ADVANCE DEVOPS EXPERIMENT 1
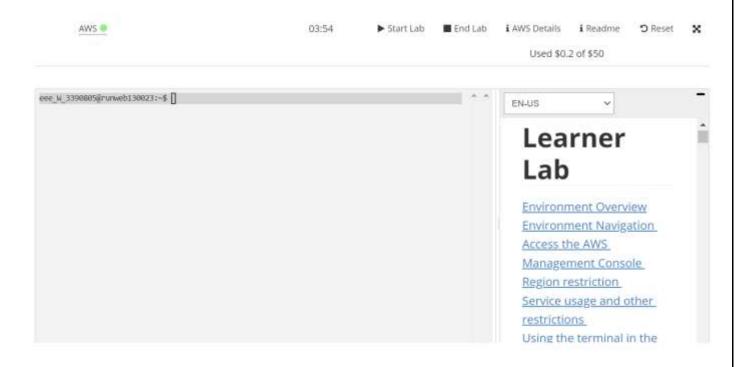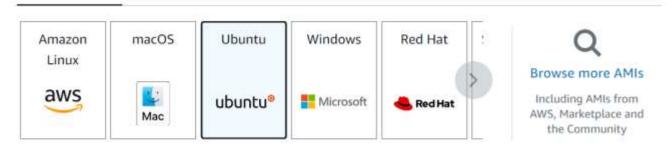
EC2 INSTANCE CREATION

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

SHRAVANI

Add additional tags

Quick Start

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	
aws	Mac	ubuntu®	Microsoft	Red Hat	Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type   Free tier eligible
ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
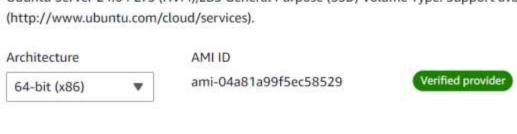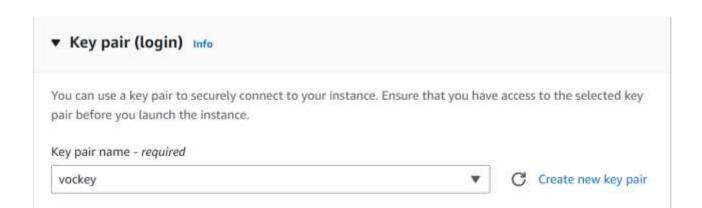Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description
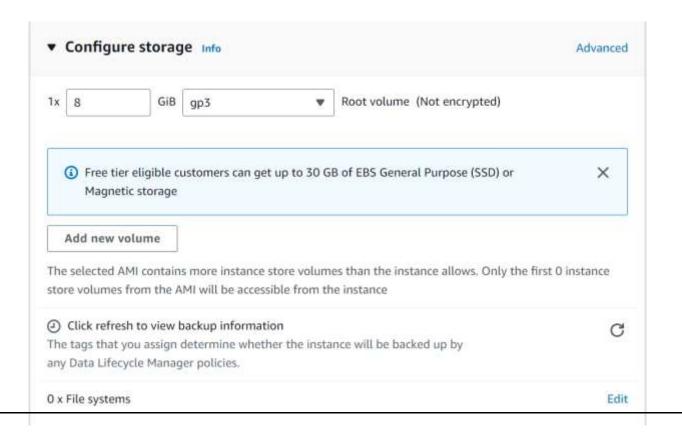
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
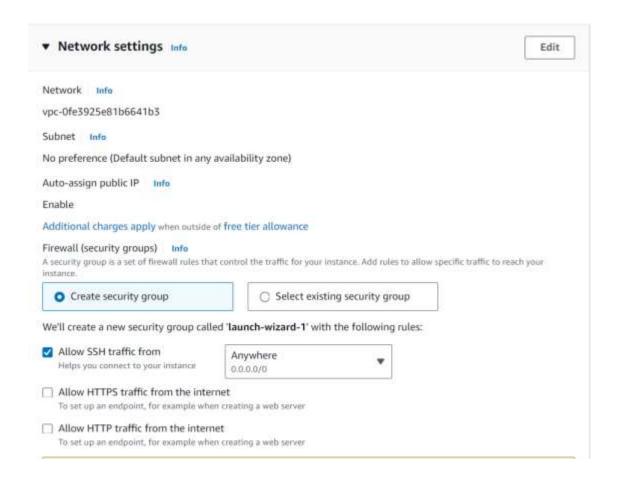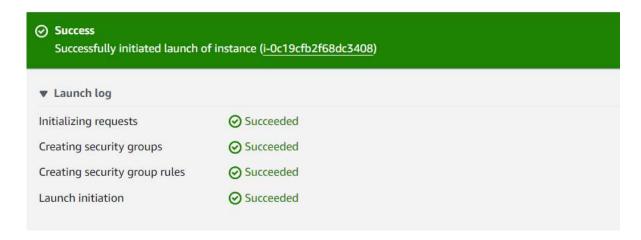
Architecture   AMI ID

