



Asignatura:

Sistemas Operativos III

Tema:

HOWTO Y VIDEO (CLUSTER DE EQUIPOS LINUX)

Participante:

Santiago M. Duval Contreras

Matricula:

2015-3246

Facilitador:

José Doñe

Fecha:

5/4/2020

HOW-TO? | Creando un Clúster de Servidores en ClearOS

HOW-TO? | Creando un Clúster de Servidores en ClearOS.

En este documento veremos los pasos requerido para crear un clúster de servidores en ClearOS en Oracle VirtualBox.

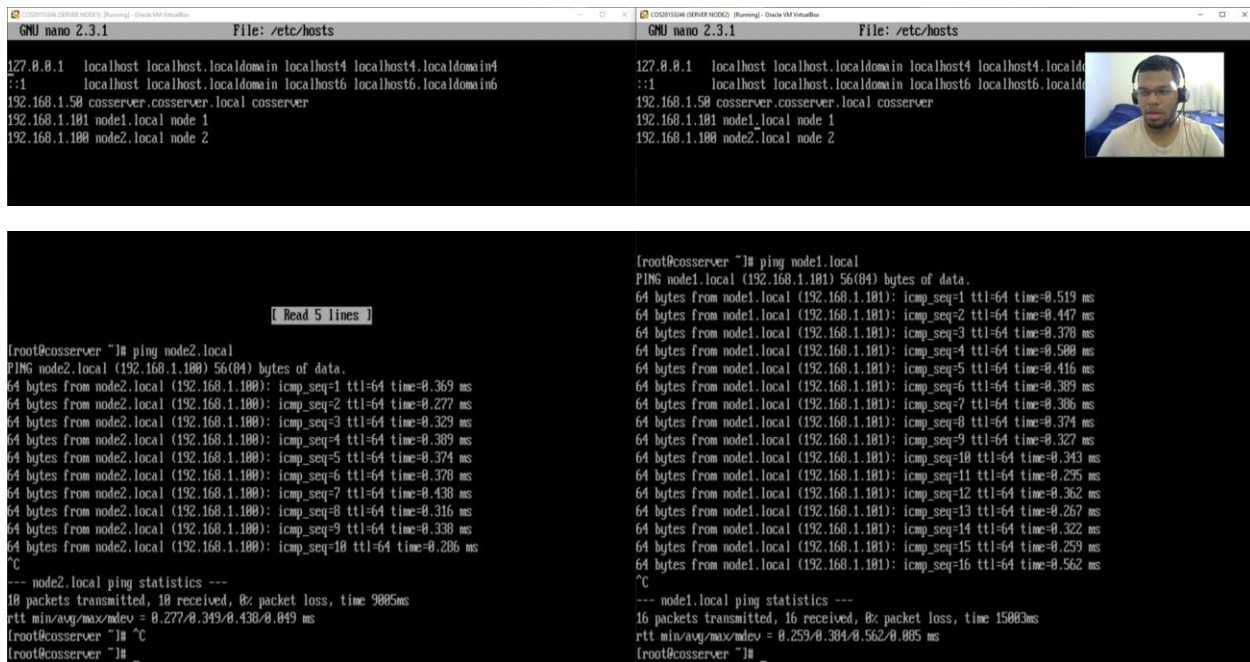
Link a demostración audiovisual: <https://youtu.be/VBBwfWaxHa4>

Requerimientos del OS:

Tener instalado el programa VirtualBox y tener el OS de ClearOS instalado en 2 máquinas virtuales con los requisitos de hardware virtuales que sean requeridos. Selinux tiene que estar desactivado. Ambas maquinas tienen que estar dentro de una red interna.

Paso 1 – Editar el archivo hosts.

