smartlearn
virtual learning. real success.

MODUL 143

M143 – AB04

Gerhard.Beutler

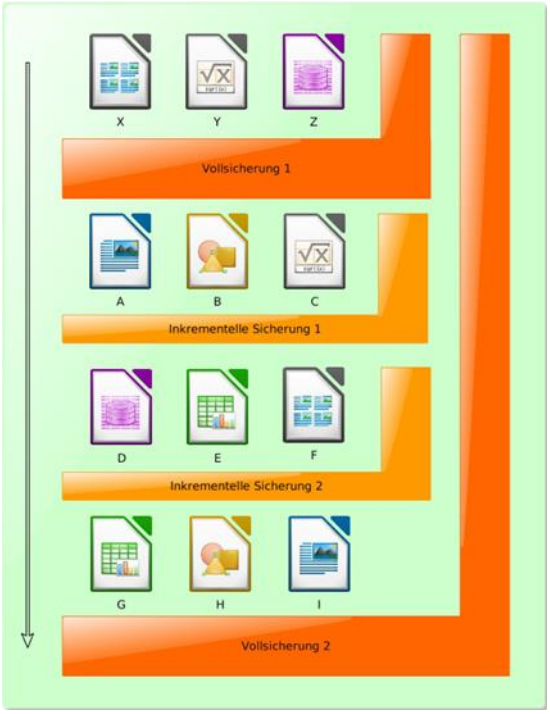
1 Sicherung 3: Inkrementelle Sicherung / Verifikation

Lernziele

- Inkrementelle Sicherung beschreiben, unterscheiden und erkennen.
- Vor-, Nachteile und Unterschiede der Inkrementellen zu Voll- und Differenzieller Sicherung erklären.
- Mit rsync eine inkrementelle Sicherung praktisch umsetzen.
- Vorgehensweise mit rsync verstehen.
- Datenfehler über Prüfsummen auffinden.
- Dateibäume überblicken und einschätzen.

1.1 Theorie: Inkrementelle Sicherung

Nach der relativ einfachen Vollsicherung und der etwas komplexeren differenziellen Sicherung wird es nun Zeit, die komplexeste Sicherungstechnik kennenzulernen: die inkrementelle Sicherung.



Hier wird das Konzept der differenziellen Sicherung weitergeführt. Wieder wird jeweils nur das gesichert, was sich verändert hat. Als Referenz wird aber nicht die jeweils letzte Vollsicherung genommen, sondern direkt die letzte «Sicherung». (Mit Ausnahme des ersten Inkrements: Das verweist auf die vorangehende Vollsicherung.)

Dadurch entsteht eine Kette kleiner Sicherungen, welche am Schluss bei einem Restore wieder zusammengeführt werden müssen. Der Vorteil besteht darin, dass nur die neusten Änderungen Platz im Backup belegen. Es gibt keine Redundanz in den Sicherungen.

Zeichnen Sie ein, welche Daten-Änderungen in welchem Backup enthalten sind:

	X	Y	Z	A	B	C	D	E	F	G	H	I
V1												
I1												
I2												
V2												

1.1.1 Inkrementelles Backup mit rsync

Nachdem eine erste Vollsicherung vorhanden ist, werden nur noch Inkremente - also «Teile», welche sich verändert haben, gesichert. Damit rsync weiss, was sich verändert hat, muss dem Kommando mitgeteilt werden, wo die letzte Sicherung liegt. So kann rsync vergleichen, welche Daten verändert wurden: Es schaut im letzten Backup nach, ob eine Datei schon existiert, oder ob sie verändert wurde. Das letzte Backup wird über den Parameter --link-dest angegeben. Als Argument kann anfangs die erste Vollsicherung und später dann die jeweils letzte inkrementelle Sicherung übergeben werden.

root@vmLS5:~$

```
rsync - av # Archiv - & Verbose - Modus
# Ordner für letztes Backup zum Vergleichen :
--link-dest=/absolute/path/to/last/backup vmadmin@192.168.220.13:/home/vmadmin
# Quelle
inc_1   # Ziel
```

Aufgabe 1)

Verändern Sie nun einige Daten auf dem Applikationsserver und machen Sie eine inkrementelle Sicherung. Dafür können Sie einfach ein Bild mit touch «aktualisieren». Etwas «spannender»: Sie können auch einige Bilder mit dem Namen Ihres Unternehmens «taggen». Dieses Kommando schreibt in die linke obere Bildecke (Koordinaten 50/50) den Text «Lorraine Rollt»:

convert BILDNAME -annotate +50+50 ”Lorraine Rollt” BILDNAME

Erstellen Sie nun eine erste inkrementelle Sicherung in den Zielordner inc_1. Die übertragenen Dateien sollten von rsync aufgelistet werden. Prüfen Sie, ob nur die veränderten Daten gelistet sind!

