Guided Lab: Amazon EC2 Management Using EventBridge for Scheduled Start/Stop

Description

Efficient resource management is essential in cloud computing to optimize costs and ensure operational efficiency. Automating the start and stop of EC2 instances based on a schedule is a practical way to achieve this.

Imagine a software development company, **"Dojo Technology, Inc."** running multiple development environments on AWS. These environments are used during specific time of the day or a week or regular timeframe (for example 9 PM to 6 AM UTC) but remain idle outside these times, incurring unnecessary costs. By automating the start and stop of these EC2 instances, Tech Innovators can significantly reduce their AWS bill while ensuring resources are available when needed.

In this lab, you will use Amazon EventBridge, a serverless event bus service that enables the creation of event-driven architectures by routing events from various sources to AWS services like Lambda. This allows you to automate tasks and processes, ensuring optimal resource usage and cost management. You will learn how to set up EventBridge to trigger Lambda functions, which will start and stop your EC2 instances on a defined schedule, demonstrating the integration of EC2, Lambda, IAM, and EventBridge for automated instance management.

Prerequisites

This lab assume you have basic understanding and knowledge of Amazon EC2, Lambda Function(Python Programming Language), and EventBridge.

If you find any gaps in your knowledge, consider taking the following lab:

- Creating an Amazon EC2 instance (Linux)
- Creating an AWS Lambda function

Objectives

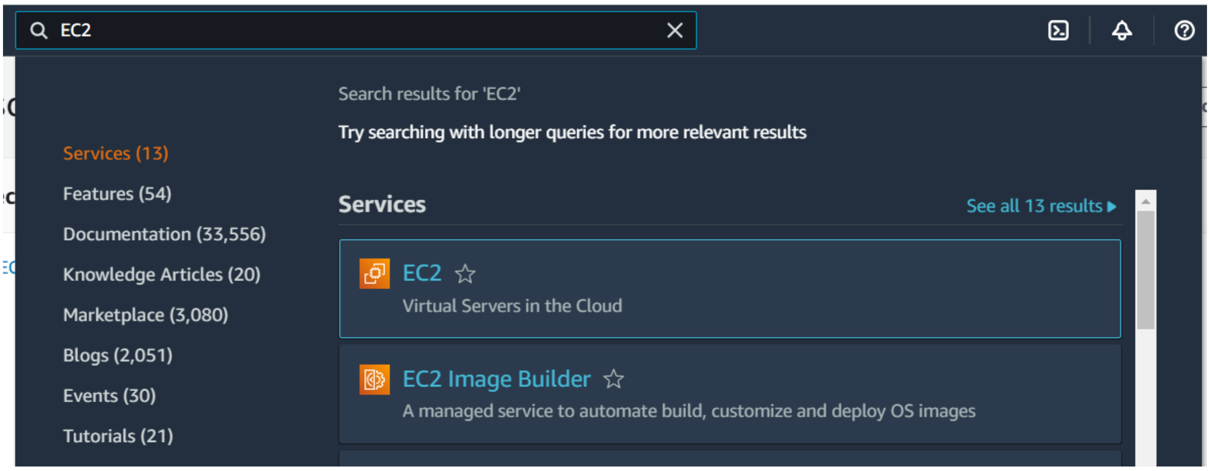
In this lab, you will:

- Create a Lambda function to start and stop an EC2 instance.
- Learn how to set up EventBridge rules to trigger events based on a schedule.
- Demonstrate the integration of EC2, Lambda, and EventBridge for automated instance management.

Lab Steps

Create an Amazon EC2 Instance

1. **Navigate the EC2 Dashboard.**

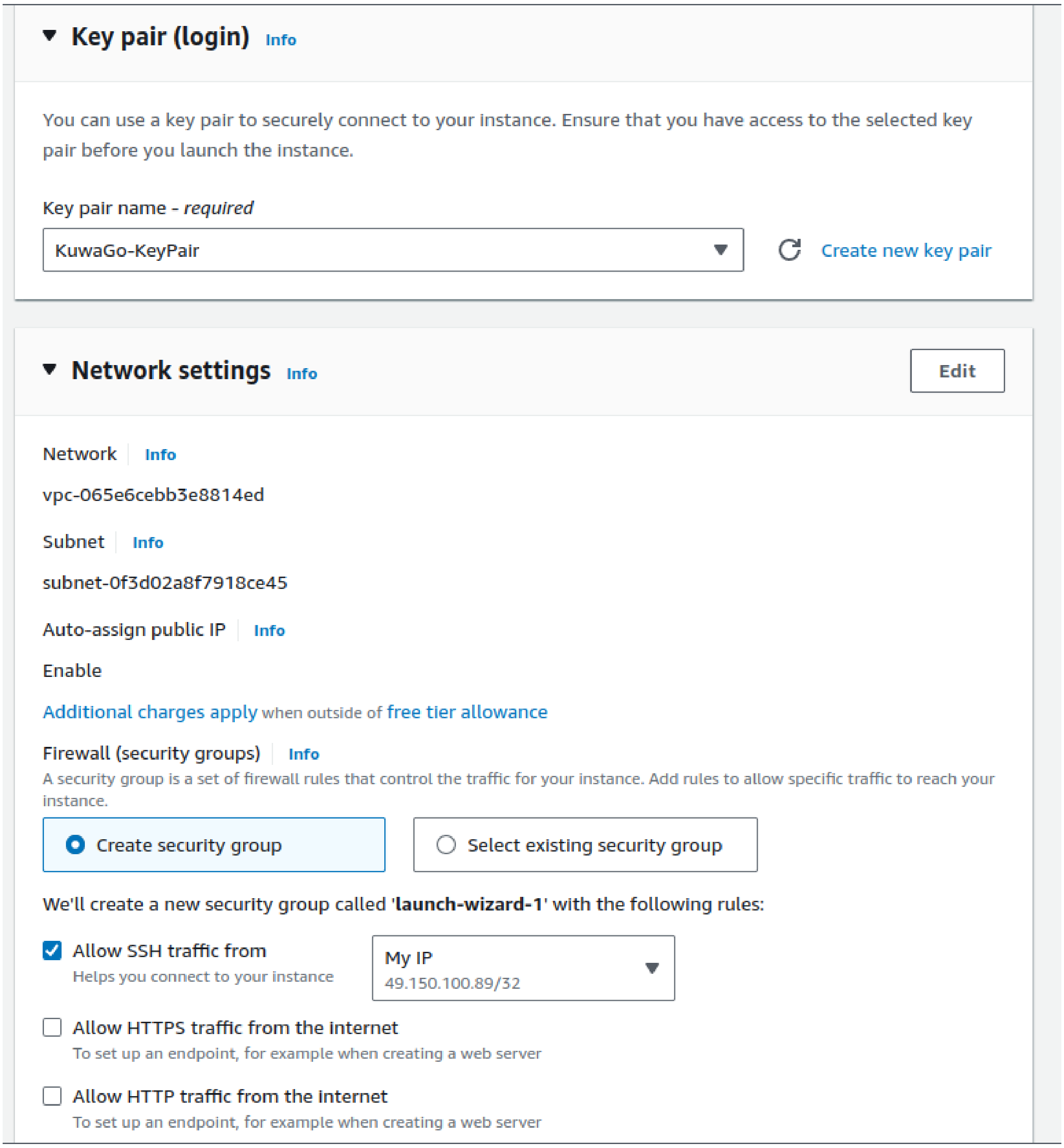

EC2

Search results for 'EC2'

Try searching with longer queries for more relevant results

2. **Launch an EC2 Instances using the following configurations:**

- Name: **WebServer-KuwaGo**
- AMI: **Amazon Linux**
- Instance type: **t2.micro**Key pair: (**Please create a new one.**)
    - Key pair name: **KuwaGo-KeyPair**
    - Key pair type: **RSA**
    - Private key file format: **.pem**
- Network settings: **(**Click **"Create security group")**
    - Auto-assign public IP: Select **Enable**
    - Firewall (security groups): tick on the **Create security group**
        - Ensure that **Allow SSH traffic from** is **checked** and is **My IP**



Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

KuwaGo-KeyPair

Network settings Info

Network Info

vpc-065e6cebb3e8814ed

Subnet Info

subnet-0f3d02a8f7918ce45

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from — My IP 49.150.100.89/32

Helps you connect to your instance

☐ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

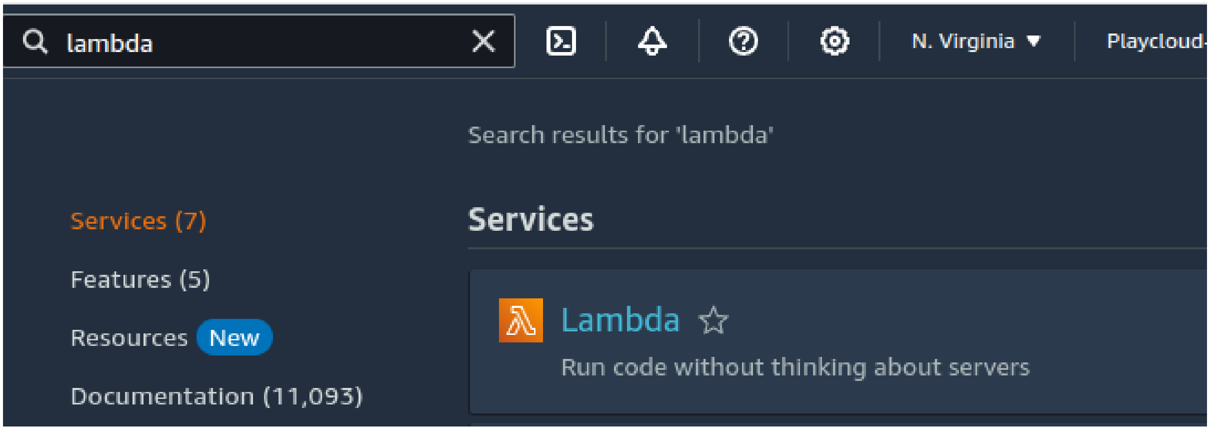
☐ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

- Click Launch Instance.

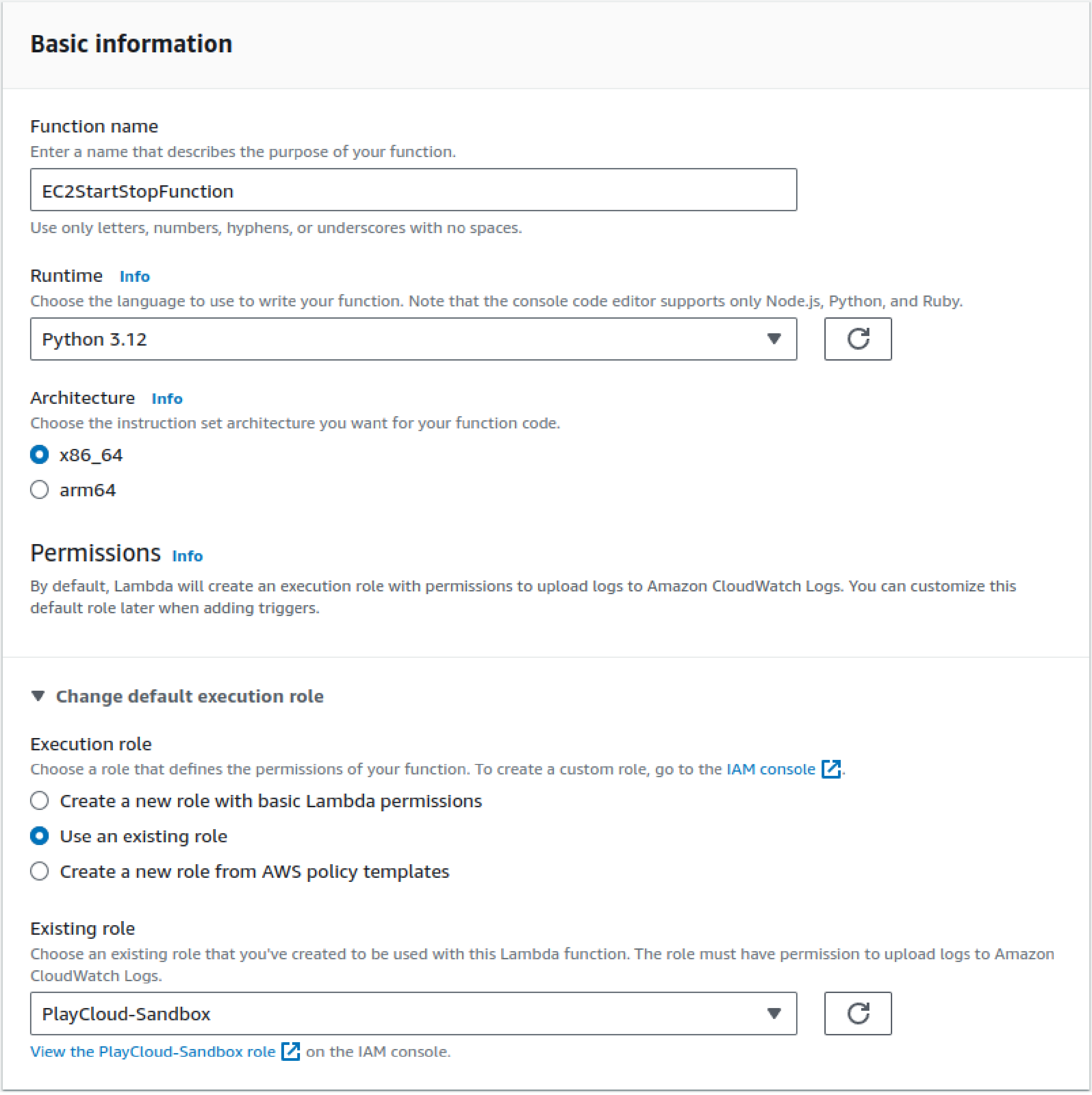
Create a Lambda Function

1. Navigate to the **AWS Lambda Dashboard**.



2.Create a Lambda function with the following Configurations:

- Choose **Author from scratch**.
- **Function name**: EC2StartStopFunction
- **Runtime**: Choose Python 3.12 or the latest one
- **Permissions:**
    - **Change default execution role** > **Execution role:** Select "**Use an existing role**" and choose the role PlayCloud-Sandbox



Basic information

Function name

Enter a name that describes the purpose of your function.

EC2StartStopFunction

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.12

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

Execution role

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

○ Create a new role with basic Lambda permissions

● 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.

PlayCloud-Sandbox

View the PlayCloud-Sandbox role on the IAM console.

- Click **Create function**.

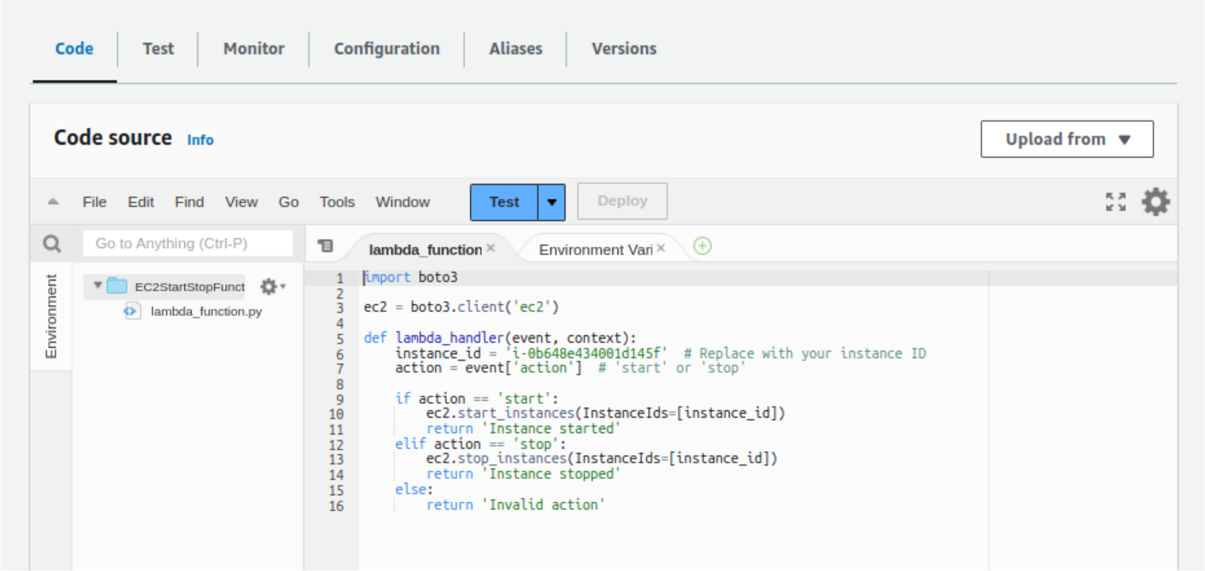
3. Add the following code to the Lambda function (Python example) Code Editor Tab:

Replace the PLACEHOLDER <EC2-INTANCE-ID> *with your instance ID*

```
import boto3


ec2 = boto3.client('ec2')


def lambda_handler(event, context):
    instance_id = '<EC2-INTANCE-ID>'  # Replace with your instance ID
    action = event['action']  # 'start' or 'stop'
```

```
if action == 'start':
    ec2.start_instances(InstanceIds=[instance_id])
    return 'Instance started'
elif action == 'stop':
    ec2.stop_instances(InstanceIds=[instance_id])
    return 'Instance stopped'
else:
    return 'Invalid action'
```

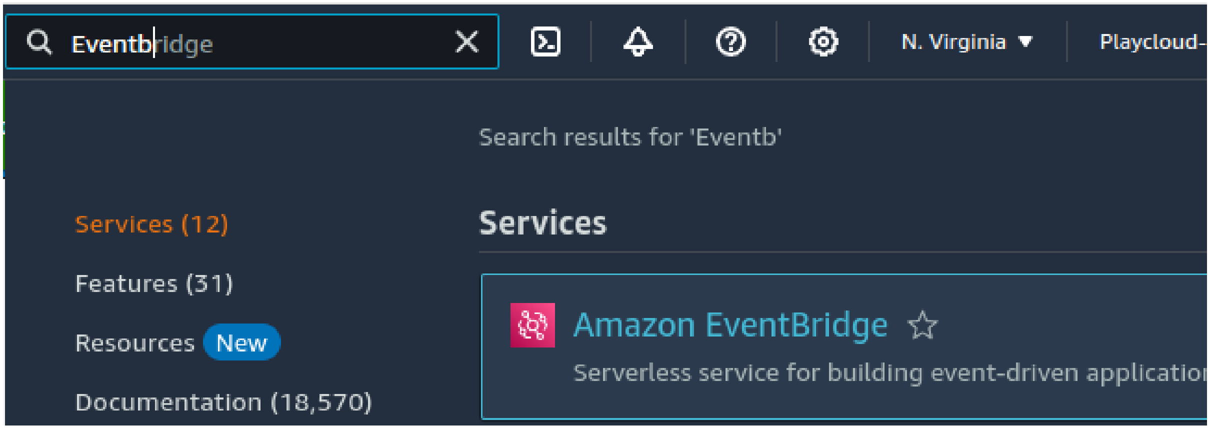
- Click **Deploy**.



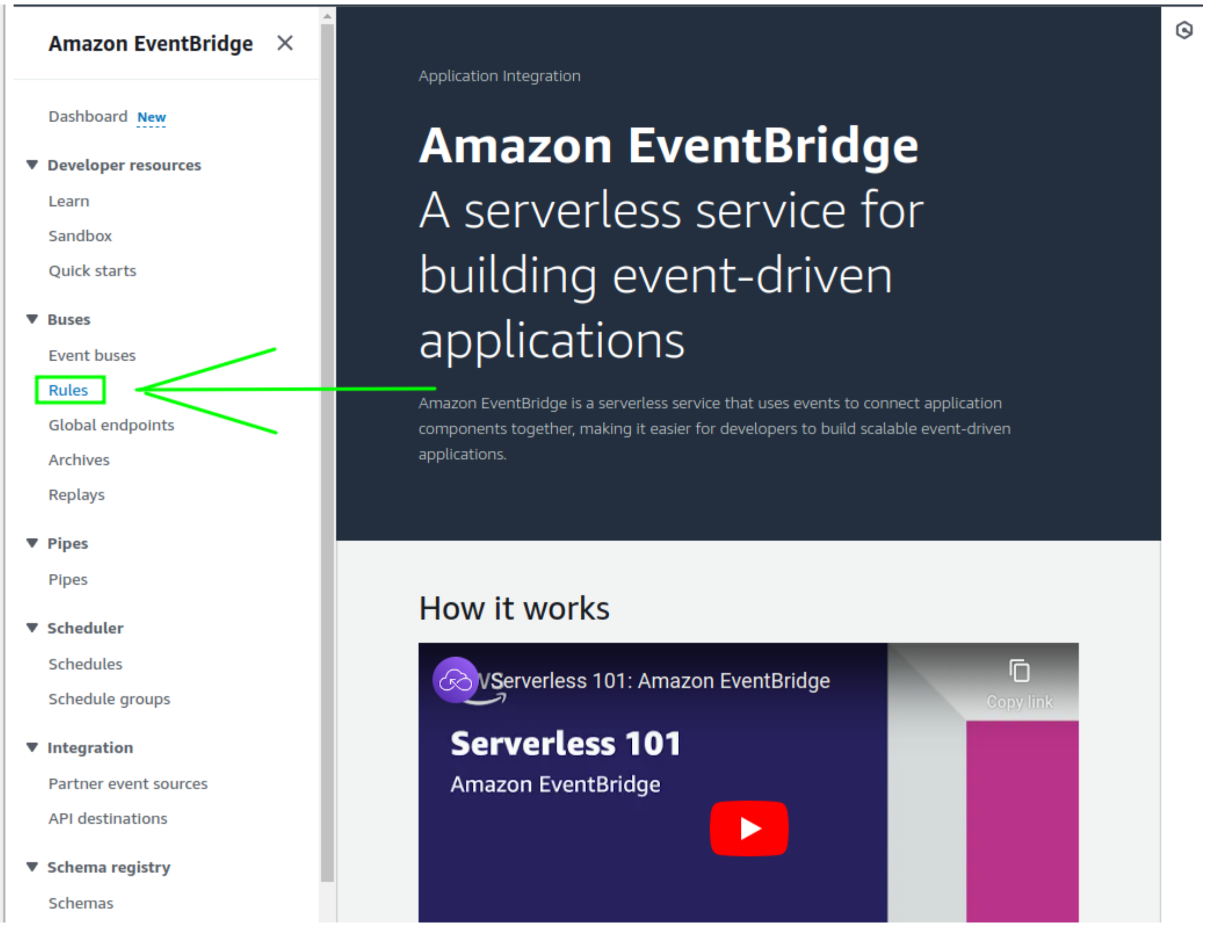
```
import boto3

ec2 = boto3.client('ec2')

def lambda_handler(event, context):
    instance_id = 'i-0b648e434001d145f'  # Replace with your instance ID
    action = event['action']  # 'start' or 'stop'

    if action == 'start':
        ec2.start_instances(InstanceIds=[instance_id])
        return 'Instance started'
    elif action == 'stop':
        ec2.stop_instances(InstanceIds=[instance_id])
        return 'Instance stopped'
    else:
        return 'Invalid action'
```

Create Amazon EventBridge Rules to Trigger Lambda

1. Navigate to the EventBridge **Dashboard**.

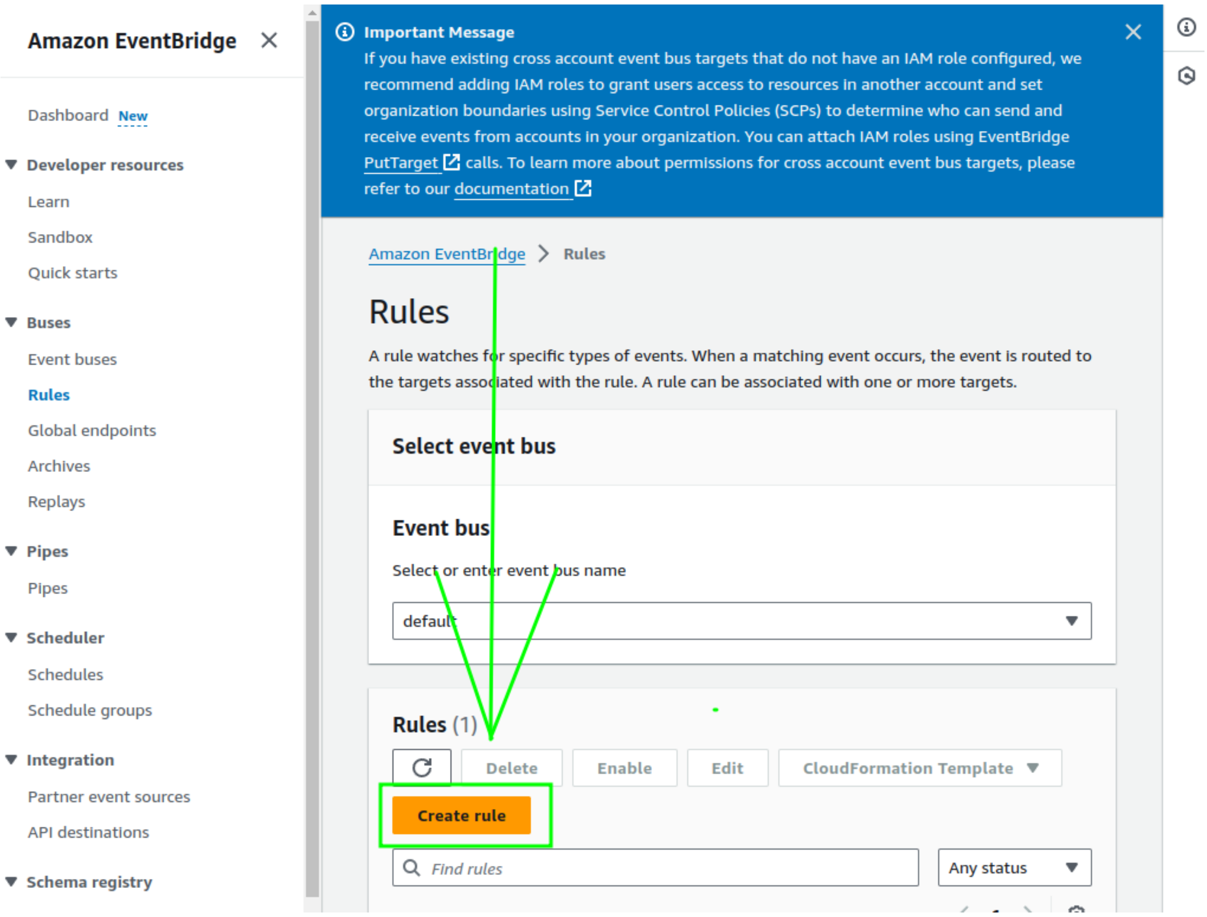


2. Click on **Rules** in the left navigation pane.



Amazon EventBridge is a serverless service that uses events to connect application components together, making it easier for developers to build scalable event-driven applications.

3. Click **Create rule**.



Important Message

If you have existing cross account event bus targets that do not have an IAM role configured, we recommend adding IAM roles to grant users access to resources in another account and set organization boundaries using Service Control Policies (SCPs) to determine who can send and receive events from accounts in your organization. You can attach IAM roles using EventBridge PutTarget calls. To learn more about permissions for cross account event bus targets, please refer to our documentation

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.

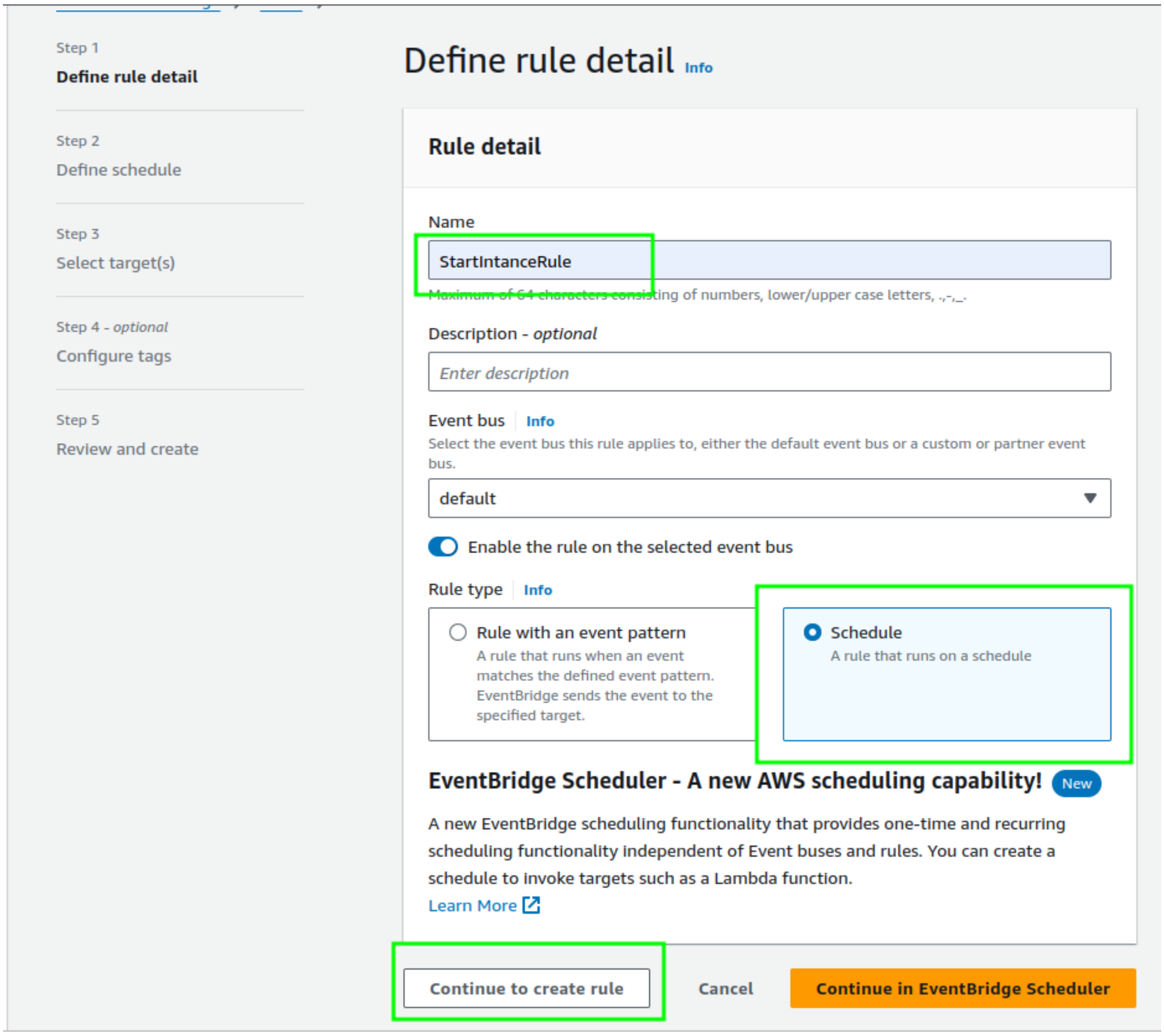
Select event bus

Event bus

Select or enter event bus name: default

4. Configure the following settings to create a **Start Rule:**

- **Name:** StartIntanceRule
- **Rule type:** Select Schedule
- Click on **Continue to create rule**


Step 1
Define rule detail
Step 2
Define schedule
Step 3
Select target(s)
Step 4 - *optional*
Configure tags
Step 5
Review and create

Define rule detail Info

Rule detail

Name: StartIntanceRule

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

Description - *optional*: Enter description

Event bus Info

Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.

default

Enable the rule on the selected event bus

Rule type Info

○ Rule with an event pattern
A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.

● Schedule
A rule that runs on a schedule

EventBridge Scheduler - A new AWS scheduling capability! New

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

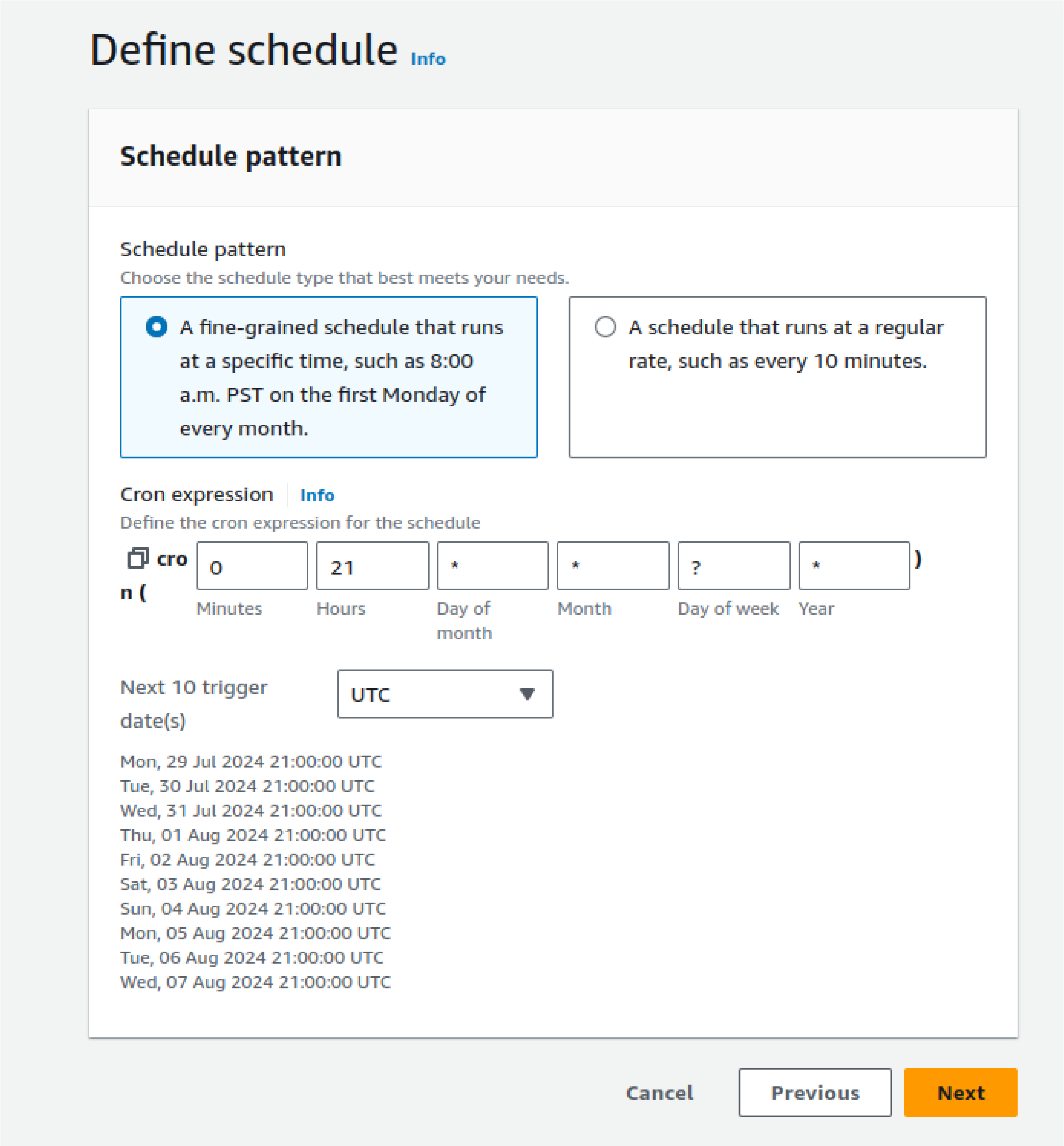
- **Define schedule Step**:

When creating a scheduled rule in EventBridge, you have two options to define how often the rule triggers:

- ***Rate Expression –*** *Use this for a simple, fixed-rate schedule.*
  Example: Trigger the rule every 5 minutes, every hour, or every week.

- ***Cron Expression-*** *Use this for a more detailed, customizable schedule.*
  Example: Trigger the rule at 8 AM on the first Monday of each month.

- Set the cron expression to stop the instance to a preferred time(e.g., 0 21 * * ? * to stop at 9 PM UTC every day).



Define schedule Info

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 (	0	21	*	*	?	*	)

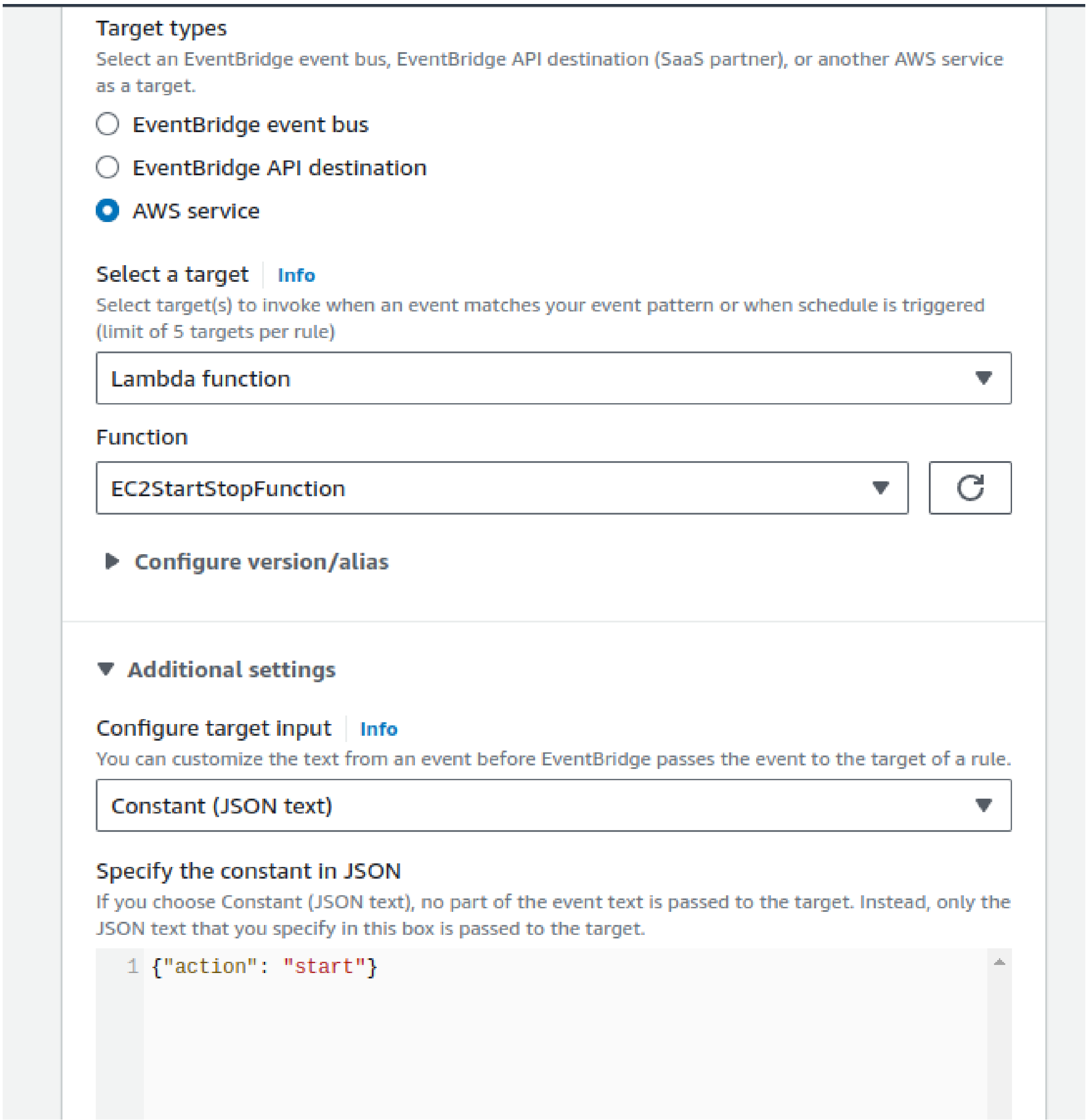
Next 10 trigger date(s) UTC

Mon, 29 Jul 2024 21:00:00 UTC
Tue, 30 Jul 2024 21:00:00 UTC
Wed, 31 Jul 2024 21:00:00 UTC
Thu, 01 Aug 2024 21:00:00 UTC
Fri, 02 Aug 2024 21:00:00 UTC
Sat, 03 Aug 2024 21:00:00 UTC
Sun, 04 Aug 2024 21:00:00 UTC
Mon, 05 Aug 2024 21:00:00 UTC
Tue, 06 Aug 2024 21:00:00 UTC
Wed, 07 Aug 2024 21:00:00 UTC

- Take time to review the Next ten trigger dates, then click **Next**

5. Under **Targets**, click **Add target**.

- **Target**: Choose Lambda function.
- **Function**: Select the Lambda function created in EC2StartStopFunction
- **Additional settings**: Select **Constant (JSON text)** and enter {"action": "start"}.



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

EC2StartStopFunction

▶ Configure version/alias

▼ Additional settings

Configure target input Info

You can customize the text from an event before EventBridge passes the event to the target of a rule.

Constant (JSON text)

Specify the constant in JSON

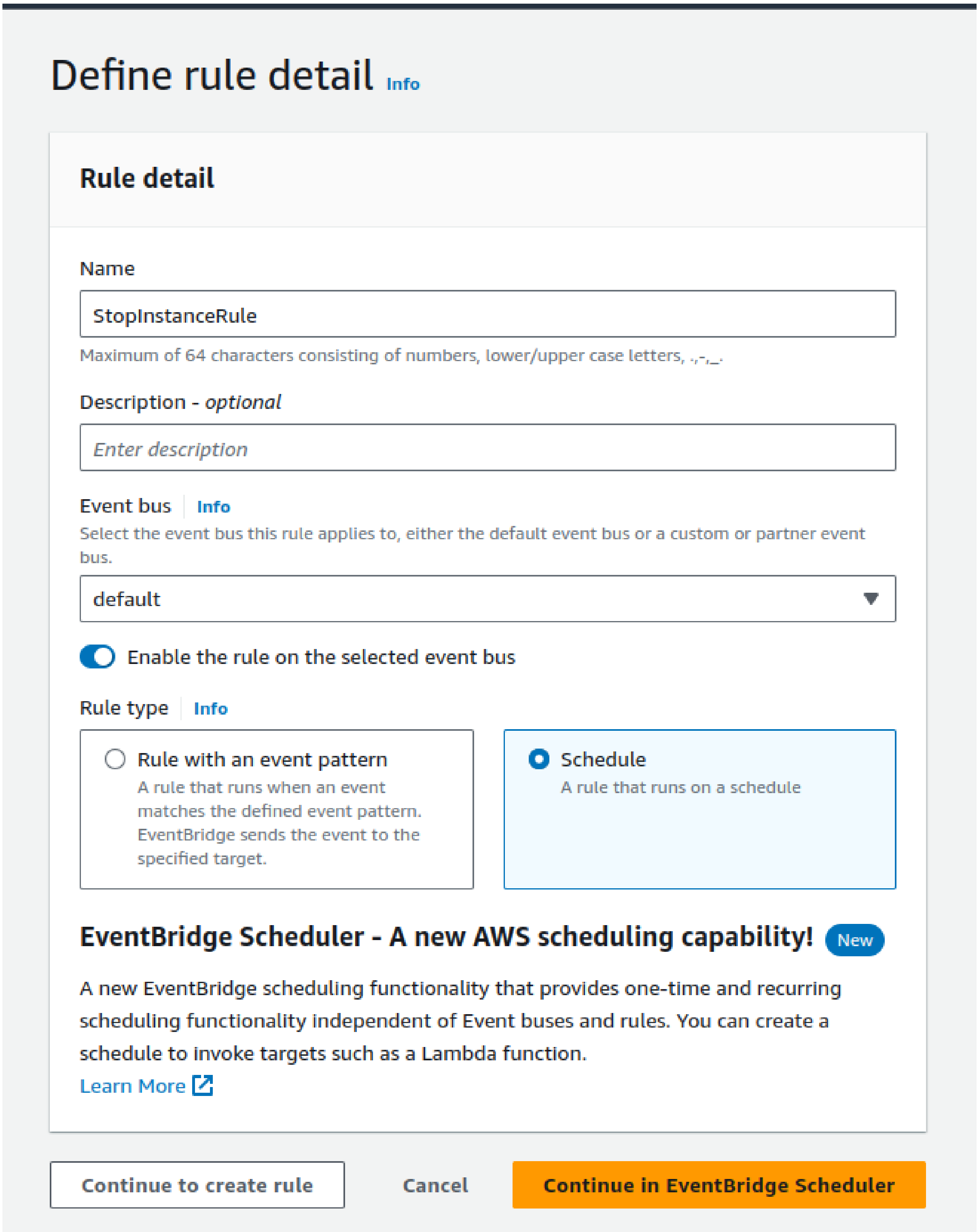
If you choose Constant (JSON text), no part of the event text is passed to the target. Instead, only the JSON text that you specify in this box is passed to the target.

```
{"action": "start"}
```

- Scroll down and Click **Next,** Click **Next u**ntil you get to the **Review and create section**
- Click **Create rule**

6. Repeat steps 4-6 to create a **Stop Rule**:

- **Name:** StartIntanceRule
- **Rule type:** Select Schedule
- Click on **Continue to create rule**



Define rule detail

Rule detail

Name

StopInstanceRule

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

Description - *optional*

Enter description

Event bus

Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.

default

Enable the rule on the selected event bus

Rule type

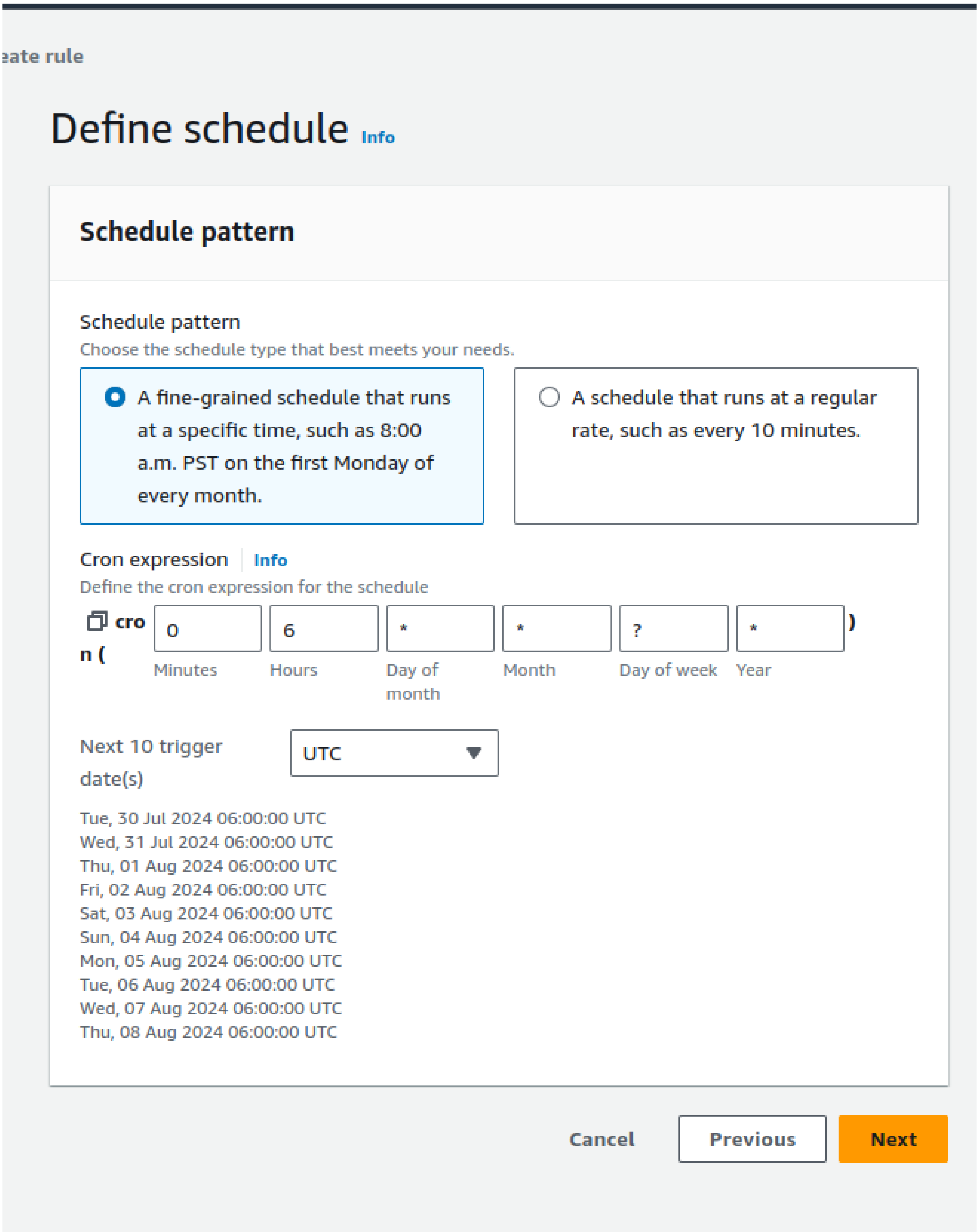
○ Rule with an event pattern
A rule that runs when an event matches the defined event pattern. EventBridge sends the event to the specified target.

● Schedule
A rule that runs on a schedule

EventBridge Scheduler - A new AWS scheduling capability! New

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.

- **Schedule**: Set the cron expression to stop the instance (e.g., 0 6 * * ? * to stop at 6 AM UTC every day).



eate rule

Define schedule Info

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

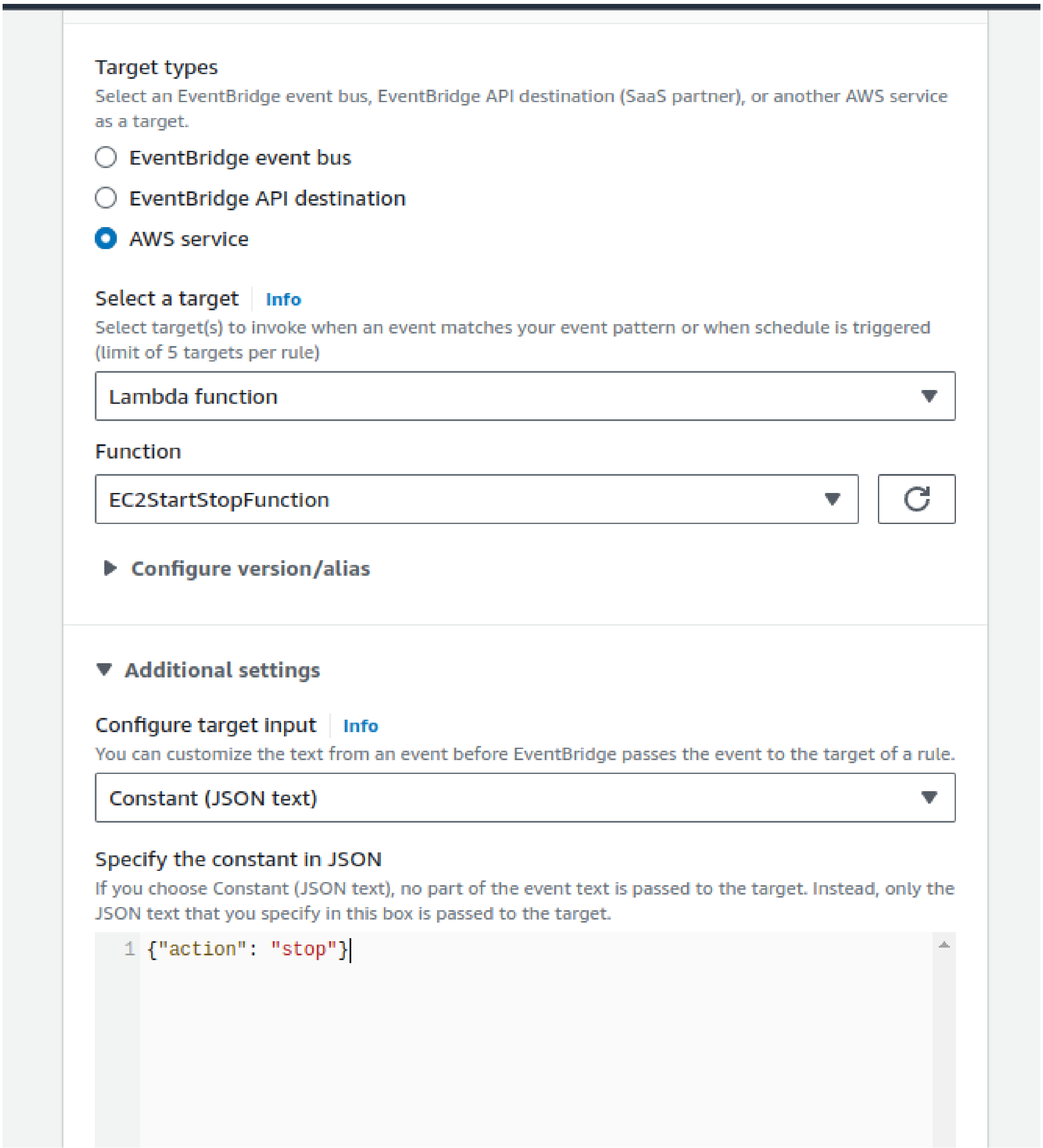
Define the cron expression for the schedule

cron (	0	6	*	*	?	*	)
	Minutes	Hours	Day of month	Month	Day of week	Year	

Next 10 trigger date(s): UTC

```
Tue, 30 Jul 2024 06:00:00 UTC
Wed, 31 Jul 2024 06:00:00 UTC
Thu, 01 Aug 2024 06:00:00 UTC
Fri, 02 Aug 2024 06:00:00 UTC
Sat, 03 Aug 2024 06:00:00 UTC
Sun, 04 Aug 2024 06:00:00 UTC
Mon, 05 Aug 2024 06:00:00 UTC
Tue, 06 Aug 2024 06:00:00 UTC
Wed, 07 Aug 2024 06:00:00 UTC
Thu, 08 Aug 2024 06:00:00 UTC
```

- Click **Next**
- Same **Target**: Choose Lambda function.
- Same **Function**: Select the Lambda function created in EC2StartStopFunction
- **Additional settings**: Select **Constant (JSON text)** and enter {"action": "stop"}.



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

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

EC2StartStopFunction

▼ Additional settings

Configure target input

You can customize the text from an event before EventBridge passes the event to the target of a rule.

Constant (JSON text)

Specify the constant in JSON

If you choose Constant (JSON text), no part of the event text is passed to the target. Instead, only the JSON text that you specify in this box is passed to the target.

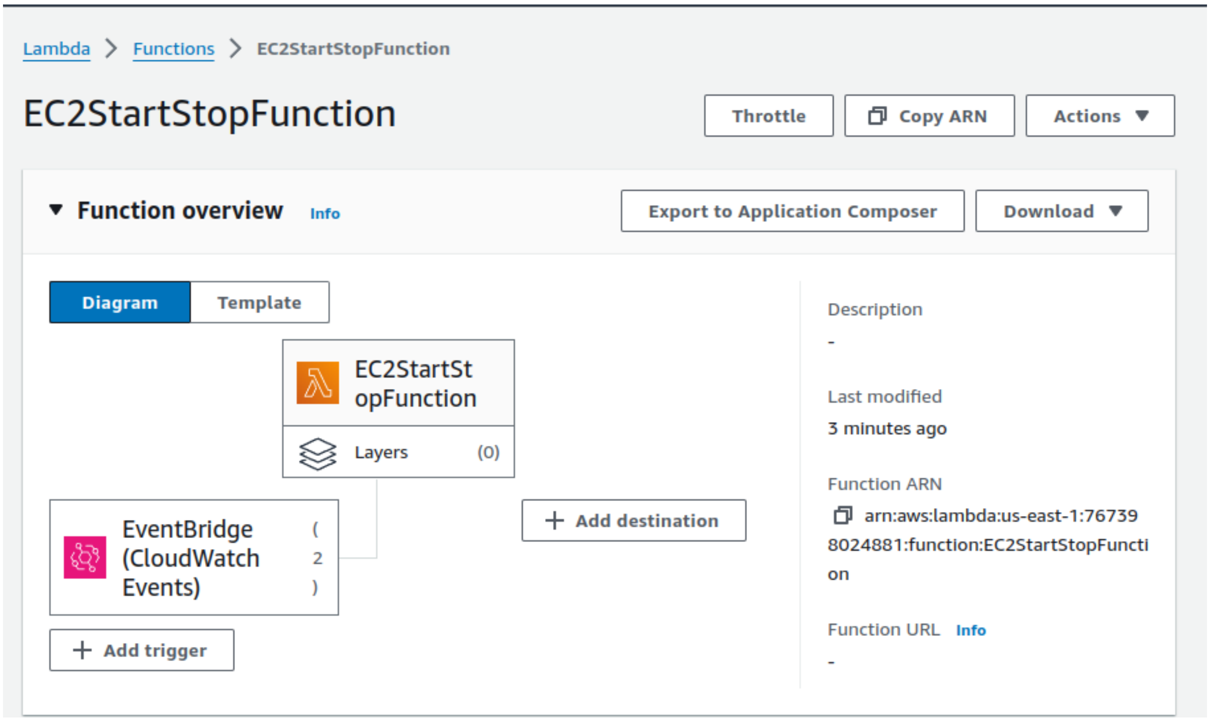
```
{"action": "stop"}
```

- Click **Next** until Step 5 to review. Then, Click **Create rule**

Test the Setup

To simulate the scheduled rule trigger of the EventBridge, we will use the Test Event Feature of our Lambda function.

