## Postwork Sesión X

### 1. Objetivo 🎯

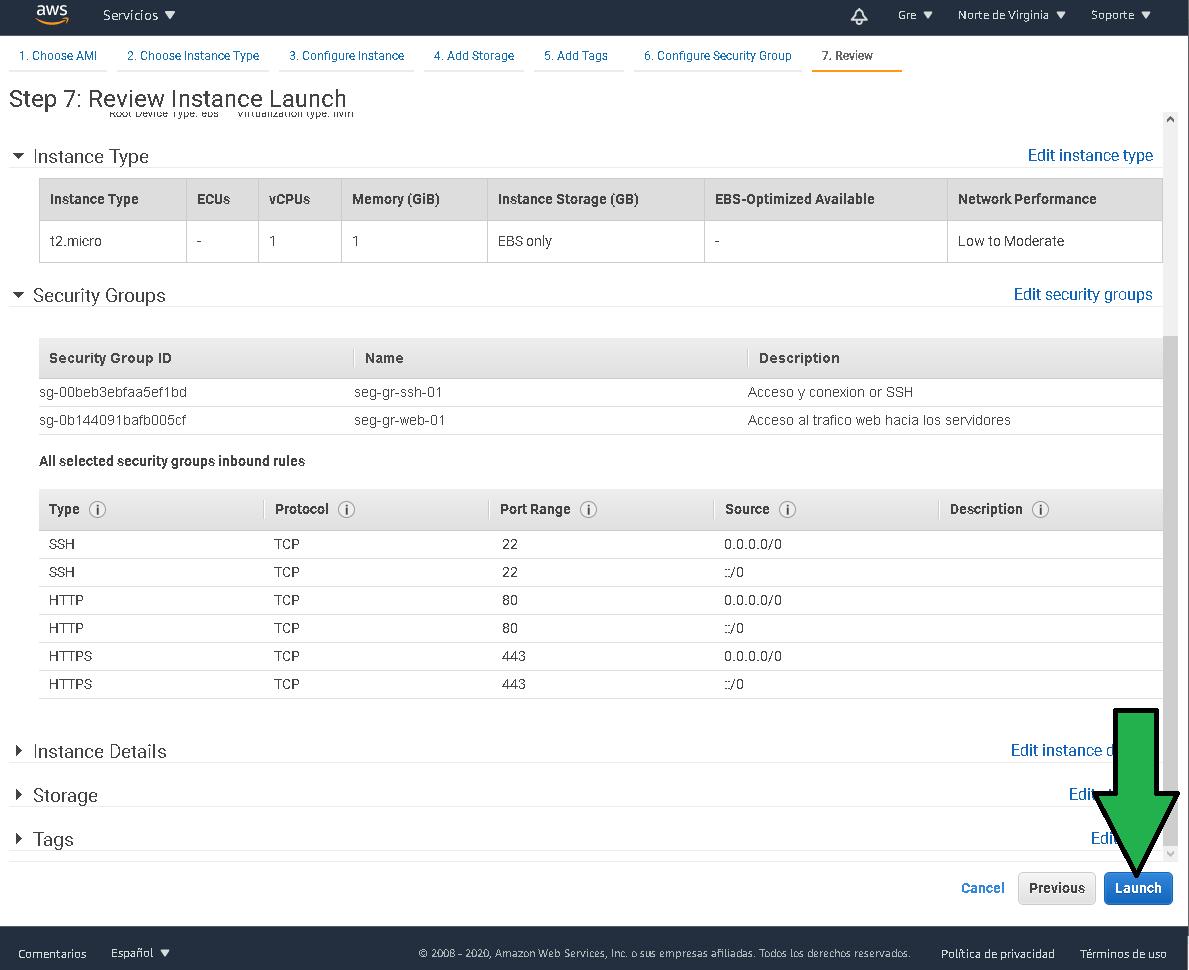
* Conocer el bastión básico para la ejecución de cómputo en la nube con EC2, dentro de las instancias EC2 se ejecutará un contenedor de Docker, el contenedor obtendrá los datos para guardarlos en base de datos relacional.

### 2. Requisitos :clipboard:

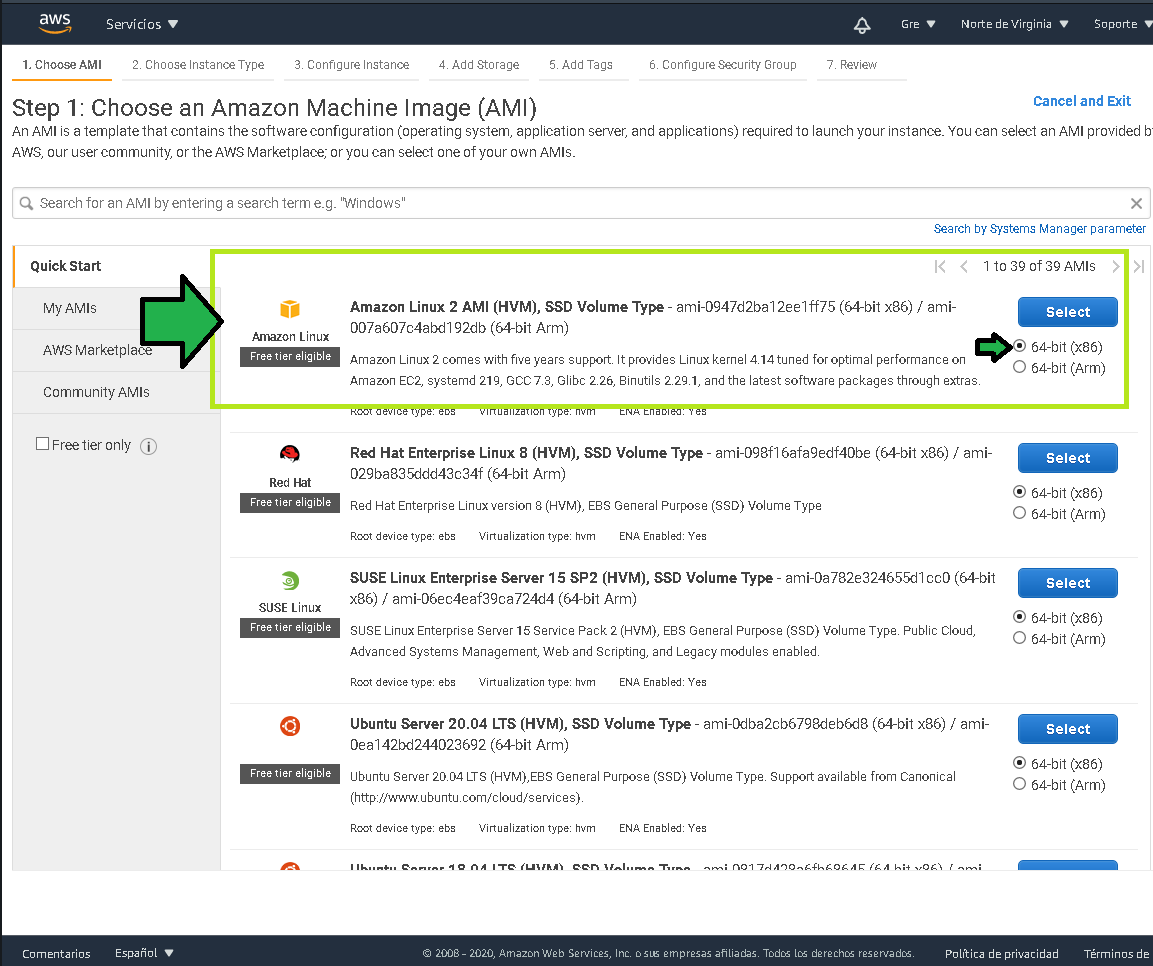
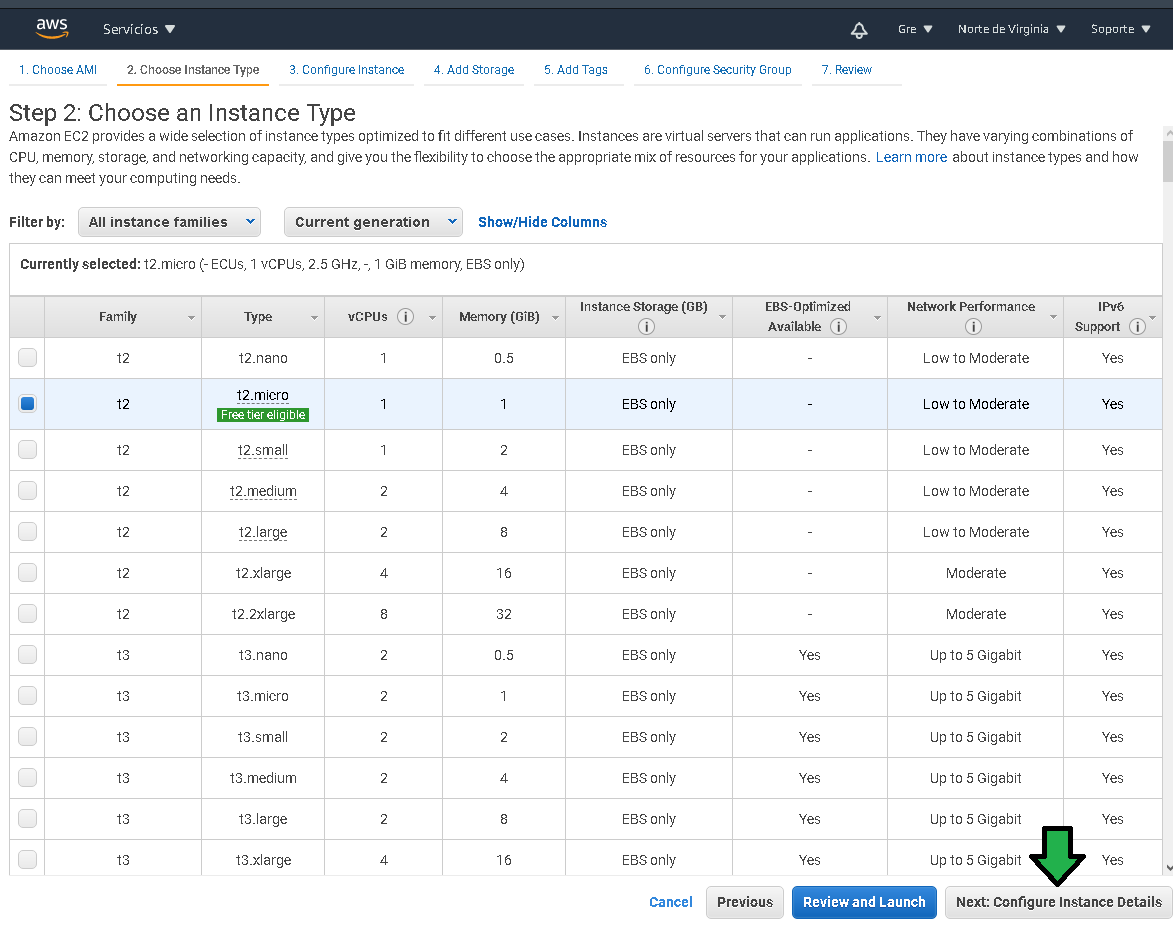
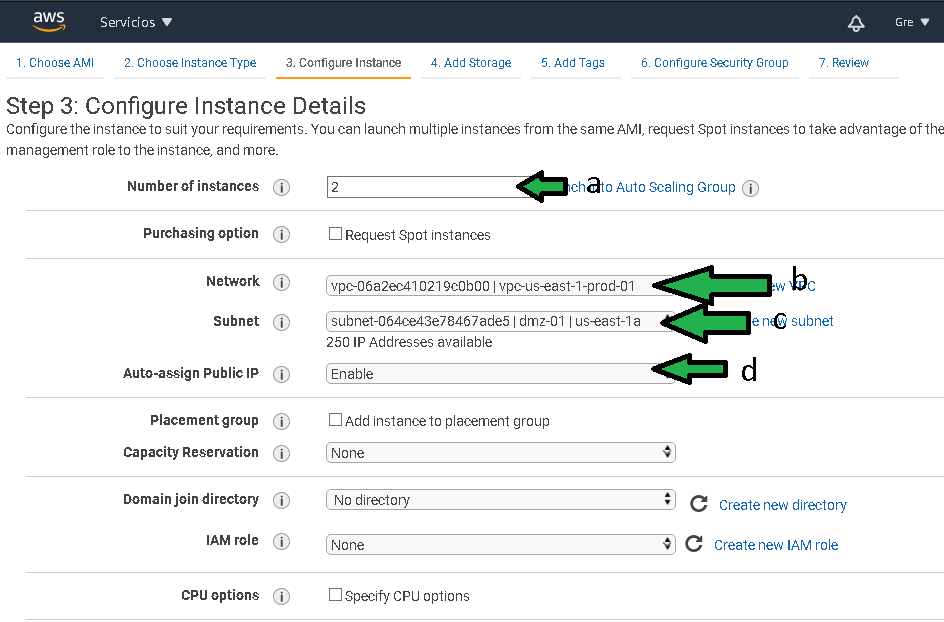
* Acceso a la consola de AWS (log)
* Una base de datos RDS generada, tener contraseña, usuario, url (Endpoint ) de la instancia.
* Tener grupos de seguridad de tráfico de entrada a puertos 22, 80, 443, 5432.
* Certificado de seguridad en Amazon Certificade Manager.
* [Postman](https://www.postman.com/product/rest-client/) instalado para verificar el funcionamiento de la API.

### 3. Desarrollo 📑

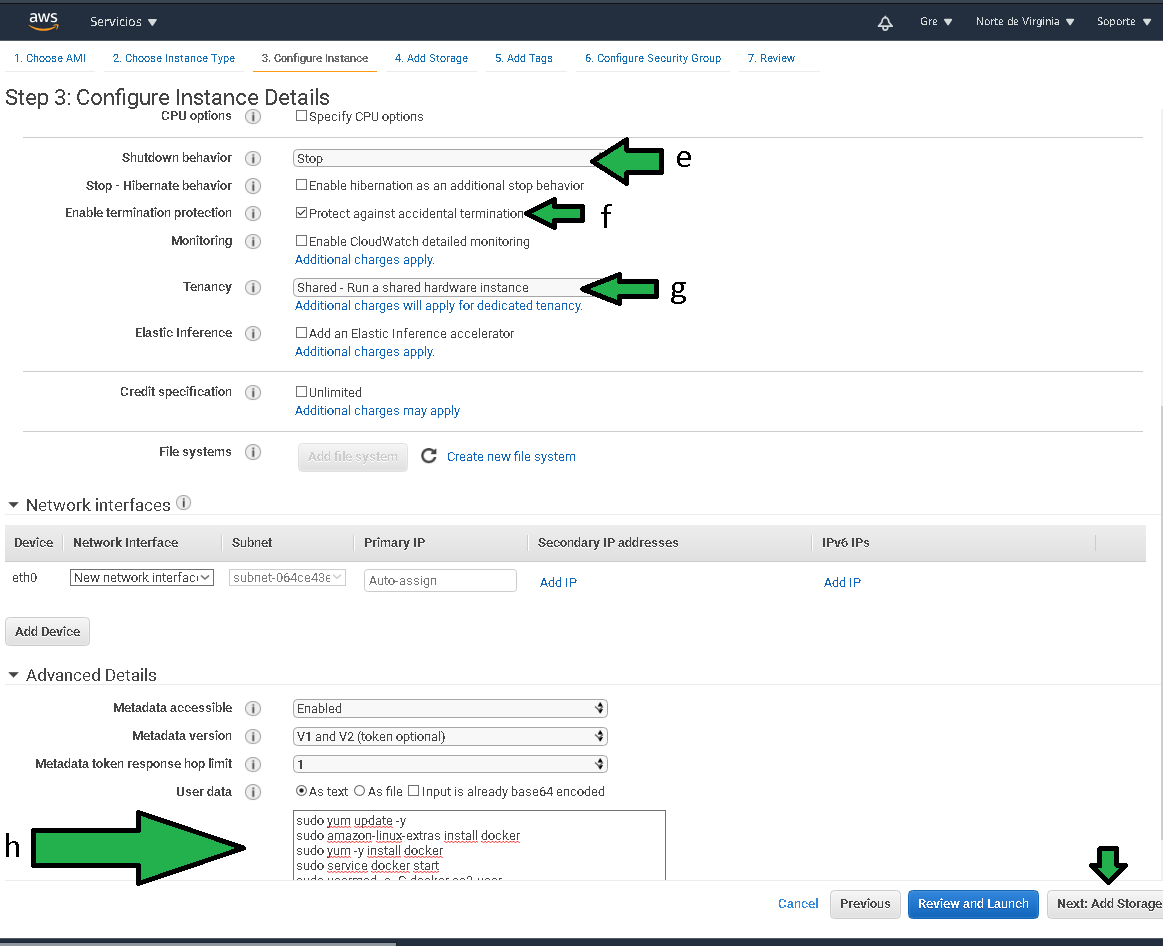
1. En el panel de EC2, dar click en “Lanzar la instancia”.



pw-launch-instance.png

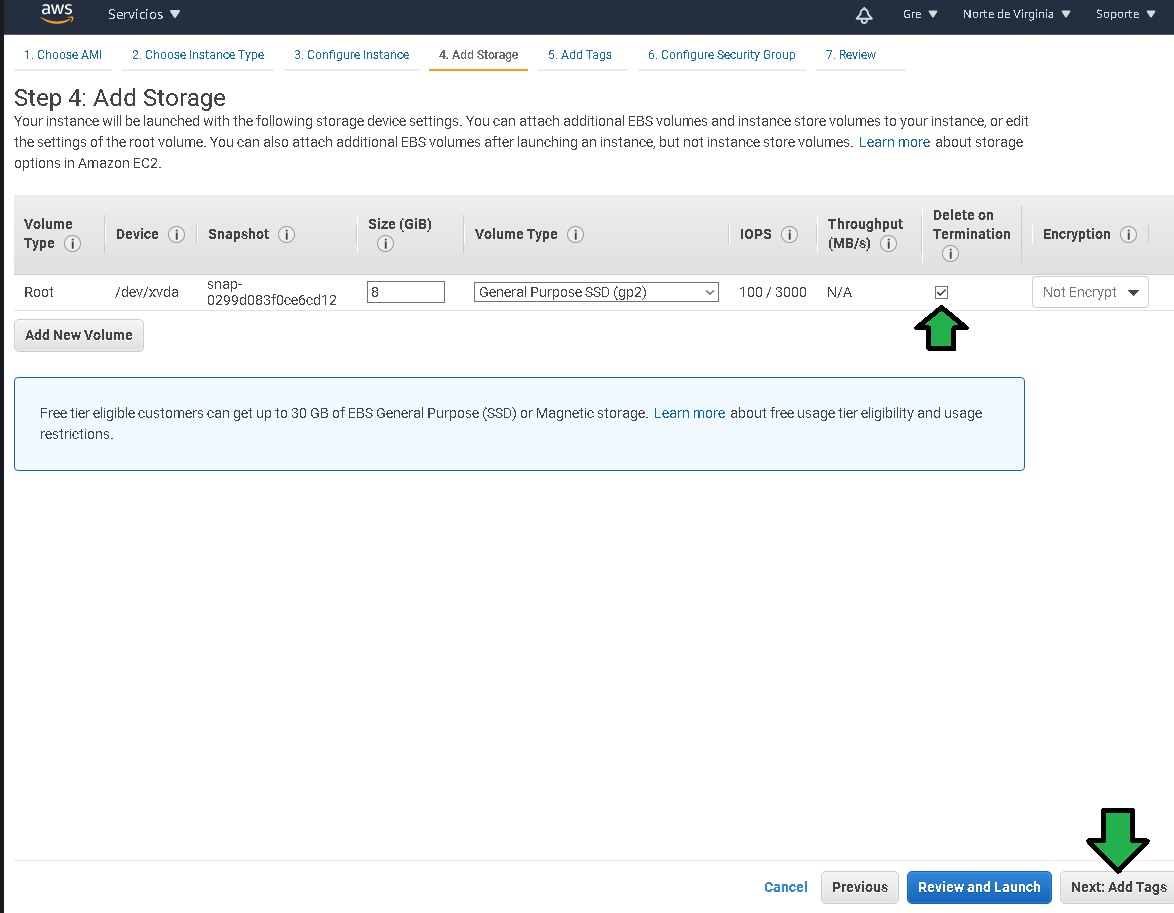
1. Seleccionar la siguiente AMI. 
2. Seleccionar el tamaño de la instancia, para este ejercicio t2.micro esta bien. 
3. Configurar la instancia como:
4. Establecer el número de instancias en **2**.
5. Seleccionar la VPC con la que se ha venido trabajando.
6. Seleccionar una de las subredes públicas.
7. Establecer asignación de IP pública al momento de generarse la instancia. 
8. Establecer el comportamiento del apagado de la instancia como “Stop”
9. Habilitar la protección contra borrado accidental (se recomienda siempre habilitarla).
10. Establecer la ejecución en hardware compartido.
11. Establecer comandos que se deben ejecutar al momento de crear la instancia, copiar y pegar los siguientes comandos (debe ir desde el # hasta la última palabra).

#!/bin/bash  
sudo yum update -y  
sudo amazon-linux-extras install -y docker  
sudo yum install -y postgresql  
sudo service docker start  
sudo usermod -a -G docker ec2-user  
sudo systemctl enable docker

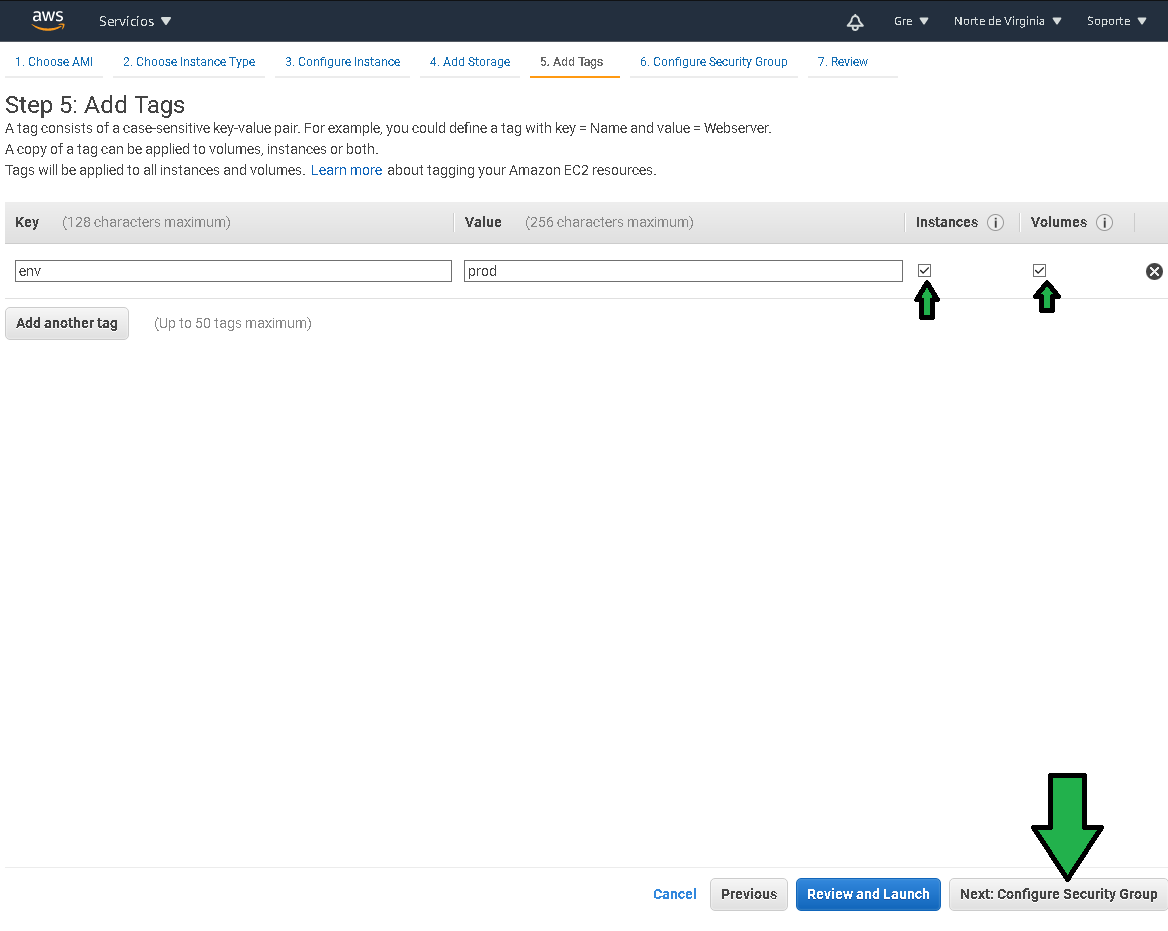


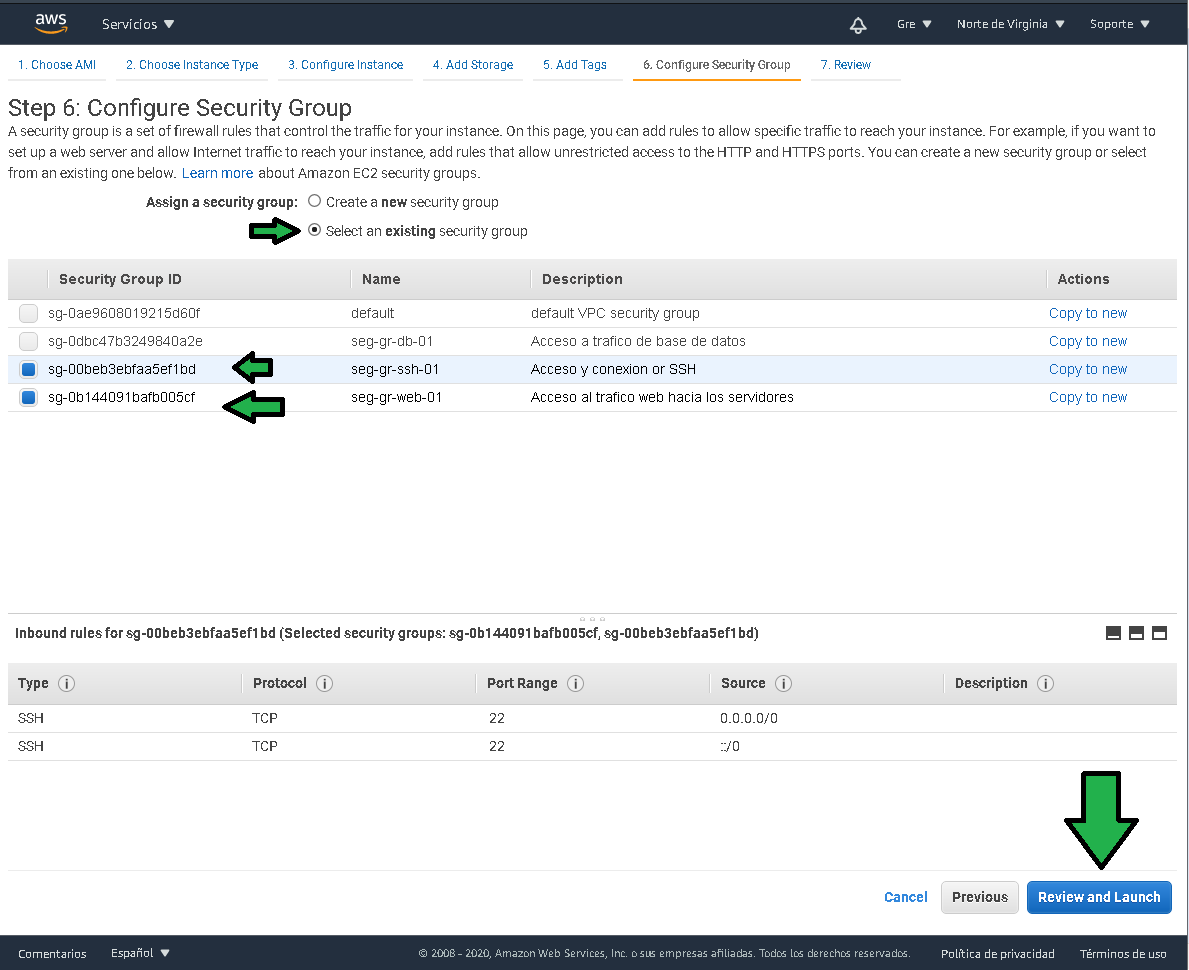
pw-configure-instance-02.png

1. Establecer el storage que tendrán las instancias, habilitar el borrado en terminación, al momento de eliminar la instancia también se eliminará el volumen.

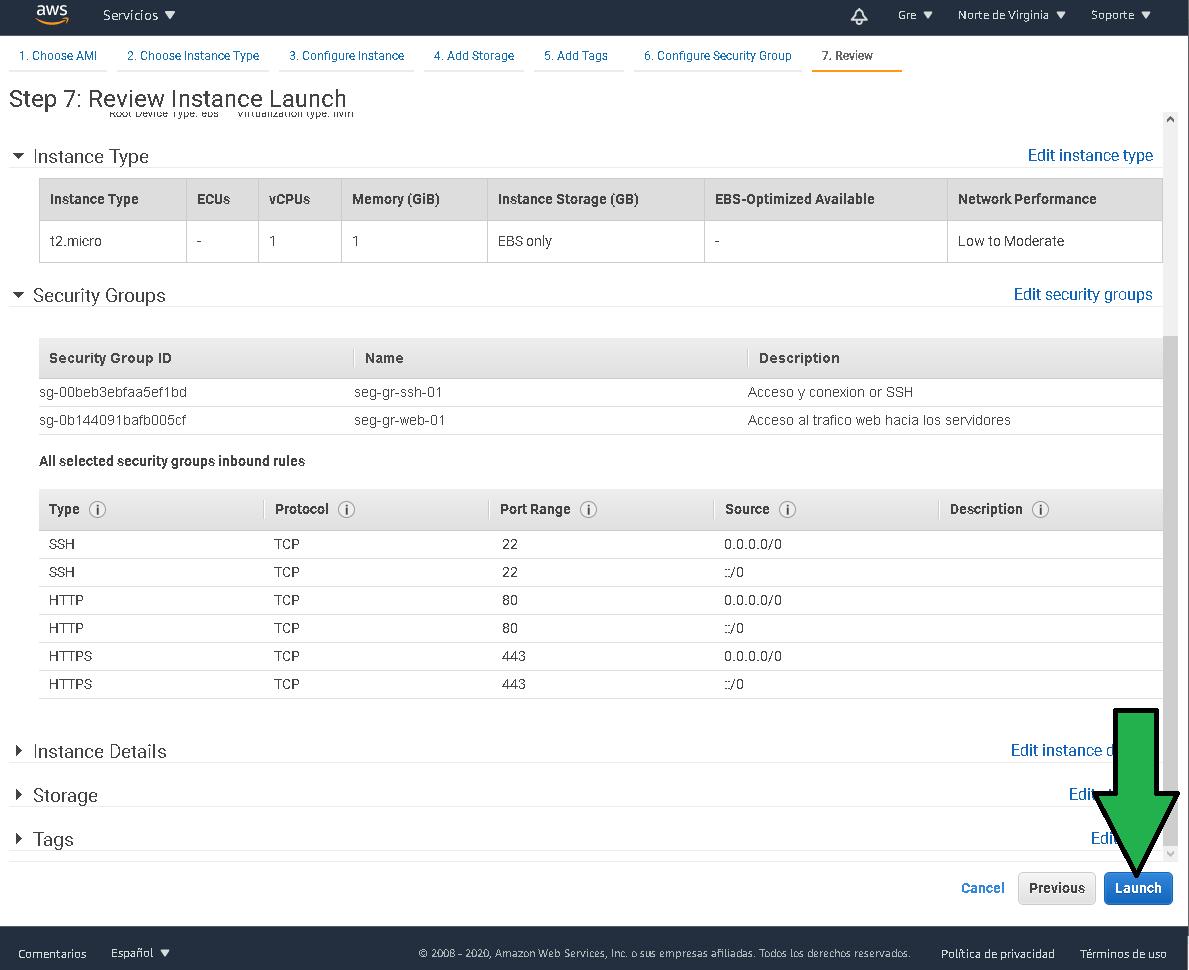


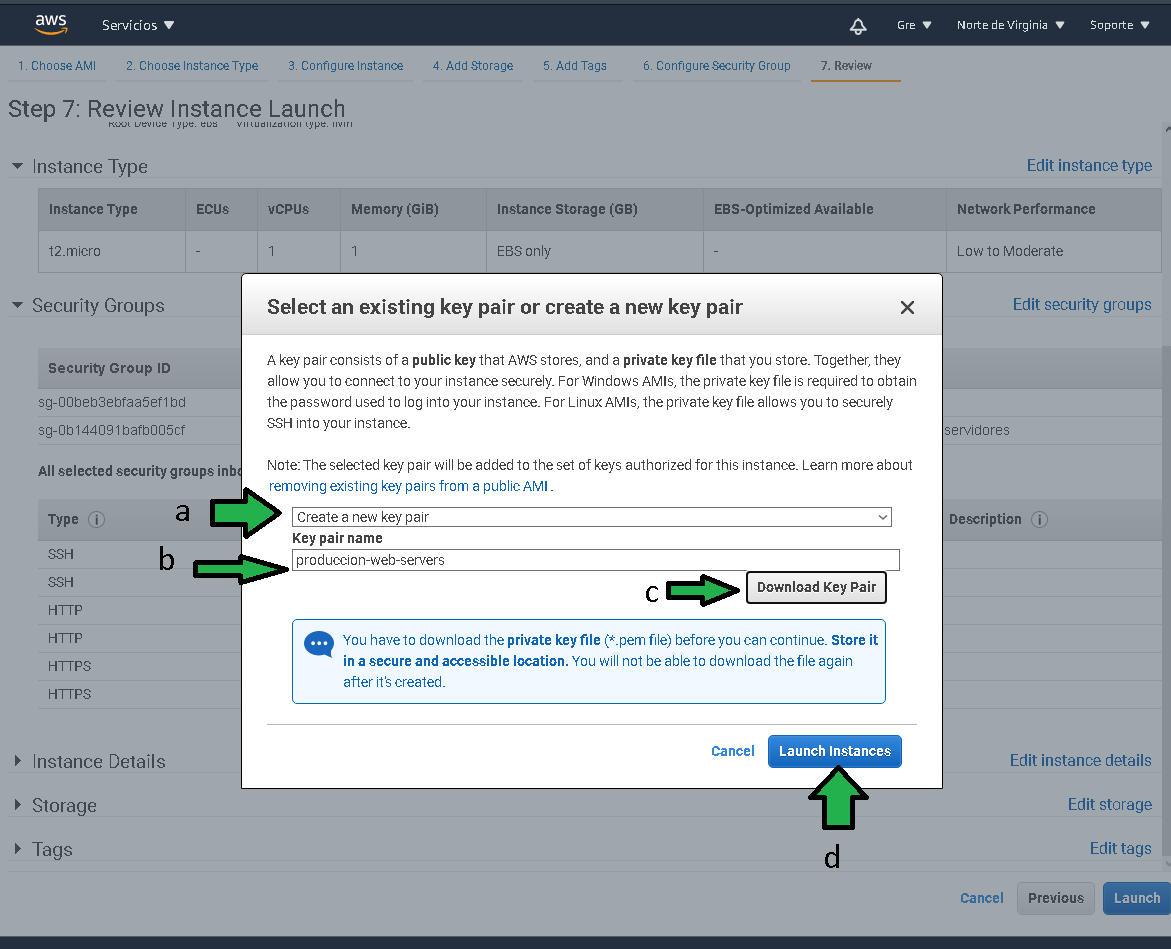
pw-configure-storage-01.png

1. Establecer Tags para facilidad de administración. 
2. Establecer los grupos de seguridad (firewall a nivel de instancia) definiendo el tráfico permitido a las instancias. En este caso SSH, HTTP y HTTPS.



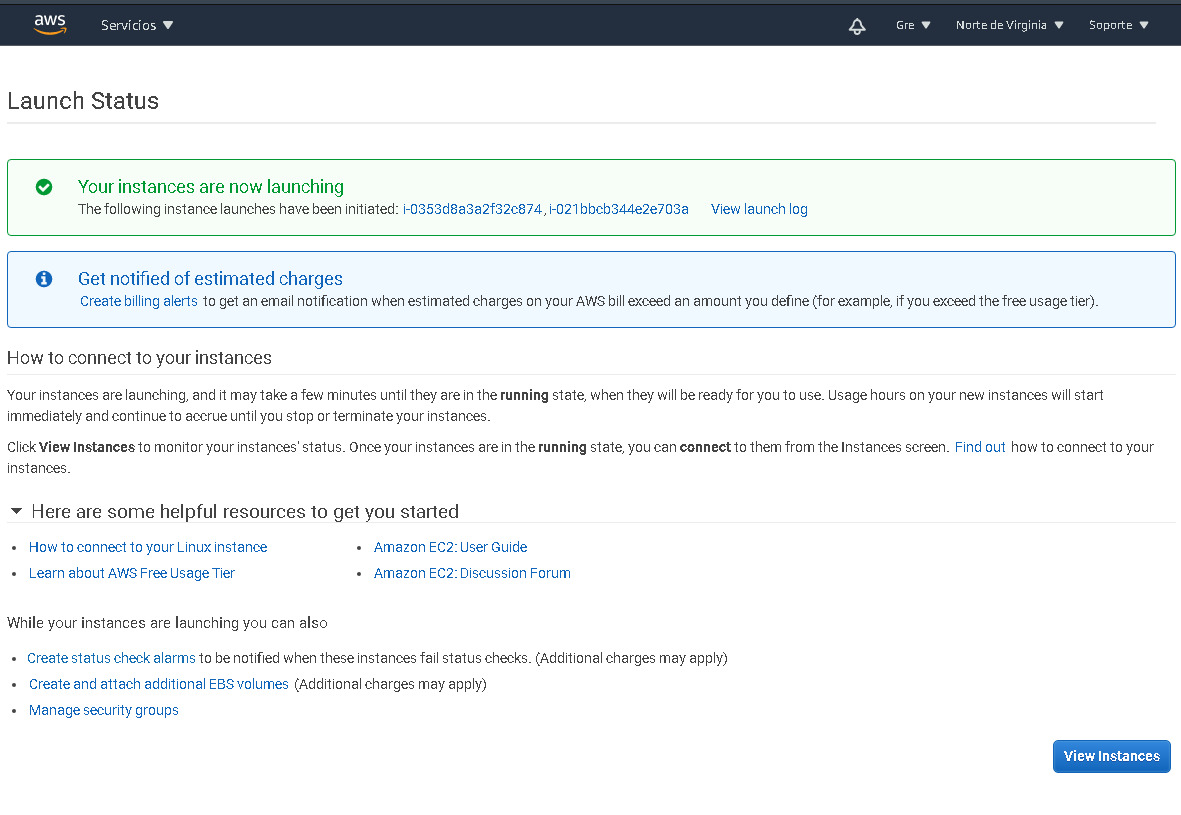
pw-security-group-01.png

1. Verificar las configuraciones y lanzar las instancias. 
2. Generar la llave de conexión nueva (a), establecer el nombre de la llave (b), descargar la llave (c) sin la llave no se podrá conectar a la instancia por SSH, finalizar en “Launch Instances” (d).



pw-launch-instance-01.png

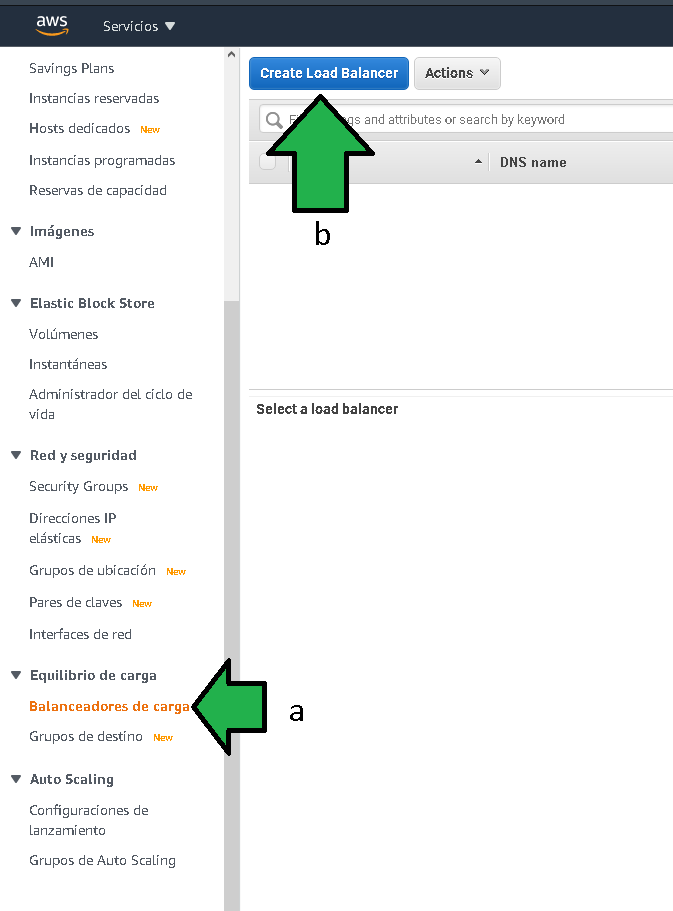
Después de algunos momentos las instancias son generadas.



