MIT World Peace University

Cloud Infrastructure and Security
Third Year B. Tech, Semester 6

Deployment on AWS and Load Balancing

Assignment 3

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

March 28, 2024

Contents

1 Aim

Create an account on AWS. Deploy a Website admission portal on EC2 Service. Configure the traffic rules for the server for a specific need. Creation of application load balancer.

2 Objectives

1. To get acquainted with AWS platform.
2. To get familiar with EC2 Service
3. Learn how to configure the traffic rules.
4. To understand creation of application load balancer

3 Theory

3.1 Introduction to AWS (Amazon Web Services)

Amazon Web Services (AWS) is a comprehensive, evolving cloud computing platform provided by Amazon. It provides a mix of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings. AWS services can offer an organization tools such as compute power, database storage and content delivery services. Moving to the cloud with AWS can help organizations cut costs, improve efficiency and innovate at a faster pace.

3.2 Features of AWS

1. **Compute:** AWS offers a wide range of compute services, including Amazon Elastic Compute Cloud (EC2), AWS Lambda, Amazon Elastic Container Service (ECS), and more. These services provide scalable compute capacity for running applications and workloads.
2. **Storage:** AWS provides various storage services, such as Amazon Simple Storage Service (S3), Amazon Elastic Block Store (EBS), Amazon Glacier, and more. These services offer scalable, durable, and cost-effective storage solutions for data storage and backup.
3. **Database:** AWS offers a range of database services, including Amazon Relational Database Service (RDS), Amazon DynamoDB, Amazon Redshift, and more. These services provide managed database solutions for different use cases and workloads.
4. **Networking:** AWS provides networking services such as Amazon Virtual Private Cloud (VPC), Amazon Route 53, AWS Direct Connect, and more. These services enable users to build secure and scalable network architectures for their applications.
5. **Security:** AWS offers a wide range of security services, including AWS Identity and Access Management (IAM), AWS Key Management Service (KMS), AWS WAF, and more. These services help users secure their applications, data, and infrastructure in the cloud.
6. **Management Tools:** AWS provides management tools such as AWS CloudFormation, AWS CloudWatch, AWS Config, and more. These tools help users automate, monitor, and manage their AWS resources and services.

7. **Analytics:** AWS offers analytics services such as Amazon Kinesis, Amazon EMR, Amazon Redshift, and more. These services help users collect, process, analyze, and visualize data to derive insights and make informed decisions.

8. **Machine Learning:** AWS provides machine learning services such as Amazon SageMaker, Amazon Rekognition, Amazon Comprehend, and more. These services enable users to build, train, and deploy machine learning models at scale.

9. **Internet of Things (IoT):** AWS offers IoT services such as AWS IoT Core, AWS IoT Greengrass, AWS IoT Device Management, and more. These services help users connect, manage, and secure IoT devices and applications.

10. **Serverless Computing:** AWS provides serverless computing services such as AWS Lambda, Amazon API Gateway, AWS Step Functions, and more. These services enable users to build and deploy applications without managing servers.

11. **Containers:** AWS offers container services such as Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), AWS Fargate, and more. These services help users run and manage containerized applications at scale.

12. **DevOps:** AWS provides DevOps services such as AWS CodePipeline, AWS CodeBuild, AWS CodeDeploy, and more. These services help users automate software development and deployment processes.

13. **Artificial Intelligence (AI):** AWS offers AI services such as Amazon Lex, Amazon Polly, Amazon Translate, and more. These services enable users to build AI-powered applications and services.

14. **Blockchain:** AWS provides blockchain services such as Amazon Managed Blockchain, Amazon Quantum Ledger Database (QLDB), and more. These services help users build scalable and secure blockchain applications.

15. **Augmented Reality (AR) and Virtual Reality (VR):** AWS offers AR and VR services such as Amazon Sumerian, Amazon S3 Glacier Deep Archive, and more. These services help users create immersive AR and VR experiences.

16. **Game Development:** AWS provides game development services such as Amazon GameLift, Amazon Lumberyard, and more. These services help users build, deploy, and scale games in the cloud.

3.3 Steps /procedure to follow for an AWS account creation

1. **Sign Up for AWS:** Go to the AWS website and click on the "Create an AWS Account" button. Follow the on-screen instructions to create a new AWS account.

2. **Provide Account Information:** Enter your email address, password, and account name. Click on the "Continue" button to proceed.

3. **Contact Information:** Enter your contact information, including name, address, and phone number. Click on the "Create Account and Continue" button.

4. **Payment Information:** Enter your payment information, including credit card details. AWS may charge a small amount to verify your identity.

5. **Identity Verification:** AWS will verify your identity by sending a verification code to your phone number. Enter the code to complete the verification process.

6. **Choose a Support Plan:** Choose a support plan based on your requirements. You can choose the Basic (free) plan or opt for a paid plan for additional support.

7. **Confirmation:** Review the account details and click on the "Create Account and Continue" button to complete the account creation process.

8. **Sign In:** Sign in to your new AWS account using your email address and password. You can now access the AWS Management Console and start using AWS services.

3.4 Briefing on EC2 Service

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

3.4.1 Features

1. **Virtual Servers:** Amazon EC2 provides virtual servers, known as instances, that can be launched in minutes. You can choose from a variety of instance types with different compute, memory, and storage configurations.

2. **Scalability:** Amazon EC2 allows you to scale your compute capacity up or down based on your application's needs. You can launch multiple instances to handle increased traffic or reduce the number of instances during off-peak hours.

3. **Security:** Amazon EC2 provides security features such as security groups, network access control lists (ACLs), and key pairs to secure your instances. You can also use AWS Identity and Access Management (IAM) to control access to your resources.

4. **Integration:** Amazon EC2 integrates with other AWS services such as Amazon S3, Amazon RDS, and Amazon VPC. You can easily deploy applications that use multiple AWS services and resources.

5. **Monitoring:** Amazon EC2 provides monitoring and logging features through Amazon CloudWatch. You can monitor your instances' performance, set alarms, and view logs to troubleshoot issues.

6. **Auto Scaling:** Amazon EC2 Auto Scaling allows you to automatically adjust the number of instances in your Auto Scaling group based on demand. You can define scaling policies to scale in or out based on metrics such as CPU utilization or request count.

7. **Load Balancing:** Amazon EC2 works with Elastic Load Balancing to distribute incoming traffic across multiple instances. You can use Application Load Balancers or Network Load Balancers to achieve high availability and fault tolerance.

8. **Pricing:** Amazon EC2 offers a pay-as-you-go pricing model with no upfront costs or long-term contracts. You only pay for the compute capacity you use, and you can choose from various pricing options based on your requirements.

3.5 Deploying the website using EC2 service

3.6 Configure the traffic rules for the server

Steps:

1. Accessing the Security Group:

   There are two primary ways to configure traffic rules for an EC2 instance:

   (a) Using the AWS Management Console:

      i. Go to the Amazon EC2 service in the AWS Management Console (https://aws.amazon.com/).

      ii. In the navigation pane on the left, select the option for Security Groups.

      iii. Choose the security group you want to modify. This security group will be assigned to your EC2 instance and controls the inbound and outbound traffic.

   (b) Using the AWS CLI: AWS provides a Command Line Interface (CLI) for managing various AWS services, including EC2 security groups. You can find instructions and reference guides for the AWS CLI here: https://aws.amazon.com/cli/

2. Adding Inbound Rules:

   (a) Inbound rules define which types of traffic are allowed to reach your EC2 instance from the internet or your VPC (Virtual Private Cloud).

   (b) To add an inbound rule, click on the "Edit inbound rules" button in the AWS Management Console.

3. Specifying Rule Details:

   When adding an inbound rule, you'll need to specify several details:

   (a) Protocol: Choose the communication protocol for the traffic (e.g., TCP, UDP, ICMP).

   (b) Port Range: Specify the port(s) on your EC2 instance that this rule applies to. For example, port 22 for SSH access or port 80 for web traffic. You can also open a range of ports if needed.

   (c) Source: Define the source of the traffic. You can allow traffic from anywhere (0.0.0.0/0 for IPv4 or ::/0 for IPv6) or restrict it to specific IP addresses, security groups, or VPCs.

4. Adding Outbound Rules (Optional):

   (a) Outbound rules define what type of traffic your EC2 instance can initiate towards the internet or other resources. By default, outbound traffic is allowed, but you can configure restrictions if needed.

   (b) The process for adding outbound rules is similar to adding inbound rules. You'll specify the protocol, port range, and destination for the outbound traffic.

5. Saving the Changes:

   Once you've added or modified the desired rules, click on the "Save" button to apply the changes to the security group.

Important Considerations:

Security Best Practices: It's recommended to follow security best practices when configuring traffic rules. Don't open unnecessary ports or allow access from unrestricted sources.

Specific Needs: Tailor the rules to your specific application or service running on the EC2 instance.

Testing: After making changes to the security group, it's advisable to test your application or service to ensure it functions as expected with the new rules in place.

3.7 Creating Application load balancer

1. **Prepare Your EC2 Instances:**
    - Launch or ensure you have EC2 instances running your application.
    - Make sure your application is properly configured and running on these instances.
2. **Create a Target Group:**
    - Go to the Amazon EC2 dashboard.
    - Under "Load Balancing," navigate to "Target Groups."
    - Click on "Create target group."
    - Specify a name for your target group, protocol, port, and VPC.
    - Define the health checks for your target instances.
3. **Create an Application Load Balancer:**
    - In the EC2 dashboard, under "Load Balancing," navigate to "Load Balancers."
    - Click on "Create Load Balancer."
    - Choose "Application Load Balancer" as the type.
    - Configure the load balancer settings, including name, listeners, availability zones, and security settings.
    - Associate the previously created target group with the load balancer.
4. **Configure Listener Rules:**
    - Define the listener rules to route traffic to the appropriate target group based on the request characteristics (e.g., path patterns, host headers).
5. **Update DNS Records (Optional):**
    - If you're using a custom domain, update your DNS records to point to the DNS name of your load balancer.
6. **Test Your Load Balancer:**
    - Ensure that your load balancer is routing traffic to your EC2 instances correctly.
    - Test your application to verify that it is functioning as expected with the load balancer in place.
7. **Monitor and Scale:**

- Monitor the performance and health of your load balancer and EC2 instances using Amazon CloudWatch metrics.
- Configure auto-scaling policies to automatically adjust the number of EC2 instances based on demand.

