**Automatically Start Stop EC2 Instances with AWS Lambda and Amazon Event Bridge**

*The Instance Scheduler on AWS solution automates the starting and stopping of various AWS services including Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) instances.*

*This solution helps reduce operational costs by stopping resources that are not in use and starting resources when their capacity is needed. For example, a company can use Instance Scheduler on AWS to automatically stop instances outside of business hours every day. If you leave all of your instances running at full utilization, this solution can result in up to 70% cost savings for those instances that are only necessary during regular business hours (weekly utilization reduced from 168 hours to 50 hours).*

*Instance Scheduler on AWS leverages Amazon Web Services (AWS) resource* ***tags*** *and*[*AWS Lambda*](https://aws.amazon.com/lambda/)*to automatically stop and restart instances across multiple AWS Regions and accounts on a customer-defined schedule.*

*There are multiple ways to achieve this. Either by using Built in Stack templates on AWS which also helps to configure the required roles such as IAM role to access the Lamda function automatically or by configuring them all manually (mentioned below)*

**AWS Well-Architected design considerations:**

*We designed this solution with best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.*

**Operational excellence:**

* *The solution pushes metrics to Amazon CloudWatch to provide observability into its components (such as its infrastructure and Lambda functions).*
* *AWS X-Ray traces Lambda functions.*
* *Uses Amazon SNS for error reporting.*

**Security:**

* *All inter-service communications use IAM roles.*
* *All multi-account communications use IAM roles.*
* *All roles used by the solution follow least-privilege access. In other words, they only contain minimum permissions required so that the service can function properly.*
* *All data storage including DynamoDB tables have encryption at rest.*

**Reliability:**

* *The solution uses serverless AWS services wherever possible (such as Lambda and DynamoDB) to ensure high availability and recovery from service failure.*
* *Data processing uses Lambda functions. The solution stores data in DynamoDB, so it persists in multiple Availability Zones by default.*

**Performance efficiency:**

* *The solution uses serverless architecture.*
* *You can launch the solution in any AWS Region that supports the AWS services used in this solution (such as Lambda and DynamoDB). For details, refer to*[*Supported AWS Regions*](https://docs.aws.amazon.com/solutions/latest/instance-scheduler-on-aws/supported-aws-regions)*.*
* *The solution is automatically tested and deployed every day. Our solution architects and subject matter experts review the solution for areas to experiment and improve.*

**Use Case:**

**UAT and Dev Teams where the phase lasts may be only for few months.**

**The team only works for limited hours in a day.**

**Rest of the week the resources are not required and hence can be shut down.**

**Underlying Services Used:**

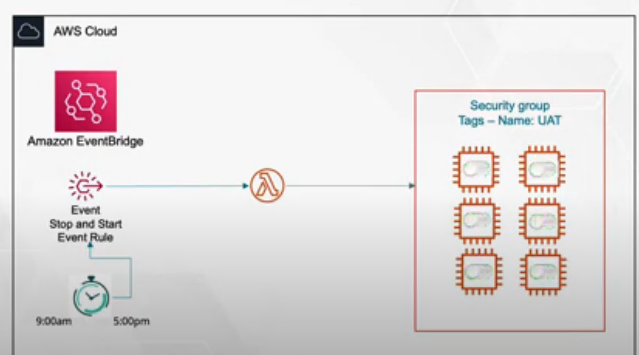
* *Amazone EventBridge*
* *Lamda Function*
* *IAM Role*
* *Dynamo DB*

*Start the VM’s at a particular Time schedule*

A computer screen shot of a computer security system

Description automatically generated

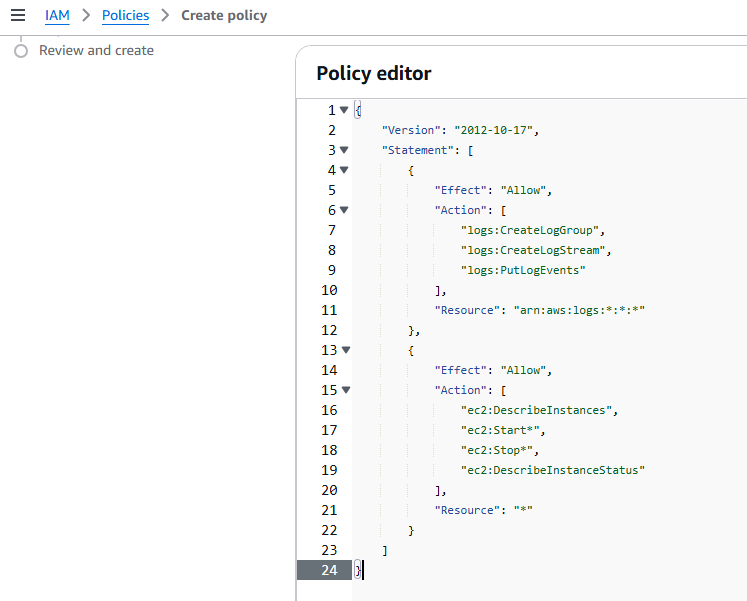
*Stop the VM’s at a particular Time schedule*

