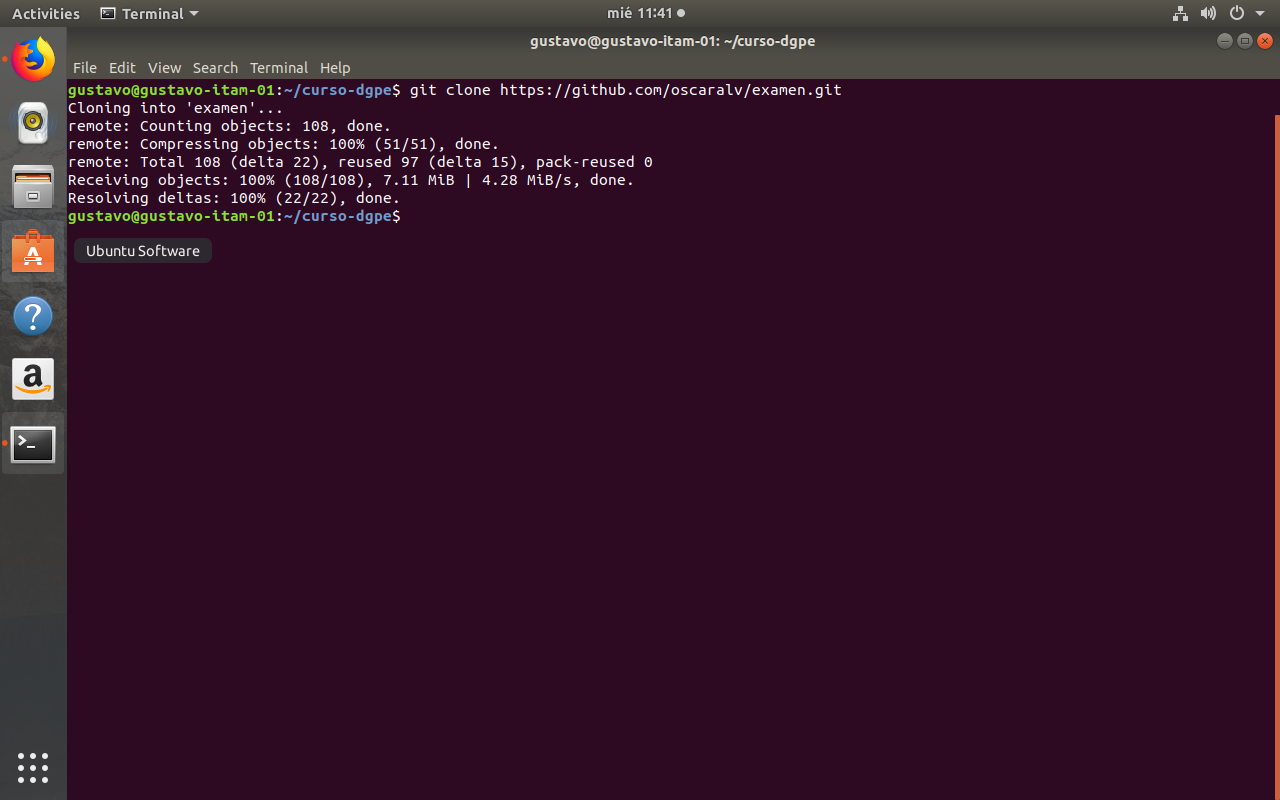
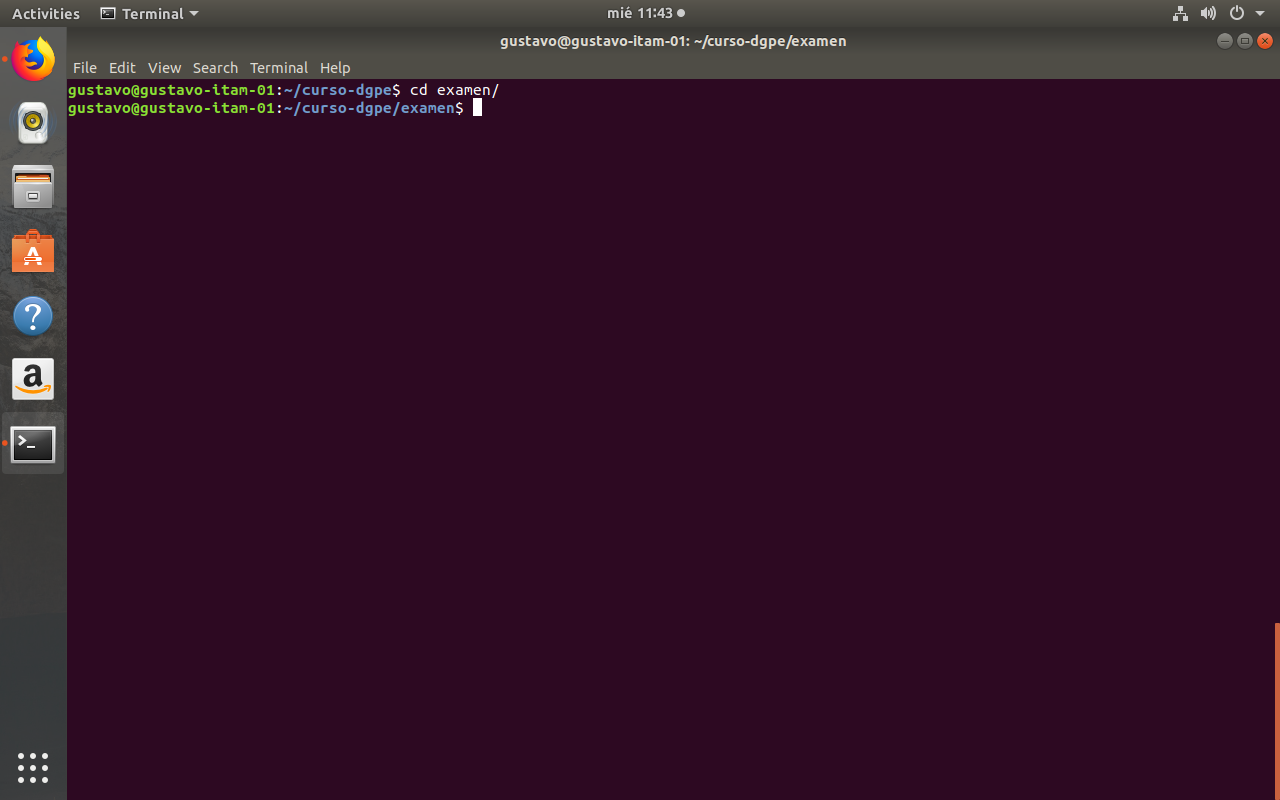
CREACIÓN Y DESPLIEGUE DE UNA VERTICAL EN UN SERVIDOR RANCHER

CREACIÓN DE LA VERTICAL CON VERTX

Para crear la vertical con el servicio REST que queremos levantar en el servidor Rancher primero debemos clonar de GitHub el proyecto “examen” en la URL <https://github.com/oscaralv/examen.git>

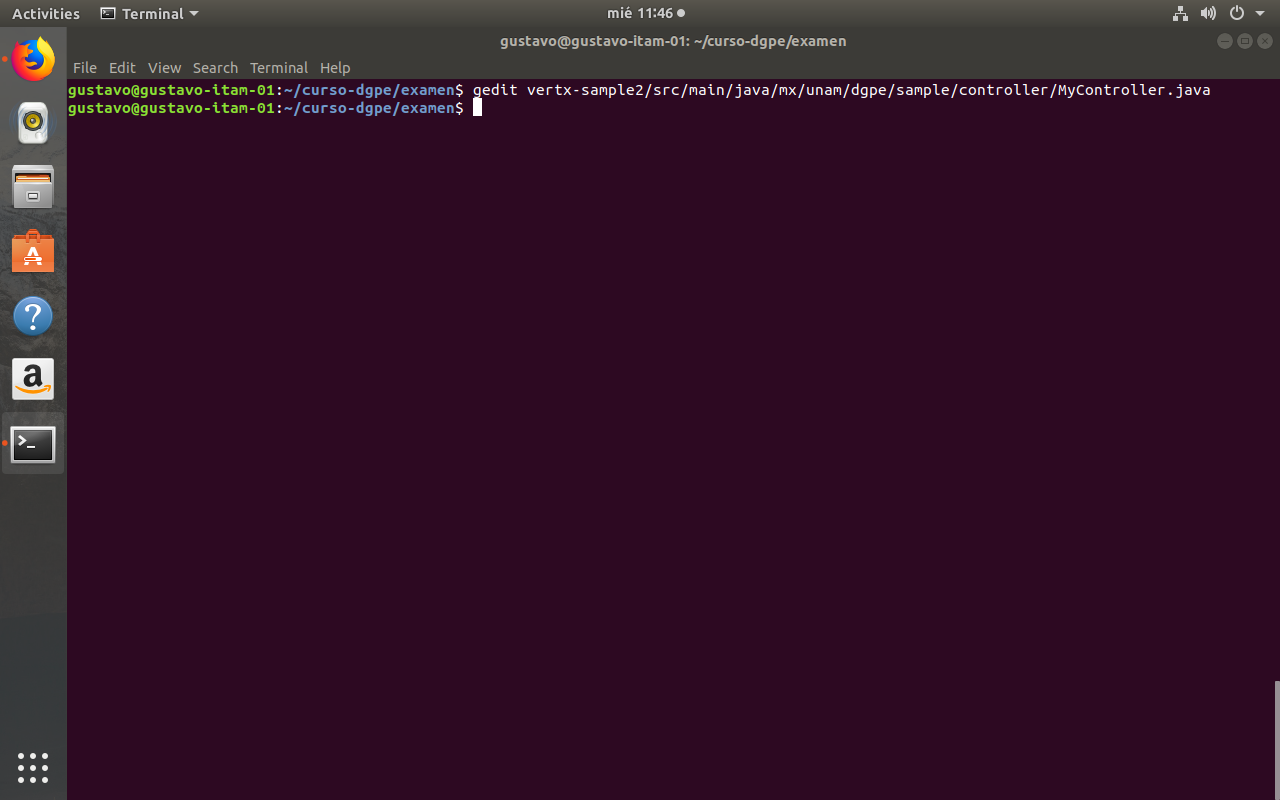


Una vez clonado el proyecto debemos posicionarnos en el directorio creado con el comando cd examen/

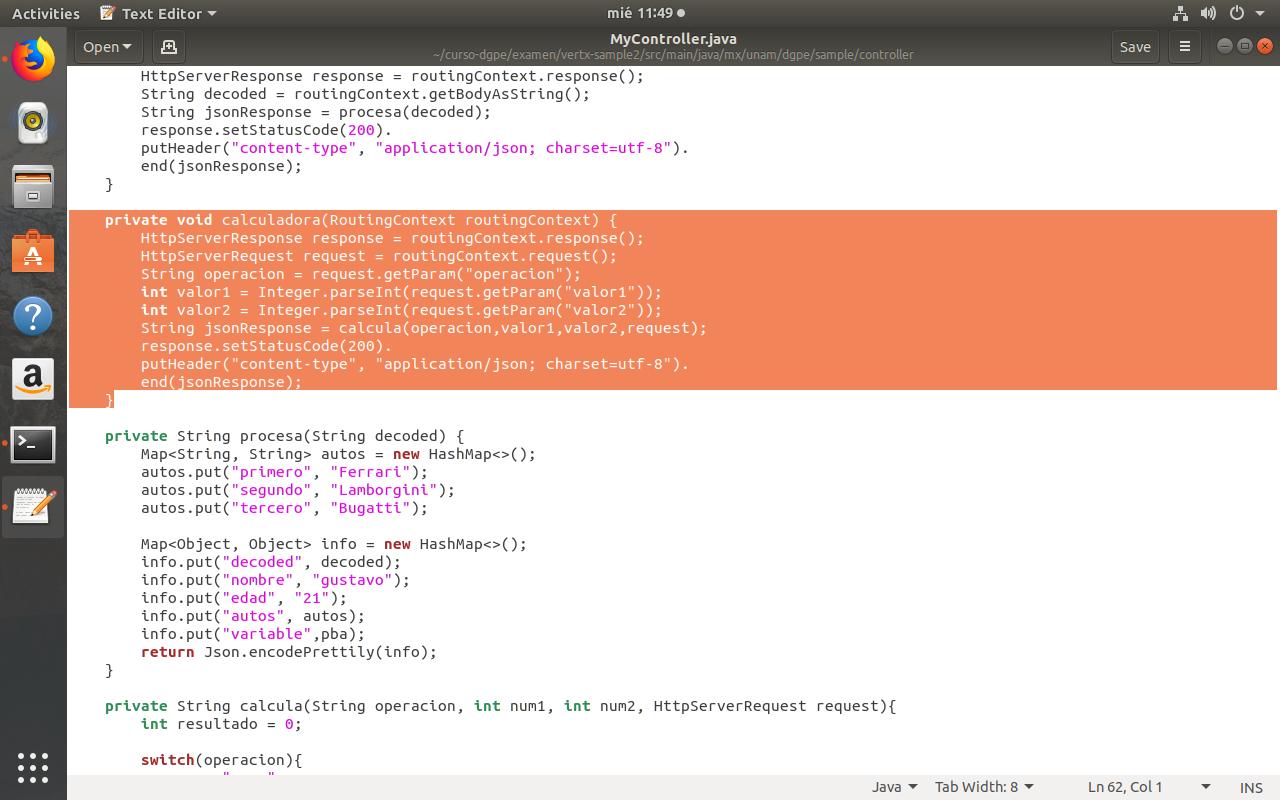


Después de posicionarnos en el directorio debemos modificar el archivo MyController.java ubicado en la ruta:

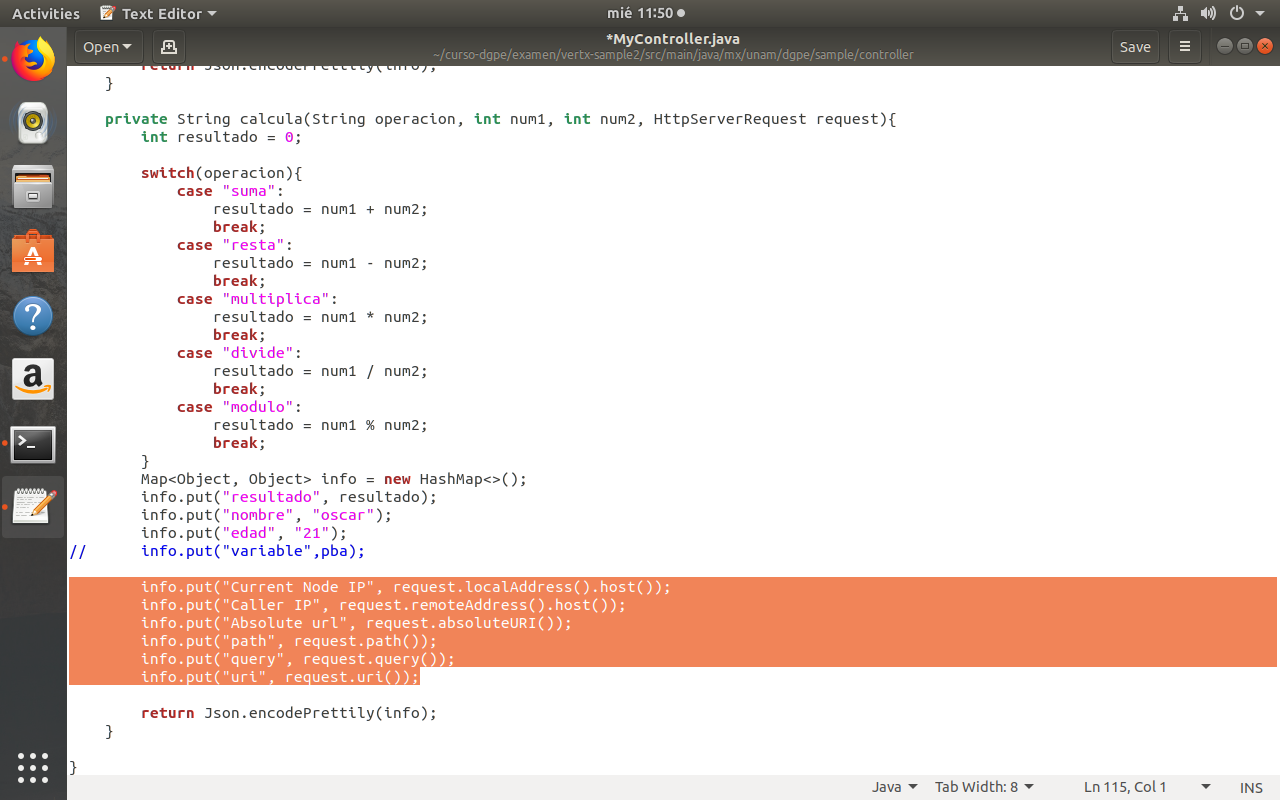
**vertx-sample2/src/main/java/mx/unam/dgpe/sample/controller/MyController.java**



Una vez abierto el archivo, primero hay que modificar el método “**calculadora**” en la invocación al método “**calcula**”, agregándole como parámetro extra el **request**.



Modificamos el método “**calcula**” para que reciba el mismo parámetro y podamos usarlo para mostrar la IP donde se encuentra corriendo el servicio. Esta información nos servirá más adelante para verificar que el balanceador de cargas en el servidor Rancher está funcionando correctamente.



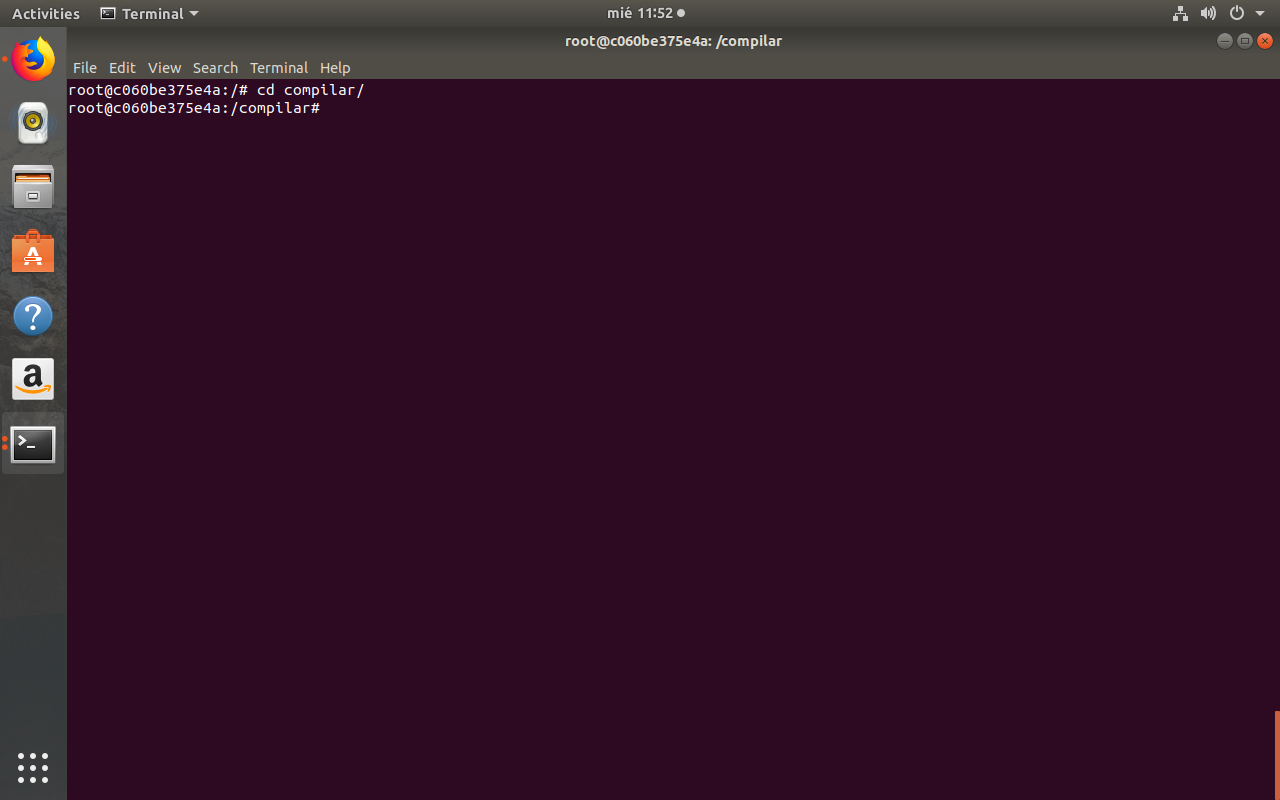
Guardamos los cambios y cerramos el archivo.

