

# Creating a Lambda Function

Log in into AWS services and visit Lambda's page and click on "Create a Function"

As our code has lots of dependencies, our best option is to use an Container Image for replicating the environment, you should be able to select the image pushed on some steps ago.

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.

Browse serverless app repository

Deploy a sample Lambda application from the AWS Serverless Application Repository.

**Basic information**

Function name [Info](#)

Enter a name that describes the purpose of your function.

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

Container image URI [Info](#)

The location of the container image to use for your function.

Requires a valid Amazon ECR image URI.

[Browse images](#)

► Container image overrides

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

► Advanced settings

On "Edit basic settings" it's important to raise Memory and Timeout values.

**AWS Lambda**

Dashboard

Applications

Functions

webmotors-webscraping

**Edit basic settings**

▼ Additional resources

Code signing configurations

Layers

Replicas

▼ Related AWS resources

Step Functions state machines

Lambda > Functions > webmotors-webscraping > Edit basic settings

## Edit basic settings

**Basic settings** [Info](#)

Description - optional

**Memory** [Info](#)

Your function is allocated CPU proportional to the memory configured.

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](#)

MB

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

**Timeout**

min  sec

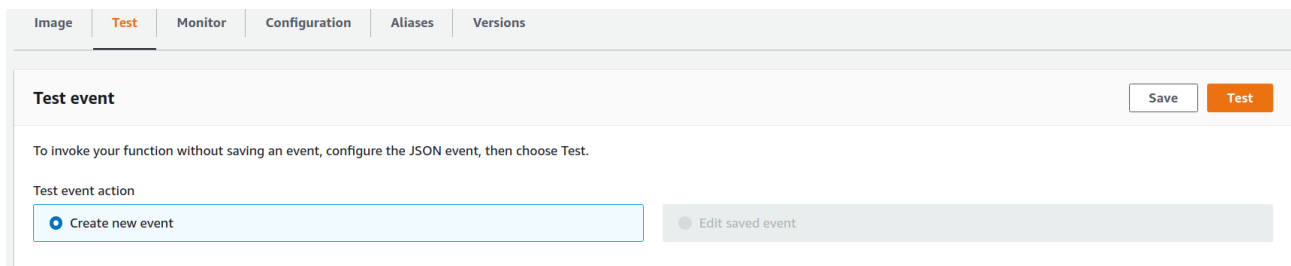
**Execution role**

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

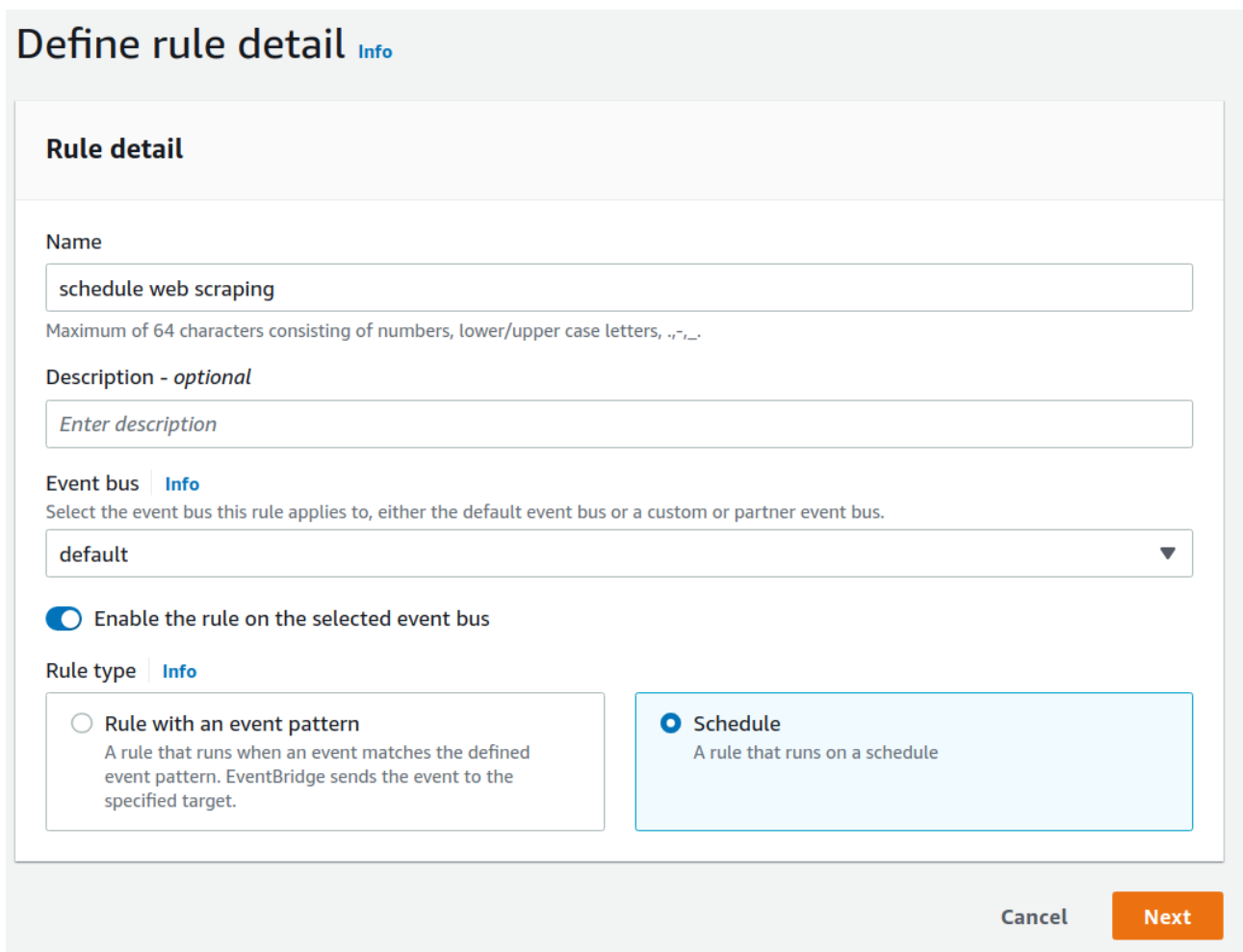
At this point, it's possible to run a test for your script and see if everything goes as expected.



The screenshot shows the 'Test event' tab in the AWS Lambda console. At the top, there are tabs for 'Image', 'Test' (which is active), 'Monitor', 'Configuration', 'Aliases', and 'Versions'. Below the tabs, there's a 'Test event' section with a 'Save' button and a 'Test' button. A message states: 'To invoke your function without saving an event, configure the JSON event, then choose Test.' Under 'Test event action', there are two options: 'Create new event' (selected with a blue radio button) and 'Edit saved event' (disabled with a grey radio button).

## Scheduling a Lambda Function

Go to AWS [EventBridge](#) to create a new schedule rule



The screenshot shows the 'Define rule detail' form in the AWS EventBridge console. The form has a header 'Define rule detail' with an 'Info' link. Below is a 'Rule detail' section. It includes a 'Name' field with the value 'schedule web scraping' and a note: 'Maximum of 64 characters consisting of numbers, lower/upper case letters, -, \_, .'. There is a 'Description - optional' field with the placeholder 'Enter description'. The 'Event bus' section has an 'Info' link and a dropdown menu set to 'default', with a note: 'Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.' Below this is a toggle switch for 'Enable the rule on the selected event bus', which is turned on. The 'Rule type' section has an 'Info' link and two options: 'Rule with an event pattern' (unselected) and 'Schedule' (selected with a blue radio button). The 'Schedule' option has a description: 'A rule that runs on a schedule'. At the bottom right, there are 'Cancel' and 'Next' buttons.

## 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 (       )

Minutes Hours Day of month Month Day of week Year

#### Next 10 trigger date(s)

Local time zone ▼

Jul 30, 2022, 08:00 PM GMT-3  
Jul 31, 2022, 08:00 PM GMT-3  
Aug 1, 2022, 08:00 PM GMT-3  
Aug 2, 2022, 08:00 PM GMT-3  
Aug 3, 2022, 08:00 PM GMT-3  
Aug 4, 2022, 08:00 PM GMT-3  
Aug 5, 2022, 08:00 PM GMT-3  
Aug 6, 2022, 08:00 PM GMT-3

## And associate this rule with your Lambda Function

### 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

webmotors-webscraping ▼ 

► **Configure version/alias**

► **Additional settings**

From now on your function should be scheduled

[Dashboard](#)[Applications](#)[Functions](#)[webmotors-webscraping](#)▼ **Additional resources**[Code signing configurations](#)[Layers](#)[Replicas](#)▼ **Related AWS resources**[Step Functions state machines](#)

## webmotors-webscraping

▼ **Function overview** [Info](#)webmotors-webscr  
apingEventBridge (CloudWatch E  
vents)

+ Add trigger

+ Add destination