

Ansible ...

"Half of it nobody understands anyway."



Dr. Bernhard Hopfenmüller

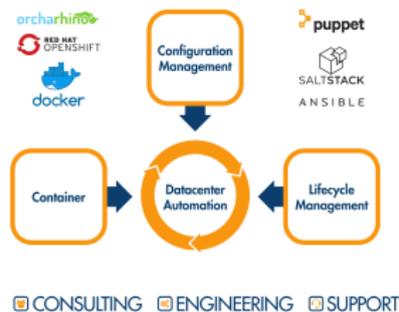
16.10.2018

Bernhard Hopfenmüller
IT Consultant @ ATIX AG

IRC: Fobhep
github.com/Fobhep

The Linux & Open Source Company
Garching @ München

over 15 years
datacenter automation, Linux
Consulting, Engineering, Support,
Training



orchardhino

RECYCLE DEPLOY
GEAR RUN
CHECK CONTROL





imdb.com

Ansible ...

- ▶ ... is „radically simple“
- ▶ ... requires: 1 Unixnode, SSH, Python
- ▶ ... needs no agent
- ▶ ... describes the desired state
- ▶ ... is written in YAML
- ▶ ... satisfies idempotency



- ▶ User: Write Statements
- ▶ Ansible: Create program
- ▶ Ansible: Copy program to Run-Node
- ▶ Ansible: Connect to Run-Node (local, SSH, PowerShell)
- ▶ Ansible: Run Program
- ▶ Ansible: Communication via JSON
- ▶ Ansible: Delete local Program
- ▶ Ansible: Give Feedback to User



- ▶ **Inventory** - infrastructure as code
- ▶ **Adhoc mode** - one task + terminal
- ▶ **Playbook** - collection of tasks/roles
- ▶ **Role** - one collection of tasks
- ▶ **Task** - one Ansible command:
'apt-get install nano'
- ▶ **Jinja 2 templating** - e.g. for variables
- ▶ **Modules** - core-components:
yum module, ping module,...
communication via JSON

```
[dbserver]
db1.server.com
db2.server.com

[webserver]
web1.server.com ansible_host=4.8.15.16
web2.server.com ansible_host=4.8.23.42

[appserver]
app1.server.com default_port=23
app2.server.com default_port=69105
```

- ▶ defining the infrastructure
- ▶ ini or yaml syntax
- ▶ combine servers into groups
- ▶ define specific vars

- ▶ run only one task quickly
- ▶ either one-timer or quicker than playbook

```
$ ansible webservers -m patch \
-a \
  "src=/src/critical.patch \
  dest=/app/bin/base"

$ ansible webservers -a "sbin/reboot"

$ ansible webservers -m ping
```

```
---  
- hosts: webservers # where ?  
  become: true # run as root  
  roles:  
    - bake_server # role1  
    - gitlab      # role2
```

- ▶ combination of roles
- ▶ written in YAML (or JSON)-Syntax
- ▶ ideally in combination with VCS

- ▶ combination of tasks
- ▶ featuring idempotency
- ▶ ideally in combination with VCS

```
---
- name: Install GitLab dependencies
  yum:
    name: "{{ item }}"
    state: latest
  loop:
    - openssh-server
    - postfix
    - curl
    - openssl

- name: Install GitLab
  yum:
    name: "{{ gitlab-version }}"
    state: present

- name: Copy config file
  template:
    src: gitlab.rb.j2
    dest: /etc/gitlab/gitlab.rb
    mode: 0600
  notify: restart gitlab
...
```

Questions - so far?



<https://imgflip.com/>



unrelated cat-content

YAML - everything easy, or not?!



My variable is not working!!"

YAML - everything easy, or not?!

```
---  
a_dict:  
  foo: "fine!"
```

YAML - everything easy, or not?!

```
--- # don't forget this!  
a_dict:  
  foo: "fine!"  
  foo42: "meh- but ok"
```

YAML - everything easy, or not?!

```
--- # don't forget this!  
a_dict:  
  foo: "fine!"  
  foo42: "meh- but ok"  
  foo-two: "no!"
```

YAML - everything easy, or not?!

```
--- # don't forget this!  
a_dict:  
  foo: "fine!"  
  foo42: "meh- but ok"  
  foo-two: "no!"  
  foo.bar: "oh god, noo!"
```

YAML - everything easy, or not?!

```
--- # don't forget this!  
a_dict:  
  foo: "fine!"  
  foo42: "meh- but ok"  
  foo-two: "no!"  
  foo.bar: "oh god, noo!"  
  foo bar: "what??!"
```

```
--- # don't forget this!  
a_dict:  
  foo: "fine!"  
  foo42: "meh- but ok"  
  foo-two: "no!"  
  foo.bar: "oh god, noo!"  
  foo bar: "what??!"  
  12: "... o0"
```

```
--- # don't forget this!  
a_dict:  
  foo: "fine!"  
  foo42: "meh- but ok"  
  foo-two: "no!"  
  foo.bar: "oh god, noo!"  
  foo bar: "what??!"  
  12: "... o0"  
  foo_12: "yes!!!"
```

YAML - everything easy, or not?!



"Ansible is not working!"

```
---  
ver_1: 2.4.2  
ver_2: 2.5  
  
- name: Diff versions  
  debug:  
    msg: "Version 1 is smaller!"  
    when: ver1 < ver2 # Nope!
```

```
---  
ver_1: "2.4.2"  
ver_2: "2.5"  
  
- name: Diff versions  
  debug:  
    msg: "Yeah!"  
    when: ver1 < ver2 # YipYip!
```

Filters ftw!!

```
{{ ver2 | version_compare('ver1', '>=') }}
```



"We need this, but it's impossible in Ansible!"

Add a new entry to a dict? Make use of implemented filters!

```
---
vars:
  dict:
    foo: "bar"
    foo2: "bar2"

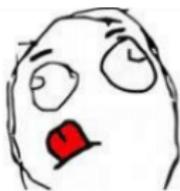
  tasks:
    - set_fact:
      dict: "{ dict | combine( { 'foo3': 'bar3' } ) }"
...

```

Fair enough - but what about removing a key?

Real life example: PUT- Request to REST - API

```
body = {  
  "id": "01189998819991197253",  
  "label": "plugin",  
  "skip_production": true #< we don't want this!  
}
```

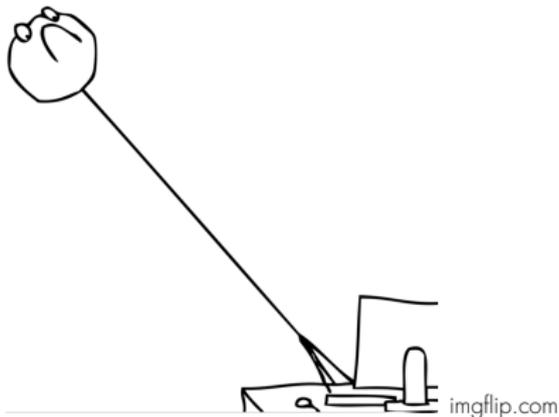


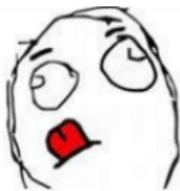
imgflip.com

a while later...

```
---  
- set_fact:  
  body: "{{ { } |combine({item.key: item.value}) }}"  
  when: "{{ {item.key not in ['skip_production'] }}"  
  with_dict: "{{ { body } }}"  
...
```

```
---  
- set_fact:  
  body: "{{ { } |combine({item.key: item.value}) }}"  
  when: "{{item.key not in ['skip_production'] }}"  
  with_dict: "{{ { body } }}"  
...
```





imgflip.com

another while later...

```
#filter_plugins/remove_key.py
#!/usr/bin/python
class FilterModule(object):
    def filters(self):
        return {
            'remove_key': self.remove_key
        }

    def remove_key(self,old_dict,key):
        old_dict.pop(key, None)
        return old_dict
```

```
#filter_plugins/remove_key.py
#!/usr/bin/python
class FilterModule(object):
    def filters(self):
        return {
            'remove_key': self.remove_key

        }

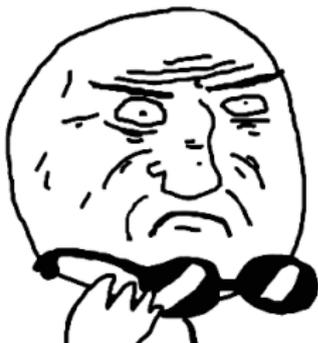
    def remove_key(self,old_dict,key):
        old_dict.pop(key, None)
        return old_dict
```

```
---
- set_fact:
    body: "{{ body | remove_key('skip_production') }}"
...

```

```
---  
- set_fact:  
  body: "{{ body | remove_key('skip_production') }}"  
...
```

```
---  
- set_fact:  
  body: "{{ body | remove_key('skip_production') }}"  
...
```



imgflip.com

accurate face expression

""Wait? There's more than ansible and ansible-playbook?""

ansible-galaxy

- ▶ search public roles
- ▶ create a template for roles
- ▶ ...



ansible-doc

- ▶ offline documentation
- ▶ any plugin type
- ▶ ...

ansible-console

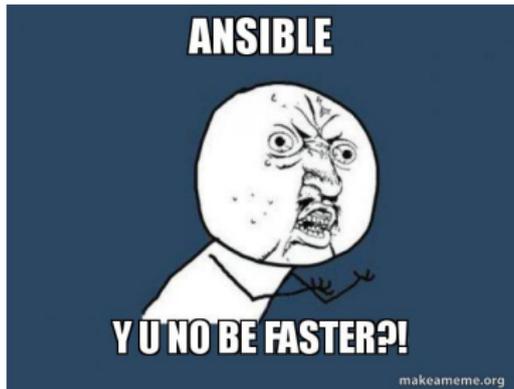
- ▶ Ansible REPL
- ▶ test stuff quickly, e.g. jinja...
- ▶ ...



ansible-pull

- ▶ automate your automation
- ▶ reset system automatically
- ▶ speed up!
- ▶ ...

It's slooow...



What are you doing Ansible



- ▶ scan used module for python deps

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir
- ▶ create temp dir in user home

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir
- ▶ create temp dir in user home
- ▶ sftp ZIP

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir
- ▶ create temp dir in user home
- ▶ sftp ZIP
- ▶ run script on target (16 quote layering! 10 steps including file operations)

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir
- ▶ create temp dir in user home
- ▶ sftp ZIP
- ▶ run script on target (16 quote layering! 10 steps including file operations)
- ▶ rm temp dir on target

What are you doing Ansible



- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir
- ▶ create temp dir in user home
- ▶ sftp ZIP
- ▶ run script on target (16 quote layering! 10 steps including file operations)
- ▶ rm temp dir on target

What are you doing Ansible

- ▶ scan used module for python deps
- ▶ add create script and add it and deps to ZIP-file
- ▶ base64 encoding of ZIP-file
- ▶ write content to local filesystem for sftp
- ▶ find target user home dir
- ▶ create temp dir in user home
- ▶ sftp ZIP
- ▶ run script on target (16 quote layering! 10 steps including file operations)
- ▶ rm temp dir on target



```
forks = 5 # default  
# parallel Ansible processes i.e. number of hosts
```

```
- name: I will work while the rest goes on
  pip:
    requirements: requirements.txt
  async: 1000 # choose this high enough!
  poll: 0 # fire and forget!
  register: pip_install

- name: More tasks...
  [...]
```

- name: Deploy a python app.. I might crash :(

```
- name: I will work while the rest goes on
  pip:
    requirements: requirements.txt
  async: 1000 # choose this high enough!
  poll: 0 # fire and forget!
  register: pip_install

- name: More tasks....
  [...]
```

```
- name: PIP result check
  async_status:
    jid: "{{ pip_install.ansible_job_id }}"
  register: job_result
  until: job_result.finished
  retries: 30

- name: Deploy a python app.. I am happy now :)
```

```
[ssh_connection]
  pipelining = True
```

Careful: This needs to disable requiretty' in /etc/sudoers on all managed hosts

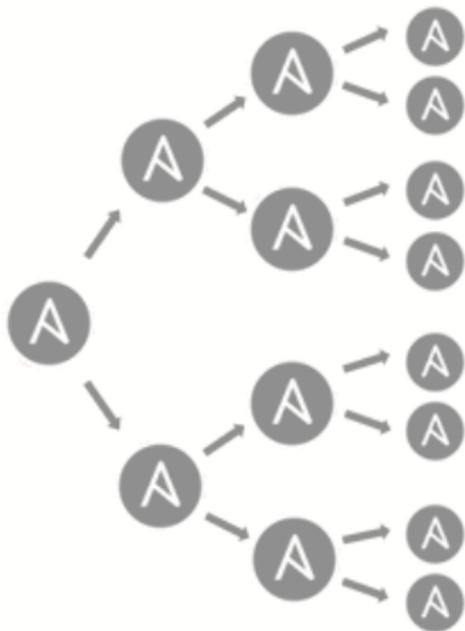
- ▶ Python library

- ▶ Python library
- ▶ self reproducing programs (cells: Mitosis)



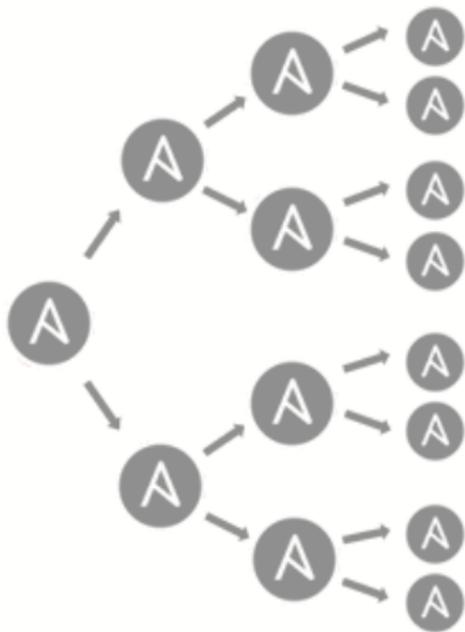
https://mitogen.readthedocs.io/en/latest/_images/cell_division.png

- ▶ Python library
- ▶ self reproducing programs (cells: Mitosis)
- ▶ “network-capable fork() on steroids”



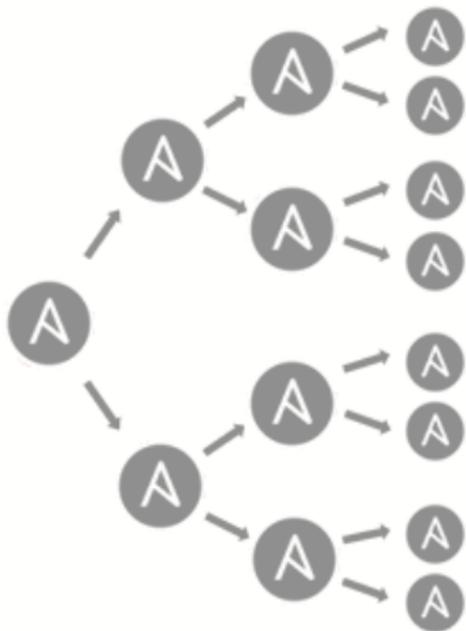
https://mitogen.readthedocs.io/en/latest/_images/cell_division.png

- ▶ Python library
- ▶ self reproducing programs (cells: Mitosis)
- ▶ “network-capable fork() on steroids”
- ▶ “lazily-loaded duplicates on remote hosts, without requiring any upfront remote disk writes”



https://mitogen.readthedocs.io/en/latest/_images/cell_division.png

- ▶ tl;dr: similar to SSH-Pipeline but no "require-ty problem"



https://mitogen.readthedocs.io/en/latest/_images/cell_division.png

- ▶ tl;dr: similar to SSH-Pipeline but no "require-tty problem"
- ▶ facts only: "1.25x - 7x speedup + CPU usage reduction $\geq 2x$ "



https://mitogen.readthedocs.io/en/latest/_images/cell_division.png

The End



More questions - your story?

atix.de/training/ansible-training/

- ▶ 2 or 3 days
- ▶ Hands-on based
- ▶ topics include
 - ▶ Ansible basics
 - ▶ From Paper to Playbook
 - ▶ Jinja Templating
 - ▶ Ansible + VCS
 - ...