Para poder compilarlo necesitamos usar la imagen Docker **kebblar/jdk18-utf8-debug-maven**. Agregamos un volumen y lo invocamos de la siguiente manera.

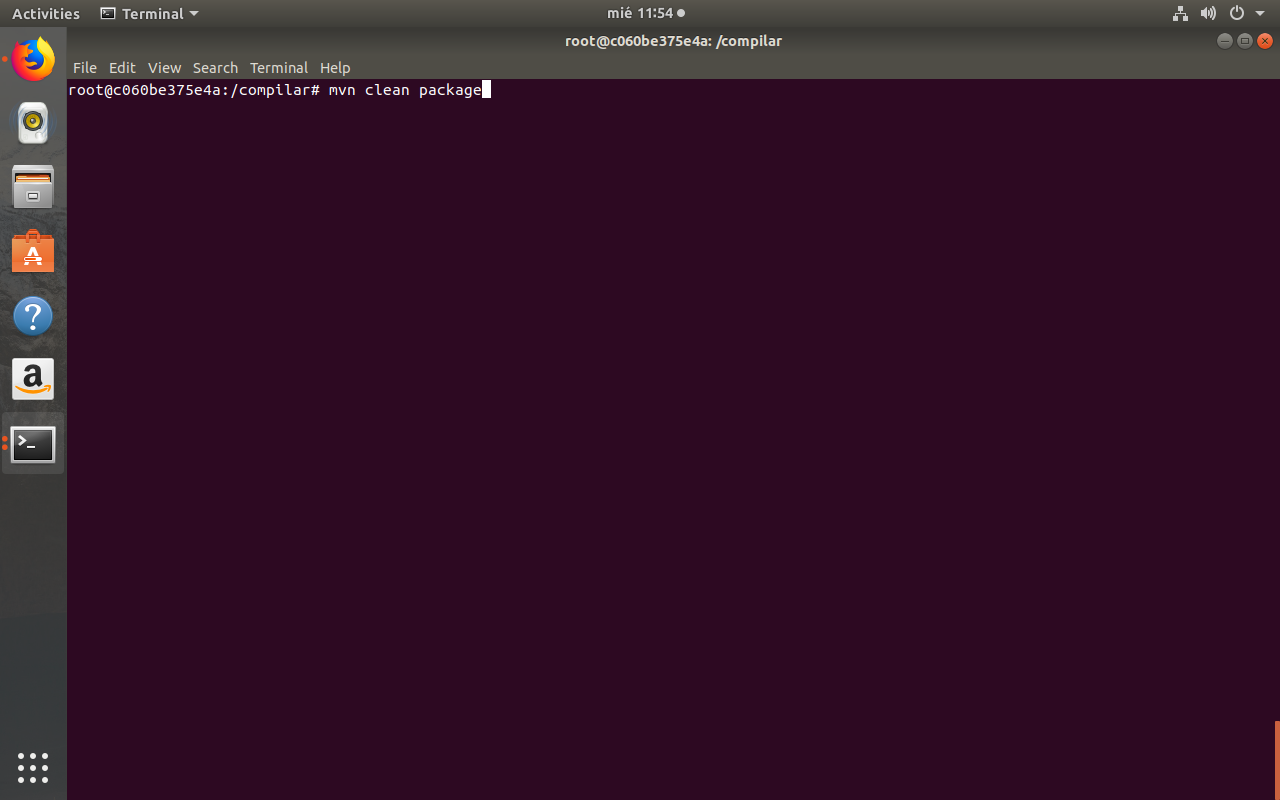
**docker run -it -v /home/gustavo/curso-dgp/examen/vertex-sample2:/compilar kebblar/jdk18-utf8-debug-maven bash**



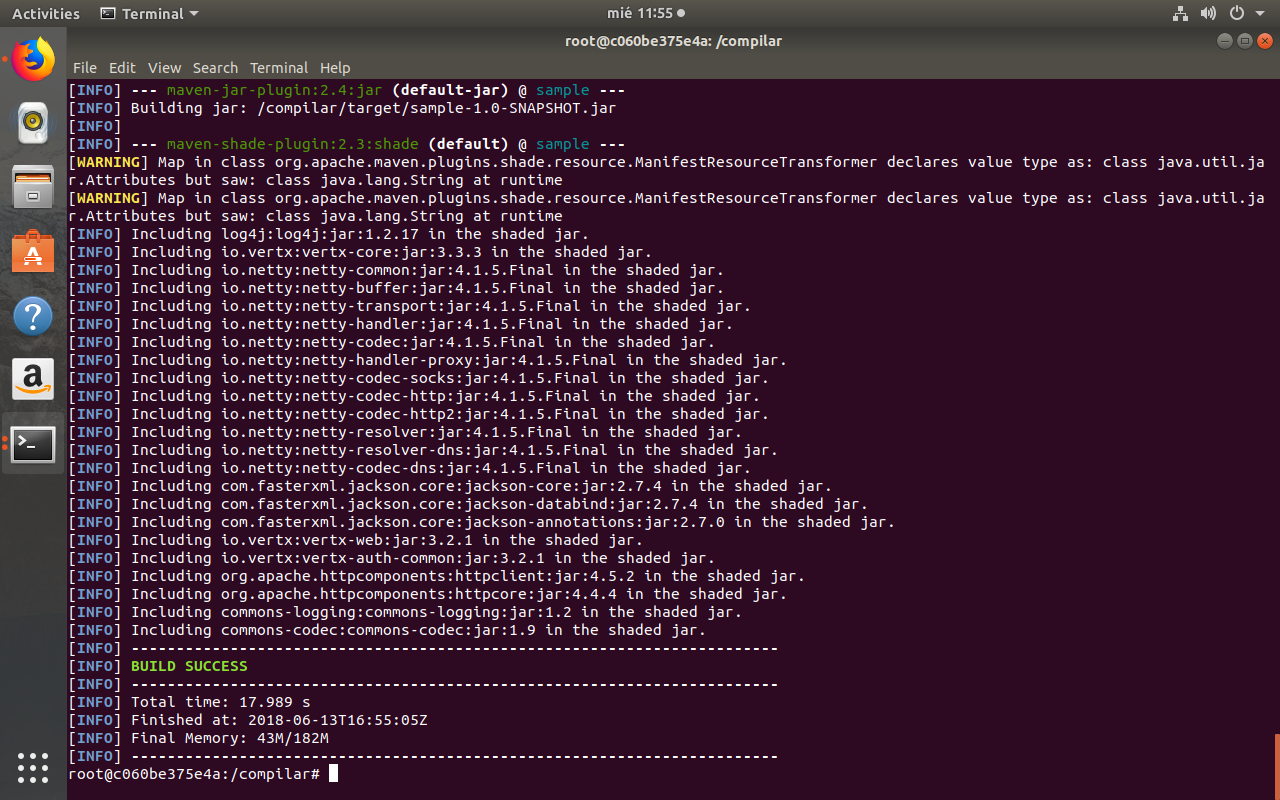
Después de ejecutarlo nos muestra el prompt que indica que estamos dentro del Docker. Nos posicionamos en el directorio que creamos con el volumen en el comando anterior.



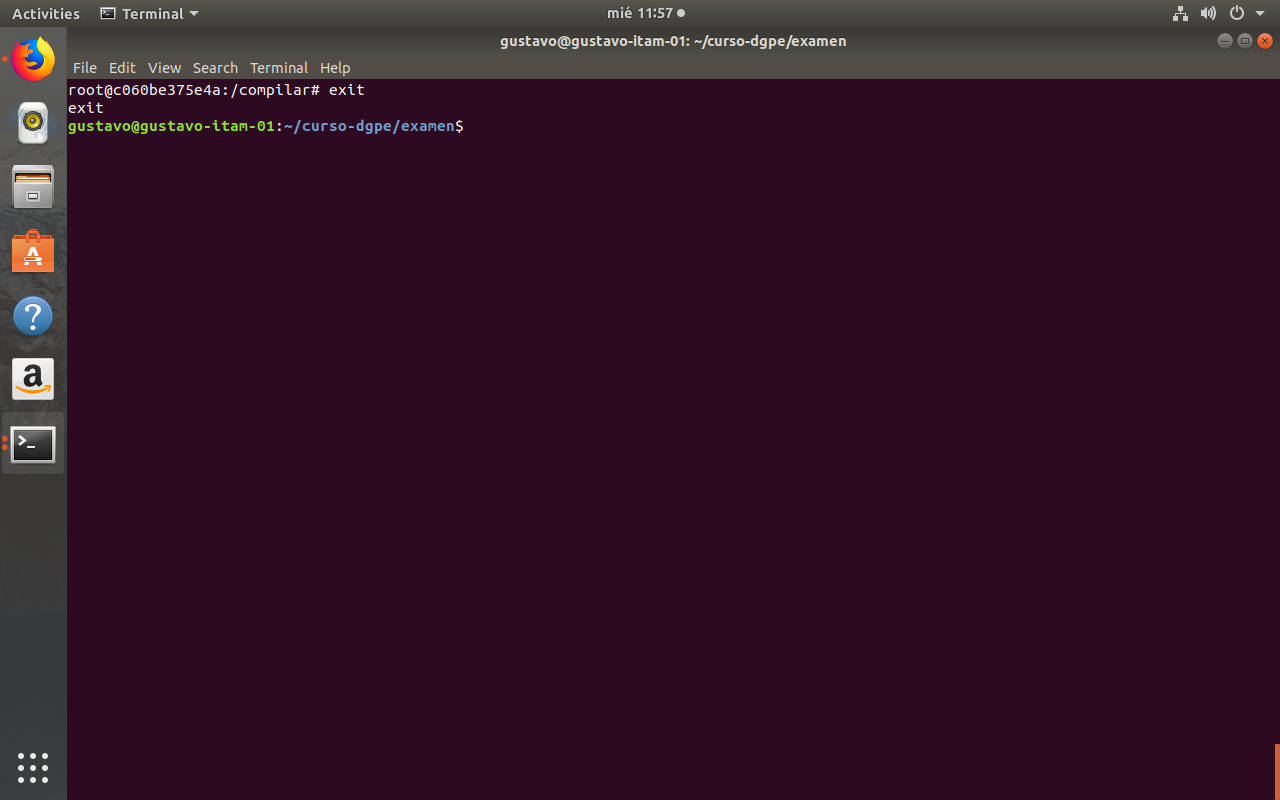
Dentro del directorio ejecutamos la instrucción **mvn clean package** para compilar nuestro proyecto de la vertical.



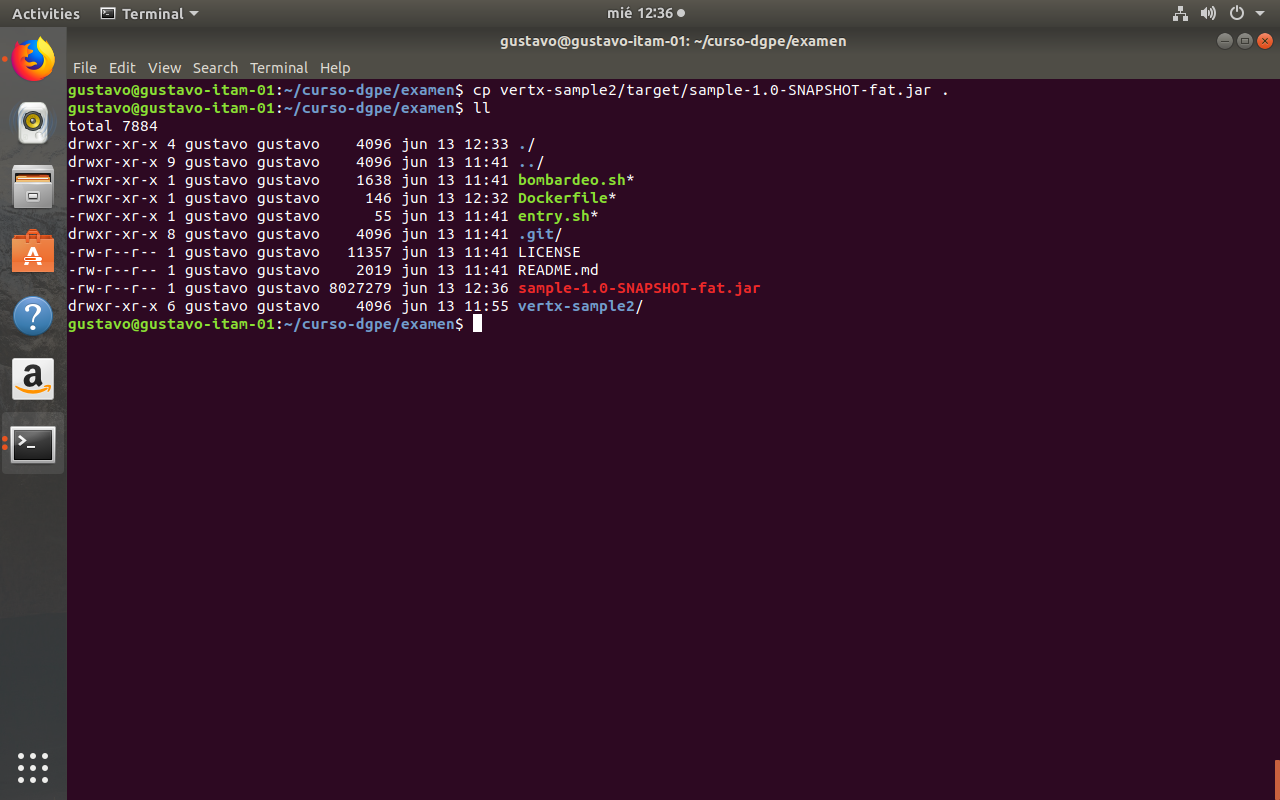
Podemos observar que la compilación se realizó correctamente.



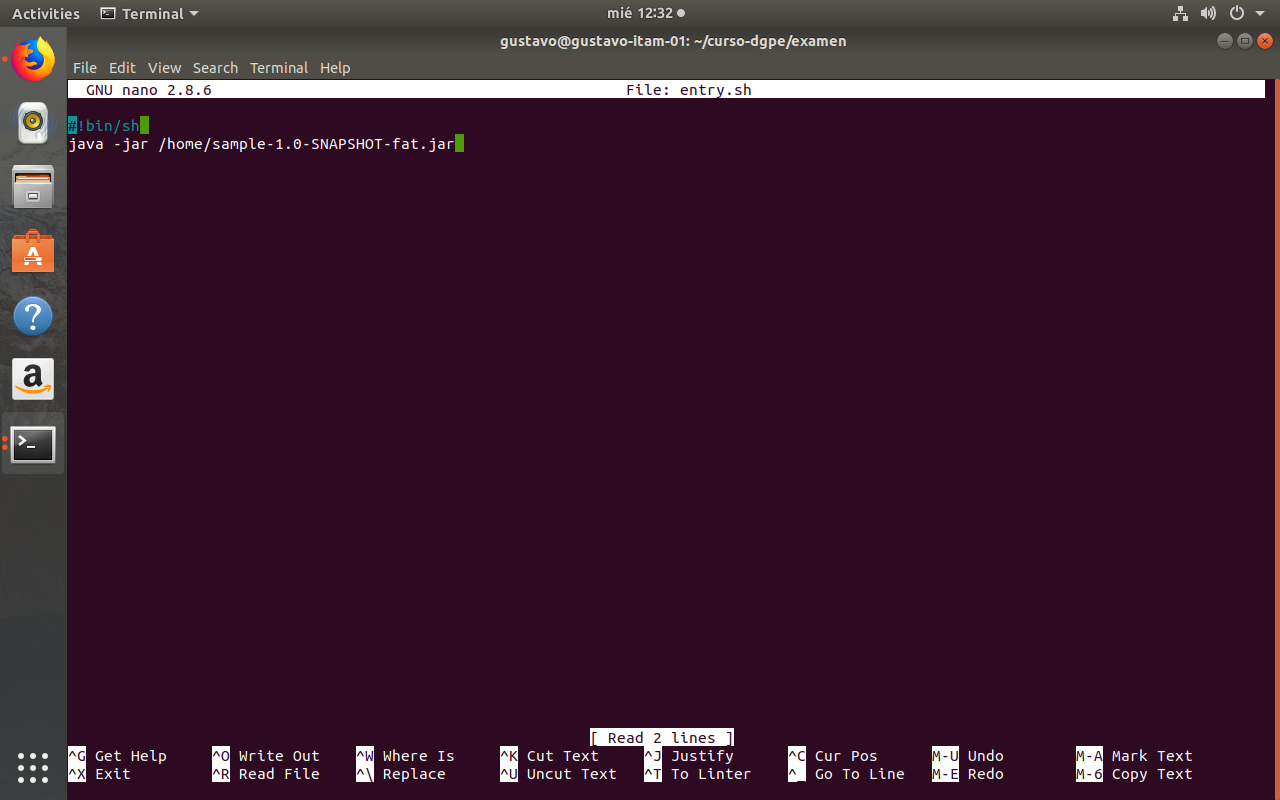
Con el comando **exit** salimos del contenedor Docker.



La compilación nos creó un directorio llamado **target** dentro del directorio **vertx-sample2**. Dentro de este directorio (**target**) fue generado un archivo **jar** llamado **sample-1.0-SNAPSHOT-fat.jar**. Copiamos ese archivo a la raíz de nuestro proyecto “**examen”** para poder usarlo más adelante con el archivo **Dockerfile**.

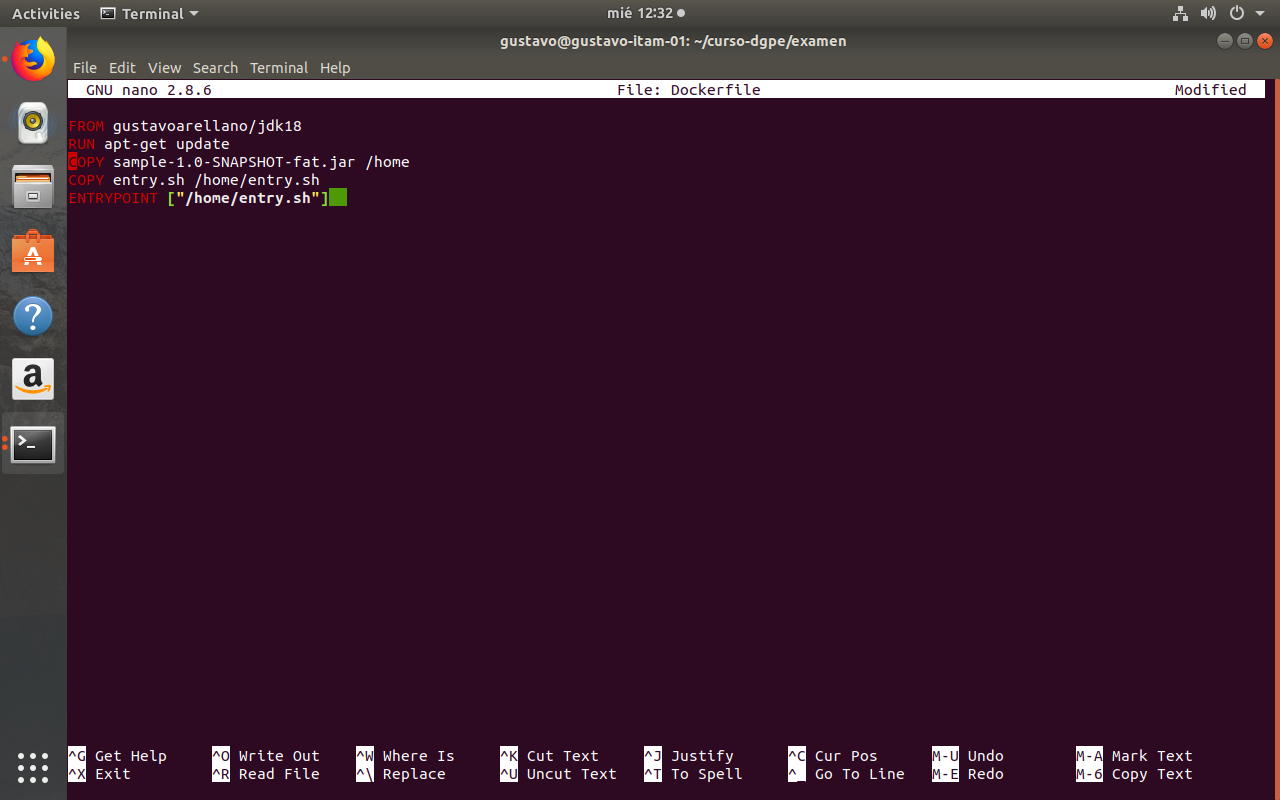


Creamos un archivo llamado “**entry.sh”** que será usado mas adelante por el **Dockerfile**. Este archivo contiene instrucciones que se ejecutaran cuando la imagen creada sea inicializada. No antes.

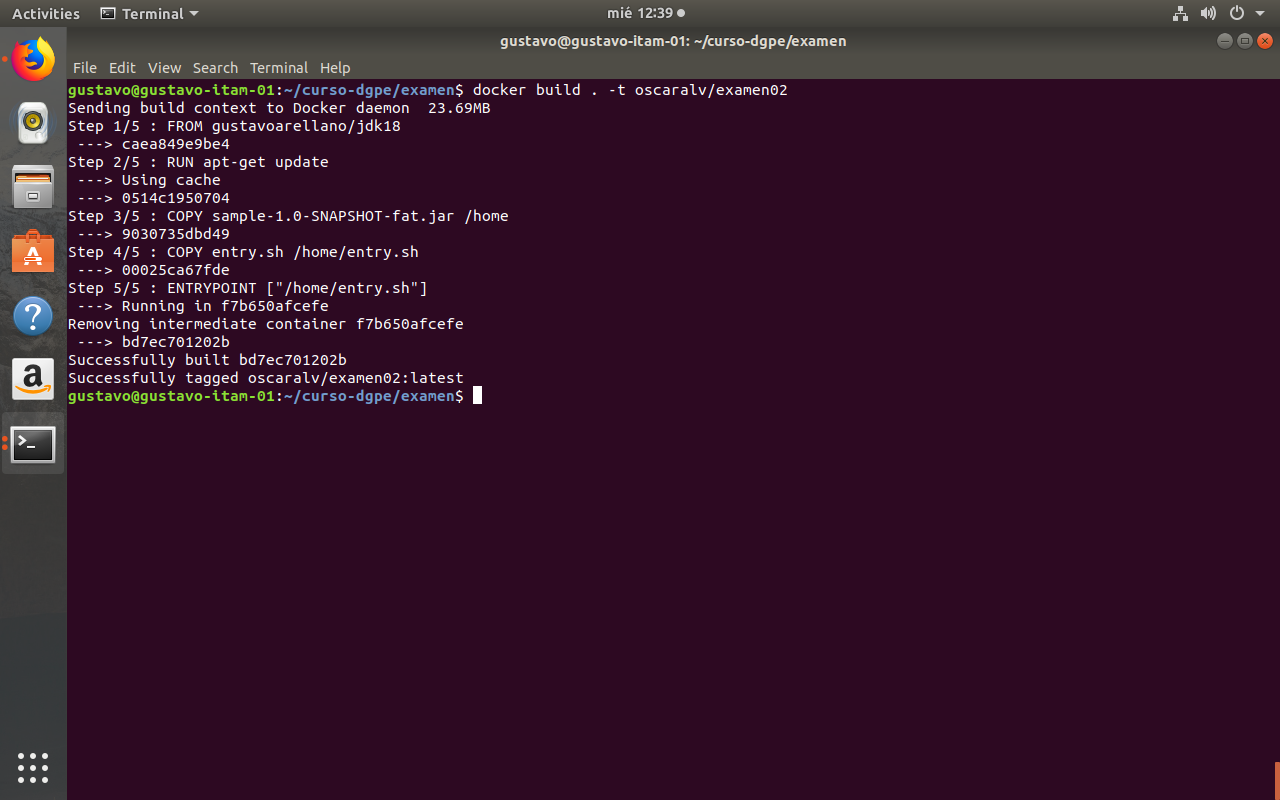


Creamos un archivo llamado **Dockerfile** con las siguientes instrucciones:

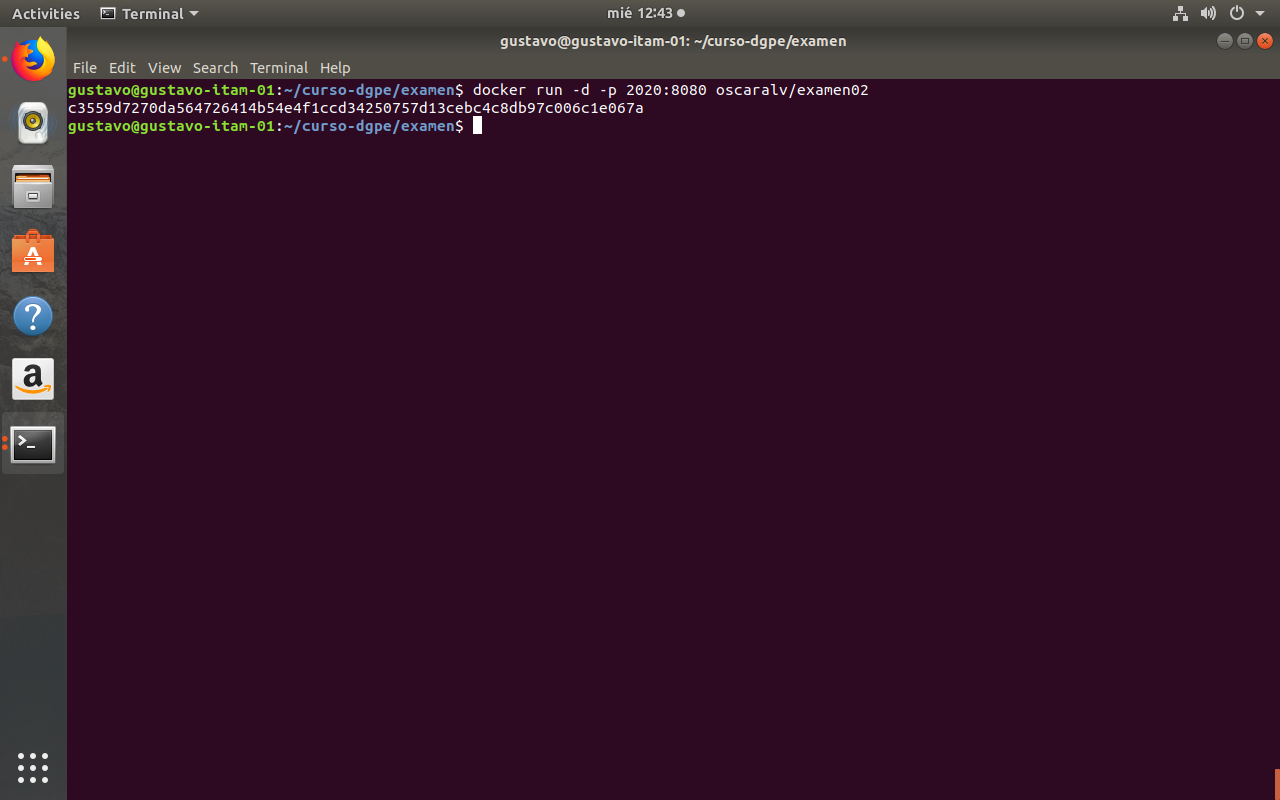
|  |  |
| --- | --- |
| INSTRUCCIÓN | ACCIÓN |
| FROM gustavoarellano/jdk18 | Nos indica que imagen tomaremos como base para crear la nuestra. |
| RUN apt-get update | Ejecuta el comando de actualización de las dependencias del SO. |
| COPY sample-1.0-SNAPSHOT-fat.jar /home | Se copia el archivo jar que creamos al directorio /home de nuestra imagen que se va crear. |
| COPY entry.sh /home/entry.sh | Copiamos el archivo entry.sh creado anteriormente al directorio /home de nuestra imagen que se va crear. |
| ENTRYPOINT [“/home/entry.sh”] | Ejecuta las instrucciones en el archivo entry.sh copiado anteriormente. |



Con los archivos listos ejecutamos el comando **docker build . -t oscaralv/examen02** para crear la imagen nueva llamada “**oscaralv/examen02**” con la información contenida en el archivo **Dockerfile**.

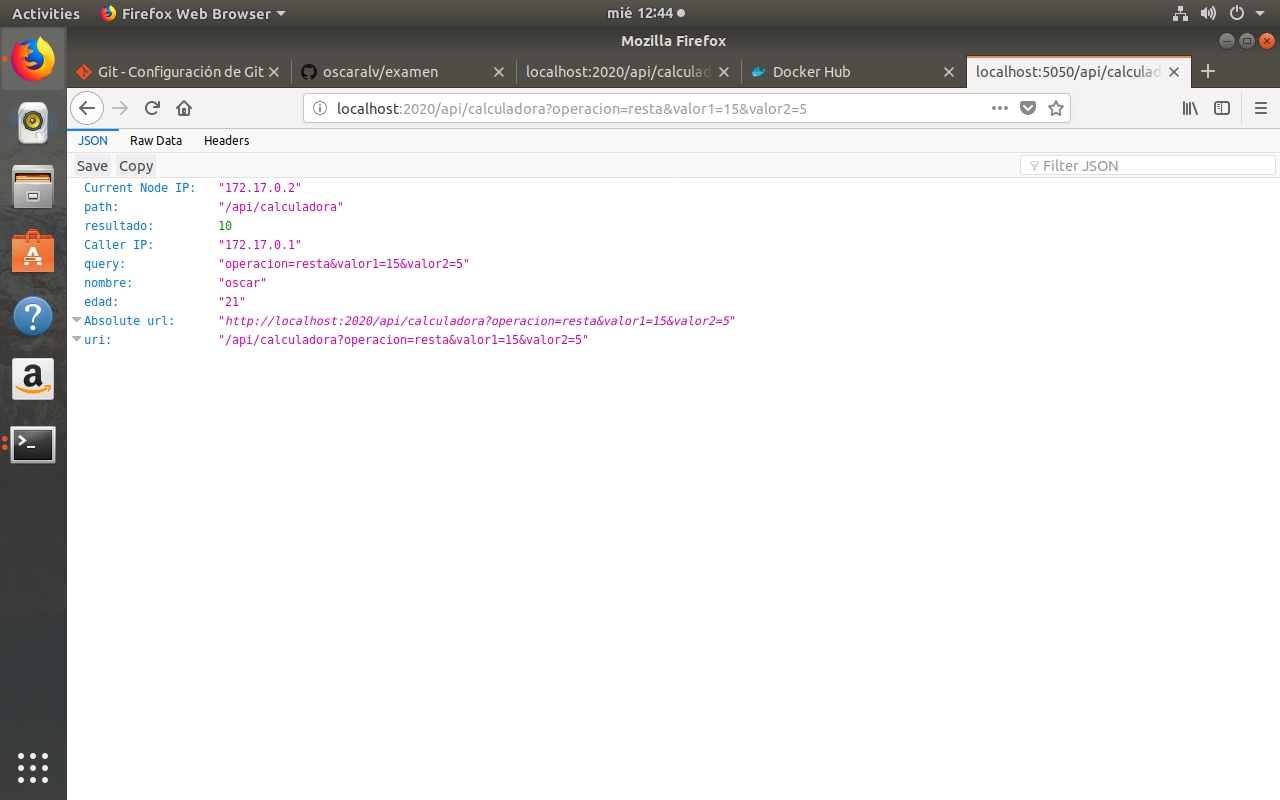


Levantamos un contenedor Docker con la imagen ya creada para poder verificar que funcione adecuadamente.



En una ventana del navegador llamamos a nuestra vertical con: **localhost:2020/api/calculadora?operacion=suma&valor1=15&valor2=5**

Observamos que la información es correcta.



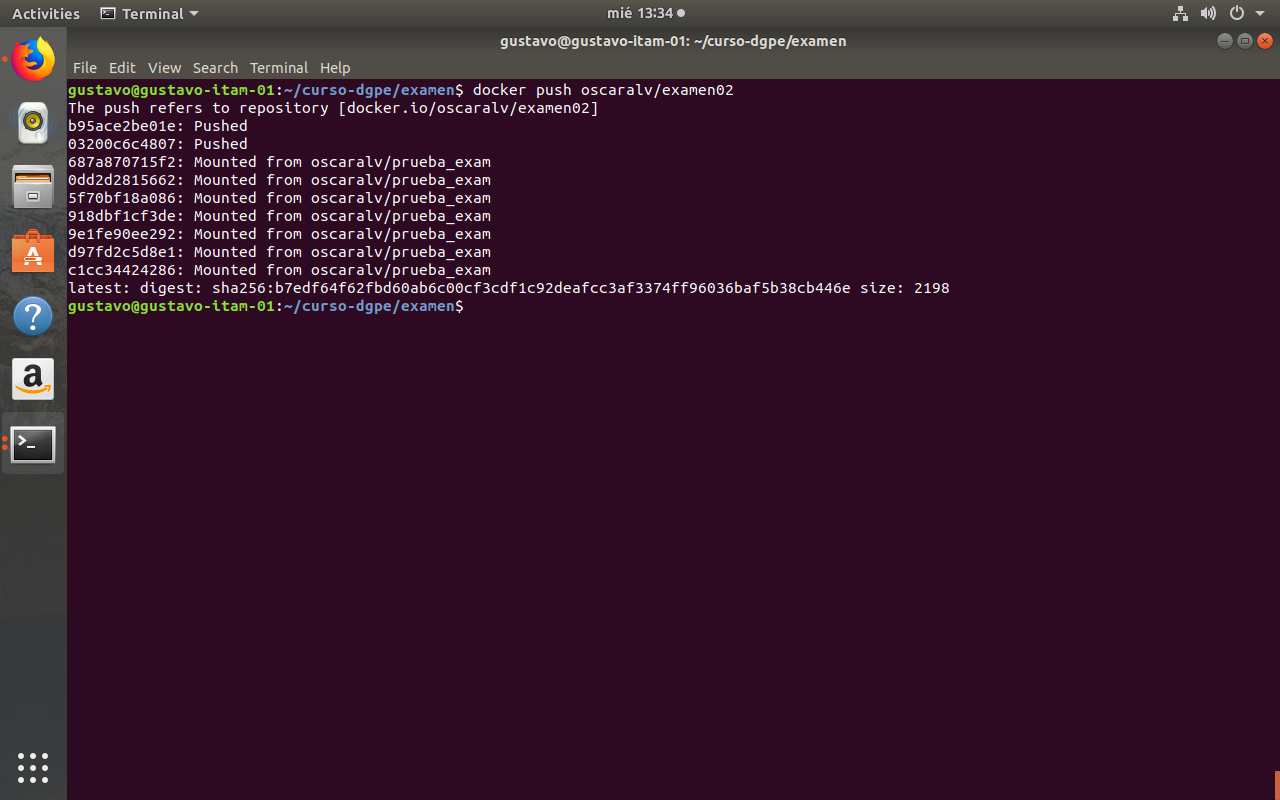
Procedemos a subir la imagen a nuestra cuenta de Docker Hub con el comando

**docker push oscaralv/examen02**

Recordemos que para poder subirla correctamente debe llevar el formato

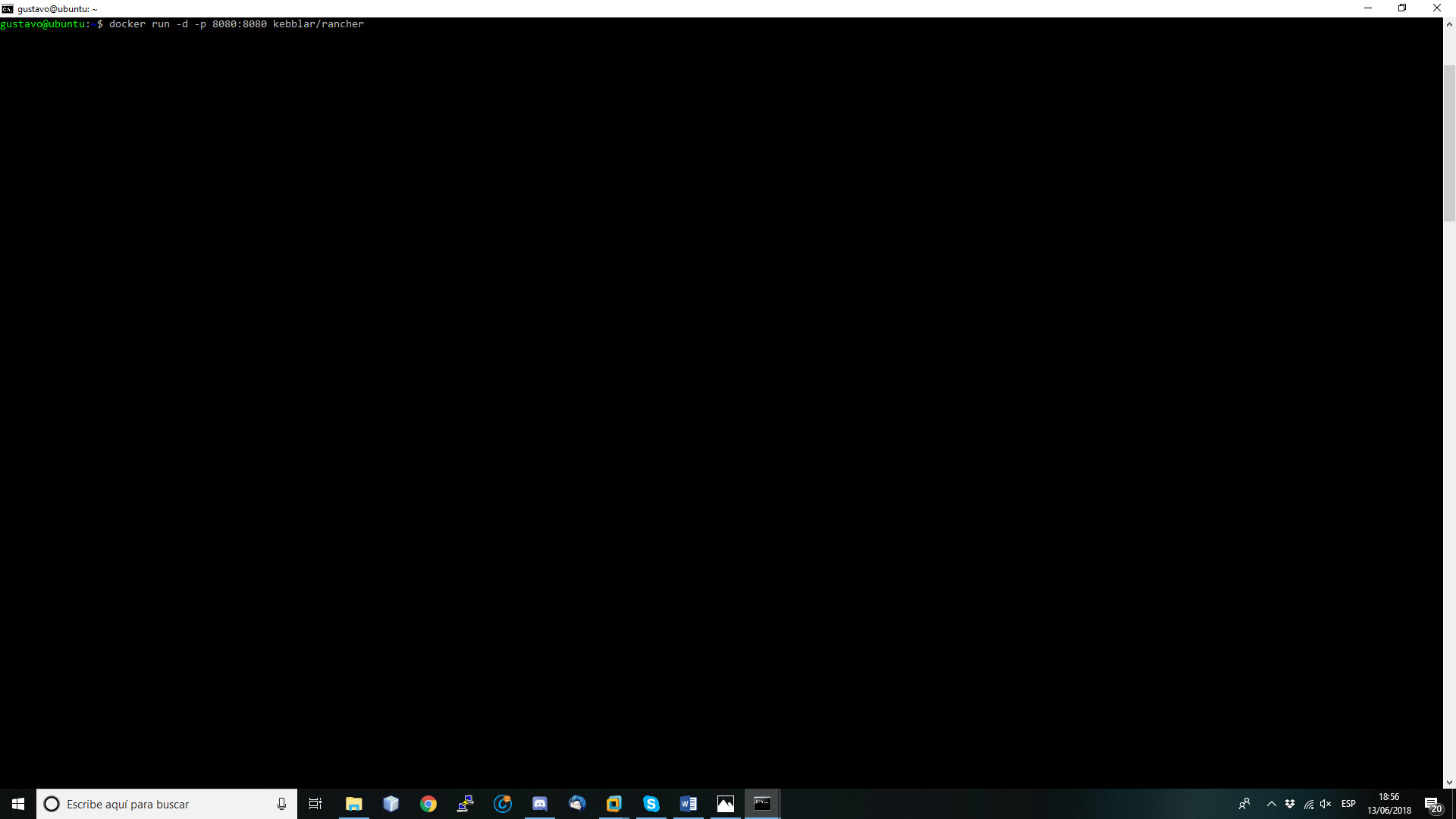
**nombre-usuario-docker/nombre-imagen**

De no ser así marcará un error.



