



One role to rule them all



Ansible in a real world environment

Ton Kersten

Antwerp / Belgium / 2019

ANSIBLE



\$ who am i

- name: Ton Kersten

creds:

work: UNIX/Linux consultant and Trainer @ [AT Computing](#)

linux: UNIX/Linux Geek

cfgmgmt: Configuration Management Addict

- ansible:

- Ansible user and contributor since 2012

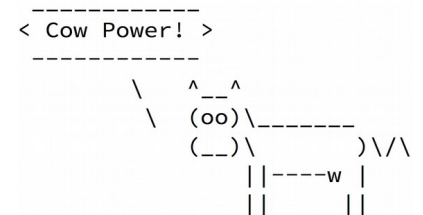
- Ansible Ambassador since 2015

- puppet:

- Puppet user since 2009

foss: Free and Open Source Software Enthusiast

works: Big fan of things that just work



What the customer wants

- Virtual machines on VMware
- Simple PXE provisioning
- Minimum of four environments: `dev`, `tst`, `acc` and `prd`
- Ansible `cfmgmt` for the complete environment
- Everything in a single setup
- No Ansible Tower and no AWX
 - ⇒ No API callbacks
- Very KISS
- Decent documentation
- Some way to monitor Ansible runs
- Easy to browse and search documentation
- ...

```
-----  
< Cow Power! >  
-----  
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
           ||----w |  
           ||     ||
```

What we created

- Git server with all repositories – `gitlab`
- One repository called `setup` containing:
 - Complete static inventory
 - Multiple environment definitions (at least)
`dev`, `tst`, `acc`, `prd`
 - All variables \Rightarrow `group_vars` and `host_vars`
 - File with all needed roles \Rightarrow `roles.yml`
 - File with all needed Galaxy roles \Rightarrow `galaxy.yml`
 - All involved playbooks and task lists
 - All needed scripts to make it tick: `refresh` and `ansible_run`
- All functionality in separate roles
- Server with all software repositories and PXE-boot
- ARA Records Ansible for monitoring
- Documentation site with `mkdocs`