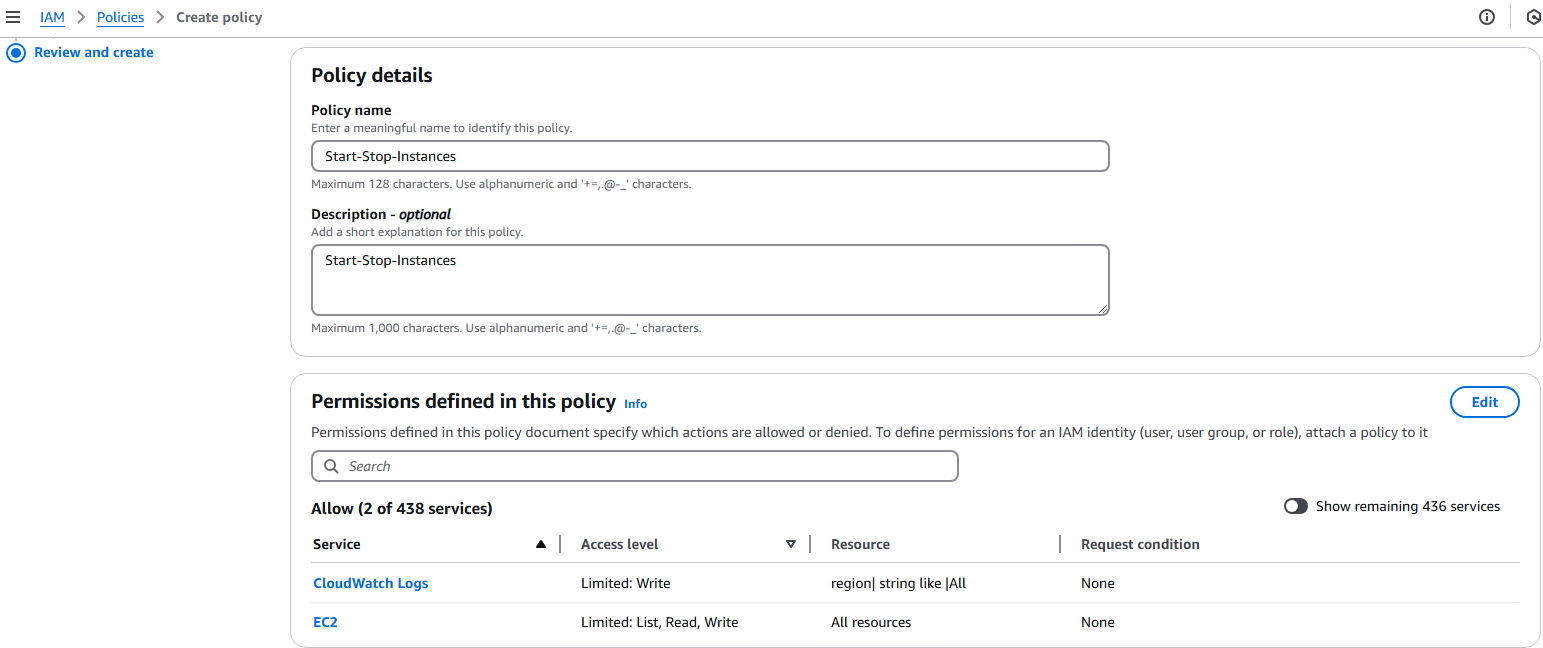


**Steps Involved:**

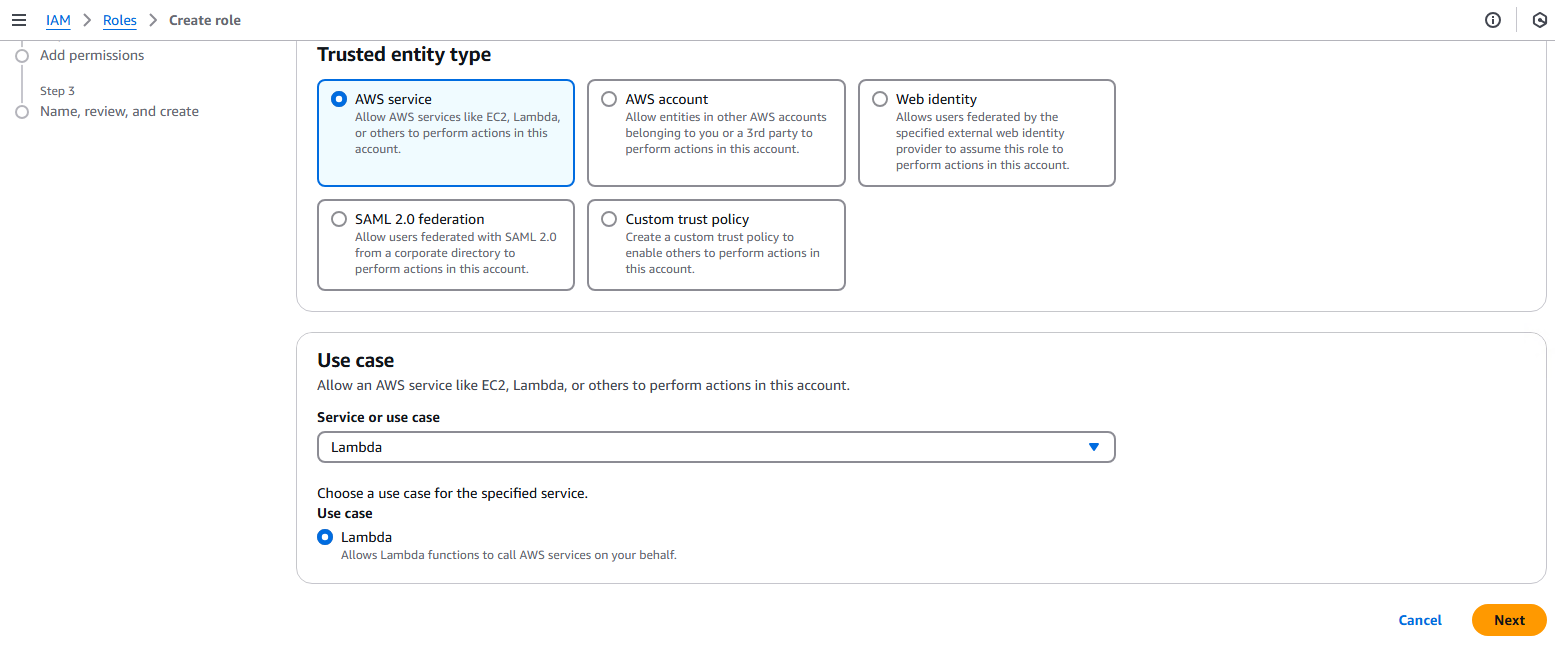


**Create an IAM policy and role for your Lambda function:**

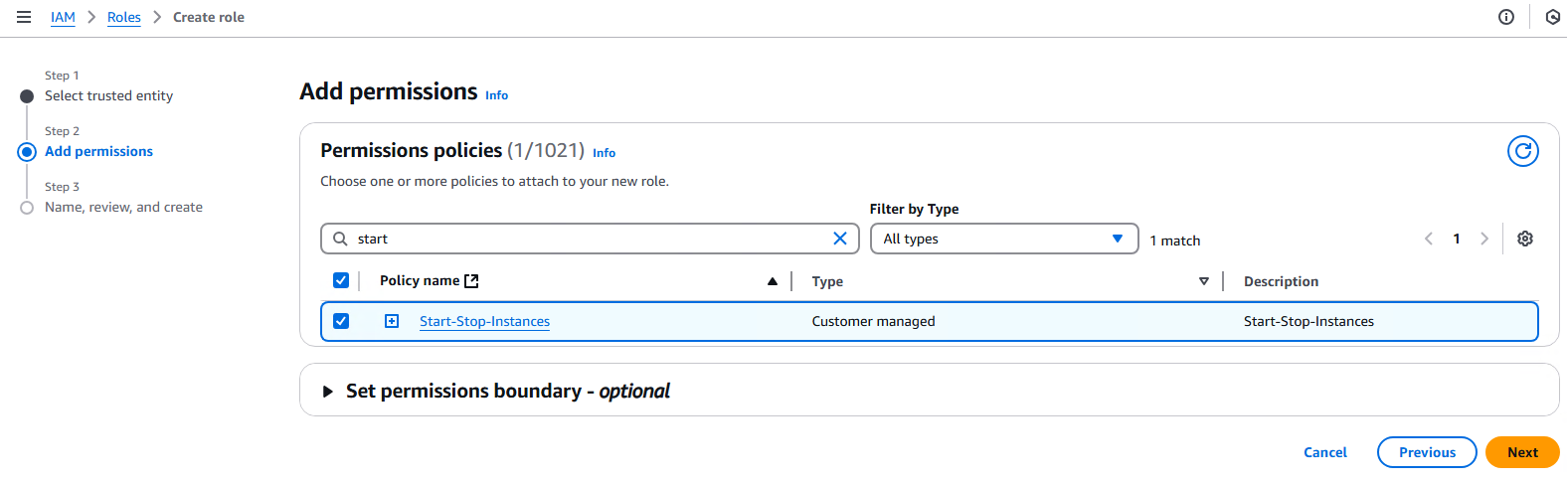




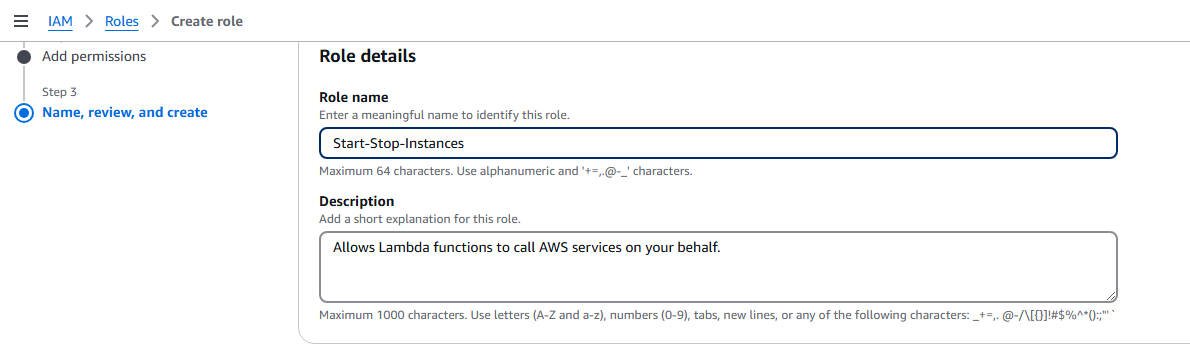
[**Create an IAM role**](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html#roles-creatingrole-service-console)**for Lambda.**

