



# Application Deployment on AWS

Architecture Details and Services Suggestions

## Contents

<b>AWS - Flow Chart Creation Application Deployment on Cloud .....</b>	<b>2</b>
<b>There are two ways to use AWS for your application .....</b>	<b>2</b>
<b>1) Infrastructure Managed by the User .....</b>	<b>2</b>
<b>Explanation and Reasons for Suggested Services: .....</b>	<b>3</b>
<b>Our Application Case High-Level Overview .....</b>	<b>4</b>
<b>2) Complete Infrastructure is Managed by AWS .....</b>	<b>5</b>
<b>Conclusion .....</b>	<b>6</b>

# AWS - Flow Chart Creation Application Deployment on Cloud

Important Points for Consideration:

- The application User base is low.
- Application is entirely based on JS Frameworks.
- The complete Application (3 Layers) should be deployed on AWS Cloud.
- The database used is Supabase.
- **The client wants a fully managed infrastructure by AWS with low stress.**

## There are two ways to use AWS for your application

1. Infrastructure Managed by User (we have more customization)
2. Complete Infrastructure is Managed by AWS (gives us a Stress-free experience)

### 1) Infrastructure Managed by the User

In this case, the user has to manage low-level infrastructure by himself requires technical knowledge and skills.

Such as:

**Create Your Infrastructure:** Using AWS CDK to create an Amazon EC2 instance.

**Automate configuration:** Use the EC2 user data to install all packages, configure them, and deploy the application.

**Deploy your stack:** Bootstrap and deploy the AWS CDK stack.

Here are some high-level steps you can follow to **design the architecture**:

1. Identify the components of your application and their dependencies.
2. Decide on the appropriate AWS services to host each component (e.g. VPC, Amazon EC2, Amazon S3, Amazon RDS, AWS LoadBalancer, Amazon ElastiCache, etc.).
3. Decide on the appropriate deployment strategy for your application components (e.g. single instance, multiple instances behind a load balancer, auto-scaling, etc.).
4. Design the networking and security for your application infrastructure (e.g. VPC, subnets, security groups, etc.).
5. Consider the scalability, availability, and resilience requirements of your application and design the infrastructure accordingly.

## Explanation and Reasons for Suggested Services:

Here are some popular AWS services for deploying applications along with the reasons for choosing them:

**Amazon Elastic Compute Cloud (EC2):** EC2 is a scalable computing service that provides on-demand access to virtual servers. It's a good choice for applications that require a flexible and scalable infrastructure, as well as the ability to run custom configurations.

**Amazon Elastic Beanstalk:** Beanstalk is a fully managed service that makes it easy to deploy, run, and scale web applications. It's a good choice for applications that have a simple architecture and don't require extensive custom configurations.

**Amazon S3:** S3 is an object storage service that can be used to store and serve static assets, such as images and videos, for your application. It's a good choice for applications that require highly durable and scalable storage.

**Amazon Relational Database Service (RDS):** RDS is a fully managed database service that makes it easy to set up, operate, and scale a relational database for your application. It's a good choice for applications that require a relational database, as it removes the need for manual database administration.

**Amazon Lambda:** Lambda is a serverless computing service that runs your code in response to events, such as an HTTP request. It's a good choice for applications that require short-lived, event-driven processing and don't require a dedicated server.

**Note:** The appropriate service or combination of services will depend on the specific requirements of your application, such as its architecture, scale, and performance needs. It's important to carefully evaluate your requirements and choose the services that best meet your needs.

## Our Application Case High-Level Overview

For an application that takes code from users, converts it into flow charts, and has a frontend and backend developed in JavaScript-based frameworks with a Supabase database, the following AWS services could be considered for deployment:

**Amazon EC2:** EC2 can be used to host the frontend and backend components of the application, as it provides scalable and customizable compute resources.

**Amazon S3:** S3 can be used to store and serve the frontend assets, such as images and videos, for the application.

**Amazon RDS for Supabase:** RDS for Supabase is a managed service that makes it easy to set up, operate, and scale a Supabase database for your application.

**Amazon Elastic Load Balancer (ELB):** ELB can be used to distribute incoming traffic to multiple EC2 instances, providing high availability and automatic scaling for the application.

**Amazon Route 53:** Route 53 can be used to manage and route DNS traffic to the application, providing a scalable and highly available DNS service.

## 2) Complete Infrastructure is Managed by AWS

For a stress-free deployment and management of your application, you can use AWS services that provide a managed infrastructure. Here are some popular AWS services that can help with this:

**Amazon Elastic Beanstalk:** Beanstalk is a fully managed service that makes it easy to deploy, run, and scale web applications. It handles all the underlying infrastructure, freeing you from the need to manage individual components like servers and databases.

Amazon says Write your code, AWS deploys and manages it

- ✓ Upload your code, AWS handles everything from capacity provisioning, load balancing, and auto-scaling to application health monitoring. This option is best if you want:
- ✓ Widest support of back-end programming languages such as Java, .NET, Go, Ruby, PHP, Node.js, and Python
- ✓ Globally scalable, fully managed infrastructure
- ✓ Simple to get started, flexibility to customize
- ✓ Deploy a non-containerized application to the cloud
- ✓ Package a Node.js app to be deployed using Elastic Beanstalk
- ✓ Create all the infrastructure needed for Elastic Beanstalk using CDK
- ✓ Update a non-containerized deployment

**Amazon App Runner:** App Runner is a new service from AWS that makes it easy to automate the building, testing, and deployment of applications. It provides a simplified, guided workflow for building and deploying your applications, making it an ideal choice for stress-free deployment and management.

- ✓ AWS App Runner is a fully managed container application service that lets you build, deploy, and run containerized web applications and API services without prior infrastructure or container experience.
- ✓ Build and run secure web applications at scale, without prior container or infrastructure experience.
- ✓ Scale your applications cost-effectively, with high availability and low latency.
- ✓ Meet your infrastructure and compliance requirements while staying focused on your application.
- ✓ Connect to database, cache, and message queue services on AWS that support your applications.

**Amazon Amplify:** Amplify is a fully managed service that makes it easy to build, deploy, and host mobile and web applications. It provides a comprehensive set of services for frontend and backend development, including authentication, APIs, and storage. Amplify is well-suited for building cloud-powered, modern web and mobile applications, with an emphasis on front-end development. If your

focus is on building cloud-powered, modern web and mobile applications with an emphasis on front-end development, Amplify may be a better choice.

- ✓ Amplify is a fully managed service that makes it easy to build, deploy, and host mobile and web applications.
- ✓ It provides a comprehensive set of services for frontend and backend development, including authentication, APIs, and storage.
- ✓ Amplify is well-suited for building cloud-powered, modern web and mobile applications, with an emphasis on front-end development.
- ✓ It provides an easy-to-use CLI, UI, and backend APIs, making it a great choice for developers who want a streamlined development experience.

## **Conclusion**

In conclusion, the choice between Elastic Beanstalk and Amplify will depend on the specific requirements and constraints of your application. If your focus is on web application deployment and you need a simple and fast way to deploy your application without having to manage the underlying infrastructure, Elastic Beanstalk may be the better choice. If your focus is on building cloud-powered, modern web and mobile applications with an emphasis on front-end development, Amplify may be a better choice.