



Ansible Rollen testen mit Molecule

Tom Morelly // Philipp Felwor

SVA

/ Agenda

/ **Einleitung**

/ **Tipps & Tricks**

/ **Getting Started**

/ **Resources**

/ **Hands-On**

Einleitung

/ Wer sind Wir

Tom Morelly



25 Jahre

System Engineer

System Vertrieb Alexander GmbH

@FalcoSuessgott 

Philipp Felwor



30 Jahre

System Engineer

System Vertrieb Alexander GmbH

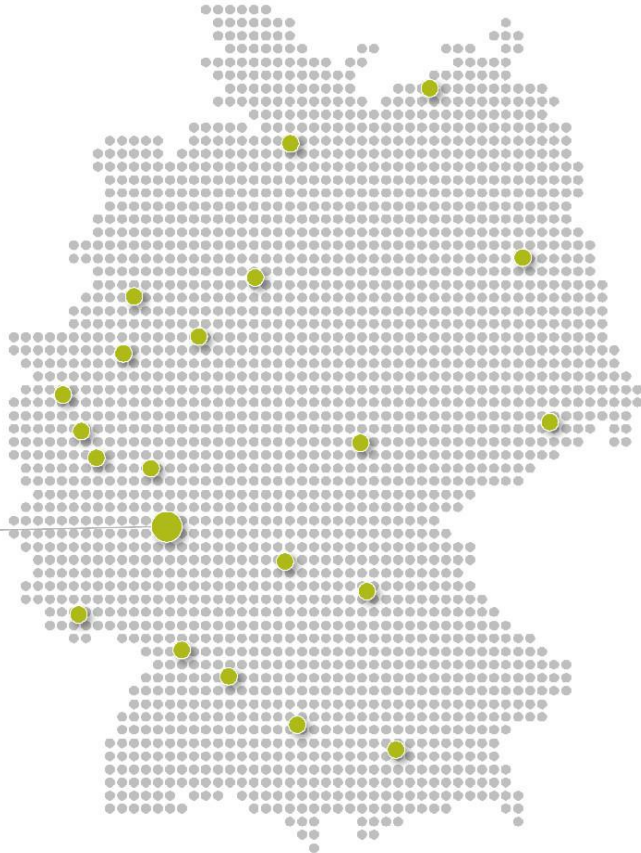
