Shravani Anil Patil **D15A-38**

Advanced DevOps Lab
Experiment 11

Aim: To understand AWS Lambda, its workflow, various functions and create your first Lambda functions using Python / Java / Nodejs.

Theory:

AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). Users of AWS Lambda create functions, self-contained applications written in one of the supported languages and runtimes, and upload them to AWS Lambda, which executes those functions in an efficient and flexible manner.The Lambda functions can perform any kind of computing task, from serving web pages and processing streams of data to calling APIs and integrating with other AWS services.

The concept of "serverless" computing refers to not needing to maintain your own servers to run these functions. AWS Lambda is a fully managed service that takes care of all the infrastructure for you. And so "serverless" doesn't mean that there are no servers involved: it just means that the servers, the operating systems, the network layer and the rest of the infrastructure have already been taken care of so that you can focus on writing application code.

Features of AWS Lambda

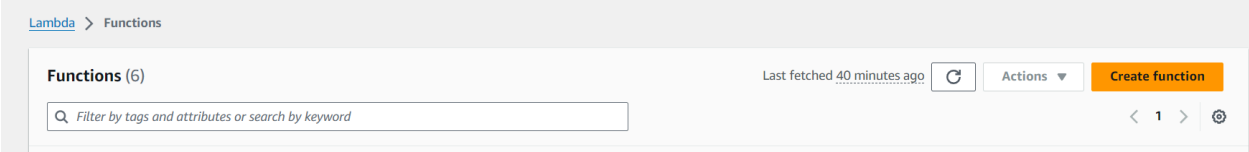
● AWS Lambda easily scales the infrastructure without any additional configuration. It reduces the operational work involved.
● It offers multiple options like AWS S3, CloudWatch, DynamoDB, API Gateway, Kinesis, CodeCommit, and many more to trigger an event.
● You don't need to invest upfront. You pay only for the memory used by the lambda function and minimal cost on the number of requests hence cost-efficient.
● AWS Lambda is secure. It uses AWS IAM to define all the roles and security policies.
● It offers fault tolerance for both services running the code and the function. You do not have to worry about the application down.

Packaging Functions

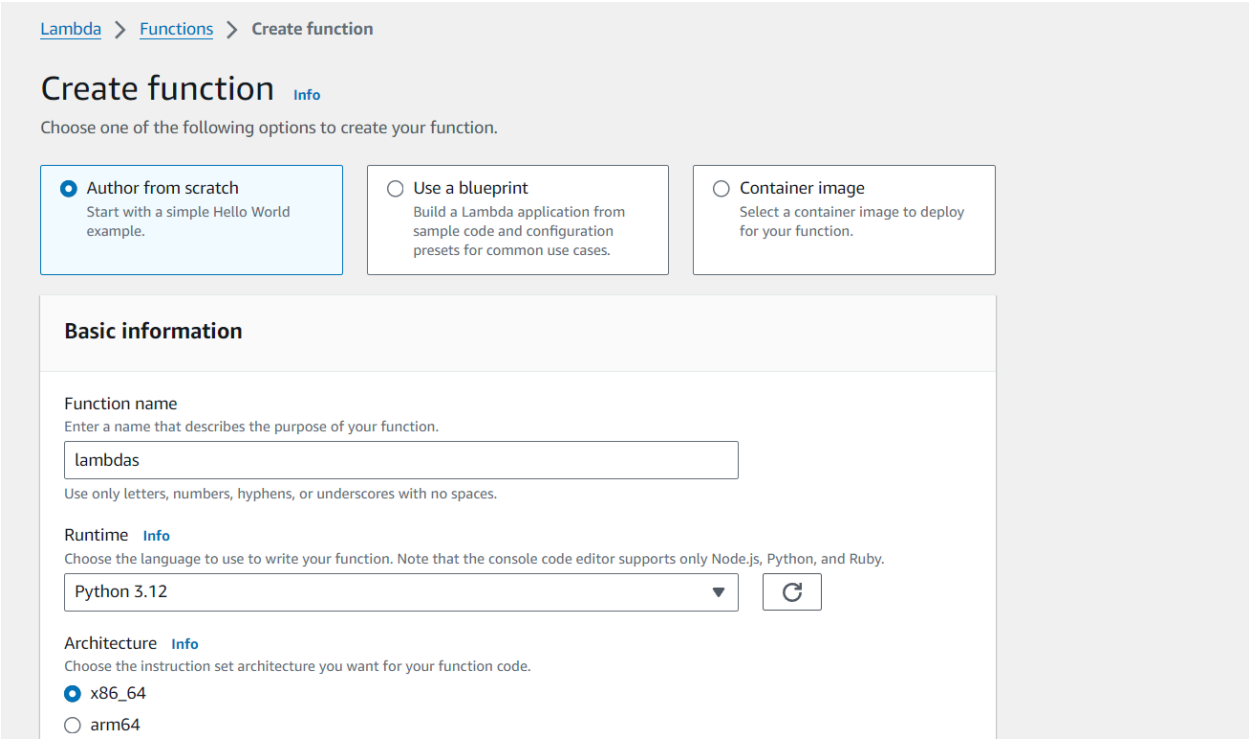
Lambda functions need to be packaged and sent to AWS. This is usually a process of compressing the function and all its dependencies and uploading it to an S3 bucket. And letting AWS know that you want to use this package when a specific event takes place. To help us with this process we use the Serverless Stack Framework (SST). We'll go over this in detail later on in this guide.

