FreeIPA Training Series

Caching Automounter Maps Using the SSSD

Jakub Hrozek

January 2013

Automounter overview

- Automounter lets the user access a share without explicitly mounting it, just by accessing the mount directory
- Usually network filesystems
  - NFS is the default, but any filesystem can be specified using the `-fstype` option in the map
- Mounts are defined in *maps*
  - Think of the maps as `/etc/fstab` equivalent for the automounter
  - Maps can be nested or use automatic variable substitution

Navigating the autofs maps

- There are several kinds of autofs maps
  - master, direct, indirect
- In general they all include data structured in a similar fashion
  - `key [-options] location`
- The difference between direct and indirect maps is the semantics of the key in the map
- The location in both map types is a network share
- See `man 5 autofs` for more information
  - Mount options
  - Substition variables, etc.

The master map

- The master map is the entry point into the hierarchy
- Loaded by the automounter after startup
- There is only one master map
- The master map is called `auto.master` by convention
- The key indicates a direct map or an indirect map mount point
- Location points to the name of a nested maps

Direct maps

- Direct maps are always denoted by a special key named "`/-`" in the master map, for example:
  - `/-  auto.direct`
- In the direct map itself, the key is a full path of a mount point
- The location is a network share
- For example:
  - `/nfs/public  nfs.example.com:/export/pub`

Indirect maps

- Linked to by an absolute mount poitn from the master map, for example:
  - `/share  auto.share`
- In the indirect map, the key is part of the path name after the mount point where the share is mounted
- For example the `auto.share` map might include:
  - `pub  filer.example.com:/export/pub`
  - `mirror  nfs.example.com:/mnt/mirror`
- Then with the master map example above, the client could mount both `/share/pub` and `/share/mirror`

Automounter map sources

- Automounter is able to fetch map data from several sources
  - Files, NIS, LDAP
- LDAP in particular is quite popular
  - Allows to centralize the maps into a single point to administer
  - But also single point of failure if the LDAP server is not reachable
- SSSD is able to cache the autofs maps and provide them to the automounter

The benefits of using the SSSD

- The automounter can access the maps stored in LDAP on its own, so why use the SSSD?
- Autofs map caching. Please note that the SSSD only caches the maps, *not* the contents of the shares
- Unified configuration of LDAP connection parameters such as the server or timeouts at one place (sssd.conf)
- Autofs can utilize advanced features like server fail over or server discovery using DNS SRV lookups
- Only one connection to the LDAP server open at a time

The autofs schema and attributes

- The default attribute names are determined based on the schema used
- Specified by the `ldap_schema` parameter in `sssd.conf`
- Attributes can be overriden to match the schema used by the LDAP server

attribute	SSSD option	RFC2307bis	RFC2307	NIS schema
map objectclass	ldap_autofs_map_object_class	automountMap	automountMap	nisMap
map attribute	ldap_autofs_map_name	automountMapName	ou	nisMapName
entry objectclass	ldap_autofs_entry_object_class	automount	automount	nisObject
entry key	ldap_autofs_entry_key	automountKey	cn	cn
entry value	ldap_autofs_entry_value	automountInformation	automountInformation	nisMapEntry

An example of autofs map

- Let's say we'd like to automount the directory `/share/pub` from a machine with hostname `filer.example.com` and export path `/export/pub`
- We need to define the master map first. The master map contains all other maps

```
dn: automountMapName=auto.master,dc=example,dc=com
automountMapName: auto.master
objectClass: top
objectClass: automountMap
```

An example of autofs map (cont)

- The second part links the /share directory with the auto.share map in an entry object

```
dn: automountKey=/share,\
    automountMapName=auto.master,\
    dc=example,dc=com
automountInformation: auto.share
automountKey: /share
objectClass: top
objectClass: automount
```

An example of autofs map (cont)

- The third part of the example illustrates the auto.share map linked to a key that points to the filer and its exported share.
- We omit the definition of auto.share map, which would look exactly as auto.master, except for the name

```
dn: automountKey=pub,\
    automountMapName=auto.share,dc=example,com
automountInformation: filer.example.com:/export/pub
automountKey: pub
objectClass: top
objectClass: automount
```

Setting up the automounter with the SSSD

1. Install the SSSD plugin for the autofs
   - `yum -y install libsss_autofs`
2. Configure automounter to fetch data from the SSSD
   - Set `sss` as a data source in `/etc/nsswitch.conf`
3. Configure the autofs service in the SSSD
   - Append `autofs` to the `services` line in the `[sssd]` section
   - See if the correct schema is used and set the attribute mapping if neccessary in the domain section

Setting up the automounter with the SSSD

4. Start the automounter and the SSSD
   - `service autofs start`
   - `service sssd start`
5. Access a share to mount it
   - `cd /share/pub`

Example sssd.conf

```
[sssd]
config_file_version = 2
services = nss, pam, autofs

[domain/example.com]
id_provider = ldap
autofs_provider = ldap
ldap_schema = rfc2307bis
ldap_search_base = dc=example,dc=com
ldap_autofs_search_base = cn=automount,dc=example,dc=com
```

Debugging the autofs configuration

- Is the automounter able to print the mounts?
  - `automount --dumpmaps`
- Run the automounter in foreground
  - `automount -df`
  - Is the `sss` module being loaded
  - Does the automounter find the maps you expect it to find?

Debugging the SSSD configuration

- Inspect the SSSD configuration file
  - Does the configuration specify the correct schema and/or attributes?
  - Is the `sssd_autofs` process running?
- Check the SSSD debug logs
  - Set debug_level in the `[autofs]` and [domain/$name] sections of the SSSD, restart the SSSD.
  - Are there any requests coming in visible in the `sssd_autofs.log` file?
  - Are there any LDAP searches visible in the `sssd_domain.log` file?

More resources

- On automounter itself
  - `man 5 autofs`, `man auto.master`
  - Storage Administration Guide at https://docs.redhat.com
- SSSD configuration
  - `man sssd.conf`, section Autofs configuration options
    - Configuring the autofs front end service
  - `man sssd-ldap`, section Autofs options
    - The attribute mappings and the search bases
  - `man sssd-ipa`
    - Configuring the autofs location