SCA NGINX Configuration

sca-nginx01 and sca-nginx02 are a pair of servers which used to reverse proxy or directly serve all glast-ground, srs, freehep.org, and lsstdesc.org domains, among others.

As of 12/1/2021, these machines pass through all traffic to sca-resty01.

Their configuration is on ~srs/sca-nginx.

Operating NGINX

To restart the nginx configuration on sca-resty01: **systemctl reload openresty**

nginx is installed on both sca-nginx01 and sca-nginx02. You can stop, start, restart, and reload the service at any time. A reload keeps the server up but refreshes it's configuration based on configuration file changes.

To start/stop/reload:

sudo /sbin/service nginx16-nginx [start|stop|reload].

Care should be taken care to verify your configuration before performing a reload.

Configuration

Hosts

sca-www: Domain name tied to a floating IP 134.79.129.86; this is a floating IP address.

sca-nginx01: Primary server, this host is a virtual machine

sca-nginx02: Failover server, this host is a virtual machine

scalnx12-vmm: Hypervisor for the sca-nginx01 virtual machine.

scalnx13-vmm: Hypervisor for the sca-nginx02 virtual machine.

sca-nginx03: Virtual Machine (2 cores, 2GB) running nginx used for PHP (forum.linearcollider.org) and miscellaneous web applications.

Operation

nginx is running on the two virtual machines, sca-nginx01 and sca-nginx02. Both physical machines are in the high-availability rack.

The web servers (sca-nginx01/sca-nginx02) operate in a master-failover configuration with no load balancer. Failover management is managed by keepalived, which runs on both web servers, and utilizes a floating IP address.

Software

keepalived: We use RHEL6’s “loadbalancer” child channel to install keepalived. nginx16: We use RHEL6’s Software Collections Library (SCL) channel to install a nginx16. taylor: Taylor is used to manage the hosts. See the Taylor section for more info.

Domains

As these machines are a pass-through for most domains to sca-resty01, please see the github repo for information on the domains handled.

https://github.com/slaclab/sca-resty/tree/main/rootfs/etc/nginx/conf.d

NGINX

nginx, as ran on sca-nginx01 and sca-nginx02, will operate purely in “reverse proxy” mode, forwarding requests and responses to/from an ensemble of tomcat servers. This is typically used for load balancing.

For example, a request to: glast-ground.slac.stanford.edu/DataCatalog will forward to a tomcat server running on glast-tomcat08.slac.stanford.edu on port 8080, which is behind the fiewall. A request to glast-ground.slac.stanford.edu/Pipeline-II, instead forwards to glast-tomcat09.slac.stanford.edu, also on port 8080. This is primarily to allow us to manage load on a per-application basis.

Installation

nginx is installed on rhel6 by adding the software collections library, then installing the nginx-nginx16 package. When RHEL6 actually installs it, it locates the install in it's own isolated root folder under /opt/rh/nginx16/root. So, for example, if you read documentation which says "edit /etc/nginx.conf", you need to actually edit /opt/rh/nginx16/root/etc/nginx.conf.

Configuration

This section isn't particularly relevant for sca-nginx01 or sca-nginx02, as they are now merely passthroughs for sca-resty01. It is kept for posterity. 12/1/2021

nginx has strong support for including file(s) at most any level of configuration. We can prevent long configurations, and thus, mistakes, by leveraging this feature. In addition to this, we can separate the configuration out for each domain into it's own file. For this, nginx is currently configured to import all *.conf files from the /etc/conf.d directory (/opt/rh/nginx16/root/etc/conf.d). Organize each domain as it's own .conf file, e.g. glast-ground.slac.stanford.edu.conf.

Inside each .conf file, we can tell nginx where to route requests based on a URL pattern.

A snippet of the glast-ground.slac.stanford.edu.conf file looks like this, for example:

```
server {
    server_name glast-ground.slac.stanford.edu;
    listen 80;
    listen 443 ssl;
    include /opt/rh/nginx16/root/etc/nginx/conf.d/ssl/glast-ground.inc;

    # host http://glast-tomcat01.slac.stanford.edu:8080
    location / {
        proxy_pass http://glast-tomcat01.slac.stanford.edu:8080;
        include /opt/rh/nginx16/root/etc/nginx/conf.d/default_reverse_proxy.inc;
    }

    location /Commons {
        proxy_pass http://glast-tomcat01.slac.stanford.edu:8080;
        include /opt/rh/nginx16/root/etc/nginx/conf.d/default_reverse_proxy.inc;
    }


    location /GroupManager {
        proxy_pass http://glast-tomcat01.slac.stanford.edu:8080;
        include /opt/rh/nginx16/root/etc/nginx/conf.d/default_reverse_proxy.inc;
    }
    ...
    ...
    # host http://glast-tomcat08.slac.stanford.edu:8080
    location /DataCatalog {
        proxy_pass http://glast-tomcat08.slac.stanford.edu:8080;
        include /opt/rh/nginx16/root/etc/nginx/conf.d/default_reverse_proxy.inc;
    }
    ...
    ...

}
```

Each location entry has a proxy_pass directive which tells nginx which server the request will be routed too, and, for simplicity, it also includes a snippet from default_reverse_proxy.inc. This snippet just sets some headers which can be used by backend applications to determine information about the request, such as the protocol of the request that came into nginx. If, for example, the protocol was https, a backend tomcat server could see that the original request was secure and modify the servlet container request accordingly to reflect that.

```
            proxy_set_header        X-Real-IP $remote_addr;
            proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header        X-Forwarded-Proto $scheme;
            proxy_set_header        Host $http_host;
```

It's best to organize applications conceptually by backend server (glast-tomcat01, glast-tomcat02, ...), then by URL pattern (/, /Commons, /GroupManager).

