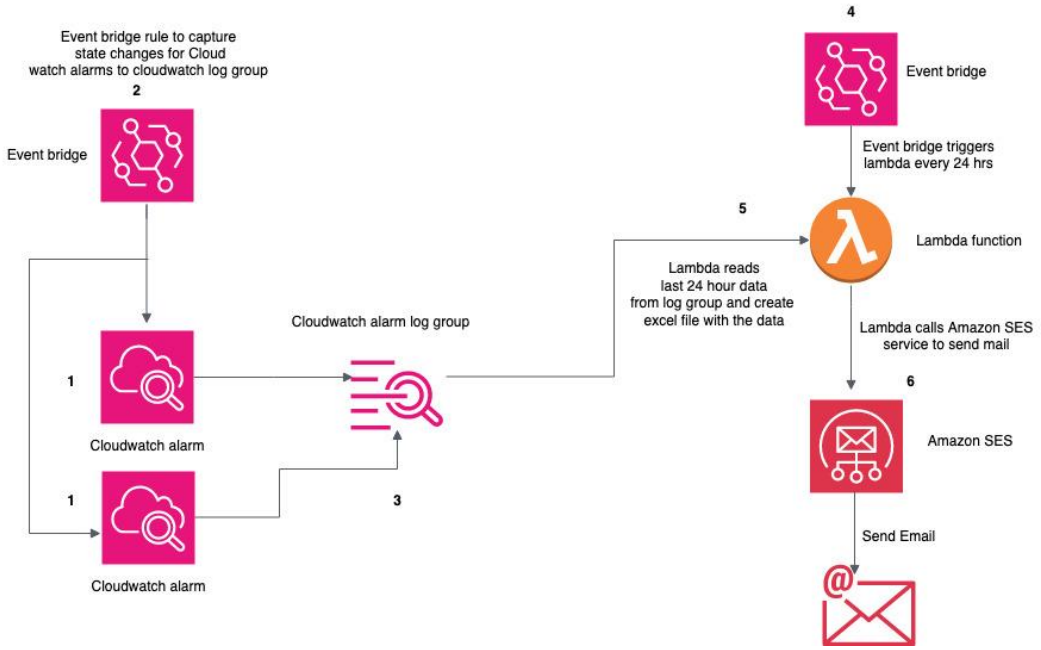
Usecase:

Create a csv file which includes the following details of all the cloudwatch alarm in a region. This csv file will be created by AWS lambda with Amazon Event Bridge trigger, which triggers AWS lambda function for every 24 hours and store the created csv file in S3 and also sends email attachment with the help of Amazon SES service. The data of the files are for the last 24 hours.

CSV file columns :

instance name (take from alarm description), instance id, Metric name, Alarm Name ,Alarm Time, Threshold value, Threshold breach value.

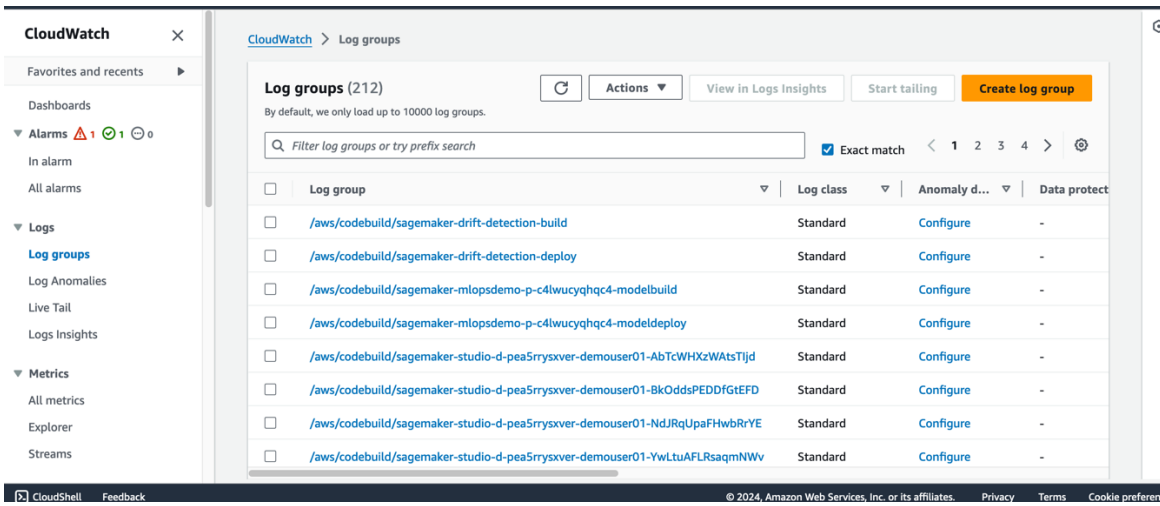
Architecture Diagram:



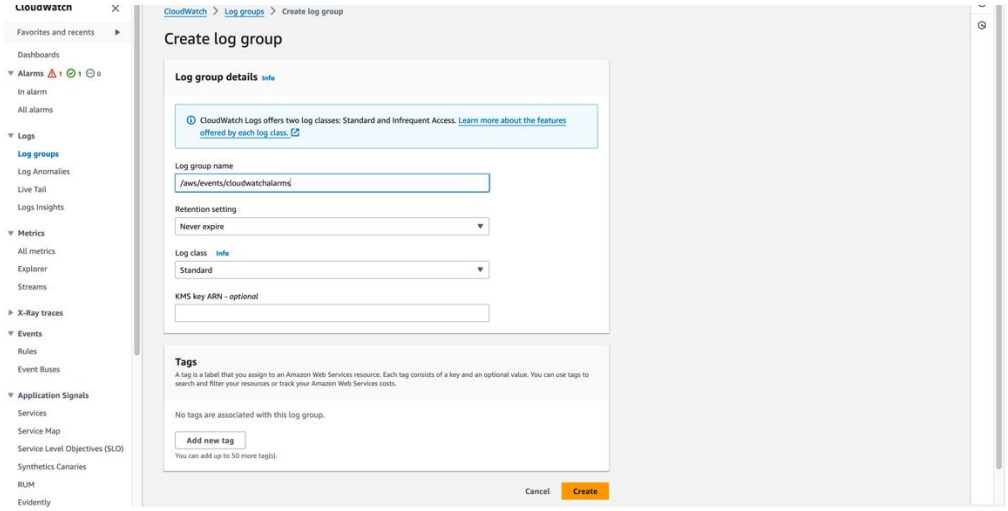
The purpose of this code is to generate a report containing details of all Amazon CloudWatch alarms triggered in a specific AWS region within the last 24 hours. The report will be generated automatically by an AWS Lambda function, triggered by an Amazon EventBridge rule every 24 hours. The report will be in CSV format and will include columns such as instance name, instance ID, metric name, alarm name, alarm time, threshold value, and threshold breach value for each triggered alarm. The generated CSV file will be stored in an S3 bucket, and an email with the CSV file attached will be sent using Amazon SES.

