Unit Testing Ansible Roles using TDD with Molecule

eVOYAGEURS SNCF

Lionel LONKAP TSAMBA
DEVOPS ENGINEER & CLOUD ARCHITECT
@lktslionel

FOSDEM'21

The digital response to SNCF Group's challenges

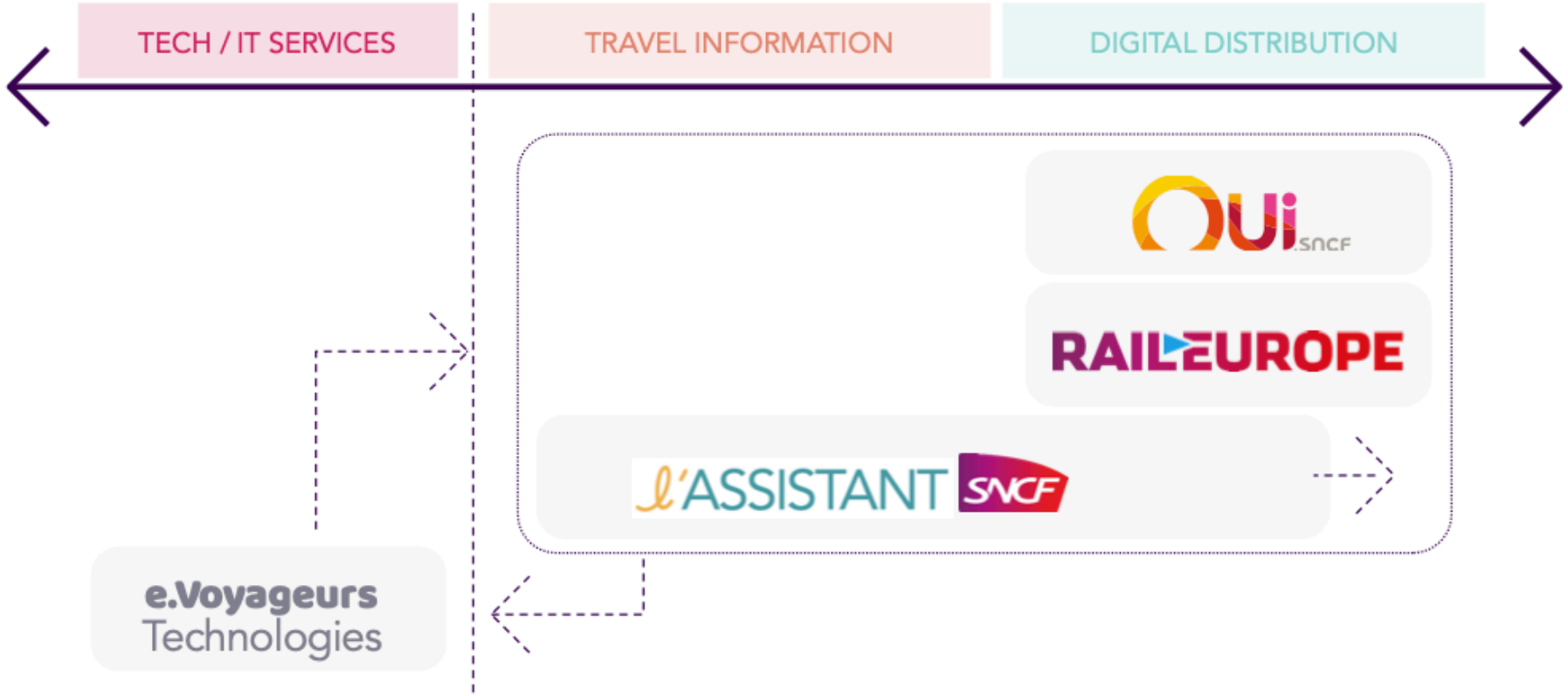
eVOYAGEURS SNCF

RAIL EUROPE · OUIGO SNCF · l'ASSISTANT SNCF

E.VOYAGEURS TECHNOLOGIES : THE GROUP'S DIGITAL FACTORY

- 1750 EMPLOYEES including +1000 DIGITAL EXPERTS
- 4,7 BILLIONS OF BUSINESS VOLUME in 2018
- 3 SITES IN FRANCE Paris, Nantes, Lille
- +30 CTRYs COVERAGE
- 110 MILLION TICKETS SOLD in 2018 98 million in France and 12 million internationally

The digital response to SNCF Group's challenges



Agenda

1. Ansible Roles
2. Testing & TDD
3. Molecule
4. Going Further
5. Q & A

eVOYAGEURS SNCF

1 Ansible Roles

FOSDEM'21

eVOYAGEURS SNCF

A role

Is a mechanism for logically breaking a playbook into re-usable components

- Defaults
- Variables
- Tasks
- Meta
- Files
- Handlers
- Templates

eVOYAGEURS SNCF

Managing roles

```
$ ansible-galaxy role --help
```

```
usage: ansible-galaxy role [-h] ROLE_ACTION ...

positional arguments:
  ROLE_ACTION
    init        Initialize new role with the base structure of a role.
    remove      Delete roles from roles_path.
    delete      Removes the role from Galaxy. It does not remove or alter the actual GitHub repository.
    list        Show the name and version of each role installed in the roles_path.
    search      Search the Galaxy database by tags, platforms, author and multiple keywords.
    import      Import a role into a galaxy server
    setup       Manage the integration between Galaxy and the given source.
    info        View more details about a specific role.
    install     Install role(s) from file(s), URL(s) or Ansible Galaxy

optional arguments:
  -h, --help    show this help message and exit
```

eVOYAGEURS SNCF

Create a new role

```
$ ansible-galaxy role init fosdem21
```

```
fosdem21
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

eVOYAGEURS SNCF

Use a role inside a playbook

```yaml
---
- hosts: all
  roles:
    - common
    - role: role1
      vars:
        r_var1: 'var1'

    - role: role2
      tags: tag1

  tasks:
    - name: Include a role
      include_role:
        name: role3
```

eVOYAGEURS SNCF

Share using Ansible Galaxy

```
$ ansible-galaxy role import\
      [-h] [-s API_SERVER]\
      [--token API_KEY]\
      […]\
      [--role-name ROLE_NAME]\
      github_user github_repo
```

Share using the roles path

```
$ ANSIBLE_ROLES_PATH=path/to/role/1 : path/to/role/2 : ...
```

eVOYAGEURS SNCF

2 Testing & TDD

FOSDEM'21

eVOYAGEURS SNCF

Unit Testing

is a way of testing a unit - the smallest piece of code that can be logically isolated in a system

Testing	Unit of Code	Isolated System
Infrastructure	Ansible Role (IaaC)	Provisioner

Infrastructure scope

Security

Usability

Compatibility

Install

Reliability

Recovery

Localization

Compliance

Ansibe role testing checklist

✓ Valid Yaml syntax

```
$ yamllint
```

✓ Valid Ansible syntax

```
$ ansible-playbook --syntax-check
$ ansible-lint
```

✓ Executable in a fresh/clean env

✓ Indempotency

✓ Rebuild an existing env from scratch

eVOYAGEURS SNCF

Applying Test-Driven Development

Write & Execute small test and watch it fail

Write enough code to make tests pass

Refactor your code for readability and maintainability

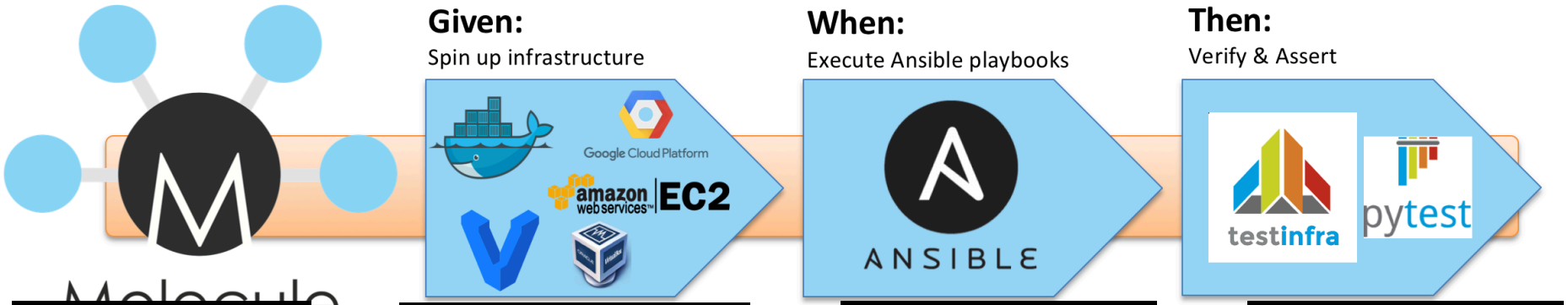
eVOYAGEURS SNCF

3

Molecule

FOSDEM'21

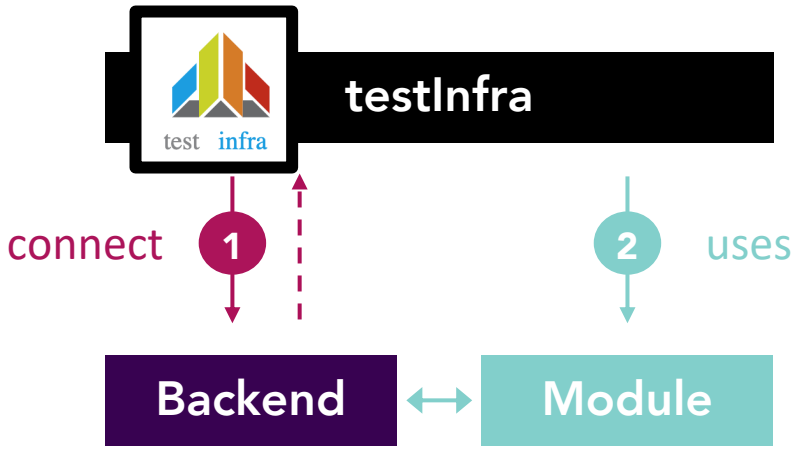
Overview



Drivers	Scenarios	Sequences	Verifiers
Docker Vagrant	default	Create Converge Verify	Testinfra Ansible

eVOYAGEURS SNCF

- ✓ Valid Yaml syntax
  ```
  $ molecule lint
  ```
- ✓ Valid Ansible syntax
  ```
  $ molecule syntax
  ```
- ✓ Executable in a fresh/clean env
  ```
  $ molecule create
  ```
- ✓ Indempotency
  ```
  $ molecule idempotence
  ```
- ✓ Rebuild an existing env from scratch
  ```
  $ molecule test
  ```



https://testinfra.readthedocs.io/en/latest/backends.html

testinfra.readthedocs.io/en/latest/modules.html

eVOYAGEURS SNCF

Example

```
import unittest
import testinfra

class Test(unittest.TestCase):
    def setUp(self):
        self.host = testinfra.get_host("paramiko://root@host")

    def test_nginx_config(self):
        self.assertEqual(self.host.run("nginx -t").rc, 0)

    def test_nginx_service(self):
        service = self.host.service("nginx")
        self.assertTrue(service.is_running)
        self.assertTrue(service.is_enabled)

if __name__ == "__main__":
    unittest.main()
```

PRACTICE

```
$ git clone https://github.com/lktslionel/FOSDEM21-Demo
```

eVOYAGEURS SNCF

eVOYAGEURS SNCF

Setup a demo env

```
$ docker run --rm\
      -v /var/run/docker.sock:/var/run/docker.sock\
      -v ${PWD}:/demo\
      -w /demo\
      -it tsklabs/ci:ansible-1.2.0  bash
```

- Docker in Docker (DinD) — `20.10.2`
- Ansible — `2.10.3`
- Molecule — `3.2`
- Pytest — `6.2.1`
- testinfra — `6.1`

eVOYAGEURS SNCF

Create a new role

```
$ molecule init role --help
```

```
Usage: molecule init role [OPTIONS] ROLE_NAME

  Initialize a new role for use with Molecule.

Options:
  --dependency-name [galaxy]      Name of dependency to initialize. (galaxy)
  -d, --driver-name [azure|containers|delegated|docker|ec2|gce|libvirt|lxd|openstack|podman|vagrant]
                                  Name of driver to initialize. (delegated)
  --lint-name [yamllint]          Name of lint to initialize. (yamllint)
  --provisioner-name [ansible]    Name of provisioner to initialize. (ansible)
  --verifier-name [ansible|testinfra]
                                  Name of verifier to initialize. (ansible)
  --help                          Show this message and exit.
```

eVOYAGEURS SNCF

Create a new role

```
$ molecule init role --driver-name docker --verifier-name testinfra ansible-role-fosdem21
```

```
.
├── README.md
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── molecule
│   └── default
│       ├── converge.yml
│       ├── molecule.yml
│       └── tests
│           ├── conftest.py
│           └── test_default.py
```

```
.
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

eVOYAGEURS SNCF

Applying TDD

Write our test case using **TestInfra** and run **molecule**

Write our **role code** to make the test pass

Improve the code and re-run **molecule** to validate our changes

eVOYAGEURS SNCF

Benefits

- Ansible-native, stable, well establish in the community
- Written in Python
- Support python-based verifier : **testinfra**
- Ansible toolset

```
$ docker pull quay.io/ansible/toolset
```

Downsides

- No windows support
- Only supports Ansible provisioning

Examples

- **molecule-demo**
  https://github.com/ehashman/molecule-demo
- **ceph-ansible**
  https://github.com/ceph/ceph-ansible
- **Securedrop**
  https://github.com/freedomofpress/securedrop

eVOYAGEURS SNCF

4 Going further

FOSDEM'21

eVOYAGEURS SNCF

Testing Ansible with Ansible

Benefits

- Flexible and powerful
- Leveraging your Ansible expertise

Downsides

- Assertions in production code can cause unexpected failures
- Write your own provisioner

eVOYAGEURS SNCF

Testing Ansible with Ansible-test

Benefits

- Simple tool with and powerful
- Written in Python
- Use Docker as a provisioner

Downsides

- No support for verifiers — simple run & check result
- Not actively maintained

Testing Ansible with Test Kitchen

Benefits

- Large community
- Well maintained and robust project
- Supports Windows testing

Downsides

- Need to learn Ruby DSL
- Verifiers are Ruby or Bash based

Try BDD with Molecule

Applying Behavior Driven Development using:

- behave
- pytest-bdd

eVOYAGEURS SNCF

Q & A

FOSDEM'21