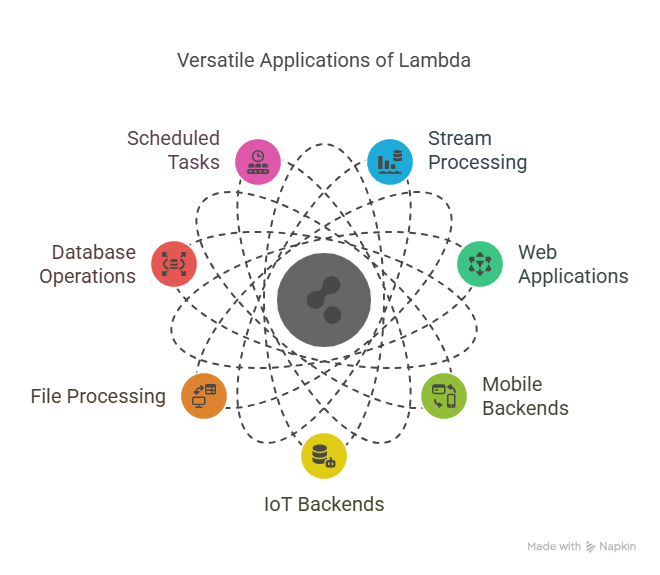
**AWS LAMBDA**

* **What is AWS Lambda?**

AWS Lambda is a powerful **serverless computing service** that automatically runs code in response to events, without requiring you to manage the underlying infrastructure. It supports event-driven applications triggered by events such as HTTP requests, DynamoDB table updates, or state transitions. You simply upload your code (as a .zip file or container image), and Lambda handles everything from provisioning to scaling and maintenance. It automatically scales applications based on traffic, handling server management, auto-scaling, security patching, and monitoring.

* **When to use Lambda?**

Lambda is an ideal compute service for application scenarios that need to scale up rapidly, and scale down to zero when not in demand. For example, you can use Lambda for:

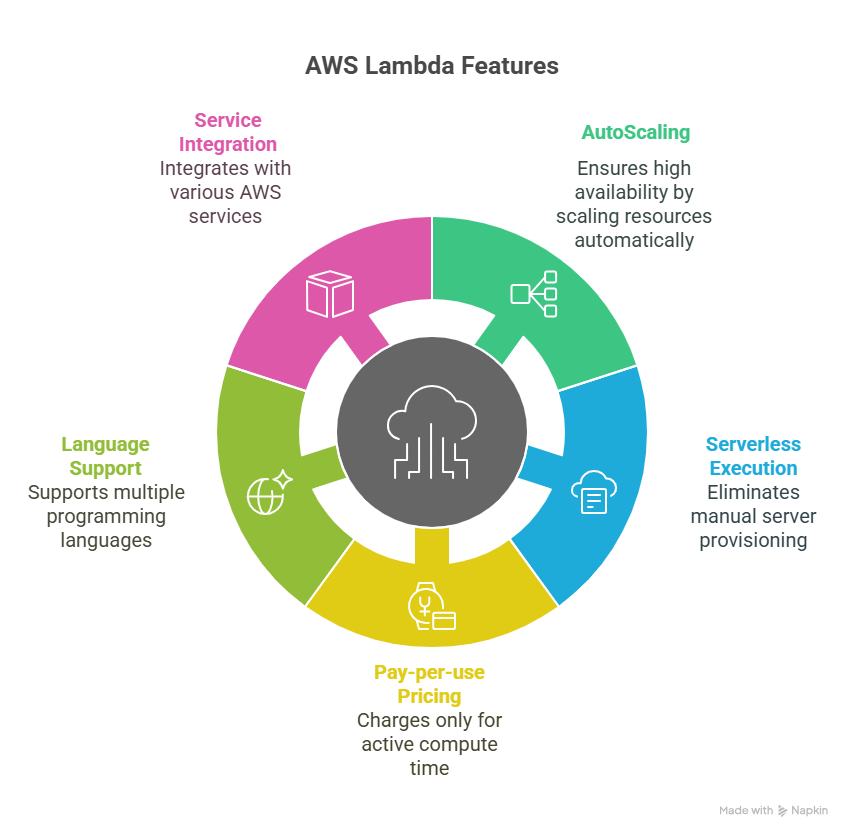


* **How Lambda Works?**

Because Lambda is a serverless, event-driven compute service, it uses a different programming paradigm than traditional web applications. The following model illustrates how Lambda fundamentally works:

1. You write and organize your code in [Lambda functions](https://docs.aws.amazon.com/lambda/latest/dg/concepts-basics.html#gettingstarted-concepts-function), which are the basic building blocks you use to create a Lambda application.
2. You control security and access through [Lambda permissions](https://docs.aws.amazon.com/lambda/latest/dg/lambda-permissions.html), using [execution roles](https://docs.aws.amazon.com/lambda/latest/dg/lambda-intro-execution-role.html) to manage what AWS services your functions can interact with and what resource policies can interact with your code.
3. Event sources and AWS services [trigger](https://docs.aws.amazon.com/lambda/latest/dg/concepts-event-driven-architectures.html) your Lambda functions, passing event data in JSON format, which your functions process (this includes event source mappings).
4. [Lambda runs your code](https://docs.aws.amazon.com/lambda/latest/dg/concepts-how-lambda-runs-code.html) with language-specific runtimes (like Node.js and Python) in execution environments that package your runtime, layers, and extensions.

* **Features Of AWS Lambda:**



* **Benefits Of AWS Lambda:**
* **Cost Efficiency**: It only charges for the compute that is only for running known as pay-as-you-go model.
* **Automatic Scaling**: AWS Lambda automatically helps in scaling your applications by running code in response to each trigger.
* **Reduced Operational Compliance**: It allows the developers to focus on building your logic, the aws itself while take care of the infrastructure.
* **Integration with AWS Services**: It provides a seamlessly integration with other[AWS services,](https://www.geeksforgeeks.org/top-aws-services/) enabling strong and scalable applications.

**AWS LAMBDA LAB WORK:**

**Task 1: Sign in to AWS Management Console:**

1. Click on the Open Console button, and you will get redirected to AWS Console in a new browser tab.

2. On the AWS sign-in page,

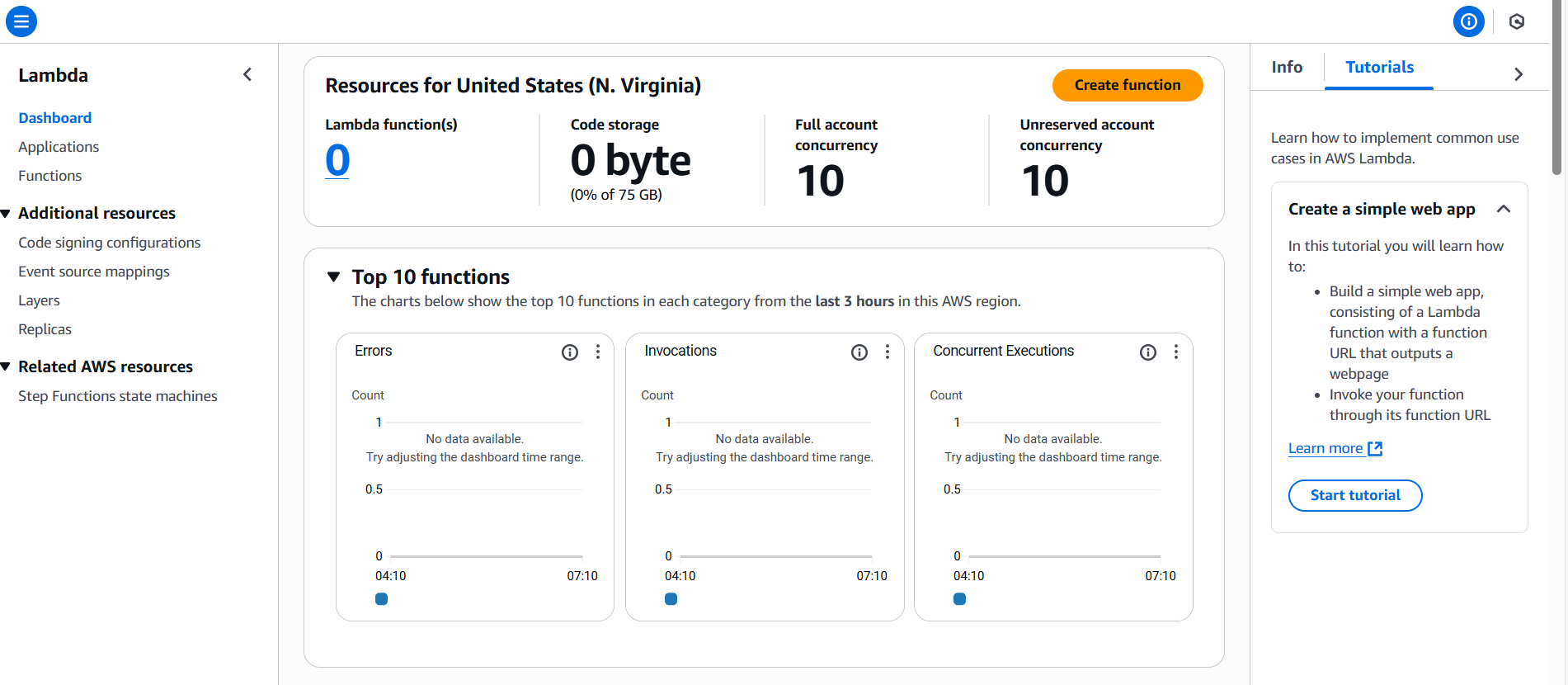
· Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.

· Now copy your User Name and Password in the Lab Console to the IAM Username and Password in AWS Console and click on the Sign in button.

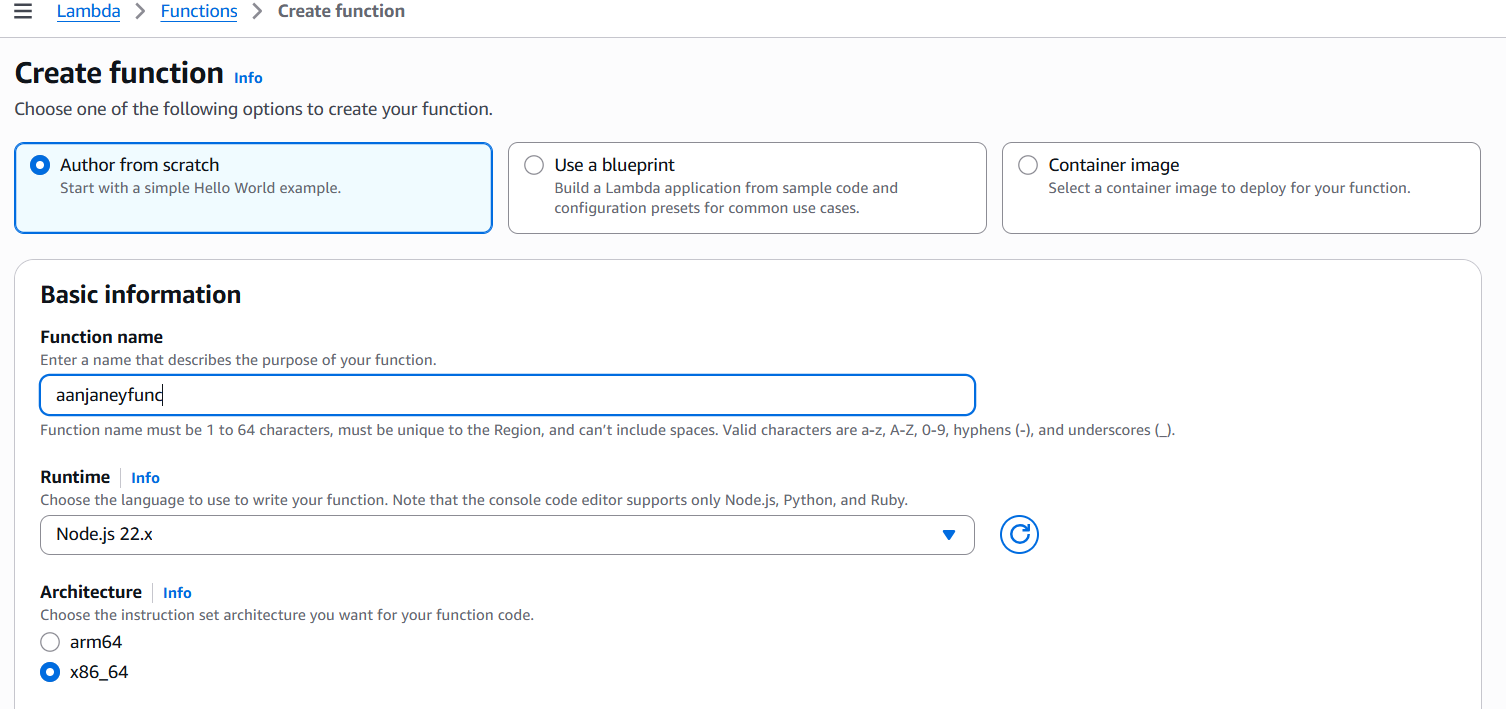
3. Once Signed In to the AWS Management Console, Make the default AWS Region as US East (N. Virginia) us-east-1.

