Database Migration Service

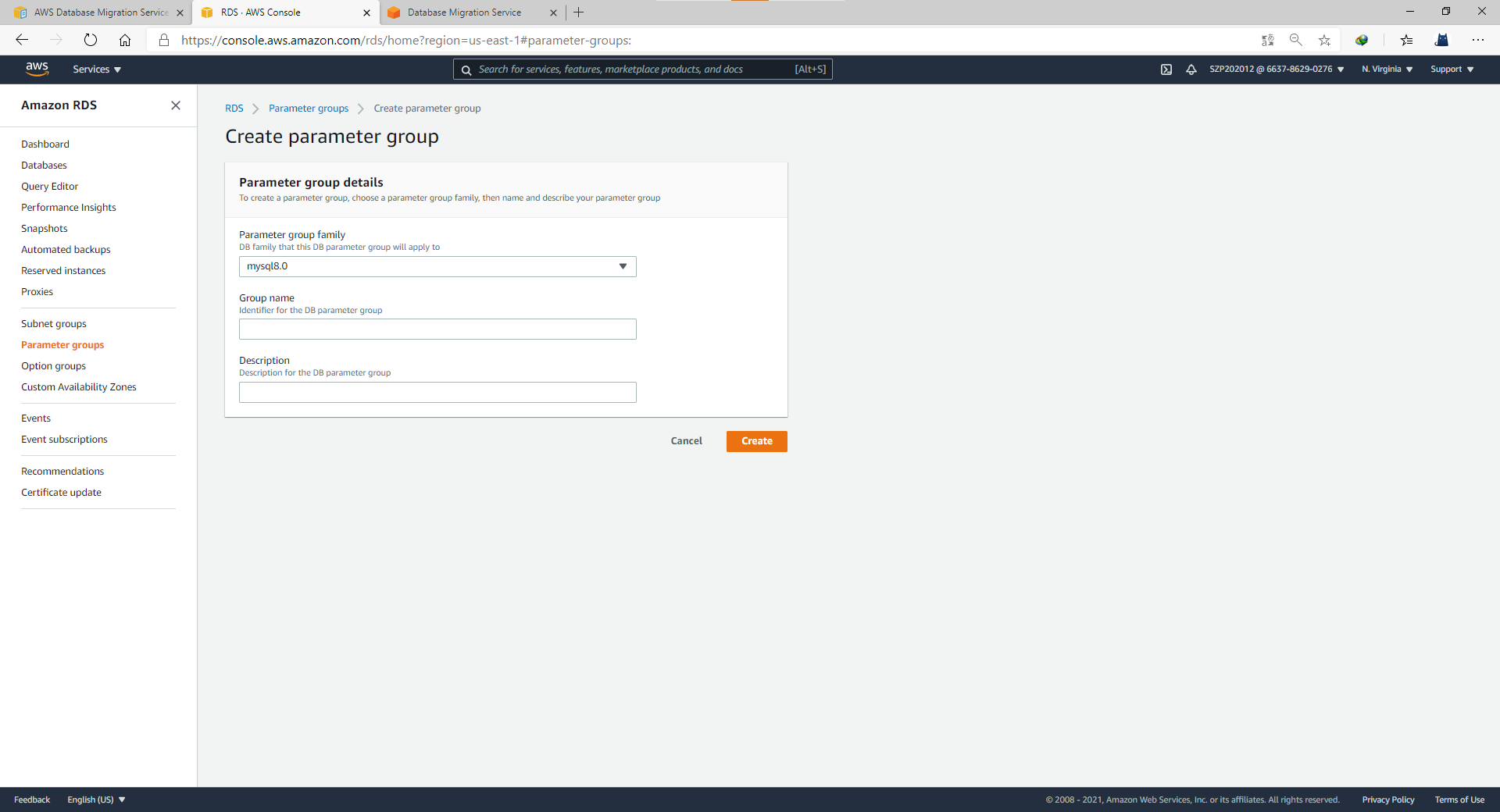
什么是DMS

AWS Database Migration Service 是一种 Web 服务，可用于将数据从本地数据库、Amazon Relational Database Service (Amazon RDS) 数据库实例上的数据库或 Amazon Elastic Compute Cloud (Amazon EC2) 实例上的数据库迁移到 AWS 服务上的数据库。这些服务可以包括 Amazon RDS 上的数据库或 Amazon EC2 实例上的数据库。您还可以将数据库从 AWS 服务迁移到本地数据库。您可以在异构和同构数据库引擎之间迁移数据。

