MEet and You

Brent Nishioka (Leader)
Gideon Essel
Joshua Ramos
Raymond Guevara
Vivian Dinh
Team Pentaskilled
October 27th, 2021

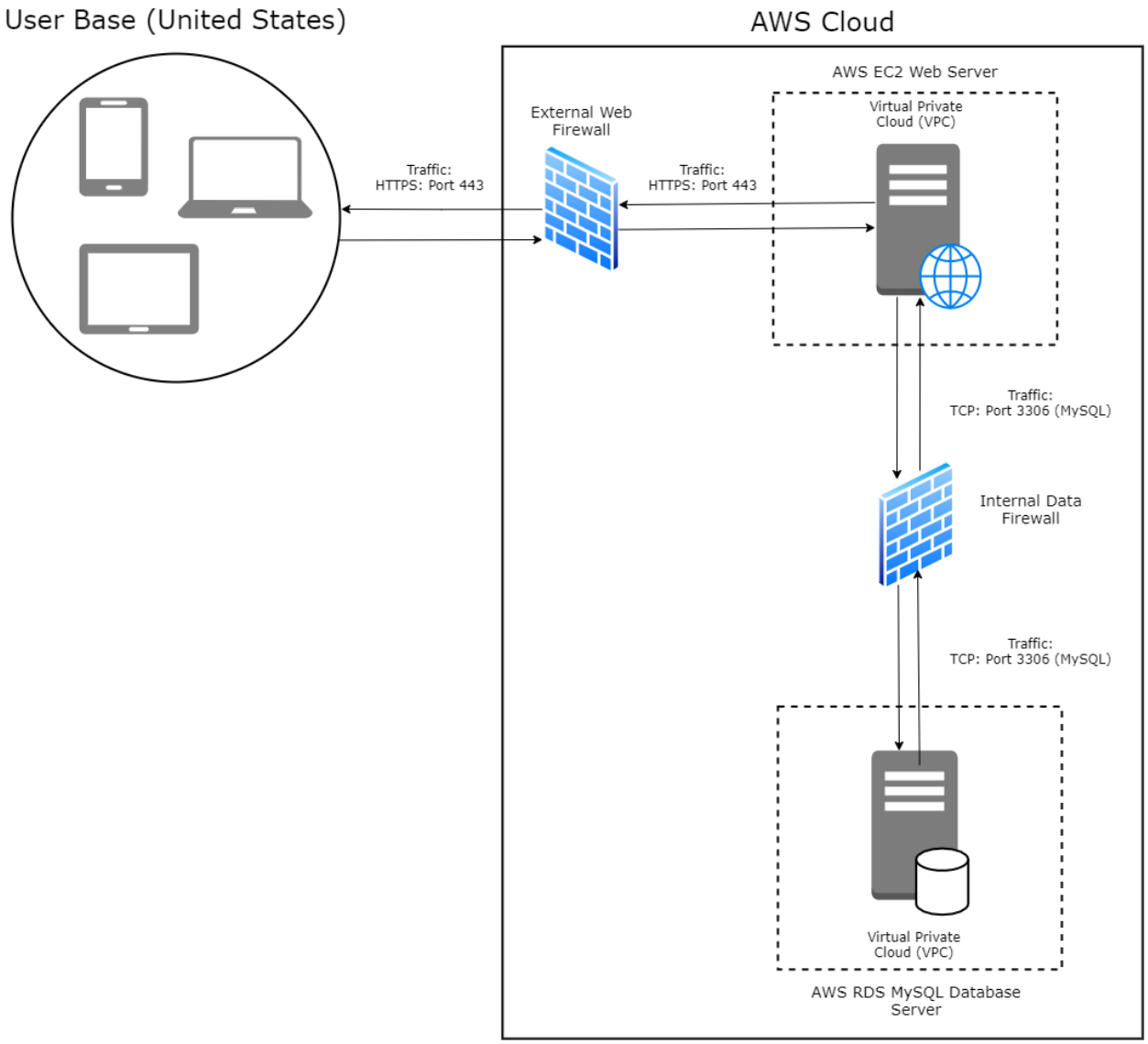
Table of Contents

Overview

Purpose

The purpose of MEet and You's network diagram is to give insight into the network infrastructure involved in our web application. It is intended to demonstrate the client-server communication which happens when clients interact with our application.

