NGO Donation Website deployed with Terraform

Major project Report Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by

Rupali Gurjar: (Roll No. 19UCSE4036)

Himadri Apurwa: (Roll No. 19UCSE4024)

Khushi Meena: (Roll NO. 19UCSE4025)

Under the Mentorship of

Dr. Anil Gupta

Professor

&

Under the Guidance of

Dr. Shrawan Ram

Associate Professor

Department of Computer Science and Engineering

MBM University, Jodhpur

<July, 2022>

NGO Donation Website deployed with Terraform

Major Project Report Submitted in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by

Rupali Gurjar: (Roll No.19UCSE4036)

Himadri Apurwa: (Roll No. 19UCSE4024)

Khushi Meena: (Roll NO. 19UCSE4025)

Under the Mentorship of

Dr. Anil Gupta

Professor

&

Under the Guidance of

Dr. Shrawan Ram

Associate Professor

Department of Computer Science and Engineering

MBM University, Jodhpur

<July, 2022>

Department of Computer Science & Engineering

MBM University
Ratanada, Jodhpur, Rajasthan, India –342011

CERTIFICATE

This is to certify that the work contained in this report entitled **"Donation Website For NGO with terraform**" is submitted by the group members. Ms. Rupali Gurjar (Roll. No: 19UCSE4036), Ms Himadri Apurwa (Roll No: 19UCSE4024) and Ms Khushi Meena, (Roll. No: 19UCSE4025) to the Department of Computer Science and Engineering, M.B.M University, Jodhpur, for the partial fulfilment of the requirements for the degree of **Bachelor of Engineering** in **Computer Science and Engineering**.

They have carried out their work under my supervision. This work has not been submitted elsewhere for the award of any other degree or diploma.

The project work in our opinion has reached the standard fulfilling the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering in accordance with the regulations of the Institute.

Dr. Shrawan Ram
Associate Professor
(Guide)
Dept. of Computer Science & Engg.
MBM University, Jodhpur

Prof. N.C. Barwar
(Head)
Dept. of Computer Science & Engg.
MBM University, Jodhpur

ACKNOWLEDGEMENT

Indeed, We have come a long way from conceptualizing this project to giving it the finishing touches. This however would not have been possible without the guidance and the support of the teacher. I want to show gratitude to Professor Anil Gupta sir and Associate Professor ShrawanRam Balach sir for their guidance and suggestions which helped us to complete our project.

I take this opportunity to express deep gratitude towards my teacher, for providing excellent guidance, encouragement and inspiration throughout the project work without his invaluable guidance, this work would never have been a successful one. I wish to express my sincere thanks to all department faculties and staff members for their support. I would also like to thank all my classmates for their valuable guidance and helpful discussion. And at last, I want to thank my college mates without their help, guidance and suggestions it was not possible to produce a project report.

Abstract

We have created a donation website for an NGO. The webpage is made very user friendly so that it can be easily operated by the user. The user can easily make the donation on the site and after successful transaction the receipt of the transaction is received on the email of the sender. Here we have integrated a third party tool-Razorpay as a payment gateway.
For this deployment, we have created a Terraform Module that can deploy any Nodejs Application on the AWS Cloud just using a single command. For the Testing of this module, we have deployed another node js application that is To_Do_List App.

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. It is a platform that offers flexible, reliable, scalable, easy-to-use and cost-effective cloud computing solutions.

Terraform is an open-source infrastructure as code software tool created by HashiCorp. Users define and provide data center infrastructure using a declarative configuration language known as HashiCorp Configuration Language (HCL), or optionally JSON.
Terraform supports a number of cloud infrastructure providers such as Amazon Web Services, Microsoft Azure, IBM Cloud, Serverspace, Google Cloud Platform, DigitalOcean, Oracle Cloud Infrastructure, Yandex. Cloud,VMware vSphere, and OpenStack.

Web development is the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex Web-based Internet applications (Web apps), electronic businesses, and social network services.

Contents

List Of Figures

Chapter 1

INTRODUCTION

1.1 Cloud Computing:

Cloud computing is a general term for anything that involves delivering hosted services over the internet. These services are divided into three main categories or types of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

A cloud can be private or public. A public cloud sells services to anyone on the internet. A private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people, with certain access and permissions settings. Private or public, the goal of cloud computing is to provide easy, scalable access to computing resources and IT services.

Cloud infrastructure involves the hardware and software components required for proper implementation of a cloud computing model. Cloud computing can also be thought of as utility computing or on-demand computing.

How does cloud computing work?

Cloud computing works by enabling client devices to access data and cloud applications over the internet from remote physical servers, databases and computers.

An internet network connection links the front end, which includes the accessing client device, browser, network and cloud software applications, with the back end, which

consists of databases, servers and computers. The back end functions as a repository, storing data that is accessed by the front end.

Communications between the front and back ends are managed by a central server. The central server relies on protocols to facilitate the exchange of data. The central server uses both software and middleware to manage connectivity between different client devices and cloud servers. Typically, there is a dedicated server for each individual application or workload.

Cloud computing relies heavily on virtualization and automation technologies. Virtualization enables the easy abstraction and provisioning of services and underlying cloud systems into logical entities that users can request and utilize. Automation and accompanying orchestration capabilities provide users with a high degree of self-service to provision resources, connect services and deploy workloads without direct intervention from the cloud provider's IT staff.

Types of cloud computing services

Cloud computing can be separated into three general service delivery categories or forms of cloud computing:

1. **IaaS.** IaaS providers, such as Amazon Web Services (AWS), supply a virtual server instance and storage, as well as application programming interfaces (APIs) that let users migrate workloads to a virtual machine (VM). Users have an allocated storage capacity and can start, stop, access and configure the VM and storage as desired. IaaS providers offer small, medium, large, extra-large, and memory- or compute-optimized instances, in addition to enabling customization of instances, for various workload needs. The IaaS cloud model is closest to a remote data center for business users.

2. **PaaS.** In the PaaS model, cloud providers host development tools on their infrastructures. Users access these tools over the internet using APIs, web portals or gateway software. PaaS is used for general software development, and many PaaS providers host the software after it's developed. Common PaaS products include Salesforce's Lightning Platform, AWS Elastic Beanstalk and Google App Engine.

3. **SaaS.** SaaS is a distribution model that delivers software applications over the internet; these applications are often called *web services*. Users can access SaaS applications and services from any location using a computer or mobile device that has internet access. In the SaaS model, users gain access to application software and databases. One common example of a SaaS application is Microsoft 365 for productivity and email services.

1.2 Automation :

Automation is a term for technology applications where human input is minimized. This includes business process automation (BPA), IT automation, personal applications such as home automation and more.

Types of automation

1) **Basic automation**: Basic automation takes simple, rudimentary tasks and automates them. This level of automation is about digitizing work by using tools to streamline and centralize routine tasks, such as using a shared messaging system instead of having information in disconnected silos. Business process management (BPM) and robotic process automation (RPA) are types of basic automation.

2) **Process automation**: Process automation manages business processes for uniformity and transparency. It is typically handled by dedicated software and business apps. Using process automation can increase productivity and efficiency within your business. It can also deliver new insights into business

challenges and suggest solutions. Process mining and workflow automation are types of process automation.

3) **Integration automation**: Integration automation is where machines can mimic human tasks and repeat the actions once humans define the machine rules. One example is the “digital worker.” In recent years, people have defined digital workers as software robots that are trained to work with humans to perform specific tasks. They have a specific set of skills, and they can be “hired” to work on teams.

4) **Artificial intelligence (AI) automation**: The most complex level of automation is artificial intelligence (AI) automation. The addition of AI means that machines can “learn” and make decisions based on past situations they have encountered and analyzed. For example, in customer service, virtual assistants powered can reduce costs while empowering both customers and human agents, creating an optimal customer service experience.

1.3 Web Development :

Web development is the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex Web-based Internet applications (Web apps), electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering, Web design, Web content development, client liaison, client-side/server-side scripting, Web server and network security configuration, and e-commerce development.

Among Web professionals, "Web development" usually refers to the main non-design aspects of building Web sites: writing markup and coding. Web development may use content management systems (CMS) to make content changes easier and available with basic technical skills.

For larger organizations and businesses, Web development teams can consist of hundreds of people (Web developers) and follow standard methods like Agile methodologies while developing Web sites. Smaller organizations may only require a single permanent or contracting developer, or secondary assignment to related job positions such as a graphic designer or information systems technician. Web development may be a collaborative effort between departments rather than the domain of a designated department. There are three kinds of Web developer specialization: front-end developer, back-end developer, and full-stack developer.

Front-end developers are responsible for behavior and visuals that run in the user browser, while back-end developers deal with the servers

1.1.1 **Software Used :**

- **Node** is an open-source, cross-platform, back-end Javascript engine that runs on the V8 engine and executes JavaScript code outside a web-browser. Node.js lets developers use JavaScript to write command-line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.
- **Bootstrap** Bootstrap is a free and open-source CSS framework directed at responsive, Mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is the seventh-most-starred project on GitHub, with more than 142,000 stars, behind freeCodeCamp (almost 312,000 stars) and marginally behind Vue.js framework.
- **Source-Code Management** GIT and GitHub. Source code management is a less-maddening way of managing changes across a project. It lets you track the entire history of your project in logical iterations called revisions (which are termed by individual SCM software as changesets, or changelists.) Initially when we started to learn how to use these version controls like GIT and GitHub, we improved our skills in

their basics. operations like how to initialise a repository, committing codes, branching, merging and creating a pull request.

- **Networking** : HTTP-The protocol that is used for transferring information between servers and clients. We learned the basic architecture and working of computer networking and gained some knowledge of various protocols, IP addresses, port number, structure behind client- server connection, request – response etc.
- Payment gateway integration by RAZORPAY

Chapter 2

TECHNOLOGY USED

These are the technologies that we have used in this project.

2.1 **AWS Cloud**

The full form of AWS is Amazon Web Services. It is a platform that offers flexible, reliable, scalable, easy-to-use and cost-effective cloud computing solutions.
AWS is a comprehensive, easy to use computing platform offered by Amazon. The platform is developed with a combination of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

Important AWS Services

Amazon Web Services offers a wide range of different business purpose global cloud-based products. The products include storage, databases, analytics, networking, mobile, development tools, enterprise applications, with a pay-as-you-go pricing model.

Some of the services that we have used in the projects are:

1) **EC2(Elastic Compute Cloud)-** EC2 is a virtual machine in the cloud on which you have OS level control. You can run this cloud server whenever you want.
2) **Amazon Elastic Block Store (EBS)-** It provides block-level storage to use with Amazon EC2 instances. Amazon Elastic Block Store volumes are network-attached and remain independent from the life of an instance.

