To create a Lambda function that can turn an RDS instance off when it hasn't been used for 45 minutes and turn it back on when needed, you need to follow these steps:

**Step 1: Create IAM Role for Lambda**

Ensure you have an IAM role that allows Lambda functions to interact with RDS and CloudWatch. The role should have policies attached that allow the following actions:

• rds:DescribeDBInstances

• rds:StopDBInstance

• rds:StartDBInstance

• cloudwatch:GetMetricStatistics

### Step 2: Create the Lambda Function to Turn Off RDS

* Go to the Lambda Console and create a new Lambda function.
* Choose Python 3.x as the runtime.
* Attach the IAM Role created in Step 1 to the Lambda function.
* Add the following code to the Lambda function:

import boto3

from datetime import datetime, timedelta

rds = boto3.client('rds')

cloudwatch = boto3.client('cloudwatch')

# RDS instance identifier

DB\_INSTANCE\_IDENTIFIER = 'mymariadbinstance'

def lambda\_handler(event, context):

# Get the current time

now = datetime.utcnow()

# Define the time period to check (last 5 minutes)

start\_time = now - timedelta(minutes=5)

end\_time = now

# Get CPU utilization metrics from CloudWatch

response = cloudwatch.get\_metric\_statistics(

Namespace='AWS/RDS',

MetricName='CPUUtilization',

Dimensions=[

{'Name': 'DBInstanceIdentifier', 'Value': DB\_INSTANCE\_IDENTIFIER}

],

StartTime=start\_time,

EndTime=end\_time,

Period=300,

Statistics=['Average']

)

# Check if the average CPU utilization is less than or equal to 5%

cpu\_utilizations = [point['Average'] for point in response['Datapoints']]

if all(cpu <= 5 for cpu in cpu\_utilizations):

# Stop the RDS instance if it hasn't been used

print(f"Stopping RDS instance {DB\_INSTANCE\_IDENTIFIER}")

rds.stop\_db\_instance(DBInstanceIdentifier=DB\_INSTANCE\_IDENTIFIER)

else:

print(f"RDS instance {DB\_INSTANCE\_IDENTIFIER} is in use")

return {

'statusCode': 200,

'body': 'Checked RDS usage and stopped instance if not used.'

