Cloud Application Development

Submitted By:-

Yuvraj Singh Pundir

B-4 CCVT Non-hons.

500087109

Submitted to: -

Mr. Saurabh Shanu

Assistant Professor

WEEK-2 Assignment

The architecture style of my application:

The architecture of a file sharing and hosting website typically works on the basis of the client-server or distributed architecture model (when we use multiple servers to serve the client). In my case, as I will be using a single server, the application will hence follow the client-server architecture.

In this architecture, clients make requests to the server, which processes the request and returns a response.

Why SISD?

Client-Server Architecture follows the **SISD (Single Instruction Single Data)** architecture because each client operates independently, with its own set of instructions and data. The server also operates independently, processing requests from multiple clients in a sequential manner. At any given time, the server is executing a single instruction on a single data item that belongs to a single client.

Which platform am I going to follow? What services am I going to use? What will be the basic architecture of my application?

I will be using AWS as my platform. It has all the required services I will need and it is also the paltform I am more familiar with. Following are the services I will use:

1. Amazon S3: This is a highly scalable and durable object storage service that can store and serve large amounts of unstructured data, such as files, images, and videos. This is the backbone service of the application. It can also be configured to backup user files using quarantine S3 buckets
2. Amazon CloudFront: This is a content delivery network (CDN) service that can be used to distribute files to users around the world with low latency and high data transfer speeds.

3. Amazon EC2: This is a web service that provides scalable computing capacity in the cloud. EC2 can be used to host the web application and database required to serve files to users.
4. Amazon RDS: This is a managed relational database service that can be used to store metadata about the files and the users who access them.
5. AWS Lambda: This provides executable triggers in response to client requests as well as to support the backend framework and perform actions on S3 buckets and databases.

I will add or remove the services as per the requirement during development.

Basic outline for the workflow of the application:

1. User registration: Users create an account on the website by providing basic information such as name, email, and password. This involves maintaining a relational database for user details. (RDS)

2. File upload: Users upload files to the cloud-based server. This can be done through the website's user interface. Uploaded files will be stored in S3 bucket.

3. File storage: The uploaded files are stored on the cloud-based server, where they are encrypted for security and optimized for efficient storage.

4. File sharing: Users can share files with others by generating a link or inviting specific users to collaborate on a file. The receivers can access the shared files through the website. This can be achieved by simply triggering a lambda function.

5. File version control: It will track changes made to files and allows users to revert to previous versions of the file if necessary. S3 bucket versioning can be used to implement this feature.

6. Access control: Allow administrators to set access controls for each file (IAM), determining who can view, edit, or download the file

7. Data backup and recovery: It will back up data regularly to ensure that files are not lost in case of a disaster or system failure. (Using backup quarantine S3 buckets)

8. Reporting and analytics: The system provides reporting and analytics capabilities, allowing administrators to track usage, performance, and other key metrics.

9. Deployment: The application will be deployed on EC2 instance, which provides the necessary computing resources and ensures high availability and scalability.

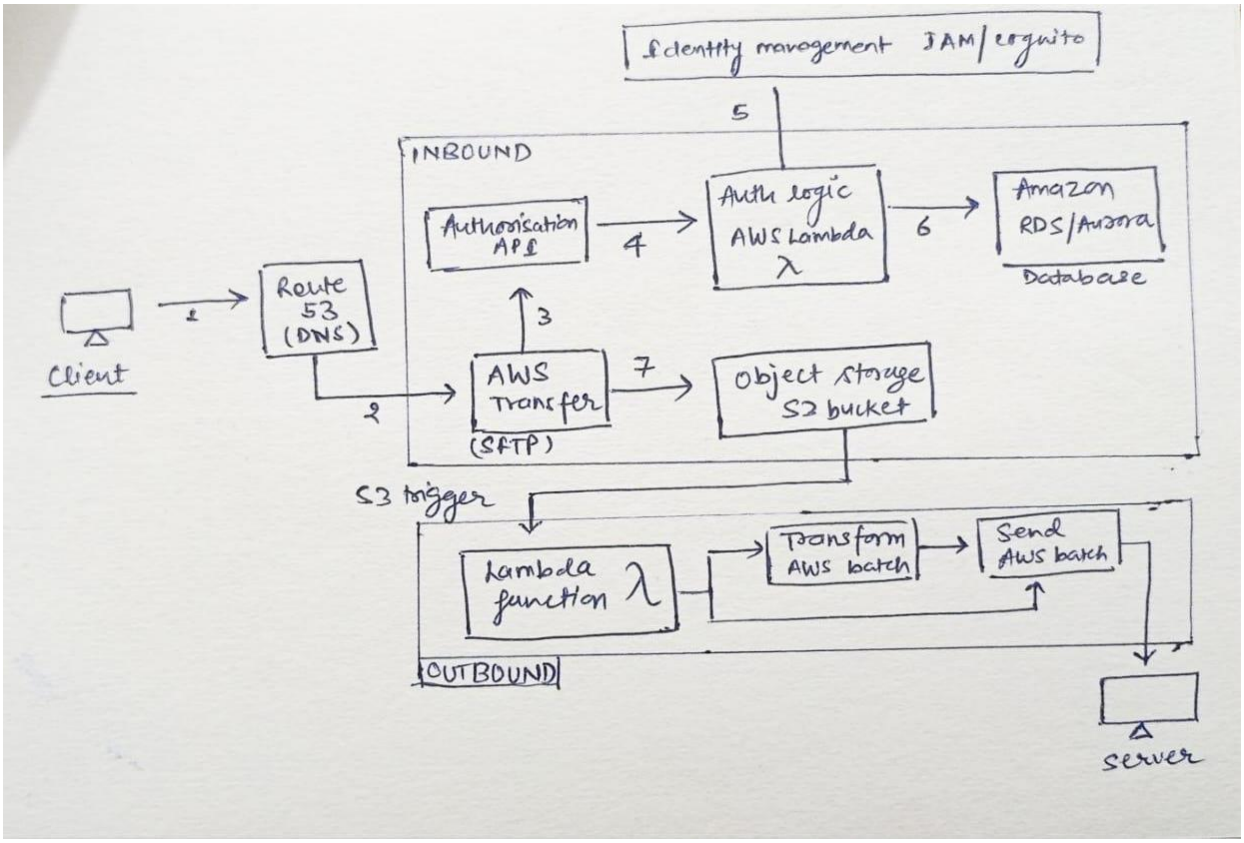

Figure : AWS architecture flowchart