PYTHON ДЛЯ СЕТЕВЫХ ИНЖЕНЕРОВ

МОДУЛИ ДЛЯ РАБОТЫ С СЕТЕВЫМ ОБОРУДОВАНИЕМ

МОДУЛЬ IOS_CONFIG

МОДУЛЬ IOS_CONFIG

ПАРАМЕТР MATCH

MATCH

Параметр **match** указывает как именно нужно сравнивать команды (что считается изменением):

- **line** - команды проверяются построчно. Этот режим используется по умолчанию
- **strict** - должны совпасть не только сами команды, но их положение относительно друг друга
- **exact** - команды должны в точности сопадать с конфигурацией и не должно быть никаких лишних строк
- **none** - модуль не будет сравнивать команды с текущей конфигурацией

MATCH: LINE

Режим `match: line` используется по умолчанию.

В этом режиме, модуль проверяет только наличие строк, перечисленных в списке lines в соответствующем режиме. При этом, не проверяется порядок строк.

MATCH: LINE

На маршрутизаторе 192.168.100.1 настроен такой ACL:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
```

MATCH: LINE

Пример использования playbook 9_ios_config_match_line.yml в режиме line:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
```

MATCH: LINE

Результат выполнения playbook:

```
$ ansible-playbook 9_ios_config_match_line.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["ip access-list extended
 IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.255 any eq www", "permit icmp any any"], "
warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

MATCH: LINE

Обратите внимание, что в списке updates только две из трёх строк ACL. Так как в режиме lines модуль сравнивает команды независимо друг от друга, он обнаружил, что не хватает только двух команд из трех.

MATCH: LINE

В итоге конфигурация на маршрутизаторе выглядит так:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit icmp any any
```

То есть, порядок команд поменялся. И, хотя в этом случае, это не важно, иногда это может привести совсем не к тем результатам, которые ожидались.

Если повторно запустить playbook, при такой конфигурации, он не будет выполнять изменения, так как все строки были найдены.

MATCH: EXACT

Пример, в котором порядок команд важен.

ACL на маршрутизаторе:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 deny   ip any any
```

MATCH: EXACT

Playbook 9_ios_config_match_exact.yml (будет постепенно дополняться):

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
          - deny   ip any any
```

MATCH: EXACT

Если запустить playbook, результат будет таким:

```
$ ansible-playbook 9_ios_config_match_exact.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["ip access-list extended
 IN_to_OUT", "permit icmp any any", "deny ip any any"], "warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

MATCH: EXACT

Теперь ACL выглядит так:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 deny   ip any any
 permit icmp any any
```

Конечно же, в таком случае, последнее правило никогда не сработает.

MATCH: EXACT

Можно добавить к этому playbook параметр before и сначала удалить ACL, а затем применять команды:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        before:
          - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
          - deny   ip any any
```

Если применить playbook к последнему состоянию маршрутизатора, то изменений не будет никаких, так как все строки уже есть.

MATCH: EXACT

Попробуем начать с такого состояния ACL:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 deny   ip any any
```

MATCH: EXACT

Результат будет таким:

```
$ ansible-playbook 9_ios_config_match_exact.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:



PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["no ip access-list exten
ded IN_to_OUT", "ip access-list extended IN_to_OUT", "permit icmp any any"], "warn
ings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

MATCH: EXACT

И, соответственно, на маршрутизаторе:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit icmp any any
```

MATCH: EXACT

Теперь в ACL осталась только одна строка:

- Модуль проверил каких команд не хватает в ACL (так как режим по умолчанию match: line),
- обнаружил, что не хватает команды `permit icmp any any` и добавил её

Но, так как в playbook ACL сначала удаляется, а затем применяется список команд lines, получилось, что в итоге в ACL одна строка.

