

EXPERIMENT NO. 1

Aim : To understand the benefits of Cloud Infrastructure and Setup AWS Cloud9 IDE,Launch AWS Cloud9 IDE and Perform Collaboration Demonstration.

Theory:

Amazon Elastic Compute Cloud (Amazon EC2) provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 reduces hardware costs so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. You can add capacity (scale up) to handle compute-heavy tasks, such as monthly or yearly processes, or spikes in website traffic. When usage decreases, you can reduce capacity (scale down) again.

An EC2 instance is a virtual server in the AWS Cloud. When you launch an EC2 instance, the instance type that you specify determines the hardware available to your instance. Each instance type offers a different balance of compute, memory, network, and storage resources. For more information, see the Amazon EC2 Instance Types Guide.

Features of Amazon EC2 :

Amazon EC2 provides the following high-level features:

Instances Virtual servers.

Amazon Machine Images (AMIs)

Preconfigured templates for your instances that package the components you need for your server (including the operating system and additional software).

Instance types

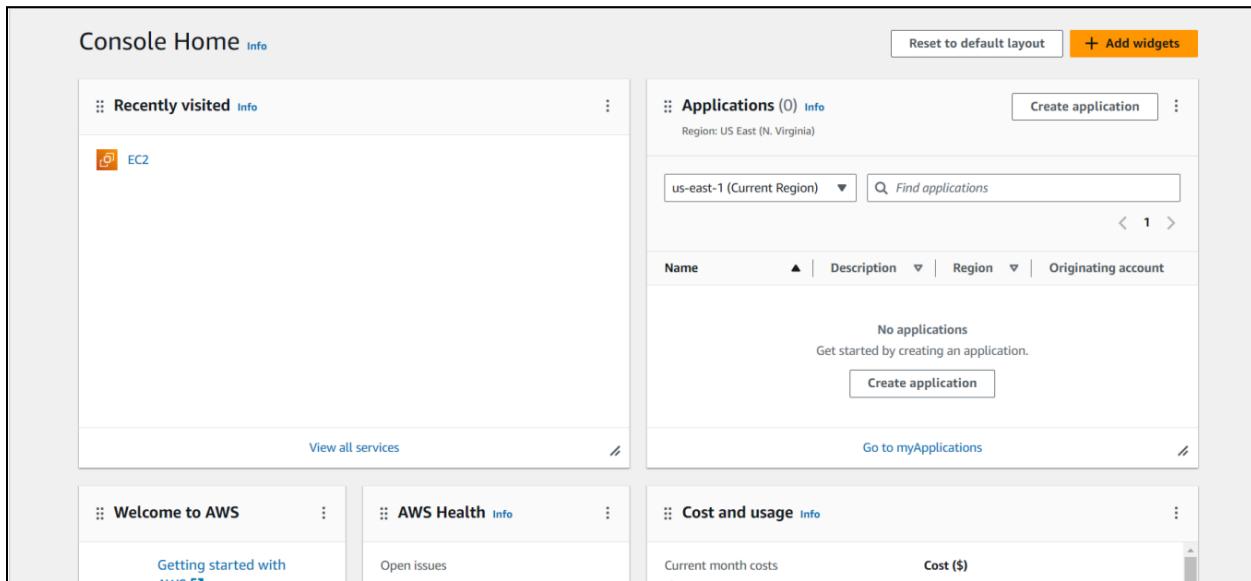
Various configurations of CPU, memory, storage, networking capacity, and graphics hardware for your instances

Amazon EBS volumes

Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS).

