

Segundo Parcial
Sistemas Multidimensionales

**Ingeniería en Desarrollo de tecnología y
software**

Brito Fuentes Alejandro

11/10/2020



Contenido

1. Objetivo.....	3
2. Alcance.....	3
3. Procesos.....	4
3.1 Repositorio	4
3.1.1 Creación de Repositorios	4
3.1.2 Clonación de repositorios en local	5
3.1.3 Creación del contenedor.....	5
3.1.4 Creación de ramas, Master y Develop.....	6
3.2 Nube	7
3.2.1 Creación de instancias Lightsail.....	7
3.2.1 Instalación Docker en instancias	7
3.2.4 Montaje de Contenedor	8
3.2.5 Configuración de aplicaciones	9
3.3 Seguridad y Redundancia.....	10
3.3.1 Configuración de firewall (Puertos e Ip's).....	10
3.3.2 Creación instantánea de Web	10
3.3.3 Creación de segunda instancia de Web	11
3.3.4 Creación de subdominio para Web	11
3.3.5 Creación de balanceador de carga	12
3.3.6 Asignación de instancias a balanceador de carga.....	12
3.3.7 Creación de certificado SSL.....	13
4. Resumen.....	14



1.Objetivo

Realizar una pagina web con su respectiva base de datos y sus apis de manera local mediante Github y contenedores en github para luego poder subirlos a la nube a través de Amazon Lightsail mediante una instancia en Linux Ubuntu. Y de esta manera poder crear snapshots y respaldos de nuestra pagina web de manera mucho mas sencilla y sin riesgo a perderla ya que le añadiremos seguridad a la misma.

2. Alcance

A través de este documento explicaremos de manera ilustrativa como crear un repositorio de tu pagina web junto con sus respectiva base de datos y sus Apis en la plataforma de GitHub, y poder clonaras de manera local a tu computadora y con esto levantar repositorios en Docker con nuestro Docker compose, y una vez este montado todo de manera local abarcaremos la parte de poder subirlo a la nube por medio de Amazon lightsail y crear las instancias de nuestros repositorios y configurarlos a través de terminales de Linux Ubuntu y una vez configurados poder aañadirles seguridad creando puertos para nuestro firewall y con la creación de nuestro balanceador de carga así como también la creación de nuestro certificado ssl.

Todo lo anterior es un pequeño resumen de lo que aprenderemos en este documento de manera ilustrativa y con sus respectivas indicaciones.



3. Procesos

3.1 Repositorio

3.1.1 Creación de Repositorios

Creación de repositorios llamados API, website y database en github.

Create a new repository

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

Owner **Brito9539** / Repository name **API** ✓

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

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 **Brito9539** / Repository name **database** ⚠

Great repository names are short and memorable. Need inspiration? How about [miniature-octo-carnival?](#)

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 **Brito9539** / Repository name **website** ✓

