



Static Website Hosting in AWS

Intern Project

Katikitala Sushma Rachel

AP19110010261

SRM University, AP

Acknowledgement

The opportunity I had with APSSDC in learning AWS was a great chance for learning and professional development. I consider myself as a lucky individual as I was provided with an opportunity to be a part of it. During this course I have done my utmost efforts and it would not be possible without the corporation of APSSDC

I would like to express my deepest gratitude to my project coordinator MR. Ramakrishna Cheeli, technical skill trainer at APSSDC to enable me to complete this project. I would also like to thank all the trainers that helped me in learning the concepts and guiding me throughout the course.

I will strive to use my knowledge and gained skills in the best possible way.

Project Description

Aim of the project is host a static website using the EC2 instance. For this project I have taken a sample website which is hosted publicly using the Amazons S3 service, so it can be accessed from anywhere in the world on different devices and web browsers. EC2 provides us with a virtual server with different capacities. It enables to scale up or down and handle traffic. The user has to rent the virtual server to deploy the applications.

The website is hosted using Amazon Machine Image (AMI) which has an operating system, Amazon Linux Server, Elastic IP, Security groups to control traffic, VPC to let us use multiple layers of security including network access control lists and security groups that help control access to EC2 instances each subnet.

AWS Services used

EC2

Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud for running applications on the AWS. It offers many options that enables us to build and run virtually any application. EC2 instance can be used to launch as many or as few virtual servers as needed, configure security and networking, and manage storage. It can particularly serve unlimited set of virtual machines (VMs). Amazon provides with different types of instances with different configurations of CPU, memory, storage and networking resources as per user's requirements.

EC2 allows to build applications to automate scaling according to changing needs and peak periods, makes it simple to deploy virtual to manage storage, lowering the need to invest in hardware and helping streamline development process.

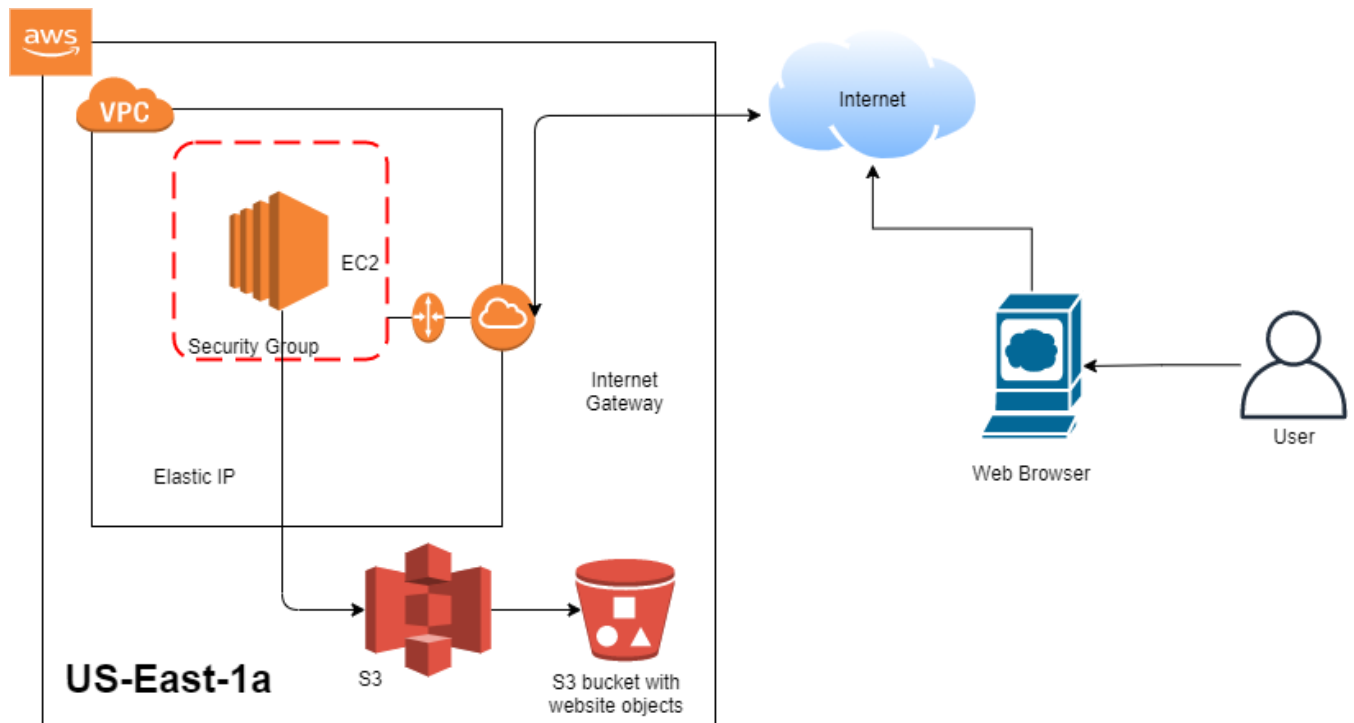
To host a website using EC2, we need to create and configure an EC2 instance in the AWS management console

