rbatim

erbatim

# Etwas weniger Kaffee bitte

Andreas Fritzsche

2024

# Inhaltsverzeichnis


1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

# Inhaltsverzeichnis

1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

# Danksagung

Danke an meine Freundin und Familie, die mir bei dem Vortrag sehr geholfen und mich unterstützt haben.

Die mir viele Aufgaben abgenommen haben.
Ohne sie gäbe es heute keinen Vortrag.

# Projektseite

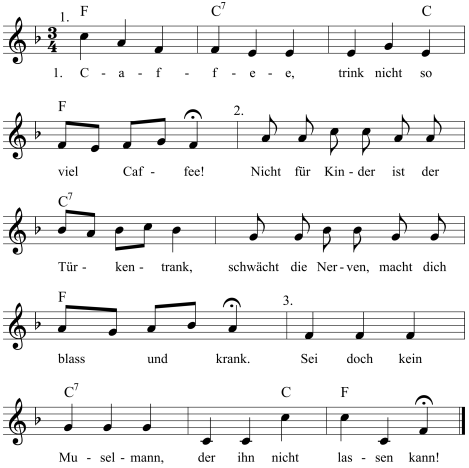
https://gitlab.com/1fridy/etwas-weniger-kaffee-bitte

# Inhaltsverzeichnis

1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

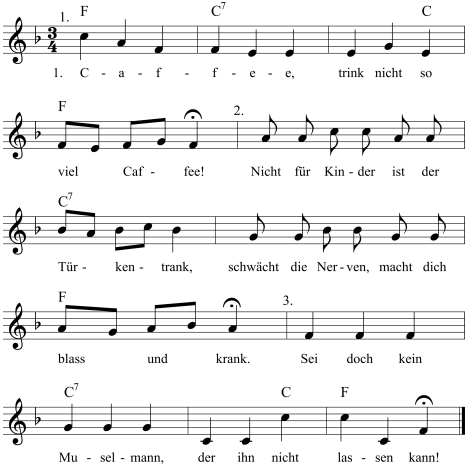
# Was ist Kaffee?

- psychotropes, coffeinhaltiges Getränk



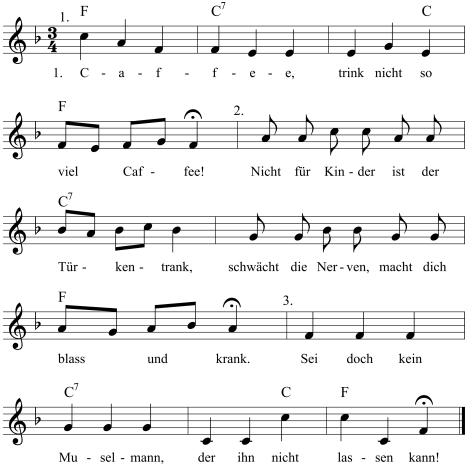
# Was ist Kaffee?

- psychotropes, coffeinhaltiges Getränk
- geröstete, gemahlene Kaffebohnen+Wasser



# Was ist Kaffee?

- psychotropes, coffeinhaltiges Getränk
- geröstete, gemahlene Kaffebohnen+Wasser
- köstlich



# Koffein

- Erhöhung der Kontraktionskraft des Herzens/Steigerung der Herzfrequenz

Der erste Liter Kaffee ist der wichtigste

# Koffein

- Erhöhung der Kontraktionskraft des Herzens/Steigerung der Herzfrequenz
- Bronchialerweiterung

Der erste Liter Kaffee ist der wichtigste

# Koffein

- Erhöhung der Kontraktionskraft des Herzens/Steigerung der Herzfrequenz
- Bronchialerweiterung
- Schwach harntreibende, Anregung der Peristaltik des Darmes

Der erste Liter Kaffee ist der wichtigste

# Koffein

- Erhöhung der Kontraktionskraft des Herzens/Steigerung der Herzfrequenz
- Bronchialerweiterung
- Schwach harntreibende, Anregung der Peristaltik des Darmes
- Leistungsteigernd (war mal auf Dopingliste)

Der erste Liter Kaffee ist der wichtigste

# Koffein

- Anregung des Zentralnervensystems

Der erste Liter Kaffee ist der wichtigste

# Koffein

- Anregung des Zentralnervensystems
- steigert Konzentrationsfähigkeit, Vigilanz u. Aufmerksamkeit wird verbesserd

Der erste Liter Kaffee ist der wichtigste

# Koffein

- Anregung des Zentralnervensystems
- steigert Konzentrationsfähigkeit, Vigilanz u. Aufmerksamkeit wird verbesserd
- erhöht Geschwindigkeit von Denkprozessen, Langzeitgedächtnis verbessern

Der erste Liter Kaffee ist der wichtigste

# Koffein

Resultat:

Der erste Liter Kaffee ist der wichtigste

# Koffein

Resultat:

- bekommt besser mit wenn etwas länger dauert

Der erste Liter Kaffee ist der wichtigste

# Koffein

Resultat:

- bekommt besser mit wenn etwas länger dauert
- man merkt es sich besser

Der erste Liter Kaffee ist der wichtigste

# Koffein

Resultat:

- bekommt besser mit wenn etwas länger dauert
- man merkt es sich besser
- und länger

Der erste Liter Kaffee ist der wichtigste

# Koffein

Kein

- Schwarzer/Grüner Tee
- Cola
- Mate
- Dunkle Schokolade

stattdessen

# Koffein

Kein

- Schwarzer/Grüner Tee
- Cola
- Mate
- Dunkle Schokolade

stattdessen

- Wasser
- Säfte

# Koffein

Kein

- Schwarzer/Grüner Tee
- Cola
- Mate
- Dunkle Schokolade

stattdessen

- Wasser
- Säfte
- Bier

# Koffein

Kein

- Schwarzer/Grüner Tee
- Cola
- Mate
- Dunkle Schokolade

stattdessen

- Wasser
- Säfte
- Bier
- Wein

# Koffein

Kein

- Schwarzer/Grüner Tee
- Cola
- Mate
- Dunkle Schokolade

stattdessen

- Wasser
- Säfte
- Bier
- Wein
- Wiskey Vodka

# Inhaltsverzeichnis

1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

# Ansibe

Open-Source-Automatisierungswerkzeug für:

- Orchestrierung

# Ansibe

Open-Source-Automatisierungswerkzeug für:

- Orchestrierung
- allgemeine Konfiguration

# Ansibe

Open-Source-Automatisierungswerkzeug für:

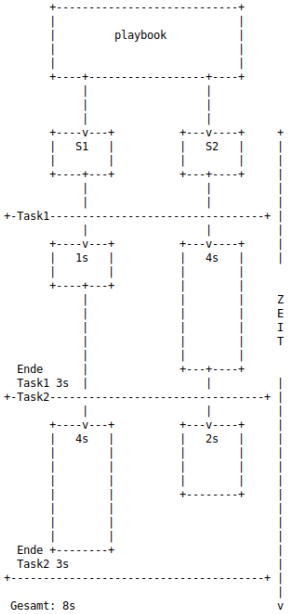
- Orchestrierung
- allgemeine Konfiguration
- Administration von Computern

# Ansibe

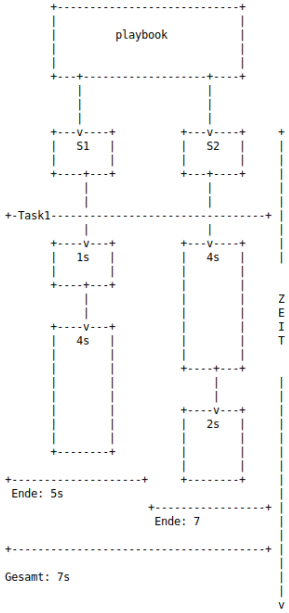
Open-Source-Automatisierungswerkzeug für:

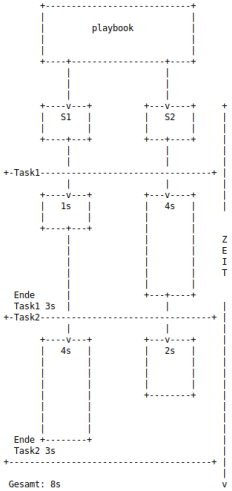
- Orchestrierung
- allgemeine Konfiguration
- Administration von Computern
- Vortrag bei CLT

- strategy: linear

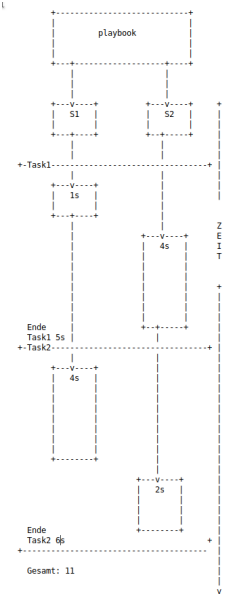


- strategy: linear
- strategy: free

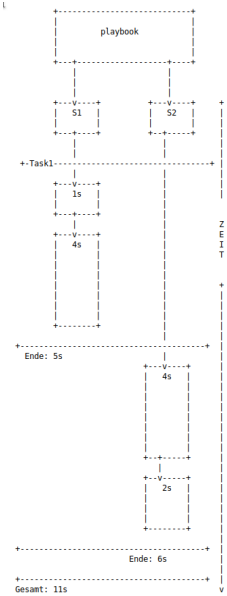


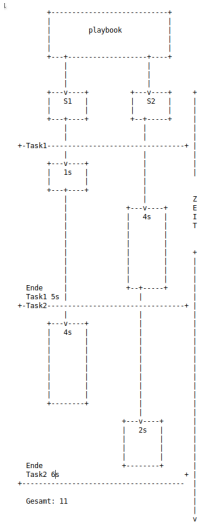



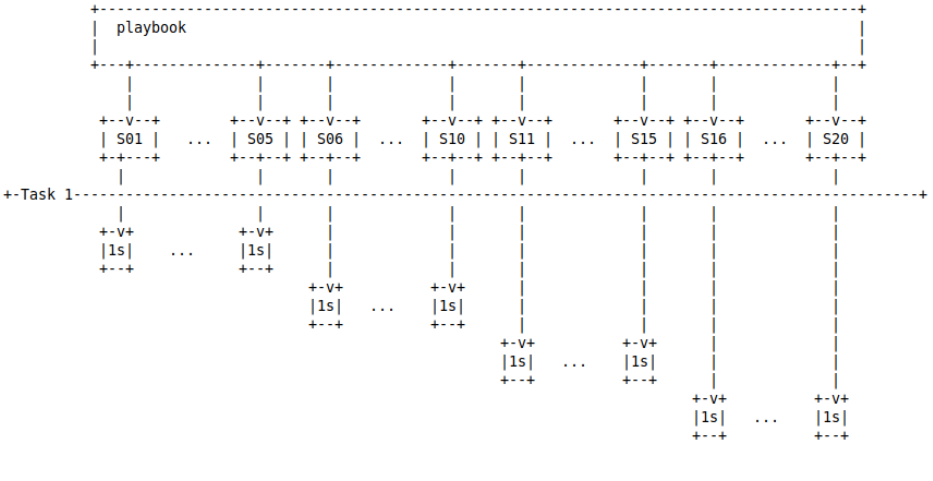
- fork: 1

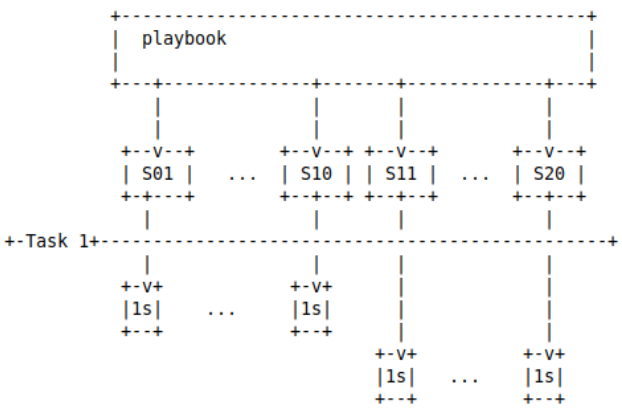

playbook
S1
S2
Task1
1s
4s
Ende
Task1 5s
Task2
4s
2s
Ende
Task2 6s
Gesamt: 11
ZEIT

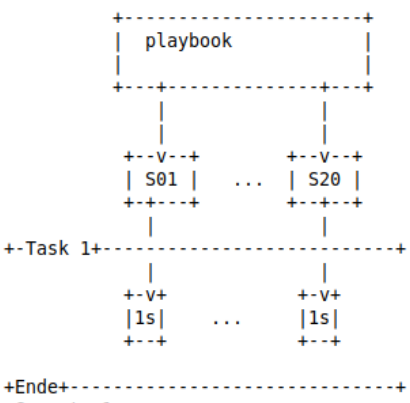
- fork: 1
- serial: 1




playbook
S01
S05
S06
S10
S11
S15
S16
S20
+-Task 1
1s
+Ende
Gesamt: 4


playbook
S01
S10
S11
S20
Task 1
1s
Ende
Gesamt: 2

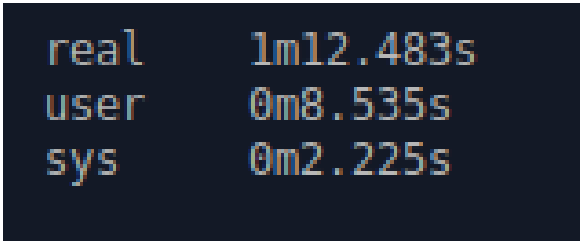

playbook
S01
...
S20
+Task 1+
1s
...
1s
+Ende+
Gesamt: 1

# Inhaltsverzeichnis

1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

# Messen aber wie

- time
  - time ansible-playbook play.yml



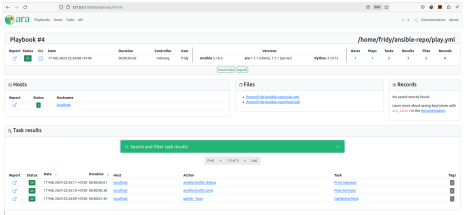
# Messen aber wie

- time
  - time ansible-playbook play.yml
- ansible
  - ansible.cfg callbacks_enabled = timer, profile_taks, profile_roles

```
===============================================================================
websv ------------------------------------------------------------ 251.00s
common ----------------------------------------------------------- 84.80s
setup ------------------------------------------------------------ 1.87s
include_role ----------------------------------------------------- 1.80s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------------------------------------------------- 339.47s
===============================================================================
Gathering Facts -------------------------------------------------------- 1.42s
websv : richte cronjob ein --------------------------------------------- 1.31s
common : bilde leere liste mit gruppen --------------------------------- 0.94s
common : zeige liste an ------------------------------------------------ 0.74s
websv : entpacke skript ------------------------------------------------ 0.68s
common : remove user_home directory ------------------------------------ 0.63s
common : change user user group ---------------------------------------- 0.63s
common : creating groups ----------------------------------------------- 0.61s
common : copy change/date to /etc/motd --------------------------------- 0.56s
common : print vars for uninstall -------------------------------------- 0.55s
common : make user_home -> _user_home ---------------------------------- 0.49s
websv : cleanJboss_loop delete rename change app websv directory ------- 0.49s
websv : do stat of WEBSV_LOG_HOME's old -------------------------------- 0.44s
```

# Messen aber wie

- time
  - time ansible-playbook play.yml
- ansible
  - ansible.cfg callbacks_enabled = timer, profile_taks, profile_roles
- ara
  - ansible.cfg CALLBACK_PLUGINS = /usr/lib/python3.10/site-packages/ara/plugins/callbacks
  - export ANSIBLE_CALLBACK_PLUGINS="$(python3 -m ara.setup.callback_plugin)"

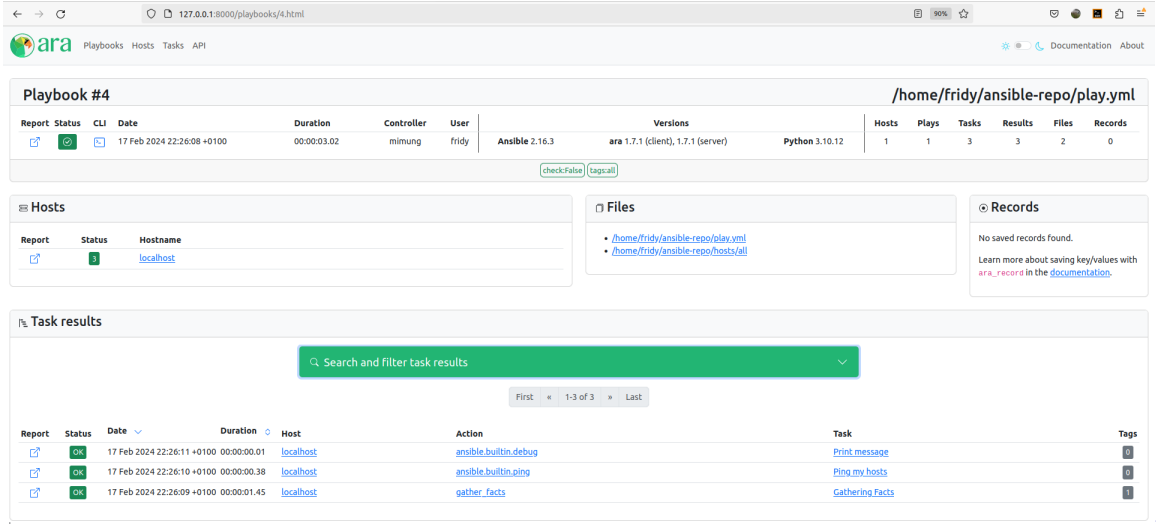


# Wer viel misst, misst ...

```
PLAY RECAP *********************************************************************
localhost                  : ok=3    changed=0    unreachable=0    failed=0    skipp

Playbook run took 0 days, 0 hours, 0 minutes, 3 seconds
Samstag 17 Februar 2024  22:26:11 +0100 (0:00:00.399)       0:00:03.604 *******
===============================================================================
Gathering Facts ---------------------------------------------------------------
Ping my hosts -----------------------------------------------------------------
Print message -----------------------------------------------------------------
Samstag 17 Februar 2024  22:26:11 +0100 (0:00:00.399)       0:00:03.601 *******
===============================================================================
gather_facts ------------------------------------------------------------ 1.72s
ansible.builtin.ping ---------------------------------------------------- 0.55s
ansible.builtin.debug --------------------------------------------------- 0.40s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ------------------------------------------------------------------- 2.67s

real    0m4,392s
user    0m2,189s
sys     0m0,267s
```

| Report | Status | Results | Date | Duration | Action | Task name |
|---|---|---|---|---|---|---|
| | ✓ | 1 | 17 Feb 2024 22:21:31 +0100 | 00:00:00.22 | ansible.builtin.debug | Print message |
| | ✓ | 1 | 17 Feb 2024 22:21:31 +0100 | 00:00:00.45 | ansible.builtin.ping | Ping my hosts |
| | ✓ | 1 | 17 Feb 2024 22:21:29 +0100 | 00:00:01.36 | gather_facts | Gathering Facts |

# ARA ARA Records Ansible



# ARA ARA Records Ansible

**Playbook #4: CLI Arguments**

**Date**
17 Feb 2024 22:26:08 +0100

**Path**
/home/fridy/ansible-repo/play.yml

| Argument | Value |
|---|---|
| version | None |
| verbosity | 0 |
| private_key_file | None |
| remote_user | None |
| connection | ssh |
| timeout | None |
| ssh_common_args | None |

# ARA ARA Records Ansible

## Playbook #4

| Report | Status | CLI | Date | Duration | Controller | User | |
|---|---|---|---|---|---|---|---|
| | | | 17 Feb 2024 22:26:08 +0100 | 00:00:03.02 | mimung | fridy | A |

## Task result details

| Status | Duration | Play | Task | Host |
|---|---|---|---|---|
| OK | 00:00:00.01 | My first play | Print message | localhost |

**Date**
17 Feb 2024 22:26:11 +0100

**Module / Action**
ansible.builtin.debug (/home/fridy/ansible-repo/play.yml:7)

| Field | Value |
|---|---|
| changed | False |
| msg | Hello world |

# Inhaltsverzeichnis

1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

# Optimum

In der Mathematik ist das Optimum der Extrempunkt einer Funktion
und somit Gegenstand der mathematischen Optimierung.
Der Extrempunkt kann ein Maximum oder Minimum sein.

[https://de.wikipedia.org/wiki/Optimum]

# Gatherfacts

- disable in playbook

```
gather_facts: false
```

# Gatherfacts

- disable in playbook
  `gather_facts: false`
- nutze Cache
  `ansible.cfg:`

```
fact_caching_timeout =
120
fact_caching_connection
= localhost:6379:0
gathering=smart
```

# Gatherfacts

- disable in playbook
  `gather_facts: false`
- nutze Cache
  `ansible.cfg:`
  `fact_caching_timeout = 120`
  `fact_caching_connection = localhost:6379:0`
  `gathering=smart`

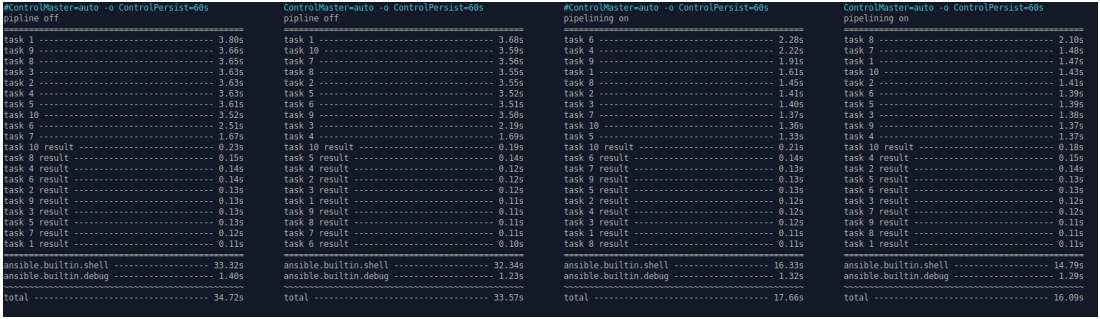
```
========================        ========================
Gathering Facts -- 7.66s        debug ------------ 1.88s
debug ------------ 1.70s

real    0m20.125s               real    1m12.483s
user    0m15.076s               user    0m8.535s
sys     0m5.225s                sys     0m2.225s
```

Apropo Cache: Wenn Inventory langsam, kann Cache möglicherweise helfen

# Testplaybook

```
- hosts: all
  gather_facts: false
  tasks:
    - name: task 1
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 1 result
      ansible.builtin.debug:
        msg: "Task 1 Script result: {{ result }}"
    - name: task 2
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 2 result
      ansible.builtin.debug:
        msg: "Task 2 Script result: {{ result }}"
    - name: task 3
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 3 result
      ansible.builtin.debug:
        msg: "Task 3 Script result: {{ result }}"
    - name: task 4
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 4 result
      ansible.builtin.debug:
        msg: "Task 4 Script result: {{ result }}"
    - name: task 5
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 5 result
      ansible.builtin.debug:
        msg: "Task 5 Script result: {{ result }}"
    - name: task 6
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 6 result
      ansible.builtin.debug:
        msg: "Task 6 Script result: {{ result }}"
    - name: task 7
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 7 result
      ansible.builtin.debug:
        msg: "Task 8 Script result: {{ result }}"
    - name: task 8
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 8 result
      ansible.builtin.debug:
        msg: "Task 8 Script result: {{ result }}"
    - name: task 9
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 9 result
      ansible.builtin.debug:
        msg: "Task 9 Script result: {{ result }}"
    - name: task 10
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 10 result
```

# Pipelining u. Controlmaster



# Pipelining u. Controlmaster

```
#ControlMaster=auto -o ControlPersist=60s
pipline off
==========================================
task 1 ------------------------------------ 3.80s
task 9 ------------------------------------ 3.66s
task 8 ------------------------------------ 3.65s
task 3 ------------------------------------ 3.63s
task 2 ------------------------------------ 3.63s
task 4 ------------------------------------ 3.63s
task 5 ------------------------------------ 3.61s
task 10 ----------------------------------- 3.52s
task 6 ------------------------------------ 2.51s
task 7 ------------------------------------ 1.67s
task 10 result ---------------------------- 0.23s
task 8 result ----------------------------- 0.15s
task 4 result ----------------------------- 0.14s
task 6 result ----------------------------- 0.14s
task 2 result ----------------------------- 0.13s
task 9 result ----------------------------- 0.13s
task 3 result ----------------------------- 0.13s
task 5 result ----------------------------- 0.13s
task 7 result ----------------------------- 0.12s
task 1 result ----------------------------- 0.11s
==========================================
ansible.builtin.shell --------------------- 33.32s
ansible.builtin.debug --------------------- 1.40s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ------------------------------------- 34.72s
```

```
#ControlMaster=auto -o ControlPersist=60s
pipelining on
==========================================
task 6 ------------------------------------ 2.28s
task 4 ------------------------------------ 2.22s
task 9 ------------------------------------ 1.91s
task 1 ------------------------------------ 1.61s
task 8 ------------------------------------ 1.45s
task 2 ------------------------------------ 1.41s
task 3 ------------------------------------ 1.40s
task 7 ------------------------------------ 1.37s
task 10 ----------------------------------- 1.36s
task 5 ------------------------------------ 1.33s
task 10 result ---------------------------- 0.21s
task 6 result ----------------------------- 0.14s
task 7 result ----------------------------- 0.13s
task 9 result ----------------------------- 0.13s
task 5 result ----------------------------- 0.13s
task 2 result ----------------------------- 0.12s
task 4 result ----------------------------- 0.12s
task 3 result ----------------------------- 0.12s
task 1 result ----------------------------- 0.11s
task 8 result ----------------------------- 0.11s
==========================================
ansible.builtin.shell --------------------- 16.33s
ansible.builtin.debug --------------------- 1.32s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ------------------------------------- 17.66s
```

# Pipelining u. Controlmaster

```
#ControlMaster=auto -o ControlPersist=60s
pipelining on
===============================================
task 6 ---------------------------------- 2.28s
task 4 ---------------------------------- 2.22s
task 9 ---------------------------------- 1.91s
task 1 ---------------------------------- 1.61s
task 8 ---------------------------------- 1.45s
task 2 ---------------------------------- 1.41s
task 3 ---------------------------------- 1.40s
task 7 ---------------------------------- 1.37s
task 10 --------------------------------- 1.36s
task 5 ---------------------------------- 1.33s
task 10 result -------------------------- 0.21s
task 6 result --------------------------- 0.14s
task 7 result --------------------------- 0.13s
task 9 result --------------------------- 0.13s
task 5 result --------------------------- 0.13s
task 2 result --------------------------- 0.12s
task 4 result --------------------------- 0.12s
task 3 result --------------------------- 0.12s
task 1 result --------------------------- 0.11s
task 8 result --------------------------- 0.11s
===============================================
ansible.builtin.shell ------------------- 16.33s
ansible.builtin.debug ------------------- 1.32s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------------------------- 17.66s
```

```
ControlMaster=auto -o ControlPersist=60s
pipelining on
===============================================
task 8 ---------------------------------- 2.10s
task 7 ---------------------------------- 1.48s
task 1 ---------------------------------- 1.47s
task 10 --------------------------------- 1.43s
task 2 ---------------------------------- 1.41s
task 6 ---------------------------------- 1.39s
task 5 ---------------------------------- 1.39s
task 3 ---------------------------------- 1.38s
task 9 ---------------------------------- 1.37s
task 4 ---------------------------------- 1.37s
task 10 result -------------------------- 0.18s
task 4 result --------------------------- 0.15s
task 2 result --------------------------- 0.14s
task 5 result --------------------------- 0.13s
task 6 result --------------------------- 0.13s
task 3 result --------------------------- 0.12s
task 7 result --------------------------- 0.12s
task 9 result --------------------------- 0.11s
task 8 result --------------------------- 0.11s
task 1 result --------------------------- 0.11s
===============================================
ansible.builtin.shell ------------------- 14.79s
ansible.builtin.debug ------------------- 1.29s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------------------------- 16.09s
```

