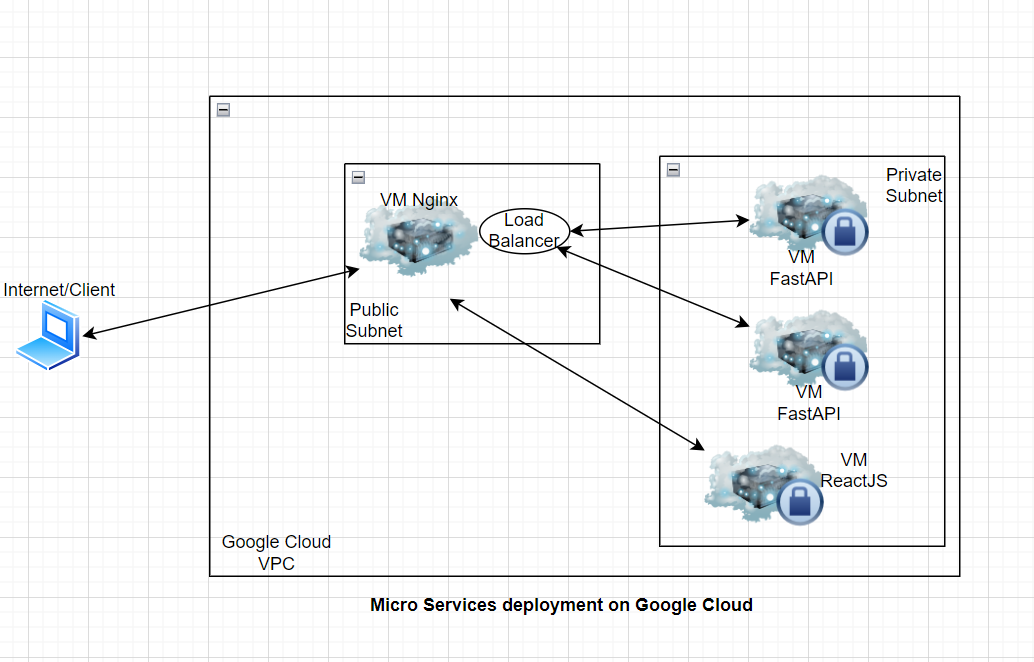
# ReactJS and FastAPI deploy with Nginx Loadbalancer on Google Cloud.

## Working Diagram

[](https://github.com/MdAhosanHabib/GCP-ReactJS-FastAPI-Nginx-Deploy/blob/main/photo/FullStackGCP.PNG)

## Introduction

In this theory document, we delve into the process of creating a Virtual Private Cloud (VPC) environment and configuring Virtual Machines (VMs) using the Google Cloud Platform (GCP). This involves the utilization of various technologies to establish a functional network architecture, deploy web applications making with python FastAPI and ReactJS, and ensure seamless communication between components with Nginx Loadbalancer.

## Creating VPC and Subnet:

#### Virtual Private Cloud (VPC) and Subnets

Virtual Private Cloud (VPC) is a foundational technology in cloud computing that allows users to create isolated network environments within a cloud provider's infrastructure. VPCs offer control over IP address ranges, subnets, and firewall rules to enhance security and manageability.

1. Creating vpc-ahosan on GCP:

Name: vpc-ahosan.

1st Subnet: vpc-proxy-ahosan-subnet.

1st IP Range: 10.0.10.0/24.

1st Private google access off.

2nd Subnet: vpc-webserver-ahosan-subnet.

2nd IP Range: 10.0.12.0/24.

2nd Private google access on.

1st, 2nd Region: us-central1.

1st, 2nd Firewalls rules all selected.

Create now.

## Creating VMs on VPC:

#### Virtual Machines (VMs)

VMs are vital components in cloud computing that provide compute resources for running applications, services, and workloads. VMs enable the provisioning of isolated instances that can be configured to suit specific requirements.

1. Creating vm-proxy-ahosan: Name: vm-proxy-ahosan.

Region: us-central1 (lowa) / us-central-a.

Network Interface: vpc-ahosan.

Subnet: vpc-proxy-ahosan-subnet.

Create now.

1. Creating vm-react-ahosan: Name: vm-react-ahosan.

Region: us-central1 (lowa) / us-central-a.

Network Interface: vpc-ahosan.

Subnet: vpc-webserver-ahosan-subnet.

Create now.

1. Creating vm-fastapia-ahosan: Name: vm-fastapia-ahosan.

Region: us-central1 (lowa) / us-central-a.

Network Interface: vpc-ahosan.

Subnet: vpc-webserver-ahosan-subnet.

Create now.

1. Creating vm-fastapib-ahosan: Name: vm-fastapib-ahosan.

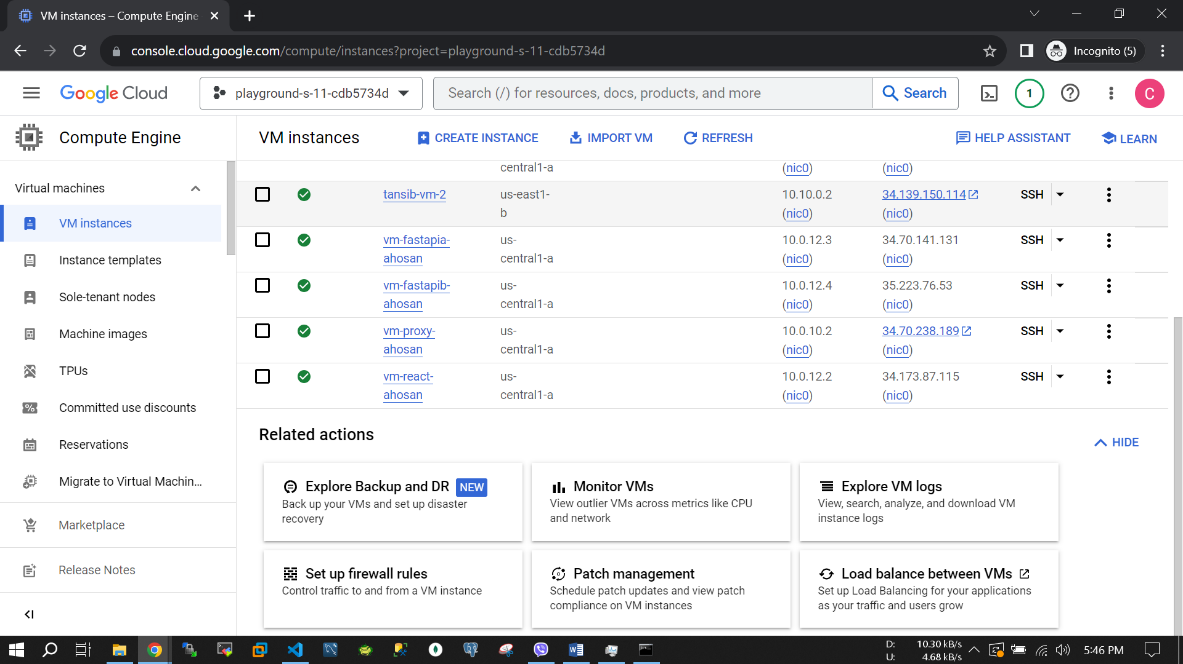
Region: us-central1 (lowa) / us-central-a.

Network Interface: vpc-ahosan.

Subnet: vpc-webserver-ahosan-subnet.

Create now.

#### VMs on GCP

[](https://github.com/MdAhosanHabib/GCP-ReactJS-FastAPI-Nginx-Deploy/blob/main/photo/vms-pic.PNG)

Ping and telnet the servers!

## Port allow from Client to Proxy and proxy to webserver:

* Network Communication and Firewalls

1. Go to Firewall. Name: client-to-proxy-tcp-80.

Network: vpc-ahosan.

Subnet: vpc-proxy-ahosan-subnet.

Trafic: ingress.

Source IP Range: 0.0.0.0/0.

TCP/80.

Create now.

1. Go to Firewall. Name: proxy-to-webserver-tcp-30008000.

Network: vpc-ahosan.

Subnet: vpc-webserver-ahosan-subnet.

Trafic: ingress.

Source IP Range: 10.0.10.0/24.

TCP/8000, TCP/3000.

Create now.

