RESTIX

Benutzerhandbuch

Version: 0.9.5

Inhaltsverzeichnis

Einführung

Restix basiert auf der Open Source Backup-Software restic. Es bietet eine grafische Oberfläche und eine Kommandozeilen-Schnittstelle für einen Teil des Funktionsumfangs von restic. Durch das Konzept von Sicherungszielen, die sich aus Zielverzeichnis, Sicherungsumfang und Zugangsdaten zusammensetzen wird die Bedienung von restic wesentlich komfortabler. Eine strikte Trennung der Sicherungen nach Rechner, Benutzer und Jahr reduziert die Gefahr von Inkonsistenzen deutlich.

Restix nimmt keinerlei Modifikationen an den restic-Daten vor, sodass restic für weitergehende Funktionen uneingeschränkt benutzt werden kann.

Die Software steht unter der MIT Lizenz und kann frei verwendet werden.

Installation

Systemvoraussetzungen

Restix läuft auf den Betriebssystemen Linux und Windows. Es benötigt Python 3 und die freie Backup-Software restic.

Debian-basiertes Linux (z.B. Ubuntu, Linux Mint, Debian)

Für diese Linux-Derivate stehen Installationspakete für die automatische Installation zur Verfügung. Falls hierbei Probleme auftreten oder die Installation nur lokal für einen Benutzer erfolgen soll, bitte der Beschreibung für die manuelle Installation im Abschnitt Andere Linux-Derivate folgen.

Vollständiges Paket installieren

Das vollständige Paket enthält sowohl die grafische Oberfläche als auch die Kommandozeilen-Version von restix.

Nach der Installation kann die grafische Oberfläche aus dem Startmenü, Rubrik Dienstprogramme bzw. Zubehör aufgerufen werden. Die Kommandozeilen-Version kann in einem Terminal mit dem Befehl `restix` gestartet werden.

Installationsschritte:

Paket restix-0.9.5.deb von https://github.com/FrankSommer-64/restix herunterladen.
Terminal öffnen und in das Verzeichnis mit der heruntergeladenen Datei wechseln.
Paket mit `sudo apt install ./restix-0.9.5.deb` installieren.

Minimales Paket installieren

Das minimale Paket enthält nur die Kommandozeilen-Version von restix.

Nach der Installation kann die Kommandozeilen-Version in einem Terminal mit dem Befehl `restix` gestartet werden.

Installationsschritte:

Paket restix_core-0.9.5.deb von https://github.com/FrankSommer-64/restix herunterladen.
Terminal öffnen und in das Verzeichnis mit der heruntergeladenen Datei wechseln.
Paket mit `sudo apt install ./restix_core-0.9.5.deb` installieren.

Red-Hat-basiertes Linux (z.B. Fedora)

Für diese Linux-Derivate stehen Installationspakete für die automatische Installation zur Verfügung. Falls hierbei Probleme auftreten oder die Installation nur lokal für einen Benutzer erfolgen soll, bitte der Beschreibung für die manuelle Installation im Abschnitt Andere Linux-Derivate folgen.

Vollständiges Paket installieren

Das vollständige Paket enthält sowohl die grafische Oberfläche als auch die Kommandozeilen-Version von restix.

Nach der Installation kann die grafische Oberfläche aus dem Startmenü, Rubrik Dienstprogramme bzw. Zubehör aufgerufen werden. Die Kommandozeilen-Version kann in einem Terminal mit dem Befehl `restix` gestartet werden.

Installationsschritte:

Paket restix-0.9.5.rpm von https://github.com/FrankSommer-64/restix herunterladen.
Terminal öffnen und in das Verzeichnis mit der heruntergeladenen Datei wechseln.
Paket mit `sudo dnf localinstall ./restix-0.9.5.rpm` installieren.

Minimales Paket installieren

Das minimale Paket enthält nur die Kommandozeilen-Version von restix.

Nach der Installation kann die Kommandozeilen-Version in einem Terminal mit dem Befehl `restix` gestartet werden.

Installationsschritte:

Paket restix_core-0.9.5.rpm von https://github.com/FrankSommer-64/restix herunterladen.
Terminal öffnen und in das Verzeichnis mit der heruntergeladenen Datei wechseln.
Paket mit `sudo dnf localinstall ./restix_core-0.9.5.rpm` installieren.

