

GNU/Linux Administration - Support #736

Install A Reverse Proxy Package Cache With Nginx On Arch

01/30/2016 01:13 PM - Daniel Curtis

Status:	Closed	Start date:	01/31/2016
Priority:	Normal	Due date:	
Assignee:	Daniel Curtis	% Done:	100%
Category:	Package Management	Estimated time:	1.00 hour
Target version:	Arch Linux	Spent time:	4.00 hours

Description

This is a guide on how I setup a dynamic reverse proxy package cache using Nginx on Arch Linux.

WARNING: This method has a limitation. You must use mirrors that use the same relative path to package files and you must configure your cache to use that same path. In this example, we are using mirrors that use the relative path `/archlinux/$repo/os/$arch` and our cache's Server setting in `mirrorlist` is configured similarly.

Prepare the Environment

- Make sure everything is up to date using the following command:

```
pacman -Syu
```

Install Nginx

- Install Nginx

```
pacman -S nginx
```

- Start and enable nginx at boot:

```
systemctl enable nginx  
systemctl start nginx
```

- Create a configuration directory to make managing individual server blocks easier

```
mkdir /etc/nginx/conf.d
```

- Edit the main nginx config file:

```
vi /etc/nginx/nginx.conf
```

- And strip down the config file and add the include statement at the end to make it easier to handle various server blocks:

```
worker_processes 1;  
error_log /var/log/nginx-error.log;  
  
events {  
    worker_connections 1024;  
}  
  
http {
```

```

include      mime.types;
default_type application/octet-stream;
sendfile     on;
keepalive_timeout 65;

# nginx may need to resolve domain names at run time. ipv6 may also break if not enabled on host.
resolver 208.67.222.222 208.67.220.220 ipv6=off;

include /etc/nginx/conf.d/*.conf;
}

```

Vanilla Arch Cache Config

- Create the directory for the cache and adjust the permissions so nginx can write files to it:

```

mkdir /srv/http/pacmirror
chown http:http /srv/http/pacmirror

```

- Configure nginx as our dynamic cache:

```

vi /etc/nginx/conf.d/pacmirror.example.com.conf

```

- And add the following:

```

server {
    listen      80;
    server_name pacmirror.example.com;
    root        /srv/http/pacmirror;
    autoindex   on;

    # Requests for actual packages should be served directly from cache if available.
    # If not available, retrieve and save the package from an upstream mirror.
    location / {
        try_files $uri @pkg_mirror;
    }

    # Retrieve package from upstream mirrors and cache for future requests
    location @pkg_mirror {
        proxy_store      on;
        proxy_redirect    off;
        proxy_store_access user:rw group:rw all:r;
        proxy_cache_use_stale error timeout http_500 http_502 http_503 http_504;
        proxy_next_upstream error timeout http_404;
        proxy_pass         http://mirrors$request_uri;
    }
}

# Upstream Arch Linux Mirrors
upstream mirrors {
    server localhost:8001;
    server localhost:8002 backup;
    server localhost:8003 backup;
}

# Arch Mirror 1 Proxy Configuration
server {
    listen      8001;
    server_name localhost;

    location / {

```

```

        proxy_pass      http://mirror.us.leaseweb.net$request_uri;
        proxy_set_header Host mirror.us.leaseweb.net;
    }
}

# Arch Mirror 2 Proxy Configuration
server {
    listen      8002;
    server_name localhost;

    location / {
        proxy_pass      http://mirror.rit.edu$request_uri;
        proxy_set_header Host mirror.rit.edu;
    }
}

# Arch Mirror 3 Proxy Configuration
server {
    listen      8003;
    server_name localhost;

    location / {
        proxy_pass      http://lug.mtu.edu$request_uri;
        proxy_set_header Host lug.mtu.edu;
    }
}

```

- Restart nginx:

```
systemctl restart nginx
```

Update System Mirrorlist

- Edit the mirrorlist file:

```
vi /etc/pacman.d/mirrorlist
```

- Add the following line to use this new cache

```
Server = http://pacmirror.example.com/archlinux/$repo/os/$arch
```

