EXPERIMENT NO. 1

NAME : PRANAV POL CLASS : D15A ROLL NO. : 42

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

Theory :

EC2 Hosting

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers by providing virtual servers, known as instances, to run applications.

Key Concepts of EC2:

1. **Instances:**
   An instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications. Instances can be launched on demand and scaled according to the needs of the application.
2. **AMI (Amazon Machine Image):**
   AMIs are pre-configured templates for instances. They include the operating system, application server, and applications themselves, allowing for quick and consistent instance launches.
3. **Instance Types:**
   EC2 offers a variety of instance types optimized for different use cases, such as compute-optimized, memory-optimized, and storage-optimized instances. Each instance type offers different combinations of CPU, memory, storage, and networking capacity.
4. **Elastic IP Addresses:**
   Elastic IPs are static IP addresses that can be associated with an EC2 instance. They are particularly useful for maintaining a consistent IP address even if the underlying instance changes.
5. **Security Groups:**
   Security groups act as virtual firewalls that control the traffic to and from an EC2 instance. You can configure rules to allow or deny traffic based on IP addresses, ports, and protocols.

6. **Auto Scaling:**
   Auto Scaling allows you to automatically adjust the number of EC2 instances in your application environment according to the current demand, ensuring optimal performance and cost-efficiency.

Amazon S3 (Simple Storage Service)

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. S3 is used to store and retrieve any amount of data, at any time, from anywhere on the web.

Key Concepts of S3:

1. **Buckets:**
   Buckets are containers for storing objects in S3. Each object is stored in a bucket, and each bucket is unique across the AWS environment. Buckets are used to organize and manage the data stored in S3.
2. **Objects:**
   Objects are the fundamental entities stored in S3. Each object consists of data, metadata, and a unique identifier (key). Objects can be any type of data, including images, videos, documents, or binary files.
3. **Keys:**
   Keys are unique identifiers for objects within a bucket. Each object in S3 is assigned a unique key that can be used to access and manage the object.
4. **Versioning:**
   S3 allows you to maintain multiple versions of an object within a bucket. Versioning helps protect against accidental overwrites and deletions by preserving older versions of objects.
5. **Access Control:**
   S3 provides several mechanisms for controlling access to data, including bucket policies, access control lists (ACLs), and IAM (Identity and Access Management) policies. These controls help ensure that only authorized users can access and manage the data.
6. **Lifecycle Management:**
   Lifecycle policies in S3 allow you to automate the transition of objects between different storage classes or delete them after a specified period. This helps optimize storage costs by moving data to less expensive storage as it ages.

7. **Storage Classes:**
   S3 offers various storage classes designed for different use cases, such as S3 Standard for frequently accessed data, S3 Intelligent-Tiering for automatically optimizing costs, and S3 Glacier for long-term archival storage.

AWS Cloud9 Infrastructure

AWS Cloud9 is a cloud-based integrated development environment (IDE) that allows you to write, run, and debug code with just a browser. It supports multiple programming languages, including Python, JavaScript, and more. AWS Cloud9 comes pre-packaged with essential tools and libraries, making it easier for developers to start coding without the need for complex setup processes.

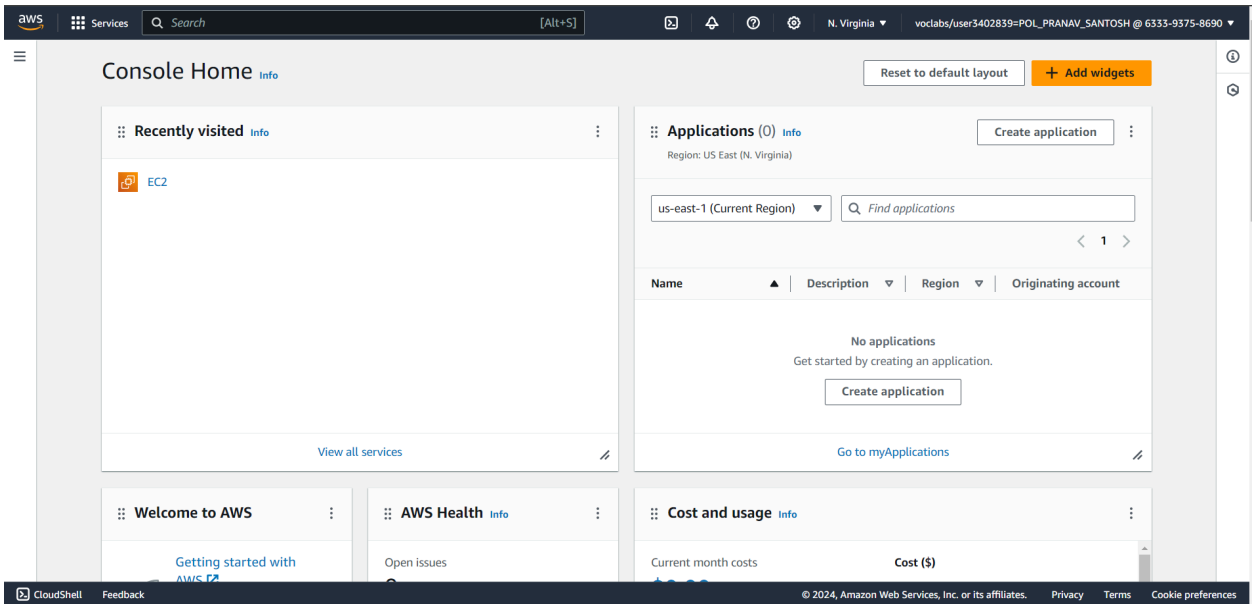
Key Features of AWS Cloud9:

1. **Cloud-Based IDE:**
   AWS Cloud9 provides a full-featured development environment accessible through a web browser. This eliminates the need for local IDE installations and configurations.
2. **Collaborative Development:**
   Developers can collaborate in real-time, with multiple users able to work on the same project simultaneously. It includes features like chat and simultaneous editing, making it ideal for pair programming and team collaborations.
3. **Pre-configured Environment:**
   Cloud9 is pre-configured with essential tools and libraries for various languages and frameworks. This allows developers to start coding immediately without worrying about environment setup.
4. **Seamless Integration with AWS Services:**
   AWS Cloud9 integrates seamlessly with other AWS services like EC2, S3, and Lambda, allowing developers to easily deploy and manage their applications directly from the IDE.
5. **Terminal Access:**
   Cloud9 provides full terminal access to the underlying instance, giving developers the ability to run shell commands and manage their environment directly.