Andere Linux-Derivate (z.B. Arch, Manjaro)

Für andere Linux-Derivate stehen keine speziellen Installationspakete zur Verfügung, hier muss die Installation manuell durchgeführt werden.

Vollständiges Paket installieren

Das vollständige Paket enthält sowohl die grafische Oberfläche als auch die Kommandozeilen-Version von restix.

Nach der Installation kann die grafische Oberfläche aus dem Startmenü, Rubrik Dienstprogramme bzw. Zubehör oder per Icon auf dem Schreibtisch aufgerufen werden. Die Kommandozeilen-Version kann in einem Terminal mit dem Befehl `restix` gestartet werden.

Installationsschritte:

Paket restix-0.9.5-py3-none-any.whl von https://github.com/FrankSommer-64/restix herunterladen.
Skript install_full.sh von https://github.com/FrankSommer-64/restix herunterladen.
Terminal öffnen und in das Verzeichnis mit den heruntergeladenen Dateien wechseln.
Variablen im Installations-Skript anpassen, siehe Abschnitt Installations-Skript anpassen.
Soll die Installation nur lokal für einen Benutzer erfolgen, das Installations-Skript mit `./install_full.sh` starten.
Soll die Installation systemweit erfolgen, das Installations-Skript mit `sudo ./install_full.sh` starten.

Minimales Paket installieren

Das minimale Paket enthält nur die Kommandozeilen-Version von restix.

Nach der Installation kann die Kommandozeilen-Version in einem Terminal mit dem Befehl `restix` gestartet werden.

Installationsschritte:

Paket restix_core-0.9.5-py3-none-any.whl von https://github.com/FrankSommer-64/restix herunterladen.
Skript install_core.sh von https://github.com/FrankSommer-64/restix herunterladen.
Terminal öffnen und in das Verzeichnis mit den heruntergeladenen Dateien wechseln.
Variablen im Installations-Skript anpassen, siehe Abschnitt Installations-Skript anpassen.
Soll die Installation nur lokal für einen Benutzer erfolgen, das Installations-Skript mit `./install_core.sh` starten.
Soll die Installation systemweit erfolgen, das Installations-Skript mit `sudo ./install_core.sh` starten.

Installations-Skript anpassen

Am Anfang des Installations-Skripts werden drei Variablen definiert, die die Installation bestimmen.

Variable **`INSTALL_PATH`** legt das Verzeichnis fest, in dem die restix-Programmdateien und das virtuelle Python-Environment erstellt werden sollen.

Variable **`LINK_PATH`** legt das Verzeichnis fest, in dem symbolische Links auf die ausführbaren restix-Dateien angelegt werden sollen. Das Verzeichnis sollte im Pfad des Benutzers liegen, damit das System die restix-Applikation finden kann.

Variable **`SHORTCUT_PATH`** legt das Verzeichnis fest, in dem die Desktop-Verknüpfungen zur restix GUI angelegt werden sollen. Bei systemweiter Installation sollte das Verzeichnis `/usr/local/share/applications` gewählt werden, bei lokaler Benutzerinstallation das Verzeichnis `$HOME/.local/applications`. In beiden Fällen ist die restix-GUI damit über das Startmenü aufrufbar. Bei Wert `$HOME/Schreibtisch` wird ein Icon auf dem Schreibtisch angelegt, die GUI ist dann über einen Doppelklick des Icons aufrufbar. Für die Minimal-Installation hat die Variable keine Bedeutung.

Windows

Vollständiges Paket installieren

tbd.

Minimales Paket installieren

tbd.

Konfiguration

Konfigurationsverzeichnis

Restix erwartet alle lokalen Einstellungen in einem Verzeichnis, standardmäßig in **`.config/restix`** unter dem Home-Verzeichnis des Benutzers. Optional kann Umgebungsvariable **`RESTIX_CONFIG_PATH`** gesetzt werden, dann werden die Einstellungen in diesem Verzeichnis gesucht.

Konfigurationsdatei

Die zentrale Konfiguration erfolgt in Datei **`config.toml`** im Konfigurationsverzeichnis. Das Dateiformat ist TOML, das einen guten Kompromiss zwischen Lesbarkeit und Eignung für maschinelle Verarbeitung bietet. Die Datei kann entweder mit einem Texteditor oder mit der grafischen Oberfläche von restix bearbeitet werden.

Variablen

In der Konfigurationsdatei können folgende Variablen benutzt werden:

`${HOME}` – wird durch das Home-Verzeichnis des Benutzers ersetzt (ohne abschließendes Verzeichnis-Trennzeichen)

`${USER}` – wird durch den Namen des Benutzers ersetzt

Verweise auf andere Dateien

Bestimmte Einstellungen benötigen weitere Dateien, wie z.B. Passwort oder Liste der zu sichernden Elemente. Erfolgt der Verweis mit relativem Pfad, so muss sich die Datei unterhalb des Konfigurationsverzeichnisses befinden.

Pfad zum restic-Programm

Das von restix zu verwendende restic-Programm kann mit Parameter **`restic`** festgelegt werden:

```
restic = "${HOME}/bin/restic"
```

Zugangsdaten

Alle gesicherten Daten werden verschlüsselt im Repository abgelegt. Für den Umgang mit der Verschlüsselung bietet restix vier verschiedene Möglichkeiten, die in den nachfolgenden Beispielen aufgeführt sind. Jede Definition von Zugangsdaten muss in einem Block **`credentials`** mit eindeutigem Aliasnamen erfolgen, mit der diese vom Sicherungsziel referenziert werden kann.

Bitte beachten, dass automatisierte Sicherungen ohne Benutzer-Interaktion nur mit den Typen Passwort im Klartext oder Passwort-Datei möglich sind.

Passwort im Klartext:

```
[[credentials]]
alias = "pwd"
comment = "Passwort im Klartext"
type = "password"
value = "geheim"
```

Passwortdatei:

```
[[credentials]]
alias = "standard"
comment = "Passwort-Datei"
type = "file"
value = "pwfile.txt"
```

Prompt:

```
[[credentials]]
alias = "Prompt"
comment = "Passwort-Abfrage"
type = "prompt"
```

PGP mit Passphrase:

```
[[credentials]]
alias = "PGP-Passphrase"
comment = "PGP-verschlüsselte Passwortdatei mit Passphrase"
type = "pgp"
value = "pgp_phrase.asc"
```

PGP mit FIDO2-Token:

```
[[credentials]]
alias = "PGP-Token"
comment = "PGP-verschlüsselte Passwortdatei mit FIDO2-Token"
type = "pgp"
value = "pgp_token.gpg"
```

Sicherungsumfänge

Das Festlegen, welche Dateien gesichert werden sollen muss in einem Block **`scope`** mit eindeutigem Aliasnamen erfolgen, mit der diese vom Sicherungsziel referenziert werden kann.

Parameter **`includes`** verweist auf eine Datei mit den Namen der zu sichernden Verzeichnisse und Dateien.

Optional kann mit Parameter **`excludes`** auf eine Datei verwiesen werden, die einzelne Verzeichnisse oder Dateien von der Sicherung ausnimmt.

Optional kann mit Parameter **`ignores`** eine Liste von Mustern angegeben werden, dann werden passende Elemente von der Sicherung ausgenommen.

Beispiel:

```
[[scope]]
alias = "minimal"
comment = "Minimaler Backup-Umfang für Upload ins Internet"
includes = "minimal.list"
excludes = "minimal_excludes.list"
ignores = [
    ".git",
    ".idea",
    ".pytest_cache",
    ".venv",
    "__pycache__",
]
```

Sicherungsziele

Die Definition von Sicherungszielen muss in einem Block **`target`** mit eindeutigem Aliasnamen erfolgen, mit der diese von der Kommandozeile bzw. der grafischen Oberfläche ausgewählt werden kann.

Parameter **`location`** verweist auf ein Verzeichnis oder die URL eines Servers, der per sftp-Protokoll angesprochen wird.

Parameter **`credentials`** enthält den Aliasnamen der zu verwendenden Zugangsdaten.

Parameter **`scope`** enthält den Aliasnamen der zu verwendenden Sicherungsumfangs.

