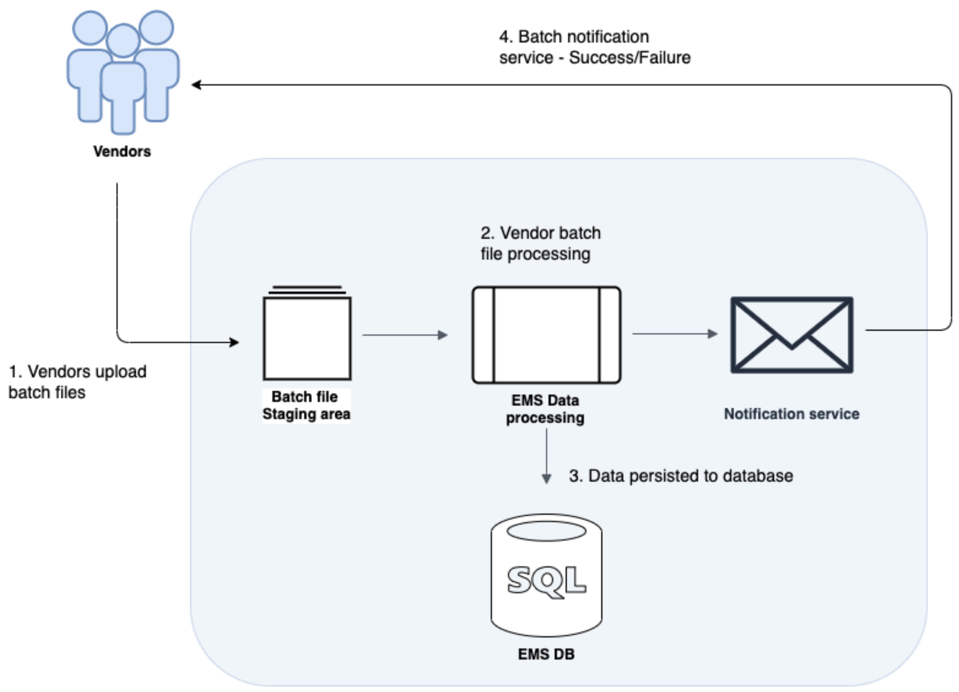
Lab 1: Working with Amazon Aurora databases

Objectives:

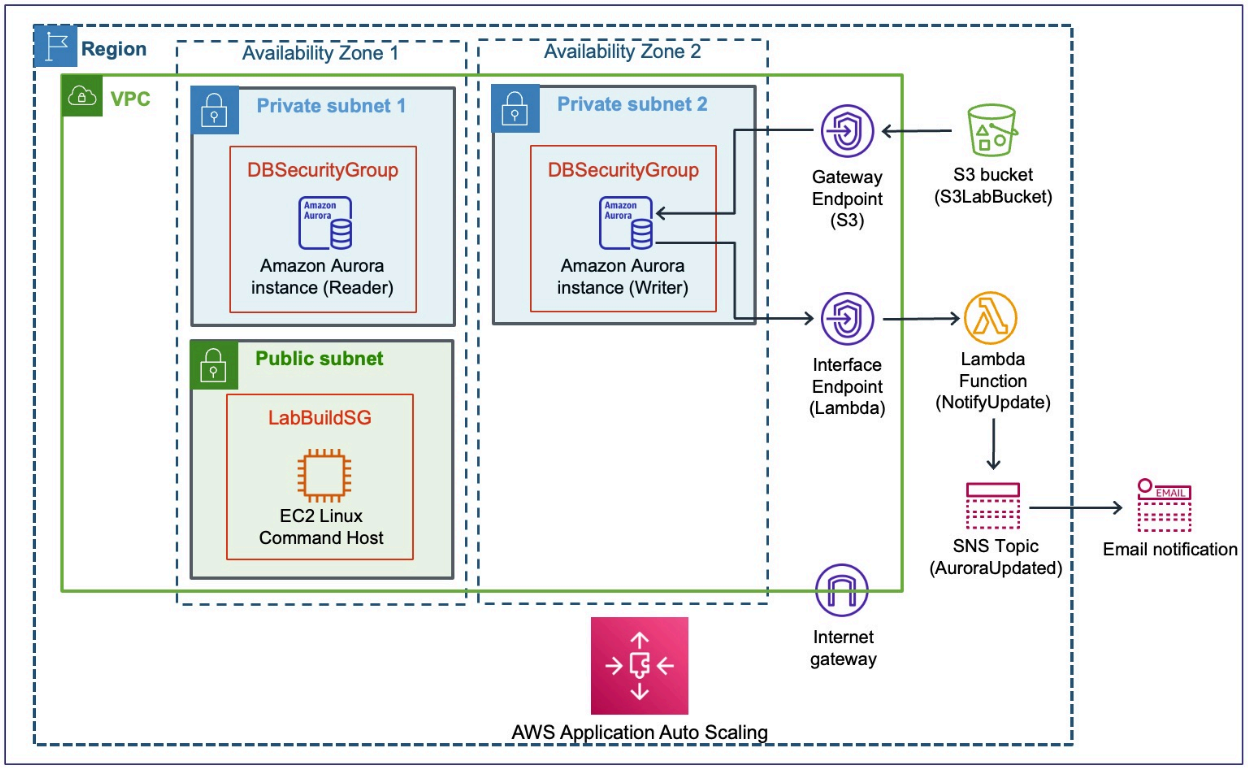
1. Design an Aurora cluster to meet application requirements.
2. Architect a solution on Aurora for high availability and scalability.
3. Integrate Aurora with other AWS services - Amazon S3.

