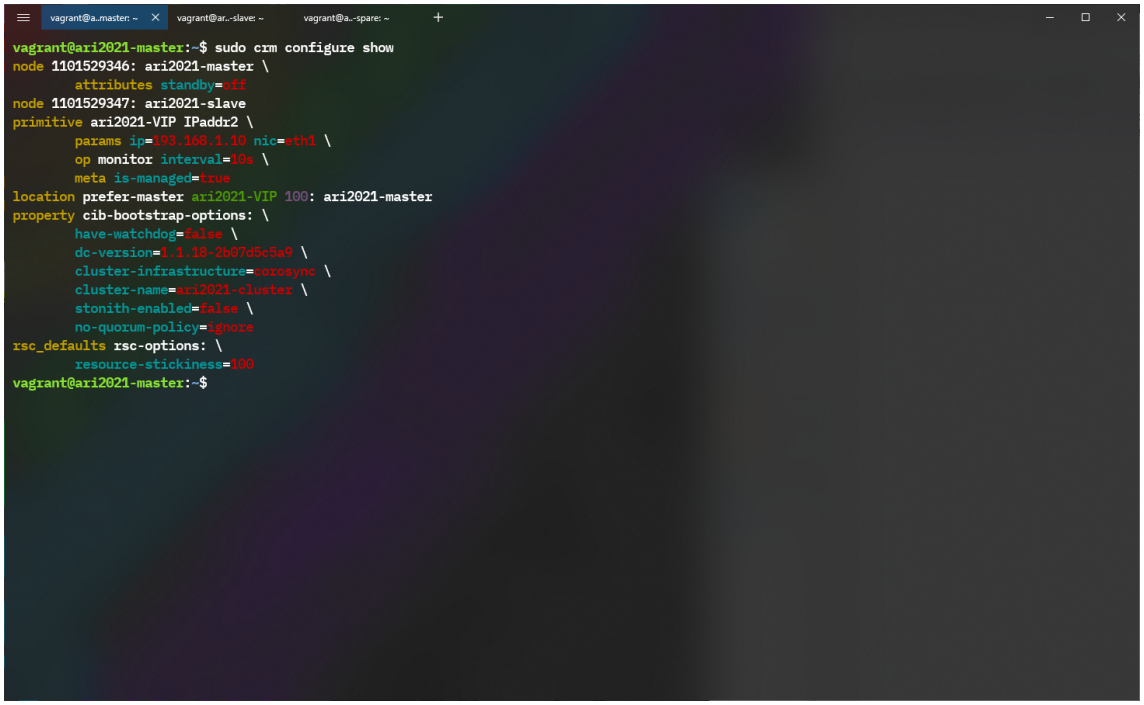
MEMORIA EII – PRÁCTICA I

Alonso Rodríguez Iglesias – EII 2020/21 – Práctica 1

JUSTIFICACIÓN DE LA PRIMERA PARTE

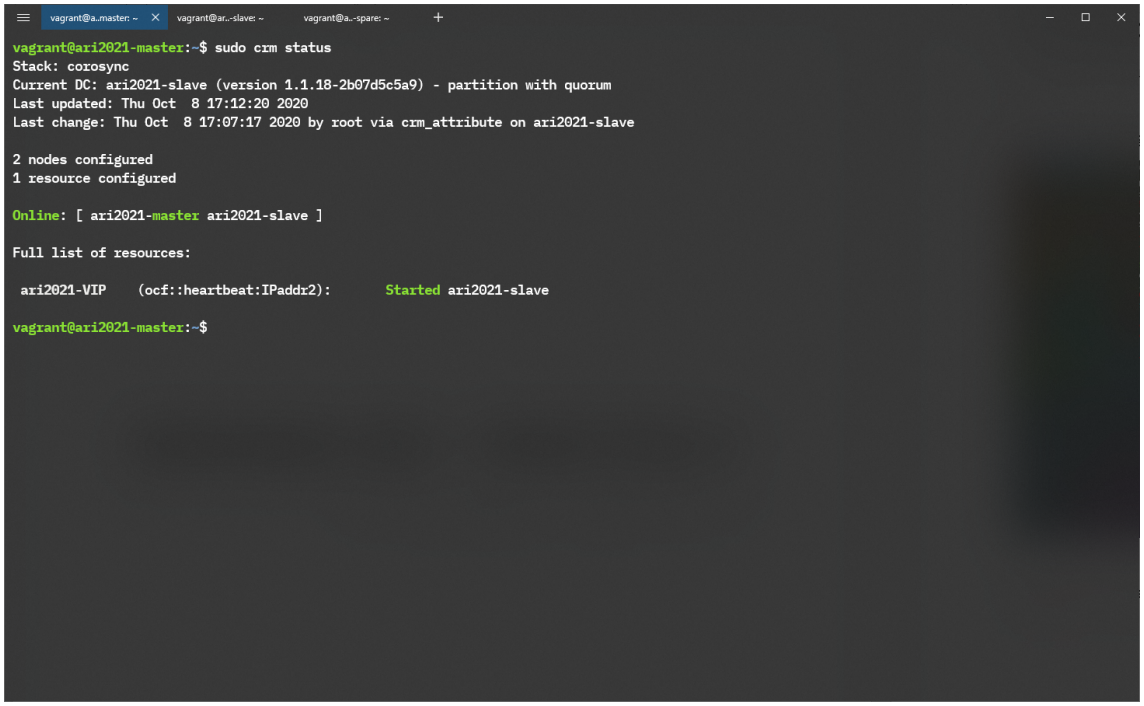
Se adjuntan capturas de la ejecución (en orden) de los siguientes comandos:

`sudo crm configure show`

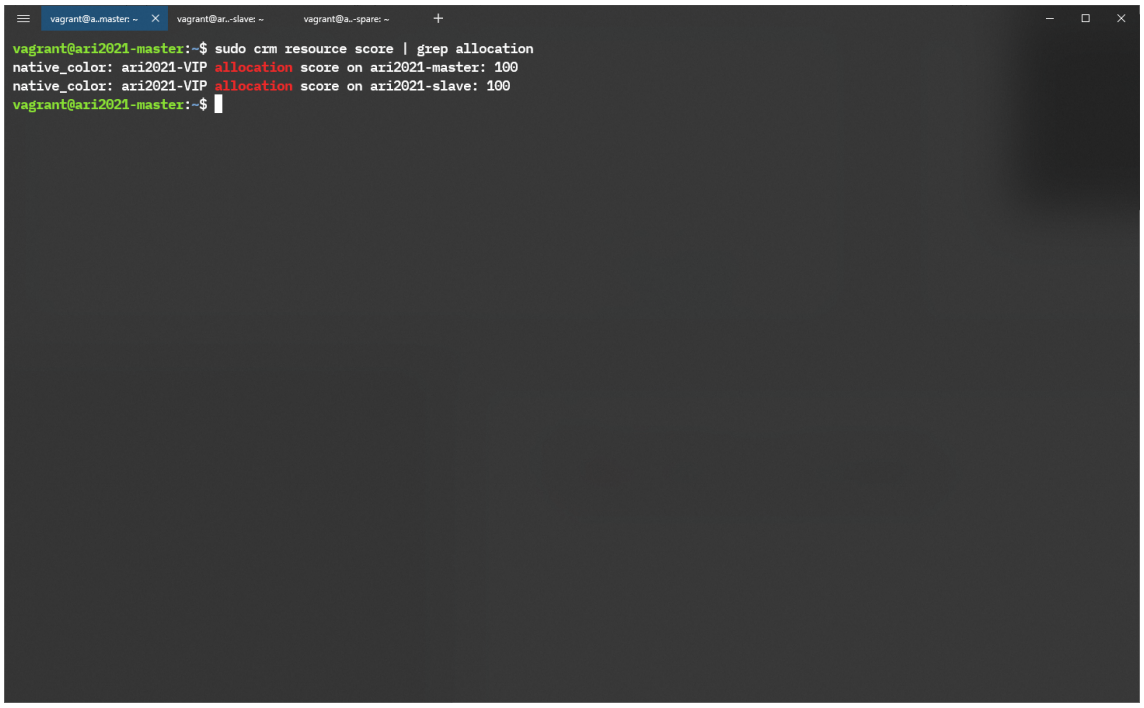


(Mostramos la configuración actual del cluster)

`sudo crm status`

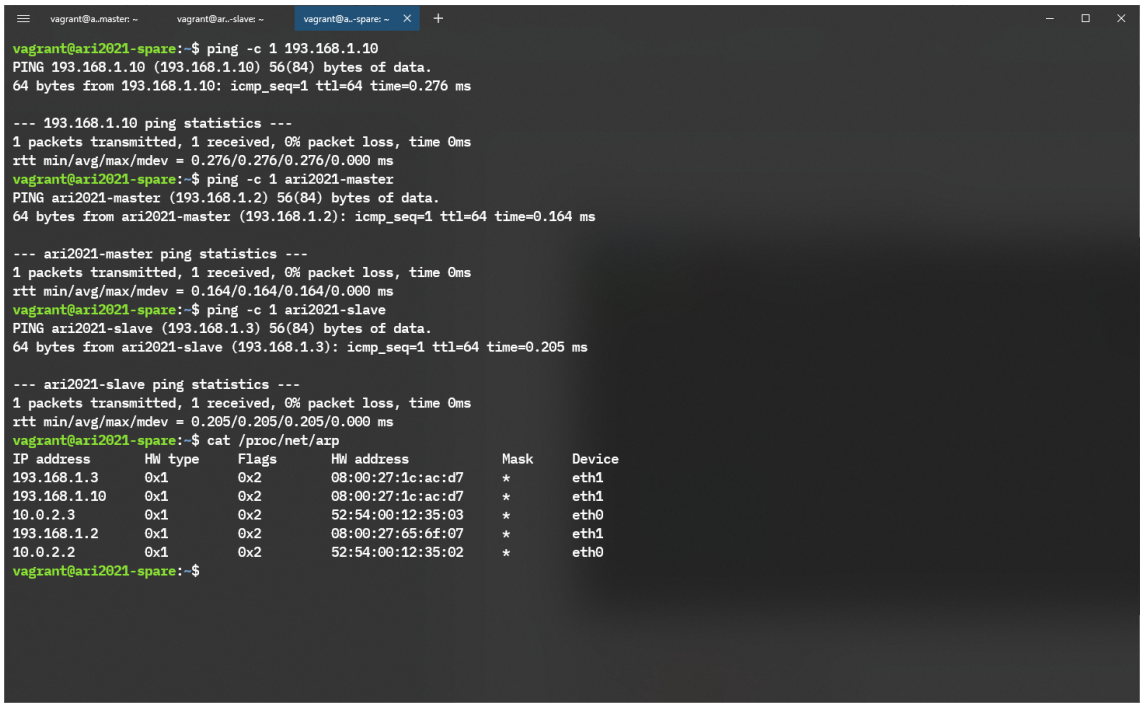


sudo crm resource score | grep allocation



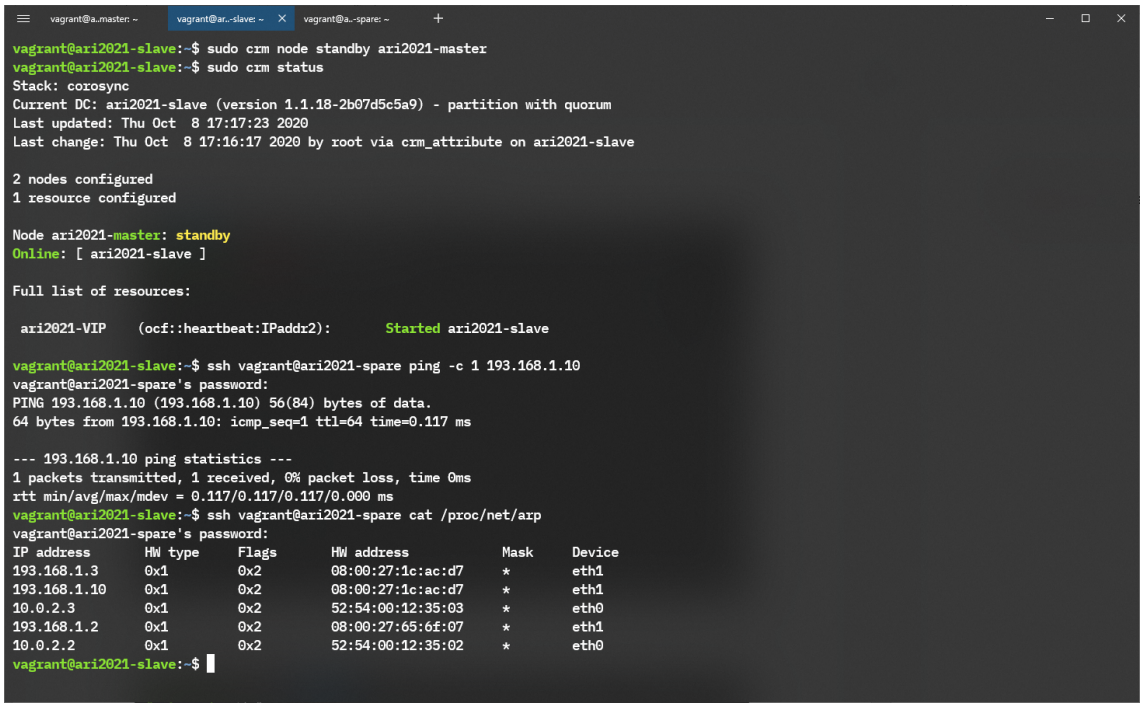
(Vemos que temenos 100 de allocation en cada nodo, 100 de preferencia en master, y 100 de adherencia en slave)

spare$ {comprobación del arp}



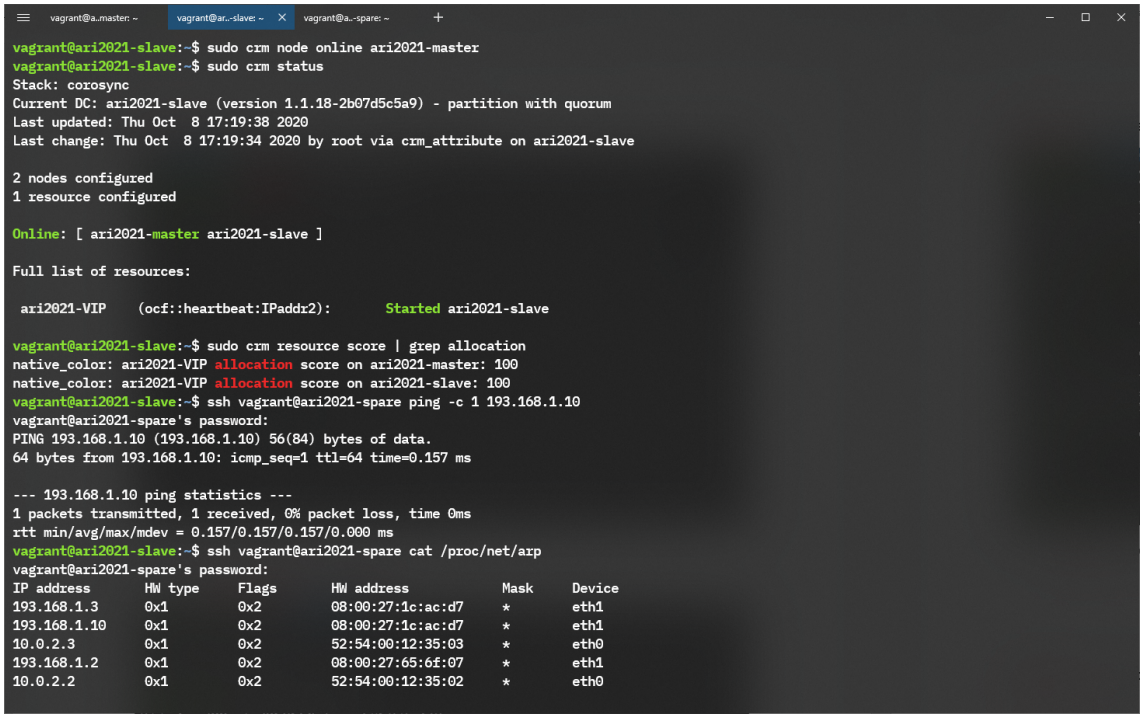
(la MAC de 193.168.1.10 es la del slave, ya que como en el último punto de la práctica configuramos la stickiness a 100, nunca llegó a cambiarse a master)

sudo crm node standby ari2021-master [...]



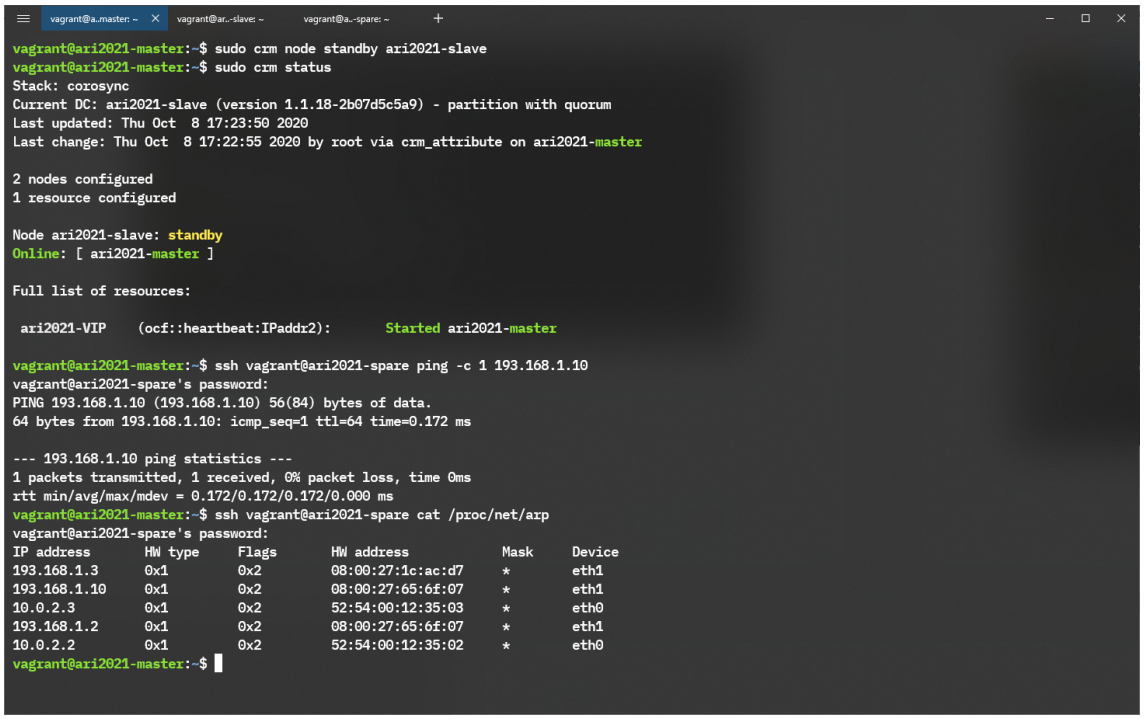
(la MAC de 193.168.1.10 sigue siendo la del slave, porque solo hemos puesto master en standby)

sudo crm node online ari2021-master [...]



