Maintenance page

Sometimes hosts need to be shut down for updates, what we would like is a webpage that will be shown instead. This page could give some explanations of why the site is down.

Plan is to have a server that can handle and impersonate the web traffic for different sites. This single server will be the backup server and only serve a single page that explains the outage and who to contact. There are 2 pieces needed for this to work. The site configuration files needed for the server to respond to requests for a specific server (see below in Site Configs). The second piece needed is a method to switch DNS entries from one server to another.

DNS switching

For this to work we need to change the DNS entry to point to the maintenance server. There are different methods to do this. Two methods we have experience with is to ask the neteng team to change the DNS manually or have loadbalancing service that checks to see if the server is up.

Manual DNS switching

Since most servers are set to have a timeout of 24 hours, which requires the neteng team to be notified at least 24 hours before hand. The steps are in general:

1. At least 24 hours prior to maintenance ask neteng file to switch TTL to 1 minute
2. When maintenance is about to start ask neteng to change the DNS to 141.142.210.188
3. make sure things work using curl https://server/ (for example curl https://browndog.ncsa.illinois.edu)
4. Do your work
5. When maintenance is done ask neteng to change the DNS back to the original IP address
6. (Optional) Ask neteng to change the TTL back to 24 hours.

Automatic DNS switching

We can leverage off a load balancing service that will automatically check to see if a server is up, and will fail back to the maintenance page if the server is not reachable.

One of such examples is Global Server Load Balancing hosted by Illinois. GSLB is a load balancing system that will respond to DNS queries and returns an IP address associated with the name. GSLB will continuously check the main IP address to see if this up and responding. Once it detects no response it will switch to the backup IP address. Once the main IP address is back up, GSLB will switch back to the regular IP address.

A discussion has started with the neteng team to see how we can support a system like this at NCSA. Examples being considered is a nginx high availability, keepalived, etc.

Site Configs

All files will be hosted in /home/maintenance/<server>, the following is what is recommended:

The nginx configuration file in /home/maintenance/<server>/nginx.conf and a symbolic link (ln -s /home/maintenance/<server>/nginx.conf /etc/nginx/sites-enabled/<server>)

nginx.conf

```
server {
  listen 80;
  server_name browndog.ncsa.illinois.edu;

  client_max_body_size 0;
  server_tokens off;

  root /home/maintenance/browndog/html;

  # create 503 Service Unavailable
  location / {
    return 503;
  }

  error_page 503 @maintenance;
  location @maintenance {
    rewrite ^(.*)$ /maintenance.html break;
  }
}

server {
  listen 443;
  server_name browndog.ncsa.illinois.edu

  client_max_body_size 0;
  server_tokens off;

  proxy_read_timeout 300; # answer from server, 5 min
  proxy_send_timeout 300; # chunks to server, 5 min

  ssl on;
  ssl_session_timeout 5m;

  ssl_certificate /home/maintenance/browndog/ssl/nginx.crt;
  ssl_certificate_key /home/maintenance/browndog/ssl/nginx.key;

  #ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
  ssl_protocols TLSv1.2;
  ssl_prefer_server_ciphers on;
  #ssl_ciphers "EECDH+ECDSA+AESGCM EECDH+aRSA+AESGCM EECDH+ECDSA+SHA384 EECDH+ECDSA+SHA256 EECDH+aRSA+SHA384
EECDH+aRSA+SHA256 EECDH+aRSA+RC4 EECDH EDH+aRSA RC4 !aNULL !eNULL !LOW !3DES !MD5 !EXP !PSK !SRP !DSS !RC4";
  ssl_ciphers "ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-
GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA";

  root /home/maintenance/browndog/html;

  # create 503 Service Unavailable
  location / {
    return 503;
  }

  error_page 503 @maintenance;
  location @maintenance {
    rewrite ^(.*)$ /maintenance.html break;
  }
}
```

A single file called maintenance.html in /home/maintenance/<server>/html/maintenance.html

maintenance.html

```
<html>
  <head>
    <title>Maintenance</title>
  </head>
  <body>
    <h1>Down for maintenance</h1>
    <p>The service you are looking for is currently down for maintenance. Please try again soon.</p>
    <p>A list of known outages can be found at <a href="http://status.ncsa.illinois.edu/">NCSA Status Page</a><
/p>
    <h1>Please pardon the dust!</h1>
    <p>If you reach this page and want to let us know about your experience during the outage or have any
questions about why you received this page, please contact us at <a href="mailto:opensource+maintenance@ncsa.
illinois.edu">browndog maintenance</a>.</p>
  </body>
</html>
```

And the SSL certificate needed to return https calls, place these in /home/maintenance/<server>/ssl/nginx.key and /home/maintenance/<server>/ssl/nginx.crt

Once the site is configured you can test it using:

curl example

```
curl --insecure --verbose --header 'Host: browndog.ncsa.illinois.edu' https://141.142.210.188/clowder/
*   Trying 141.142.210.188...
* Connected to 141.142.210.188 (141.142.210.188) port 443 (#0)
* found 148 certificates in /etc/ssl/certs/ca-certificates.crt
* found 594 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.2 / ECDHE_RSA_AES_256_GCM_SHA384
*          server certificate verification SKIPPED
*          server certificate status verification SKIPPED
*          common name: browndog.ncsa.illinois.edu (does not match '141.142.210.188')
*          server certificate expiration date OK
*          server certificate activation date OK
*          certificate public key: RSA
*          certificate version: #3
*          subject: C=US,postalCode=61801,ST=IL,L=Urbana,street=1205 W. Clark St,O=University of Illinois,
OU=NCSA,CN=browndog.ncsa.illinois.edu
*          start date: Fri, 16 Feb 2018 00:00:00 GMT
*          expire date: Mon, 15 Feb 2021 23:59:59 GMT
*          issuer: C=US,ST=MI,L=Ann Arbor,O=Internet2,OU=InCommon,CN=InCommon RSA Server CA
*          compression: NULL
* ALPN, server accepted to use http/1.1
> GET /clowder/ HTTP/1.1
> Host: browndog.ncsa.illinois.edu
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 503 Service Temporarily Unavailable
< Server: nginx
< Date: Tue, 26 Feb 2019 16:27:08 GMT
< Content-Type: text/html
< Content-Length: 630
< Connection: keep-alive
< ETag: "5c75661f-276"
<
<html>
  <head>
    <title>Maintenance</title>
  </head>
  <body>
    <h1>Down for maintenance</h1>
    <p>The service you are looking for is currently down for maintenance. Please try again soon.</p>
    <p>A list of known outages can be found at <a href="http://status.ncsa.illinois.edu/">NCSA Status Page</a><
/p>
    <h1>Please pardon the dust!</h1>
    <p>If you reach this page and want to let us know about your experience during the outage or have any
questions about why you received this page, please contact us at <a href="mailto:opensource+maintenance@ncsa.
illinois.edu">browndog maintenance</a>.</p>
  </body>
</html>
* Connection #0 to host 141.142.210.188 left intact
```

This example shows that it will pretend the browndog webserver in this case, and returns the actual webpage. The curl command requires the --insecure flag since the IP address does not match the actual hostname. The --header 'Host: browndog.ncsa.illinois.edu' will allow the curl command to pretend to connect to a specific host and will use the right configuration section for nginx (the servername in the nginx.conf).