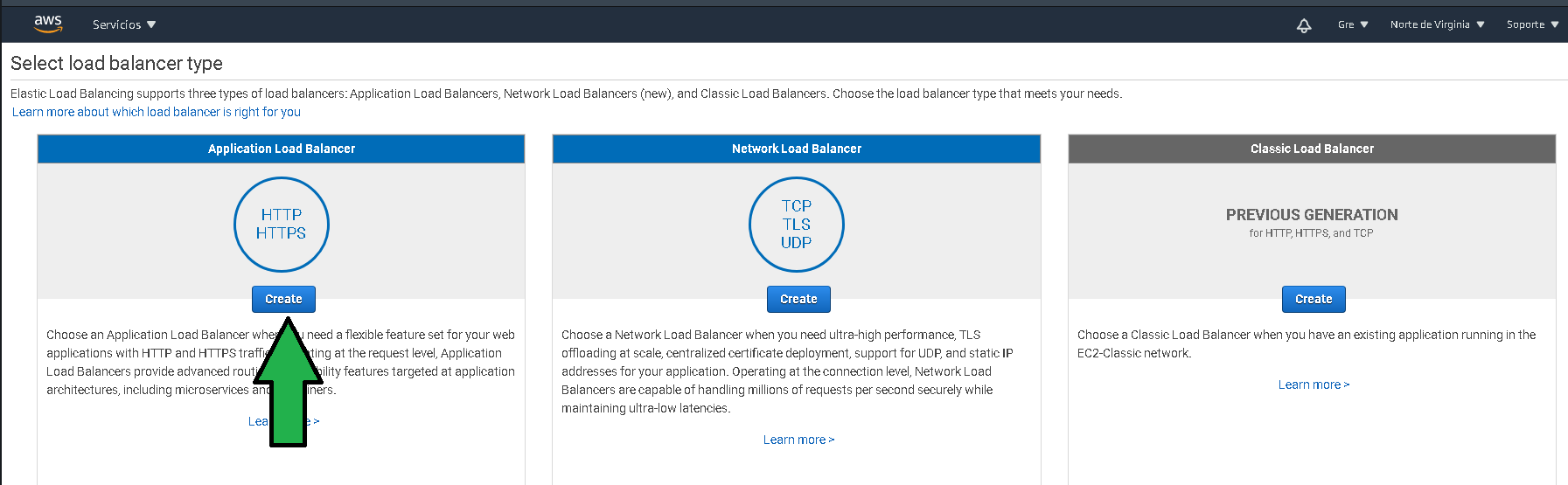
pw-launching-instances-01.png

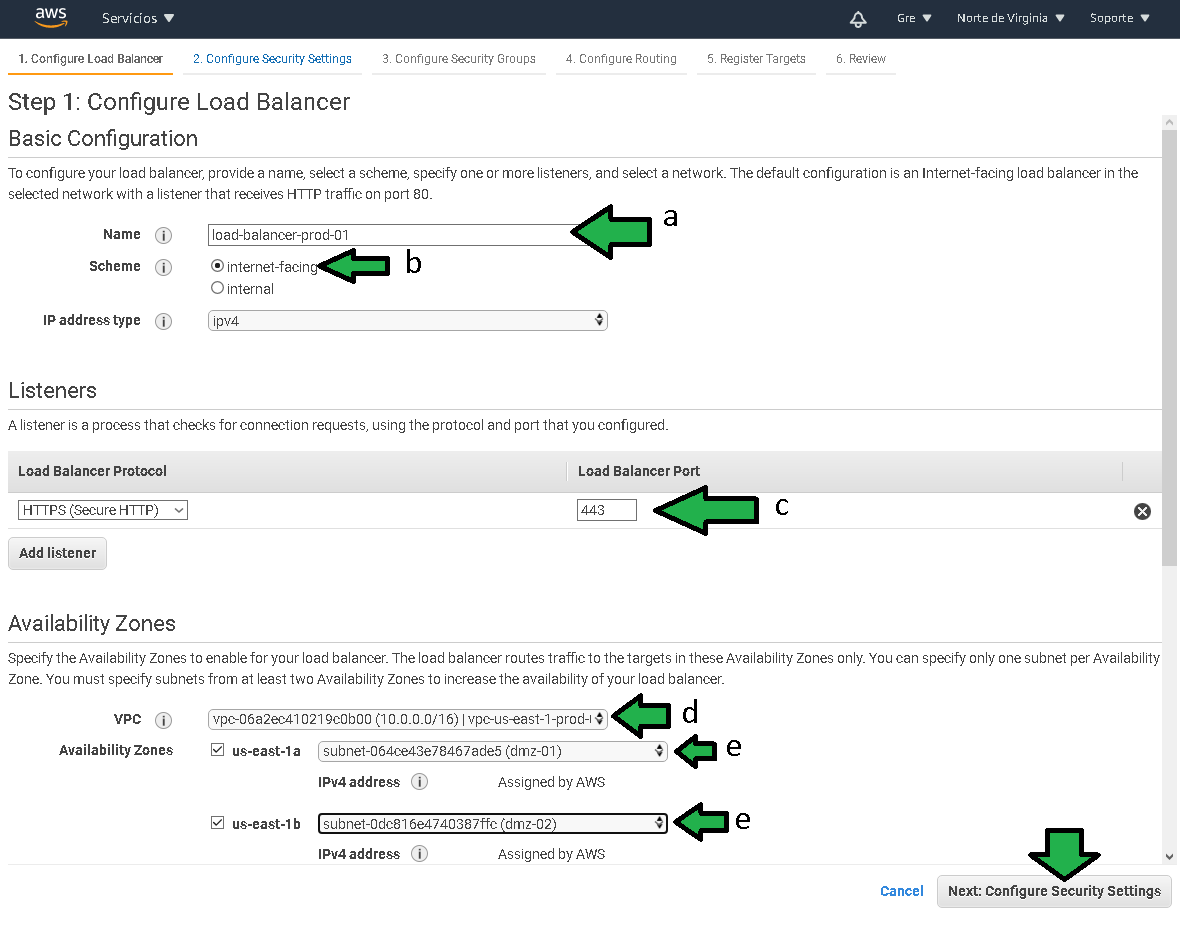
Toca dar de alta el balanceador de carga, se encargará de despachar las solicitudes a cada una de las instancias lanzadas.

1. En el panel del servicio EC2, seleccionar “Balanceadores de carga” (a), después “Create Load Balancer” (b).



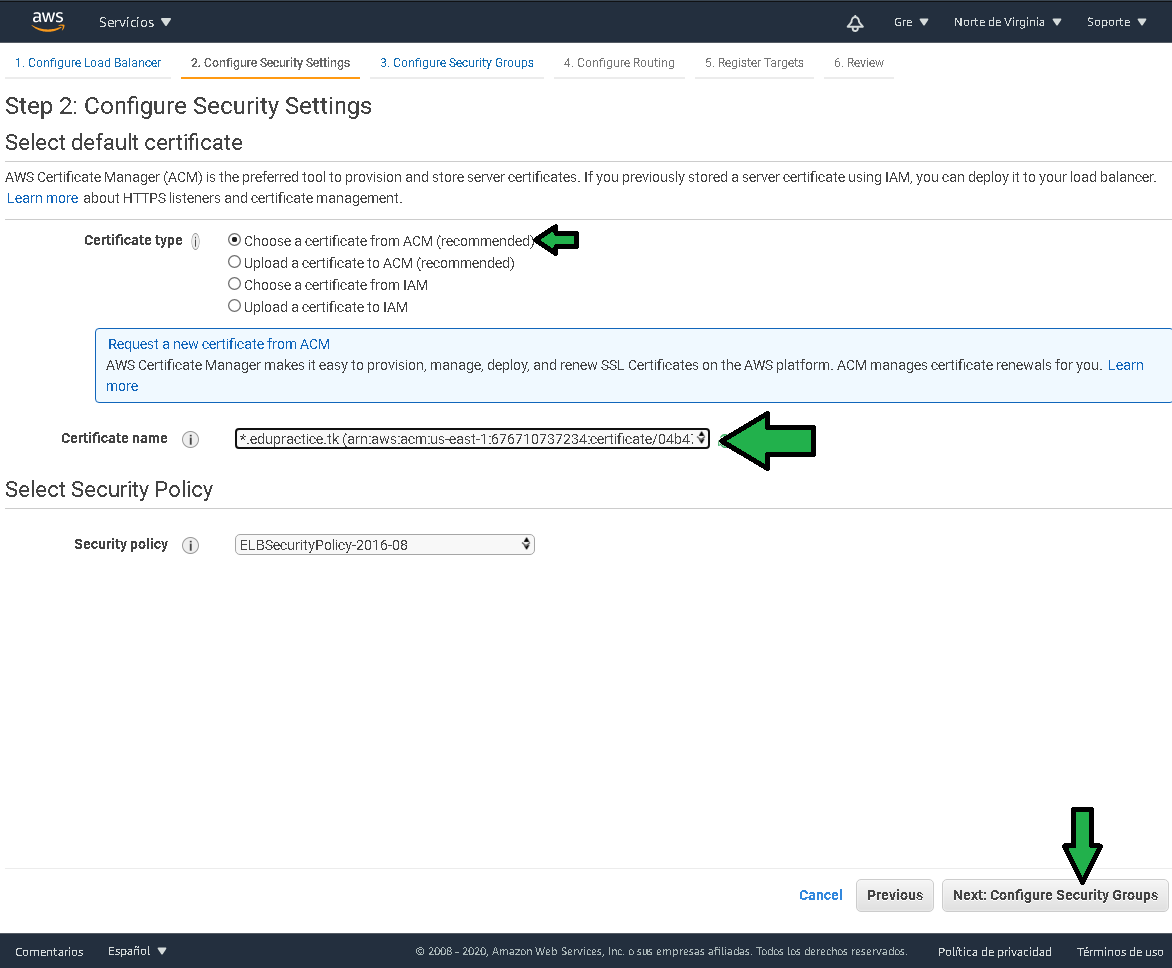
pw-create-balancer-01.png

1. Seleccionar el balanceador de carga de tipo “Application” 
2. Configurar el balanceador con:
3. Establecer un balanceador de carga.
4. Establecer un balanceador de carga de cara hacia internet
5. Seleccionar el tráfico https
6. Seleccionar la red VPC con la que se ha venido trabajando
7. Seleccionar las subredes públicas de cara a internet.

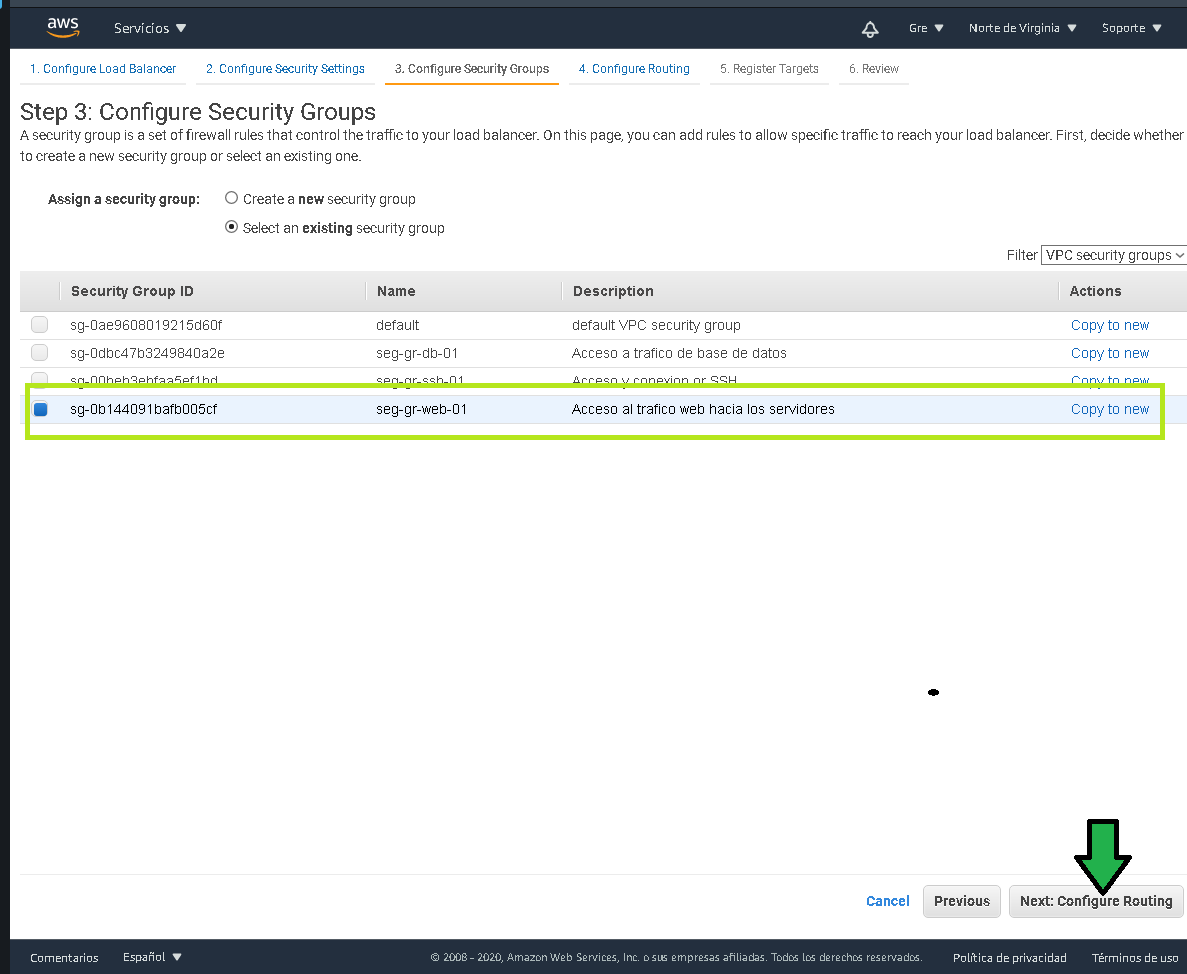


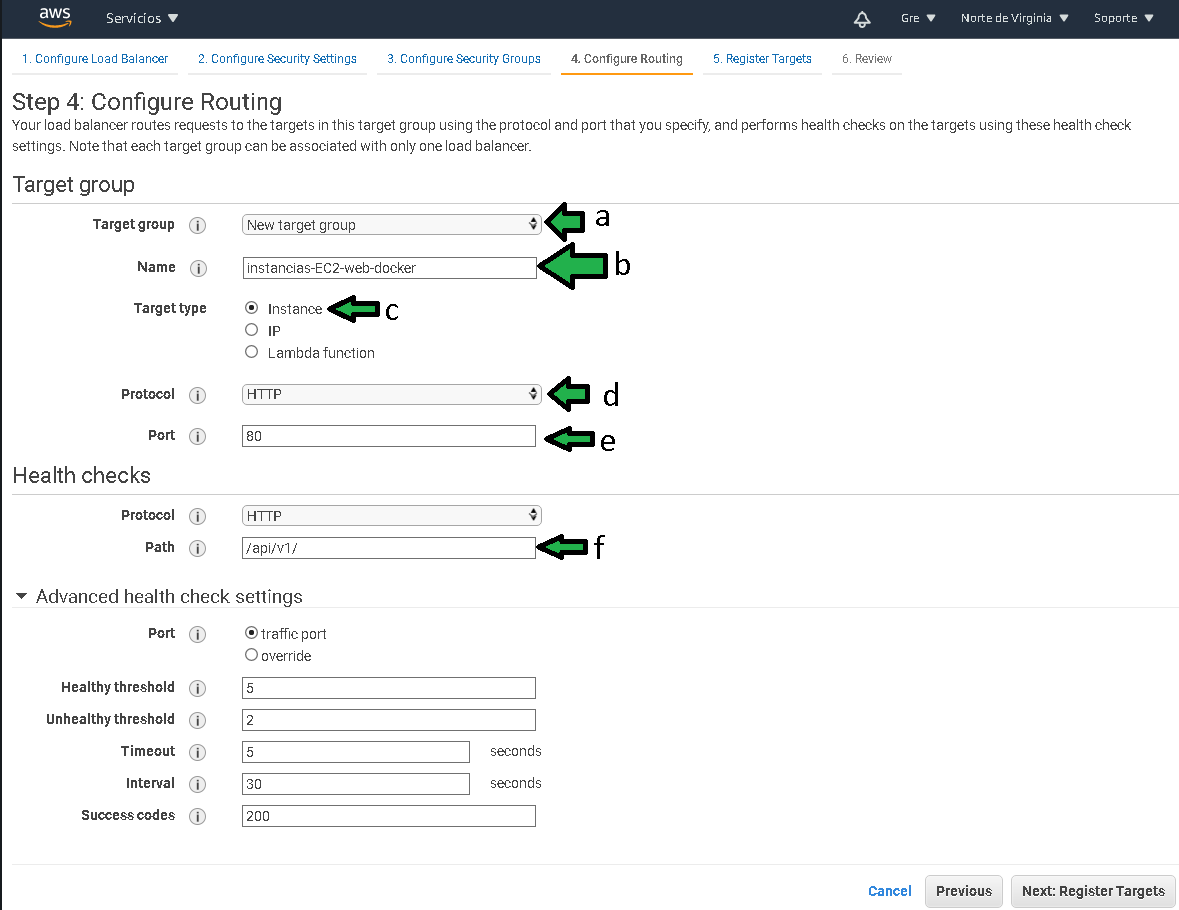
pw-load-balancer-configure-01.png

1. Seleccionar el certificado de seguridad.

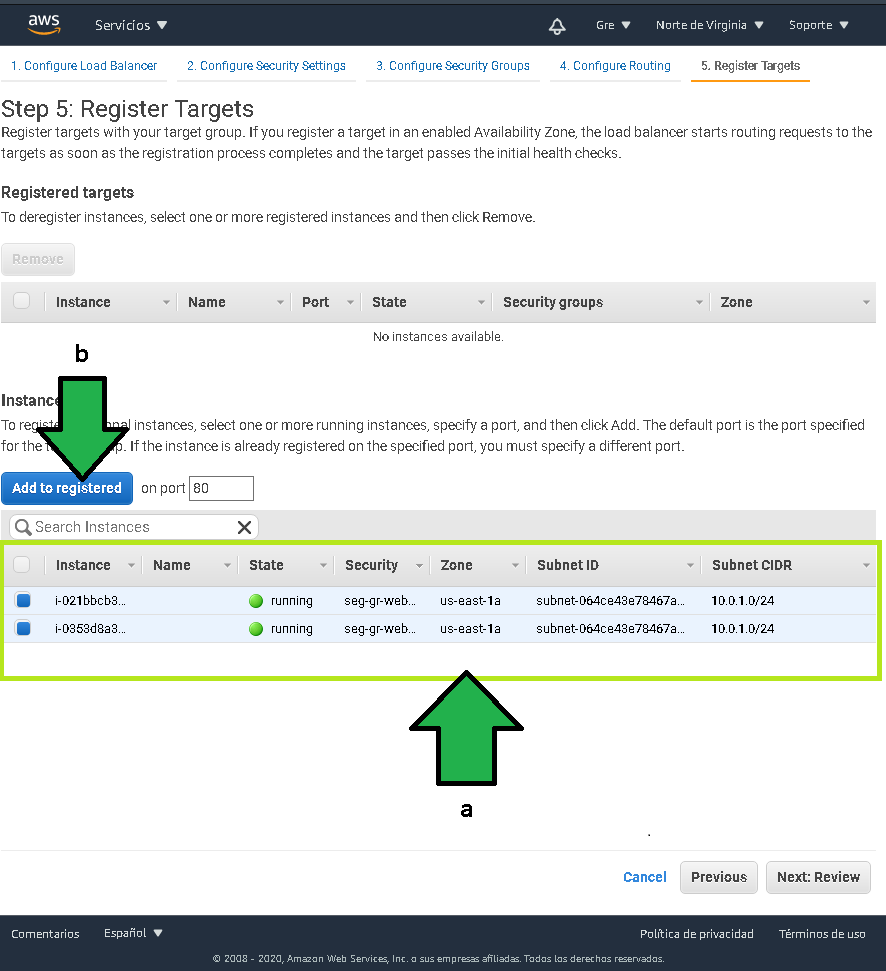


pw-add-certificate-manager-01.png

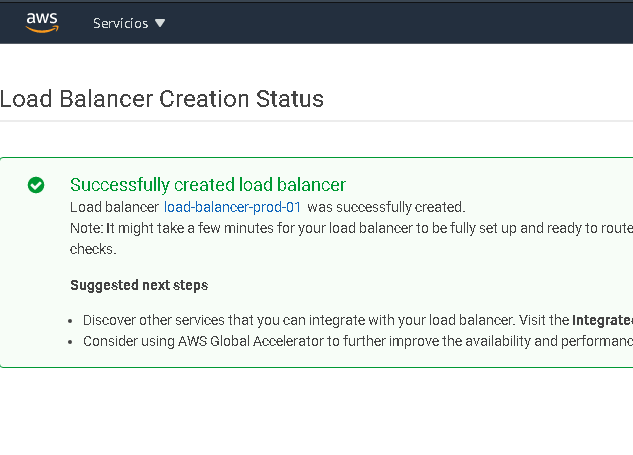
1. Seleccionar el permiso para tráfico https. 
2. Establecer el “target group”, básicamente con esto se le indica al balanceador de carga cuales son las instancias EC2 a las que debe ser redirigido el tráfico.

a), b) Especificar un nuevo target, dar un nombre específico al target para fácil administración. c) especificar que se darán de alta en el grupo instancias. d), e) Especificar protocolo y puerto al cual será redirigido el tráfico hacia las instancias. f) Para este caso especial por el tipo de programa que se usará, especificar el protocolo http y la ruta /api/v1/ como path para verificar que el servicio se este ejecutando. 

