

PROFESSIONAL TRAINING REPORT
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfilment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By

Vignesh K S

(40120105)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI, TAMILNADU – 600119

April 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

(www.sathyabama.com)

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VIGNESH K S (Reg.no.40120105)** who carried out the project as a **"CREATE A WORDPRESS WEBSITE IN AWS"** under our supervision from January 2023 to April 2024.

 29/04/2023

Internal Guide

Dr. G. Mathivanan, M.E., PhD.

 29/4/23

Head of the Department

Dr. R. Subhashini, M.E., Ph.D.,



Internal Examiner



External Examiner

DECLARATION

I **VIGNESH K S (Reg. No. 40120105)** hereby declare that the Project Report entitled "**CREATE A WORDPRESS WEBSITE IN AWS**" is done by me under the guidance of and **Dr.G.Mathivanan, M.E., PhD.** is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 29/09/2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T Sasikala, M.E., Ph.D.**, Dean, School of Computing **Dr. R Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology for providing me with necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.M.Mathivannan, M.E., Ph.D.** for his valuable guidance, suggestions and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

CERTIFICATE OF COMPLETION

Ref.No.: 28 /FSPCC/BC001/04/2023



इलेक्ट्रॉनिक्स एवं
सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF
ELECTRONICS AND
INFORMATION TECHNOLOGY
सत्यमेव जयते



COURSE COMPLETION LETTER

It is certified that

Vignesh K S

Sathyabama Institute of Science and Technology

has successfully undergone the Bridge Course Training Program on

Cloud Computing Technology

Under "FutureSkills PRIME" Programme

commenced from 11 Mar, 2023

V. S. M.

Shri. V. Sridharan
Chief Investigator
Cloud Computing Technology
Lead Resource Centre

Cdr.(Retd.) L.R. Prakash
Senior Director
Centre Head
C-DAC, Chennai

ABSTRACT

This whitepaper provides system administrators with specific guidance on how to get started with WordPress on Amazon Web Services (AWS) and how to improve both the cost efficiency of the deployment and the end user experience. It also outlines a reference architecture that addresses common scalability and high availability requirements.

In this project, you will learn how to deploy and host WordPress, an open-source blogging tool and content management system (CMS) based on PHP and MySQL. You will implement an architecture to host WordPress for a production workload with minimal management responsibilities required from you. To accomplish this, you will use AWS Elastic Beanstalk and Amazon Relational Database Service (RDS). Once you upload the WordPress files, Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. Amazon RDS provides cost-efficient and resizable capacity, while managing time-consuming database administration task.

TABLE OF CONTENT

Chapter No.	Title	Page No.
1	Introduction	1
	1.1 Cloud Computing	
	1.2 Amazon Web Service	
	1.3 Benefits of Cloud Computing	
2	Aim and Scope of AWS	3
	2.1 Aim	
	2.2 Scope	
3	Problem Statement	4
	3.1 AWS Service	
	3.2 Objectives	
4	Methodology	7
	4.1 Connect To Your Windows Instance	
	4.2 Prerequisties	
	4.3 Accessing Cloud watch	
	4.4 Acessing EC2 Console	
5	Instances	9
6	Instances	14
7	Query Used in MobaXterm To Create a Wordpress Website	18
8	Result and Discussion	23
9	SSH Into The EC2 Instance You Want To Monitor	25
10	Future Work And Conclusion	27
	10.1 Future Work	
	10.2 Conclusion	

List of Images

Sl. No	Figure Number	Figure Description
1	6.1	Selecting an Instance
2	6.2	Elastic IP Address Running Successfully
3	6.3	Installing MobaXterm
4	6.4	IP Address Running In Ubuntu
5	6.5	Connecting To SSH Server
6	6.6	Login To Wordpress Website
7	7.1	SUDO APT Install APACHE2
8	7.2	SUDO APT Install PHP LIBAPACHE2-MOD-PHP PHP-MYSQL
9	7.3	SUDO APT Install MYSQL-SERVER
10	7.4	SUDO MYSQL -U ROOT
11	7.5	Create Database WB;
12	7.6	CD/TMP
13	7.7	TAR -XVF LATEST.TAR.GZ
14	7.8	SUDO APT-GET UPDATE

15	7.9	SUDO CERTBOT --APACHE
16	8.1	FINAL OUTPUT 1
17	8.2	FINAL OUTPUT 2
18	8.3	FINAL OUTPUT 3
19	8.4	FINAL OUTPUT 4

Chapter 1

INTRODUCTION

1.1 CLOUD COMPUTING

Cloud computing is a computing technique in which all the data is stored in the cloud. All the SDKs and databases are stored in the cloud only. A Cloud is primarily a platform that allow execution in various forms across multiple resources and potentially across enterprise boundaries. A Cloud is an infrastructure that enables execution of codes (applications, services, etc.) in a managed and elastic fashion, here the meaning of managed is reliability according to pre-defined quality parameters which are automatically ensured and elastic here means that the resources are used according to the current actual requirements. Cloud fits the client-server model, and so far the typical cloud client is the same as the typical enterprise client (i.e. single desktop or laptop computer), some observers have a tendency to stop at this level of analysis. But of course, the real action in the cloud happens on the server side and that's where things get interesting.

1.2 AMAZON WEB SERVICE

The leading cloud provider in the marketplace is Amazon Web Services. It provides over 170 AWS services to the developers so they can access them from anywhere at the time of need. AWS has customers in over 190 countries worldwide including 5000 ed-tech institutions and 2000 government organizations. Many companies ESPN, Adobe, Twitter, Netflix, Facebook, BBC, etc, use AWS services. For example, Adobe creates and updates software without depending upon the IT teams. It uses its services by offering multi-terabyte operating environments for its clients. By deploying its services with Amazon services, Adobe integrated and operated its software in a simple manner.

1.3 BENEFITS OF CLOUD COMPUTING

Cloud computing has been around for approximately two decades and despite the data pointing to the business efficiencies, cost-benefits, and competitive advantages it holds, a large portion of the business community continues to operate without it. According to a study by the International Data Group, 69% of businesses are already using cloud technology in one capacity or another, and 18% say they plan to implement cloud-computing solutions at some point. At the same time, Dell reports that 11 companies that invest in big data, cloud, mobility, and security enjoy up to 53% faster revenue growth than their competitors. As this data shows, an increasing number of tech-savvy businesses and industry leaders are recognizing the many benefits of the cloud-computing trend. But more than that, they are using this technology to more efficiently run their organizations, better serve their customers, and dramatically increase their overall profit margins.

- Cost Savings
- Security
- Flexibility
- Mobility
- Insight
- Increased Collaboration
- Disaster Recovery
- Loss Prevention
- Automatic Software Updates
- Competitive Edge
- Sustainability

CHAPTER 2

AIM AND SCOPE OF AWS

2.1 AIM

The aim of the project is to create a Virtual Machine (VM) on a Cloud Based system called AWS (Amazon Web Service). It allows using cloud storage so that it enables us to create a server and initialize an alert while the server of the particular instance goes down. Instances are useful for running memory-intensive workloads such as real-time data ingestion, distributed in-memory caches, big data analytics, memory-intensive enterprise applications, and high-performance databases.

2.2 SCOPE

Amazon EC2 provides the broadest and deepest instance choice to match your workload's needs. General purpose, compute optimized, memory optimized, storage optimized, and accelerated computing instance types are available that provide the optimal compute, memory, storage, and networking balance for your workloads. Processors from Intel, AMD, NVIDIA and AWS power these instance types and provide additional performance and cost optimizations. Local storage and enhanced networking options available with instance types further help optimize performance for workloads that are disk or network I/O bound. Many instance types also offer bare metal instances that provide your applications with direct access to the processor and memory of the underlying server for running in non-virtualized environments or for applications where you want to use your own hypervisor.

Chapter 3

PROBLEM STATEMENT

Whenever you install a WordPress on Machines you have config files related to WP. Now if you install the WordPress on multiple machine each machine will have its own files and if you run those machines behind some Load Balancer then you will encounter an issues with it. one more thing is Database maintenance for your WordPress site is critical. Your database instance holds all of your important data for your WordPress site. If the database goes down, your website may go down with it, and you could even lose your data.

3.1 AWS SERVICE

1. Relational Database Service — We will use AWS RDS MySQL which is fully Managed MySQL Database services provided by AWS
2. Application Load Balancer — We will use AWS ALB as a single-entry point for our website. ALB will route requests to multiple EC2 behind ALB
3. Elastic File System — AWS EFS is fully managed network file system which we will use for storing all our WordPress config files, plugins, Webpages. Another advantage of EFS is we can mount EFS to Multiple EC2 Instances, so that every EC2 Instance points to same files and we can run multiple EC2 to achieve High Availability and Fault Tolerance.

3.2. OBJECTIVES

Successfully complete this lab by achieving the following learning objectives:

Create RDS Database

Using the AWS console, create an RDS database with the following configurations:

- Choose a database creation method: Standard create
- Engine options: MySQL
- Edition: MySQL Community
- Version: MySQL 8.0.28
- Templates: Free tier
- DB instance class: db.t2.micro
- DB instance identifier: word press

Chapter 4

LITERATURE SURVEY

The Hosting word press on AWS reference architecture available on GitHub outlines best practices for deploying WordPress on AWS and includes a set of AWS Cloud Formation templates to get you up and running quickly. The following architecture is based on that reference architecture. The rest of this section will review the reasons behind the architectural choices.

The based AMI in the GitHub was changed from Amazon Linux1 to Amazon Linux2 in July 2021. However, deployment templates at S3 were not changed yet. It is recommended to use templates at GitHub if there is an issue to deploy the reference architecture with templates at S3.

The reference architecture illustrates a complete best practice deployment for a WordPress website on AWS.

