A Website on Cloud Platform

Ratan Karan Srivastava[1], Vipul Rai[2], Mohini Singh[3], Vishal Kumar Sharma[4]

[1] B.Tech Student, Department of Computer Science and Engineering, Institute of Technology And Management Dr. A.P.J. Abdul Kalam Technical University , Uttar Pradesh, India

[2] B.Tech Student, Department of Computer Science and Engineering, Institute of Technology And Management Dr. A.P.J. Abdul Kalam Technical University , Uttar Pradesh, India

[3] B.Tech Student, Department of Computer Science and Engineering, Institute of Technology And Management Dr. A.P.J. Abdul Kalam Technical University , Uttar Pradesh, India

[4] B.Tech Student, Department of Computer Science and Engineering, Institute of Technology And Management Dr. A.P.J. Abdul Kalam Technical University , Uttar Pradesh, India

ABSTRACT

This paper covers comprehensive architectural guidance for developing, deploying, and managing static websites on Amazon Web Services (AWS) while keeping operational simplicity and business requirements in mind. We also recommend an approach that provides 1) insignificant cost of operation, 2) little or no management required, and 3) a highly scalable, resilient, and reliable website.
This paper first reviews how static websites are hosted in traditional hosting environments. Then, we explore a simpler and more cost-efficient approach using Amazon Simple Storage Service (Amazon S3). Finally, we show you how you can enhance the AWS architecture by encrypting data in transit and to layer on functionality and improve quality of service by using Amazon CloudFront.

Keyword : - *Cloud Computing , Scalability , Availability , and Cloud Architecture*

1. Introduction

As enterprises become more digital operations, their websites span a wide spectrum, from mission-critical e-commerce sites to departmental apps, and from business-to-business (B2B) portals to marketing sites. Factors such as business value[1], mission criticality[2], service level agreements (SLAs)[3], quality of service[4], and information security[5] drive the choice of architecture and technology stack. The simplest form of website architecture is the static website, where users are served static content (HTML, images, video, JavaScript, style sheets, and so on). Some examples include brand microsites, marketing websites, and intranet information pages. Static websites are straightforward in one sense, but they can still have demanding requirements in terms of scalability, availability, and service-level guarantees. For example, a marketing site for a consumer brand may need to be prepared for an unpredictable onslaught of visitors when a new product is launched.

1.1 Static Website

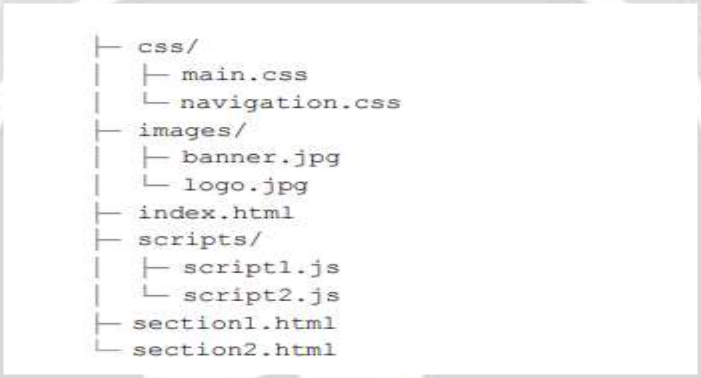
A static website delivers content in the same format in which it is stored. No server-side code execution is required. For example, if a static website consists of HTML documents displaying images, it delivers the HTML and images as-is to the browser, without altering the contents of the files. Static websites can be delivered to web browsers on desktops, tablets, or mobile devices. They usually consist of a mix of HTML documents, images, videos, CSS style sheets, and JavaScript files. Static doesn't have to mean boring—static sites can provide client-side interactivity as well. Using HTML5 and client-side JavaScript technologies such as jQuery, AngularJS, React, and Backbone, you can deliver rich user experiences that are engaging and interactive.

1.2 Dynamic Website

Dynamic websites can display dynamic or personalized content. They usually interact with data sources and web services, and require code development expertise to create and maintain. For example, a sports news site can display information based on the visitor's preferences, and use server-side code to display updated sport scores. Other examples of dynamic sites are e- commerce shopping sites, news portals, social networking sites, finance sites, and most other websites that display ever-changing information.

2. Core Architecture

In a traditional (non-AWS) architecture, web servers serve up static content. Typically, content is managed using a content management system (CMS), and multiple static sites are hosted on the same infrastructure. The content is stored on local disks, or on a file share on network-accessible storage. The following example shows a sample file system structure.



```
├─ css/
│   ├─ main.css
│   └─ navigation.css
├─ images/
│   ├─ banner.jpg
│   └─ logo.jpg
├─ index.html
├─ scripts/
│   ├─ script1.js
│   └─ script2.js
├─ section1.html
└─ section2.html
```

Chart -1: File System Structure

A network firewall protects against unauthorized access. It's common to deploy multiple web servers behind a load balancer for high availability (HA) and scalability. Since pages are static, the web servers don't need to maintain any state or session information and the load balancer doesn't need to implement session affinity ("sticky sessions"). The following diagram shows a traditional (non-AWS) hosting environment:

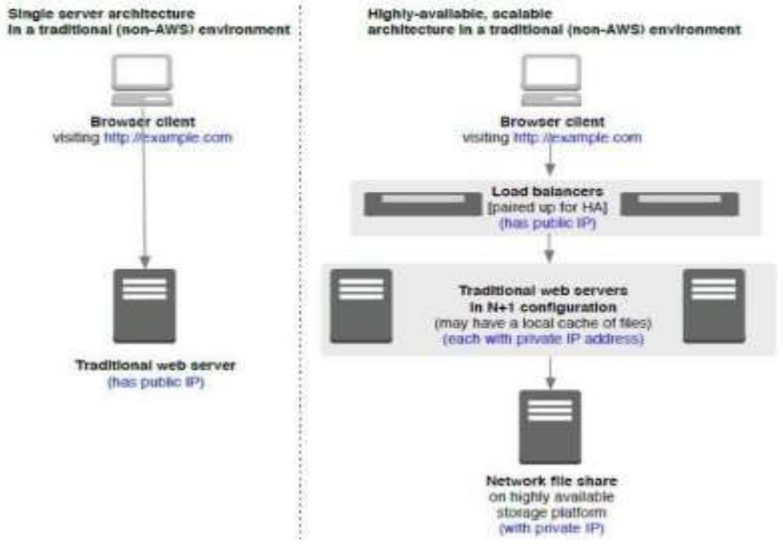

Single server architecture in a traditional (non-AWS) environment
Browser client visiting http://example.com
Traditional web server (has public IP)
Highly-available, scalable architecture in a traditional (non-AWS) environment
Browser client visiting http://example.com
Load balancers (paired up for HA) (has public IP)
Traditional web servers in N+1 configuration (may have a local cache of files) (each with private IP address)
Network file share on highly available storage platform (with private IP)

Fig -1: Basic architecture of a traditional hosting environment

3. Moving to an AWS Architecture

Using this AWS architecture, you gain the security, scalability, cost, and agility benefits of running in AWS. This architecture benefits from AWS world-class infrastructure and security operations. By using Auto Scaling, the website is ready for traffic spikes, so you are prepared for product launches and viral websites. With AWS, you only pay for what you use, and there's no need to over-provision for peak capacity. In addition, you gain increased agility because AWS services are available on demand. (Compare this to the traditional process in which provisioning servers, storage, or networking can take weeks.) You don't have to manage infrastructure, so this frees up time and resources to create business differentiating value. AWS challenges traditional IT assumptions and enables new "cloud-native" architectures. You can architect a modern static website without needing a single web server.

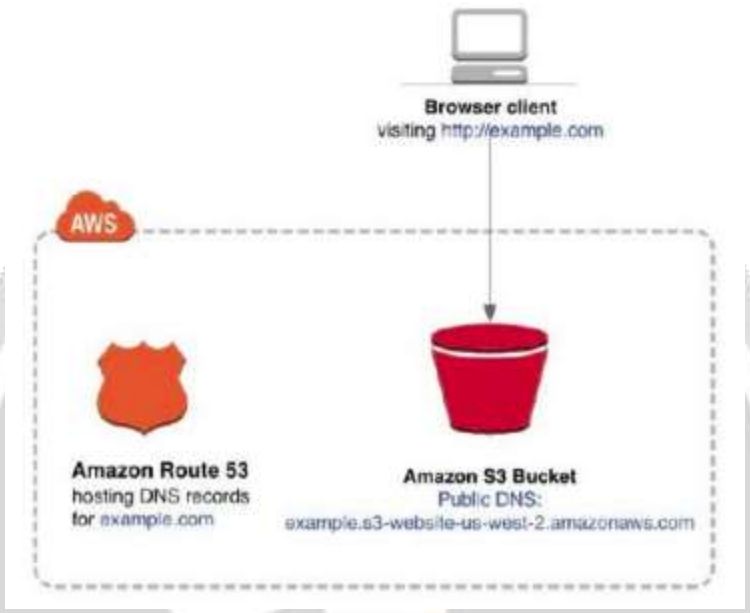


Fig -2: AWS S3 Website hosting

3.1 Scalability and Availability

Amazon S3 is inherently scalable. For popular websites, Amazon S3 scales seamlessly to serve thousands of HTTP or HTTPS requests per second without any changes to the architecture. In addition, by hosting with Amazon S3, the website is inherently highly available. Amazon S3 is designed for 99.999999999% durability, and carries a service level agreement (SLA) of 99.9% availability. Amazon S3 gives you access to the same highly scalable, reliable, fast, and inexpensive infrastructure that Amazon uses to run its own global network of websites. As soon as you upload files to Amazon S3, Amazon S3 automatically replicates your content across multiple data centers. Even if an entire AWS data center were to be impaired, your static website would still be running and available to your end users.

3.2 Encrypt Data in Transit

We recommend you use HTTPS to serve static websites securely. HTTPS is the secure version of the HTTP protocol that browsers use when communicating with websites. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS). TLS protocols are cryptographic protocols designed to provide privacy and data integrity between two or more communicating computer applications. HTTPS protects against man-in-the-middle (MITM) attacks. MITM attacks intercept and maliciously modify traffic.

4. CONCLUSIONS

This paper began with a look at traditional (non-AWS) architectures for static websites. We then showed you an AWS Cloud-native architecture based on Amazon S3, Amazon CloudFront, and Amazon Route 53. The AWS architecture is highly available and scalable, secure, and provides for a responsive user experience at very low cost. By enabling and analyzing the available logs, you can understand your visitors and how well the website is performing. Fewer moving parts means less maintenance is required. In addition, the architecture costs only a few dollars a month to run.

5. REFERENCES

[1]. Jim Tran, AWS Principal Enterprise Solutions Architect
[2]. Bhushan Nene, Senior Manager, AWS Solutions Architecture
[3]. Jonathan Pan, Senior Product Marketing Manager, AWS
[4]. Brent Nash, Senior Software Development Engineer, AWS