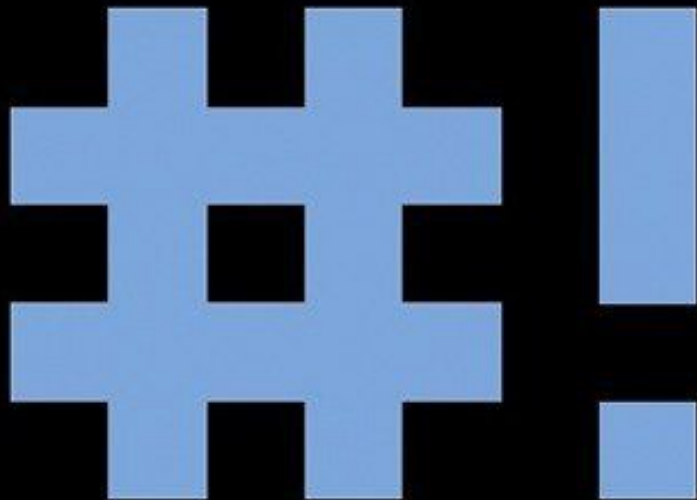


Bash/Ansible

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

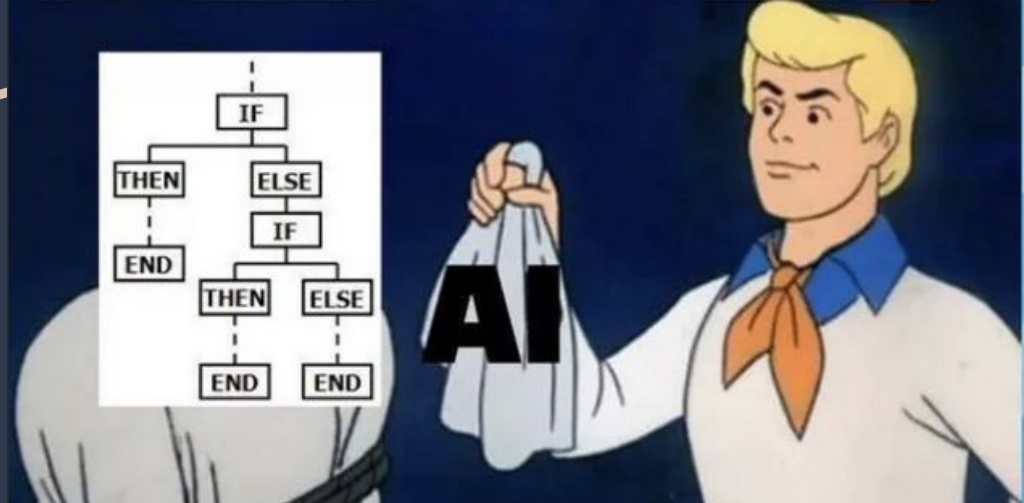
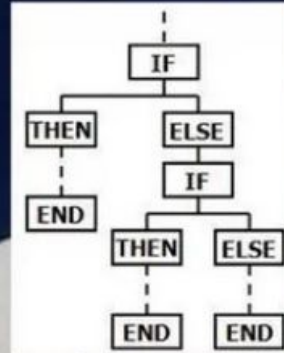


УСЛОВИЯ

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 Обработать 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/ym`l который транслируется в `python`
- Много модулей

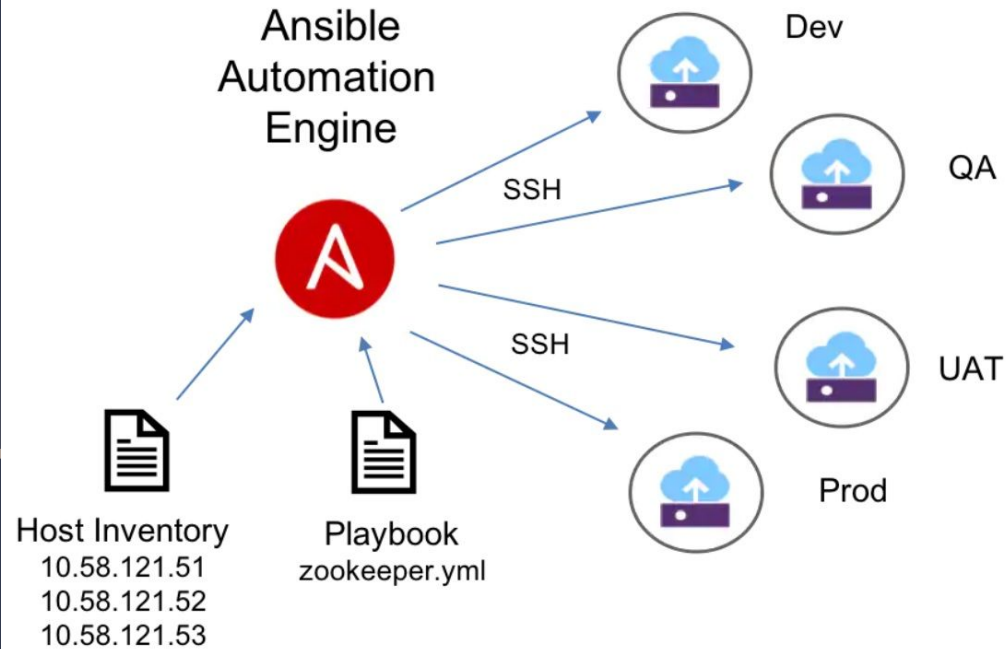
<https://external-preview.redd.it/GxtnpilUVfrC4FY315toV-PYFe9qLDdRltvJzaQHSWUE.png?auto=webp&s=2d620907819a1d0b5e292c7f9233ca5919fc4f36>