3) **IAM (Identity and Access Management)**— IAM is a secure cloud security service which helps you to manage users, assign policies, form groups to manage multiple users.

Advantages of AWS

Following are the pros of using AWS services:

- AWS allows organizations to use the already familiar programming models, operating systems, databases, and architectures.
- It is a cost-effective service that allows you to pay only for what you use, without any up-front or long-term commitments.
- You will not require to spend money on running and maintaining data centers.
- Offers fast deployments
- You can easily add or remove capacity.
- You are allowed cloud access quickly with limitless capacity.
- Total Cost of Ownership is very low compared to any private/dedicated servers.
- Offers Centralized Billing and management
- Offers Hybrid Capabilities
- Allows you to deploy your application in multiple regions around the world with just a few clicks

Disadvantages of AWS

- If you need more immediate or intensive assistance, you'll have to opt for paid support packages.
- Amazon Web Services may have some common cloud computing issues when you move to a cloud. For example, downtime, limited control, and backup protection.
- AWS sets default limits on resources which differ from region to region. These resources consist of images, volumes, and snapshots.
- Hardware-level changes happen to your application which may not offer the best performance and usage of your applications.

2.2 **Terraform**

Terraform is an open-source infrastructure as code software tool created by HashiCorp. Users define and provide data center infrastructure using a declarative configuration language known as HashiCorp Configuration Language (HCL), or optionally JSON.

Terraform manages external resources (such as public cloud infrastructure, private cloud infrastructure, network appliances, software as a service, and platform as a service) with "providers". HashiCorp maintains an extensive list of official providers, and can also integrate with community-developed providers.Users can interact with Terraform providers by declaring resources or by calling data sources. Rather than using imperative commands to provision resources, Terraform uses declarative configuration to describe the desired final state. Once a user invokes Terraform on a given resource, Terraform will perform CRUD actions on the user's behalf to accomplish the desired state. The infrastructure as code can be written as modules, promoting reusability and maintainability.

Terraform supports a number of cloud infrastructure providers such as Amazon Web Services, Microsoft Azure, IBM Cloud, Serverspace, Google Cloud Platform, DigitalOcean, Oracle Cloud Infrastructure, Yandex.Cloud VMware vSphere, and OpenStack.

HashiCorp also supports a Terraform Module Registry, launched in 2017. In 2019, Terraform introduced the paid version called Terraform Enterprise for larger organizations.

Terraform has four major commands:

$ terraform init

$ terraform plan

$ terraform apply

$ terraform destroy

2.3 HTML (Front-End Development)

2.3.1 HTML Introduction

HTML stands for HyperText Markup Language. (**HTML**) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML is used to design the structure of a web page. It is a set of instructions on how to display content on a web page. With HTML constructs, images and other objects such as Interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML documents are written in html editors. Since the html document is written in plain , we can use any text editor.

Ex : notepad, notepad++, sublime text, eclipse, netbeans etc.

2.3.2 **HTML Page Structure**

```
<!DOCTYPE html>
<!-
Add your comments here
-->
<html>
<head>
<title>                    </title>
</head>
<body>
```

</body>

</html>

2.3.3 TAGS AND ELEMENTS HTML

TAGS

HTML tags are the hidden keyword within a web page that define how the browser must format and display the contents. Most of these tags have 2 parts , opening tag and closing tag.

 : Opening Tag   : Closing Tag

Opening Tag and Closing Tag together are called as a Container .

- **<html>**

Tags are used to inform the browser how to format or display the text, except for the declaration of document type and comment. Comments can be included with html tags also.

- **<body>**

It contains all the visible contents of the page. It may include texts, links and tables, or videos.

- ** **

It is used to create different sections in a web page.

TEXT FORMATTING

There are six different heading tags.

<h1> I am Heading </h1>

<h2>I am Heading</h2>

Number defines the size of the text.

<i> : Tag to make text look italic.

 : Tag to make text look bold.

: Tag to make text look italic.

These tags help the browser know about the special importance of text , not just styling. It also helps in SEO purposes .

<u> : Tag element is used to underline text.

<mark> : Tag element is used to highlight text.

HYPERLINKS

Any content , image or text can be linked to a new page. The text linked is called anchor text. Link is created using html <a>click herelink.

href=”hypertext reference” click here

target=”_blank”

 This
will open in another tab. target=”_self”
 This will
open in the same tab.

2.4 CASCADING STYLE SHEET

Cascading Style Sheet is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is
a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. There are three ways of styling:

1) Inline
2) Internal
3) External

BOX MODEL

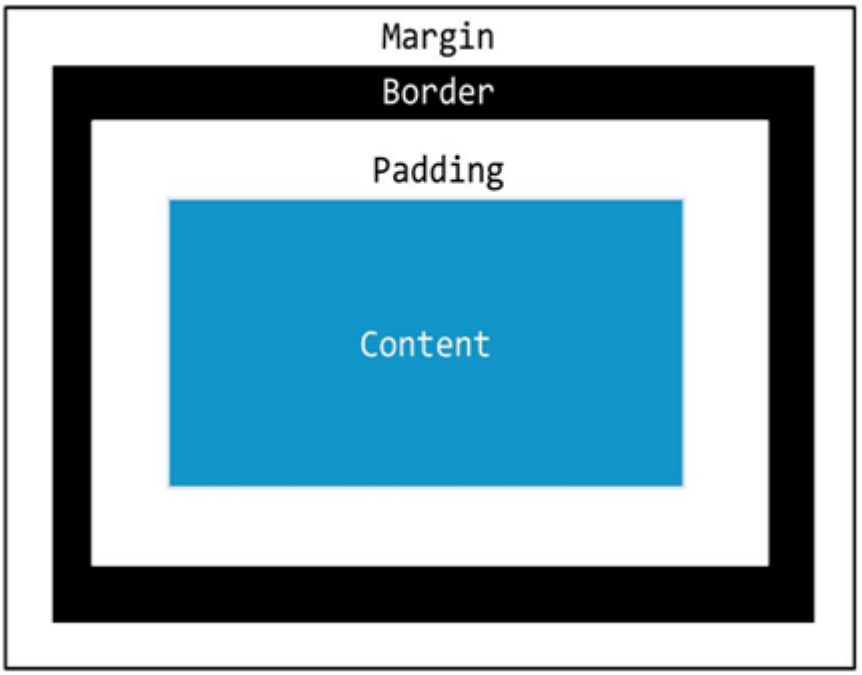

fig 2.4.1 box model

- Each and every element can be considered as a box.
- Innermost box called content contains content which could be image , text.
- Padding clears the area around the content. It is always transparent .
- Padding is surrounded by a border whose colour can be chosen.
- Margin clears the area outside the border. Default margin is 0.Margin is also transparently applied.

FONTS

Properties :

1) Font Style

2) Font Variant

3) Font weight

4) Font Size

5) Line height

6) Font family

Font size and Font family are mandatory.

For Ex : h4 { font: italic small-caps bolder 28px arial, sans-serif ; } Comma is used in font family if there is unavailability of any font.

NAVIGATION BAR

Navigation Bar and logo are together often referred to as a header. Navigation bar links different sections within a page.

Navigation links are added using the <nav> tag. Using the <ul> tag inside the

<nav> tag we can create a navigation bar.

To create logo in navigation bar ;

<img src=”./img/logo.jpg” alt=”msd”

FORMS

It collects information stored in databases or sent to the server. Using the <form> element to create form.

<form>

<input type=”text” placeholder =”Enter first Name” name=”First name”>

<input type=”submit”>

2.5 BOOTSTRAP

2.5.1 BOOTSTRAP INTRODUCTION (FRONT END DEVELOPMENT)

Bootstrap is a free and open-source CSS framework directed at responsive, mobile first front- end web development. It contains CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other components.

Bootstrap is one of the simplest to implement and use in the market. Its implementation is as simple as importing a CSS and using the classes available. Bootstrap's responsiveness makes it all much simpler. It can intelligently sense the device's resolution and screen width and adjust the content accordingly. Bootstrap is supported by the huge open source community present on GitHub. Any bugs or issues are resolved in no time for the releases.

Linking Javascript File(Jquery) :

Javascript is used to make the page more dynamic, to add effects , animation etc. Javascript files in the JS folder of bootstrap are used to add predefined javascript functions in the html code. Script element in html is used to embed script within an html document.

<script type="text/javascript src="bootstrap/js/bootstrap.min.js"></script>

★ **Second method to link Bootstrap file :**

"https://max.edn.bootstrapedn.com/bootstrap/3.3.7/css/bootstrap.min.js">

" https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery/1.12.4/jquery.min.js">

" https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"

BREAKPOINTS

MULTIPLE CLASSES :

It is one of the best ways of defining complex web pages layout and design. At times we have to make minor changes in an element while keeping everything else just the same. This can be achieved using multiple classes on the element we plan to make changes to. Use of multiple classes on an element is a core part of designing web pages using BOOTSTRAP , Bootstrap has a number of predefined classes.

BREAKPOINTS IN BOOTSTRAP :

DEVICES	BREAKPOINT
Large Devices	>=1200px
Medium Devices	992px to 1199px
Small Devices	768px to 991px
Extra Small Devices	<768px

JUMBOTRON AND GLYPHICONS

Jumbotron is a big box for calling extra attention to some special attention content or information. Class Jumbotrons are used to create jumbotrons.

It is represented in grey box and round colors. It also enlarges the text inside it. Jumbotron takes the full width of the element.

<body>

<h1>This is a jumbotron</h1>

<div class=”jumbotron”

We can also insert tables, content, text etc inside a jumbotron. GLYPHICONS : It is the icon displayed on any web page.

<body>

<h3>search</h3>

<h3 search <span class=”glyphicon glyphicon_search”></h3>

THUMBNAILS

Thumbnails are reduced versions of audio and videos. Thumbnails take the available width.

<h1>Thumbnails<h1>

<img src=”img path” alt=”Responsive image”> Adding

<a> tag to make thumbnails clickable. Thumbnails are treated as links using anchor tags.

.thumbnail { padding:4px;

border:1px solid #ddd; border- radius:4px; }
To add content in thumbnail ;

<h3> </h3>

FORMS

FORM STYLING :

<div class=”row”

<h1>Styling forms</h1>

By default form-control class can take the entire width for the element.

<form>

<input type=”text” class=”form-control” name=”first_name placeholder=”first_name”>

.form-group { margin-bottom:5px;

}

To stop users from entering data add disabled attributes.

To make the first name and last name appear above the input field use <label> tag.