- It starts with edge caching in **Amazon Cloud Front** (1) to cache content close to end users for faster delivery.
- Cloud Front pulls static content from an **S3 bucket** (2) and dynamic content from an **Application Load Balancer** (4) in front of the web instances.
- The web instances run in an **Auto Scaling group** of **Amazon EC2instances** (6).
- An **Elastic Cache** cluster (7) caches frequently queried data to speed up responses.

An **Amazon Aurora** MySQL instance (8) hosts the WordPress database.

- The WordPress EC2 instances access shared WordPress data on an **Amazon EFS** file system via an **EFS Mount Target** (9) in each Availability Zone.
- An **Internet Gateway** (3) enables communication between resources in your VPC and the internet.
- **NAT Gateways** (5) in each Availability Zone enable EC2 instances in private subnets (App and Data) to access the internet.

Chapter 4

METHODOLOGY

4.1. CONNECT TO YOUR WINDOWS INSTANCE

AWS provides a variety of services that can be used to host a WordPress website. Some of the popular services include Amazon Elastic Compute Cloud (EC2), Amazon Light sail, and AWS Elastic Beanstalk. It is available on most editions of Windows and is also available for Mac OS.

4.2. PREREQUISITES

To connect using MobaXterm

- Install MobaXterm
- Once your instance is up and running, you can create a SSH session, there you need to enter the remote host and specify ubuntu and in advice settings you need to use your private key to access the terminal.

4.3. ACCESSING CLOUD WATCH

Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications.

- The CloudWatch home page automatically displays metrics about every AWS service you use. You can additionally create custom dashboards to display metrics about your custom applications, and display custom collections of metrics that you choose.
- You can create alarms that watch metrics and send notifications or automatically make changes to the resources you are monitoring when a threshold is breached. With CloudWatch, you gain system-wide visibility into resource utilization, application performance, and operational health

4.4. ACCESING EC2 COSOLE

- Open the Amazon EC2 console.
- From the navigation pane, choose Instances.
- Select the instance and then choose Connect.

On the Connect to instance page, choose the option to Connect using Fleet Manager, then choose Fleet Manager Remote Desktop. This opens the Fleet Manager Remote Desktop page in the AWS Systems Manager cons

Chapter 5

CloudFront distribution creation

Create a CloudFront web distribution by following the Distribution, the default Origin and Behavior automatically created will be used for dynamic content. Create four additional behaviors to further customize the way both static and dynamic requests are treated.

The following table summarizes the configuration properties for the five behaviors. You can also skip this manual configuration and use the AWS for WordPress plugin covered in Appendix B: Plugins Installation and Configuration, which is the easiest way to configure CloudFront to accelerate your WordPress site.

For the default behavior, AWS recommends the following configuration:

- Allow the Origin Protocol Policy to Match Viewer, so that if viewers connect to CloudFront using HTTPS, CloudFront connects to your origin using HTTPS as well, achieving end-to-end encryption. Note that this requires you install a trusted SSL certificate on the load balancer. For details, refer to Requiring HTTPS for Communication Between CloudFront and Your Custom Origin.

Allow all HTTP methods since the dynamic portions of the website require both GET and POST requests (for example, to support POST for the comment submission forms)

- Forward only the cookies that vary the WordPress output, for example, `wordpress_*`, `wp-settings-*` and `comment_*`. You must extend that list if you have installed any plugins that depend on other cookies not in the list.
- Forward only the HTTP headers that affect the output of WordPress, for example `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer`, and `CloudFront-is-Tablet-Viewer`:
- Host allows multiple WordPress websites to be hosted on the same origin.
- `CloudFront-Forwarded-Proto` allows different versions of pages to be cached depending on whether they are accessed via HTTP or HTTPS.

- CloudFront-is-Desktop-Viewer, CloudFront-is-Mobile-Viewer, and CloudFront-is-Tablet-Viewer allow you to customize the output of your themes based on the end user's device type.
- Forward all the query strings to cache based on their values because WordPress relies on these, they can also be used to invalidate cached objects.

If you want to serve your website under a custom domain name (not *.cloudfront.net), enter the appropriate URIs under **Alternate Domain Names** in the Distribution Settings. In this case, you also need an SSL certificate for your custom domain name. You can request SSL certificates via the AWS Certificate Manager and configure them against a CloudFront distribution.

Now, create two more cache behaviors for dynamic content: one for the login page (path pattern: wp-login.php) and one for the admin dashboard (path pattern: wp-admin/*). These two behaviors have the exact same settings, as follows:

- Enforce a Viewer Protocol Policy of HTTPS Only.
- Allow all HTTP methods.
- Cache based on all HTTP headers.
- Forward all cookies.
- Forward and cache based on all query strings.

The reason behind this configuration is that this section of the website is highly personalized and typically has just a few users, so caching efficiency isn't a primary concern. The focus is to keep the configuration simple to ensure maximum compatibility with any installed plugins by passing all cookies and headers to the origin.

The AWS for WordPress plugin covered in Appendix B automatically creates a CloudFront distribution that meets the preceding configuration.

By default, WordPress stores everything locally on the web server, which is block storage (Amazon EBS) for single server deployment, and file storage (Amazon EFS) for elastic deployment. In addition to reducing storage and data transfer costs, moving

static assets to Amazon S3 offers scalability, data availability, security and performance. There are several plugins that make it easy to move static content to Amazon S3; one of them is W3 Total Cache, also covered in Appendix B.

LightSail provides templates for commonly used applications such as WordPress. This template is a great starting point for running your own WordPress website, as it comes pre-installed with most of the software you need. You can install additional software or customize the software configuration by using the in-browser terminal or your own SSH client, or via the WordPress administration web interface.

Amazon LightSail has a partnership with GoDaddy Pro Sites product to help WordPress customers easily manage their instances for free. LightSail WordPress virtual servers are preconfigured and optimized for fast performance and security, making it easy to get your WordPress site up and running in no time. Customers running multiple WordPress instances find it challenging and time-consuming to update, maintain and manage all of their sites. With this integration, you can easily manage your multiple WordPress instances in minutes with only a few clicks.

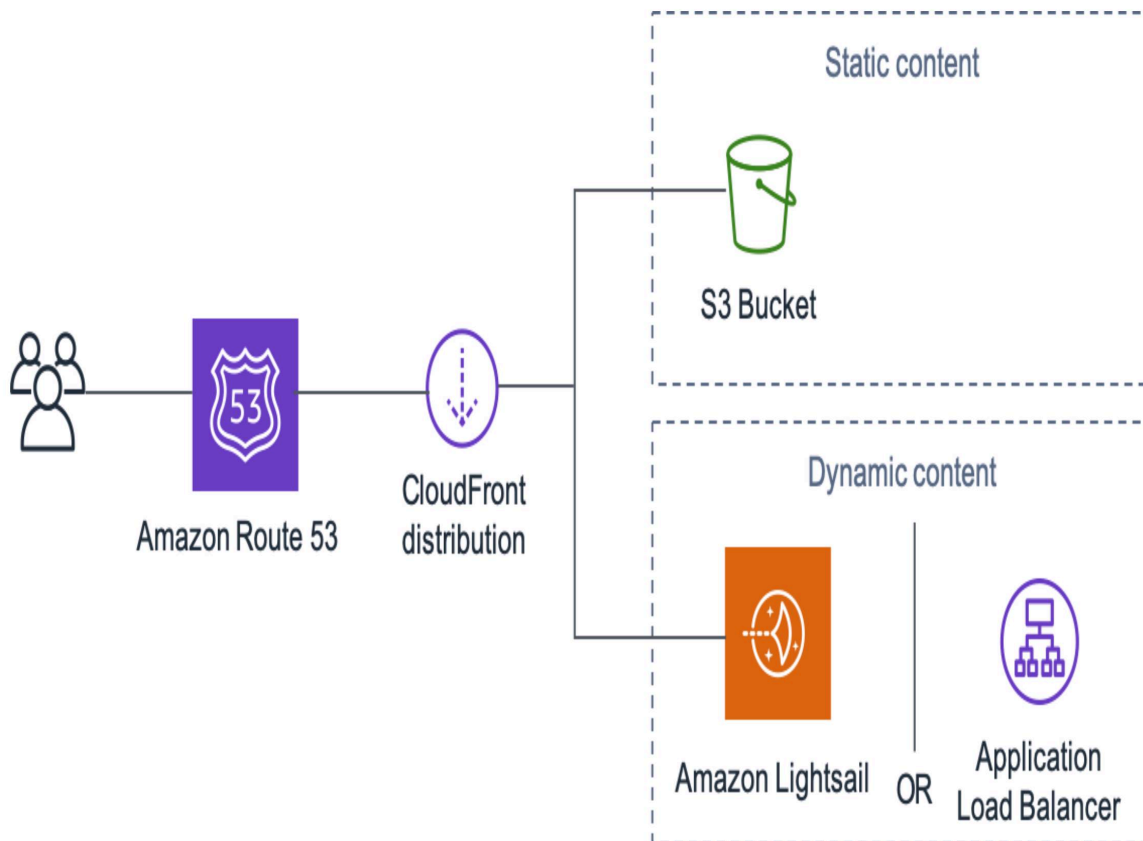
For more information about managing WordPress on LightSail, refer to *Getting started using WordPress from your Amazon LightSail instance*. Once you are finished customizing your WordPress website, AWS recommends that you take a snapshot of your instance.

5.1 Recovering from failure

- A single web server is a single point of failure, so you must ensure that your website data is backed up. The snapshot mechanism described earlier can also be used for this purpose. To recover from failure, you can restore a new instance from your most recent. To reduce the amount of data that could be lost during a restore, your snapshots must be as recent as possible.
- To minimize the potential for data loss, ensure that snapshots are taken on a regular basis. You can schedule automatic snapshots of your Lightsail Linux/Unix instances. For instructions, refer to [Enabling or disabling automatic snapshots for instances or disks in Amazon Lightsail](#).
- AWS recommends that you use a static IP—a fixed, public IP address that is dedicated to your Lightsail account. If you need to replace your instance with another one, you can reassign the static IP to the new instance. In this way, you don't have to reconfigure any external systems (such as DNS records) to point to a new IP address every time you want to replace your instance.
- A number of options that can improve aspects of your WordPress website's performance and scalability. Some can be applied to a single-server deployment, whereas others take advantage of the scalability of multiple servers. Many of those modifications require the use of one or more WordPress plugins. Although various options are available, W3 Total Cache is a popular choice that combines many of those modifications in a single plugin.

5.2 Dynamic content

Dynamic content includes the output of server-side WordPress PHP scripts. Dynamic content can also be served via CloudFront by configuring the WordPress website as an origin. Since dynamic content includes personalized content, you need to configure CloudFront to forward certain HTTP cookies and HTTP headers as part of a request to your custom origin server. CloudFront uses the forwarded cookie values as part of the key that identifies a unique object in its cache. To ensure that you maximize the caching efficiency, configure CloudFront to forward only those HTTP cookies and HTTP headers that actually vary the content (not cookies that are only used on the client side or by third-party applications, for example, for web analytics).



Chapter 6

INSTANCES

What is Amazon EC2 Instance?

An **EC2 instance** is nothing but a virtual server in Amazon Web services terminology. It stands for **Elastic Compute Cloud**. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud. An **on-demand** EC2 instance is an offering from AWS where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications. The instance will be charged per hour with different rates based on the type of the instance chosen. AWS provides multiple instance types for the respective business needs of the user.

Thus, you can rent an instance based on your own CPU and memory requirements and use it as long as you want. You can terminate the instance when it's no more used and save on costs. This is the most striking advantage of an on-demand instance- you can drastically save on your CAPEX.

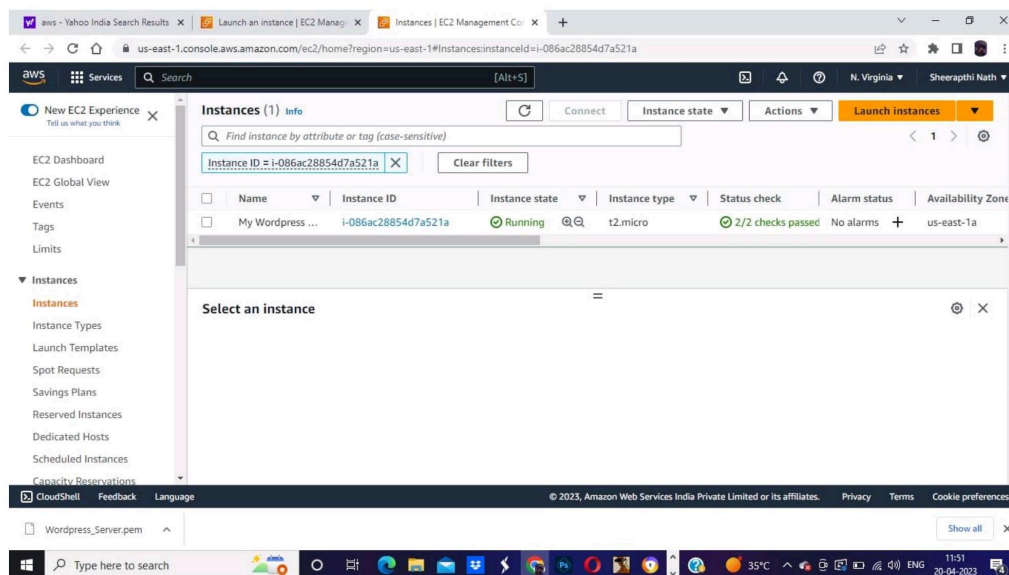


Fig 6.1: SELECTING AN INSTANCE

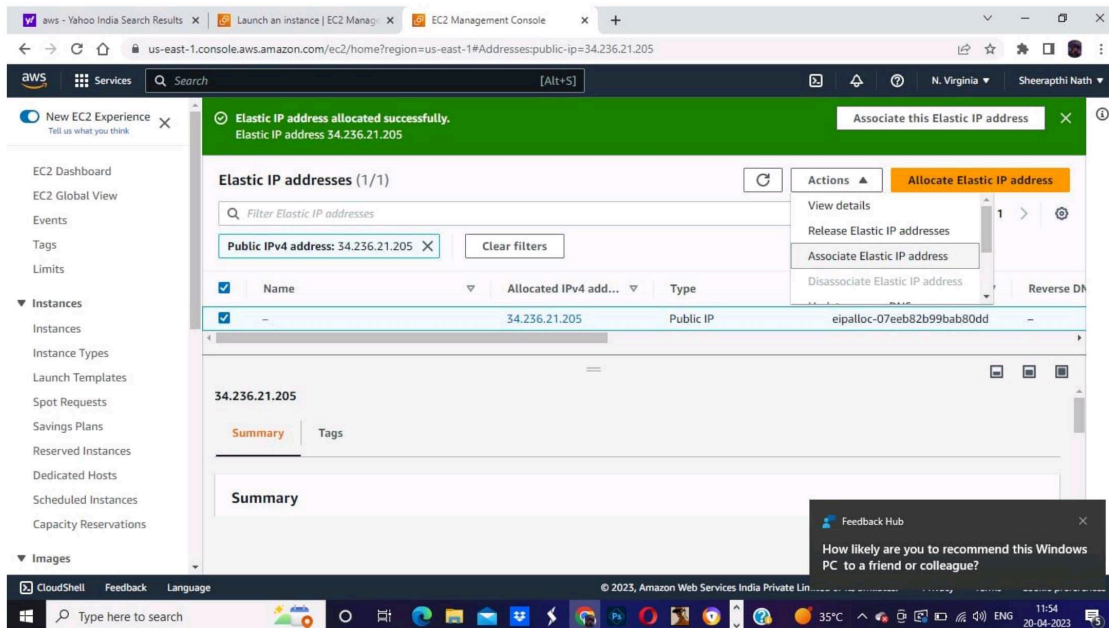


Fig 6.2: ELASTIC IP ADDRESS RUNNING SUCCESSFULLY

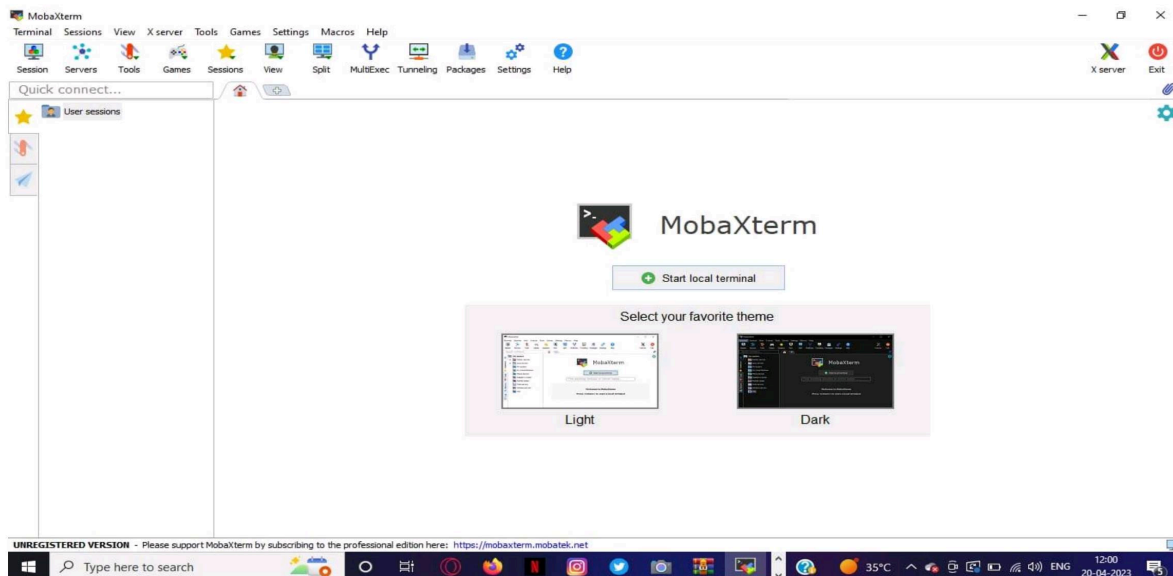


Fig 6.3 : INSTALLING MOBAXTERM

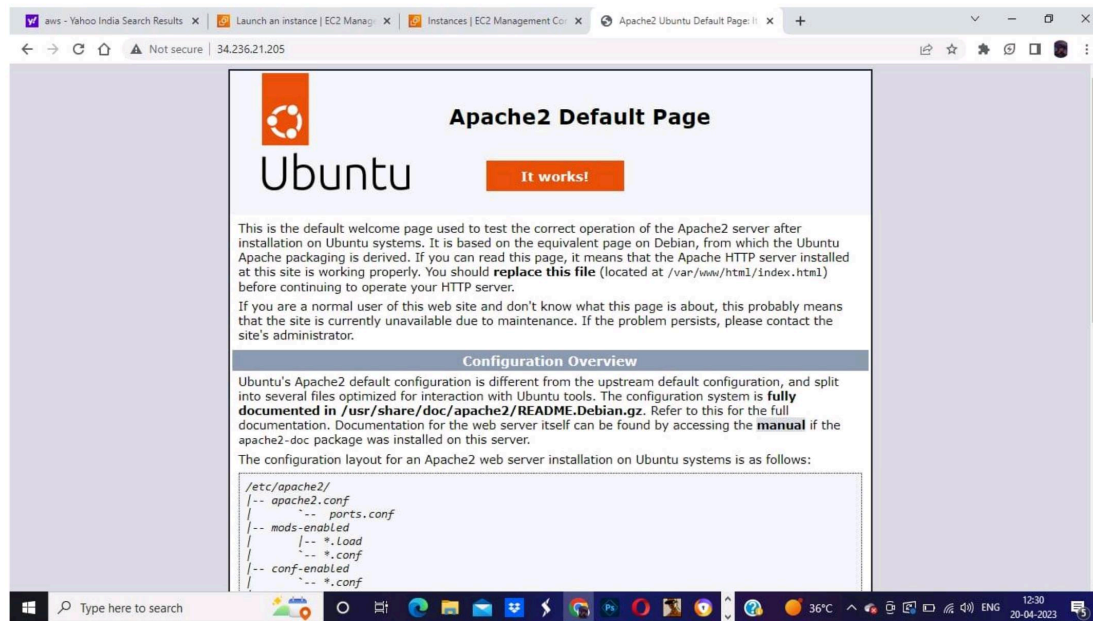


Fig 6.4: IP ADDRESS RUNNING IN UBUNTU

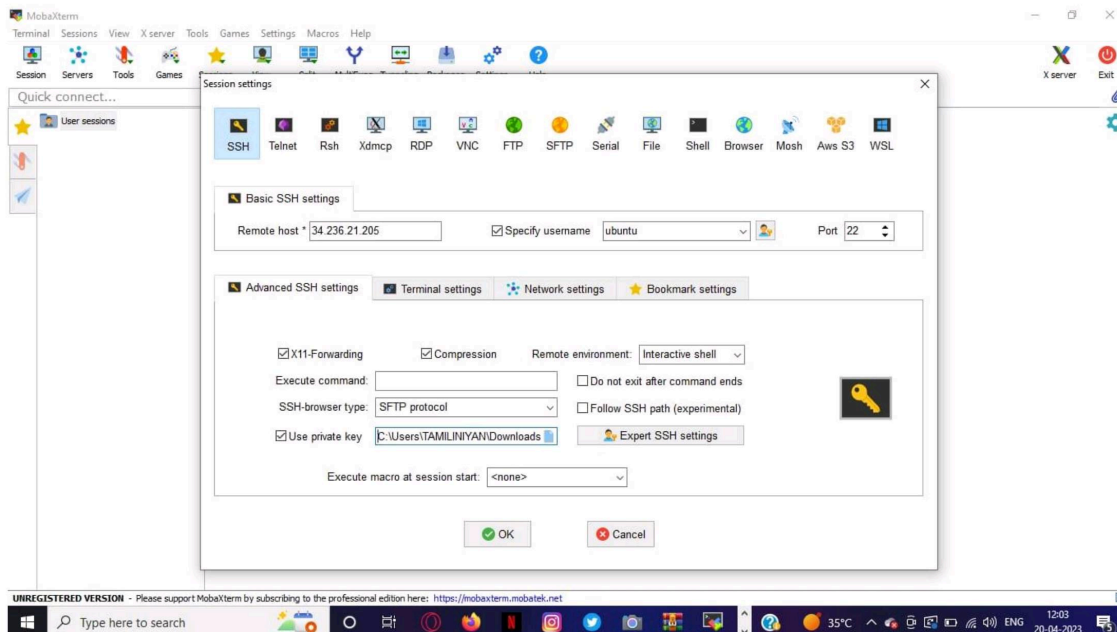


Fig 6.5: CONNECTING TO SSH SERVER

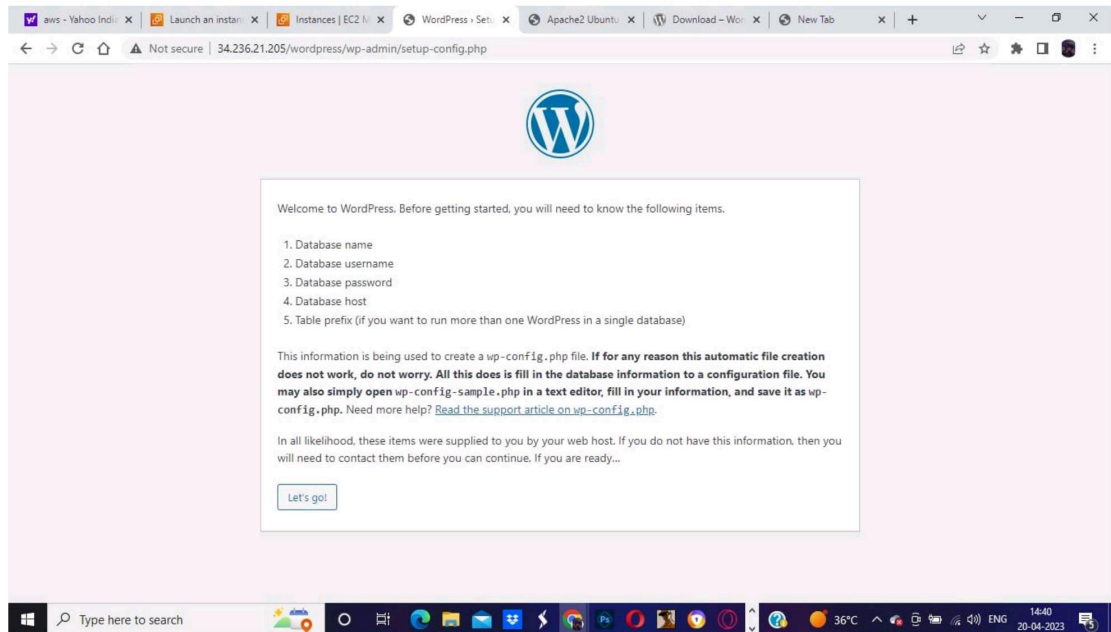
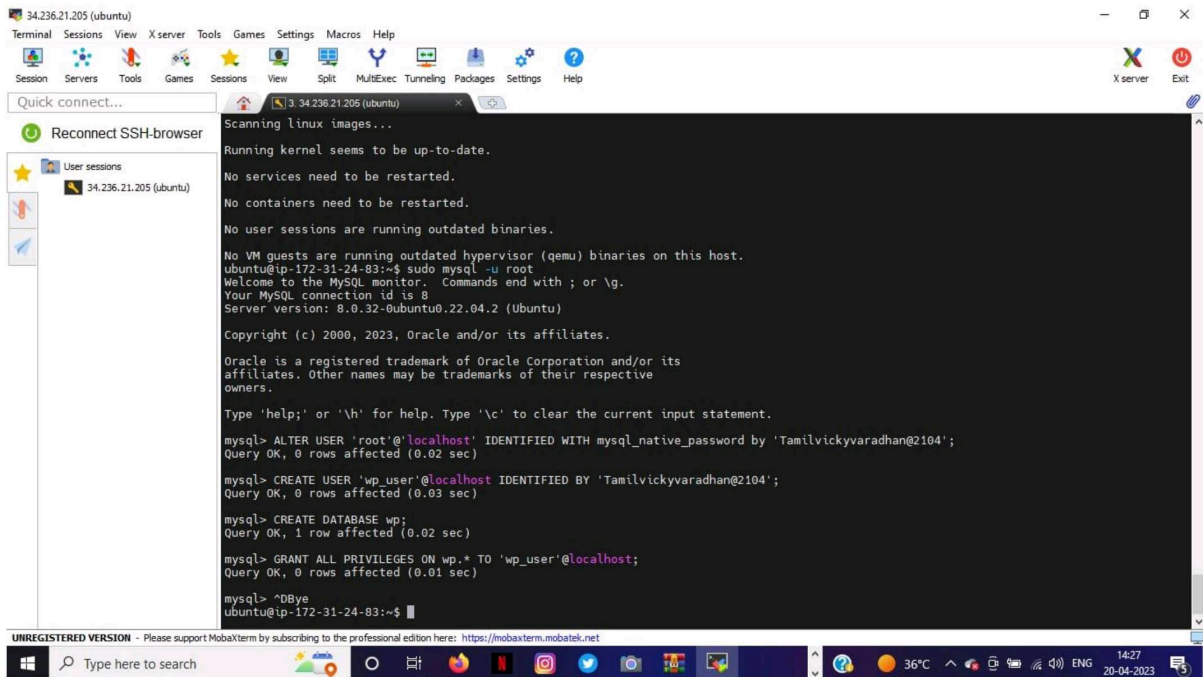


Fig 6.6: LOGIN TO WORDPRESS WEBSITE

CHAPTER 7

QUERY USED IN MOBAXTERM TO CREATE A WORD PRESS WEBSITE



The screenshot shows a MobaXterm window with a terminal session on a remote Ubuntu server (34.236.21.205). The terminal output shows the MySQL prompt and several commands being executed to set up a database for WordPress. The commands include altering the root user, creating a new user 'wp_user', creating a database 'wp', and granting all privileges to 'wp_user'.

```
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-24-83:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'Tamilvickyvaradhan@2104';
Query OK, 0 rows affected (0.02 sec)

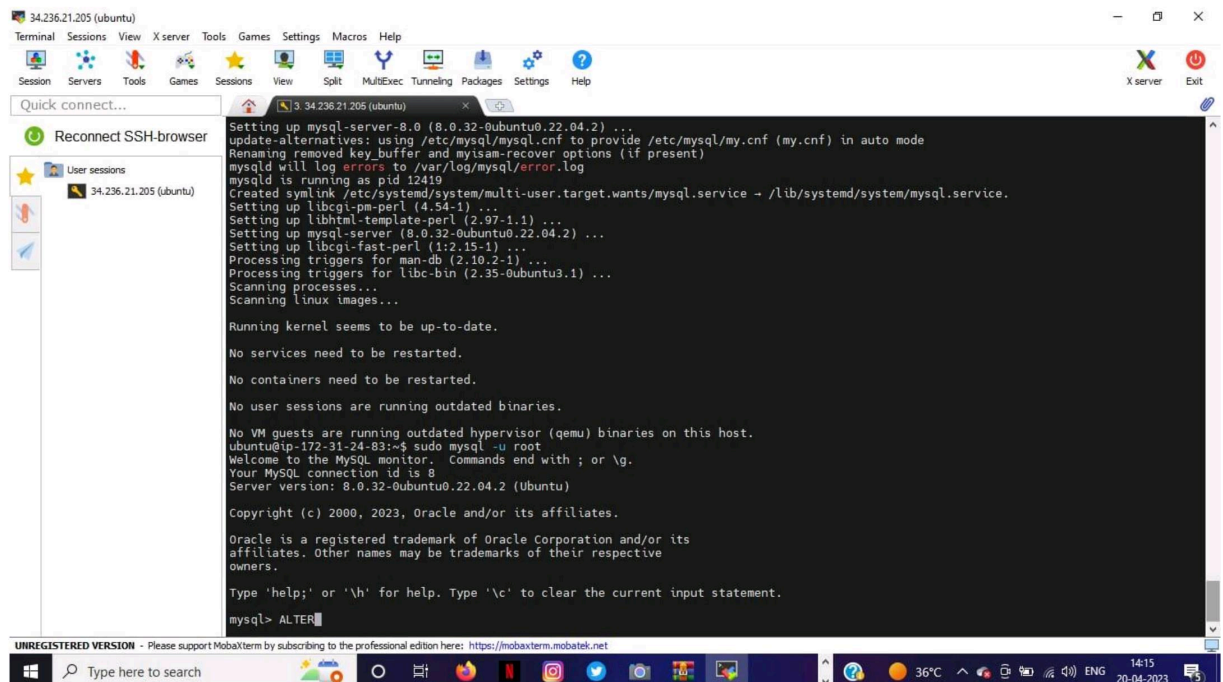
mysql> CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'Tamilvickyvaradhan@2104';
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE DATABASE wp;
Query OK, 1 row affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON wp.* TO 'wp_user'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> ^DBye
ubuntu@ip-172-31-24-83:~$
```

