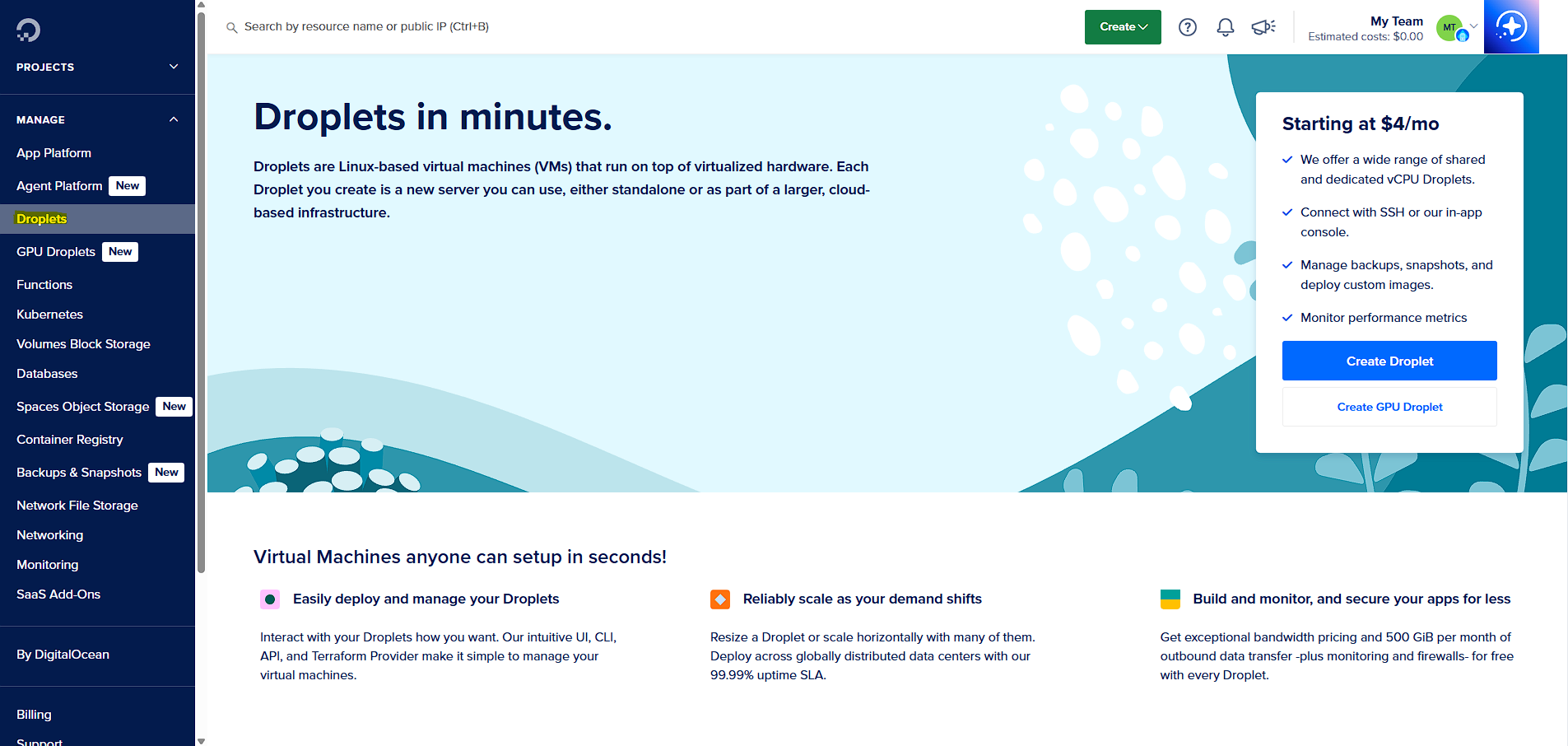
Una vez finalizada mi página web, voy a proceder a subirla a un servidor para que esté accesible desde cualquier dispositivo. Para ello, dado que al tener el plan de estudiantes de Github tenemos la posibilidad de solicitar un crédito de 200$ en la plataforma de DigitalOcean, me he decantado por el servicio que tienen de droplets …..EXPLICAR AQUÍ QUE SON LOS DROPLETS……

Antes de proceder a crear el droplet, primero procedí a crearme una cuenta de DigitalOcean, vincularla con mi cuenta de Github y verificar la identidad con mi tarjeta de débito mediante un cargo de unos 11€. Una vez verificada mi identidad, procedieron a realizarme el reembolso del cargo realizado y a darme los créditos tal y como se puede ver en esta imagen:

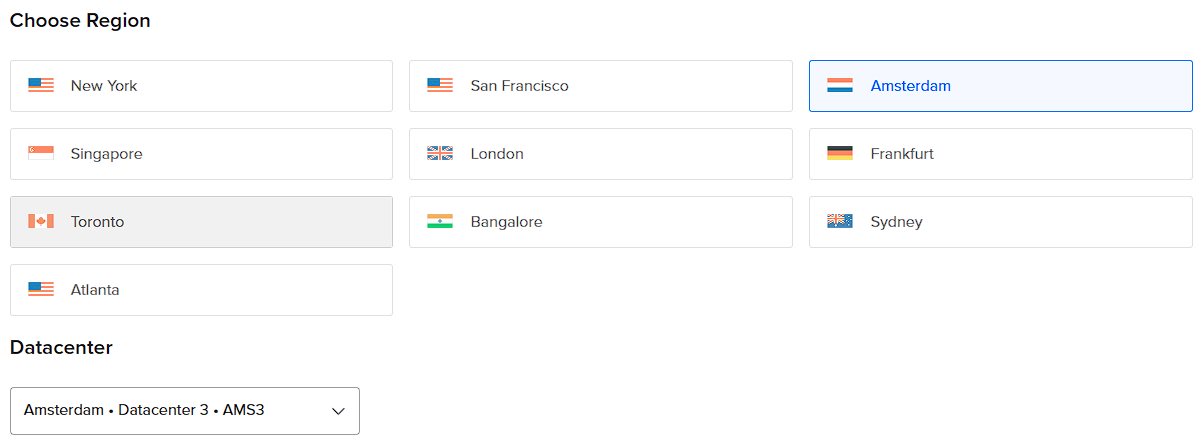


Ahora me dirijo a la ventana “Droplets” y le doy click al botón “Create Droplet”:

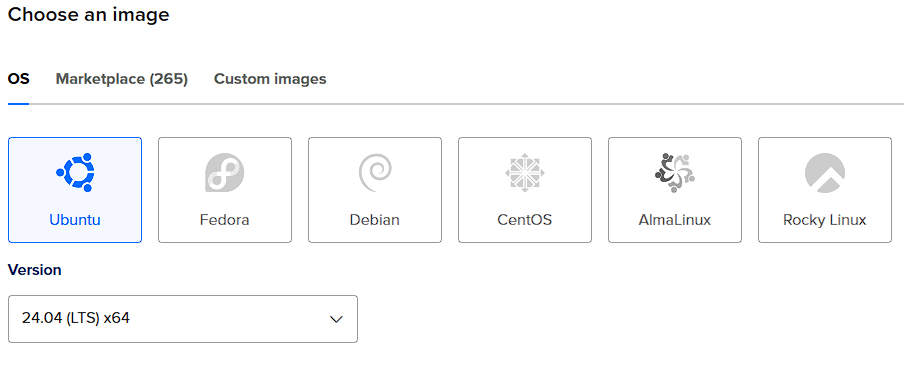


Ahora configuro mi máquina virtual con las siguientes características:

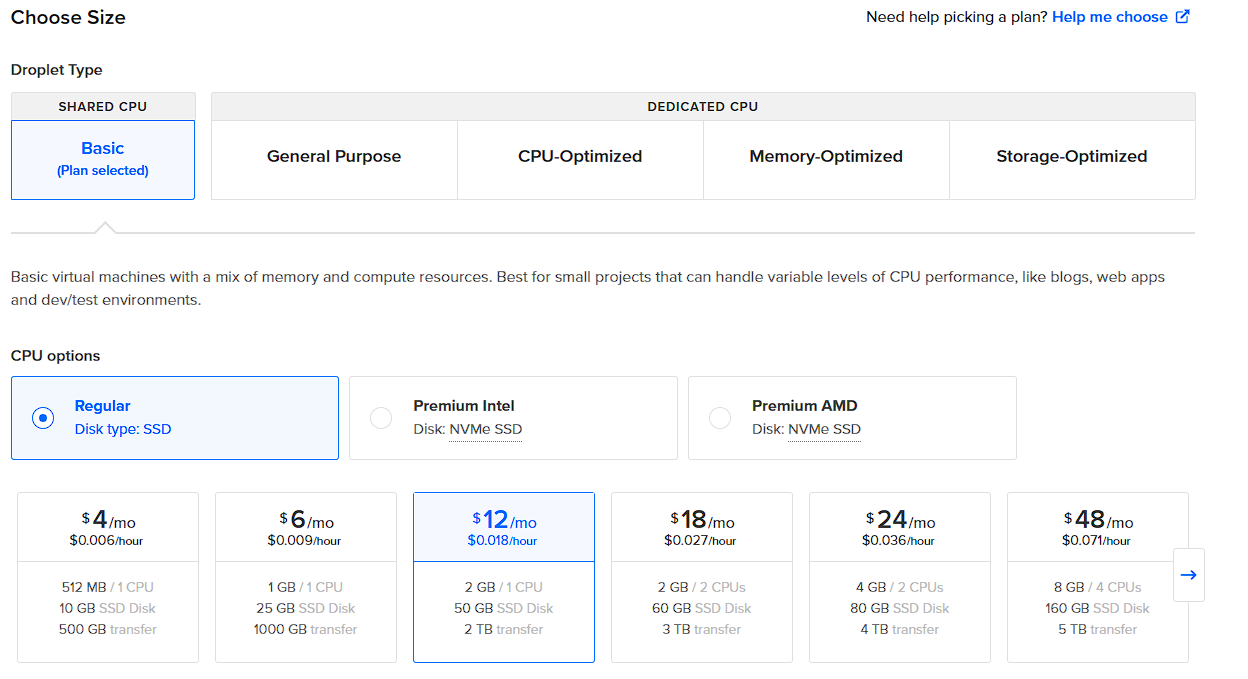
* Región: Para una mayor latencia he escogido el de Amsterdam.



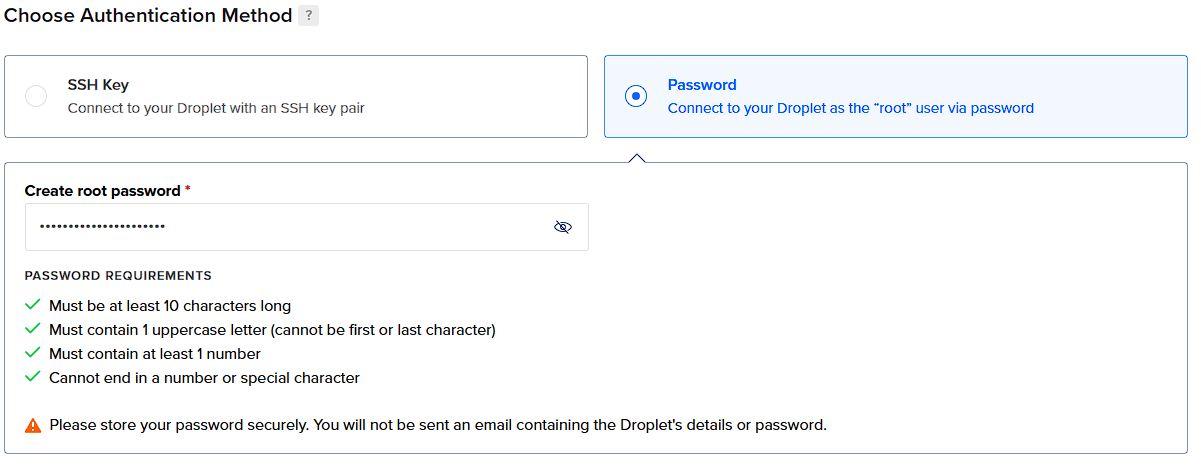
* Sistema operativo: Ubuntu 24.04 ya que es la recomendada por defecto por su estabilidad.



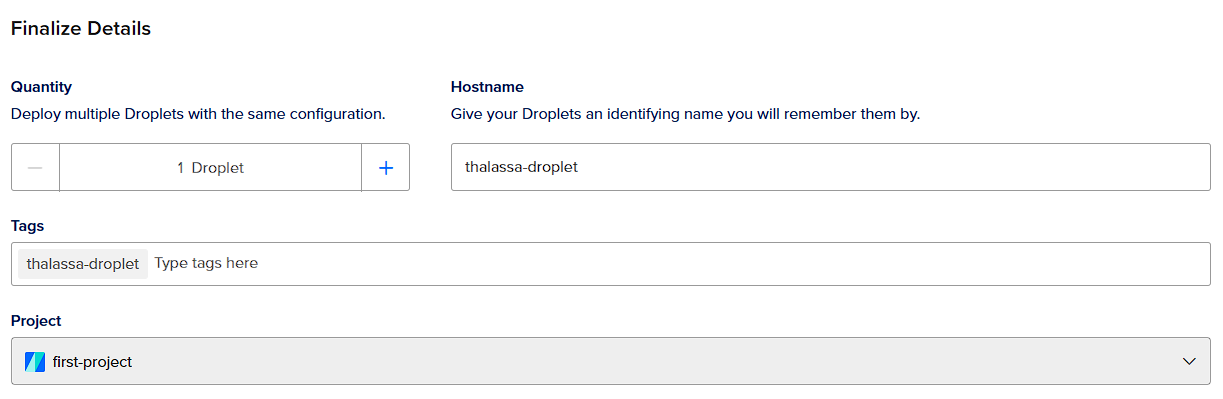
* Tamaño: Para tener suficiente RAM para ejecutar los dos contenedores a la vez de forma simultanea, he seleccionado el plan Basic con la CPU option Regular de 12€/mes, lo cual me deja con una MV de 2GB de RAM, 1 CPU y memoria SSD de 25GB. Siendo una página web muy básica, podría haber escogido otro plan mas económico, pero he querido que sobre antes de que me falte y me de problemas.



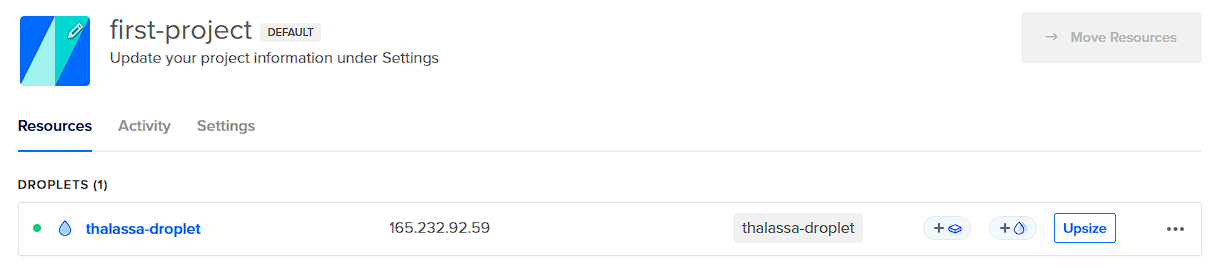
* Método de autenticación: Como método de autenticación, la mejor opción es mediante clave SSH, pero de primeras para no complicarme lo he hecho con una contraseña estable.



* Nombre del host: Para el nombre del host y del droplet le he asignado el nombre de “thalassa-droplet”.

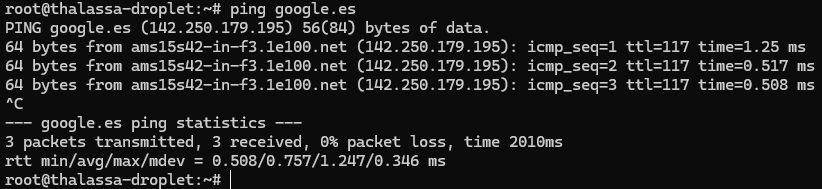


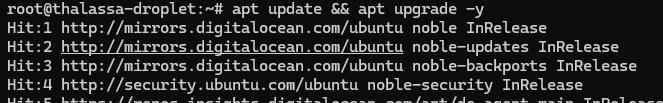
Bien, ahora que ya tengo creado el droplet, se ha generado con una IP pública, la cua deberé usar para conectarme por SSH usando la password que le he definido:



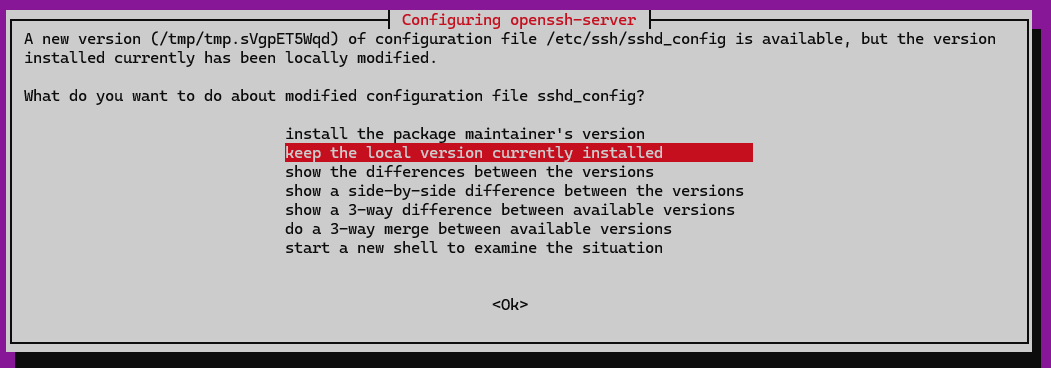
Me conecto desde la terminal por SSH, verifico conexión y actualizo los repositorios del sistema:



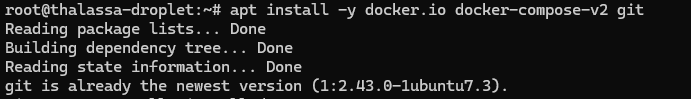




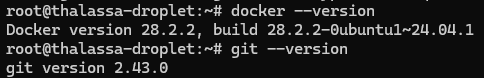
Aquí seleccionamos la opción de mantener la versión instalada actualmente para no romper la conexión del droplet:



Una vez hemos actualizado los repositorios del sistema, procedo a instalar Docker y Git en la mv:



Verifico su instalación:



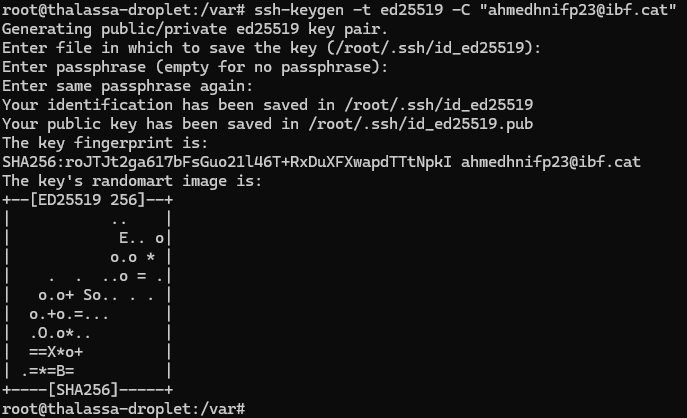
Ahora viene la parte en la cual ya creo el directorio local donde irá el repositorio de mi web y procedo a iniciar sesión en git y hacer un pull de mi repositorio en la nube. Primero creo el directorio donde irá el repositorio local:



Ahora voy a proceder a generar mi clave de acceso a git. Para ello, antes defino las variables de user.name y user.email que serán necesarias para los logs de github si hago algún cambio:



Ahora genero una llave de acceso ssh para este droplet con el siguiente comando. Le indico la ruta donde almacenar la llave y luego sigo dándole enter a todas las opciones y sin contraseña:



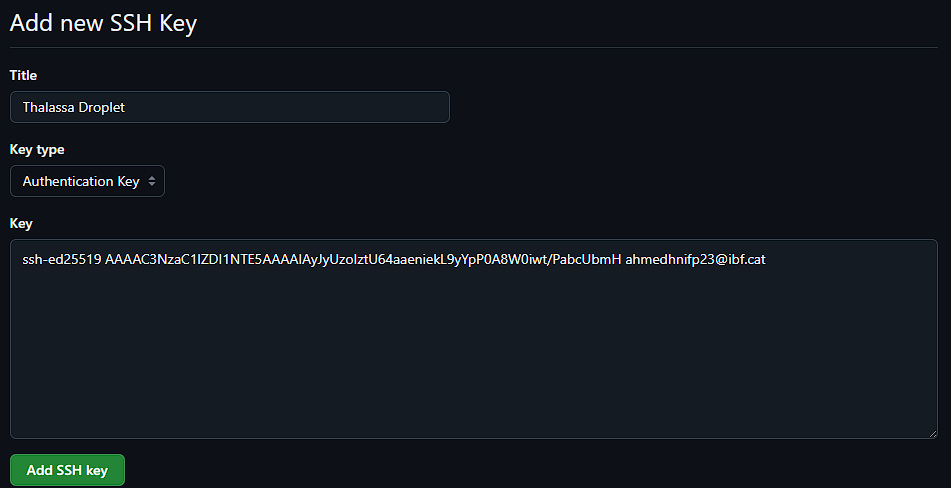
Ahora voy a abrir el archivo con la clave pública y la voy a copiar:



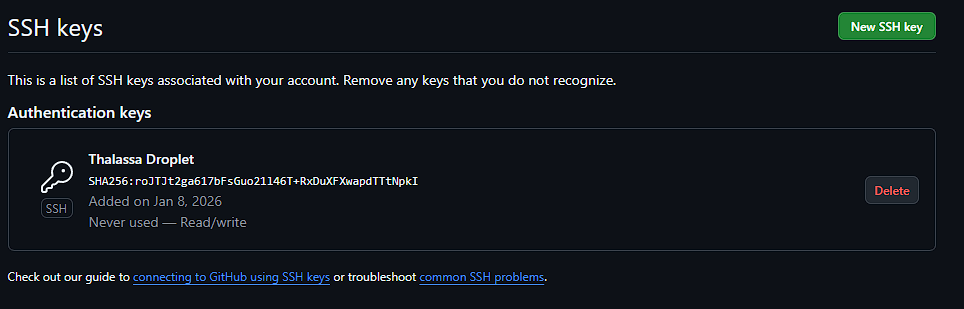
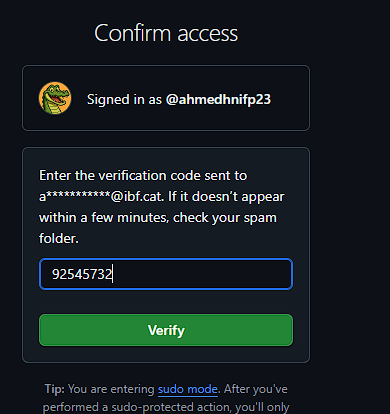
Ahora para darle autorización de acceso a mi cuenta de github, esta clave publica la voy a indicar como autorizada en github. Para ello, voy a la ventana de “Settings” -> “SSH and GPG Keys” y la doy al botón de “New SSHKey”:



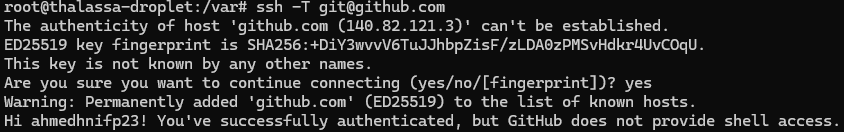
Luego le doy un nombre como por ejemplo “Thalassa Droplet”, pego la clave y le doy a añadir la clave SSH:



Confirmamos identidad y ya se habría añadido la clave correctamente:

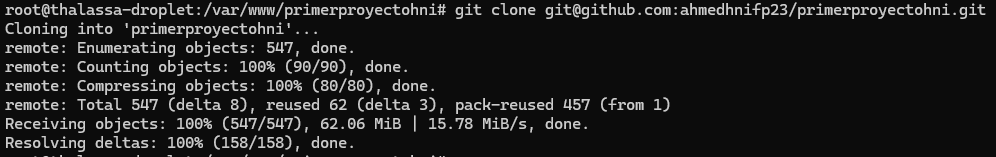


Ahora para probar la conexión, uso el comando “ssh -T [git@github.com](mailto:git@github.com)”, el cual se usa para verificar si la conexión por clave SSH con github está correctamente configurada:



Ahora finalmente ya puedo clonar mi repositorio, por lo que me dirigo al directorio que había creado antes y clono ahí el respositorio en la nube con el comando “git clone git@github.com:ahmedhnifp23/primerproyectohni.git”:







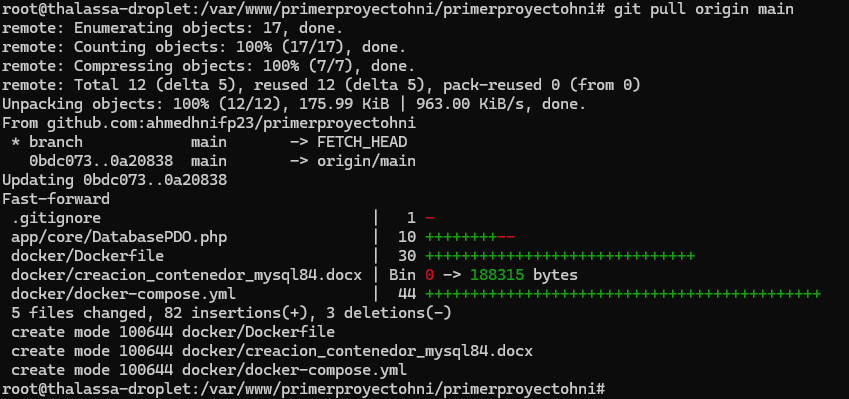
Levantar contenedores apache y mysql

Ahora que ya tengo la base, procedo a generar el archivo “Dockerfile” que se encargará de crear el contenedor de Apache con PHP 8. En este dockerfile se ejecutan varis comandos com para instalar liberías y extensiones necesarias para la conexión con mysql, modifica la carpeta raíz para que sea la “/public”, instala Composer y da permisos de lectura y escritura al usuario de apache sobre los directorios.

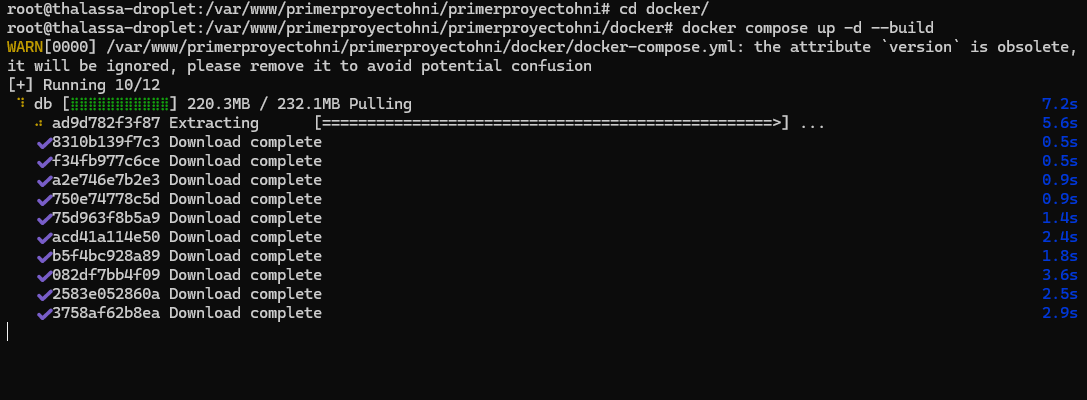
También genero el archivo “docker-compose.yml” el cual se encarga de coordinar los servicios usando primero el Dockerfile para levantar el contenedor del servidor web conectando el puerto 80 del droplet con el 80 del contenedor, pasa las credenciales a PHP como variables de entorno para luego recuperarlas en el archivo “DatabasePDO.php” que realiza la conexión a la bd. También para la persistencia de datos, crea un volumen que enlaza el directorio raíz del proyecto en el droplet con el directorio “var/www/html” del contenedor, de tal forma que si actualizo repositorio local del droplet, se actualizará también en el contenedor.

