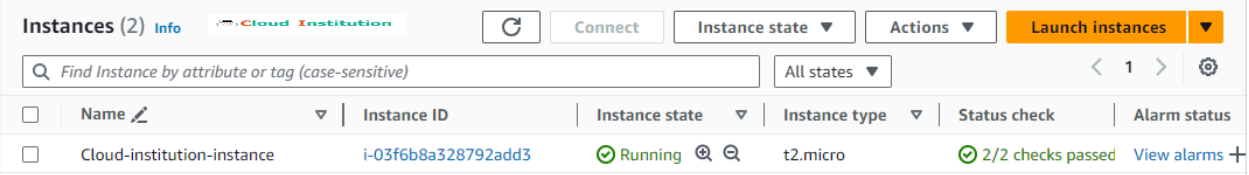
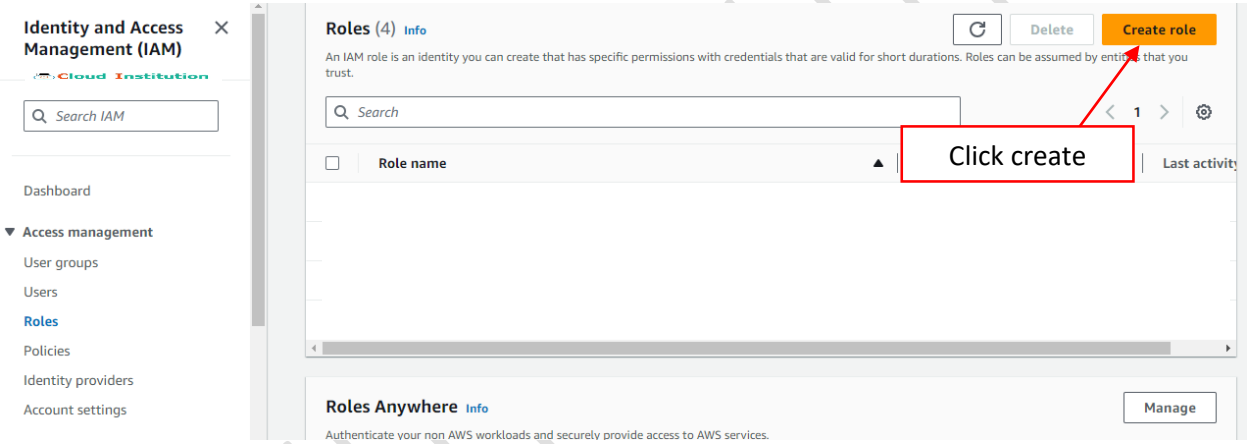
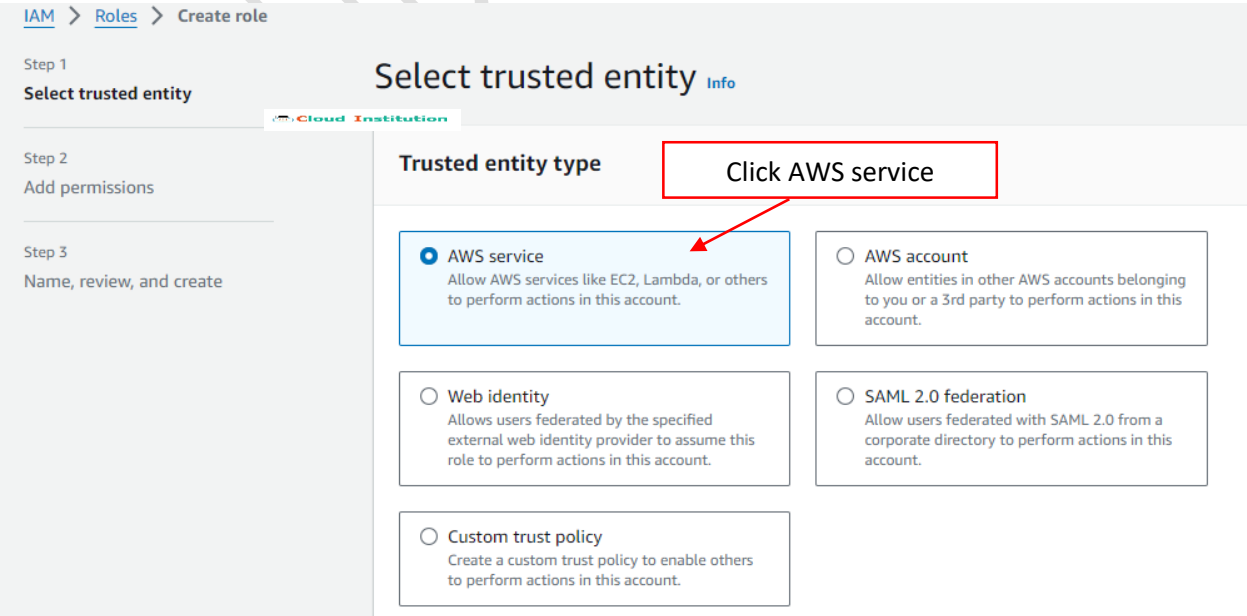
STOP EC2 INSTANCE USING LAMBDA CLOUDWATCH