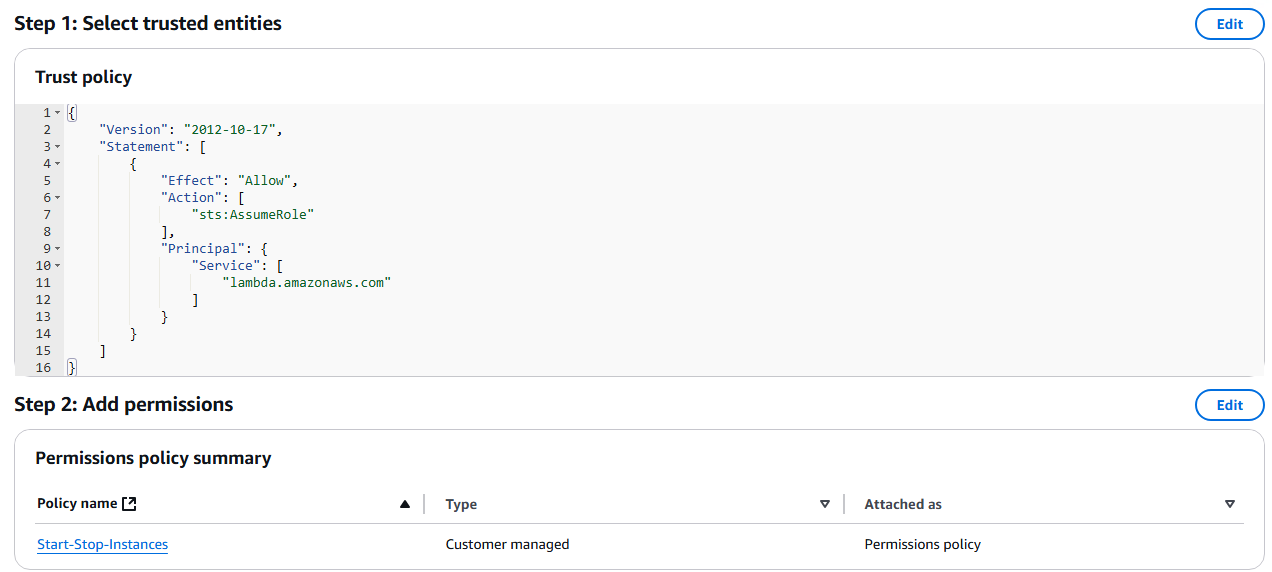


**Add the Policy Created in the above step to the Role:**

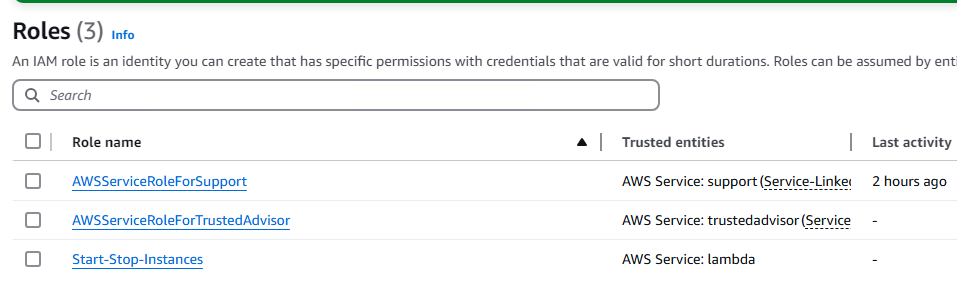


**Provide a Name to the Role:**



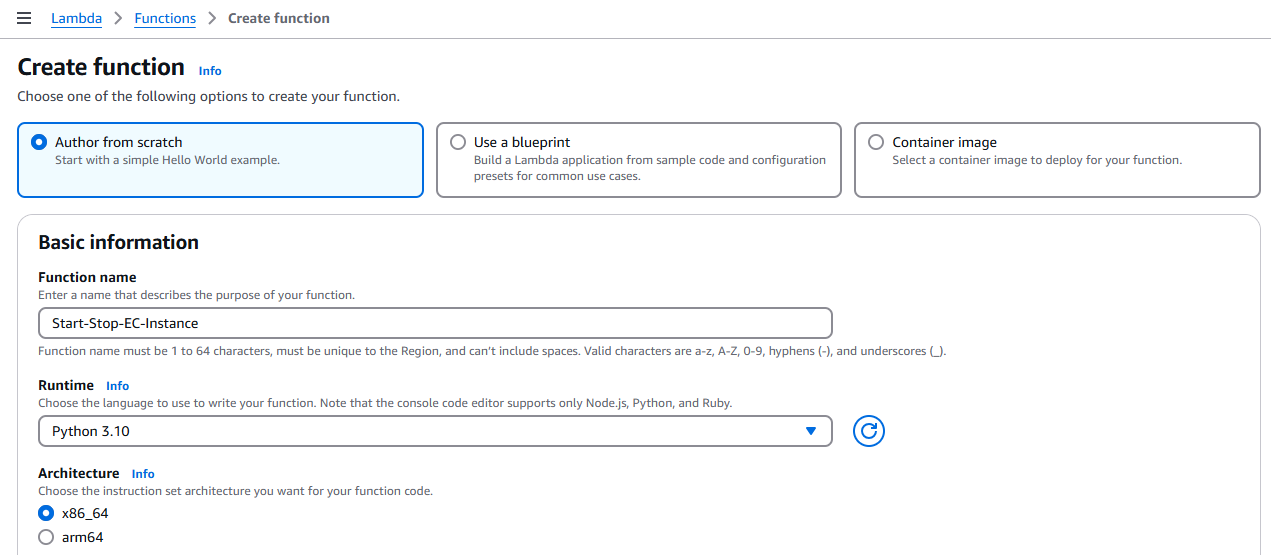


**Create the Role:**

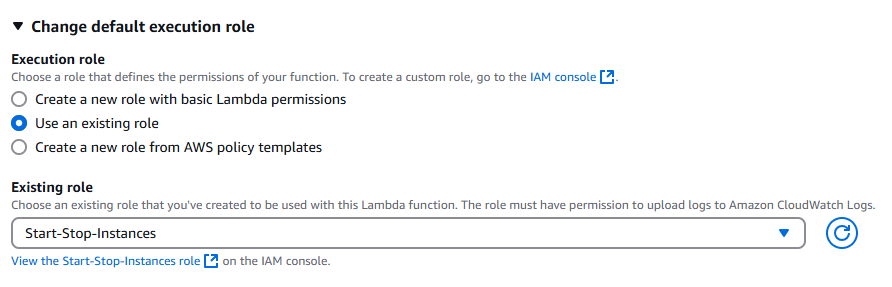


**Create Lambda functions that stop and start your instances:**

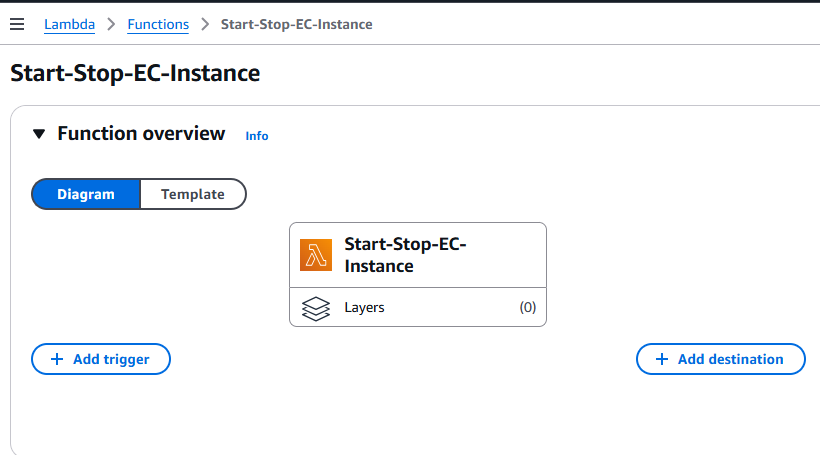
* 1. **Open the**[**Lambda console**](https://console.aws.amazon.com/lambda/)**, and then choose Create function.** Choose **Author from scratch and provide a Name**
  2. **Select runtime as Python**



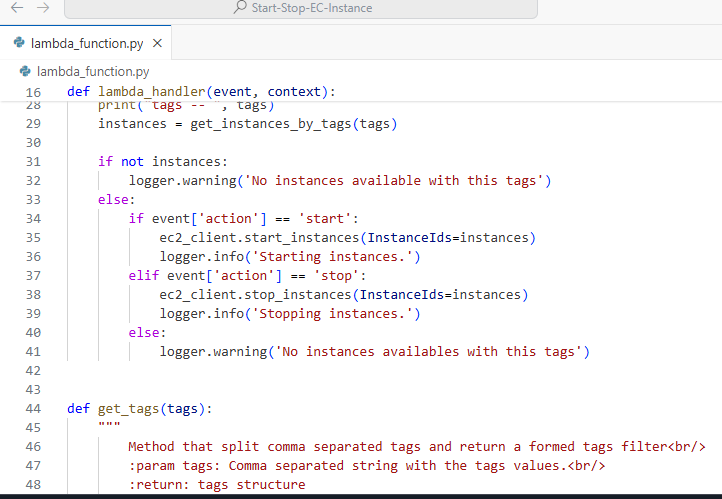
Under **Permissions**, expand **Change default execution role**.  
Under **Execution role**, choose **Use an existing role**.  
Under **Existing role**, choose the IAM role.



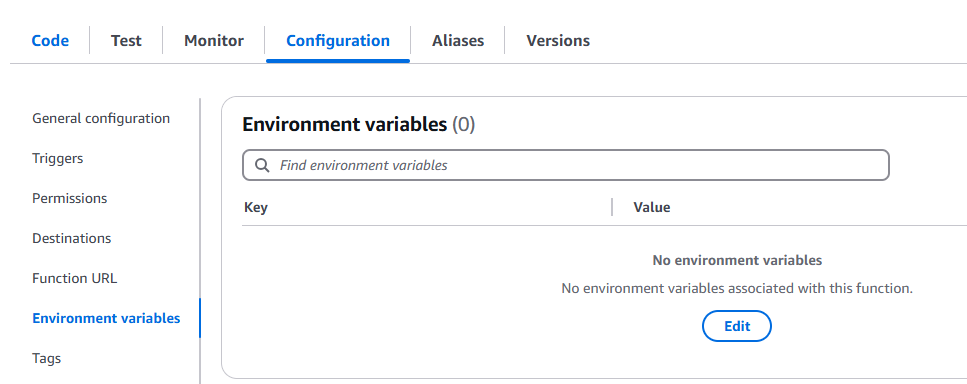
* 1. Choose **Create function**.