Aufgabe 2)

Gehen Sie in den Backup-Ordner inc_1 und schauen Sie nach. Sie werden überraschenderweise nicht nur die übertragenen Daten sehen! Erklären Sie, warum dies so ist und wie sich dies auf den Speicherplatz des Backup-Servers auswirkt:

.....

.....

.....

.....

Aufgabe 3)

Ändern und erzeugen Sie wieder einige Daten. Schreiben Sie nun selbst das Kommando, um die nächste inkrementelle Sicherung (inc2) zu machen:

.....

.....

.....

Aufgabe 4)

Worin besteht auf Ebene des Kommandos bei rsync genau der Unterschied zwischen einer differenziellen und einer inkre- mentellen Sicherung? Beschreiben Sie den Unterschied kurz in Worten:

.....

.....

.....

Reflexion

Überlegen Sie sich ein Beispiel aus Ihrem Betrieb, wo Sie rsync einsetzen könnten, oder schon (etwas ähnliches) einsetzen.

.....

.....

.....

Sehen Sie Limitierungen von rsync?

.....

.....

.....

Schon fertig mit rsync? Sichern Sie das komplette /etc des Applikationsservers!
Aber Achtung: Es folgen noch drei weitere Seiten zum Thema Verifikation eines Backups!

1.2 Verifikation Backup

Ein Backup ist erst einmal einfach ein «Klumpen» an Daten. Von «aussen» sieht man eigentlich nicht mehr als einen Ordner (rsync) oder einen Dateinamen (tar). Manchmal möchte man mehr Informationen haben als das. Man möchte z.B. die Integrität des Backups prüfen. Was heisst das?

Integrität einer Sicherung prüfen: Sie wollen sicherstellen, dass ein Backup auch wirklich das Backup ist, welches Sie früher angelegt haben. Oder Sie wollen sicherstellen, dass der Inhalt des Backups «gültig» ist und demjenigen der Quelle entspricht.

Warum ist das wichtig? Einige mögliche Gründe:

- Die Beschriftung ist falsch / Das Backup wurde verwechselt.
- Es hat sich durch Datentransfer, äussere Einflüsse oder Fehlkonfiguration ein Fehler im Backup eingeschlichen.
- Ein «Angreifer» hat das Backup absichtlich manipuliert, um das System bei einem Restore zu kompromittieren.

Die Art und Weise, wie sich ein Backup verifizieren lässt, hängt davon ab, in was für einem Format das Backup angelegt ist.

- **Binär:** Der Zustand von binären Daten wie z.B. tar und zip-Archive oder auch iso-Dateien lässt sich mit ihrer «Hash-Summe» festhalten.
- **Dateibäume:** Um den Zustand eines Dateisystems mit einem anderen zu Vergleichen bieten sich eigene Skripts oder diverse Tools aus dem Internet an. Dies bietet sich für Backups von rsync an.
- **Proprietär:** Bei eingekaufter Software ist man normalerweise auf die Unterstützung des Herstellers angewiesen, da die Formate meist nicht offen vorliegen.

1.2.1 Integritätsprüfung mittels Hash- / Prüfsumme

Hashsumme: (Deutsch: Prüfsumme) Was ist eine Hashsumme und warum eignet sich diese zum Prüfen von Daten? Aus dem Englischen kann to hash mit «zerhacken» übersetzt werden. Eine Hashsumme steht also für «zerhackte» Daten. Es ist etwa so, als ob eine Datei durch den Fleischwolf gedreht würde: Übrig bleibt ein heilloses Durcheinander - der Hash. Ein Beispiel einer MD5-Summe eines Passwortes:

```
echo 'meinPasswort' | md5sum
6993468f79db35810a3ce10a602be8d7 -
```

In der Informatik hat diese Summe spezielle Eigenschaften, welche sie sehr wertvoll machen:

- Der gleiche Input (Datei) ergibt immer den gleichen Output (Hashsumme).
- Die Hashsumme hat eine gleichbleibende definierte Länge.
- Es ist «sehr selten» der Fall, dass zwei verschiedene Inputs (Dateien) gleiche Outputs (Hashsummen) generieren (das nennt man «Kollision»).
- Ändert sich der Input (Datei) auch nur in einem einzigen Bit, kommt ein komplett anderer Output (Hashsumme) zustande.

Diese Eigenschaften erlauben es, dass eine Hashsumme eine Art «Fingerabdruck» einer Datei sein kann. Wie Personen auch, kann man dann Dateien anhand dieser viel kürzeren Summe prüfen. Hat sich eine Datei verändert, muss sie auch eine andere Summe haben. (Hash-Kollisionen sind Ausnahmen, dies ist für sicherheitskritische Anwendungen relevant, wir lassen es aber weg)

Aufgabe 5)

Holen Sie sich die Datei m143-07_text_achive.tgz vom Share auf den Linux-Desktop. Entpacken Sie das Archiv. Sie sollten nun die folgende Struktur sehen:

text_archive/

texte_backup2013.tar
texte_backup2014.tar
texte_backup2015.tar
texte_backup2016.tar
texte_backup2017.tar
texte_backup2018.tar
texte_prod.tar

Alle Backups sollen denselben Inhalt haben, da sich auf dem System nie etwas verändert hat. Überprüfen Sie mit Hilfe des Kommandos md5sum, ob alle Backups identisch sind!

Wie lautet das optimale Kommando, um alle Dateien zu vergleichen?

.....

Der Algorithmus von md5sum gilt nicht mehr als sicher. Theoretisch können Kollisionen erzeugt werden (andere Dateien mit gleichem Hash). Dies spielt für unsere Fälle keine grosse Rolle, dennoch sollte man wissen: Es gibt auch andere Hash- Algorithmen. SHA256 gilt beispielsweise als sicherer als MD5.

Aufgabe 6)

Vergleichen Sie MD5 und SHA256 gegeneinander! Zuerst soll eine grosse Datei mit Zufallsdaten (1 GB) erstellt werden:
`head -c 1G </dev/urandom >eingiga.bin`

Messen Sie nun die Zeit:

› `time md5sum eingiga.bin`

.....

› `time sha256sum eingiga.bin`

.....

Sie können auch die Datei unter Windows hashen! Probieren Sie es auf dem Host oder einer VM aus. Bekommen Sie dieselbe Hashsumme wie auf Linux? Wäre es egal, wenn nicht?

`CertUtil -hashfile eingiga.bin MD5`

.....

.....

1.2.2 Vergleich von Dateibäumen

Wie man mittels Hashsummen Änderungen in Binärdaten feststellen kann, wurde bereits behandelt. Wie aber soll man bei grösseren Dateibäumen die Übersicht behalten? Stellen Sie sich vor, Sie möchten:

- Sicher gehen, dass das Backup geklappt hat und die beiden Dateibäume gleich sind.
- Die Grösse oder Struktur eines Dateibaumes wissen.
- Wissen, ob sich irgendeine Datei verändert hat, ohne das Backupprogramm zu starten.


Die hier vorgestellten Kommandos «müssen» Sie für die Prüfung nicht können. Sie vereinfachen Ihnen aber möglicherweise das Lösen der Aufgaben. Oder sie helfen Ihnen sich in der Aufgabe zu orientieren

```
tar tf /path/to/archive.tar

tar tzf /path/to/archive.tar.gz
unzip -l /path/to/archive.zip
```

Zählen von Dateien:

```
find /path/to/dir | wc -l # Alles

find /path/to/dir -type
d | wc -l # Ordner find
/path/to/dir -type f |
wc -l # Dateien

find /path/to/dir -name \*.txt | wc -l # Bestimmtes
```

Grösse eines Ordners:

```
du -sh /path/to/dir
```

Grobe Übersicht über kleinere Ordner:

```
tree /path/to/dir
```

Vergleich zweier einzelner Text-Dateien auf Unterschiede:

```
diff /path/to/file1 /path/to/file2
```

