

20-Aug-2024

Internship Day - 24 Report:

What is DynamoDB?

- 1) **Amazon DynamoDB:** DynamoDB is a fully managed NoSQL database service provided by AWS that offers fast and predictable performance with seamless scalability.
- 2) **Key-Value and Document Store:** It supports both key-value and document data structures, making it versatile for various applications.

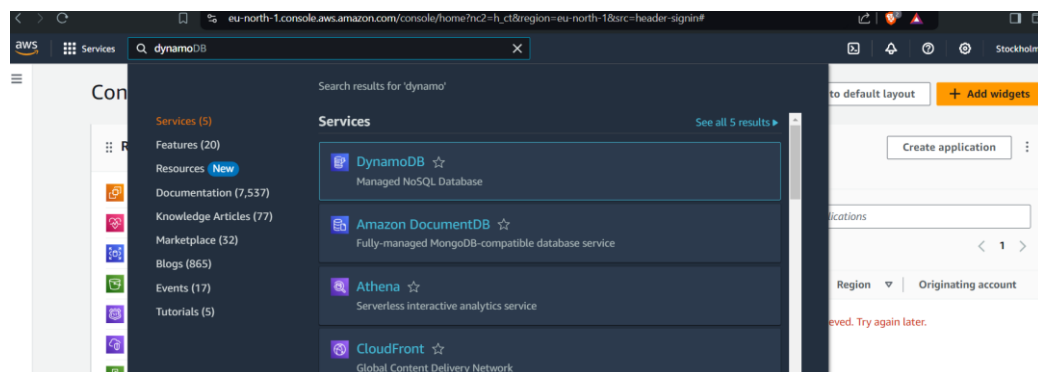
Why Use DynamoDB?

- **Scalability:** DynamoDB automatically scales up or down to adjust for capacity and maintain performance.
- **Performance:** It provides low-latency responses, making it suitable for applications that require real-time data access.
- **Fully Managed:** As a managed service, it eliminates the administrative burden of operating and scaling a distributed database.
- **Flexible Data Model:** The schema-less design allows for storing diverse data types and structures.
- **Integrated with AWS Services:** It easily integrates with other AWS services, enabling powerful application architectures.

Getting Started with Amazon DynamoDB: A Step-by-Step Guide for Database Management and Integration with PHP

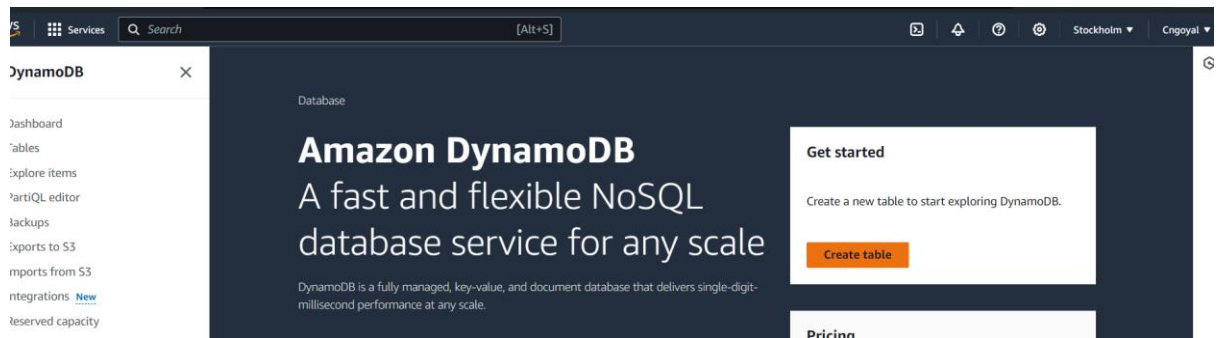
Step 1: Access DynamoDB

- **Search for DynamoDB:** In the AWS Management Console, search for "DynamoDB" and select it.

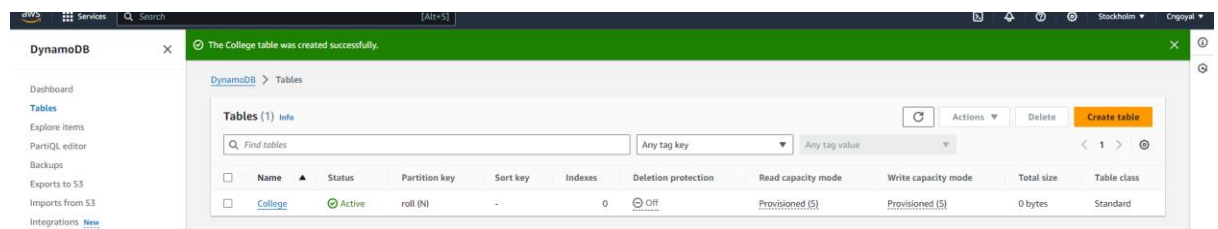


Step 2: Create a Table

- **Create Table:** Click on "Create table".



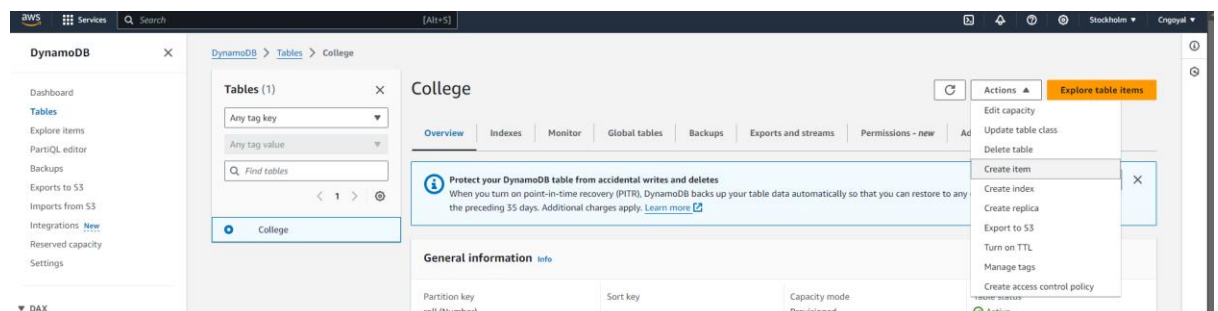
- **Enter Table Name:** Provide a name for your table (e.g., "College").



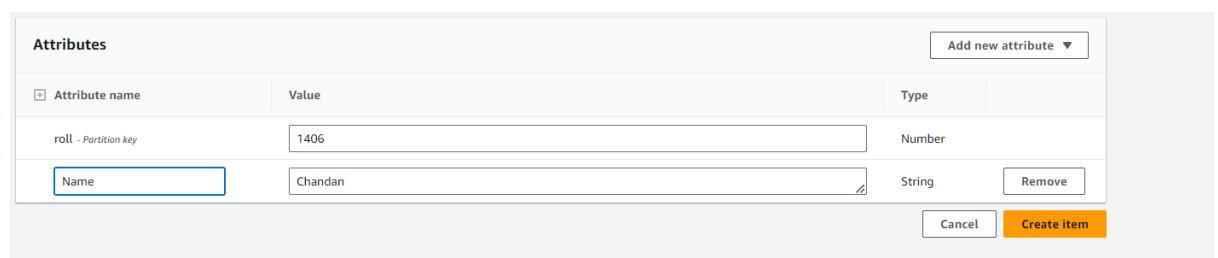
- **Define Partition Key:** Specify the partition key for the table and click "Create table".

Step 3: Add Items to the Table

- **Select Table:** Click on the table you created (e.g., "College").
- **Create Item:** In the "Actions" dropdown, click on "Create item".



- **Add Attributes:** Add new attributes as needed by entering the attribute name and value.

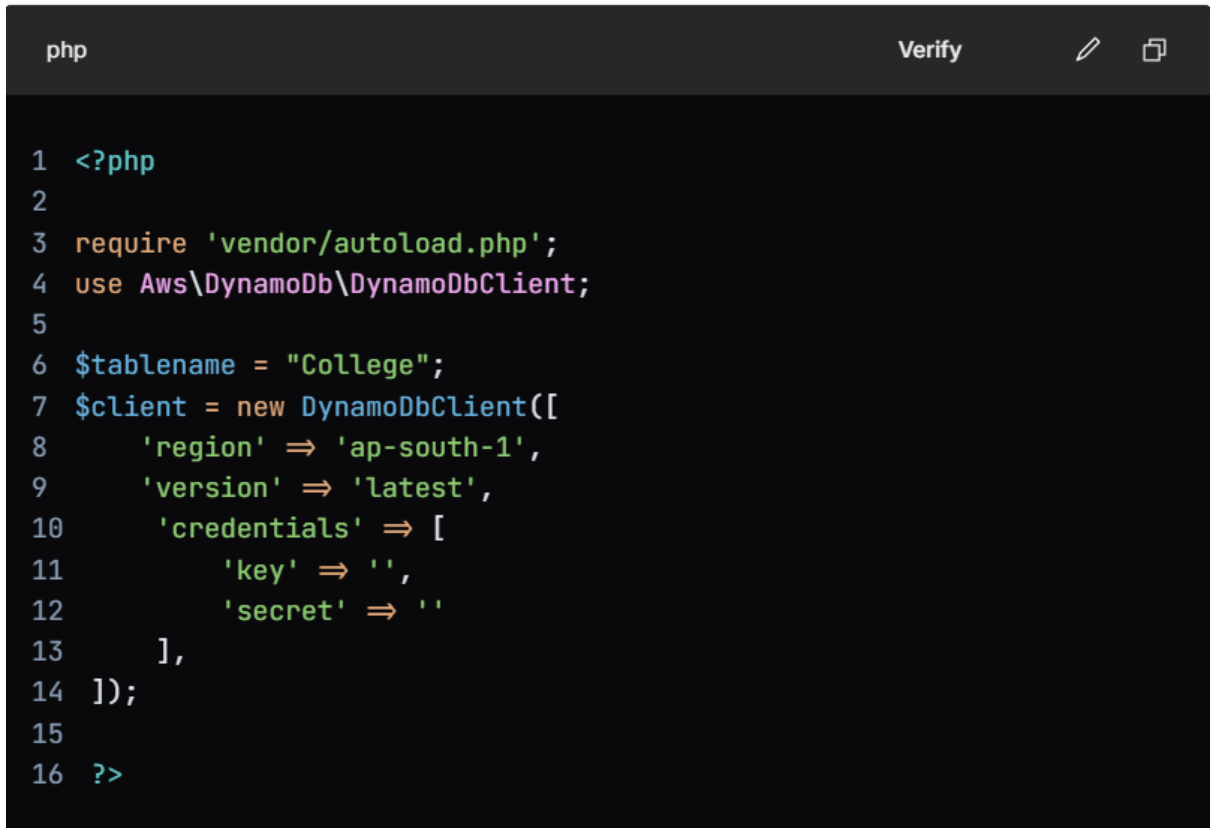


Step 4: Explore Items

- **View Entries:** Navigate to "Items" in the left sidebar to see the entry data for your table.

Step 5: Edit PHP Code for Connection

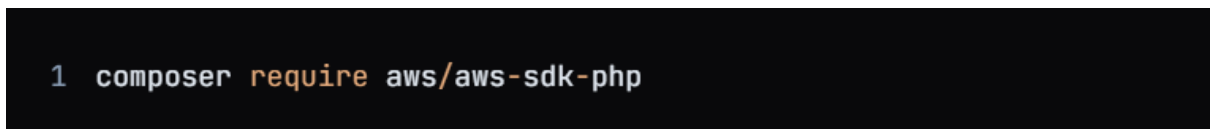
- **Open PHP File:** In your code editor (e.g., VS Code), open the PHP file where you want to connect to DynamoDB.
- **Edit Code:** Update the PHP code to include the necessary AWS SDK for PHP:

A screenshot of a code editor window with a dark theme. The title bar shows 'php' on the left and 'Verify' with edit and copy icons on the right. The code is as follows:

```
1 <?php
2
3 require 'vendor/autoload.php';
4 use Aws\DynamoDb\DynamoDbClient;
5
6 $tablename = "College";
7 $client = new DynamoDbClient([
8     'region' => 'ap-south-1',
9     'version' => 'latest',
10    'credentials' => [
11        'key' => '',
12        'secret' => ''
13    ],
14 ]);
15
16 ?>
```

Step 6: Install AWS SDK for PHP

- **Open Terminal:** In the VS Code terminal, run the following command to install the AWS SDK for PHP:

A screenshot of a terminal window with a dark theme. It shows a single command being entered:

```
1 composer require aws/aws-sdk-php
```

21-Aug-2024

Internship Day - 25 Report:

Comprehensive Guide to Load Balancers in AWS: Purpose, Types, and Benefits

Understanding Load Balancers in AWS

A load balancer in AWS is a service that distributes incoming application traffic across multiple targets, such as EC2 instances, to ensure high availability and reliability.

Why Use a Load Balancer?

- 1) **High Availability:** Ensures that applications remain accessible even in the event of server failures by rerouting traffic to healthy instances.
- 2) **Scalability:** Automatically adjusts to varying traffic loads, allowing applications to handle increased demand without performance degradation.
- 3) **Fault Tolerance:** Monitors the health of registered targets and only routes traffic to those that are healthy, minimizing downtime.
- 4) **Security:** Provides an additional layer of security by hiding the backend instances from direct access and can integrate with AWS WAF for web application security.

Types of Load Balancers in AWS

1. Application Load Balancer (ALB):

- Operates at Layer 7 (Application Layer) and is ideal for HTTP/HTTPS traffic.
- Supports advanced routing features based on content, making it suitable for microservices architectures.

2. Network Load Balancer (NLB):

- Operates at Layer 4 (Transport Layer) and is designed for high-throughput and low-latency applications.
- Best for TCP and UDP traffic, handling millions of requests per second.

3. Classic Load Balancer (CLB):

- The original load balancer that operates at both Layer 4 and Layer 7.
- Suitable for applications built within the EC2-Classic network but lacks some advanced features of ALB and NLB.

4. Gateway Load Balancer (GWLb):

- Combines a transparent network gateway with a load balancer.
- Ideal for deploying third-party virtual appliances, such as firewalls, in a scalable manner.

