



Addressing a billion-entries multi-petabyte distributed filesystem backup problem with *cback*: from files to objects

Roberto VALVERDE (CERN), Hugo GONZALEZ (CERN)

19/05/2021 - International Conference on Computing in High-Energy and Nuclear Physics (CHEP)

Outline

- Challenges of a multi- petabyte backup system
- CERNBox Legacy backup system
- New CERNBox backup design
- cback: introduction
- cback: backup
- cback: restore / prune
- cback: verify
- What's next / summary

Challenges of a multi-petabyte backup system

- **Distributed source data:** Source data is distributed in several instances and thousands of disks.
- **Time constraints.** If the backup policy of the service requires daily backups, we are limited by 24h (forever). Parallelisation techniques can be explored.
- **Backup should scale with the service itself.** The backup could perform well once is deployed, but it will continue to perform well while the service scales?
- **Impact on the running service.** Backup adds load to a running service. The approach of performing the backups “at night” might not help on services that are accessed by people all around the world.
- **Cost of the backup.** Backup storage and operational costs (maintenance, restores, verifications)
- **Data lifecycle.** Data protection rules could require deleting data contained in backups.
- **Backup verification.** Regular restores and integrity / consistency checks.

CERNBox legacy backup system

- Backup data is physically located in **same building** and stored using the **same underlying technology** as the source data
 - Susceptible to natural / software disaster scenarios.
- **File based. No snapshots.** Very difficult to restore a folder as it was at a certain moment.
- Restores are performed **manually** by the operators.
- **No backup verifications.**

New CERNBox backup design

- **8 EOS** clusters. **37,000** users / projects. **5,3 PB** of storage, **1.76 Bil** files , **4.4k disks**
 - 5 instances for users. Users sharded by username's initial letter
 - 3 instances for Project spaces. Projects sharded by project's initial letter

```
[root@eoshome-ns-i01-00 (mgm:master mq:master) ~]$ eos ls -lh /eos/user/  
drwxr-sr-+  1 root  root      209.81 T May  8 06:51 a  
drwxr-sr-+  1 root  root       70.28 T May  3 15:56 g  
drwxr-sr-+  1 root  root      114.69 T May  4 10:04 j  
drwxr-sr-+  1 root  root       70.80 T May 10 16:36 k  
drwxr-sr-+  1 root  root       26.53 T May 10 20:05 w  
[root@eoshome-ns-i01-00 (mgm:master mq:master) ~]$ eos ls -lh /eos/user/a | wc -l  
4557
```

CERNBox backup design (II)

- backup jobs per instance/letter VS backup jobs per user

	instance / letter	user
Storage Efficiency - Deduplication	+	-
Data security - How much data will be compromised?	-	+
Repository operations - restore, purging, verification...	-	+
Data Lifecycle - How easy to remove data from backup	-	+
Scaling - How well backup scales with the service	-	+
Parallelisation - How easy to parallelise	-	+
Disaster recovery flexibility - recovery priorities	-	+
Backup system simplicity	+	-

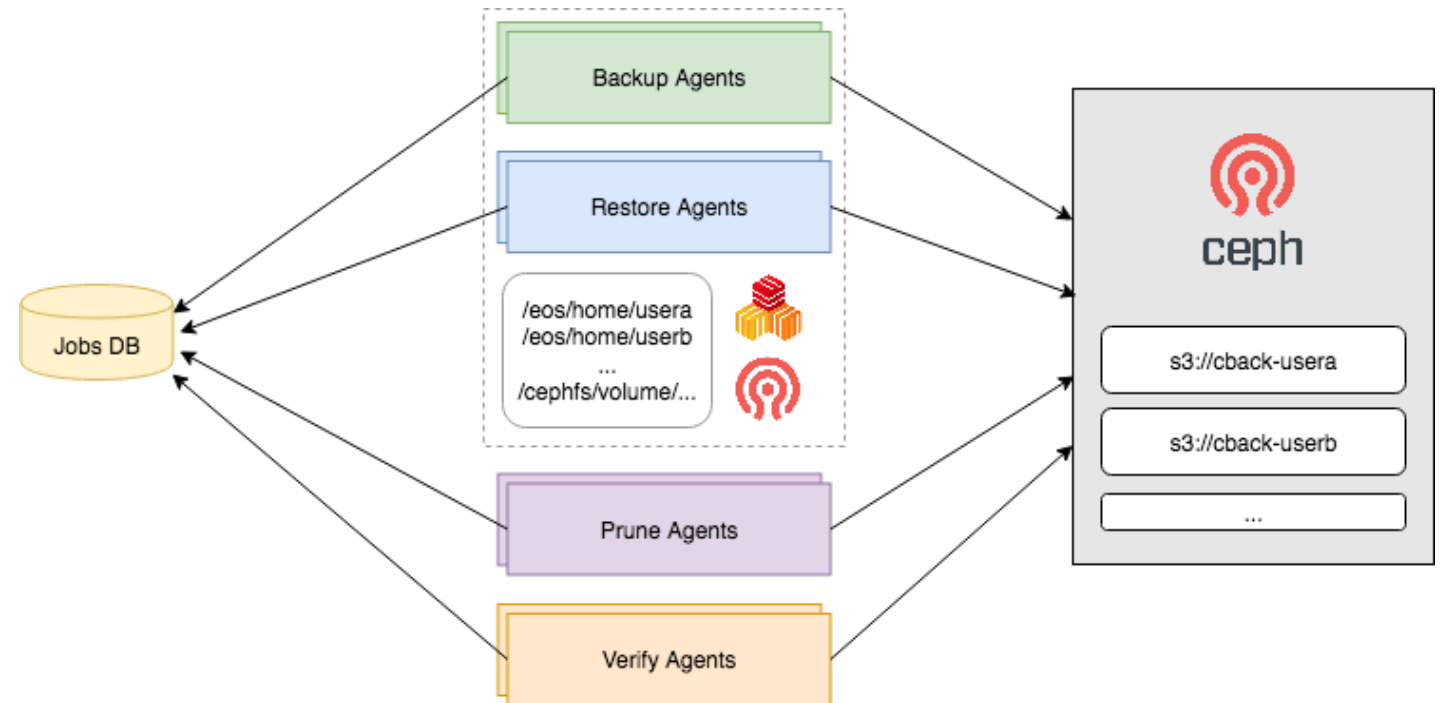
cback: Introduction

- **cback** is a backup orchestrator developed at CERN with the goal of offering daily snapshot-based backups for all CERNBox users and projects spaces.
- Optimised for **eosxd**, **cephfs** source filesystems but can be used with any other network mount system.
- Powered by **restic**
 - Easy, fast and secure
 - Repository based, incremental snapshots
 - Efficient: File & Chunk de-duplication (CDC)
 - Encryption, integrated purging of old snapshots, fuse mount...



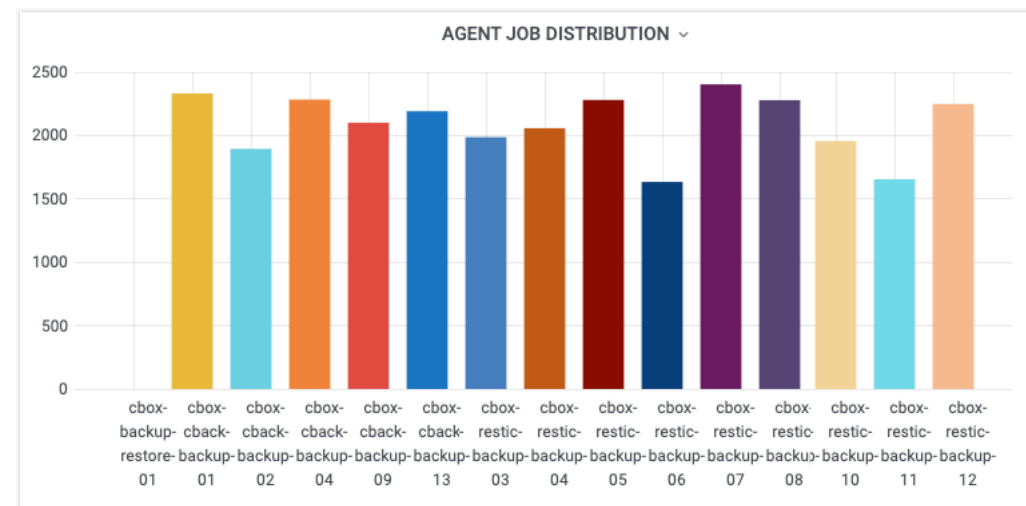
cback: Introduction (II)

- Composed by stateless agents that perform **backup**, **restore**, **prune** and **verify** tasks in an unattended way, coordinated via a central job database (pulling)
- S3 backend powered by **Ceph** in **another geographical** location (Prevessin, France)
 - **Erasure coding 4+2** layout, rack as a failure domain.
- Multi S3 backend support
- **Multi-replica backup** support with **automatic synchronisation**.



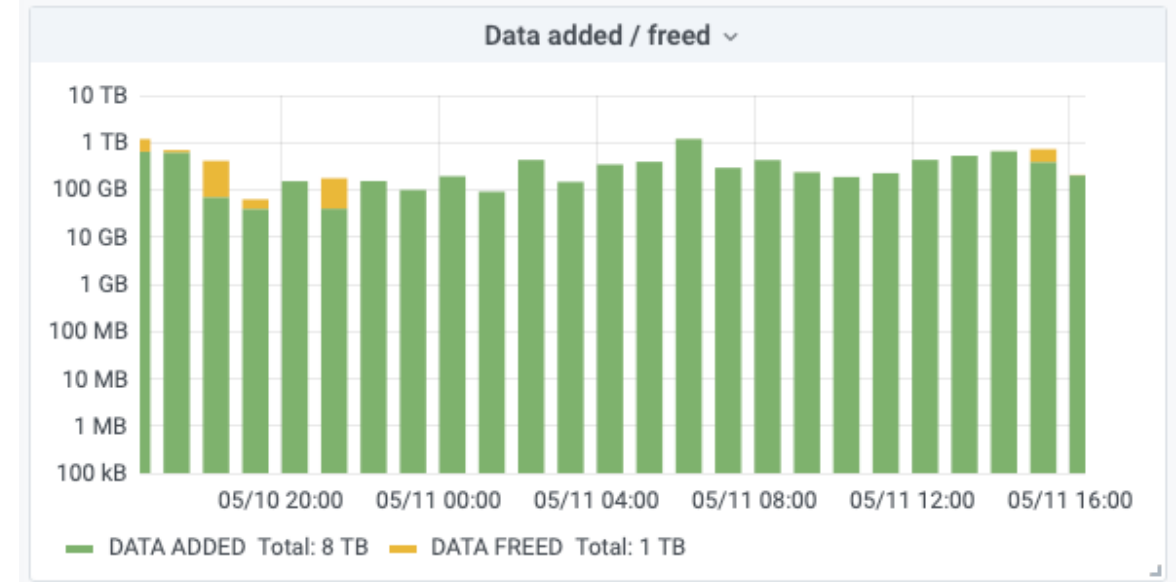
cback: backup

- **Per user** backup / only when needed
- **Always running** backup system with **no schedule**.
- **First backup** start time is based on the time when the job is added to the system (generally when a new user joins CERNBox)
- Subsequent backups will start **~24h after the last one** finished successfully. Because of this, start time of the jobs will slip it time depending of the duration.
- No concept of “**full snapshot**” of all backup jobs at any given time.