Aufzeigen von Unterschieden in Dateien zweier Dateibäume:

```
diff -r /path/to/dir1 /path/to/dir2 # mit Details

diff -qr /path/to/dir1 /path/to/dir2 # ohne Details
```

Einzeiler-Gimmik für Hartgesottene: Finden von Unterschieden in Zeitstempel oder Dateirechten:

```
diff <(find app/data -printf '%f %m %t\n' | sort) \
     <(find full_1/data -printf '%f %m %t\n' | sort)
```

2 Wiederherstellung rsync

Lernziele

- Grundlegende Schritte einer Wiederherstellung nennen.
- Benennen welche Deltas/Sicherungen zur Wiederherstellung gebraucht werden.
- Restore einer verlinkten Sicherung manuell durchführen.
- Problematik von gelöschten Dateien im Backup-Vorgang formulieren.

2.1 Allgemeines Vorgehen bei einer Wiederherstellung

Unabhängig von der verwendeten Methode gibt es einige Punkte, welche beim Zurückspielen von Daten zu beachten sind.

- **System in konsistenten Zustand bringen:** Sie müssen sich zumindest überlegen, welche Applikationen vom Restore betroffen sind. Meistens ist es sicherer diese Applikationen zu stoppen und so das System in einen definierten Zustand zu bringen. Würden die Applikationen noch laufen, könnten diese den Restore beeinflussen. Es gibt aber auch Applikationen, welche für die Wiederherstellung im laufenden Betrieb konzipiert wurden.
- **Bereinigen:** Vor einem Restore sollten Sie sich überlegen, ob alte Daten bereinigt werden müssen. Sie können z.B. ein Datenverzeichnis erst komplett löschen, bevor das Backup eingespielt wird. So stellen Sie sicher, dass nur die Daten des Backups im Verzeichnis sind. Ansonsten besteht die Gefahr, dass noch «Datenleichen» vom zurückliegenden Crash auf dem System verbleiben und später zu Problemen führen.
- **Zurückspielen:** Sie sollten die Daten jeweils möglichst mit der gleichen Technologie zurückspielen, mit welcher Sie auch das Backup erstellt haben. Wenn Sie z.B. mit rsync die Sicherung machen, aber mit scp zurückspielen, könnte es zu Inkonsistenzen bei der Behandlung von Dateirechten, Symlinks und Ähnlichem kommen.
- **Applikationen starten:** Nach dem Backup sollten Sie die Applikationen wieder starten, damit diese die neuen Daten einlesen können. Ob dies nötig ist und wie lange das dauert, hängt von der Applikation ab.
- **Nachkontrolle:** Produktive Applikationen müssen nach dem Restore getestet werden! Es kann sogar sein, dass erst betriebsinterne Testprozesse durchlaufen werden müssen, bevor die Freigabe erfolgen kann.

2.2 Restore mit rsync

Wie Ihr Restore ausschaut hängt stark vom Szenario ab. Wenn Sie sich genau an die vorherigen Arbeitsblätter gehalten haben, sollten Sie nun mehrere Sicherungen des Verzeichnis /home/vmadmin des Applikationsservers auf dem Backupserver abgelegt haben.

Da wir mit rsync verlinkte Backups haben, reicht es, nur jeweils den Ordner der aktuellen Sicherung zurück zu spielen!

2.2.1 Kompletter Restore

Aufgabe 1)

Melden Sie den Benutzer `vmadmin` vom Applikationsserver ab. Löschen Sie nun als `root` das komplette Heimverzeichnis von `vmadmin` auf dem Applikationsserver mit `rm -Rf /home/vmadmin`.

Sehe Sie die Fotos noch im Share des Büro-PCs?

Aufgabe 2)

Stellen Sie das Verzeichnis mit `rsync` wieder her! Schreiben Sie alle Schritte auf, welche Sie unternehmen.

.....

.....

.....

2.2.2 Teilweiser Restore

Datenverlust tritt nicht nur wegen Sicherheitslücken oder Systemfehlern auf. Oft werden Daten durch Missgeschicke von internen und berechtigten Mitarbeitern gelöscht. Der Verlust einzelner Daten rechtfertigt meistens nicht einen kompletten Restore des ganzen Systems! Meistens genügt es einen einzelnen Ordner wieder herzustellen.

Aufgabe 3)

Vernichten Sie nun aus «versehen» die Fotos, z.B. indem Sie auf der Loro-Freigabe über den Büro-PC die Daten löschen. Stellen Sie nun lediglich den vermissten Ordner wieder her. Es gibt verschiedene Möglichkeiten wie Sie vorgehen können, entscheiden Sie sich selbst für ein Vorgehen! Schreiben Sie die Schritte auf, welche Sie machen und notieren Sie welche Backups benötigt werden.

.....

.....

2.3 Der Vorteil einer verlinkten Teilsicherung

Wir haben gesehen, dass rsync mit dem Parameter --link-dest Teilsicherungen (inkrementell oder differenziell) mit älteren Backups verlinken kann. Der Vorteil ist klar:

- Die Teilsicherung kann zurückgespielt werden, als wäre es eine Vollsicherung.
- Die Teilsicherung belegt nur den Platz neu, der wirklich nötig ist.

Ein noch grösserer Vorteil blieb aber bis jetzt unbemerkt. Um ihn zu verstehen, müssen wir eine Sicherung ohne Verlinkung durchspielen.

2.4 Verhalten gelöschter Dateien bei --compare-dest

Wir hatten den Parameter --compare-dest bereits früher angeschaut. Er bewirkt, dass nur neue oder veränderte Dateien kopiert werden. Unveränderte Dateien werden nicht über Hardlinks abgebildet. Auf den ersten Blick scheint das Verfahren ganz praktikabel. Es birgt aber Gefahren.

Aufgabe 4)

Machen Sie auf dem Backupserver nochmals eine neue Vollsicherung. Verändern Sie danach Dateien auf der Samba-Freigabe, indem Sie einige löschen und andere umbenennen.

Führen Sie nun eine differenzielle Sicherung mit dem Parameter --compare-dest durch. Was stellen Sie fest betreffend die gelöschten und umbenannten Dateien?

. . . . . .

. . . . . .

Was ist der Effekt bei einem Restore der Differenziellen Sicherung? Probieren Sie es aus!

. . . . . .

. . . . . .

Beschreiben Sie nochmals konkret, was das Problem an den obigen Effekten ist. Je einmal für gelöschte Dateien und um- benannte Dateien.

. . . . . .

. . . . . .

Wir haben «Differenzen» gesichert, indem wir die neuen oder geänderten Dateien in einem Ordner ablegt haben. Für gelöschte Daten funktioniert dies nicht, diese können ja nicht abgelegt werden, sie sind nicht mehr vorhanden. Eine Löschung bleibt so im Backup «unsichtbar», bzw. verhält sich gleich wie eine unveränderte Datei. Dies hat zur Folge, dass gelöschte Dateien bei einem Restore wieder erscheinen würden! Das verhindert zwar, dass Daten, welche aus Versehen gelöscht wurden, für immer verloren sind, ist aber dennoch meistens nicht gewünscht! Es ist ja normalerweise legitim und gewollt, Dateien zu löschen.

Das Löschen einer Datei lässt sich nicht so einfach abbilden. Dazu würde ein zusätzliches Protokoll benötigt: Beispielsweise eine «Meta-Datei» in welcher die Pfade aller gelöschten Dateien vermerkt sind, ähnlich der Protokollsicherung bei einer Datenbank (Siehe Modul 141 letztes Semester).

Zusammengefasst: Gelöschte Dateien sind schwieriger als «Differenz» abzubilden als neue oder veränderte Dateien. Dies gilt insbesondere dann, wenn es um Backups auf Datei-Basis geht.

2.5 Repetitionsfragen zur differentiellen und inkrementellen Sicherung

Aufgabe 5)
Beantworten Sie die Repetitionsfragen.
› Der Chef möchte, dass ein Backup (nicht verlinkt) eingespielt wird, da ein System gehackt wurde. Sie finden die Festplatte mit der letzten Vollsicherung nicht mehr, haben aber alle inkrementellen Sicherungen. Können Sie die Daten wiederherstellen? Begründen Sie Ihre Antwort:

. . . . . .

. . . . . .

- Nennen Sie einen Vor- und Nachteil der (nicht verlinkten) Differenziellen- im Vergleich zur Vollsicherung:

. . . . . .

. . . . . .

- Sie haben folgende Sicherungen (nicht verlinkt, Namensschema: TYPE_YYYY-MM-DD):
  - FULL_2015-11-06, FULL_2016-01-01
  - DIFF_2015-11-20, DIFF_2015-12-04, DIFF_2015-12-18, DIFF_2016-01-15, DIFF_2016-02-05, DIFF_2016-02-19
    Ein Tag vor Silvester am 30.12.2015 versagt eine wichtige Applikation. Dies merken Sie aber erst Mitte Februar. Sie müssen das System auf den letztmöglichen korrekten Stand zurücksetzen.
    Welche Backup-Dateien benötigen Sie zur Wiederherstellung?

    . . . . . .

    . . . . . .

    Wie viele Tage wurden «verloren» durch den Restore (Tage, welche nicht durch das Backup wiederherstellbar sind)?

    . . . . . .
- Sie haben eine verlinkte inkrementelle Sicherung, alles auf demselben Dateisystem. Der Lehrmeister löscht aus Versehen den Ordner mit der letzten Vollsicherung vom Montag. Das Sekretariat möchte einen Restore der letzten inkrementellen Sicherung vom Mittwoch. Lässt sich das machen?

. . . . . .

3 LB1: Wiederholung und Vorbereitung

Lernziele

- - Ordner erstellen
- - Archiv entpacken
- - Korrupte Datei erkennen und löschen
- - Archiv erstellen .tgz und .zip
- - Daten über Netzwerk kopieren (scp)
- - lokale Vollsicherung (rsync)
- - entfernte Vollsicherung (rsync)
- - inkrementelle Sicherung (rsync)
- - Restore entfernte Sicherung (rsync)

Dieses Kapitel soll Sie in der Vorbereitung für die erste Leistungsbeurteilung (LB1) unterstützen.

In der LB1 wird ein von der Klasse erarbeiteter Spicker verwendet.

Aufgabe 1) Wichtig:

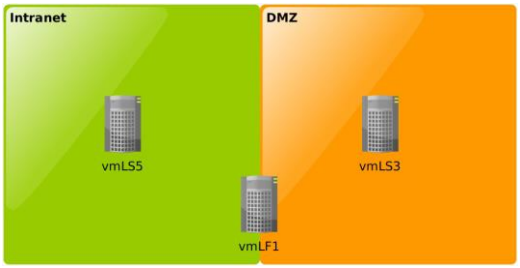
Erarbeiten Sie in der Klasse einen Spicker, den Sie eine Woche vor der Prüfung der Lehrperson zukommen lassen (Teams, Klassenshare, etc.)
Notizen und Input für den Spicker:

.....

.....

.....

Achtung: Auf der Prüfungsumgebung wir keine graphische VM vorhanden sein



Aufgabe 2)

Gehen Sie alle behandelten Kommandos für die Prüfung nochmals durch (Hinweis: Vorlage auf der nächsten Seite!). Probieren Sie diese auf dem System aus. Denken Sie daran, Wissen reicht nicht. Sie müssen es *können*!

Machen Sie gegebenenfalls nochmals die Übungen durch. Beachten Sie auch welche Optionen verwendet wurden.

Viel Erfolg bei den Vorbereitungen!

3.1 Fazit Backup-Methoden

Aufgabe 3)

Überlegen Sie sich zu jedem behandelten Kommando mögliche Einsatz-Zwecke, das Verhalten bezüglich Metadaten (Datei- rechte und User sowie Gruppen-Zuordnungen), Behandlung von Symlinks sowie Vor- und Nachteile. Es reicht, wenn Sie die Tools so bewerten, wie wir sie eingesetzt haben (manche Programme wie rsync haben noch viel mehr Einsatzmöglichkeiten, diese müssen aber nicht berücksichtigt werden).

	Einsatz / Zweck / Beispiel	Dateirechte	Symlinks	Vor-/Nachteile
cp				
scp				
rsync				
tar				
zip				

3.2 Fazit Hilfsprogramme

Aufgabe 4)

Überlegen Sie sich den Einsatzzweck, die Verwendungsart und wichtige Optionen der unten gelisteten Porgramme. Sie machen zwar selbst keine Backups, sind aber hilfreich beim Umsetzen.

	Einsatz / Zweck / Beispiel
md5sum	
sha256sum	
tree	
diff	
find	
ls	