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

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

Overview Repositories Projects Packages

Find a repository... Type: All Language: All New

database Private
Updated 27 seconds ago

website Private
Updated 58 seconds ago

API Private
Updated 2 minutes ago

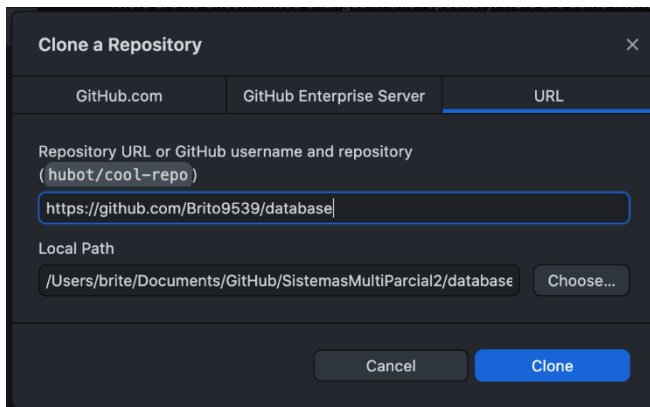
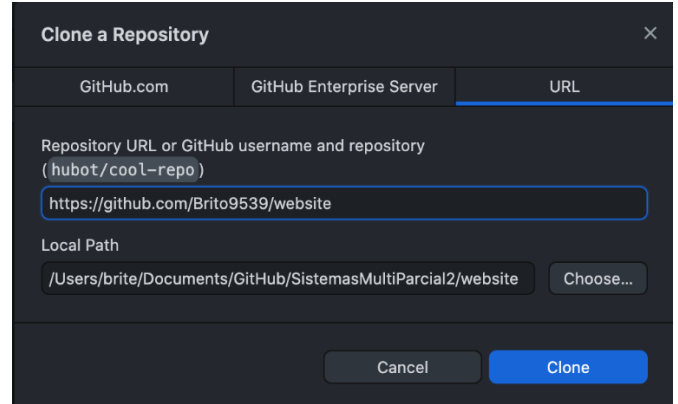
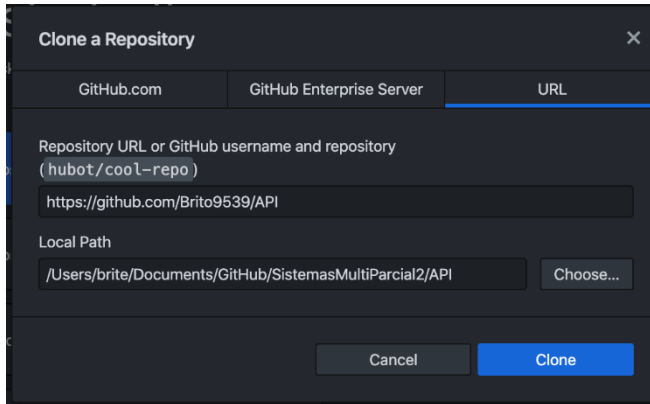
Alejandro Brito
Brito9539

Edit profile



3.1.2 Clonación de repositorios en local

Clonamos nuestros repositorios localmente desde GitHub desktop



3.1.3 Creación del contenedor

Creamos nuestros contenedores de Docker desde la terminal y en la última foto podemos ver cómo están levantados desde Docker desktop.

```
database — -bash — 80x24
Last login: Mon Nov  9 11:48:08 on console

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) MBPdeAlejandro:~ brite$ cd Documents
(base) MBPdeAlejandro:Documents brite$ cd Github
(base) MBPdeAlejandro:Github brite$ cd SistemasMultiParcial2
(base) MBPdeAlejandro:SistemasMultiParcial2 brite$ cd database
(base) MBPdeAlejandro:database brite$ docker-compose up -d
```

```
API — -bash — 80x24
latest: Pulling from library/mariadb
6a5697faee43: Pull complete
ba13d3bc422b: Pull complete
a254829d9e55: Pull complete
2ee2cadd29fc: Pull complete
6915a184049d: Pull complete
5ca6ffdb5f56: Pull complete
1537f7bbe8b: Pull complete
5790e54322d1: Pull complete
ea98cb829471: Pull complete
a46bde67834a: Pull complete
7833acd64cf3: Pull complete
5ffdad591cf6: Pull complete
Digest: sha256:2960a3d1ddb35dd454066b45005b4f694e18af76648833f1b9d93ab90cee7cf2
Status: Downloaded newer image for mariadb:latest
Creating database_database_1 ... done
(base) MBPdeAlejandro:database brite$ cd
(base) MBPdeAlejandro:~ brite$ cd API
-bash: cd: API: No such file or directory
(base) MBPdeAlejandro:~ brite$ cd Documents
(base) MBPdeAlejandro:Documents brite$ cd Github
(base) MBPdeAlejandro:Github brite$ cd SistemasMultiParcial2
(base) MBPdeAlejandro:SistemasMultiParcial2 brite$ cd API
(base) MBPdeAlejandro:API brite$ docker-compose up d
```

