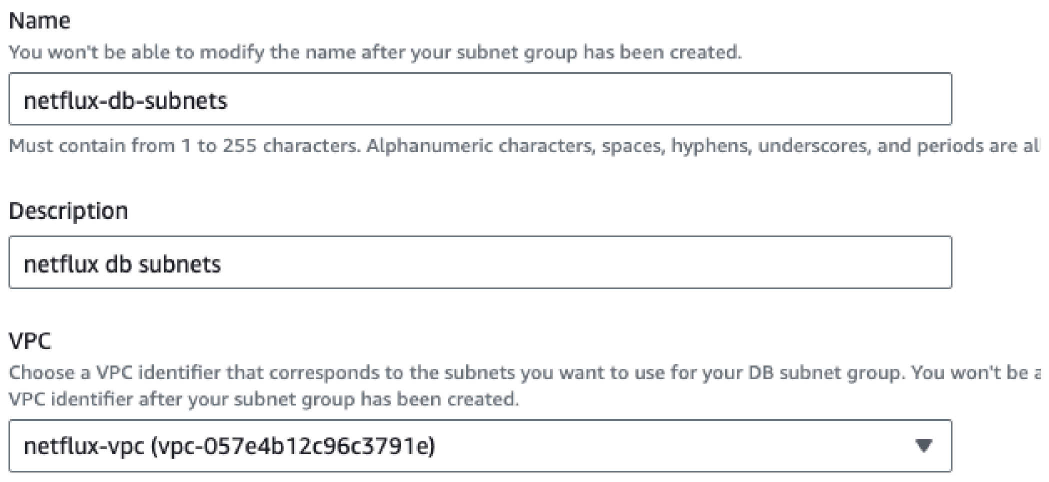
RDS & Secrets Manager

Create DB Subnet Group

- First let's create a DB subnet group.
    - Give some name and select the VPC

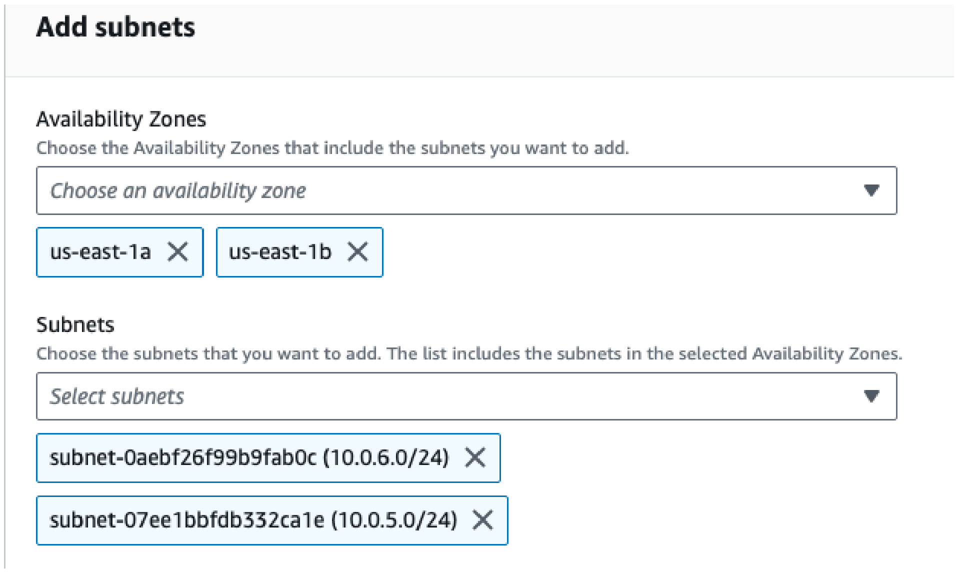


Name
You won't be able to modify the name after your subnet group has been created.
netflux-db-subnets
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are al

Description
netflux db subnets

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be a
VPC identifier after your subnet group has been created.
netflux-vpc (vpc-057e4b12c96c3791e)

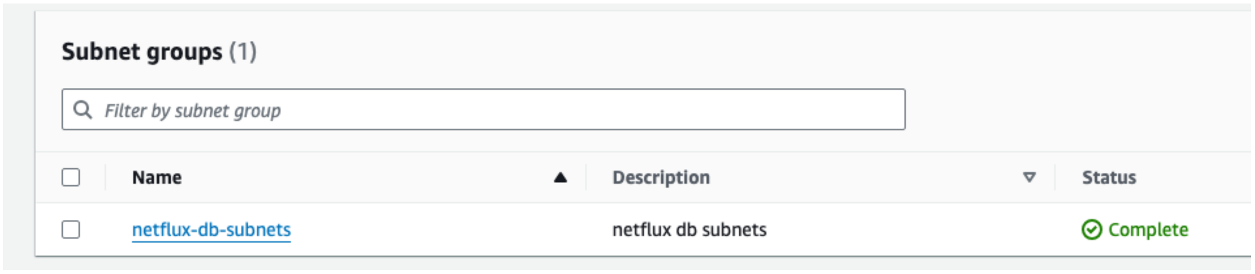
- Select the subnets. In our case 10.0.5.0/24 and 10.0.6.0/24 were created for db.


Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a
us-east-1b

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-0aebf26f99b9fab0c (10.0.6.0/24)
subnet-07ee1bbfdb332ca1e (10.0.5.0/24)

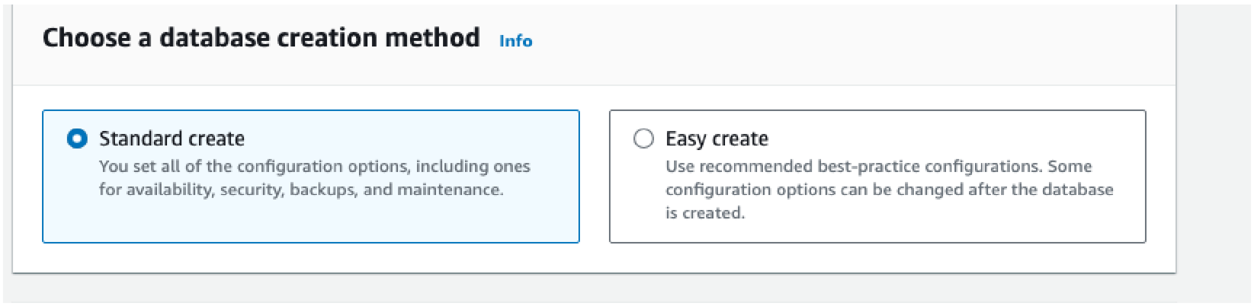
- Create


Subnet groups (1)

