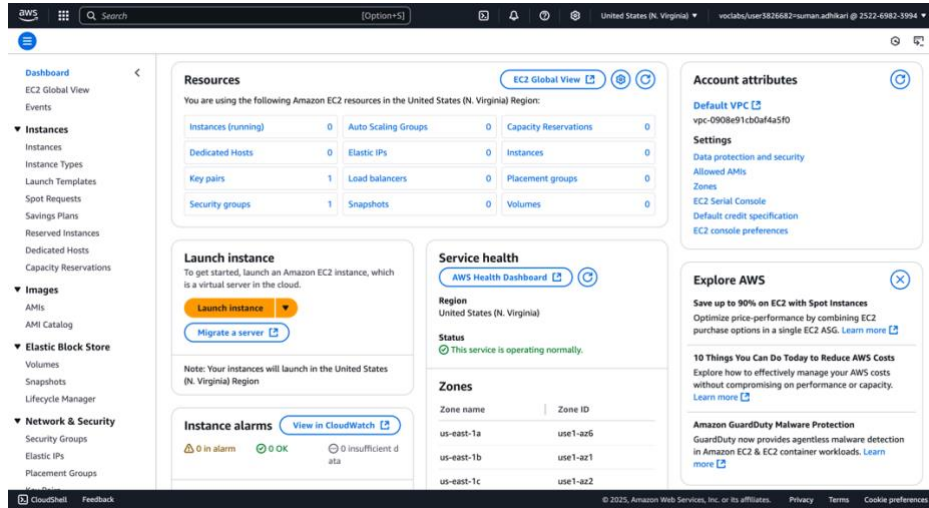


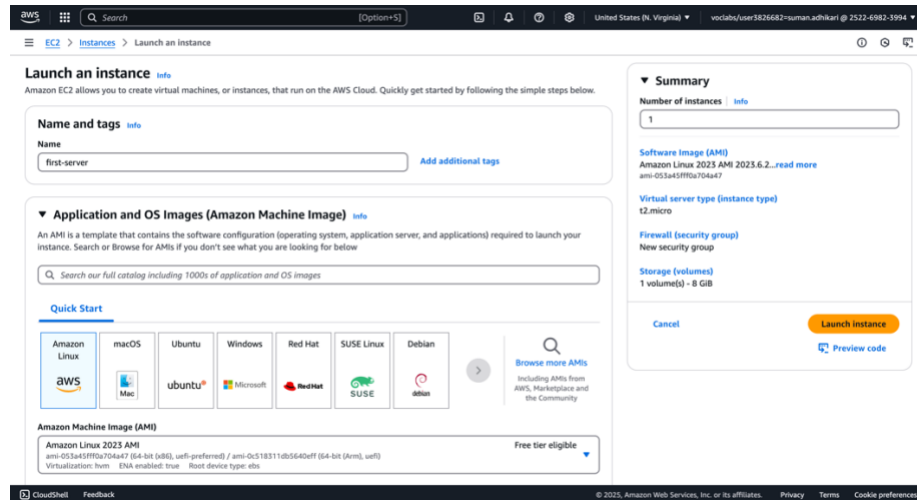
Assignment #1

Task 1 – Webpage that shows my name from EC2

1. Ec2 Dashboard



2. Creating EC2



Launch an instance

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI
ami-053a45ff0a704a47 (64-bit (x86), uefi-preferred) / ami-053b311db564b0eff (64-bit (arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250211.0 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86) Boot mode: uefi-preferred AMI ID: ami-053a45ff0a704a47 Username: ec2-user Verified provider

Instance type

Instance type: t2.micro Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0762 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour

Additional costs apply for AMIs with pre-installed software

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name: vockey Create new key pair

Network settings

Network: vpc-0908e91cb0af4a5f0
Subnet: No preference (Default subnet in any availability zone)
Auto-assign public IP: Enable
Additional charges apply when outside of free tier allowance

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

- ☒ Allow SSH traffic from Anywhere (0.0.0.0/0)
- ☒ Allow HTTPS traffic from the internet
- ☒ Allow HTTP traffic from the internet

Summary

Number of instances: 1

Software image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-053a45ff0a704a47

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Preview code

Launch an instance

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name: vockey Create new key pair

Network settings

Network: vpc-0908e91cb0af4a5f0
Subnet: No preference (Default subnet in any availability zone)
Auto-assign public IP: Enable
Additional charges apply when outside of free tier allowance

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

- ☒ Allow SSH traffic from Anywhere (0.0.0.0/0)
- ☒ Allow HTTPS traffic from the internet
- ☒ Allow HTTP traffic from the internet

Summary

Number of instances: 1

Software image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-053a45ff0a704a47

