

# Sysadministrivia

## Linux, Lagers, and Late Nights

---

# S4E5: "Join the Borg? Restic is Futile"

**Posted** 2019-04-29 03:59

**Modified** 2019-08-18 13:48

**Comments** 8

### Navigation

| Previous Episode         | Next Episode                    |
|--------------------------|---------------------------------|
| S4E4: "Special Delivery" | S4E6: "Be a Rancher, Not a Vet" |

### Log

| Recorded (UTC)      | Aired (UTC)         | Editor  |
|---------------------|---------------------|---------|
| 2019-04-17 03:07:10 | 2019-04-27 23:10:30 | "Edita" |

### Verification

| Format | SHA256   | GPG   | Audio File |
|--------|--|-------|------------|
| MP3    | f754b8347313d9f91eaa4ebce239a4d31ccc02466a49e081557c0a148000cd18 | click | click      |
| OGG    | b98e2e3efe6e28266e9b0dc8a448099490c447c74c0f4343904aff9c0fcc01f4 | click | click      |

Quicklisten:

Borg vs. Restic for performing backups.

- Just the Tip
- Notes
- Results
  - Single File
    - Repository Initialization
    - Backup (Non-Compressed)
      - Borg Compression (1GB File)
    - Differential (100MB change) (single file)
    - Restoring/Pulling a Backup
  - Multiple Files
    - Backup
    - Differential
    - Restoring/Pulling a Backup
- 15 Clams
- Errata
- Music

## Just the Tip

- Per-user resource limits
  - ulimit is dead. Jthan says it isn't but he's probably the type of person that still uses net-tools instead of iproute2.
  - cgroups are the way to go.
    - They can be used for a lot of different resources.
    - There is also a newer, systemd-explicit management.
      - You can access the RHEL 8 beta for free.

## Notes

Starts at **15m40s**.

I was drinking Victory DirtWolf again (and a Snapple Apple). Paden was drinking Absolut Vodka. Jthan was drinking Nikka Whisky From the Barrel.

- Borg vs. Restic
  - Paden:
    - Borg:
      - Smaller resulting snapshot (even with no compression enabled)
      - Requires an archive/snapshot name (restic generates checksum-based IDs)
    - Restic:
      - Faster
      - Progress display by default (borg requires `-e` or `--verbose`)
  - Me (Brent):
    - Borg:
      - Has wider maintenance base; more people working on it.
      - In python; easier to directly access. No public python API, but written cleanly enough to import and directly interact with.
      - Tends to have features before restic copies them (e.g. append-only mode - borg had this way before restic did).
      - The remote server **forces** use of SSH tunneling via piping, thus a secure and built-in authentication method (with more flexible ssh connection options than restic)
      - One would need to write their own REST API with Flask or the like.
      - The same binary (python script) is used for server and client; this is not the case for restic.
      - Several different backup protection (encryption-at-rest) models available.
      - Borg has support for reporting etc. in JSON, which can help with interacting in a programmatic manner.

- Restic:
  - Has an HTTP REST API, which can be nice for maintenance...
    - Except it's the **default** method for backing up. which means backup traffic over HTTP. SFTP is supported, but most of the emphasis for self-hosted remote backup servers is HTTP.
  - Written in Go, which has a fraction of accessibility of python.
  - It does not **force** HTTPS or auth for the server component.
  - It is not packaged for CentOS/RHEL; it requires COPR (gateway for third-party repos/packages); borg is in EPEL for CentOS/RHEL 7.
  - rest-server hasn't had a release since Feb 2018, and examples in their docs do not match latest release, just master:HEAD.
  - No compression.
- Jthan:
  - If you have data that will be shared **across repositories** or if you are using a cloud-storage provider (the bigger names), restic has some benefits.
  - He liked the nomenclature that restic uses more.

## Results

Please reference the tables below for results to tests that I (Brent) performed.

### Single File

A test backup file of 1GB of random data was created via `dd if=/dev/urandom of=junkdata.dat bs=32M count=32 iflag=fullblock`.

### Repository Initialization

Repositories were initialized via the following:

```
export RESTIC_PASSWORD="test"
export BORG_PASSPHRASE="test"
export BORG_RSH="ssh -l restic -p 22"
restic -r sftp:restic@[server]:testrepo1 init
restic -r rest:http://test:test@[server]:8000/testrepo2 init
borg init -e repokey restic@[server]:testrepo3
```

New repositories were created fresh for all tests (except for tests that were based on previous existing snapshots for e.g. differentials).