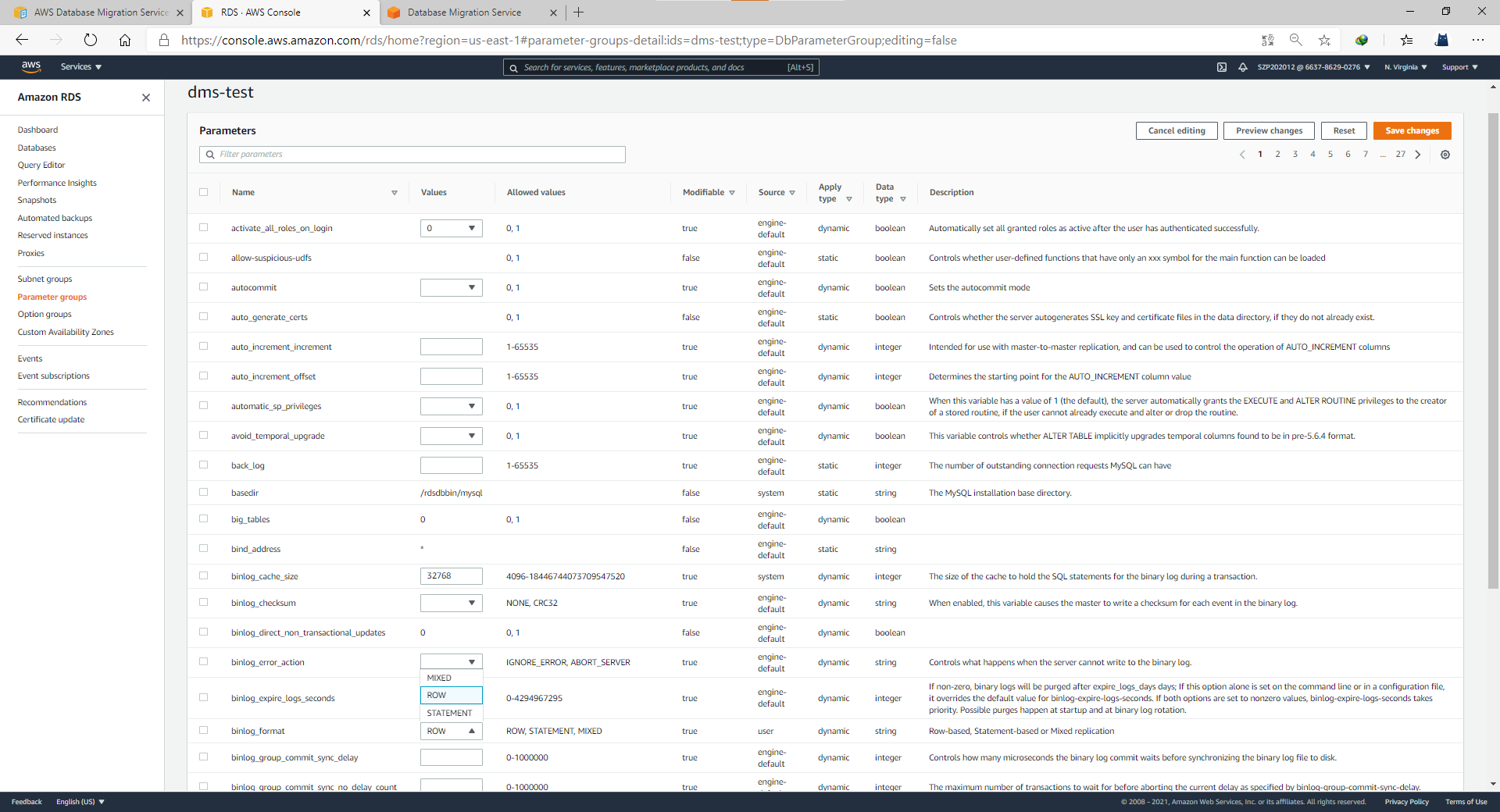
实验：迁移一个数据库的数据到一个新数据库

环境预设的是一个Mysql数据库在持续写入数据，我们需要将这台数据库的数据复制到另一个数据库中去

这次需要配置参数组



创建完成后编辑这个参数组，找到binlog\_format，选择ROW



1. 我们先来创建一个RDS数据库和一台会向他一直写入数据的EC2

RDS配置

子网：位于私网中

引擎：Mysql

参数组：刚刚创建的参数组

初始数据库名字：test\_db

用户名：unicorn

密码：12345678

EC2配置

安装Python3

安装pymsql包

安装mysql客户端

1. 在test\_db中创建一张“kv”表

创建kv表的命令

CREATE TABLE IF NOT EXISTS `kv`(

`mykey` VARCHAR(100) NOT NULL,

`myvalue` VARCHAR(40) NOT NULL,

PRIMARY KEY ( `mykey` )

)DEFAULT CHARSET 'UTF8';

1. 在EC2中创建一个文件，insert.py

import pymysql

import time

table\_name = "kv"

db\_name = "test\_db"

user = "unicorn"

pwd = "12345678"

host = "<your RDS host>"

mykey=1

myvalue=1

while True:

db = pymysql.connect(host, user, pwd, db\_name)

cursor = db.cursor()

try:

sql = 'INSERT INTO kv(mykey,myvalue) VALUES(%s,%s)'

args = (str(mykey),str(myvalue))

cursor.execute(sql, args=args)

db.commit()

db.close()

except:

print("error")

mykey+=1

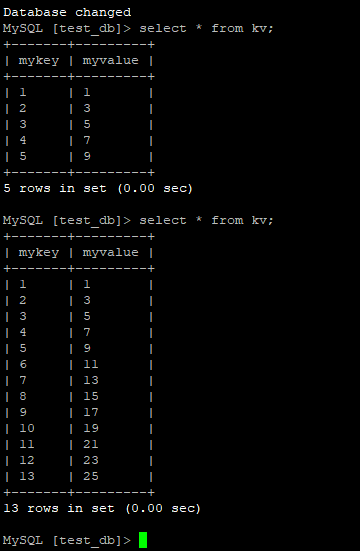
myvalue+=2

time.sleep(30)

运行命令

python3 insert.py &

回去检查RDS，kv表中是否插入了数据



好了，我们完成了一个正在持续写入数据的数据库的模拟环境。

1. 创建一个新数据库，老数据库的数据会持续往里面写入

RDS配置

子网：位于私网中

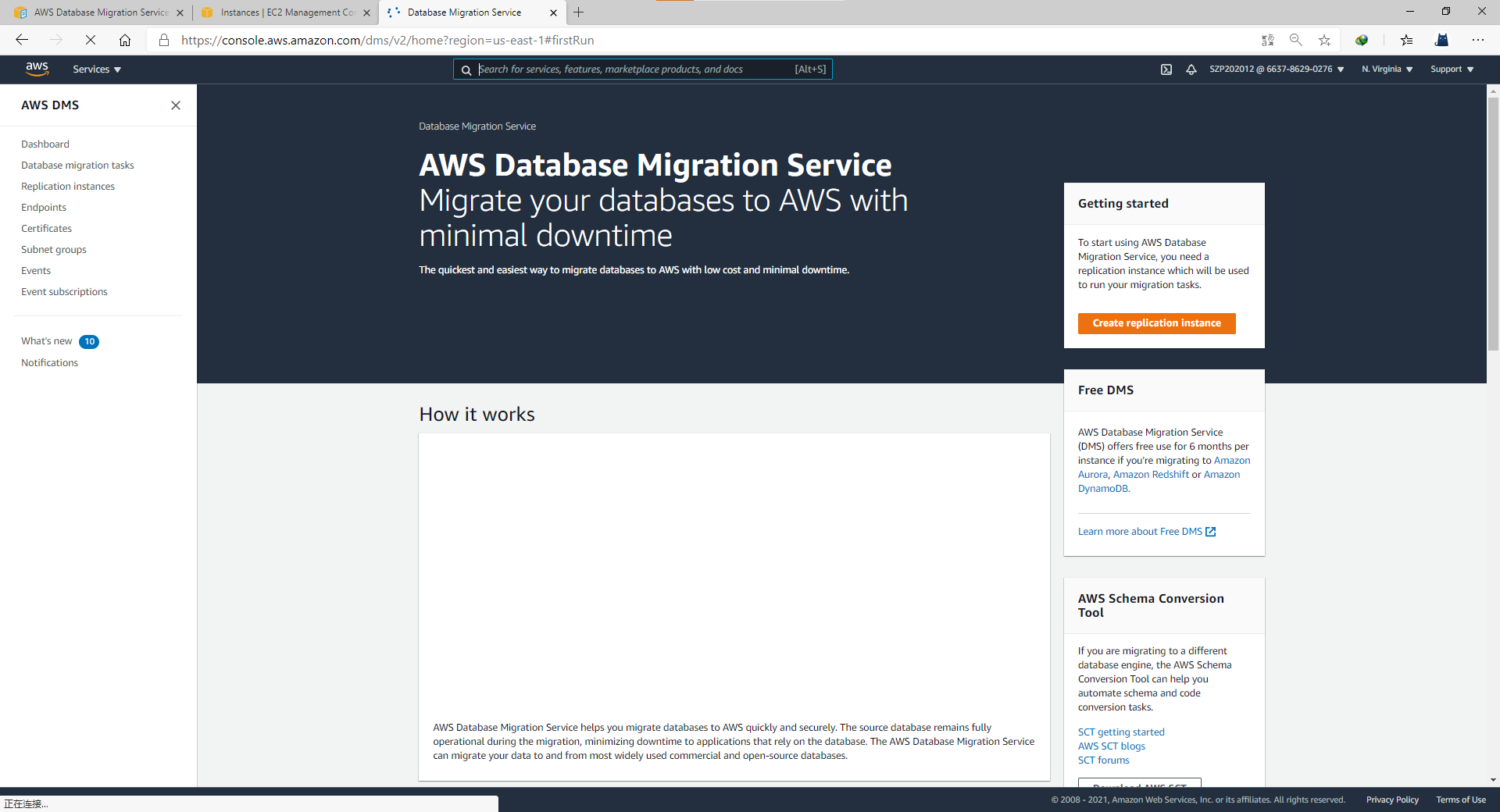
引擎：Mysql

参数组：刚刚创建的参数组

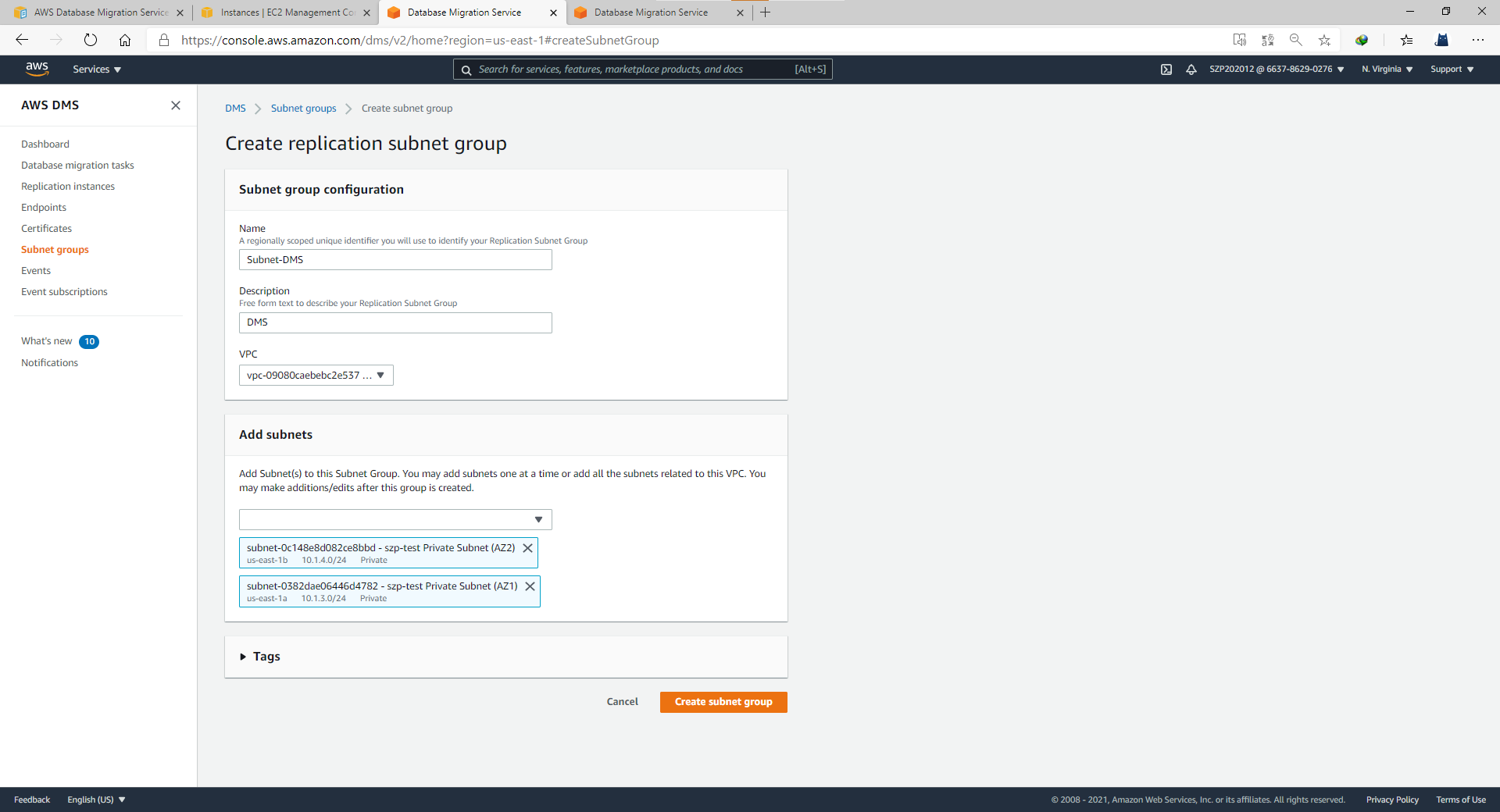