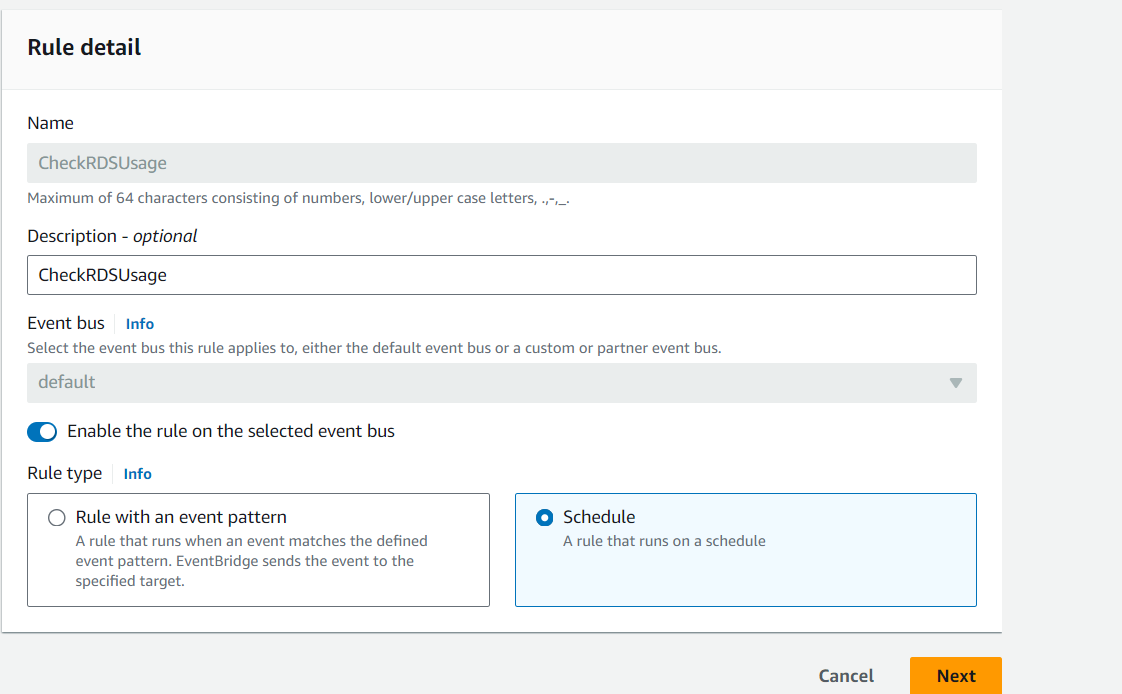
}

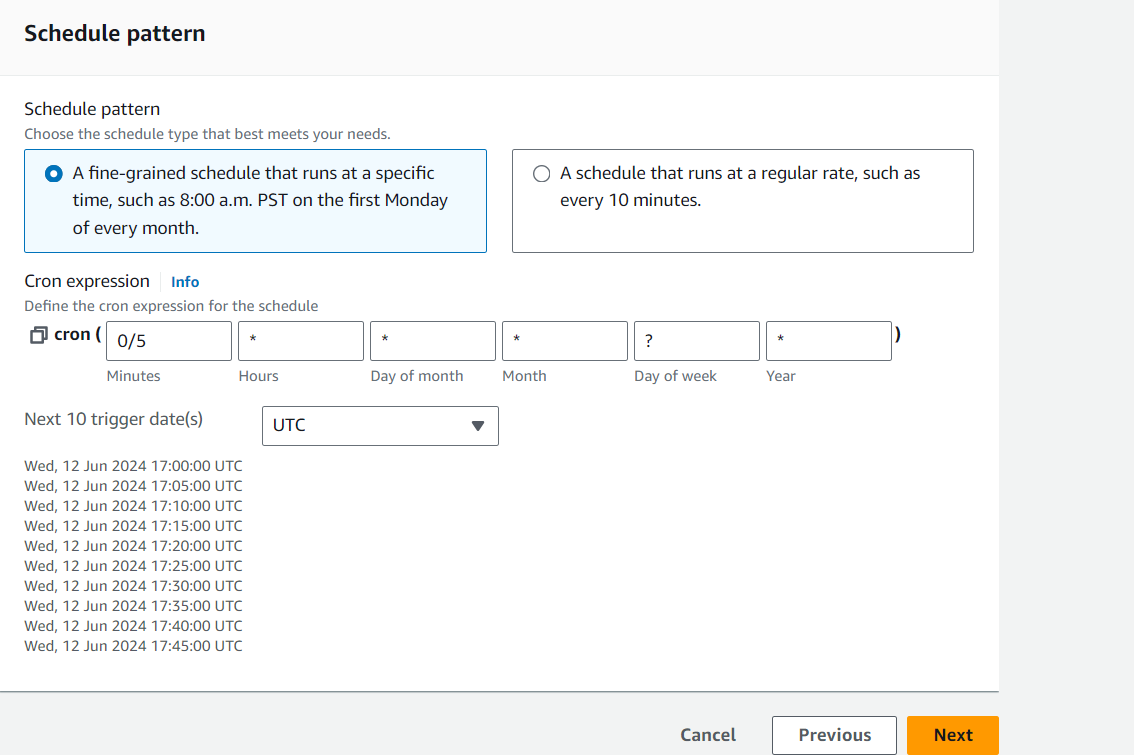
Step 3: Set Up CloudWatch Events

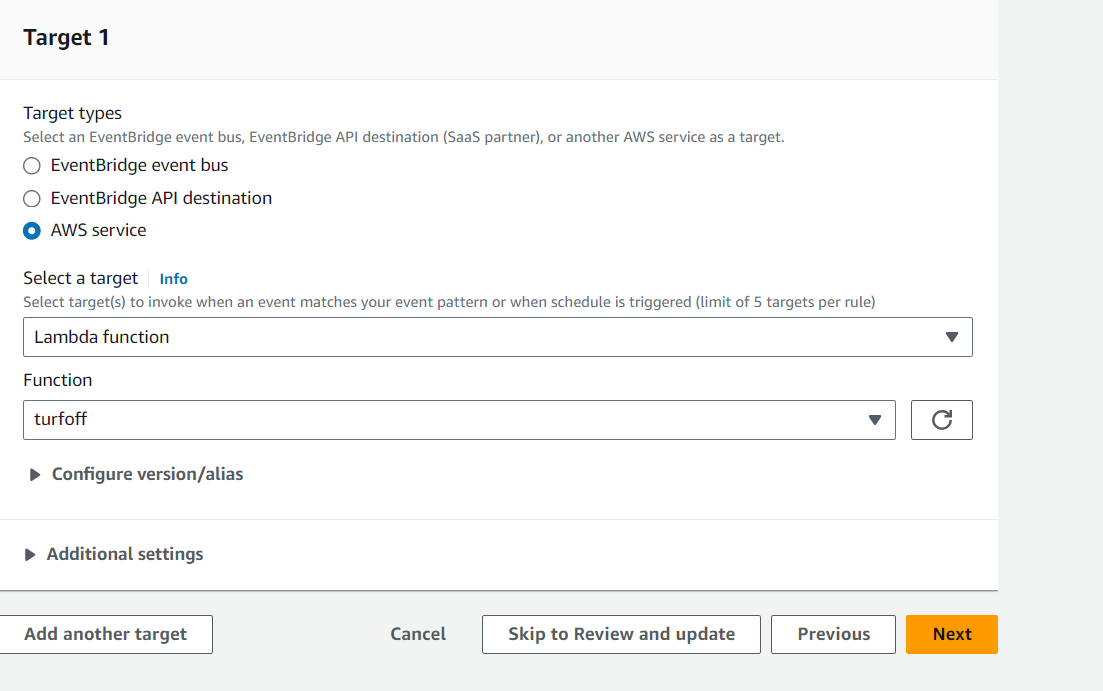
 **Go to the CloudWatch Console** and create a new rule.

 **Configure the rule** to trigger the Lambda function every 5 minutes.

* **Event Source**: Schedule
* **Rule Name**: CheckRDSUsage
* **Schedule Expression**: rate(5 minutes)







### Step 4: Create the Lambda Function to Turn On RDS

1. **Create another Lambda function** similar to the one above.
2. **Add the following code to the new Lambda function**:

import boto3

rds = boto3.client('rds')

# RDS instance identifier

DB\_INSTANCE\_IDENTIFIER = 'your-rds-instance-identifier'

def lambda\_handler(event, context):

# Start the RDS instance

print(f"Starting RDS instance {DB\_INSTANCE\_IDENTIFIER}")

rds.start\_db\_instance(DBInstanceIdentifier=DB\_INSTANCE\_IDENTIFIER)

return {

'statusCode': 200,

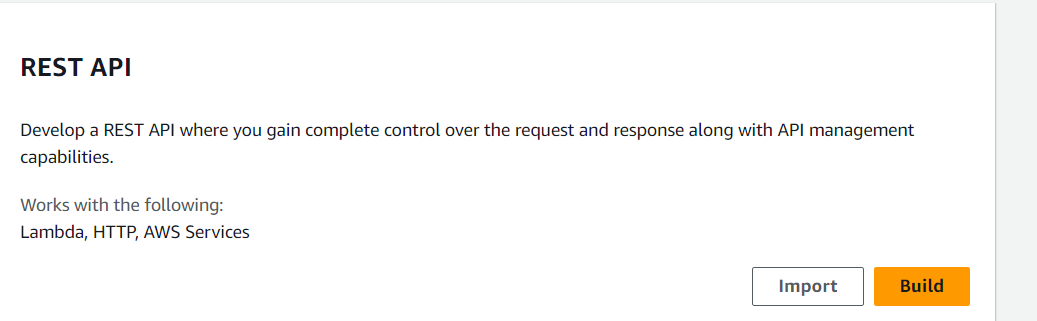
'body': 'Started RDS instance.'