64-bit (x86)   ▼   ami-04a81a99f5ec58529   Verified provider
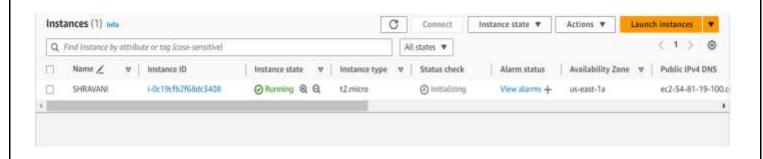
▼ **Instance type** Info | Get advice

Instance type

t2.micro	Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true	
On-Demand Windows base pricing: 0.0162 USD per Hour	
On-Demand SUSE base pricing: 0.0116 USD per Hour	
On-Demand RHEL base pricing: 0.026 USD per Hour	
On-Demand Linux base pricing: 0.0116 USD per Hour	

🔘 All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ **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*

vockey ▼   ↻ Create new key pair

▼ **Configure storage** Info   Advanced

1x   8   GiB   gp3   ▼   Root volume (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage   ✕

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

🕐 Click refresh to view backup information   ↻
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems   Edit

▼ **Network settings** Info   Edit

Network   Info

vpc-0fe3925e81b6641b3

Subnet   Info

No preference (Default subnet in any availability zone)

Auto-assign public IP   Info

Enable

Additional charges apply when outside of free tier allowance

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-1**' with the following rules:

☑ Allow SSH traffic from   Anywhere ▼
   Helps you connect to your instance   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

---

⊘ **Success**
   Successfully initiated launch of instance (**i-0c19cfb2f68dc3408**)

▼ **Launch log**

Initializing requests   ⊘ Succeeded

Creating security groups   ⊘ Succeeded

Creating security group rules   ⊘ Succeeded

Launch initiation   ⊘ Succeeded

Instances (1) Info   ⟳ Connect Instance state ▼ Actions ▼ **Launch instances** ▼

Q Find Instance by attribute or tag (case-sensitive)   All states ▼   ‹ 1 › ⚙

☐	Name ∠ ▽	Instance ID	Instance state ▽	Instance type ▽	Status check	Alarm status	Availability Zone ▽	Public IPv4 DNS
☐	SHRAVANI	i-0c19cfb2f68dc3408	⊘ Running ⊕ ⊖	t2.micro	⊕ Initializing	View alarms +	us-east-1a	ec2-54-81-19-100.c

Static website hosting using EC2

```
root@ip-172-31-38-150:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 l
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-s
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 1677 kB/2080 kB of archives.
After this operation, 8091 kB of additional disk space will be used.
```

```
Jul 30 09:05:25 ip-172-31-38-150 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 30 09:05:25 ip-172-31-38-150 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-38-150:/home/ubuntu# cd /var/www/html/
root@ip-172-31-38-150:/var/www/html# nano index.html
root@ip-172-31-38-150:/var/www/html# nano index1.html
root@ip-172-31-38-150:/var/www/html# cat index1.html
<h1>Hi Shravani here...</h1>
root@ip-172-31-38-150:/var/www/html#
```

Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.
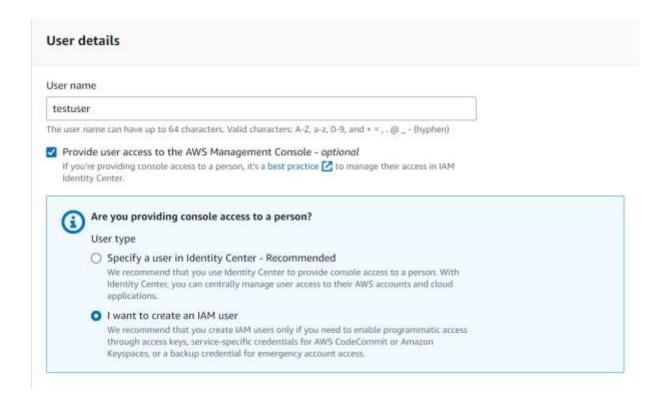
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
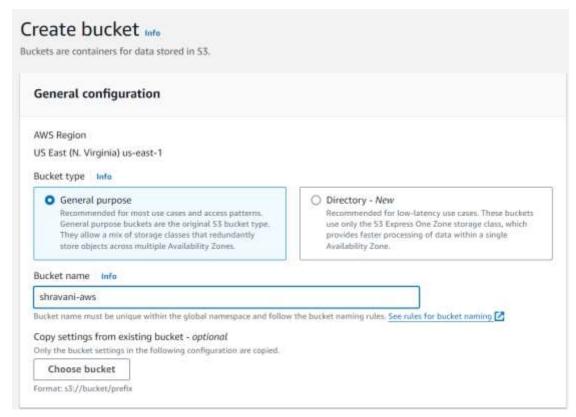
```
/etc/apache2/
|-- apache2.conf
```

← → C ⚠ Not secure 54.162.207.48/index1.html

Hi Shravani here...

STATIC WEBSITE HOSTING USING S3 BUCKET

User details

User name

testuser

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☑ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a best practice ☑ to manage their access in IAM Identity Center.

ⓘ **Are you providing console access to a person?**

User type

○ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

● I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

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

Bucket type   Info

● General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory - *New*

Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name   Info

shravani-aws

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming ☑

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

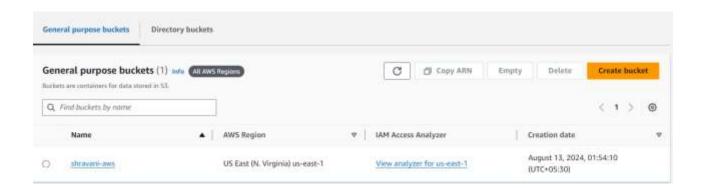Format: s3://bucket/prefix

`

▸ **Advanced settings**

ⓘ After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel   **Create bucket**

General purpose buckets   Directory buckets

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

🔍 Find buckets by name

↻   ⬚ Copy ARN   Empty   Delete   **Create bucket**

< 1 > ⚙

Name	▲	AWS Region	▽	IAM Access Analyzer		Creation date	▽
○ shravani-aws		US East (N. Virginia) us-east-1		View analyzer for us-east-1		August 13, 2024, 01:54:10 (UTC+05:30)	

Files and folders (13 Total, 3.6 MB)

🔍 Find by name

Name	Folder	▽	Type	▽	Size	▽	Status	▽	Error
facebook.sv... 🔗	-		image/svg+...		283.0 B		✅ Succeeded		-
hero.png 🔗	-		image/png		439.9 KB		✅ Succeeded		-
img1.jpg 🔗	-		image/jpeg		122.6 KB		✅ Succeeded		-
img2.jpg 🔗	-		image/jpeg		8.7 KB		✅ Succeeded		-
img3.jpg 🔗	-		image/jpeg		98.1 KB		✅ Succeeded		-
img4.jpg 🔗	-		image/jpeg		87.2 KB		✅ Succeeded		-
index.html 🔗	-		text/html		3.2 KB		✅ Succeeded		-
instagram.s... 🔗	-		image/svg+...		566.0 B		✅ Succeeded		-

`

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☐ **Block *all* public access**

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

☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Cancel   **Save changes**

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

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

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

● **ACLs enabled**
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.
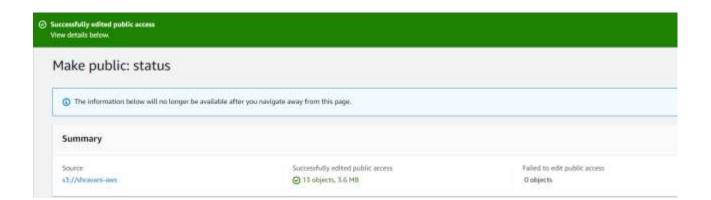
⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

⚠ **Enabling ACLs turns off the bucket owner enforced setting for Object Ownership**
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.