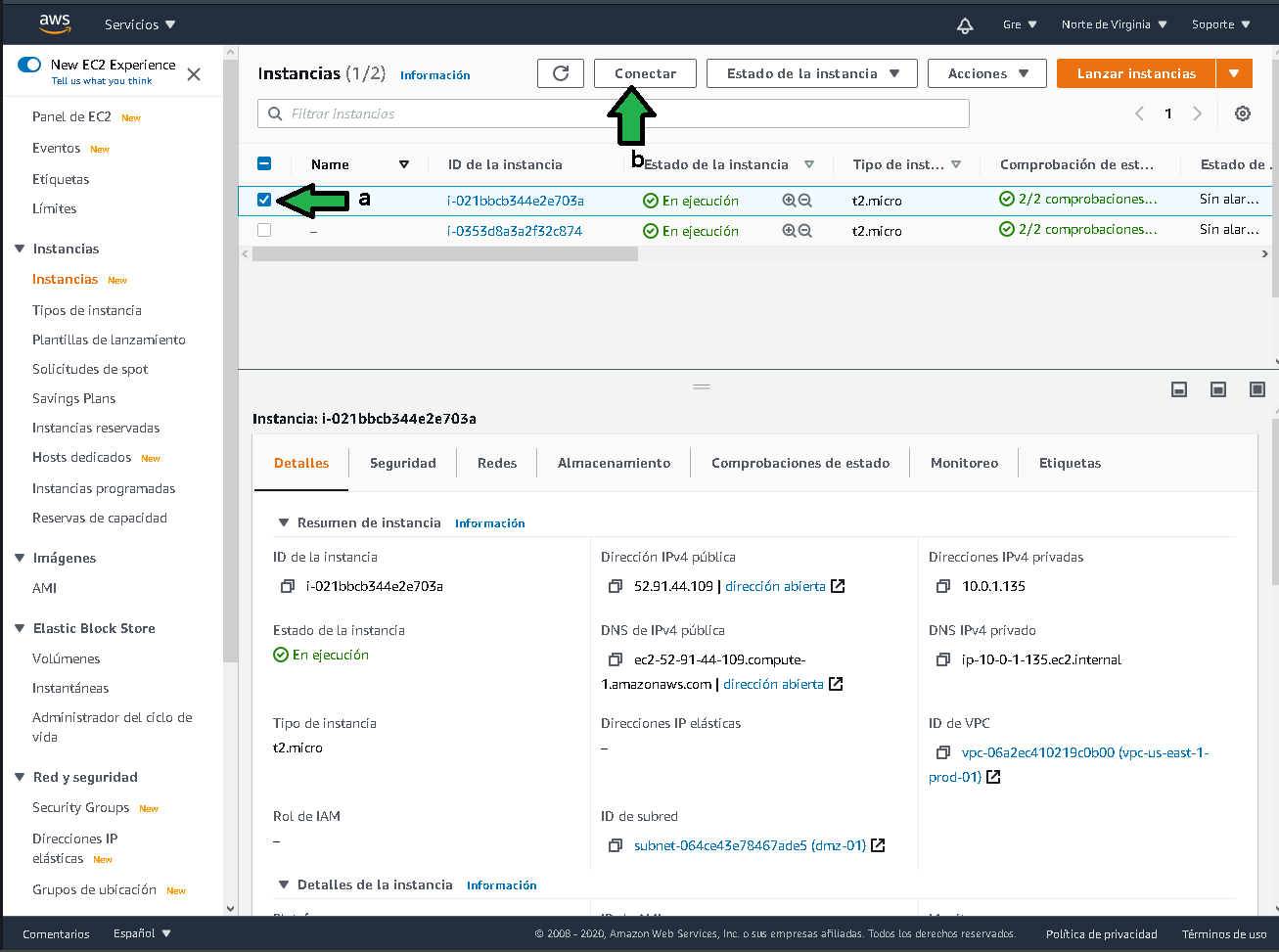
1. Agregar las instancias que se agregaran al target group para redirigir el tráfico web.
2. Seleccionar las instancias.
3. Agregar las instancias al balanceador.



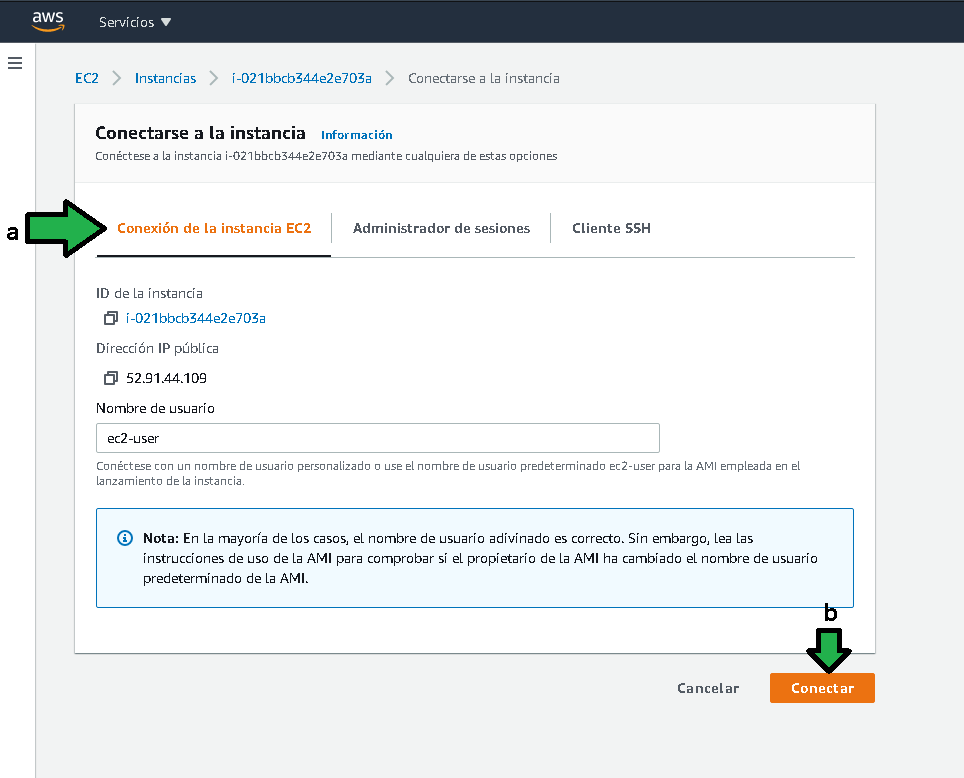
pw-add-instances-to-target-group.png

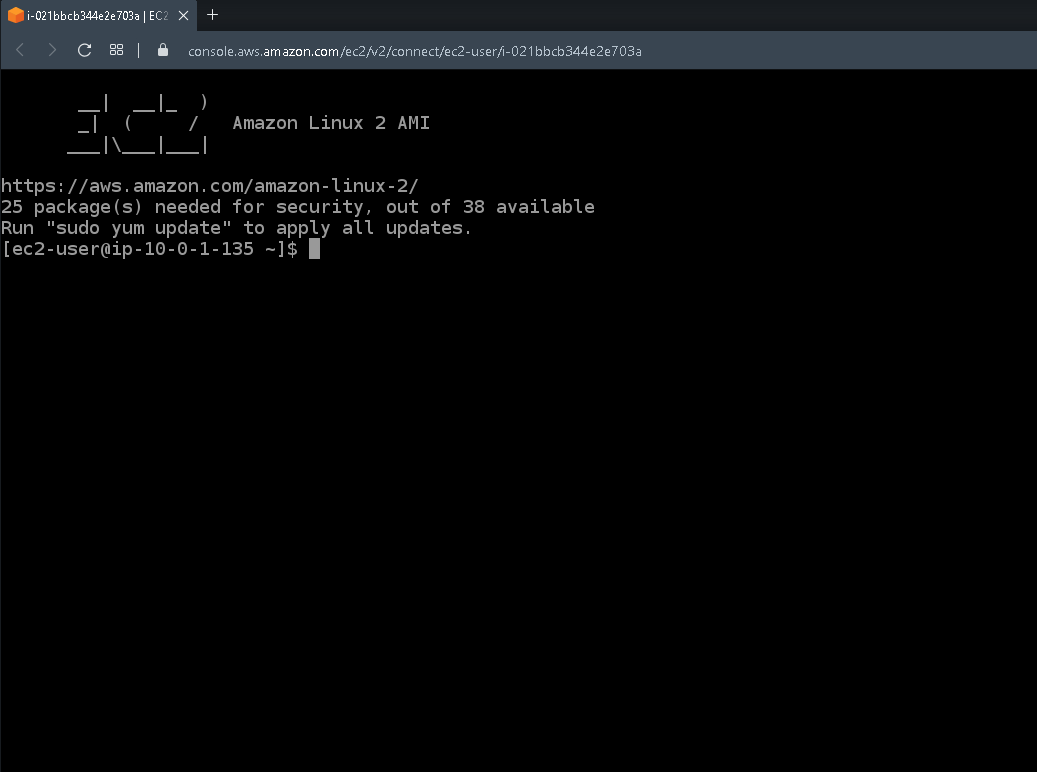
Pocos segundos después el balanceador es generado. 

Generado el balanceador de carga ahora hay que desplegar el código que ejecutará cada una de las instancias. 1. Para ingresar a la instancia hay que ir al panel de EC2, luego a la sección “Instancias”, dar click en una instancia (b) y dar click en “Conectar” (b)



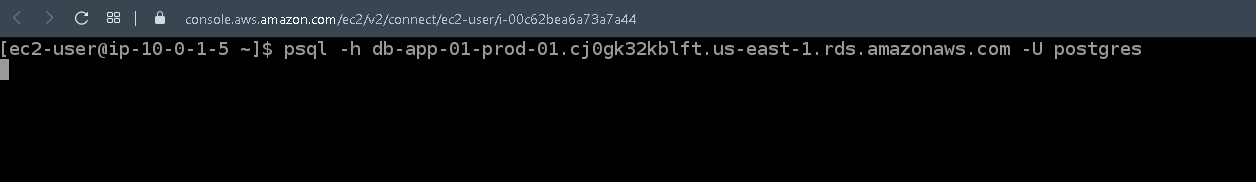
pw-connect-instance-01.png

1. Seleccionar “Conexión de la instancia EC2” (a), luego “Conectar” (b). 

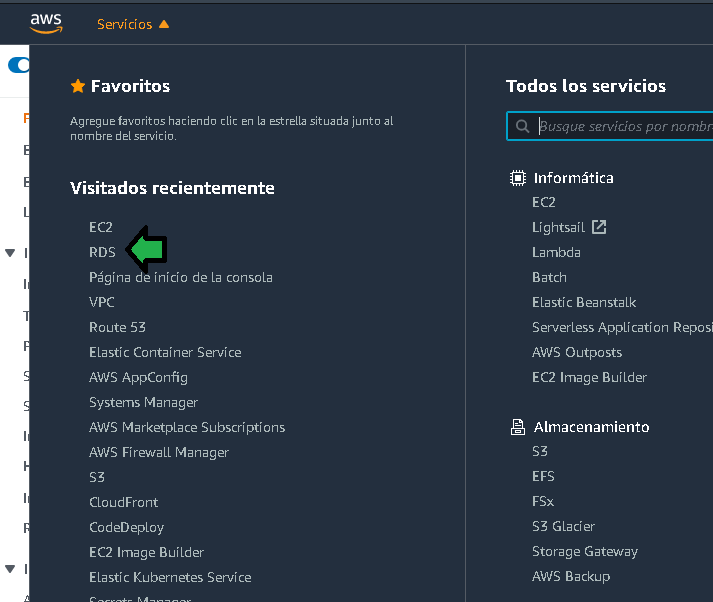
La conexión se establece en unos segundos a una consola bash conectada a la instancia. 

1. Habrá que corroborar el acceso a la base de datos generada con anterioridad. Para lo cual habrá que ejecutar el comando

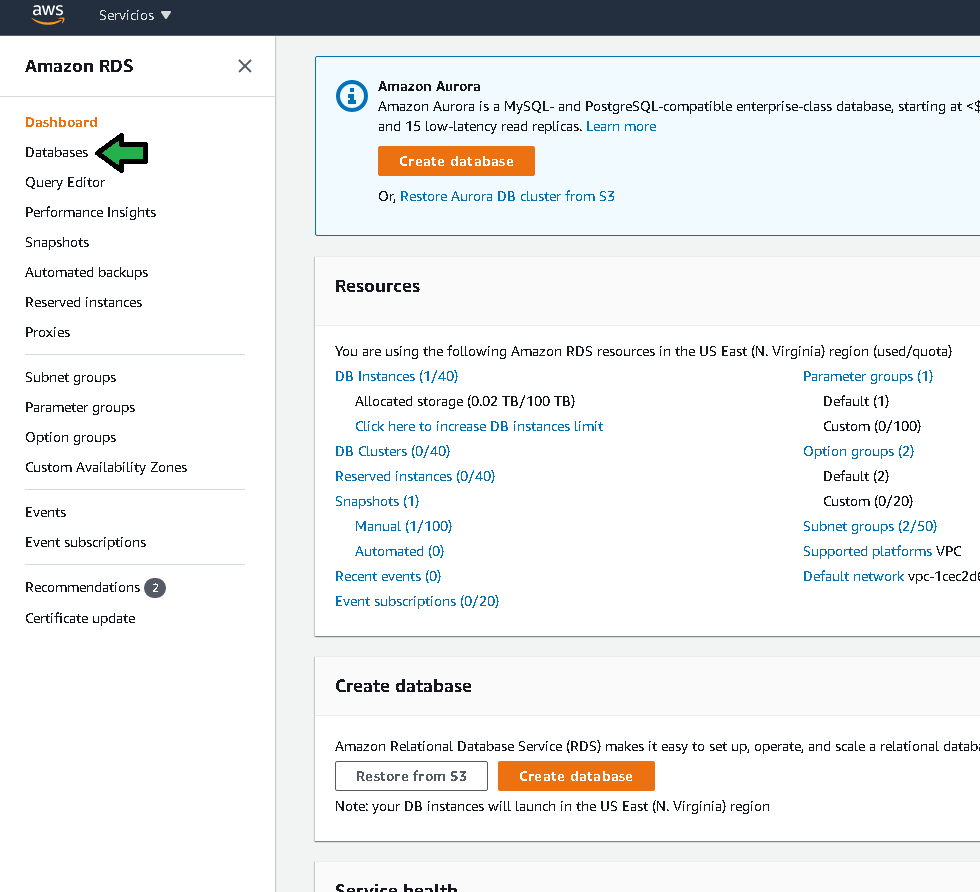
psql -h <endpoint del host> -U <usuario\_de\_base\_datos>

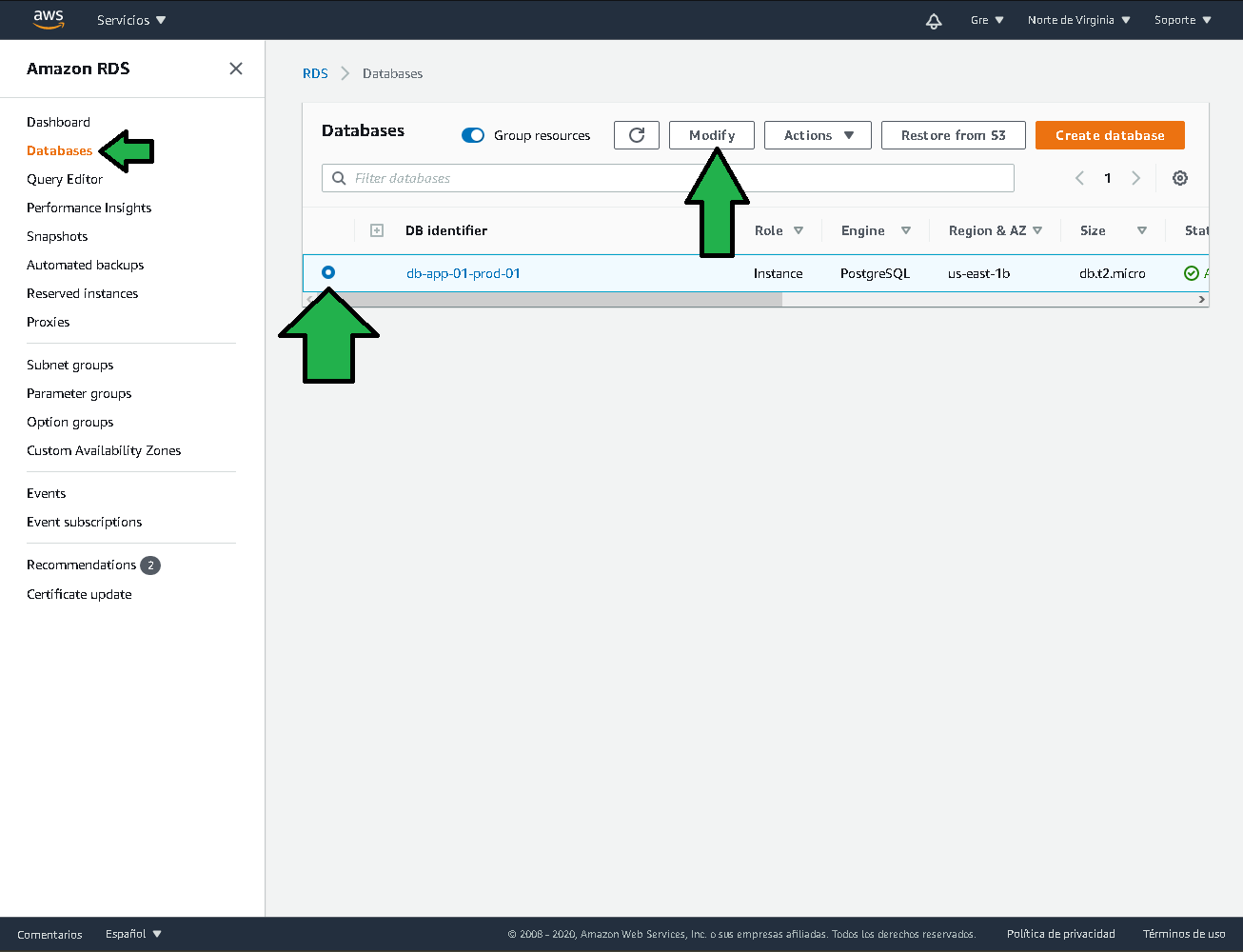
Ejemplo: 

¡En este caso hay un problema!, no se logra la conexión con la base de datos, al no lograr conexión, lo primero que hay que hay que verificar son los grupos de seguridad, recordar que los grupos de seguridad actuan como un firewall, para lo que habrá que dirigirse a la sección de RDS.