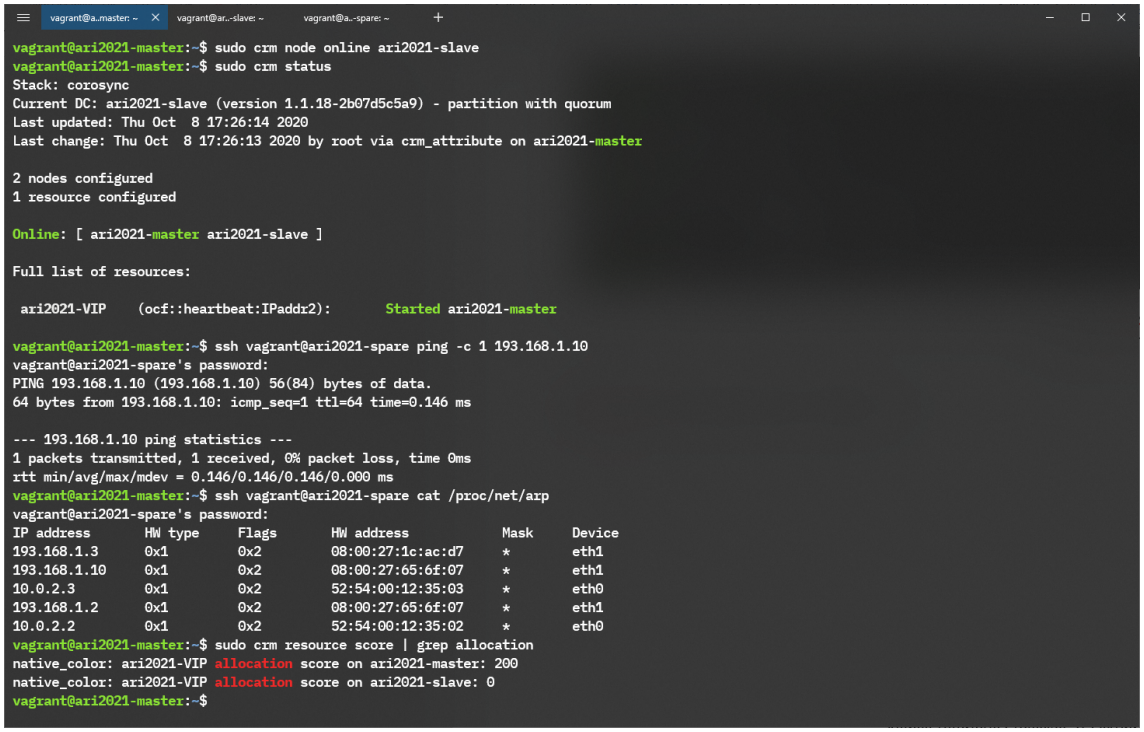
(la MAC sigue sin cambiar aunque encendamos de nuevo master)

sudo crm node standby ari2021-slave [...]



(ahora sí, coinciden las MAC de master y de la IP virtual, ya que al apagar el slave, no ha tenido otra que mover la IP al master)

sudo crm node online ari2021-slave [...]



(encendemos el slave, pero de nuevo, la MAC no cambia debido a la stickiness, es decir, que la IP virtual queda apuntando al master, hasta que le ocurra algo a éste)

Algo que es interesante mencionar aquí es ver cómo **ari2021-master** tiene ahora un score de 200 de allocation, esto es porque, de por sí, con el comando:

sudo crm configure location prefer-master ari2021-VIP 100: ari2021-master

configuramos una preferencia de 100 al master, y además, tiene los 100 de stickiness de la IP virtual. Esto es porque **allocation = preference + adherence**.

JUSTIFICACIÓN DE LA SEGUNDA PARTE

Los comandos ejecutados han sido los siguientes (tras la eliminación del recurso antiguo):

```
master$ sudo crm configure primitive ari2021-VIP1 ocf:heartbeat:IPaddr2 \
        params ip="193.168.1.10" nic="eth1" \
        op monitor interval="10s" \
        meta is-managed="true"
```

```
slave$ sudo crm configure primitive ari2021-VIP2 ocf:heartbeat:IPaddr2 \
       params ip="193.168.1.11" nic="eth1" \
       op monitor interval="10s" \
       meta is-managed="true"
```

```
master$ sudo crm configure location prefer-master ari2021-VIP1 100: ari2021-master
```

```
master$ sudo crm configure location prefer-slave ari2021-VIP2 100: ari2021-slave
```

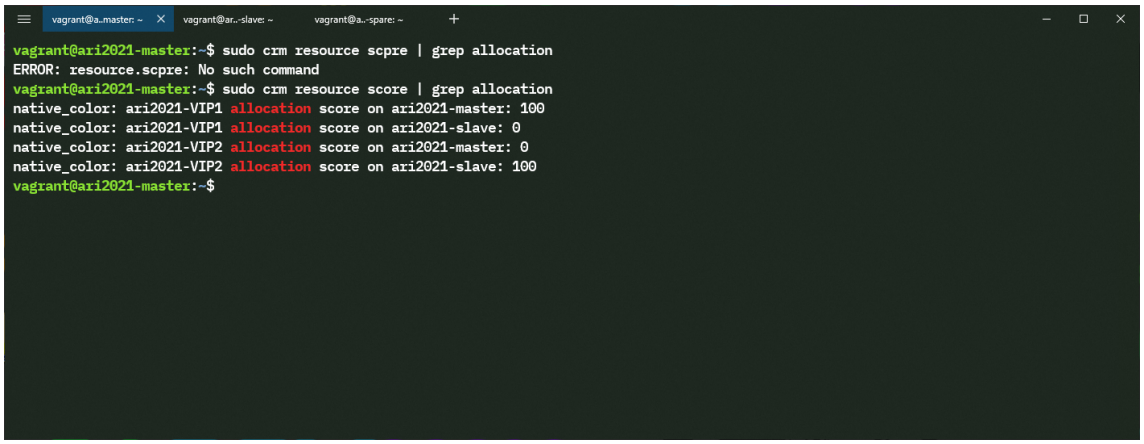
```
master$ sudo crm configure rsc_defaults resource-stickiness=0
```

Se adjuntan capturas de la ejecución (en orden) de los siguientes comandos:

sudo crm configure show && sudo crm status

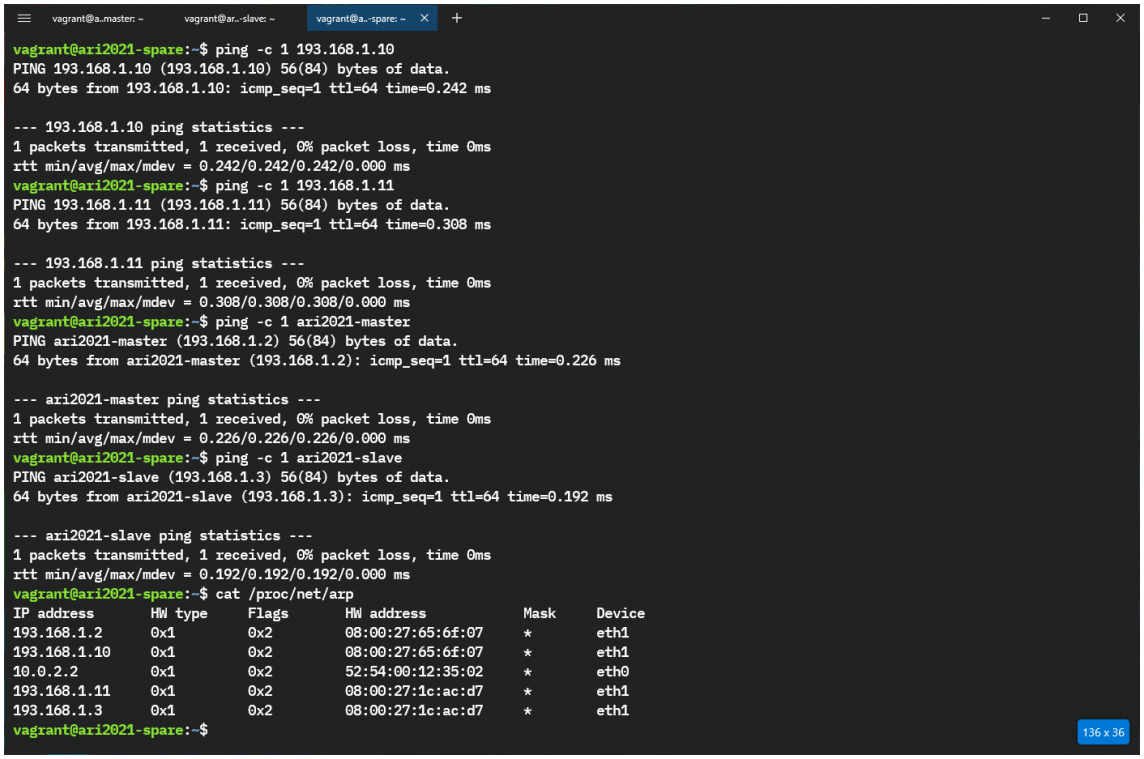
(Mostramos la configuración actual del cluster así como su estado)

sudo crm resource score | grep allocation



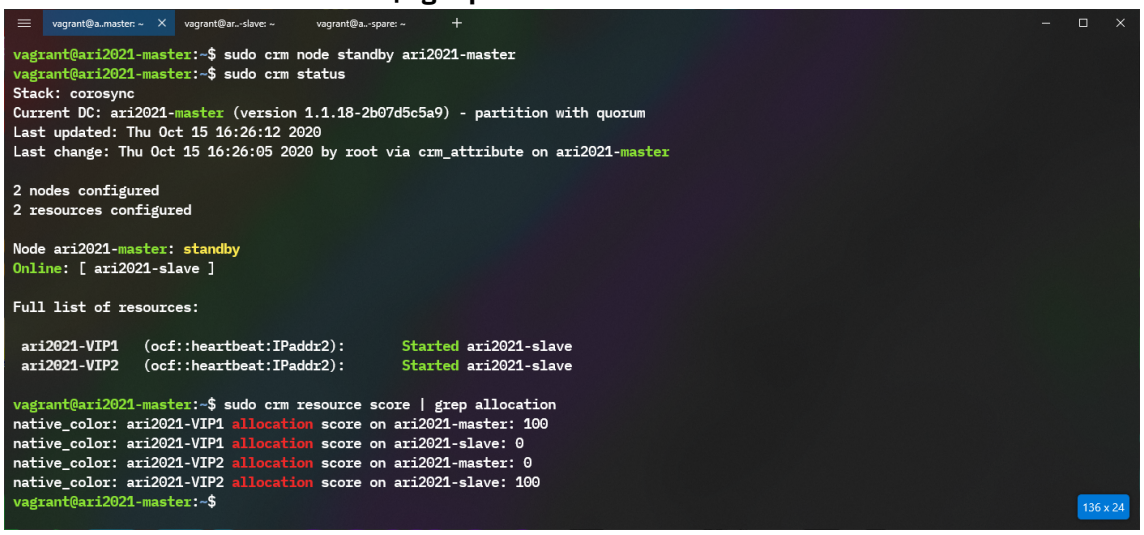
(Afectivamente, la allocation está a 100 en cada nodo, por el que tiene preferencia cada VIP, ya que la adherence está en 0)

ping -c 1 193.168.1.10 &&
ping -c 1 193.168.1.11 &&
ping -c 1 ari2021-master &&
ping -c 1 ari2021-slave &&
cat /proc/net/arp



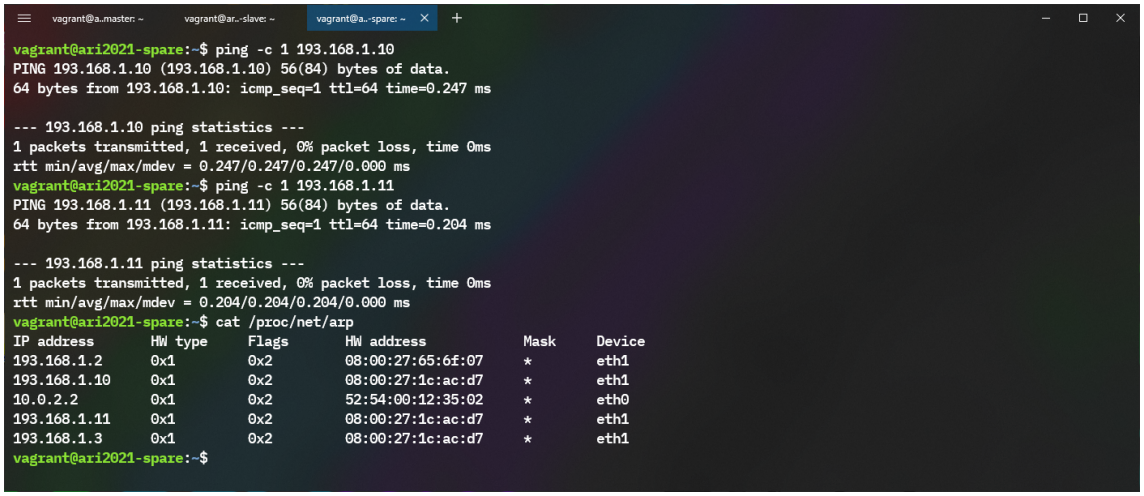
(Hacemos ping para poblar la ARP caché, y comprobamos que la MAC de -master se corresponde a la VIP1, así como la de -slave a la VIP2)

sudo crm node standby ari2021-master &&
sudo crm status &&
sudo crm resource score | grep allocation



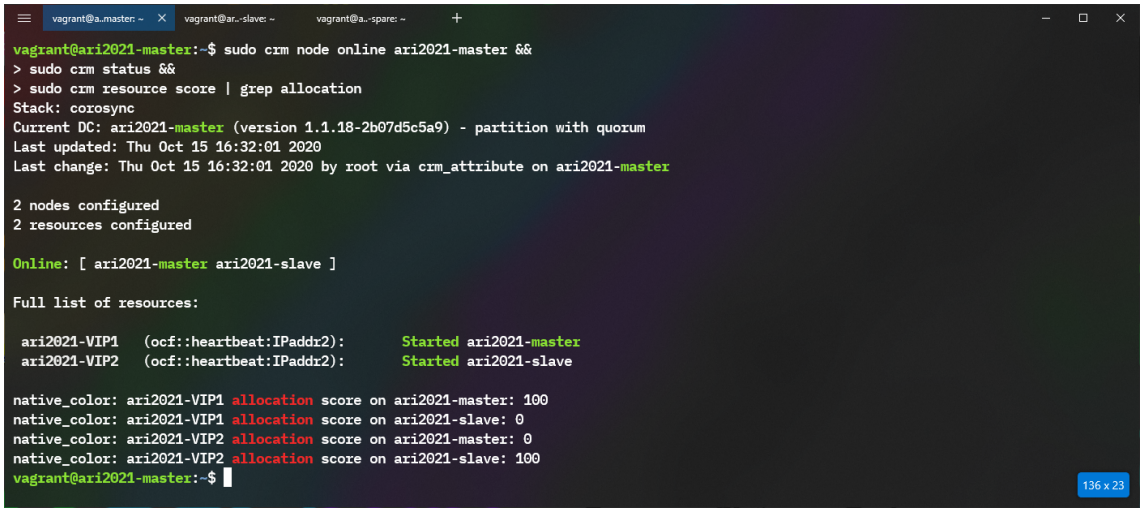
(Ponemos -master en standby y vemos qué ocurre con la VIP1)

ping -c 1 193.168.1.10 &&
ping -c 1 193.168.1.11 &&
cat /proc/net/arp



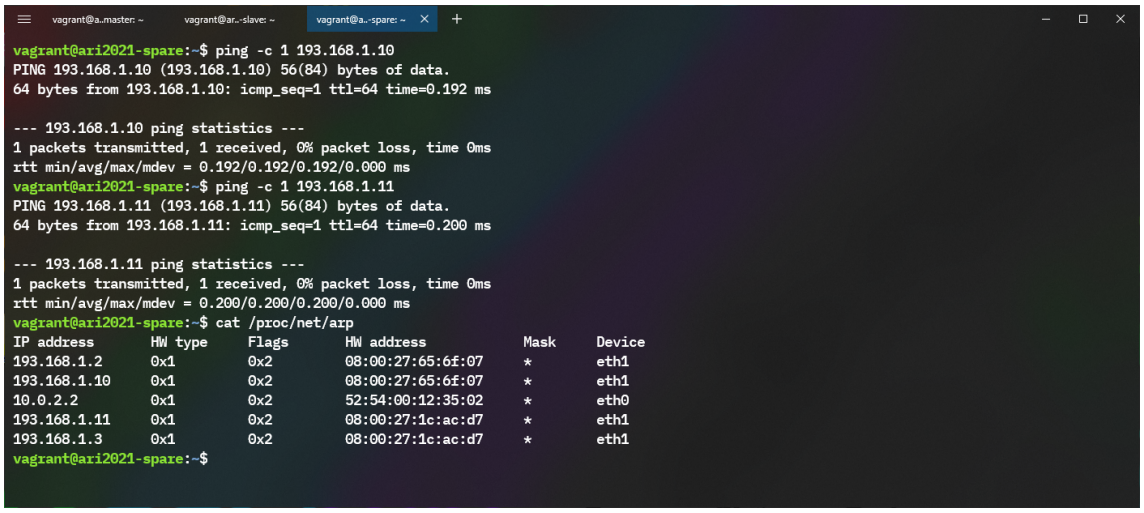
(Podemos comprobar que se ha migrado a -slave)

sudo crm node online ari2021-master &&
sudo crm status &&
sudo crm resource score | grep allocation



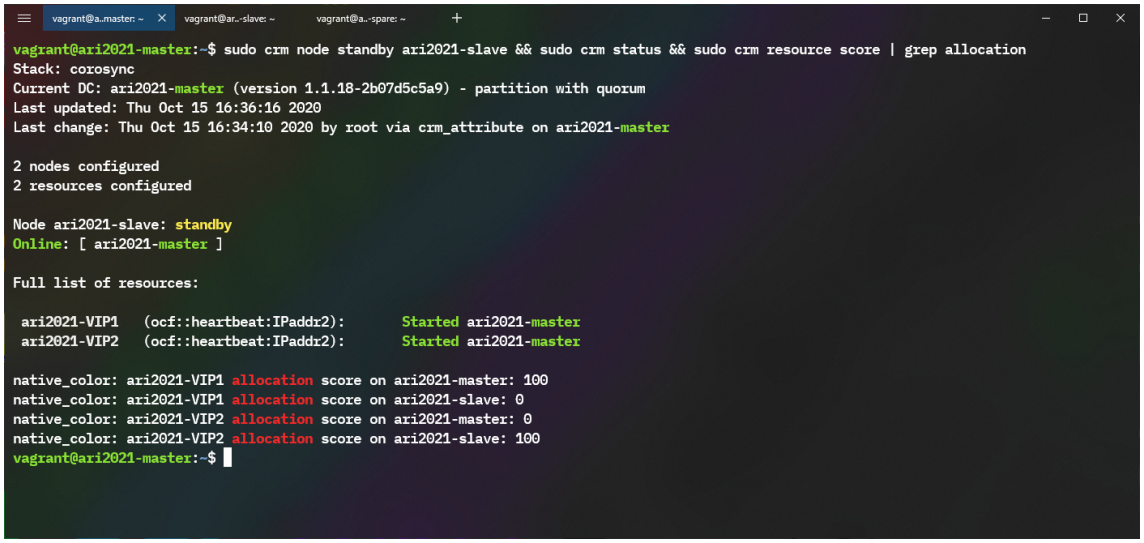
(Ponemos -master de vuelta online y vemos qué ocurre con la VIP1)

ping [...] && cat /proc/net/arp



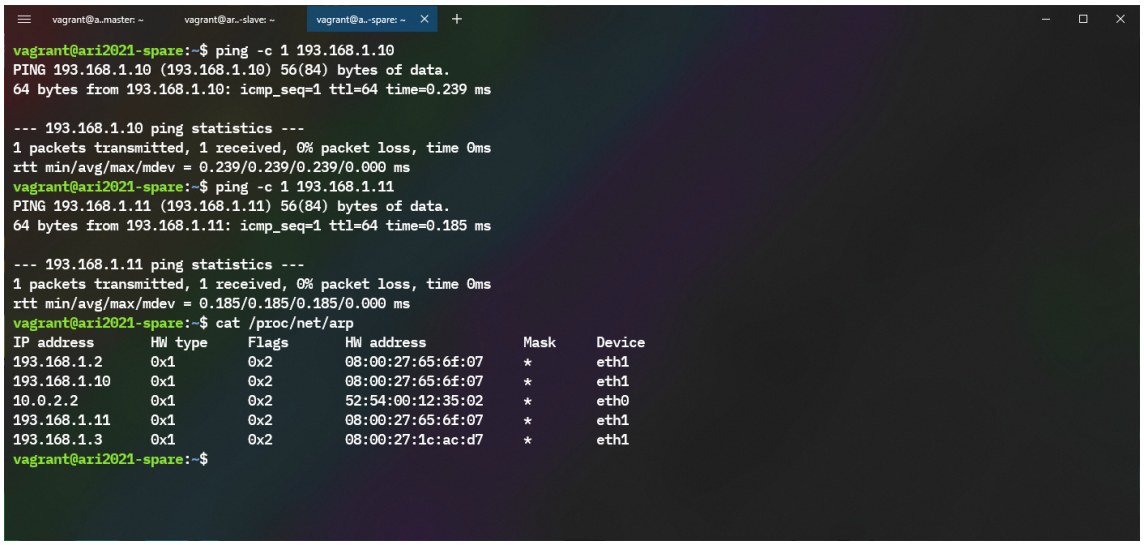
(Podemos comprobar que ha vuelto a -master)

sudo crm node standby ari2021-slave &&
sudo crm status &&
sudo crm resource score | grep allocation



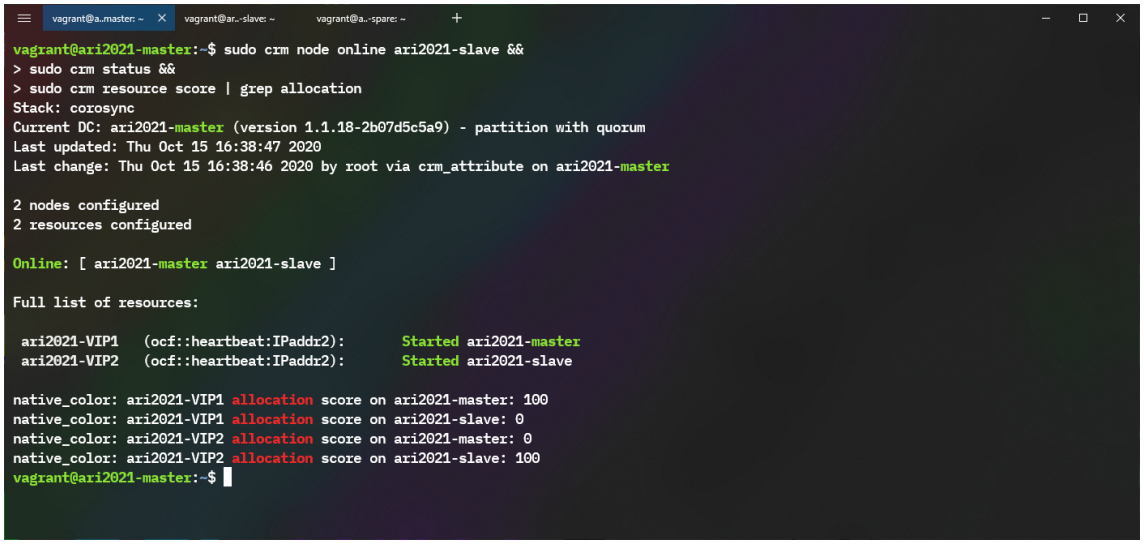
(Ponemos -slave en standby y vemos qué ocurre con la VIP2)

ping [...] && cat /proc/net/arp



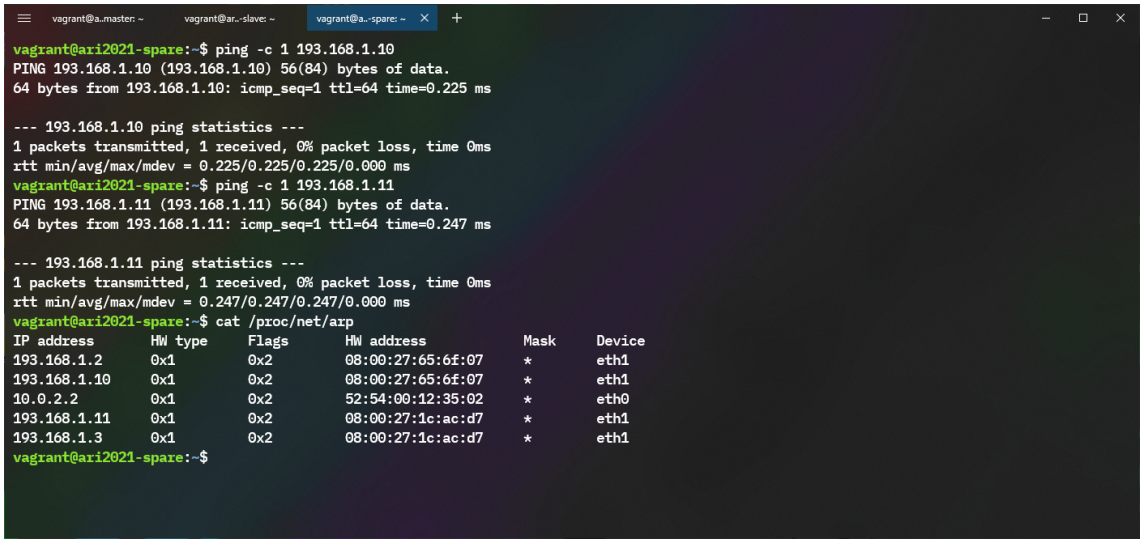
(Podemos comprobar que se ha migrado a -master)

sudo crm node online ari2021-slave &&
sudo crm status &&
sudo crm resource score | grep allocation



(Ponemos -slave de vuelta online y vemos qué ocurre con la VIP2)

ping [...] && cat /proc/net/arp



(Podemos comprobar que ha vuelto a -slave)

Podemos ver que el cluster se comporta como deseamos. El comportamiento es tan poco sorprendente que he sido bastante escueto con las descripciones, porque básicamente hemos hecho lo mismo cuatro veces. En algunas partes no me requería que ejecutase algunos comandos, como:

```
sudo crm resource score | grep allocation
```

```
cat /proc/net/arp
```

pero como es información que tampoco sobra, y que además nos ayuda a comprender qué está pasando, por si no estuviese lo suficientemente claro, opté por ponerlo todo junto.

JUSTIFICACIÓN DE LA TERCERA PARTE

Los comandos ejecutados han sido los siguientes (tras haber ejecutado previamente los de la segunda parte):

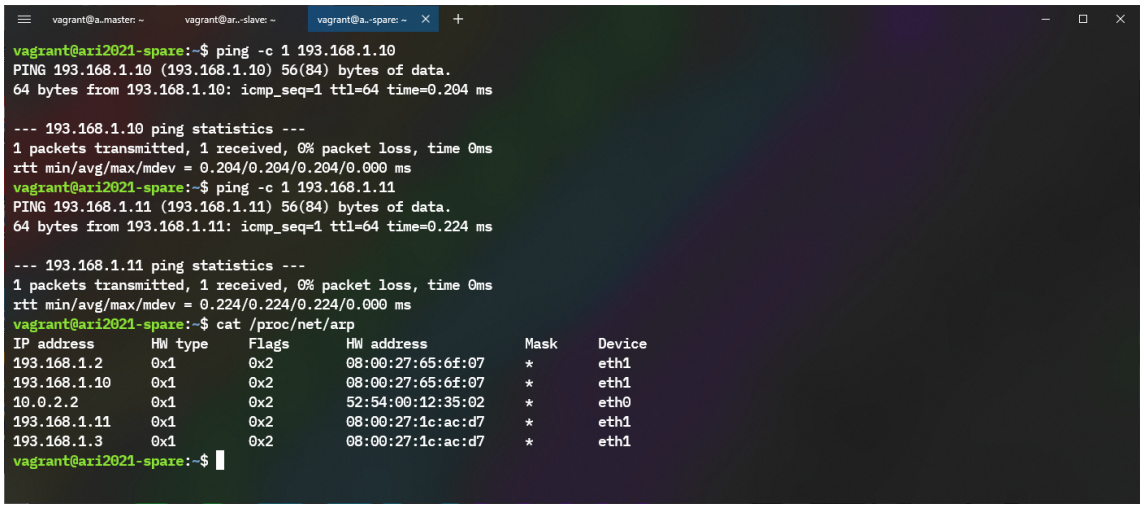
master$ sudo crm configure location prefer-master2spare ari2021-VIP1 40: ari2021-spare
master$ sudo crm configure location prefer-slave2spare ari2021-VIP2 40: ari2021-spare
master$ sudo crm configure rsc_defaults resource-stickiness=50

Se adjuntan capturas de la ejecución (en orden) de los siguientes comandos:

sudo crm configure show &&
sudo crm status &&
sudo crm resource score | grep allocation

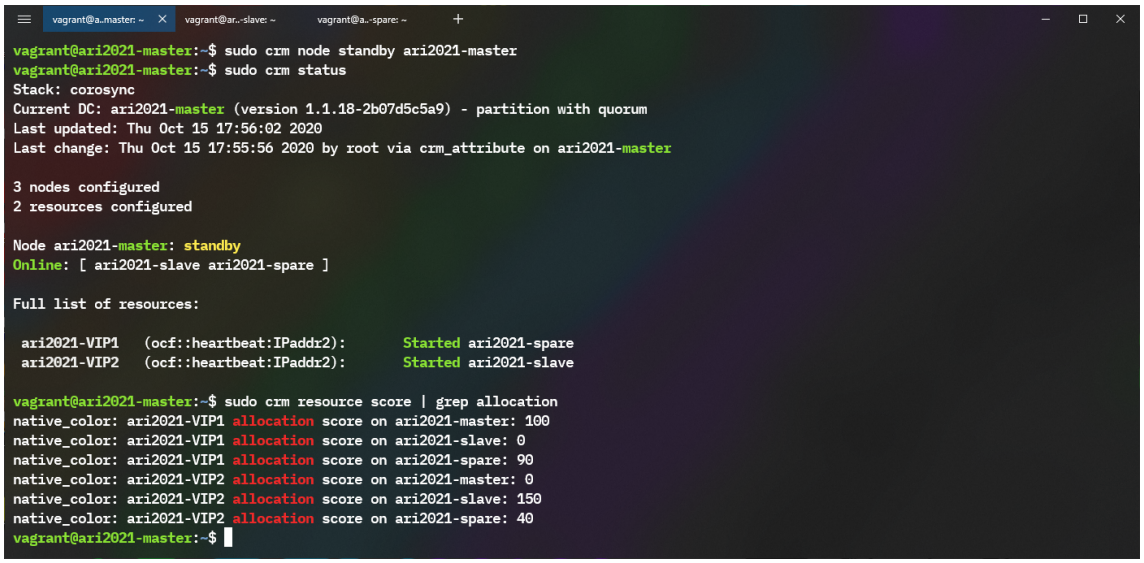
(Simplemente miramos la configuración)

ping [...] && cat /proc/net/arp



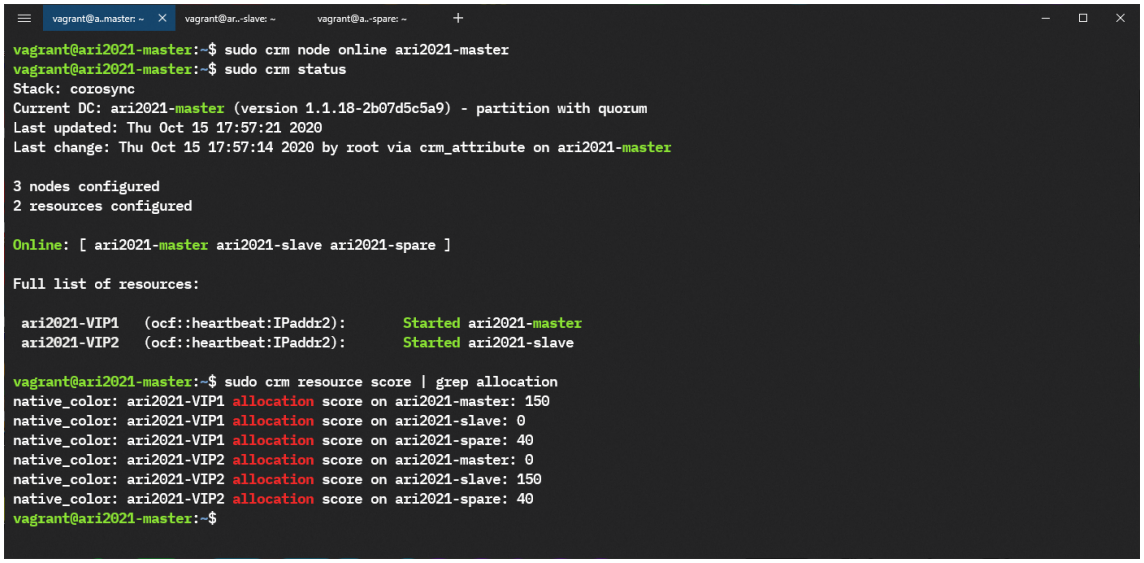
(Vemos que cada VIP está en su nodo favorito)

sudo crm node standby ari2021-master
sudo crm status
sudo crm resource score | grep allocation



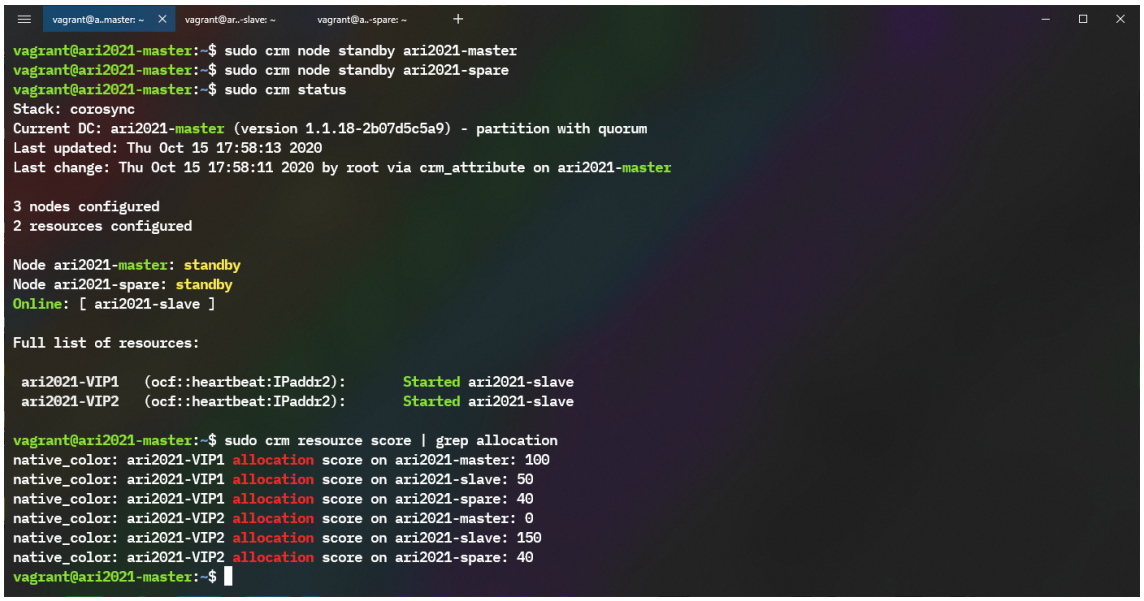
(Al apagar el master, se va al spare)

sudo crm node online ari2021-master
sudo crm status
sudo crm resource score | grep allocation