cback: restore / prune

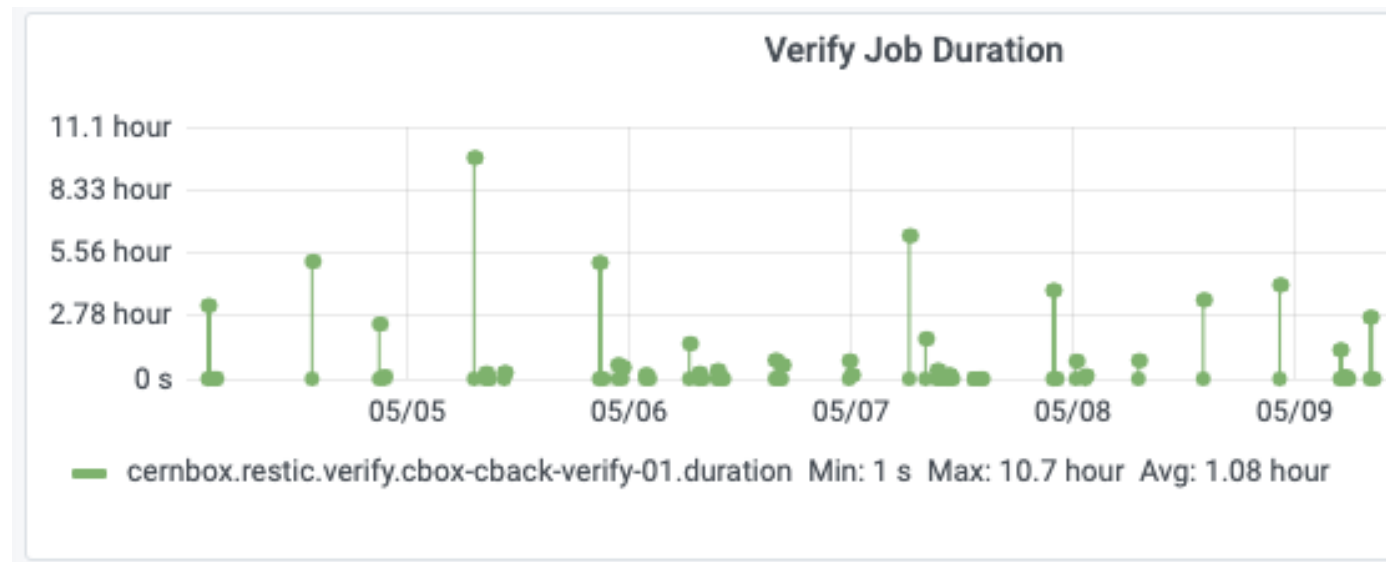
- Restore
 - **Full** and **partial** restore
 - **Synchronous / asynchronous** (unattended) based on restore jobs
 - Restore **in local** / directly **on source**
- Prune
 - **7 daily** snapshots, **5 weekly** snapshots and **6 monthly** snapshots are stored.
 - Automatic **archival** of backup jobs (only **last snapshot** is kept)



- +60 restore jobs this year
- Integrated into internal durability processes

cback: verify

- **Two phase** automatic verification **with file restoration**.
 - **Consistency** check using **restic** internal verification
 - **Integrity** check using EOS namespace dump previously saved.
 - Size and checksum.
- **Monitored and measured.**



What's next: Future plans

- Optimisations
 - Still a heavy process for the backend. Vast majority of files are unmodified
- Restore “**self-service**” for end-users
 - Access from CERNBox web interface / Access via command line using **restic** mount feature.
- Restic and **cold storage**.
 - There is effort in the community to bring **cold-storage** use cases to **restic**.
 - This would allow **hot/cold** backup or even a combination of both

More info

- Paper, EOS Workshop presentations [2020](#) [2021](#)
 - rvalverd@cern.ch, gonzalhu@cern.ch



home.cern

Backup Slides

cback: How it works

- Data base is populated with job information:
 - **user:** rvalverd, **path:** /eos/user/r/rvalverd, **repository:** <https://s3.cern.ch/cback-rvalverd>
 - The different agents, running as **services**, will query this database regularly (30s) and get one **random** job (same kind of the type of agent) which is in **Pending** status. No extra selection logic on the agents.
 - Job will perform the work, updating the status to **Completed/Failed**.
 - All agents have a **fast-retry** mechanism that retries the backup right away after failing.
 - A component called **cback-switch** is the responsible of updating the database and mark jobs from **Completed** to **Pending** when the job is considered as **outdated** (older than **24h** for backups, **72h** for prune jobs and **1 month** for verify jobs)
 - Having this a separate component allows to **control the frequency** and time distribution of the jobs in real time without needing to stop/start the agents.

cback: command line interface

```
[22:37][root@cbox-restic-backup-03 (qa:box/restic/agent/backup) ~]# cback backup status
CERNBack - Backup Status Summary
```

Total	Completed	Pending	Running	Failed	Not Enabled	Archived	Coverage
36102	34332	10	30	4	1645	1726	99.86 %

```
Running Backups
```

Id	User	Path	Status	Worker	Started
14875	a	/eos/project-	Running	cbox- <i>restic</i> -backup-01.cern.ch:1115	2021-02-17 03:18:00
15453	v	/eos/home-v/v	Running	cbox- <i>restic</i> -backup-08.cern.ch:1341	2021-03-02 05:45:50
1759	y	/eos/home-y/y	Running	cbox- <i>restic</i> -backup-12.cern.ch:19815	2021-03-02 07:29:51
528	s	/eos/home-s/s	Running	cbox- <i>restic</i> -backup-06.cern.ch:1125	2021-03-02 09:05:07
40842	c	/eos/project-	Running	cbox- <i>cback</i> -backup-13.cern.ch:1073	2021-03-02 14:09:57
17293	d	/eos/home-d/d	Running	cbox- <i>restic</i> -backup-05.cern.ch:1119	2021-03-02 16:49:55
13417	d	/eos/home-d/d	Running	cbox- <i>restic</i> -backup-12.cern.ch:1193	2021-03-02 18:47:40
16704	j	/eos/home-j/j	Running	cbox- <i>restic</i> -backup-11.cern.ch:7085	2021-03-02 20:08:27
2929	n	/eos/home-n/n	Running	cbox- <i>restic</i> -backup-04.cern.ch:1116	2021-03-02 20:23:01
48223	b	/eos/home-b/b	Running	cbox- <i>restic</i> -backup-08.cern.ch:1184	2021-03-02 21:14:26
40109	g	/eos/project-	Running	cbox- <i>cback</i> -backup-04.cern.ch:1520	2021-03-02 21:42:37
36192	e	/eos/project-	Running	cbox- <i>restic</i> -backup-05.cern.ch:1120	2021-03-02 21:42:38
6189	l	/eos/home-l/l	Running	cbox- <i>restic</i> -backup-11.cern.ch:6994	2021-03-02 22:10:30
13319	a	/eos/home-a/a	Running	cbox- <i>cback</i> -backup-13.cern.ch:19898	2021-03-02 22:19:41
40576	l	/eos/project-	Running	cbox- <i>restic</i> -backup-07.cern.ch:1124	2021-03-02 22:23:19
16361	m	/eos/home-m/m	Running	cbox- <i>cback</i> -backup-02.cern.ch:1118	2021-03-02 22:25:28
6175	k	/eos/home-k/k	Running	cbox- <i>restic</i> -backup-07.cern.ch:1121	2021-03-02 22:25:32
17017	f	/eos/home-f/f	Running	cbox- <i>cback</i> -backup-13.cern.ch:1065	2021-03-02 22:25:36
17814	a	/eos/home-a/a	Running	cbox- <i>restic</i> -backup-07.cern.ch:1127	2021-03-02 22:25:45
15426	v	/eos/home-v/v	Running	cbox- <i>restic</i> -backup-12.cern.ch:1188	2021-03-02 22:27:26
16169	m	/eos/home-m/m	Running	cbox- <i>restic</i> -backup-05.cern.ch:1127	2021-03-02 22:33:29
18109	t	/eos/home-t/t	Running	cbox- <i>cback</i> -backup-09.cern.ch:1131	2021-03-02 22:33:30
5169	c	/eos/home-c/c	Running	cbox- <i>cback</i> -backup-09.cern.ch:1122	2021-03-02 22:33:31
37813	l	/eos/home-l/l	Running	cbox- <i>cback</i> -backup-02.cern.ch:14408	2021-03-02 22:33:40
5067	c	/eos/home-c/c	Running	cbox- <i>restic</i> -backup-08.cern.ch:1182	2021-03-02 22:33:43
4175	h	/eos/home-h/h	Running	cbox- <i>restic</i> -backup-10.cern.ch:1188	2021-03-02 22:35:40
16838	h	/eos/home-h/h	Running	cbox- <i>cback</i> -backup-01.cern.ch:1191	2021-03-02 22:36:26
17344	c	/eos/home-c/c	Running	cbox- <i>cback</i> -backup-04.cern.ch:25956	2021-03-02 22:36:47
40814	d	/eos/project-	Running	cbox- <i>cback</i> -backup-02.cern.ch:1123	2021-03-02 22:37:13
16153	m	/eos/home-m/m	Running	cbox- <i>cback</i> -backup-01.cern.ch:1199	2021-03-02 22:37:14

```
[22:35][root@cbox-restic-backup-03 (qa:box/restic/agent/backup) ~]# cback backup status rvalverd
```