用户名：unicorn

密码：12345678

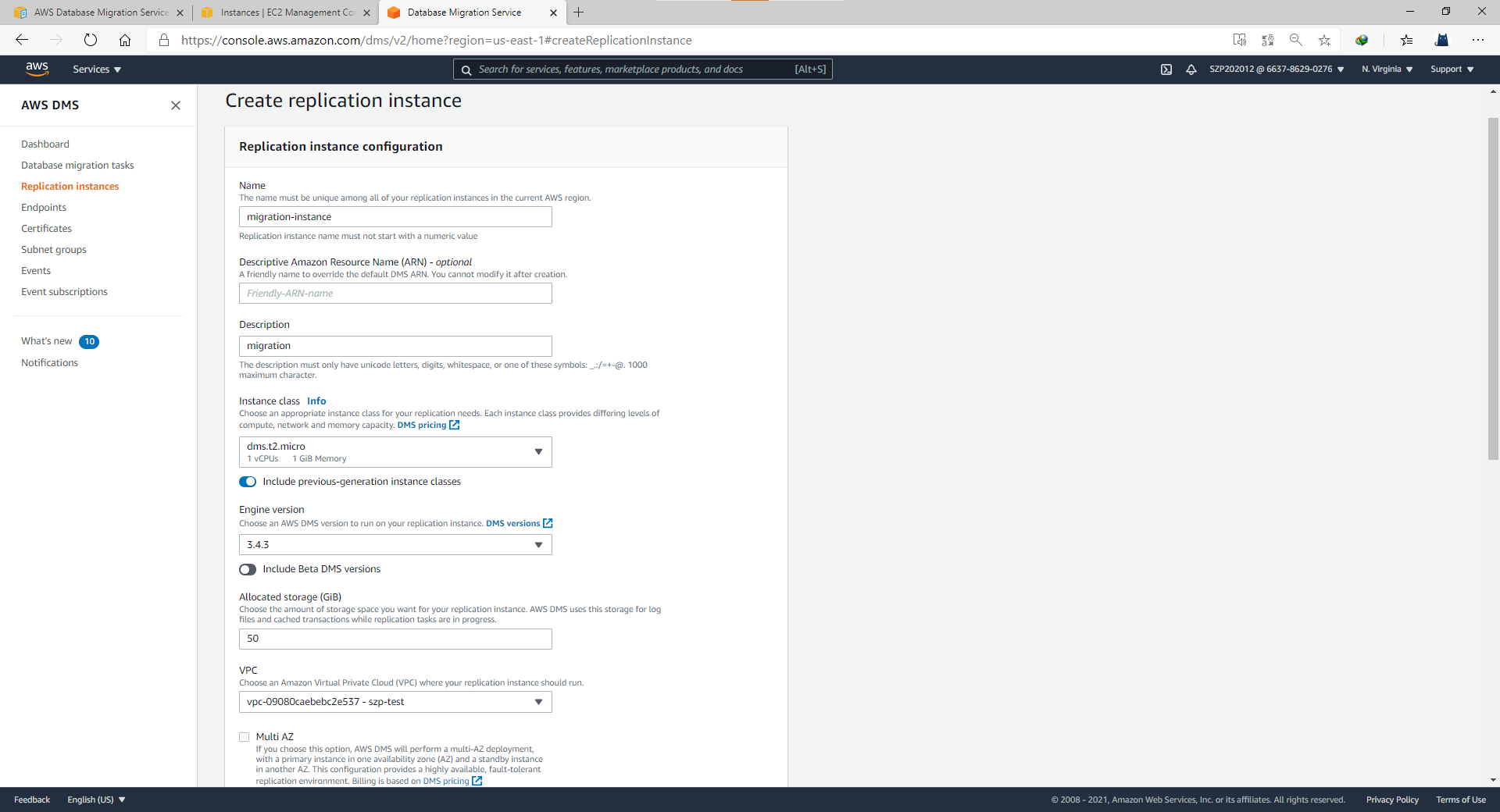
1. 现在打开DMS的主界面，可以开始配置迁移工作了



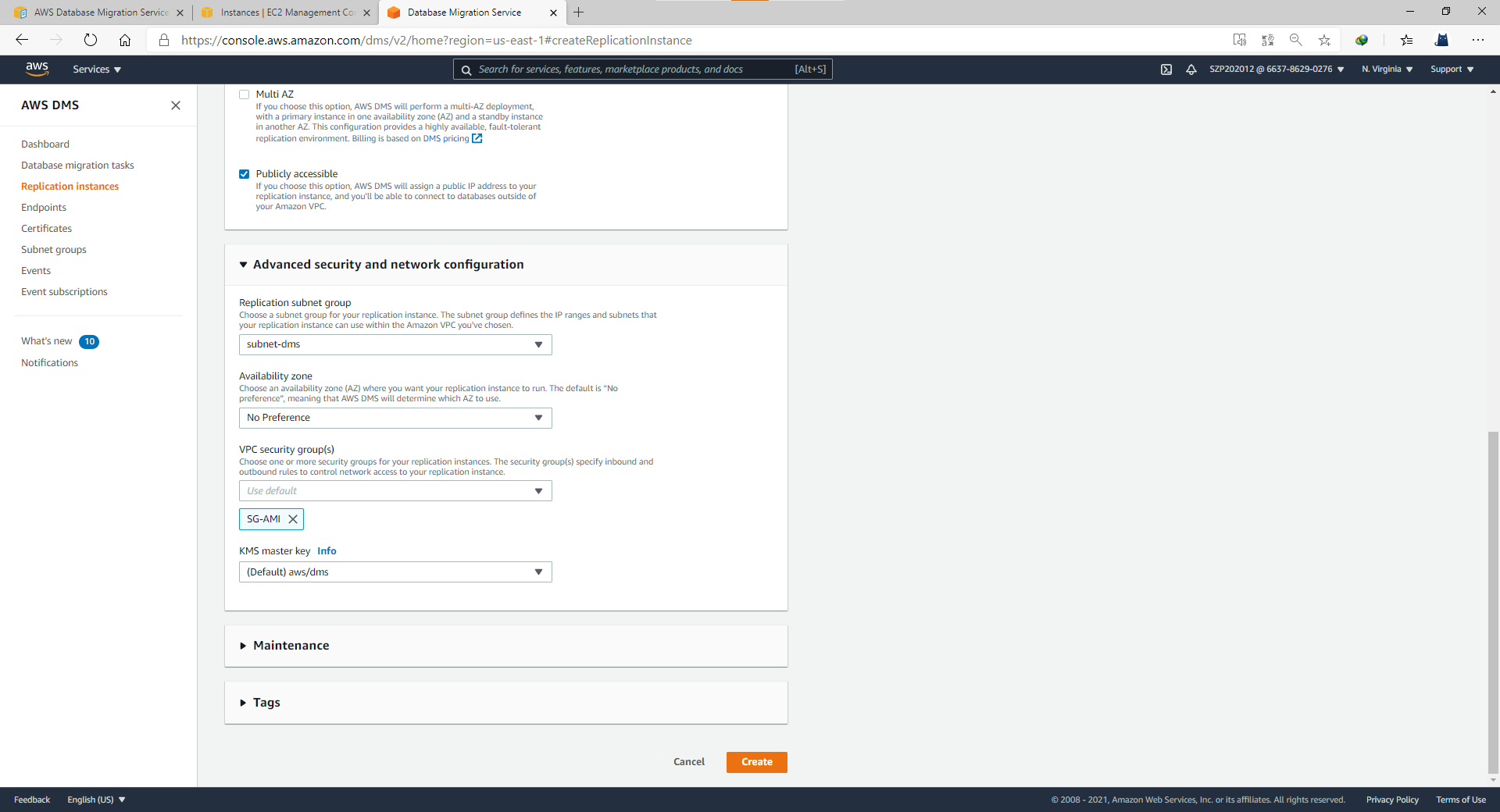
1. 创建一个子网组，选择私有子网，毕竟没人想被别人看见这些秘密工作吧？



1. 点击Replication instances，创建一台迁移实例，迁移工作将由它来进行

