A

Cloud Computing and Wireless Sensor Networks

Lab Project

Report on

CV Builder Website Deployment on AWS EC2 Instance

Submitted to

The Department of Computer Science and Engineering(Internet of Things)

Bachelor of Technology

in

Computer Science and Engineering(Internet of Things)

(2021 – 2025)

A.PALLAVI 21R11A6901

J.NIVEDITHA 21R11A6928

K.SAI KARTHIK 20R11A6949

Under the Guidance of

Mr.B.RAMAVATH

Department of Computer Science and Engineering(Internet of Things)

Geethanjali College of Engineering and Technology (UGC Autonomous)

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi, Accredited by NBA)

Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

June-2024

CONTENTS

ABSTRACT

Deploying a CV Builder website on AWS EC2 involves setting up a scalable and secure environment for managing professional resumes. Starting with selecting an appropriate EC2 instance type and configuring security groups, the process moves to installing essential software like Apache/Nginx for web serving and MySQL/PostgreSQL for database management. Uploading and configuring the CV Builder application on the instance follows, ensuring seamless integration with the database for storing user CV data securely.Security is a priority throughout, with SSL/TLS certificates enabling HTTPS, firewall rules restricting access, and adherence to best practices for server and application security. Thorough testing validates the functionality of resume creation, editing, and downloading. Monitoring tools like AWS CloudWatch ensure performance metrics are tracked, while automated backups mitigate data loss risks.Deploying on AWS EC2 provides scalability to handle varying user demands, reliability through AWS infrastructure, and ease of management with scalable resources. This abstract encapsulates the essential steps and benefits of deploying a CV Builder website on AWS EC2, highlighting its role in delivering a secure and efficient platform for users to manage their professional profiles effectively.

INTRODUCTION

1.1 Objective

The objective is to deploy a CV Builder website on AWS EC2, leveraging its scalability, reliability, and security features. This deployment aims to provide users with a robust platform for creating, editing, and downloading resumes seamlessly. By utilizing AWS EC2, the goal is to ensure high availability, efficient resource management, and adherence to best practices in security and scalability. Ultimately, the project aims to deliver a user-friendly CV creation experience while maintaining data integrity and security, supported by AWS's cloud infrastructure capabilities.

1.2 The Domain

Cloud computing is a rapidly growing domain in the field of technology and information systems. It refers to the delivery of computing resources over the internet, allowing users to access and utilize a wide range of services and applications without the need for on-premises infrastructure.

In the cloud computing domain, various types of services are offered, including:

1. Infrastructure as a Service (IaaS): This service provides virtualized computing resources such as virtual machines, storage, and networks. Users can manage and control these resources while avoiding the need for physical hardware.

2. Platform as a Service (PaaS): PaaS provides a platform for developing, running, and managing applications without the complexity of infrastructure management. It offers a complete development and deployment environment, including tools, libraries, and frameworks.

3. Software as a Service (SaaS): SaaS delivers software applications over the internet on a subscription basis. Users can access and use these applications through a web browser without the need for local installation or maintenance.

Cloud computing offers several key advantages:

1. Scalability: Cloud services can scale up or down based on demand, allowing organizations to easily adjust their computing resources to match their needs. This flexibility ensures optimal resource allocation and costefficiency.

2. Cost Savings: By eliminating the need for on-premises hardware and infrastructure, organizations can significantly reduce their capital and operational expenses. Cloud computing services often follow a pay-asyou-go model, where users only pay for the resources they consume.

3. Accessibility and Mobility: Cloud services can be accessed from anywhere with an internet connection, enabling users to work remotely and access their applications and data on various devices.

4. Reliability and Redundancy: Cloud providers typically have robust infrastructure with multiple data centers, ensuring high availability and redundancy. This minimizes the risk of data loss or service disruptions.

However, there are also some considerations and challenges in the cloud computing domain:
1. Security and Privacy: Storing data and applications in the cloud raises concerns about data security, privacy, and compliance with regulations. Organizations must implement proper security measures, encryption, and access controls to protect sensitive information.

2. Connectivity and Reliance on the Internet: Cloud services depend on internet connectivity, and any disruptions can impact accessibility and productivity. Organizations must have reliable internet connections and contingency plans for potential outages.

3. Vendor Lock-In: Moving applications and data to the cloud can create dependencies on specific cloud providers and their proprietary technologies. This can limit flexibility and make it challenging to switch providers in the future.

4. Data Transfer and Bandwidth Costs: Transferring large amounts of data to and from the cloud can incur additional costs and require sufficient bandwidth. Organizations need to consider these factors when planning their cloud computing strategies.

The cloud computing domain continues to evolve, with advancements in technologies such as serverless computing, edge computing, and hybrid cloud architectures. These innovations provide even more options and possibilities for organizations to leverage cloud services and optimize their IT infrastructure.

1.3 The Problem

In this final project, you have created a highly available (HA), scalable and fault-tolerant deployment of the WordPress application. We deployed the WordPress application in such a way that the application server, load balancer and database will scale independently of one another. We also deploy the application's components like the webserver and database into two availability zones to distribute it and guard against failure of the anyone availability zone. The WordPress application will be deployed in a stateless fashion so that we can add or remove web application servers in response to the requests flowing into the system. Finally, we create a CloudFront distribution as CDN and change the configuration of WordPress.

1.4 The Technology

AMAZON WEB SERVICES

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing web services provide distributed computing processing capacity and software tools via AWS server farms. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet. AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded

application software such as web servers, databases, and customer relationship management (CRM).

Scalability: AWS offers elastic computing resources, allowing e-learning platforms to scale their infrastructure up or down based on demand. This flexibility ensures that the platform can handle varying levels of traffic and accommodate a growing user base without any disruptions.

Reliability: AWS operates on a global infrastructure that is designed to deliver high availability and uptime. Its data centers are distributed across different regions, providing redundancy and ensuring that the e-learning platform remains accessible even in the event of hardware failures or natural disasters.

Cost-Effectiveness: With AWS, e-learning platforms can optimize their infrastructure costs. AWS offers a pay-as-you-go pricing model, allowing organizations to pay only for the resources they consume. Additionally, AWS provides various cost management tools to monitor and control spending, ensuring cost efficiency.

Security: AWS implements robust security measures to protect the data and applications hosted on its platform. It offers features such as encryption, identity and access management, and network security controls, ensuring that e-learning platforms can maintain the confidentiality, integrity, and availability of their content and user information.

WordPress

WordPress is a popular and widely used content management system (CMS) that enables users to create and manage websites and blogs with ease. It was initially developed as a blogging platform but has evolved into a versatile CMS that powers a significant portion of the internet. WordPress offers a user-friendly interface, a wide range of themes and plugins, and extensive customization options, making it suitable for various types of websites. WordPress takes security seriously and offers several features and best practices to ensure website security. It regularly releases updates and patches to address vulnerabilities, and users are encouraged to keep their WordPress installations, themes, and plugins up to date. Additionally, users can implement security plugins, use strong passwords, enable two-factor authentication, and follow other security practices to enhance the security of their WordPress websites.

EXAMPLE WITH EXPLANATION

Our project involves deploying a CV Builder website on AWS EC2 to facilitate seamless resume creation and management. AWS EC2 is chosen for its scalability, which allows us to handle varying levels of user traffic efficiently. The website will feature a user-friendly interface where users can input their professional details, edit their resumes, and download them in multiple formats.

Explanation:

1. **AWS EC2 Selection**: We opted for AWS EC2 due to its ability to scale horizontally and vertically based on demand. This ensures our CV Builder website remains responsive even during peak usage times, providing a consistent user experience.
2. **CV Builder Features**: The website will include intuitive form fields for users to input their educational background, work experience, skills, and other relevant information. Users can dynamically edit their CVs and preview changes before finalizing and downloading them.
3. **Security and Reliability**: AWS EC2's security features, such as customizable firewall settings and the ability to use SSL certificates for secure connections, ensure that user data remains protected. Regular backups and monitoring through AWS CloudWatch further enhance reliability and data integrity.
4. **User Experience**: The focus is on creating a straightforward and efficient user experience. The CV Builder interface is designed to be intuitive, guiding users through the resume creation process with ease while providing robust editing and formatting options.
5. **Scalability and Performance**: By deploying on AWS EC2, we can easily adjust compute resources to match user demand, ensuring the website remains responsive and accessible at all times.

In conclusion, deploying a CV Builder on AWS EC2 enables us to provide a reliable, secure, and scalable platform for users to manage their professional profiles effectively.

SYSTEM CONFIGURATION

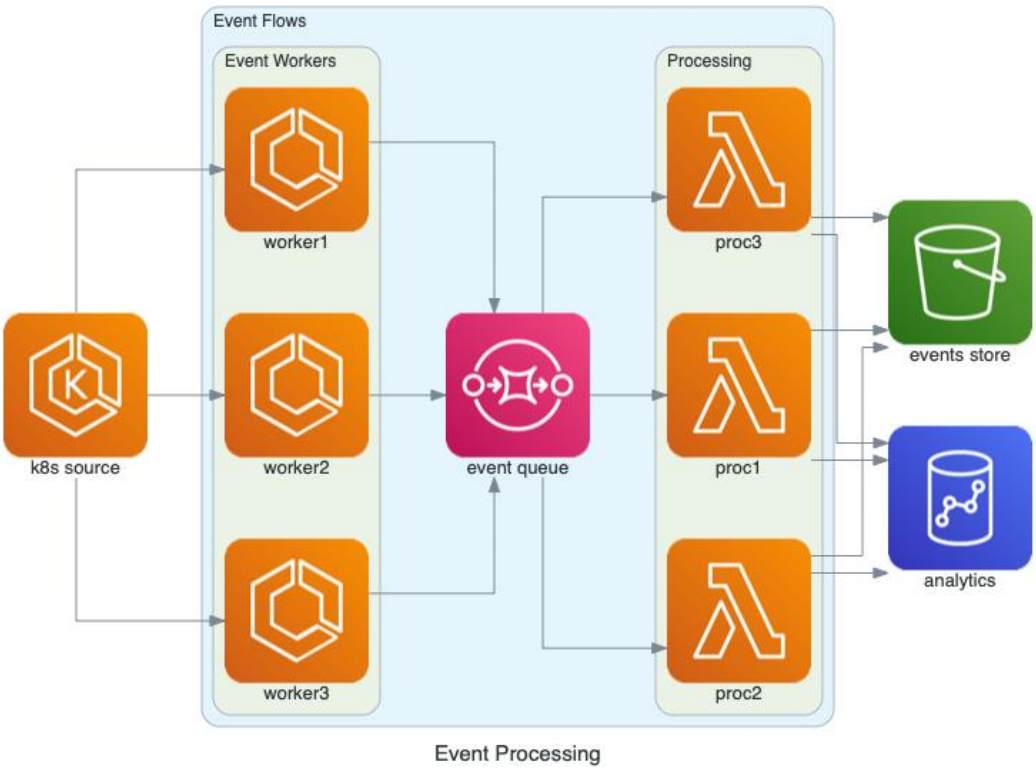
Hardware Specifications:

1. Instance Type Selection
2. Compute Power Allocation
3. Storage Configuration (Amazon EBS)
4. Networking Setup

Software Specifications:

1. Operating System Choice
2. Web Server Installation (Apache/Nginx)
3. Database Management System (MySQL/PostgreSQL)
4. Programming Language Support
5. Security Implementation (SSL/TLS, Firewall)
6. Monitoring and Management Tools

Architecture Diagram



CODE/IMPLEMENTATION

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
crossorigin="anonymous">
    <link rel="stylesheet" href="style.css"/>
    <title>Online Resume Builder</title>
</head>
<body>
    <!-- here we have to write html code -->

    <div class ="container" id="cv-form">
        <h1 class="text-center" my-2>Online Resume Builder</h1>
        <div class ="row">
            <div class="col-md-6">
                <!-- first column -->
                <h3>Personal Information</h3>

                <div class="form-group">
                    <label for="nameField">Your Name</label>
                    <input type="text" id="nameField" placeholder="Enter here" class="form-control"/>
                </div>

                <div class="form-group mt-2">
                    <label for="contactField">Your Contact</label>
```

```
            <input type="text" id="contactField" placeholder="Enter here" class="form-control"/>
        </div>

        <div class="form-group mt-2">
            <label for="addressField">Your Address</label>
            <textarea id="addressField" placeholder="Enter here" class="form-control"
rows="6"></textarea>
        </div>

        <p class ="text-secondary my-3">Important Links</p>

        <div class="form-group mt-2">
            <label for="fbField">Facebook</label>
            <input type="text" id="fbField" placeholder="Enter here" class="form-control"/>
        </div>

        <div class="form-group mt-2">
            <label for="instaField">Instagram</label>
            <input type="text" id="instaField" placeholder="Enter here" class="form-control"/>
        </div>

        <div class="form-group mt-2">
            <label for="linkedField">LinkedIn</label>
            <input type="text" id="linkedField" placeholder="Enter here" class="form-control"/>
        </div>

    </div>
    <div class="col-md-6">
        <!--second column-->
        <h3>Professional Information</h3>
```

```
<div class="form-group mt-2">
    <label for="">Objective</label>
    <textarea id="ObjectiveField" rows="5" placeholder="Enter here" class="form-
control"></textarea>
</div>

<div class="form-group mt-2" id="we">
    <label for="">Work Experience</label>
    <textarea placeholder="Enter here" class="form-control weField"
rows="3"></textarea>

    <!--new text area-->



    <div class="container text-center mt-2" id="weAddButton">
        <button onclick="addNewWEField()"class="btn btn-primary btn-sm">Add</button>
    </div>
</div>

<div class="form-group mt-2" id="aq">
    <label for="">Academic Qualification</label>
    <textarea placeholder="Enter here" class="form-control eqField"
rows="3"></textarea>

    <div class="container text-center mt-2" id="aqAddButton">
        <button onclick="addNewAQField()"class="btn btn-primary btn-sm">Add</button>
    </div>
</div>
```

