Conception Phase: Cloud Architecture for Hosting a Simple Webpage on AWS

Introduction:

This project focuses on hosting a webpage on AWS using a variety of cloud tools. These tools help automate the process of storing and delivering the webpage to users around the world, ensuring both high performance and availability.

Infrastructure Setup and Management:

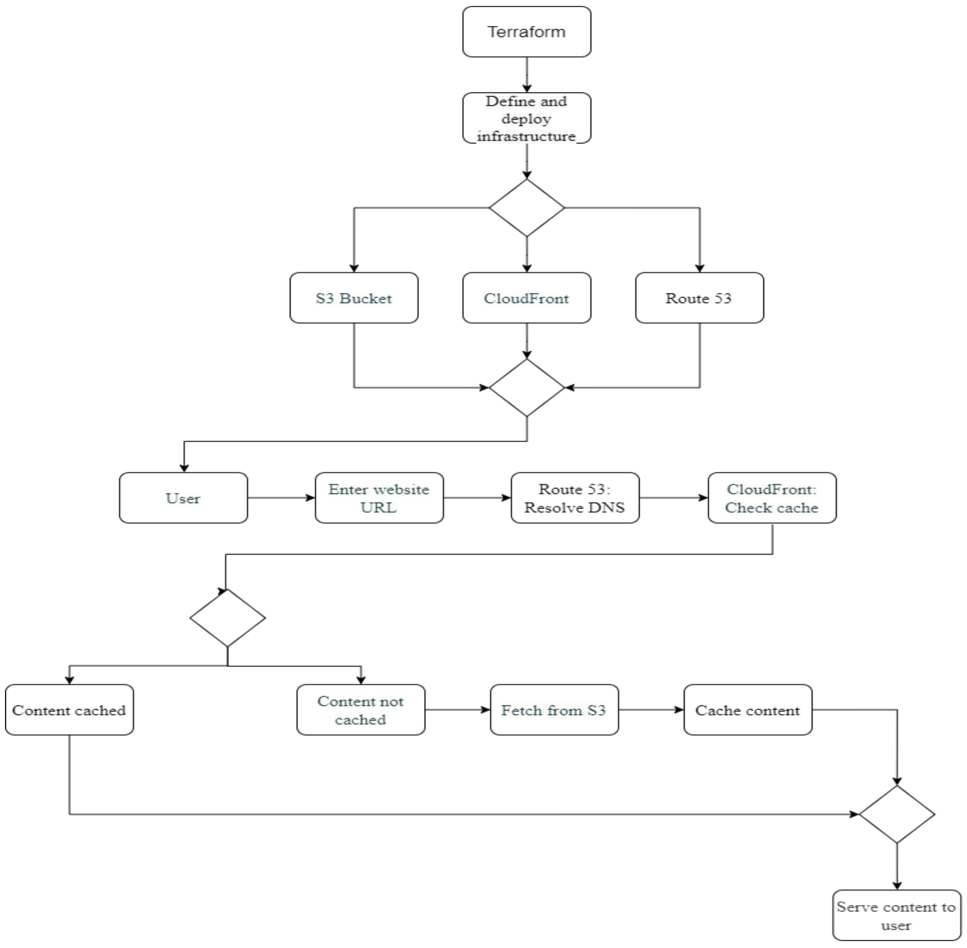
1. **Terraform for Infrastructure Management:** Terraform is used to define and manage the AWS infrastructure as code (HashiCorp Cloud Platform, n.d.). It enables a scalable setup of resources like S3, CloudFront, and Route 53, simplifying the deployment process (HashiCorp Cloud Platform, n.d.).
2. **Infrastructure Deployment:** The infrastructure setup includes writing Terraform scripts that define AWS resources, such as:
   - **S3 Bucket:** Stores static website files, in my case an HTML file (AWS, 2024).
   - **CloudFront:** Caches website content closer to users for faster load times (AWS, n.d.a).
   - **Route 53:** Manages DNS to route user requests to the correct location (AWS, n.d.b).

   The infrastructure is then deployed using these scripts, ensuring consistency and ease of management.

Workflow: How the Hosting System Operates:

1. **A user** types the website URL in their browser, initiating a request for the webpage.
2. **Route 53** translates the domain name to an IP address and directs the request to **CloudFront** for content delivery (AWS, n.d.b).
3. **CloudFront** checks if it already has the requested content cached in one of its edge locations (AWS, n.d.a).
   - **If cached:** CloudFront quickly serves the stored content from a nearby edge location (AWS, n.d.a).
   - **If not cached:** CloudFront retrieves the content from the **S3 Bucket**, stores it for future requests, and then delivers it to the user (AWS, n.d.a).
4. The **S3 Bucket** stores the website’s static files, in this case an HTML file, ensuring it is available for CloudFront to fetch if needed (AWS, 2024).
5. After the content is retrieved or cached, **CloudFront** delivers the webpage to the user, ensuring the website loads quickly regardless of their location (AWS, n.d.a).

The following diagram illustrates the architecture:



Conclusion:

This project successfully demonstrates the end-to-end process of hosting a webpage on AWS. By using tools like Terraform and AWS services, such as S3, CloudFront, Route 53. The setup ensures a fast, reliable, and scalable hosting environment, optimizing user experience through efficient content delivery (AWS, 2024).

Reference:

AWS. (n.d.a). *What is Amazon CloudFront?* AWS. Retrieved from https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html.

AWS. (n.d.b). *What is Amazon Route 53?* AWS. Retrieved from https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html.

AWS. (2024). *What is Amazon S3?* AWS. Retrieved from https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html.

HashiCorp Cloud Platform. (n.d.). *What is Terraform?* HashiCorp. Retrieved from https://developer.hashicorp.com/terraform/intro.