Ansible Resource Consumption

Resourcenverbrauch von Playbooks messen

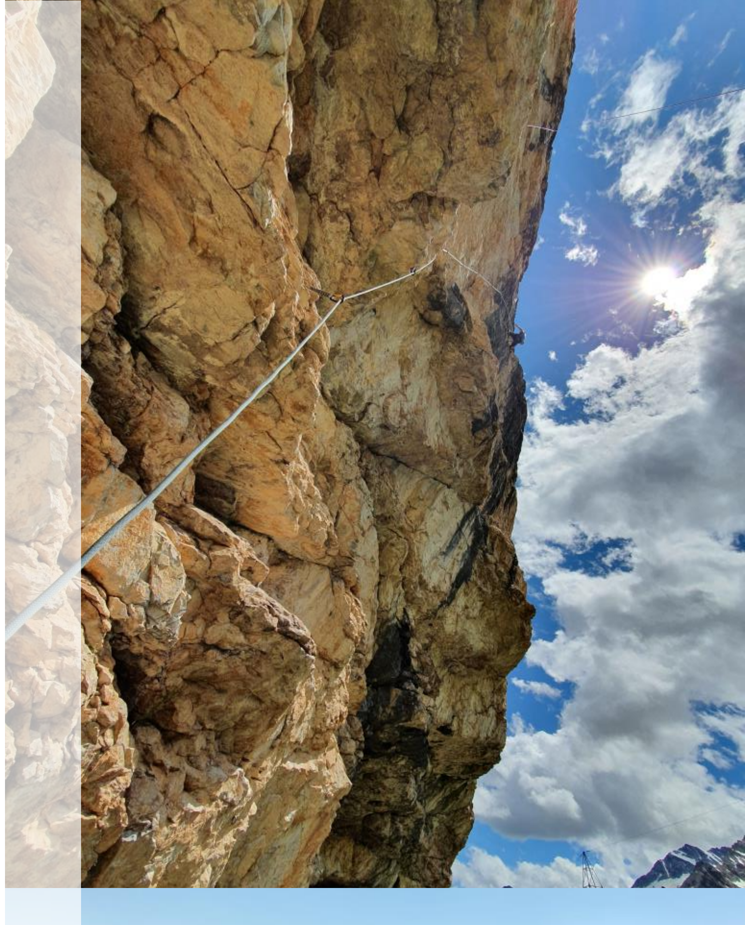
Computacenter

$ whoami

Tim Grützmacher

Senior Consultant
@ Computacenter since 2016

- started in Big Data - Hadoop
  - Big Data needs Automation
- 6+ years in DevOps & Automation
- Puppet, Terraform, but mostly Ansible
- Contributor to ansible-builder and multiple Ansible Collections



Callback Plugins

Overview

Different types available:

stdout callback

how Ansible outputs stuff to the CLI, you can use **only one stdout callback at a time**

Example plugins:

- *ansible.builtin.oneline* (single line per host/task)
- *community.general.null* (doesn't display stuff to screen)
- *community.general.counter_enabled* (adds counters to the output items)

aggregate callback

adds *additional* console output next to the configured stdout callback

Example plugins:

- ansible.builtin.junit (write playbook output to a JUnit formatted xml file)
- ansible.posix.timer (adds time to play stats)

notification callback

sends information of a playbook run to other applications, services, or systems

Example plugins:

- community.general.log_plays (write playbook output to log file)
- community.general.splunk (sends task result events to Splunk HTTP Event Collector)
- community.general.slack callback (sends play events to a Slack channel)

Link to all available callback plugins: Index of all Callback Plugins

How many resources does my playbook need?

