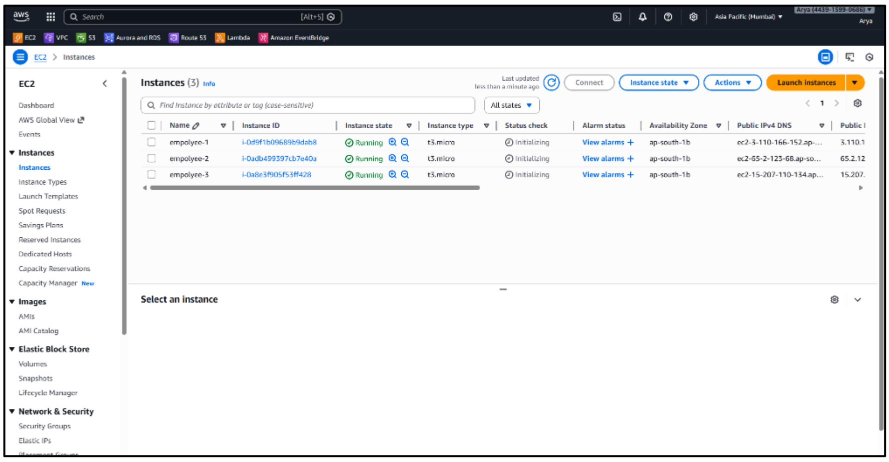
Stop And Start 3 instances by using Lambda Function and EventBridge

Step 1: First, create 3 Instances with your configuration.



Instances (3) Info

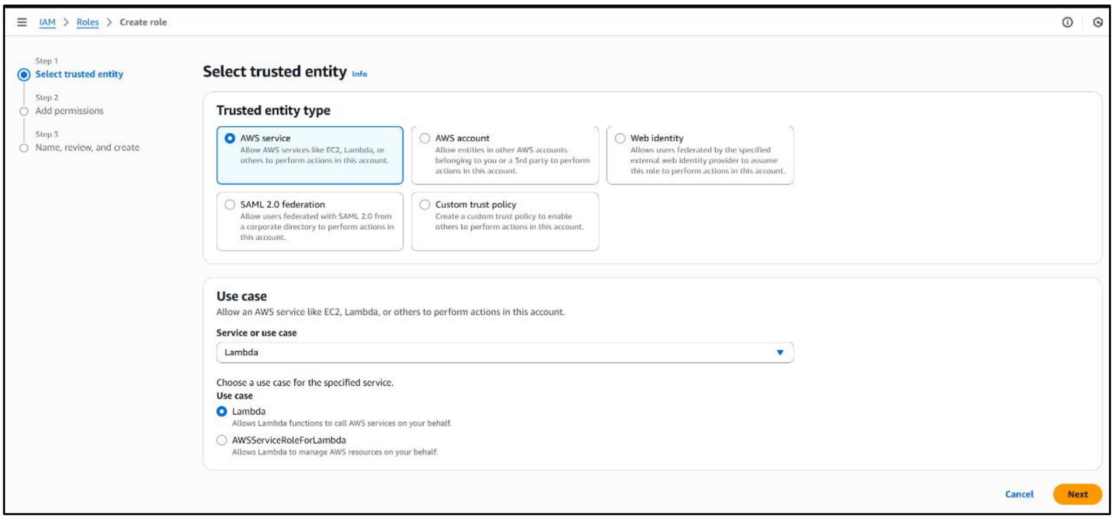
Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public I
□	empolyee-1	i-0d9f1b09689b9dab8	Running	t3.micro	Initializing	View alarms +	ap-south-1b	ec2-3-110-166-152.ap-...	3.110.1
□	empolyee-2	i-0adb499397cb7e40a	Running	t3.micro	Initializing	View alarms +	ap-south-1b	ec2-65-2-123-68.ap-so...	65.2.12
□	empolyee-3	i-0a8e3f905f53ff428	Running	t3.micro	Initializing	View alarms +	ap-south-1b	ec2-15-207-110-134.ap...	15.207.

Select an instance

Step 2: Now you need an IAM Role to give to the lambda function creation (EC2fullAccess).



Select trusted entity

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

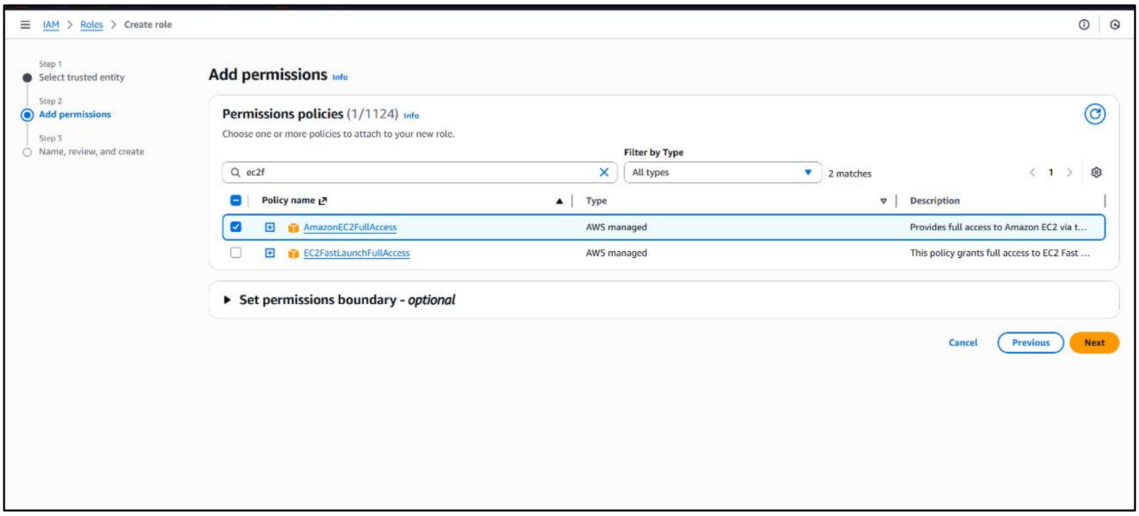
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

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.
- ○ AWSServiceRoleForLambda
  Allows Lambda to manage AWS resources on your behalf.



