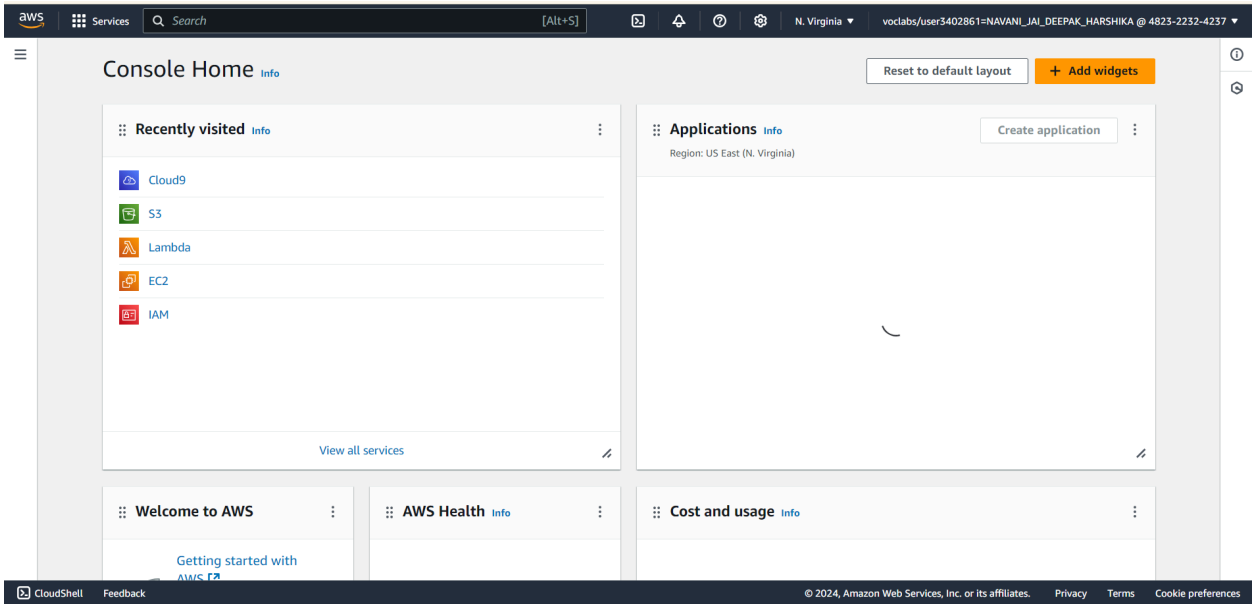
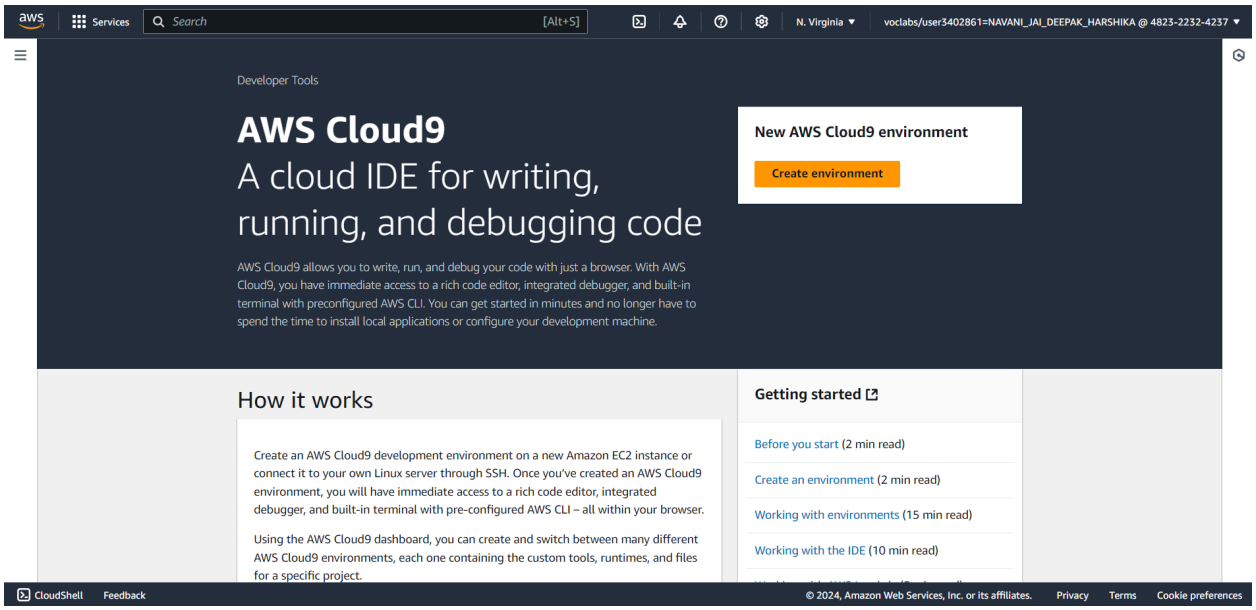
Name:Jai Deepak Navani
Rollno:31
Advance Deveops

