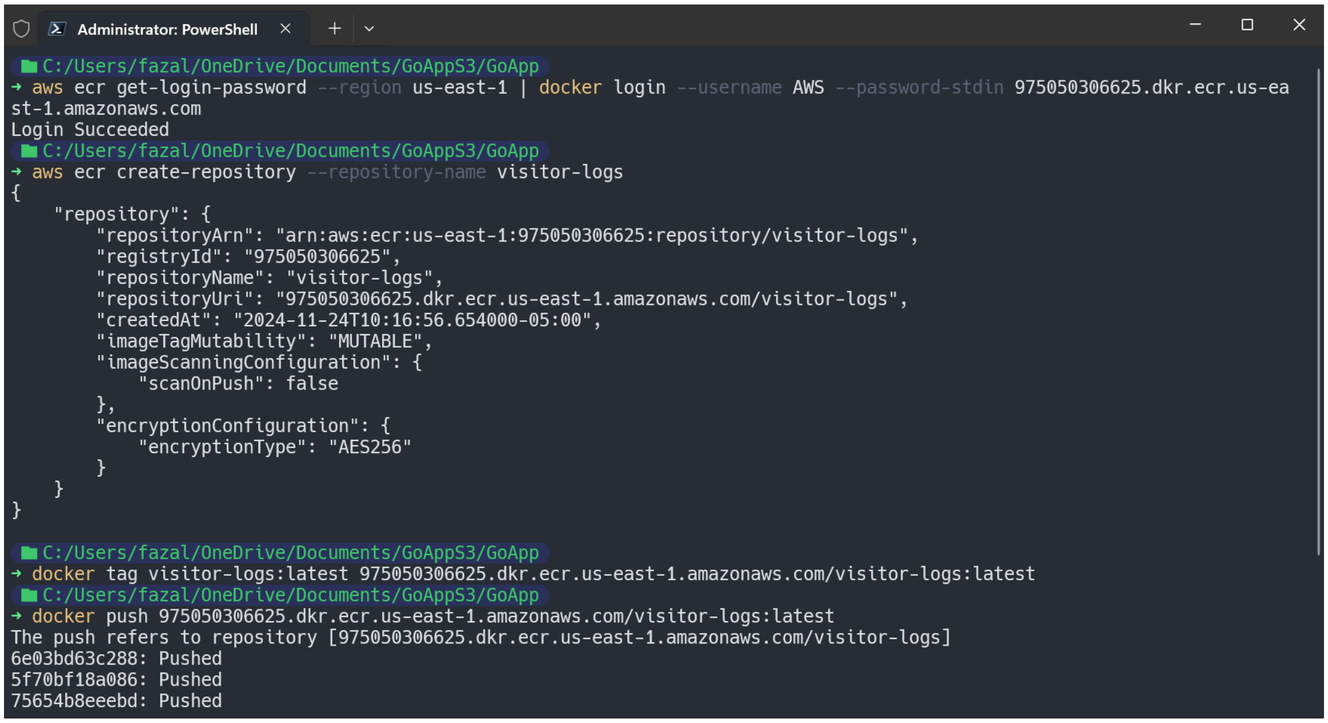
Deployment Solution 2: Using S3, Lambda and DynamoDB.

Step 1: Update the Go Application:

- Make necessary changes in the main.go to use DynamoDB instead of MongoDB.
- Create a Dockerfile to package your Go application as a container.
- Build the container Image and push it to AWS ECR Repository:
  docker build --platform linux/amd64 -t visitor-logs .