Step 1: Create a Log group in Amazon cloudwatch.

- Navigate to cloudwatch console and click '**Log groups**' from the left panel and then Click "**Create log group"** button in right top corner.

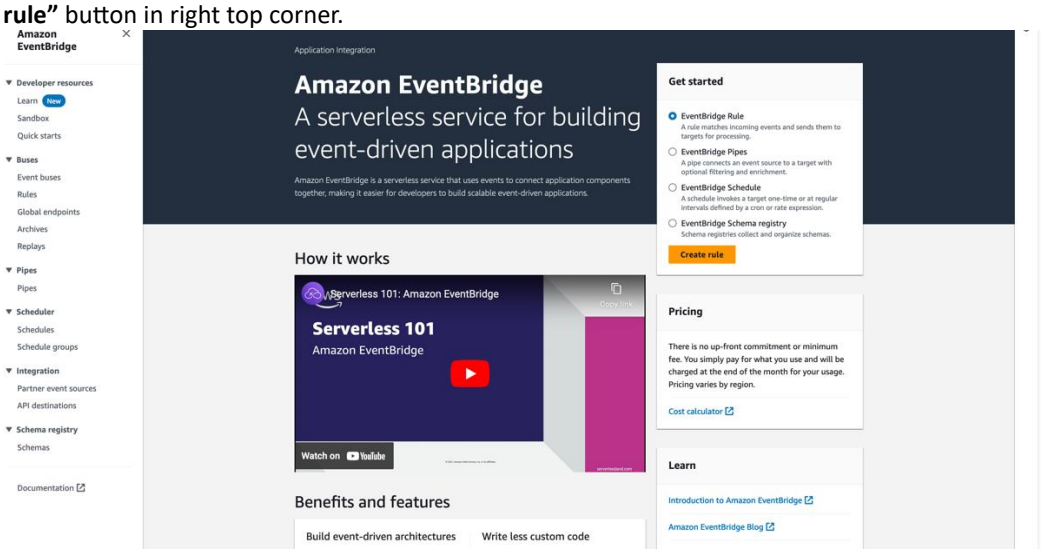


- Enter any name in the “**Log group name**” field and click create button at the right bottom corner of the page.

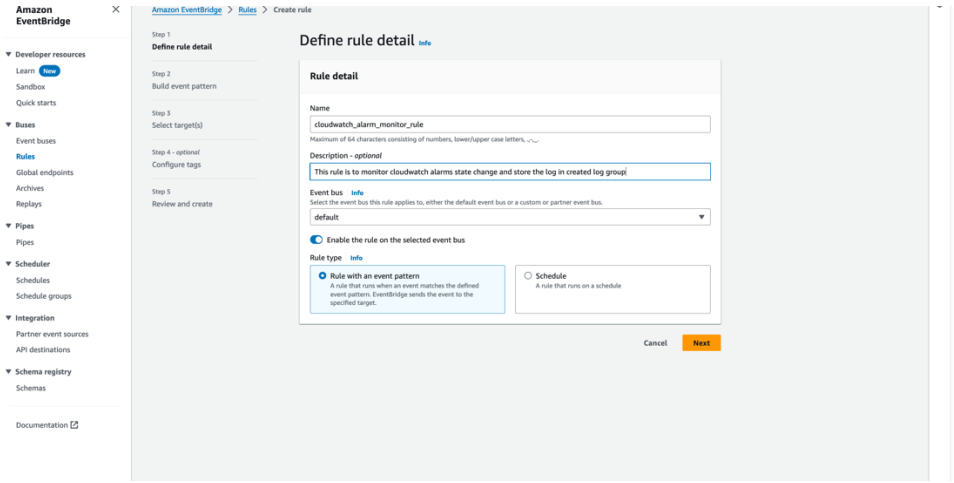


Step 2: Create an Amazon EventBridge rule to capture the cloudwatch alarm state changes into the previously created Log group in step 1.

- Navigate to Amazon Event bridge console and click ‘**Create rule**’ button in the Amazon Event bridge home page or from the left panel click “Rules” and then Click “**Create rule”** button in right top corner.

