



DEPLOYING A QUIZ APPLICATION WRITTEN IN NODEJS TO KUBERNETES WITH A SERVERLESS ETL PIPELINE

Our Team



Andrew Mullen
Co-Director
System Admin



Khalil Elkharbibi
Co-Director



Nalani Daniels
Project Manager



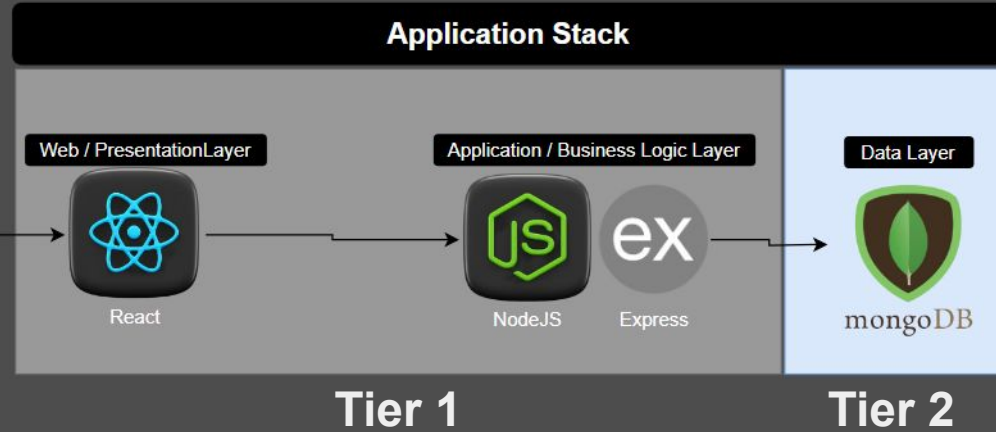
Dwayne Toler
Lead Architect

Notes:

Cloning the repo to an EC2 instance



Notes:
3 layer, 2 tier
application



Notes:
Using MongoDB
cloud for the
database

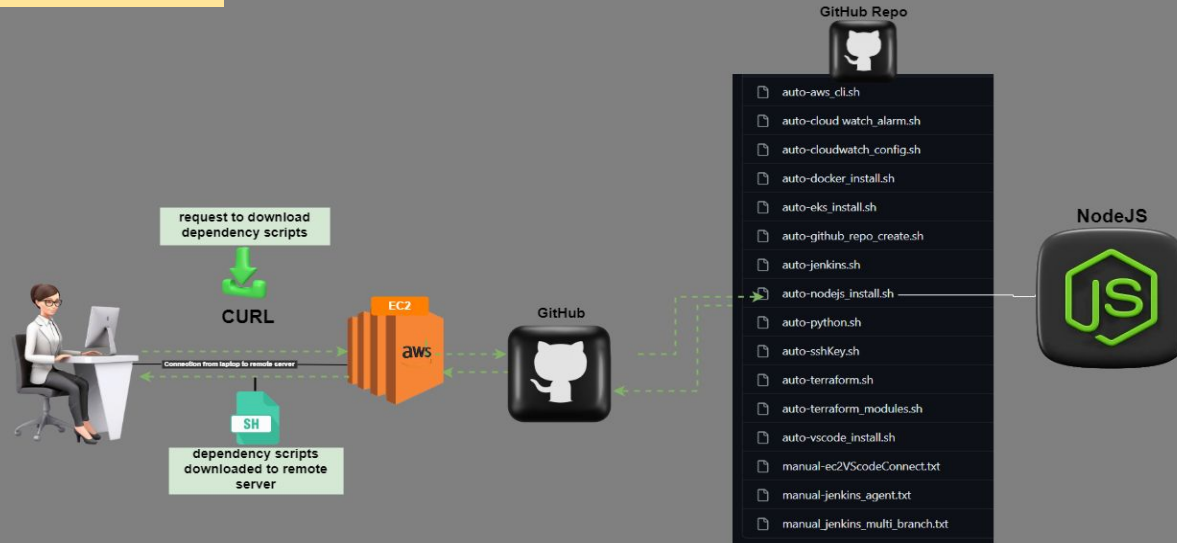


mongoDB

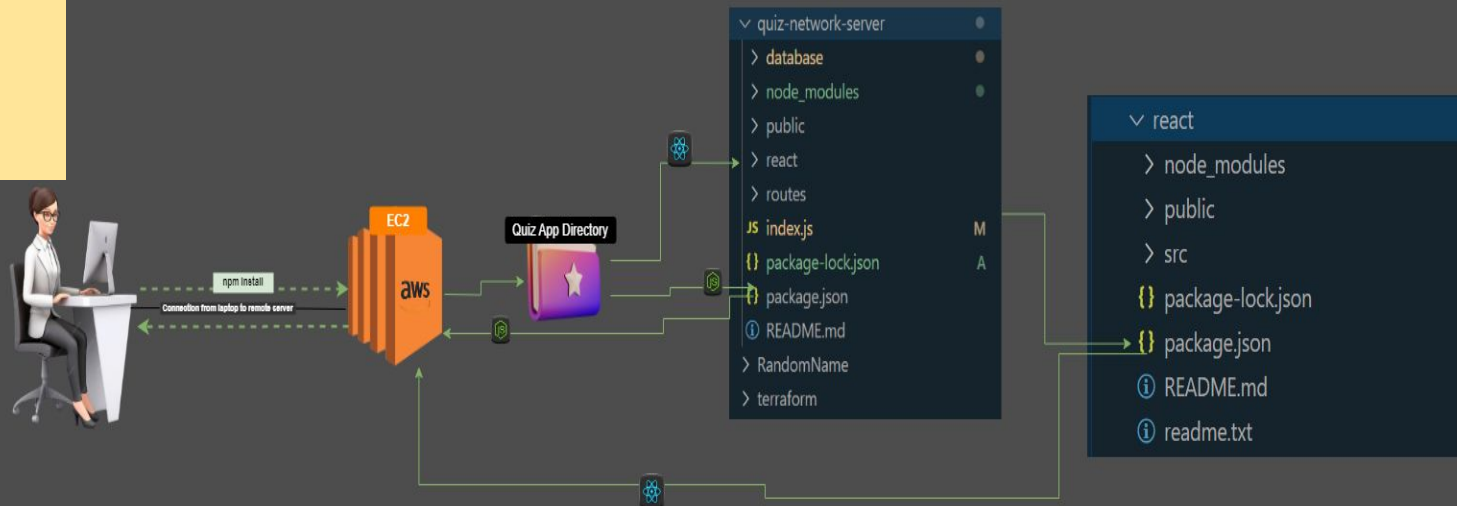
Managed Service



Notes: Installing NodeJS



Notes: Install package.json dependencies



lse>::value' evaluates to false

npm ERR! make: *** [binding.target.mk:126: Release/obj.target/binding/src/binding.o] Error 1

npm ERR! gyp ERR! build error

npm ERR! gyp ERR! stack Error: `make` failed with exit code: 2

npm ERR! gyp ERR! stack at ChildProcess.onExit (/home/ubuntu/docrepo/quiz-network-server/react/node_modules/node-gyp/lib/build.js:194:23)

npm ERR! gyp ERR! stack at ChildProcess.emit (node:events:519:28)

npm ERR! gyp ERR! stack at ChildProcess._handle.onexit (node:internal/child_process:294:12)

npm ERR! gyp ERR! System Linux 6.2.0-1016-aws

npm ERR! gyp ERR! command "/home/ubuntu/.nvm/versions/node/v21.4.0/bin/node" "/home/ubuntu/docrepo/quiz-network-server/react/node_modules/node-gyp/bin/node-gyp.js" "rebuild" "--verbose" "--libsass_ext=" "--libsass_cflags=" "--libsass_ldflags=" "--libsass_library="

npm ERR! gyp ERR! cwd /home/ubuntu/docrepo/quiz-network-server/react/node_modules/node-sass

npm ERR! gyp ERR! node -v v21.4.0

npm ERR! gyp ERR! node-gyp -v v8.4.1

npm ERR! gyp ERR! not ok

npm ERR! Build failed with error code: 1

npm ERR! A complete log of this run can be found in: /home/ubuntu/.npm/_logs/2023-12-10T12_39_03_015Z-debug-0.log

ubuntu@ip-172-31-86-253:~/docrepo/quiz-network-server/react\$

Notes:

Errors installing dependencies



Does the app actually work?

What operating system is he using?



quiz-network-server

Public



main



1 branch



0 tags



yashksaini Create README.md



database

Create db.js



public

Add files via upload



react

Update README.md



routes

Update route.js



README.md

Create README.md



index.js

Update index.js



package.json

Update package.json

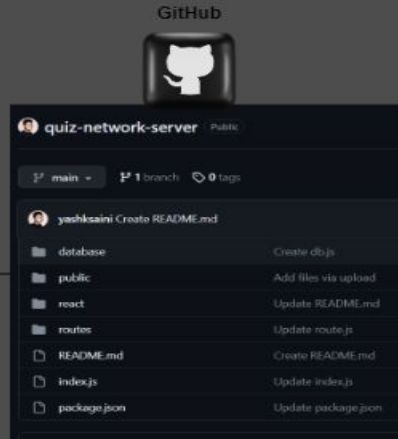
Notes:

Reaching out to the original repo owner to see what OS was used to run the app



Notes:

Application was
originally ran on
Windows OS



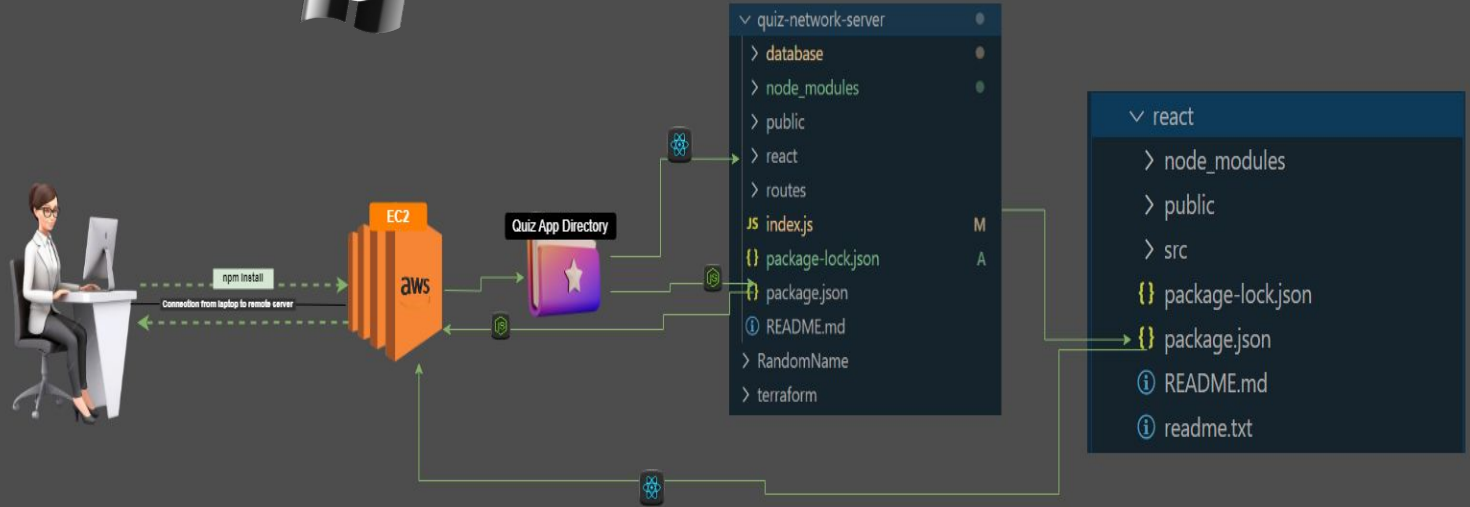
Windows



YES



Local Windows OS



Notes:
Ran application
from local Windows
OS

RUNS ON WINDOWS



```
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!
```

```
The project needs your help! Please consider supporting of core-js:
```

```
> https://opencollective.com/core-js  
> https://patreon.com/zloirock  
> bitcoin: bclqlea7544qtsnj2rayg@lthvza9fau63ux@fstcz
```

```
Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job -)
```

```
> core-js-pure@3.26.1 postinstall /home/ubuntu/docrepo/quiz-network-server/react/node_modules/core-js-pure  
> node -e "try{require('./postinstall')}catch(e){}"
```

```
> node-sass@7.0.3 postinstall /home/ubuntu/docrepo/quiz-network-server/re  
> node scripts/build.js
```

Notes:

App works when running
on Windows

Local machine



<http://localhost:3000>

Notes:
Frontend runs



Quiz Network

Welcome Back

Please enter your details

Username

Password

Log In

Don't have account? [Sign Up](#)

