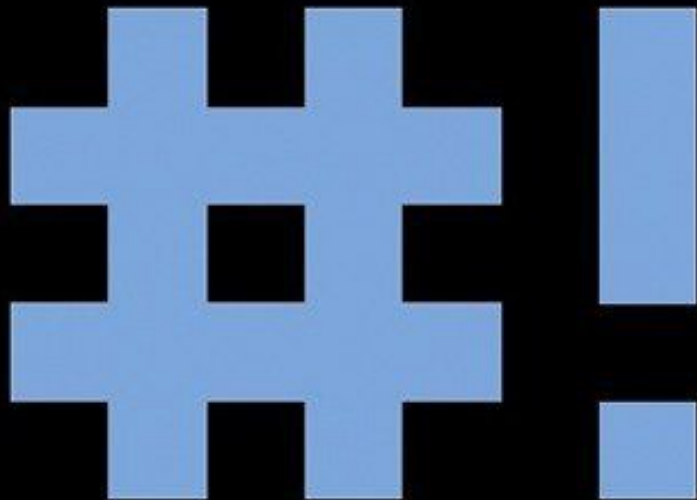


Bash/Ansible

Редькин Максим
Сергеевич

tg @barbudonospes

Bash



- Описание сценариев
- Отсутствие зависимостей
- Простота использования

Переменные

- Переменные среды
- Пользовательские переменные в теле скрипта
- Подстановка команд в переменные `$(command)`
- Математические операции

Циклы

- for
- while
- break
- continue

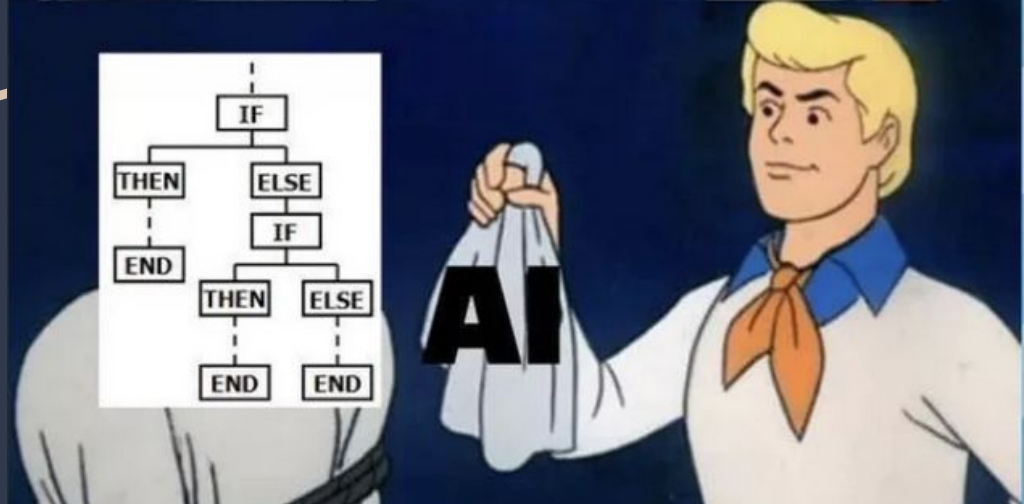
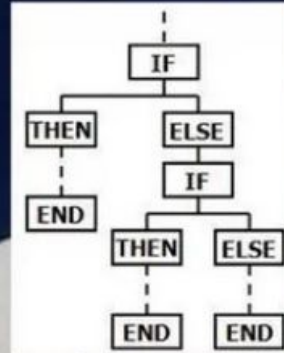


УСЛОВИЯ

if/then

if/then/else

- Сравнение чисел
- Сравнение строк
(длина/совпадение)
- Файлы
- Коды выполнения команд



Параметры и ключи запуска

- Проверка количества параметров (наличия)
- Работа с параметрами без ключей
- Захват всех параметров
- Ключи+shift+while
- Ключи с параметрами

```
/opt/bash.sh ccat paramkey.sh
#!/bin/bash
while [ -n "$1" ]
do
case "$1" in
-a) echo "Found the -a option";;
-b) param="$2"
echo "Found the -b option, with parameter value $param"
shift ;;
-c) echo "Found the -c option";;
--) shift
break ;;
*) echo "$1 is not an option";;
esac
shift
done
count=1
for param in "$@"
do
echo "Parameter #$count: $param"
count=$(( $count + 1 ))
done

/opt/bash.sh ./paramkey.sh -a -h -b 123 --help
Found the -a option
-h is not an option
Found the -b option, with parameter value 123
--help is not an option
```

Рекомендация: Используй стандартные флаги

- -a Вывести все объекты.
- -c Произвести подсчёт.
- -d Указать директорию.
- -e Развернуть объект.
- -f Указать файл, из которого нужно прочитать данные.
- -h Вывести справку по команде.
- -i Игнорировать регистр символов.
- -l Выполнить полноформатный вывод данных.
- -n Использовать неинтерактивный (пакетный) режим.
- -o Позволяет указать файл, в который нужно перенаправить вывод.
- -q Выполнить скрипт в quiet-режиме.
- -r Обрабатывать папки и файлы рекурсивно.
- -s Выполнить скрипт в silent-режиме.
- -v Выполнить многословный вывод.
- -x Исключить объект.
- -y Ответить «yes» на все вопросы.

