

Konténerizált webalkalmazás Azure SQL Adatbázissal, terheléselosztással

Bedő Sebestyén Péter

Infrastruktúra leírása

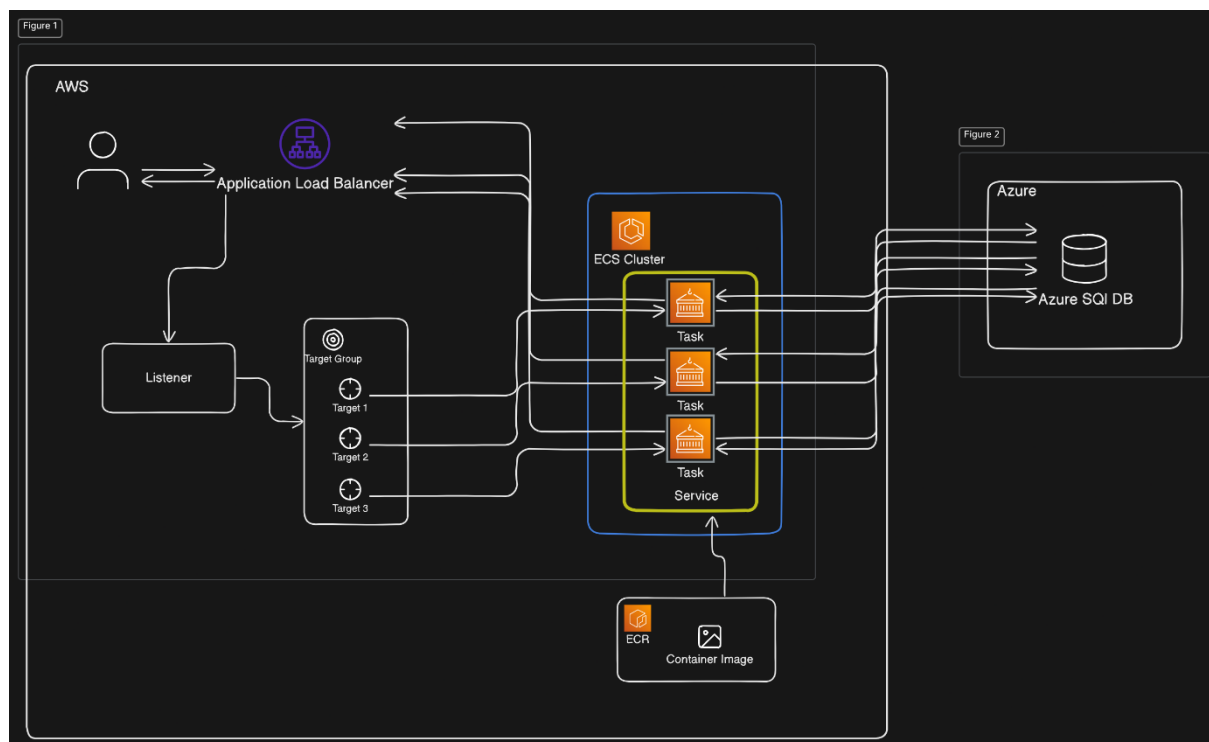
A feladat elvégzéséhez először írtam egy nagyon egyszerű webalkalmazást, amivel egy adatbázison tudok CRUD műveleteket végrehajtani. Az adatbázis az Azure SQL Database szolgáltatásban tároltam el. Az adatbázishoz egy Connection String segítségével tudtam a csatlakoztatni az alkalmazást.

Az alkalmazásból készült Docker Image-t az AWS Elastic Container Registry szolgáltatásban tároltam el. Ezt az Image-t használja fel későbbiekben az ECS Cluster-ben létrehozott Service a Task-ok létrehozására. A Service-hez hozzárendelhető Load Balancer szolgáltatás. Én ebben az esetben Application Load Balancer-t használtam.