2 QUESTIONS...

**WINDOWS
CONTAINER???**



**DEPENDENCY
VERSION??**

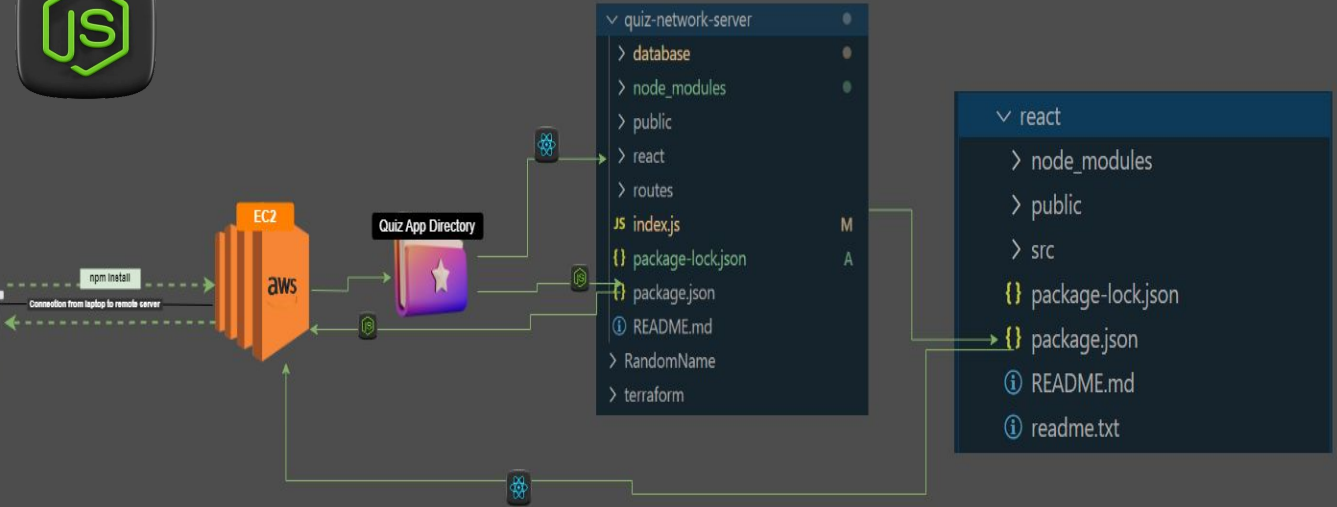


NODE V21



Notes: Deciding whether to run Windows container or find NodeJS version that works

Install NVM Use Node V14



RUNS ON LINUX



```
added 355 packages from 142 contributors, removed 273 packages, updated 1351 packages and audited 1707 packages in 91.106s
```

```
254 packages are looking for funding
```

```
run `npm fund` for details
```

```
found 5 vulnerabilities (4 moderate, 1 high)
```

```
run `npm audit fix` to fix them, or `npm audit` for details
```

New **major** version of npm available! **6.14.18** → **10.2.5**
Changelog: <https://github.com/npm/cli/releases/tag/v10.2.5>
Run `npm install -g npm` to update!

ubuntu@ip-172-31-86-253:~/final-kube\$ npm start

> server@1.0.0 start /home/ubuntu/final-kube

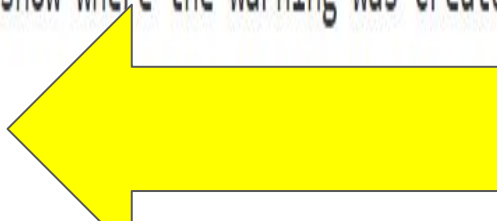
> node index.js

(node:1436956) [MONGOOSE] DeprecationWarning: Mongoose: the `strictQuery` option will be switched back to `false` by default in Mongoose 7. Use `mongoose.set('strictQuery', false);` if you want to prepare for this change. Or use `mongoose.set('strictQuery', true);` to suppress this warning.

(Use `node --trace-deprecation ...` to show where the warning was created)

Server is running

Database connected successfully





Quiz Network

Sign Up

Username

Lanilani

Full Name

Lani Lani

Create Password

.....|

Sign Up



cluster0.mjwldb...



Documents
test.users



{ } My Queries

☰ Databases



Search

▶ ☰ admin

▶ ☰ config

▶ ☰ kurafinalpj

▶ ☰ local

▼ ☰ test

📁 attempts

📁 questions

📁 quizzes

📁 sessions

📁 users



test.users

Documents

Aggregations

Schema

Explain Plan

Filter



Type a query: { field: 'value' }


⬇️ ADD DATA

🔗 EXPORT COLLECTION

__v: 0

```
_id: ObjectId('6573414782002a95bb0bf0a4')
username: "Lanilani"
password: "october5"
fullName: "Lanilani"
__v: 0
```

```
_id: ObjectId('65734c0a3b5359148a870829')
username: "Charles"
password: "qwerqwer"
fullName: "Charles"
__v: 0
```



Create Quiz

My Quizzes

Quiz Title

Select Questions

Create Quiz

Create quiz

Questions

My Quizzes

Profile

Add Question

Question Text

Which of these is a popular tool for continuous integration?

Option A

Docker

Option B

Jenkins

Option C

Kubernetes

Option D

All of the above

Select Answer

B

Add Question

DevOps Questions Test



Quiz Link

<https://quiz-network-lhfp.onrender.com/#/quiz/65757773cc3bb088e4f53b9b>

Play

Copy Link

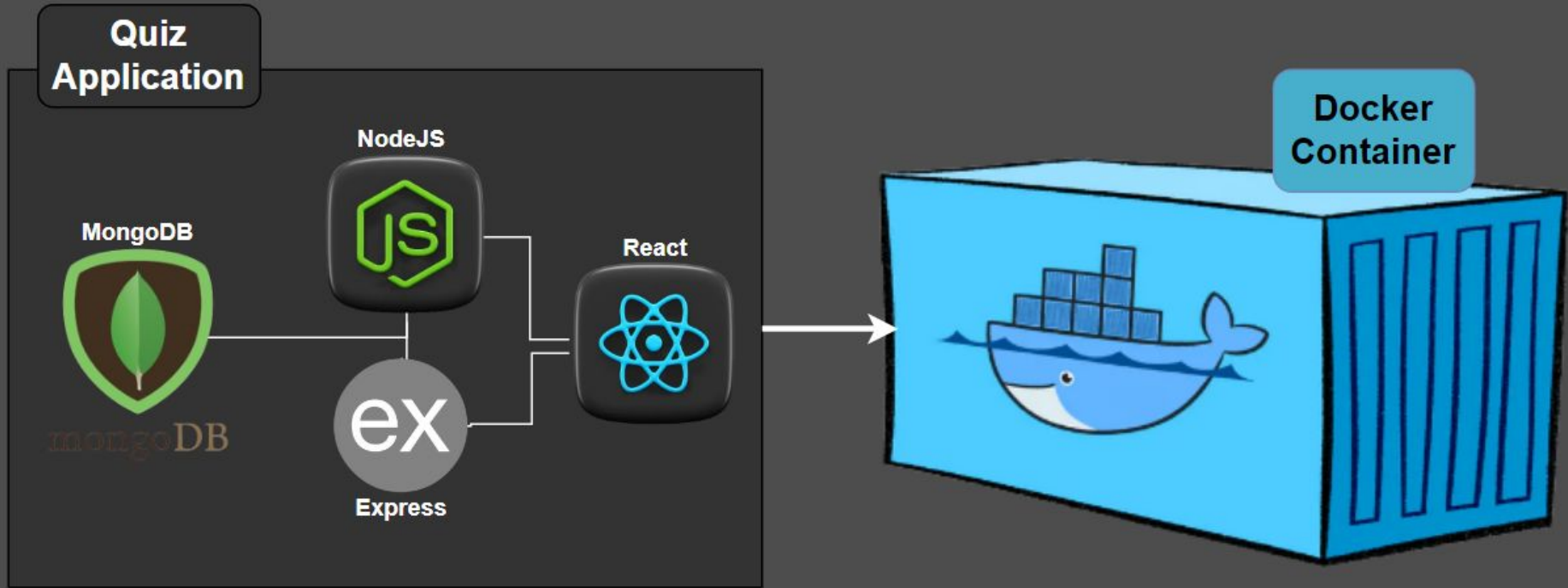
Q. What is Docker primarily used for?

