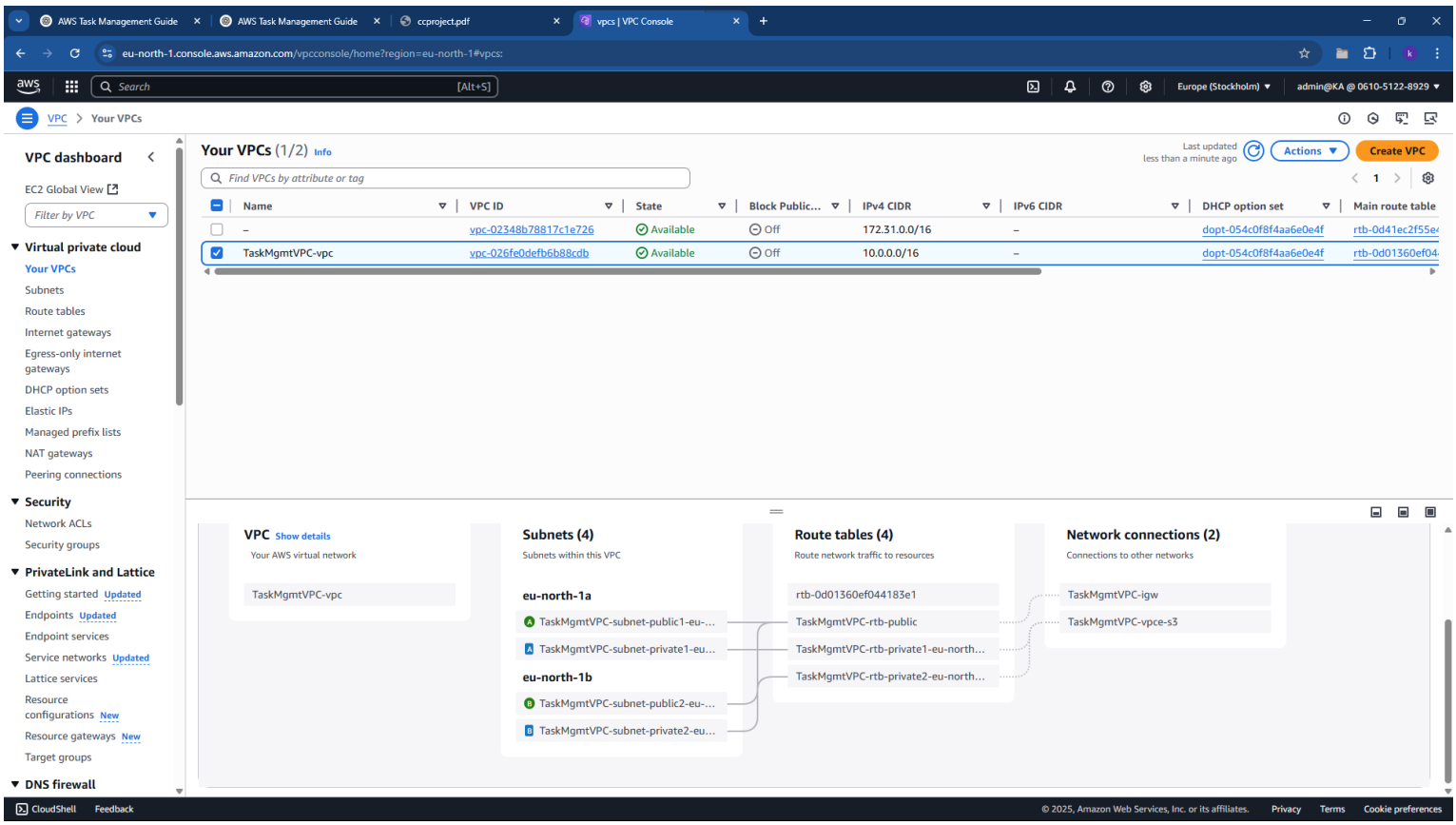
🛠 Phase 1: **Core AWS Infrastructure Setup**

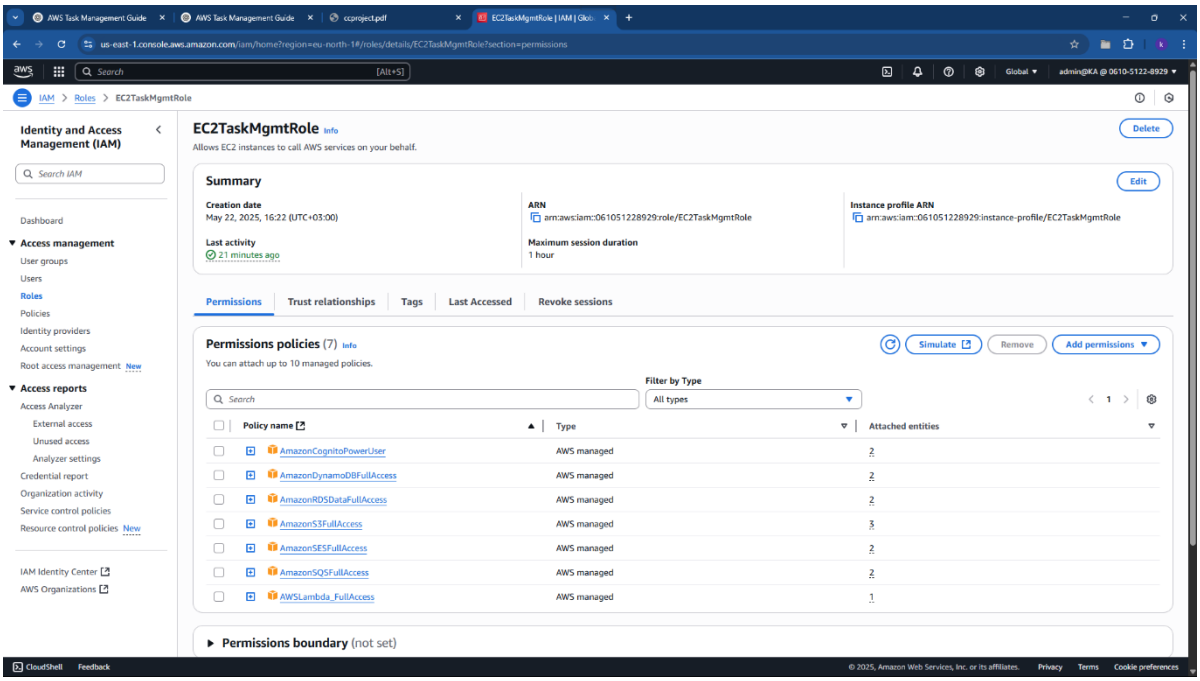
☁ Set up VPC

- **1 Public Subnet** (for EC2).
- **1 Private Subnet** (for databases).
- Configure routing tables and an Internet Gateway.

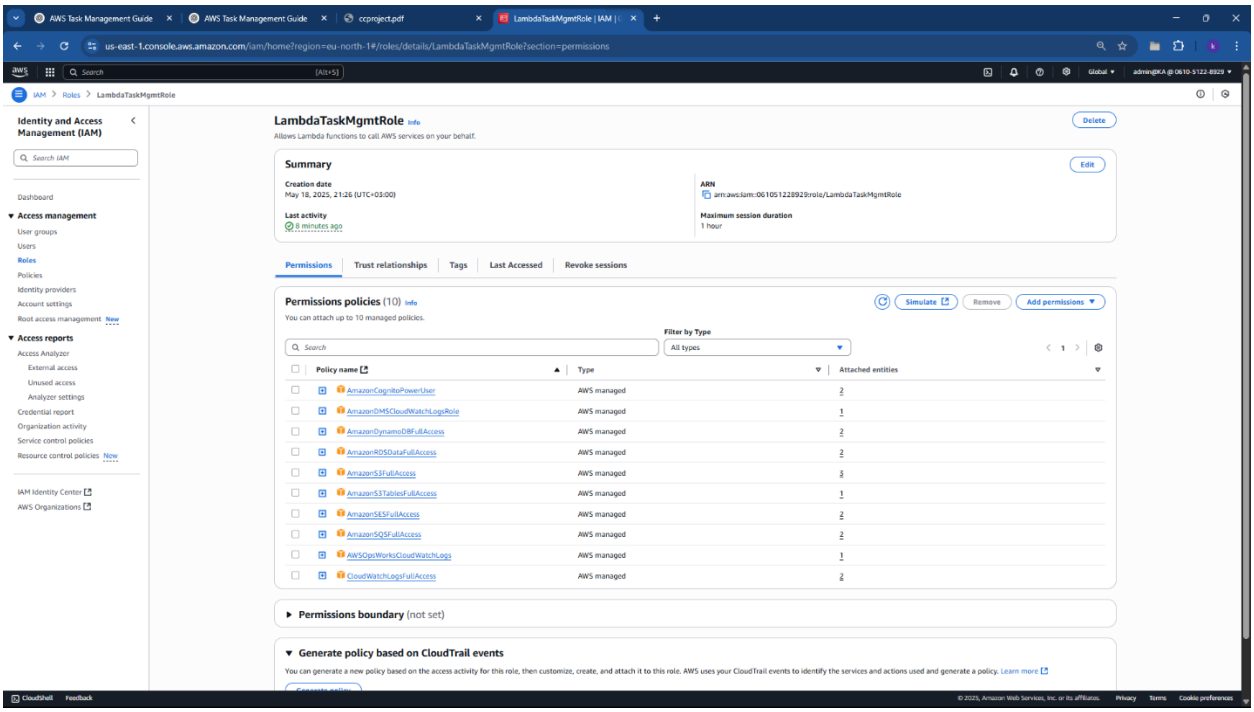


Set up IAM Roles/Policies

EC2 Role:



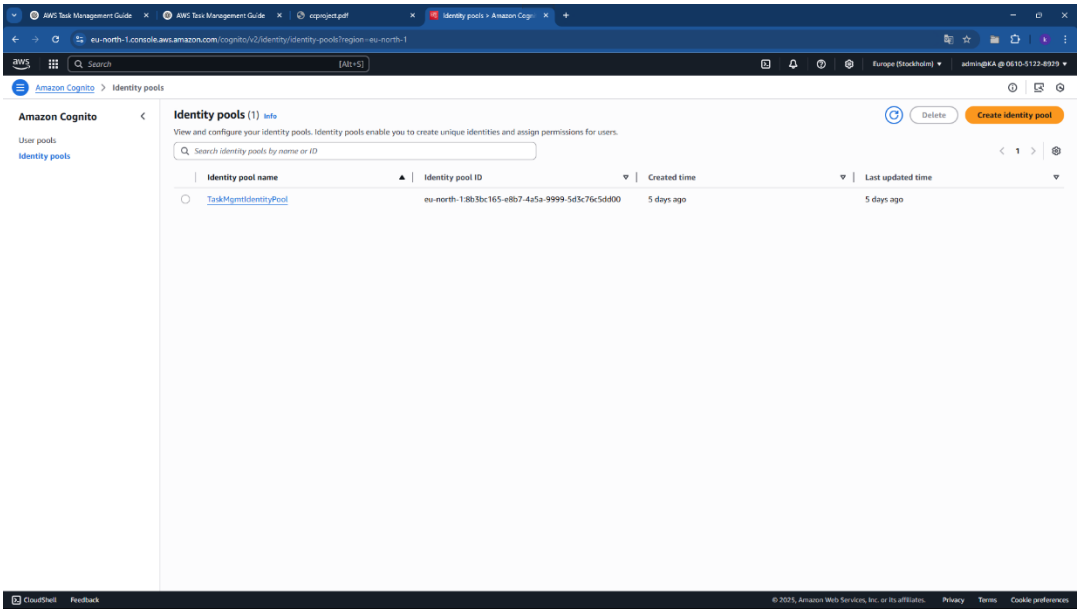
Lambda Role:



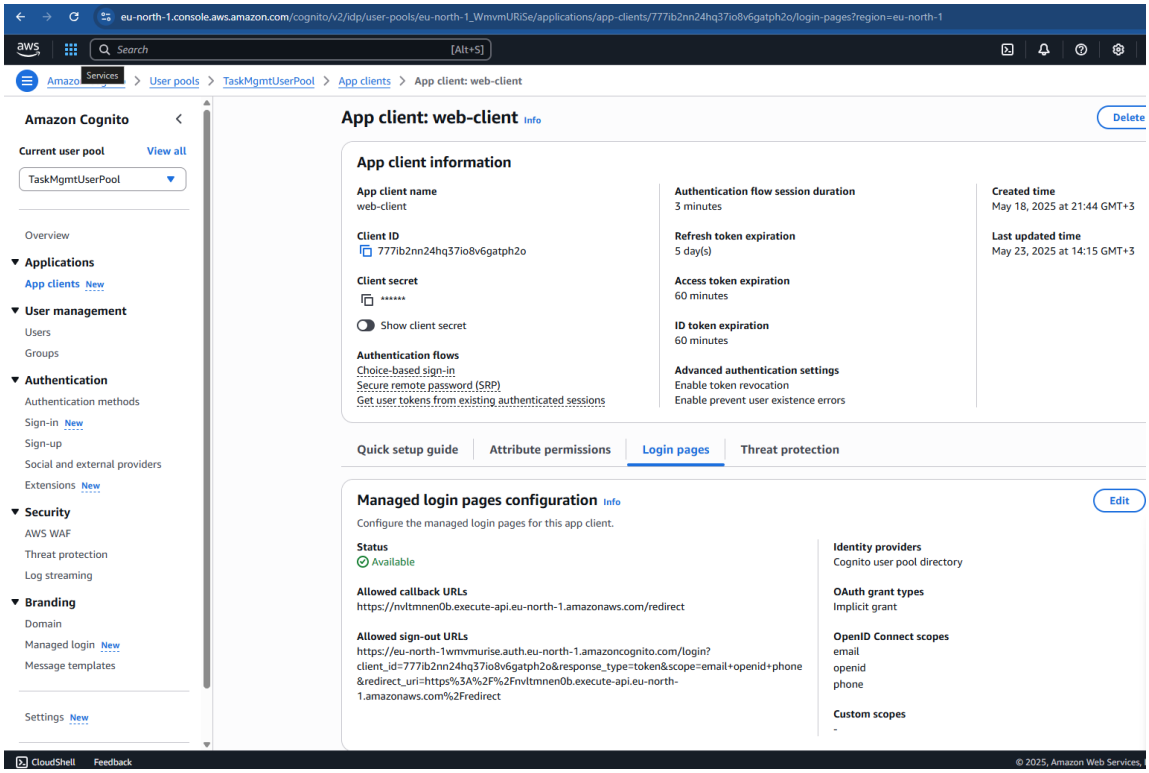
Phase 2: **Authentication and Authorization**

