### Posibles fases de la práctica

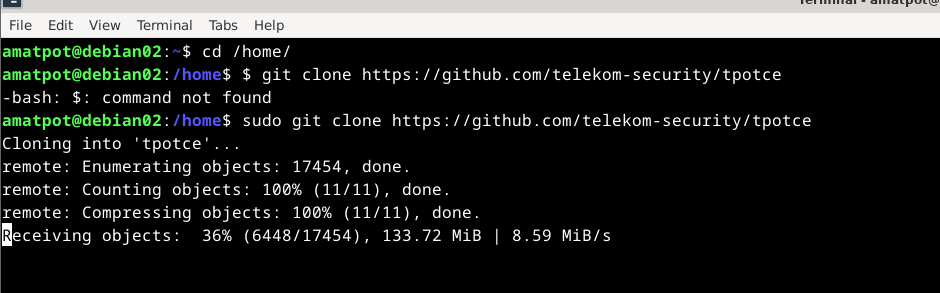
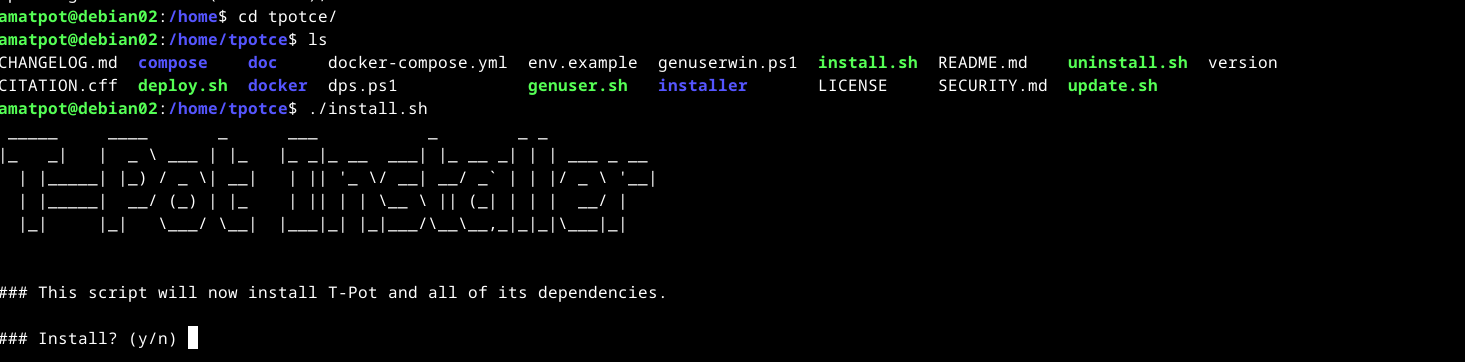
1. Montar Tpot de manera local para identificar configuraciones
2. Definir Honeypot a desplegar
3. Configurar de manera integral la implementación
4. Alta de servidores – configuraciones
5. Replica de configuraciones
6. Despliegue (Definir duración del Tpot online)
7. Primer avance: definir primer corte para revisar datos, primer progreso. Revisamos el tiempo restante.
8. Confeccionamos informe con el material previo
9. Cierre de T-pot en servidor
10. Recolectamos información, tableros, dashboard
11. Preparación informe ejecutivo, técnico
12. Ensayamos exposición

### Instalación Tpot – Local

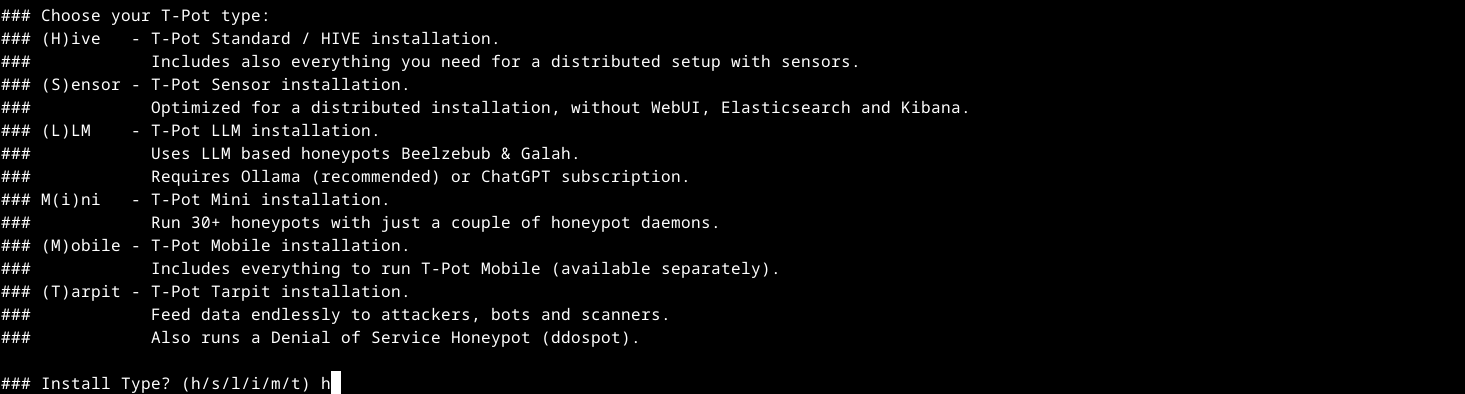
Instalamos de manera local la herramienta sobre la distribución Debian12. Nos vamos a respaldar del repositorio oficial <https://github.com/telekom-security/tpotce> (actualizado el 11.12.2024).

Creamos el usuario ordinario amatpot con permisos de sudo en Debian. Desde la ubicación $HOME clonaremos el repositorio, nos ubicamos en el directorio y lanzamos el instalador

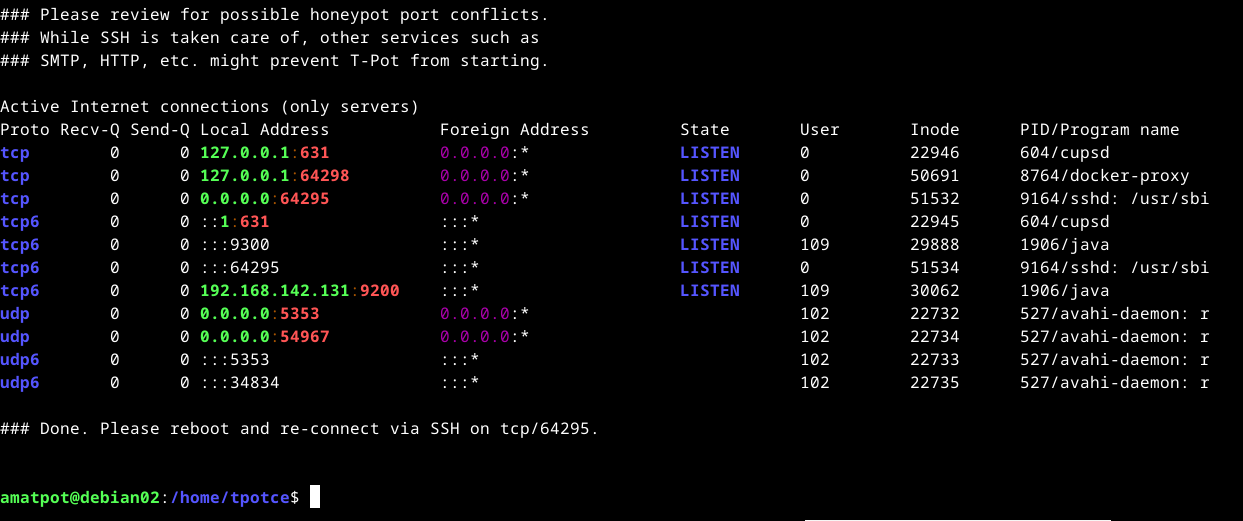
$ git clone <https://github.com/telekom-security/tpotce>  
$ cd tpotce  
$ ./install.sh

