😊 AWS Lambda

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It allows you to run code without provisioning or managing servers. With Lambda, you can upload your code and AWS will take care of everything required to run and scale your code with high availability. It's often used for event-driven applications and for executing code in response to triggers such as changes in data, HTTP requests, or timers.

Lambda supports multiple programming languages including Node.js, Python, Java, and more recently, .NET Core and Go. You only pay for the compute time that you consume, measured in milliseconds, so it's cost-effective for applications with irregular or infrequent workloads. Lambda integrates with many other AWS services, making it a versatile tool for building various types of applications.

😄 Use cases of AWS Lambda

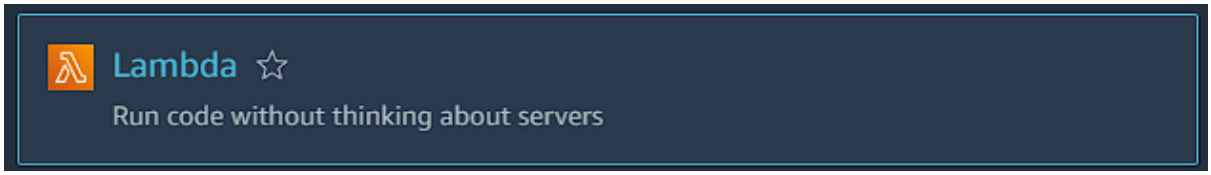
AWS Lambda has a wide range of use cases due to its flexibility, scalability, and cost-effectiveness. Here are some common scenarios where AWS Lambda is often used:

1. **Event-driven processing:** Lambda functions can be triggered by various AWS services such as S3 (object uploads), DynamoDB (database changes), SNS (notification messages), Kinesis (streaming data), and more. This makes it ideal for building event-driven architectures where you want to react to changes or events in your system.
2. **Backend for web and mobile applications:** Lambda functions can serve as the backend logic for web and mobile applications. For example, you can use Lambda to handle API requests, authenticate users, process data, and interact with databases.
3. **Real-time data processing:** Lambda can process streaming data in real-time, allowing you to perform analytics, transformations, and aggregations on data as it arrives. This is useful for building real-time dashboards, monitoring systems, and IoT applications.
4. **Scheduled tasks and cron jobs:** Lambda functions can be scheduled to run at specific intervals using CloudWatch Events. This is useful for running periodic tasks such as data backups, log rotation, or cleanup jobs.
5. **File and data processing:** Lambda can process files stored in S3, perform data validation, transformation, and enrichment, and then store the results back in S3 or other storage services.
6. **Machine learning inference:** Lambda can be used to deploy and run machine learning models for inference. You can invoke Lambda functions with input data, and the function can execute the model inference and return the results.
7. **Chatbots and conversational interfaces:** Lambda can power the backend logic of chatbots and conversational interfaces. It can process user messages, invoke natural language processing (NLP) services, retrieve data from backend systems, and generate responses.
8. **Microservices architecture:** Lambda functions can be used to implement individual microservices within a larger application architecture. Each function can be

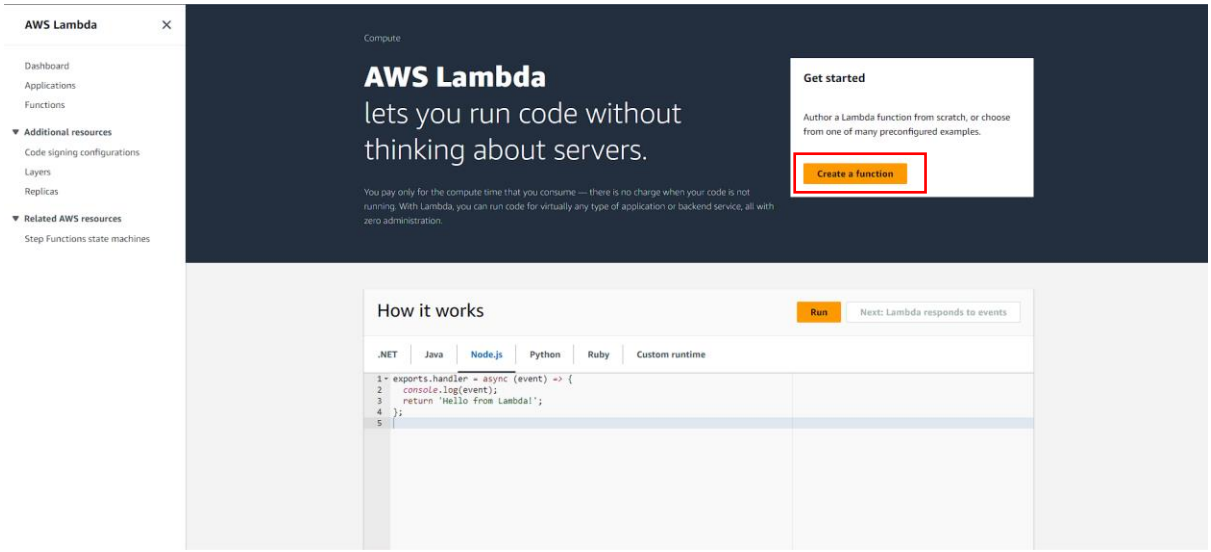
responsible for a specific task or functionality, making it easier to manage and scale the application.