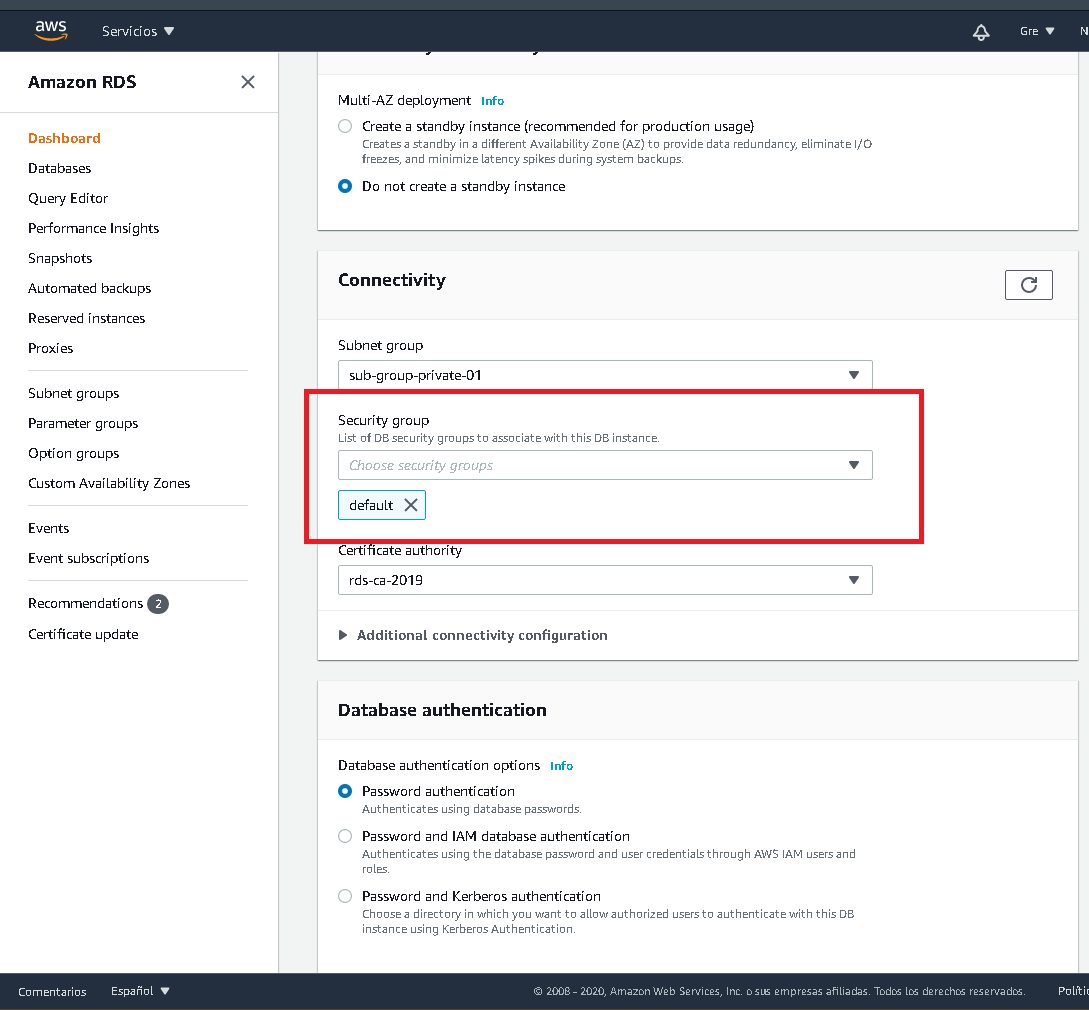
pw-redirect-to-rds-panel.png

* Click en “Databases” 
* Seleccionar al instancia y luego dar click en “Modify”

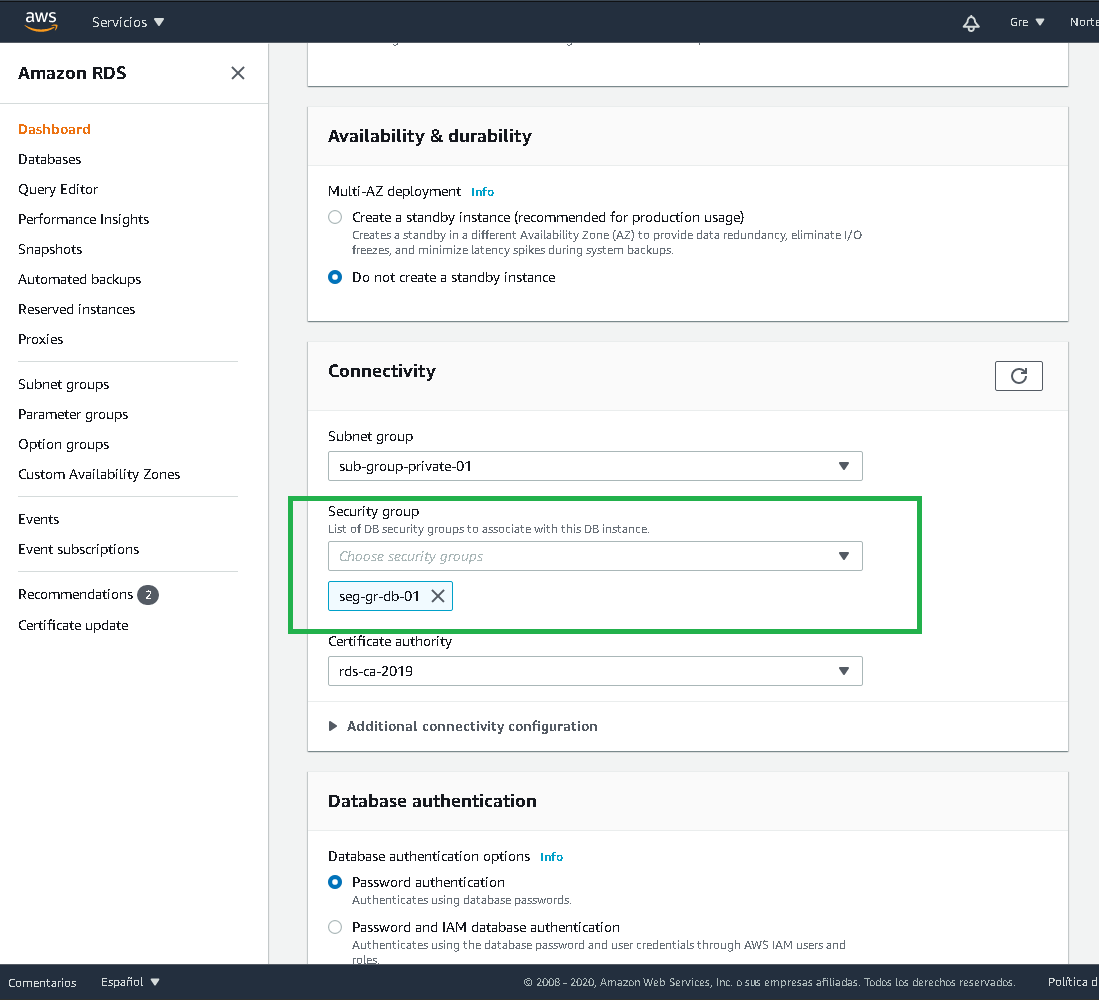
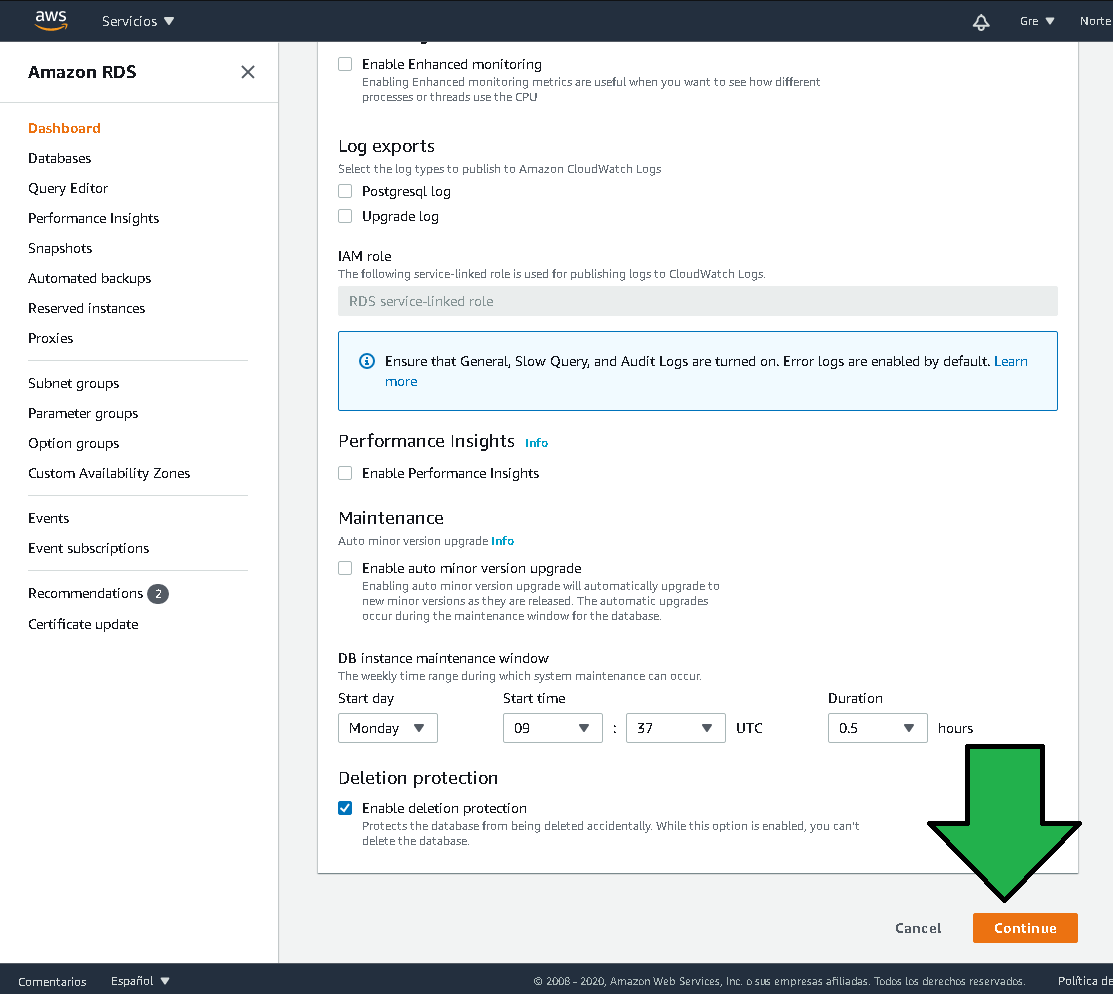
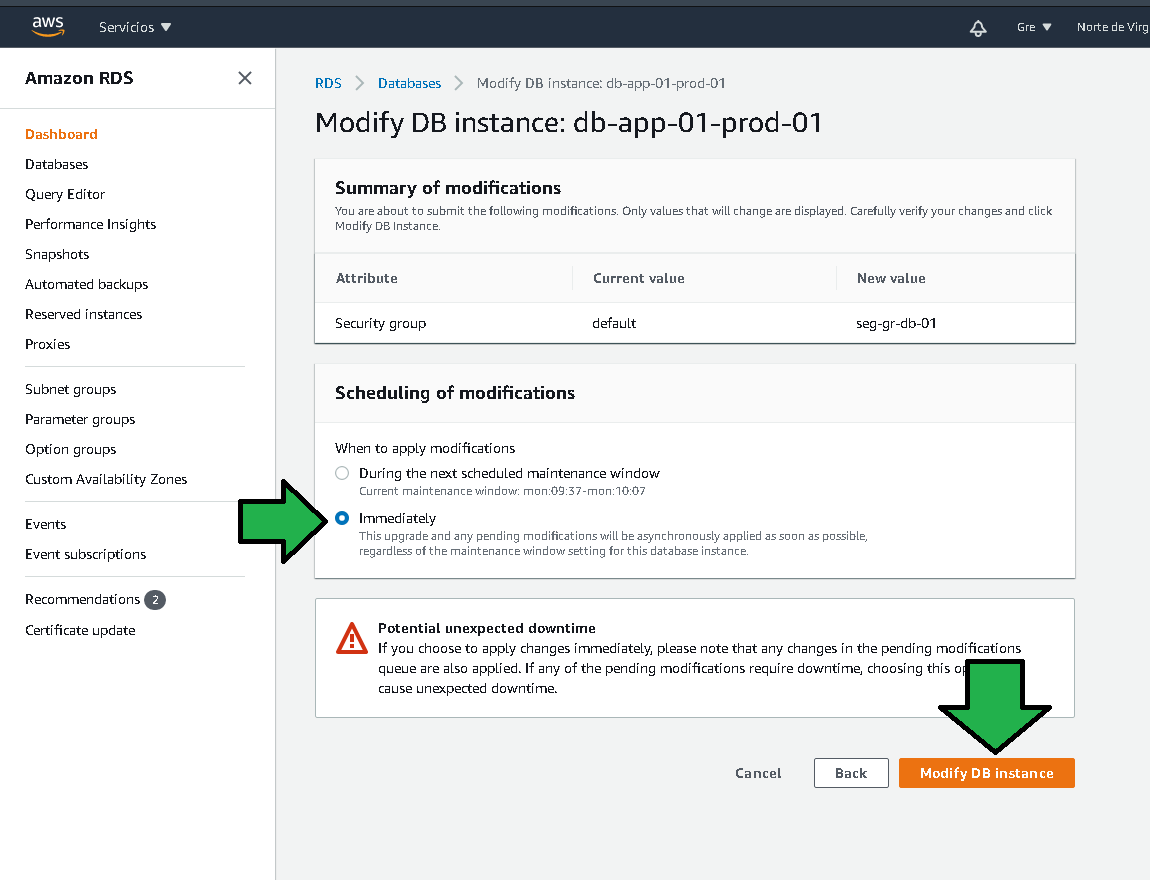
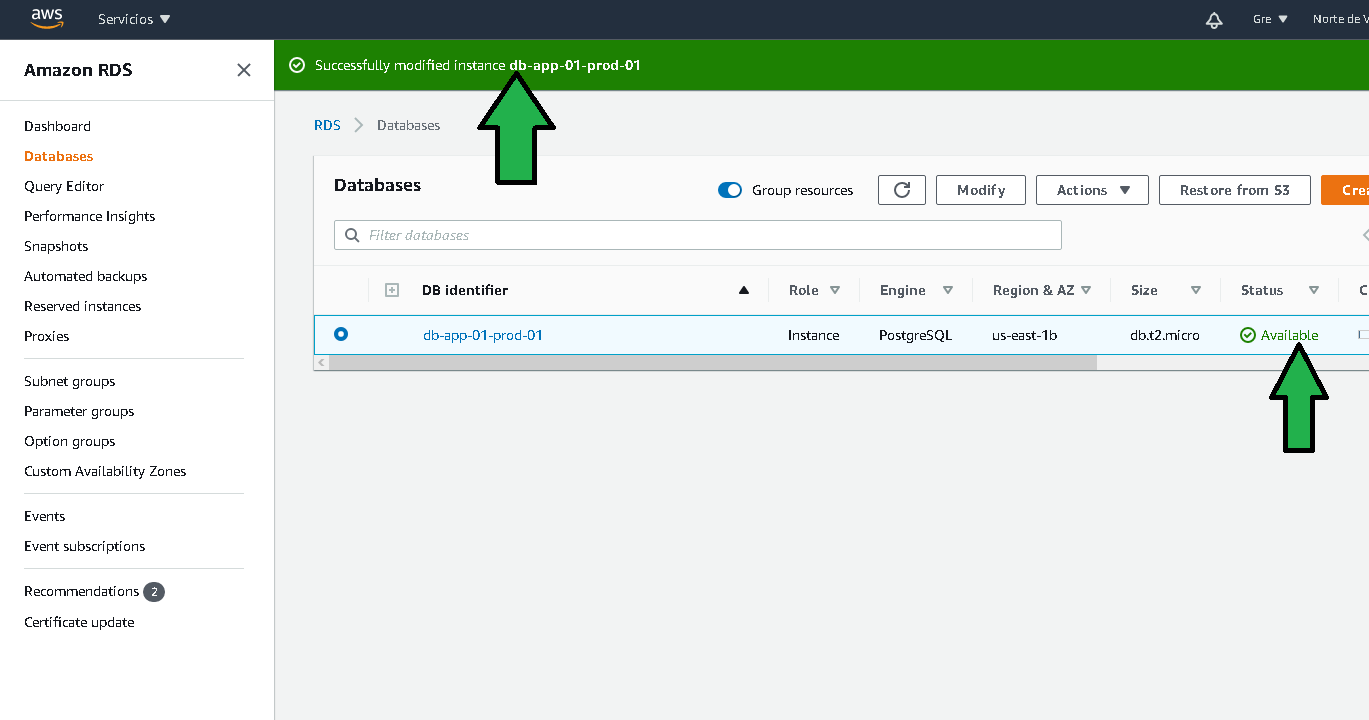


pw-modify-instance.png

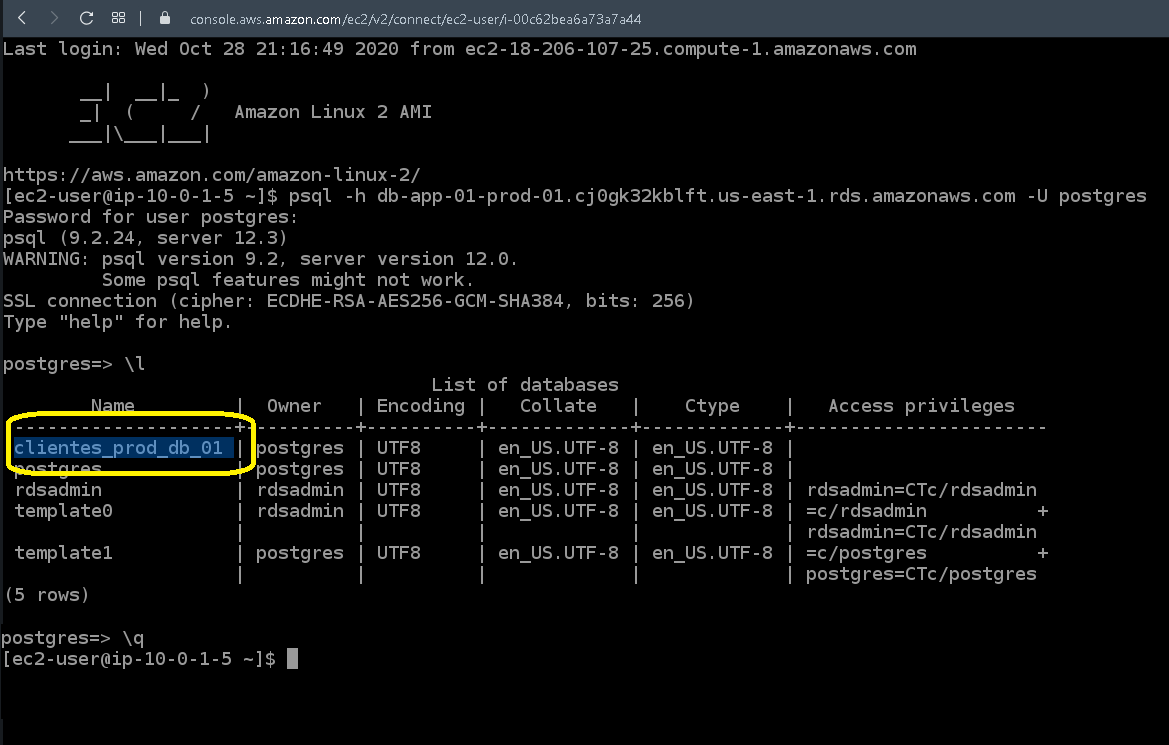
* Hacer scroll hasta la parte de “Connectivity”, aquí se observa el problema, se tiene un grupo de seguridad por defecto, habrá que cambiar el grupo para explícitamente permitir el tráfico al puerto 5432.



pw-bad-security-group.png

* Se selecciona el grupo de seguridad adecuado para el tráfico de Postgres en el puerto 5432. 
* Hacer scroll al final de la pantalla y dar click en “Continuar” 
* Especificar que los cambios se generen inmediatamente, luego dar click en “Modify DB instance”. 
* El cambio es prácticamente instantáneo. 

Regresando a la linea de comandos (si deja de responder solo refrescar pantalla con F5) se puede corroborar la conexión a la instancia de base de datos, incluso se pueden listar las bases de datos, tener el nombre de la base de datos en cuenta.



