

How to Deploy Galera Cluster for MySQL using Docker Containers

Virtual Machines are great, and very useful when trying out new software. However, they might be an unnecessarily heavyweight solution when testing clusters, especially if these consist of multiple nodes running exactly the same software. Each VM runs a full-blown OS image. On the other hand, Linux Containers (LXC) are an efficient alternative to OS-level virtualization and can run multiple isolated systems on a single host. Docker is a wrapper around LXC, it automates the deployment of applications inside containers.



A notable advantage if you run with Docker + LXC is you can run many containers on a single host. They all share the same OS as the host, and when possible, the same binaries. Deployment can be extremely fast. Using Docker could be a good method if you want to spin multiple Galera nodes within a single host.

In this post, we will create MySQL Galera Cluster containers using Docker, and fire them up to form a database cluster running on a single host.

Install LXC and Docker

1. Docker works best on kernel 3.8 (due to a bug in LXC, at the time of writing). Install the kernel image for Raring:

```
$ sudo apt-get update
$ sudo apt-get install -y linux-image-generic-lts-raring linux-headers-generic-lts-raring
$ sudo init 6
```

2. Verify the kernel version:

```
$ uname -a
Linux lxc 3.8.0-35-generic #52~precise1-Ubuntu SMP Thu Jan 30 17:24:40 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
```

3. Add the repository keychain locally, add Docker repository, update the package list and perform installation:

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 36A1D7869245C8950F966E92D8576A8BA88D21E9
$ sudo sh -c "echo deb http://get.docker.io/ubuntu docker main > /etc/apt/sources.list.d/docker.list"
$ sudo apt-get update
$ sudo apt-get install -y lxc-docker mysql-client
```

4. Modify the forwarding policy for ufw:

```
$ sudo nano /etc/default/ufw
```

Modify the following line:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Create Docker Image

1. Pull Ubuntu images from Docker repository:

```
$ sudo docker pull ubuntu
```

2. Create Docker build script file:

```
$ vim Dockerfile
```

And paste the following lines:

```
# # MySQL Galera Cluster 5.5.34-25.9/Galera 2.8/Ubuntu 12.04 64bit
FROM ubuntu:12.04
MAINTAINER Severalnines AB <info@severalnines.com>

RUN apt-key adv --keyserver keys.gnupg.net --recv-keys 1C4CBDCDCD2EFD2A
RUN sed -i '/repo.percona.com/d' /etc/apt/sources.list
RUN sh -c 'echo "deb http://repo.percona.com/apt precise main" >> /etc/apt/sources.list'
RUN sh -c 'echo "deb-src http://repo.percona.com/apt precise main" >> /etc/apt/sources.list'
RUN apt-get -q -y update
RUN LC_ALL=en_US.utf8 DEBIAN_FRONTEND=noninteractive apt-get -o Dpkg::Options::='--force-confnew' -qqy install nano wget psmisc libdbi-perl
RUN wget --no-check-certificate https://launchpad.net/codership-mysql/5.5/5.5.34-25.9/+download/mysql-server-wsrep-5.5.34-25.9-amd64.deb
RUN wget --no-check-certificate https://launchpad.net/galera/2.x/25.2.8/+download/galera-25.2.8-amd64.deb
RUN dpkg -i galera-25.2.8-amd64.deb
RUN dpkg -i mysql-server-wsrep-5.5.34-25.9-amd64.deb
ADD ./my.cnf /etc/mysql/my.cnf
EXPOSE 3306 4444 4567 4568
```

3. Create a default MySQL configuration file for our Galera nodes that will be used by Docker (as shown at the ADD line above):

```
$ vim my.cnf
```

And paste following lines:

```
[mysqld]
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_address=gcomm://
wsrep_sst_method=rsync
wsrep_cluster_name=galera_cluster
binlog_format=ROW
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
innodb_locks_unsafe_for_binlog=1
```

4. Build the image:

```
$ sudo docker build -t ubuntu_precise/galera:codership .
```

5. Verify whether the newly created image is listed by Docker:

```
$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
ubuntu_precise/galera	codership	ade20ad1d10b	3 minutes ago	425.8 MB
ubuntu	13.10	9f676bd305a4	46 hours ago	178 MB
ubuntu	saucy	9f676bd305a4	46 hours ago	178 MB
ubuntu	13.04	eb601b8965b8	46 hours ago	166.5 MB
ubuntu	raring	eb601b8965b8	46 hours ago	166.5 MB
ubuntu	12.10	5ac751e8d623	46 hours ago	161 MB
ubuntu	quantal	5ac751e8d623	46 hours ago	161 MB
ubuntu	lucid	9cc9ea5ea540	46 hours ago	180.8 MB
ubuntu	10.04	9cc9ea5ea540	46 hours ago	180.8 MB
ubuntu	latest	9cd978db300e	46 hours ago	204.4 MB
ubuntu	12.04	9cd978db300e	46 hours ago	204.4 MB
ubuntu	precise	9cd978db300e	46 hours ago	204.4 MB

Launch Containers

1. Launch three Galera Containers (this just takes like less than 5 seconds!):

```
$ for n in {1..3}; do sudo docker run -name galera$n -i -t -d ubuntu_precise/galera:codership /bin/bash; done
```

2. Verify that the Containers have been created correctly:

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
be2b2927c40e	ubuntu_precise/galera:codership	/bin/bash	About a minute ago	Up About a minute	3306/tcp, 4444/tcp, 4567/
ddb1e1db236d	ubuntu_precise/galera:codership	/bin/bash	About a minute ago	Up About a minute	3306/tcp, 4444/tcp, 4567/
99e33954479a	ubuntu_precise/galera:codership	/bin/bash	About a minute ago	Up About a minute	3306/tcp, 4444/tcp, 4567/