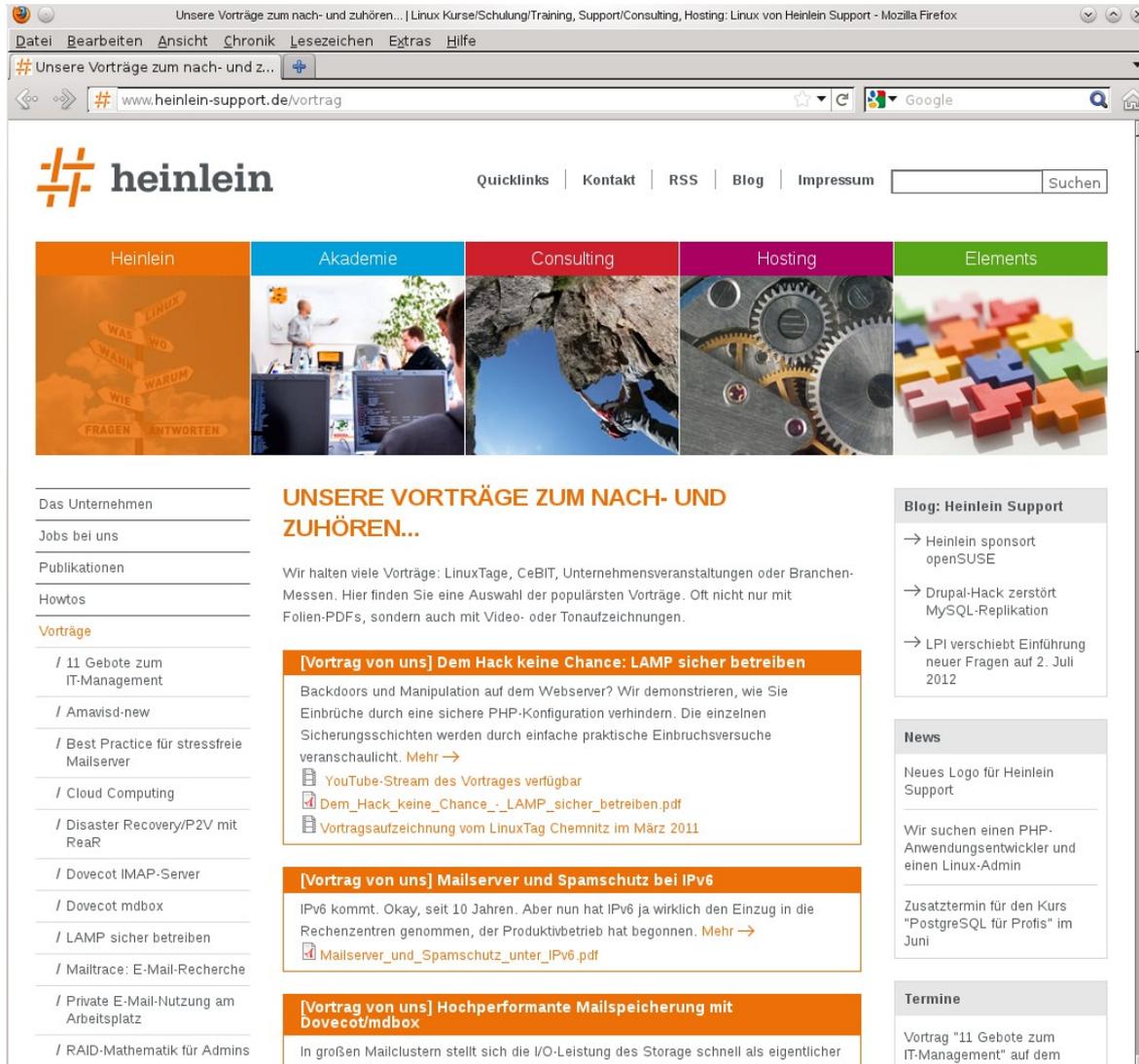


# **Igitt?! Git für Anfängerinnen und Anfänger**



Unsere Vorträge zum nach- und zuhören... | Linux Kurse/Schulung/Training, Support/Consulting, Hosting: Linux von Heinlein Support - Mozilla Firefox

File Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

# Unsere Vorträge zum nach- und zuhören...

www.heinlein-support.de/vortrag

heinlein Quicklinks Kontakt RSS Blog Impressum Suchen

Heinlein Akademie Consulting Hosting Elements

Das Unternehmen  
 Jobs bei uns  
 Publikationen  
 Howtos  
**Vorträge**

- / 11 Gebote zum IT-Management
- / Amavisd-new
- / Best Practice für stressfreie Mailserver
- / Cloud Computing
- / Disaster Recovery/P2V mit ReaR
- / Dovecot IMAP-Server
- / Dovecot mbox
- / LAMP sicher betreiben
- / Mailtrace: E-Mail-Recherche
- / Private E-Mail-Nutzung am Arbeitsplatz
- / RAID-Mathematik für Admins

**UNSERE VORTRÄGE ZUM NACH- UND ZUHÖREN...**

Wir halten viele Vorträge: LinuxTage, CeBIT, Unternehmensveranstaltungen oder Branchen-Messen. Hier finden Sie eine Auswahl der populärsten Vorträge. Oft nicht nur mit Folien-PDFs, sondern auch mit Video- oder Tonaufzeichnungen.