Utilizando el comando **nano /etc/hosts** y añadiremos la ip de ambos servidores junto con el nombre **node1.local node2.local** para que ambos servidores ser puedan reconocer dentro de la red interna utilizando estos nombres.



```
GNU nano 2.3.1 File: /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.1.58 cosserv.cosserv.local cosserv
192.168.1.101 node1.local node 1
192.168.1.100 node2.local node 2

[ Read 5 lines ]

[root@cosserv ~]# ping node2.local
PING node2.local (192.168.1.100) 56(84) bytes of data:
64 bytes from node2.local (192.168.1.100): icmp_seq=1 ttl=64 time=0.369 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=2 ttl=64 time=0.277 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=3 ttl=64 time=0.329 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=4 ttl=64 time=0.389 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=5 ttl=64 time=0.374 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=6 ttl=64 time=0.378 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=7 ttl=64 time=0.438 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=8 ttl=64 time=0.316 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=9 ttl=64 time=0.338 ms
64 bytes from node2.local (192.168.1.100): icmp_seq=10 ttl=64 time=0.286 ms
^C
--- node2.local ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9005ms
rtt min/avg/max/mdev = 0.277/0.349/0.438/0.049 ms
[root@cosserv ~]# ^C
[root@cosserv ~]#

[root@cosserv ~]# ping node1.local
PING node1.local (192.168.1.101) 56(84) bytes of data:
64 bytes from node1.local (192.168.1.101): icmp_seq=1 ttl=64 time=0.519 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=2 ttl=64 time=0.447 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=3 ttl=64 time=0.378 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=4 ttl=64 time=0.588 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=5 ttl=64 time=0.416 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=6 ttl=64 time=0.389 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=7 ttl=64 time=0.386 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=8 ttl=64 time=0.374 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=9 ttl=64 time=0.327 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=10 ttl=64 time=0.343 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=11 ttl=64 time=0.295 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=12 ttl=64 time=0.362 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=13 ttl=64 time=0.267 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=14 ttl=64 time=0.322 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=15 ttl=64 time=0.259 ms
64 bytes from node1.local (192.168.1.101): icmp_seq=16 ttl=64 time=0.562 ms
^C
--- node1.local ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15803ms
rtt min/avg/max/mdev = 0.259/0.384/0.562/0.085 ms
[root@cosserv ~]#
```

Paso 2 – Descargar el Servicio de Clúster.

Utilizando el comando **yum install pacemaker**, descargaremos e instalaremos el servicio de clúster en el servidor.



```
[root@cosserv ~]# yum install pacemaker pcs
Loaded plugins: clearcenter-marketplace, fastestmirror
ClearCenter Marketplace: fetching repositories...
Loading mirror speeds from cached hostfile
* clearos: mirror1-newyork.clearos.com
* clearos-centos: download3.clearsdn.com
* clearos-centos-scllo-rh: download3.clearsdn.com
* clearos-centos-updates: download3.clearsdn.com
* clearos-centos-verified: mirror1-newyork.clearos.com
* clearos-contribs: mirror1-newyork.clearos.com
* clearos-epel-verified: mirror1-newyork.clearos.com
* clearos-fast-updates: download3.clearsdn.com

[root@cosserv ~]# yum install
```

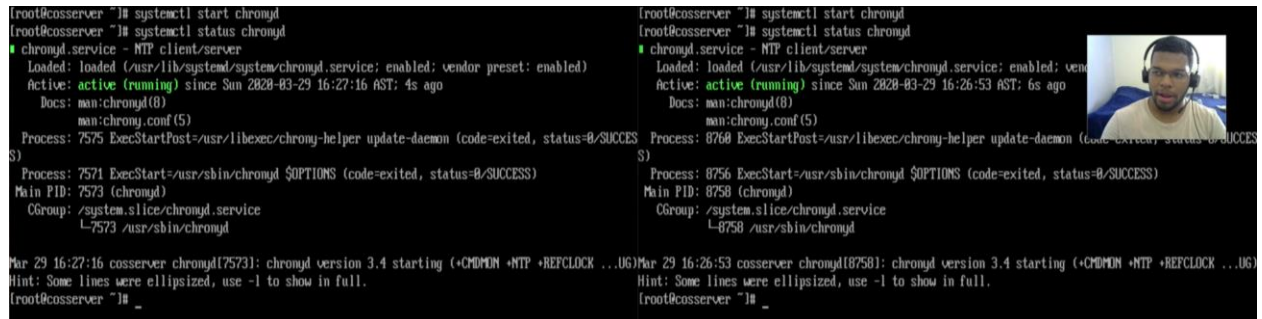
HOW-TO? | Creando un Clúster de Servidores en ClearOS