A felhasználó kérései először a Load Balancer-t érik el, amely ezután elosztja a forgalmat a Target Taskok között. A Load Balancer és a Target-ek között egy Listener van, aminek feladata, hogy a kérést szabályok alapján a megfelelő Target-hez irányítsa. A Targetek itt egy-egy Taskot (azaz konténert) jelképeznek.

Ezután a konténerek feldolgozzák a kérést, változtatásokat hajtanak végre az Azure SQL Adatbázison, majd választ küld a Load Balancer-nek ami továbbítja a választ a felhasználónak.

Infrastruktúra felépítése diagramon ábrázolva



Infrastruktúra létrehozása

Az adatbázist Azure SQL Adatbázisban tároltam. Az adatbázist egy Connection String-el tudtam az alkalmazáshoz kötni.

```

{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "CRUDAppContext": "Server=tc-userdbsqlserver.database.windows.net,1433;Initial Catalog=userdb;Persist Security Info=False;User ID=[redacted];Password=[redacted];MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;"
  }
}

```

Ahhoz, hogy csatlakozni tudjak különböző eszközökről az adatbázishoz engedélyeznem kellett az eszközök IP címeit az adatbázis tűzfalán.

Public Endpoints allow access to this resource through the internet using a public IP address. An application or resource that is granted access with the following network rules still requires proper authorization to access this resource. [Learn more](#)

Public network access

☐ Disable

☒ Selected networks

Connections from the IP addresses configured in the Firewall rules section below will have access to this database. By default, no public IP addresses are allowed. [Learn more](#)

Virtual networks

Allow virtual networks to connect to your resource using service endpoints. [Learn more](#)

+ Add a virtual network rule

Rule	Virtual network	Subnet	Address range	Endpoint status	Resource group	Subscription	State
------	-----------------	--------	---------------	-----------------	----------------	--------------	-------

Firewall rules

Allow certain public internet IP addresses to access your resource. [Learn more](#)

+ Add your client IPv4 address (89.245.24.205) + Add a firewall rule

Rule name	Start IPv4 address	End IPv4 address	
aws_ip_address_1	51.20.116.151	51.20.116.151	
aws_ip_address_2	13.48.204.141	13.48.204.141	
aws_ip_address_3	13.49.228.3	13.49.228.3	
ClientIPAddress_2024-4-24_11-39-9	89.245.24.206	89.245.24.206	
ClientIPAddress_2024-4-28_14-30-5	89.245.24.221	89.245.24.221	

Docker segítségével Docker Image-t hoztam létre a webalkalmazásból és ezt töltöttem fel az AWS Elastic Container Registry szolgáltatásába. Az ehhez szükséges parancsok az alábbi képeken láthatóak.

```

Admin@DESKTOP-SKF317N MINGW64 ~/source/repos/CRUDApp
$ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/a2d0j4j0
WARNING! Your password will be stored unencrypted in C:\Users\Admin\.docker\config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

