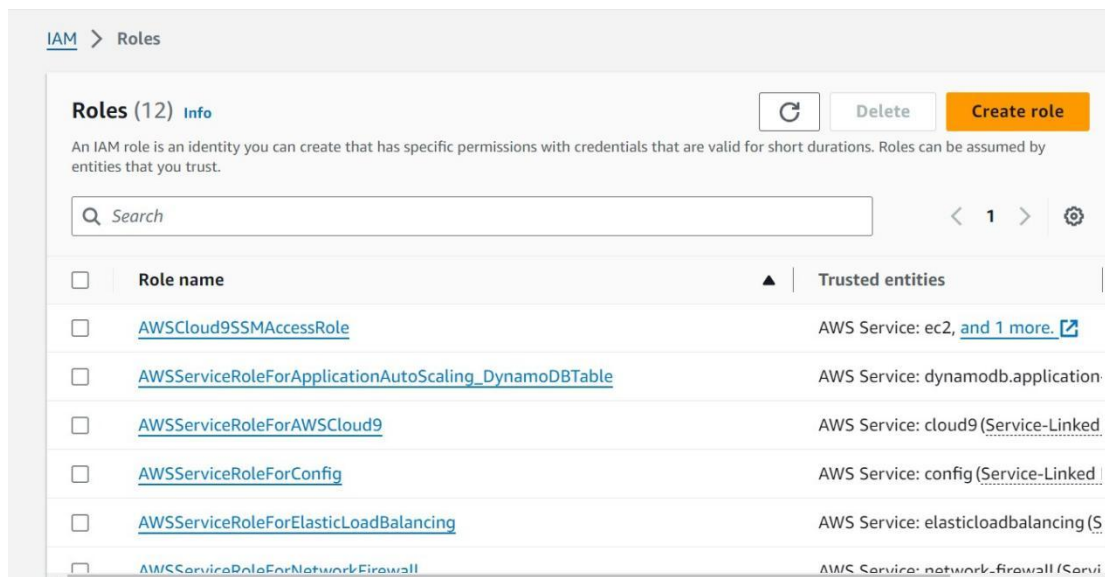


# EVENTBRIDGE AUTOMATION WITH LAMBDA FUNCTION TO DELETE EC2 INSTANCE

- 1) Create a IAM role and give it access of fullEC2, S3 and Lambda access.
- 2) Create a Lambda Function.
- 3) Create a Event Bridge Rule.
- 4) Select the Sample Event and Event Source.
- 5) Select Method and Event Pattern.
- 6) Select Target
- 7) Apply and Test by creating Instance
- 8) Check Logs

## STEP 1 :- Created IAM Role as Lambda\_Full\_Access



## STEP 2 :- Creating Lambda Function.

The screenshot shows the 'Create function' page in the AWS Lambda console. The breadcrumb navigation at the top reads 'Lambda > Functions > Create function'. The main heading is 'Create function' with an 'Info' link. Below the heading, a prompt says 'Choose one of the following options to create your function.' There are three radio button options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Author from scratch' option has a subtext: 'Start with a simple Hello World example.' Below these options is a section titled 'Basic information'. It contains three fields: 'Function name' with the value 'Delete\_EC2', 'Runtime' with the value 'Python 3.12', and 'Architecture' with the value 'x86\_64'. The 'Permissions' section is partially visible at the bottom.

[Lambda](#) > [Functions](#) > Create function

### Create function [Info](#)

Choose one of the following options to create your function.

- ☒ **Author from scratch**  
Start with a simple Hello World example.
- ☐ **Use a blueprint**  
Build a Lambda application from sample code and configuration presets for common use cases.
- ☐ **Container image**  
Select a container image to deploy for your function.

#### Basic information

**Function name**  
Enter a name that describes the purpose of your function.  
  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime** [Info](#)  
Choose the language to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  
☒ x86\_64  
☐ arm64

**Permissions** [Info](#)  
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

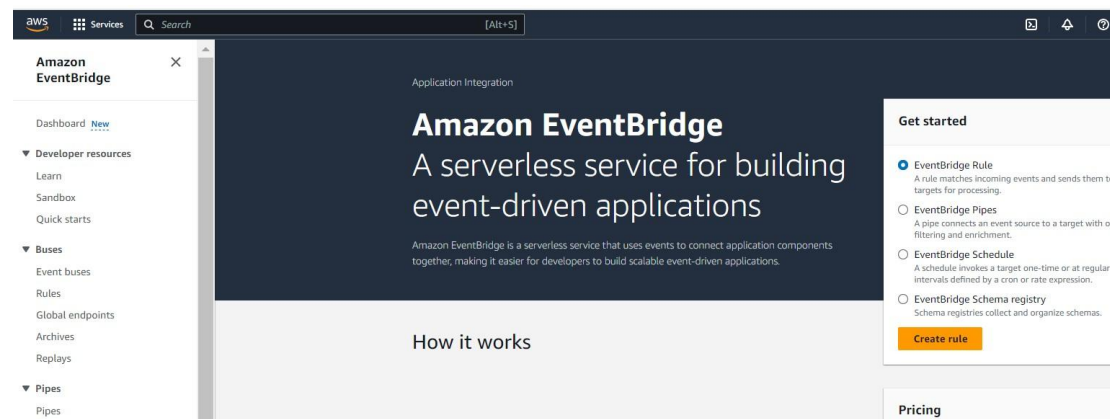
[► Change default execution role](#)

## STEP 3 :- Lambda Code in Python for deleting EC2 Instance.

The screenshot shows the 'Code source' page in the AWS Lambda console. The breadcrumb navigation at the top reads 'Lambda > Functions > Create function > Code source'. The main heading is 'Code source' with an 'Info' link. Below the heading, there is a toolbar with 'Test' and 'Deploy' buttons. The code editor shows the following Python code:

```
1 import boto3
2 import json
3
4 def lambda_handler(event, context):
5
6     print("event: " + json.dumps(event))
7     # Extract instance details from the EventBridge event
8     instance_id = event['detail']['instance-id']
9     print("Instance id: " + instance_id)
10
11     ec2_client = boto3.client('ec2')
12     print(f"Terminating instance with id: {instance_id}")
13     ec2_client.terminate_instances(InstanceIds=[instance_id])
```

## STEP 4 :- Creating Event Bridge



## STEP 5 :- Follow the steps.

### Build event pattern [Info](#)

#### Event source

Event source  
Select the event source from which events are sent.

☒ **AWS events or EventBridge partner events**  
Events sent from AWS services or EventBridge partners.

☐ **Other**  
Custom events or events sent from more than one source, e.g. events from AWS services and partners.

☐ **All events**  
All events sent to your account.

#### Sample event - optional

You don't have to select or enter a sample event, but it's recommended so you can reference it when writing and testing the event pattern, or filter criteria.

You can reference the sample event when you write the event pattern, or use the sample event to test if it matches the event pattern. Find a sample event, enter your own, or edit a sample event below. [Learn more about the required fields in a sample event.](#)

Sample event type

☒ **AWS events** ☐ EventBridge partner events ☐ Enter my own

Sample events  
Filter by event source and type or by keyword.

Select ▼

1

## STEP 6 :- Follow the Steps

Method

☐ Use schema  
Use an Amazon EventBridge schema to generate the event pattern.

☒ Use pattern form  
Use a template provided by EventBridge to create an event pattern.

☐ Custom pattern (JSON editor)  
Write an event pattern in JSON.

Event pattern [Info](#)

Event source  
AWS service or EventBridge partner as source

AWS services ▼

AWS service  
The name of the AWS service as the event source

EC2 ▼

Event type  
The type of events as the source of the matching pattern

EC2 Instance State-change Notification ▼

Event Type Specification 1

☐ Any state

☒ Specific state(s)

Specific state(s)

▼

pending ✕

Event Type Specification 2

☒ Any instance

☐ Specific instance Id(s)

Event pattern  
Event pattern, or filter to match the events

```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"],
4   "detail": {
5     "state": ["pending"]
6   }
7 }
```


Copy

Test pattern

Edit pattern

## STEP 7 :- Select Target.

### Select target(s)

**Permissions**

Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.


#### Target 1

**Target types**  
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

☐ EventBridge event bus  
☐ EventBridge API destination  
☒ AWS service

**Select a target** [Info](#)  
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

Lambda function ▼

**Function**  
Delete\_EC2 ▼ 

► **Configure version/alias**

► **Additional settings**

Add another target

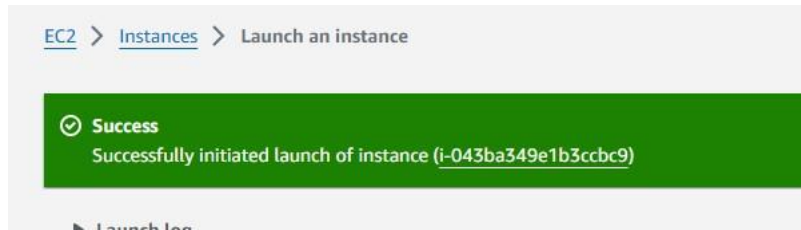
Cancel

Skip to Review and create

Previous

Next

## STEP 8 :- Test on EC2 Instance.



## STEP 9 :- Check the Logs.