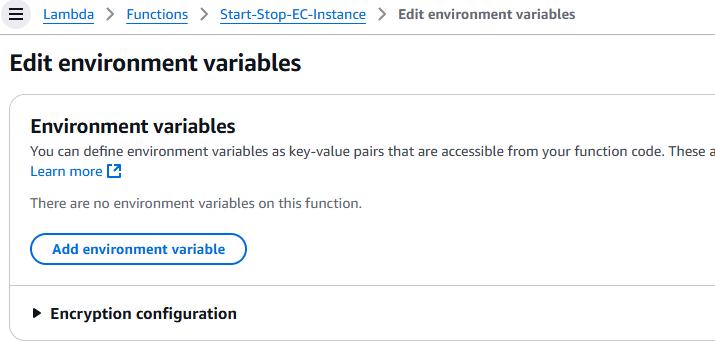
* 1. Choose the **Code** tab and update the JSON code to Stop the Instance and deploy the code.



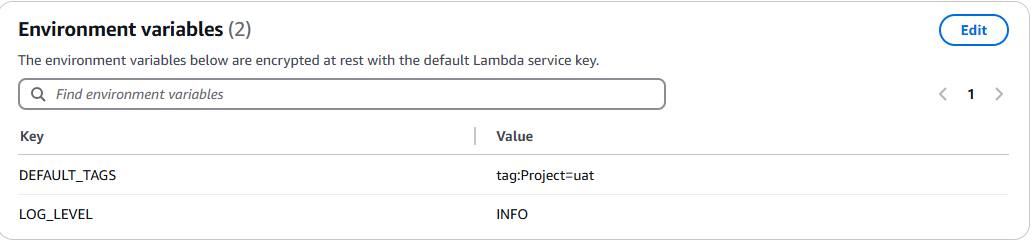
* 1. **Under Configuration click on environmental Variables**



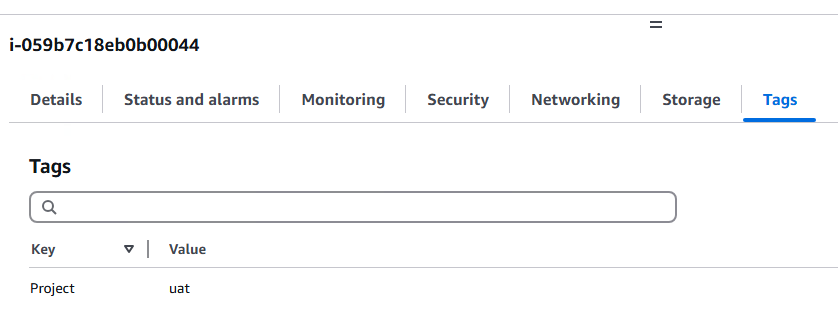
**Click on Edit and Add Environment Variables.**

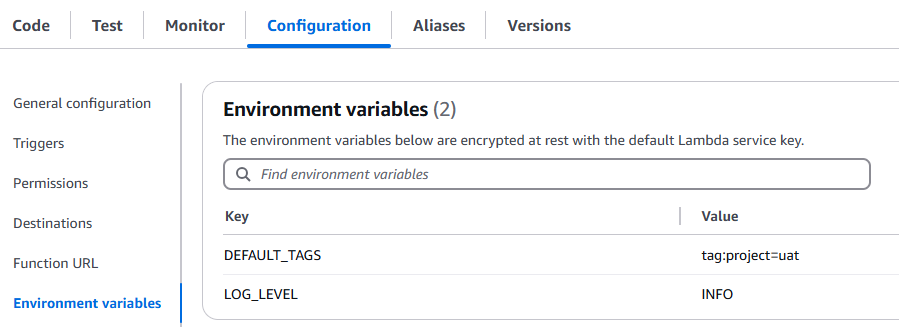


**NOTE: The details entered in the Variables is case sensitive**

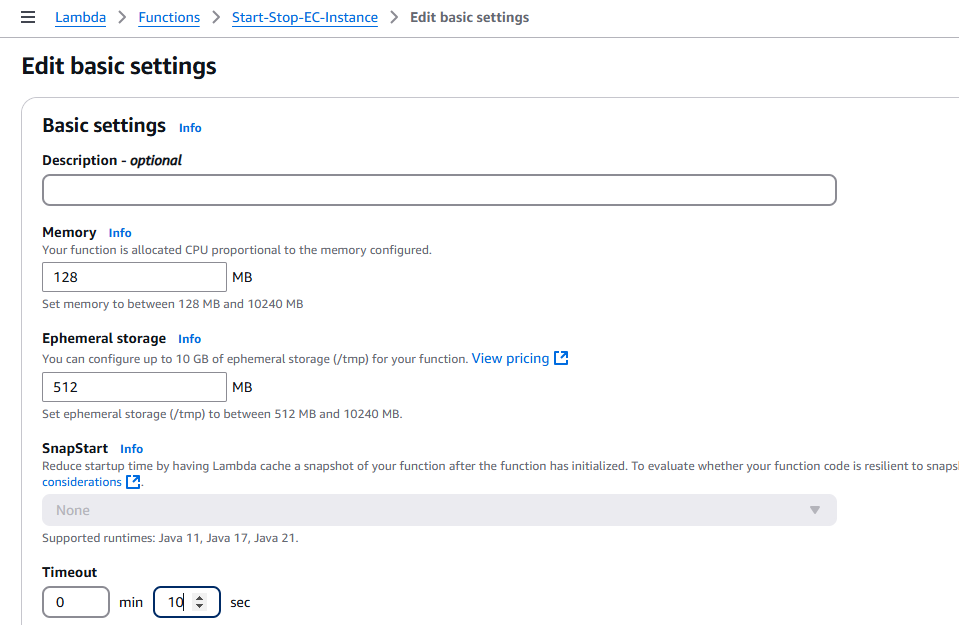


**NOTE: Tags details from the VM**

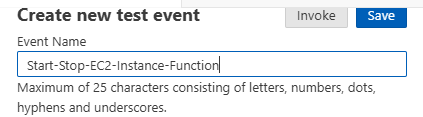




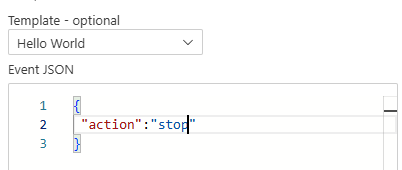
**Under General Configuration Set the Time Out value to 10 Secs.**



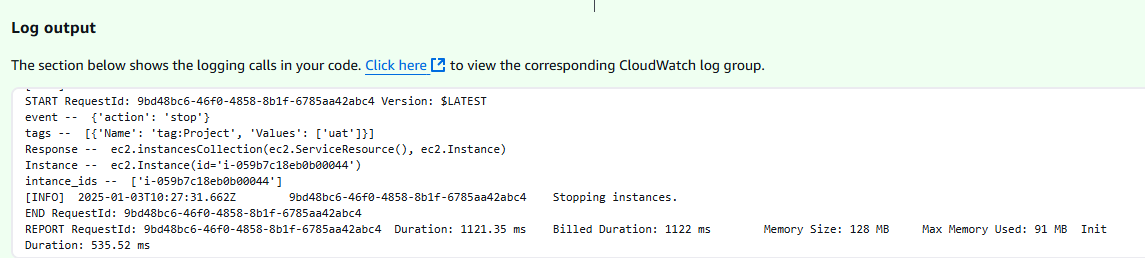
**Create a Test case for Lamda Function**



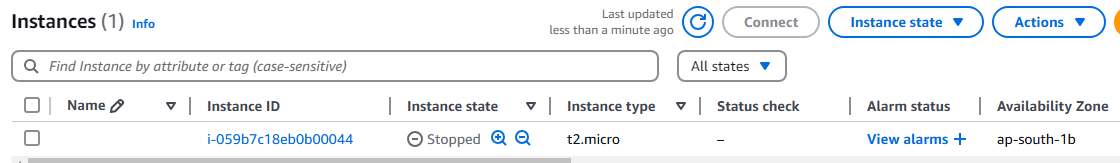
**Update the JSON Code to Stop the Instance.**



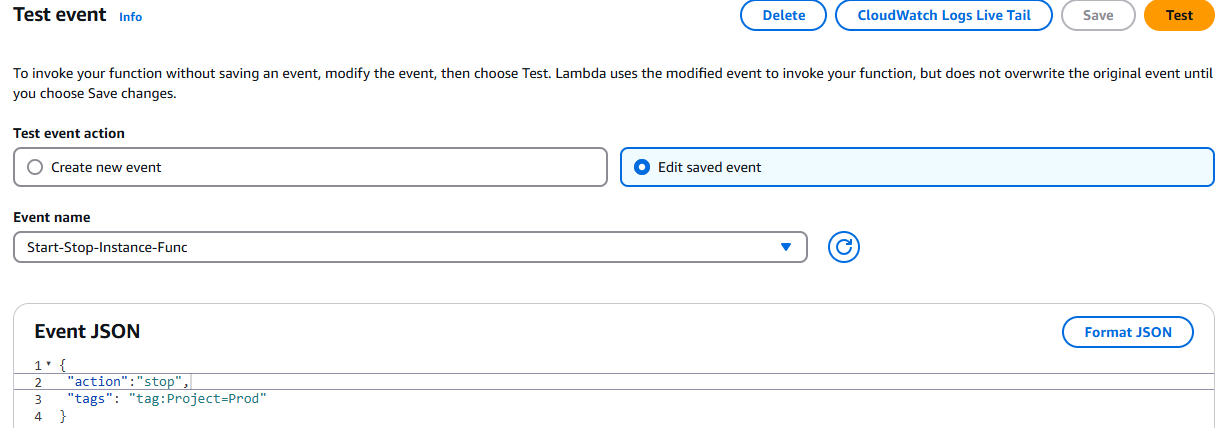
**Test the function and verify the logs**



