



SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

SRM Nagar, Kattankulathur – 603 203,

SCHOOL OF COMPUTING

DEPARTMENT OF NETWORKING & COMMUNICATIONS

COURSE PROJECT

TITLE: HOSTING A WEBSITE ON CLOUD

COURSE CODE: 18CSC310J

**COURSE NAME: DATA CENTRIC NETWORKING AND
SYSTEM DESIGN**

FACULTY : Dr.SENTHAMARAI

TEAM MEMBERS:

KOLLI LOKA PRASHANTH REDDY-RA2011028010111

ABSTRACT:

Cloud Computing is a paradigm in which data, applications or software are accessed over a network. This network of servers is called as "Cloud". Using a client such as desktops, entertainment centers, tablet computers, notebooks, wall computers, handhelds etc, users can reach into the cloud for resources as they need them. Cloud computing is on-demand access to virtualized IT resources that are housed outside of your own data center, shared by others, simple to use, paid for via subscription, and accessed over the Web. The main work in this project is to host Permit System as an application in the GoGrid cloud to analyze cloud services and architecture.

INDEX

No	Particulars	PgNo.
1.	Objective	3
2.	Methodology	3
3.	Approach	4
4.	Proof of GitHub upload	16
5.	Conclusion	16

TITLE : HOSTING A WEBSITE ON CLOUD

OBJECTIVE:

Objective of this project to demonstrate the Platform as a Service (PaaS) cloud service delivery model and to demonstrate the deployment of a website on cloud platform.

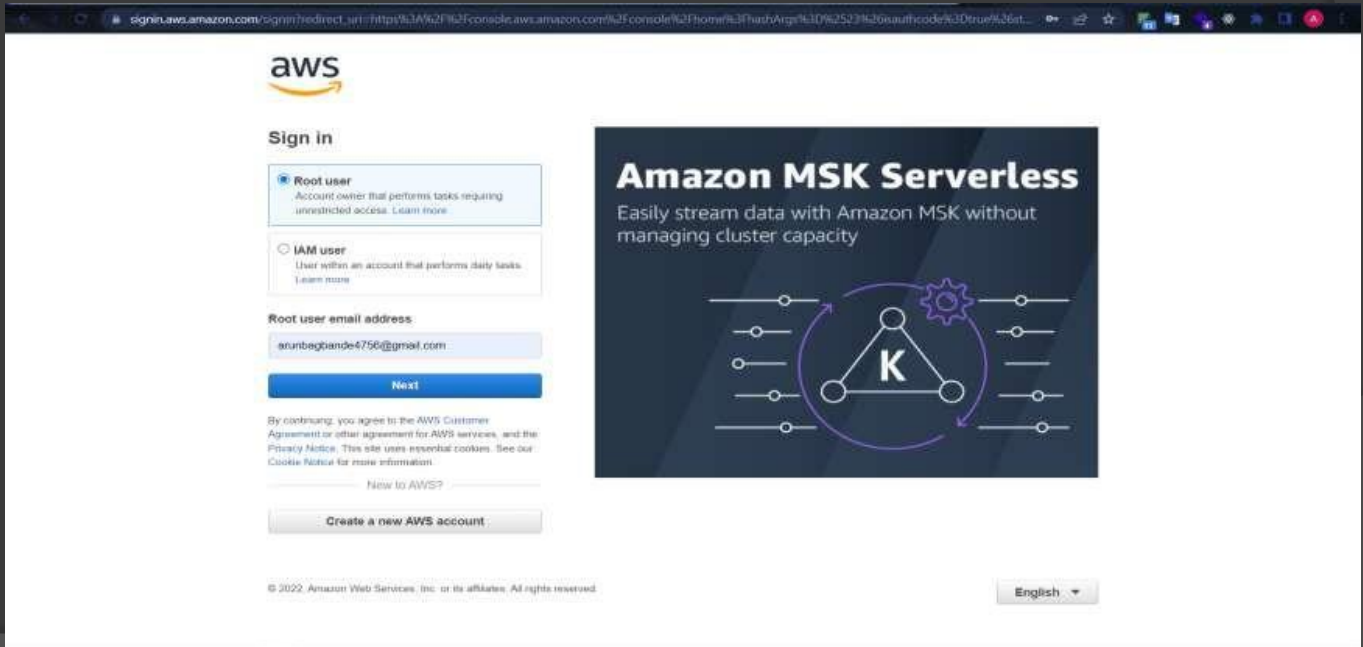
METHODOLOGY

What is Cloud Computing The term cloud referring to a network or the internet. It is a technology that uses remote servers on the internet to store, manage, and access data online rather than local drives. The data can be anything such as files, images, documents, audio, video, and more. There are the following operations that we can do using cloud computing:

- ☐ Developing new applications and services
- ☐ Storage, back up, and recovery of data
- ☐ Hosting blogs and websites
- ☐ Delivery of software on demand
- ☐ Analysis of data
- ☐ Streaming videos and audio

Approach

Step 1 - Create an AWS Account



The screenshot shows the AWS sign-in page in a web browser. The URL bar displays a long redirect URL. The page features the AWS logo at the top left. Below it, the 'Sign in' section offers two options: 'Root user' (selected) and 'IAM user'. The 'Root user' option is described as the 'Account owner that performs tasks requiring unrestricted access'. Below these options, there is a field for 'Root user email address' containing 'anunbagbande4756@gmail.com' and a blue 'Next' button. A disclaimer states: 'By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.' Below the disclaimer, there is a link 'New to AWS?' and a button 'Create a new AWS account'. On the right side of the page, there is a promotional banner for 'Amazon MSK Serverless' with the text 'Easily stream data with Amazon MSK without managing cluster capacity' and a diagram showing a cluster of nodes with a 'K' in the center. At the bottom left, the copyright notice reads '© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.' and at the bottom right, there is a language selector set to 'English'.

aws

Sign in

☒ **Root user**
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

☐ **IAM user**
User within an account that performs daily tasks. [Learn more](#)

Root user email address:
anunbagbande4756@gmail.com

Next

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.


[New to AWS?](#)

Create a new AWS account

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.

