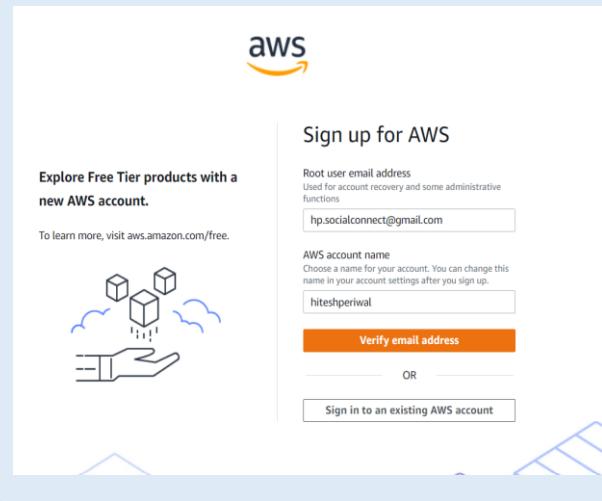


# CREATING AN AWS ACCOUNT

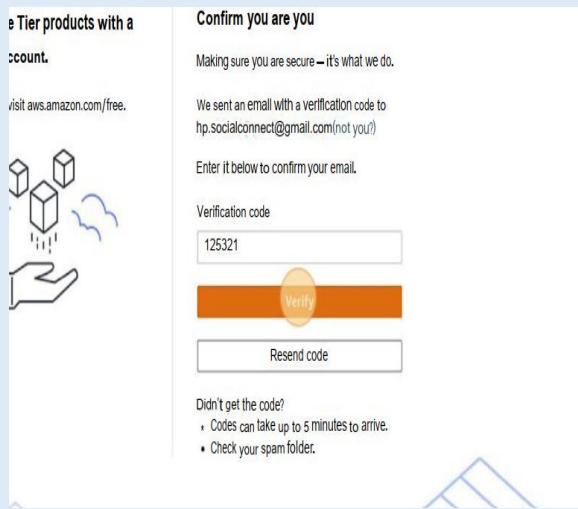
1. Visit [aws.amazon.com](https://aws.amazon.com) and click on “Create an AWS Account”



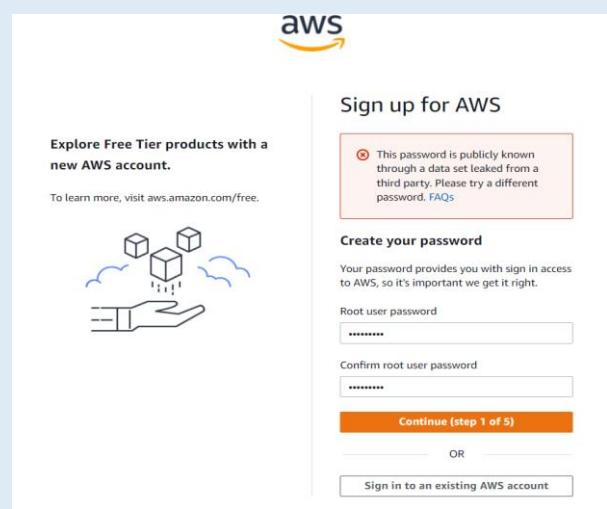
2. Enter the root user email address ,AWS account name and click on “Verify email address”



3. Enter the verification code received in the given email address and click on “Verify”



4. Set your account’s password



## 5. Fill in your contact details and click on continue

**Sign up for AWS**

**Contact Information**

How do you plan to use AWS?

- Business - for your work, school, or organization
- Personal - for your own projects

Who should we contact about this account?

Full Name: Hitesh

Phone Number: +91 222-333-4444

Country or Region: India

Address: Howrah

Apartment, suite, unit, building, floor, etc.: Howrah

City: Howrah

State, Province, or Region: West Bengal

Postal Code: 711202

Customers with an Indian contact address are served by Amazon Web Services India Private Limited, the local seller for AWS services in India.

I have read and agree to the terms of the AWS Customer Agreement [\[Link\]](#)

**Continue (step 2 of 5)**

**Free Tier offers**

All AWS accounts can explore 3 different types of free offers, depending on the product used.

- Always free**: Never expires
- 12 months free**: Start from initial sign-up date
- Trials**: Start from service activation date

## 6. Enter your billing information and click on “Verify and Continue”

**Sign up for AWS**

**Billing Information**

Credit or Debit card number:

AWS accepts all major credit and debit cards. To learn more about payment options, review our [FAQ](#).

Expiration date: Month: Year:

Cardholder's name:

CVV:

Billing address:

- Use my contact address  
addr: sf dsf 711100 IN
- Use a new address

Do you have a PAN? Permanent Account Number (PAN) is a ten-digit alphanumeric number issued by the Indian Income Tax Department. This 10-digit number is printed on the front of your PAN card.

- Yes
- No

You can go on the Tax Settings Page on Billing and Cost Management Console to update your PAN information.

**Verify and Continue (step 3 of 5)**

You might be redirected to your bank's website to authorize the verification charge.

## 7. Confirm and verify your phone number

**Sign up for AWS**

**Confirm your identity**

Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code.

How should we send you the verification code?

- Text message (SMS)
- Voice call

Country or region code: United States (+1)

Mobile phone number:

## 8. Choose Basic Support plan and hence your AWS account is created

You can change your plan anytime in the AWS Management Console.

<input checked="" type="radio"/> <b>Basic support - Free</b>	<input type="radio"/> <b>Developer support - From \$29/month</b>	<input type="radio"/> <b>Business support - From \$100/month</b>
<ul style="list-style-type: none"> <li>• Recommended for new users just getting started with AWS</li> <li>• 24x7 self-service access to AWS resources</li> <li>• For account and billing issues only</li> <li>• Access to Personal Health Dashboard &amp; Trusted Advisor</li> </ul>	<ul style="list-style-type: none"> <li>• Recommended for developers experimenting with AWS</li> <li>• Email access to AWS Support during business hours</li> <li>• 12 (business)-hour response times</li> </ul>	<ul style="list-style-type: none"> <li>• Recommended for running production workloads on AWS</li> <li>• 24x7 tech support via email, phone, and chat</li> <li>• 1-hour response times</li> <li>• Full set of Trusted Advisor best-practice recommendations</li> </ul>

**Need Enterprise level support?**

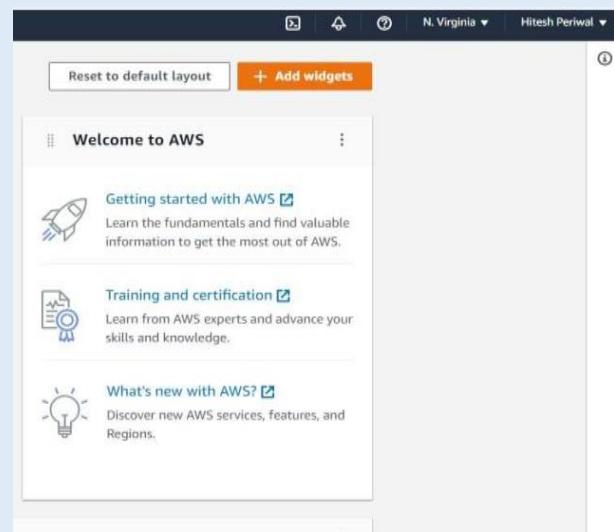
From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. Learn more [\[Link\]](#)

# CONFIGURE A BUDGET AND SETTING A THRESHOLD VALUE

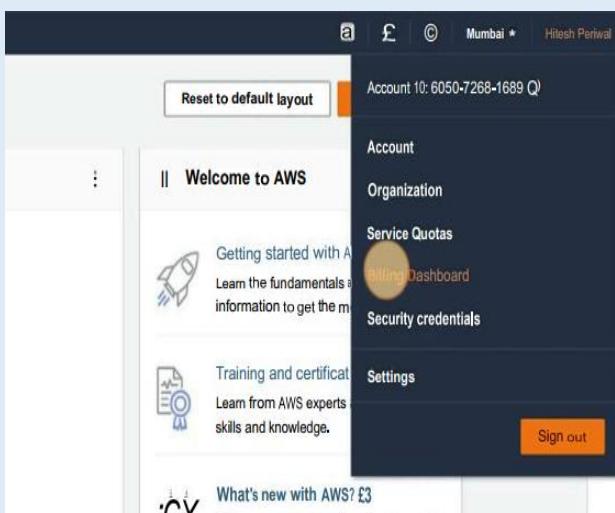
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in



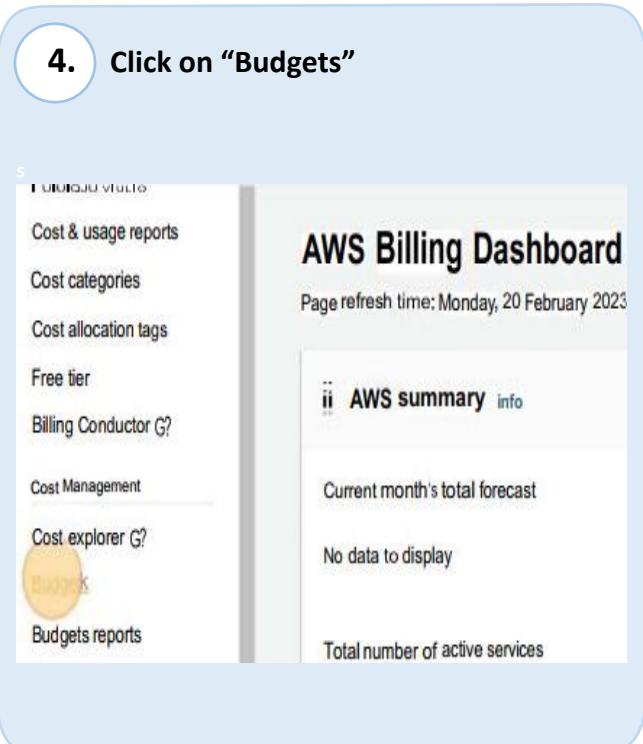
2. Click on "Hitesh Periwal"



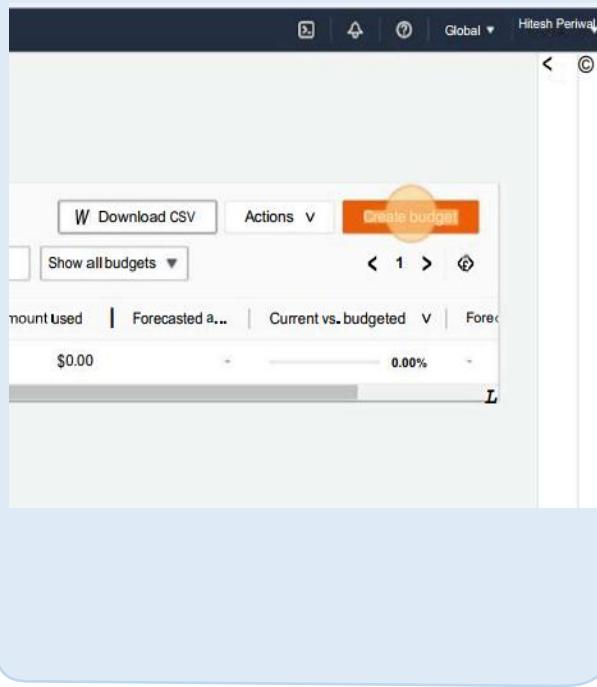
3. Click on "Billing Dashboard"



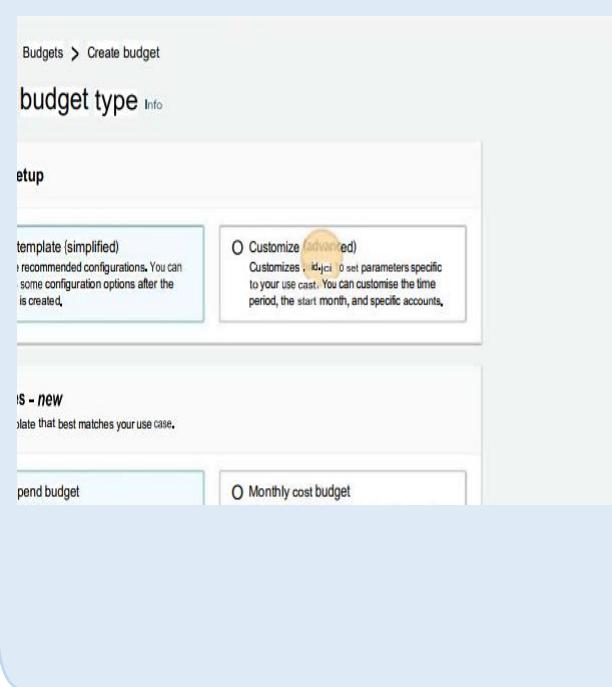
4. Click on "Budgets"



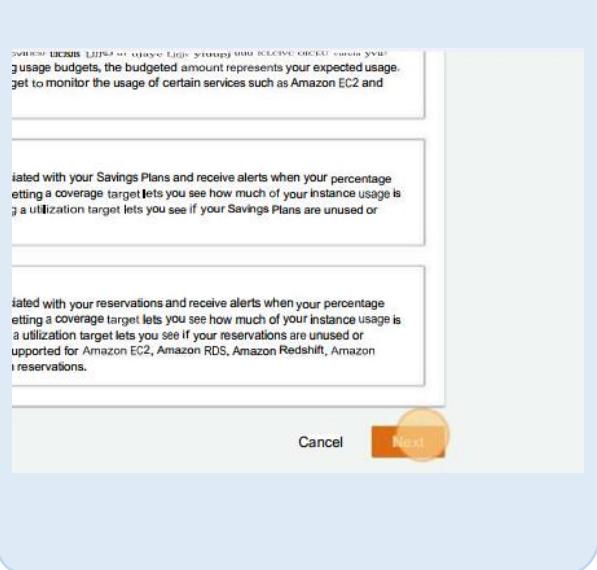
## 5. Click on “Create Budget”



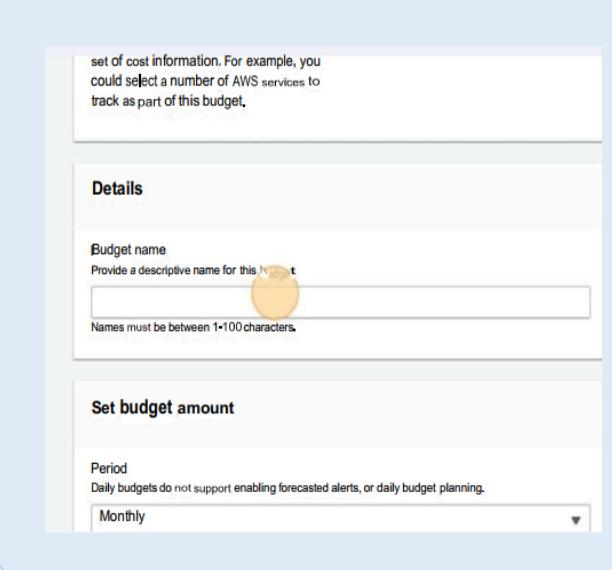
## 6. Select “Customize (advanced)“



## 7. Click on “Next”



## 8. Give your budget a name



## 9. Enter the budgeted amount in \$

Start month: Feb 2023

Budgeting method: **Fixed** (info)

Create a budget that tracks against a single monthly budgeted amount.

Enter your budgeted amount (\$): Last month's cost: \$100

Budget scope: **info**  
Add filtering and use advanced options to narrow the set of cost information tracked as part of this budget.

Scope options:  All AWS services (Recommended) |  Filter specific AWS cost

## 10. Click on “Next”

Budgets > Create budget

**budget type** [Info](#)

**setup**

template (simplified)  
Recommended configurations. You can some configuration options after the is created.

Customize (advanced)  
Customizes . You can customize the time period, the start month, and specific accounts.

**use case** [IS - new](#)  
state that best matches your use case.

**pend budget**  Monthly cost budget

## 11. Click on “Add an alert threshold”

SNS and AWS Chatbot.

**Budget amount**

Your budgeted amount: \$1.00  
To change your budgeted amount, go back to step 2.

( No alert thresholds created.) **Add an alert threshold**

Cancel Previous Next

## 12. Set the threshold and then specify the email recipients you want to notify when threshold has exceeded

**Threshold**  
When should this alert be triggered?

80 % of budgeted amount

**Trigger**  
How should this alert be triggered?

Actual

Summary: When your actual cost is greater than 80.00% (\$0.80) of your budgeted amount (\$1.00), the alert threshold will be exceeded.

**Notification preferences**  
Select one or more notification preferences to receive alerts.

**Email recipients**  
Specify the email recipients you want to notify when the threshold has exceeded.  
Separate email addresses using commas

Maximum number of email recipients is 10.

▶ Amazon SNS Alerts - [Optional info](#)

**13. Click on “Create Budget”**

References to receive alerts,

Want to notify when the threshold has exceeded.

Alert frequency: Daily

Alert message: Actual cost has exceeded the budget by 10%.

