

Deploy de la Aplicación AIDA en un Servidor en la Nube

Objetivo:

Aprender a desplegar (deployar) una aplicación Node.js con una base de datos productiva en la nube utilizando **Render**, entendiendo las etapas del proceso de build, configuración y ejecución.

0. Encuesta

Creamos una nueva encuesta para que nos cuenten un poco como vienen experimentando la cursada. Porfavor intenten completarla que nos ayuda a mejorar!

<https://docs.google.com/forms/d/e/1FAIpQLScb1mloBNP0UhYo1YsiblntleUzShdBdNyNhAK8GIumGFoBQ/viewform?usp=dialog>

1. Introducción

Hasta ahora venimos trabajando en nuestra aplicación **AIDA** en entorno local y recientemente agregamos una base de datos externa. En esta clase vamos a dar el siguiente paso: **publicarla online para que cualquiera pueda acceder a ella** desde un dominio público.

¿Qué es la nube (cloud)?

La nube hace referencia a servidores remotos accesibles por Internet que permiten ejecutar aplicaciones, almacenar datos y ofrecer servicios sin necesidad de infraestructura local. En lugar de tener el código corriendo en nuestras propias máquinas, usamos los recursos de un proveedor que los mantiene activos y disponibles todo el tiempo.

En la materia *Redes de Comunicaciones y Cómputo Distribuido* se da más profundidad a estos temas.

¿Qué es Render?

[Render](#) es una plataforma de hosting en la nube que permite desplegar aplicaciones de forma **sencilla**, conectándolas directamente con GitHub.

Permite ejecutar proyectos Node.js, Python, Go, etc., sin tener que configurar servidores manualmente.

¿Por qué usamos Render?

Porque se puede usar de forma gratuita :)

Existen otros proveedores de Cloud mucho más famosos como Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). Son libres de utilizar otro proveedor si les es más cómodo.

Limitaciones del Plan Gratuito

Tengan en cuenta que Render “duerme” las aplicaciones cuando no reciben tráfico durante un tiempo.. Esto significa que la primera vez que entran a la pagina, o si pasan varios minutos sin usarlo, el servicio puede tardar unos minutos en responder

2. Crear Cuenta en Render

Crear una cuenta en Render vinculándose con la cuenta de Github en donde tienen su proyecto. Les va a hacer distintas preguntas sobre qué uso piensan darle a Render respondan que es por motivos académicos y que es solo un desarrollador para que les deje continuar sin ingresar una tarjeta de crédito.



Create an account



Dado que vamos a estar usando el plan gratuito, no está habilitado compartir el proyecto con otras personas, por lo que el dueño del repositorio debe ser el que se conecte.

3. Crear nuevo Web Service

Una vez creada su cuenta, crean un nuevo **Web Service**. Acá en la sección de *Provider* deberia aparecerles su Repositorio de Github

New Web Service

Adicionen los valores para Name, Language* y Branch.

*para grupos que no estén usando lenguajes que estén incluidos en la lista avisar a los docentes para ver como proceder!

Name
A unique name for your web service.
un-nombre-fabuloso

Project Optional
Add this web service to a [project](#) once it's created.
Create a new project to add this to?
You don't have any projects in this workspace. [Projects](#) allow you to group resources into environments so you can better manage related resources.
+ Create a project

Language
Choose the [runtime environment](#) for this service.
Node

Branch
The Git branch to build and deploy.
main

Asegúrense de utilizar la versión Free y de cargar las variables de ambientes que necesiten. La versión productiva debe apuntar a su base de datos productiva (Supabase)

Pick an Instance Type

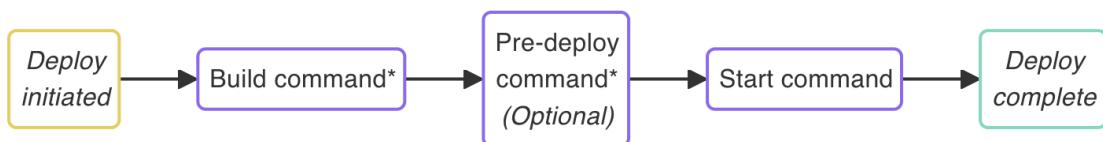
For hobby projects	Free 512 MB (RAM) \$0 / month 0.1 CPU	⚠ Upgrade to enable more features Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.									
For professional use For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:											
<ul style="list-style-type: none"> ▪ Zero Downtime ▪ SSH Access ▪ Scaling ▪ One-off jobs ▪ Support for persistent disks 											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;"> Starter 512 MB (RAM) \$7 / month 0.5 CPU </td> <td style="width: 33%; text-align: center;"> Standard 2 GB (RAM) \$25 / month 1 CPU </td> <td style="width: 33%; text-align: center;"> Pro 4 GB (RAM) \$85 / month 2 CPU </td> </tr> <tr> <td colspan="2" style="text-align: center;"> Pro Plus 8 GB (RAM) \$175 / month 4 CPU </td> <td style="text-align: center;"> Pro Max 16 GB (RAM) \$225 / month 4 CPU </td> </tr> <tr> <td colspan="2" style="text-align: center;"> Pro Ultra 32 GB (RAM) \$450 / month 8 CPU </td> <td></td> </tr> </table>			Starter 512 MB (RAM) \$7 / month 0.5 CPU	Standard 2 GB (RAM) \$25 / month 1 CPU	Pro 4 GB (RAM) \$85 / month 2 CPU	Pro Plus 8 GB (RAM) \$175 / month 4 CPU		Pro Max 16 GB (RAM) \$225 / month 4 CPU	Pro Ultra 32 GB (RAM) \$450 / month 8 CPU		
Starter 512 MB (RAM) \$7 / month 0.5 CPU	Standard 2 GB (RAM) \$25 / month 1 CPU	Pro 4 GB (RAM) \$85 / month 2 CPU									
Pro Plus 8 GB (RAM) \$175 / month 4 CPU		Pro Max 16 GB (RAM) \$225 / month 4 CPU									
Pro Ultra 32 GB (RAM) \$450 / month 8 CPU											

Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

KEY	VALUE
DATABASE_URL
PORT
SESSION_SECRET

4. Etapas del Deploy



Build Command	\$
Render runs this command to build your app before each deploy.	✖ Required
Start Command	\$ yarn start
Render runs this command to start your app with each deploy.	✖ Required

Cuales son los comandos para la etapa de build, cual para el pre-deploy y cual para el Start

5. Información del Deploy

Acá tenemos bastante información.

1. Por un lado nos da la URL que tiene nuestro servicio. En este caso aida-slip.onrender.com
2. Nos permite deployar un último commit, uno específico o reiniciar el servicio (apagar y volver a prenderlo)
3. Podemos ver los logs de nuestra aplicación. Cualquier "print" va a aparecer en esa pantalla de logs.

The screenshot shows the Render service dashboard for a project named 'aida'. At the top, there's a summary bar with 'WEB SERVICE' status, 'Node' runtime, 'Free' tier, and a link to 'Upgrade your instance'. To the right are 'Connect' and 'Manual Deploy' buttons, and a dropdown menu with options: 'Deploy latest commit', 'Deploy a specific commit', 'Clear build cache & deploy', and 'Restart service'. A note below the summary says: 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.' Below this is a deployment log entry for 'November 5, 2025 at 1:16 PM' with status 'In Progress'. The log itself shows the deployment process: npm run, npm i, yarn start, tsc, upload build, and successful build. At the bottom, there's a log viewer with tabs for 'All logs' and 'Live tail', and a search bar.

4. Por default, el servidor va deployar la ultima version cada vez que hay un commit nuevo. Piensen que estrategia quieren tener

Auto-Deploy

By default, Render automatically deploys your service whenever you update its code or configuration. Disable to handle deploys manually. [Learn more.](#)

On Commit

After CI Checks Pass

Off

4. Importante

A partir de ahora, la aplicación debería estar siempre disponible y funcionando (descontando la limitación de Render por usar el plan gratuito). Eso significa que no se debe deployar a la versión productiva hasta estar “seguros” de que el código alterado no genera bugs.