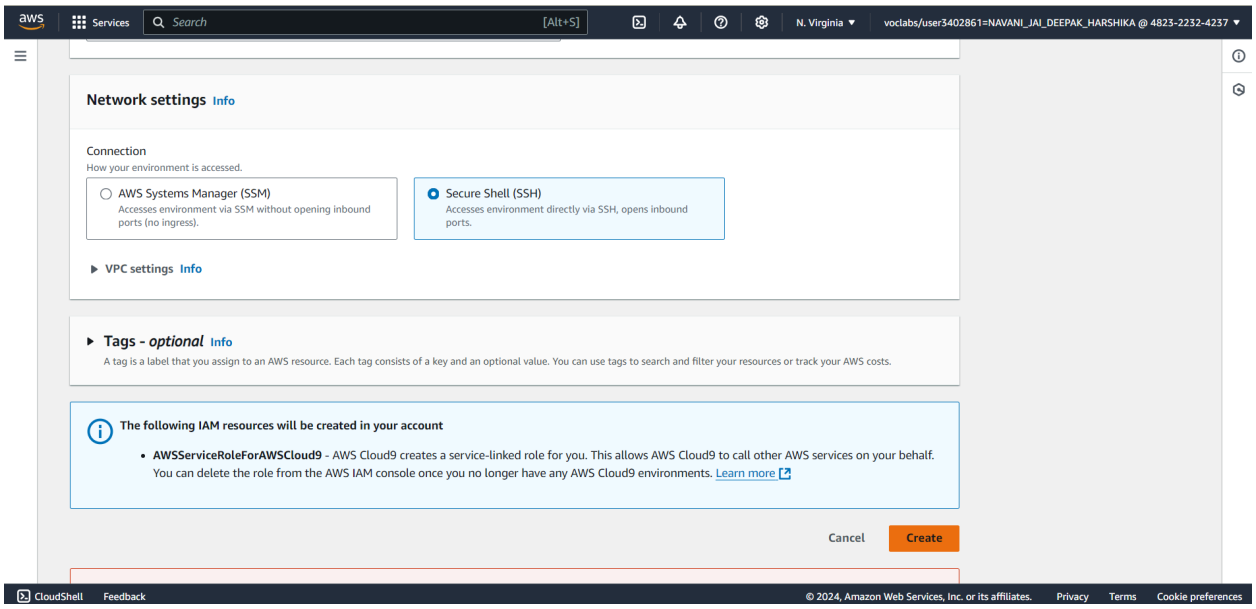
Study case(11):
Simple Cloud-Based Application Deployment
● Concepts Used: AWS Cloud9, S3, and EC2.
● Problem Statement: "Develop a simple HTML page using AWS Cloud9 and deploy it to an S3 bucket for static website hosting. Then, set up an EC2 instance to serve as a backup server for the website."
● Tasks:
○ Create a basic HTML page using AWS Cloud9.
○ Deploy the HTML page to an S3 bucket and enable static website hosting.
○ Launch an EC2 instance and configure it to serve the same HTML page as a backup.

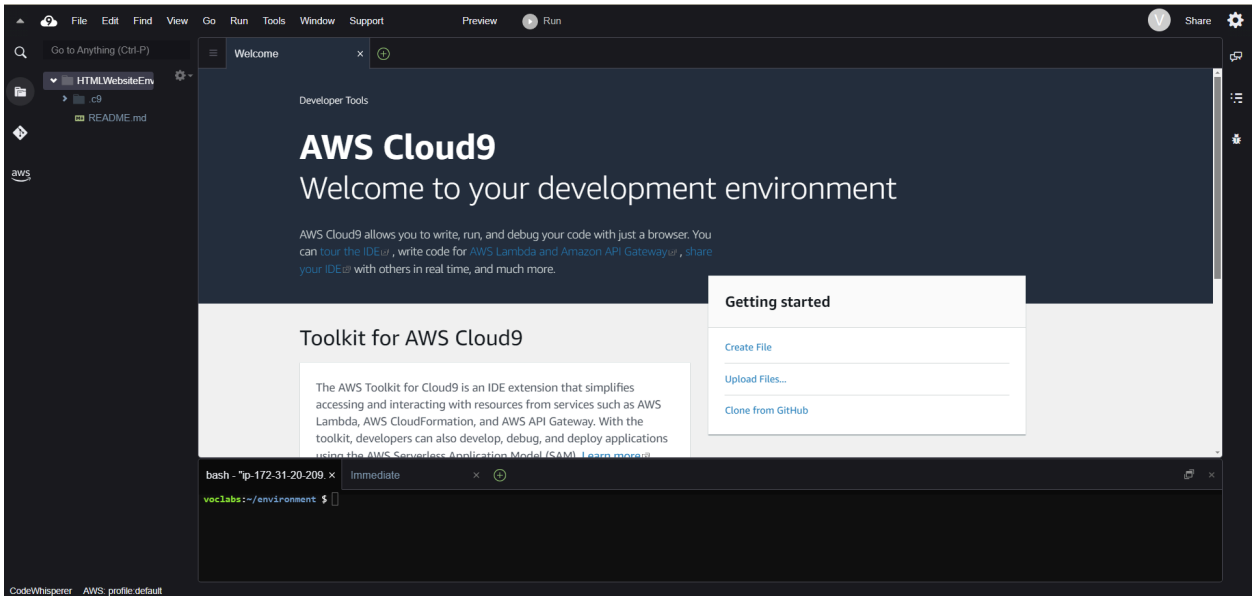
Sign-in to the console:



