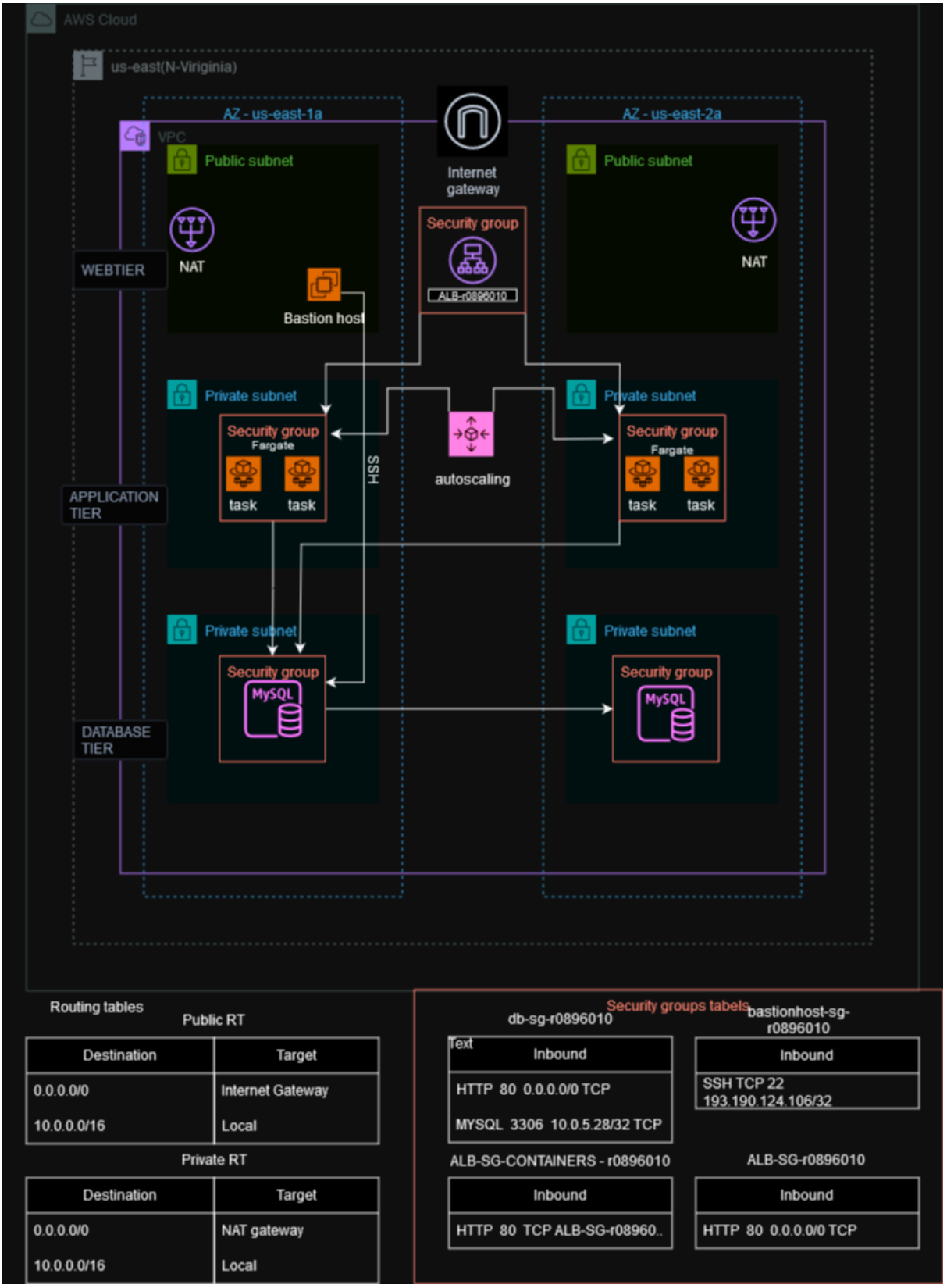
Multi-tier-architecture

Ward Boeckx - R0896010

Architectuur diagram



*all de SG's hebben Outbound rules: **All tcp 0.0.0.0/0**

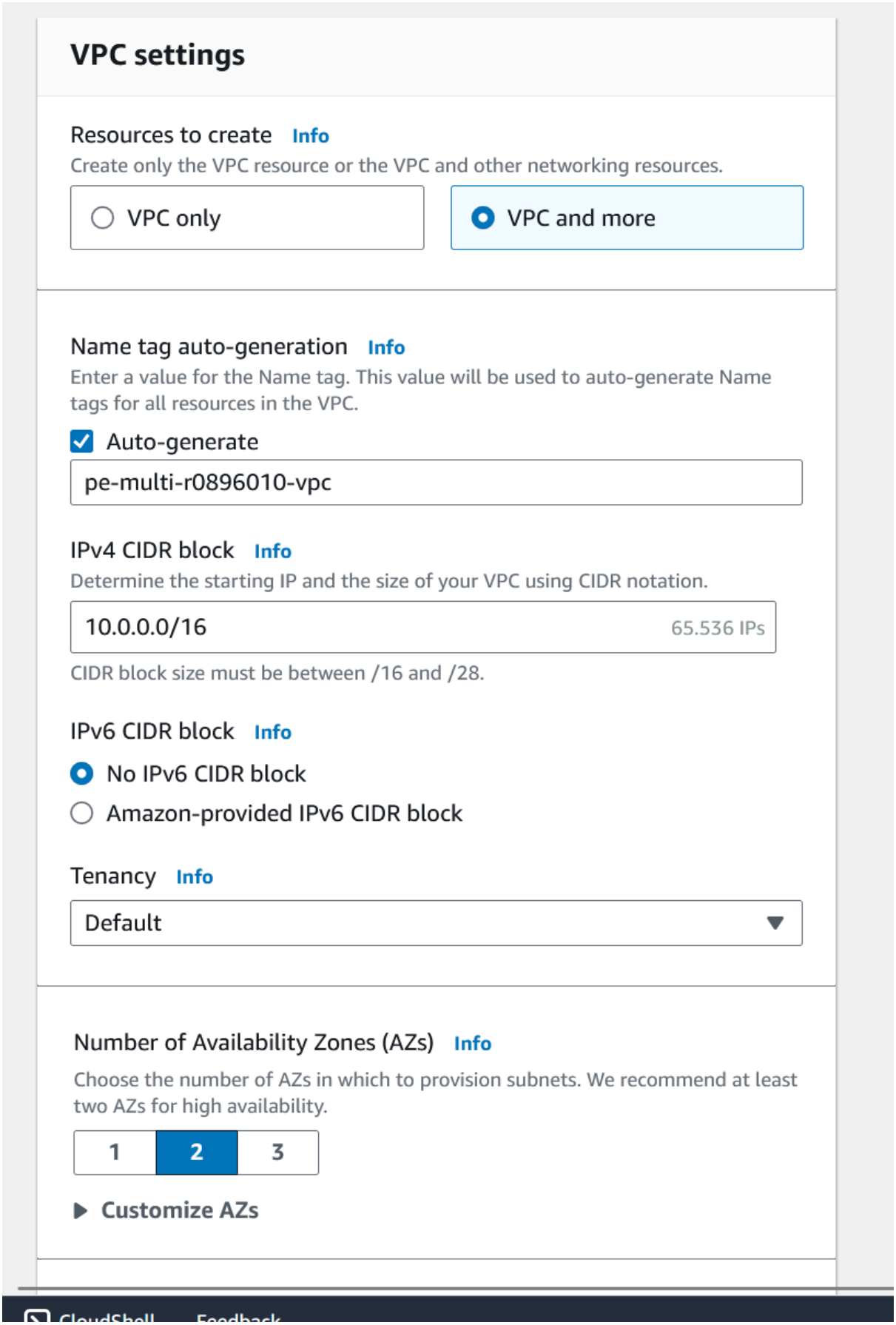
Netwerk gedeelte instellen (VPC,Subnets,RT’s)

We beginnen met het maken van onze VPC. Vervolgens maken we subnets aan en routing tables, ook maken we de NAT en internet gateways aan.

Het maken van onze VPC.

We kiezen de service “**VPC**” en we kiezen voor de volgende opties:

- VPC naam: **pe-multi-r0896010-vpc**
- IPv4 CIDR block: **10.0.0.0/16**
- IPv6 CIDR block: **None**
- Number of Availability Zones (AZs): **2**
- Number of public subnets: **2**
- Number of private subnets: **4**
- NAT gateways: **In 1 AZ**
- DNS options: **Enable DNS hostnames, Enable DNS resolution**


VPC settings
Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
VPC only
VPC and more
Name tag auto-generation Info
Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.
Auto-generate
pe-multi-r0896010-vpc
IPv4 CIDR block Info
Determine the starting IP and the size of your VPC using CIDR notation.
10.0.0.0/16
65.536 IPs
CIDR block size must be between /16 and /28.
IPv6 CIDR block Info
No IPv6 CIDR block
Amazon-provided IPv6 CIDR block
Tenancy Info
Default
Number of Availability Zones (AZs) Info
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.
1
2
3
Customize AZs

Number of public subnets Info

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 | 2

Number of private subnets Info

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 | 2 | 4

▶ Customize subnets CIDR blocks

NAT gateways ($) Info

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None | In 1 AZ | 1 per AZ

VPC endpoints Info

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None | S3 Gateway

DNS options Info

- [x] Enable DNS hostnames
- [x] Enable DNS resolution

▶ Additional tags

Cancel | Create VPC

Subnets

Het maken van de subnets gaat redelijk makkelijk, hierboven hebben we gezegd hoeveel wij er willen en door de **"VPC and more"** functie kunnen we gewoon de aantallen geven en private of public en dan maakt hij die aan. Ook de **Internet Gateway** en **NAT-gateway** worden automatisch aangemaakt.

Lijst van **subnets, gateway(s) en internet gateway(s)**

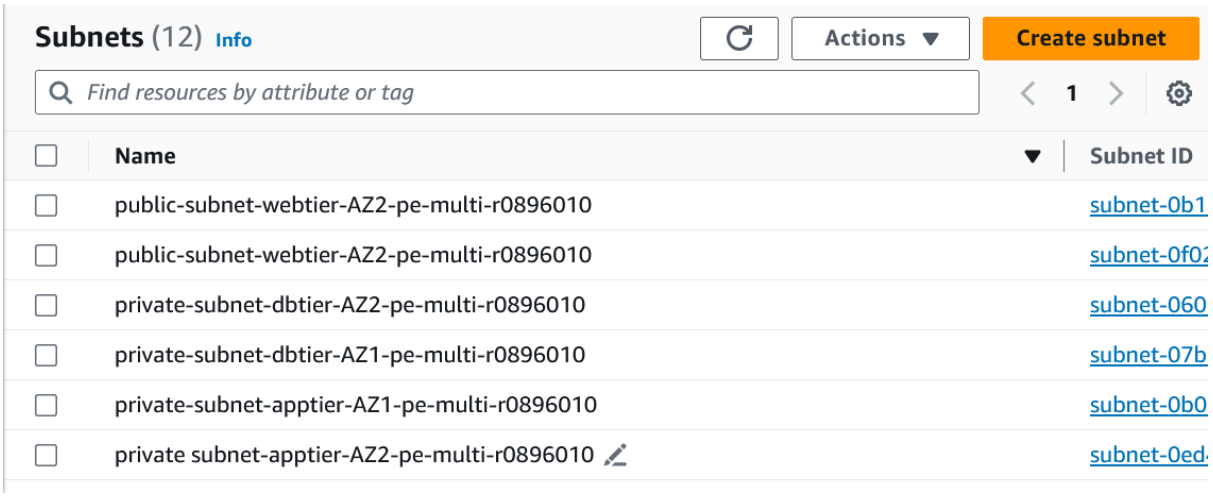
Public subnets

- public-subnet-**webtier**-AZ1-r0896010
- public-subnet-**webtier**-AZ2-r0896010

Private subnets

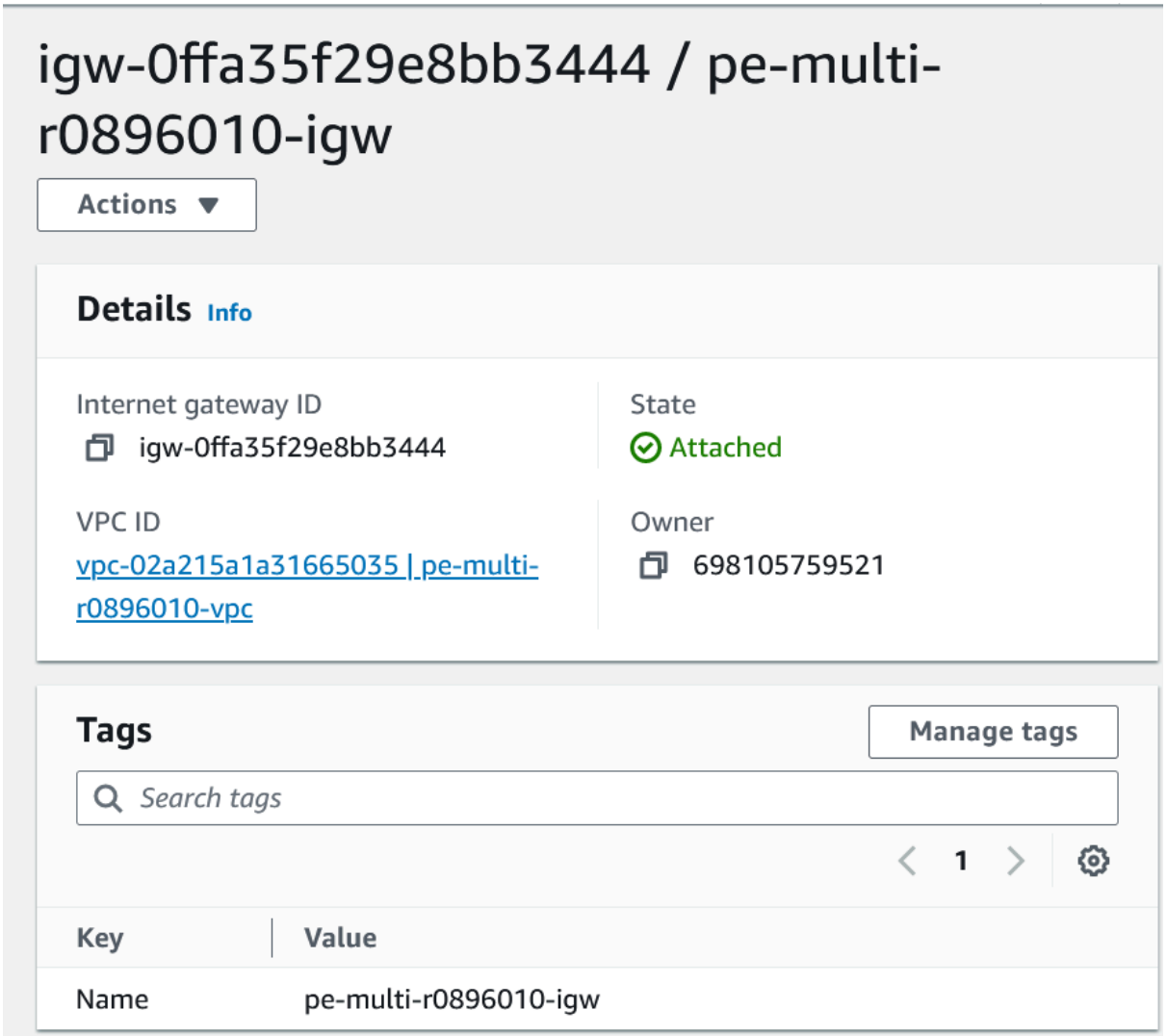
- private-subnet-**dbtier**-AZ2-pe-multi-r0896010
- private-subnet-**dbtier**-AZ1-pe-multi-r0896010
- private-subnet-**apptier**-AZ1-pe-multi-r0896010
- private subnet-**apptier**-AZ2-pe-multi-r0896010

*Je kan ook gewoon **subnets** 1 voor 1 aanmaken en het ip per subnet kiezen bv 10.0.1.0/24 voor een subnet en het volgende subnet 10.0.2.0/24. Indien je dat op deze manier doet heb je meer controle maar moet je ook bv de **internet gateway** vasthangen, zelfde voor **nat gateway.**



Internet gateway(s)

- pe-multi-r0896010-igw



NAT gateway(s)

- pe-multi-r0896010-nat-public1-us-east-1a

nat-046fd6e445d6ed519 / pe-multi-r0896010-nat-public1-us-east-1a

Actions

Details

NAT gateway ID
nat-046fd6e445d6ed519

NAT gateway ARN
arn:aws:ec2:us-east-1:698105759521:natgateway/nat-046fd6e445d6ed519

VPC
vpc-02a215a1a31665035 / pe-multi-r0896010-vpc

State
Available

Primary private IPv4 address
10.0.6.236

Created
Thursday, 11 January 2024 at 20:24:18 CET

Connectivity type
Public

Primary public IPv4 address
3.92.156.205

Subnet
subnet-0b13baccbd1536677 / public-subnet-webtier-AZ2-pe-multi-r0896010

State message Info
–

Primary network interface ID
eni-0bea44b191f27f7e3

Deleted
–

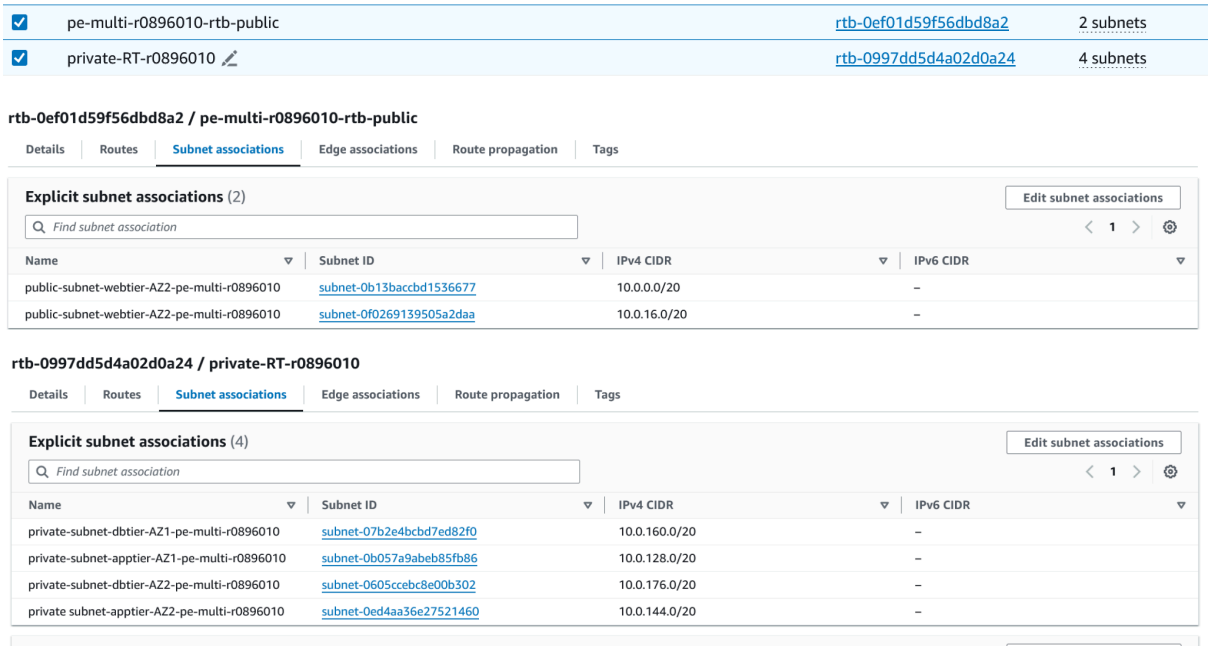
Routing tables

Ik gebruik voor deze PE opdracht 2 routing tables(soms in dit document ook afgekort als **RT**).

Routing table + subnet associations:

- private-RT-r0896010
    - private-subnet-dbtier-AZ1-pe-multi-r0896010
    - private-subnet-apptier-AZ1-pe-multi-r0896010
    - private-subnet-dbtier-AZ2-pe-multi-r0896010
    - private subnet-apptier-AZ2-pe-multi-r0896010

- pe-multi-r0896010-rtb-public
    - public-subnet-webtier-AZ2-pe-multi-r0896010
    - public-subnet-webtier-AZ2-pe-multi-r0896010
    -



☑	pe-multi-r0896010-rtb-public	rtb-0ef01d59f56dbd8a2	2 subnets
☑	private-RT-r0896010	rtb-0997dd5d4a02d0a24	4 subnets

rtb-0ef01d59f56dbd8a2 / pe-multi-r0896010-rtb-public

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Explicit subnet associations (2) — Edit subnet associations

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
public-subnet-webtier-AZ2-pe-multi-r0896010	subnet-0b13baccbd1536677	10.0.0.0/20	–
public-subnet-webtier-AZ2-pe-multi-r0896010	subnet-0f0269139505a2daa	10.0.16.0/20	–

rtb-0997dd5d4a02d0a24 / private-RT-r0896010

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Explicit subnet associations (4) — Edit subnet associations

Find subnet association

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
private-subnet-dbtier-AZ1-pe-multi-r0896010	subnet-07b2e4bcbd7ed82f0	10.0.160.0/20	–
private-subnet-apptier-AZ1-pe-multi-r0896010	subnet-0b057a9abeb85fb86	10.0.128.0/20	–
private-subnet-dbtier-AZ2-pe-multi-r0896010	subnet-0605ccebc8e00b302	10.0.176.0/20	–
private subnet-apptier-AZ2-pe-multi-r0896010	subnet-0ed4aa36e27521460	10.0.144.0/20	–

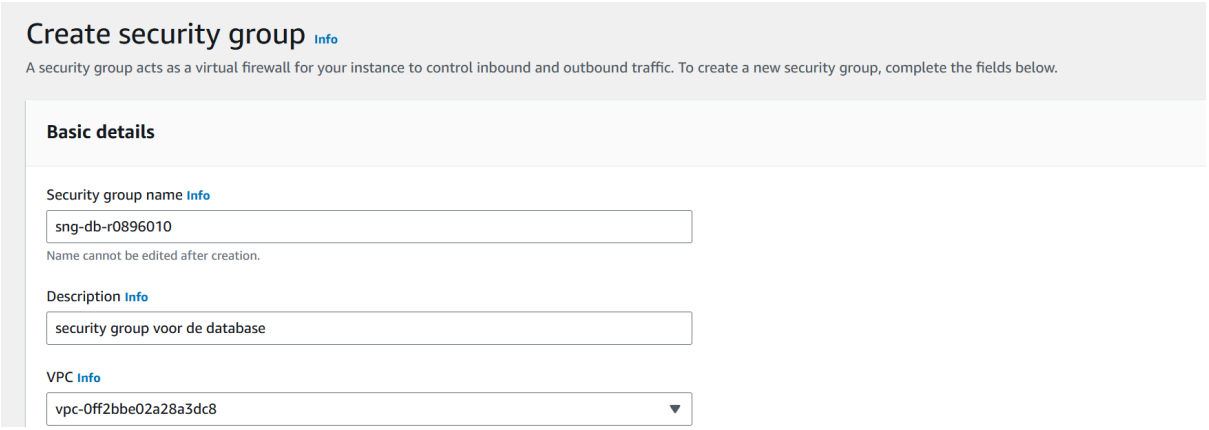
Belangrijk is dat je de juiste subnetten effectief toevoegt aan de routing tables. Dit doe je onder de service **VPC** vervolgens uit het menu “**Routing tables”** kiezen dan de **RT** aanvinken waar je subnets wilt met associëren. Dan kies tabblad “**Subnet associations”** Kijk screenshots hier net boven.

Database(RDS)

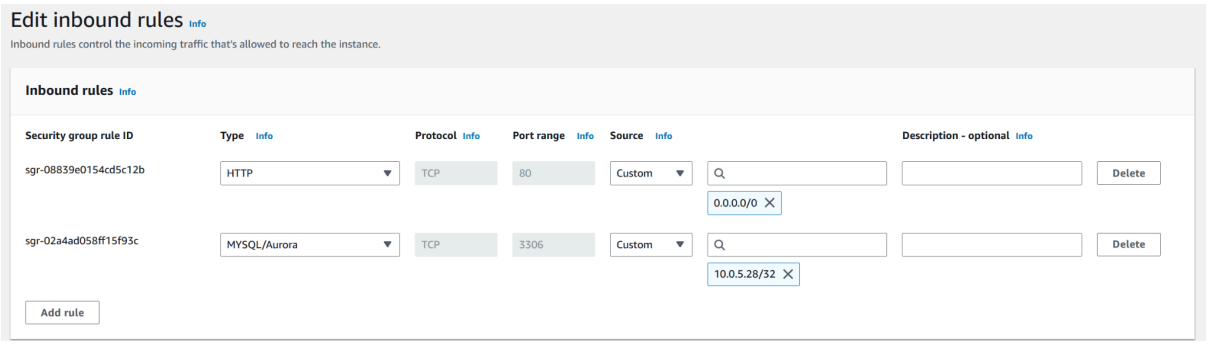
Vervolgens gaan we onze **database(RDS)** aanmaken. Voor deze database kiezen we voor MYSQL.

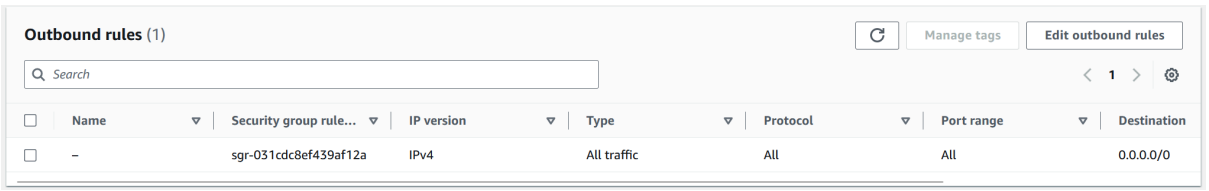
Aanmaken Security Group Database

Voor we de database gaan aanmaken gaan we de Security group al aanmaken voor de database. We kiezen de service “**EC2**”. In het menu links kiezen we “**Subnet groups”** dan klikken we op de oranje knop -> **”Create security group”** en dan volgende instellingen:



We geven ook volgende rules mee(ook terug te vinden op de diagram):

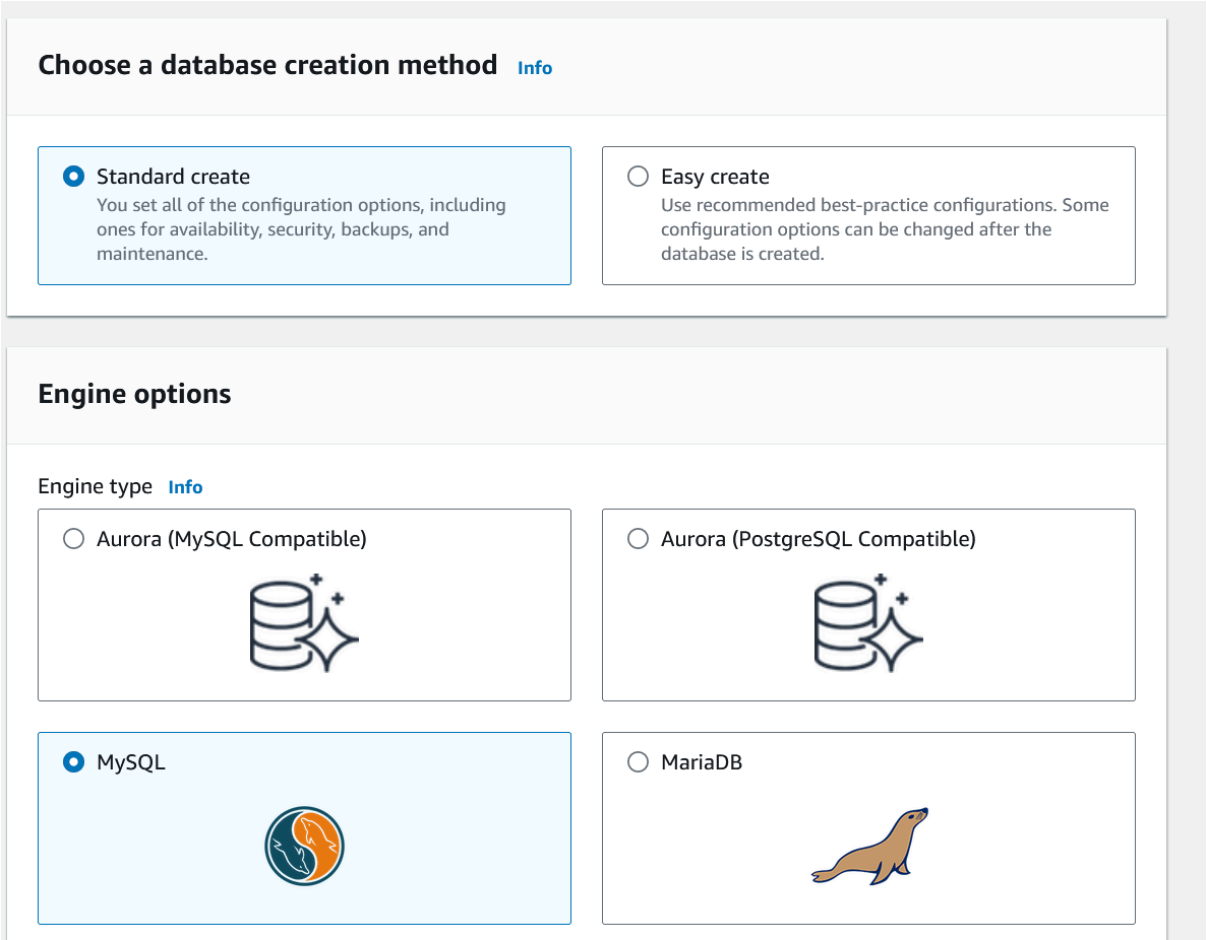




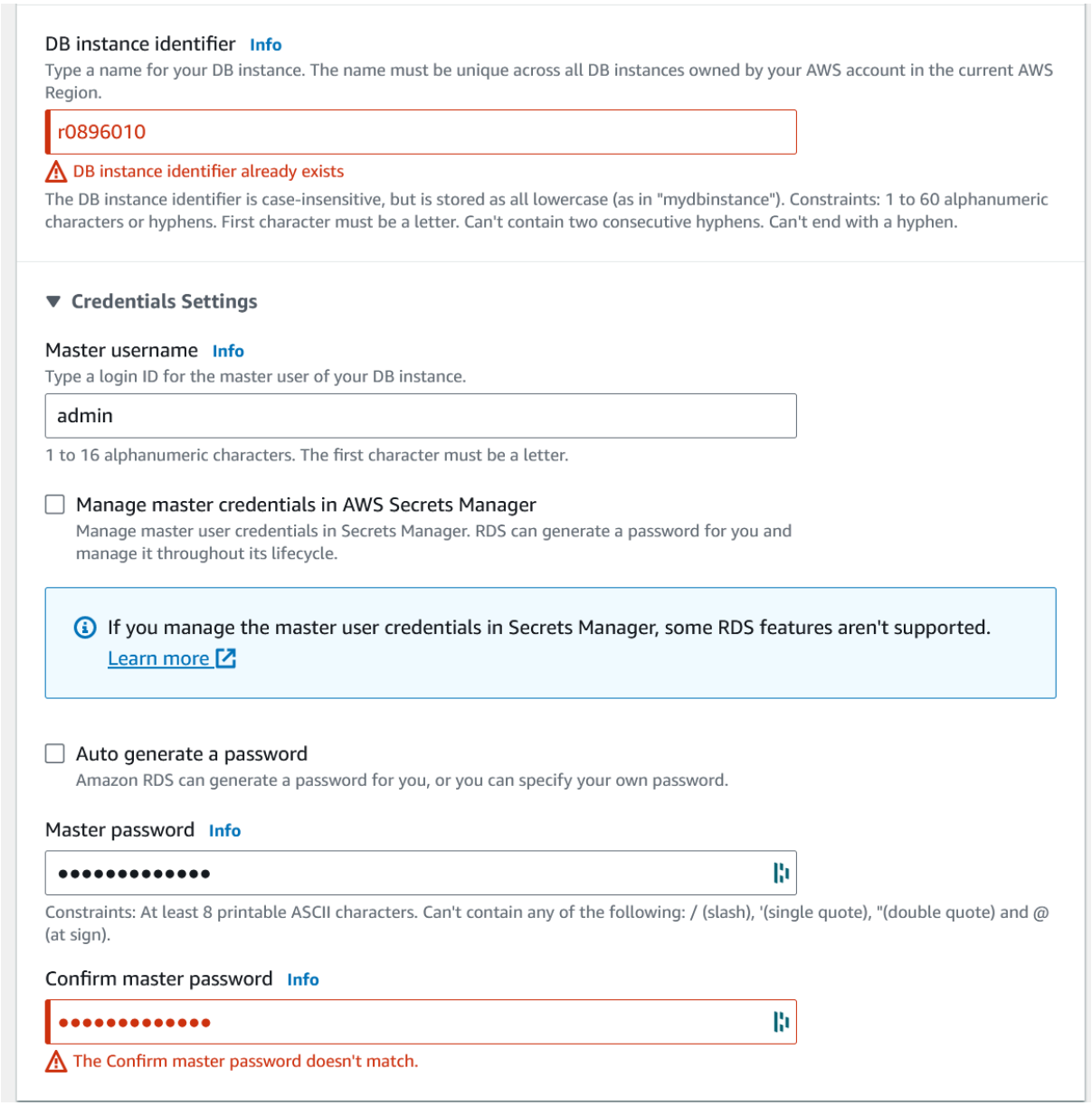
Aanmaken Database

We openen de **RDS** service en kiezen “**Create Database", vervolgens** kiezen we de volgende opties.