Дескрипторы файлов

Подавление вывода

```
-----  
/ stdin - 0 \  
| stdout - 1 |  
\ stderr - 2 /  
-----
```

```
      ^__^  
      (oo)\_____  
      (__)\\       )\/\  
           ||----w |  
           ||     ||
```

```
❏ /opt/bash.sh ccat std.sh
```

```
#!/bin/bash
```

```
exec 2>myerror
```

```
echo "This is the start of the script"
```

```
echo "now redirecting all output to another location"
```

```
exec 1>myfile
```

```
echo "This should go to the myfile file"
```

```
echo "and this should go to the myerror file" >&2
```

```
❏ /opt/bash.sh ./std.sh
```

```
This is the start of the script
```

```
now redirecting all output to another location
```

```
❏ /opt/bash.sh cat myerror
```

```
and this should go to the myerror file
```

```
❏ /opt/bash.sh cat myfile
```

```
This should go to the myfile file
```


Функции

- bash умеет в функции
- умеет в return
- умеет в передачу аргументов функции
- умеет подключать функции как библиотеку в окружение пользователя, что бывает полезно

```
q  /opt/bash.sh  tail -n2 ~/.zshrc
. /opt/bash.sh/func.sh

q  /opt/bash.sh  source ~/.zshrc
Usage: myfunc a b

q  /opt/bash.sh  ccat /opt/bash.sh/func.sh
#!/bin/bash
function myfunc {
echo $(( $1 + $2 ))
}
if [ $# -eq 2 ]
then
value=$(myfunc $1 $2)
echo "The result is $value"
else
echo "Usage: myfunc a b"
fi
q  /opt/bash.sh  myfunc 1 2
3
```



Expect

- expect
- autoexpect

```
/opt/bash.sh ccat questions
#!/bin/bash
echo "Hello, who are you?"
read $REPLY
echo "Can I ask you some questions?"
read $REPLY
echo "What is your favorite topic?"
read $REPLY

/opt/bash.sh ccat expect
#!/usr/bin/expect -f
set timeout -1
spawn ./questions
expect "Hello, who are you?\r"
send -- "Im Adam\r"
expect "Can I ask you some questions?\r"
send -- "Sure\r"
expect "What is your favorite topic?\r"
send -- "Technology\r"
expect eof

/opt/bash.sh ./expect
spawn ./questions
Hello, who are you?
Im Adam
Can I ask you some questions?
Sure
What is your favorite topic?
Technology
```

Ansible

Плюсы:

- Минимальные зависимости от окружения
- Гибкость
- Простота
- Много модулей
- ssh

Минусы:

- Капризный `yaml/yml` который транслируется в `python`
- Много модулей

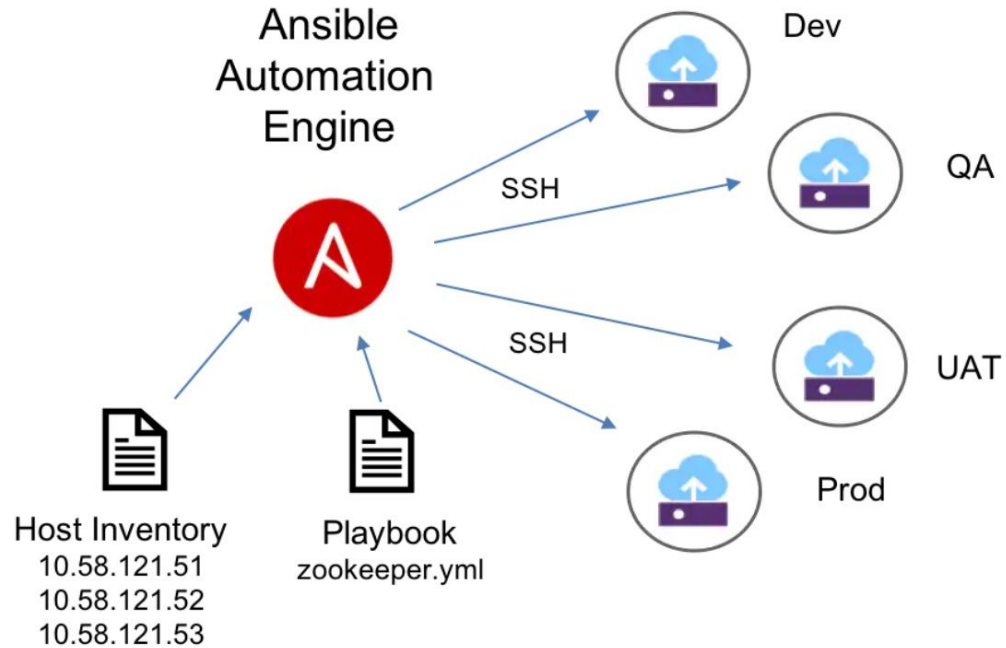
<https://external-preview.redd.it/GxtnpilUVfrC4FY315toV-PYFe9qLDdRltvJzaQHSWUE.png?auto=webp&s=2d620907819a1d0b5e292c7f9233ca5919fc4f36>



