

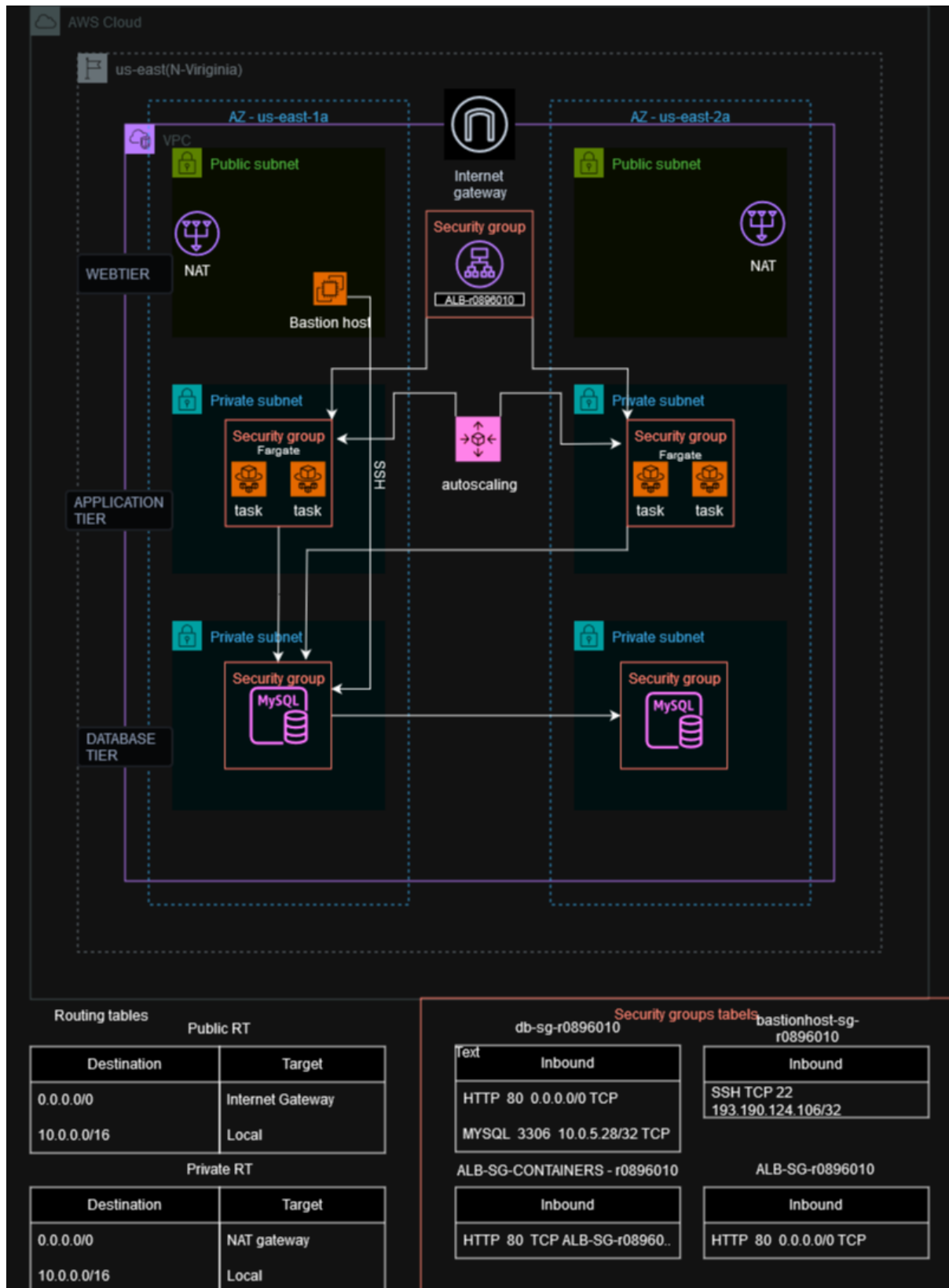
Multi-tier-architecture



Ward Boeckx - R0896010

Architectuur diagram	3
Netwerk gedeelte instellen (VPC,Subnets,RT's)	4
Het maken van onze VPC.	4
Subnets	7
Routing tables	10
Database(RDS)	10
Aanmaken Security Group Database	11
Aanmaken Database	11
Bastionhost/jump server.	16
Aanmaken van de Security Group voor de Bastion host.	16
Aanmaken van de EC2 voor onze Bastion host	16
SSH testen richting de bastion host.	18
Connecteren vanaf de Bastion host naar de Database(RDS)	21
Aanpassen Crud app	23
Aanpassingen code crud-app	24
Instellen Gunicorn(WSGI)	25
Aanmaken van de Dockerfile.	25
Aanmaken van ECR-repository en pushen van Docker-image	26
Inloggen op AWS CLI en bouwen van docker image.	28
Aanmaken van Application Load Balancer	31
Aanmaken van Target Group voor de loadbalancer.	31
Aanmaken van de Security Groups mbt de Application Load Balancer	34
De Application Load Balancer aanmaken.	34
Opzetten van de Cluster/Targets/Service met ECS en Fargate.	36
Aanmaken van een cluster.	36
Aanmaken van de Tasks	36
Aanmaken van de service	37
Testen of het werkt.	41
Auto Scaling	41

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](#)

- ☒ Enable DNS hostnames
- ☒ 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**.

Subnets (12) Info			Actions ▼	Create subnet
Find resources by attribute or tag		< 1 >		
☐	Name	▼	Subnet ID	
☐	public-subnet-webtier-AZ2-pe-multi-r0896010		subnet-0b1	
☐	public-subnet-webtier-AZ2-pe-multi-r0896010		subnet-0f0	
☐	private-subnet-dbtier-AZ2-pe-multi-r0896010		subnet-060	
☐	private-subnet-dbtier-AZ1-pe-multi-r0896010		subnet-07b	
☐	private-subnet-apptier-AZ1-pe-multi-r0896010		subnet-0b0	
☐	private subnet-apptier-AZ2-pe-multi-r0896010		subnet-0ed	

Internet gateway(s)

- pe-multi-r0896010-igw

igw-0ffa35f29e8bb3444 / pe-multi-r0896010-igw

Actions ▼

Details [Info](#)

Internet gateway ID

 igw-0ffa35f29e8bb3444

VPC ID

[vpc-02a215a1a31665035](#) | [pe-multi-r0896010-vpc](#)

State

 Attached

Owner

 698105759521

Tags

Manage tags

 Search tags

< 1 > 

Key	Value
Name	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)					
Find subnet association					
< 1 > 					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR		
public-subnet-webtier-AZ2-pe-multi-r0896010	subnet-0b13bacbd1536677	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)					
Find subnet association					
< 1 > 					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR		
private-subnet-dbtier-AZ1-pe-multi-r0896010	subnet-07b2e4bcd7ed82f0	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-0605ccebcb8e00b302	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:

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

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

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 Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-08839e0154cd5c12b	HTTP	TCP	80	Custom 0.0.0.0/0	
sgr-02a4ad058ff15f93c	MySQL/Aurora	TCP	3306	Custom 10.0.5.28/32	

Add rule

Outbound rules (1)

☐

Name	Security group rule...	IP version	Type	Protocol	Port range	Destination
-	sgr-031cdc8ef439af12a	IPv4	All traffic	All	All	0.0.0.0/0

Aanmaken Database

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

"Standard create" -> "MySQL" -> "Free tier"

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.

Engine options

Engine type [Info](#)

☐ Aurora (MySQL Compatible)


☐ Aurora (PostgreSQL Compatible)


☒ **MySQL**


☐ MariaDB


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"**

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

r0896010

DB instance identifier already exists

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

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

admin

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

☐ Manage master credentials in AWS Secrets Manager

Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

 If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
[Learn more](#) 

☐ Auto generate a password

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

Master password [Info](#)

.....



Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm master password [Info](#)

.....



 The Confirm master password doesn't match.

Instance config. en **Storage** heb ik gelaten hoe het was. Vervolgens onder de noemer **Connectivity** kies ik voor mijn **VPC** -> "**pe-multi-r0896010-vpc**" vervolgens 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:

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

sng-db-r0896010

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

subnetgroup voor database

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

pe-multi-r0896010-vpc (vpc-02a215a1a31665035)

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-0b057a9abeb85fb86 (10.0.128.0/20) ✕

subnet-0605cceb8e00b302 (10.0.176.0/20) ✕

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**)

Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

pe-multi-r0896010-vpc (vpc-02a215a1a31665035)

6 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

 After a database is created, you can't change its VPC.

DB subnet group [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

sng-db-r0896010

2 Subnets, 2 Availability Zones

Public access [Info](#)

☐ Yes

RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

☒ No

RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

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 allow the appropriate 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

db-sg-r089610 

*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

Availability & durability

Multi-AZ deployment [Info](#)

☐ Do not create a standby instance

☒ Create a standby instance (recommended for production usage)

Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

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 host** 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.:

Inbound rules (1)							
Q Search							
< 1 > ⚙							
me	Security group rule...	IP version	Type	Protocol	Port range	Source	
	sgr-09f29a838be6074bc	IPv4	SSH	TCP	22	193.190.124.106/32	

Outbound rules (1)							
Q Search							
< 1 > ⚙							
☐	Name	Security group rule...	IP version	Type	Protocol	Port range	Destination
☐	-	sgr-0076e6d505d628...	IPv4	All traffic	All	All	0.0.0.0/0

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

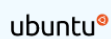
Amazon
Linux



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](#)

▼ **Network settings** [Info](#)

VPC - required [Info](#)

vpc-02a215a1a31665035 (pe-multi-r0896010-vpc)
10.0.0.0/16

Subnet [Info](#)

subnet-0b13bacbd1536677 public-subnet-webtier-AZ2-pe-multi-r0896010
VPC: vpc-02a215a1a31665035 Owner: 698105759521
Availability Zone: us-east-1a IP addresses available: 4088 CIDR: 10.0.0.0/20

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

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

☐ Create security group ☒ Select existing security group

Common security groups [Info](#)

Select security groups

bastionhost-sg-r0896010 sg-0071f0a8f76f43277 ✕
VPC: vpc-02a215a1a31665035

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

► **Advanced network configuration**

Dan kunnen we zien dat deze is opgedraaid.

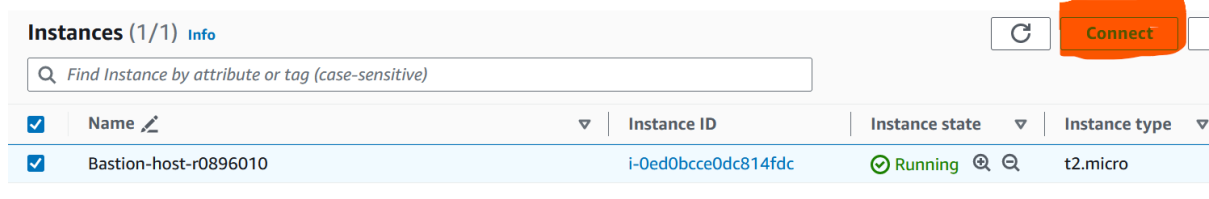
☐	Bastion-host-r0896010	i-Oed0bcce0dc814fdc	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a
--------------------------	-----------------------	---------------------	---------	----------	-------------------	-----------------------------	------------

SSH testen richting de bastion host.

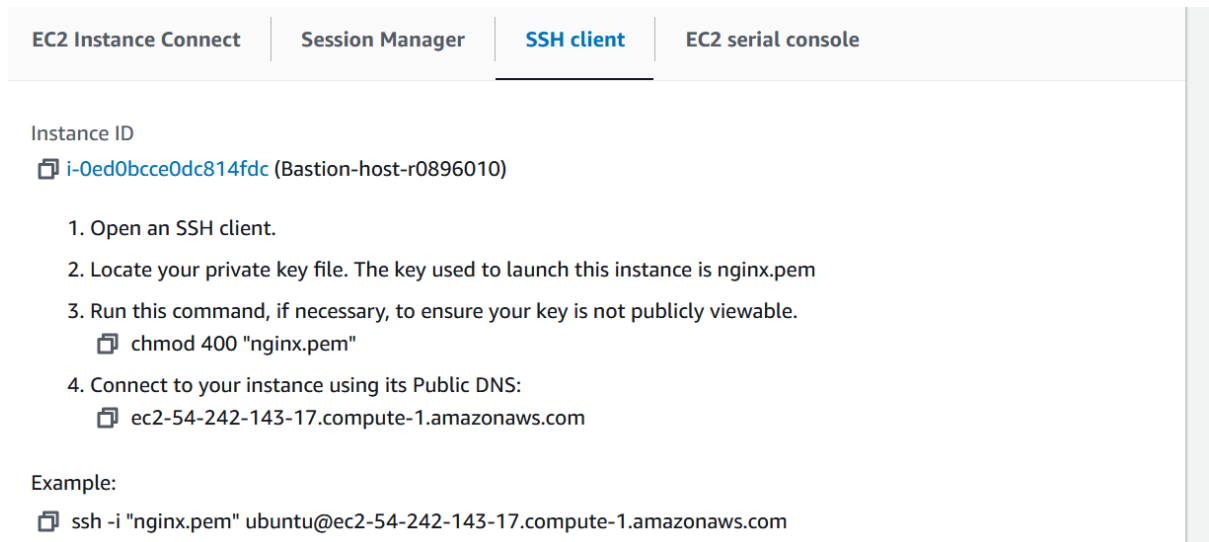
Nu gaan we testen of we kunnen ssh'en vanaf onze ubuntu(vm) naar onze Bastion host. Bij het aanmaken van onze bastion host kan je "**Create new keypair**", als je dit doet, download je een .pem file in mijn geval. Deze file heb ik op volgende locatie gezet: **home/ward/multi-tier/Credentials**

```
ward@linuxUbuntu:~/multi-tier/Credentials$ ls
nginx.pem
ward@linuxUbuntu:~/multi-tier/Credentials$ pwd
/home/ward/multi-tier/Credentials
ward@linuxUbuntu:~/multi-tier/Credentials$
```

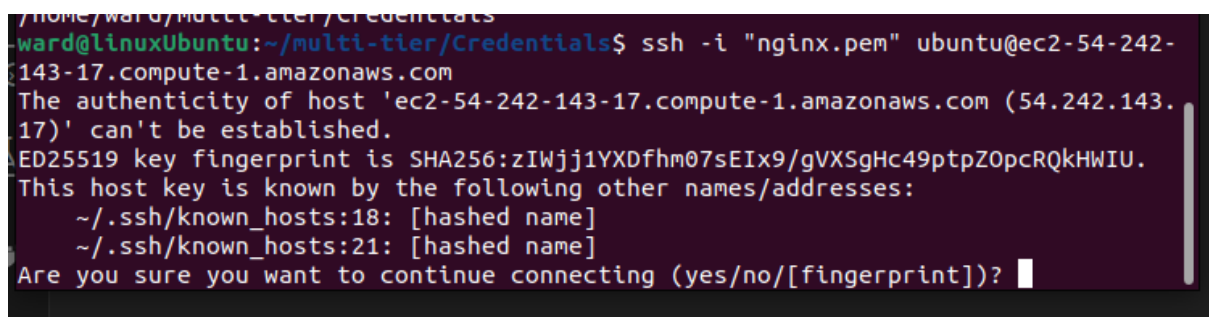
Als we dit hebben gedaan kunnen we simpelweg in de AWS console onze **bastion host** instance selecteren en vervolgens rechtsboven op de knop “**Connect**” te klikken



Dan volgen we simpelweg die stappen.



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



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

```
System information as of Fri Jan 12 14:32:39 UTC 2024

System load:  0.0                Processes:           97
Usage of /:   25.4% of 7.57GB    Users logged in:    0
Memory usage: 20%              IPv4 address for eth0: 10.0.5.28
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

12 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Jan 12 09:53:54 2024 from 193.190.124.106
ubuntu@ip-10-0-5-28:~$
```

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.

Connectivity & security

Mo

Connectivity & security

Endpoint & port

Endpoint

r0896010.c5uc6e2sc91f.us-east-1.rds.amazonaws.com

Port

3306

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

```
ubuntu@ip-10-0-5-28:~$ sudo su newuser
newuser@ip-10-0-5-28:~/home/ubuntu$ mysql -h r0896010.c5uc6e2sc91f.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 314
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> s
```

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.

```
1 import os
2 basedir = os.path.abspath(os.path.dirname(__file__))
3
4 class Config(object):
5     SECRET_KEY = 'do-or-do-not-there-is-no-try'
6     # SECRET_KEY = os.environ.get('SECRET_KEY') or 'do-or-do-not-there-is-no-try'
7     SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://admin:administrator@r0896010.c5uc6e2sc91f.us-east-1.rds.amazonaws.com/r0896010'
8     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 geval “**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 > ...  
1  from flask import Flask  
2  from app.config import Config  
3  from flask_sqlalchemy import SQLAlchemy  
4  from flask_migrate import Migrate  
5  import pymysql  
6  app = Flask(__name__)  
7  app.config.from_object(Config)  
8  db = SQLAlchemy(app)  
9  migrate = Migrate(app, db)  
10  
11 from app import routes, models  
12 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 > ...  
1  bind = '0.0.0.0:80'  
2  workers = 4
```

Simpele configuratie.

Aanmaken van de Dockerfile.

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

example-flask-crud > Dockerfile > ...

```
1 FROM python:3.10.12
2 WORKDIR /app
3 COPY . /app/
4 RUN pip install -r requirements.txt
5 EXPOSE 80
6 ENV FLASK_APP=crudapp.py
7 CMD [ "gunicorn", "-c", "gunicorn_config.py", "crudapp:app"]
```

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**”

Create repository

General settings

Visibility settings [Info](#)

Choose the visibility setting for the repository.

☒ **Private**

Access is managed by IAM and repository policy permissions.

☐ **Public**

Publicly visible and accessible for image pulls.

Repository name

Provide a concise name. A developer should be able to identify the repository contents by the name.

698105759521.dkr.ecr.us-east-1.amazonaws.com/

 **A repository with this name already exists**

23 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

Tag immutability [Info](#)

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

☒ **Disabled**

 Once a repository has been created, the visibility setting of the repository can't be changed.

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

Private repositories					
Repositories (1)		 View push commands		Delete	Actions
Filter status				 1 	
Repository name	URI	Created at	Tag immutability	Scan frequency	Encryption type
☒ r0896010-schoolopdracht	698105759521.dkr.ecr.us-east-1.amazonaws.com/r0896010-schoolopdracht	11 January 2024, 20:59:56 (UTC+01)	Disabled	Manual	AES-256

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

Push commands for r0896010-schoolopdracht

macOS / Linux

Windows

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry.
Use the AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 698105759521.dkr.ecr.us-east-1.amazonaws.com
```

Note: if you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch, see the instructions [here](#). You can skip this step if your image has already been built:

```
docker build -t r0896010-schoolopdracht .
```
3. After the build is completed, tag your image so you can push the image to this repository:

```
docker tag r0896010-schoolopdracht:latest 698105759521.dkr.ecr.us-east-1.amazonaws.com/r0896010-schoolopdracht:latest
```
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 698105759521.dkr.ecr.us-east-1.amazonaws.com/r0896010-schoolopdracht:latest
```

Close

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**

Cloud Access

Close

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////////wEaDDCqNkrp1C7gLTsFAyK8AXY5vcEkPwzYjwHEm1BXu6
ISkJptVJzJ0yCC4p0CVSR7E06uAZ0e20LLF34L262cWHoLHo0uW5WL4oqzGrVIwdkQ4BKt4mRDz43R4CI/W8Aq2V
czMow01iSsr6QfD208SnxC1YjSrMOGaYzbe30jHuuXI2rf0m5GCQilumg0Ri1ewPy3Ak4L3wIW2QF8YAcgXio1TS
UMHtwSrj4X3950AipClX0LP90xoM696EKpm1tZnNJDIGZNAzx2hTnRKJCJh60GMi1z16TYZinUgqXRl3jzZg52Sd
UFSsjvNGGWgpIMXeD150izmVX9gTqe/sJo/NE=
```

Cloud Labs

```
ward@linuxUbuntu:~$ ls
aws           Documents      multi-tier    Public        tier
awscli.v2.zip Downloads      Music         snap          Videos
Desktop       example-flask-crud Pictures       Templates
ward@linuxUbuntu:~$ cd .aws
ward@linuxUbuntu:~/.aws$ ls -a
.  ..  config  credentials
ward@linuxUbuntu:~/.aws$ sudo nano credentials
```

```
GNU nano 6.2 credentials
[default]
aws_access_key_id=ASIA2FCSYQ4QWHMROY2E
aws_secret_access_key=v9VQfb82EApP8uGtWfA9AFPo4du09/PKwPohUxps
aws_session_token=FwoGZXIvYXdzEj3////////wEaDOMMb/bc/OtV9YoLcCK8Ad078IvD/fhSM
```

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

```
(venv) ward@linuxUbuntu:~/multi-tier/example-flask-crud$ aws ecr get-login-password --region us-east-1 | sudo docker login --username AWS --password-stdin 698105759521.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
(venv) ward@linuxUbuntu:~/multi-tier/example-flask-crud$ s
```

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.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Unable to locate credentials. You can configure credentials by running "aws configure".
Error: Cannot perform an interactive login from a non TTY device
ward@linuxubuntu:~/example-flask-cruds$ sudo aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 698105759521.dkr.ecr.us-east-1.amazonaws.com
[sudo] password for ward:
Unable to locate credentials. You can configure credentials by running "aws configure".
Error: Cannot perform an interactive login from a non TTY device
ward@linuxubuntu:~/example-flask-cruds$ aws configure
AWS Access Key ID [None]:
ward@linuxubuntu:~/example-flask-cruds$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 698105759521.dkr.ecr.us-east-1.amazonaws.com
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/auth": dial unix /var/run/docker.sock: c
onnect: permission denied
ward@linuxubuntu:~/example-flask-cruds$ sudo aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 698105759521.dkr.ecr.us-east-1.amazonaws.com
Unable to locate credentials. You can configure credentials by running "aws configure".
Error: Cannot perform an interactive login from a non TTY device
ward@linuxubuntu:~/example-flask-cruds$ sudo docker login --username AWS --password-stdin 698105759521.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ward@linuxubuntu:~/example-flask-cruds$ docker tag cp_crudapp:latest 698105759521.dkr.ecr.us-east-1.amazonaws.com/crud-app-r0896010:latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/cp_crudapp:latest/tag?repo=698105
759521.dkr.ecr.us-east-1.amazonaws.com%2Fcrud-app-r0896010%2Ftag=latest": dial unix /var/run/docker.sock: connect: permission denied
ward@linuxubuntu:~/example-flask-cruds$ sudo docker tag cp_crudapp:latest 698105759521.dkr.ecr.us-east-1.amazonaws.com/crud-app-r0896010:latest
ward@linuxubuntu:~/example-flask-cruds$ docker push 698105759521.dkr.ecr.us-east-1.amazonaws.com/crud-app-r0896010:latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/698105759521.dkr.ecr.us-east-1.am
azonaws.com/crud-app-r0896010/push?tag=latest": dial unix /var/run/docker.sock: connect: permission denied
ward@linuxubuntu:~/example-flask-cruds$ sudo docker push 698105759521.dkr.ecr.us-east-1.amazonaws.com/crud-app-r0896010:latest
The push refers to repository [698105759521.dkr.ecr.us-east-1.amazonaws.com/crud-app-r0896010]
83080bda9f937: Pushed
253f205d7212: Pushed
ccf30980520c: Pushed
84f93ee72c46: Pushed
c85e1a3b336f: Pushed
d7f0499080ba: Pushed
428ddb13266b: Pushed
b2e5b1eee192: Pushed
b485c6cd33a6: Pushing [=====>] 44.78MB/587.1MB
6aa072026017: Pushing [=====>] 43.08MB/176.7MB
43ba18a5ea9f8: Pushing [=====>] 10.4MB/48.4MB
ff61a9b258e5: Pushing [=====>] 19.58MB/116.5MB
```