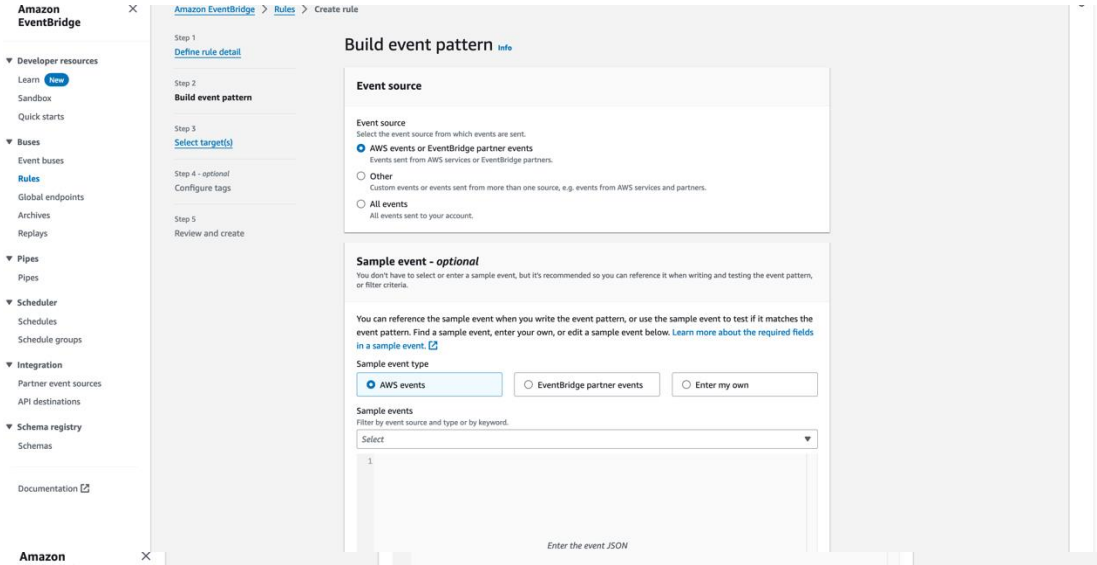


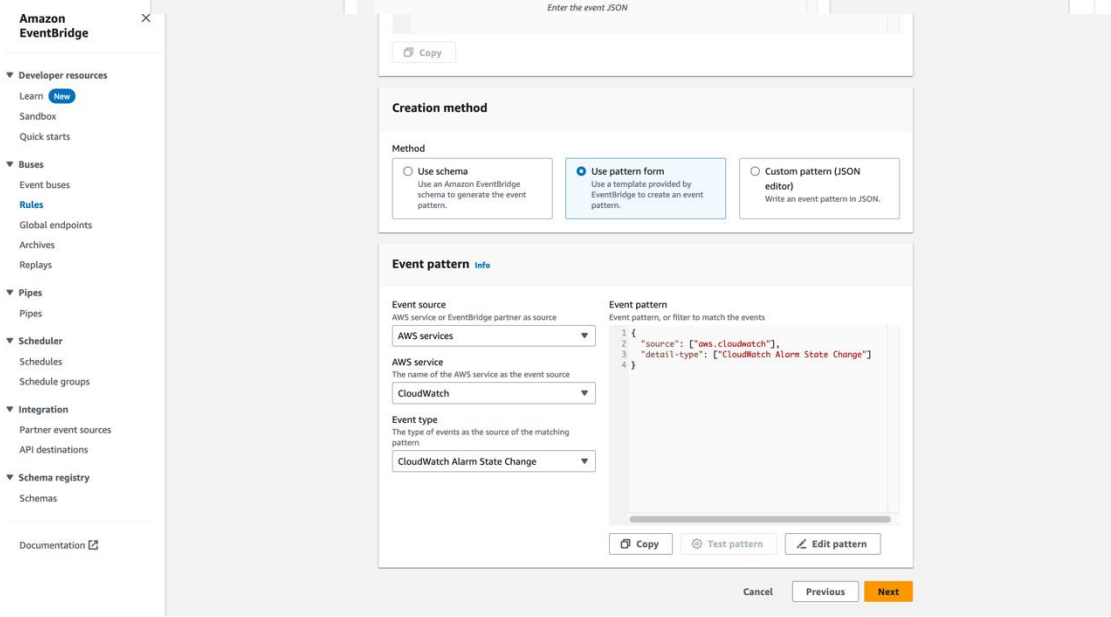
- Enter any name for this rule in the field “Name” and give the description of this rule in “Description” field and then click “Next” button.



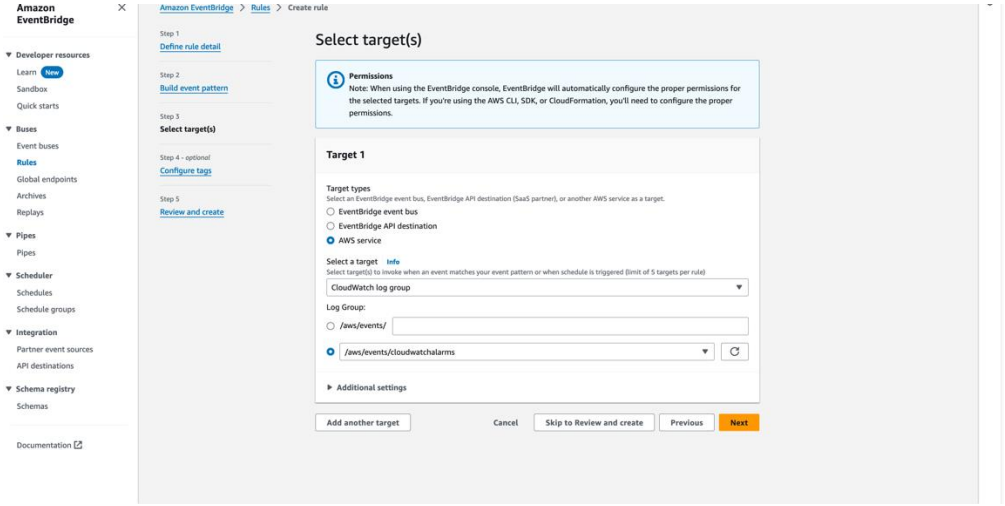
-

- Choose **“AWS Events or event bridge partner events”** for event source radio button. Scroll down and choose “Use pattern form” for “creation method” section. In the “Event pattern” section choose AWS service as “CloudWatch” from dropdown and choose “CloudWatch Alarm State Change” from the drop down for “Event type”. Click Next.



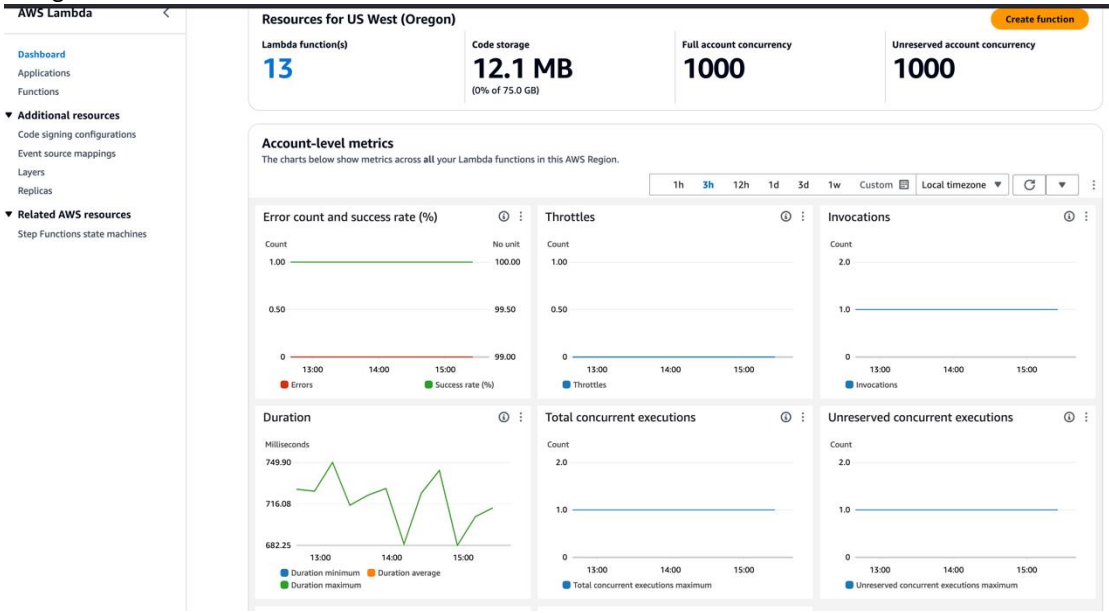
- 

- Choose “AWS service” radio button from the Target 1 section and choose “CloudWatch log group” from the “select a target” drop down. In the “Log group” section drop down, choose the log group name created in the step 1. Click Next and then next and then “create rule” button in the review and create section.