MATCH: EXACT

Поможет, в такой ситуации, вариант `match: exact`:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        before:
          - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
          - deny   ip any any
        match: exact
```

MATCH: EXACT

Применение playbook 9_ios_config_match_exact.yml к текущему состоянию маршрутизатора (в ACL одна строка):

```
$ ansible-playbook 9_ios_config_match_exact.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["no ip access-list exten


ded IN_to_OUT", "ip access-list extended IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.25
5 any eq www", "permit tcp 10.0.1.0 0.0.0.255 any eq 22", "permit icmp any any", "
deny   ip any any"], "warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

MATCH: EXACT

Теперь результат такой:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
 deny   ip any any
```

То есть, теперь ACL выглядит точно так же, как и строки в списке lines и в том же порядке.

MATCH: EXACT

Закомментируем в playbook строки с удалением ACL:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        #before:
        #  - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
          - deny   ip any any
        match: exact
```

MATCH: EXACT

В начало ACL добавлена строка:

```
ip access-list extended IN_to_OUT
 permit udp any any
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
 deny   ip any any
```

MATCH: EXACT

То есть, последние 4 строки выглядят так, как нужно, и в том порядке, котором нужно. Но, при этом, есть лишняя строка. Для варианта match: exact - это уже несовпадение.

MATCH: EXACT

В таком варианте, playbook будет выполняться каждый раз и пытаться применить все команды из списка lines, что не будет влиять на содержимое ACL:

```
$ ansible-playbook 9_ios_config_match_exact.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["ip access-list extended
 IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.255 any eq www", "permit tcp 10.0.1.0 0.0.
0.255 any eq 22", "permit icmp any any", "deny   ip any any"], "warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

MATCH: EXACT

Это значит, что при использовании `match: exact`, важно, чтобы был какой-то способ удалить конфигурацию, если она не соответствует тому, что должно быть (или чтобы команды перезаписывались). Иначе, эта задача будет выполняться каждый раз, при запуске playbook.

MATCH: STRICT

Вариант `match: strict` не требует, чтобы объект был в точности как указано в задаче, но, команды, которые указаны в списке lines, должны быть в том же порядке.

Если указан список parents, команды в списке lines должны идти сразу за командами parents.

MATCH: STRICT

На маршрутиазаторе такой ACL:

```
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
 deny   ip any any
```

MATCH: STRICT

Playbook 9_ios_config_match_strict.yml:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        before:
          - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
        match: strict
```

MATCH: STRICT

Выполнение playbook:

```
$ ansible-playbook 9_ios_config_match_strict.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
ok: [192.168.100.1] => {"changed": false, "warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=0    unreachable=0    failed=0
```

MATCH: STRICT

Так как изменений не было, ACL остался таким же.

В такой же ситуации, при использовании `match: exact`, было бы обнаружено изменение и ACL бы состоял только из строк в списке lines.

MATCH: NONE

Использование `match: none` отключает идемпотентность задачи: каждый раз при выполнении playbook, будут отправляться команды, которые указаны в задаче.

MATCH: NONE

Пример playbook 9_ios_config_match_none.yml:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        before:
          - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
        match: none
```

MATCH: NONE

Каждый раз при запуске playbook результат будет таким:

```
$ ansible-playbook 9_ios_config_match_none.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["no ip access-list exten
ded IN_to_OUT", "ip access-list extended IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.25
5 any eq www", "permit tcp 10.0.1.0 0.0.0.255 any eq 22", "permit icmp any any"],
"warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

Использование `match: none` подходит в тех случаях, когда, независимо от текущей конфигурации, нужно отправить все команды.

МОДУЛЬ IOS_CONFIG

ПАРАМЕТР REPLACE

REPLACE

Параметр replace указывает как именно нужно заменять конфигурацию:

- **line** - в этом режиме отправляются только те команды, которых нет в конфигурации. Этот режим используется по умолчанию
- **block** - в этом режиме отправляются все команды, если хотя бы одной команды нет

REPLACE: LINE

Режим `replace: line` - это режим работы по умолчанию. В этом режиме, если были обнаружены изменения, отправляются только недостающие строки.

Например, на маршрутизаторе такой ACL:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
```

REPLACE: LINE

Попробуем запустить такой playbook 10_ios_config_replace_line.yml:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        before:
          - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
          - deny   ip any any
```

REPLACE: LINE

Выполнение playbook:

```
$ ansible-playbook 10_ios_config_replace_line.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["no ip access-list exten
ded IN_to_OUT", "ip access-list extended IN_to_OUT", "deny   ip any any"], "warnin
gs": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

REPLACE: LINE

После этого на маршрутизаторе такой ACL:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 deny   ip any any
```

REPLACE: LINE

В данном случае, модуль проверил каких команд не хватает в ACL (так как режим по умолчанию match: line), обнаружил, что не хватает команды `deny ip any any` и добавил её. Но, так как ACL сначала удаляется, а затем применяется список команд lines, получилось, что у теперь ACL с одной строкой.

