How do I stop and start Amazon EC2 instances at regular intervals using Lambda?

Last updated: 2020-10-21

I want to reduce my Amazon Elastic Compute Cloud (Amazon EC2) usage by stopping and starting my EC2 instances automatically. How do I use AWS Lambda and Amazon CloudWatch Events to do that?

Short description

Note: This example setup is a simple solution. For a more robust solution, use the AWS Instance Scheduler. For more information, see How do I stop and start my instances using the AWS Instance Scheduler?

For this setup, do the following:

1. Create a custom AWS Identity and Access Management (IAM) policy and execution role for your Lambda function.

2. Create Lambda functions that stop and start your EC2 instances.

3. Create CloudWatch Events rules that trigger your function on a schedule. For example, you could create a rule to stop your EC2 instances at night, and another to start them again in the morning.

Note: You can also create rules that trigger on an event that takes place in your AWS account.

Resolution

Get the IDs of the EC2 instances that you want to stop and start, and then follow these instructions.

Create an IAM policy and role

1. Create an IAM policy using the JSON policy editor. Paste this JSON policy document into the policy editor:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
        "Action": [
          "logs:CreateLogGroup",
          "logs:CreateLogStream",
          "logs:PutLogEvents"
        ],
        "Resource": "arn:aws:logs:*:*:*"
      },
      {
        "Effect": "Allow",
        "Action": [
          "ec2:Start*",
          "ec2:Stop*"
        ],
        "Resource": "*"
      }
    ]
}
```

2. Create an IAM role for Lambda. When attaching a permissions policy, search for and choose the IAM policy that you created.

Create Lambda functions that stop and start your EC2 instances

1. In the AWS Lambda console, choose **Create function**.

2. Choose **Author from scratch**.

3. Under **Basic information**, add the following:
For **Function name**, enter a name that identifies it as the function used to stop your EC2 instances. For example, "StopEC2Instances".
For **Runtime**, choose **Python 3.7**.
Under **Permissions**, expand **Choose or create an execution role**.
Under **Execution role**, choose **Use an existing role**.
Under **Existing role**, choose the IAM role that you created.

4. Choose **Create function**.

5. Copy this code, and then under **Function code**, paste it into the editor pane in the code editor (**lambda_function**). This code stops the EC2 instances that you identify.

Note: For **region**, replace "us-west-1" with the AWS Region that your instances are in. For **instances**, replace the example EC2 instance IDs with the IDs of the specific instances that you want to stop and start.

```
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

6. Under **Basic settings**, set **Timeout** to 10 seconds.

Note: Configure the Lambda function settings as needed for your use case. For example, if you want to stop and start multiple instances, you might need a different value for **Timeout**, as well as **Memory**.

7. Choose **Save**.

8. Repeat steps 1-7 to create another function. Do the following differently so that this function starts your EC2 instances:
In step 3, enter a **Function name** it as the function used to start your EC2 instances. For example, "StartEC2Instances".
In step 5, copy and paste this code into the editor pane in the code editor (**lambda_function**):

Note: For **region** and **instances**, use the same values that you used for the code to stop your EC2 instances.

```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

Test your Lambda functions

1. In the AWS Lambda console, choose **Functions**.

2. Select one of the functions that you created.

3. Choose **Actions**, and then choose **Test**.

4. In the **Configure test event** dialog, choose **Create new test event**.

Note: You don't need to change the JSON code for the test event—the function doesn't use it.

6. Choose **Test** to run the function.

7. Repeat steps 1-6 for the other function that you created.

Tip: You can check the status of your EC2 instances before and after testing to confirm that your functions work as expected.

Create rules that trigger your Lambda functions

1. Open the Amazon CloudWatch console.

2. In the left navigation pane, under **Events**, choose **Rules**.

3. Choose **Create rule**.

4. Under **Event Source**, choose **Schedule**.

5. Do either of the following:
For **Fixed rate of**, enter an interval of time in minutes, hours, or days.
For **Cron expression**, enter an expression that tells Lambda when to stop your instances. For information on the syntax of expressions, see Schedule expressions for rules.
Note: Cron expressions are evaluated in UTC. Be sure to adjust the expression for your preferred time zone.

6. Under **Targets**, choose **Add target**.

7. Choose **Lambda function**.

8. For **Function**, choose the function that stops your EC2 instances.

9. Choose **Configure details**.

10. Under **Rule definition**, do the following:
For **Name**, enter a name to identify the rule, such as "StopEC2Instances".
(Optional) For **Description**, describe your rule. For example, "Stops EC2 instances every night at 10 PM."
For **State**, select the **Enabled** check box.

11. Choose **Create rule**.

12. Repeat steps 1-11 to create a rule to start your EC2 instances. Do the following differently:
In step 5, for **Cron expression**, enter an expression that tells Lambda when to start your instances.
In step 8, for **Function**, choose the function that starts your EC2 instances.

Related information

Tutorial: Schedule AWS Lambda functions using CloudWatch Events

CloudWatch Events event examples from supported services

Adding stop actions to Amazon CloudWatch alarms

Scheduled Reserved Instances

Did this article help?

Submit feedback

Do you need billing or technical support?

Contact AWS Support

Contact Us
AWS Careers
File a Support Ticket
Knowledge Center
AWS Support Overview
Legal

Sign In to the Console

Language
عربي |
Bahasa Indonesia |
Deutsch |
English |
Español |
Français |
Italiano |
Português |
Tiếng Việt |
Türkçe |
Русский |
ไทย |
日本語 |
한국어 |
中文 (简体) |
中文 (繁體)

Privacy
|
Site Terms
|
Cookie Preferences
|