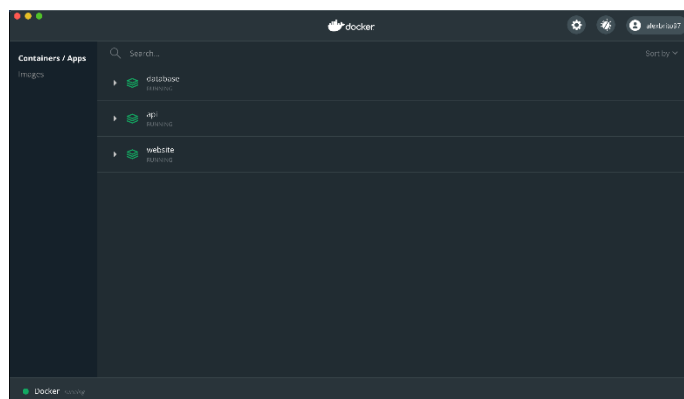


```

-20180731/
find . -name *.gcno -o -name *.gcdc | xargs rm -f
find . -name *.lo -o -name *.o | xargs rm -f
find . -name *.la -o -name *.a | xargs rm -f
find . -name libs -a -type d|xargs rm -rf
rm -f libphp.la modules/* libs/*
Removing intermediate container 13ce6d20b267
--> 84e946112862
Step 7/7 : COPY php.ini $PHP_INI_DIR/php.ini
--> 80b3fffe5514

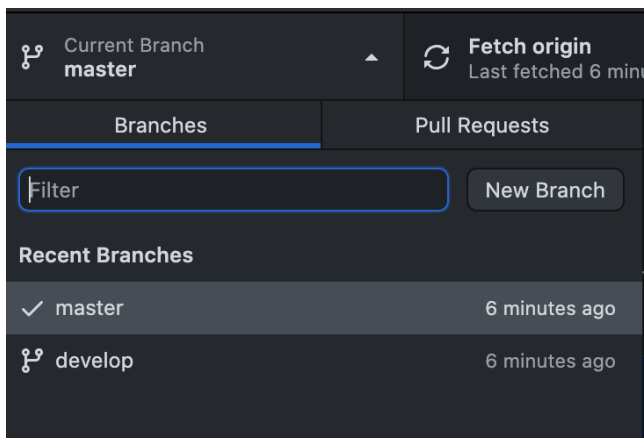
Successfully built 80b3fffe5514
Successfully tagged api_phpapicase:latest
WARNING: Image for service phpapicase was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.
Creating nginxapicase ... done
Creating phpapicase ... done
(base) MBPdeAlejandro:API brite$ cd ..
-bash: cd: command not found
(base) MBPdeAlejandro:API brite$ cd ..
(base) MBPdeAlejandro:SystemasMultiParcial2 brite$ cd website
(base) MBPdeAlejandro:website brite$ docker-compose up -d

```



3.1.4 Creación de ramas, Master y Develop

Creación de rama master y develop en GitHub Desktop



3.2 Nube

3.2.1 Creación de instancias Lightsail

Creamos nuestras database, web y api en amazon lightsail.

Identify your instance

Your Lightsail resources must have unique names.

databaseBrito x 1

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

Create instance

Identify your instance

Your Lightsail resources must have unique names.

databaseBrito x 1

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

Create instance

Identify your instance

Your Lightsail resources must have unique names.

websiteBrito x 1

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

Create instance

Tokyo (ap-northeast-1)

ZONE A

websiteBrito
512 MB RAM, 1 vCPU, 20 GB SSD
Running
54.238.189.214
Tokyo, Zone A

Singapore (ap-southeast-1)

ZONE A

apiBrito
512 MB RAM, 1 vCPU, 20 GB SSD
Running
54.255.240.162
Singapore, Zone A

databaseBrito
512 MB RAM, 1 vCPU, 20 GB SSD
Running
13.229.123.205
Singapore, Zone A

3.2.1 Instalación Docker en instancias

Instalamos Docker y Docker compose en nuestras 3 instancias desde la terminal, también se pueden crear snapshots para no tener que instalarlo en cada una.

```
databaseBrito - Terminal [Lightsail]
https://github.com: amazon.com/s3/terminal/instances/databaseBrito/terminal/protocol-shell
Unpacking cgroups-mount (1.4) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../2-containerd.io_1.3.7-1_amd64.deb ...
Unpacking containerd.io (1.3.7-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5.3a19.03.13-3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-cli (5.19.03.13-3-0-ubuntu-bionic) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5.3a19.03.13-3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5.19.03.13-3-0-ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../5-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1.4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.3.7-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service - /lib/systemd/system/containerd.service.
Processing triggers for ureadahead (0.100.0-20) ...
Setting up cgroups-mount (1.4) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up docker-ce-cli (5.19.03.13-3-0-ubuntu-bionic) ...
Setting up docker-ce (5.19.03.13-3-0-ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service - /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket - /lib/systemd/system/docker.socket.
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
root@ip-172-26-5-145:~#
```

```
apiBrito - Terminal [Lightsail]
https://github.com: amazon.com/s3/terminal/instances/apiBrito/terminal/protocol-shell
Unpacking cgroups-mount (1.4) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../2-containerd.io_1.3.7-1_amd64.deb ...
Unpacking containerd.io (1.3.7-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5.3a19.03.13-3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-cli (5.19.03.13-3-0-ubuntu-bionic) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5.3a19.03.13-3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5.19.03.13-3-0-ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../5-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1.4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.3.7-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service - /lib/systemd/system/containerd.service.
Processing triggers for ureadahead (0.100.0-20) ...
Setting up cgroups-mount (1.4) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up docker-ce-cli (5.19.03.13-3-0-ubuntu-bionic) ...
Setting up docker-ce (5.19.03.13-3-0-ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service - /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket - /lib/systemd/system/docker.socket.
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
root@ip-172-26-14-28:~#
```



```
Unpacking cgroups-mount (1.4) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../2-containerd.io_1.3.7-1_amd64.deb ...
Unpacking containerd.io (1.3.7-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a19.03.13-3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce-cli (5:19.03.13-3-0-ubuntu-bionic) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a19.03.13-3-0-ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5:19.03.13-3-0-ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../5-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.3.7-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/syst
amd/systemd/containerd.service.
Processing triggers for ureadahead (0.100.0-20) ...
Setting up cgroups-mount (1.4) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up docker-ce-cli (5:19.03.13-3-0-ubuntu-bionic) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (5:19.03.13-3-0-ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/
system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/syst
em/docker.socket.
Processing triggers for ureadahead (0.100.0-20) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.3) ...
root@ip-172-26-7-105:~#
```

