

## Práctica 2. Despliegue aplicaciones con JUJU / LXD

### Objetivo:

Familiarizarse con el entorno JUJU y la creación de contenedores basados en LXD.

Para los apartados IV, V, VI, VII deben presentarse en un documento pdf, con la siguiente información:

- url de la dirección \*.onion donde está activa la página wiki
- wiki modificado donde aparece el nombre del alumno
- captura de la página wiki desde el TBB (Tor Browser Bundle)
- captura del acceso por ssh al servidor utilizando la red tor
- captura de gitlab desde el TBB (Tor Browser Bundle)
- captura del mysql con replicación

El pdf se subirá al SWAD, en el área de mis trabajos. Fecha tope de entrega: **17-octubre-2018**

### Desarrollo:

JUJU permite crear fácilmente contenedores con aplicaciones preinstaladas y una configuración básica para evaluarlas y administrarlas.

### I) Configuramos LXD (este paso no es necesario en el contenedor remoto de abfs, ya que está preinstalado)

*Asegurarse que tenemos instalado lxd y zfsutils-linux*

```
lxd init
```

Durante la inicialización:

```
Name of the storage backend to use (dir or zfs) [default=zfs]: dir
Would you like to have your containers share their parent's allocation (yes/no) [default=yes]?
Would you like LXD to be available over the network (yes/no) [default=no]?
Do you want to configure the LXD bridge (yes/no) [default=yes]?

Would you like to setup a network bridge for LXD containers now? <Yes>
Bridge interface name: lxdbr0
Do you want to setup an IPv4 subnet? <Yes>
OK confirmando la configuración propuesta.
Do you want to NAT the IPv4 traffic? <Yes>

Do you want to setup an IPv6 subnet? <No>
```

No es necesario configurar una subred IPv6.

*Desactivamos IPv6 en lxdbr*

```
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
```

Recordamos

```
Creación de un contenedor: lxc launch ubuntu:xenial c1
Mostrar contenedores: lxc list
Mostrar imágenes: lxc image list
Ejecutar en el contenedor (c1): lxc exec c1 bash
```

Si queremos borrar por completo el almacenamiento de lxd:

```
sudo service lxd stop
rm -rf var/lib/lxd
sudo service lxd start
```

## II) Instalación de JUJU (este paso no es necesario en el contenedor remoto de abfs, ya que está preinstalado)

Para instalarlo en nuestro equipo con snap

```
sudo snap install juju
```

La documentación de Juju está disponible en : <https://docs.jujucharms.com/2.4/en/>

## III) Creación del controlador

Una vez instalado definimos un controlador. Los controladores establecen los recursos que se van a utilizar para desplegar los contenedores.

En nuestro caso estamos utilizando contenedores locales basados en LXD, por lo que podemos crear nuestro contenedor con:

```
juju bootstrap localhost lxd-test
```

Se crea un contenedor básico con los recursos y servicios básicos para que Juju pueda instalar aplicaciones. Este proceso puede durar 1-2 minutos.

Para mostrar la lista de controladores :

```
juju controllers
```

## III) Despliegue de aplicaciones

Una vez disponemos de un controlador podemos crear contenedores con aplicaciones. Veamos un ejemplo:

```
juju deploy cs:bundle/wiki-simple
```

Comprobamos su funcionamiento con :

```
juju status
```

Y comprobamos en qué máquina se está ejecutando wiki:

```
Model  Controller Cloud/Region  Version  SLA
default lxd-test  localhost/localhost 2.2.4  unsupported

App  Version  Status  Scale  Charm   Store   Rev  OS   Notes
mysql      unknown   1  mysql  jujucharms  55  ubuntu
wiki       unknown   1  mediawiki jujucharms  5  ubuntu

Unit    Workload Agent Machine Public address Ports  Message
mysql/0* unknown idle 0    10.95.56.64 3306/tcp
wiki/0* unknown idle 1    10.95.56.170 80/tcp

Machine State  DNS      Inst id    Series AZ  Message
0    started 10.95.56.64 juju-cdb1f3-0 trusty Running
1    started 10.95.56.170 juju-cdb1f3-1 trusty Running

Relation provider Requirer  Interface Type
mysql:cluster    mysql:cluster mysql-ha peer
mysql:db         wiki:db      mysql  regular
```

Abrimos una conexión desde nuestro ordenador con un túnel SSH.  
Editamos el fichero *config* del directorio *~/.ssh*

*.ssh/config* (ej. para el usuario **7**)

```
Host abfs
  hostname abfs.ugr.es
  Port 3307
  User cpd7
  LocalForward 8080 10.95.56.170:80
```

Indicamos la IP del servidor wiki

Esto abre un túnel desde el puerto 8080 local al puerto 80 de la máquina que ejecuta el wiki.

Podemos abrir un navegador para acceder a la configuración

```
http://localhost:8080
```

La lista de charms y bundles disponibles está en: <https://jujucharms.com/store>

Si queremos eliminar las aplicaciones de juju:

```
juju remove-application wiki --destroy-storage
juju remove-application mysql --destroy-storage
```

Tras unos segundos desaparecen los recursos utilizados por las aplicaciones (comprobar con *juju status*).

#### IV) Utilizar la red TOR para acceder directamente al contenedor

La red TOR permite crear servicios ocultos y acceder de forma transparente a dicho servidor, ya sean nodos, máquinas virtuales o contenedores.

Configurando el servidor

instalamos tor en el servidor:

```
apt install tor
```

Editamos /etc/tor/torrc

```
RunAsDaemon 1
HiddenServiceDir /root/servicio1
HiddenServicePort 22 127.0.0.1:22
HiddenServicePort 80 10.95.56.170:80
```

Indicamos la IP del servidor wiki

Iniciamos con:

```
tor
```

Esta orden inicia el acceso a la red TOR y permite el acceso a los puertos definidos.

El directorio que utilizamos para definir el servicio debe tener permisos de grupo y otros desactivado:

```
chmod go-rwx /etc/servicio1
```

En el directorio /root/servicio1 se crean dos ficheros:

hostname: Contiene el nombre del servidor para ser utilizado por los clientes: Ej.

i549xykbkbyeupam4.onion

private\_key: Clave privada que permite identificar al servidor

Configurando el cliente

Instalamos el TBB (Tor Browser Bundle)

Iniciamos el TBB

```
./start-tor-browser
```

Se puede acceder directamente utilizando la dirección que aparecen el fichero *hostname* del directorio definido en el servidor.

Para acceder mediante ssh, añadimos las siguientes líneas al fichero .ssh/config que editamos anteriormente en nuestro equipo ( ojo: estas líneas NO se incluyen dentro del contenedor).

```
Host *.onion
  Port 22
  ProxyCommand nc -X 5 -x localhost:9150 %h %p
```

## V) interfaz GUI para Juj

iniciamos el entorno GUI con:

```
juju gui
```

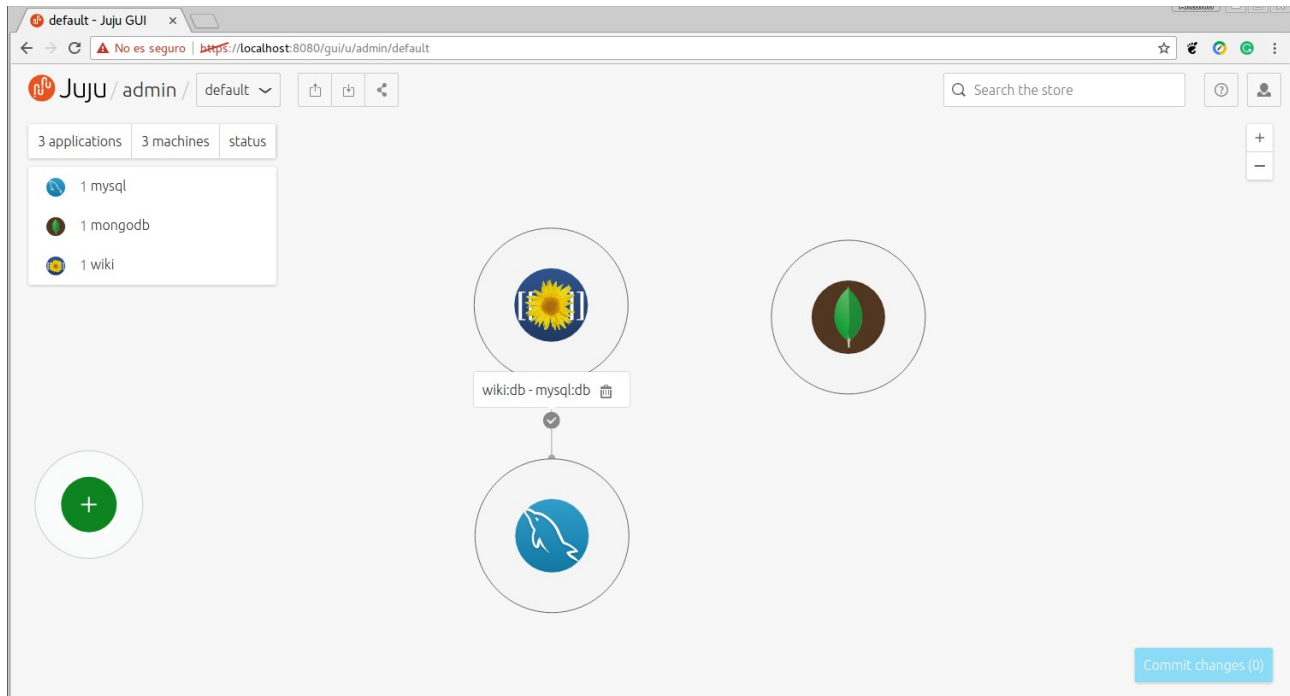
<https://10.95.57.36:15070/gui/u/admin/default>

Your login credential is:

username: admin

password: \*\*\*\*\*

Creamos un túnel ssh para acceder al entorno:



Podemos añadir *mongodb* mediante el entorno gui.

## VI) Instalación de otras aplicaciones:

Procedemos a instalar otras aplicaciones como gitlab:

```
juju deploy cs:~spiculecharms/trusty/gitlab-0  
juju expose gitlab
```

O gitlab server:

```
juju deploy cs:~spiculecharms/gitlab-server-8  
juju expose gitlab
```

## VII) Instalación de Mysql con replicación

MySQL permite replicar bases de datos mediante nuevas instancias. Esto permite, por ejemplo, balancear la carga de consultas entre múltiples instancias o utilizar otra instancia para realizar copias de seguridad, todo ello sin afectar al funcionamiento de la instancia principal.

Seguir las indicaciones de : <https://jujucharms.com/mysql/>