

# SSH Authorized Keys Command

In `.ssh/authorized_keys` kann man nicht nur einfach SSH-Public-Keys hinterlegen, sondern auch direkt Skripte ausführen lassen.

## authorized\_keys

```
command="~/script" ssh-dsa ...
```

Wenn sich mit dem entsprechenden Key angemeldet wird, wird direkt das Skript ausgeführt und die Verbindung nach Beendigung von diesem geschlossen.

Wenn das Skript leer ist, dann sollte das ungefähr so aussehen:

```
$ ssh user@host  
Connection to host closed.
```

## Output

Wir können natürlich Text ausgeben:

## script

```
#!/usr/bin/env python3  
  
print("Hello World")
```

```
$ ssh user@host  
Hello World  
Connection to host closed.
```

## Optionen

Man kann bei einer SSH-Verbindung Kommandos übergeben. Diese sind in der Environment-Variable `SSH_ORIGINAL_COMMAND` hinterlegt.

## script

```
#!/usr/bin/env python3  
  
import os
```

```
print(os.environ.get("SSH_ORIGINAL_COMMAND"))
```

```
$ ssh user@host hello world  
hello world
```

## Daten übertragen

Es ist auch möglich Pipes über SSH weiter zu leiten. Das Skript kann dann ganz normal über stdin darauf zugreifen.

[script](#)

```
#!/usr/bin/env python3  
  
import sys  
  
for line in sys.stdin:  
    print(line)
```

```
$ echo "Hello World!" | ssh user@host  
Hello World!
```

```
Connection to host closed.
```

## Security

Um diesem SSH Keys ausschließlich diesen Command zu erlauben, kann mit `restrict` jedes andere SSH-Feature für den Key deaktiviert werden. Z.B. Port-Forwarding.

[authorized\\_keys](#)

```
command="~/script",restrict ssh-dsa ...
```

From:

<https://wiki.clerie.de/> - **clerie's Wiki**

Permanent link:

<https://wiki.clerie.de/notiz/ssh-authorized-keys-command>

Last update: **2021/05/05 14:22**