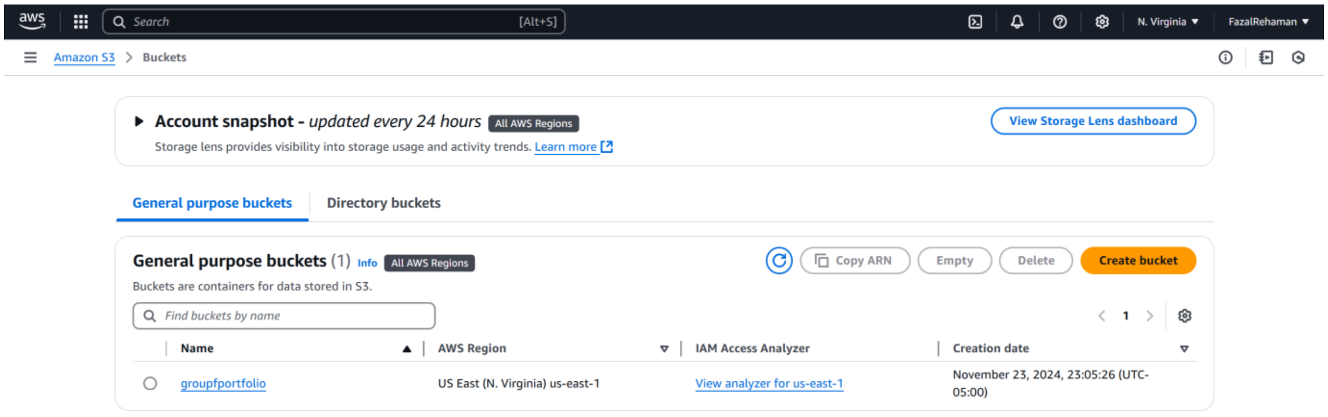


```
C:/Users/fazal/OneDrive/Documents/GoAppS3/GoApp
→ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 975050306625.dkr.ecr.us-ea
st-1.amazonaws.com
Login Succeeded
C:/Users/fazal/OneDrive/Documents/GoAppS3/GoApp
→ aws ecr create-repository --repository-name visitor-logs
{
    "repository": {
        "repositoryArn": "arn:aws:ecr:us-east-1:975050306625:repository/visitor-logs",
        "registryId": "975050306625",
        "repositoryName": "visitor-logs",
        "repositoryUri": "975050306625.dkr.ecr.us-east-1.amazonaws.com/visitor-logs",
        "createdAt": "2024-11-24T10:16:56.654000-05:00",
        "imageTagMutability": "MUTABLE",
        "imageScanningConfiguration": {
            "scanOnPush": false
        },
        "encryptionConfiguration": {
            "encryptionType": "AES256"
        }
    }
}

C:/Users/fazal/OneDrive/Documents/GoAppS3/GoApp
→ docker tag visitor-logs:latest 975050306625.dkr.ecr.us-east-1.amazonaws.com/visitor-logs:latest
C:/Users/fazal/OneDrive/Documents/GoAppS3/GoApp
→ docker push 975050306625.dkr.ecr.us-east-1.amazonaws.com/visitor-logs:latest
The push refers to repository [975050306625.dkr.ecr.us-east-1.amazonaws.com/visitor-logs]
6e03bd63c288: Pushed
5f70bf18a086: Pushed
75654b8eeebd: Pushed
```

Step 2: Set Up the S3 Bucket for Static Hosting

1. **Create an S3 Bucket**:
   - Go to the AWS Management Console.
   - Navigate to S3 and click **Create bucket**.
   - Provide a unique bucket name (groupfportfolio).

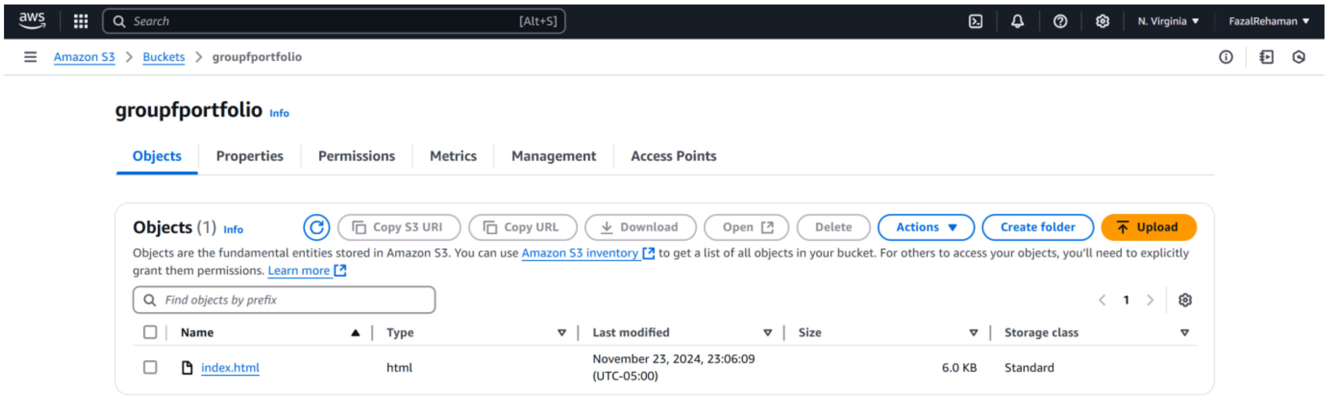


Account snapshot - *updated every 24 hours* All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
groupfportfolio	US East (N. Virginia) us-east-1	View analyzer for us-east-1	November 23, 2024, 23:05:26 (UTC-05:00)

2. **Upload the index.html File**:
   - Go to the **Objects** section of your bucket.
   - Click **Upload** and add the index.html file.