“**Standard create**” -> “**MySQL**” -> “**Free tier**”



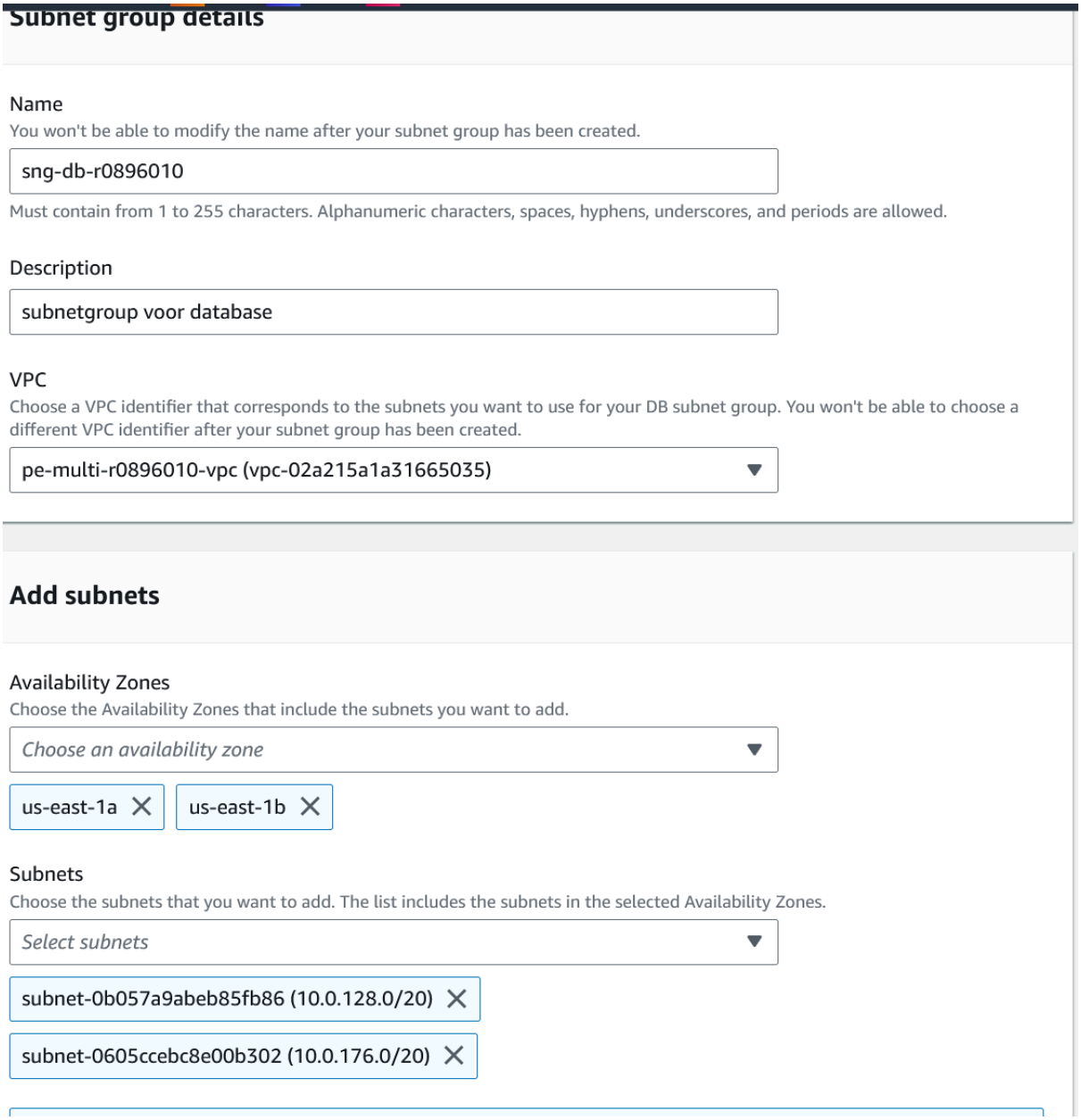
de DB Instance identifier -> “**r0896010**” -> Master username(belangrijk deze hebben we nog nodig voor de DB te connecteren in samenhang met onze crudapp) -> “**admin”** -> Master password -> “**Administrator**”



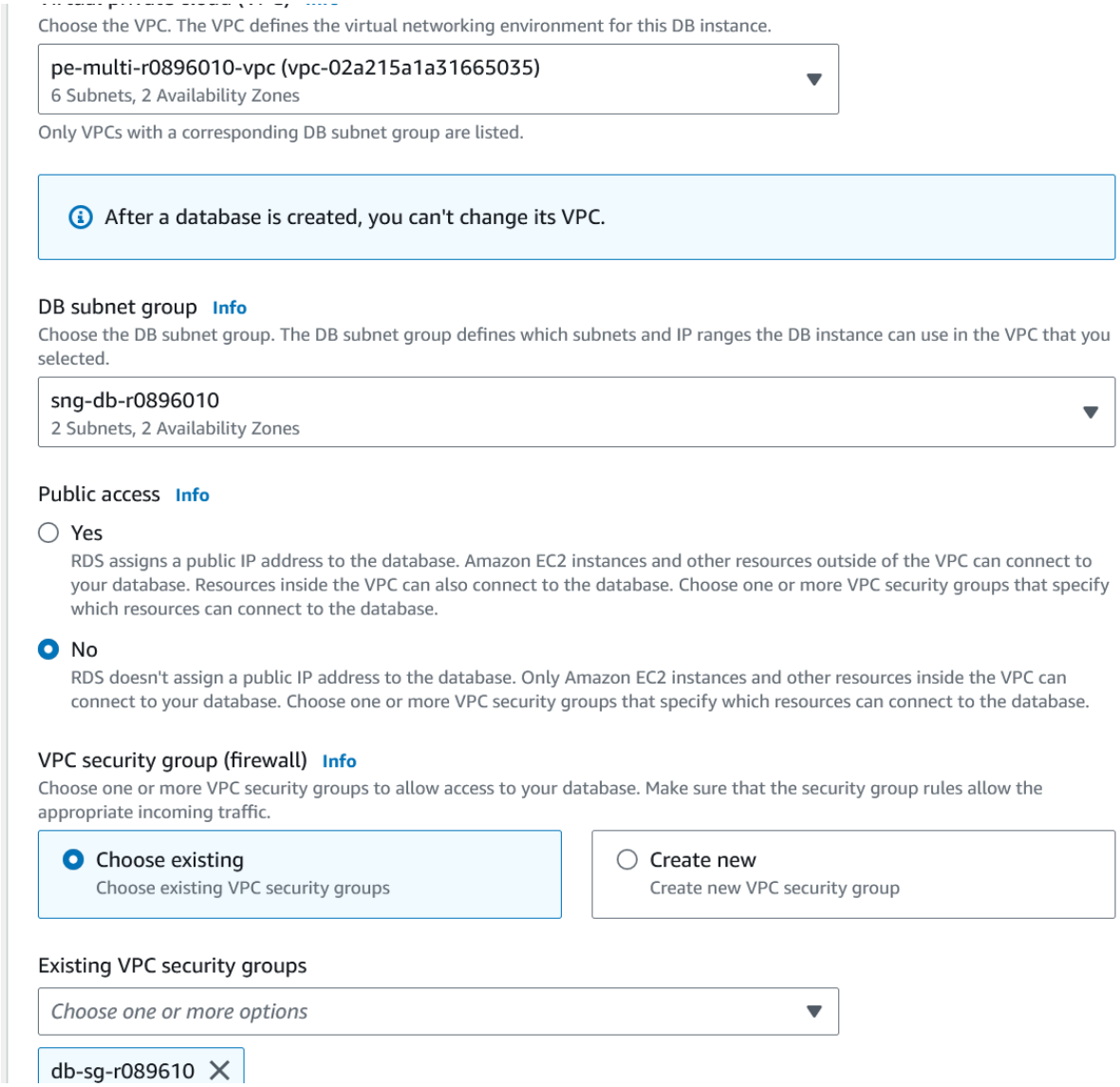
Instance config. en **Storage** heb ik gelaten hoe het was. Vervolgens onder de noemer **Connectivity** kies ik voor mijn **VPC** -> **“pe-multi-r0896010-vpc”** vervolgen kiezen wij de subnet group "**default-vpc-02a2**…" echter is dit niet de best practice werkwijzen, kijk “*”. Vervolgens kiezen we nog onze **Security group** die we hebben gemaakt voor onze DB -> “**db-sg-r0896010**” Vervolgens maken we de database aan.

*onder **DB subnet group** kies je de subnet group die de database kan kiezen om te implementeren, in mijn opdracht heb ik dit niet toegepast omdat ik hier achteraf na vragen aan leerkracht ben achtergekomen. Wel geef ik even het stappenplan mee hoe dit kan.

Kies in het menu “**Subnet groups”** vervolgens “**Create Subnet groups”** en vervolgens geef ik volgende instellingen mee:

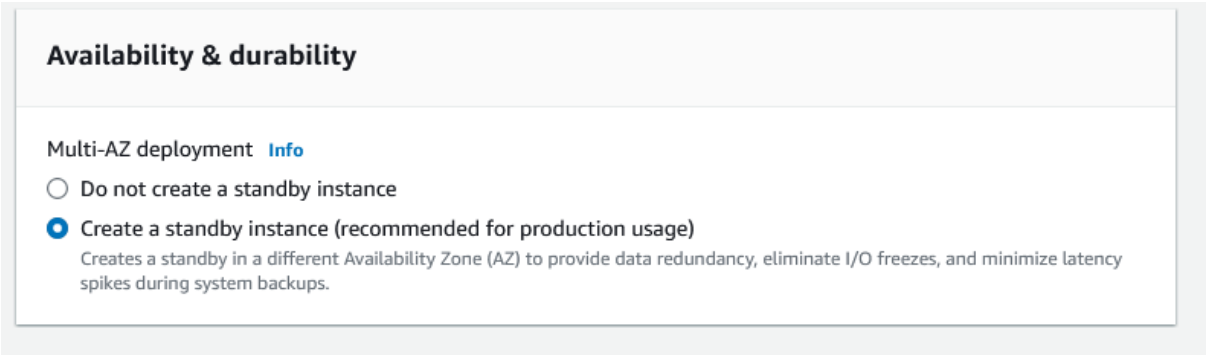


Als we dan de database aanmaken en we kijken in de lijst **subnet groups** dan kunnen we kiezen voor de subnet groep “**sng-db-r0896010”** Dan kan de database maar kiezen uit de 2 private subnets die we hebben aangemaakt specifiek voor de database (Kijk deel **subnets**)



*Mijn database naam bestaat al omdat ik deze al had opgezet.

Belangrijk is ook om een database standby klaar te hebben staan voor data redundancy



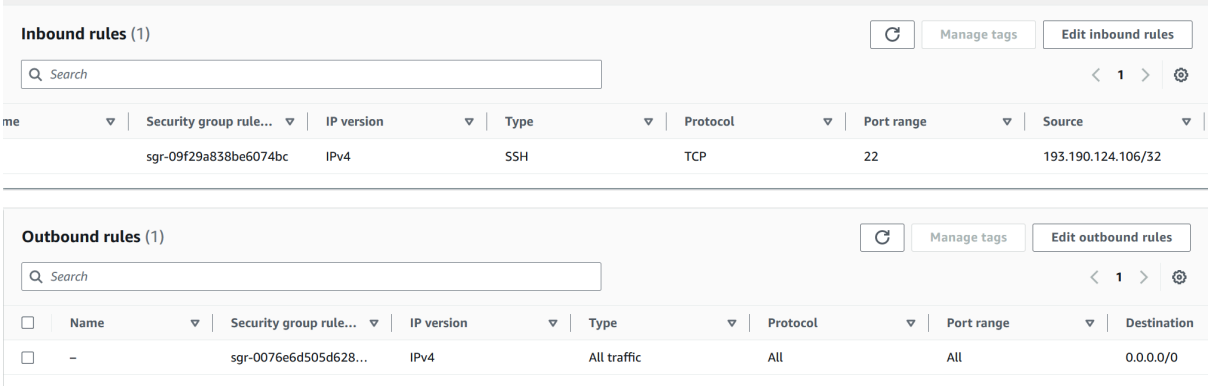
Dus we kiezen ook voor “**Create a standby instance**”

Bastionhost/jump server.

We gaan een EC2 bastion host maken die we in het public subnet gaan zetten zodanig wij kunnen ssh'en naar de **Bastion hos**t vanaf onze **ubuntu vm** of gewoon je computer, ik gebruik in dit geval een Linux ubuntu VM.

Aanmaken van de Security Group voor de Bastion host.

Voor we de EC2 instance gaan opzetten moeten we eerst een security group maken. Dit doen we weer door links in het menu de optie **"Security groups"** te selecteren, we doen dit op dezelfde manier als bij de database, de naam van mijn bastion host security group is "**bastion-host-sg-r0896010**" we kiezen voor volgende inbound rules, hierdoor kunnen we SSH'en naar de EC2 bastion host server, ik gebruik ook als source het **IP van mijn Ubuntu VM**. Hierdoor kan ik enkel vanaf mijn ubuntu vm SSH'en naar de bastion host.:



Aanmaken van de EC2 voor onze Bastion host

We gaan naar de service **"EC2"** en kiezen **"Launch instance"** Met volgende opties:

- Name: **Bastion-host-r0896010**
- Os image: **Ubuntu**
- Keypair: **Nginx** (*Deze key pair had ik al eens gemaakt bij het maken van een nginx, ik heb in deze PE opdracht die keypair blijven gebruiken).
- Network settings (kijk screenshot)

Name and tags Info

Name

Bastion-host-r0896010

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible

ami-0c7217cdde317cfec (64-bit (x86)) / ami-05d47d29a4c2d19e1 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-12-07

Architecture AMI ID

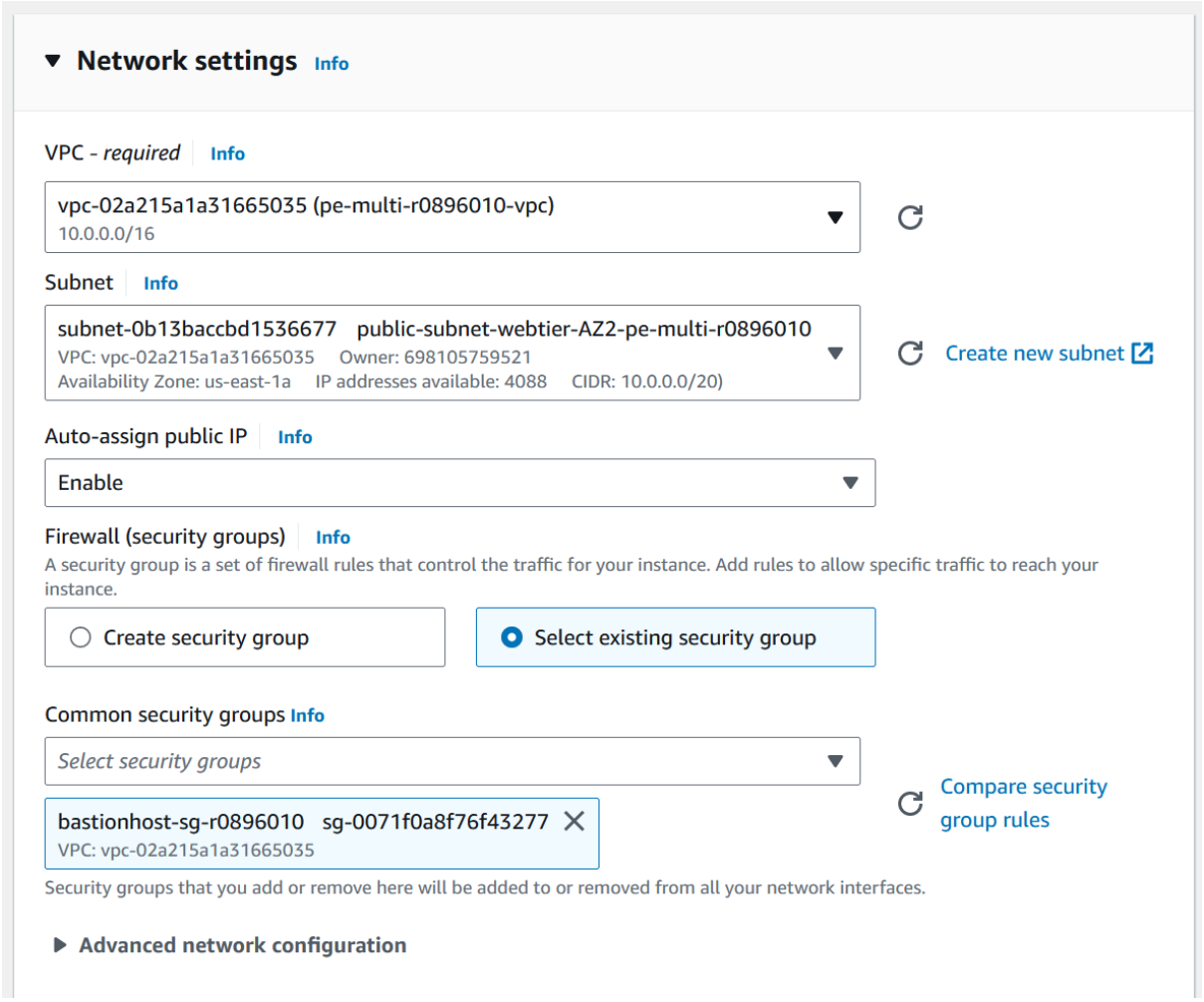
Key pair (login) Info

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

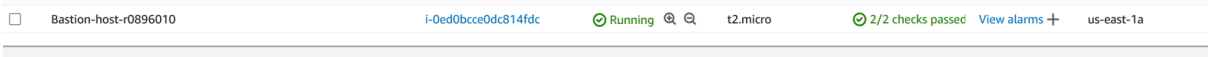
Key pair name - *required*

nginx

Create new key pair

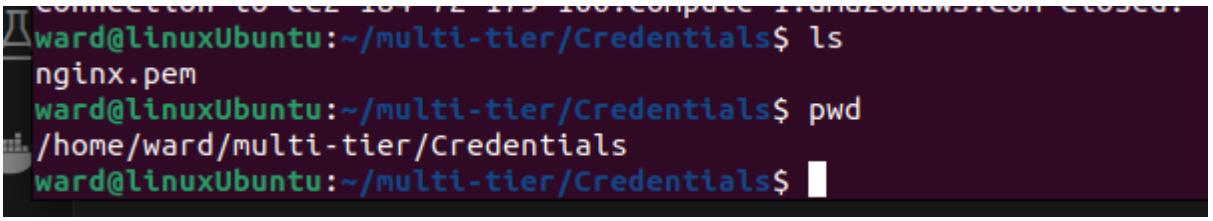


Dan kunnen we zien dat deze is opgedraaid.

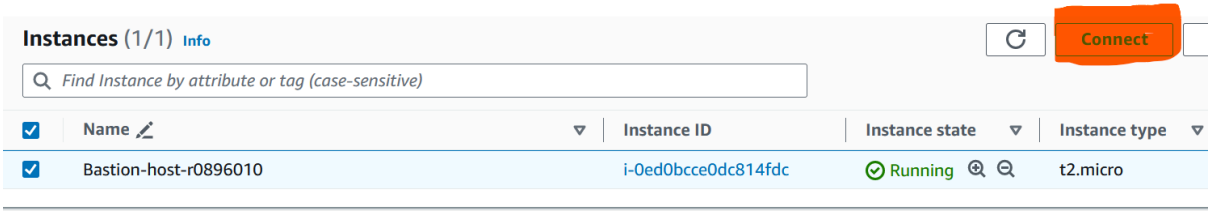


SSH testen richting de bastion host.

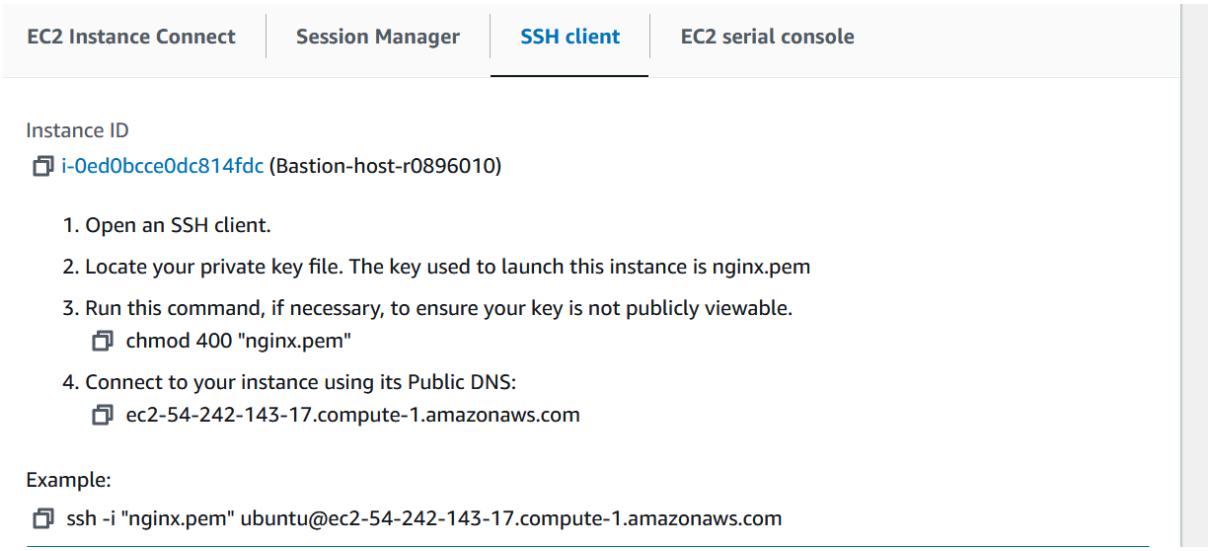
Nu gaan we testen of we kunnen ssh'en vanaf onze ubuntu(vm) naar onze Bastion host. Bij het aanmaken van onze bastion host kan je "**Create new keypair",** als je dit doet, download je een .pem file in mijn geval. Deze file heb ik op volgende locatie gezet: **home/ward/multi-tier/Credentials**



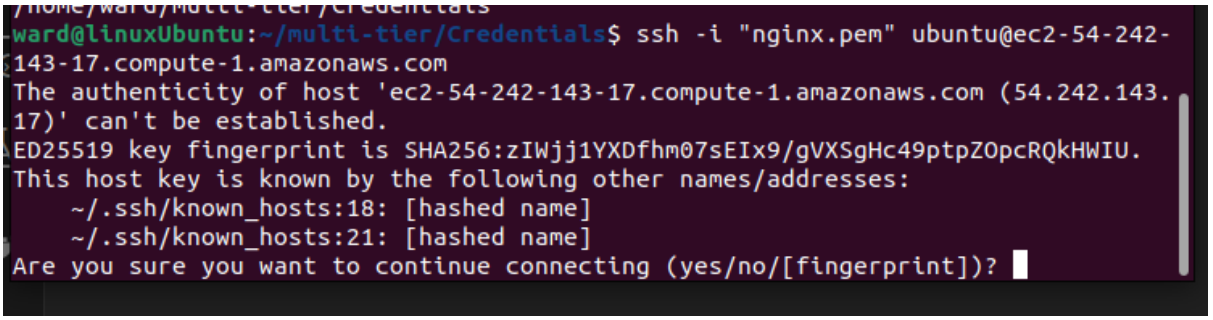
Als we dit hebben gedaan kunnen we simpelweg in de AWS console onze **bastion host** instance selecteren en vervolgens rechtsboven op de knop “**Connect”** te klikken



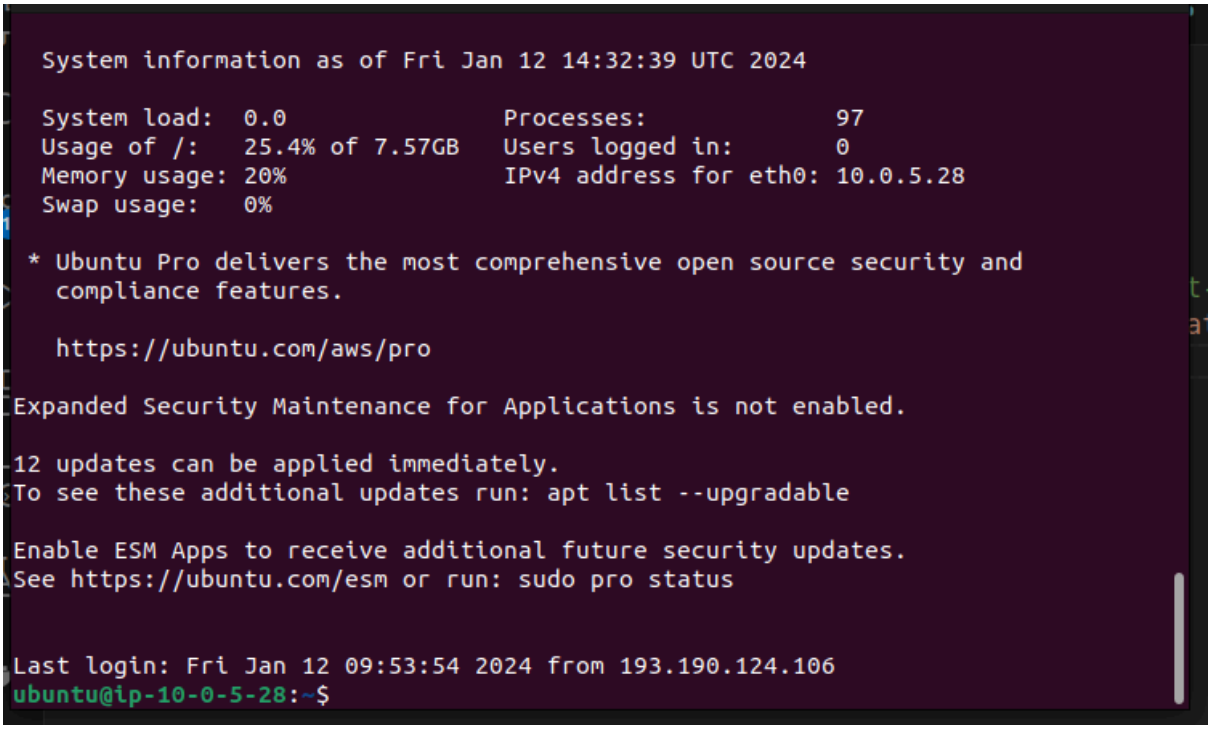
Dan volgen we simpelweg die stappen.



Ik heb de eerste 3 stappen al gedaan, vervolgens gebruik kopieer ik de code onder **Example**



We kiezen “**yes”.** Vervolgens zien we dat we geconnecteerd zijn met onze Bastion host.



Vervolgens volg ik de stappen van volgende website die we hebben meegekregen als nuttige links voor de opdracht. https://neil.tesaluna.com/posts/connecting-flask-to-aws-rds/

Once the EC2 instance is up and running. Access the instance through SSH or use Sessions Manager

One inside the instance, update and upgrade the package manager then create and switch to a **sudo** user

```
$ apt update && apt upgrade
$ useradd -ms /bin/bash newuser
$ usermod -aG sudo newuser
$ sudo su newuser
newuser >
```

copy

Find out which MySQL is available to install from the package manager and install your preference.

```
newuser > mysql --version
Command 'mysql' not found, but can be installed with:
sudo apt install mysql-client-core-8.0    # version 8.0.23-0ubuntu0.20.04.1, or
sudo apt install mariadb-client-core-10.3  # version 1:10.3.25-0ubuntu0.20.04.1
newuser > sudo apt install mysql-client-core-8.0
```

copy

Connecting to the database

Once install connect to the database with the following

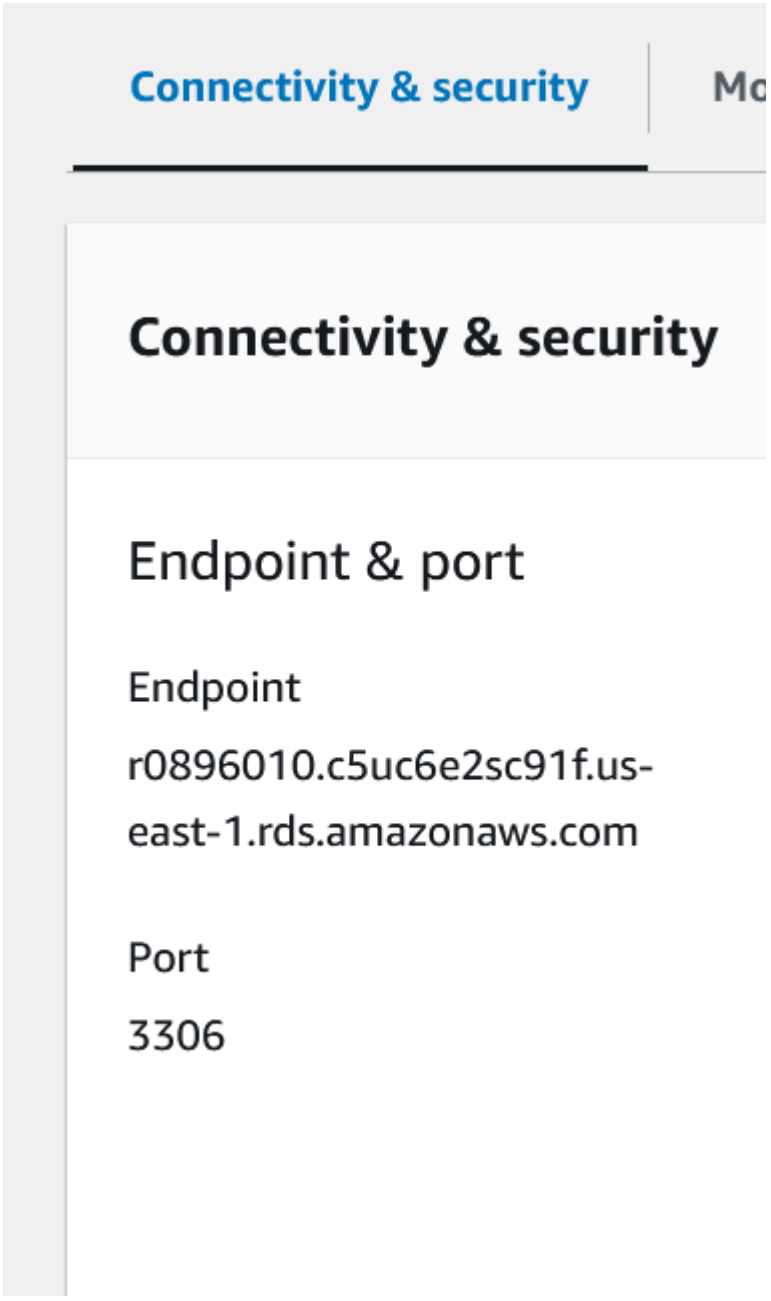
```
newuser > mysql -h ...ap-southeast-2.rds.amazonaws.com -P 3306 -u admin -p
```

copy

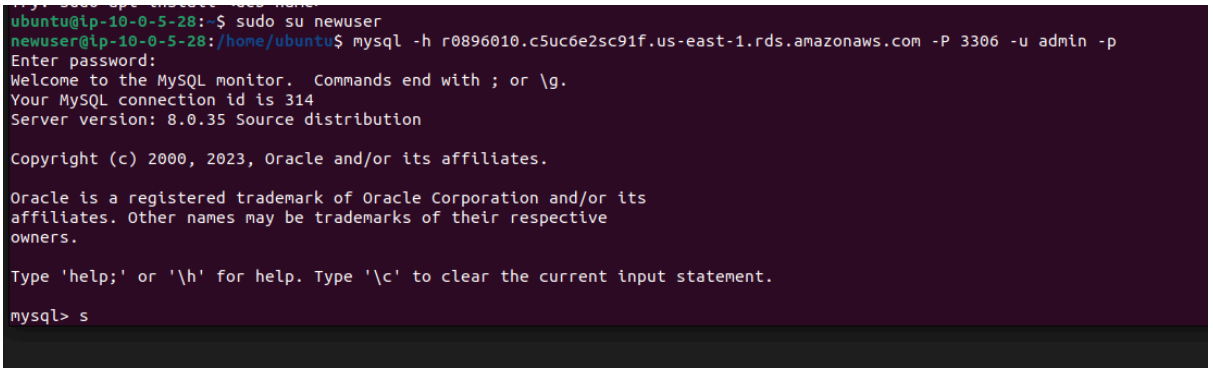
*Ik geef de screenshot hierboven mee omdat ik bovenstaande al had gedaan.

Connecteren vanaf de Bastion host naar de Database(RDS)

Als we zijn ingelogd als “newuser” gaan we connecteren naar de database(Mysql) door middel van de bovenstaande commando: **newuser > mysql -h ...ap-southeast-2.rds.amazonaws.com -P 3306 -u admin -p** -> we gaan natuurlijk wel het endpoint aanpassen naar onze database. Die endpoint is te vinden onder “**RDS**” -> Database selecteren, in mijn geval **“R0896010”** vervolgens kies je vanonder “**Connectivity & Security**” die endpoint kopiëren we.



Dus we geven dit commando in en kijken of we kunnen connecteren naar onze DB:



Volgende wat we doen is in mysql een database aanmaken. Dat kan met volgend commando “**CREATE DATABASE r0896010;**”. Vervolgens geef ik het cmd in “**SHOW databases**;” s
Hier zie je dat ik een database heb gemaakt genaamd “**r0896010”** Ik had mijn database “**r0896010”** al aangemaakt maar hieronder nog een screenshot waar ik een database “**voorbeeld”** aanmaak.

```
mysql> CREATE DATABASE voorbeeld;
Query OK, 1 row affected (0.04 sec)

mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| r0896010           |
| sys                |
| voorbeeld          |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

Aanpassen Crud app

We doen een git clone van de applicatie zoals op de vorige opdracht. Ik volg onderstaande stappen(dit had ik al wel gedaan, te vinden in het document vorige PE opdracht.

```
git clone https://github.com/gurkanakdeniz/example-flask-crud.git
```

```
cd example-flask-crud/
```

```
python3 -m venv venv
```

```
source venv/bin/activate
```

```
pip install --upgrade pip
```

```
pip install -r requirements.txt
```

We installeren ook “**Pymysql en Gunicorn”.** Vervolgens doen we wat pipeline-trickery, namelijk “**Pip freeze”-r** gevolgd door “**pip freeze > requirements.txt”** Dit zet eigenlijk de **Gunicorn** en **Pymysql** in de **requirements.txt**.

Aanpassingen code crud-app

We willen ervoor zorgen dat onze database in aws bereikbaar gaat zijn vanaf we de crud-app gaan draaien in AWS. Dit doen we door de **config.py**-file aan te passen.

```
import os
basedir = os.path.abspath(os.path.dirname(__file__))

class Config(object):
    SECRET_KEY = 'do-or-do-not-there-is-no-try'
#    SECRET_KEY = os.environ.get('SECRET_KEY') or 'do-or-do-not-there-is-no-try'
    SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://admin:administrator@r0896010.c5uc6e2sc91f.us-east-1.rds.amazonaws.com/r0896010'
    SQLALCHEMY_TRACK_MODIFICATIONS = False
```

```
mysql+pymysql://admin:administrator@r0896010.c5uc6e2sc91f.us-east-1.rds
.amazonaws.com/r0896010
```

Wat uitleg over de database uri: **admin:administrator** = de gebruikersnaam en wachtwoord die we hebben ingesteld bij het aanmaken van onze **RDS.** vervolgens hebben we de endpoint en achter de / nog eens de naam van de database, in mijn gevan “**R0896010”**

Let op: Bovenstaande code is niet de beste manier, omdat we het paswoord van de RDS in dit geval “**administrator**” gewoon in plain text meegeven. Beste is dit mee te geven en in een environment variabele te steken. Ik heb dit niet gedaan, omdat alles al draait tijdens het schrijven van dit document, maar wil toch meegeven dat ik hiervan op de hoogte ben.

Vervolgens passen we in de **__init__py** de file niet perse aan maar we voegen code toe. “**import pymysql”** en vervolgens ook van onder zet ik nog **db.create_all()**. We importeren hier pymysql zodat we dit kunnen gebruiken, Ook de **db.create_all()** gebruiken we zodat de

tabellen van de DB automatisch worden aangemaakt.

example-flask-crud > app > __init__.py > ...

```python
from flask import Flask
from app.config import Config
from flask_sqlalchemy import SQLAlchemy
from flask_migrate import Migrate
import pymysql
app = Flask(__name__)
app.config.from_object(Config)
db = SQLAlchemy(app)
migrate = Migrate(app, db)

from app import routes, models
db.create_all()
```

Instellen Gunicorn(WSGI)

Ik had dit moeilijker verwacht, in principe is het gewoon de Gunicorn installeren dit hadden we al gedaan -> "**pip install gunicorn**". Vervolgens heb ik een py-file aangemaakt namelijk "**gunicorn_config.py**" Ik vermeld volgende in de py-file.

example-flask-crud > gunicorn_config.py > ...