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

Images (1)									
Search artefacts									
☐	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Scan status	Vulnerabilities	
☐	latest	Image	11 January 2024, 21:17:15 (UTC+01)	401.21	Copy URI	sha256:f7b540680cb9522645564f5252651eac995faa5b4965d44be711efb3100f57ff	-	-	

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”**

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

☐ Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#)  to manage and scale your EC2 capacity.

☒ IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

☐ Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

☐ Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

containers-r0896010

Vervolgens kiezen we onze **VPC**.

Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

☒ IPv4

☐ IPv6

VPC

Select the VPC that hosts the load balancer. Only VPCs that support the IP address type selected above are available in this list. On the **Register targets** page, you can register IP addresses from this VPC, or from private IP addresses located outside of this load balancer's VPC (such as a peered VPC, EC2-Classic, or on-premises targets that are reachable over Direct Connect or VPN).

pe-multi-r0896010-vpc
vpc-02a215a1a31665035
IPv4: 10.0.0.0/16

Protocol version

☒ HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

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

IP addresses

Step 1: Choose a network

You can add IP addresses from the VPC selected for your target group or from outside the VPC. Note that you can assemble a mix of targets from multiple network sources by returning to this step and choosing another network.

Network

pe-multi-r0896010-vpc
vpc-02a215a1a31665035
IPv4: 10.0.0.0/16

Step 2: Specify IPs and define ports

You can manually enter IP addresses from the selected network.

Enter an IPv4 address from a VPC subnet.

Add IPv4 address

You can add up to 5 more IP addresses.

Ports

Ports for routing to this target.

80

1-65535 (separate multiple ports with commas)

Include as pending below

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"**

► How Elastic Load Balancing works

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

☒ **Internet-facing**
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ **Internal**
An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)
Select the type of IP addresses that your subnets use.

☒ **IPv4**
Recommended for internal load balancers.

☐ **Dualstack**
Includes IPv4 and IPv6 addresses.

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

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

pe-multi-r0896010-vpc
vpc-02a215a1a31665035
IPv4: 10.0.0.0/16

↻

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ **us-east-1a (use1-az4)**

Subnet

subnet-0b13baccbd1536677

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

IPv4 address
Assigned by AWS

☒ **us-east-1b (use1-az6)**

Subnet

subnet-0f0269139505a2daa

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

IPv4 address
Assigned by AWS

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

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

↻

ALB-SG-r0896010 ✕
sg-0d866fc7d4a8b73ee VPC: vpc-02a215a1a31665035

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Remove

Protocol HTTP

Port 80
1-65535

Default action [Info](#)
Forward to

Select a target group

[Create target group](#)

↻

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

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**

The screenshot shows the 'Create cluster' configuration page in the AWS Management Console. The 'Cluster name' field is filled with 'Cluster-r0896010'. Below it, a note states: 'There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.' The 'Default namespace - optional' field is also filled with 'Cluster-r0896010'. Below this, the 'Infrastructure' section is expanded, showing 'AWS Fargate (serverless)' selected with a checkbox. A 'Serverless' badge is visible in the top right of this section. Other options like 'Amazon EC2 instances' and 'External instances using ECS Anywhere' are unselected. Below the infrastructure section, there are sections for 'Monitoring - optional' and 'Tags - optional', both with their respective 'Info' links. At the bottom right, there are 'Cancel' and 'Create' buttons.

Cluster configuration

Cluster name

Cluster-r0896010

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

Default namespace - *optional*

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

Cluster-r0896010

▼ Infrastructure [Info](#) Serverless

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances, or external instances using ECS Anywhere.

☒ AWS Fargate (serverless)

Pay as you go. Use if you have tiny, batch or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

☐ Amazon EC2 instances

Manual configurations. Use for large workloads with consistent resource demands.

☐ External instances using ECS Anywhere

Manual configurations. Use to add data centre compute.