Actual cost vs. Budget

Feb 2022 May 2022 Aug 2022

Actual cost    Budget    Alert if

View in AWS Cost Explorer (3)

Alerts

Actual cost > 80% | No actions

Cancel Previous Next

**14. Select “Customize (advanced)“**

type of action you want

attached

Email recipients  
hp.socialconnect@gmail.com

Amazon SNS  
Not configured

Feb 2022 May 2022 Aug 2022

Actual cost    Budget    Alert if

View in AWS Cost Explorer (3)

Alerts

Actual cost > 80% | No actions

Cancel Previous Next

**15. Click on “Create budget ” and hence the budget is created**

optional

Edit

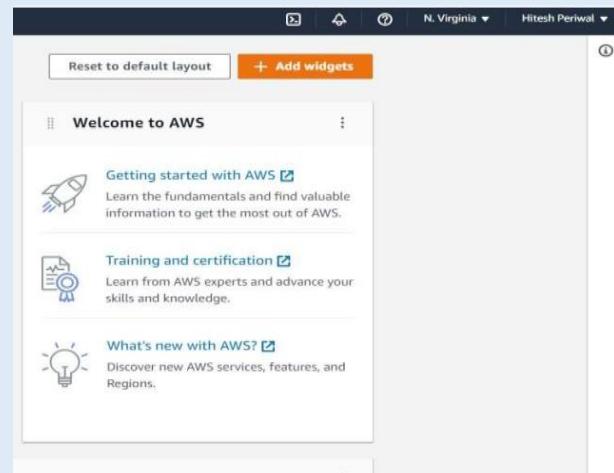
Cancel Previous Create budget Next

# ACTIVATION OF MFA SECURITY

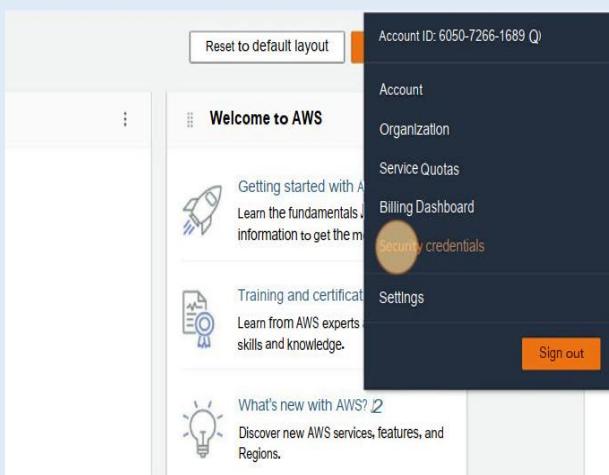
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in



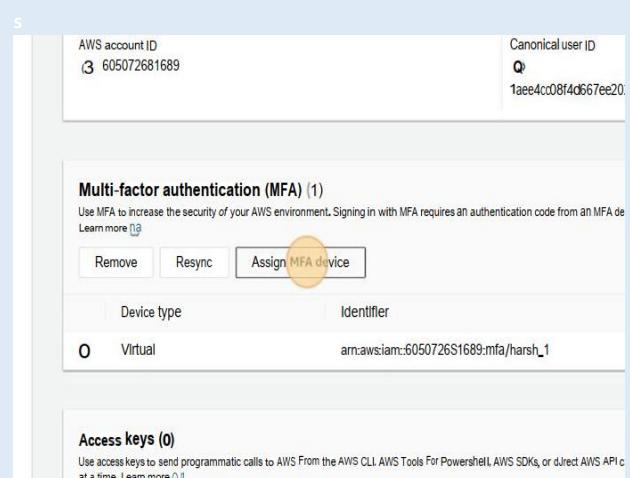
2. Click on "Hitesh Periwal"



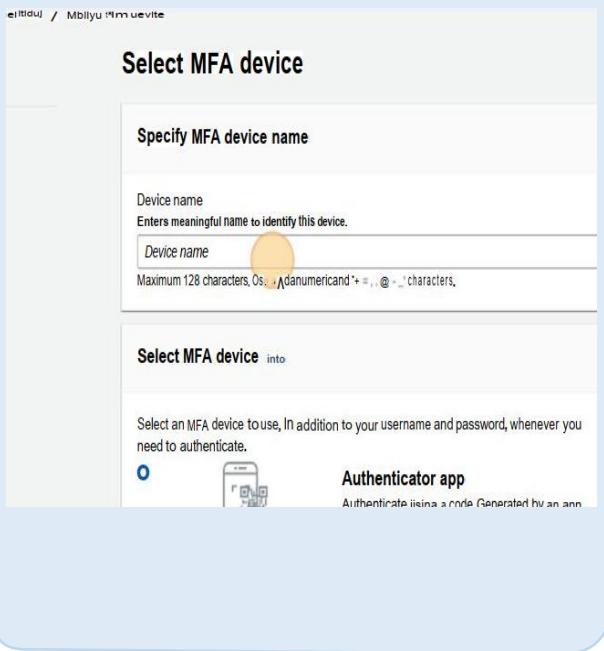
3. Click on "Security credentials"



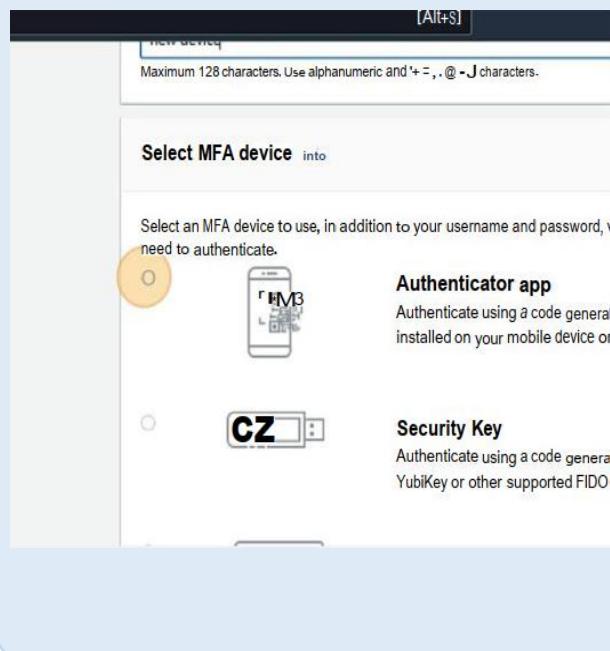
4. Click on "Assign MFA device"



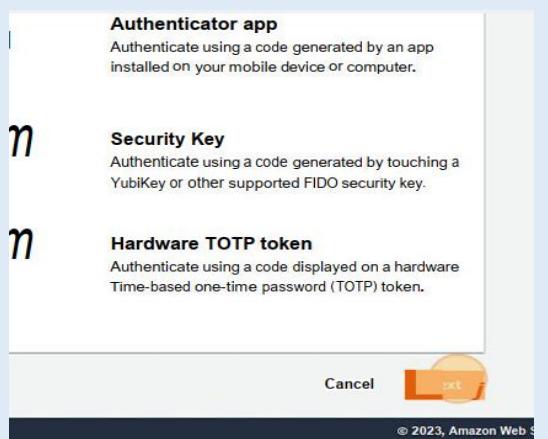
## 5. Enter the Device name



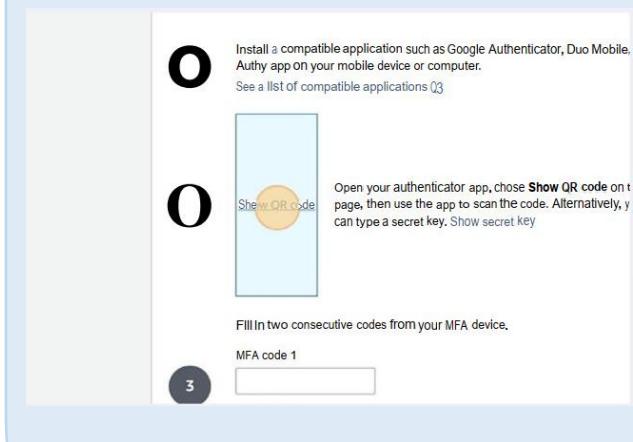
## 6. Click on the "Authenticator app" option



## 7. Click on "Next"



## 8. Download Authenticator from Google Play Store on your Smart Phone. Then click on PLUS button. Then click on "Scan a QR Code". Then come back to Desktop Screen. Click "Show QR code"



**9. Get two consecutive MFA code from smartphone and enter in the respective fields**



Show QR code on this page, then use the app to scan the code. Alternatively, you can type a secret key. Show secret key

Fill in two consecutive codes from your MFA device.

MFA code 1

MFA code 7  123456

Cancel Previous Add MFA

**10. Click on “Add MFA” and thus MFA has been added**



Show QR code on this page, then use the app to scan the code. Alternatively, you can type a secret key. Show secret key

Fill in two consecutive codes from your MFA device.

Code 1

Code 2

Cancel Previous Add MFA

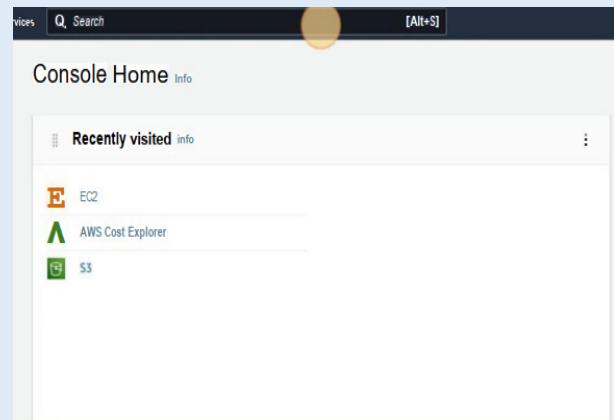
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy

# CREATE AN IAM USER GIVING FULL S3 ACCESS

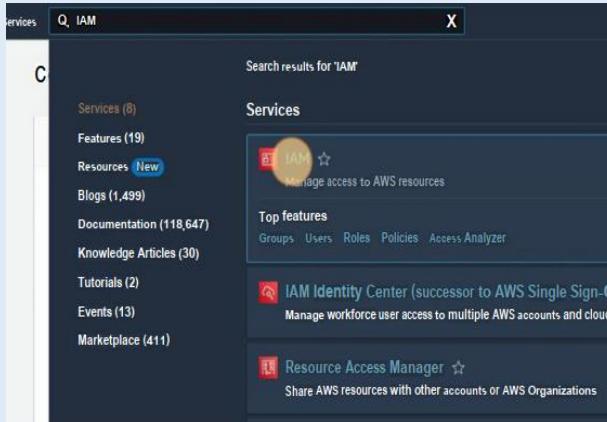
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in



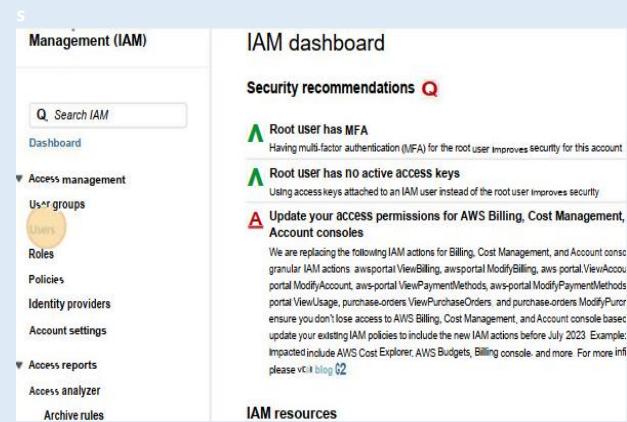
2. Click on the search field and type "IAM" in the search field



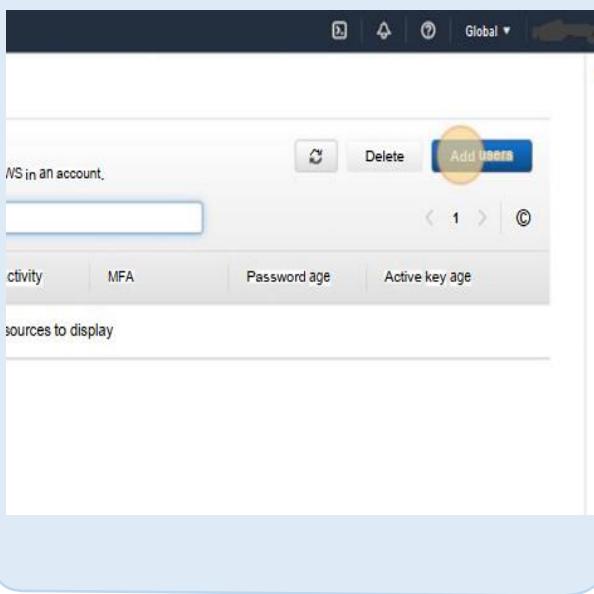
3. Click on "IAM"



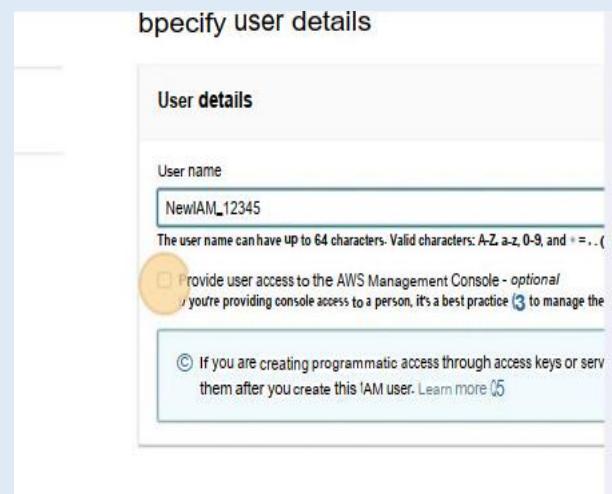
4. Click on "Users"



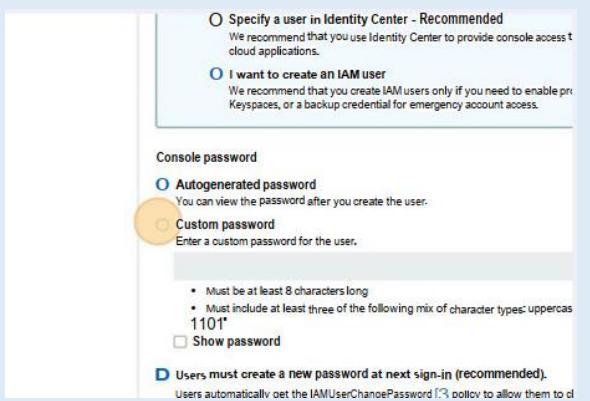
## 5. Click on “Add users”



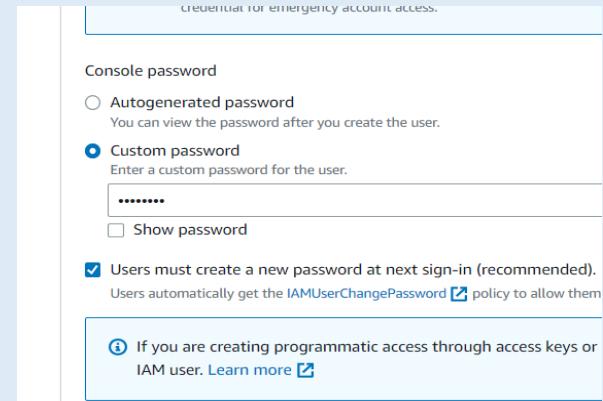
**6.** Enter the user name and click on the checkbox to provide user access to AWS Management Console



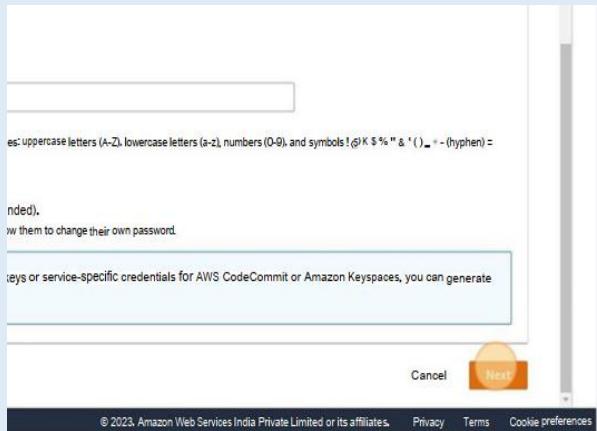
7. Select “I want to create an IAM user” as well as “custom password”



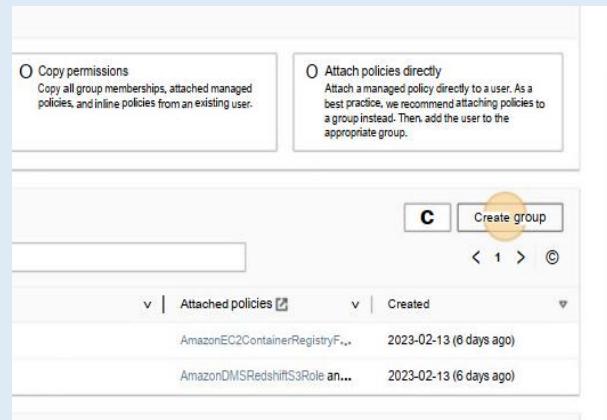
- Set the password and click the checkbox “Users must create a new password at next sign-in”



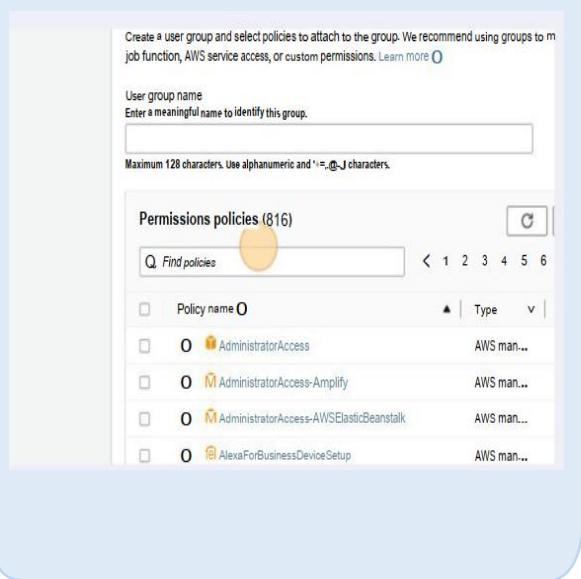
## 9. Click on “Next”



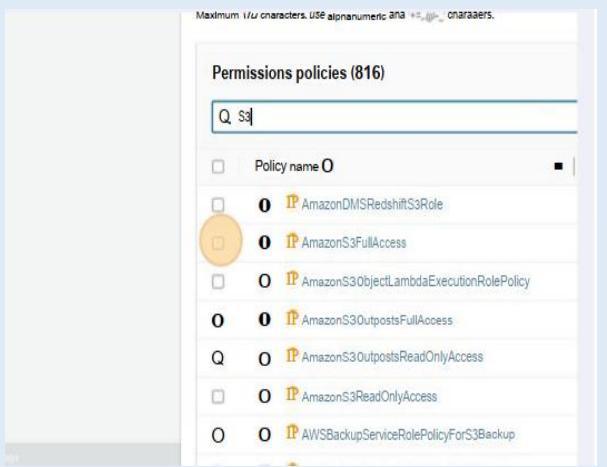
## 10. Click on “Create group”



## 11. Click the “find policies” field and type “S3”



## 12. Click the checkbox “AmazonS3FullAccess”



### 13. Enter a user group name

Create user group

Create a user group and select policies to attach to the group. We recommend using groups by job function, AWS service access, or custom permissions. [Learn more](#)

User group name  
Enter a meaningful name to identify this group.