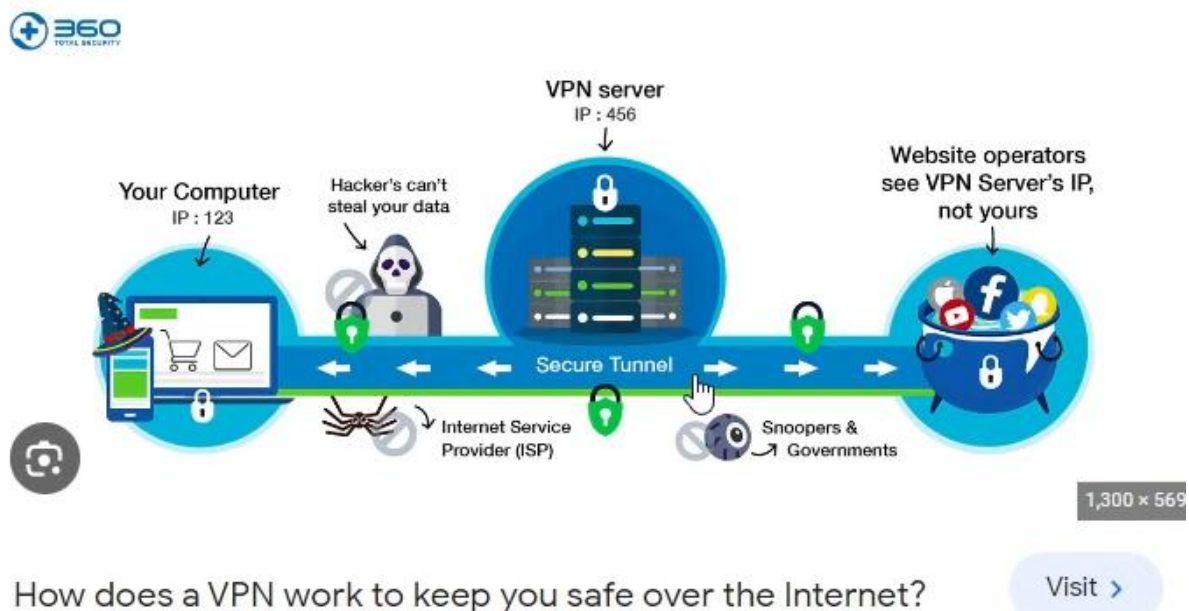
22-Aug-2024

Internship Day - 26 Report:

Site-to-Site VPN in AWS: Steps to Set Up with Virtual Private Gateway

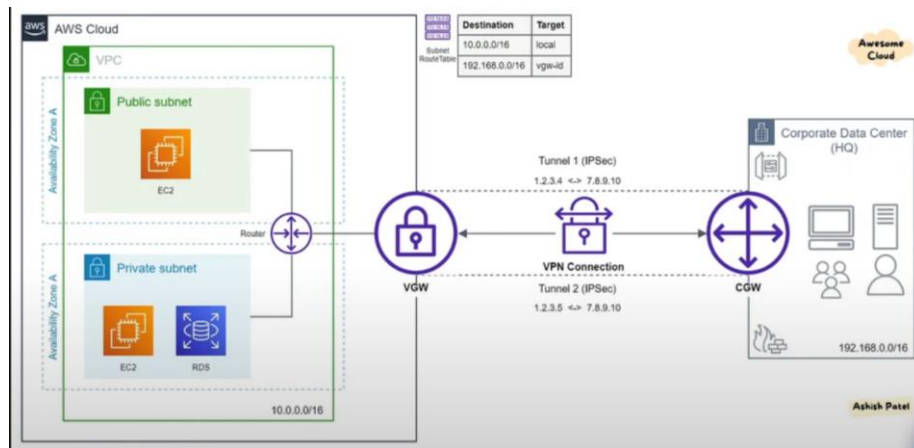
Overview

A Site-to-Site VPN in AWS allows you to securely connect your on-premises network or data center (Customer Gateway) to your AWS Virtual Private Cloud (VPC) through a Virtual Private Gateway. This connection enables secure communication between your local network and AWS resources.



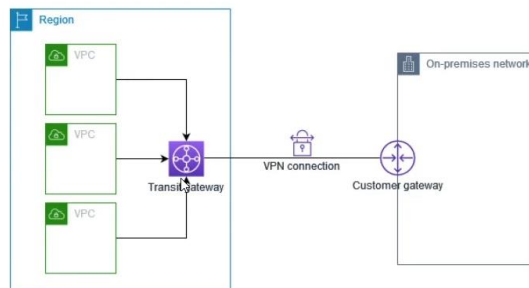
Components

- Customer Gateway (CGW): Represents your on-premises VPN device (VPN server).
- Virtual Private Gateway (VGW): The VPN concentrator on the AWS side that connects your VPC to the VPN.



- **Transit Gateway:** A service that enables you to connect multiple VPCs and on-premises networks through a central hub.

The following diagram shows a VPN connection between multiple VPCs and your on-premises network using a transit gateway. The transit gateway has three VPC attachments and a VPN attachment.



Your Site-to-Site VPN connection on a transit gateway can support either IPv4 traffic or IPv6 traffic inside the VPN tunnels. For more information, see [IPv4 and IPv6 traffic](#).

Steps to Set Up Site-to-Site VPN

1. Create a VPC:

Log in to the AWS Management Console.

Navigate to the VPC dashboard.

Click on "Create VPC" and configure the required settings (CIDR block, name, etc.).

VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.

☐ VPC only
 ☒ VPC and more

Preview

VPC Show details
Your AWS virtual network

2. Launch an EC2 Instance (Optional):

- If needed, launch an EC2 instance within the created VPC for further testing or application deployment.

