

Copyright (C) Open Source Press

Philipp Storz

Bacula

Backup-Strategien und -Lösungen im Netzwerk

Copyright (C) Open Source Press

Alle in diesem Buch enthaltenen Programme, Darstellungen und Informationen wurden nach bestem Wissen erstellt. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grunde sind die in dem vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor(en), Herausgeber, Übersetzer und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht, auch nicht für die Verletzung von Patentrechten, die daraus resultieren können. Ebenso wenig übernehmen Autor(en) und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. werden ohne Gewährleistung der freien Verwendbarkeit benutzt und können auch ohne besondere Kennzeichnung eingetragene Marken oder Warenzeichen sein und als solche den gesetzlichen Bestimmungen unterliegen.

Dieses Werk ist urheberrechtlich geschützt. Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches – oder Teilen daraus – vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlags in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Bibliografische Information Der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Copyright © 2012 Open Source Press, München
Gesamtlektorat: Dr. Markus Wirtz
Satz: Open Source Press (L^AT_EX)
Umschlaggestaltung: Olga Saborov, Open Source Press
Gesamtherstellung: Kösel, Krugzell

ISBN 978-3-941841-41-3

<http://www.opensourcepress.de>

Inhaltsverzeichnis

Vorwort von Kern Sibbald	17
Vorwort des Autors	21
1 Einleitung	23
1.1 Die Bacula-Architektur	24
1.1.1 Zusammenspiel der Komponenten	25
1.1.2 Eigenschaften von Bacula	27
1.1.3 Die Entwicklung von Bacula	28
1.2 Über dieses Buch	29
1.2.1 Kapitelübersicht	29
1.2.2 Inhalt der beiliegenden DVD	30
1.3 Sprachgebrauch und Terminologie	31
2 Installation	33
2.1 Import der Appliance von DVD	36
2.1.1 Installation der Virtualisierungssoftware	37
2.1.2 Import und Start der virtuellen Maschine	40
2.2 Datenbankauswahl und -installation	44
2.2.1 Paketbasierte Installation der Datenbanksysteme	45
2.3 Paketbasierte Installation von Bacula	46
2.3.1 Updates und Versionskompatibilität	46
2.3.2 Pakete aus dem dass IT Bacula Repository	47
2.3.3 Pakete von der Webseite des Bacula-Projekts	49
2.3.4 Pakete der eingesetzten Distribution	50
2.4 Bacula aus dem Quellcode übersetzen und installieren	50

2.4.1	Source-RPM auf der eigenen Distribution bauen	51
2.4.2	Berechtigungen der Bacula-Daemons	52
2.4.3	Quellcode manuell übersetzen	53
2.5	Datenbankeinrichtung	57
2.5.1	Allgemeines zur Datenbankeinrichtung	57
2.6	Installation grafischer Programme zu Bacula	59
2.6.1	Bat	59
2.6.2	Webbasierte Oberflächen	60
3	Erste Schritte	61
3.1	Die Bacula-Appliance	62
3.2	Läuft Bacula?	62
3.2.1	Dienste überprüfen	62
3.2.2	Prüfung mit Bacula-Bordmitteln	65
3.3	Erste Schritte	71
3.3.1	Erste Sicherung	71
3.3.2	Erste Rücksicherung	74
3.4	Konfiguration für produktive Sicherung auf Festplatte	78
3.4.1	Änderung des Speicherorts für die Bacula Volumes	79
3.4.2	Änderung der zu sichernden Dateien	80
3.4.3	Integration der Appliance in das Netzwerk	86
3.4.4	Konfiguration des Netzwerks	87
3.4.5	Anpassung der Adressen in Bacula	87
4	Backup-Techniken und -Konzepte	89
4.1	Sicherungsarten	90
4.1.1	Vollsicherung	90
4.1.2	Differentielle Sicherung	91
4.1.3	Inkrementelle Sicherung	91
4.2	Sicherungsintervalle und Vorhaltezeit	92
5	Konfiguration von Bacula	97
5.1	Die Bacula-Eigensicherung	97
5.2	Allgemeines zu Bacula-Konfigurationsdateien	98
5.3	Die Bacula-Ressourcen	102

