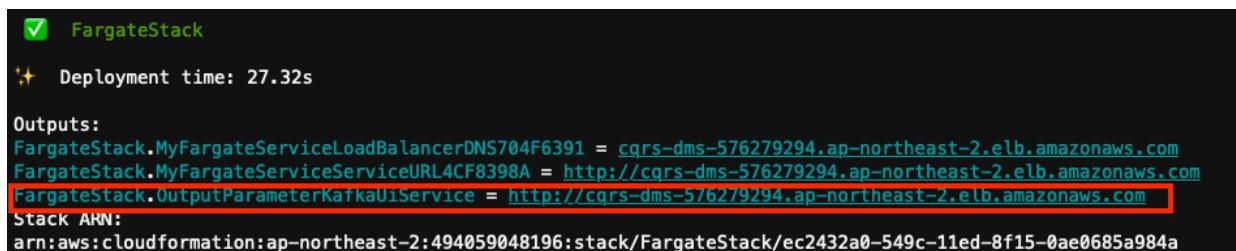
```

AWS Console > MSK > Cluster > Client Information

Cloud9에서 FargateStack을 배포합니다.

```
# cd cqrss-dms-workshop/cdk
cdk deploy FargateStack
```

Output 중 OutputParameterKafkaUiService의 주소를 복사하고 브라우저에서 엽니다.



오픈소스인 Kafka-ui 화면으로 진입합니다.

Topics > diner > Messages로 이동하여 DMS Full Load 이벤트가 어떻게 쌓이는지 파악할 수 있습니다.

Offset	Partition	Timestamp	Key	Content
0	0	2022-10-26T17:43:14	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.awsdms_apply_exceptions	{ "metadata": { "timestamp": "2022-10-25T17:43:14.306460Z", "record-type": "control", "ope
1	0	2022-10-26T17:43:15	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.diner.delivery	{ "metadata": { "timestamp": "2022-10-25T17:43:15.173889Z", "record-type": "control", "ope
2	0	2022-10-26T17:43:15	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.diner.delivery	{ "metadata": { "timestamp": "2022-10-25T17:43:15.180533Z", "record-type": "control", "ope
3	0	2022-10-26T17:43:15	1	{ "data": { "delivery_id": 1, "status": "DELIVERED", "created_at": "2022-10-25T12:25:10Z", "up
4	0	2022-10-26T17:43:15	2	{ "data": { "delivery_id": 2, "status": "DELIVERED", "created_at": "2022-10-25T12:25:46Z", "u
5	0	2022-10-26T17:43:15	3	{ "data": { "delivery_id": 3, "status": "DELIVERED", "created_at": "2022-10-25T12:31:53Z", "u
6	0	2022-10-26T17:43:15	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.diner.kitchen	{ "metadata": { "timestamp": "2022-10-25T17:43:15.268621Z", "record-type": "control", "ope
7	0	2022-10-26T17:43:15	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.diner.kitchen	{ "metadata": { "timestamp": "2022-10-25T17:43:15.272921Z", "record-type": "control", "ope
8	0	2022-10-26T17:43:15	1	{ "data": { "kitchen_id": 1, "status": "DONE", "created_at": "2022-10-25T12:25:05Z", "update
9	0	2022-10-26T17:43:15	2	{ "data": { "kitchen_id": 2, "status": "DONE", "created_at": "2022-10-25T12:25:39Z", "update
10	0	2022-10-26T17:43:15	3	{ "data": { "kitchen_id": 3, "status": "DONE", "created_at": "2022-10-25T12:28:17Z", "update
11	0	2022-10-26T17:43:15	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.diner.orders	{ "metadata": { "timestamp": "2022-10-25T17:43:15.354425Z", "record-type": "control", "ope
12	0	2022-10-26T17:43:15	KNZTV4l7XG36LL3PRXNJi67YTEFXLYKDUCASCKQ.diner.orders	{ "metadata": { "timestamp": "2022-10-25T17:43:15.361134Z", "record-type": "control", "ope
13	0	2022-10-26T17:43:15	1	{ "data": { "order_id": 1, "status": "CLOSED", "kitchen_id": 1, "delivery_id": 1, "created_at": "20
14	0	2022-10-26T17:43:15	2	{ "data": { "order_id": 2, "status": "CLOSED", "kitchen_id": 2, "delivery_id": 2, "created_at": "2
15	0	2022-10-26T17:43:15	3	{ "data": { "order_id": 3, "status": "CLOSED", "kitchen_id": 3, "delivery_id": 3, "created_at": "2

## 4. CDC 테스트를 위한 위한 Producer App 실행 (dinerbot)

Cloud9에서 './cqrss-dms-workshop/dinerbot/'으로 이동합니다.

1초 간격으로 Source DB에 Transaction을 만드는 스프링 어플리케이션입니다.

src/main/resources/dinerbotapplication-deploy.properties에서 다음의 값을 Source DB 값으로 변경합니다.

```
spring.datasource.url=${DB_CONN_STR}
spring.datasource.username=${DB_USRNAME}
spring.datasource.password=${DB_PASSWORD}
```

src/main/resources/dinerbotapplication.properties에서 다음의 값으로 변경합니다.

```
# spring.profiles.active=local
spring.profiles.active=deploy
```

어플리케이션을 실행합니다.

```
# cd cqrss-dms-workshop/dinerbot/
./gradlew clean bootRun
```

Target DB의 조회 테이블 확인합니다.

Kafka Diner 토픽의 메세지를 확인합니다.

## 5. 조회모델 생성을 위한 Consumer App 개발 및 실행

Kafka Diner 토픽의 메세지를 받아 리드 모델 테이블에 저장하는 스켈레톤 코드가 첨부되어 있습니다 (cqrss-dms-workshop/consumer/).

동작하는 어플리케이션은 아니지만 조회 모델을 구성하는 로직을 com.cqrss.consumer.message.DinerMessageHandler에 커멘트로 작성하였습니다.

```
@KafkaListener(topics = "{$kafka.topic.diner}")  
public void onMessage(DmsMessage message) {  
    log.info("metadata: {}, payload: {}", message.getMetadata(), message.getPayload());  
    // 메세지 타입 분류하여 적절한 핸들러에 위임 (Full Load vs CDC)  
    // Full Load 타입의 경우: CREATE-TABLE 과 DROP-TABLE 서브 타입으로 구분  
    // --> 예: Full Load 메세지가 들어온 경우 시스템을 종료한다 (예외)  
    // CDC 타입의 경우: INSERT, UPDATE, DELETE 의 서브 타입으로 구분  
    // --> 예: 1. 계약 아이디를 꺼낸다.  
    //           2. 조회 모델을 구성하기 위해 Target DB의 원본 테이블들을 Join 하여 조회한다.  
    //           3. 이를 Target DB의 조회 모델에 저장한다.  
}
```

## 6. View vs Read Table 조회 비교

이 워크샵의 경우 사용하는 데이터가 많아서 체감이 되기 어려울 수 있지만, Read Table 을 별도로 관리하는 경우 View 를 사용하는 것 보다 조회 성능이 크게 향상될 수 있습니다.