Search cloud9 and create environment:

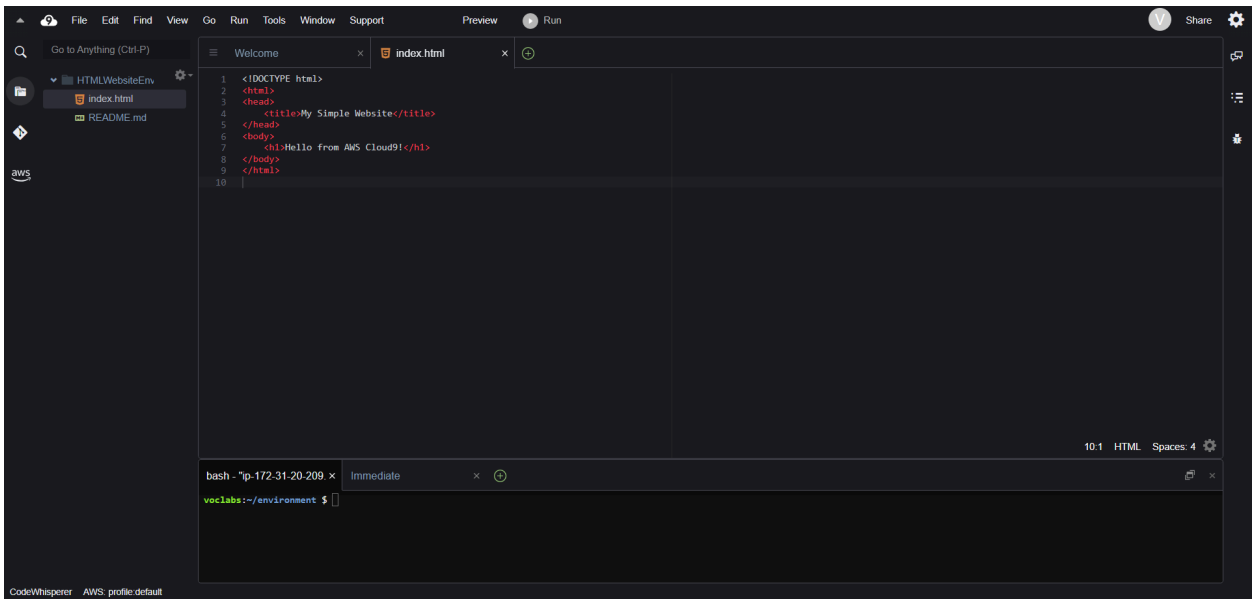

aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237
Developer Tools
AWS Cloud9
A cloud IDE for writing, running, and debugging code
AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.
New AWS Cloud9 environment
Create environment
How it works
Create an AWS Cloud9 development environment on a new Amazon EC2 instance or connect it to your own Linux server through SSH. Once you've created an AWS Cloud9 environment, you will have immediate access to a rich code editor, integrated debugger, and built-in terminal with pre-configured AWS CLI – all within your browser.
Using the AWS Cloud9 dashboard, you can create and switch between many different AWS Cloud9 environments, each one containing the custom tools, runtimes, and files for a specific project.
Getting started
Before you start (2 min read)
Create an environment (2 min read)
Working with environments (15 min read)
Working with the IDE (10 min read)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3402861=NAVANI_JAI_DEEPAK_HARSHIKA @ 4823-2232-4237
Network settings Info
Connection
How your environment is accessed.
AWS Systems Manager (SSM)
Accesses environment via SSM without opening inbound ports (no ingress).
Secure Shell (SSH)
Accesses environment directly via SSH, opens inbound ports.
VPC settings Info
Tags - optional Info
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
The following IAM resources will be created in your account
AWSServiceRoleForAWSCloud9 - AWS Cloud9 creates a service-linked role for you. This allows AWS Cloud9 to call other AWS services on your behalf. You can delete the role from the AWS IAM console once you no longer have any AWS Cloud9 environments. Learn more
Cancel
Create
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

