

Разработка и тестирование Ansible ролей и плейбуков

План

- Vagrant и локальная разработка ролей
- Интеграционное тестирование конфигурации и инструменты
- Основы работы с Molecule
- Тестовые фреймворки
- Примеры работы с TestInfra и Goss

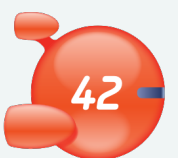


Vagrant

- Инструмент для декларативного описания сред разработки
- Описание инфраструктуры содержится в Vagrantfile
- Большой выбор образов VM (boxes)
- Поддержка большого количества провайдеров и провижинеров

Пример Vagrantfile

```
Vagrant.configure("2") do |config|  
  config.vm.define "ubuntu16" do |ub|  
    ub.vm.box = "ubuntu/xenial64"  
  end  
end
```



\$ vagrant up

Bringing machine 'ubuntu16' up with 'virtualbox' provider...

==> **ubuntu16: Importing base box 'ubuntu/xenial64'...**

==> **ubuntu16: Matching MAC address for NAT networking...**

==> **ubuntu16: Checking if box 'ubuntu/xenial64' is up to date...**

==> **ubuntu16: A newer version of the box 'ubuntu/xenial64' is available! You currently**

==> **ubuntu16: have version '20170620.0.0'. The latest is version '20170922.0.0**

==> **ubuntu16: `vagrant box update` to update.**

==> **ubuntu16: Mounting shared folders...**

ubuntu16: /vagrant => /Users/artemkin/vagrant/ubuntu16

\$ vagrant status

Current machine states:

ubuntu16 running (virtualbox)

Shared folder

```
~/vagrant/ubuntu16 $ echo "test" > file.txt  
~/vagrant/ubuntu16 $ vagrant ssh ubuntu16
```

Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-81-generic x86_64)

- * Documentation: <https://help.ubuntu.com>
- * Management: <https://landscape.canonical.com>
- * Support: <https://ubuntu.com/advantage>

```
ubuntu@ubuntu-xenial:~$ cat /vagrant/file.txt  
test
```

Провижининг

```
Vagrant.configure("2") do |config|

  config.vm.define "ubuntu16" do |ubl|
    ubl.vm.box = "ubuntu/xenial64"
    ubl.vm.network :private_network, ip: "192.168.66.66"
    ubl.vm.hostname = "mongohost"
  end

  config.vm.provider :virtualbox do |v|
    v.memory = 512
  end

  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "db.yml"
  end
end
```

Убедимся, что установлен Python 2.X

```
- name: Converge
  hosts: all
  gather_facts: false
  become: true
  vars:
    mongo_bind_ip: 0.0.0.0
```

pre_tasks:

```
- name: Install python for Ansible
  raw: test -e /usr/bin/python || (apt -y update && apt install -y python-
minimal)
  changed_when: false
```

roles:

- db

Проверка плейбука

```
$ vagrant provision
```

```
==> ubuntu16: Running provisioner: ansible...  
ubuntu16: Running ansible-playbook...
```

```
TASK [db : Install mongodb package] *****  
changed: [ubuntu16]
```

```
TASK [db : Configure service supervisor] *****  
changed: [ubuntu16]
```

```
TASK [db : Change mongo config file] *****  
changed: [ubuntu16]
```

```
RUNNING HANDLER [db : restart mongod] *****  
changed: [ubuntu16]
```

```
PLAY RECAP *****  
ubuntu16 : ok=8 changed=6 unreachable=0 failed=0
```

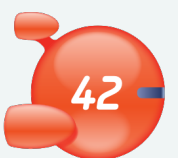
Создание VM в GCE

```
Vagrant.configure("2") do |config|  
  config.vm.box = "google/gce"  
  
  config.vm.provider :google do |google, override|  
    google.google_project_id = "YOUR_GOOGLE_CLOUD_PROJECT_ID"  
    google.google_client_email = "YOUR_SERVICE_ACCOUNT_EMAIL_ADDRESS"  
    google.google_json_key_location = "/path/to/your/private-key.json"  
  
    override.ssh.username = "USERNAME"  
    override.ssh.private_key_path = "~/.ssh/id_rsa"  
  end  
end
```

Тестирование конфигурации

Зачем тестировать конфигурацию?

- ...
- ...
- ...