A Network Configuration

B All of the above

C Creating and managing

D Creating & managing source



Notes:
Containerize the app

Frontend Docker Image

```
1 FROM node:14
2
3 WORKDIR /app
4
5 RUN git clone https://github.com/djtoler/finalKube.git .
6
7 WORKDIR /app/react
8
9 RUN npm install
10
11 EXPOSE 3000
12
13 CMD ["npm", "start"]
```



Backend Docker Image

```
1 FROM node:14
2
3 WORKDIR /app
4
5 RUN git clone https://github.com/djtoler/finalKube.git .
6
7 RUN npm install --force
8
9 EXPOSE 8000
10
11 CMD ["npm", "start"]
```



Notes:

Make Docker images for frontend and backend

Frontend Docker Image

```
1 FROM node:14
2
3 WORKDIR /app
4
5 RUN git clone https://github.com/djtoler/finalKube.git .
6
7 WORKDIR /app/react
8
9 RUN npm install
10
11 EXPOSE 3000
12
13 CMD ["npm", "start"]
```



Backend Docker Image

```
1 FROM node:14
2
3 WORKDIR /app
4
5 RUN git clone https://github.com/djtoler/finalKube.git .
6
7 RUN npm install --force
8
9 EXPOSE 8000
10
11 CMD ["npm", "start"]
```



Notes:

Test the containers with
Docker Compose



docker
Compose

Notes:

App runs successfully on containers



Quiz Network

Welcome Back

Please enter your details

Username

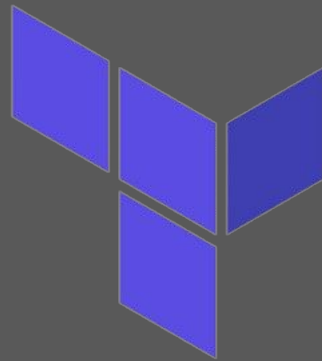
Password

Log In

Don't have account? **Sign Up**



kubernetes



HashiCorp

Terraform

Notes:

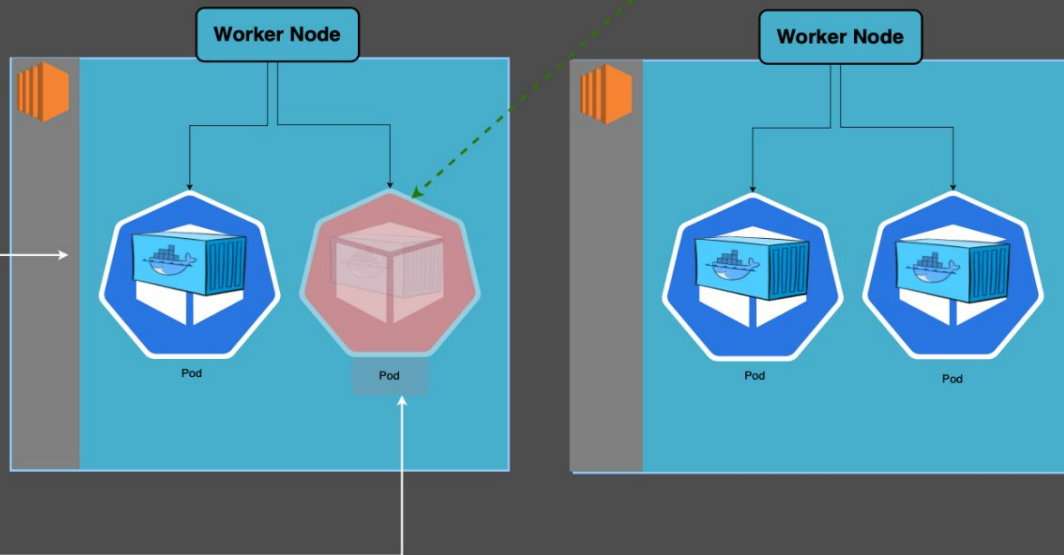
Build infrastructure using Terraform and deploy containers to Kubernetes

Notes:

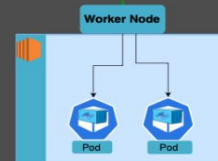
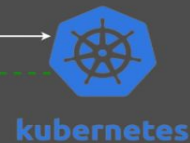
Fault tolerance at the Pod/Container level

deployment.yaml

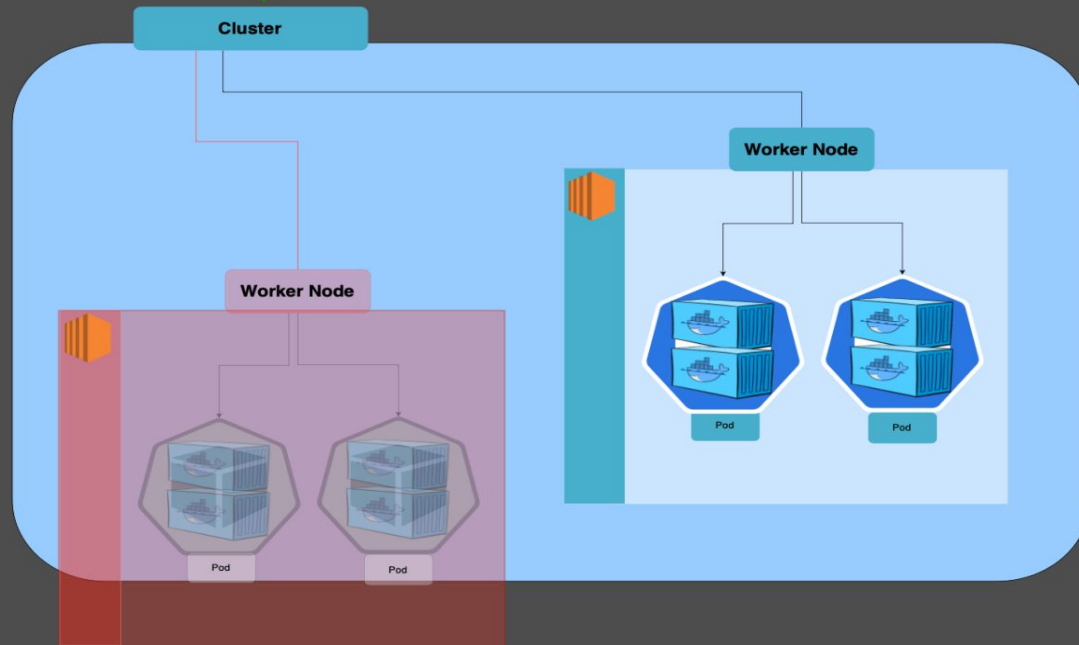
```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: fpj-frontend-app
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: fpjfe
10   template:
11     metadata:
12       labels:
13         app: fpjfe
14     spec:
15       containers:
16       - name: fpjfe
17         image: djtoller/frontkube1:latest
18         ports:
19         - containerPort: 3000
```



```
64  
65 eksctl create nodegroup --cluster cluster11 --node-type t2.medium --nodes 2  
66
```