Beispiel mit Verzeichnis:

```
[[target]]
alias = "extssd"
comment = "externe SSD"
location = "/media/${USER}/58af5a30-36b5-a70683ae3e7e/restix"
scope = "vollstaendig"
credentials = "standard"
```

Beispiel mit Server-URL:

```
[[target]]
alias = "heimserver"
comment = "Heim-Server"
location = "sftp:restic_localsrv:data"
scope = "minimal"
credentials = "standard"
```

Grafische Oberfläche

Die grafische Oberfläche kann aus dem Startmenü heraus aufgerufen werden, je nach Betriebssystem findet sie sich in Rubrik Dienstprogramme oder Zubehör. Vom Terminal aus muss **`grestix`** eingegeben werden.

Kommandozeile

Die Kommandozeilen-Schnittstelle von restix wird durch Eingabe des Befehls **`restix`** im Terminal aufgerufen. Die Schnittstelle ist hauptsächlich zum automatischen Ausführen von Aufgaben gedacht.

Befehle

Version anzeigen

`restix --version` gibt die installierte Version aus: `restix-Version 0.9.5`

Sicherungsziele anzeigen

`restix targets` gibt Aliasname und Kurzbeschreibung aller Sicherungsziele aus, die in der Konfigurationsdatei definiert sind. Falls keine Konfigurationsdatei existiert, wird eine Fehlermeldung ausgegeben.

Beispiel:

```
restix targets

Sicherungsziele:
heimserver – Heim-Server
sticka – USB Stick Backup-A
stickb – USB Stick Backup-B
```

Repository anlegen

`restix init` ***`Aliasname`*** erzeugt ein neues restic-Repository im Sicherungsziel mit dem angegebenen Aliasnamen. Das Repository wird im Unterverzeichnis Benutzername/Hostname/Jahr angelegt. Zur Verschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet.

Option **`--batch`** führt den Befehl sofort aus; ohne Angabe der Option muss der Befehl vom Benutzer bestätigt werden.

Beispiel für Benutzer ***karl***, Host ***notebook***, Jahr ***2025***, Sicherungsziel ***sticka*** mit URL ***`/media/karl/Backup-A/restix`***:

```
restix init --batch sticka
```

erzeugt ohne Rückfrage ein restic-Repository im Verzeichnis `/media/karl/Backup-A/restix/karl/notebook/2025`

Daten sichern

`restix backup` ***`Aliasname`*** sichert lokale Daten im Sicherungsziel mit dem angegebenen Aliasnamen. Zur Verschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet. Zur Ermittlung der zu sichernden Daten wird der in der Konfigurationsdatei hinterlegte Umfang verwendet.

Option **`--batch`** führt den Befehl sofort aus; ohne Angabe der Option muss der Befehl vom Benutzer bestätigt werden.

Option **`--dry-run`** zeigt nur an, was gesichert werden würde, überträgt aber keine Daten zum restic-Repository.

Option **`--auto-create`** erzeugt automatisch ein restic-Repository, falls noch keines existiert. Kann nur verwendet werden, falls restic Version 0.17 oder höher installiert ist.

Daten wiederherstellen

`restix restore` ***`Aliasname`*** kopiert lokale Daten aus dem Sicherungsziel mit dem angegebenen Aliasnamen ins lokale Dateisystem. Zur Entschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet.

Option **`--batch`** führt den Befehl sofort aus; ohne Angabe der Option muss der Befehl vom Benutzer bestätigt werden.

Option **`--dry-run`** zeigt nur an, was aus dem restic-Repository kopiert werden würde, überträgt aber keine Daten ins lokale Dateisystem.

Option **`--snapshot`** legt die ID des Snapshots im restic-Repository fest, aus dem die Daten übertragen werden sollen. Standardmäßig wird ID ***`latest`*** verwendet.

Option **`--restore-path`** legt das lokale Wurzelverzeichnis fest, in das die Daten aus dem restic-Repository kopiert werden sollen. Standardmäßig werden die Dateien an ihren Ursprungsort kopiert.

Option **`--host`** legt fest, dass die Daten aus dem restic-Repository für den angegebenen Hostnamen übertragen werden sollen. Standardmäßig wird der lokale Hostname verwendet.