► Monitoring - *optional* [Info](#)

Container Insights is off by default. When you use Container Insights, there is a cost associated with it.

► Tags - *optional* [Info](#)

Tags help you to identify and organise your clusters.

Cancel Create

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**

Launch type | [Info](#)
 Selection of the launch type will change task definition parameters.

☒ **AWS Fargate**
 Serverless compute for containers.

☐ **Amazon EC2 instances**
 Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode
 Network mode is used for tasks and is dependent on the compute type selected.

Operating system/Architecture | [Info](#)
 Linux/X86_64 ▼

Network mode | [Info](#)
 awsvpc ▼

Task size | [Info](#)
 Specify the amount of CPU and memory to reserve for your task.

CPU **Memory**

1 vCPU ▼ 3 GB ▼

▼ **Task roles - conditional**

Task role | [Info](#)
 A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the [IAM console](#).

LabRole ▼

Task execution role | [Info](#)
 A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.

LabRole ▼

▼ **Task placement - optional**

ⓘ Task placement constraints are not supported for AWS Fargate launch type.

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

Aanmaken van de service

We laten volgende instellingen staan:

Environment

AWS Fargate

Existing cluster

Cluster-r0896010

▼ Compute configuration (advanced)

Compute options

Info

To ensure task distribution across your compute types, use appropriate compute options.

☒ Capacity provider strategy

Specify a launch strategy to distribute your tasks across one or more capacity providers.

☐ Launch type

Launch tasks directly without the use of a capacity provider strategy.

Capacity provider strategy

Info

Select either your cluster default capacity provider strategy or select the customised option to configure a different strategy.

☐ Use cluster default

☒ Use custom (Advanced)

Capacity provider

Base

Weight

FARGATE

0

1

Add capacity provider

Platform version

Info

Specify the platform version on which to run your service.

LATEST

Task definition

Select an existing task definition. To create a new task definition, go to [Task definitions](#).

☐ Specify the revision manually

Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family

Revision

multi-tier-app-r0896010

1 (LATEST)

Service name

Assign a unique name for this service.

CrudService-r0896010

Service type

Info

Specify the service type that the service scheduler will follow.

☒ Replica

Place and maintain a desired number of tasks across your cluster.

☐ Daemon

Place and maintain one copy of your task on each container instance.

Desired tasks

Specify the number of tasks to launch.

2

► Deployment options

► Deployment failure detection

Info

Volgende Network Settings:

▼
Networking

VPC
Info

Choose the Virtual Private Cloud to use.

vpc-02a215a1a31665035
pe-multi-r0896010-vpc

Subnets

Choose the subnets within the VPC that the task scheduler should consider for placement.

Choose subnets
Clear current selection

subnet-0605cceb8e00b302
private-subnet-dbtier-AZ2-pe-multi-r0896010
us-east-1b
10.0.176.0/20

subnet-07b2e4bcd7ed82f0
private-subnet-dbtier-AZ1-pe-multi-r0896010
us-east-1a
10.0.160.0/20

Security group
Info

Choose an existing security group or create a new security group.

☒ Use an existing security group
☐ Create a new security group

Security group name

Choose an existing security group.

Choose security groups

sg-0f36cd59537cda2b5
ALB-SG-CONTAINERS-r0896010

Public IP
Info

Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).

☒ Turned on

vervolgens gaan we de **ALB** toevoegen.

Load balancer

Load balancer type
Info

Configure a load balancer to distribute incoming traffic across the tasks running in your service.

Application Load Balancer

Application Load Balancer

Specify whether to create a new load balancer or choose an existing one.

☐ Create a new load balancer
☒ Use an existing load balancer

Load balancer

Select the load balancer you wish to use to distribute incoming traffic across the tasks running in your service.

ALB-r0896010

Health check grace period
Info

0

seconds

Container

Choose container to load balance

crud-r0896010 80:80

Listener
Info

Specify the port and protocol that the load balancer will listen for connection requests on.

☒ Create new listener
☐ Use an existing listener

Port

80

Protocol

HTTP

Target group
Info

Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

☒ Create new target group

Target group name

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

Choose container to load balance

crud-r0896010 80:80 ▼

Listener [Info](#)

Specify the port and protocol that the load balancer will listen for connection requests on.

☐ Create new listener

☒ Use an existing listener

Listener

80:HTTP ▼

Target group [Info](#)

Specify whether to create a new target group or choose an existing one that the load balancer will use to route requests to the tasks in your service.

☐ Create new target group

☒ Use an existing target group

Target group name

container-r0896010 ▼

Health check path

/

Health check protocol

HTTP

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

▼ **Service auto scaling - optional**

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

☒ Use service auto scaling

Configure service auto scaling to adjust your service's desired count

Minimum number of tasks

The lower boundary to which service auto scaling can adjust the desired count of the service.