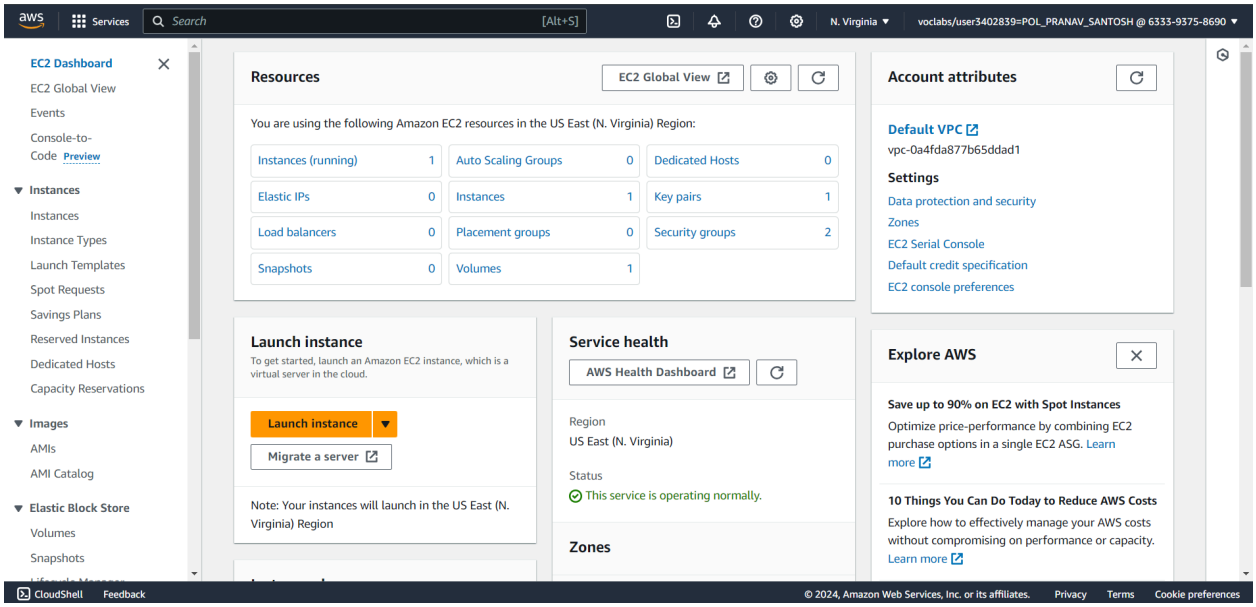
Implementation :

EC2 Instance Creation and static site hosting

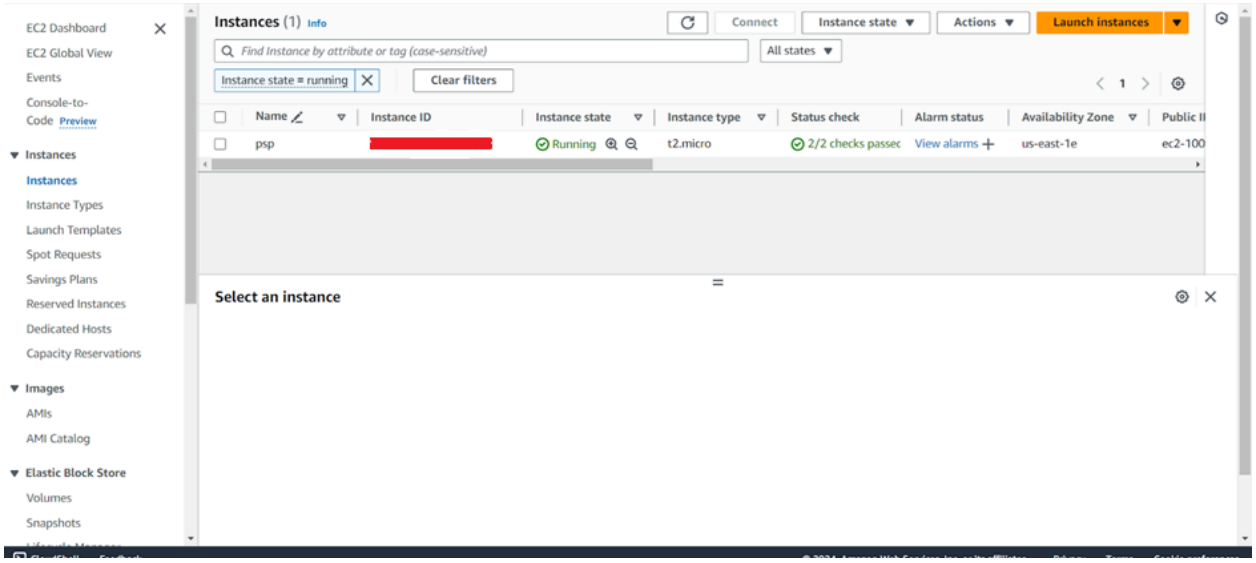
1) Login to your AWS account



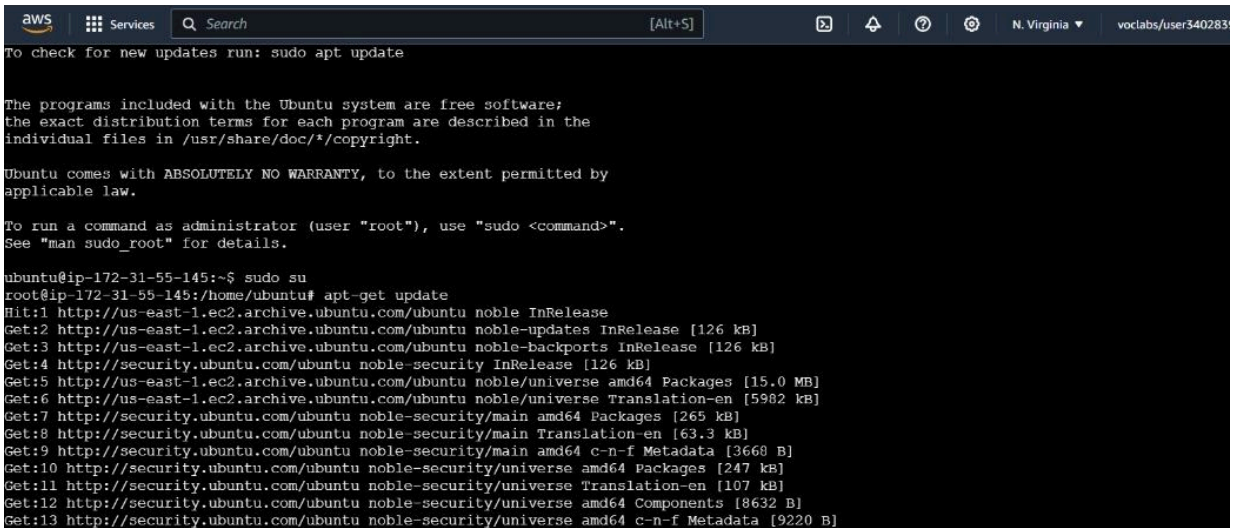
2) Click on EC2 and then create an instance by clicking on instances