```

```
Admin@DESKTOP-SKF317N MINGW64 ~/source/repos/CRUDApp
$ docker build -t user-crud-app .
[+] Building 50.2s (18/18) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 701B
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:6.0
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:6.0
=> [internal] load .dockerignore
=> => transferring context: 382B
=> [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:6.0@sha256:5c0d4c483ce2781fa522c96a6069d12f719f1313635cba76f9e061dd17740077
=> => resolve mcr.microsoft.com/dotnet/sdk:6.0@sha256:5c0d4c483ce2781fa522c96a6069d12f719f1313635cba76f9e061dd17740077
=> => sha256:d64f43f58a712c3a66c1c69892cc484b7761a00a9976277b8c4d0d9276 5.31kB / 5.31kB
=> => sha256:f3956a6bfeb01399125ee1170e4d17dc3120b410932f8b683cd1a15a26d4d3f 148.61MB / 148.61MB
=> => sha256:5c0d4c483ce2781fa522c96a6069d12f719f1313635cba76f9e061dd17740077 1.79kB / 1.79kB
=> => sha256:95f441c04c43c4173984a5da29a869c5622b75c757206cfa9c22ab60faebf0e 2.01kB / 2.01kB
=> => sha256:6b5acadc41f379ec987dfccfc19c83007a587a82a048296fada01c8a72c897f3 25.37MB / 25.37MB
=> => sha256:9e730d91975056f609b69cc310b48ec06916d36519204331074511feb239dd6 13.70MB / 13.70MB
=> => extracting sha256:6b5acadc41f379ec987dfccfc19c83007a587a82a048296fada01c8a72c897f3
=> => extracting sha256:f3956a6bfeb01399125ee1170e4d17dc912db410932f8b683cd1a15a26d4d3f
=> => extracting sha256:9e730d91975056f609b69cc310b48ec06916d36519204331074511feb239dd6
=> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:6.0
=> [internal] load build context
=> => transferring context: 14.86kB
=> [base 2/2] WORKDIR /app
=> [final 1/2] WORKDIR /app
=> [build 2/7] WORKDIR /src
=> [build 3/7] COPY [CRUDApp.csproj, .]
=> [build 4/7] RUN dotnet restore ".\CRUDApp.csproj"
=> [build 5/7] COPY . .
=> [build 6/7] WORKDIR /src/.
=> [build 7/7] RUN dotnet build "CRUDApp.csproj" -c Release -o /app/build
=> [publish 1/1] RUN dotnet publish "CRUDApp.csproj" -c Release -o /app/publish /p:UseAppHost=false
=> [final 2/2] COPY --from=publish /app/publish .
=> exporting to image
=> => exporting layers
=> => writing image sha256:fdic3808d2a64b456a8db3698fb6b83ccaa213ef94db7ee4e3173fd5768294e4
=> => naming to docker.io/library/user-crud-app
View build details: docker-desktop://dashboard/build/default/default/11pqoyjwuxxwh15mre1j0j9b
What's Next?
1. Sign in to your Docker account -> docker login
2. View a summary of image vulnerabilities and recommendations -> docker scout quickview
```

```
Admin@DESKTOP-SKF317N MINGW64 ~/source/repos/CRUDApp
$ docker tag user-crud-app:latest public.ecr.aws/a2d0j4j0/user-crud-app:latest
```

```
Admin@DESKTOP-SKF317N MINGW64 ~/source/repos/CRUDApp
$ docker push public.ecr.aws/a2d0j4j0/user-crud-app:latest
The push refers to repository [public.ecr.aws/a2d0j4j0/user-crud-app]
3bb99fc83225: Pushed
5f70bf18a086: Pushed
ac848927dc23: Pushed
9ab11312ee1a: Pushed
1de75ad3b618: Pushed
ee6f2f4e1db2: Pushed
5b4020ebdae3: Pushed
a28e71738952: Pushed
latest: digest: sha256:35d297c0fcdd4c619d07c3778578b2626e7b16270d1e9c8cf87f4b3d2ce86bab size: 1995
```

user-crud-app

View public listing

View push commands

Edit

Images (1)

🔄

Delete

Details

🔍 Search artifacts

< 1 > ⓘ

<input type="checkbox"/>	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
<input type="checkbox"/>	latest	Image	2024. április 26., 13:49:46 (UTC+02)	107.47	<div><div>📄 Copy URI</div></div>	<div><div>📄 sha256:35d297c0fcdd4c619d07c3778578b2...</div></div>

Az ide feltöltött image-t használta fel a későbbiekben az AWS Elastic Container Service szolgáltatása. Itt először egy Cluster-t hoztam létre, ahol meg tudtam adni, hogy hány EC2 példány álljon rendelkezésre minimum, és hány példányig skálazza fel magát, ha a terhelés azt kívánja. Ezeken a példányokon fognak futni a konténerek. Erről a folyamatról itt láthatóak a kép.

