

Go to IAM roles and create a role

- Select AWS Service
- Select lambda in use case.

Trusted entity type

- AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Lambda

Choose a use case for the specified service.

Use case

Lambda
Allows Lambda functions to call AWS services on your behalf.

In the permissions select AWSLambdaBasicExecutionRole.

Add permissions

Permissions policies (1/1083) Info

Choose one or more policies to attach to your new role.

Filter by Type

Policy name	Type	Description
AWSLambdaBasicExecutionRole	AWS managed	Provides write permissions to CloudWat...

Set permissions boundary - optional

Cancel Previous Next

Give the name and click on create role.

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. The 'Role details' section is displayed, showing the role name 'lambda-status-logger-role' and a description 'Allows Lambda functions to call AWS services on your behalf.'. The sidebar on the left indicates the current step is 'Name, review, and create'.

Step 2:

Go to aws console and search lambda and click on that.

The screenshot shows the AWS Lambda service page. The search bar at the top contains 'lambda'. The main content area displays the 'Services' and 'Features' sections. Under 'Services', there are three items: 'Lambda' (Run code without thinking about servers), 'CodeBuild' (Build and Test Code), and 'AWS Signer' (Ensuring trust and integrity of your code). Under 'Features', there are two items: 'Lambda Insights' (CloudWatch feature) and 'Object Lambda Access Points' (S3 feature). A sidebar on the left lists 'Services', 'Features', 'Documentation', 'Knowledge articles', 'Marketplace', 'Blog posts', 'Events', and 'Tutorials'. A message at the bottom asks 'Were these results helpful?' with 'Yes' and 'No' buttons.

Click on create a function.

The screenshot shows the AWS Lambda landing page. At the top, there's a dark header with the AWS logo, a search bar, and account information. Below the header, the word "Compute" is displayed. The main content area features the heading "AWS Lambda" and the subtext "lets you run code without thinking about servers." A note below states: "You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration." To the right, a "Get started" box contains the text "Author a Lambda function from scratch, or choose from one of many preconfigured examples." It includes a "Create a function" button. At the bottom of the main content, there are tabs for ".NET", "Java", "Node.js" (which is selected), "Python", "Ruby", and "Custom runtime". Below the tabs is a code editor window with a snippet of Node.js code:

```
1 * exports.handler = async (event) => {  
2     console.log(event);  
3     return 'Hello from Lambda!';  
4 };  
5
```

Buttons for "Run" and "Next: Lambda responds to events" are at the top of the code editor. At the very bottom, there are links for "CloudShell", "Feedback", and copyright information.

Give the function name and select python language and default architecture.

The screenshot shows the "Create function" wizard. The top navigation bar includes the AWS logo, a search bar, and account information. Below the navigation, the path "Lambda > Functions > Create function" is shown. The main section is titled "Create function" with a "Info" link. It says "Choose one of the following options to create your function." Three options are available: "Author from scratch" (selected), "Use a blueprint", and "Container image".

Basic information

Function name
Enter a name that describes the purpose of your function.
StatusLoggerFunction

Runtime | **Info**
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture | **Info**
Choose the instruction set architecture you want for your function code.
 x86_64
 arm64

Permissions | **Info**

Change the default execution role select existing role and select the role Lambda-status-logger-role. And create the function.

In the function page go to the editor and replace the code with this python code.

Click on deploy to save the changes.

```

EXPLORER
STATUSLOGGERFUNCTION
lambda_function.py

lambda_function.py
1 import json
2 import datetime
3
4 def lambda_handler(event, context):
5     # Get the current time
6     current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
7
8     # Define the status message
9     status_message = f"[{current_time}] System Health Check: OK. Function executed successfully."
10
11    # Print the message (This automatically writes to CloudWatch Logs)
12    print(status_message)
13
14    # You can also return a response if needed
15    return {
16        'statusCode': 200,
17        'body': json.dumps({'message': status_message})
18    }

```

On the functions page click on add trigger.