2

Maximum number of tasks

The upper boundary to which service auto scaling can adjust the desired count of the service.

4

Scaling policy type [Info](#)

Create either a target tracking or step scaling policy.

☒ Target tracking

Increase or decrease the number of tasks that your service runs based on a target value for a specific metric.

☐ Step scaling

Increase or decrease the number of tasks that your service runs based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.

Policy name

R0896010AutoScalePolicy

ECS service metric

ECSServiceAverageCPUUtilization ▼

Target value

70

Scale-out cooldown period

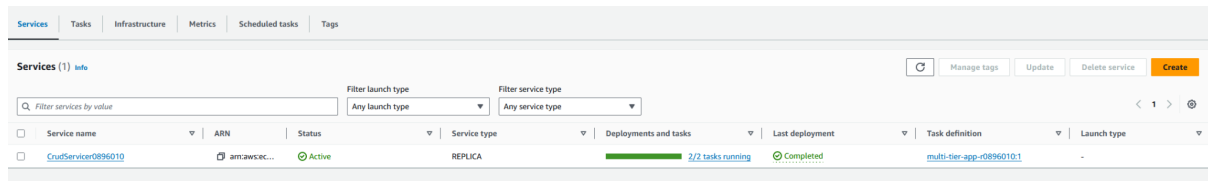
300

Scale-in cooldown period

300

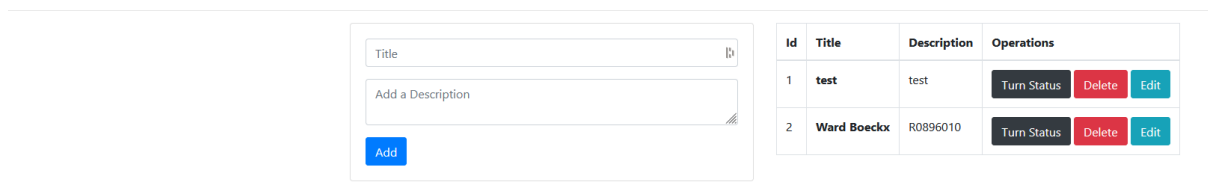
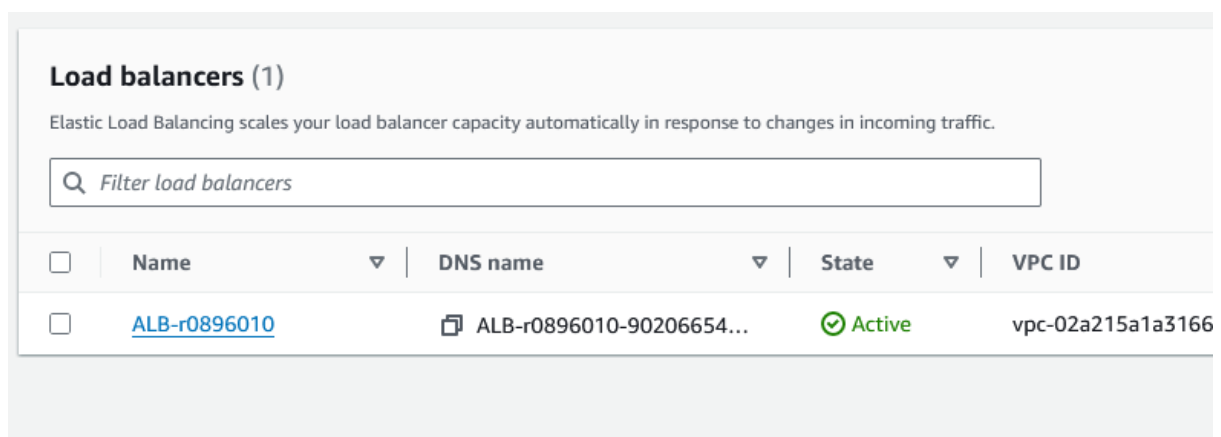
☐ Turn off scale-in

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 minimum aantal tasken en de maximum aantal. Vervolgens maken we een policy aan met waardes wanneer er aan auto scaling moet worden gedaan.

▼ **Service auto scaling - optional**

Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

☒ **Use service auto scaling**

Configure service auto scaling to adjust your service's desired count

Minimum number of tasks

The lower boundary to which service auto scaling can adjust the desired count of the service.

2

Maximum number of tasks

The upper boundary to which service auto scaling can adjust the desired count of the service.

4

Policy name

CrudServicer0896010

ECS service metric

ALBRequestCountPerTarget

Target value

70

Scale-out cooldown period

300

Scale-in cooldown period

300

☐ Turn off scale-in

+ Add more scaling policies

BOODSCHAP

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

Ward Boeckx - R0896010