Day9 of learning AWS

14 February 2025 09:04 AM

Creating and Testing Your Load Balancer

This tutorial offers a practical introduction to Application Load Balancers using the AWS Management Console, an online interface. Follow these steps to create your initial Application Load Balancer.

Step 1: Configure Your Target Group

To set up a target group for managing requests, follow these simple steps:

1. Go to the Amazon EC2 console by clicking here.
2. In the navigation pane, find "Load Balancing" and click on "Target Groups."
3. Select "Create target group."
4. In the Basic configuration, leave the Target type as "instance."
5. Give your new target group a name in the "Target group name" field.
6. Keep the default settings for protocol (HTTP) and port (80).
7. Choose the VPC containing your instances and maintain the protocol version as HTTP1.
8. For Health checks, stick with the default settings and click "Next."
9. On the "Register targets" page, follow these optional steps. This step is not mandatory for creating the load balancer, but if you want to test your load balancer and ensure it routes traffic correctly, you should register this target.
   - Choose one or more instances from the "Available instances."
   - Keep the default port as 80 and select "Include as pending" below.
10. Click "Create target group" to finish.

Step 2: Choose a Load Balancer Type

Elastic Load Balancing provides various load balancers, and in this guide, we'll focus on creating an Application Load Balancer. Follow these steps to create an Application Load Balancer:

1. Visit the Amazon EC2 console by going to https://console.aws.amazon.com/ec2/.
2. On the navigation bar, select a Region for your load balancer. Make sure it's the same Region you used for your EC2 instances.
3. In the navigation pane under Load Balancing, click on Load Balancers.
4. Choose "Create Load Balancer."
5. Select "Create" for Application Load Balancer.

Step 3: Configure Your Load Balancer and Listener

To establish an Application Load Balancer, start by providing essential configuration details such as a name, scheme, and IP address type. Subsequently, input information regarding your network and configure one or more listeners. A listener acts as a process that monitors connection

requests, configured with a specified protocol and port for client connections to the load balancer. Refer to the Listener configuration for details on supported protocols and ports.
Follow these steps to configure your load balancer and listener:

1. **Load Balancer Name:** Enter a name for your load balancer, e.g., "my-alb."
2. **Scheme and IP Address Type:** Retain the default values.
3. **Network Mapping:** Choose the VPC used for your EC2 instances. Select at least two Availability Zones and one subnet per zone. For each Availability Zone used to launch EC2 instances, select the Zone and choose one public subnet for that Zone.
4. **Security Groups:** Choose the default security group for the selected VPC. Alternatively, select a different security group. Ensure the security group has rules permitting communication between the load balancer and registered targets on both the listener port and the health check port. Refer to Security group rules for more details.
5. **Listeners and Routing:** Keep the default protocol and port. Select your target group from the list. This configures a listener accepting HTTP traffic on port 80, forwarding it to the chosen target group by default. No HTTPS listener is created for this tutorial.
6. **Default Action:** Choose the target group created and registered in Step 1: Configure your target group.
7. **Optional: Add a Tag:** Include a tag to categorize your load balancer. Tag keys must be unique, allowing letters, spaces, numbers (in UTF-8), and special characters like + - = . _ : / @. Avoid leading or trailing spaces, and note that tag values are case-sensitive.
8. **Review Configuration:** Assess your settings and select "Create load balancer." The system applies a few default attributes during creation, editable post-creation. Refer to Load balancer attributes for additional information.

Step 4: Verify Your Load Balancer's Performance

After creating the load balancer, ensure it's directing traffic to your EC2 instances.
Once you've successfully created the load balancer, ensure that it effectively directs traffic to your EC2 instances.
To check your load balancer, follow these steps:

1. **Close Notification:**
   - Once you receive confirmation that your load balancer was created successfully, click "Close."
2. **Navigate to Target Groups:**
   - In the navigation pane, go to "Load Balancing" and select "Target Groups."
3. **Select Target Group:**
   - Pick the newly created target group.
4. **Verify Instance Readiness:**
   - Click on "Targets" and confirm that your instances are ready. If an instance's status is "initial," it may still be in the registration process or has not passed the minimum health checks to be considered healthy.

Once at least one instance shows a healthy status, you can proceed to test your load balancer.

5. **Access Load Balancers:**
   - In the navigation pane, under "Load Balancing," choose "Load Balancers."
6. **Choose Load Balancer:**
   - Select the recently created load balancer.
7. **Copy DNS Name:**
   - Under "Description," copy the DNS name of the load balancer (e.g., my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com).
8. **Paste in Browser:**
   - Paste the copied DNS name into the address field of an internet-connected web browser. If everything is functioning correctly, the browser should display the default page of your server.
9. **Optional - Add Listener Rules:**
   - To include additional listener rules, refer to "Add a rule." To enhance your setup, consider adding additional listener rules.

AWS Tutorials - 42 - Application Load Balancer | Layer 7 Load Balancer

Path-Based Routing

Path-based routing is a crucial concept in modern web architecture, allowing us to direct incoming traffic to different backend services based on the URL path. In this blog post, we'll delve into what path-based routing is, how it can be used effectively, and walk through a step-by-step guide to configuring it using AWS Application Load Balancer (ALB). By the end, you'll have a clear understanding of path-based routing and be ready to implement it in your own projects.

1. **What is Path-Based Routing?**
   - Path-based routing involves directing incoming requests to different backend services based on the URL path. For example, you can route requests to /app1/* to one service and requests to /app2/* to another service.
2. **How Can We Use Path-Based Routing?**
   - Path-based routing is incredibly useful in scenarios where you have multiple applications or microservices running on the same domain or server. It allows you to efficiently manage traffic by directing requests to the appropriate backend service based on the requested path.
3. **Use Cases of Path-Based Routing:**
   - **Microservices Architecture:** In a microservices architecture, different microservices often handle different functionalities. Path-based routing enables you to route requests to the corresponding microservice based

on the requested path.

- **API Gateway:** When building APIs, path-based routing can be used to route requests to different API endpoints based on the URL path, providing a clean and organized API structure.
- **Multi-Tenancy Applications:** In multi-tenancy applications, path-based routing can be used to route requests to different tenants based on the URL path, allowing for efficient resource allocation and isolation.

4. **How to Configure Path-Based Routing in AWS ALB:**

- Step 1: Create an Application Load Balancer (ALB) in the AWS Management Console.
- Step 2: Define target groups for different backend services.
- Step 3: Configure listener rules to route traffic based on the URL path.
- Step 4: Test and validate the path-based routing configuration.
- Step 5: Monitor and optimize your path-based routing setup for performance.

Conclusion:

Path-based routing is a powerful mechanism for managing and directing traffic in modern web applications. By understanding its concepts and implementing it using AWS ALB, you can effectively route incoming requests to different backend services based on the URL path. Whether you're building microservices, APIs, or multi-tenancy applications, path-based routing provides a flexible and efficient way to organize and manage your infrastructure.

AWS Tutorials - 43 - Path Base Routing in Application Load Balancer | Application Load Balancer