Maximum 128 characters. Use alphanumeric and '-' characters.

Permissions policies (1/816)

Policy name	Type
AmazonDMSRedshiftS3Role	AWS managed

### 14. Click on “Create user group”

A	Type	V	Used as	V	D...
Role	AWS man...	Permissio...	P...		
aExecutionRolePolicy	AWS man...	Permissio...	P...		
Access	AWS man...	None	P...		
dOnlyAccess	AWS man...	None	P...		
ess	AWS man...	None	P...		
PolicyForS3Backup	AWS man...	None	P...		
PolicyForS3Restore	AWS man...	None	P...		
StorageManagementAnal...	AWS man...	None	P...		

[Create group](#)

Created  
2023-02-13 (6 days ago)  
2023-02-13 (6 days ago)

[Cancel](#) [Previous](#) [Next](#)

### 15. Click on the checkbox “S3FullAccess”

group. We recommend using groups to manage user permissions by job function.

policies, and in

User groups (3)

Group name	Users
hosting_group	0
S3FullAccess	0
storage_group	0

### 16. Click on “Next”

G Create group

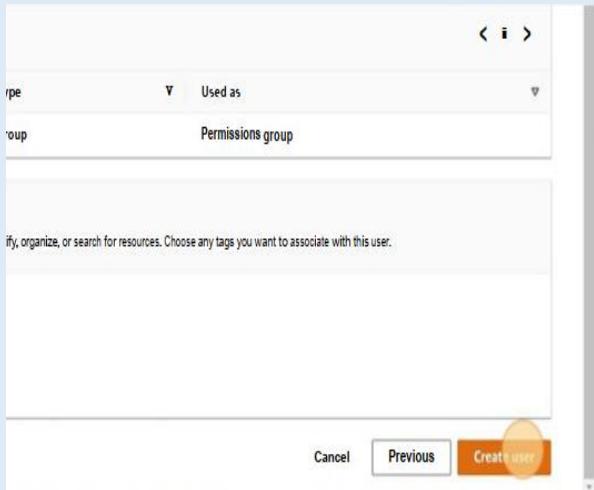
< 1 > 0

Users	Attached policies	Created
0	AmazonEC2ContainerRegistryF...	2023-02-13 (6 days ago)
0	AmazonS3FullAccess	2023-02-20 (Now)
0	AmazonDMSRedshiftS3Role an...	2023-02-13 (6 days ago)

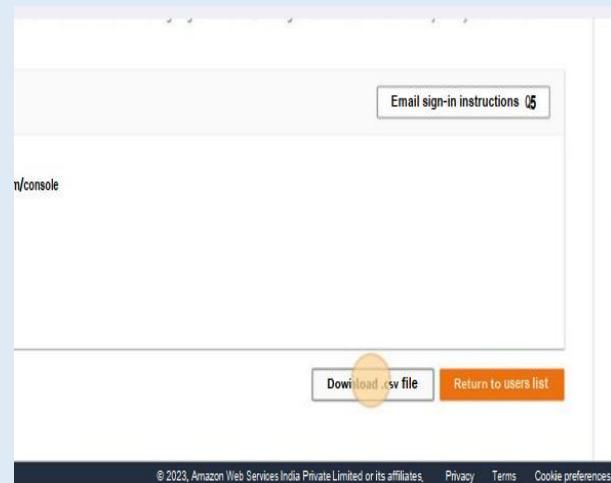
permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

[Cancel](#) [Previous](#) [Next](#)

**17. Click on “Create user”**



**18. Click on “Download .csv file”**



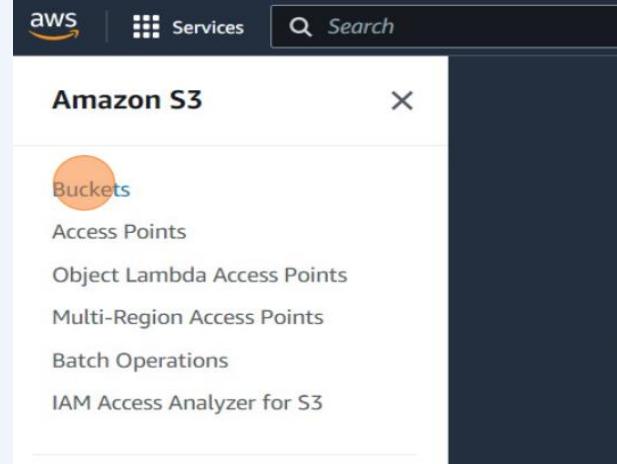
**17. IAM user is thus created**

# CREATE A PRIVATE BUCKET IN AWS.UPLOAD A FILE AND CHECK BY PRESIGNED URL THAT YOU CAN ACCESS THE FILE OR NOT.

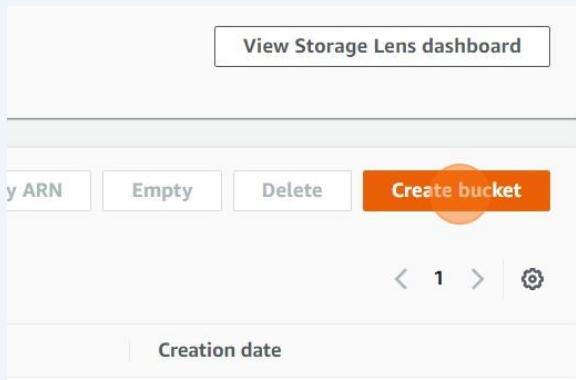
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in.Click on the search field and type “s3” in the search field and select S3



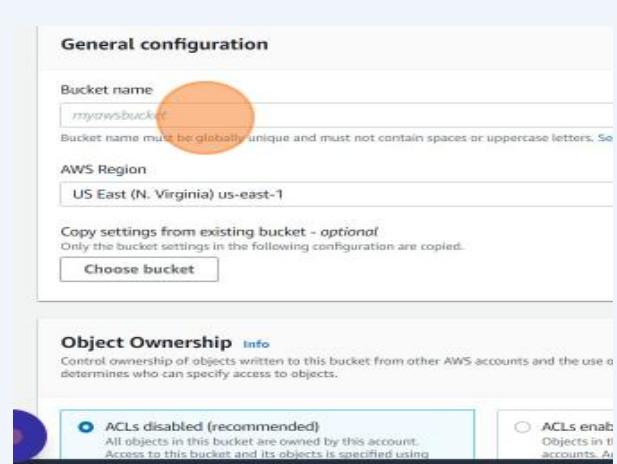
2. Click on “Buckets”



3. Click on “Create Bucket”



4. Click the "Bucket name" field and type "hiteshpvtbucket" and keep all other options unchanged.



## 5. Click on “Create bucket”

the bucket key reduces encryption costs by lowering calls to AWS KMS.

ders to the bucket, and configure additional bucket settings.

Cancel

Create bucket

## 6. Click on “hiteshpvtbucket”

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

### Buckets (1) [Info](#)

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

Find buckets by name

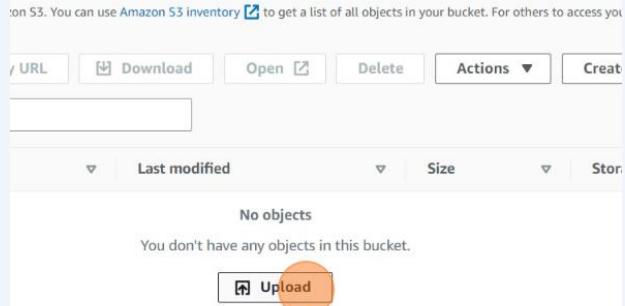
Name

AWS Region

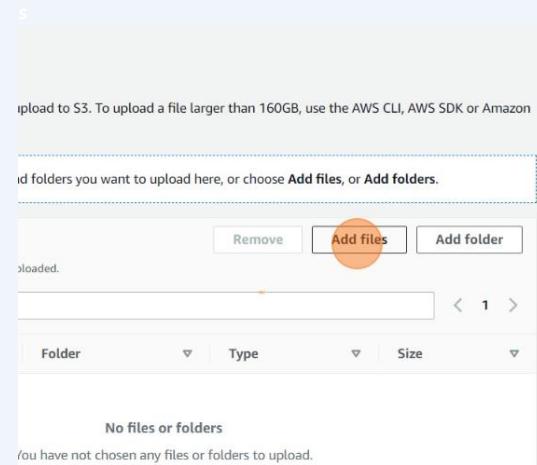
[hiteshpvtbucket](#)

US East (N. Virginia) us-east-1

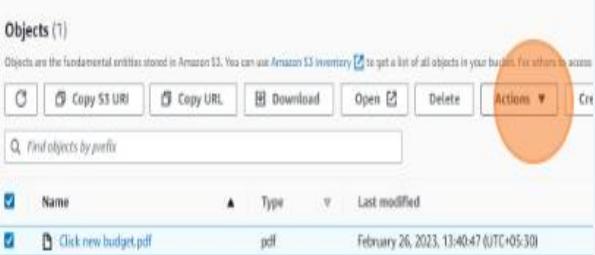
## 7. Click on “Upload”



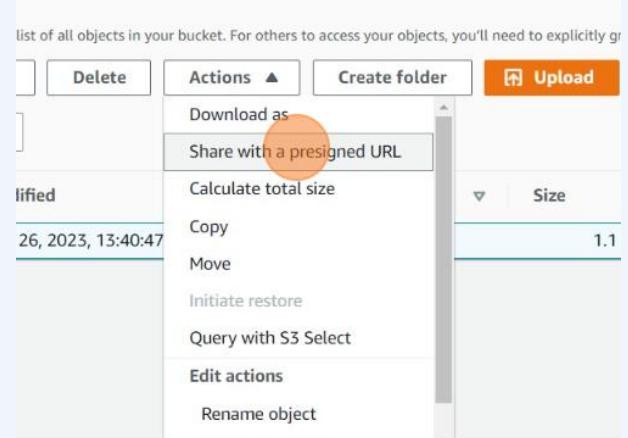
## 8. Add files and folders which you want to upload and click upload



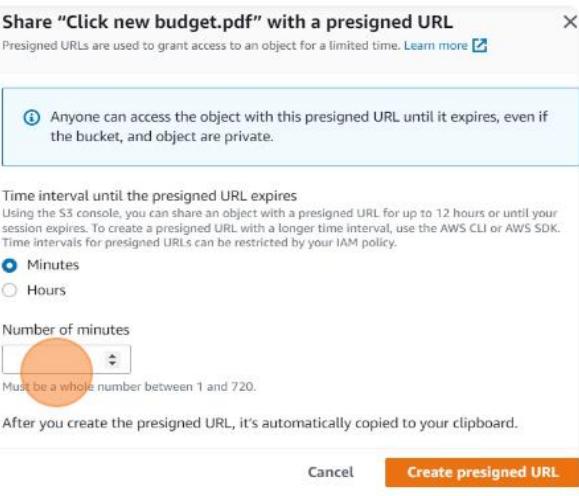
- 9. Select the file for which you want to generate presigned url and click "Action"**



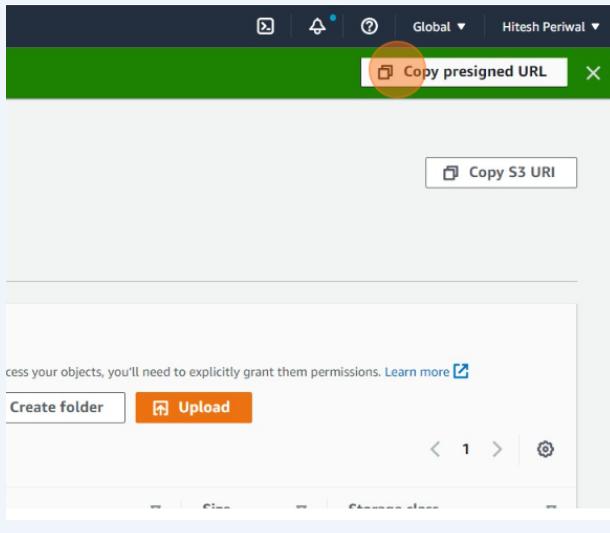
- 10. Click on “Share with a presigned URL”**



- 11. Set the time interval until which the presigned URL expires and click on “Create presigned URL”**



- 12. Click on “Copy presigned URL” and you will be able to access the file using the URL for the set time period only.**

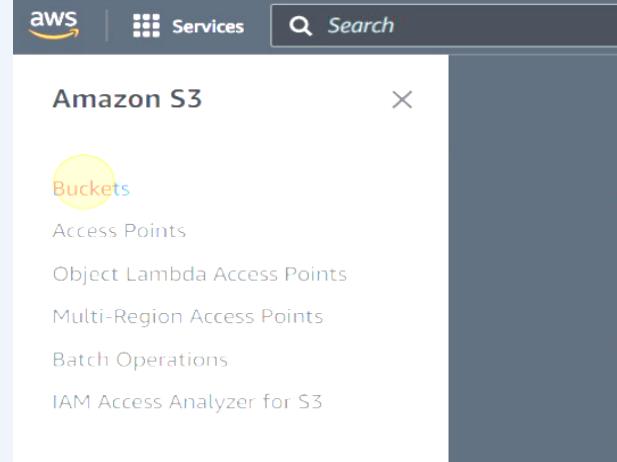


# CREATE A PUBLIC BUCKET IN AWS. UPLOAD A FILE AND GIVE THE NECESSARY PERMISSION TO CHECK WHETHER THE FILE URL IS WORKING OR NOT.

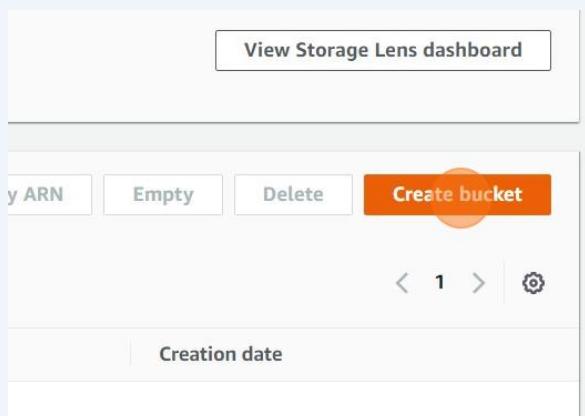
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Click on the search field and type "s3" in the search field and select S3



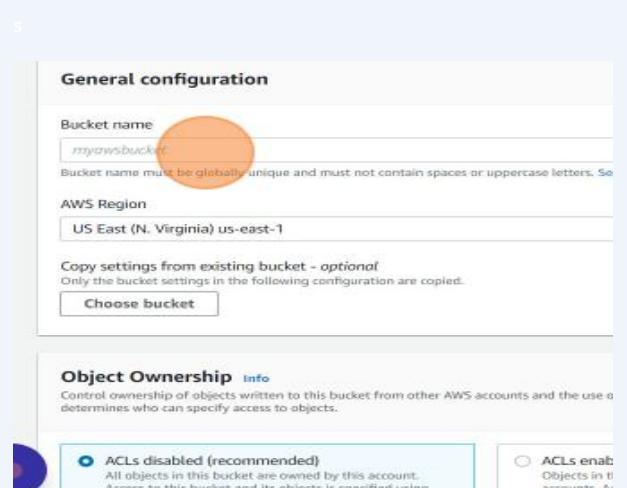
2. Click on "Buckets"



3. Click on "Create Bucket"



4. Click the "Bucket name" field and type "hiteshpublicbucket".



## 5. Select ACLs enabled and uncheck “Block all public access”

The screenshot shows the 'Block Public Access' section of the AWS S3 Bucket Properties page. It includes two radio button options: 'ACLs disabled (recommended)' and 'ACLs enabled'. The 'ACLs enabled' option is highlighted with an orange circle. Below this, there's a note about upcoming permission changes to disable ACLs. At the bottom, there's a checkbox for 'Block all public access', which is checked.

## 6. Click on this check box

The screenshot shows the 'Block Public and Cross-Account Access' section. It contains a checkbox labeled 'Block public and cross-account access to buckets and objects through any public bucket policies'. Below the checkbox is a note about turning off block all public access. There is also a warning message: 'Turning off block all public access might result in this bucket and the objects within it becoming public.' A checkbox for acknowledging this risk is also present.

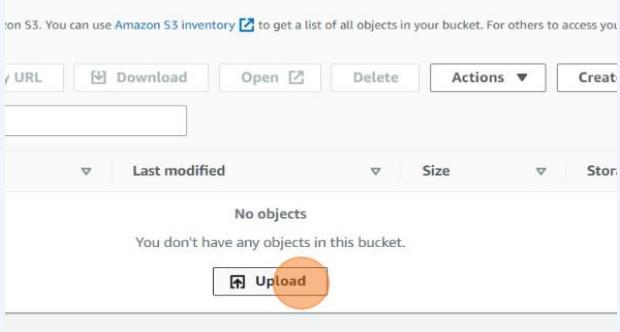
## 7. Click on “Create bucket”

The screenshot shows the 'Step 3: Set Bucket Name and Region' of the AWS Create Bucket wizard. It includes fields for 'Bucket name' (containing 'hiteshppublicbucket'), 'Region' (set to 'US East (N. Virginia) us-east-1'), and 'Storage class' (set to 'Standard'). A note at the top states: 'the bucket key reduces encryption costs by lowering calls to AWS KMS.' At the bottom are 'Cancel' and 'Create bucket' buttons, with 'Create bucket' highlighted with an orange circle.

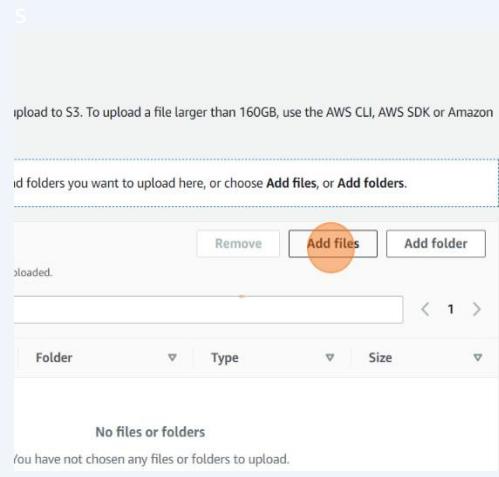
## 8. Click on “hiteshppublicbucket”

The screenshot shows the 'Buckets (1)' page in the AWS Management Console. It lists a single bucket named 'hiteshppublicbucket'. The bucket details show it was created in 'US East (N. Virginia) us-east-1' and has an 'Info' link. A note at the top left says: 'Storage lens provides visibility into storage usage and activity trends. Learn more [link]'. The bucket name 'hiteshppublicbucket' is highlighted with an orange circle.