**The instance has been stopped**

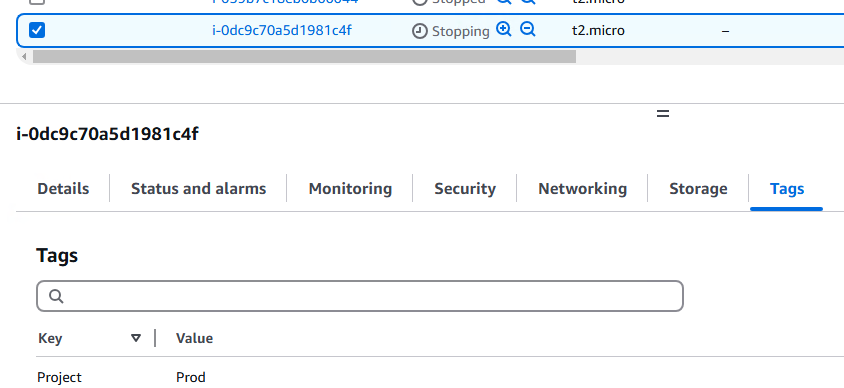


**The same Lamda function can be used to Start or Stop instances with other tags. Just update the JSON with the appropriate code.**

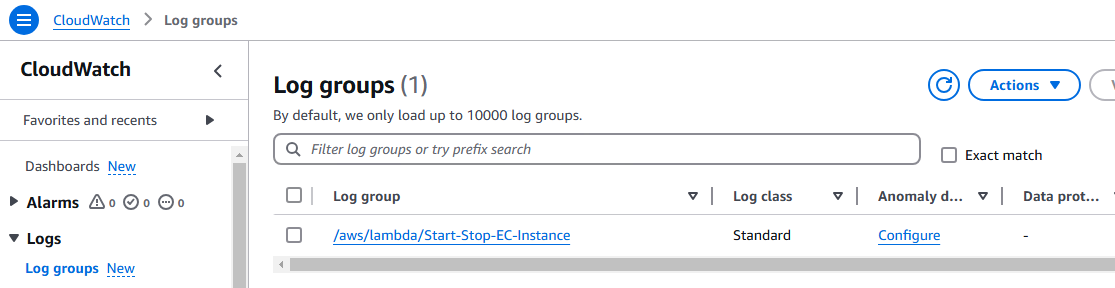


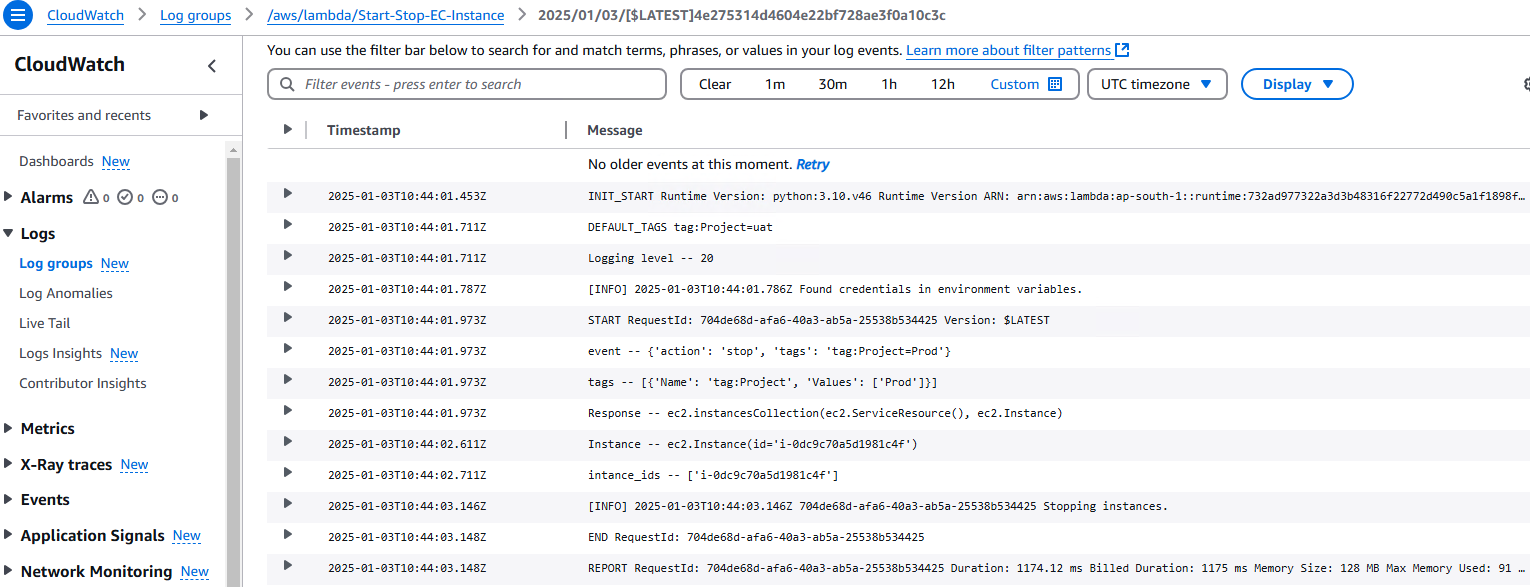
A screenshot of a computer

Description automatically generated

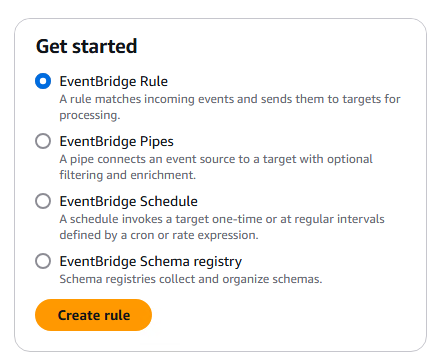


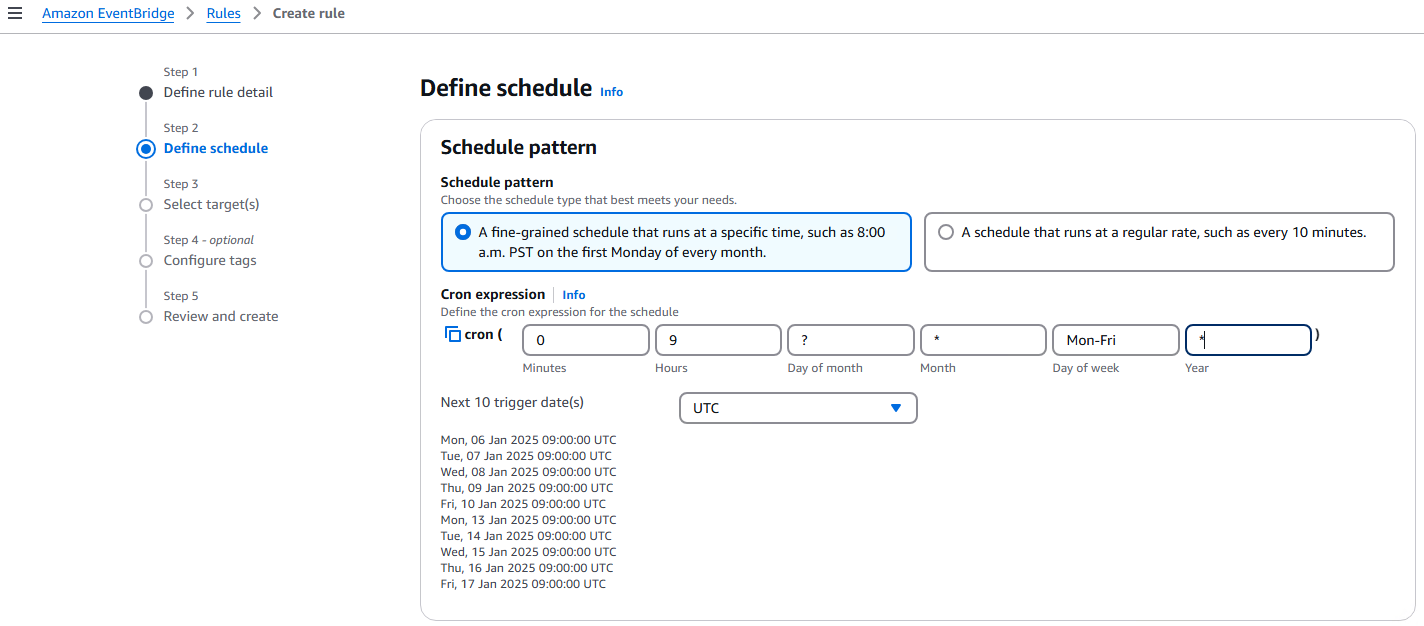
**The logs can be seen under CloudWatch**