INSTALACIÓN DE RANCHER

Ejecutamos en una maquina el siguiente comando



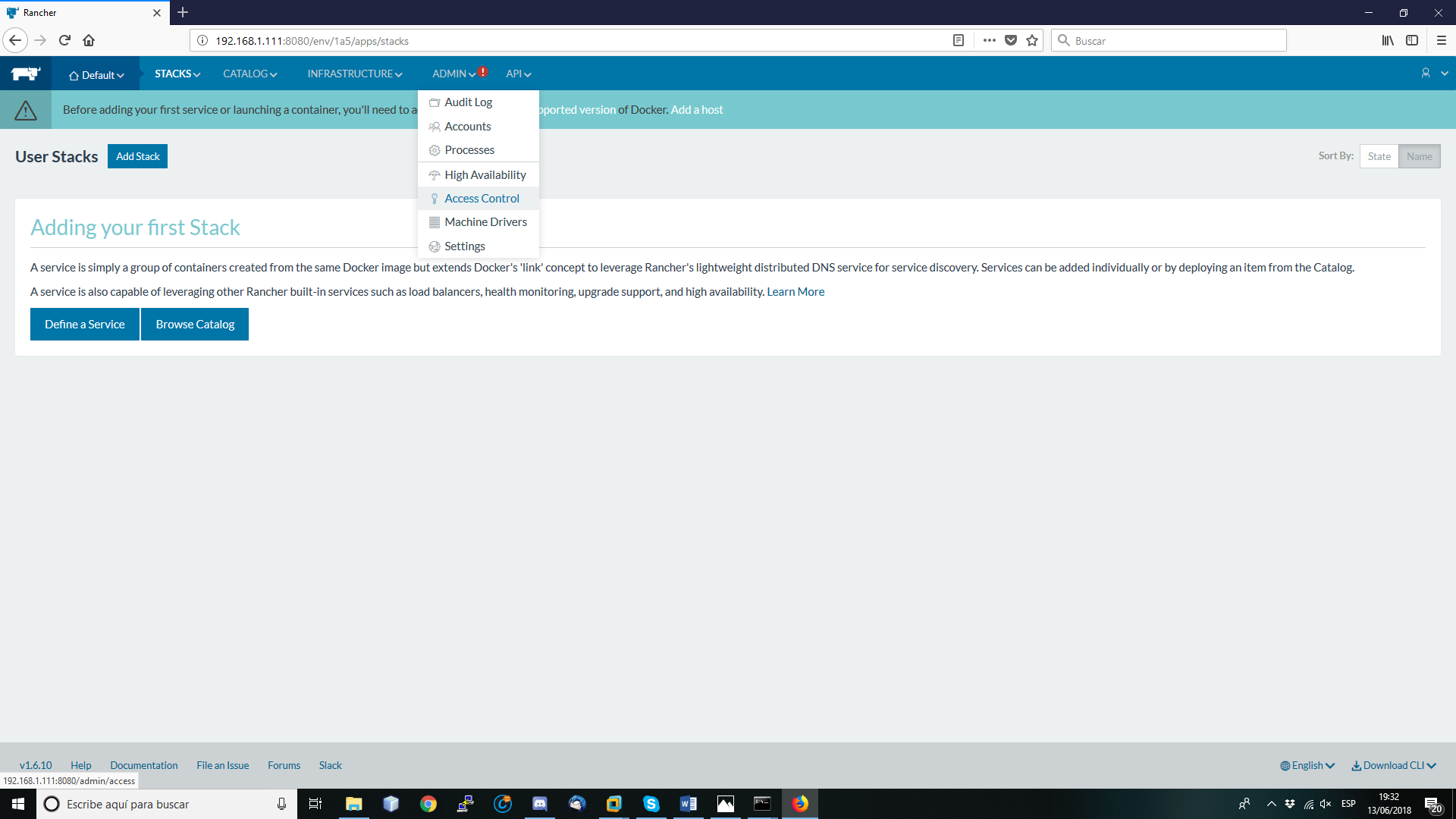
Verificamos que se haya levantado correctamente.



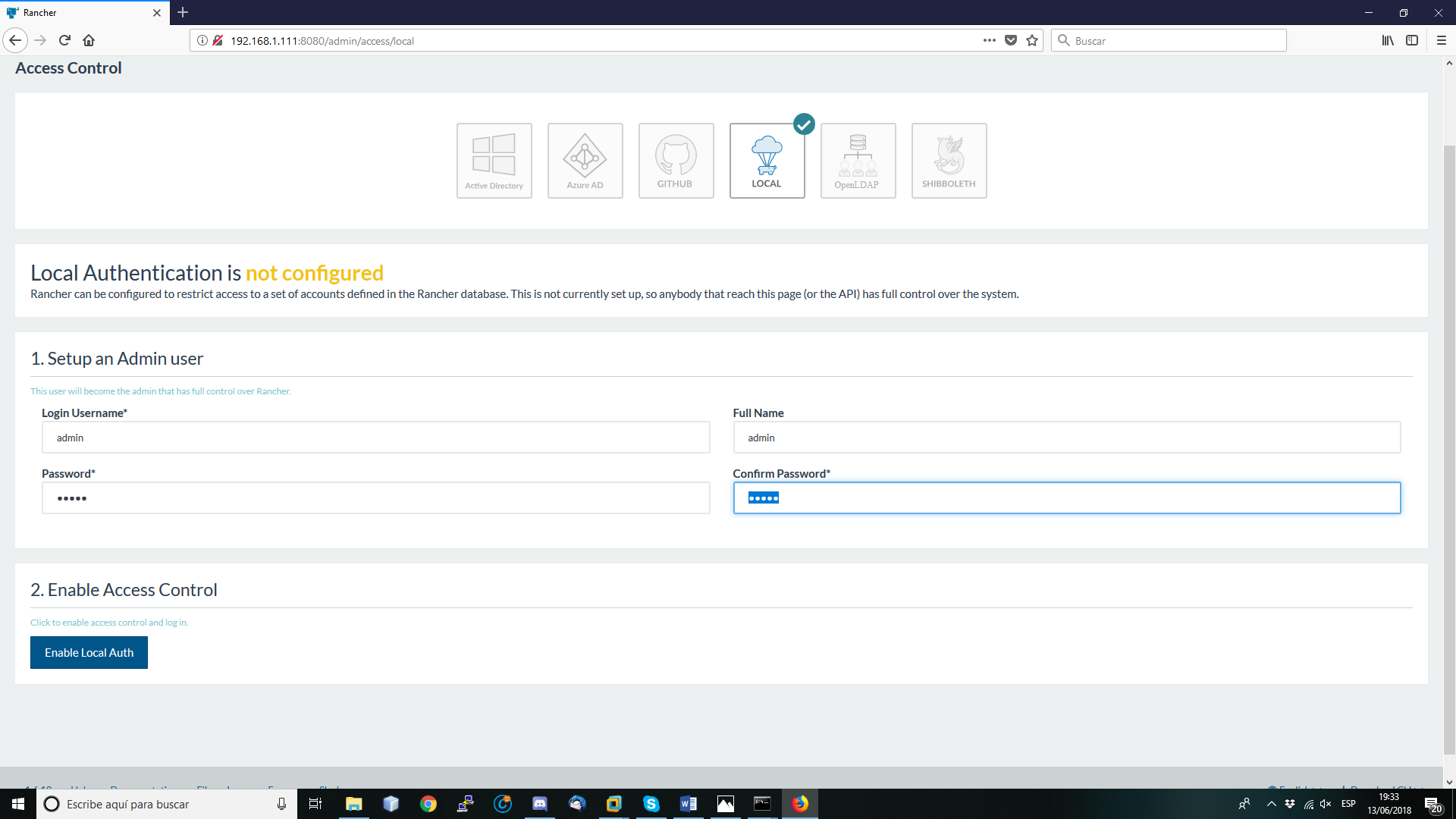
En el navegador entramos a la administración del servidor Rancher.

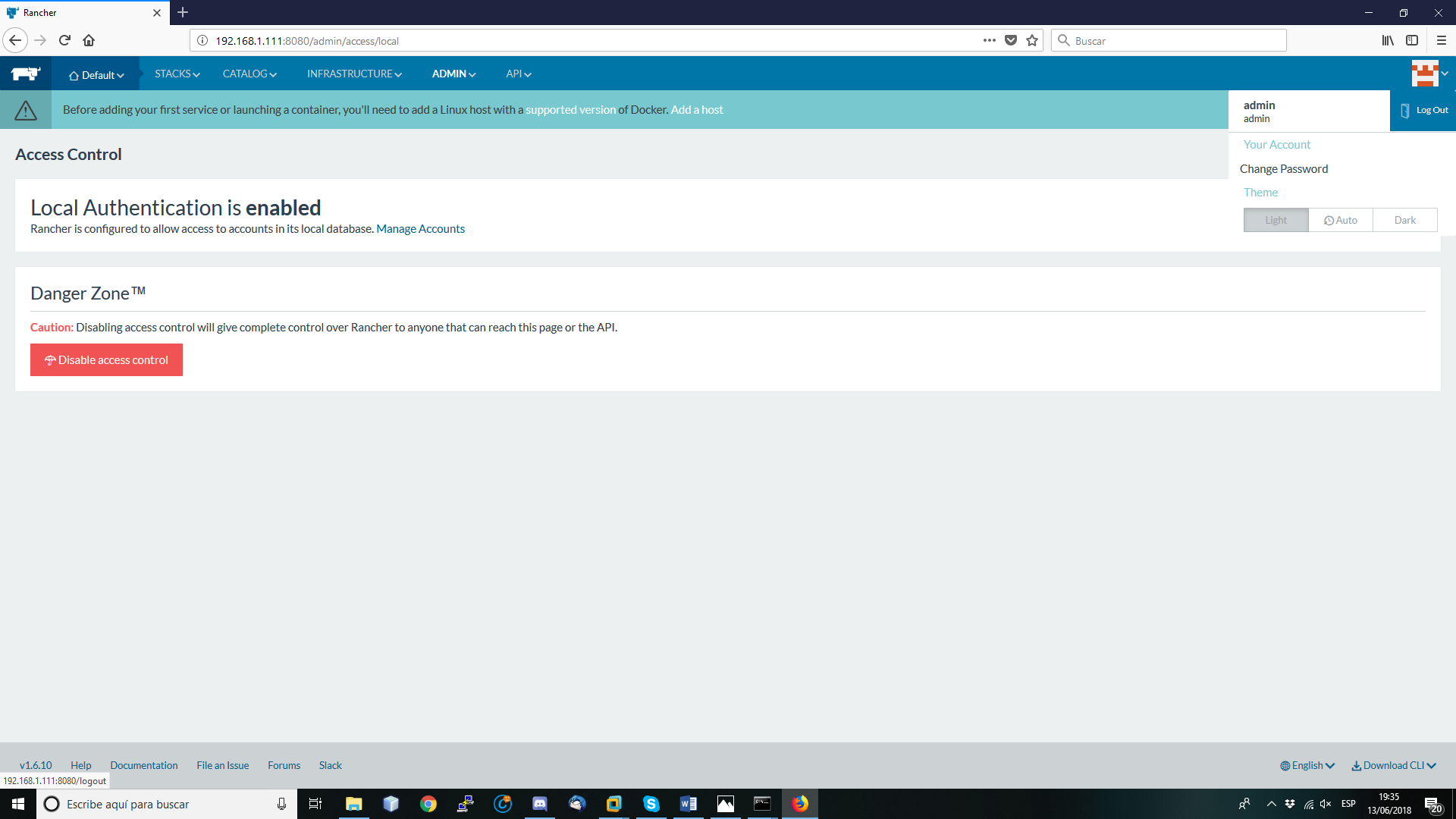


Nos dirigimos a la pestaña “**ADMIN”** y luego a “**Access Control”.**

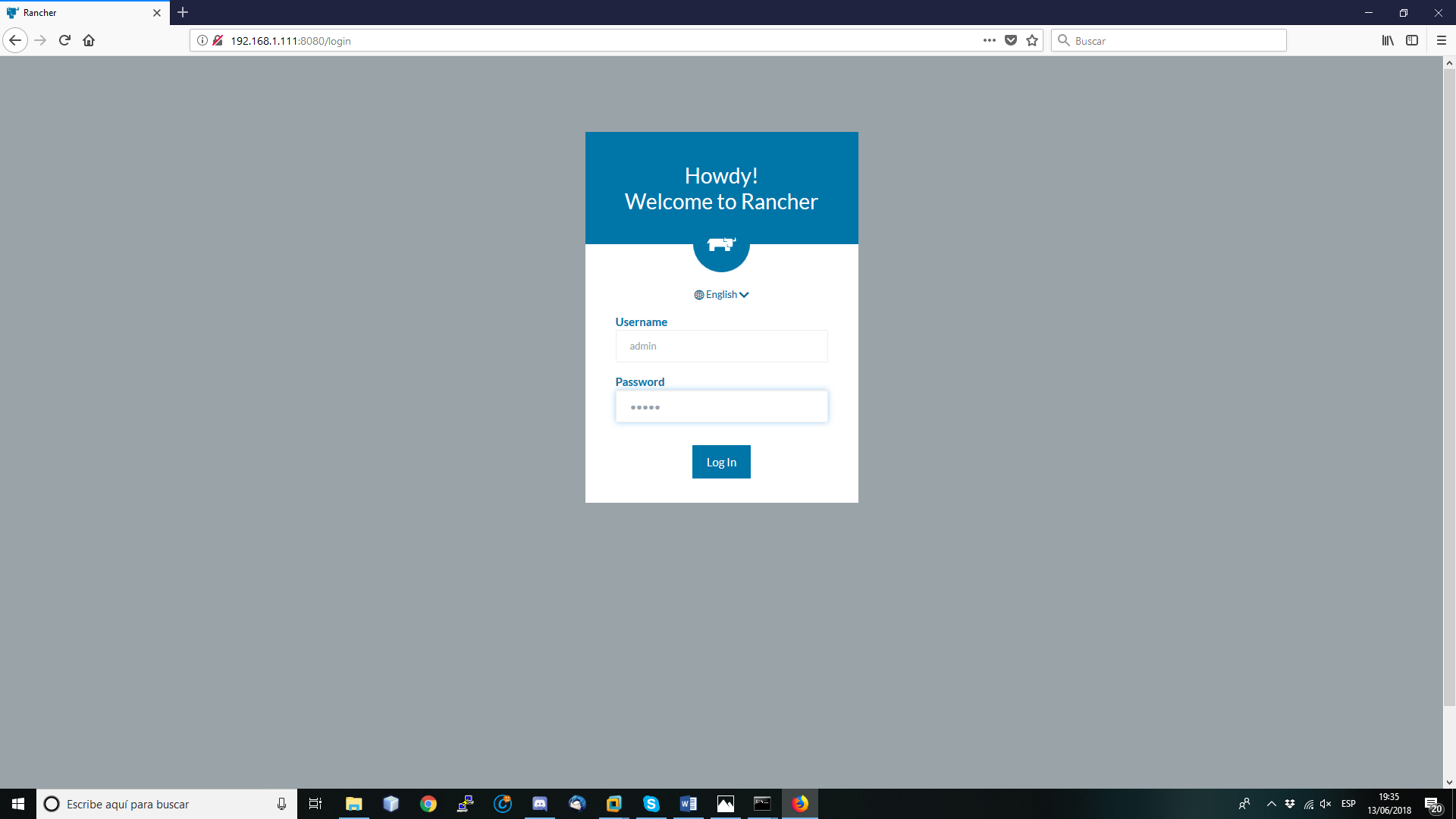


Seleccionamos “**LOCAL**” e ingresamos los datos para autenticarnos. “**Enable Local Auth**”.



Hacemos click en “**Log Out**” en la parte superior derecha. 

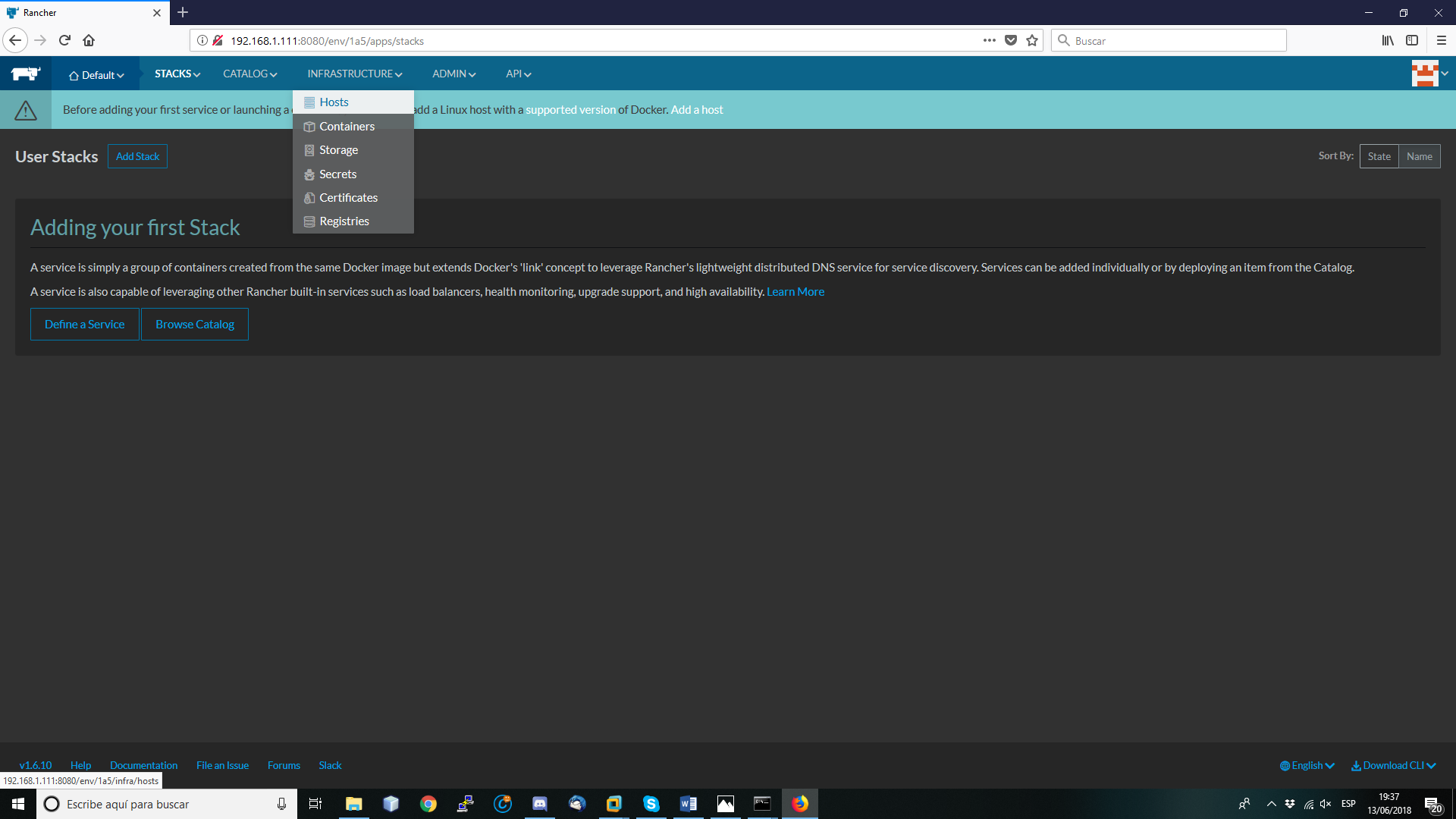
Ingresamos con los datos que registramos anteriormente.



