

AWS Monitoring

Day 3 - Assignment

27th July 2023

Assignment 1 - Creating multiple profiles and using them on the same server.

Step 1 - Create two IAM users with different permissions assigned to them.

- **IAM user with only EC2 full access permission.**

The screenshot shows the AWS IAM User Management console. At the top, there is a search bar labeled "Find users by username or access key". Below it is a table with columns: "User name", "Groups", "Last activity", "MFA", "Password age", and "Active key age". A single user entry is shown: "IAM_user-ShreyasK" with "None" in the Groups column, "5 hours ago" in the Last activity column, and "None" in the MFA column. The Password age and Active key age columns show "92 days ago" and "9 days ago" respectively. Below the table, there is a "User details" section for the newly created user. It includes fields for "User name" (set to "IAM-EC2_user"), a note about character restrictions, and a checkbox for "Provide user access to the AWS Management Console - optional". There is also a note about generating programmatic access keys. Finally, there is a "Permissions summary" section showing one policy named "AmazonEC2FullAccess" which is an "AWS managed" policy used as a "Permissions policy".

Enable console access for the user and set a password.

The screenshot shows the "Manage console access" dialog box. It has sections for "Console access" (with "Enable" selected), "Set password" (with "Custom password" selected and the value "IAM-EC2_user" entered), and "Show password" (which is checked). There is also a note about users creating new passwords at sign-in. At the bottom are "Cancel" and "Apply" buttons.

Create an Access Key for the user with command line access.

Access key best practices & alternatives [Info](#)

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code
You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Set description tag - optional [Info](#)

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: . : / = + - @

[Cancel](#) [Previous](#) [Create access key](#)

The secret access key should be saved securely and in a durable store as it is retrievable only at the time of creation. Make sure to download the .csv file.

Retrieve access keys [Info](#)

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIATOMGSBSVGRNSC6H7	***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

- **IAM user with only S3 full access permission.**

User details

User name

 The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

Provide user access to the AWS Management Console - *optional*
 If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type
 Specify a user in Identity Center - Recommended
 We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user
 We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

Console password
 Autogenerated password
 You can view the password after you create the user.
 Custom password
 Enter a custom password for the user.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } !'

Show password

Users must create a new password at next sign-in - Recommended
 Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

Create an Access Key for the user.

Access key best practices & alternatives Info

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

Command Line Interface (CLI)
 You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code
 You plan to use this access key to enable application code in a local development environment to access your AWS account.

Application running on an AWS compute service
 You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

Set description tag - optional Info

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

[Cancel](#)

[Previous](#)

[Create access key](#)

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

Secret access key

AKIATOMGSBSVFJ4KSGIY

***** Show

Step 2 - Go to the AWS CLI and run the “aws configure --profile <accountName>” command to set up multiple profiles for different IAM users in a single machine.

The .csv files for both the users' access key and secret access key, downloaded when creating the access keys.

A	B	C
Access key ID AKIATOMGSBSVGRNSC6H7	Secret access key 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	

A	B	C
Access key ID AKIATOMGSBSVFJ4KSGIY	Secret access key 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	

Run the configure command and enter the access key and secret key values of both the users.

```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws configure --profile ec2_user
AWS Access Key ID [None]: AKIATOMGSBSVGRNSC6H7
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws configure --profile s3_user
AWS Access Key ID [None]: AKIATOMGSBSVFJ4KSGIY
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
```

Use the “list-profiles” command to list the profiles.

```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws2 configure list-profiles
default
ec2_user
s3_user
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$
```

Step 3 - Try to access the S3 buckets with both the profiles and check the output.

With the use of “--profile <profileName>” option with AWS CLI commands we can use multiple profiles or IAM users with different permissions on the same machine.

The profile of “s3_user” can perform actions on S3 buckets as it is the profile of the IAM user with permission of “S3FullAccess”.

But the profile “ec2_user” is not able to perform any action on S3 because the IAM user does not have permissions for S3.

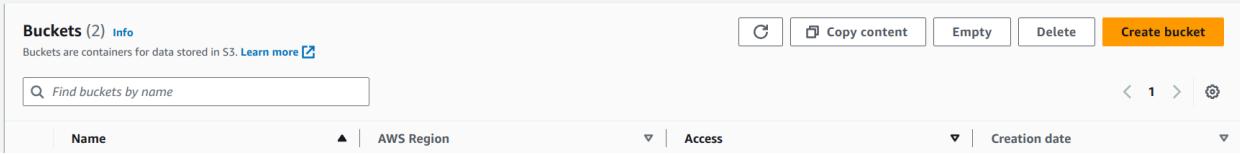
```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws2 s3 ls --profile s3_user
2023-07-27 10:54:41 aws-cloudtrail-logs-237042273450-300568af
2023-07-11 16:56:55 aws-cloudtrail-logs-237042273450-e4573353
2023-07-11 11:59:34 axcessstestbucket1
2023-07-11 08:59:47 test-bucket-sk
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws2 s3 ls --profile ec2_user
An error occurred (AccessDenied) when calling the ListBuckets operation: Access
Denied
17:39:27 - awscliv2 - ERROR - Command failed with code 254
```

Same with the “ec2_user” as it only has permissions for EC2FullAccess, it cannot access S3 or other services.

```
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws2 ec2 describe-instances --profile ec2_user
{
    "Reservations": []
}
shreyaskayarkar@rahulraj-TravelMate-P214-53:~$ aws2 ec2 describe-instances --profile s3_user
An error occurred (UnauthorizedOperation) when calling the DescribeInstances operation: You are not authorized
to perform this operation.
17:42:25 - awscliv2 - ERROR - Command failed with code 254
```

Assignment 2 - Creating CloudTrail data events for S3.

Step 1 - Create a sample S3 bucket.



General configuration

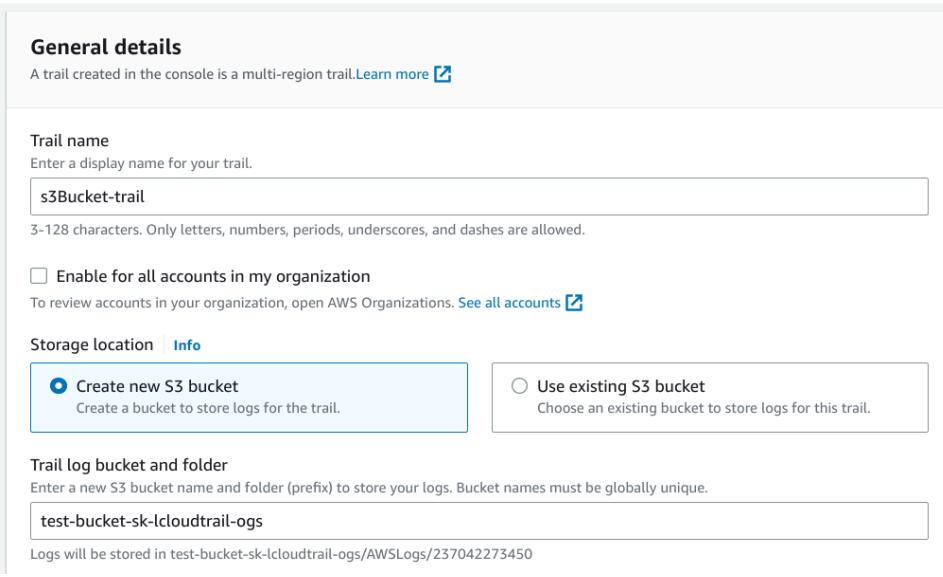
Bucket name: test-bucket-sk1

AWS Region: US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.
[Choose bucket](#)

Step 2 - Create a CloudTrail data event for the S3 bucket.

Create a new bucket with a unique and identifiable name. This S3 bucket will store the logs of this trail.



General details
A trail created in the console is a multi-region trail.[Learn more](#)

Trail name: s3Bucket-trail
Enter a display name for your trail.
3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

Enable for all accounts in my organization
To review accounts in your organization, open AWS Organizations. [See all accounts](#)

Storage location | [Info](#)

Create new S3 bucket
Create a bucket to store logs for the trail.

Use existing S3 bucket
Choose an existing bucket to store logs for this trail.

Trail log bucket and folder
Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique.
test-bucket-sk-lcloudtrail-ogs

Logs will be stored in test-bucket-sk-lcloudtrail-ogs/AWSLogs/237042273450

Enable CloudWatch logs, create a new log group which will store the logs coming for this trail.

CloudWatch Logs - optional
Configure CloudWatch Logs to monitor your trail logs and notify you when specific activity occurs. Standard CloudWatch and CloudWatch Logs charges apply. [Learn more](#)

CloudWatch Logs | [Info](#)

Enabled

New
 Existing

Log group name

1-512 characters. Only letters, numbers, dashes, underscores, forward slashes, and periods are allowed.

AWS CloudTrail assumes this role to send CloudTrail events to your CloudWatch Logs log group.

New
 Existing

Role name

► [Policy document](#)

Select “Data events”, which will allow the trail to collect resource level events.

Event type
Choose the type of events that you want to log.

Management events
Capture management operations performed on your AWS resources.

Data events
Log the resource operations performed on or within a resource.

Insights events
Identify unusual activity, errors, or user behavior in your account.

The custom event will allow the trail to collect logs for whenever an object is downloaded from the bucket with the specified ARN.

Specifying the resource.ARN field will collect logs only for the put and get object requests with ARN starting from the specified value. It should be ensured that only selected buckets should be selected or the bucket in which the logs are to be stored should be excluded by specifying its resource ARN. It will avoid any logging loop.

▼ Data event: S3

Remove

Data event type
Choose the source of data events to log.
S3

Log selector template
Custom

Selector name - optional
Enter a name
1,000 character limit

Collect events
Log all events, or choose a template to log specific, filtered events to your trail. You can edit templates later.

Advanced event selectors | Info
Log or exclude events from specific resources.