## NodeJS install and ReactJS app create on vm-react-ahosan:

Utilizing Node.js for React App (on vm-react-ahosan) & Node.js Installation, Creating a React Application, Configuring and Running the React App.

apt install telnet

apt install curl

curl -fsSL https://deb.nodesource.com/setup\_lts.x | sudo -E bash –

apt-get install -y nodejs

node -v

v18.17.1

npx -v

9.6.7

mkdir -p /reactjs

cd /reactjs/

npm install -g npm@9.8.1

npx create-react-app react-ahosan

cd /reactjs/react-ahosan

nano src/App.js #replcae with

import React, { useState, useEffect } from "react";

import "./App.css";

function App() {

const [message, setMessage] = useState("");

useEffect(() => {

fetch("http://34.70.238.189/fastapi")

.then((response) => response.json())

.then((data) => setMessage(data.message));

}, []);

return (

<div className="App">

<header className="App-header">

<p>{message}</p>

</header>

</div>

);

}

export default App;

npm start

## Nginx Install and Loadbalancer Config on vm-proxy-ahosan:

Nginx for Reverse Proxy (on vm-proxy-ahosan) & Nginx Installation and Connectivity Check, Nginx Configuration, Restart Nginx.

apt install telnet

telnet 10.0.12.2 3000 #vm-react-ahosan

apt install nginx

rm /etc/nginx/sites-enabled/default

mv /etc/nginx/sites-available/default /etc/nginx/sites-available/default-bkp

nano /etc/nginx/conf.d/my\_app.conf #replcae with

upstream backend\_servers {

zone backend\_server\_zone 64k;

server 10.0.12.2:3000;

}

upstream fastapi\_backend {

server 10.0.12.4:8000;

server 10.0.12.3:8000;

}

server {

listen 80;

server\_name 10.0.10.2;

proxy\_set\_header Host $host;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Real-IP $remote\_addr;

location / {

proxy\_pass http://backend\_servers/;

}

location /fastapi {

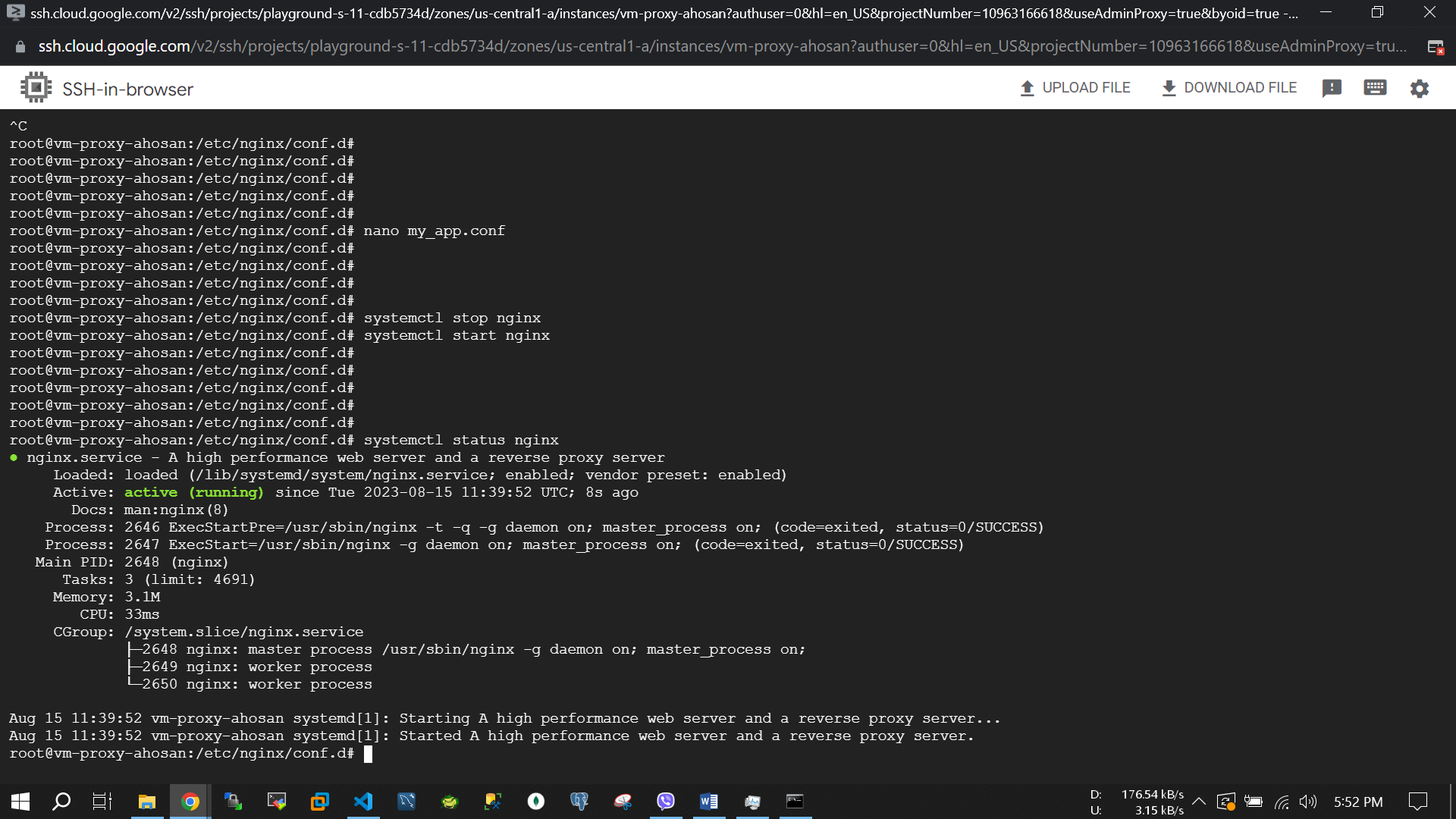
proxy\_pass http://fastapi\_backend/ahosan;

}

}

service nginx restart

#### Nginx spinup

[](https://github.com/MdAhosanHabib/GCP-ReactJS-FastAPI-Nginx-Deploy/blob/main/photo/nginxLB.PNG)

## FastAPI Install and API create:

FastAPI Deployment (on vm-fastapia-ahosan) & Python and FastAPI Setup, FastAPI Application Configuration, Running FastAPI.

apt install telnet

python3 -V

apt install python3-pip

pip install fastapi uvicorn

mkdir -p /fastapi1

cd /fastapi1/

nano main.py #replcae with

from fastapi import FastAPI

from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

#Add CORS middleware with allowed origins

origins = [

"http://localhost",

"http://localhost:3000",

"http://34.70.238.189",

"http://34.70.238.189:80",

]

app.add\_middleware(

CORSMiddleware,

allow\_origins=origins,

allow\_credentials=True,

allow\_methods=["\*"],

allow\_headers=["\*"],

)

@app.get("/ahosan")

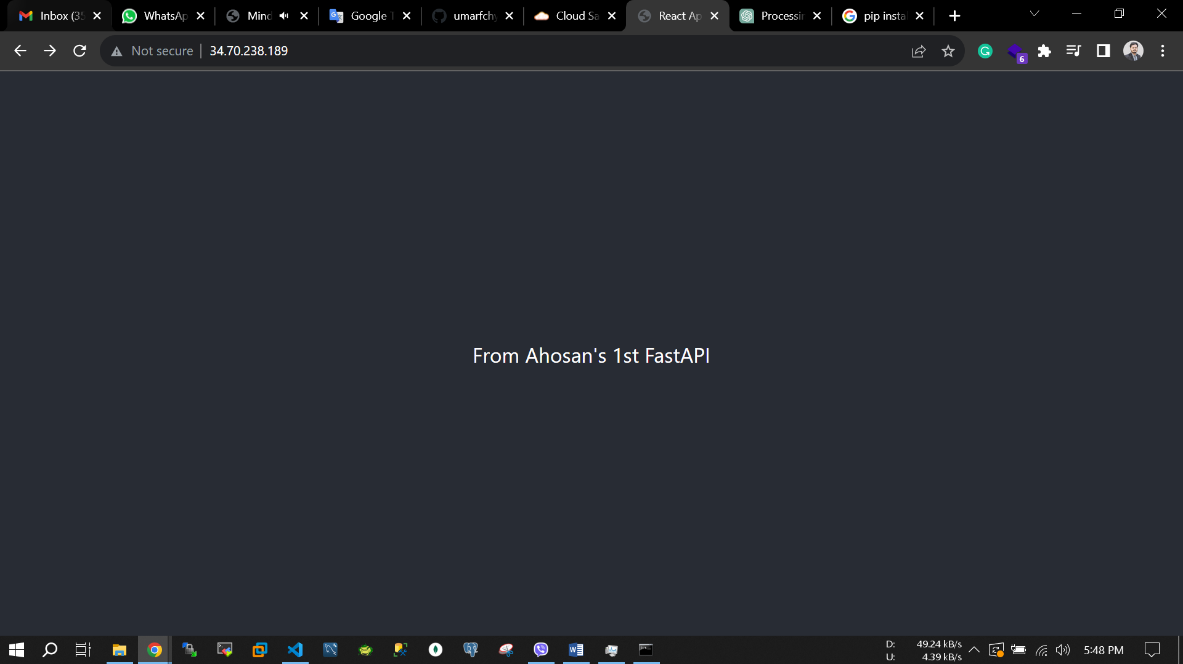
def read\_root():

