

Proyecto número 2 (My Company S.A. Deployment)

Escuela de Ciencias Aplicadas e Inge-
niería, Departamento de Informática y
Sistemas, Universidad EAFIT

Integrantes

| Nombre | Correo |
|-----------------------------|-------------------------|
| Miguel Angel Cabrera Osorio | macabrerao@eafit.edu.co |
| Juan Manuel Young Hoyos | jmyoungh@eafit.edu.co |
| Juan Carlos Montoya Mendoza | |



Medellín, 4 de noviembre de 2022

Índice

| | | |
|----------|---|----------|
| 1 | Objetivo | 2 |
| 2 | Problemática a resolver | 2 |
| 3 | Solución | 2 |
| 3.1 | Descripción | 2 |
| 3.2 | Configuración | 3 |
| 3.3 | Disponibilidad para miles de usuarios | 11 |
| 3.4 | Demo | 12 |

1 | Objetivo

Desarrollar habilidades en la configuración, así como despliegue de aplicaciones y servicios en red, particularmente las que requieren de una arquitectura cliente/servidor.

2 | Problemática a resolver

La empresa MiCompany S.A., requiere de un mecanismo basado en tecnologías de información (TI) que le permita capturar la opinión de los consumidores en relación con los diferentes productos que tienen en el mercado.

En este sentido, le han solicitado a su empresa de soluciones de TI, que implemente una estrategia que le permita recolectar lo que piensan los usuarios de los productos. En la actualidad, la empresa cuenta con un catálogo de cincuenta (50) productos. Básicamente, lo que se requiere es que el usuario pueda expresar lo que piensa de uno o varios productos en particular, en no más de ciento cincuenta caracteres (150). Tenga en cuenta que, la única intención de la empresa es que, posterior al proceso de recolección de datos, se pueda aplicar técnicas de minería de texto sobre la información recolectada (p.ej, sentiment analysis, topic detection, etc) con el fin de extraer información de cómo perciben los productos los clientes.

El director de TI de su empresa, ha decidido para satisfacer la necesidad, implementar y desplegar una aplicación web para este proyecto. Al respecto, se ha definido que la aplicación debe contar con un proceso de registro de usuarios el cual capture: nombre, edad, ciudad, dirección, correo electrónico, entre otros datos relevantes (información sociodemográfica de los usuarios). De igual forma, la aplicación debe permitir la visualización de cada uno de los productos que tiene la compañía en el mercado y le debe permitir al usuario seleccionar el producto para que este pueda dar su opinión de este. Vale resaltar que un usuario, puede seleccionar uno o varios productos con el fin de dar su opinión de este.

Su área es la encargada de tanto del desarrollo como el despliegue de la aplicación web. Para esto se requiere que usted realice todo el proceso de diseño de la aplicación, la implementación, pruebas y puesta en producción de esta. Con respecto al proceso de despliegue, su empresa ha decidido que ésta se despliegue considerando una infraestructura de TI robusta y escalable para soportar la operación de la aplicación. Todo esto con el fin de garantizar como mínimo una disponibilidad de 99,5 %. Tenga en cuenta que la aplicación debe desarrollarse que soporte de manera concurrente miles de usuarios al igual que debe permitir escalar de manera horizontal.