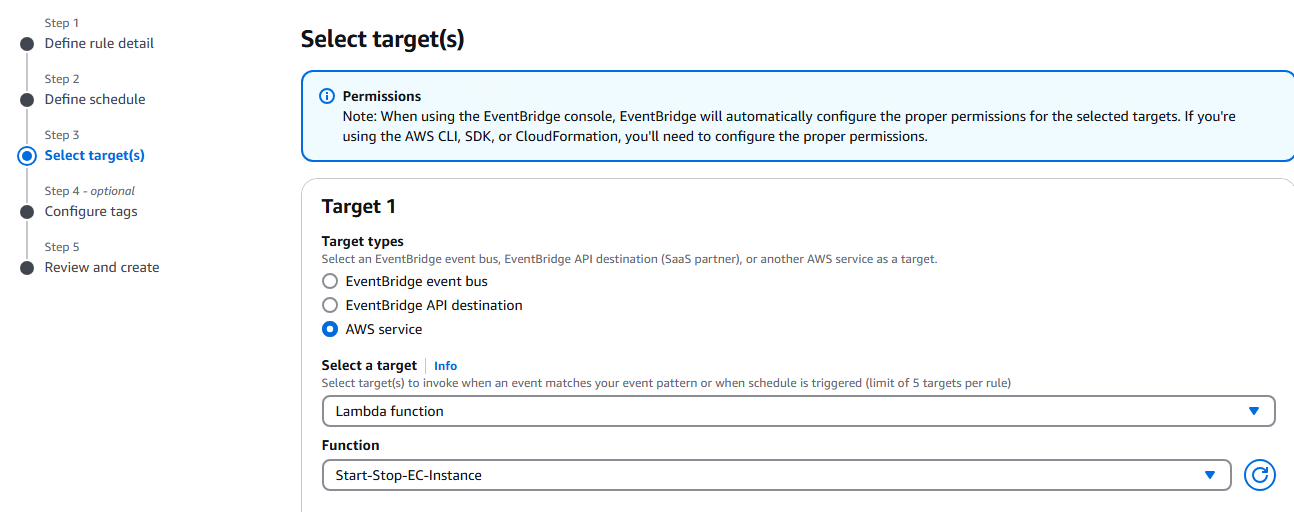


**Configure Rule on EventBridge.**

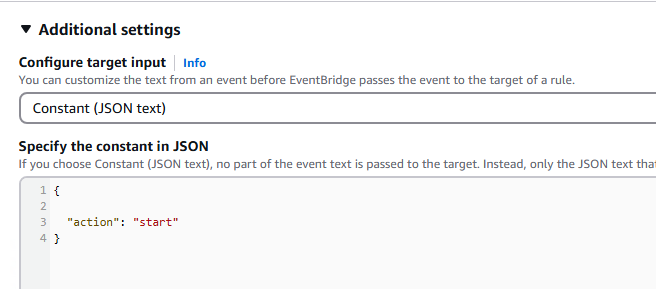




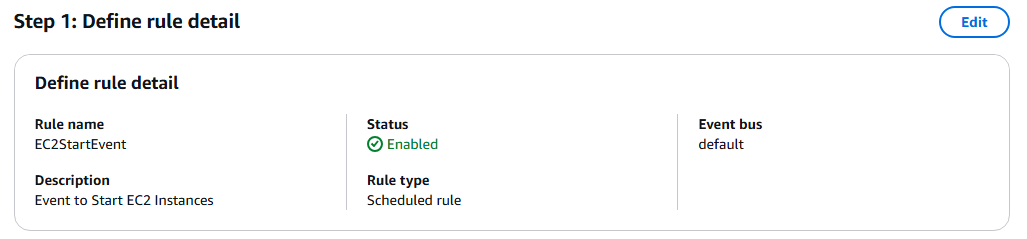
**Select Target as Lamda Function which we have created.**

