



ANSIBLE COMPATIBILITY TESTING

WHAT TO EXPECT

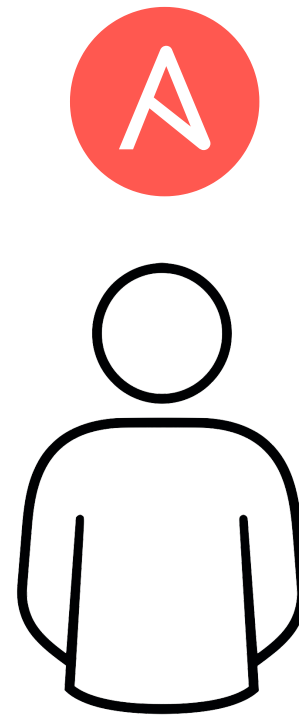
- // ansible - a short story
- // functional testing with molecule
- // compatibility testing - what's that?
- // compatibility testing for ansible

PRESS "S" FOR ADDITIONAL INFORMATION

The presentation is enriched with speaker notes
that can be revealed right in your browser

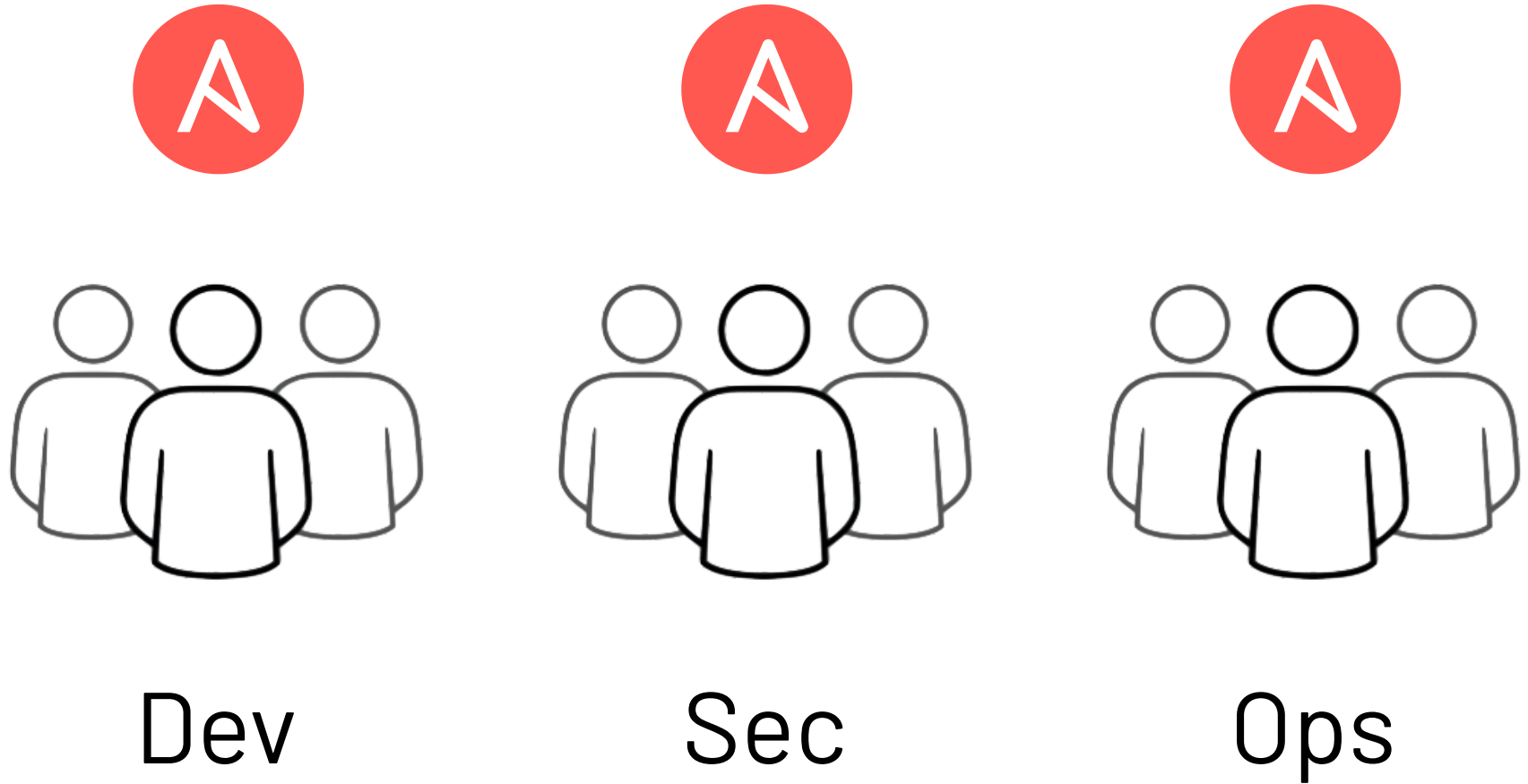
ANSIBLE

a short story

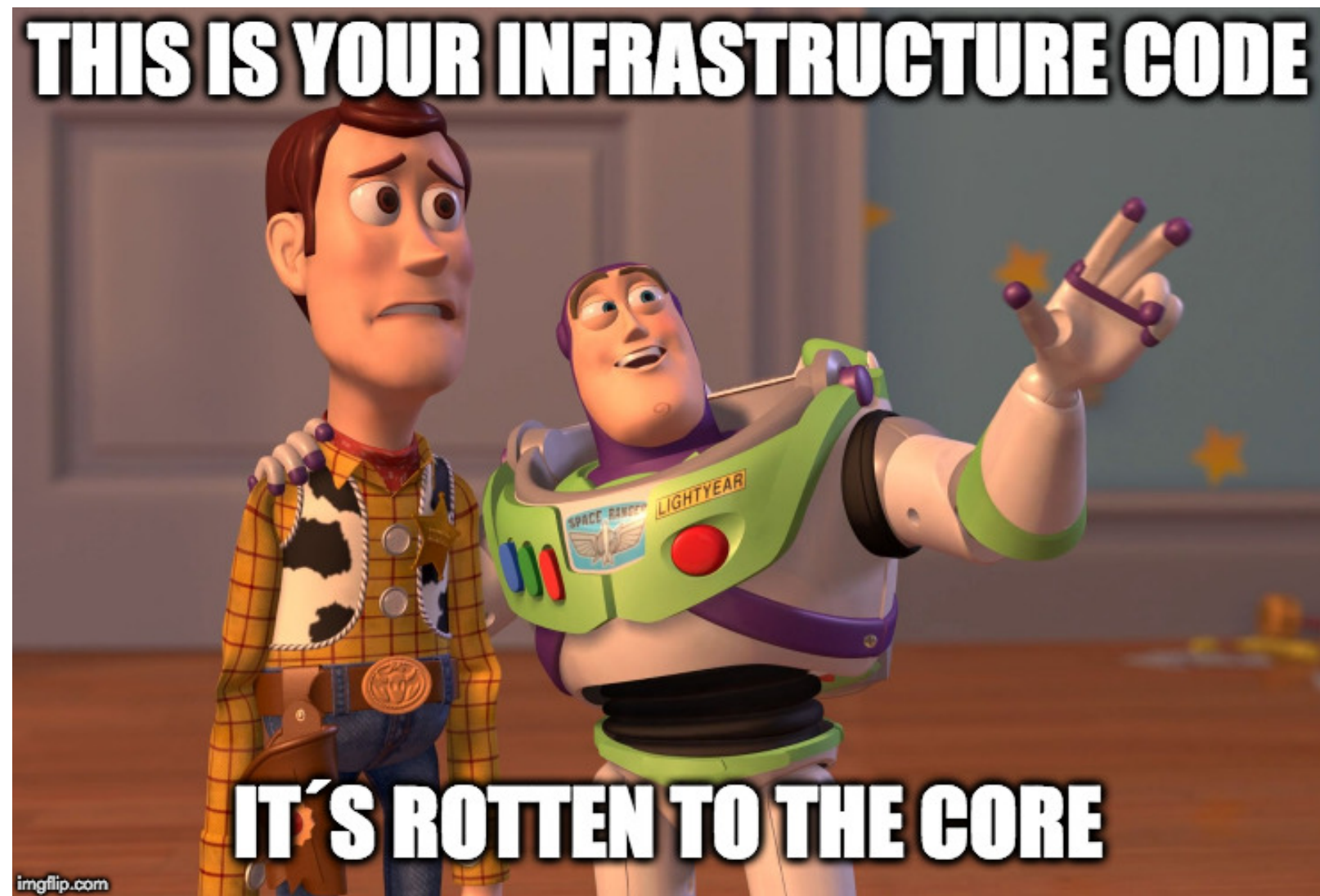


automation happens when you meet a problem
you never want to solve again

Congrats!
you established a new tool



not all automation is created equal



RULES OF THE GAME

write ansible roles

// distinct purpose

// configurable

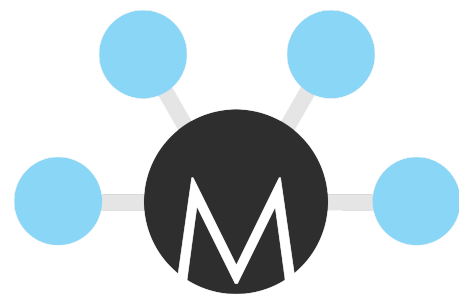
// test it

// reusable

// share it with the community

FUNCTIONAL TESTING

// check whether your code is working as expected



MOLECULE

Molecule encourages consistently
developed roles that are

// well-written

// easily understood

// maintainable

Molecule supports testing with multiple

- // instances
- // operating systems and distributions
- // virtualization providers
- // test frameworks
- // testing scenarios

simple tests with assertions in python

```
# Example role
- name: create /tmp/hello-world.txt copy:
  dest: "/tmp/hello-world.txt" content: "hello world" tags:
- example
- example.hello-world
```

```
# Tests
def test_file_exists(host):
    f = host.file('/tmp/hello-world.txt')
    assert f.exists

def test_file_content(host):
    content = host.file('/tmp/hello-world.txt').content
    assert b'hello world' in content
```


COMPATIBILITY TESTING

// check whether your code is capable of running on different

- hardware
- applications
- versions
- operating systems
- networks

version compatibility testing

// backward compatibility

// forward compatibility

PYTHON 2 IS OFFICIALLY END OF LIFE...

