

Pledin

(<http://www.josedomingo.org/pledin/>)

Plataforma Educativa Informática

Creando servidores docker con Docker Machine

Posted on 23 Mayo, 2016 (<http://www.josedomingo.org/pledin/2016/05/creando-servidores-docker-con-docker-machine/>) by admin
(<http://www.josedomingo.org/pledin/author/admin/>)



(<http://www.josedomingo.org/pledin/wp-content/uploads/2016/05/machine.png>)

Docker Machine es una herramienta que nos ayuda a crear, configurar y manejar máquinas (virtuales o físicas) con Docker Engine. Con Docker Machine podemos iniciar, parar o reiniciar los nodos docker, actualizar el cliente o el demonio docker y configurar el cliente docker para acceder a los distintos Docker Engine. El propósito principal del uso de esta herramienta es la de crear máquinas con Docker Engine en sistemas remotos y centralizar su gestión.

Docker Machine utiliza distintos drivers que nos permiten crear y configurar Docker Engine en distintos entornos y proveedores, por ejemplo virtualbox, AWS, VMWare, OpenStack, ...

Las tareas fundamentales que realiza Docker Machine, son las siguientes:

- Crea una máquina en el entorno que hayamos indicado (virtualbox, openstack,...) donde va a instalar y configurar Docker Engine.
- Genera los certificados TLS para la comunicación segura.

También podemos utilizar un driver genérico (generic) que nos permite manejar máquinas que ya están creadas (físicas o virtuales) y configurarlas por SSH.

Instalación de Docker Machine

Para instalar la última versión (0.7.0) de esta herramienta ejecutamos:

```
$ curl -L https://github.com/docker/machine/releases/download/v0.7.0/docker-machine-`uname -s`-`uname -m` > /usr/local/bin/docker-machine && \
chmod +x /usr/local/bin/docker-machine
```

Y comprobamos la instalación:

```
$ docker-machine -version
docker-machine version 0.7.0, build a650a40
```

Utilizando Docker Machine con VirtualBox

Vamos a ver distintos ejemplos de Docker Machine, utilizando distintos drivers. En primer lugar vamos a utilizar el driver de VirtualBox que nos permitirá crear una máquina virtual con Docker Engine en un ordenador donde tengamos instalado VirtualBox. Para ello ejecutamos la siguiente instrucción:

PLEDIN 2.0

```
$ docker-machine create -d virtualbox nodo1
Running pre-create checks...
Creating machine...
(nodo1) Copying /home/jose/.docker/machine/cache/boot2docker.iso to /home/jose/.docker/machine/machines/nodo1/boot2docker.iso...
(nodo1) Creating VirtualBox VM...
(nodo1) Creating SSH key...
(nodo1) Starting the VM...
(nodo1) Check network to re-create if needed...
(nodo1) Found a new host-only adapter: "vboxnet0"
(nodo1) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env nodo1
```

Esta instrucción va a crear una nueva máquina (nodo1) donde se va a instalar una distribución Linux llamada boot2docker con el Docker Engine instalado. Utilizando el driver de VirtualBox podemos indicar las características de la máquina que vamos a crear por medio de parámetros, por ejemplo podemos indicar las características hardware (`--virtualbox-memory`, `--virtualbox-disk-size`, ...) Para más información de los parámetros que podemos usar puedes mirar la documentación del driver (<https://docs.docker.com/machine/drivers/virtualbox/>).

Una vez creada la máquina podemos comprobar que lo tenemos gestionados por Docker Machine, para ello:

```
$ docker-machine ls
NAME      ACTIVE   DRIVER        STATE     URL                  SWARM   DOCKER   ERRORS
nodo1     -        virtualbox    Running   tcp://192.168.99.100:2376          v1.11.1
```

A continuación para conectarnos desde nuestro cliente docker al Docker Engine de la nueva máquina necesitamos declarar las variables de entornos adecuadas, para obtener las variables de entorno de esta máquina podemos ejecutar:

```
$ docker-machine env nodo1
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.100:2376"
export DOCKER_CERT_PATH="/home/jose/.docker/machine/machines/nodo1"
export DOCKER_MACHINE_NAME="nodo1"
```

Y para ejecutar estos comandos y que se creen las variables de entorno, ejecutamos:

```
$ eval $(docker-machine env nodo1)
```

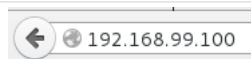
A partir de ahora, y utilizando el cliente docker, estaremos trabajando con el Docker Engine de nodo1:

```
$ docker run -d -p 80:5000 training/webapp python app.py
1450b7b2c785333834b43332a4b86505c0167893306ac511489b0f922560a938
```

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
1450b7b2c785        training/webapp     "python app.py"     8 minutes ago       Up 8 minutes       0.0.0.0:80->5000/tcp   cranky_hopper
```

Y para acceder al contenedor deberíamos utilizar la ip del servidor docker, que la podemos obtener:

```
$ docker-machine ip nodo1
192.168.99.100
```



Hello world!

(<http://www.josedomingo.org/pledin/wp-content/uploads/2016/05/docker1.png>)

Otras opciones de docker-machine que podemos utilizar son:

- `inspect` : Nos devuelve información de una máquina.
- `ssh`, `scp` : Nos permite acceder por ssh y copiar ficheros a una determinada máquina.
- `start`, `stop`, `restart`, `status` : Podemos controlar una máquina.
- `rm` : Es la opción que borra una máquina de la base de datos de Docker Machine. Con determinados drivers también elimina la máquina.
- `upgrade` : Actualiza a la última versión de docker la máquina indicada.

Utilizando Docker Machine con OpenStack

En el ejemplo anterior hemos utilizado el driver VirtualBox que nos permite crear una máquina docker en un entorno local. Quizás lo más interesante es utilizar Docker Machine para crear nodos docker en máquinas remotas, para ello tenemos varios drivers según el proveedor: AWS, Microsoft Azure, Google Compute Engine,... En este ejemplo vamos a hacer uso del driver OpenStack para crear una máquina en nuestra infraestructura OpenStack con el demonio docker instalado. Para ello vamos a cargar nuestras credenciales de OpenStack y a continuación creamos la nueva máquina:

```
$ source openrc.sh
Please enter your OpenStack Password:

$ docker-machine create -d openstack \
> --openstack-flavor-id 3 \
> --openstack-image-id 030b7fe8-ed5d-46b8-81fb-62587c944936 \
> --openstack-net-name red \
> --openstack-floatingip-pool ext-net \
> --openstack-ssh-user debian \
> --openstack-sec-groups default \
> --openstack-keypair-name jdmr \
> --openstack-private-key-file ~/.ssh/id_rsa \
> nodo2

Running pre-create checks...
Creating machine...
(nodo2) Creating machine...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with debian...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
```

Al ejecutar el comando anterior se han realizado las siguientes acciones:

- Docker Machine se ha autenticado en OpenStack utilizando las credenciales que hemos cargado.
- Docker Machine ha creado una nueva instancia con las características indicadas. Si no se indica la clave SSH se genera una nueva que será la que se usará para acceder a la instancia.
- Cuando la instancia es accesible por SSH, Docker Machine se conecta a la instancia, instala Docker Engine y lo configura de forma adecuada habilitando TLS).
- Finalmente, recordar que el comando `docker-machine rm` no sólo elimina la referencia local de la máquina, también elimina la instancia que hemos creado.

Como podemos comprobar se ha creado una instancia desde una imagen Debian Jessie, con un sabor *m1.nano*, conectada a nuestra red interna, se le ha asociado una ip flotante, se ha configurado el grupo de seguridad por defecto (recuerda que debes abrir el puerto TCP 2376 para que el servidor docker sea accesible) y se han inyectado mis claves SSH. Podemos comprobar que ya tenemos dada de alta la nueva máquina:

```
$ docker-machine ls
NAME      ACTIVE  DRIVER        STATE     URL                  SWARM   DOCKER   ERRORS
nodo1     -       virtualbox    Running   tcp://192.168.99.100:2376   v1.11.1
nodo2     -       openstack     Running   tcp://172.22.206.15:2376   v1.11.1
```

A continuación creamos las variables de entorno para trabajar con esta máquina y comprobamos (como es evidente) que no tiene ningún contenedor creado:

```
$ eval $(docker-machine env nodo2)
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
```

Puedes obtener más información del uso del driver OpenStack en la documentación oficial (<https://docs.docker.com/machine/drivers/openstack/>).

Utilizando Docker Machine con máquinas que ya tenemos funcionando

En los dos casos anteriores Docker Machine ha sido la responsable de crear las máquinas donde se va a instalar y configurar el demonio docker. En este último caso, ya tenemos una máquina (física o virtual) ya funcionando y queremos gestionarla con Docker Machine. Para conseguir esto tenemos que utilizar el driver genérico (generic) que ejecutará las siguientes tareas:

- Si la máquina no tiene instalado docker, lo instalará y lo configurará.
- Actualiza todos los paquetes de la máquina.
- Genera los certificados TLS para la comunicación segura.
- Reiniciará el Docker Engine, por lo tanto si tuviéramos contenedores, estos serán detenidos.
- Se cambia el nombre de la máquina para que coincida con el que le hemos dado con Docker Machine.

Vamos a gestionar una máquina que ya tenemos funcionando con la siguiente instrucción:

```
$ docker-machine create -d generic \
--generic-ip-address=172.22.205.103 \
--generic-ssh-user=debian \
nodo3

Running pre-create checks...
Creating machine...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with debian...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
```

Por último comprobamos que Docker Machine gestiona el nuevo nodo:

```
$ docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
nodo1	-	virtualbox	Running	tcp://192.168.99.100:2376		v1.11.1	
nodo2	-	openstack	Running	tcp://172.22.206.15:2376		v1.11.1	
nodo3	-	generic	Running	tcp://172.22.205.103:2376		v1.11.1	

Cargamos las variables de entorno del nuevo nodo y ya podemos empezar a trabajar con él:

```
$ eval $(docker-machine env nodo3)
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Como siempre puede obtener más información del driver generic en la documentación oficial (<https://docs.docker.com/machine/drivers/generic/>).

Conclusiones

Esta entrada ha sido una introducción a la herramienta Docker Machine y al uso de diferentes drivers para la creación de nodos docker. Para más información sobre esta herramienta consulta la documentación oficial (<https://docs.docker.com/machine/overview/>).

Posted in General (<http://www.josedomingo.org/pledin/category/general/>).

Tagged docker (<http://www.josedomingo.org/pledin/tag/docker/>), OpenStack (<http://www.josedomingo.org/pledin/tag/open-stack/>), Virtualización (<http://www.josedomingo.org/pledin/tag/virtualizacion/>).

Comments are closed.

Next Post → (<http://www.josedomingo.org/pledin/2016/05/gestion-del-almacenamiento-en-docker/>)

← Previous Post (<http://www.josedomingo.org/pledin/2016/04/transferencias-de-volumenes-en-openstack/>)

Buscar


Etiquetas

ansible (<http://www.josedomingo.org/pledin/tag/ansible/>) Apache (<http://www.josedomingo.org/pledin/tag/apache/>) bottle (<http://www.josedomingo.org/pledin/tag/bottle/>) Caché (<http://www.josedomingo.org/pledin/tag/cache/>) **Cloud Computing** (<http://www.josedomingo.org/pledin/tag/cloud-computing/>) CMS (<http://www.josedomingo.org/pledin/tag/cms/>) **Cursos** (<http://www.josedomingo.org/pledin/tag/cursos/>) devstack (<http://www.josedomingo.org/pledin/tag/devstack/>) dns (<http://www.josedomingo.org/pledin/tag/dns/>) **docker** (<http://www.josedomingo.org/pledin/tag/docker/>) **Educación** (<http://www.josedomingo.org/pledin/tag/educacion/>) General (<http://www.josedomingo.org/pledin/tag/general/>) gns3 (<http://www.josedomingo.org/pledin/tag/gns3/>) Google (<http://www.josedomingo.org/pledin/tag/google/>) Guadalinex (<http://www.josedomingo.org/pledin/tag/guadalinex/>) Hardware (<http://www.josedomingo.org/pledin/tag/hardware/>) html (<http://www.josedomingo.org/pledin/tag/html/>) **Internet** (<http://www.josedomingo.org/pledin/tag/internet/>) Javascript (<http://www.josedomingo.org/pledin/tag/javascript/>) **Libros** (<http://www.josedomingo.org/pledin/tag/libros/>) **Linux** (<http://www.josedomingo.org/pledin/tag/linux/>) **Manuales** (<http://www.josedomingo.org/pledin/tag/manuales/>) Moodle (<http://www.josedomingo.org/pledin/tag/moodle/>) openOffice (<http://www.josedomingo.org/pledin/tag/openoffice/>) OpenShift (<http://www.josedomingo.org/pledin/tag/openshift/>) **OpenStack** (<http://www.josedomingo.org/pledin/tag/open-stack/>) PaaS (<http://www.josedomingo.org/pledin/tag/paas/>) php (<http://www.josedomingo.org/pledin/tag/php/>) **Pledin** (<http://www.josedomingo.org/pledin/tag/pledin/>) Programación (<http://www.josedomingo.org/pledin/tag/programacion/>) **Python** (<http://www.josedomingo.org/pledin/tag/python/>) **Redes** (<http://www.josedomingo.org/pledin/tag/redes/>) Redes Sociales (<http://www.josedomingo.org/pledin/tag/redes-sociales/>) **Revistas** (<http://www.josedomingo.org/pledin/tag/revistas/>) Ruby on Rails (<http://www.josedomingo.org/pledin/tag/ruby-on-rails/>) **Seguridad** (<http://www.josedomingo.org/pledin/tag/seguridad/>) **Software Libre** (<http://www.josedomingo.org/pledin/tag/software-libre/>) Squid (<http://www.josedomingo.org/pledin/tag/squid/>) **TIC** (<http://www.josedomingo.org/pledin/tag/tic/>) Vagrant (<http://www.josedomingo.org/pledin/tag/vagrant/>) **Virtualización** (<http://www.josedomingo.org/pledin/tag/virtualizacion/>) **Web** (<http://www.josedomingo.org/pledin/tag/web/>) Web 2.0 (<http://www.josedomingo.org/pledin/tag/web-2-0/>) **Webinars** (<http://www.josedomingo.org/pledin/tag/webinars/>) xml (<http://www.josedomingo.org/pledin/tag/xml/>)


Twitter

Tweets por @Pledin_JD

José Domingo Muñoz retweeted



Konamito
@Konamito

 Head y Heels 30 años después... tumblr.co/Z-g5xf2OCTEjc

holaaquikonamito

Head y Heels 30 años después...
konamito.es

25 jul. 2017

Insertar

Ver en Twitter

Cloud Computing

- Cloud Computing (html) (<http://www.josedomingo.org/cloud>)
- Cloud Computing (moodle) (<http://www.josedomingo.org/web/course/view.php?id=68>)

Cursos impartidos

- Curso de infraestructura Cloud con OpenStack. Universidad de Almería (2017) (github) (<https://github.com/iesgn/curso-ual17>)
- Curso OpenStack (2016) (github.io) (<http://iesgn.github.io/emergya/>)
- Introducción al Cloud Computing con OpenStack y OpenShift (2015) (github.io) (<http://iesgn.github.io/cloud3/>)
- Administración de infraestructura en la nube con OpenStack (2015) (github.io) (<http://iesgn.github.io/cloud2/>)
- Fundamentos de Cloud Computing con OpenStack y OpenShift (2014) (<http://plataforma.josedomingo.org/pledin/course/view.php?id=23>)
- Fundamentos de Cloud Computing con OpenStack y OpenShift (2014) (github.io) (<http://iesgn.github.io/cloud/>)
- Servicios en GNU/Linux (Nivel Intermedio) (2011) (<http://www.josedomingo.org/web/course/view.php?id=67>)
- Introducción a los Servicios en GNU/Linux (<http://www.josedomingo.org/web/course/view.php?id=63>)
- Máquinas virtuales para la puesta en marcha de un portal educativo (<http://www.josedomingo.org/web/course/view.php?id=43>)
- Servicios en GNU/Linux: Puesta en marcha de un portal educativo (<http://www.josedomingo.org/web/course/view.php?id=65>)
- Curso TIC – IES Gonzalo Nazareno 2007 (<http://www.josedomingo.org/web/course/view.php?id=48><http://>)

Blogroll

- Desde los alto del cerro (<http://albertomolina.wordpress.com/>)
- Gonzalo Nazareno (<http://informatica.gonzalonazareno.org>)
- Plataforma (<http://www.josedomingo.org/web/>)
- Software Libre (<http://sw-libre.blogspot.com/>)
- sysadmit (<http://www.sysadmit.com/>)

Entradas Recientes

- Curso de infraestructura Cloud con OpenStack. Universidad de Almería (<http://www.josedomingo.org/pledin/2017/05/curso-de-infraestructura-cloud-con-openstack-universidad-de-almeria/>)
- Curso de Python3 en OpenWebinars (<http://www.josedomingo.org/pledin/2017/05/curso-de-python3-en-openwebinars/>)
- Despliegue de una aplicación Python Bottle en Heroku (<http://www.josedomingo.org/pledin/2017/04/despliegue-de-una-aplicacion-python-bottle-en-heroku/>)
- Entornos de desarrollo virtuales con python3 (<http://www.josedomingo.org/pledin/2017/03/entornos-de-desarrollo-virtuales-con-python3/>)
- IV Semana Informática. Universidad de Almería (<http://www.josedomingo.org/pledin/2017/03/iv-semana-informatica-universidad-de-almeria/>)

Archivos

- Mayo 2017 (<http://www.josedomingo.org/pledin/2017/05/>) (2)
- Abril 2017 (<http://www.josedomingo.org/pledin/2017/04/>) (1)
- Marzo 2017 (<http://www.josedomingo.org/pledin/2017/03/>) (2)
- Febrero 2017 (<http://www.josedomingo.org/pledin/2017/02/>) (1)
- Noviembre 2016 (<http://www.josedomingo.org/pledin/2016/11/>) (1)
- Mayo 2016 (<http://www.josedomingo.org/pledin/2016/05/>) (2)
- Abril 2016 (<http://www.josedomingo.org/pledin/2016/04/>) (2)
- Febrero 2016 (<http://www.josedomingo.org/pledin/2016/02/>) (7)
- Enero 2016 (<http://www.josedomingo.org/pledin/2016/01/>) (1)
- Diciembre 2015 (<http://www.josedomingo.org/pledin/2015/12/>) (1)
- Noviembre 2015 (<http://www.josedomingo.org/pledin/2015/11/>) (1)
- Julio 2015 (<http://www.josedomingo.org/pledin/2015/07/>) (2)
- Abril 2015 (<http://www.josedomingo.org/pledin/2015/04/>) (1)
- Marzo 2015 (<http://www.josedomingo.org/pledin/2015/03/>) (4)
- Febrero 2015 (<http://www.josedomingo.org/pledin/2015/02/>) (3)
- Enero 2015 (<http://www.josedomingo.org/pledin/2015/01/>) (3)
- Diciembre 2014 (<http://www.josedomingo.org/pledin/2014/12/>) (3)
- Noviembre 2014 (<http://www.josedomingo.org/pledin/2014/11/>) (1)
- Octubre 2014 (<http://www.josedomingo.org/pledin/2014/10/>) (2)
- Mayo 2014 (<http://www.josedomingo.org/pledin/2014/05/>) (1)
- Abril 2014 (<http://www.josedomingo.org/pledin/2014/04/>) (1)
- Marzo 2014 (<http://www.josedomingo.org/pledin/2014/03/>) (1)
- Febrero 2014 (<http://www.josedomingo.org/pledin/2014/02/>) (3)
- Enero 2014 (<http://www.josedomingo.org/pledin/2014/01/>) (2)
- Noviembre 2013 (<http://www.josedomingo.org/pledin/2013/11/>) (3)