<label=”first-name”>First Name</label>

<input type=”text” class=”form-control” name=”first_name”>

STYLING CHECKBOXES AND RADIO BUTTONS :

 I want to order :
<label>

<input type=”checkbox” name=”food” value=”coffee” checked>Coffee

</label>

.label { padding-left:20px; margin- bottom:20px; }

2.6 NODE

Node.js is an open source, a system application and furthermore is an environment for servers. Nodejs is an independent development platform built on Chrome's JavaScript Runtime that we can build network applications quickly and easily. Google V8 JavaScript engine is used by Node.js to execute code. Moreover, a huge proportion of essential modules are written in JavaScript

Node.js accommodate a built-in library which allows applications to serve as a Web-server left out demanding software like Apache HTTP Server, Nginx or IIS.

An event-driven, non-blocking I / O mechanisms (Input / Output) are implemented by Node.js. It optimizes application throughout and is extremely high extensible. Node.js use asynchronous in it functions. Therefore, Node.js processes and executes all tasks in the background (background processing).

products that have a lot of traffic are applying Node.js. Nonetheless, Node.js handle the application that need to spread expeditiously, develop innovation, or build Startup projects as rapidly as possible.

Applications using NodeJS:

- WebSocket server
- Notification system
- Applications that need to upload files on the client
- Other real-time data applications.

NodeJS Pros:

Node.js is the exclusive application that with only a single thread, it can obtain and handle numerous connections. Building new threads for each query is not needed, therefore the structure expends the least amount of RAM and run rap- idly. Secondly, Node.js produces the most of server property without generate latency with the JavaScript's non-blocking I/O.

JSON APIs. JSON Web services can take advantages of that because of the event-driven, non-blocking I/O structures and JavaScript-enabled model.

Single page application. NodeJS is very suitable with an application on a single page. Node.js has the capability to handle different requests concurrent and quick return. Node JS should be used in an application that does not have to reload the page, including users who makes a vast number of requests and need a quick procedure to show professionalism.

Shelling tools Unix. Node.js usually uses Unix to work. They can handle multiple processes and return them for best performance. Programmers often use Node.js to build real Web applications like chat, feeds, etc.

Streaming Data. Typical websites send HTTP requests and also receive re- sponses. Node.js can handle many questions and feedback, so they are suitable if the developer wants to create an application on the page. In addition, Node.js also builds proxies to stream the data, this is to ensure maximum operation for other data streams.

Real-time Web Application. Node.js is sufficient to develop real-time innova- tions like chat apps, social networking services like Facebook, Twitter because of the opening of mobile application.

NodeJS Cons:

Resource-intensive applications. Node.js is written in C ++ & JavaScript, so when programmers need to handle applications that use a lot of file conversion, video encoding, decoding, etc., they should not be used Node.js. Programmers need to use it more carefully in this case.

The final purpose of NodeJS is like other programming languages such as Ruby, PHP, .NET, Python, that is developing web application. Therefore, do not expect NodeJS to outperform other language for now. But with NodeJS the application can be developed successfully as expected. [5]

NodeJS should not be used when:

- **Build resource-intensive applications**: Do not use Node.js when creating a video converter application. Node.js often comes down to bottlenecks when work- ing with large files.
- **An all-CRUD-only application**: Node.js is not faster than PHP when doing heavy I/O tasks. In addition, with the long-term stability of other webserver scripts, its CRUD tasks have been optimized. Node.js will come up with odd APIs and never be used.
- **Stability in the application**: Within 11 years of development (2009-2020), the current version of Node.js is already v14.2.0. Every API can be changed – in a way that is not backwards compatible.
- **Lack of knowledge about Node.js**: Node.js is extremely dangerous in this case, you will fall into a world full of difficulties. With most non-blocking/async APIs, not understanding the problem will cause an error that you do not even know where it came from. Moreover, when the Node.js community is not strong enough, and there will be less support from the community.

NodeJS should be used when:

- **Building RESTful API (JSON)**. You can use Node.js in building RESTful API (JSON). They handle JSON very easily, even more than JavaScript. API servers when using Node.js usually do not have to perform heavy processing, but the number of concurrent requests is high.
- **Applications that demand alternative connection protocols**, not just http. With TCP protocol backing, any custom protocol can be built easily.

- **Real-time applications**.
- **Stateful websites**. Every request on the invariable procedure is handled by Node.js, therefore building caching is simpler: store it to a comprehensive varia- ble then all requests can approach the cache. The status of one client can be saved and shared with other clients and do not have to go through external memory.

Chapter 3

PROJECT/WORK DETAILS

We have made a donation website for an NGO, it uses Razor pay as third-party integration and when the transaction is successful an email of transaction receipt is sent to the email of the donor.

This project is deployed in the real-world environment using AWS cloud for infrastructure and terraform for automation.

This Terraform module can be used to deploy any Node Application.

1) Terraform Module

To launch an aws application using terraform , first we have to tell terraform who is the provider (aws , openstack ,azure etc) and do authentication between provider (here aws) and terraform . So I have created a profile . In this profile I have provided my access key and secret key to login aws .

Here My Profile name is 'Rupali'

```
provider "aws" {

  region = var.region

  profile = var.profile

}
```

We will launch an instance using ec2 service of the aws so we need a key to login that instance .

```
resource "tls_private_key" "this" {
  algorithm = "RSA"
}

module "key_pair" {
  source = "terraform-aws-modules/key-pair/aws"

  key_name   = var.key_name
  public_key = tls_private_key.this.public_key_openssh
}
```

Here I used a resource "tls_private_key" to create the key . It uses the algorithm RSA . It created a key named "deployer-key" .