Paso 3 – Habilitar el servicio chronyd.

Utilizando el comando **systemctl start chronyd** y **systemctl enable chronyd** habilitaremos el servicio de sincronización horaria.

```
[root@cosserv ~]# systemctl start chronyd
[root@cosserv ~]# systemctl status chronyd
chronyd.service - NTP client/server
Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; vendor preset: enabled)
Active: active (running) since Sun 2020-03-29 16:27:16 AST; 4s ago
Docs: man:chronyd(8)
      man:chrony.conf(5)
Process: 7575 ExecStartPost=/usr/libexec/chrony-helper update-daemon (code=exited, status=0/SUCCESS)
Process: 7571 ExecStart=/usr/sbin/chronyd $OPTIONS (code=exited, status=0/SUCCESS)
Main PID: 7573 (chronyd)
CGroup: /system.slice/chronyd.service
        └─7573 /usr/sbin/chronyd

Mar 29 16:27:16 cosserv chronyd[7573]: chronyd version 3.4 starting (+CHROND +NTP +REFCLOCK ...UG)
Hint: Some lines were ellipsized, use -l to show in full.
[root@cosserv ~]#
```

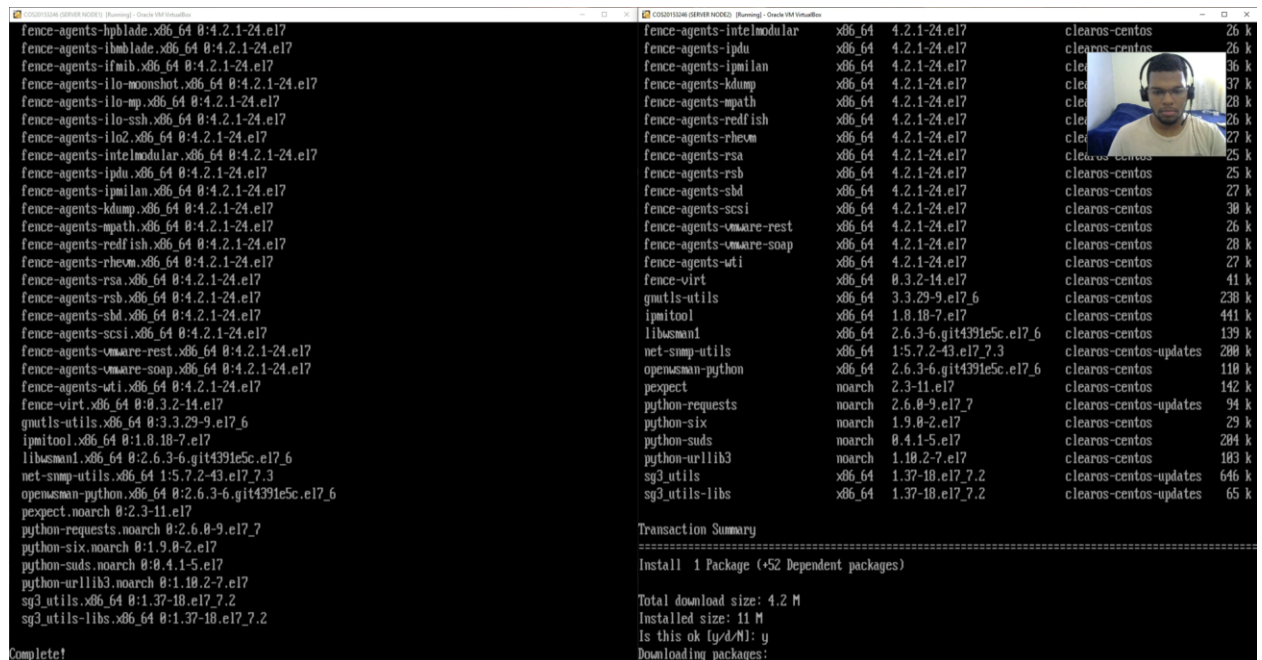


Paso 4 – Descargar fence-agents.

Utilizando el comando **yum install fence-agents-all**, descargaremos e instalaremos el servicio de usuarios de clúster en el servidor.

```
fence-agents-ibmblade.x86_64 0:4.2.1-24.el7
fence-agents-ibmblade.x86_64 0:4.2.1-24.el7
fence-agents-ibmb.x86_64 0:4.2.1-24.el7
fence-agents-ilo-moonshot.x86_64 0:4.2.1-24.el7
fence-agents-ilo-mp.x86_64 0:4.2.1-24.el7
fence-agents-ilo-ssh.x86_64 0:4.2.1-24.el7
fence-agents-ilo2.x86_64 0:4.2.1-24.el7
fence-agents-intelmodular.x86_64 0:4.2.1-24.el7
fence-agents-ipdu.x86_64 0:4.2.1-24.el7