Id	User	Path	Instance	Repository	Status	Started	Finished
44837	rvalverd	/eos/home-r/rvalverd	home-r	s3:https://s3-fr-prevessin-1.cern.ch/cboxback-rvalverd	Completed	2021-03-02 18:16:43	2021-03-02 18:58:47



cback: command line interface (II)

- **All** cback operations are managed via a single command line user interface. This command line tool allows:

- CRUD for backup/restore jobs

```
cback backup add rvalverd /eos/user/r/rvalverd
```

- Running backup/restore/verify/prune jobs in interactive mode or agent (service) mode.

```
cback backup run <job_id>
```

```
cback backup agent
```

- Review the general status of backup/restore/verify/prune jobs or of any particular job

```
cback backup status
```

```
cback backup status rvalverd
```

- Wraps different **restic** commands: **restic find**, **restic snapshots**..

cback: command line interface (III)

- List backup/restore/verify/prune jobs

```
cback backup ls [-l] [completed|failed|disabled...]
```

- Manage the status of the backup jobs

```
cback prune enable|disable|reset|kill <job_id>
```

- As **cback** supports multiple S3 backends, it will load dynamically the different S3 environment configuration on the go.

```
cback backup snapshots <job_id>
```

- Other features like gently migrating a backup repository to a new S3 backend (uses **rclone**)

```
cback sync <job_id> --target-repo=s3.new.repository
```

cback: command line interface (IV)

- Running a backup job on demand

```
[22:51][root@cbox-restic-backup-03 (qa:box/restic/agent/backup) ~]# cback backup run 44837
[job:44837] >>> Backup started
[job:44837] Loading environment configuration for s3:https://s3-fr-prevessin-1.cern.ch/cboxback-rvalverd
[job:44837] Environment loaded.
[job:44837] id rvalverd
[job:44837] All source checks were successful
[job:44837] Checking Restic Repository s3:https://s3-fr-prevessin-1.cern.ch/cboxback-rvalverd
[job:44837] Repository seems fine!
Getting last snapshot information
/usr/bin/restic --no-cache -r s3:https://s3-fr-prevessin-1.cern.ch/cboxback-rvalverd snapshots --path=/eos/home-r/rvalverd --json
[job:44837] Repository data loaded
/usr/sbin/runuser -s /bin/sh rvalverd -c "eos root://eoshome-r.cern.ch file info /eos/user/r/rvalverd -m "
[job:44837] /usr/sbin/runuser -s /bin/sh rvalverd -c "eos root://eoshome-r.cern.ch file info /eos/user/r/rvalverd -m "
[job:44837] Source timestamp: 1614712724.918
[job:44837] Repository timestamp: 1614715410.0
[job:44837] Source timestamp < Snapshot timetamp 1614712724.918 < 1614715410.0
Sending metrics to monitoring
cernbox.restic.backup.cbox-restic-backup-03.backup_skipped 1
[job:44837] <<< Backup ended.
```

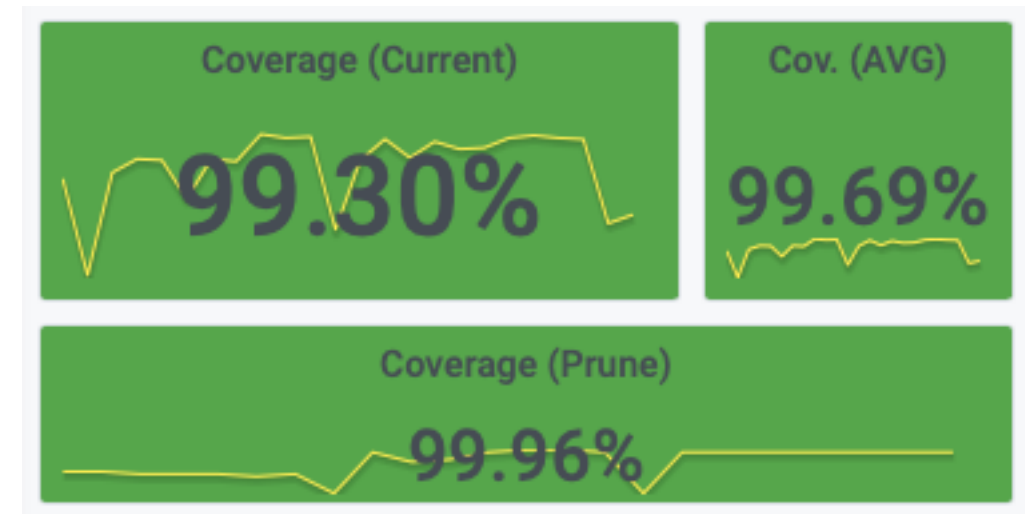
cback: command line interface (V)

