Open Source Backup Conference
29 - 30 September 2015 | Cologne

backy

VM backup beyond Bacula/Bareos

Christian Theune
@theuni
ct@flyingcircus.io

Mea Culpa

And I almost missed it — again

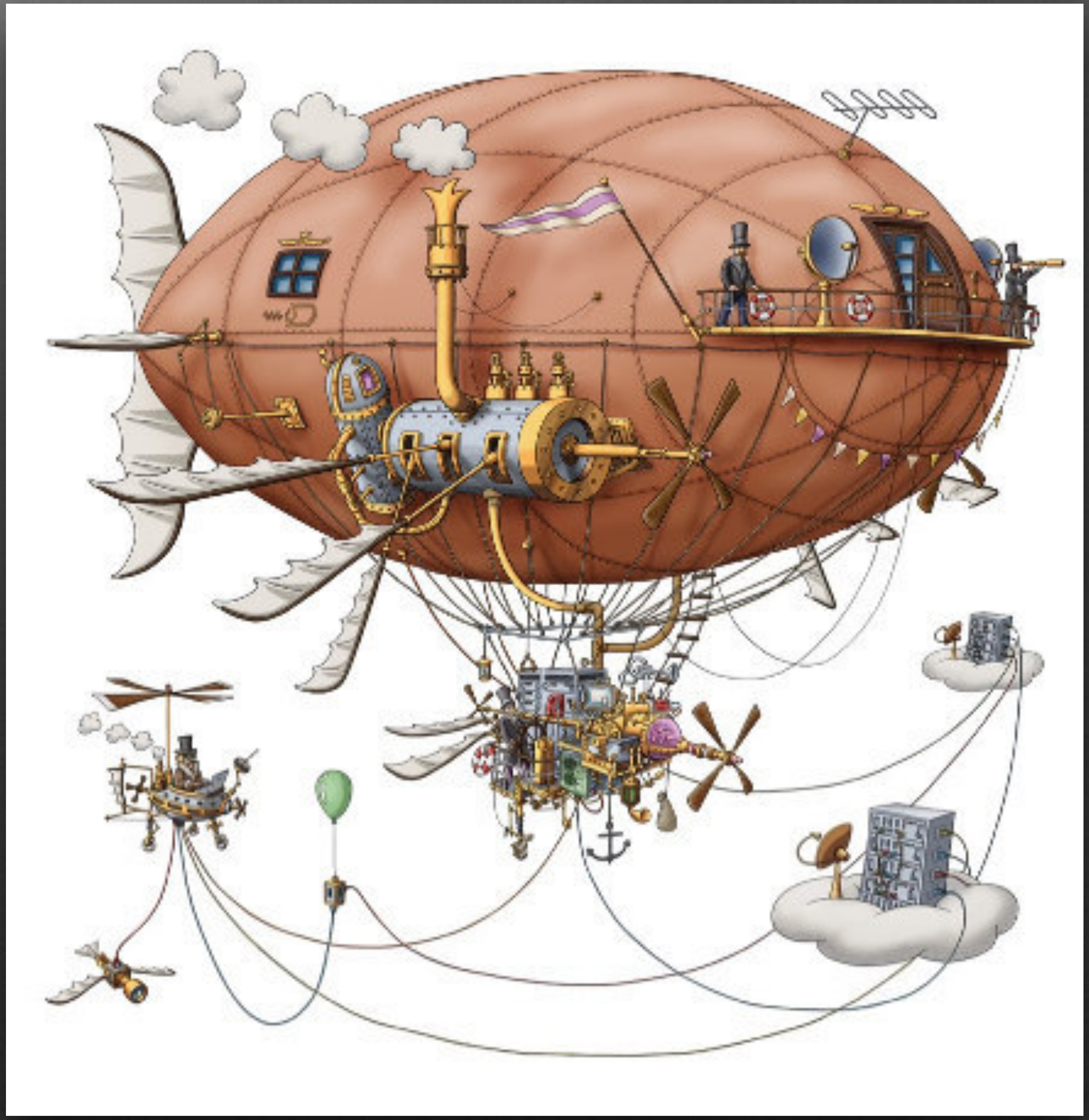
Backup!!11!!