**[Vortrag von uns] Dem Hack keine Chance: LAMP sicher betreiben**

Backdoors und Manipulation auf dem Webserver? Wir demonstrieren, wie Sie Einbrüche durch eine sichere PHP-Konfiguration verhindern. Die einzelnen Sicherungsschichten werden durch einfache praktische Einbruchsversuche veranschaulicht. [Mehr →](#)

- YouTube-Stream des Vortrages verfügbar
- Dem\_Hack\_keine\_Chance\_-\_LAMP\_sicher\_betreiben.pdf
- Vortragsaufzeichnung vom LinuxTag Chemnitz im März 2011

**[Vortrag von uns] Mailserver und Spamschutz bei IPv6**

IPv6 kommt. Okay, seit 10 Jahren. Aber nun hat IPv6 ja wirklich den Einzug in die Rechenzentren genommen, der Produktivbetrieb hat begonnen. [Mehr →](#)

- Mailserver\_und\_Spamschutz\_unter\_IPv6.pdf

**[Vortrag von uns] Hochperformante Mailspeicherung mit Dovecot/mbox**

In großen Mailclustern stellt sich die I/O-Leistung des Storage schnell als eigentlicher

**Blog: Heinlein Support**

- Heinlein sponsort openSUSE
- Drupal-Hack zerstört MySQL-Replikation
- LPI verschiebt Einführung neuer Fragen auf 2. Juli 2012

**News**

Neues Logo für Heinlein Support

Wir suchen einen PHP-Anwendungsentwickler und einen Linux-Admin

Zusatztermin für den Kurs "PostgreSQL für Profis" im Juni

**Termine**

Vortrag "11 Gebote zum IT-Management" auf dem

**Ja, diese Folien stehen auch als PDF im Netz...**  
<https://www.heinlein-support.de/vortrag>

## Einsatzbereiche von git

- Kollaboratives Arbeiten an allen Arten von Inhalten
  - Code
  - Konfigurationsdateien
  - Texte
- Als Backup für die eigene Arbeit
- Für Installation und Updates von Software (statt .tar.gz)

## Git-Plattformen

- Github.com (SaaS, private Repos kostenpflichtig)
- Bitbucket.com (Atlassian, bis 5 Accounts kostenlos, auch private Repos)
- Selbst hosten mit gitlab (freie Software in Ruby, MIT-Lizenz)
- Dieser Vortrag bezieht sich auf github.com, das meiste ist jedoch git, also übertragbar.

## Der Ablauf bei Projekten auf github

- Account erstellen
- Das Originalprojekt kopieren (Webinterface: Klick auf „Fork“)
- Die eigene Kopie runterladen
- Darin Änderungen vornehmen
- Änderungen lokal in git speichern
- Änderungen hochladen in die eigene Kopie bei github.com
- Pull Request stellen ans Originalrepo (Webinterface)

## Lokale Konfiguration

- `aptitude install git(-core)`
- Konfigs systemweit / pro Useraccount / pro Git-Repo
- `ich@tux:~/testrepo $ cat ~/.gitconfig`

```
[user]
    name = Silke Meyer
    email = s.meyer@helein-hosting.de
[core]
    editor = vim
[merge]
    tool = meld
```

## Lokale Konfiguration

- Upload (git push) per ssh vorbereiten:
  - Webinterface: ssh public key hochladen
  - `ich@tux:~/testrepo $ cat ~/.ssh/config`  
`Host = github.com`  
`IdentityFile /home/ich/.ssh/mykey`
  - `ich@tux:~/testrepo $ ssh -T git@github.com`  
`Hi silke! You've successfully authenticated, but GitHub does not provide shell access.`
- Protipp: aktuellen Branch im Shell Prompt anzeigen lassen
  - `ich@tux:~/testrepo (master) $`
  - Link zu [Beispiel für die Bash](#)

# Commit

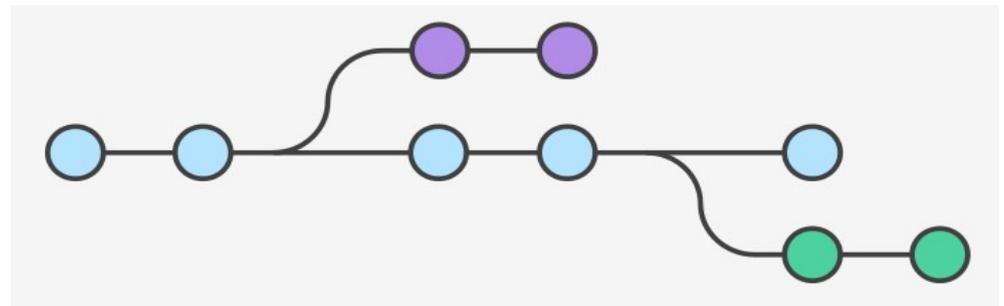
- Eine in git gespeicherte Änderung
  - Der Zustand des ganzen Repos zu einem Zeitpunkt
  - Metadaten im git log: eindeutige Commit-ID, Autor, Zeitstempel, Commit Message

```
→ ich@tux:~/testrepo $ git log
commit 55bc83dffc398ffe8e1466262993c8c1bd342f64
Author: Silke Meyer <silke@silkemeyer.net>
Date: Tue Mar 15 12:11:13 2016 +0100
```

Abkürzung ausgeschrieben

## Branches

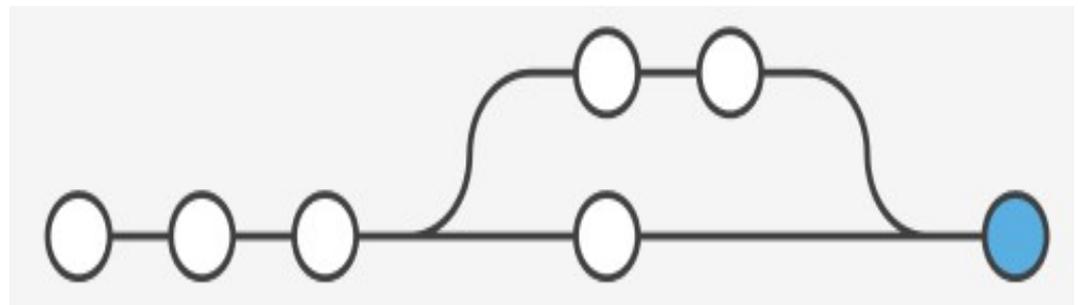
- „master“ (der Hauptzweig)
- Eigene Branches mit beliebigen Namen („development“, „feature“)
- `ich@tux:~/testrepo $ git checkout -b veranstaltungsinfos`  
Switched to a new branch 'veranstaltungsinfos'
- Absprache im Team: Wie arbeiten wir mit Branches? (git-Workflows)



<https://www.atlassian.com/git/tutorials/using-branches>

## Mergen

- Das Zusammenführung von Branches
- Einen anderen Branch in den aktuellen Branches „mergen“
- `ich@tux:~/testrepo $ git checkout master`  
`ich@tux:~/testrepo $ git merge veranstaltungsinfos`
- Pull Request: „Bitte meine Änderung aufnehmen/mergen!“



<https://www.atlassian.com/git/tutorials/using-branches/git-merge>

## Orte und Zustände

### → Serverseitig

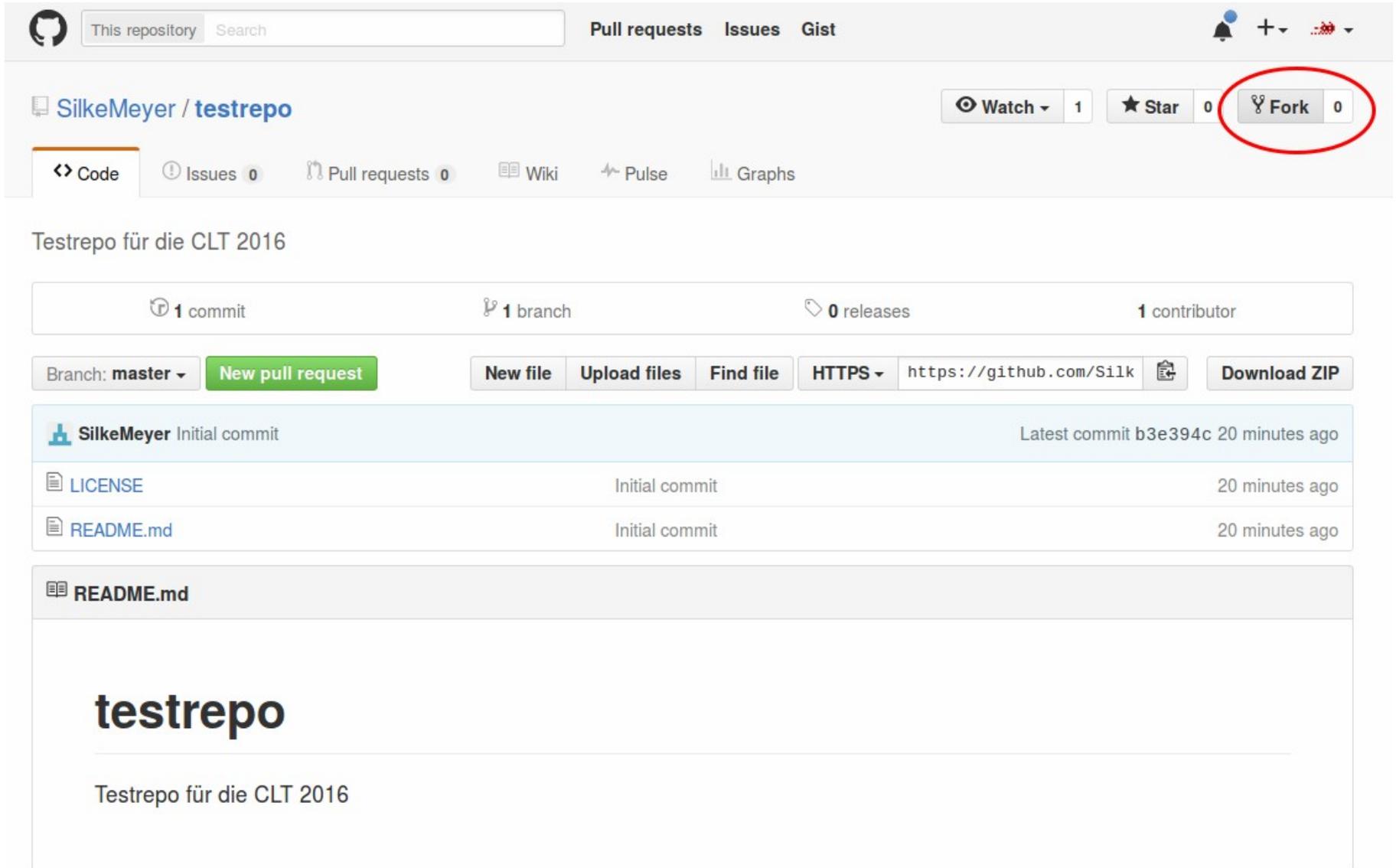
- **Upstream**: das Repo des Originalprojektes auf Github
- **Origin**: meine Kopie (Fork) des Originalprojektes auf Github

### → Lokal

- **Lokales Repository**: der lokale Klon des Origin plus meine lokalen (nicht hochgeladenen) Commits
- **Staging Area / Index**: alles, was ich lokal in meinen nächsten Commit aufnehmen möchte / was ich unter Versionskontrolle stelle (`git add <datei>`)
- **Working Copy**: meine Arbeitsdateien, inkl. aller uncommitteter Änderungen und Branches, auch Dateien im Verzeichnis, die nicht unter Versionskontrolle stehen

## Nochmal: Unser Commit

- Das Originalrepo forken (1 Klick)
- Den eigenen Fork runterladen (`git clone <url>`)
- einen Branch anlegen (`git checkout -b <name>`)
- Änderungen vornehmen
- Änderungen für den nächsten Commit vormerken (`git add <dateien>`)
- Committen (`git commit -m „Kommentar“`)
- Hochladen (`git push origin <branchname>`)
- Pull request stellen (im Webinterface)



The screenshot shows the GitHub interface for the repository 'SilkeMeyer / testrepo'. At the top, there is a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. On the right, there are icons for notifications, a dropdown menu, and a repository icon. Below this, the repository name 'SilkeMeyer / testrepo' is displayed, along with 'Watch' (1), 'Star' (0), and 'Fork' (0) buttons. The 'Fork' button is circled in red. Below the repository name, there are links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', and 'Graphs'. The main content area shows the repository title 'Testrepo für die CLT 2016' and statistics: '1 commit', '1 branch', '0 releases', and '1 contributor'. Below this, there are buttons for 'New pull request', 'New file', 'Upload files', 'Find file', 'HTTPS' (with a dropdown arrow), the repository URL 'https://github.com/Silk', and 'Download ZIP'. A commit history table follows, showing 'SilkeMeyer Initial commit' as the latest commit (b3e394c, 20 minutes ago). Below the commit history, there is a section for 'README.md' which contains the repository name 'testrepo' and the description 'Testrepo für die CLT 2016'.