```python
bind = '0.0.0.0:80'
workers = 4
```

Simpele configuratie.

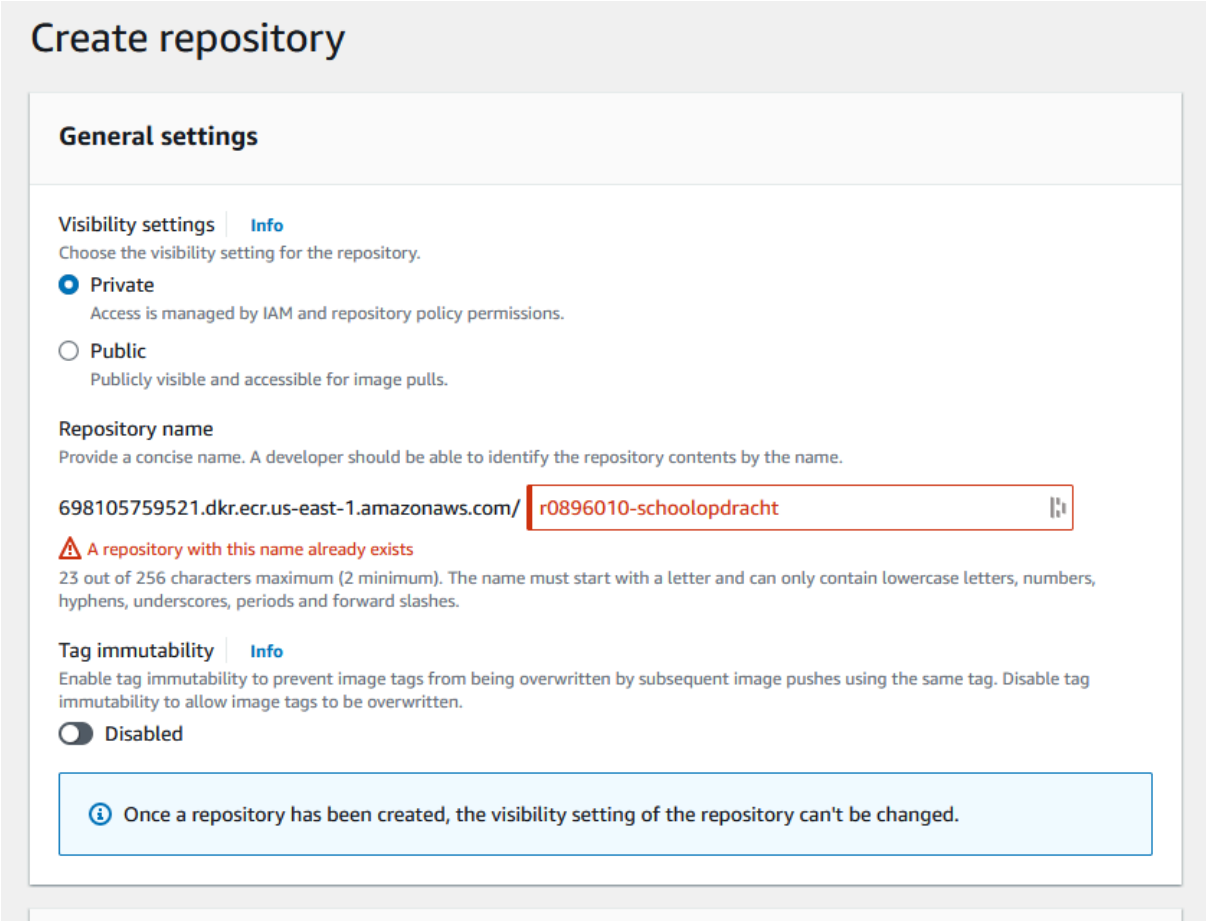
Aanmaken van de Dockerfile.

Vervolgens maken we een Dockerfile aan, ik geef volgende mee.

Ik wil laten weten dat volgens mij de ENV FLASK_APP=crudapp.py niet noodzakelijk is. Ik heb geen tijd meer voor dit nog te testen maar wil dit ook even meegeven.

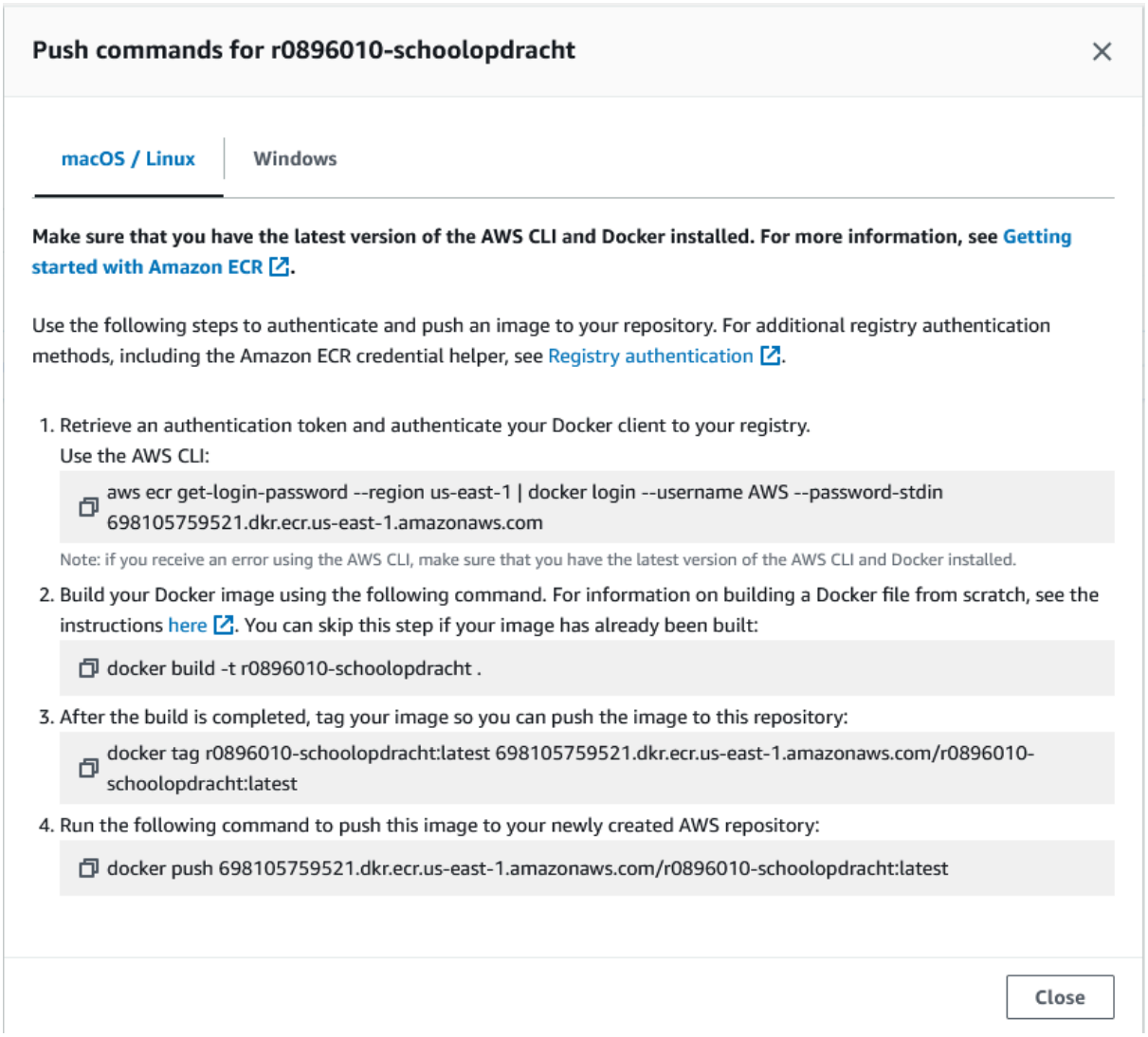
Aanmaken van ECR-repository en pushen van Docker-image

Rechtsboven klikken we op “**Create repository.** Ik kies voor een private repo. Dan geef ik het een naam en vervolgens klik ik op “**Create repository”**



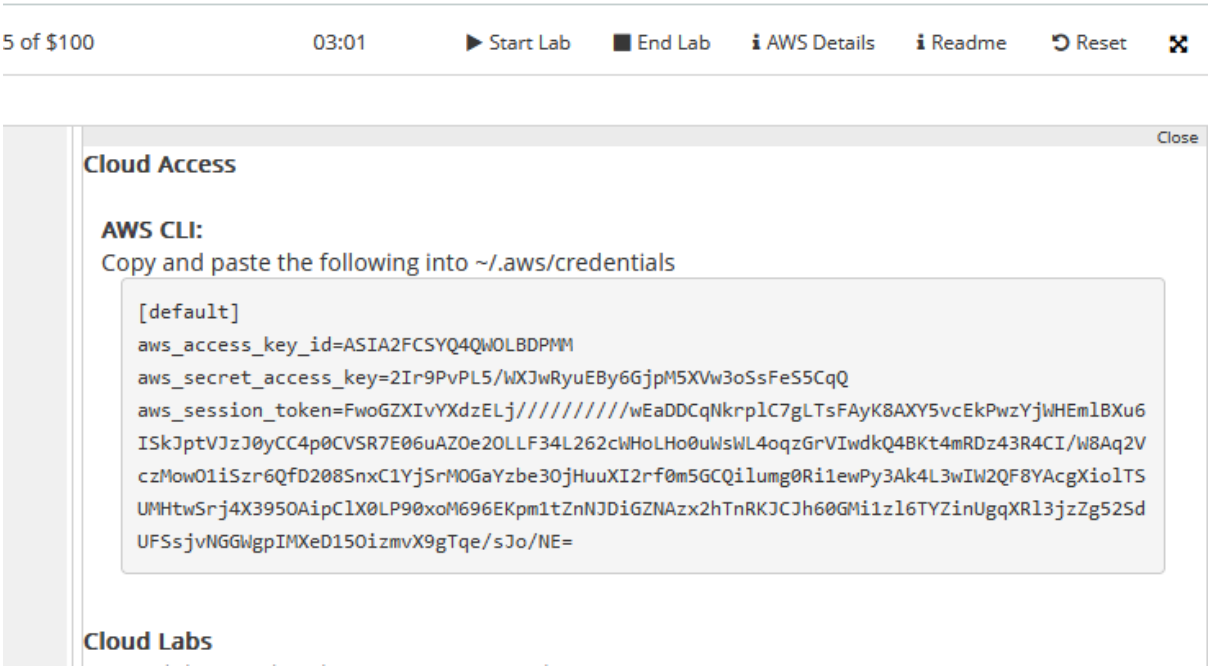
Vervolgens selecteren wij onze repo en klikken rechtsboven op **“View push commands”**

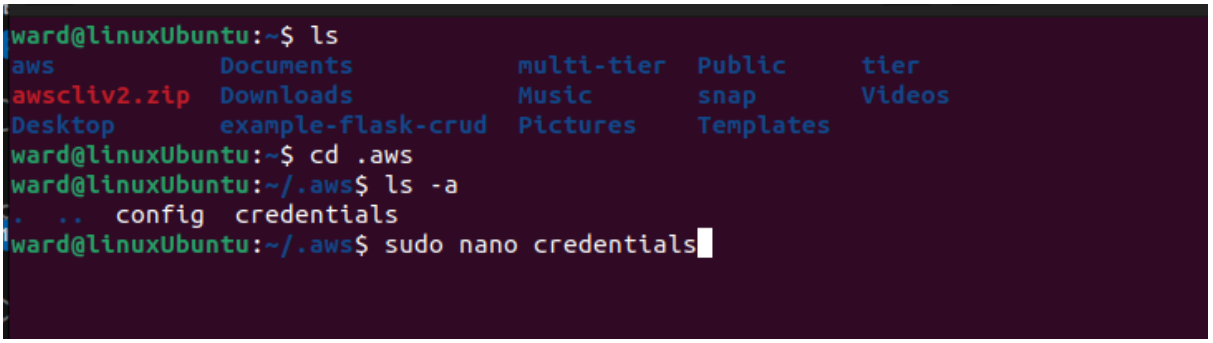
Dan krijgen we een kadertje met de commando's die we kunnen gebruiken vanaf we zijn ingelogt op de AWS cli.

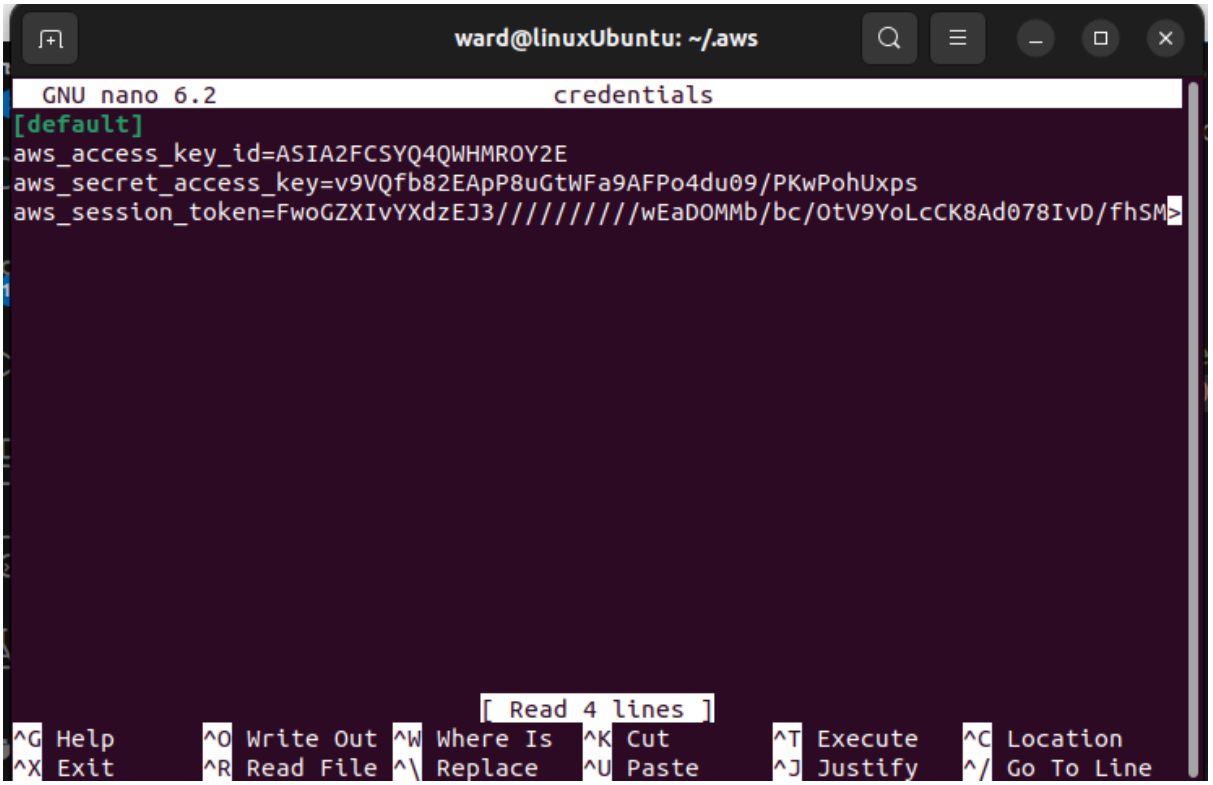


Inloggen op AWS CLI en bouwen van docker image.

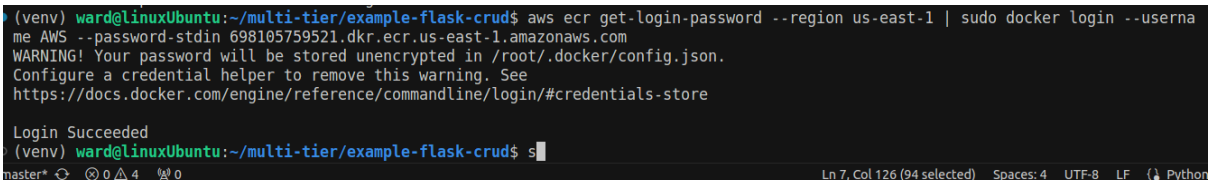
Vervolgens gaan we naar AWS academy learning lab en klikken op AWS Details. Bij **Cloud access -> AWS CLI** Kopiëren en plakken wij de volgende credentials op onze ubuntu vm in **.aws/credentials**


5 of $100
03:01
Start Lab
End Lab
AWS Details
Readme
Reset
Close
Cloud Access
AWS CLI:
Copy and paste the following into ~/.aws/credentials
[default]
aws_access_key_id=ASIA2FCSYQ4QWOLBDPMM
aws_secret_access_key=2Ir9PvPL5/WXJwRyuEBy6GjpM5XVw3oSsFeS5CqQ
aws_session_token=FwoGZXIvYXdzELj//////////wEaDDCqNkrplC7gLTsFAyK8AXY5vcEkPwzYjWHEmlBXu6
ISkJptVJzJ0yCC4p0CVSR7E06uAZOe2OLLF34L262cWHoLHo0uWsWL4oqzGrVIwdkQ4BKt4mRDz43R4CI/W8Aq2V
czMowO1iSzr6QfD208SnxC1YjSrMOGaYzbe3OjHuuXI2rf0m5GCQilumg0Ri1ewPy3Ak4L3wIW2QF8YAcgXiolTS
UMHtwSrj4X395OAipClX0LP90xoM696EKpm1tZnNJDiGZNAzx2hTnRKJCJh60GMi1zl6TYZinUgqXRl3jzZg52Sd
UFSsjvNGGWgpIMXeD15OizmvX9gTqe/sJo/NE=
Cloud Labs



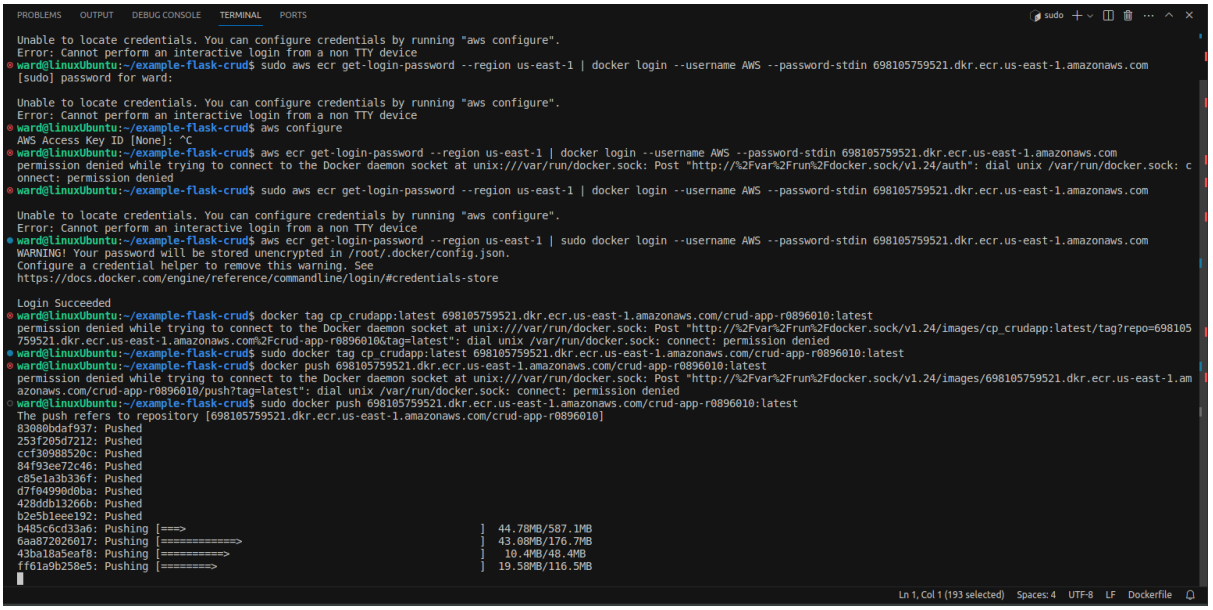


Vervolgens loggen we in op de **AWS cli**

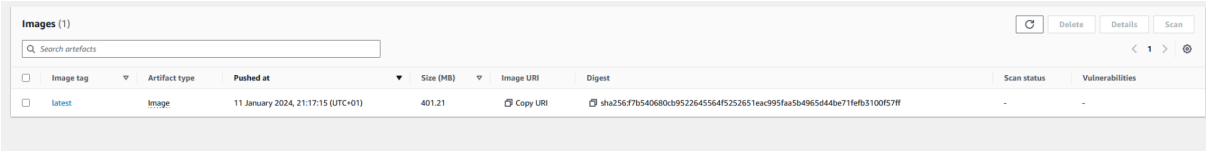


En volgen we de commandos voor de docker container te pushen naar onze ECR

Ik gebruik even de screenshots van mijn vorige document, dit is letterlijk hetzelfde van stappen dat je moet doen om de container te bouwen en naar de ECR te pushen.



Als we dan in onze repo gaan kijken zien we dat de image gepushed is.



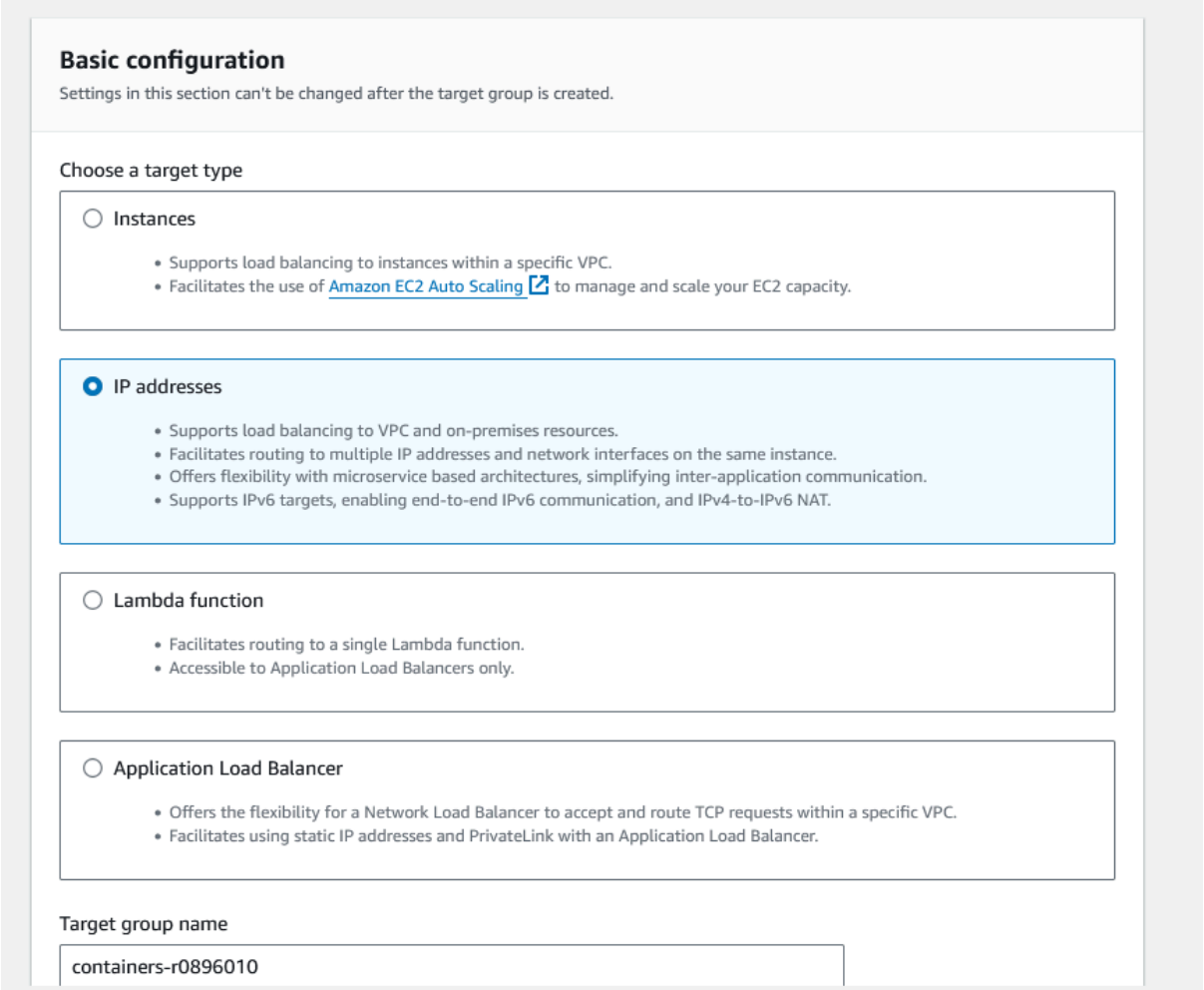
Aanmaken van Application Load Balancer

Ik heb gekozen om een **applicatie loadbalancer** te gebruiken. Voor het verkeer te sturen en verdelen over mijn verschillende **fargate tasks**

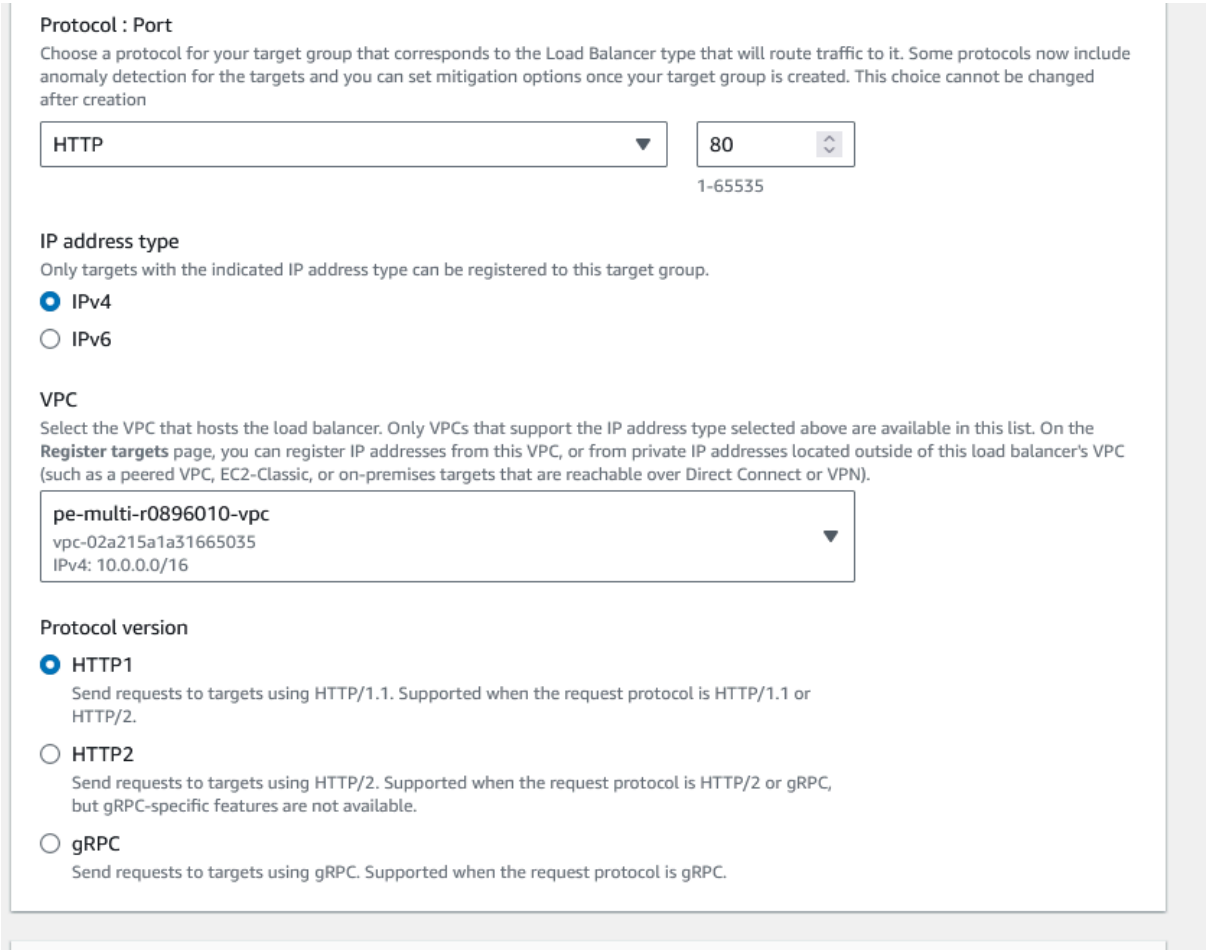
Aanmaken van Target Group voor de loadbalancer.

We gaan naar de “**EC2”** service en links in het menu kiezen we **target groups**

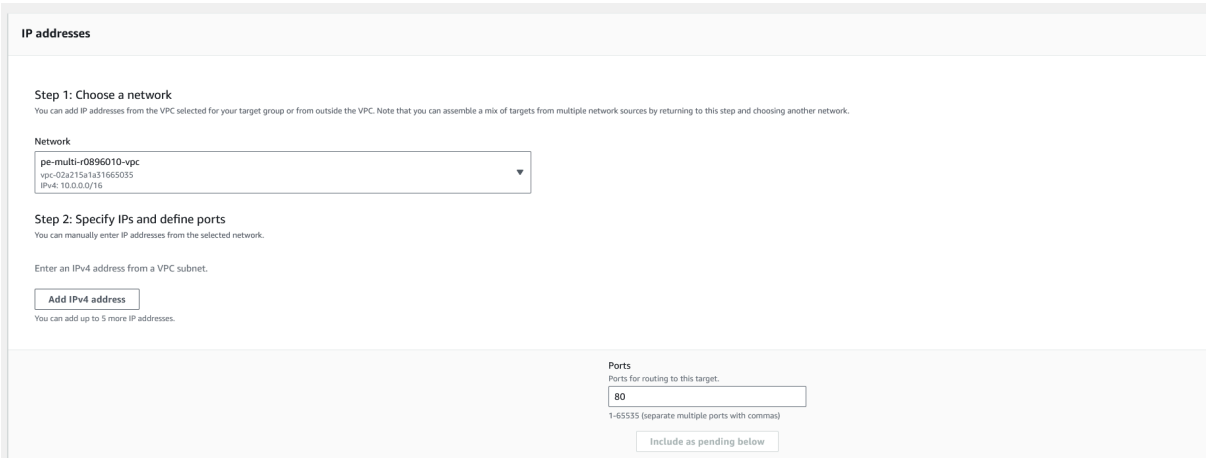
Klikken op “**Create new target group”** Dan kiezen we voor “**IP addresses”** onder de noemer **Basic configuration.**En geven we de target group een naam. “**container-r089600”**



Vervolgens kiezen we onze **VPC.**