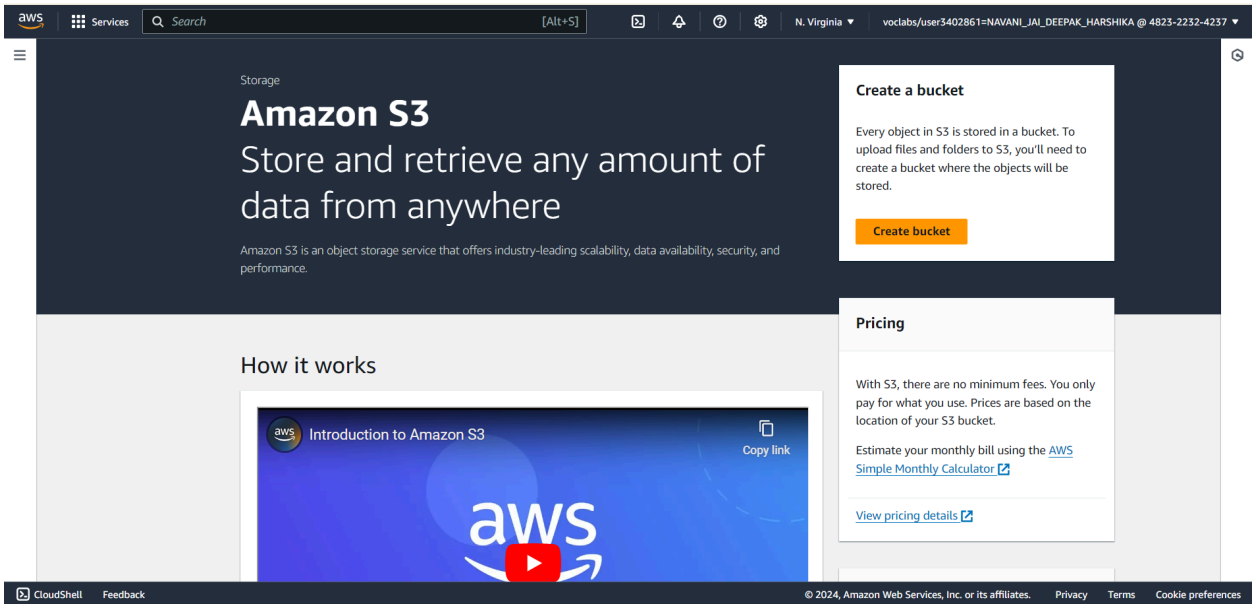


Make index.html file and save the code inside the file:

```
<!DOCTYPE html>
<html>
<head>
    <title>My Simple Website</title>
</head>
<body>
    <h1>Hello from AWS Cloud9!</h1>
</body>
</html>
```

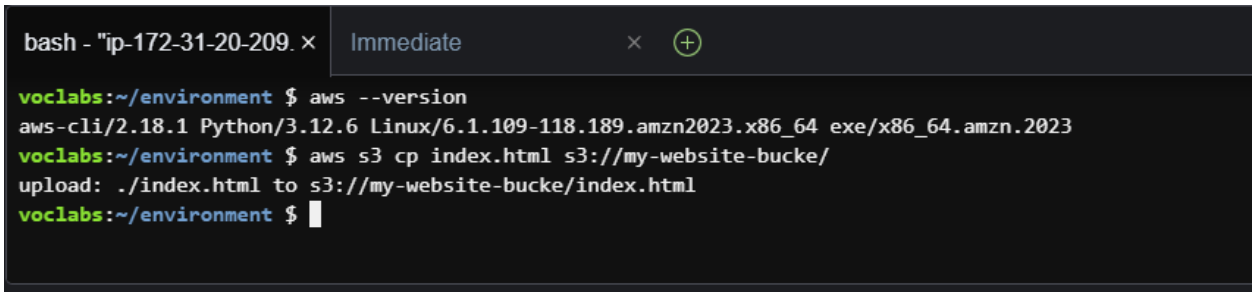


Now create s3 bucket:

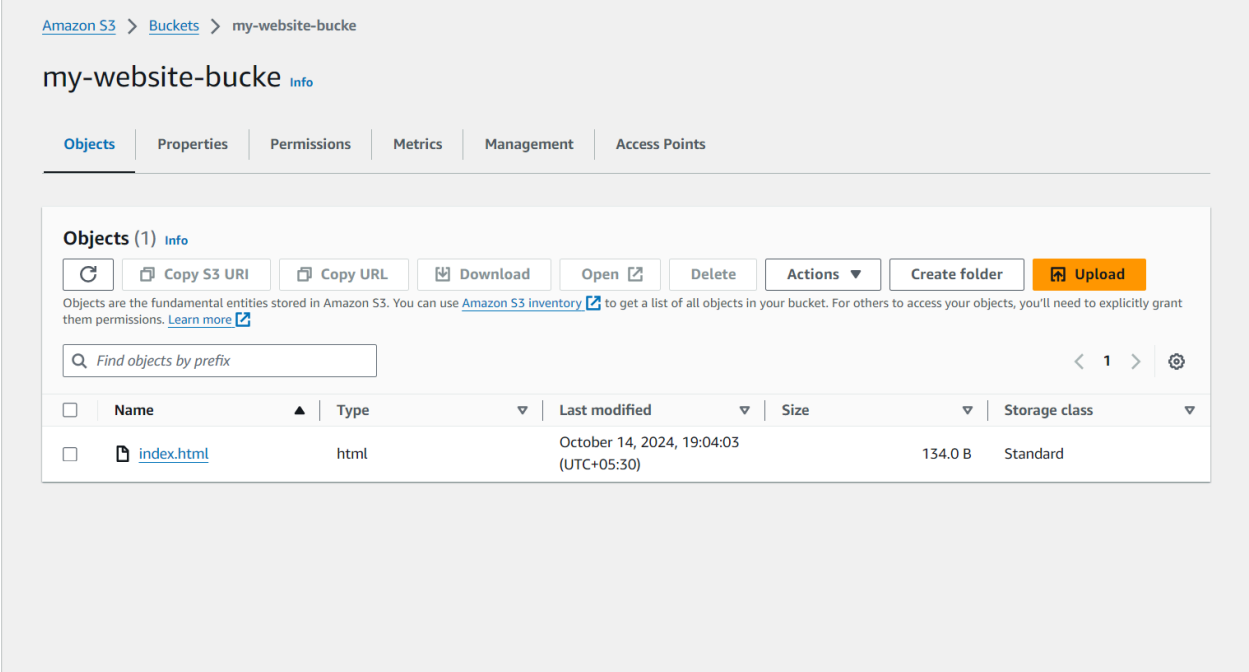


Now after creating the bucket upload the cloud9 file into the s3 bucket using simple commands:
aws --version

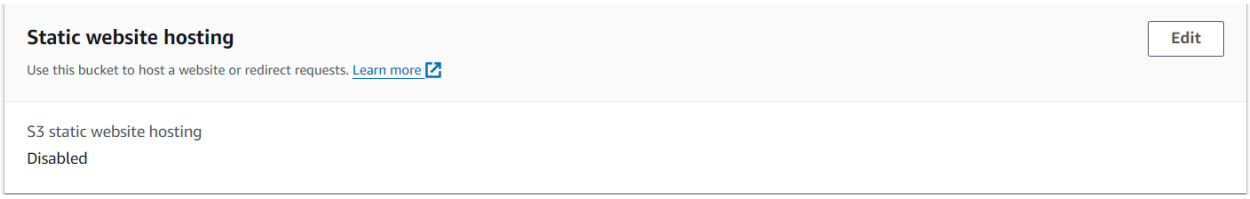
- aws --version
- aws s3 cp index.html s3://my-website-bucke/

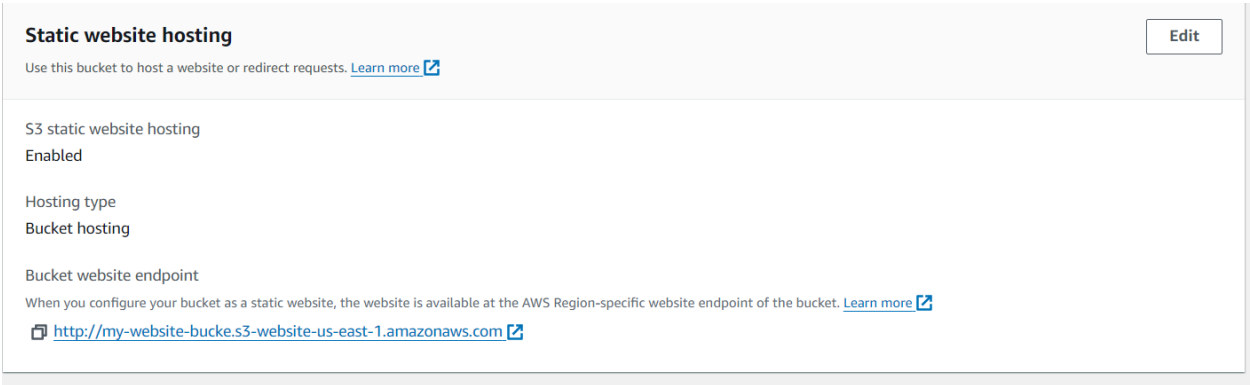


Check the uploaded file in s3 bucket:



Now go the properties of the bucket and edit static website enable from the disable:

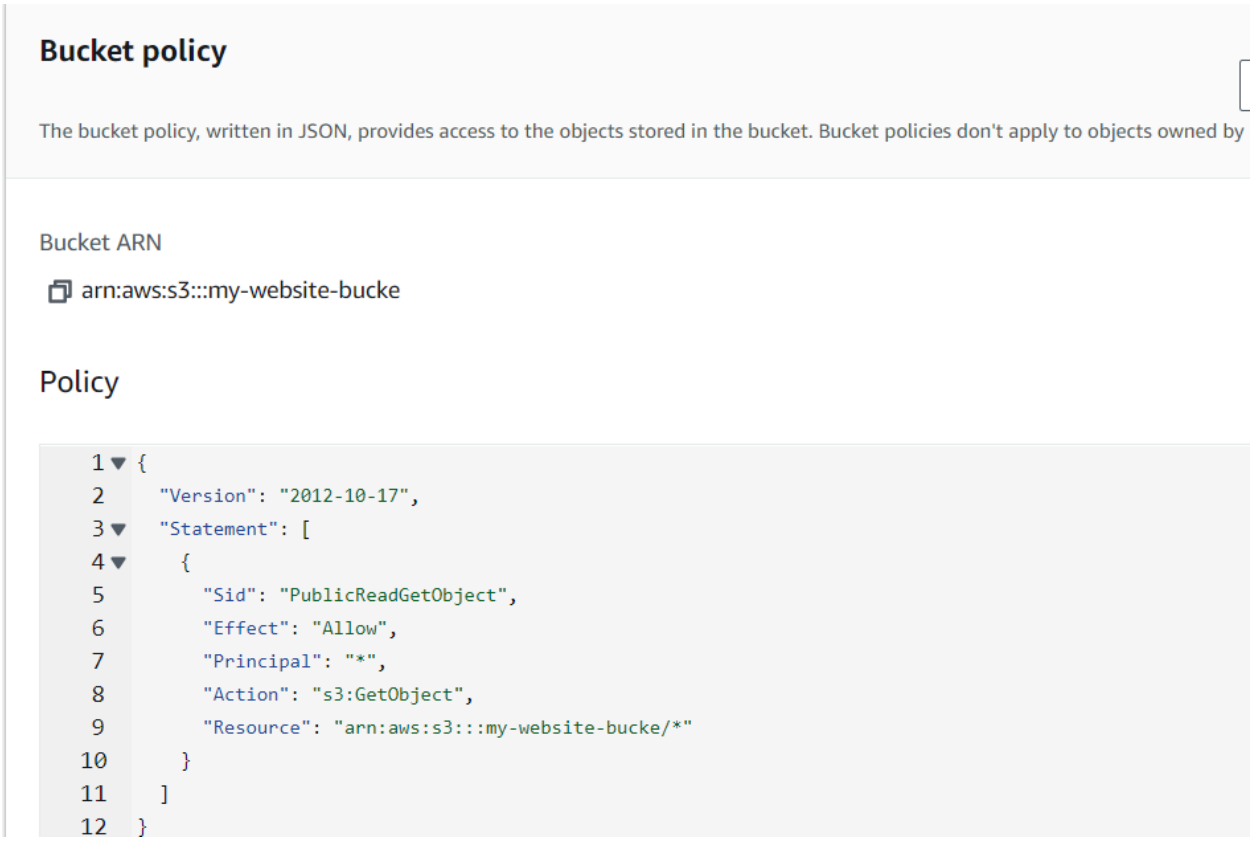




Go to the permission and write the bucket policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
```

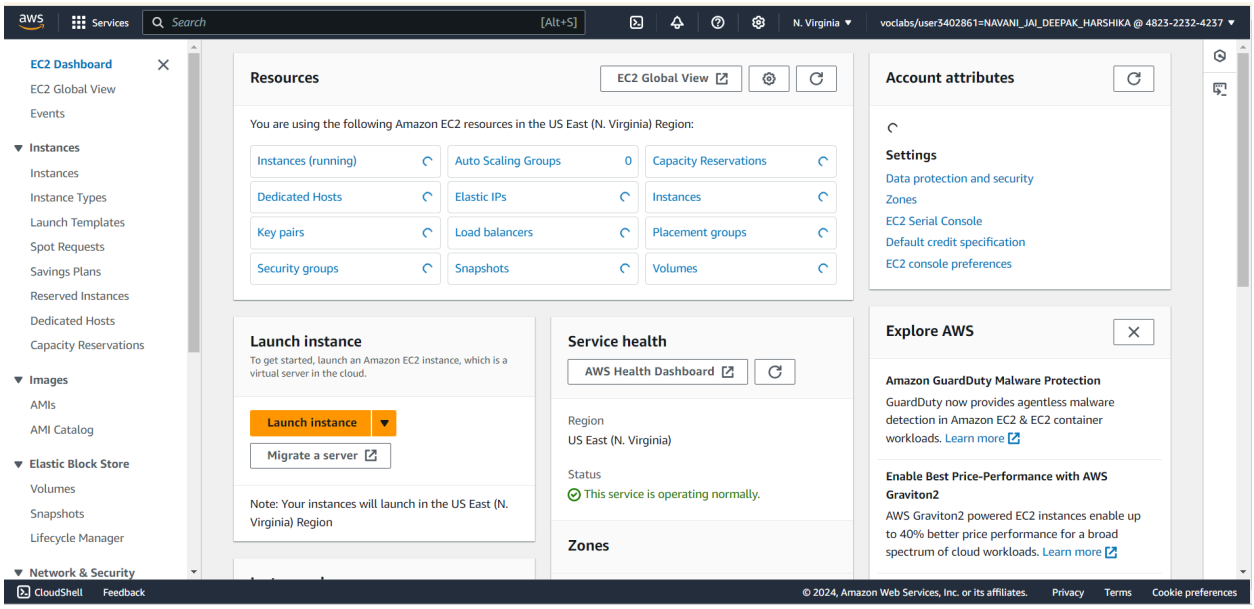
```
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::my-website-bucke/*"
  }
 ]
}
```

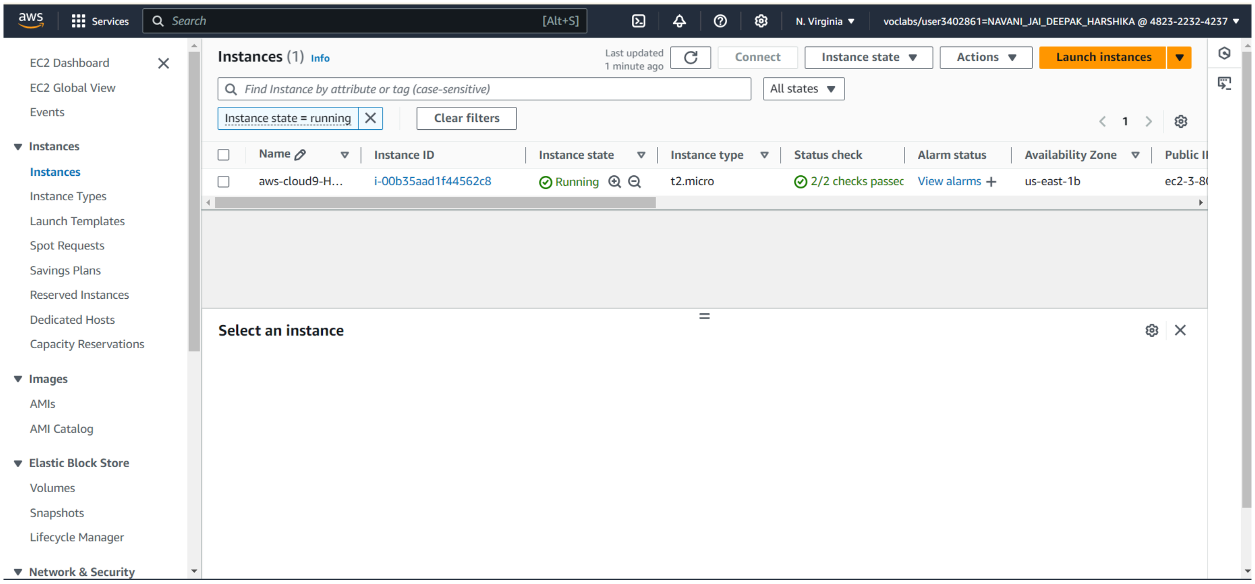


Now go the properties,then in the static hosting and see the url and check weather the html file is hosted or not:

Hello from AWS Cloud9!

Now launch the instance:





Now connect and add this commands:

sudo yum update -y

```
[ec2-user@ip-172-31-44-199 ~]$ sudo yum update -y
Last metadata expiration check: 0:01:04 ago on Mon Oct 14 13:53:59 2024.
Dependencies resolved.
Nothing to do.
Complete!
```

sudo yum install httpd -y

