# AWS Tennis Court Booking Service

**Group 3:** Olga Bienzobas, Erik Claros, Anousith Keomaly

# *Our solution:*

# *Tennis Court Booking Website*

Our solution is an intuitive platform for users to create court reservations. Users are able to pick a court and see whether or not it is reserved during specific time frames. Users can also use another site to see the general hours of operation.

**Front-end:** HTML          **Back-end:** PHP & Javascript          **Database:** MySQL
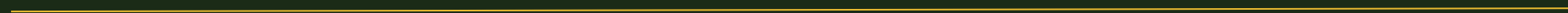
# Deployment Components

1. VPC Networking

2. Resources/Storage

3. Security

4. Scaling/Load Balancing

5. Containerization/Orchestration

# VPC Networking

## VPC

- VPC "lab03-vpc" houses our solution

- Has 2 Availability Zones: Us-east-1a and Us-east1b

- 1 public and private subnet per AZ (4 subnets in total)

| | lab03-vpc | vpc-0980b4f0fbc7abca3 | ⊘ Available | 10.0.0.0/16 |

# Networking

| | Name | Subnet ID | State | VPC | IPv4 CIDR |
|---|---|---|---|---|---|
| ☐ | lab03-subnet-private1-us-east-1a | subnet-08113d7adba3be52e | ✓ Available | vpc-0980b4f0fbc7abca3 \| lab03... | 10.0.2.0/24 |
| ☐ | lab03-subnet-public1-us-east-1a | subnet-0808100cea8c4bfb3 | ✓ Available | vpc-0980b4f0fbc7abca3 \| lab03... | 10.0.0.0/24 |
| ☐ | lab03-subnet-private2-us-east-1b | subnet-08fb9c5c7a86ae53f | ✓ Available | vpc-0980b4f0fbc7abca3 \| lab03... | 10.0.3.0/24 |
| ☐ | lab03-subnet-public2-us-east-1b | subnet-0f09a4e4bbcf7b06c | ✓ Available | vpc-0980b4f0fbc7abca3 \| lab03... | 10.0.1.0/24 |

subnets in our VPC

# Resources/Storage

## EC2 Instances

- Bastion uses an amazon image. It allows remote connection through SSH from anywhere. From there, you can SSH into other instances which accepts only local traffic. It's in lab03-subnet-public2-us-east1b.

- Benchmark server is an ubuntu instance that is on lab03-subnet-public2-us-east-1b.

- Booking Website uses an amazon image in public subnet 1. It retrieves information from the RDS and displays whether or not a court is taken during some time slot. It's on lab03-subnet-public2-us-east1b.
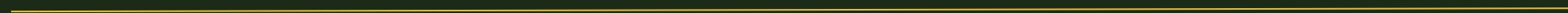
# Booking Website

# Resources/Storage

## RDS

- We've used RDS as a database. It lies on both lab03-subnet-private1-us-east-1a and lab03-subnet-private2-us-east-1b, It contains information about whether or not a court is open during a certain time slot.

```
+----------+------------+----------+--------+
| court_id | court_name | timeslot | isOpen |
+----------+------------+----------+--------+
|        1 | 1          | 8-9 AM   |      1 |
|        2 | 1          | 1-5 PM   |      0 |
|        3 | 2          | 8-9 AM   |      1 |
|        4 | 2          | 1-5 PM   |      1 |
|        5 | 3          | 8-9 AM   |      0 |
|        6 | 3          | 1-5 PM   |      1 |
|        7 | 4          | 8-9 AM   |      1 |
|        8 | 4          | 1-5 PM   |      0 |
|        9 | 5          | 8-9 AM   |      1 |
|       10 | 5          | 1-5 PM   |      1 |
|       11 | 6          | 8-9 AM   |      0 |
|       12 | 6          | 1-5 PM   |      1 |
|       13 | 7          | 8-9 AM   |      1 |
|       14 | 7          | 1-5 PM   |      0 |
|       15 | 8          | 8-9 AM   |      1 |
|       16 | 8          | 1-5 PM   |      0 |
+----------+------------+----------+--------+
```

# Resources/Storage

## EBS

- Bastion has extra storage mounted onto it after we attached an EBS block to it. (/dev/xvdf)

```
[ec2-user@ip-10-0-0-128 ~]$ df -h
Filesystem          Size    Used  Avail  Use%  Mounted on
devtmpfs            4.0M      0   4.0M    0%  /dev
tmpfs               475M      0   475M    0%  /dev/shm
tmpfs               190M   448K   190M    1%  /run
/dev/xvda1          8.0G   1.6G   6.5G   20%  /
tmpfs               475M      0   475M    0%  /tmp
/dev/xvda128        10M    1.3M   8.7M   13%  /boot/efi
tmpfs                95M      0    95M    0%  /run/user/1000
/dev/xvdf           975M    60K   924M    1%  /mnt/data-store
```

# Resource/Storage

| | Name | ▲ | AWS Region | ▽ | IAM Access Analyzer |
|---|---|---|---|---|---|
| ○ | finalprojectlab3 | | US East (N. Virginia) us-east-1 | | View analyzer for us-east-1 |

**Objects (2)** Info

[ ↻ ] [ ⧉ Copy S3 URI ] [ ⧉ Copy URL ] [ ⬇ Download ] [ Open ↗ ] [ Delete ] [ Actions ▼ ] [ Create folder ] [ ⬆ Upload ]

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory ↗ to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more ↗

🔍 Find objects by prefix                                                                                    ‹ 1 › ⚙

| | Name | ▲ | Type | ▽ | Last modified | ▽ | Size | ▽ | Storage class | ▽ |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 📄 benchmark.yaml | | yaml | | August 4, 2024, 23:49:39 (UTC-05:00) | | 2.6 KB | | Standard | |
| ☐ | 📄 tenniscourts.png | | png | | August 4, 2024, 23:50:44 (UTC-05:00) | | 1.9 MB | | Standard | |

S3 bucket "finalprojectlab3" holds the image of the tennis court from the Booking Site and the yaml file to create a benchmark server.

# Security



KMS > Customer managed keys > Key ID: 4f9fe811-6564-4a9a-ade2-5cb9edd87464

## 4f9fe811-6564-4a9a-ade2-5cb9edd87464

### General configuration

**Alias**
IAC-Key

**ARN**
⧉ arn:aws:kms:us-east-1:084409326317:key/4f9fe811-6564-4a9a-ade2-5cb9edd87464

**Status**
Enabled

**Description**
This is for the YAML File

**Creation date**
Aug 03, 2024 14:54 CDT

**Regionality**
Single Region

S3 bucket was encrypted with a KMS-SSE key

# Security

## Security groups

- Security groups were made for each of the instances as well as the RDS. It controls what traffic goes through it though all are able to send any traffic outwards.

-  Webserver enables HTTP traffic from any host and SSH traffic within local network

-  Website2 enables HTTP traffic and port 8080 anywhere and SSH traffic locally.

- Database enables MySQL traffic within local network

-  Bastion enables SSH traffic from any host

- Benchmark enables SSH traffic and HTTP access anywhere.

# Monitoring

# Scaling/Load Balancing

## Load Balancing

- Website instance is registered under the ELB. There are a minimum of 2 instances running at a time and a maximum of 6.

## Horizontal Scaling

- Whenever Cloudwatch detects traffic to the booking page load balancer's exceeding 60% , more instances of the Web server are created.
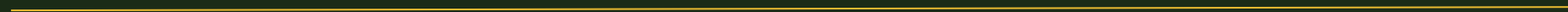
# Scaling/Load Balancing

| | Name ✎ ▽ | Instance ID | Instance state ▽ | Instance type ▽ |
|---|---|---|---|---|
| ☐ | Lab Instance | i-06557811e966e5889 | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | | i-0850204f81b9615ff | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | Container | i-05aad3cab2f2b289c | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | | i-0613d5f5732c88c95 | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | Website | i-01588c1c888f9a286 | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | Lab Instance | i-0a3607b2920070c32 | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | Bastion | i-0e6ed84ba1eff837b | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | Lab Instance | i-068865f46b994e110 | ⊘ Running ⊕ ⊖ | t2.micro |
| ☐ | Benchmarking... | i-0f82ed19980c1c744 | ⊘ Running ⊕ ⊖ | t2.micro |

2 extra Instances called "Lab Instance" were formed by horizontal scaling. They use the AMI from our Website page.