FLYING CIRCUS

- flyingcircus.io

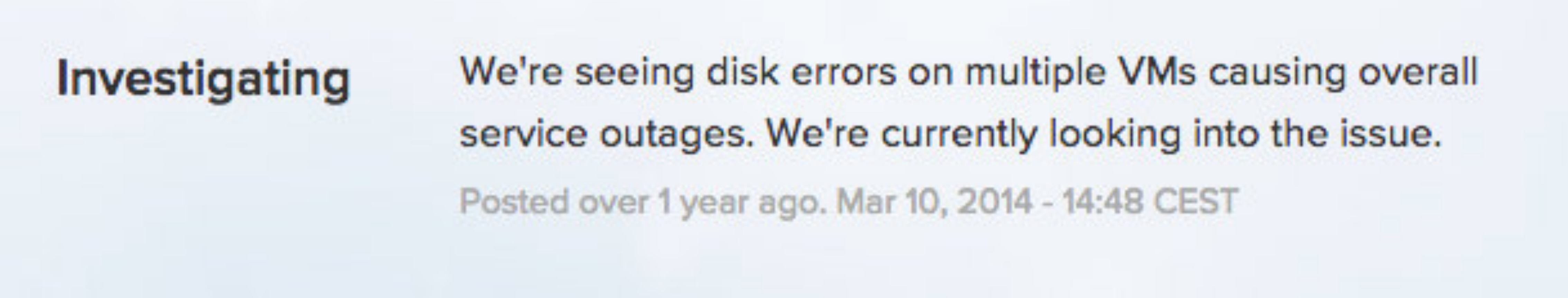
- DevOps as a Service

- custom, mission-critical web applications



Part I - Oh the Pain

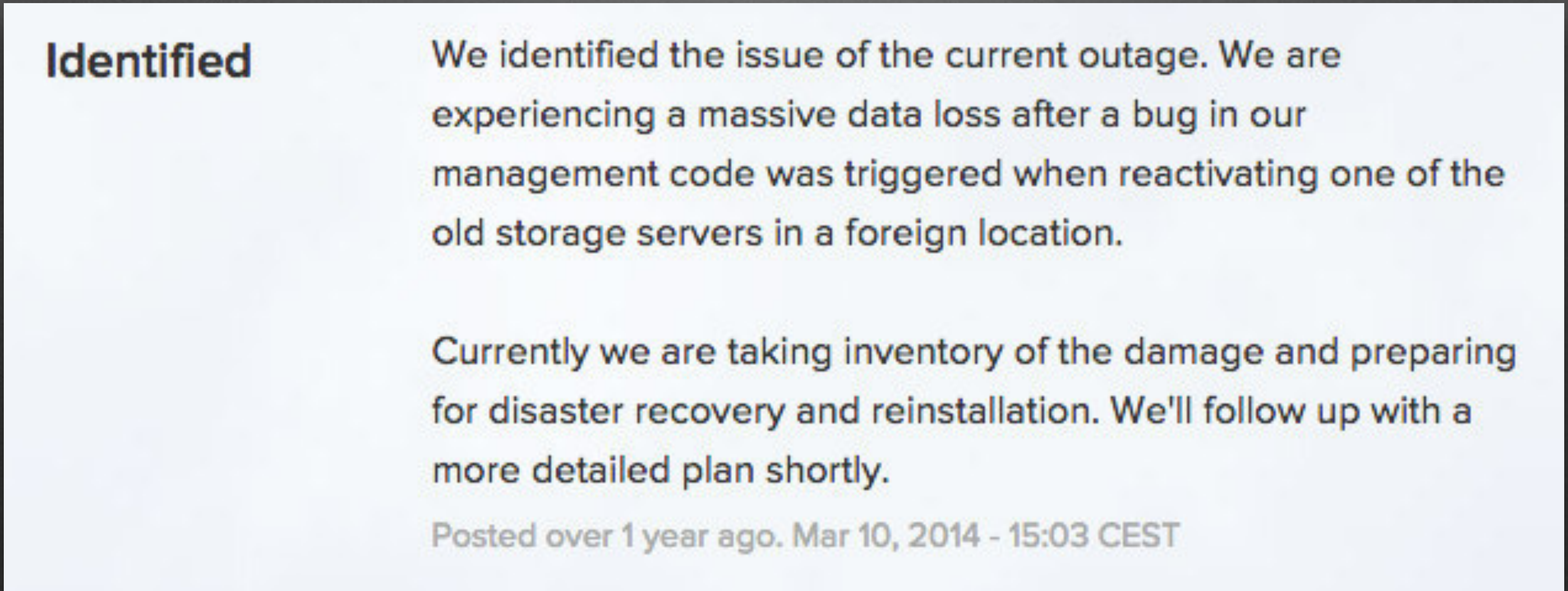
The story unfolds …



Investigating

We're seeing disk errors on multiple VMs causing overall service outages. We're currently looking into the issue.

Posted over 1 year ago. Mar 10, 2014 - 14:48 CEST

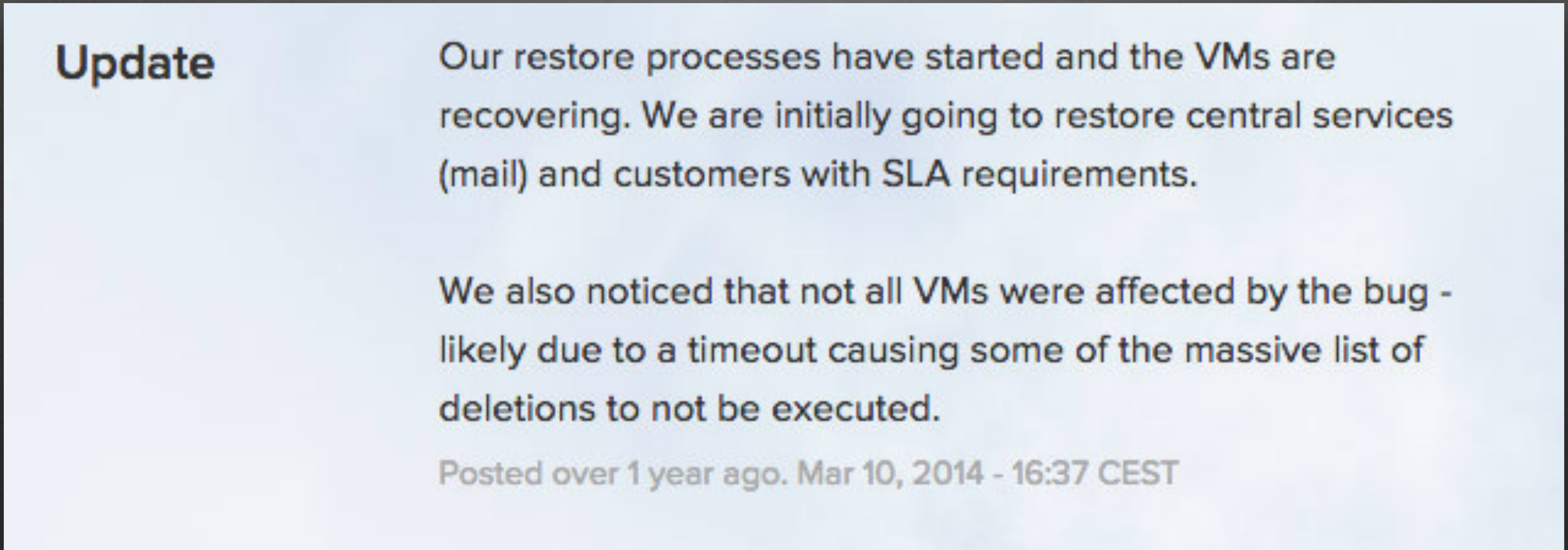


Identified

We identified the issue of the current outage. We are experiencing a massive data loss after a bug in our management code was triggered when reactivating one of the old storage servers in a foreign location.

Currently we are taking inventory of the damage and preparing for disaster recovery and reinstallation. We'll follow up with a more detailed plan shortly.

Posted over 1 year ago. Mar 10, 2014 - 15:03 CEST

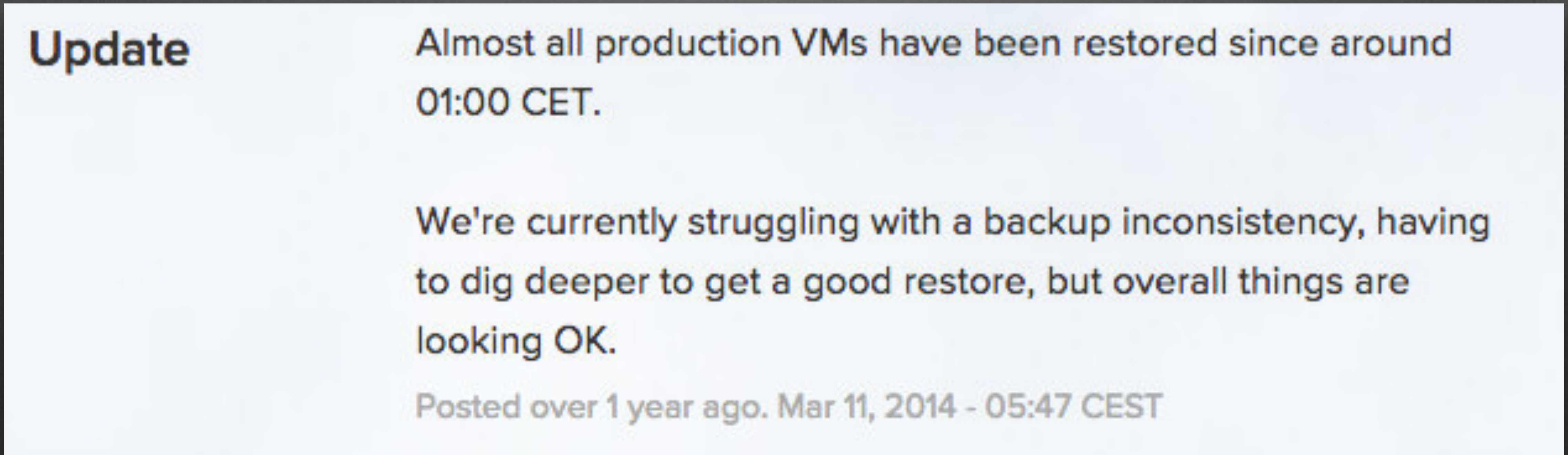


Update

Our restore processes have started and the VMs are recovering. We are initially going to restore central services (mail) and customers with SLA requirements.

We also noticed that not all VMs were affected by the bug - likely due to a timeout causing some of the massive list of deletions to not be executed.

Posted over 1 year ago. Mar 10, 2014 - 16:37 CEST

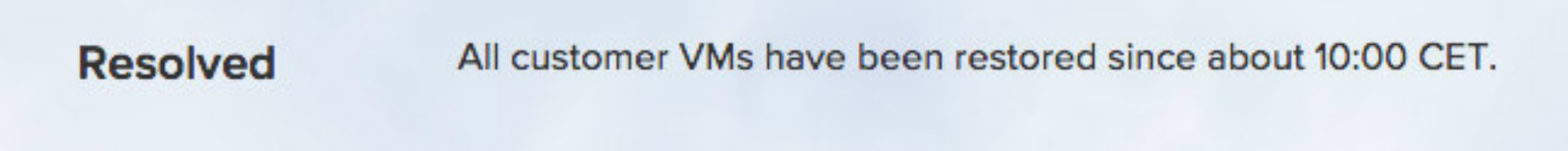


Update

Almost all production VMs have been restored since around 01:00 CET.

We're currently struggling with a backup inconsistency, having to dig deeper to get a good restore, but overall things are looking OK.

Posted over 1 year ago. Mar 11, 2014 - 05:47 CEST



Resolved All customer VMs have been restored since about 10:00 CET.

Root Cause Analysis

http://flyingcircus.io/postmortems/13266.pdf

Restore script bottleneck: global lock

Undetected inconsistency in important customer database

Bacula: complexity and the VTL

Not “everything” backed up.

24 hours are not a sufficient RPO in quite a few cases

Paper cuts

- Hard link farms
- Boot loaders
- The director as a “most valuable bottleneck”

Recap

- Restore fiddly to script
- Undetected inconsistency that was hard to deal with
- Blind spots
- Daily Interval
- Overall complexity, performance and the VTL
- Paper cuts

Part II - Make a wish

Simplicity

- Restore with basic Unix tools
- No VTL
- Not mixing data of different VMs

Reliability

- Verification / Scrubbing / (Repair)
- High frequency
- Integration with storage snapshots
- Not inventing new formats

Operability

- Avoid bottlenecks / head-of-line blocking
- Efficient deltas for large files (ZODB)
- Parallelisation (multiple jobs and multiple servers)
- Simple scripting and environment-specific integration
- Coordination: pre/post actions on storage, hypervisor, VM …

Operability II

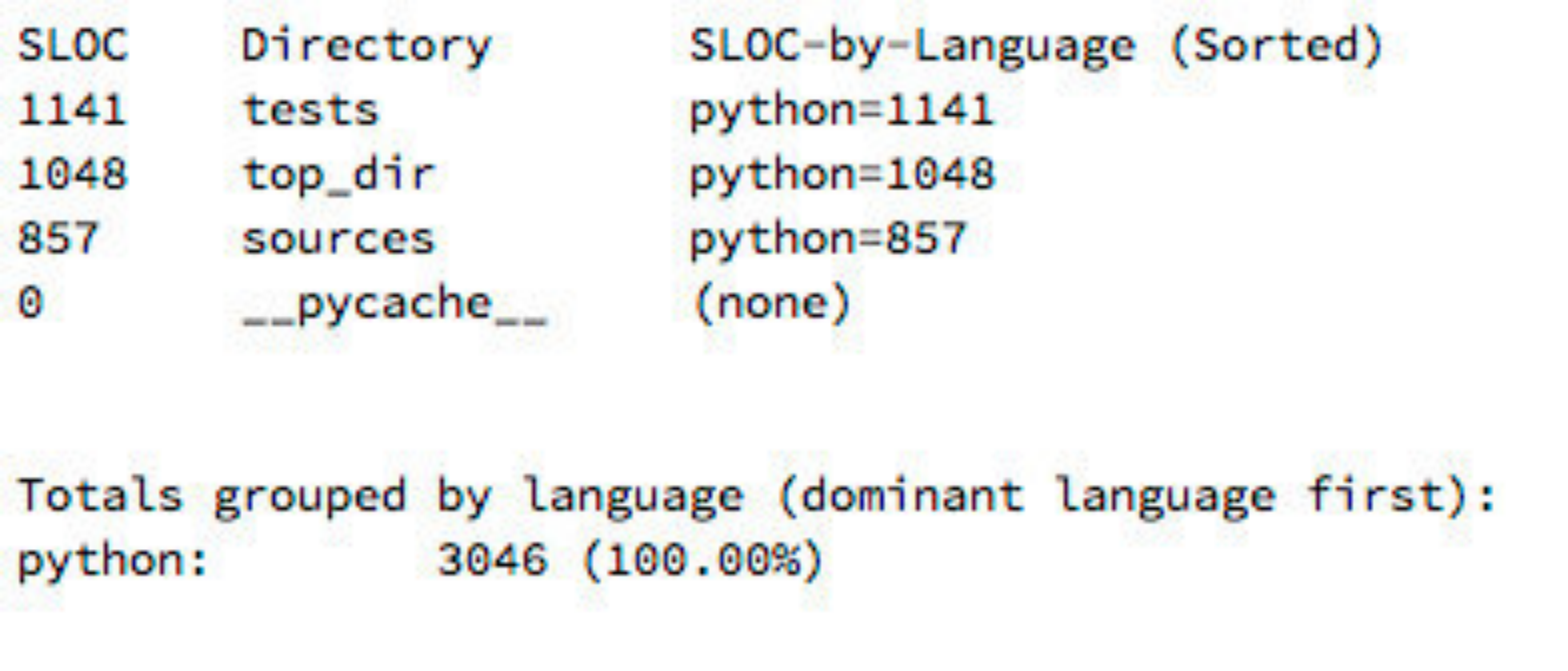
- Simple Nagios integration to ensure we notice RPO/SLA failures
- RTO-compliance during mass-restore
- Self-service for customers to restore files or VMs

Part III - Let’s do this!

It’s all about size

> “One size fits all … not”
>
> *–Probably someone, maybe me*

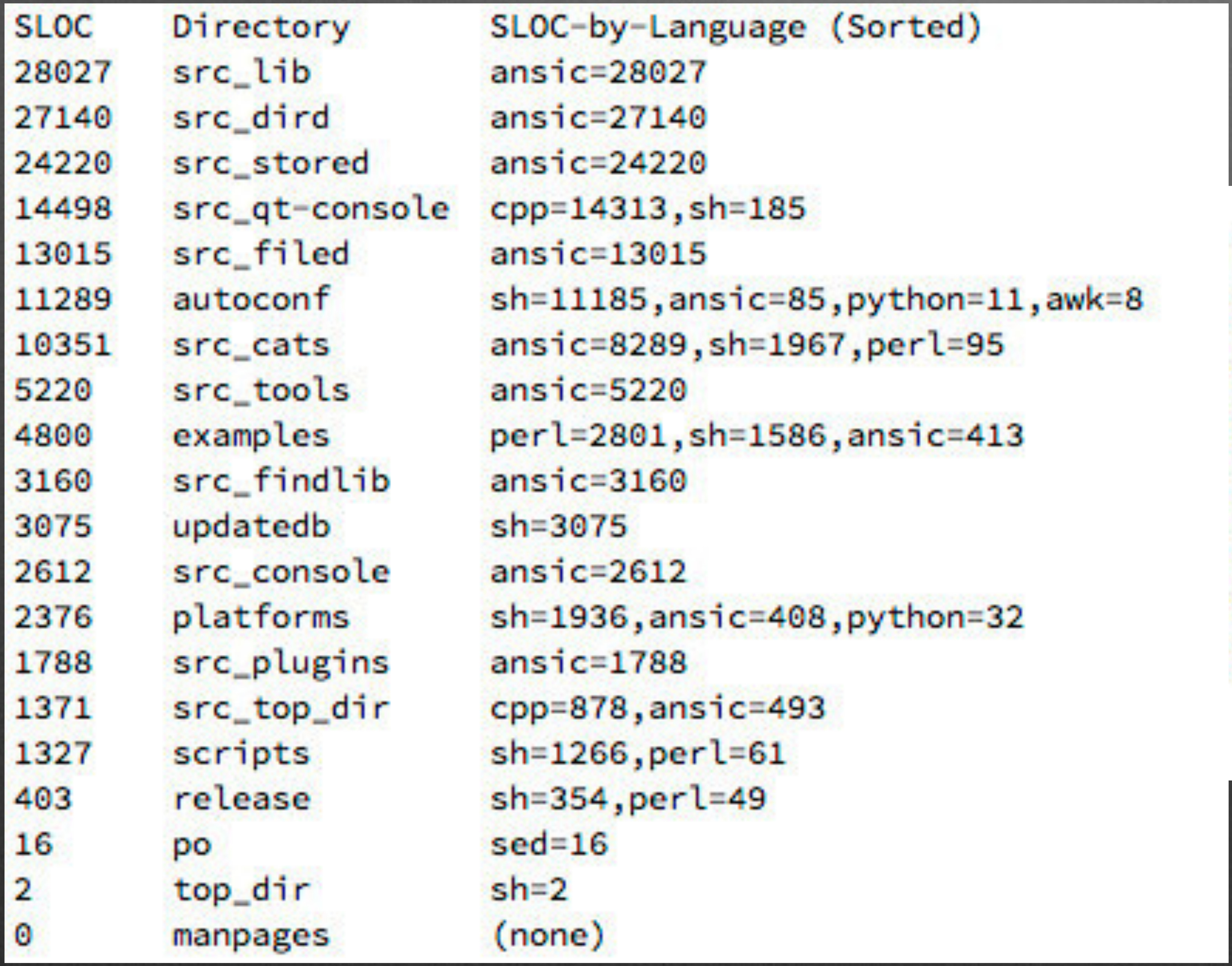
It’s all about size: backy



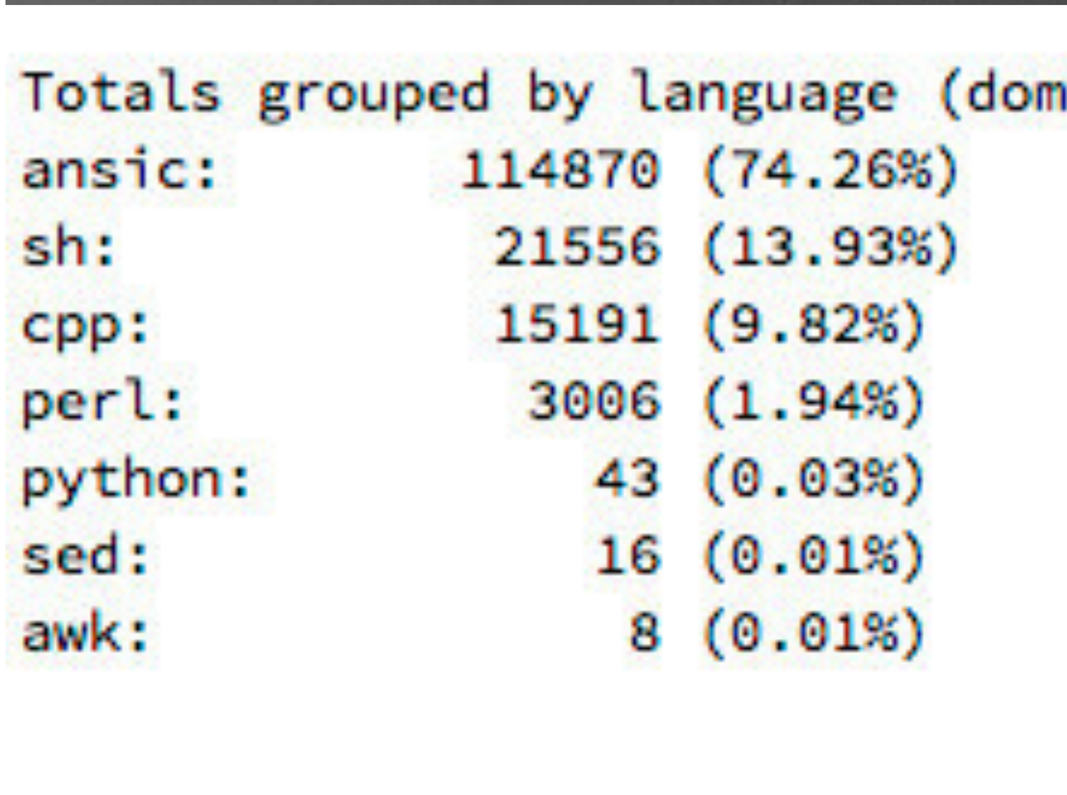
```
SLOC    Directory       SLOC-by-Language (Sorted)
1141    tests           python=1141
1048    top_dir         python=1048
857     sources         python=857
0       __pycache__     (none)


Totals grouped by language (dominant language first):
python:       3046 (100.00%)
```