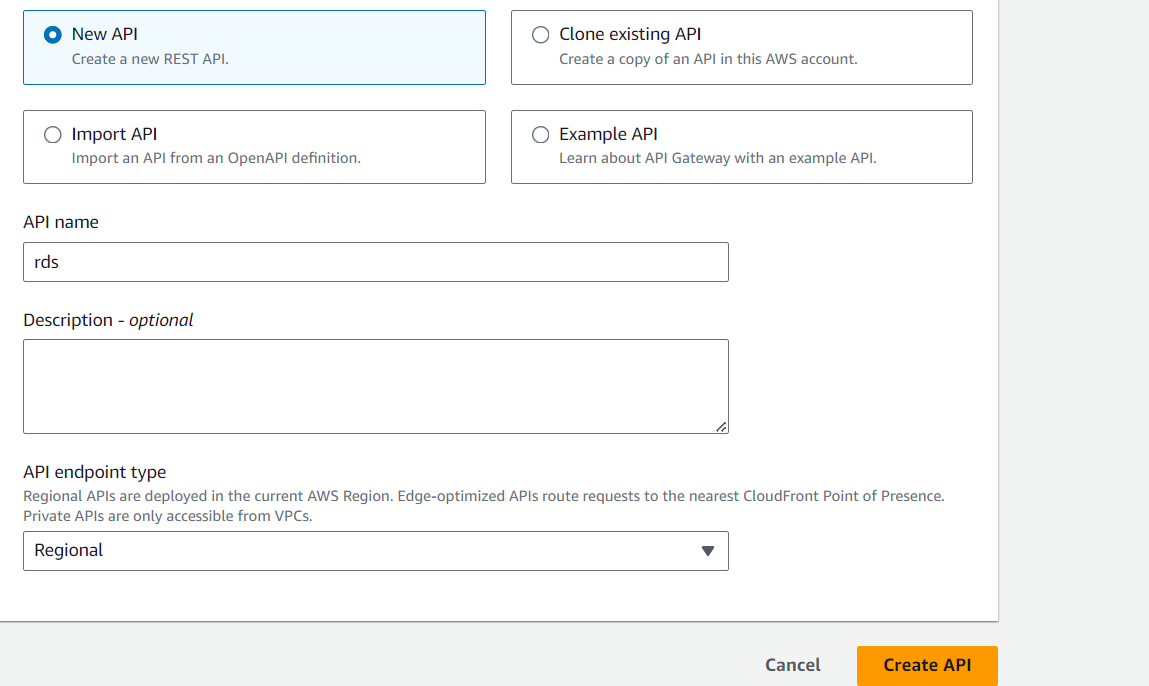
}

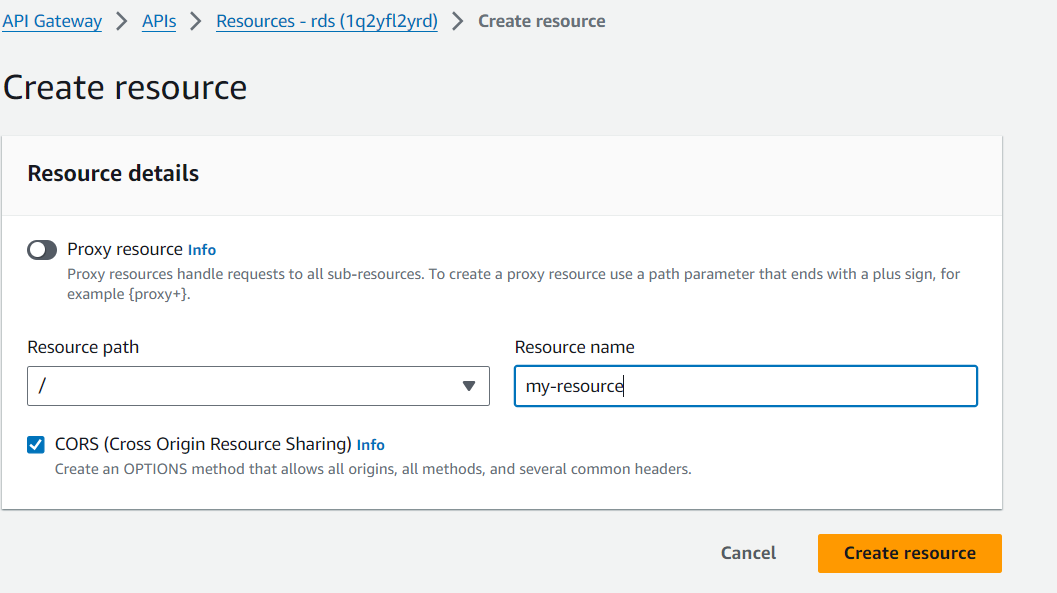
### Step 5: Set Up an API Gateway (Optional)