fence-agents-ipmilan.x86_64 0:4.2.1-24.el7
fence-agents-kdump.x86_64 0:4.2.1-24.el7
fence-agents-mpath.x86_64 0:4.2.1-24.el7
fence-agents-redfish.x86_64 0:4.2.1-24.el7
fence-agents-rhevm.x86_64 0:4.2.1-24.el7
fence-agents-rsa.x86_64 0:4.2.1-24.el7
fence-agents-rsb.x86_64 0:4.2.1-24.el7
fence-agents-sbd.x86_64 0:4.2.1-24.el7
fence-agents-ssci.x86_64 0:4.2.1-24.el7
fence-agents-vmware-rest.x86_64 0:4.2.1-24.el7
fence-agents-vmware-soap.x86_64 0:4.2.1-24.el7
fence-agents-uti.x86_64 0:4.2.1-24.el7
fence-virt.x86_64 0:8.3.2-14.el7
gnutls-utils.x86_64 0:3.3.29-9.el7_6
ipmitool.x86_64 0:1.8.10-7.el7
libman1.x86_64 0:2.6.3-6.git4391e5c.el7_6
net-snmp-utils.x86_64 1:5.7.2-43.el7_7.3
openman-python.x86_64 0:2.6.3-6.git4391e5c.el7_6
pexpect.noarch 0:2.3-11.el7
python-requests.noarch 0:2.6.0-9.el7_7
python-six.noarch 0:1.9.0-2.el7
python-suds.noarch 0:0.4.1-5.el7
python-urllib3.noarch 0:1.10.2-7.el7
sq3_utils.x86_64 0:1.37-10.el7_7.2
sq3_utils-libs.x86_64 0:1.37-10.el7_7.2

Complete!
```



Paso 5 – Cambiar el password del usuario hacluster.

Utilizamos el comando **passwd** para cambiar el password del usuario hacluster que se creo al descargar el servicio de fence-agents.

Paso 6 – Habilitar high-availability.

Para que el clúster funcione correctamente tenemos que habilitar high-availability en el firewall para hacer esto utilizamos el siguiente comando: **firewall-cmd --permanent --add-service: high-availability**.

HOW-TO? | Creando un Clúster de Servidores en ClearOS

Paso 7 – Iniciar el servicio de clúster.