3. Create a Customer Gateway:

- Go to the VPC dashboard and select "Customer Gateways."
- Click on "Create Customer Gateway."
- Enter the public IP address of your on-premises VPN device (CGW).
- Specify the routing type (static or dynamic) based on your network setup.

4. Create a Virtual Private Gateway:

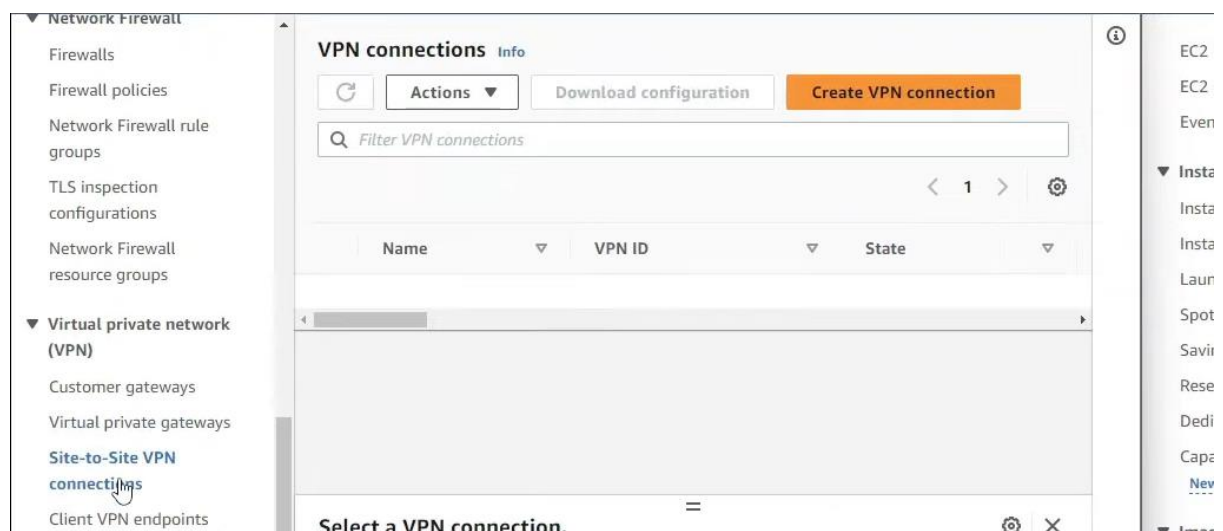
- In the VPC dashboard, select "Virtual Private Gateways."
- Click on "Create Virtual Private Gateway."
- Provide a name and create the gateway.

5. Attach the Virtual Private Gateway to the VPC:

- After creating the VGW, select it and click on "Attach to VPC."
- Choose the VPC you created earlier and attach the VGW.

6. Create a Site-to-Site VPN Connection:

- In the VPC dashboard, navigate to "VPN Connections."
- Click on "Create VPN Connection."
- Select the Virtual Private Gateway you created and the Customer Gateway you set up.
- Configure the tunnel options (encryption, routing, etc.) and create the connection.



7. Update Routing Tables:

- Go to the VPC dashboard and select "Route Tables."
- Choose the route table associated with your VPC.
- Add a route to direct traffic destined for your on-premises network through the Virtual Private Gateway.

8. Configure the On-Premises VPN Device:

- Download the VPN configuration file provided by AWS.
- Use this configuration to set up your on-premises VPN device (CGW) to establish the VPN connection.

9. Testing the Connection:

- Once configured, test the VPN connection by pinging resources in the VPC from your on-premises network.
- Monitor the connection status in the AWS Management Console.

10. Using Transit Gateway (Optional):

- If you have multiple VPCs, consider creating a Transit Gateway.
- Attach your VPCs and Customer Gateways to the Transit Gateway for centralized management and routing.