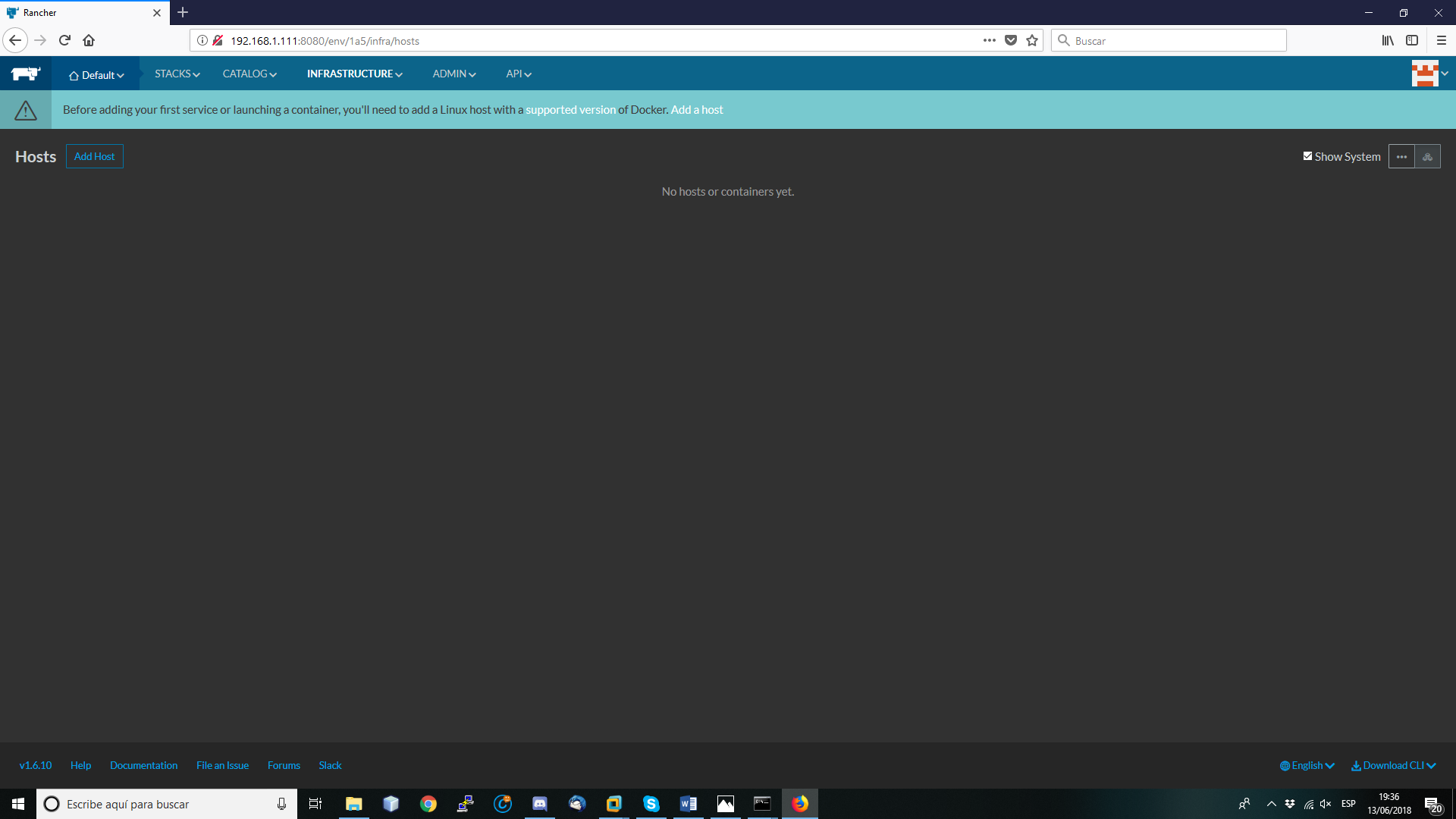
En la pestaña superior derecha elegimos el tema “**Dark”**. Paso vital para el correcto funcionamiento del servidor Rancher.



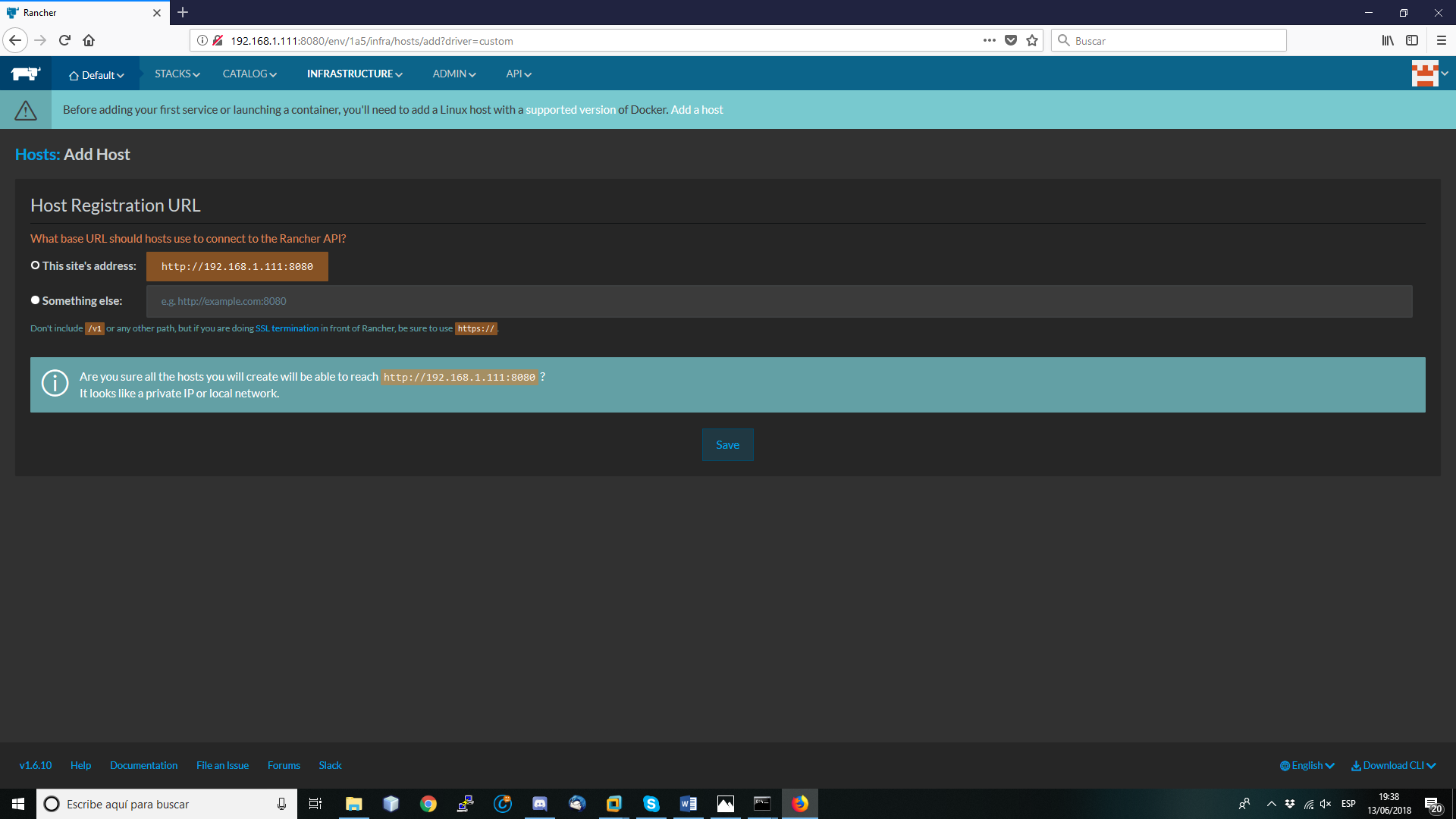
Nos dirigimos a la pestaña “**INFRASTRUCTURE**” – “**Hosts”.**



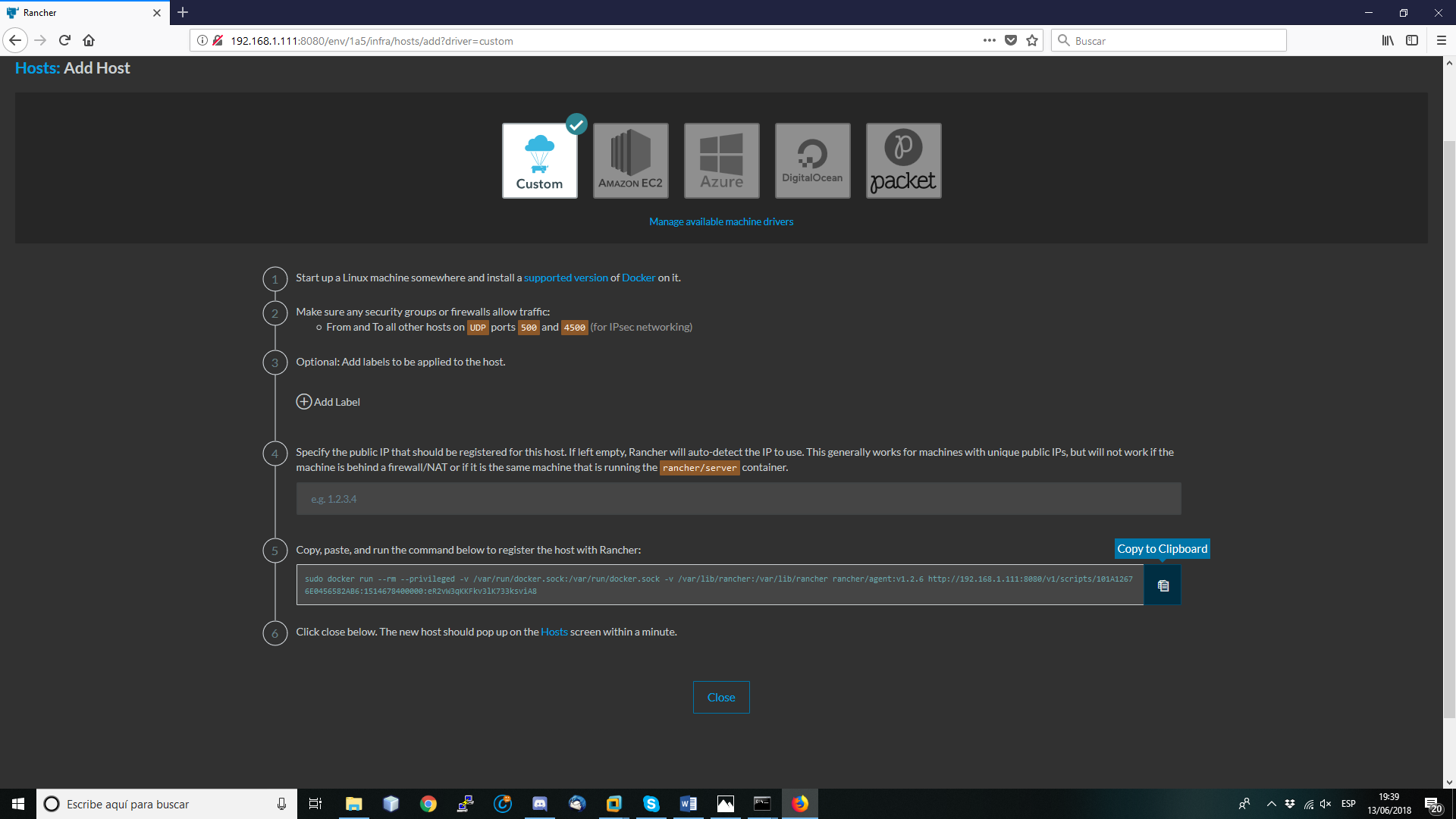
Seleccionamos “**Add Host**”.



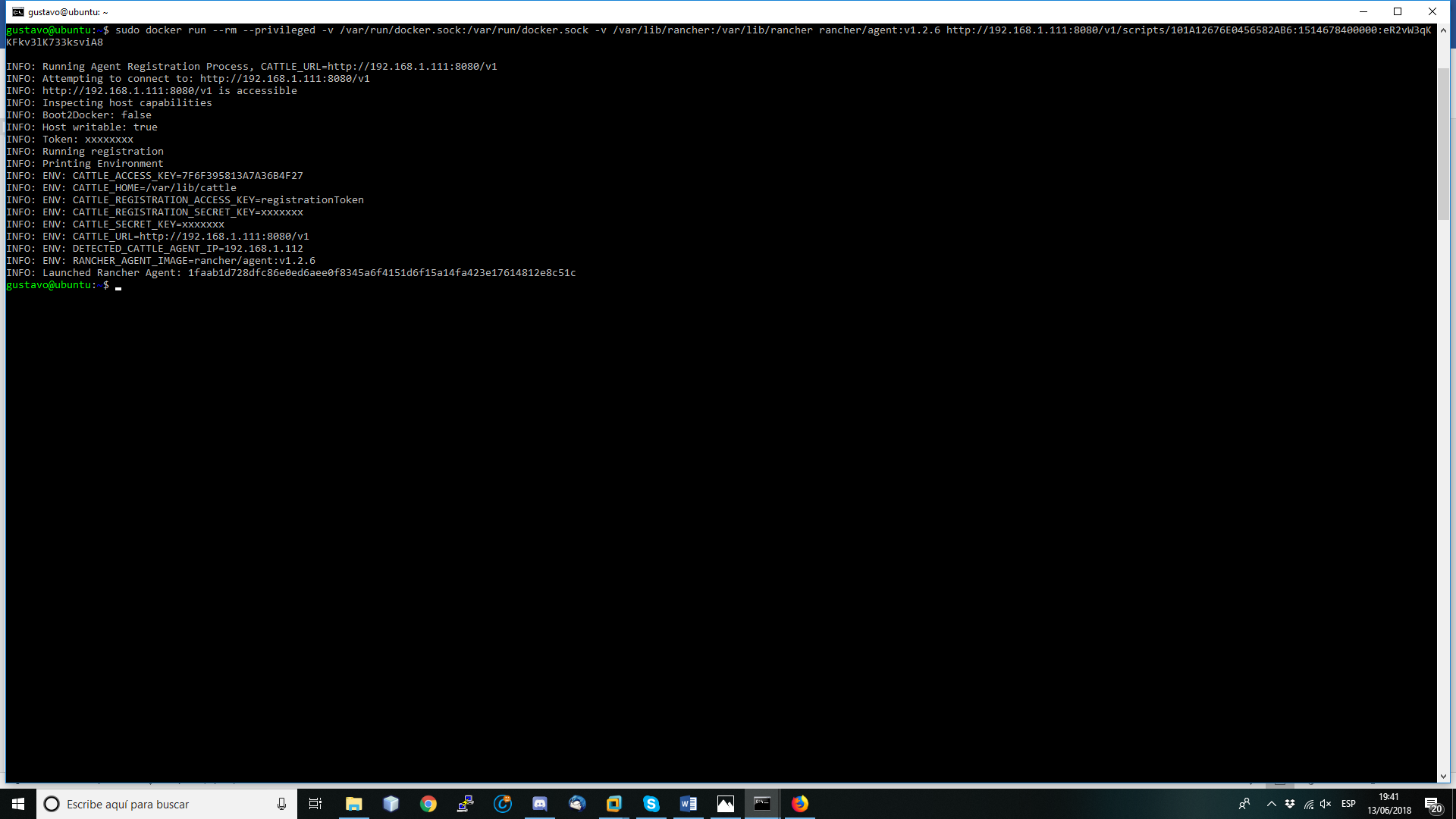
Damos click en “**Save**”.



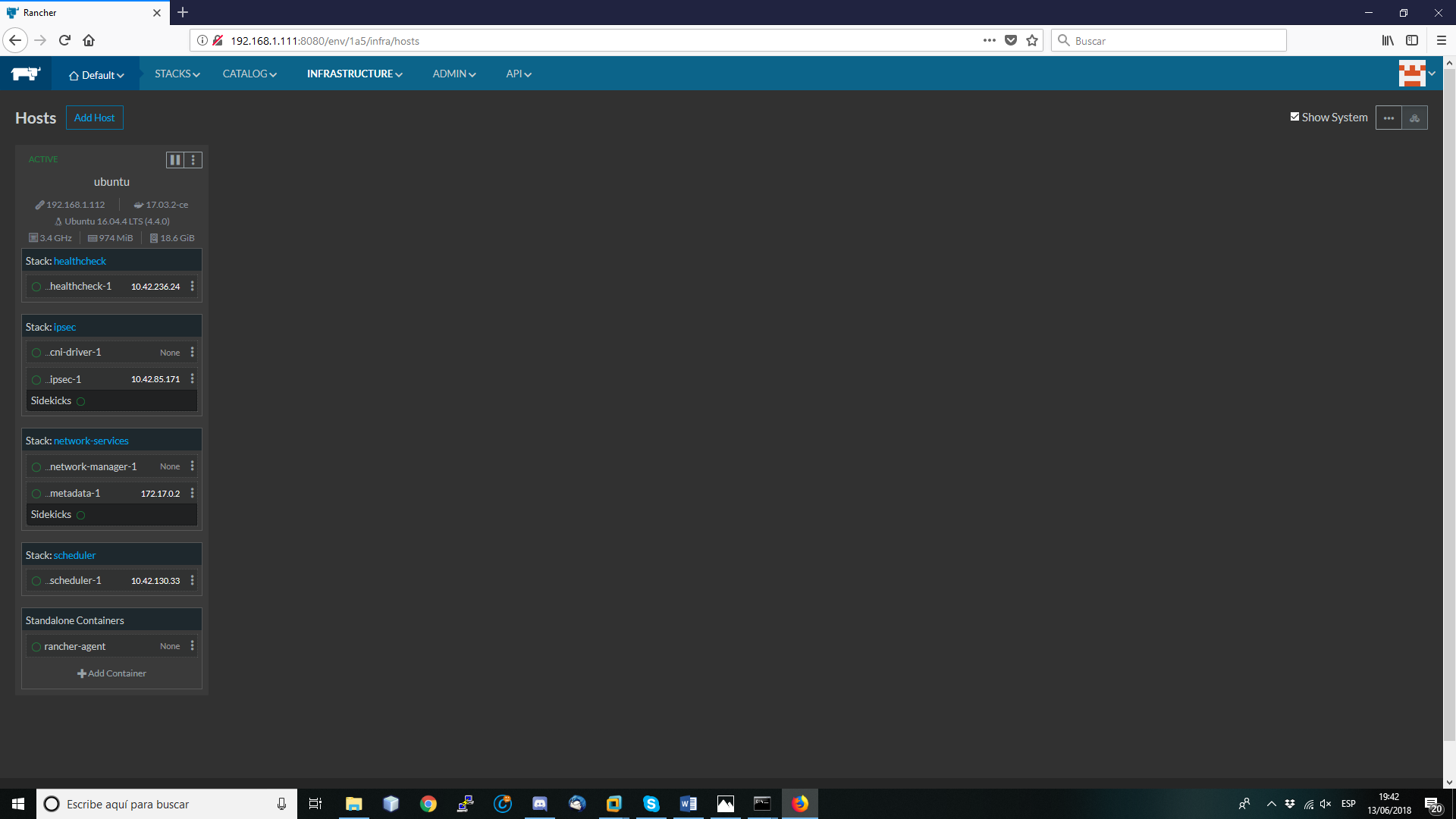
Damos click en “**Copy to Clipboard**” para copiar el comando que ejecutarán los servidores clientes para levantar el agente y conectarse al servidor Rancher.