Assess consumption with aggregate callback plugins

ansible.posix.cgroup_memory_recap

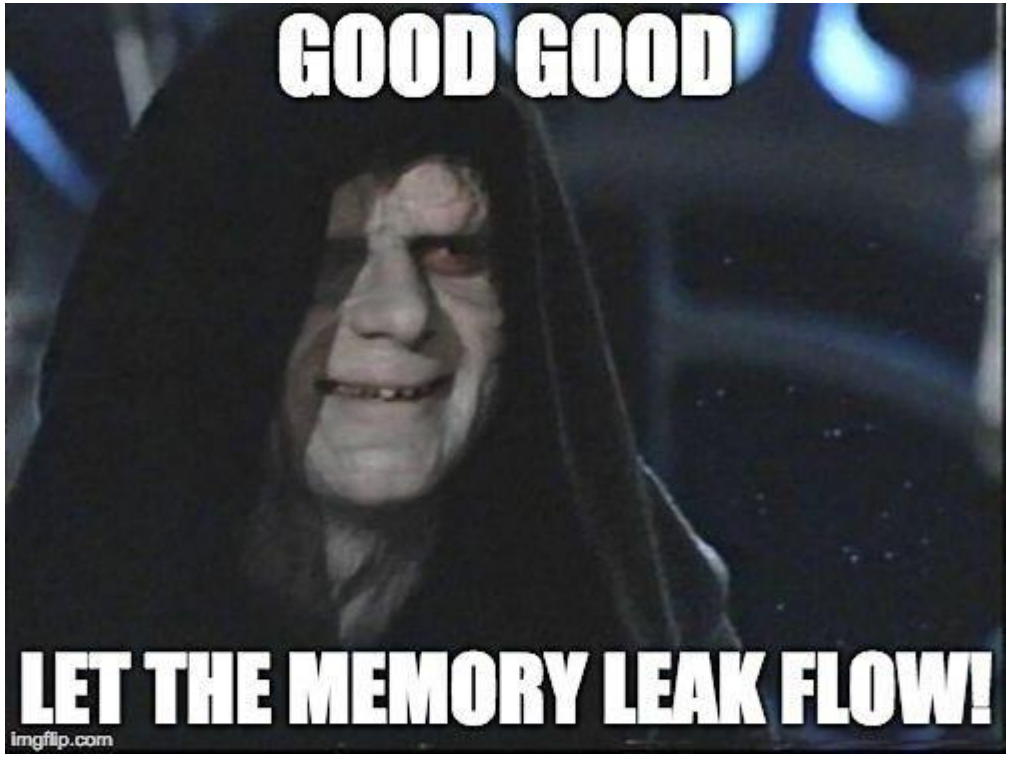
  - Profiles **maximum memory usage** of Ansible and individual tasks and displays a recap at the end

ansible.posix.cgroup_perf_recap

  - Profiles **system activity** of Ansible and individual tasks and display a recap at the end of the playbook execution.

How-to steps:

- Requires cgroups, install the *libcgroup-tools* (Fedora), *cgroup-tools* (Debian) package and create a *cgroup* profile
  - You'll need sudo for this!
- Install necessary collection (ansible.posix)
- Adjust your ansible.cfg
- You have to reference the *cgroup* profile when running your playbook



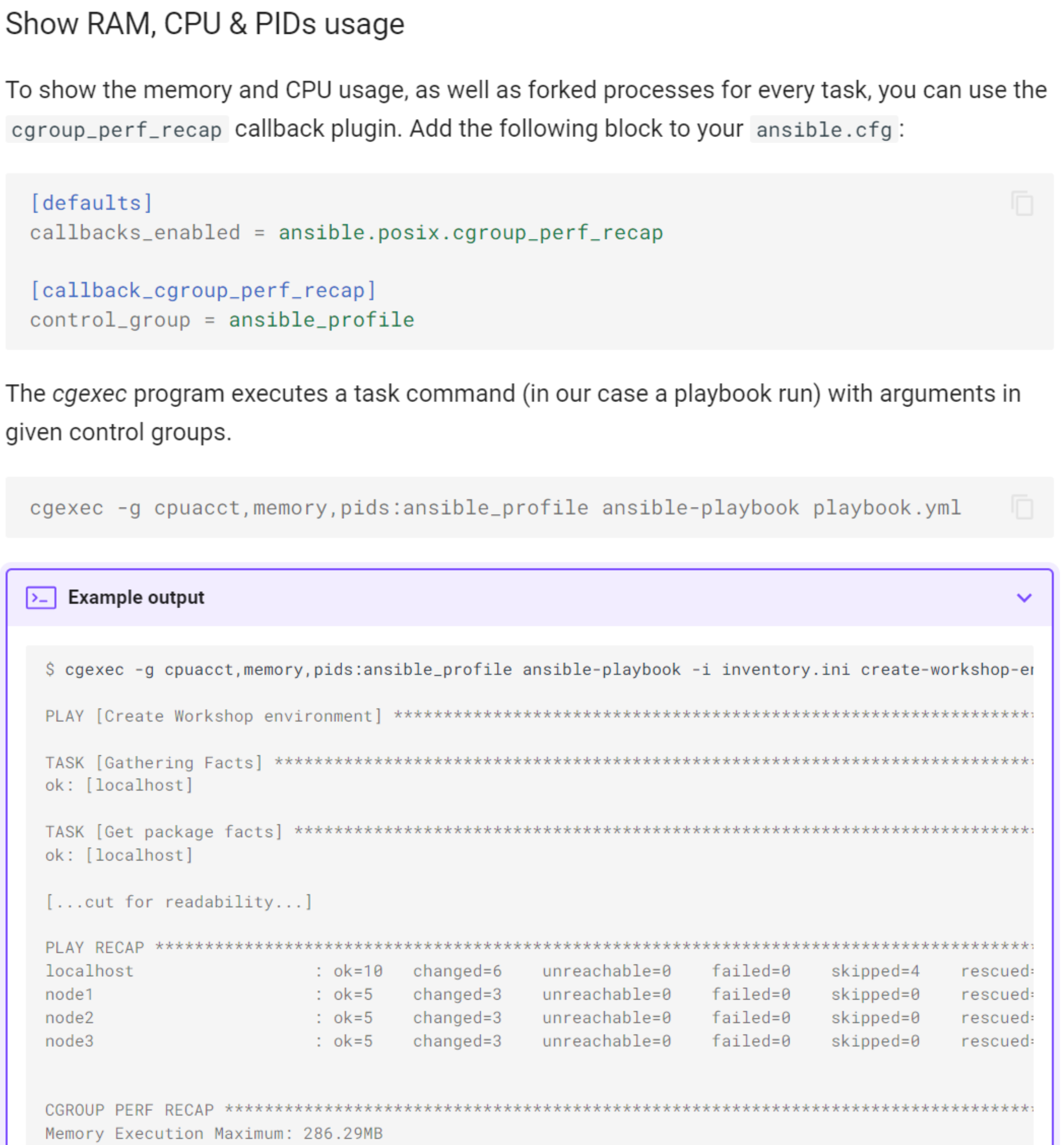
Demo

Demo Recap

How-to Guide and Example Output can be found here:

[Monitoring & Troubleshooting - Ansible Best Practices](https://timgrt.github.io/Ansible-Best-Practices/development/monitoring/#how-much-resources-are-consumed)

(https://timgrt.github.io/Ansible-Best-Practices/development/monitoring/#how-much-resources-are-consumed)



Show RAM, CPU & PIDs usage

To show the memory and CPU usage, as well as forked processes for every task, you can use the `cgroup_perf_recap` callback plugin. Add the following block to your `ansible.cfg`:

```
[defaults]
callbacks_enabled = ansible.posix.cgroup_perf_recap

[callback_cgroup_perf_recap]
control_group = ansible_profile
```

The *cgexec* program executes a task command (in our case a playbook run) with arguments in given control groups.

```
cgexec -g cpuacct,memory,pids:ansible_profile ansible-playbook playbook.yml
```

Example output

```
$ cgexec -g cpuacct,memory,pids:ansible_profile ansible-playbook -i inventory.ini create-workshop-e

PLAY [Create Workshop environment] *******************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [localhost]

TASK [Get package facts] *****************************************************************************
ok: [localhost]

[...cut for readability...]

PLAY RECAP *******************************************************************************************
localhost                  : ok=10   changed=6    unreachable=0    failed=0    skipped=4    rescued=
node1                      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=
node2                      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=
node3                      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=


CGROUP PERF RECAP ************************************************************************************
Memory Execution Maximum: 286.29MB
```

Danke