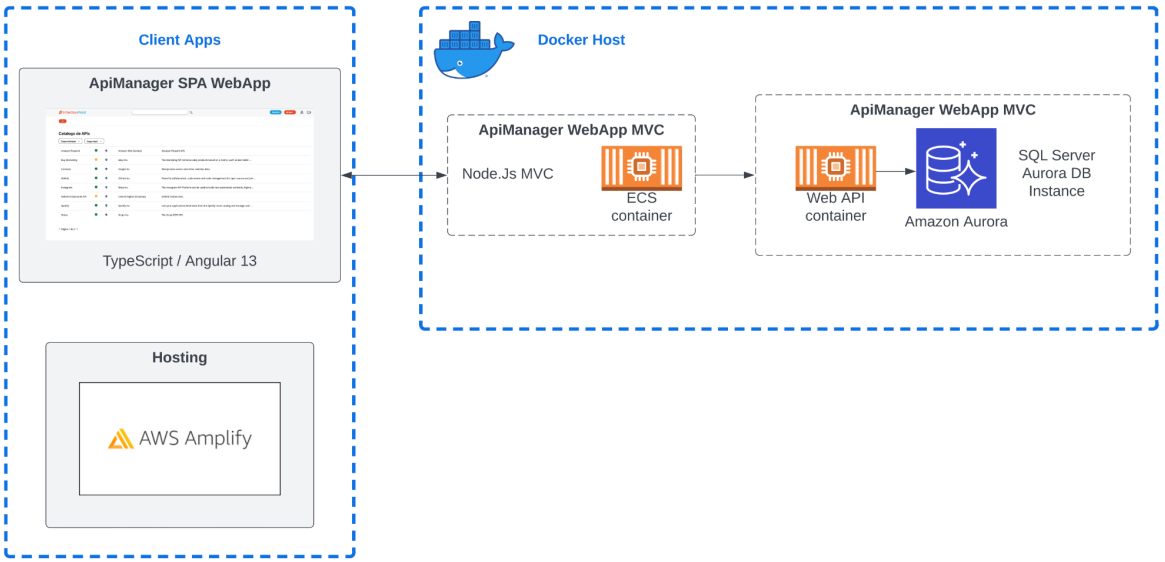
Cristóbal Alberto Escamilla Sada	A00827074
Fernando Doddoli Lankenau	A00827038
Aldo Beráin Cárdenas	A00827874
Juan Pablo Yáñez González	A00829598
Samuel Vieira Restrepo	A00828215

Deployment Detallado

Lo que queremos lograr es tener separación de nuestros componentes para poder tener un sistema robusto y escalable. Este es el diagrama que decidimos.



Base De Datos

Vamos a tener nuestra base de datos en RDS donde vamos a tener una instancia de MS SQL. No necesitamos una instancia muy poderosa para esta aplicación entonces nos vamos a ir por el *free tier*.

Configuration

Engine type Info

- Amazon Aurora
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

DB instance size

- Production: db.r5.xlarge, 4 vCPUs, 32 GiB RAM, 500 GiB, 3.198 USD/hour
- Dev/Test: db.m5.large, 2 vCPUs, 8 GiB RAM, 100 GiB, 0.993 USD/hour
- Free tier: db.t2.micro, 1 vCPUs, 1 GiB RAM, 20 GiB, 0.025 USD/hour

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

apimanager-db-instance

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info

Type a login ID for the master user of your DB instance.

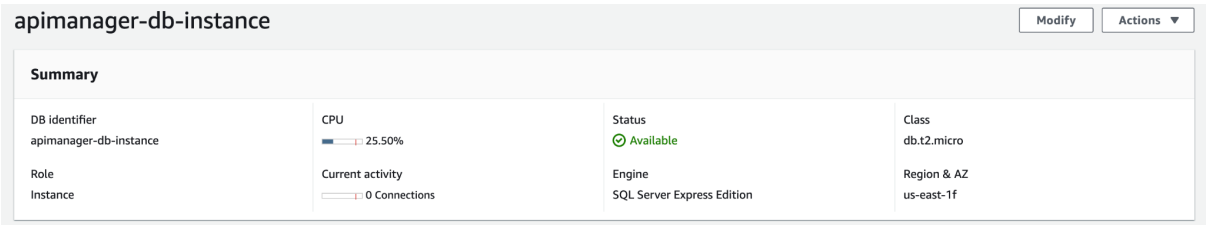
admin

1 to 16 alphanumeric characters. First character must be a letter.

Auto generate a password

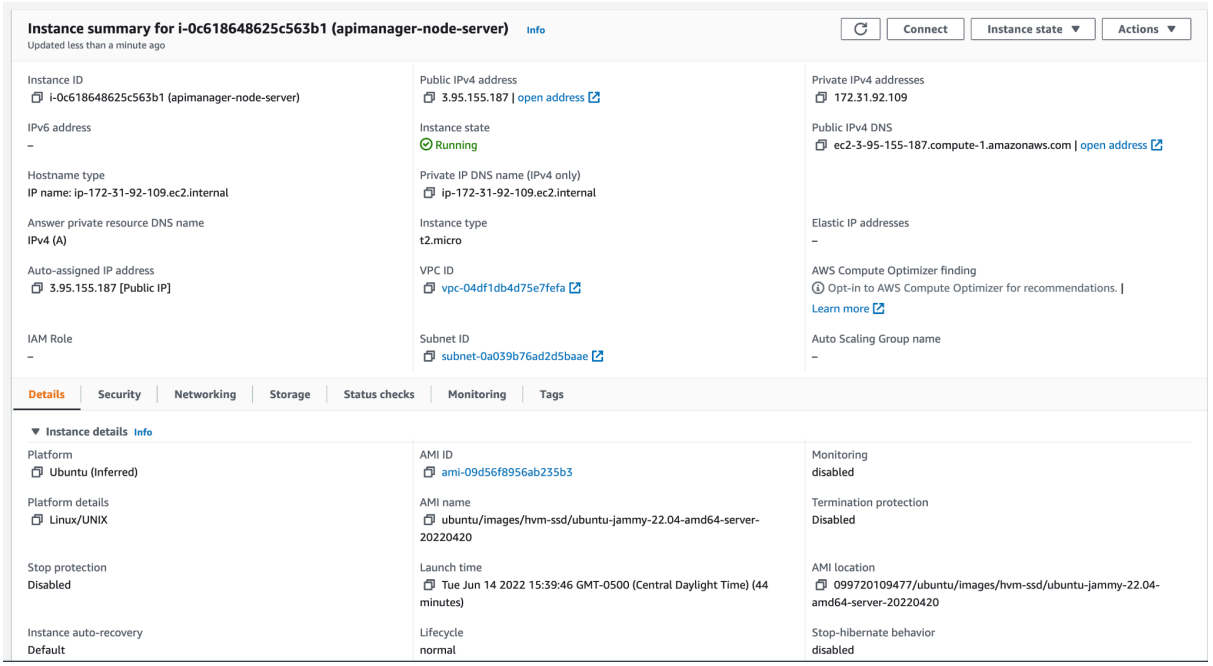
Amazon RDS can generate a password for you, or you can specify your own password.

Aquí podemos ver un resumen de la instancia que fue creada.



Node.Js Back-end

Para el backend de Node vamos a utilizar un contenedor elástico de EC2. Este contiene una imagen de Ubuntu Server, lo cual genera mucha versatilidad para nuestra aplicación, ya que podemos acceder a toda la funcionalidad de un sistema operativo con el kernel de Linux.



Front-end

Ya tenemos listas las bases de datos y el Web API que nos va a permitir tener la información dinámica en la aplicación web. Ahora debemos desplegar la aplicación en sí. Para esto vamos a utilizar AWS Amplify que simplifica inmensamente el proceso. Lo único que debemos hacer es establecer el repositorio que queremos desplegar, y generar la construcción.

Estos son los detalles del app de Amplify que creamos: