

Installing & Configuring Varnish - Memcached

***The following configurations were made in Drupal 7.*

Installing Varnish:

`sudo apt-get install varnish`

Changing varnish settings:

First, we have to change the default port. Edit `/etc/default/varnish`:

`vim /etc/default/varnish`

Scroll down a bit, until you find an uncommented line starting with `"DAEMON_OPTS"`.

- Change `*:6081` to `*:80` so it will listen on the default HTTP port.
- Change the ip in the `-T` parameter
- Edit `default.vcl` to something else, I took `"drupal-test.vcl"`.
- Change the storage amount of memory to the 25% of your server's memory.

```
DAEMON_OPTS="-a :80 \  
-T 172.16.17.30:6082 \  
-f /etc/varnish/drupal-test.vcl \  
-S /etc/varnish/secret \  
-s file,/var/lib/varnish/$INSTANCE/varnish_storage.bin,512MB"
```

Save the file.

Edit the VCL file you mentioned in the previous file. In my case, I'll be editing `/etc/varnish/drupal-test.vcl`. Paste the following contents:

```
## Redirect requests to Apache, running on port 8000 on localhost  
backend apache {  
    .host = "172.16.17.30";  
    .port = "8000";  
}  
## Fetch  
sub vcl_fetch {  
    ## Remove the X-Forwarded-For header if it exists.  
    remove req.http.X-Forwarded-For;  
  
    ## insert the client IP address as X-Forwarded-For. This is the normal IP address of the use$  
    set req.http.X-Forwarded-For = req.http.rlnclientipaddr;  
    ## Added security, the "w00tw00t" attacks are pretty annoying so lets block it before it rea$  
    if (req.url ~ "^/w00tw00t") {  
        error 403 "Not permitted";  
    }  
    ## Deliver the content  
    return(deliver);  
}  
  
## Deliver  
sub vcl_deliver {
```

```

    if (obj.hits > 0) {
        set resp.http.X-Cache = "HIT";
    } else {
        set resp.http.X-Cache = "MISS";
    }
}

sub vcl_recv {
    // Remove has_js and Google Analytics __* cookies.
    set req.http.Cookie = regsuball(req.http.Cookie, "(^|;|s*)(__[a-z]+|has_js)=[^;]*", "");
    // Remove a ";" prefix, if present.
    set req.http.Cookie = regsub(req.http.Cookie, "^;s*", "");
    // Remove empty cookies.
    if (req.http.Cookie ~ "^s*$") {
        unset req.http.Cookie;
    }

    // Cache all requests by default, overriding the
    // standard Varnish behavior.
    // if (req.request == "GET" || req.request == "HEAD") {
    //     return (lookup);
    // }
}

sub vcl_hash {
    if (req.http.Cookie) {
        set req.hash += req.http.Cookie;
    }
}

```

Changing Apache settings:

We have to let Apache2 listen on localhost. For this, there are a few small changes required.

`vim /etc/apache2/ports.conf`

Change:

`NameVirtualHost *:80` to `NameVirtualHost *:8000`

`Listen 80` to `<your_ip>:8000`

Apache will listen on that port. You will have to edit your vhosts as well. Open your vhost(s) file(s) – usually they are the `/etc/apache2/sites-available/default` - and replace `<VirtualHost *:80>` to `<VirtualHost *:8000>`

We now have to install an extra Apache module to make sure the IP address of the user ends up correct. Since Varnish is basically talking with Apache2, you would see your_ip as visitor IP

`apt-get install libapache2-mod-rpaf`

The RPAF (Reverse Proxy Add Forward) module will make sure the IP of your_ip will be replaced with the IP set in X-Forwarded-For set by Varnish.

Restart Daemons:

`/etc/init.d/apache2 restart`

Check if it is bound to the correct IP/Port by executing: `netstat -lp | grep apache2`

If you see something like

`tcp 0 0 localhost:8000 *:* LISTEN 4586/apache2` this is correct.

Now we have to restart Varnish to let it listen on port 80:

`/etc/init.d/varnish restart`

We check this again by executing: `netstat -lp | grep varnish`

The result will be something like:

```
tcp  0  0  *:www  *:~ LISTEN  4498/varnishd
tcp6 0  0  [::]:www [::]:~ LISTEN  4498/varnishd
```

Configuration in drupal's 7 settings.php file:

We change the following lines:

```
$conf['reverse_proxy'] = TRUE;
$conf['reverse_proxy_addresses'] = array('put.your.ip.here');
$conf['reverse_proxy_header'] = 'HTTP_X_FORWARDED_FOR';
```

* In case we will install the varnish drupal module, we add also the following cache configuration in the settings.php file options:

```
// Add Varnish as the page cache handler.
$conf['cache_backends'] = array('sites/all/modules/varnish/varnish.cache.inc');
$conf['cache_class_cache_page'] = 'VarnishCache';
// Drupal 7 does not cache pages when we invoke hooks during bootstrap. This needs
// to be disabled.
$conf['page_cache_invoke_hooks'] = FALSE;
```

Installing Memcached:

We install first the memcached package and the php5-memcached module.

```
apt-get install memcached php5-memcached
```

Memcached configuration and test:

We modify the file `/etc/memcached.conf` setting our ip address:

```
# Specify which IP address to listen on. The default is to listen on all IP addresses
# This parameter is one of the only security measures that memcached has, so make sure
# it's listening on a firewalled interface.
#-l 127.0.0.1
-l 172.16.17.30
```