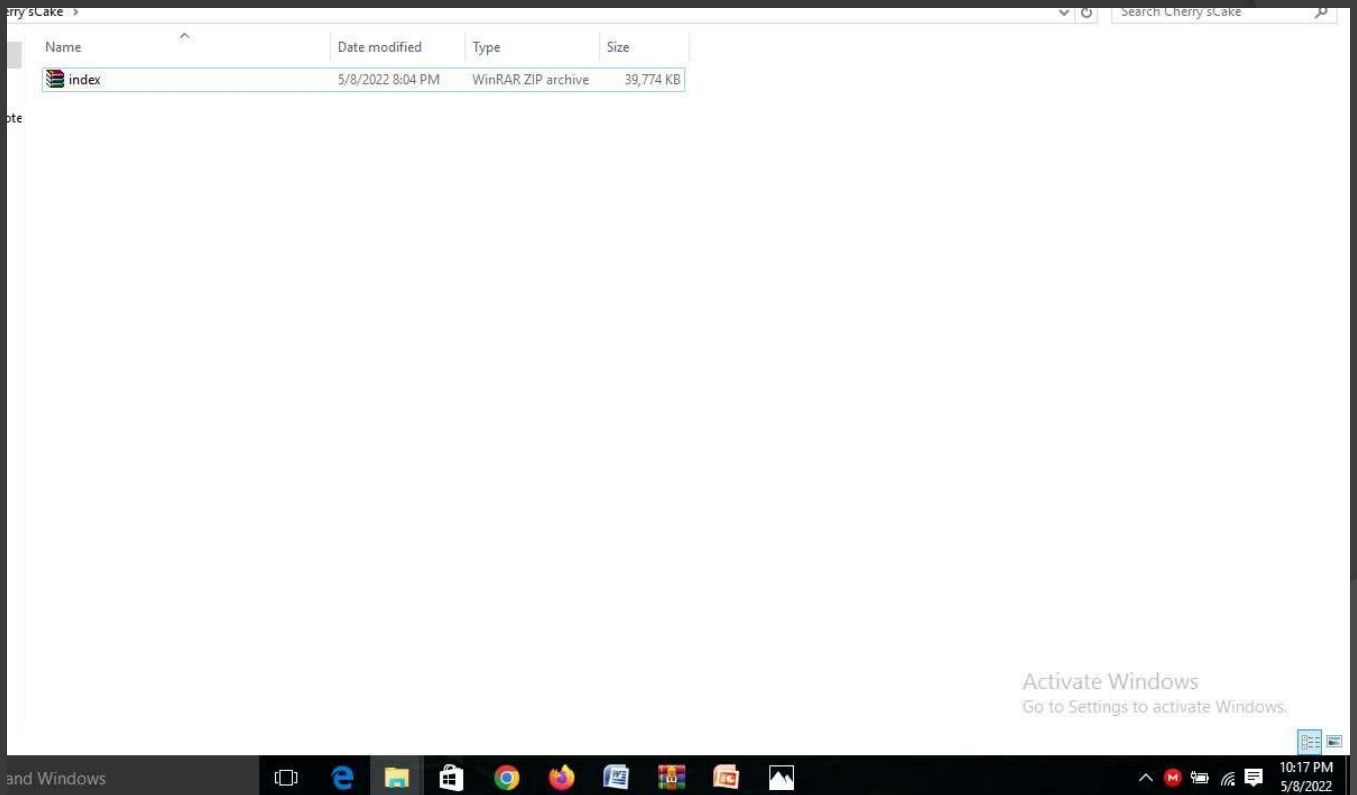
English

Amazon MSK Serverless
Easily stream data with Amazon MSK without managing cluster capacity

