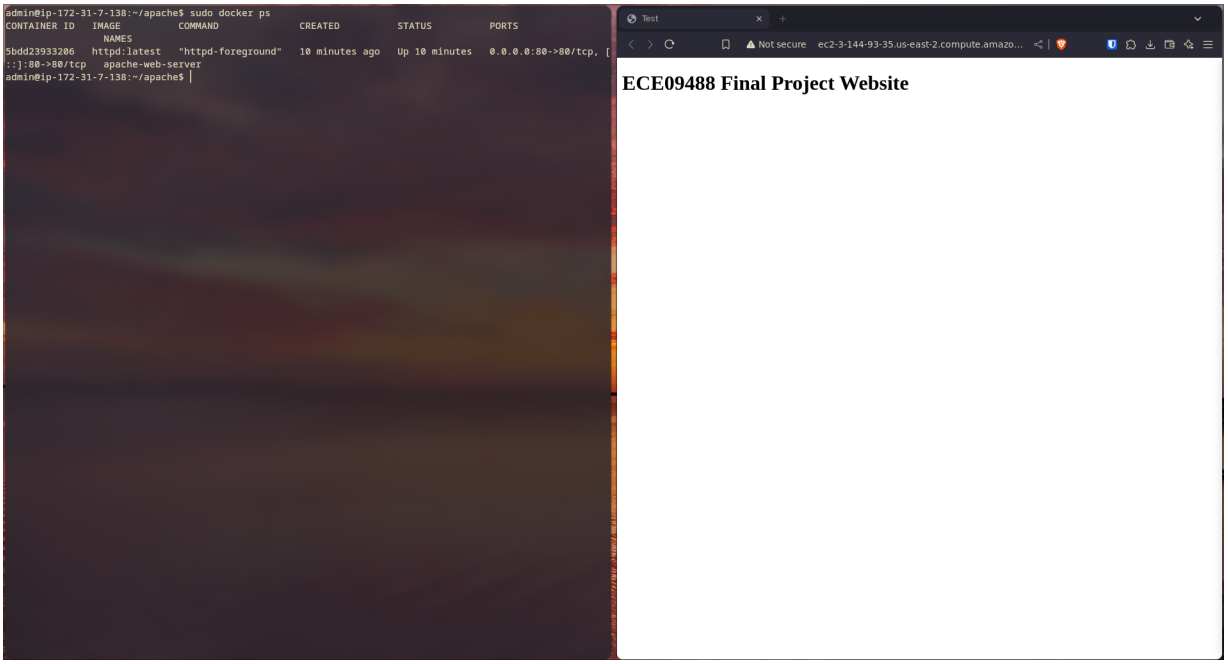
ECE09488 Final Project

Aidan Sharpe

May 2nd, 2025

Part 1: Host a Website

I created an EC2 instance running Debian. I downloaded the OpenSSH key to my computer, and SSH-ed into the VM. From there, I installed Docker, and created a folder for an Apache Container. Then, I copied a Docker compose file to configure the container. I added a file called `index.html`, where I added a bare bones "Hello World" HTML file. Finally, I configured the firewall rules on my EC2 instance to allow HTTP traffic on port 80 (no sense messing around setting up HTTPS). As seen below, I was able to access the web page using the DNS link provided by AWS.



Part 2: Given a Scenario

Your company, TechCo, has been successfully running its critical applications on a public cloud platform for several years. These applications handle sensitive customer data, financial transactions, and inventory management. The cloud provider offers robust infrastructure and high availability, but unforeseen events can still disrupt services.

Although TechCo has been lucky, they know they must plan for unfortunate events. Your management wants to know how Cloud an be used for disaster recovery and business continuity.

Background and Assumptions

Every company should be thinking about disaster recovery well before any disaster has the chance to occur. It is critical in any high availability infrastructure to have in place a disaster recovery plan. We will assume that the cloud provider has 99.999% (5-nines) uptime, which averages to about 5 minutes of downtime per year. We will also assume that some or most of our customer's data is sensitive, and therefore, we will have to treat all data as sensitive. This means following HIPPA laws, as some of our customers may hold medical records. Finally, we will assume that we are operating out of the Eastern United States.

Technical Solution

To start, I would recommend one of the big three cloud providers (GCP, AWS, Azure). Due to their sheer size, they have redundant datacenters and are able to take the most advantage of economies of scale. Therefore, our infrastructure will be more robust and cost-effective. All three providers have similar (yet incompatible) options when it comes to redundancy and security, so for the sake of choosing one, we will go with AWS, as it has been highlighted the most throughout this course.

Architecture

We will start our redundant system by setting up two VPCs. The primary VPC will be located on an EC2 instance located on `us-east-1` and our secondary instance will operate on `us-west-2`. Our database (lets assume we use Aurora) will be replicated accross both instances. Then, in the event of an outage, we can use Amazon Route 53 to redirect DNS failures to our secondary instance. We can automate this process using a health check trigger. For backups, we will use S3 Glacier for long-term cold storage.

Why Cloud?

The cloud is the best platform for this infrastructure, as cloud technology offers a wide array of tools to enable redundancy and automatic backups. It also enables easy monitoring, in our case we would use Amazon CloudWatch. The cloud also ensures that our services can scale with the growth of our company.

Security Features

All traffic must be end-to-end encrypted to protect data in transit. We will also block all unused ports in our firewall to reduce our attack surface. The data stored in our Aurora database shall also be encrypted without any possibility of data access on our part. The customers only will be responsible for holding the private keys. This will prevent our employees from accessing customer data, and reduce our responsibility for maintaining and protecting the private keys. Generally, we will enforce the principle of least privilege using IAM rules and policies. We will also use AWS GuardDuty for threat detection. Finally, we will protect ourselves from DDoS attacks using AWS WAF.

Cost Breakdown

Without any idea of the scale of TechCo and the number of customers, I cannot assume the monthly cost. To calculate the cost would involve the following:

Compute Costs

- 2 EC2 instances
- Load balancer

Database Costs

- 1 active (primary) database
- 1 read-only (backup) database

Storage Costs

- 2 S3 buckets for cross-region data replication
- S3 Glacier archival backups

Network Costs

- Route 53
- Automatic failover
- DNS health check

Monitoring Costs

- CloudWatch with logs

Security Costs

- GuardDuty
- IAM
- WAF

Other

- Data transfer from primary to backup instance

Conclusions

The cloud offers a plethora of tools to enable secure, reliable services. Unfortunately, simply spinning up a VM will not give you all the benefits of the cloud. To fully take advantage of the benefits of CSPs, a lot of work has to be done ahead of time to plan for and prevent disasters. This involves setting up redundant systems, creating regular backups in archival storage, protecting customer data, and automating monitoring and failovers.

Even still, no matter how much preparation work you do, it will never be enough. You can only protect yourself against known uncertainties, yet it is the unknown unknowns that have the potential to sneak up and cause the most damage. So when disaster inevitably strikes, it is absolutely critical to have a safety net to fall back on.