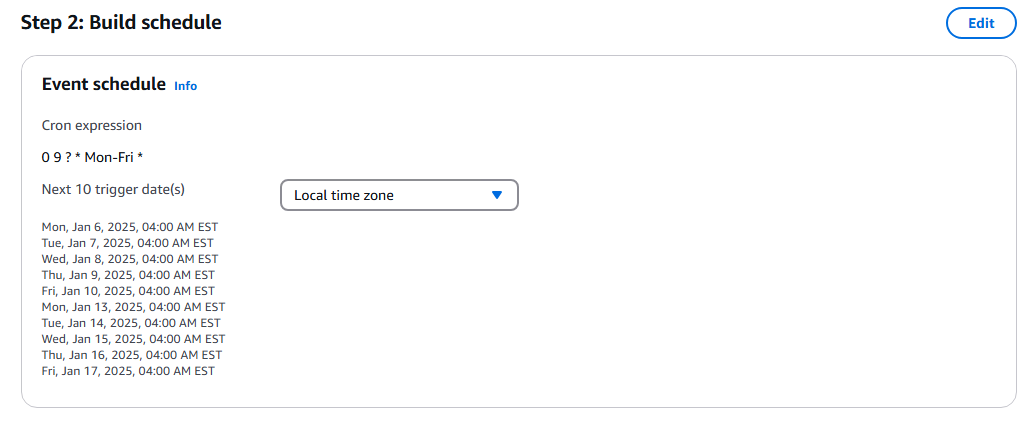


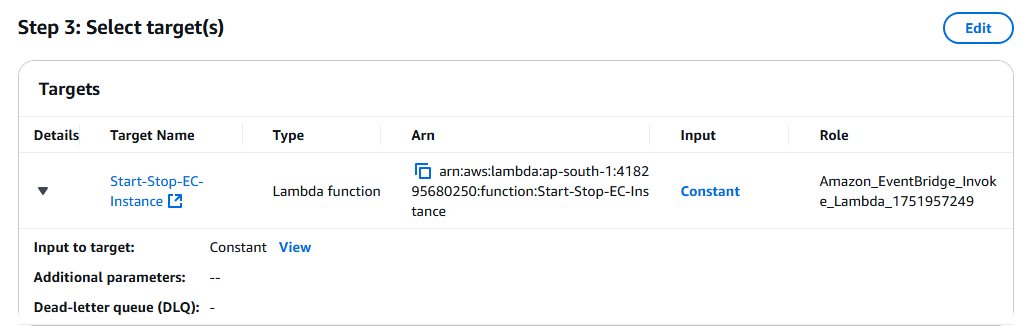
**Configure Additional settings**

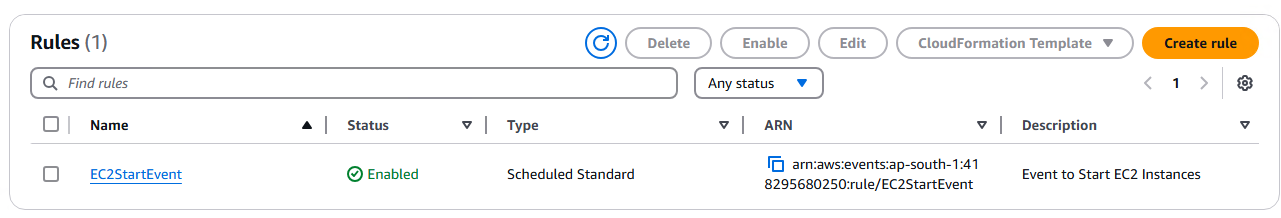


**Review and Create**

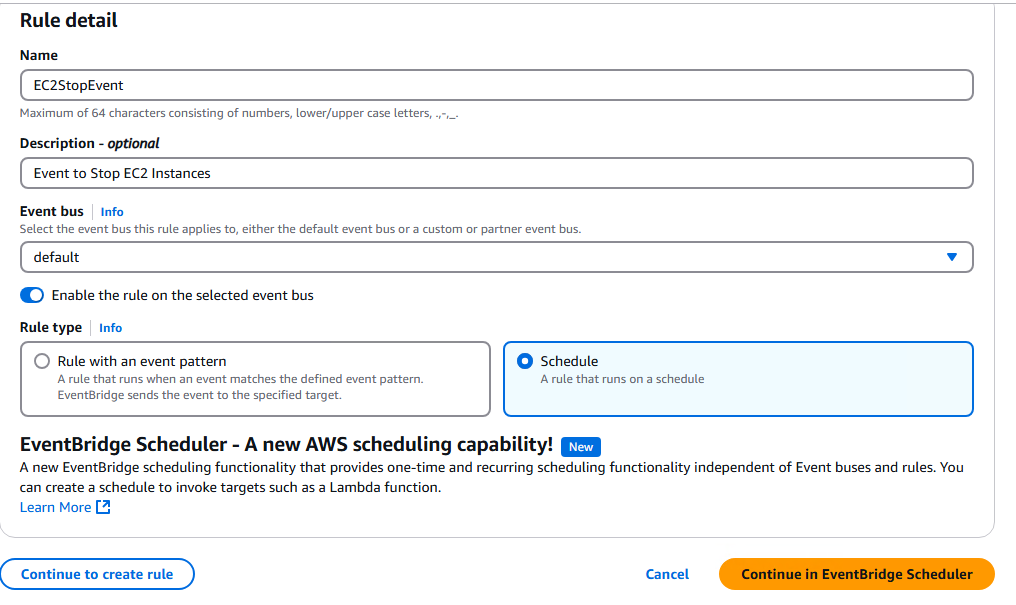


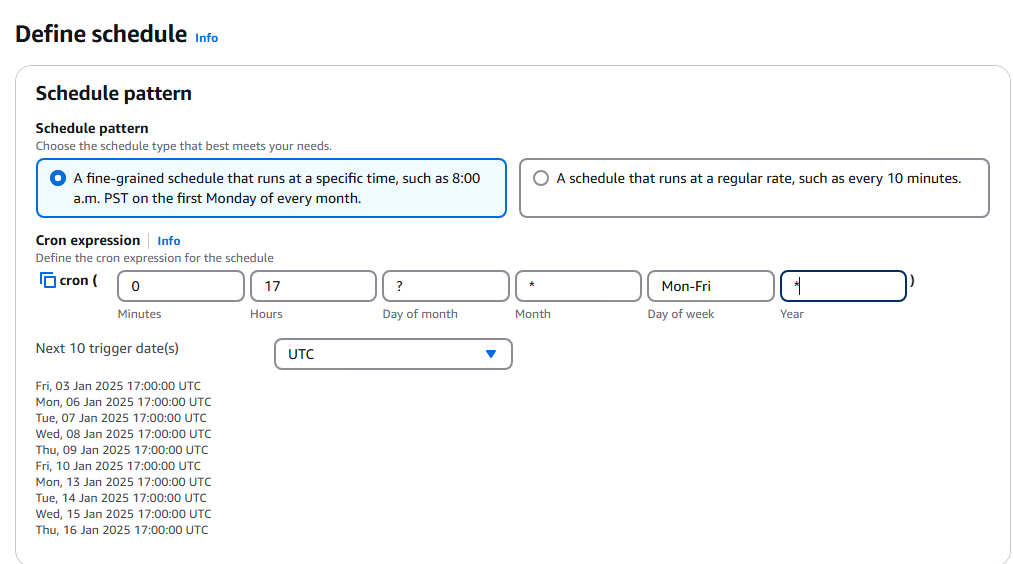


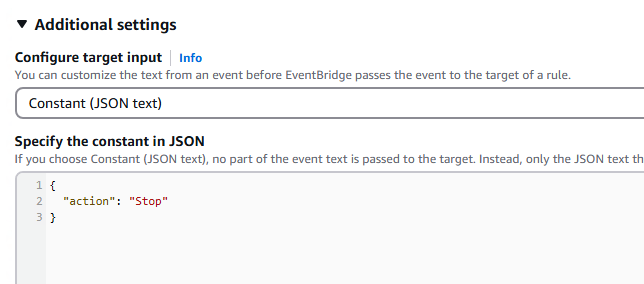




**Create one more Rule to stop the Instance.**

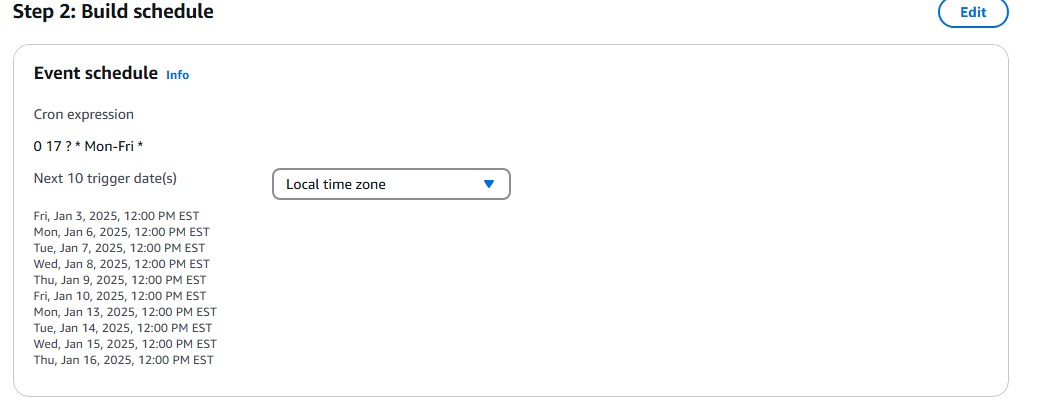


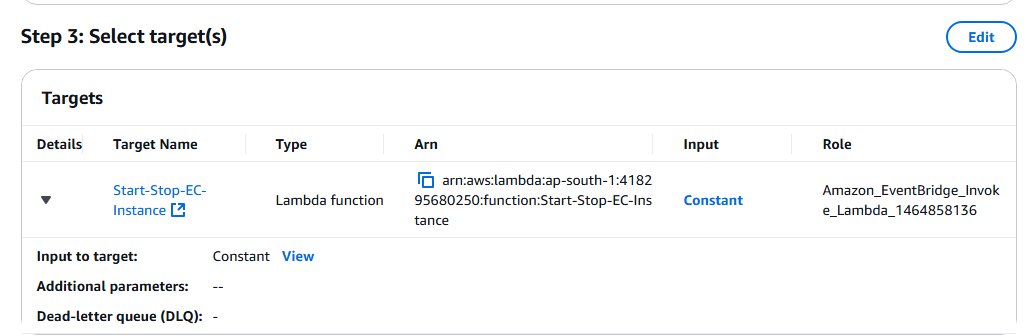


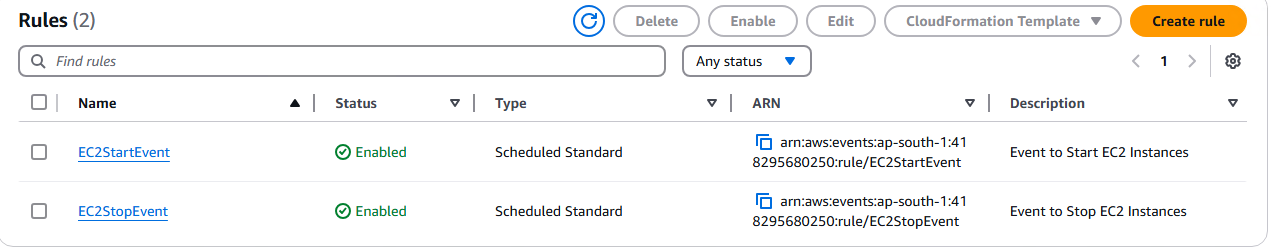


A screenshot of a computer

Description automatically generated







**You can check the logs in the CloudWatch**

**Schedule types in EventBridge Scheduler:**

*The following topic describes the different schedule types that Amazon EventBridge Scheduler supports, as well as how EventBridge Scheduler handles daylight savings time, and scheduling in different time zones. You can choose from three schedule types when configuring your schedule:****rate-based, cron-based, and one-time schedules****.*

***Both rate-based and cron-based schedules*** *are recurring schedules. You configure each recurring schedule type using a schedule expression for the type of schedule you want to configure, and specifying a time zone in which EventBridge Scheduler evaluates the expression.*

*A* ***one-time schedule is a schedule that invokes a target only once****. You configure a one-time schedule when by specifying the time, date, and time zone in which EventBridge Scheduler evaluates the schedule.*

**Rate-based schedules:**

*A rate-based schedule starts after the start date you specify for your schedule, and runs at a regular rate that you define until the schedule's end date. You can set up most common recurrent scheduling use cases using a rate-based schedule. For example, if you want a schedule that invokes it's target every 15 minutes, once every two hours, or once every five days, you can use a rate-based schedule to achieve this. You configure a rate-based schedule using a rate expression.*

***rate(value unit)***

**Cron-based schedules:**

*A cron expression creates a fine-grained recurring schedule that runs at a specific time of your choosing. EventBridge Scheduler supports configuring cron-based schedules in Universal Coordinated Time (UTC), or in the time zone that you specify when you create your schedule. With cron-based schedules, you have more control over when and how often your schedule runs. Use cron-based schedules when you need a customized recurrence schedule that is not supported by one of EventBridge Scheduler's rate expressions. For example, you can create a cron-based schedule that runs at 8:00 a.m. PST on the first Monday of every month. You configure a cron-based schedule using a cron expression.*

*A cron expression consists of five required fields separated by white space:* ***minutes, hours, day-of-month, month, day-of-week, and one optional field****.*