Field Operator Value
eventName equals GetObject X

AND + Condition

resources.ARN equals arn:aws:s3:::test-bucket-sk1/ Browse X

+ Field + Condition

This screenshot shows the AWS CloudTrail configuration interface for creating a new data event. The 'Data event type' is set to 'S3'. The 'Log selector template' is 'Custom'. In the 'Collect events' section, there is an 'Advanced event selectors' panel with two conditions defined using the 'eventName' and 'resources.ARN' fields. The first condition checks if the event name equals 'GetObject'. The second condition checks if the resource ARN equals 'arn:aws:s3:::test-bucket-sk1/'. Both conditions are connected by an 'AND' operator.

This event will collect logs whenever an object is uploaded to the bucket.

▼ Data event: S3

Remove

Data event type
Choose the source of data events to log.
S3

Log selector template
Custom

Selector name - optional
Enter a name
1,000 character limit

Collect events
Log all events, or choose a template to log specific, filtered events to your trail. You can edit templates later.

Advanced event selectors | Info
Log or exclude events from specific resources.

Field Operator Value
eventName equals PutObject X

AND + Condition

resources.ARN equals arn:aws:s3:::test-bucket-sk1/ Browse X

+ Field + Condition

This screenshot shows the AWS CloudTrail configuration interface for creating a new data event. The 'Data event type' is set to 'S3'. The 'Log selector template' is 'Custom'. In the 'Collect events' section, there is an 'Advanced event selectors' panel with two conditions defined using the 'eventName' and 'resources.ARN' fields. The first condition checks if the event name equals 'PutObject'. The second condition checks if the resource ARN equals 'arn:aws:s3:::test-bucket-sk1/'. Both conditions are connected by an 'AND' operator.

Step 3 - Upload files to the S3 bucket.

Upload [Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (2 Total, 196.0 KB)		Remove	Add files	Add folder
All files and folders in this table will be uploaded.				
<input type="text"/> Find by name < 1 >				
<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	Screenshot from 20...	-	image/png	93.1 KB
<input type="checkbox"/>	Screenshot from 20...	-	image/png	102.9 KB

Step 4 - Checking the CloudWatch log group for the logs of the “PutObject” event.

Step 5 - Check the S3 bucket used for storing the logs.

Amazon S3 > Buckets > test-bucket-sk-cloudtrail-logs > AWSLogs/ > 237042273450/ > CloudTrail/ > us-east-1/ > 2023/ > 07/ > 27/ > 237042273450_CloudTrail_us-east-1_20230727T1340Z_ob0lUTm9MocV2oI7.json.gz

237042273450_CloudTrail_us-east-1_20230727T1340Z_ob0lUTm9MocV2oI7.json.gz [Info](#)

[Copy S3 URI](#) [Download](#) [Open](#) [Object actions](#)

[Properties](#) [Permissions](#) [Versions](#)

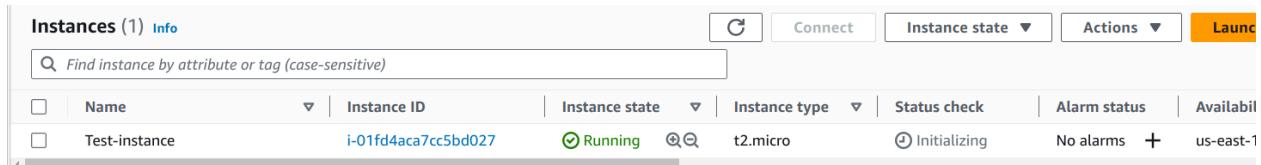
Object overview

Owner	S3 URI
shreyas.k7407	s3://test-bucket-sk-cloudtrail-logs/AWSLogs/237042273450/CloudTrail/7/237042273450_CloudTrail_us-east-1_20230727T1340Z_ob0lUTm9MocV2oI7.json.gz
AWS Region	Amazon Resource Name (ARN)
US East (N. Virginia) us-east-1	arn:aws:s3:::test-bucket-sk-cloudtrail-logs/AWSLogs/237042273450/CloudTrail/7/237042273450_CloudTrail_us-east-1_20230727T1340Z_ob0lUTm9MocV2oI7.json.gz
Last modified	Entity tag (Etag)
July 27, 2023, 19:14:38 (UTC+05:30)	046801019f5ca3a8f8c8fb212bf643af
Size	Object URL
1.2 KB	https://test-bucket-sk-cloudtrail-logs.s3.amazonaws.com/AWSLogs/237042273450/CloudTrail/7/237042273450_CloudTrail_us-east-1_20230727T1340Z_ob0lUTm9MocV2oI7.json.gz
Type	
gz	

```
{
  "Records": [
    {
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDATOMGSBSVAAKATBQ3F",
        "arn": "arn:aws:iam::237042273450:user/IAM_user-ShreyasK",
        "accountId": "237042273450",
        "accessKeyId": "ASIAJOMGSBSVNRT5IPMT",
        "userName": "IAM_user-ShreyasK",
        "sessionContext": {
          "attributes": {
            "creationDate": "2023-07-27T05:47:13Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2023-07-27T13:39:34Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "PutObject",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "103.6.157.71",
      "userAgent": "[Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36]",
      "requestParameters": {
        "X-Amz-Date": "20230727T133932Z",
        "bucketName": "test-bucket-sk1",
        "X-Amz-Algorithm": "AWS4-HMAC-SHA256",
        "x-amz-acl": "bucket-owner-full-control"
      }
    }
  ]
}
```

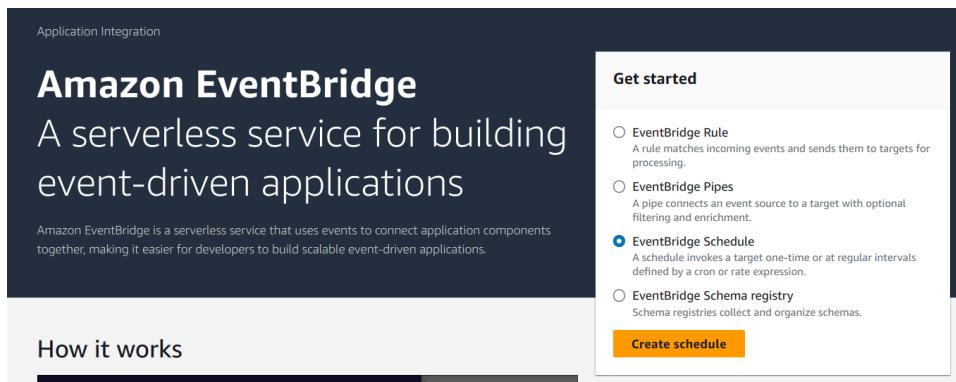
Assignment 3 - Using event bridge to schedule automatic starting and stopping EC2 instances on a schedule.

Step 1 - Launch an EC2 instance.



The screenshot shows the AWS CloudWatch Metrics console with a single instance listed. The instance is named "Test-instance", has an ID of "i-01fd4aca7cc5bd027", is in a "Running" state, is of type "t2.micro", and is in the "us-east-1" region. There are no alarms defined for this instance.

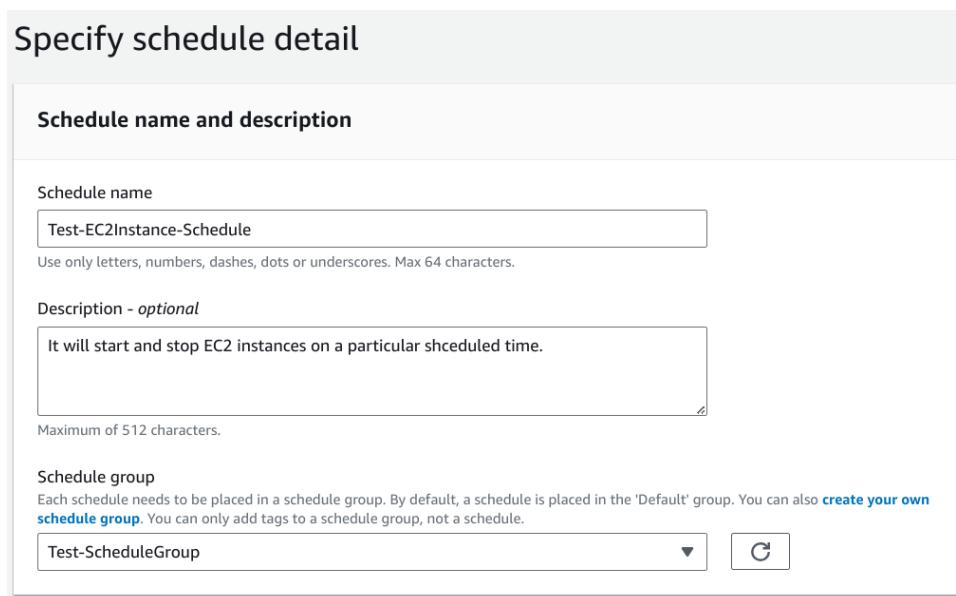
Step 2 - Navigate to the EventBridge console. Select event scheduler and click on “Create schedule”



The screenshot shows the Amazon EventBridge console. On the left, there's a sidebar with "Application Integration" and a "How it works" section. On the right, under "Get started", there are four options: "EventBridge Rule" (a rule matches incoming events and sends them to targets for processing), "EventBridge Pipes" (a pipe connects an event source to a target with optional filtering and enrichment), "EventBridge Schedule" (selected, described as a schedule invokes a target one-time or at regular intervals defined by a cron or rate expression), and "EventBridge Schema registry" (schema registries collect and organize schemas). A prominent orange "Create schedule" button is at the bottom of the "EventBridge Schedule" section.

Create a schedule for starting the instance at a scheduled time.

Schedule groups are used to organize the schedules.



The screenshot shows the "Specify schedule detail" form. It includes fields for "Schedule name and description", "Schedule name" (set to "Test-EC2Instance-Schedule"), and a "Description - optional" field containing the text "It will start and stop EC2 instances on a particular shceduled time.". Below these, there's a "Maximum of 512 characters." note. At the bottom, there's a "Schedule group" section with a dropdown menu set to "Test-ScheduleGroup" and a "Create" button.

Schedule pattern

Occurrence | [Info](#)

You can define a one-off or recurrent schedule.

One-off schedule

Recurring schedule

Schedule type

Choose the schedule type that best meets your needs.

Cron-based schedule

A schedule set using a cron expression that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

Rate-based schedule

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

This cron expression will trigger an event everyday at 14:35 hours.

Cron expression | [Info](#)

Define the cron expression for the schedule

[Copy](#) [X](#) [Clear](#)

cron (35 14 * * ? *)

Minutes Hours Day of month Month Day of the week Year

Next 10 trigger date

Date and time are displayed in your current time zone in UTC format, e.g. 'Wed, Nov 9, 2022 09:00 (UTC - 08:00)' for Pacific time

Sat, 29 Jul 2023 14:35:00 (UTC+05:30)

Sun, 30 Jul 2023 14:35:00 (UTC+05:30)

Mon, 31 Jul 2023 14:35:00 (UTC+05:30)

Tue, 01 Aug 2023 14:35:00 (UTC+05:30)

Wed, 02 Aug 2023 14:35:00 (UTC+05:30)

Thu, 03 Aug 2023 14:35:00 (UTC+05:30)

Time will be based on the selected timezone. Here it is set to Asia/Calcutta (IST).

Timeframe

Daylight saving time

Amazon EventBridge scheduler automatically adjusts your schedule for daylight saving time. When the clocks go forward in spring, if a cron expression falls on a non-existent date, your schedule invocation is skipped. When the clocks go back in autumn, your schedule runs only once and does not repeat its invocation. The following invocations occur normally at the specified date and time.

Time zone

The time zone for the schedule.

(UTC+05:30) Asia/Calcutta



Start date and time - optional

The start date and time of the schedule.

YYYY/MM/DD



hh:mm

YYYY/MM/DD hh:mm

Use 24-hour format timestamp (hh:mm)

End date and time - optional

The end date and time of the schedule.

YYYY/MM/DD



hh:mm

YYYY/MM/DD hh:mm

Use 24-hour format timestamp (hh:mm)

Cancel

Next

Target detail

Target API | [Info](#)

Select an API that will be invoked as a target for your schedule.

Templatized targets All APIs

All AWS services > Amazon EC2

Search: start | 5 of 389

Amazon EC2 StartInstances

Amazon EC2 ModifyInstanceEv...

Amazon EC2 StartNetworkInsti...

Amazon EC2 StartVpcEndpoint...

Amazon EC2 StartNetworkInsti...

As selected above, the event will be a StartInstance API call for the instance with the instance id specified in the below input section.

Enter the instance id of the instance created in the first step.

Input
JSON object containing the parameters to pass into the API. Contains sample values. Update the JSON with your own values. Note: parameter names must be in PascalCase. [Learn more](#)

```
1 ▼ {  
2 ▼ "InstanceIds": [  
3   "i-01fd4aca7cc5bd027"  
4 ]  
5 }
```

The schedule can be enabled after its creation. Here the retry policy is set for 1 hour after the scheduled time, in which there are 5 retry attempts.

Schedule state

Enable schedule
You can choose not to enable the schedule now. You will be able to enable the schedule after it has been created.

Enable

Retry policy and dead-letter queue (DLQ)

Retry policy [Info](#)
 By default, EventBridge Scheduler attempts to retry failed invocations for up to 24 hours. You can specify the maximum age of the event and the maximum number of times to retry.

Retry

Maximum age of event - optional
 The maximum amount of time to keep unprocessed events. The maximum and default value is 24 hours.

hour(s) minute(s)

Retry attempts - optional
 The maximum number of times to retry when a target returns an error. The maximum value is 185 times.

times

Dead-letter queue (DLQ)
 Standard Amazon SQS queues that EventBridge Scheduler uses to store events that couldn't be delivered successfully to a target.

None

- Select an Amazon SQS queue in my AWS account as a DLQ
- Specify an Amazon SQS queue in other AWS accounts as a DLQ

Creating an IAM role for the scheduler to perform actions on the EC2 instance.

Role details

Role name
 Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+-=_,@-_` characters.

Description
 Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+-=_,@-_` characters.

Here, a custom trust policy is created which will allow the scheduler to assume the attached role.

Step 1: Select trusted entities

[Edit](#)

```

1  [
2   {
3     "Version": "2012-10-17",
4     "Statement": [
5       {
6         "Sid": "Statement1",
7         "Effect": "Allow",
8         "Principal": {
9           "Service": "scheduler.amazonaws.com"
10        },
11        "Action": "sts:AssumeRole"
12      }
13    ]
  ]

```

The role has permission policy attached for “EC2FullAccess”.

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2FullAccess	AWS managed	Permissions policy

Permissions Info

Permissions

EventBridge Scheduler requires permission to send events to the target and, based on the preferences you select, integrate with other AWS services, such as AWS KMS and Amazon SQS.

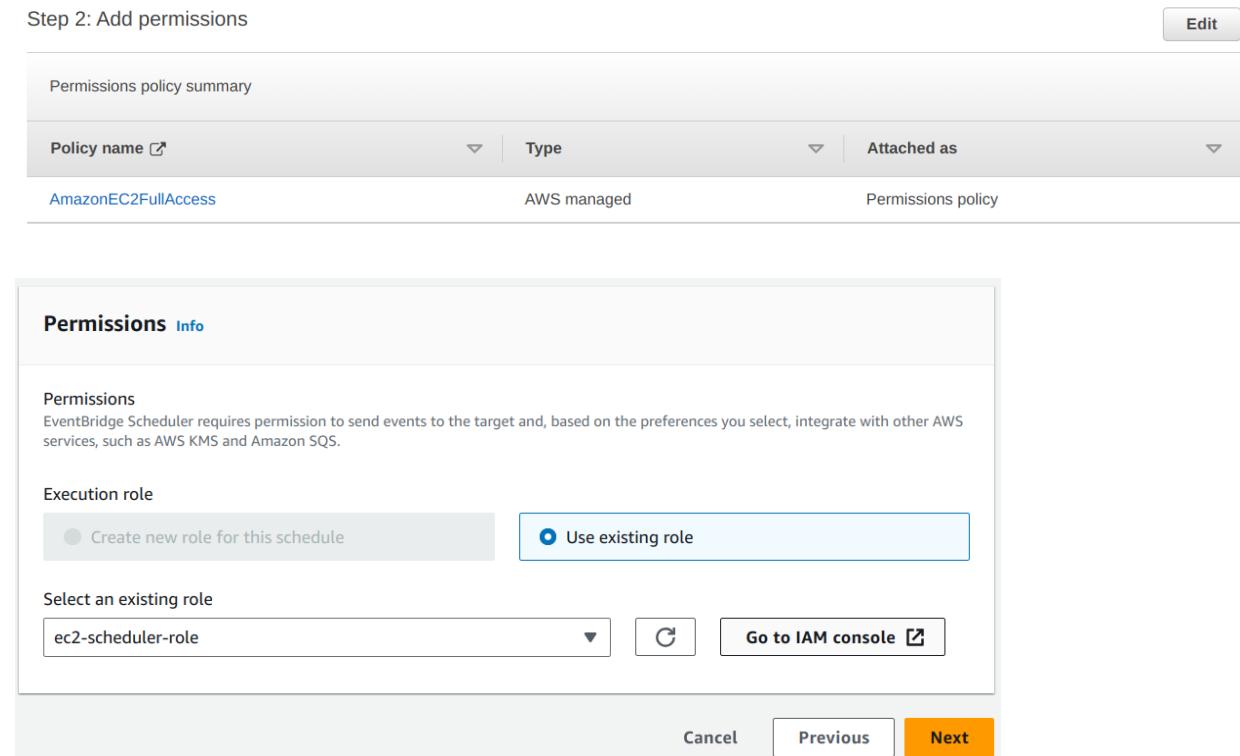
Execution role

Create new role for this schedule Use existing role

Select an existing role

ec2-scheduler-role [Go to IAM console](#)

Cancel Previous Next



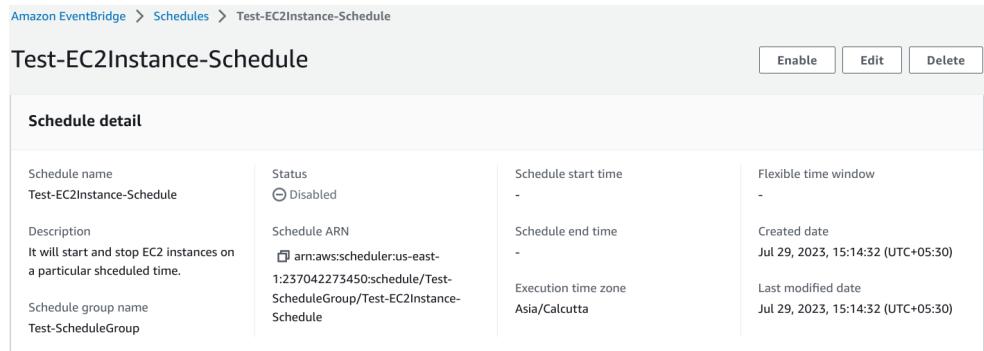
The schedule is created and is initially disabled.

Amazon EventBridge > Schedules > Test-EC2Instance-Schedule

Test-EC2Instance-Schedule

Enable Edit Delete

Schedule detail			
Schedule name Test-EC2Instance-Schedule	Status <input type="radio"/> Disabled	Schedule start time -	Flexible time window -
Description It will start and stop EC2 instances on a particular scheduled time.	Schedule ARN arn:aws:scheduler:us-east-1:237042273450:schedule/Test-ScheduleGroup/Test-EC2Instance-Schedule	Schedule end time -	Created date Jul 29, 2023, 15:14:32 (UTC+05:30)
Schedule group name Test-ScheduleGroup	Execution time zone Asia/Calcutta	Last modified date Jul 29, 2023, 15:14:32 (UTC+05:30)	



Step 3 - Create a schedule for stopping the instance at a scheduled time.