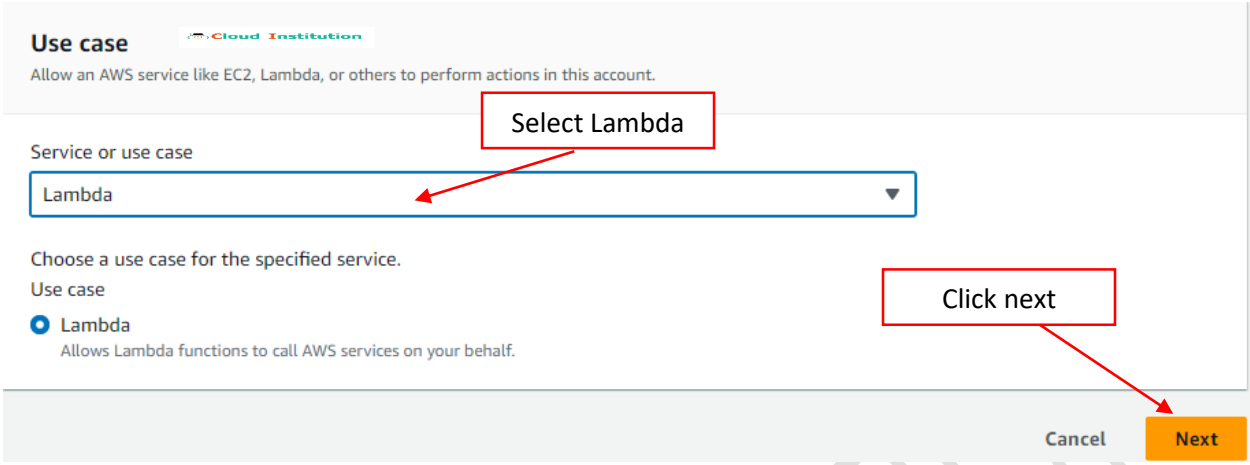
Step 1 : Create a EC2 instance

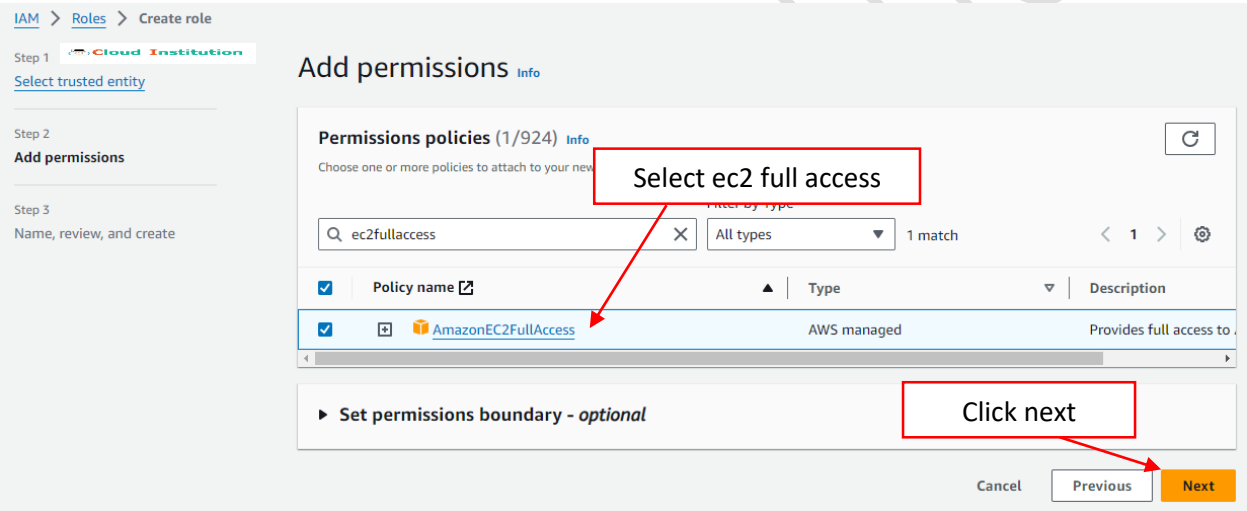


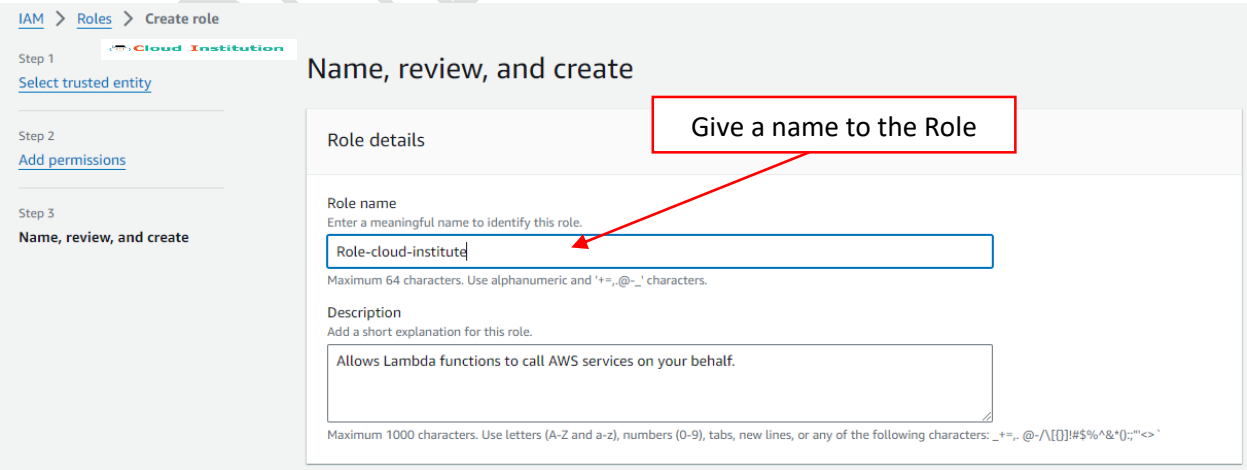
Step 2 : Create a IAM Role

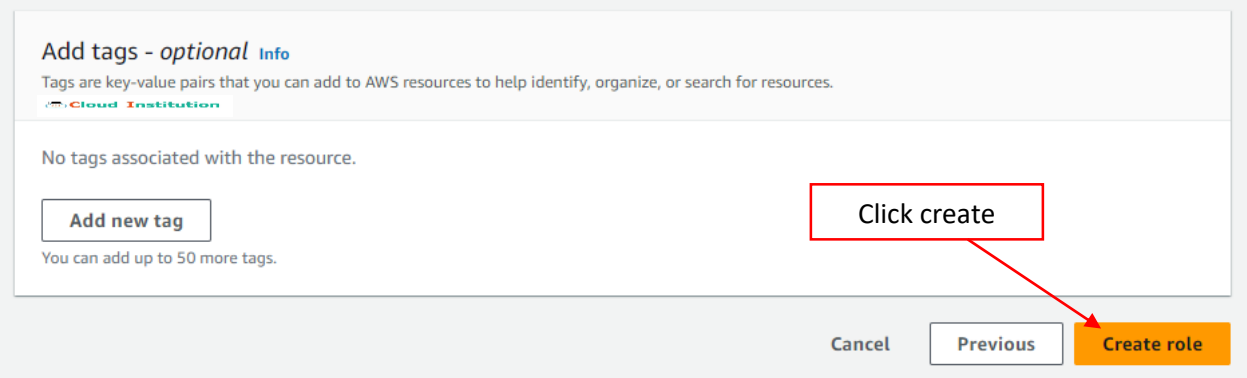




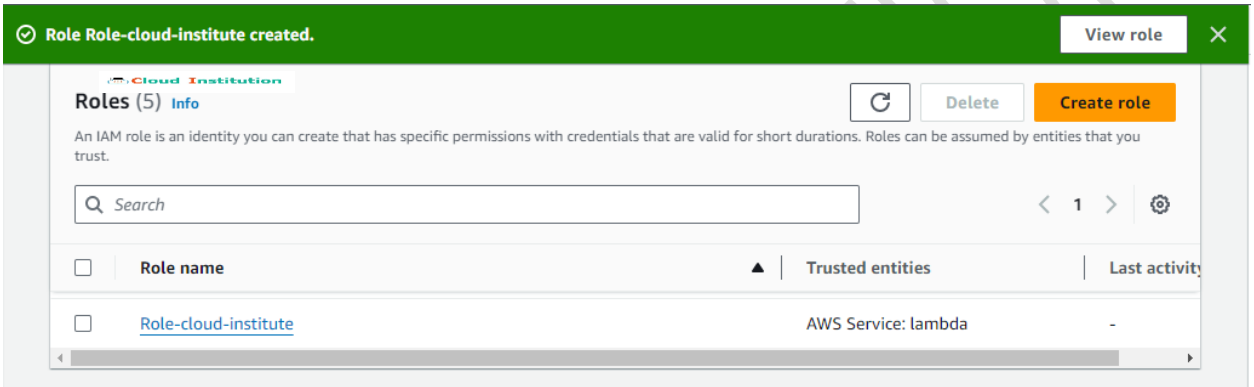

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.
Select Lambda
Service or use case
Lambda
Choose a use case for the specified service.
Use case
Lambda
Allows Lambda functions to call AWS services on your behalf.
Click next
Cancel
Next


IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Add permissions Info
Permissions policies (1/924) Info
Choose one or more policies to attach to your new
Select ec2 full access
ec2fullaccess
All types
1 match
Policy name
Type
Description
AmazonEC2FullAccess
AWS managed
Provides full access to
Set permissions boundary - optional
Click next
Cancel
Previous
Next

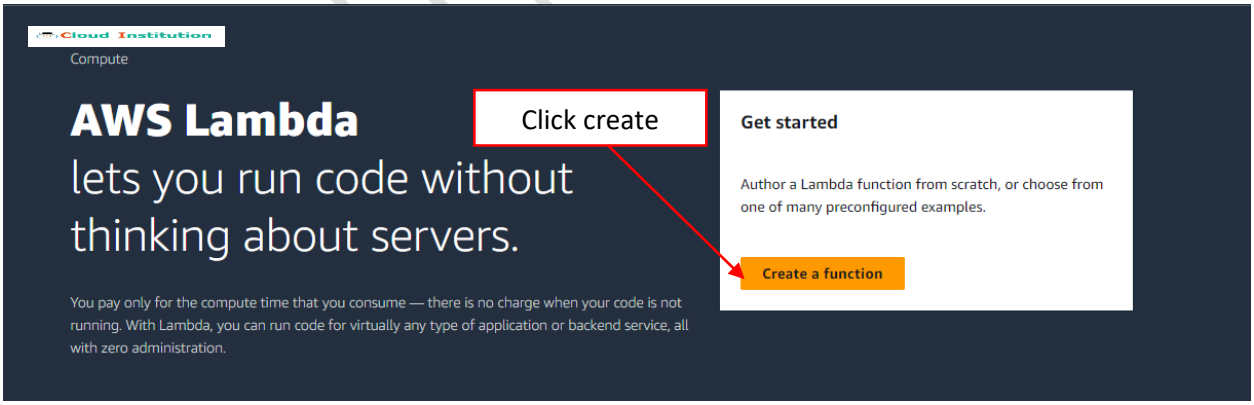

IAM > Roles > Create role
Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create
Name, review, and create
Role details
Give a name to the Role
Role name
Enter a meaningful name to identify this role.
Role-cloud-institute
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.
Description
Add a short explanation for this role.
Allows Lambda functions to call AWS services on your behalf.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^&*();"'<>`