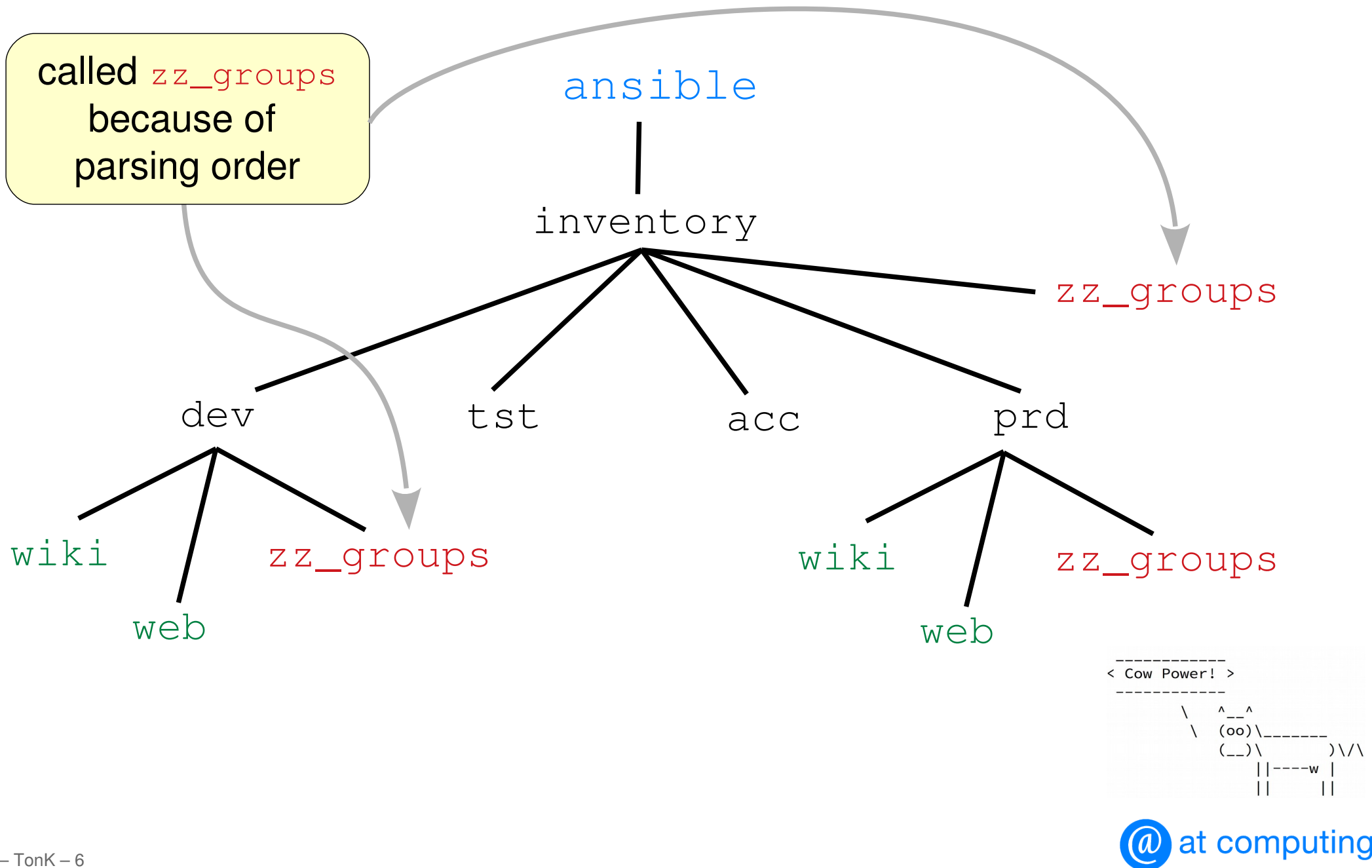
```
-----  
< Cow Power! >  
-----  
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
           ||----w |  
           ||     ||
```

Inventory design

- Static inventory directory called `inventory` containing
 - A group directory for `dev`, `tst`, `acc` and `prd`
 - A child definition group file \Rightarrow `zz_groups`
 - A group directory per functional group
 - A child definition group file \Rightarrow `zz_groups`

```
-----  
< Cow Power! >  
-----  
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
      ||----w |  
      ||     ||
```

Inventory layout – Host definitions



Inventory layout – groups

dev/wiki

```
[dev_wiki]  
wiki1.dev.example.net
```

dev/web

```
[dev_web]  
web1.dev.example.net  
web2.dev.example.net
```

dev/zz_groups

```
[dev:children]  
dev_web  
dev_wiki
```

zz_groups

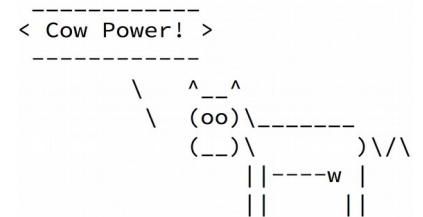
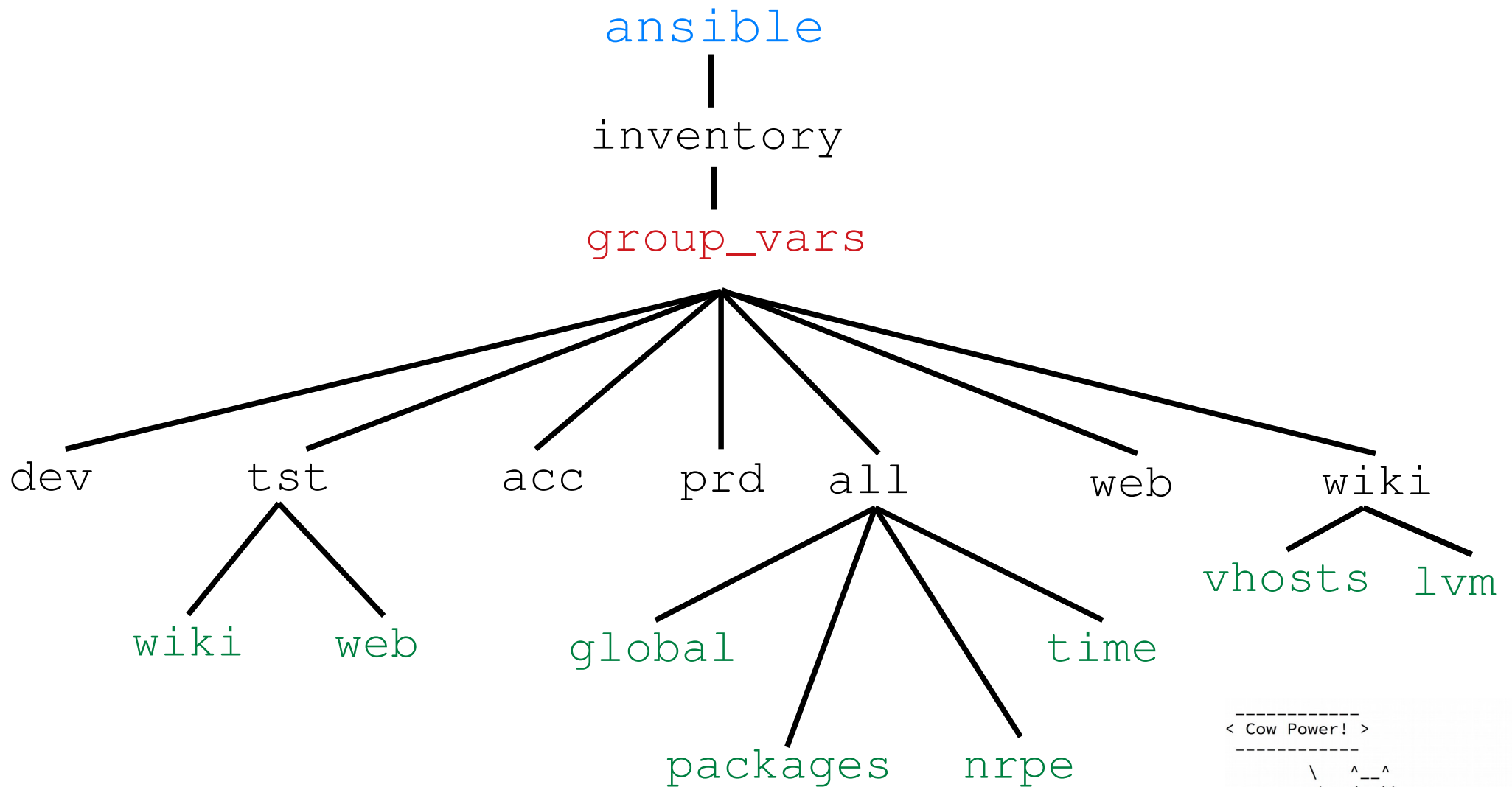
```
[wiki]  
dev_wiki  
tst_wiki  
acc_wiki  
prd_wiki
```

```
[web]  
dev_web  
tst_web  
acc_web  
prd_web
```

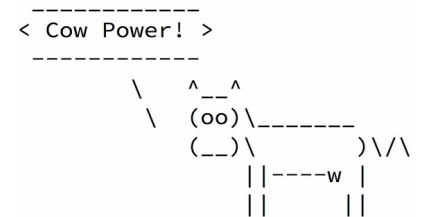
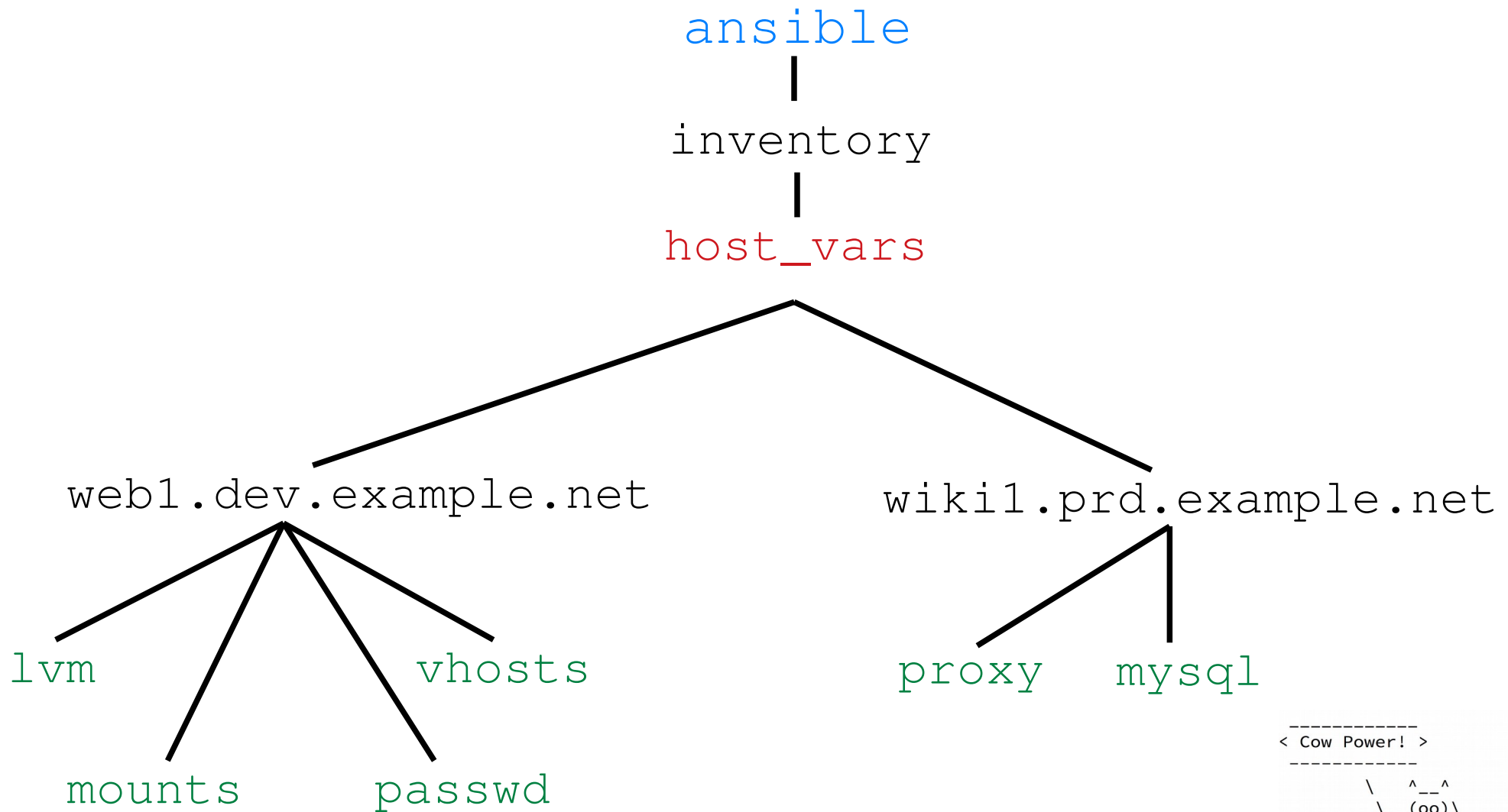
< Cow Power! >

```
\  ^__^  
 \ (oo)\_____   
  (__)\       )\/\   
     ||----w |   
     ||     ||
```

Inventory layout – group_vars definitions



Inventory layout – host_vars definitions



Variable definitions

role name,
to prevent
name clashes

dev/wiki

```
mysql_users:  
  - name: localweb-admin  
    host: '192.168.0.%'  
    password: !vault |  
      $ANSIBLE_VAULT;1.1;AES256  
      623435...201902051455  
      656464...201202231417  
    priv: '*.*:SELECT'
```

encrypted with

```
printf "${str}" | \  
ansible-vault \  
encrypt_string \  
--stdin-name="${name}" \  
--vault-password-file=${vault}
```