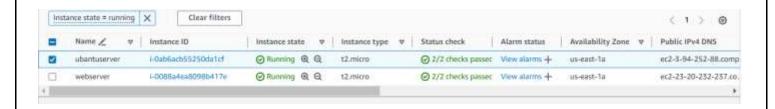☑ I acknowledge that ACLs will be restored.

⊘ **Successfully edited public access**
View details below.

Make public: status

ⓘ The information below will no longer be available after you navigate away from this page.

Summary

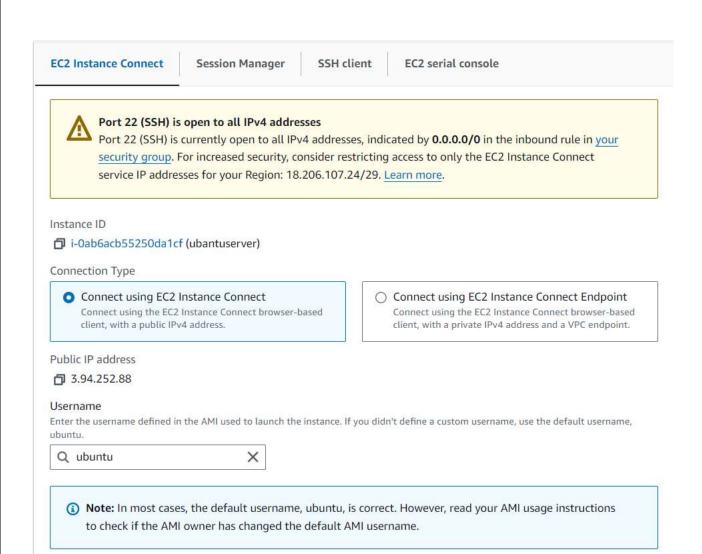| Source	Successfully edited public access	Failed to edit public access
s3://shravani-aws	⊘ 13 objects, 3.6 MB	0 objects

`

SHRAVANI RASAM D15A 46

HOSTED LINK:

https://shravani-aws.s3.amazonaws.com/index.html

`

EC2 DYNAMIC HOSTING

```
ubuntu@ip-172-31-38-150:~$ mkdir aws2
ubuntu@ip-172-31-38-150:~$ cd aws2
ubuntu@ip-172-31-38-150:~/aws2$ git clone https://github.com/ShravaniR2412/dynamic-hosting.git
Cloning into 'dynamic-hosting'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), 11.68 KiB | 2.92 MiB/s, done.
ubuntu@ip-172-31-38-150:~/aws2$
```

```
ubuntu@ip-172-31-38-150:~/aws2$ ls
dynamic-hosting
ubuntu@ip-172-31-38-150:~/aws2$ cd dynamic-hosting/
ubuntu@ip-172-31-38-150:~/aws2/dynamic-hosting$ ls
index.js  package-lock.json  package.json
ubuntu@ip-172-31-38-150:~/aws2/dynamic-hosting$ npm i
```
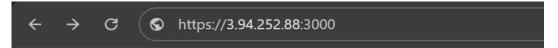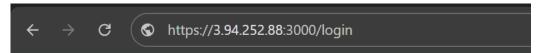
```
ubuntu@ip-172-31-38-150:~$ cd aws2
ubuntu@ip-172-31-38-150:~/aws2$ ls
dynamic-hosting
ubuntu@ip-172-31-38-150:~/aws2$ cd dynamic-hosting
ubuntu@ip-172-31-38-150:~/aws2/dynamic-hosting$ ls
index.js  package-lock.json  package.json
ubuntu@ip-172-31-38-150:~/aws2/dynamic-hosting$ npm i

added 93 packages, and audited 94 packages in 3s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ubuntu@ip-172-31-38-150:~/aws2/dynamic-hosting$ npm start
npm ERR! Missing script: "start"
npm ERR!
npm ERR! Did you mean one of these?
npm ERR!     npm star # Mark your favorite packages
npm ERR!     npm stars # View packages marked as favorites
npm ERR!
npm ERR! To see a list of scripts, run:
npm ERR!   npm run
npm ERR!
npm ERR! A complete log of this run can be found in:
npm ERR!     /home/ubuntu/.npm/_logs/2024-08-12T21_17_06_213Z-debug-0.log
ubuntu@ip-172-31-38-150:~/aws2/dynamic-hosting$ node index.js
Server is running on port 3000
```