#### Repository Initialization

|             | Restic (SFTP) | Restic (REST) | Borg   |
|-------------|---------------|---------------|--------|
| <b>Time</b> | 9.982s        | 5.653s        | 2.302s |

### Backup (Non-Compressed)

Backups were then performed via:

```
restic backup -r sftp:restic@[server]:testrepo1 junkdata.dat
restic backup -r rest:http://test:test@[server]:8000/testrepo2 junkdata.dat
borg create restic@[server]:testrepo3::test1 junkdata.dat
```

#### Backup (Non-Compressed)

|             | Restic (SFTP) | Restic (REST) | Borg      |
|-------------|---------------|---------------|-----------|
| <b>Time</b> | 1m37.193s     | 1m35.313s     | 1m36.724s |

### Borg Compression (1GB File)

The difference in time consumption when using the highest compression was then tested with borg with a single 1GB file. A new repository was created and initialized (*testrepo4*), and the command:

```
borg create --compression lzma,9 restic@[server]:testrepo4::test1 junkdata.dat
```

was used. This process took **5m24.973s**, which is a big difference from the **1m36.724s** it took the uncompressed snapshot for the same data file. However, do keep in mind this is for binary data which does not compress very well.

### Differential (100MB change) (single file)

A change of 100MB of binary data was then created via `dd if=/dev/urandom bs=100M count=1 iflag=fullblock >> junkdata.dat`. The compression test for borg was not done (as this would be inaccurate representation for size, since restic does not support compression).

The following commands were executed:

```
restic backup -r sftp:restic@[server]:testrepo1 junkdata.dat
restic backup -r rest:http://test:test@[server]:8000/testrepo2 junkdata.dat
borg create restic@[server]:testrepo3::test2 junkdata.dat
```

#### Backup (Differential)

|  | Restic (SFTP) | Restic (REST) | Borg (Uncompressed) |
|--|---------------|---------------|---------------------|
| <b>Time</b>                                    | 0m14.400s     | 0m14.962s     | 0m17.608s           |
| <b>Size (before differential) (Bytes)</b>      | 1050228       | 1050236       | 1052844             |
| <b>Size (after 100MB differential) (Bytes)</b> | 1157008       | 1153272       | 1157348             |

### Restoring/Pulling a Backup

The restoration of a backup for a single ~1.1GB file was then compared.

Commands used:

```
mkdir restore
cd restore
restic -r sftp:restic@[server]:testrepo1 restore latest --target restic_sftp
restic -r rest:http://test:test@[server]:8000/testrepo2 restore latest --target restic_http
```

```
borg extract restic@[server]:testrepo3::test3
```

### Pulling 1.1GB Backup

|      | Restic (SFTP) | Restic (REST) | Borg      |
|------|---------------|---------------|-----------|
| Time | 3m56.694s     | 1m49.766s     | 1m41.365s |

## Multiple Files

For this test, it was attempted to mimic more realistic backup data - the number of files was changed from 1x1GB to 1000x1MB, and were text-only (to emulate logs, config files, MySQL dumps, LDAP LDIF dumps, etc.). You can find the script I wrote to generate the test text files here. Borg compression was tested in this case for comparison (as text compresses better than binary data, so the differences would be more easily noticed). Initialization was done the same as before, with the addition of a second borg repository for separate compression testing:

```
export RESTIC_PASSWORD="test"
export BORG_PASSPHRASE="test"
export BORG_RSH="ssh -l restic -p 22"
restic -r sftp:restic@[server]:testrepo1 init
restic -r rest:http://test:test@[server]:8000/testrepo2 init
borg init -e repokey restic@[server]:testrepo3
borg init -e repokey restic@[server]:testrepo3_compressed
```

## Backup

```
restic backup -r sftp:restic@[server]:testrepo1 /var/tmp/dirtest
restic backup -r rest:http://test:test@[server]:8000/testrepo2 /var/tmp/dirtest
borg create restic@[server]:testrepo3::test1 /var/tmp/dirtest
borg create --compression lzma,9 restic@[server]:testrepo3_compressed::test1 /var/tmp/dirtest
```