Cognito for User Management

-create an identity pool to give temporary AWS credentials to users



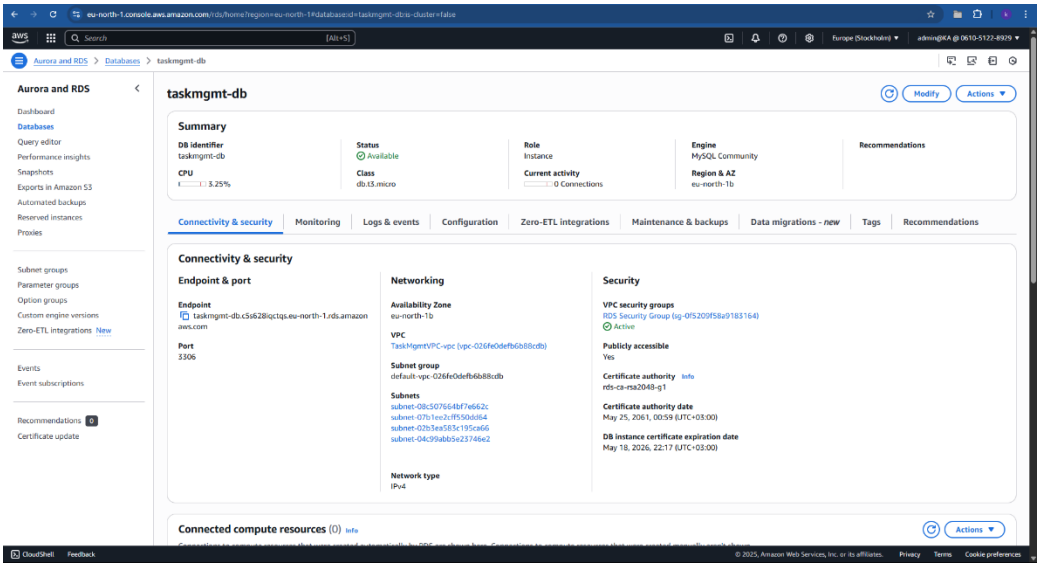
-create a user pool with a hosted UI and an app client



Phase 3: **Data Management**

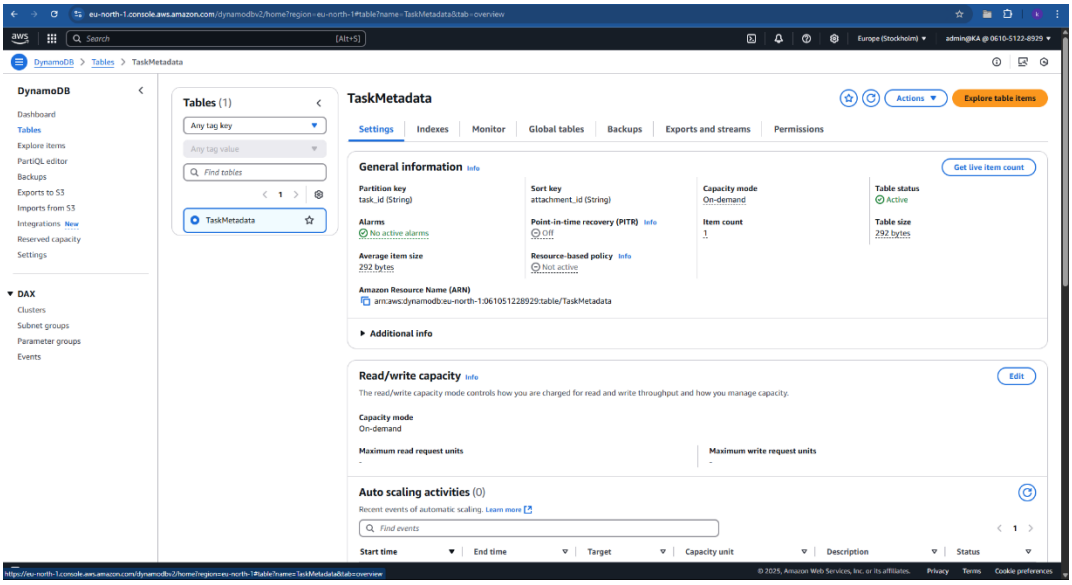
Amazon RDS

Create a MySQL database that stores user profiles and task relationships and security group rules for allowing inbound connection



