Rocks-Cluster und NIS

NIS Server Konfiguration allgemein

Der Rocks-Cluster (http://www.rocksclusters.org) ist ein Cluster, der in vielen verschiedenen Konfigurationen gestartet werden kann. Ein tolles Ding, das wirklich kaum Wünsche offen lässt. Trotzdem hat er auch so seine Probleme, z.B. wenn es um eine zentrale User-Verwaltung geht, denn mit einem externen NIS-Server kann der Cluster nicht umgehen.

Das kompliziert die Lage, da man für jeden User, der im Netzwerk unterwegs ist, eine eigene Konfiguration am Cluster ablegen muss. Eine zentrale User-Verwaltung bringt aber viele Vorteile!

Also liegt es nahe, den Cluster selbst zum NIS-Server zu machen und die User-Verwaltung auf dem Server abzulegen.

Ich habe in diesem HOWTO den Cluster-Head verwendet, der ja eigentlich nur die Jobverteilung übernimmt und selbst nicht am „clustern“ teilnimmt.

Das Vorhaben!

Im Netz ist neben dem Cluster auch ein Storage, welches ca 500TB Datenspeicher zur Verfügung stellt. Manche der bestehenden Daten sind „sensibel“ sprich sie dürfen nicht prinzipiell jedem zugänglich gemacht werden, weil sie z.B. private Informationen (also un-anonymisiert) enthalten. Andere sind schon anonymisiert und können von allen genutzt werden.
Dazu wäre es also schön, wenn man Gruppen definieren könnte, die auf die Daten zugreifen können und solche, die keinen Zugriff haben.
Neben dem Zugriff über den Cluster soll natürlich auch direkt auf das Storage-System zugegriffen werden. Da dort die genannten Daten abgelegt werden, müssen hier auch die Rechte ziehen.

Für kleine Netze und geringe Anforderungen an die Rechtevergabe bietet sich also NIS (yellow-pages oder auch ypserv) geradezu an. Ab 100 User oder mehr, würde ich LDAP einsetzen.

Ich gehe von einem neu installierten Rocks-Cluster aus, auf dem es noch keine User gibt und den wir komplett neu installiert haben. Weiters setze ich voraus, dass man in der Konfiguration eines Linux.-Servers nicht kompletter Neuling ist. Ich erkläre nicht alle Befehle im Detail und setze das Eine oder Andere voraus.

Das Storage wird per NFS angebunden und die Rechte sollen hier direkt übergeben werden. Storage-Installation und NFS-Server erkläre ich auch nicht, nur wie ich die Einträge mit der Rechteübergabe an den Cluster-Head bewerkstellige.

Dieses Howto kann auch für die allgemeine NIS-Server/Client Konfiguration verwendet werden, man lässt die Rocks spezifischen Hinweise einfach aus.

NIS

Um sich an einem Linux-System anzumelden werden im Wesentlichen die Dateien **/etc/passwd**, **/etc/shadow** und **/etc/group** abgefragt. Hier liegen alle Informationen um einen User oder oder eine Gruppe im System eindeutig zu identifizieren.

NIS-Server sorgen wiederum eigentlich nur dafür, dass auf allen angeschlossenen Client-Systemen diese Informationen indentisch sind, indem es aus obige Dateien vom Server Informationen an die Clients ausliefert, b.z.w Clients den Server nach Einträgen in diesen Dateien abfragen.

Man kann noch weitere Informationen vom Server an Clients senden, NIS ist da recht flexibel, darauf gehe ich aber nicht extra ein.

ACHTUNG !

ALLE Server und Clients die NIS nutzen sollen, müssen im selben Netzwerk liegen. Ich sichere den NIS-Server entsprechend ab (siehe ***/var/yp/securenets)***.

Den NIS-Server (auf dem Rocks Cluster-Head) installieren

Um auf dem Rocks-Cluster-Head (derzeit CentOS 6.5 als Systembasis) NIS als Server zu installieren lädt man sich zunächst das passende Paket ypserv herunter und installiert das Paket.

```
wget ftp://rpmfind.net/linux/centos/6.6/os/x86_64/Packages/ypserv-2.19-26.el6_4.2.x86_64.rpm
rpm -ivh ypserv-2.19-26.el6_4.2.x86_64.rpm
```

Sofern noch nicht installiert brauchen wir noch rpcbind:

```
yum -y install rpcbind
```

Als Nächstes editieren wir die Datei **/etc/sysconfig/network** und setzen folgenden Eintrag:

```
NISDOMAIN=nis.world
```

Der Server ist nun installiert und sollte schon startfähig sein. Test:

```
/etc/init.d/ypserv start
```

Für eine CentOS Installation siehe auch

http://www.server-world.info/en/note?os=CentOS_6&p=nis

UID UND GID

Rechte an Dateien und Ordnern werden im Linux/Unix Dateisystem über User und Gruppen-ID`s vergeben. Es ist also logisch, dass man dafür sorgen muss, dass auf allen Geräten, auf denen man Rechte für einen User/Gruppe setzen möchte immer die selbe ID braucht, sonst gibt es Rechtesalat.

Ich lasse es zu, dass man auf einem Client sowohl lokale User/Gruppen als auch den NIS-Server verwenden kann. NIS-Clients lassen sich so einstellen, dass sie zuerst lokal nachschaut, ob es einen User gibt und wenn nein, erst dann den NIS-Server abfragen.
So kann ich auch später nahezu jeden Server an den NIS anbinden, ohne die User auf den NIS portieren zu müssen oder Rechte auf dem Client zu verlieren. Solche Mischvarianten sind aber mit Vorsicht zu geniessen.

Um also Konflikte auf den zukünftigen Clients mit z.B. lokal eingerichteten Usern zu verhindern, setzen wir die Vergabe der UID`s (User-Id) und GID's (Gruppen ID) für unsren NIS-Server auf einen neuen Wert (ich beginne mit User-ID's ab 5000 und Gruppenid's ab 5000, die Werte können aber problemlos angepasst werden).

Wir editieren am NIS-Server die Datei **/etc/login.defs**

```
UID_MIN          5000
UID_MAX         60000

GID_MIN          5000
GID_MAX         60000
```

Ab nun sollten neue User und neue Gruppen mit einer fortlaufenden ID ab 5000 angelegt werden.

Man kann am Rocks-Cluster nun die ersten Gruppen und User anlegen.

```
    groupadd closed
    groupadd open
    groupadd sonstige
```

Ein cat **/etc/group** gibt mir dann hoffentlich folgendes aus (also ID`s ab 5000):

```
open:x:5000:
closed:x:5001:
sonstige:x:5002:
```

User können nun ebenfalls angelegt werden:

```
Useradd testuser
Passwd testuser
```

Um den User „*testuser*“ nun zusätzlich der Gruppe „*closed*“ zuzuweisen genügt ein

```
usermod -aG closed testuser
```

Wichtig!

Um alle User des Servers im Netzwerk verfügbar zu machen, muss der NIS-Server die User erst „übernehmen“ und entsprechende Dateien erstellen. Nach der Anlage der Users oder der Gruppen muss immer der Befehl

```
make -C /var/yp
```

gestartet werden, erst dann sind alle User und Gruppen im Netzwerk verfügbar. Es gilt:

> Lieber einmal zuviel, als einmal zu wenig.

Ein „***/etc/init.s/ypserv restart***“ kann auch nicht schaden.

In der Datei ***/var/yp/Makefile*** muss noch eine Einstellung vorgenommen werden. Wir stellen in der Datei folgende Parameter um/ein:

```
MINUID=5000
MINGID=5000
```

Diese Einstellung sichert, dass auch nur User und Gruppen-Informationen ab der ID 5000 über den Server ausgeliefert werden. Alle ID`s davor bleiben unberücksichtigt.