3.2.4 Montaje de Contenedor

Montamos nuestro contenedor de Docker en nuestras instancias

```
Unpacking objects: 100% (6/6), done.
root@ip-172-26-7-105:/home# cd website
root@ip-172-26-7-105:/home/website# docker-compose up -d
Creating network "website_default" with the default driver
Pulling wesbsiteclase (webdevops/php-apache)...
latest: Pulling from webdevops/php-apache
5bed26d33875: Pull complete
f11b29a9c730: Pull complete
930bda195c84: Pull complete
78bf9a5ad49e: Pull complete
4c8b85719399: Pull complete
30611fcc1b5e: Pull complete
f70f19977fc5: Pull complete
5131985698ec: Pull complete
65538e0c8933: Pull complete
d43f884b2c0e: Pull complete
20d592fa47f1: Pull complete
d3f32720a7f3: Pull complete
2f8bf6213034: Pull complete
8356225c2f81: Pull complete
Digest: sha256:acf080564bcb304227fa26af92457dd486b37c812debd7b0042b84010268d4bd
Status: Downloaded newer image for webdevops/php-apache:latest
Creating wesbsiteclase ... done
root@ip-172-26-7-105:/home/website#
```

```
Build complete.
Don't forget to run 'make test'.

Installing shared extensions:      /usr/local/lib/php/extensions/no-debug-non-zts-20180731/
find . -name \*.gcno -o -name \*.gcda | xargs rm -f
find . -name \*.lo -o -name \*.o | xargs rm -f
find . -name \*.la -o -name \*.a | xargs rm -f
find . -name \*.so | xargs rm -f
find . -name *.libs -a -type d|xargs rm -rf
rm -f libphp.la      modules/* libs/*
Removing intermediate container 22029855a36f
--> 149abb906ede
Step 7/7 : COPY php.ini $PHP_INI_DIR/php.ini
--> e2f28be3de15

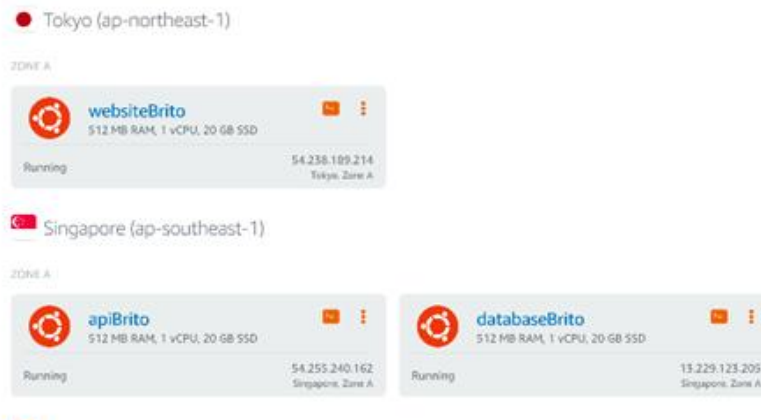
Successfully built e2f28be3de15
Successfully tagged api_phpapiclase:latest
WARNING: Image for service phpapiclase was built because it did not already exist. To rebu
ild this image you must use 'docker-compose build' or 'docker-compose up --build'.
Creating nginxapiclase ... done
Creating phpapiclase ... done
root@ip-172-26-14-28:/home/API#
```




```
root@ip-172-26-5-145:/home/database# docker-compose up -d
Creating network "database_default" with the default driver
Pulling database (mariadb:...)
latest: Pulling from library/mariadb
5a5697faee43: Pull complete
ba13d3bc422b: Pull complete
a254829d9e55: Pull complete
2ee2cadd29fc: Pull complete
3915a184049d: Pull complete
5ca6ffdb5f56: Pull complete
1537f7bbe8b: Pull complete
5790e54322d1: Pull complete
aa98cb829471: Pull complete
a46bde67834a: Pull complete
78334cd64cf3: Pull complete
5ffdad591cf6: Pull complete
Digest: sha256:2960a3d1ddb35dd454066b45005b4f694e18af76648833f1b9d93ab90cee7cf2
Status: Downloaded newer image for mariadb:latest
Creating database_database_1... done
root@ip-172-26-5-145:/home/database#
```

databaseBrito
13.229.123.205

3.2.5 Configuración de aplicaciones





3.3 Seguridad y Redundancia

3.3.1 Configuración de firewall (Puertos e Ip's)

Configuramos nuestros puertos y añadimos una rule al firewall.

[Connect](#) [Storage](#) [Metrics](#) **[Networking](#)** [Snapshots](#) [Tags](#) [History](#) [Delete](#)

IP addresses

Public IP ?

13.229.123.205

Create static IP

Private IP ?

172.26.5.145

Private IP addresses allow you to communicate securely with other internal resources.

Firewall ?

Create rules to open ports to the internet, or to a specific IP address or range.

[Learn more about firewall rules](#)

+ Add rule

Application	Protocol	Port or range / Code	Restricted to	
Custom	TCP	3886	Any IP address	

3.3.2 Creación instantánea de Web

Creamos una snapshot para nuestra instancia web.

Manual snapshots ?

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

+ Create snapshot

Give your snapshot a name.

websiteBrito-1605000383

Manual snapshots ?

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

+ Create snapshot

> November 10, 2020 - 3:26 AM

"websiteBrito-1605000372"

Showing 1 of 1 snapshots




3.3.3 Creación de segunda instancia de Web

Una vez creada nuestra snapshot creamos la segunda instancia para web a partir de ella, en mi caso la llame "websiteBritoSnapshot".

Snapshot source

You are restoring from the following snapshot:



websiteBrito-1605000372
November 10, 2020 - 3:26 AM
512 MB RAM, 1 vCPU, 20 GB SSD, instance snapshot

Instance location ?



You are creating this instance in **Tokyo, Zone A** (ap-northeast-1a)
[Change zone](#)

Identify your instance

Your Lightsail resources must have unique names.