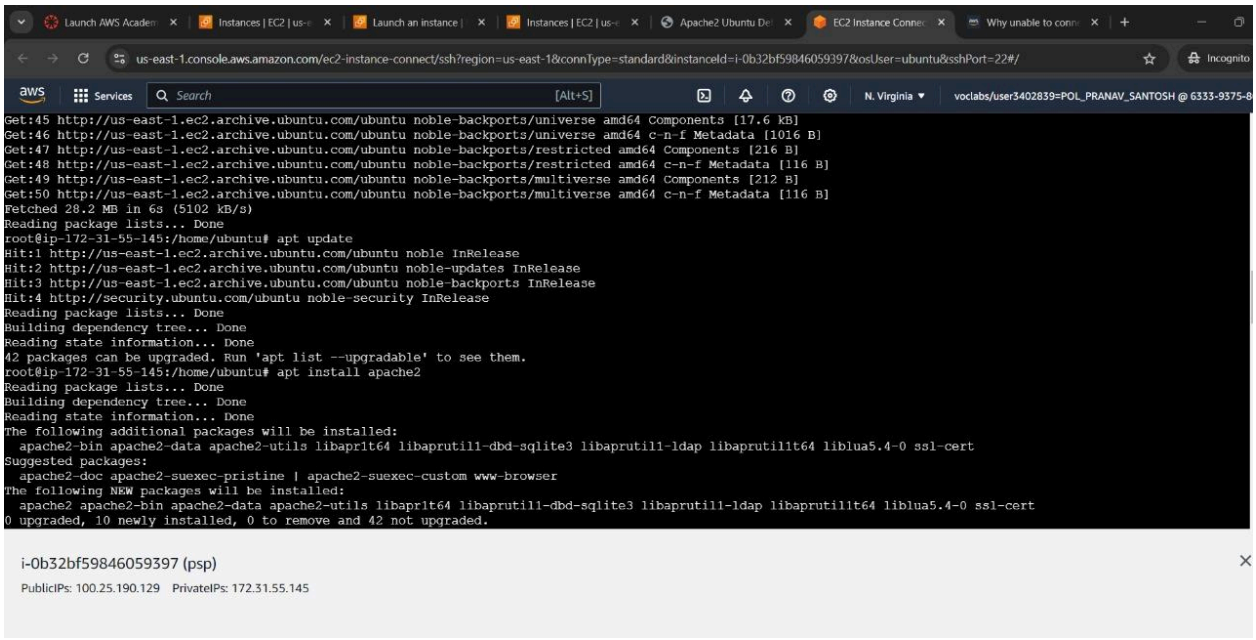
3) After an instance is created wait for it to come to Running state

