

# ***My Personal Computer Builder***

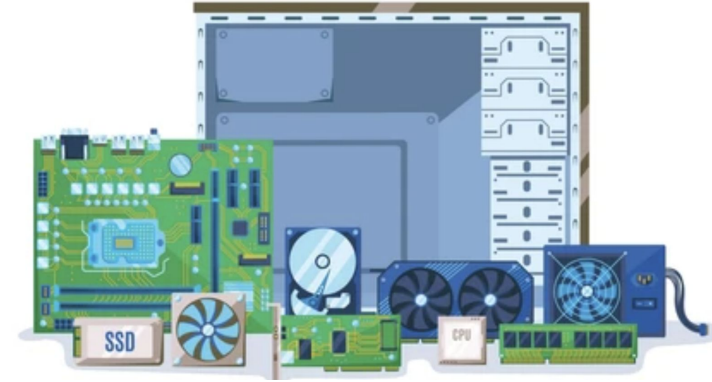
*Course Project for E6156 Cloud Computing*

*Team VegiC*

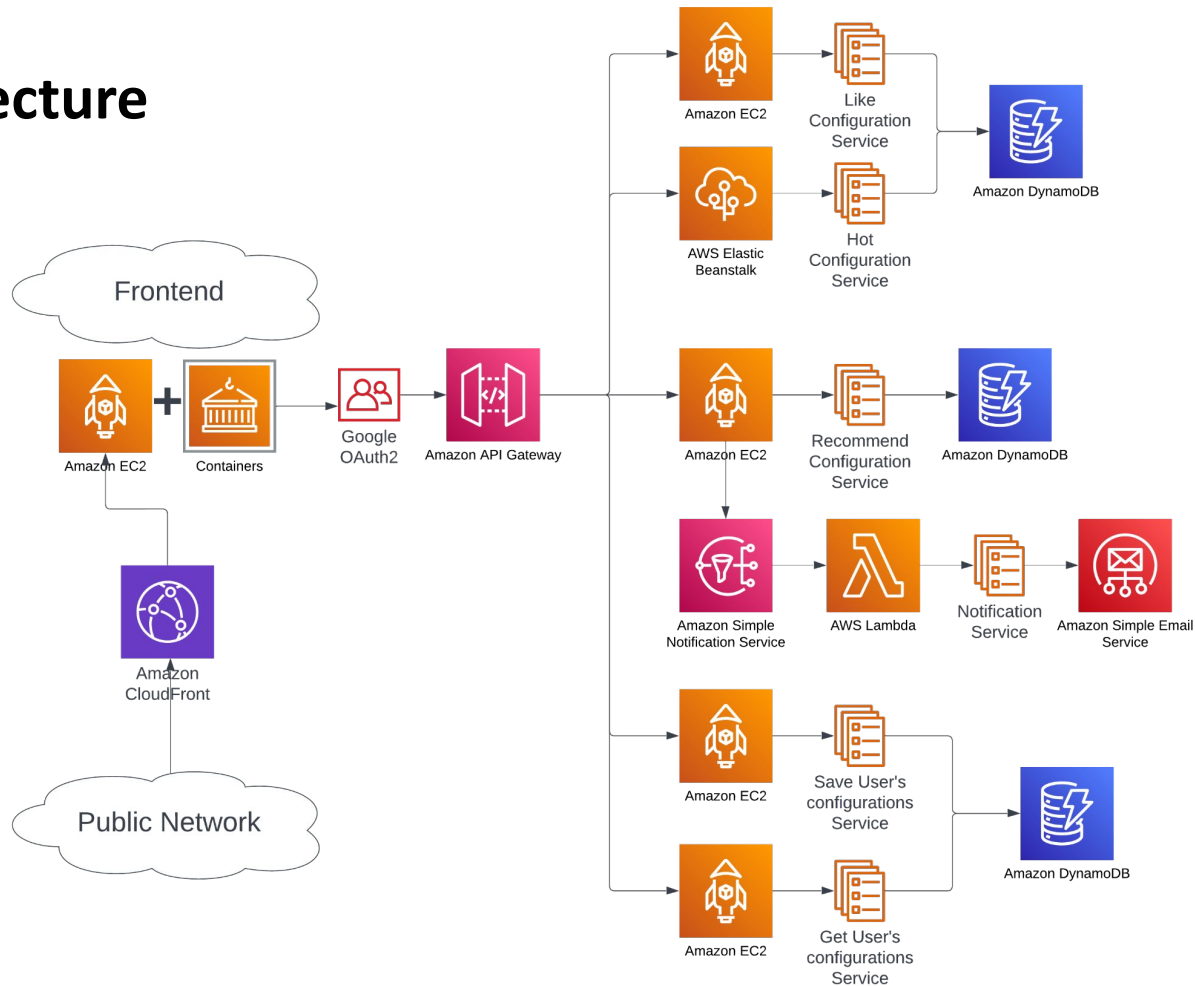
Ruihan Yu, Yuanqi Dan, Junqi Zhang, Tracy Li, Yuerong Zhang, Chang Gao, Jiawen Li

# Project Overview

- User account management (sign up & sign in)
- Recommendation system for PC hardware configurations
- vote/load hot configurations, save user's configurations and notification service



# Architecture



# Data Source

- UserBenchmark web scraping → hardware name & benchmark information
- Rainforest(Amazon) API → hardware price & purchase link information



**UserBenchmark**



**Rainforest**

# Frontend

Tech Stack:

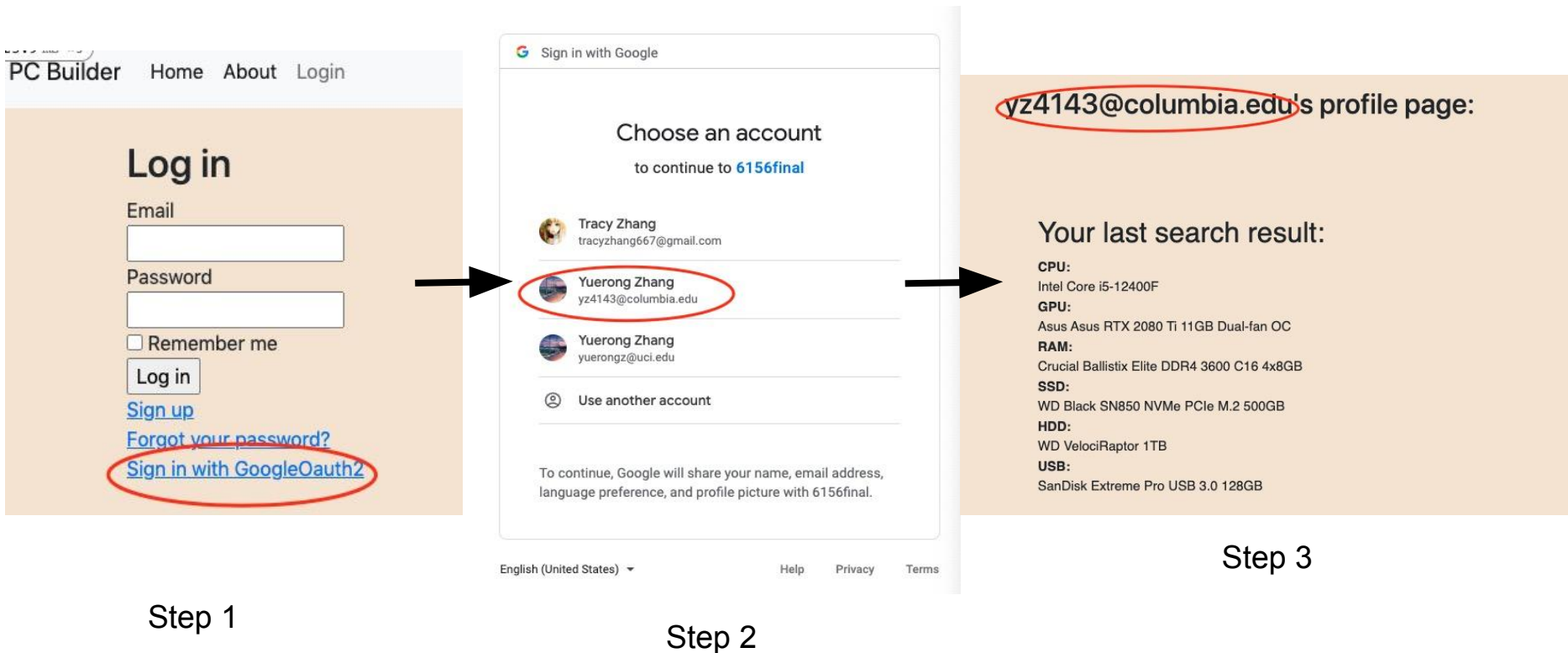
1. Ruby on Rails
2. Sqlite3 (Stores all user information for user login function)
3. html
4. css
5. JavaScript

Continuous Deployment: GitHub Actions

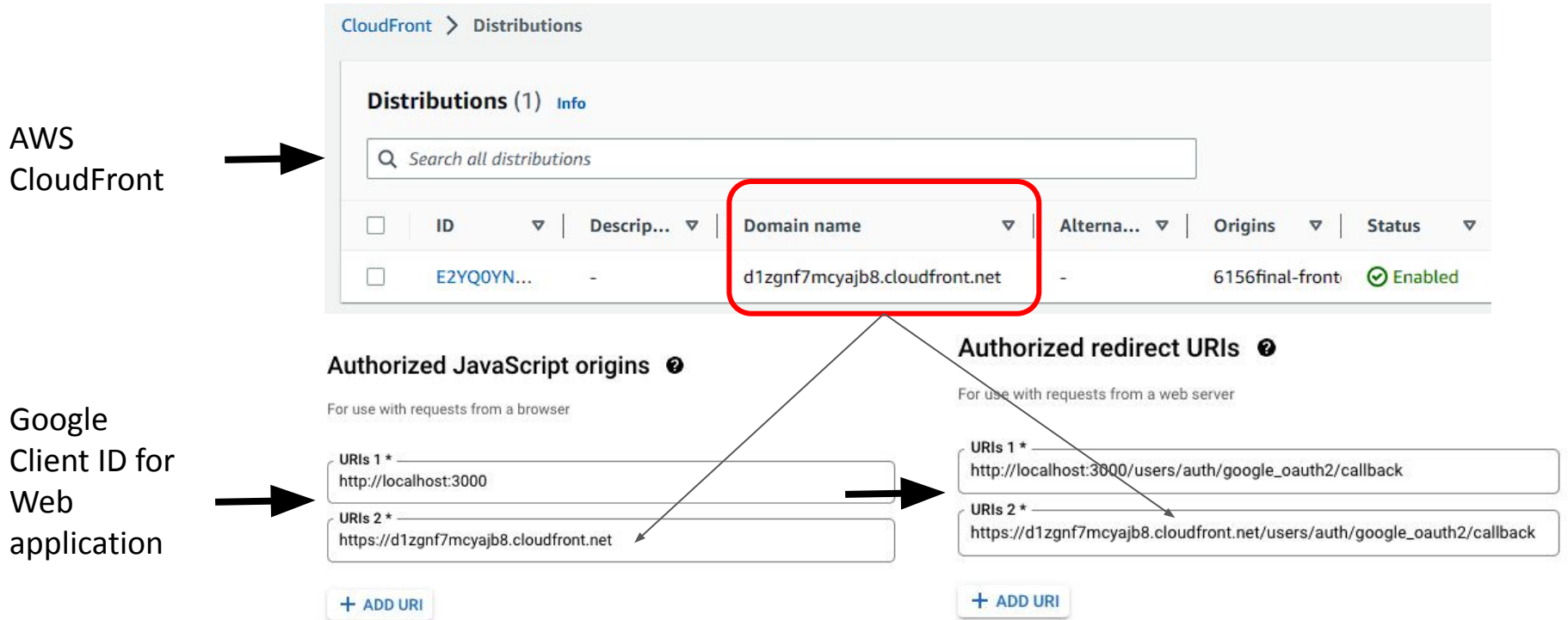
# Frontend Deployment

1. Leverage Github Action to enable Continuous Deployment
  - a. For each push to master branch, the github action will be triggered
2. Do ssh to EC2
3. Build container using Podman
4. Run container using Podman on EC2

# Google Login OAuth2 Workflow Overview



# Google Login OAuth2 & CloudFront Deployment





# API Gateway



## Method Execution /recommend - GET - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

### Path

No path parameters exist for this resource. You can define path parameters by using the syntax **{myPathParam}** in a resource path.

### Query Strings

#### {recommend}

level=gamer&budget=2000&email=y

eg. param1=value1&param2=value2

### Headers

#### {recommend}

### Stage Variables

No [stage variables](#) exist for this method.

### Client Certificate

No client certificates have been generated.

### Request Body

Request Body is not supported for GET methods.

### Request: /recommend?

level=gamer&budget=2000&email=y

Status: 200

Latency: 406 ms

### Response Body

```
{
  "statusCode": 200,
  "body": {
    "CPU": [
      "Intel Core i5-13600K",
      "Intel Core i7-12700K",
      "Intel Core i5-12600KF",
      "Intel Core i9-11900K",
      "Intel Core i9-1900KF"
    ],
    "GPU": [
      "Asus Asus RTX 3080 24GB Tuf OC",
      "MSI MSI RTX 3080 10GB Ventus 3X OC",
      "Gigabyte Gigabyte RTX 3080 10GB Gaming OC",
      "AMD RX 6950-XT",
      "Asus Asus RTX 2080 Ti 11GB Dual-fan OC"
    ],
    "SSD": [
      "WD Black SN850 NVMe PCIe M.2 1TB",
      "WD Black SN850 NVMe PCIe M.2 500GB",
      "Samsung 980 Pro NVMe PCIe M.2 500GB",
      "HP EX950 NVMe PCIe M.2 2TB",
      "Crucial P5 3D NVMe PCIe M.2 1TB"
    ],
    "HDD": [
      "WD Velociraptor 1TB",
      "HGST Ultrastar He8 Helium 8TB",
      "Seagate Barracuda 7200.14 1TB",
      "Toshiba P300 3TB",
      "Toshiba DT01ACA200 2TB"
    ],
    "RAM": [
      "G.SKILL Flare X DDR4 3200 C14 4x8GB",
      "Corsair Vengeance LPX DDR4 4 3600 C18 2x16GB",
      "G.SKILL Trident Z DDR4 3600 C17 2x8GB",
      "G.SKILL Ripjaws 4 DDR4 2800 C16 4x4GB",
      "Corsair Vengeance LPX DDR4 3600 C18 2x8GB"
    ],
    "USB": [
      "SanDisk Extreme Pro USB 3.0 128GB",
      "Samsung FIT USB 3.0 64GB",
      "SanDisk Extreme USB 3.0 32GB",
      "Kingston DataTraveler HyperX USB 3.0 64GB",
      "BUFFALO HD-LXU3 1TB"
    ]
  }
}
```

### Response Headers

```
{
  "Access-Control-Allow-Origin": ["*"],
  "Content-Type": "application/json"
}
```

# Recommendation Service

**Principle:** Maximize users' satisfaction based on their preferences and budgets.

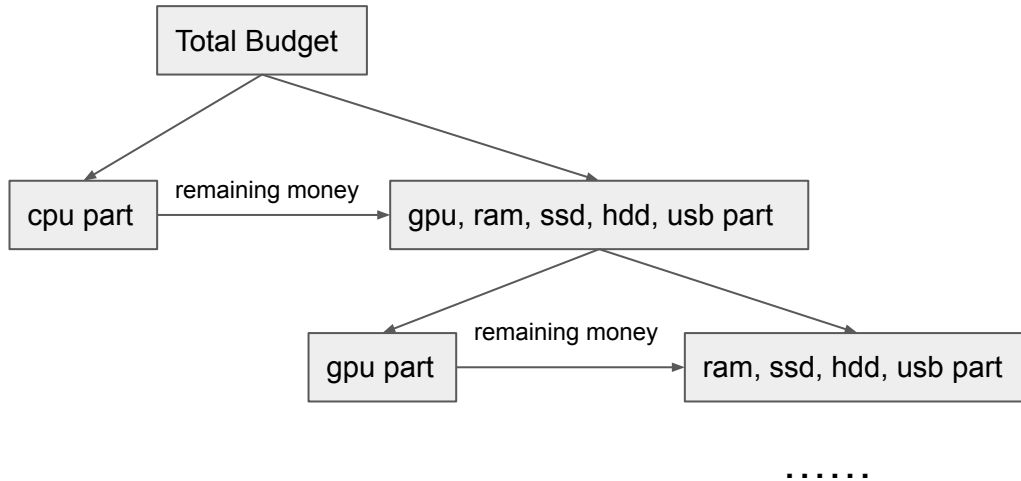
Three Categories & initial models based on UBM performance:

STUDENT: {"CPU": 0.50,"GPU": 0.32, "SSD": 0.05,"HDD": 0.05,"RAM": 0.07, "USB": 0.01}

COMMON\_U: {"CPU": 0.60,"GPU": 0.17, "SSD": 0.10,"HDD": 0.05,"RAM": 0.07, "USB": 0.01}

GAMMER: {"CPU": 0.22,"GPU": 0.60, "SSD": 0.05,"HDD": 0.05,"RAM": 0.07, "USB": 0.01}

Parameters source: [userbenchmark.com](https://userbenchmark.com)

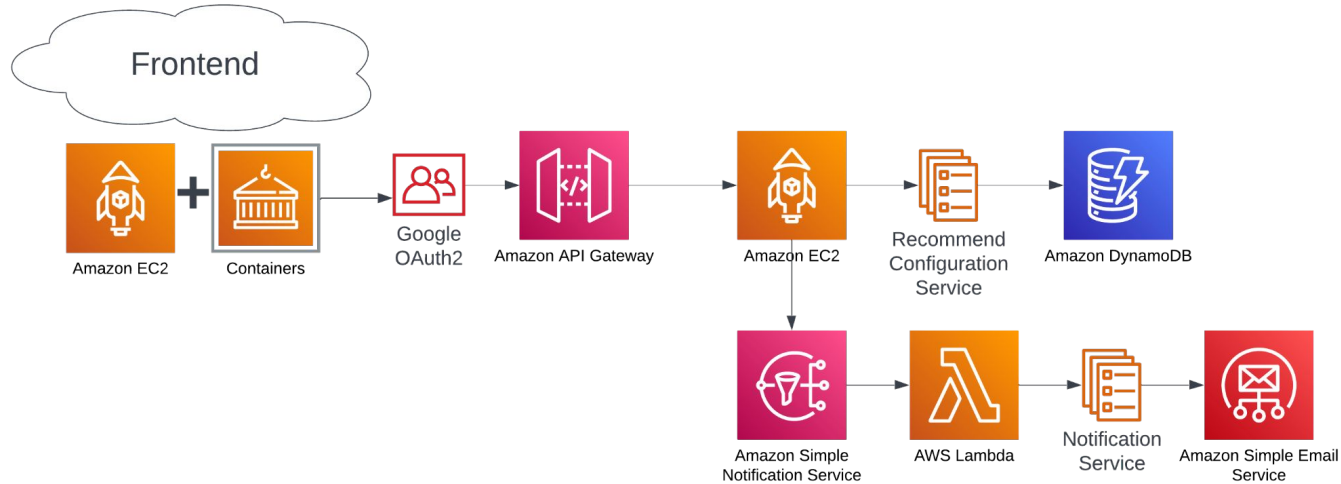


First, allocate cpu budget according to the proportion, the rest of total budget as well as the unused cpu budget go to next part;

For next part, allocate gpu part in relative proportion, the rest of budget as well as the unused gpu budget go to next part;

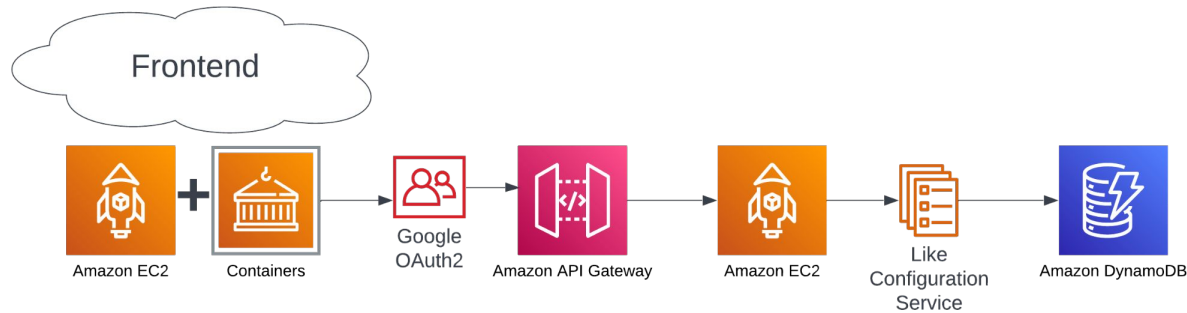
# Email Notification Service

Our service running on EC2 handles the recommendation request submitted by the user and give 5 different configurations with highest benchmark. The results will be sent to AWS SNS, which will trigger AWS Lambda. Next, an email will be sent to user's email address via AWS SES.



# Like Configurations Service

After the user gets the recommended builds, the user can vote for all the currently displayed hardwares by clicking the “Like this result” button below. This service is deployed on EC2, and the data is passed directly to DynamoDB for subsequent services to read.



## Here is our recommendations:

### CPU:

Intel Core i5-13600K

### GPU:

Asus Asus RTX 3080 24GB Tuf OC

### RAM:

G.SKILL Trident Z DDR4 3200 C14 4x16GB

### SSD:

WD Black SN850 NVMe PCIe M.2 1TB

### HDD:

WD Gold 6TB (2016)

### USB:

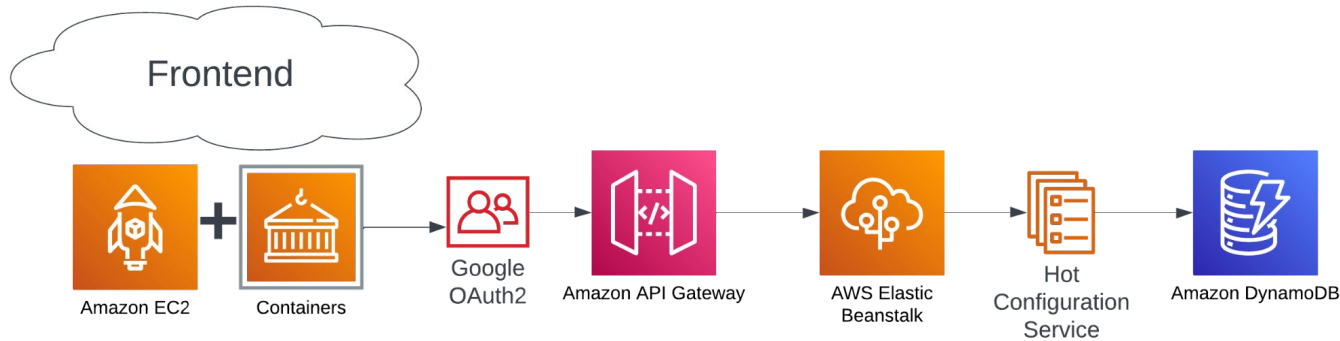
Mushkin Ventura Ultra USB 3.0 60GB

Save

Like this result

# Hot Configurations Service

After the user login into the system, our service will fetch the most liked device in each category and present it to the user. The service is deployed in Elastic Beanstalk and fetches user liked data from DynamoDB which has been indexed by category.



## Top Hits

### Top CPU:

Intel Core i5-13600K

### Top GPU:

Asus Asus RTX 3080 24GB Tuf OC

### Top RAM:

G.SKILL Flare X DDR4 3200 C14 4x8GB

### Top SSD:

WD Black SN850 NVMe PCIe M.2 1TB

### Top HDD:

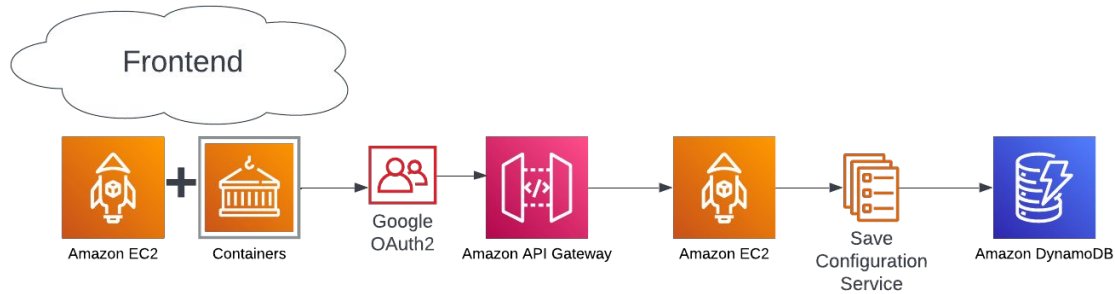
WD VelociRaptor 1TB

### Top USB:

SanDisk Extreme Pro USB 3.0 128GB

# Save Configurations Service

After the user gets the recommended build, the user can save this build by clicking the “Save” button below. This service is deployed on EC2, and the partition key is the user's email, and the full build is stored in DynamoDB.



## Here is our recommendations:

### CPU:

Intel Core i5-13600K

### GPU:

Asus Asus RTX 3080 24GB Tuf OC

### RAM:

G.SKILL Trident Z DDR4 3200 C14 4x16GB

### SSD:

WD Black SN850 NVMe PCIe M.2 1TB

### HDD:

WD Gold 6TB (2016)

### USB:

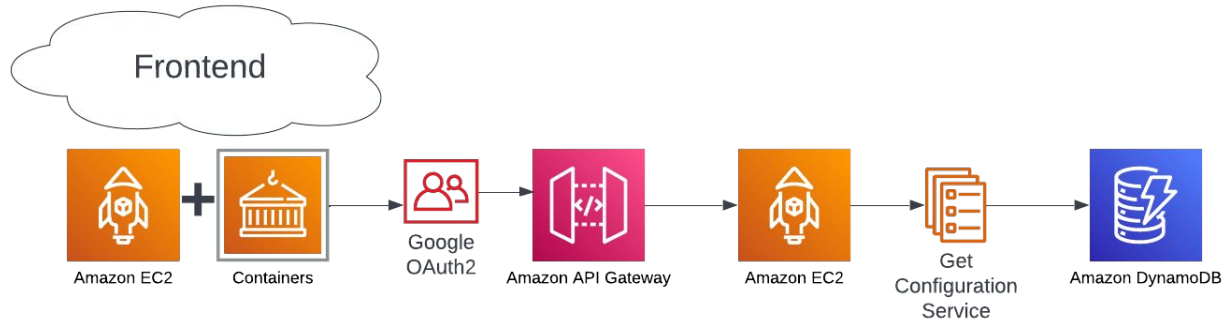
Mushkin Ventura Ultra USB 3.0 60GB

Save

Like this result

# Get Configurations Service

When the user enters the profile page, this service will be called automatically to retrieve their previously saved build. Our service on EC2 will query DynamoDB by user's email address, and send the returned results to this page.