This repository Search

Pull requests Issues Gist

SilkeMeyer / testrepo

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs

Testrepo für die CLT 2016

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

New file Upload files Find file HTTPS https://github.com/Silk Download ZIP

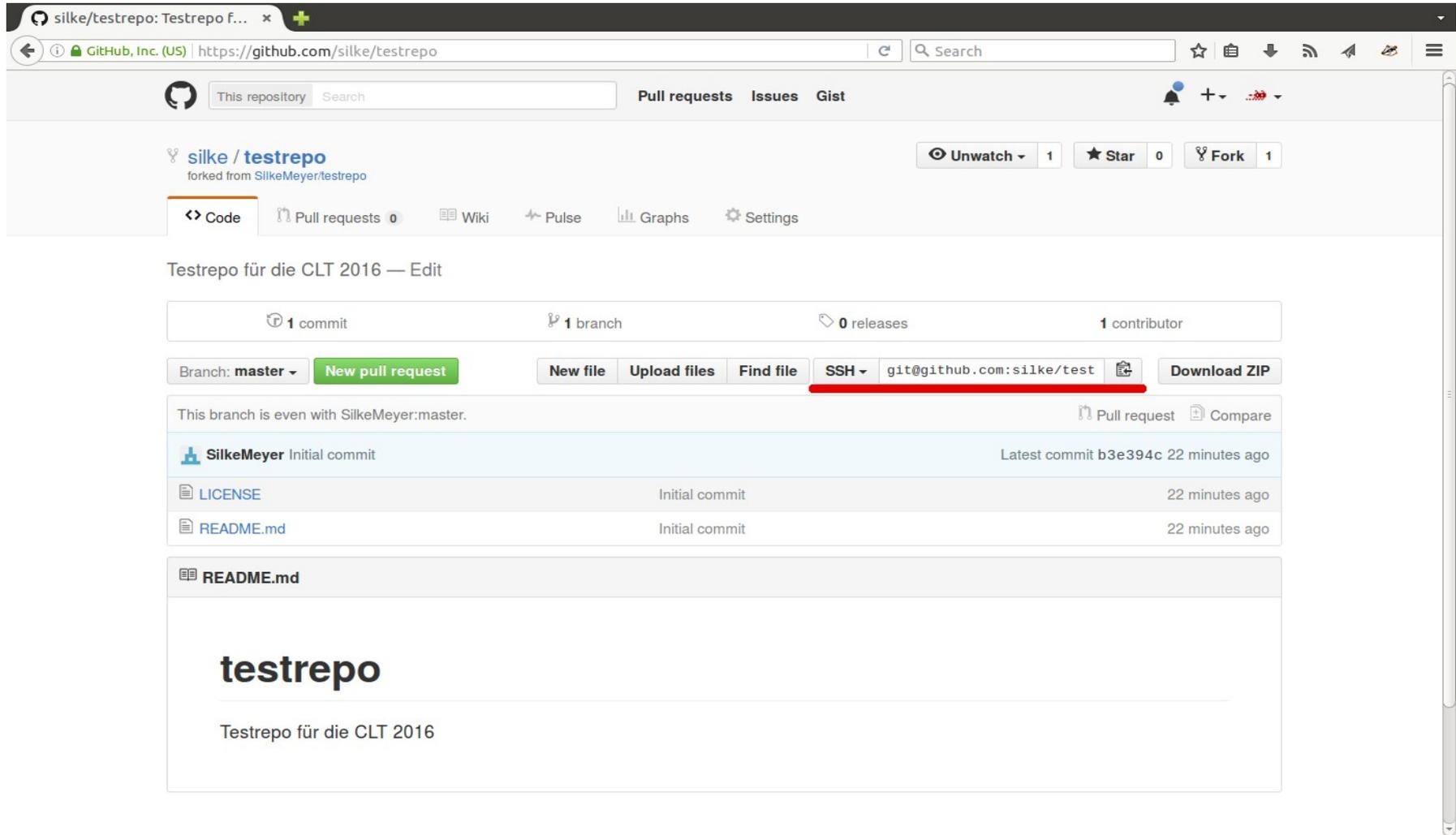
SilkeMeyer Initial commit Latest commit b3e394c 20 minutes ago

LICENSE	Initial commit	20 minutes ago
README.md	Initial commit	20 minutes ago

README.md

# testrepo

Testrepo für die CLT 2016



The screenshot shows a web browser window displaying the GitHub repository page for 'silke/testrepo'. The browser's address bar shows the URL 'https://github.com/silke/testrepo'. The repository page includes a search bar, navigation links for 'Pull requests', 'Issues', and 'Gist', and a header for the repository 'silke / testrepo' with 'Unwatch', 'Star', and 'Fork' buttons. Below the header, there are tabs for 'Code', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area shows the repository name 'Testrepo für die CLT 2016' and a summary of the repository: '1 commit', '1 branch', '0 releases', and '1 contributor'. A 'New pull request' button is visible. The 'SSH' button is highlighted with a red underline, and the corresponding SSH URL 'git@github.com:silke/test' is displayed. Below this, a table lists the repository's files: 'LICENSE' and 'README.md', both with 'Initial commit' and '22 minutes ago' timestamps. The 'README.md' file content is displayed below the table, showing the repository name 'testrepo' and the description 'Testrepo für die CLT 2016'.

silke/testrepo: Testrepo f... x

GitHub, Inc. (US) | https://github.com/silke/testrepo

This repository Search Pull requests Issues Gist

silke / testrepo  
forked from SilkeMeyer/testrepo

Unwatch 1 Star 0 Fork 1

Code Pull requests 0 Wiki Pulse Graphs Settings

Testrepo für die CLT 2016 — Edit

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request New file Upload files Find file SSH git@github.com:silke/test Download ZIP