Data flow diagram


4. Batch notification service - Success/Failure
Vendors
2. Vendor batch file processing
1. Vendors upload batch files
Batch file Staging area
EMS Data processing
Notification service
3. Data persisted to database
SQL
EMS DB

DataFlow: (as example of event mgmt company, which has event vendors:

1. Vendor uploads event data files to Amazon S3.
2. The system automatically picks up these files and loads the data into the Aurora database.
3. Vendors are notified about the success or failure of the data loading process.

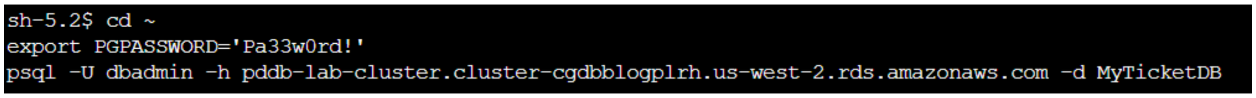


Connecting to the Database via CLI

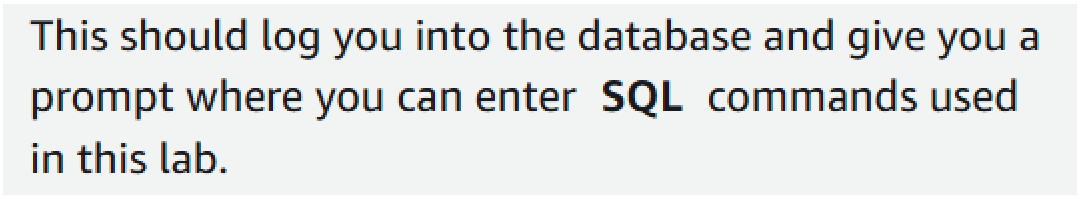
Steps -

1. Open CLI using given URL to connect to PostgreSQL Database
2. Run this

```
cd ~
export PGPASSWORD='DBUserPasswd'
psql -U DBUserName -h DBClusterEndpoint -d MyTicketDB
```



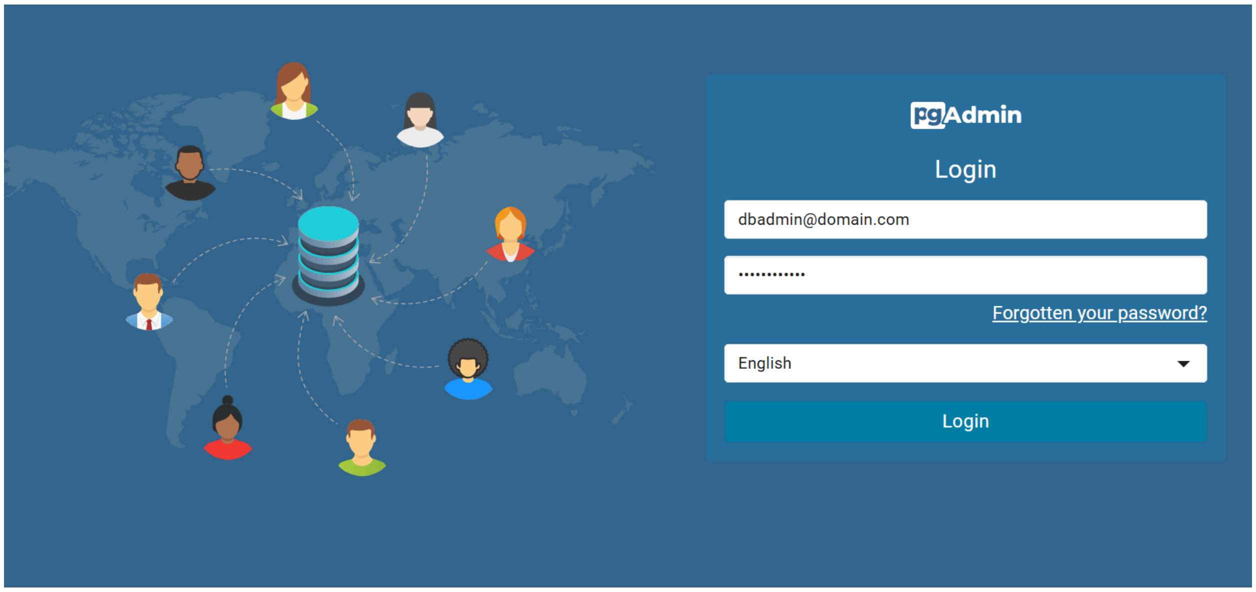
```
sh-5.2$ cd ~
export PGPASSWORD='Pa33w0rd!'
psql -U dbadmin -h pddb-lab-cluster.cluster-cgdbblogplrh.us-west-2.rds.amazonaws.com -d MyTicketDB
```



This should log you into the database and give you a prompt where you can enter **SQL** commands used in this lab.