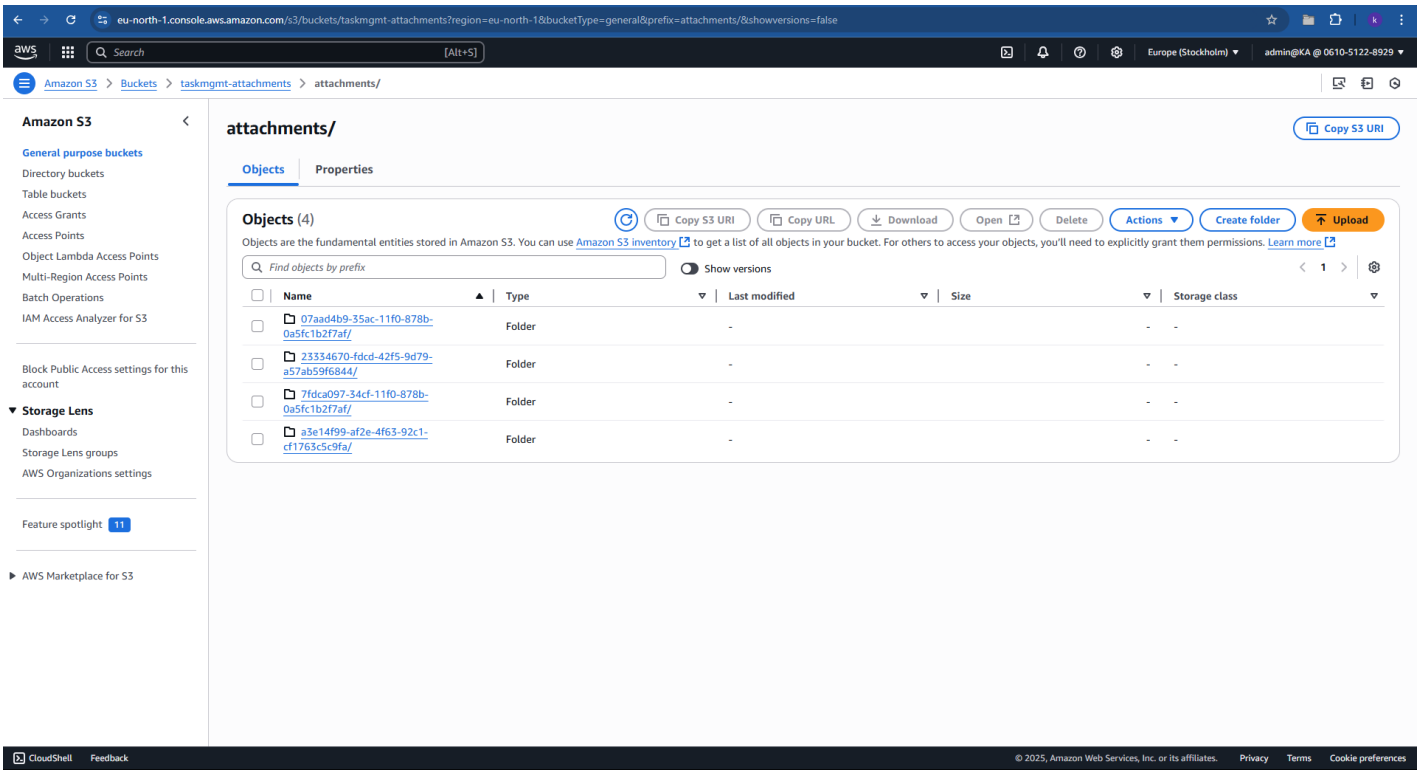
Dynamo DB

Store task metadata including attachment's s3 key and name for retrieving them from the s3 bucket they are stored in



Amazon S3

Create a bucket for file attachments.

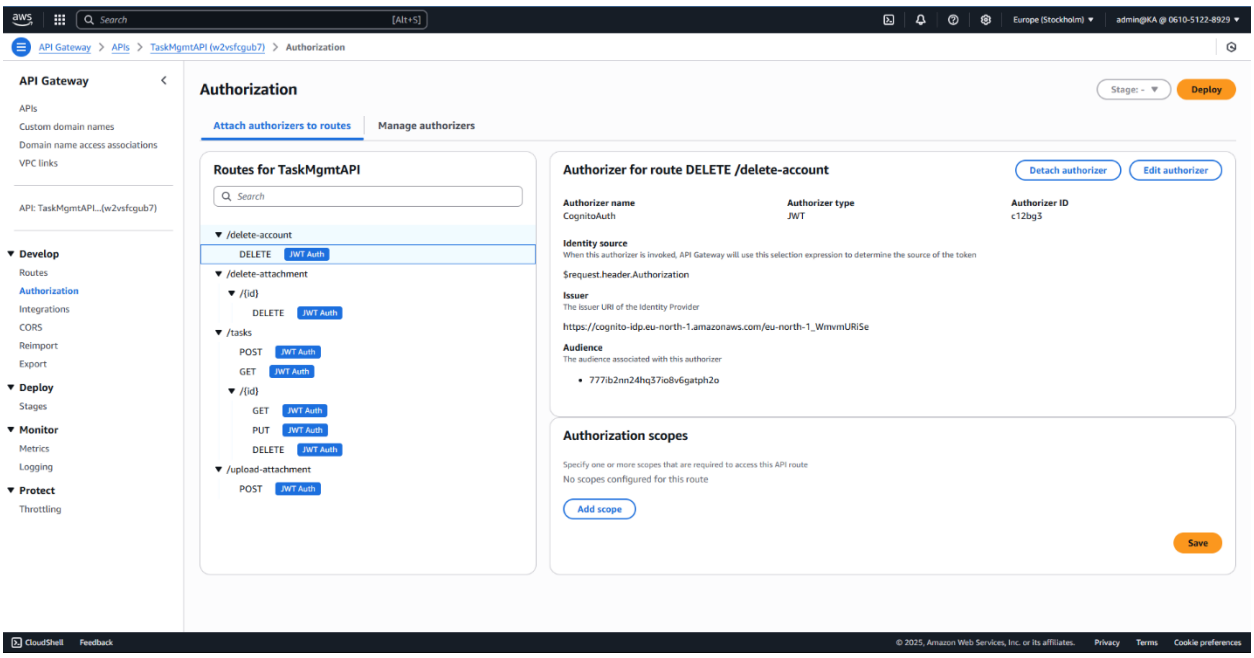


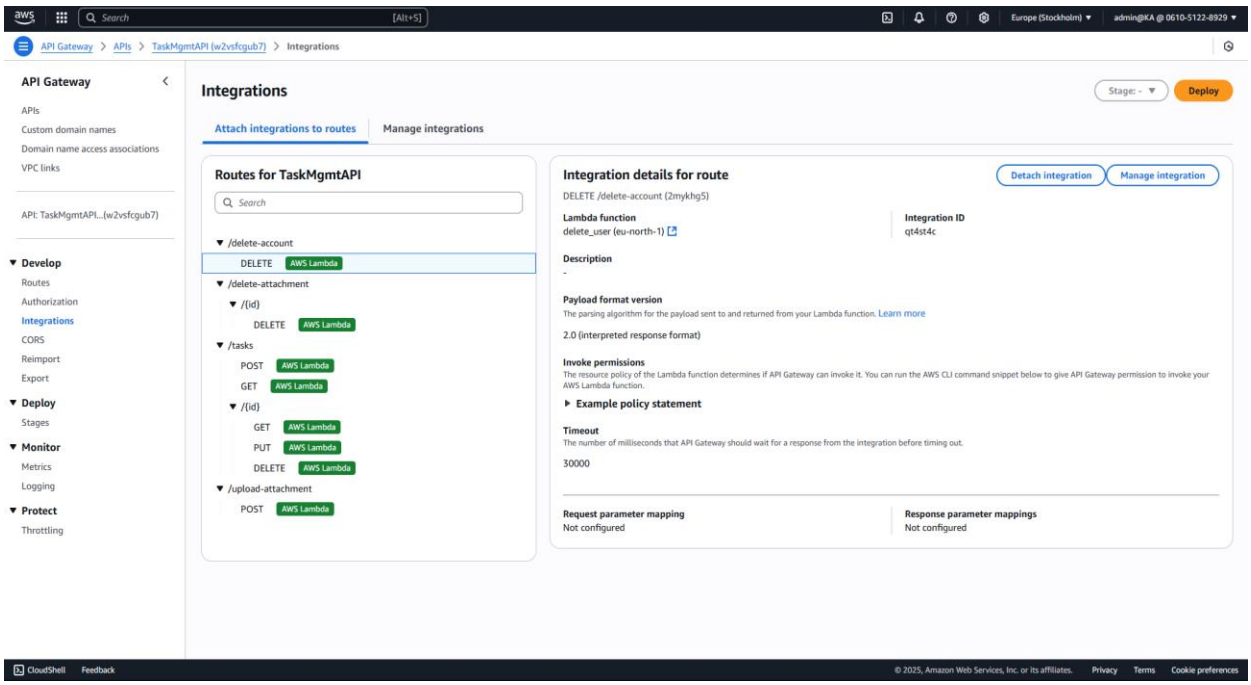
Phase 4: **Backend Logic**

API Gateway

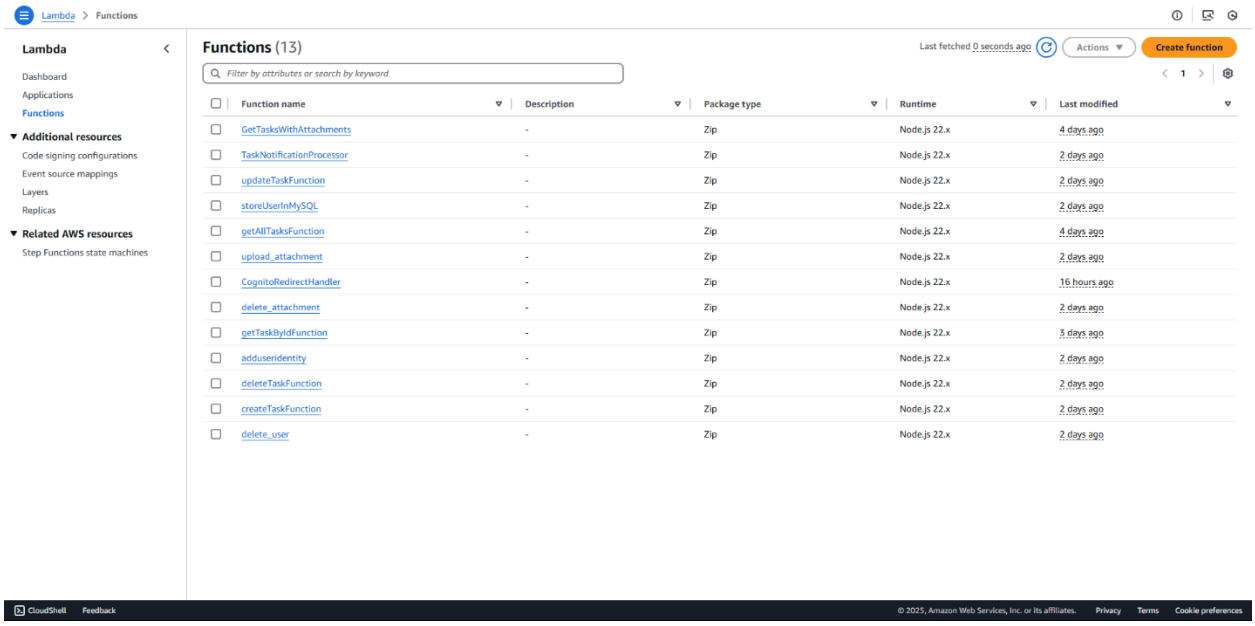
Create an Api gateway with each Api configured with authorization (Cognito authorizer) and integrations to trigger lambda functions when Api is called.

Configure CORS: Access-Control-Allow-Origin, Access-Control-Allow-Headers, Access-Control-Allow-Methods





Lambda Functions

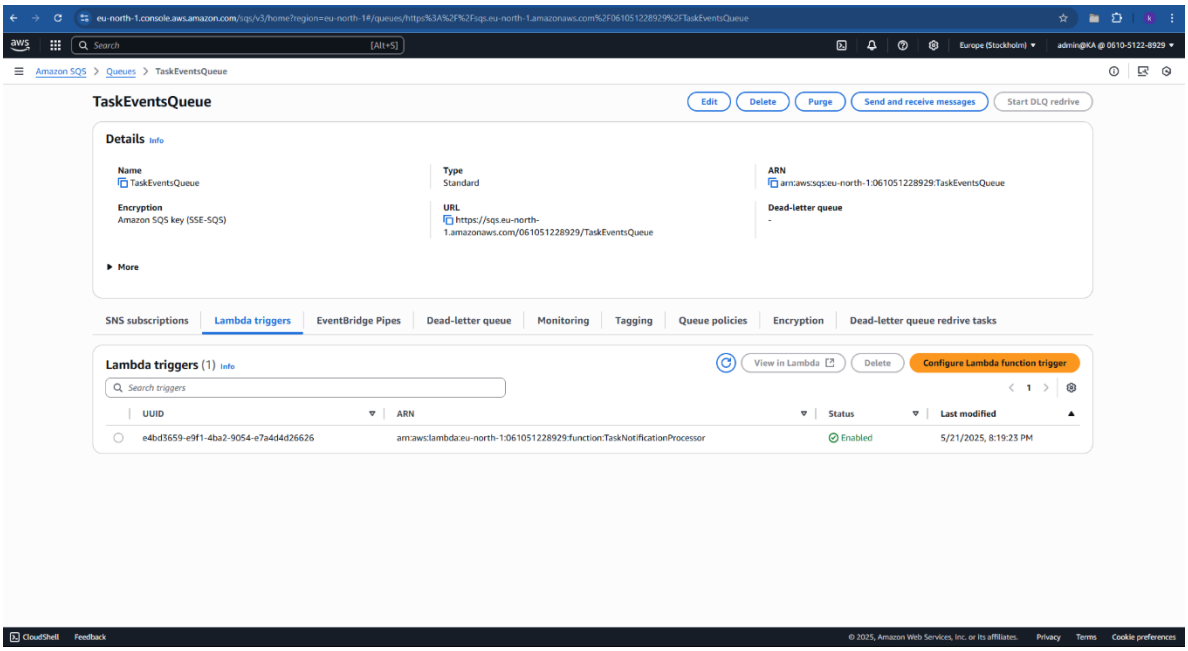


- Task operations (CRUD).
- Handle RDS/DynamoDB queries via AWS SDK.
- Handle S3 file uploads/deletions.
- Publish messages to SQS on task updates.
- Manage Notifications triggering.
- handle syncing data between all services.

