Concepts and Architecture

Backup

Author(s): M.Jerger
Version 0.5 on 13.02.2015
State: valid

Changehistory

Version	Author / Actor	Date	Change / Activity	State
0.1	M,Jerger	14.11.14	initial	draft
0.2	M.Mörike, M.Caspari, T.Scherer	12.12.14	Reviewed	valid
0.3	M.Jerger	22.12.14	Described Solution Architecture and refactored document structure.	valid
0.4	M.Jerger	28.01.15	Reviewed for consistency, added restore	valid
0.5	meissa	30.01.15	Reviewed and translated.	valid

Legend

New State	Consequence
draft	Content not yet complete according to the authors.
valid	Content is complete and valid according to the authors.
verified (with → responsible)	The content is also valid in the opinion of other people
approved (by → responsible)	The client / responsible declares the content to be valid.

1 Document Scope

1.1 Target and Objective

This document describes backup concepts on an example company.

1.2 Intended Reader

Software Operating, Architects, Developer and PO.

1.3 Additional Documents

1. BSI Datensicherung: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/baust/b01/b01004.html
2. HostEurope Virtual Server Offer: https://www.hosteurope.de/Server/Virtual-Server/
3. https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/itgrundschutzkataloge_node.html

2 Requirements Analysis

2.1 Types of Backup Data

1. Application Code: Code for all individual application installations.
2. Application Data: Data of installed and used applications.
3. Configuration: Configuration of components or applications.
4. Log Data: Logfiles for all running applications.
5. Security Log Data: For secured systems, there are special security log files.

2.2 Backup Data De-Duplication

Although there are many types in between we will distinguish between the two major types here:

1. full backup or
2. incremental backup

Due to the small backup sizes at Example Company, we will use full backup only.

2.3 Relevant Questions to answer

Relevant questions with relation to backup are

1. Access-Security: Who has access to backups?
2. Computer Center Outage: What happens, if the whole computer center fails?
3. Protection Requirements: How sensible are backup data?
4. Recovery: Recovery describes the context for rescuing applications, their data and configuration in case of disaster. So the question is: What's the duration for recovery?

2.4 Specialties at Log File Backup

2.4.1 Further Analysis Options on Log Backups

Backup allows some special analytics, for example, log files can be analyzed for not authorized changes.

2.4.2 Logdata handling on ubuntu12.04

Current Name	Frequency	Compress	Generations	Uncompressed Name
auth.log	2d	1+	6	auth.log.0
syslog	1d	1+	6	syslog.0
apache*	1w	1+	52	access.log.1
catalina.out	1w	0+	52	--

2.5 Specialties at HostEurope

The Example Company is hosted at HostEurope. According to HostEurope product specification, the Example Company has the following backup (options) available:

- Snapshot: stores the whole server (application, configuration, data) – the recovery takes about 2-5h. The snapshots are stored for three months.
- Permanent Snapshot: Like snapshots, but with unlimited storage duration.
- File system backup: On daily backup basis, stored of the last 14 days.

3 Backup Decisions

3.1 Global Decisions

3.1.1 Data Types to Backup

We backup application data and log data.

Code and configuration needs no backup, because code is saved by Version Management System.

3.1.1.1 Application Data

3.1.1.2 Log Data

Logfiles are synchronized daily.

Storage duration is:	1 Year

3.1.1.3 Security Log Data

Security logs need no backup, because they're synchronized in real time.

Storage duration is:	1 Year

3.1.2 Availability

3.1.2.1 High

High important data are stored on another computing center.

3.1.2.2 Normal

Normal important data are stored on another server.

3.1.2.3 Low

Low important data are stored only on the same server and on the hosting providers backup store.

3.1.3 Decisions and Analysis per Application

ID	Measurement
Application name	
Current application data size	
Estimated application data growth for upcoming year	
Log data growth / year	
App data backup on SourceSystem	
Generations in daily interval	
Generations in weekly interval	
Generations in monthly interval	
App data backup on SinkSystem	
Generations in daily interval	
Generations in weekly interval	
Generations in monthly interval	
Application needs	
App data importance / availability	
App data confidentiality	
Log data confidentiality	
Time for disaster recovery	

3.2 Decision per Application

3.2.1 Example Portal Server

ID	Measurement
Application name	Size
Current application data size	3,5G
Estimated application data growth for upcoming year	500M
Log data growth / year	250M
App data backup on SourceSystem	
Generations in daily interval	1 (14 by HostEurope)
Generations in weekly interval	0
Generations in monthly interval	0
App data backup on SinkSystem	
Generations in daily interval	2
Generations in weekly interval	
Generations in monthly interval	
Application needs	
App data importance / availability	high
App data confidentiality	normal
Log data confidentiality	normal
Time for disaster recovery	1 day