- Running a backup job on demand


```
[10:52][root@cbox-restic-backup-10 (qa:box/restic/agent/backup) ~]# cback backup run 135
[job:135] >>> Backup started
[job:135] Loading environment configuration for s3:https://s3-fr-prevessin-1.cern.ch/cboxback-jcaladov
[job:135] Environment loaded.
[job:135] id jcaladov
[job:135] All source checks were successful
[job:135] Checking Restic Repository s3:https://s3-fr-prevessin-1.cern.ch/cboxback-jcaladov
[job:135] Repository seems fine!
Getting last snapshot information
/usr/bin/restic --no-cache -r s3:https://s3-fr-prevessin-1.cern.ch/cboxback-jcaladov snapshots --path=/eos/home-j/jcaladov --json
[job:135] Repository data loaded
/usr/sbin/runuser -s /bin/sh jcaladov -c "eos root://eoshome-j.cern.ch file info /eos/user/j/jcaladov -m "
[job:135] /usr/sbin/runuser -s /bin/sh jcaladov -c "eos root://eoshome-j.cern.ch file info /eos/user/j/jcaladov -m "
[job:135] Source timestamp: 1614704214.328
[job:135] Repository timestamp: 1614680197.0
[job:135] Source timestamp > Snapshot timestamp 1614704214.328 > 1614680197.0
[job:135] Running backup!
[job:135] Post successful
[job:135] Backup Successful!
[job:135] Updated force_backup flag with value: 0
[job:135] Backup completed with 0 file errors.
Sending metrics to monitoring
cernbox.restic.backup.cbox-restic-backup-10.files_new 0
cernbox.restic.backup.cbox-restic-backup-10.files_changed 1
cernbox.restic.backup.cbox-restic-backup-10.files_unmodified 3161
cernbox.restic.backup.cbox-restic-backup-10.dirs_new 0
cernbox.restic.backup.cbox-restic-backup-10.dirs_changed 3
cernbox.restic.backup.cbox-restic-backup-10.dirs_unmodified 1122
cernbox.restic.backup.cbox-restic-backup-10.file_errors 0
cernbox.restic.backup.cbox-restic-backup-10.backup_success 1
cernbox.restic.backup.cbox-restic-backup-10.data_added 927391
cernbox.restic.backup.cbox-restic-backup-10.duration 16
[job:135] <<< Backup ended.
```

cback: coverage

- Having this dynamic job workflow, how we know if we have recent backups for everyone ?
- Concept of backup/prune **coverage**:
 - Is the main **KPI** for the service and summarises in one value, the percentage of jobs with a **completed (successful) execution** in the last time window period (**24h** for backups, **72h** for prune, **1 month** for verify)
 - This is computed in real time, every time **cback backup status** is called.
 - Ideally **100%** at anytime of the day.



What we learned: Cache or not to cache

- Stateless agents goal, what about **cache**?
- **Restic** uses local cache to speed up the checking for changes. This cache contains metadata information from the S3 repository.
- **1st approach:** Not to use cache
 - Bad idea, very slow and lot of pressure on the S3 backend.
- **2nd approach:** Use cache
 - **Problem 1:** As any backup job can run on any backup agent, what happens if a node contains outdated cache? Not a problem, **restic** will handle that, great !
 - **Problem 2:** Cache size for big jobs could be in the order of hundreds of megabytes. As we could potentially at some point backup all jobs with the same agent, agents should have a decent local storage.
 - **Solution 1:** Use **cephfs** pool shared across all the agents.
 - **Problem:** Adds another dependency to the system.
 -  **Solution 2:** Use **disposable cache** (delete it after the backup). After some tests, we realised that the major benefit of having cache was while running big jobs. For this jobs, the time it takes for **restic** to re-create the cache was minimal compared to the duration of the actual job.