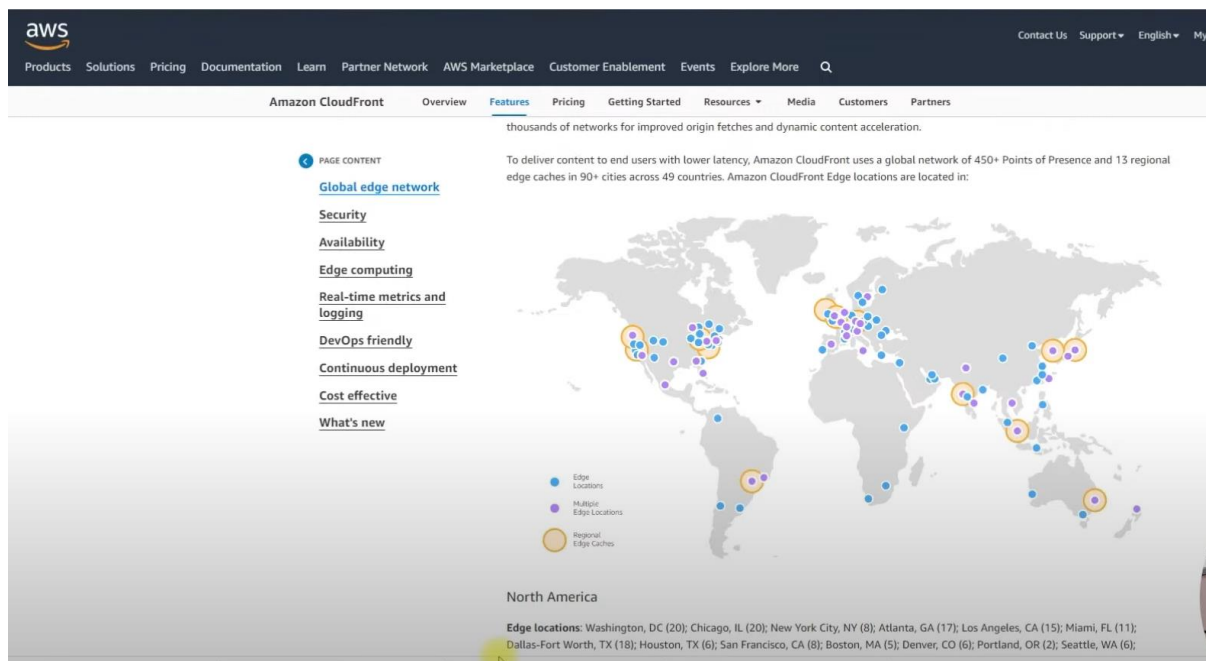
23-Aug-2024

Internship Day - 27 Report:

AWS CloudFront: Overview and Implementation Steps

Introduction to AWS CloudFront

AWS CloudFront is a content delivery network (CDN) service that provides a way to deliver data, videos, applications, and APIs to users globally with low latency and high transfer speeds. It achieves this by caching content at edge locations close to the users, ensuring faster access and reduced load on the origin server.



How CloudFront Works?

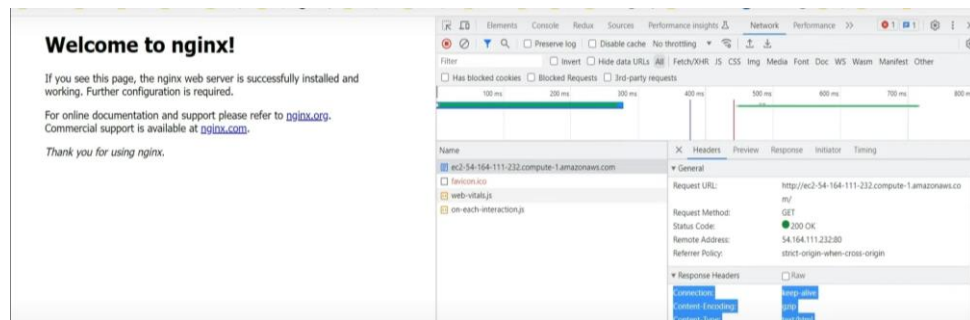
1. **Edge Locations:** CloudFront uses a network of edge locations around the world. When a user requests content, the request is routed to the nearest edge location.
2. **Data Retrieval:** If the content is cached at the edge location, it is served directly to the user. If not, CloudFront fetches the content from the origin server (e.g., an EC2 instance) and caches it for future requests.
3. **Latency Reduction:** By serving content from the nearest edge location, CloudFront minimizes latency, improving the performance of websites, especially for e-commerce, streaming, and blogs.

4. **SEO Benefits:** Faster response times can enhance user experience, leading to better SEO rankings as search engines favor quick-loading websites.

Practical Steps to Set Up CloudFront with an EC2 Instance

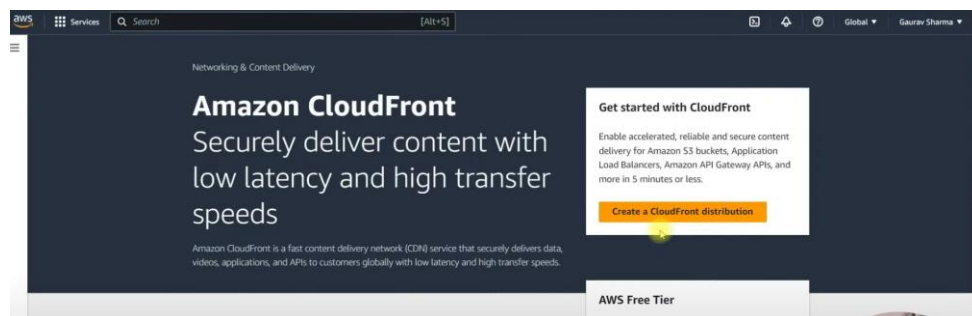
Step 1: Create an EC2 Instance

- Launch an EC2 instance that will serve as your origin server.

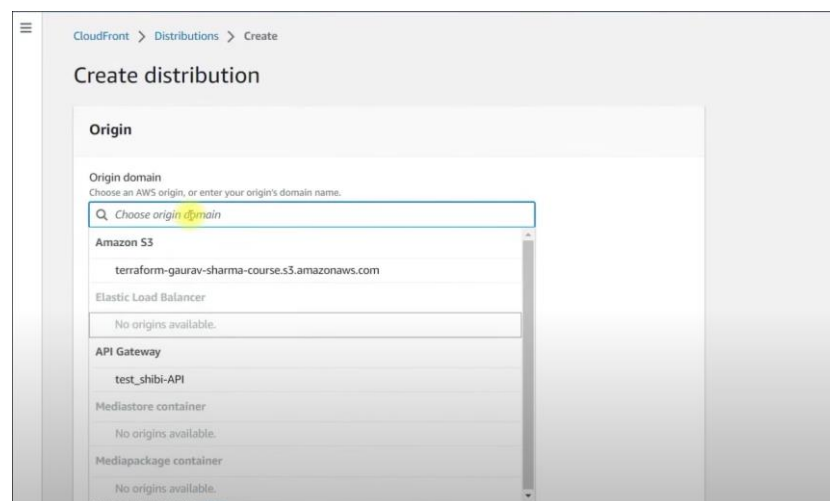


Step 2: Create a CloudFront Distribution

- Navigate to the CloudFront console and create a new distribution.

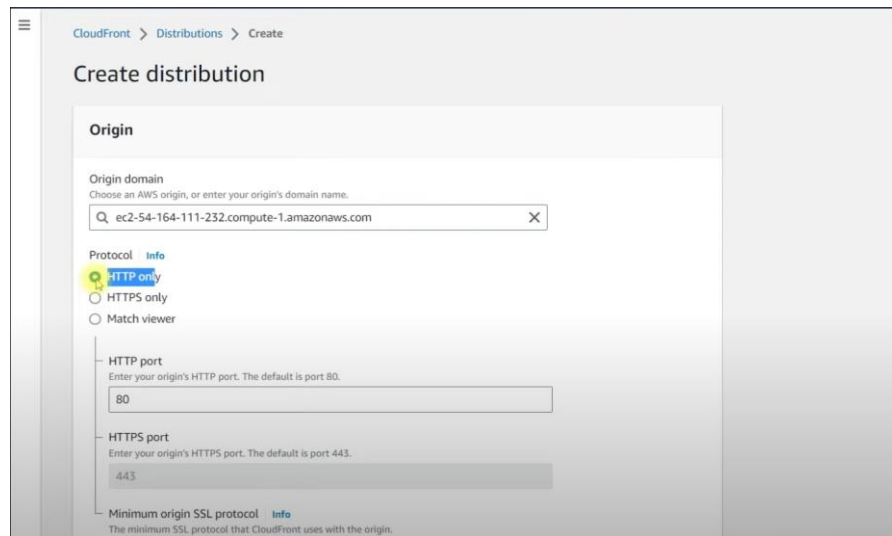


- Enter the DNS URL of your EC2 instance as the origin.



Step 3: Select Protocol

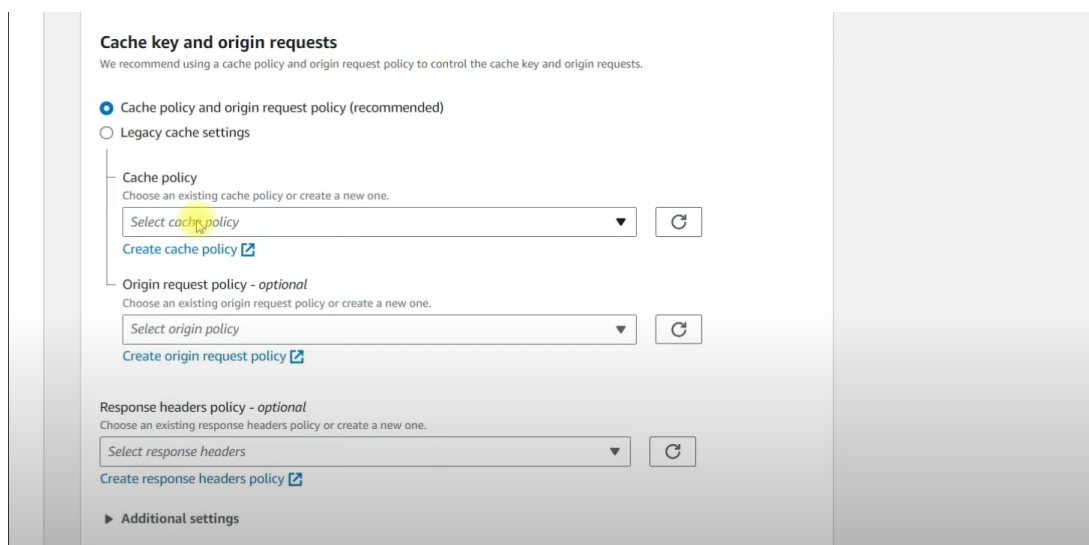
- Choose the appropriate protocol for your distribution (HTTP/HTTPS).



The screenshot shows the 'Create distribution' page in the AWS CloudFront console, specifically the 'Origin' tab. The 'Origin domain' field contains 'ec2-54-164-111-232.compute-1.amazonaws.com'. Under the 'Protocol' section, 'HTTP only' is selected with a radio button. Below this, the 'HTTP port' is set to '80' and the 'HTTPS port' is set to '443'. The 'Minimum origin SSL protocol' is also visible at the bottom.

