

Create a virtual network with 2 subnets. Each subnet should have 16 Ips only.

1). Step:- creating vpc

**Create VPC** [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

**VPC settings**

Resources to create [Info](#)  
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional  
Creates a tag with a key of 'Name' and a value that you specify.

aws-assignment1

IPv4 CIDR block [Info](#)  
☒ IPv4 CIDR manual input  
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR  
172.31.0.0/16

IPv6 CIDR block [Info](#)  
☒ No IPv6 CIDR block  
☐ IPAM-allocated IPv6 CIDR block  
☐ Amazon-provided IPv6 CIDR block  
☐ IPv6 CIDR owned by me

Tenancy [Info](#)  
Default

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key: Name Value - optional: aws-assignment1

[Add new tag](#)  
You can add 49 more tags.

[Cancel](#) [Create VPC](#)

2). Adding the vpc cidr range and clicking on creating.

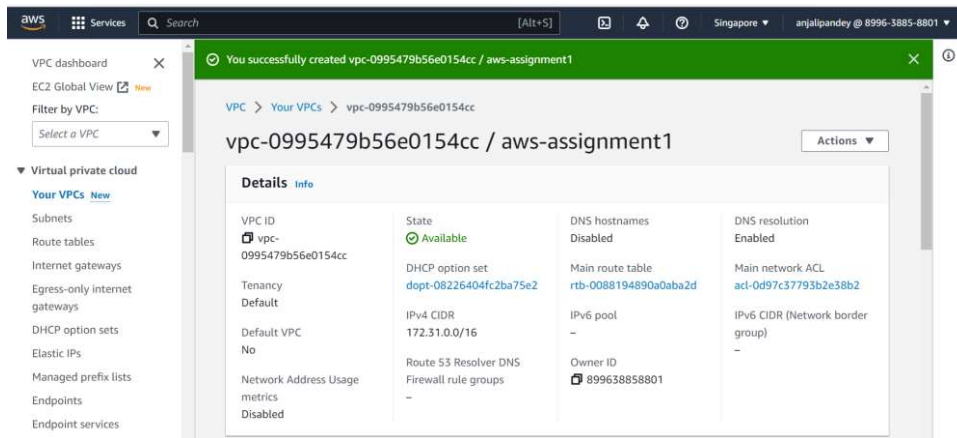
**You successfully created vpc-0995479b56e0154cc / aws-assignment1**

**vpc-0995479b56e0154cc / aws-assignment1** [Actions](#)

**Details** [Info](#)

VPC ID vpc-0995479b56e0154cc	State <b>Available</b>	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-08226404fc2ba75e2	Main route table rtb-0088194890a0aba2d	Main network ACL acl-0d97c37793b2e38b2
Default VPC No	IPv4 CIDR 172.31.0.0/16	IPv6 pool -	IPv6 CIDR (Network border group) -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 899638858801	

3).step: Now creating subnet-1 which is having 16 ip only.



4).step: Now creating subnet-2 which is having 16 ip only.

**Subnet 2 of 2**

**Subnet name**  
Create a tag with a key of 'Name' and a value that you specify.  
aws-subnet2  
The name can be up to 256 characters long.

**Availability Zone** [Info](#)  
Choose the zone in which your subnet will reside, or let Amazon choose one for you.  
Asia Pacific (Singapore) / ap-southeast-1a

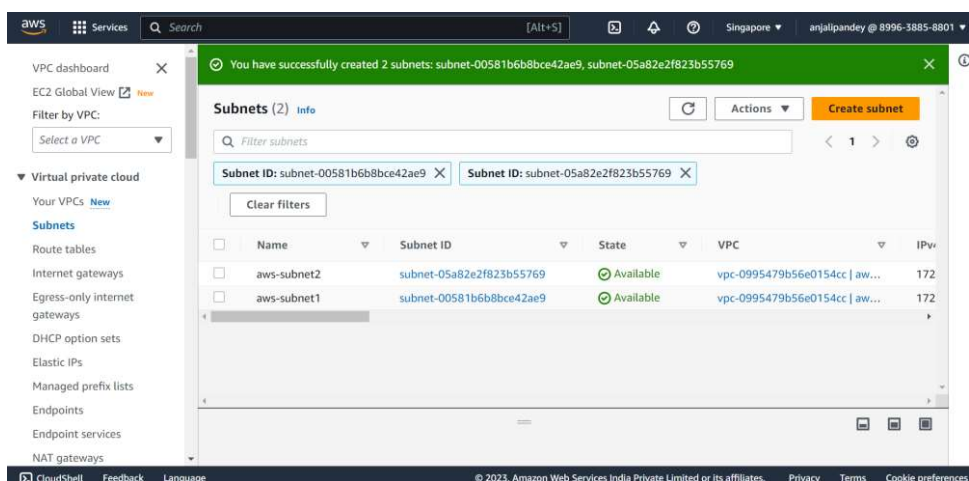
**IPv4 CIDR block** [Info](#)  
172.31.127.64/28

**Tags - optional**

Key	Value - optional	
Name	aws-subnet2	Remove

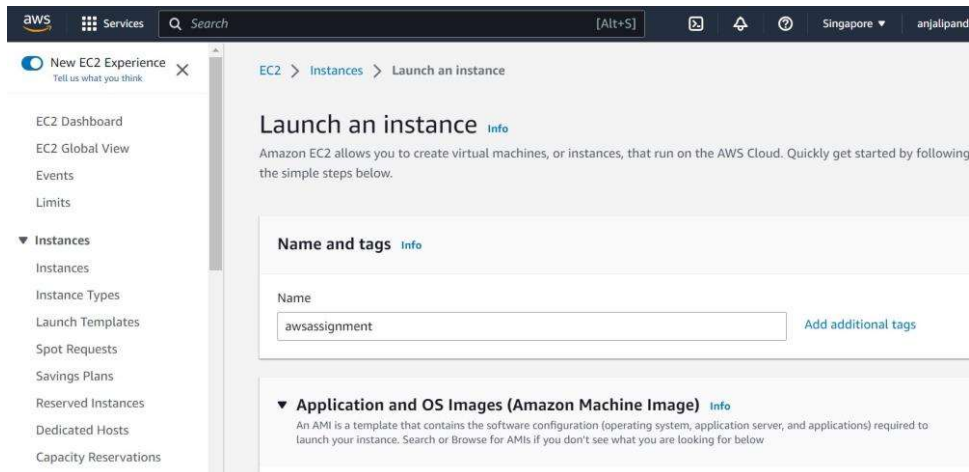
[Add new tag](#)  
You can add 49 more tags.

[Remove](#)

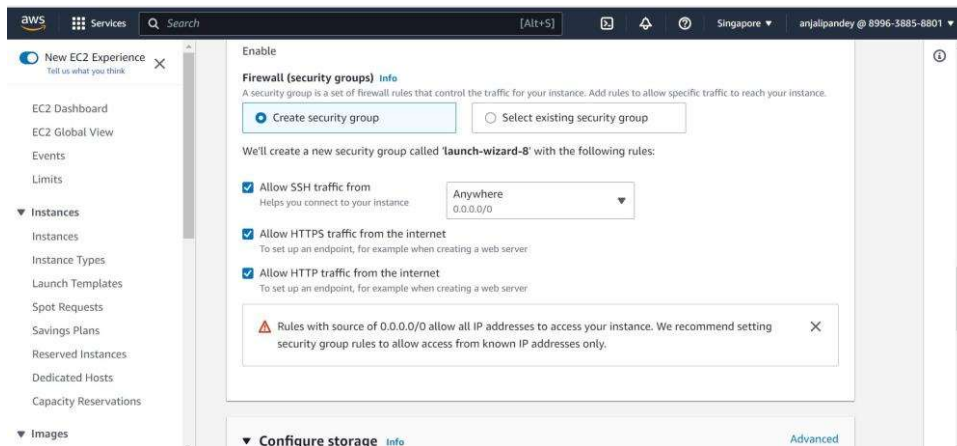


**Inside one of the subnets, create a VM and deploy an application code inside it (any existing application created by you before). Make sure to use appropriate NACLs and SGs.**

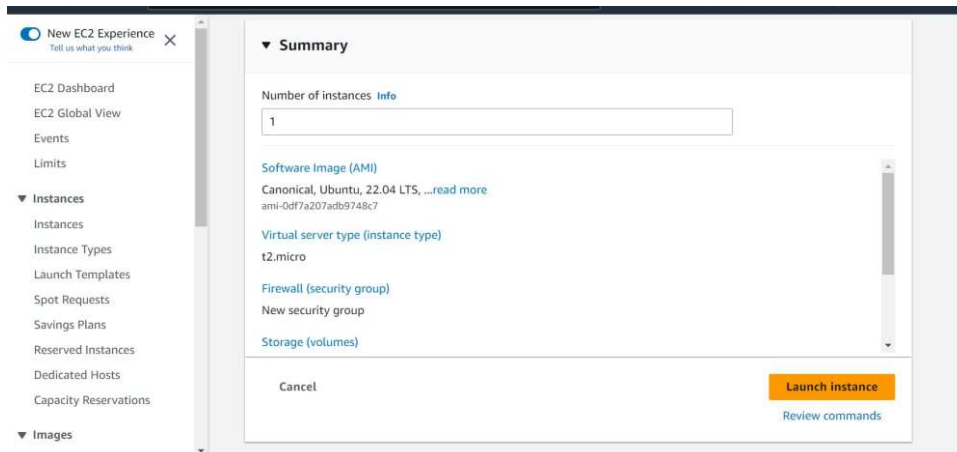
1).step:- Go to EC2 and click on launch instance.



2). Step:- Adding custom sg in vm(EC2)



3). Step:- choose instance type and images which is based on our requirement.



Services

Search

[Alt+S]

Singaporeanjali.pandey @ 8996-3885-8801

New EC2 Experience

EC2 Dashboard

EC2 Global View

Events

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

EC2 > Instances > Launch an instance

**Success**  
Successfully initiated launch of instance (i-0a62920368dff7ae32)  
[Launch log](#)

Next Steps


Create billing and free tier usage alerts

Once your instance is running, log into it from your local computer.

Connect to instance

Connect an RDS database

Configure the connection between an EC2 instance and a database to allow traffic flow between them.



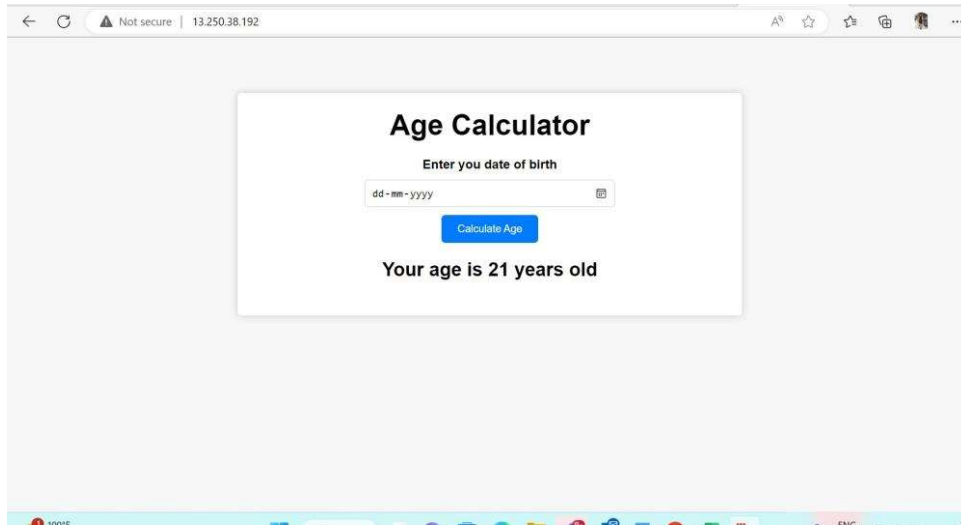
```

aws  Services  Q Search [Alt+S] Singapore anjalpandey @ 8996-3885-5801
Enabling module authz.user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service - /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service - /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.1-4build1) ...

i-Oa62920368df7ae32 (awsassignment)

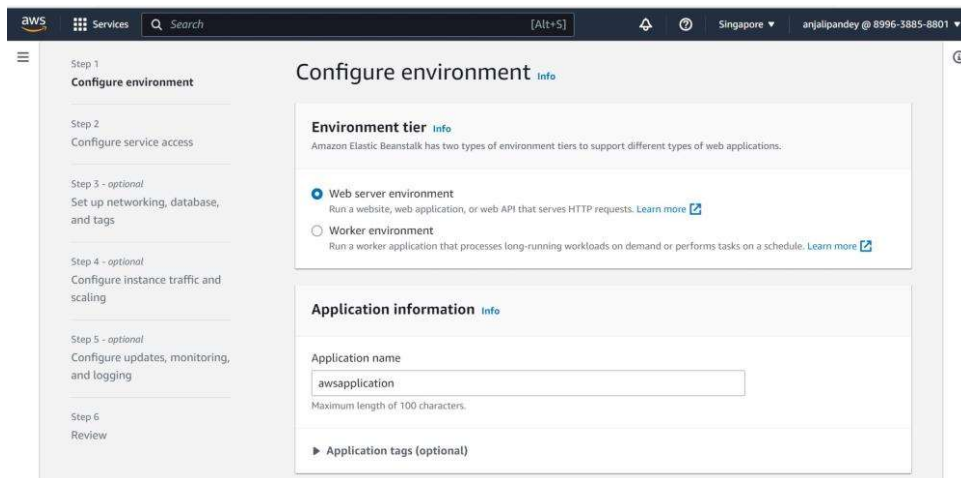
```