5.3.1	Director Daemon-Konfiguration – bacula-dir.conf	102
5.3.2	Storage Daemon-Konfiguration – bacula-sd.conf	113
5.3.3	File Daemon/Client-Konfiguration – bacula-fd.conf	115
5.3.4	Bacula Console-Konfiguration – bconsole.conf	116
5.4	Hinzufügen eines neuen Clients/File Daemons	116
5.4.1	Installation der Client-Software	117
5.4.2	File Daemon-Installation	117
5.4.3	File Daemon konfigurieren	118
5.4.4	File Daemon in der Director-Konfiguration	120
5.4.5	Verbindung zwischen Director und File Daemon prüfen	121
5.4.6	Zusammenspiel von Passwörtern und Namen	121
5.4.7	Konfigurieren eines FileSets für den neuen Client	123
5.4.8	Konfigurieren eines Jobs für den neuen Client	124
5.4.9	Überprüfen der Client-Anbindung	124
5.5	Umsetzung einer Mehrpool-Sicherungsstrategie	125
5.5.1	Sicherungsart und Intervall	126
5.5.2	Vorhaltezeit	127
5.5.3	Simulation unseres Sicherungskonzepts	129
5.6	Wartung der Katalog-Datenbank und Medienverwaltung	130
5.6.1	Vorhaltezeiten und automatisches Medienmanagement	130
5.6.2	Wie ermittelt Bacula das nächste Medium?	136
5.6.3	Lebenszyklus eines Bacula-Volumes	138
6	Backup-Hardware	141
6.1	Sicherung auf Festplatten	142
6.2	Tapelibrary in Bacula integrieren	143
6.2.1	Bandlaufwerke und Medienwechsler – Grundlagen	143
6.2.2	Bandlaufwerke und Medienwechsler in der Praxis	146
6.2.3	Bandlaufwerke und Medienwechsler mit Bacula	152
6.2.4	Testen mit btape	153
6.2.5	Data Spooling	161
6.2.6	Nutzung eines Autochangers mit Bacula	162
6.3	Einzelnes Bandlaufwerk mit Bacula nutzen	164
6.4	Sicherungskonzept für Einzelaufwerk	166

7	Bedienung von Bacula	171
7.1	Befehle mit Argumenten	172
7.1.1	Interaktive Angabe von Argumenten	172
7.1.2	Automatisches Befüllen von Argumenten	173
7.1.3	Direkte Angabe von Argumenten	173
7.1.4	Tab-Completion	173
7.1.5	Automatische Ergänzung bei unvollständiger Eingabe	174
7.2	Befehle der Bacula Console	175
7.2.1	Statusinformationen	175
7.2.2	Jobsteuerung	185
7.2.3	Medienverwaltung	195
7.2.4	Konfiguration	199
7.2.5	Sonstiges	202
8	Fortgeschrittene Anwendungen	207
8.1	FileSet im Detail	207
8.1.1	Angabe von Dateien und Verzeichnissen	208
8.1.2	Zu sichernde Dateien bestimmen: Include und Exclude	209
8.1.3	FileSet-Optionen	209
8.1.4	Dynamische Erzeugung	218
8.1.5	Veränderte FileSets erkennen	220
8.1.6	Wie verarbeitet Bacula die Include/Exclude-Direktiven?	221
8.1.7	Windows	222
8.1.8	Alle lokalen Dateisysteme sichern	223
8.1.9	Beispiele	227
8.2	Run-Direktive im Detail	229
8.2.1	Zeitangaben	230
8.2.2	Job Overrides	232
8.3	Befehle vor, während und nach der Sicherung	234
8.3.1	RunScript-Ressource	235
8.3.2	RunScripts unter Windows	237
8.4	Messages-Ressource im Detail	239
8.4.1	Direktiven zum Nachrichtenversand per E-Mail	240
8.4.2	Definition von Nachrichtenzielen	241