There's an additional include at the top of the site's file for SSL settings. When we get an SSL certificate for our other sites, the configuration for each site should be a .inc file under the /etc/conf.d/ssl which we can include in sites to be secured by SSL.

SSL/TLS

sca-resty servers now handle all certificates with dehydrated and let's encrypt. There is an update script on the sca-nginx servers to pull in the most recent certificates.

Tomcat

In the nginx section, a snippet for the reverse proxy configuration of nginx was demonstrated. In order for SSL termination to work, we use those nginx-set headers and tell tomcat to interpret them through a valve, so we need to add a valve to server.xml. The valve which does that is included in Tomcat, and it is the RemoteIpValve.

The configuration for that valve looks like this:

```
          <Valve className="org.apache.catalina.valves.RemoteIpValve"
               internalProxies="134.79.129.91|134.79.129.92"
               remoteIpHeader="x-forwarded-for"
               remoteIpProxiesHeader="x-forwarded-by"
               protocolHeader="x-forwarded-proto" />
```

The internalProxies attribute is important; By default, the RemoteIpValve only looks at localhost/internal network IP addresses when it is interpreting those headers. Tomcat will not bother to read the headers from our sca-nginx01/02 servers and will not bother upgrading the requests. The IP addresses in this configuration correspond to sca-nginx01 and sca-nginx02 respectively. Notice that the headers in this valve configuration are the same headers as the nginx snippet.

Taylor

On sca-nginx* machines, /etc/taylor.opts is configured as follows:

```
%%include opts/scalnx-v keepalived
```

When including opts/scalnx-v, we also inherit the following taylor configuration:

```
automounter=autofs
%%if ($ENV{HOSTNAME} ne 'scalnx-v03')
#limit_login=u-scalnx
%%endif
network_device=eth0
monitoring=nagios,ganglia
iptables
sudo_workgroups=scalnx-vmm
```

keepalived

keepalived is to be configured on both machines.

The following is roughly the configuration for sca-nginx01. The configuration for sca-nginx02, which will be the failover machine, will be nearly identical, but the priority on the vrrp_instance **MUST BE LOWER** than the priority on the master. The password will be different.

The configuration is located in /etc/keepalived/keepalived.conf

```
global_defs {
   notification_email {          # This should email you when there's a failover, but it's not working right now
     bvan@slac.stanford.edu
   }
   notification_email_from bvan@slac.stanford.edu
   smtp_server smtp.slac.stanford.edu
   smtp_connect_timeout 30
   router_id SCA_WWW             # This is the virtual router ID.
}

vrrp_script chk_nginx {           # The check script
      script "killall -0 nginx"   # Could use curl instead or call out to custom script.
      interval .2                 # execute every .2 seconds
      weight 2                    # This is subtracted from priority below. Highest point count = master
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0             # The virtual IP address is assigned to eth0
    virtual_router_id 85       # This should be unique per subnet. 85 is fine. If SCS uses keepalived,
                               #   we may need to coordinate/register this with them.
    priority 101               # This primary machine should have the highest priority,
                               #   the failover should be at least one point lower.

    advert_int 1               # Both of these are for authentication, but it's not really necessary
    authentication {           #   unless we can't trust the computers on the same subnet.
        auth_type PASS
        auth_pass 1111
    }

    virtual_ipaddress {
        134.79.129.86          # This is the sca-www IP address
    }

    track_script {
         chk_nginx             # Tell keepalived there is a script it should execute (Defined above)
    }

   unicast_src_ip 134.79.129.91   # Unicast specific option, this is the IP of the interface keepalived listens
on
   unicast_peer {                 # Unicast specific option, this is the IP of the peer instance
     134.79.129.92
   }
}
```

sca-nginx03

This server runs forum.linearcollider.org and serves glast-isoc.

Installation

```
sudo yum install nginx16
sudo yum install php54 php54-php-fpm php54-php-mysqlnd
sudo vi /opt/rh/php54/root/etc/php.ini # add or change: cgi.fix_pathinfo=0
```

Background

We have several webservers behind the firewall that serve up scientific applications. We’ve been using a few IIS servers to act as gateway to those servers and provide services such as SSL termination.

Our IIS servers are ancient. Currently, glast-win01 and glast-win02 operate behind a load balancer, and it serves up glast-ground only. web08 is not behind a load balancer, and has no redundancy. The OSs on at least the glast-win01/glast-win02 machines is EOL and unsupported, and as such glast-win01 and glast-win02 are operating on a security exception with the assumption that we will finish the nginx migration soon. We are replacing glast-win01, glast-win02, and web08 all with the proposed service.

As for the choice of nginx, nginx as it has much better performance characteristics in the application of an HTTP reverse proxy, which is necessitated by our migration towards REST APIs which serve up scientific data for many experiments.