### Initial Backup (1000 1MB Files)

|                               | Restic (SFTP) | Restic (REST) | Borg (Uncompressed) | Borg (Compressed) |
|-------------------------------|---------------|---------------|---------------------|-------------------|
| Time                          | 1m33.719s     | 2m0.071s      | 1m34.584s           | 4m53.559s         |
| Size After Completion (Bytes) | 1026424       | 1026384       | 1028340             | 866992            |

## Differential

A modification of +1kB of text was then added to each of the 1000 files (by changing filesize to 1) for a total change size of ~1MB to the entire raw dataset.

```
restic backup -r sftp:restic@[server]:testrepo1 /var/tmp/dirtest
restic backup -r rest:http://test:test@[server]:8000/testrepo2 /var/tmp/dirtest
borg create restic@[server]:testrepo3::test2 /var/tmp/dirtest
borg create --compression lzma,9 restic@[server]:testrepo3_compressed::test2 /var/tmp/dirtest
```

### Differential Backup (1000 +1kB Change)

|                               | Restic (SFTP) | Restic (REST) | Borg (Uncompressed) | Borg (Compressed) |
|-------------------------------|---------------|---------------|---------------------|-------------------|
| Time                          | 1m6.949s      | 1m19.984s     | 1m22.133s           | 4m9.129s          |
| Size After Completion (Bytes) | 1751120       | 1755388       | 1903876             | 1588848           |

## Restoring/Pulling a Backup

It's worth noting that restic via HTTP REST actually **failed** this test, giving the following errors:

```
ignoring error for /var/tmp/dirtest/0/1/8: decrypting blob 4cb6d10fbd7ca23759d050f117255f70ee73be74d2e5b494eee8a5a4e593f8c9 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/6/3/2: decrypting blob c743cbe9f85d28bb4eb9d06b8687fc6463596db1f9955a8b8d78a8f8d0c32a75 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/9/4/7: decrypting blob 726423b01e90f98b294a7f3e360a6f0038812e1e2416a6c30c750d554d7acad6 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/9/3/6: decrypting blob 61b5e1a10587c44205c83d75b6b13db4dc1b9d0e09541dfc7563ee2e46eae112 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/9/4/5: decrypting blob 00022b138ea6057ca3a9392b35de85f4eaa2f389052c07ba8e02b6fbc42177e9 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/3/2/7: decrypting blob 6c799cffa51bb1c4341df2665654e9e83fa6b7dda44b1064a4ec9b9785f7b370 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/7/2/7: decrypting blob 3dc9a3b5690f134506a8ce5a6a33a3b4118b3784ffc92caea7641e8b1523a3b4 failed: ciphertext verification failed
ignoring error for /var/tmp/dirtest/0/1/8: UtimesNano: no such file or directory
ignoring error for /var/tmp/dirtest/3/2/7: UtimesNano: no such file or directory
ignoring error for /var/tmp/dirtest/7/2/7: UtimesNano: no such file or directory
ignoring error for /var/tmp/dirtest/9/3/6: UtimesNano: no such file or directory
ignoring error for /var/tmp/dirtest/9/4/5: UtimesNano: no such file or directory
ignoring error for /var/tmp/dirtest/9/4/7: UtimesNano: no such file or directory
There were 13 errors
```

So. There's that. But here's the results, keeping that in mind:

```
restic -r sftp:restic@[server]:testrepo1 restore latest --target testrepo1/sftp
restic -r rest:http://test:test@[server]:testrepo2 restore latest --target testrepo2/http
borg extract restic@[server]:testrepo3::test2
borg extract restic@[server]:testrepo3_compressed::test2
```

### Backup (Non-Compressed)

|      | Restic (SFTP) | Restic (REST) | Borg (Uncompressed) | Borg (Compressed) |
|------|---------------|---------------|---------------------|-------------------|
| Time | 1m47.816s     | 3m58.514s     | 1m34.612s           | 1m17.199s         |

## 15 Clams

In this segment, Jthan shares with you a little slice of life. The title is a reference to this video. (2m16s in)

Starts at **1h20m10s**.

Jthan has been trying to turn up Vault (we talk about it in S4E0). Jthan wants to know why time-based access for protecting digital resources as a security layer is pointless.

## Errata

- The “Jthan is a Neopet” reference is because he kept calling himself a compukoala during pre-show and I said “that sounds like a neopet”.
- No, Jthan, that actually is NOT a “double”.
- Jthan wants you to visit [git.jthan.io](http://git.jthan.io).
- This is Jthan’s award he got for the podcast.
- ~~I’ll write up an article on how to set up a hardened central borg server in about a week or so:~~ I finally wrote one up!
- Jthan asked about whether borg can be configured to mark certain snapshots (“archives”) as “read-only”, or persistent. You can’t do this on a per-archive level, but you CAN do it on a per-repo/per-pubkey level.
- The “how/where do you want to pay?” discussion is in S4E3.
- Paden, it is indeed 100% egg man.
- Thanks to amayer for catching a typo in the shownotes! You da man.

## Music

### Music Credits

| Track | Title                | Artist       | Link                  | Copyright/License |
|-------|----------------------|--------------|-----------------------|-------------------|
| Intro | Activation           | Spaceinvader | <a href="#">click</a> | CC-BY-ND 3.0      |
| Outro | Temperature’s Rising | LoveFine     | <a href="#">click</a> | CC-BY-ND 3.0      |

(All music is royalty-free, properly licensed for use, used under fair use, or public domain.)

**Author** r00t^2

**Categories** Season Four

## Comments

### 1. Bob Collins

2019-05-17 22:47 (799 days ago)

The current Borg depends on an old version Python package which causes breakage. Check out the Arch Linux discussion. I was planning on Borg, but now I’m leaning to Restic.

2.

2019-05-18 02:12 (799 days ago)

Bob-

Not sure which Arch Linux discussion you’re referring to, but I’m assuming you’re referring to this:

<https://bugs.archlinux.org/task/61346?project=5>

That’s for the msgpack library, which is fixed already (<https://git.archlinux.org/svntogit/community.git/tree/trunk/0001-Fix-msgpack-version-constraints-using-proper-setupto.patch?h=packages/borg>).

You’re welcome to use whatever you’re more comfortable with, but if you go with Restic – watch your snapshot integrity, because as I have shown above and talked about in-episode, I experienced snapshot corruption with it rendering the snapshot irretrievable.

### 3. Aigini

2019-09-25 02:33 (669 days ago)

For restic, the command to initialize a remote directory is :

```
restic -r rest:http://test:test@[server]:8000/testrepo2 init
```

But from this command, what does the first ‘test’ in test:test represent?  
I do not understand the syntax.

4.

2019-09-25 04:20 (669 days ago)

Aigini-

Sorry! To clarify, with contextual syntax it is:

```
restic -r rest:<proto>://<user>:<password>@<server>:<port>/<repo name> init
```

### 5. root3

2019-10-11 09:40 (653 days ago)

It actually seems that restic does not even have “append-only” functionality still at this point (October 2019). In the podcast you say that restic has implemented it, and I am not sure what you are referring to, or where did you get this information?

If you search for “restic append only” there is a well-circulated article on how to set up rclone as a restic server with the append-only option of rclone. Then you could use the combination of restic and rclone to get the append-only functionality. Is this the article you are referring to? While it does work, it would be hard to call this functionality built in into restic, because it has to use an external tool to achieve this. At this rate it would be almost just as easily to hack up some small bash script to manipulate repository file rights on the server, and achieve the same – restic still does not support it out of the box.

At the very least google(“site:restic.readthedocs.io/en/stable/ “append only”) brings up nothing.

So the original point stands, borg has richer functionality, restic lags behind in many features, and the reverse cannot be said for any almost any legitimate feature.

6.

2019-10-13 07:43 (651 days ago)

Thanks for your input! I’m not sure where I saw an append-only mode, but there apparently is an “—append-only” flag to the REST server component for Restic.

I definitely agree personally; I prefer borg.

### 7. Fabio Brizzolla

2020-01-20 19:33 (551 days ago)

A little late comment, but there’s one thing Restic has that Borg doesn’t: binaries for Windows x86/x64.

Windows Server doesn’t have WSL support yet in most of the deployed base (2016 – actually you just can add if no Win10 1909 Clients). This adds a problem since you cannot pull data from an Windows machine directly into the repository (or extra steps needed to mount.cifs it and so on).

IMHO this is a +1 point to Restic.

8.

2020-01-20 20:07 (551 days ago)

This is true; as you might have guessed, we all admin/engineer Linux systems so our Windows input is pretty low!

As you hinted at, borg does indeed run under WSL but there is also indeed the issue of no WSL for Windows Server clients.

For the Windows admins out there, Restic may indeed be the way to go (but I cannot stress enough to make sure you check your backups for corruption, as it was an issue I encountered!).