Filter by subnet group

Name	Description	Status
netflux-db-subnets	netflux db subnets	Complete

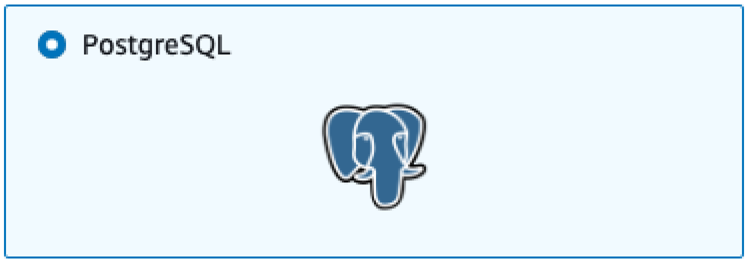
Create Database Instance


Choose a database creation method Info

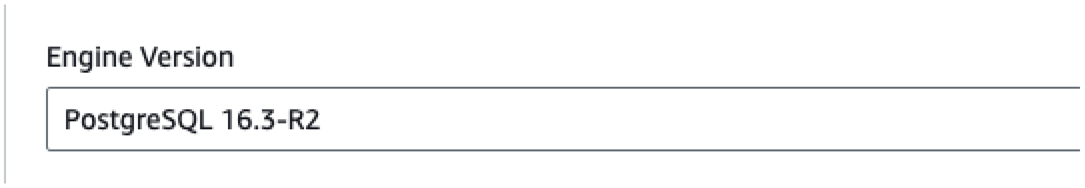
[✓ Standard create]
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

- Select postgres



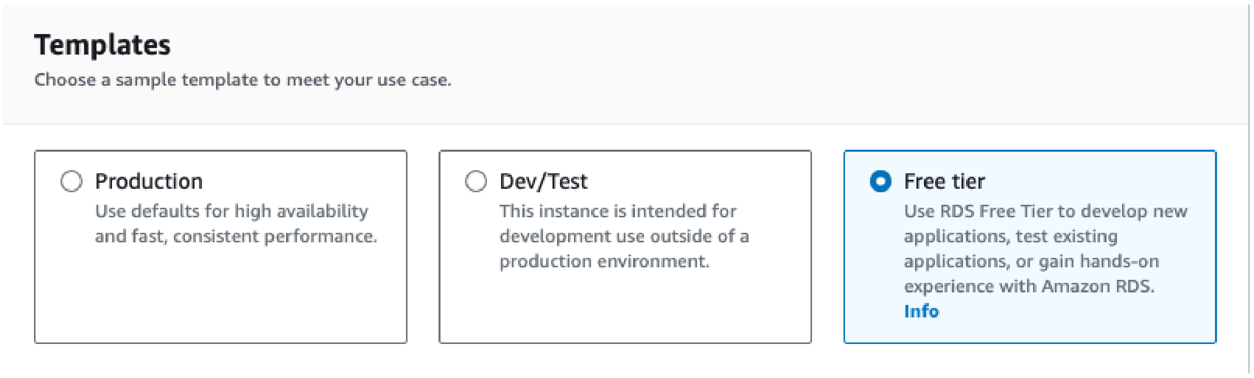
- Engine version can be latest



Engine Version

PostgreSQL 16.3-R2

- For our learning purposes, Let’s use the free tier. But for production application, choose Production with multi AZ



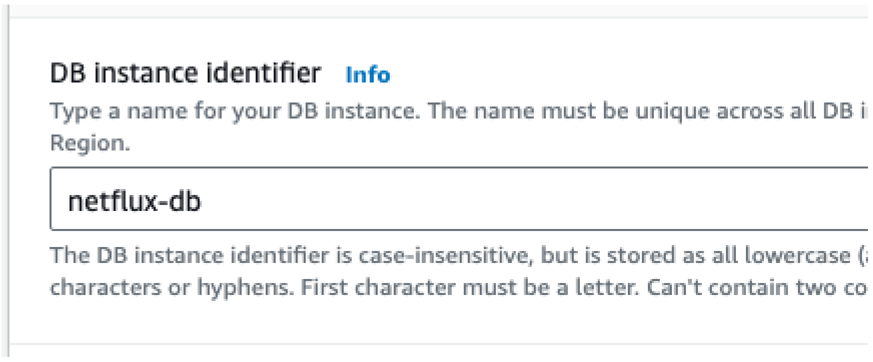
Templates
Choose a sample template to meet your use case.

( ) Production
Use defaults for high availability and fast, consistent performance.

( ) Dev/Test
This instance is intended for development use outside of a production environment.

(●) Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
Info

- Give a name for the DB Instance

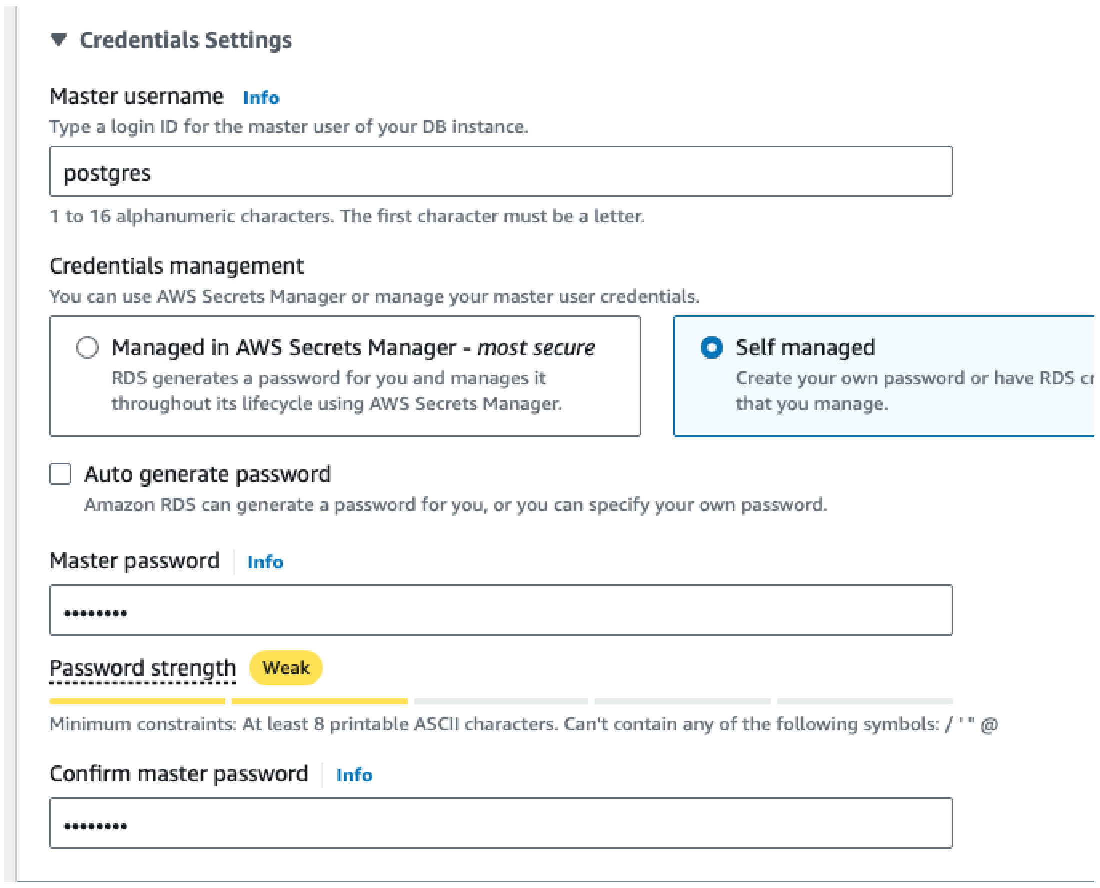


DB instance identifier Info
Type a name for your DB instance. The name must be unique across all DB i
Region.

netflux-db

The DB instance identifier is case-insensitive, but is stored as all lowercase (
characters or hyphens. First character must be a letter. Can't contain two co

- I give the credentials (for learning purposes) **postgres / admin123**



▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

[✓ Self managed]
Create your own password or have RDS cr
that you manage.

