*RESTIC*

Project Analysis

INTRODUCTION

Restic is a modern backup program that can back up your files:

* from **Linux, BSD, Mac and Windows**
* to **many different storage types**, including self-hosted and online services
* **easily**, being a single executable that you can run without a server or complex setup
* **effectively**, only transferring the parts that actually changed in the files you back up
* **securely**, by careful use of cryptography in every part of the process
* **verifiably**, enabling you to make sure that your files can be restored when needed
* **freely** - restic is entirely free to use and completely open source

PROJECT SUMMARY

|  |  |
| --- | --- |
| WEBSITE | https://restic.net/ |
| ORGANISATION NAME | RESTIC OPEN SOURCE COMMUNITY |
| LICENSE | “BSD 2-Clause License”. |

|  |  |
| --- | --- |
| OPEN PROPRIETARY | OPEN SOURCE |
| SOURCE PATH(OPEN SOURCE) | GITHUB |
| BRIEF DESCRIPTION | RESTIC IS A BACKUP PROGRAM THAT IS FAST AND BACKUPS OUR FILES |

PROJECT DETAILS

* KEY FEATURES

Installation

To install, please follow the [Installation Instructions Page](https://restic.readthedocs.io/en/stable/020_installation.html) in the manual or download the latest native binary on the [GitHub Download Page](https://github.com/restic/restic/releases/latest).

Compatibility

Backward compatibility for backups is important so that our users are always able to restore saved data. Therefore Restic follows [Semantic Versioning](http://semver.org/) to clearly define which versions are compatible. The repository and data structures contained therein are considered the “Public API” in the sense of Semantic Versioning.

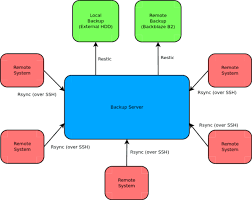
Contributing

Contributions are welcome! More information can be found in [the restic contribution guidelines](https://github.com/restic/restic/blob/master/CONTRIBUTING.md). A document describing the design of restic and the data structures stored on disc is contained in [the design document](http://restic.readthedocs.io/en/latest/100_references.html#design).

# Preparing a new repository

The place where your backups will be saved is called a “repository”. This is simply a directory containing a set of subdirectories and files created by restic to store your backups, some corresponding metadata and encryption keys.

ARCHITECTURE



CURRENT USAGE

1. Restic
2. Rest-server
3. Builder
4. Chunker
5. Restic.net

Technical details

TUNING BACKUP PARAMETERS

Restic offers a few parameters that allow tuning the backup. The default values should work well in general although specific use cases can benefit from different non-default values. As the restic commands evolve over time, the optimal value for each parameter can also change across restic versions.

## Disabling Backup Progress Estimation

When you start a backup, restic will concurrently count the number of files and their total size, which is used to estimate how long it will take. This will cause some extra I/O, which can slow down backups of network file systems or FUSE mounts. To avoid this overhead at the cost of not seeing a progress estimate, use the --no-scan option which disables this file scanning.

BACKEND CONNECTIONS

Restic uses a global limit for the number of concurrent connections to a backend. This limit can be configured using -o <backend-name>.connections=5, for example for the REST backend the parameter would be -o rest.connections=5. By default restic uses 5 connections for each backend, except for the local backend which uses a limit of 2. The defaults should work well in most cases.

OTHER INFORMATION

Project comparison

COMPARISON BETWEEN RESTIC AND BORG BACKUP

* Restic supports many different targets, including regular file systems SSH/SFTP targets and blob storage. Borg on the other hand only supports regular file systems/SSH targets.
* Borg supports compression using a variety of compression algorithms, Restic does not support this at all.
* Borg assumes a backup repository is used exclusively by one system. Restic allows multiple systems to use the same repository, which would lead to much greater deduplication when those systems store the same data.