

Troubleshooting: Command-Timeout und SSH-Timeout

Timeout Fehler können insbesondere bei langsamen CPU's auftreten, in meinem Fall war das eine i5-4570. Die Fehler treten anscheinend vollkommen unsystematisch auf: Sie können bei ein und derselben Einstellung mal auftreten und beim nächsten Mal auch wieder nicht. Ein Verursachungszusammenhang ist bislang für mich nicht erkennbar, sodass man es auch einfach nochmal versuchen kann, ohne etwas zu ändern. Es gibt aber auch Möglichkeiten, die manchmal funktionieren bzw. zu funktionieren scheinen (und manchmal aber auch nicht):

(1) Den Start Delay erhöhen (nicht verringern, auch wenn das auf den ersten Blick logischer erscheint!).

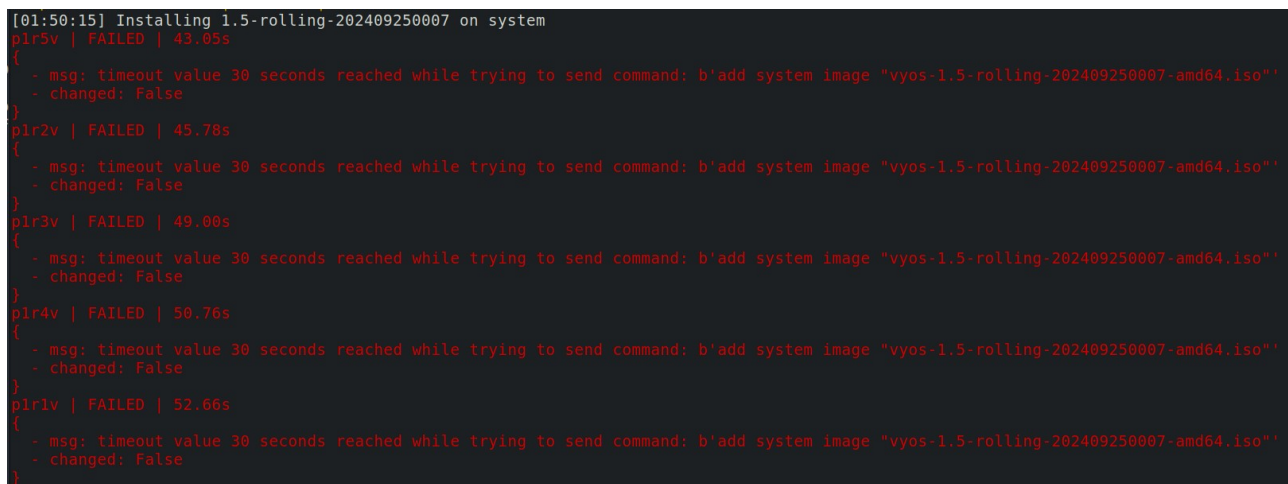
(2) Den anstomlog in der ansible.cfg auskommentieren:

```
[defaults]
inventory = /home/user/streams/ansible/inventories/
host_key_checking = False
deprecation_warnings = False
# callback_plugins= ./callbacks
# stdout_callback = anstomlog
```

(3) Weniger Router in einem einzigen Durchgang erstellen.

(4) In den seriellen Modus wechseln.

(5) Den jeweiligen Default-Timeout Wert erhöhen. Das wäre die logischste Reaktion. Dabei ist zu unterscheiden, zwischen Command-Timeout, d.h. einer Zeitüberschreitung von (per default) 30 Sekunden bei der Ausführung eines Ansible Tasks und SSH-Timeout. Ein **Command-Timeout** sieht folgendermaßen aus:



```
[01:50:15] Installing 1.5-rolling-202409250007 on system
p1r5v | FAILED | 43.05s
{
  - msg: timeout value 30 seconds reached while trying to send command: b'add system image "vyos-1.5-rolling-202409250007-amd64.iso"'
  - changed: False
}
p1r2v | FAILED | 45.78s
{
  - msg: timeout value 30 seconds reached while trying to send command: b'add system image "vyos-1.5-rolling-202409250007-amd64.iso"'
  - changed: False
}
p1r3v | FAILED | 49.00s
{
  - msg: timeout value 30 seconds reached while trying to send command: b'add system image "vyos-1.5-rolling-202409250007-amd64.iso"'
  - changed: False
}
p1r4v | FAILED | 50.76s
{
  - msg: timeout value 30 seconds reached while trying to send command: b'add system image "vyos-1.5-rolling-202409250007-amd64.iso"'
  - changed: False
}
p1r1v | FAILED | 52.66s
{
  - msg: timeout value 30 seconds reached while trying to send command: b'add system image "vyos-1.5-rolling-202409250007-amd64.iso"'
  - changed: False
}
```

Das sollte durch Setzen einer globalen Variable im jeweiligen Playbook zu beheben sein:

```
---
- name: VyOS - install new image and schedule machine reboot.
  gather_facts: False
  connection: ansible.netcommon.network_cli
  vars:
    vyos_dir: "vyos-files/"
    vyos_file: "vyos-{{ vyos_version }}-amd64.iso"
```

```
ansible_user: vyos
ansible_password: vyos
ansible_network_os: vyos.vyos.vyos
ansible_command_timeout: 100
```

```
hosts:
  - router
```

```
tasks:
  - name: "Copying {{vyos_file}} to system"
    net_put:
      src: "{{ vyos_dir }}{{ vyos_file }}"
      dest: "{{ vyos_file }}"
```

Oder wenn ein Task innerhalb einer Rolle betroffen ist:

```
- hosts: router
  gather_facts: "no"
  connection: network_cli
  vars:
    ansible_command_timeout: 100

  roles:
    - role: vyos_setup
      tags: vyos
      when: os == "vyos"
```

In meinen Testläufen habe ich nicht feststellen können, dass diese Herangehensweise funktioniert. Dasselbe als lokale Variable in einzelne Tasks reinzuschreiben hat bei mir ebenfalls nichts gebracht.

Ein **SSH-Timeout** sieht z.B. so aus:

```
p3r1v | FAILED | 10.48s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
[14:18:56] vyos_setup : save config
p3r2v | FAILED | 10.65s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
p3r3v | FAILED | 10.71s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
p3r6v | FAILED | 10.72s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
p3r4v | FAILED | 10.73s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
p3r5v | FAILED | 10.76s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
p3r9v | CHANGED | 12.70s
p3r7v | FAILED | 21.18s
{
  - msg: Failed to authenticate: Authentication timeout.
  - changed: False
}
p3r8v | FAILED | 21.24s
{
  - msg: Failed to authenticate: Authentication timeout.
```

In so einem Fall sollte die Lösung darin bestehen, eine entsprechend höhere Umgebungsvariable im Skript festzulegen:

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

```
export ANSIBLE_HOST_KEY_CHECKING=False
```

```
export ANSIBLE_SSH_TIMEOUT=100
```

```
node=$1
```

```
provider=$2
```

```
[...]
```

Das wäre zwar logisch, funktioniert aber offenbar nicht, denn das steht bereits so im Skript.

Insgesamt habe ich keinen systematischen Verursachungszusammenhang, noch systematisch funktionierende Lösungen für solche Timeout-Probleme finden können. Gefühlt funktioniert am ehesten eine Erhöhung des Start Delay Wertes. Deshalb Fazit: Einfach mehrmals probieren, evtl. mit einer der genannten Variationsmöglichkeiten, meistens funktioniert es dann relativ schnell wieder.