☐ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

••••••••

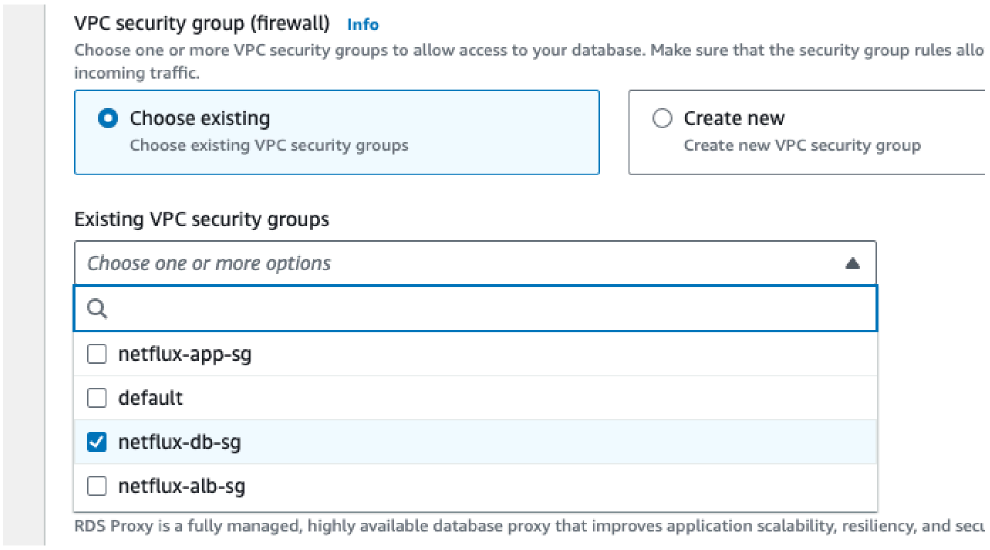
Password strength Weak

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password Info

••••••••

- Security Group - We will choose the DB security group and attach it to the DB



VPC security group (firewall) Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allo
incoming traffic.

[✓ Choose existing]
Choose existing VPC security groups

○ Create new
Create new VPC security group

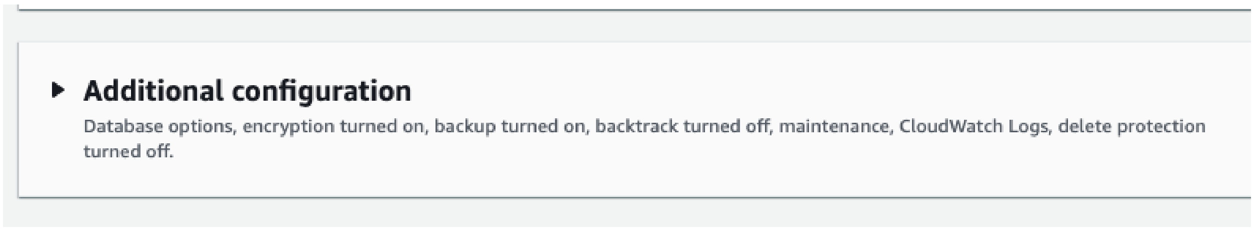
Existing VPC security groups

Choose one or more options

☐ netflux-app-sg
☐ default
☑ netflux-db-sg
☐ netflux-alb-sg

RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and secu

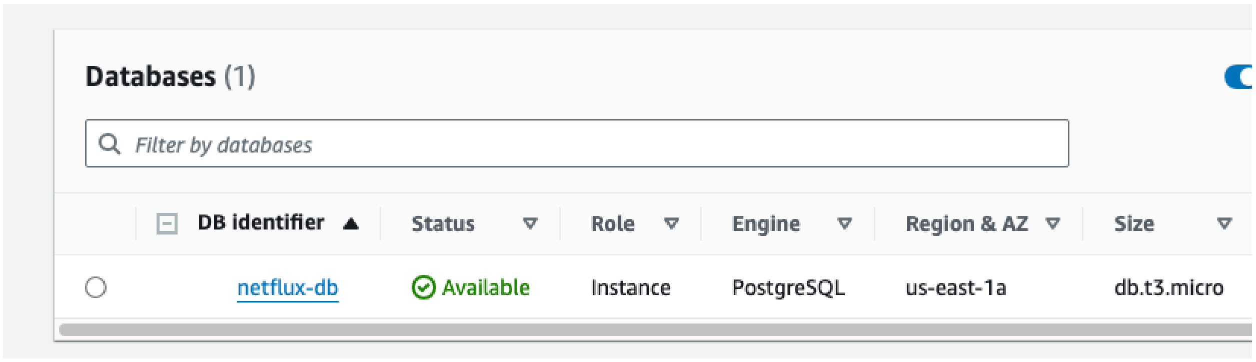
- No additional configuration is required



Additional configuration

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

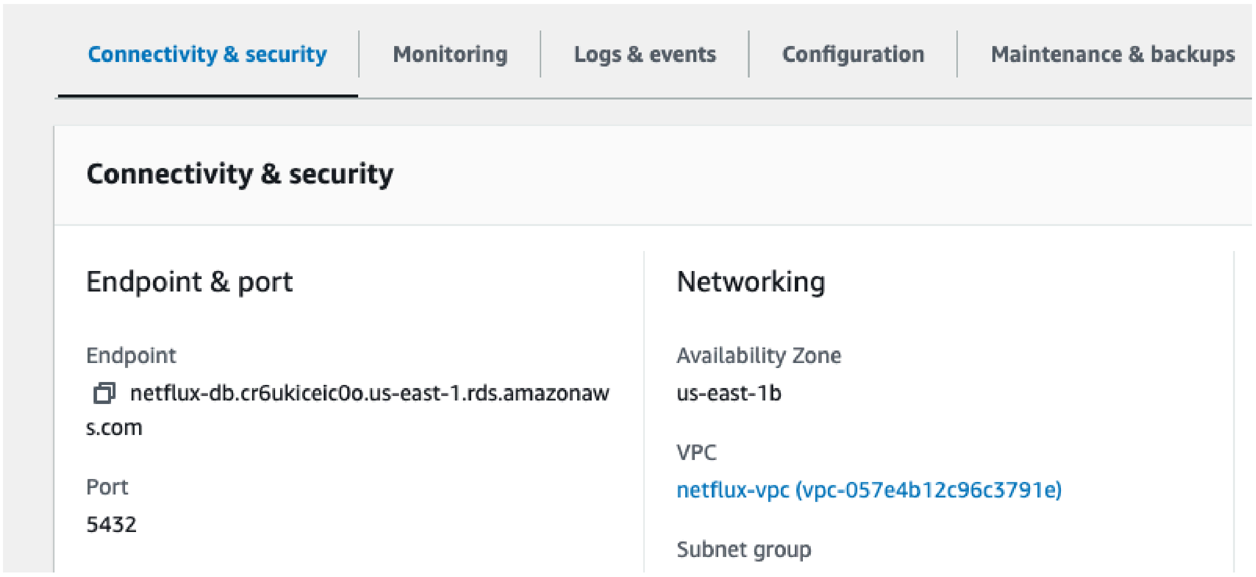
- Click on “Create database”. It might take 10+ minutes. Wait for the status to be “Available”



Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size
netflux-db	Available	Instance	PostgreSQL	us-east-1a	db.t3.micro

- What we created is the DB Instance!
- Click on the DB Instance to get DB connectivity details



Connectivity & security

Endpoint & port

Endpoint
netflux-db.cr6ukiceic0o.us-east-1.rds.amazonaws.com

Port
5432

Networking

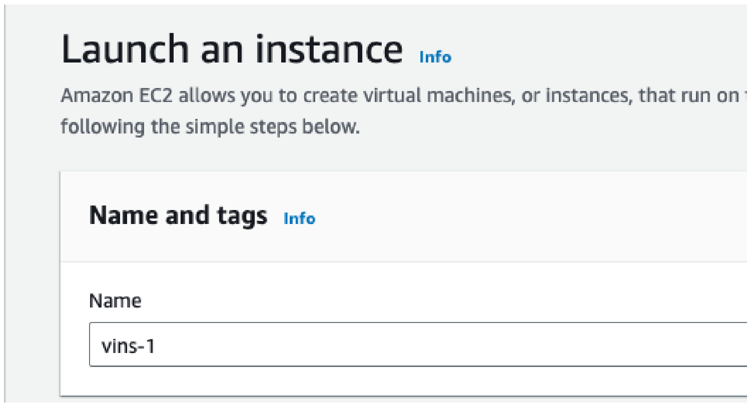
Availability Zone
us-east-1b

VPC
netflux-vpc (vpc-057e4b12c96c3791e)

Subnet group

Initializing Database

- Once the DB Instance is up and running, We need to create databases with our tables, data etc.
- Go to EC2 to create a simple instance



Launch an instance Info

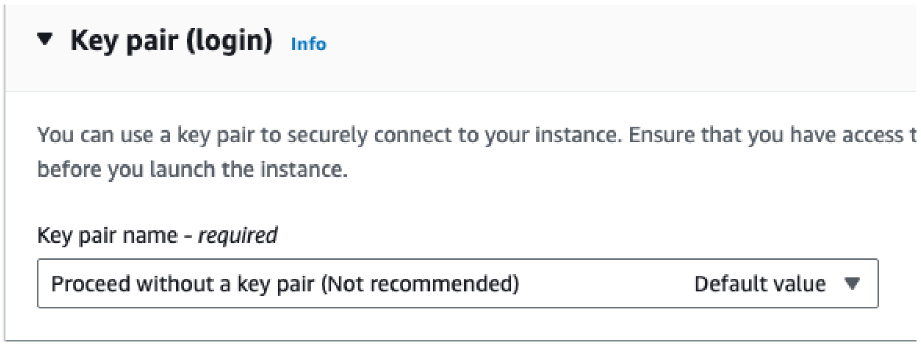
Amazon EC2 allows you to create virtual machines, or instances, that run on
following the simple steps below.

Name and tags Info

Name

vins-1

- Choose our AMI which has the ***psql*** installed
- No Key pair is required. We will destroy this instance immediately.



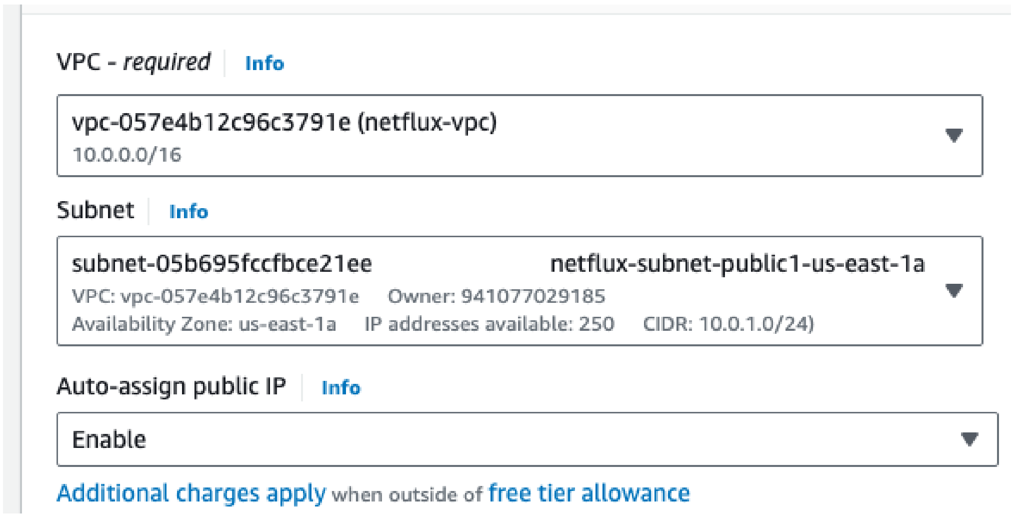
Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access t
before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended) Default value

- Network Settings
    - Keep this in the public subnet
    - We need to assign public IP



VPC - *required* Info

vpc-057e4b12c96c3791e (netflux-vpc)
10.0.0.0/16

Subnet Info

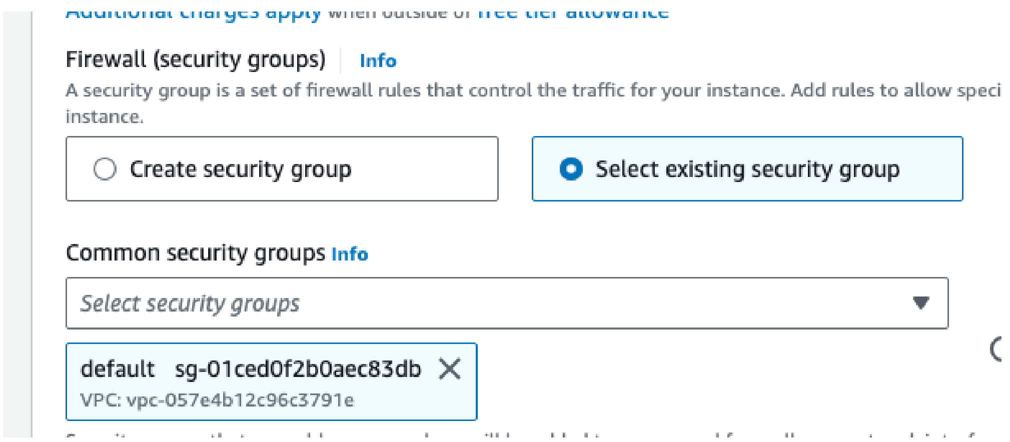
subnet-05b695fccfbce21ee netflux-subnet-public1-us-east-1a
VPC: vpc-057e4b12c96c3791e Owner: 941077029185
Availability Zone: us-east-1a IP addresses available: 250 CIDR: 10.0.1.0/24)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

- Let’s attach default Security Group


Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow speci instance.

