Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

Experiment No. 11

Aim: To explain AWS Lambda, its workflow, various functions and create your first Lambda functions using Python/Java/Nodejs. **(LO1, LO6)**

Theory:

What is AWS Lambda?

AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. Therefore, you don't need to worry about which AWS resources to launch, or how will you manage them. Instead, you need to put the code on Lambda, and it runs. In AWS Lambda the code is executed based on the response of events in AWS services such as add/delete files in S3 bucket, HTTP request from Amazon API gateway, etc. AWS Lambda function helps you to focus on your core product and business logic instead of managing operating system (OS) access control, etc.

How does AWS Lambda work?

The following AWS Lambda example with block diagram explains the working of AWS Lambda in a few easy steps:

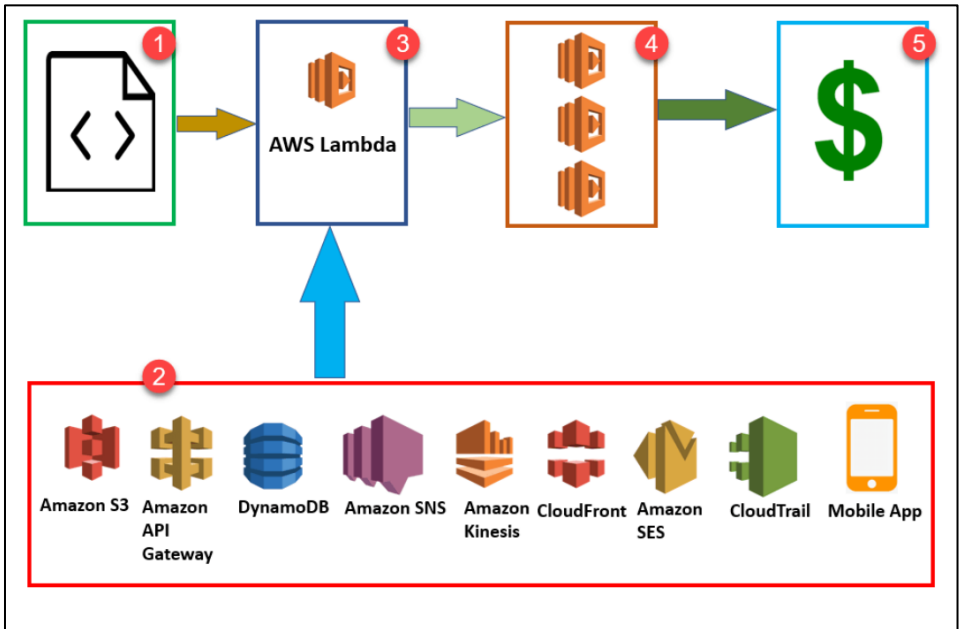
Step 1: First upload your AWS Lambda code in any language supported by AWS Lambda. Java, Python, Go, and C# are some of the languages that are supported by AWS Lambda function.

Step 2: These are some AWS services which allow you to trigger AWS Lambda.

Step 3: AWS Lambda helps you to upload code and the event details on which it should be triggered.

Step 4: Executes AWS Lambda Code when it is triggered by AWS services:

Step 5: AWS charges only when the AWS lambda code executes, and not otherwise.



AWS Lambda Block Diagram

AWS Lambda Concepts:

- **Function:** A function is a program or a script which runs in AWS Lambda. Lambda passes invocation events into your function, which processes an event and returns its response.

- **Runtimes:** Runtime allows functions in various languages which runs on the same base execution environment. This helps you to configure your function in runtime. It also matches your selected programming language.

- **Event source:** An event source is an AWS service, such as Amazon SNS, or a custom service. This triggers function helps you to executes its logic.

- **Lambda Layers:** Lambda layers are an important distribution mechanism for libraries, custom runtimes, and other important function dependencies. This AWS component also helps you to manage your development function code separately from the unchanging code and resources that it uses.

- **Log streams:** Log stream allows you to annotate your function code with custom logging statements which helps you to analyse the execution flow and performance of your AWS Lambda functions.

Use Cases of AWS Lambda:

AWS Lambda used for a wide range of applications like:

1. Helps you for ETL process
2. Allows you to perform real-time file processing and real-time stream processing
3. Use for creating web applications
4. Use in Amazon products like Alexa Chatbots and Amazon Echo/Alexa
5. Data processing (real-time streaming analytics)
6. Automated Backups of everyday tasks
7. Scalable back ends (mobile apps, loT devices)
8. Helps you to execute server-side backend logic
9. Allows you to filter and Transform data

Advantages of using AWS Lambda:

Here, are pros/benefits of using AWS lambda:

1. You can use it as a plugin for Eclipse and Visual Studio.
2. As it is serverless architecture, you don't need to worry about managing or provisioning servers.
3. Helps developers to run and execute the code's response to events without building any infrastructure.
4. You just need to for the compute time taken, only when your code runs.
5. You can monitor your code performance in real time through CloudWatch.
6. It allows you to run your code without provisioning or to manage any other server
7. You can scale it automatically to handle a few requests per day and even support more than thousands of requests per second.
8. AWS Lambda can be configured with the help of external event timers to perform scheduled tasks.
9. Lambda function in AWS should be configured with external event and timers so; it can be used for scheduling.
10. Lambda functions are stateless so that it can be scaled quickly.