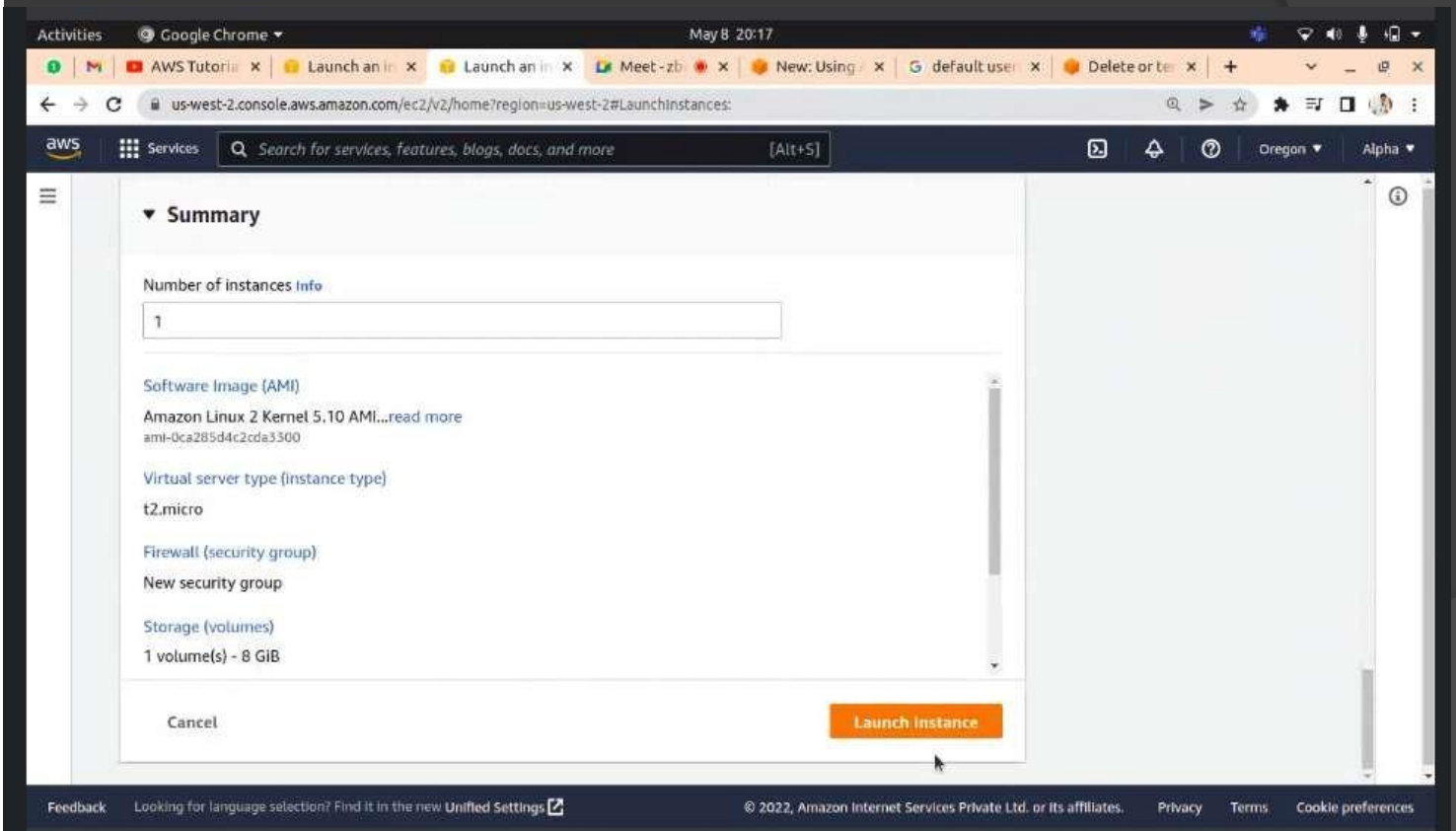


The diagram for Amazon MSK Serverless shows a central node labeled 'K' connected to three other nodes in a triangular arrangement. This central cluster is surrounded by a circular flow of data, represented by arrows. The entire diagram is set against a dark blue background with a subtle grid pattern.


Step 2 - Move all the website files in one folder and zip it.



Step 3 – Create an EC2 instance



EC2 > Instances > Launch an instance

**Success**
Successfully initiated launch of instance (i-0e42026b6195e6294)

► Launch log

Next Steps

Get notified of estimated charges

Create [billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)

How to connect to your instance

Your instance is launching and it might be a few minutes until it is in the running state, when it will be ready for you to use

Click [View Instances](#) to monitor your instance's status. Once your instance is in the 'running' state, you can connect to it from the Instances screen. Find out [how to connect to your instance](#)

Activities Google Chrome May 8, 2018

us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#instances:instanceId=i-0e42026b6195e6294

aws Services Search for services, features, blogs, docs, and more [Alt+S] Oregon Alpha

New EC2 Experience Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

▼ Instances

instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations

Instances (1) Info Refresh Connect Instance state ▼ Actions ▼ Launch Instances

Search

Instance ID = i-0e42026b6195e6294 × Clear filters

<input type="checkbox"/>	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status
<input type="checkbox"/>	Cloud Comput...	i-0e42026b6195e6294	⌚ Pending 🔍	t2.micro	—	No alarms +

Select an instance ⚙️ ×

Feedback Looking for language selection? Find it in the new Unified Settings [↗](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Activities

Google Chrome

May 8 20:19

us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#instanceDetails:instanceId=i-0e42026b6195e6294

aws Services Search for services, features, blogs, docs, and more [Alt+S]

New EC2 Experience
Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

EC2 > Instances > i-0e42026b6195e6294

Instance summary for i-0e42026b6195e6294 (Cloud Computing) Info

Updated less than a minute ago

Refresh

Connect

Inst

Public IPv4 address copied

Instance ID

i-0e42026b6195e6294 (Cloud Computing)

IPv6 address

-

Hostname type

IP name: ip-172-31-21-216.us-west-2.compute.internal

Instance type

t2.micro

Instance state

Pending

Private IP DNS name (IPv4 only)

ip-172-31-21-216.us-west-2.compute.internal

Elastic IP addresses

-

Private IPv4 addresses

172.31.21.216

Public IPv4 DNS

ec2-34-222-160-128.us-west-2.compute.amazonaws.com | open address

Answer private resource DNS name

IPv4 (A)