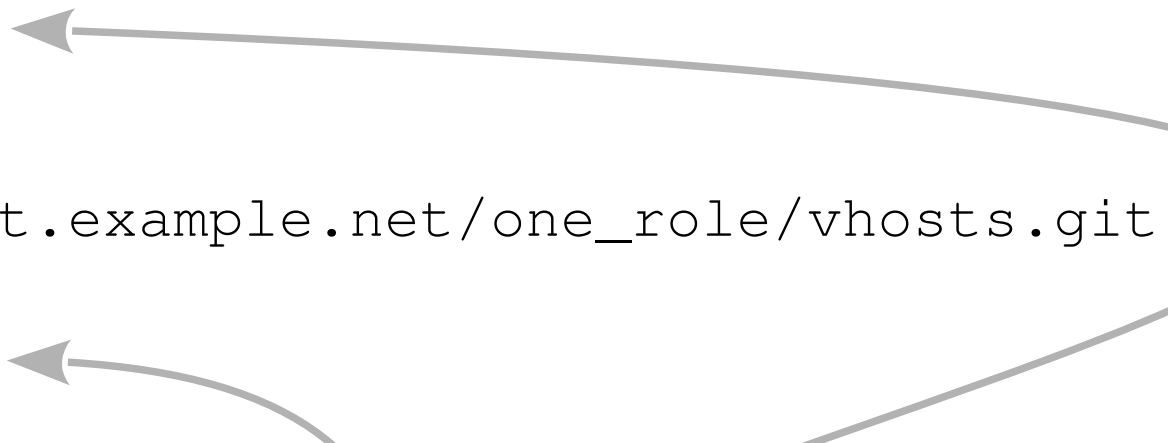
< Cow Power! >

```
\      ^__^  
 \    (oo)\_____  
  (__)\\       )\/\  
       ||----w |  
       ||     ||
```

Roles file

roles.yml

```
---  
- src: https://git.example.net/one_role/apache.git  
  scm: git  
  version: master  
  name: apache  
  
- src: https://git.example.net/one_role/vhosts.git  
  scm: git  
  version: master  
  name: vhosts
```



version ignored
used by refresh script

< Cow Power! >

```
\      ^__^  
 \    (oo)\_____  
  (__)\\       )\\/\  
     ||----w |  
     ||     ||
```

Pre – playbook

pre.yml

- copy:
 - content: '{ "managed": "{{ ansible_managed }}" }'
 - dest: /etc/ansible/facts.d/ansible_managed.fact
 - check_mode: no
- setup:
 - filter: ansible_local
- group_by:
 - key: "ansiblemanaged_\n {{ ansible_local.ansible_managed.managed }}"
 - changed_when: False

```
-----  
< Cow Power! >  
-----  
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
      ||----w |  
      ||     ||
```

Playbooks

dev.yml

pre.yml creates
ansiblemanaged groups

```
---
- import_playbook: pre.yml
- name: run all for 'dev'
  hosts: ansiblemanaged_True:&dev
  user: ansible
  become: True

tasks:
  - name: dev | include "common" tasks
    import_tasks: tasks/common.yml
    tags: [ common ]

  - name: dev | include "wiki" tasks
    import_tasks: tasks/wiki.yml
    when: "'wiki' in group_names"
    tags: [ wiki ]
```

< Cow Power! >

```
\      ^__^
 \    (oo)\_____
  (__)\\        )\/\
       ||----w |
       ||     ||
```

