Laboratorio.

Curso: ST0263 – Tópicos Especiales en Telemática
Título: Desplegando un Sitio Web Seguro.
Objetivo: Desplegar un sitio web basado en un CMS utilizando certificados digitales.
Duración: 40 mins.

1 Introducción.

En la actualidad, la seguridad de las aplicaciones es un factor fundamental. Para el caso del despliegue, se hace necesario garantizar protocolos que transporte la información de forma segura, garantizando así, la confidencialidad e integridad de los datos que viajan en la red.

Para efectos del despliegue de aplicaciones, entre otros, el utilizar el protocolo https, permite el despliegue de aplicaciones web seguras. Es por esta razón, que este laboratorio, desarrollaran las habilidades para el despliegue de un sitio web seguro utilizando certificados digitales, así como https.

2 Recursos

Para el desarrollo del siguiente laboratorio se dispondrán de los siguientes recursos web a través de la cuenta de AWS educate:.

- Servicio de EC2.

3 Desarrollo

A continuación, se ilustran el conjunto de pasos que se requieren para desplegar un sitio web desarrollado utilizando un CMS como wordpress y donde se asocien certificados digitales. Para este caso vamos a emplear una autoridad certificadora como lo es let's encrypt.

3.1 Instalar Requisitos.

- Se requiere que despliegue una instancia EC2 en AWS. Para efectos de este laboratorio vamos a utilizar Ubuntu 20.
- Tenga en cuenta que para esta instancia debe permitir el acceso de tráfico http (puerto 80) y https (443). Igualmente, el acceso via ssh (puerto 22). Esto debe quedar definido en el security group definido y asociado a la máquina.
- Solicite una dirección IP elástica (pública) y asóciela a la instancia creada.
- Instale docker y docker compose en la máquina e inicie el servicio.
    - Verifique la instalación con

```
$ docker --version
$ docker-compose --version
```

3.2 Configure el dominio y los registros de recursos.

Nota: Donde encuentre *.misitioseguro.ga por favor reemplácelo con el nombre de su dominio.

Solicite un dominio para el sitio web. Para efectos de este laboratorio utilizaremos el dominio mislibros.ga. Cree los diferentes registros de recursos para el dominio:

- Registros tipo A y CNAME para el nombre de la maquina.
  - misitioseguro.ga. IN A <dirIPElástica>
  - odin.misitioseguro.ga. IN A <dirIPElástica>
  - www IN CNAME odin

Por favor verifique con la herramienta network-tools (https://network-tools.com/nslookup/), que el dominio y los diferentes recursos de registros ya son visibles en Internet.

3.3 Instalar servidor web: NGINX.

Proceda a instalar ahora el servidor web NGINX.

```
$ sudo apt-get install nginx
```

Verifique que el servicio este instalado y funcionando correctamente

```
$ systemctl status nginx
```

3.4 Instalar Certbot.

```
$ sudo apt-get update
$ sudo apt-get instal certbot
$ apt-get install python3-certbot-nginx
```

3.5 Solicitar el Certificado a través de Certbot

Para solicitar el certificado a la entidad certificadora, se requiere hacer uso de un agente. Para efectos de este laboratorio, emplearemos certbot. Tenga en cuenta que cerbot que obtiene el certificado digital y lo instala en un servidor web. Este certificado lo obtiene de let's encrypt, la cual es una autoridad de certificación abierta. Tenga presente que el uso de cerbot and let's encrypt es totalmente gratis. Igualmente, certbot esta concebido para correr en el servidor que va a soportar la ejecución del servidor web. El cliente certbot utiliza un challenge http-01 (puerto 80) con el fin de validar que realmente se tenga control sobre el dominio para el cual se desea establecer el certificado.

Para esto ejecute el siguiente comando:

$ sudo certbot --nginx -certonly -d misitioseguro.ga -d www.misitioseguro.ga

Una vez ejecute el comando, proporcione una dirección de correo electrónico y acepte los términos (A). A continuación, observará que el agente de certbot realiza los retos para verificar que tiene autoridad sobre el dominio (los realiza via http). Finalmente, debe observar algo parecido en pantalla como lo muestra la figura.

En este punto, los certificados fueron generados exitosamente y se encuentran ubicados en la ruta /etc/letsencrypt/.

Se requiere que cree el siguiente directorio:

$ sudo mkdir /home/ubuntu/wordpress
$ sudo mkdir /home/ubuntu/wordpress/ssl

Ahora, debemos efectuar la copia de los certificados como superusuario,

$ sudo -s
$ cp /etc/letsencrypt/live/misitioseguro.ga/* /home/ubuntu/wordpress/ssl/
$ cp /etc/letsencrypt/options-ssl-nginx.conf /home/ubuntu/wordpress/ssl/
$ cp /etc/letsencrypt/ssl-dhparams.pem /home/ubuntu/wordpress/ssl/
$exit

3.6 Docker compose

Ahora vamos a crear el docker compose para desplegar nuestro sitio web. Este va a utilizar un stack conformado por Linux, NGINX, MySQL y PHP (LEMP).

En este directorio vamos a crear un archivo denominado docker-compose.yml. Recuerde que docker-compose es una herramienta que nos permite definir y ejecutar múltiples contenedores. Los contenedores que se van a considerar son para nginx, wordpress y mysql.