Steps to create an AWS Lambda function

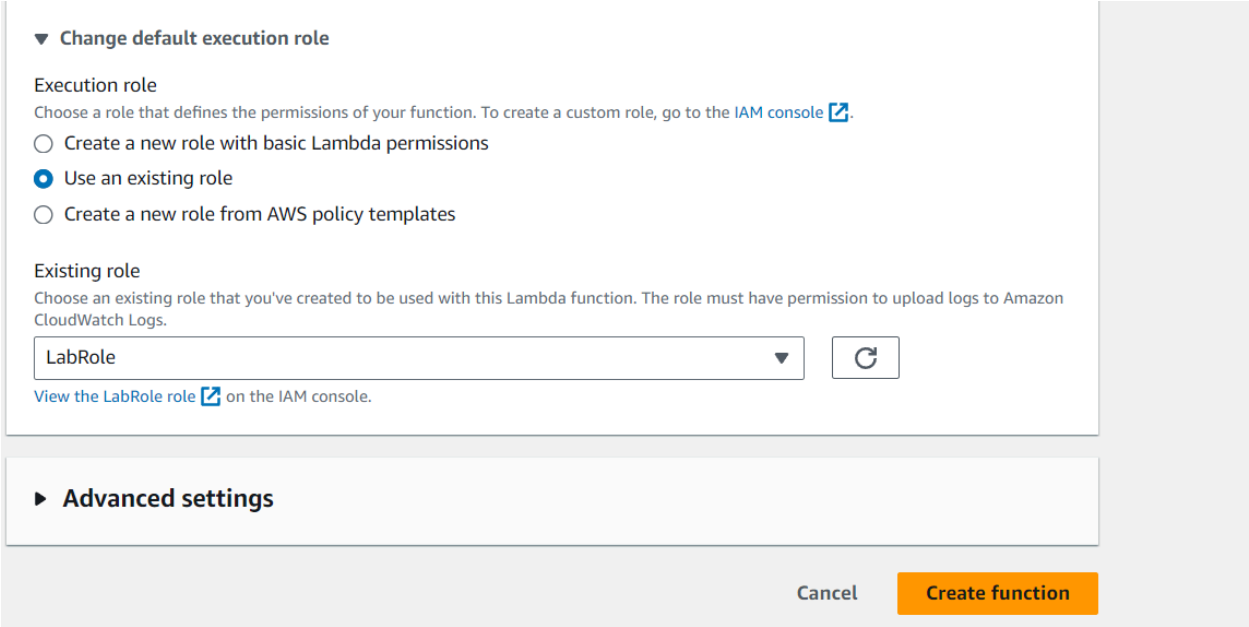
1. Open up the Lambda Console and click on the Create button.