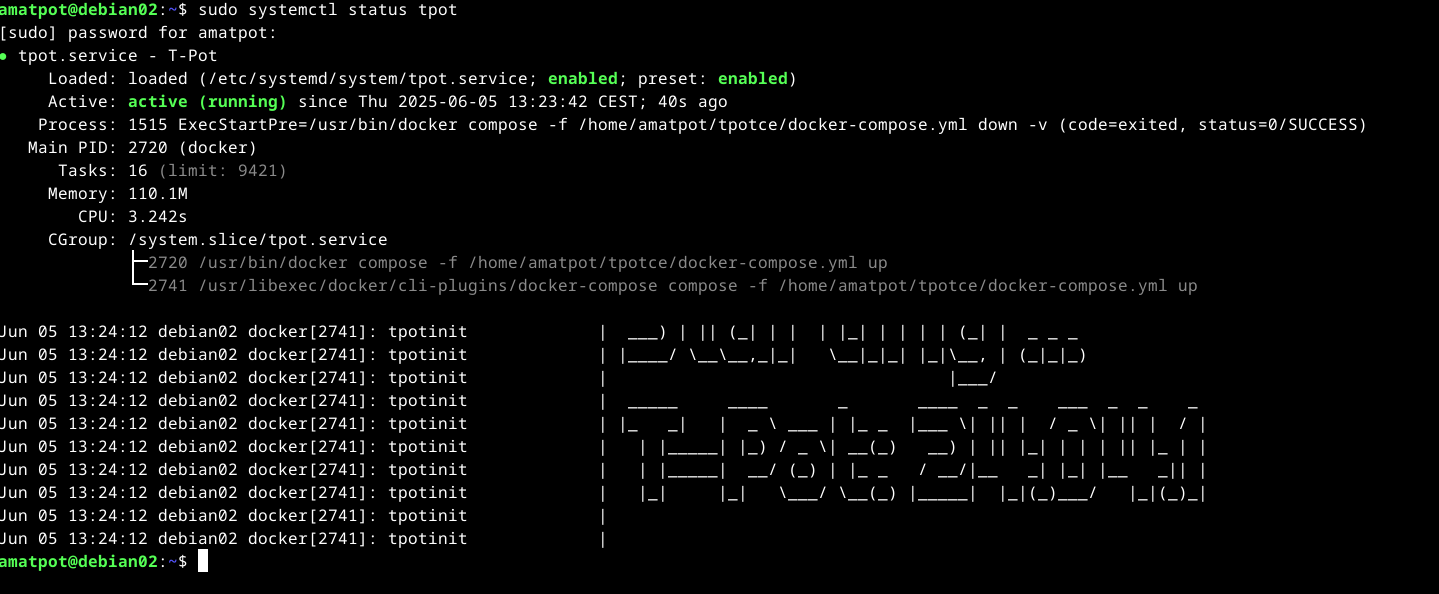
 

Instalaremos la T-Pot Standard / HIVE installation.



Asignaremos el user amatpot y el pass amatpot. Este tipo de credenciales tienen un nivel bajo de protección ya que solo haremos un acercamiento local a la herramienta y configuraciones antes del despliegue en el servidor.





Nota: instalar Kibata, instalar Elastic

Configurar los .yml de cada herramienta, puerto 9200 – iplocal

### Configuraciones Honeypot

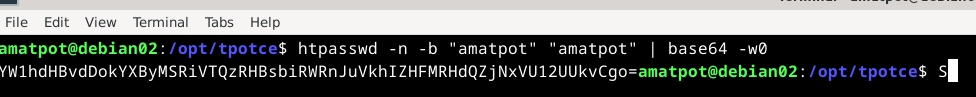
Para realizar las configuraciones de los Honeypot a utilizar, nos respaldaremos en el repositorio <https://github.com/telekom-security/tpotce>. Haremos uso de los siguientes elementos (explicar motivos y enfoque de la elección):

* Cowrie
* Maps
* Dionaea
* Conpot
* Elasticpot
* Elasticsearch
* Kibana
* Mailoney
* Honeytrap
* Suricata
* DDoSPot
* WordPot
* Spiderfoot

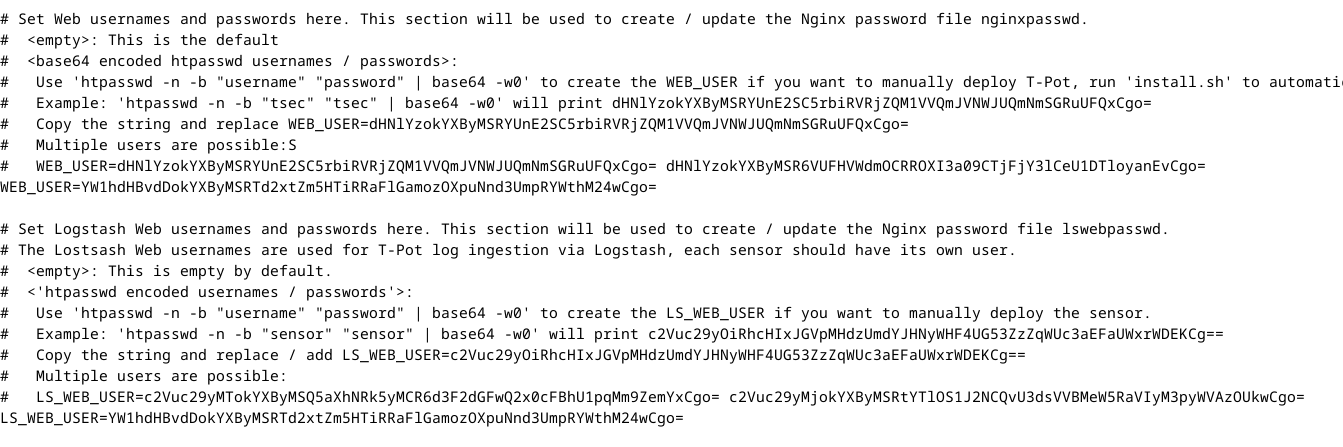
Previamente a las configuraciones, debemos definir el usuario WEB\_USER y LS\_WEB\_USER en el fichero oculto ~/tpotce/.env. Desde consola creamos el user y pass (utilizamos el mismo user para ambos usuarios):

(Todas las configuraciones se realizan con usuario ordinario)

htpasswd -n -b "amatpot" "amatpot" | base64 -w0



Reemplazamos la información en el fichero mencionado.



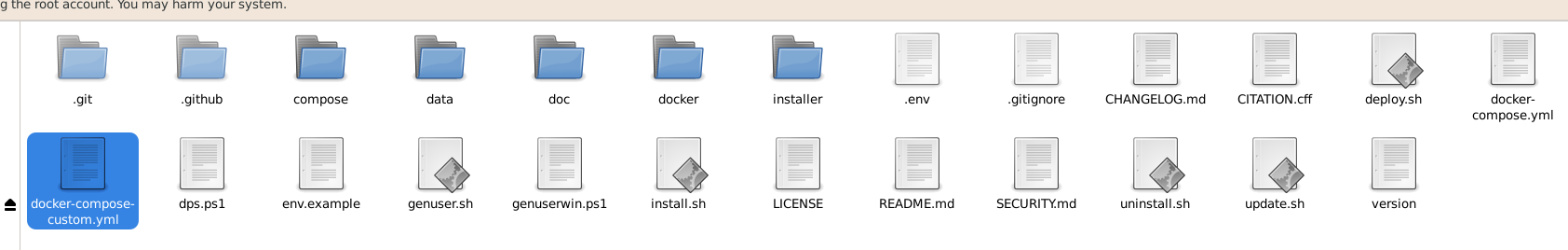
Ya realizada la configuración, lanzaremos desde el directorio ~/tpotce/compose el script customizer.py

sudo python3 customizer.py

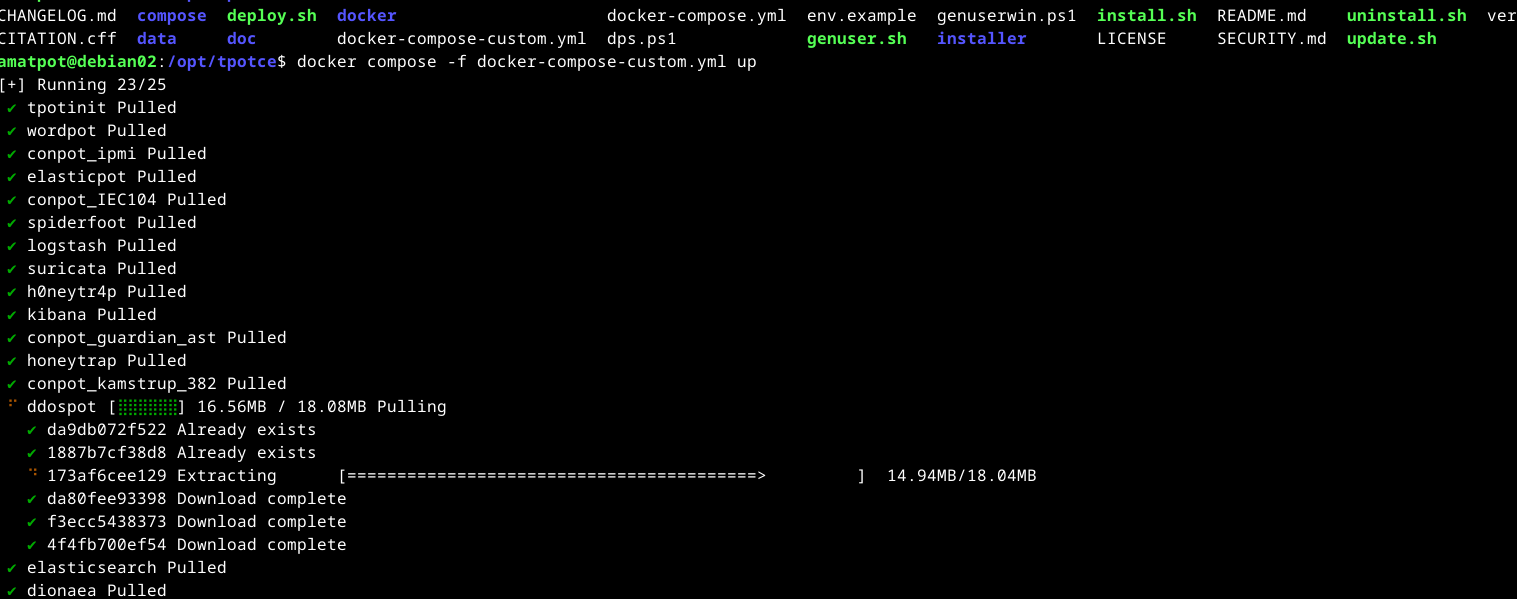


Ya creado el fichero customizado, lo copiamos en el directorio ~/tpotce

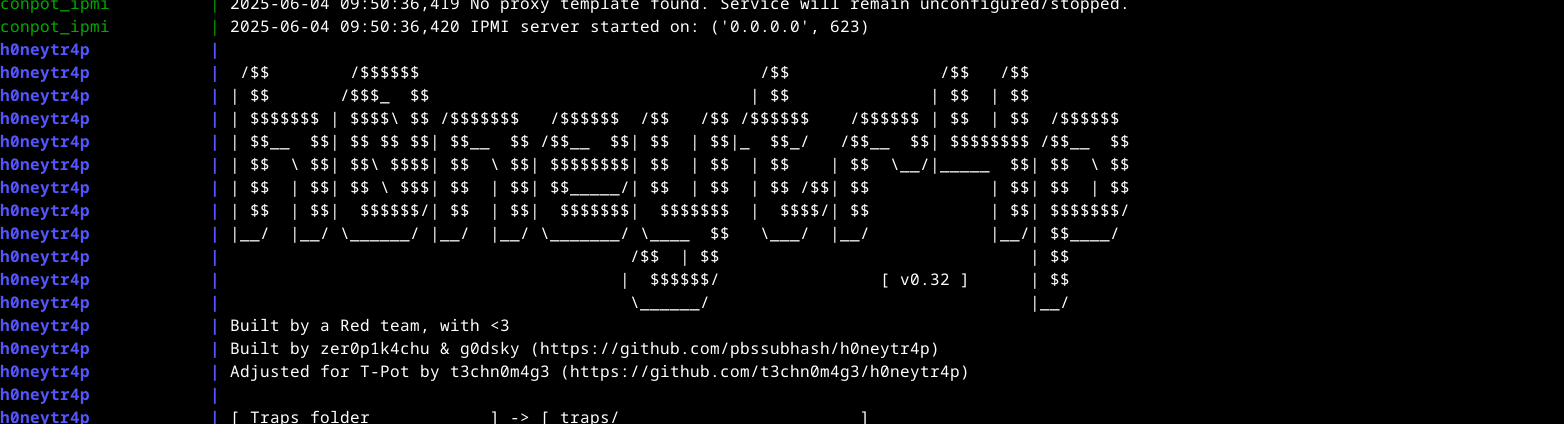
sudo cp docker-compose-custom.yml ~/tpotce/