Add permissions

Permissions policies (1/1124)

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

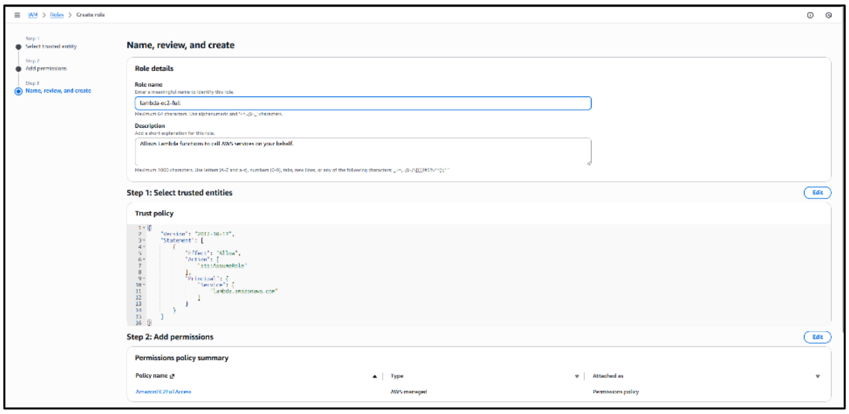
Search: ec2f

Filter by Type: All types

2 matches

□	Policy name	Type	Description
☑	AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via t...
□	EC2FastLaunchFullAccess	AWS managed	This policy grants full access to EC2 Fast ...

Stop And Start 3 instances by using Lambda Function and EventBridge



IAM > Roles > Create role

Step 1 Select trusted entity
Step 2 Add permissions
Step 3 Name, review, and create

Name, review, and create

Role details

Role name: lambda-ec2-full

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

Step 1: Select trusted entities

Trust policy

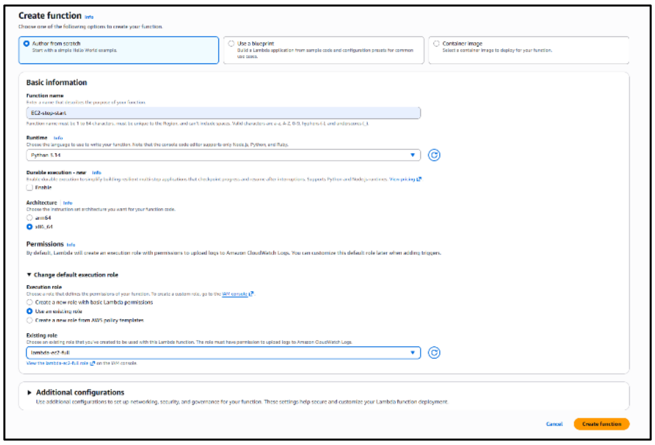
```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Principal": {
                "Service": [
                    "lambda.amazonaws.com"
                ]
            }
        }
    ]
}
```

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2FullAccess	AWS managed	Permissions policy

Step 3: After creating a role, create a Lambda function.



Create function

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: EC2-stop-start

Runtime: Python 3.14

Durable execution - new: □ Enable

Architecture: ○ arm64 ● x86_64

Permissions: 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

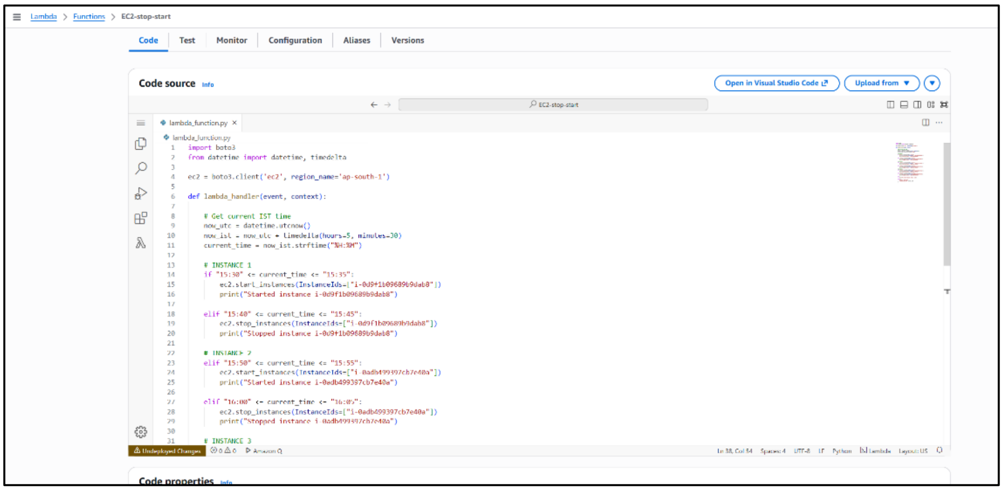
Execution role:
○ Create a new role with basic Lambda permissions
● Use an existing role
○ Create a new role from AWS policy templates

Existing role: lambda-ec2-full

▶ Additional configurations

In this, select Python 3.14 because we are writing code in Python to start and stop instances. And also, in Change default execution role, select Use existing role and select the role we created here.

Step 4: Now write code or paste code in the code tab. And change region, instance ID, and timing.



Lambda > Functions > EC2-stop-start

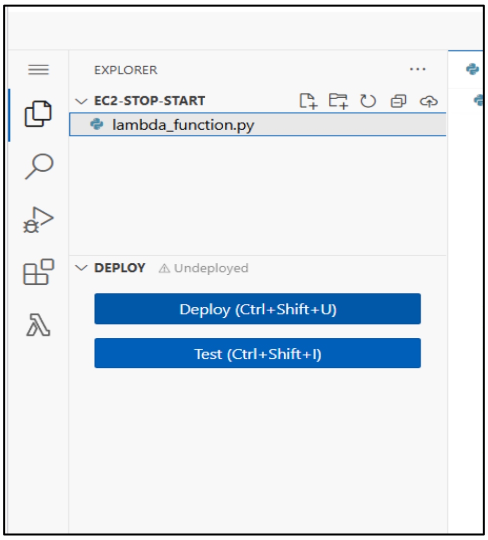
Code | Test | Monitor | Configuration | Aliases | Versions

Code source

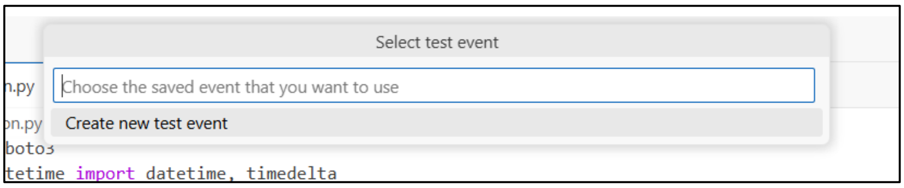
lambda_function.py

```python
import boto3
from datetime import datetime, timedelta

ec2 = boto3.client('ec2', region_name='ap-south-1')

def lambda_handler(event, context):

    # Get current IST time
    now_utc = datetime.utcnow()
    now_ist = now_utc + timedelta(hours=5, minutes=30)
    current_time = now_ist.strftime("%H:%M")

    # INSTANCE 1
    if "15:30" <= current_time <= "15:35":
        ec2.start_instances(InstanceIds=["i-0d9f1b09689b9dab8"])
        print("Started instance i-0d9f1b09689b9dab8")

    elif "15:40" <= current_time <= "15:45":
        ec2.stop_instances(InstanceIds=["i-0d9f1b09689b9dab8"])
        print("Stopped instance i-0d9f1b09689b9dab8")

    # INSTANCE 2
    elif "15:50" <= current_time <= "15:55":
        ec2.start_instances(InstanceIds=["i-0adb499397cb7e40a"])
        print("Started instance i-0adb499397cb7e40a")

    elif "16:00" <= current_time <= "16:05":
        ec2.stop_instances(InstanceIds=["i-0adb499397cb7e40a"])
        print("Stopped instance i-0adb499397cb7e40a")

    # INSTANCE 3
```

Step 5: After the changes, click on deploy the code to save it and click on test to create a test to run the code.

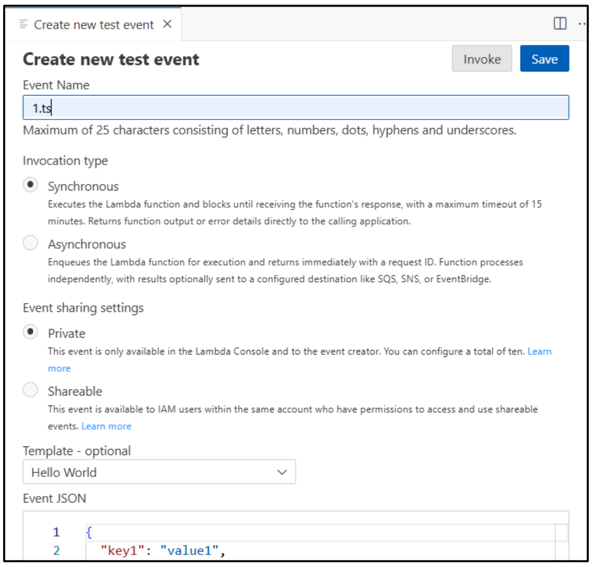


Step 6: After this, there is a window in pooped to create a test click to create a new test event.



```
n.py
on.py
boto3
tetime import datetime, timedelta
```

Step 7: Give the name to the Test event and save to only.



Create new test event

Event Name

1.ts

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Invocation type

● Synchronous
Executes the Lambda function and blocks until receiving the function's response, with a maximum timeout of 15 minutes. Returns function output or error details directly to the calling application.

○ Asynchronous
Enqueues the Lambda function for execution and returns immediately with a request ID. Function processes independently, with results optionally sent to a configured destination like SQS, SNS, or EventBridge.

Event sharing settings

● Private
This event is only available in the Lambda Console and to the event creator. You can configure a total of ten. Learn more

○ Shareable
This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

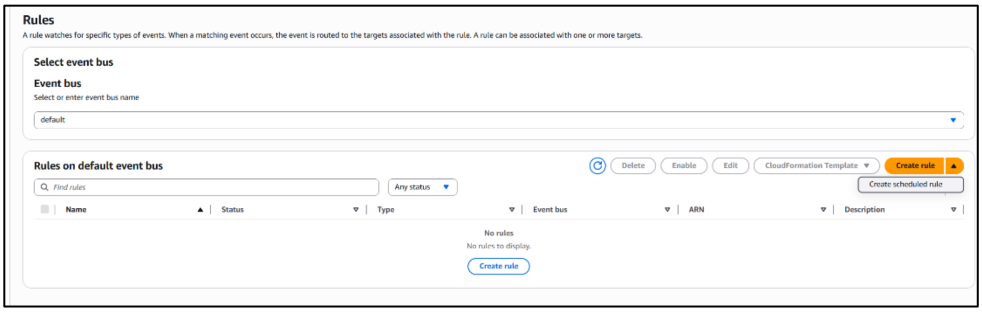
Template - optional

Hello World

Event JSON

```
{
  "key1": "value1",
```

Step 8: Now go to Event Bridge, in that Bus got rule and create a scheduled rule.



Rules

A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

Select event bus

Event bus

Select or enter event bus name

default

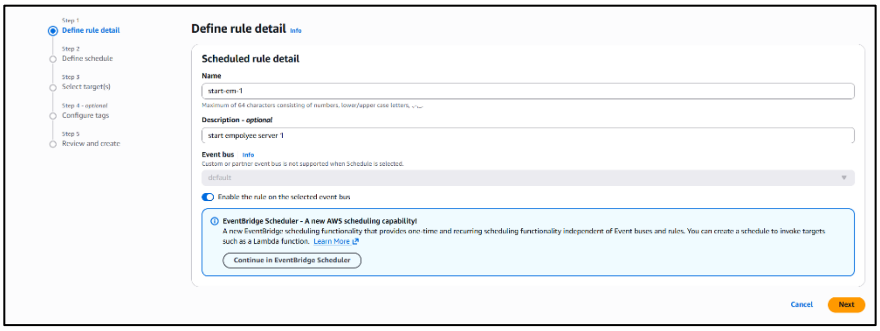
Rules on default event bus

Name	Status	Type	Event bus	ARN	Description

No rules

No rules to display.

Step 9: In this give a name to the rule – we create a start rule, for instance and click next.



Define rule detail Info

Scheduled rule detail

Name: start-em-1

Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-,_.

Description - *optional*: start empolyee server 1

Event bus Info

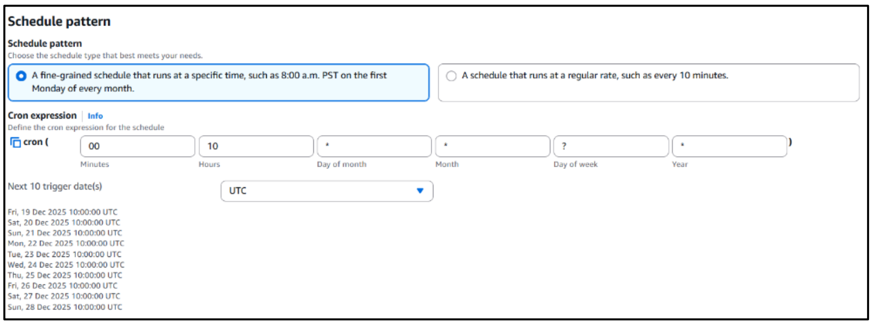
Custom or partner event bus is not supported when Schedule is selected.

default

● Enable the rule on the selected event bus

EventBridge Scheduler - A new AWS scheduling capability!
A new EventBridge scheduling functionality that provides one-time and recurring scheduling functionality independent of Event buses and rules. You can create a schedule to invoke targets such as a Lambda function. Learn More

Step 10: Next, create a pattern or give the crontab. Here you have to give Minutes, hours, Day of month, month, day of week, and year. Here in 3 or 5 position, you have to give **"?"** for all selections.



Schedule pattern

Schedule pattern

Choose the schedule type that best meets your needs.

● A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

○ A schedule that runs at a regular rate, such as every 10 minutes.

Cron expression Info

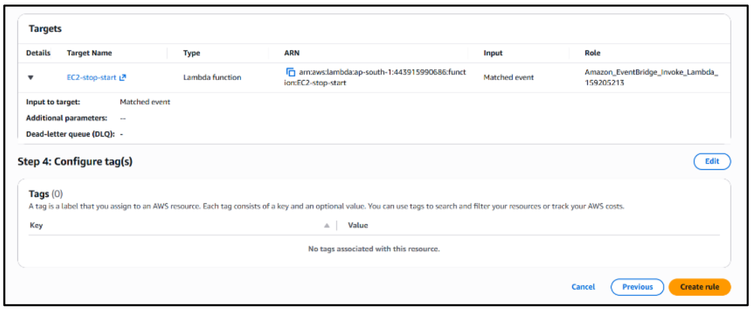
Define the cron expression for the schedule

	Minutes	Hours	Day of month	Month	Day of week	Year	
cron (	00	10	*	*	?	*	)

Next 10 trigger date(s) — UTC

```
Fri, 19 Dec 2025 10:00:00 UTC
Sat, 20 Dec 2025 10:00:00 UTC
Sun, 21 Dec 2025 10:00:00 UTC
Mon, 22 Dec 2025 10:00:00 UTC
Tue, 23 Dec 2025 10:00:00 UTC
Wed, 24 Dec 2025 10:00:00 UTC
Thu, 25 Dec 2025 10:00:00 UTC
Fri, 26 Dec 2025 10:00:00 UTC
Sat, 27 Dec 2025 10:00:00 UTC
Sun, 28 Dec 2025 10:00:00 UTC
```

Step 11: Now here select the function – lambda function and select the ARN of your function, and in execution role - create a new role of specific resource.



Targets

Details	Target Name	Type	ARN	Input	Role
▼	EC2-stop-start	Lambda function	arn:aws:lambda:ap-south-1:443915990686:function:EC2-stop-start	Matched event	Amazon_EventBridge_Invoke_Lambda_159205213

Input to target: Matched event

Additional parameters: --

Dead-letter queue (DLQ): -

Step 4: Configure tag(s)

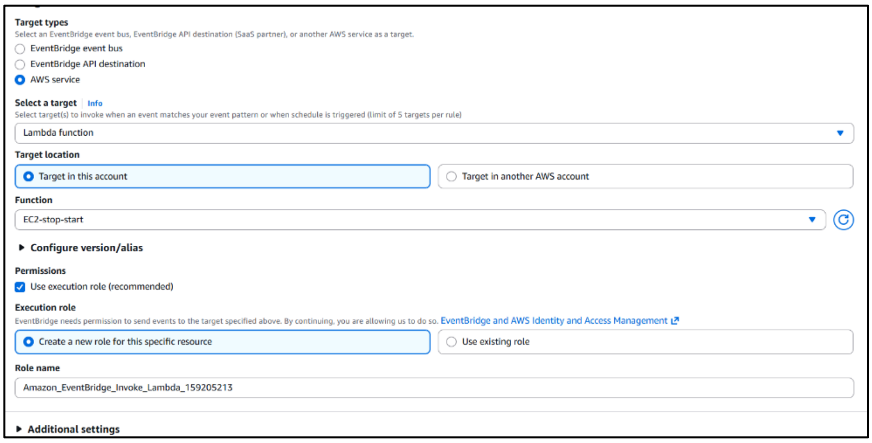
Tags (0)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value

No tags associated with this resource.

Stop And Start 3 instances by using Lambda Function and EventBridge



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

Target location

● Target in this account ○ Target in another AWS account

Function

EC2-stop-start

▸ **Configure version/alias**

Permissions

☑ Use execution role (recommended)

Execution role

EventBridge needs permission to send events to the target specified above. By continuing, you are allowing us to do so. EventBridge and AWS Identity and Access Management

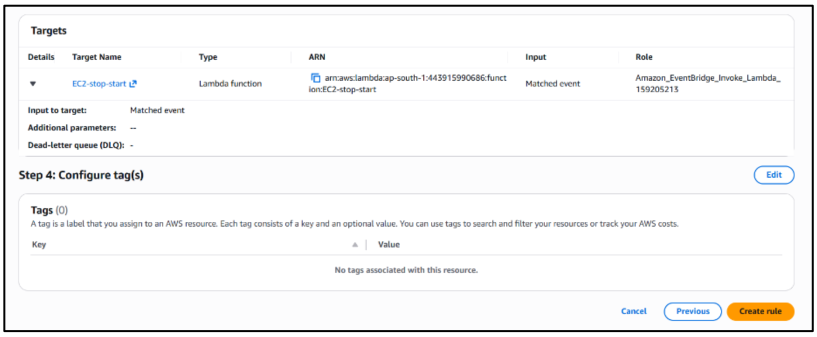
● Create a new role for this specific resource ○ Use existing role

Role name

Amazon_EventBridge_Invoke_Lambda_159205213

▸ **Additional settings**

Step 12: Now, create a rule.



Targets

Details	Target Name	Type	ARN	Input	Role
▾	EC2-stop-start	Lambda function	arn:aws:lambda:ap-south-1:443915990686:function:EC2-stop-start	Matched event	Amazon_EventBridge_Invoke_Lambda_159205213

Input to target: Matched event

Additional parameters: --

Dead-letter queue (DLQ): -

Step 4: Configure tag(s)

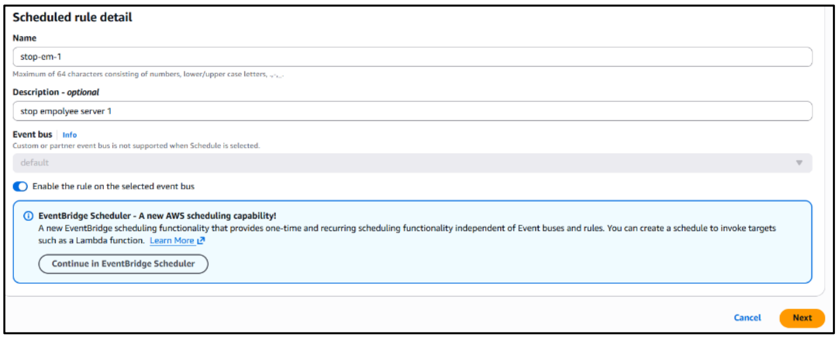
Tags (0)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value

No tags associated with this resource.

Step 13: Now, also create a rule to stop instances with time.



Scheduled rule detail

Name

stop-em-1

Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-,_.

Description - *optional*

stop empolyee server 1

Event bus | Info

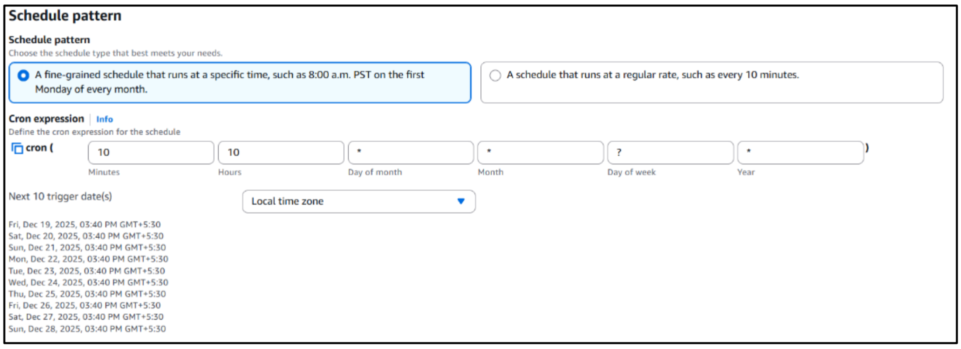
Custom or partner event bus is not supported when Schedule is selected.

default

● Enable the rule on the selected event bus

EventBridge Scheduler - A new AWS scheduling capability!