$ sudo nano docker-compose.yml

Copie el siguiente texto:

```
version: '3.1'
services:
  nginx:
    container_name: nginx
    image: nginx
    volumes:
    - ./nginx.conf:/etc/nginx/nginx.conf:ro
    - ./ssl:/etc/nginx/ssl
    ports:
    - 80:80
    - 443:443
    depends_on:
    - wordpress
  wordpress:
    container_name: wordpress
    image: wordpress
    restart: always
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html
  db:
    image: mysql:5.7
    restart: always
```

```
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql
volumes:
  wordpress:
  db:
```

3.7 Archivo de Configuración para NGINX.

Ahora vamos a crear el archivo de configuración para el servidor nginx.

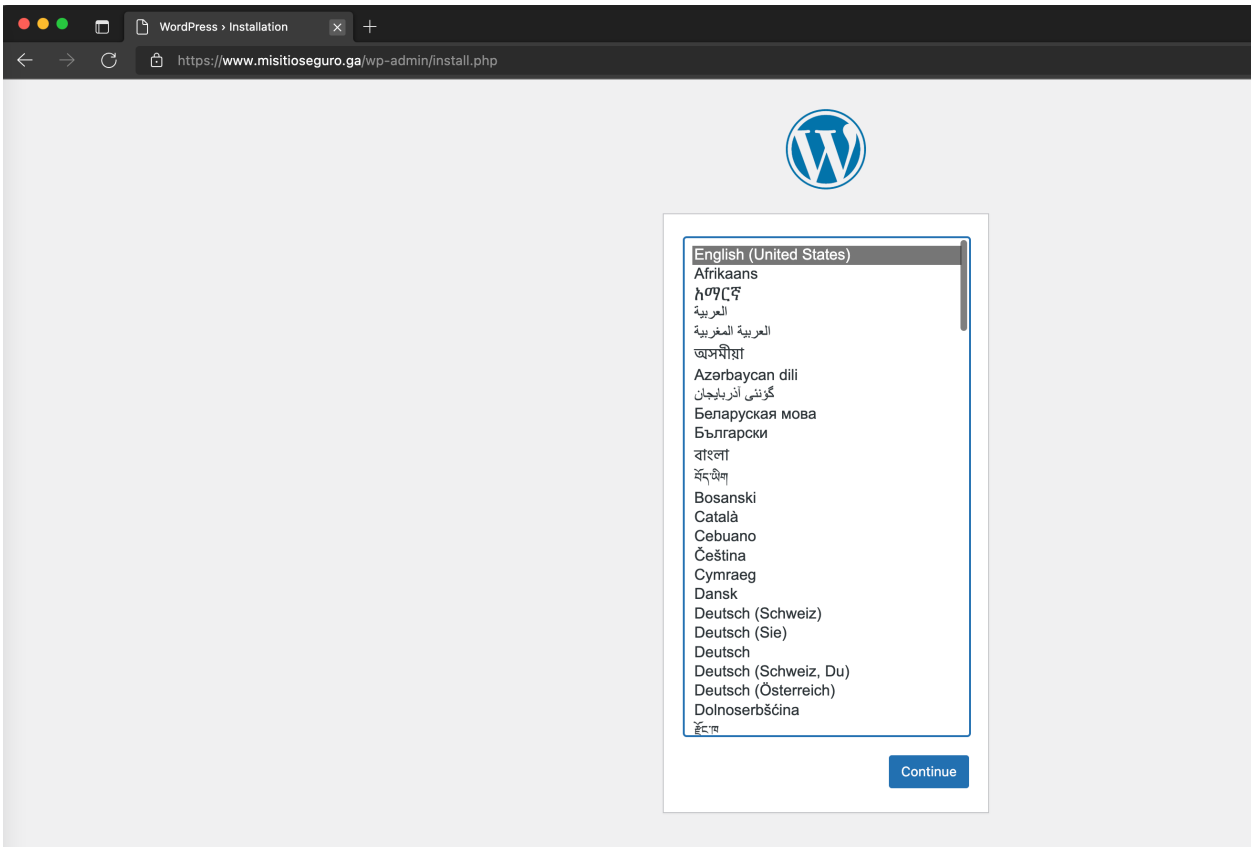
$ sudo nano nginx.conf

Copie el siguiente texto:

```
worker_processes auto;
pid /run/nginx.pid;
error_log /var/log/nginx/error.log;

events {
    worker_connections 768;
}

http{

    server {

        listen [::]:443 ssl ipv6only=on;
        listen 443 ssl;
        server_name misitioseguro.ga www.misitioseguro.ga;

        ssl_certificate /etc/nginx/ssl/fullchain.pem;
        ssl_certificate_key /etc/nginx/ssl/privkey.pem;
        ssl_dhparam /etc/nginx/ssl/ssl-dhparams.pem;

        location / {
            proxy_pass http://wordpress:80;
            proxy_redirect off;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Host $host;
            proxy_set_header X-Forwarded-Server $host;
```

```
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }

    server {
        if ($host = www.misitioseguro.ga) {
            return 301 https://$host$request_uri;
        }

        if ($host = misitioseguro.ga) {
            return 301 https://$host$request_uri;
        }

        listen 80 ;
        listen [::]:80 ;
        server_name misitioseguro.ga www.misitioseguro.ga;

        return 404;

    }

}
```

Ahora verifiquemos el funcionamiento del sitio web vía https, para esto digite https://www.misitioseguro.ga. Debe ver la siguiente página de configuración de wordpress:


WordPress › Installation
https://www.misitioseguro.ga/wp-admin/install.php
English (United States)
Afrikaans
አማርኛ
العربية
العربية المغربية
অসমীয়া
Azərbaycan dili
گؤنئی آذربایجان
Беларуская мова
Български
বাংলা
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch (Sie)
Deutsch
Deutsch (Schweiz, Du)
Deutsch (Österreich)
Dolnoserbšćina
Continue