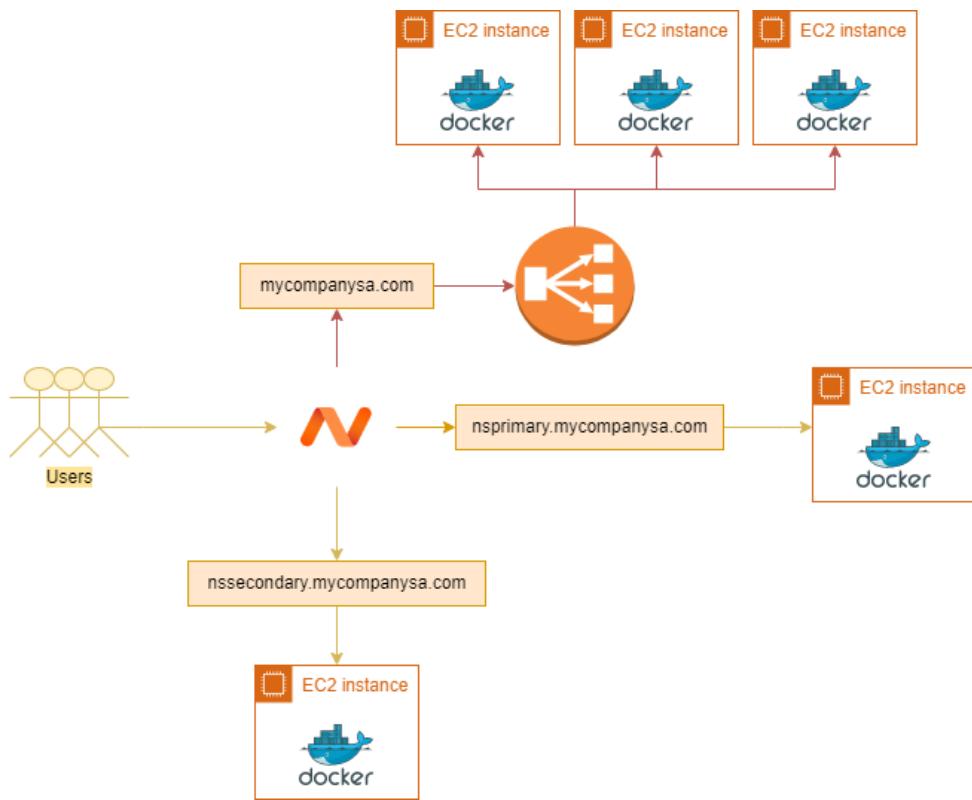
En este sentido, desde la perspectiva del desarrollo de software a nivel de tecnologías, usted puede considerar cualquiera de los Manejadores de Contenido (CMS por sus siglas en inglés) disponibles en la actualidad (Drupal o Wordrpess), con el fin de desarrollar la aplicación para satisfacer la problemática planteada. Para efectos de la persistencia de datos, se debe considerar utilizar bases de datos propias que emplean este tipo de soluciones. Por otro lado, en los aspectos relacionados con el despliegue, se ha decidido que la aplicación web debe desplegarse utilizando un proveedor de computación en nube, utilizando el modelo de infraestructura como servicio (IaaS). Tenga en cuenta que para desplegar la aplicación se requiere que usted considere elementos como balanceadores de carga para lograr un buen despliegue de la solución, de tal forma, que el balanceador de carga sea configurado para distribuir las peticiones entrantes entre diferentes servidores web que se tienen y los usuarios puedan tener acceso a la aplicación web desplegada. Es importante resaltar que, para acceder la aplicación usted debe tener un gestionar y conseguir un dominio público gratis de tal forma que la aplicación debe accederse a través de una URL, como, por ejemplo, <http://www.micompany.tk>.

3 | Solución

3.1 | Descripción

Como se mencionó anteriormente se espera **una escalabilidad horizontal** y principalmente una **alta disponibilidad (99.5 %)**, además de esto el cliente requiere de desplegarlo en *AWS*.

Por lanto, decidimos usar un *balanceador de cargas clásico* de *AWS* para desplegar los *EC2* (Una instancia virtual de un servidor de Amazon) requeridos por la organización. Por lo que decidimos desplegar la siguiente arquitectura:



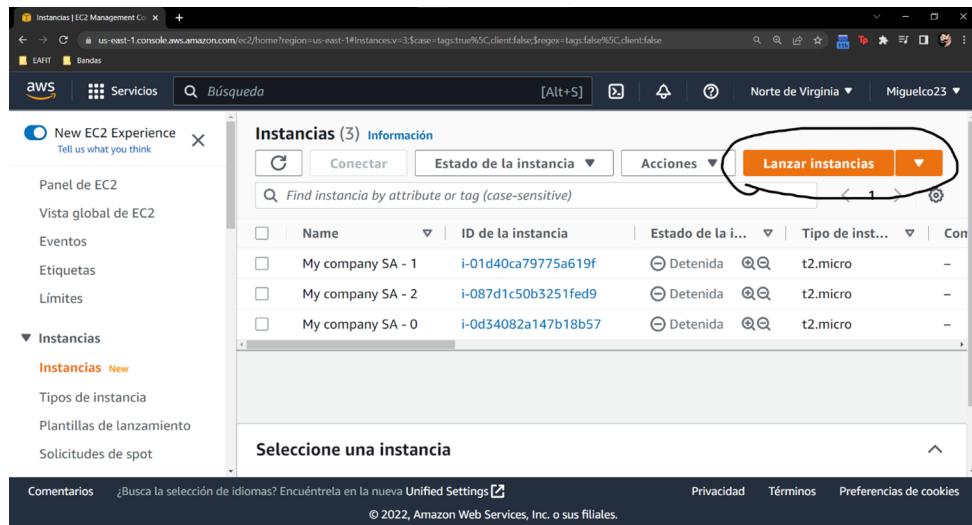
(a) Arquitectura de mycompanysa

3.2 | Configuración

En el siguiente apartado se documentará el proceso de cómo se hizo, esto ya sea para una replica del proceso a futuros proyectos o para la mejora del mismo proyecto.

