Network Diagram: Eden Fresh

Provided by Team Web Crawlers

Alexander Dung

Devarsh Patel

Justin Calma (Team Leader)

Kayla Chu

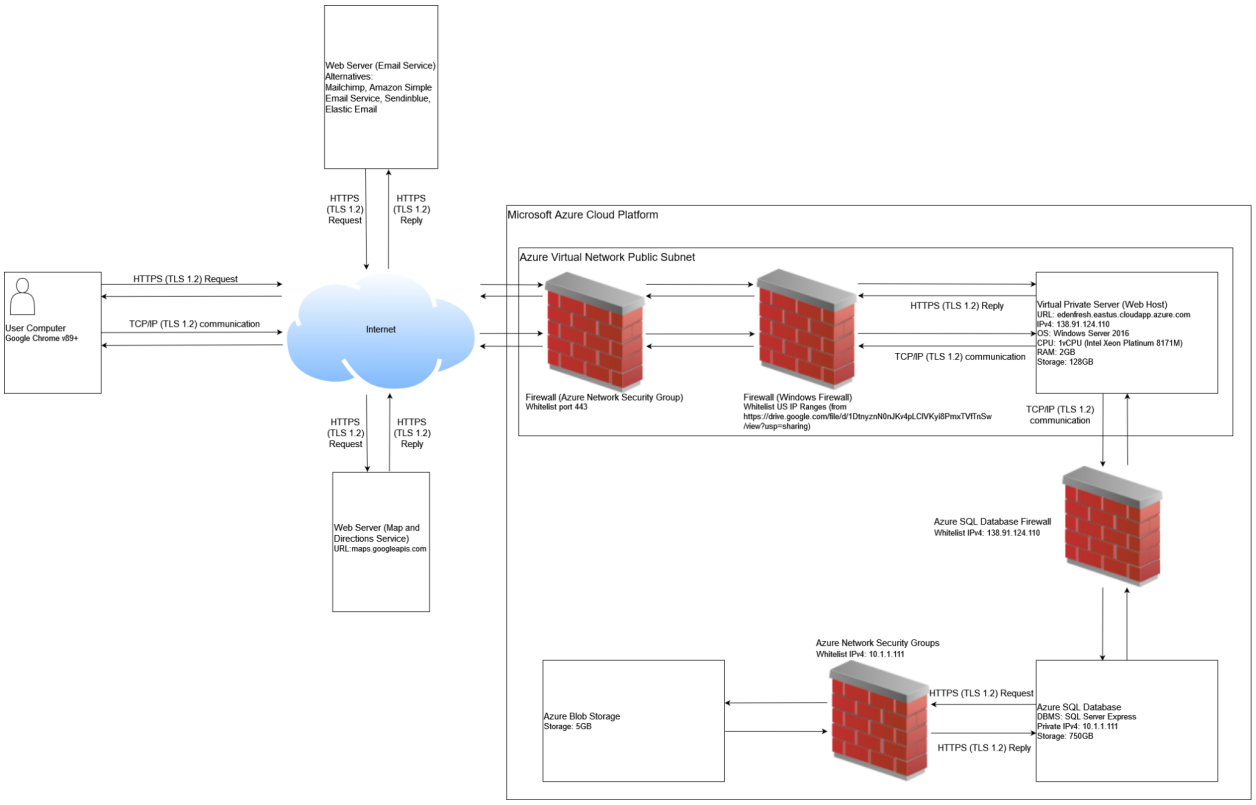
Sze Man Tang

Submission Date: December 15, 2021

1. Introduction

The network diagram is a high-level overview of all network traffic within our web app's system, including communication between the user's device and back-end resources, as well as communication between back-end resources themselves.

2. Target Architecture Diagram



3. Diagram Elements

Network traffic involving our web app consists of six components: the Internet, the user computer, the web host server, the web host firewall, the maps and directions service server,

and the email service server. Each of these components is represented as a block on the network diagram. Communication between two components is represented as an arrow from one component to another, where the base of the arrow is connected to the source and the arrowhead is the destination.

The Internet is the intermediary network that almost all communication will go through to its destination.

The user computer represents the device that our web app's front-end code will be executed on. It will send requests to different nodes on the network according to actions that the user takes on the web app's user interface.

The web host firewall will be used to filter internet communication to the web server to act as a security layer, in order to minimize the web app's attack surface in terms of network communication. In our case, our firewall will filter any traffic not on port 443, in order to enforce web host traffic to HTTPS and synchronous communication protocols encrypted with TLS.

The web host is the server that will host and execute our web app's back-end code. It will reply to requests initiated from the user computer and initiate requests to the maps and directions service.