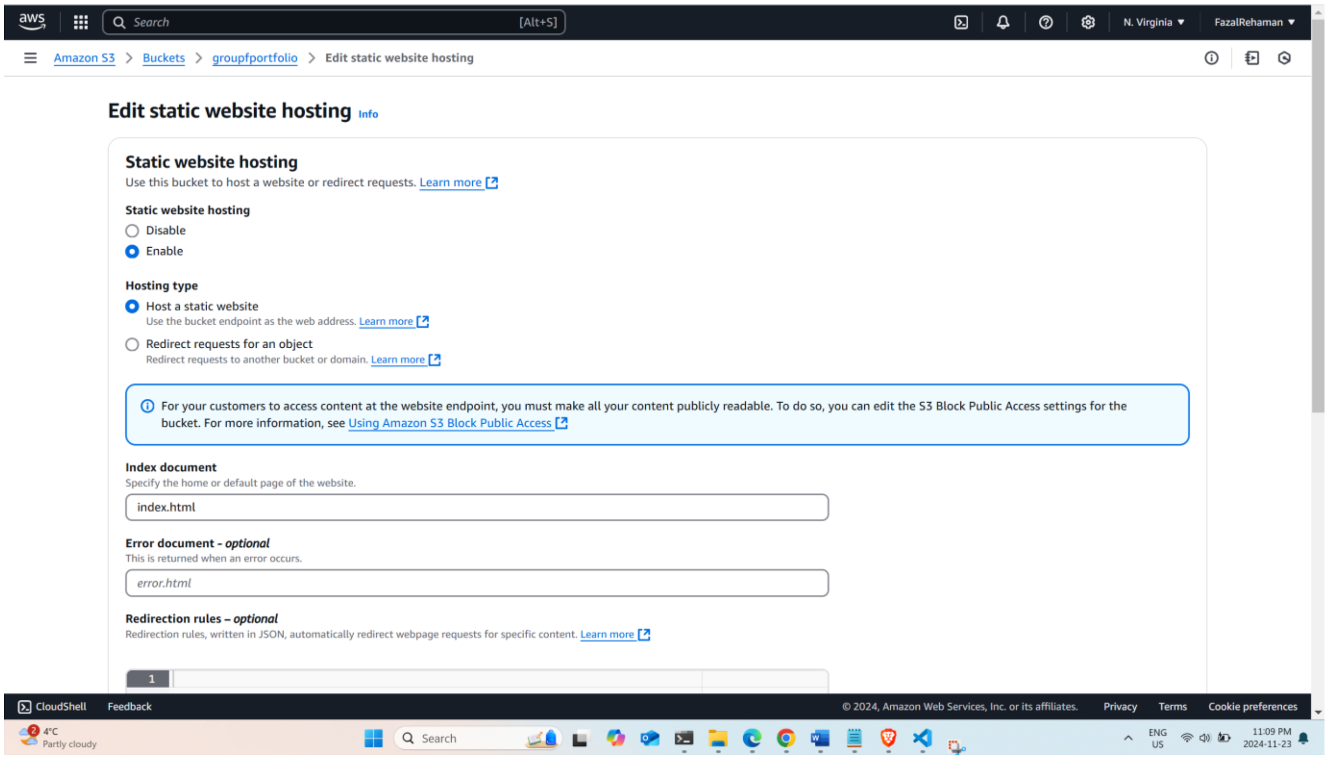
groupfportfolio Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
index.html	html	November 23, 2024, 23:06:09 (UTC-05:00)	6.0 KB	Standard

3. **Configure the Bucket for Static Hosting**:
   - Go to the **Properties** tab.
   - Scroll to **Static website hosting** and enable it.
   - Set index.html as the **Index document**.
   - Note the **Endpoint URL**.


aws
Search [Alt+S]
N. Virginia
FazalRehaman
Amazon S3 > Buckets > groupfportfolio > Edit static website hosting