Schedule name and description

Schedule name
Use only letters, numbers, dashes, dots or underscores. Max 64 characters.

Description - optional
Maximum of 512 characters.

Schedule group
Each schedule needs to be placed in a schedule group. By default, a schedule is placed in the 'Default' group. You can also [create your own schedule group](#). You can only add tags to a schedule group, not a schedule.
▼

Schedule pattern

Occurrence [Info](#)
You can define a one-off or recurrent schedule.
 One-off schedule Recurring schedule

Schedule type
Choose the schedule type that best meets your needs.

Cron-based schedule
A schedule set using a cron expression that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

Rate-based schedule
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 the week Year

Next 10 trigger date
Date and time are displayed in your current time zone in UTC format, e.g. 'Wed, Nov 9, 2022 09:00 (UTC - 08:00)' for Pacific time

Sat, 29 Jul 2023 16:10:00 (UTC+05:30)
Sun, 30 Jul 2023 16:10:00 (UTC+05:30)
Mon, 31 Jul 2023 16:10:00 (UTC+05:30)
Tue, 01 Aug 2023 16:10:00 (UTC+05:30)
Wed, 02 Aug 2023 16:10:00 (UTC+05:30)
Thu, 03 Aug 2023 16:10:00 (UTC+05:30)

The target API call is to stop the instance.

Target detail

Target API [Info](#)
Select an API that will be invoked as a target for your schedule.

Templated targets All APIs

All AWS services > Amazon EC2

X 1 of 389 < 1 ... >

 Amazon EC2
StopInstances

Enter the instance id of the EC2 instance.

StopInstances
Amazon EC2

Input
JSON object containing the parameters to pass into the API. Contains sample values. Update the JSON with your own values. Note: parameter names must be in PascalCase. [Learn more](#)

```
1 ▼ {  
2 ▼ "InstanceIds": [  
3 "i-01fd4aca7cc5bd027"  
4 ]  
5 }
```

Schedule state

Enable schedule
You can choose not to enable the schedule now. You will be able to enable the schedule after it has been created.
 Enable

Retry policy and dead-letter queue (DLQ)

Retry policy [Info](#)
By default, EventBridge Scheduler attempts to retry failed invocations for up to 24 hours. You can specify the maximum age of the event and the maximum number of times to retry.
 Retry

Maximum age of event - optional
The maximum amount of time to keep unprocessed events. The maximum and default value is 24 hours.
 hour(s) minute(s)

Retry attempts - optional
The maximum number of times to retry when a target returns an error. The maximum value is 185 times.
 times

Permissions [Info](#)

Permissions
EventBridge Scheduler requires permission to send events to the target and, based on the preferences you select, integrate with other AWS services, such as AWS KMS and Amazon SQS.

Execution role

Create new role for this schedule Use existing role

Select an existing role

ec2-scheduler-role [Go to IAM console](#)

Step 4 - Check if both the schedulers are working as intended.

Stop the instance manually and check if it is started by the scheduler.

Change the cron expression of the start schedule.

Cron expression [Info](#)

Define the cron expression for the schedule

cron (55 15 * * ? *)

Minutes Hours Day of month Month Day of the week Year

Next 10 trigger date

Date and time are displayed in your current time zone in UTC format, e.g. 'Wed, Nov 9, 2022 09:00 (UTC - 08:00)' for Pacific time

Sat, 29 Jul 2023 15:55:00 (UTC+05:30)
Sun, 30 Jul 2023 15:55:00 (UTC+05:30)
Mon, 31 Jul 2023 15:55:00 (UTC+05:30)
Tue, 01 Aug 2023 15:55:00 (UTC+05:30)
Wed, 02 Aug 2023 15:55:00 (UTC+05:30)

Enable the schedule as it was not enabled when it was created.

Enable schedule

Are you sure you want to enable schedule **Test-EC2Instance-Schedule**?

[Cancel](#) [Enable](#)

The instance is currently in stopped state.

Instances (1) [Info](#)

[Find instance by attribute or tag \(case-sensitive\)](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Ala
<input type="checkbox"/>	Test-instance	i-01fd4aca7cc5bd027	Stopped	t2.micro	-	No

Instances (1) Info		C	Connect	Instance state ▾	Alarms
<input type="text"/> Find instance by attribute or tag (case-sensitive)					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>	Test-instance	i-01fd4aca7cc5bd027	Running	t2.micro	Initializing No

CloudTrail event history shows that the instance was started at 15:55 by the event scheduler.

Event history (50+) Info						C	Download events ▾	Create Athena table	
Lookup attributes						Read-only	<input type="text"/> false	Filter by date and time	1 2 ... > ⚙️
<input type="checkbox"/>	Event name	Event time	User name	Event source	Resource type	Resource name			
<input type="checkbox"/>	StartInstances	July 29, 2023, 15:55:26 (UTC...)	81e73594f88b3b3a...	ec2.amazonaws.com	AWS::EC2::Instance	i-01fd4aca7cc5bd027			
<input type="checkbox"/>	UpdateSchedule	July 29, 2023, 15:53:19 (UTC...)	IAM_user-ShreyasK	scheduler.amazonaws.c...	AWS::Scheduler::Sc...	arn:aws:scheduler:us-e...			
<input type="checkbox"/>	UpdateSchedule	July 29, 2023, 15:53:03 (UTC...)	IAM_user-ShreyasK	scheduler.amazonaws.c...	AWS::Scheduler::Sc...	arn:aws:scheduler:us-e...			
<input type="checkbox"/>	StopInstances	July 29, 2023, 15:51:52 (UTC...)	IAM_user-ShreyasK	ec2.amazonaws.com	AWS::EC2::Instance	i-01fd4aca7cc5bd027			

Now checking if the instance stops again at the scheduled time of 16:10.

Currently it is in running state.

Instances (1) Info						C	Connect	Instance state ▾	
<input type="text"/> Find instance by attribute or tag (case-sensitive)									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Read-only	<input type="text"/> false	Filter by date and time	1 2 ... > ⚙️
<input type="checkbox"/>	Test-instance	i-01fd4aca7cc5bd027	Running	t2.micro	2/2 checks passed				

The instance turns to the stopping state at the scheduled time i.e. 16:10 hours.

Instances (1) Info						C	Connect	Instance state ▾	
<input type="text"/> Find instance by attribute or tag (case-sensitive)									
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Read-only	<input type="text"/> false	Filter by date and time	1 2 ... > ⚙️
<input type="checkbox"/>	Test-instance	i-01fd4aca7cc5bd027	Stopping	t2.micro	-				

CloudTrail event history shows that the instance was stopped at 16:10 hours by the scheduler.

Event history (50+) Info						C	Download events ▾	Create Athena table	
Lookup attributes						Read-only	<input type="text"/> false	Filter by date and time	1 2 ... > ⚙️
<input type="checkbox"/>	Event name	Event time	User name	Event source	Resource type	Resource name			
<input type="checkbox"/>	StopInstances	July 29, 2023, 16:10:07 (UTC...)	81e73594f88b3b3a...	ec2.amazonaws.com	AWS::EC2::Instance	i-01fd4aca7cc5bd027			

Assignment 4 - Create a SNS alert to send notification if instances in an ASG are manually terminated.

Step 1 - Create a launch template.

Launch template name and description

Launch template name - required
Test-template
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description
initial version

Quick Start

Recent AMIs: Don't include in launch template, Amazon Linux, macOS, Ubuntu, Windows, Red Hat

Browse more AMIs: Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-0f34c5ae932e6f0e4 (64-bit (x86)) / ami-0964d1dc1edd4bd2f (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible ▾

Description
Amazon Linux 2023 AMI 2023.1.20230725.0 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86) ▾ AMI ID: ami-0f34c5ae932e6f0e4 Verified provider

Instance type [Info](#) [Advanced](#)

Instance type: t2.micro
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

Free tier eligible ▾ All generations Compare instance types

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: asg-Pem [Create new key pair](#)

▼ Network settings [Info](#)

Subnet Info

subnet-051f9be32ca056d18
VPC: vpc-0b7165e7709485a86 Owner: 237042273450
Availability Zone: us-east-1a IP addresses available: 4091 CIDR: 172.31.0.0/20

When you specify a subnet, a network interface is automatically added to your template.

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.

Select existing security group Create security group

Common security groups [Info](#)

Select security groups

default sg-05fdffebcfefbf6de X
VPC: vpc-0b7165e7709485a86

[Compare security group rules](#)

Step 2 - Create an Auto Scaling Group.

Name

Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#) [Switch to launch configuration](#)

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
 [Create a launch template](#)

Version
Default (1) [Create a launch template version](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
 [Create a VPC](#)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
 [Create a subnet](#)

us-east-1a | subnet-051f9be32ca056d18 X
172.31.0.0/20 Default

Instance type requirements [Info](#)

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
Test-template	Default	initial version

Instance type
t2.micro

Group size - optional [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
2

Minimum capacity
2

Maximum capacity
2

The capacity is updated and two instances are launched.

Instances (2)						
<input type="button" value="Actions ▾"/> ◀ 1 ▶ ⌂						
<input type="text"/> Filter instances						
□	Instance ID	▲ Lifecycle	▼ Instance type	▼ Weighted ca...	▼ Launch tem...	▼ Availability ...
□	i-0ace801d501bfd34e	InService	t2.micro	-	Test-template	us-east-1a Healthy
□	i-0fdbb8894cd2bcd6	InService	t2.micro	-	Test-template	us-east-1a Healthy

Step 3 - Create a rule.

Select the default event bus.

Select event bus

Event bus

Select or enter event bus name

default

Define rule detail [Info](#)

Rule detail

Name
Test-EC2-termination
Maximum of 64 characters consisting of numbers, lower/upper case letters, .,-_,.

