

The MicroTik logo is displayed in a stylized, italicized font. The word "Mikro" is in a lighter weight, while "Tik" is in a bold, black weight. A small, curved graphic element is positioned above the letter 'i' in "Mikro".

*Mikro***Tik**

The word "ANSIBLE" is written in a clean, white, uppercase, sans-serif font. A light blue diagonal line is positioned to the left of the text, starting from the top left and extending towards the bottom right, partially overlapping the letter 'A'.

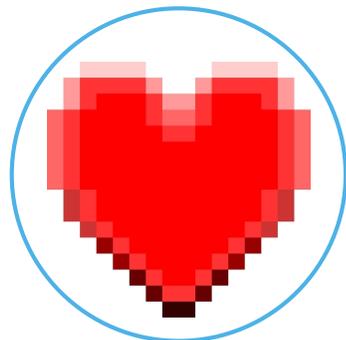
ANSIBLE

Вадим Шепелев. MikroTik User Meeting. Россия, Москва 2016

# О себе



**Шепелев Вадим**



**Первое  
знакомство с  
MikroTik ~2003  
году, любовь с  
первого взгляда ;)**



**Работаю  
системным  
администратором**



**МТСТСЕ,  
МТСВЕ, МТСРЕ**



**В компании  
используется ~500  
RouterBoard's**

## Ссылки на код

---

<https://github.com/ox566164696D/mikroansible>

<http://bit.ly/2cjF7wl>



# Идея и принципы взаимодействия Ansible и RouterOS

# Идея и принципы взаимодействия Ansible и RouterOS >

## ANSIBLE

Популярное ПО для управления конфигурациями серверов и проч. оборудования

Написан на Python, лицензия GPL

Работает через SSH (аутентификация по user\passwd или ключам)

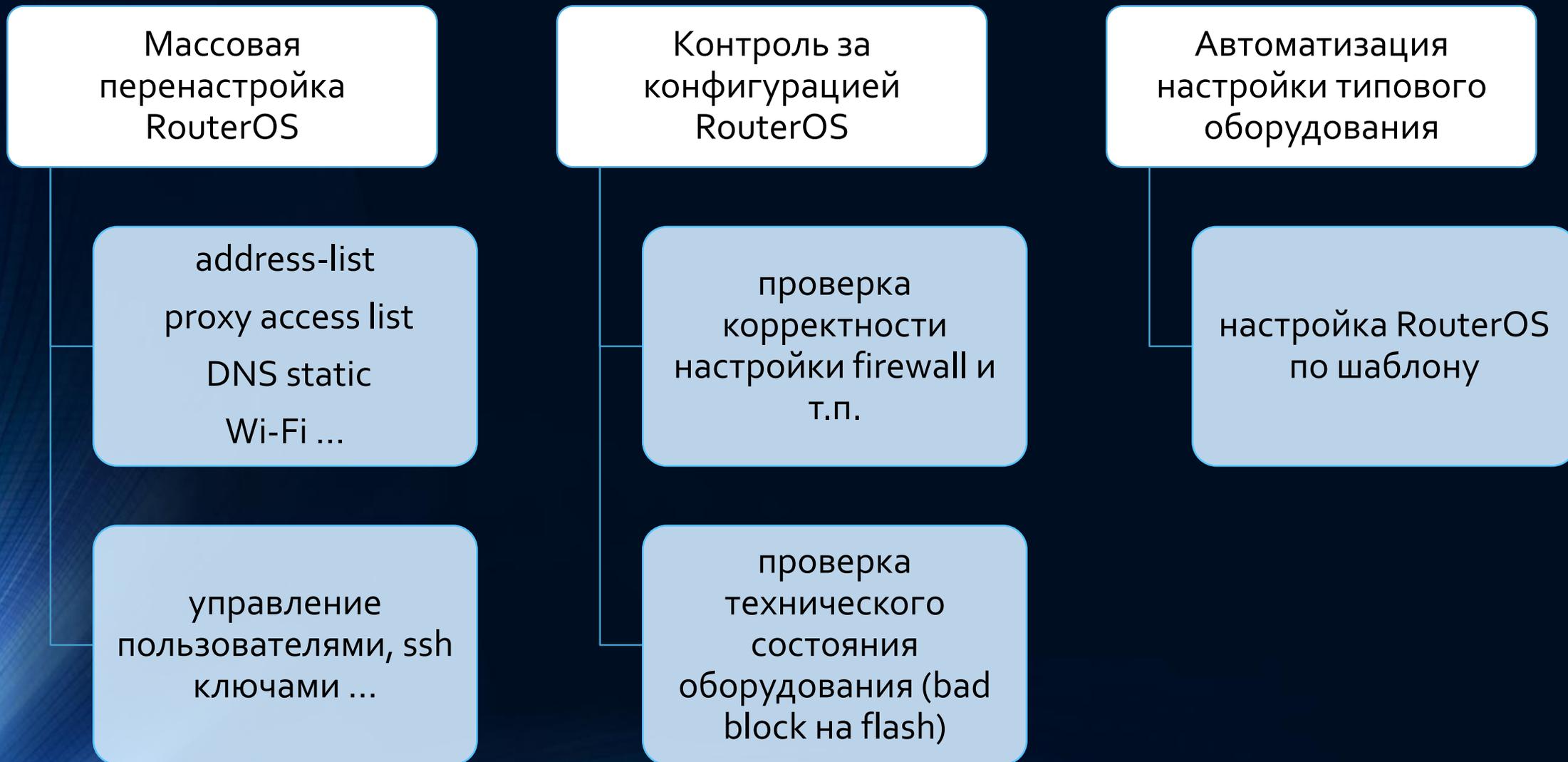
Доп. ПО (агент) на стороне управляемого оборудования не требуется