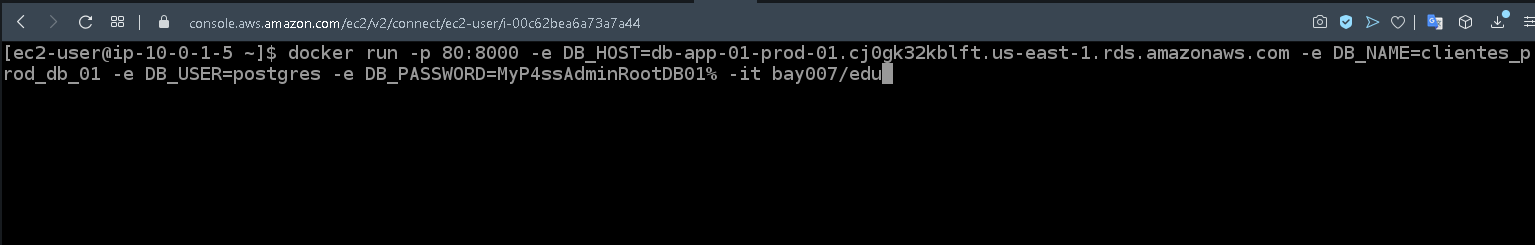
pw-database-connexion-done.png

1. Para seguir con el ejemplo, se deberá ejecutar en la linea de comando el siguiente comando

docker run -p 80:8000 -e DB\_HOST=db-app-01-prod-01.cj0gk32kblft.us-east-1.rds.amazonaws.com -e DB\_NAME=clientes\_prod\_db\_01 -e DB\_USER=postgres -e DB\_PASSWORD=<el-password-de-la-base> -it bay007/edu

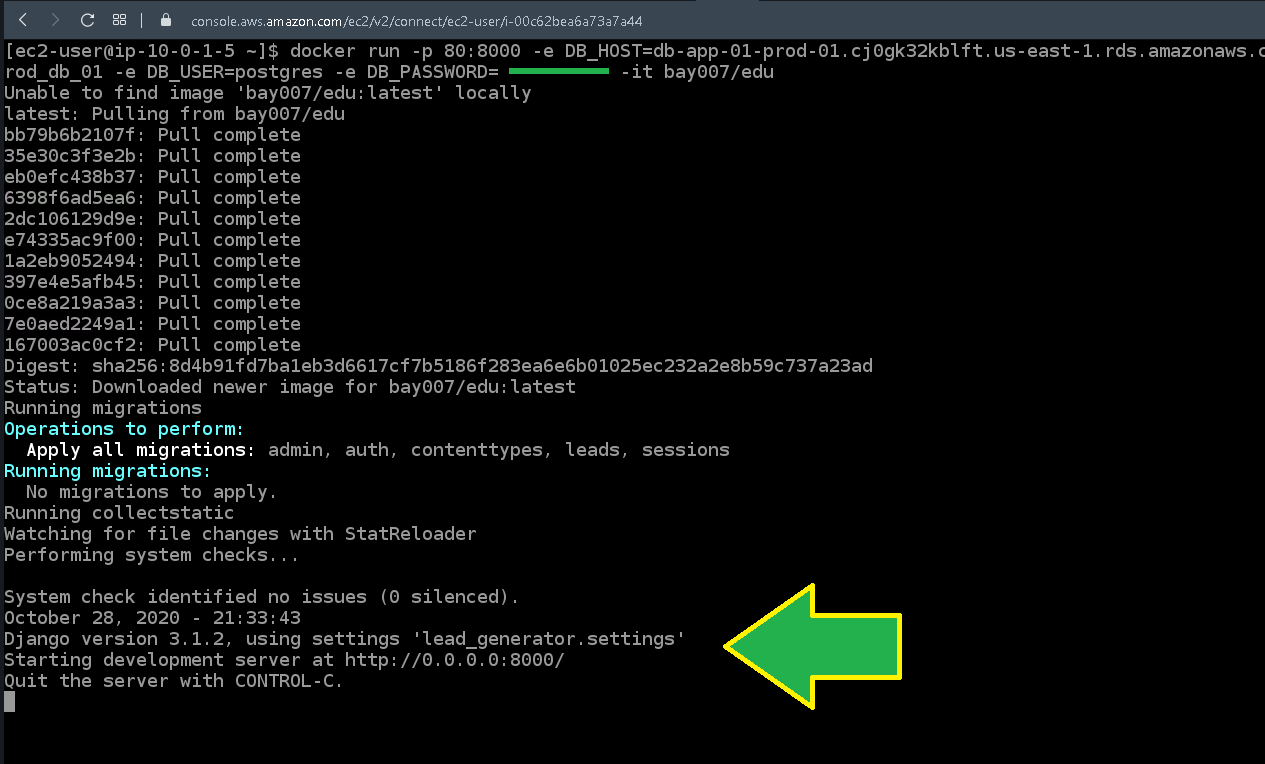
docker run: Indica que se deberá ejecutar un comando de docker En las variables de entorno DB\_HOST, DB\_NAME, DB\_USER, DB\_PASSWORD se deberán establecer los parámetros propios para la conexión con la base de datos.

Por último # TODO Actualizar el DOCKER Hub de BEDU.



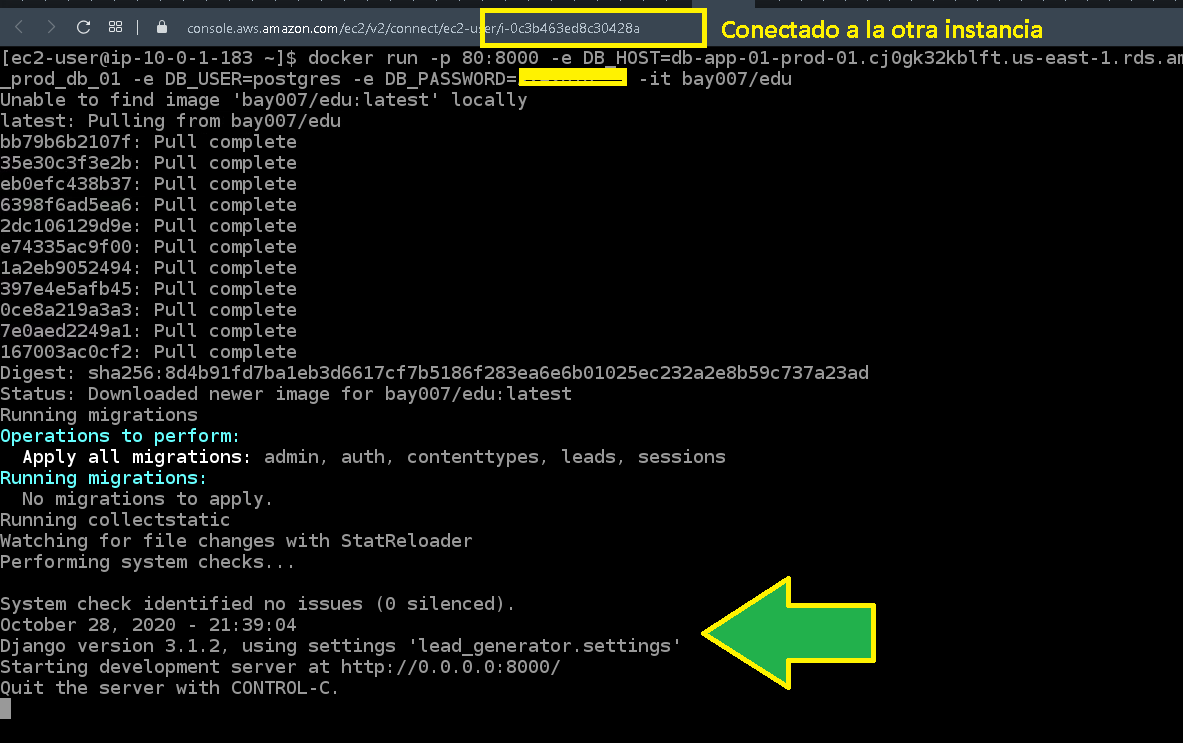
pw-run-docker-instance-01.png

Ejecutado el comando se tendrá un contenedor de Docker con el código fuente funcionando.



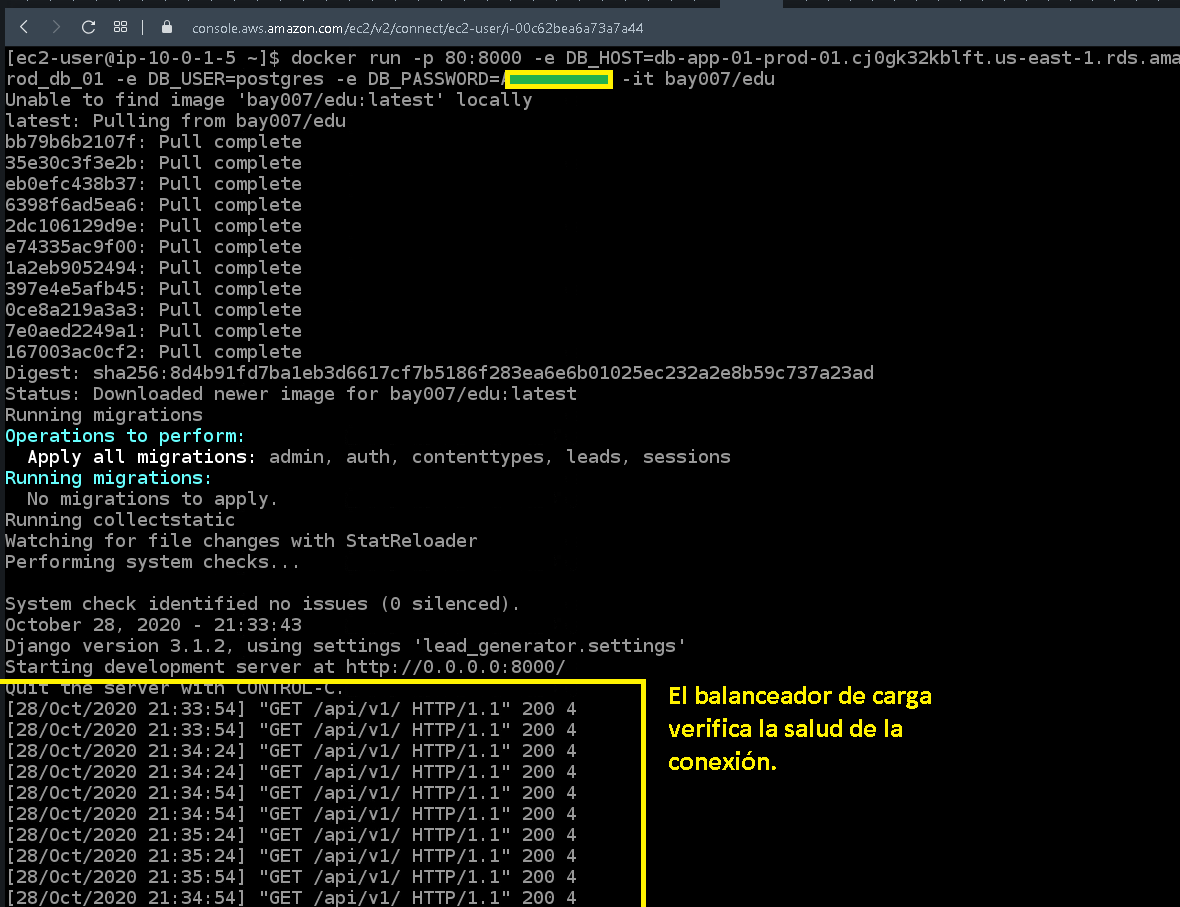
pw-docker-running-instance-01.png

1. La ejecución del contenedor de Docker se debe efectuar en la otra instancia de EC2, para lo que hay que conectarse a ella y ejecutar el mismo comando de Docker del paso 4.

