Varnish HTTP cache

CONCEPTS

Varnish is a caching reverse proxy: clients hit varnishd, which answers from cache or fetches from the backend. Behaviour is set in VCL, changed live with varnishadm.

Client HTTP request
↓ vcl_recv
Cache hit: serve object
↓ miss
Backend fetch, vcl_backend_response
↓ store
Client vcl_deliver

RUN VARNISHD

varnishd -a :<port> **-f** <file.vcl>
Listen on a port with a VCL file

varnishd -T localhost:6082 -S <secretfile>
Open the management CLI

varnishd -s malloc,<size>
Cache in memory, e.g. malloc,1G

VARNISHADM

varnishadm <command>
Run one CLI command

varnishadm status
Is the cache process running?

VCL MANAGEMENT

varnishadm vcl.load <name> <file>
Compile and load a VCL

varnishadm vcl.use <name>
Make a loaded VCL active

varnishadm vcl.list
List VCLs, state and temperature

varnishadm vcl.show <name>
Print the VCL source

varnishadm vcl.discard <name>
Unload a VCL

BANS AND PURGING

varnishadm ban req.url ~ "<regex>**"**
Invalidate by URL pattern

varnishadm ban.list
Show active bans

BACKENDS AND HEALTH

varnishadm backend.list
List backends and health

varnishadm backend.list -p
Include probe details

varnishadm backend.set_health <backend> **sick**
Take a backend out of rotation

VARNISHSTAT

varnishstat
Live counters (curses)

varnishstat -1
Print all counters once

varnishstat -1 -f MAIN.cache_hit -f MAIN.cache_miss
Select counters (globs)

VARNISHLOG

varnishlog -g request
Group by client request tree

varnishlog -q '<VSL query>**'**
Filter transactions

varnishlog -q 'RespStatus ⩾ 500'
Only server errors

VCL ESSENTIALS

backend default { .host = "<host>**"; .port = "**<port>**"; }**
Declare a backend

sub vcl_recv { ... }
Request arrives; choose hash, pass, purge

sub vcl_backend_response { set beresp.ttl = 5m; }
Set cache lifetime