Amazon S3

Amazon Simple Storage Service is a storage for the internet. It has a simple web service interface which is scalable and has a high-speed web passed cloud storage service, which is built for storing and recovering any amount of data from anywhere on the web. This service aims to maximize benefits of scale and to pass those benefits on to developers.

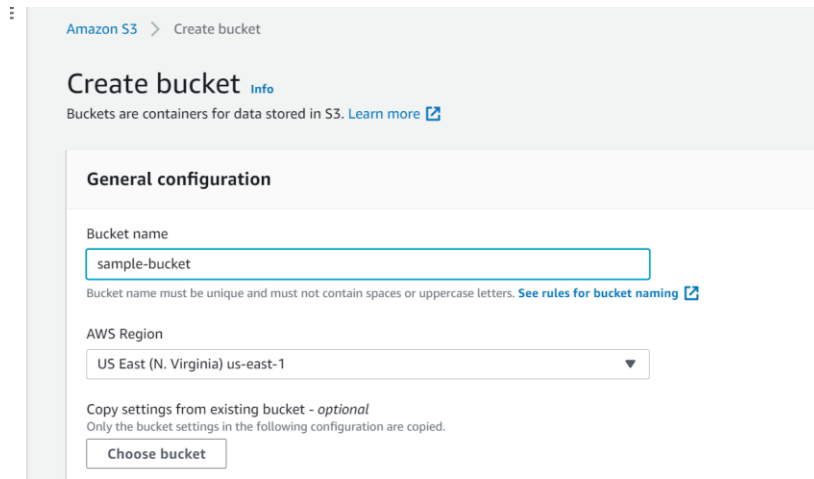
It can be used by small businesses or large enterprises. S3 service scalability, availability, security, and performance makes it suitable for a variety of data storage cases. Data storage, backup, archiving, hosting (images, videos, and music files), and website hosting are few common use cases.

Architecture



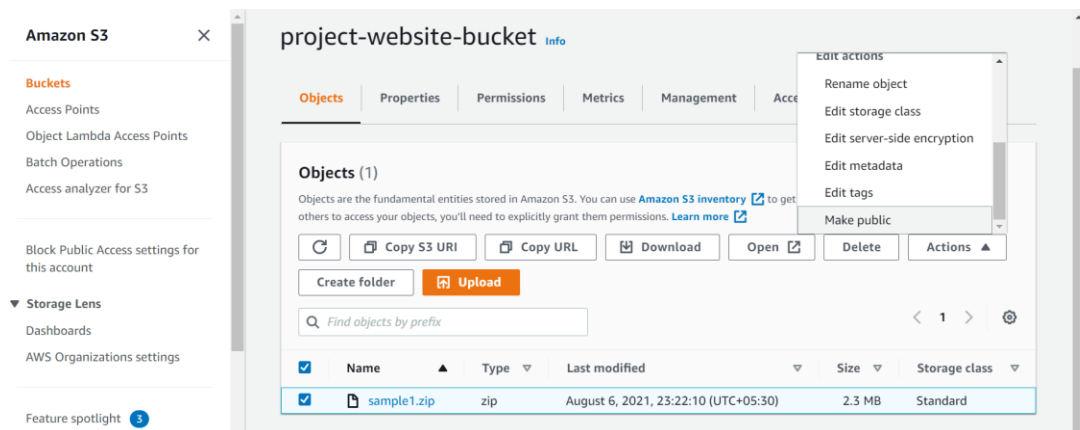
Procedure

1. Login to the AWS account and search for S3 service. Click on the create bucket option and create a bucket of any name. Choose a region to deploy the AWS resources.



The screenshot shows the 'Create bucket' page in the AWS Management Console. The breadcrumb navigation at the top reads 'Amazon S3 > Create bucket'. The main heading is 'Create bucket' with an 'Info' link. Below this, a sub-header reads 'Buckets are containers for data stored in S3. [Learn more](#)'. The 'General configuration' section contains a 'Bucket name' input field with the text 'sample-bucket'. A note below the field states: 'Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)'. The 'AWS Region' is set to 'US East (N. Virginia) us-east-1'. At the bottom, there is a section for 'Copy settings from existing bucket - optional' with a note: 'Only the bucket settings in the following configuration are copied.' and a 'Choose bucket' button.