Option **`--year`** legt fest, dass die Daten aus dem restic-Repository für das angegebene Jahr übertragen werden sollen. Standardmäßig wird das aktuelle Jahr verwendet.

Snapshots im Sicherungsziel auflisten

`restix snapshots` ***`Aliasname`*** zeigt alle im Sicherungsziel mit dem angegebenen Aliasnamen enthaltene Snapshots an. Zur Entschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet.

Option **`--host`** legt fest, dass das restic-Repository für den angegebenen Hostnamen betrachtet werden soll. Standardmäßig wird der lokale Hostname verwendet.

Option **`--year`** legt fest, dass das restic-Repository für das angegebene Jahr betrachtet werden soll. Standardmäßig wird das aktuelle Jahr verwendet.

Dateien im Sicherungsziel auflisten

`restix ls *Aliasname*` zeigt alle im Sicherungsziel mit dem angegebenen Aliasnamen enthaltene Dateien an. Zur Entschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet.

Option **`--snapshot`** legt die ID des Snapshots im restic-Repository fest. Standardmäßig wird ID ***`latest`*** verwendet.

Option **`--host`** legt fest, dass das restic-Repository für den angegebenen Hostnamen betrachtet werden soll. Standardmäßig wird der lokale Hostname verwendet.

Option **`--year`** legt fest, dass das restic-Repository für das angegebene Jahr betrachtet werden soll. Standardmäßig wird das aktuelle Jahr verwendet.

Dateien im Sicherungsziel suchen

`restix find *Aliasname*` zeigt alle im Sicherungsziel mit dem angegebenen Aliasnamen enthaltene Dateien an, die auf ein angegebenes Suchmuster passen. Zur Entschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet.

Option **`--pattern`** legt das Suchmuster für die Elemente fest. Diese Option muss angegeben werden. Die Pattern-Syntax ist mit der von restic identisch.

Option **`--snapshot`** legt die ID des Snapshots im restic-Repository fest. Standardmäßig wird ID ***`latest`*** verwendet.

Option **`--host`** legt fest, dass das restic-Repository für den angegebenen Hostnamen betrachtet werden soll. Standardmäßig wird der lokale Hostname verwendet.

Option **`--year`** legt fest, dass das restic-Repository für das angegebene Jahr betrachtet werden soll. Standardmäßig wird das aktuelle Jahr verwendet.

Repository entsperren

`restix unlock *Aliasname*` entsperrt ein restic-Repository im Sicherungsziel mit dem angegebenen Aliasnamen. Zur Entschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet. Es kommt hin und wieder vor, dass restic bei einem vorangegangenen Befehl eine Sperre nicht korrekt entfernt hat, dann kommt dieser Befel zur Anwendung.

Option **`--batch`** führt den Befehl sofort aus; ohne Angabe der Option muss der Befehl vom Benutzer bestätigt werden.

Option **`--host`** legt fest, dass das restic-Repository für den angegebenen Hostnamen entsperrt werden soll. Standardmäßig wird der lokale Hostname verwendet.

Option **`--year`** legt fest, dass das restic-Repository für das angegebene Jahr entsperrt werden soll. Standardmäßig wird das aktuelle Jahr verwendet.

Repository aufräumen

`restix cleanup` ***`Aliasname`*** entfernt alle im Sicherungsziel mit dem angegebenen Aliasnamen enthaltene Snapshots bis auf den letzten eines Monats. Zur Entschlüsselung werden die in der Konfigurationsdatei für das Sicherungsziel hinterlegten Zugangsdaten verwendet. Der Befehl kann z.B. am Jahresende verwendet werden, um ein Repository vor der Archivierung zu verkleinern.

Option `--`**`batch`** führt den Befehl sofort aus; ohne Angabe der Option muss der Befehl vom Benutzer bestätigt werden.

Option **`--dry-run`** zeigt nur an, welche Snapshots aus dem restic-Repository entfernt werden würden, führt aber keine Veränderungen aus.

Option **`--host`** legt fest, dass das restic-Repository für den angegebenen Hostnamen betrachtet werden soll. Standardmäßig wird der lokale Hostname verwendet.

Option **`--year`** legt fest, dass das restic-Repository für das angegebene Jahr betrachtet werden soll. Standardmäßig wird das aktuelle Jahr verwendet.