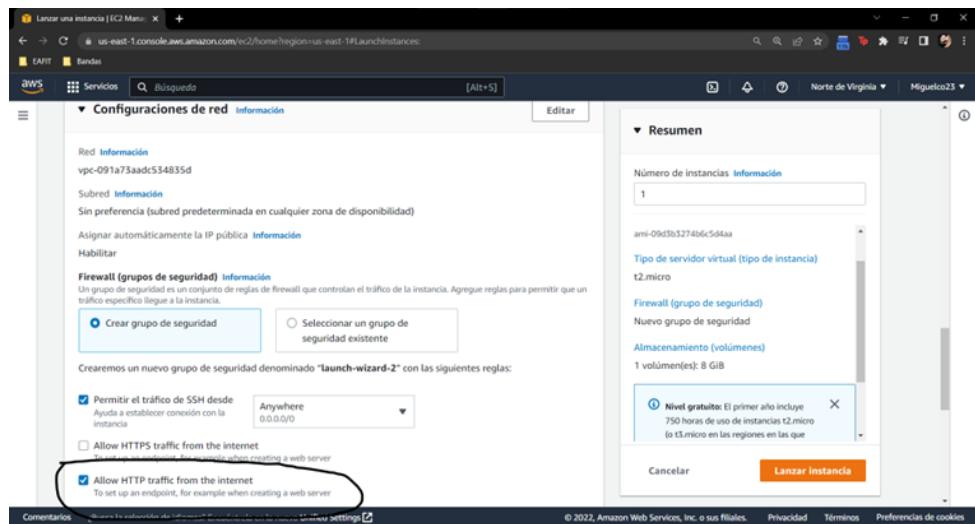
3.2.1 | Creación de Load Balancer con 3 instancias en AWS

Primero debemos entrar en ec2 y crear una nueva instancia en "Lanzar instancia"



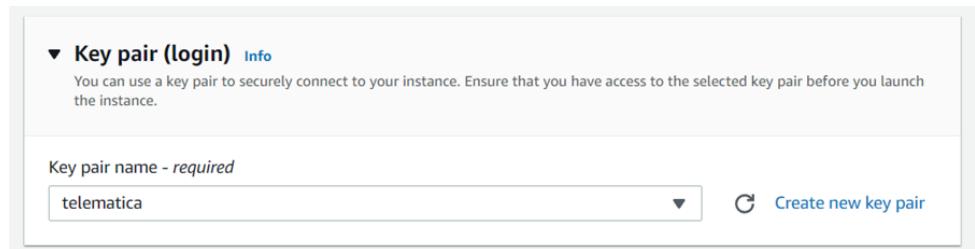
(a) Creamos una nueva instancia

En el área de configuraciones de *red/grupos* de seguridad debemos activar el tráfico a través de *HTTP*.



(a) Configuraciones de red/grupos de seguridad

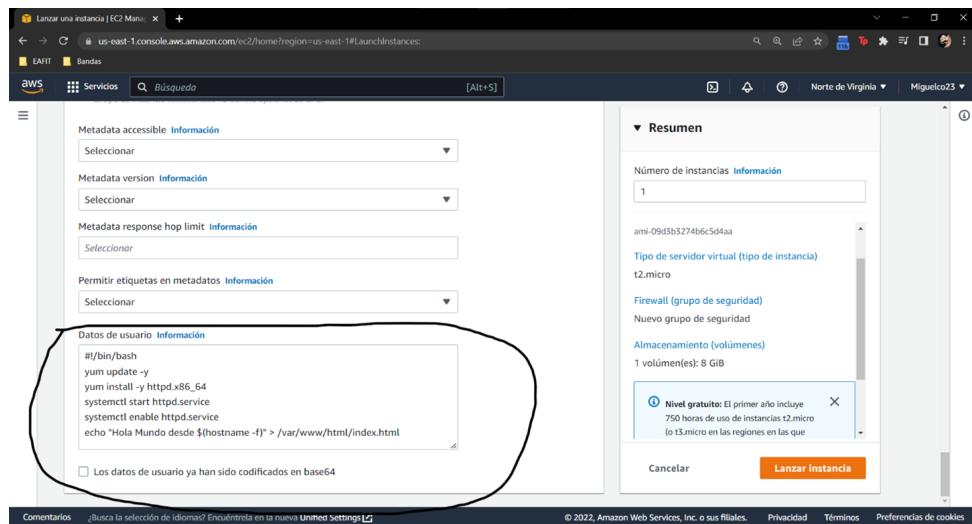
Además, agregamos nuestra llave (Key pair) para acceder a la instancia remotamente.



(a) Opciones Key Par

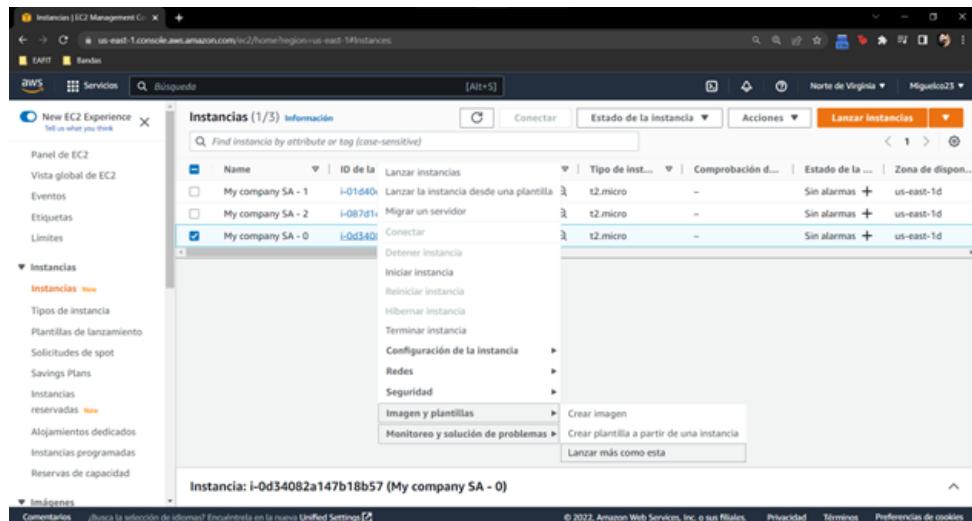
En este caso (Aunque no es estrictamente necesario para crear el balanceador) pondremos un script en la sección de datos de usuario, para así ver a cuál instancia en específico estamos accediendo. Despues de esto lanzamos la instancia.

```
#!/bin/bash
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
echo "Hola Mundo desde $(hostname -f)" > /var/www/html/index.html
```



(a) Script de configuración inicial

Al tener nuestra instancia, damos clic derecho en ella: **Imagen y plantillas/lanzar más como esta.** Para así duplicarla (En este caso lo hicimos dos veces para tener 3 instancias diferentes).



(a) Imagen y plantillas

Para continuar buscamos en el panel de EC2: **equilibrio de carga/balanceadores de carga** y entramos.

The screenshot shows the AWS EC2 Management Console. In the left sidebar, under the 'Instancias' section, there is a 'Equilibrio de carga' (Load Balancing) category which is expanded. Inside this category, 'Balanceadores de carga' (Load Balancers) is selected. A red circle highlights this specific section.

(a) Equilibrio de cargas

Damos click en *Crear balanceador de carga*:

The screenshot shows the 'Crear balanceador de carga' (Create Load Balancer) button highlighted with a red circle. This button is located at the top of the main content area where a new load balancer is being configured.

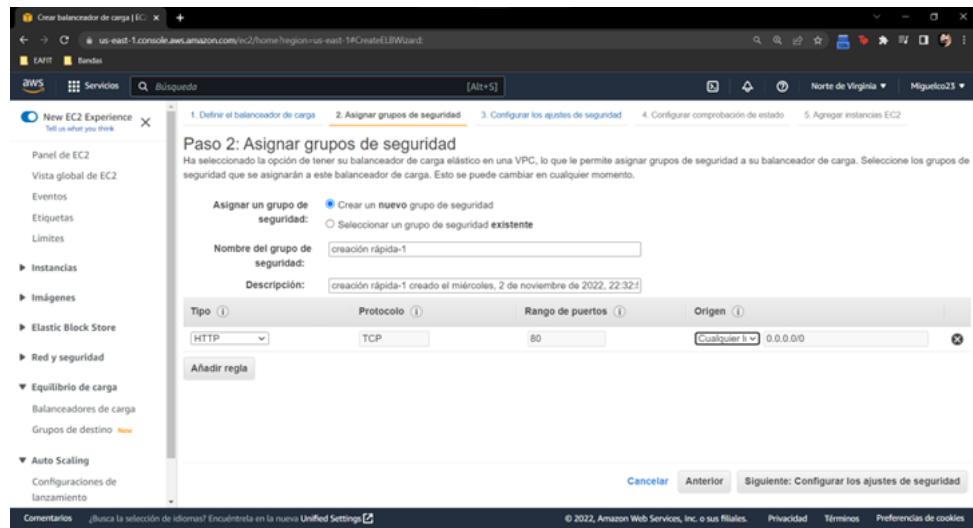
(a) Click en balanceador de cargas

Seleccionamos *Classic load balancer*.

The screenshot shows the 'Create ELB Wizard' step 1. It has two options: 'Classic Load Balancer - previous generation' and 'Classic Load Balancer - Info'. The 'Classic Load Balancer - previous generation' option is selected and highlighted with a red circle. Below it, there is a diagram illustrating how traffic flows from an external source through a CLB (Classic Load Balancer) to multiple EC2 instances via HTTP, HTTPS, TCP, and SSL protocols.

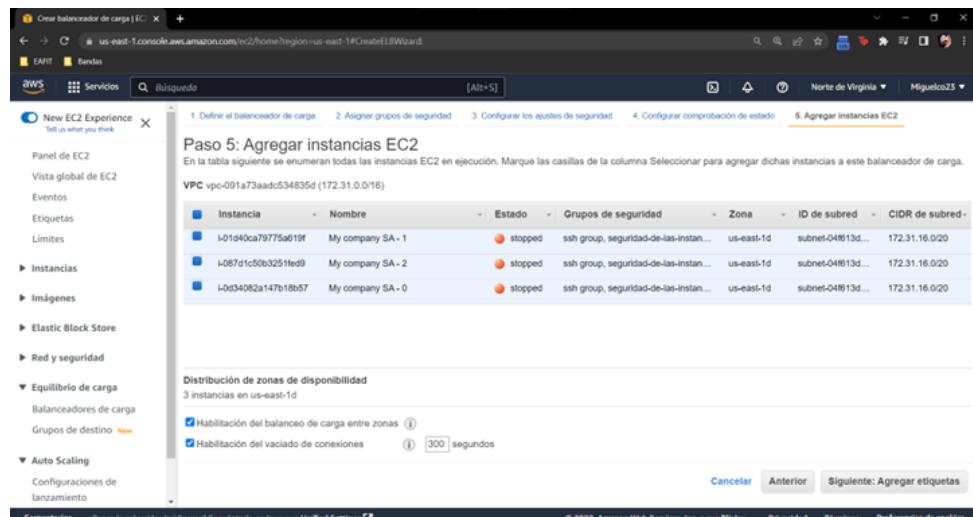
(a) Click en Classic load balancer

En el paso de *“signar grupos de seguridad”* creamos un nuevo grupo con una regla de tipo HTTP y origen cualquier lugar.



(a) Click en *Asignar grupos de seguridad*

En el paso siguiente agregamos las instancias que creamos previamente y que queremos que hagan parte del balanceador.



(a) Agregar las *instancias*

Una vez que nuestro balanceador esté listo, al darle clic, podremos ver debajo el nombre del DNS para acceder a este.

| Nombre | Nombre de DNS | Estado | ID de VPC | Zonas de disponibilidad | Tipo |
|--------------------------|---|--------|-----------------------|-----------------------------|---------|
| My-Classic-Load-Balancer | My-Classic-Load-Balancer-675345738.us-east-1.elb.amazonaws.com (Registro A) | Activo | vpc-091a73aadc534835d | us-east-1f, us-east-1e, ... | classic |

(a) Revisamos las instancias

Al ingresar a nuestro DNS e ir actualizando podemos ver como desde el mismo URL accedemos a las diferentes instancias. Por lo que está funcionando perfectamente.

Hola Mundo desde ip-172-31-16-228.ec2.internal

(a) Las instancias cambian con cada request que realizoas

3.2.2 | Compra del dominio

Lo primero que se realizó fue la compra del dominio, una de las mejores ofertas las ofrece *Namecheap*, además de que el equipo ya contaba con cierta experiencia con esta empresa. Los pasos realizados fueron:

- Registro en la plataforma.
- Búsqueda del dominio deseado.
- Compra del dominio deseado.
- Configuración del dominio

Ahora bien, los 3 primeros pasos son casi mecánicos para casi cualquier compra en línea, pero ¿cómo realizamos el último paso? Bueno, lo primero que haremos es ingresar a nuestro dominio y nos vamos a *Configuración Avanzada*.

The screenshot shows the Namecheap dashboard. On the left, there's a sidebar with links like Dashboard, Expiring / Expired, Domain List, Hosting List, Private Email, SSL Certificates, Apps, and Profile. The main area displays a message about 2FA Authentication App (TOTP) and the account balance (\$0.00). Below that, it says "Hello Juan manuel Young hoyos" and "Last logged in on Nov 4, 2022, 01:38 PM (EST)". There's a search bar and a "Beast Mode" toggle. Under "Recently Active in Your Account", there are two entries: "mycompanyasa.com" (Expiration: Oct 31, 2023) and "jmyounghoyos.com" (Expiration: May 7, 2023). A red arrow points to the "MANAGE" button for the first domain.

(a) Selección del dominio

The screenshot shows the "Domains → Details" page for the domain "mycompanyasa.com". The sidebar is identical to the previous dashboard screenshot. The main area shows the domain name and several tabs: Domain, Products, Sharing & Transfer, and Advanced DNS (which is highlighted with a red arrow). Below these tabs are sections for DNS TEMPLATES and HOST RECORDS, along with actions and filters.

(a) Opción de configuración avanzada de DNS

Después de esto añadiremos las siguientes rutas, los dominios e IPs son las dadas por AWS, entonces se cambiarán las primeras IPs son reemplazadas por las proporcionadas por el balanceador de cargas y los demás dominios hacen referencia a lo que quieras desplegar en los diferentes dominios o subdominios.

The screenshot shows the "HOST RECORDS" configuration page. It has a header with Actions, Filters, and a search bar. The table lists host records:

| Type | Host | Value | TTL |
|--------------|-------------|--|-----------|
| A Record | @ | 52.71.6.188 | Automatic |
| A Record | @ | 52.5.219.25 | Automatic |
| CNAME Record | @ | classic-load-balancer-675345738.us-east-1.elb.amazonaws.com | Automatic |
| CNAME Record | www | my-classic-load-balancer-675345738.us-east-1.elb.amazonaws.com | Automatic |
| CNAME Record | nsprimary | ec2-3-83-165-234.compute-1.amazonaws.com | Automatic |
| CNAME Record | nssecondary | ec2-107-23-152-109.compute-1.amazonaws.com | Automatic |

At the bottom are buttons for "ADD NEW RECORD" and "SAVE ALL CHANGES". A red arrow points to the CNAME record entry for '@'.

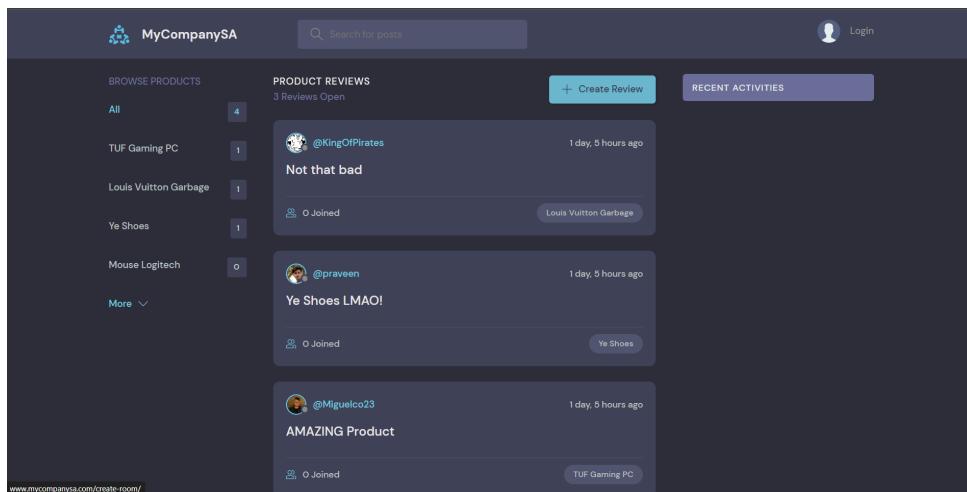
(a) Configuración del dominio y subdominio en Namecheap

3.2.3 | Configuración del aplicativo en cada dominio

Primero realizmos el cambio a www.mycompanysa.com, entonces tomamos las 3 primeras instancias que creamos, cuando accedemos realizamos lo siguiente en la terminal.

```
curl -sL https://rpm.nodesource.com/setup_16.x | sudo bash -
sudo yum install git nodejs gcc-c++ make python3 docker -y
# * Services
sudo systemctl enable docker
sudo systemctl start docker
sudo systemctl stop httpd.service
# * Project setup
git clone https://github.com/Youngermaster/ST0255-2022-2-Projects.git
cd ST0255-2022-2-Projects/project-2/MyCompanySA
sudo docker build -t mycompanysa:release .
sudo docker run -d -it -p 80:8000 mycompanysa:release
```

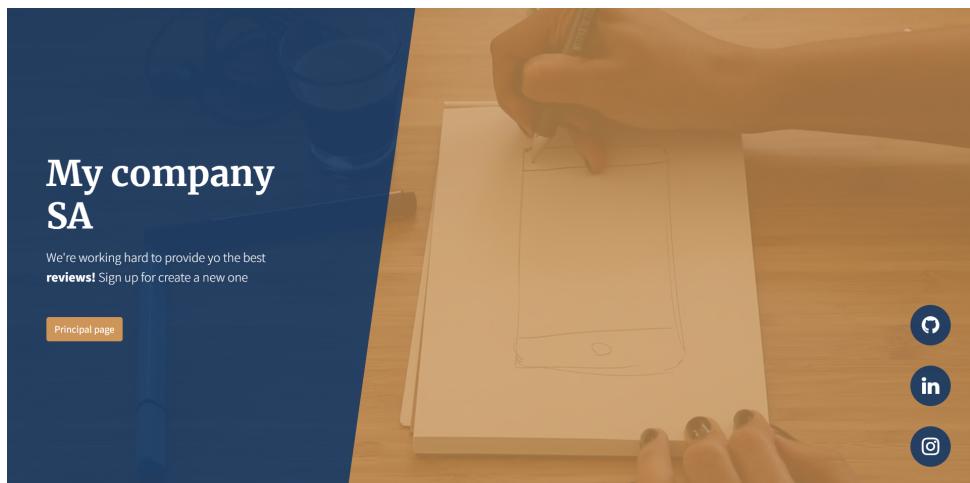
Al realizarlo en los 3 nodos si accedemos a www.mycompanysa.com esto nos devolverá el siguiente aplicativo, donde podremos escribir y leer reviews de diferentes productos:



(a) www.mycompanysa.com

Ahora si creamos una instancia de EC2 para el subdominio nsprimary.mycompanysa.com, entramos a través de SSH e ingresamos lo siguiente y obtendremos una landing page:

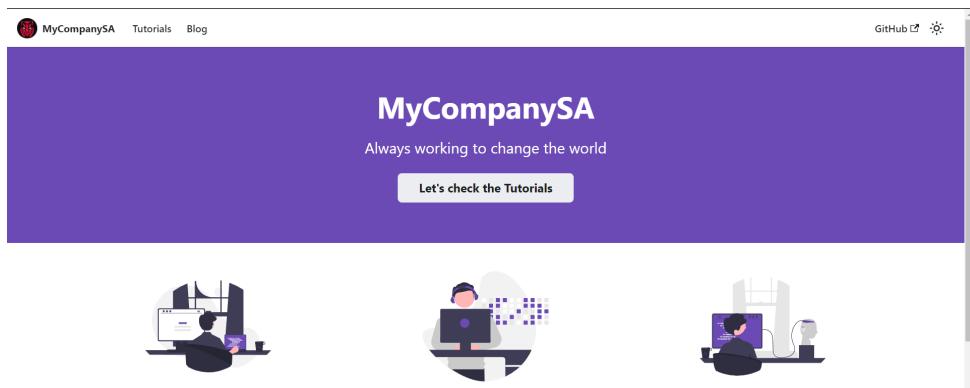
```
curl -sL https://rpm.nodesource.com/setup_16.x | sudo bash -
sudo yum install git nodejs gcc-c++ make python3 docker -y
# * Services
sudo systemctl enable docker
sudo systemctl start docker
sudo systemctl stop httpd.service
# * Project setup
git clone https://github.com/Youngermaster/ST0255-2022-2-Projects.git
cd ST0255-2022-2-Projects/project-2/nsprimary
docker build -t nsprimary-mycompanysa:release .
docker run -d -it -p 80:8000 nsprimary-mycompanysa:release
```



(a) nsprimary.mycompanysa.com

Ahora si creamos una instancia de EC2 para el subdominio nssecondary.mycompanysa.com, entramos a través de SSH e ingresamos lo siguiente y obtendremos una Blog para MyCompanySA:

```
curl -sL https://rpm.nodesource.com/setup_16.x | sudo bash -
sudo yum install git nodejs gcc-c++ make python3 docker -y
# * Services
sudo systemctl enable docker
sudo systemctl start docker
sudo systemctl stop httpd.service
# * Project setup
git clone https://github.com/Youngermaster/ST0255-2022-2-Projects.git
cd ST0255-2022-2-Projects/project-2/nssecondary
docker build -t nssecondary-mycompanysa:release .
docker run -d -it -p 80:80 nssecondary-mycompanysa:release
```



(a) nssecondary.mycompanysa.com

3.3 | Disponibilidad para miles de usuarios

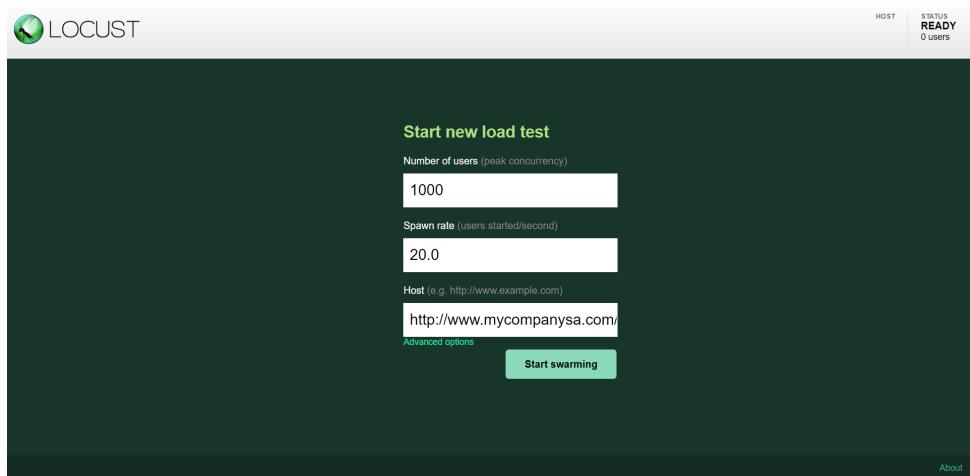
Ahora bien, como se mencionó al principio, lo más importante es una alta disponibilidad, entonces probemos 1000, 1500 y 3000 usuarios concurrentes para ver el comportamiento. Para esto usaremos [locust](#).

En nuestra máquina lanzamos lo siguiente, pero hay que estar seguros que estemos en la carpeta del proyecto.

```
cd project-1/Load-Testing
# Primero con 1000
locust -f locust.py --host http://www.mycompanysa.com/ --users 1000 --spawn-rate 20
# Luego con 1500
```

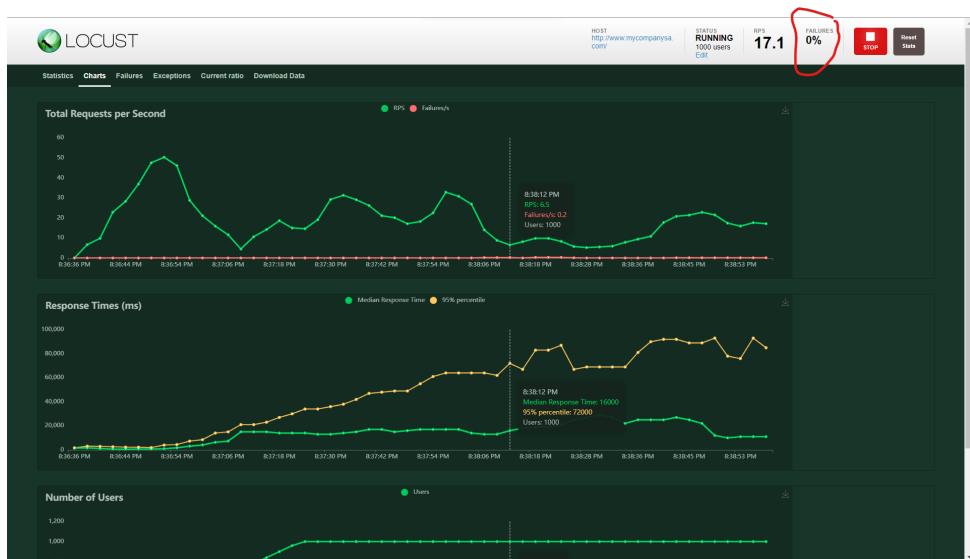
```
locust -f locust.py --host http://www.mycompanysa.com/ --users 1500 --spawn-rate 20
# Luego con 3000
locust -f locust.py --host http://www.mycompanysa.com/ --users 3000 --spawn-rate 20
```

Obtendremos la siguiente interfaz y le damos en el botón de empezar:



(a) Locust configuración

Un ejemplo de cómo se ve la interfaz es la siguiente:



(a) Locust 1000

En conclusión revisamos a fondo después de realizar las pruebas obtendremos que la mayor cantidad de usuarios soportados concurrentemente es de 1840.

3.4 | Demo

Las direcciones para probar el despliegue son:

- www.mycompanysa.com
- nsprimary.mycompanysa.com
- nssecondary.mycompanysa.com