2. Upload the .zip folder containing the website files into the bucket.

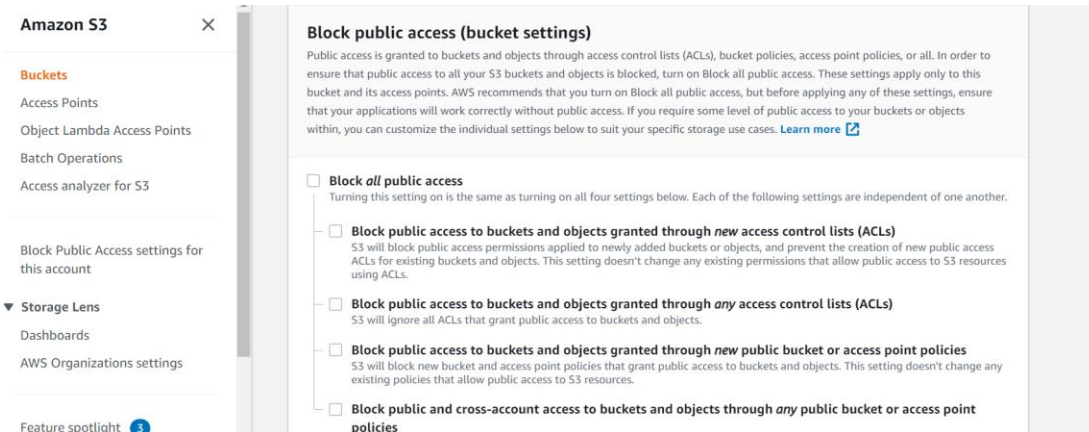


The screenshot shows the AWS S3 console interface. On the left is a sidebar with navigation options: 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Batch Operations', 'Access analyzer for S3', 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'AWS Organizations settings', and 'Feature spotlight'. The main panel displays the 'project-website-bucket' page. It has tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access'. The 'Objects' tab is active, showing 'Objects (1)'. A description states: 'Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)'. Below this are buttons for 'Refresh', 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', and 'Actions'. There are also buttons for 'Create folder' and 'Upload'. A search bar says 'Find objects by prefix'. Below the search bar is a table with one object:

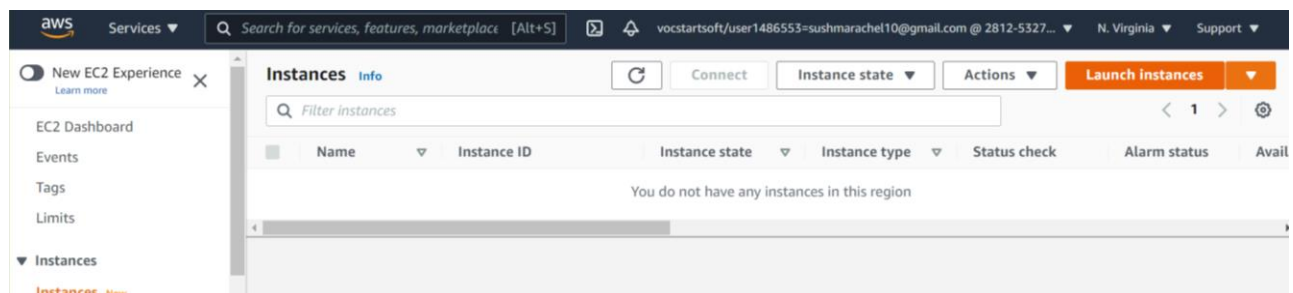
<input checked="" type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input checked="" type="checkbox"/>	sample1.zip	zip	August 6, 2021, 23:22:10 (UTC+05:30)	2.3 MB	Standard

An 'Edit actions' dropdown menu is open, showing options: 'Rename object', 'Edit storage class', 'Edit server-side encryption', 'Edit metadata', 'Edit tags', and 'Make public'.