Description - *optional*

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

[Cancel](#) [Next](#)

The event source is events sent from AWS services.

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.

The event pattern will catch the events which match the pattern. Here it will catch EC2 instance manual termination events (source EC2 console).

<input type="radio"/> Use schema Use an Amazon EventBridge schema to generate the event pattern.	<input checked="" type="radio"/> Use pattern form Use a template provided by EventBridge to create an event pattern.	<input type="radio"/> 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 ▾

Any state
 Specific state(s)
shutting-down X terminated X

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": ["shutting-down", "terminated"]
6   }
7 }
```

Copy Test pattern Edit pattern

Add a “anything-but matching” pattern to catch events for only manual termination of the instances not by auto scaling group.

Event pattern Info

Event pattern
Write an event pattern in JSON. You can test the event pattern against the sample event. You can also go to pre-defined pattern.

Prefix matching ▾ Insert Content-based filter syntax

```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change Notification"],
4   "detail": {
5     "state": ["shutting-down", "terminated"]
6   },
7   "userAgent": [
8     "anything-but": ["autoscaling.amazonaws.com"]
9   ]
10 }
```

The rule will send an alert through the SNS topic for the event.

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)

SNS topic ▾

Topic
ASG-Alarm-SNS ▾ C

► Additional settings

Add another target Cancel Skip to Review and create Previous Next

Select Input transformer, it will send a formatted message through the SNS notification. Then click on “Configure input transformer”.

▼ 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.

Input transformer

Configure input transformer

Retry policy [Info](#)
A retry policy determines the maximum number of hours and number of times to retry sending an event to a target after an error occurs.

In the input path, create variables that will be used in the message. Here variables are created for account, instance id, region, etc.

Target input transformer

You can customize the text from an event before EventBridge passes the event to the target of a rule. Using the input transformer in the console or the API, you define variables that use JSON path to reference values in the original event source. You can define up to 100 variables, assigning each a value from the input. Then you can use those variables in the Input Template as <variable-name>.

Input path

The Input Path defined as key-value pairs is used to define variables. You use JSON path to reference items in your event and store those values in variables. For instance, you could create an Input Path to reference values in the event.

```
1 {"account": "$.account", "instance-id": "$.detail.instance-id", "region": "$.region", "state": "$.det,"
```

Define the message in the “template” section. Here it is visible that the variables are used in the message.

Template

The Input Template is a template for the information you want to pass to your target. You can create a template that passes either a string or JSON to the target.

```
1 At <time>, the status of your EC2 instance <instance-id> on account <account> in the AWS Region <AWS Region>
```

Select event bus

Event bus

Select or enter event bus name

default

Rules (1)

Name	Status	Type	Description
Test-EC2-termination	Enabled	Standard	-

Step 5 - Check if the rule sends an alert, manually terminate one of the EC2 instances in the auto scaling group.

The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being run against the AWS CloudWatch Metrics namespace. The results are displayed in a table format. One row of the table is highlighted in yellow, indicating it is the current item selected. This row corresponds to the termination of an EC2 instance. The table includes columns for Metric Name, Metric Value, and the timestamp of the metric. The timestamp for the highlighted row is 2023-07-29T13:54:37Z.

The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being run against the AWS CloudWatch Metrics namespace. The results are displayed in a table format. One row of the table is highlighted in yellow, indicating it is the current item selected. This row corresponds to the termination of an EC2 instance. The table includes columns for Metric Name, Metric Value, and the timestamp of the metric. The timestamp for the highlighted row is 2023-07-29T13:54:37Z.

Received alert on for both the state changes, shutting down and terminated.

The screenshot shows an email from AWS Notifications. The subject is "AWS Notifications <no-reply@sns.amazonaws.com> to me". The email body contains a message about the status change of an EC2 instance. It includes a link to unsubscribe from the topic and a support contact link.

The screenshot shows an email from AWS Notifications. The subject is "AWS Notifications <no-reply@sns.amazonaws.com> to me". The email body contains a message about the status change of an EC2 instance. It includes a link to unsubscribe from the topic and a support contact link.

The screenshot shows an email from AWS Notifications. The subject is "AWS Notifications <no-reply@sns.amazonaws.com> to me". The email body contains a message about the status change of an EC2 instance. It includes a link to unsubscribe from the topic and a support contact link.

The screenshot shows an email from AWS Notifications. The subject is "AWS Notifications <no-reply@sns.amazonaws.com> to me". The email body contains a message about the status change of an EC2 instance. It includes a link to unsubscribe from the topic and a support contact link.

Assignment 5 - Launch EC2 instance using AWS CLI.

Step 1 - Login to the AWS CLI.

```
[cloudshell-user@ip-10-2-14-12 ~]$ aws sts get-caller-identity
{
    "UserId": "AIDATOMGSBSVAAKATBQ3F",
    "Account": "237042273450",
    "Arn": "arn:aws:iam::237042273450:user/IAM_user-ShreyasK"
}
[cloudshell-user@ip-10-2-14-12 ~]$ █
```

Step 2 - Create a key pair for the instance.

This command creates a new keypair “linuxKeyPair” and saves it to a file “linuxKeyPair.pem”.

```
[cloudshell-user@ip-10-4-164-203 ~]$ aws ec2 create-key-pair --key-name linuxKeyPair --query 'KeyMaterial' --output text > linuxKeyPair.pem
[cloudshell-user@ip-10-4-164-203 ~]$ ls
ec2-sk-SH.pem  linuxKeyPair.pem
[cloudshell-user@ip-10-4-164-203 ~]$ aws ec2 describe-key-pairs --key-name linuxKeyPair
{
    "KeyPairs": [
        {
            "KeyId": "key-0ecc1876ca4134ac0",
            "KeyFingerprint": "d9:e1:f7:f1:d0:c6:3e:97:cb:9c:f6:7d:52:b2:fd:1b:02:86:b2:1c",
            "KeyName": "linuxKeyPair",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2023-07-29T16:48:34.053000+00:00"
        }
    ]
}
```

Step 3 - Create a security group for the instance.

This command creates a security group “Instance-SG” in the default VPC.

```
[cloudshell-user@ip-10-4-164-203 ~]$ aws ec2 create-security-group --group-name Instance-SG --description "Allows SSH on port 22" --vpc-id vpc-0b7165e7709485a86
{
    "GroupId": "sg-080377966684077b1"
}
[cloudshell-user@ip-10-4-164-203 ~]$ aws ec2 describe-security-groups --group-ids sg-080377966684077b1
{
    "SecurityGroups": [
        {
            "Description": "Allows SSH on port 22",
            "GroupName": "Instance-SG",
            "IpPermissions": [],
            "OwnerId": "237042273450",
            "GroupId": "sg-080377966684077b1",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": [],
                    "UserIdGroupPairs": []
                }
            ],
            "VpcId": "vpc-0b7165e7709485a86"
        }
    ]
}
```

This command adds an incoming traffic rule to allow SSH on port 22 from anywhere.

```
[cloudshell-user@ip-10-4-38-209 ~]$ aws ec2 authorize-security-group-ingress --group-id sg-080377966684077b1 --protocol tcp --port 22 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-09171e7919ad89037",
      "GroupId": "sg-080377966684077b1",
      "GroupOwnerId": "237042273450",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
[cloudshell-user@ip-10-4-38-209 ~]$
```

```
[cloudshell-user@ip-10-4-38-209 ~]$ aws ec2 describe-security-groups --group-ids sg-080377966684077b1
{
  "SecurityGroups": [
    {
      "Description": "Allows SSH on port 22",
      "GroupName": "Instance-SG",
      "IpPermissions": [
        {
          "FromPort": 22,
          "IpProtocol": "tcp",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "ToPort": 22,
          "UserIdGroupPairs": []
        }
      ],
      "OwnerId": "237042273450",
      "GroupId": "sg-080377966684077b1",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "UserIdGroupPairs": []
        }
      ],
      "VpcId": "vpc-0b7165e7709485a86"
    }
  ]
}
```

Step 4 - Launch an EC2 instance.

This command launches an EC2 instance with AMI of amazon linux 2023, instance type of t2.micro, attaching the key pair and security group created earlier in the subnet in AZ “us-east-1a” of the default VPC.

```
[cloudshell-user@ip-10-6-186-151 ~]$ aws ec2 run-instances --image-id ami-0f34c5ae932a6f0e4 --count 1 --instance-type t2.micro
{
  "Groups": []
}

--key-name linuxKeyPair --security-group-ids sg-080377966684077b1 --subnet-id subnet-051f9be32ca056d18
```

```
{  
    "Groups": [],  
    "Instances": [  
        {  
            "AmiLaunchIndex": 0,  
            "ImageId": "ami-0f34c5ae932e6f0e4",  
            "InstanceId": "i-02a791218617c108d",  
            "InstanceType": "t2.micro",  
            "KeyName": "linuxKeyPair",  
            "LaunchTime": "2023-07-29T17:06:10+00:00",  
            "Monitoring": {  
                "State": "disabled"  
            },  
            "Placement": {  
                "AvailabilityZone": "us-east-1a",  
                "GroupName": "",  
                "Tenancy": "default"  
            },  
            "PrivateDnsName": "ip-172-31-3-207.ec2.internal",  
            "PrivateIpAddress": "172.31.3.207",  
            "ProductCodes": [],  
            "PublicDnsName": "",  
            "State": {  
                "Code": 0,  
                "Name": "pending"  
            },  
            "StateTransitionReason": "",  
            "SubnetId": "subnet-051f9be32ca056d18",  
            "VpcId": "vpc-0b7165e7709485a86",  
            "Architecture": "x86_64",  
            "BlockDeviceMappings": [],  
            "ClientToken": "107d962e-2b76-49e2-ba95-5939e93f0cf0",  
            "EbsOptimized": false,  
            "EnaSupport": true,  
            "Hypervisor": "xen",  
            "NetworkInterfaces": [  
                {  
                    "Attachment": {  
                        "AttachTime": "2023-07-29T17:06:10+00:00",  
                        "AttachmentId": "eni-attach-0ff9a2bfb051976a9",  
                        "DeleteOnTermination": true,  
                        "DeviceIndex": 0,  
                        "Status": "attaching",  
                        "NetworkCardIndex": 0  
                    },  
                    "Description": "",  
                    "Groups": [  
                        {  
                            "GroupName": "Instance-SG",  
                            "GroupId": "sg-080377966684077b1"  
                        }  
                    ],  
                    "Ipv6Addresses": [],  
                    "MacAddress": "02:cb:9e:4d:ab:21",  
                    "NetworkInterfaceId": "eni-0a503802a899b45dd",  
                    "OwnerId": "237042273450",  
                    "PrivateDnsName": "ip-172-31-3-207.ec2.internal",  
                    "PrivateIpAddress": "172.31.3.207",  
                    "PrivateIpAddresses": [  
                        {  
                            "Primary": true,  
                            "PrivateDnsName": "ip-172-31-3-207.ec2.internal",  
                            "PrivateIpAddress": "172.31.3.207"  
                        }  
                    ],  
                    "SourceDestCheck": true,  
                    "Status": "in-use",  
                    "SubnetId": "subnet-051f9be32ca056d18",  
                    "VpcId": "vpc-0b7165e7709485a86",  
                    "InterfaceType": "interface"  
                }  
            ]  
        }  
    ]  
}
```