1. Navigate to the **Lambda Dashboard** and verify the triggers of your Lambda.



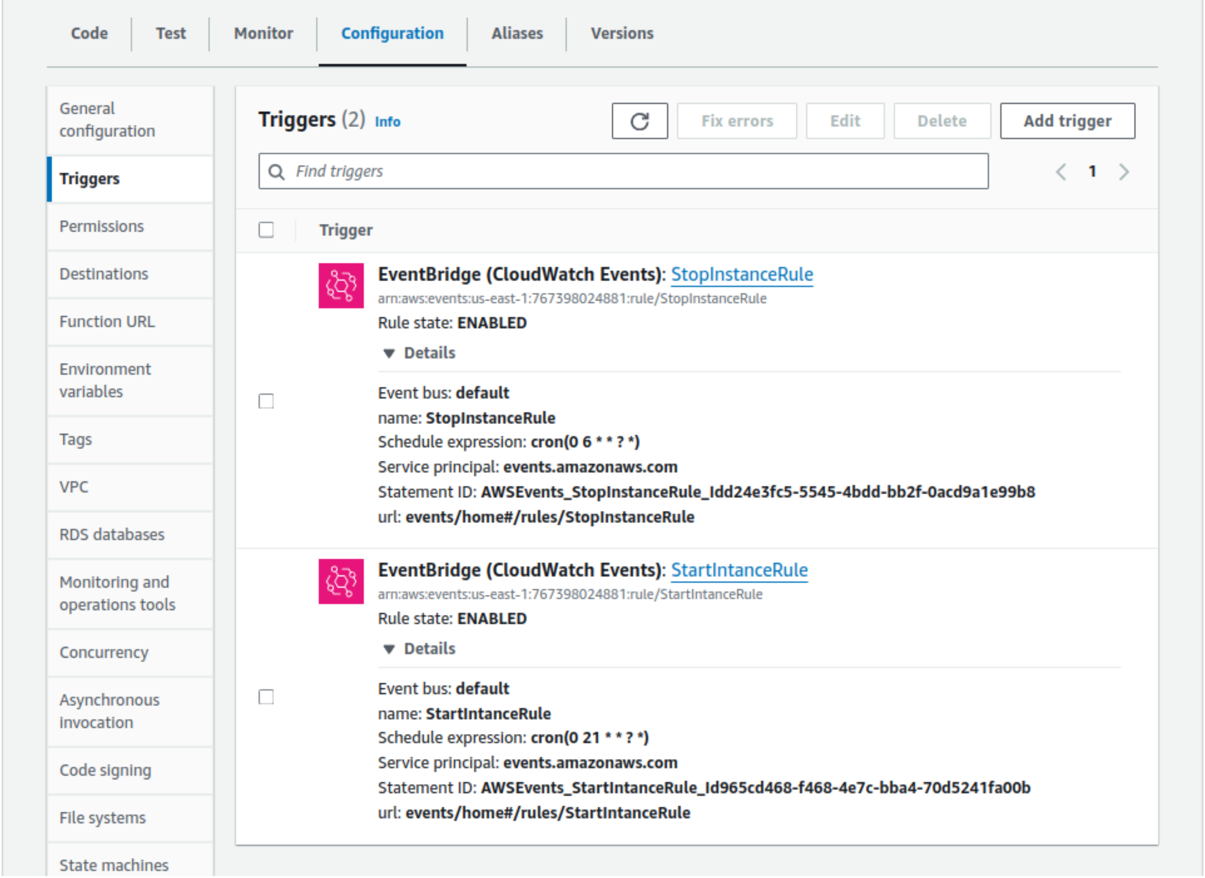
EC2StartStopFunction

Description
-

Last modified
3 minutes ago

Function ARN
arn:aws:lambda:us-east-1:767398024881:function:EC2StartStopFunction

Function URL Info
-



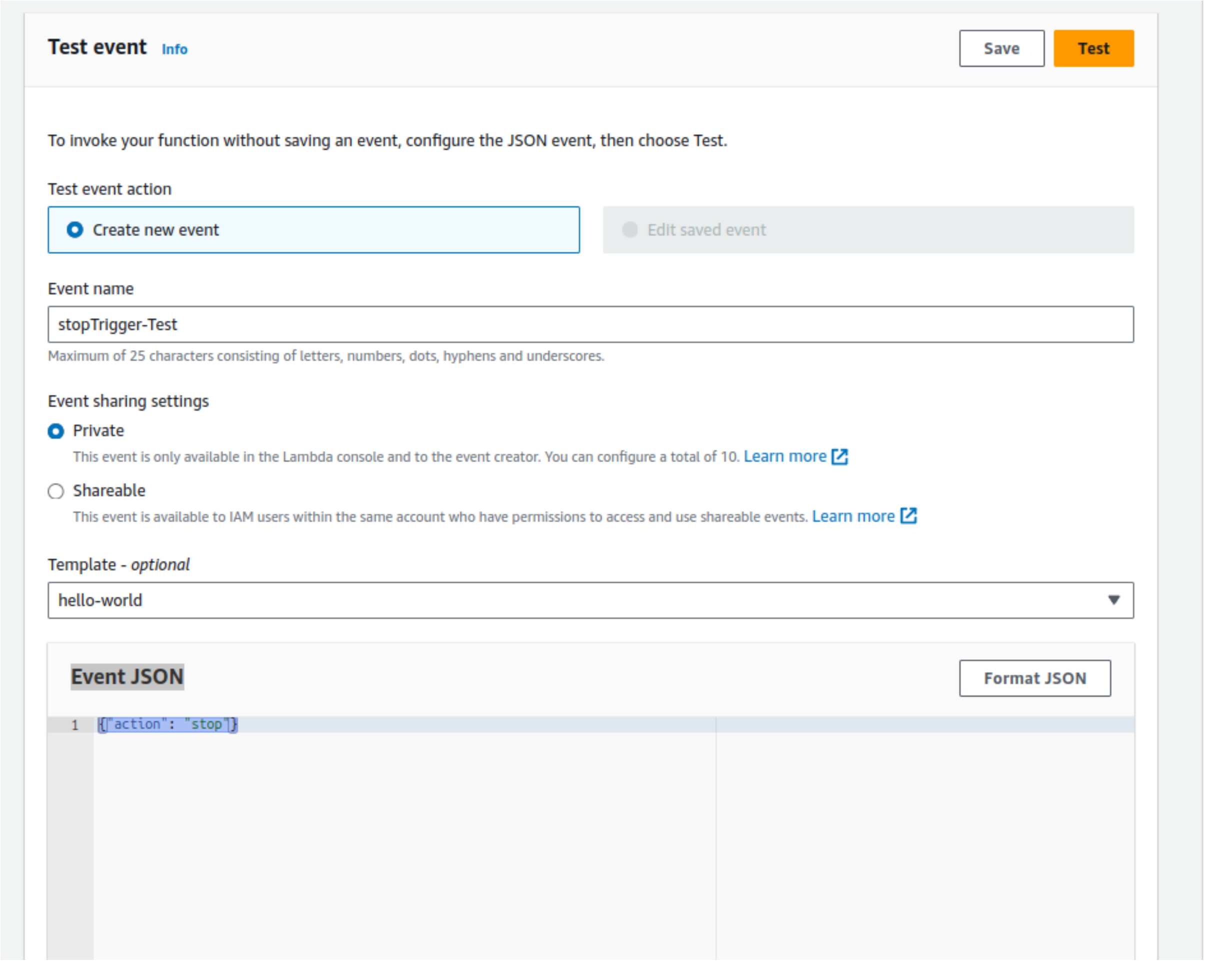
Triggers (2) Info

□ EventBridge (CloudWatch Events): StopInstanceRule
arn:aws:events:us-east-1:767398024881:rule/StopInstanceRule
Rule state: **ENABLED**
▼ Details
Event bus: **default**
name: **StopInstanceRule**
Schedule expression: **cron(0 6 * * ? *)**
Service principal: **events.amazonaws.com**
Statement ID: **AWSEvents_StopInstanceRule_Idd24e3fc5-5545-4bdd-bb2f-0acd9a1e99b8**
url: **events/home#/rules/StopInstanceRule**

□ EventBridge (CloudWatch Events): StartIntanceRule
arn:aws:events:us-east-1:767398024881:rule/StartIntanceRule
Rule state: **ENABLED**
▼ Details
Event bus: **default**
name: **StartIntanceRule**
Schedule expression: **cron(0 21 * * ? *)**
Service principal: **events.amazonaws.com**
Statement ID: **AWSEvents_StartIntanceRule_Id965cd468-f468-4e7c-bba4-70d5241fa00b**
url: **events/home#/rules/StartIntanceRule**

2. Trigger the Lambda by Adding this Test Event:

- **Event name:** stopTrigger-Test
- **Event JSON**: {"action": "stop"}



Test event Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

● Create new event ○ Edit saved event

Event name

stopTrigger-Test

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

Event sharing settings

● Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. 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