Использует язык разметки YAML для написания своих сценариев

Доступен для установки через пакетный менеджер в большинстве популярных Linux дистрибутивов.

# Идея и принципы взаимодействия Ansible и RouterOS >

## Какие задачи можно решать, применяя Ansible для работы с RouterOS



# Идея и принципы взаимодействия Ansible и RouterOS >

## Документация, примеры, статьи об Ansible

---

- <http://docs.ansible.com/ansible/index.html>
- <https://galaxy.ansible.com/list#/roles>
- <https://habrahabr.ru/company/selectel/blog/196620/>
- <https://habrahabr.ru/post/306998/>

# Идея и принципы взаимодействия Ansible и RouterOS > «Симбиоз» Ansible и RouterOS Script (.rsc)

## Ansible

(“Хранилище и Транспорт”)

Централизованное  
хранилище конфигураций

Средство доставки \ запуска  
RouterOS script (.rsc)

Контролер доставки  
скриптов на роутер

Генератор .rsc скриптов \  
шаблонов конфигураций



RouterOS script (.rsc)  
(“Полезная нагрузка”)

Осуществляет  
все операции  
на RouterOS.

# Идея и принципы взаимодействия Ansible и RouterOS >

## Пожелание к сети

---

Логичная организация адресации в сети.

Сеть должна быть структурирована и организована по определенному принципу.

Легче организовать автоматический поиск новых устройств или массовое добавление уже имеющихся.