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](#)


Create instance

Tokyo (ap-northeast-1)

ZONE A




websiteBrito
512 MB RAM, 1 vCPU, 20 GB SSD
Running
54.238.189.214
Tokyo, Zone A




websiteBritoSnapshot
512 MB RAM, 1 vCPU, 20 GB SSD
Running
18.183.140.162
Tokyo, Zone A

Singapore (ap-southeast-1)

ZONE A



apiBrito
512 MB RAM, 1 vCPU, 20 GB SSD
Running
54.255.240.162
Singapore, Zone A



databaseBrito
512 MB RAM, 1 vCPU, 20 GB SSD
Running
13.229.123.205
Singapore, Zone A

Creamos nuestro subdominio para web.

A record ▼

Subdomain
brito

.idts.com.mx

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

Resolves to
websiteBrito



3.3.5 Creación de balanceador de carga

Creamos nuestro Load balancer desde la pestaña de networking en Amazon Lightsail

Your Lightsail load balancers must all have unique names.

LoadBalancerBrito

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

3.3.6 Asignación de instancias a balanceador de carga

Añadimos nuestras 2 instancias de web a nuestro Load Balancer

Target instances

Traffic will be evenly distributed to the following instances:



Attach another

All available instances in **Tokyo** attached



websiteBrito

512 MB RAM, 1 vCPU, 20 GB SSD
Ubuntu

⌘ Attaching...



websiteBritoSnapshot

512 MB RAM, 1 vCPU, 20 GB SSD
Ubuntu

⌘ Attaching...



Your instances will receive traffic from this load balancer on port 80




3.3.7 Creación de certificado SSL

Creamos nuestro 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](#)

©2008-2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy](#)

Certificates ?

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




idts-com-mx

SSL certificate, idts.com.mx

Requested on: November 10, 2020, 3:48 AM

[Delete](#) ✕

Status:  **Validation in progress...**

You must prove you control the domains and subdomains specified in this certificate before it can be used for HTTPS encryption.

Please create a DNS record for each domain with the following values:

idts.com.mx	Validation
-------------	------------



4. Resumen

Como podemos ver el crear nuestros repositorios, clonarlos de manera local y montar los contenedores de Docker es una práctica muy fácil y esto nos puede ayudar mucho ya que es una manera muy sencilla de montar nuestra página web de manera practica y sencilla. Tal cual vimos en la práctica solo es crear nuestros repositorios en GitHub web y descargarnos GitHub desktop para que sea un poco más fácil la configuración, se puede hacer por medio de GitHub desde la terminal de nuestra computadora como lo hicimos en las terminales de Ubuntu, pero recomiendo que si se pueden descargar GitHub Desktop es mucho más fácil de configurar e ir pusheando nuestros cambios. Y lo mismo recomendaría para Docker, como vimos el montar los contenedores si tiene que ser por medio de la terminal, pero una vez los montemos podemos controlarlos desde nuestro Docker Desktop y es mucho más fácil de visualizar, por lo que también recomiendo Docker desktop.

Y ya una vez hecho todo lo anterior de manera local nos pasamos a la nube, que, aunque al principio puede ser un poco difícil de entender o manejar ya que Amazon Lightsail es una plataforma nueva para quien no la conoce, pero es una plataforma muy fácil de entender y sobre todo muy fácil de usar. Por supuesto tiene un costo, pero considero que no es tan caro para lo que nos brinda. Como pudimos ver el crear nuestras instancias en Lightsail es muy fácil, desde luego las creamos en Linux Ubuntu y entramos a la terminal e instalamos rápidamente Docker y Docker compose y procedemos a clonar nuestros repositorios, después de esto enseguida montamos nuestros contenedores y ya tendríamos montado todo desde la nube. Luego como podemos ver procedemos a configurar la parte de seguridad y es muy fácil, ya que añadimos nuestros puertos para el firewall de manera muy rápido y procedemos a crear nuestra snapshot para nuestra página web y creamos nuestra segunda instancia en caso de que se pueda perder la primera tengamos un respaldo de la misma. También configuramos nuestro dominio comprado, creamos nuestro load balancer y añadimos nuestro certificado SSL y listo así de fácil tenemos montada nuestra pagina web.

Como pudimos ver es una manera muy fácil y segura de montar nuestras paginas web y a la vez es una manera eficiente.