```
"NetworkInterfaces": [  
    {  
        "Attachment": {  
            "AttachTime": "2023-07-29T17:06:10+00:00",  
            "AttachmentId": "eni-attach-0ff9a2bfb051976a9",  
            "DeleteOnTermination": true,  
            "DeviceIndex": 0,  
            "Status": "attaching",  
            "NetworkCardIndex": 0  
        },  
        "Description": "",  
        "Groups": [  
            {  
                "GroupName": "Instance-SG",  
                "GroupId": "sg-080377966684077b1"  
            }  
        ],  
        "Ipv6Addresses": [],  
        "MacAddress": "02:cb:9e:4d:ab:21",  
        "NetworkInterfaceId": "eni-0a503802a899b45dd",  
        "OwnerId": "237042273450",  
        "PrivateDnsName": "ip-172-31-3-207.ec2.internal",  
        "PrivateIpAddress": "172.31.3.207",  
        "PrivateIpAddresses": [  
            {  
                "Primary": true,  
                "PrivateDnsName": "ip-172-31-3-207.ec2.internal",  
                "PrivateIpAddress": "172.31.3.207"  
            }  
        ],  
        "SourceDestCheck": true,  
        "Status": "in-use",  
        "SubnetId": "subnet-051f9be32ca056d18",  
        "VpcId": "vpc-0b7165e7709485a86",  
        "InterfaceType": "interface"  
    }  
],  

```

Step 5 - Connect to the instance.

```
[cloudshell-user@ip-10-6-84-53 ~]$ chmod 400 linuxKeyPair.pem
[cloudshell-user@ip-10-6-84-53 ~]$ ssh -i "linuxKeyPair.pem" ec2-user@ec2-44-193-80-47.compute-1.amazonaws.com
^C
[cloudshell-user@ip-10-6-84-53 ~]$ ssh -i "linuxKeyPair.pem" ec2-user@ec2-44-193-80-47.compute-1.amazonaws.com
The authenticity of host 'ec2-44-193-80-47.compute-1.amazonaws.com (44.193.80.47)' can't be established.
ECDSA key fingerprint is SHA256:gopG2StAyZpHyuChy0WNy0MQBeaD9NERN2YAzezVVG.
ECDSA key fingerprint is MD5:70:df:cf:43:85:64:f4:35:d4:63:0f:59:d7:89:47:53.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-44-193-80-47.compute-1.amazonaws.com,44.193.80.47' (ECDSA) to the list of known hosts.
,      #
~\_\#\#\#          Amazon Linux 2023
~~ \_\#\#\#\\
~~   \#\#
~~   \#/   https://aws.amazon.com/linux/amazon-linux-2023
~~   V~'`-->
~~~  /`/`/
~~~  /`/`/
~/`/`/
[ec2-user@ip-172-31-3-207 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-3-207 ~]$ ]
```

Assignment 6 - Installing grafana and collecting EC2 metrics.

Step 1 - Create an IAM role with permissions to get and list CloudWatch metrics and describe instances. This role will be attached to the EC2 instance in which grafana will be configured, so that grafana can get metrics from AWS.

Select use case for EC2 as this role will be attached to the EC2 instance.

Select trusted entity [Info](#)

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.

Common use cases

EC2
Allows EC2 instances to call AWS services on your behalf.

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

Use cases for other AWS services:

Choose a service to view use case

[Cancel](#) [Next](#)

Create a policy, this policy allows to list metrics, get metric statistics, and get metric data from cloudwatch, and describe tags, instances and regions.

Policy editor

```
1 * {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Sid": "AllowReadingMetricsFromCloudWatch",
6             "Effect": "Allow",
7             "Action": [
8                 "cloudwatch>ListMetrics",
9                 "cloudwatch>GetMetricStatistics",
10                "cloudwatch>GetMetricData"
11            ],
12            "Resource": "*"
13        },
14        {
15            "Sid": "AllowReadingTagsInstancesRegionsFromEC2",
16            "Effect": "Allow",
17            "Action": [
18                "ec2>DescribeTags",
19                "ec2>DescribeInstances",
20                "ec2>DescribeRegions"
21            ],
22            "Resource": "*"
23        }
24    ]
25 }
```

Role name
Enter a meaningful name to identify this role.

GetMetrics_EC2Role

Maximum 64 characters. Use alphanumeric and '+=,.@-_` characters.

Description
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_` characters.

Step 1: Select trusted entities

```

1 ~ [{}]
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10         "Service": [
11           "ec2.amazonaws.com"
12         ]
13     }
14   }
15 ]
16 ]

```

Attached the previously created policy to the IAM role.

Step 2: Add permissions

Permissions policy summary		
Policy name	Type	Attached as
GetMetrics_Policy	Customer managed	Permissions policy

Step 2 - Launch an EC2 instance.

Name and tags [Info](#)

Name

Test-Grafana_Instance

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

[Search our full catalog including 1000s of application and OS images](#)

Recents **Quick Start**



[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI ami-0f34c5ae932e6f0e4 (64-bit (x86)) / ami-0964d1dc1edd4bd2f (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible
---	------------------------------------

Description

Amazon Linux 2023 AMI 2023.1.20230725.0 x86_64 HVM kernel-6.1

Architecture

▾

AMI ID

ami-0f34c5ae932e6f0e4

▼ Instance type [Info](#)

Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing:	0.0162 USD per Hour
On-Demand SUSE pricing:	0.0116 USD per Hour
On-Demand RHEL pricing:	0.0716 USD per Hour
On-Demand Linux pricing:	0.0116 USD per Hour

All generations

[Compare instance types](#)

▼ 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

▾

[Create new key pair](#)

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0b7165e7709485a86 172.31.0.0/16	(default) ▾	
--	-------------	--

Subnet Info

subnet-051f9be32ca056d18 VPC: vpc-0b7165e7709485a86 Owner: 237042273450 Availability Zone: us-east-1a IP addresses available: 4090 CIDR: 172.31.0.0/20	▼	Create new subnet
--	---	-------------------

Auto-assign public IP [Info](#)

Enable	▼
--------	---

The security group allows incoming traffic for SSH, HTTP, and custom TCP on port 3000 from anywhere.

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

Security group name - required

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and .,-:/()#@[]+=&;!\$^*

Description - required [Info](#)

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type Info ssh	Protocol Info TCP	Port range Info 22
Source type Info Anywhere	Source Info <input type="text" value="Add CIDR, prefix list or security"/>	Description - optional Info e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 80, Multiple sources)

Type Info HTTP	Protocol Info TCP	Port range Info 80
Source type Info Anywhere	Source Info <input type="text" value="Add CIDR, prefix list or security"/>	Description - optional Info e.g. SSH for admin desktop

Security group rule 3 (TCP, 3000, Multiple sources)

Type [Info](#) Protocol [Info](#) Port range [Info](#)

Custom TCP TCP 3000

Source type [Info](#) Source [Info](#) Description - optional [Info](#)

Anywhere Add CIDR, prefix list or security e.g. SSH for admin desktop

0.0.0.0/0 X ::/0 X

Remove

Attaching the role created in the previous step.

IAM instance profile [Info](#)

GetMetrics_EC2Role arn:aws:iam::237042273450:instance-profile/GetMetrics_EC2Role

Create new IAM profile

Step 3 - Install grafana in the EC2 instance.

Create a repository for grafana before installing it.

