Tarea 2: Despliegue en AWS

1. Diagrama una arquitectura en AWS que permita escalar la aplicación.
2. Despliegue la aplicación en AWS utilizando EC2, Auto Scaling Groups, Load Balancer, RDS, etc.
3. Asegure los accesos y credenciales en AWS.

   Este punto se cumple utilizando AWS Identity and Access Management (IAM) en la fase de Seguridad, Monitoreo y Logging.
4. Proporcione alta disponibilidad y plan de recuperación ante desastres.

   Este punto no se garantiza con los servicios y componentes mencionados en la tabla. Para lograr la alta disponibilidad y el plan de recuperación ante desastres, se requerirían configuraciones adicionales, como la implementación en múltiples regiones y el uso de servicios de replicación y respaldo. Estos aspectos deben ser considerados en el diseño y la configuración de la arquitectura específica.
5. La aplicación debe estar accesible públicamente sobre HTTPS.

   Este punto se cumple utilizando Amazon CloudFront en combinación con Amazon S3 para proporcionar acceso público a la aplicación React a través de HTTPS.

Diagrama de servicios en AWS para la aplicación

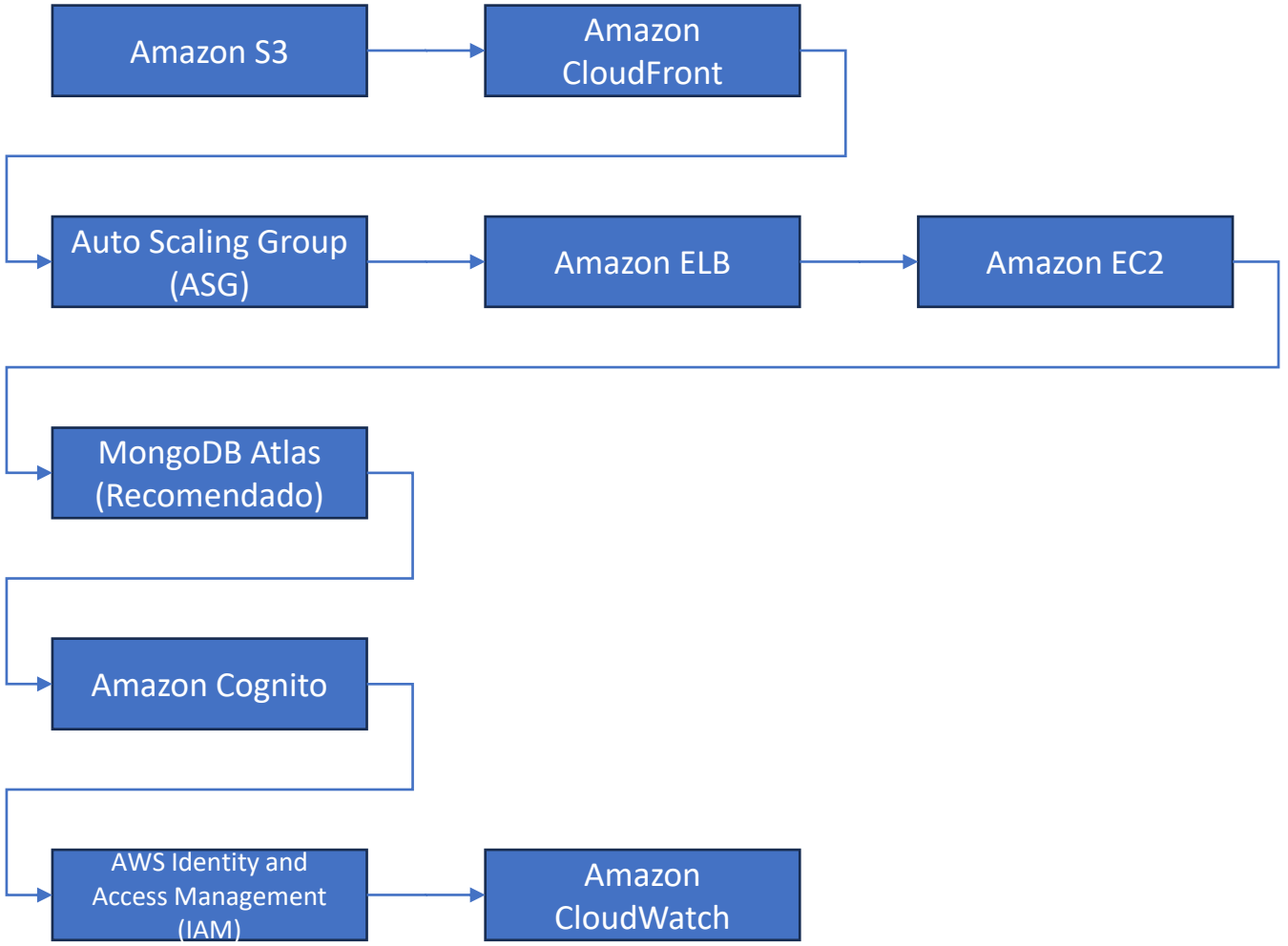
1. Frontend (Cliente React)