Edit static website hosting Info

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting
- Disable
- [✓ Enable]

Hosting type
- [✓ Host a static website]
  Use the bucket endpoint as the web address. Learn more
- Redirect requests for an object
  Redirect requests to another bucket or domain. Learn more

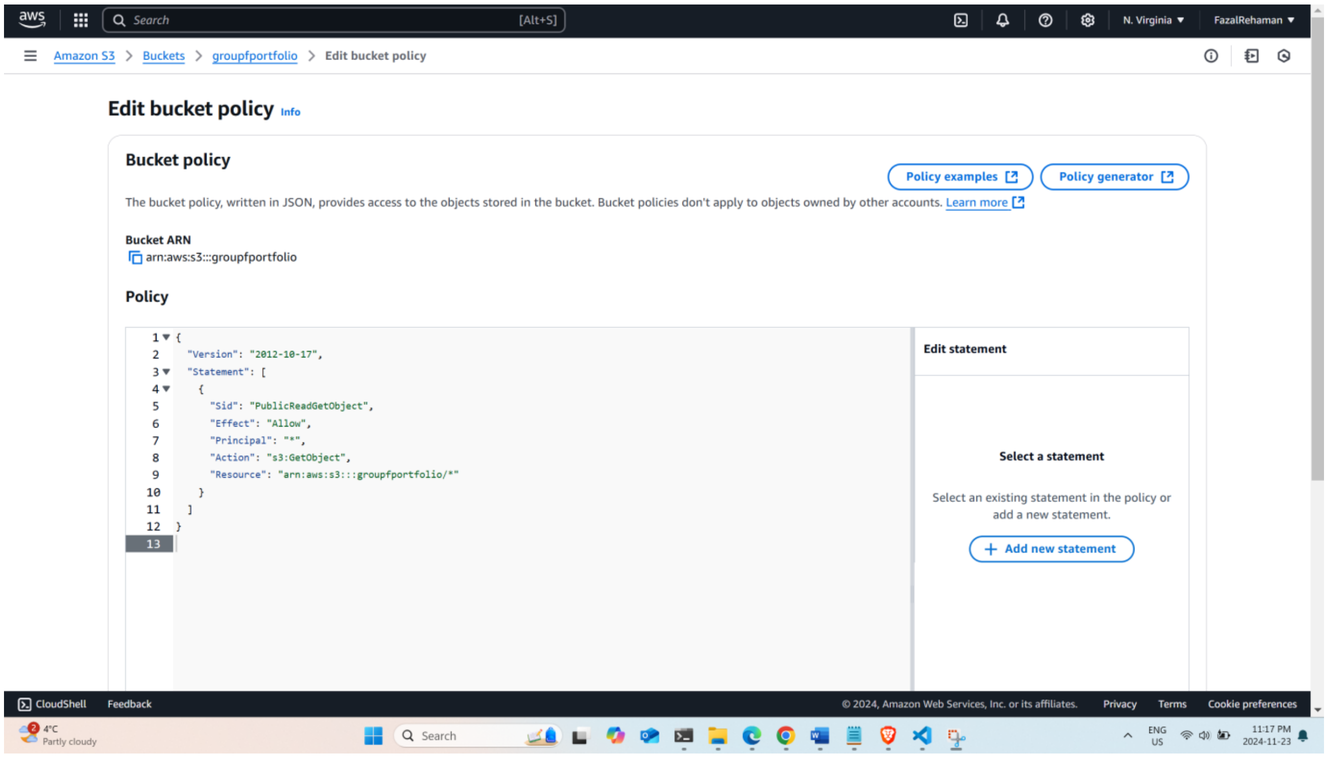
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access

Index document
Specify the home or default page of the website.
index.html

Error document - *optional*
This is returned when an error occurs.
error.html

Redirection rules – *optional*
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. Learn more

4. **Update Bucket Permissions**:
   - Go to the **Permissions** tab.
   - Add a **Bucket Policy**:


Amazon S3 > Buckets > groupfportfolio > Edit bucket policy

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN
arn:aws:s3:::groupfportfolio