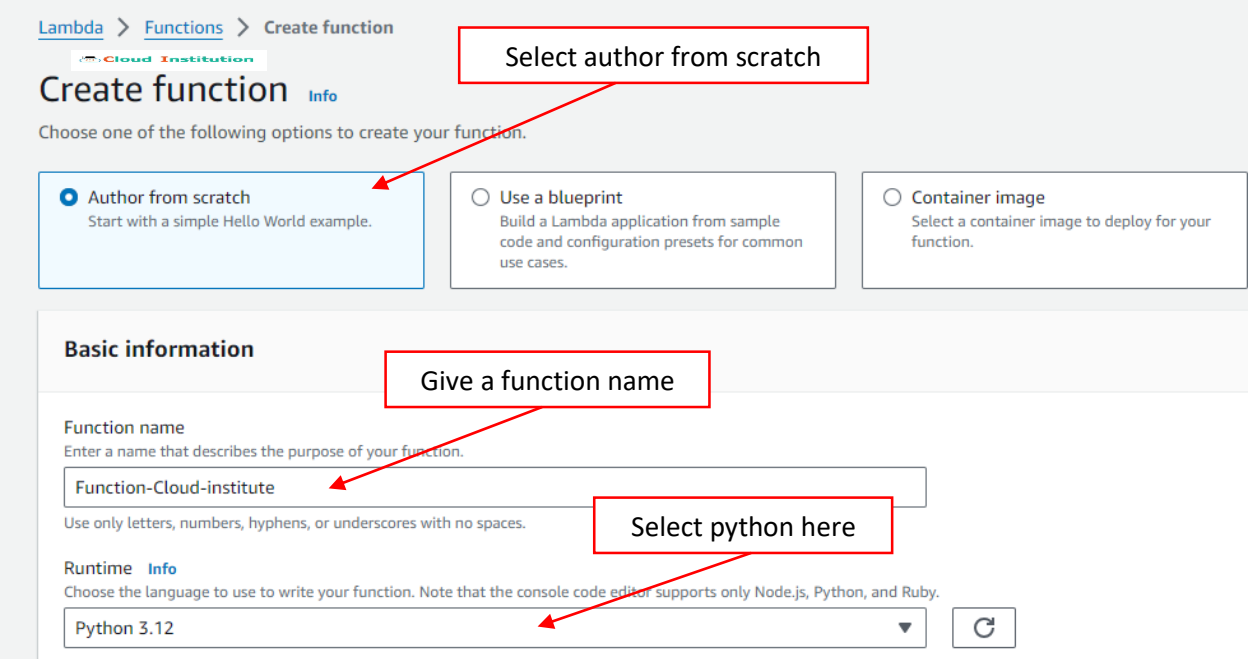


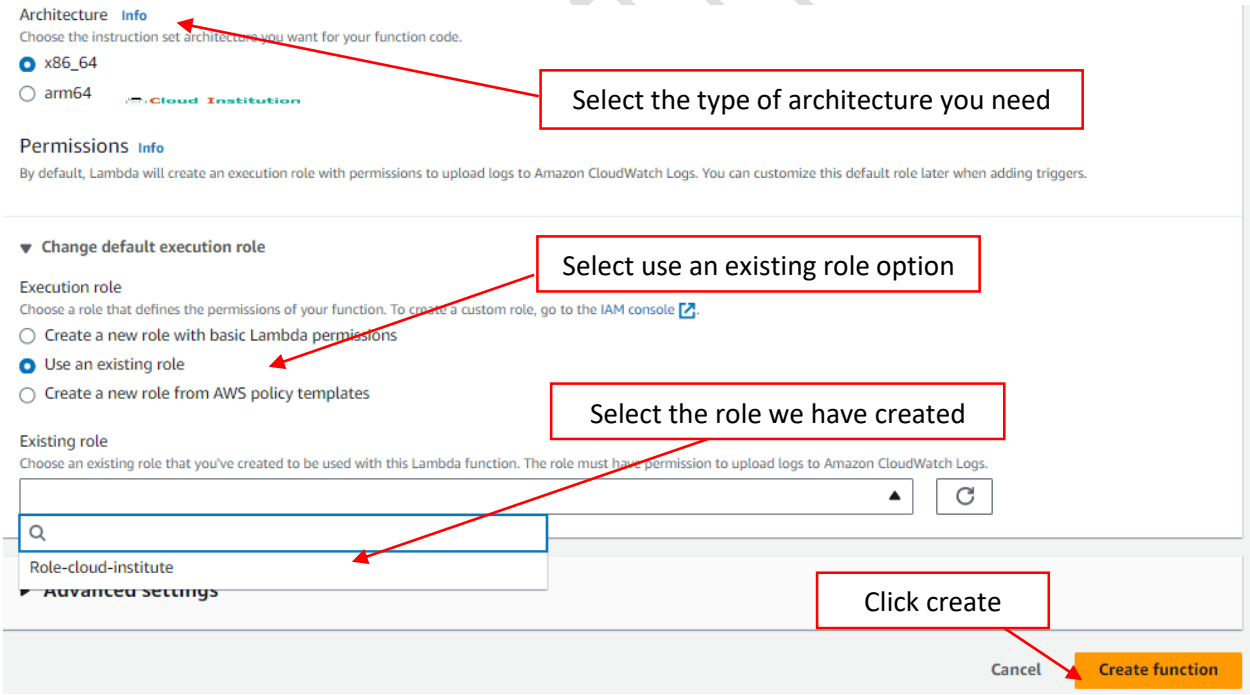
Role Created



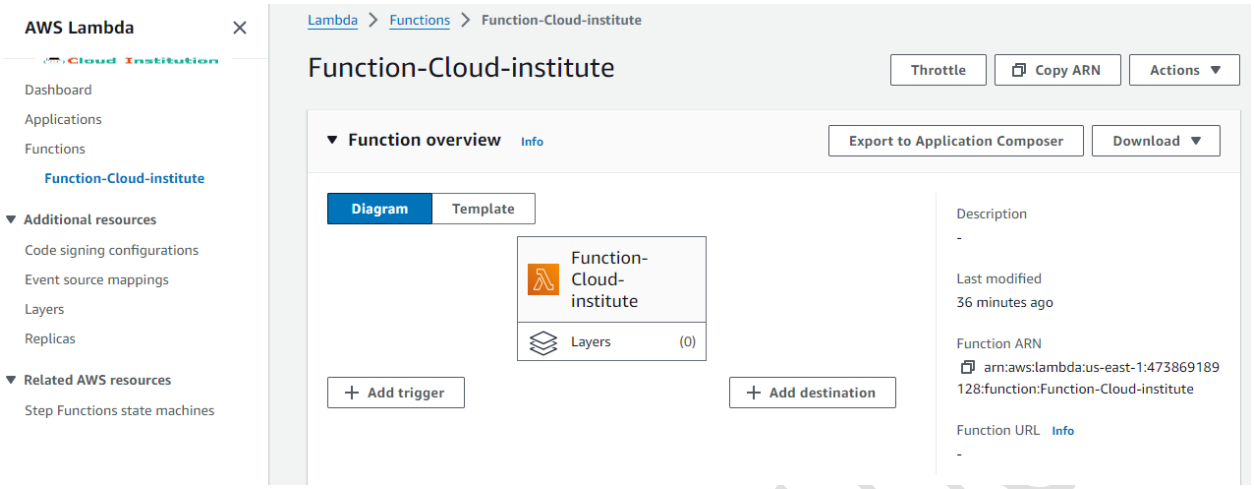
Step 3 : Create a Lambda Function




Lambda > Functions > Create function
Cloud Institution
Create function Info
Choose one of the following options to create your function.
Select author from scratch
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
Give a function name
Function name
Enter a name that describes the purpose of your function.
Function-Cloud-institute
Use only letters, numbers, hyphens, or underscores with no spaces.
Select python here
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

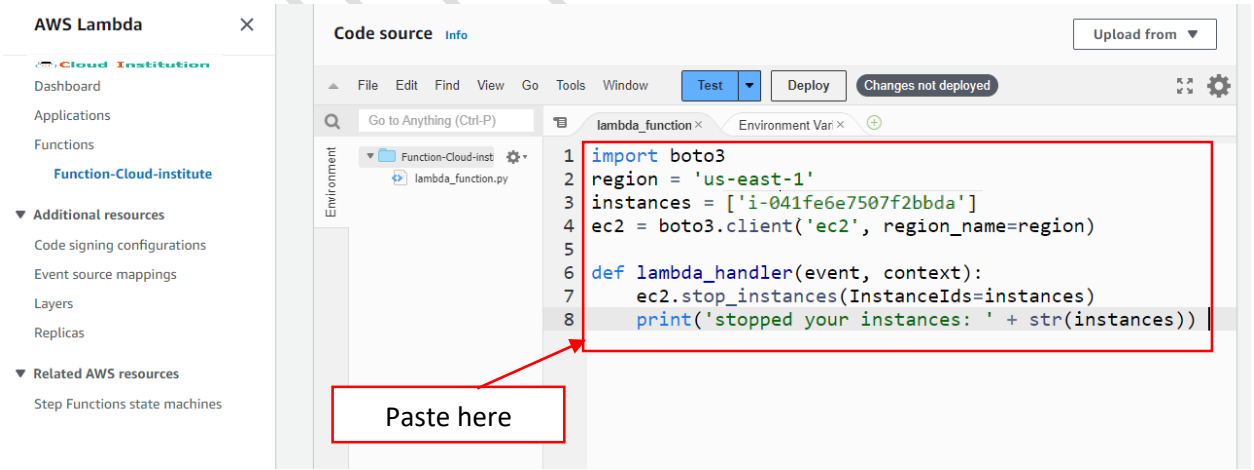

Architecture Info
Choose the instruction set architecture you want for your function code.
x86_64
arm64
Select the type of architecture you need
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Select use an existing role option
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
Select the role we have created
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.
Role-cloud-institute
Click create
Cancel
Create function

Function created

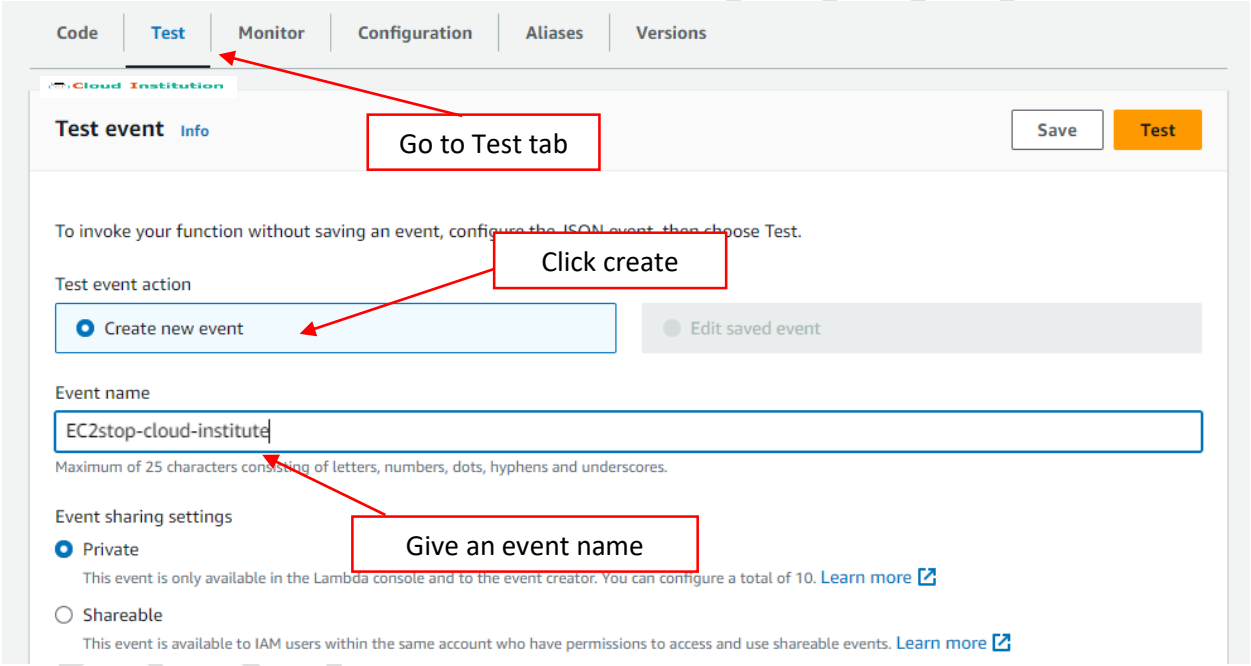


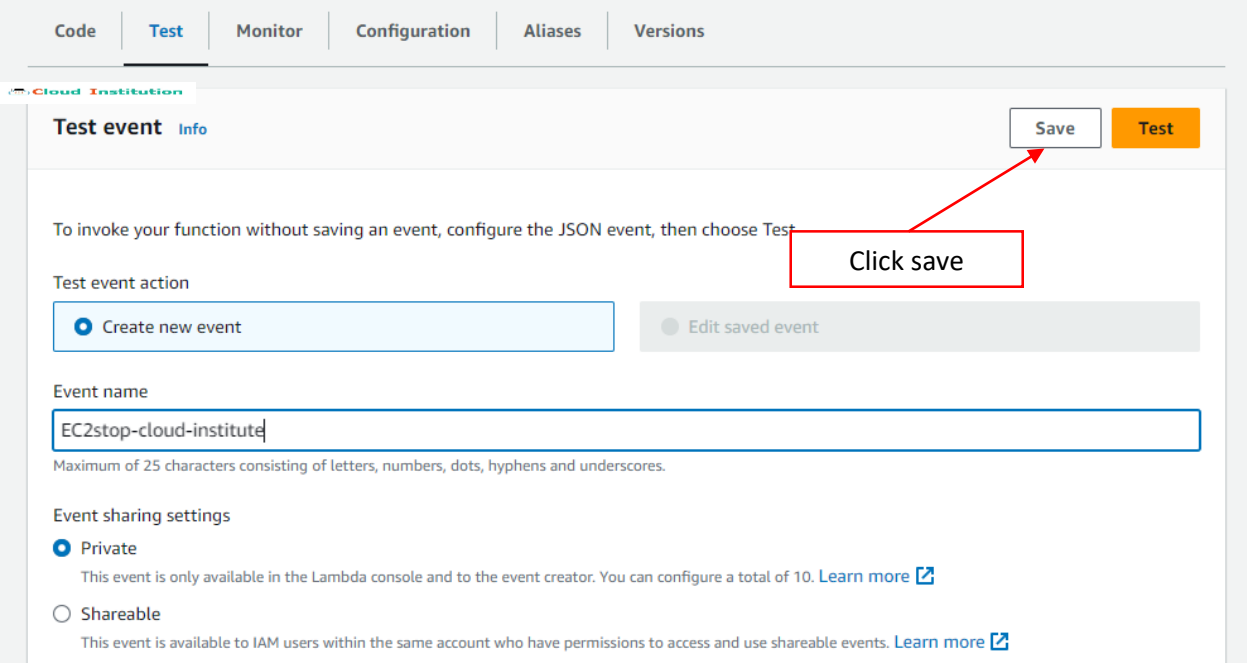
Step 4 : Copy the code

```
import boto3
region = 'us-west-1'
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))
```

