

Versionierung: git

Nachfolgend soll die Versionierung mit [GIT](#) beschrieben werden. Hintergrund ist Vorhalten mehrerer Versionen einer Datei oder bestimmter Verzeichnisse (zum Beispiel /etc), damit Änderungen festgehalten werden.

Installation

Für DEBIAN stehen die notwendigen Pakete im Repository bereit:

```
~# apt-get install git
```

Erste Versionierung

Für die erste Versionierung eines bestimmten Verzeichnisses (nachfolgend das Verzeichnis /etc) sind mehrere Schritte notwendig.

- Wechsel in das zu versionierende Verzeichnis:

```
~# cd /etc/
```

- Initialisierung des Repository

```
~# git init
```

- Hinzufügen aller Dateien im Verzeichnis:

```
~# git add .
```

- Hochladen der Dateien ins Repository:

```
~# git commit -m "Eine Beschreibung hinzufügen"
```

Arbeitsweise

Alle Änderungen die im Verzeichnis vorgenommen werden, müssen in das GIT-Repository aufgenommen werden. Nachfolgend ein kleiner Auszug der Arbeitsweise mit GIT.

- Anzeigen des Repository-Status (direkt nach dem Hochladen):

```
~# git status
# On branch master
nothing to commit (working directory clean)
```

- Anzeigen des Repository-Status nach Änderung einer Datei (zum Beispiel /etc/hosts):

```
git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
directory)
#
#   modified:   hosts
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- Hochladen der geänderten Datei in das Repository:

```
~# git add /etc/hosts
~# git commit -m "Änderung der Datei '/etc/hosts'"
[master f05a12c] Änderung der Datei '/etc/hosts'
...
```

- Herunterladen einer Datei aus dem Repository (Überschreiben der lokalen Datei) oder Wiederherstellen der lokal gelöschten Datei aus dem Repository:

```
~# git checkout -- /etc/hosts
```

- Anzeigen der Unterschiede zwischen lokalen Dateien und den Repository-Daten:

```
~# git diff
diff --git a/hosts b/hosts
index 3126eb2..812e530 100644
--- a/hosts
```

```
+++ b/hosts
@@ -1,4 +1 @@
 192.168.10.135 usb
-127.0.0.1      localhost
```

- Aktualisieren des Repository, wenn lokal Dateien gelöscht wurden:

```
~# git add -u .
~# git commit -m "Lokale Datei gelöscht"
```

- Hochladen der Änderungen zum entfernten Repository:

```
~# git push
```

- Herunterladen der Änderungen vom entfernten Repository:

```
~# git pull origin master
```

Es wird der 'master'-Zweig heruntergeladen.

Überwachung

Soll das Verzeichnis überwacht werden, damit alle Änderungen automatisch ins Repository hochgeladen werden, hilft das Programm **watcher**, welches [hier](#) beschrieben ist. Die Konfigurationsdatei `/etc/watcher.ini` wird um zwei Überwachungen ergänzt.

- Überwachung auf Erstellen und Verändern einer Datei:

```
[etc-git-add]
watch=/etc
events=create,modify,write_close
excluded=/etc/.git
recursive=true
autoadd=true
command=cd /etc/ && git add $filename && git commit -m "Erstellen oder
Verändern der Datei: $filename"
```

- Überwachung auf Löschen einer Datei:

```
[etc-git-remove]
watch=/etc
events=delete
excluded=/etc/.git
recursive=true
autoadd=true
command=cd /etc/ && git add -u $filename && git commit -m "Löschen der
Datei: $filename"
```

- Neustart des Dienstes, damit die Änderungen wirksam werden:

```
~# /etc/init.d/watcher.py restart
```

Datensicherung und -wiederherstellung

Die Sicherung und wiederherstellung eines komplettes Repository, geht über den Befehl „**git bundle**“.

- Repository in eine Datei sichern:

```
~# git bundle create <Dateiname> master
Counting objects: 1421, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (1369/1369), done.
Writing objects: 100% (1421/1421), 152.35 KiB, done.
Total 1421 (delta 977), reused 0 (delta 0)
```

Voraussetzung für das erfolgreiche durchführen des Befehls ist, dass im aktuellen Verzeichnis ein GIT-Repository vorhanden ist.

- Wiederherstellen des Repository:

```
git clone <Dateiname> -b master .
Cloning into '.'...
Receiving objects: 100% (1421/1421), 152.35 KiB, done.
Resolving deltas: 100% (977/977), done.
```

Die Dateien werden im aktuellen Verzeichnis wiederhergestellt.

[DEBIAN](#), [GIT](#), [Backup](#), [Restore](#)

From:

<https://looper.de/wiki/> - **Linux4Ever**

Permanent link:

<https://looper.de/wiki/doku.php?id=software:git>

Last update: **2025/12/11 15:00**