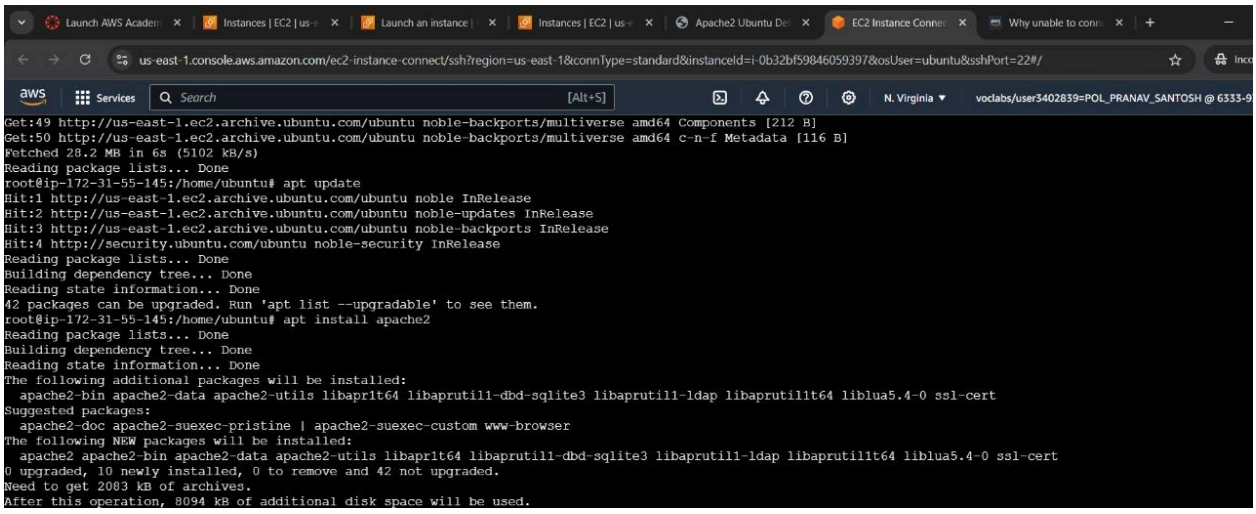


4) After doing that you will see this UI

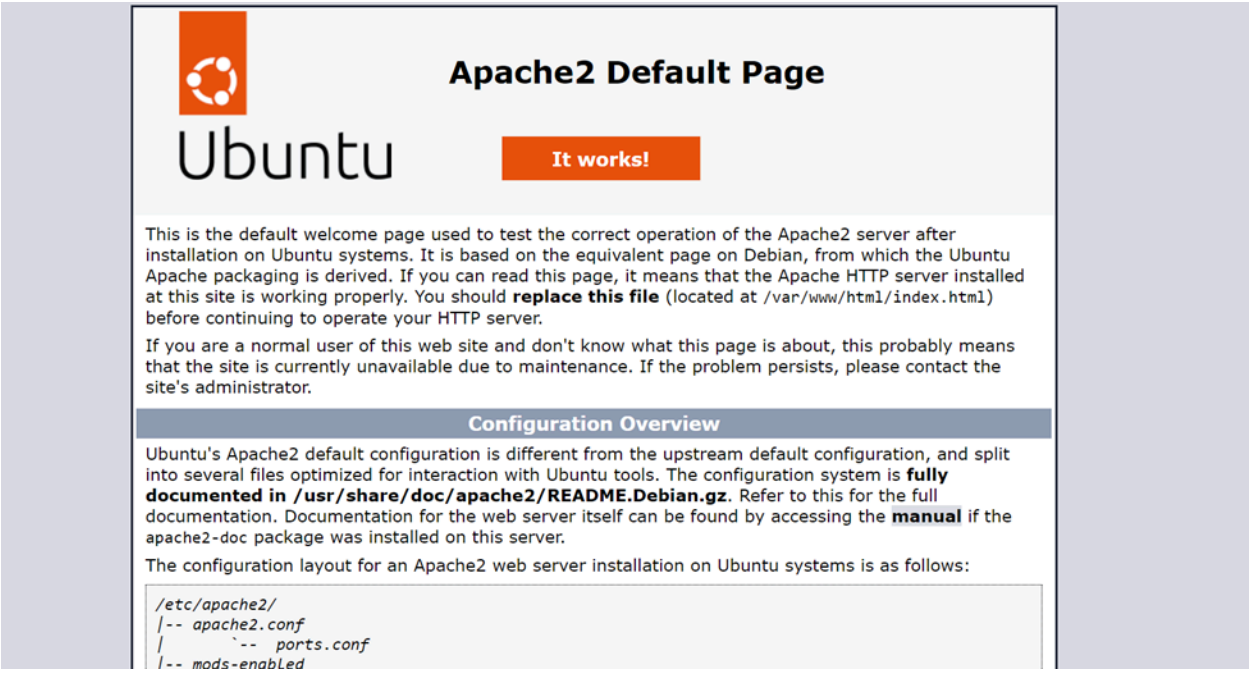


5) Follow these steps and then run these commands

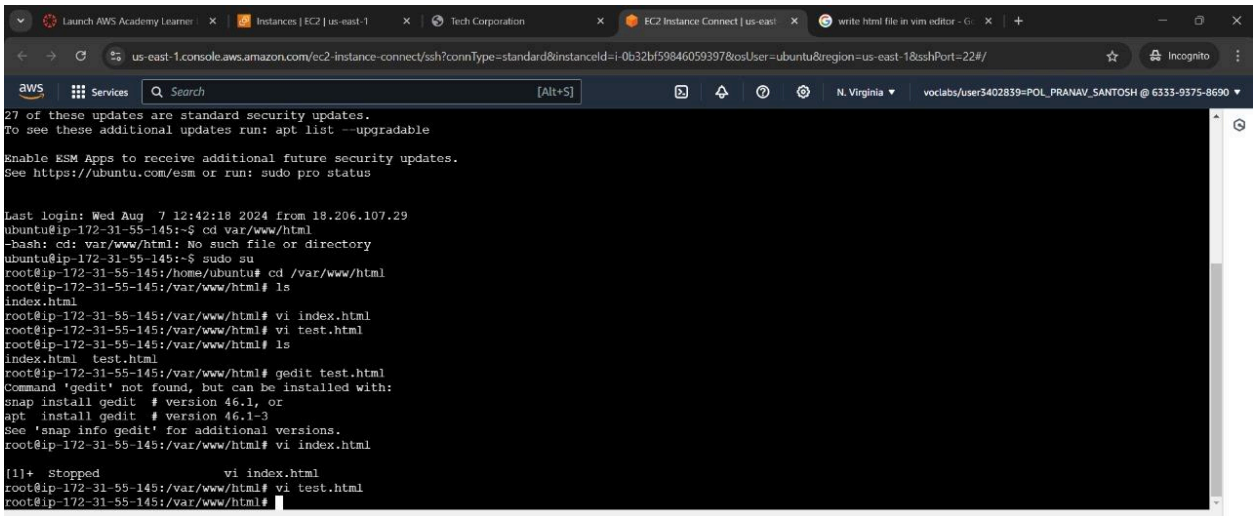




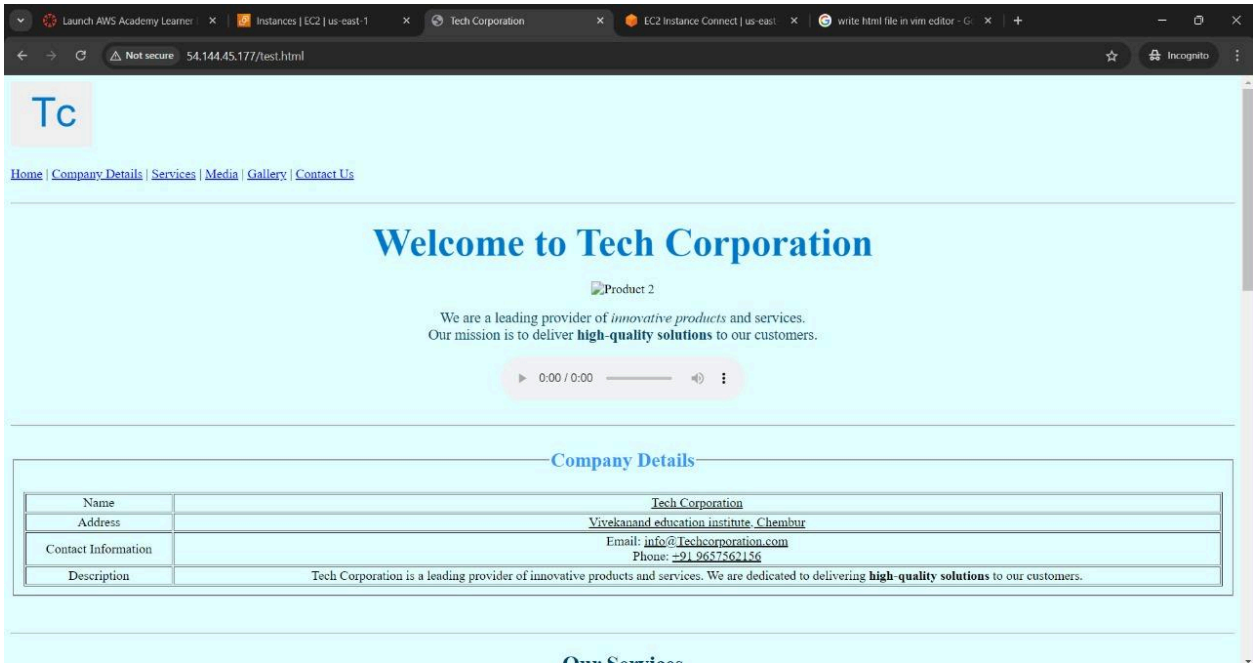
6) After that the ip-address which was given while running the instance, copy that and paste that on chrome, make sure that it is http and not https



7) Create a file using vi command and save it using :wq

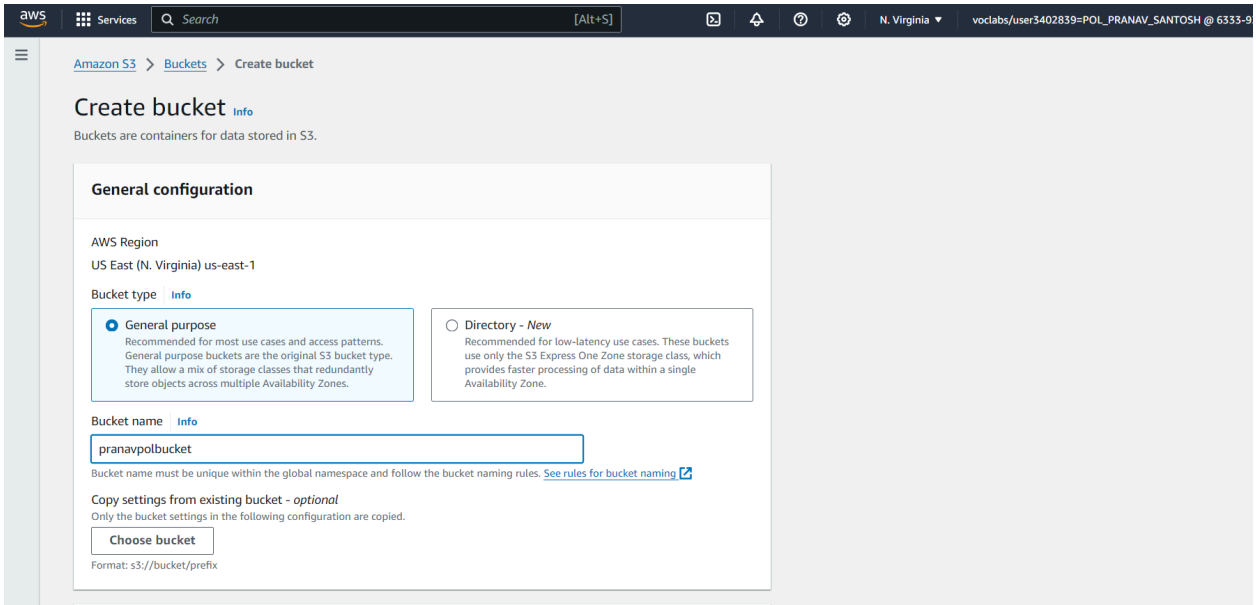


8) After saving that file go that page where ubuntu is listed and then on the link add “/your_file_name.html” and then whatever you saved on that file will be displayed