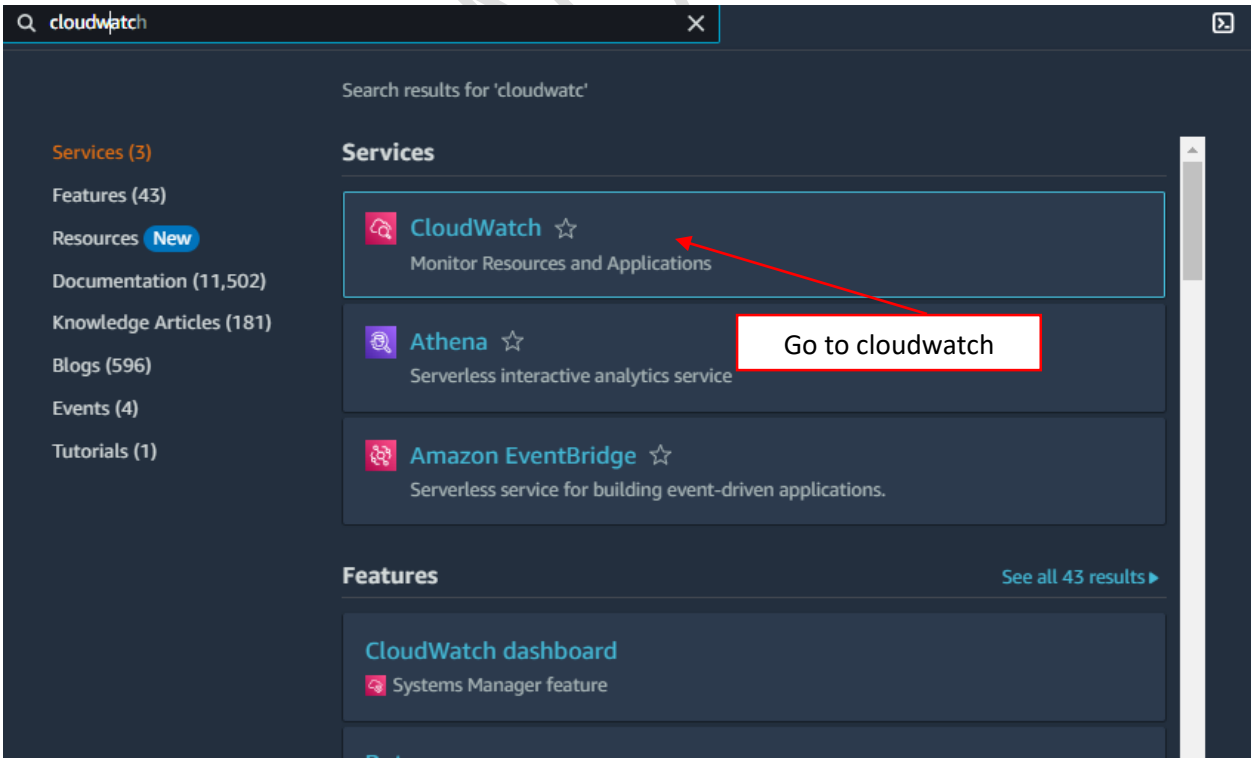


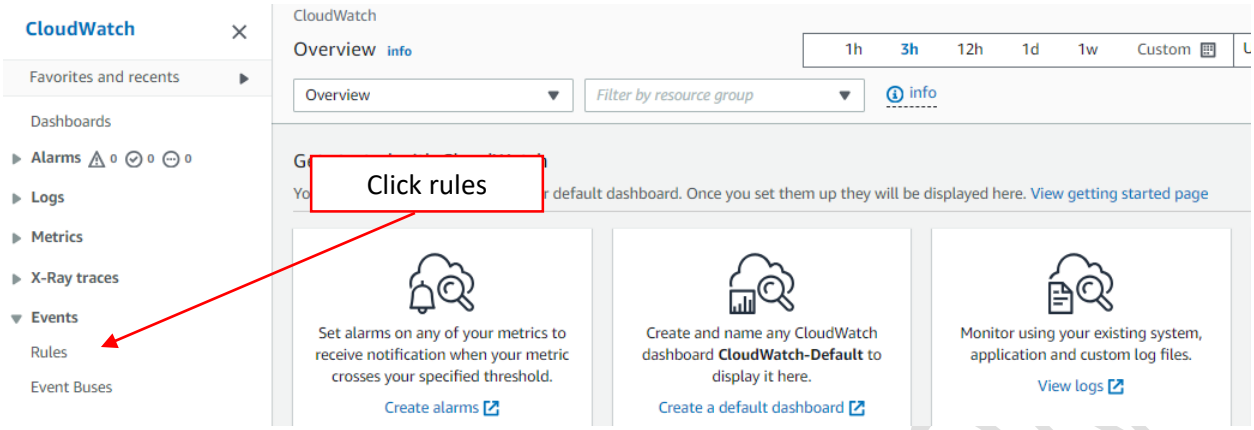

AWS Lambda
Cloud Institution
Dashboard
Applications
Functions
Function-Cloud-institute
Additional resources
Code signing configurations
Event source mappings
Layers
Replicas
Related AWS resources
Step Functions state machines
Code source Info
File Edit Find View Go Tools Window
Test
Deploy
Go to Anything (Ctrl-P)
lambda_function
Environment Var
Environment
Function-Cloud-inst
lambda_function.py
1 import boto3
2 region = 'us-east-1'
3 instances = ['i-041fe6e7507f2bbda']
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7 ec2.stop_instances(InstanceIds=instances)
8 print('stopped your instances: ' + str(instances))
Enter the region where you have created the instance
Enter the created instance ID

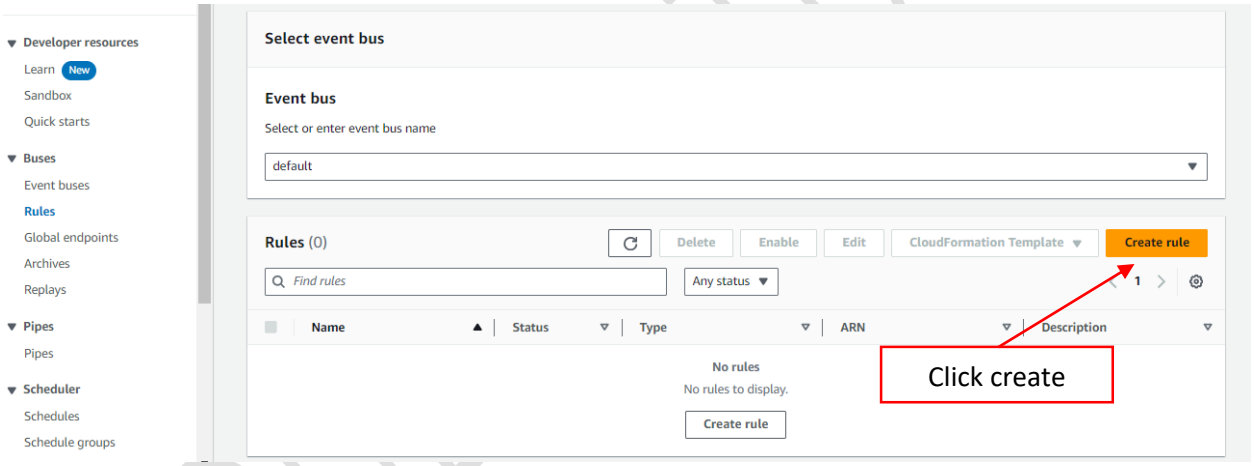

Code
Test
Monitor
Configuration
Aliases
Versions
Cloud Institution
Test event Info
Save
Test
Go to Test tab
To invoke your function without saving an event, configure the JSON event, then choose Test.
Click create
Test event action
Create new event
Edit saved event
Event name
EC2stop-cloud-institute
Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.
Event sharing settings
Give an event name
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

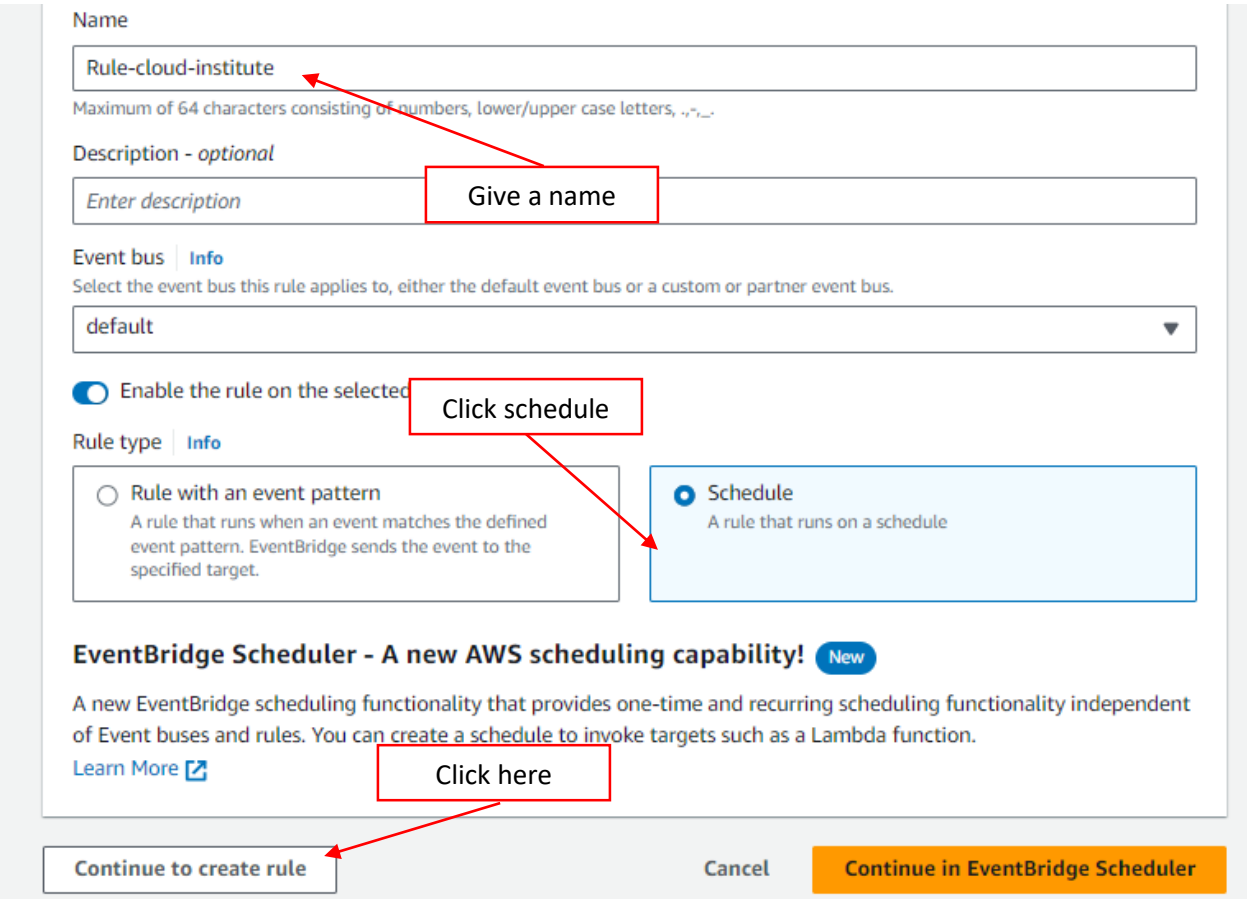


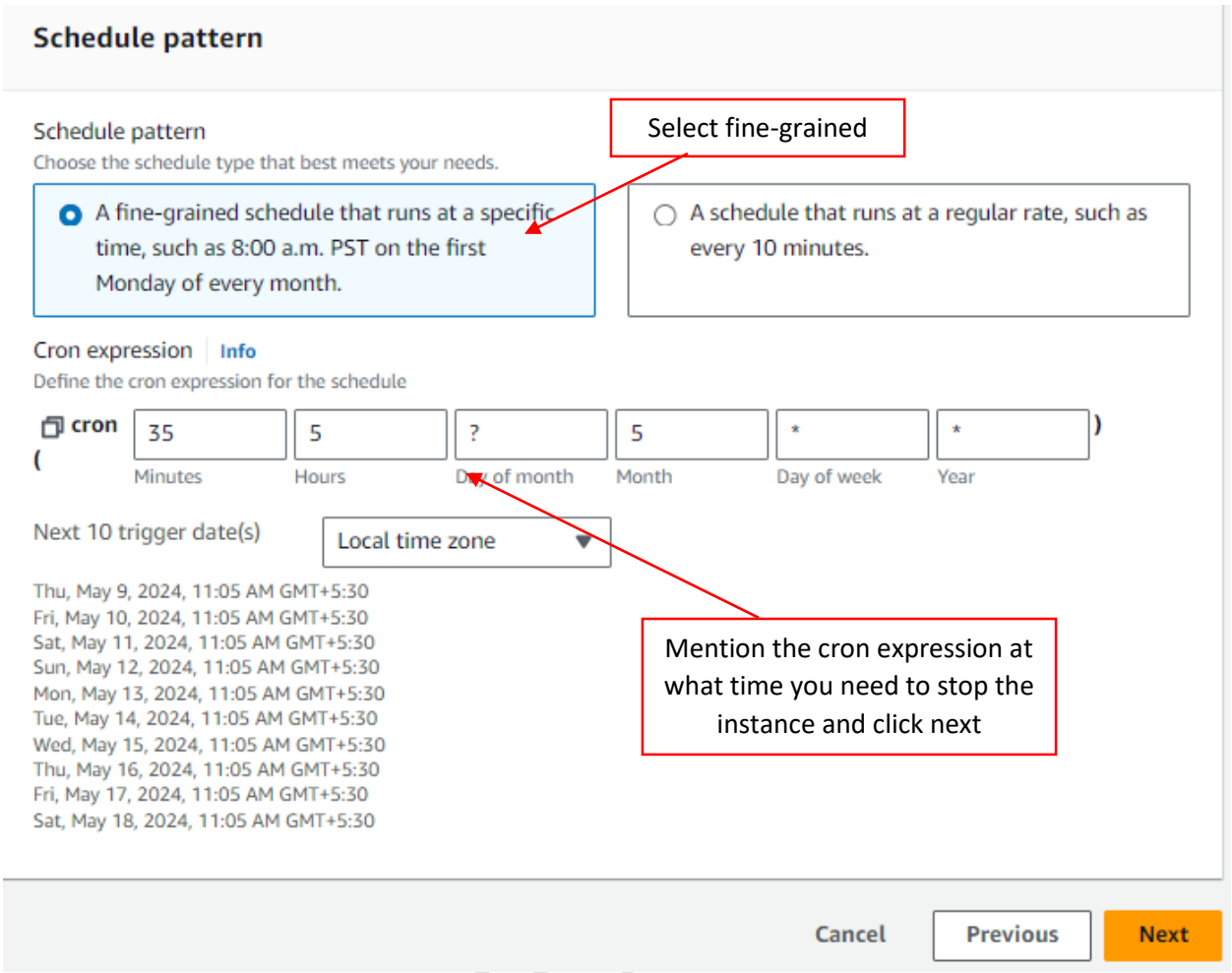
Step 5 : Go to Cloudwatch

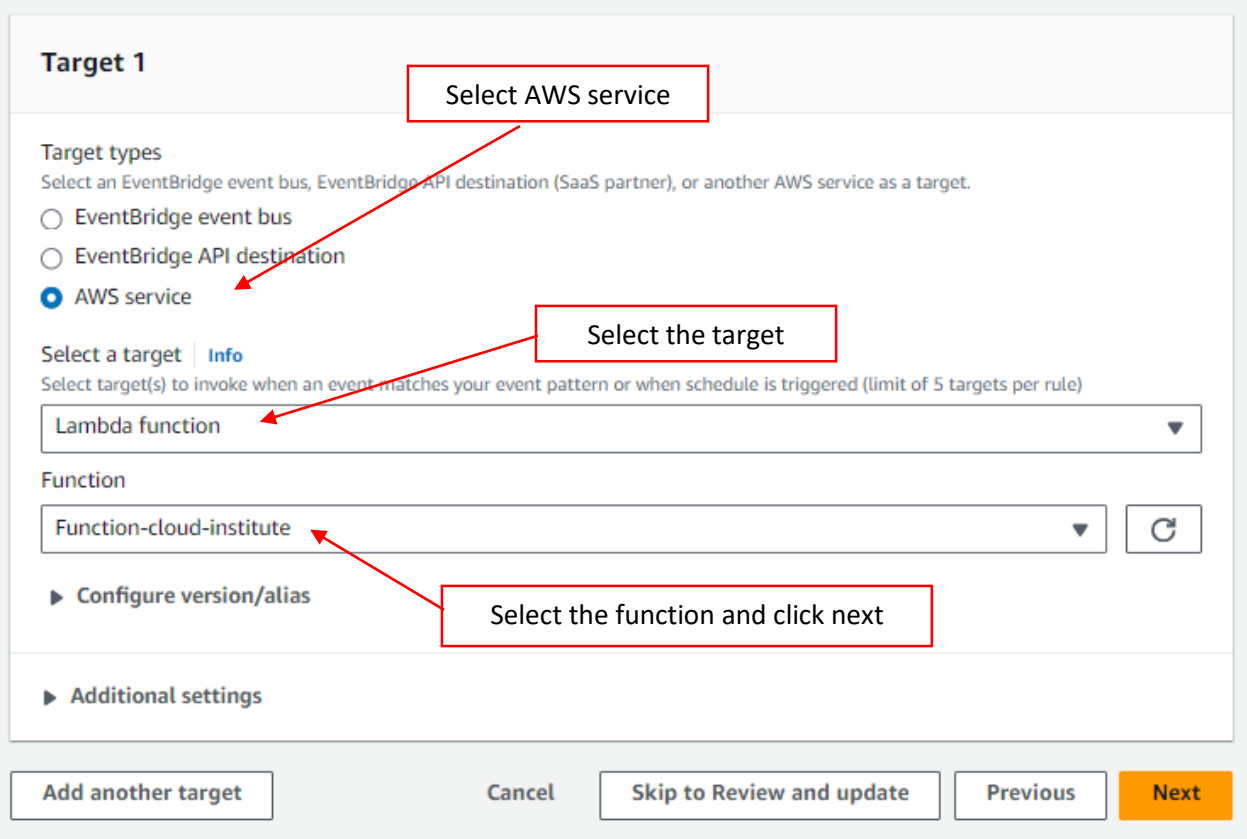


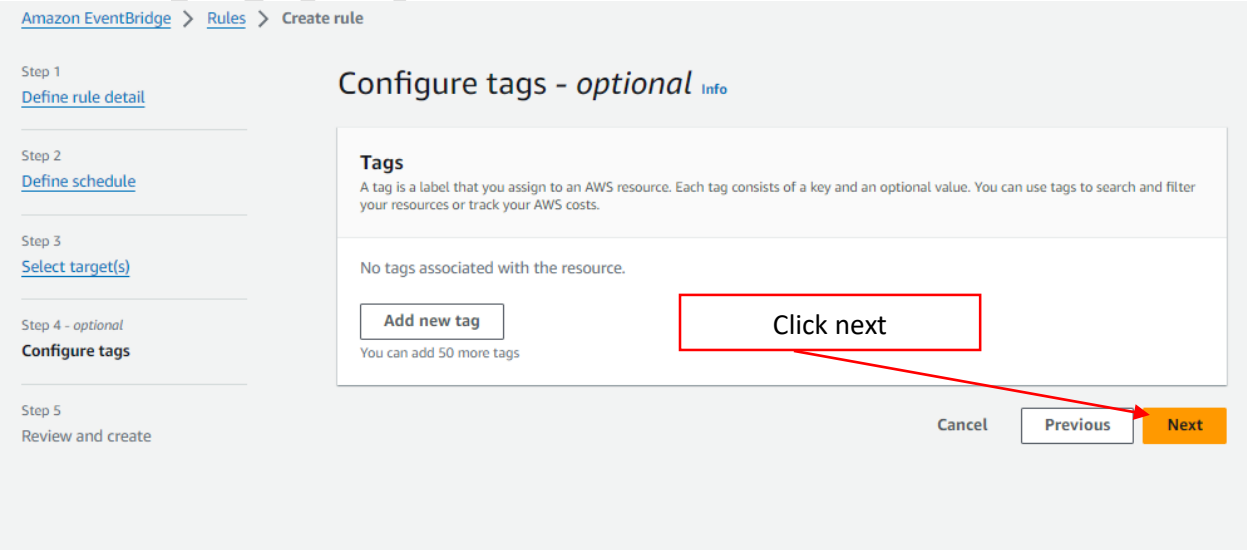

CloudWatch
Favorites and recents
Dashboards
Alarms
Logs
Metrics
X-Ray traces
Events
Rules
Event Buses
CloudWatch
Overview info
1h
3h
12h
1d
1w
Custom
Overview
Filter by resource group
info
Click rules
default dashboard. Once you set them up they will be displayed here. View getting started page
Set alarms on any of your metrics to receive notification when your metric crosses your specified threshold.
Create alarms
Create and name any CloudWatch dashboard CloudWatch-Default to display it here.
Create a default dashboard
Monitor using your existing system, application and custom log files.
View logs