Connecting to DB via UI

Steps-

1. Use Url to get to login page, and login
2. 
pgAdmin

Login

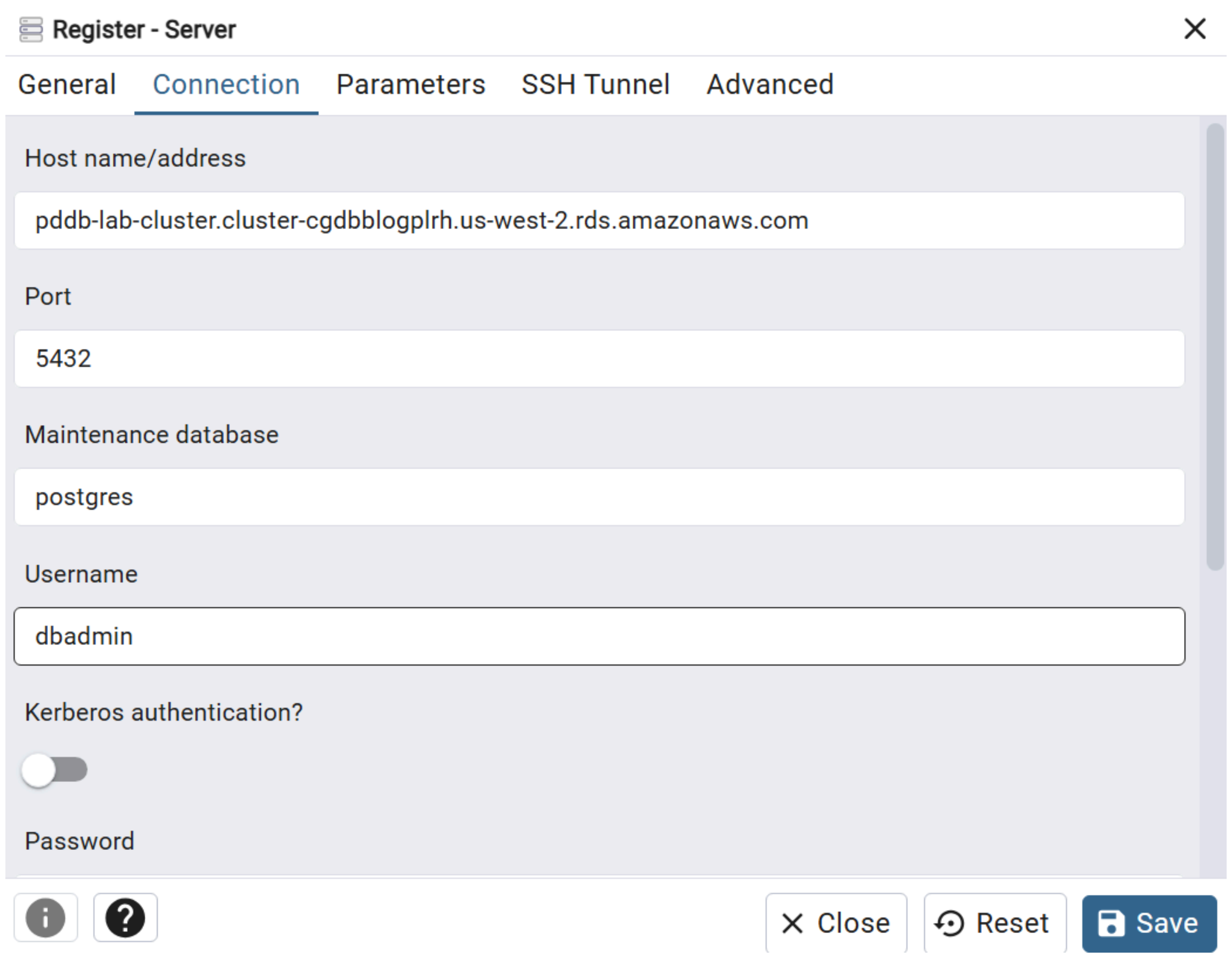
dbadmin@domain.com

•••••••••••••

Forgotten your password?

English

Login
3. Create a new server as follows


Register - Server

General | Connection | Parameters | SSH Tunnel | Advanced

Host name/address: pddb-lab-cluster.cluster-cgdbblogplrh.us-west-2.rds.amazonaws.com

Port: 5432

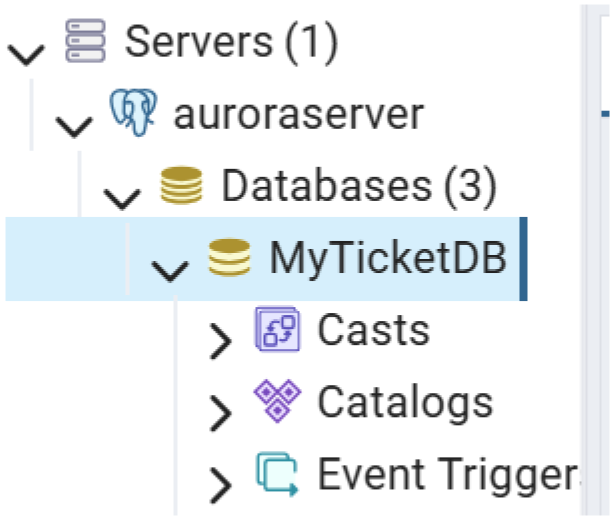
Maintenance database: postgres

Username: dbadmin

Kerberos authentication? Off

Password

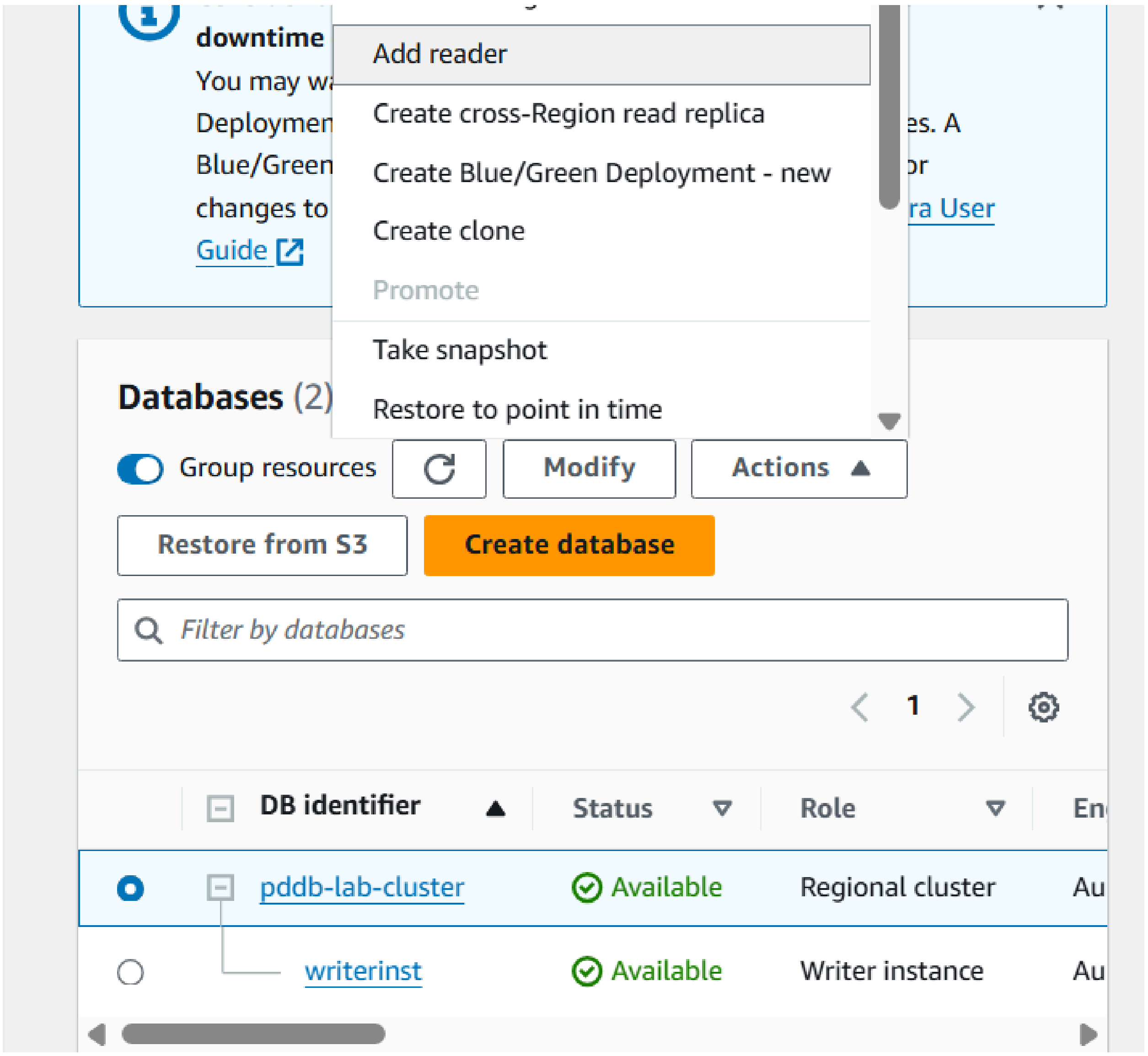
4.
5. This referenced this with the code we ran in CLI in last task.
6. Open this DB