В таких ситуациях подходит режим `replace: block`.

REPLACE: BLOCK

В режиме `replace: block` отправляются все команды из списка lines (и parents), если на устройстве нет хотя бы одной из этих команд.

Повторим предыдущий пример.

REPLACE: BLOCK

ACL на маршрутизаторе:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
```

REPLACE: BLOCK

Playbook 10_ios_config_replace_block.yml:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        before:
          - no ip access-list extended IN_to_OUT
        parents:
          - ip access-list extended IN_to_OUT
        lines:
          - permit tcp 10.0.1.0 0.0.0.255 any eq www
          - permit tcp 10.0.1.0 0.0.0.255 any eq 22
          - permit icmp any any
          - deny   ip any any
        replace: block
```

REPLACE: BLOCK

Выполнение playbook:

```
$ ansible-playbook 10_ios_config_replace_block.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"changed": true, "updates": ["no ip access-list exten
ded IN_to_OUT", "ip access-list extended IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.25
5 any eq www", "permit tcp 10.0.1.0 0.0.0.255 any eq 22", "permit icmp any any", "
deny   ip any any"], "warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

REPLACE: BLOCK

В результате на маршрутизаторе такой ACL:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
 deny   ip any any
```

МОДУЛЬ IOS_CONFIG

ПАРАМЕТР SRC

SRC

Параметр **src** позволяет указывать путь к файлу конфигурации или шаблону конфигурации, которую нужно загрузить на устройство.

Этот параметр взаимоисключающий с lines (то есть, можно указывать или lines или src). Он заменяет модуль ios_template, который скоро будет удален.

КОНФИГУРАЦИЯ

Пример playbook 11_ios_config_src.yml:

```
- name: Run cfg commands on router
  hosts: 192.168.100.1

  tasks:

    - name: Config ACL
      ios_config:
        src: templates/acl_cfg.txt
```

SRC

В файле templates/acl_cfg.txt находится такая конфигурация:

```
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
 deny   ip any any
```

SRC

Удаляем на маршрутизаторе этот ACL, если он остался с прошлых разделов, и запускаем playbook:

```
$ ansible-playbook 11_ios_config_src.yml -v
```

```
PLAY [Run cfg commands on router] **********************************************

TASK [Config ACL] **************************************************************
changed: [192.168.100.1] => {"banners": {}, "changed": true, "commands": ["ip access-list e
xtended IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.255 any eq www", "permit tcp 10.0.1.0 0.0.0.
255 any eq 22", "permit icmp any any", "deny   ip any any"], "failed": false, "updates": ["
ip access-list extended IN_to_OUT", "permit tcp 10.0.1.0 0.0.0.255 any eq www", "permit tcp
 10.0.1.0 0.0.0.255 any eq 22", "permit icmp any any", "deny   ip any any"]}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=1    unreachable=0    failed=0
```

SRC

Неприятная особенность параметра src в том, что не видно какие изменения были внесены. Но, возможно, в следующих версиях Ansible это будет исправлено.

SRC

Теперь на маршрутизаторе настроен ACL:

```
R1#sh run | s access
ip access-list extended IN_to_OUT
 permit tcp 10.0.1.0 0.0.0.255 any eq www
 permit tcp 10.0.1.0 0.0.0.255 any eq 22
 permit icmp any any
 deny   ip any any
```

SRC

Если запустить playbook ещё раз, но никаких изменений не будет, так как этот параметр также идемпотентен:

```
$ ansible-playbook 11_ios_config_src.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config ACL] ***************************************************************
ok: [192.168.100.1] => {"changed": false, "warnings": []}

PLAY RECAP **********************************************************************
192.168.100.1              : ok=1    changed=0    unreachable=0    failed=0
```

ШАБЛОН JINJA2

В параметре src можно указывать шаблон Jinja2.