The screenshot shows the AWS Lambda Functions console. The top navigation bar includes 'Lambda > Functions > StatusLoggerFunction'. The main area displays the 'StatusLoggerFunction' details. It features a 'Function overview' section with tabs for 'Diagram' (selected) and 'Template'. Below this is a card for 'StatusLoggerFunction' showing its icon, name, and 'Layers (0)'. Buttons for '+ Add trigger' and '+ Add destination' are present. To the right is an 'Info' sidebar with sections for 'Description', 'Last modified', 'Function ARN', and 'Function URL'. A 'Create' sidebar on the far right lists actions like 'Build', 'Deploy', and 'InVOKE'. At the bottom are tabs for 'Code' (selected), 'Test', 'Monitor', 'Configuration', 'Aliases', and 'Versions', along with code source options and a 'Star' button.

- Select EventBridge (CloudWatch Events)
- Create a new role -rule name as FiveMinutesScheduleRule.
- Select rule type as Schedule expression.

The screenshot shows the 'Add trigger' configuration page. The top navigation bar includes 'Lambda > Add triggers'. The main area is titled 'Trigger configuration' with tabs for 'EventBridge (CloudWatch Events)' (selected), 'aws asynchronous schedule management-tools'. Below this is a 'Rule' section with 'Create a new rule' selected. The 'Rule name' field contains 'FiveMinuteScheduleRule'. The 'Rule description' field is empty. The 'Rule type' section shows 'Schedule expression' selected. At the bottom are buttons for 'CloudShell' and 'Feedback'.

- Schedule expression as rate(5 minutes).
- Click on add.

Lambda > Add triggers

Rule name
Enter a name to uniquely identify your rule.

Rule description
Provide an optional description for your rule.

Rule type
Trigger your target based on an event pattern, or based on an automated schedule.
 Event pattern
 Schedule expression

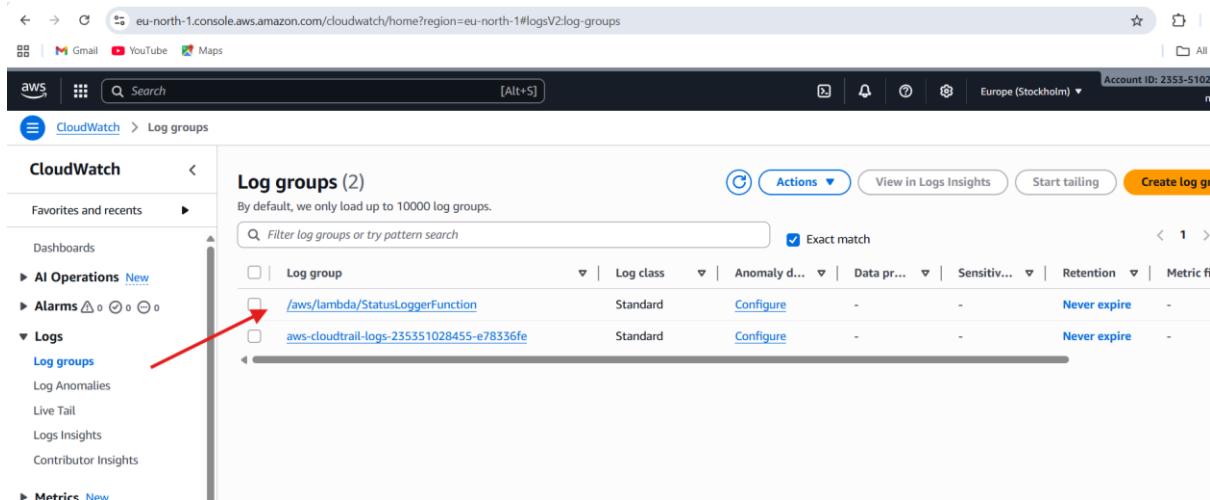
Schedule expression
Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC.

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

[Cancel](#) [Add](#)

Navigate to cloudwatch and go to log groups there you will find created lambdastatusloggerfunction click on that.



eu-north-1.console.aws.amazon.com/cloudwatch/home?region=eu-north-1#logsV2:log-groups

CloudWatch > Log groups

Log groups (2)

Log group	Log class	Anomaly d...	Data pr...	Sensitiv...	Retention	Metric fi...
/aws/lambda/StatusLoggerFunction	Standard	Configure	-	-	Never expire	-
aws-cloudtrail-logs-235351028455-e78336fe	Standard	Configure	-	-	Never expire	-

In that go to logstream select the logstream.

You should see entries repeating every five minutes.