7.
8. This is where we’ll run our SQL commands

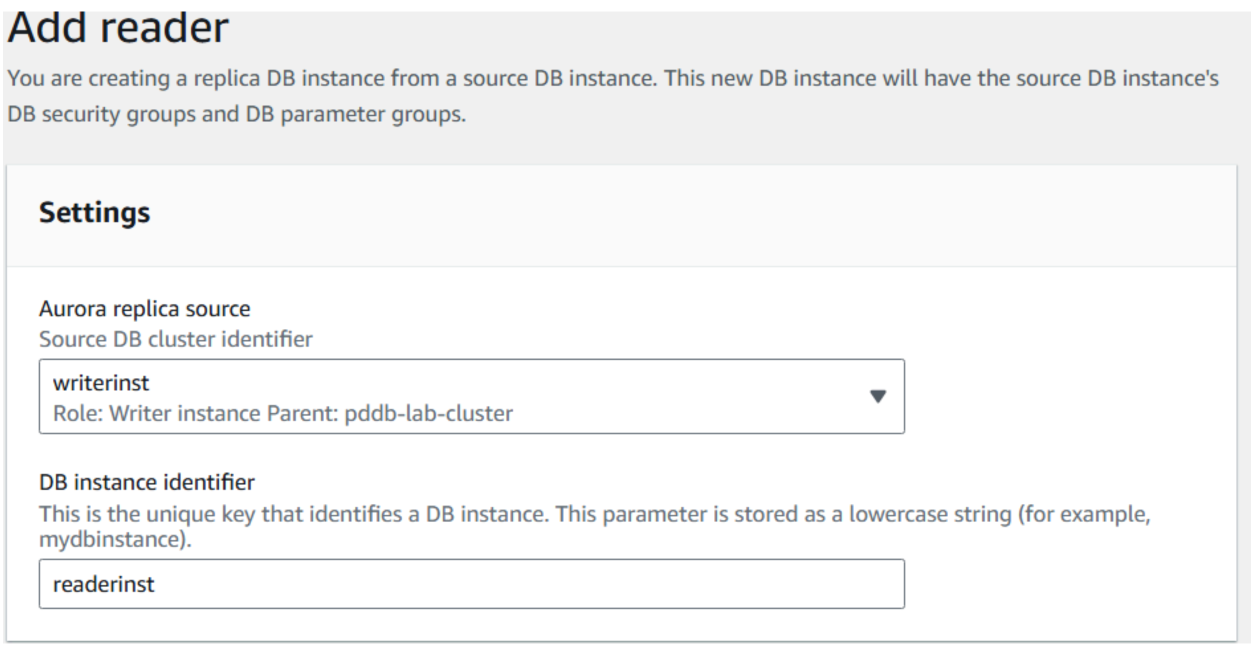
Task 1: Configure Amazon Aurora for high availability

1.1 open console, go to RDS
1.2 Select the created DB



DB identifier	Status	Role	En
pddb-lab-cluster	Available	Regional cluster	Au
writerinst	Available	Writer instance	Au

1.3 select add reader (a read-only instance of a database) option, note we already have a writer
1.4 enter the name for reader, and go ahead


Add reader

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

Settings

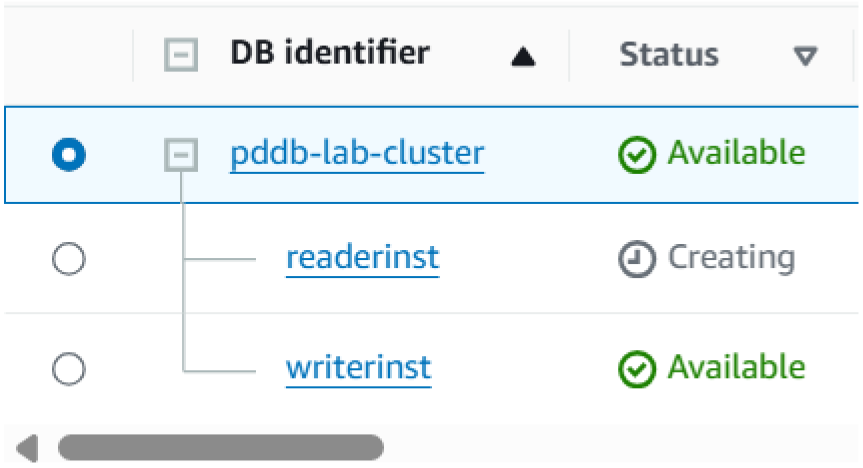
Aurora replica source
Source DB cluster identifier

writerinst
Role: Writer instance Parent: pddb-lab-cluster

DB instance identifier
This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (for example, mydbinstance).

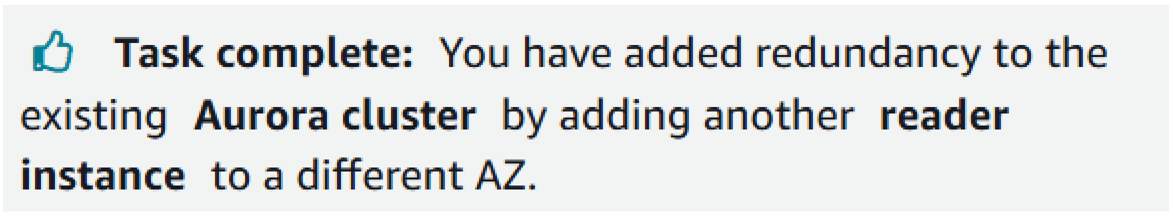
readerinst

1.5 creating



	DB identifier	Status
●	pddb-lab-cluster	Available
○	readerinst	Creating
○	writerinst	Available

Goal -



Task complete: You have added redundancy to the existing **Aurora cluster** by adding another **reader instance** to a different AZ.