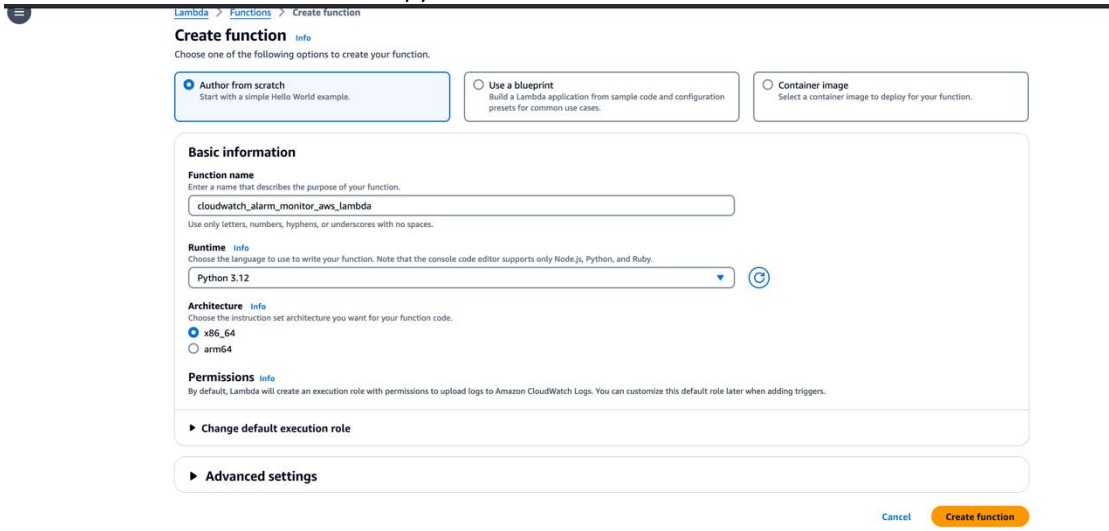


Step 3: Create a AWS lambda function with the attached sample code automated-cloudwatch-alarm-reporting-system.py

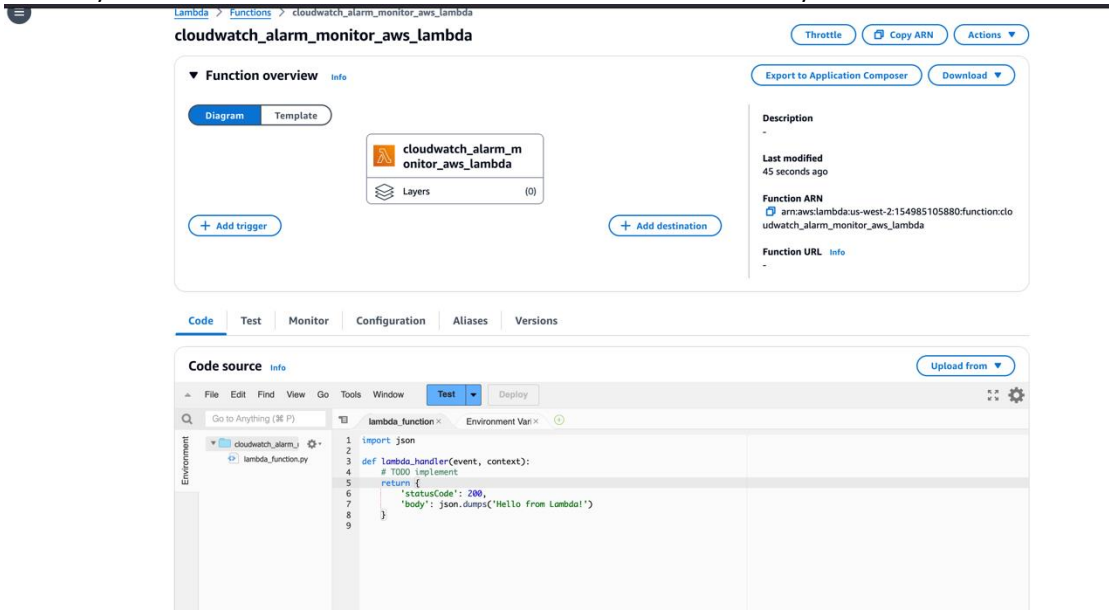
- Navigate to AWS Lambda dashboard and click create function button.

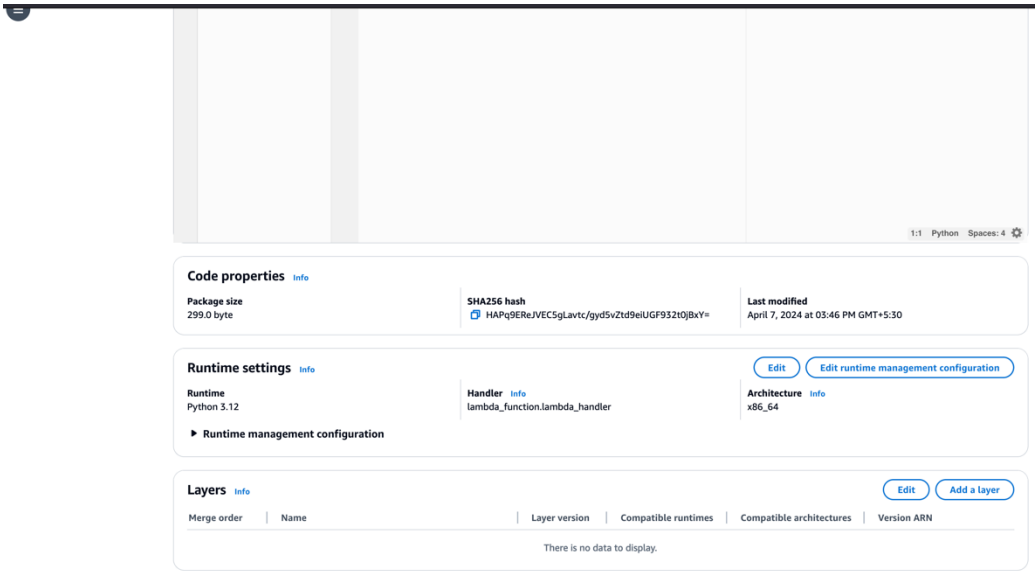


- Enter the function name and choose python as runtime and click create function button.