@pfelwor 

Über uns / Unternehmen

21

Lokationen

Wiesbaden



6 TOP

Branchen



Automotive



Retail



Public



Maschinen &
Anlagenbau

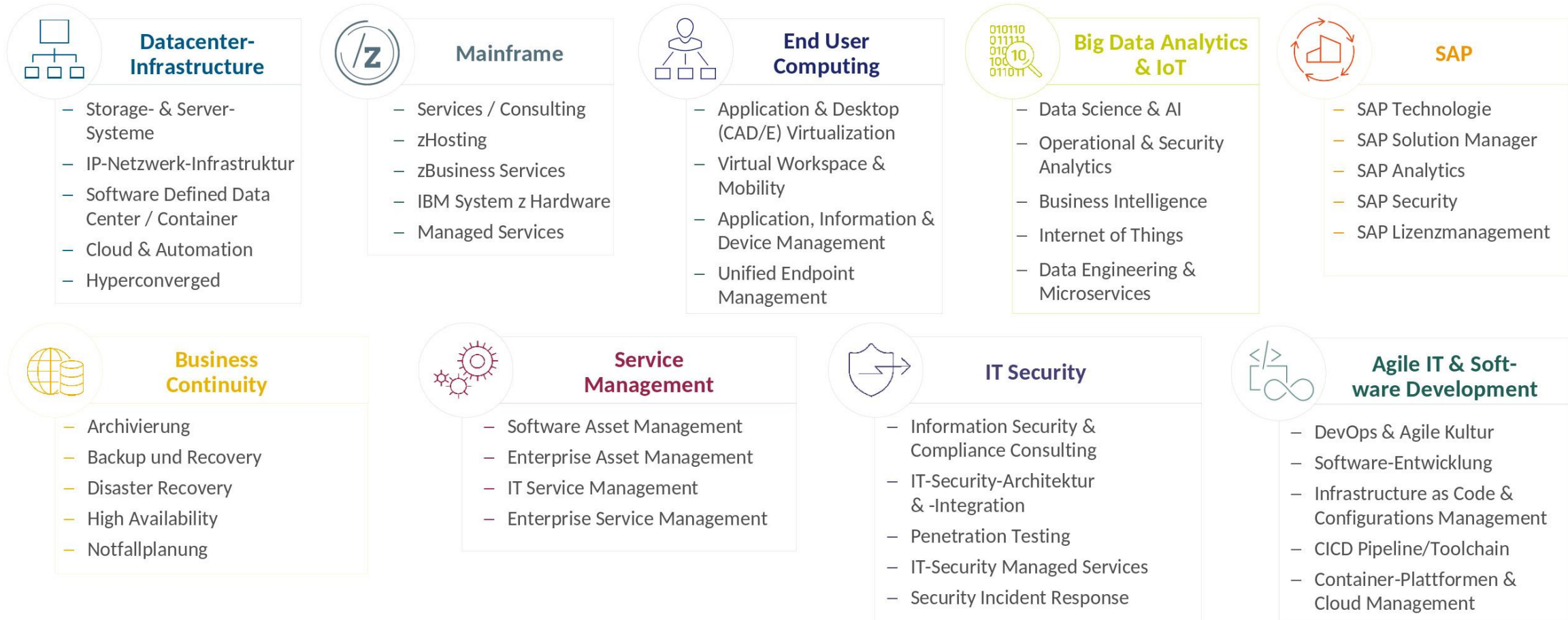


Finance &
Insurance



Telekommu-
nikation

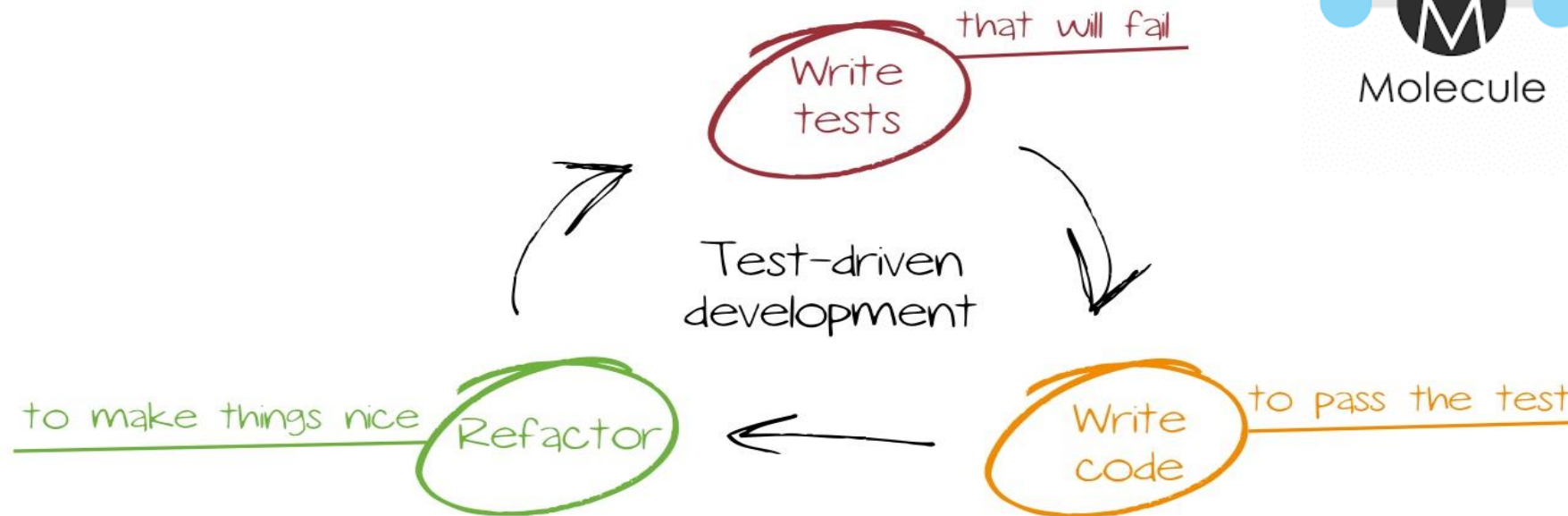
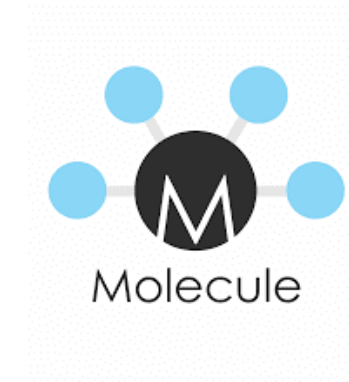
/ Produktportfolio auf einen Blick



/ Einleitung

Was ist Molecule?

