

Ansible: There's a module for that!

but what if there isn't?

Marc Richter

Principal Technical Account Manager

Brian Jarvis

Technical Account Manager

What we'll be discussing today

Intros

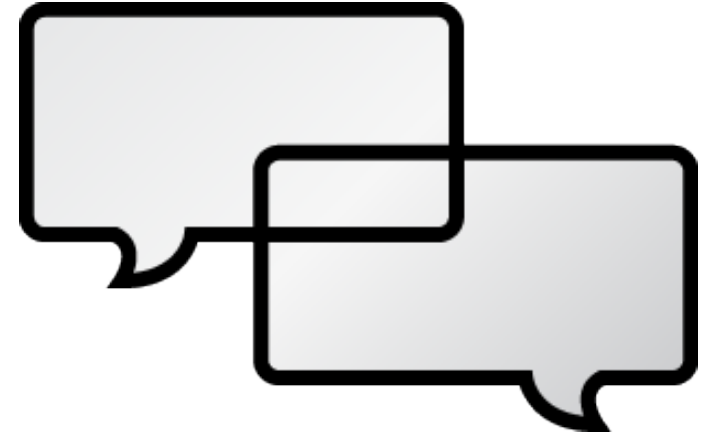
Extending Ansible

The Problem

A hacky solution

A better solution

Thoughts



/whois unclenarc



Linux Nerd since 1998
RHCE, cause that's fun
Red Hatter since 2015
- Principal Technical Account Manager
Scout Leader since 2009
1 wife, 4 kids, 3 dogs = mild chaos

/whois bry

RHCE, cause wanna be like unclemarc
Red Hatter since 2018

- Technical Account Manager - OpenShift
1 wife, 4 kids, 2 dogs, 2 cats, 2 grand kids

What happens when my automation requirements exceed Ansible's capabilities?

Ansible Roles

Roles provide a framework for fully independent, or interdependent collections of variables, **tasks**, files, templates, and **modules**.

What are **modules**?

Discrete units of code can be run as a task or as an ad-hoc command.

- Also called task plugins or library plugins
- Written in Python
- Execute on the remote target node
- Idempotent

And **plugins** too?

- Plugins run before the module
 - An action plugin runs before every module
- Written in Python
- Execute on the ansible control node
- Filters are a special type of plugin from Jinja2

The Problem!

So What's The Problem We're Trying to Solve?

Control Groups

Performance tuning feature. Services and users can have limits set on CPU, memory, storage and a few other items


systemd Controls Most Cgroups

While not all of the controllers are under systemd's purview, most are and it's best practice to use systemd to manipulate the settings

Ansible is systemd aware

There's an actual module for controlling many services already. BUT this module **does not** support the set-property feature

[systemd module] configure cgroups #54302

 Closed juju4 opened this issue on Mar 24 · 3 comments



juju4 commented on Mar 24

...

SUMMARY

systemd module implements mostly the service management part from https://docs.ansible.com/ansible/latest/modules/systemd_module.html

It would be useful to allow other commands like set-property which allows to configure cgroups like

```
- name: Enforce cgroups limit on osqueryd
  command: >
    systemctl set-property osqueryd.service
      CPUQuota={{ osquery_cgroups_cpuquota | default('10%') }}
      MemoryLimit={{ osquery_cgroups_memorylimit | default('1G') }}
      CPUShares={{ osquery_cgroups_cpushares | default('100') }}
  args:
    creates: /etc/systemd/system/osqueryd.slice.d/50-CPUQuota.conf
```

ISSUE TYPE

- Feature Idea



bcoca commented on Apr 9

Member ...

this looks like a job for a template file or at most a role that uses such a template.



bcoca commented on Apr 9

Member ...

closing as per above



bcoca closed this on Apr 9

We want the systemd module to support set-property!

Use a template!



Possible Solution #1

Use the “command” module

One can use the command module to run an arbitrary command on a host. Seems simple enough

```
---  
- name: cap_sshd  
  hosts: all  
  tasks:  
    - command: "systemctl set-property sshd.service CPUQuota=50%"  
    - become: yes
```

Solution #1 is Terrible

It's hard coded

The command sets one specific service to one specific setting. It can't scale or change based on anything

It's not idempotent

Regardless of the current state of the system, the command is always run. This is not the expected behavior for Ansible (change only if needed)

It's not easy to read

It's not the normal key pair format of a playbook so a novice needs to decode the actual command

Solution #2 – Create a Role

```
---
- name: Apache is important
  hosts: web
  vars:
    cgroup_limits:
      - service: httpd
        setting: CPUShares
        value: 2048
      - service: mscan
        setting: CPUQuota
        value: 10%
      - service: ramhogd
        setting: MemoryLimit
        value: 512M
  roles:
    - CGrouper
```

We can abstract the task so it can scale

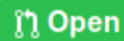
It's still not idempotent

Regardless of the current state of the system, the command is always run. This is not the expected behavior for Ansible (change only if needed)

But it does suck less as it's easier to read.

An inside look at CGrouper

```
---  
# tasks file for CGrouper  
- name: execute systemctl set-property for each setting  
  command: "systemctl set-property {{ item.service }} {{ item.setting }}={{ item.value }}"  
  become: yes  
  with_items: "{{ cgroup_limits }}"
```

brian-jarvis wants to merge 8 commits into `ansible:devel` from `brian-jarvis:bry-properties`

Conversation 5

Commits 8

Checks 0

Files changed 1



brian-jarvis commented on Oct 7



SUMMARY

Add setting properties to the systemd module by executing the `systemctl set-property` command.

ISSUE TYPE

- Feature Pull Request

COMPONENT NAME

systemd

ADDITIONAL INFORMATION

Specify systemd properties as dictionary items. Output from module is updated after any changes to reflect the change in the results.

- name: set cpushare and memory on sshd.service
systemd:
name: sshd.service
properties:
CPUShares: 750
MemoryLimit: 750M



1



1



sivel commented on Oct 7

Member



We decided somewhat recently that we did not want to accept such a feature: [#54302](#)



Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat