ACL-Syntax (Compute-Knoten, Storage-Knoten)

Syntax (Compute-Nodes, Storage-Knoten)

Auf den Compute-Nodes und den Storage-Knoten kommen die Tools

NFSv4 ACLs (Compute Nodes)

```
mmeditacl
mmputacl
mmgetacl
mmdelacl
```

zum Einsatz, die sich in ihrer Verwendung von den Tools auf dem Login-Knoten grundlegend unterscheiden. Die unterstützen Berechtigungen und Regelformen sind natürlich identisch zu der oben beschriebenen Variante. Allerdings wird statt einer auf flags basierenden Syntax eine tabellenbasierte Form verwendet:

mmgetacl Syntax

```
special:owner@:rwxc:allow
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
(-)DELETE    (X)DELETE_CHILD (X)CHOWN        (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

user:Nutzername:--x-:allow:FileInherit:DirInherit
(-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (X)READ_ACL  (-)READ_ATTR  (-)READ_NAMED
(-)DELETE    (-)DELETE_CHILD (-)CHOWN        (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED
```

Je drei Zeilen bilden gemeinsam eine Regel. In der ersten Zeile wird zunächst der Zielnutzer / die Zielgruppe spezifiziert: `special:owner@` bezeichnet beispielsweise den "Spezialnutzer" `owner@`, also den jeweiligen Eigentümer einer Datei. Analog dazu existieren `special:group@` oder `special:everyone@`. `user:Nutzername` betrifft den Nutzer mit dem entsprechenden Nutzernamen:

Nutzername	Bedeutung
special:owner@	Besitzernutzer der Datei
special:group@	Besitzergruppe der Datei
special:everyone@	Jeder Nutzer
user:Nutzername	Der Nutzer mit dem Namen "Nutzername"

Anschließend folgen 4 Zeichen, die die direkten Berechtigungen des jeweiligen Nutzers / der jeweiligen Gruppe repräsentieren. Diese müssen allerdings nicht manuell aktualisiert werden, eine Bearbeitung der Berechtigungen in den runden Klammern wie unten beschrieben genügt dabei völlig, die 4 Berechtigungen stellen nur eine Art Zusammenfassung da, die automatisch aktualisiert wird. Dabei existieren die folgenden Belegungen:

Berechtigung	Bedeutung
r	Lesen
w	Schreiben
x	Ausführen
c	ACLs bearbeiten

An dritter Stelle wird der Regeltyp angegeben. Hier sollte allow verwendet werden, da Berechtigungen über Erlaubnisse und nicht über Verbote definiert werden sollten.

Zuletzt folgen weitere Flags, die das Verhalten der Regel beeinflussen, z.B. die Vererbung auf Dateien und Unterordner. Hierbei existieren beispielsweise folgende Flags:

Flag	Bedeutung
FileInherit	Diese Regel beim Anlegen von Dateien vererben
DirInherit	Diese Regel beim Anlegen von Unterverzeichnissen vererben

...	...

Anschließend folgen zwei Zeilen mit den jeweiligen Berechtigungen für den angegebenen Nutzer. Die Berechtigungen sind in diesem Fall benannt und mit runden Klammern versehen. Ein X innerhalb der Klammer bedeutet, dass die jeweilige Berechtigung erteilt ist, bei einem - hat der entsprechende Nutzer / die entsprechende Gruppe die jeweilige Berechtigung nicht.

Verpflichtende Angabe eines EDITORs

Für die interaktiven Funktionen der Kommandos ist die Angabe eines Editors verpflichtend. Ist dies nicht der Fall, bricht das Kommando mit folgender Fehlermeldung ab:

```
mmeditacl: EDITOR environment variable not set
```

In diesem Fall einfach folgenden Befehl ausführen:

```
echo "export EDITOR=/usr/bin/nano" >> ~/.bashrc
```

Und anschließend die Verbindung zum HPC einmal trennen und wieder neu aufbauen. Dann sollte das Kommando funktionieren.

Ein paar Beispiele

Standard-ACLs

Standardmäßig werden keine Berechtigungen vererbt. Jede neue Datei / Unterordner wird standardmäßig mit den Standard-ACLs initialisiert, die wie folgt lauten:

Standard-ACLs

```
#NFSv4 ACL
#owner:luros101
#group:ngs-admins
special:owner@:rwxc:allow
 (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (X)DELETE_CHILD (X)CHOWN        (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

special:group@:r-x-:allow
 (X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (-)DELETE_CHILD (-)CHOWN        (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED

special:everyone@:r-x-:allow
 (X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (-)DELETE_CHILD (-)CHOWN        (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED
```

Dateien / Ordner mit einzelnen Nutzern freigeben

ACLs können sehr gut dazu verwendet werden, um bestimmte Dateien oder Ordnern mit bestimmten Nutzern / Nutzergruppen freizugeben. Eine vollständige ACL für einen Ordner mit zusätzlichen Berechtigungen für einen weiteren Nutzer und Vererbung für zukünftige Unterordner / Dateien könnte beispielsweise so aussehen:

ACL Nutzerfreigabe

```
#NFSv4 ACL
#owner:luros101
#group:ngs-admins
special:owner@:rwxc:allow:FileInherit:DirInherit
 (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (X)DELETE_CHILD (X)CHOWN        (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

special:group@:rwx-:allow:FileInherit:DirInherit
 (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (X)DELETE_CHILD (-)CHOWN        (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED

special:everyone@:rwx-:allow:FileInherit:DirInherit
 (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (X)DELETE_CHILD (-)CHOWN        (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED

user:EigenerNutzername:rwxc:allow:FileInherit:DirInherit
 (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (X)DELETE    (X)DELETE_CHILD (X)CHOWN        (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

user:NeuerNutzername:rwx-:allow:FileInherit:DirInherit
 (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR  (X)READ_NAMED
 (-)DELETE    (X)DELETE_CHILD (-)CHOWN        (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED
```

Bestehende Regeln anpassen!

Beim Hinzufügen von Regeln mit Vererbung (`FileInherit:DirInherit`) wie in diesem Beispiel erforderlich, ist es wichtig, dass auch die bestehenden Regeln um die Vererbungs-Flags `FileInherit:DirInherit` erweitert werden. Andernfalls werden nur die Berechtigungen für die hinzugefügten Nutzer vererbt, der Eigentümer der Dateien hat anschließend keinen Zugriff mehr.

Außerdem muss neben der Regel für den neuen Benutzer ebenfalls eine Regel für den eigenen Benutzer hinzugefügt werden. Normalerweise ist der eigene Benutzer über `OWNER@` abgedeckt, wenn aber andere Nutzer in dem jetzt freigegebenen Verzeichnis Dateien anlegen, sind diese der Eigentümer (`OWNER@`) der neuen Dateien, und für den eigenen Benutzer existiert dann keine Regel mehr.