Utilizando el comando **systemctl start pcsd** y **systemctl enable pcsd** iniciamos y habilitamos el servicio para que inicie onboot.

```
[root@cosserv ~]# firewall-cmd --permanent --add-service=high-availability
success
[root@cosserv ~]# systemctl start pcsd
[root@cosserv ~]# systemctl status pcsd
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; disabled; vendor preset: disabled)
Active: active (running) since Sun 2020-03-29 16:44:11 AST; 21s ago
Docs: man:pcsd(8)
      man:pcs(8)
Main PID: 8322 (pcsd)
CGroup: /system.slice/pcsd.service
        └─8322 /usr/bin/ruby /usr/lib/pcs/pcsd
[root@cosserv ~]# systemctl enable pcsd
Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to /usr/lib/systemd/system/pcsd.service.
[root@cosserv ~]#
```

```
[root@cosserv ~]# systemctl status pcsd
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; disabled; vendor preset: disabled)
Active: active (running) since Sun 2020-03-29 16:44:21 AST; 5s ago
Docs: man:pcsd(8)
      man:pcs(8)
Main PID: 9576 (pcsd)
CGroup: /system.slice/pcsd.service
        └─9576 /usr/bin/ruby /usr/lib/pcs/pcsd
[root@cosserv ~]# systemctl status pcsd
```

Paso 8 – En ambos servidores autorizamos el servicio de clúster.

Esto lo hacemos con el comando **pcs cluster auth node1.local node2.local**, después de colocar este comando nos va a pedir que ingresemos el nombre de nuestro usuario de cluster, en este caso hacluster y el password del mismo.

```
[root@cosserv ~]# pcs cluster auth node1.local node2.local
Username: ha_cluster
Password:
Error: node1.local: Username and/or password is incorrect
Error: node2.local: Username and/or password is incorrect
[root@cosserv ~]# pcs cluster auth node1.local node2.local
Username: hacluster
Password:
node1.local: Authorized
node2.local: Authorized
[root@cosserv ~]#
```

```
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]# pcs cluster auth node1.local node2.local
node1.local: Already authorized
node2.local: Already authorized
[root@cosserv ~]#
```

Paso 9 – Iniciamos el cluster únicamente en el servidor nodo1.