ANSIBLE

Минимальные термины:

- playbook
- inventory
- ansible.cfg



ansible inventory

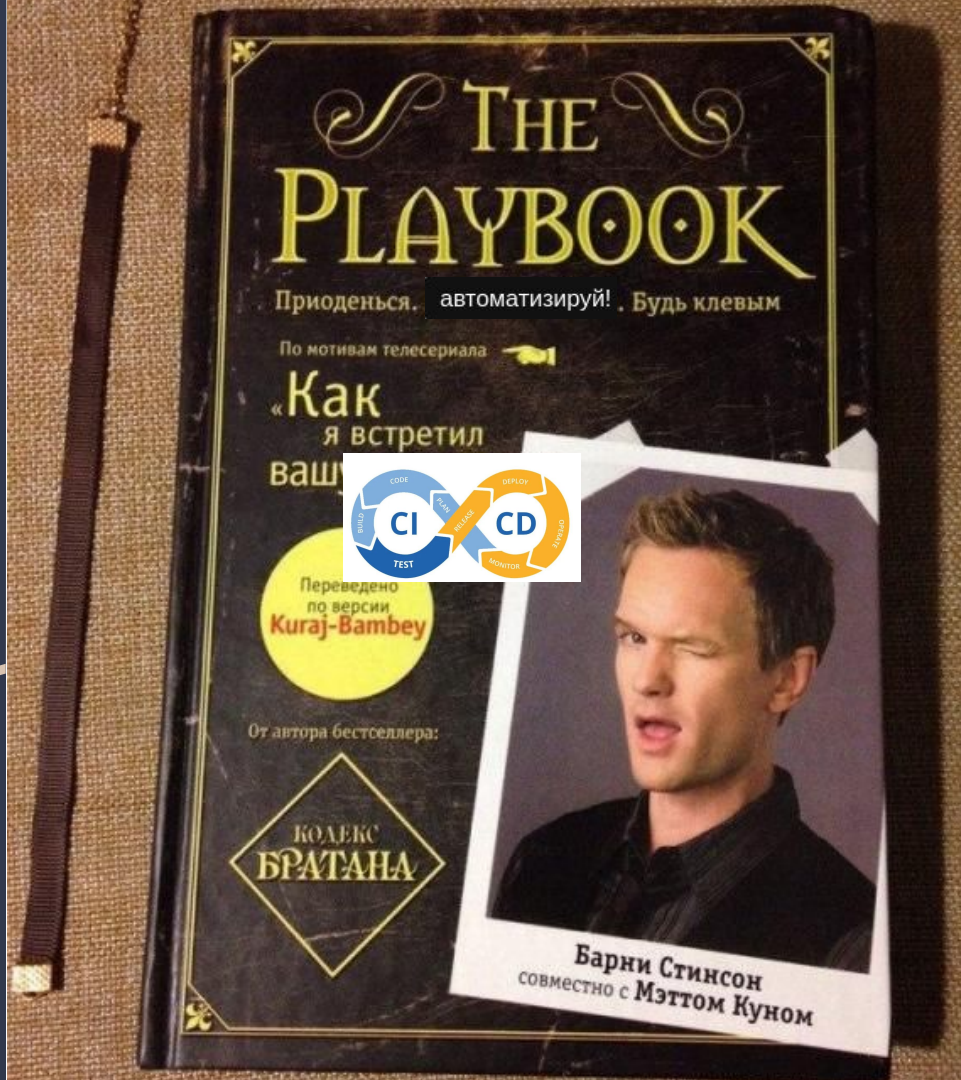
- yml/yaml
- описание хостов
- определение групп и дерева

```
---
all:
  children:
    uk:
      children:
        london:
          hosts:
            www1.example.com:
            db1.example.com:
        manchester:
          hosts:
            www2.example.com:
            db2.example.com:
      database:
        hosts:
          db1.example.com:
          db2.example.com:
    www:
      hosts:
        www1.example.com:
        www2.example.com:

```


ansible playbooks

- yml/yml
- сценарии
- модули



```
— ansible.cfg
— chrony.conf
— chrony.yaml
— inventory.yaml
— ntp.conf
```

0 directories, 5 files

PLAY [noname] *****

TASK [Gathering Facts] *****

```
ok: [smev-db-02]
ok: [smev-lb-02]
ok: [smev-db-01]
ok: [smev-lb-01]
ok: [smev-web-03]
ok: [smev-web-02]
ok: [smev-logs-1]
ok: [smev-web-01]
ok: [smev-file-01]
```

TASK [Check /etc/chrony.conf] *****

```
ok: [smev-db-02]
ok: [smev-db-01]
ok: [smev-lb-02]
ok: [smev-lb-01]
ok: [smev-file-01]
ok: [smev-web-02]
ok: [smev-web-03]
ok: [smev-logs-1]
ok: [smev-web-01]
```

TASK [Check that the somefile.conf exists] *****

```
ok: [smev-db-02]
ok: [smev-db-01]
ok: [smev-lb-02]
ok: [smev-lb-01]
ok: [smev-file-01]
ok: [smev-web-02]
ok: [smev-web-03]
ok: [smev-logs-1]
ok: [smev-web-01]
```



```
ccat chrony.yaml
- name: noname
  hosts: all
  become: true

tasks:
- name: Check /etc/chrony.conf
  stat:
    path: /etc/chrony.conf
    register: stat_chrony

- name: Check that the somefile.conf exists
  stat:
    path: /etc/ntp.conf
    register: stat_ntp

- name: backup chrony conf
  copy:
    src: /etc/chrony.conf
    dest: /etc/chrony.conf_backup
    remote_src: yes
  when: stat_chrony.stat.exists

- name: Copy new chrony conf to server
  copy:
    src: ./chrony.conf
    dest: /etc/chrony.conf
  when: stat_chrony.stat.exists

- name: backup ntp conf
  copy:
    src: /etc/ntp.conf
    dest: /etc/ntp.conf_backup
    remote_src: yes
  when: stat_ntp.stat.exists

- name: Copy new ntp conf to server
  copy:
    src: ./ntp.conf
    dest: /etc/ntp.conf
  when: stat_ntp.stat.exists

- name: Restart ntp service
  service:
    name: ntpd
    state: restarted
  when: stat_ntp.stat.exists

- name: Restart chrony service
  service:
    name: chronyd.service
    state: restarted
  when: stat_chrony.stat.exists
```

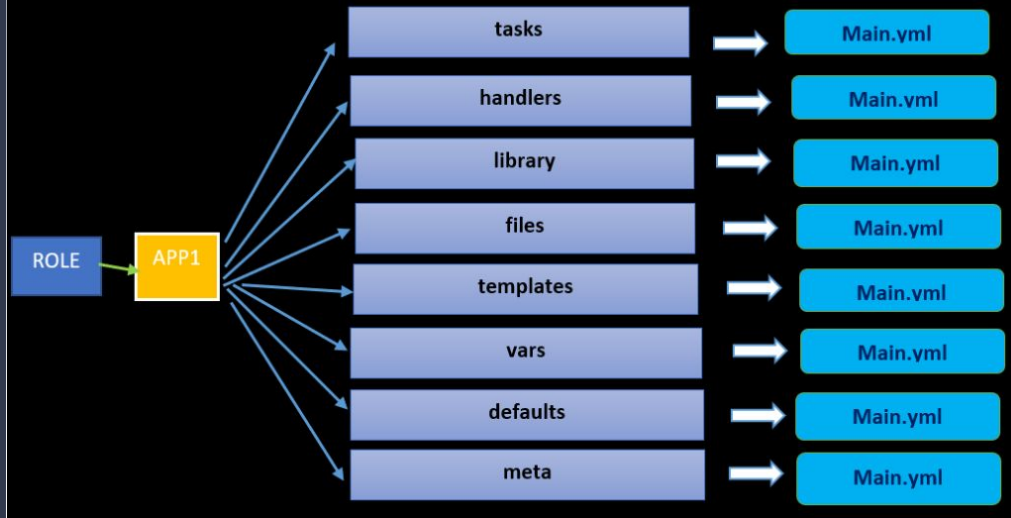
Modules

https://docs.ansible.com/ansible/2.9/modules/list_of_all_modules.html

Ansible role

- Новый уровень структурирования
- Реюзабельность

ansible-galaxy
molecule



```
- hosts: test_servers
  user: dmosk
  become: true
  become_method: su
  become_user: root
  roles:
    - Bind
```



Спасибо
за внимание!



Всех благ!