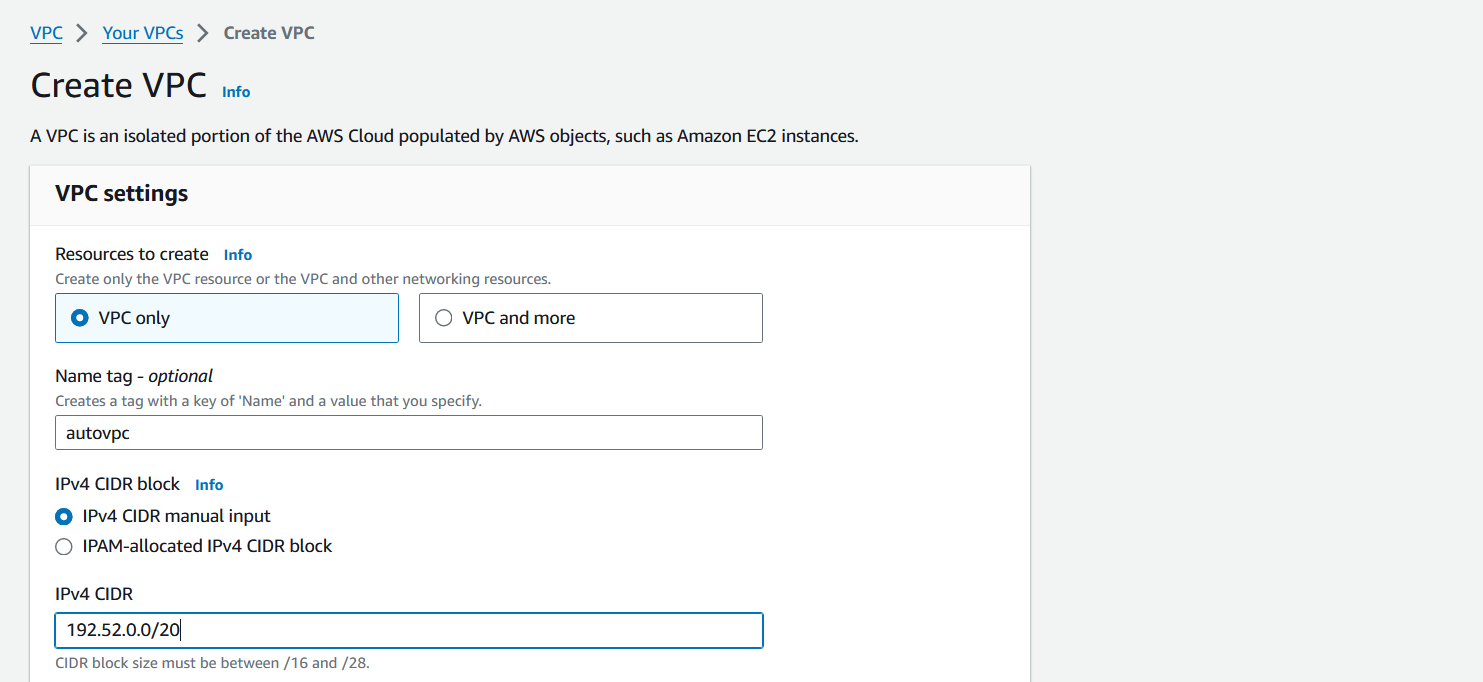
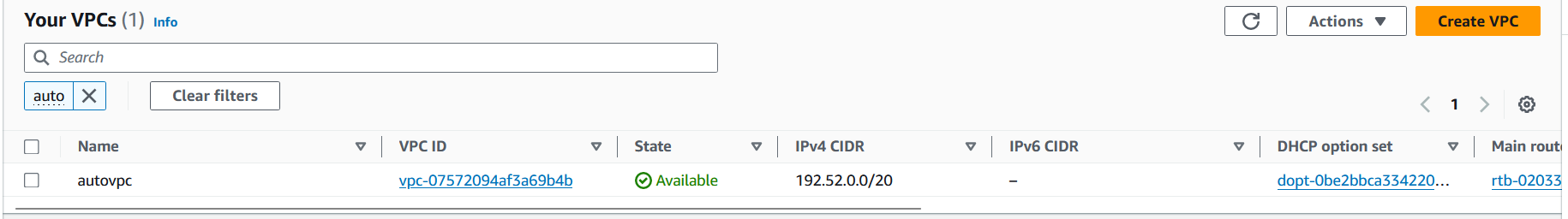
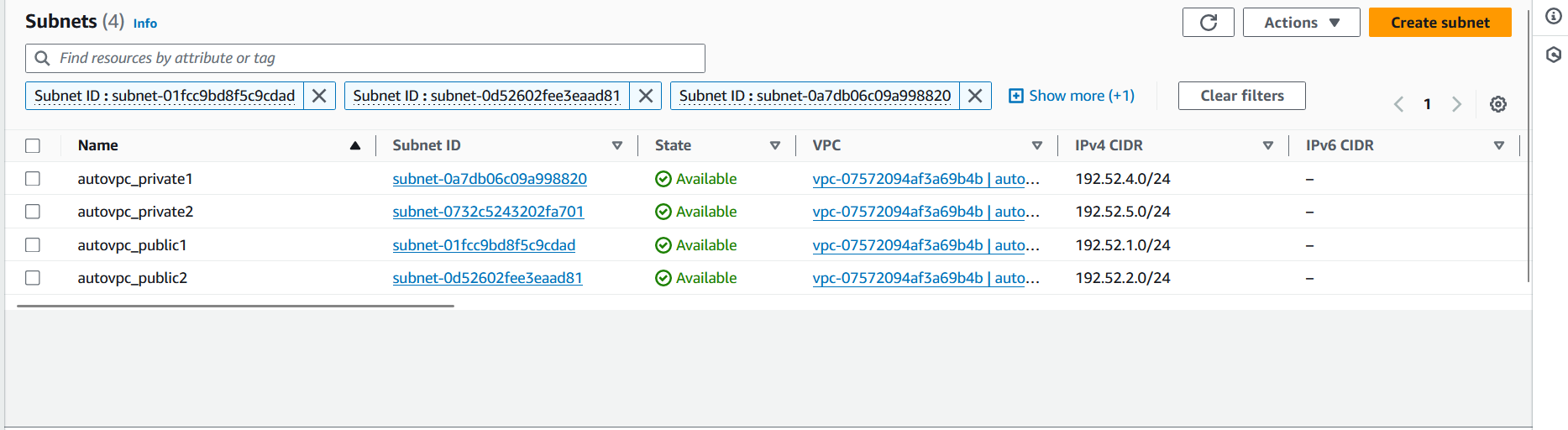
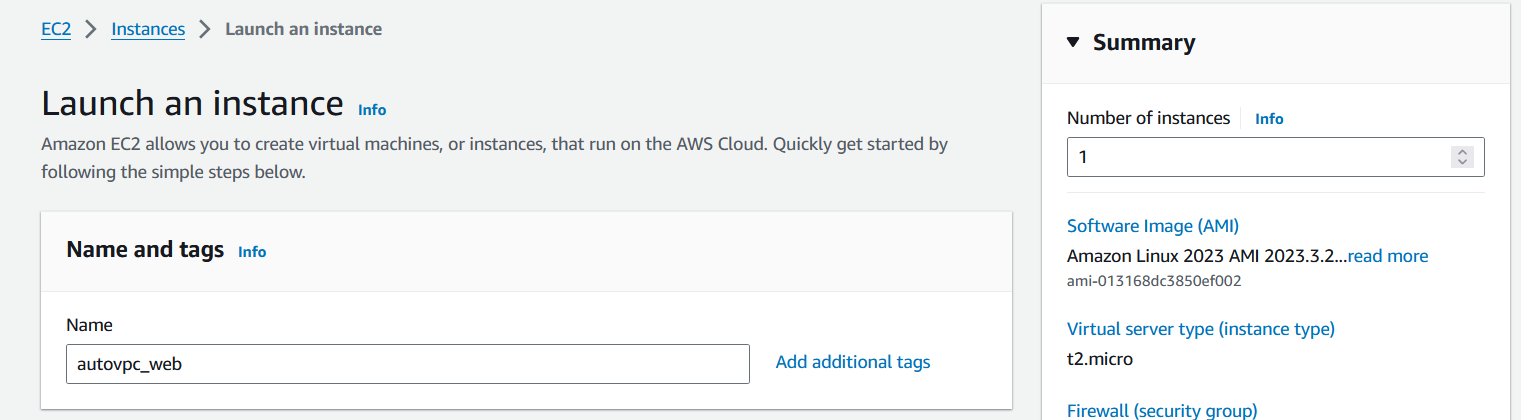
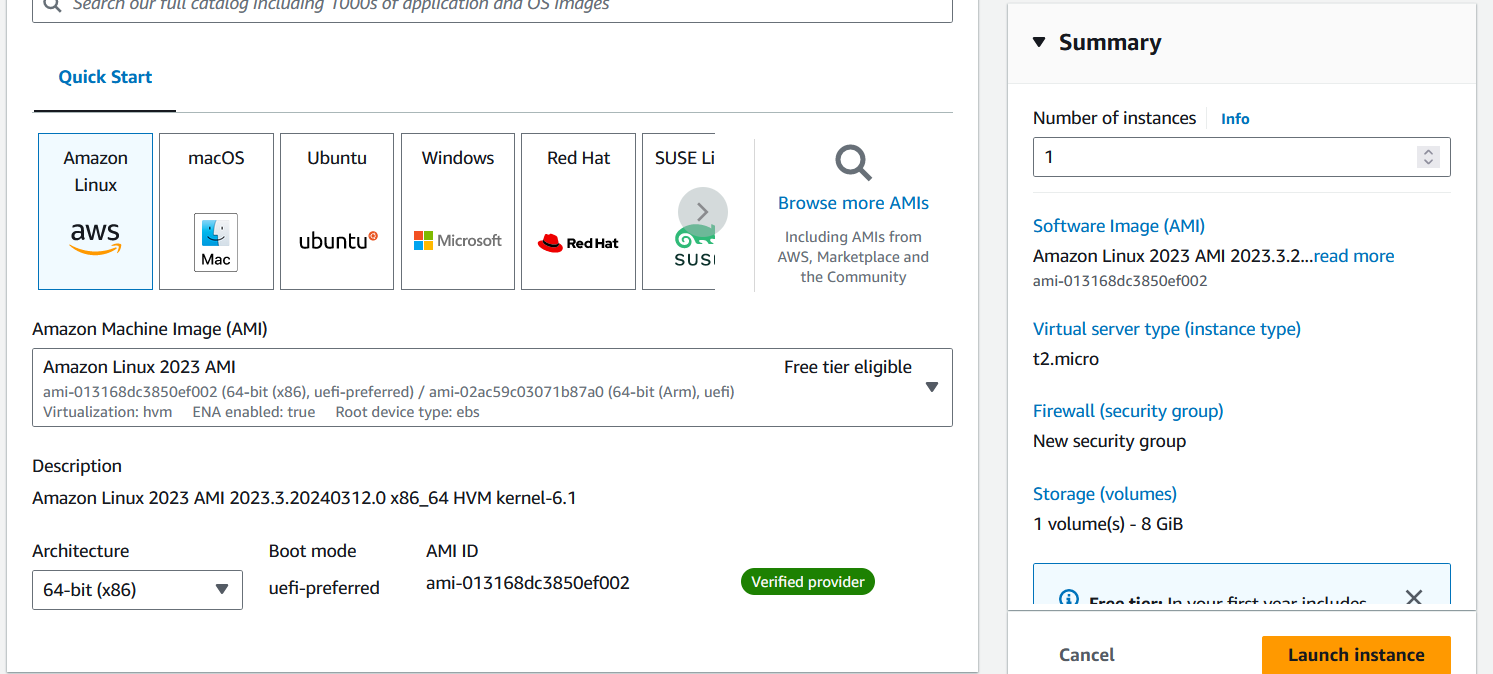
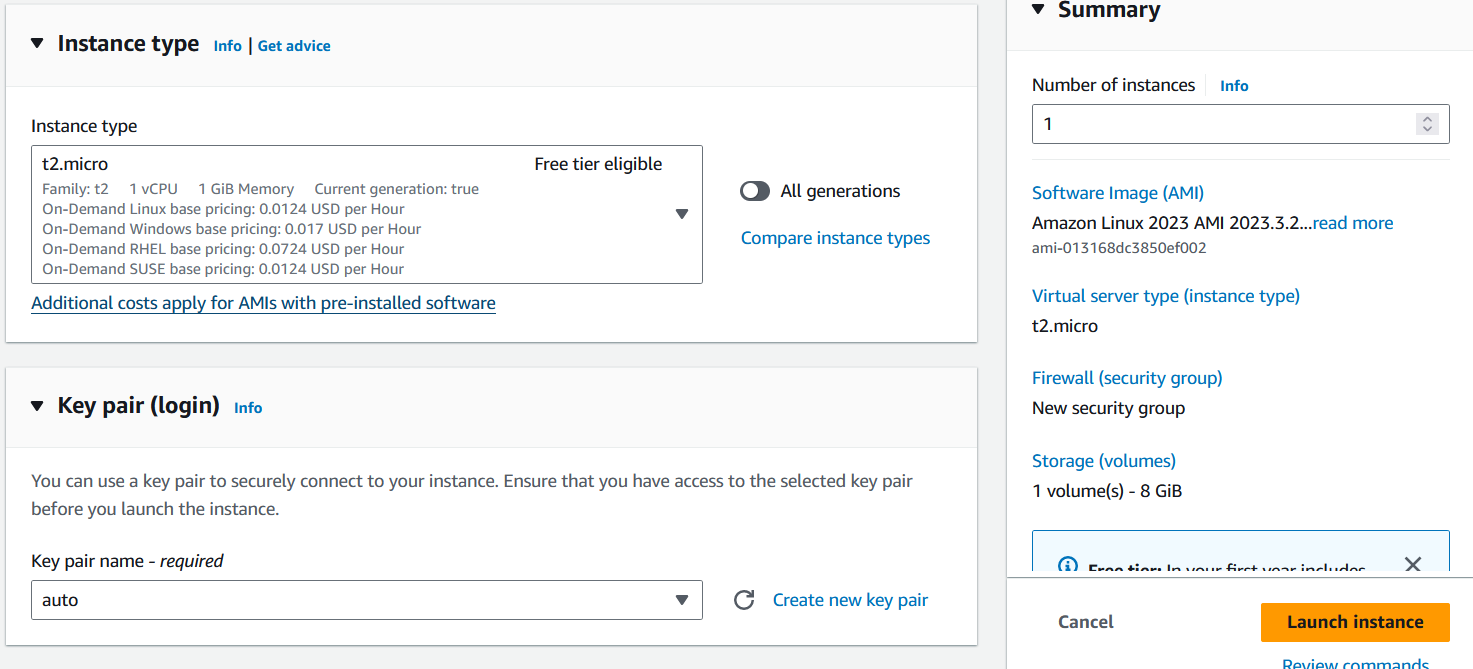
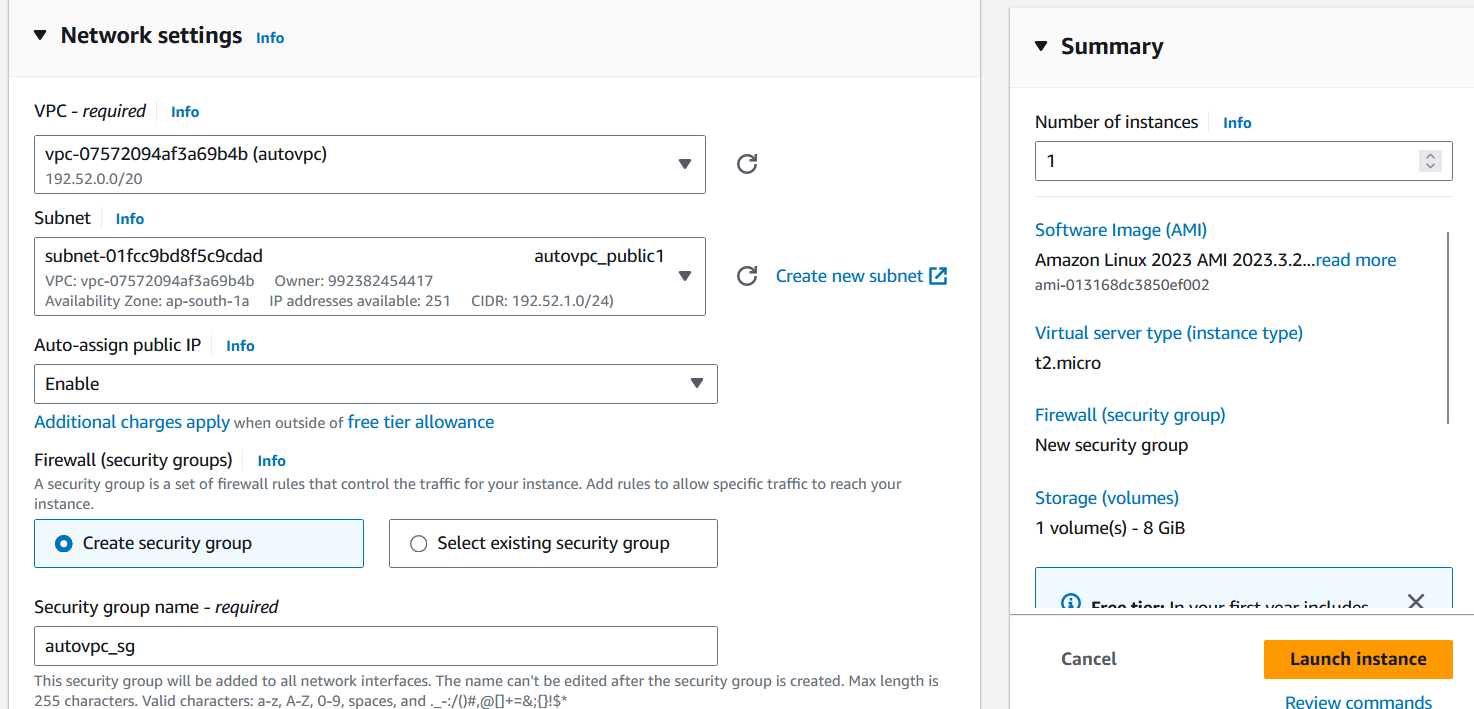
**Start and Stop EC2 Instances at specific time of a day using CloudWatch and Lambda - Assignment**

**Prerequisites**

An Ec2 instance within an isolated VPC.





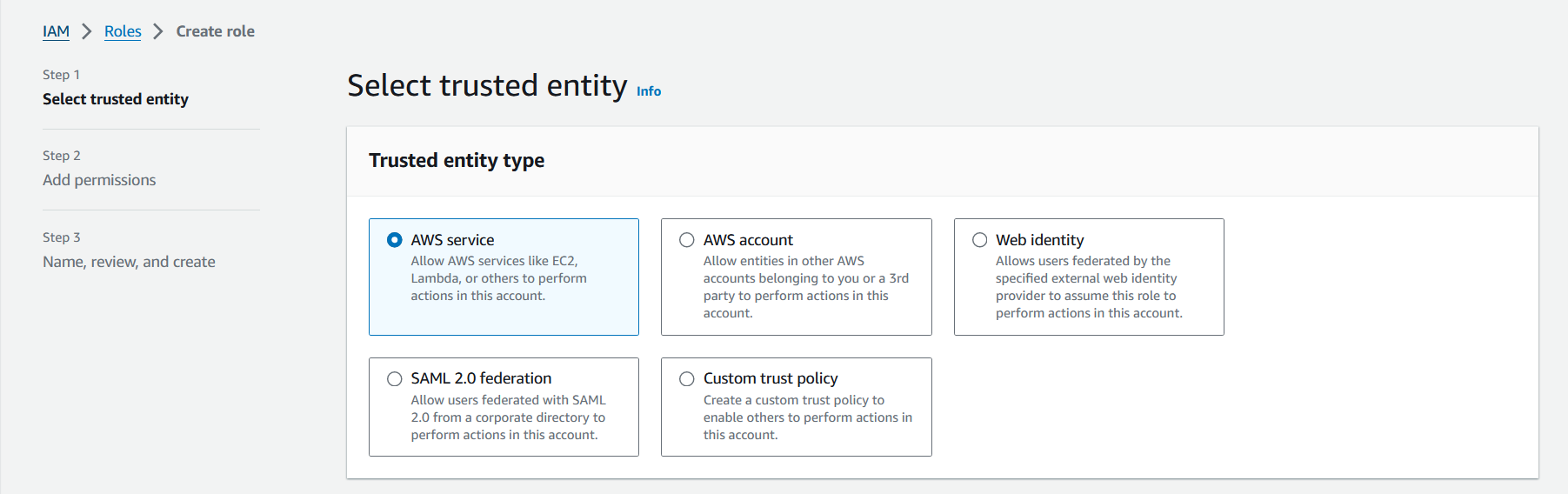
    

1. **IAM Role Setup**

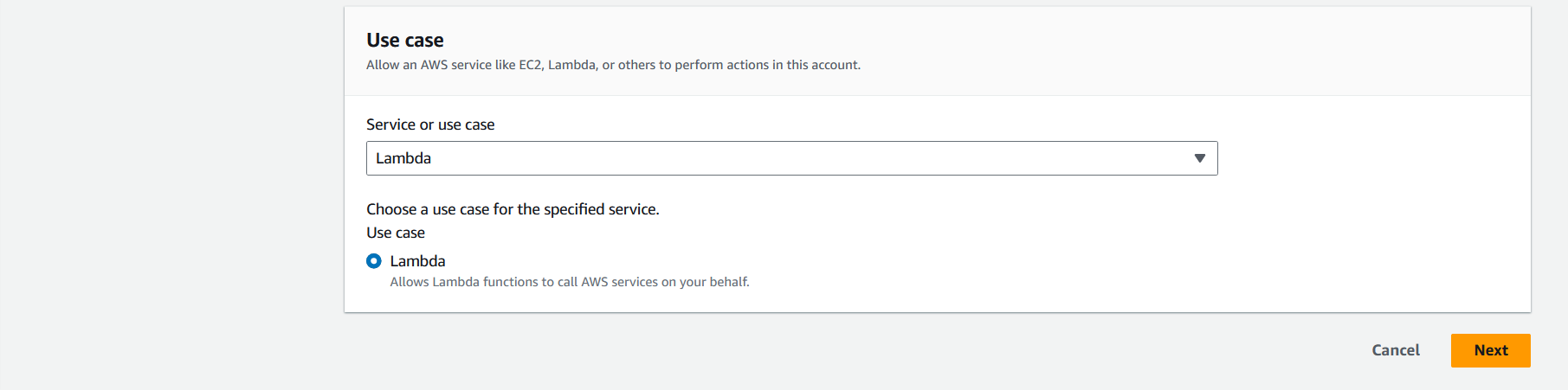
**IAM Role Configuration:**

First, you need to create an IAM role with permissions to interact with EC2 instances. Follow these steps:

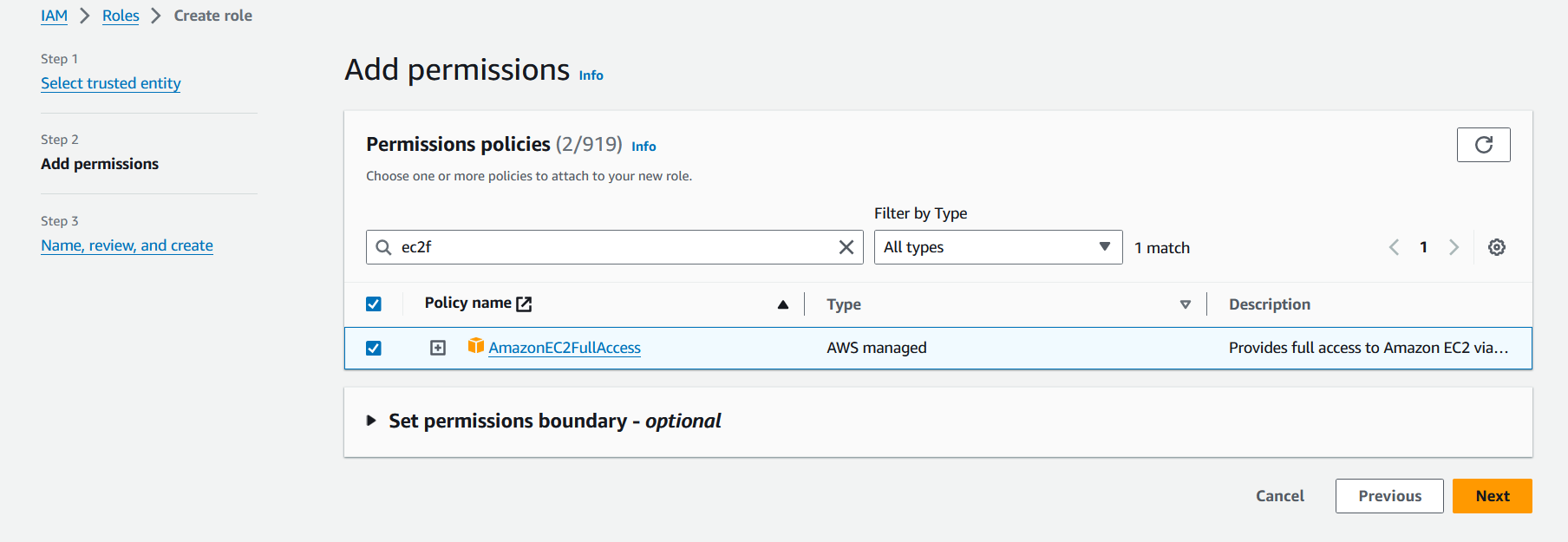
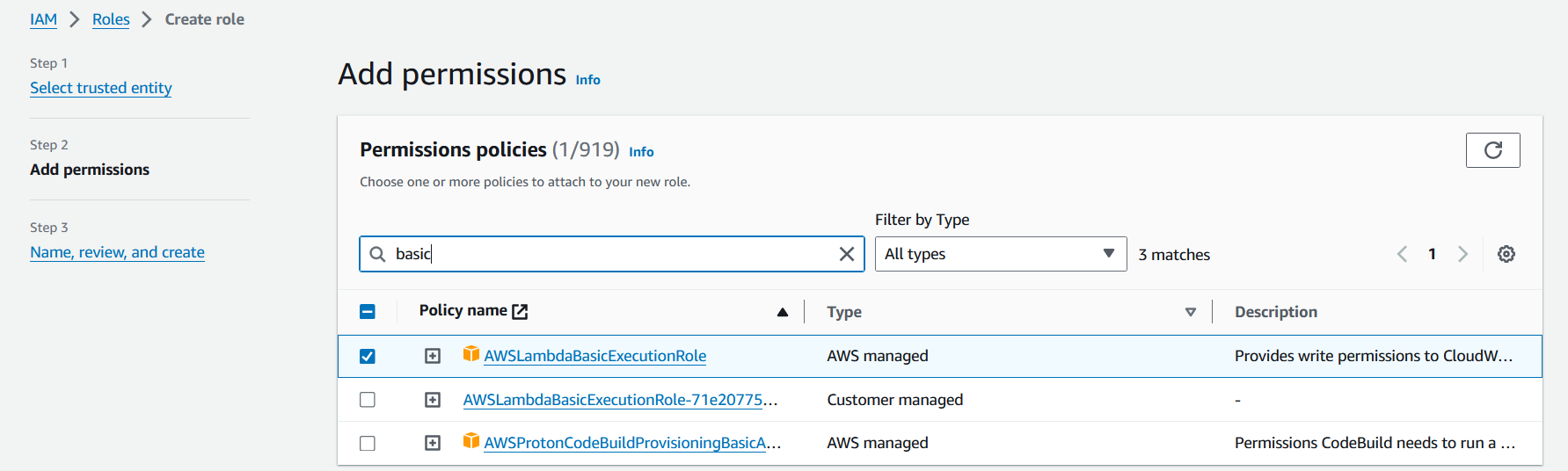
* Click on "Roles" in the left sidebar.
* Click "Create role".