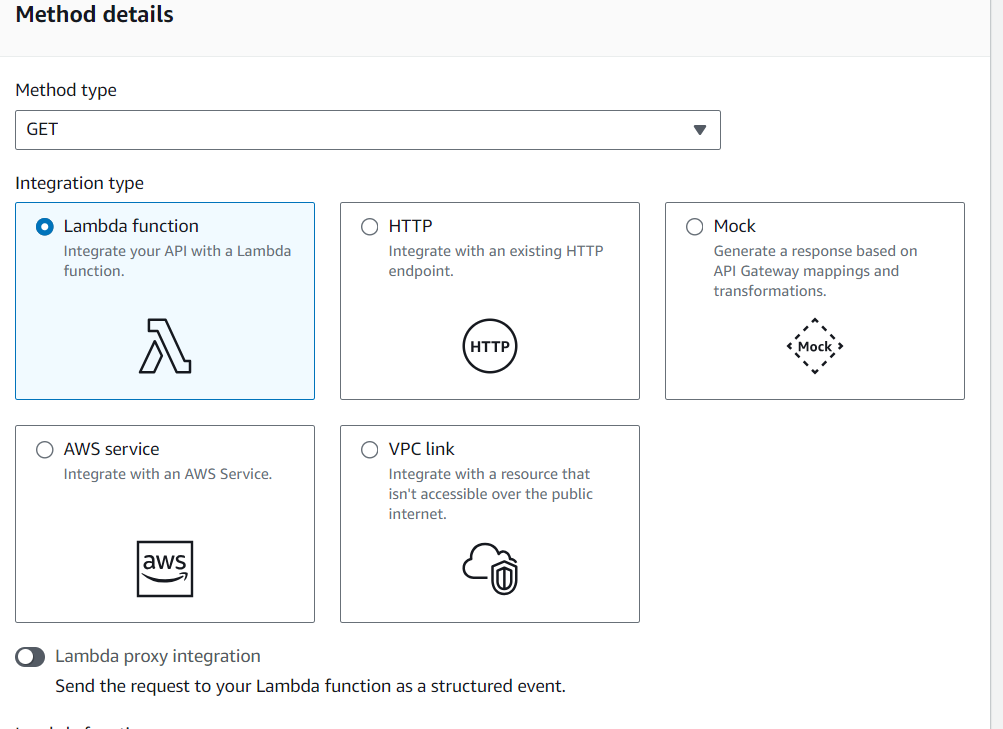
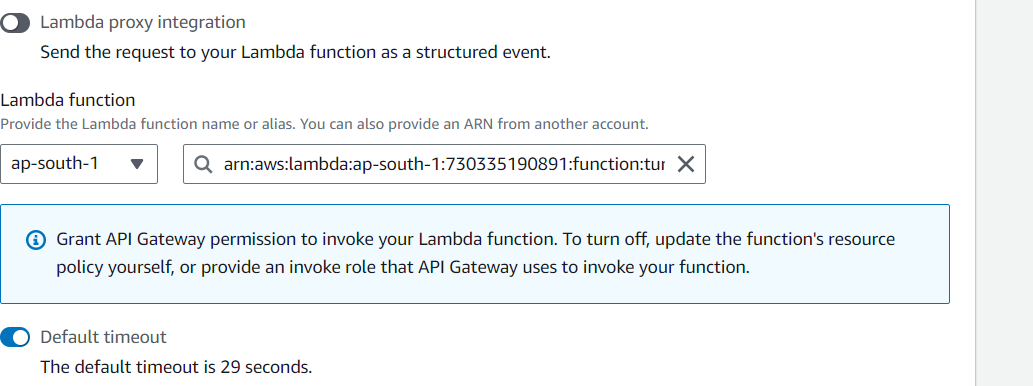
If you want to start the RDS instance on demand via an HTTP request, you can set up an API Gateway that triggers the Lambda function to start the RDS instance.

1. **Go to the API Gateway Console** and create a new API.
2. **Create a new resource and method**

Build a new REST API  
  


  
  
and then create a API

Click on API and create a resource  
  
  


Next create a method  
  
  
  
  
  
  
This can create a method

once everything is configured we will get an invoke url like this

Hit this url , this can turn on the RDS instance

