

1 Account setup & IAM configuration

1.1 Enable Multi-Factor Authentication (MFA) for root account

done

1.2 Create an administrative IAM user

done

1.3 Connect to AWS using the CLI

I configured the AWS CLI in the terminal with the command `aws configure`. After running this command, I entered my AWS Access Key ID, AWS Secret Access Key, AWS Session Token and the default region.

1.4 Set up billing alerts

Budgets (1) Info									
<input type="text"/> Find a budget		Type - Show all budgets							
<input type="checkbox"/>	Name	▲ Thresholds	▼ Health status	Billing View	Budget	Amount used	Forecasted amount	Current vs. budgeted	Forecast
<input type="checkbox"/>	monthly-7usd	OK	Healthy	Primary View	\$7.00	-	-	-	0.00%

2 Amazon S3 setup and file management

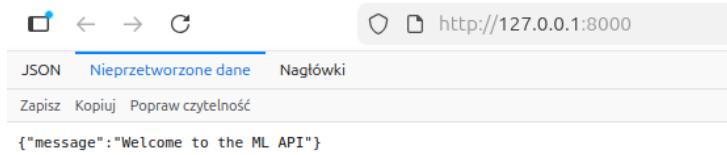
2.1 Create an S3 bucket and 2.2 Upload files to S3 bucket

Files and folders									
Configuration									
Files and folders (1 total, 79.9 MB)									
Find by name									
Name	Folder	Type	Size	Status	Error				
model.zip	-	application/zip	79.9 MB	Succeeded	-				

2.3 Integrate S3 into an ML workflow

```
(lab_9MLProject) jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject$ python downloads.py
downloading s3://mllops-lab9-jacek-01/model.zip to lab_1MLHomework/models/models/model.zip
download finished
extracting lab_1MLHomework/models/models/model.zip to lab_1MLHomework/models
extraction finished
removed ZIP file: lab_1MLHomework/models/models/model.zip
```

```
✓ lab_1mlhomework-ml-app      Built
0.0s
✓ Network lab_1mlhomework_default    Created
0.0s
✓ Container lab_1mlhomework-ml-app-1  Created
0.1s
Attaching to ml-app-1
ml-app-1  | Downloading mypy (13.0MiB)
ml-app-1  | Downloading ruff (13.4MiB)
ml-app-1  | Downloaded mypy
ml-app-1  | Downloaded ruff
ml-app-1  | Installed 7 packages in 65ms
ml-app-1  | INFO:    Started server process [32]
ml-app-1  | INFO:    Waiting for application startup.
ml-app-1  | INFO:    Application startup complete.
ml-app-1  | INFO:    Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```



3 Elastic Container Registry (ECR) & Docker management

3.1 Create an ECR repository

Repository name	URI	Created at	Tag immutability	Encryption type
sentiment-app-lab9	471112970968.dkr.ecr.us-east-1.amazonaws.com/sentiment-app-lab9	December 08, 2025, 19:41:04 (UTC+01)	Mutable	AES-256

3.2 Authenticate your Docker client to ECR

```
jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject/lab_1MLHomework$ aws ecr get-login-password --region us-east-1 | docker login --username AWS
WARNING! Your credentials are stored unencrypted in '/home/jacek-tyszkiewicz/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
```

3.3 Build and push a Docker image to ECR

```
jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject/lab_1MLHomework$ sudo docker tag lab_1mlhomework-ml-app:latest \
471112970968.dkr.ecr.us-east-1.amazonaws.com/sentiment-app-lab9:14.01.25
```