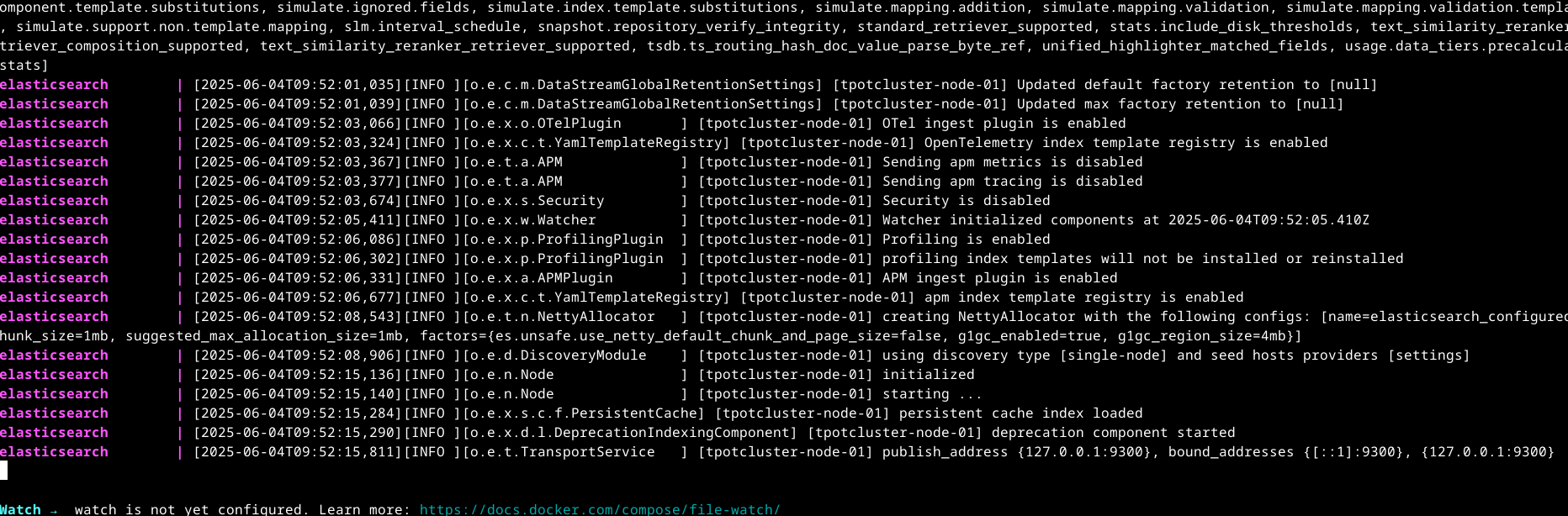


Nos posicionamos en el directorio donde colocamos el fichero y lanzamos el comando docker compose -f docker-compose-custom.yml up para revisar que funciona de manera correcta



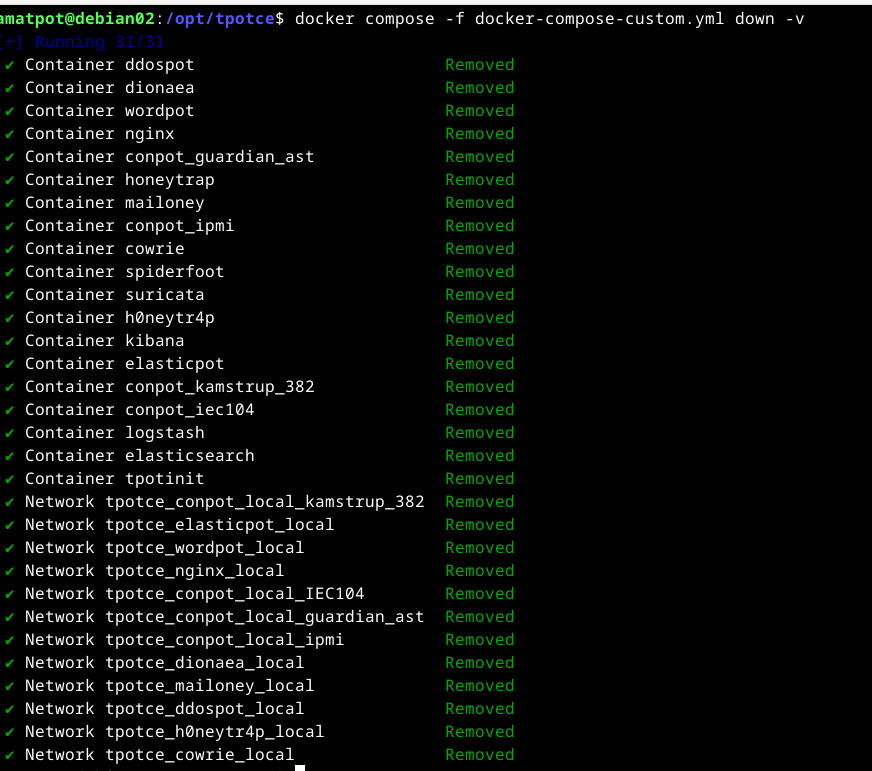


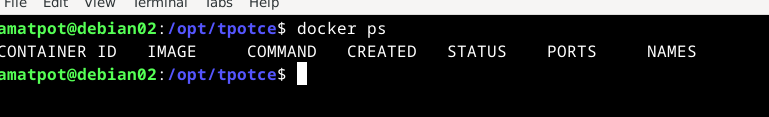




Docker irá lanzando el contenedor de cada Honeypot seleccionado. Considerando que las modificaciones se realizaron de manera correcta, interrumpimos el proceso y detenemos los contenedores con el comando

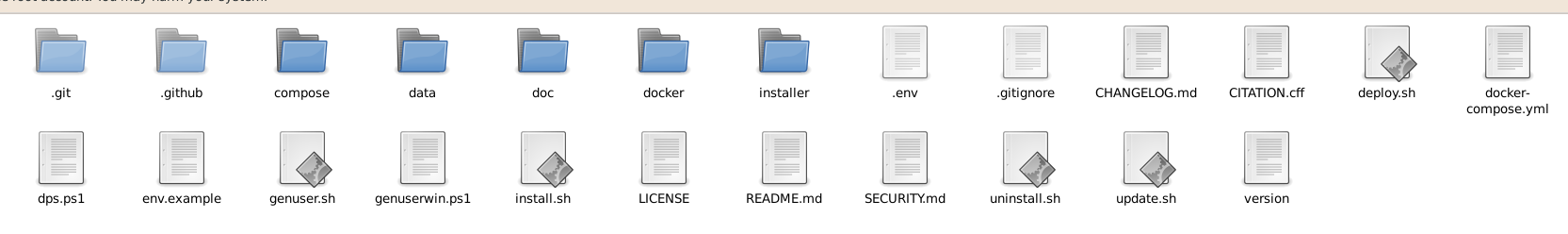
docker compose -f docker-compose-custom.yml down -v