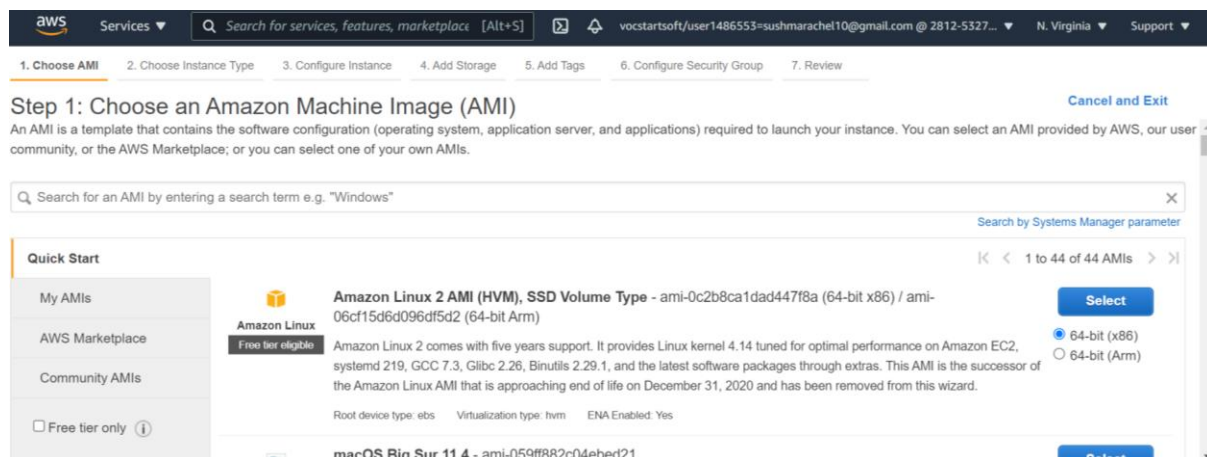
3. Go to bucket permissions and edit block access. Deselect block all public access. Then go to object actions and make the objects public.



4. Search for EC2 in AWS services. In EC2 dashboard select Instances and then select launch instances.



5. Select Amazon Linux 2 AMI SSD Volume Type of 64-bit (x86) [AMI]



6. Choose t2 micro as instance type. Leave the Configure Instance Details and add storage to default. Add a name tag and value. Then in Configure security group select create a new security group add rule HTTP and review and launch and then launch.

aws Services Search for services, features, marketplace [Alt+S] vocstartsoft/user1486553=sushmarachel10@gmail.com @ 2812-5327... N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	Network Interfaces
Name	sample	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

aws Services Search for services, features, marketplace [Alt+S] vocstartsoft/user1486553=sushmarachel10@gmail.com @ 2812-5327... N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:
 Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, :::0	e.g. SSH for Admin Desktop

[Add Rule](#)

[Cancel](#) [Previous](#) [Review and Launch](#)

- Choose an existing key pair or create a new key pair. If created a new keypair convert the downloaded PEM file to .ppk file using puttygen.exe.

Search for services, features, marketplace [Alt+S] vocstartsoft/user1486553=sushmarachel10@gmail.com @ 2812-5327...

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

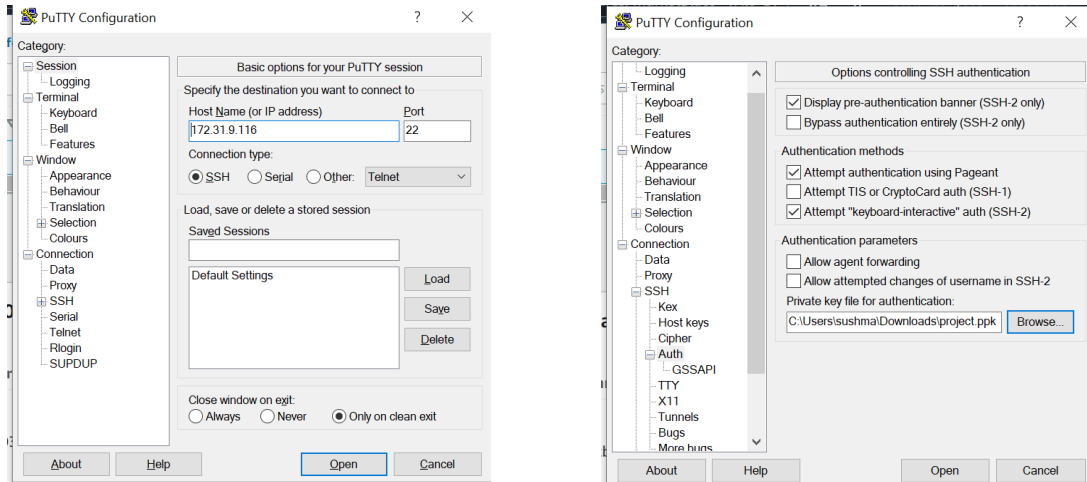
Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