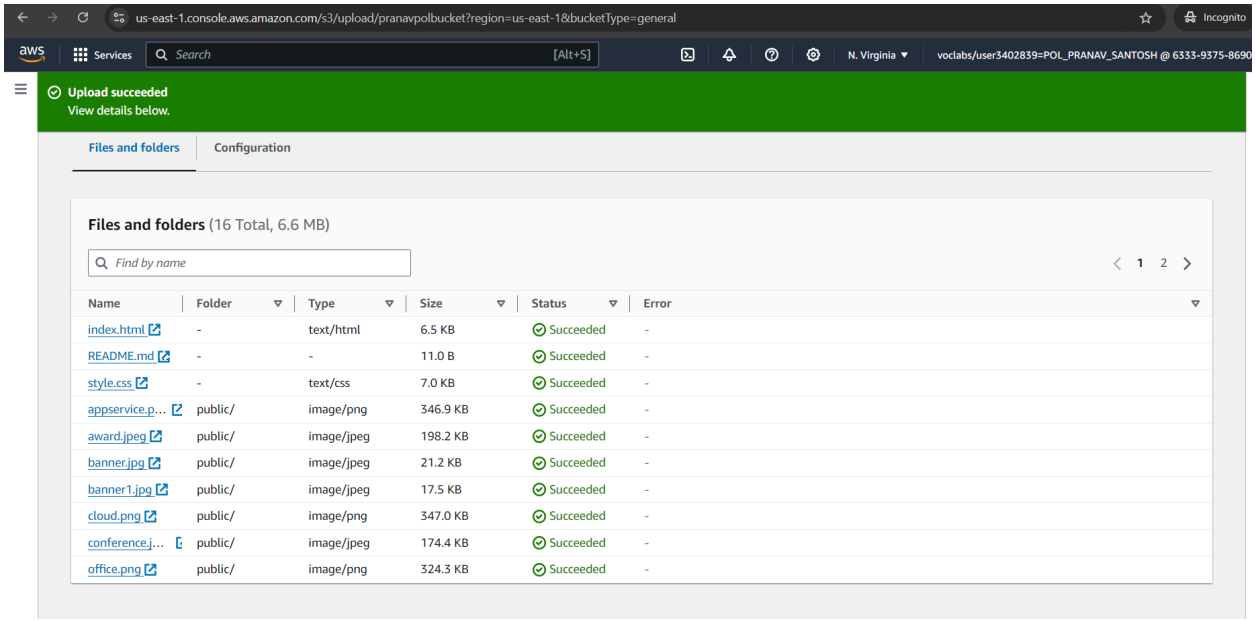


Static Site Hosting using S3 bucket

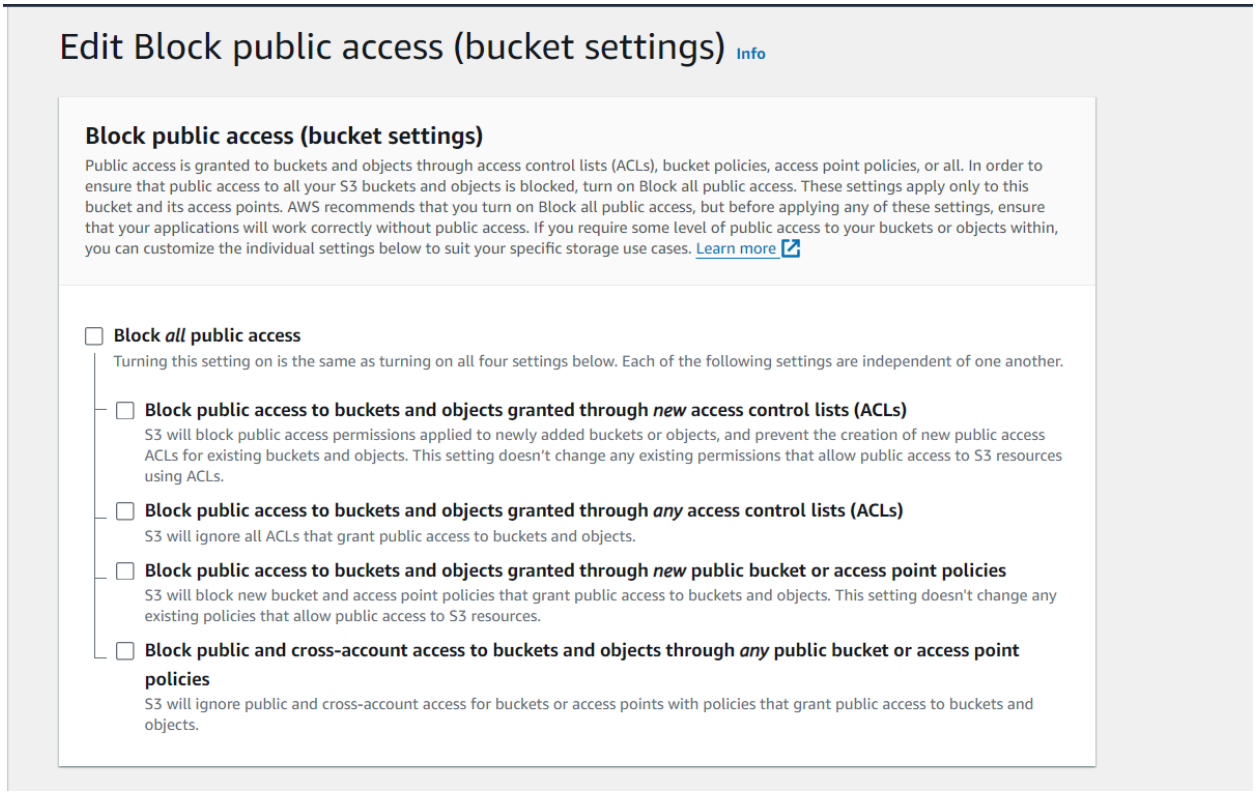
Step1: Create bucket

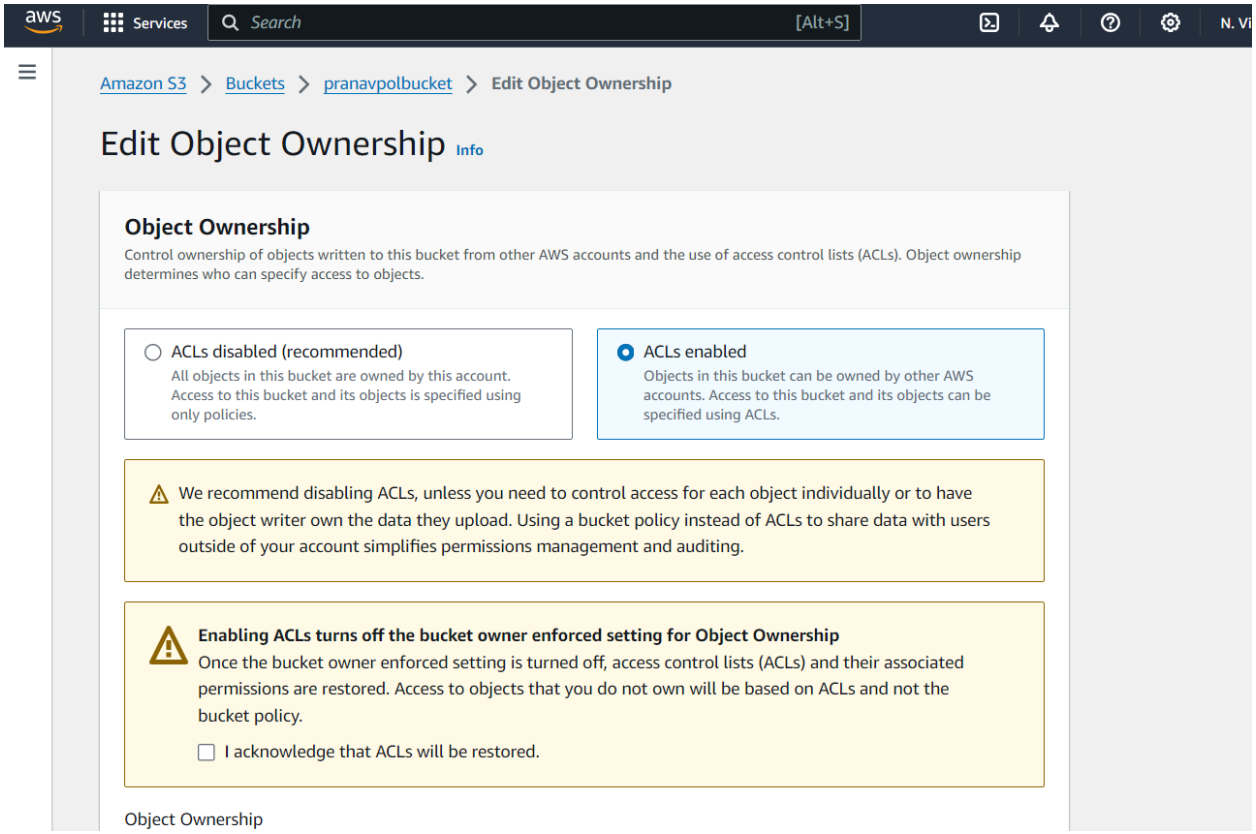


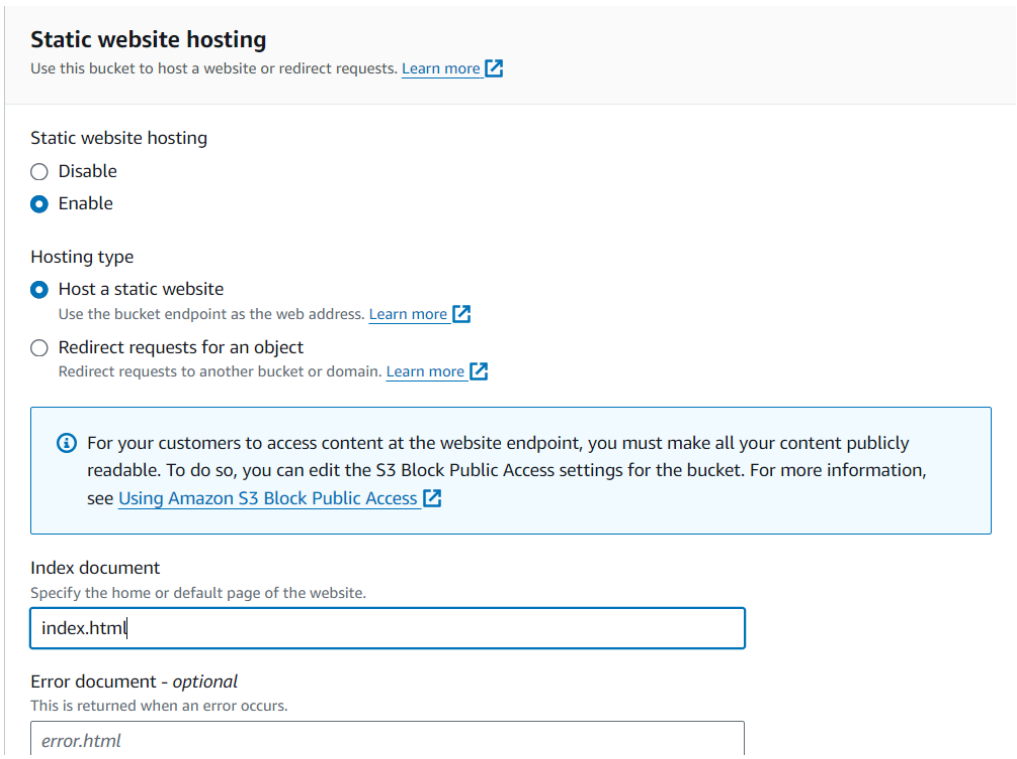
Step 2: Add resources

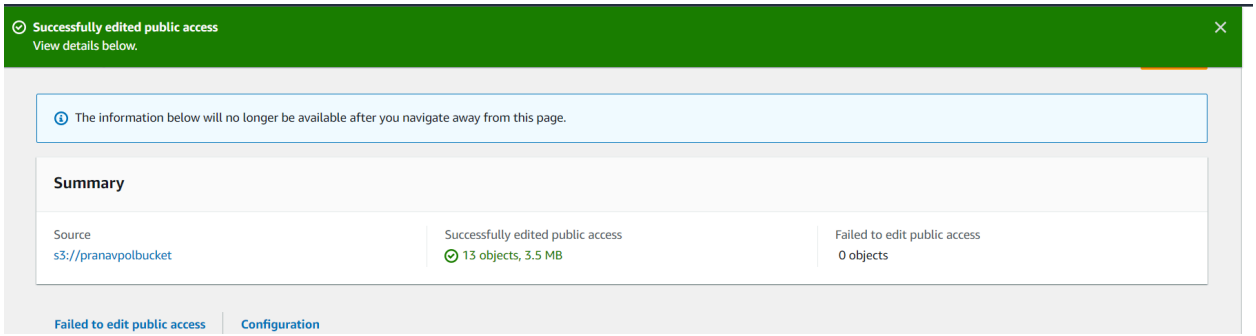


Step 3 : Provide public access

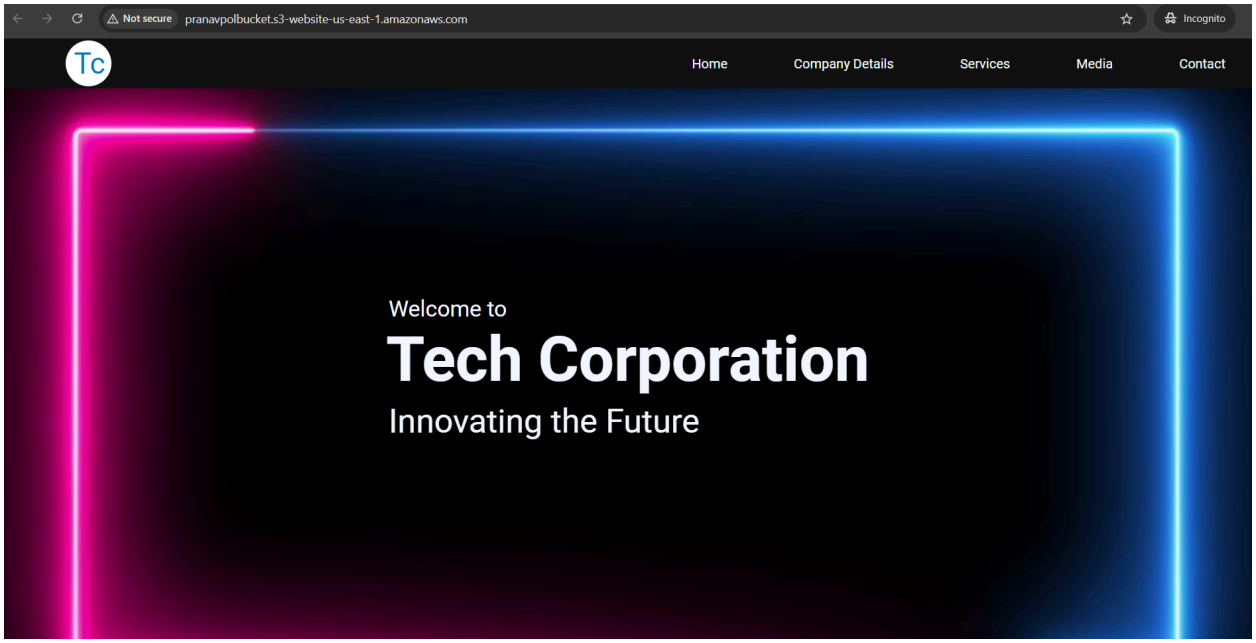



aws
Services
Search
[Alt+S]
N. Vi
Amazon S3 > Buckets > pranavpolbucket > Edit Object Ownership
Edit Object Ownership Info
Object Ownership
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify 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 policy.
I acknowledge that ACLs will be restored.
Object Ownership