A new EventBridge scheduling functionality that provides one-time and recurring scheduling functionality independent of Event buses and rules. You can create a schedule to invoke targets such as a Lambda function. Learn More



Schedule pattern

Schedule pattern

Choose the schedule type that best meets your needs.

● A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

○ A schedule that runs at a regular rate, such as every 10 minutes.

Cron expression | Info

Define the cron expression for the schedule

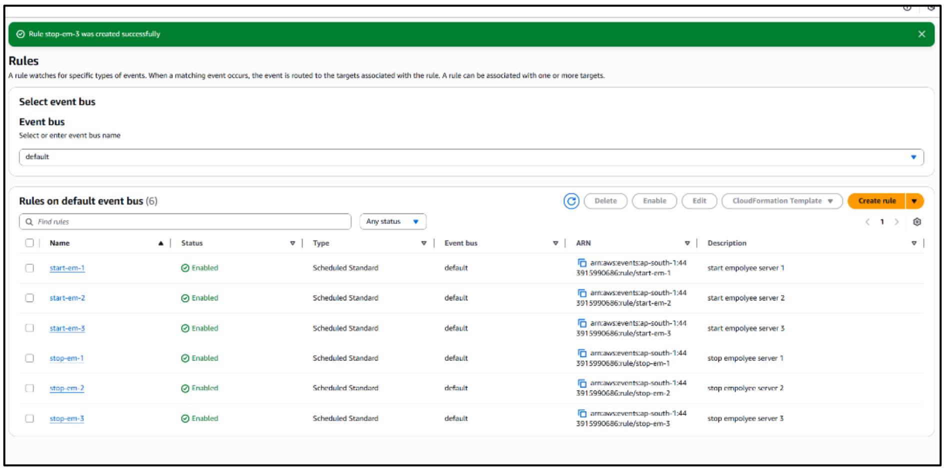
cron (	10	10	*	*	?	*	)
	Minutes	Hours	Day of month	Month	Day of week	Year	

Next 10 trigger date(s) — Local time zone

```
Fri, Dec 19, 2025, 03:40 PM GMT+5:30
Sat, Dec 20, 2025, 03:40 PM GMT+5:30
Sun, Dec 21, 2025, 03:40 PM GMT+5:30
Mon, Dec 22, 2025, 03:40 PM GMT+5:30
Tue, Dec 23, 2025, 03:40 PM GMT+5:30
Wed, Dec 24, 2025, 03:40 PM GMT+5:30
Thu, Dec 25, 2025, 03:40 PM GMT+5:30
Fri, Dec 26, 2025, 03:40 PM GMT+5:30
Sat, Dec 27, 2025, 03:40 PM GMT+5:30
Sun, Dec 28, 2025, 03:40 PM GMT+5:30
```

Like this, also create start and stop rules for other instances. After this, you have to Associate it with a trigger in a lambda function.

Stop And Start 3 instances by using Lambda Function and EventBridge



Rule stop-em-3 was created successfully

Rules

A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

Select event bus

Event bus

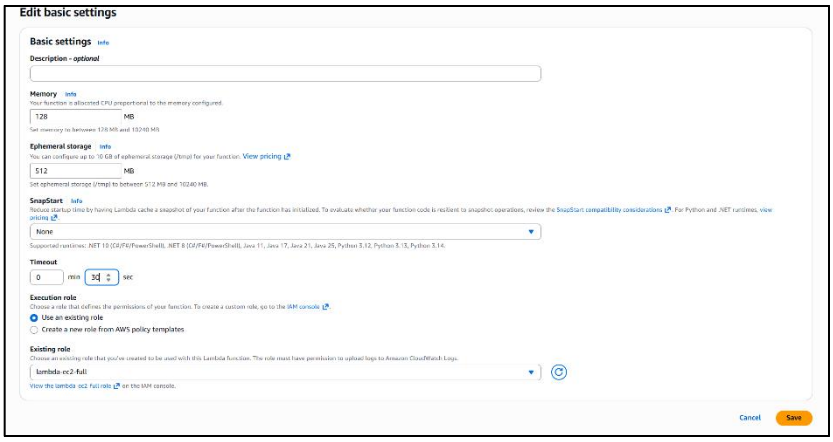
Select or enter event bus name

default

Rules on default event bus (6)

□	Name	Status	Type	Event bus	ARN	Description
□	start-em-1	Enabled	Scheduled Standard	default	arn:aws:events:ap-south-1:443915990686:rule/start-em-1	start empolyee server 1
□	start-em-2	Enabled	Scheduled Standard	default	arn:aws:events:ap-south-1:443915990686:rule/start-em-2	start empolyee server 2
□	start-em-3	Enabled	Scheduled Standard	default	arn:aws:events:ap-south-1:443915990686:rule/start-em-3	start empolyee server 3
□	stop-em-1	Enabled	Scheduled Standard	default	arn:aws:events:ap-south-1:443915990686:rule/stop-em-1	stop empolyee server 1
□	stop-em-2	Enabled	Scheduled Standard	default	arn:aws:events:ap-south-1:443915990686:rule/stop-em-2	stop empolyee server 2
□	stop-em-3	Enabled	Scheduled Standard	default	arn:aws:events:ap-south-1:443915990686:rule/stop-em-3	stop empolyee server 3

Step 14: Now go to the Lambda Function, then go to configure in that first General configuration, change the Timeout from 3 sec to 30 sec, and save it.



Edit basic settings

Basic settings Info

Description - *optional*

Memory Info

Your function is allocated CPU proportional to the memory configured.

128 MB

Set memory to between 128 MB and 10240 MB

Ephemeral storage Info

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. View pricing

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart Info

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the SnapStart compatibility considerations. For Python and .NET runtimes, view pricing.

None

Supported runtimes: .NET 10 (C#/F#/PowerShell), .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Java 25, Python 3.12, Python 3.13, Python 3.14.

Timeout

0 min 30 sec

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

● Use an existing role

○ Create a new role from AWS policy templates

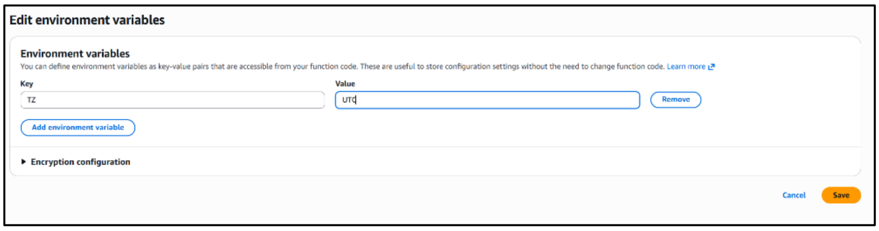
Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

lambda-ec2-full

View the lambda-ec2-full role on the IAM console.

Step 15: Now, go to environment variables and add the variable here. **Key – TZ and Value – UTC**.



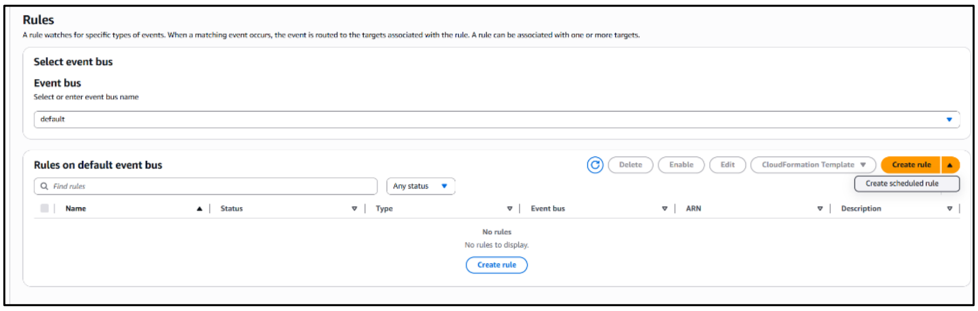
Edit environment variables

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

Key: TZ

Value: UTC



Rules

A rule watches for specific types of events. When a matching event occurs, the event is routed to the targets associated with the rule. A rule can be associated with one or more targets.

Select event bus

Event bus

Select or enter event bus name

default

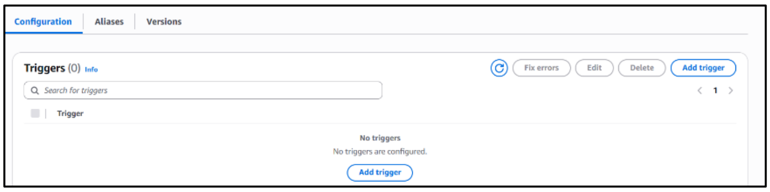
Rules on default event bus

□	Name	Status	Type	Event bus	ARN	Description

No rules

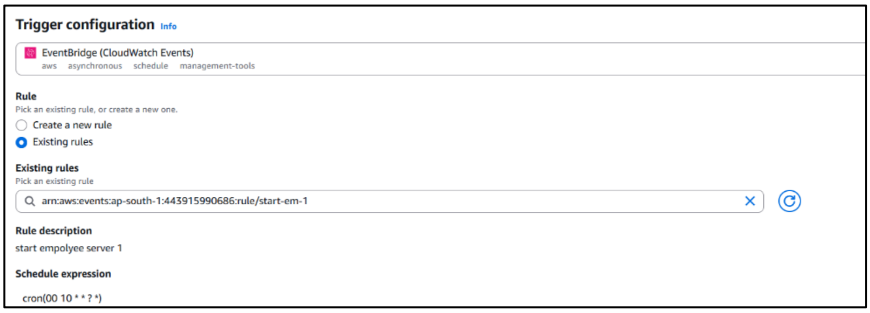
No rules to display.

Step 16: Now go to Trigger and add the rule you created one by one.



No triggers

No triggers are configured.



EventBridge (CloudWatch Events)
aws asynchronous schedule management-tools

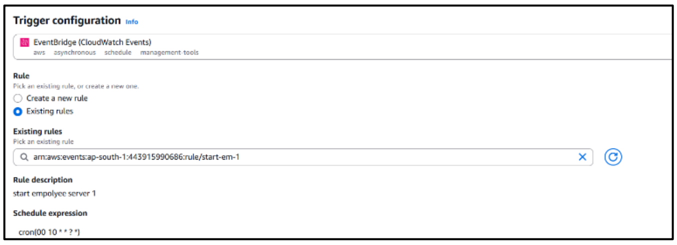
Rule
Pick an existing rule, or create a new one.
○ Create a new rule
● Existing rules

Existing rules
Pick an existing rule
arn:aws:events:ap-south-1:443915990686:rule/start-em-1

Rule description
start empolyee server 1

Schedule expression
cron(00 10 * * ? *)

Here, you have to select EventBridge and the existing rule, add select rule start and stop.