Notes:
Fault tolerance at
the Node/Cluster
level



Notes: Terraform to build infrastructure

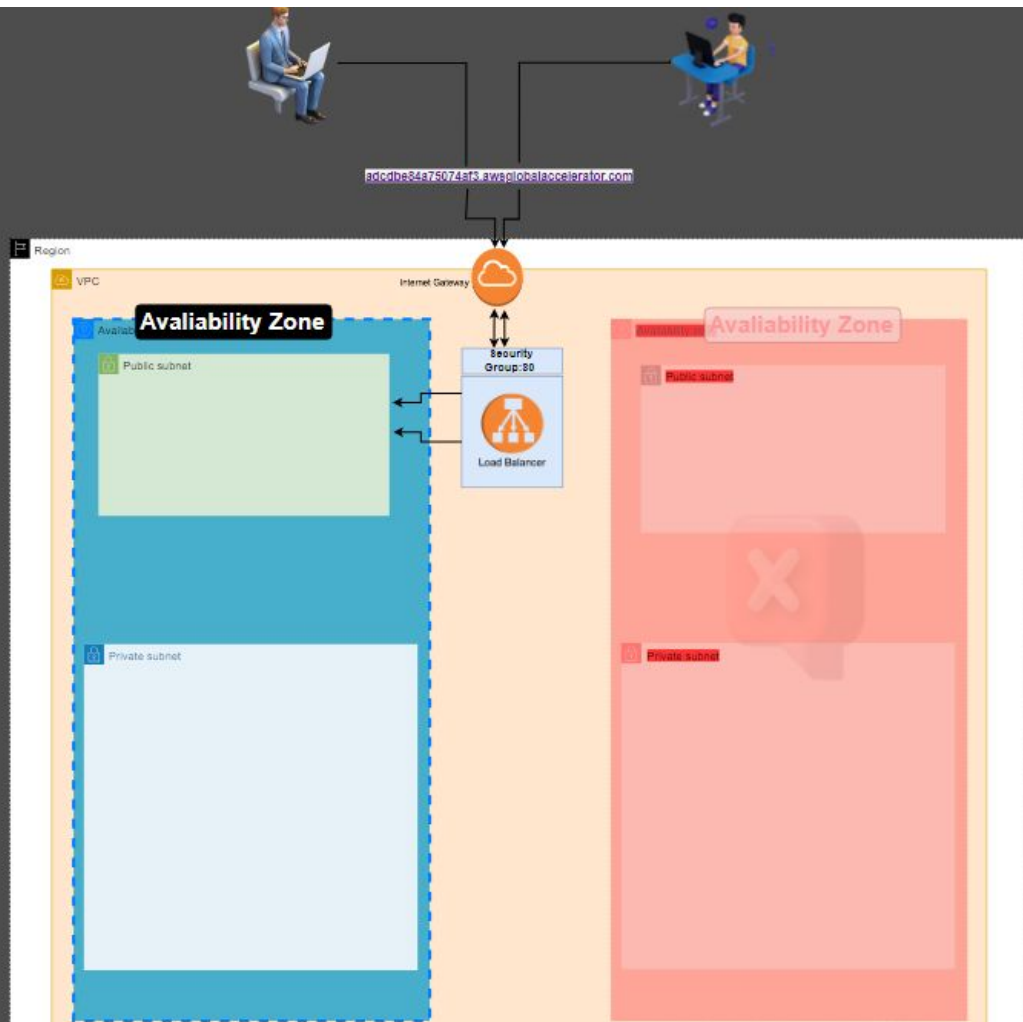


Terraform

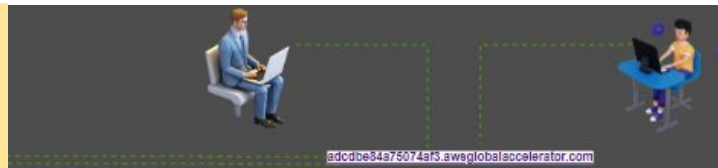
Notes:
Fault
tolerance at
the
availability
zone level