7). Now running my application in the server and starting apache to server.



**Deploy the same application to Elastic beanstalk Service.**

1). Step:- Go to Elasticbeanstalk and create a enviroment.



2). Add the information about application enviroment.

**Environment information** [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Domain  
 .ap-southeast-1.elasticbeanstalk.com [Check availability](#)

awsapplication.ap-southeast-1.elasticbeanstalk.com is available

Environment description

**Platform** [Info](#)

3). Add the information about platform and I am using nodejs.

**Platform** [Info](#)

Platform type  
☒ Managed platform  
 Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ Custom platform  
 Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Platform branch

Platform version

**Application code** [Info](#)

☐ Sample application

4). Uploading the application code.

**Application code** [Info](#)

☐ Sample application

☐ Existing version  
 Application versions that you have uploaded.

☒ Upload your code  
 Upload a source bundle from your computer or copy one from Amazon S3.

Version label  
 Unique name for this version of your application code.

Source code origin. Maximum size 2 GB

☒ Local file

Upload application

File name: age-calculator.zip  
 File must be less than 2GB max file size

☐ Public S3 URL

5). Configure the Ec2 role and key pair.

The screenshot shows the AWS IAM console with the 'EC2 instance profile' configuration page. The left sidebar lists steps: Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area shows the 'Service role' section with 'Use an existing service role' selected. Below this, the 'EC2 key pair' section shows 'awsass' selected. The 'EC2 instance profile' section has a dropdown menu and a 'View permission details' button. At the bottom, there are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons.

6). Add the vpc and subnets

The screenshot shows the AWS Elastic Beanstalk console with the 'Set up networking, database, and tags' step. The left sidebar lists steps: Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area shows the 'Launch your environment in a custom VPC' section with a dropdown menu showing 'vpc-0995479b56e0154cc | (172.31.0.0/16) | aws-assignment1'. Below this, the 'Instance settings' section shows 'Public IP address' as 'Activated'. The 'Instance subnets' section shows a table with one subnet selected:

Availability Zone	Subnet	CIDR	Name
ap-southeast-1a	subnet-00581b6b...	172.31.127.80/28	aws-subnet1

7). Now click on review and create.

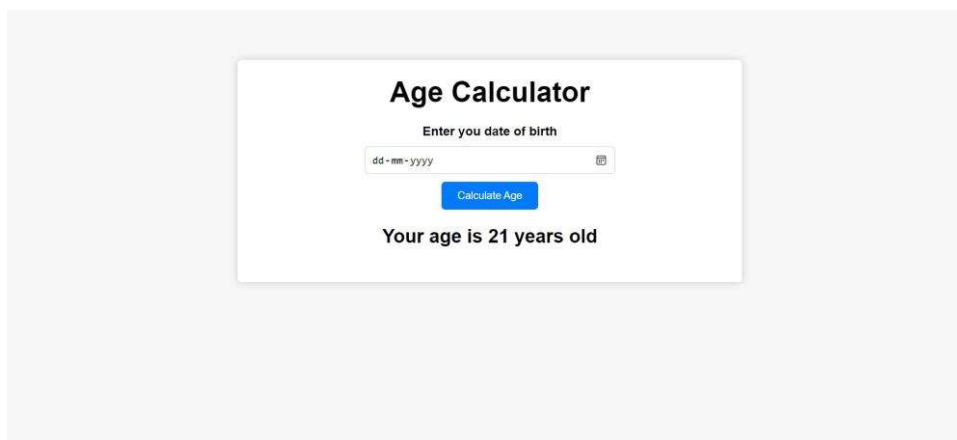
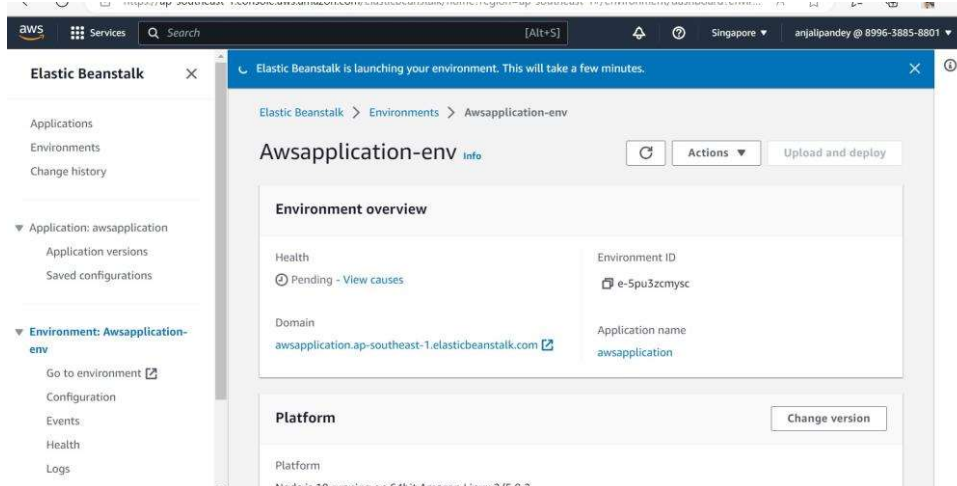
The screenshot shows the AWS Elastic Beanstalk console with the 'Review' step. The left sidebar lists steps: Step 1 (Configure environment), Step 2 (Configure service access), Step 3 - optional (Set up networking, database, and tags), Step 4 - optional (Configure instance traffic and scaling), Step 5 - optional (Configure updates, monitoring, and logging), and Step 6 (Review). The main content area shows the 'Review' page with 'Step 1: Configure environment' and 'Step 2: Configure service access'. The 'Environment information' section shows:

Environment tier	Application name
Web server environment	awsapplication

Environment name	Application code
Awsapplication-env	age-calculator.zip

Platform: amaws:elasticbeanstalk-ap-southeast-1:platform/Node.js 18 running on 64bit Amazon Linux 2/5.8.2

8). Now application will get on a while we check the application.



Application is working fine.

Create a Lambda that should trigger as soon as you upload a file in the S3 bucket. Function should be able to print the name of the file uploaded in the function.

1). Step:- creating the S3 bucket



**Create bucket** [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

**General configuration**

Bucket name  
lambdatriggerbucket3  
Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region  
Asia Pacific (Singapore) ap-southeast-1

Copy settings from existing bucket - optional  
Only the bucket settings in the following configuration are copied.  
[Choose bucket](#)

**Object Ownership** [Info](#)

**Successfully created bucket "lambdatriggerbucket3"**  
To upload files and folders, or to configure additional bucket settings choose [View details](#).

**Account snapshot**  
Last updated: Jun 6, 2023 by Storage Lens. Metrics are generated every 24 hours. [Learn more](#)  
[View Storage Lens dashboard](#)

Total storage	Object count	Average object size	
1.2 MB	973	1.3 KB	You can enable advanced metrics in the "default-account-dashboard" configuration.

**Buckets (10)** [Info](#)  
Buckets are containers for data stored in S3. [Learn more](#)

[Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

[Find buckets by name](#)

2). Step:- now creating the lambda function.

**Create function** [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**  
Start with a simple Hello World example.

☐ **Use a blueprint**  
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**  
Select a container image to deploy for your function.

**Basic information**

Function name  
Enter a name that describes the purpose of your function.  
lambdafunctionassignment2  
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
Node.js 18.x

Choose the instruction set architecture you want for your function code.

☒ x86\_64

☐ arm64

**Permissions** [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

s3-trigger-role

[View the s3-trigger-role role on the IAM console.](#)

► **Advanced settings**

Successfully created the function **lambdafunctassignment2**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Lambda > Functions > lambdafunctassignment2

**lambdafunctassignment2** Throttle Copy ARN Actions

▼ **Function overview** [Info](#)

lambdafunctassignment2
 

Layers (0)

+ Add trigger

+ Add destination

**Description**  
-

**Last modified**  
5 seconds ago

**Function ARN**  
arn:aws:lambda:ap-southeast-1:899638858801:function:lambdafunctassignment2

**Function URL** [Info](#)

3). Step:- writing the code for printing the file name when uploaded into S3 bucket.

Successfully updated the function **lambdafunctassignment2**.

**Code source** [Info](#) Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

- lambdafunctassignment2
- index.mjs

```

1  import { S3Client, GetObjectCommand } from '@aws-sdk/client-s3';
2  const s3 = new S3Client({region: 'ap-southeast-1'});
3
4  export const handler = async (event, context) => {
5    //console.log('Received event:', JSON.stringify(event, null, 2));
6
7    // Get the object from the event and show its content type
8    const bucket = event.Records[0].s3.bucket.name;
9    const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/^\//, ''));
10   const params = {
11     Bucket: bucket,
12     Key: key,
13   };
14
15   try {
16     const { ContentType } = await s3.send(new GetObjectCommand(params));
17     console.log('CONTENT TYPE:', ContentType);
18     console.log('Filename:', `${key}`);
19     return ContentType;
20   } catch (err) {
21     console.log(err);
22   }
23 }

```

4). Now adding the trigger in lambda function.

The screenshot shows the 'Add trigger' configuration page in the AWS Lambda console. The 'Trigger configuration' section is active, showing a dropdown menu for 'S3' with the text 'aws storage'. Below this, the 'Bucket' section instructs the user to select an S3 bucket. A search box contains 's3/lambda-trigger-bucket3', and the 'Bucket region' is set to 'ap-southeast-1'. The 'Event types' section has a dropdown menu and two buttons: 'All object create events' and 'PUT'.

5). Now uploading the file in s3 bucket and check whether filename is printing or not in logs.

The screenshot shows the 'Upload' page in the AWS S3 console for the bucket 'lambda-trigger-bucket3'. The 'Files and folders' section shows a table with one file: 'testfile.txt', which is a 'text/plain' file of size '0 B'. The 'Destination' section is visible below the table.

The screenshot shows the 'Upload: status' page in the AWS S3 console. A green banner at the top indicates 'Upload succeeded'. The 'Summary' section shows the destination as 's3://lambda-trigger-bucket3' and the upload status as 'Succeeded' with '1 file, 0 B (0%)' and 'Failed' with '0 files, 0 B (0%)'. The 'Files and folders' section is visible at the bottom.

6). Now checking my file name in cloud watch logs.

The screenshot shows the AWS CloudWatch console interface. The left sidebar contains navigation links: Dashboards, Alarms (0), Logs (0), Log groups, Logs Insights, Live tail (New), Metrics, X-Ray traces, Events, Application monitoring, and Insights. The main content area displays a list of log events for a specific log group. The events are filtered by a search term 'Filter events' and a time range of 1m. The events are as follows:

Timestamp	Message
2023-06-07T17:35:45.942+05:30	INIT_START Runtime Version: nodejs18.v7 Runtime Version ARN: arn:aws:lambda...
2023-06-07T17:35:46.474+05:30	2023-06-07T12:05:46.474Z undefined INFO Loading function
2023-06-07T17:35:46.523+05:30	START RequestId: 3c0a672a-ff3d-482c-854e-6ff0573ffa94 Version: \$LATEST
2023-06-07T17:35:47.033+05:30	2023-06-07T12:05:47.033Z 3c0a672a-ff3d-482c-854e-6ff0573ffa94 INFO CONTENT ...
2023-06-07T12:05:47.033Z	3c0a672a-ff3d-482c-854e-6ff0573ffa94 INFO CONTENT TYPE: text/plain <a href="#">Copy</a>
2023-06-07T17:35:47.033+05:30	2023-06-07T12:05:47.033Z 3c0a672a-ff3d-482c-854e-6ff0573ffa94 INFO filename:...
2023-06-07T12:05:47.033Z	3c0a672a-ff3d-482c-854e-6ff0573ffa94 INFO filename: testfile.txt <a href="#">Copy</a>
2023-06-07T17:35:47.054+05:30	END RequestId: 3c0a672a-ff3d-482c-854e-6ff0573ffa94
2023-06-07T17:35:47.054+05:30	REPORT RequestId: 3c0a672a-ff3d-482c-854e-6ff0573ffa94 Duration: 531.13 ms ...

At the bottom of the log events list, it says "No newer events at this moment. Auto retry paused. [Resume](#)". There is also a "Back to top" button.