OpenAFS eine eigene Zelle Aufsetzten

Lars Schimmer, Andreas Läßer, Markus Köberl, Mathias Feiler

CLT2012, Samstag, 13:30 Uhr, Raum V5

Die folgende Auflistung von Command-Line Befehlen und kurzen Erläuterungen soll eine Hilfestellung zum Workshop "OpenAFS - Eine eigene AFS-Zelle aufsetzen" der Chemnitzer Linuxtage sein. Sie ist absichtlich zur Zeit nicht online verfügbar, da der beste Lerneffekt durch eigenhändiges ausführen der Befehle gegeben ist.
Eine digitale Version dieses Dokuments wird es nach dem Workshop auf der Website www.openafs.at zum Download bereitgestellt.

1 Einleitung/Basisteil

Als erstes soll das Virtualbox Image gestartet werden. Es enthält eine aktuelle Debian Version die als Grundlage für den Workshop und die virtuellen AFS Server dient, das Root Passwort ist auf "123456" gesetzt. Es sei vorweggenommen das virtualisierte AFS-Server ggf. nicht die gewünschte Performance bringen, darum wird auch von einer Virtualisierung im Produktivbetrieb ohne besondere Anpassungen abgeraten.
Man kann im Image die /etc/resolv.conf ändern, damit beim Booten der DNS Check keine Wartezeit verursacht. Einfach auf "local" setzen.
Zum Aufsetzen eines 2. Filservers clont man das Image und startet das 2. Image hoch und ändert die Files /etc/hosts.txt und /etc/hostname (wir setzen die IP und den Namen einfach 1 hoch). Jetzt starten wir die VM neu.

2 Setup des ersten AFS-Servers

```
root@machine01# mkdir /vicepa
```

Als Erstes muss die Partition `/vicepa` gemounted werden. In userem Fall ist dies eine kleine Partition, die den Namen `vicepa` trägt. Auf einem Produktivsystem wird dies eine eigene Partition auf einem Raid oder externem Storage sein.

```
root@machine01# mount /dev/sdb1 /vicepa
```

Hierbei nicht vergesssen die fstab ändern, damit die Partition auch nach einem reboot wieder gemounted wird.
Nun können wir beginnen die notwendige Software zu installieren:

```
root@machine01# aptitude install krb5-kdc krb5-admin-server
```

Dabei beachten, das der `KRB5-REALM CLT12.DE` ist und beide Krb5 Server mit `192.168.1.100` eingetragen werden in der Konfigurationsabfrage. Jetzt müssen wir noch diese REALM im krb5 Server anlegen, das Passwort setzten wir hier einfach auf `123456`:

```
root@machine01# krb5_newrealm
```

Zur Betreuung des krb5 Servers wird ein Administrator Benutzer benötigt, den legen wir in kadmin.local an. Passwort wird einfach `123456` gesetzt:

```
root@machine01# kadmin.local
> addprinc admin
> quit
```

Nun müssen wir noch dem Administrator Benutzer auf dem krb5 Server alle Rechte zuweisen in der `kadm5.acl` Datei (`*/admin *`):

```
root@machine01# vim /etc/krb5kdc/kadm5.acl
```

Nun den AFS Principal zur `clt12.de` Zelle erstellen. Dieser wird zur Authentifizierung der Nutzer benötigt:

```
root@machine01# kadmin.local -q "addprinc -randkey afs/clt12.de"
```

Danach fügen wir dem Eintrag in das Keytab file eine `des-cbc-crc:afs3` Verschlüsselung an:

```
root@machine01# kadmin.local -q "ktadd -e des-cbc-crc:afs3 afs/clt12.de"
```

Hinweis: Man kann hier auch die Datei angeben, in welches das Keytab exportiert wird, Standard ist /etc/krb5.keytab!
Hinweis 2: hier wird die aktuelle KVNO ausgegeben. Die benötigen wir später noch, unbedingt notieren!
Jetzt müssen wir noch die `/etc/krb5.conf` editieren um die alte des-cbc-crc Verschlüsselung im aktuellen krb5 nutzen zu können. Eine Änderung des OpenAFS Protokolls zur Verwendung anderer, aktueller Methoden ist zur Zeit im Prozess der Standardisierung. Dazu fügen wir unter *permitted_ enctypes* folgendes hinzu:

```
allow_weak_crypto = true
```

Nun installieren wir die OpenAFS Serverpakete für die Workstation:

```
root@machine01# aptitude install openafs-dbserver openafs-krb5
```

Die Frage nach der Zelle beantworten wir mit "clt12.de" den AFS Cache lassen wir vorerst einmal auf den Standard Werten.

Hinweis: Da der OpenAFS Client installiert wird, wird das KernelModul automatisch mitgebaut. Dieses kann etwas Zeit beanpsruchen.

Nun fügen wir den krb5-Key dem OpenAFS Keyfile hinzu. Hinweis: die KVNO (hier 2) ist abhängig von der KVNO, die wir uns weiter oben notiert haben. Immer die aktuelle nutzen!:

```
root@machine01# asetkey add 2 /etc/krb5.keytab afs/clt12.de
```

Hinweis: falls oben eine andere Datei angeben wurde, muß /etc/krb.keytab ersetzt werden mit der oben angegebenen Datei!

Danach beenden wir alle BOS-Server und editieren die folgenden zwei Dateien:
`/etc/openafs/CellServDB` und `/etc/openafs/server/CellServDB`

```
>clt12.de          #CLT2012 test Zelle
192.168.1.100     #cltafs1
```

Hinweis: die Dateien unter /server betreffen nur den Server, die im Pfad darüber nur den Client!
Hinweis 2: das Format der CellServDB ist festgelegt, das ist KEIN Kommentarzeichen!
Hinweis 3: cltafs1 ist der DNS Eintrag für die IP 192.168.1.100, OpenAFS probiert zuerst den DNS Eintrag, dann die IP zu erreichen.

Nun wird der BOS-Server mit lokaler Authentifizierung gestartet...

```
root@machine01# bosserver -noauth &
```

Nun müssen wir sicherstellen das der BOS-Server auf dem richtigen Server läuft...

```
root@machine01# bos listhost 192.168.1.100 -noauth
```

Danach können wir nun die erste OpenAFS Serverinstanz im BOS-Server anlegen, hier der PTServer (ProTection Server), welcher die Benutzer und Gruppen der Zelle verwaltet:

```
root@machine01# bos create -server 192.168.1.100 -instance ptserver -type
     simple -cmd /usr/lib/openafs/ptserver -cell clt12.de -noauth
```

Nun müssen wir noch den oben angelegten admin Benutzer der Gruppe der OpenAFS-Adminuser hinzufügen (lokal auf dem Server):

```
root@machine01# bos adduser 192.168.1.100 admin -cell clt12.de -noauth
```

Auch am PTServer muss der Administrator Benutzer angelegt werden (Der Administrator Benutzer bzw. erste Benutzer der angelegt wird bekommt immer die `ID 1`; hiernach kann der Administrator Benutzer nicht nur mit lokaler Authentifizierung den Server verwalten) und er wird der Gruppe der OpenAFS Administratoren hinzugefügt:

```
root@machine01# pts createuser -name admin -cell clt12.de -noauth
root@machine01# pts adduser  admin system:administrators -cell clt12.de -
    noauth
```

Nun Starten wir mittels BOS-Server alle OpenAFS Services neu:

```
root@machine01# bos restart 192.168.1.100 -all -cell clt12.de -noauth
```

Jetzt kann die erste Fileserver (Daten) und VolumeLocationserver (Wo ist welches Volume) Instanz angelegt werden:

```
root@machine01# bos create -server 192.168.1.100 -instance fs -type fs -
    cmd /usr/lib/openafs/fileserver -cmd /usr/lib/openafs/volserver -cmd /
    usr/lib/openafs/salvager -cell clt12.de -noauth
root@machine01# bos create -server 192.168.1.100 -instance vlserver -type
     simple -cmd /usr/lib/openafs/vlserver -cell clt12.de -noauth
```

Jetzt sind wir bereit ein root-Volume ins Leben zu rufen:

```
root@machine01# vos create 192.168.1.100 a root.afs -cell clt12.de -
    noauth
```

Hinweis: hier nutzen wir die Abkürzung der vicepa Angabe zu a. Alle OpenAFS Programme akzeptiere Abkürzungen der Parameter, solange diese eindeutig sind!

Nun stoppen wir den BOS-Server und machen ein Processkill damit auch sicher alle Instanzen beendet sind...

```
root@machine01# bos shutdown 192.168.1.100 -wait -noauth
root@machine01# pkill bosserver
```

Schlussendlich starten wir den Fileserver per Debian Startskript im Hintergrund neu ohne lokale Authentifizierung:

```
root@machine01# /etc/init.d/openafs-fileserver start
```

Für den Client passen wir jetzt noch die Standardeinstellungen für einen problemlosen Betrieb an. Dabei wird die Datei `/etc/openafs/afs.conf.client` geändert auf folgende Werte:

```
AFS_CLIENT=true
AFS_AFSDB=true
AFS_CRYPT=false
AFS_DYNROOT=false
AFS_FAKESTAT=true
```

"Client" meint den Start eines OpenAFS Clients auf der Workstation, *"Crypt"* ist eine einfache Streamverschlüsselung aller OpenAFS Daten zwischen Client und Server, *"Dyn-Root"* simuliert den Zugriff auf das root.afs Volume einer Zelle und mountet diese erst bei Bedarf und *"FakeStat"* legt bei Bedarf Fake Volumes an, damit ein `ls` Aufruf schneller geht (timeouts).
Alternativ ruft man dpkg-reconfigure openafs-client auf und setzt die Werte entsprechend.

```
root@machine01# vim /etc/openafs/afs.conf.client
```

Nachdem wir nun den Client soweit konfiguriert haben, starten wir den gesamten Server neu, um ein frisches Setup aller Services zu haben:

```
root@machine01# reboot
```

Nach dem Reboot loggen wir uns als *"root"* ein, authentifizieren uns als Administrator beim krb5-Server und holen uns ein OpenAFS Token als Administrator:

```
root@machine01# kinit admin
root@machine01# aklog
```

Mit den Zugriffsrechten als Administrator können wir nun Leserechte für den Pfad `/afs/` vergeben, in diesem Fall Read/List für alle OpenAFS Nutzer weltweit:

```
root@machine01# fs setacl /afs system:anyuser rl
```

Um unsere neue Zelle verfügbar und nutzbar zu machen, benötigen wir weitere Volumes. Wir legen das root-Volume für unsere Zelle auf unserem OpenAFS Fileserver in Partition `/vicepa` an:

```
root@machine01# vos create 192.168.1.100 vicepa root.cell
```

Damit das Volume im Pfad erscheint, muß es mit den OpenAFS Befehlen eingehangen werden, zur einfachen Erkennung nennen wir den Pfad gleich wie unsere Zelle:

```
root@machine01# fs mkmount /afs/clt12.de root.cell
```

Da OpenAFS ReadWrite und ReadOnly Volumes kennt, erstellen wir einen Pfad, auf dem immer ein RW Volume eingehangen ist:

```
root@machine01# fs mkmount /afs/.clt12.de root.cell -rw
```

Nun setzen wir die ACLs (Zugriffsrechte) auf den neu eingehangenen Volumes für jeden OpenAFS Benutzer weltweit auf RL (system:administrator hat schon per se volle Zugriffsrechte):

```
root@machine01# fs setacl /afs/.clt12.de system:anyuser rl
```

Somit ist der Basisteil zum Aufsetzen einer funktionsfähigen OpenAFS Zelle fertig. Mit diesem Setup kann man weitere Benutzer, Gruppen, Volumes, Partitionen anlegen und nutzen.
Die volle Funktionalität spielt OpenAFS jedoch erst mit mehreren Servern aus, da dann mehrere RO Kopien eines Volumes angelegt werden können und diese einerseits ein einfaches Backup darstellen, andererseits dem Load-Balancing zwischen den Fileservern dienen.

3 Einen zweiten Fileserver aufsetzen - Optional

Wie schon mit der ersten VM hängen wir die zweite Partition als `/vicepa` in das Dateisystem:

```
root@machine02# mount /dev/sdb1 /vicepa
```

Wie gehabt ändern wir auch die fstab unter `/etc/fsatb` zum automatischem Mounten der Partition nach einem Neustart. Danach installieren das OpenAFS-Fileserver Paket. Falls der Server auch ein DBServer werden soll, das `OpenAFS-dbserver` Paket mit installieren:

```
root@machine02# aptitude install  openafs-fileserver
```

Damit der Server selber und die OpenAFS Zelle die Zugehörigkeit untereinander wissen, ändern wir zuerst die CellServDB auf Server 1 und kopieren dann einige Daten vom ersten Server auf jeden weiteren Server. Darunter fallen: `CellServDB`, `ThisCell`, `afs.keyfile` und `Userlist` (Alternativ nutzt man bos addhost):

```
>clt12.de          #CLT2012 test Zelle
192.168.1.100    #cltafs1
192.168.1.101  #cltafs2
```

Jetzt kopieren:

```
root@machine02# scp root@192.168.1.100:/etc/openafs/CellServDB /etc/
    openafs/CellServDB
root@machine02# scp root@192.168.1.100:/etc/openafs/CellServDB /etc/
    openafs/server/CellServDB
root@machine02# scp root@192.168.1.100:/etc/openafs/ThisCell /etc/openafs
    /ThisCell
root@machine02# scp root@192.168.1.100:/etc/openafs/ThisCell /etc/openafs
    /server/ThisCell
root@machine02# scp root@192.168.1.100:/etc/openafs/server/afs.keyfile /
    etc/openafs/server/afs.keyfile
root@machine02# scp root@192.168.1.100:/etc/openafs/server/UserList /etc/
    openafs/server/UserList
```

Hinweis: Man beachte die Unterscheidung der CellServDB für Client und Server, ebenso das afs.keyfile.

Nun erstellen wir auf dem zweiten Fileserver eine OpenAFS FileServer Instanz, rufen allerdings den Befehl dazu auf dem 1. Server auf, da wir dort schon admin der OpenAFS Zelle sind:

```
root@machine01#   bos create -server 192.168.1.101 -instance fs -type fs
    -cmd /usr/lib/openafs/fileserver -cmd /usr/lib/openafs/volserver -cmd
    /usr/lib/openafs/salvager -cell clt12.de -noauth
```

Nun starten wir den zweiten Server und anschliessend OpenAFS auf dem ersten Server neu:

```
root@machine02# reboot
root@machine01# /etc/init.de/openafs-fileserver restart
```

4 Bevölkern der OpenAFS Zelle mit Volumes

Nachdem die Zelle `clt12.de` jetzt über zwei Fileserver und zwei Partitionen verfügt, bevölkern wir diese Zelle mit weiteren Benutzern, Gruppen und Volumes. Zuerst legen wir neue Volumes an (vorerst müssen wir aber ein ticket/token Paar als `admin` geholt haben!):

```
root@machine01# vos create 192.168.1.100 a user
root@machine01# vos create 192.168.1.100 a user.meyer
root@machine01# vos create 192.168.1.100 a user.schulz
root@machine01# vos create 192.168.1.100 a data
root@machine01# vos create 192.168.1.100 a data.video
root@machine01# vos create 192.168.1.100 a data.fotos
```

Hinweis: Wir nutzen wieder die Abürzungen bei den Optionen, die sind hier eindeutig.
Hinweis 2: Wir nutzen hier nur den 1. Filserver, man kann als Server aber auch jeweils den 2. Server angeben (192.168.1.101).
Nachdem die Volumes angelegt sind, teilen wir dem VLServer mit, das wir gerne eine RO Kopie der RW Volumes erstellen möchten:

```
root@machine01# vos addsite 192.168.1.100 a root.afs
root@machine01# vos addsite 192.168.1.100 a root.cell
root@machine01# vos addsite 192.168.1.100 a user
root@machine01# vos addsite 192.168.1.100 a user.meyer
root@machine01# vos addsite 192.168.1.100 a user.schulz
root@machine01# vos addsite 192.168.1.100 a data
root@machine01# vos addsite 192.168.1.100 a data.video
root@machine01# vos addsite 192.168.1.100 a data.fotos
```

Jetzt hängen wir die neu erstellen Volumes in den Dateibaum ein:

```
root@machine01# fs mkmount /afs/.clt12.de/home user -rw
root@machine01# fs mkmount /afs/.clt12.de/home/meyer user.meyer -rw
root@machine01# fs mkmount /afs/.clt12.de/home/schulz user.schulz -rw
root@machine01# fs mkmount /afs/.clt12.de/data data
root@machine01# fs mkmount /afs/.clt12.de/data/video data.video
root@machine01# fs mkmount /afs/.clt12.de/data/fotos data.fotos -rw
```

Eine Zelle benötigt natürlich noch Benutzer neben dem Administrator:

```
root@machine01# pts createuser -name meyer -id 1000
root@machine01# pts createuser -name schulz -id 1001
root@machine01# pts creategroup -name users -owner meyer -id -1000
root@machine01# pts adduser -user meyer -group users
root@machine01# pts adduser -user schulz -group users
```

Damit die Benutzer auch zugreifen können, setzten wir die Zugriffsrechte (ACLs) auf den Verzeichnissen:

```
root@machine01# fs setacl /afs/.clt12.de/home system:anyuser rl
root@machine01# fs setacl /afs/.clt12.de/home/meyer meyer write
root@machine01# fs setacl /afs/.clt12.de/home/schulz schulz write
root@machine01# fs setacl /afs/.clt12.de/data users rl
root@machine01# fs setacl /afs/.clt12.de/data/video users rl
root@machine01# fs setacl /afs/.clt12.de/data/video schulz write
root@machine01# fs setacl /afs/.clt12.de/data/fotos users write
```

Anschliessend setzen wir jeweils ein Quota:

```
root@machine01# fs setquota /afs/.clt12.de/home/schulz 2048000
root@machine01# fs setquota /afs/.clt12.de/home/meyer 4096000
root@machine01# fs setquota /afs/.clt12.de/data/video 2048000
root@machine01# fs setquota /afs/.clt12.de/data/fotos 10240000
```

Nachdem ACLs und Quota gesetzt sind, kopieren wir den Status des RW Volumes auf das RO Volume (Daten, ACL und Quota Informationen):

```
root@machine01# vos release root.cell
root@machine01# vos release root.afs
root@machine01# vos release user
root@machine01# vos release user.schulz
root@machine01# vos release user.meyer
root@machine01# vos release data
root@machine01# vos release data.video
root@machine01# vos release data.fotos
```

Jetzt müssen die neu erstellten Benutzer nur noch im krb5 Server erstellt werden und jeweils ein Passwort zugewiesen bekommen. Danach können die Benutzer auf der OpenAFS Zelle arbeiten mit Dateien rein/raus kopieren, lesen, ändern,...

Generelle Informationen zu einer OpenAFS Zelle:

- 3 DB-Server sind ideal
- jedes Volume sollte eine RO Kopie haben
- RW/RO Pfad beachten
- Volumes sollten nicht zu groß werden (quota)
- mittels vos dump / vos backup kann ein Dump eines RW Volumes erstellt werden. Dieses kann dann einfach als Backup verwendet werden.
- in der Datei /var/lib/openafs/local/NetRestrict kann man angeben, auf welchen Netzen/IPs die OpenAFS Server nicht laufen sollen
- in der Datei /var/lib/openafs/local/NetInfo kann man den OpenAFS Server angeben, auf welcher IP sie laufen sollen