CIT481 Internet Explorers – Group Project Update

Spring 2022

Members: Kristina Utas, Karo Torosyan, Cesar Ramirez, Tim Abad

Github: https://github.com/TimAbad/CIT480SeniorProject.git

Project Updates:

Kristina:

Built last semester using Terraform:

1. VPC
2. Two private and two public subnets across two availability zones
3. Two EC2 instances
4. Routing Tables
5. Security groups
6. ALB
7. Target groups, health checks, rules

Anticipate building this semester using Terraform:

1. Route 53
2. Autoscaling group
3. RDS
4. IAM
5. NAT gateway

Karo:

This semester I plan to refine my ansible monitoring code in an effort to make the automation run on a single command on startup along with working with other parts of our infrastructure. I will also tweak the code to target the ec2 instance instead of localhost. In an effort to make this process easier on our group, I will also write up a readme file to make the process of setting up the monitoring easy for everyone to understand how it works and how to deploy it.

Cesar:

- Create a Playlist on Ansible to manage the configuration of the Apache Server that would hold our Covid-19 application on a EC2 instance on the AWS environment.

- This playbook would be targeted to the EC2 created previously with Terraform.
- It includes the installation and the latest Apache2 version.

- Installation of modules such as re-write for which allow you to make custom and simplified URL as needed.

- Installation of headers module, this allows you to control and modify HTTP request and response headers in Apache. If you try to modify headers in Apache web server without installing mod_headers, it may throw an internal server error.
- Installation of Mod_expires, and by setting expiration periods for web content, enables web browsers to cache content for specific periods of time.

- Installation of Macro module, allows you to set up scripted templates to use for hosting all of your websites. Like functions, they can be called with parameters allowing you the flexibility to set up various scenarios for each website.

- WSGI module implements a simple to use Apache module which can host any Python web application which supports the Python WSGI specification.

- Restart Apache2 after modules are installed

Tim:

1. Successfully transitioned from Flask development to Streamlit development over the winter break.
2. Major work done on the front-end over the winter break. Will demo during first presentation.
3. Work on ansible script to deploy new application to our infrastructure.
4. Work with Karo to integrate application deployment and monitoring in one ansible script.
5. Work on HTTPS using self-signed certificates for our site and automating the process,
6. Possibly look into HTTP->HTTPS forwarding using proxypass modules.