( ) Create security group [✓ Select existing security group]

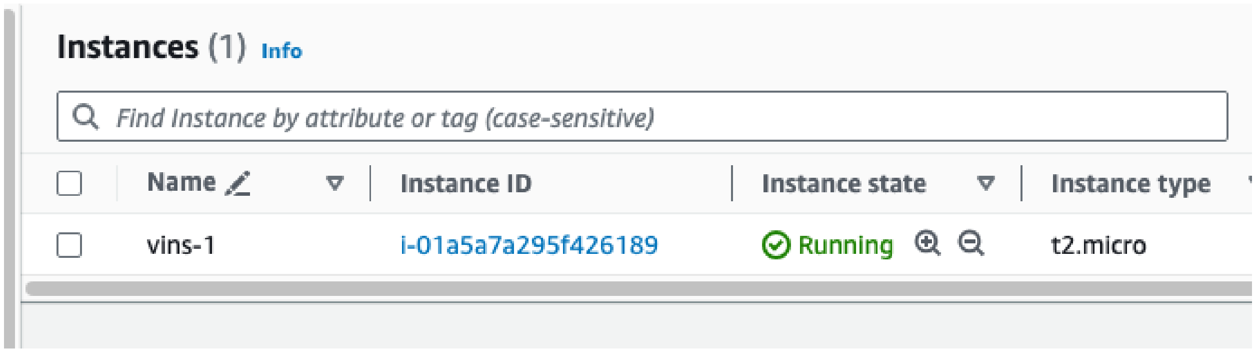
Common security groups Info

Select security groups

default sg-01ced0f2b0aec83db

VPC: vpc-057e4b12c96c3791e

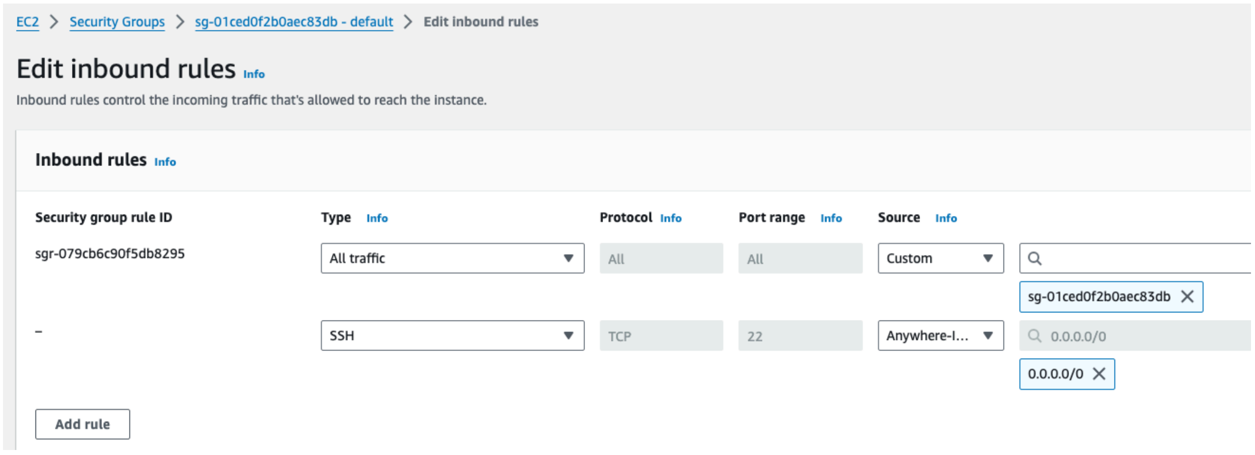
- Everything else is optional
- Create the instance


Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type
☐	vins-1	i-01a5a7a295f426189	✓ Running	t2.micro

- Let's open the "default" security group. allow port 22 for SSH access


EC2 > Security Groups > sg-01ced0f2b0aec83db - default > Edit inbound rules

Edit inbound rules Info

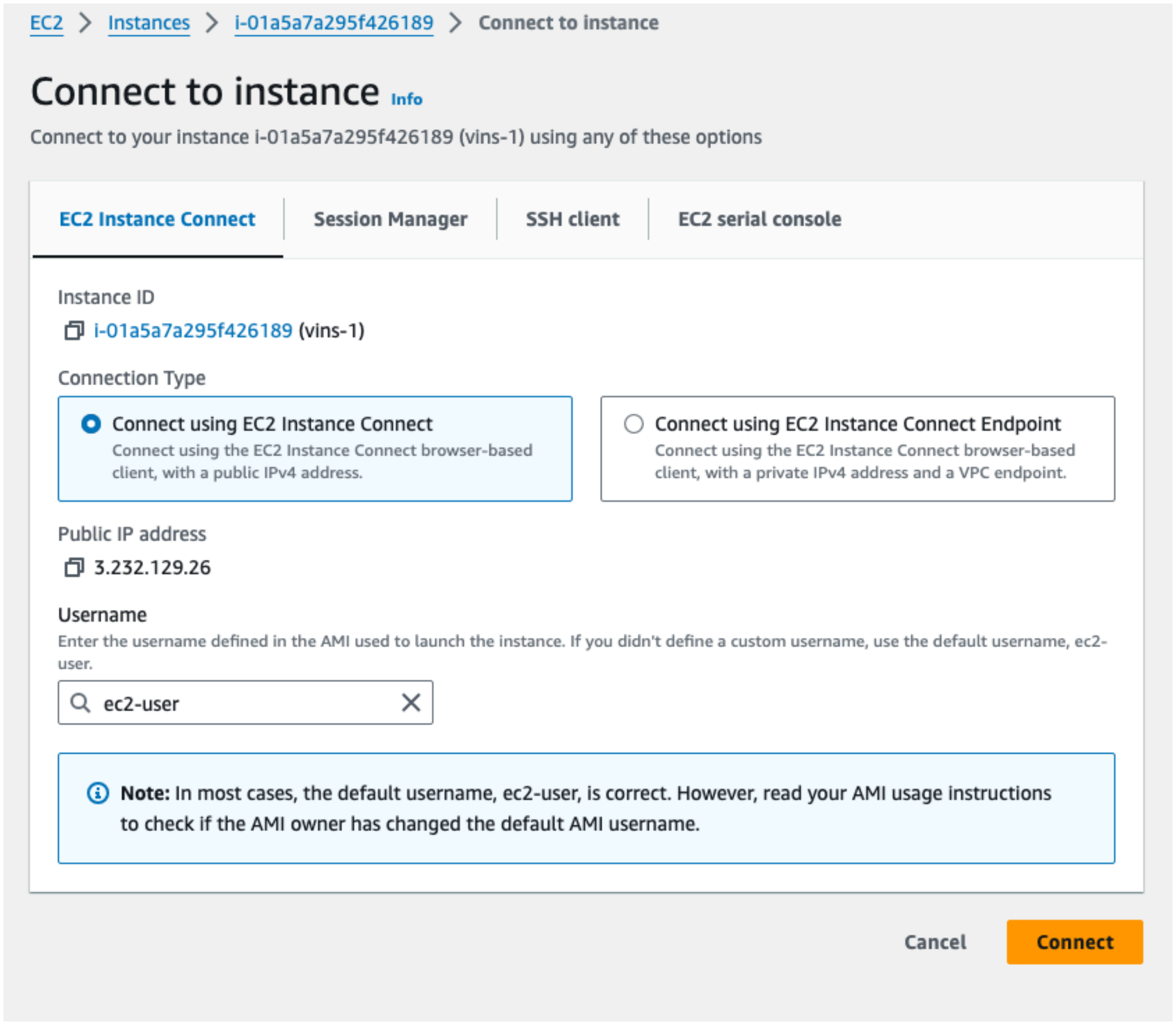
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source	
sgr-079cb6c90f5db8295	All traffic	All	All	Custom	sg-01ced0f2b0aec83db
–	SSH	TCP	22	Anywhere-I...	0.0.0.0/0