Limitations of AWS Lambda:

Here are the cons/disadvantages of using AWS Lambda:

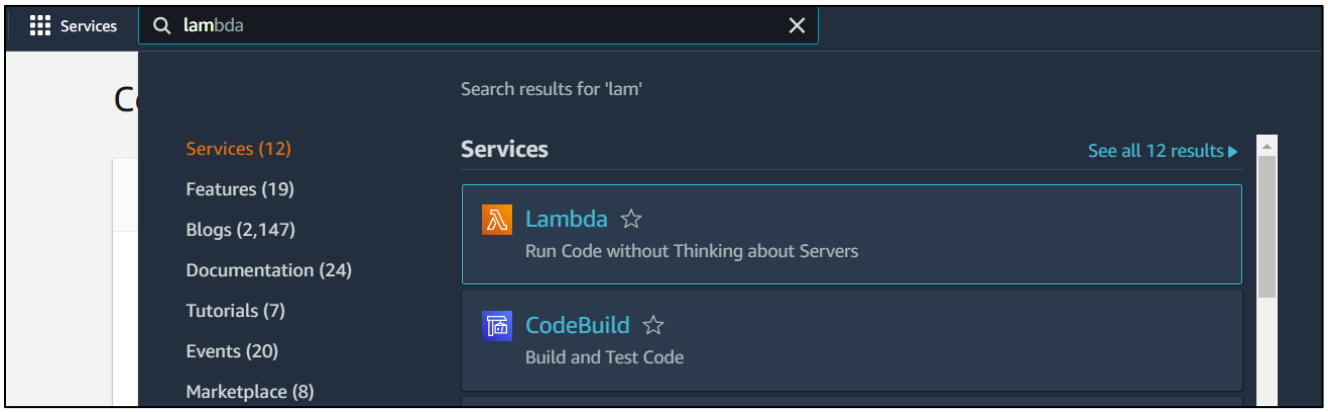
1. AWS Lambda tool is not suitable for small projects.
2. AWS Lambda entirely relies on AWS for the infrastructure, so you can't install any additional software if your code demands it.
3. Concurrent execution is limited to 100
4. AWS Lambda completely depended on AWS for the infrastructure; you cannot install anything additional software if your code demands it.
5. Its memory volume can vary between 128 to 1536 MB.
6. Event request should not exceed 128 KB.
7. Lambda functions help you to write their logs only in CloudWatch. This is the only tool that allows you to monitor or troubleshoot your functions.
8. Its code execution timeout is just 5 minutes.

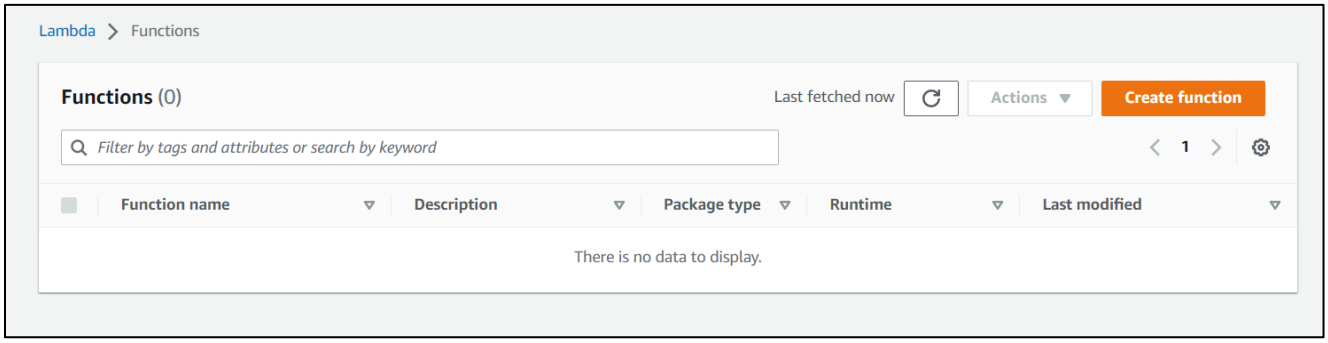
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

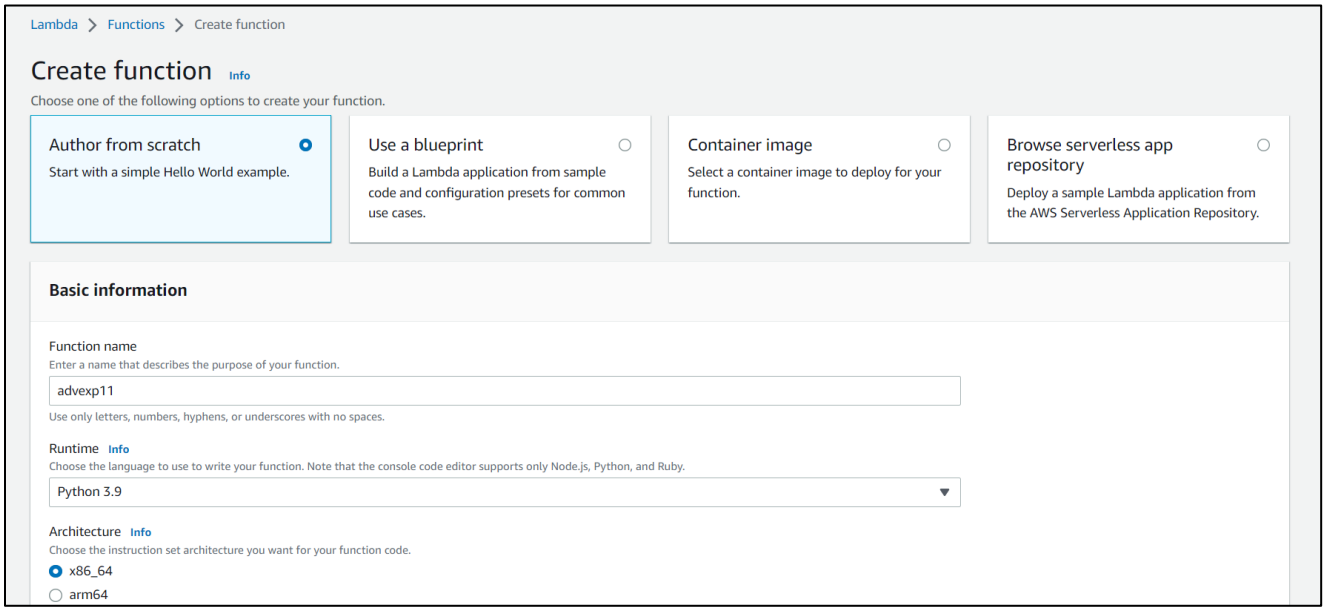
Steps to perform the Experiment:

Step1: Sign in to your AWS account and search Lambda and click on Lambda.