Phase 5: **Notifications and Asynchronous Processing**

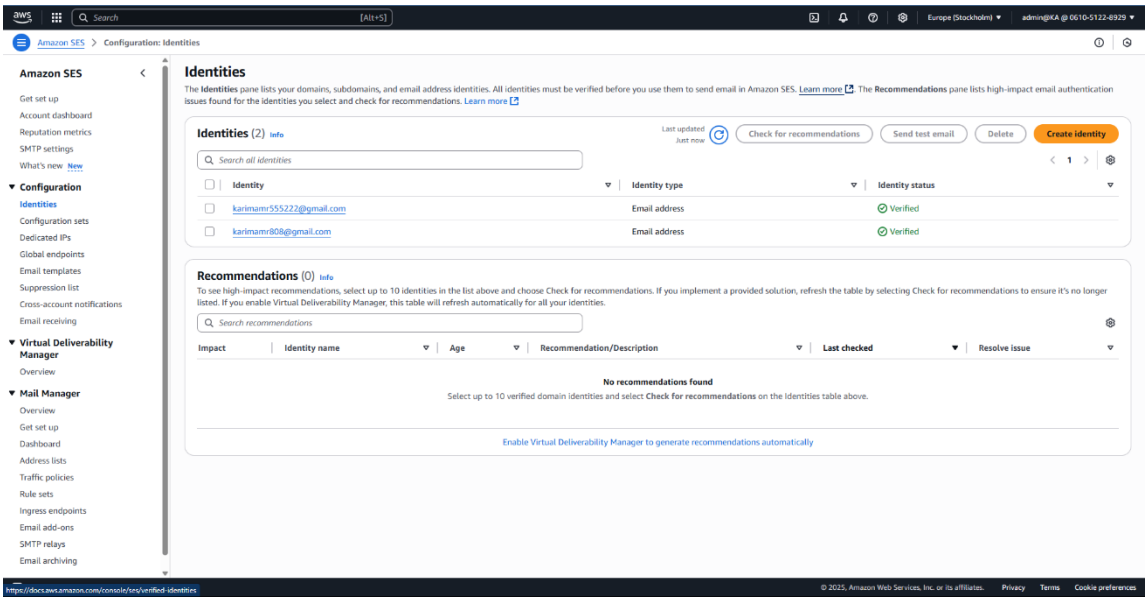
Amazon SQS

Create a queue for task update notifications, Lambda function reads messages and sends emails (via SES).



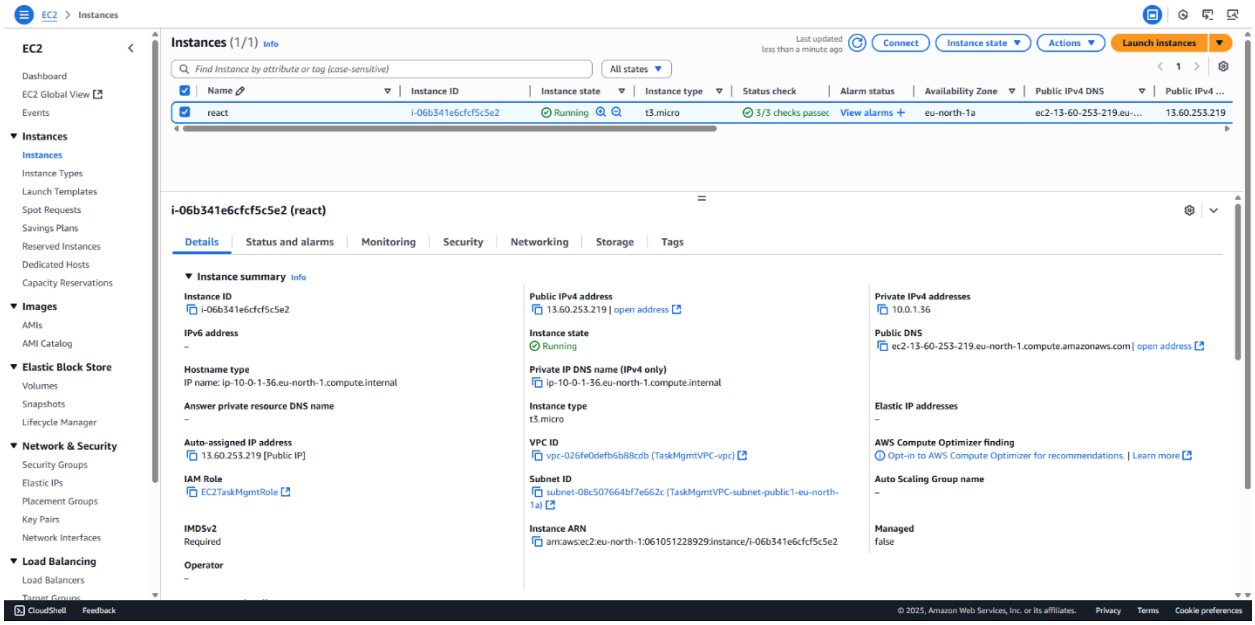
Amazon SES

create SES identity to send emails to users



Phase 6: **Frontend and Hosting**

EC2 instance



- created an ec2 instance to host the react frontend I created
- github containing the frontend code: https://github.com/KarimAmr15/AWS-TaskManagment

after connecting to the ec2 instance using SSH:

1. Prepare React App (run build)
2. Install NGINX
3. Move files to web root
4. Edit NGINX config (/etc/nginx/nginx.conf) to handle the react app routing correctly
5. Restart NGINX

📊 Phase 7: **Monitoring and Logging**

🔹 Amazon CloudWatch

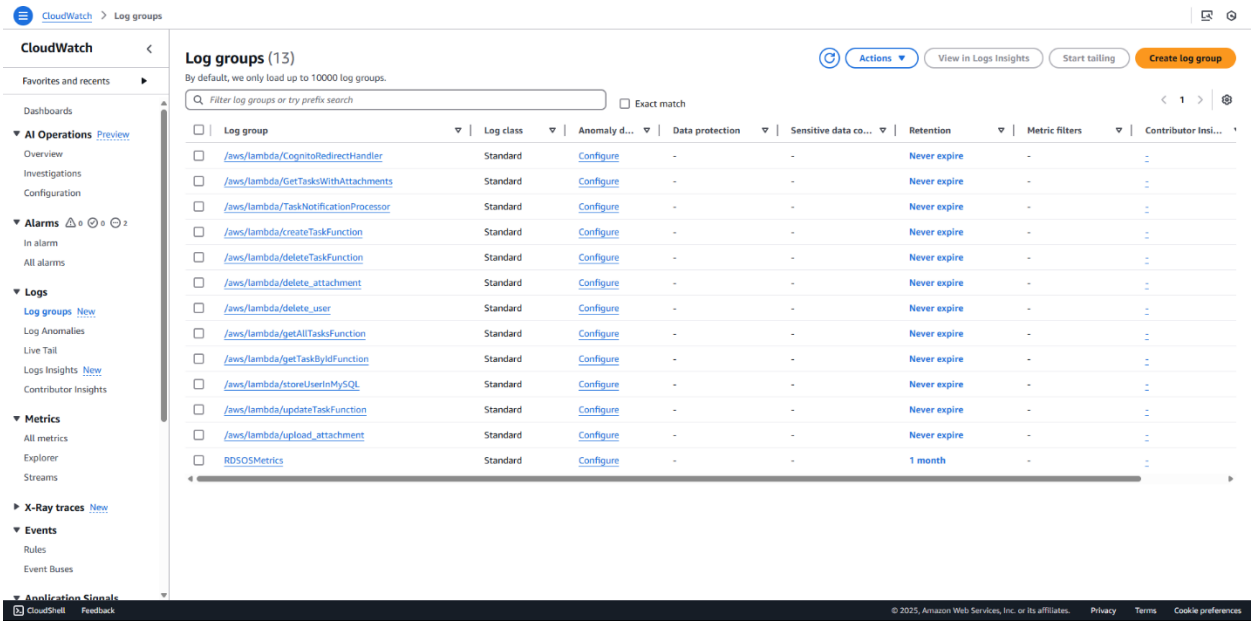
- Enable logging for:

  -Lambda functions.