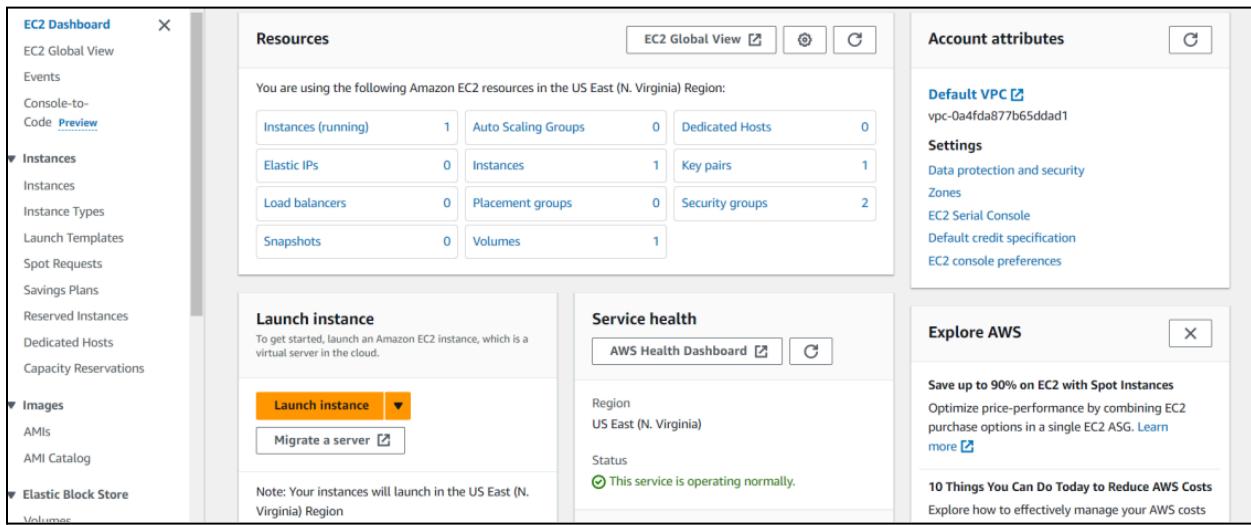
EC2 Instance Creation and static site hosting

1) Login to your AWS account:



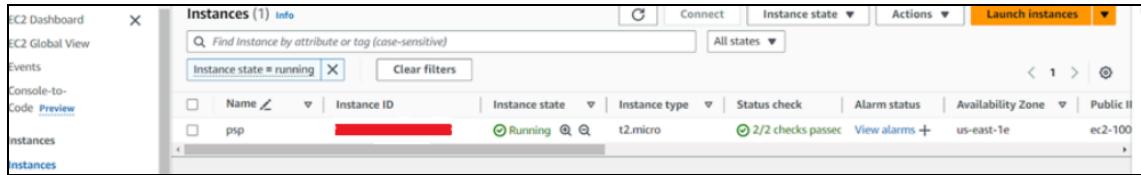
The screenshot shows the AWS Console Home page. On the left, under 'Recently visited', the 'EC2' service is highlighted with an orange icon. To the right, there's a section titled 'Applications' with a sub-section for 'us-east-1 (Current Region)'. It shows a search bar and a message: 'No applications. Get started by creating an application.' A 'Create application' button is present. Below this are sections for 'Cost and usage' and 'AWS Health'. At the bottom, there are links for 'Getting started with AWS' and 'Open issues'.

2) Click on EC2 and then create an instance by clicking on instances



The screenshot shows the AWS EC2 Dashboard. The left sidebar has sections for 'Instances', 'Images', and 'Elastic Block Store'. The main area is titled 'Resources' and displays a grid of EC2 metrics: Instances (running) 1, Auto Scaling Groups 0, Dedicated Hosts 0, Elastic IPs 0, Instances 1, Key pairs 1, Load balancers 0, Placement groups 0, Security groups 2, Snapshots 0, and Volumes 1. Below this is a 'Launch instance' button and a note about launching in the US East (N. Virginia) Region. To the right, there's a 'Service health' section showing the status as 'operating normally'. Further right are 'Account attributes' (Default VPC set to 'vpc-0a4fda877b65ddad1') and an 'Explore AWS' section with tips for cost reduction.

3) After an instance is created successfully .



4) After that you will go to command prompt and perform the following commands:

```
Last login: Thu Aug  8 08:40:05 2024 from 18.206.107.27
[ec2-user@ip-172-31-43-4 ~]$ wget https://github.com/ronak03rsk/IP_Lab_Exp1/archive/refs/heads/main.zip
--2024-08-08 08:55:27-- https://github.com/ronak03rsk/IP_Lab_Exp1/archive/refs/heads/main.zip
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/ronak03rsk/IP_Lab_Exp1/zip/refs/heads/main [following]
--2024-08-08 08:55:27-- https://codeload.github.com/ronak03rsk/IP_Lab_Exp1/zip/refs/heads/main
Resolving codeload.github.com (codeload.github.com)... 140.82.113.10
Connecting to codeload.github.com (codeload.github.com)|140.82.113.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'main.zip'
```

```
2024-08-08 08:55:27 (20.2 MB/s) - 'main.zip' saved [5978110]

[ec2-user@ip-172-31-43-4 ~]$ ls -lrt
total 5840
-rw-r--r--. 1 ec2-user ec2-user 5978110 Aug  8 08:55 main.zip
[ec2-user@ip-172-31-43-4 ~]$ cd aws_expla
-bash: cd: aws_expla: No such file or directory
[ec2-user@ip-172-31-43-4 ~]$ sudo su -
Last login: Thu Aug  8 08:44:56 UTC 2024 on pts/1
[root@ip-172-31-43-4 ~]# ls -lrt
total 0
drwxr-xr-x. 2 root root 29 Aug  8 08:51 aws_expla
[root@ip-172-31-43-4 ~]# cd aws_expla
[root@ip-172-31-43-4 aws_expla]# wget https://github.com/ronak03rsk/IP_Lab_Exp1/archive/refs/heads/main.zip
--2024-08-08 08:58:02-- https://github.com/ronak03rsk/IP_Lab_Exp1/archive/refs/heads/main.zip
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/ronak03rsk/IP_Lab_Exp1/zip/refs/heads/main [following]
--2024-08-08 08:58:02-- https://codeload.github.com/ronak03rsk/IP_Lab_Exp1/zip/refs/heads/main
Resolving codeload.github.com (codeload.github.com)... 140.82.114.9
Connecting to codeload.github.com (codeload.github.com)|140.82.114.9|:443... connected.
```

```
Resolving codeload.github.com (codeload.github.com)... 140.82.114.9
Connecting to codeload.github.com (codeload.github.com)|140.82.114.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'main.zip'

main.zip                                [ =>]

2024-08-08 08:58:02 (16.7 MB/s) - 'main.zip' saved [5978110]

[root@ip-172-31-43-4 aws_expla]# ls -lrt
total 6108
-rw-r--r--. 1 root root 271422 Aug  8 08:51 IP_Lab_Exp1.git
-rw-r--r--. 1 root root 5978110 Aug  8 08:58 main.zip
[root@ip-172-31-43-4 aws_expla]# unzip main.zip
Archive:  main.zip
af1da3ee5df3966b9c50a2aa25b1bf6de214ef2f
      creating: IP_Lab_Exp1-main/
      extracting: IP_Lab_Exp1-main/README.md
```

```

-rw-r--r--. 1 root root    7678 Aug  4 10:53 index.html
-rw-r--r--. 1 root root 5960098 Aug  4 10:53 heartbreak-piano-love-song-207235.mp3
drwxr-xr-x. 2 root root     94 Aug  4 10:53 assets
-rw-r--r--. 1 root root    13 Aug  4 10:53 README.md
[root@ip-172-31-43-4 IP_Lab_Expl-main]# mv * /var/www/html/
mv: target '/var/www/html/' is not a directory
[root@ip-172-31-43-4 IP_Lab_Expl-main]# mv * /var/www/html
[root@ip-172-31-43-4 IP_Lab_Expl-main]# cd /var/www/html
[root@ip-172-31-43-4 html]# ls -lrt
total 5836
-rw-r--r--. 1 root root    7678 Aug  4 10:53 index.html
-rw-r--r--. 1 root root 5960098 Aug  4 10:53 heartbreak-piano-love-song-207235.mp3
drwxr-xr-x. 2 root root     94 Aug  4 10:53 assets
-rw-r--r--. 1 root root    13 Aug  4 10:53 README.md
[root@ip-172-31-43-4 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
      Active: inactive (dead)
        Docs: man:httpd.service(8)
[root@ip-172-31-43-4 html]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service
[root@ip-172-31-43-4 html]# systemctl start httpd
[root@ip-172-31-43-4 aws_expla]# ls -lrt
total 6108
-rw-r--r--. 1 root root 271422 Aug  8 08:51 IP_Lab_Expl.git
-rw-r--r--. 1 root root 5978110 Aug  8 08:58 main.zip
[root@ip-172-31-43-4 aws_expla]# unzip main.zip
Archive: main.zip
af1da3ee5df3966b9c50a2aa25b1bf6de214ef2f
  creating: IP_Lab_Expl-main/
  extracting: IP_Lab_Expl-main/README.md
  creating: IP_Lab_Expl-main/assets/
  inflating: IP_Lab_Expl-main/assets/consulting.jpg
  inflating: IP_Lab_Expl-main/assets/consulting1.jpg
  inflating: IP_Lab_Expl-main/assets/consulting2.jpg
  extracting: IP_Lab_Expl-main/assets/download.png
  inflating: IP_Lab_Expl-main/heartbreak-piano-love-song-207235.mp3
  inflating: IP_Lab_Expl-main/index.html
[root@ip-172-31-43-4 aws_expla]# ls -lrt
total 6108
drwxr-xr-x. 3 root root    100 Aug  4 10:53 IP_Lab_Expl-main
-rw-r--r--. 1 root root 271422 Aug  8 08:51 IP_Lab_Expl.git
-rw-r--r--. 1 root root 5978110 Aug  8 08:58 main.zip
[root@ip-172-31-43-4 aws_expla]# cd IP_Lab_Expl-main
[root@ip-172-31-43-4 IP_Lab_Expl-main]# ls -lrt