Auto-assigned IP address

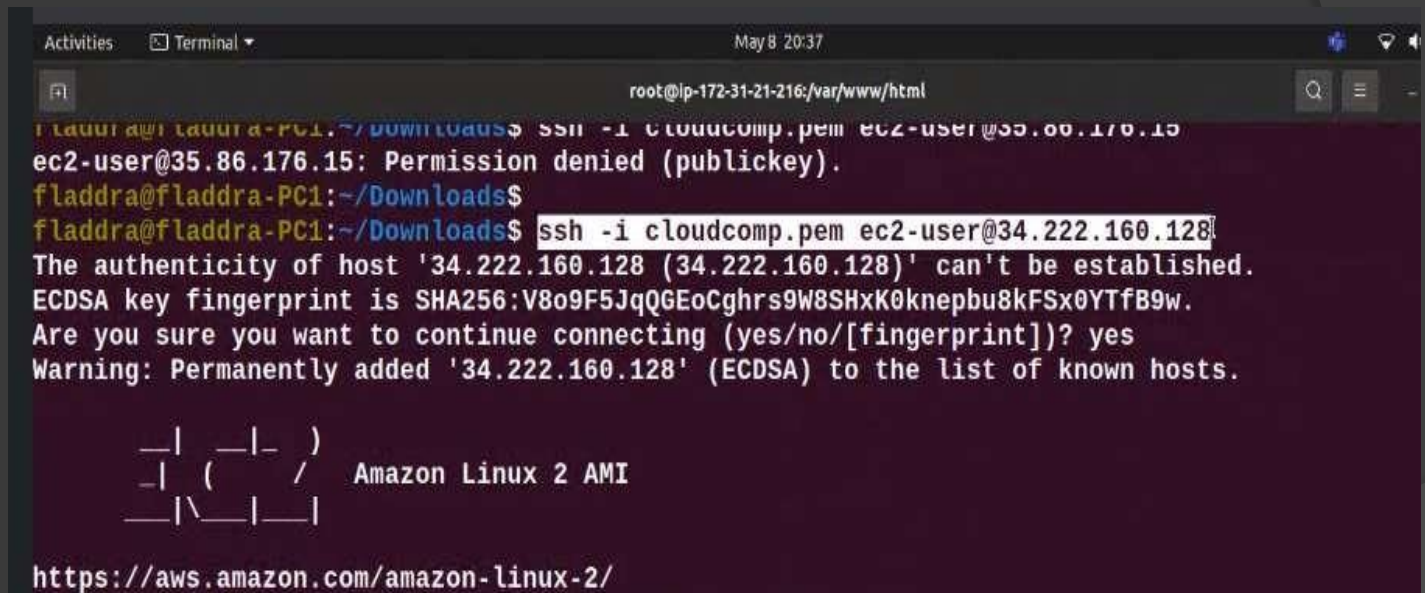
34.222.160.128 [Public IP]

Waiting for pinpoint.us-east-1.amazonaws.com...

Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Step 4 – instance into SSH



A terminal window titled 'Terminal' with a timestamp of 'May 8 20:37'. The prompt is 'root@ip-172-31-21-216:/var/www/html'. The user runs the command `ssh -i cloudcomp.pem ec2-user@35.86.176.15`, which results in 'Permission denied (publickey)'. The user then runs `ssh -i cloudcomp.pem ec2-user@34.222.160.128`. The terminal displays a warning about the host's authenticity, showing the ECDSA key fingerprint: 'SHA256:V8o9F5JqQGEoCghrs9W8SHxK0knepbu8kFSx0YTfB9w'. The user responds 'yes' to continue. A warning message states: 'Warning: Permanently added '34.222.160.128' (ECDSA) to the list of known hosts.' Below this, the ASCII art logo for Amazon Linux 2 is shown, followed by the URL <https://aws.amazon.com/amazon-linux-2/>.

```
Activities Terminal May 8 20:37
root@ip-172-31-21-216:/var/www/html
fladdra@fladdra-PC1:~/Downloads$ ssh -i cloudcomp.pem ec2-user@35.86.176.15
ec2-user@35.86.176.15: Permission denied (publickey).
fladdra@fladdra-PC1:~/Downloads$
fladdra@fladdra-PC1:~/Downloads$ ssh -i cloudcomp.pem ec2-user@34.222.160.128
The authenticity of host '34.222.160.128 (34.222.160.128)' can't be established.
ECDSA key fingerprint is SHA256:V8o9F5JqQGEoCghrs9W8SHxK0knepbu8kFSx0YTfB9w.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.222.160.128' (ECDSA) to the list of known hosts.

 _|_ _|_ )
 _| ( / Amazon Linux 2 AMI
 _|\_|_|_|

https://aws.amazon.com/amazon-linux-2/
```