**9. Click on “Upload”**



**10. Add files and folders which you want to upload and click upload**



**11. You may click the filename and then go to permission and select edit and allow everyone to access the file. Acknowlegde and then save changes.**

Amazon S3 > Buckets > hiteshpublicbucket > Freedom of mind.docx > Edit access

**Edit access control list**

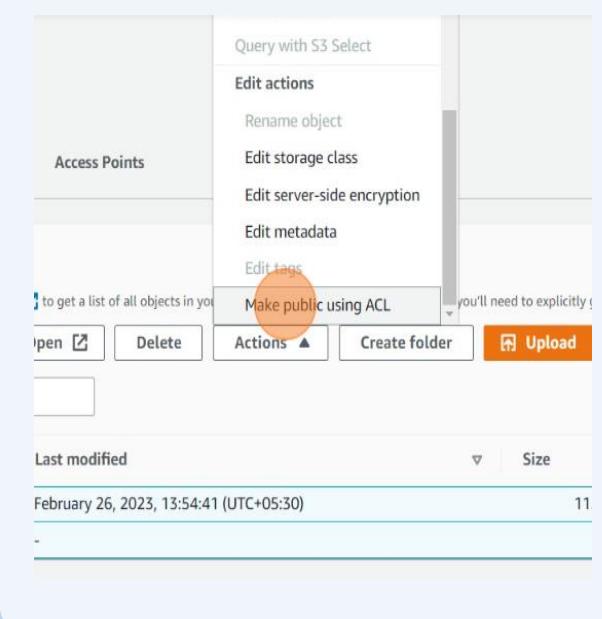
**Access control list (ACL)**

Grant basic read/write permissions to AWS accounts. Learn more

Grantee	Objects	Object ACL
Object owner (your AWS account)	<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Canonical ID: 3e0632790154348e5df9acd87d16317a16fd0f771f049bc8a045e6d0845247ca		
Everyone (public access)	<input checked="" type="checkbox"/> <span style="color: red;">Read</span>	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Authenticated users group (anyone with an AWS account)	<input type="checkbox"/> Read	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Group: http://acs.amazonaws.com/groups/global/AllUsers		
Group: http://acs.amazonaws.com/groups/global/Authen		

**12. Select the file click “Action”. Click on “Make public using ACL” and then click on make public.**

or



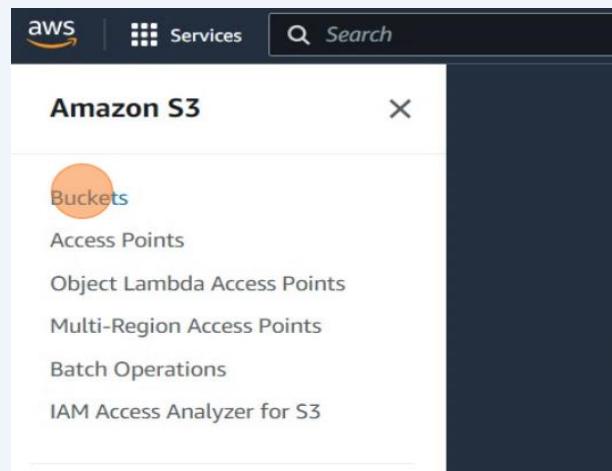
**13. Now the file is accessible through both the object URL and the presigned URL.**

# UPLOAD A STATIC WEBSITE ON S3

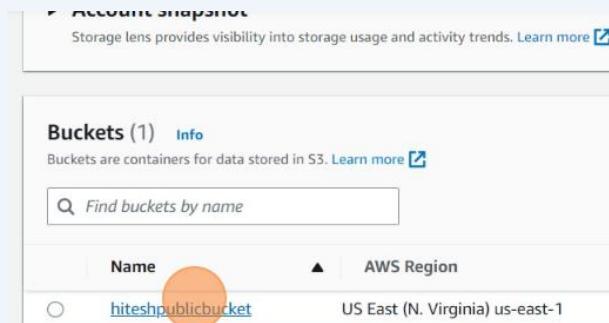
1. Keep the html files ready. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Click on the search field and type "S3" in the search field and select "S3"



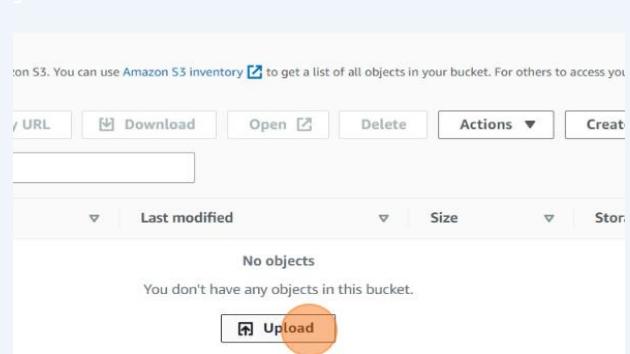
2. Click on "Buckets"



3. Create a public bucket and click on it



4. Click on upload button



**5. Select the html files you want to upload by clicking on “Add files”**

Files and folders (0)

All files and folders in this table will be uploaded.

Find by name		< 1 >	
Name	Folder	Type	Size
No files or folders			
You have not chosen any files or folders to upload.			

**6. Click on “Permissions”**

Find by name

Name	Folder	Type
index.html	-	text/html
next.html	-	text/html

Destination

Destination  
s3://hiteshpublicbucket

▶ Destination details  
Bucket settings that impact new objects stored in the specified destination.

▶ Permissions  
Grant public access and access to other AWS accounts.

**7. Select the option “Choose from predefined ACLs”, “Grant public-read access”, check the checkbox “I understand the risk of ....” and then upload the file**

AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#)

Access control list (ACL)

- Choose from predefined ACLs
- Specify individual ACL permissions

Predefined ACLs

- Private (recommended)  
Only the object owner will have read and write access.
- Grant public-read access  
Anyone in the world will be able to access the specified objects. The object owner will have read and write access.

**Granting public-read access is not recommended**  
Anyone in the world will be able to access the specified objects. [Learn more](#)

I understand the risk of granting public-read access to the specified objects.

**8. Go to public bucket in which the website was uploaded. Select the properties tab**

Amazon S3 > Buckets > hiteshpublicbucket

hiteshpublicbucket [Info](#)

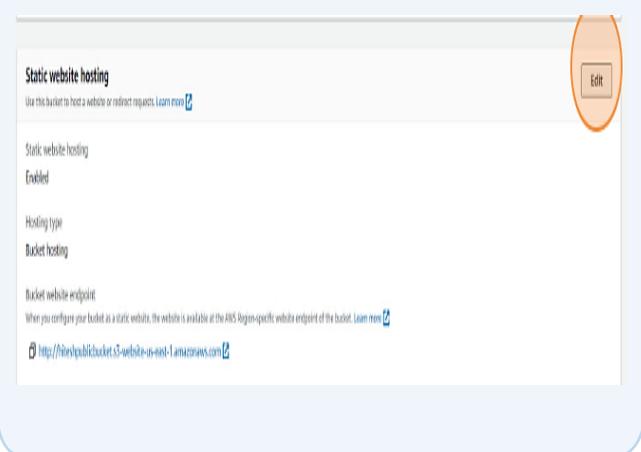
Objects Properties Permissions Metrics Management Access

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 Inventory [Get a report](#) to get a detailed report of your objects.

C	Copy S3 URI	Copy URL	Download	Open
Find objects by prefix				
Name	Type	Last modified		
index.html	html	March 21, 2023, 18:54:22		
next.html	html	March 21, 2023, 18:54:22		

- 9.** Go to static website section and click on “Edit”

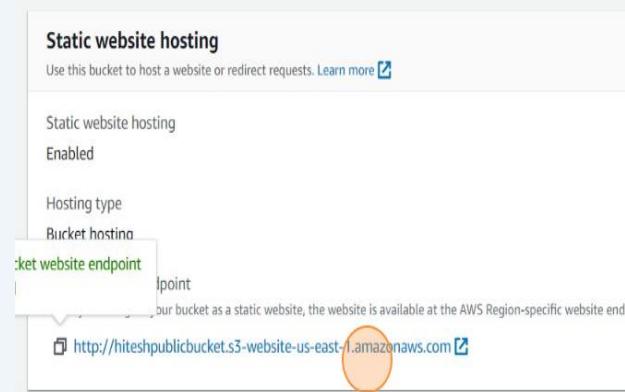


- 10.** Choose the option “Enable” for static website hosting,then choose “Host a static website” and specify the index document of the website.Finally click on save changes

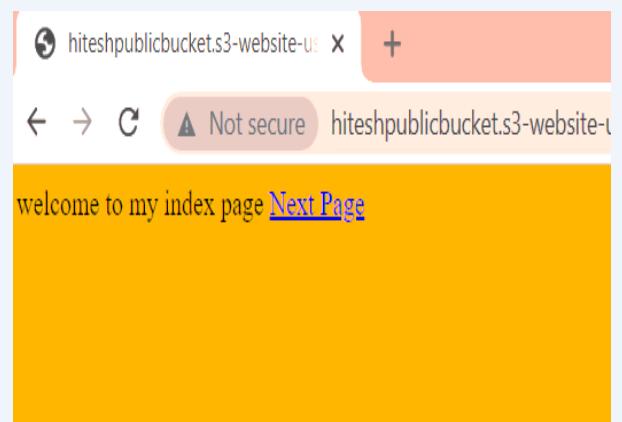
Edit static website hosting [Info](#)

The screenshot shows the 'Edit static website hosting' configuration. It has sections for 'Static website hosting' (Enable selected), 'Hosting type' (Host a static website selected), and 'Index document' (index.html). A note about using S3 Block Public Access is visible. The 'Save changes' button is at the bottom.

- 11.** After saving changes go down and find the bucket website endpoint link.See whether the website is accessible or not across different devices

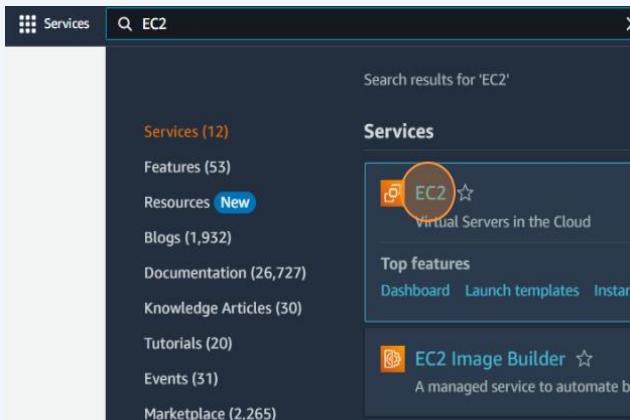


- 12.** The website is accessible



# UPLOAD A STATIC WEBSITE ON EC2 SERVER

1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Click on the search field and type “EC2” in the search field and select EC2



2. Select “Launch instance” option

A screenshot of the 'Launch instance' wizard. At the top, there's a note about easily sizing, configuring, and deploying Microsoft SQL Server Always On Wizard for SQL Server with a 'Learn more' link. Below this is a section titled 'Launch instance' with the sub-instruction 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' There are three buttons: 'Launch instance' (highlighted with a red circle), 'Migrate a server' (with a checkmark icon), and 'Launch instance from template'. At the bottom right, it says 'N. Virginia Region'.

3. Select Ubuntu

A screenshot of the 'Quick Start' section. It shows four options: Amazon Linux, macOS, Ubuntu, and Windows. The 'Ubuntu' option is highlighted with a blue box and has the word 'ubuntu' written below it. Below this section, it says 'Amazon Machine Image (AMI)'. Underneath, it provides details for the selected Ubuntu Server 22.04 LTS (HVM), SSD Volume Type: ami-0557a15b87f6559cf (64-bit (x86)) / ami-0f9bd9098aca2d42, Virtualization: hvm, ENA enabled: true, Root device type: ebs.

4. Select instance type as “t2.micro” and and then “Click on create new key pair”

A screenshot of the instance configuration steps. Step 1: 'Instance type' dropdown showing 't2.micro' selected. It includes details: Family: t2, 1 vCPU, 1 GiB Memory, Free tier eligible. Step 2: 'Key pair (login)' dropdown showing 'Select' and a 'Create new key pair' button.

- 5.** Type the key pair name,Select “RSA” as key type and “.pem” as key file format.Then click on “Create key pair” and download the .pem file

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Key pair name  
Enter key pair name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type  
 RSA RSA encrypted private and public key pair  
 ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format  
 .pem For use with OpenSSH  
 .ppk For use with PuTTY

Cancel Create key pair

- 6.** Allow all the following three traffics and click on “Launch instance”

Firewall (security groups) Info

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-6' with the following rules:

Allow SSH traffic from Anywhere  
Helps you connect to your instance  
Anywhere  
0.0.0.0/0

Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

- 7.** On the instances page click on the instance id and then copy the public IPv4 address

Instances (2) Info

Find instance by attribute or tag (case-sensitive)

Name	Instance ID
—	i-08ccae6a2fd78e2d0
—	i-0fa5c8ea04e93088c

Instance summary for i-0fa5c8ea04e93088c Info

Updated less than a minute ago

Instance ID i-0fa5c8ea04e93088c	Public IPv4 address 34.201.25.194   open address
IPv6 address —	Instance state Running

- 8.** Open Bitwise SSH client,paste the copied public IPv4 address in the host box.Select username as Ubuntu,initial method as none and elevation as default.Then click on client key manager

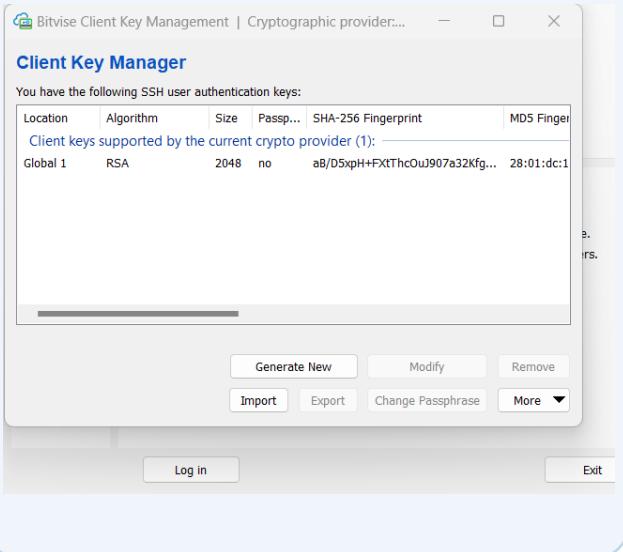
Bitwise SSH Client 9.27

Default profile

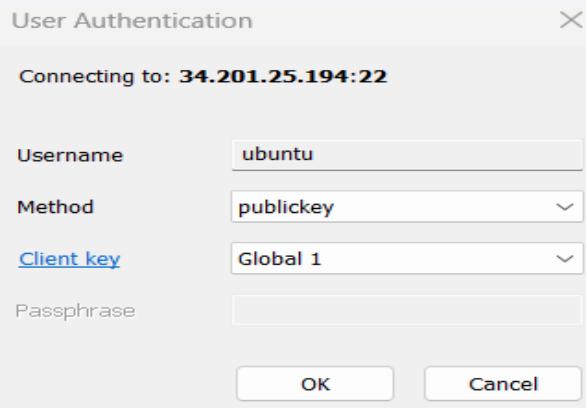
Load profile	Host: 34.201.25.194	Authentication: Username: ubuntu
Save profile as	Port: 22	Initial method: none
New profile	Enable obfuscation	Elevation: Default
Reset profile	Obfuscation keyword:	Kerberos
	SPN:	GSS/Kerberos key exchange
		Request delegation
		<input checked="" type="checkbox"/> gssapi-keyex authentication

Proxy settings Host key manager Client key manager Help

- 9.** Import the download .pem file and then click on login on the SSH client.



- 10.** Select accept and continue and then type the username as Ubuntu, method as public key and client key as Global 1. Then click on "OK".



- 11.** Open the terminal console in the SSH client and type and run the following commands individually. If asked for confirmation click on yes and ok depending on the options available.

```
ubuntu@ip-172-31-6-2:~$ sudo apt-get update
```



```
ubuntu@ip-172-31-6-2:~$ sudo apt-get upgrade
```



```
Do you want to continue? [Y/n] y
```



<Ok>

<Cancel>

```
ubuntu@ip-172-31-6-2:~$ sudo apt-get install nginx
```



```
Do you want to continue? [Y/n] y
```



Daemons using outdated libraries

Which services should be restarted?

- dbus.service
- networkd-dispatcher.service
- systemd-logind.service
- unattended-upgrades.service
- user@1000.service

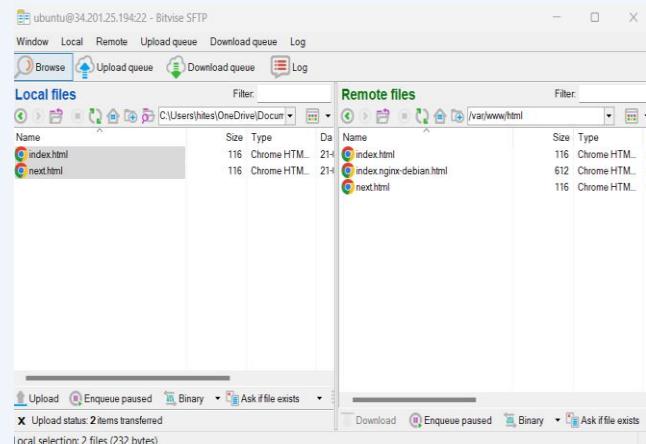
<Ok>

<Cancel>

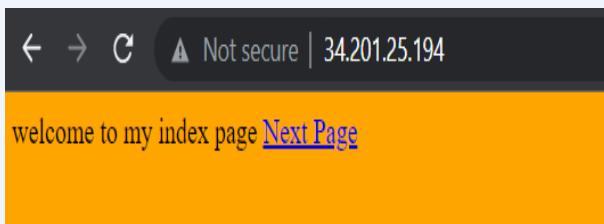
**12.** Type the following commands in the terminal to change the working directory to the root and then move to var/www/ to give the permission add files to that folder.

```
ubuntu@ip-172-31-6-2:~$ pwd  
/home/ubuntu  
ubuntu@ip-172-31-6-2:~$ cd /  
ubuntu@ip-172-31-6-2:/$ cd var/www/  
ubuntu@ip-172-31-6-2:/var/www$ sudo chmod 777 html
```

**13.** Open new SFTP window and move to the var/www/html directory in the server machine. Now drag and drop the required html files from the local machine to the server machine.