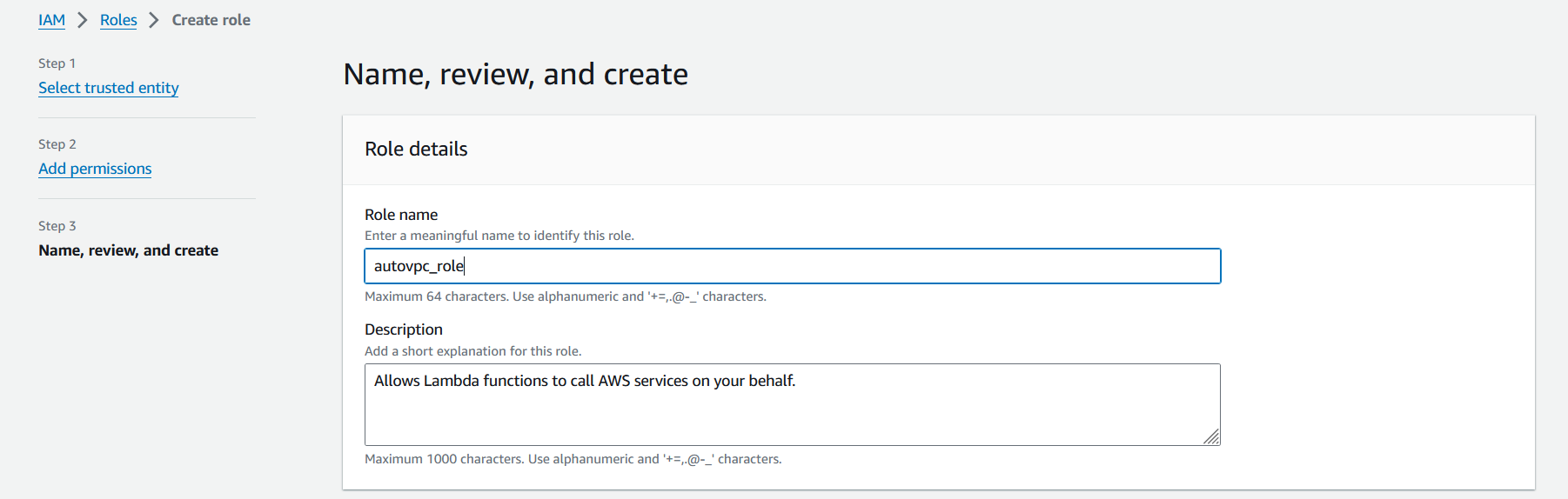
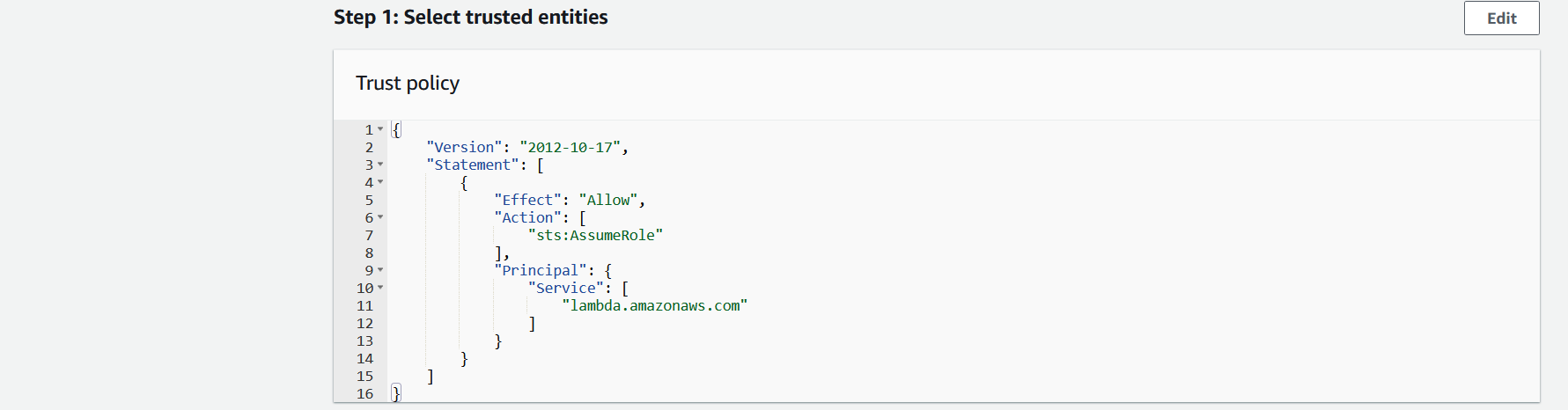
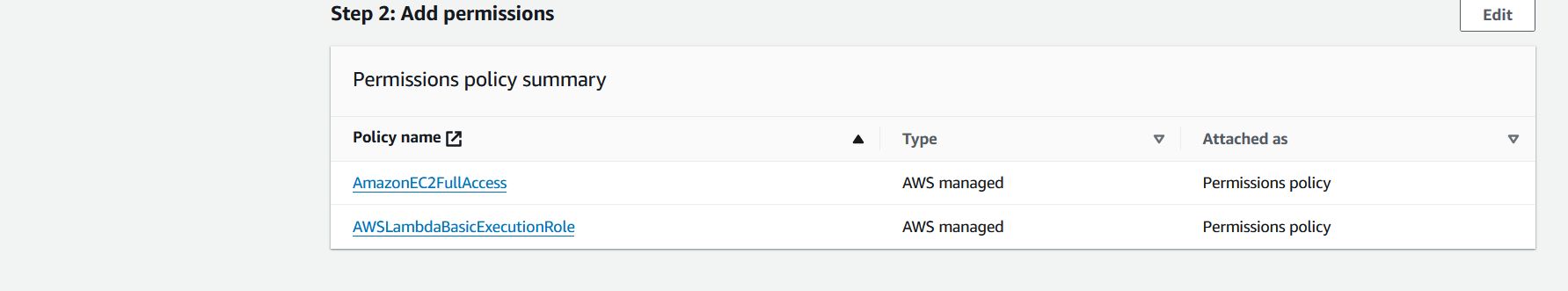
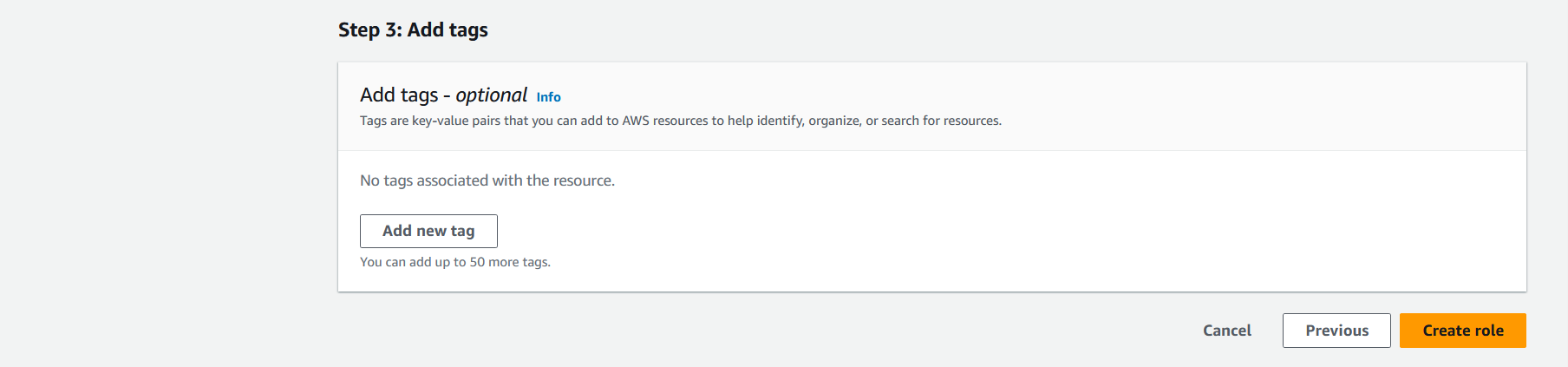
* Choose "Lambda" as the service that will use this role,



* Ensure your Lambda functions have the necessary permissions to start and stop EC2 instances. Create an IAM role with the following policies attached:
* AWSLambdaBasicExecutionRole (basic Lambda execution permissions)
* AmazonEC2FullAccess (full access to EC2, or you can customize the permissions based on your needs)



* Give your role a name and description, then click "Create role".

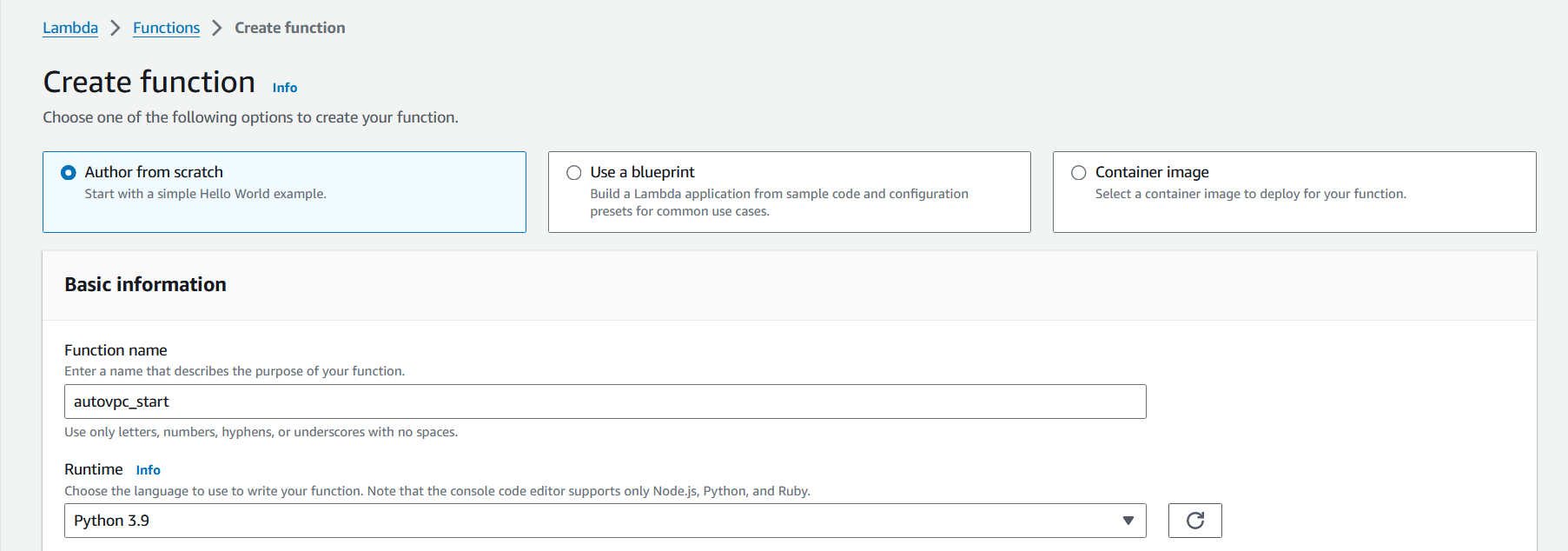
1. **Create Lambda Functions**

You'll need two Lambda functions: one to start instances and another to stop them.

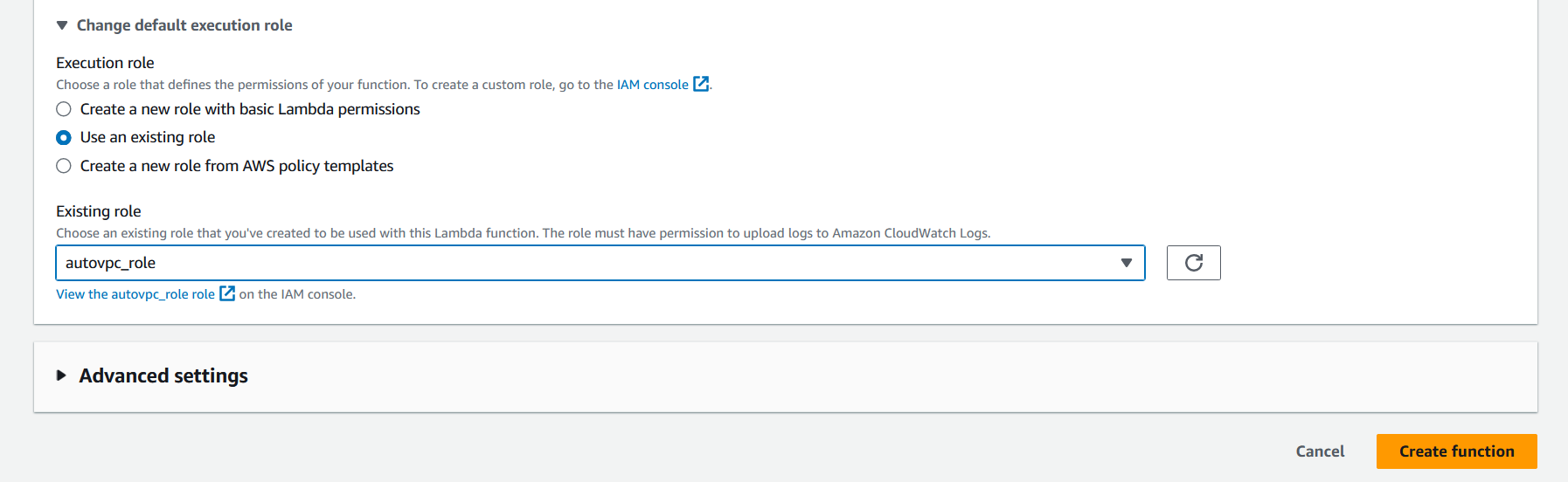
* 1. **Start EC2 Instances Lambda Function:**

This function starts your EC2 instances.

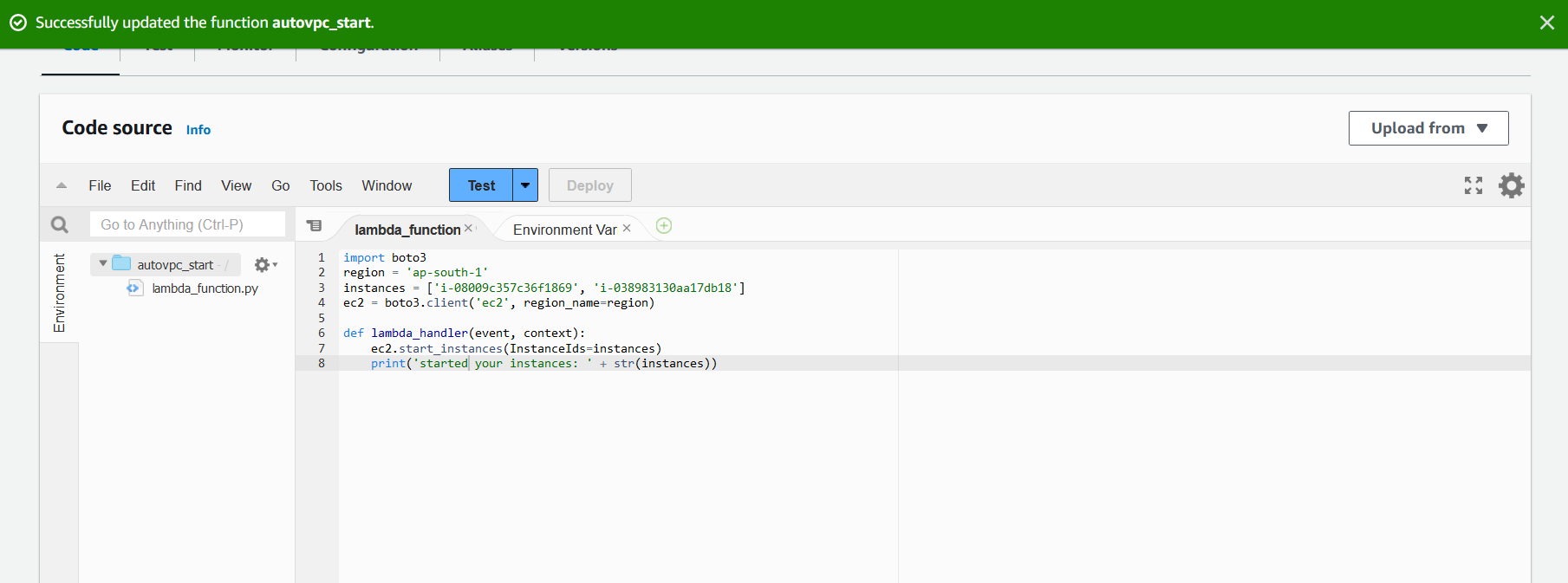
* Click on "Create function".
* Use “Author from scratch” option, choose a function name and runtime (Python 3.9) in our case



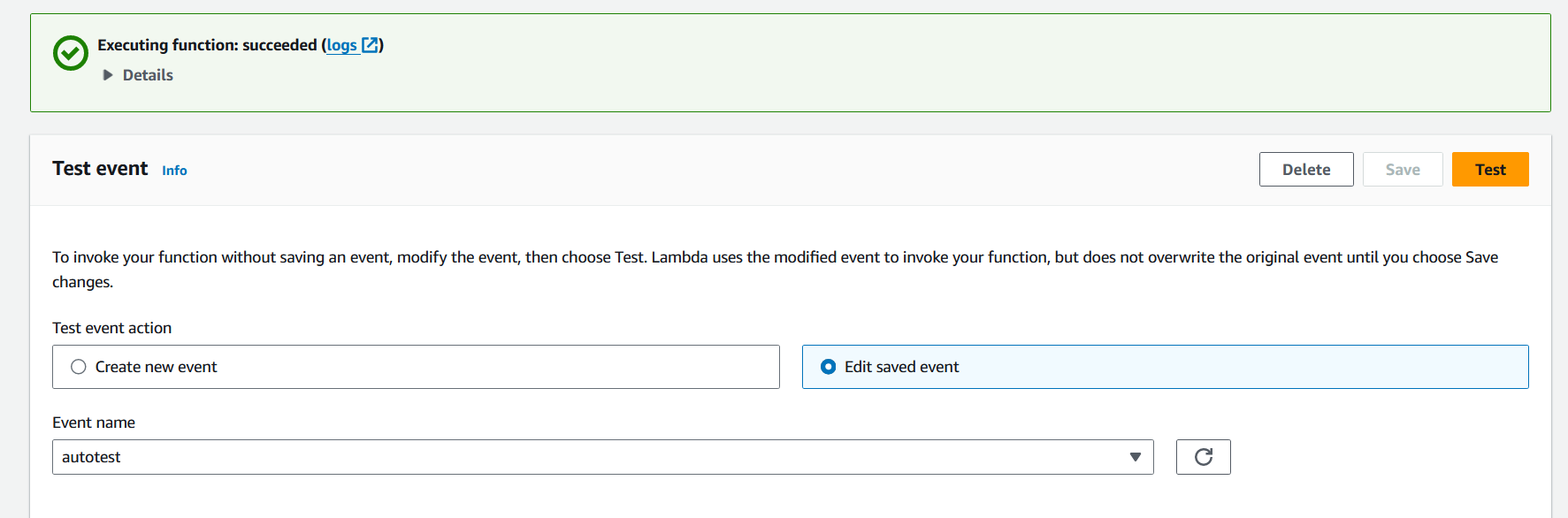
* Attach the IAM role in execution role



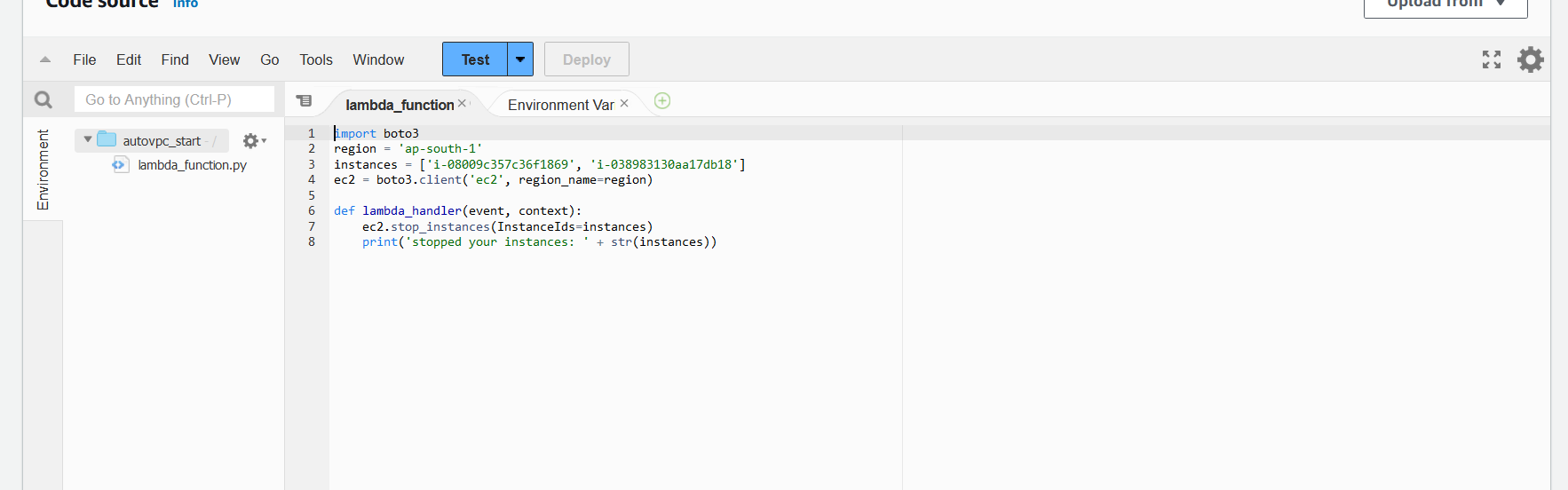
* Write the Lambda function code that starts the EC2 instances.

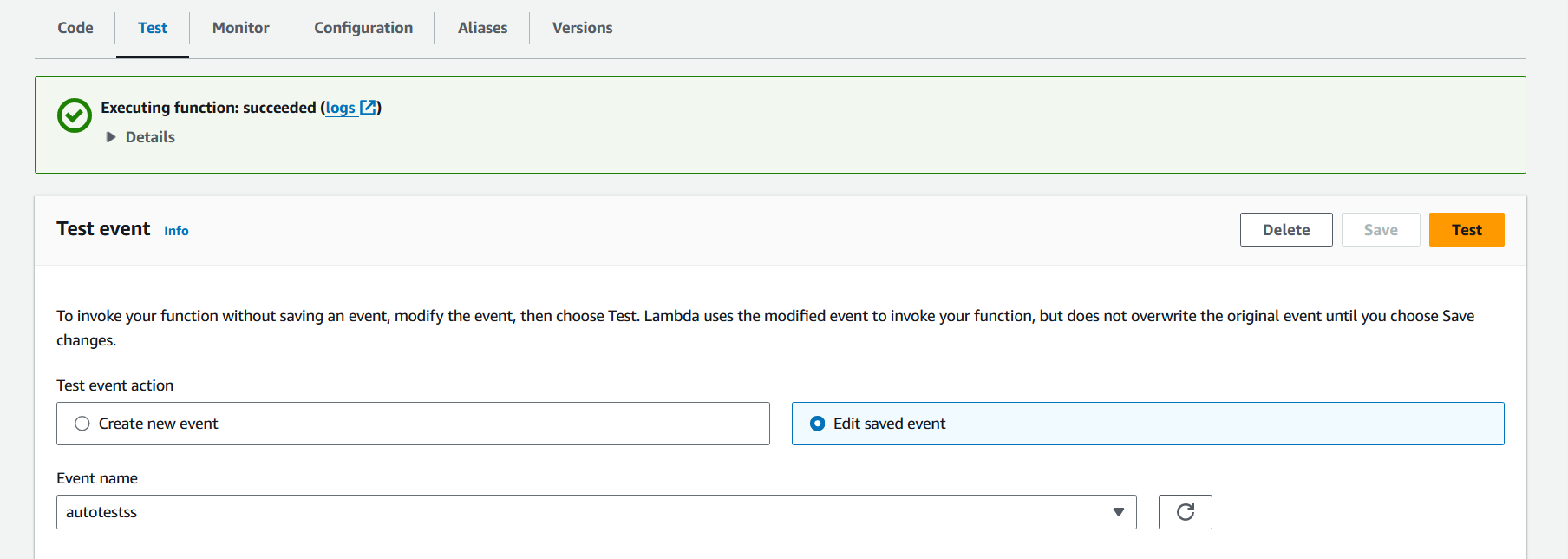


* Test the lambda function to verify the code



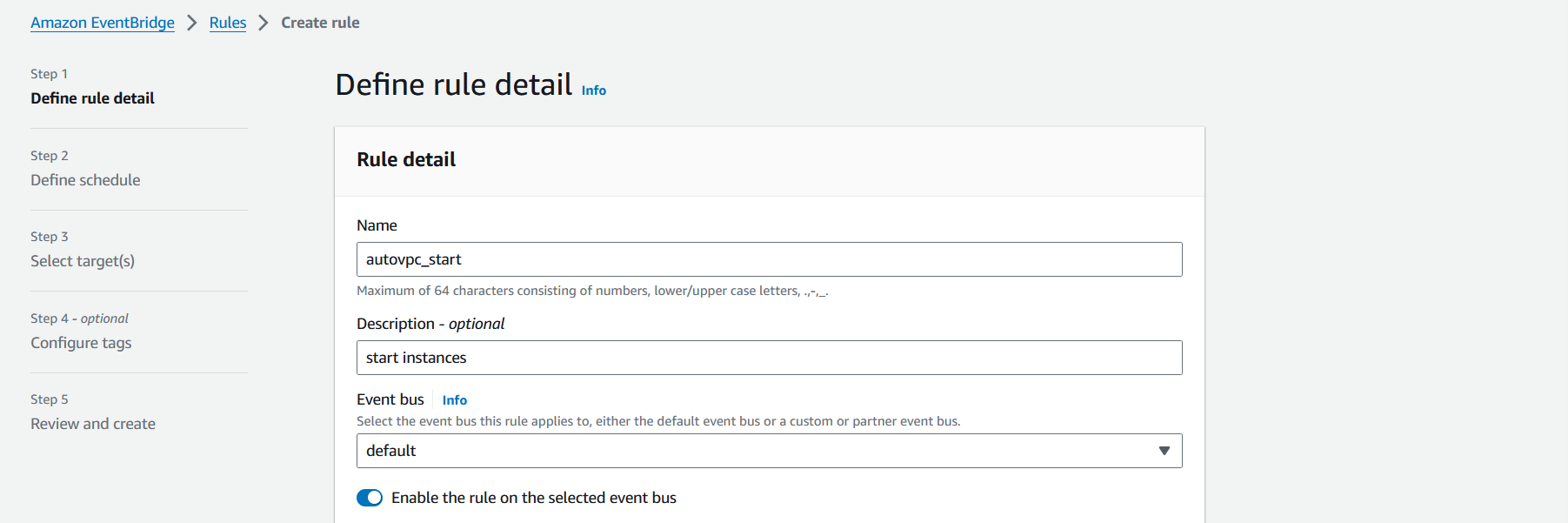
* Once this Lambda function is deployed, we can set up a CloudWatch Events rule to trigger it at the desired schedule.
  1. **Stop EC2 Instances Lambda Function:**
* Repeat the same steps as 2.1 and the only change needed is to change the function to ec2.stopinstances



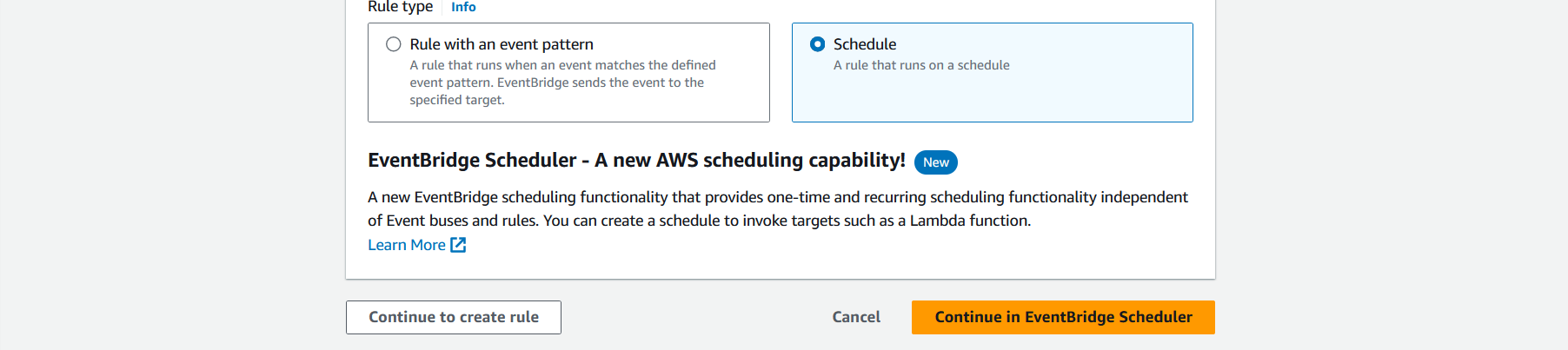


1. **Step 3: CloudWatch Events Setup**
   1. **Create a Rule for Starting EC2 Instances:**

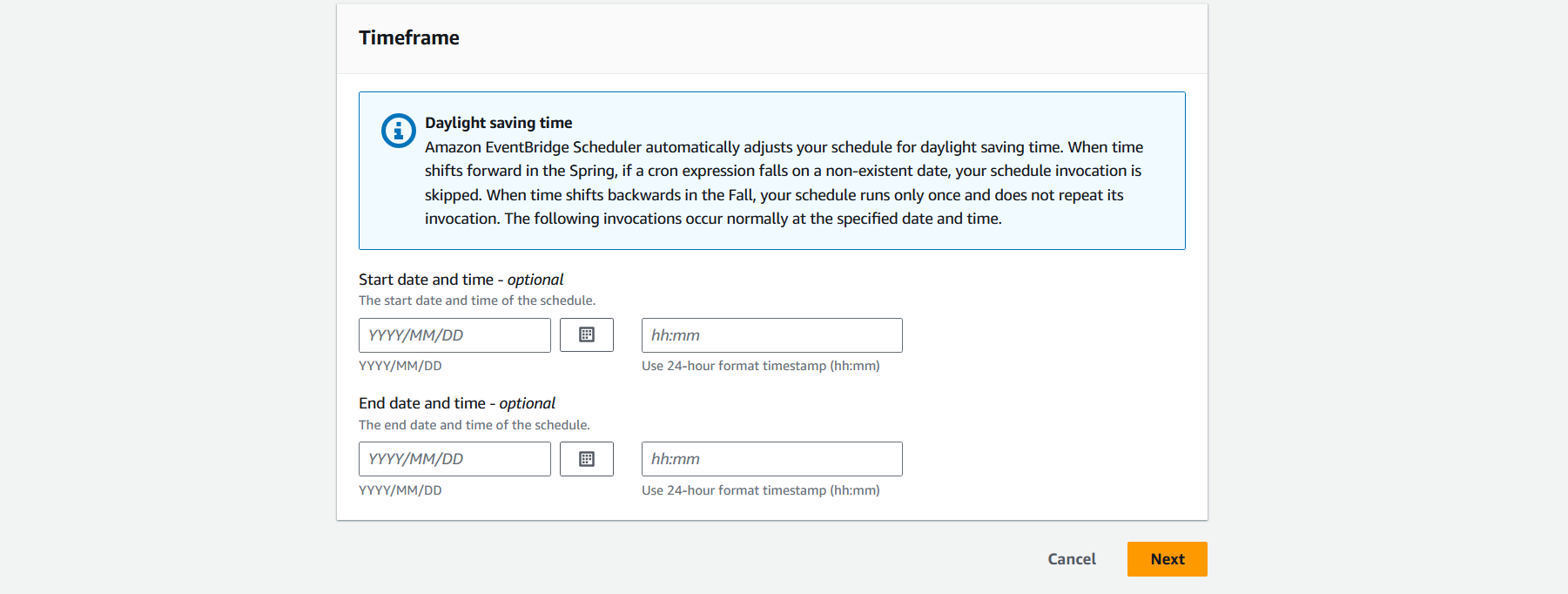
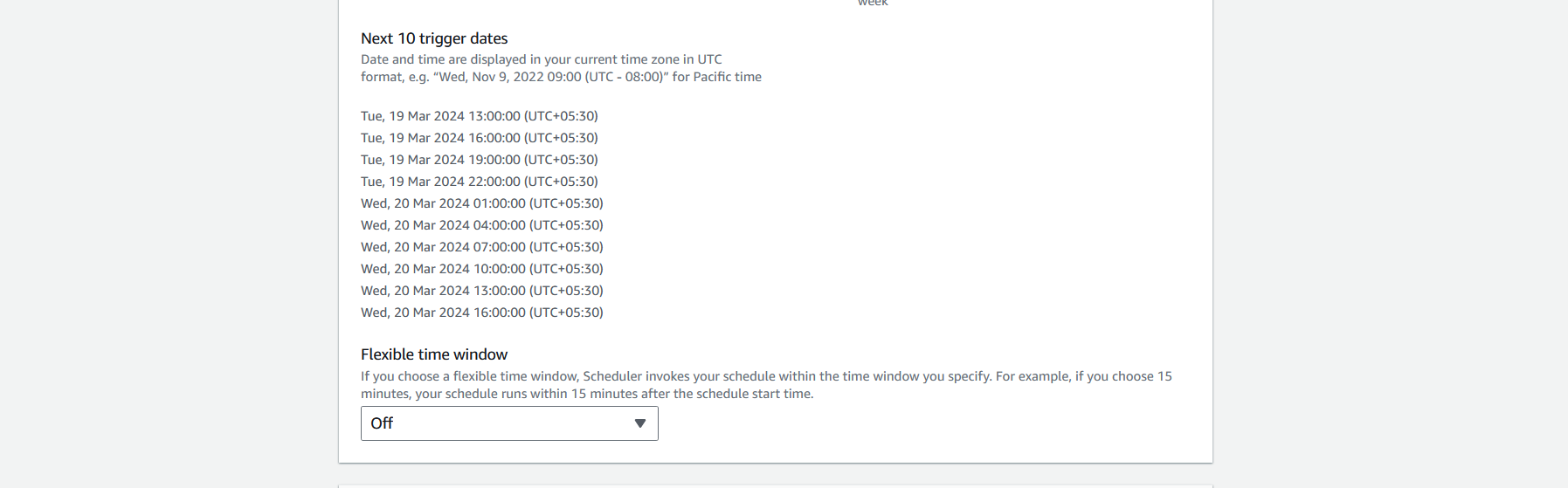
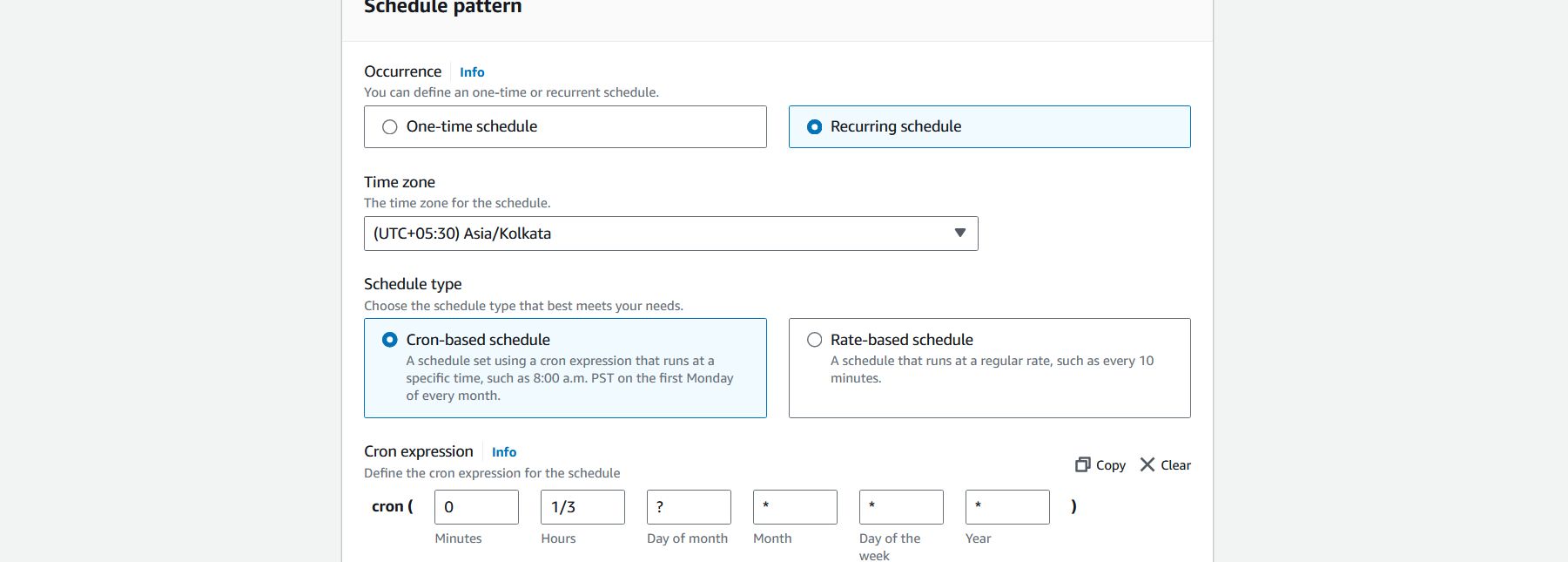
* Click on "Events" in the left sidebar under "Events" section.
* Click on "Create rule" button.