- ☒ **Amazon EC2 instances**
Manual configurations. Use for large workloads with consistent resource demands.

Auto Scaling group (ASG) [Info](#)

Use Auto Scaling groups to scale the Amazon EC2 instances in the cluster.

Create new ASG ▼

Provisioning model

Select a provisioning model for your instances

☒ **On-demand**

With on-demand instances, you pay for compute capacity by the hour, with no long-term commitments or upfront payments.

☐ **Spot**

Amazon EC2 Spot instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot instances are available at up to a 90% discount compared to on-demand prices.

Container instance Amazon Machine Image (AMI)

Choose the Amazon ECS-optimized AMI for your instance.

Amazon Linux 2 (kernel 5.10) ▼

EC2 instance type

Choose based on the workloads you plan to run on this cluster.

t3.micro

x86_64

2 vCPU 1 GiB Memory

Free tier eligible ▼

EC2 instance role

An instance role is used by Amazon EC2 instances to make AWS API requests. If you don't already have an instance IAM role created, we can create one for you.

ecsInstanceRole ▼

arn:aws:iam::409434914288:instance-profile/ecsInstanceRole

Desired capacity

Specify the number of instances to launch in your cluster.

Minimum

3

Maximum

5

SSH Key pair

If you do not specify a key pair, you can't connect to the instances via SSH unless you choose an AMI that is configured to allow users another way to log in.

ECSService ▼



Create a new key pair [↗](#)

Root EBS volume size

Azt is be lehet itt állítani, hogy ezek a példányok milyen VPC-n és milyen alhálózatokban jöhetnek létre.

▼

Network settings for Amazon EC2 instances

Info

By default Amazon EC2 instances are launched in the default subnets for your default VPC. To use the non-default VPC, specify the VPC and subnets.

VPC

Use a VPC with public and private subnets. By default, VPCs are created for your AWS account. To create a new VPC, go to the [VPC Console](#).

vpc-02cab2e34ecff4c45
default

Subnets

Select the subnets where your tasks run. We recommend that you use three subnets for production.

Choose subnets

Clear current selection

subnet-0e977ad1447db6b2f
eu-north-1c 172.31.0.0/20

subnet-0147b20e2f98837d0
eu-north-1b 172.31.32.0/20

subnet-0ce5d8f307a15df06
eu-north-1a 172.31.16.0/20

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-03163280fb2b3d7d8
default

Auto-assign public IP

Info

Choose whether to auto-assign a public IP to the Amazon EC2 instances

Use subnet setting

Amazon Elastic Container Service > Clusters

Clusters (1)

Info

Create cluster

Search clusters

< 1 >

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strategy
CRUD-cluster	1	0 Pending 3 Running	3 EC2	Container Insights	ASG

Ezután Task Definition-t hoztam létre. Itt be tudtam állítani, hogy az egyes konténereknek mekkora teljesítmény igényük legyen, valamint ide kellett belinkelni az ECR-ben tárolt image linkjét.

▼ Infrastructure requirements

Specify the infrastructure requirements for the task definition.

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](#)

default

Task size [Info](#)

Specify the amount of CPU and memory to reserve for your task.

CPU

1 vCPU

Memory

0.3 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](#).

-

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.

ECSTASKEXECUTIONROLE

▼ Task placement - *optional*

Constraint [Info](#)

Task placement constraints allow you to filter the container instances used for the placement of your tasks using built-in or custom attributes. The service scheduler first filters the container instances that match the constraints and then applies the placement strategy to place the task.

▼ Container - 1 [Info](#)

Essential container

Remove

Container details

Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name

crud-app-container

Image URI

public.ecr.aws/a2d0j4j0/user-crud-app:latest

Essential container

Yes

Private registry [Info](#)

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

☐ Private registry authentication

Port mappings [Info](#)

Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Host port

0

Container port

80

Protocol

TCP

Port name

container-port-protocol

App protocol

HTTP

Remove

Add port mapping



The bridge network mode is selected. By default, dynamic host port mappings are used for each container. To configure a static host port mapping, specify a host port for each container port.