https://3.94.252.88:3000
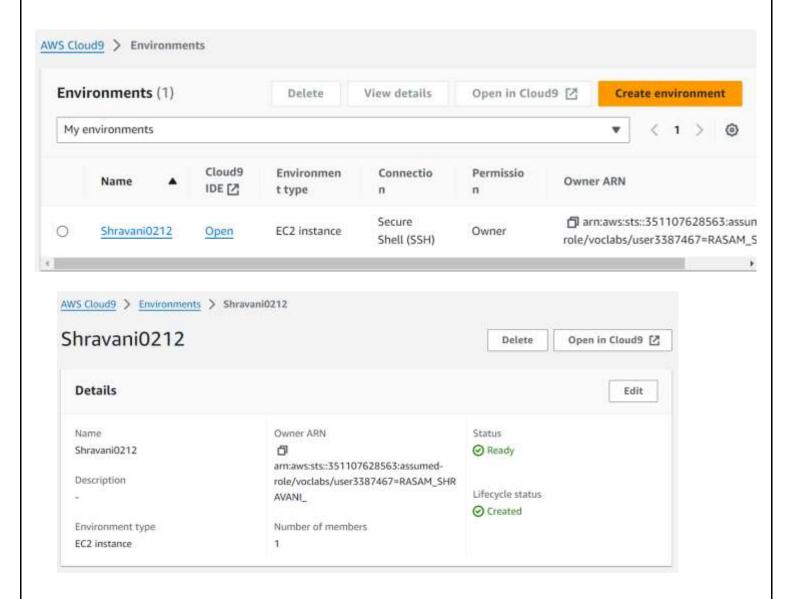
Hey Shravani this is Dynamic Website.

https://3.94.252.88:3000/login
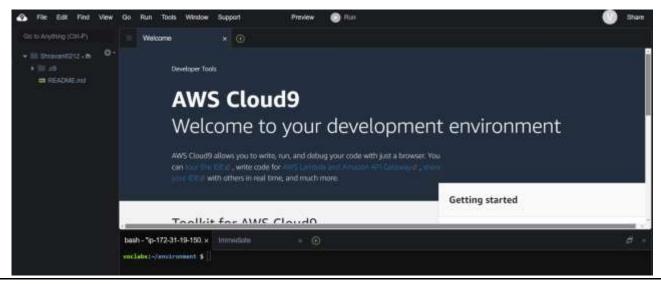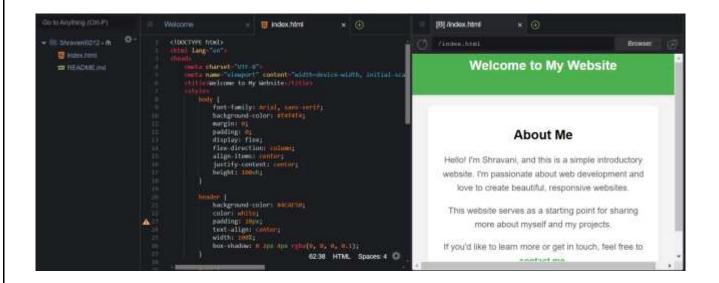
Hey this is login page.

`

CLOUD 9 HOSTING

AWS Cloud9 > Environments

Environments (1)

Delete | View details | Open in Cloud9 ☒ | **Create environment**

My environments ▼ | < 1 > | ⚙

Name ▲	Cloud9 IDE ☒	Environment type	Connection	Permission	Owner ARN
○ Shravani0212	Open	EC2 instance	Secure Shell (SSH)	Owner	⬚ arn:aws:sts::351107628563:assum role/voclabs/user3387467=RASAM_S

AWS Cloud9 > Environments > Shravani0212

Shravani0212

Delete | Open in Cloud9 ☒

Details

Edit

Name
Shravani0212

Description
-

Environment type
EC2 instance

Owner ARN
⬚
arn:aws:sts::351107628563:assumed-role/voclabs/user3387467=RASAM_SHR AVANI_

Number of members
1

Status
⊘ Ready

Lifecycle status
⊘ Created

User details

User name

shravani

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☑ Provide user access to the AWS Management Console - *optional*
   If you're providing console access to a person, it's a best practice 🔗 to manage their access in IAM Identity Center.

ⓘ **Are you providing console access to a person?**

User type

◯ Specify a user in Identity Center - Recommended
   We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

◉ I want to create an IAM user
   We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

`

Permissions options

○ **Add user to group**
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

○ **Copy permissions**
Copy all group memberships, attached managed policies, and inline policies from an existing user.

○ **Attach policies directly**
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

(i) **Get started with groups**
Create a group and select policies to attach to the group. We recommend using groups to manage user permissions by job function, AWS service access, or custom permissions. Learn more [↗]

[Create group]

▶ **Set permissions boundary - *optional***

Cancel   Previous   **Next**

Create user group

Name the group

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

```
webgrp
```

Maximum 128 characters. Use alphanumeric and '+=,.@-_' characters.

Add users to the group - *Optional* (1) Info

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

Q Search   ⟨ 1 ⟩ ⚙

☐	User name [↗] ▲	Groups	Last activity ▽	Creation time ▽
☐	shravani	0	None	1 minute ago

`

Attach permissions policies - *Optional* (946) Info

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

	Policy name ▲	Type ▽	Used as ▽	Description
⊞	📦 AdministratorAccess	AWS managed - j...	None	Provides full access to AWS service
⊞	📦 AdministratorAcce...	AWS managed	None	Grants account administrative per
⊞	📦 AdministratorAcce...	AWS managed	None	Grants account administrative per
⊞	📦 AlexaForBusinessD...	AWS managed	None	Provide device setup access to Alex
⊞	📦 AlexaForBusinessF...	AWS managed	None	Grants full access to AlexaForBusin
⊞	📦 AlexaForBusinessG...	AWS managed	None	Provide gateway execution access
⊞	📦 AlexaForBusinessLi...	AWS managed	None	Provide access to Lifesize AVS devi
⊞	📦 AlexaForBusinessP...	AWS managed	None	Provide access to Poly AVS devices

Filter by Type: All types ▼

⟨ 1 2 3 4 5 6 7 ... 48 ⟩ ⚙

IAM › User groups

User groups (1) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

[C] [Delete] [**Create group**]

⟨ 1 ⟩ ⚙

	Group name ▲	Users ▽	Permissions ▽	Creation time ▽
☐	webgrp	⚠ 0	⚠ Not defined	Now

IAM › Users › shravani

shravani Info

[Delete]

Summary

ARN
📋 arn:aws:iam::361769589277:user/shravani

Console access
⚠ Enabled without MFA

Access key 1
Create access key
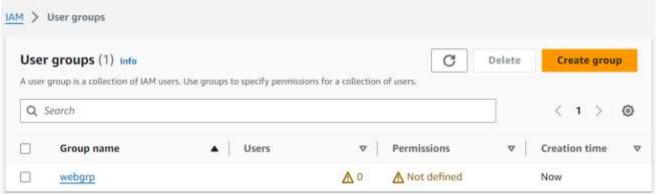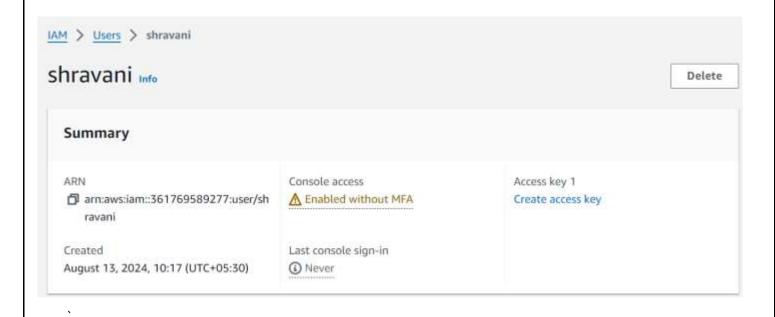
Created
August 13, 2024, 10:17 (UTC+05:30)

Last console sign-in
ⓘ Never

`

SHRAVANI RASAM D15A 46

`