SSH-Forwarding, Fabric und GitHub

Django-UserGroup Hamburg 13.11.2013

Arne Brodowski
www.arnebrodowski.de

Lokale Entwicklung

- Identifizierung erfolgt durch SSH-Key
- Verifizieren mittels:

```
(local)$ ssh -T git@github.com
Hi arneb! You've successfully authenticated,
but GitHub does not provide shell access.
Connection to github.com closed.
```

Auf dem Server

- Entweder: GitHub Deploy Keys
  - Nachteil: Key ist an genau ein Repo gebunden
- Oder: SSH-Forwarding
  - Nachteil: Deployment muss per SSH erfolgen

SSH-Forwarding

Aktivieren in der Datei ~/.ssh/config:

```
Host server.example.com
    ForwardAgent yes
```

Ausprobieren:

```
(local)$ ssh server.example.com
(remote)$ ssh -T git@github.com
Hi arneb! You've successfully authenticated,
but GitHub does not provide shell access.
Connection to github.com closed.
```

Fabric

- env.forward_agent = True
  - Ab Version 1.4 möglich
- env.use_ssh_config = True

Fabric

```
from fabric.api import env, run
env.forward_agent = True
env.hosts = ["server.example.com"]

def github():
    run("ssh -T git@github.com", shell=False)
```

Fabric

(local)$ fab github
["server...com"] Executing Task 'github'
["server...com"] run: ssh -T git@github.com
["server...com"] out:
["server...com"] out: Hi arneb! You‘ve successfully ...
(ein paar Fehlermeldungen ...)

Hat dieser Schritt funktioniert, kann man alle anderen Git(Hub) Operationen im fabfile mit der lokalen Entwickler-Identität ausführen.

Fabric

Ein Beispiel (vereinfacht):

```
def git_pull():
    with cd(env.path):
        run("git pull origin %s" % env.branch)
```

Weitere Informationen

- https://help.github.com/articles/using-ssh-agent-forwarding
- https://help.github.com/articles/managing-deploy-keys
- http://docs.fabfile.org