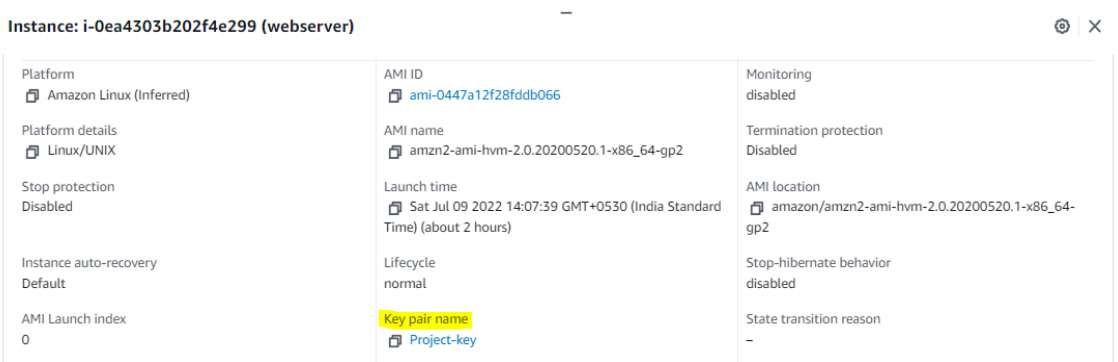

fig 3.1 key-pair

Now we have to create a Security Group .

A security group acts as a virtual firewall for your instance to control incoming and outgoing traffic.

```
resource "aws_security_group" "allow" {
 name        = var.security_group
 description = "Allow TLS inbound traffic"
 vpc_id      =  var.vpc_id


 ingress {
  description = "ssh"
  from_port   = 0
  to_port     = 22
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }
 ingress {
  description = "http"
  from_port   = 0
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }

 ingress {
  description = "https"
  from_port   = 0
  to_port     = 443
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }

 ingress {
  from_port   = 0
  to_port     = 3000
```

```
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }

 ingress {
  from_port   = 0
  to_port     = 5000
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
 }

 egress {
  from_port   = 0
  to_port     = 0
  protocol    = "-1"
  cidr_blocks = ["0.0.0.0/0"]
 }

 tags = {
  Name = var.security_group
 }
}
```

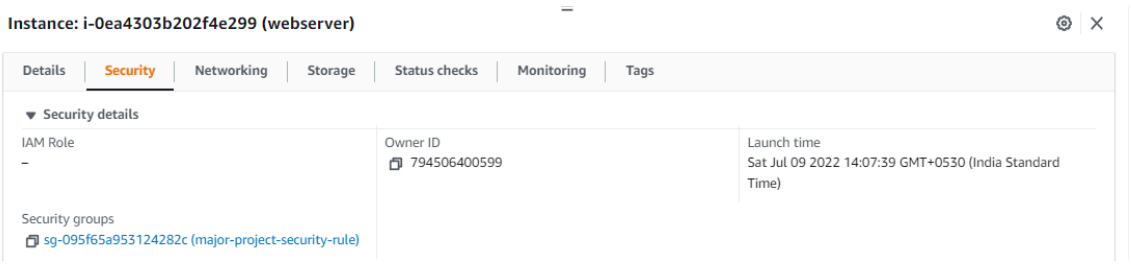

fig 3.2 Security group

To launch an instance using terraform , we used a resource "aws_instance" and provided ami id , instance type , key and security group .

```
resource "aws_instance" "web" {
 ami               = var.ami
 instance_type  = var.instance_type
 key_name       = var.key_name
 vpc_security_group_ids =  [  aws_security_group.allow.id  ]

 connection {
  type     = "ssh"
  user     = "ec2-user"
  private_key = tls_private_key.this.private_key_pem
  host     = aws_instance.web.public_ip
 }

 provisioner "remote-exec" {
  inline = [
   "sudo su <<END",
   "sudo su - root",
   "yum install httpd  php git -y",
   "systemctl restart httpd",
   "systemctl enable httpd",
    "END"
  ]
 }

 tags = {
  Name = var.instance_name
 }
}
```

There are two types of Executors :

1. Local executor : we use this executor whenever we need to run any command in our local system .
2. Remote Executor : To run any command in the remote system.

Here we have used Remote Executor to install httpd service , git and php in the instance that we have launched .

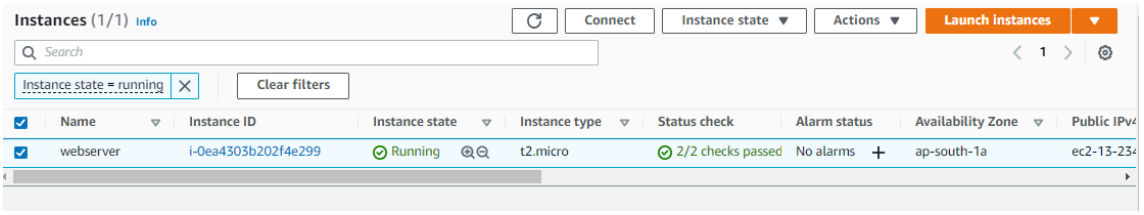

fig 3.3 EC2 Instance

We want to store our data permanently so that it would be safe even after rebooting the system so we have attached an EBS storage with the instance .

```
resource "aws_ebs_volume" "esb1" {
  availability_zone = aws_instance.web.availability_zone
  size              = 1
  tags = {
    Name = var.ebs
  }
}

resource "aws_volume_attachment" "ebs_att" {
  device_name = "/dev/sdh"
  volume_id   =  aws_ebs_volume.esb1.id
  instance_id =  aws_instance.web.id
  force_detach = true
}
```