Пример шаблона (файл templates/ospf.j2):

```
router ospf 1
 router-id {{ mgmnt_ip }}
 ispf
 auto-cost reference-bandwidth 10000
{% for ip in ospf_ints %}
 network {{ ip }} 0.0.0.0 area 0
{% endfor %}
```

ШАБЛОН JINJA2

В шаблоне используются две переменные:

- mgmnt_ip - IP-адрес, который будет использоваться как router-id
- ospf_ints - список IP-адресов интерфейсов, на которых нужно включить OSPF

Для настройки OSPF на трёх маршрутизаторах, нужно иметь возможность использовать разные значения этих переменных для разных устройств. Для таких задач используются файлы с переменными в каталоге host_vars.

В каталоге host_vars нужно создать такие файлы (если они ещё не созданы):

SRC

Файл host_vars/192.168.100.1:

```
hostname: london_r1
mgmnt_loopback: 100
mgmnt_ip: 10.0.0.1
ospf_ints:
  - 192.168.100.1
  - 10.0.0.1
  - 10.255.1.1
```

SRC

Файл host_vars/192.168.100.2:

```
hostname: london_r2
mgmnt_loopback: 100
mgmnt_ip: 10.0.0.2
ospf_ints:
  - 192.168.100.2
  - 10.0.0.2
  - 10.255.2.2
```

SRC

Файл host_vars/192.168.100.3:

```
hostname: london_r3
mgmnt_loopback: 100
mgmnt_ip: 10.0.0.3
ospf_ints:
  - 192.168.100.3
  - 10.0.0.3
  - 10.255.3.3
```

SRC

Теперь можно создавать playbook 11_ios_config_src_jinja.yml:

```
- name: Run cfg commands on router
  hosts: cisco-routers

  tasks:

    - name: Config OSPF
      ios_config:
        src: templates/ospf.j2
```

SRC

Так как Ansible сам найдет переменные в каталоге host_vars, их не нужно указывать. Можно сразу запускать playbook:

```
$ ansible-playbook 11_ios_config_src_jinja.yml -v
```

SRC

Теперь на всех маршрутизаторах настроен OSPF:

```
R1#sh run | s ospf
router ospf 1
 router-id 10.0.0.1
 ispf
 auto-cost reference-bandwidth 10000
 network 10.0.0.1 0.0.0.0 area 0
 network 10.255.1.1 0.0.0.0 area 0
 network 192.168.100.1 0.0.0.0 area 0

R2#sh run | s ospf
router ospf 1
 router-id 10.0.0.2
 ispf
 auto-cost reference-bandwidth 10000
 network 10.0.0.2 0.0.0.0 area 0
 network 10.255.2.2 0.0.0.0 area 0
 network 192.168.100.2 0.0.0.0 area 0

router ospf 1
 router-id 10.0.0.3
 ispf
 auto-cost reference-bandwidth 10000
 network 10.0.0.3 0.0.0.0 area 0
 network 10.255.3.3 0.0.0.0 area 0
 network 192.168.100.3 0.0.0.0 area 0
```

SRC

Если запустить playbook ещё раз, но никаких изменений не будет:

```
$ ansible-playbook 11_ios_config_src_jinja.yml -v
```

```
Using /home/nata/pyneng_course/chapter15/ansible.cfg as config file
SSH password:

PLAY [Run cfg commands on router] ***********************************************

TASK [Config OSPF] *************************************************************
ok: [192.168.100.3] => {"changed": false, "warnings": []}
ok: [192.168.100.2] => {"changed": false, "warnings": []}
ok: [192.168.100.1] => {"changed": false, "warnings": []}

PLAY RECAP *********************************************************************
192.168.100.1              : ok=1    changed=0    unreachable=0    failed=0
192.168.100.2              : ok=1    changed=0    unreachable=0    failed=0
192.168.100.3              : ok=1    changed=0    unreachable=0    failed=0
```

СОВМЕЩЕНИЕ С ДРУГИМИ ПАРАМЕТРАМИ

Параметр **src** совместим с такими параметрами:

- backup
- config
- defaults
- save (но у самого save в Ansible 2.2 проблемы с работой)