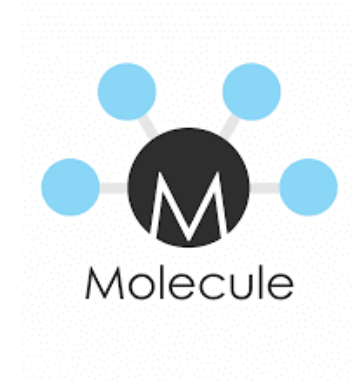
- Testframework für Ansible Rollen und Playbooks
- Ermöglicht:
 - Test driven Development (TDD)
 - Automatisiertes Testen gegen verschiedene Distributionen und Versionsstände



/ Einleitung

Was ist Molecule?

- Existiert seit Ende 2015
 - Community-Maintained als Teil des Projekts “Ansible by Red Hat”
 - in Python geschrieben



ansible-community / **molecule** Watch 123 Unstar 2.9k Fork 512

[Code](#) [Issues 40](#) [Pull requests 1](#) [Discussions](#) [Actions](#) [Wiki](#) [Security](#) [Insights](#)

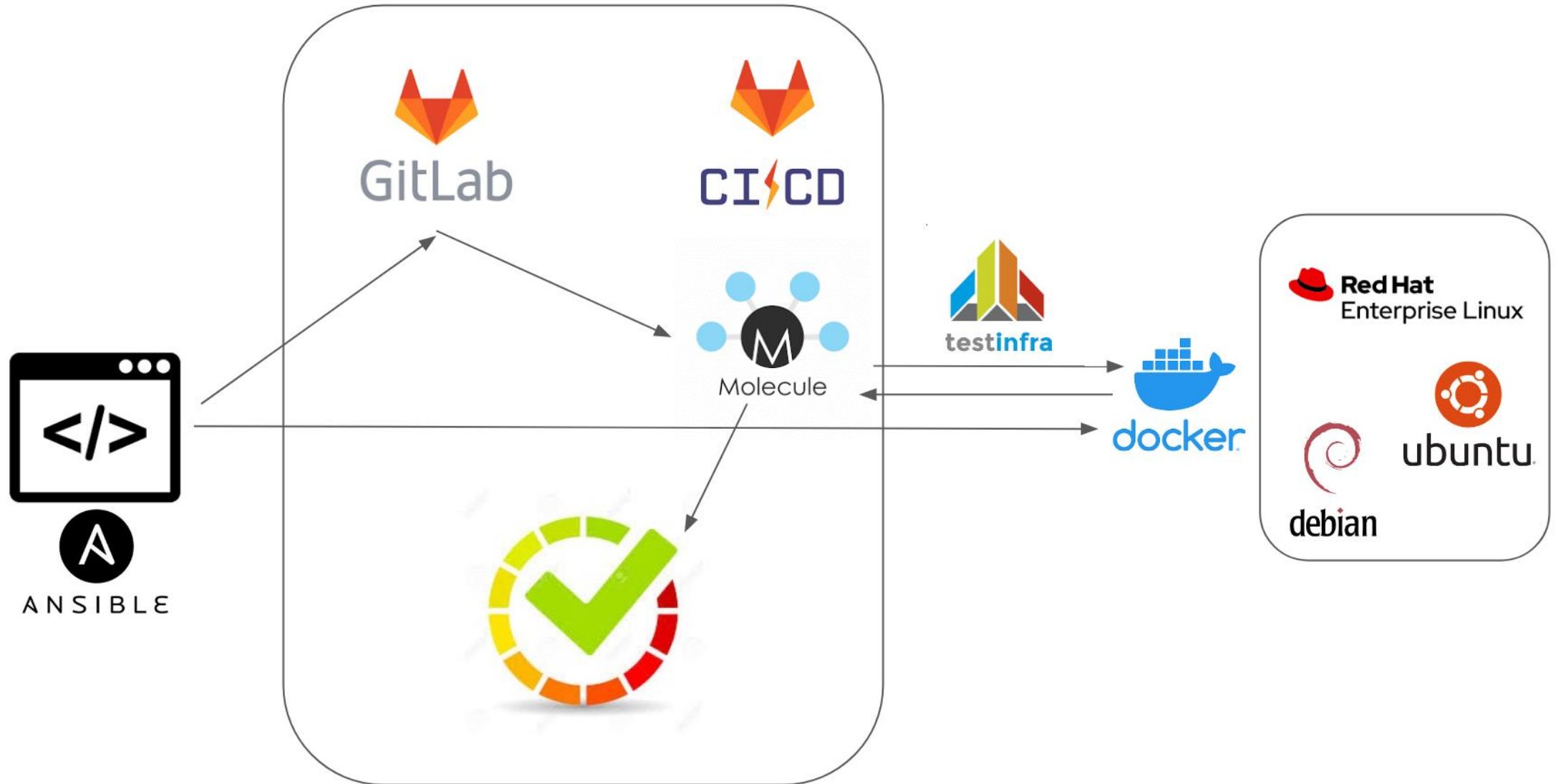
master 10 branches 172 tags Go to file Add file Code