Warum das denn?
Das ist deshalb wichtig, weil installierte Serverdienste (z.B. der Apache2 Webserver) auch ID's bei ihrer Installation zugewiesen bekommen und so Rechte an Verzeichnissen und Dateien erhalten. Normalerweise sind das ID`s von 1 bis 999. Diese ID' s „können“ aber von Server zu Server unterschiedlich sein, je nachdem, welcher Dienst dort läuft und in welcher Reihenfolge Dienste installiert werden.
Manche Server weisen die ID dynamisch zu und vergeben keine feste ID, andere setzten eine ID mit einer festen Nummer. Diese Einstellung gewährleistet also, dass bestehende Server weiterhin korrekt arbeiten und nicht mit falschen Dienst-ID`s versorgt werden. Wir gehen auf Nummer sicher.

Wir kontrollieren in dieser Datei (***/var/yp/Makefile***) noch folgende Parameter:

```
MERGE_PASSWD=false
MERGE_GROUP=false
all: passwd shadow group hosts rpc services netid protocols
```

Sollte die Datei ***/var/yp/securenets*** nicht vorhanden sein, erstellen wir sie und fügen dort ein:

```
255.255.255.0  NETADRESSE_DES_SERVERS
```

Natürlich muss die NETADRESSE der, des eigenen Netztes, angepasst sein. Diese Einstellung stellt sicher, dass nur Clients aus dem eingetragenen Netzwerk auf den NIS-Server zugreifen können.

Zum Schluss stellen wir sicher, dass der ypserver und der Dienst rpcbind automatisch nach einen Boot gestartet werden.

```
chkconfig rpcbind on
chkconfig ypserv on
```

Ich würde nun den Server einmal rebooten...!

Zumindest:
Nochmal ***make -C /var/yp*** schadet nicht!

Der Server sollte nun die Informationen aus /etc/passwd, /etc/group und /etc/shadow ab der ID 5000 ausliefern und die Clients damit versorgen.

NIS-Client installieren

Die Clients erfordern etwas mehr Arbeit.

Zur Information warum ich das folgende tue:
Rocks legt User etwas anders an, als herkömmliche Linux-Server. So werden z.B. die User-Verzeichnisse bei der Useranlage nicht wie gewöhnlich unter */home* erstellt, sondern ins Verzeichnis */export/home* verlagert. Legt man nun einen User neu an, steht in der Datei */etc/passwd* als Homedir zunächst folgendes:

```
testuser:x:5001:5003::/export/home/testuser:/bin/bash
```

Im Rocks-Cluster werden User normalerweise, nach der Anlage, mit dem Befehl

```
rocks sync users
```

den angeschlossenen Nodes zu Verfügung gestellt. Rocks sorgt dann dafür, dass alle Cluster-Nodes diese User-Verzeichnisse (/export/home/USER) automatisch einbinden können. Es werden einige interne Dsteien erstellt, die an die Cluster-Nodes weitergereicht werden.
Erst dann wird der *Homedir-Pfad* (oben */export/home/testuser*) in der Datei /etc/passwd auf den Standard-Linux-Pfad */home/testuser* angepasst.

Wir basteln nicht an der Rocks-Konfiguration herum, gehen aber auf Nummer sicher. Sollte einmal der Befehl „*rocks sync users*“ nicht ausgeführt werden, hätte das auf den Clients eventuell Folgen, die ich verhindern möchte.

Daher:

Um nun generell Login-Probleme auf den Clients zu verhindern habe ich auf den angeschlossenen Clients die home-Stuktur an Rocks so angepasst, dass User, die sich anmelden immer ein Homedir finden.

```
mv /home /homeold
mkdir /export
mkdir /export/home
ln -s /export/home /home
```

Sollte sich ein User nun anmelden, sollte er immer sein Homedir finden, egal, was in /etc/passwd steht.