2. Backend (Node.js)

3. Base de Datos (MongoDB)

4. Autenticación

5. Seguridad, Monitoreo y Logging



En este diagrama, el usuario accede al sistema a través de CloudFront, que es un servicio de distribución de contenido. Luego, la aplicación React se almacena en Amazon S3 y se entrega al usuario a través de CloudFront.

La interacción con el backend se realiza a través de un equilibrador de carga de aplicaciones (ALB) que dirige las solicitudes a instancias EC2, que ejecutan una aplicación Node.js en un grupo de escalado automático (ASG).

El backend se comunica con MongoDB Atlas o DocumentDB, que son servicios de base de datos administrados en la nube.

Para la autenticación, se utilizan interacciones con Amazon Cognito, que proporciona flujos de autenticación y gestión de usuarios.

En este diagrama actualizado, IAM se muestra como un servicio fundamental en la arquitectura del sistema. IAM se encuentra en la capa de gestión de identidades y permisos, y es responsable de administrar los roles y permisos de los diferentes componentes del sistema, como instancias EC2, bases de datos, servicios y usuarios.

Finalmente, se utiliza CloudWatch para el monitoreo y las alertas del sistema.

Servicio AWS	Servicio GCP	Componente de la aplicación	Fase
Amazon S3	Google Cloud Storage	Frontend (Cliente React)	Frontend
Amazon CloudFront	Google Cloud CDN	Frontend (Cliente React)	Frontend
Auto Scaling Group (ASG)	Google Cloud Instance Group	Backend (Node.js)	Backend
Amazon ELB / ALB (Application Load Balancer)	Google Cloud Load Balancing	Backend (Node.js)	Backend
Amazon EC2	Google Compute Engine	Backend (Node.js)	Backend
MongoDB Atlas (Recomendado)	Google Cloud Firestore o Cloud Datastore	Base de Datos (MongoDB)	Backend
Amazon Cognito	Google Identity Platform	Autenticación	Backend
AWS Identity and Access Management (IAM)	Google Cloud Identity and Access Management	Seguridad, Monitoreo y Logging	Monitoreo
Amazon CloudWatch	Google Cloud Monitoring	Seguridad, Monitoreo y Logging	Monitoreo

Diagrama de Arquitectura General:

1.Usuario Accede:

Usuario -> CloudFront -> S3 (React App).

2.Interacción con Backend:

React App -> ALB -> EC2 Instances (Node.js en ASG).

3.Backend a Base de Datos:

EC2 Instances -> MongoDB Atlas / DocumentDB.

4.Autenticación:

Interacciones con Amazon Cognito para flujos de autenticación.

5.Monitoreo y Logging:

CloudWatch para monitoreo y alertas.