```
[root@ip-172-31-14-3 ec2-user]# vi /etc/yum.repos.d/grafana.repo
```

```
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
~
~
```

```
[root@ip-172-31-14-3 ec2-user]# yum install grafana -y
grafana
grafana
Importing GPG key 0x2CF3C0C6:
Userid      : "Grafana Labs <engineering@grafana.com>"
Fingerprint: 0E22 EB88 E39E 1227 7A77 60AE 9E43 9B10 2CF3 C0C6
From        : https://packages.grafana.com/gpg.key
grafana
Last metadata expiration check: 0:00:08 ago on Sun Jul 30 10:33:30 2023.
```

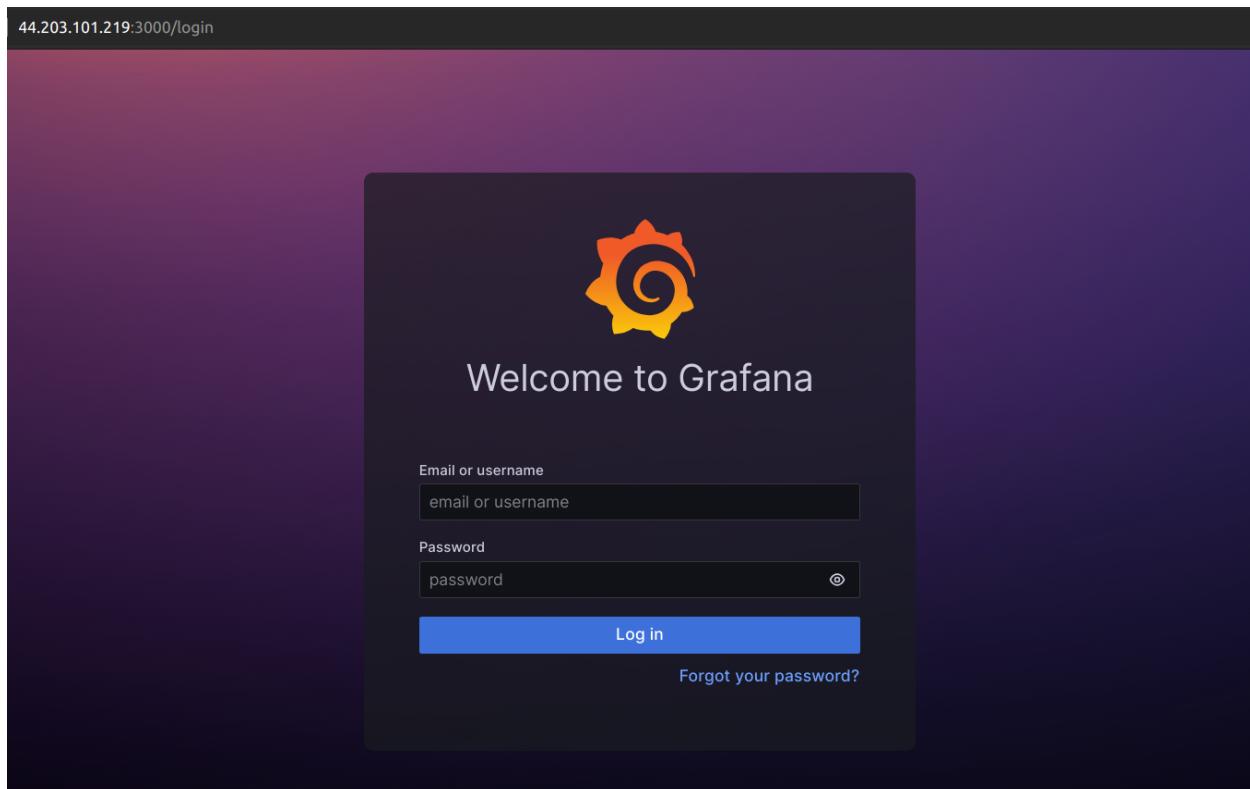
Start the grafana service.