```
jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject$ sudo docker push \
471112970968.dkr.ecr.us-east-1.amazonaws.com/sentiment-app-lab9:14.01.25
The push refers to repository [471112970968.dkr.ecr.us-east-1.amazonaws.com/sentiment-app-lab9]
910e281fc135: Pushed
bd20e3d2420a: Pushed
8e45e80a935f: Pushed
9c7def175d13: Pushed
e00fdfcad471: Pushed
e6644e81cf69: Pushed
0f40e99006df: Pushed
400af0b05c1f: Pushed
c07f79cd8299: Pushed
db419cb0c9eb: Pushed
bd2be69c2f99: Pushed
14.01.25: digest: sha256:a2420faf0f67ca4649b41dc7951c3c7b0abdb7246aac7e0a09cf3b6463abd02d size: 2635
```

4 Virtual Private Cloud (VPC) configuration

4.1 Create a new VPC

VPC ID: `vpc-0b5ff41ef06f747d5`

DNS resolution: Enabled

Main network ACL: `acl-0b36c71e43cfb0da6`

IPv6 CIDR (Network border group): -

Encryption control ID: -

Details Info

State: Available

Tenancy: default

Default VPC: No

Network Address Usage metrics: Disabled

Encryption control mode: -

Block Public Access: Off

DHCP option set: `dopt-08d953498c497b0e`

IPv4 CIDR: `10.0.0.0/16`

Route 53 Resolver DNS Firewall rule groups: Failed to load rule groups

DNS hostnames: Disabled

Main route table: `rtb-0ff23db64c15b82e3`

IPv6 pool: -

Owner ID: `471112970968`

4.2 Create public and private subnets

4.2.1 Create public subnets and 4.2.2 Create private subnets

Name	Subnet ID	Status	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID	Available IPv4 addresses	Availability Zone
private-subnet-1	<code>subnet-05109151b40963c3</code>	Available	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	Off	<code>10.0.3.0/24</code>	-	-	251	use1-ac2 (us-east-1a)
public-subnet-1	<code>subnet-0510911846963c3</code>	Available	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	Off	<code>10.0.1.0/24</code>	-	-	251	use1-ac1 (us-east-1a)
private-subnet-2	<code>subnet-0b0e185784499f97</code>	Available	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	Off	<code>10.0.4.0/24</code>	-	-	251	use1-ac2 (us-east-1b)
public-subnet-2	<code>subnet-0c539f9b4afcced4</code>	Available	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	Off	<code>10.0.2.0/24</code>	-	-	251	use1-ac2 (us-east-1b)

4.3 Set up an Internet Gateway

Internet gateway ID: `igw-0a21321a55d09f021`

Details Info

Owner: `471112970968`

4.4 Configure route tables

4.4.1 Public route table

Updated routes for `rtb-0c116f86a42a56f27` / public-route-table successfully

Details Info

Route table ID: `rtb-0c116f86a42a56f27`

VPC: `vpc-0b5ff41ef06f747d5` | sentiment-app-vpc

Main: No

Owner ID: `471112970968`

Explicit subnet associations: 2 subnets

Edge associations: -

Routes (2)

Destination	Target	Status	Propagated	Route Origin
<code>0.0.0.0/0</code>	<code>igw-0a21321a55d09f021</code>	Active	No	Create Route
<code>10.0.0.0/16</code>	local	Active	No	Create Route Table

4.4.2 Private route table

Opis prywatnej tablicy tras (na razie bez trasy internetowej).

You have successfully updated subnet associations for `rtb-077740547f95e099f` / private-route-table.

Route tables (4) Info

Last updated 1 minute ago

Create route table

Name	Route table ID	Explicit subnet assoc.	Edge associations	Main	VPC	Owner ID
-	<code>rtb-0019011196df541d4</code>	-	-	Yes	<code>vpc-05646a00948014d4</code>	<code>471112970968</code>
-	<code>rtb-0f298684413bb2e3</code>	-	-	Yes	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	<code>471112970968</code>
public-route-table	<code>rtb-0c116f86a42a56f22</code>	2 subnets	-	No	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	<code>471112970968</code>
private-route-table	<code>rtb-077740547f95e099f</code>	2 subnets	-	No	<code>vpc-0b5ff41ef06f747d5</code> sentiment-app-vpc	<code>471112970968</code>

4.5 Set up a NAT Gateway

