

Тема

Ansible

Разработка ролей



Константин Брюханов

Lead DevSecOps **ВТБ**

- ❑ 7 лет администрирования, из них 3 в DevOps
- ❑ Получаю PhD с диссертацией о DevOps
- ❑ Руководитель и лектор курсов
- ❑ Ведущий подкастов об IT и образовании
- ❑ Несу DevOps в массы



@deusops

Содержание урока

- ❑ Знакомство с понятием Ansible-роли
- ❑ Учимся использовать переменные
- ❑ Работаем с Ansible-galaxy
- ❑ Знакомство с Molecule
- ❑ Пишем плейбук и роль

Цели вебинара

После занятия вы сможете

Писать атомарные
модульные сущности
Ansible



Параметризовать
плейбуки и роли
через переменные

Смысл

Зачем вам это уметь?



Вынесение
повторяемых задач в
отдельные модули



Возможность
тестировать
отдельные куски
инфраструктурного
кода

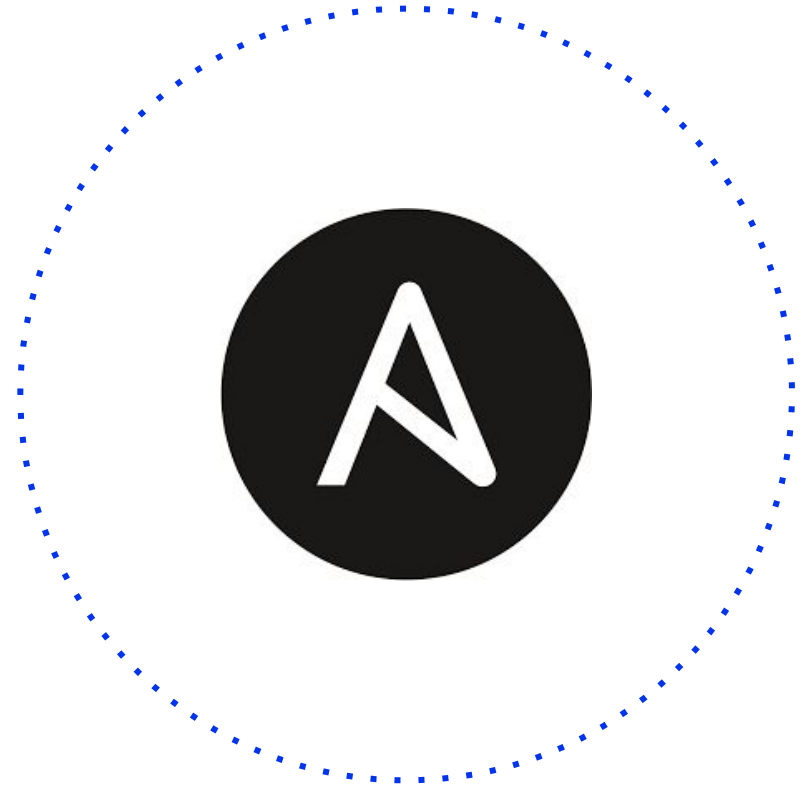


Переиспользование
разработок на всех
проектах

Ansible-роли

Что такое Ansible-роль

- ★ Отдельно вынесенные задачи. Инкапсуляция и абстракция
- ★ Атомарная сущность
- ★ Имеют свои переменные, шаблоны, файлы
- ★ Могут (должны) быть покрыты тестами



Что такое Ansible-роль

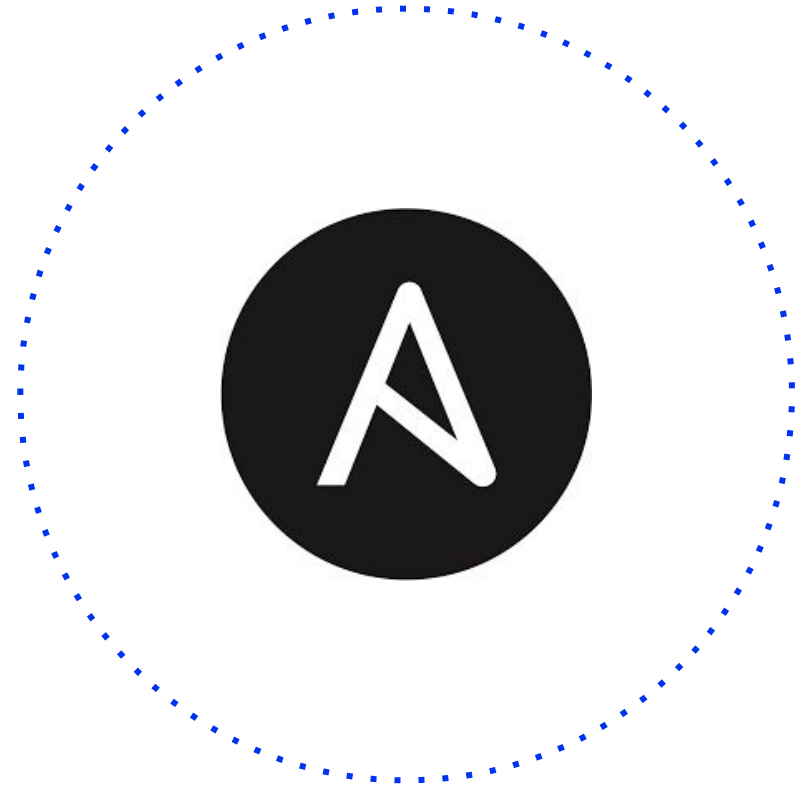
- ★ Отдельно вынесенные задачи. Инкапсуляция и абстракция
- ★ Атомарная сущность
- ★ Имеют свои переменные, шаблоны, файлы
- ★ Могут (должны) быть покрыты тестами