2. Choose to create a function from scratch or use a blueprint, i.e templates defined by AWS
for you with all configuration presets required for the most common use cases.
Then,choose a runtime env for your function, under the dropdown, you can see all the options AWS supports, Python, Nodejs, .NET and Java being the most popular ones.

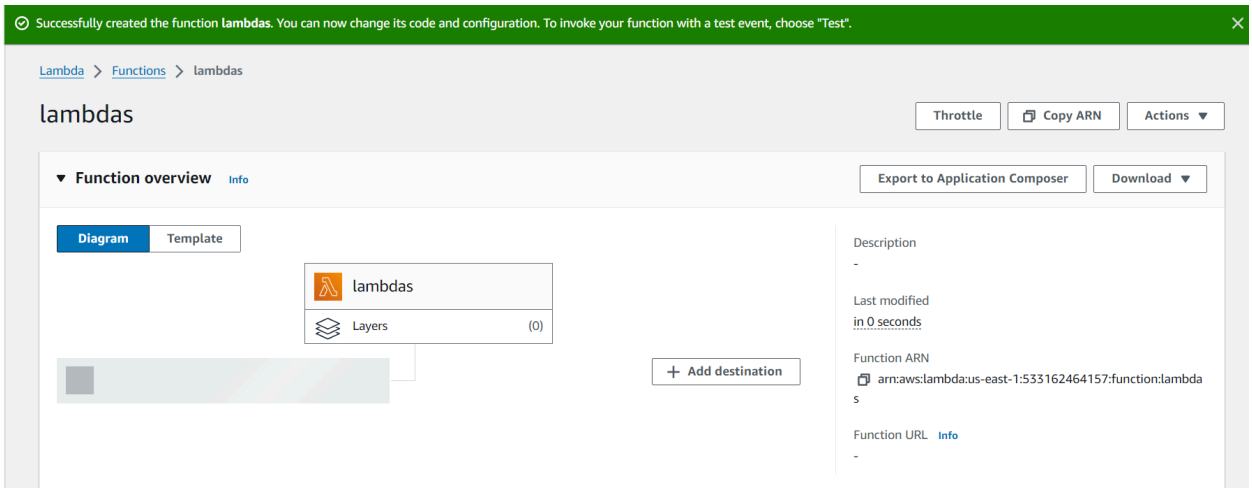


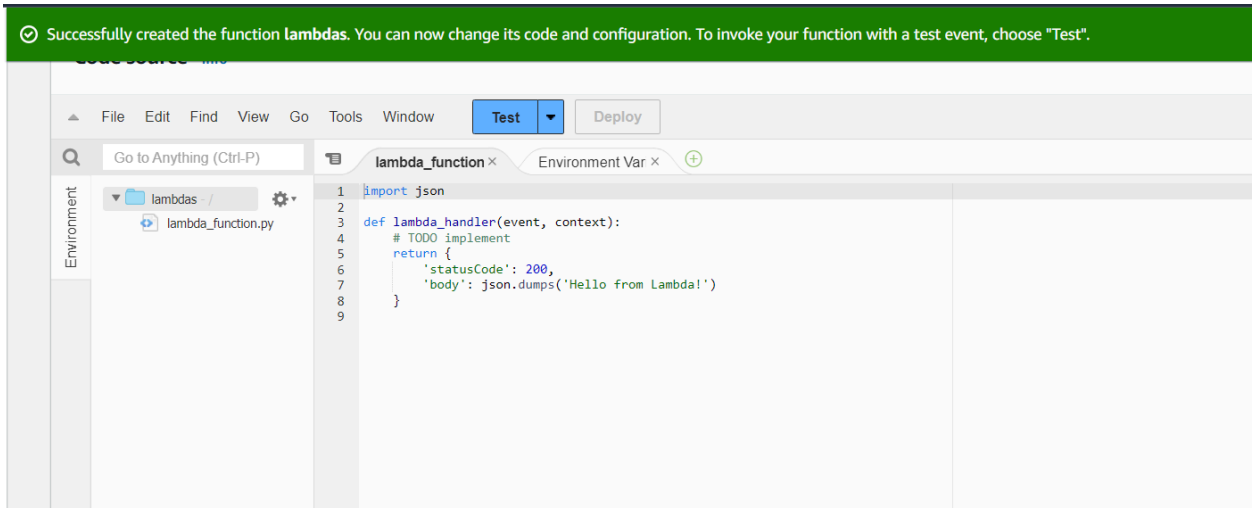
After that, choose an existing role or create a new role with basic Lambda permissions if you don't have an existing one.



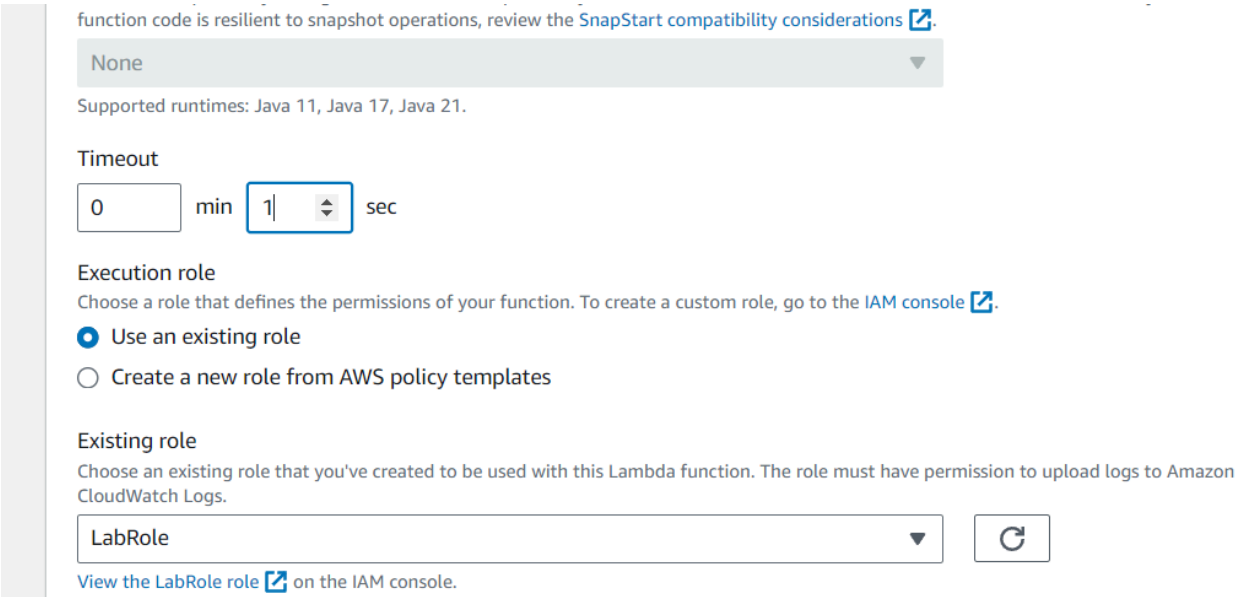
Click on the Create button.

3. This process will take a while to finish and after that, you'll get a message that your function was successfully created.