The screenshot shows the AWS CloudFormation 'NAT gateways' page. A new NAT gateway named 'sentiment-app-nat-gw' has been created, shown in the table with its details: Name: sentiment-app-nat-gw, NAT gateway ID: nat-11c29e29445191816, Connectivity: Public, State: Available, State message: Available, Availability: Regional, Route table ID: rtb-0ce403b0ddf..., Primary public IP: 16.204.20.244, Primary private IP: 16.204.20.244, Primary network interface: vpc-0b5ff41ef06f747d5, VPC: vpc-0b5ff41ef06f747d5, and Subnet: subnet-0... . A success message at the top indicates 'Updated routes for rtb-077f40847f95e899f / private-route-table successfully'.

4.6 Security Groups

The screenshot shows the AWS CloudFormation 'Security groups' page. A security group named 'sentiment-app-lb-sg' is displayed with the following details: Security group name: sentiment-app-lb-sg, Security group ID: sg-0e14afc3e563fa57f, Owner: 471112970968, Description: Security group for ALB, and VPC ID: vpc-0b5ff41ef06f747d5. The 'Inbound rules' tab is selected, showing one rule: sgr-017d58665c5826c58, Type: IPv4, Protocol: HTTP, Port range: 80, Source: 0.0.0.0/0, and Description: -.

The screenshot shows the AWS CloudFormation 'Security groups' page. A security group named 'sentiment-app-resources-sg' is displayed with the following details: Security group name: sentiment-app-resources-sg, Security group ID: sg-0f0cf3d1477fb7c27, Owner: 471112970968, Description: Backend resources security group, and VPC ID: vpc-0b5ff41ef06f747d5. The 'Inbound rules' tab is selected, showing one rule: sgr-0d1da44fb3598d70b, Type: Custom TCP, Protocol: TCP, Port range: 8000, Source: sg-0e14afc3e563fa57f..., and Description: -.

4.7 Set up an Application Load Balancer (ALB)

4.7.1 Target Group

The screenshot shows the AWS Lambda Target Groups page for the target group 'sentiment-app-target-group'. The 'Details' section includes:

- Target type: IP
- Protocol: Port
- Protocol version: HTTP1
- VPC: vpc-0b5ff41ef06f747d5
- IP address type: IPv4
- Load balancer: None associated

The status summary table shows:

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
0	0	0	0	0	0

The 'Targets' tab is selected, showing no registered targets. A message indicates that 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.

4.7.2 Create the ALB

The screenshot shows the AWS Lambda Application Load Balancers page for the load balancer 'sentiment-app-alb'. The 'Details' section includes:

- Load balancer type: Application
- Status: Provisioning
- VPC: vpc-0b5ff41ef06f747d5
- Load balancer IP address type: IPv4
- Scheme: Internet-facing
- Hosted zone: Z355XDOTRQ7X7K
- Availability Zones: subnet-0c510915184b963c3 (us-east-1a (use1-az1)), subnet-06309f864dfcc6edd (us-east-1b (use1-az2))
- Date created: December 12, 2025, 17:16 (UTC+01:00)
- DNS name: sentiment-app-alb-2003052105.us-east-1.elb.amazonaws.com (A Record)

The 'Listeners and rules' tab is selected, showing one rule:

Protocol/Port	Default action	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store
HTTP:80	Forward to target group sentiment-app-target-group: 1 (100%)	1 rule	ARN	Not applicable	Not applicable	Not applicable

5 AWS Fargate deployment & CloudWatch configuration

5.1 Create an ECS Cluster

The screenshot shows the AWS Lambda ECS Clusters page for the cluster 'sentiment-app-cluster'. The 'Clusters' section includes:

- Cluster: sentiment-app-cluster has been created successfully.
- Introducing Express Mode: Amazon ECS Express Mode provides a simplified way to quickly launch highly available, scalable containerized applications. This streamlined experience automates infrastructure setup, including a default domain for each ECS service. [Learn more]
- Clusters (1) info
- Search clusters: sentiment-app-cluster
- Last updated: December 12, 2025, 17:59 (UTC+00:00)
- Create cluster

The 'Tasks' section shows:

Container instances	CloudWatch monitoring	Capacity provider strategy
D EC2	Default	No default found

5.2 Create a Task Definition

sentiment-app-task:1

Overview [Info](#)

ARN: arn:aws:ses:us-east-1:47112970968:task-definition/sentiment-app-task:1 Status: ACTIVE

Task role: Lambda Role Task execution role: Lambda Role