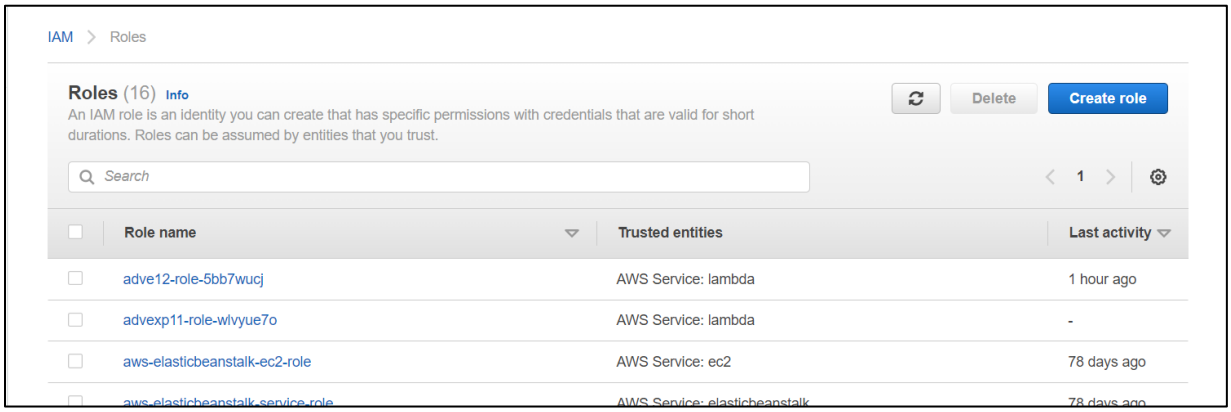


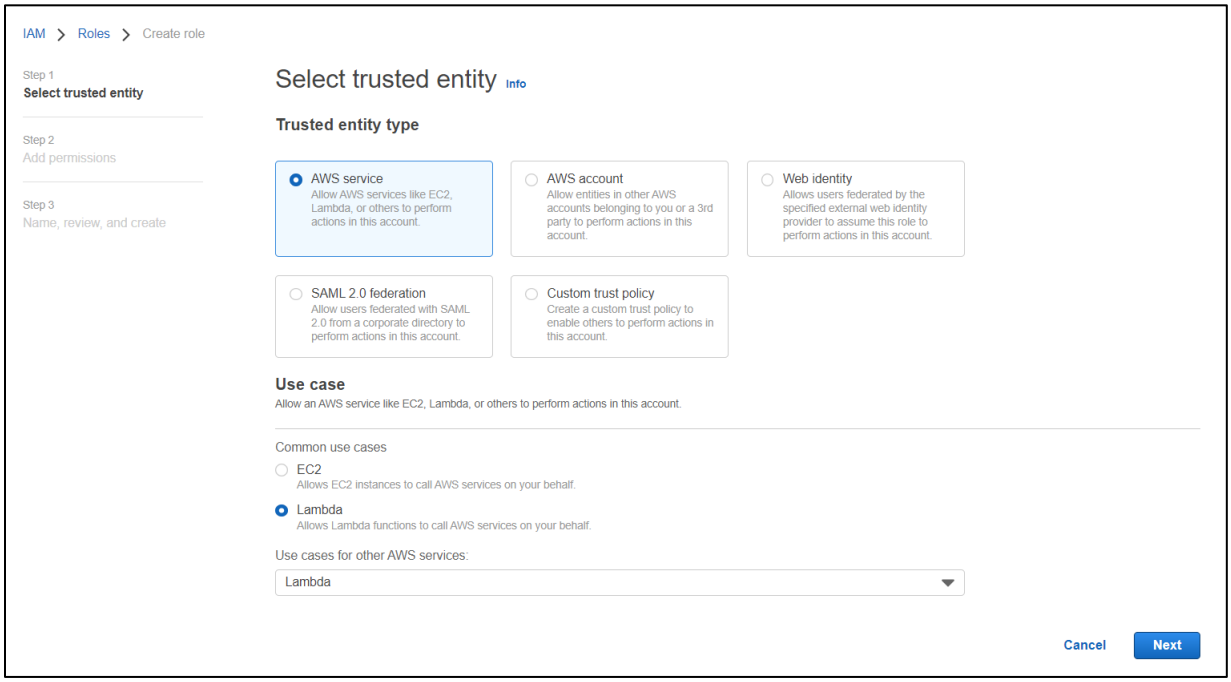


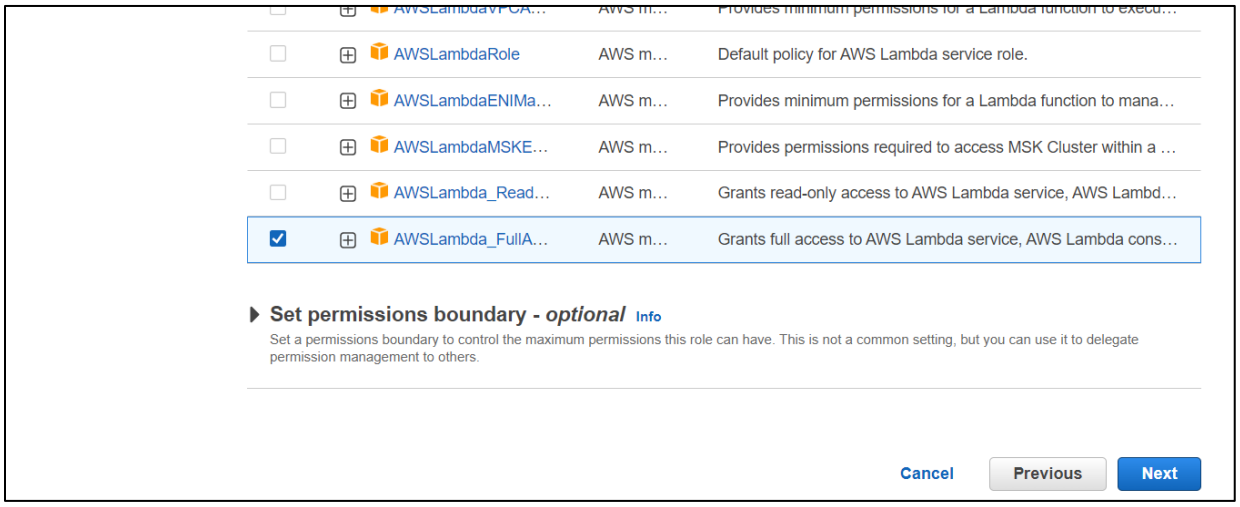
Step 2: Choose Create function.



Step 3: Go to IAM Roles and create a new role.