---

# Идея и принципы взаимодействия Ansible и RouterOS >

## Требования к управляемому роутеру

---

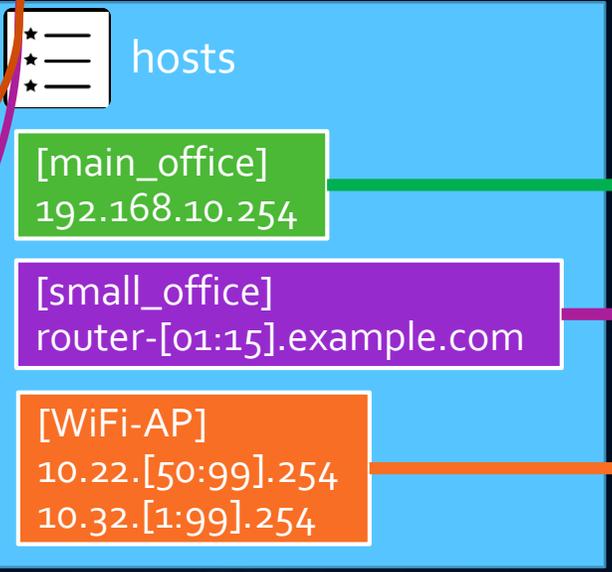
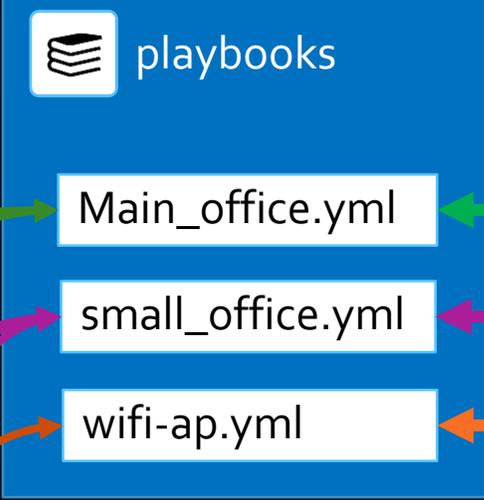
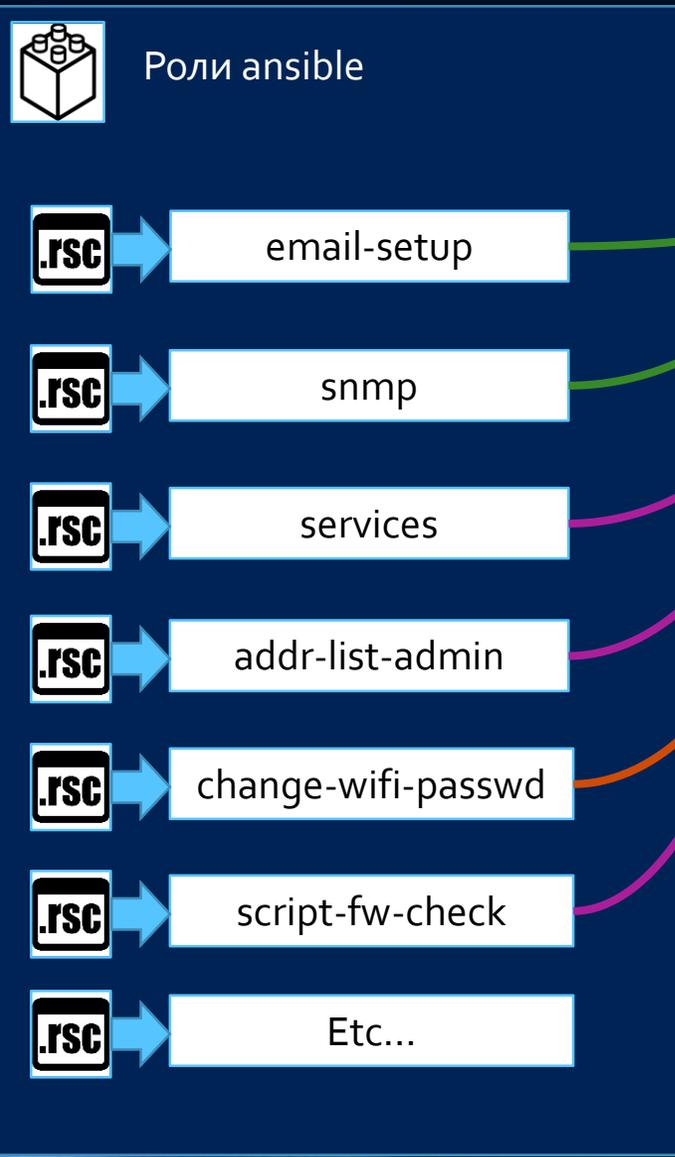
### Обязательно

- Доступный ssh на RouterOS

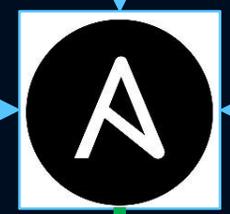
### Желательно

- Поддержка аутентификации по RSA ключам (RouterOS 6.31 +)

# Массовая перенастройка RouterOS средствами Ansible



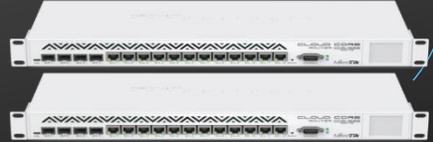
переменные      ansible.cfg



SSH

SSH

SSH



Main office RB's



Small offices RB's



WiFi AP RB's

# Массовая перенастройка RouterOS средствами Ansible > Конфигурационный файл ansible.cfg

Позволяет настроить множество важных для работы параметров.

- Количество потоков (кол-во одновременно настраиваемых роутеров)
- Имя пользователя для подключения к роутерам
- Пути до ssh ключей
- Журналирование
- Многое другое

## Фрагмент файла ansible.cfg

```
hostfile      = $HOME/ansible/hosts
library       = /usr/share/ansible
remote_tmp    = $HOME/.ansible/tmp
pattern       = *
forks         = 40
poll_interval = 15
#sudo_user    = root
#ask_sudo_pass = True
#ask_pass     = True
transport     = smart
remote_port   = 22

# additional paths to search for roles in, colon seperated
roles_path    = $HOME/ansible/roles

# uncomment this to disable SSH key host checking
host_key_checking = False
```

# Массовая перенастройка RouterOS средствами Ansible > Аутентификация

---

Удобнее всего использовать аутентификацию по ssh ключам.

На сервере с Ansible рекомендую создать отдельного пользователя, из-под которого будет происходить управление роутерами и сгенерировать ключевую пару RSA ключей.

Публичный ключ нужно установить на управляемом оборудовании, а приватный будет находиться на сервере с Ansible.

Тема работы с ssh ключами хорошо освещена.  
Более подробно читайте в Интернет.

# Массовая перенастройка RouterOS средствами Ansible > файл hosts

Определяет перечень оборудования, которым нужно управлять.

- Может содержать IP адреса и их диапазоны
- DNS имена и их диапазоны.
- Управляемые хосты нужно группировать
- Возможно задание переменных

```
[init]
10.22.[50:99].254
10.32.[1:99].254

[main-office]
192.168.10.254 ansible_ssh_port=44333

[test]
router-[01:15].example.com ansible_user=admin

[all:children]
init
main-office
test
```

# Массовая перенастройка RouterOS средствами Ansible >

## Роли. Что это?

---

Роль – универсальный, самодостаточный и независимый от других блок, описывающий определенный аспект конфигурации роутера.

Например:

- Настройка сервисов RouterOS
- Настройка email
- Настройка SNMP
- Настройка DNS
- Настройка WiFi
- etc

