LB configuration file /opt/polaris/etc/polaris-lb.yaml consists of two sections:

* pools:
  + Defines pools of backend server end-points, health monitor, load balancing method and other associated information.
* globalnames:
  + Defines FQDNs to perform GSLB against

Example:

pools:

www-example:

monitor: http

monitor\_params:

use\_ssl: true

hostname: www.example.com

url\_path: /healthcheck?check\_all=true

lb\_method: twrr

fallback: any

max\_addrs\_returned: 2

members:

- ip: 10.1.1.2

monitor\_ip: 172.16.1.3

name: www1-dc1

weight: 1

- ip: 10.1.1.3

name: www2-dc1

weight: 1

- ip: 192.168.1.2

name: www3-dc2

weight: 1

- ip: 192.168.1.3

name: www4-dc2

weight: 1

globalnames:

www.example.com:

pool: www-example

ttl: 1

# pools

Parameters:

* monitor, monitor\_params: health monitor configuration, see below.
* lb\_method - load balancing method to use
  + wrr: weighted round-robin
  + twrr: topology weighted round-robin
  + fogroup: failover group, when this method is assigned, IP address of the first configured member that is healthy is handed out continuously, unless the member becomes unhealthy, then, the next healthy member IP is handed out etc.
* fallback - resolution behavior when all members of the pool are DOWN
  + any: default, perform distribution amongst all the configured members with non-0 weight(ignore health status)
  + refuse: refuse all queries Note: fallback is set to "any" with all member weights set to 0 will result in a NOERROR response with no answer section data.
* max\_addrs\_returned - maximum number of A records to return in response, large responses will go over TCP min: 1, max: 1024, default: 1

### members

Parameters:

* name: name or description of the member, informational only
* weight: weight of the server, min: 0(server is disabled), max: 10
* monitor\_ip: optional IP address to issue health checks against, allows to configure health endpoints on a server that is different from the server providing the service itself

## Monitors

Common optional params:

* interval: how often to perform a probe, seconds, min: 1, max: 3600
* timeout: timeout, seconds, min: 0.1, max: 10
* retries: how many times to retry before declaring the member DOWN, min: 0, max: 5

#### http

Perform HTTP(S) GET, succeeds if response HTTP status is 200(or one of expected\_codes is specified)

monitor\_params:

* optional:
  + interval: default 10
  + timeout: default 5
  + retries: default 2
  + use\_ssl: whether to use SSL, default is false
  + url\_path: url path to request, appended after the member's IP address, default is /.
  + hostname: hostname to supply in HTTP Host: header, when using SSL this will also be supplied in SNI, default is none.
  + port: port number to use, integer between 1 and 65535, if value is not provided, port 80 will be used with use\_ssl set to false, port 443 will be used with use\_ssl set to true
  + expected\_codes: an array of HTTP codes to match in a response

Example:

pools:

example1:

monitor: http

monitor\_params:

use\_ssl: true

hostname: www.example.com

url\_path: /health.html

interval: 30

expected\_codes:

- 201

- 203

#### tcp

Perform a TCP connect. (Optionally: send text, read response, match a reg exp pattern).

monitor\_params:

* optional:
  + interval: default 10
  + timeout: default 5
  + retries: default 2
  + send\_string: a string to send after connecting a socket
  + match\_re: a reg exp to match in response (the reg exp compiles with re.IGNORECASE flag set)
* required:
  + port: port number, integer between 1 and 65535

Example:

pools:

example1:

monitor: tcp

monitor\_params:

port: 2222

timeout: 0.1

match\_re: up

#### forced

Forces a member to be either UP or DOWN effectively disabling the health checking.

monitor\_params:

* optional:
  + interval: default 10
  + timeout: default 5
  + retries: default 2
  + status: a string, one of "up" or "down", default is "up"

Example:

pools:

example1:

monitor: forced

# globalnames

Parameters:

* pool: pool name to associate with this globalname
* ttl: Time To Live value in DNS responses