Arch Linux ARM Cache Config

- Create the directory for the cache and adjust the permissions so nginx can write files to it:

```
mkdir /srv/http/pacmirror
chown http:http /srv/http/pacmirror
```

- Configure nginx as our dynamic cache:

```
vi /etc/nginx/conf.d/almarm.example.com.conf
```

- And add the following:

```
server {
    listen      80;
```

```

server_name alarm.example.com;
root          /srv/http/pacmirror;
autoindex    on;

# Requests for actual packages should be served directly from cache if available.
# If not available, retrieve and save the package from an upstream mirror.
location / {
    try_files $uri @pkg_mirror;
}

# Retrieve package from upstream mirrors and cache for future requests
location @pkg_mirror {
    proxy_store    on;
    proxy_redirect off;
    proxy_store_access user:rw group:rw all:r;
    proxy_cache_use_stale error timeout http_500 http_502 http_503 http_504;
    proxy_next_upstream error timeout http_404;
    proxy_pass     http://mirrors$request_uri;
}
}

# Upstream Arch Linux Mirrors
upstream mirrors {
    server localhost:8001;
    server localhost:8002 backup;
    server localhost:8003 backup;
}

# Arch Mirror 1 Proxy Configuration
server {
    listen      8001;
    server_name localhost;

    location / {
        proxy_pass     http://ca.us.mirror.archlinuxarm.org$request_uri;
        proxy_set_header Host ca.us.mirror.archlinuxarm.org;
    }
}

# Arch Mirror 2 Proxy Configuration
server {
    listen      8002;
    server_name localhost;

    location / {
        proxy_pass     http://co.us.mirror.archlinuxarm.org$request_uri;
        proxy_set_header Host co.us.mirror.archlinuxarm.org;
    }
}

# Arch Mirror 3 Proxy Configuration
server {
    listen      8003;
    server_name localhost;

    location / {
        proxy_pass     http://il.us.mirror.archlinuxarm.org$request_uri;
        proxy_set_header Host il.us.mirror.archlinuxarm.org;
    }
}

```

- Restart nginx:

```
systemctl restart nginx
```

Update System Mirrorlist

- Edit the mirrorlist file:

```
vi /etc/pacman.d/mirrorlist
```

- Add the following line to use this new cache

```
Server = http://pacmirror.example.com/$arch/$repo
```

Manual Cache Cleaning

NOTE: You will need to create a method to clear old packages, as this directory will continue to grow over time. paccache (which is included with pacman) can be used to automate this using retention criteria of your choosing. For example the following command will keep the last 3 versions of packages in your cache directory.:

```
find /var/cache/pacmirror/ -type d -exec paccache -v -r -k 3 -c {} \;
```

Pacman Metadata Backup

- Install rename:

```
pacman -S rename
```

- Create the pacman metadata backup script:

```
vi /usr/local/bin/backup-pacman-meta.sh
```

- And add the following:

```
#!/bin/sh
cd /srv/http/pacmirror

find . -name "*.db" -exec rename --yes -s 's/.db/.db.bak/g' {} \;
```

- Make the script executable:

```
chmod +x /usr/local/bin/backup-pacman-meta.sh
```

- Since the nginx proxy cache config is set to cache the pacman metadata files, they need to be moved so the newer metadata files can be downloaded and cached. Run the script frequently or setup a systemd timer to run the script:

```
backup-pacman-meta.sh
```

Resources

- https://wiki.archlinux.org/index.php/Pacman/Tips_and_tricks#Dynamic_reverse_proxy_cache_using_nginx

History

#1 - 01/30/2016 02:26 PM - Daniel Curtis

- Description updated

- Status changed from New to In Progress

#2 - 01/30/2016 02:40 PM - Daniel Curtis

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

#3 - 01/30/2016 05:49 PM - Daniel Curtis

- Description updated

#4 - 01/30/2016 07:08 PM - Daniel Curtis

- Start date changed from 01/30/2016 to 01/31/2016

#5 - 02/04/2016 02:45 PM - Daniel Curtis

- Description updated

#6 - 02/20/2016 07:32 PM - Daniel Curtis

- Description updated

- Status changed from Resolved to Closed

#7 - 03/19/2018 08:22 PM - Daniel Curtis

- Description updated