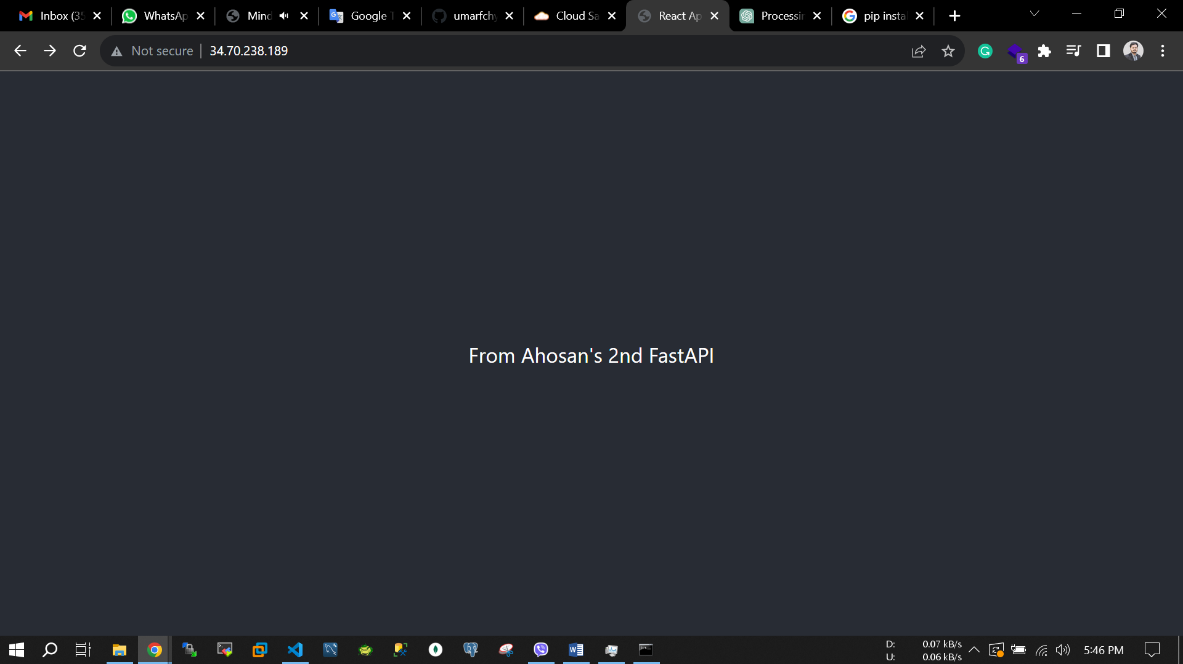
return {"message": "From Ahosan's 1st FastAPI"}

uvicorn main:app --host 10.0.12.3 --port 8000 –reload

## Accessing the Setup

#### ReactJS app fetch data from two FastAPI by Nginx

[](https://github.com/MdAhosanHabib/GCP-ReactJS-FastAPI-Nginx-Deploy/blob/main/photo/1st_fastAPI.PNG)

[](https://github.com/MdAhosanHabib/GCP-ReactJS-FastAPI-Nginx-Deploy/blob/main/photo/2nd_fastAPI.PNG)

* Access React App via Proxy VM's External IP:

Paste Proxy VM's External IP in your browser to load the React app and display data from FastAPI.

Github Link: <https://github.com/MdAhosanHabib/GCP-ReactJS-FastAPI-Nginx-Deploy>

#### Thanks and Regards from Ahosan Habib