Esto lo hacemos con el comando **pcs cluster setup --start --name hacluster node1.local node2.local**, esto solo se realizará en el servidor que fungirá la función master.

```
[root@cosserv ~]# pcs cluster setup --start --name hacluster node1.local node2.local
Destroying cluster on nodes: node1.local, node2.local...
node1.local: Stopping Cluster (pacemaker)...
node2.local: Stopping Cluster (pacemaker)...
node1.local: Successfully destroyed cluster
node2.local: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to 'node1.local', 'node2.local'
node1.local: successful distribution of the file 'pacemaker_remote authkey'
node2.local: successful distribution of the file 'pacemaker_remote authkey'
Sending cluster config files to the nodes...
node1.local: Succeeded
node2.local: Succeeded

Starting cluster on nodes: node1.local, node2.local...
node1.local: Starting Cluster (corosync)...
node2.local: Starting Cluster (corosync)...
node1.local: Starting Cluster (pacemaker)...
node2.local: Starting Cluster (pacemaker)...

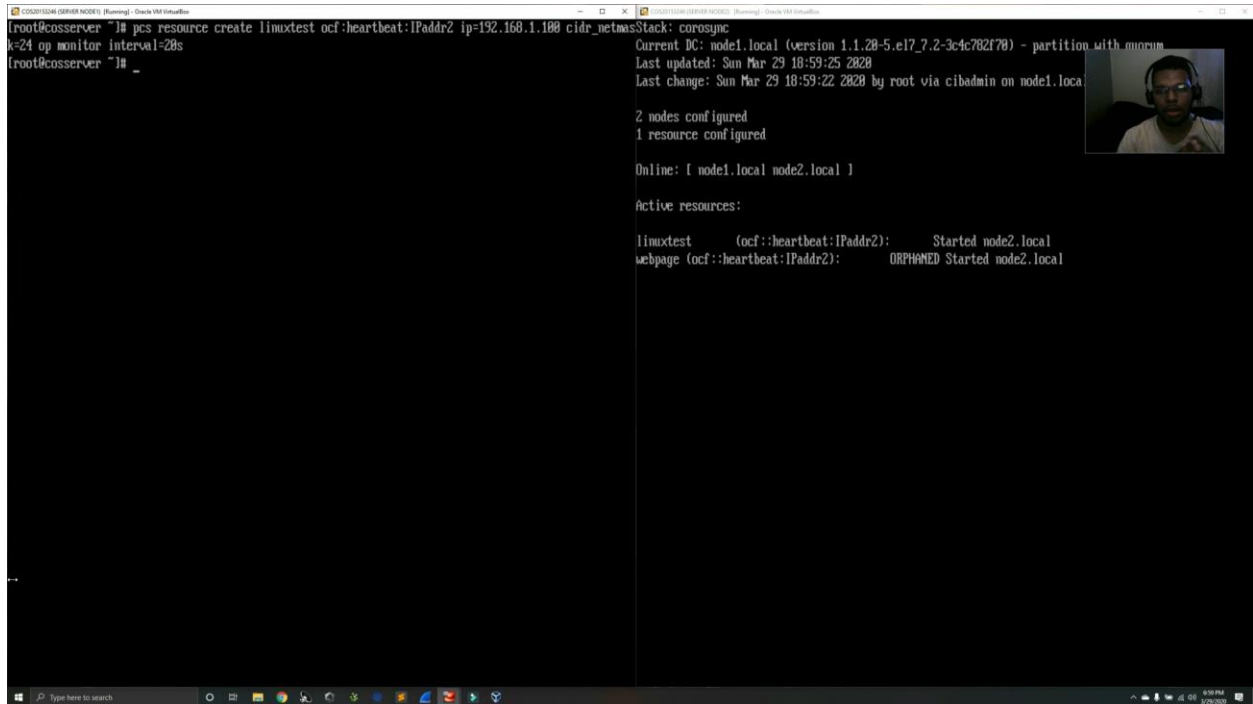
Synchronizing pcsd certificates on nodes node1.local, node2.local...
node1.local: Success
node2.local: Success
Restarting pcsd on the nodes in order to reload the certificates...
node1.local: Success
node2.local: Success

[root@cosserv ~]# pcs cluster start --all
node1.local: Starting Cluster (corosync)...
node2.local: Starting Cluster (corosync)...
node1.local: Starting Cluster (pacemaker)...
node2.local: Starting Cluster (pacemaker)...
[root@cosserv ~]#
```

```
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]#
[root@cosserv ~]# pcs cluster auth node1.local node2.local
node1.local: Already authorized
node2.local: Already authorized
[root@cosserv ~]#
```

HOW-TO? | Creando un Clúster de Servidores en ClearOS

Paso 9 – Creamos un proceso a compartir.



```
[root@cosserver ~]# pcs resource create linuxtest ocf::heartbeat:IPaddr2 ip=192.168.1.100 cidr_netmaskStack: corosync
k=24 op monitor interval=28s
[root@cosserver ~]#
```

Current DC: node1.local (version 1.1.28-5.el7_7.2-3c4c782f70) - partition with muncm
Last updated: Sun Mar 29 18:59:25 2020
Last change: Sun Mar 29 18:59:22 2020 by root via cibadmin on node1.local

2 nodes configured
1 resource configured

Online: [node1.local node2.local]

Active resources:

linuxtest (ocf::heartbeat:IPaddr2): Started node2.local
webpage (ocf::heartbeat:IPaddr2): ORPHANED Started node2.local