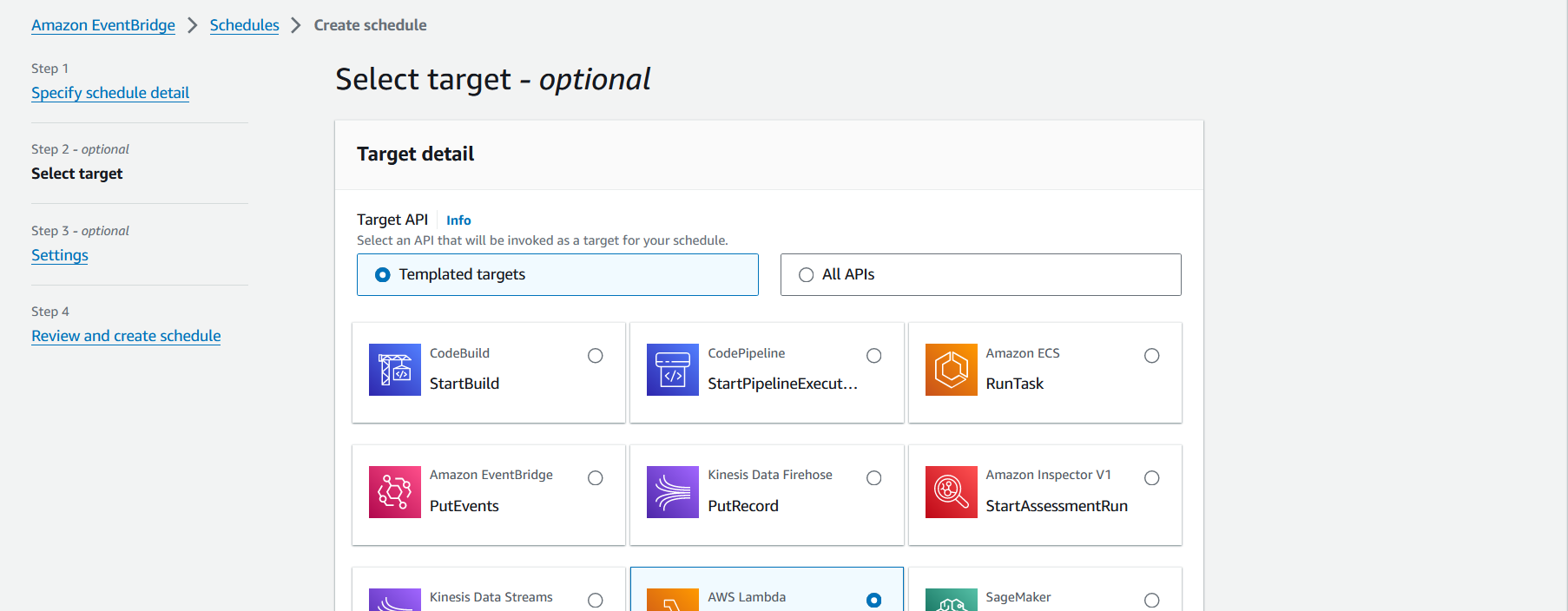
* Select "Schedule" as the event source.



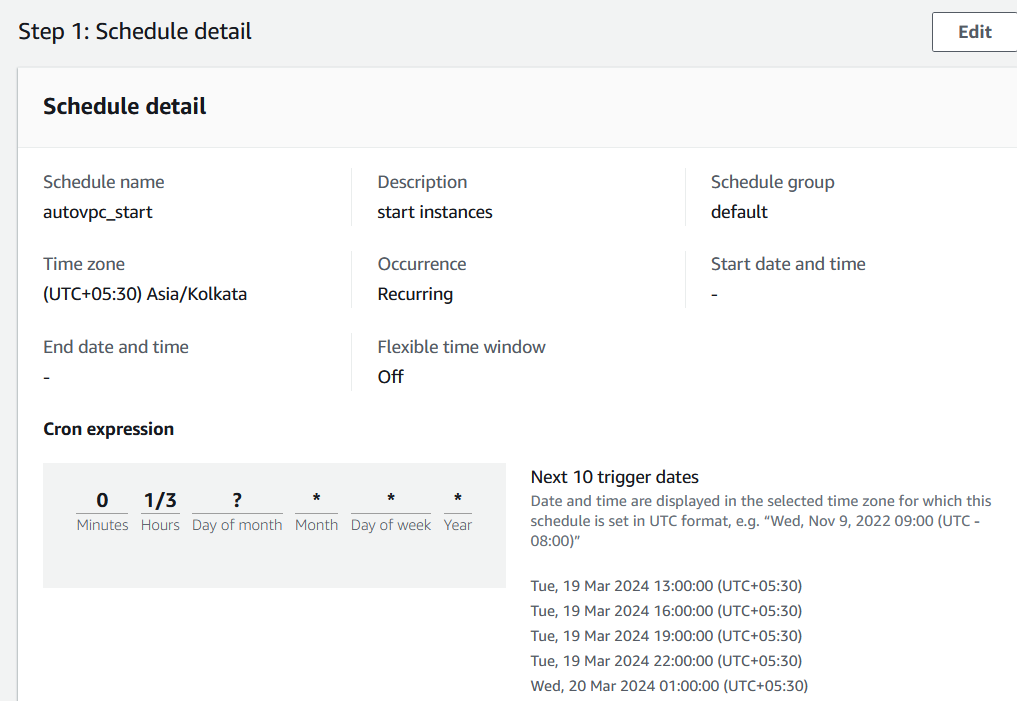
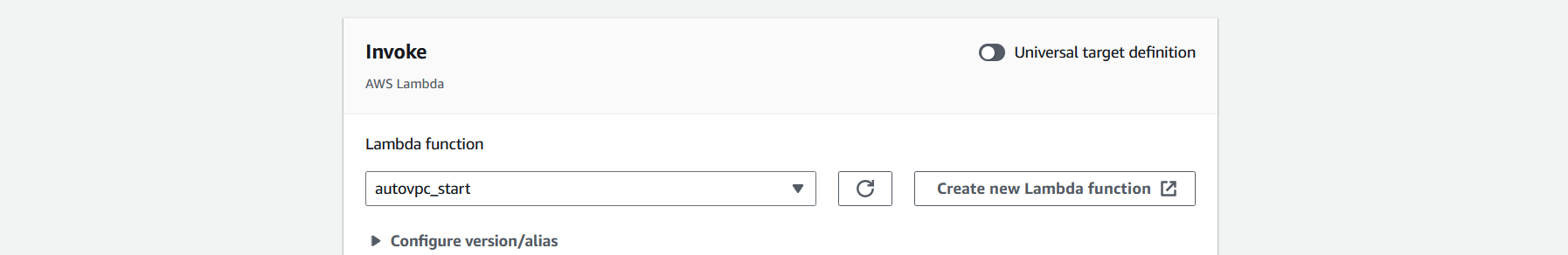
* Choose "Fixed rate of" and specify the frequency at which you want the rule to trigger (e.g., once every day).
* Configure the cron expression to specify the exact time when you want the rule to trigger.



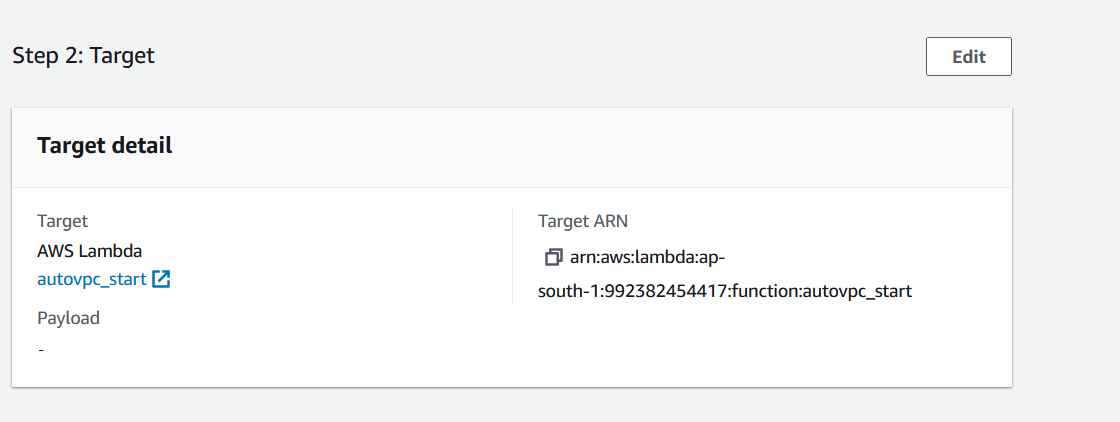
* Click on "Add target" and select "Lambda function".

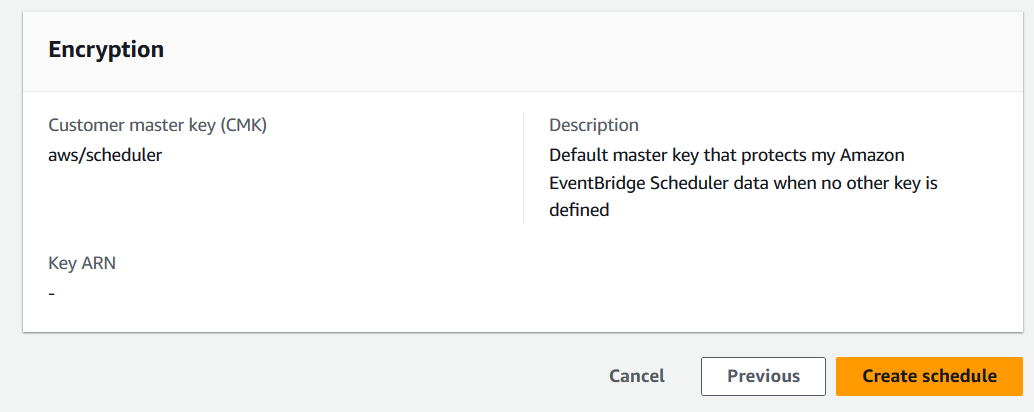


* Choose the Lambda function you created earlier that starts EC2 instances. Provide a name and description for the rule.

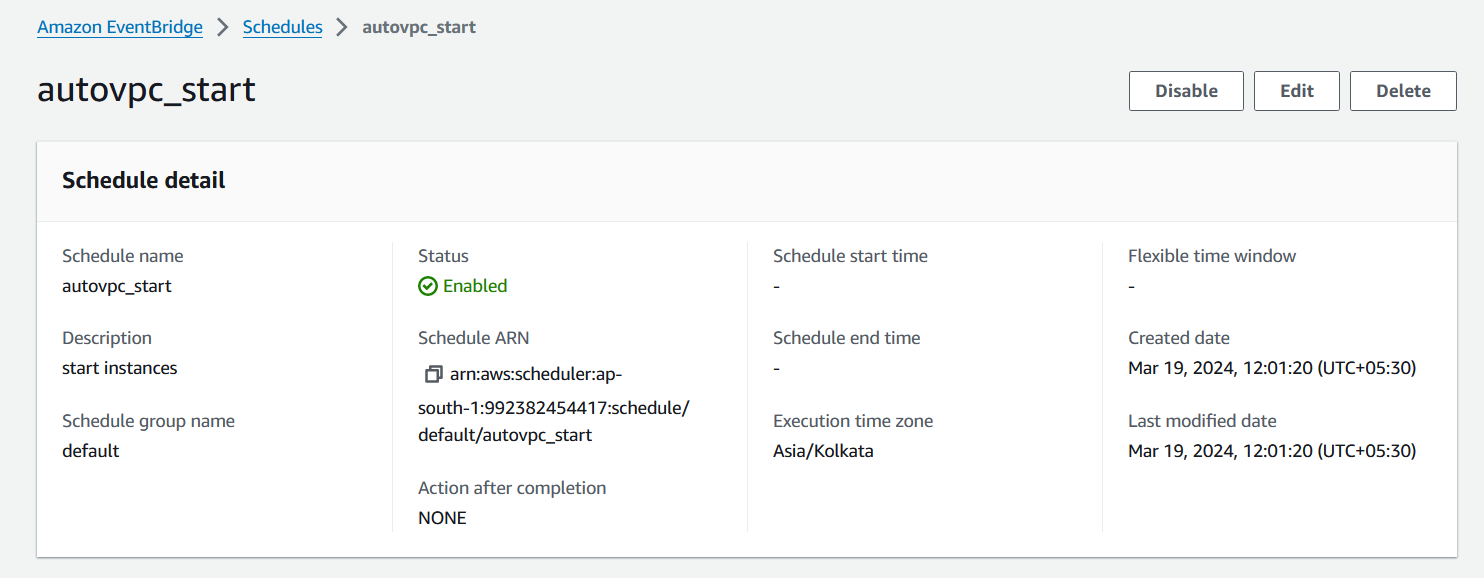


The cron is configured to trigger the lamba function after 1 hour for every 3 hours in a recurring fashion. This can be verified from the next trigger dates autogenerated by the aws.





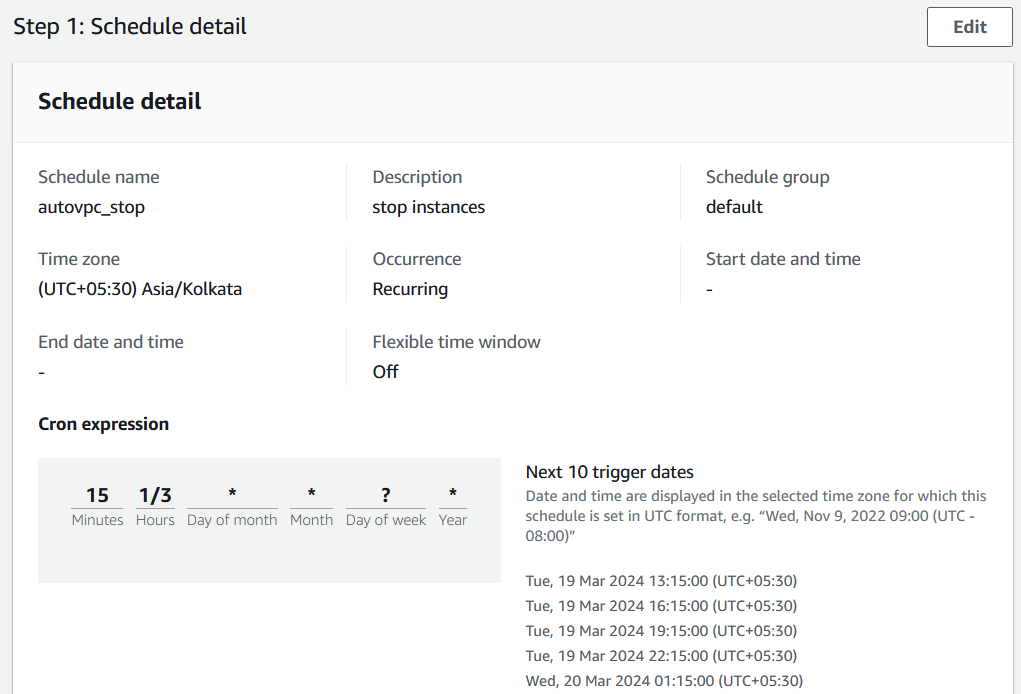
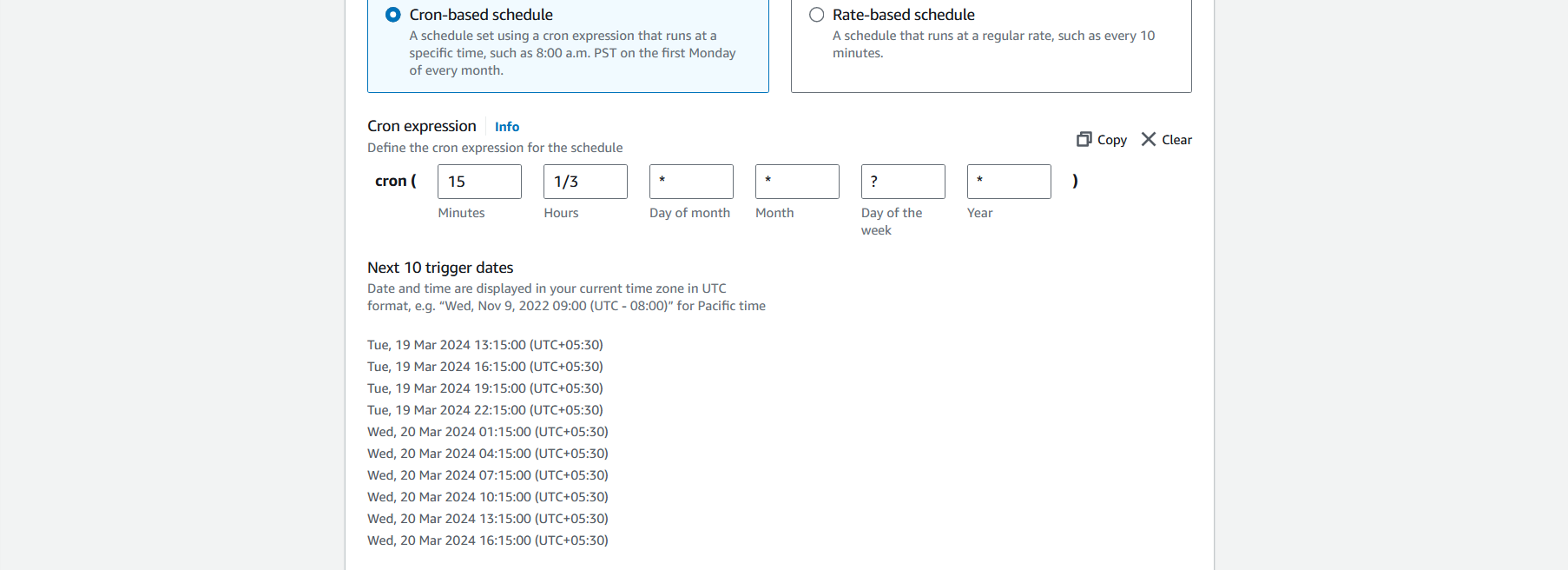
* Click "Create rule" to create the rule.



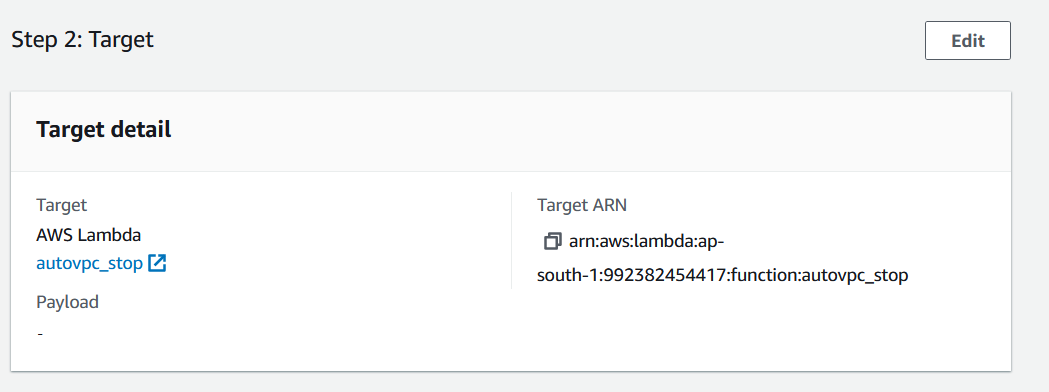
* 1. **Stop EC2 Instances Rule:**

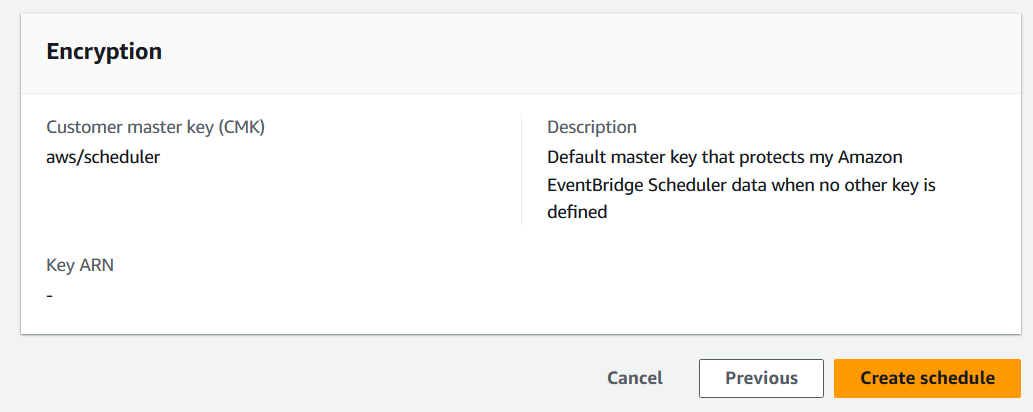
Follow the same steps as above, but create a new rule for stopping EC2 instances:

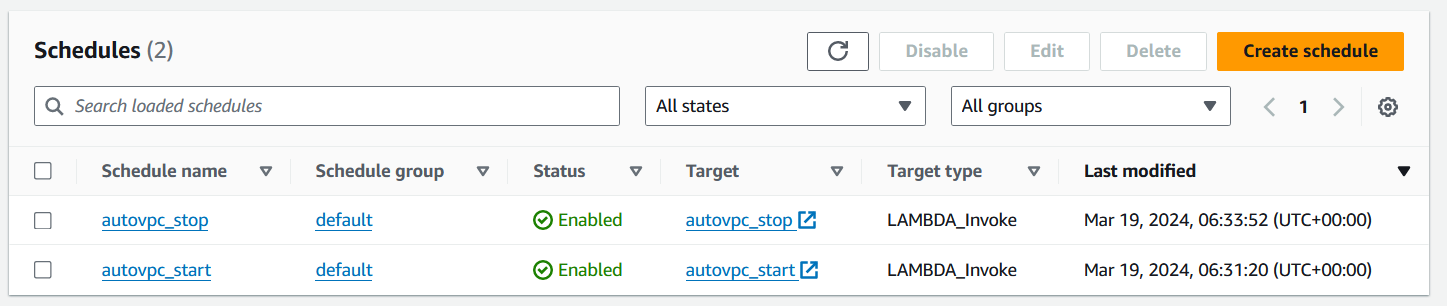
* Choose the Lambda function you created earlier that stops EC2 instances.
* Provide a name and description for the rule, indicating that it's for stopping EC2 instances.
* Click "Create rule" to create the rule.

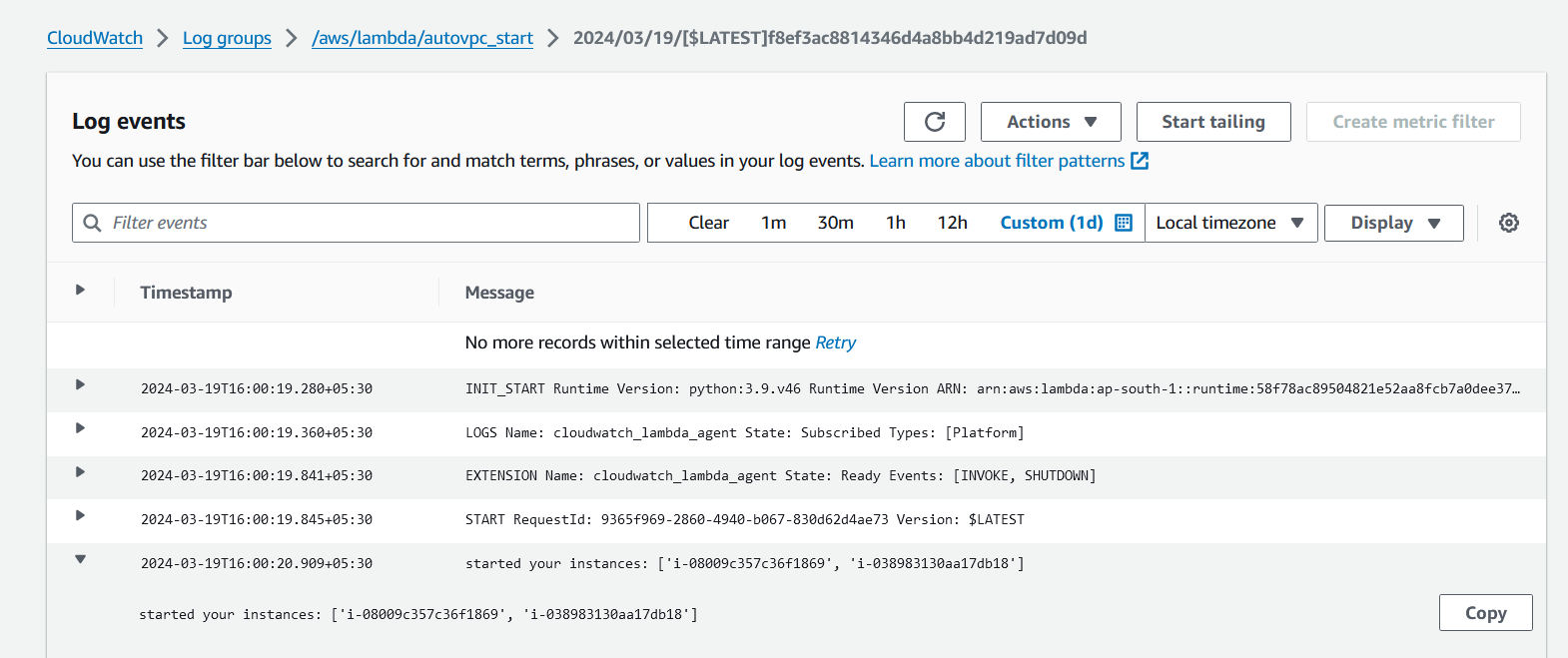


The cron is configured to trigger the lamba function after 15 minutes from the start instance timing every 3 hours in a recurring fashion.









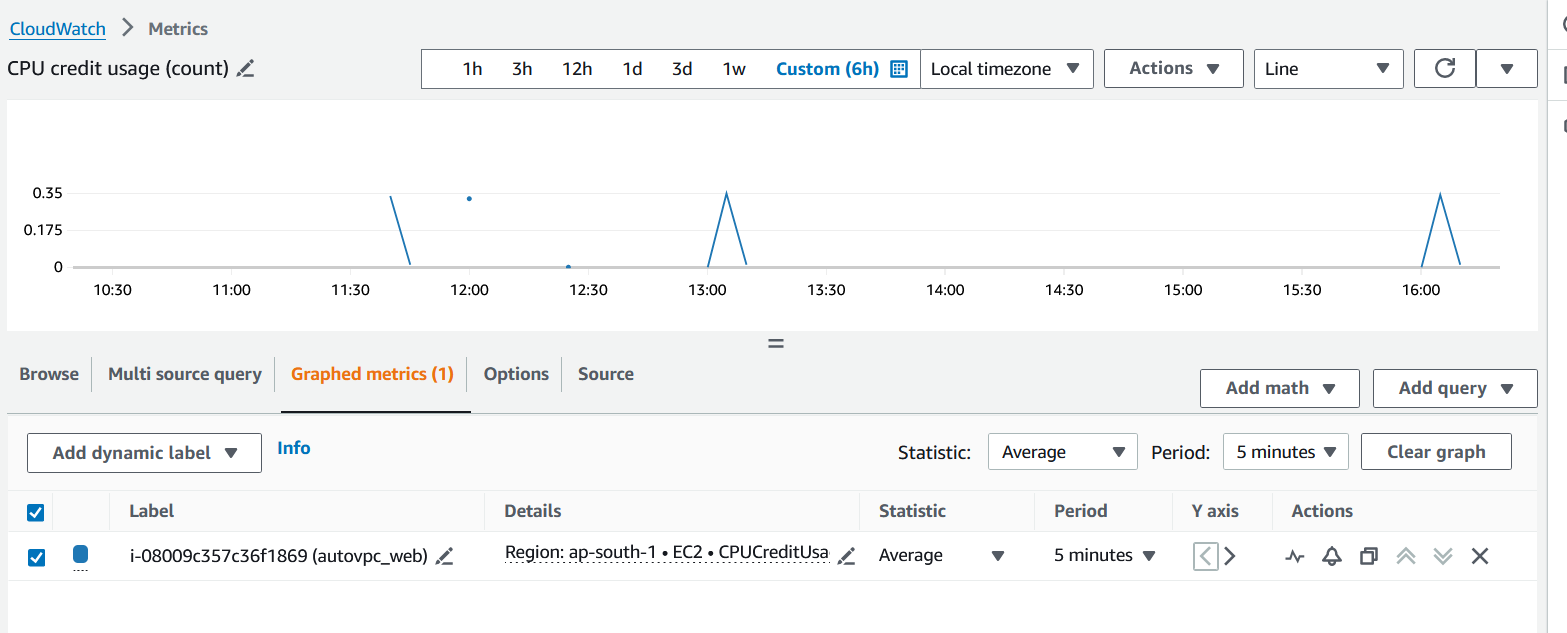
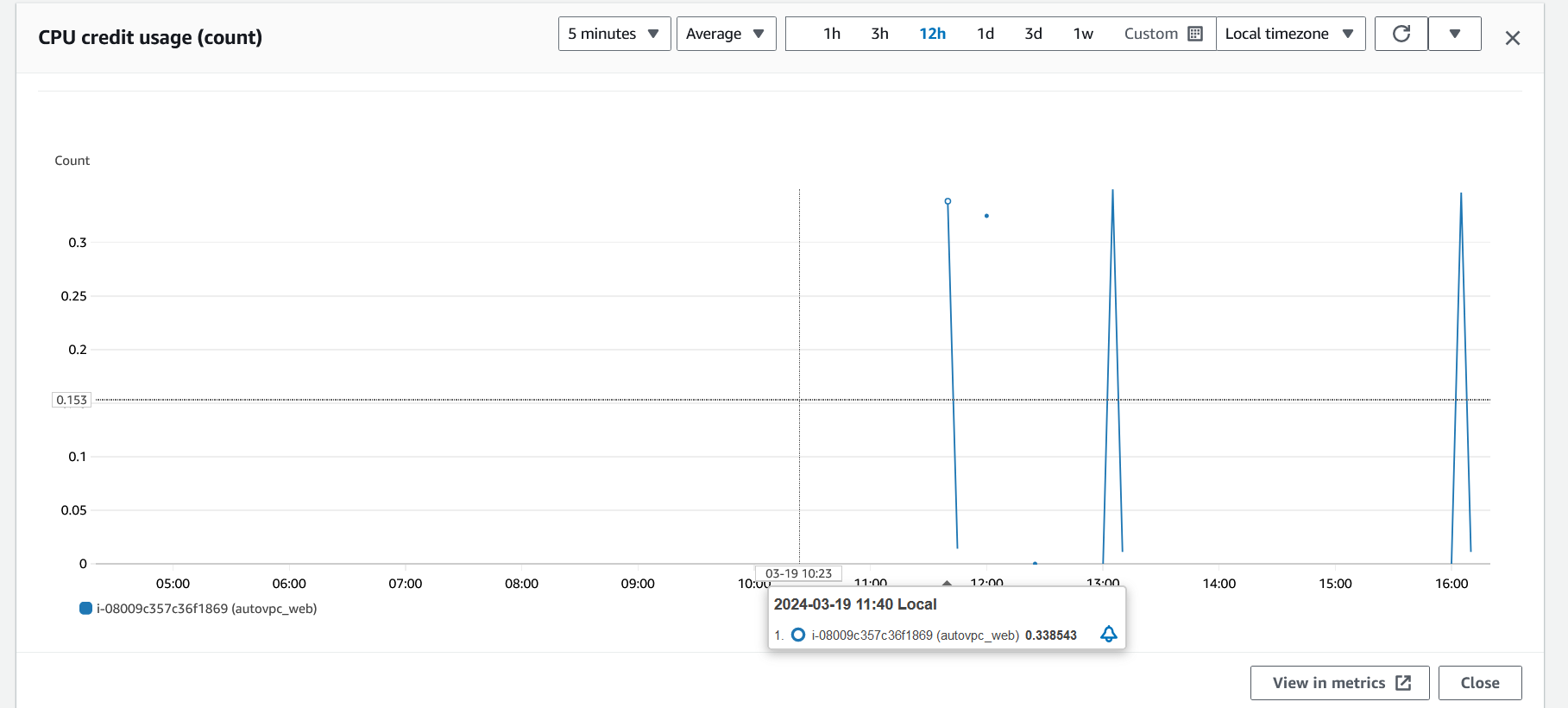
The log events and the cloudwatch monitor showing the start and stop of instances.

Start instance: 13:00

Stop instance: 13:15

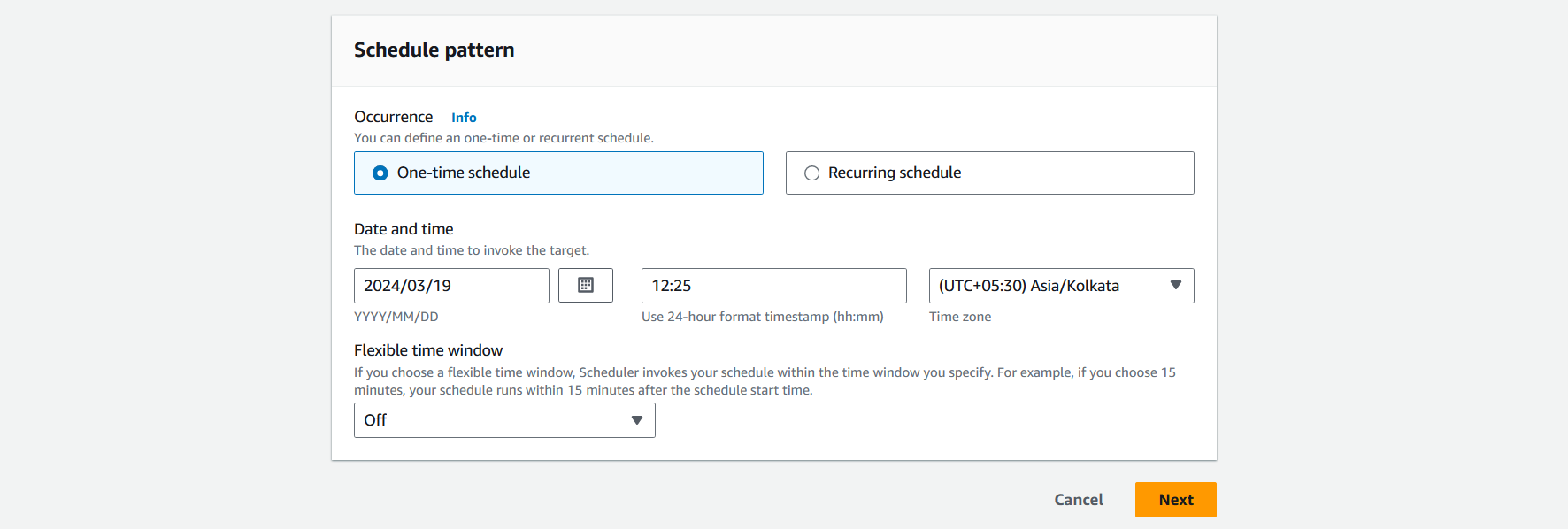
Start instance: 16:00

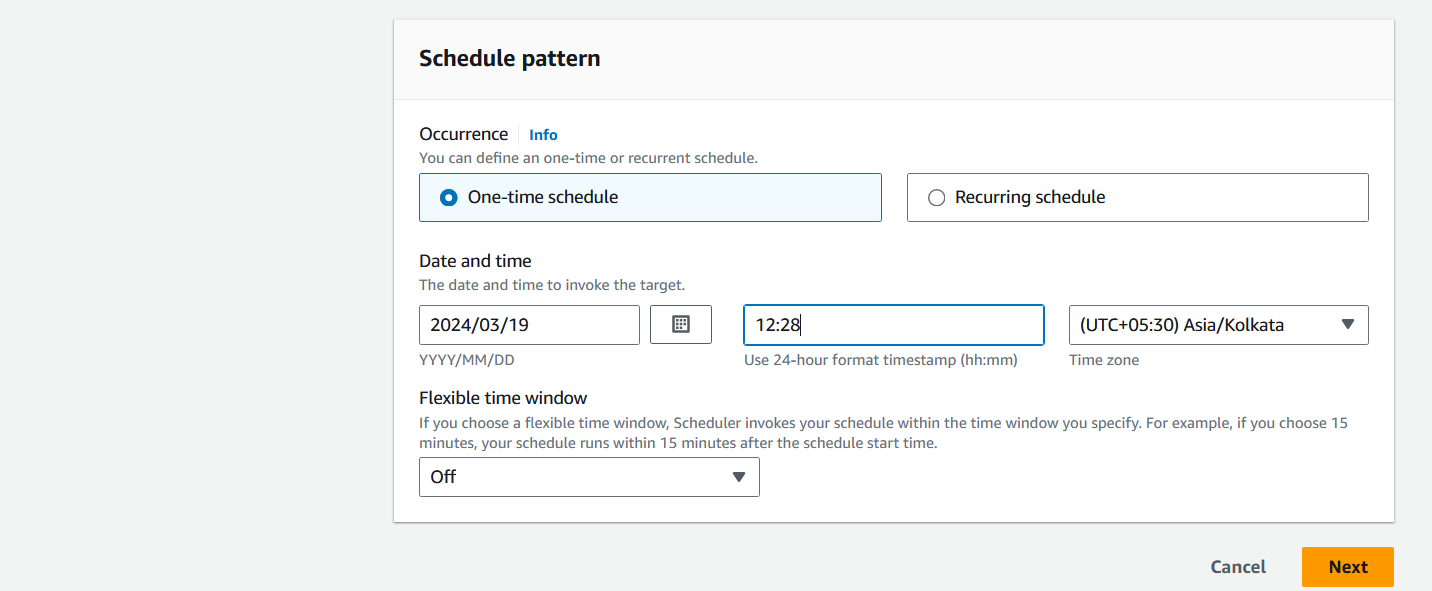
Stop instance: 16:15



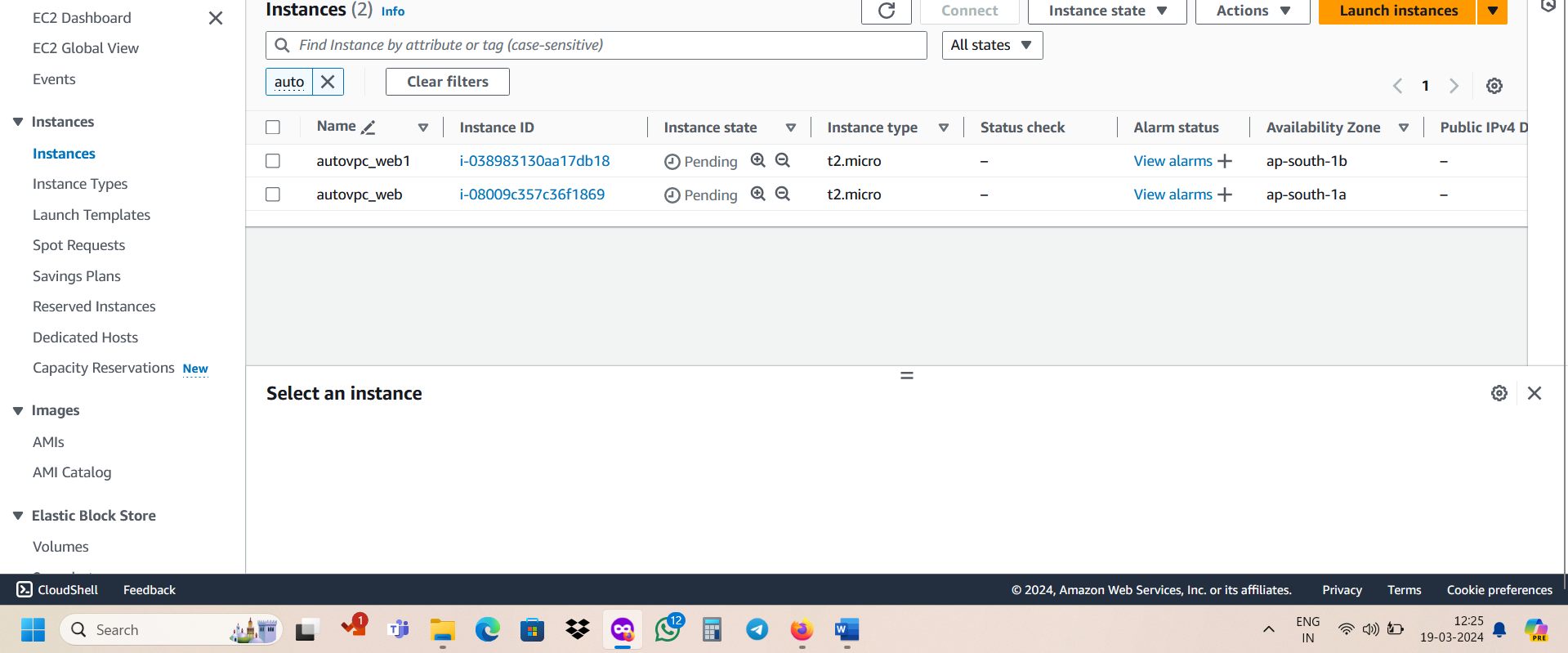
1. **Testing (Optional-To be followed before configuring for longer intervals)**

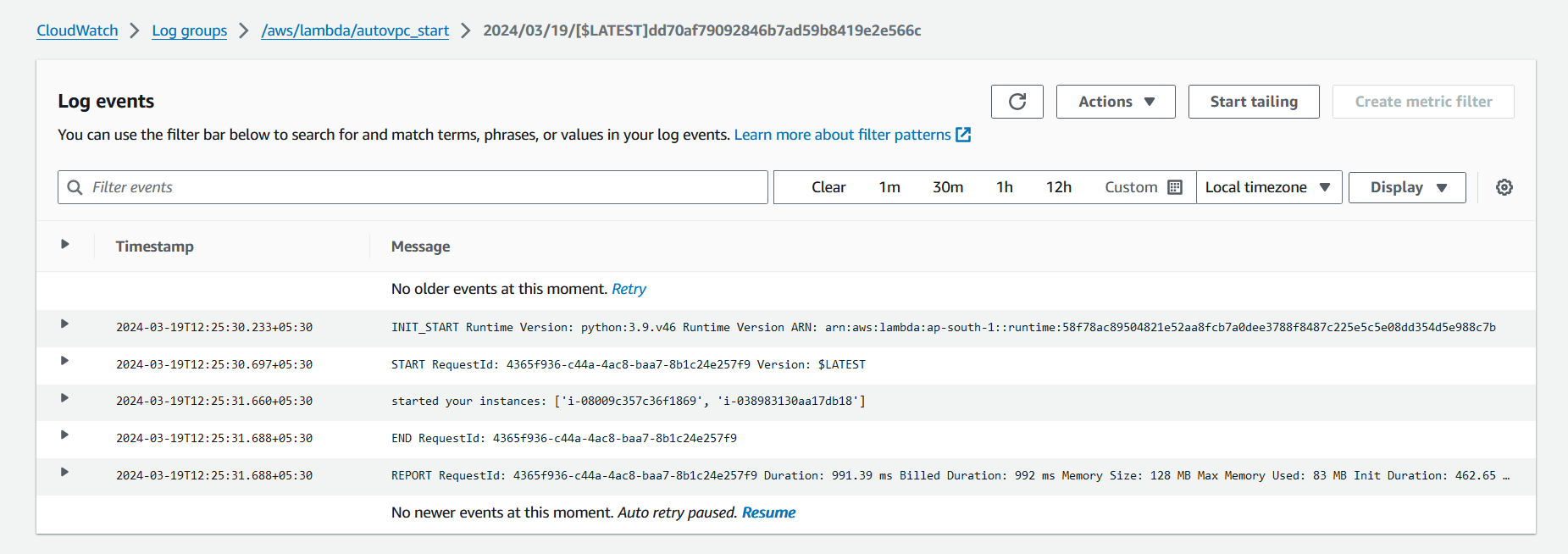
* Test your Lambda functions manually first to ensure they work as expected.
* Set up test schedules in CloudWatch Events with short intervals for testing purposes. (Start instances at 12:25 and stop instances at 12:28)

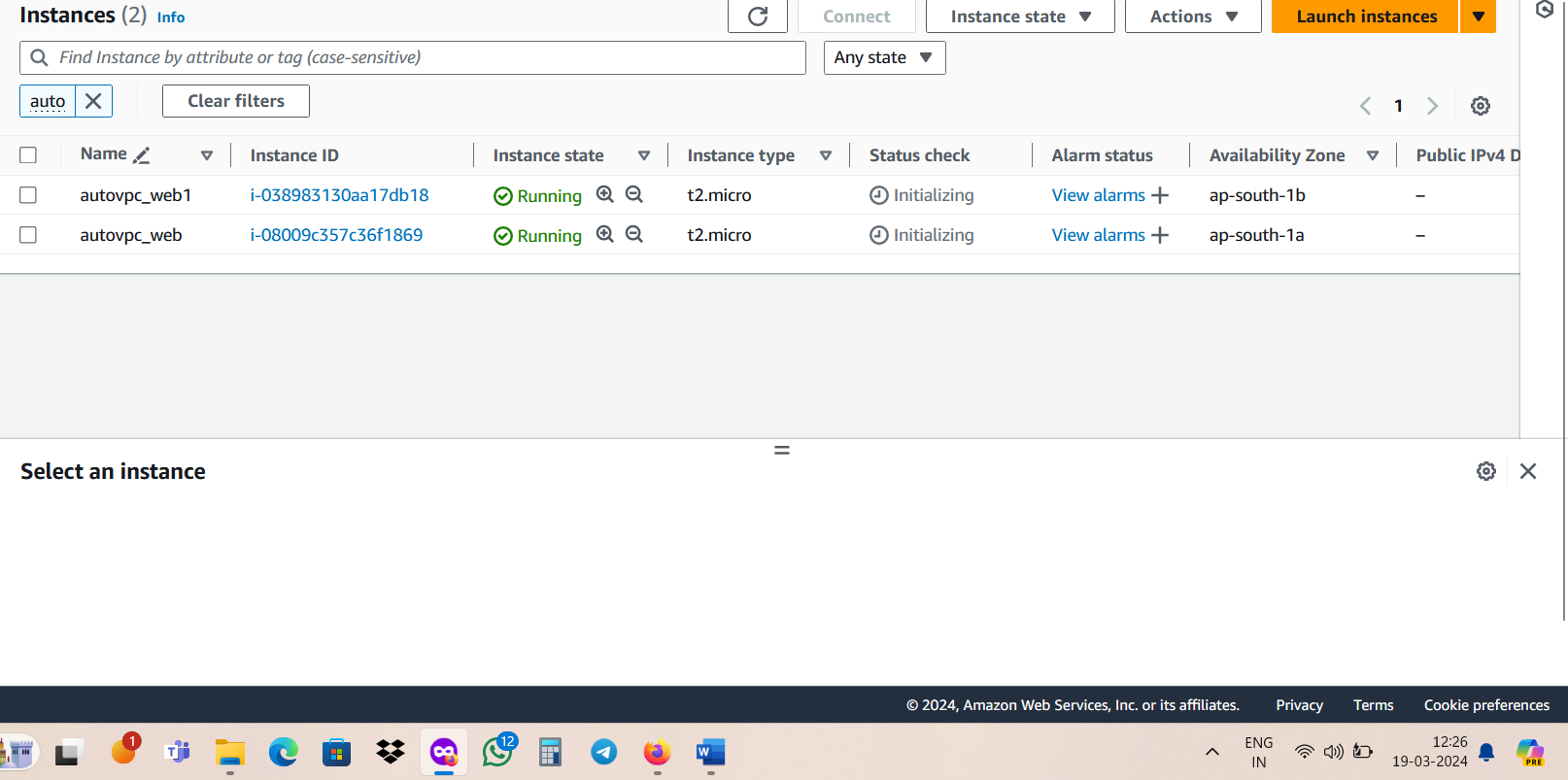


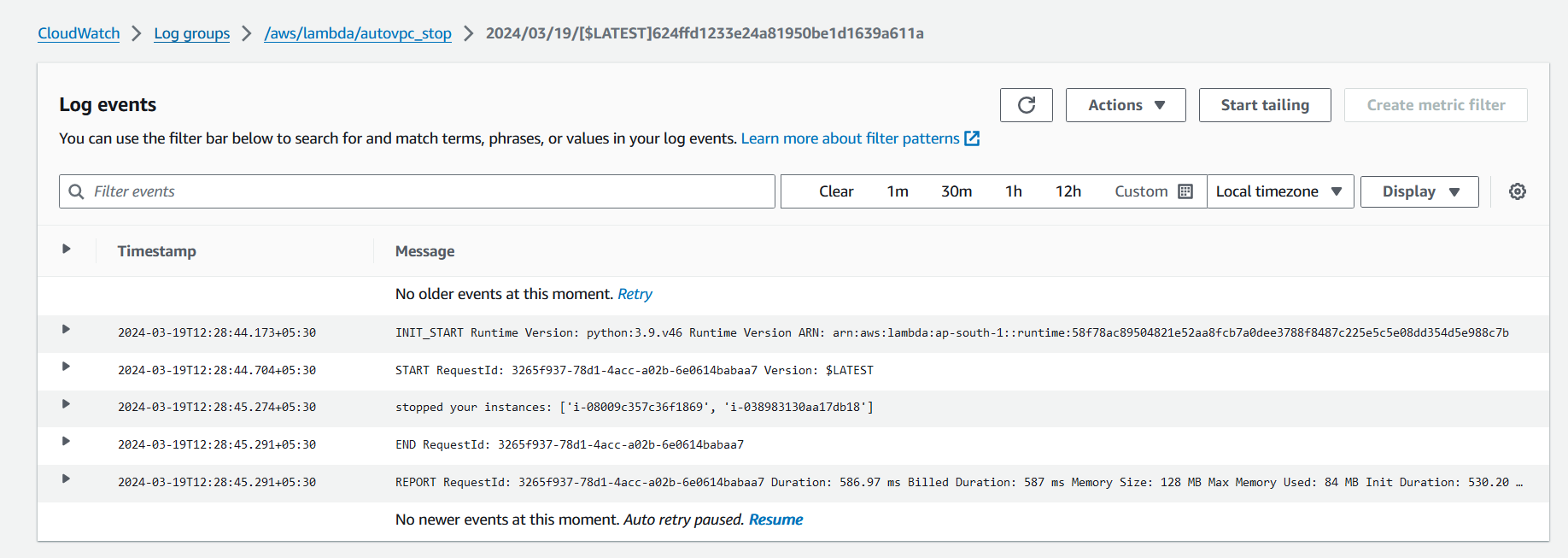


* Instances started at 12:25

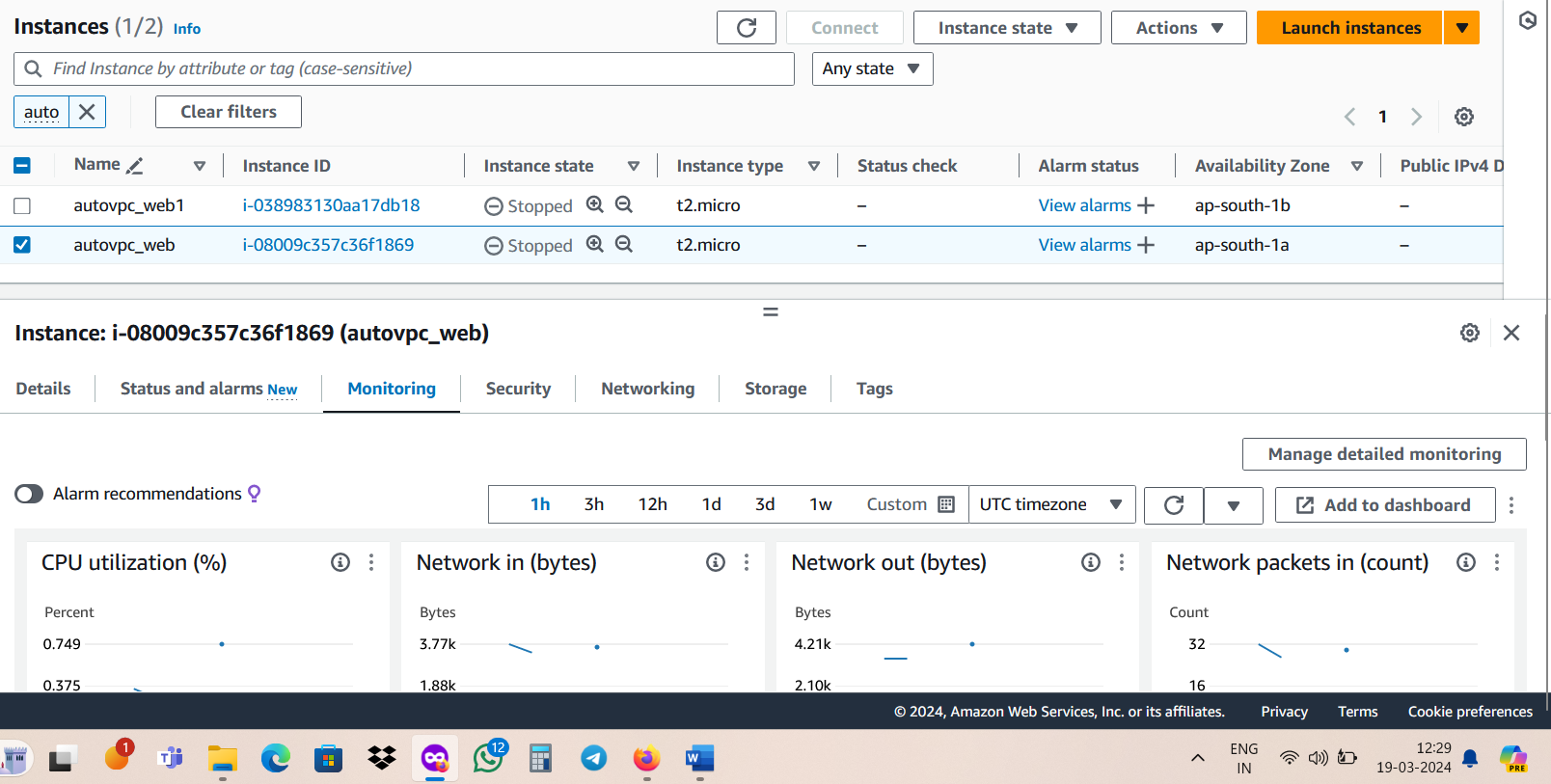


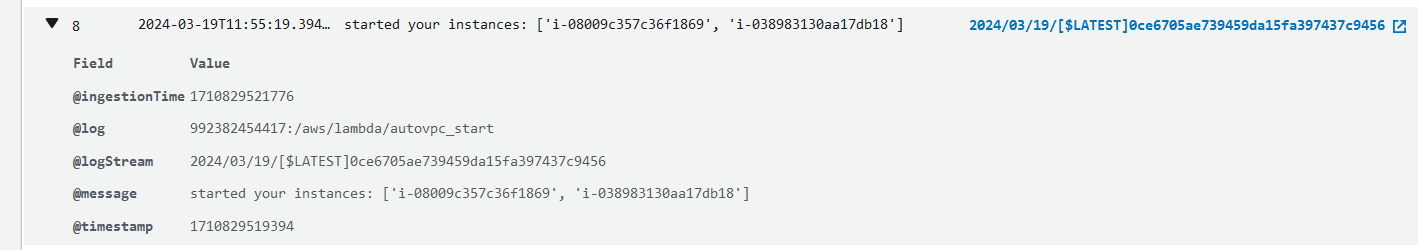
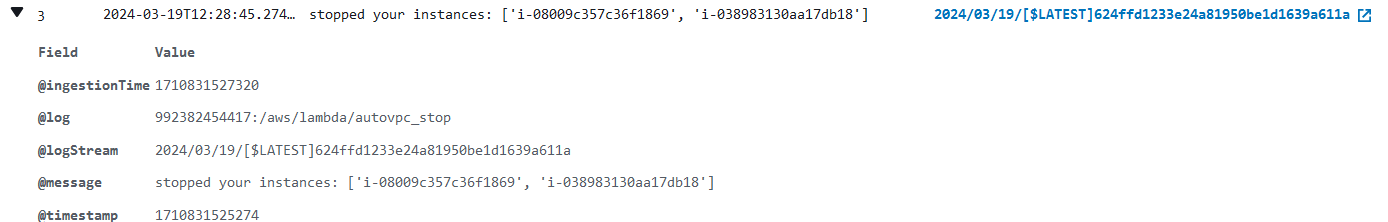






* Instances stopped at 12:28



**Points to Remember:**

1. **IAM Permissions:**
   * Ensure that the IAM role attached to your Lambda function has the necessary permissions to start and stop EC2 instances.
2. **Error Handling:**
   * Implement proper error handling in your Lambda functions to handle any issues that may arise during execution, such as failures to start or stop instances.
3. **Monitoring:**
   * Monitor CloudWatch Logs for your Lambda functions to troubleshoot any errors or unexpected behavior.
   * Use CloudWatch Alarms to alert you to any issues with the Lambda functions or EC2 instances.
4. **Cost Considerations:**
   * Be mindful of the cost implications of running Lambda functions and EC2 instances. Ensure that your setup is cost-effective and aligns with your budget.
5. **Execution Time:**
   * Keep the execution time of your Lambda functions within the maximum allowed duration (15 minutes). If your operations take longer, consider breaking them into smaller tasks or using other AWS services like Step Functions.

**Best Practices:**

1. **Separation of Concerns:**
   * Consider separating the logic for starting and stopping EC2 instances into separate Lambda functions. This improves modularity and makes it easier to manage and update your code.
2. **Tagging:**
   * Tag your EC2 instances appropriately to facilitate easier management and identification. You can use tags to specify which instances should be started or stopped by your Lambda functions.
3. **Logging and Monitoring:**
   * Use CloudWatch Logs to capture logs from your Lambda functions. Ensure that you log relevant information to help with troubleshooting and auditing.
   * Set up CloudWatch Alarms to monitor Lambda function invocations, errors, and EC2 instance status changes.
4. **Testing:**
   * Test your Lambda functions and CloudWatch Events rules thoroughly before deploying them in a production environment. Consider using staging or development environments for testing.
5. **Security:**
   * Follow AWS security best practices when configuring IAM roles and policies for your Lambda functions and EC2 instances.
   * Implement encryption and access controls as needed to protect sensitive data and resources.
6. **Documentation:**
   * Maintain clear and up-to-date documentation for your Lambda functions, CloudWatch Events rules, and EC2 instance configurations. This helps ensure that your setup is well-documented and understandable by other team members.

By following these points to remember and best practices, you can set up and manage CloudWatch Events rules and Lambda functions effectively for starting and stopping EC2 instances in your AWS environment.