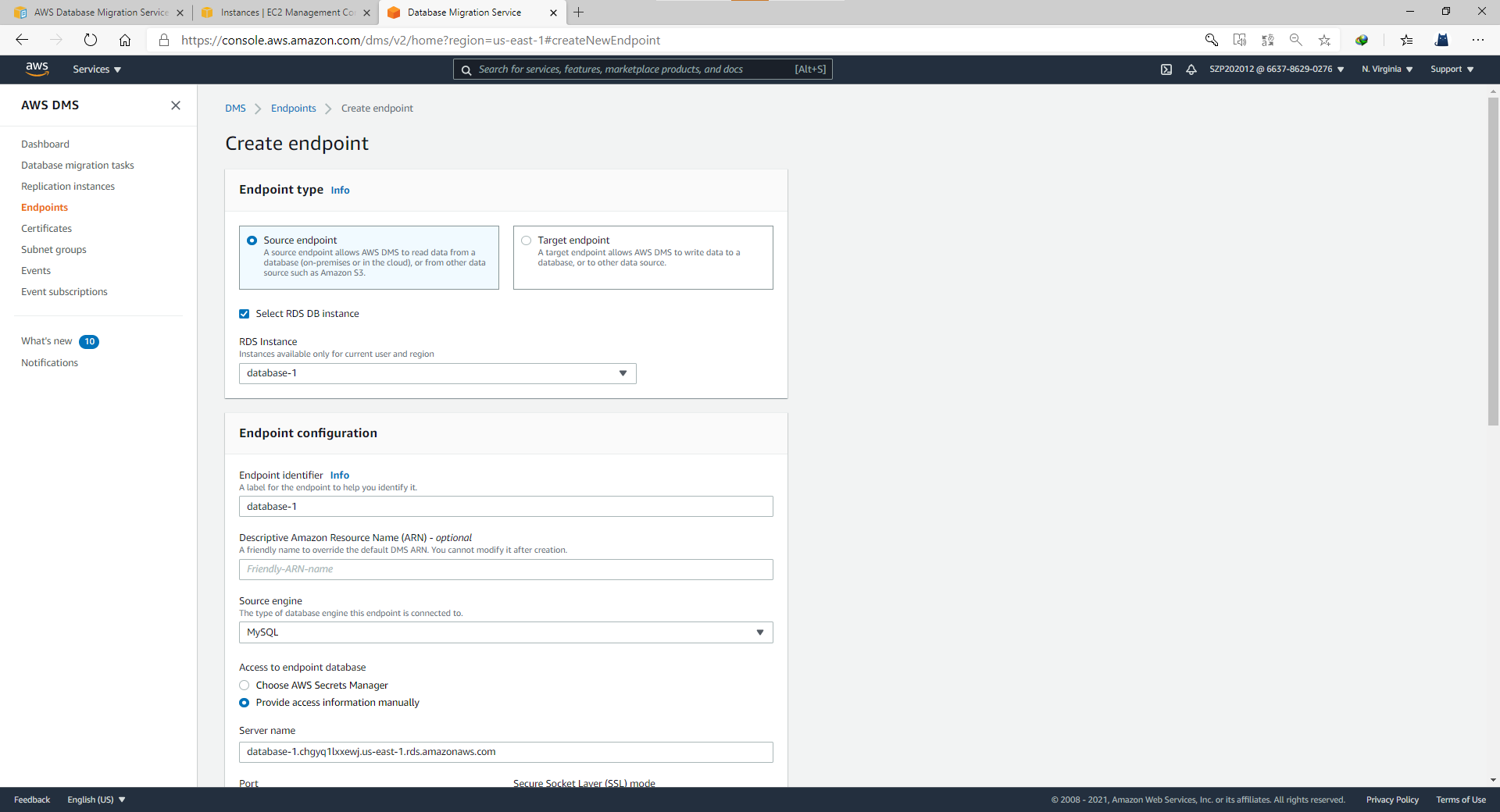


选择安全组和子网组，安全组一定要可以访问数据库才行！

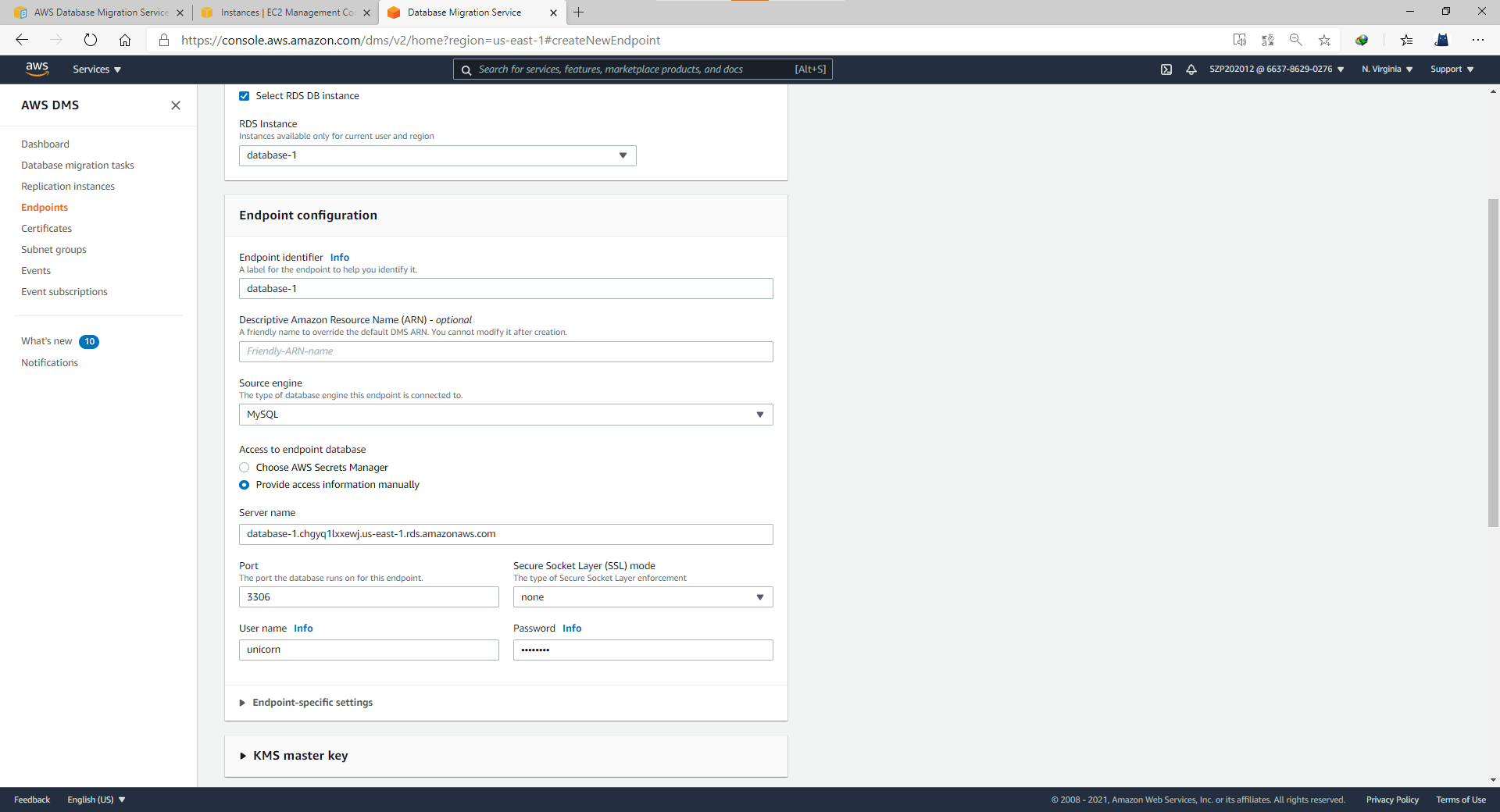


1. 点击Endpoint界面，创建一个源终端节点，也就是来源数据库

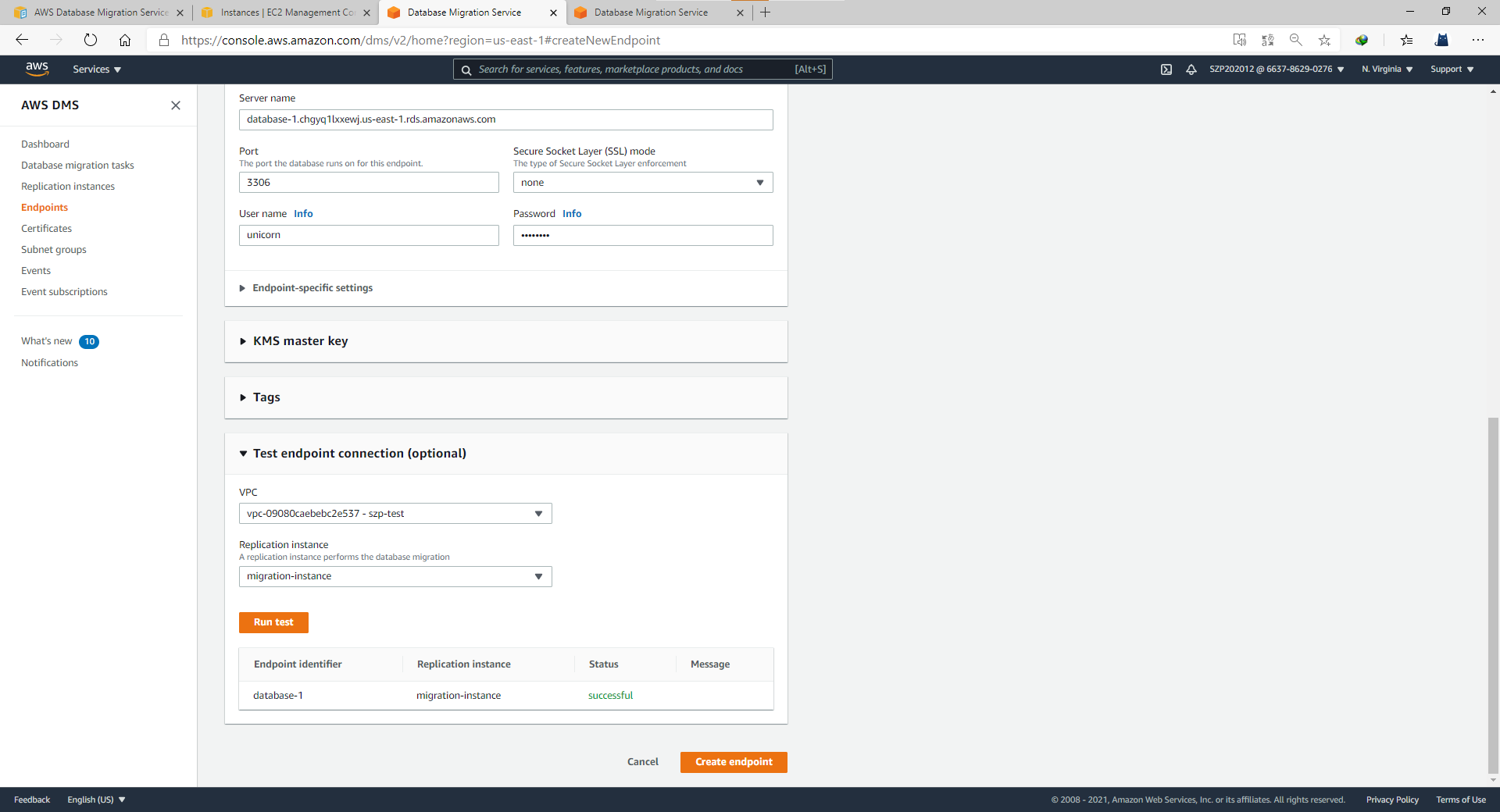
勾选Select RDS DB instance，这会弹出数据库让我们选择，当然如果是第三方数据库来源则不要点选



选择Provide access information manually，手动填写密码

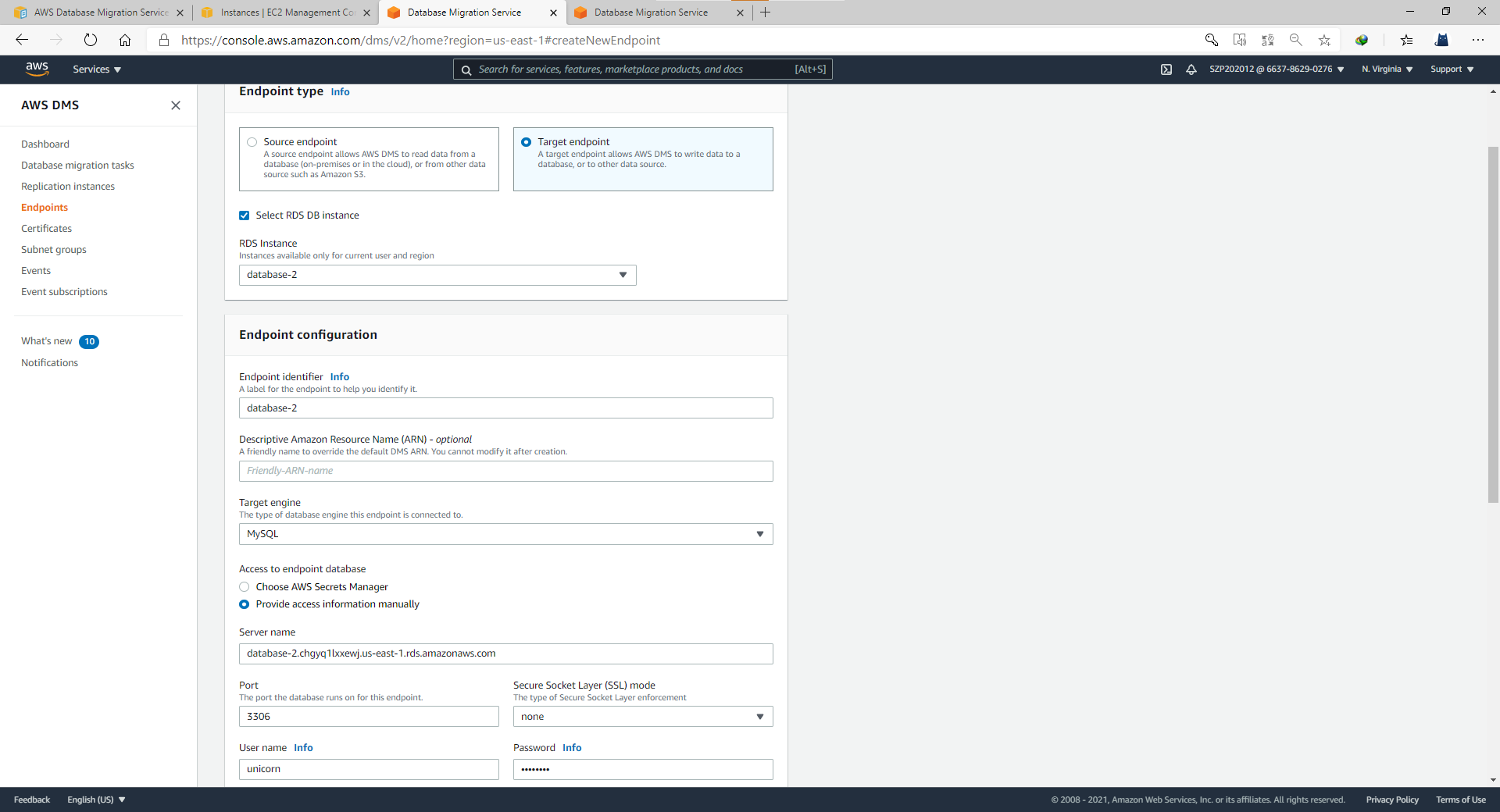


拉到最下方，有个Test endpoint connection，测试连通性，用来测试迁移实例能否连通你的数据库

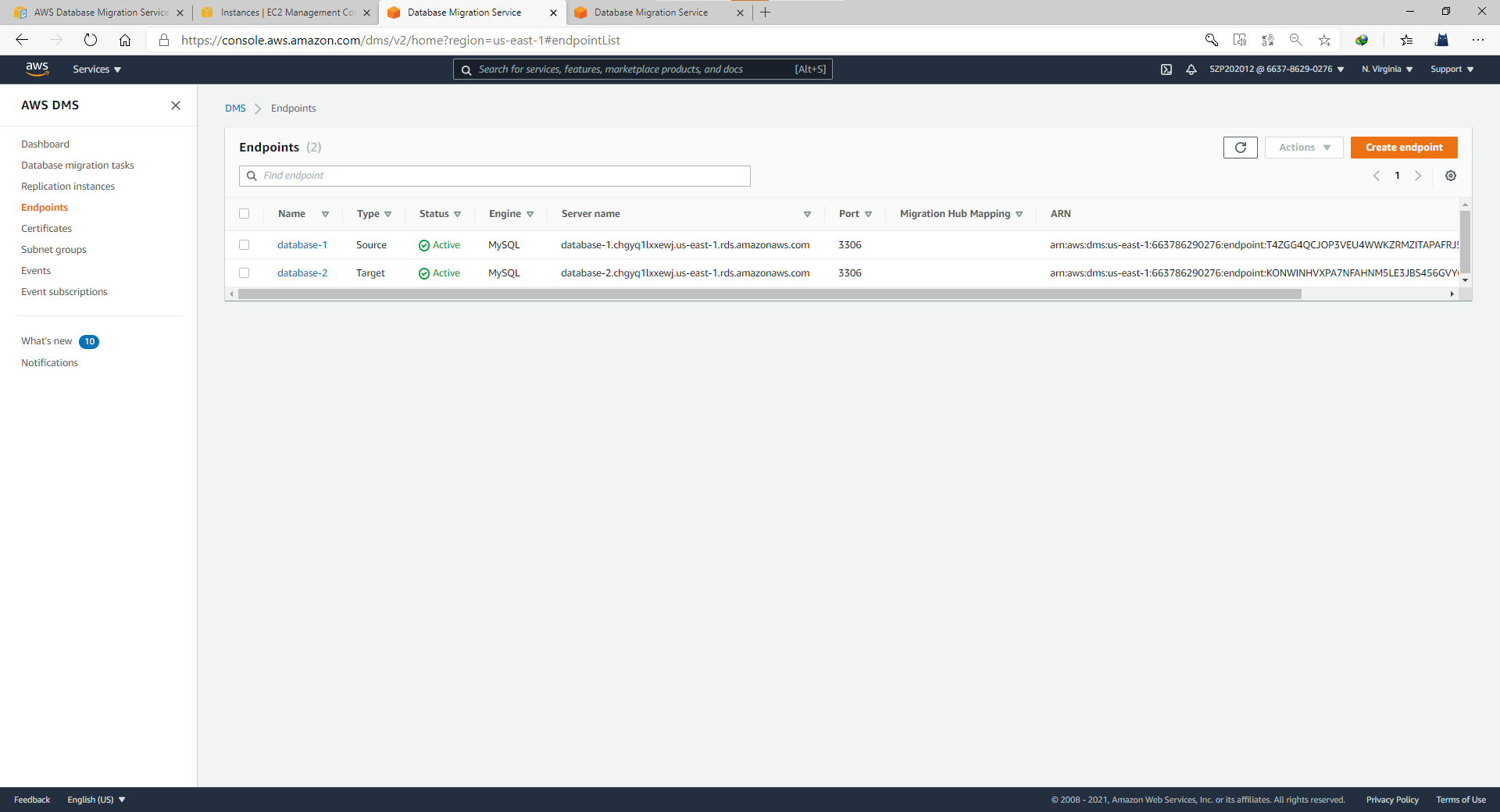


1. 我们现在再创建一个目标终端节点，也就是目标数据库