```
[ec2-user@ip-172-31-44-199 ~]$ sudo yum install httpd -y
Last metadata expiration check: 0:01:44 ago on Mon Oct 14 13:53:59 2024.
Dependencies resolved.
================================================================================
 Package               Architecture   Version                  Repository    Size
================================================================================
Installing:
 httpd                 x86_64         2.4.62-1.amzn2023        amazonlinux   48 k
Installing dependencies:
 apr                   x86_64         1.7.2-2.amzn2023.0.2     amazonlinux  129 k
 apr-util              x86_64         1.6.3-1.amzn2023.0.1     amazonlinux   98 k
 generic-logos-httpd   noarch         18.0.0-12.amzn2023.0.3   amazonlinux   19 k
 httpd-core            x86_64         2.4.62-1.amzn2023        amazonlinux  1.4 M
 httpd-filesystem      noarch         2.4.62-1.amzn2023        amazonlinux   14 k
 httpd-tools           x86_64         2.4.62-1.amzn2023        amazonlinux   81 k
 libbrotli             x86_64         1.0.9-4.amzn2023.0.2     amazonlinux  315 k
 mailcap               noarch         2.1.49-3.amzn2023.0.3    amazonlinux   33 k
Installing weak dependencies:
 apr-util-openssl      x86_64         1.6.3-1.amzn2023.0.1     amazonlinux   17 k
```

```
Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64                 apr-util-1.6.3-1.amzn2023.0.1.x86_64     apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.62-1.amzn2023.x86_64        httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch       httpd-tools-2.4.62-1.amzn2023.x86_64     libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch            mod_http2-2.0.27-1.amzn2023.0.3.x86_64   mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
```

sudo systemctl start httpd

```
[ec2-user@ip-172-31-44-199 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-44-199 ~]$
```

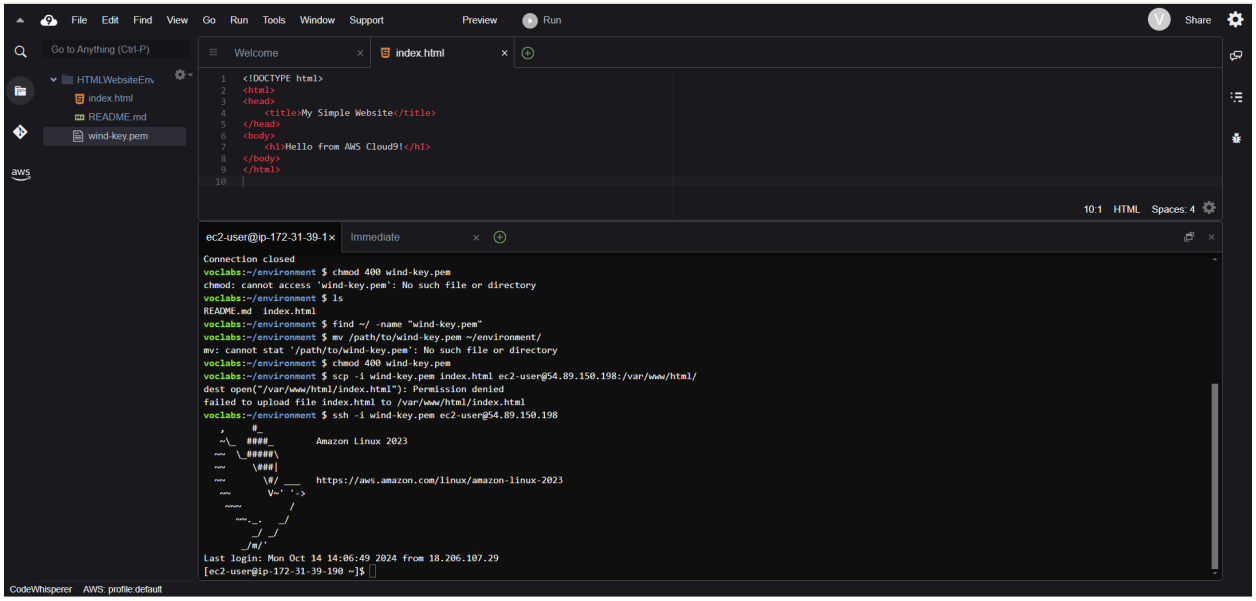
sudo systemctl enable httpd

```
[ec2-user@ip-172-31-44-199 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

Your EC2 instance now has an Apache web server running.
Now run this command in the cloud9 to connect the server to the ec2 instance:

scp -i wind-key.pem index.html aws@54.89.150.198:/var/www/html/

ssh -i wind-key.pem ec2-user@54.89.150.198



Now going to ec2 instance and clicking on the public IPV4 address we can see the same html page serve by the ec2instance:

Hello from AWS Cloud9!