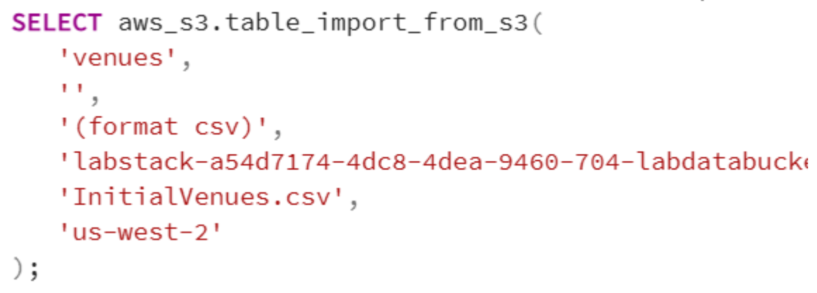
Task 2: Configure DB to ingest a batch file from Amazon S3

2.1 Connect to DB (done in pre-task steps), eihter way

2.2 Install S3 extension in sql

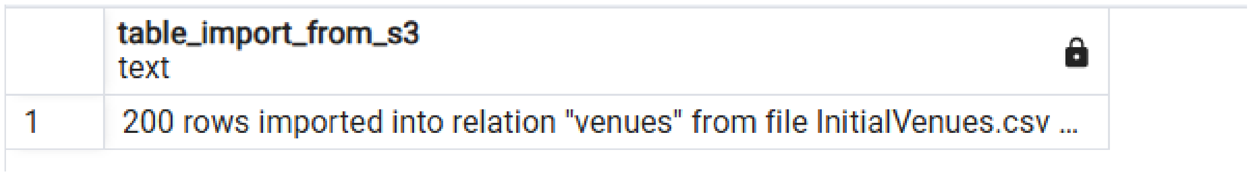
CREATE EXTENSION IF NOT EXISTS aws_s3 CASCADE;

2.3 Run following command to ingest data from S3 to DB



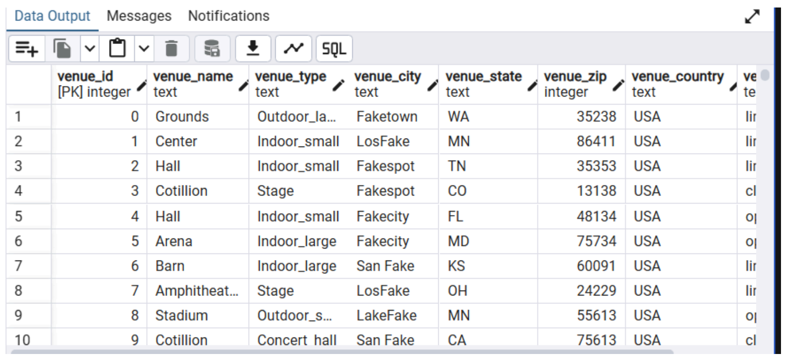
```
SELECT aws_s3.table_import_from_s3(
    'venues',
    '',
    '(format csv)',
    'labstack-a54d7174-4dc8-4dea-9460-704-labdatabuck
    'InitialVenues.csv',
    'us-west-2'
);
```

Labstack-a5… is the S3 bucket location
o/p-



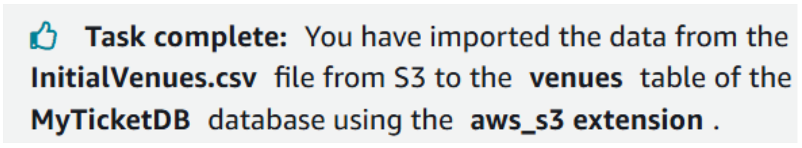
	table_import_from_s3 text
1	200 rows imported into relation "venues" from file InitialVenues.csv ...

2.4 view table



Data Output Messages Notifications

	venue_id [PK] integer	venue_name text	venue_type text	venue_city text	venue_state text	venue_zip integer	venue_country text	ve te
1	0	Grounds	Outdoor_la...	Faketown	WA	35238	USA	lir
2	1	Center	Indoor_small	LosFake	MN	86411	USA	lir
3	2	Hall	Indoor_small	Fakespot	TN	35353	USA	lir
4	3	Cotillion	Stage	Fakespot	CO	13138	USA	cl
5	4	Hall	Indoor_small	Fakecity	FL	48134	USA	o
6	5	Arena	Indoor_large	Fakecity	MD	75734	USA	o
7	6	Barn	Indoor_large	San Fake	KS	60091	USA	lir
8	7	Amphitheat...	Stage	LosFake	OH	24229	USA	lir
9	8	Stadium	Outdoor_s...	LakeFake	MN	55613	USA	o
10	9	Cotillion	Concert_hall	San Fake	CA	75613	USA	cl