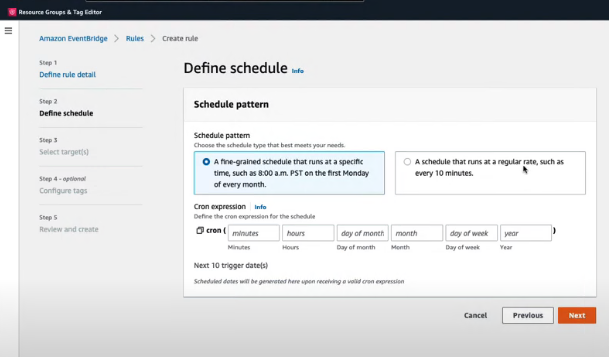
***cron(minutes hours day-of-month month day-of-week year)***

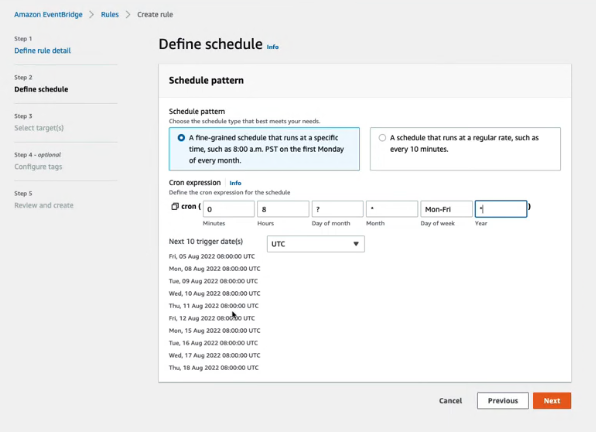
**One-time schedules:**

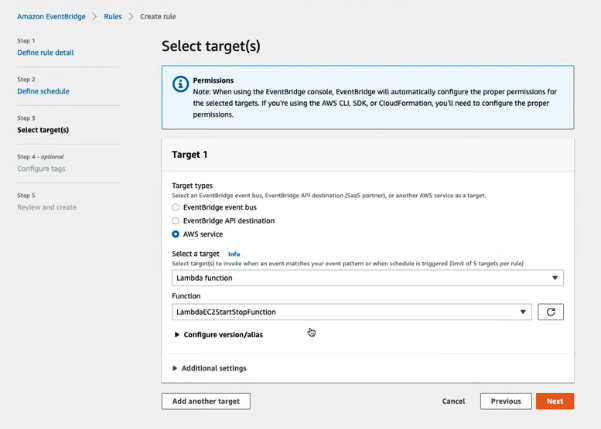
*A one-time schedule will invoke a target only once at the date and time that you specify using a valid date, and a timestamp. EventBridge Scheduler supports scheduling in Universal Coordinated Time (UTC), or in the time zone that you specify when you create your schedule.*

***at(yyyy-mm-ddThh:mm:ss)***

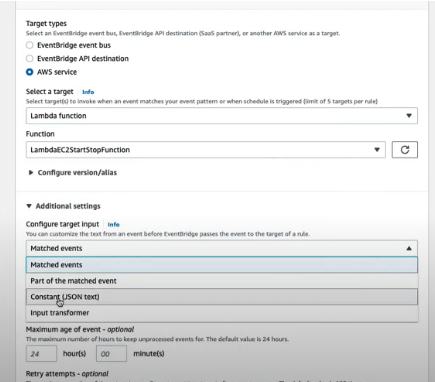
**Define schedule for the EventBridge to trigger:**

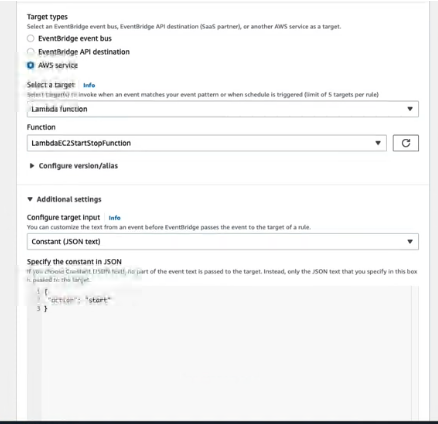




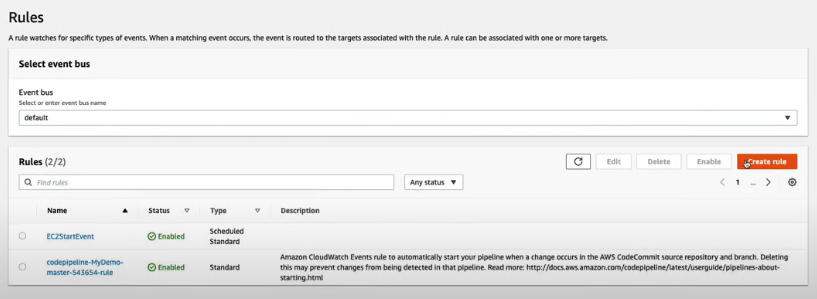


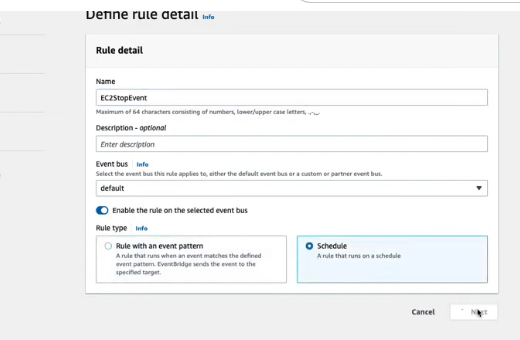
**Expand Additional Settings:**

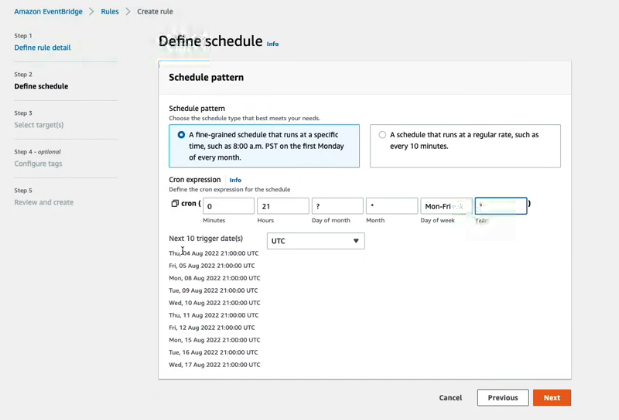




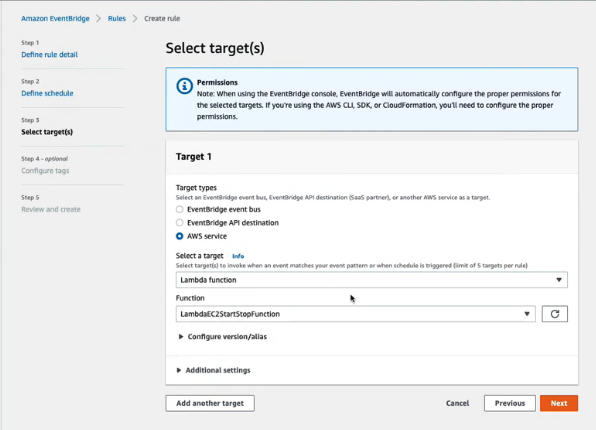
**Create Stop Event:**

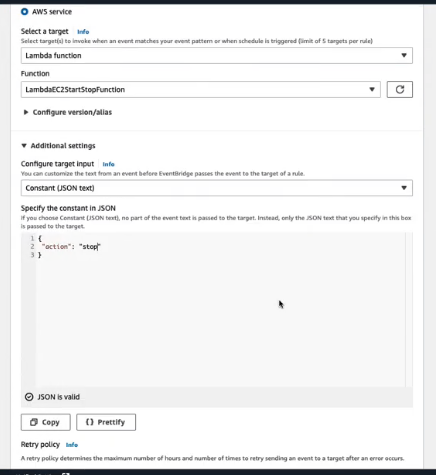


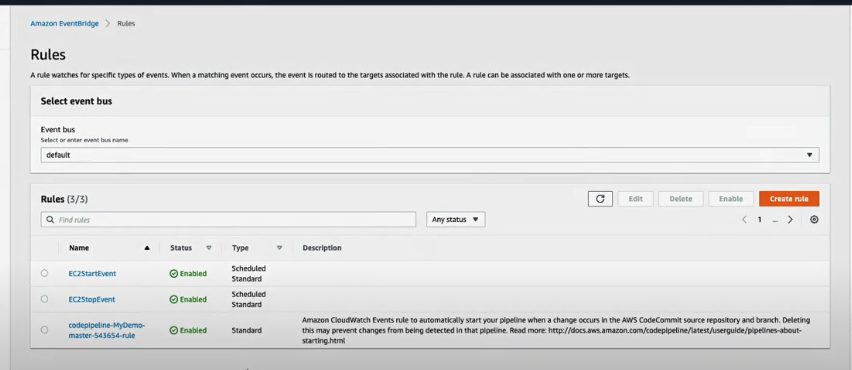




**Select the Lamda Function created as the Target:**







**The activities can be seen in CloudWatch**