 **klorenz** Add description of environment variables (#2915) ✓ 4ff8c20 42 minutes ago 🕒 2,241 commits

 `.config/molecule` Load molecule config from VCS is possible 8 months ago

About
Molecule aids in the development and testing of Ansible roles
molecule.readthedocs.io

/ Einleitung



Getting Started

/ Installation



```
$> python3 -m pip install "molecule[ansible]"
```

aktuelle Version: v3.1.5

Requirements:

- Python ≥ 3.6
- Ansible ≥ 2.9
 - "latest two major versions"

/ Initialisierung

```
$> molecule init role ansible-role-httpd
--> Initializing new role ansible-role-httpd...
Initialized role in $PWD/ansible-role-httpd successfully.
```

```
$> tree
.
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── molecule # molecule configuration directory
│   └── default
│       ├── converge.yml
│       ├── INSTALL.rst
│       ├── molecule.yml # main configuration file
│       └── verify.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

10 directories, 12 files
```

/ Konfiguration

```
---
dependency:
  name: galaxy
driver:
  name: docker
platforms:
  - name: instance
    image: docker.io/pycontribs/centos:7
    pre_build_image: true
provisioner:
  name: ansible
verifier:
  name: ansible
```

- **dependency**
 - Use Ansible Galaxy to resolve role dependencies
 - Default: galaxy
 - Options: ...
- **lint:**
 - lint: |
ansible-lint
yaml-lint
- **driver:**
 - The driver to delegate the task of creating instances
 - Default: docker
 - Options: podman, delegated, vagrant
- **platforms**
 - Which instances to create, name and to which group each instance belongs
- **provisioner**
 - Tool for managing the lifecycle of the instances
- **Scenario:**
 - Scenario sequence order
 - 1 scenario = N Actions (lint, syntax, idempotence)
- **verifier**
 - Verifier framework that executes the tests
 - Default: ansible
 - Options: testinfra, goss, ...

/ Workflow



```
# Run test suite (lint, cleanup, syntax, destroy, ..)
$> molecule test

# configure instances
$> molecule converge

# login to instances
$> molecule login

# destroy instances
$> molecule destroy

# run only tests
$> molecule verify
```




Hands-On

/ Szenario

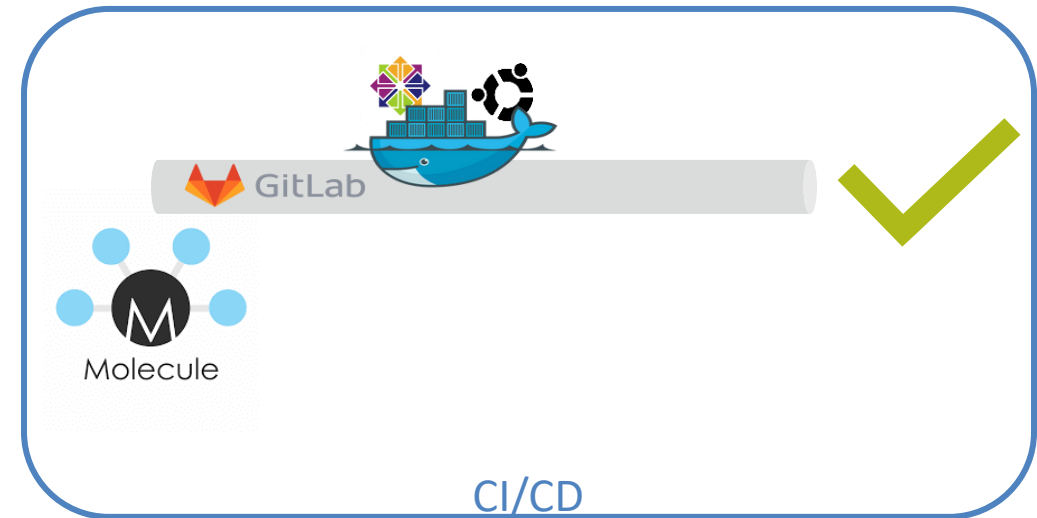
Ausgangssituation:

- ansible-role-httpd bisher getestet für CentOS7

Ziel:

- support CentOS8, Ubuntu, Debian, ..

Wie kann ich eine Ansible Rolle effizient und automatisiert für verschiedene Distributionen und Versionen maintainen?



/ molecule/default/tests/test_apache.py

Testdriven development (TDD) - first things first!

testinfra:

- Docs: <https://testinfra.readthedocs.io/>
- Pfad: molecule/default/tests/
- Benötigter Datei-Präfix: test_*
- Verfügbare Module:
 - process
 - package
 - ip
 - sysinfo
 - ...



/ tasks/main.yml

```
---
- name: apt update
  become: true
  apt:
    upgrade: yes
    update_cache: yes
    cache_valid_time: 86400
  when: ansible_os_family == 'Debian'

- name: Install apache2
  package:
    name:
      - "{{ apache_package }}"
      - curl
    state: present

- name: Copy a web page.
  copy:
    content: |
      <html>
      <head><title>Hello world!</title></head>
      <body>Hello world!</body>
      </html>
    dest: "/var/www/html/index.html"
    mode: '0600'
    notify: restart apache

- name: Ensure Apache is running and starts at boot.
  service:
    name: "{{ apache_service }}"
    state: started
    enabled: true
```

molecule/default/playbook.yml

```
---
- name: Converge
  hosts: all

  pre_tasks:
    - name: Override Apache vars for Red Hat.
      set_fact:
        apache_package: httpd
        apache_service: httpd
      when: ansible_os_family == 'RedHat'

  roles:
    - role: ansible_molecule_demo
```

/ molecule/default/molecule.yml

```
---
dependency:
  name: galaxy
driver:
  name: docker
lint: |
  set -e
  ansible-lint
  yamllint -c molecule/default/yaml-lint.yml .

platforms:
- name: "${CI_JOB_NAME:-centos7}"
  image: "geerlingguy/docker-${CI_JOB_NAME:-centos7}-ansible:latest"
  command: "${MOLECULE_DOCKER_COMMAND:-""}"
  volumes:
    - /sys/fs/cgroup:/sys/fs/cgroup:ro
  privileged: true
  pre_build_image: true

provisioner:
  name: ansible
  playbooks:
    converge: ${MOLECULE_PLAYBOOK:-playbook.yml}
scenario:
  name: default
  test_sequence:
    - lint
    - destroy
    - syntax
    - create
    - converge
    - idempotence
    - verify
    - destroy
verifier:
  name: testinfra
```

- **dependency**
 - galaxy
- **driver:**
 - docker
- **platforms**
 - aktueller \$CI-Job oder "centos7"
 - SystemD Container für Molecule testing (<https://hub.docker.com/u/geerlingguy/>)
- **provisioner**
 - ansible
 - playbook.yml
- **Scenario:**
 - ☐ lint → run yamllint and ansible-lint on YAML files, and flake8 on the Python test files
 - ☐ destroy → make sure that any infrastructure from previous tests is gone
 - ☐ dependency → (optional) download any dependencies from Ansible Galaxy
 - ☐ syntax → run ansible-playbook --syntax-check on the molecule/default/playbook.yml file
 - ☐ create → create the instances using the configured driver (docker, ec2, vagrant, etc.)
 - ☐ prepare → (optional) run a playbook to prepare the instances after create has finished
 - ☐ converge → run molecule/default/playbook.yml on the infrastructure
 - ☐ idempotence → run the playbook again to check that nothing is marked as changed
 - ☐ side_effect → (optional) run a playbook that has side effects on the instance
 - ☐ verify → run tests on the instances (testinfra is the default)
 - ☐ destroy → tear down the infrastructure and clean up
- **verifier**
 - testinfra