```
        </div>



    </div>

    <div class="container text-center mt-3">
        <button onclick="generateCV()" class ="btn btn-primary btn-lg">Generate CV</button>
    </div>
</div>


<!--cv template-->

<div class="container" id="cv-template">

    <div class="row">

        <div class="col-md-4 text-center py-5 background">
            <!--first column-->
            <img
src="https://static.vecteezy.com/system/resources/previews/026/619/142/non_2x/default-avatar-profile-icon-of-social-media-user-photo-image-vector.jpg" alt="" class="img-fluid myimg"/>

            <div class="container">
                <p id="nameT1">Shruti Jadhav</p>
                <p id="contactT">+919768594737,+917875341042</p>
                <p id="addressT">B/603, Sadhguru Heights 68 Mumbai, INDIA</p>

                <hr />
```

```
            <p id="fbT"><a
href="#1">https:www.facebook.com/LearnCodeWithDurgesh</a></p>
            <p id="instaT"><a href="#1">https:www.instagram.com/Durgesh_k_t</a></p>
            <p id="linkedT"><a
href="#1">https:www.linkedin.com/LearnCodeWithDurgesh</a></p>
        </div>
    </div>
    <div class="col-md-8 py-5">
        <!--second column-->
        <h1 id="nameT2" class="text-center" style="font-weight: 980">Shruti Jadhav</h1>

        <!--objective card-->

        <div class="card mt-4">
            <div class="card-header background">
                <h3>Objective</h3>
            </div>
            <div class="card-body">
                <p id="objectiveT">Lorem ipsum dolor sit amet consectetur, adipisicing elit. Unde
rerum laboriosam fugiat, accusamus perferendis facere porro praesentium ad dolorum, cupiditate
pariatur aperiam obcaecati atque ratione. Numquam placeat animi velit veniam deserunt, quo
mollitia, maiores, optio eveniet totam in ipsa ea excepturi laudantium voluptatibus porro cum
asperiores. Omnis quae possimus debitis?</p>
            </div>
        </div>

        <!--work experience-->

        <div class="card mt-4">
            <div class="card-header background">
                <h3>Work Experience</h3>
```

```
    </div>
    <div class="card-body">


        <ul id="weT">
            <li>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, fugiat.</li>
            <li>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, fugiat.</li>
            <li>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, fugiat.</li>
        </ul>
    </div>
</div>

<!--academic qualification-->

<div class="card mt-4">
    <div class="card-header background">
        <h3>Academic Qualification</h3>
    </div>
    <div class="card-body">


        <ul id="aqT">
            <li>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, fugiat.</li>
            <li>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, fugiat.</li>
            <li>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quasi, fugiat.</li>
        </ul>
    </div>
</div>

<div class="container mt-3 text-center">
    <button onclick="printCV()" class="btn background">
```

```
                Print CV
              </button>
            </div>


        </div>
      </div>
    </div>
```

SCRIPT.JS

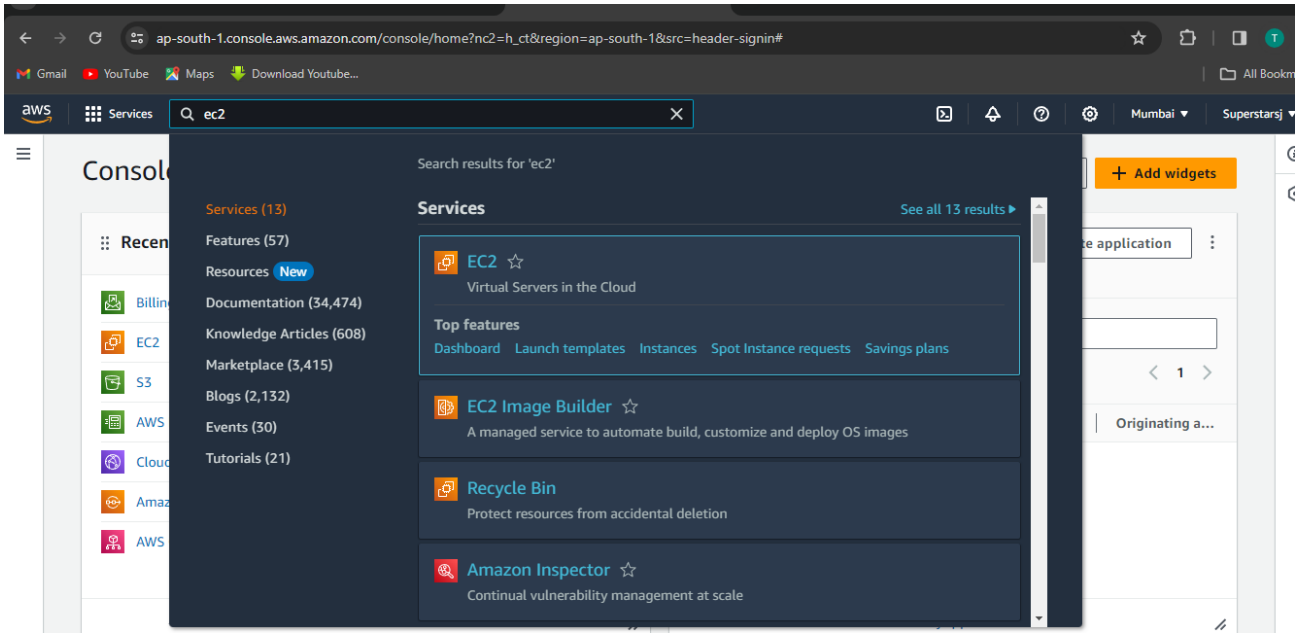
```
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
crossorigin="anonymous"></script>

