AWS hosting for Workspace Application

Contents

1. What is Amazon Web Services

AWS or amazon web service is a secure cloud computing platform that provides computing power, database, networking, content storage and much more. The platform also works with a pay-as-you-go pricing model which means you only pay for how much of the service is offered by aws you use.

Some of the other advantages of AWS are security. AWS provides a secure and durable platform that offers end-to-end privacy and security. You can use the same ways of managing infrastructure that Amazon has developed over many years. It allows users to choose the OS, language, databases and other services. AWS is also easy to use, users can host applications quickly and securely. AWS can also be scalable, depending upon the use applications can be scaled-up or down.

There are more than 100 services that AWS is providing. Storage buckets, compute instances and messaging queue. Compute engine is what we are going to use to host our application.

1.1 AWS Compute Services

One of the original AWS products was elastic compute cloud. It allows you to create virtual computer in the cloud, choose your operating system, memory and computing power. We can use an instance as a server for the application.

It can support the app development process from start to finish, from developing, deploying, running to scaling the application. The popular services include EC2, AWS Lambda, Amazon Lightsail and Elastic Beanstalk.

1.2 AWS Storage Services

AWS storage services can be used to store, govern, access and analyze data to ensure that costs are reduced and speed is improved.

Services in this domain include: Amazon S3, EBS, S3 Glacier, and Elastic file storage.

1.3 AWS Database Services

AWS has many different database services to use. These services are divided into two types: relational databases and non-relational (NoSQL) databases.

Relational databases keep data in tables with columns and rows, and we can use SQL to get the data. Columns are like categories(attributes), and rows are like individual entries(records). Each place in the table holds a piece of data.

Relational databases aren't always the best choice when we need high performance or flexible scaling. NoSQL databases, or non-relational databases, don't use tables with columns and rows. This makes it easier to spread out and process data. NoSQL is often used for handling big data, which is large amounts of unstructured or semi-structured data.

2. Types of AWS Web Hosting

2.1 Amazon Elastic Compute Cloud (EC2)

EC2 offers a scalable and flexible solution for deploying virtual machines (instances) in the cloud. It allows users to quickly set up and manage virtual servers, making it easy to adjust resources as needed.

Advantages:

- Handles increased traffic or workload easily
- Enables quick addition of new servers
- Helps businesses respond quickly to changing demands
- Allows infrastructure scaling as required

EC2 is used for businesses that need a scalable and flexible computing infrastructure.

2.2 AWS Lambda

AWS Lambda is a serverless service that runs your code in response to events without requiring you to manage servers. It is useful for building applications such as data processing, chatbots, and web apps.

Advantages:

- Cost-effective: Users only pay for the compute time their code runs.
- Automatic scaling: Handles many requests or events, ensuring applications can manage any level of traffic.
- Easy integration: Works seamlessly with other AWS services.

2.3 Amazon Lightsail

Amazon Lightsail is designed to be ease to use and affordable, making it perfect for small businesses and individuals.

Advantages:

- User-friendly
- Cost-effective

3. Application hosting with Amazon Amplify

3.1 What is Amazon Amplify

AWS Amplify is all that a frontend developer needs to develop and deploy cloud-based full-stack applications without hassle. We can easily connect our frontend to the cloud for data modelling, authentication, storage, serverless functions, SSR app deployment, and more.

3.2 AWS Amplify Features

1. Data: Realtime APIs can be build quickly, securely and easily backed by AWS databases.

2. Auth: Amplify enables secure authentication flows and control access to data, files and more.

3. Storage: Amplify stores and manages app content and data.

4. Functions: Functions can be added which will configure environment variables.

5. Extensibility: Any AWS services can be added, extended and customized by simply authoring CDK code.

6. Mobile: Leverage a single backend across web and mobile apps.

3.3 Step by Step Process for integrating Amazon Amplify into the App

1. Create an AWS Account: Sign up for an AWS account if you don't have one.

2. Install the Amplify CLI: Open your terminal and run the following command to install the Amplify CLI:
"npm install -g @aws-amplify/cli"
Verify the installation by typing amplify in the terminal.

3. Configure Amplify: Use the command amplify configure to start the configuration process.

- Log in to your AWS account.
- Select your preferred region.
- Create a username or use the default one.
- Click "Next" in the new window.
- Select "Amplify Administrator Access" and click "Next".
- Copy the Access Key ID and Secret Access Key.
- Paste the Access Key ID and Secret Access Key in the terminal.
- Create a profile name.

4. Create a Flutter Application: Set up your Flutter application if you haven't already.

5. Initialize Amplify in the Flutter Project

- Navigate to the root folder of your Flutter project.
- Run the following command:
  "amplify init"
- Press Enter to use the default project settings.
- Select "AWS profile" and press Enter.
- Choose the profile name you created during the configuration (e.g., "declarativecast") and press Enter.

6. Set Up Amplify Authentication

- Run the following command to add authentication:
  "amplify add auth"
- Select "Default configuration".

- Choose "Username" for sign-in.
- Select "No" for advanced settings.

7. Push Changes to AWS

- Run the following command to apply the changes to the cloud:
  “amplify push”

Follow these steps, to successfully set up AWS Amplify for Flutter application.

4. AWS Storage Pricing

AWS pricing is based on your usage of each individual service. The total combined usage of each service will create your monthly bill.

4.1 AWS S3 Pricing

Based on four types of usage:

1. where you store your website content

2. the amount you store

3. the number of requests you or your users make to store new content or retrieve the content

4. the amount of data that is transferred from Amazon S3 to you or your users

4.2 AWS Route 53

Pricing will be based on three things:

1. number of domain names managed

2. the number of Hosted Zones you use

3. the number of end user queries for your website.

4.3 AWS Cloud front

Pricing for the cloud front will be based on three things:

1. the amount of data you transfer to your end users

2. the number of user requests

3. CloudFront locations you use (which will depend on the global diversity of your users).