Policy

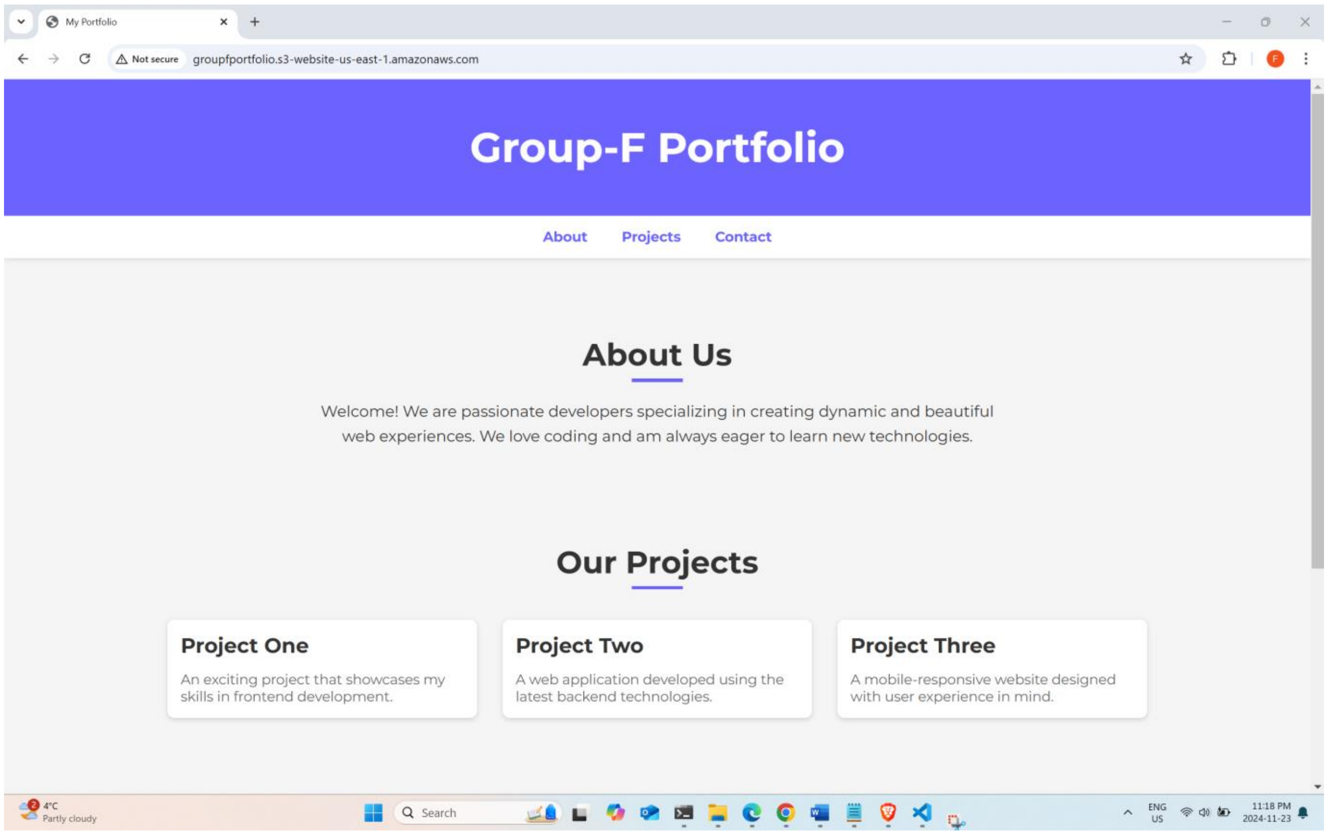
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::groupfportfolio/*"
    }
  ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

Verify that your static website is accessible:


groupfportfolio.s3-website-us-east-1.amazonaws.com

Group-F Portfolio

About Projects Contact

About Us

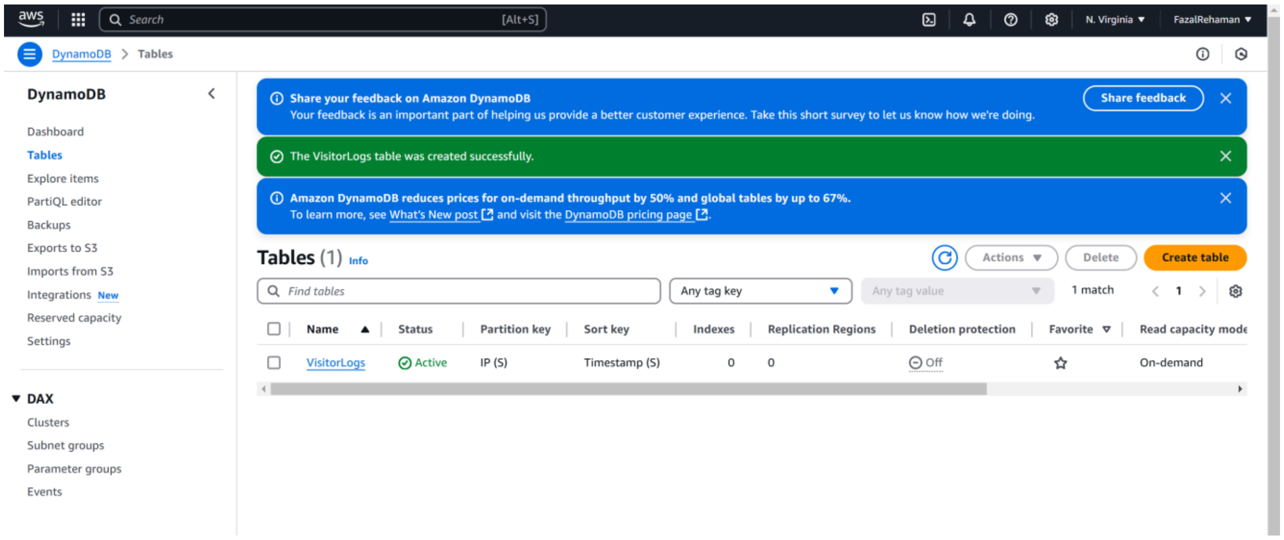
Welcome! We are passionate developers specializing in creating dynamic and beautiful web experiences. We love coding and am always eager to learn new technologies.

Our Projects

Project One	**Project Two**	**Project Three**
An exciting project that showcases my skills in frontend development.	A web application developed using the latest backend technologies.	A mobile-responsive website designed with user experience in mind.

Step 3: Set Up the DynamoDB Table

1. Navigate to the **DynamoDB** service.
2. Create a new table:
   - Table Name: VisitorLogs
   - Partition Key: IP (String)
   - Sort Key: Timestamp (String).



Share your feedback on Amazon DynamoDB
Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

The VisitorLogs table was created successfully.

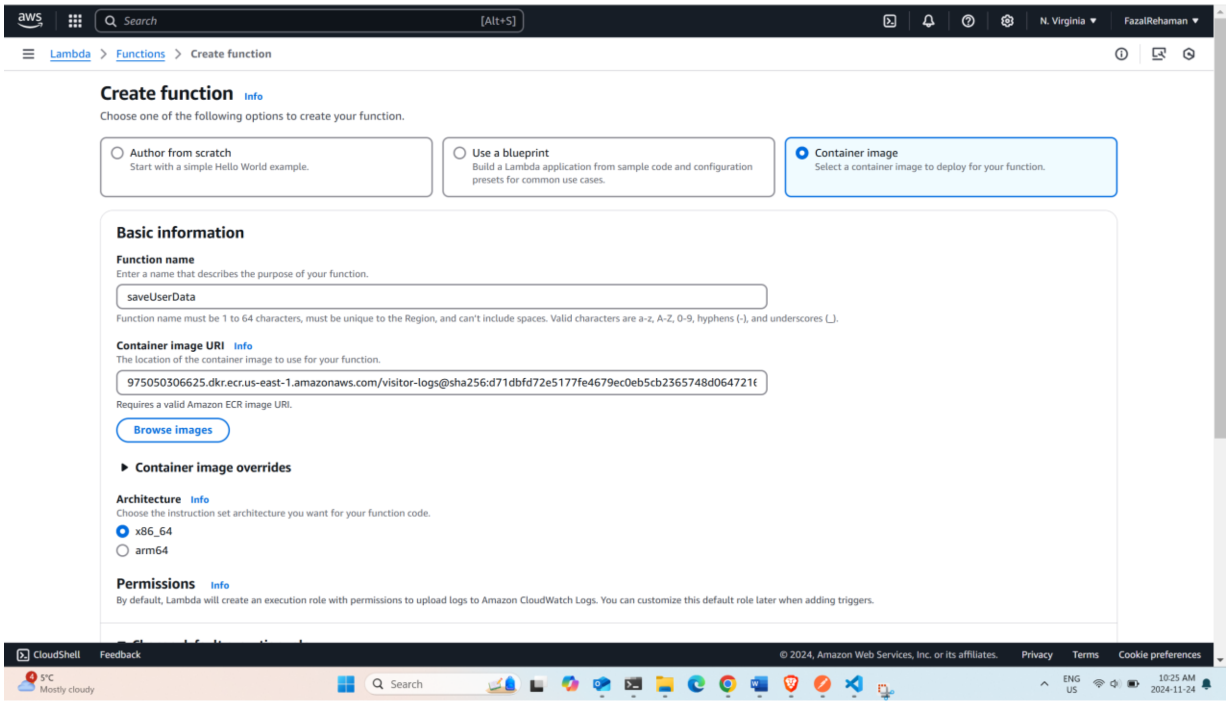
Amazon DynamoDB reduces prices for on-demand throughput by 50% and global tables by up to 67%.
To learn more, see What's New post and visit the DynamoDB pricing page.