    <script src="script.js"></script>
</body>
</html>

function addNewWEField(){

    let newNode=document.createElement('textarea');
    newNode.classList.add('form-control');
    newNode.classList.add('weField');
    newNode.classList.add("mt-2");
    newNode.setAttribute("rows",3);
    newNode.setAttribute("placeholder", "Enter Here");

    let weOb=document.getElementById("we");
```

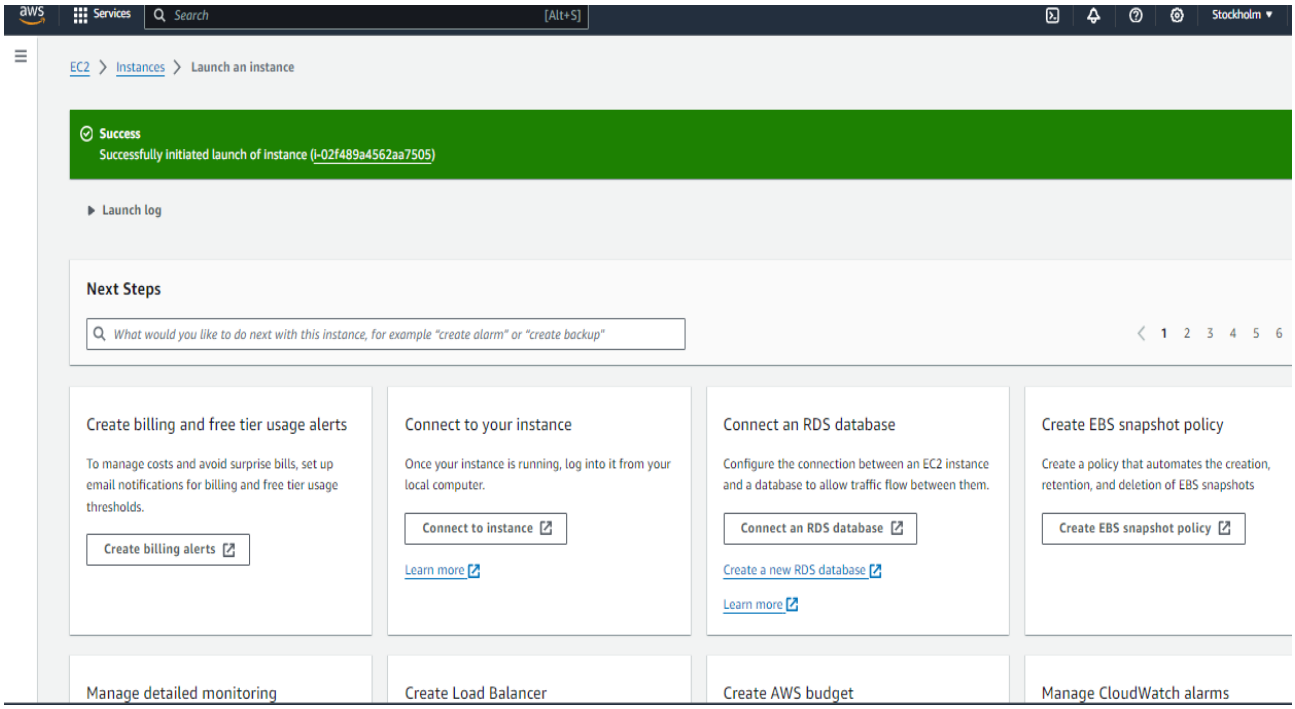
```
    let weAddButtonOb=document.getElementById("weAddButton");

    weOb.insertBefore(newNode, weAddButtonOb);

}

function addNewAQField(){

    let newNode=document.createElement('textarea');
    newNode.classList.add('form-control');
    newNode.classList.add('eqField');
    newNode.classList.add("mt-2");
    newNode.setAttribute("rows",3);
    newNode.setAttribute("placeholder", "Enter Here");

    let aqOb=document.getElementById("aq");
    let aqAddButtonOb=document.getElementById("aqAddButton");

    aqOb.insertBefore(newNode, aqAddButtonOb);


}

//generating cv
function generateCV(){
    let nameField=document.getElementById("nameField").value;

    let nameT1=document.getElementById('nameT1');

    nameT1.innerHTML=nameField;
```

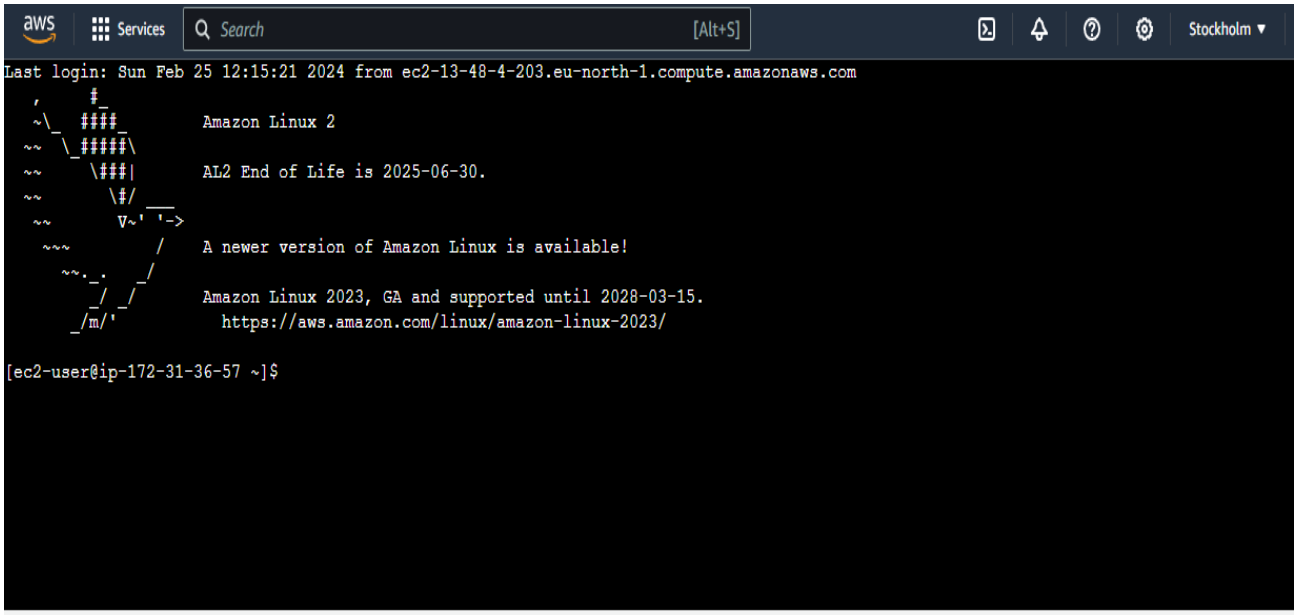
```
    document.getElementById("nameT2").innerHTML= nameField;

    document.getElementById('contactT').innerHTML=
document.getElementById("contactField").value;

    document.getElementById("addressT").innerHTML =
document.getElementById("addressField").value;

    document.getElementById("fbT").innerHTML = document.getElementById("fbField").value;

    document.getElementById("instaT").innerHTML =
document.getElementById("instaField").value;

    document.getElementById("linkedT").innerHTML =
document.getElementById("linkedField").value;

    //objective


document.getElementById('objectiveT').innerHTML=document.getElementById('ObjectiveField').v
alue;

    //we

    let wes=document.getElementsByClassName("weField");

    let str="";

    for(let e of wes)
    {
        str = str + `<li> ${e.value} </li>`;
```

```
    }
    document.getElementById("weT").innerHTML=str;

    //aq

    let aqs = document.getElementsByClassName("eqField");

    let str1 ="";

    for(let e of aqs){
        str1+= `<li> ${e.value} </li>`;

    }

    document.getElementById("aqT").innerHTML=str1;

    document.getElementById("cv-form").style.display ="none";
    document.getElementById("cv-template").style.display = "block";

}

//print cv
function printCV(){
    window.print();
}
```

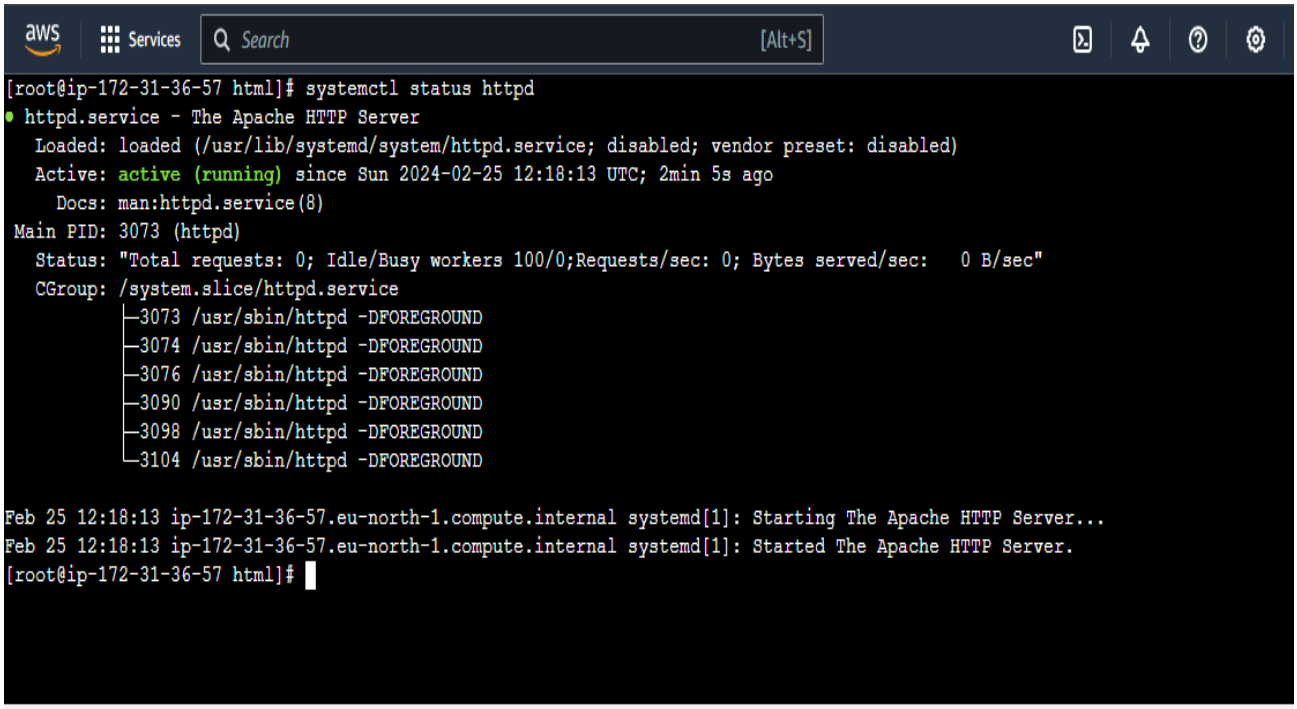
RESULT/OUTPUT SCREENS

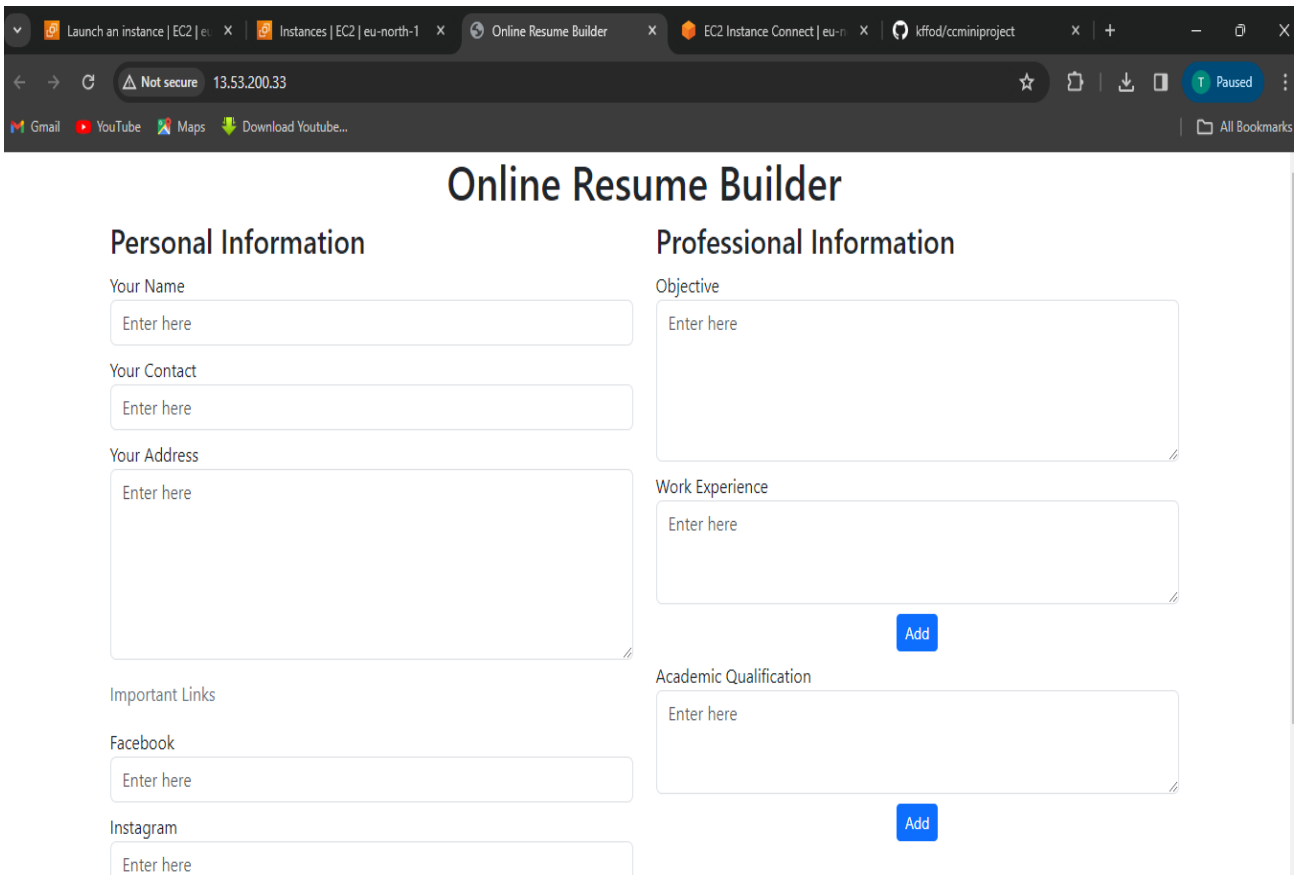







```
No packages marked for update
[root@ip-172-31-36-57 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.58-1.amzn2 will be installed
--> Processing Dependency: httpd-filesystem = 2.4.58-1.amzn2 for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: httpd-tools = 2.4.58-1.amzn2 for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.58-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.58-1.amzn2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.7.2-1.amzn2 will be installed
---> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.58-1.amzn2 will be installed
```


aws
Services
Search
[Alt+S]
[root@ip-172-31-36-57 html]# systemctl status httpd
● httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
Active: active (running) since Sun 2024-02-25 12:18:13 UTC; 2min 5s ago
Docs: man:httpd.service(8)
Main PID: 3073 (httpd)
Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
CGroup: /system.slice/httpd.service
├─3073 /usr/sbin/httpd -DFOREGROUND
├─3074 /usr/sbin/httpd -DFOREGROUND
├─3076 /usr/sbin/httpd -DFOREGROUND
├─3090 /usr/sbin/httpd -DFOREGROUND
├─3098 /usr/sbin/httpd -DFOREGROUND
└─3104 /usr/sbin/httpd -DFOREGROUND
Feb 25 12:18:13 ip-172-31-36-57.eu-north-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Feb 25 12:18:13 ip-172-31-36-57.eu-north-1.compute.internal systemd[1]: Started The Apache HTTP Server.
[root@ip-172-31-36-57 html]#


Launch an instance | EC2 |
Instances | EC2 | eu-north-1
Online Resume Builder
EC2 Instance Connect | eu-n
kffod/ccminiproject
Not secure
13.53.200.33
Paused
Gmail
YouTube
Maps
Download Youtube...
All Bookmarks
Online Resume Builder
Personal Information
Your Name
Enter here
Your Contact
Enter here
Your Address
Enter here
Important Links
Facebook
Enter here
Instagram
Enter here
Professional Information
Objective
Enter here
Work Experience
Enter here
Add
Academic Qualification
Enter here
Add

CONCLUSION

Deploying the CV Builder website on AWS EC2 has provided a robust platform for users to create, edit, and manage their resumes efficiently. Leveraging AWS's scalable infrastructure has ensured high availability and responsiveness to varying user demands. The project successfully integrated essential components such as Apache/Nginx for web serving, MySQL/PostgreSQL for secure data storage, and implemented stringent security measures like SSL/TLS encryption and firewall configurations to safeguard user data.The user-centric design facilitated an intuitive experience, allowing users to dynamically update their CVs and download them in multiple formats with ease. Continuous monitoring through AWS CloudWatch ensured optimal performance, while automated backups guaranteed data integrity and disaster recovery readiness.

BIBILOGRAPHY

- SaiAkash Neela, Yashwanth Neyyala, VamsiNadh Pendem, Kanishk Peryala, Vasantham Vijay Kumar, “Cloud Computing Based Learning Web Application Through Amazon Web Services”, 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)
- https://nevonprojects.com/e-learning-cloud-computing/ https://ieeexplore.ieee.org/document/10101254