Инструменты тестирования

- Test Kitchen
- Molecule
- Тестовые фреймворки
- Линтеры (cookstyle, ansible-lint, tflint)

Molecule

- Инструмент для разработки и тестирования Ansible ролей
- Отвечает за создание инфраструктуры для тестирования, применение конфигурации и запуск тестов
- Поддержка различных драйверов для провижининга: gce, aws, vagrant, docker
- Поддержка различных тестовых фреймворков: TestInfra, goss

Molecule

```
$ molecule --help
```

Options:

| | |
|-----------------------------------|--|
| <code>--debug / --no-debug</code> | Enable or disable debug mode. Default is disabled. |
| <code>--version</code> | Show the version and exit. |
| <code>--help</code> | Show this message and exit. |

Commands:

| | |
|--------------------------|---|
| <code>check</code> | Use the provisioner to perform a Dry-Run... |
| <code>converge</code> | Use the provisioner to configure instances... |
| <code>create</code> | Start instances. |
| <code>destroy</code> | Destroy instances. |
| <code>idempotence</code> | Use the provisioner to configure the... |
| <code>init</code> | Initialize a new role or scenario. |
| <code>lint</code> | Lint the role. |
| <code>list</code> | Lists status of instances. |
| <code>login</code> | Log in to one instance. |
| <code>syntax</code> | Use the provisioner to syntax check the role. |
| <code>test</code> | Test (destroy, create, converge, lint,... |
| <code>verify</code> | Run automated tests against instances. |

molecule init

- Позволяет создать заготовку роли с тестами
- Использует ansible-galaxy init
- По умолчанию используется docker драйвер и Testinfra

пример

```
$ molecule init role --role-name db --driver-name vagrant
```

для готовой роли

```
$ molecule init scenario --scenario-name default -r db -d vagrant
```


molecule.yml

```
---  
driver:  
  name: vagrant  
  provider:  
    name: virtualbox  
platforms:  
  - name: instance  
    box: ubuntu/xenial64  
provisioner:  
  name: ansible  
  lint:  
    name: ansible-lint  
verifier:  
  name: testinfra  
  lint:  
    name: flake8
```

Тестовые фреймворки

- ServerSpec
- InSpec
- Testinfra
- Goss

Testinfra

- Тестовый фреймворк для инфраструктуры
- Поддержка основных CM инструментов: Chef, Ansible, Puppet, Salt
- Содержит множество готовых модулей для проверки настроек конфигурации
- Реализован на Python

Testinfra пример

db/molecule/default/tests/test_default.py

```
def test_mongo_running_and_enabled(host):
    mongo = host.service("mongod")
    assert mongo.is_running
    assert mongo.is_enabled

def test_config_file(File):
    config_file = File('/etc/mongod.conf')
    assert config_file.contains('bindIp: 0.0.0.0')
    assert config_file.is_file

def test_socket_listening(Socket):
    socket = Socket('tcp://0.0.0.0:27017')
    assert socket.is_listening
```



```
$ molecule create
```

```
--> Test matrix
```

```
└─ default
  └─ create
--> Scenario: 'default'
--> Action: 'create'
```

```
PLAY [Create] *****
```

```
TASK [Create molecule instance(s)] *****
changed: [localhost] => (item={'box': u'ubuntu/xenial64', 'name': u'instance'})
```

```
TASK [Populate instance config dict] *****
ok: [localhost] => (item={u'IdentityFile': u'/Users/artemkin/mol/ansible/roles/db/
molecule/default/.molecule/.vagrant/machines/instance/virtualbox/private_key',
'_ansible_parsed': True, '_ansible_item_result': True, '_ansible_no_log': False, 'item':
{'box': u'ubuntu/xenial64', 'name': u'instance'}, u'LogLevel': u'FATAL', u'HostName':
u'127.0.0.1', u'Host': u'instance', u'PasswordAuthentication': u'no', u'IdentitiesOnly':
u'yes', u'UserKnownHostsFile': u'/dev/null', u'User': u'ubuntu', u'changed': True,
u'invocation': {u'module_args': {u'platform_box_url': None, u'provider_cpus': 2,
u'provider_raw_config_args': None, u'platform_box': u'ubuntu/xenial64',
u'platform_box_version': None, u'instance_interfaces': [], u'instance_name': u'instance',
u'state': u'up', u'provider_memory': 512, u'instance_raw_config_args': None,
u'provider_options': {}, u'provider_name': u'virtualbox', u'force_stop': False}},
u'StrictHostKeyChecking': u'no', u'Port': u'2222'})
```

\$ molecule converge

--> Test matrix

```
└─ default
   └─ create
      └─ converge
```

--> Scenario: 'default'

--> Action: 'create'

Skipping, instances already created.