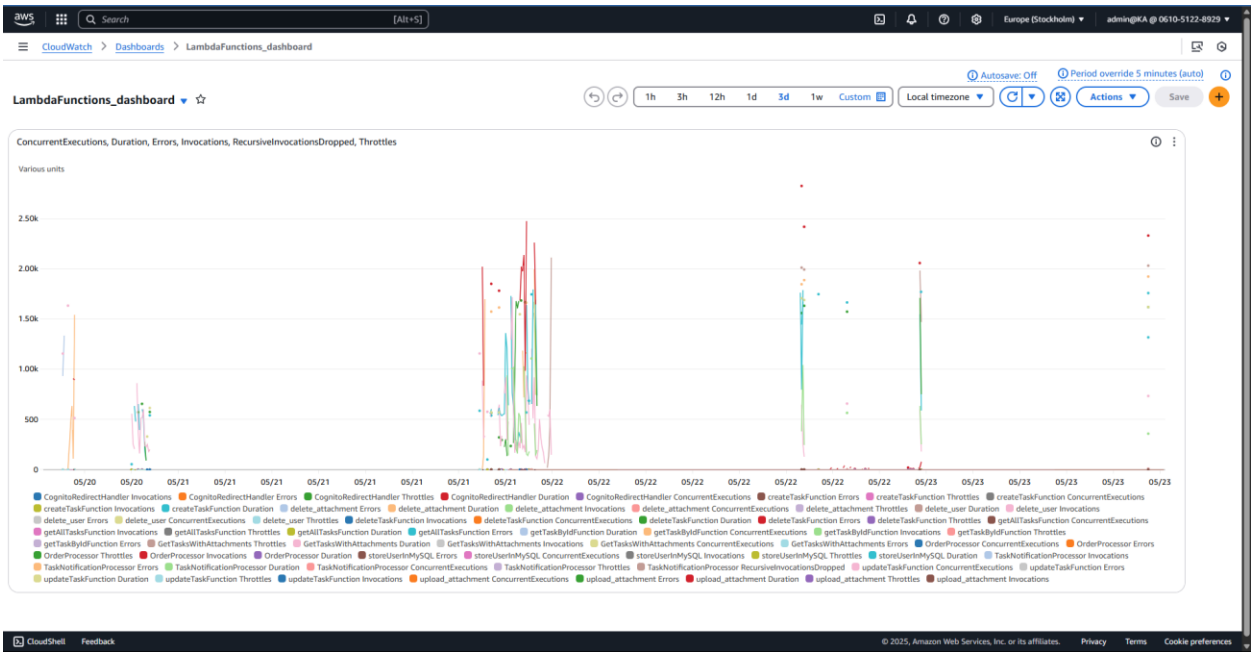
  -API Gateway access logs.

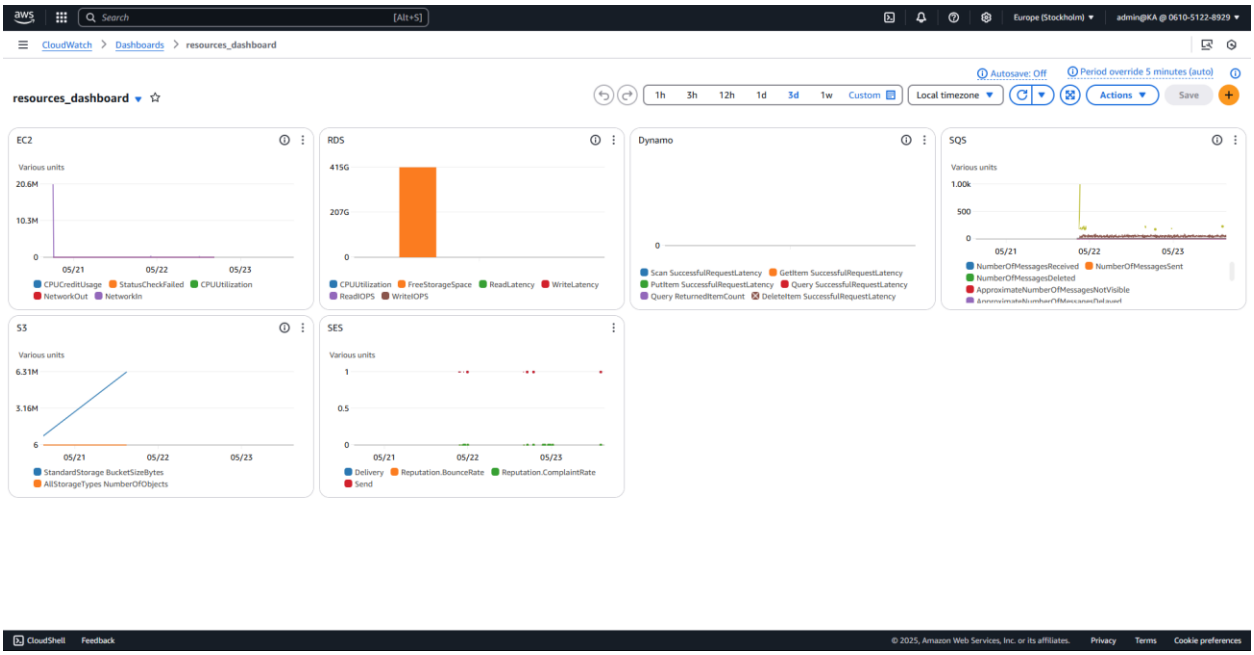
- Set up metrics dashboards.
- Create CloudWatch Alarms for errors or latency spikes.

LOG GROUPS

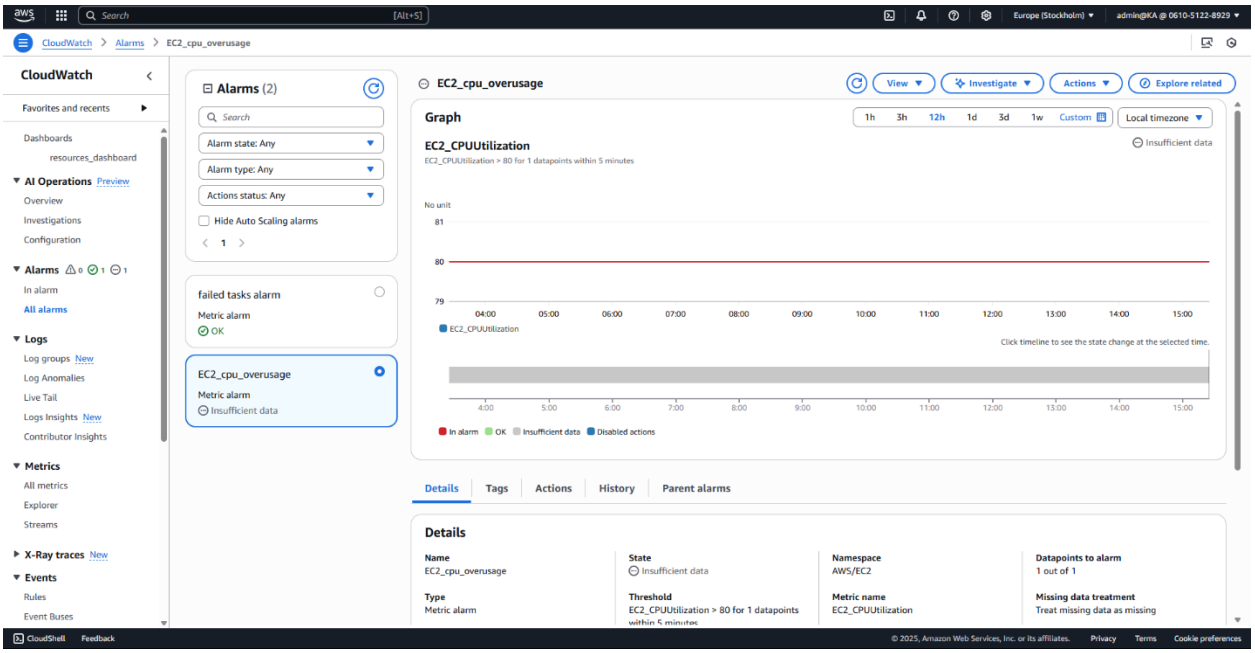


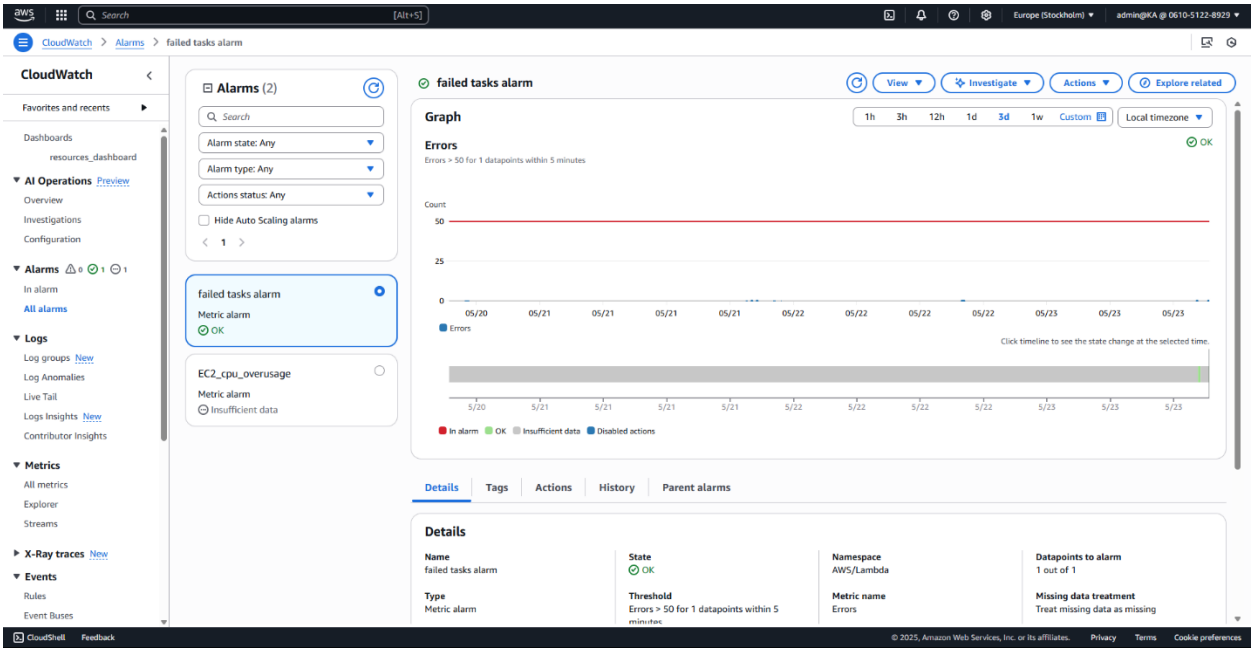
DASHBOARDS


CloudWatch > Dashboards > LambdaFunctions_dashboard
LambdaFunctions_dashboard
Autosave: Off
Period override 5 minutes (auto)
1h 3h 12h 1d 3d 1w Custom
Local timezone
Actions
Save
ConcurrentExecutions, Duration, Errors, Invocations, RecursiveInvocationsDropped, Throttles
Various units
2.50k
2.00k
1.50k
1.00k
500
0
05/20 05/20 05/21 05/21 05/21 05/21 05/21 05/21 05/21 05/21 05/22 05/22 05/22 05/22 05/22 05/22 05/22 05/22 05/23 05/23 05/23 05/23 05/23 05/23
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


CloudWatch > Dashboards > resources_dashboard
resources_dashboard
Autosave: Off
Period override 5 minutes (auto)
1h 3h 12h 1d 3d 1w Custom
Local timezone
Actions
Save
EC2
Various units
20.6M
10.3M
0
05/21 05/22 05/23
CPUCreditUsage StatusCheckFailed CPUUtilization
NetworkOut NetworkIn
RDS
415G
207G
0
CPUUtilization FreeStorageSpace ReadLatency WriteLatency
ReadIOPS WriteIOPS
Dynamo
0
Scan SuccessfulRequestLatency GetItem SuccessfulRequestLatency
PutItem SuccessfulRequestLatency Query SuccessfulRequestLatency
Query ReturnedItemCount DeleteItem SuccessfulRequestLatency
SQS
Various units
1.00k
500
0
05/21 05/22 05/23
NumberOfMessagesReceived NumberOfMessagesSent
NumberOfMessagesDeleted
ApproximateNumberOfMessagesNotVisible
S3
Various units
6.31M
3.16M
6
05/21 05/22 05/23
StandardStorage BucketSizeBytes
AllStorageTypes NumberOfObjects
SES
Various units
1
0.5
0
05/21 05/22 05/23
Delivery Reputation.BounceRate Reputation.ComplaintRate
Send
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

ALARMS


CloudWatch > Alarms > EC2_cpu_overusage
CloudWatch
Alarms (2)
failed tasks alarm
Metric alarm
OK
EC2_cpu_overusage
Metric alarm
Insufficient data
EC2_cpu_overusage
View
Investigate
Actions
Explore related
Graph
EC2_CPUUtilization
EC2_CPUUtilization > 80 for 1 datapoints within 5 minutes
Insufficient data
Details
Tags
Actions
History
Parent alarms
Name
EC2_cpu_overusage
State
Insufficient data
Namespace
AWS/EC2
Datapoints to alarm
1 out of 1
Type
Metric alarm
Threshold
EC2_CPUUtilization > 80 for 1 datapoints within 5 minutes
Metric name
EC2_CPUUtilization
Missing data treatment
Treat missing data as missing


CloudWatch > Alarms > failed tasks alarm
CloudWatch
Alarms (2)
failed tasks alarm
Metric alarm
OK
EC2_cpu_overusage
Metric alarm
Insufficient data
failed tasks alarm
View
Investigate
Actions
Explore related
Graph
Errors
Errors > 50 for 1 datapoints within 5 minutes
OK
Details
Tags
Actions
History
Parent alarms
Name
failed tasks alarm
State
OK
Namespace
AWS/Lambda
Datapoints to alarm
1 out of 1
Type
Metric alarm
Threshold
Errors > 50 for 1 datapoints within 5 minutes
Metric name
Errors
Missing data treatment
Treat missing data as missing

📐 Architecture