Fig 7.1: SUDO APT INSTALL APACHE2



The screenshot shows a MobaXterm window with a terminal session on a remote Ubuntu server (34.236.21.205). The terminal output shows the command 'sudo apt install apache2' being executed, followed by the installation progress and the resulting system state. The output includes messages about updating alternatives, removing key buffers, and creating symlinks for the Apache service.

```
Setting up mysql-server-8.0 (8.0.32-0ubuntu0.22.04.2) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Removing removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 12419
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up libcgi-pm-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mysql-server (8.0.32-0ubuntu0.22.04.2) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-24-83:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER
```

Fig 7.2: SUDO APT INSTALL PHP LIBAPACHE2-MOD-PHP PHP-MYSQL

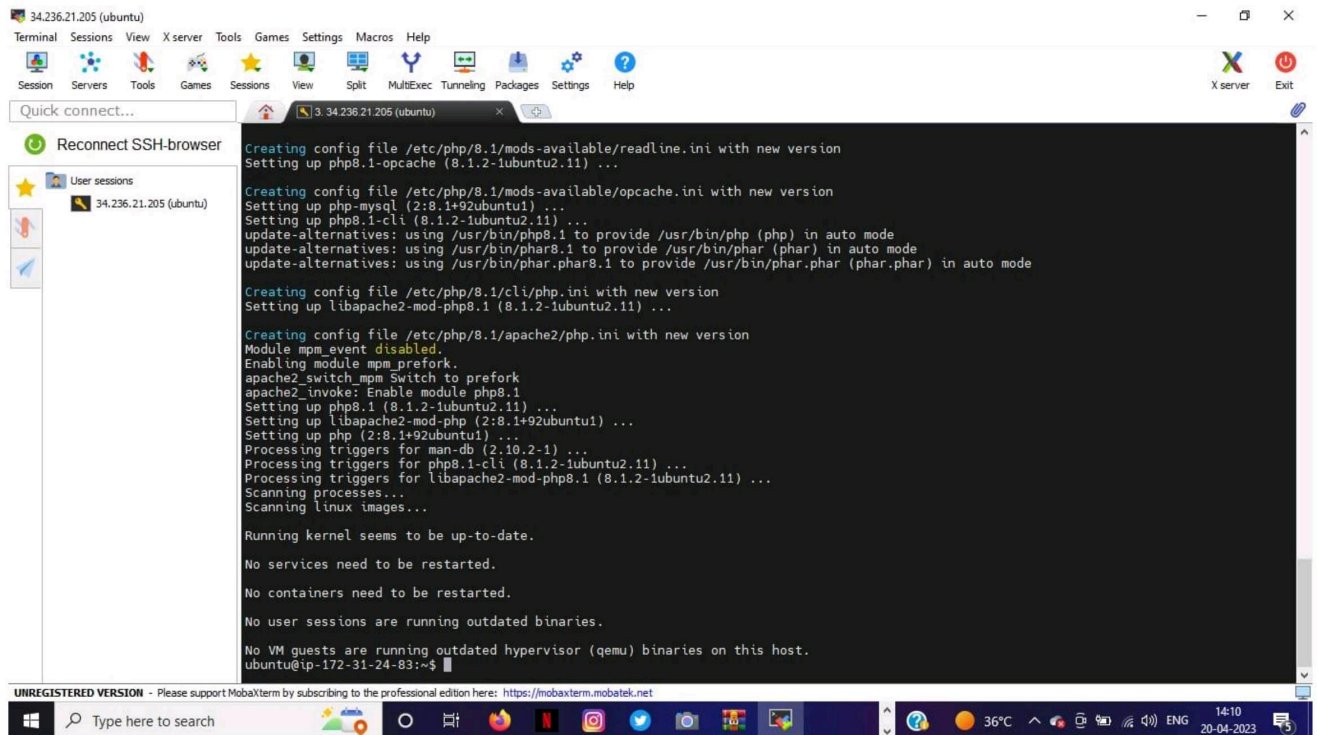


Fig 7.3: SUDO APT INSTALL MYSQL-SERVER

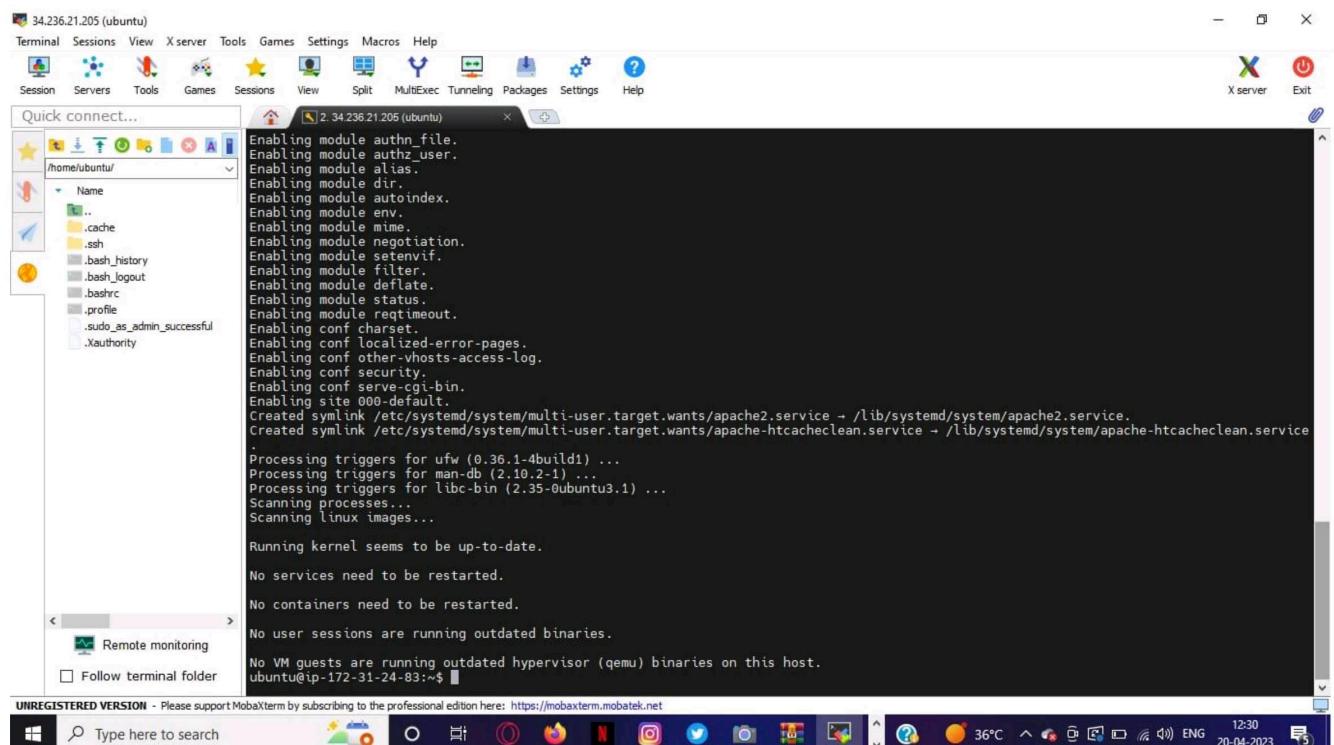


Fig 7.4: SUDO MYSQL -U ROOT

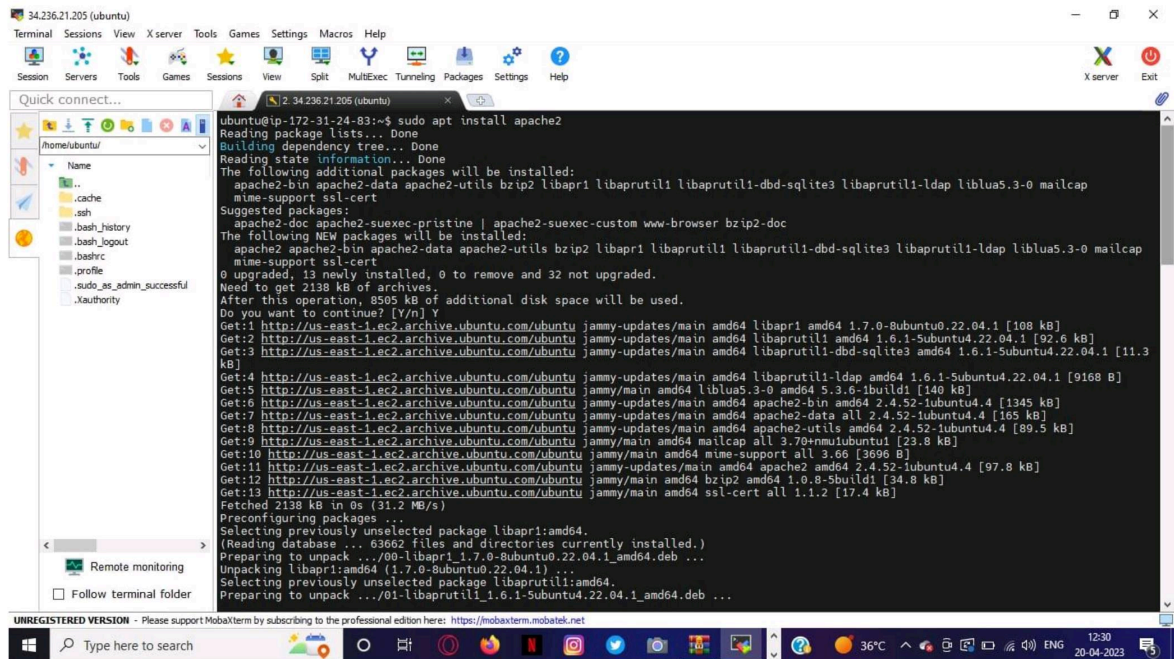


Fig 7.5: CREATE DATABASE WB;

