

Install and Run Memcached as a service using docker

Table of Contents

Requirements.....	2
Installing Memcached on Docker	3
Check for docker daemon	3
Installing Memcached	3
Commit your new container	4
Running and using Memcached on Docker	5
Running Memcached on Docker	6
Using Memcached.....	6
Testing Memcached service.....	8
Testing with Ruby.....	8
Testing with Python	8
Testing with PHP	9
Testing with Go	9

Install and Run Memcached as a service using docker on your server

Requirements

1. Install last version of docker on your Operating System

Please visit <http://www.docker.io/gettingstarted/> to get docker installed on your Ubuntu or using Vagrant + VirtualBox on any other Operating system

2. Make sure you are running docker as a daemon. If it is not the case you can launch docker daemon by running

```
sudo docker -d &
```

```
julien@cs50:~/docker-master$ sudo ./docker -d &  
[1] 5549  
julien@cs50:~/docker-master$ 2013/04/10 16:33:49 Listening for RCLI/tcp on 127.0.0.1:4242
```

Install and Run Memcached as a service using docker on your server

Installing Memcached on Docker

Check for docker daemon

Before starting, make sure docker is running as a daemon on your system. If it is not the case, please read the “Requirements” Section.

```
ps aux | grep docker
```

```
julien@cs50:~/docker-master$ ps aux | grep docker
root      5549  0.0  0.1 56816 1988 pts/4    S   16:33   0:00 sudo ./docker -d
root      5550  0.0  0.2 137436 4300 pts/4    Sl  16:33   0:00 ./docker -d
julien    5566  0.0  0.0  8108   884 pts/4    S+  16:37   0:00 grep --color=auto docker
```

You should see a line with `docker -d`

Installing Memcached

```
docker run -d base apt-get -y install memcached
```

```
julien@cs50:~/docker-master$ docker run -d base apt-get -y install memcached
2013/04/10 16:40:33 docker run -d base apt-get -y install memcached
f1ab59fbc9d5
```

This will give you the id of the new created container running your command `apt-get`. You will need to keep this id in order to use it later. In our example, the id is `f1ab59fbc9d5`

We can check that the installation is completed by using the `docker logs` command
`docker logs f1ab59fbc9d5 | less`

Remember to replace `f1ab59fbc9d5` by your id before running this command.

```
julien@cs50:~/docker-master$ docker logs f1ab59fbc9d5 | less
```

Install and Run Memcached as a service using docker on your server

```
Reading package lists...
Building dependency tree...
The following extra packages will be installed:
  libclass-isa-perl libevent-2.0-5 libgdbm3 libsasl2-2 libsasl2-modules
  libswitch-perl perl perl-modules
Suggested packages:
  libsasl2-modules-otp libsasl2-modules-ldap libsasl2-modules-sql
  libsasl2-modules-gssapi-mit libsasl2-modules-gssapi-heimdal
  libcache-memcached-perl libmemcached perl-doc libterm-readline-gnu-perl
  libterm-readline-perl-perl make libpod-plainer-perl
The following NEW packages will be installed:
  libclass-isa-perl libevent-2.0-5 libgdbm3 libsasl2-2 libsasl2-modules
  libswitch-perl memcached perl perl-modules
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 8207 kB of archives.
After this operation, 33.2 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu/ quantal/main libgdbm3 amd64 1.8.3-11 [34.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu/ quantal/main libsasl2-2 amd64 2.1.25.dfsg1-5 [69.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu/ quantal/main libevent-2.0-5 amd64 2.0.19-stable-3 [126 kB]
Get:4 http://archive.ubuntu.com/ubuntu/ quantal/main libsasl2-modules amd64 2.1.25.dfsg1-5 [63.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu/ quantal/main libclass-isa-perl all 0.36-3 [11.9 kB]
Get:6 http://archive.ubuntu.com/ubuntu/ quantal/main perl-modules all 5.14.2-13 [3390 kB]
Get:7 http://archive.ubuntu.com/ubuntu/ quantal/main perl amd64 5.14.2-13 [4417 kB]
Get:8 http://archive.ubuntu.com/ubuntu/ quantal/main libswitch-perl all 2.16-2 [19.2 kB]
Get:9 http://archive.ubuntu.com/ubuntu/ quantal/main memcached amd64 1.4.14-0ubuntu1 [75.6 kB]
Fetched 8207 kB in 0s (12.1 MB/s)
Selecting previously unselected package libgdbm3:amd64.
(Reading database ... 9893 files and directories currently installed.)
Unpacking libgdbm3:amd64 (from .../libgdbm3_1.8.3-11_amd64.deb) ...
Selecting previously unselected package libsasl2-2:amd64.
Unpacking libsasl2-2:amd64 (from .../libsasl2-2_2.1.25.dfsg1-5_amd64.deb) ...
Selecting previously unselected package libevent-2.0-5:amd64.
Unpacking libevent-2.0-5:amd64 (from .../libevent-2.0-5_2.0.19-stable-3_amd64.deb) ...
Selecting previously unselected package libsasl2-modules:amd64.
Unpacking libsasl2-modules:amd64 (from .../libsasl2-modules_2.1.25.dfsg1-5_amd64.deb) ...
Selecting previously unselected package libclass-isa-perl.
Unpacking libclass-isa-perl (from .../libclass-isa-perl_0.36-3_all.deb) ...
Selecting previously unselected package perl-modules.
Unpacking perl-modules (from .../perl-modules_5.14.2-13_all.deb) ...
Selecting previously unselected package perl.
Unpacking perl (from .../perl_5.14.2-13_amd64.deb) ...
Selecting previously unselected package libswitch-perl.
Unpacking libswitch-perl (from .../libswitch-perl_2.16-2_all.deb) ...
:
```