Просто писать и читать

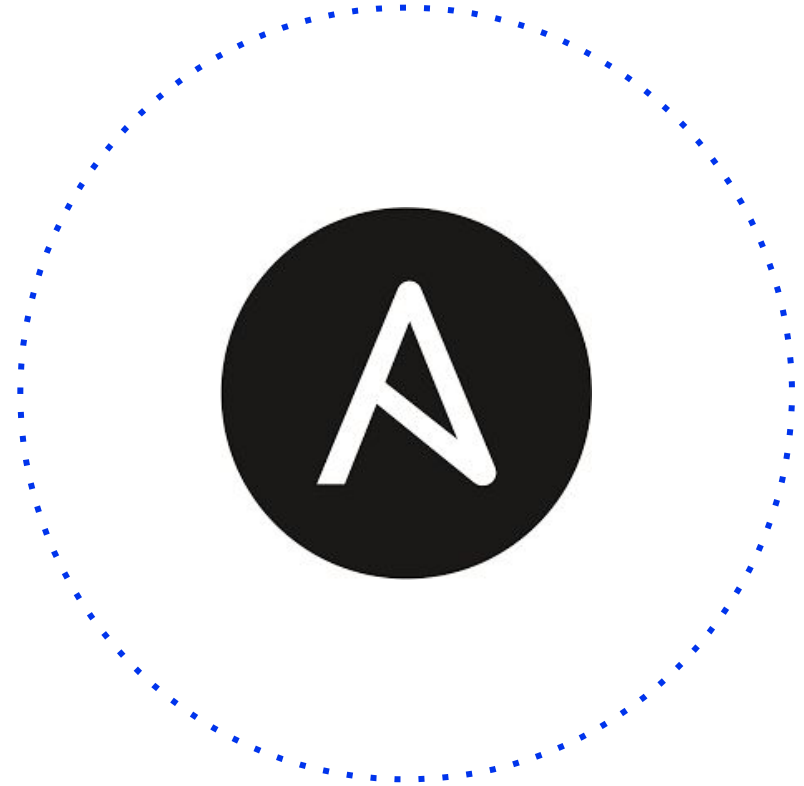


Просто отслеживать



Почему не плейбук

- ★ Быстро разрастается в размере
- ★ Быстро разрастаются в количествах (копипаст)
- ★ Не версионизируются
- ★ Не подходят как формат распространения и переиспользования кода: нет версий, зависимостей, метаданных, зато много хардкода



Структура

```
$ tree example-role
example-role
├── README.md
├── defaults
│   └── main.yml # <- Переменные и значения по умолчанию
├── files
├── handlers
│   └── main.yml # <-- Обработчики (aka хэндлеры)
├── meta
│   └── main.yml # <-- Информация о роли и зависимостях
├── tasks
│   └── main.yml # <-- Основные задачи в роли
├── templates
│   └── mongod.conf.j2 # <-- Шаблоны конфигурации
├── tests
│   ├── inventory # <-- Сценарии и данные для тестирования
│   └── test.yml
└── vars
    └── main.yml # <-- Внутренние переменные роли
```

Структура

```
example-role
├── README.md
├── defaults
│   └── main.yml
...
├── tasks
│   ├── repository.yml      # example-role/tasks/main.yml
│   ├── install.yml        - include: repository.yml
│   ├── configure.yml      - include: install.yml
│   ├── update.yml         - include: configure.yml
│   └── main.yml
...
└── meta
    └── main.yml
```

Структура

```
example-role
├── README.md
├── defaults
│   └── main.yml
...
├── tasks
│   ├── linux.yml
│   ├── windows.yml
│   └── main.yml
...
└── meta
    └── main.yml
```

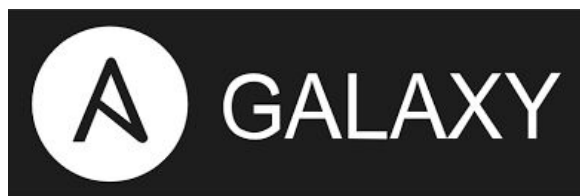
```
# example-role/tasks/main.yml
- include: linux.yml
  when: ansible_os_family == 'RedHat'
- include: windows.yml
  when: ansible_os_family == 'Windows'
```

Galaxy

Ansible Galaxy

Что такое

Облачное хранилище ролей



Основные плюсы

- ★ Не надо хранить роли рядом
- ★ Можно версионировать роли
- ★ Каждая роль - отдельный продукт
- ★ Получать гарантированный результат

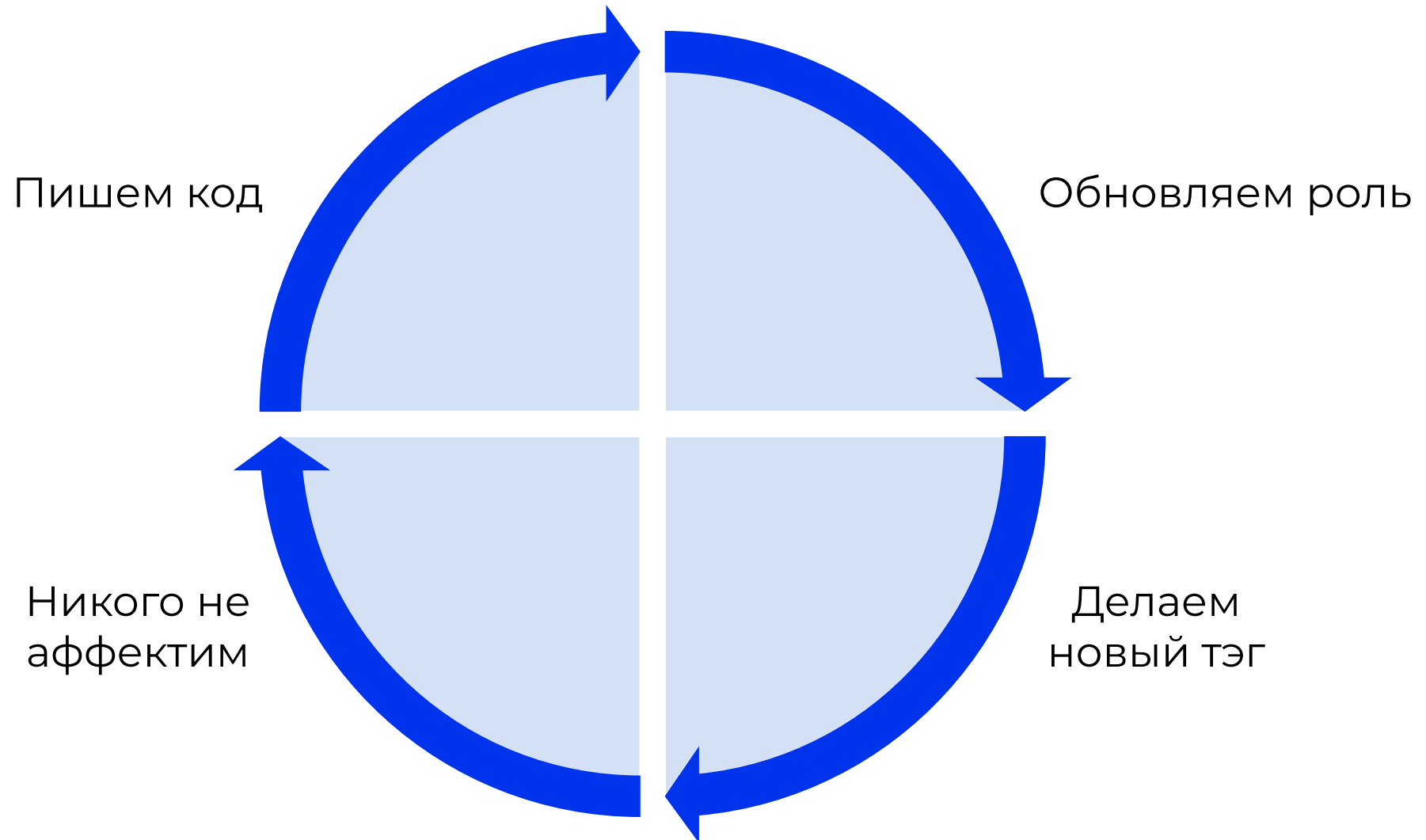
Ansible Galaxy

```
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-system-users.git
  version: origin/master
  name: system-users
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-nodejs.git
  version: origin/v.1.2.0
  name: nodejs
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-nginx.git
  version: 2.3.9
  name: nginx
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-timezone.git
  version: origin/master
  name: timezone
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-locales.git
  version: origin/master
  name: locales
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-redis.git
  version: develop
  name: redis
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-postgresql.git
  version: origin/feature/add-balancing
  name: postgresql
- src: git+ssh://git@gitlab.com/ansible-core/ansible-role-cryptopro.git
  version: v1.0.2
  name: cryptopro
```

version:

В version может быть ветка, тэг или коммит, который необходимо взять

Жизненный цикл роли



Molecule

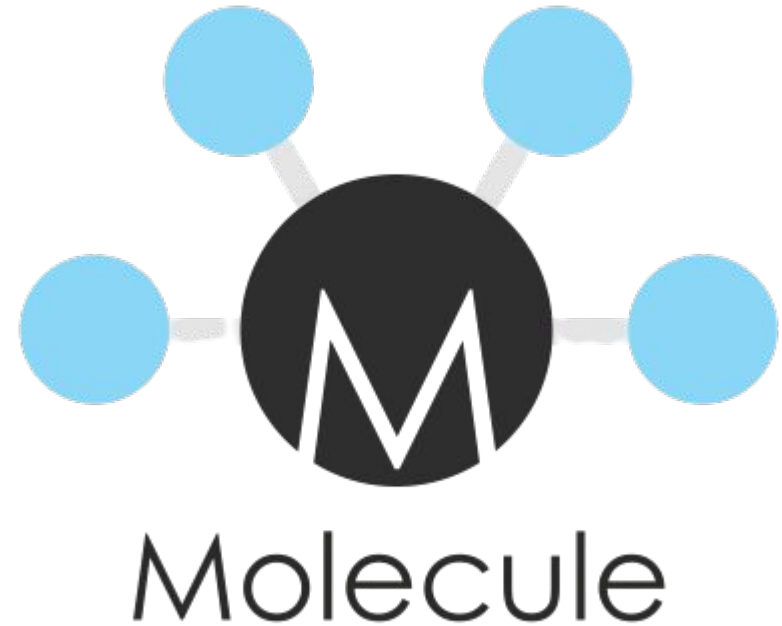
Molecule

Что такое

- ★ Фреймворк тестирования ansible-ролей

Основные плюсы

- ★ Проверять синтаксиса ролей
- ★ Имитация раскатывания роли
- ★ Проверка линтеров
- ★ Умеет работать с ansible galaxy



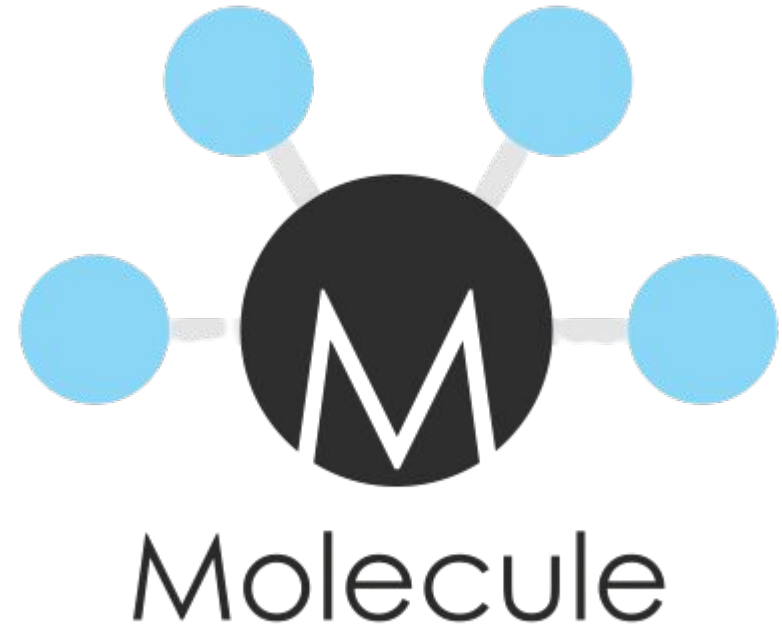
Различные платформы и тесты

Платформы

- ★ Azure
- ★ DigitalOcean
- ★ Docker
- ★ EC2
- ★ GCE
- ★ Hetzner Cloud
- ★ LXC / LXD
- ★ Openstack
- ★ Vagrant

Test matrix

- ★ lint
- ★ dependency
- ★ syntax
- ★ converge
- ★ idempotence
- ★ side_effect



LIVE



Домашнее задание

Проверка достижения целей

#1 Научились писать Ansible-роли

#2 Научились версионировать и подключать роли

#3 Научились тестировать роли



Домашнее задание

- #1** Создать собственную роль установки Docker
- #2** Добавить опциональный шаг сборки докерфайла из шаблона
- #3** Сохранить роль в отдельном репозитории, проверсионировать, и создать плейбук с вызовом этой роли

