

Cristóbal Alberto Escamilla Sada

A00827074

Fernando Doddoli Lankenau

A00827038

Aldo Beráin Cárdenas

A00827874

Juan Pablo Yáñez González

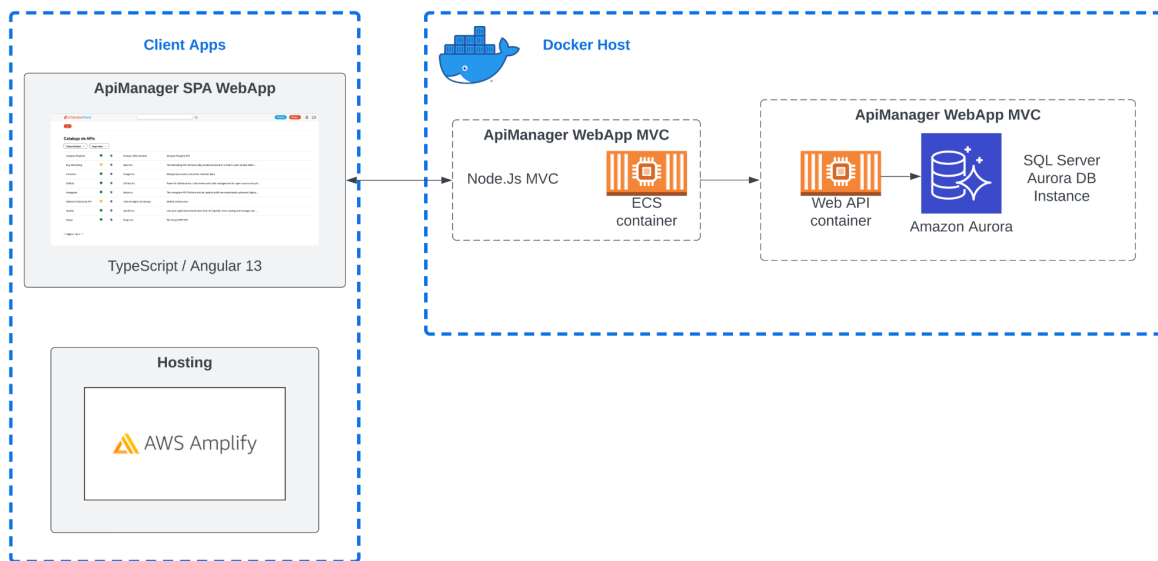
A00829598

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](#)

<input type="radio"/> Amazon Aurora 	<input type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 	<input checked="" type="radio"/> Microsoft SQL Server 

DB instance size

<input type="radio"/> Production db.r5.xlarge 4 vCPUs 32 GiB RAM 500 GiB 3.198 USD/hour	<input type="radio"/> Dev/Test db.m5.large 2 vCPUs 8 GiB RAM 100 GiB 0.993 USD/hour	<input checked="" type="radio"/> 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.

apimanager-db-instance				Modify	Actions ▼
Summary					
DB identifier apimanager-db-instance	CPU <div><div></div></div> 25.50%	Status Available	Class db.t2.micro		
Role Instance	Current activity <div></div> 0 Connections	Engine SQL Server Express Edition	Region & AZ us-east-1f		

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.

Instance summary for i-0c618648625c563b1 (apimanager-node-server)

Info

Updated less than a minute ago

Refresh

Connect

Instance state

Actions

Instance ID i-0c618648625c563b1 (apimanager-node-server)	Public IPv4 address 3.95.155.187 open address	Private IPv4 addresses 172.31.92.109
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-95-155-187.compute-1.amazonaws.com open address
Hostname type IP name: ip-172-31-92-109.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-92-109.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 3.95.155.187 [Public IP]	VPC ID vpc-04df1db4d75e7fefa	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0a039b76ad2d5baae	

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Instance details Info

Platform Ubuntu (Inferred)	AMI ID ami-09d56f8956ab235b3	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20220420	Termination protection Disabled
Stop protection Disabled	Launch time Tue Jun 14 2022 15:39:46 GMT-0500 (Central Daylight Time) (44 minutes)	AMI location 099720109477/ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20220420
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled

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:

App details

Reconnect repository

Edit

<div>App name</div> <div>MTY.TC3004B.102.2211-Equipo3-ApiManager</div> <div>Source repository</div> <div>https://github.com/tecnologico-de-monterrey-oficial/MTY.TC3004B.102.2211-Equipo3/tree/master</div> <div>Production branch URL</div> <div>https://main.d1ly95vaak2tc2.amplifyapp.com</div> <div>Framework</div> <div>Angular</div>	<div>App ARN</div> <div>arn:aws:amplify:us-east-1:061873689996:apps/d1ly95vaak2tc2</div> <div>Created at</div> <div>6/14/2022, 4:27:48 PM</div> <div>Updated at</div> <div>6/14/2022, 4:56:08 PM</div>
--	--

Settings

Production branch

main

Service role

-

Branch autodetection

Automatically connect branches to Amplify Hosting that match a pattern set.

☐ Disabled

Branch auto-disconnection

Automatically disconnect branches in Amplify Hosting when deleting a branch in your source repository

☐ Disabled