Commit your new container

We can commit our new container with the `docker commit` command, using our id.

```
docker commit f1ab59fbc9d5 jbarbier/memcached
```

Remember to replace `f1ab59fbc9d5` by your id before running this command. You can also replace `jbarbier/memcached` with your own repository name.

Install and Run Memcached as a service using docker on your server

```
julien@cs50:~/docker-master$ docker commit f1ab59fbc9d5 jbarbier/memcached
2013/04/10 16:46:55 docker commit f1ab59fbc9d5 jbarbier/memcached
c3b6fcb48266
```

This command gives you a new id which is the image id of your container. In our example it is `c3b6fcb48266`. Note that id so we can use it later.

Let's check that Memcached is installed on this image. To do so we can spawn a new container from this image and run bash inside.

```
docker run -i -t jbarbier/memcached /bin/bash
```

Remember to replace `jbarbier/memcached` by your repository name before running this command.

```
julien@cs50:~/docker-master$ docker run -i -t jbarbier/memcached /bin/bash
2013/04/10 16:48:26 docker run -i -t jbarbier/memcached /bin/bash
root@67d1669256d8:/#
```

We are now inside a container spawned from our image. Let's see if Memcached is installed

```
memcached
```

```
root@67d1669256d8:/# memcached
can't run as root without the -u switch
```

OK!

Note that you could have used the id of your image instead of the name of your repository.

```
docker run -i -t c3b6fcb48266 /bin/bash
```

Remember to replace `c3b6fcb48266` by your image id before running this command.

```
julien@cs50:~/docker-master$ docker run -i -t c3b6fcb48266 /bin/bash
2013/04/10 16:52:07 docker run -i -t c3b6fcb48266 /bin/bash
root@48e8f568ba0c:/# memcached
can't run as root without the -u switch
```

Running and using Memcached on Docker

Now that we have an image with Memcached installed, let's use it :)

Install and Run Memcached as a service using docker on your server

Running Memcached on Docker

```
docker run -d -p 11211 jbarbier/memcached memcached -u daemon
```

Remember to replace `jbarbier/memcached` by your repository name before running this command.

We need the `-u` option because Memcache can not run as root.

In order to be able to use Memcached from outside our server, we can use the `-p 11211` option, which will ask docker to map the internal port of the container used by Memcached (11211), with a public port of the host.

```
julien@cs50:~/docker-master$ docker run -d -p 11211 jbarbier/memcached memcached -u daemon
2013/04/10 16:54:27 docker run -d -p 11211 jbarbier/memcached memcached -u daemon
c360f228e22f
```

As usual, docker gives you back the id of the container you launched. In our case it is `c360f228e22f`. Note that id in order to use it later.

Using Memcached

In order to use Memcached from outside the localhost we need to know the host public port mapped by docker. In order to know that we can use the `docker inspect` command.

```
docker inspect c360f228e22f
```

Remember to replace `c360f228e22f` by your container id before running this command.