It’s all about size: Bacula

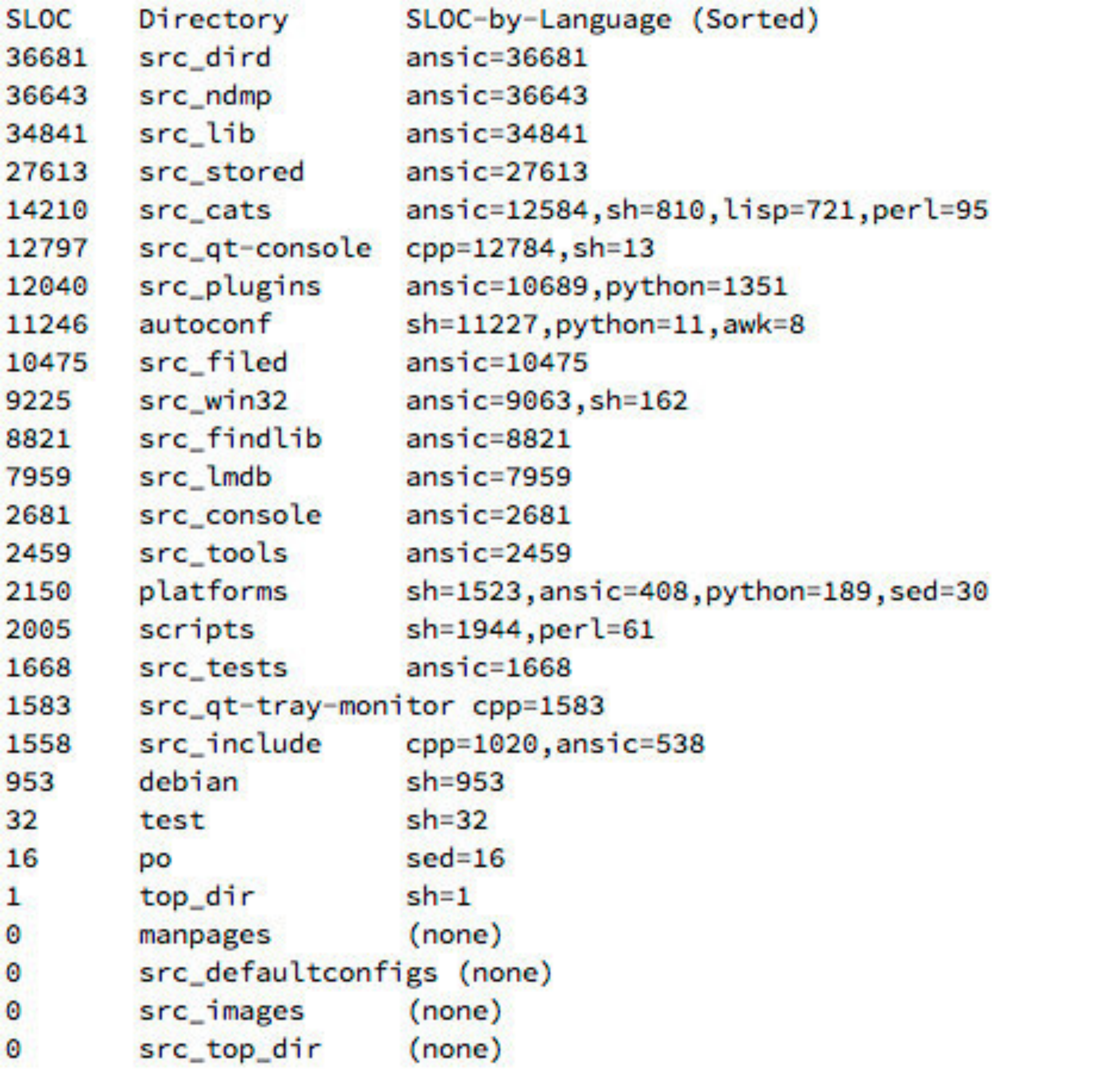


```
SLOC    Directory        SLOC-by-Language (Sorted)
28027   src_lib          ansic=28027
27140   src_dird         ansic=27140
24220   src_stored       ansic=24220
14498   src_qt-console   cpp=14313,sh=185
13015   src_filed        ansic=13015
11289   autoconf         sh=11185,ansic=85,python=11,awk=8
10351   src_cats         ansic=8289,sh=1967,perl=95
5220    src_tools        ansic=5220
4800    examples         perl=2801,sh=1586,ansic=413
3160    src_findlib      ansic=3160
3075    updatedb         sh=3075
2612    src_console      ansic=2612
2376    platforms        sh=1936,ansic=408,python=32
1788    src_plugins      ansic=1788
1371    src_top_dir      cpp=878,ansic=493
1327    scripts          sh=1266,perl=61
403     release          sh=354,perl=49
16      po               sed=16
2       top_dir          sh=2
0       manpages         (none)
```

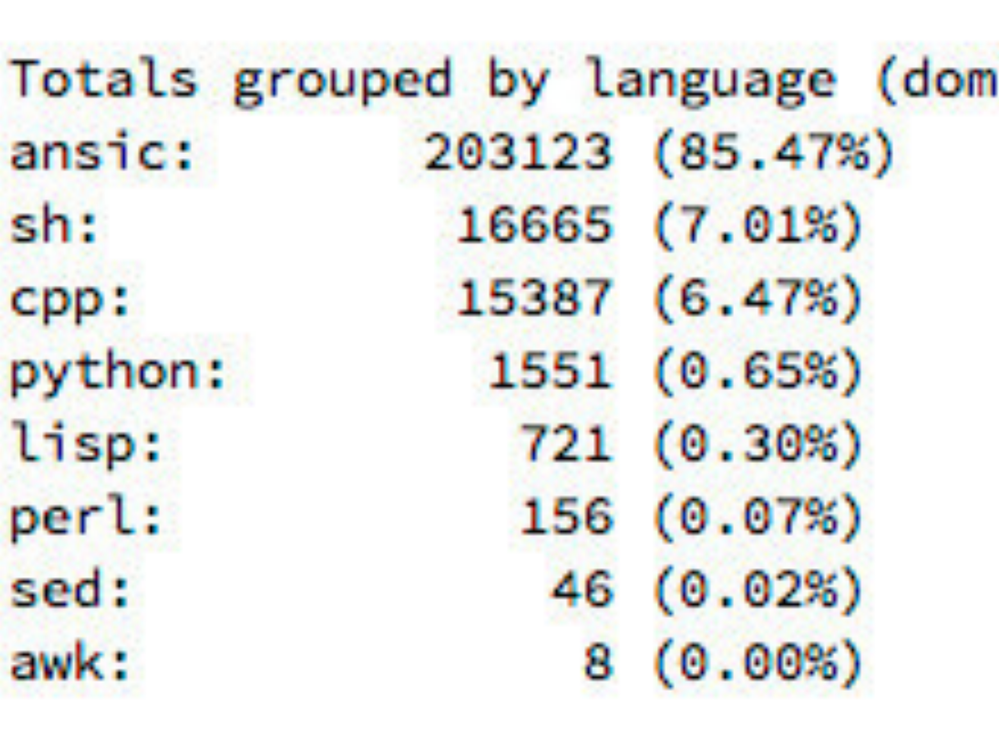


```
Totals grouped by language (dom
ansic:       114870 (74.26%)
sh:           21556 (13.93%)
cpp:          15191 (9.82%)
perl:          3006 (1.94%)
python:          43 (0.03%)
sed:             16 (0.01%)
awk:              8 (0.01%)
```

It’s all about size: Bareos



```
SLOC     Directory          SLOC-by-Language (Sorted)
36681    src_dird           ansic=36681
36643    src_ndmp           ansic=36643
34841    src_lib            ansic=34841
27613    src_stored         ansic=27613
14210    src_cats           ansic=12584,sh=810,lisp=721,perl=95
12797    src_qt-console     cpp=12784,sh=13
12040    src_plugins        ansic=10689,python=1351
11246    autoconf           sh=11227,python=11,awk=8
10475    src_filed          ansic=10475
9225     src_win32          ansic=9063,sh=162
8821     src_findlib        ansic=8821
7959     src_lmdb           ansic=7959
2681     src_console        ansic=2681
2459     src_tools          ansic=2459
2150     platforms          sh=1523,ansic=408,python=189,sed=30
2005     scripts            sh=1944,perl=61
1668     src_tests          ansic=1668
1583     src_qt-tray-monitor cpp=1583
1558     src_include        cpp=1020,ansic=538
953      debian             sh=953
32       test               sh=32
16       po                 sed=16
1        top_dir            sh=1
0        manpages           (none)
0        src_defaultconfigs (none)
0        src_images         (none)
0        src_top_dir        (none)
```



```
Totals grouped by language (dom
ansic:       203123 (85.47%)
sh:           16665 (7.01%)
cpp:          15387 (6.47%)
python:        1551 (0.65%)
lisp:           721 (0.30%)
perl:           156 (0.07%)
sed:             46 (0.02%)
awk:              8 (0.00%)
```