## Your last search result:

### CPU:

Intel Core i5-13600K

### GPU:

Asus Asus RTX 3080 24GB Tuf OC

### RAM:

G.SKILL Trident Z DDR4 3200 C14 4x16GB

### SSD:

WD Black SN850 NVMe PCIe M.2 1TB

### HDD:

WD Gold 6TB (2016)

### USB:

Mushkin Ventura Ultra USB 3.0 60GB

# CI/CD

```
1 name: "Front end CI/CD"
2 on:
3   push:
4     branches: [ "master" ]
5   pull_request:
6     branches: [ "master" ]
7 jobs:
8   build:
9     name: Build
10    runs-on: ubuntu-latest
11    steps:
12      - name: GIT CLONE
13        uses: appleboy/ssh-action@v0.1.6
14        with:
15          host: 3.234.246.243
16          username: centos
17          key: ${ secrets.SSH_USER }}
18          script: cd frontend_with_user && git pull
19      - name: BUILD
20        uses: appleboy/ssh-action@v0.1.6
21        with:
22          username: centos
23          host: 3.234.246.243
24          key: ${ secrets.SSH_USER }}
25          script: cd frontend_with_user && podman build -t 6156 -f Dc
26      - name: RUN
27        uses: appleboy/ssh-action@v0.1.6
28        with:
29          host: 3.234.246.243
30          username: centos
31          key: ${ secrets.SSH_USER }}
32          script: podman run -d -e PORT=8000 -p 8000:8000 localhost/6156:latest
```

```
1 name: "lambda 1 CI/CD"
2 on:
3   push:
4     branches: [ "main" ]
5   pull_request:
6     branches: [ "main" ]
7 jobs:
8   build:
9     name: Build
10    runs-on: ubuntu-latest
11    steps:
12      - name: GIT CLONE
13        uses: appleboy/ssh-action@v0.1.6
14        with:
15          host: 54.234.250.231
16          username: ec2-user
17          key: ${ secrets.EC2_SSH_KEY }}
18          script: cd ~/e6156-lambda-1-ec2 && git pull
19      - name: RUN
20        uses: appleboy/ssh-action@v0.1.6
21        with:
22          key: ${ secrets.EC2_SSH_KEY }}
23          host: 54.234.250.231
24          username: ec2-user
25          script: cd ~/e6156-lambda-1-ec2 && python3 lambda_1
```

```
36 deploy:
37   # Only run this job if "build" has ended successfully
38   needs:
39     - build
40
41   runs-on: ubuntu-latest
42
43   steps:
44     # Checks-out your repository under $GITHUB_WORKSPACE
45     - uses: actions/checkout@v2
46
47     # Set up Python 3.7 environment
48     - name: Set up Python 3.7
49       uses: actions/setup-python@v1
50       with:
51         python-version: "3.7"
52
53     # Elastic Beanstalk CLI version
54     - name: Get EB CLI version
55       run: |
56         python -m pip install --upgrade pip
57         pip install awsebcli --upgrade
58         eb --version
59
60     # Configure AWS Credentials
61
62     # Configure AWS Credentials
63     - name: Configure AWS Credentials
64       uses: aws-actions/configure-aws-credentials@v1
65       with:
66         aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }}
67         aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }}
68         aws-region: us-east-1
69
70     # Create the Elastic Beanstalk application
71     - name: Create EBS application
72       run: |
73         eb init -p python-3.7 get-hot-config --region us-east-1
74
75     # Deploy to (or Create) the Elastic Beanstalk environment
76     - name: Create test environment & deploy
77       run: |
78         eb deploy hot-config-env
```



# Demo

**Thank you!!**

# Appendix 1. How to Compute Price Allocation ratio

Percentage measures of a PC's suitability for gaming, desktop and workstation use.



## Gaming

Average fps, and more importantly, 0.1% and 1% lows.

2060S + 9600K  $\approx$  100%. Gaming is primarily influenced by the GPU:

64% GPU Bench, 24% CPU Bench, 7% RAM Bench, 5% SSD Bench

CPU Bench: Quad > Octa



## Desktop

Web browsing, office apps, music/video playback.

Desktop tasks are mainly dependent on single/quad core CPU performance:

73% DCPU Bench, 14% GPU Bench, 10% SSD Bench, 3% RAM Bench

DCPU Bench: Single > Quad



## Workstation

Audio/video encoding, number crunching, virtual machines, databases:

Workstation tasks require strong multi core CPU and GPU performance:

54% WCPU Bench, 34% GPU Bench, 7% RAM Bench, 5% SSD Bench

WCPU Bench: 64-core > Octa > Quad > Single

[link](#)