4 Platform

Operating System: Windows 11
IDEs or Text Editors Used: Visual Studio Code

5 FAQs

1. **How do you deploy a website on AWS using EC2?**
   - Launch an EC2 instance and select an appropriate instance type based on your website's requirements.
   - Choose a suitable Amazon Machine Image (AMI) for your EC2 instance, such as Amazon Linux or Ubuntu.
   - Configure security groups to control inbound and outbound traffic to your EC2 instance, allowing access to HTTP (port 80) and HTTPS (port 443) if your website uses SSL/TLS.
   - Connect to your EC2 instance using SSH or Remote Desktop Protocol (RDP) and set up the necessary software stack for hosting your website, such as a web server (e.g., Apache, Nginx) and a database server (e.g., MySQL, PostgreSQL).
   - Upload your website files to the appropriate directory on the EC2 instance, typically the web server's document root.
   - Configure the web server to serve your website and test the deployment to ensure it's accessible over the internet.
   - Optionally, assign an Elastic IP (EIP) to your EC2 instance to ensure a static IP address for your website.
2. **What's the purpose of an Application Load Balancer (ALB) in this setup?**
   - An Application Load Balancer (ALB) helps distribute incoming traffic across multiple EC2 instances hosting the website, improving availability, fault tolerance, and scalability.
   - It acts as a single point of contact for clients, routing HTTP and HTTPS requests to the appropriate EC2 instances based on configured rules.
   - The ALB provides advanced routing features such as path-based routing, host-based routing, and support for multiple SSL certificates, enabling flexible traffic management for different parts of the website.
   - Additionally, the ALB performs health checks on EC2 instances and automatically routes traffic away from unhealthy instances, ensuring high availability and reliability of the website.
3. **How do you ensure security for your EC2 instances?**

- Utilize AWS Identity and Access Management (IAM) to manage user access and permissions for interacting with EC2 instances.
- Configure security groups to control inbound and outbound traffic, allowing only necessary ports and protocols.
- Implement network access control lists (ACLs) at the subnet level to filter traffic entering and leaving the EC2 instances.
- Regularly update and patch the operating system and software running on EC2 instances to address security vulnerabilities.
- Enable encryption for data at rest using AWS Key Management Service (KMS) and encrypt data in transit using SSL/TLS.
- Utilize AWS services such as AWS Config, AWS CloudTrail, and Amazon GuardDuty to monitor and detect security threats and compliance violations.

4. **Why is scalability important for a web application?**

   - Scalability allows a web application to handle increasing amounts of traffic, user requests, and data without experiencing performance degradation or downtime.
   - It ensures that the application can grow or shrink its resource capacity dynamically in response to changing demand, improving user experience and satisfaction.
   - Scalability enables efficient resource utilization, cost optimization, and the ability to accommodate spikes in traffic during peak periods or special events.
   - It enhances the application's availability, fault tolerance, and resilience by distributing workloads across multiple resources and avoiding single points of failure.
   - Scalability is essential for meeting business objectives, accommodating business growth, and maintaining competitiveness in the digital marketplace.

5. **Can you explain how the ALB helps with scalability and high availability?**

   - The ALB distributes incoming traffic evenly across multiple EC2 instances, allowing the web application to handle a higher volume of requests and scale horizontally.
   - It automatically detects unhealthy instances and reroutes traffic to healthy instances, ensuring continuous availability and reliability of the application.
   - The ALB supports dynamic scaling by integrating with AWS Auto Scaling, allowing EC2 instances to be added or removed based on demand, traffic patterns, or other metrics.
   - With features like path-based routing and host-based routing, the ALB enables flexible traffic management and can direct requests to different EC2 instances based on specific criteria, improving performance and resource utilization.
   - By offloading SSL/TLS termination, the ALB reduces the computational overhead on EC2 instances, improving their efficiency and scalability.

6 Conclusion

In this assignment, we have learned about web services, PaaS tools, and the deployment of web services using Python on Cyclic. We have explored the different types of web services, including SOAP and RESTful services, and the standards and protocols used in web service development.

We have also discussed the architecture of web services, the components involved, and the security mechanisms used to protect sensitive data. Finally, we have demonstrated the deployment of a web service using FastAPI and Cyclic, showcasing the process of creating, deploying, and accessing web services on a cloud platform.

References

[1] Amazon Web Services. *AWS Documentation*. Available at: `https://docs.aws.amazon.com/`.