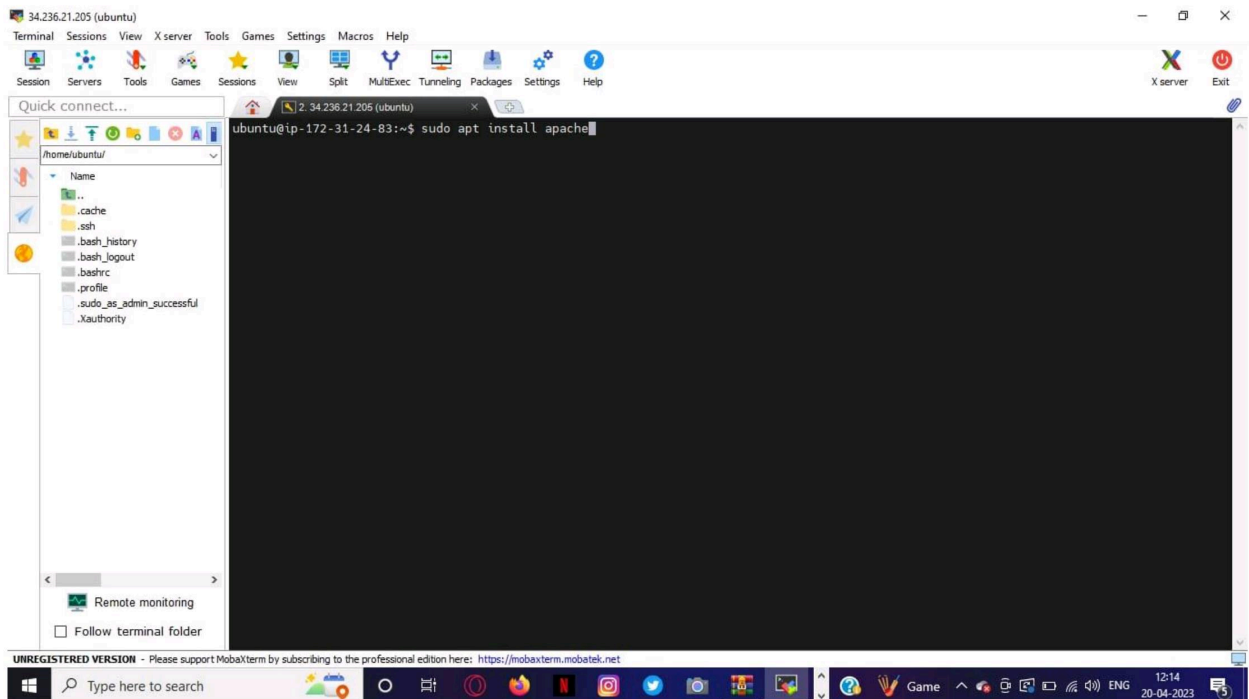


Fig 7.6: CD/TMP

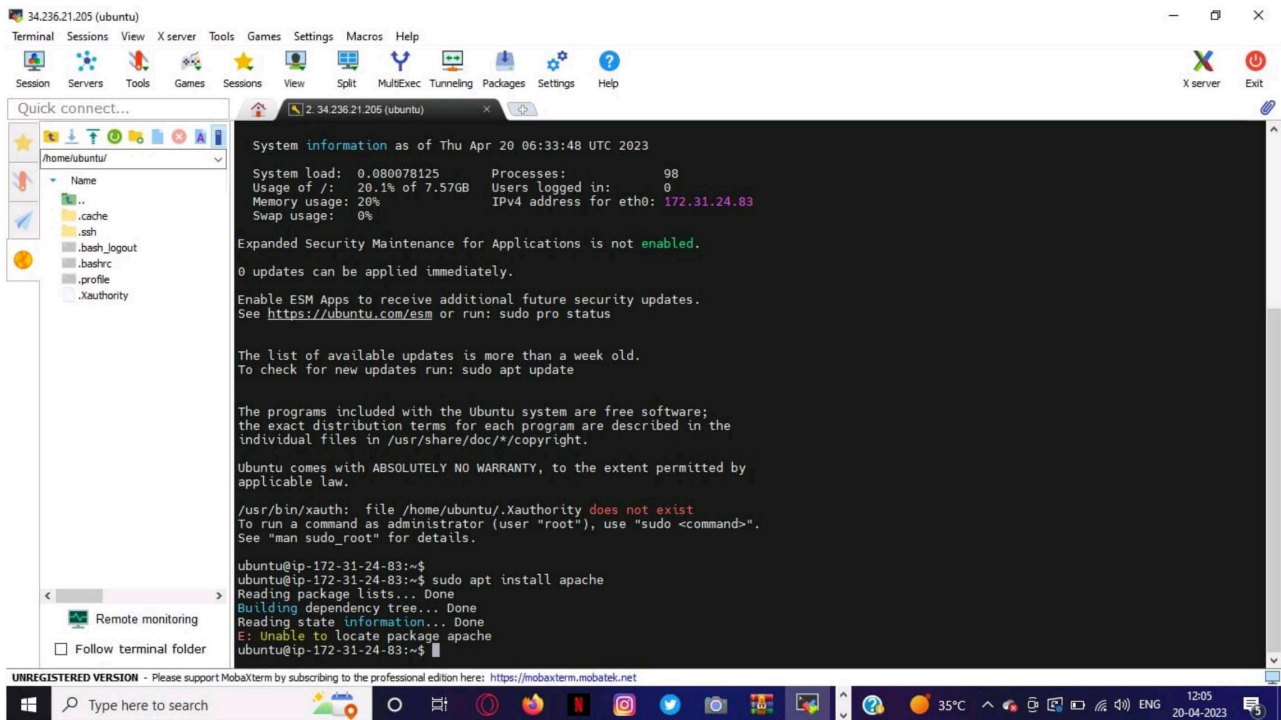


Fig 7.7: TAR -XVF LATEST.TAR.GZ

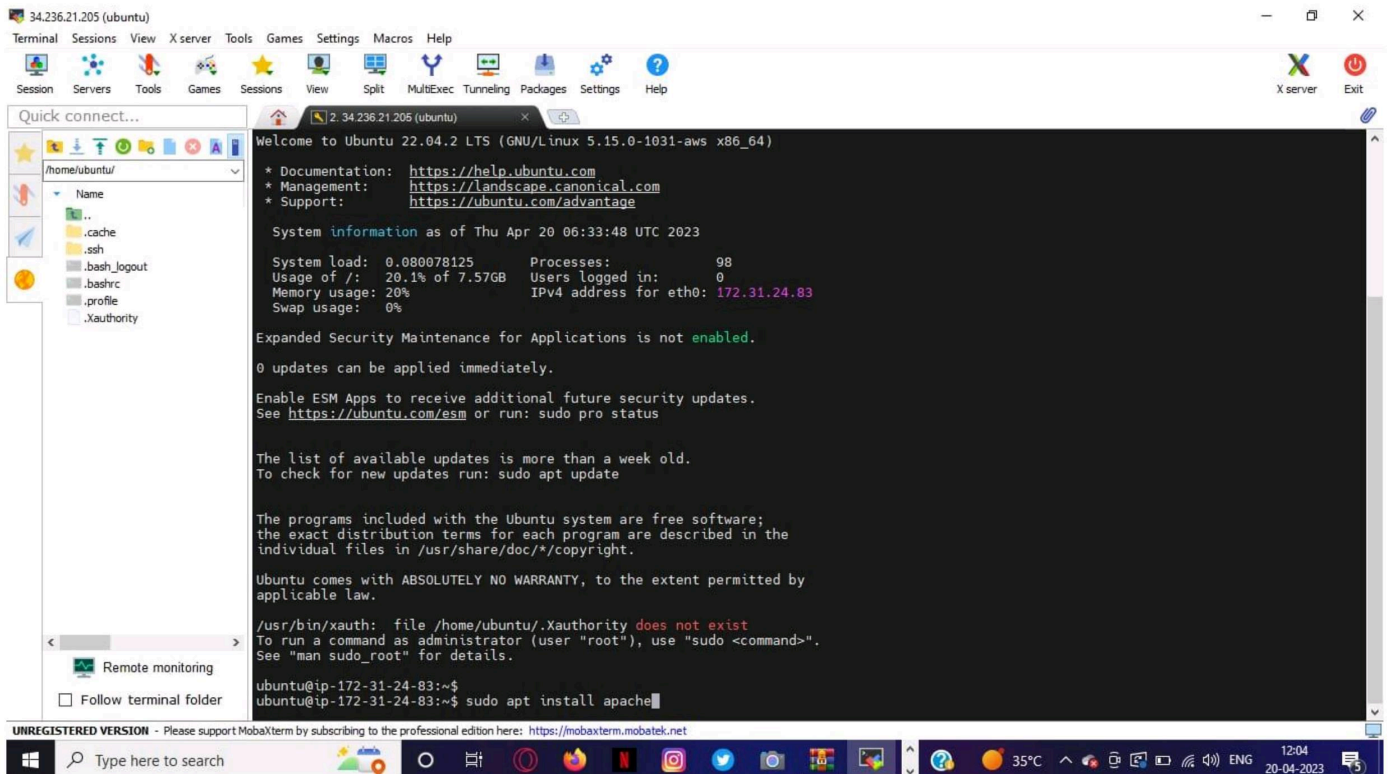


Fig 7.8: SUDO APT-GET UPDATE

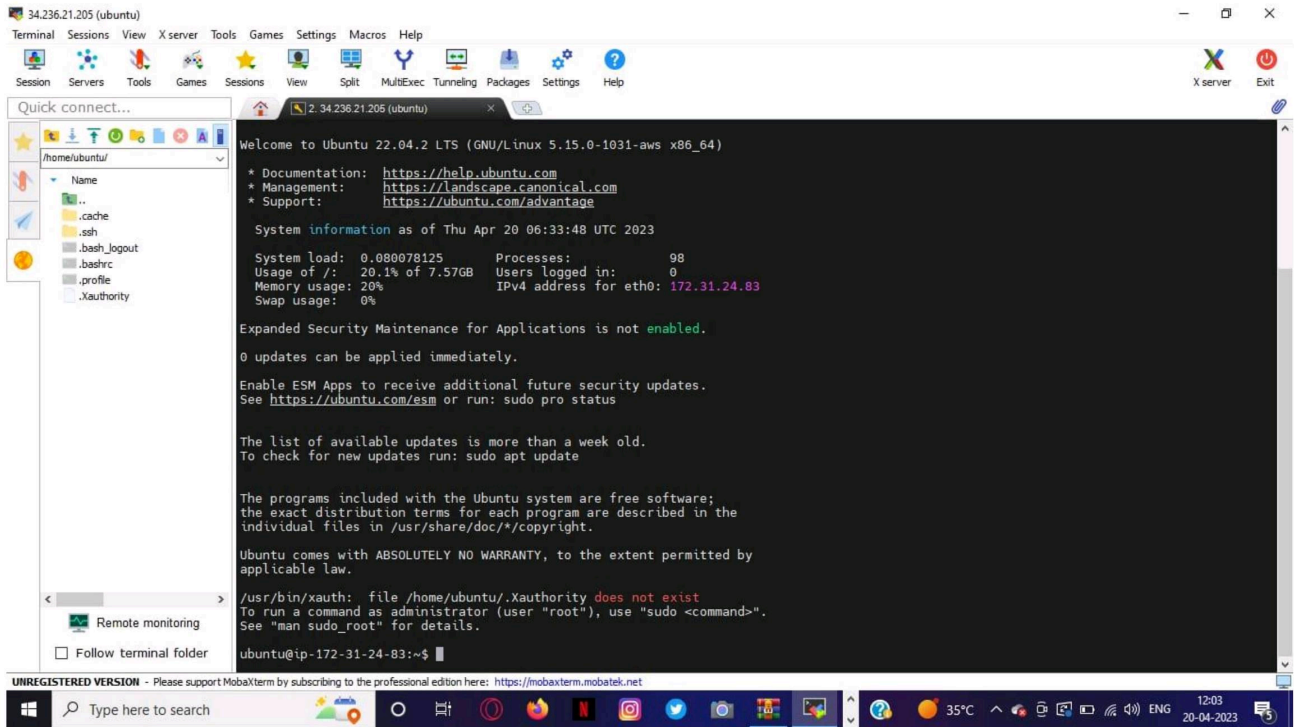


Fig 7.9: SUDO CERTBOT --APACHE

CHAPTER 8

RESULT AND DISCUSSION

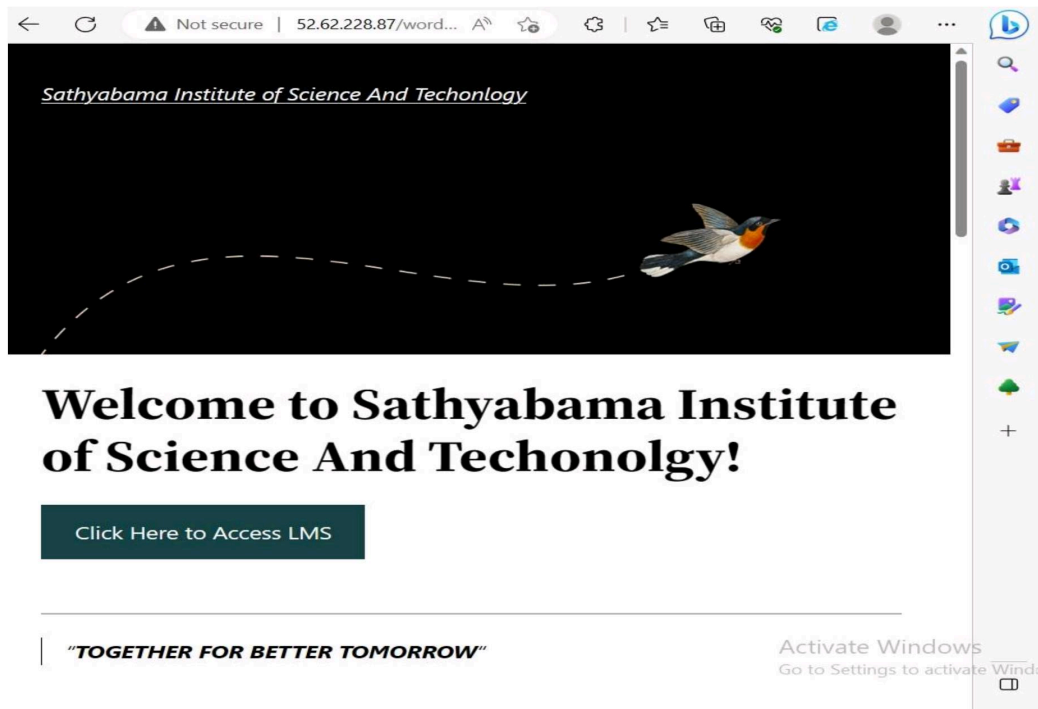


Fig 8.1: OUTPUT 1

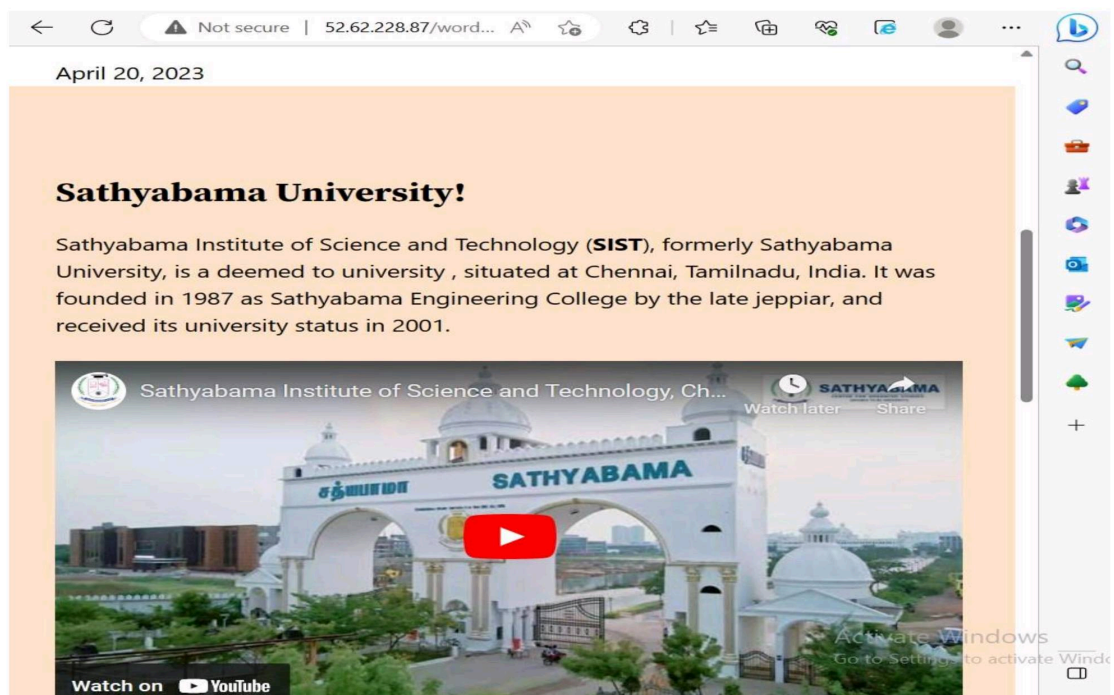


Fig 8.2: OUTPUT 2

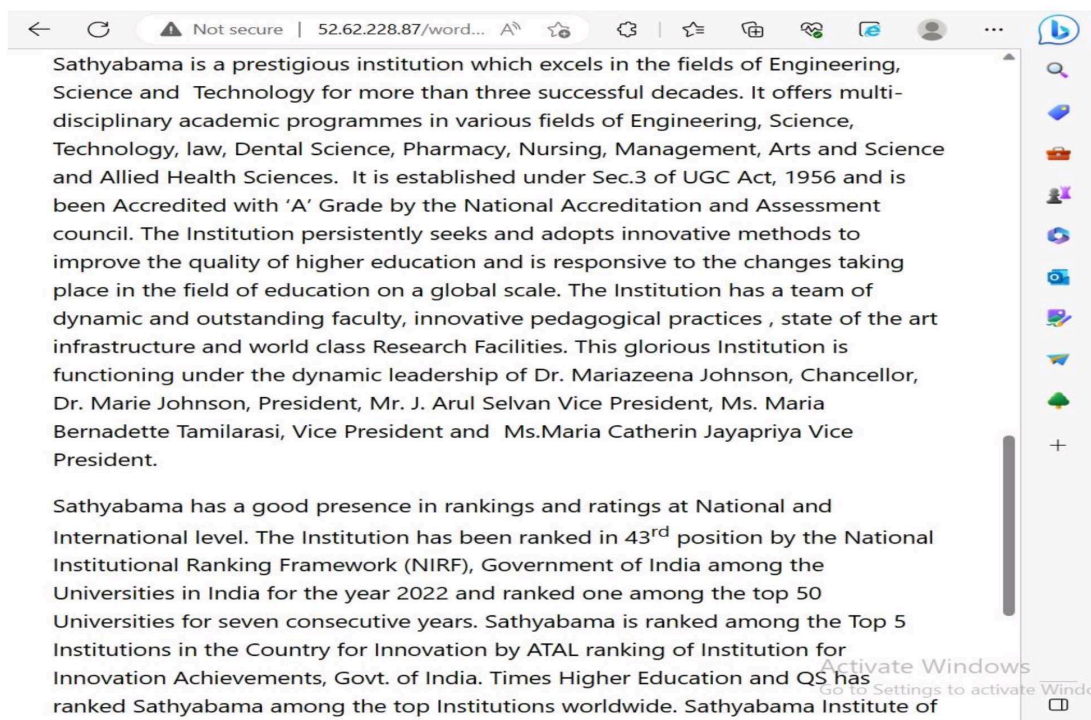


Fig 8.3

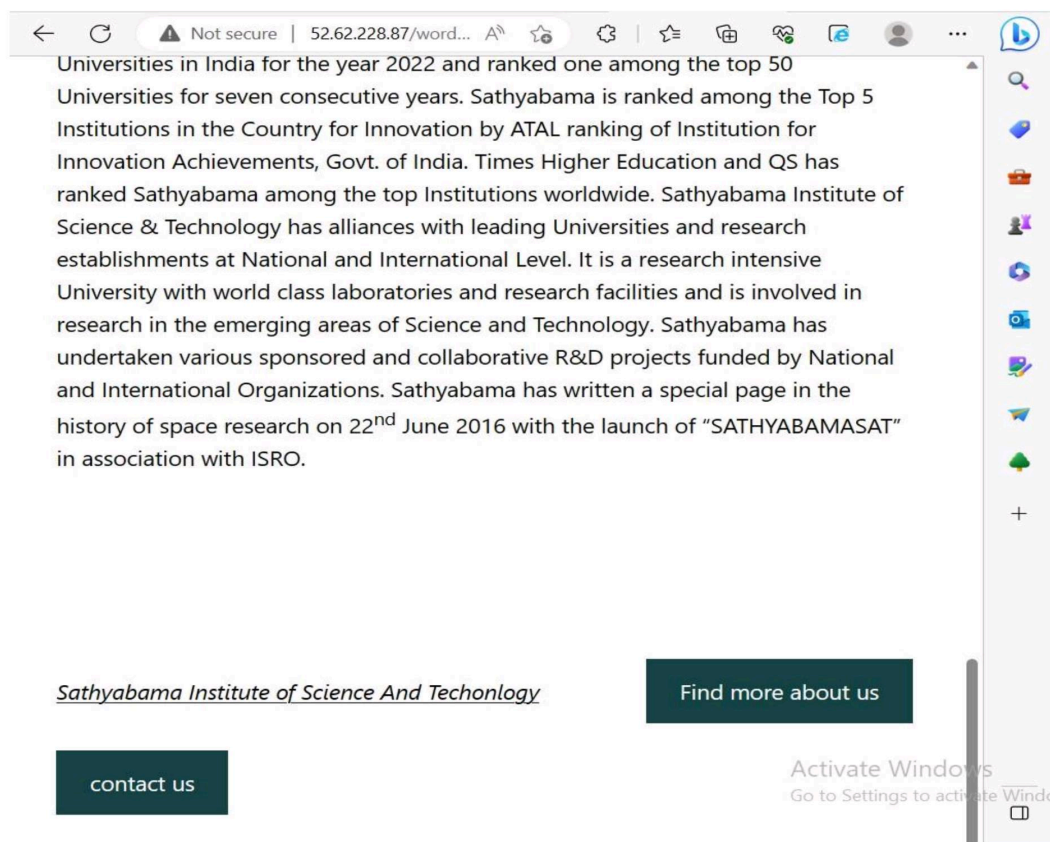


Fig 8.4

CHAPTER 9

SSH INTO THE EC2 INSTANCE YOU WANT TO MONITOR

9.1 Download and install the CloudWatch agent by following the instructions [CloudWatch-Agent-on-EC2-Instance.html](#)

9.2 Configure the CloudWatch agent by creating a configuration file in the `/opt/aws/amazon-cloudwatch-agent/etc` directory. Here's an example configuration file:

```
{
  "metrics": { "append_dimensions": { "InstanceId": "${aws:InstanceId}"
  },
  "metrics_collected": { "mem": { "measurement": [ "mem_used_percent"
  ]
  },
  "swap": { "measurement": [ "swap_used_percent"
  ]
  },
  "disk": {
```

To permit reencryption from or to a KMS key, include the `"kms:ReEncrypt"` permission in your key policy. This permission is automatically included in the key policy when you use the console to create a KMS key. But you must include it manually when you create a KMS key programmatically or when you use the `PutKeyPolicy` operation to set a key policy.

9.3 The server pushes the notification on every 5 minutes of interval to the authorized admin. Push notification services, such as APNs and FCM, maintain a connection with each app and associated mobile device registered to use their service. When an app and mobile device register, the push notification service returns a device token. Amazon SNS uses the device token to create a mobile endpoint, to which it can send direct push notification messages. In order for Amazon SNS to communicate with the different push notification.

CHAPTER – 10

FUTURE WORK AND CONCLUSION

10.1. FUTURE WORK