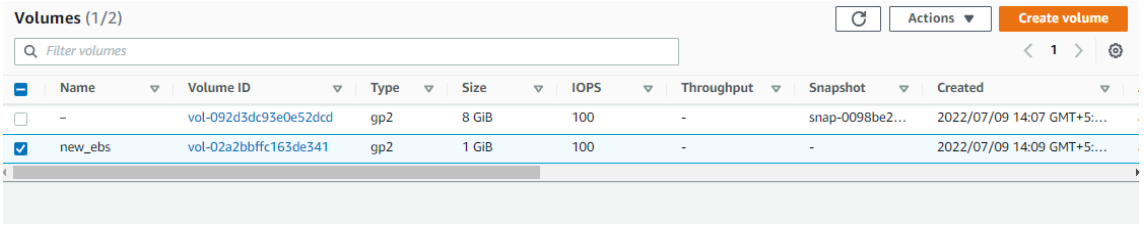

fig 3.4 EBS Volume

We are going to use "appache-webserver" and it keeps all the pages in its default folder "/var/www/html" so I mounted this storage on that folder .

```
resource "null_resource" "nullremote3"  {
depends_on = [
   aws_volume_attachment.ebs_att,
 ]
 connection {
   type     = "ssh"
   user     = "ec2-user"
   private_key = tls_private_key.this.private_key_pem
   host     = aws_instance.web.public_ip
 }

provisioner "remote-exec" {
   inline = [
     "sudo su <<END",
     "sudo su - root",
     "mkfs.ext4  /dev/xvdh",
     "mount  /dev/xvdh  /var/www/html",
     "rm -rf /var/www/html/*",
        "curl  -o-  https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh |
bash",
     ". ~/.nvm/nvm.sh",
     "nvm install --lts",
     "git clone ${var.Github} /var/www/html/",
```

```
    "cd /var/www/html",
    "npm i",
    "node ${var.js_file}",
     "END"
   ]
 }
}
```

I downloaded the code from the github and copy that code in the /var/www/html folder.
Also installed nodejs, npm and all the dependencies.
After that started the node server.

TO DOWNLOAD ALL THE PLUGINS , WE HAVE TO RUN THIS COMMAND

"terraform init "

After it, terraform has been initialized successfully so to run the code we run the command
"terraform apply "

But here we have to write 'yes' manually so instead of using the above command, we run this command.
"terraform apply -auto-approve"

To get the ip of instance , we run this command

```
output "myos_ip" {
  value = aws_instance.web.public_ip
}
```

Chapter 4

RESULTS/OUTCOME

We have successfully deployed an NGO Donation website using Terraform Module. These are some of the pages of our NGO Donation website.