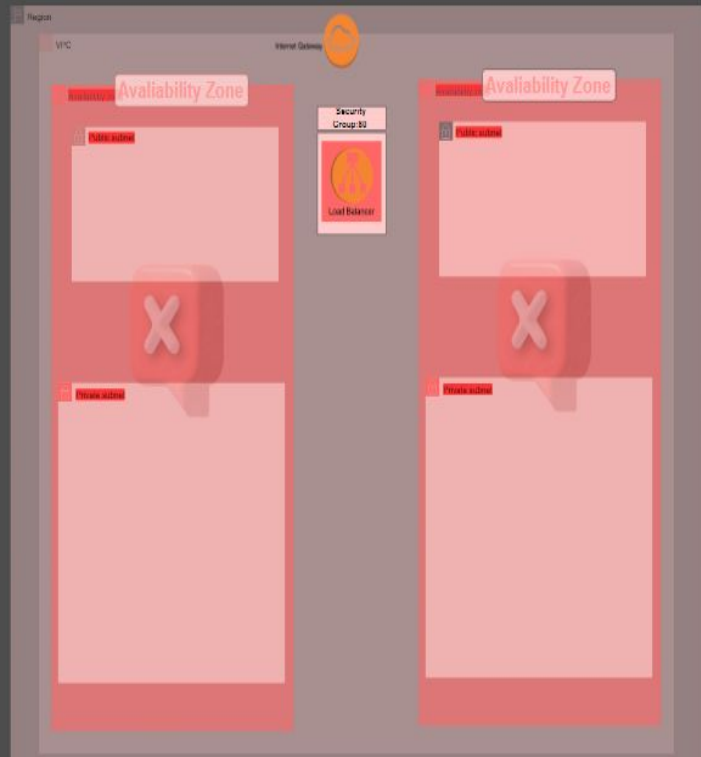
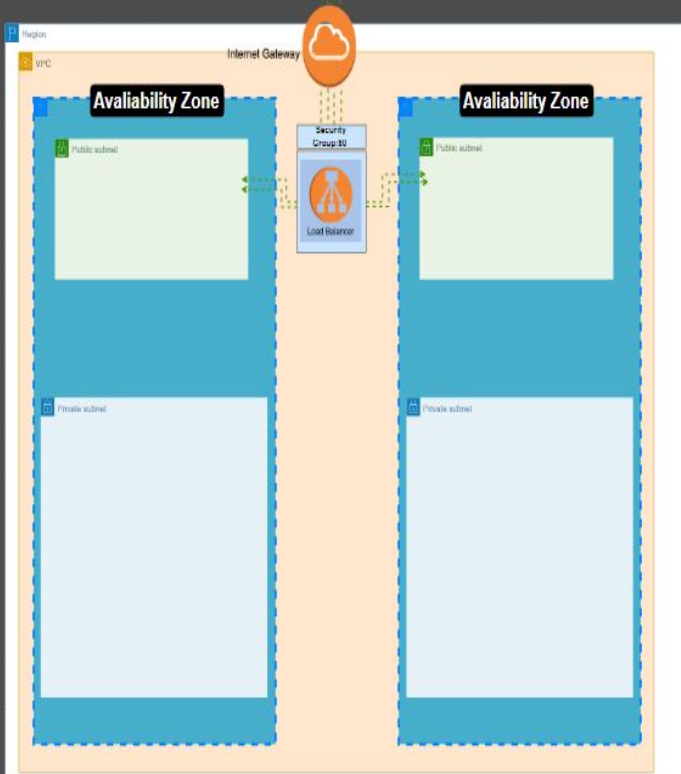
Terraform



Notes:
If both availability
zones go down...



Notes:
We have fault tolerance at the region level



Terraform



Notes:
Using *AWS Global Accelerator* for controlled load balancing between regions and deployment strategies

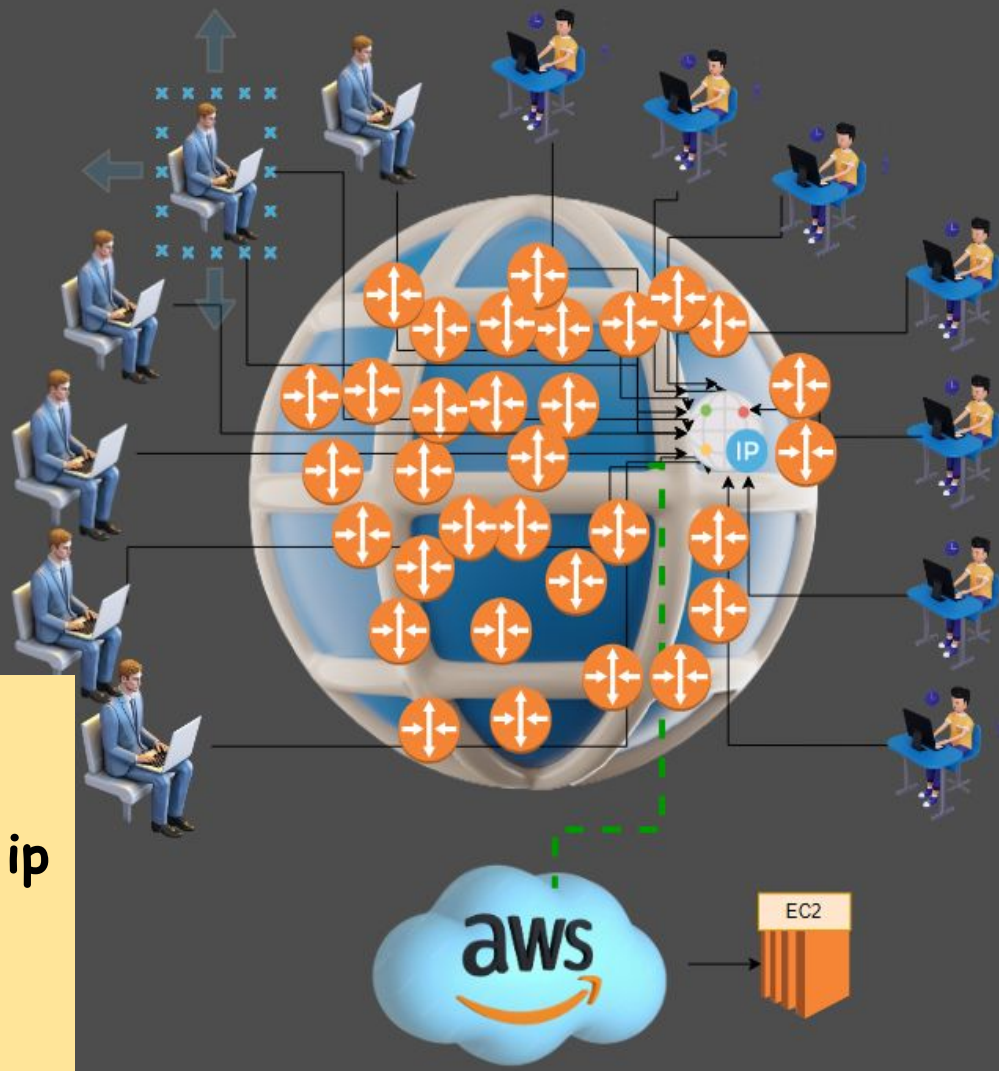
Managed service

Anycast IP technique

Traditional routing



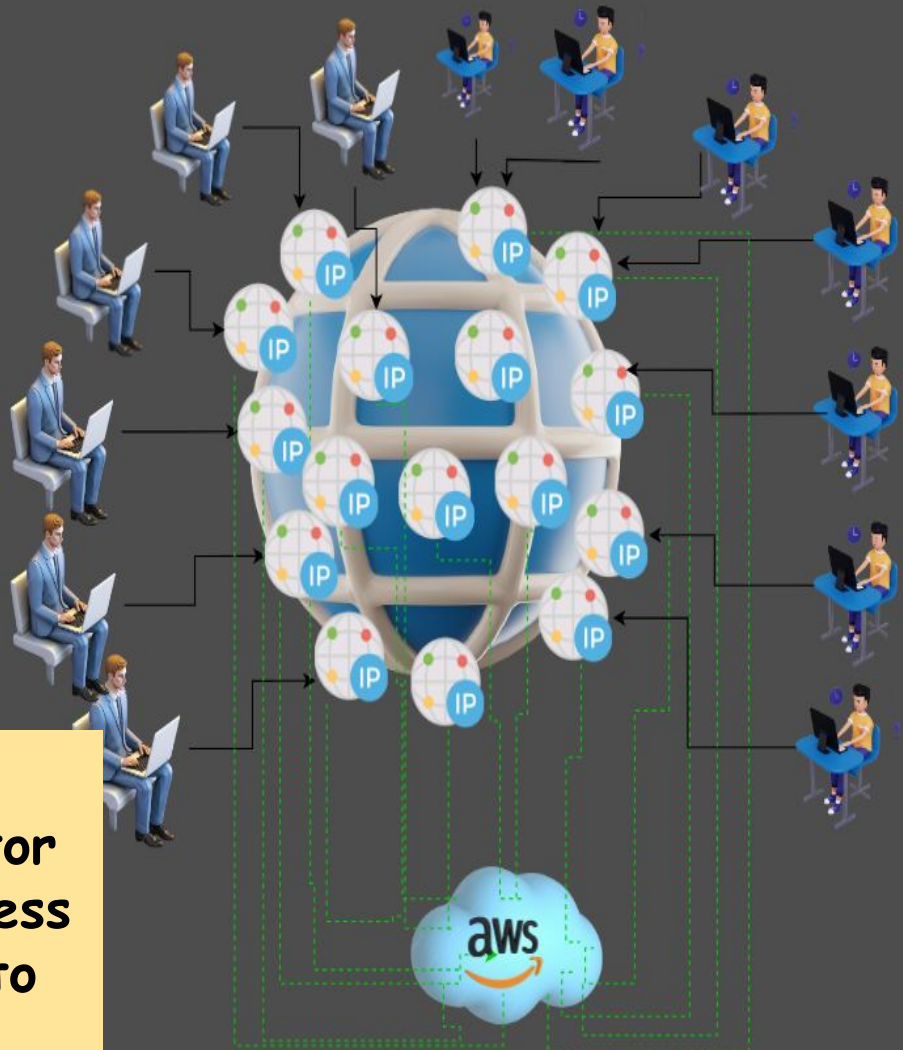
Notes:
Traditional routing is single ip to single server over multiple routers