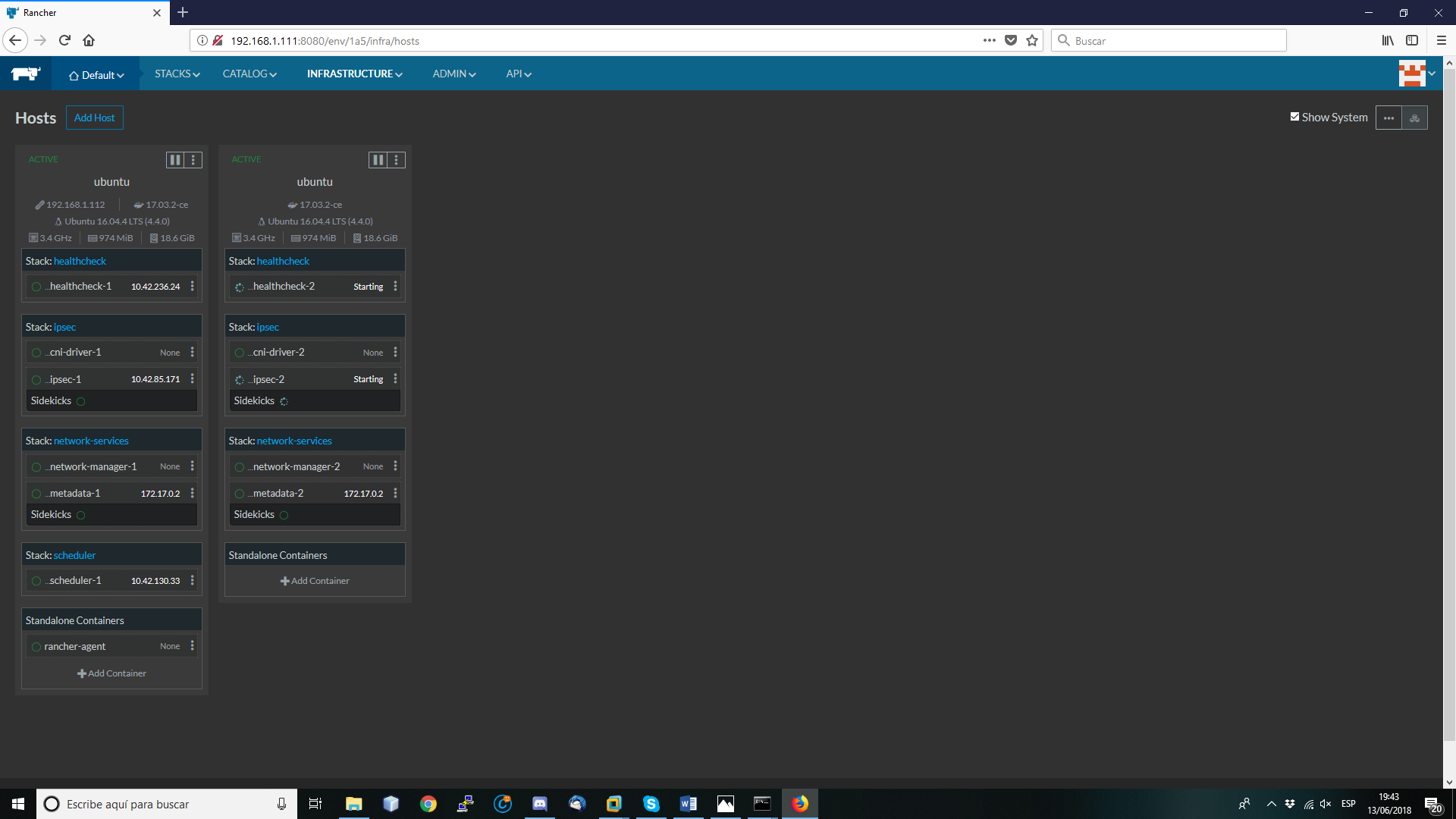


Ejecutamos el comando copiado en ambos clientes.



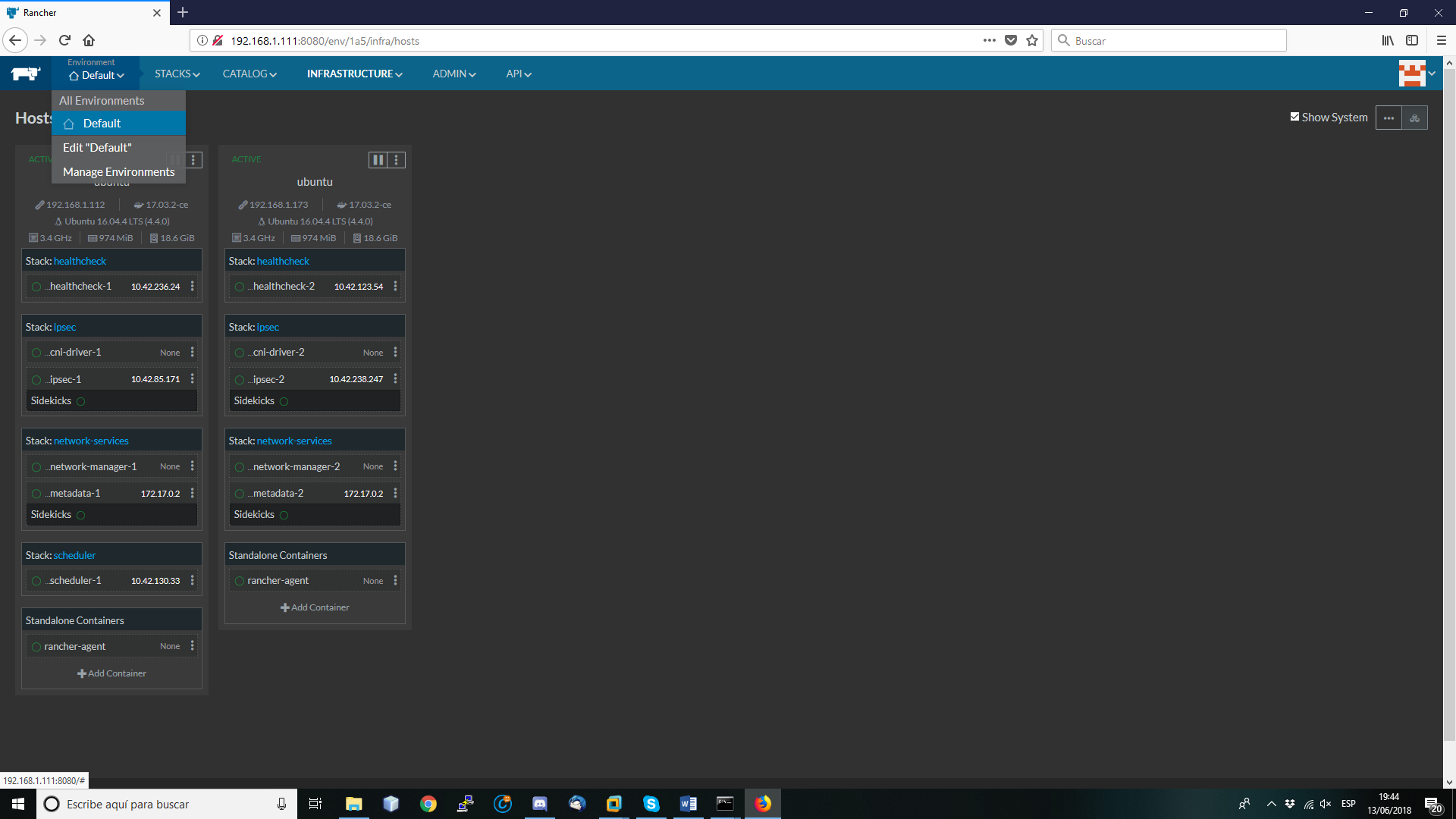
Verificamos que en el servidor aparecen los clientes que se van conectando



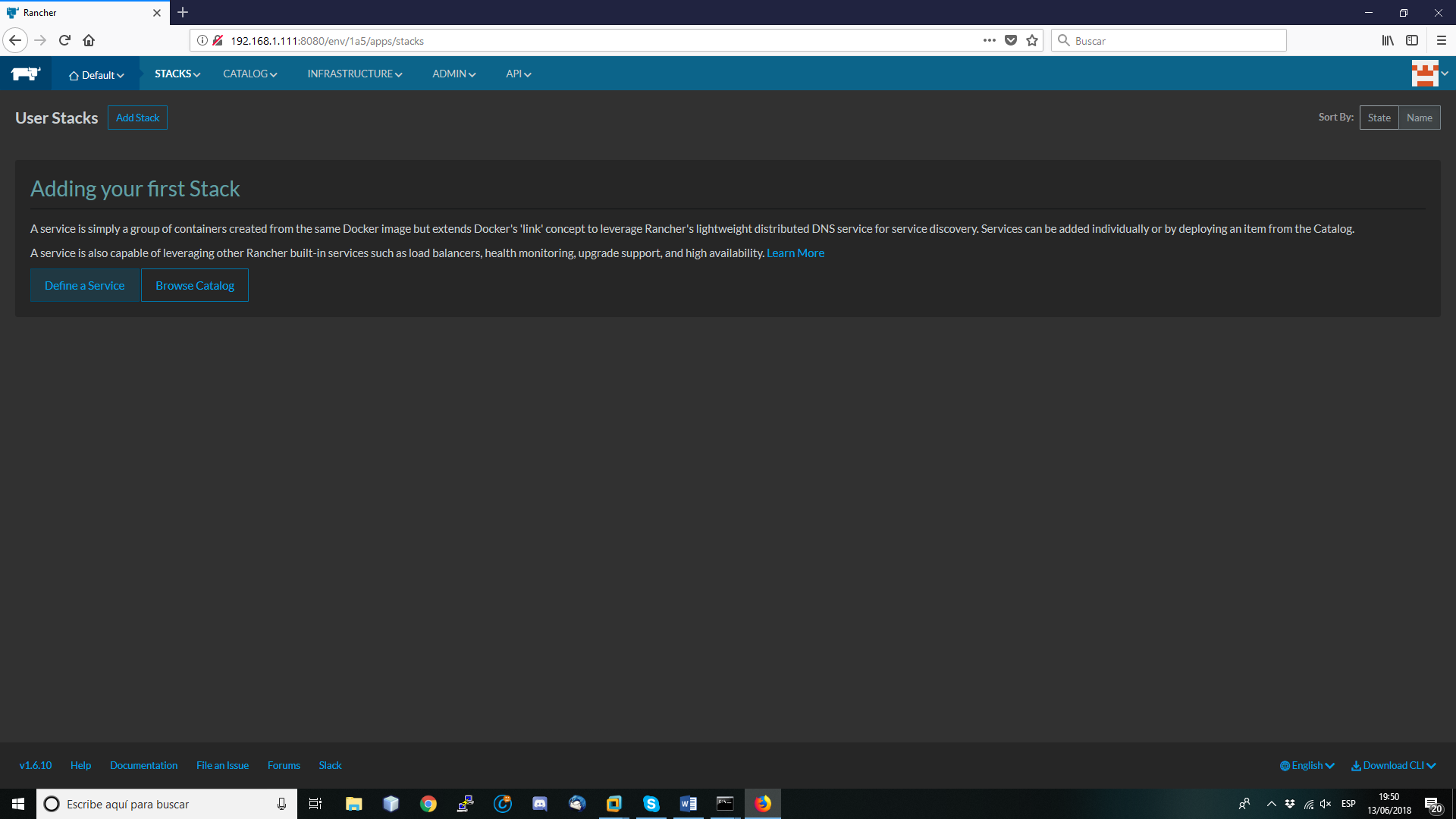


CREACIÓN DE UN SERVICIO EN RANCHER

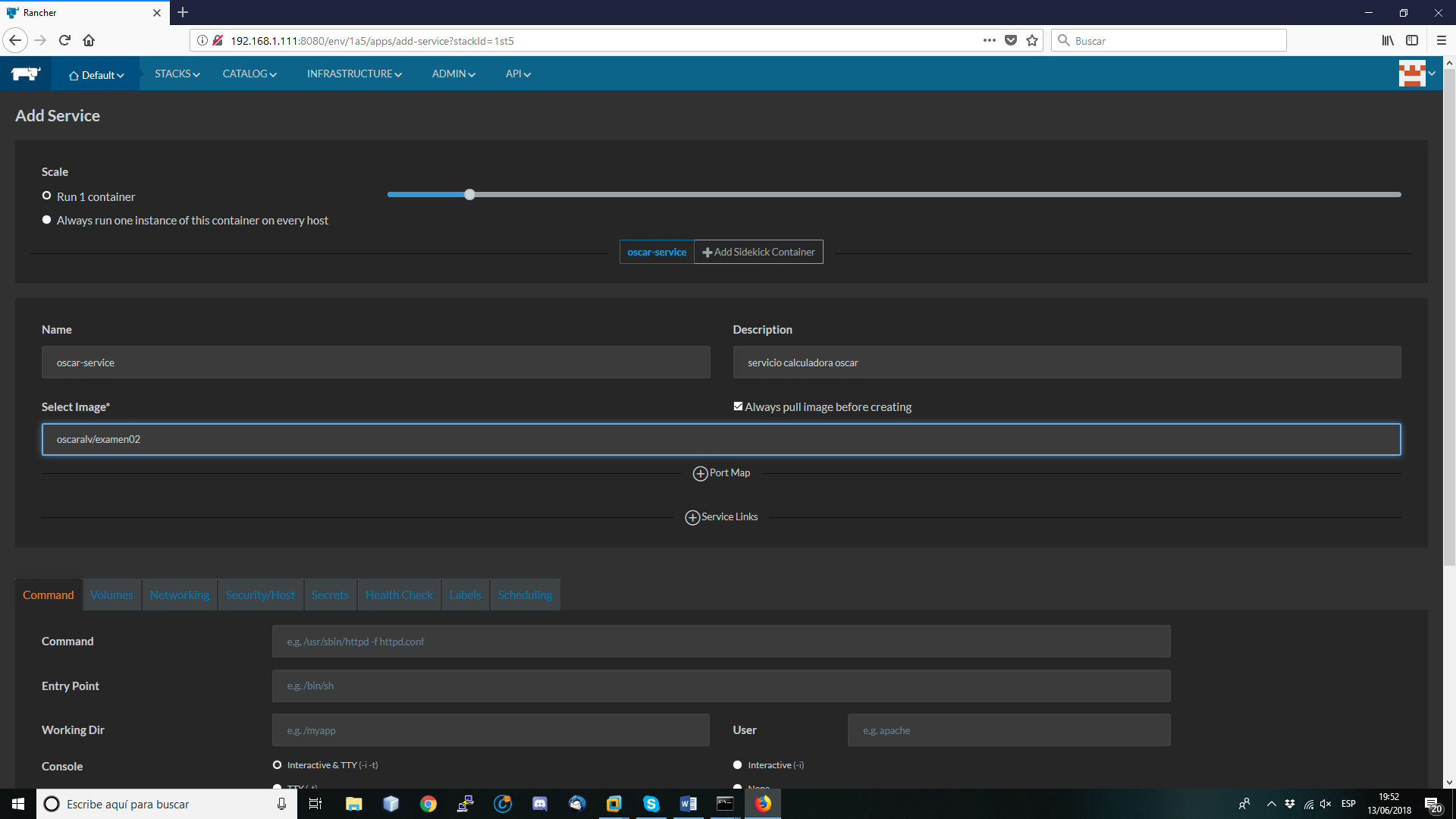
Damos click en la pestaña “**Default** – “**Default**”.



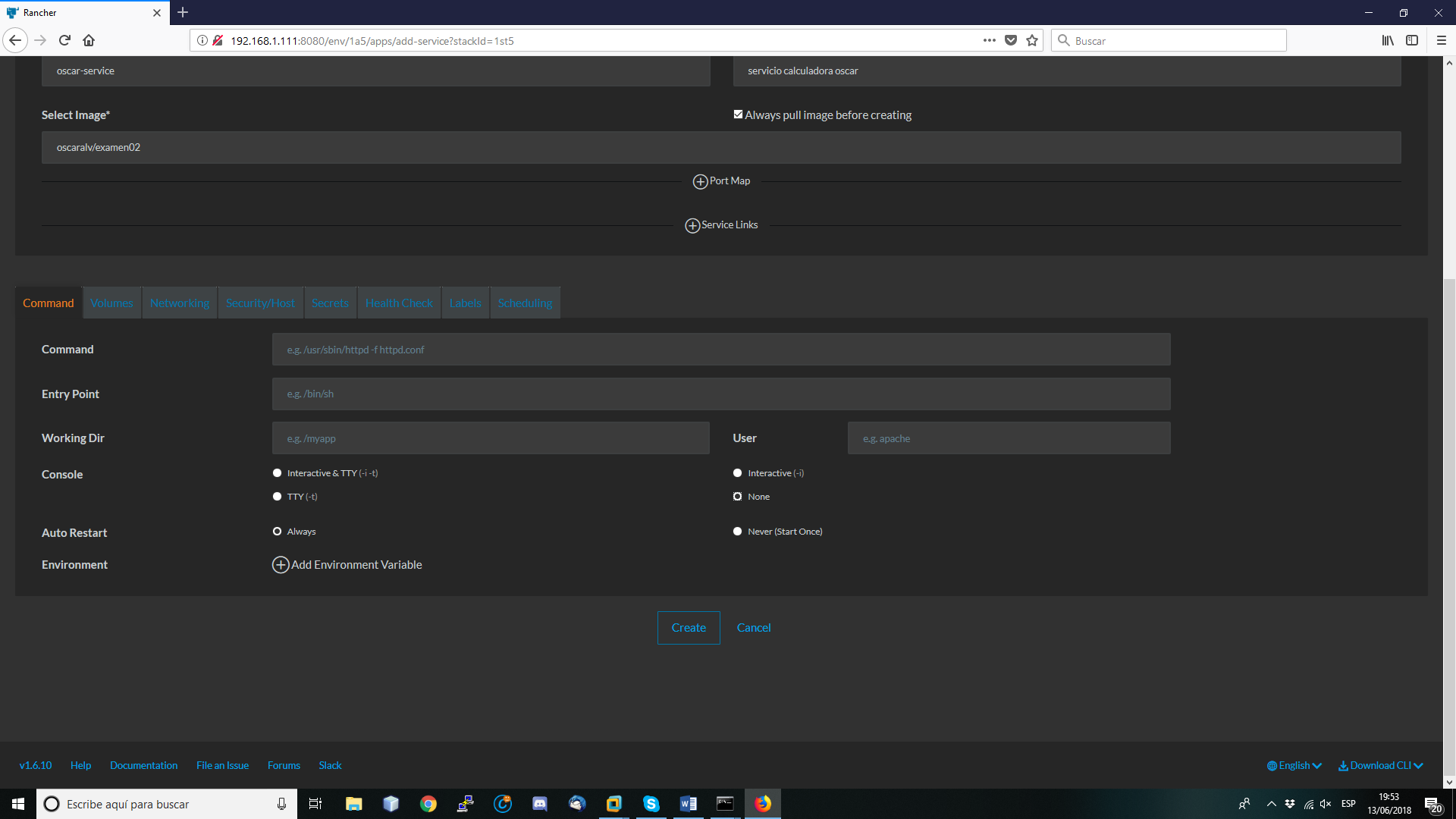
Seleccionamos “**Define a Service**”.