```

5) After that the ip-address which was given while running the instance, copy that and paste that on chrome, make sure that it is http and not https



Ubuntu

Apache2 Default Page

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
```

- 6) After the commands being successfully executed and everything is perfectly fine you will see the website you wanted on the same IP address.

Not secure 98.80.248.184

 SK Company

[Services](#) | [About Us](#) | [Contact](#)

Home



Welcome to Sk Company! We provide exceptional *services* to meet your needs.

SK Consulting is a premier consulting firm dedicated to helping *businesses achieve their goals* through expert guidance and strategic insights. With a focus on delivering high-quality solutions across various industries, SK Consulting leverages a team of experienced consultants to drive innovation and foster growth.

About Us



Static Site Hosting using S3 bucket:

Step1: Create bucket

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](#)

pranavpolbucket

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

Step 2: Add resources

Upload succeeded

View details below.

Files and folders

Configuration

Files and folders (16 Total, 6.6 MB)

Find by name

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	6.5 KB	Succeeded	-
README.md	-	-	11.0 B	Succeeded	-
style.css	-	text/css	7.0 KB	Succeeded	-
appservice.p...	public/	image/png	346.9 KB	Succeeded	-
award.jpeg	public/	image/jpeg	198.2 KB	Succeeded	-
banner.jpg	public/	image/jpeg	21.2 KB	Succeeded	-
banner1.jpg	public/	image/jpeg	17.5 KB	Succeeded	-
cloud.png	public/	image/png	347.0 KB	Succeeded	-
conference.j...	public/	image/jpeg	174.4 KB	Succeeded	-
office.png	public/	image/png	324.3 KB	Succeeded	-

Step 3 : Provide public access

Edit Block public access (bucket settings) Info

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.

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

- Disable
 Enable

Hosting type

Host a static website

Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object

Redirect requests to another bucket or domain. [Learn more](#)

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

Index document

Specify the home or default page of the website.

index.html

Error document - optional

This is returned when an error occurs.

Edit Object Ownership Info

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 owner enforced setting.

Successfully edited public access
View details below.

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

Summary		
Source s3://pranavpolbucket	Successfully edited public access 13 objects, 3.5 MB	Failed to edit public access 0 objects

Step 4 : visit hosted website

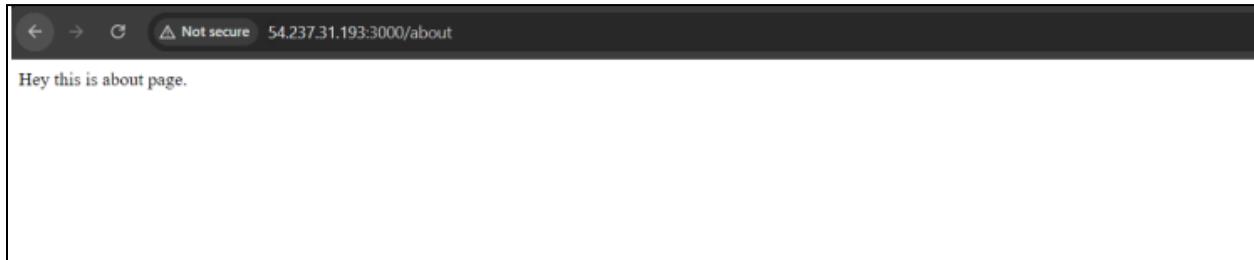


EC2 Dynamic Site Hosting:

Step 1 : Open Console and clone the github repository