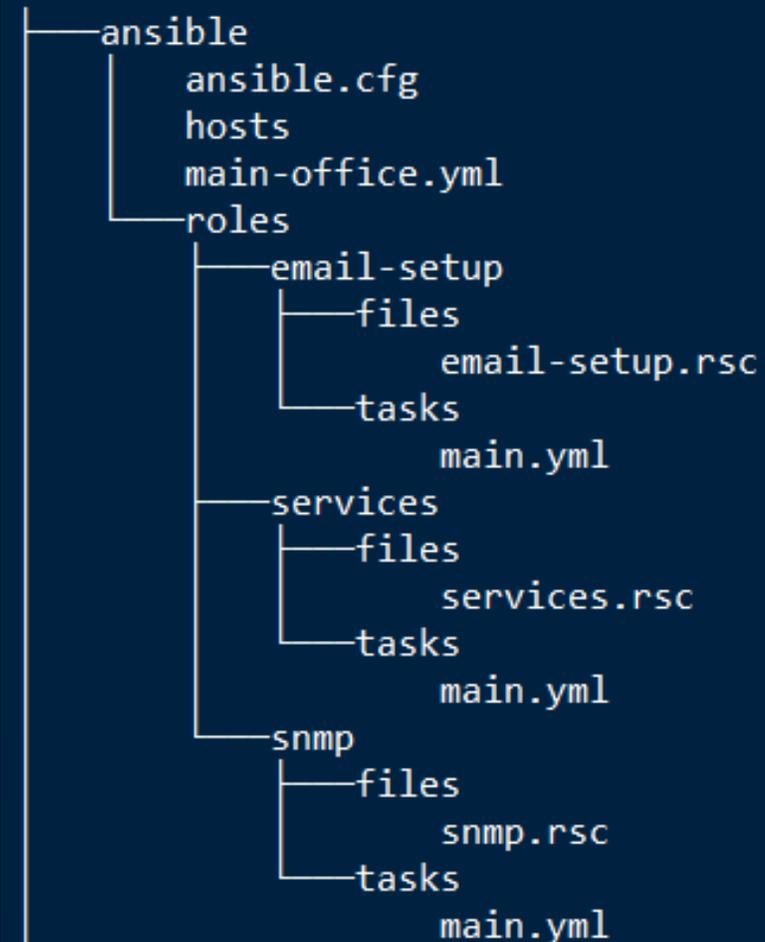
# Массовая перенастройка Mikrotik Router средствами Ansible >

## Как описать роль в Ansible

- Роль в ansible представляет из себя подкаталог внутри каталога ansible/roles
- Название каталога и есть название роли
- Внутри каждой роли должна быть определенная структура подкаталогов

Каталог tasks – содержит файлы, которые исполняет ansible. В них описывается то, что нужно сделать. (main.yml обязателен)

Каталог files – содержит в себе дополнительные файлы. В нашем случае там будут скрипты .rsc, которые ansible будет запускать на RouterOS



# Массовая перенастройка Mikrotik Router средствами Ansible >

## Принципы написания .rsc для ролей Ansible

---

Простота и блочность.

Каждая роль описывает только один аспект конфигурации и не зависит от других ролей.

---

Щадящее использование flash роутера.

Нужно проверять необходимость перенастройки прежде, чем вносить изменения в конфигурацию.

---

Осторожность.

С осторожностью редактировать параметры, смена которых может привести к потере доступа к оборудованию (firewall, interfaces и тп.)

---

# Массовая перенастройка RouterOS средствами Ansible >

## Особенности работы Ansible с RouterOS

---

Штатно Ansible использует `python` на управляемом оборудовании, разумеется, в таком режиме с RouterOS он не работает.

Есть возможность действовать в `raw` режиме - просто исполнять команды на удаленном оборудовании, но нам это не подходит, т.к. часто нужно исполнить множество команд или небольшой скрипт. В `raw` режиме это слишком громоздко и неудобно получается.

# Массовая перенастройка RouterOS средствами Ansible >

## Особенности работы Ansible с RouterOS

---

Мне показалось наиболее удобным использовать обычный SSH клиент и конвейер (pipe) для перенаправления команд\скриптов из текстового файла в SSH

Пример файла roles/snmp/tasks/main.yml

```
---  
- name: Configure SNMP  
  command: bash -c "cat roles/snmp/files/snmp.rsc | ssh -T {{inventory_hostname}} -p {{ansible_ssh_port}}"
```

# Массовая перенастройка RouterOS средствами Ansible > Особенности работы Ansible с RouterOS

---

Пример файла roles/snmp/files/snmp.rsc

```
/snmp set enabled=yes
```

```
/snmp community set [ find default=yes ] addresses=192.168.0.1, 192.168.1.1  
name=ExampleCommunity
```

**В .rsc файле обязательно должна быть пустая строка в конце!**

# Массовая перенастройка RouterOS средствами Ansible >

## Еще пример роли

Пример файла `roles/services/tasks/main.yml`

```
---  
- name: Configure services  
  command: bash -c "cat roles/services/files/services.rsc | ssh -T {{inventory_hostname}} -p {{ansible_ssh_port}}"
```

Пример файла `roles/services/files/services.rsc`

```
:if ([/ip service get telnet disabled] !=true) do={/ip service set telnet disabled=yes}  
:if ([/ip service get ftp disabled] !=true) do={/ip service set ftp disabled=yes}  
:if ([/ip service get api disabled] !=true) do={/ip service set api disabled=yes}  
:if ([/ip service get api-ssl disabled] !=true) do={/ip service set api-ssl disabled=yes}  
:if ([/ip service get www disabled] !=true) do={/ip service set www disabled=yes}  
:if ([/ip service get www-ssl disabled] !=true) do={/ip service set www-ssl disabled=yes}
```

# Массовая перенастройка RouterOS средствами Ansible >

## Что такое playbook?

Файл, который описывает каким хостам (из файла hosts) какие роли применять.

В playbook могут быть указаны дополнительные опции и переменные.

Именно его выполняет Ansible

```
---
- hosts: main-office
  gather_facts: no
  connection: local
  roles:
    - email-setup
    - services
    - snmp

vars:
  some_variable: "value"
```

# Массовая перенастройка RouterOS средствами Ansible >

## Запуск playbook

После того, как нужные роли созданы, playbook составлен, его можно запустить.

**ansible-playbook** имя\_playbook

Можно сделать sh скрипты для удобства запуска (например, если нужно использовать доп. ключи запуска).

```
mikroansible@example:~/ansible$ ansible-playbook main-office.yml

PLAY [main-office] *****

TASK: [email-setup | Email setup] *****
changed: [192.168.10.254]

TASK: [services | Configure services] *****
changed: [192.168.10.254]

TASK: [snmp | Configure SNMP] *****
changed: [192.168.10.254]

PLAY RECAP *****
192.168.10.254      : ok=3  changed=3  unreachable=0  failed=0

mikroansible@example:~/ansible$
```