```
[root@ip-172-31-14-3 ec2-user]# systemctl daemon-reload
[root@ip-172-31-14-3 ec2-user]# systemctl start grafana-server
[root@ip-172-31-14-3 ec2-user]# systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; preset: disabled)
   Active: active (running) since Sun 2023-07-30 10:44:51 UTC; 13s ago
     Docs: http://docs.grafana.org
 Main PID: 26686 (grafana)
    Tasks: 9 (limit: 1114)
   Memory: 147.7M
      CPU: 3.331s
     CGroup: /system.slice/grafana-server.service
             └─26686 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/var/r
```

Enable the grafana service so that it will start automatically whenever the instance is restarted.

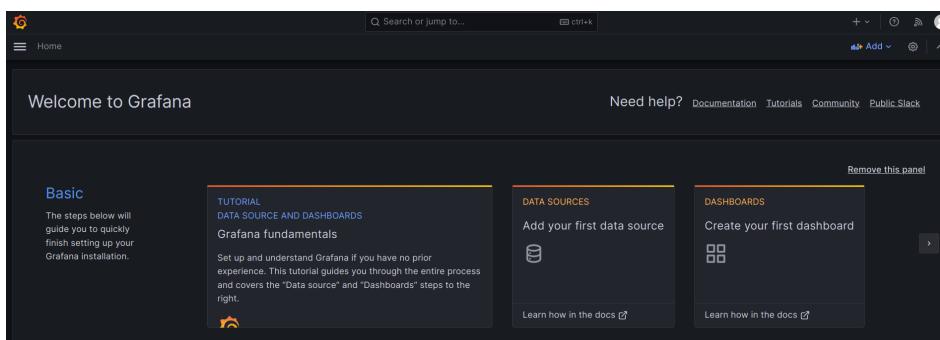
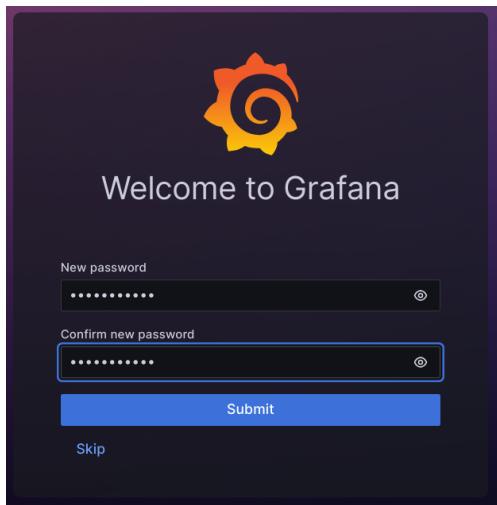
```
[root@ip-172-31-14-3 ec2-user]# systemctl enable grafana-server.service
Synchronizing state of grafana-server.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /usr/lib/systemd/system/grafana-server.service.
```

Enter the public IPv4 address of the instance with the custom port number 3000 - “44.203.101.219:3000”



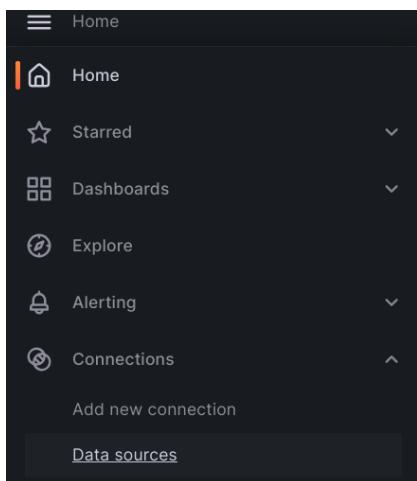
Step 4 - Configure grafana to collect metrics from AWS Cloudwatch.

Login to grafana with default username and password (admin). Then reset the password.



Step 5 - Connect grafana to AWS CloudWatch.

In the left navigation bar, go to the data sources section and choose CloudWatch as the data source.



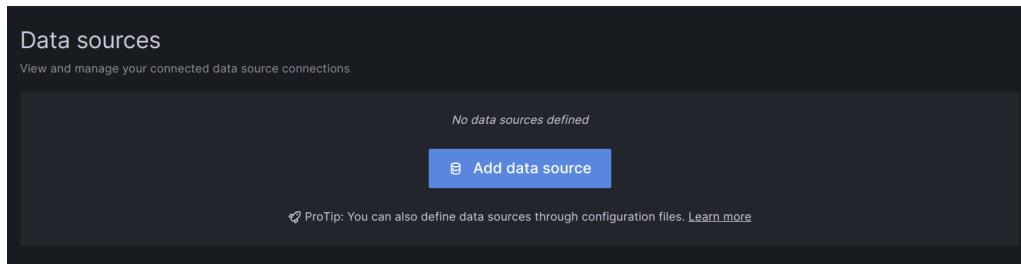
Data sources

View and manage your connected data source connections

No data sources defined

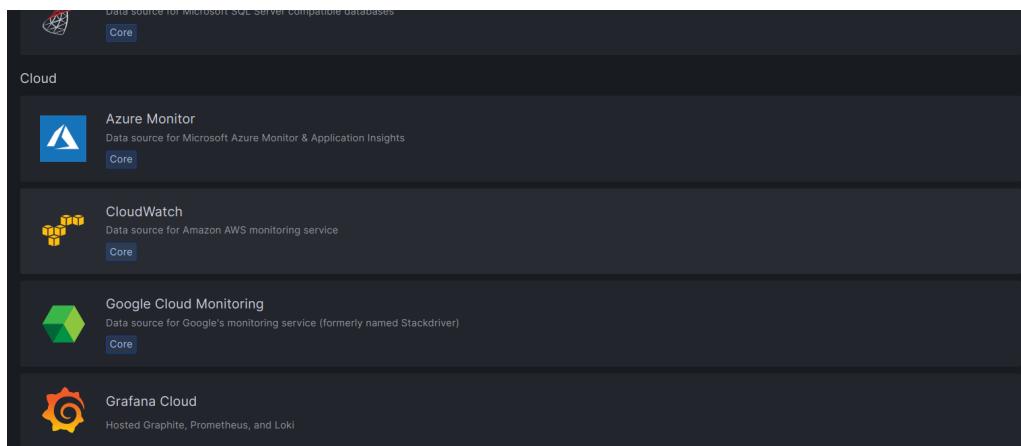
Add data source

ProTip: You can also define data sources through configuration files. [Learn more](#)



Cloud

- Azure Monitor
Data source for Microsoft Azure Monitor & Application Insights
Core
- CloudWatch
Data source for Amazon AWS monitoring service
Core
- Google Cloud Monitoring
Data source for Google's monitoring service (formerly named Stackdriver)
Core
- Grafana Cloud
Hosted Graphite, Prometheus, and Loki



Step 6 - Configure the connection details.

Enter the default region. Here it is “us-east-1”.

CloudWatch

Type: CloudWatch

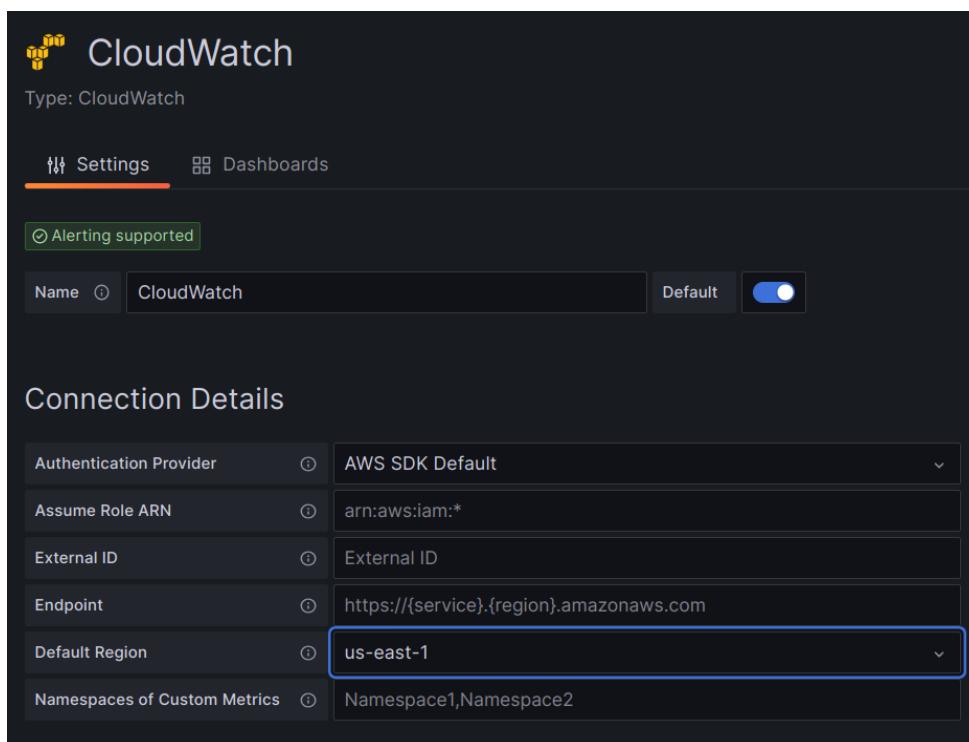
Settings Dashboards

⌚ Alerting supported

Name: CloudWatch Default:

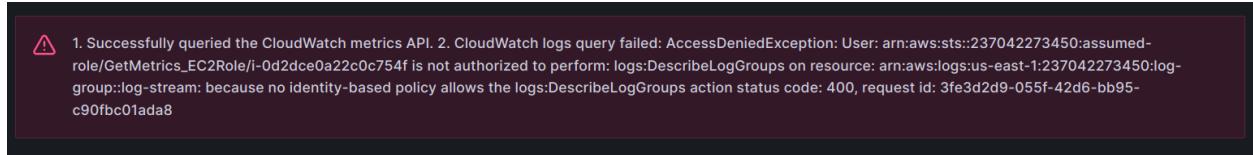
Connection Details

Authentication Provider	<input type="text"/> AWS SDK Default
Assume Role ARN	<input type="text"/> arn:aws:iam:*
External ID	<input type="text"/> External ID
Endpoint	<input type="text"/> https://(service).(region).amazonaws.com
Default Region	<input type="text"/> us-east-1
Namespaces of Custom Metrics	<input type="text"/> Namespace1, Namespace2

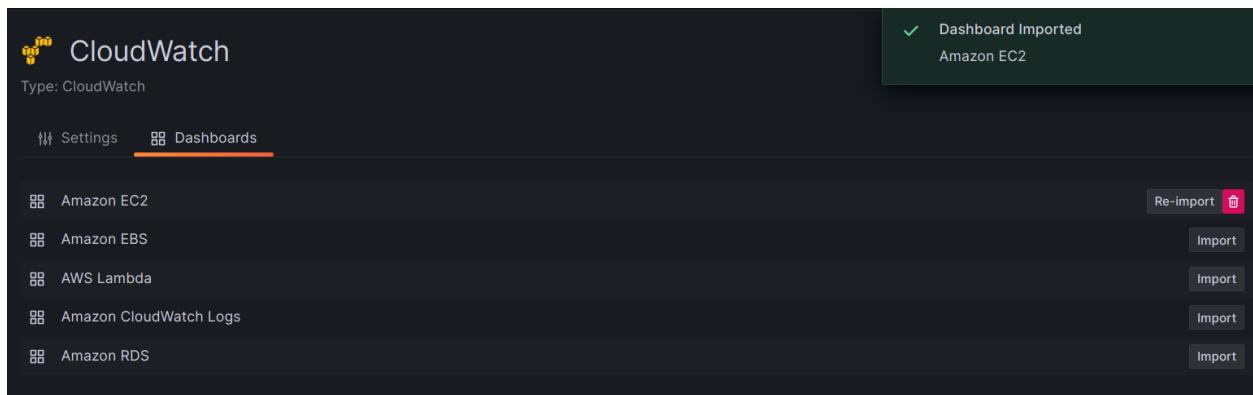


Click on “Test & Save”.

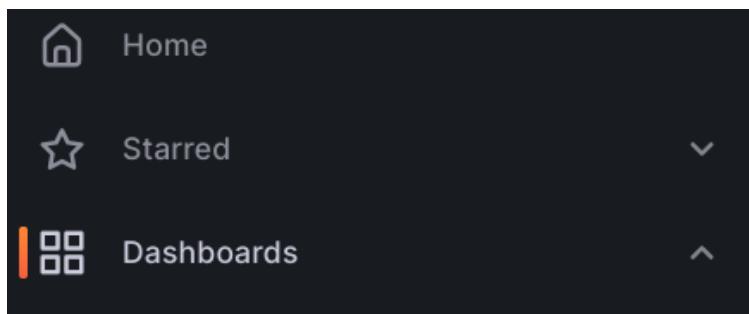
The message shows that metrics are successfully queried but shows error for querying the logs. This is because the IAM role attached to the EC2 instance in which grafana is install, has permissions for only to collect metrics but not the logs.



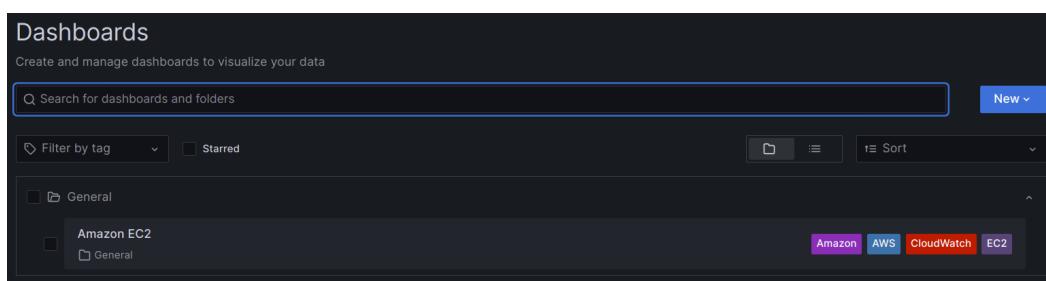
Go to the Dashboards section. Click on import for the service for the which the metrics are required. Here only EC2 metrics are imported.



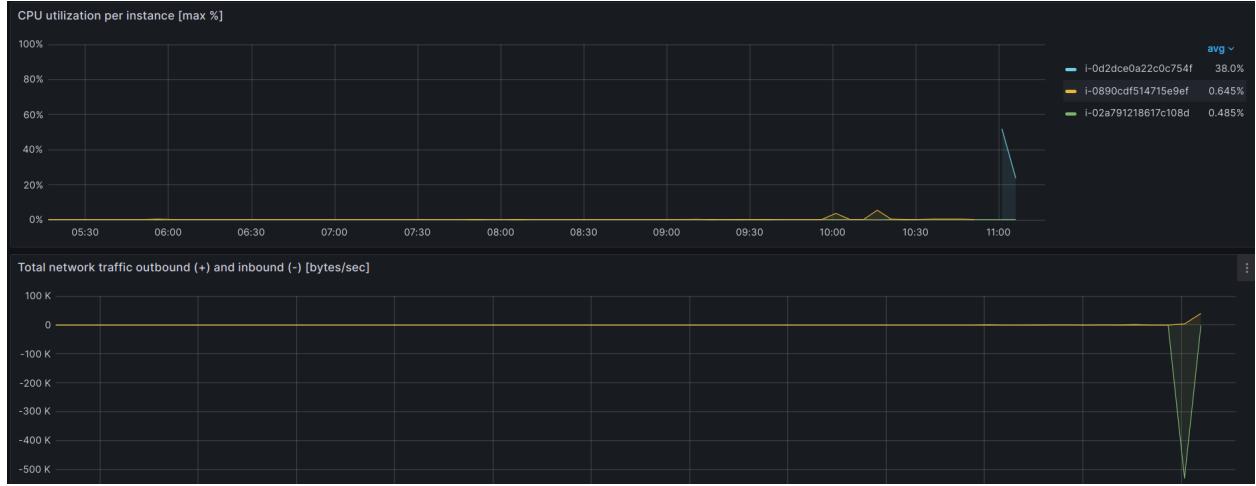
Click on the Dashboard button in the left navigation bar.



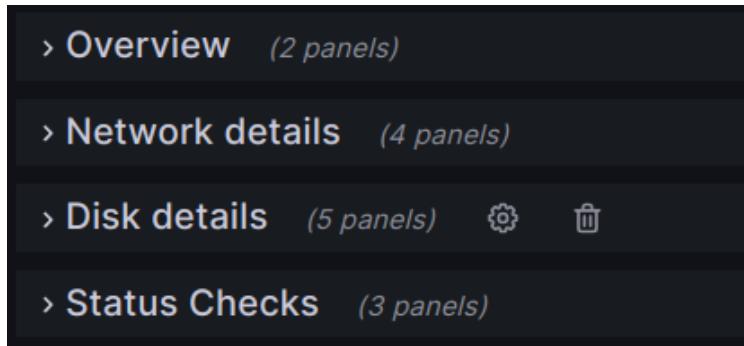
Click on “General” and select the service name. Here EC2.



Here the graphs for CPU utilization and other metrics from the EC2 instances running in the default selected region are displayed.



Here we can see the default metrics published by EC2 instances.



Here we can configure the graphs according to the requirements.