Add rule

- Important: Also temporarily allow the default security group to access the postgres
    - **netflux-db-sg**
- Go back to EC2, connect to this EC2 instance


EC2 > Instances > i-01a5a7a295f426189 > Connect to instance

Connect to instance Info

Connect to your instance i-01a5a7a295f426189 (vins-1) using any of these options

Instance ID
i-01a5a7a295f426189 (vins-1)

Connection Type
[✓ Connect using EC2 Instance Connect] Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint — Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
3.232.129.26

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

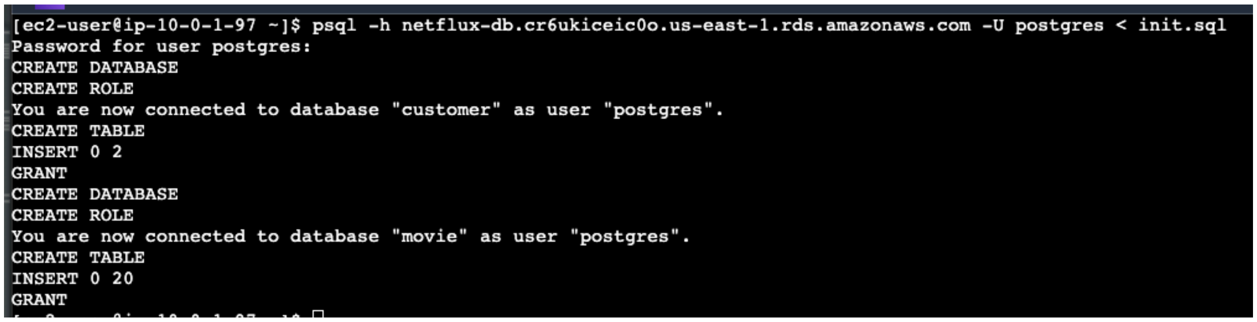
- Create a file **init.sql** and use the data I have provided.

```
cat > init.sql
```

- Then connect to the DB and run the init sql - Update the DB endpoint.

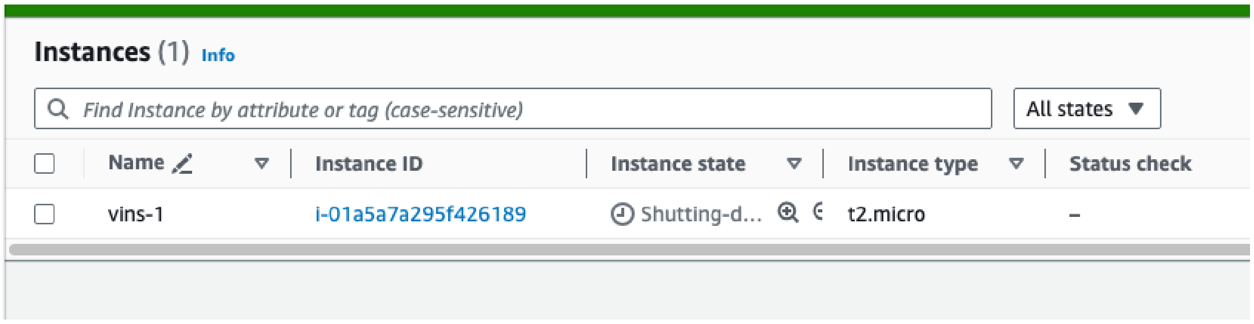
```
psql -U postgres -h netflux-db.cr6ukiceic0o.us-east-1.rds.amazonaws.com <
init.sql
```

- It will ask for the password. It is **admin123**
  - At this point, it will create 2 different databases for our application with 2 users for individual applications to access.



```
[ec2-user@ip-10-0-1-97 ~]$ psql -h netflux-db.cr6ukiceic0o.us-east-1.rds.amazonaws.com -U postgres < init.sql
Password for user postgres:
CREATE DATABASE
CREATE ROLE
You are now connected to database "customer" as user "postgres".
CREATE TABLE
INSERT 0 2
GRANT
CREATE DATABASE
CREATE ROLE
You are now connected to database "movie" as user "postgres".
CREATE TABLE
INSERT 0 20
GRANT
```

- We no longer need the EC2 instance. We can terminate.



