

Demo contenedores

HPC Admintech

Valencia, Mayo 2018

Jesús Cuenca
Senior HPC Consultant @ SIE

Maquina virtual demo

Maquina virtual demo

Descargar MV:

<http://IP/d>

Demodocker .ova

(Instalar Virtualbox)

Importar servicio virtualizado → demodocker.ova

(ajustar RAM MV)

(Crear red de anfitrión, conectar MV a red)

Maquina virtual demo

Usuario: demo

Pass: demodocker

LXC

LXC

Lxc commands:

exec, info, launch, list, start, stop, restart...

```
lxc image list
```

```
lxc launch <image> <name> [--profile|-p <profile>...]
```

```
lxc launch ubuntu:16.04 test -p default -p docker
```

```
lxc list
```

```
lxc exec test bash  
cat /etc/os-release
```

```
lxc stop && lxc delete
```

MCC

mcc

My Container Cluster (MCC)

“a tool that automates the creation of LXC container based computing clusters”

<https://github.com/grycap/mcc>

mcc

```
mcc create --help
```

```
LXC_PROFILE="HPCmD" mcc --verbose create --front-end-image  
local:HPCmD:0.1 --context-folder ./HPCmeetsDocker/ -n 1 -e -d home
```

```
mcc list
```

```
mcc delete
```

Actualizar configuración slurm:

```
/usr/local/etc/slurm.conf
```

Nodo master:

```
hostname  
ip addr
```

Obtener configuración nodos:

```
ssh node1  
slurmd -C
```

```
ssh node1  
slurmd -C
```

mcc/slurm

Ejemplo slurm.conf:

```
ControlMachine=mcc-0b7a3b05ed43  
ControlAddr=10.116.30.88  
(...)
```

```
NodeName=mcc-0b7a3b05ed43-node1 CPUs=1 SocketsPerBoard=1 CoresPerSocket=1 ThreadsPerCore=1  
RealMemory=992 TmpDisk=7194  
NodeName=mcc-0b7a3b05ed43-node2 CPUs=1 SocketsPerBoard=1 CoresPerSocket=1 ThreadsPerCore=1  
RealMemory=992 TmpDisk=7194
```

```
PartitionName=mcc Nodes=mcc-0b7a3b05ed43-node1,mcc-0b7a3b05ed43-node2 Default=YES  
MaxTime=INFINITE State=UP
```

Iniciar servicio en nodo principal

```
slurmctld
```

Replicar slurm.conf para nodos de cómputo

```
cp /usr/local/etc/slurm.conf /home/slurm/
```

Iniciar servicio en nodos cómputo

```
ssh node1
```

```
cp /home/slurm/slurm.conf /usr/local/etc/  
slurmd
```

```
ssh node2
```

```
cp /home/slurm/slurm.conf /usr/local/etc/  
slurmd
```

mcc/usando el cluster

MPI

```
mpicc mpi-hello.c  
srun mpirun -np 4 /home/ubuntu/mpi-hello
```

Docker

Docker: imágenes

Docker Hub:

<https://hub.docker.com/explore/>

Ubuntu:

https://hub.docker.com/_/ubuntu/

Tags: latest, 18.04, 16.04, ...

```
sudo docker pull ubuntu:16.04
```

Docker: usuarios

Por defecto, creación de contenedores deshabilitada

```
/etc/group: grupo docker
```

El proceso se ejecuta como root...

Docker: comandos

```
sudo docker images  
sudo docker run -it centos bash
```

```
sudo docker ps
```

```
sudo docker commit
```

```
sudo docker rm
```

mcc/usando el cluster

Contenedor a medida

```
ssh node1  
docker pull ubuntu:16.04
```

```
docker run -it ubuntu:16.04 bash  
apt-get update  
apt-get install python2.7
```

```
docker tag docker image tag IMG_HASH python-2.7  
docker images
```

Docker: ejecución

```
hostname.py:  
#!/usr/bin/env python  
import socket  
print "Running in...", socket.gethostname()
```

```
sudo docker run -v /home/ubuntu:/home/ubuntu/:rw python-2.7  
python2.7 /home/ubuntu/hostname.py
```

mcc/usando el cluster

Contenedor a medida

```
srun -N 1 docker run -v /home/ubuntu:/home/ubuntu:rw python-2.7  
python2.7 /home/ubuntu/hostname.py
```

Dockerfile

```
cd examples/minicom/
```

```
FROM ubuntu:latest
```

```
RUN apt-get update && apt-get install -y ssh iproute2 iputils-ping wget
```

```
docker build . -t etiqueta
```

dosh

dosh

Docker Shell

Use Docker containers to run as the shell of the users in your Linux system.

When a user logs-in a Linux system, a customized (or standard) container will be created for him.

Limit the resources that the user is able to use, the applications, etc.
Provide custom linux flavour for each user or group of users

Container is started as sudo, but processes **run as user**

minicon

minicon

General tool to analyze applications and their executions to obtain a filesystem that contains all the dependencies that have been detected.

In particular, it can be used to reduce Docker containers.

```
docker build . -t minicon:example1
```

```
docker images
```

```
minidock -i minicon:example1 -t minicon:mini1 -apt  
-E bash -E 'ssh localhost' -E ip -E id -E cat -E ls -E mkdir  
-E 'ping -c 1 www.google.es' -- wget www.google.es
```

```
docker images
```

singularity

singularity

<https://singularity-hub.org/>

<https://singularity-hub.org/collections/738/usage>



singularity/imágenes

```
sudo singularity build --writable centos.img /usr/share/doc/singularity-  
container/examples/centos/Singularity
```

```
singularity exec centos.img bash
```

```
sudo singularity shell centos.img  
which  
yum install which
```

```
singularity shell --writable centos.img
```

```
singularity pull --name centos.simg shub://singularityhub/centos
```

singularity/ímaĝenes

```
rm -rf centos/  
sudo singularity build --sandbox centos/ docker://centos
```

```
sudo singularity shell --writable centos  
yum install openmpi
```

```
mpirun -np 2 singularity exec centos /home/demo/examples/openmpi-hello
```

singularity/ejecución

Montaje por defecto: /home/\$USER, /tmp, \$PWD

```
singularity shell --bind /tmp:/tmp centos.img
```

```
singularity instance.start centos.img centos1
```

```
singularity instance.list
```

```
singularity shell instance://centos1
```

```
singularity instance.stop centos1
```

Gracias

Contacto



Sistemas Informáticos Europeos

Calle Marqués de Mondejar nº 29

913 61 10 02

www.sie.es



[/HPCSIE](https://www.facebook.com/HPCSIE)



soporte@sie.es



[@HPCSIE](https://twitter.com/HPCSIE)



[+SistemasInformaticosEuropeosSLMadrid](https://plus.google.com/+SistemasInformaticosEuropeosSLMadrid)