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Preview code

3. Access & Install Apache server on instance

```

AWS
Search [Option+S]
United States (N. Virginia) vccabts/user3826682=human.adhikari @ 2522-6982-3994

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-86-28 ~]$

I-0a17Aaf37520ebb40 (first-server)
PublicIP: 18.208.161.33 PrivateIP: 172.31.86.28

CloudShell Feedback
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

```

AWS CloudShell
Search [Options+]
United States (N. Virginia) vclab0/user325682~human.adhikari @ 2522-6982-3994

[ec2-user@ip-172-31-86-28 ~]$ sudo yum install httpd -y
amazon linux 2023 Kernel Livepatch repository
Dependencies resolved.
105 kB/s | 14 KB | 00:00

Package Architecture Version Repository Size
-----
Installing:
httpd x86_64 2.4.42-1.amzn2023 amazonlinux 48 k
Installing dependencies:
apr x86_64 1.7.5-1.amzn2023.0.2 amazonlinux 130 k
apr-util x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 98 k
generic-logos-httpd noarch 18.0.0-12.amzn2023.0.3 amazonlinux 19 k
httpd-core x86_64 2.4.42-1.amzn2023 amazonlinux 1.4 M
httpd-filesystem noarch 2.4.42-1.amzn2023 amazonlinux 14 k
httpd-tools x86_64 2.4.42-1.amzn2023 amazonlinux 81 k
libbrotli x86_64 1.0.9-4.amzn2023.0.2 amazonlinux 315 k
mailcap noarch 2.1.49-3.amzn2023.0.3 amazonlinux 33 k
Installing weak dependencies:
apr-util-openssl x86_64 1.6.3-1.amzn2023.0.1 amazonlinux 17 k
mod_http2 x86_64 2.0.27-1.amzn2023.0.3 amazonlinux 146 k
mod_lua x86_64 2.4.42-1.amzn2023 amazonlinux 61 k

Transaction Summary
-----
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-1.7.5-1.amzn2023.0.2.x86_64.rpm 309 kB/s | 17 kB | 00:00
(2/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm 2.1 MB/s | 130 kB | 00:00
(3/12): apr-util-1.6.3-1.amzn2023.0.1.x86_64.rpm 1.5 MB/s | 98 kB | 00:00
(4/12): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm 1.0 MB/s | 19 kB | 00:00
(5/12): httpd-2.4.42-1.amzn2023.x86_64.rpm 2.4 MB/s | 48 kB | 00:00
(6/12): httpd-filesystem-2.4.42-1.amzn2023.noarch.rpm 651 kB/s | 14 kB | 00:00

I-0a17aaf37520ebb40 (first-server)
PublicPv: 18.208.161.33 PrivatePv: 172.31.86.28

CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

```

AWS CloudShell
Search [Options+]
United States (N. Virginia) vclab0/user325682~human.adhikari @ 2522-6982-3994

Installing 1/12
Installing 2/12
Installing 3/12
Installing 4/12
Installing 5/12
Installing 6/12
Running scriptlet: httpd-filesystem-2.4.42-1.amzn2023.noarch 7/12
Installing 7/12
Installing 8/12
Installing 9/12
Installing 10/12
Installing 11/12
Running scriptlet: httpd-2.4.42-1.amzn2023.x86_64 12/12
Installing 12/12
Verifying 1/12
Verifying 2/12
Verifying 3/12
Verifying 4/12
Verifying 5/12
Verifying 6/12
Verifying 7/12
Verifying 8/12
Verifying 9/12
Verifying 10/12
Verifying 11/12
Verifying 12/12

Installed:
apr-1.7.5-1.amzn2023.0.2.x86_64 apr-util-1.6.3-1.amzn2023.0.1.x86_64 apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch httpd-2.4.42-1.amzn2023.x86_64 httpd-core-2.4.42-1.amzn2023.x86_64
httpd-filesystem-2.4.42-1.amzn2023.noarch httpd-tools-2.4.42-1.amzn2023.x86_64 libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch mod_http2-2.0.27-1.amzn2023.0.3.x86_64 mod_lua-2.4.42-1.amzn2023.x86_64

Complete!
[ec2-user@ip-172-31-86-28 ~]$

I-0a17aaf37520ebb40 (first-server)
PublicPv: 18.208.161.33 PrivatePv: 172.31.86.28

CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

4. Create index.html file

```

AWS CloudShell
Search [Options+]
United States (N. Virginia) vclab0/user325682~human.adhikari @ 2522-6982-3994

[ec2-user@ip-172-31-86-28 html]$ pwd
/var/www/html
[ec2-user@ip-172-31-86-28 html]$ touch index.html
touch: cannot touch 'index.html': Permission denied
[ec2-user@ip-172-31-86-28 html]$ sudo touch index.html
[ec2-user@ip-172-31-86-28 html]$ ls
index.html
[ec2-user@ip-172-31-86-28 html]$ vi index.html
[ec2-user@ip-172-31-86-28 html]$ sudo vi index.html
[ec2-user@ip-172-31-86-28 html]$ ls
index.html
[ec2-user@ip-172-31-86-28 html]$ ls -la
total 4
drwxr-xr-x. 2 root root 24 Feb 14 15:22 .
drwxr-xr-x. 4 root root 33 Feb 14 15:16 ..
-rw-r--r--. 1 root root 289 Feb 14 15:22 index.html
[ec2-user@ip-172-31-86-28 html]$ chmod 600 index.html
chmod: changing permissions of 'index.html': Operation not permitted
[ec2-user@ip-172-31-86-28 html]$ sudo chmod 600 index.html
[ec2-user@ip-172-31-86-28 html]$ ls -la
total 4
drwxr-xr-x. 2 root root 24 Feb 14 15:22 .
drwxr-xr-x. 4 root root 33 Feb 14 15:16 ..
-rw-----. 1 root root 289 Feb 14 15:22 index.html
[ec2-user@ip-172-31-86-28 html]$

I-0a17aaf37520ebb40 (first-server)
PublicPv: 18.208.161.33 PrivatePv: 172.31.86.28

CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

```
aws CloudShell Search [Option+S] United States (N. Virginia) voclabs/user1826682=human.adhikari @ 2522-6982-3994

[ec2-user@ip-172-31-86-28 html]$ pwd
/var/www/html
[ec2-user@ip-172-31-86-28 html]$ cat index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Task 1</title>
  </head>
  <body>
    <main>
      <h1>Welcome to Assignment1 - Task 1</h1>
      <p>My name is <b>Suman Adhikari</b>
    </main>
  </body>
</html>
[ec2-user@ip-172-31-86-28 html]$
```

i-0a17aaf37520ebb40 (first-server)
PublicIPs: 18.208.161.33 PrivateIPs: 172.31.86.28

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

5. Start httpd service

```
aws CloudShell Search [Option+S] United States (N. Virginia) voclabs/user1826682=human.adhikari @ 2522-6982-3994

0 httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)

Feb 14 15:28:49 ip-172-31-86-28.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 14 15:28:49 ip-172-31-86-28.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 14 15:28:49 ip-172-31-86-28.ec2.internal httpd[26719]: Server configured, listening on: port 80
Feb 14 15:29:54 ip-172-31-86-28.ec2.internal systemd[1]: Stopping httpd.service - The Apache HTTP Server...
Feb 14 15:29:55 ip-172-31-86-28.ec2.internal systemd[1]: httpd.service: Deactivated successfully.
Feb 14 15:29:55 ip-172-31-86-28.ec2.internal systemd[1]: Stopped httpd.service - The Apache HTTP Server.
[ec2-user@ip-172-31-86-28 html]$ sudo systemctl start httpd
-bash: sudo: command not found
[ec2-user@ip-172-31-86-28 html]$ sudo systemctl start httpd
[ec2-user@ip-172-31-86-28 html]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-02-14 15:30:16 UTC; 2s ago
     Docs: man:httpd.service(8)
   Main PID: 27066 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 177 (limit: 1111)
   Memory: 12.9M
      CPU: 51ms
   CGroup: /system.slice/httpd.service
           └─27066 /usr/sbin/httpd -DFOREGROUND
             └─27067 /usr/sbin/httpd -DFOREGROUND
               └─27068 /usr/sbin/httpd -DFOREGROUND
                 └─27069 /usr/sbin/httpd -DFOREGROUND
                   └─27070 /usr/sbin/httpd -DFOREGROUND

Feb 14 15:30:16 ip-172-31-86-28.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Feb 14 15:30:16 ip-172-31-86-28.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Feb 14 15:30:16 ip-172-31-86-28.ec2.internal httpd[27066]: Server configured, listening on: port 80
[ec2-user@ip-172-31-86-28 html]$
```

i-0a17aaf37520ebb40 (first-server)
PublicIPs: 18.208.161.33 PrivateIPs: 172.31.86.28

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

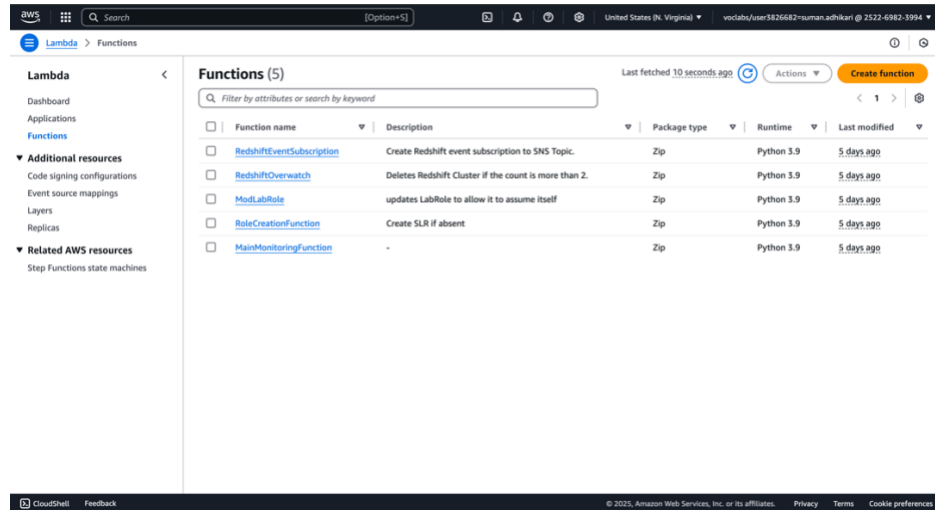
6. Access the site.



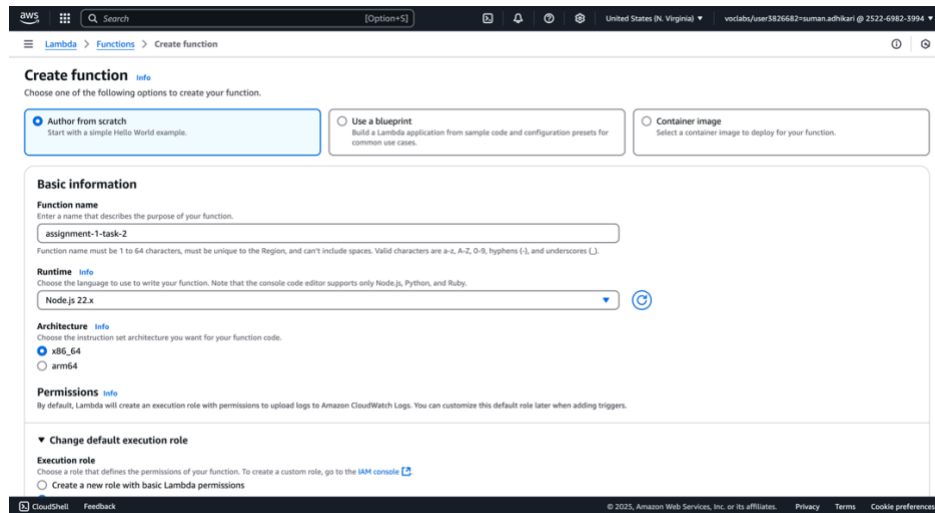
Task 2 – Simple API with lambda function URL

Lambda URL: <https://ppnhhiplr5eswdfxxytdescnii0npnux.lambda-url.us-east-1.on.aws/>

1. Access Lambda Dashboard



2. Create a lambda function



aws

Search

[Option+5]

United States (N. Virginia)

voiclabs/user3826682~suman.adhikari @ 2522-6982-3994

Lambda > Functions > Create function

▼ 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.

LabRole

View the [LabRole](#) role on the IAM console.

▼ Additional Configurations

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

☐ Enable Code signing

Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

☐ Enable encryption with an AWS KMS customer managed key

By default, Lambda encrypts the zip file archive using an AWS owned key.

☒ Enable function URL

Use function URLs to assign HTTP(S) endpoints to your Lambda function.

Auth type

Choose the auth type for your function URL. [Learn more](#)

☐ AWS_IAM

Only authenticated IAM users and roles can make requests to your function URL.

☒ NONE

Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

Function URL permissions

CloudShell

Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

aws

Search

[Option+5]

United States (N. Virginia)

voiclabs/user3826682~suman.adhikari @ 2522-6982-3994

Lambda > Functions > Create function

Auth type

Choose the auth type for your function URL. [Learn more](#)

☐ AWS_IAM

Only authenticated IAM users and roles can make requests to your function URL.

☒ NONE

Lambda won't perform IAM authentication on requests to your function URL. The URL endpoint will be public unless you implement your own authorization logic in your function.

Function URL permissions

When you choose auth type **NONE**, Lambda automatically creates the following resource-based policy and attaches it to your function. This policy makes your function public to anyone with the function URL. You can edit the policy later. To limit access to authenticated IAM users and roles, choose auth type **AWS_IAM**.

▼ View policy statement

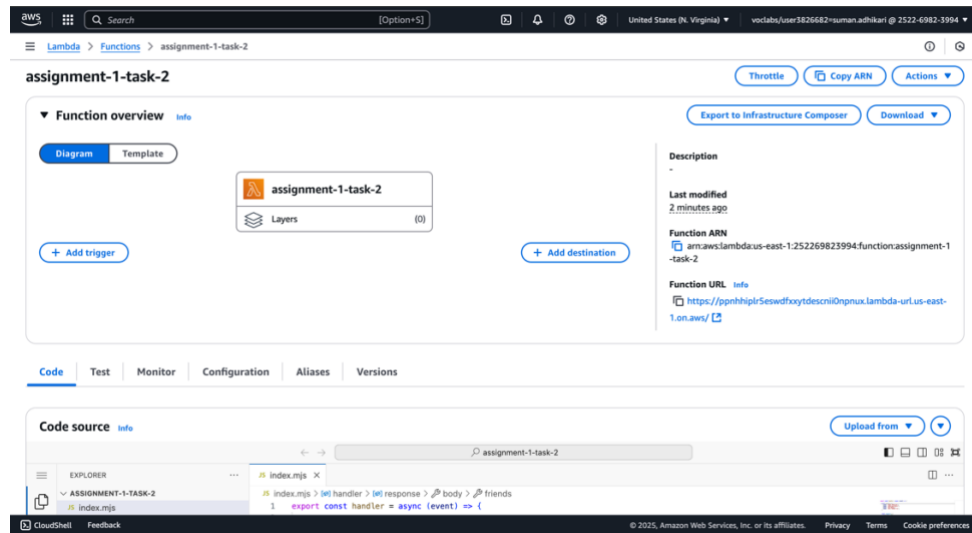
```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "StatementId": "FunctionURLAllowPublicAccess",
6       "Effect": "Allow",
7       "Principal": "*",
8       "Action": "lambda:InvokeFunctionUrl",
9       "Resource": "arn:aws:lambda:us-east-1:252269823994:function:assignment-1-task-2",
10      "Condition": {
11        "StringEquals": {
12          "lambda:FunctionUrlAuthType": "NONE"
13        }
14      }
15    ]
16  }
17 }
```

CloudShell

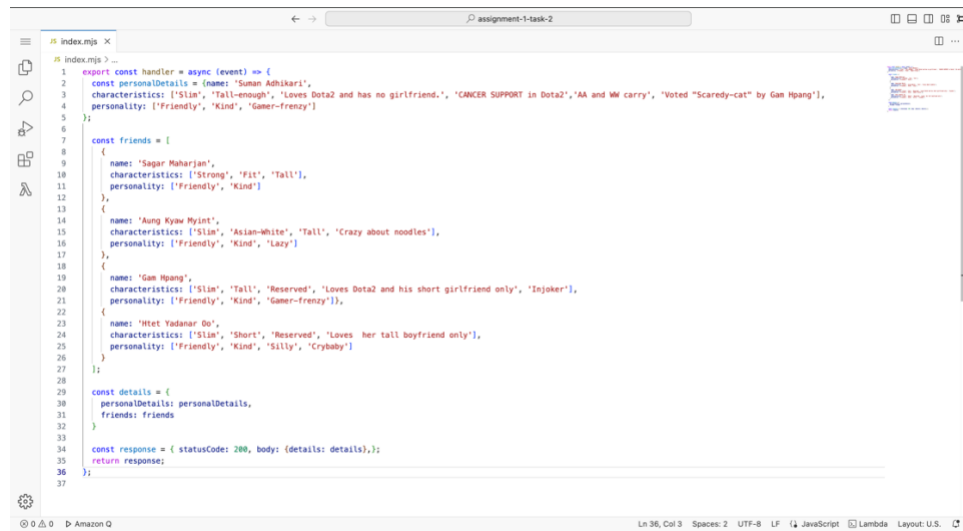
Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

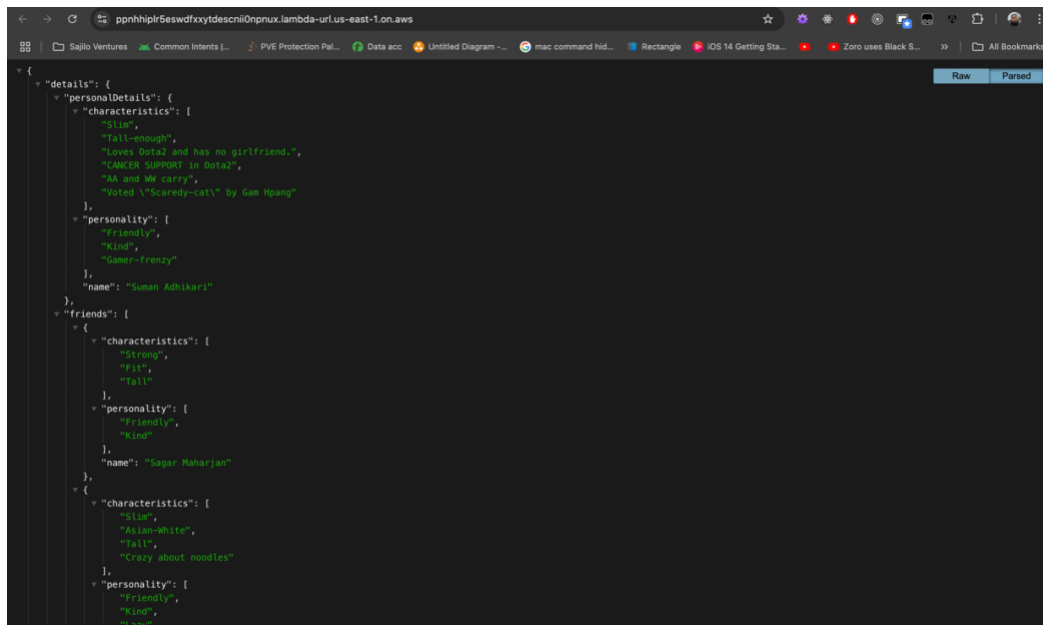
3. Lambda is created



4. Define the API details



5. Access the API response



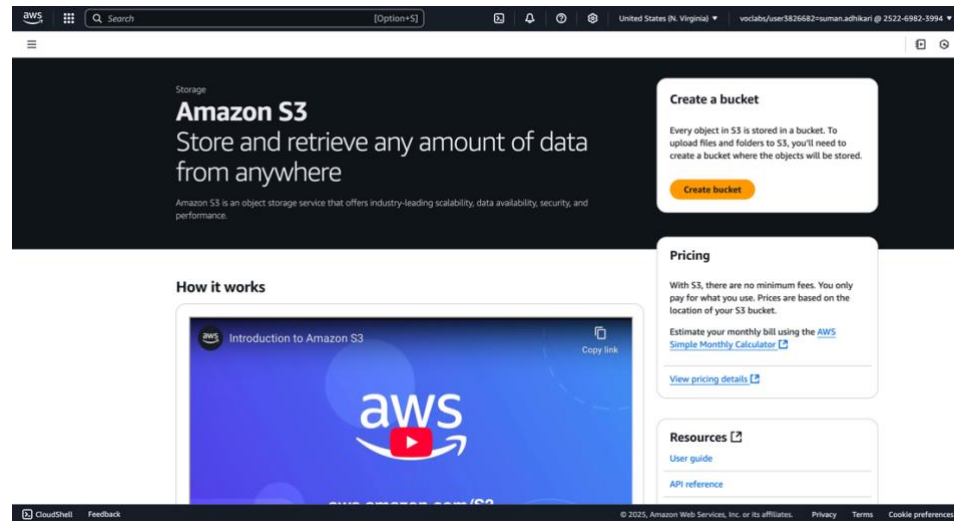
The screenshot shows a web browser window with a dark theme. The address bar displays the URL `ppnhhlpr6eswdfxxytdescni0npxnux.lambda-url.us-east-1.on.aws`. The browser's tab bar shows several open tabs, including "Sajilo Ventures", "Common Intents...", "PVE Protection Pal...", "Data acc", "Untitled Diagram...", "mac command hid...", "Rectangle", "iOS 14 Getting Sta...", "Zoro uses Black S...", and "All Bookmarks". The main content area of the browser displays a JSON response, which is expanded to show its structure. The JSON is formatted with syntax highlighting and includes expand/collapse icons for each level. The response contains details about a person named Suman Adhikari and their friends.

```
{
  "details": {
    "personalDetails": {
      "characteristics": [
        "Slim",
        "Tall-enough",
        "Loves Data2 and has no girlfriend.",
        "CANCER SUPPORT in Data2",
        "AA and WW carry",
        "Voted \"Scaredy-cat\" by Sam Hwang"
      ],
      "personality": [
        "Friendly",
        "Kind",
        "Gamer-frenzy"
      ],
      "name": "Suman Adhikari"
    },
    "friends": [
      {
        "characteristics": [
          "Strong",
          "Fit",
          "Tall"
        ],
        "personality": [
          "Friendly",
          "Kind"
        ],
        "name": "Sagar Maharjan"
      },
      {
        "characteristics": [
          "Slim",
          "Asian-white",
          "Tall",
          "Crazy about noodles"
        ],
        "personality": [
          "Friendly",
          "Kind",
          "Kissa"
        ]
      }
    ]
  }
}
```

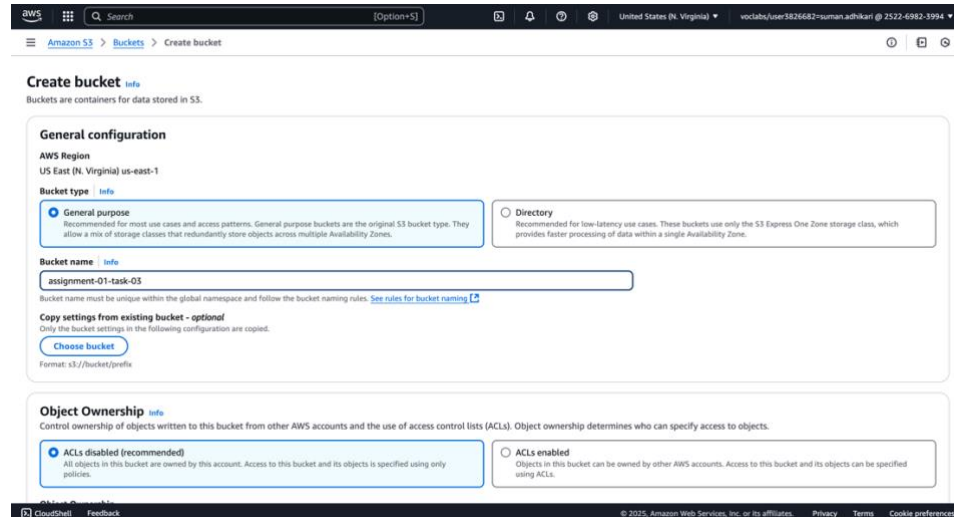

Task 3 – Hosting static site on S3

LINK: <http://assignment-01-task-03.s3-website-us-east-1.amazonaws.com/>

1. Access S3 Dashboard



2. Create S3 bucket



aws [Search] [Option+S] United States (N. Virginia) voclabs/user3826682~suman.adhikari @ 2522-6982-3994

Amazon S3 > Buckets > Create bucket

☒ **ACLs disabled (recommended)**
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

Block Public Access settings for this bucket
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ **Block all public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- ☐ **Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☐ **Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- ☐ **Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☐ **Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws [Search] [Option+S] United States (N. Virginia) voclabs/user3826682~suman.adhikari @ 2522-6982-3994

Amazon S3 > Buckets > Create bucket

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning
☒ Disable
☐ Enable

Tags - optional (0)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

Default encryption [info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [info](#)
☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)
☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key
Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

☐ Disable
☒ Enable

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

3. S3 Bucket is created

aws [Search] [Option+S] United States (N. Virginia) voclabs/user3826682~suman.adhikari @ 2522-6982-3994

Amazon S3 > Buckets

Successfully created bucket "assignment-01-task-01"
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

Account snapshot - updated every 24 hours [View Storage Lens dashboard](#)
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. [Learn more](#)

General purpose buckets | Directory buckets

General purpose buckets (1) [info](#) [All AWS Regions](#)
Buckets are containers for data stored in S3.

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

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	assignment-01-task-01	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 15, 2025, 17:16:59 (UTC-06:00)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

4. Upload files to bucket

The screenshot shows the AWS Management Console for the bucket 'assignment-01-task-03'. The 'Objects' tab is selected, showing a list of 6 objects. The 'Upload' button is highlighted in orange. The table below lists the objects:

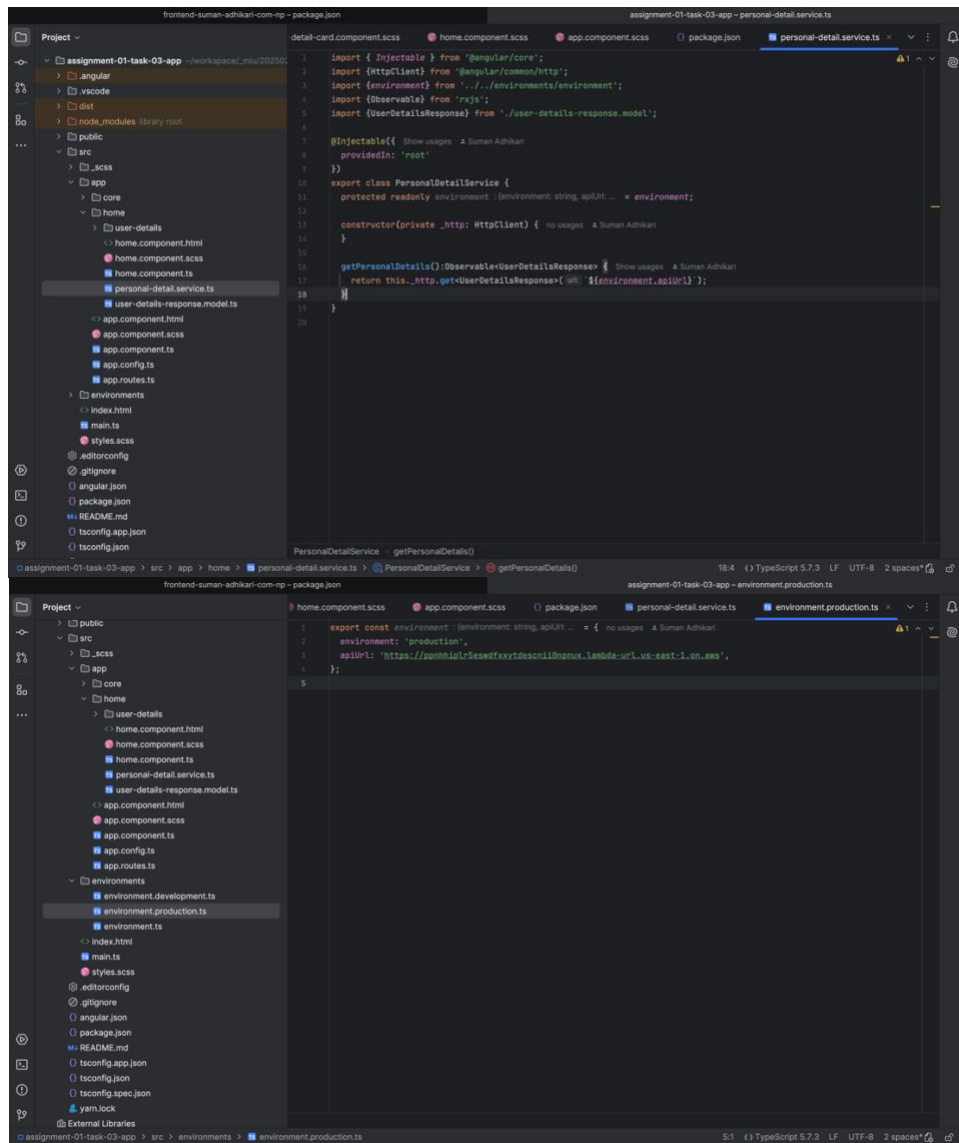
Name	Type	Last modified	Size	Storage class
favicon.ico	ico	February 15, 2025, 17:32:17 (UTC-06:00)	14.7 KB	Standard
index.html	html	February 15, 2025, 17:32:18 (UTC-06:00)	1.1 KB	Standard
main-3K7K4GE.js	js	February 15, 2025, 17:32:18 (UTC-06:00)	219.2 KB	Standard
polyfills-FFH4MD2TL.js	js	February 15, 2025, 17:32:18 (UTC-06:00)	33.7 KB	Standard
profile-images/	Folder	-	-	-
styles-GRC9X0M7.css	css	February 15, 2025, 17:32:18 (UTC-06:00)	1.2 KB	Standard

5. Update bucket policy and enable static hosting

The screenshot shows the AWS Management Console for the bucket 'assignment-01-task-03'. The 'Bucket policy' tab is selected, showing the bucket policy in JSON format. The policy grants 's3:GetObject' permission to the principal 'arn:aws:s3::assignment-01-task-03/*'.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3::assignment-01-task-03/*"
    }
  ]
}
```

6. Code calling Lambda API



The screenshot displays two files in a code editor. The top file, `personal-detail.service.ts`, is an Angular service that uses `HttpClient` to fetch user details from an API. It is decorated with `@Injectable` and provides a default environment of 'test'. The `getPersonalDetails` method returns an `Observable<UserDetailsResponse>` by calling `._http.get<UserDetailsResponse>({url: `${environment.apiUrl}`})`. The bottom file, `environment.production.ts`, defines the environment configuration for production, setting `environment` to 'production' and `apiUrl` to 'https://p9mh1p1r5esedfxytdeconli@pmux.lambda-url.us-east-1.on.aws'.

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { environment } from '../environments/environment';
4 import { Observable } from 'rxjs';
5 import { UserDetailsResponse } from './user-details-response.model';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class PersonalDetailsService {
11   protected readonly environment: (environment: string, apiUrl: string) = environment;
12
13   constructor(private _http: HttpClient) {}
14
15   getPersonalDetails(): Observable<UserDetailsResponse> {
16     return this._http.get<UserDetailsResponse>({url: `${environment.apiUrl}`});
17   }
18 }
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
1 export const environment = {
2   environment: 'production',
3   apiUrl: 'https://p9mh1p1r5esedfxytdeconli@pmux.lambda-url.us-east-1.on.aws',
4 };
5
```

7. Preview