# Массовая перенастройка RouterOS средствами Ansible > .retry файлы

---

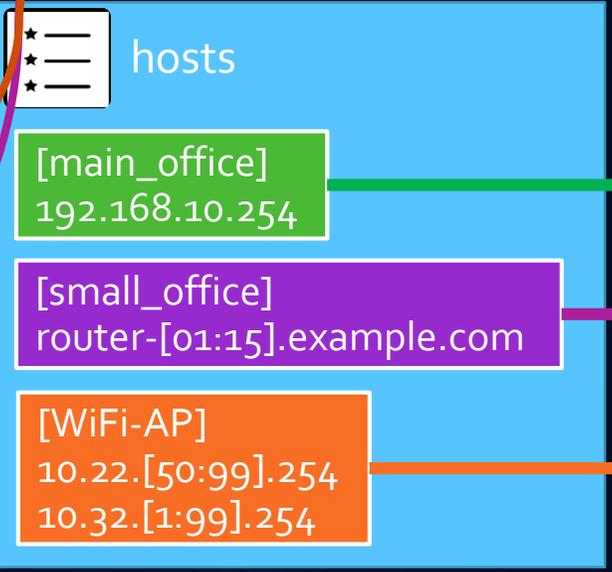
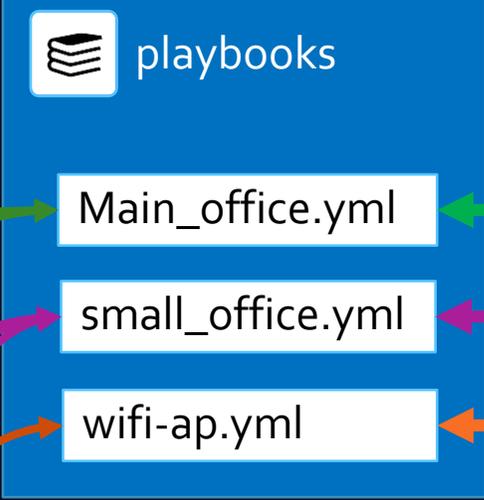
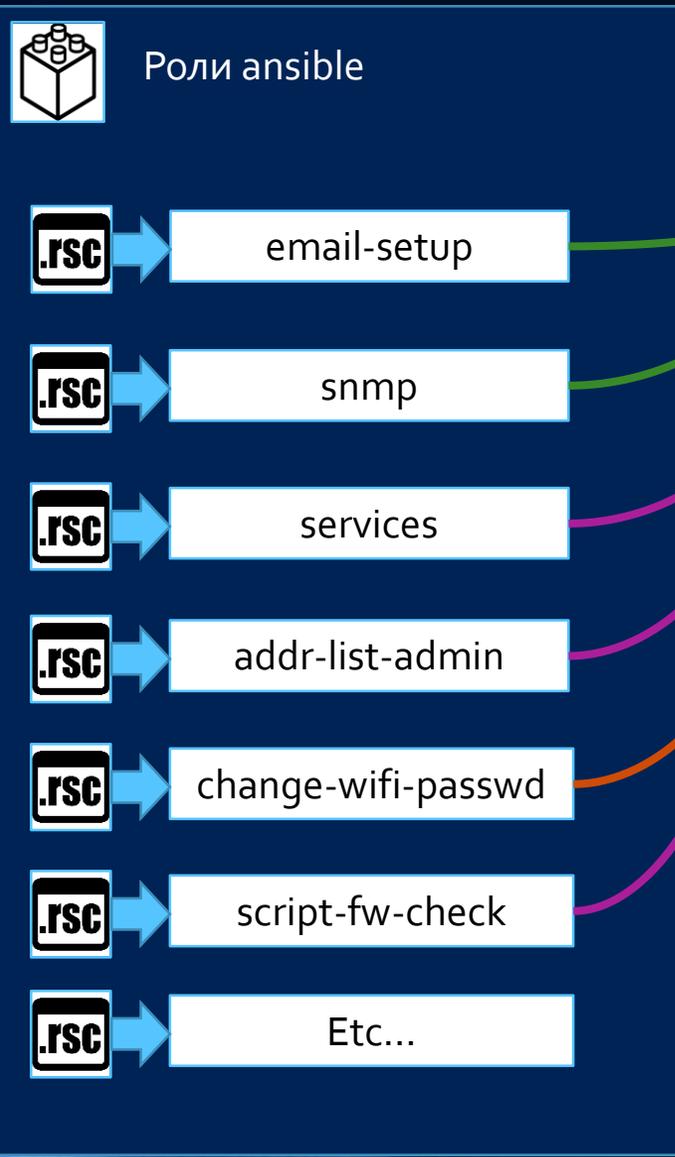
Ansible записывает адреса недоступных или сбойных хостов в файлы