ANSIBLE

Терминология:

- playbook
- inventory
- ansible.cfg



ansible inventory

- yml/yaml/ini
- 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:

```

```
# fmt: ini
# Example 1

[web]
host1
host2 ansible_port=222 # defined inline, interpreted as an integer

[web:vars]
http_port=8080 # all members of 'web' will inherit these
myvar=23 # defined in a :vars section, interpreted as a string

[web:children] # child groups will automatically add their hosts to parent group
apache
nginx

[apache]
tomcat1
tomcat2 myvar=34 # host specific vars override group vars
tomcat3 mysecret="'03#pa33w0rd'" # proper quoting to prevent value changes

[nginx]
jenkins1

[nginx:vars]
has_java = True # vars in child groups override same in parent

[all:vars]
has_java = False # 'all' is 'top' parent

# Example 2
host1 # this is 'ungrouped'

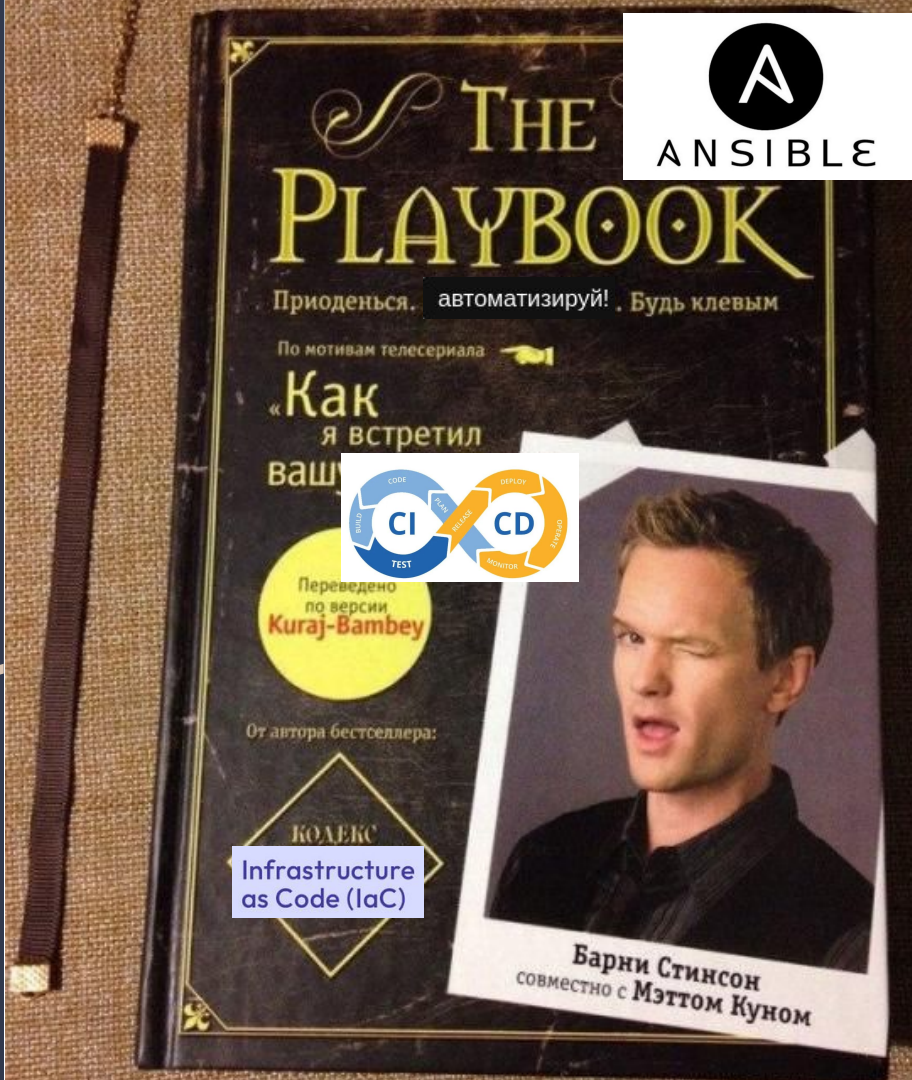
# both hosts have same IP but diff ports, also 'ungrouped'
host2 ansible_host=127.0.0.1 ansible_port=44
host3 ansible_host=127.0.0.1 ansible_port=45