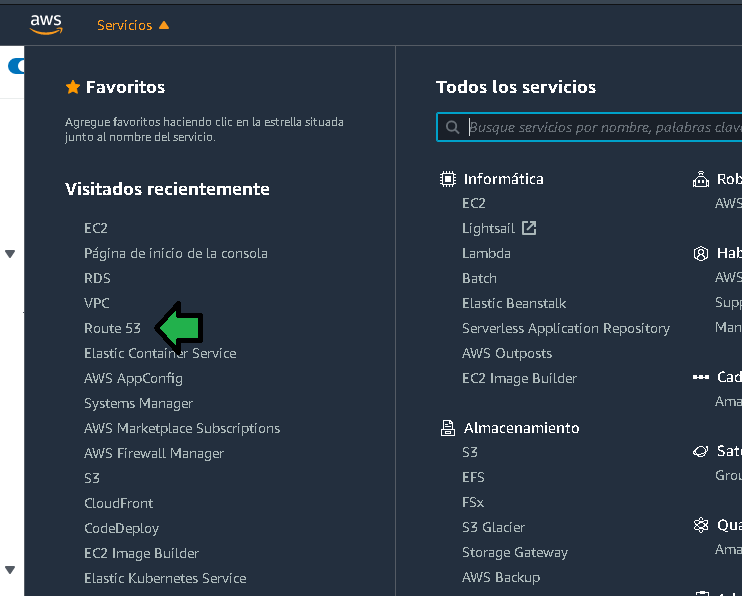
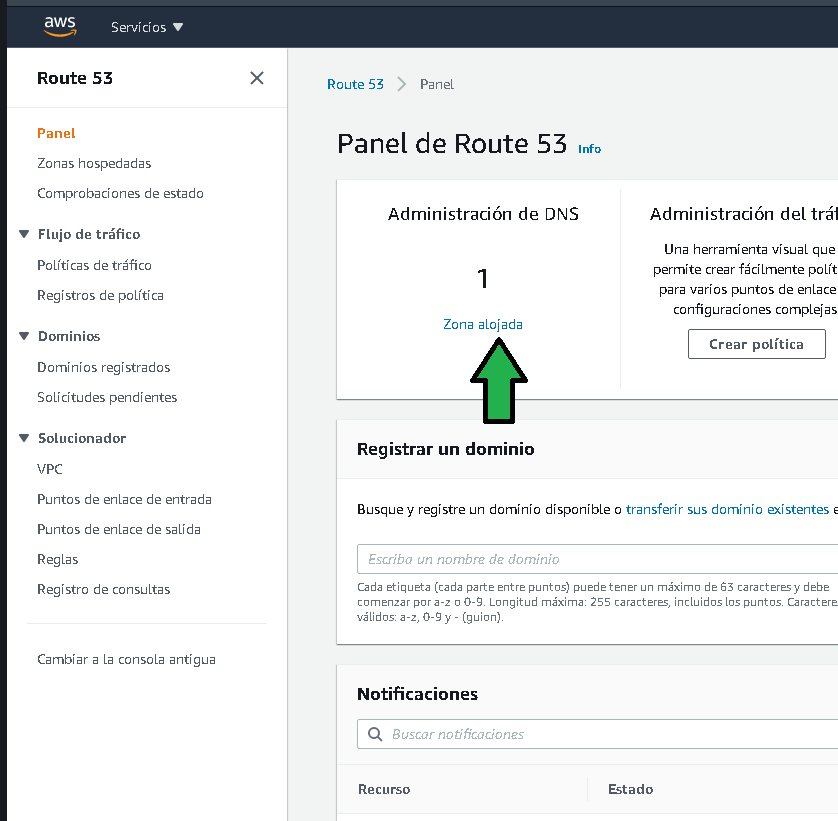
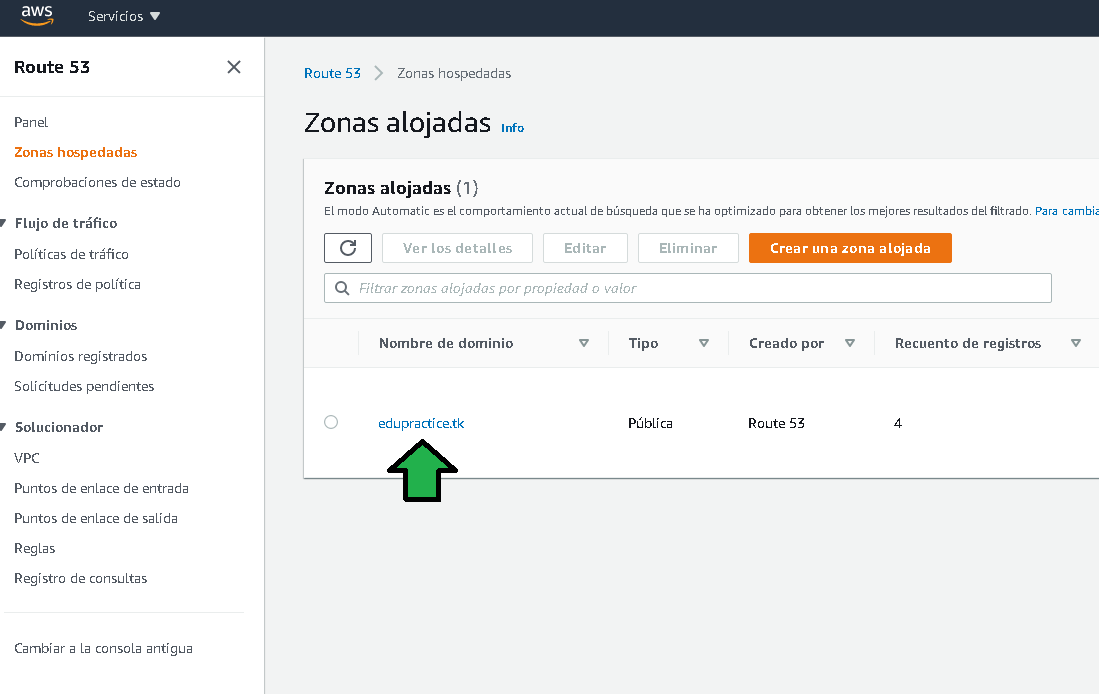
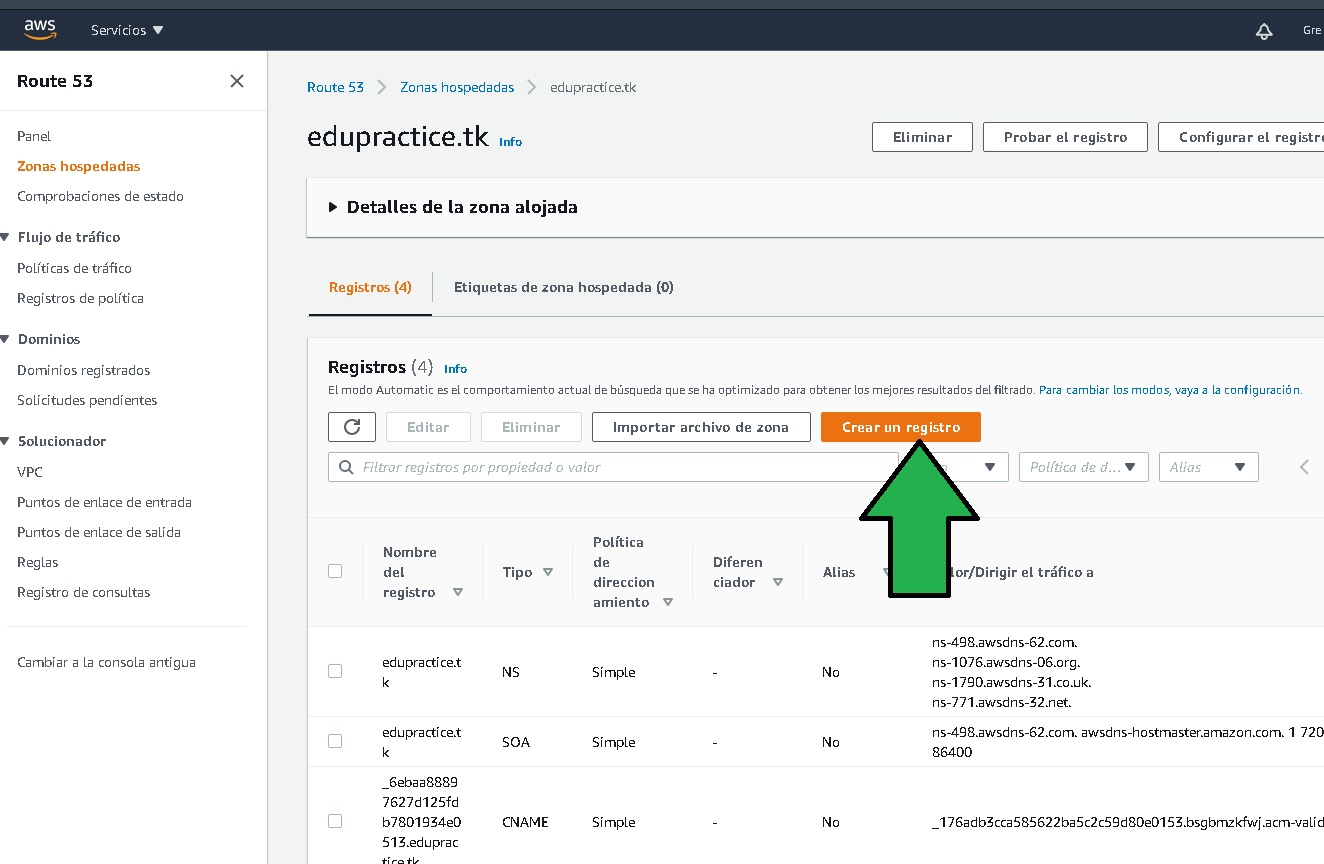
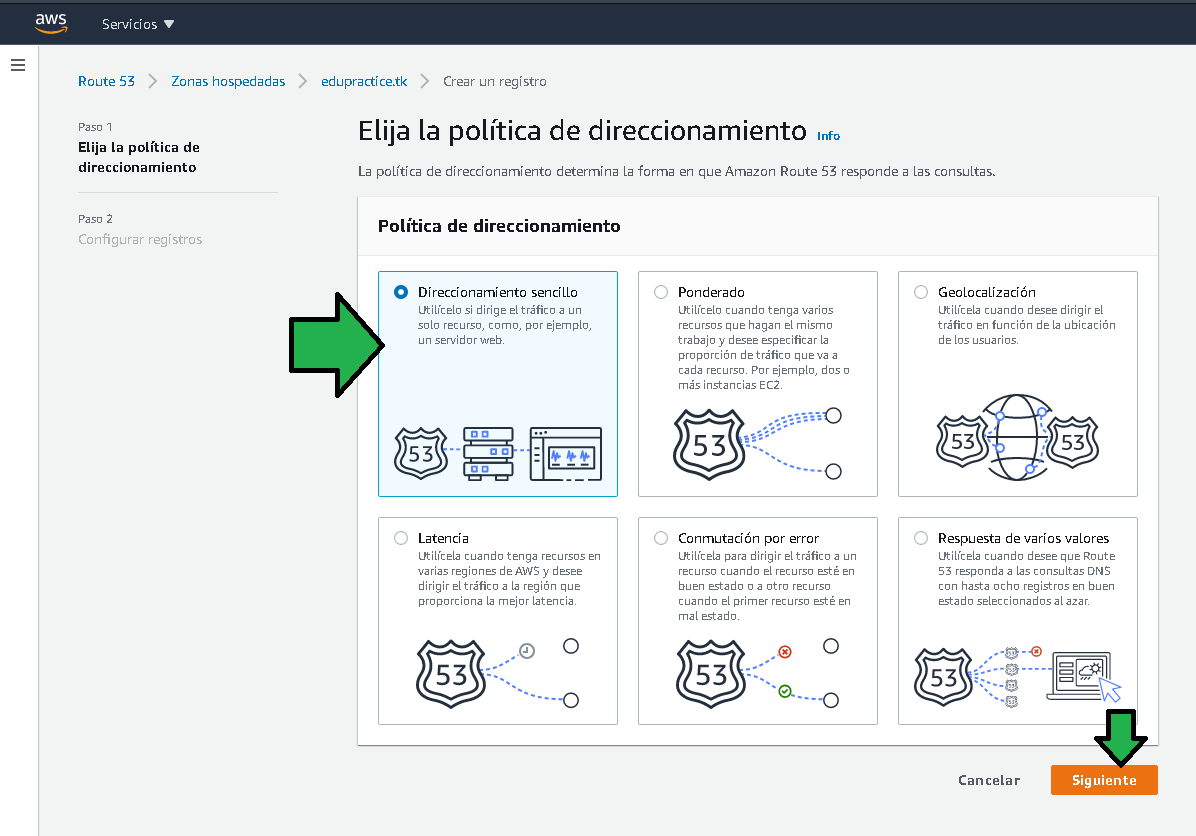
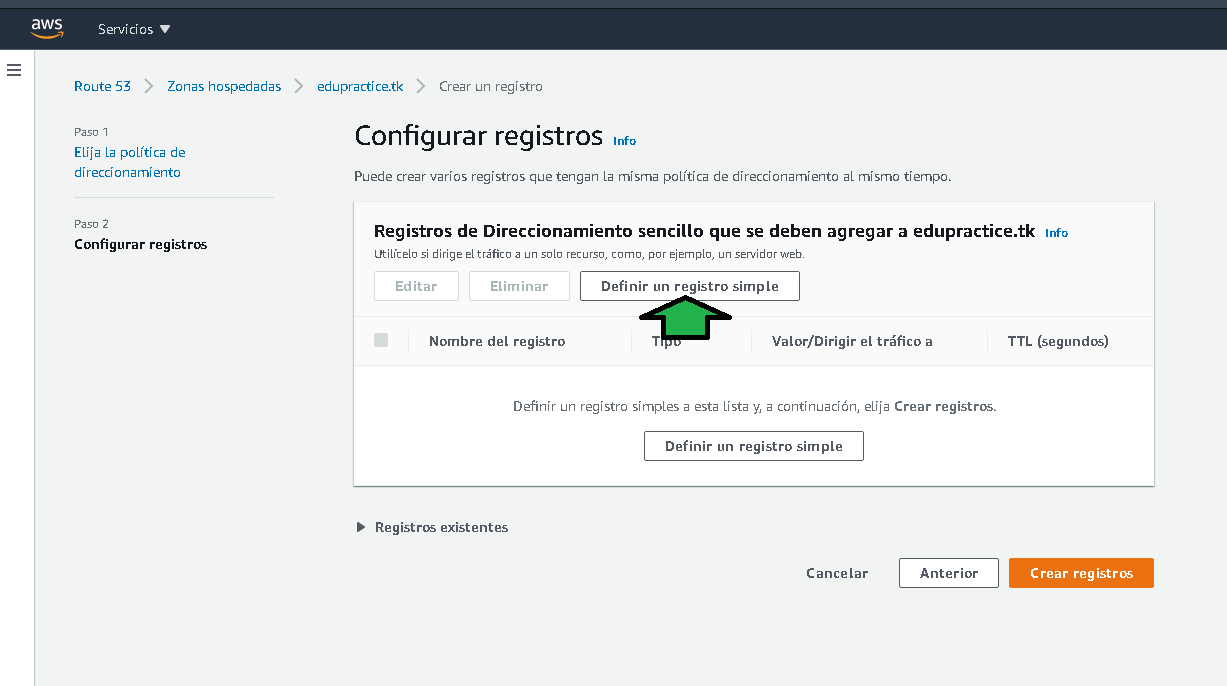
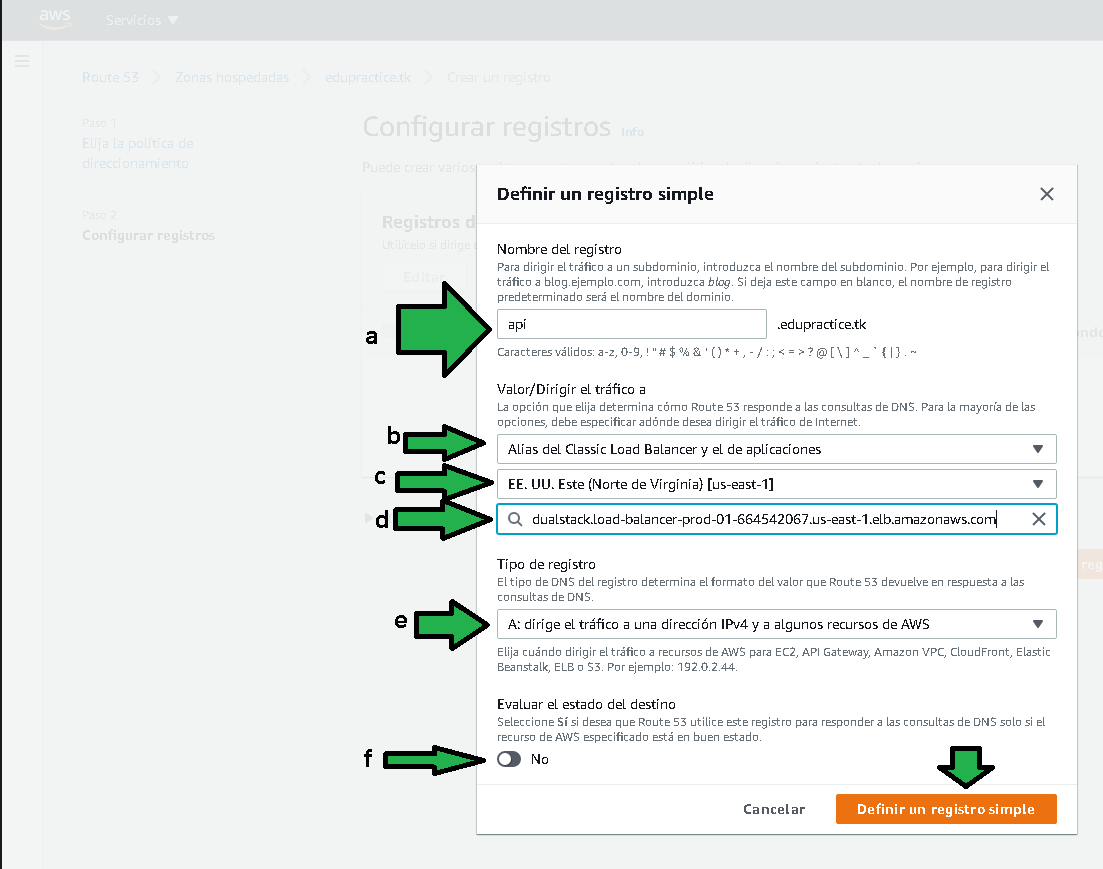
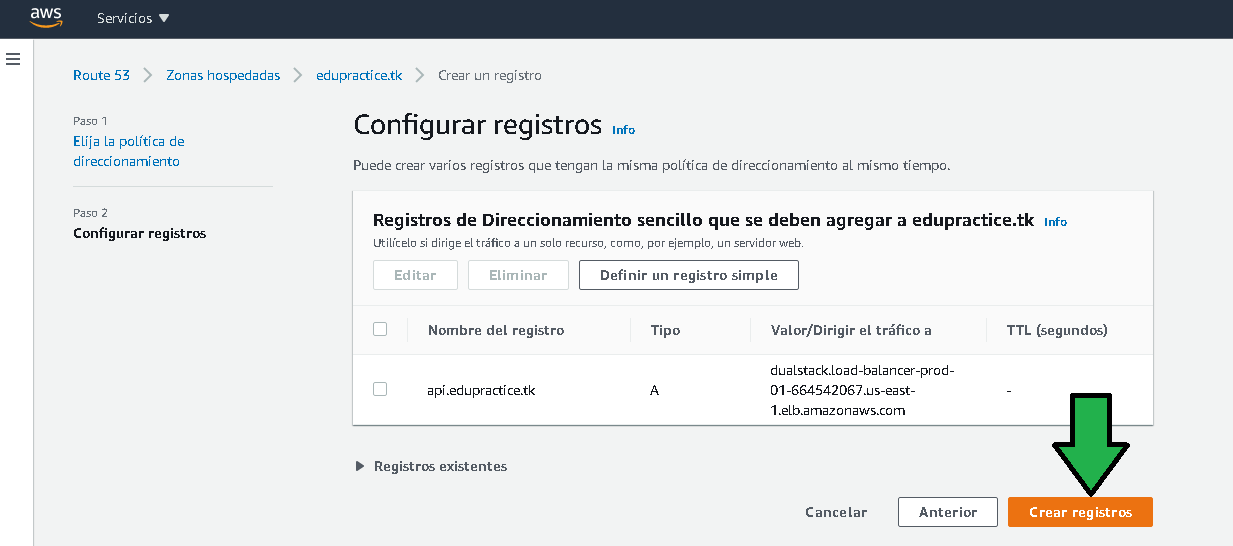


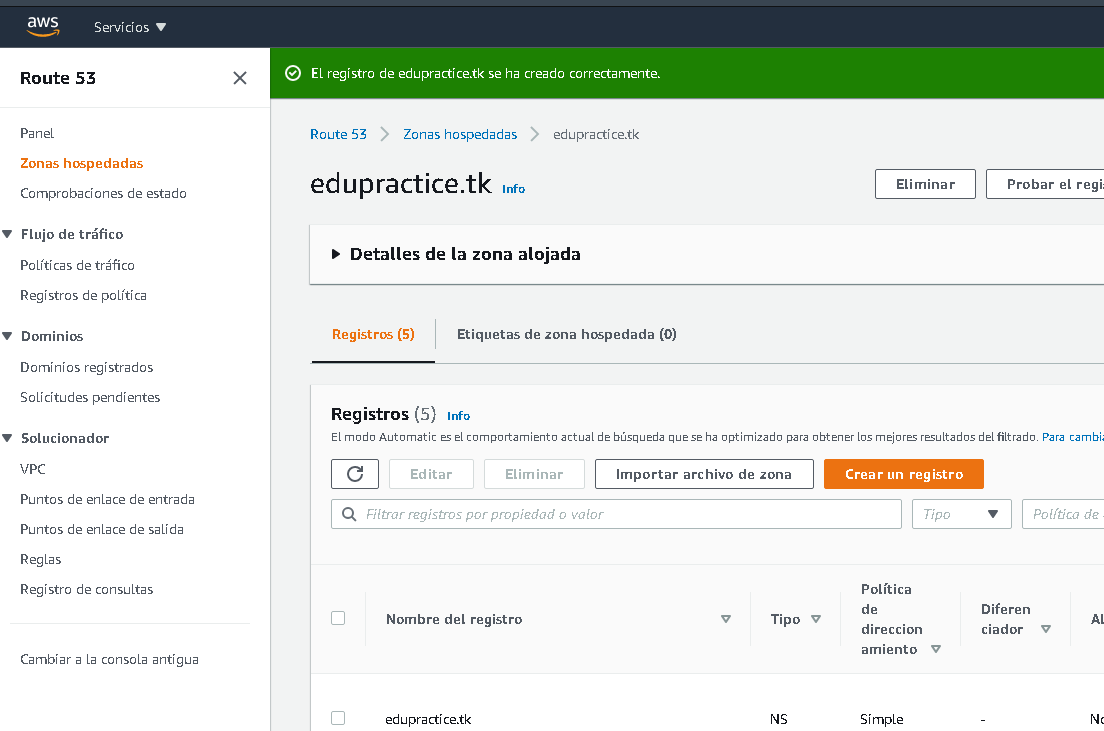
pw-docker-running-instance-02.png

Pasados algunos minutos se comienza a observar tráfico hacia la instancia EC2 y en consecuencia a los contenedores ejecutándose. Este tráfico es generado por el balanceador de carga probando la salud del servicio.

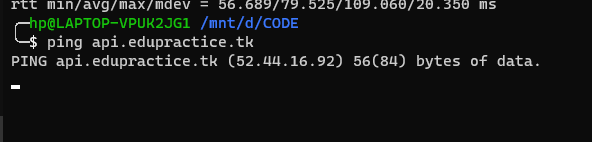


pw-health-check-load-balancer.png

1. Ahora como paso siguiente se se debe configurar el balanceador de carga con un subdominio, ya que las peticiones web no llegarán directamente a las instancias de EC2 y a los contenedores que estan dentro de ellas, quien recibe el tráfico HTTP y HTTPS será el balanceador de carga, y para poder llegar a él se debe configurar un subdominio, para lo cual se debe ir al servicio Route 53. 
2. Click en “Zonas alojadas” 
3. Entrar al dominio para su modificación. 
4. Click en “Crear registro” 
5. Seleccionar “Direccionamiento sencillo” 
6. Click en “Definir un registro simple” 
7. Configurar el registro como:
8. Establecer el subdominio como api
9. Buscar el servicio de “Balanceo de carga clásico y de aplicaciones”
10. Seleccionar la región donde se encuentra el balanceador de carga.
11. Seleccionar el balanceador de carga, hay que notar el dominio del balanceador, por ello importante nombrar los recursos.
12. Definir redirigir el tráfico a IPv4
13. Deshabilitar la opción. 
14. Click en “Crear registros” 

Segundos después el registro es creado 

1. Cinco minutos después de generado el registro al hacer ping ya hay una resolución del subdominio api, esperar hasta que el ping tenga resolución a una IP. (Notar que el balanceador no responderá al ping, es normal ya que el balanceador solo tiene permitido por el grupo de seguridad y la propia configuracion del balanceador majenar tráfico de los puertos 80 y 443 es decir HTTP y HTTPS)

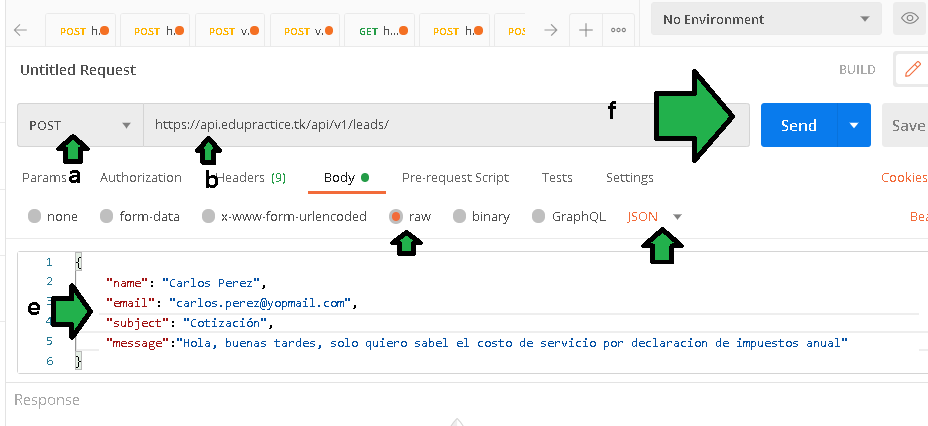


pw-ping-done.png

1. Con Postman se comenzará la prueba de funcionamiento de la API.
2. Seleccionar el protocolo POST.
3. Establecer la la URL a la que se hará el request, conformada por https:// más el subdominio recién configurado en Route 53 más el path /api/v1/leads/ (poner atención en la diagonal al final).
4. Establecer la petición como raw.
5. Establecer el tipo de petición como JSON.
6. Establecer el cuerpo del mensaje como:

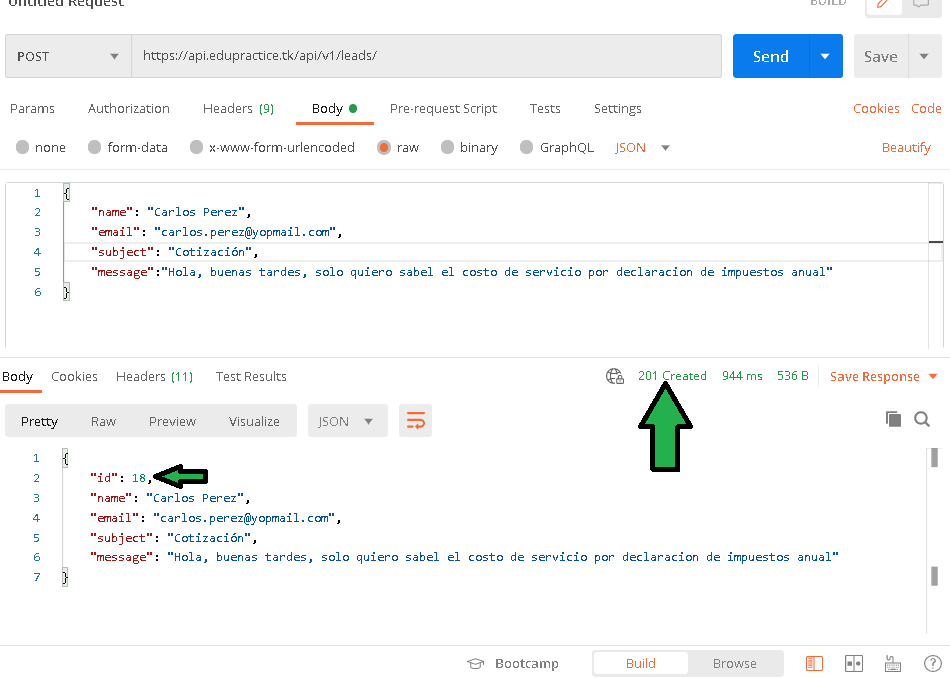
{  
 "name": "Carlos Perez",  
 "email": "carlos.perez@yopmail.com",  
 "subject": "Cotización",  
 "message":"Hola, buenas tardes, solo quiero saber el costo de servicio por declaración de impuestos anual"  
}

1. Generar la petición.



pw-make-petition-request.png

Hecha la petición se ve el código de estatus 201 lo que indica que el registro fue creado en la base de datos, además del ID que la base ha asignado al registro.



pw-postman-request-done.png

Viendo la consola se puede ver la petición recién hecha.