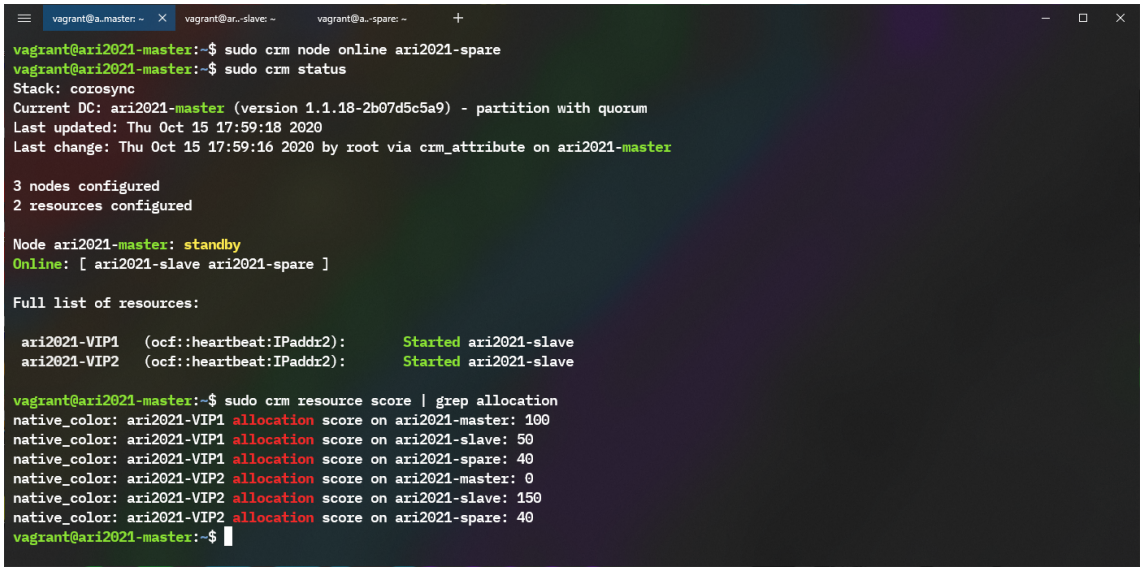
(Y al volver a encenderlo, por la stickiness y la preferencia que le hemos puesto, se vuelve al nodo master, como se espera)

sudo crm node standby ari2021-master
sudo crm node standby ari2021-spare
sudo crm status
sudo crm resource score | grep allocation



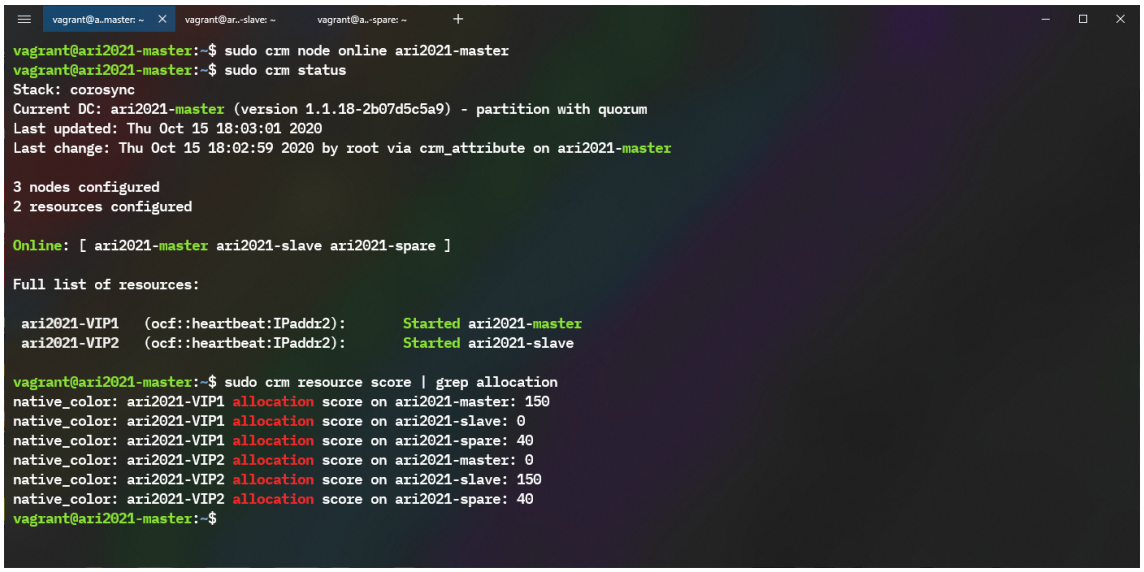
(Al apagar ambos nodos master y spare, la VIP se va al único que queda y de menor preferencia, que es el nodo slave)

sudo crm node online ari2021-spare
sudo crm status
sudo crm resource score | grep allocation



(Pero, dada la preferencia en 40 del nodo spare, se le indica al sistema que no compensa mover el recurso de vuelta al spare, por lo que se mantiene en el slave)

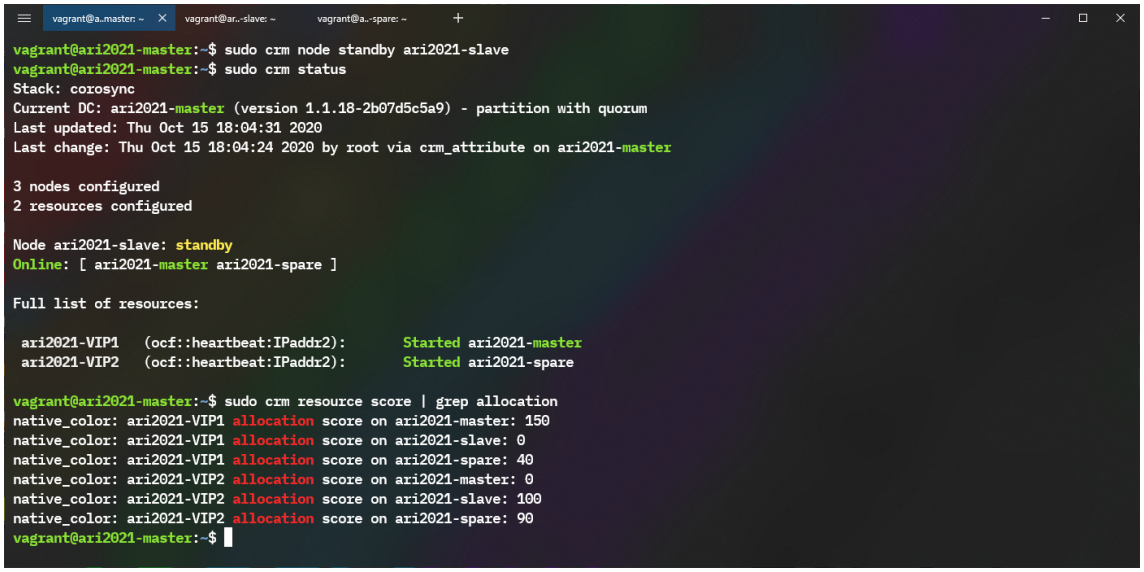
sudo crm node online ari2021-master
sudo crm status
sudo crm resource score | grep allocation



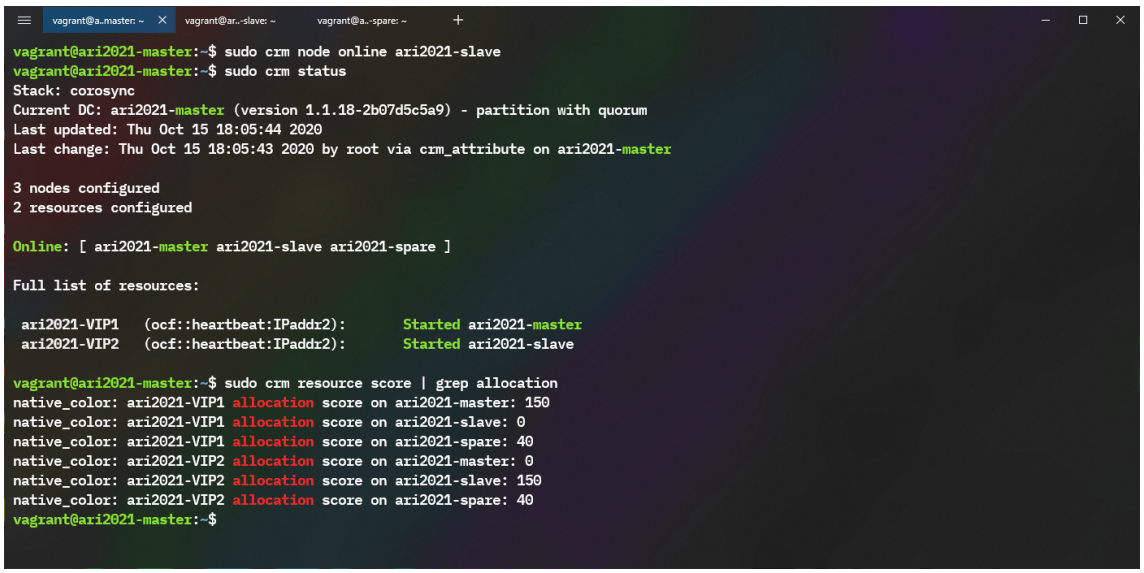
(Ahora que hemos encendido el master, ya sí compensa mover el recurso al nodo master, por lo que se mueve a la misma)

Aquí repetimos los pasos anteriores, pero con el slave, por lo que no se va a indicar nada en el pie de foto.