Instances (1) Info

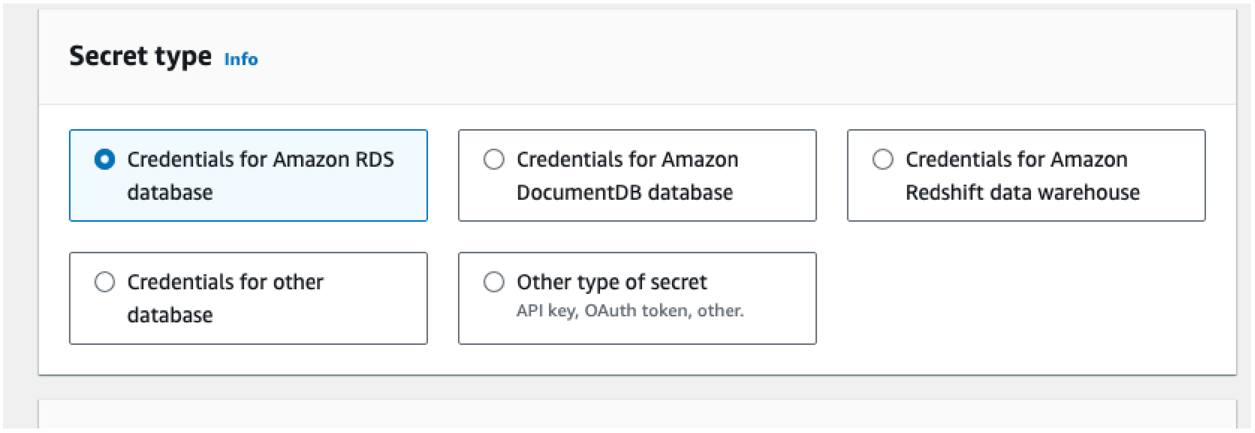
Name	Instance ID	Instance state	Instance type	Status check
vins-1	i-01a5a7a295f426189	Shutting-d...	t2.micro	–

- We can also remove
  - ***default*** security group - allow port 22 for ssh entry.
  - ***db*** security group - allow inbound from default security group

At this point, you can temporarily stop the DB instance and resume later.

Secrets Manager

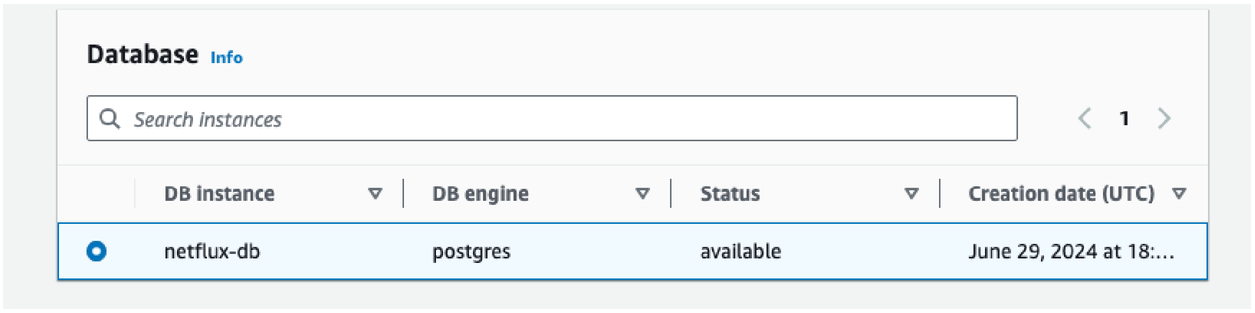
- Go to Secrets Manager to store these credentials



Secret type Info

[✓ Credentials for Amazon RDS database]
Credentials for Amazon DocumentDB database
Credentials for Amazon Redshift data warehouse
Credentials for other database
Other type of secret
API key, OAuth token, other.

- select the DB Instance

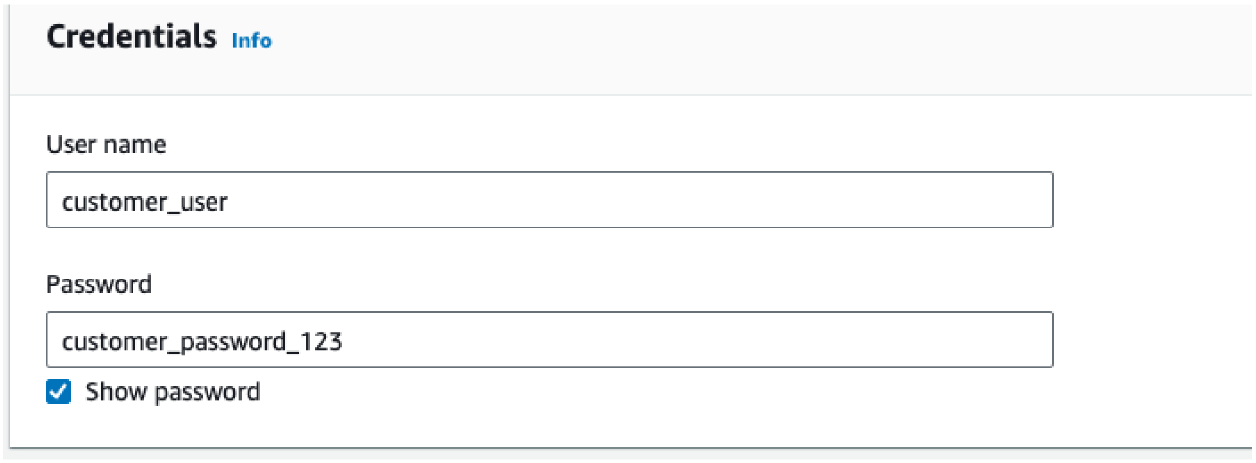


Database Info

Search instances

	DB instance	DB engine	Status	Creation date (UTC)
●	netflux-db	postgres	available	June 29, 2024 at 18:…

- Store the credentials for the database "customer"



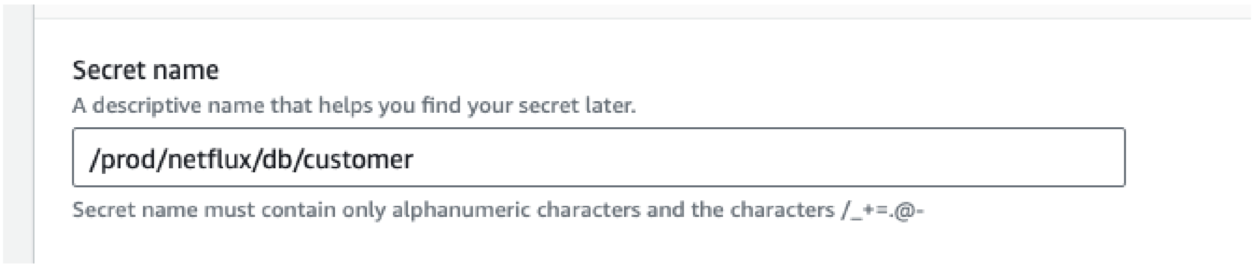
Credentials Info

User name: customer_user

Password: customer_password_123

☑ Show password

- provide a name for the secret. You can follow any meaningful naming convention.