8.4.3	Nachrichtenziele im Überblick	243
9	Fortgeschrittene Konfiguration	245
9.1	Bacula skalieren	245
9.1.1	Mehrere Jobs gleichzeitig ausführen	245
9.1.2	Mehrere Jobs auf ein Medium schreiben	248
9.1.3	Mehrere Bandlaufwerke in einem Autochanger	249
9.2	backup-2-disk-2-tape / Migration / Copy Jobs	255
9.2.1	Warum Migrationen?	257
9.2.2	Einschränkungen bei Migrationen und Copy Jobs	257
9.2.3	Konfiguration von Migrationen und Copy Jobs	257
9.2.4	Migrationen in der Praxis	259
9.2.5	Copy Jobs	263
9.2.6	Copy Jobs in der Praxis	263
9.3	Erweiterte Jobsteuerung	267
9.3.1	Umgang mit Duplicate Jobs	267
9.3.2	Beschränkung der Laufzeit von Jobs	268
9.4	Backupstrategie für Laptops	269
9.5	Accurate Backup	271
9.5.1	Warum Accurate Backup?	271
9.5.2	Beispiel für Probleme ohne Accurate Backup	273
9.5.3	Wie arbeitet Accurate Backup?	275
9.5.4	Konfiguration von Accurate Backup	275
9.6	Daten verschlüsseln	277
9.6.1	Erzeugen der benötigten Schlüssel	279
9.6.2	Konfiguration im File Daemon	280
9.6.3	Wiederherstellung mit dem Masterschlüssel	281
9.6.4	TLS – Authentifizierung, Transport- und Kommunika- tionsverschlüsselung	282
9.7	Berechtigungen für Benutzerschnittstellen mit ACLs	292
9.8	Virtual Full Backup	298
9.8.1	Virtual Full Backup in der Praxis	300
9.9	Deduplizierung – Base Jobs	306
9.9.1	Grundüberlegungen zu Base Jobs	307

9.9.2	Praxisbeispiel einer Sicherung mit Base Jobs	308
9.9.3	Zeitplanung bei Base Jobs	311
9.9.4	Funktionsweise von Base Jobs	311
9.9.5	Feineinstellung der Base Jobs	312
9.10	Verify Jobs	312
9.10.1	Einbruchserkennung mit Bacula	313
9.11	Sicherung durch Firewalls	316
9.11.1	Ports	317
9.11.2	NAT	317
9.11.3	Timeouts	318
9.12	Read-Only File Daemon	318
9.13	Kommandozeilenwerkzeuge	319
9.13.1	bwild	319
9.13.2	bregex	320
9.13.3	bcopy	321
9.13.4	bsmtp	321
10	Interne Abläufe	323
10.1	Netzwerkkommunikation	323
10.1.1	Portnummern	323
10.1.2	Ablauf der Kommunikation	324
10.1.3	Verteilung der Zieladressen	325
10.1.4	Wann erfolgt welche Kommunikation?	327
10.2	Datenbankschema	331
11	Erweiterte Anwendungen	337
11.1	Benutzerschnittstellen	337
11.1.1	Bacula Tray Monitor	337
11.1.2	Bat	338
11.1.3	mhvtl	339
11.2	Bacula Console mit Skripten automatisieren	340
11.3	Sicherung virtueller Maschinen	342
11.3.1	VMware	343
11.3.2	VMware Consolidated Backup	344

11.3.3 Citrix XenServer	354
11.4 Sicherung des Novell Filesystems NSS	362
11.5 Plugins	363
11.5.1 Konfiguration und Aktivierung	364
11.5.2 bpipe	365
11.5.3 Citrix-XenServer und bpipe	367
12 Notfallwiederherstellung	371
12.1 Vorkehrungen zum Schutz des Bacula-Systems	372
12.2 Wiederherstellungsansätze	372
12.3 Werkzeuge	373
12.3.1 bextract	373
12.3.2 bls	373
12.3.3 bscan	373
12.3.4 dbcheck	373
12.4 Durchführung einer Notfallwiederherstellung	374
13 Fehlersuche	377
13.1 Keine Berechtigung	377
13.2 Cannot label Volume because it is already labeled	378
13.3 No route to host	378
13.4 Unable to authenticate with xxx daemon	379
13.5 is waiting for Client xxx-fd to connect to Storage yyy	379
13.6 gethostbyname() for host xxx failed	381
Anhang	383
A Client-Installation	385
A.1 Microsoft Windows	385
A.1.1 Funktionsprüfung	390
A.1.2 Windows Firewall	391
A.2 Mac OS X	395
A.3 Solaris	398
B Variablenersetzung in Bacula	401
B.1 Variablenersetzung für Medien-Labels	401

B.2	Variablenersetzung in RunScripts	402
B.3	Variablenersetzung im Autochanger Command	403
B.4	Mount Command	403
B.5	Variablenersetzung in Mail und Operator Command	404
C	Direktiven und ihre Datentypen	405
C.1	Director-Konfiguration (bacula-dir.conf)	406
C.1.1	Catalog-Ressource	406
C.1.2	Client-Ressource	406
C.1.3	Console-Ressource	407
C.1.4	Counter-Ressource	408
C.1.5	Director-Ressource	408
C.1.6	FileSet-Ressource	410
C.1.7	FileSet Include/Exclude-Ressource	410
C.1.8	FileSet Options-Ressource	410
C.1.9	Job-Ressource / JobDefs-Ressource	411
C.1.10	Pool-Ressource	414
C.1.11	RunScript-Ressource	415
C.1.12	Schedule-Ressource	416
C.1.13	Storage-Ressource	416
C.1.14	Messages-Ressource	417
C.2	Console-Konfiguration (bconsole.conf)	417
C.2.1	Console-Ressource	417
C.2.2	Director-Ressource	418
C.3	File-Daemon-Konfiguration (bacula-fd.conf)	418
C.3.1	Client/File-Daemon-Ressource	418
C.3.2	Director-Ressource	420
C.3.3	Messages-Ressource	420
C.4	Storage-Daemon-Konfiguration (bacula-sd.conf)	421
C.4.1	AutoChanger-Ressource	421
C.4.2	Device-Ressource	421
C.4.3	Director-Ressource	423
C.4.4	Storage-Ressource	424
C.4.5	Messages-Ressource	425

C.5	Gemeinsame Konfigurationsressourcen	425
C.5.1	Messages-Ressource	425
C.6	Bacula-Typen	426
C.6.1	FileSet-Options	430
C.6.2	Befehle der Bacula Console	431
C.7	Modifikatoren für Datentypen	431

Copyright (C) Open Source Press

Copyright (C) Open Source Press

Für meine Familie

Vorwort von Kern Sibbald

Die Geschichte von Bacula begann Ende 1999, nachdem ich einige Jahre für das Apcupsd-Projekt (ein Kontrollprogramm für unterbrechungsfreie Stromversorgungen) gearbeitet hatte. Damals war ich bereit für ein neues, größeres Projekt, also begann ich nach Ideen zu suchen. John Walker, der Gründer von Autodesk, hatte einige Vorschläge; einer war ein Ersatz für das Programm NetWorker, das er für das Backup seiner Solaris-Rechner verwendete.

Seit Beginn meiner beruflichen Laufbahn als Programmierer, als ich an einem Backup-Programm für Großrechner arbeitete, interessierte mich die Idee eines Netzwerk-Backup-Programms, insbesondere weil es verschiedene interessante Technologien umfasst.

John schlug auch einige Namen für ein solches Programm vor. „Bacula“ klingt in den meisten Sprachen gut und weckt zahlreiche Assoziationen; so fiel die Wahl schnell darauf.

Das eigentliche Bacula-Design und -Coding begann im Januar 2000 in Neuchâtel, Schweiz, und wurde seit Juni 2000 bis heute in Lausanne, Schweiz, fortgeführt, wo ich derzeit lebe.

Da ein Backup-Programm in erster Linie robust und frei von schwerwiegenden Fehlern sein muss, war eine Menge Basis-Programmierarbeit zu leisten. Ich begann mit einem Set von Subroutinen zum Umgang mit verlorengegangenen Speicherinhalten, korrumpiertem Speicher, Low-Level-Networking, Buffer-Pool-Management usw. Dies nahm etwa ein Jahr in Anspruch. Ein weiteres Jahr brauchte ich für die drei Haupt-Daemons (Director, Storage Daemon und File Daemon) und die Datenbank zum Speichern der Backup-Informationen.

Im April 2002 funktionierte Bacula bereits ganz gut, und ich veröffentlichte den ersten Release auf Source Forge unter der GPL v2. Zu dieser Zeit fehlten noch eine Reihe grundlegender Features, so dass ich immer wusste, was noch zu tun war. Was ich in dieser Phase sehr genoss, war, erste Anwender bei Problemen (und Bugs) zu unterstützen.

Heute erscheint vieles im Design von Bacula selbstverständlich, aber im Januar 2000 war das nicht so. Eine der schwierigen Design-Entscheidungen

war die über die Datenbank für die Speicherung der Backup-Metadaten (Dateinamen, Dateigrößen, Speicherort in den Backup-Volumes etc.). Viele empfahlen mir, eine spezialisierte, für Bacula geschwindigkeitsoptimierte Datenbank zu programmieren, aber nach vielem Nachdenken fiel die Wahl eindeutig auf eine Open-Source-Datenbank-Engine.

Zu diesem Zeitpunkt (Mitte 2000) waren die verbreitetsten Datenbanken MySQL und SQLite, PostgreSQL kam etwas später. Der Hauptgrund für eine Open-Source-Datenbank war, den Nutzern direkten Zugriff auf die Daten mittels SQL zu gewähren. Meine Hauptsorge bestand darin, dass die Performance leiden könnte. Aber als die Bacula-Datenbanken größer wurden, haben wir immer Wege gefunden, dass die Datenbank-Engines mithalten konnten.

Eine weitere wichtige Design-Entscheidung bestand darin, Threads zu nutzen, statt mehrere Programme als Prozesse zu starten. Der Vorteil liegt in deutlich höherer Effizienz, allerdings muss der Programmierer sehr viel von Threads verstehen, um Deadlock-Situationen zu vermeiden.

Die Entscheidung, Bacula als Open-Source-Software unter der GPL (GNU General Public License) zu veröffentlichen, war ebenfalls von großer Bedeutung – auch wenn es keine Design-Entscheidung war. Dadurch entstand eine Community engagierter Anwender, die kontinuierlich Bugfixes beitrugen, Code für neue Features produzierten, Anregungen gaben und vor allem auch über Bugs berichteten. Dies alles hat Bacula zu einem extrem robusten und mit vielen Features ausgestatteten Programm werden lassen.

Im Laufe der Jahre haben sich diese und andere Entscheidungen als richtig erwiesen; das Projekt wuchs, die Zahl der Features und die Komplexität von Bacula nahmen zu, ohne dass Design-Probleme auftraten.

In Jahr 2006 wurde deutlich, dass mehr und mehr Firmen Bacula nutzten, doch sein Einsatz war nicht optimal, weil Professional Services fehlten, die die meisten größeren Firmen benötigen. Dies führte folgerichtig zur Gründung eines Unternehmens, um diese Dienste mithilfe von Bacula-Experten anzubieten. Für mich war dies auch ein Weg, eine schnellere Bacula-Entwicklung zu finanzieren und die Fortsetzung des Projekts langfristig zu sichern.

Mein erster Kontakt zu Philipp Storz war zu Beginn des Jahres 2007, als er darum bat, seine Firma das IT auf der Bacula.org-Website als Dienstleister für Professional Services rund um Bacula aufzuführen. Philipp hatte schon einige Jahre Erfahrung mit Bacula. Später in diesem Jahr stieß Philipp in Basel zu uns, als sich das Gründer-Team von Bacula Systems traf. Wir diskutierten Möglichkeiten, wie die (noch nicht gegründete) Bacula Systems mit unabhängigen Partnern wie das IT zusammenarbeiten könnte. Es war unser erster Kontakt zu einem Bacula-Partner – noch bevor Bacula Systems gegründet war.

Die Firma Bacula Systems an den Start zu bringen, dauerte ziemlich lang, weil es nicht einfach war, die richtigen Leute zu finden, die Zeit und Geld in ein Unternehmen investieren wollten, bei dem der Quellcode frei verfügbar war. Doch Anfang 2008 war das Team vollständig, und Bacula Systems wurde in der Schweiz am 1. Juli 2008 offiziell gegründet. Acht Gründungsmitglieder investierten in das Unternehmen und arbeiteten ohne Bezahlung, um die Firma aufzubauen. Bacula Systems ist seitdem beträchtlich gewachsen, mehr als die Hälfte der Gründer ist bei der Firma angestellt, und die Zukunft sieht vielversprechend aus.

Es ist schwer vorherzusagen, wie sich Bacula in den nächsten fünf oder zehn Jahren entwickeln wird, aber es scheint sicher, dass Bacula bestehen bleibt und dass noch einige High-End-Features gebraucht werden. So benötigen viele große Unternehmen spezielle Module (Plugins), um Oracle zu sichern und wiederherzustellen. Daran arbeitet Bacula Systems derzeit. Weitere Projekte sind: Optimierung des Datentransfers über Kommunikationsverbindungen durch den Einsatz von Deduplizierung, Hinzufügen von VMware-Backup mit Integration von vSphere, Backup von KVM Virtual Machines, mehr grafische Benutzeroberflächen etc. Solange die Informationstechnologie ihre rasante Entwicklungsgeschwindigkeit behält, wird es immer Bedarf nach weiteren Features und Fähigkeiten in Bacula geben.

Philipp war nicht nur ein früher Bacula-Anwender und der erste Reseller von Bacula Systems, er ist auch der Erste, der ein Buch über Bacula schreibt und veröffentlicht. Es fügt sich bestens, dass jemand mit Philipps Erfahrung das tut. Es freut mich außerordentlich zu wissen, dass bald Menschen dieses Buch in Händen halten werden, die Bacula bereits nutzen oder es nutzen wollen, und ich freue mich darauf, es in den Bücherregalen zu sehen.