1 IP Address



1 Location
Multiple Routers



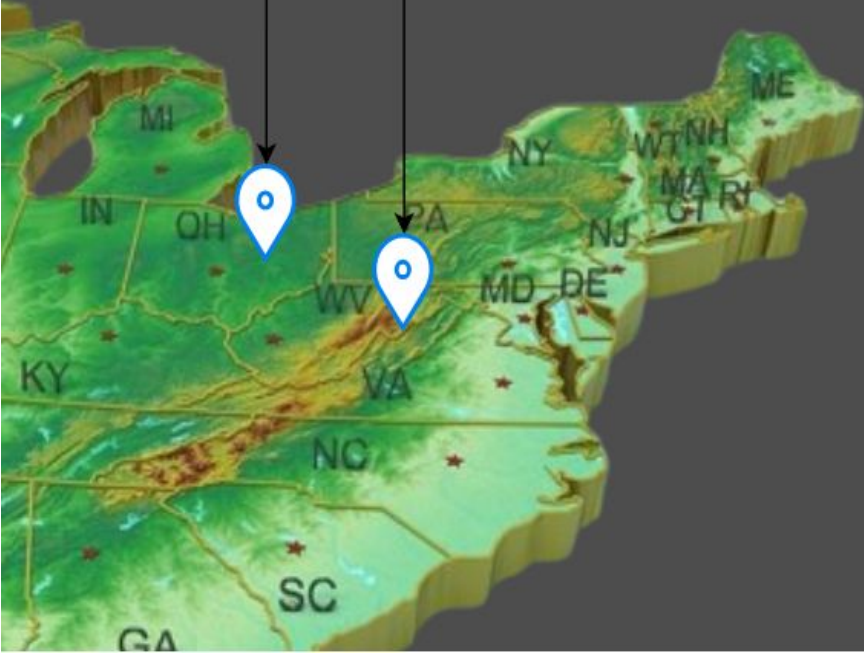
Same Address



Multiple Locations

Notes:
Global accelerator
allows 1IP address
to be assigned to
multiple servers

Global Accelerator IP



Global Accelerator



Global Accelerator IP

Notes: We attach our load balancer IP addresses to a global accelerator IP



US-East-1 [Ohio]
Load Balancer

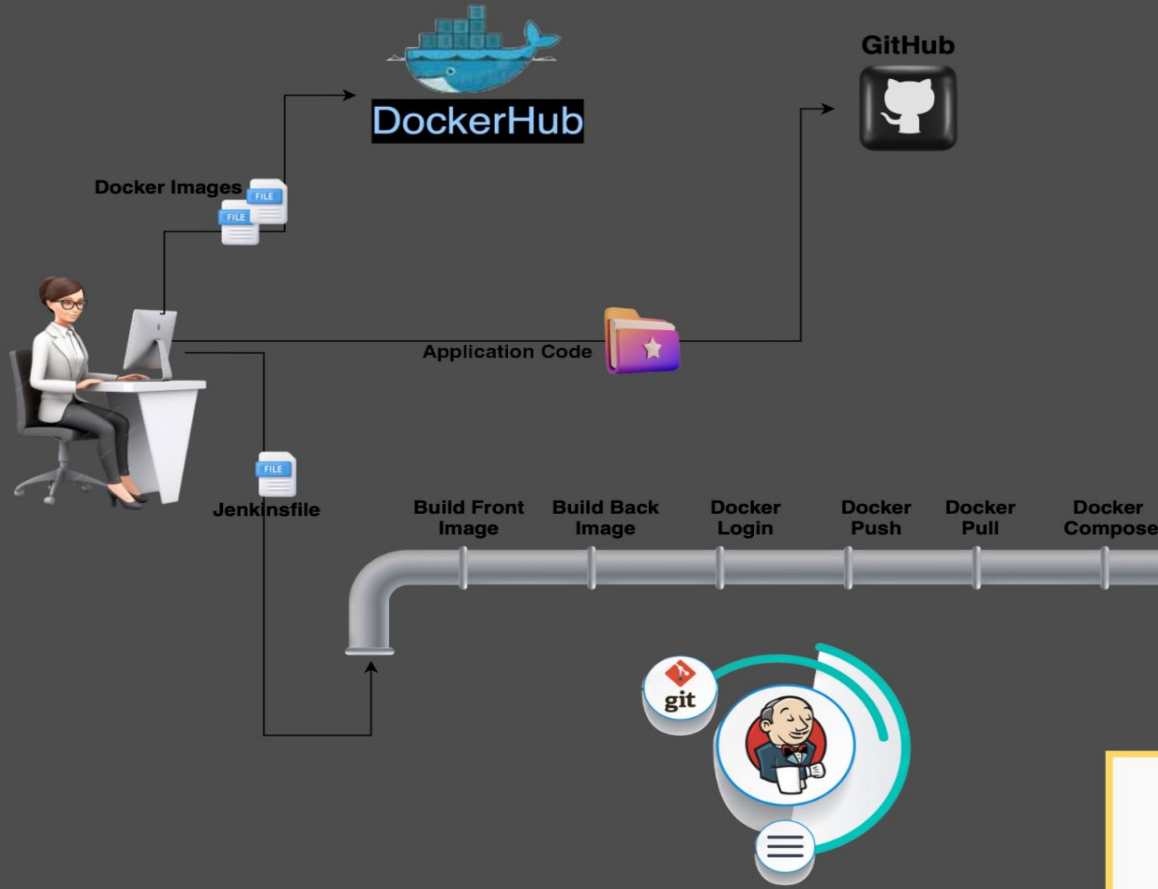


US-East-2 [W. Virginia]
Load Balancer

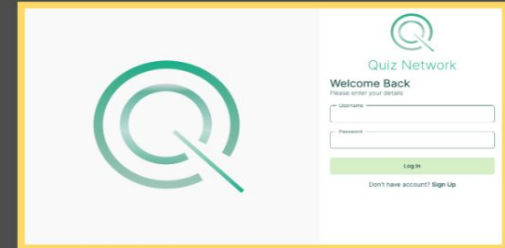


EC2

EC2



Notes: Push our images to DockerHub and application code to GitHub and build our Jenkins pipeline



Notes: Using
JMeter for load
testing



```
<ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="blitz-c4" enable
  <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
  <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass
    <boolProp name="LoopController.continue_forever">false</boolProp>
    <stringProp name="LoopController.loops">1</stringProp>
  </elementProp>
  <stringProp name="ThreadGroup.num_threads">100000</stringProp>
  <stringProp name="ThreadGroup.start_time">1000</stringProp>
  <boolProp name="ThreadGroup.scheduler">false</boolProp>
  <stringProp name="ThreadGroup.duration">10</stringProp>
  <stringProp name="ThreadGroup.delay">0</stringProp>
  <boolProp name="ThreadGroup.same_user_on_next_iteration">true</boolProp>
</ThreadGroup>
<hashTree>
  <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname=
    <elementProp name="HTTPSampler.Arguments" elementType="Arguments" guiclass="HTTP
      <collectionProp name="Arguments.arguments">
    </collectionProp>
    </elementProp>
    <stringProp name="HTTPSampler.domain">adcdbe84a75074af3.awsglobalaccelerator.com</stringProp>
    <stringProp name="HTTPSampler.port">80</stringProp>
    <stringProp name="HTTPSampler.path">/</stringProp>
    <stringProp name="HTTPSampler.method">GET</stringProp>
  </HTTPSamplerProxy>
```

**Notes: Configuring
JMeter for
100,000 GET
requests in 10
seconds**

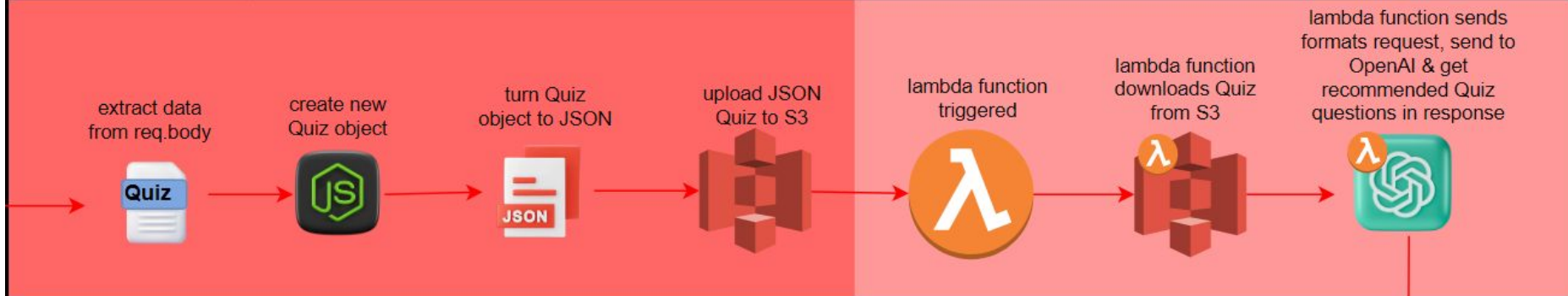
99790	01,75001,HTTP Request,Non HTTP response code:
99791	18,75002,HTTP Request,Non HTTP response code:
99792	27,75000,HTTP Request,Non HTTP response code:
99793	57,75003,HTTP Request,Non HTTP response code:
99794	66,75000,HTTP Request,Non HTTP response code:
99795	72,75005,HTTP Request,Non HTTP response code:
99796	86,75003,HTTP Request,Non HTTP response code:
99797	57,75004,HTTP Request,Non HTTP response code:
99798	

189	Min:	3	Max:	902	Err	0 (0.00%)	Active:
535	Min:	9	Max:	75008	Err	312 (0.47%)	Active:
420	Min:	3	Max:	75008	Err	312 (0.31%)	

99.4%

EXTRACT

TRANSFORM



LOAD



lambda function
formats response
from OpenAI &
uploads to S3

Serverless ETL Pipeline

Notes: Building a serverless ETL pipeline that takes a quiz question that a user creates, then using a lambda function to send a request to OpenAI API to generate similar questions and build a database of suggested questions

Notes: Lambda
code for the
request to
OpenAI API

main [fpfeatures / lambda / lambda_function.py](#)

Ubuntu new start

Code Blame 122 lines (90 loc) · 4.61 KB Code 55% faster with GitHub Copilot

```
1 import boto3
2 import json
3 from openai import OpenAI
4 from botocore.exceptions import NoCredentialsError
5 import random
6
7
8 client = OpenAI()
9 print("Running code")
10
11 s3 = boto3.client('s3')
12
13 def get_latest_s3_file(bucket_name):
14     """Retrieve the latest JSON file from an S3 bucket."""
15     response = s3.list_objects_v2(Bucket=bucket_name)
16
17     # Check if the bucket is not empty and has contents
18     if 'Contents' in response:
19         # Filter out only JSON files and sort them by the last modified date
20         json_files = [file for file in response['Contents'] if file['Key'].endswith('.json')]
21         json_files_sorted = sorted(json_files, key=lambda x: x['LastModified'], reverse=True)
22
23         # Return the name of the latest JSON file
24         return json_files_sorted[0]['Key'] if json_files_sorted else None
25     return None
```