Network Architecture



For a client to obtain access to our application, they must progress through our network architecture. Our infrastructure consists of four main components: an external web firewall, a web server, an internal data firewall, and a database server.

Component Definitions

User Base

Our user base, or clients, must be accessing our web application from the United States on the desktop version of Google Chrome v94.0.4606.61. Our application will only accept client HTTPS requests on port 443.

External Web Firewall

We intend for our system to be protected from malicious external traffic via a firewall. This firewall will be based on a set of rules to filter out all non-HTTPS and non-U.S. traffic to ensure that no unauthorized requests infect our system. Additionally, our system will be limited to accepting requests from IP addresses in the format 0.0.0.0/0. The system will apply these filters on both incoming and outgoing traffic to ensure the overall robustness of our application.

Web Server

The web server component of our network architecture will house the platform for our web application. This will serve as the interface for Microsoft IIS, the software which we are utilizing for our web server. We intend to create an Amazon EC2 instance to configure our IIS server. This instance will be run on the Microsoft Windows Server 2019 Base (64-bit) with 1 virtual CPU core, 1 GB of memory, and 50 GB of storage. We also plan to encompass our web server entity in a Virtual Private Cloud (VPC) provided by AWS. The IP address of our instance will be 52.9.186.2, as assigned by AWS.

Internal Data Firewall

To ensure extra protection for our database, an internal data firewall will be implemented. Any inbound or outbound traffic to our firewall must be from the TCP port 3306, which indicates a MySQL database. Any request not from port 3306 will not pass through our internal data firewall and thus is not allowed access to our database.

Database

Our database server will host all of the data storage necessary for our application. It will leverage the Structured Query Language MySQL and will be configured as a relational database utilizing AWS's Relational Database Service (RDS). The database will be powered on the MySQL 8.0.26 engine with 1 virtual CPU core, 1 GB of memory, 20 GB of General Purpose (SSD) database storage, and 20 GB of database backup storage. We also intend to encapsulate our database inside of a VPC for an extra layer of security.

Network Flow

Our network diagram begins with the initial request sent to our application from the client. In order for the client to be able to send a request, they must first have access to the application in the United States via a web browser. This request will be an HTTPS request on port 443 and sent to our application's server on the AWS Cloud.

Before entering the internals of our application, the request must first pass through our external firewall, which validates whether the client's IP is within the specified range (0.0.0.0/0) and if the request is of HTTPS protocol. If these two requirements are met, the request is sent via HTTPS to our AWS EC2 IIS web server, where the request is broken down and processed.

If a request requires database access, it will first need to diffuse through our internal data firewall. This firewall will ensure that all requests come from port 3306, the port required for access to our MySQL database. All requests will be connected to our database via TCP/IP and be encrypted using SSL/TLS to serve as utmost protection for our client's data.

Once the request is fully processed, it will be sent from our EC2 server to our external web firewall via the HTTPS protocol. This request is again validated by our external firewall to check the IP address range and protocol. If the firewall validation is successful, the client then receives the appropriate return request in their web browser.

(NOTE: If the request needed access to our database, it would need to pass back through our internal data firewall before returning to the EC2 server, where a similar return-to-client propagation process would occur.)

Glossary

EC2: An abbreviation for Elastic Compute Cloud, a service provided by AWS which enables the creation of servers for application deployment.

HTTP: An abbreviation for Hypertext Transfer Protocol, a method of web communication utilized widely in information systems.

HTTPS: An abbreviation for Hypertext Transfer Protocol Secure, a more secure and robust version of HTTP (hypertext transfer protocol).

RDS: An abbreviation for Relational Database Service, a service on AWS which allows for the creation of relational databases.

Secure Socket Layer & Transport Layer Security (SSL/TLS): Describes protocols to encrypt a connection to a web application. They require certificates distributed by a certificate authority (CA) to use.

Transmission Control Protocol (TCP/IP): Protocols which work together to allow computers to communicate; creates a connection where information can be sent between the machines.

Virtual Private Cloud (VPC): a feature of AWS which allows for web server (EC2) and relational database (RDS) instances to be placed inside of a virtual environment where the user has full control over things like resource placement, connectivity, and security.