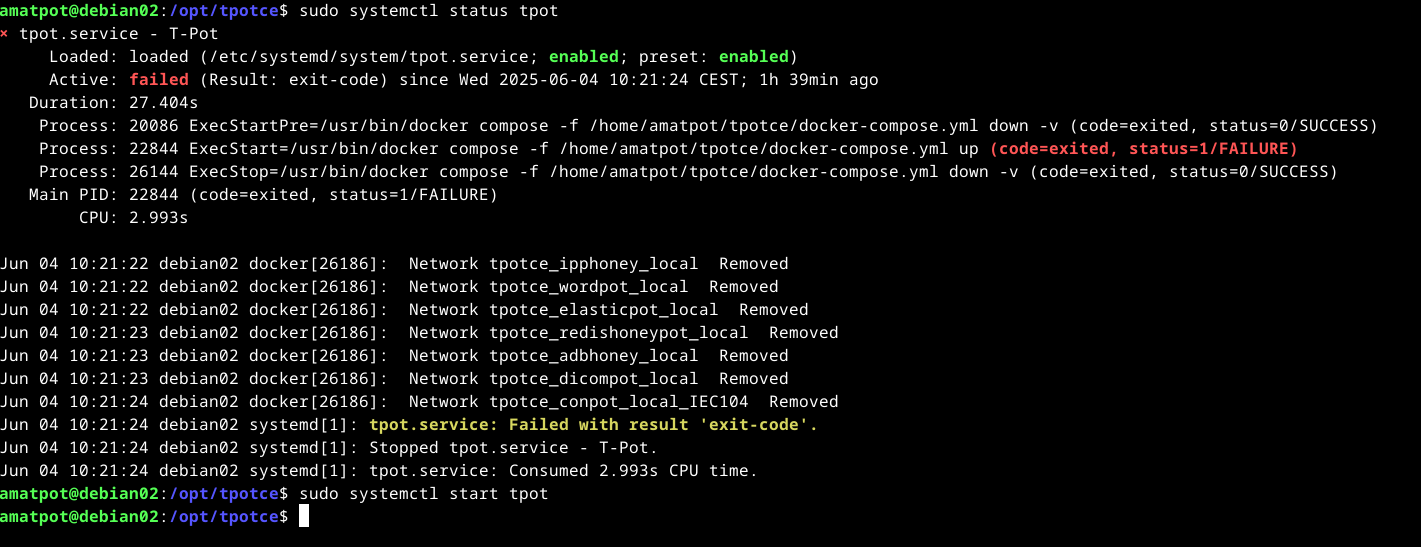


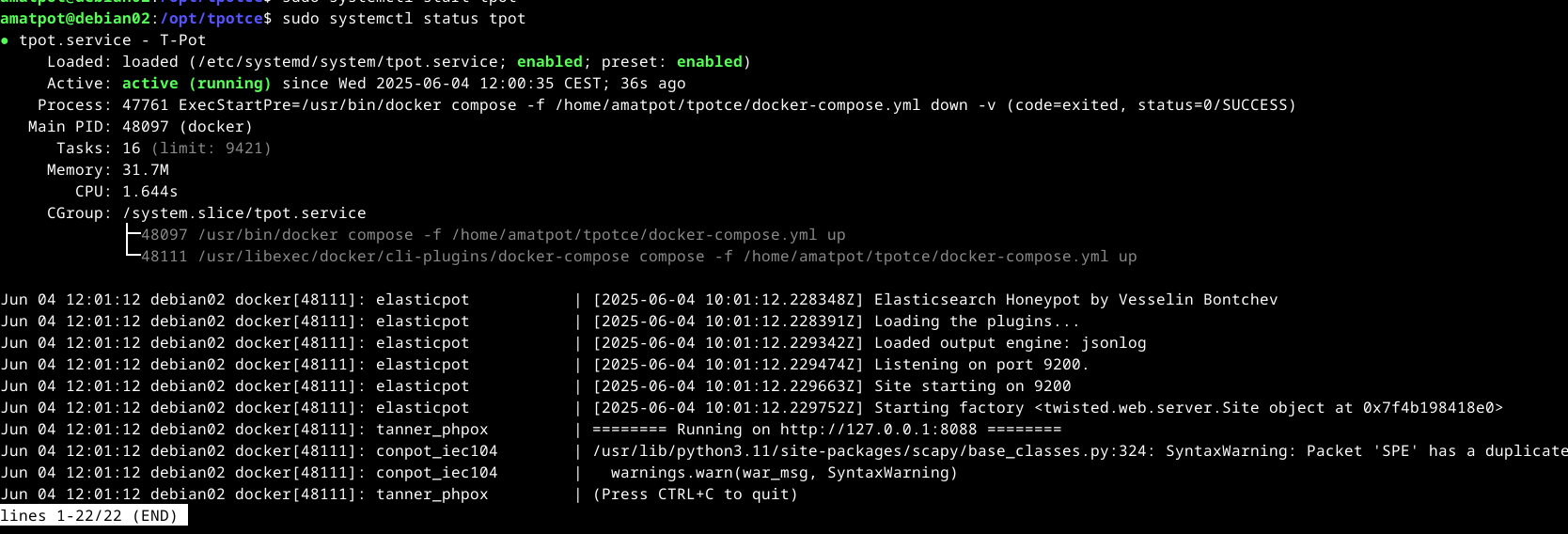
En este punto reemplazamos el fichero docker-compose.yml por las configuraciones personalizadas.