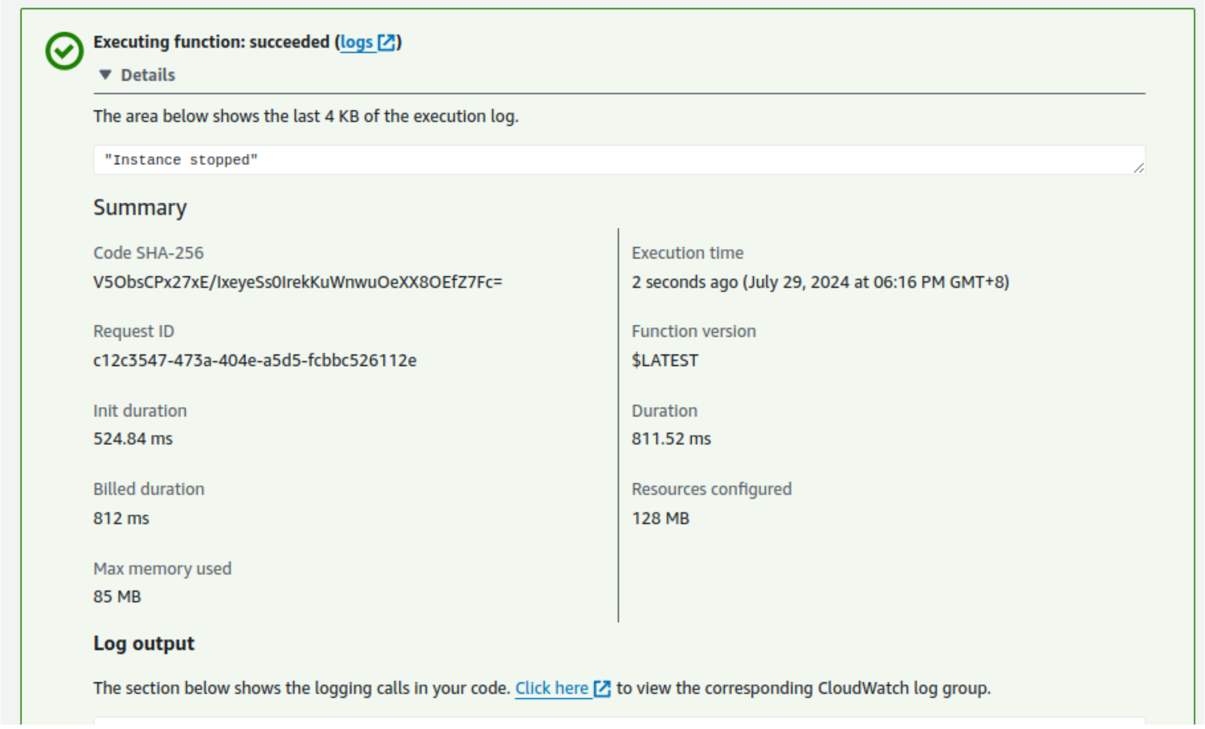
Template - *optional*

hello-world

Event JSON

```
{"action": "stop"}
```

3. Save it and click on **Test**



Executing function: succeeded (logs)

▼ Details

The area below shows the last 4 KB of the execution log.

```
"Instance stopped"
```

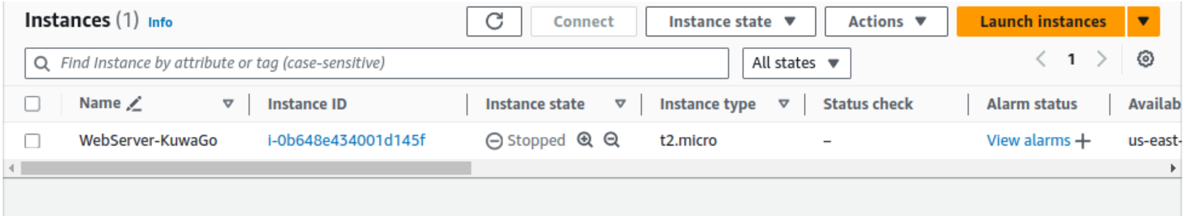
Summary

Code SHA-256 V5ObsCPx27xE/IxeyeSs0lrekKuWnwuOeXX8OEfZ7Fc=	Execution time 2 seconds ago (July 29, 2024 at 06:16 PM GMT+8)
Request ID c12c3547-473a-404e-a5d5-fcbbc526112e	Function version $LATEST
Init duration 524.84 ms	Duration 811.52 ms
Billed duration 812 ms	Resources configured 128 MB
Max memory used 85 MB	

Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

a. Navigate to your instance to check if it really stopped. Refresh it if necessary

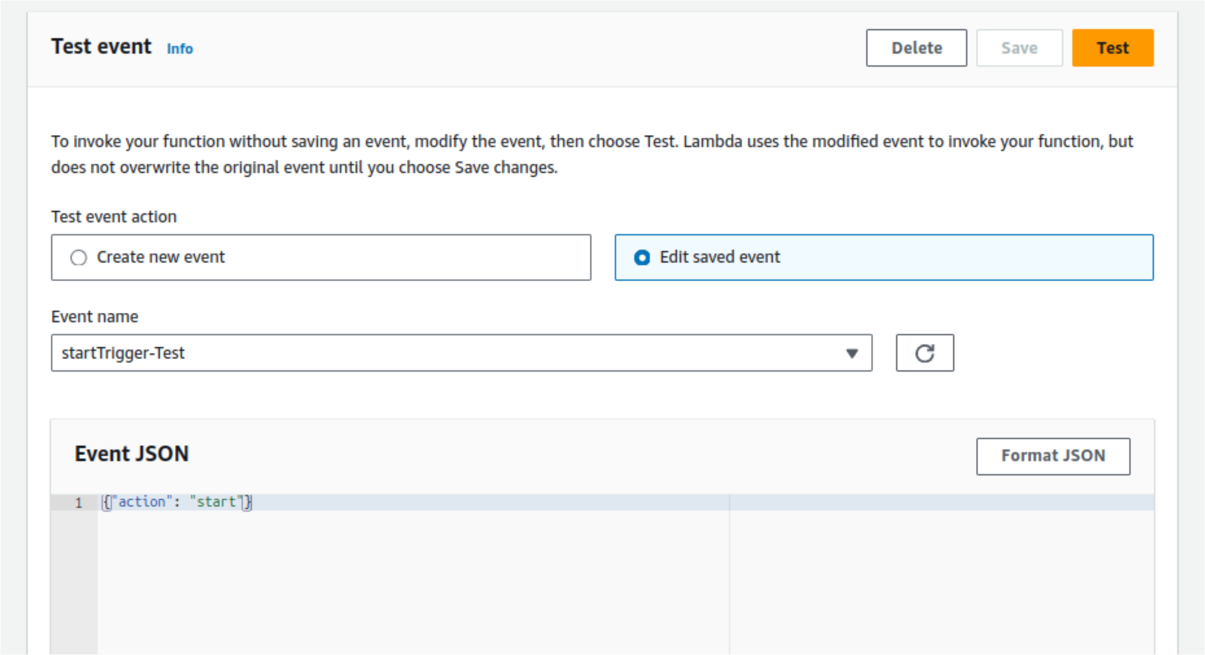


Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availab
□	WebServer-KuwaGo	i-0b648e434001d145f	Stopped	t2.micro	–	View alarms +	us-east-

4. Repeat step 2 and step 3 with the following details this time:

- **Event name:** startTrigger-Test
- **Event JSON**: {"action": "start"}



Test event Info

To invoke your function without saving an event, modify the event, then choose Test. Lambda uses the modified event to invoke your function, but does not overwrite the original event until you choose Save changes.

Test event action

○ Create new event ● Edit saved event

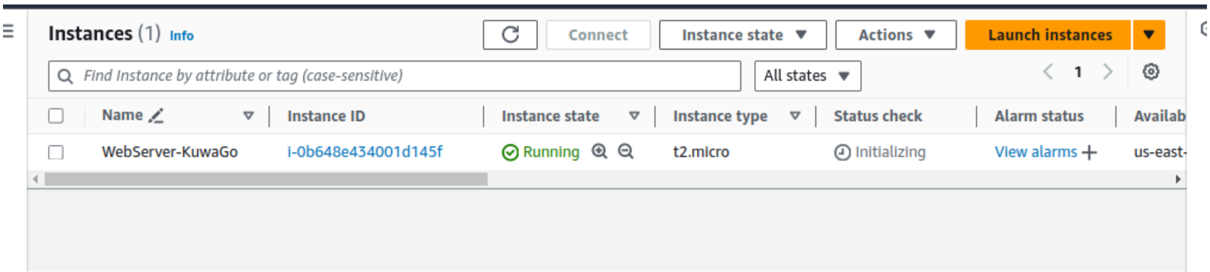
Event name

startTrigger-Test

Event JSON

```
{"action": "start"}
```

- Save it Click **Test**
- Navigate to your instance again to check if the instance started running. Refresh it if necessary.



Instances (1) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availab
□	WebServer-KuwaGo	i-0b648e434001d145f	Running	t2.micro	Initializing	View alarms +	us-east-

That's it! Conratulations! You have just gained practical experience in automating the start and stop of EC2 instances using Amazon EventBridge and AWS Lambda. This automation is crucial for efficient resource management, cost optimization, and operational efficiency in cloud computing environments. You have learned how to create and manage an EC2 instance, set up Lambda functions, and schedule events with EventBridge to control the lifecycle of your instances. This knowledge equips you to implement similar solutions in real-world scenarios, ensuring that resources are available when needed and minimizing unnecessary costs.