Fault injection: Turned off

Last updated: December 12, 2025, 19:26 (UTC+1:00) Deploy Actions Create new revision

Containers JSON Task placement Volumes (0) Requires attributes Tags

Task size

Task CPU: 1024 units (1 vCPU)

Task CPU maximum allocation for containers: 1024 units (1 vCPU)

CPU (units): 0 to 1000

sentiment-app-container Shared task CPU

Task memory: 5072 MB (3 GB)

Task memory maximum allocation for container memory reservation: 5072 MB (3 GB)

Memory (MB): 0 to 5000

sentiment-app-container Shared task memory

Container: sentiment-app-container [Info](#)

Image: 47112970968.us-east-1.amazonaws.com/sentiment-app-lab9:sha256:a2420fa0f067ca4649d410c79513c7b0a0db7246ac7e0d50ef3b6463ab92d Private registry: Trained on

CPU: 0 Memory hard/soft limit: 3 GB/- GPU: -

Environment and secrets Network settings Security and permissions Lifecycle and dependencies Monitoring and logging Runtime configuration Additional settings

Environment variables (0): No environment variables. No environment variables to display.

Environment files (S3 ARN):

5.3 Create a Service

sentiment-app-cluster-221

Cluster overview

ARN: arn:aws:ses:eu-north-1:80592855002:cluster/sentiment-app-cluster-2 Status: Active

CloudWatch monitoring: Default Registered container instances: -

Services

Draining: - Active: 1 Pending: - Running: 2

Services (1) [Info](#)

Filter services by value: sentiment-app-task-service-kmbafogd Filter launch type: Any launch type Filter scheduling strategy: Any scheduling strategy Filter resource management type: Any resource management type

Service name	ARN	Status	CloudWatch monitoring	Registered container instances
sentiment-app-task-service-kmbafogd	arn:aws:ses:eu-north-1:80592855002:service/sentiment-app-task-service-kmbafogd	Active	Default	-

Last updated: December 15, 2025, 01:10 (UTC+1:00) Actions Create with Express Mode

5.4 Running the service

sentiment-app-cluster-221

Cluster overview

ARN: arn:aws:ses:eu-north-1:80592855002:cluster/sentiment-app-cluster-2 Status: Active

CloudWatch monitoring: Default Registered container instances: -

Services

Draining: - Active: 1 Pending: - Running: 2

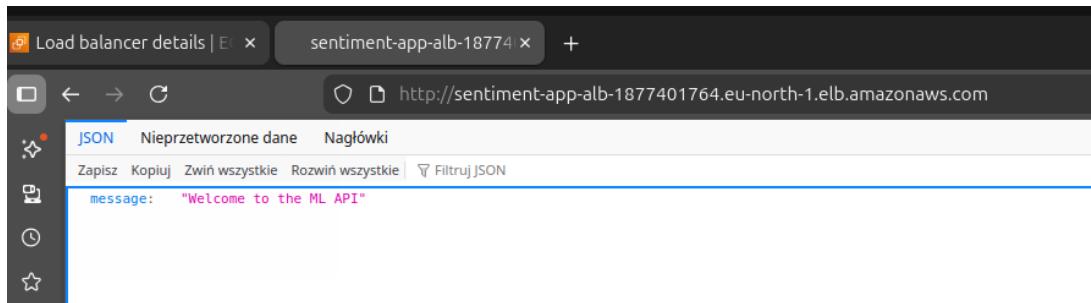
Services (1) [Info](#)

Filter services by value: sentiment-app-task-service-kmbafogd Filter launch type: Any launch type Filter scheduling strategy: Any scheduling strategy Filter resource management type: Any resource management type

Service name	ARN	Status	CloudWatch monitoring	Registered container instances
sentiment-app-task-service-kmbafogd	arn:aws:ses:eu-north-1:80592855002:service/sentiment-app-task-service-kmbafogd	Active	Default	-

Last updated: December 15, 2025, 01:16 (UTC+1:00) Actions Create with Express Mode

5.5 Access the application



6 Application testing and monitoring

6.1 Test the application

A detailed screenshot of the AWS Lambda function configuration interface, specifically focusing on the 'Test' tab. The top navigation bar shows 'GET /health Health Check'.

- Parameters:** No parameters.
- Responses:**
 - Curl:** A command-line example using curl to make a GET request to the specified URL with an accept header of application/json.
 - Request URL:** The URL for the health check: http://sentiment-app-alb-1877401764.eu-north-1.elb.amazonaws.com/health.
 - Server response:**
 - Code:** 200
 - Details:** Response body: { "status": "ok" } (with 'Download' button)
 - Response headers:** connection: keep-alive, content-length: 15, content-type: application/json, date: Mon, 06 Nov 2023 00:27:23 GMT, server: Amazon/1.0, via: 1.1 ds2 (squid), x-cache: MISS from ds2, x-cache-lookup: MISS from ds2:3120
- Responses:**

Code	Description	Links
200	Successful Response	No links

 - Media type:** application/json (selected)
 - Controls Accept header:** (button)
 - Example Value:** string
 - Schema:** (button)

POST /predict Predict

Parameters

No parameters

Request body required

application/json

Edit Value Schema