**Task 2: Open Lambda and create a Function:**

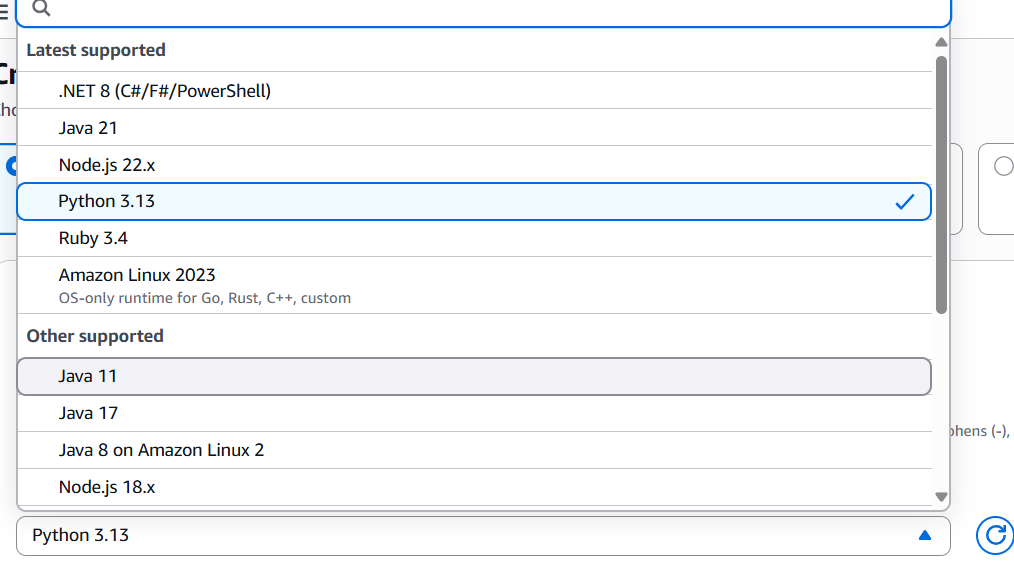
1. Open lambda in AWS.
2. Now, go to create function in the Lambda.



3.Now, a new window will appear. Here you have to give name to your function and you can also choose your desired programming language.



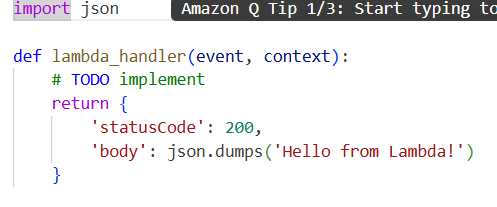
* Here, we have given the name of the function as aanjaneyfunc.
* Now choose your desired programming language. Here, we have chosen python.



1. Keep all other descriptions as default and click on create function.
2. Now, your Lambda function is created.

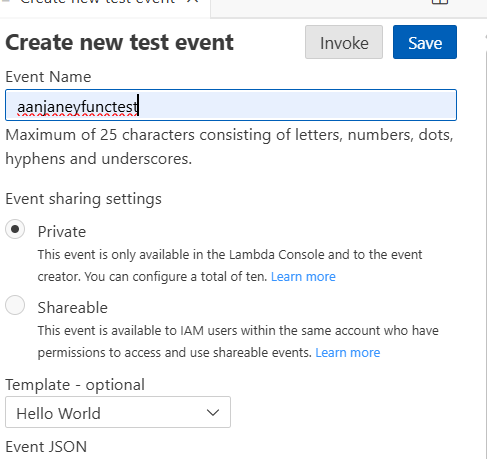
**Task 3: Running the python default code in lambda function code editor.**

1. Now, in the Lambda function code editor you will be able to see the following code:



1. Now, create a test event and test the default program.

* Click the ”Test” button.
* Create a new test event and give the event a name and save it.



3.Click “Test” again.

4.You will see the output “Hello from Lambda”



**Task 4: View logs in Lambda monitor**

1. In the Lambda dashboard, go to the “Monitor” tab.

2. Click “View logs in CloudWatch.

3. You can see the execution logs there.

**Task 5: Add a Trigger**

1. Go to the “configuration” tab.
2. In the “Triggers” section click “Add Trigger”.
3. Here, you can select any service like S3, API GATEWAY etc.
4. Configure the trigger and save