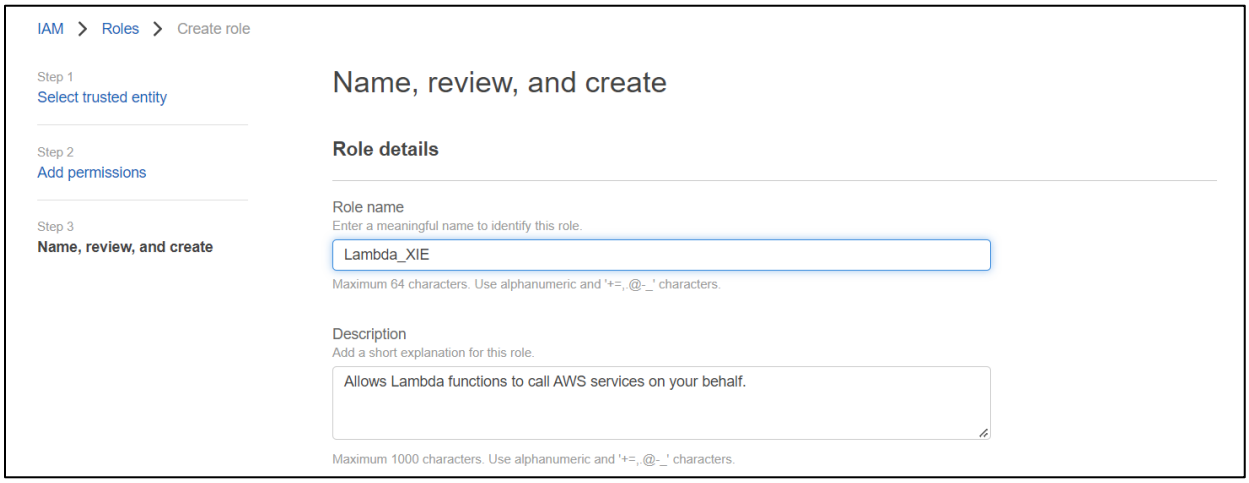


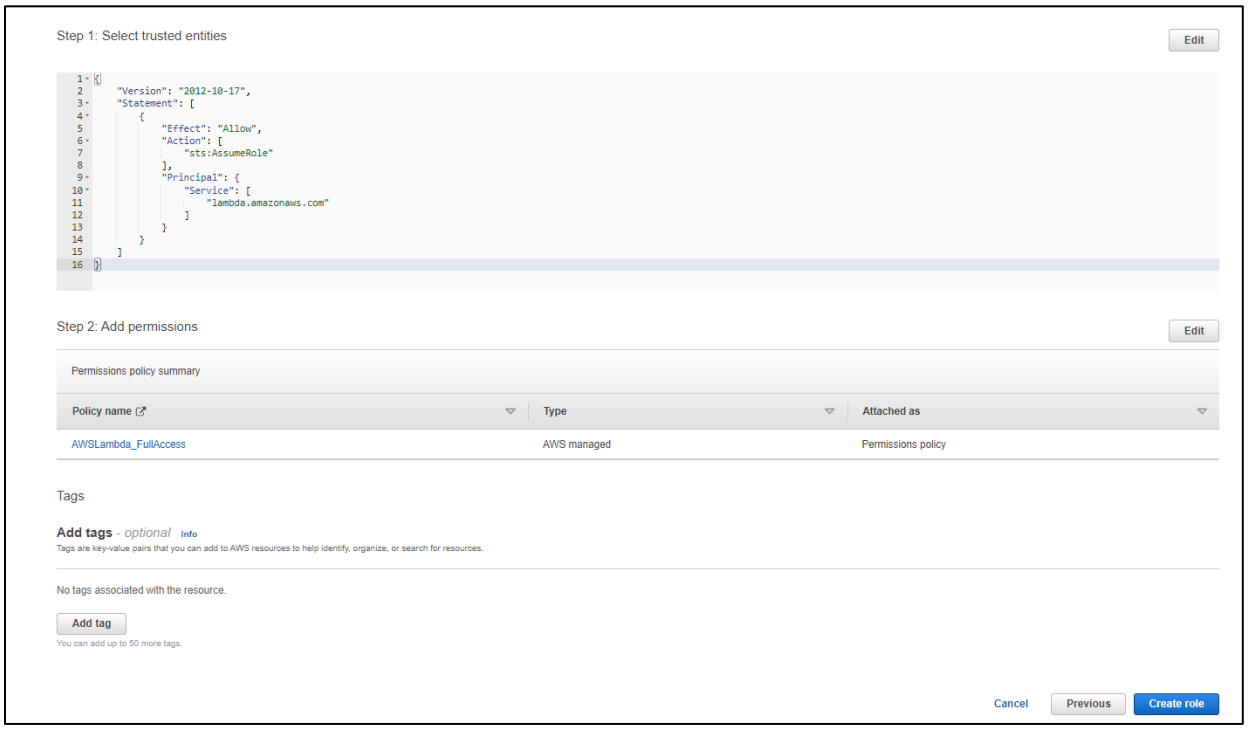


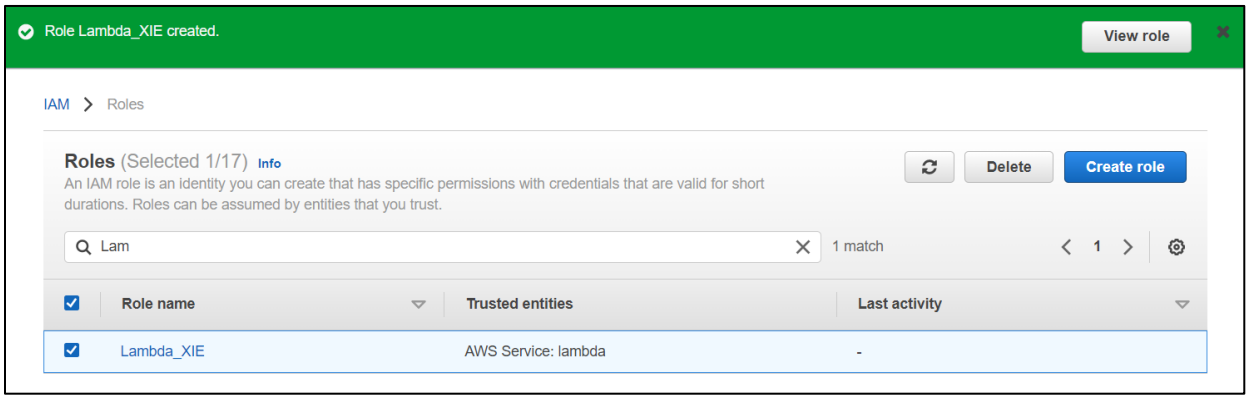


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



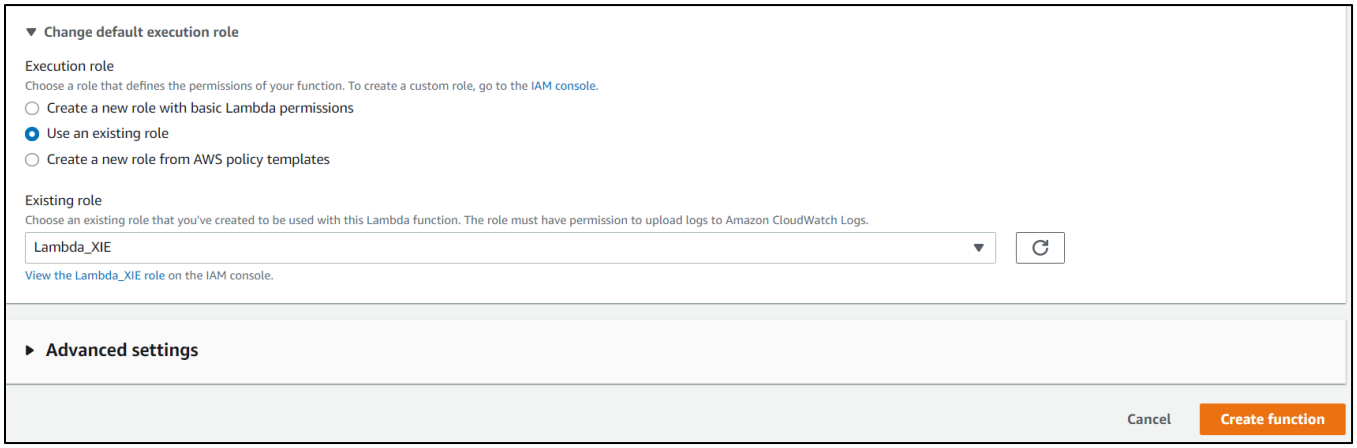




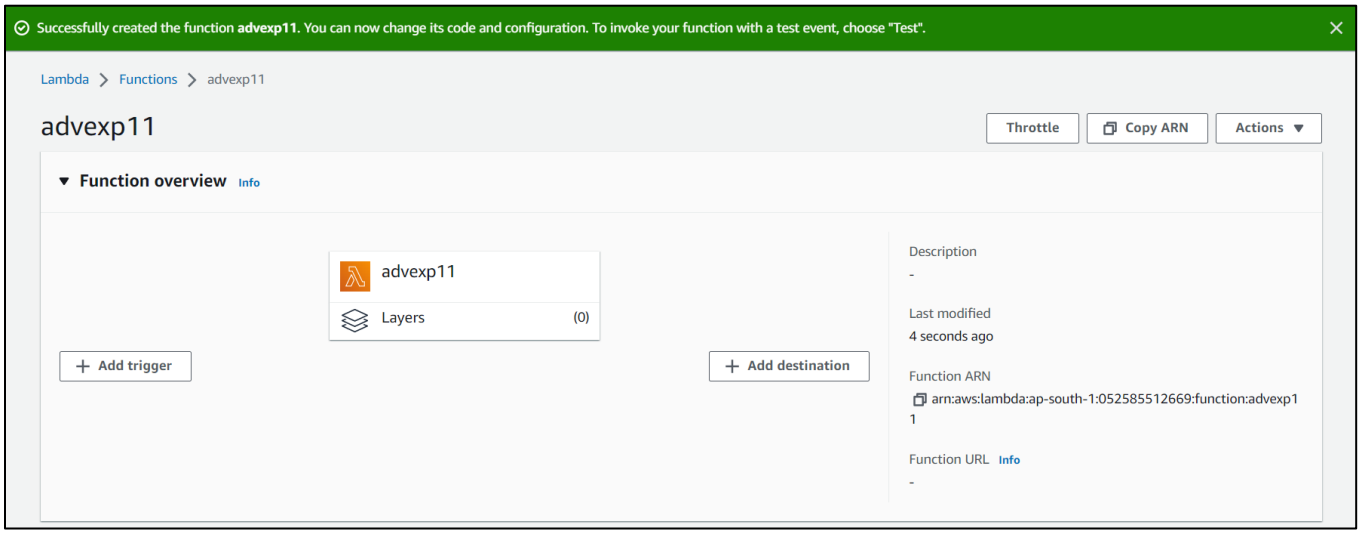
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

Step 4: Add IAM role to Lambda function.

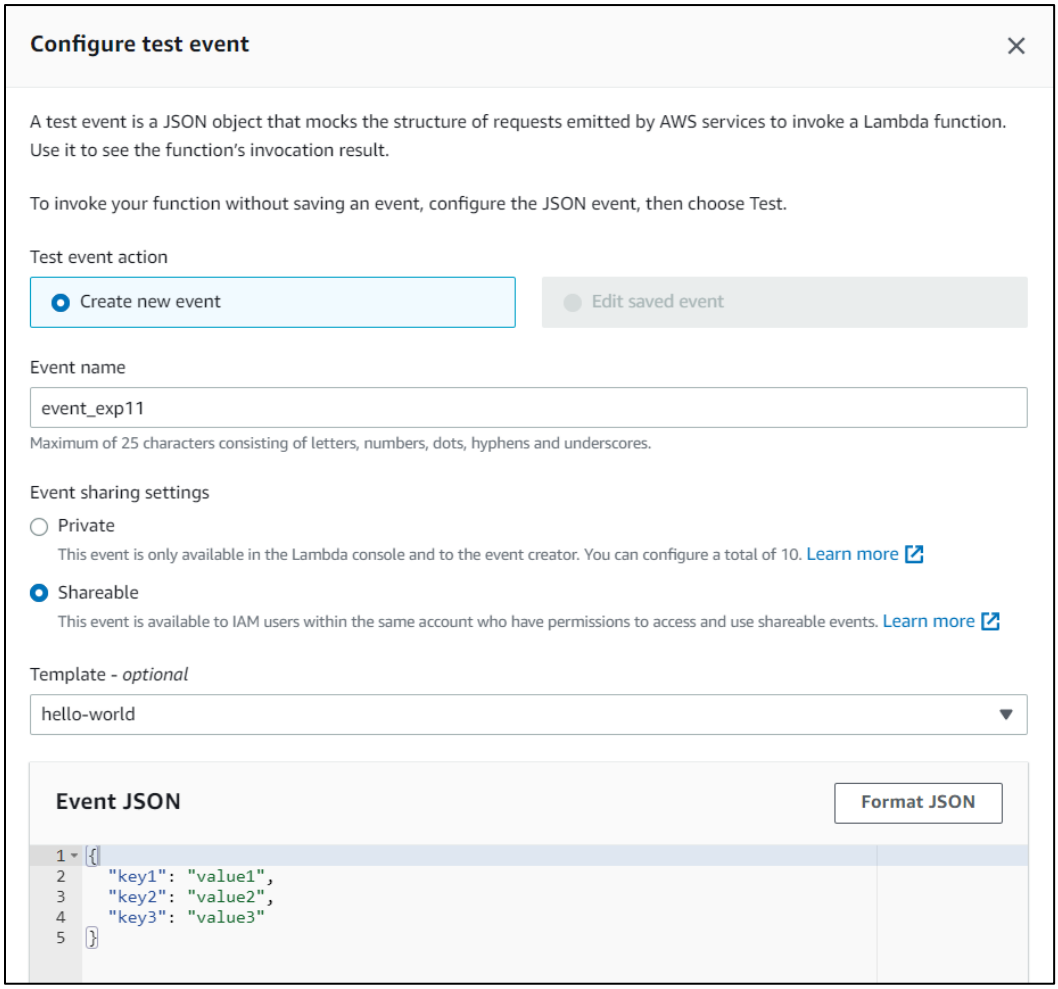


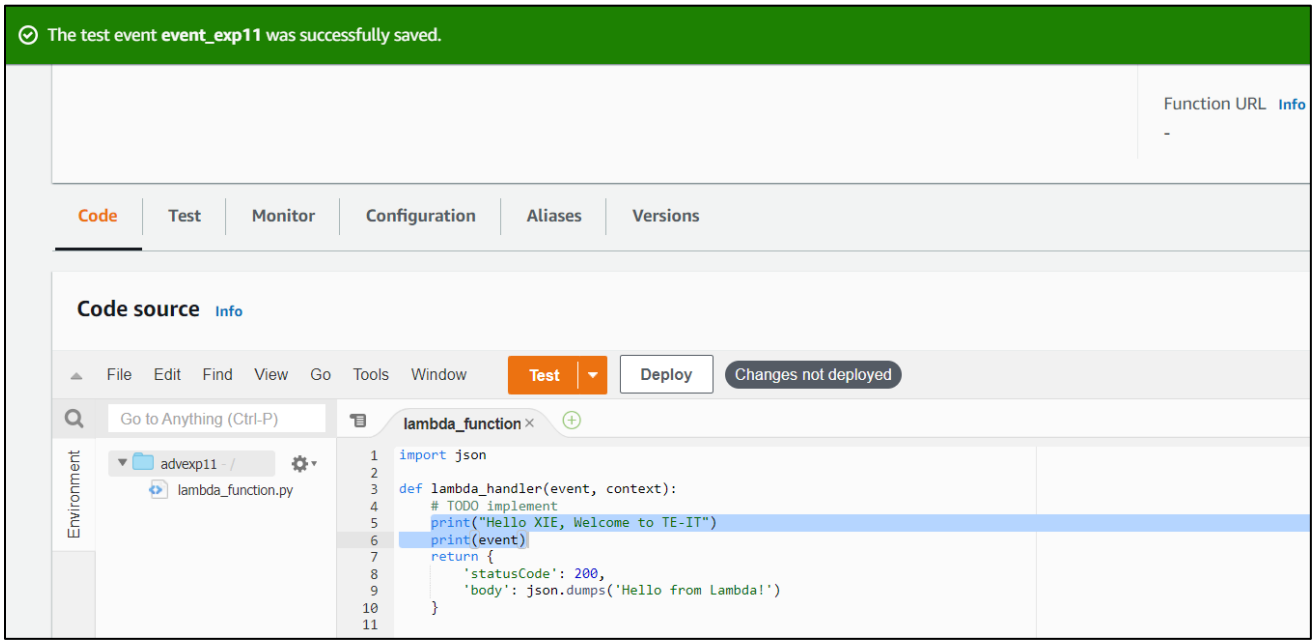
And click on Create function.