Amazon Web Services (AWS) provides a variety of services to help businesses manage their cloud infrastructure. One such service is the ability to send notifications via email from an Elastic Compute Cloud (EC2) instance. This can be useful for a variety of purposes, such as sending alerts when a certain threshold is exceeded, or notifying administrators when a critical process has completed. There are a few best practices to keep in mind when setting up email notifications from an EC2 instance. First, it's important to ensure that your SES account is properly secured. This means using strong passwords and enabling multi-factor authentication (MFA) to prevent unauthorized access. Additionally, you should use SSL/TLS encryption when sending emails to ensure that they are not intercepted or tampered with. Another best practice is to limit the number of emails that are sent from your EC2 instance. This can help prevent your SES account from being flagged as a spammer, which can result in your emails being blocked by recipient email servers. To limit the number of emails sent, you can use a rate limiter or a queueing system to ensure that emails are sent at a controlled pace. Overall, setting up email notifications from an EC2 instance using AWS SES is a straightforward process that can be extremely useful for a variety of purposes. By following best practices and ensuring that your SES account is properly secured, you can ensure that your notifications are delivered reliably and securely.

10.2. CONCLUSION

In conclusion, AWS push notifications of email from EC2 provide numerous benefits to businesses and individuals using the AWS platform. These notifications offer real-time alerts and notifications, making it easy to identify and address issues quickly, which can help to minimize downtime and improve overall reliability. The notifications are easy to configure and highly customizable, allowing users to tailor them to their specific needs and preferences. Push notifications are also cost-effective, meaning that businesses of all sizes can implement and use them without having to worry about high costs. Furthermore, push notifications can improve productivity by reducing the time and effort required to troubleshoot and resolve issues. Overall, AWS push notifications of email from EC2 are a powerful tool that can help businesses stay informed and responsive, improving their overall performance and success in the cloud.