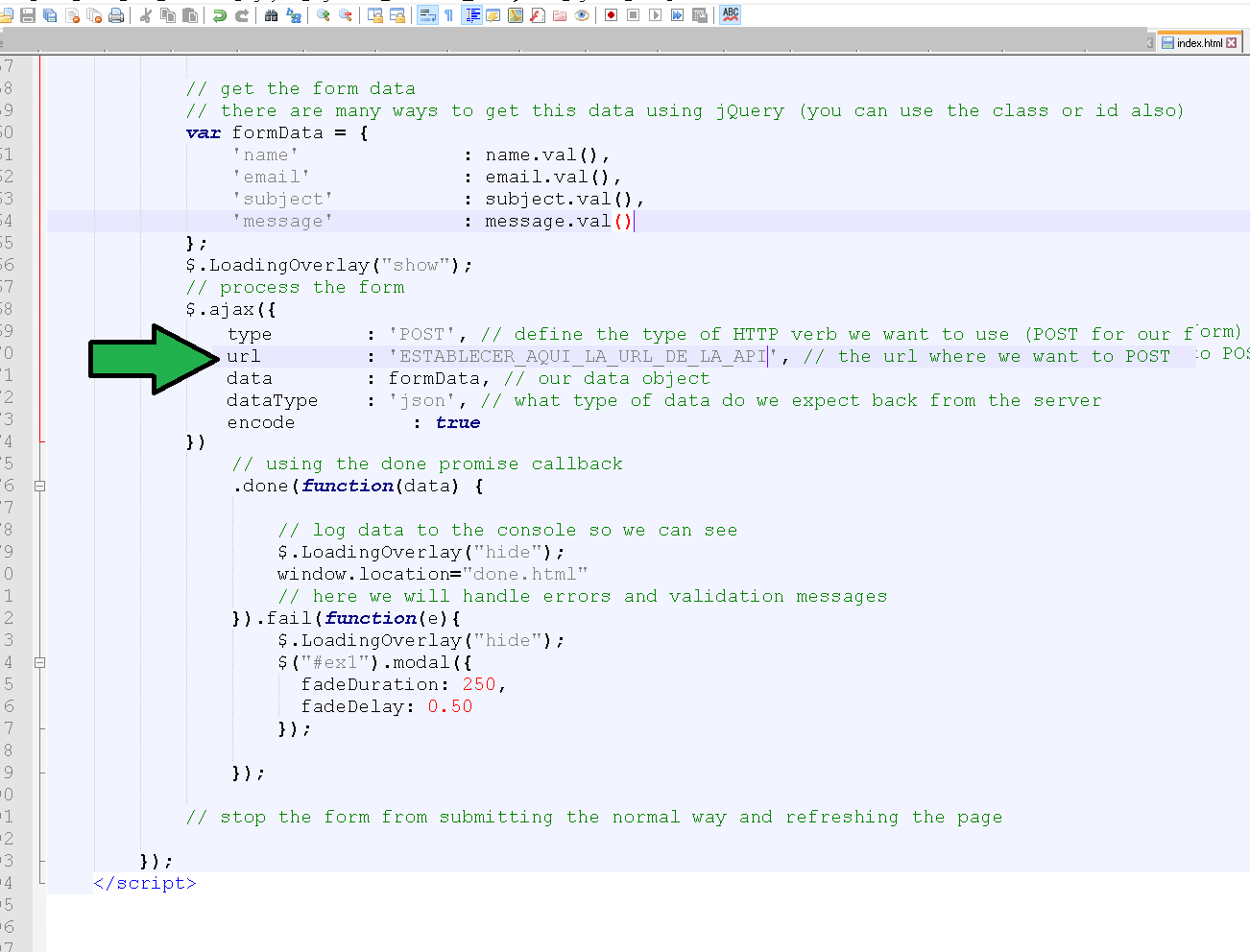


pw-request-done-on-cli.png

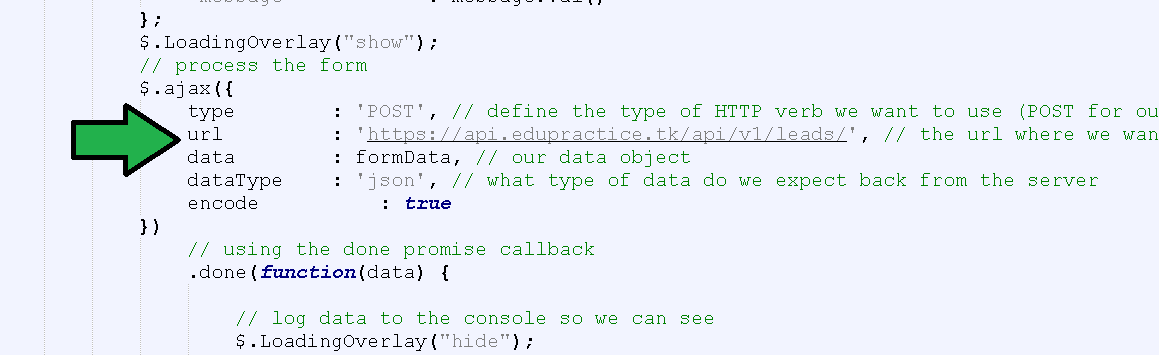
Ya corroborado el funcionamiento de la API, se procede con la descarga del código fuente HTML que estará de cara al usuario final.

Se debe descargar desde #TODO Establecer el repositorio desde donde se descargará el código fuente.

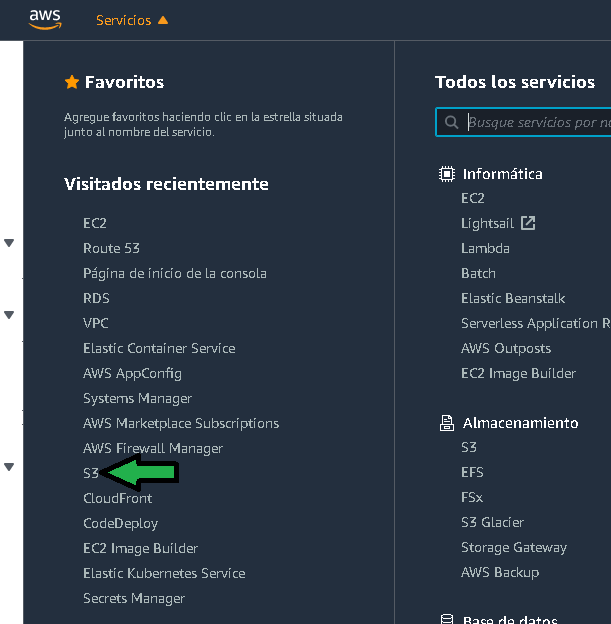
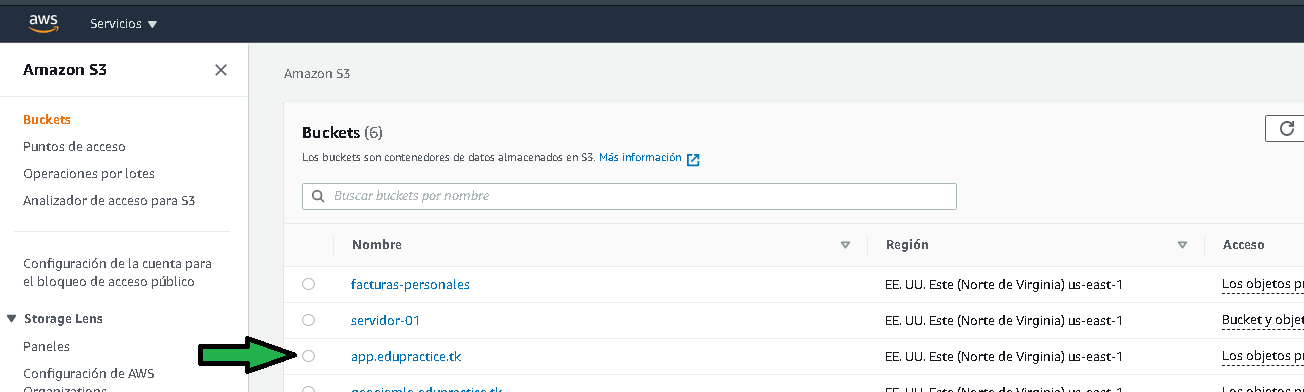
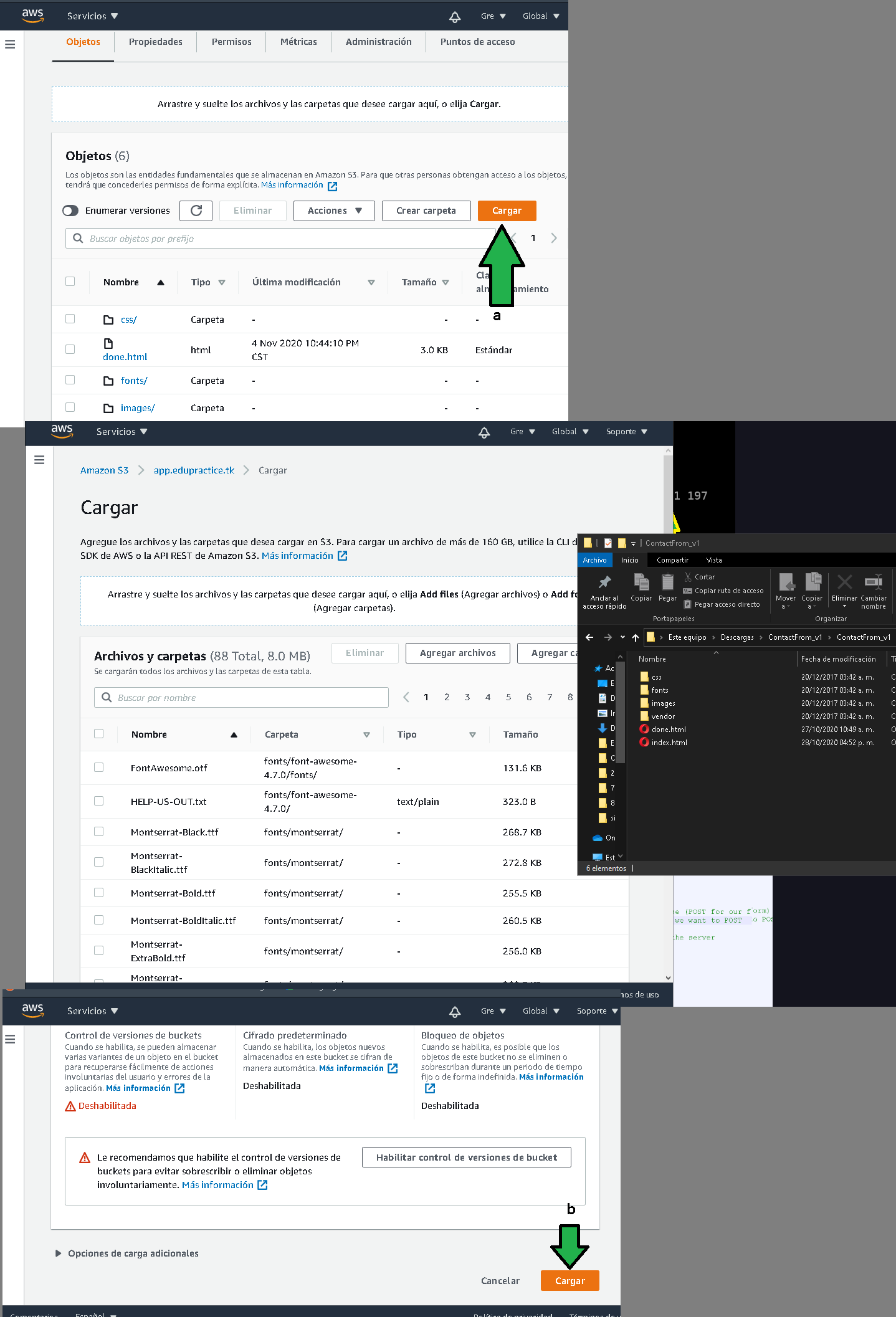
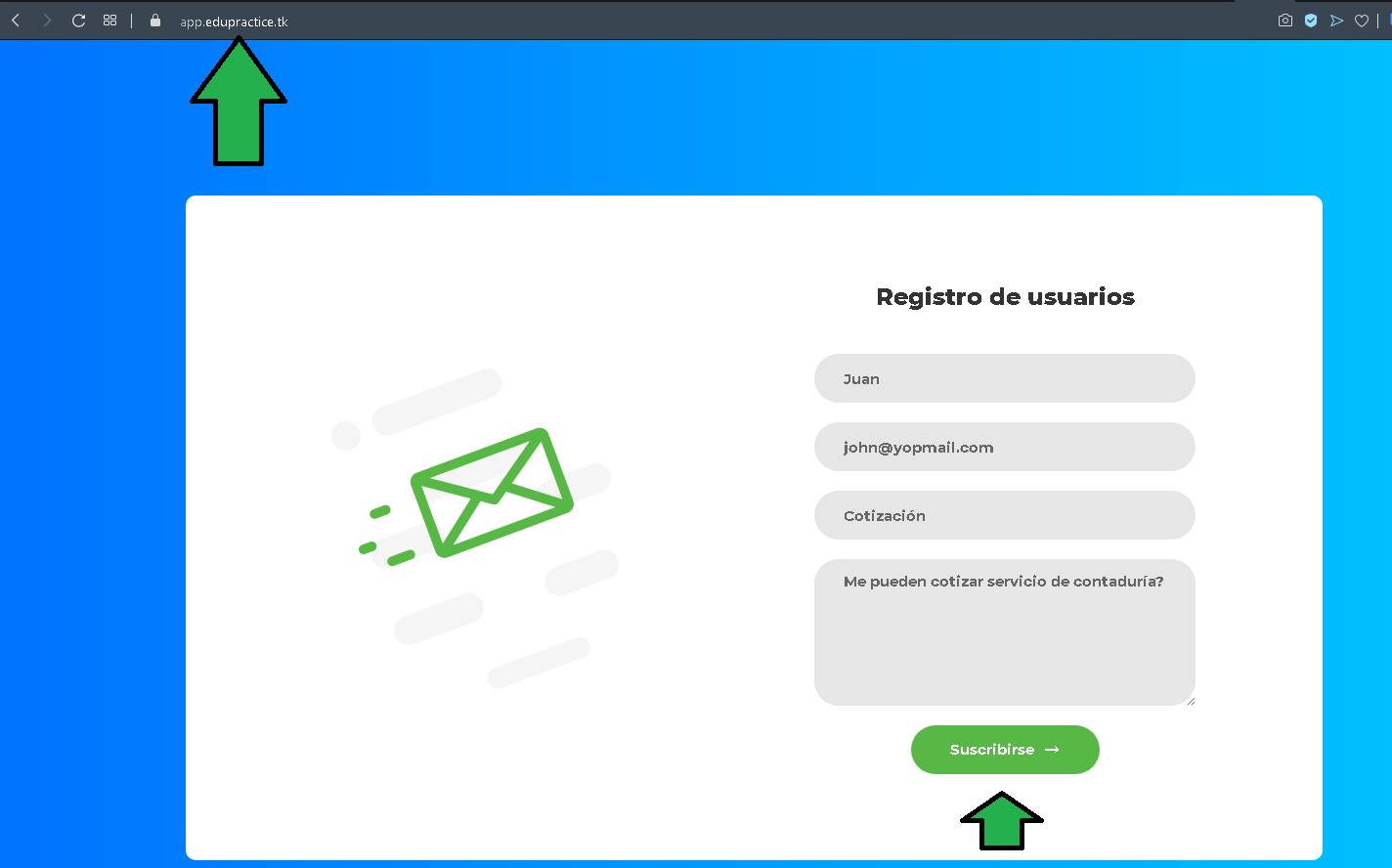
1. Con el código fuente descargado, descomprimir el zip, abrir el archivo index.html en un editor como Notepad, Vim, VSCode, SublimeText, Atom, Notepad++ o cualquier otro.
2. Ir a la siguiente sección de código para establecer la URL de la API, misma que fue usada en Postman.

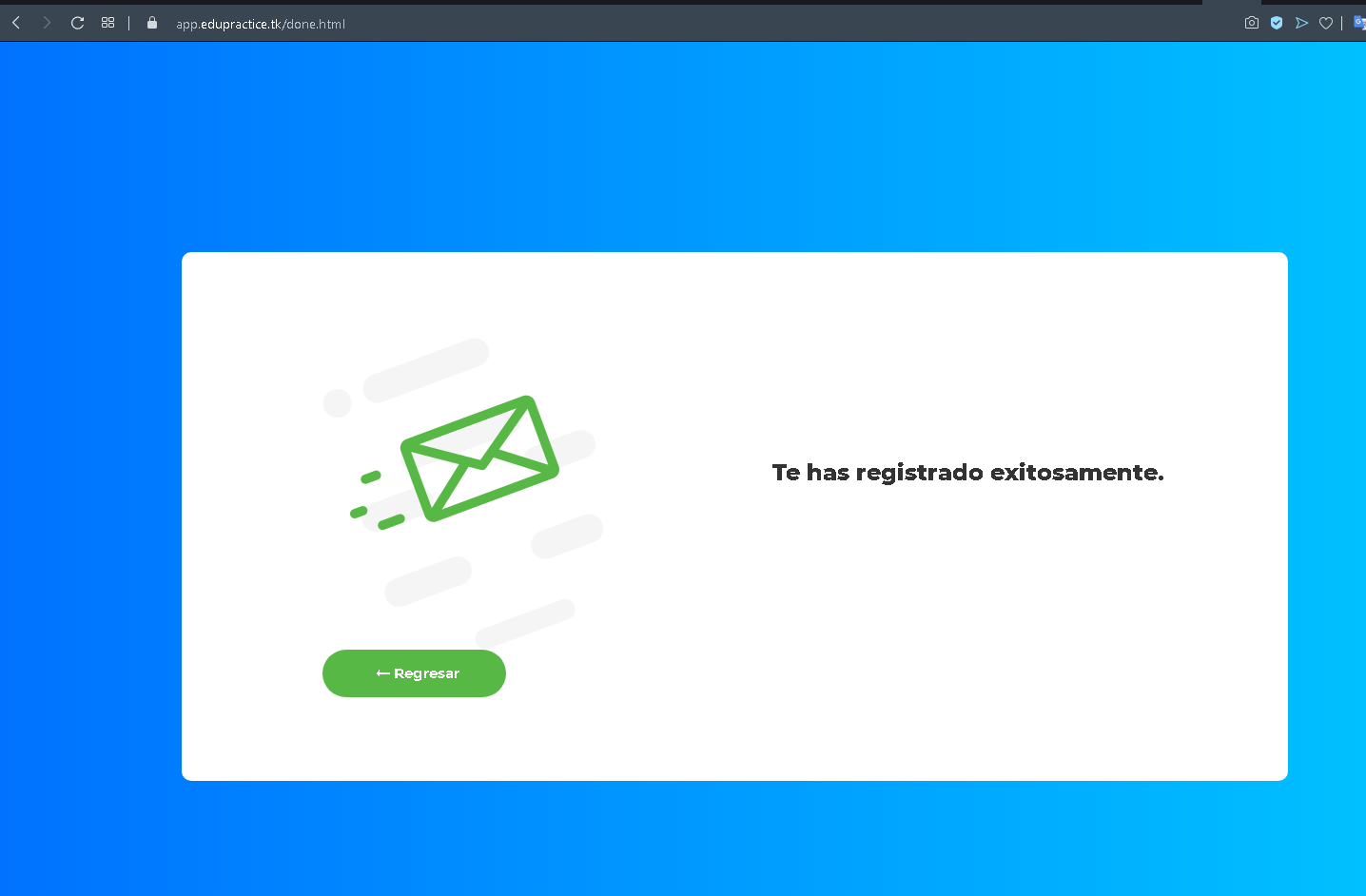


pw-edit-domain-01.png

En el ejemplo trabajado se tiene la URL, el alumno deberá establecer la URL con su propio dominio. 

Se guarda el archivo.

1. Subir los archivos al bucket S3. 
2. Seleccionar el bucket que hostea la página web. 
3. Dar click en “Cargar” (a), arrastrar los archivos y carpetas, dar click en “Cargar” (b). 
4. Ya subidos los archivos, acceder al subdominio app, se debe visualizar un formulario de captación de datos que al modificar la URL ya guardará los datos en la base de datos. 

Al hacer click en “Suscribirse” los datos serán guardados. 

¡Felicidades!, ha sido un trabajo arduo llegar hasta este punto, el estado del proyecto es el siguiente: 