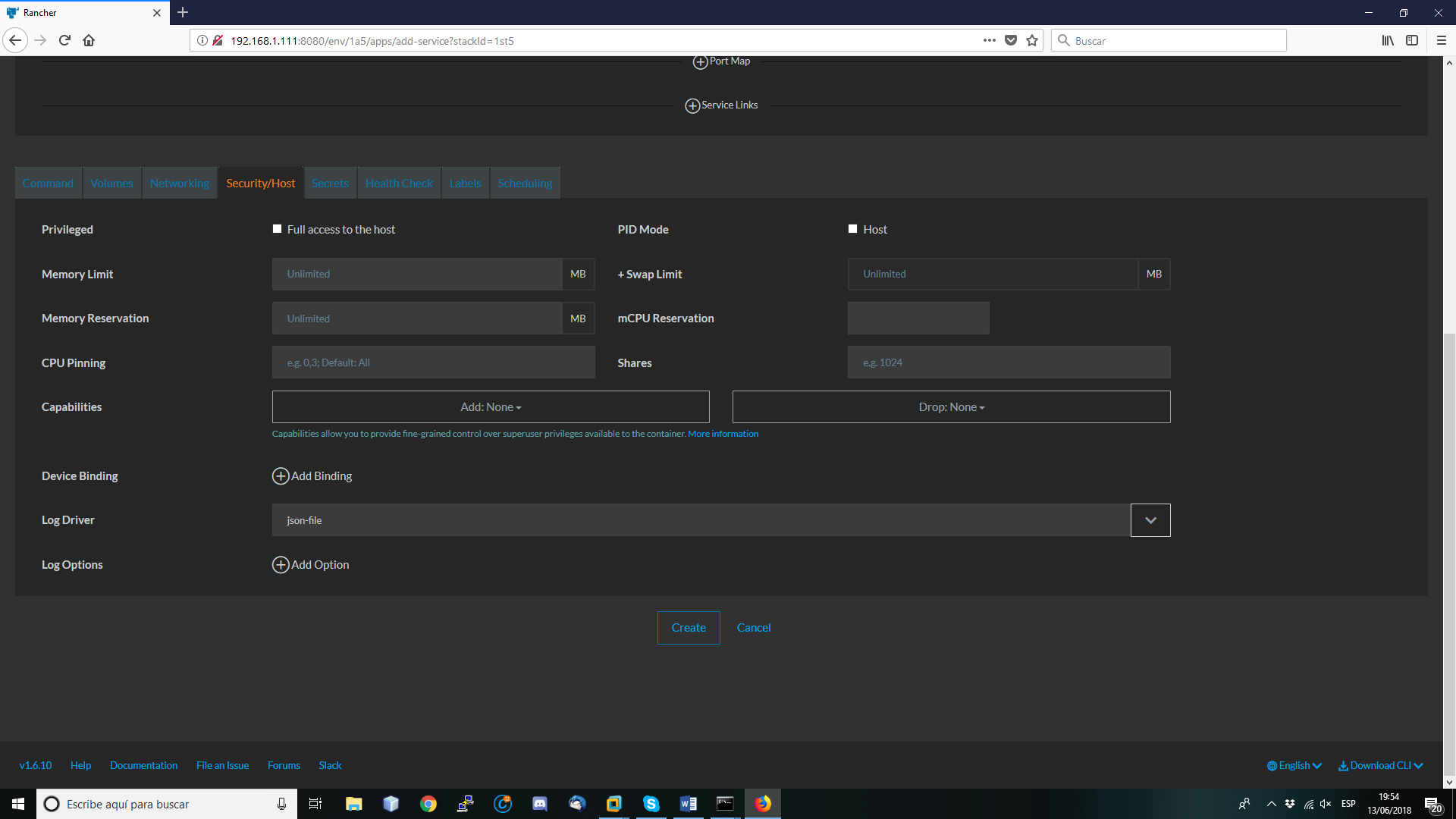
Ingresamos el nombre del servicio, una descripción y la imagen que vamos a utilizar para levantar el servicio. En este caso es la imagen creada con anterioridad, **oscaralv/examen02**.



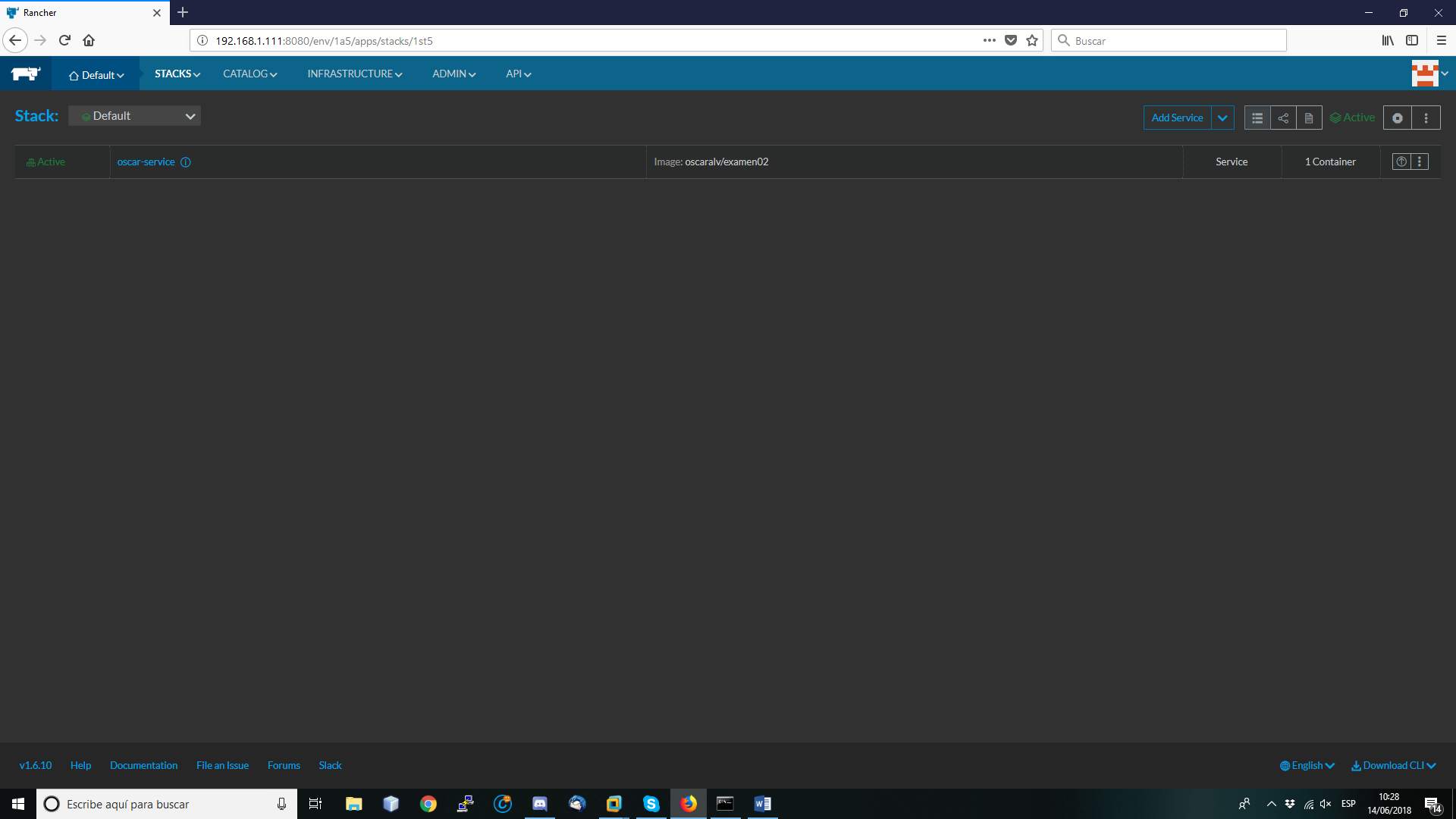
En la parte inferior seleccionamos en la sección “**Console**” la opción “**None**”.



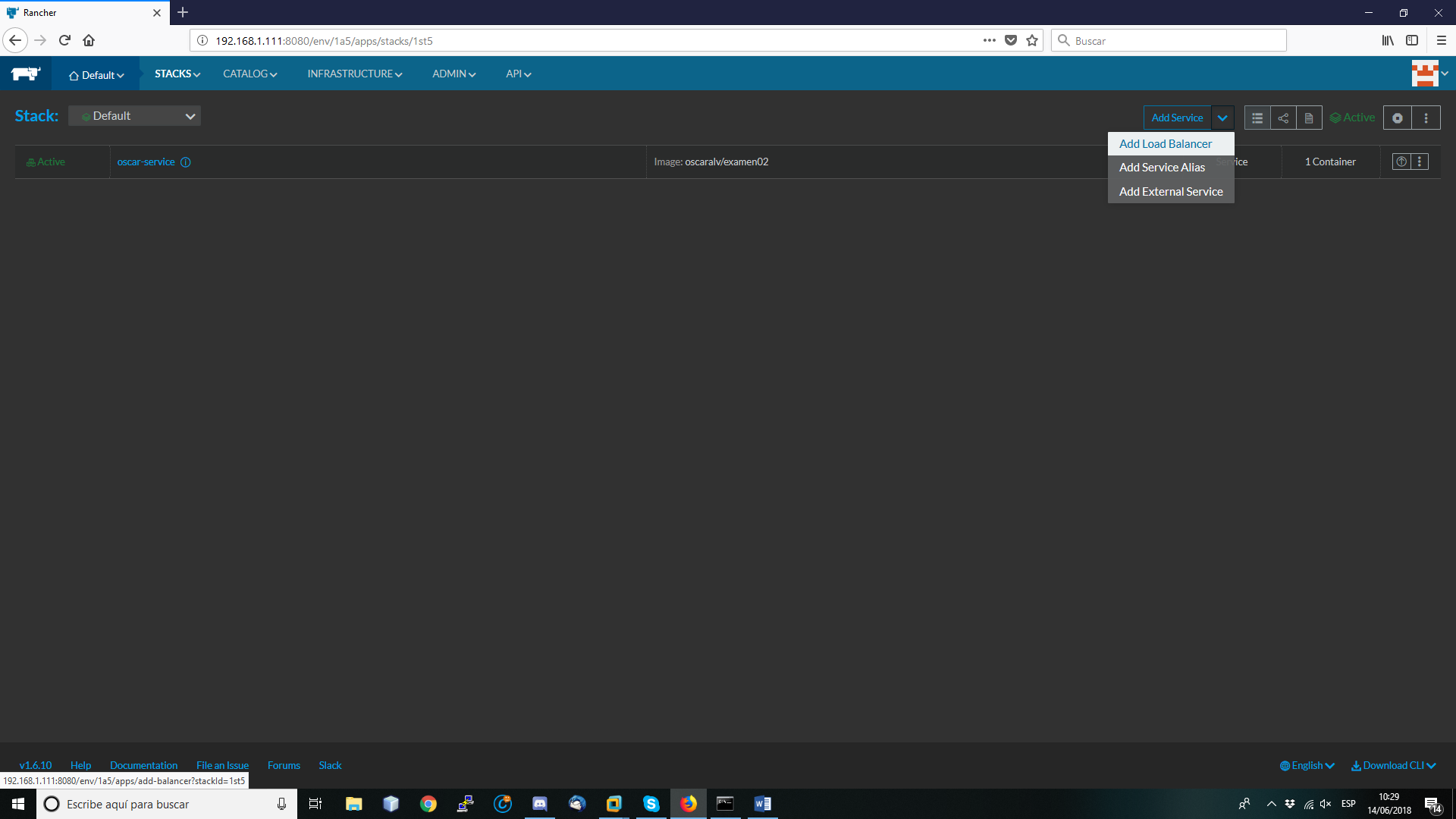
Cambiamos a la pestaña “**Security/Host**” y en la sección “Log Driver” elegimos la opción “**json-file**”. Damos click en “**Create**”.



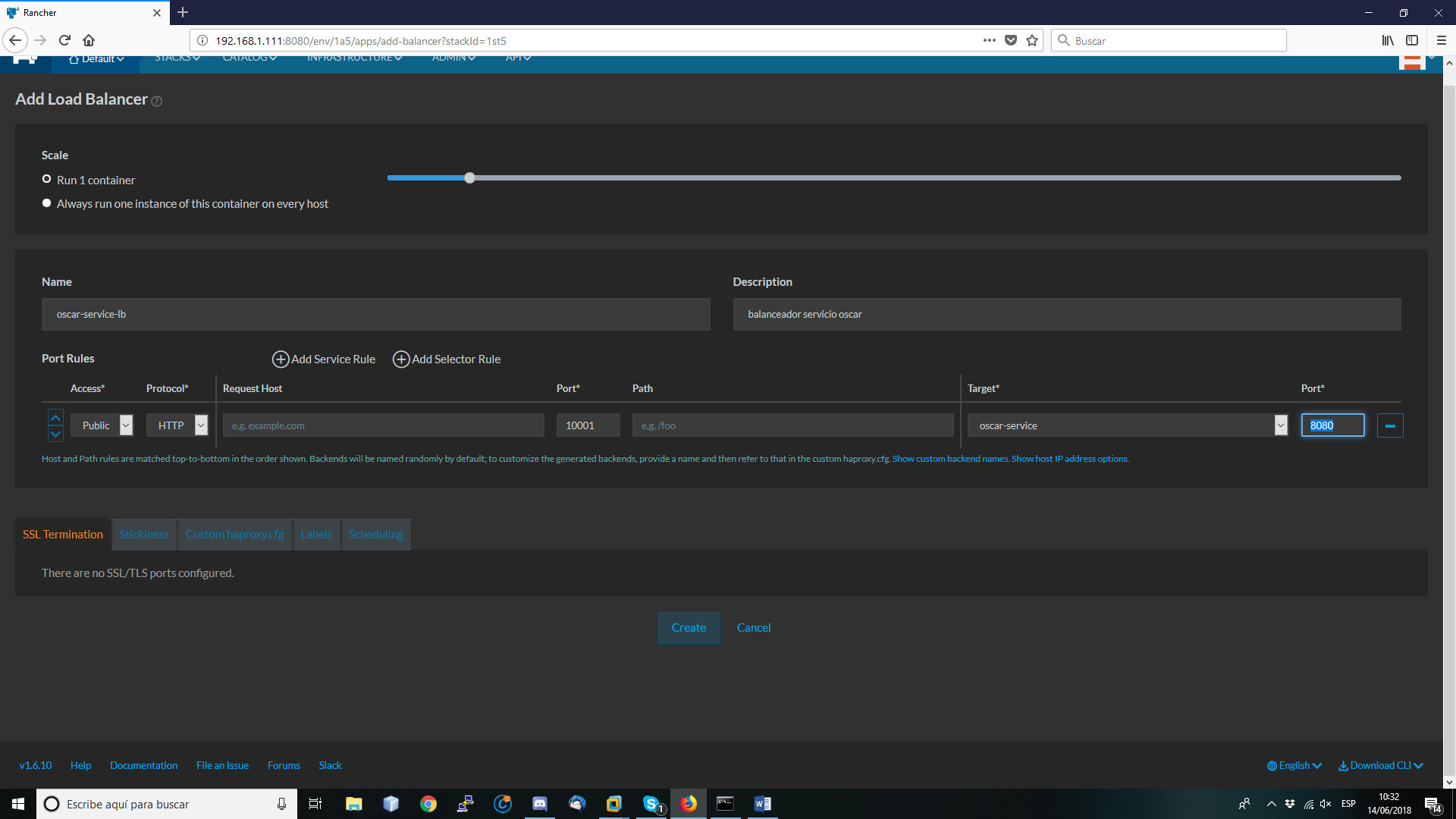
Abrimos de nuevo la pestaña “**Default**”- “**Default**” y vemos que nuestro servicio ya aparece activo.



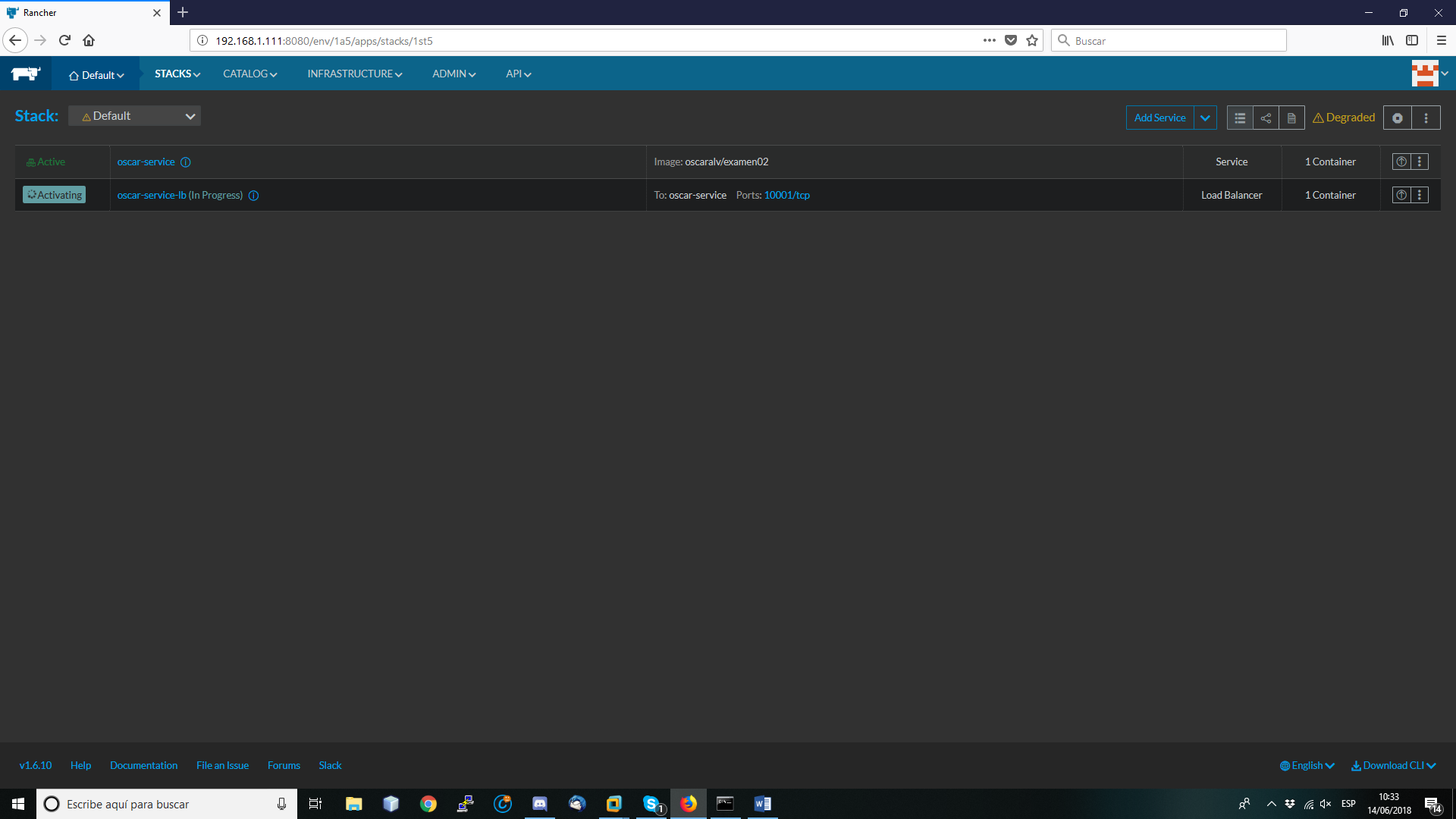
En la parte superior derecha desplegamos el menú “**Add Service**” y seleccionamos la opción “**Add Load Balancer**”.



