

1 Account setup & IAM configuration

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.

2 Amazon S3 setup and file management

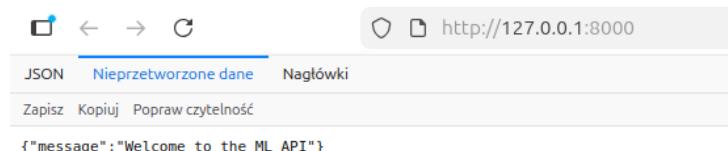
2.1 Create an S3 bucket and 2.2 Upload files to S3 bucket

The screenshot shows the AWS S3 'Files and folders' view. It displays one item: 'model.zip' (1 total, 79.9 MB). The file is of type 'application/zip' and has a status of 'Succeeded'. There are filters for 'Name' and 'Type', and columns for 'Name', 'Folder', 'Type', 'Size', 'Status', and 'Error'.

2.3 Integrate S3 into an ML workflow

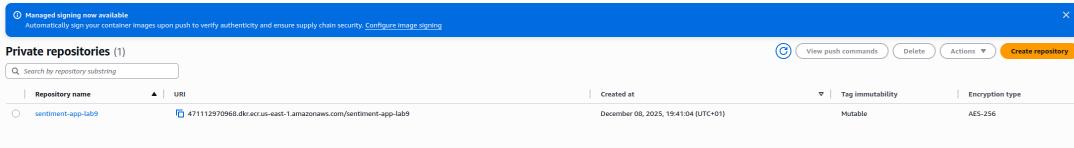
```
(lab_9MLProject) jacek-tyszkiewicz@jacek-tyszkiewicz-Nitro-AN16-41:~/lab_9MLProject$ python downloads.py
downloading s3://mllops-lab9-jacek-01/model.zip to lab_1MLHomework/models/models/model.zip
download finished
extracting lab_1MLHomework/models/models/model.zip to lab_1MLHomework/models
extraction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



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-0ff23d864c15b82e3`

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-077740b47f95e099f` / 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-0f29b864c13bb2e3</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-077740b47f95e099f</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. It is set to 'Public' connectivity, is 'Available', and is regional. Its route table ID is 'rtb-0ce403b0ddf...', primary public IP is '16.204.20.244', and it is associated with VPC 'vpc-0b5ff41ef06f747d5'. A success message indicates that routes have been updated for the private route table.

4.6 Security Groups

The screenshot shows the AWS CloudFormation 'Security groups' page for a security group named 'sentiment-app-lb-sg'. It has a description of 'Security group for ALB' and is associated with VPC 'vpc-0b5ff41ef06f747d5'. It contains one inbound rule allowing traffic from 0.0.0.0/0 on port 80 to the security group rule ID 'sgr-017d58665c5826c58' via HTTP.

The screenshot shows the AWS CloudFormation 'Security groups' page for a security group named 'sentiment-app-resources-sg'. It has a description of 'Backend resources security group' and is associated with VPC 'vpc-0b5ff41ef06f747d5'. It contains one inbound rule allowing traffic from the security group 'sg-0e14afc3e563fa57f' on port 8000 to the security group rule ID 'sgr-0d1da44fb3598d70b' via Custom TCP.

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

The 'Targets' section displays the following status summary:

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

Below the summary, there are tabs for Targets, Monitoring, Health checks, Attributes, and Tags. The Registered targets section shows 0 registered 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' section shows 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 Clusters page for the cluster 'sentiment-app-cluster'. The 'Clusters' section shows:

- Cluster: sentiment-app-cluster
- Services: 0
- Tasks: No tasks running
- Container instances: 0 EC2
- CloudWatch monitoring: Default
- Capacity provider strategy: No default found

The 'Clusters' section also includes a 'Create cluster' button.

5.2 Create a Task Definition

The screenshot shows the AWS Task Definition creation page for 'sentiment-app-task:1'. At the top, a green banner indicates 'Task definition successfully created' with the message 'sentiment-app-task:1 has been successfully created. You can use this task definition to deploy a service or run a task.' Below the banner, the task details are displayed:

- ARN:** arn:aws:ecs:us-east-1:147112970968:task-definition/sentiment-app-task:1
- Status:** ACTIVE
- Time created:** December 12, 2025, 19:26 (UTC+1:00)
- App environment:** Fargate
- Task role:** LabRole
- Task execution role:** LabRole
- Operating system/Architecture:** Linux/X86_64
- Network mode:** awsvpc

The 'Containers' tab is selected, showing one container named 'sentiment-app-container'. The container configuration includes:

- Task CPU:** 1024 units (1 vCPU)
- Task CPU maximum allocation for containers:** 1000 units
- Task memory:** 5072 MB (5 GB)
- Task memory maximum allocation for container memory reservation:** 3000 MB
- Image:** 471112970968.dkr.ecr.us-east-1.amazonaws.com/sentiment-app-lab9@sha256:a2420fa0f67cad649b41dc7951c3c7b0abdb7246ac7e0a9cf5b6463ab902d
- Private registry:** Turned off
- Memory hard/soft limit:** 5 GB/-
- GPU:** -

The 'Environment and secrets' tab is selected, showing no environment variables or secrets.

5.3 Create a Service

The screenshot shows the AWS Service creation page. At the top, a red banner displays an error message: 'There was an error while describing a list of clusters.' followed by 'Encountered 1 error' and a detailed error log: 'User: arn:aws:sts::471112970968:assumed-role/vocabs/USER4047799=virtual@student.agh.edu.pl is not authorized to perform: ecs:ListClusters on resource: * with an explicit deny in an identity-based policy'.

A note about 'Introducing Express Mode' is present, stating: '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'.

The 'Clusters' tab is selected, showing 'No clusters' and a 'Create cluster' button.

5.4 Running the service

Impossible

5.5 Access the application

Impossible

6 Application testing and monitoring

6.1 Test the application

Impossible