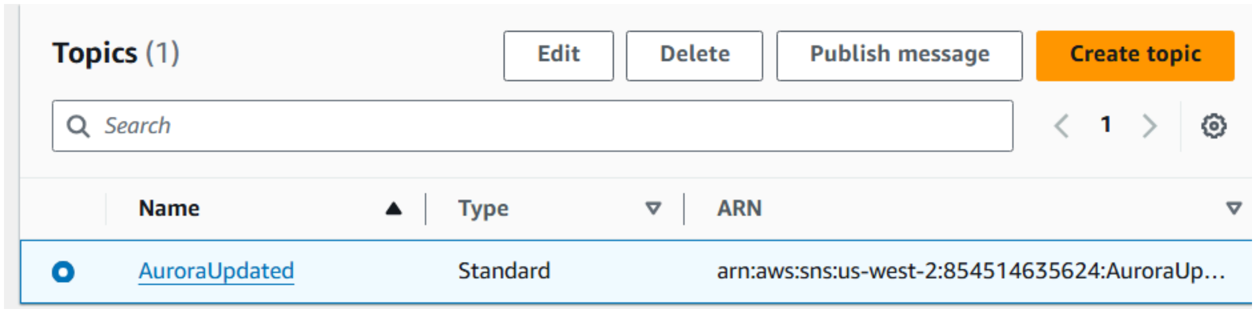
Task complete: You have imported the data from the **InitialVenues.csv** file from S3 to the **venues** table of the **MyTicketDB** database using the **aws_s3 extension** .

Task 3: Configure Aurora to notify a vendor of the batch data processing status

Subscribe to an SNS topic and
Configure the database to invoke a Lambda function for notifications.

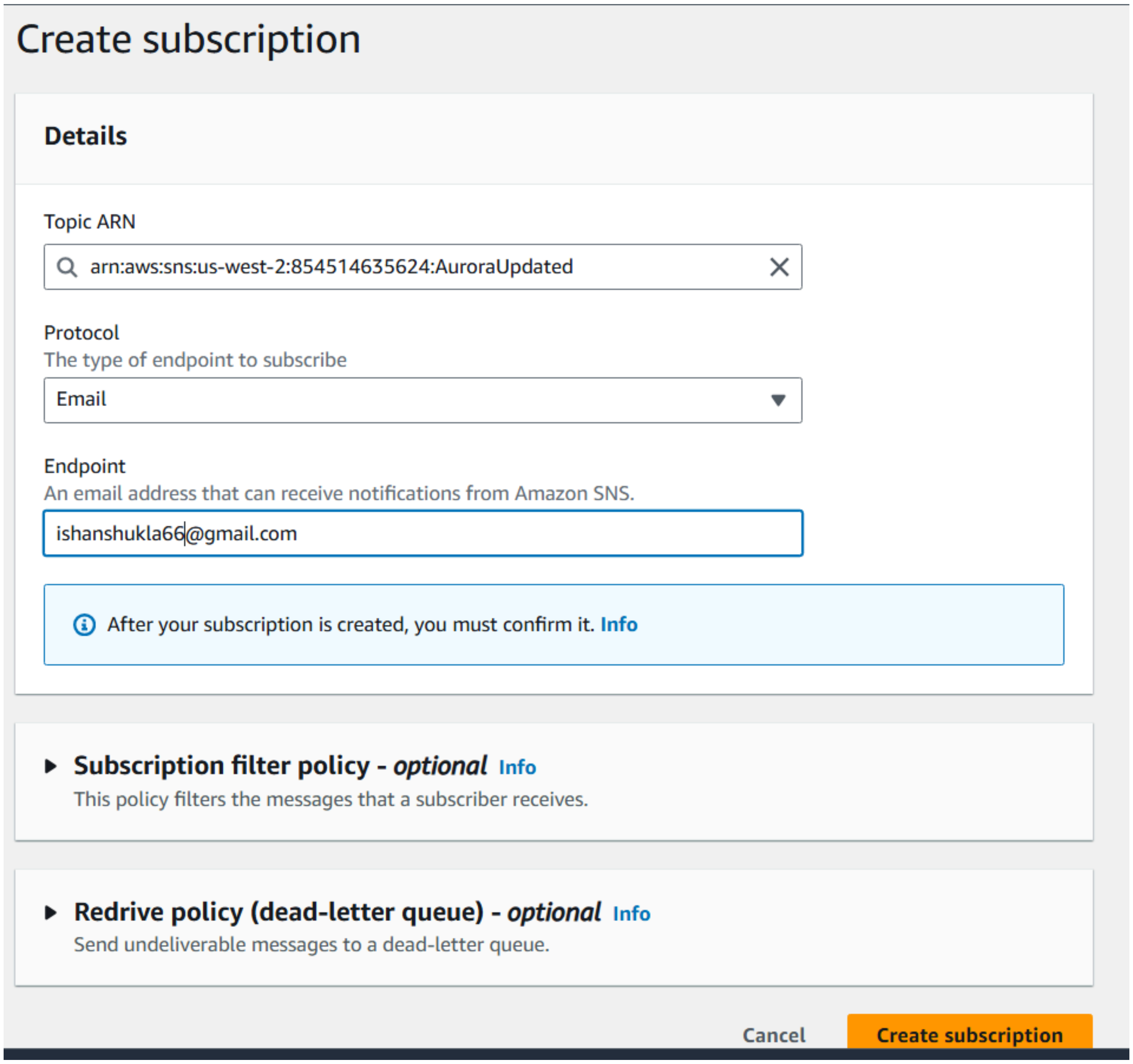
TASK 3.1: **SUBSCRIBE TO AN SNS TOPIC**

3.11 open console and open SNS(Simple Notification Service)
3.12 open this topic



Name	Type	ARN
AuroraUpdated	Standard	arn:aws:sns:us-west-2:854514635624:AuroraUp...