CHAPTER 11

REFERENCES

- 1) AWS Simple Notification Service (SNS) Documentation:
<https://docs.aws.amazon.com/sns/index.html>
- 2) Amazon EC2 User Guide for Linux Instances:
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
- 3) AWS Lambda Documentation:
<https://docs.aws.amazon.com/lambda/index.html>
- 4) Amazon S3 Developer Guide:
<https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>
- 5) Amazon CloudWatch Documentation:
<https://docs.aws.amazon.com/cloudwatch/index.html>
- 6) AWS Identity and Access Management (IAM) Documentation:
<https://docs.aws.amazon.com/iam/index.html>
- 7) Amazon Route 53 Developer Guide:
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>
- 8) Amazon RDS Documentation:
<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>
- 9) AWS Elastic Beanstalk Documentation:
<https://docs.aws.amazon.com/elasticbeanstalk/index.html>

10) [tps://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html)

11) AWS CloudFormation Documentation: <https://docs.aws.amazon.com/cloudformation/index.html>

12) Amazon SQS Developer Guide:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/Welcome.html>

13) AWS SDK for Java Documentation: <https://docs.aws.amazon.com/sdk-for-java/index.html>

14) Amazon SNS Developer Guide:

<https://docs.aws.amazon.com/sns/latest/dg/Welcome.html>

15) AWS SDK for JavaScript Documentation: <https://docs.aws.amazon.com/sdk-for-javascript/index.html>

16) Amazon API Gateway Developer Guide: <https://docs.aws.amazon.com/apigateway/index.html>

17) AWS architecture: https://aws.amazon.com/architecture/?cards-all.sort-by=item.additionalFields.sortDate&cards-all.sort-order=desc&awsf.content-type=*all&awsf.methodology=*all&awsf.tech-category=*all&awsf.industries=*all&awsf.business-category=*all

18) <https://aws.amazon.com/ec2/faqs/>

19) https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html