This branch is even with SilkeMeyer:master. Pull request Compare

SilkeMeyer Initial commit	Latest commit b3e394c 22 minutes ago
LICENSE Initial commit	22 minutes ago
README.md Initial commit	22 minutes ago

README.md

# testrepo

Testrepo für die CLT 2016

## Das erste Herunterladen

- `ich@tux:~/$ git clone  
git@github.com:silke/testrepo.git  
Cloning into 'testrepo'...  
remote: Counting objects: 4, done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 4 (delta 0), reused 4 (delta 0), pack-  
reused 0  
Receiving objects: 100% (4/4), 12.14 KiB | 0 bytes/s,  
done.  
Checking connectivity... done.`
- `ich@tux:~/$ cd testrepo`
- `ich@tux:~/testrepo (master) $`

## Branch anlegen

- `ich@tux:~/testrepo $ git branch -v`  
\* master                                   b3e394c Initial commit  
...                                           ...
- `ich@tux:~/testrepo $ git checkout -b veranstaltungsinfos`  
Switched to a new branch 'veranstaltungsinfos'
- `ich@tux:~/testrepo $ git branch -v`  
  master                                   b3e394c ...  
\* veranstaltungsinfos                   b3e394c  
...                                           ...
- `ich@tux:~/testrepo $ git checkout master`  
Switched to branch 'master'

## Nach Änderung: Status prüfen

→ `ich@tux:~/testrepo (veranstaltungsinfos)$ git status`

**On branch `veranstaltungsinfos`**

**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:     `README.md`**

**Untracked files:**

(use "`git add <file>...`" to include in what will be committed)

**`programm.txt`**

## Diff prüfen

```
→ ich@tux:~/testrepo (veranstaltungsinfos)$ git diff
diff --git a/README.md b/README.md
index edb7f85..6201877 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,5 @@
 # testrepo
 Testrepo für die Chemnitzer Linuxtage 2016
+
+Datum: 19./20. März 2016
+Ort: TU Chemnitz, Reichenhainer Straße 90, 09126 Chemnitz
```

## Dateien ins Staging aufnehmen

→ `ich@tux:~/testrepo (veranstaltungsinfos)$ git add README.md  
programm.txt`

→ `ich@tux:~/testrepo (veranstaltungsinfos)$ git status`

On branch `veranstaltungsinfos`

**Changes to be committed:**

(use "`git reset HEAD <file>...`" to unstage)

**modified:**      `README.md`

**new file:**      `programm.txt`

## Ins lokale Repo committen

→ `ich@tux:~/testrepo (veranstaltungsinfos)$ git commit -m „mehr infos zu den CLT“`

```
[veranstaltungsinfos 2751fb3] mehr infos zu den CLT
 2 files changed, 3 insertions(+)
 create mode 100644 programm.txt
```

→ `ich@tux:~/testrepo (veranstaltungsinfos)$ git log`  
`commit 2751fb3c6b7b7a0c1739a56666e38b7de04b0168`  
`Author: Silke Meyer <silke@silkemeyer.net>`  
`Date: Tue Mar 15 12:43:52 2016 +0100`

```
mehr infos zu den CLT
```

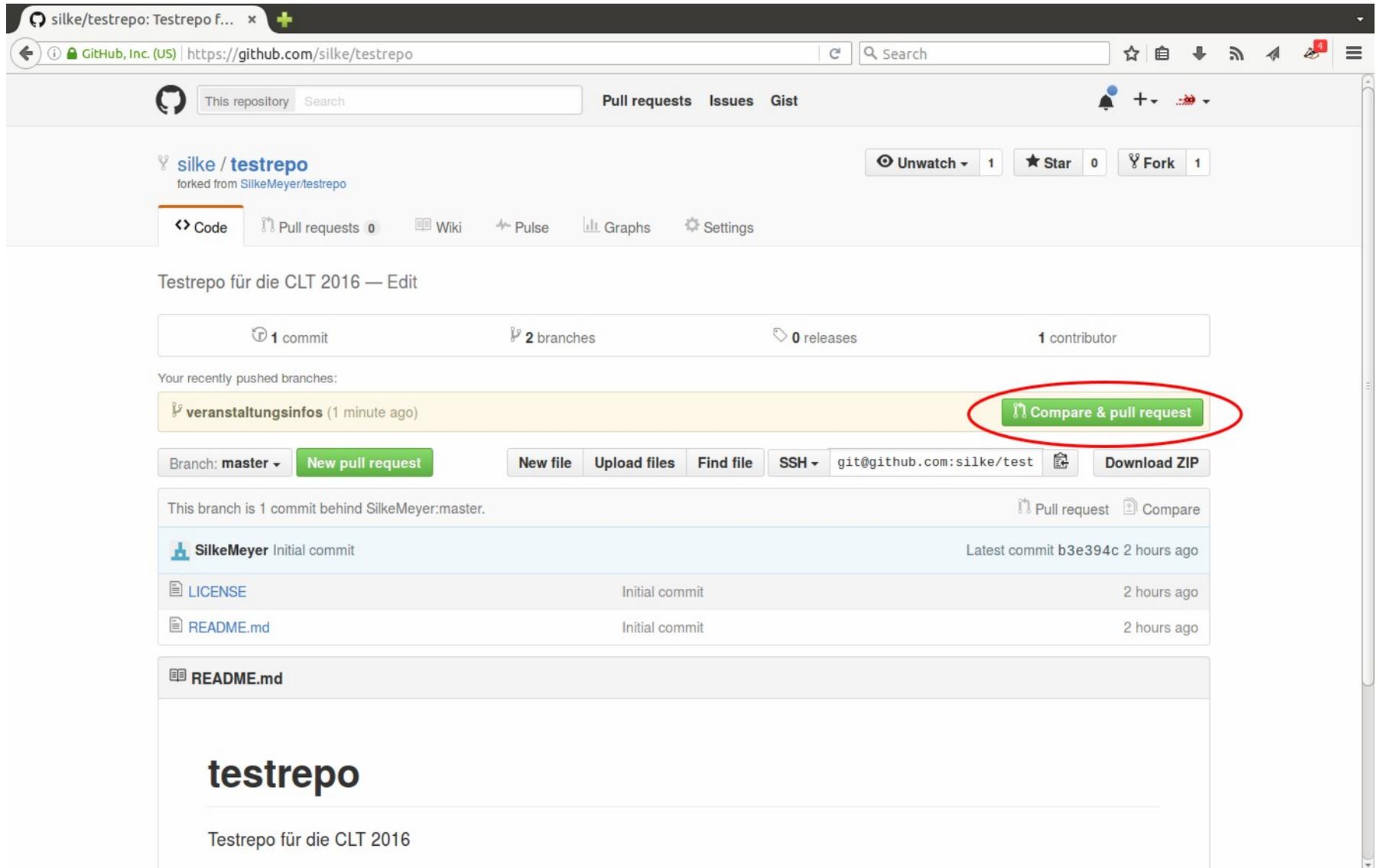
...

## Ins Origin-Repo hochladen

- Der lokale Branch soll ins eigene entfernte Repository.
- `ich@tux:~/testrepo (veranstaltungsinfos)$ git push origin veranstaltungsinfos`  
Counting objects: 9, done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (7/7), 715 bytes | 0 bytes/s, done.  
Total 7 (delta 1), reused 0 (delta 0)  
To git@github.com:silke/testrepo.git  
\* [new branch]      veranstaltungsinfos ->  
veranstaltungsinfos

## Der Pull Request

- Nicht möglich auf der Kommandozeile
- Ab ins Webinterface!



The screenshot shows a GitHub repository page for 'silke/testrepo'. The repository is forked from 'SilkeMeyer/testrepo'. The page displays the repository name, a search bar, and navigation links for Pull requests, Issues, and Gist. The repository statistics show 1 commit, 2 branches, 0 releases, and 1 contributor. A recently pushed branch named 'veranstaltungsinfos' (1 minute ago) is highlighted, with a green 'Compare & pull request' button circled in red. The page also shows the current branch 'master' and a 'New pull request' button. The commit history shows three commits: 'Initial commit' by SilkeMeyer, 'LICENSE', and 'README.md', all committed 2 hours ago. The README.md file is visible at the bottom of the page.

silke/testrepo: Testrepo f... x +

GitHub, Inc. (US) | https://github.com/silke/testrepo

This repository Search

Pull requests Issues Gist

silke / testrepo  
forked from SilkeMeyer/testrepo

Unwatch 1 Star 0 Fork 1

Code Pull requests 0 Wiki Pulse Graphs Settings

Testrepo für die CLT 2016 — Edit

1 commit 2 branches 0 releases 1 contributor

Your recently pushed branches:

veranstaltungsinfos (1 minute ago) **Compare & pull request**

Branch: master New pull request New file Upload files Find file SSH git@github.com:silke/test Download ZIP

This branch is 1 commit behind SilkeMeyer:master. Pull request Compare

SilkeMeyer Initial commit Latest commit b3e394c 2 hours ago

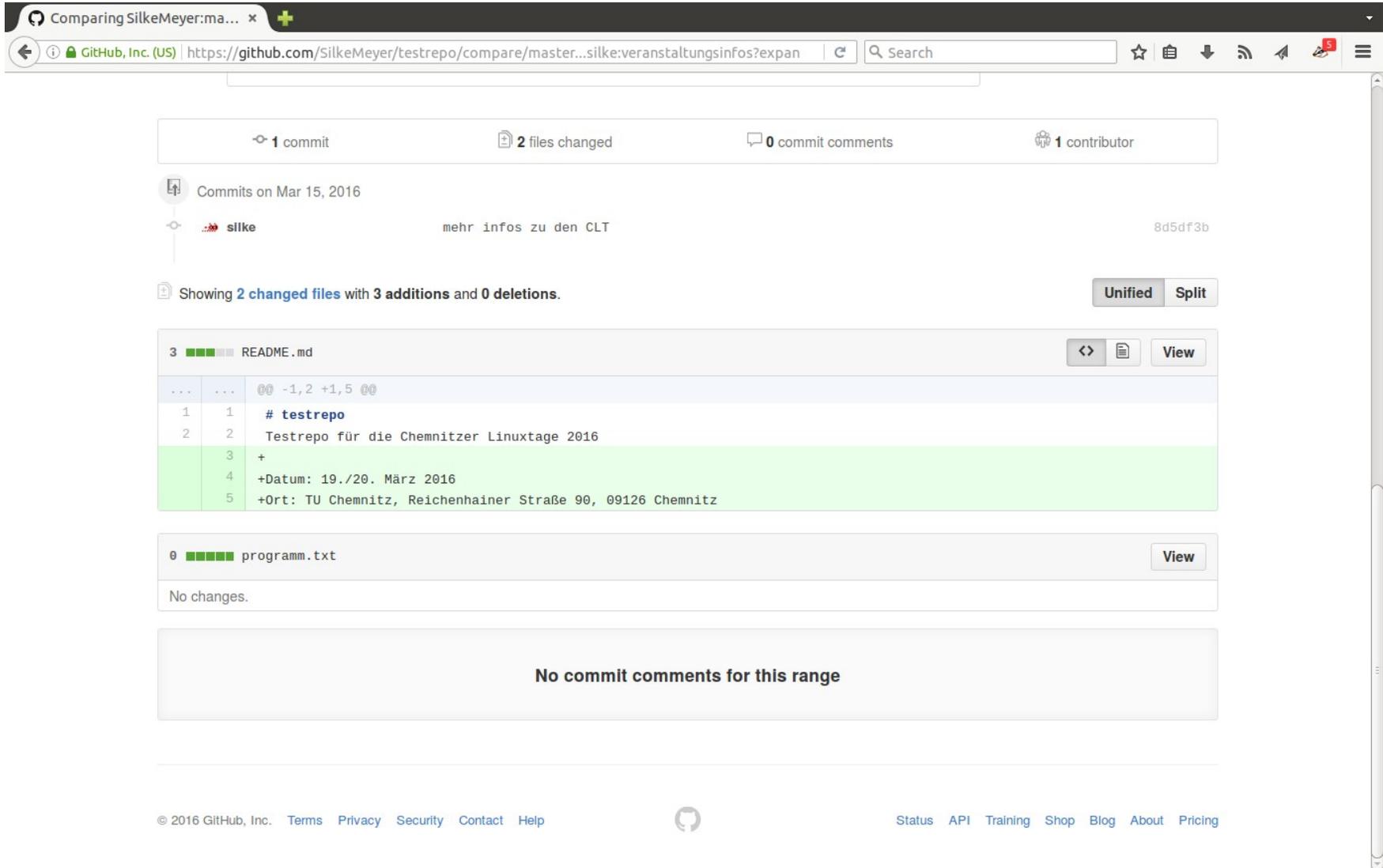
LICENSE Initial commit 2 hours ago

README.md Initial commit 2 hours ago

README.md

# testrepo

Testrepo für die CLT 2016



Comparing SilkeMeyer:ma... x

GitHub, Inc. (US) | <https://github.com/SilkeMeyer/testrepo/compare/master...silke:veranstaltungsinfos?expand=1> Search

1 commit   2 files changed   0 commit comments   1 contributor

Commits on Mar 15, 2016

silke mehr infos zu den CLT 8d5df3b

Showing 2 changed files with 3 additions and 0 deletions. Unified Split

3 ■■■■■ README.md <> 📄 View

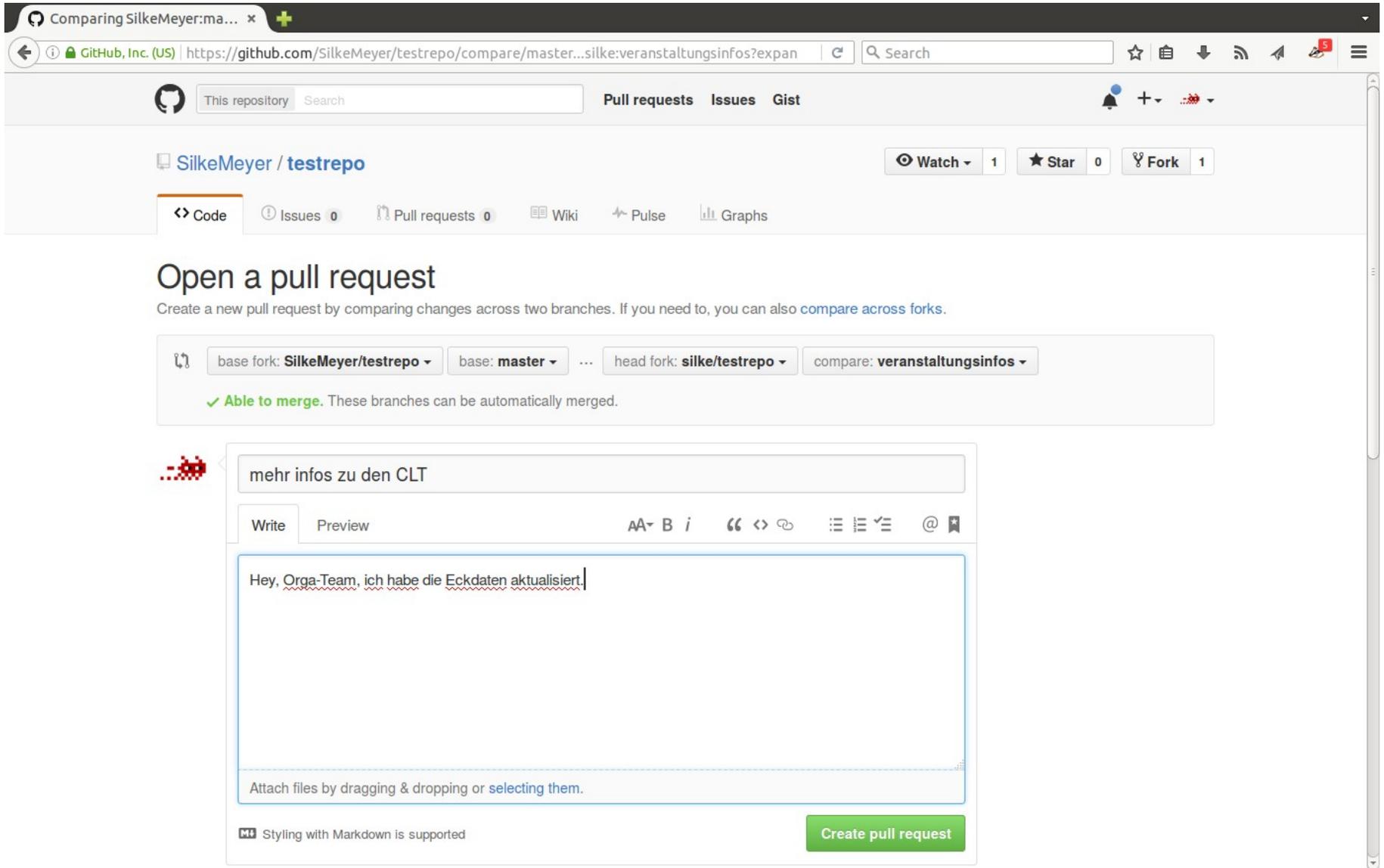
...	...	00 -1,2 +1,5 00
1	1	# testrepo
2	2	Testrepo für die Chemnitzer Linxstage 2016
	3	+
	4	+Datum: 19./20. März 2016
	5	+Ort: TU Chemnitz, Reichenhainer Straße 90, 09126 Chemnitz

0 ■■■■■ programm.txt View

No changes.

No commit comments for this range

© 2016 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#) [Help](#)  [Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#) [Pricing](#)



Comparing SilkeMeyer:ma... x

GitHub, Inc. (US) | <https://github.com/SilkeMeyer/testrepo/compare/master...silke:veranstaltungsinfos?expand=1>

This repository Search Pull requests Issues Gist

SilkeMeyer / testrepo Watch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Wiki Pulse Graphs

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: SilkeMeyer/testrepo base: master ... head fork: silke/testrepo compare: veranstaltungsinfos

✓ Able to merge. These branches can be automatically merged.

mehr infos zu den CLT

Write Preview AA- B i “ <> @

Hey, Orga-Team, ich habe die Eckdaten aktualisiert.

Attach files by dragging & dropping or selecting them.

Styling with Markdown is supported

Create pull request

## Merge-Konflikte

- Beispiel: README.md wurden in beiden lokalen Branches (master und veranstaltungsinfos) bearbeitet.
- `ich@tux:~/testrepo (master)$ git merge veranstaltungsinfos`  
Auto-merging README.md  
CONFLICT (content): Merge conflict in README.md  
Automatic merge failed; fix conflicts and then commit the result.
- Die konfligierende(n) Datei(en) werden markiert.

## Merge-Konflikte

```
→ ich@tux:~/testrepo (master)$ cat README.md
# testrepo
<<<<<<< HEAD
Testrepo für die Chemnitzer Linxstage 2016
Wann? 19./20.3.2016
Wo? Campus Reichenhainer Str.
Hashtag: #CLT2016
=====
Veranstaltungsankündigung für die Chemnitzer Linxstage
2016
Datum: 19./20. März 2016
Ort: TU Chemnitz, Reichenhainer Straße 90, 09126
Chemnitz
>>>>>> veranstaltungsinfos
```

## Merge-Konflikt lösen

→ Abbrechen...?

→ `ich@tux:~/testrepo (master)$ git merge --abort`

→ Manuell nachbearbeiten

→ `ich@tux:~/testrepo (master)$ git mergetool`

Merging:

README.md

Normal merge conflict for 'README.md':

{local}: modified file

{remote}: modified file

Hit return to start merge resolution tool (meld):

→ Editor lädt drei Versionen der Datei

→ Alte Version (vor dem Konflikt)

→ Neue Version im aktuellen Branch („local“)

→ Neue Version im reingeholten Branch („remote“)

## Waaah! Kopf ab?

- Einen Commit auschecken: auf *keinem* Branch sein.
- `ich@tux:~/testrepo (master)$ git checkout 14d767e35c85b2eb14d3c0eb077309dd3c413848`  
Note: checking out  
'14d767e35c85b2eb14d3c0eb077309dd3c413848'.  
**You are in 'detached HEAD' state.** You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.
- Zurück auf einen Branch gehen mit `git checkout <branchname>`

## Lokales Repo aus Upstream aktualisieren

- `ich@tux:~/testrepo (master) $ git remote -v`  
`origin git@github.com:silke/testrepo.git (fetch)`  
`origin git@github.com:silke/testrepo.git (push)`
- `ich@tux:~/testrepo (master) $ git remote add upstream`  
`https://github.com/SilkeMeyer/testrepo.git`
- `ich@tux:~/testrepo (master) $ git remote -v`  
`origin git@github.com:silke/testrepo.git (fetch)`  
`origin git@github.com:silke/testrepo.git (push)`  
`upstream https://github.com/SilkeMeyer/testrepo.git`  
`(fetch)`  
`upstream https://github.com/SilkeMeyer/testrepo.git`  
`(push)`

## Lokales Repo aus Upstream aktualisieren

- `ich@tux:~/testrepo (master)$ git fetch upstream`  
remote: Counting objects: 3, done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
From <https://github.com/SilkeMeyer/testrepo>  
\* [new branch] master -> upstream/master
- `ich@tux:~/testrepo (master) $ git merge upstream/master`  
Updating b3e394c..55bc83d  
Fast-forward  
README.md | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)

## Tags

- Markierte Commits, z.B. für Release oder Release Candidate
- Tag erstellen:  

```
ich@tux:~/testrepo $ git tag -a v1.0 -m "Release v1.0" 8d5df3bd0998508403c1eaac240e26fa5462b24a
```
- Auf getaggten Commit wechseln:  

```
ich@tux:~/testrepo $ git checkout v1.0
```
- Lokal erstellte Tags hochladen:  

```
ich@tux:~/testrepo $ git push --tags
```

## Lektüre

- [Offizielle Doku](#)
- Valentin Haenel / Julius Plenz: [Das Git-Buch](#), Open Source Press, 2014 (CC BY-NC-SA 4.0)
- Scott Chacon / Ben Straub: [Pro Git](#), Apress, 2009 (CC BY-NC-SA 3.0)
- Seth Robertson: [On undoing, fixing, or removing commits in git](#)
- Vincent Driessen: [A successful Git branching model](#)
- Christie Koehler: [Fun with git submodules](#)
- [Offizielle Github Hilfe](#)
- Sucht Euch Online-Tutorials!

- Danke für das Interesse!
- Natürlich und gerne stehe ich Ihnen jederzeit mit Rat und Tat zur Verfügung und freue mich auf neue Kontakte.
  - Silke Meyer
  - Mail: [s.meyer@heinlein-support.de](mailto:s.meyer@heinlein-support.de)
  - Telefon: 030/40 50 51 - 51
- Wenn's brennt:
  - Heinlein Support 24/7 Notfall-Hotline: 030/40 505 - 110

## Wir suchen:

Admins, Consultants, Trainer!

## Wir bieten:

Spannende Projekte, Kundenlob, eigenständige Arbeit, keine Überstunden, Teamarbeit

...und natürlich: Linux, Linux, Linux...

<http://www.helein-support.de/jobs>

# Heinlein Support hilft bei allen Fragen rund um Linux-Server

## HEINLEIN AKADEMIE

Von Profis für Profis: Wir vermitteln die oberen 10% Wissen: geballtes Wissen und umfangreiche Praxiserfahrung.

## HEINLEIN HOSTING

Individuelles Business-Hosting mit perfekter Maintenance durch unsere Profis. Sicherheit und Verfügbarkeit stehen an erster Stelle.

## HEINLEIN CONSULTING

Das Backup für Ihre Linux-Administration: LPIC-2-Profis lösen im CompetenceCall Notfälle, auch in SLAs mit 24/7-Verfügbarkeit.

## HEINLEIN ELEMENTS

Hard- und Software-Appliances und speziell für den Serverbetrieb konzipierte Software rund ums Thema eMail.