选择Target endpoint，其他操作和源目标节点的创建类似

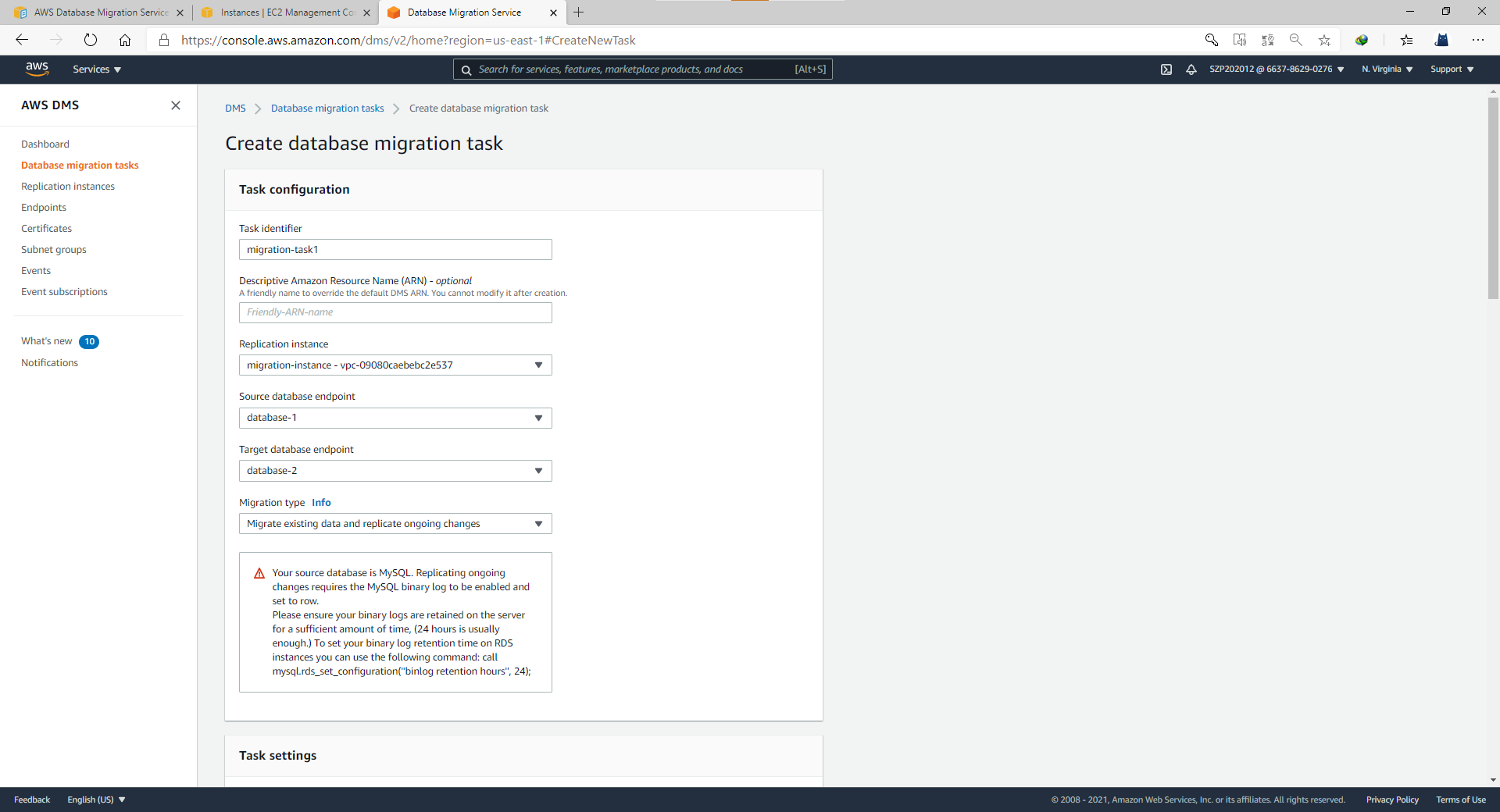


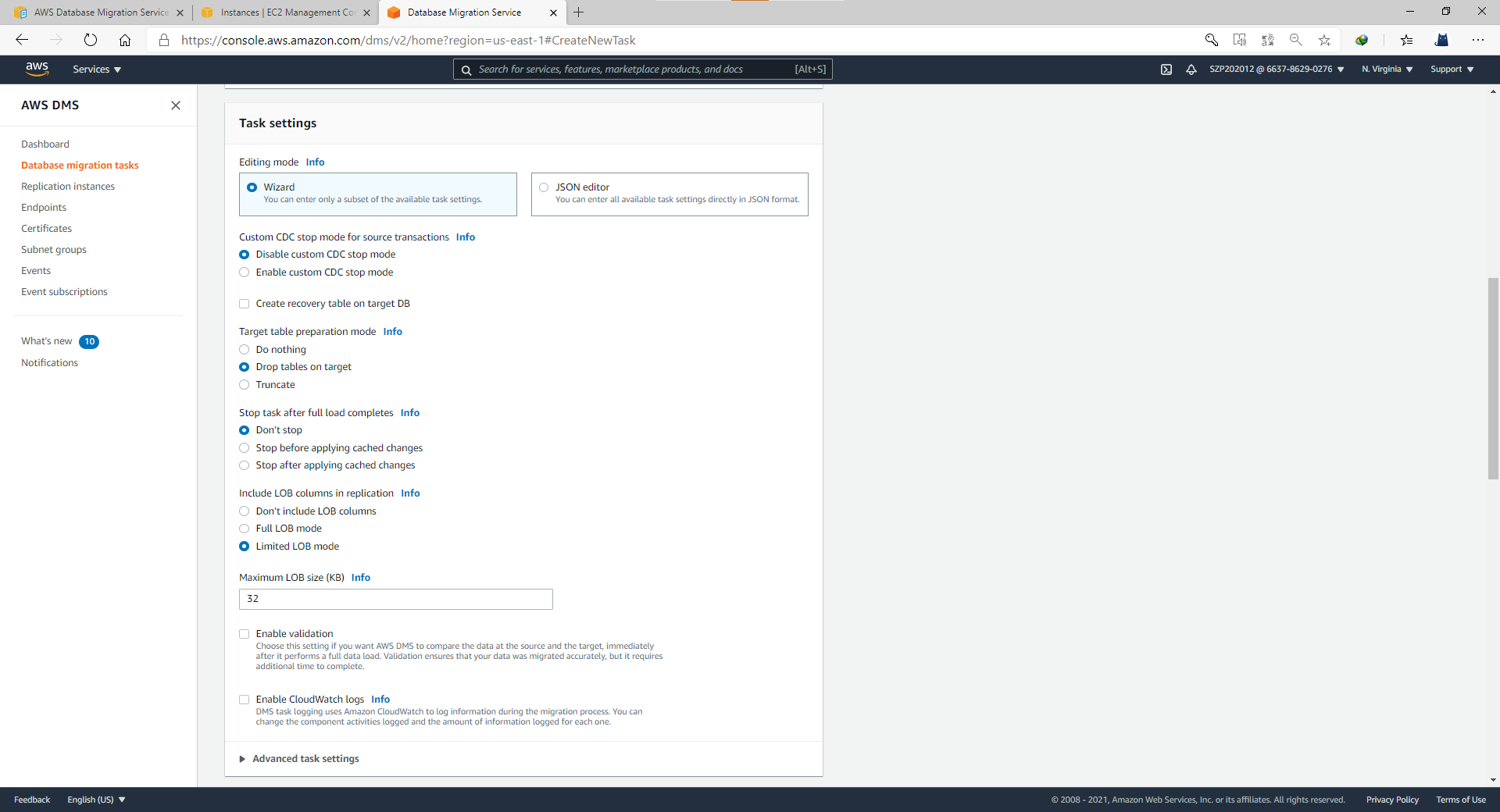
就这样两个终端节点创建完成了，且都可连通



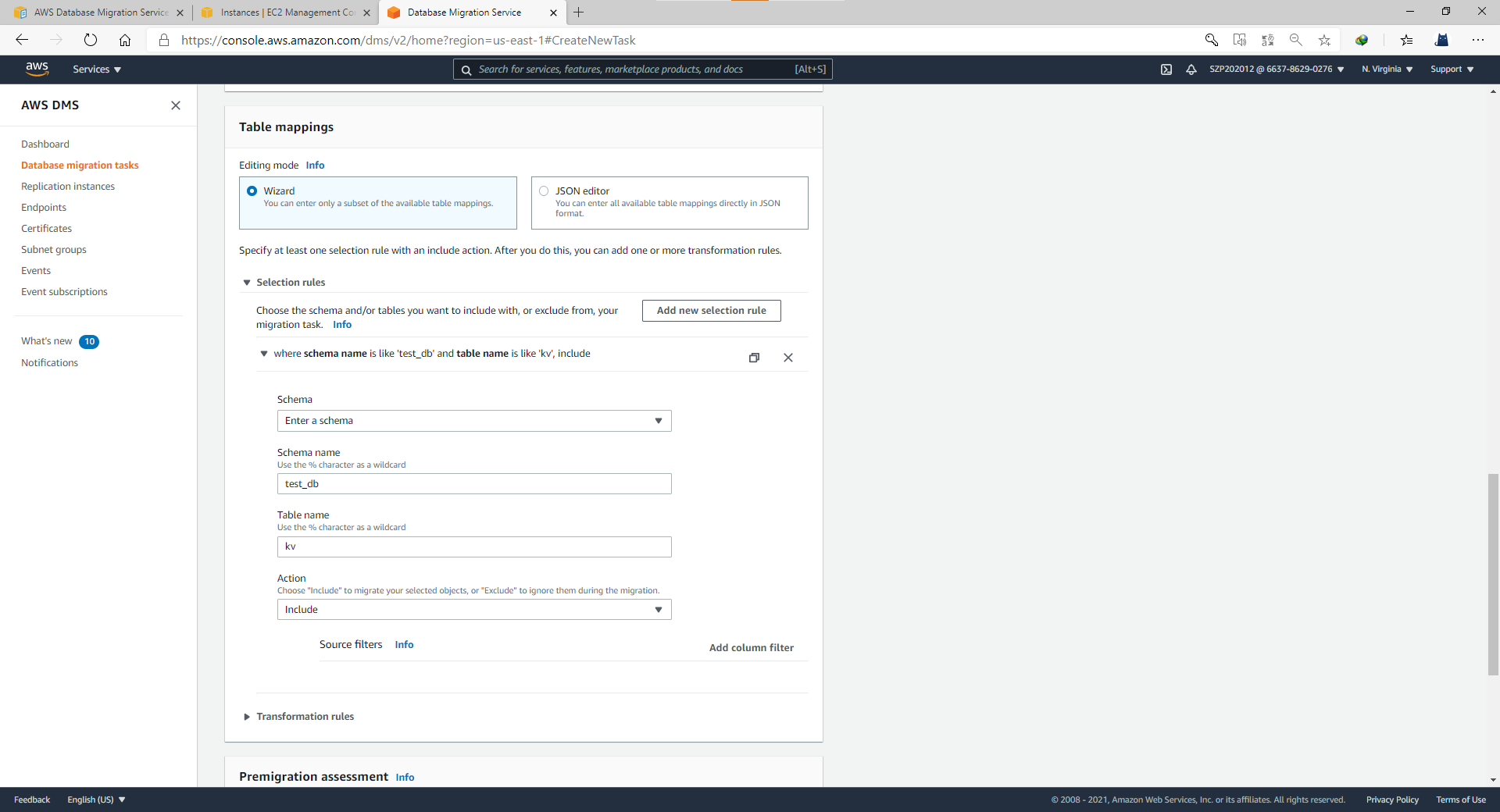
10、现在选择Database migration tasks，创建迁移任务

选择迁移实例，源节点，目标终端节点，Migration type选择Migrate existing data and replicate ongoing changes（迁移现有和正在写入的数据）

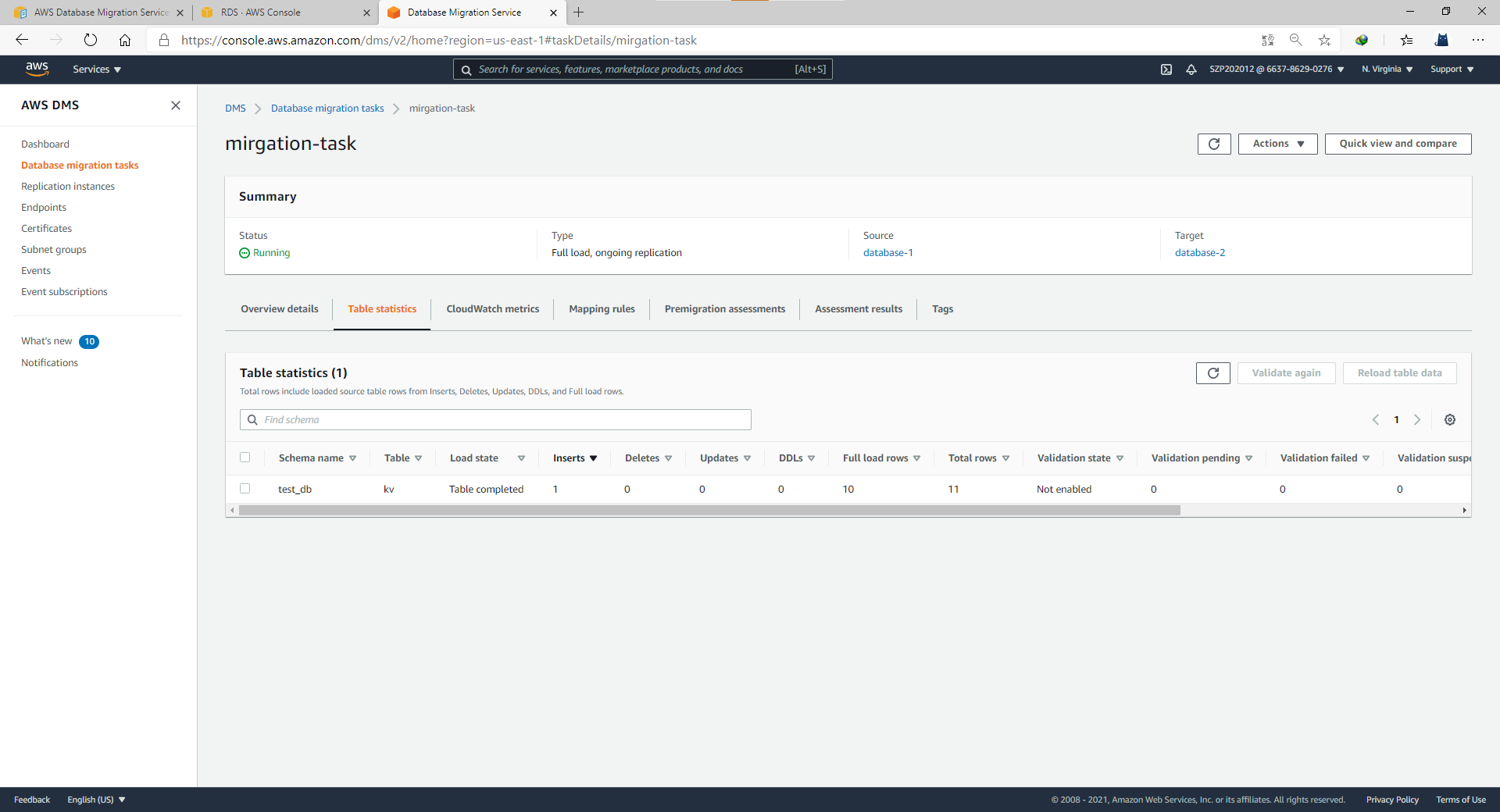




添加规则点击Add new selection rule，以下规则是迁移test\_db内的kv表



开始迁移了



这时候去看目标数据库就会看见一个一样的数据库出现了，表的数据可能会有一定的延迟。恭喜！实验成功！