We create next the info.php file that contains simple: `<?php phpinfo();?>`

Restart the Apache:

```
/etc/init.d/apache2 restart
```

and executing the info.php in our browser we will see the following table with the memcached details.

memcache

memcache support	enabled
Version	3.0.4
Revision	\$Revision: 1.83.2.36 \$

Directive	Local Value	Master Value
memcache.allow_failover	1	1
memcache.chunk_size	32768	32768
memcache.compress_threshold	20000	20000
memcache.default_port	11211	11211
memcache.hash_function	crc32	crc32
memcache.hash_strategy	consistent	consistent
memcache.lock_timeout	15	15
memcache.max_failover_attempts	20	20
memcache.protocol	ascii	ascii
memcache.redundancy	1	1
memcache.session_redundancy	2	2

Installing Memcache API drupal's module:

drush dl memcache

Configuration in drupal's 7 settings.php file:

We add the following lines at the end of the file.

```
// the path to the core cache file
include_once('./includes/cache.inc');
// the path to the memcache cache file
include_once('./sites/all/modules/memcache/memcache.inc');
// make MemCacheDrupal the default cache class
$conf = array(
  'cache_default_class' => 'MemCacheDrupal',
  'memcache_servers' => array('172.16.17.30:11211' => 'default'),
  'memcache_bins' => array('cache' => 'default')
);
```

***The following configurations were made in Pressflow 6. This is a distribution of Drupal with integrated performance, scalability, availability, and testing enhancements. Pressflow addresses a long-standing problem: High-traffic sites use stable versions of Drupal, and stable versions of Drupal are ineligible for enhancements to solve performance bottlenecks discovered after widespread deployment*

Configuring varnish to work with Pressflow:

As we did before, we create the drupal6-test.vcl file in the /etc/varnish/ folder. We open next the /etc/default/varnish file and we change the line `-f/etc/varnish/drupal-test.vcl \` to point to the drupal6-test.vcl (`-f /etc/varnish/drupal6-test.vcl \`)

```
backend default {
    .host = "172.16.17.30";
    .port = "8000";
}

## Fetch
sub vcl_fetch {
    ## Remove the X-Forwarded-For header if it exists.
    remove req.http.X-Forwarded-For;

    ## insert the client IP address as X-Forwarded-For. This is the normal IP address of$
    set req.http.X-Forwarded-For = req.http.rlnclientipaddr;
    ## Added security, the "w00tw00t" attacks are pretty annoying so lets block it befor$
    if (req.url ~ "^/w00tw00t") {
        error 403 "Not permitted";
    }
    ## Deliver the content
    return(deliver);
}

## Deliver
sub vcl_deliver {
    /*
    ## We'll be hiding some headers added by Varnish. We want to make sure people are not seeing
    ## Since we're not caching (yet), why bother telling people we use it?
    remove resp.http.X-Varnish;
    remove resp.http.Via;
    remove resp.http.Age;

    ## We'd like to hide the X-Powered-By headers. Nobody has to know that we can run PHP
    remove resp.http.X-Powered-By;
    */

    if (obj.hits > 0) {
        set resp.http.X-Cache = "HIT";
    } else {
        set resp.http.X-Cache = "MISS";
    }
}

sub vcl_recv {
    ## Remove has_js and Google Analytics cookies.
    set req.http.Cookie = regsuball(req.http.Cookie, "(^|;|s*)(__[a-z]+|has_js)=[^;]*", "");
    ## Remove a ";" prefix, if present.
    set req.http.Cookie = regsub(req.http.Cookie, "^;s*", "");
    ## Remove empty cookies.
    if (req.http.Cookie ~ "^s*$") {
        unset req.http.Cookie;
    }
}
```

```

## Catch Drupal theme files . THIS BREAKS UPDATE.PHP
# if (req.url ~ "^/sites/") {
#   unset req.http.Cookie;
# }
# Catch Drupal misc files (like drupal.js and jquery.js)
# if (req.url ~ "^/misc/") {
#   unset req.http.Cookie;
# }

# Site still uses some static files out of /files, cache them
if (req.url ~ "^/files/site.*") {
  unset req.http.Cookie;
}
# enable caching of theme files (can't enable globally due to update.php problem above)
# if (req.url ~ "^/sites/www.site.*") {
#   unset req.http.Cookie;
# }

# Drupal js/css doesn't need cookies, cache them
if (req.url ~ "^/modules/.*\\.js|css\\.?") {
  unset req.http.Cookie;
}

## Moodle themes . disabled, seems to cause random problems
# if (req.url ~ "^/(theme|pix)/") {
#   unset req.http.Cookie;
# }

}

sub vcl_hash {
  if (req.http.Cookie) {
    set req.hash += req.http.Cookie;
  }
}

```

Configuring settings.php in Pressflow

In the settings.php file we add the following lines:

```

$conf = array(
  'reverse_proxy' => TRUE,
  'reverse_proxy_addresses' => array('172.16.17.30'),
  'reverse_proxy_header' => 'HTTP_X_FORWARDED_FOR',
);

```

Installing Memcache API drupal's module in Pressflow:

```
drush dl memcache
```

Configuring settings.php in Pressflow

And we add `$conf['cache_inc'] = 'sites/all/modules/memcache/memcache.inc';`