FRiCKLE / ngx_supervisord

★ Star 65 Fork 3

nginx module providing API to communicate with supervisord and manage (start/stop) backends on-demand.
http://labs.frickle.com/nginx_ngx_supervisord/

Code
Issues 1
Pull Requests 0
Pulse
Graphs

7 commits **1** branch **0** releases **1** contributor

branch: **master** ngx_supervisord / +

Update information about compatibility with different nginx versions.

PiotrSikora authored on Jun 24, 2010 latest commit 2325fd0a89

patches	ngx_supervisord-1.4	4 years ago
CHANGES	ngx_supervisord-1.4	4 years ago
LICENSE	ngx_supervisord-1.3	5 years ago
README	Update information about compatibility with different nginx versions.	4 years ago
config	Define nginx build-time variable NGX_SUPERVISORD_MODULE.	4 years ago
ngx_supervisord.c	ngx_supervisord-1.3	5 years ago
ngx_supervisord.h	ngx_supervisord-1.0	5 years ago

HTTPS clone URL

https://github.com/l

You can clone with HTTPS or Subversion.

Clone in Desktop

Download ZIP

README

```
ABOUT:
------
ngx_supervisord is a module that provides API to communicate with
supervisord daemon (http://supervisord.org).

As a "side effect", it also provides a way for dynamically
taking backend servers out of rotation (configuration example #2).

Interface is described in ngx_supervisord.h. For example of implementation
please check patches/ngx_http_upstream_fair_module.patch.


SPONSORS:
---------
ngx_supervisord-1.0 was fully funded by megiteam.pl.


REQUIREMENTS:
-------------
* nginx >= 0.7.63 or >= 0.8.7,
* ngx_http_upstream_init_busy patch by Ryan Lienhart Dahl (included in patches).
* ngx_supervisord-aware module(s).


INCLUDED PATCHES:
-----------------
ngx_http_upstream_fair_module.patch:
Patch against nginx-upstream-fair load balancer by Grzegorz Nosek
(http://github.com/gnosek/nginx-upstream-fair), which adds capabilities to:
* start first backend server,
* start/stop backend servers depending on the load,
* set number of minimum running backend servers.

ngx_http_upstream_init_busy-0.8.0.patch:
Patch (by Ryan Lienhart Dahl) against nginx versions 0.7.65+ and 0.8.0-0.8.16
which adds ability to stop/resume request processing.
```

```
ngx_http_upstream_init_busy-0.8.17.patch:
Same as above, for versions 0.8.17+ (last tested version is 0.8.42).

ngx_http_upstream_round_robin.patch:
Patch against bulit-in load balancer, which adds ability to control
status of the backend servers ("alive" / "down") on-the-fly without
modifications in nginx configuration.


INSTALLATION (with patched nginx-upstream-fair, versions ommited):
------------------------------------------------------------------
// unpack releases
$ tar -zxf nginx.tar.gz
$ tar -zxf ngx_supervisord.tar.gz
$ tar -zxf gnosek-nginx-upstream-fair.tar.gz

// patch gnosek-nginx-upstream-fair
$ cp ngx_supervisord/patches/ngx_http_upstream_fair_module.patch
     gnosek-nginx-upstream-fair/
$ cd gnosek-nginx-upstream-fair; patch -p0 < ngx_http_upstream_fair_module.patch

// patch nginx
$ cp ngx_supervisord/patches/ngx_http_upstream_init_busy.patch nginx/
$ cd nginx
$ patch -p0 < ngx_http_upstream_init_busy.patch

// build
$ ./configure --add-module=/path/to/ngx_supervisord
  --add-module=/path/to/gnosek-nginx-upstream-fair
$ make && make install


CONFIGURATION NOTES:
--------------------
Following applies only to versions older than nginx-0.8.28:
Since ngx_supervisord-1.2 you cannot set "worker_connections" to 512,
it can be set to either lower or higher number, just not equal to 512.
For details please check CHANGES log and/or source code.


CONFIGURATION DIRECTIVES:
-------------------------
  supervisord path [user:pass] (context: upstream)
  ------------------------------------------------
  Path to supervisord's listening socket, path can be:
  * IP:port (127.0.0.1:8000)
  * UNIX socket path (unix:/path/to/supervisord.sock)
  * none.

  When supervisord is explicity set to "none" then module will execute
  every command instantly without even talking to supervisord daemon,
  but it will still notice registered modules about backend servers
  status changes, current load, etc.
  This basically enables all ngx_supervisord features without
  the need to run supervisord daemon.

  NOTE:
  When supervisord is set to "none" it won't try to auto-start first
  backend server when all backend servers are considered down.

  supervisord_name name (context: upstream)
  -----------------------------------------
  Use name instead of upstream{}'s name when communicating with supervisord.

  supervisord_inherit_backend_status (context: upstream)
  ------------------------------------------------------
  Use configured backend statuses (server is considered alive unless it
  is followed by "down" in nginx.conf). When this directive isn't used
  supervisord assumes that all servers are down and it will try to start
  them when needed.

  supervisord_start upstream_name (context: location)
  supervisord_stop  upstream_name (context: location)
  ---------------------------------------------------
  Executes "start" / "stop" command on given upstream.

  Using one of above handlers creates valid URIs:
  /location/0, /location/1, ..., /location/n-1,
  /location/any
  for upstreams with n backends.

  NOTE:
  Successful response means that request was processed by ngx_supervisord,
  not that the command was already executed by supervisord daemon.


EXAMPLE CONFIGURATION #1:
-------------------------
upstream backend {
    server 127.0.0.1:8000;
    server 127.0.0.1:8001;
    supervisord 127.0.0.1:9001 admin:super;
    fair;
}

server {
    location / {
        proxy_pass http://backend;
    }
}

With such configuration, ngx_supervisord will be starting/stopping
[program:backend0] (which should be listening on 127.0.0.1:8000)
and [program:backend1] (which should be listening on 127.0.0.1:8001)
from supervisord's configuration.


EXAMPLE CONFIGURATION #2:
-------------------------
upstream backend {
    server 127.0.0.1:8000;
    server 127.0.0.1:8001;
    server 127.0.0.1:8002 down;
    supervisord none;
    supervisord_inherit_backend_status;
}

server {
    location / {
        proxy_pass http://backend;
    }

    location /_start/ {
        allow               127.0.0.1;
        deny                all;
        supervisord_start   backend;
    }

    location /_stop/ {
        allow               127.0.0.1;
        deny                all;
        supervisord_stop    backend;
    }
}

With such configuration, ngx_supervisord will assume that 2 out of 3
backends are alive and it will never talk to supervisord daemon.

Calling "http://localhost/_start/2" will change status of
"127.0.0.1:8002" backend to "alive" and ngx_supervisord will notice
all ngx_supervisord-aware load balancers about this change.


CREDITS:
--------
* Magda Zarych (megiteam.pl),
* Grzegorz Nosek (megiteam.pl),
* Ryan Lienhart Dahl.
```