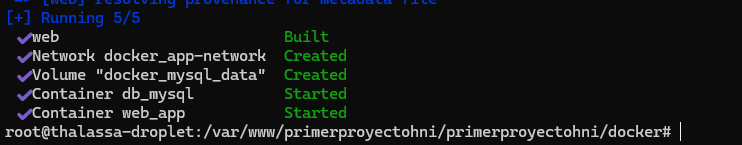
Por último, descarga la imagen de mysql:8 configurando el usuario con los usados en mi proyecto y para la persistencia de datos crea el volumen llamado “mysql\_data” en la ruta “/var/lib/mysql” y ahí es donde se almacenarán los datos de la base de datos.

Actualizo los cambios en el repositorio en la nube y hago un pull de los cambios en el droplet:

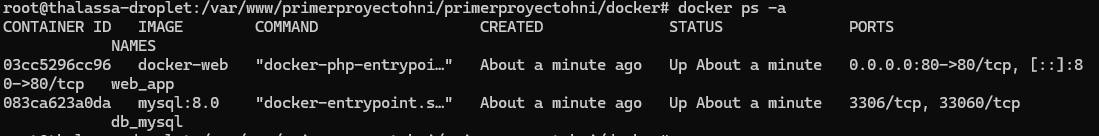


Ahora desde el directorio donde tengo los archivos “Dockerfile” y “Docker-compose.yml”, ejecuto el comando “docker compose up -d –build” para levantar el servicio y crear los contenedores:

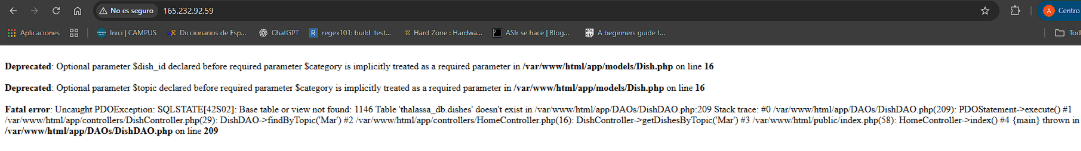




Una vez finalizado, procedo a verificar que se han creado correctamente los contenedores y que están activos:



Ahora ya podía acceder a la IP de mi droplet <http://165.232.92.59/> para visualizar mi web y verificar que los contenedores con los servicios habían sido generados correctamente:



Crear las tablas y insertar datos

En este punto, me faltaba crear las tablas e insertar datos, puesto que ahora mismo la base de datos estaba vacía y por eso al acceder a mi web me saltaba el error de la imagen anterior. Para ello, subí al repositorio en la nube un directorio llamado “database” en el cual tenía los scripts DDL y DML para generar las tablas, los triggers y los campos, así como alguna modificación que hice. Una vez teniendo el este directorio en el repositorio local de mi droplet, ejecute el comando “docker exec -i db\_mysql mysql -u root -p'Asdqwe!23' < database/DDLscript\_create\_thalassaDB.sql” para meterme en la terminal del contenedor mysql, iniciar sesión en la base de datos y ejecutar ese script:



Una vez se habían creado las tablas, procedí a recargar mi web <http://165.232.92.59/> y ya se estaba visualizando correctamente y en este punto solo me faltaba generar el usuario admin y empezar a cargar todos los platos así como editar las imágenes:

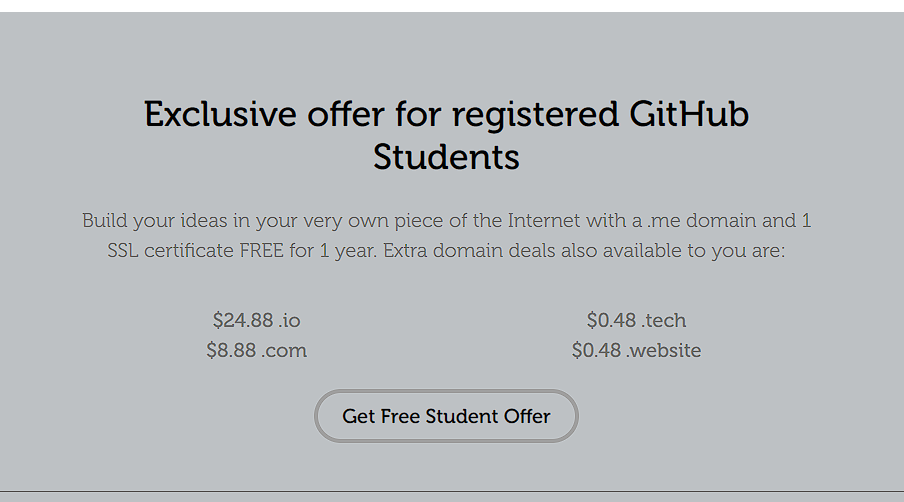


Adquirir un dominio y asignarlo

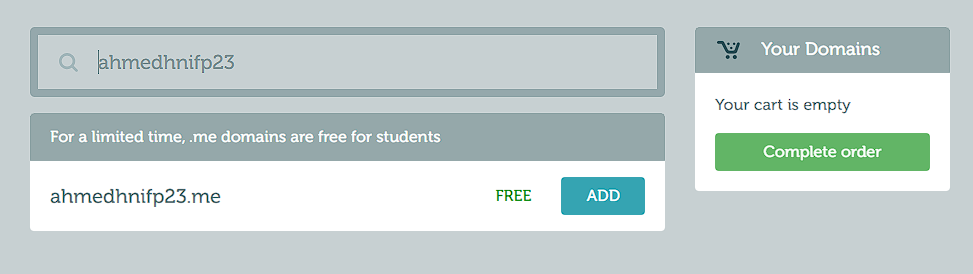
Para el tema del dominio, también vi que con el plan de estudiante de Github, Namecheap ofrecía 1 año de uso gratuito de un dominio con extensión “.me”, por lo que procedí a canjear la oferta:

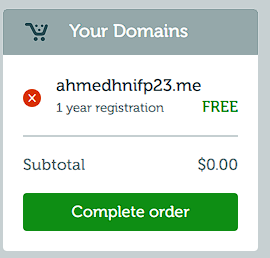
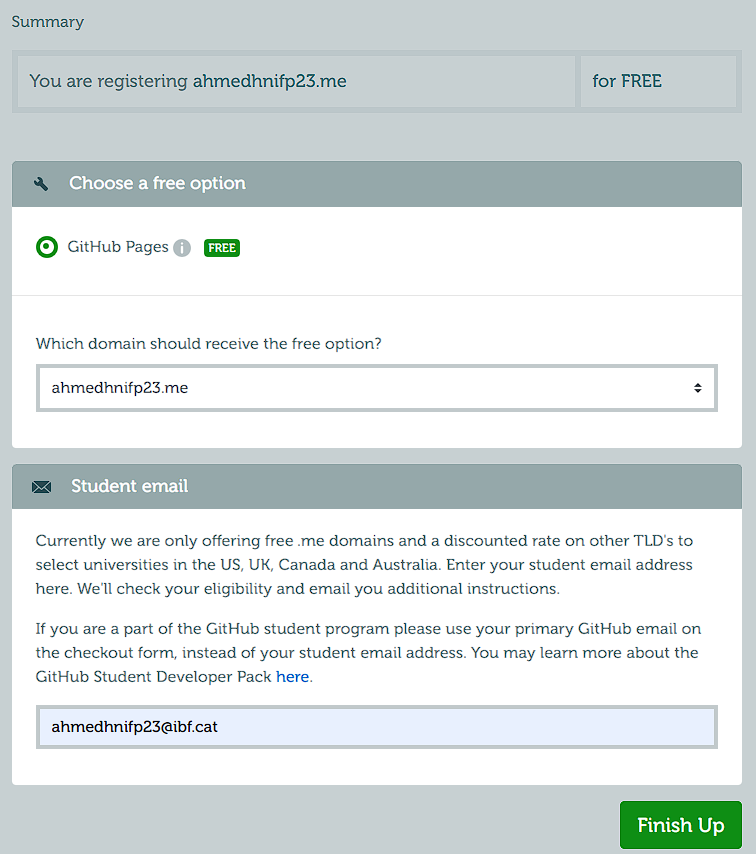
<https://youtu.be/0nT8Aj-m1xk>

<https://nc.me/landing/github>



Verifiqué que el dominio que me interesaba de “ahmedhnifp23.me” estaba disponible:

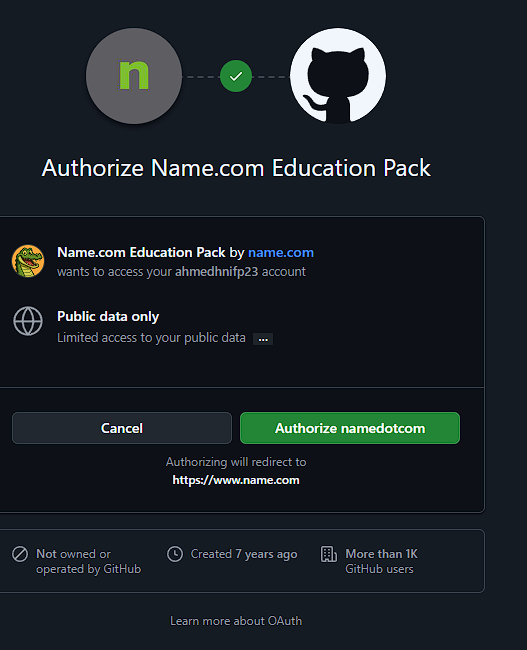


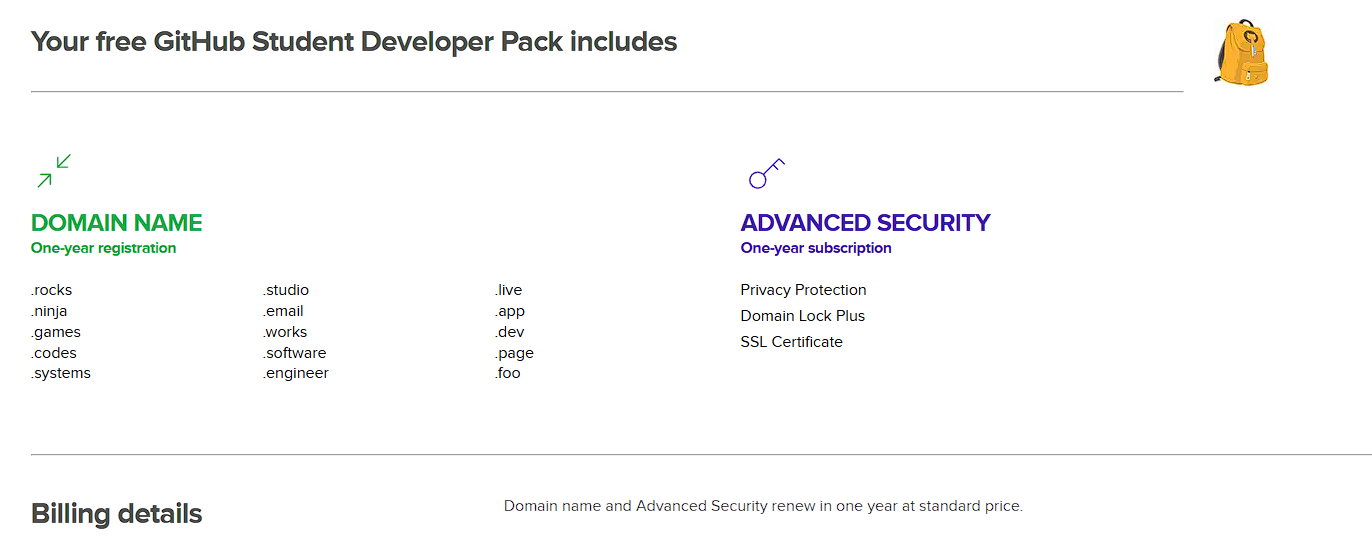
Me pide verificación, or lo que ahora prueba con la web Name que también son partners con github:

<https://www.name.com/partner/github-students>

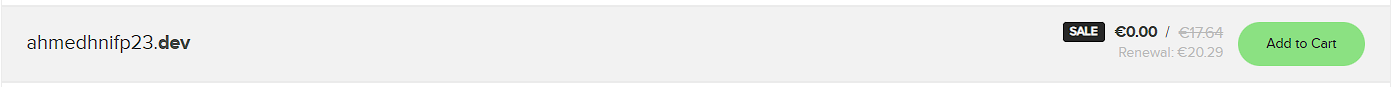
Inicio sesión con mi cuenta de github:

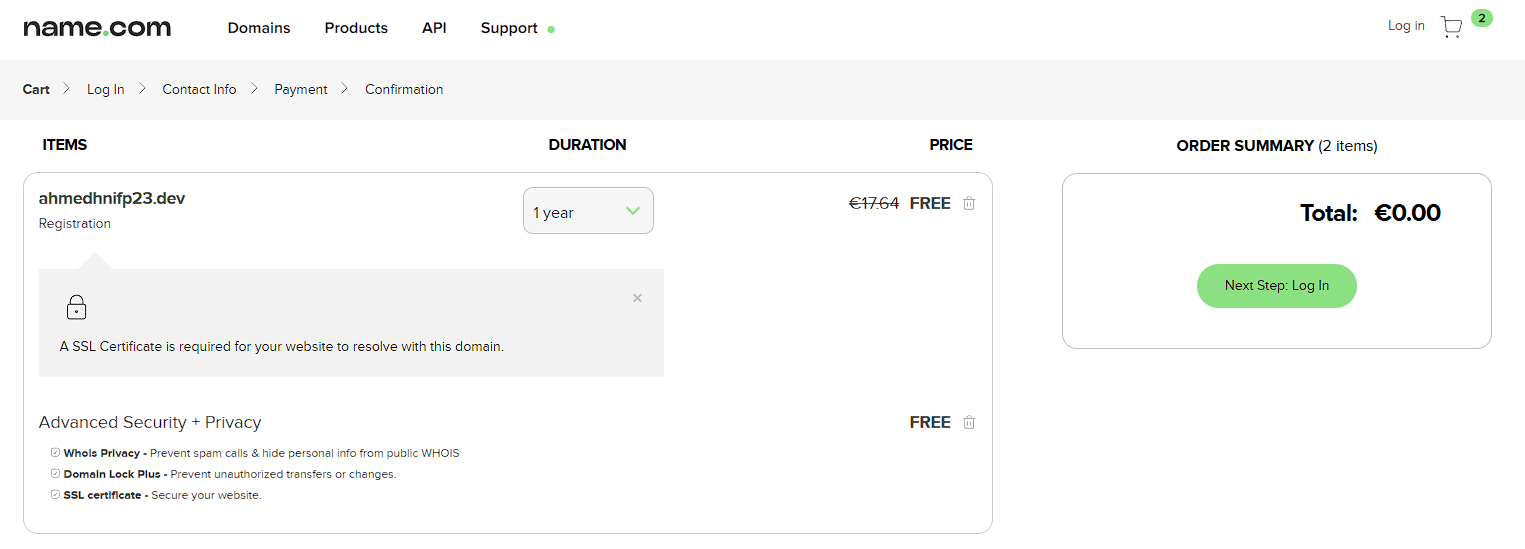


Aquí me indica las extensiones de dominio permitidas gratuitas:

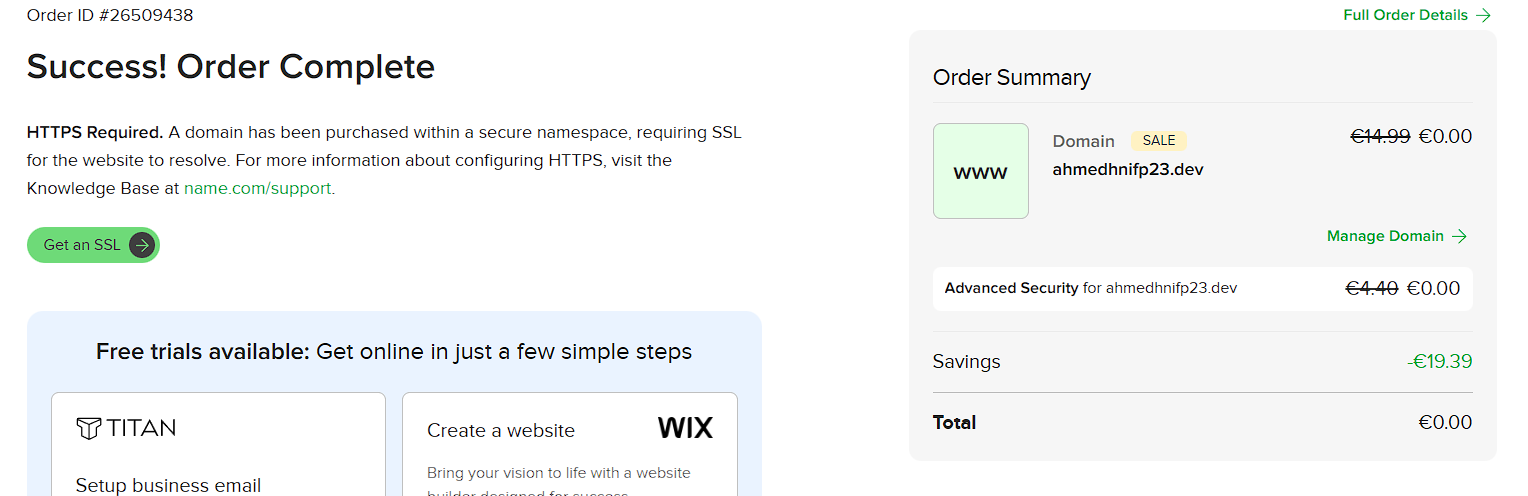


Elijo el domino “ahmedhnifp23” con la extensión “.dev”:

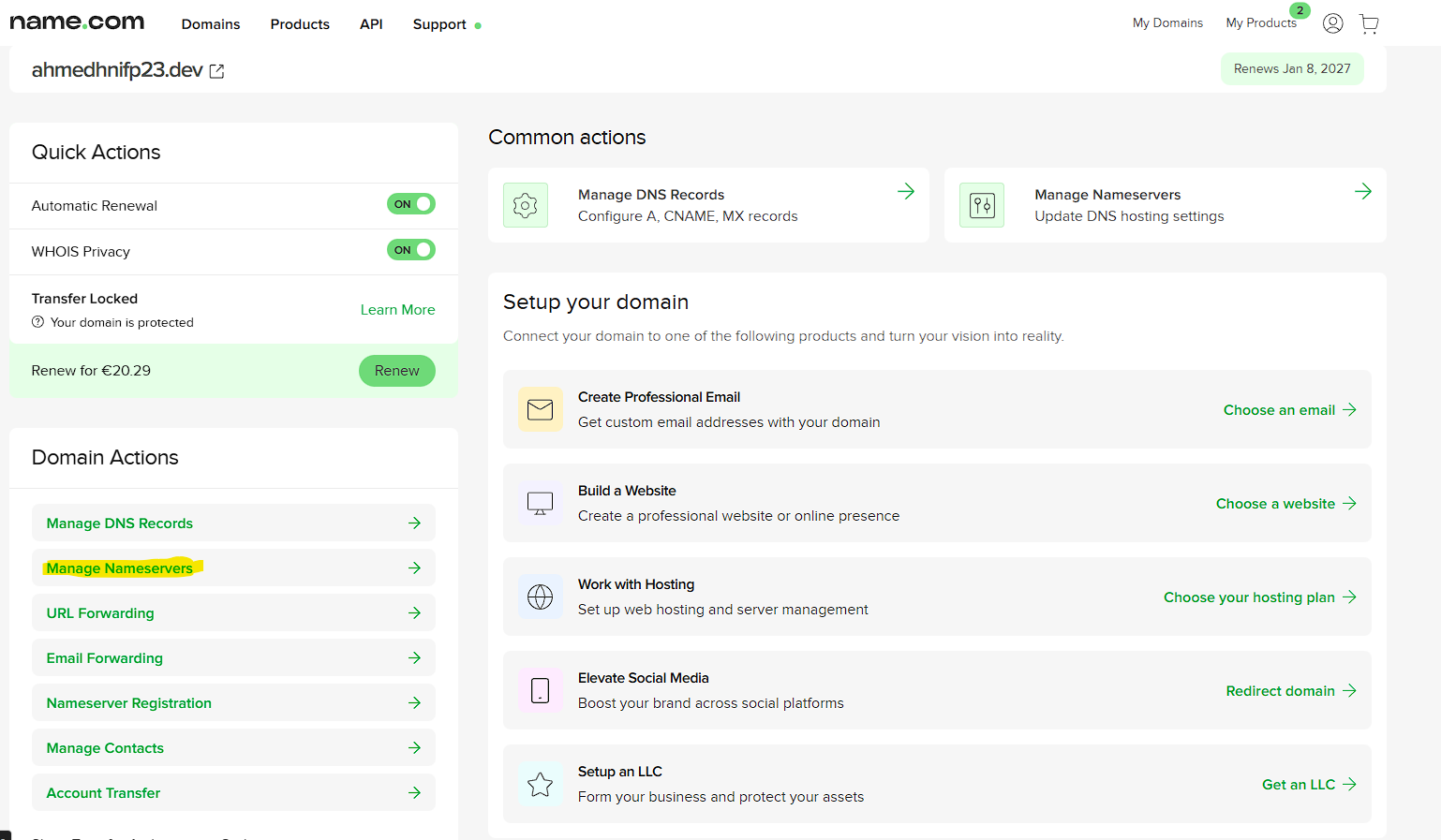




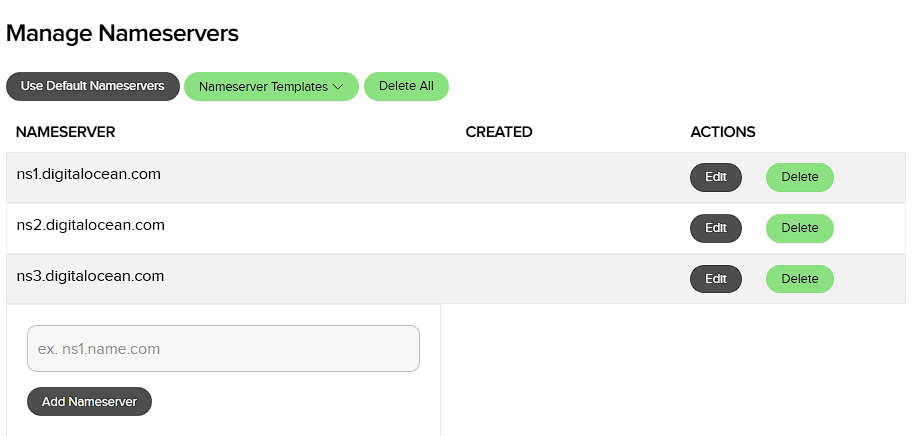
Una vez creada una cuenta en name.com y verificados los datos como mail y método de pago(sin cobro), ya tenía mi dominio:



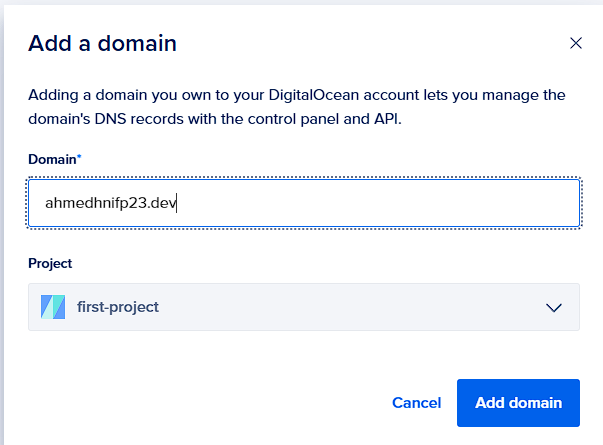
Ahora voy a la gestión del dominio y a configurarle el nameserver de DigitalOcean para que sea el DNS de ellos quien haga la traducción:



Y añado los siguientes nameservers y guardo los cambios:

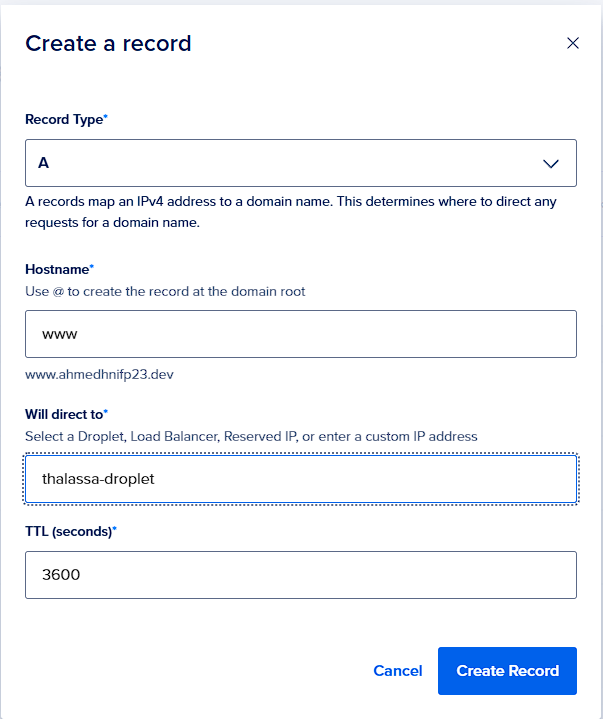


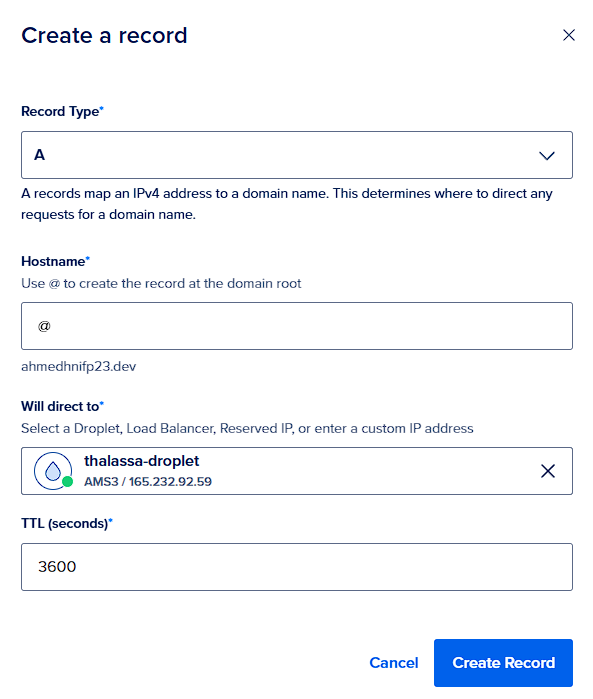
Ahora voy a la sección de “Networking” en DigitalOcean y voy a añadir el dominio:



Ahora necesito crear los registros de tipo A que redirigan al droplet:







El cambio tarda un rato en propagarse:

