

 免费电子书

学习

varnish

Free unaffiliated eBook created from
Stack Overflow contributors.

#varnish

.....	1
1:	2
.....	2
.....	2
Examples.....	2
.....	2
CentOS 7.....	2
Ubuntu.....	2
Debian.....	2
VCL.....	3
2: VCL	4
.....	4
Examples.....	4
3.0.....	4
4.0.....	6
3: vmods	10
.....	10
Examples.....	10
vmod.....	10
4:	11
.....	11
Examples.....	11
-.....	11
.....	11
.....	12
.....	12
.....	12
.....	12
.....	14

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [varnish](#)

It is an unofficial and free varnish ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official varnish.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1:

o

o o

5.1.2	201747
5.1.1	2017316
5	2016915
4.1.5	201629
4.0.4	
3.0.7	2015323

Examples

LinuxVarnish

CentOS 7

```
curl -s https://packagecloud.io/install/repositories/varnishcache/varnish5/script.rpm.sh |
sudo bash
```

Ubuntu

```
apt-get install apt-transport-https
curl https://repo.varnish-cache.org/GPG-key.txt | apt-key add -
echo "deb https://repo.varnish-cache.org/ubuntu/ trusty varnish-4.1" \
  >> /etc/apt/sources.list.d/varnish-cache.list
apt-get update
apt-get install varnish
```

Debian

```
apt-get install apt-transport-https
curl https://repo.varnish-cache.org/GPG-key.txt | apt-key add -
echo "deb https://repo.varnish-cache.org/debian/ jessie varnish-4.1" \
  >> /etc/apt/sources.list.d/varnish-cache.list
apt-get update
apt-get install varnish
```

VCL

Varnish VCL HTTP。 VCL cookie/ images

```
sub vcl_recv {
    if (req.url ~ "^/images") {
        unset req.http.cookie;
    }
}
```

<https://riptutorial.com/zh-CN/varnish/topic/4705/>

2: VCL

VCLVarnish。

VCL。 POST/cookie。

VCLreturn()。

Examples

3.0

```
/*-
 *
 * The default VCL code.
 *
 * NB! You do NOT need to copy & paste all of these functions into your
 * own vcl code, if you do not provide a definition of one of these
 * functions, the compiler will automatically fall back to the default
 * code from this file.
 *
 */

sub vcl_recv {
    if (req.restarts == 0) {
        if (req.http.x-forwarded-for) {
            set req.http.X-Forwarded-For =
                req.http.X-Forwarded-For + ", " + client.ip;
        } else {
            set req.http.X-Forwarded-For = client.ip;
        }
    }
    if (req.request != "GET" &&
        req.request != "HEAD" &&
        req.request != "PUT" &&
        req.request != "POST" &&
        req.request != "TRACE" &&
        req.request != "OPTIONS" &&
        req.request != "DELETE") {
        /* Non-RFC2616 or CONNECT which is weird. */
        return (pipe);
    }
    if (req.request != "GET" && req.request != "HEAD") {
        /* We only deal with GET and HEAD by default */
        return (pass);
    }
    if (req.http.Authorization || req.http.Cookie) {
        /* Not cacheable by default */
        return (pass);
    }
    return (lookup);
}

sub vcl_pipe {
    # Note that only the first request to the backend will have
```

```

# X-Forwarded-For set.  If you use X-Forwarded-For and want to
# have it set for all requests, make sure to have:
# set bereq.http.connection = "close";
# here.  It is not set by default as it might break some broken web
# applications, like IIS with NTLM authentication.
return (pipe);
}

sub vcl_pass {
    return (pass);
}

sub vcl_hash {
    hash_data(req.url);
    if (req.http.host) {
        hash_data(req.http.host);
    } else {
        hash_data(server.ip);
    }
    return (hash);
}

sub vcl_hit {
    return (deliver);
}

sub vcl_miss {
    return (fetch);
}

sub vcl_fetch {
    if (beresp.ttl <= 0s ||
        beresp.http.Set-Cookie ||
        beresp.http.Vary == "*") {
        /*
         * Mark as "Hit-For-Pass" for the next 2 minutes
         */
        set beresp.ttl = 120 s;
        return (hit_for_pass);
    }
    return (deliver);
}

sub vcl_deliver {
    return (deliver);
}

sub vcl_error {
    set obj.http.Content-Type = "text/html; charset=utf-8";
    set obj.http.Retry-After = "5";
    synthetic {"
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>} + obj.status + " " + obj.response + {"</title>
</head>
<body>
<h1>Error "} + obj.status + " " + obj.response + {"</h1>
<p>} + obj.response + {"</p>