😄 To begin with the Lab:

1. Login to AWS Console. Then navigate to AWS Lambda. Choose this service accordingly.

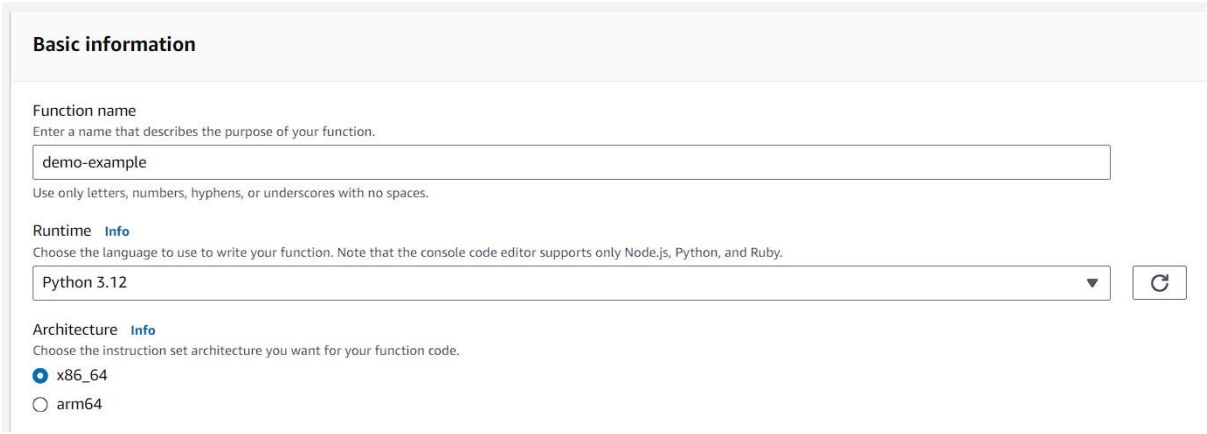


2. This is the dashboard of AWS Lambda. Now click on create a function. Then you will be directed on a new page where you can create your function.

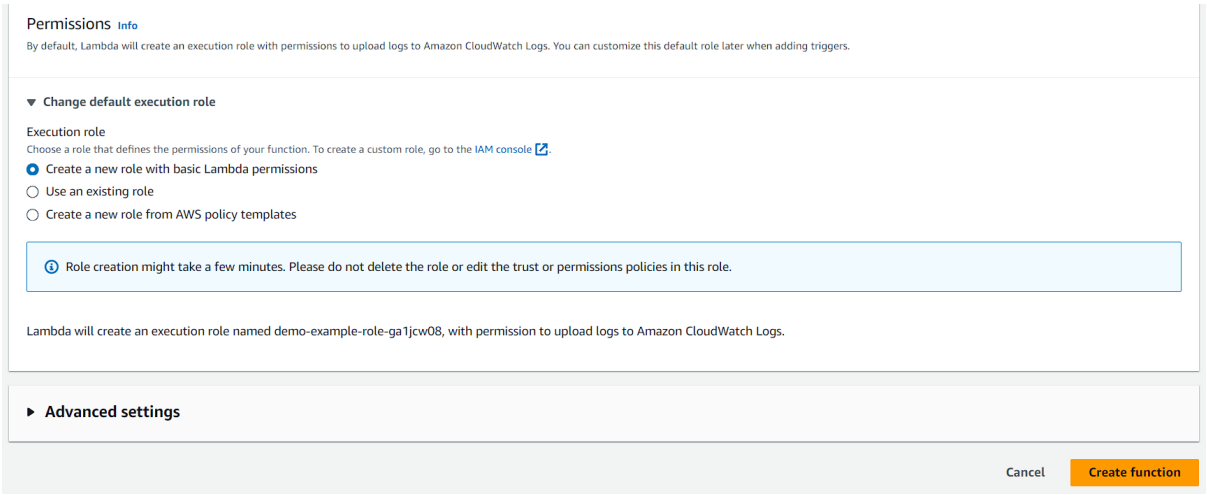


3. Now you have to select Author from scratch.

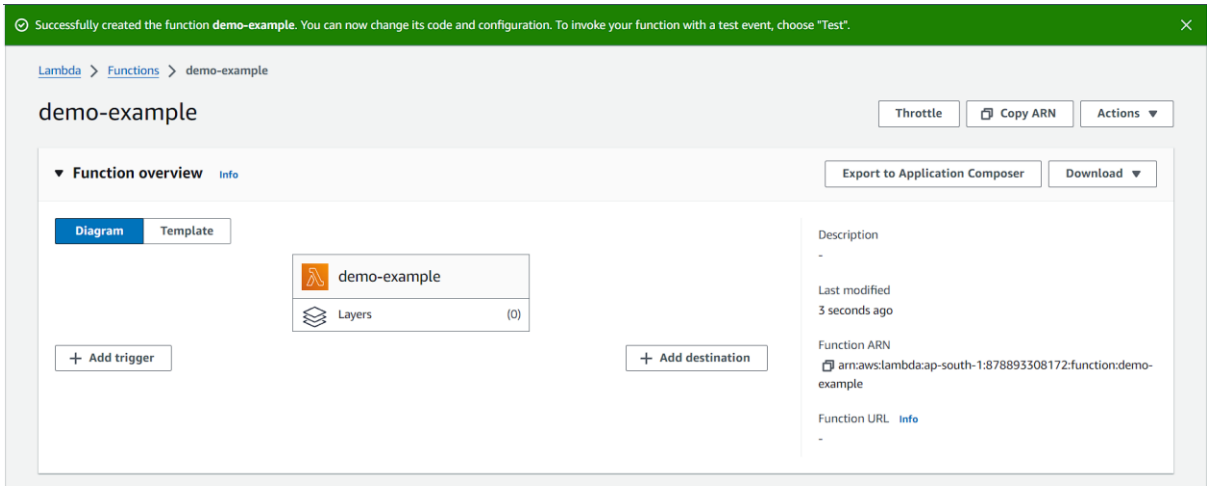
4. Then give your function a name and select its runtime as Python. You can also choose other runtimes too.
5. Then choose the architecture as x86.



6. Now in the permissions for the execution role. You have to select create a new role with basic lambda permission. This will create an IAM role for you itself. If you want to modify the permissions in the role you can do that too.
7. Then just click on create function.

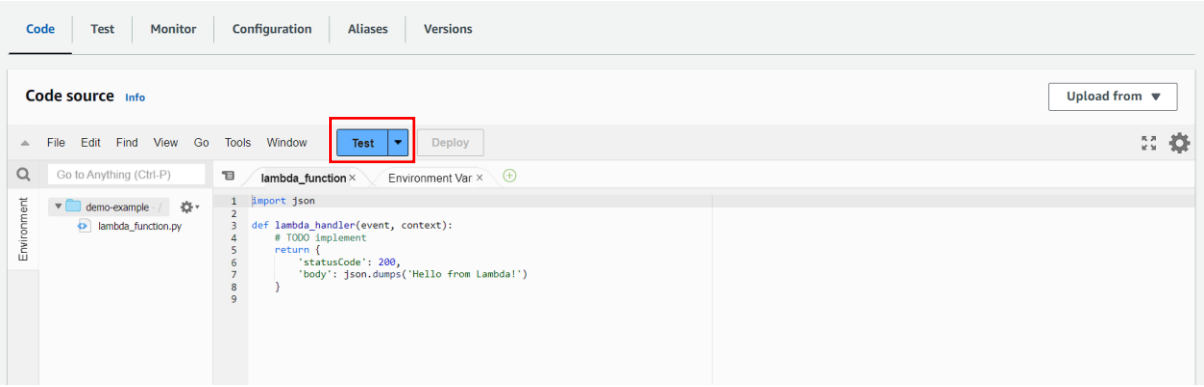


8. This is how your function will look like.



9. If you'll scroll down a little you will see that there is a lot of things to do. But for now, you are just going to test the code written here for you by Lambda.

10. For that click on Test. You will see that an event manager has been opened you need to give some information in that.



11. Here you just need to give it a name that's it. Then click on save.

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function's invocation result.

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event

Edit saved event

Event name

demo-event

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Event sharing settings

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

Template - *optional*

hello-world

12. After saving the test event you need to click on test. You will see the output.