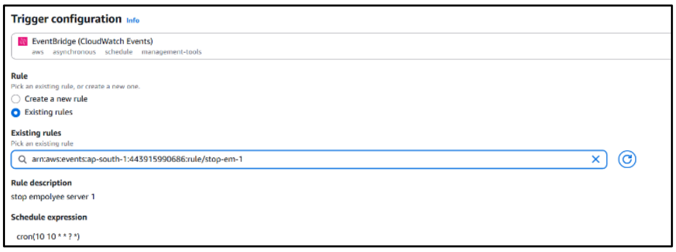
EventBridge (CloudWatch Events)
aws asynchronous schedule management-tools

Rule
Pick an existing rule, or create a new one.
○ Create a new rule
● Existing rules

Existing rules
Pick an existing rule
arn:aws:events:ap-south-1:443915990686:rule/start-em-1

Rule description
start empolyee server 1

Schedule expression
cron(00 10 * * ? *)



EventBridge (CloudWatch Events)
aws asynchronous schedule management-tools

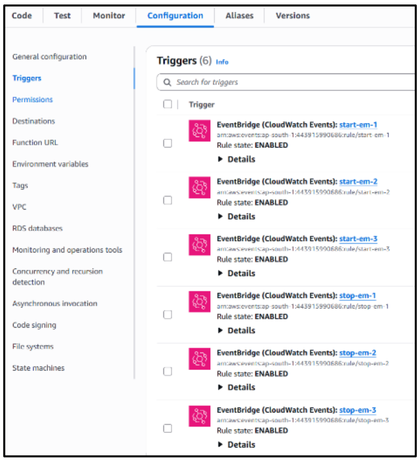
Rule
Pick an existing rule, or create a new one.
○ Create a new rule
● Existing rules

Existing rules
Pick an existing rule
arn:aws:events:ap-south-1:443915990686:rule/stop-em-1

Rule description
stop empolyee server 1

Schedule expression
cron(10 10 * * ? *)

Like this, add other rules to trigger the lambda function.



EventBridge (CloudWatch Events): start-em-1
arn:aws:events:ap-south-1:443915990686:rule/start-em-1
Rule state: ENABLED
▸ Details

EventBridge (CloudWatch Events): start-em-2
arn:aws:events:ap-south-1:443915990686:rule/start-em-2
Rule state: ENABLED
▸ Details

EventBridge (CloudWatch Events): start-em-3
arn:aws:events:ap-south-1:443915990686:rule/start-em-3
Rule state: ENABLED
▸ Details

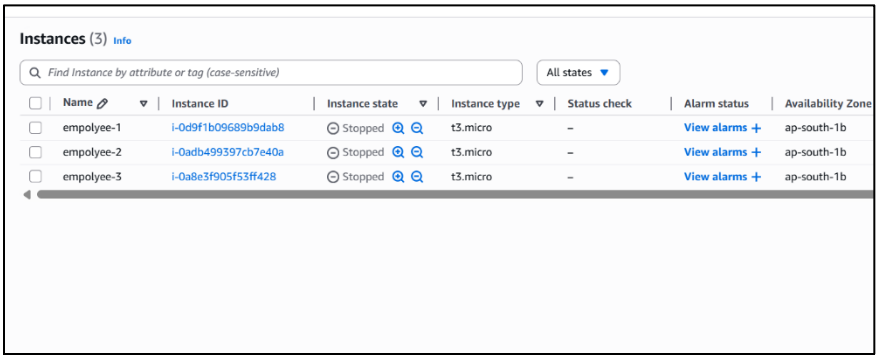
EventBridge (CloudWatch Events): stop-em-1
arn:aws:events:ap-south-1:443915990686:rule/stop-em-1
Rule state: ENABLED
▸ Details

EventBridge (CloudWatch Events): stop-em-2
arn:aws:events:ap-south-1:443915990686:rule/stop-em-2
Rule state: ENABLED
▸ Details

EventBridge (CloudWatch Events): stop-em-3
arn:aws:events:ap-south-1:443915990686:rule/stop-em-3
Rule state: ENABLED
▸ Details

Stop And Start 3 instances by using Lambda Function and EventBridge

Now you can see our instances are stop.

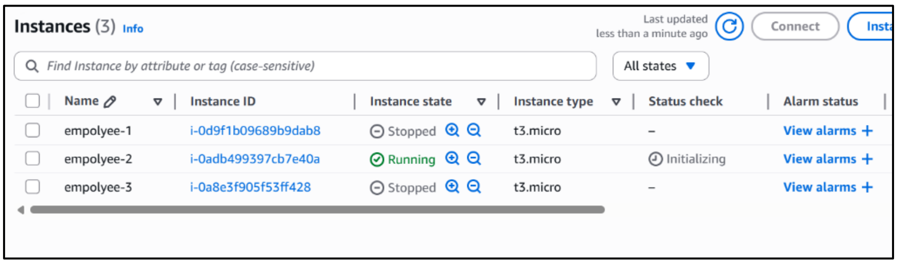


Instances (3) Info

Find Instance by attribute or tag (case-sensitive) | All states

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	empolyee-1	i-0d9f1b09689b9dab8	Stopped	t3.micro	–	View alarms +	ap-south-1b
☐	empolyee-2	i-0adb499397cb7e40a	Stopped	t3.micro	–	View alarms +	ap-south-1b
☐	empolyee-3	i-0a8e3f905f53ff428	Stopped	t3.micro	–	View alarms +	ap-south-1b

And then in some time instance, second in the running state.



Instances (3) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive) | All states

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	empolyee-1	i-0d9f1b09689b9dab8	Stopped	t3.micro	–	View alarms +
☐	empolyee-2	i-0adb499397cb7e40a	Running	t3.micro	Initializing	View alarms +
☐	empolyee-3	i-0a8e3f905f53ff428	Stopped	t3.micro	–	View alarms +