# Pipelining u. Controlmaster

Pipelining bringt einen deutlichen Geschwindigkeitszuwachs. Controlmaster im Beispiel nicht so sehr

# Timeout

```
timeout 1 (sleep 2)                    timeout 60 (sleep 2)
===================================    ===================================
task 8 ------------------- 4.34s       task 6 ------------------- 2.95s
task 7 ------------------- 4.31s       task 1 ------------------- 2.41s
task 9 ------------------- 3.28s       task 7 ------------------- 2.37s
task 5 ------------------- 2.82s       task 3 ------------------- 2.37s
task 1 ------------------- 2.42s       task 2 ------------------- 2.37s
task 3 ------------------- 2.38s       task 4 ------------------- 2.36s
task 10 ------------------ 2.37s       task 10 ------------------ 2.36s
task 4 ------------------- 2.36s       task 9 ------------------- 2.36s
task 6 ------------------- 2.35s       task 5 ------------------- 2.35s
task 2 ------------------- 2.34s       task 8 ------------------- 2.34s
task 10 result ----------- 0.20s       task 10 result ----------- 0.20s
task 9 result ------------ 0.14s       task 1 result ------------ 0.15s
task 8 result ------------ 0.13s       task 8 result ------------ 0.13s
task 2 result ------------ 0.13s       task 5 result ------------ 0.13s
task 6 result ------------ 0.13s       task 4 result ------------ 0.12s
task 1 result ------------ 0.12s       task 9 result ------------ 0.12s
task 7 result ------------ 0.12s       task 2 result ------------ 0.12s
task 4 result ------------ 0.12s       task 7 result ------------ 0.12s
task 3 result ------------ 0.11s       task 6 result ------------ 0.11s
task 5 result ------------ 0.10s       task 3 result ------------ 0.11s
===================================    ===================================
ansible.builtin.shell --- 28.98s       ansible.builtin.shell --- 24.26s
ansible.builtin.debug ---- 1.30s       ansible.builtin.debug ---- 1.32s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 30.28s       total -------------------- 25.57s

real    0m30,960s                      real    0m26,201s
user    0m6,955s                       user    0m6,712s
sys     0m3,345s                       sys     0m3,283s
```

# asynchron

```
- hosts: all
  gather_facts: false
  tasks:
    - name: task 1 - background task
      ansible.builtin.command: /bin/sleep 5
      async: 10
      poll: 0
      register: async_task_result

    - name: task 2
      ansible.builtin.debug:
        msg: task 2

    - name: task 3
      ansible.builtin.debug:
        msg: task 3

    - name: Task 4 - wait for async task
      ansible.builtin.async_status:
        jid: "{{ async_task_result.ansible_job_id }}"
      until: job_result.finished
      register: job_result
      retries: 10
      delay: 2

    - name: task 5
      ansible.builtin.debug:
        msg: task 4
```

```
=====================================
Task 4 - wait for async task ------ 7.16s
task 1 - background task ---------- 0.82s
task 5 -------------------------- 0.18s
task 2 -------------------------- 0.11s
task 3 -------------------------- 0.10s
=====================================
ansible.builtin.async_status ------ 7.16s
ansible.builtin.command ----------- 0.82s
ansible.builtin.debug ------------- 0.39s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------------------- 8.37s

real    0m8,956s
user    0m2,577s
sys     0m1,071s
```

# asynchron fire and forget

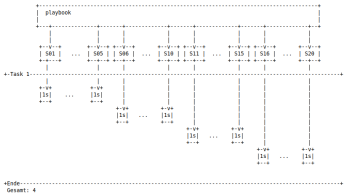
```
pol 2                                pool 0
================================     ================================
task 3 ------------------- 17.42s     task 1 ------------------- 2.91s
task 9 ------------------- 16.94s     task 3 ------------------- 2.70s
task 2 ------------------- 16.93s     task 7 ------------------- 2.67s
task 7 ------------------- 16.62s     task 9 ------------------- 2.63s
task 8 ------------------- 16.50s     task 8 ------------------- 0.90s
task 1 ------------------- 16.40s     task 6 ------------------- 0.81s
task 4 ------------------- 16.31s     task 4 ------------------- 0.76s
task 10 ------------------ 14.87s     task 5 ------------------- 0.74s
task 6 ------------------- 14.86s     task 2 ------------------- 0.71s
task 5 ------------------- 14.84s     task 10 ------------------ 0.71s
task 10 result ------------ 0.20s     task 10 result ------------ 0.19s
task 2 result ------------- 0.13s     task 4 result ------------- 0.14s
task 1 result ------------- 0.13s     task 5 result ------------- 0.12s
task 8 result ------------- 0.12s     task 7 result ------------- 0.12s
task 4 result ------------- 0.12s     task 1 result ------------- 0.12s
task 7 result ------------- 0.12s     task 3 result ------------- 0.11s
task 9 result ------------- 0.12s     task 9 result ------------- 0.11s
task 6 result ------------- 0.12s     task 2 result ------------- 0.11s
task 3 result ------------- 0.11s     task 6 result ------------- 0.11s
task 5 result ------------- 0.11s     task 8 result ------------- 0.11s
================================     ================================
ansible.builtin.shell --- 161.68s    ansible.builtin.shell ---- 15.55s
ansible.builtin.debug ----- 1.28s    ansible.builtin.debug ----- 1.23s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 162.96s    total -------------------- 16.78s
```

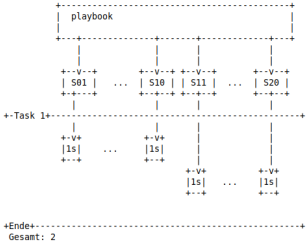
# asynchron fire and forget

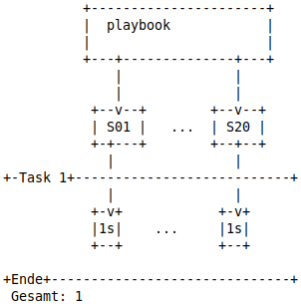
```
"msg": "Task 10 Script result:
{'started': 1, 'finished': 1, 'stdout': '', 'stderr': '', 'stdout_lines': [], 'stderr_lines': [],
 'ansible_job_id': '656541655692.3401700', 'results_file': '*/656541655692.3401700',
 'changed': True, 'rc': 0, 'cmd': '/bin/sleep 12',
'start': '2024-03-14 10:39:55.574215',
'end': '2024-03-14 10:40:07.579631',
'delta': '0:00:12.005416', 'msg': '', 'failed': False}"

"msg": "Task 10 Script result:
{'failed': 0, 'started': 1, 'finished': 0, 'ansible_job_id': '625777812121.3237193',
'results_file': '*/625777812121.3237193', 'changed': True}"
```

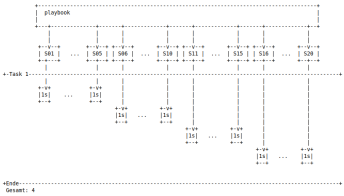
# asynchron fire and forget

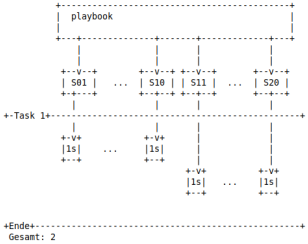
Wenn wir Ergebnisse nicht brauchen:
Tasks mit asynchron und poll 0 ausführen
Ideal: alle Task unabhängig und mit asynchron beschleunigen

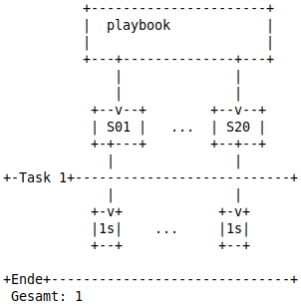

playbook
S01
S05
S06
S10
S11
S15
S16
S20
Task 1
1s
Ende
Gesamt: 4


playbook
S01
S10
S11
S20
Task 1
1s
Ende
Gesamt: 2

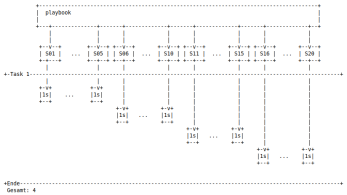

playbook
S01
S10
S20
Task 1
1s
Ende
Gesamt: 1

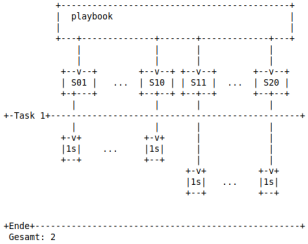


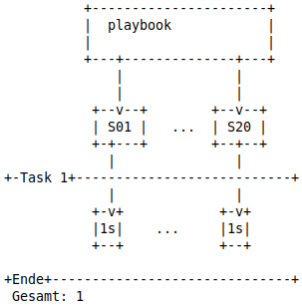




```
forks 20
================================
dnf ------------------- 21.05s
ansible.builtin.file --  3.50s
shell ----------------   2.42s
debug ----------------   1.86s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------- 28.83s
```




```
forks 20
================================
dnf ------------------- 21.05s
ansible.builtin.file --  3.50s
shell ----------------   2.42s
debug ----------------   1.86s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------- 28.83s
```

```
forks 60
================================
dnf ------------------ 10.34s
ansible.builtin.file -  6.36s
debug ----------------  6.16s
shell ----------------  2.50s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------- 25.36s
```

# Forks

- zuwachs nicht linear

# Forks

- zuwachs nicht linear
- Monitor bei neun Werten

# Forks

- zuwachs nicht linear
- Monitor bei neun Werten
- CPU begrenzt

# Forks

- zuwachs nicht linear
- Monitor bei neun Werten
- CPU begrenzt
- meine Beobachtung Playbook läuft in einem Prozess

# forks 60

1. time ansible-playbook –become-password-file=.password -i 60 test_mit3tasks.yml

# forks 60

1. time ansible-playbook –become-password-file=.password -i 60 test_mit3tasks.yml

```
ohne split
================================
dnf ---------------------- 10.32s
ansible.builtin.file -----  4.40s
shell --------------------  1.63s
debug --------------------  1.40s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 17.74s
```

1

# forks 60

3. time $( ansible-playbook –limit all[0:19] –become-password-file=.password -i 60 test_mit3tasks.yml > split_20_hosts_1_1  ansible-playbook –limit all[20:39] –become-password-file=.password -i 60 test_mit3tasks.yml > split_20_hosts_2_1 ansible-playbook –limit all[40:59] –become-password-file=.password -i 60 test_mit3tasks.yml > split_20_hosts_3_1  )

# forks 60

1. time ansible-playbook –become-password-file=.password -i 60 test_mit3tasks.yml
2. time $( ansible-playbook –limit all[0:19] –become-password-file=.password -i 60 test_mit3tasks.yml > split_20_hosts_1_1 ansible-playbook –limit all[20:39] –become-password-file=.password -i 60 test_mit3tasks.yml > split_20_hosts_2_1 ansible-playbook –limit all[40:59] –become-password-file=.password -i 60 test_mit3tasks.yml > split_20_hosts_3_1 )

1

```
ohne split
================================
dnf ---------------------- 10.32s
ansible.builtin.file -----  4.40s
shell --------------------  1.63s
debug --------------------  1.40s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 17.74s
```

2

```
split_20_hosts_1_1
================================
dnf ---------------------- 5.49s
ansible.builtin.file ----- 1.22s
debug -------------------- 1.04s
shell -------------------- 0.88s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 8.64s
split_20_hosts_2_1
================================
dnf ---------------------- 5.85s
ansible.builtin.file ----- 1.03s
debug -------------------- 0.85s
shell -------------------- 0.73s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 8.45s
```

```
split_20_hosts_3_1
================================
dnf ---------------------- 5.50s
ansible.builtin.file ----- 1.19s
debug -------------------- 0.95s
shell -------------------- 0.89s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total -------------------- 8.54s
```

# kleines Skript

```
echo "Übergabe $ubergabe"
enthalt_ansibleplaybook=$( egrep -c '^ansible-playbook' <<< $ubergabe )
if [[ $enthalt_ansibleplaybook -eq 0 ]]; then   echo "kein ansible-playbook enthalten";  exit 3; fi
enthalt_limits=$( grep -c '\--limit' <<< $ubergabe)
list_hosts=$( sed 's/ansible-playbook/ansible-playbook --list-hosts/' <<< $ubergabe)
anzahlhosts=$( $list_hosts | awk '/hosts/{ sub("\\(", "", $2);sub("\\):", "", $2);print $2 }')
if [[ -z $anzahlhosts ]]; then   echo "keine Hostanzahl bestimmbar";   exit 2; fi
weite=$( echo "$anzahlhosts/$anzahl_paralleler_aufrufe" | bc )
if [[ -z $weite ]]; then   echo "keine Weite bestimmbar";   exit 4; fi
#echo "Übergabe $ubergabe anzahl: $anzahlhosts"
if [[ $enthalt_limits -eq 0 ]]
then
  if [[ $weite -eq 0 ]]; then runmax=$[$anzahlhosts-1]; weite=1; else runmax=$[$anzahl_paralleler_aufrufe-1]; fi
  echo "runmax $runmax"
  for ((i=0 ; i<$runmax ; i++))
  do
     startwert=$[$i * $weite ]
     endwert=$[ ( ( ($i+1) * $weite) - 1 ) ]
     line=$( sed 's/ansible-playbook/ansible-playbook --limit all:['$startwert':'$endwert']/; s/$/&/' <<< $ubergabe  )
     echo $line
  $line&
  done
  startwert=$[$i*$weite]
  endwert=$[$anzahlhosts -1]
  line=$( sed 's/ansible-playbook/ansible-playbook --limit all:['$startwert':'$endwert']/; s/$/&/' <<< $ubergabe  )
  echo $line
  $line&
else
  if [[ $weite -eq 0 ]]  then runmax=$[$anzahlhosts-1];  weite=1;  else  runmax=$[$anzahl_paralleler_aufrufe-1]; fi
  echo "runmax $runmax"
  for ((i=0 ; i<$runmax ; i++))
  do
     startwert=$[$i*$weite]
     #endwert=$[($i+1)*($weite-1)]
     endwert=$[ ( ( ($i+1) * $weite) - 1 ) ]
     line=$( echo $ubergabe | awk '{ split($0, field); for( position in field){ word = field[position]; if( limit == 1
rd ~ /*$/){  sub(/*$/, "['$i':'$end']\"", word); print "Wort nach sub: " word }else{  sub(/$/, "['$startwert':'$endwer
d)}   }; if ( word ~ "--limit"){ limit=1}else{ limit=0}; line=line" "word } print line)'   )
```

https://gitlab.com/1fridy/etwas-weniger-kaffee-bitte

# kleines Skript

```
parallel.sh ansible-playbook -i 60 test_mit3tasks.yml

ansible-playbook -limit all[0:19] -i 60 test_mit3tasks.yml &
ansible-playbook -limit all[20:39] -i 60 test_mit3tasks.yml &
ansible-playbook -limit all[40:59] -i 60 test_mit3tasks.yml &
```

# weiter Parallelisieren

1. Nutzen mehrer Ausführungshosts

# weiter Parallelisieren

1. Nutzen mehrer Ausführungshosts
   - AAP/AWX
   - Polemarch
   - Rundeck (Enterprice)
   - Semaphore (runners)

# weiter Parallelisieren

1. Nutzen mehrer Ausführungshosts
   - AAP/AWX
   - Polemarch
   - Rundeck (Enterprice)
   - Semaphore (runners)
2. ansible-pull

# weiter Parallelisieren

1. Nutzen mehrer Ausführungshosts
   - AAP/AWX
   - Polemarch
   - Rundeck (Enterprice)
   - Semaphore (runners)
2. ansible-pull
   - Client fragt bei Server/Worker an
   - auf Client unterschiedliche Server/Worker eingestellt
   - Server/Worker hinter LB
   - ARA für callback

# Testplaybook

```
- hosts: all
  gather_facts: false
  tasks:
    - name: task 1
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 1 result
      ansible.builtin.debug:
        msg: "Task 1 Script result: {{ result }}"
    - name: task 2
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 2 result
      ansible.builtin.debug:
        msg: "Task 2 Script result: {{ result }}"
    - name: task 3
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 3 result
      ansible.builtin.debug:
        msg: "Task 3 Script result: {{ result }}"
    - name: task 4
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 4 result
      ansible.builtin.debug:
        msg: "Task 4 Script result: {{ result }}"
    - name: task 5
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 6
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 6 result
      ansible.builtin.debug:
        msg: "Task 6 Script result: {{ result }}"
    - name: task 7
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 7 result
      ansible.builtin.debug:
        msg: "Task 8 Script result: {{ result }}"
    - name: task 8
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 8 result
      ansible.builtin.debug:
        msg: "Task 8 Script result: {{ result }}"
    - name: task 9
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 9 result
      ansible.builtin.debug:
        msg: "Task 9 Script result: {{ result }}"
    - name: task 10
      ansible.builtin.shell: /bin/sleep 1
      register: result
    - name: task 10 result
      ansible.builtin.debug:
```

# 10 50 58 Hosts a 25 Forks

```
free                                 linear
============================== ==============================
task 10 result ---------- 1.36s task 7 ------------------ 3.36s
task 1 ------------------ 1.34s task 8 ------------------ 1.45s
task 2 ------------------ 1.19s task 1 ------------------ 1.43s
task 3 ------------------ 1.11s task 10 ----------------- 1.42s
task 8 ------------------ 1.09s task 9 ------------------ 1.35s
task 10 ----------------- 1.08s task 6 ------------------ 1.35s
task 7 ------------------ 1.07s task 5 ------------------ 1.34s
task 9 ------------------ 1.05s task 4 ------------------ 1.34s
task 5 ------------------ 1.05s task 2 ------------------ 1.33s
task 4 ------------------ 1.03s task 3 ------------------ 1.32s
task 6 ------------------ 1.03s task 10 result ---------- 0.20s
task 5 result ----------- 0.27s task 3 result ----------- 0.15s
task 8 result ----------- 0.24s task 5 result ----------- 0.13s
task 4 result ----------- 0.23s task 2 result ----------- 0.12s
task 6 result ----------- 0.22s task 4 result ----------- 0.12s
task 3 result ----------- 0.22s task 9 result ----------- 0.12s
task 9 result ----------- 0.21s task 7 result ----------- 0.11s
task 7 result ----------- 0.19s task 8 result ----------- 0.11s
task 2 result ----------- 0.17s task 6 result ----------- 0.11s
task 1 result ----------- 0.14s task 1 result ----------- 0.10s
============================== ==============================
ansible.builtin.shell -- 11.03s ansible.builtin.shell -- 15.70s
ansible.builtin.debug --- 3.27s ansible.builtin.debug --- 1.26s
------------------------------ ------------------------------
total ------------------- 14.29s total ------------------- 16.96s

real    0m14,922s              real    0m17,597s
user    0m8,585s               user    0m5,577s
sys     0m3,303s               sys     0m2,967s
```

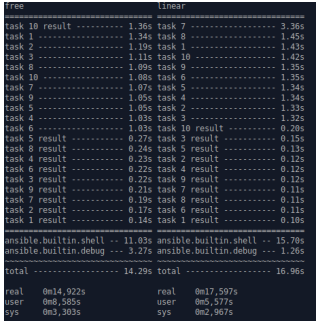
# 10 50 58 Hosts a 25 Forks

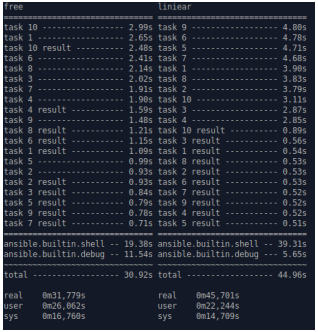
```
free                                liniear
=================================== ===================================
task 10 ------------------ 2.99s     task 9 ------------------ 4.80s
task 1 ------------------- 2.65s     task 6 ------------------ 4.78s
task 10 result ----------- 2.48s     task 5 ------------------ 4.71s
task 6 ------------------- 2.41s     task 7 ------------------ 4.68s
task 8 ------------------- 2.14s     task 1 ------------------ 3.90s
task 3 ------------------- 2.02s     task 8 ------------------ 3.83s
task 7 ------------------- 1.91s     task 2 ------------------ 3.79s
task 4 ------------------- 1.90s     task 10 ------------------ 3.11s
task 4 result ------------ 1.59s     task 3 ------------------ 2.87s
task 9 ------------------- 1.48s     task 4 ------------------ 2.85s
task 8 result ------------ 1.21s     task 10 result ----------- 0.89s
task 6 result ------------ 1.15s     task 3 result ------------ 0.56s
task 1 result ------------ 1.09s     task 1 result ------------ 0.54s
task 5 ------------------- 0.99s     task 8 result ------------ 0.53s
task 2 ------------------- 0.93s     task 2 result ------------ 0.53s
task 2 result ------------ 0.93s     task 6 result ------------ 0.53s
task 3 result ------------ 0.84s     task 7 result ------------ 0.52s
task 5 result ------------ 0.79s     task 9 result ------------ 0.52s
task 9 result ------------ 0.78s     task 4 result ------------ 0.52s
task 7 result ------------ 0.71s     task 5 result ------------ 0.51s
=================================== ===================================
ansible.builtin.shell -- 19.38s     ansible.builtin.shell -- 39.31s
ansible.builtin.debug -- 11.54s     ansible.builtin.debug --- 5.65s
----------------------------------- -----------------------------------
total -------------------- 30.92s     total -------------------- 44.96s

real    0m31,779s                   real    0m45,701s
user    0m26,062s                   user    0m22,244s
sys     0m16,760s                   sys     0m14,709s
```

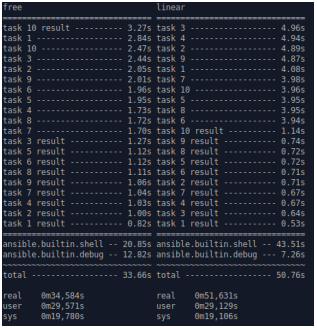
# 10 50 58 Hosts a 25 Forks

```
free                                 linear
==================================== ====================================
task 10 result ---------- 3.27s      task 3 ------------------ 4.96s
task 1 ------------------ 2.84s      task 4 ------------------ 4.94s
task 10 ----------------- 2.47s      task 2 ------------------ 4.89s
task 3 ------------------ 2.44s      task 9 ------------------ 4.87s
task 2 ------------------ 2.05s      task 1 ------------------ 4.08s
task 9 ------------------ 2.01s      task 7 ------------------ 3.98s
task 6 ------------------ 1.96s      task 10 ----------------- 3.96s
task 5 ------------------ 1.95s      task 5 ------------------ 3.95s
task 4 ------------------ 1.73s      task 8 ------------------ 3.95s
task 8 ------------------ 1.72s      task 6 ------------------ 3.94s
task 7 ------------------ 1.70s      task 10 result ---------- 1.14s
task 3 result ----------- 1.27s      task 9 result ----------- 0.74s
task 5 result ----------- 1.12s      task 8 result ----------- 0.72s
task 6 result ----------- 1.12s      task 5 result ----------- 0.72s
task 8 result ----------- 1.11s      task 6 result ----------- 0.71s
task 9 result ----------- 1.06s      task 2 result ----------- 0.71s
task 7 result ----------- 1.04s      task 7 result ----------- 0.67s
task 4 result ----------- 1.03s      task 4 result ----------- 0.67s
task 2 result ----------- 1.00s      task 3 result ----------- 0.64s
task 1 result ----------- 0.82s      task 1 result ----------- 0.53s
==================================== ====================================
ansible.builtin.shell -- 20.85s      ansible.builtin.shell -- 43.51s
ansible.builtin.debug -- 12.82s      ansible.builtin.debug --- 7.26s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ------------------ 33.66s      total ------------------ 50.76s

real    0m34,584s                    real    0m51,631s
user    0m29,571s                    user    0m29,129s
sys     0m19,780s                    sys     0m19,106s
```

# 10 50 58 Hosts a 25 Forks



# Mitogen

# Mitogen

1. forks 60 mitogen free/linear vs ohne mitogenfree/linear

```
1 mitogen
  linear              free
  ================  ================
  dnf ----- 120.06s  dnf ----- 165.00s
  debug ---  39.09s  debug ---  60.54s
  file ----  22.03s  file ----  43.88s
  shell ---  21.78s  shell ---  43.02s
  ~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
  total --- 202.96s  total --- 312.44s
  ohne mitogen
  linear              free
  ================  ================
  dnf ----- 100.98s  dnf ----- 126.73s
  debug ---  36.75s  debug ---  61.03s
  file ----  99.05s  file ---- 104.95s
  shell --- 100.56s  shell --- 110.64s
  ~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
  total --- 337.35s  total --- 403.36s
```

# Mitogen

1. forks 60 mitogen free/linear vs ohne mitogenfree/linear
2. mitogen forks 20 mitogen free/linea

1
```
mitogen
linear              free
================  ================
dnf ----- 120.06s  dnf ----- 165.00s
debug ---  39.09s  debug ---  60.54s
file ----  22.03s  file ----  43.88s
shell ---  21.78s  shell ---  43.02s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 202.96s  total --- 312.44s
ohne mitogen
linear              free
================  ================
dnf ----- 100.98s  dnf ----- 126.73s
debug ---  36.75s  debug ---  61.03s
file ----  99.05s  file ---- 104.95s
shell --- 100.56s  shell --- 110.64s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 337.35s  total --- 403.36s
```

2
```
mitogen
linear              free
================  ================
dnf ----- 156.75s  dnf ----- 184.31s
debug ---  59.73s  debug ---  60.94s
file ---- 117.33s  file ----  43.71s
shell --- 112.56s  shell ---  43.60s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 446.38s  total --- 332.56s
```

# Mitogen

1. forks 60 mitogen free/linear vs ohne mitogenfree/linear
2. mitogen forks 20 mitogen free/linea

1
```
mitogen
linear              free
================  ================
dnf ----- 120.06s  dnf ----- 165.00s
debug ---  39.09s  debug ---  60.54s
file ----  22.03s  file ----  43.88s
shell ---  21.78s  shell ---  43.02s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 202.96s  total --- 312.44s
ohne mitogen
linear              free
================  ================
dnf ----- 100.98s  dnf ----- 126.73s
debug ---  36.75s  debug ---  61.03s
file ----  99.05s  file ---- 104.95s
shell --- 100.56s  shell --- 110.64s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 337.35s  total --- 403.36s
```

2
```
mitogen
linear              free
================  ================
dnf ----- 156.75s  dnf ----- 184.31s
debug ---  59.73s  debug ---  60.94s
file ---- 117.33s  file ----  43.71s
shell --- 112.56s  shell ---  43.60s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 446.38s  total --- 332.56s
```

- läuft nur mit älteren Ansibleversionen

# Mitogen

1. forks 60 mitogen free/linear vs ohne mitogenfree/linear
2. mitogen forks 20 mitogen free/linea

1
```
mitogen
linear              free
================  ================
dnf ----- 120.06s  dnf ----- 165.00s
debug ---  39.09s  debug ---  60.54s
file ----  22.03s  file ----  43.88s
shell ---  21.78s  shell ---  43.02s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 202.96s  total --- 312.44s
ohne mitogen
linear              free
================  ================
dnf ----- 100.98s  dnf ----- 126.73s
debug ---  36.75s  debug ---  61.03s
file ----  99.05s  file ---- 104.95s
shell --- 100.56s  shell --- 110.64s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 337.35s  total --- 403.36s
```

2
```
mitogen
linear              free
================  ================
dnf ----- 156.75s  dnf ----- 184.31s
debug ---  59.73s  debug ---  60.94s
file ---- 117.33s  file ----  43.71s
shell --- 112.56s  shell ---  43.60s
~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~
total --- 446.38s  total --- 332.56s
```

- läuft nur mit älteren Ansibleversionen
- hat Einschränkungen bei become

# Ausblicke

Optimierung ist ein Prozess.

# Inhaltsverzeichnis

1. Das Wichtigste
2. Kaffee
3. Ansible
   - wie Ansible arbeitet
4. Messen
5. Optimieren
   - Gatherfacts
   - SSH
   - asynchron
   - Parallelisieren
   - Strategy
   - Mitogen
6. Ende

# Danke

Danke für die Aufmerksamkeit.