/ .gitlab-ci.yml

```
---
image: registry.gitlab.com/falcosuessgott/ansible_molecule_demo
services:
  - docker:dind

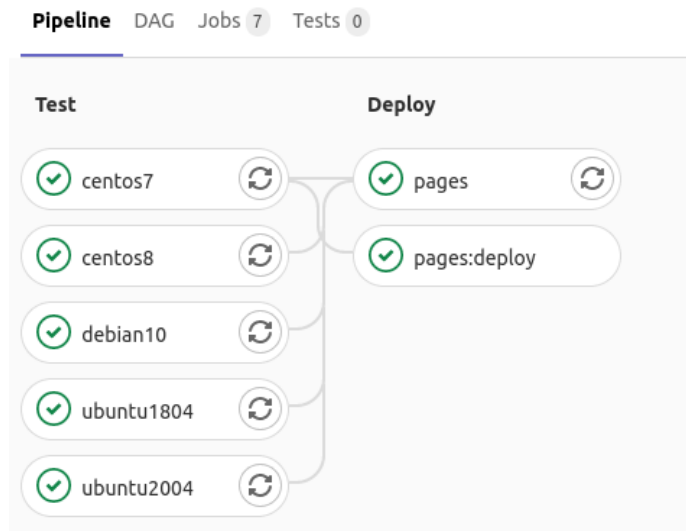
.versions:
  before_script: &versions
  - docker -v
  - ansible --version
  - molecule --version

.script:
  script: &script
  - mkdir public
  - anybadge -l $CI_JOB_NAME -v=failing -f public/$CI_JOB_NAME.svg passing=green failing=red
  - molecule test
  - anybadge -o -l $CI_JOB_NAME -v=passing -f public/$CI_JOB_NAME.svg passing=green failing=red

.artifacts:
  artifacts: &artifacts
  expire_in: 4 weeks
  when: always
  paths:
    - public/$CI_JOB_NAME.svg

centos8:
  before_script: *versions
  script: *script
  artifacts: *artifacts
  allow_failure: true
```

/ git push



ansible_molecule_demo

Ansible Role for installing and configuring apache on :

os	centos8	centos8	centos7	ubuntu1804	ubuntu2004	debian10
	passing	passing	passing	passing	passing	passing

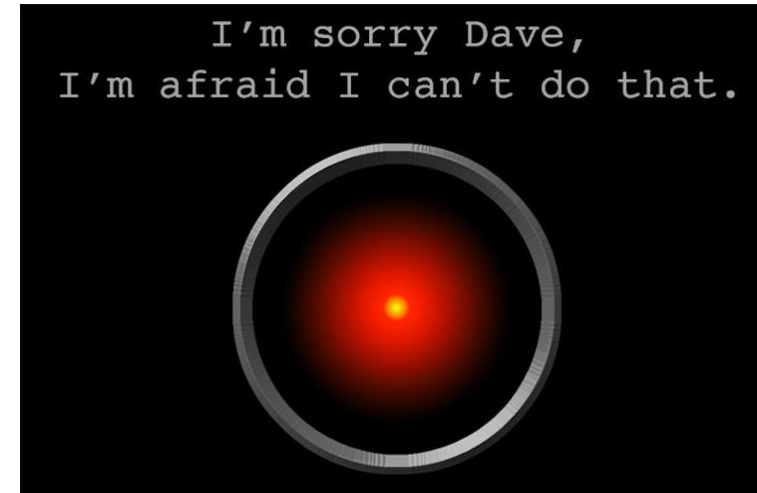
Description

Demo Project for testing ansible with molecule for ansible-anwender meeting 29.10.2020 (https://www.ansible-anwender.de/2020/10/06/__trashed/)

Tipps & Tricks

/ Tipps & Tricks

- `$USER` muss docker ausführen können
 - `usermod -aG $USER docker`
- `testinfra`:
 - `ansible_facts` nur via SSH (vagrant, delegated) verfügbar
- `molecule init scenrio $ROLE_NAME`
 - `rm $PWD/molecule/default/{create,destroy}.yml`



Resources

/ Links

- https://gitlab.com/FalcoSuessgott/ansible_molecule_demo
- <https://molecule.readthedocs.io/en/latest/>
- <https://github.com/ansible-community/molecule>
- <https://github.com/aelsabbahy/goss>
- <https://github.com/philpep/testinfra>
- <https://www.jeffgeerling.com/blog/2018/testing-your-ansible-roles-molecule>
- <https://docs.gitlab.com/ee/ci/yaml/README.html>
- <https://www.youtube.com/watch?v=gCSa8N-EvJ0>

Vielen Dank für
Ihre Aufmerksamkeit!

**Server manuell
konfigurieren**



**Server mit Skripten
konfigurieren**



**Server mit Ansible
konfigurieren**



**Serverkonfiguration mit
Ansible automatisiert
testen lassen**