3.2.1.1 On System Space

13 * 4G = 52G

3.2.1.2 Remote Space

35 * 4G = 140G

3.2.2 Example Owncloud Server

ID	Measurement
Application name	Size
Current application data size	15G
Estimated application data growth for upcoming year	5G
Log data growth / year	50M
App data backup on SourceSystem	
Generations in daily interval	1 (14 by HostEurope)
Generations in weekly interval	0
Generations in monthly interval	0
App data backup on SinkSystem	
Generations in daily interval	1
Generations in weekly interval	
Generations in monthly interval	
Application needs	
App data importance / availability	low
App data confidentiality	high
Log data confidentiality	normal
Time for disaster recovery	5 days

3.2.2.1 On System Space

3 * 15G = 45G

3.2.2.2 Remote Space

12 * 15G = 180G

3.2.3 Example CRM Server

ID	Measurement
Application name	Size
Current application data size	50M
Estimated application data growth for upcoming year	10M
Log data growth / year	5M
App data backup on SourceSystem	
Generations in daily interval	1 (14 by HostEurope)
Generations in weekly interval	52
Generations in monthly interval	-
App data backup on SinkSystem	
Generations in daily interval	1
Generations in weekly interval	-
Generations in monthly interval	-
Application needs	
App data importance / availability	high
App data confidentiality	normal
Log data confidentiality	normal
Time for disaster recovery	5 days

3.2.3.1 On System Space

12 * 50M = 1G

3.2.3.2 Remote Space

23 * 50M = 1G

3.2.4 Example Ticket System

ID	Measurement
Application name	Size
Current application data size	2M
Estimated application data growth for upcoming year	100K
Log data growth / year	100K
App data backup on SourceSystem	
Generations in daily interval	1 (14 by HostEurope)
Generations in weekly interval	52
Generations in monthly interval	-
App data backup on SinkSystem	
Generations in daily interval	1
Generations in weekly interval	-
Generations in monthly interval	-
Application needs	
App data importance	low
App data confidentiality	normal
Log data confidentiality	normal
Time for disaster recovery	5 days

3.2.4.1 On System Space

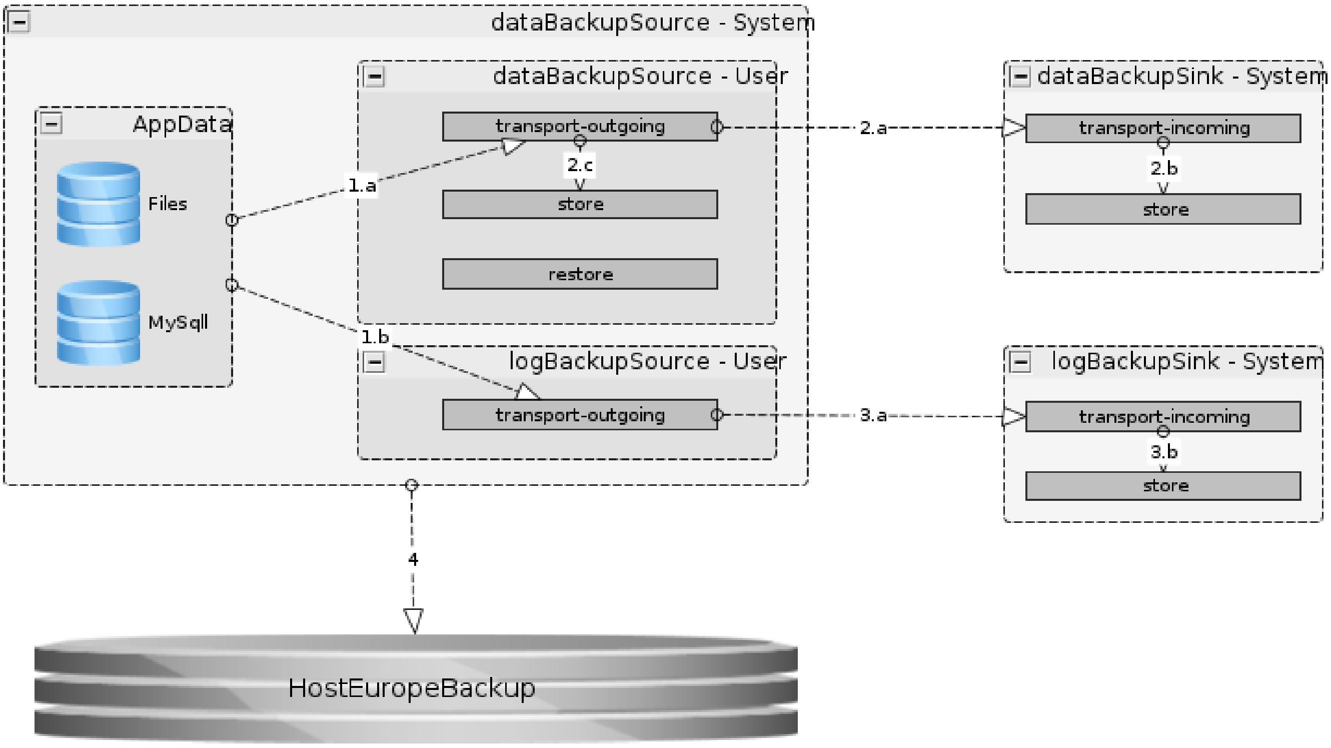
12 * 50M = 1G

3.2.4.2 Remote Space

23 * 50M = 1G

4 Solution Architecture

4.1 Backup Process Steps



4.1.1 Backup Source

4.1.1.1 Backing up (1)

In the backup step, a source system cron job will

1. collect all application (1.a) and log (1.b) data.
2. deliver this data to the “Transport Handover Point”
3. handle the “previous transport failed” case (send a mail).

4.1.2 Backup Data Transport (2)

4.1.2.1 Do the Transport (2.a)

In the transport step a sink system cron job will

1. do the transport (2.a): using ssh and rsync. For ssh, the sink system is authorized on dataBackupSource user.
2. (optional) verify correctness: Done by hash comparison.
3. move to Sink-Store (2.b): Moves the received backup to Sink-Store.
4. handle Sink generations: Deletes the eldest backup up to the number of the defined generations to be preserved.
5. move to Source-Store (2.c): Moves the received backup to Source-Store.

6. handle Source Generations: Deletes backups, bailing out of the defined generations to be preserved.

4.1.3 Backup Log Transport (3)

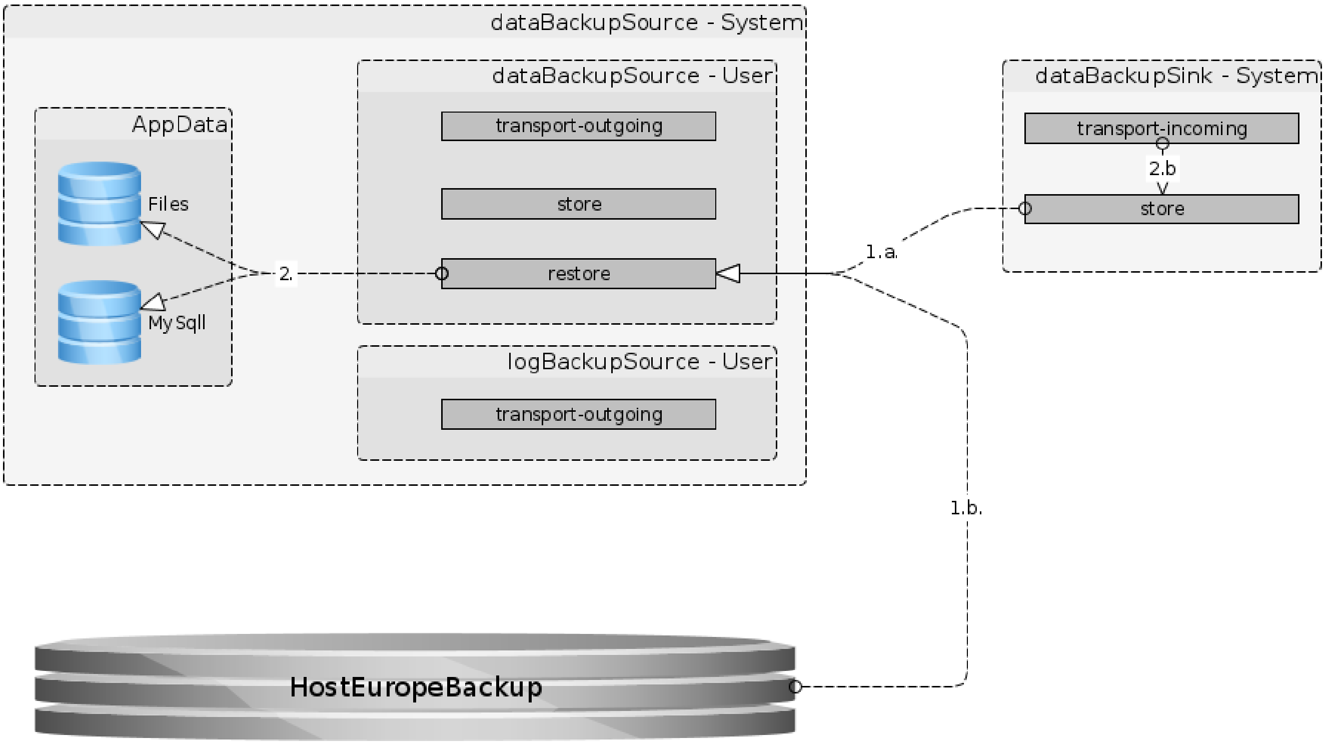
In the transport step a sink system cron job will

1. do the transport (3.a): using ssh and rsync. For ssh, the sink system is authorized on logBackupSource user.
2. (optional) verify correctness: Done by hash comparison
3. move to Sink-Store (3.b): Moves the received backup to Sink-Store
4. handle Sink generations: Deletes the eldest backup of the defined generations to be preserved.

4.1.4 Use HostEurope Backup (4)

HostEurope backs up the whole file system of the last 14 days. If we just store one daily generation on the local file system, we have the ability to rollback to one of the last 14 days.

4.2 Restore Process Steps



4.2.1 Restore Data Transport (1)

1. Source for restores can either be
   1. the dataBackupSink (1.a)
   2. or the HostEurope Backup (1.b). How to get backuped files back is described at https://kis.hosteurope.de/support/faq/index.php?cpid=16063&in_object=40&searchword=backup

2. Access to the system being restored has to be assigned manually.
3. Backups to be restored, have to be placed in /home/dataBackupSource/restore. The restore script will take the newest backup available in the restore folder.

4.2.2 Restore (2)

The restore step is part of the system applications' individual procedure.

4.3 Backup Source

On nodes serving as backup source, there are two backup users called dataBackupSource and logBackupSource.

4.3.1 Source System Principles

1 Backups are pulled from the source systems.

2 Process

2.1 The current day's files are held for transport-outgoing to the backup sink system.

2.2 After successful transport, files are eventually moved to a local storage folder.

3 Access

3.1 The backup data and the backup log are owned by different users:

3.1.1 dataBackupSource

3.1.2 logBackupSource

3.2 Access to the backup source users is managed by /home/[sourceUserName]/.ssh/authorized_keys.

4.3.2 Transport outgoing Folder

1 Folder-name

1.1 transport-outgoing

2 File-name

2.1 [application name]

2.2 [backup source type] (eg. file system | mysql)

2.3 time stamp

3 Example:

3.1 transport-outgoing/dda-owncloud_meissa_mysql_2015-01-28_04-52-01.sql

4.3.3 Local Backup Store Folder

1 Folder-name

1.1 store

2 File-name (same as in 4.3.2 Transport outgoing Folder)

4.4 Backup Sink

4.4.1 Sink System Principles

1 Process

1.1 The backups are pulled from the source systems and stored to a transport incoming folder

1.2 After the transport steps are triggered from the sink system:

1.2.1 Check successful transport.

1.2.2 Execute the source systems after the transport steps.

1.2.3 Execute the sink systems after the transport steps.

1.2.3.1 Rotate backup files

1.2.3.2 Do after process analysis

2 Access

2.1 The backup data and the backup log are pulled from different users:

2.1.1 dataBackupSink

2.1.2 logBackupSink

2.2 the sink users have to be authorized at the source user,

2.3 the sink user is managed by /home/[sinkUserName]/.ssh/authorized_keys.

4.4.2 Transport incoming Folder

1 Folder-name

1.1 transport-incoming

1.2 \source-systems-dns-name

2 File-name

2.1 as named on source system

3 Example

3.1 transport-incoming/owncloud.example.org/dda-owncloud_meissa_mysql_2015-01-28_04-52-01.sql

4.4.3 Backup Store Folder

1 Folder-name

1.1 store

1.2 \[source-systems-dns-name]

1.3 \[generation store] (e.g. daily | weekly | monthly)

2 File-name as described in 4.4.2 Transport incoming Folder

3 Example:

3.1 store/owncloud.example.org/daily/dda-owncloud_meissa_mysql_2015-01-28_04-52-01.sql

5 Solution Details

5.1 Interface between application and backup

The applications will have their own backup and restore routines.

1. Will be located at /usr/lib/[app-name]/bin

6 Appendix

6.1 Solutions for more Advanced Backup

6.1.1 rsnapshot

Rsnapshot uses hardlinks and rsync in order to generate differential backups with the apperance of full backups.

6.1.1.1 Documentation

- http://wiki.ubuntuusers.de/rsnapshot
- http://www.rsnapshot.org/
- http://how-to.linuxcareer.com/guide-to-rsnapshot-and-incremental-backups-on-linux

6.1.2 bacula

bacula is a full blown BackupSystem. Beside of great scaling abilities, bacula will also provide the feature, that users can inspect and restore single files from the backup.

6.1.2.1 Documentation

- http://www.bacula.org/
- http://wiki.ubuntuusers.de/Bacula