

Universidad Modelo



Sistemas Multidimensionales Parcial 2

Juan Carlos López Villajuana

10 de noviembre de 2020

Objetivo

Crear un proyecto de manera local mediante docker y github de manera local, para luego migrarlo a la nube

Alcance del documento

El documento abarca desde el inicio de la creación de un proyecto de manera local, así como subir cada una de las partes del proyecto a github y luego poder subir toda la infraestructura a la nube y configurar algunas opciones de seguridad así como un balanceador de cargas y subdominios para que nuestro proyecto funcione de manera correcta.

Procesos

Repositorios

Creación de Repositorios

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * JuanLoVill / Repository name * API ✓

Great repository names are short and memorable. Need inspiration? How about [potential-system](#)?

Description (optional)

- ☐ Public
Anyone on the internet can see this repository. You choose who can commit.
- ☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

- ☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)
- ☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * JuanLoVill / Repository name * WEBSITE ✓

Great repository names are short and memorable. Need inspiration? How about [potential-system](#)?

Description (optional)

- ☐ Public
Anyone on the internet can see this repository. You choose who can commit.
- ☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

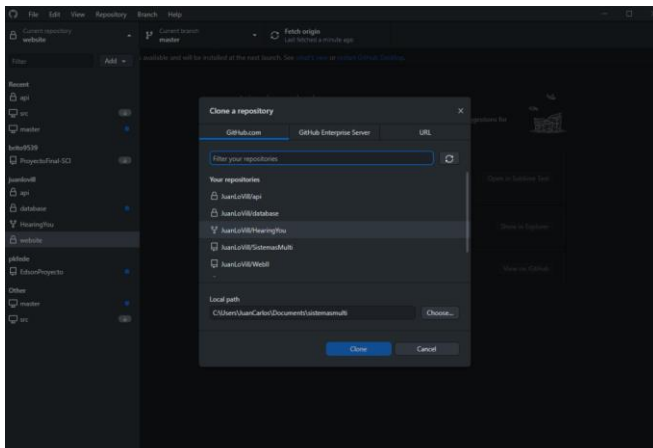
Skip this step if you're importing an existing repository.

- ☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)
- ☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
- ☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Aquí vemos el ejemplo de la creación de 2 repositorios

Clonación de Repositorios en Local



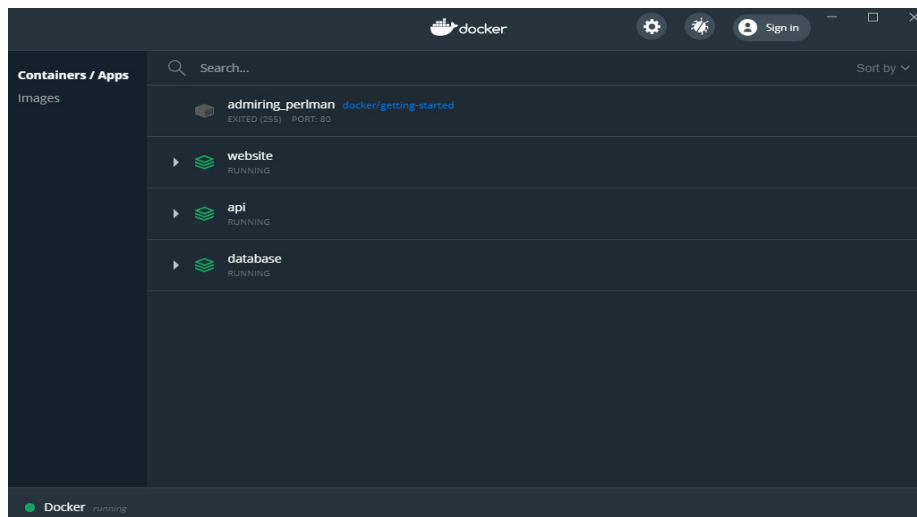
Aquí vemos de manera gráfica como podemos clonar localmente los contenedores aunque de igual manera podemos usar el comando git clone “link del repositorio” para clonar cada uno

Creación del contenedor

```
Command Prompt
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

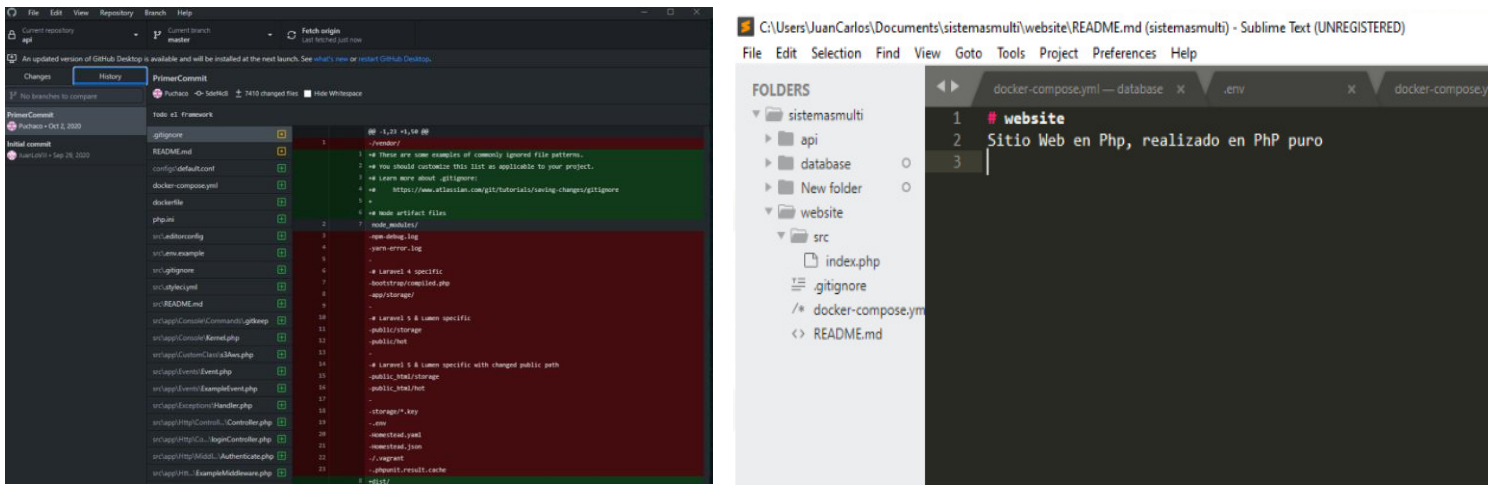
C:\Users\JuanCarlos>cd Documents
C:\Users\JuanCarlos\Documents>cd sistemasmulti
C:\Users\JuanCarlos\Documents\sistemasmulti>cd database
C:\Users\JuanCarlos\Documents\sistemasmulti\database>docker-compose up -d
```

Navegamos a la carpeta donde se encuentra nuestro docker-compose y ponemos el comando que se muestra en la imagen, esto lo tenemos que hacer por cada uno de nuestros docker-compose que tengamos.



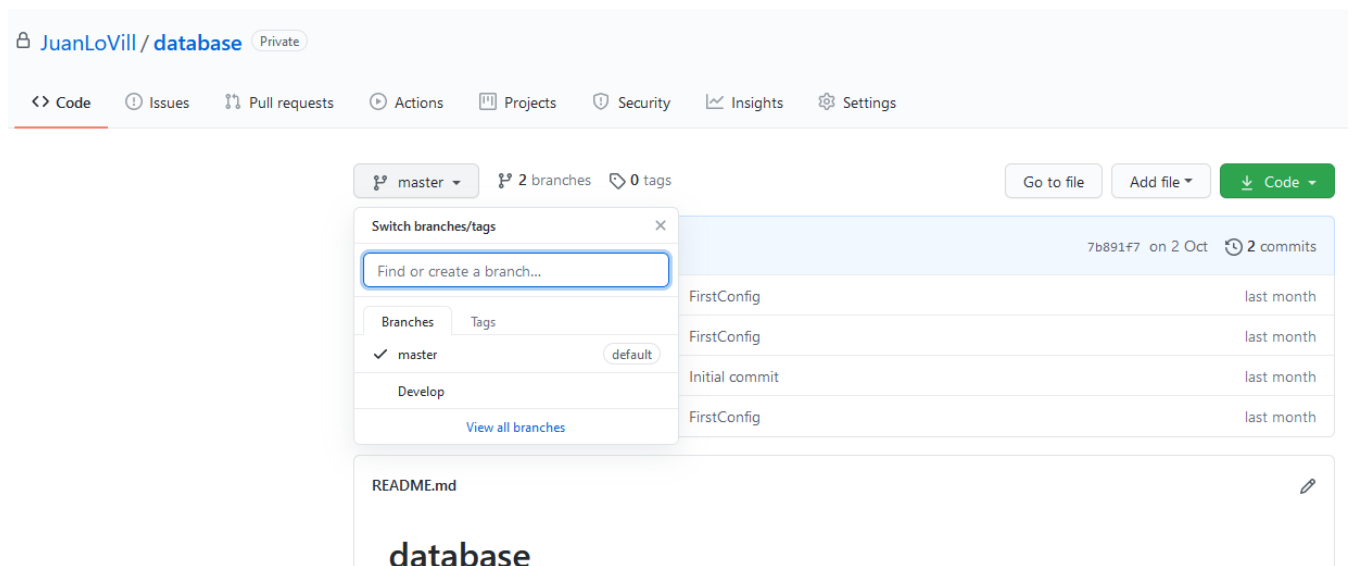
Aquí podemos ver los 3 repositorios ya corriendo

Carga de código fuente inicial



Aquí vemos como el proyecto tienes cosas ya adentro y estas están siendo subidas a github, con los respectivos commits hechos para subirlos

Creación de ramas, Master y Develop




Podemos ver ambas ramas creadas en el contenedor

Nube

Creación de Instancias LightSail

🇰🇷 Seoul (ap-northeast-2)

ZONE A


**webLopez**
512 MB RAM, 1 vCPU, 20 GB SSD

Running

54.180.202.130
Seoul, Zone A


🇯🇵 Tokyo (ap-northeast-1)

ZONE A

**apiLopez**
512 MB RAM, 1 vCPU, 20 GB SSD

Running

3.112.2.85
Tokyo, Zone A

**databaseLopez**
512 MB RAM, 1 vCPU, 20 GB SSD

Running

54.168.191.93
Tokyo, Zone A

Aquí vemos las 3 instancias ya creadas (la manera de instalarlas de manera más específica la vimos en el documento pasado)

Instalación de Docker en Instancias

```
databaseLopez - terminal [Lightbulb - Multiple Profiles]
https://lightsail.amazonaws.com/terminal?ap-northeast-1/instances/databaseLopez/terminal/processor/ssh
c-n-f Metadata [116 B]
Get:26 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64
Packages [4012 B]
Get:27 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe Transl
ation-en [1448 B]
Get:28 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu focal-backports/universe amd64
c-n-f Metadata [224 B]
Get:29 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu focal-backports/multiverse amd6
c-n-f Metadata [116 B]
Get:30 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [82.4 kB]
Get:31 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [5380 B]
Get:32 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [66.5 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [10.9 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [517 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [66.7 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [923
B]
Get:37 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [1256 B]
Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [548 B]
Get:39 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [1
16 B]
Fetched 17.5 MB in 4s (4138 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
128 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-26-13-127:/home/ubuntu# apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4 python3-software-properties
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
The following packages will be upgraded:
  ca-certificates curl libcurl4 python3-software-properties software-properties-common
2 upgraded, 2 newly installed, 0 to remove and 123 not upgraded.
```

Este proceso lo repetimos para cada una de las instancias que estamos utilizando (web, database, api)

Clonación de repositorio

```
oot@ip-172-26-13-127:/home/ubuntu# cd ..
oot@ip-172-26-13-127:/home# git clone https://github.com/JuanLoVill/database.git
Cloning into 'database'...
Username for 'https://github.com': juan.lopez@modelo.edu.mx
Password for 'https://juan.lopez@modelo.edu.mx@github.com':
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 3.18 KiB | 1.59 MiB/s, done.
oot@ip-172-26-13-127:/home# dir
database ubuntu
oot@ip-172-26-13-127:/home#
```



databaseLopez
54.168.191.93

```
ubuntu@ip-172-26-1-35:~$ sudo -s
root@ip-172-26-1-35:/home/ubuntu# cd..
cd..: command not found
root@ip-172-26-1-35:/home/ubuntu# cd ..
root@ip-172-26-1-35:/home# git clone https://github.com/JuanLoVill/api.git
Cloning into 'api'...
Username for 'https://github.com': juan.lopez@modelo.edu.mx
Password for 'https://juan.lopez@modelo.edu.mx@github.com':
remote: Enumerating objects: 9113, done.
remote: Counting objects: 100% (9113/9113), done.
remote: Compressing objects: 100% (5827/5827), done.
remote: Total 9113 (delta 2744), reused 9109 (delta 2744), pack-reused 0
Receiving objects: 100% (9113/9113), 10.83 MiB | 6.21 MiB/s, done.
Resolving deltas: 100% (2744/2744), done.
root@ip-172-26-1-35:/home# dir
api ubuntu
root@ip-172-26-1-35:/home# cd api
root@ip-172-26-1-35:/home/api# apt-get update
Hit:1 http://ap-northeast-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
```



apiLopez
3.112.2.85

Clonamos los repositorios correspondientes dependiendo de la instancia en la que nos encontremos

Montaje de Contenedor

```
oot@ip-172-26-13-127:/home/database# docker-compose up -d
Starting database_database_1 ... done
oot@ip-172-26-13-127:/home/database# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
19f97ec9b65f   mariadb       "docker-entrypoint.s..." About a minute ago Up 15 seconds  0.0.0.0:3306->3306/tcp   database_database_1
oot@ip-172-26-13-127:/home/database#
```



databaseLopez
54.168.191.93

```
oot@ip-172-26-1-35:/home/api# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
9bc51ce65cd   nginx         "/docker-entrypoint..." 28 seconds ago Up 19 seconds  0.0.0.0:86->80/tcp      nginxapiclaase
c6e8fe1b043   api_phpapicla "docker-php-entrypoi..." 28 seconds ago Up 19 seconds  9000/tcp, 0.0.0.0:8006->8000/tcp  phpapiclaase
oot@ip-172-26-1-35:/home/api#
```



apiLopez
3.112.2.85

Aquí vemos el ejemplo de 2 instancias con sus respectivos contenedores ya montados

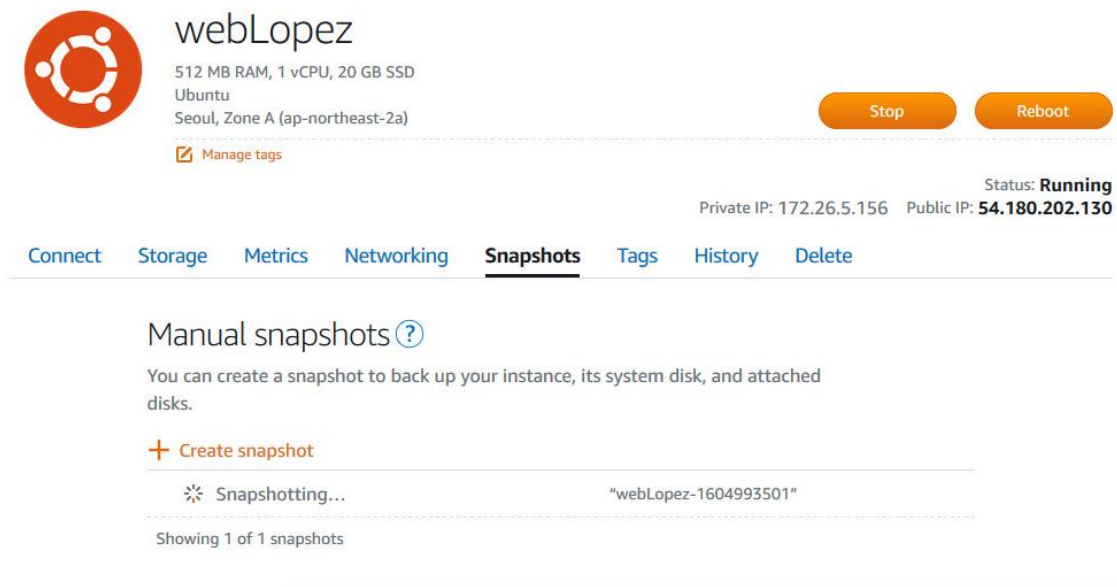
Seguridad y Redundancia

Configuración de firewall



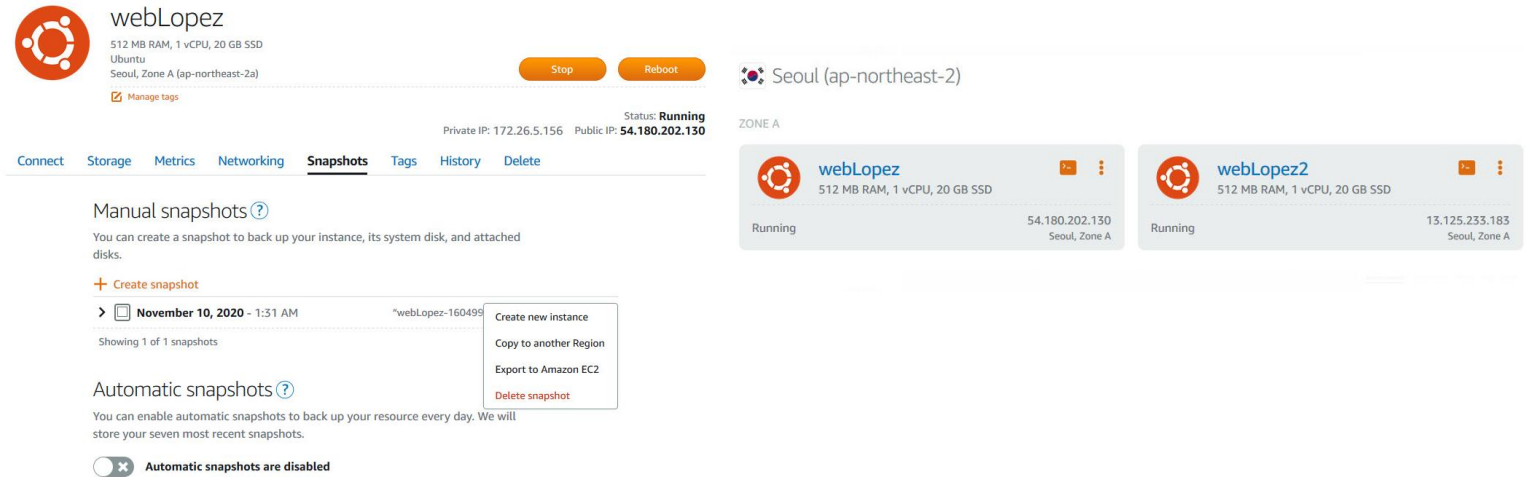
Aquí configuramos los puertos e ips que queramos que pasen a través del firewall

Creación de Instantanea de Web



En la imagen se muestra como se está creando el snapshot de nuestra instancia de web para luego poder meterlas al balanceador de cargas

Creación de segunda Instancia de web



Manual snapshots ?

You can create a snapshot to back up your instance, its system disk, and attached disks.

+ Create snapshot

Showing 1 of 1 snapshots

Automatic snapshots ?

You can enable automatic snapshots to back up your resource every day. We will store your seven most recent snapshots.

Automatic snapshots are disabled

Creamos la segunda instancia de nuestra web mediante una snapshot y podemos ver como ambas instancias están creadas correctamente.

Creación de subdominio para Web



A record — Associate your domain or a subdomain with an IP address.

Subdomain

lopez .idts.com.mx

Resolves to

webLopez

Aquí creamos el subdominio para nuestra instancia

Creación de balanceador de cargas

Identify your load balancer

Your Lightsail load balancers must all have unique names.

LoadBalancerLopez

TAGGING OPTIONS

Use tags to filter and organize your resources in the Lightsail console. Key-value tags can also be used to organize your billing, and to control access to your resources. [Learn more about tagging.](#)

Key-only tags ?

+ Add key-only tags

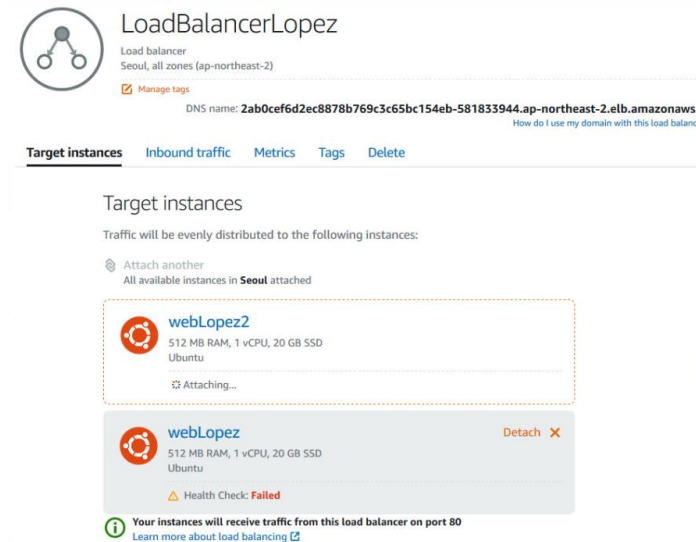
Key-value tags ?

+ Add key-value tag

This load balancer will cost 18 USD per month.

Create load balancer

Asignación de Instancias a balanceador de carga



Aquí asignamos al balanceador de carga las instancias de las que se va a encargar

Creación de certificado SSL

Certificates ?

You may create and store up to two SSL/TLS certificates per load balancer to choose from

Create a certificate
You must associate SSL/TLS certificates with a domain name to encrypt connections.

PRIMARY DOMAIN
Domain names must be a series of labels separated by '.'

CERTIFICATE NAME
Your Lightsail resources must have unique names.

ALTERNATE DOMAINS AND SUBDOMAINS

You can add up to 9 optional domains and subdomains for this certificate.

[Understanding SSL/TLS certificates](#)

Cancel Create

Creamos el certificado SSL para nuestro dominio y subdominio

Resumen

Un proyecto que se encuentra de manera local en servidores propios, es mucho más fácil de manejar, replicar y mantener funcionando si se utilizan las tecnologías de los contenedores de Docker y github para mantener todo en su propio ecosistema sin preocuparnos por compatibilidades y si aún mejor esto se sube a instancias en la nube es aún más fácil de manejar y replicar.

Aquí vimos primeramente como importamos un proyecto de manera local a github y a contenedores de Docker mediante docker-compose, y creamos las ramas master y develop en github para tener un mejor control.

Después nos pasamos a la nube, en donde creamos las instancias de lightsail y se le instaló Docker a cada una de las instancias para poder jalar cada uno de los repositorios y subir toda la infraestructura del proyecto a la nube en poco tiempo

Por último nos encargamos de la seguridad de nuestras instancias configurando los puertos y las ips que van a tener acceso a nuestros contenedores y además creamos una instantánea de nuestra instancia de web, para que creamos un balanceador de cargas y este se encargue de redirigir el tráfico entre las dos y que el sitio pueda atender mejor las peticiones que se le hagan y por último le creamos un certificado SSL a nuestro dominio y subdominios deseados.