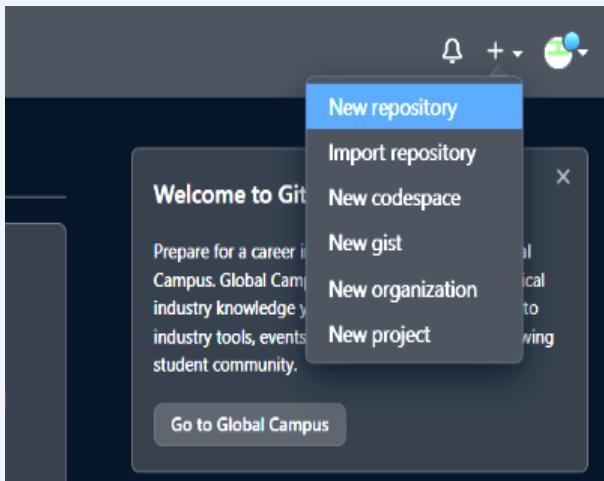
**14.** Open the site using the public IPv4 address and it will be accessible



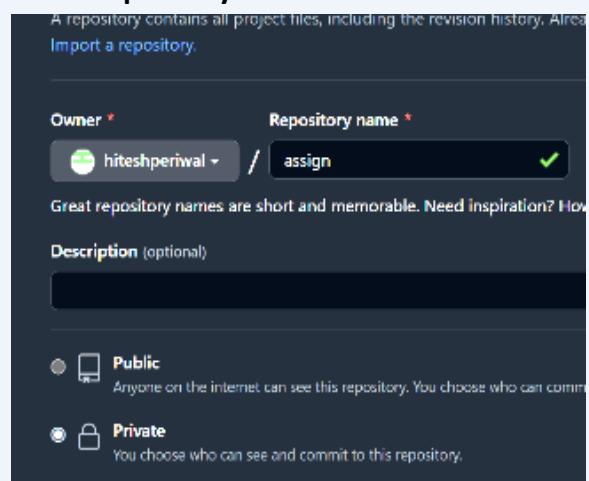
# Assignment No. 8

## DEPLOY A PROJECT FROM LOCAL MACHINE TO GITHUB AND VICE-VERSA

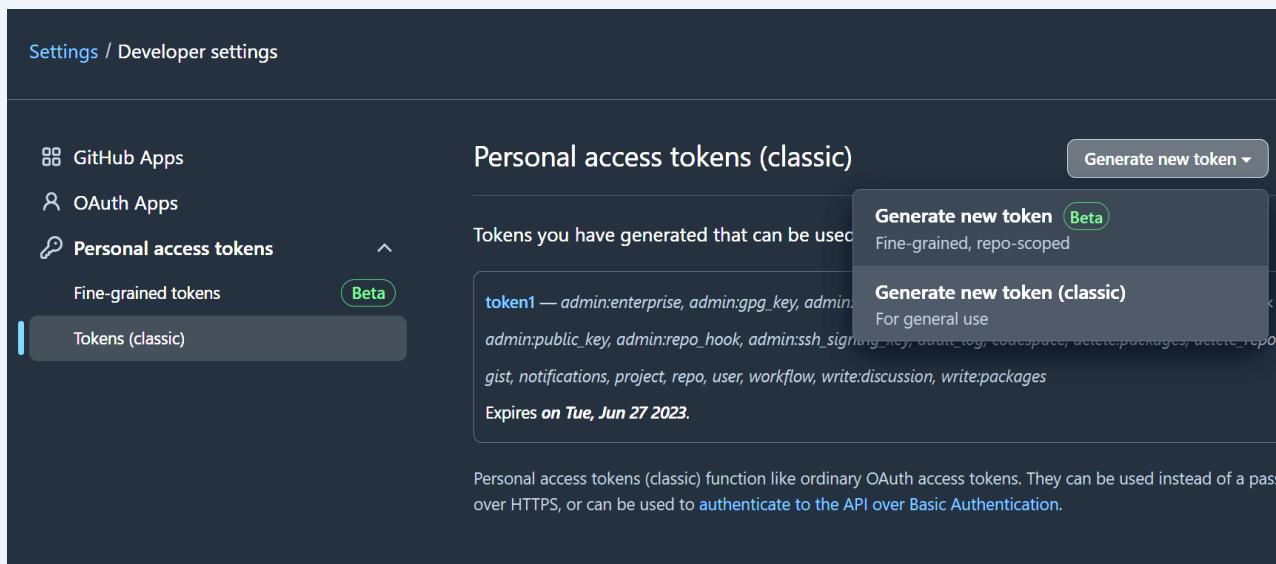
1. Visit [github.com](https://github.com), login in it and click on "New repository".



2. Give a Repository name and set it as private. Then click on "Create repository".



3. Go to Settings, on the bottom of the left panel click on "developer settings". Then go to "Personal access tokens", choose "Tokens(classic)". Now on the right side select "Generate new token" and then "Generate new token(classic)".



- 4.** Now give this token a name,select expiration duration as 90 days.Click on all the checkbox listed under scopes to get the full control.Then click on “Generate token”.

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note  
token2

What's this token for?

Expiration \*  
90 days The token will expire on Mon, Jul 3 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status

- 5.** Copy and save the generated token for future use.

Personal access tokens (classic) [Generate new token](#) [Revoke all](#)

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

Copied!

✓ ghp_ehIRDwytfb0s3qVrJ75NiKFsjhquXw0zCx0R ✓	<a href="#">Delete</a>
token1 — admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, Last used within the last week	<a href="#">Delete</a>

- 6.** Go to settings option inside “assign” repository.Click on “Collaborators” and “add people” with whom you want to collab this project.Invitation can be sent to the respective person using the github username or email id with which his/her github account is associated.The invitation can be accepted using the link that the person will receive on the registered email or by navigating to the notification section of his/her github account.Upon acceptance of the invitation , the private repository will be accessible to that collaborator.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

**Collaborators**

Code and automation

Actions Webhooks Environments Codespaces Pages

Security

Code security and analysis Deploy keys Secrets and variables

Who has access

PRIVATE REPOSITORY Only those with access to this repository can view it. Manage

DIRECT ACCESS 0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

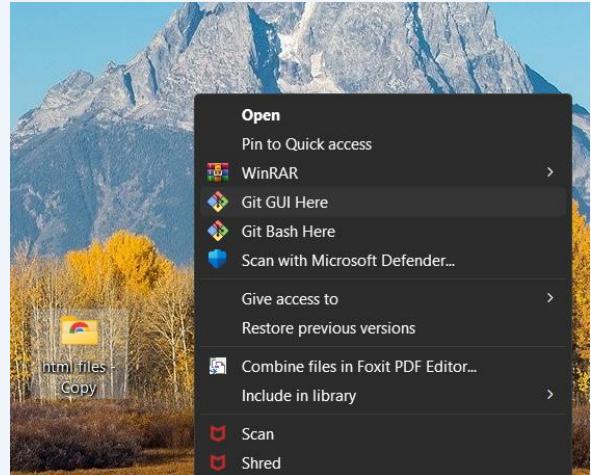
You haven't invited any collaborators yet

Add people

- 7.** Copy the https link of the repository which will be required in the upcoming steps.



- 8.** Right click on the folder whose file you wish to deploy on github and click on "Git Bash here".



- 9.** Type the following commands in the git bash to deploy a project from local machine to Github

```

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy
$ git init
Initialized empty Git repository in C:/Users/hites/OneDrive/Documents/onedrive/Desktop/html files

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ ls -A
.git/ index.html next.html

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html
    next.html

nothing added to commit but untracked files present (use "git add" to track)

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git add .

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html
    new file:   next.html
hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git config --global user.email hp.socialconnect@gmail.com

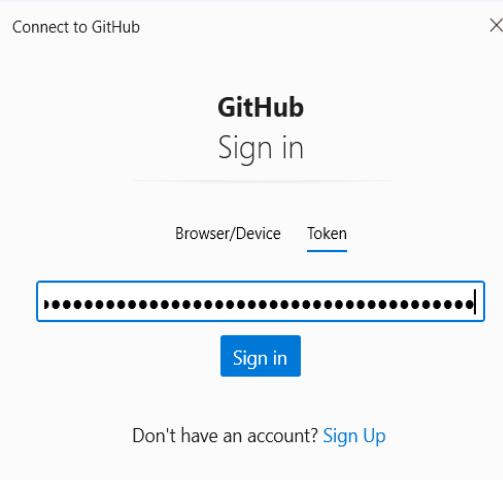
hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git config --global user.name hiteshperiwal

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git commit -m "done"
hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git remote add origin https://github.com/hiteshperiwal/assign.git

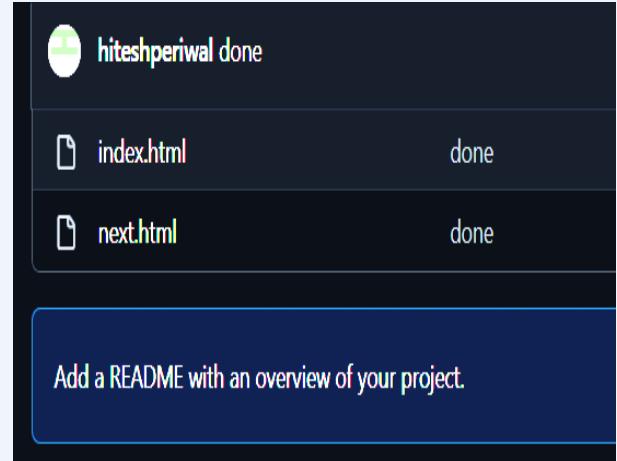
hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/html files - Copy (master)
$ git push -u origin master

```

- 10.** In the window which popped up, select sign in using token, paste the token saved previously and click “Sign in”.



- 11.** The project is thus deployed from local machine to github”.



- 12.** To deploy a project from Github to local machine :-

- Copy the link of the repository which you want to deploy.
- Create a new folder named “N1” and open gitbash there.
- Type the following commands in the git bash (use the link of the repository in the command).
- Thus, the task is complete

```

hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/N1
$ git init
Initialized empty Git repository in C:/Users/hites/OneDrive/Documents/onedrive/Desktop/N1/.git/
hites@LAPTOP-Q12SOP5J MINGW64 ~/OneDrive/Documents/onedrive/Desktop/N1 (master)
$ git clone https://github.com/sudip7407/New-Repo1.git
Cloning into 'New-Repo1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 15 (delta 6), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (6/6), done.

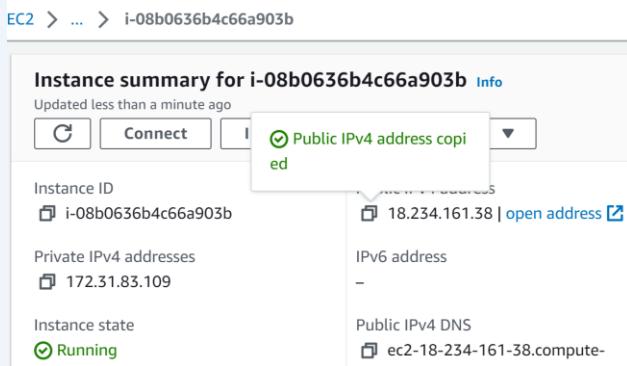
```

Name	Date modified	Type	Size
.git	04-04-2023 20:48	File folder	
.gitignore	04-04-2023 20:48	txtfile	1 KB
index	04-04-2023 20:48	JavaScript File	1 KB
New Text Document	04-04-2023 20:48	Text Document	0 KB
package.json	04-04-2023 20:48	JSON File	1 KB

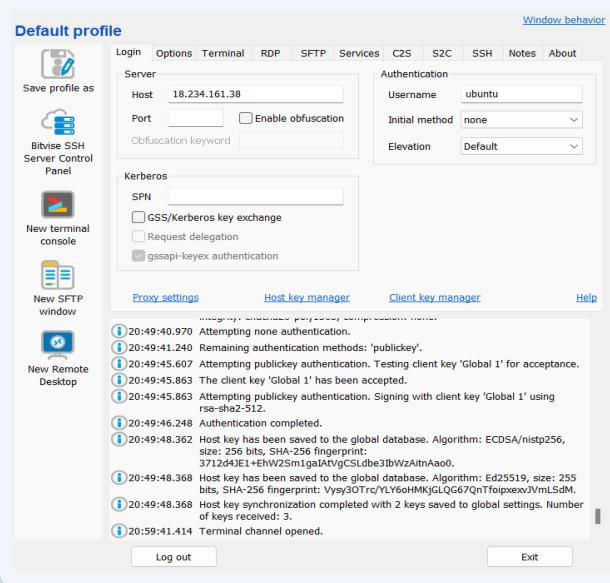
# Assignment No. 9

## DEPLOY A PROJECT FROM GITHUB TO EC2

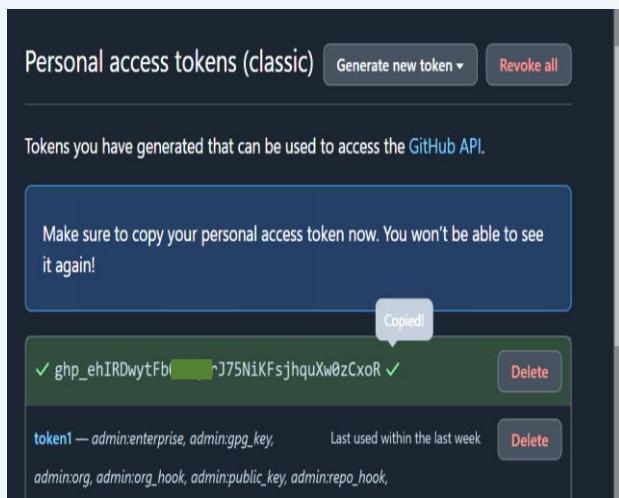
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Create an EC2 server and copy its IPv4 address.



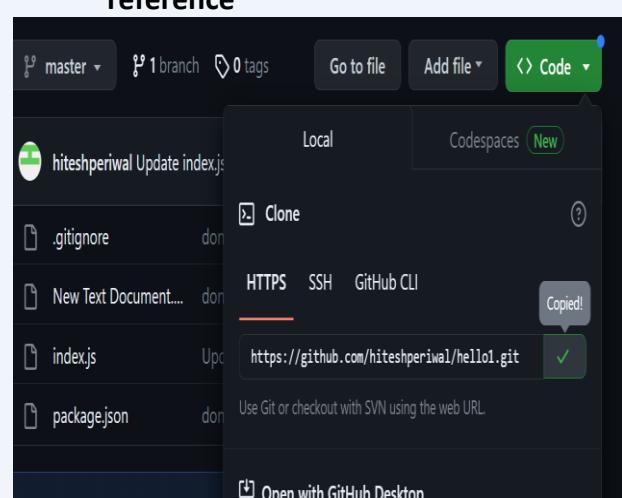
2. Login to Bitwise SSH client using the copied IPv4 address as host address.



3. Create a personal access token in the Github and save it for future reference



4. Create a repository containing the project files in the Github and copy and save its HTTPS address for future reference



5. Open a “New terminal console” in the Bitwise SSH client and type the following commands in it to install nodejs:-

**pwd** (to check if it is in server)  
**sudo apt-get update**  
**sudo apt-get upgrade**  
**sudo apt install nginx**  
**nginx -v** (to check nginx version)  
**curl -s1 [https://deb.nodesource.com/setup\\_18.x](https://deb.nodesource.com/setup_18.x) |sudo -E bash -**  
**sudo apt install nodejs**  
**node -v** (to check nodejs version)

```
ubuntu@ip-172-31-83-109:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-83-109:~$ sudo apt-get update
ubuntu@ip-172-31-83-109:~$ sudo apt-get upgrade
ubuntu@ip-172-31-83-109:~$ sudo apt install nginx
ubuntu@ip-172-31-83-109:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-83-109:~$ curl -s1 https://deb.nodesource.com/setup_18.x |sudo -E bash -
ubuntu@ip-172-31-83-109:~$ sudo apt install nodejs
ubuntu@ip-172-31-83-109:~$ node -v
v18.15.0
```

6. To upload the project on EC2 server type the following commands:-

**git clone** *https address of the github hub repository where project is uploaded*  
(For username type your github username and for password enter the personal access token)  
**dir** (to check the directory of the project)  
**cd hello1/** (to enter into repository)  
**npm install** (node package manager installation to run node commands)  
**node index.js** (as project is uploaded start server)

```
ubuntu@ip-172-31-82-194:~$ git clone https://github.com/hiteshperiwal/hello1.git
Cloning into 'hello1'...
Username for 'https://github.com': hiteshperiwal
Password for 'https://hiteshperiwal@github.com':

ubuntu@ip-172-31-83-109:~$ dir
hello1
ubuntu@ip-172-31-83-109:~$ cd hello1/
ubuntu@ip-172-31-83-109:~/hello1$ npm install

ubuntu@ip-172-31-83-109:~/hello1$ node index.js
Started server
```

- 7.** On pasting the public Pv4 address address of EC2 server ,the project wont load because only through specific ports it is accessible.Go to git hub and check for the ports permitted for the project by opening the index.js file.

Now to address this issue go to instance(in aws) -> security -> security groups -> edit inbound rules -> Add rules -> set type as custom tcp , port range as specified in index.js file, source as anywhere-IPv4 and 0.0.0.0/0 ->save rules.Now type the port number after colon in IPv4 address and the website will be accessible.

**Hence the project is successfully deployed from github to EC2.**

The screenshot shows two side-by-side interfaces. On the left is a GitHub repository page for 'hiteshperiwal/Update index.js'. It displays 11 lines of code for an Express.js application that sends a 'Hello MCKVIANS' response. On the right is the AWS Management Console showing the 'Edit inbound rules' for a security group. It lists four rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0beeb224ef97d6ee1	HTTPS	TCP	443	Custom	0.0.0.0/0 X
sgr-0a176715a9ba724a4	Custom TCP	TCP	4000	Anywhere	0.0.0.0/0 X
sgr-0fb02dd5a726f5f9d	HTTP	TCP	80	Custom	0.0.0.0/0 X
sgr-042f4a7008619dfeb	SSH	TCP	22	Custom	0.0.0.0/0 X

Below the rules is a browser screenshot showing the website at 18.234.161.38:4000 displaying the message 'Hello MCKVIANS'.

- 8.** If there is some editing done in the project it will not be reflected in the website.For that we need to terminate the server using **ctrl+c** and then type the follow command:-

**git pull**

(enter the github username and its token as password)

**node index.js**

Now,refresh the browser to see the changes reflected on it.

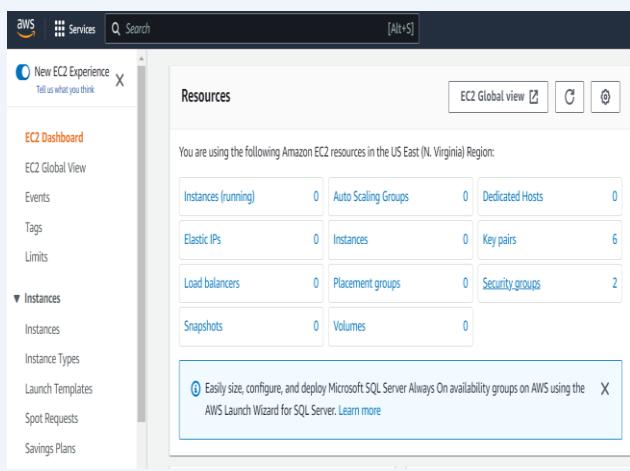
```
ubuntu@ip-172-31-83-109:~/hello1$ git pull
Username for 'https://github.com': hiteshperiwal
Password for 'https://hiteshperiwal@github.com':
ubuntu@ip-172-31-83-109:~/hello1$ node index.js
Started server
```

The screenshot shows a browser window with the URL 18.234.161.38:4000. The page content has been updated to 'Hello MCKVIANS....how are you'.

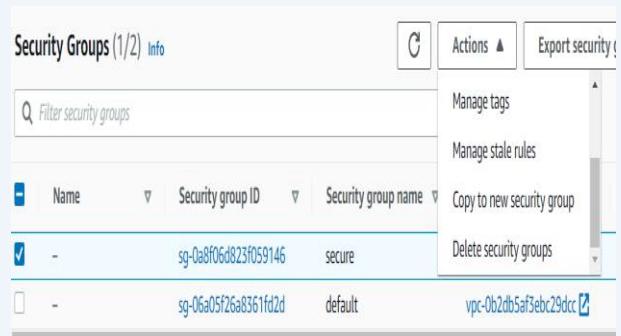
# Assignment No. 9

## DEPLOY A PROJECT FROM GITHUB TO EC2 BY CREATING NEW SECURITY GROUP AND USER DATA

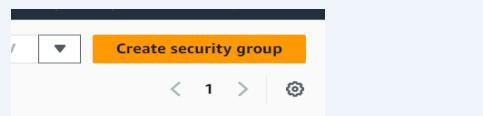
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Go to EC2 Service and click on “Security Groups”



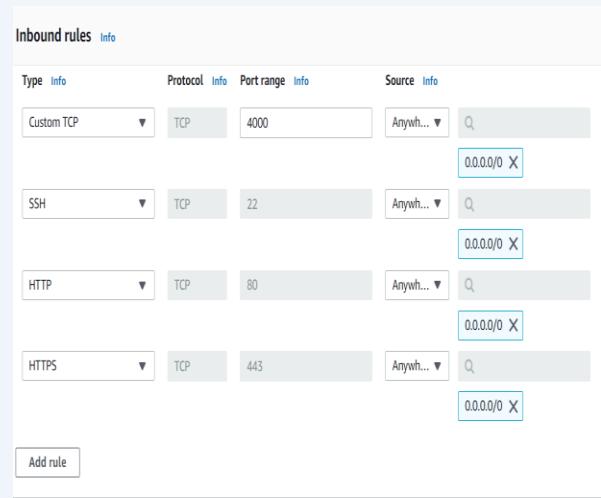
2. Select all security groups except the default and go to Actions dropdown menu and choose “Delete security groups”



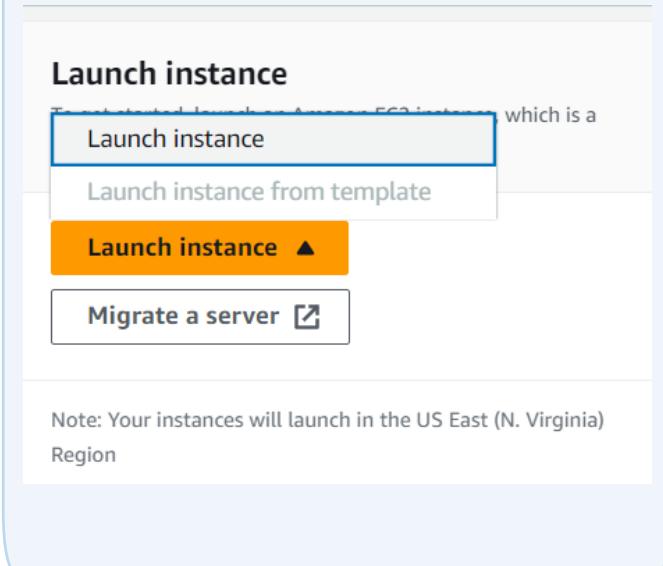
3. Click on “Create security group”. Enter the “Security group name” and “Description”



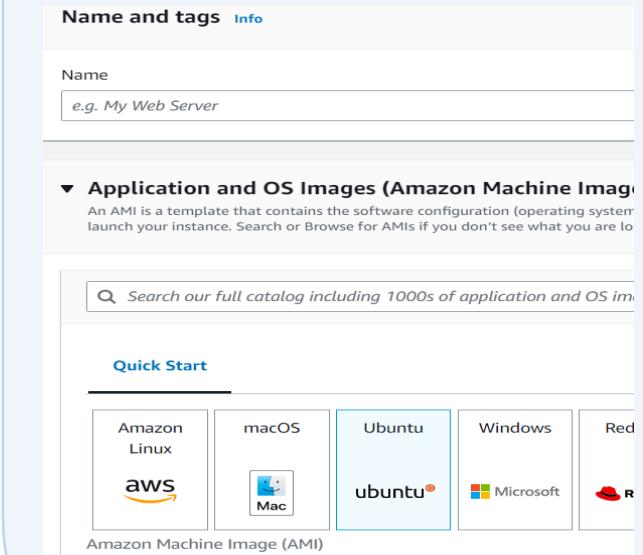
4. Add the Inbound rules with details as shown in the figure. After entering the details click on “Create security group”



5. Go to EC2 dashboard and Click on “Launch Instance”



6. Enter the instance name and select Ubuntu as Amazon Machine Image



7. Select key pair

▼ Key pair (login) [Info](#)

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 - *required*

Select

Create new key pair

8. Under network settings ,click on “Select existing security group” and in the security group dropdown select the security group which you just created

Firewall (security groups) [Info](#)

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

Security groups [Info](#)

Select security groups



- secure  
VPC: vpc-0b2db5af3ebc29dcc
- default  
VPC: vpc-0b2db5af3ebc29dcc

sg-0a8f06d823f059146

sg-06a05f26a8361fd2d

Compare security group rules

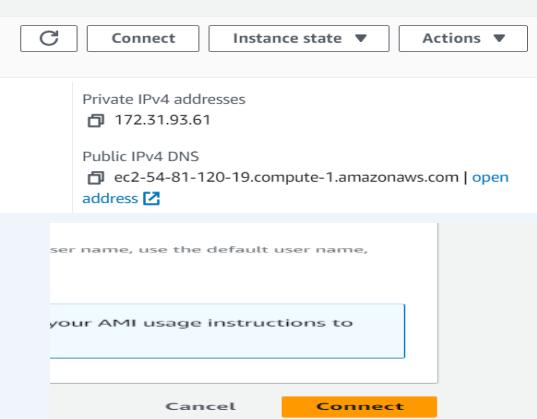
- 9.** Type the following user data under advanced details section.Then click on create instance.

User data - optional [Info](#)

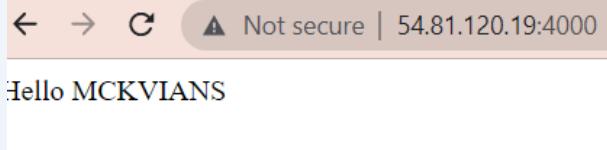
Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/hiteshperiwal/hello1.git
cd ...
npm install
node index.js
```

- 11.** However for certain machines the project will not be accessible directly.To access the project,go to instance and click on connect and again click on connect



- 10.** Now using the public ip address of the instance and the corresponding port number of the project ,we can access the project.



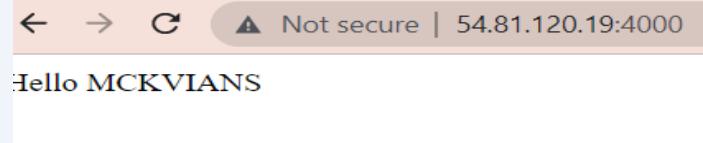
- 12.** Now type the following commands in the terminal.Keep the repository url and the github personal access token ready as they will be required in these commands

```
ubuntu@ip-172-31-93-61:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-93-61:~$ npm -v
9.5.1
ubuntu@ip-172-31-93-61:~$ node -v
v18.16.0
ubuntu@ip-172-31-93-61:~$ git clone https://github.com/hiteshperiwal/hello1.git
Cloning into 'hello1'...
Username for 'https://github.com': hiteshperiwal
Password for 'https://hiteshperiwal@github.com':
```

```
ubuntu@ip-172-31-93-61:~$ ls
hello1
ubuntu@ip-172-31-93-61:~$ cd hello1/
ubuntu@ip-172-31-93-61:~/hello1$ npm install
```

```
ubuntu@ip-172-31-93-61:~/hello1$ node index.js
Started server
```

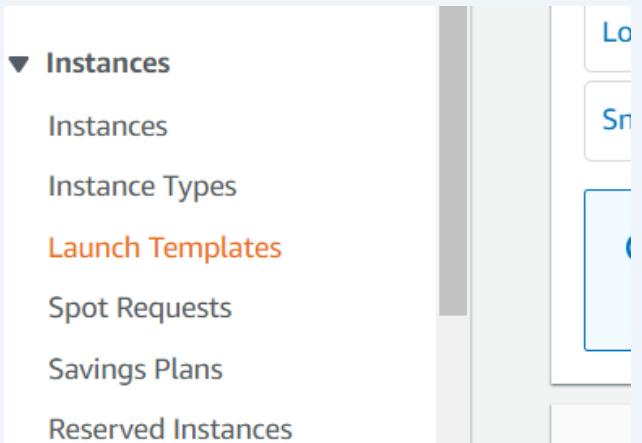
- 13.** Now using the public ip address of the instance and the corresponding port number of the project ,we can access the project.Hence the project is successfully deployed from github to EC2 using new security group and user data



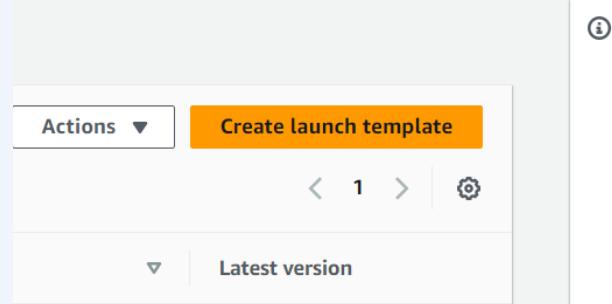
# Assignment no. 11

## Build scaling plans in AWS that balances load on different EC2 instances

1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Go to EC2 dashboard and click “Launch Template”



2. Click on “Create launch instance” option



3. Give template name,template version description and select autoscaling guidance

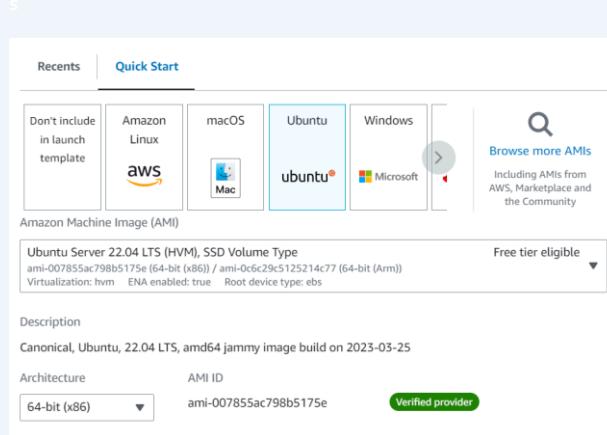
Launch template name and description

Launch template name - required  
MyTemplate  
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '@'.

Template version description  
ver1  
Max 255 chars

Auto Scaling guidance [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling  
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

4. Select Ubuntu as AMI



## 5. Select “t2.micro” as instance type and key pair for login

▼ Instance type [Info](#)

Instance type

t2.micro	Free tier eligible
Family: t2	1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows pricing:	0.0162 USD per Hour
On-Demand SUSE pricing:	0.0116 USD per Hour
On-Demand RHEL pricing:	0.0716 USD per Hour
On-Demand Linux pricing:	0.0116 USD per Hour

All generations [Compare instance types](#)

▼ Key pair (login) [Info](#)

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

 [Create new key pair](#)

## 6. Now select the existing security group which you created for the project

Subnet Info

Don't include in launch template [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

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.

Select existing security group [Create security group](#)

Security groups [Info](#)

Select security groups

secure sg-0a8f06d823f059146 X  
VPC: vpc-0b2db5af3ebc29dcc

[Compare security group rules](#)

► Advanced network configuration

## 7. Type the following user data and click on create launch template

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/hiteshperiyal/hello1.git
cd hello1/
npm install
node index.js
```

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#)

[Create launch template](#)

## 8. In the EC2 dashboard click on “Auto scaling groups and then click on create Auto Scaling groups”

EC2 Dashboard

Target Groups

▼ Auto Scaling

Launch Configurations

[Auto Scaling Groups](#)

### Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

[Create Auto Scaling group](#)

**9. Enter the Auto Scaling group name. Select the template created by you and its version as latest. Click on “Next”**

Name

Auto Scaling group name  
Enter a name to identify the group.  
Auto  
Must be unique to this account in the current Region and no more than 255 characters.

Launch template [Info](#) [Switch to launch configuration](#)

Launch template  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.  
MyTemplate [Create a launch template](#) [C](#)

Version  
Latest (1) [C](#)

Description ver1

Launch template MyTemplate <a href="#">C</a> lt-0c30a5f5aaabe1a845	Instance type t2.micro
--	---------------------------

**10. Select all the Availability zones and subnets and click on “Next”**

VPC  
Choose the VPC that defines the virtual network for your Auto Scaling group.  
vpc-0b2db5af3ebc29dcc  
172.31.0.0/16 Default [C](#)

Create a VPC [C](#)

Availability Zones and subnets  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets [C](#)

- us-east-1a | subnet-040ff3fa6f65a9a1b X  
172.31.16.0/20 Default
- us-east-1b | subnet-025116bf423799fb2 X  
172.31.32.0/20 Default
- us-east-1c | subnet-07e257cda4f004995 X  
172.31.0.0/20 Default
- us-east-1d | subnet-0c2a22a6e4867e89b X  
172.31.80.0/20 Default
- us-east-1e | subnet-0125834c110fd13d3 X  
172.31.48.0/20 Default
- us-east-1f | subnet-06285a888678b6059 X  
172.31.64.0/20 Default

[Create a subnet](#) [C](#)

**11. In the load balancing ,select “Attach to a new load balancer”.In the load balancer type select “Application load balancer”. Give the load balancer name and select the load balancer scheme as internet facing**

Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

Load balancer type  
Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

Application Load Balancer  
HTTP, HTTPS

Network Load Balancer  
TCP, UDP, TLS

Load balancer name  
Name cannot be changed after the load balancer is created.  
AutoScaling-1

Load balancer scheme  
Scheme cannot be changed after the load balancer is created.  
 Internal  Internet-facing

Network mapping  
Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

**12. In the listeners section, enter the port as mentioned in your project and health check grace period as**

Listeners and routing  
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	4000	Create a target group New target group name An instance target group with default settings will be created. AutoScaling-1

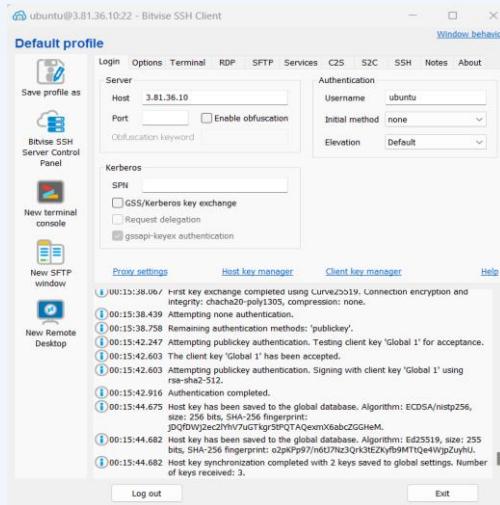
Tags - optional  
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.  
[Add tag](#)  
50 remaining

Health check grace period [Info](#)  
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.  
300 seconds

13. Enter the desired capacity as 2,minimum capacity as 2 and maximum capacity as 3.Select target tracking scaling policy,give it a name,target value 50 and 300 seconds as warm up .Click on “Next”, “Next”, “Next” and then “Create auto scaling group”.

<b>Group size - optional</b>	<a href="#">Info</a>
Specify the size of the Auto Scaling group. You can set minimum, maximum capacity limits. Your desired capacity will be scaled up or down as needed.	
Desired capacity	<input type="text" value="2"/>
Minimum capacity	<input type="text" value="2"/>
Maximum capacity	<input type="text" value="3"/>
<b>Scaling policies - optional</b>	
Choose whether to use a scaling policy to dynamically resize your group based on demand.	
<input checked="" type="radio"/> <b>Target tracking scaling policy</b> Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.	
Scaling policy name	<input type="text" value="Target Tracking Policy"/>
Metric type	<input type="text" value="Average CPU utilization"/>
Target value	<input type="text" value="50"/>
Instances need	<input type="text" value="300"/> seconds warm up before including in metric

- 15.** Login in Bitwise SSH client using one of the public ip address of the instances created and the key pair used in those instance.



- 14.** Now two instances are running. Overload one of them to see whether auto scaling functionality is working or not.

Instances (2) <a href="#">Info</a>				<a href="#">Create instance</a>
	Name	Instance ID	Instance state	Instance type
<a href="#">-</a>	-	i-0e62dbb433ee46ae3	<span>Running</span>	t2.micro
<a href="#">-</a>	-	i-01fdaed5d372515b5	<span>Running</span>	t2.micro

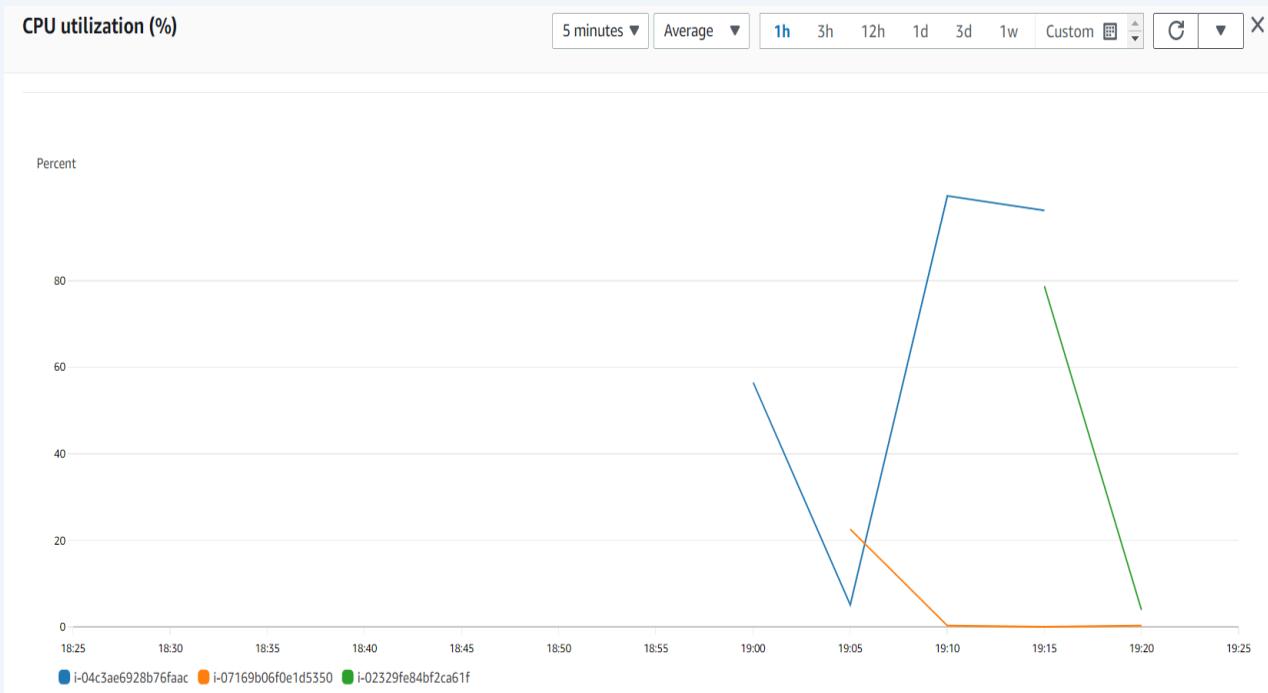
- 16.** Open the terminal console,create a shell executables file,type the following commands and execute the file to overload the server

```
ubuntu@ip-172-31-68-249:~$ vi inf1.sh
```

```
#!/bin/bash
while true
do
    echo "Looping forever"
    # Add other commands to run in the loop here
done
~
```

```
ubuntu@ip-172-31-68-249:~$ chmod +x inf1.sh  
ubuntu@ip-172-31-68-249:~$ ./inf1.sh
```

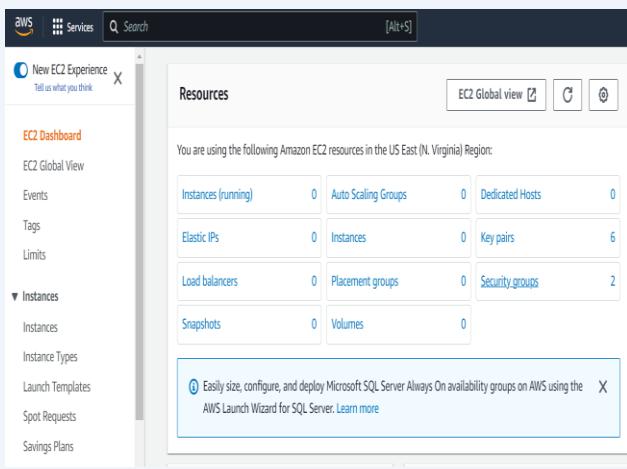
- 17.** Select the running instances and click on monitoring. You will observe creation of new instance during the 300 seconds warm up when the server is overloaded. Hence load is balanced on different EC2 instances



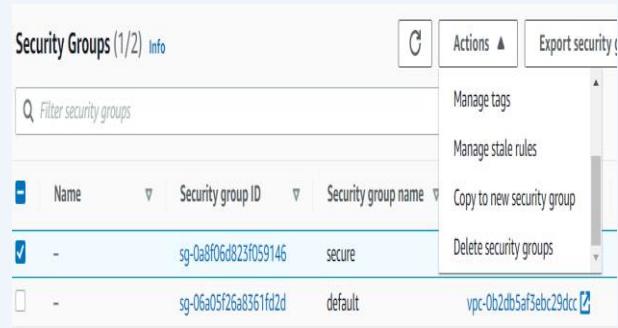
# Assignment No. 12

## Deploy and run project in aws without using port

1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Go to EC2 Service and click on “Security Groups”



2. Select all security groups except the default and go to Actions dropdown menu and choose “Delete security groups”



3. Click on “Create security group”. Enter the “Security group name” and “Description”

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To

**Basic details**

Security group name [Info](#)  
MyWebServerGroup  
Name cannot be edited after creation.

Description [Info](#)  
Allows SSH access to developers

VPC [Info](#)  
vpc-0b2db5af3ebc29dcc

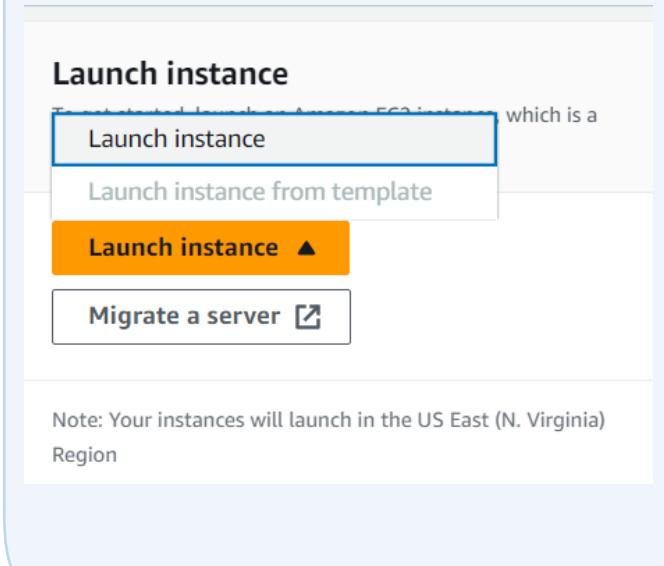
4. Add the Inbound rules with details as shown in the figure. After entering the details click on “Create security group”

Inbound rules [Info](#)

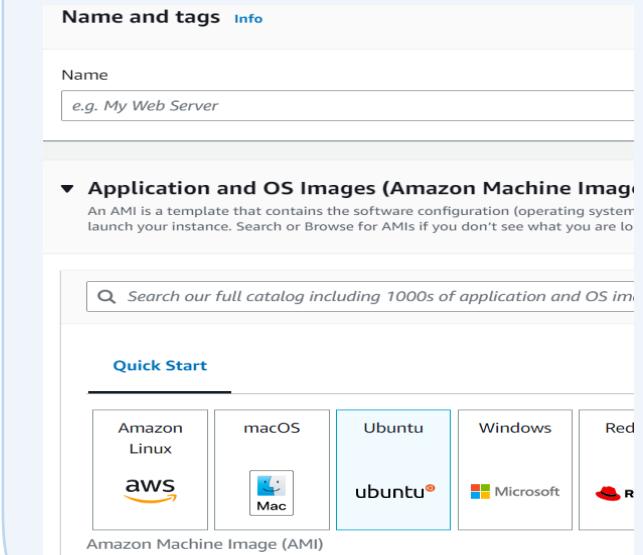
Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>
Custom TCP	TCP	4000	Anywhere 0.0.0.0/0
SSH	TCP	22	Anywhere 0.0.0.0/0
HTTP	TCP	80	Anywhere 0.0.0.0/0
HTTPS	TCP	443	Anywhere 0.0.0.0/0

Add rule

5. Go to EC2 dashboard and Click on “Launch Instance”



6. Enter the instance name and select Ubuntu as Amazon Machine Image



7. Select key pair

▼ Key pair (login) [Info](#)

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 - *required*

Select

Create new key pair

8. Under network settings ,click on “Select existing security group” and in the security group dropdown select the security group which you just created

Firewall (security groups) [Info](#)

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

Security groups [Info](#)

Select security groups



- secure  
VPC: vpc-0b2db5af3ebc29dcc
- default  
VPC: vpc-0b2db5af3ebc29dcc

sg-0a8f06d823f059146

sg-06a05f26a8361fd2d

Compare security group rules

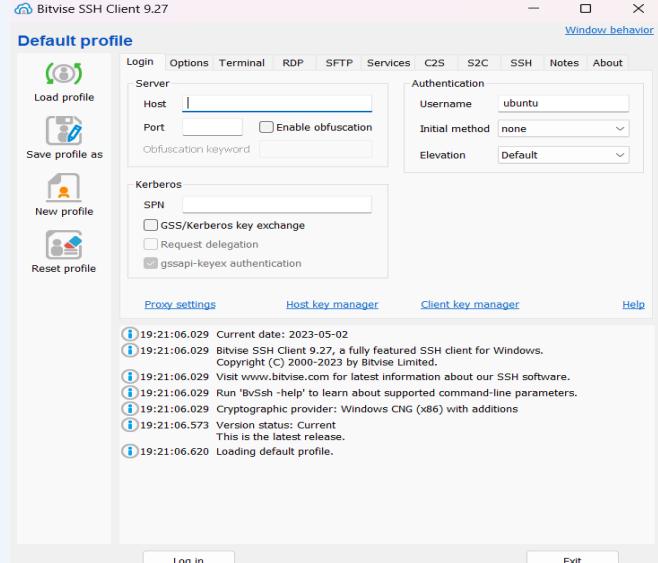
- 9.** Type the following user data under advanced details section.Then click on create instance.

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/hiteshperiwal/hello1.git
cd ...
npm install
node index.js
```

- 10.** Login to Bitwise SSH client using the public ip address.



- 11.** Now login in Bitwise SSH client using the public ip address.Open the terminal and type the following commands and then open nano editor.In the nano editor after commenting the previous location type the new location as given below and then save and exit.

```
ubuntu@ip-172-31-87-57:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-87-57:~$ cd /
ubuntu@ip-172-31-87-57:$ pwd
/
ubuntu@ip-172-31-87-57:$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-87-57:/etc/nginx/sites-available$ sudo nano default
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

- 12.** After exiting the nano editor,type the following command in the terminal to restart the nginx web server.After restarting the nginx web server you will be able to access the website without the port number . Hence we are running project in aws without using port number.

```
ubuntu@ip-172-31-87-57:/etc/nginx/sites-available$ sudo systemctl restart nginx
ubuntu@ip-172-31-87-57:/etc/nginx/sites-available$
```

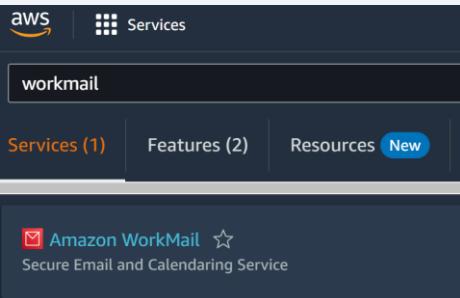
← → ⌂ Not secure | 54.81.120.19

Hello MCKVIANS

# Assignment No. 13

## Create a workmail for your organization

1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Click on the search field and type “workmail” in the search field and select Amazon WorkMail. Click on “Create organization”

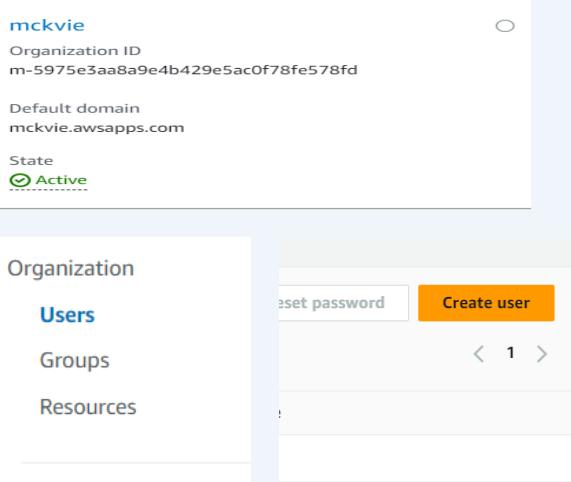


### Get started

Create an Amazon WorkMail organization to get started.

[Create organization](#)

3. Click on the organization which was just created. On the left panel select users and then click on “Create user”



2. Select “Free test domain” in the organization settings and click on “Create organization”

### Create an Amazon WorkMail organization

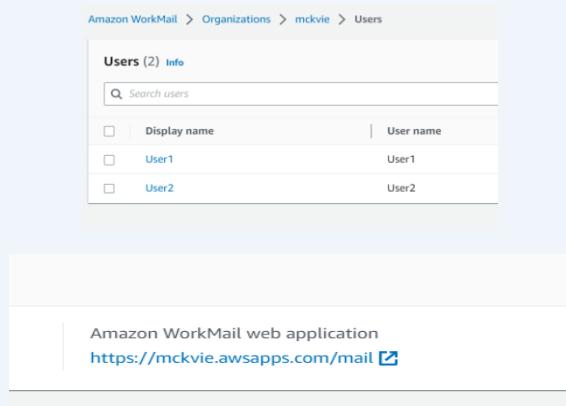
Create an Amazon WorkMail organization to provide email addresses to groups of users in your company. The email addresses include the domains that you select for your organization.

The screenshot shows the 'Organization settings' page. It has sections for 'Email domain' (with 'Free test domain' selected), 'Alias' (set to 'mckvie'), and 'Advanced settings'. At the bottom, there are 'Cancel' and 'Create organization' buttons.

4. Setup the user email and click on “Create user”

The screenshot shows the 'User details' and 'Email setup' pages for creating a new user. In 'User details', fields are filled for 'User name' (user1), 'First name - optional' (User), 'Last name - optional' (1), and 'Display name' (User1). In 'Email setup', fields are filled for 'Email address' (user1@mckvie.awssapps.com) and 'Password' (two password fields). At the bottom of each page, there are 'Cancel' and 'Create user' buttons.

- 6.** Create two such users. Go back to organization dashboard and click on the organization created by you. Now click on the Amazon WorkMail web application link.



The screenshot shows the 'Users' page in the Amazon WorkMail console. It lists two users: 'User1' and 'User2'. The 'Display name' column shows 'User1' and 'User2' respectively. The 'User name' column shows 'User1' and 'User2' respectively. There is a search bar at the top labeled 'Search users'.

Amazon WorkMail web application  
<https://mckvie.awssapps.com/mail>

- 7.** Login to the amazon workmail by any of the user of your organization's workmail



Please log in with your mckvie credentials

Username (not email address)

user1

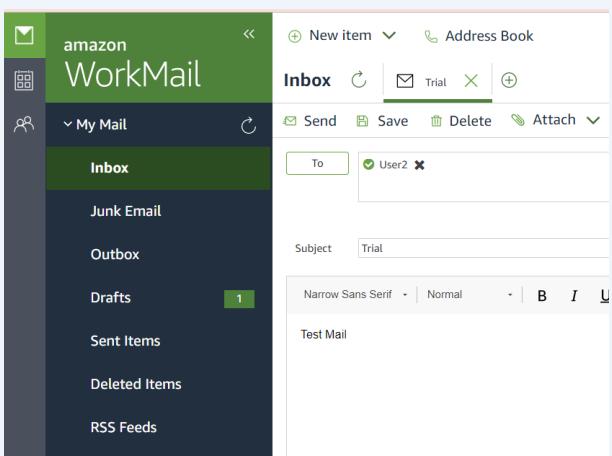
Remember username

Password

\*\*\*\*\*

**Sign In**

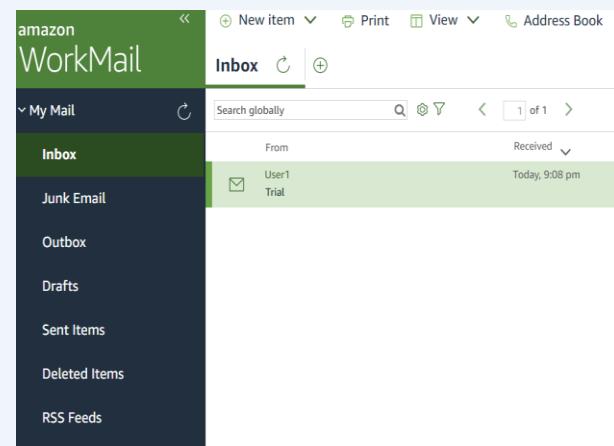
- 7.** Send an email from one user account of the organization to the another.



The screenshot shows the 'Inbox' screen in the Amazon WorkMail web application. A new email is being composed to 'User2'. The subject is 'Trial'. The message body contains the text 'Test Mail'.

- 8.** Check if the email is received by the other user of the same organization.

Hence organization's workmail is setup successfully.