Choose an existing key pair
 Select a key pair

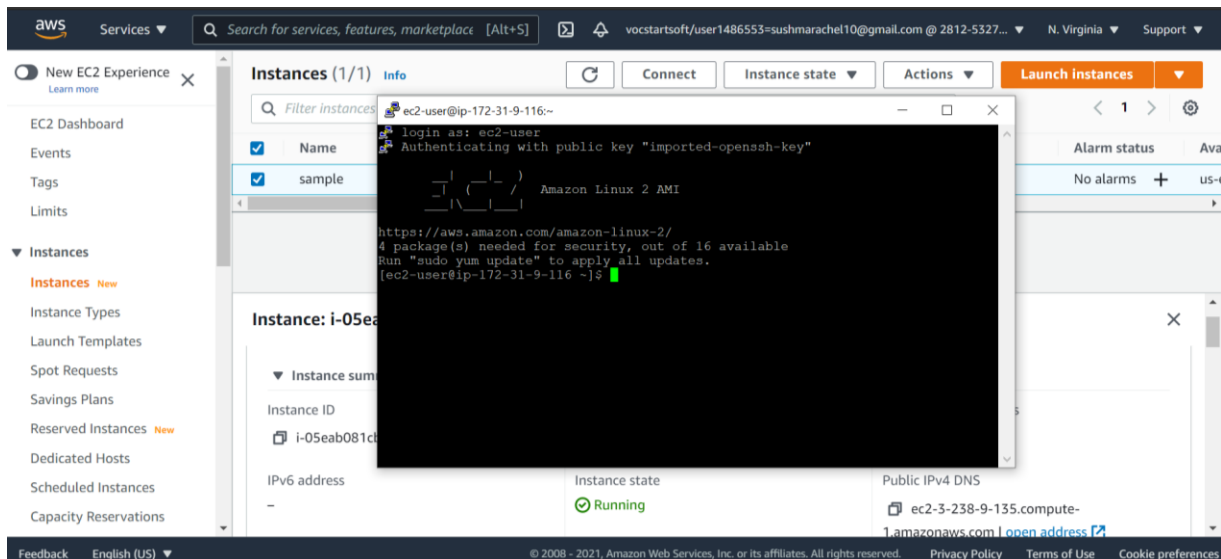
☒ I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

[Cancel](#) [Launch Instances](#)

8. Open Putty.exe. In Host Name paste the public IPv4 address that is copied from the created instance. In SSH go to Auth and then browse the .ppk file in the private key file for authentication section. Change Appearances if required and then open.



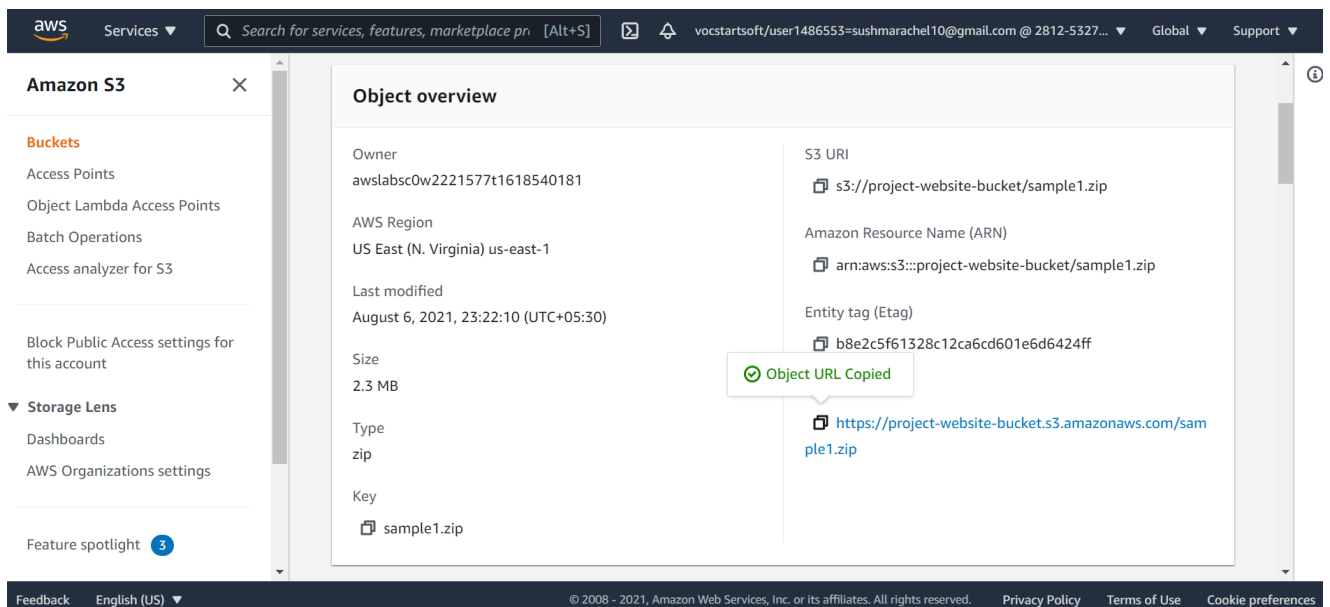
9. Then the command prompt appears on the screen. Login as ec2-user.

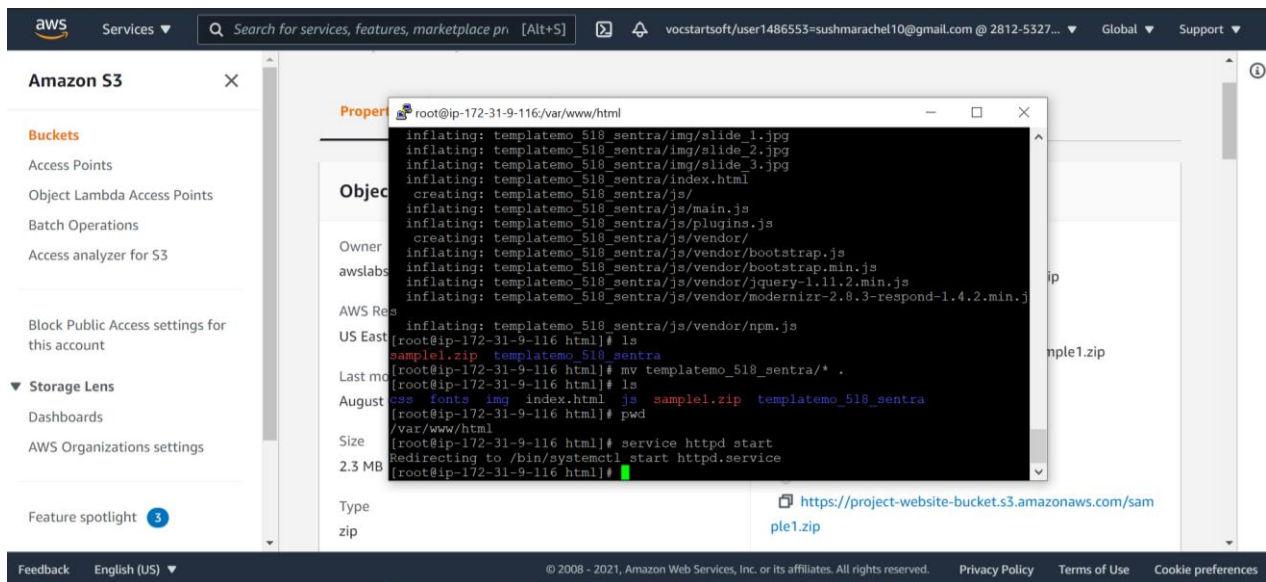


10. Then enter the following linux commands

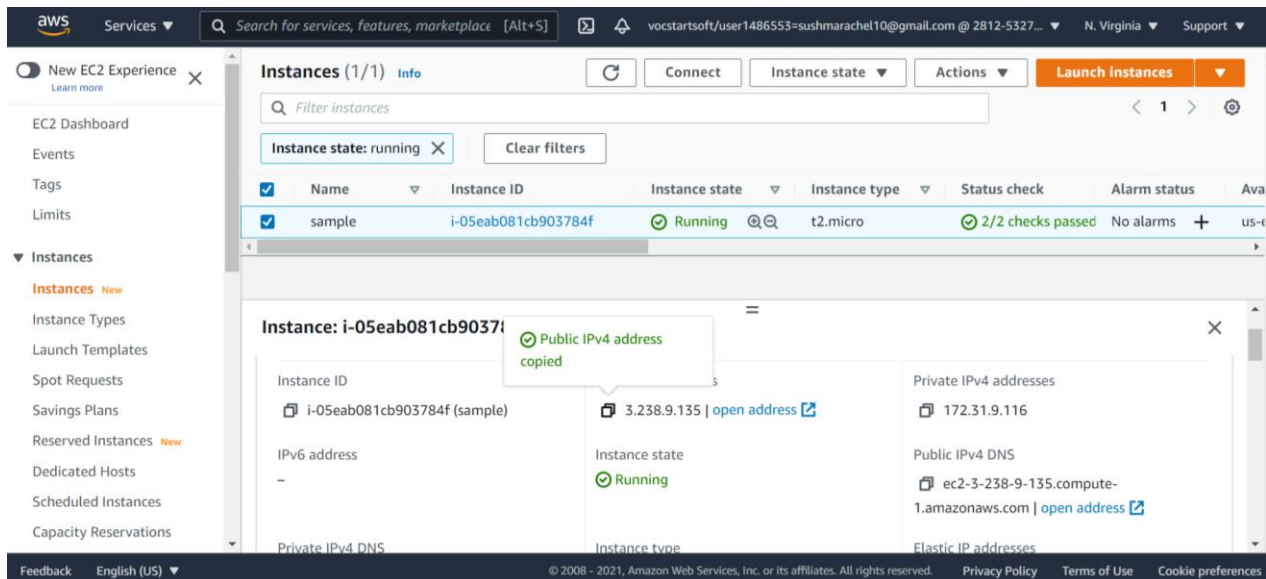
- `sudo su` it allows to run programs as another user
- `yum update -y` Updates packages to latest version
- `yum install httpd -y` Installs packages

- `pwd` to know the directory
- `cd /var/www/html` change to html directory
- `ls` list the files (we don't find any files)
- `wget Object_URL` Downloads files from that URL
- `ls` It shows the .zip file
- `unzip folder_name.zip` Unzip to access the files
- `ls` Now we find the unzipped folder
- `mv folder_name/* .` Move the unzipped folder
- `ls` All the files within the folder (html, css, etc,.)
- `pwd` Prints the working directory
- `service httpd start` To start our apache

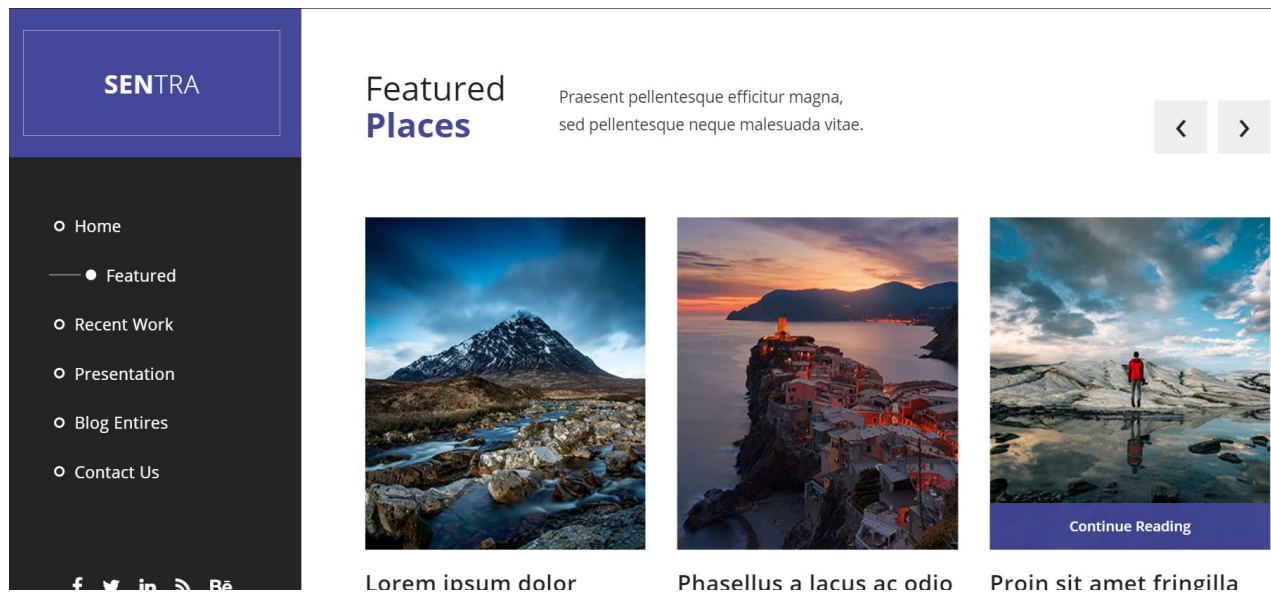
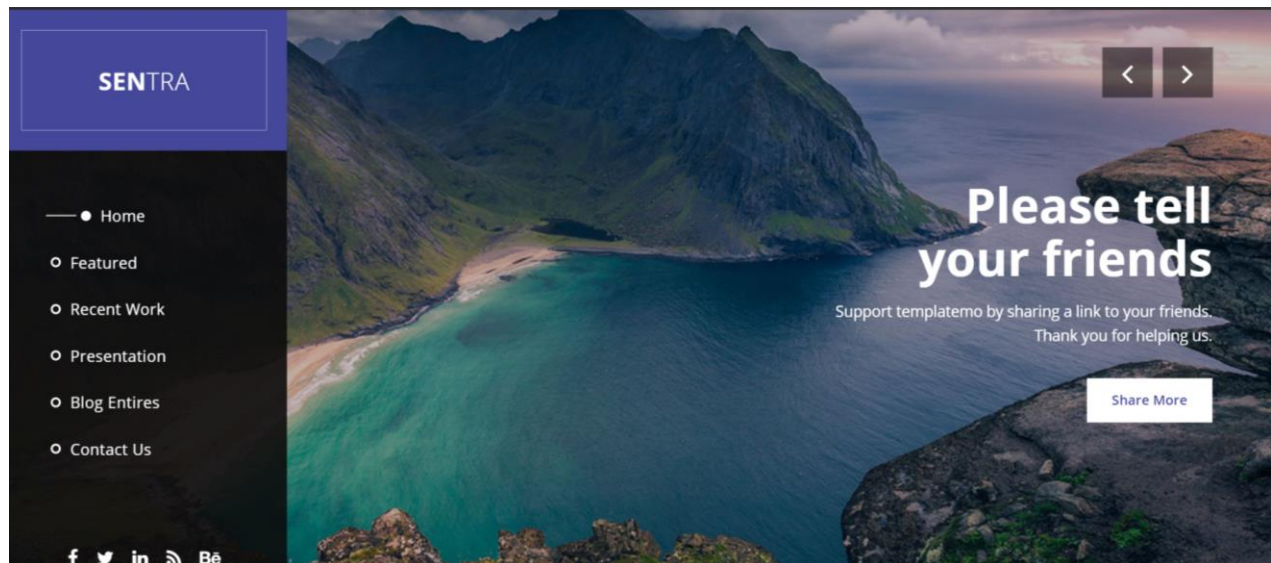


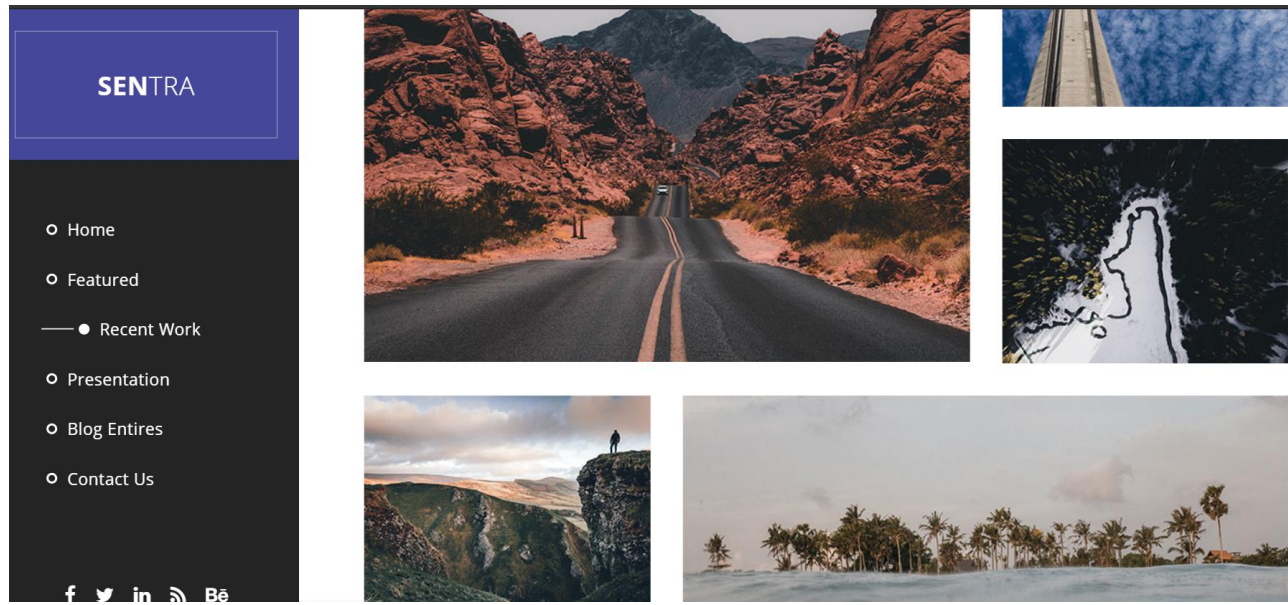


11. Then go to the EC2 instance, copy the Public IPv4 Address and paste it in a new tab.



Output





References

AWS account - <https://aws.amazon.com/education/awseducate/>

Sample Website - <https://www.free-css.com/free-css-templates/page258/sentra>