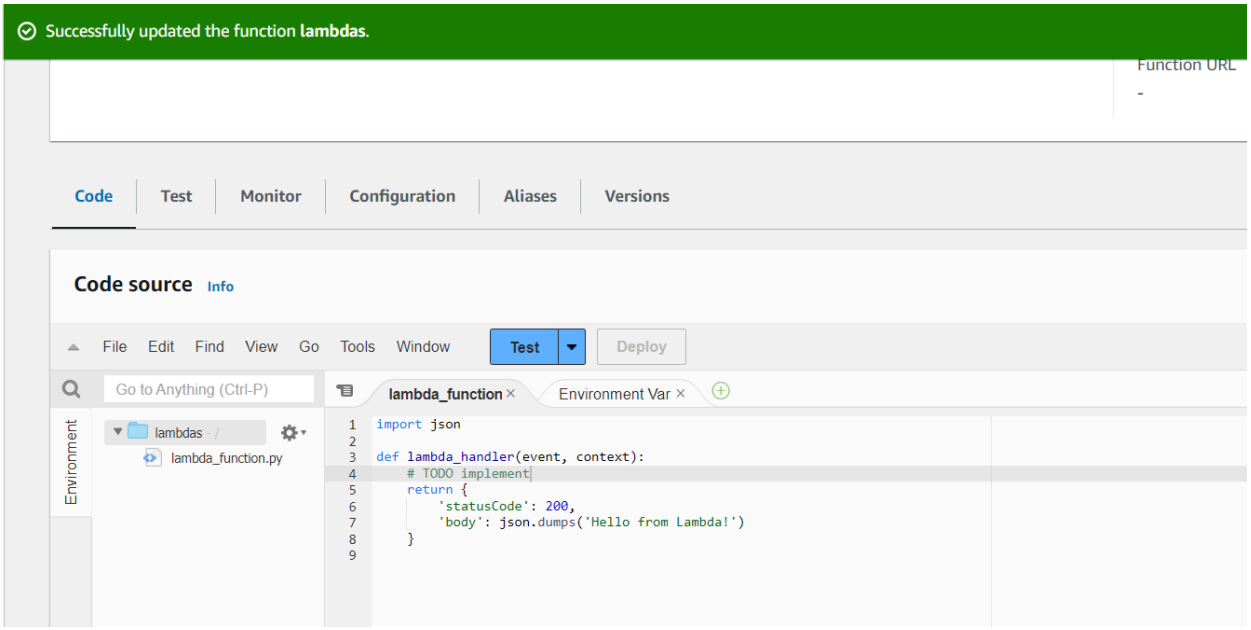




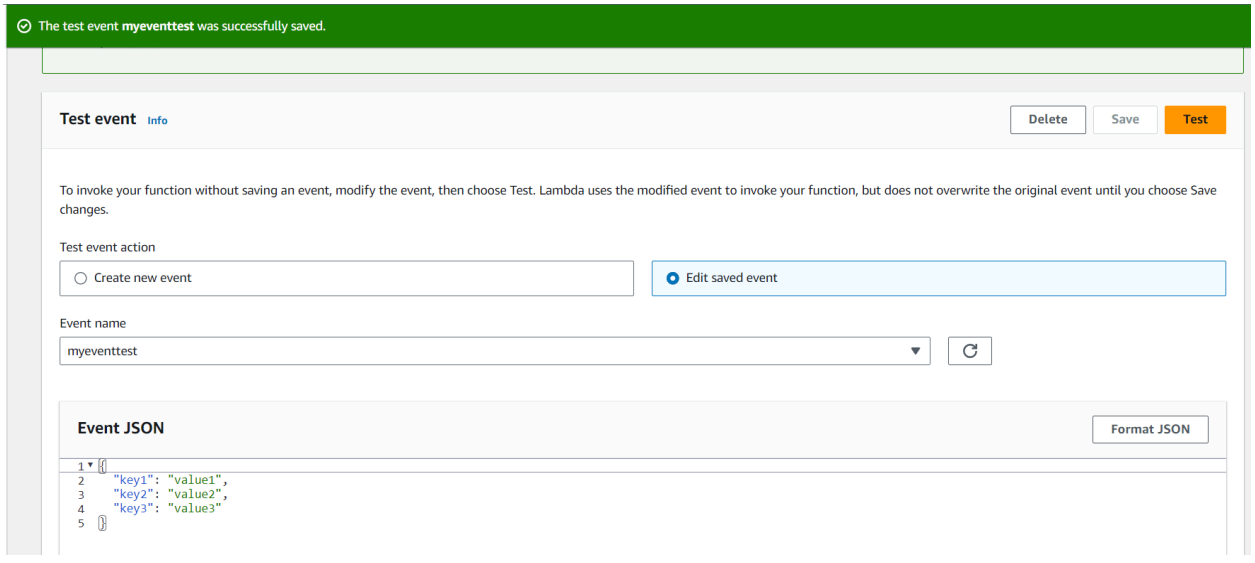
4. To change the configuration, open up the Configuration tab and under General Configuration, choose Edit.
Here, you can enter a description and change Memory and Timeout. I've changed the Timeout period to 1 sec since that is sufficient for now.



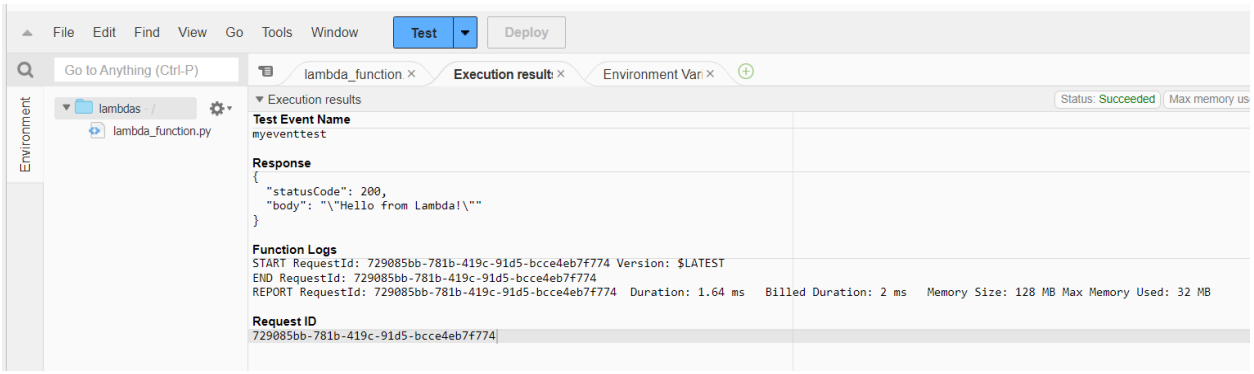
5. You can make changes to your function inside the code editor. You can also upload a zip file of your function or upload one from an S3 bucket if needed.
Press Ctrl + S to save the file and click Deploy to deploy the changes.



6. Click on Test and you can change the configuration, like so. If you do not have anything in the request body, it is important to specify two curly braces as valid JSON, so make sure they are there.



7. Now click on Test and you should be able to see the results.



Conclusion:

In conclusion, AWS Lambda offers an excellent opportunity for students to implement event-driven applications without the hassle of managing server infrastructure. By using Lambda, students can focus on developing their coding skills while creating scalable and cost-effective applications.

For instance, students can create projects that automatically process data when a file is uploaded to an Amazon S3 bucket, such as image processing or data analysis, reinforcing their understanding of real-time data handling. The straightforward workflow of defining functions, configuring triggers, and deploying applications allows students to quickly bring their ideas to life.

Additionally, with support for popular programming languages like Python, Java, and Node.js, students can choose the language they are most comfortable with or explore new ones. This hands-on experience with AWS Lambda not only enhances their technical skills but also prepares them for modern software development practices in a cloud-based environment. Overall, AWS Lambda is a valuable tool for students looking to build innovative, efficient, and scalable applications in their learning journey.