CentOS

Content caching

(for faster build-and-deploy ops)

Fabian Arrotin
arrfab@centos.org
@arrfab

CentOS

/whois arrfab

CentOS

Why “caching” content ?

CentOS

Why “caching” content ?

Disclaimer :
following slides will present solutions/workaround mainly
Used for CentOS CI environment / YMMV

CentOS

What to “cache” ?

CentOS

Proxies ?

Squid, nginx, varnish (with caching)
no cache for TLS (*CONNECT*)

CentOS

CentOS CI requirements

=> As close a possible as “real CentOS user experience”

CentOS

“normal” way

(for $corp users)

- Setup internal mirror
  - Select a mirror offerning rsync (https://www.centos.org/download/mirrors/)
  - Use config management to distribute .repo files under /etc/yum.repos.d/
  - Enjoy !

CentOS

“intrusive” way

(aka “transparent way” for CI.centos.org)

CentOS

“intrusive” way

(aka “transparent way” for CI.centos.org)

- Setup vhost that handles mirrorlist (default in shipped yum config)
- Redirect to internal mirror (similar to mirror.centos.org)
- DNS overrides to redirect A/AAAA to internal mirrorlist vhost

CentOS

“intrusive” way

(aka “transparent way” for CI.centos.org)

“fake” mirrorlist vhost

```
#cat /etc/httpd/conf.d/11_mirrorlist.conf
<VirtualHost *:80>
ServerName mirrorlist.centos.org
DocumentRoot /var/www/mirrorlist/
</VirtualHost>

#cat /var/www/mirrorlist/index.php
<?php
$release=$_GET['release'];
$repo=$_GET['repo'];
$arch=$_GET['arch'];

print "http://mirror.centos.org/centos/$release/$repo/$arch\n" ;
?>
```

CentOS

“intrusive” way

(aka “transparent way” for CI.centos.org)

DNS Overrides

Dnsmasq (easy, but “light”, enough for laptop/VMs)

```
sudo su -c "echo ${ip} mirrorlist.centos.org >> /etc/hosts"
sudo pkill --signal HUP dnsmasq
```

CentOS

“intrusive” way

(aka “transparent way” for CI.centos.org)

DNS Overrides

- Unbound

```
grep mirror /etc/unbound/conf.d/01_local_override.conf
# First aliases for local mirrorlist/mirror.c.o
local-data: "mirror.centos.org. IN A 192.168.3.4"
local-data: "mirrorlist.centos.org. IN A 192.168.3.4"
```

CentOS

“intrusive” way

(aka “transparent way” for CI.centos.org)

DNS Overrides

- Bind (rpz zones , bind > 9.8.1)

```
zone “rpz” {
 (…)
}

options {
 (…)
 response-policy { zone “rpz”; } ;
}
```

CentOS

Containers ?

CentOS

Containers ?

Docker.io registry ? (caching server side)

- docker-distribution package !
- Use registry-1.docker.io as upstream registry
- Caches “on demand”

```
--- /etc/docker-distribution/registry/config.yml.orig
+++ /etc/docker-distribution/registry/config.yml
@@ -9,3 +9,6 @@
         rootdirectory: /var/lib/registry
 http:
     addr: :5000
+proxy:
+    remoteurl: https://registry-1.docker.io
+
```

CentOS

Containers ?

Docker.io registry ? (client side)

- dockerd is “mirror” aware
- --registry-mirror=<scheme>://<host>
  Prepend a registry mirror to be used for image pulls. May be specified multiple times.

```
echo "ADD_REGISTRY='--registry-mirror=http://registry.ci.centos.org:5000'"
>> /etc/sysconfig/docker
systemctl restart docker
```

CentOS

Containers ?

(the “unethical” way)

Impersonate registry-1.docker.io

- Get a TLS cert/key that claims to be registry-1.docker.io (bad)
- Setup proxy => docker-distribution (mirror)
- Dns overrides for registry-1.docker.io

```
Disclaimer :
Playing MITM is bad , really bad
Shown just for academic reasons !
```

CentOS

Containers ?

(the “unethical” way)

```
<VirtualHost *:443>
 ServerAdmin webmaster@centos.org
 ServerName registry-1.docker.io
 DocumentRoot /var/www/html
 # Docker read-only registry
 Header always set "Docker-Distribution-Api-Version" "registry/2.0"
 Header onsuccess set "Docker-Distribution-Api-Version" "registry/2.0"
 RequestHeader set X-Forwarded-Proto "https"
 <Location /v2>
   ProxyPass   http://localhost:5000/v2
   <LimitExcept GET HEAD>
     Order Allow,Deny
     Deny from all
   </LimitExcept>
 </Location>
(...)
SSLCertificateFile /etc/pki/tls/certs/registry.crt
SSLCertificateKeyFile /etc/pki/tls/private/registry.key
SSLCertificateChainFile /etc/pki/tls/certs/registry-CAChain.crt
(...)
</VirtualHost>
```

Disclaimer :
Playing MITM is **bad** , really **bad**
Shown just for academic reasons !

CentOS

Containers ?

(the “unethical” way)

Docker node (client side)

- Get the CA that signed the fake registry-1.docker.io cert
- Add it to CA trusted (all that is **bad**)
- Transparently use the caching server

```
# cp $your_ca_that_signed_fake_registry.crt > /etc/pki/ca-trust/source/anchors/
# update-ca-trust extract
```

Disclaimer :
Playing MITM is **bad** , really **bad**
Shown just for academic reasons !

CentOS

Q&A

Contact :
arrfab@centos.org
@arrfab

CentOS