sudo mv docker-compose-custom.yml docker-compose.yml

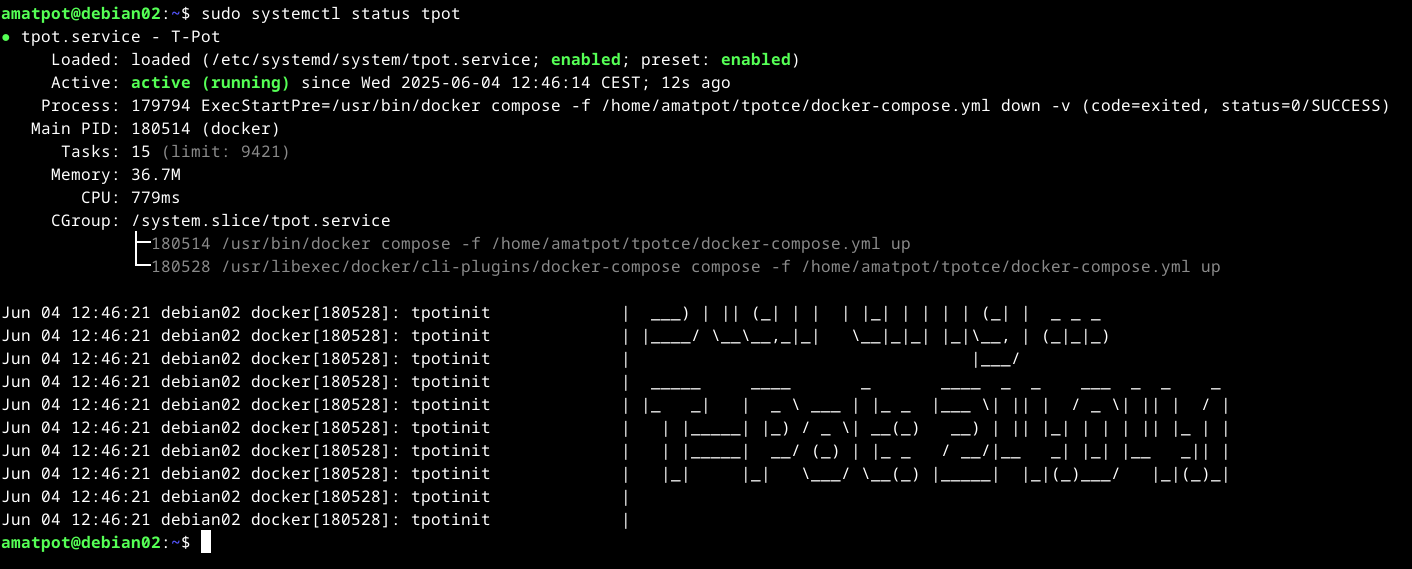


Validamos el estado actual de Tpot y lo iniciamos



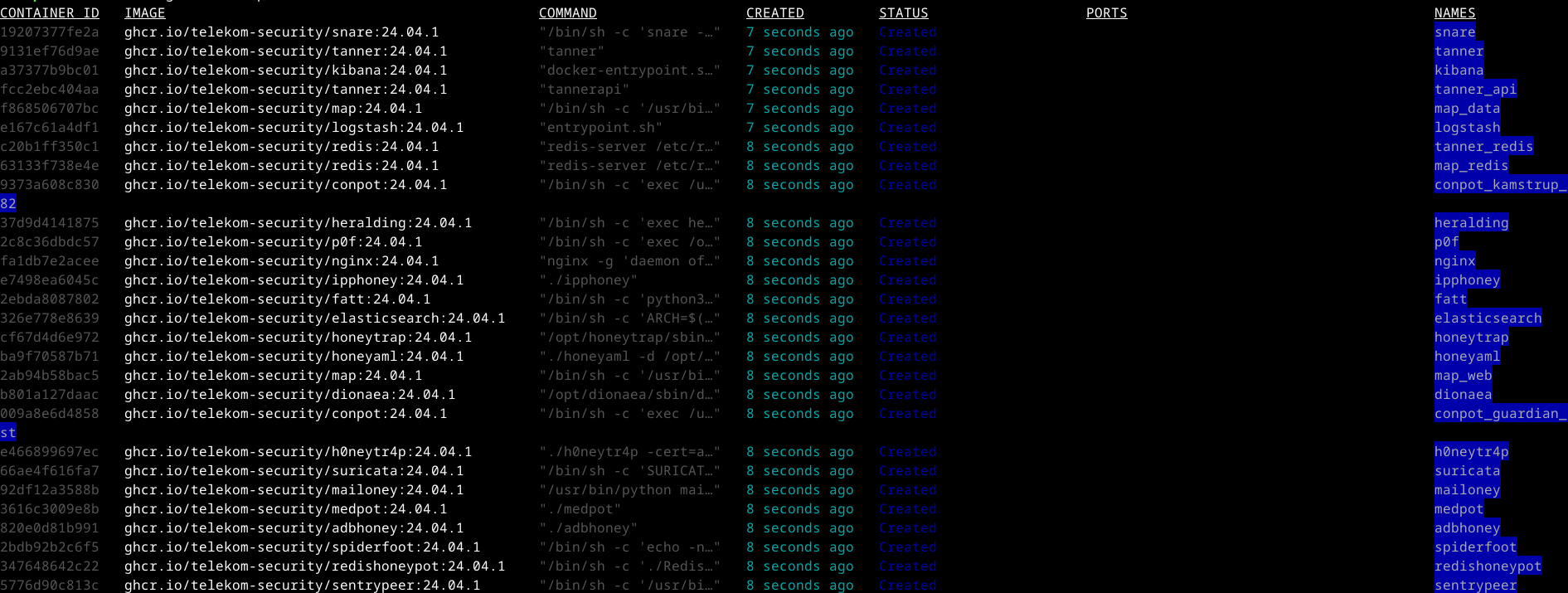


Realizamos un reinicio del sistema para el impacto correcto de las configuraciones sudo reboot

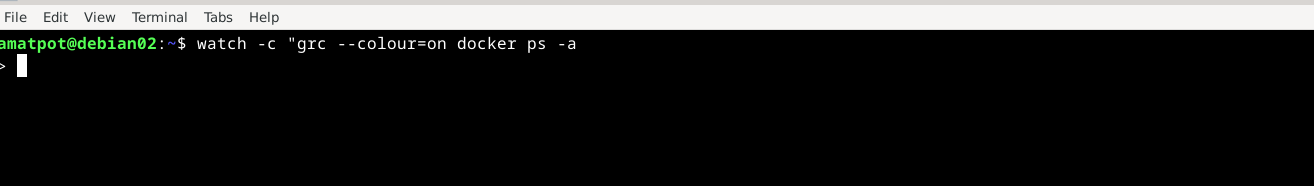


Una vez activo Tpot, podremos revisar los contenedores activos con el comando

grc docker ps -a

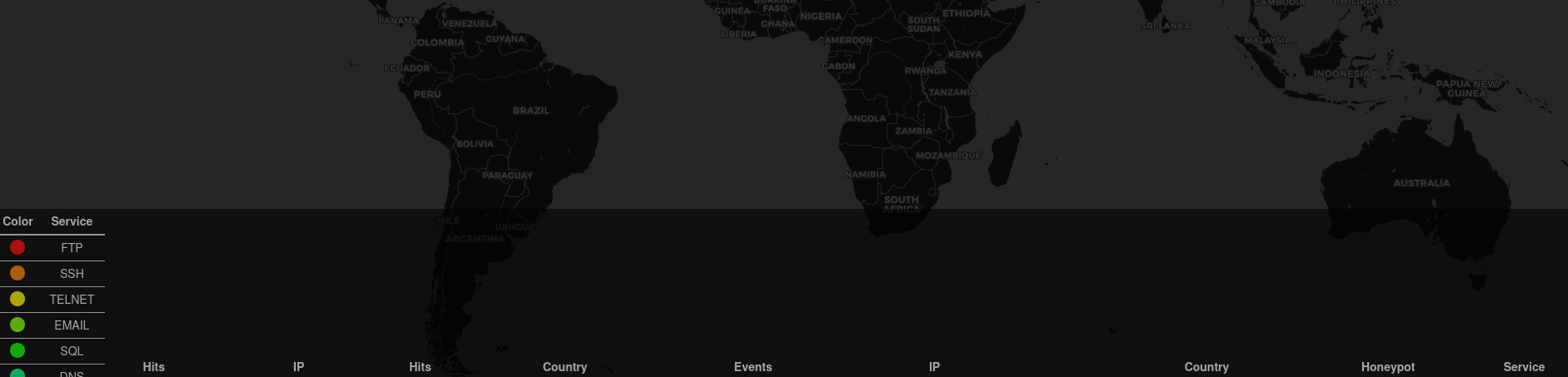


También podremos monitorizar en directo con watch -c "grc --colour=on docker ps -a



De manera local, desde el navegador, podremos acceder al apartado Web desde el siguiente enlace https://192.168.142.131:64297/ (ip local)





**Configuraciones en servidor AWS**

<https://eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#InstanceDetails:instanceId=i-0d55b7cab8d388425>

Correo: amapothoney@gmail.com

Nombre: Amapot

Contraseña: KeepCoding2025

**Gmail**

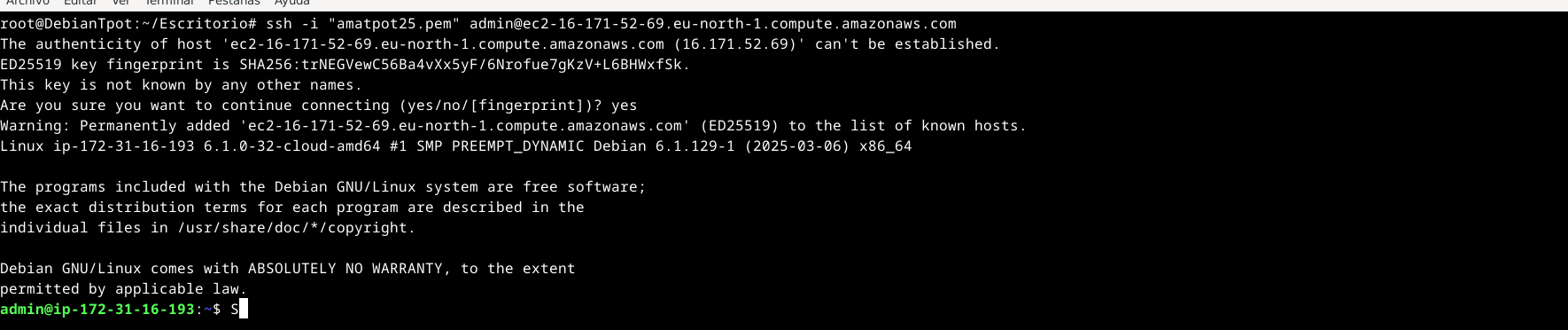
amapothoney@gmail.com

KeepCoding2025

--

Conectarse desde SSH

root@DebianTpot:~/Escritorio# ssh -i "amatpot25.pem" [admin@ec2-16-171-52-69.eu-north-1.compute.amazonaws.com](mailto:admin@ec2-16-171-52-69.eu-north-1.compute.amazonaws.com)