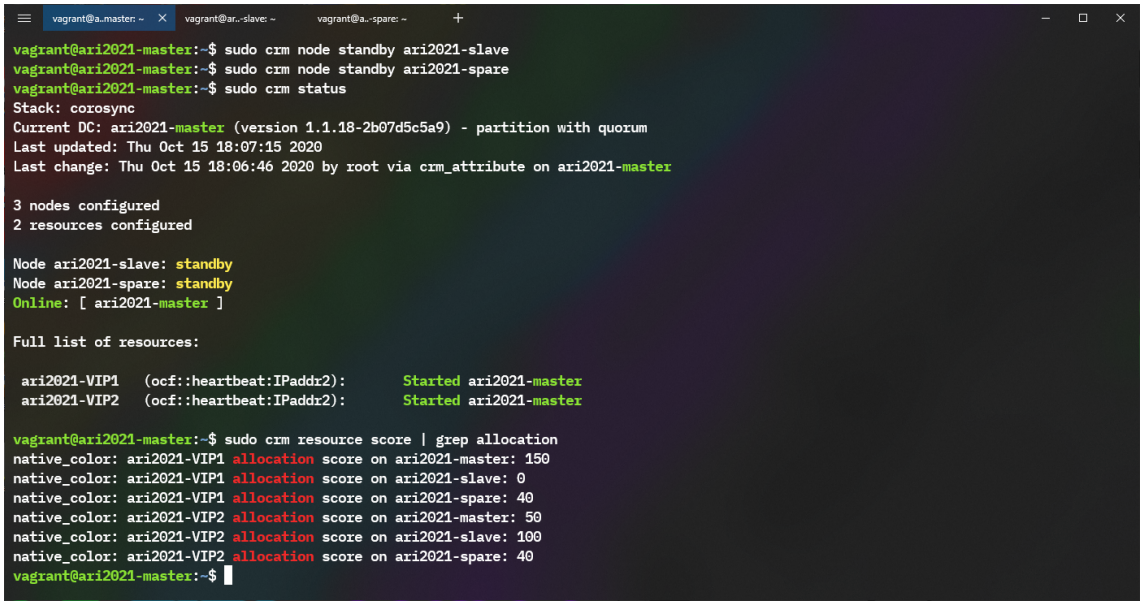
sudo crm node standby ari2021-slave
sudo crm status
sudo crm resource score | grep allocation



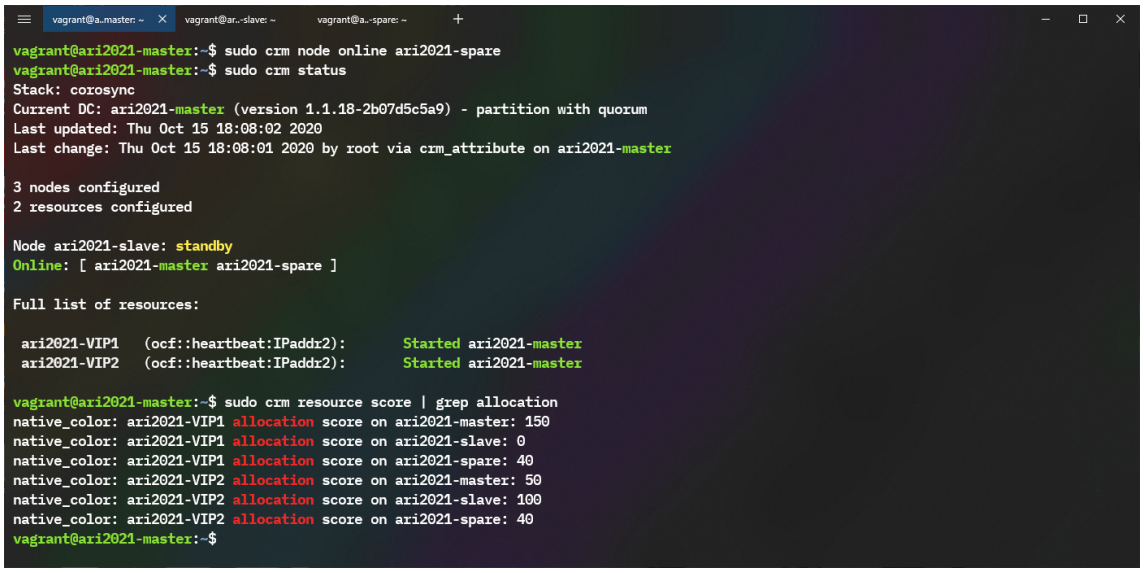
sudo crm node online ari2021-slave
sudo crm status
sudo crm resource score | grep allocation



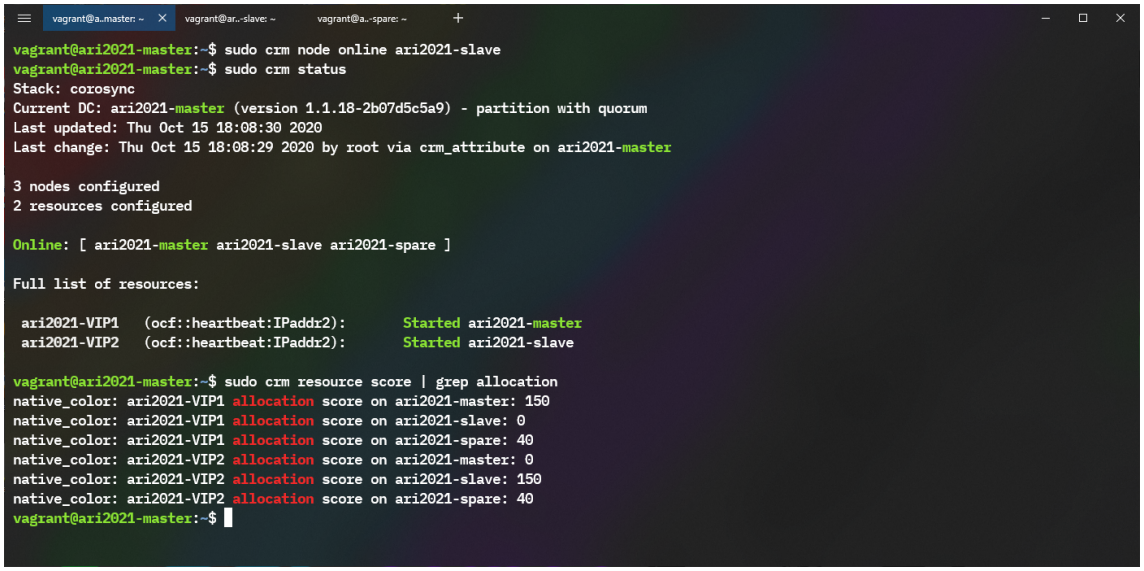
```
sudo crm node standby ari2021-slave
sudo crm node standby ari2021-spare
sudo crm status
sudo crm resource score | grep allocation
```




```
sudo crm node online ari2021-spare
sudo crm status
sudo crm resource score | grep allocation
```



sudo crm node online ari2021-slave
sudo crm status
sudo crm resource score | grep allocation



Podemos ver que el cluster (de nuevo) se comporta como deseamos.

Este sistema que acabamos de implementar es más útil cuanto más tiempo tarde la migración del sistema, ya que evitamos migraciones innecesarias entre el tercer y segundo nodo por orden de preferencia.

Poco que comentar la verdad. El cluster funciona bien, y creo que las salidas de los comandos son bastante autoexplicativas.

La preferencia de **`prefer-master2spare`** y **`prefer-slave2spare`** están puestas a **`40`** para que cuando **`spare`** vuelva a estar disponible, pero el nodo con más preferencia no, no se haga esa migración innecesaria comentada más arriba, ya que, **`40<50`**, y, con esos datos, el sistema estima acertadamente, que no es conveniente mover el sistema.

Alonso Rodriguez Iglesias. 15-Octubre-2020