Static website hosting
Use this bucket to host a website or redirect requests. Learn more
Static website hosting
Disable
Enable
Hosting type
Host a static website
Use the bucket endpoint as the web address. Learn more
Redirect requests for an object
Redirect requests to another bucket or domain. Learn more
For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access
Index document
Specify the home or default page of the website.
index.html
Error document - optional
This is returned when an error occurs.
error.html



Step 4 : visit hosted website



EC2 Dynamic Site Hosting

Step 1 : Open Console and clone the github repository

```
root@ip-172-31-55-145:/home/ubuntu/dynamic/dyanamic_site# npm i
(                    ) ⠼ 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
```

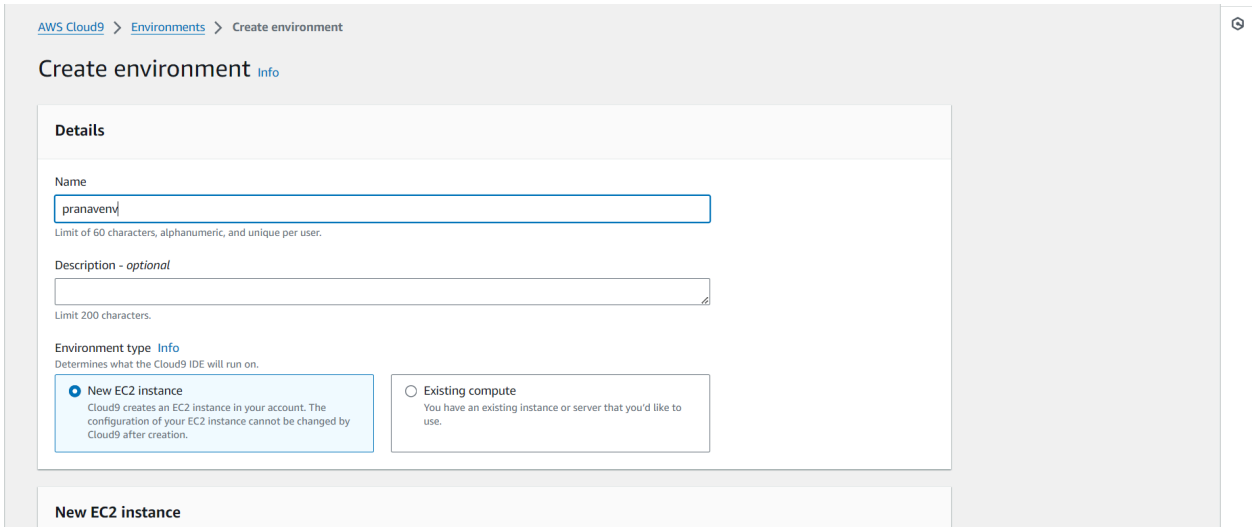
Step 2 : Install necessary Packages and run website on port 3000

Hey this is Dynamic Website.

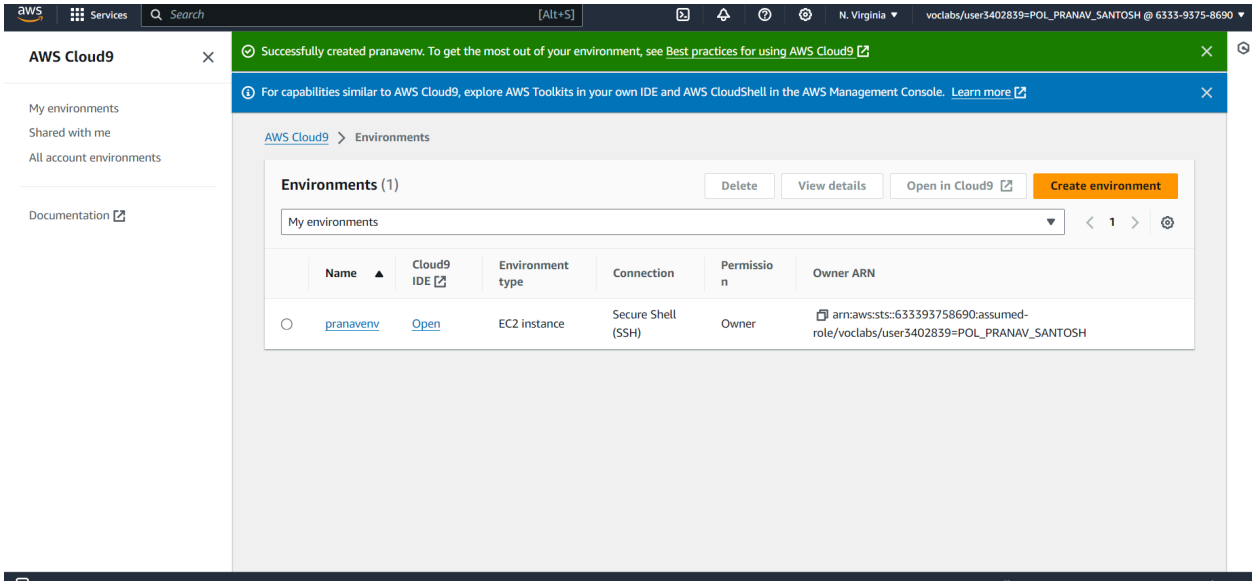
Hey this is about page.

Cloud 9 IDE Site Hosting

Step 1: Create Environment



Step 2 :Open the Environment IDE



Step 3: Add the code and preview the website