vervolgens bij '**IP addresses"** kiezen we weer onze **VPC.** En maken we de target group aan.



Aanmaken van de Security Groups mbt de Application Load Balancer

Voor de loadbalancer zelf maken we een SG aan met volgende regels

Naam SG: **ALB-SG-r0896010**
Inbound rules:

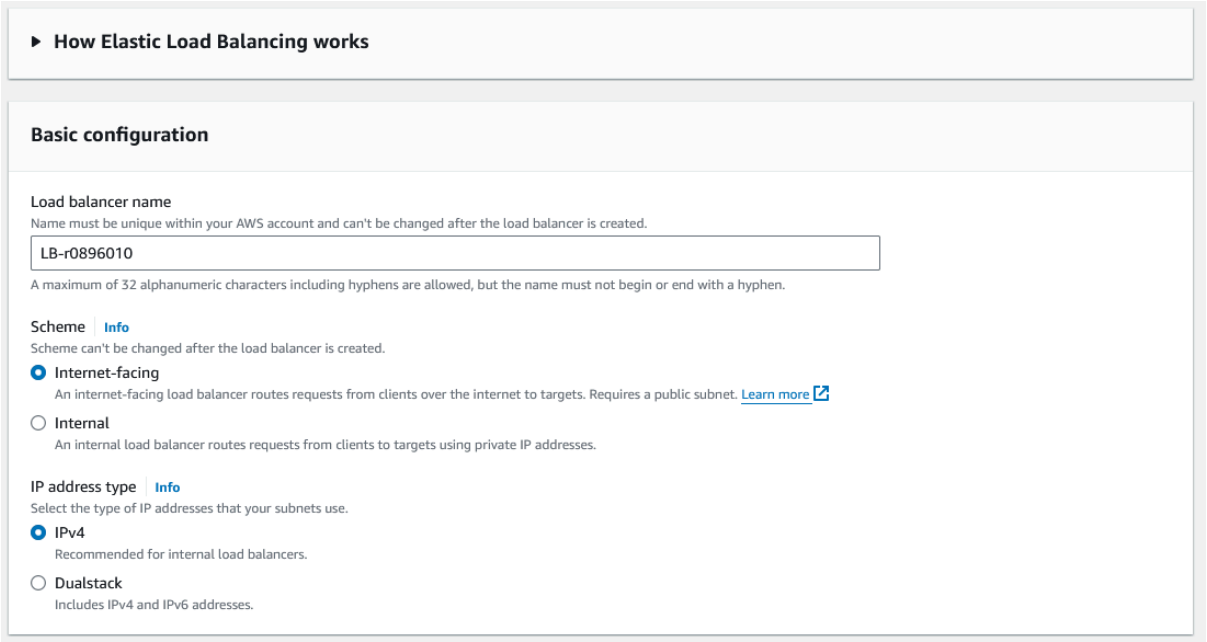
1. **HTTP TCP 80 CUSTOM** Source: **0.0.0.0/0**

Naam SG: **ALB-SG-CONTAINERS-r0896010**
Inbound rules:

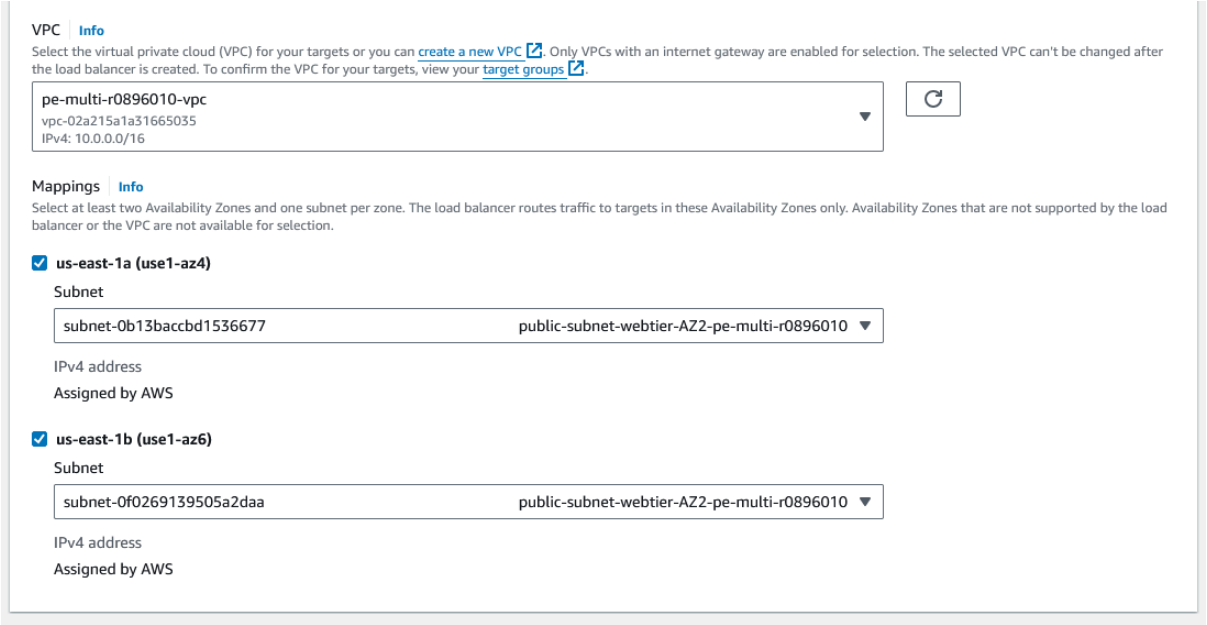
1. **HTTP TCP 80 CUSTOM** Source: **ALB-SG-r0896010**

De Application Load Balancer aanmaken.

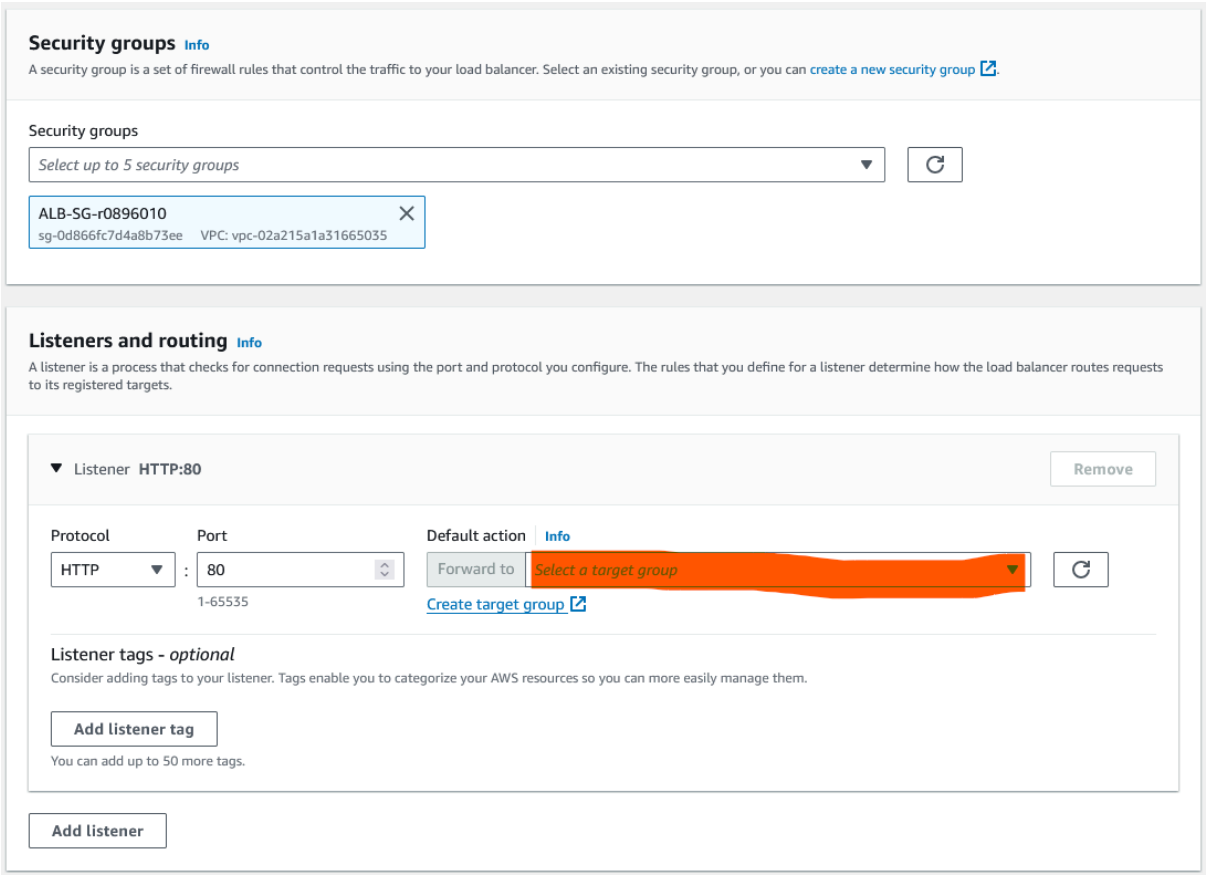
We kiezen links in het menu boven **"Target groups"** voor '**Load Balancers**" -> **"Create Load balancer"** vervolgens geven we deze een naam: **"LB-r0896010"** We kiezen voor **"Internet-facing"**



Dan kiezen we voor onze VPC dan de 2 **AZ's** en in elke **AZ** kiezen we onze **PUBLIC** subnets.



Bij **security groups** kiezen we de nieuw gemaakte **ALB-SG-r0896010**



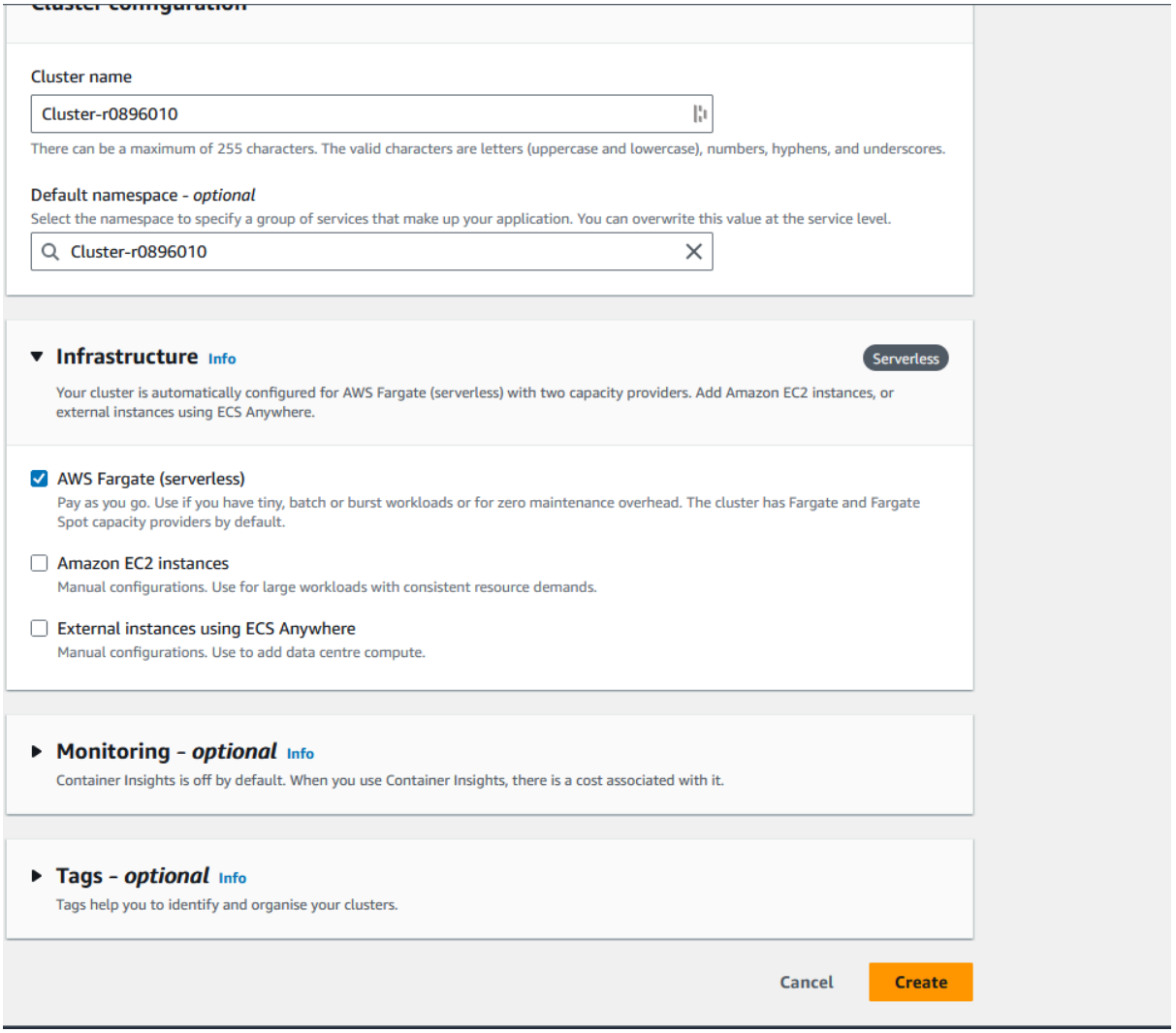
Dan selecteren we de **Target group** dat we hebben gemaakt: **container-r0896010**(ik kan deze niet selecteren omdat ik de ALB al heb gemaakt) je selecteert deze in de oranje gemarkeerde balk.

Opzetten van de Cluster/Targets/Service met ECS en Fargate.

We gaan onze container opspinnen. Door eerst een **Cluster** en **Tasks** aan te maken en vervolgens een **service.**

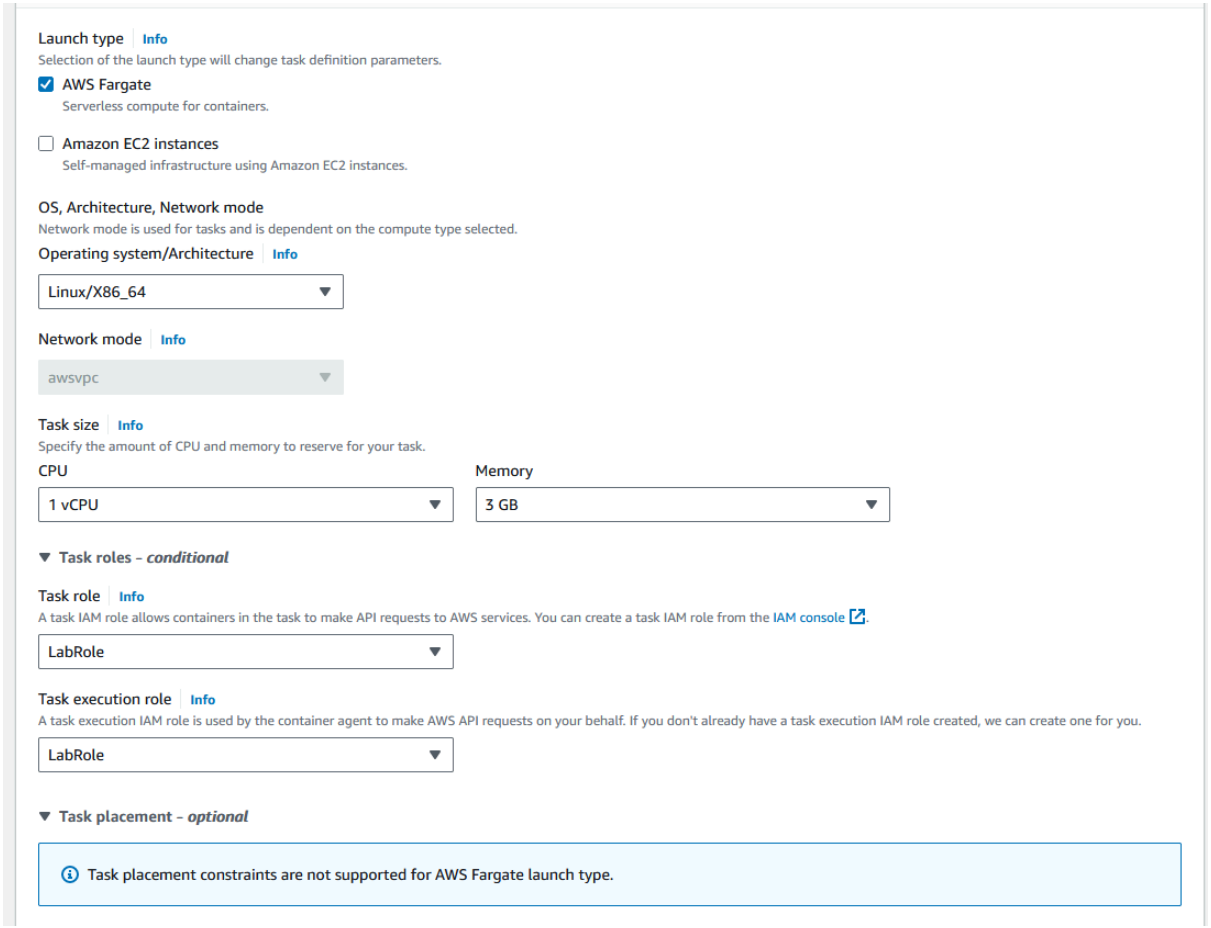
Aanmaken van een cluster.

We geven dit gewoon een **naam** en **default namespace**



Aanmaken van de Tasks

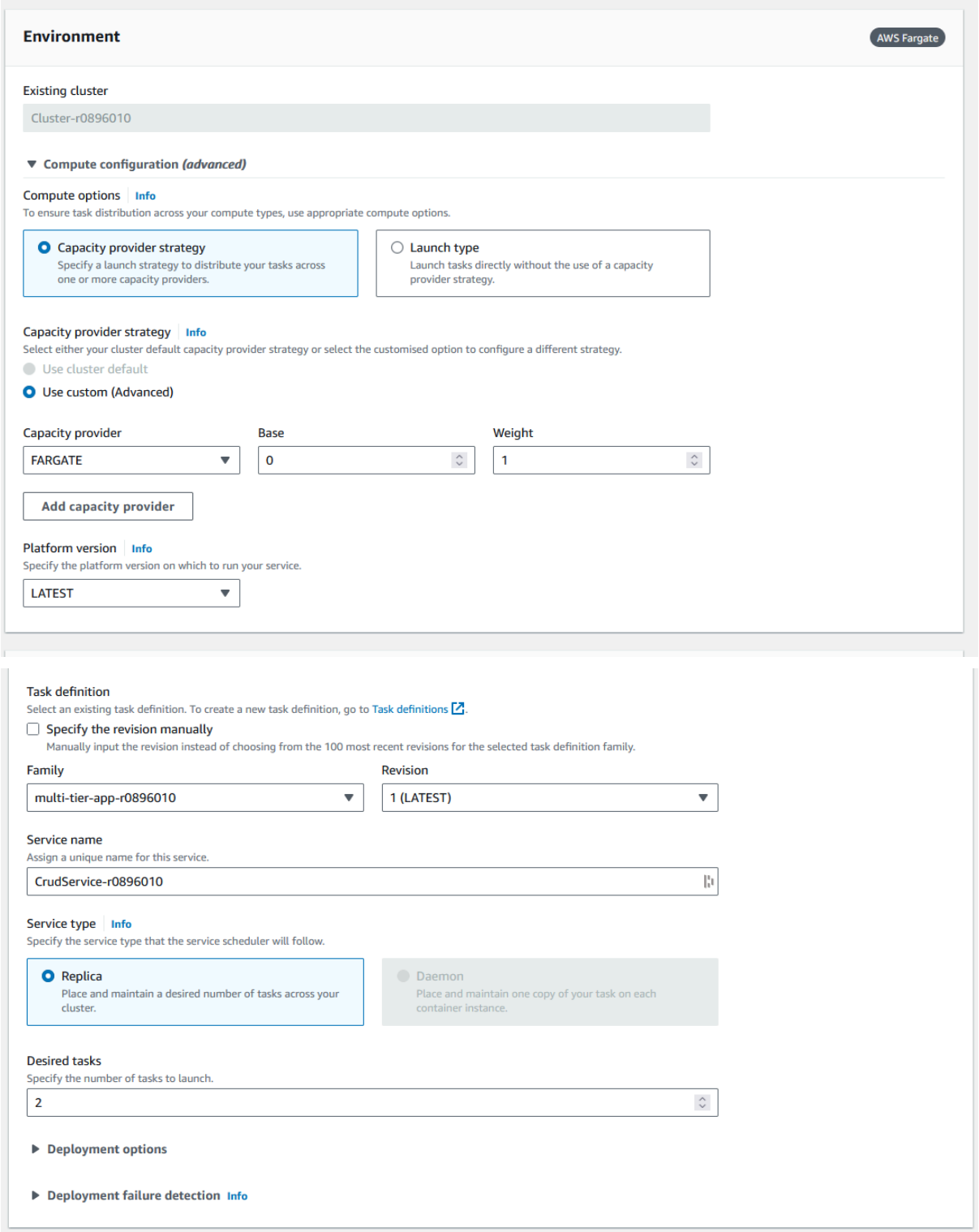
Vervolgens gaan we een **task definition** aanmaken. We klikken op “**Create task definition**”
Vervolgens geven we dit een naam: **multi-tier-app-r0896010**



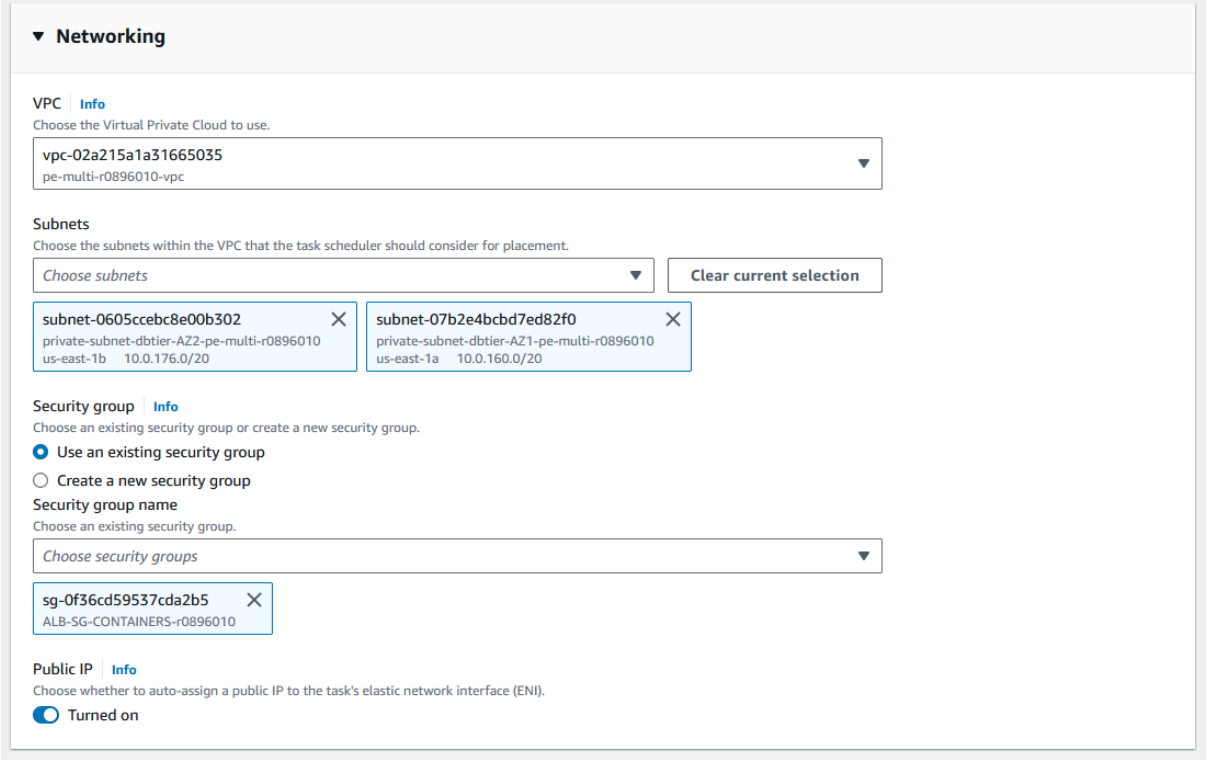
voor de rest laten we de instellingen zien hoe ze zijn.

Aanmaken van de service

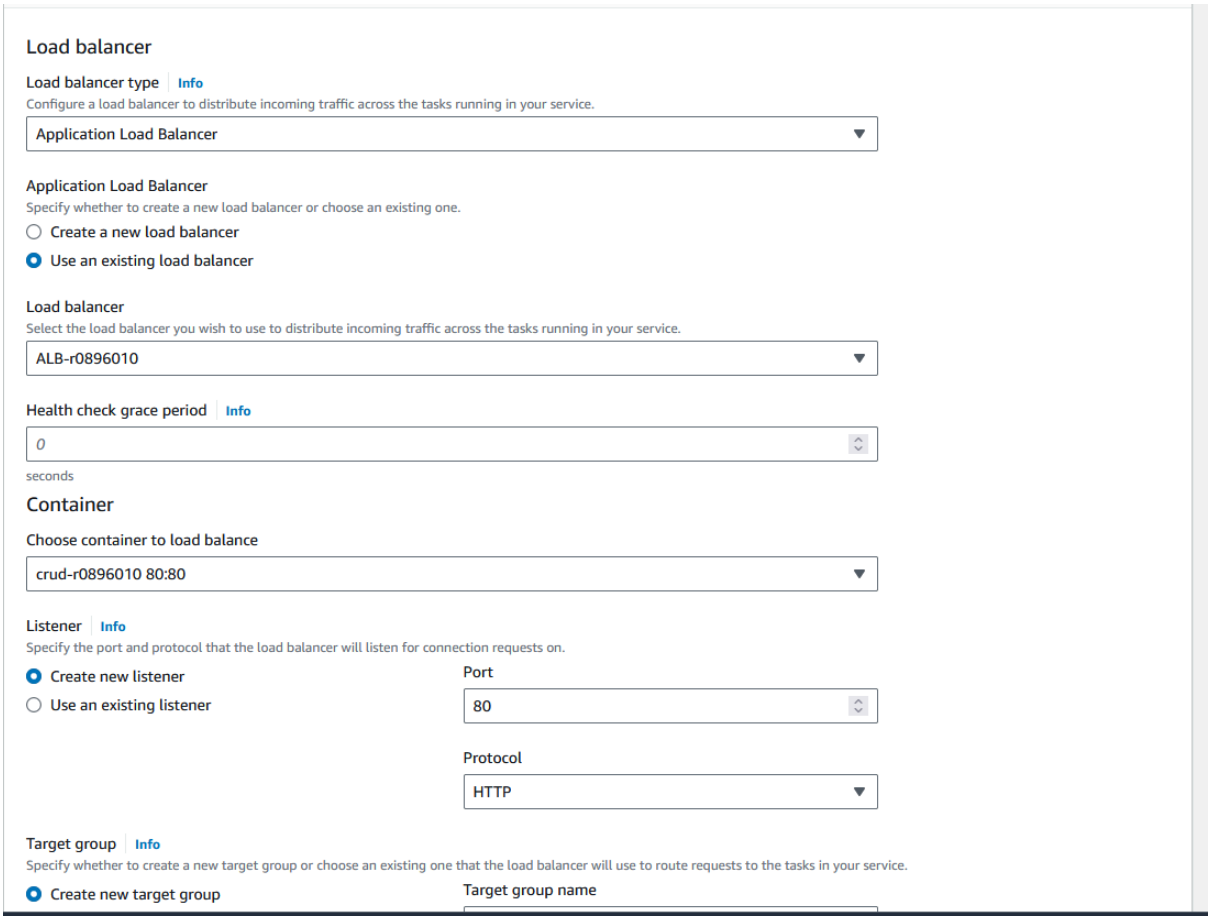
We laten volgende instellingen staan:



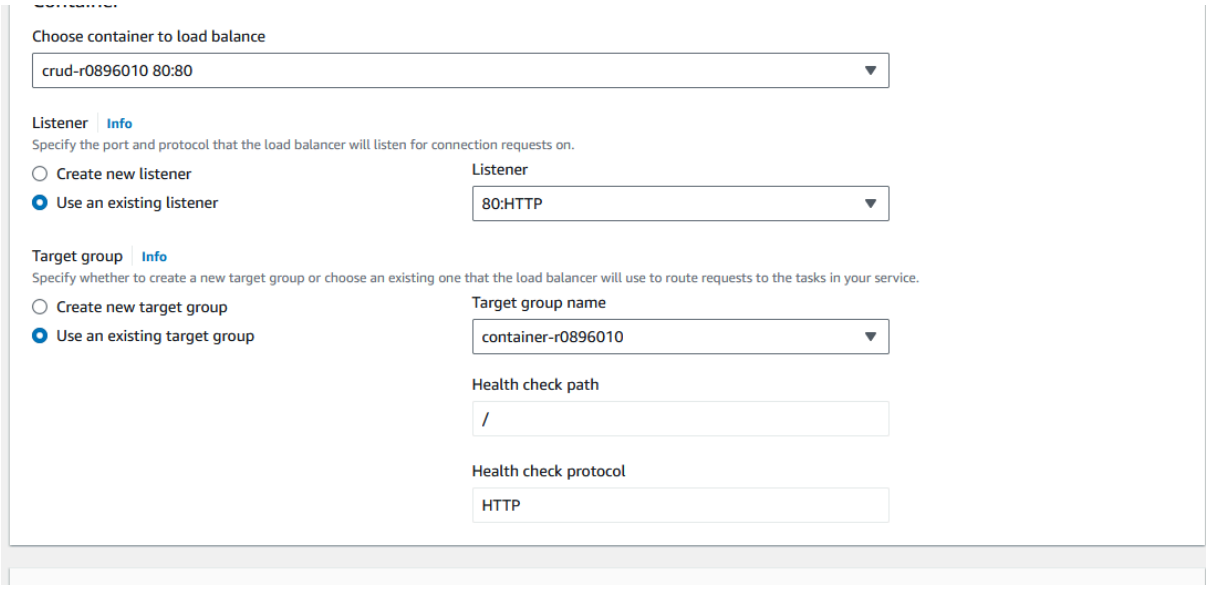
Volgende **Network Settings:**



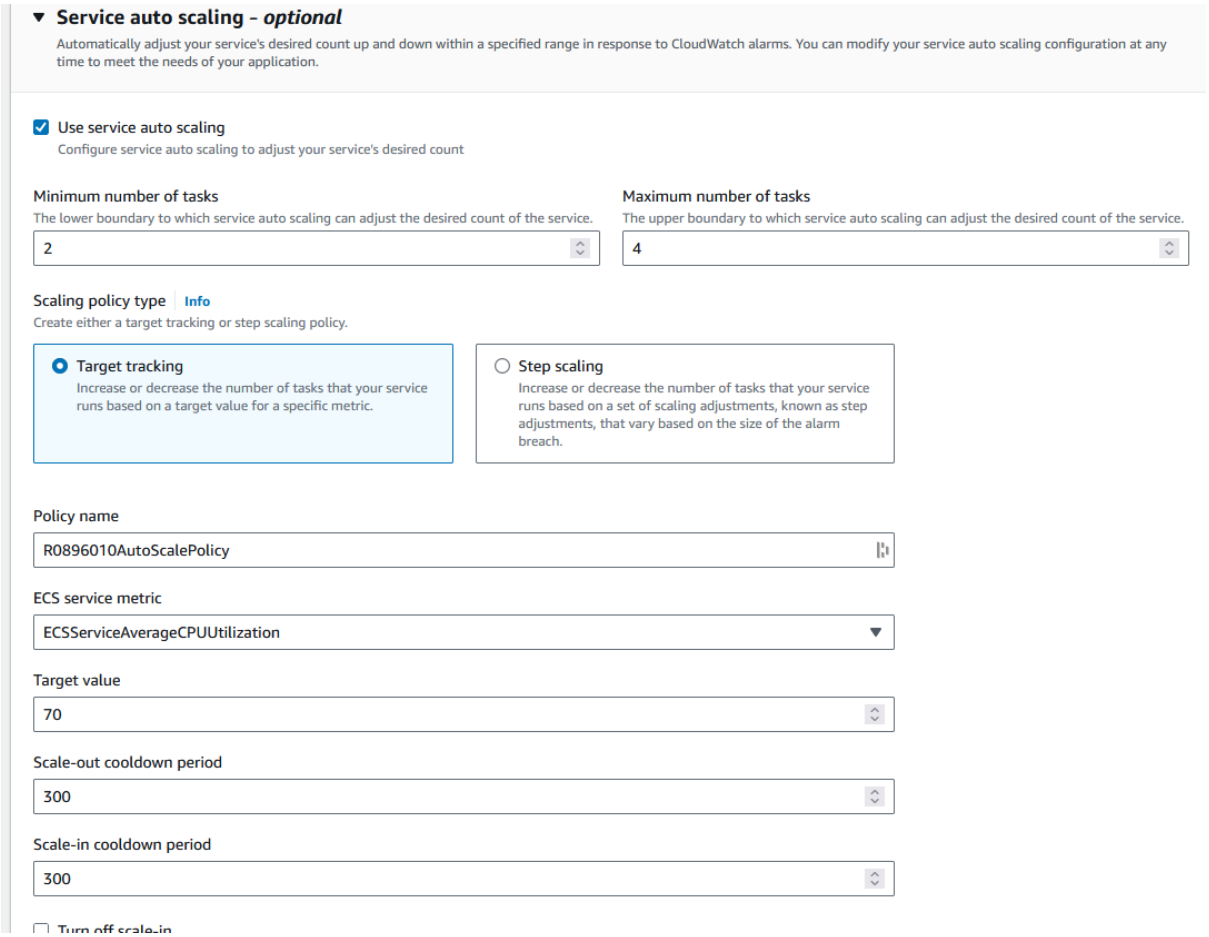
vervolgens gaan we de **ALB** toevoegen.



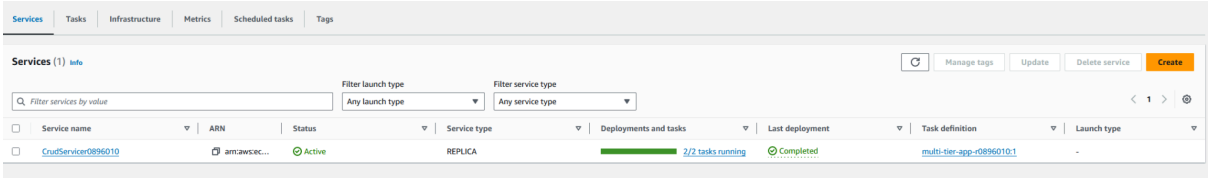
Next we make sure to attach our **Target group**



We voegen als extra **auto-scaling** toe.

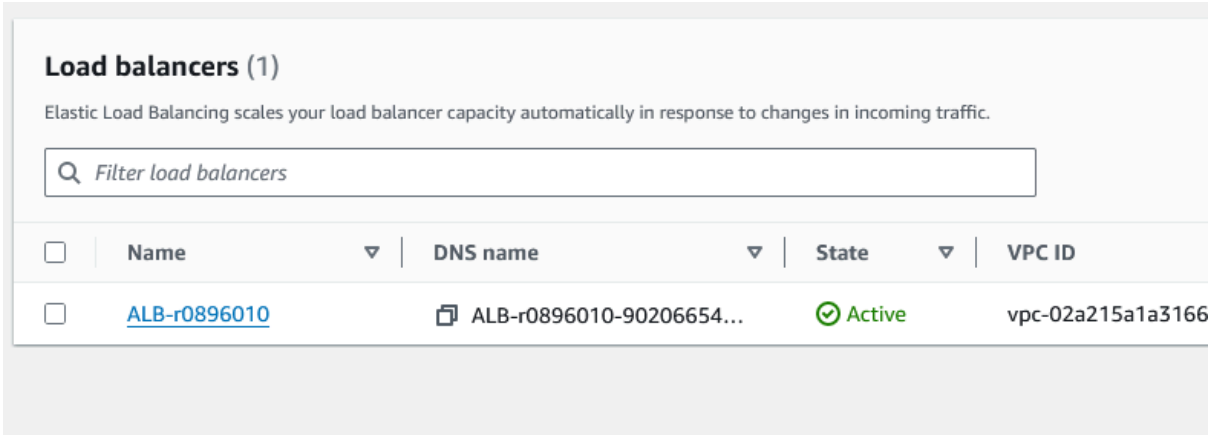


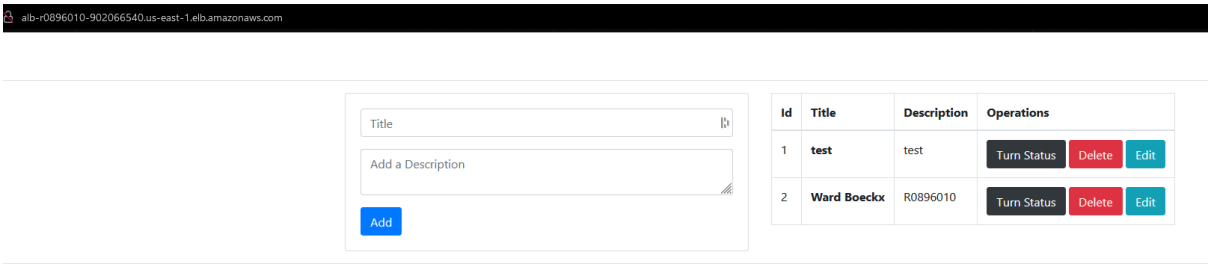
We starten de service en wachten tot we groen licht zien.



Testen of het werkt.

We gaan vervolgens naar **“EC2”** -> “**Loadbalancers”** We kiezen onze **ALB** en plakken de **DNS** naam in onze url.

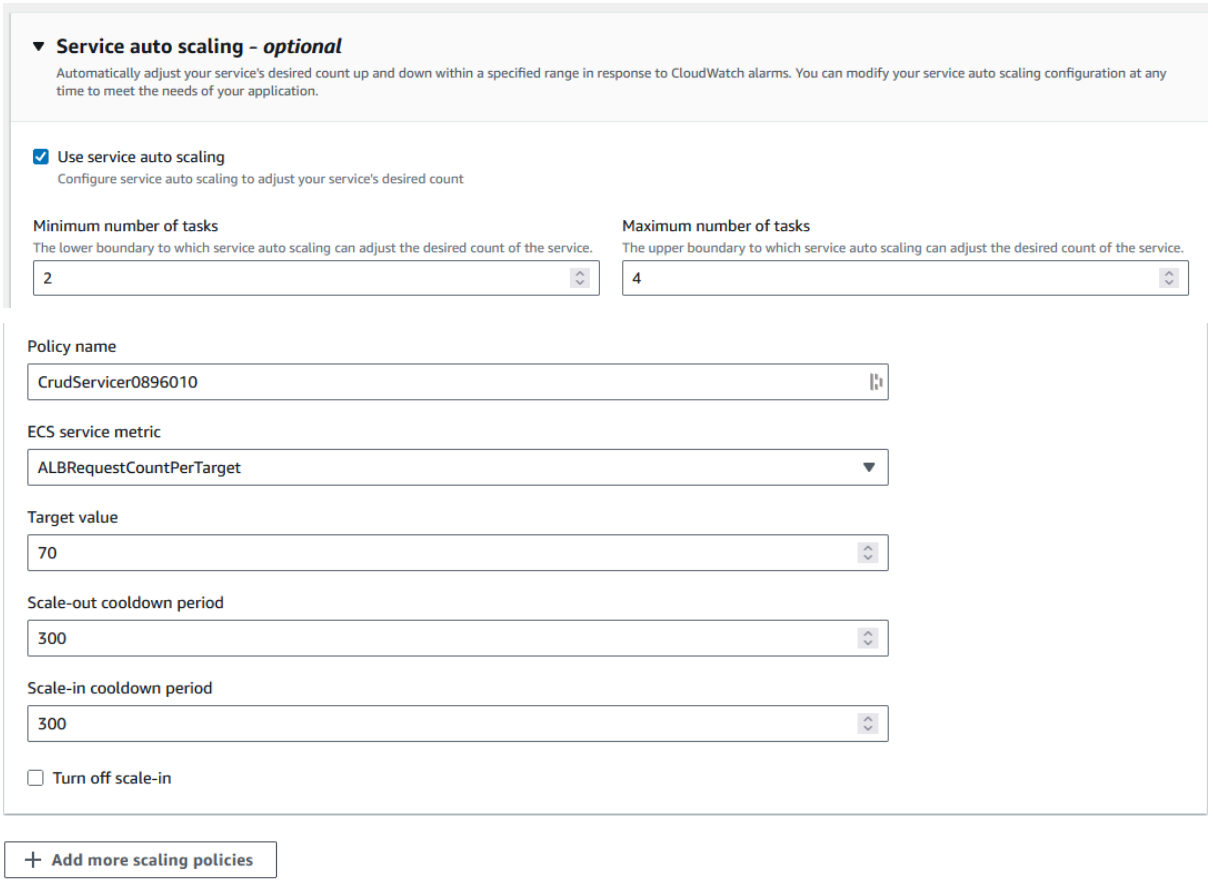




We zien dat deze werkt.

Auto Scaling

We gaan terug naar onze **Service** we kiezen voor “**Update Service”** We vinken “**Use service auto scaling**” aan. Vervolgens maken we een keuzen van de minimun aantal tasken en de maximum aantal. Vervolgens maken we een policy aan met waardes wanneer er aan auto scaling moet worden gedaan.



BOODSCHAP

Alexander, dankjewel voor de zeer aangename lessen, Deze waren zeer interessant en heel goed uitgelegd. Mvg

Ward Boeckx - R0896010