[g1]
host4

[g2]
host4 # same host as above, but member of 2 groups, will inherit vars from both
# inventory hostnames are unique
```


- уml/yaml
- сценарии
- модули

- yml/yaml
- сценарии
- модули



Составные части

Имя хоста (Name) и Подключения (Connections) (RUN)

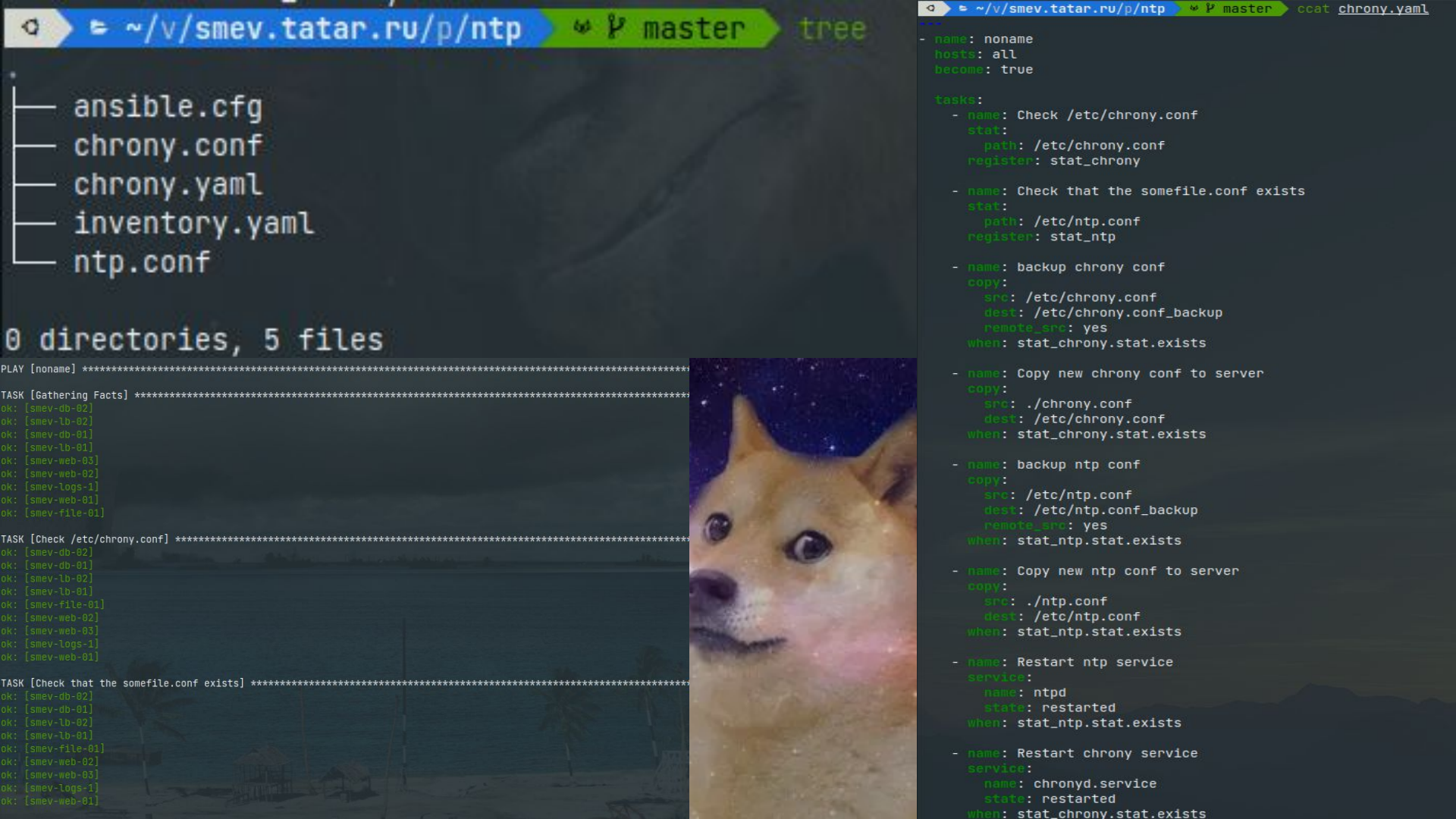
Переменные (Variables)

Инструкции (Tasks)

Обработчики (Handlers)

Шаблоны (Templates)

Используемые роли (Roles)