Configuración AWS 2

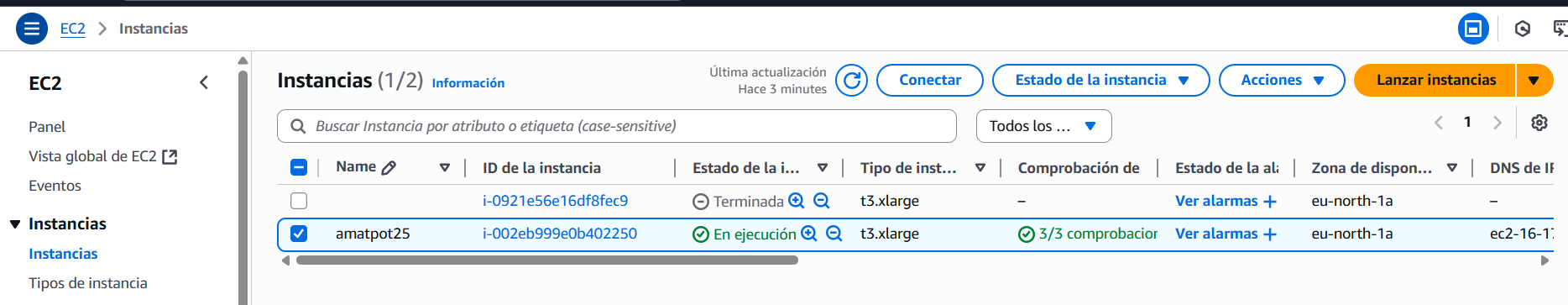
amatpot25.v2 > **i-0d55b7cab8d388425 (amatpot25.v2)**

Usuario ordinario Debian

amatpot > Am4tp07.25

Usuario Tpot

amatpot > Am4tp07.25



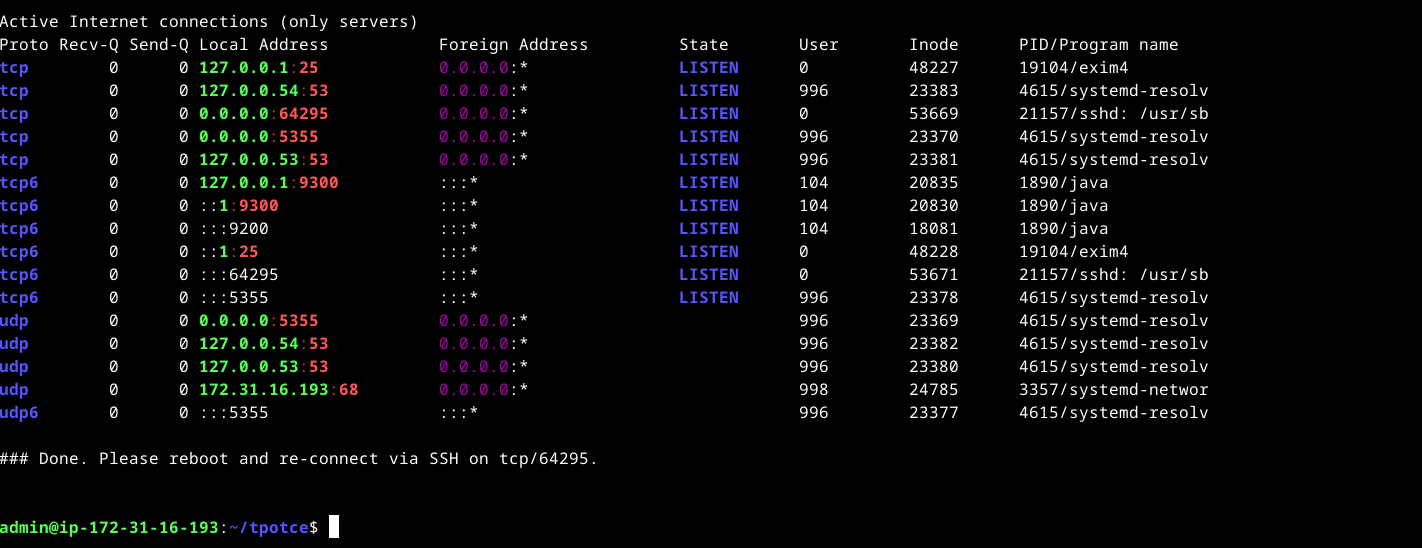
Instancia configurada

Debian en AWS ya tiene instalado elastic, kibana, Docker. Tpot

Credenciales Tpot

User: amatpot

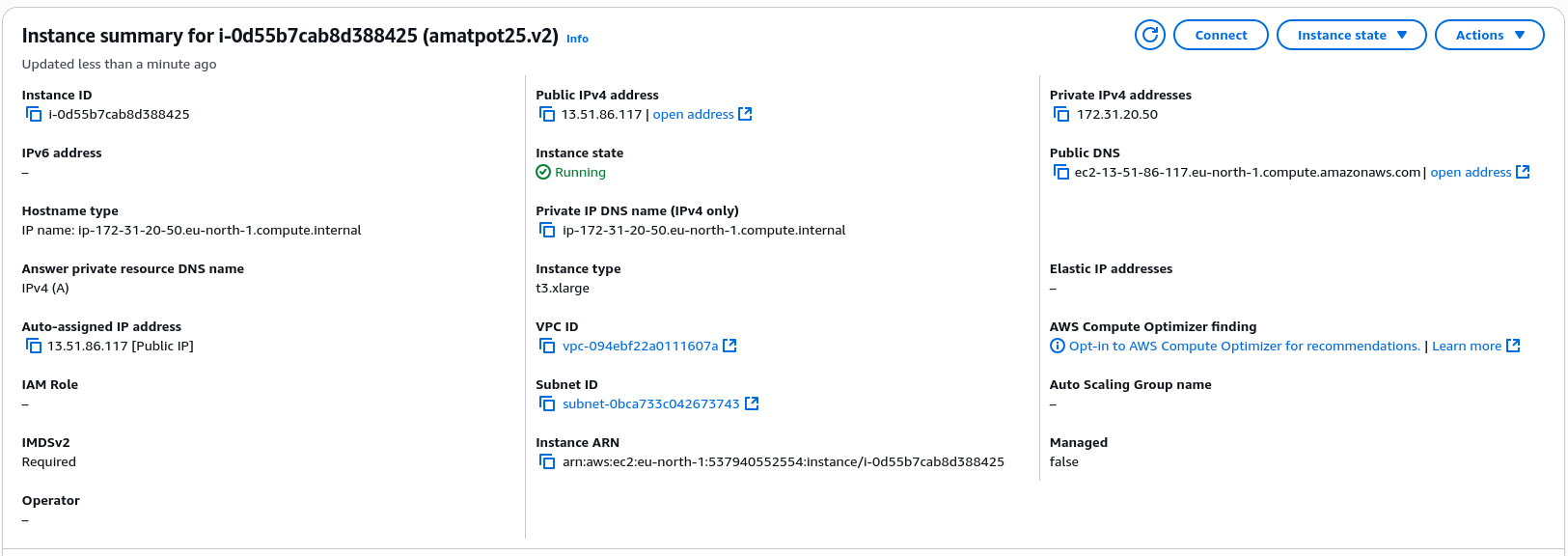
Pass: Am4tp07.25



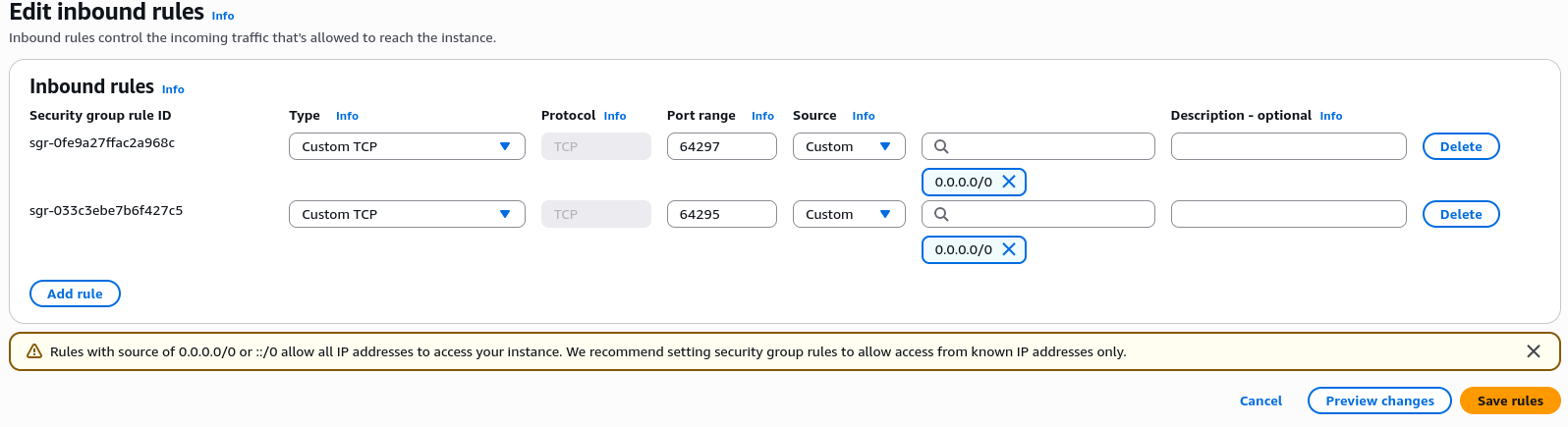
ssh -i "amatpot25.pem" [admin@](mailto:admin@16.171.52.69)ip-publica -p 64295

Evidencias configuraciones AWS

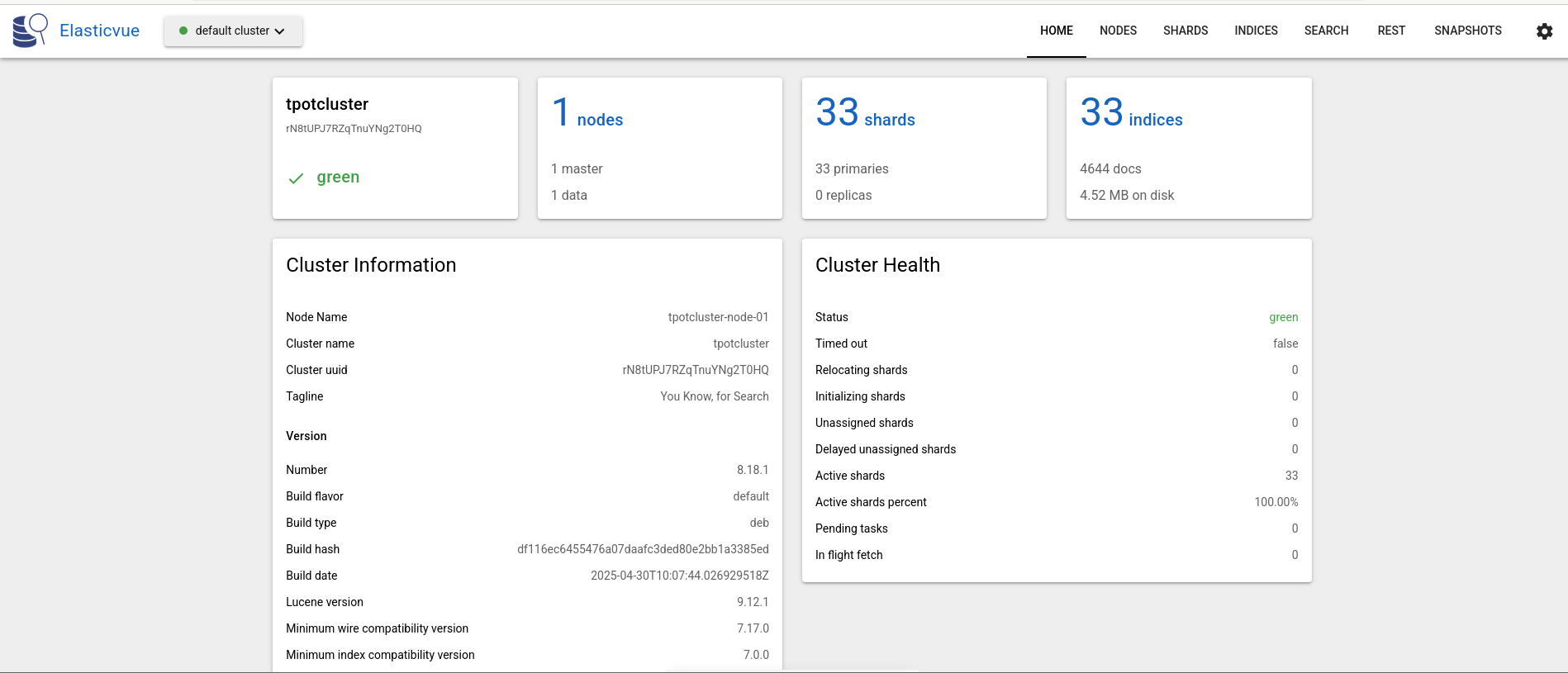
Instance summary for i-0d55b7cab8d388425 (amatpot25.v2)



Asignamos los puertos 64297 y 64295 para gestionar las primeras configuraciones de Tpot, evitando el tráfico externo.



Instalamos Elastic, Kibana, Docker y las dependencias necesarias.



<https://13.62.58.126:64297/elasticvue/cluster/0/nodes>

Configuraciones de Tpot



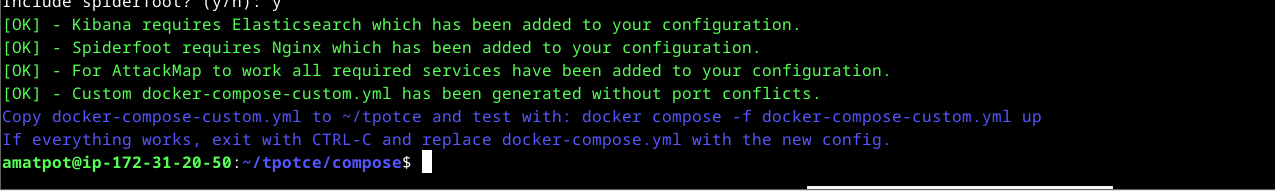
Ampliamos memoria



### Configuraciones Honeypot.v2:

* Cowrie
* Maps
* Dionaea
* Conpot
* Elasticpot
* Elasticsearch
* Kibana
* Mailoney
* Honeytrap
* Suricata
* DDoSPot
* WordPot
* Spiderfoot

Sin conflictos en la primera config



Validamos el funcionamiento de los Dockers sin conflicto de puertos