# Stress Testing

**CPU utilization (%)**

Percent

8

4

0

07:15  07:30  07:45  08:00  08:15

**Network in (bytes)**

Bytes

4.32M

2.16M

0

07:15  07:30  07:45  08:00  08:15

**Network out (bytes)**

Bytes

4.52M

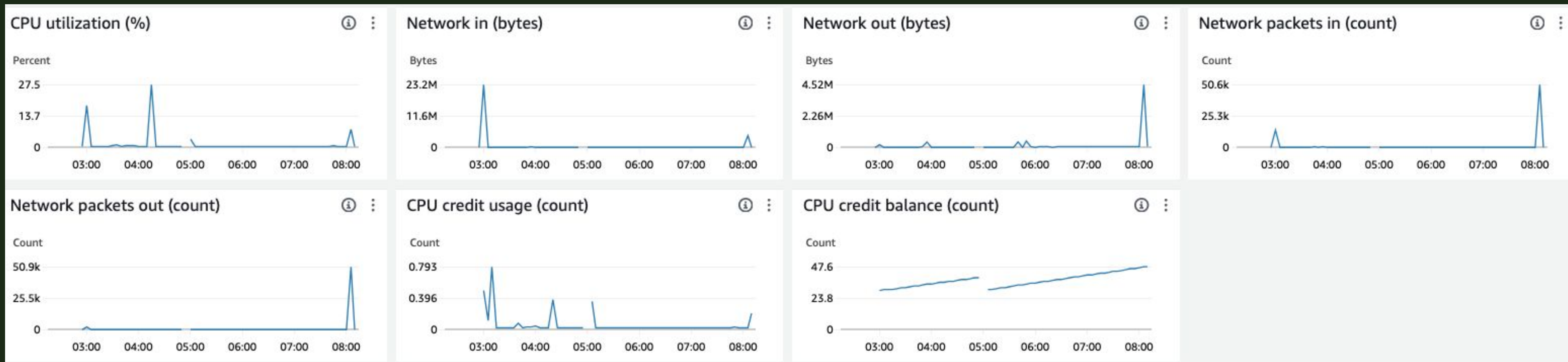2.26M

0

07:15  07:30  07:45  08:00  08:15

**Network packets in (count)**

Count

50.6k

25.3k

0

07:15  07:30  07:45  08:00  08:15

**Network packets out (count)**

Count

50.9k

25.5k

0

07:15  07:30  07:45  08:00  08:15

**CPU credit usage (count)**

Count

0.21

0.105

0

07:15  07:30  07:45  08:00  08:15

**CPU credit balance (count)**

Count

47.6

23.8

0

07:15  07:30  07:45  08:00  08:15

## 6 hours scale

**CPU utilization (%)**

Percent

27.5

13.7

0

03:00  04:00  05:00  06:00  07:00  08:00

**Network in (bytes)**

Bytes

23.2M

11.6M

0

03:00  04:00  05:00  06:00  07:00  08:00

**Network out (bytes)**

Bytes

4.52M

2.26M

0

03:00  04:00  05:00  06:00  07:00  08:00

**Network packets in (count)**

Count

50.6k

25.3k

0

03:00  04:00  05:00  06:00  07:00  08:00

**Network packets out (count)**

Count

50.9k

25.5k

0

03:00  04:00  05:00  06:00  07:00  08:00

**CPU credit usage (count)**

Count

0.793

0.396

0

03:00  04:00  05:00  06:00  07:00  08:00

**CPU credit balance (count)**

Count

47.6

23.8

0

03:00  04:00  05:00  06:00  07:00  08:00

# Containerization/Orchestration

## Containerization

- A Docker image was created from the Hours-of-operation Webserver. It is stored under ECR in the private repository "operating-hours"

# Containerization/Orchestration



**Images** (1)

| | Image tag | ▽ | Artifact type | Pushed at |
|---|---|---|---|---|
| ☐ | latest | | Image | August 05, 2024, 01:56:45 (UTC-05) |

Docker image stored in repository "operating-hours"

# Containerization/Orchestration

## Orchestration

- From the URL of the image in ECR, we created a cluster in ECS called "hours-of-operation"

# Containerization/Orchestration

## Orchestration

- There, we can run a task so we can connect to the web page hosted by the ECS Service.

# Hours of Operation Website

It's on lab03-subnet-public1-us-east1a.

# Challenges Faced

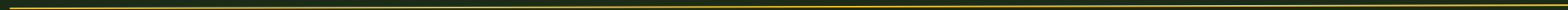- Debugging
- Communication between front end and back end
- Security groups
- Redoing the lab
- IAM Permissions



**Abhijeet Soni**
@__abhij33t__

10 Stages of Debugging:

🤔
😕
😫
🤔
😡
😲
🤦‍♂️
😌
😅
😎

# Lessons Learned

- Don't over complicate things
- Better time management
- Security Best Practices
- Look at Documentation

- **Backups!!!!!!!**

# Demo