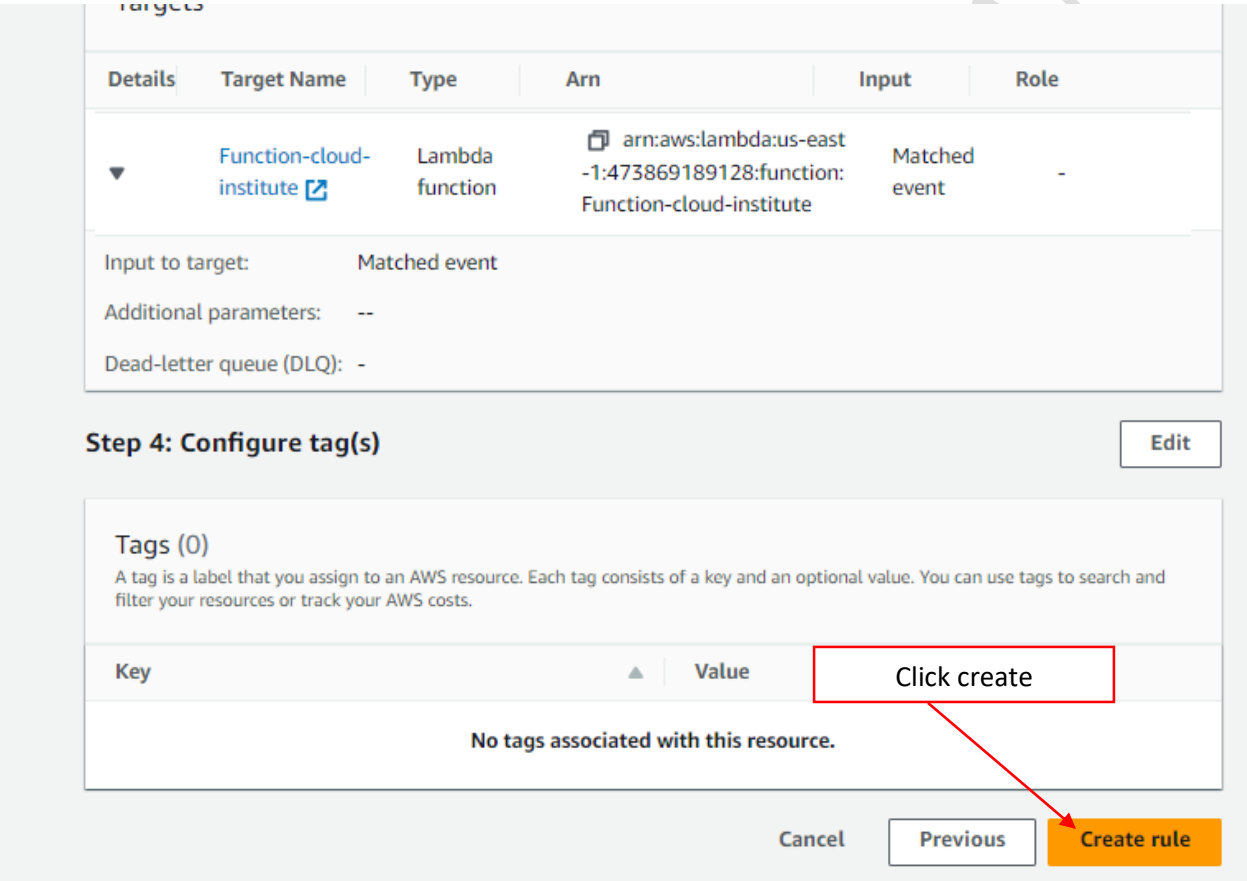

Developer resources
Learn
New
Sandbox
Quick starts
Buses
Event buses
Rules
Global endpoints
Archives
Replays
Pipes
Pipes
Scheduler
Schedules
Schedule groups
Select event bus
Event bus
Select or enter event bus name
default
Rules (0)
Delete
Enable
Edit
CloudFormation Template
Create rule
Find rules
Any status
Name
Status
Type
ARN
Description
No rules
No rules to display.
Create rule
Click create

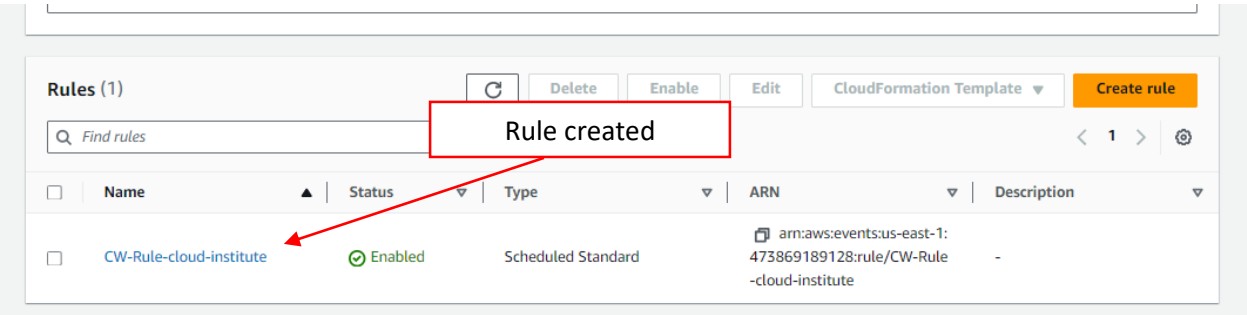

Name
Rule-cloud-institute
Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-,_.
Description - optional
Enter description
Give a name
Event bus Info
Select the event bus this rule applies to, either the default event bus or a custom or partner event bus.
default
Enable the rule on the selected
Click schedule
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
Click here
Continue to create rule
Cancel
Continue in EventBridge Scheduler



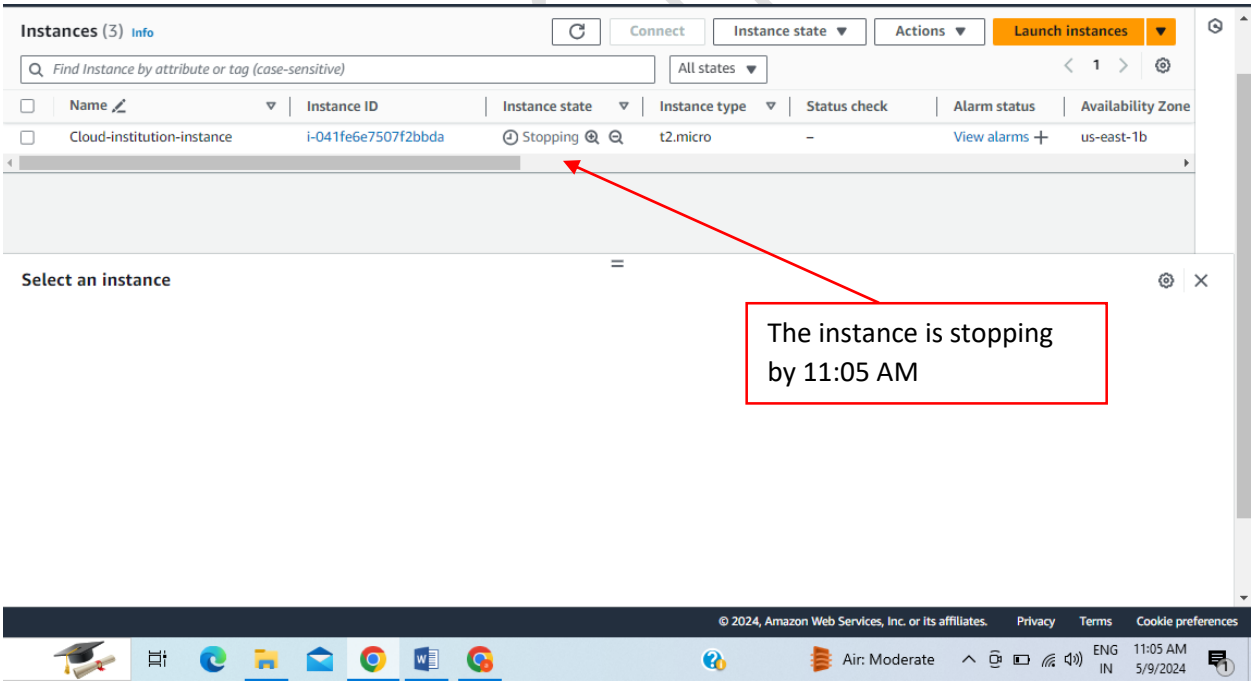

Target 1
Select AWS service
Target types
Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.
EventBridge event bus
EventBridge API destination
AWS service
Select the target
Select a target
Info
Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)
Lambda function
Function
Function-cloud-institute
Configure version/alias
Select the function and click next
Additional settings
Add another target
Cancel
Skip to Review and update
Previous
Next


Amazon EventBridge > Rules > 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
Configure tags - optional
Info
Tags
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.
No tags associated with the resource.
Add new tag
You can add 50 more tags
Click next
Cancel
Previous
Next


Details
Target Name
Type
Arn
Input
Role
Function-cloud-institute
Lambda function
arn:aws:lambda:us-east-1:473869189128:function:Function-cloud-institute
Matched event
-
Input to target: Matched event
Additional parameters: --
Dead-letter queue (DLQ): -
Step 4: Configure tag(s)
Edit
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
Click create
No tags associated with this resource.
Cancel
Previous
Create rule


Rules (1)
Delete
Enable
Edit
CloudFormation Template
Create rule
Find rules
Rule created
Name
Status
Type
ARN
Description
CW-Rule-cloud-institute
Enabled
Scheduled Standard
arn:aws:events:us-east-1:473869189128:rule/CW-Rule-cloud-institute

Step 6:


Instances (3) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Cloud-institution-instance
i-041fe6e7507f2bbda
Stopping
t2.micro
View alarms
us-east-1b
Select an instance
The instance is stopping by 11:05 AM
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Air: Moderate
ENG
IN
11:05 AM
5/9/2024