```
— 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]
```

```
~ /v/smev.tatar.ru/p/ntp master tree
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
```

Имя хоста (Name) и Подключения (Connections)
(RUN)

```
---  
- name: Configure web servers  
  hosts: webserver  
  remote_user: root  
  gather_facts: no
```



Переменные (Variables)

```
---  
vars:  
  http_port: 80  
  max_clients: 200  
  document_root: "/var/www/html"
```

Extra vars (from command-line) always win.

Task vars (only for the specific task).

Block vars (only for the tasks within the block).

Role and include vars.

Vars created with `set_fact`.

Vars created with the `register task` directive.

Play vars_files.

Play vars_prompt.

Play vars.

Host facts.

Playbook host_vars.

Playbook group_vars.

Inventory host_vars.

Inventory group_vars.

Inventory vars.

Role defaults.

TMP

TEMP_

TEMPORARY

**THIS VARIABLE WILL ONLY
BE USED FOR
A SHORT PERIOD OF TIME**



Modules

Google

🔍 ansible modules



Поиск в Google

Мне повезёт!



Инструкции (Tasks)

```
---  
- name: Ensure Apache is installed  
  package:  
    name: apache2  
    state: present  
  become: yes  
  
- name: Ensure Apache is running  
  service:  
    name: apache2  
    state: started  
    enabled: yes  
  become: yes
```

When you add shell
command `rm -rf *` in
ansible playbook with
hosts all with `-b` option



Обработчики (Handlers)

механизм для определения действий, которые должны быть выполнены только в случае изменений на управляемых узлах

Умеет работать по notify

Не сработают если хоть 1 таск упадет в процессе

tasks:

- name: Restart memcached
command: echo "this task will restart the services"
notify: "restart memcached"

handlers:

- name: Restart memcached
service:
name: memcached
state: restarted
listen: "restart web memcached"
- name: Restart apache
service:
name: apache
state: restarted
listen: "restart web"

~
~
~

Шаблоны (Templates)

Динамическое создание конфигов на базе переменных

jinja2



Line
In File



Jinja2
Templates

```
2: vi
# inventory.ini
[zabbix_agents]
server1 ansible_host=192.168.1.101
server2 ansible_host=192.168.1.102
server3 ansible_host=192.168.1.103
```

-- РЕЖИМ ВСТАВКИ --

```
1: vi
# zabbix_agentd.conf.j2
Server={{ zabbix_server_ip }}
Hostname={{ ansible_hostname }}
ListenPort=10050
Timeout=30
```

-- РЕЖИМ ВСТАВКИ --

```
3: vi
# playbook.yaml
- name: Конфигурирование Zabbix агентов
  hosts: zabbix_agents
  become: true
  vars:
    zabbix_server_ip: "192.168.1.100" # IP-адрес сервера Zabbix

  tasks:
    - name: Создать конфигурационный файл Zabbix агента
      template:
        src: /путь/к/zabbix_agentd.conf.j2
        dest: /etc/zabbix/zabbix_agentd.conf
```

Используемые роли (Roles)

Не забывайте про конфигурацию `ansible`, где будет искать роли

```
- name: Установка и настройка Nginx
hosts: web_servers
become: true
roles:
  - nginx
```



Ansible role

- Новый уровень структурирования
- Реюзабельность

ansible-galaxy
molecule

```
❏ /tmp/test ➤ ansible-galaxy role init test_role
- Role test_role was created successfully
❏ /tmp/test ➤ ls -la
итого 0
drwxr-xr-x  3 maksimredkin maksimredkin  60 map  6 09:58 .
drwxrwxrwt 21 root          root          780 map  6 09:57 ..
drwxr-xr-x 10 maksimredkin maksimredkin 240 map  6 09:58 test_role
❏ /tmp/test ➤ cd test_role
❏ /tmp/test/test_role ➤ tree
.
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml

9 directories, 8 files
```

Заметки

AWX

Ansible tower

ansible semaphore ui

rundeck

ansible-vault



Спасибо
за внимание!



Всех благ!