ARE YOU READY?

HOW TO SURVIVE

ansible + molecule **x** python 2.7 – production

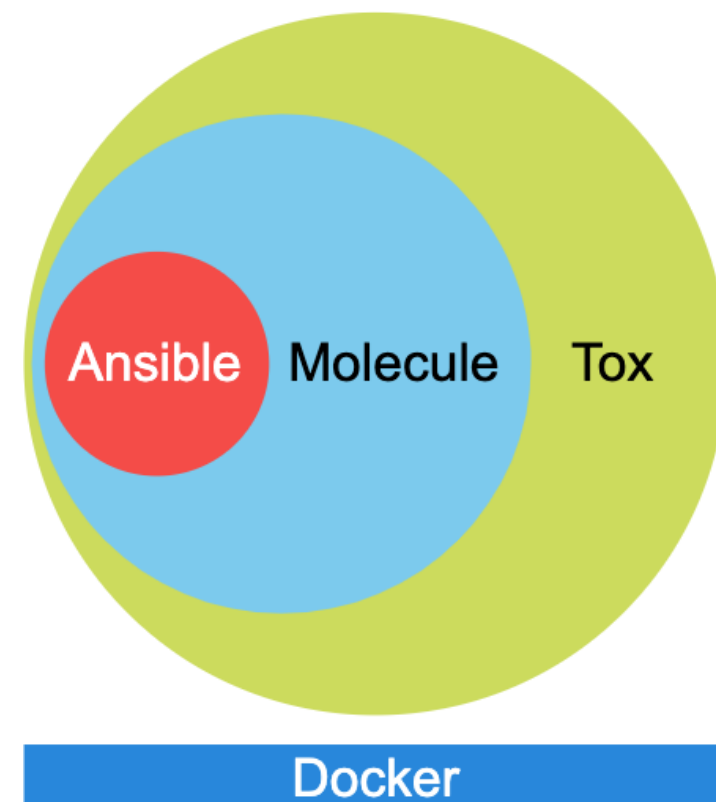
ansible + molecule x python 2.5 – backward compatibility
2.7 – production
3.8 – forward compatibility



// virtualenv management and test command line tool

// vision: standardize testing in Python

Survival Strategy



- // automation via ansible
- // functional tests via molecule
- // multiple python versions via tox
- // container as target environment

Why stop here?

// add various ansible versions

- can I safely update to a newer ansible version?
- which ansible version do I need at the very least?

// add multiple linux flavors

- verify that your role can run on RHEL 7 and RHEL 8
- Debian 9 and 10, Ubuntu 16.04 and 18.04, ...

Well, I couldn't stop myself...

// ansible - latest, 2.9, 2.8

// OS

- alpine - latest, 3.10, 3.9, 3.8

- centos - latest, 8, 7

- clearlinux - latest

- debian - latest, 10, 9, 8

- fedora - latest, 31, 30, 29, 28

- opensuse - latest, 15,

- ubuntu - latest, 20.04, 19.04, 18.04, 16.04

// python - 3.x, 3.8, 3.7, 3.6

COMPLETION TIME

7 hrs 27 min 2 sec

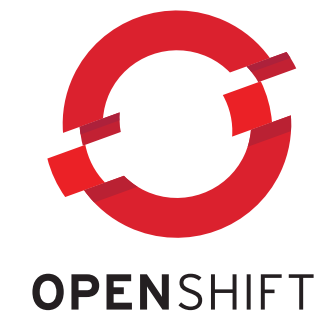
RECAP

- // you meet a problem you never want to solve again
- // you establish a new tool - congrats!
- // it gets adopted
- // jointly define the rules of the game
 - create reusable ansible roles
 - functional testing with molecule
 - version compatibility testing
 - choose purposeful compatibility scenarios

GEPARD WHO?

THAT'S US

- // office: vienna / linz
- // size: ~ 40 people
- // avg age: 31
- // we do what we love..
 - custom software solution
 - cloud transformation
 - devOps automation



..and we are extremely good at it

THANK YOU

// slides: <https://gepardec.github.io/presentations/ansible-compatibility-testing>

// code: <https://github.com/gepardec/ansible-role-example>

// author: Clemens Kaserer

