Introduction

What is Bareos?

Reasons for forking

Feature highlights

Stringent open source strategy

How to participate
Overview

Agenda

1. What is bareos?
2. Reasons for forking and speed up development
3. Some new feature highlights
4. Stringent open source strategy
5. How to participate
1. Introduction

2. What is Bareos?

3. Reasons for forking

4. Feature highlights

5. Stringent open source strategy

6. How to participate
What is Bareos?

- Network-based multi-platform backup solution
- bacula fork
- License: AGPL
- Massive code cleanup
- Sourcecode hosted at GitHub
Overview

1 Introduction

2 What is Bareos?

3 Reasons for forking

4 Feature highlights

5 Stringent open source strategy

6 How to participate
Why Bareos?

- bacula.org development virtually stopped
- ohloh.net: -88% commits and -57% contributors in 12 month
- community patches were not introduced
- new things only in closed source "Bacula Enterprise Edition"

<table>
<thead>
<tr>
<th>30 Day Summary</th>
<th>12 Month Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit</td>
<td>Commit</td>
</tr>
<tr>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>Commit</td>
<td>Commit</td>
</tr>
<tr>
<td>0</td>
<td>Down -247 (88%) from previous 12 months</td>
</tr>
<tr>
<td>Contributor</td>
<td>Contributor</td>
</tr>
<tr>
<td>0</td>
<td>Down -4 (57%) from previous 12 months</td>
</tr>
</tbody>
</table>
Goals

- 100% free software
- continuous development
- compatible to bacula as far as possible and senseful
- cleanup the configuration where needed
Overview

1. Introduction
2. What is Bareos?
3. Reasons for forking
4. Feature highlights
5. Stringent open source strategy
6. How to participate
SD-SD Replication

Before only copy and migration inside of one SD was possible
Now we can copy and migrate between two SDs.
Data can be moved or copied over the network also remotely
Passive Client

- Before: init of data channel from FD to SD
- fd needs name resolution
- NAT/Firewall requires setup of back channel
passive client option reverses the init of the data channel
all problems solved
LTO tape drives, beginning with LTO Ultrium generation 4, use "AES encryption algorithm, with 256-bit keys, in GCM mode." A shorter way to say it is "AES256-GCM." Encrypting at the tape drive level can provide performance and capacity advantages for administrators. Encryption at the tape level works well with the compression before encryption, maximizing tape capacities, and allows high performance during backup. Encrypting using an appliance device introduces latencies that affect backup performance, and requires management of the additional device. Encrypting data at tape speed helps to avoid the need for encryption of data — and the concurrent drain on host performance.

- Standard since LTO-4
- AES256-GCM
- virtually no speed impact
- tape cannot be read without key
- complete bareos integration incl. bextract
Open Source affine crew

100% AGPL

100% owner-financed

https://github.com/bareos

- Sourcecode
- Regression tests
- Docs
- Contrib projects
Introduction
What is Bareos?
Reasons for forking
Feature highlights
Stringent open source strategy
How to participate

Sourcecode

Sourcecode on GitHub

https://github.com/bareos/bareos
Sourcecode

- Sourcecode, regressions and Docs at GitHub
- `git clone git://github.com/bareos/bareos`
- `git clone git://github.com/bareos/bareos-regress`
- `git clone git://github.com/bareos/bareos-docs`
Doxygen Documentation

http://regress.bareos.org/doxygen/html/
Overview

1. Introduction
2. What is Bareos?
3. Reasons for forking
4. Feature highlights
5. Stringent open source strategy
6. How to participate
get in touch with Bareos

- bareos-users@googlegroups.com
- bareos-devel@googlegroups.com
- bugs.bareos.org
- github.com/bareos
Come to our Stand: Building K, Level 2, Stand 2

Ask People with Bareos T-Shirts: