

Getting ready

We have a directory that contains several files and subdirectories. We need to keep track of changes occurring to the directory contents and back them up. If data becomes corrupted or goes missing, we must be able to restore a previous copy of that data. We will either need to take the backup at different locations in the local machine, or to a remote machine. To install git, just use your distro's package manager and let's get started:

How to do it...

Let's see how to use git to version control data, in this case backups:

1. In the directory which is to be backed up use:

```
$ cd /home/data/source
```


Let it be the directory source to be tracked.
2. Set up and initiate the remote backup directory. In the remote machine, create the backup destination directory:

```
$ mkdir -p /home/backups/backup.git  
$ cd /home/backups/backup.git  
$ git init --bare
```

The following steps are to be performed in the source host machine:

1. Add user details to Git in the source host machine:

```
$ git config --global user.name "Sarath Lakshman"  
  
$ git config --global user.email slynux@slynux.com
```
2. Initiate the source directory to backup from the host machine. In the source directory in the host machine whose files are to be backed up, execute the following commands:

```
$ git init  
Initialized empty Git repository in /home/backups/backup.git/  
  
$ git commit --allow-empty -am "Init"  
[master (root-commit) b595488] Init
```

3. In the source directory, execute the following command to add the remote git directory and synchronize backup:

```
$ git remote add origin user@remotehost:/home/backups/backup.git
```

```
$ git push origin master
```

```
Counting objects: 2, done.
```

```
Writing objects: 100% (2/2), 153 bytes, done.
```

```
Total 2 (delta 0), reused 0 (delta 0)
```

```
To user@remotehost:/home/backups/backup.git
```

```
* [new branch]      master -> master
```

4. Add or remove files for Git tracking.

The following command adds all files and folders in the current directory to the backup list:

```
$ git add *
```

We can conditionally add certain files only to the backup list as follows:

```
$ git add *.txt
```

```
$ git add *.py
```

We can remove the files and folders not required to be tracked by using:

```
$ git rm file
```

It can be a folder or even a wildcard as follows:

```
$ git rm *.txt
```

5. Check-pointing or marking backup points.

We can mark checkpoints for the backup with a message using the following command:

```
$ git commit -m "Commit Message"
```

We need to update the backup at the remote location at regular intervals. Hence, set up a cron job (for example, backing up every five hours):

Create a file crontab entry with lines:

```
0 */5 * * * /home/data/backup.sh
```

Create a script /home/data/backup.sh as follows:

```
#!/bin/ bash
```

```
cd /home/data/source
```

```
git add .
```

```
git commit -am "Backup taken at @ $(date)"
```

```
git push
```