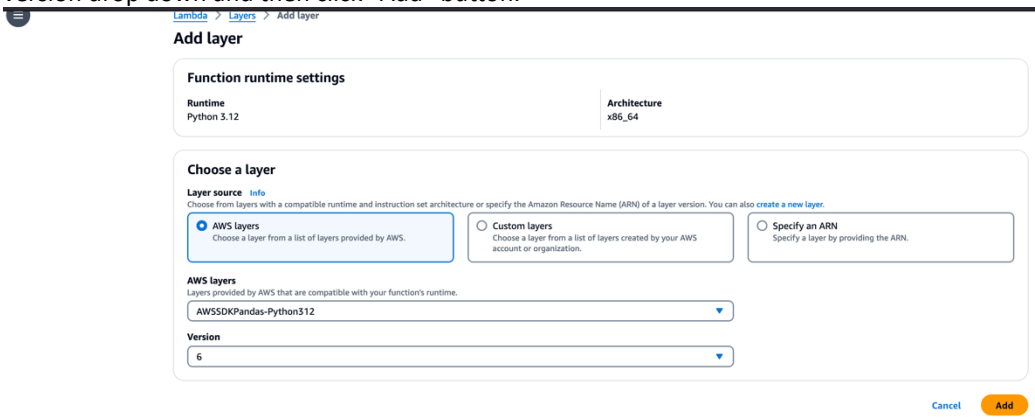


- Click “Layers” from the function overview section and then click “Add a layer” button.

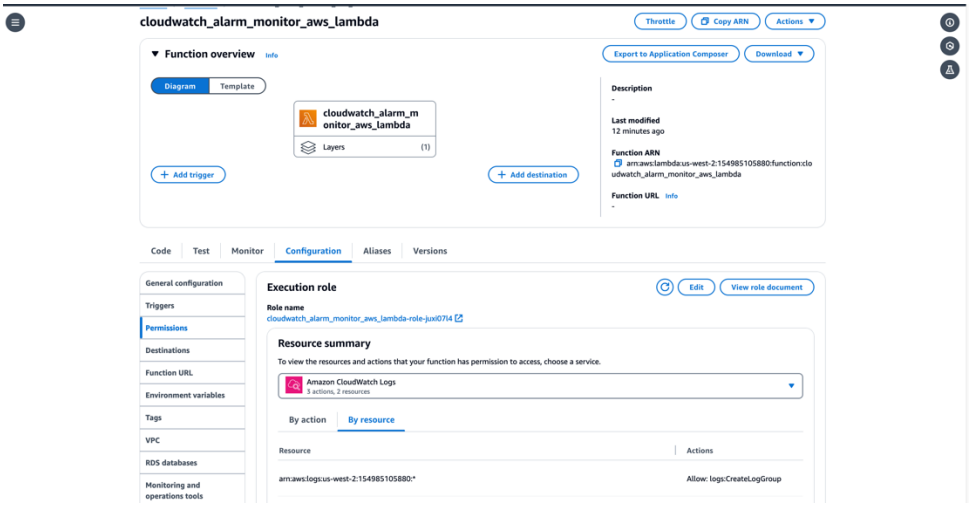




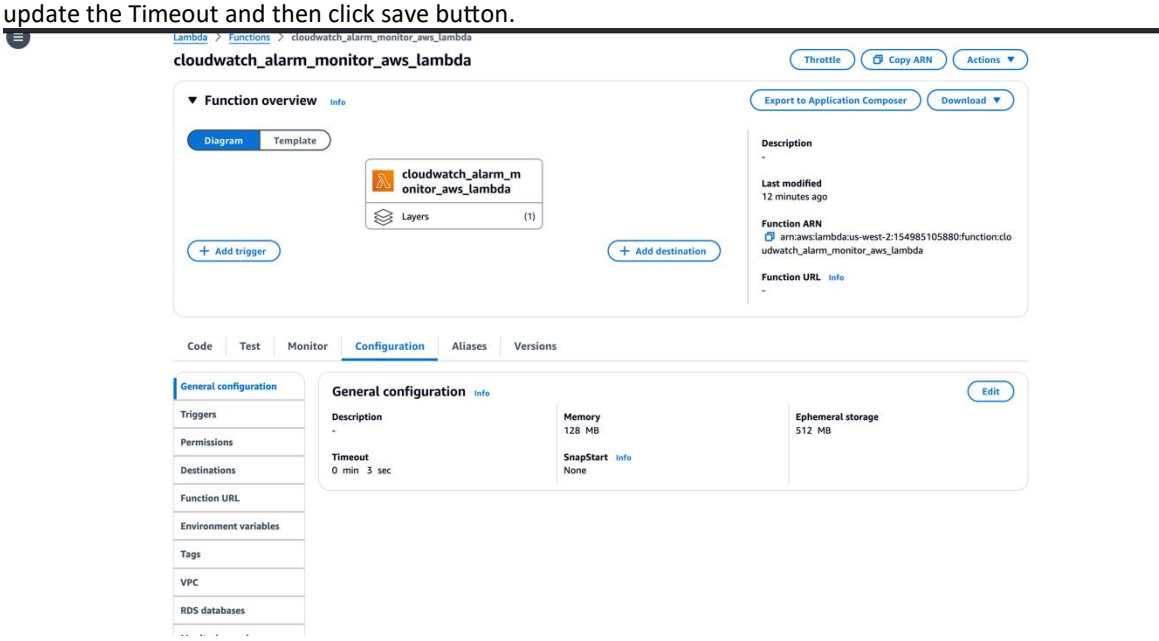
-
- Choose “AWSSDKPandas-Python312” as AWS layers and choose latest version from version drop down and then click “Add” button.