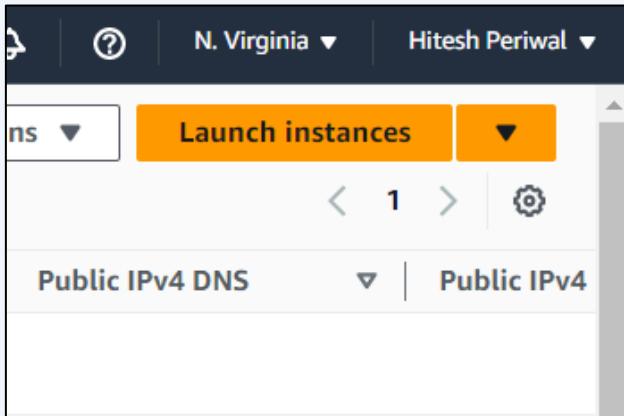


The screenshot shows the 'Inbox' screen in the Amazon WorkMail web application. An email from 'User1' with the subject 'Trial' is listed in the inbox. The message body shows 'User1 Trial Today, 9:08 pm'.

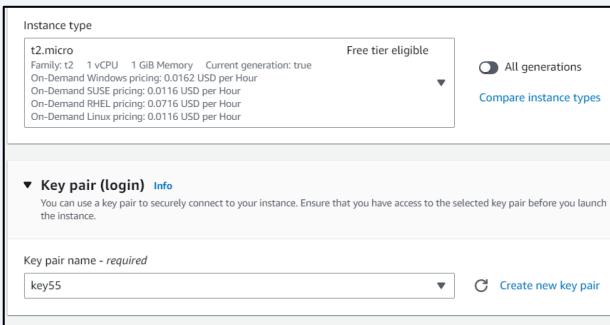
# Assignment No. 14

## Create an elastic IP for an instance.

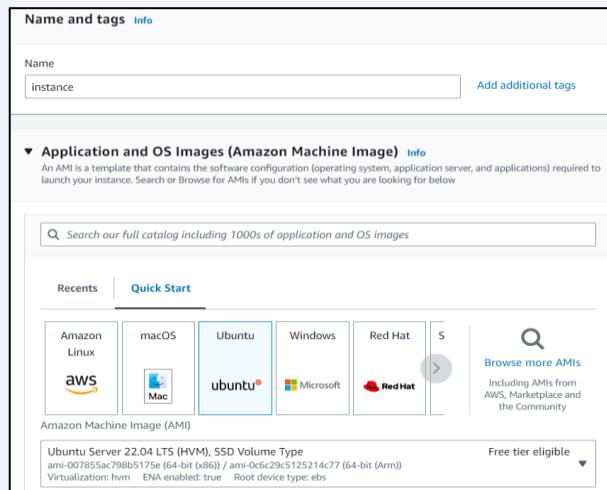
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Go to EC2 dashboard and click “Instance” and then click on “Launch instances”



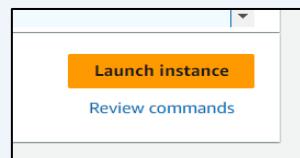
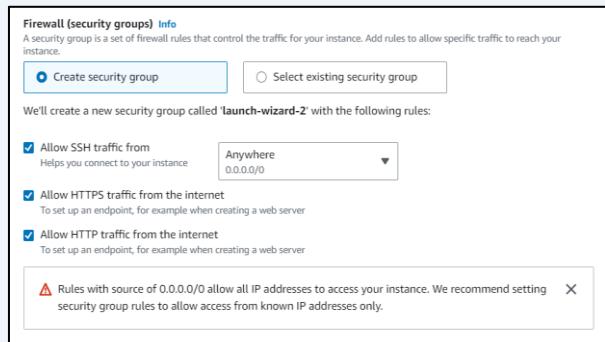
3. Select t2.micro as instance type and a keypair



2. Enter instance name and select Ubuntu as AMI



4. Under firewall select create security group and select all the three traffic options



- 5.** Now go to the instance created .Change its state to stop.After its successfully stopped ,again start it.It will be observed that its public IPv4 will change.

- 6.** Go to EC2 dashboard and select “Elastic IPs”.Then click on “Allocate Elastic IP address” and “Allocate”. Select the elastic ip address,go to “Actions” dropdown and select “Associate Elastic IP addresses”.

- 7.** Now select the instance and its private IP address.Click on checkbox which reads “Allow this Elastic IP address to be reassociated”.Then click on “Associate”

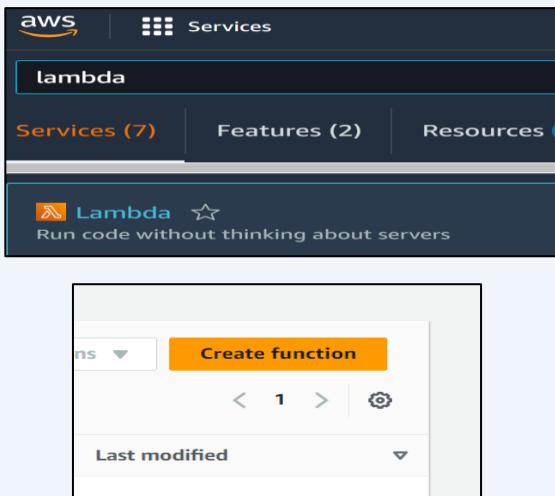
- 8.** Now go to the instance.Change its state to stop.After its successfully stopped ,again start it.It will be observed that its public IPv4 will remain same.Hence Elastic IP address is successfully associated with this instance.

- 9.** To remove this Elastic IP address we need to first disassociate it and then release it.

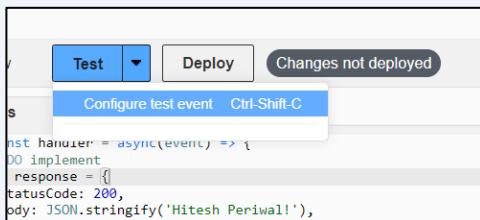
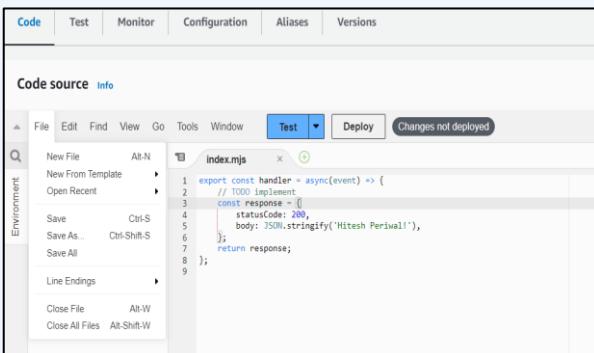
# Assignment No. 15

## Create serverless computing service

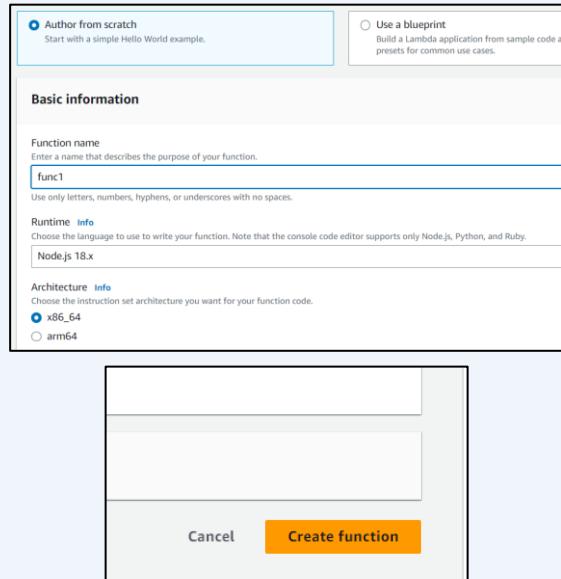
1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in. Go to Search box, type lambda and click on "Lambda". Now click on "Create function"



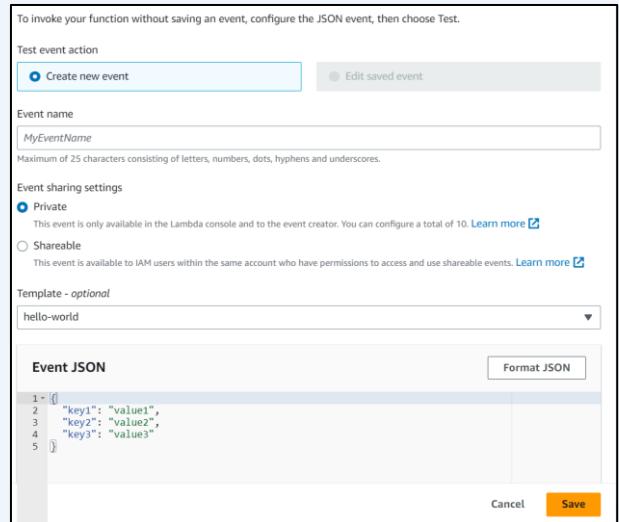
3. Go to Code section. You may edit the code and save it. Click on Test dropdown and select "Configure test event"



2. Type function name, select Runtime as "Node.js 18.x" and architecture as "x86\_64". Click on "Create function"



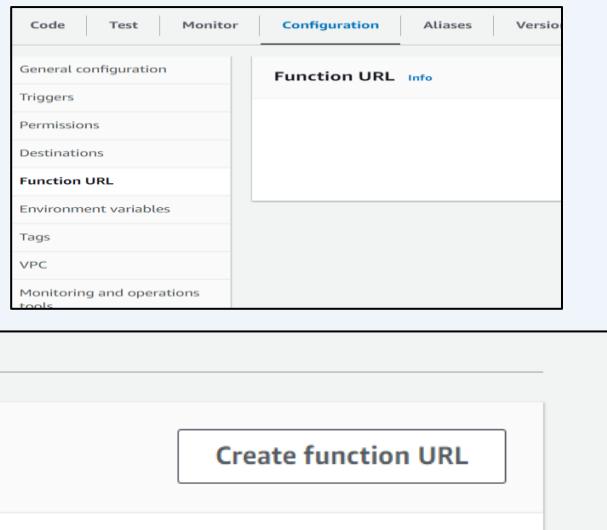
4. Select "Create new event", give event name, event sharing as private, select any template(optional) and click on "Save"



- 5.** Now test the event by clicking on “Test” and deploy it by clicking on “Deploy”

The screenshot shows the AWS Lambda Test & Deploy interface. The 'Test' tab is selected. Under 'Execution results', it shows a successful test event named 'Event'. The response body is: { "statusCode": 200, "body": "\"Hello from Lambda!\""}'. The function logs show a single log entry indicating a successful execution. The request ID is 85428538-78dc-48f7-970c-899d302224bc.

- 6.** Go to configuration ,select Function URL and click on “Create function URL”



- 7.** Select Authorization type as “NONE” and Click on“Save”

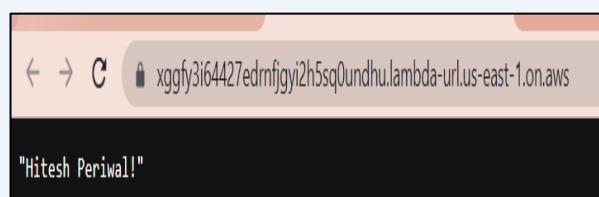
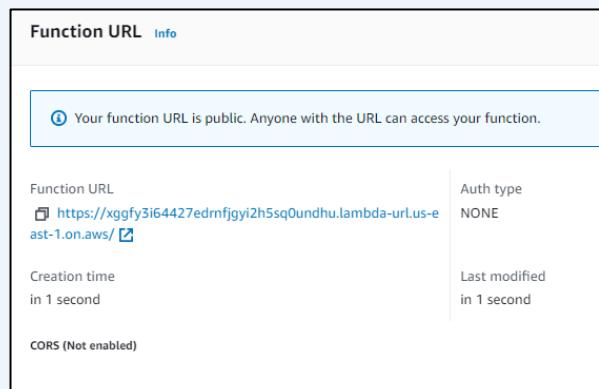
The screenshot shows the 'Function URL permissions' section of the AWS Lambda configuration. The 'Auth type' dropdown is set to 'NONE'. A note explains that Lambda won't perform IAM authentication and the endpoint will be public. The 'View policy statement' section shows the generated policy document:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "StatementId": "FunctionURLAllowPublicAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "lambda:InvokeFunctionUrl",
      "Resource": "arn:aws:lambda:us-east-1:294734627895:function:func1",
      "Condition": {
        "StringEquals": {
          "lambda:FunctionUrlAuthType": "NONE"
        }
      }
    }
  ]
}
  
```

At the bottom, there are 'Cancel' and 'Save' buttons.

- 8.** Function URL is thus generated.You can use it access and view your deployed project.



# Assignment No. 16

## Manage Amazon DNS service and run a project using domain-name and URL

1. Visit [aws.amazon.com](https://aws.amazon.com) and Sign in.
2. Go to EC2 Service
3. Create an instance with custom security group and user data.( As done in Assignment 10)
4. Click on the instance and copy the public IPv4 address and paste in the browser.
5. Check accessibility of project by appending port number after the public IPv4 address.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with options like EC2 Dashboard, Events, Tags, Limits, Instances, and Launch Templates. The main area displays resource counts: Instances (running) 0, Auto Scaling Groups 0, Dedicated Hosts 0; Elastic IPs 0, Instances 0, Key pairs 6; Load balancers 0, Placement groups 0, Security groups 2; Snapshots 0, Volumes 0. A message at the bottom says "Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more".

The screenshot shows the Instance summary for an EC2 instance with the ID i-08b0636b4c66a903b. It lists the instance ID, private IP (172.31.83.109), instance state (Running), and public IP (18.234.161.38). A tooltip indicates "Public IPv4 address copied".

The screenshot shows a web browser window with the title "Welcome to nginx!". The page content includes: "If you see this page, the nginx web server is successfully installed and working. Further configuration is required.", "For online documentation and support please refer to [nginx.org](http://nginx.org). Commercial support is available at [nginx.com](http://nginx.com).", and "Thank you for using nginx.".

The screenshot shows a web browser window with the title "Not secure | 18.234.161.38:4000". The page content is identical to the one in the previous screenshot, showing the Nginx welcome message.

6. Our EC2 instance works as intended.  
However, to access the webpage one always requires the public IPv4 address of the server instance which is very complicated/less accessible for end-users of our webpage/web application. To make it easier for our end-users, we need to bind a domain name to the server instance so that anyone can use the domain name and the URL to access the project.  
Now Search “Route 53” in the search bar of AWS console and click on “Route 53”

The screenshot shows the AWS search results for "route 53". The search bar contains "route 53". The results section shows "Services (4)" and "Features (19)". Under "Services", there is a card for "Route 53" with the subtext "Scalable DNS and Domain Name Registration". Under "Features", there is a card for "Route 53 Resolver" with the subtext "Resolve DNS queries in your Amazon VPC and on-premises network".

**7. We require a registered Domain name for this assignment. So, after obtaining one (free or paid) go to the Webpage of your Domain provider and log-in to your account where you can find all the details of your purchased Domains.**

This may vary from site to site, so you will have to do this based on what site you are using. We (for now) will be using GoDaddy.com, because we have purchased a Domain from them.

**8. In the Route 53 dashboard “Click on “Create Hosted Zone button”.( Alternatively, you can go to hosted zones from the left-side bar and then select create hosted zone option.)**

**9. Now, copy your Domain name from your Domain providers website. Here we used GoDaddy.com. Paste the domain name in the given field in Hosted Zone configuration page.**

**10. Click on “Create Hosted Zone”.**

**11. Click on “Create a record button”.**

The screenshot shows the 'Create hosted zone' configuration page. It includes fields for 'Domain name' (example.com), 'Description' (The hosted zone is used for...), and 'Type' (Public hosted zone). The 'Public hosted zone' option is selected, indicating it determines how traffic is routed on the internet.

The screenshot shows the 'Records' list page with two entries. It includes filters for 'Type', 'Routing policy', and 'Alias'. One entry is visible with the value 'ns-758.awsdns-30.net'.

**12. Need not to give any name. Keep the record name blank and record type as it is.**

**13. Under the value, copy and paste your server instance public IPv4 address which you want to route to using your DNS.Then click on create records button**

The screenshot shows the 'Quick create record' dialog. It has fields for 'Record name' (subdomain), 'Value' (192.0.2.35), 'TTL (seconds)' (300), and 'Routing policy' (Simple routing). The 'Create records' button is highlighted at the bottom right.

14. Click on the Create Record button again. But this time give the record name as "www". Select Record type as CNAME. In the text box under value, write the full domain-name there. (For example: example.com). Click on "create records".
15. Now select the record with type nameserver (NS) .( The values seen on the right-hand side are required for the next steps.)
16. Now go to your Domain providers webpage. Go to your purchased Domains settings.
17. Click on DNS section. (This may vary from provider to provider)
18. Click on the nameservers option.
19. Click on the Change nameservers and add here all the values opened in the Route 53 page. Select use my own nameservers option. Add nameservers. Then click on "Save".

The first screenshot shows the 'Quick create record' interface in Route 53. A CNAME record named 'www' is being created, pointing to the domain 'debrup.co.in'. The second screenshot shows the 'Hosted zone details' page in Route 53, listing four existing records: one SOA record and three NS records (ns-758.awsdns-30.net, ns-1483.awsdns-57.org, ns-2015.awsdns-59.co.uk, ns-327.awsdns-40.com). The third screenshot shows the 'Nameservers' section in a domain provider's control panel, where the user has selected 'I'll use my own nameservers' and entered the same four NS values.

This screenshot shows the 'Domain Portfolio' interface of a domain provider. The 'DNS' tab is selected, and the 'Nameservers' sub-tab is active. Other tabs include 'Overview', 'Forwarding', 'Products', 'DNS Records', 'Forwarding', 'Nameservers', 'Premium DNS', and 'Hostnames'.

This screenshot shows the 'Edit nameservers' dialog box. The user has chosen the 'I'll use my own nameservers' option and listed the four NS records: ns-758.awsdns-30.net, ns-1483.awsdns-57.org, ns-2015.awsdns-59.co.uk, and ns-327.awsdns-40.com. There is also an 'Add Nameserver' button.

**20. Search from any browser using your domain name with www.**

**21. Append port no. to access the project.**

Thus, we have successfully run our project using our custom domain-name and URL.