This will give you a JSON output with plenty of configuration details. In the `NetworkSettings/PortMapping` you will find the public port you can use Memcached with from outside your server.

Install and Run Memcached as a service using docker on your server

```
julien@cs50:~/docker-master$ docker inspect c360f228e22f
2013/04/10 16:56:13 docker inspect c360f228e22f
{
  "Id": "c360f228e22f6b01fc833993bf85b109677931e5ab6a14d41586cfd6e06d7dbe",
  "Created": "2013-04-10T16:54:27.349299Z",
  "Path": "memcached",
  "Args": [
    "-u",
    "daemon"
  ],
  "Config": {
    "Hostname": "c360f228e22f",
    "User": "",
    "Memory": 0,
    "MemorySwap": 0,
    "AttachStdin": false,
    "AttachStdout": false,
    "AttachStderr": false,
    "PortSpecs": [
      "11211"
    ],
    "Tty": false,
    "OpenStdin": false,
    "StdinOnce": false,
    "Env": null,
    "Cmd": [
      "memcached",
      "-u",
      "daemon"
    ],
    "Image": "jbarbier/memcached"
  },
  "State": {
    "Running": true,
    "Pid": 5931,
    "ExitCode": 0,
    "StartedAt": "2013-04-10T16:54:27.426066Z"
  },
  "Image": "c3b6fc4826610a3aa9f21bdd47466c782769b8d36be18f4b53e96deaa9bebc4",
  "NetworkSettings": {
    "IpAddress": "172.16.42.5",
    "IpPrefixLen": 24,
    "Gateway": "172.16.42.1",
    "Bridge": "docker0",
    "PortMapping": {
      "11211": "49153"
    }
  },
  "SysInitPath": "/home/julien/docker-master/docker"
}
```

Install and Run Memcached as a service using docker on your server

In our case the public port is 49153.

Testing Memcached service

Let's test and use our Memcached service, from an outside machine. In the following examples I will use 142.242.242.42 as the IP of the server where the container is running, and 49153 as the public port.

Before running any of these examples be sure to replace the IP with your server IP, and the port number with the one `docker inspect` gave you.

Testing with Ruby

```
Guillotine:test_memcached jbarbier$ cat test.rb
# run gem install dalli first

require 'dalli'

ip = '142.242.242.42'
port = 49153

dc = Dalli::Client.new("#{ip}:#{port}")
dc.set('abc', "Always Be Closing")
value = dc.get('abc')

puts value
```

```
ruby test.rb
```

```
Guillotine:test_memcached jbarbier$ ruby test.rb
Always Be Closing
```

Testing with Python

```
Guillotine:test_memcached jbarbier$ cat test.py
# pip install python-memcached if import memcache fails

import memcache

ip = '142.242.242.42'
port = 49153

mc = memcache.Client(["{0}:{1}".format(ip, port)], debug=0)
```


Install and Run Memcached as a service using docker on your server

```
mc.set("best_dev", "Guillaume C.")
value = mc.get("best_dev")

print value
```

```
python test.py
```

```
Guillotine:test_memcached jbarbier$ python test.py
Guillaume C.
```

Testing with PHP

```
Guillotine:test_memcached jbarbier$ cat test.php
<?php

$ip = '142.242.242.42';
$port = 49153;

$memcache_obj = new Memcache;
$memcache_obj->connect($ip, $port);

$memcache_obj->set('rule_1', 'You DO NOT talk about FIGHT CLUB');
$v = $memcache_obj->get('rule_1');

echo "$v\n";

?>
```

```
php test.php
```

```
Guillotine:test_memcached jbarbier$ php test.php
You DO NOT talk about FIGHT CLUB
```

Testing with Go

```
Guillotine:test_memcached jbarbier$ cat test.go
package main

import (
    "fmt"
    "github.com/kklis/gomemcache"
)
```

Install and Run Memcached as a service using docker on your server

```
func main() {  
  
ip := "142.242.242.42"  
port := 49153  
  
memc, err := gomemcache.Connect(ip, port)  
if err != nil {  
panic(err)  
}  
err = memc.Set("foo", []byte("bar"), 0, 0)  
if err != nil {  
panic(err)  
}  
val, _, _ := memc.Get("foo")  
fmt.Printf("%s\n", val)  
}
```

Go run test.go

```
Guillotine:test_memcached jbarbier$ go run test.go  
bar
```