```

```

    <h3>Guru Meditation:</h3>
    <p>XID: " + req.xid + {"</p>
    <hr>
    <p>Varnish cache server</p>
</body>
</html>
";
    return (deliver);
}

sub vcl_init {
    return (ok);
}

sub vcl_fini {
    return (ok);
}

```

4.0

```

/*
 * The built-in (previously called default) VCL code.
 *
 * NB! You do NOT need to copy & paste all of these functions into your
 * own vcl code, if you do not provide a definition of one of these
 * functions, the compiler will automatically fall back to the default
 * code from this file.
 *
 * This code will be prefixed with a backend declaration built from the
 * -b argument.
 */

vcl 4.0;

#####
# Client side

sub vcl_recv {
    if (req.method == "PRI") {
        /* We do not support SPDY or HTTP/2.0 */
        return (synth(405));
    }
    if (req.method != "GET" &&
        req.method != "HEAD" &&
        req.method != "PUT" &&
        req.method != "POST" &&
        req.method != "TRACE" &&
        req.method != "OPTIONS" &&
        req.method != "DELETE") {
        /* Non-RFC2616 or CONNECT which is weird. */
        return (pipe);
    }

    if (req.method != "GET" && req.method != "HEAD") {
        /* We only deal with GET and HEAD by default */
        return (pass);
    }
    if (req.http.Authorization || req.http.Cookie) {
        /* Not cacheable by default */

```



```

        return (pass);
    }
    return (hash);
}

sub vcl_pipe {
    # By default Connection: close is set on all piped requests, to stop
    # connection reuse from sending future requests directly to the
    # (potentially) wrong backend. If you do want this to happen, you can undo
    # it here.
    # unset bereq.http.connection;
    return (pipe);
}

sub vcl_pass {
    return (fetch);
}

sub vcl_hash {
    hash_data(req.url);
    if (req.http.host) {
        hash_data(req.http.host);
    } else {
        hash_data(server.ip);
    }
    return (lookup);
}

sub vcl_purge {
    return (synth(200, "Purged"));
}

sub vcl_hit {
    if (obj.ttl >= 0s) {
        // A pure unadulterated hit, deliver it
        return (deliver);
    }
    if (obj.ttl + obj.grace > 0s) {
        // Object is in grace, deliver it
        // Automatically triggers a background fetch
        return (deliver);
    }
    // fetch & deliver once we get the result
    return (fetch);
}

sub vcl_miss {
    return (fetch);
}

sub vcl_deliver {
    return (deliver);
}

/*
 * We can come here "invisibly" with the following errors: 413, 417 & 503
 */
sub vcl_synth {
    set resp.http.Content-Type = "text/html; charset=utf-8";
    set resp.http.Retry-After = "5";
    synthetic( {"<!DOCTYPE html>"

```

```

<html>
  <head>
    <title>} + resp.status + " " + resp.reason + {"</title>
  </head>
  <body>
    <h1>Error "} + resp.status + " " + resp.reason + {"</h1>
    <p>} + resp.reason + {"</p>
    <h3>Guru Meditation:</h3>
    <p>XID: "} + req.xid + {"</p>
    <hr>
    <p>Varnish cache server</p>
  </body>
</html>
"} );
  return (deliver);
}

#####
# Backend Fetch

sub vcl_backend_fetch {
  return (fetch);
}

sub vcl_backend_response {
  if (beresp.ttl <= 0s ||
      beresp.http.Set-Cookie ||
      beresp.http.Surrogate-control ~ "no-store" ||
      (!beresp.http.Surrogate-Control &&
       beresp.http.Cache-Control ~ "no-cache|no-store|private") ||
      beresp.http.Vary == "*") {
    /*
     * Mark as "Hit-For-Pass" for the next 2 minutes
     */
    set beresp.ttl = 120s;
    set beresp.uncacheable = true;
  }
  return (deliver);
}

sub vcl_backend_error {
  set beresp.http.Content-Type = "text/html; charset=utf-8";
  set beresp.http.Retry-After = "5";
  synthetic( {"<!DOCTYPE html>
<html>
  <head>
    <title>} + beresp.status + " " + beresp.reason + {"</title>
  </head>
  <body>
    <h1>Error "} + beresp.status + " " + beresp.reason + {"</h1>
    <p>} + beresp.reason + {"</p>
    <h3>Guru Meditation:</h3>
    <p>XID: "} + bereq.xid + {"</p>
    <hr>
    <p>Varnish cache server</p>
  </body>
</html>
"} );
  return (deliver);
}

```

```
#####  
# Housekeeping  
  
sub vcl_init {  
    return (ok);  
}  
  
sub vcl_fini {  
    return (ok);  
}
```

VCL <https://riptutorial.com/zh-CN/varnish/topic/5001/vcl>

3: vmods

vmodvmod。 CentOS 67IngvarCOPRVarnish Software <https://copr.fedorainfracloud.org/coprs/ingvar/varnish51/>

Examples

vmod

vmodVarnish Cache。 Varnish。

autotools

```
sudo apt-get install libvarnishapi-dev || sudo yum install varnish-libs-devel
./bootstrap # If running from git.
./configure
make
make check # optional
sudo make install
```

vmods <https://riptutorial.com/zh-CN/varnish/topic/9669/vmods>

4:

varnishstat Varnish ◦ ◦ varnishstat -lgrep -able ◦

varnishtop varnishlog ◦

Examples

-

Varnish ◦

- sess_conn - ◦
- client_req - ◦
- sess_dropped - ◦

sess_conn client_req - ◦

sess_dropped ◦ thread_pool_max ◦

```
varnishstat -1 | grep "sess_conn\\|client_req \\|sess_dropped"
MAIN.sess_conn          62449574          3.38 Sessions accepted
MAIN.client_req         184697229         9.99 Good client requests received
MAIN.sess_dropped       0                 0.00 Sessions dropped for thread
```

◦

Varnish

- ◦ hitmiss ◦
- Hitpass ◦

miss hitpass ◦ hit ◦

- cache_hit - ◦
- cache_miss - ◦
- cache_hitpass - hitpasses ◦

```
varnishstat -1 | grep "cache_hit \\|cache_miss \\|cache_hitpass"
MAIN.cache_hit          99032838          5.36 Cache hits
MAIN.cache_hitpass      0                 0.00 Cache hits for pass
MAIN.cache_miss         42484195          2.30 Cache misses
```

```
cache_hit / (cache_hit + cache_miss)
```

0.770 ◦ ◦ 70 ◦ vcl ◦ ◦

“”。

- n_expired - 。
- n_lru_nuked - nuked 。

```
varnishstat -1 | grep "n_expired\\|n_lru_nuked"
MAIN.n_expired          42220159      .    Number of expired objects
MAIN.n_lru_nuked       264005        .    Number of LRU nuked objects
```

n_lru_nuked 。

n_expired 。

Varnish Cache OS。

- threads - 。
- threads_created - 。
- threads_failed - Varnish。
- threads_limited - Varnish。
- thread_queue_len - 。
- sess_queued - 。

```
varnishstat -1 | grep "threads\\|thread_queue_len\\|sess_queued"
MAIN.threads           100           .    Total number of threads
MAIN.threads_limited   1             0.00 Threads hit max
MAIN.threads_created   3715          0.00 Threads created
MAIN.threads_destroyed 3615          0.00 Threads destroyed
MAIN.threads_failed    0             0.00 Thread creation failed
MAIN.thread_queue_len  0             .    Length of session queue
MAIN.sess_queued       2505          0.00 Sessions queued for thread
```

thread_queue_len 0 Varnish 。

threads_failed threads_limited thread_pool_max 。

Varnish。

- backend_busy - http 5xx VCL Varnish。
- backend_fail - Varnish TCP 。
- backend_unhealthy - Varnish“ping”HTTP 200。

```
varnishstat -1 | grep "backend_"
MAIN.backend_conn      86913481      4.70 Backend conn. success
MAIN.backend_unhealthy 0             0.00 Backend conn. not attempted
MAIN.backend_busy      0             0.00 Backend conn. too many
MAIN.backend_fail      7             0.00 Backend conn. failures
MAIN.backend_reuse     0             0.00 Backend conn. reuses
MAIN.backend_toolate   0             0.00 Backend conn. was closed
MAIN.backend_recycle   0             0.00 Backend conn. recycles
MAIN.backend_retry     0             0.00 Backend conn. retry
```

<https://riptutorial.com/zh-CN/varnish/topic/9072/>

S. No		Contributors
1		alejdg , Community , Daniel V.
2	VCL	Daniel V. , fgsch , Redithion
3	vmods	Daniel V.
4		Jensd