--> Scenario: 'default'

--> Action: 'converge'

TASK [db : Configure service supervisor] *****
changed: [instance]

TASK [db : Show info about the env this host belongs to] *****
ok: [instance] => {
 "msg": "This host is in local environment!!!"
}

TASK [db : Change mongo config file] *****
changed: [instance]

RUNNING HANDLER [db : restart mongod] *****
changed: [instance]

PLAY RECAP *****
instance : ok=9 changed=6 unreachable=0 failed=0

Первый тест прошел, два зафейлилось

\$ molecule verify

collected 3 items

tests/test_default.py .FF

```
===== FAILURES =====
File = <class 'testinfra.modules.base.GNUFile'>

    def test_config_file(File):
        config_file = File('/etc/mongod.conf')
>       assert config_file.contains('bindIp: 0.0.0.0')
E       AssertionError: assert False
E       + where False = <bound method GNUFile.contains of <file /etc/
mongod.conf>>('bindIp: 0.0.0.0')
E       + where <bound method GNUFile.contains of <file /etc/mongod.conf>> = <file
etc/mongod.conf>.contains
tests/test_default.py:15: AssertionError
_____ test_socket_listening[ansible://instance] _____

Socket = <class 'testinfra.modules.base.LinuxSocket'>

    def test_socket_listening(Socket):
        socket = Socket('tcp://0.0.0.0:27017')
>       assert socket.is_listening
E       assert False
E       + where False = <socket tcp://0.0.0.0:27017>.is_listening
tests/test_default.py:20: AssertionError
===== 2 failed, 1 passed, 4 warnings in 2.70 seconds =====
```

Изменим переменную

db/molecule/default/playbook.yml

```
- name: Converge  
  hosts: all  
  become: true
```

```
vars:  
  mongo_bind_ip: 0.0.0.0
```

```
roles:  
  - role: db
```



Три точки = три успешных теста

```
$ molecule verify
```

```
===== test session starts =====
```

```
platform darwin -- Python 2.7.10, pytest-3.2.2, py-1.4.34, pluggy-0.4.0  
rootdir: /Users/artemkin/mol/ansible/roles/db/molecule/default, inifile:  
plugins: testinfra-1.6.3
```

```
collected 3 items
```

```
tests/test_default.py ...
```

```
===== 3 passed, 4 warnings in 3.02 seconds
```

```
=====
```

```
Verifier completed successfully.
```

Проверка на идемпотентность

```
$ molecule idempotence
```

```
--> Test matrix
```

```
└─ default  
   └─ idempotence
```

```
--> Scenario: 'default'
```

```
--> Action: 'idempotence'
```

```
Idempotence completed successfully
```

\$ molecule lint

...

57:42 **error** no new line character at the end of file (new-line-at-end-of-file)

/Users/artemkin/mol/ansible/roles/db/tasks/configure.yml
1:1 **warning** missing document start "---" (document-start)

/Users/artemkin/mol/ansible/roles/db/tasks/install.yml
3:1 **warning** missing document start "---" (document-start)

/Users/artemkin/mol/ansible/roles/db/tests/test.yml
5:9 **error** no new line character at the end of file (new-line-at-end-of-file)

/Users/artemkin/mol/ansible/roles/db/vars/main.yml
2:19 **error** no new line character at the end of file (new-line-at-end-of-file)

Полная проверка с нуля

```
$ molecule test
```

```
--> Test matrix
```

```
└─ default
    ├── destroy
    ├── dependency
    ├── syntax
    ├── create
    ├── converge
    ├── idempotence
    ├── lint
    ├── side_effect
    ├── verify
    └── destroy
```

```
--> Scenario: 'default'
```

```
--> Action: 'destroy'
```

Goss

- Инструмент для проверки настроек конфигурации
- Быстрый
- Возможность генерации тестов на основе существующих настроек

Пример

```
---
service:
  mongod:
    running: true
    enabled: true

file:
  /etc/mongod.conf:
    contains: ['bindIp: 0.0.0.0']
    exists: true

port:
  tcp:27017:
    listening: true
  ip:
    - 0.0.0.0
```

Вопросы по проекту infra