3.13 Create a subscription as follows


Create subscription

Details

Topic ARN
arn:aws:sns:us-west-2:854514635624:AuroraUpdated

Protocol
The type of endpoint to subscribe
Email

Endpoint
An email address that can receive notifications from Amazon SNS.
ishanshukla66@gmail.com

After your subscription is created, you must confirm it. Info

Subscription filter policy - *optional* Info
This policy filters the messages that a subscriber receives.

Redrive policy (dead-letter queue) - *optional* Info
Send undeliverable messages to a dead-letter queue.

3.14 open email and click confirmation link

TASK 3.2: CONFIGURE DATABASE WITH LAMBDA FUNCTION

3.21 go back to sql
3.22 Install the needed Lambda CASCADE extension with the following SQL command.

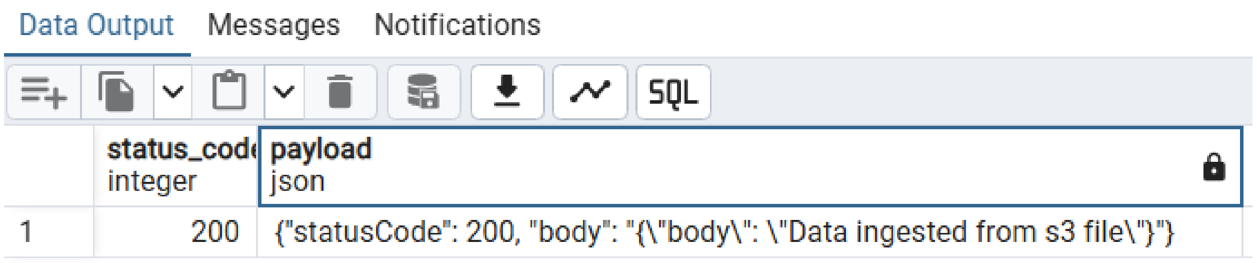
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;

3.23 run the command (why? -it's answer not given in lab)
SELECT * from
aws_lambda.invoke(aws_commons.create_lambda_function_arn('LAMBDA ARN', 'AWSREGION'), '{"body": "Data ingested from s3 file"}'::json);

Here LAMBDAARN value it to be replaced with :
arn:aws:lambda:us-west-2:854514635624:function:NotifyUpdate

Note the "notifyupdate" in above line

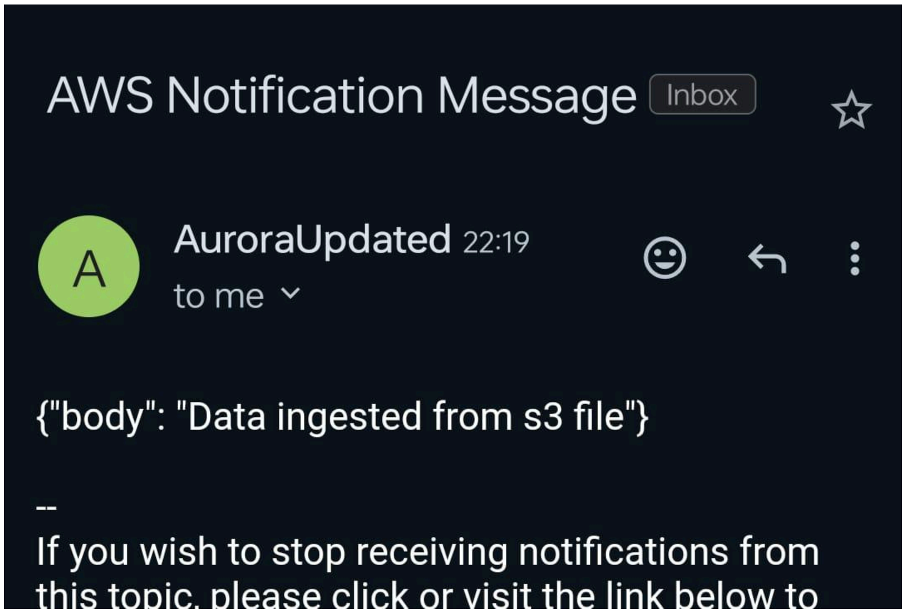
o/p-



	status_code integer	payload json
1	200	{"statusCode": 200, "body": "{\"body\": \"Data ingested from s3 file\"}"}

This will notify vendors with the shown message

3.24 email recd.



AWS Notification Message Inbox

AuroraUpdated 22:19
to me

{"body": "Data ingested from s3 file"}

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to