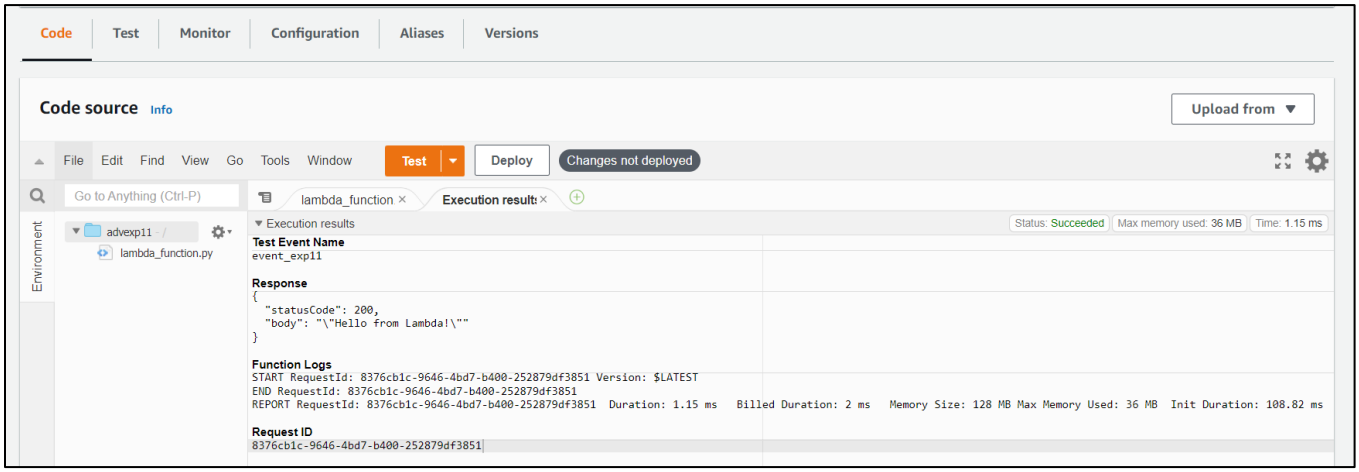
Step 5: Go to the Code section and print something.
To configure a test event, choose Test.

Step 6: For Event name, enter event_exp11.

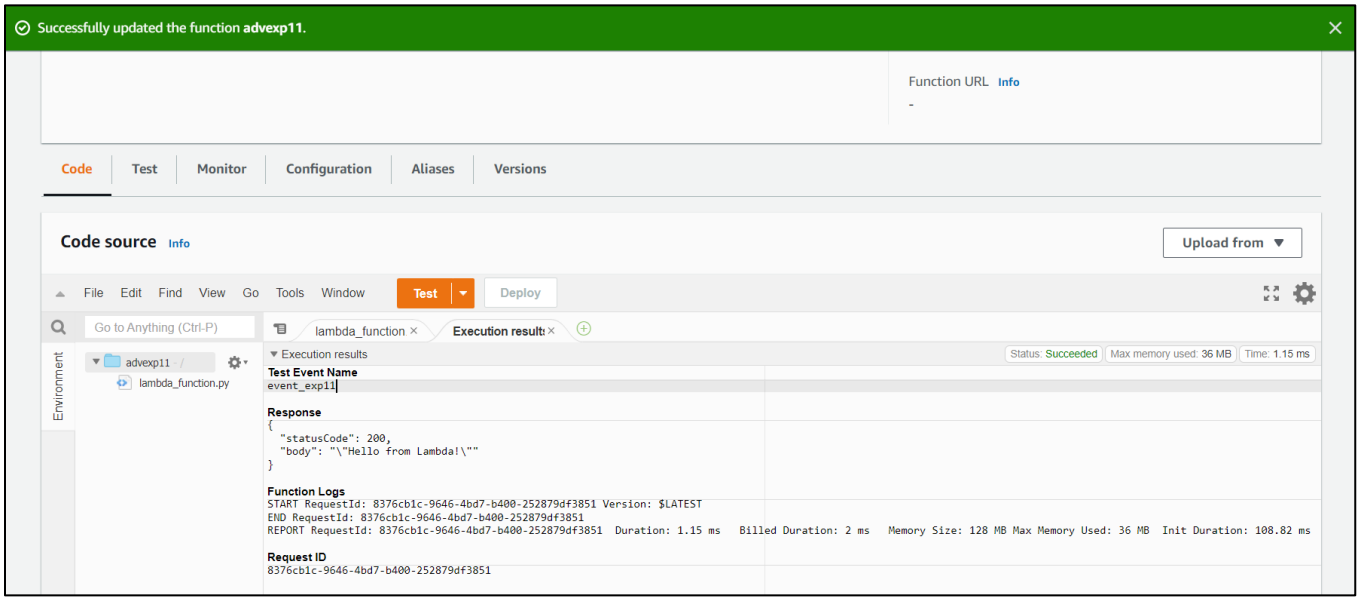




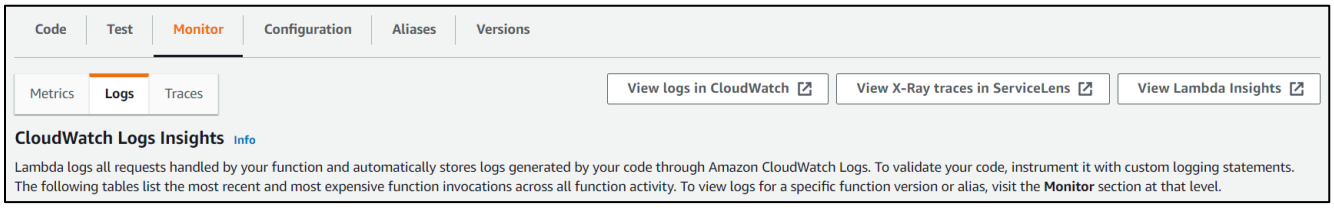
Step 7: To invoke the function, choose Test.

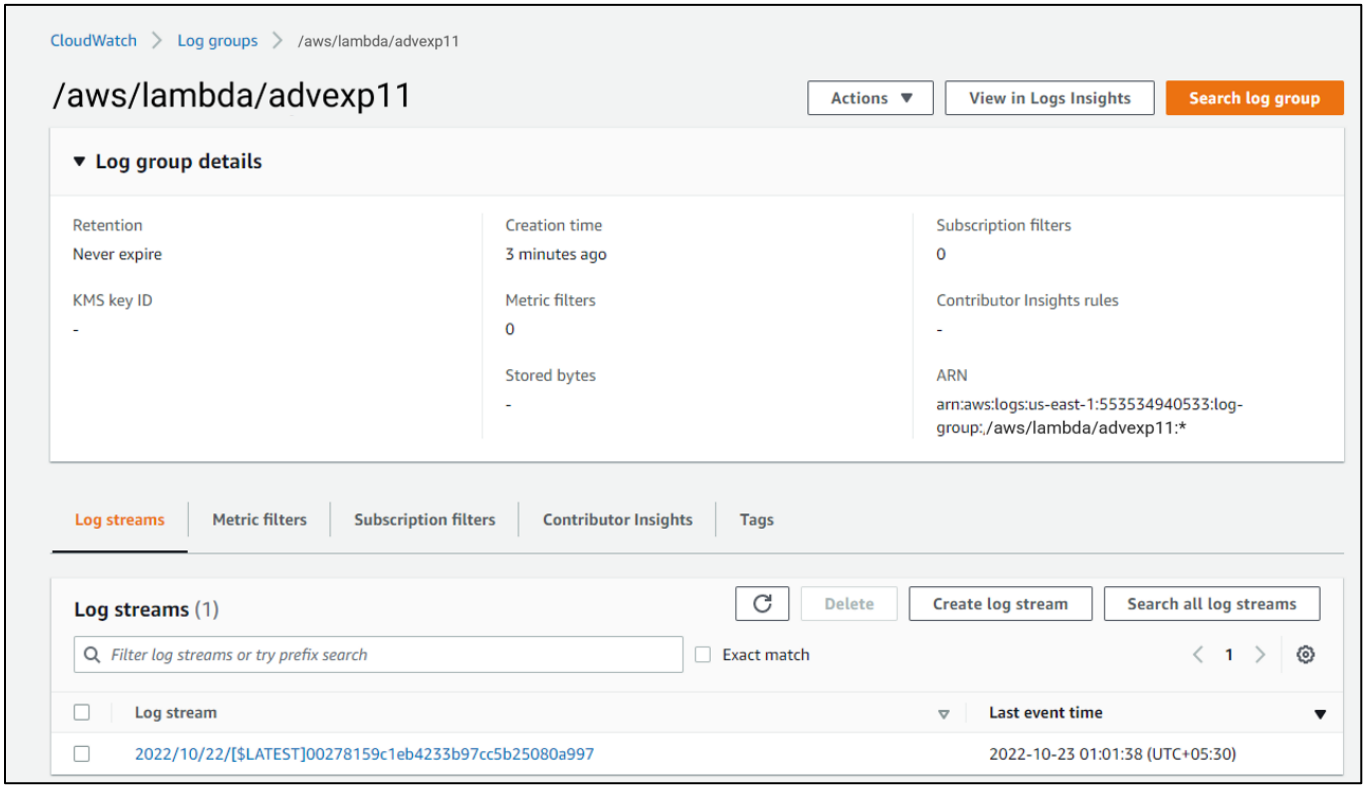


Click on Deploy.



Step 8: Now go to the Monitor section and click on View logs in CloudWatch





Conclusion: From this experiment, it is concluded that we have understood the concepts of AWS Lambda, its workflow and various functions. In this experiment, we have created Lambda functions using Python. AWS Lambda function helps us to focus on our core product and business logic instead of managing the operating system (OS) access control, OS patching, right-sizing, provisioning, scaling, etc. Hence, we have successfully achieved the Lab Outcome One and Lab Outcome Six (LO1 and LO6). Also, we have achieved PO1, PO2, PO3, PO4, PO5, PO9, PO10 and PO12 from this experiment.