Sollten schon User auf dem zukünftigen NIS-Client bestehen, muss man die User-Verzeichnisse nun noch mit allen Attributen kopieren.

```
cp -a /homeold/* /export/home
```

Wer einen simplen NIS-Server, ohne Rocks Bezug installieren möchte b.z.w sich sicher ist, dass er IMMER die User auf dem Rocks synced, kann diesen Schritt aber auch auslassen.

Wir konfigurieren nun NIS auf den Clients (zuerst in CENTOS, RED-HAT)

Mein Storage läuft ebenfalls mit Centos. Damit ich meine Daten mit korrekten Rechten ausliefern kann muss das Storage ebenfalls an den NIS-Server angebunden werden, damit ihm User und Gruppen zur Verfügung stehen. Zuerst installieren wir NIS

```
yum -y install ypbind rpcbind
```

Dann editieren wir folgende Dateien:

Datei ***/etc/sysconfig/network***

```
NETWORKING=yes
NISDOMAIN=server.world
```

Datei ***/etc/sysconfig/authconfig***

```
USENIS=yes
```

Datei ***/etc/yp.conf***

```
ypserver IP_DES_NIS_SERVERS
```

Datei ***/etc/nsswitch.conf***

```
Passwd: files nis
Shadow: files nis
Group: files nis
Hosts: files dns nis
```

Hinweis:

Die Reihenfolge **files nis** stellt sicher, dass auf dem Client zuerst die lokalen Files für User/Gruppen abgefragt werden und sollte ein Login dort nicht gefunden werden, der NIS-Server. So bleiben z.B. alle Root-Passwörter und bestehende User auf dem jeweiligen NIS-Client erhalten.

Wenn auf dem Client beim ersten Login automatisch ein USER-Verzeichnis erstellt werden soll, dann müssen noch einige Dateien unter ***/etc/pam.d/*** editert werden. Bei mir sind das die Dateien login, system-auth und sshd (eventuell noch gdm)

Hinzufügen folgender Zeile

```
session     optional      pam_mkhomedir.so skel=/etc/skel umask=077
```

Dieser Eintrag stellt sicher, dass bei einem Login eines Users ein User-Verzeichnis erstellt wird, sofern noch nicht vorhanden. Ausserdem vergibt es für das Homedir die Rechte 700, so kann nur der User auf dieses Verzeichnis zugreifen.

Zum Schluss noch das automatische Starten der Dienste:

```
chkconfig rpcbind on
chkconfig ypbind on
```

Siehe auch: http://www.server-world.info/en/note?os=CentOS_6&p=nis&f=2

Nach einem Reboot, kann man auf dem Client nun versuchen sich mit dem ersten User einzuloggen.

Der Befehl:

```
ypact passwd
```

sollte nun alle User vom NIS-Server ab der ID 5000 auflisten.

Wir konfigurieren nun NIS auf den Clients (Debian)

Ich habe noch einige andere Server, die mit Debian laufen und ebenfalls den NIS-Server als zentrale User-Verwaltung nutzen sollen. Unter Debian sind die Clients etwas anders zu konfigurieren.

```
apt-get -y install nis
```

installiert die notwendigen Pakete

Wir editieren folgende Dateien:

Datei ***/etc/yp.conf*** einfügen

```
ypserver IP_DES_NIS_SERVERS
```

Datei ***/etc/nsswitch.conf***

```
passwd: files nis
shadow: files nis
group: files nis
hosts: files dns nis
```

Datei ***/etc/defaultdomain***

```
nis.world
```

Datei ***/etc/default/nis*** wir kontrollieren/setzen

```
NISSERVER=false
NISCLIENT=true
YPPWDDIR=/etc
```

Nach einem Reboot sollte auch hier ein Login mit Userdaten des NIS-Servers funktionieren.

Als User das Passwort ändern:

User wollen Ihre Passwörter natürlich selbst verwalten. NIS bietet hierzu einen Dienst Namens yppasswdd an, der am Server gestartet werden kann und Passwortänderungen durchführt.
Leider hakt es da aber bei mir (ich konnte noch nicht herausfinden wieso) und ich erhalte einen Fehler (auf mehreren Rocks-Clustern den selben).