The maps and directions service server is a third-party server that will provide street map and route information to both the user computer and web server as needed for certain unique features.

The email service server is a third-party server that will provide email services to allow our app to send users emails. This is necessary to carry out requirements for core components of our web app.

4. Node Interactions

Internet traffic involving our web app's system involves four interactions: two between the user and the web server, one between the user and the map and directions service, and one between the web server and the email service.

Interactions between the user and the web server consist of HTTPS communication encrypted with TLS 1.2 and TCP communication under the WebSocket protocol also encrypted with TLS 1.2. Requests move from the user to the internet, through the web host firewall, and end at the web server. Replies from the web server take the reverse path. HTTPS communication between the user and web server will be used to convey the initial web app document to the user's device, as well as information used to update the document's body content. HTTPS communication will be used to perform most of our app's unique features.

HTTPS communication will also be used to perform a WebSocket handshake between the user and the web host. Afterward, the user's device will communicate through the same path with our web server via TCP encrypted with TLS for synchronous communication. Synchronous communication via TCP will be used for our web app's live chat feature.

HTTPS communication will occur between the client and the map and directions service. Requests from the client move through the internet directly to the map and directions server. Replies from the service take the reverse path. When the user uses our street directions feature, their device will receive location information about the target destination from the web server. This information will be sent to the map and directions server via a URL to receive the street route information and the graphical map to display it.

HTTPS communication will occur between the web server and the email service. Requests from the web server will move through the internet to the email service server. Replies

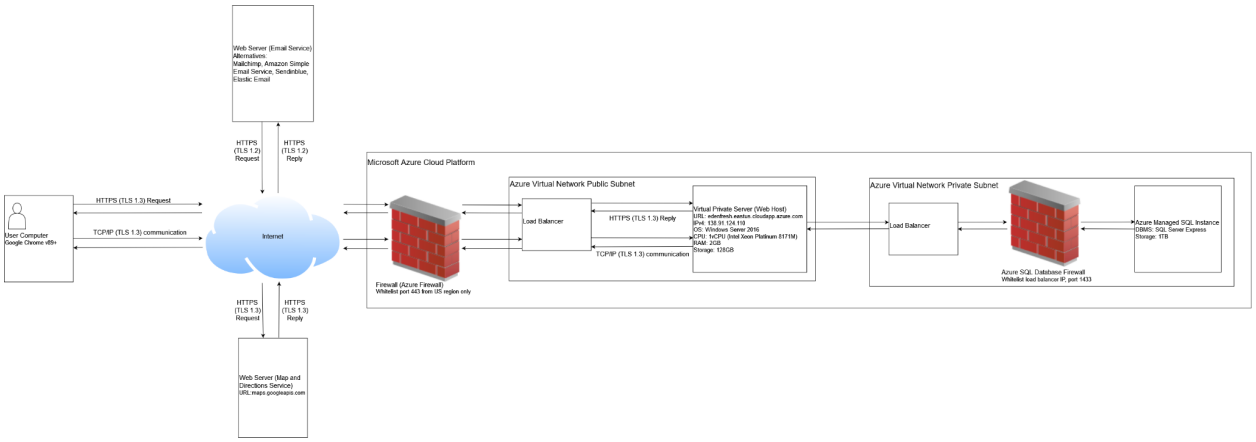
from the service take the reverse path. Generally these requests will be for the email service to send an email to a specified email address.

There will also be two interactions within our back-end, between the web server and our Azure SQL DB, and between the Azure SQL DB and Azure Blob Storage.

The web server will communicate with the Azure SQL DB for data storage purposes. This will require TCP requests from the web server to the DB, and replies from the DB to the web server.

The Azure SQL DB will communicate with the Azure Blob Storage node to perform our archiving functionality. The Azure SQL DB will send a request to the Blob Storage via HTTPS to begin data transfer. After the reply acknowledging, the SQL DB will send files to the Blob Storage.

5. Ideal Architecture Diagram



6. Diagram Elements

Components outside of the Azure back-end remain the same. The Azure back-end is replaced with a web farm structure in the public subnet and a database cluster structure in the private subnet.

7. Node Interactions

Interactions outside of the Azure back-end will remain the same as in the target architecture interactions. Interactions between nodes within the Azure back-end will be different.

The web host server will communicate with the database cluster through the load balancer to perform data access functionalities. The load balancer will direct traffic to an available SQL Instance. While the load balancer currently is only able to route traffic to one SQL Instance according to the ideal architecture diagram, the introduction of a load balancer allows for future expandability of the database cluster without having to update code in the web host servers to take new IP addresses into account.