Tables (1)

	Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode
☐	VisitorLogs	Active	IP (S)	Timestamp (S)	0	0	Off	☆	On-demand

Step 4: Deploy the Container to AWS Lambda

1. **Create a Lambda Function**:
   - Go to the **Lambda** service.
   - Click **Create function** and choose **Container image** as the runtime.
   - Function Name: saveUserData
   - Image: select the ECR image you pushed.



Create function
Choose one of the following options to create your function.

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.
saveUserData
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Container image URI
The location of the container image to use for your function.
975050306625.dkr.ecr.us-east-1.amazonaws.com/visitor-logs@sha256:d71dbfd72e5177fe4679ec0eb5cb2365748d064721
Requires a valid Amazon ECR image URI.

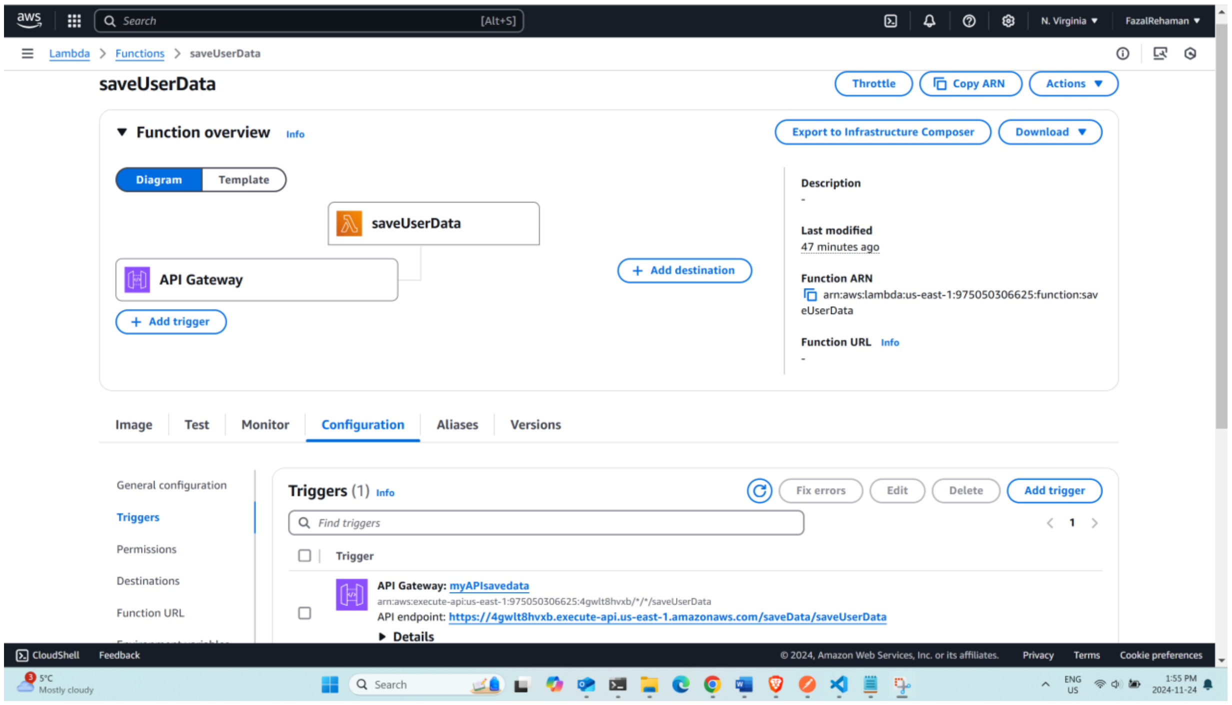
Container image overrides

Architecture
Choose the instruction set architecture you want for your function code.
[✓ x86_64] arm64

Permissions
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.

2. **Add IAM Permissions**:
   - Attach the AmazonDynamoDBFullAccess policy to the Lambda function's execution role.

3. **Set Up an API Gateway Trigger**:
   - Navigate to **Triggers** and add an API Gateway trigger.
   - Create a new REST API with a POST method and enable CORS.
   - Deploy the API and note the **Invoke URL**.



saveUserData

Function overview

Description: -

Last modified: 47 minutes ago

Function ARN: arn:aws:lambda:us-east-1:975050306625:function:saveUserData

Function URL: -

Triggers (1)

API Gateway: myAPIsavedata
arn:aws:execute-api:us-east-1:975050306625:4gwlt8hvxb/*/*/saveUserData
API endpoint: https://4gwlt8hvxb.execute-api.us-east-1.amazonaws.com/saveData/saveUserData

Step 5: Test the Setup

1. Open the web page from different devices and check if the IPs are saved in DynamoDB.

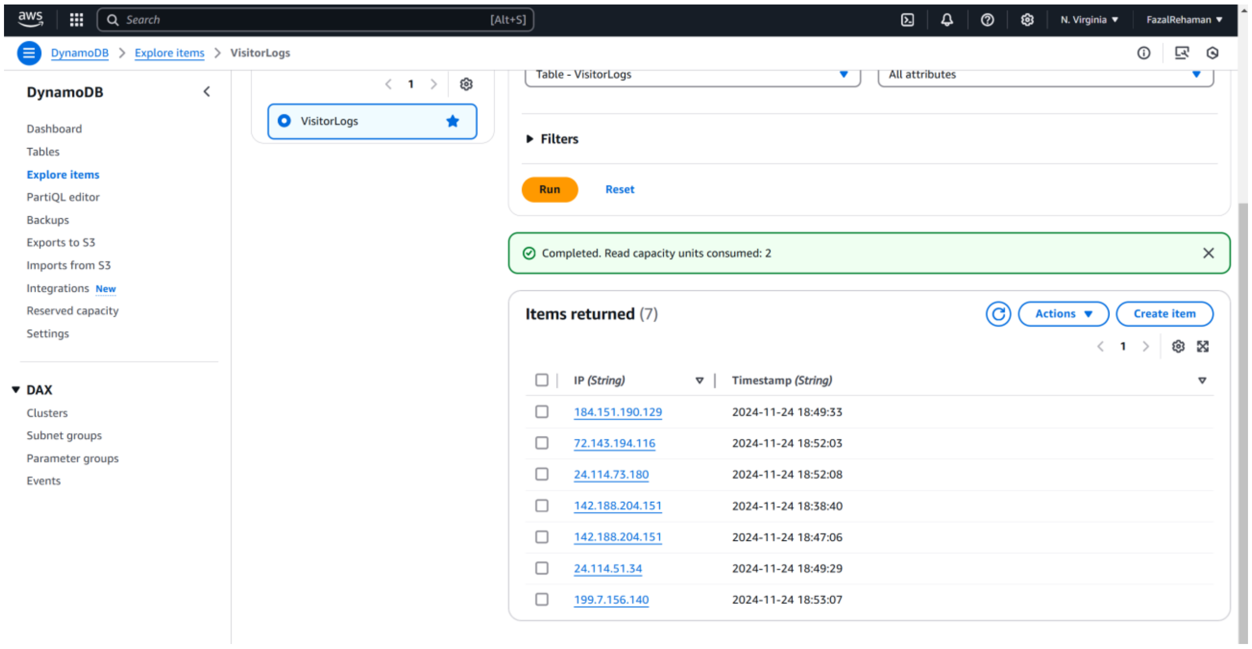


Table - VisitorLogs

Completed. Read capacity units consumed: 2

Items returned (7)

IP (String)	Timestamp (String)
184.151.190.129	2024-11-24 18:49:33
72.143.194.116	2024-11-24 18:52:03
24.114.73.180	2024-11-24 18:52:08
142.188.204.151	2024-11-24 18:38:40
142.188.204.151	2024-11-24 18:47:06
24.114.51.34	2024-11-24 18:49:29
199.7.156.140	2024-11-24 18:53:07