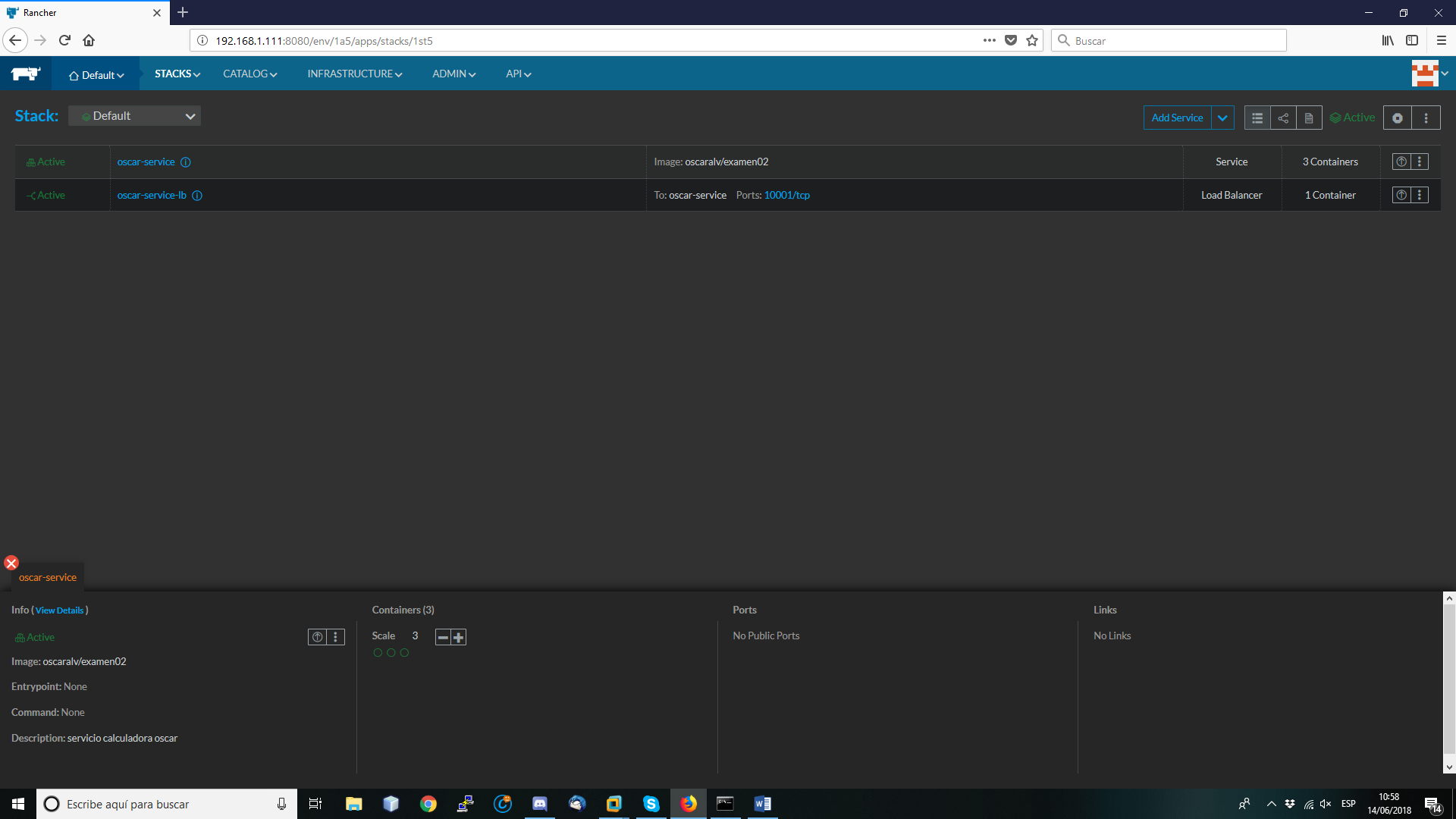
Definimos un nombre para el balanceador, una descripción, el puerto donde se va desplegar, el servicio que va balancear y el puerto que usa ese servicio internamente. Damos click en “**Create**”.



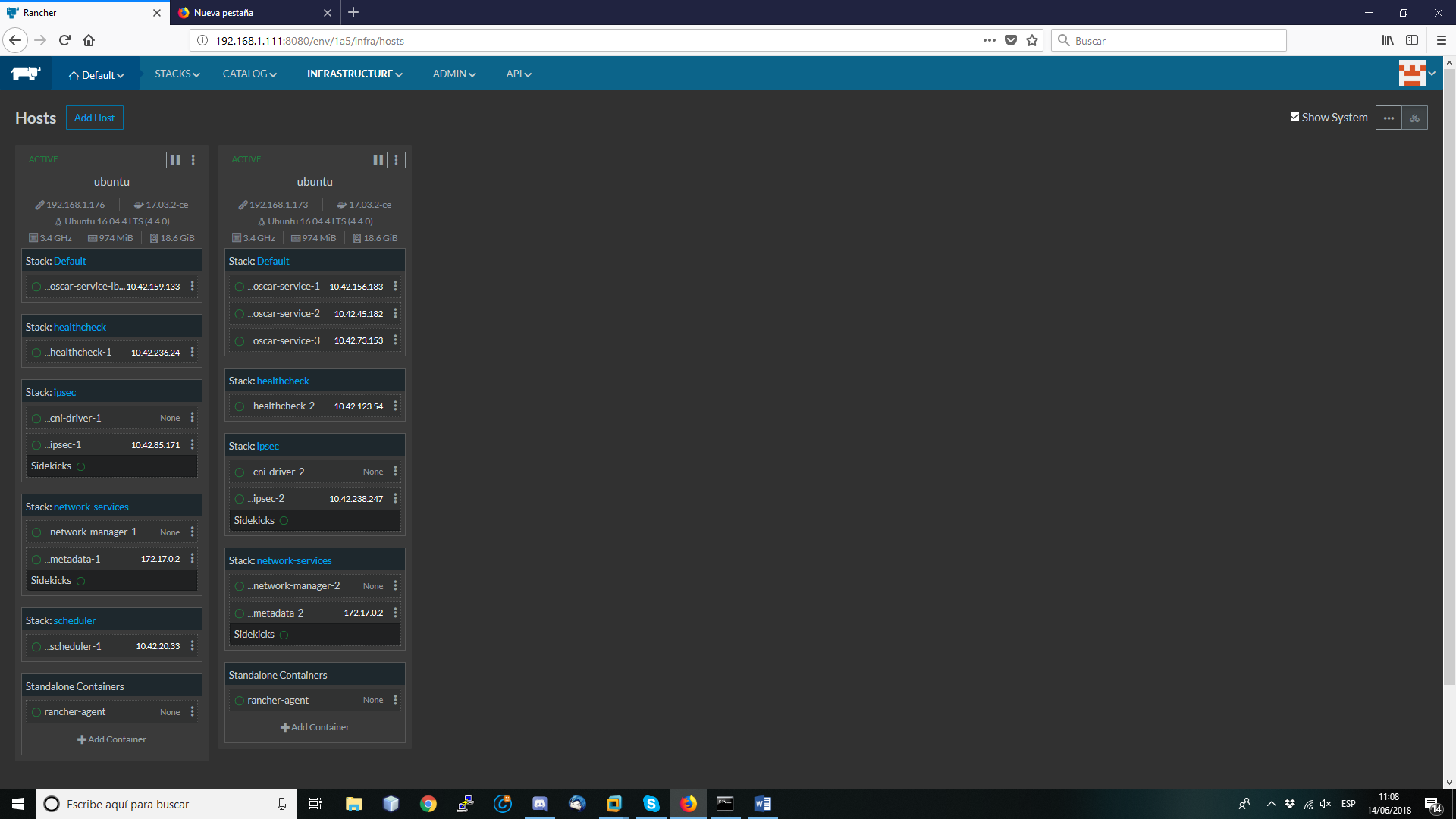
Vemos que en la pestaña default aparece el balanceador junto a nuestro servicio. Debemos esperar a que se despliegue y quede en estado activo.



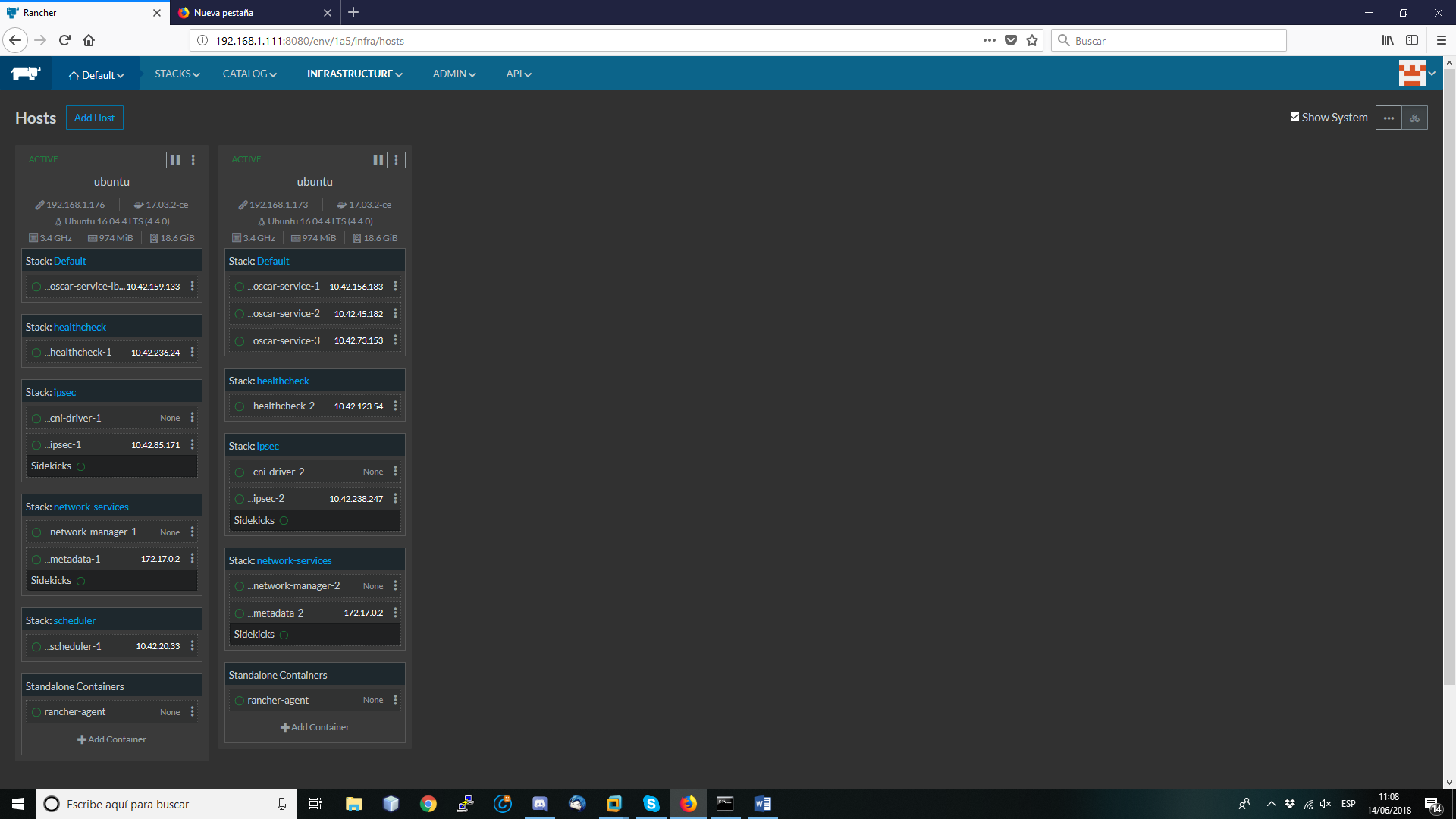
Seleccionamos la “**i**” junto al nombre del servicio y en la parte inferior en el segmento “**Containers**” aumentamos la escala a 3.



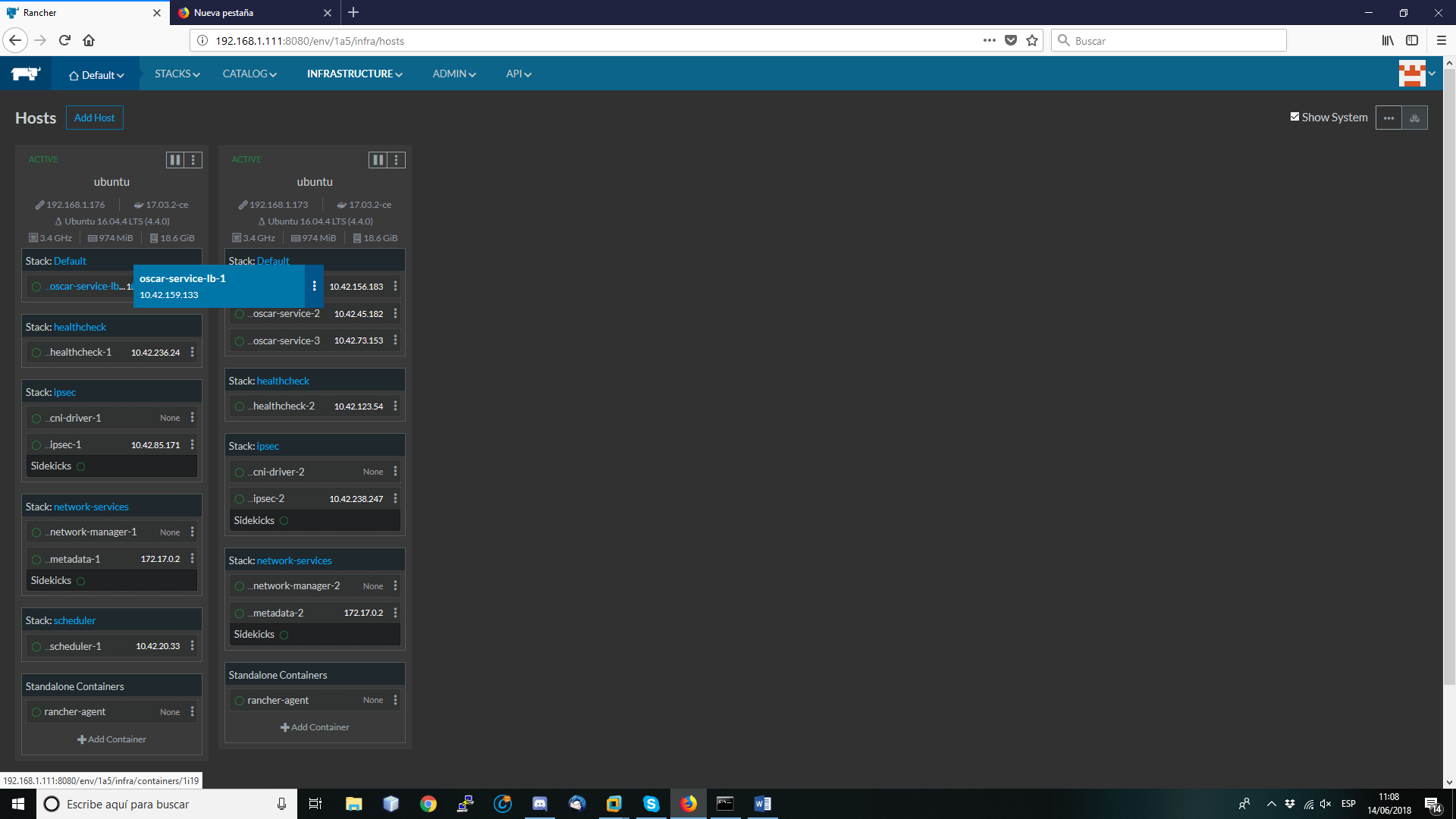
Nos dirigimos al menú “**INFRASTRUCTURE**”- “**Hosts**” y verificamos que nuestro servidor ha distribuido las instancias y el balanceador entre los 2 servidores agentes creados.



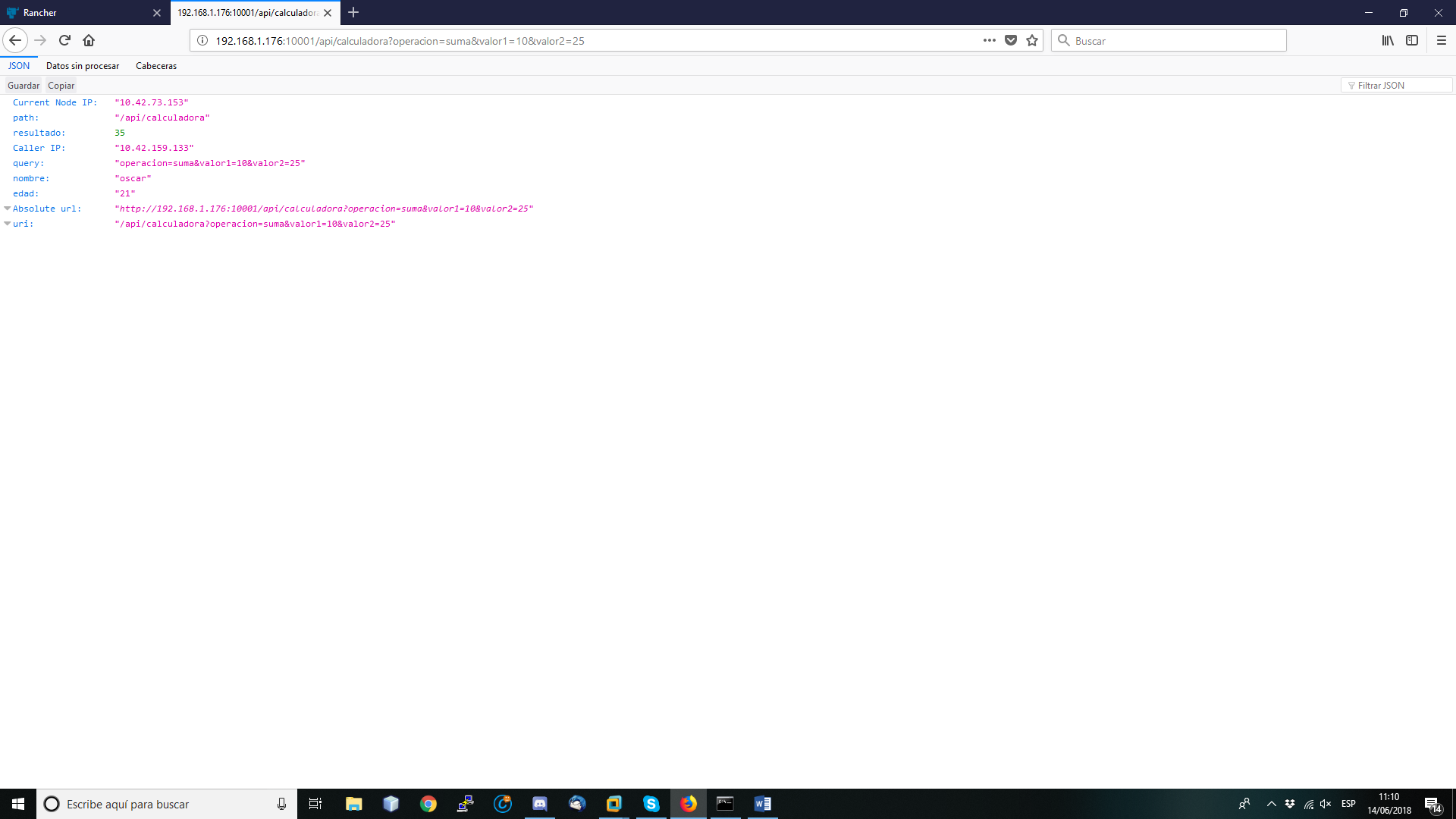
Observamos que nuestras 3 instancias del servicio tienen asignadas una **IP** distinta cada una.



Observamos que nuestro balanceador esta corriendo en la maquina **192.168.1.176**



En una ventana del navegador ejecutamos nuestra vertical llamando al balanceador con la URL <http://192.168.1.176:10001/api/calculadora?operacion=suma&valor1=10&valor2=25>



Al refrescar varias veces podemos observar que la **IP** donde se encuentra el servicio va cambiando. Esto nos indica que el balanceador está funcionando correctamente.