```
{
  "text": "Lecimy z tym koxem!"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://sentiment-app-alb-1877401764.eu-north-1.elb.amazonaws.com/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "text": "Lecimy z tym koxem!"
}'
```

Request URL

<http://sentiment-app-alb-1877401764.eu-north-1.elb.amazonaws.com/predict>

Server response

Code	Details	Links
200	Response body <pre>["prediction": "neutral"]</pre> Response headers <pre> connection: keep-alive content-length: 24 content-type: application/json date: Mon, 15 Dec 2025 00:30:22 GMT server: uvicorn via: 1.1 varnish (squid) x-cache: MISS from ds2 x-cache-lookup: MISS from ds2:3120 </pre>	Copy Download
200	Successful Response	No links

Media type

application/json

Controls Accept header

Example Value | Schema

default

GET / Welcome Root

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
  'http://sentiment-app-alb-1877401764.eu-north-1.elb.amazonaws.com/' \
  '-H 'accept: application/json'
```

Request URL

<http://sentiment-app-alb-1877401764.eu-north-1.elb.amazonaws.com/>

Server response

Code	Details
200	Response body <pre>{ "message": "Welcome to the ML API" }</pre> <p>Download</p> Response headers <pre>connection: keep-alive content-length: 35 content-type: application/json date: Mon, 15 Dec 2025 00:33:35 GMT server: Amazon CloudFront via: 1.1 ds2 (squid) x-cache: MISS from ds2 x-cache-lookup: HIT from ds2:3120</pre>

Responses

Code	Description	Links
200	Successful Response	No links

Media type: [application/json](#)

Controls Accept header.

[Example Value](#) | [Schema](#)

```
"string"
```

```
jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject/lab_1MLHomework$ ALB_DNS="sentiment-app-alb-1877401764.eu-north-1.elb.amazonaws.com"
jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject/lab_1MLHomework$ curl -i "http://${ALB_DNS}/health"
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2025 00:35:49 GMT
Content-Type: application/json
Content-Length: 15
Server: uvicorn
X-Cache: MISS from ds2
X-Cache-Lookup: HIT from ds2:3120
Via: 1.1 ds2 (squid)
Connection: keep-alive

{"status":"ok"}jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject/lab_1curl -i -X POST "http://${ALB_DNS}/predict" \
-H "Content-Type: application/json" \
-H "accept: application/json" \
-d '[{"text": "Lecimy z tym koksem!"}]'
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2025 00:35:57 GMT
Content-Type: application/json
Content-Length: 24
Server: uvicorn
X-Cache: MISS from ds2
X-Cache-Lookup: MISS from ds2:3120
Via: 1.1 ds2 (squid)
Connection: keep-alive

{"prediction": "neutral"}jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject/lab_1MLHomework$ []
```

	Timestamp	Message
▶	2025-12-15T00:32:07.233Z	INFO: 10.0.2.187:52314 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:32:13.921Z	INFO: 10.0.1.176:54674 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:32:37.264Z	INFO: 10.0.2.187:16862 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:32:43.951Z	INFO: 10.0.1.176:26900 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:33:07.294Z	INFO: 10.0.2.187:62556 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:33:13.979Z	INFO: 10.0.1.176:5744 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:33:35.181Z	INFO: 10.0.2.187:59396 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:33:37.325Z	INFO: 10.0.2.187:12216 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:33:44.009Z	INFO: 10.0.1.176:9576 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:34:07.356Z	INFO: 10.0.2.187:9126 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:34:14.035Z	INFO: 10.0.1.176:55740 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:34:37.386Z	INFO: 10.0.2.187:13246 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:34:44.039Z	INFO: 10.0.1.176:61320 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:35:07.387Z	INFO: 10.0.2.187:34308 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:35:14.069Z	INFO: 10.0.1.176:13058 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:35:37.391Z	INFO: 10.0.2.187:63650 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:35:44.083Z	INFO: 10.0.1.176:42508 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:35:57.514Z	INFO: 10.0.1.176:11610 - "POST /predict HTTP/1.1" 200 OK
▶	2025-12-15T00:36:07.419Z	INFO: 10.0.2.187:4266 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:36:14.113Z	INFO: 10.0.1.176:64266 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:36:37.437Z	INFO: 10.0.2.187:62726 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:36:44.124Z	INFO: 10.0.1.176:52272 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:37:07.467Z	INFO: 10.0.2.187:46270 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:37:14.155Z	INFO: 10.0.1.176:40914 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:37:37.498Z	INFO: 10.0.2.187:17182 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:37:44.176Z	INFO: 10.0.1.176:14444 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:38:07.517Z	INFO: 10.0.2.187:61634 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:38:14.198Z	INFO: 10.0.1.176:34520 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:38:37.539Z	INFO: 10.0.2.187:63034 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:38:44.210Z	INFO: 10.0.1.176:24620 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:39:07.563Z	INFO: 10.0.2.187:47494 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:39:14.223Z	INFO: 10.0.1.176:50624 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:39:37.583Z	INFO: 10.0.2.187:41328 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:39:44.253Z	INFO: 10.0.1.176:41302 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:40:07.597Z	INFO: 10.0.2.187:7742 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:40:14.269Z	INFO: 10.0.1.176:5358 - "GET / HTTP/1.1" 200 OK
▶	2025-12-15T00:40:37.615Z	INFO: 10.0.2.187:3982 - "GET / HTTP/1.1" 200 OK

All > ApplicationELB > Per AppELB, per AZ, per TG Metrics						Alarm recommends
Stockholm						
<input type="text"/> Search for any metric, dimension, resource id or account id						
RequestCount	X	Clear filters	▲ AvailabilityZone	▼ TargetGroup	▼ Metric name	▼ Alarms
<input type="checkbox"/>	LoadBalancer 4/4		eu-north-1a	targetgroup/entiment-app-tg/76d194c9696f4...	<input checked="" type="checkbox"/> RequestCount	No alarms
<input type="checkbox"/>	app/entiment-app-alb/d3d5c8f06bc9f3ed		eu-north-1b	targetgroup/entiment-app-tg/76d194c9696f4010	<input checked="" type="checkbox"/> RequestCount	No alarms
<input type="checkbox"/>	app/entiment-app-alb/d3d5c8f06bc9f3ed		eu-north-1b	targetgroup/entiment-app-tg/76d194c9696f4010	<input checked="" type="checkbox"/> RequestCountPerTarget	No alarms
<input type="checkbox"/>	app/entiment-app-alb/d3d5c8f06bc9f3ed		eu-north-1a	targetgroup/entiment-app-tg/76d194c9696f4010	<input checked="" type="checkbox"/> RequestCountPerTarget	No alarms