Home page

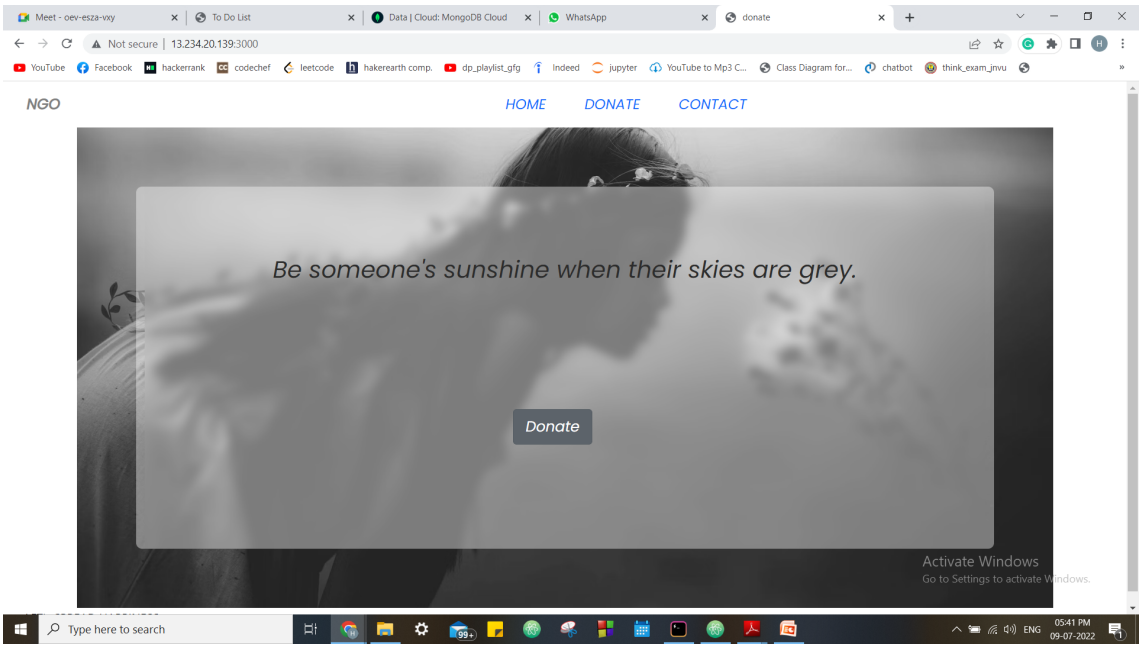

fig 4.1 home page

Transaction Status page

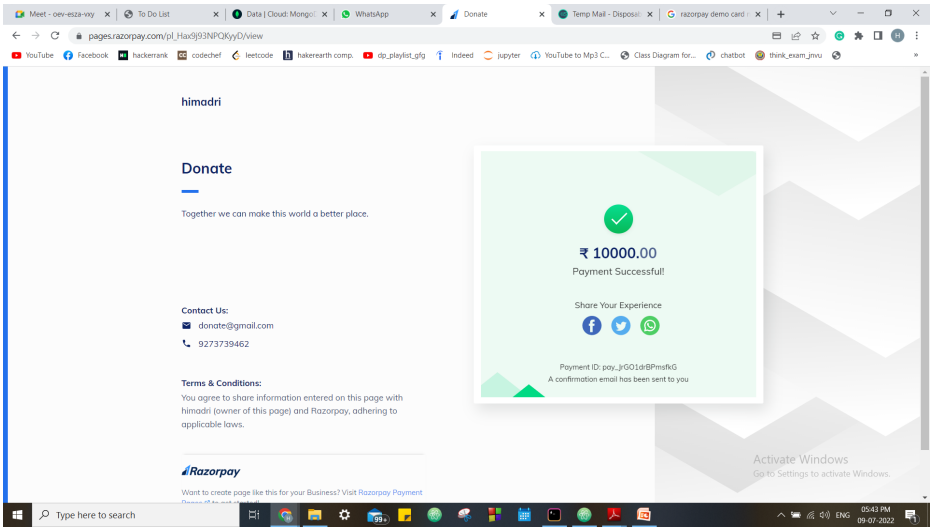

fig 4.2 transaction successful page

Email Transaction Receipt

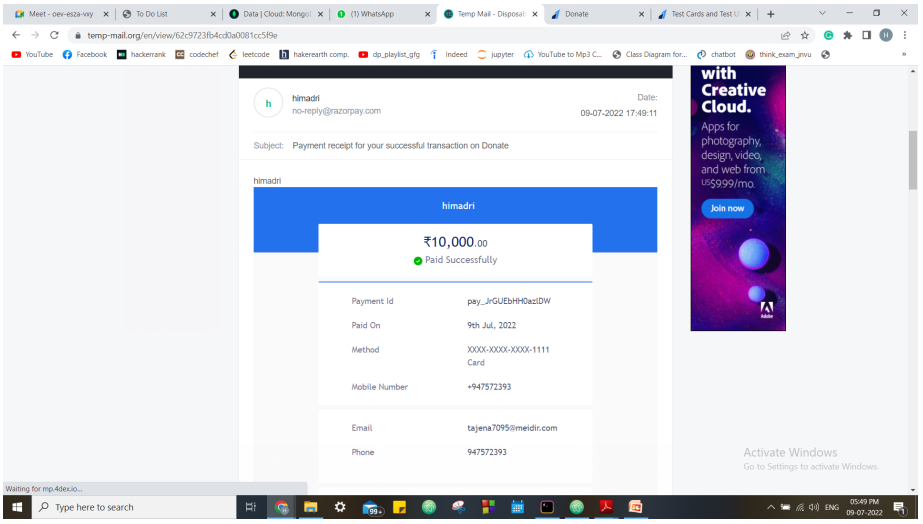

fig 4.3 email receipt

Chapter 5

CONCLUSION & FUTURE WORK

CONCLUSION

we have created a donation website for NGO. The webpage is made every user friendly so that it can be easily operated by the user. The user can easily make the donation on the site and after successful transaction the receipt of the transaction is received on the email of the sender.

We have created a Terraform Module that can deploy any Nodejs Application on the aws Cloud Just using a single command.

Future Work

This terraform Module can be uploaded to the Terraform registry after some enhancement. We can also integrate it with some CI/CD tools like Jenkins for the complete automation

The Web Developer bears the responsibility for the coding, design, and layout of the website according to the company's specifications. A website is more important than anything for every business when it comes to reaching clients online. Every business knows today the need to have a website and is trying to design and create the best website to take its products or services online. With incredible progress in launching websites, businesses are searching for people who can build outstanding designs and platforms for their online presence.Web developers and designers bring their technical skills and experience to the creation and production of exclusive websites capable of attracting the crowd. They are striving to create more reliable sites through all-new frameworks, tools, and advancements.

References

[1] HashiCorp. "HashiCorp Terraform - Provision & Manage any Infrastructure". *HashiCorp: Infrastructure enables innovation*. Retrieved 2020-04-15

[2] Amazon Elastic Compute Cloud Documentation. Available from : https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

[3] Cloud Infrastructure design and management - TechTarget. Available from: https://www.techtarget.com/searchcloudcomputing/definition/cloud-computing

[4] JavaScript [Internet]. Mozilla.org. Available from: https://developer.mozilla.org/en-US/docs/Web/JavaScript

[5] NodeJS Introduction [Internet]. Tutorialspoint.com. Available from : https://www.tutorialspoint.com/nodejs/nodejs_introduction.html

[6] NodeJS Pros and Cons [Internet]. Mindinventory.com. Available from: https://www.mindinventory.com/blog/pros-and-cons-of-node-js-web-app-development/

[7] NodeJS use cases [Internet]. Credencys.com. Available from: https://www.credencys.com/blog/node-js-development-use-cases/