Task lists

tasks/common.yml

```
---
- include_role:
    name: firewallld
    tags: [ firewall ]

- include_role:
    name: environment
    tags: [ environment ]

- include_role:
    name: common
    tags: [ common ]

- include_role:
    name: rsyslog
    tags: [ rsyslog ]
```

tasks/wiki.yml

```
---
- include_role:
    name: apache
    tags: [ wiki, apache ]

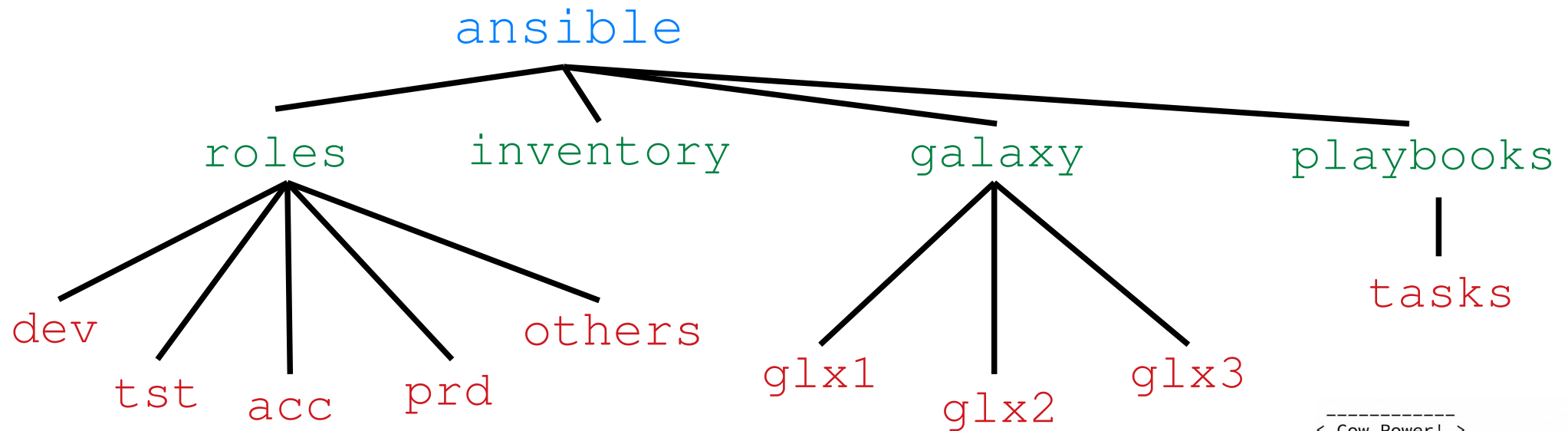
- include_role:
    name: wiki
    tags: [ wiki ]

- include_role:
    name: vhosts
    tags: [ wiki, vhosts ]
```

```
-----
< Cow Power! >
-----
      ^__^
      (oo)\_______
      (__)\\       )\/\
           ||----w |
           ||     ||
```

Ansible tree

- All roles in Git
- Every role with different branches
 - dev, tst, acc, prd or more
- Special script to create Ansible roles tree ⇒ `refresh`



```
ANS="/etc/ansible"
```

```
ANSIBLE_ROLES_PATH=${ANS}/roles/${envi}:${ANS}/galaxy
```

```
< Cow Power! >
  ^__^
  (oo)\_______
  (__)\\       )\/)
      ||----w |
      ||     ||
```

Putting it together

On your develop machine

- Edit role in `dev` branch, test, commit and push
Later: Merge with `tst`, `acc` and `prd`

On the Ansible control node

- Login as `root`
- Go to the Ansible tree, e.g. `/etc/ansible`
- Refresh all roles for the `dev` environment
`./refresh -f dev`
- Run Ansible with the `dev.yml` playbook for the `dev` environment
`ansible_run -l wiki.dev.example.net dev dev`

install all `git` branches
for `dev` environment

limit to host

this environment

this playbook

< Cow Power! >

```
\  ^__^
 \  (oo)\_____
  (__) \       )\/\
       ||----w |
       ||     ||
```


Questions

Contact me

- - Ton.Kersten@ATComputing.nl
 - Santa@TonKersten.com
- - <http://www.atcomputing.nl>
 - <http://www.tonkersten.com>
- <https://github.com/one-role>
- <https://github.com/tonk>
- <https://speakerdeck.com/tonk>
- @TonKersten on Twitter
- TKersten on IRC