Step 5 – Download the project files

[illegible]

```
[ec2-user@ip-172-31-21-216 ~]$ ls
[ec2-user@ip-172-31-21-216 ~]$ cd ..
[ec2-user@ip-172-31-21-216 home]$ ls
ec2-user
[ec2-user@ip-172-31-21-216 home]$ cd ec2-user/
[ec2-user@ip-172-31-21-216 ~]$ ls
[ec2-user@ip-172-31-21-216 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-21-216 ~]$ ls
index.zip
[ec2-user@ip-172-31-21-216 ~]$ unzip index.zip
```

```
[root@ip-172-31-21-216 ec2-user]# mv index.zip /var/www/html/
```

```
[root@ip-172-31-21-216 ec2-user]# ls
```

```
images  index.html  new.html  Saniyas  cake  scripts  styles
```

```
[root@ip-172-31-21-216 ec2-user]# cd /var/www/html/
```

```
[root@ip-172-31-21-216 html]#
```

```
[root@ip-172-31-21-216 html]# ls
```

```
index.zip
```

```
[root@ip-172-31-21-216 html]# unzip index.zip
```

```
Archive:  index.zip
```

```
  inflating: images/1logo-banner.png
```

```
  inflating: images/about-banner.jpg
```

```
  inflating: images/back.jpg
```

```
  inflating: images/cake-bn/1.png
```

```
  inflating: images/cake-bn/2.png
```

```
  inflating: images/cake-bn/3.png
```

```
  inflating: images/cake-img/1.png
```

```
  inflating: images/cake-img/2.png
```

```
  inflating: images/cake-img/3.png
```

```
  inflating: images/cake-img/4.png
```

```
  inflating: images/cake-img/5.png
```

```
  inflating: images/cake-img/6.png
```

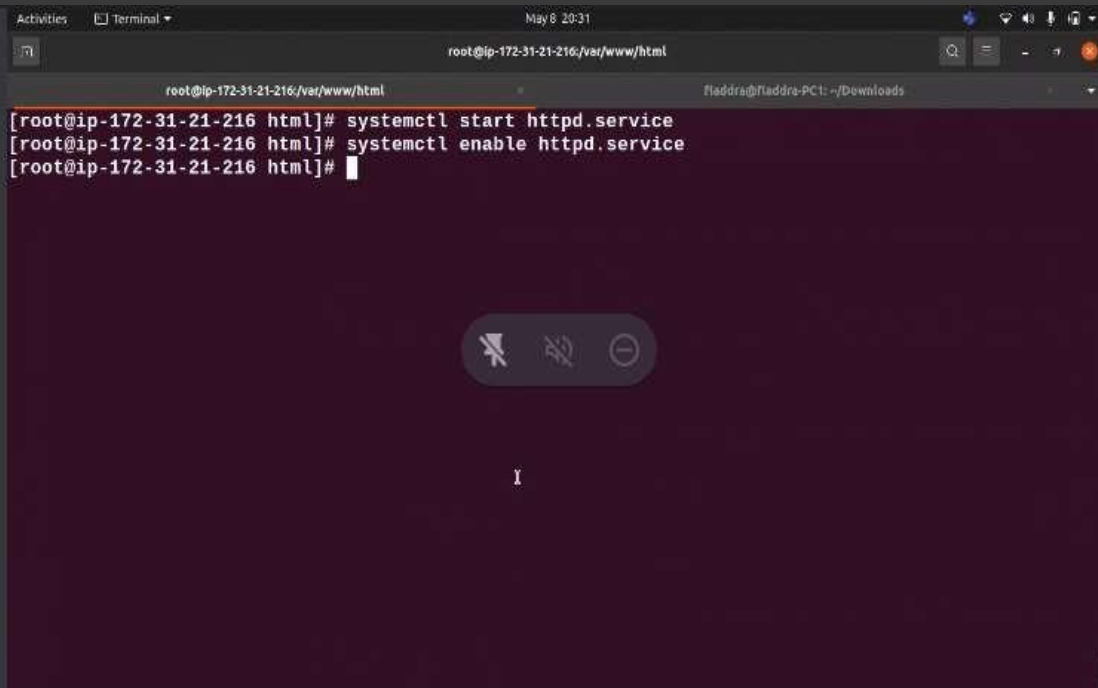
```
  inflating: images/cake-img/m-1.png
```

```
  inflating: images/cake-img/m-2.png
```

```
  inflating: images/intro.jpg
```

```
  inflating: images/logo.png
```

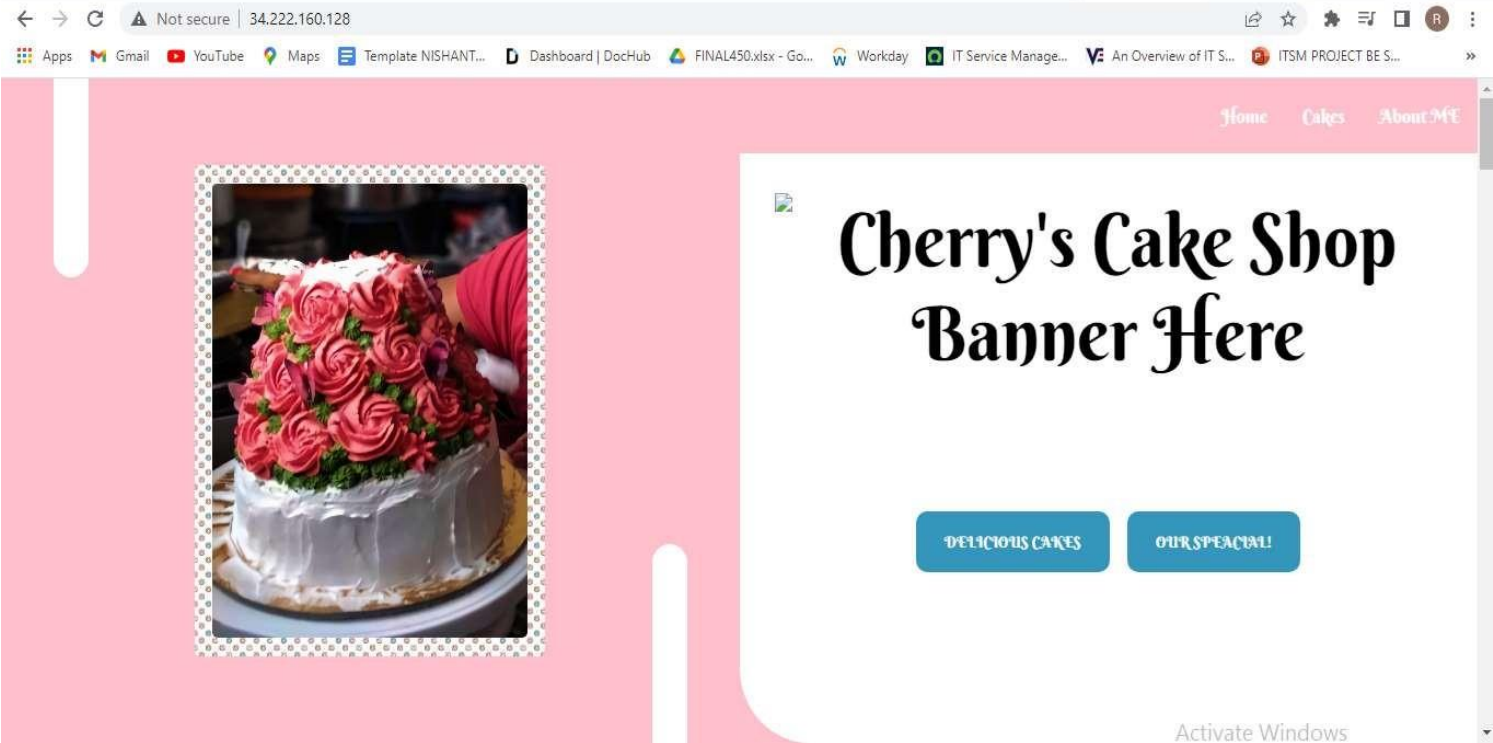
Step 6 – Enable the service



A terminal window titled "Terminal" with a timestamp of "May 8 20:31". The window shows a root user at an IP address of 172-31-21-216, currently in the directory /var/www/html. The user has entered two commands: "systemctl start httpd.service" and "systemctl enable httpd.service". The prompt is now ready for the next command. A floating window with three icons (a square, a circle, and a triangle) is visible in the center of the terminal.

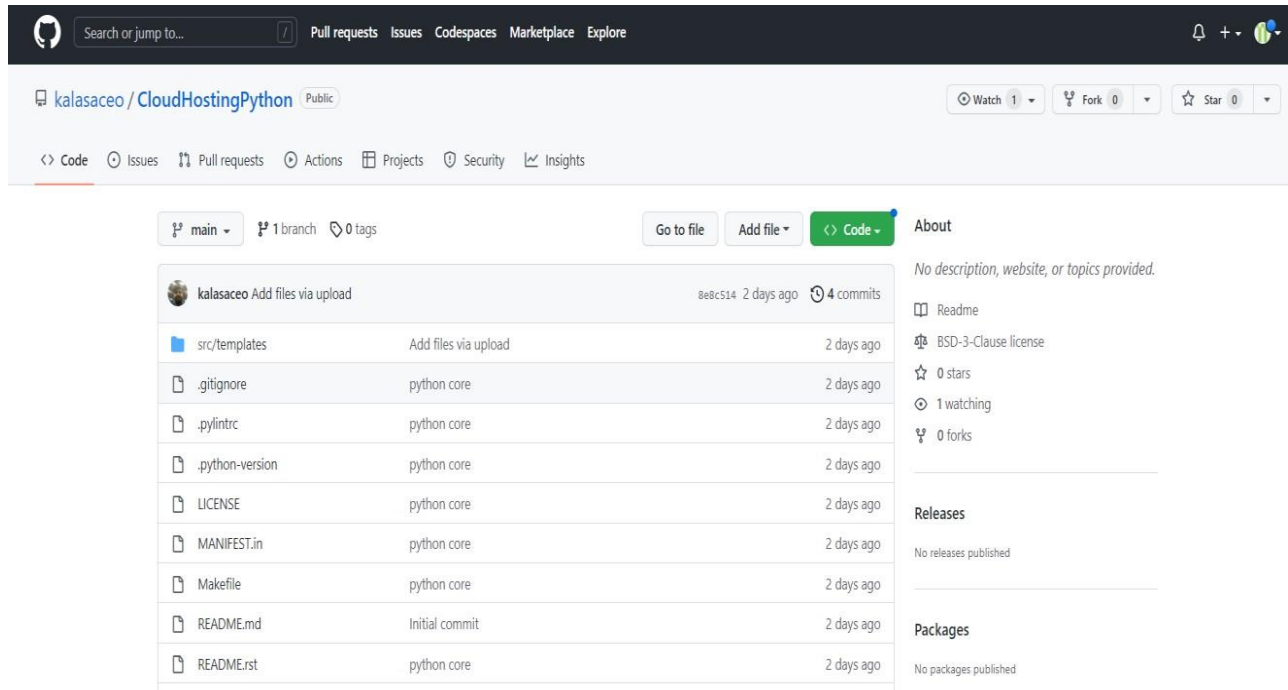
```
root@ip-172-31-21-216/html# systemctl start httpd.service
root@ip-172-31-21-216/html# systemctl enable httpd.service
root@ip-172-31-21-216/html#
```


Step 7 – Run the website



PROOF OF GITHUB UPLOAD:

Repository Link: <https://github.com/kalasaceo/CloudHostingPython>



The screenshot shows the GitHub repository page for `kalasaceo/CloudHostingPython`. The repository is public and has 1 watch, 0 forks, and 0 stars. The main branch is `main` with 1 branch and 0 tags. The repository contains 4 commits and 10 files. The files are listed in a table with columns for file name, commit message, and commit time. The files are: `src/templates` (Add files via upload, 2 days ago), `.gitignore` (python core, 2 days ago), `.pylintrc` (python core, 2 days ago), `.python-version` (python core, 2 days ago), `LICENSE` (python core, 2 days ago), `MANIFEST.in` (python core, 2 days ago), `Makefile` (python core, 2 days ago), `README.md` (Initial commit, 2 days ago), and `README.rst` (python core, 2 days ago). The right sidebar shows the repository's metadata, including the license (BSD-3-Clause), 0 stars, 1 watching, and 0 forks. The releases and packages sections show no published releases or packages.

File Name	Commit Message	Commit Time
<code>src/templates</code>	Add files via upload	2 days ago
<code>.gitignore</code>	python core	2 days ago
<code>.pylintrc</code>	python core	2 days ago
<code>.python-version</code>	python core	2 days ago
<code>LICENSE</code>	python core	2 days ago
<code>MANIFEST.in</code>	python core	2 days ago
<code>Makefile</code>	python core	2 days ago
<code>README.md</code>	Initial commit	2 days ago
<code>README.rst</code>	python core	2 days ago

CONCLUSION:

We have successfully deployed website on AWS cloud platform.