Read only root file system [Info](#)

When this parameter is turned on, the container is given read-only access to its root file system.

☐ Read only

Resource allocation limits - *conditional* [Info](#)

Container-level CPU, GPU, and memory limits are different from task-level values. They define how much resources are allocated for the container. If container attempts to exceed the memory specified in hard limit, the container is terminated.

CPU

1

in vCPU

GPU

1

Memory hard limit

3

in GB

Memory soft limit

1

in GB

crud-app-task (2) Info		Refresh	Deploy	Actions	Create new revision
<input type="text" value="Filter task definition revisions by value"/>		Filter status	<input type="text" value="Active"/>		
<input type="checkbox"/> Task definition: revision		▼		Status	
<input type="checkbox"/> crud-app-task7				ACTIVE	

Ezekután létrehozhattam egy Service-t a már létrehozott Cluster-en belül. Ezen folyamat során ki tudtam választani, hogy milyen Task-ot szolgáltasson a service és a Load Balancer-t is itt tudtam hozzáfűzni az infrastruktúrához.

Deployment configuration

Application type [Info](#)
Specify what type of application you want to run.

☒ **Service**
Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.

☐ **Task**
Launch a standalone task that runs and terminates. For example, a batch job.

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:

Service name
Assign a unique name for this service.

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.

► **Deployment options**

► **Deployment failure detection** [Info](#)

▼ Load balancing - *optional*

Configure load balancing using Amazon Elastic Load Balancing to distribute traffic evenly across the healthy tasks in your service.

Load balancer type [Info](#)

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

Application Load Balancer ▼

Container

The container and port to load balance the incoming traffic to

crud-app-container 80:80 ▼

Host port:Container port

Load balancer

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

CRUDAppLoadBalancer ▼

Health check grace period [Info](#)

5

seconds

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 ▼

Listener rules for 80:HTTP [🔗](#) (1)

Traffic received by the listener is routed according to its rules. Rules are evaluated in priority order, from the lowest value to the highest value. The default rule is evaluated last.

< 1 >

Evaluation order ▼

Rule path ▼

Target group ▼

default

/

[CRUDAppTargetGroup](#) [🔗](#)

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

CRUDAppTargetGroup ▼

Health check path

/

Health check protocol

HTTP

Amazon Elastic Container Service > Clusters > CRUD-cluster > Services > crud-app-service > Health

crud-app-service Info

[Refresh](#) [Update service](#) [Delete service](#)

[Health and metrics](#) [Tasks](#) [Logs](#) [Deployments](#) [Events](#) [Configuration and networking](#) [Tags](#)

Status Info

ARN: [arn:aws:ecs:eu-north-1:409434914288:service/CRUD-cluster/crud-app-service](#) Status: Active Tasks (3 Desired): 0 Pending | 3 Running

Deployments current state: 3 Completed Health check grace period: 5 seconds

▼ **Load balancer health**

(Application Load Balancer) [CRUDAppLoadBalancer](#) [View load balancer](#)

Listener protocol:port: [HTTP:80](#) Target group name:protocol: [CRUDAppTargetGroup:HTTP](#) Health check path: [/](#) Targets (3 total): 3 Healthy 0 Unhealthy

Health

A Load Balancer létrehozásáról itt láthatóak a képek.

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**
Includes only IPv4 addresses.

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

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

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](#).

-
vpc-02cab2e34ecff4c45
IPv4 VPC CIDR: 172.31.0.0/16



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.

☒ eu-north-1a (eun1-az1)

Subnet

subnet-0ce5d8f307a15df06

IPv4 address

Assigned by AWS

☒ eu-north-1b (eun1-az2)

Subnet

subnet-0147b20e2f98837d0

IPv4 address

Assigned by AWS

☒ eu-north-1c (eun1-az3)

Subnet

subnet-0e977ad1447db6b2f

IPv4 address

Assigned by AWS

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



ApplicationLoadBalancer

sg-0b9d114a464d323e8 VPC: vpc-02cab2e34ecff4c45



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

Forward to

Select a target group

Create target



In use

CRUDAppTargetGroup
Target type: Instance, IPv4

HTTP

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources.

Add listener tag

You can add up to 50 more tags.

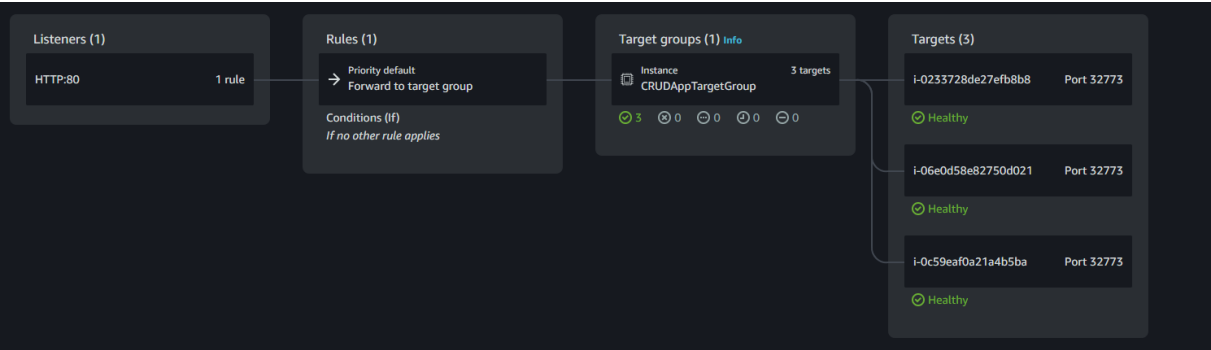
Add listener

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

<input checked="" type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
<input checked="" type="checkbox"/>	CRUDAppLoadBalancer	CRUDAppLoadBalancer-10...	Active	vpc-02cab2e34ecff4c45	3 Availability Zones	application	April 28, 2024, 12:52 (UTC+02:00)



A Target Group mutat azokra az EC2 példányokra, amelyek futtatják a konténereket. Az alábbi képen jól látható, hogy mindegyik elérhető a Load Balancer számára ez is azt jelzi, hogy megfelelően működik a szolgáltatás.

CRUDAppTargetGroup

Actions

Details

arn:aws:elasticloadbalancing:eu-north-1:409434914288:targetgroup/CRUDAppTargetGroup/98a33640ec85fc77

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-02cab2e34ecff4c45
IP address type IPv4	Load balancer CRUDAppLoadBalancer		
3 Total targets	3 Healthy 0 Anomalous	0 Unhealthy	0 Unused
		0 Initial	0 Draining

Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets

Monitoring

Health checks

Attributes

Tags

Registered targets (3)

Anomaly mitigation: Not applicable

Deregister

Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection result
<input type="checkbox"/>	i-0c59eaf0a21a4b5ba	ECS Instance - ...	32773	eu-north-1b	Healthy	-	April 28, 2...	Normal
<input type="checkbox"/>	i-06e0d58e82750d021	ECS Instance - ...	32773	eu-north-1c	Healthy	-	April 28, 2...	Normal
<input type="checkbox"/>	i-0233728de27efb8b8	ECS Instance - ...	32773	eu-north-1a	Healthy	-	Anril 28. 2...	Normal

Magát az alkalmazást a Load Balancer DNS címén keresztül érhetjük el.

Index - CRUDApp

crudapploadbalancer-1074175019.eu-north-1.elb.amazonaws.com/users

CRUDAppHomeUsers

Index

[Create New](#)

Name	Email
sebi	sebi@gmail.com
b	b
c	c
d	d
f	f