What we learned: Open file optimisation

- Backup was affecting the instance **performance**
- Open file optimisation
 - **Restic** performs a **fopen** system call to open files for checking for changes.
 - Not a problem on local filesystems
 - But on network filesystems implies to read the actual replica on the disk servers.
 - A fix was proposed to upstream and it's included since **restic 0.11.0**
[GITHUB-2969](#)

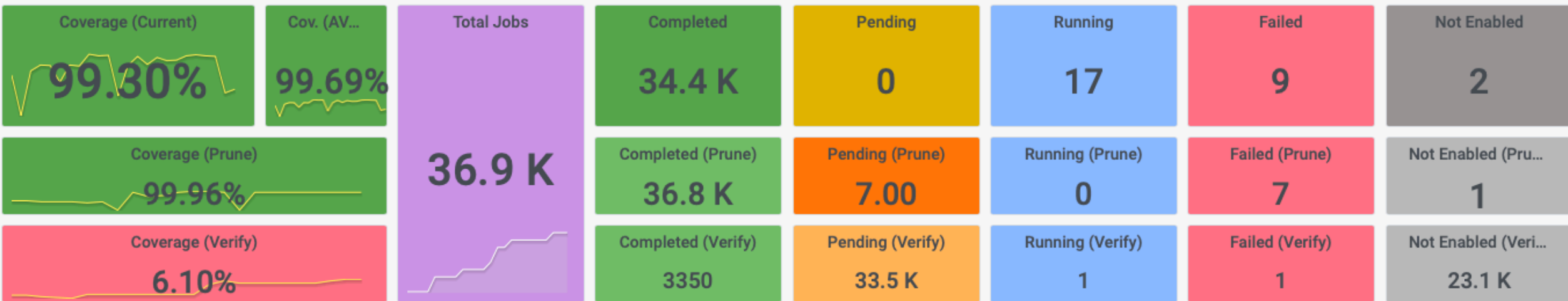
Cback: summary

- **Time constraints.** High level of parallelisation, “always running” to maximise the effective time for doing the backups.
- **Backup should scale with the service itself.** All main components are stateless, which allows vertical / horizontal scale.
- **Impact on the running service.** No fixed schedule. Backup jobs are scattered across the time.
- **Cost of the backup.** File/chunk deduplication, erasure coding. Easy operation using centralised command line tool. Possibility to expose restore to the users.
- **Data lifecycle.** Same granularity as user data. Automatic archival of backups.
- **Data security and backup verification.** Regular restores and integrity / consistency checks. Backup repositories encrypted individually.

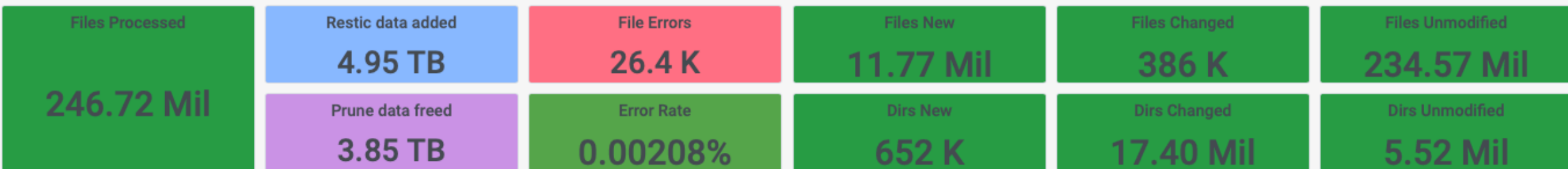
Backup storage

- CERNBox data storage is **5.3PB** (logical)
 - Recycle bin is not backed up (**-2.2PB**)
 - Effective data volume backed up: ~**3.1PB** (file versions are also excluded from the backup)
 - **Current backup storage: 3.3PB** (7 last daily snapshots, 5 last weekly snapshots, 6 last monthly snapshots of each home folder / project)





Backup Activity ▾



S3 Statistics