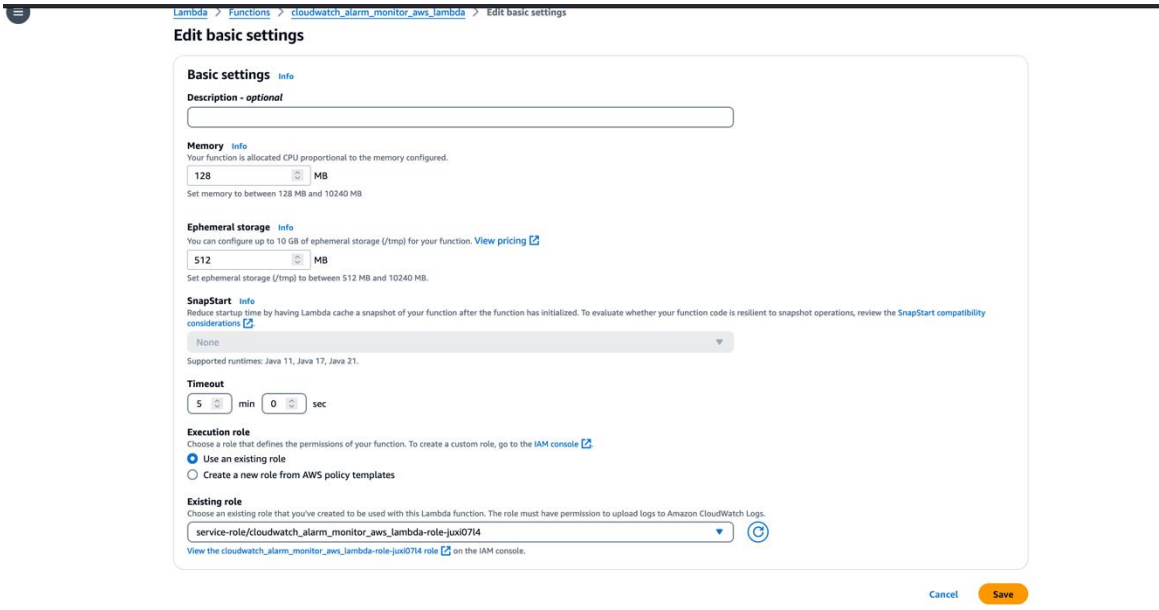
-
- Click “configuration” tab and then click “permissions” tab. Click on the Role name and it will take us to IAM and then attach policies for this Lambda function to have access to Amazon Cloudwatch, S3 bucket and Amazon SES service.



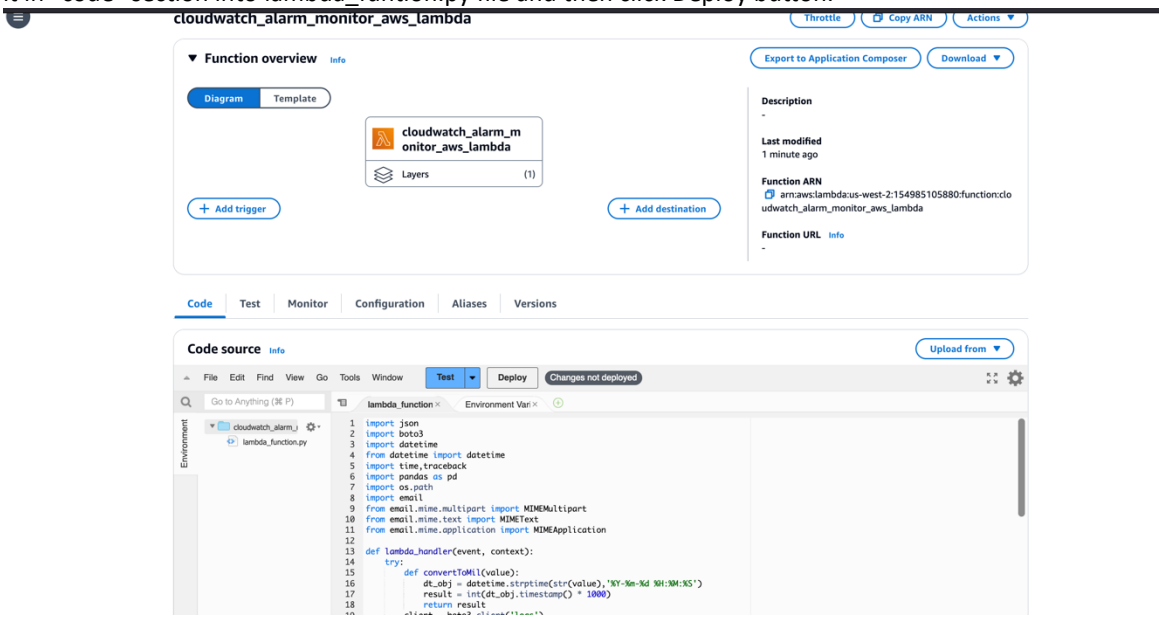
- Click “configuration” tab and then click “general configuration” tab. Click Edit button to update the Timeout and then click save button.



-



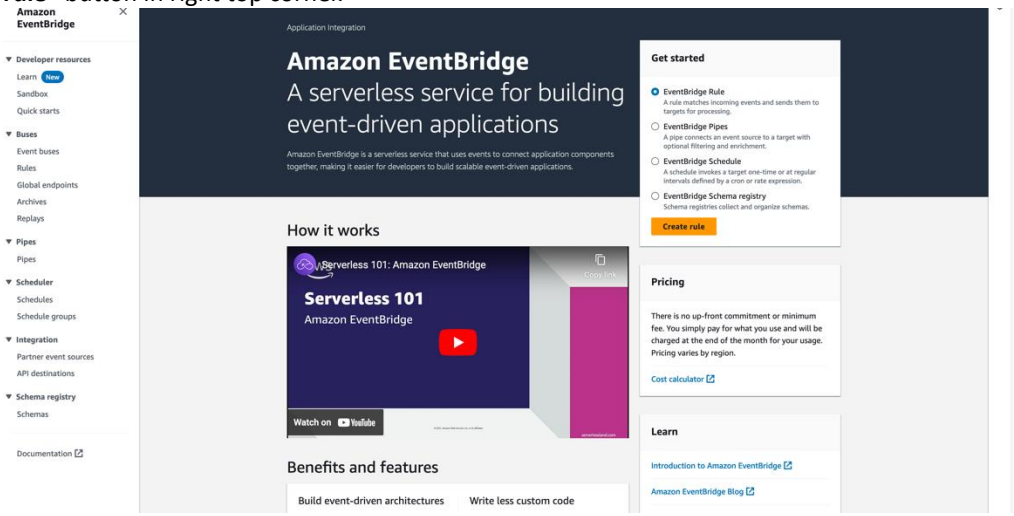
-
- Copy the sample code from “Cloudwatch_alarm_dataCSV_to_mail_s3.py” file and paste it in “code” section into lambda_funtion.py file and then click Deploy button.



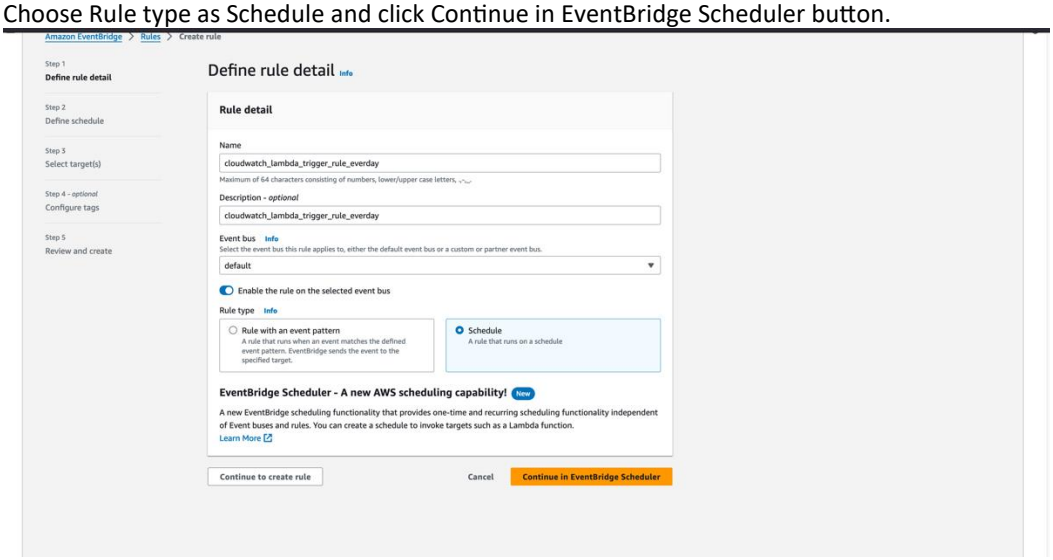
-

Step 4: Create a Amazon Event Bridge rule to trigger the AWS Lambda function created in step:2, everyday at a specific time of a day.

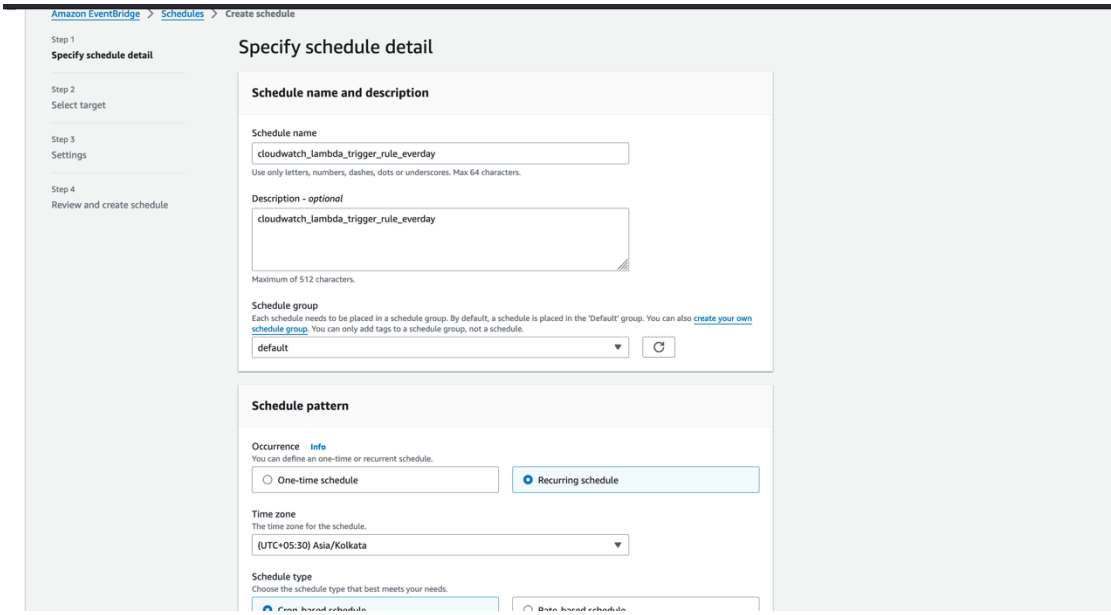
- https://docs.aws.amazon.com/scheduler/latest/UserGuide/schedule-types.html?icmpid=docs_console_unmapped#cron-based
- Navigate to Amazon Event bridge console and click '**Create rule**' button in the Amazon Event bridge home page or from the left panel click "Rules" and then Click "**Create rule"** button in right top corner.



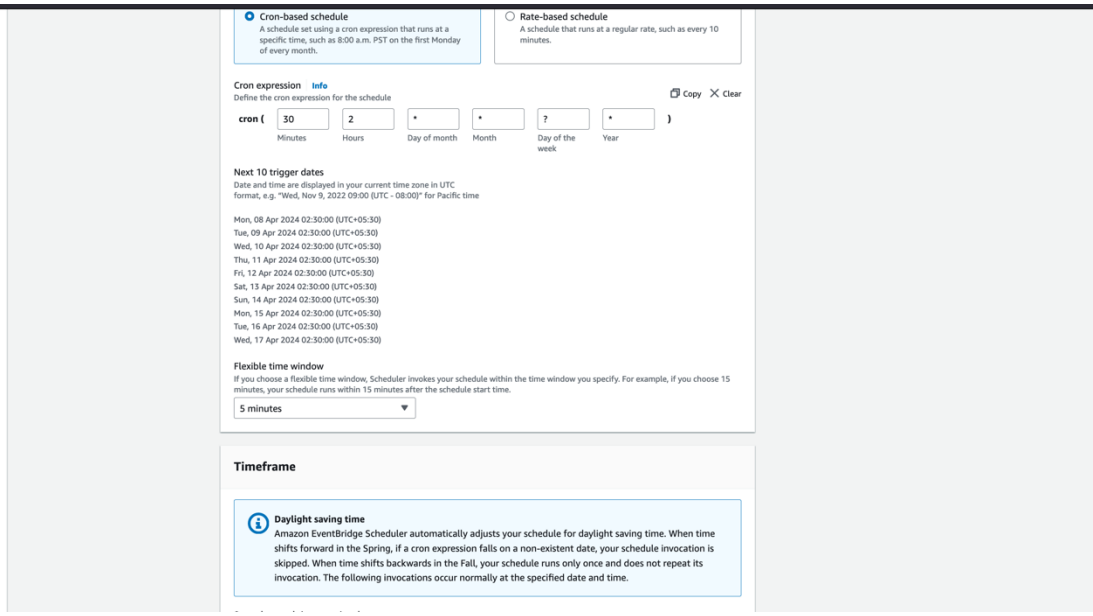
- Choose Rule type as Schedule and click Continue in EventBridge Scheduler button.



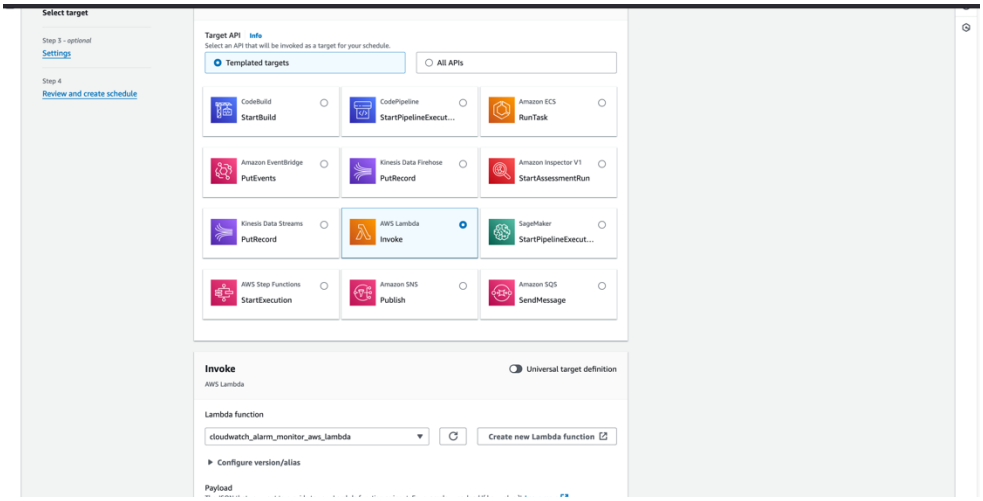
- Enter the schedule name, description and choose radio button Recurring Schedule.



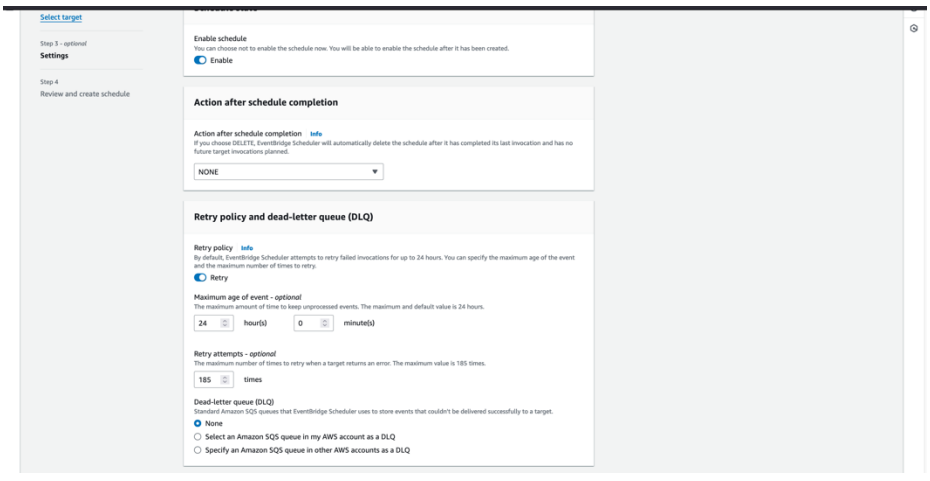
- Under Schedule type, choose Cron-based schedule. Enter the Cron expression, The below screenshot shows cron expression to run this schedule at everyday 2.30 a.m. Choose Flexible time window as 5 minutes and click Next.

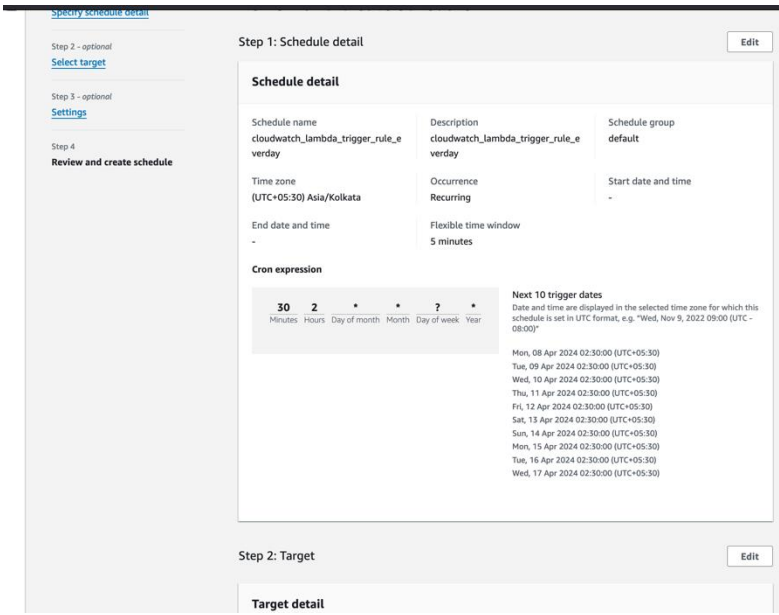


- Choose AWS Lambda radio button and select the lambda function which we created in step 3 and click next.



- Choose NONE from Action after schedule completion dropdown and click Next and in Review and create schedule page click the review and create.




Specify schedule detail
Step 2 - optional
Select target
Step 3 - optional
Settings
Step 4
Review and create schedule
Step 1: Schedule detail
Edit
Schedule detail
Schedule name
cloudwatch_lambda_trigger_rule_everday
Description
cloudwatch_lambda_trigger_rule_everday
Schedule group
default
Time zone
(UTC+05:30) Asia/Kolkata
Occurrence
Recurring
Start date and time
-
End date and time
-
Flexible time window
5 minutes
Cron expression
30 2 * * ? *
Minutes Hours Day of month Month Day of week Year
Next 10 trigger dates
Date and time are displayed in the selected time zone for which this schedule is set in UTC format, e.g. "Wed, Nov 9, 2022 09:00 (UTC - 08:00)"
Mon, 08 Apr 2024 02:30:00 (UTC+05:30)
Tue, 09 Apr 2024 02:30:00 (UTC+05:30)
Wed, 10 Apr 2024 02:30:00 (UTC+05:30)
Thu, 11 Apr 2024 02:30:00 (UTC+05:30)
Fri, 12 Apr 2024 02:30:00 (UTC+05:30)
Sat, 13 Apr 2024 02:30:00 (UTC+05:30)
Sun, 14 Apr 2024 02:30:00 (UTC+05:30)
Mon, 15 Apr 2024 02:30:00 (UTC+05:30)
Tue, 16 Apr 2024 02:30:00 (UTC+05:30)
Wed, 17 Apr 2024 02:30:00 (UTC+05:30)
Step 2: Target
Edit
Target detail

-