Step 4: Create a Cache Policy

- Define a cache policy by entering a name and setting the TTL (Time to Live) for how long the content should be cached at the edge location.



The screenshot shows the 'Cache key and origin requests' section of the AWS CloudFront console. It features three radio buttons: 'Cache policy and origin request policy (recommended)' (selected), 'Legacy cache settings', and 'Legacy cache settings'. Below the first option, there are three dropdown menus for selecting a 'Cache policy', 'Origin request policy - optional', and 'Response headers policy - optional'. Each dropdown has a 'Select' button and a 'Create' link. The 'Additional settings' section is partially visible at the bottom.

Step 5: Create the Distribution

- Click "Create Distribution" and wait for it to be deployed.

Web Application Firewall (WAF)

☐ **Enable security protections**
 Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.

☒ **Do not enable security protections**
 Select this option if your application does not need security protections from AWS WAF.

Settings

Price class: [Info](#)

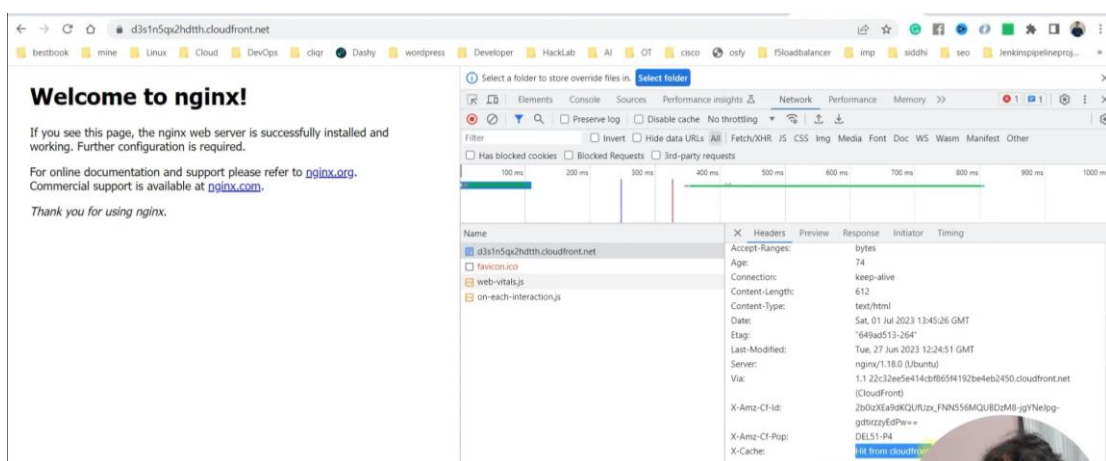
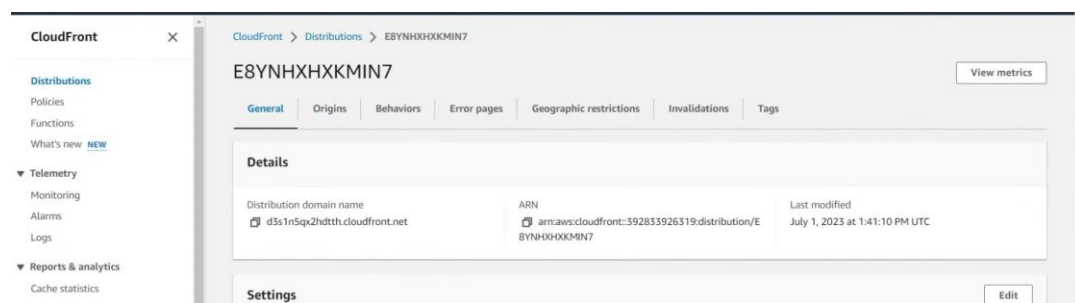
On

Description - optional

Cancel **Create distribution**

Step 6: Monitor Last Modified Status

- The first request will fetch data from the origin server to the edge location, allowing subsequent requests to be served from CloudFront.



Problem: Direct Access to EC2 Instance**Solution: Restrict Access to CloudFront Only****Find CloudFront IP Addresses:**

- Use the AWS-managed prefix list to find the IP addresses associated with CloudFront.

Modify EC2 Security Group:

- Go to the security group settings of your EC2 instance.
- Remove the existing HTTP (port 80) rule.
- Create a new rule allowing HTTP access only from the CloudFront IP prefix list.

Invalidate CloudFront Cache:

- To ensure users receive the latest content, perform an invalidation in CloudFront by specifying the paths to delete or using /* to clear all cached content.

Reload CloudFront DNS:

- After making these changes, reload your CloudFront distribution to ensure it serves the latest content.

Important Notes

- Always ensure that any updates made to your EC2 instance or load balancer are reflected in CloudFront by performing cache invalidation.
- Regularly check and update the security group rules to maintain access control and security.

By following these steps, you can effectively utilize AWS CloudFront to enhance the performance and security of your web applications.