Hinw.

> Auf einem NIS-Server, der nicht auf dem Rocks-Cluster läuft, funktioniert der Dienst und User können mit yppasswd ihr Passwort ändern. Sollte ich den Fehler finden, reiche ich die Information nach.

Trotzdem habe ich mir einen Weg gebaut. User loggen sich auf dem Rocks-Server (und nur dort) ein und erstellen sich mittels „***passwd“***auf der Konsole ein neues Passwort. Ein Mal je Stunde lassen ich ein Script laufen, dass diese Passwörter nun für den NIS-Server übernimmt und den NIS-Server neu startet.
Ich habe mir ein Verzeichnis /scripts erstellt, in dem das Script „*restart_ypserv.sh*“ liegt:

```
#!/bin/bash

echo "-------------------------------------------------" >> /var/log/restart_ypserv.log
echo `date` " --> ypserver update STARTED...." >> /var/log/restart_ypserv.log
make -C /var/yp
/etc/init.d/ypserv restart
```

Mit crontab -e kann man nun einen Cronjob anlegen, der dieses Script stündlich startet. Der Eintrag lautet:

```
0 * * * * /scripts/restart_ypserv.sh
```

Storage Rechte absichern

Wie schon Oben erwähnt, greift mein Cluster auf ein Storage zu, auf dem die Rechte des NIS-Servers gesetzt werden sollen. Das Storage liefert die Verzeichnuisse per NFS-Server aus und ist bereits an den NIS-Server als Client angebunden und ein „*ypcat group*“ zeigt mir folgendes (siehe oben User und Gruppen am Server anlegen)

```
open:x:5000:
closed:x:5001:
sonstige:x:5002:
```

Die Rechte der Verzeichnisse vom Storage an Hand zweier Beispiele

/open
/closed

mit

```
chown root.open /open
chown root.closed /closed
chmod 770 /open
chmod 770 /closed
```

sind die Verzeichnisse dem User root und den Gruppen open b.z.w closed zugeordnet und durch das chmod dürfen nur User=root und Gruppe=closed auf /closed zugreifen.

Nun ist es so, dass beim mounten eines NFS-Laufwerks diese Rechte nicht zwangsweise mit übergeben werden (je nach NFS-Version und Konfiguration).
NFS-Laufwerke werden dann mit USER=nobody und Gruppe=nobody eingebunden.

NFS (ACHTUNG! Nur Version 3) kann aber bei der Übergabe sicherstellen, dass eine bestimmte User-ID bz.w Gruppen-ID für nobody zu setzen ist.

Auf dem NFS-Server (meinem Storage) lässt sich in der Datei ***/etc/exports*** die Rechtevergabe erzwingen:

```
/closed 192.168.10.0/24(rw,async,no_root_squash,no_subtree_check,anonuid=0,anongid=5001)
/open 192.168.10.0/14(rw,async,no_root_squash,no_subtree_check,anonuid=0,anongid=5000)
```

```
anonuid=0 erzwingt den User mit der ID=0 (das ist immer root)
anongid=5000/5001 erzwingt die Gruppe open b.z.w closed
```

So ist sichergestellt, dass ein „*mount*“ mit „*nobody*“ auch wirklich die User und Gruppenrechte zugewiesen werden.
Voraussetzung ist aber, dass die ID`s auch bekannt sind, was wiederum durch die Anbindung an den NIS-Server sichergestellt ist.

Alle Server und Clients, die diese Laufwerke einbinden sollen, müssen ihre Rechte vom NIS-Server beziehen.

That`s it.

Anmerkungen:

Um auf CentOS NFS zu zwingen in der Version 3 zu laufen kann man folgenden Eingriff vornehmen:

/etc/sysconfig/nfs das Rem (#) entfernen.

```
RPCNFSDARGS="-N 4"
```

Den NFS-Server danach neu starten.