```
([REDACTED]) :: reify:define-data-property: http fetch GET 200 https://registry.npmjs.org/define-data-property  
added 93 packages, and audited 94 packages in 3s  
16 packages are looking for funding  
  run `npm fund` for details  
found 0 vulnerabilities  
root@ip-172-31-55-145:/home/ubuntu/dynamic/dyanamic_site# npm start  
> hosting-dynamic-website@1.0.0 start  
> nodemon index.js  
  
[nodemon] 3.1.4  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,cjs,json  
[nodemon] starting `node index.js`  
Server is running on port 3000
```

Step 2 : Install necessary Packages and run website on port 3000



Cloud 9 IDE Site Hosting:

Step 1: Create Environment

AWS Cloud9 > Environments > Create environment

Create environment [Info](#)

Details

Name

Limit of 60 characters, alphanumeric, and unique per user.

Description - optional

Limit 200 characters.

Environment type [Info](#)
Determines what the Cloud9 IDE will run on.

New EC2 instance
Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

Existing compute
You have an existing instance or server that you'd like to use.

New EC2 instance

Step 2 :Open the Environment IDE

⌚ Successfully created WebAppIDE. To get the most out of your environment, see [Best practices for using AWS Cloud9](#) [X](#)

⌚ For capabilities similar to AWS Cloud9, explore AWS Toolkits in your own IDE and AWS CloudShell in the AWS Management Console. [Learn more](#) [X](#)

AWS Cloud9 > Environments

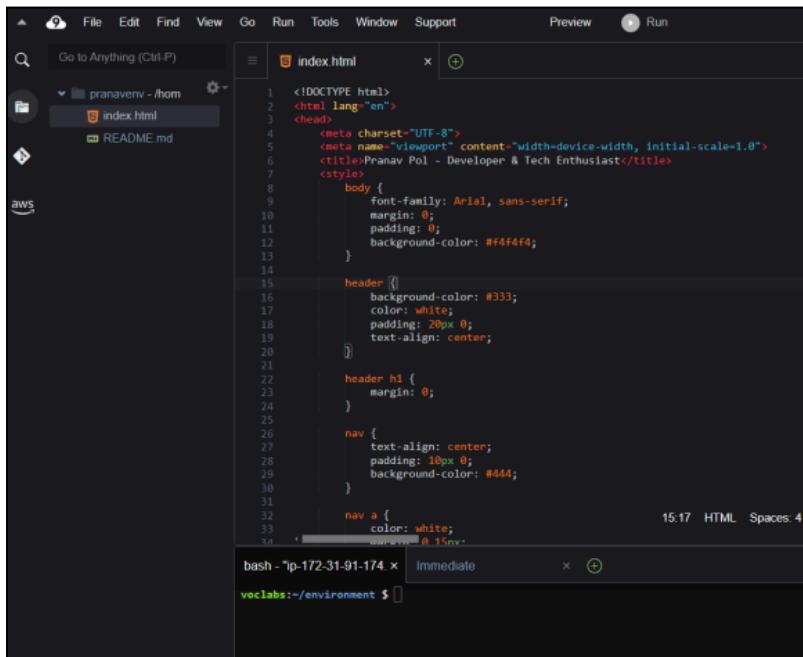
Environments (1)

[Delete](#) [View details](#) [Open in Cloud9](#) [Create environment](#)

My environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
WebAppIDE	Open	EC2 instance	Secure Shell (SSH)	Owner	 arn:aws:sts::773777131705:assumed-role/voclabs/user3402809=KATARIYA_RONAK

Step 3: Add the code and preview the website



The screenshot shows a terminal window with the following content:

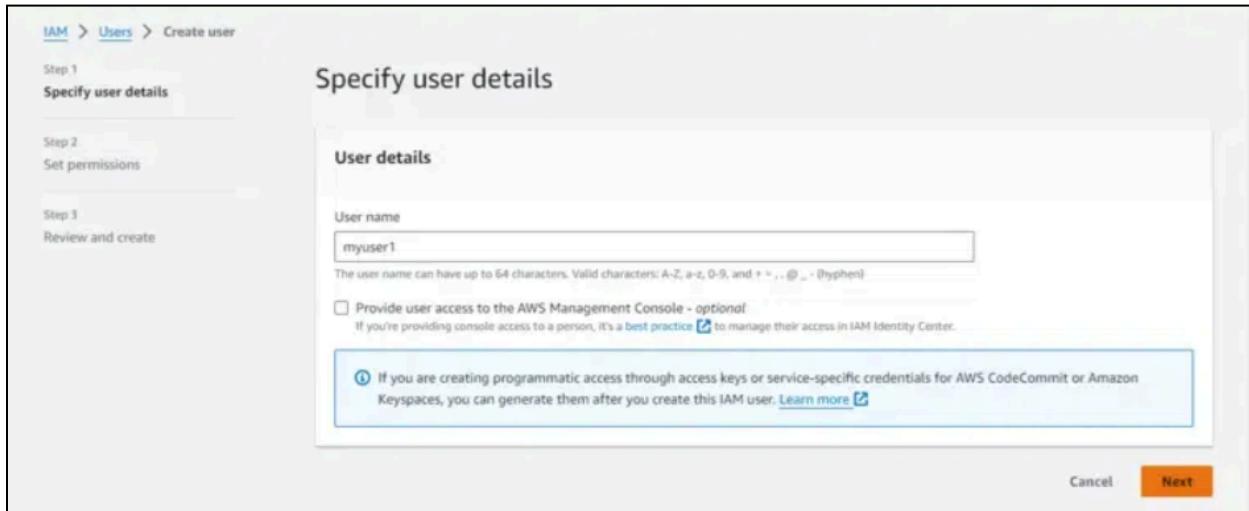
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pranav Pol - Developer & Tech Enthusiast</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }
        header {
            background-color: #333;
            color: white;
            padding: 10px 0;
            text-align: center;
        }
        header h1 {
            margin: 0;
        }
        nav {
            text-align: center;
            padding: 10px 0;
            background-color: #444;
        }
        nav a {
            color: white;
            text-decoration: none;
        }
    </style>
</head>
<body>
    <header>
        <h1>Pranav Pol - Developer & Tech Enthusiast</h1>
    </header>
    <nav>
        <a href="#">Home</a>
        <a href="#">About</a>
        <a href="#">Contact</a>
    </nav>
</body>

```

At the bottom of the terminal, there is a prompt: `vocabs:~/environment $`.

IAM:

Step 1: Create a user.



The screenshot shows the "Specify user details" step of the IAM user creation wizard. The left sidebar shows navigation steps: Step 1 (selected), Step 2 (Set permissions), and Step 3 (Review and create). The main form has the following fields:

- User details**:
 - User name**: myuser1
 - A note below says: "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](#).
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)"

At the bottom right are "Cancel" and "Next" buttons.

Step 2: Create a group.

Create user group

Name the group

User group name

Enter a meaningful name to identify this group.

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

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

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

< 1 >

Step 3: Give some role to the user.

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.

Filter by Type

All types

< 1 2 3 4 5 6 7 ... 48 >

Policy name	Type	Used as	Description
AdministratorAccess	AWS managed - j...	None	Provides full access to AWS services.
AdministratorAcce...	AWS managed	None	Grants account administrative permission.
AdministratorAcce...	AWS managed	None	Grants account administrative permission.
AlexaForBusinessD...	AWS managed	None	Provide device setup access to AlexaForBusiness.
AlexaForBusinessF...	AWS managed	None	Grants full access to AlexaForBusiness.
AlexaForBusinessG...	AWS managed	None	Provide gateway execution access.
AlexaForBusinessLi...	AWS managed	None	Provide access to Lifesize AVS devices.