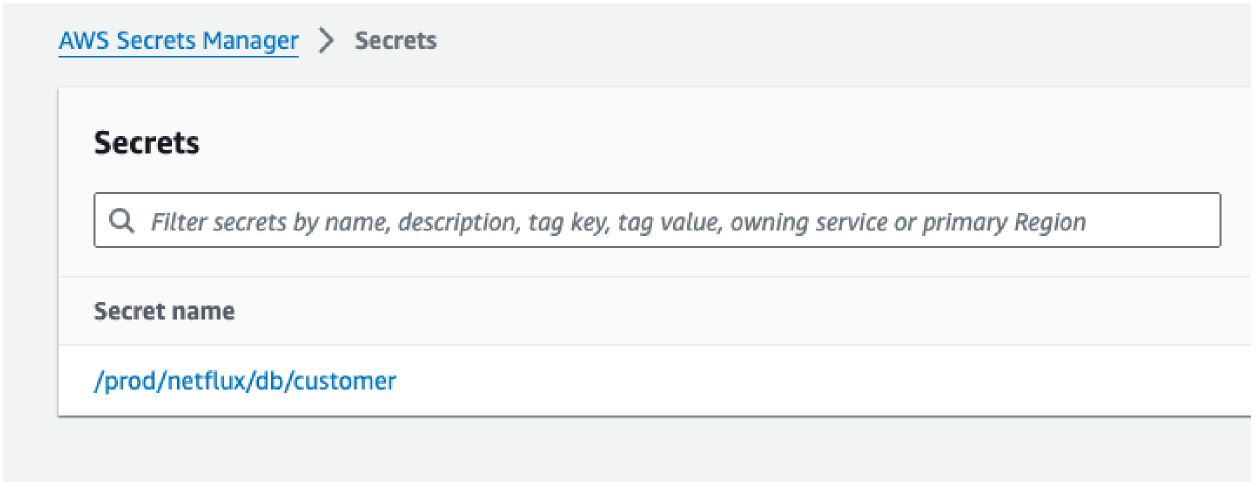
Secret name

A descriptive name that helps you find your secret later.

/prod/netflux/db/customer

Secret name must contain only alphanumeric characters and the characters /_+=.@-

- Click "Next" ... finally "Create"



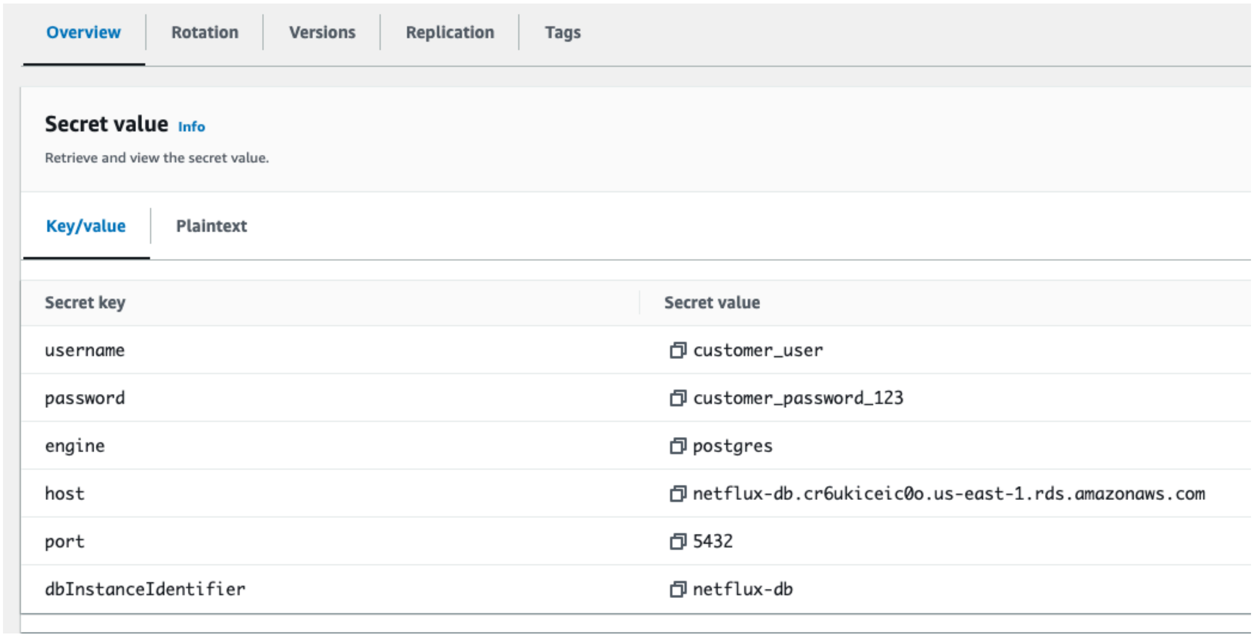
AWS Secrets Manager > Secrets

Secrets

Filter secrets by name, description, tag key, tag value, owning service or primary Region

Secret name
/prod/netflux/db/customer

- We can view what it stores

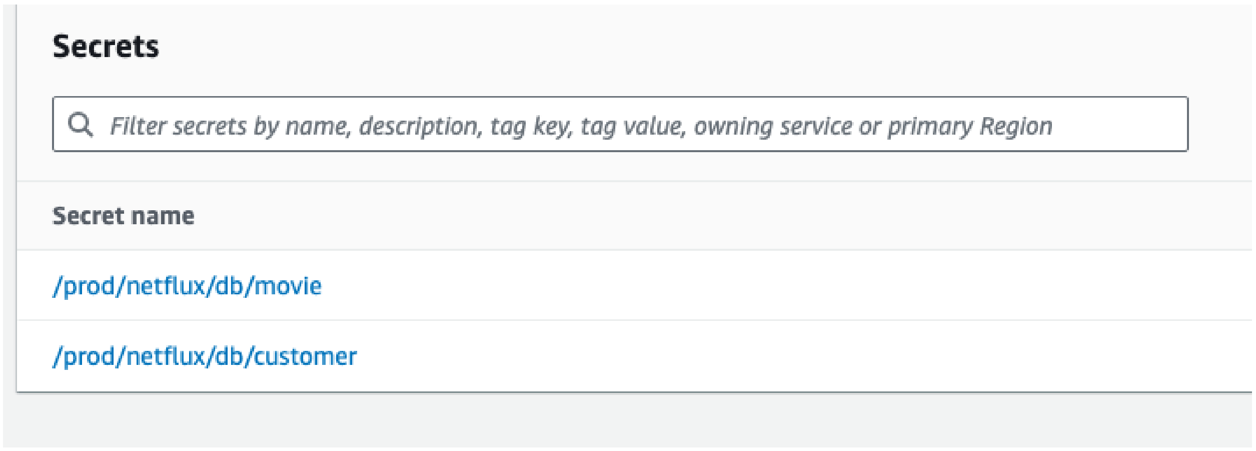


Secret value Info

Retrieve and view the secret value.

Secret key	Secret value
username	customer_user
password	customer_password_123
engine	postgres
host	netflux-db.cr6ukiceic0o.us-east-1.rds.amazonaws.com
port	5432
dbInstanceIdentifier	netflux-db

- repeat the above steps for “movie” db



Secrets

Filter secrets by name, description, tag key, tag value, owning service or primary Region

Secret name
/prod/netflux/db/movie
/prod/netflux/db/customer