`.ansible-retry/имя_playbook.retry`

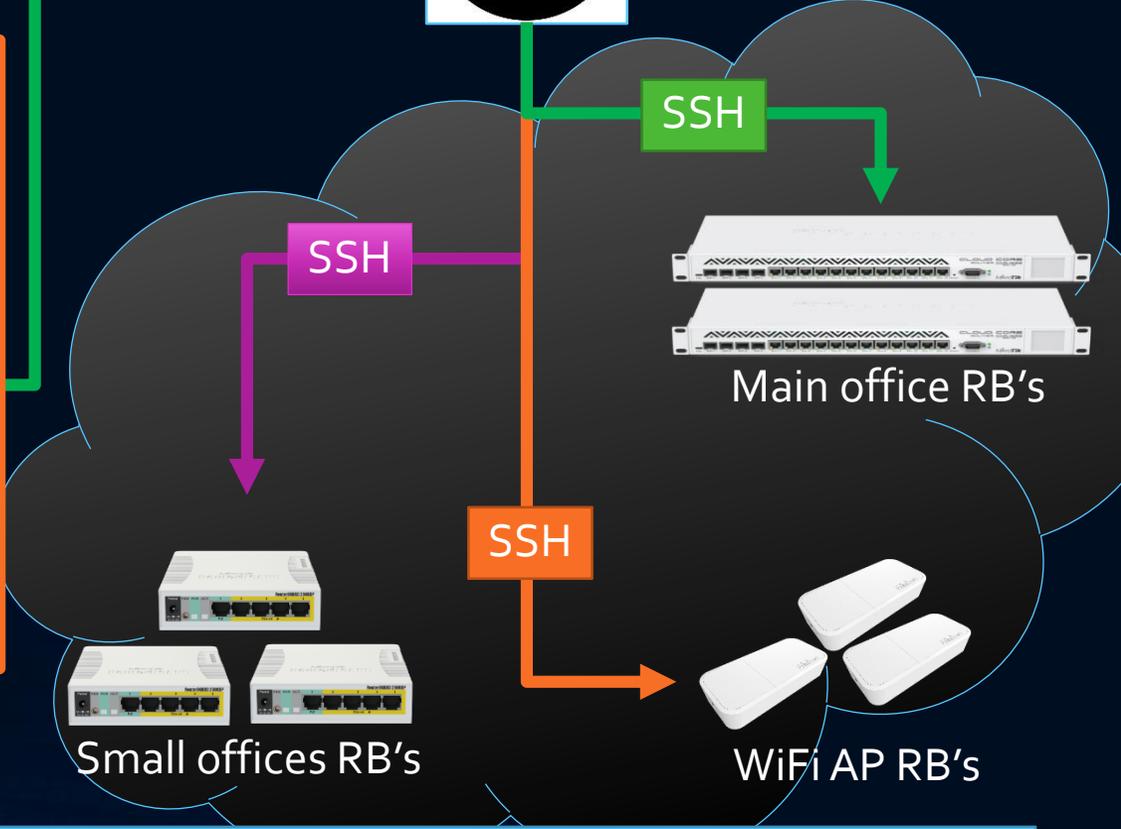
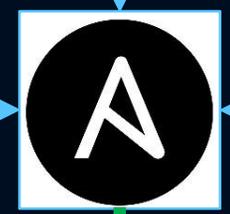
При помощи ключа `--limit` можно ограничить работу Ansible списком хостов из `.retry` файла.

```
ansible-playbook office.yml --limit @/home/mikroansible/ansible/.ansible-retry/office.retry
```

Разумеется, можно добавить это в cron.



переменные      ansible.cfg



# Массовая перенастройка RouterOS средствами Ansible > Внимание!

---

**Проводите тест перед  
глобальным запуском на  
работающем оборудовании.**

Желательно выделить несколько единиц оборудования для тестирования, а не сразу перенастраивать большое кол-во роутеров. В случае ошибки не «ляжет» вся сеть.



# Контроль за конфигурацией RouterOS

# Контроль за конфигурацией RouterOS > О пользе контроля

---

Не всегда есть возможность просто менять конфигурацию на нужную, т.к править firewall на сотне роутеров, например, может быть проблематично.

В такой ситуации лучше написать и распространить средствами Ansible скрипт, который проверяет нужные нам аспекты конфигурации и генерирует оповещение при неверных настройках.

Еще полезно будет добавить такой скрипт в планировщик, чтобы он запускался далее сам без Ansible.

---

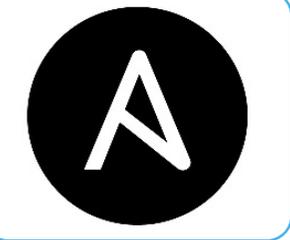
# Контроль за конфигурацией RouterOS >



Пишем скрипт на языке RouterOS, который контролирует нужный аспект конфигурации.

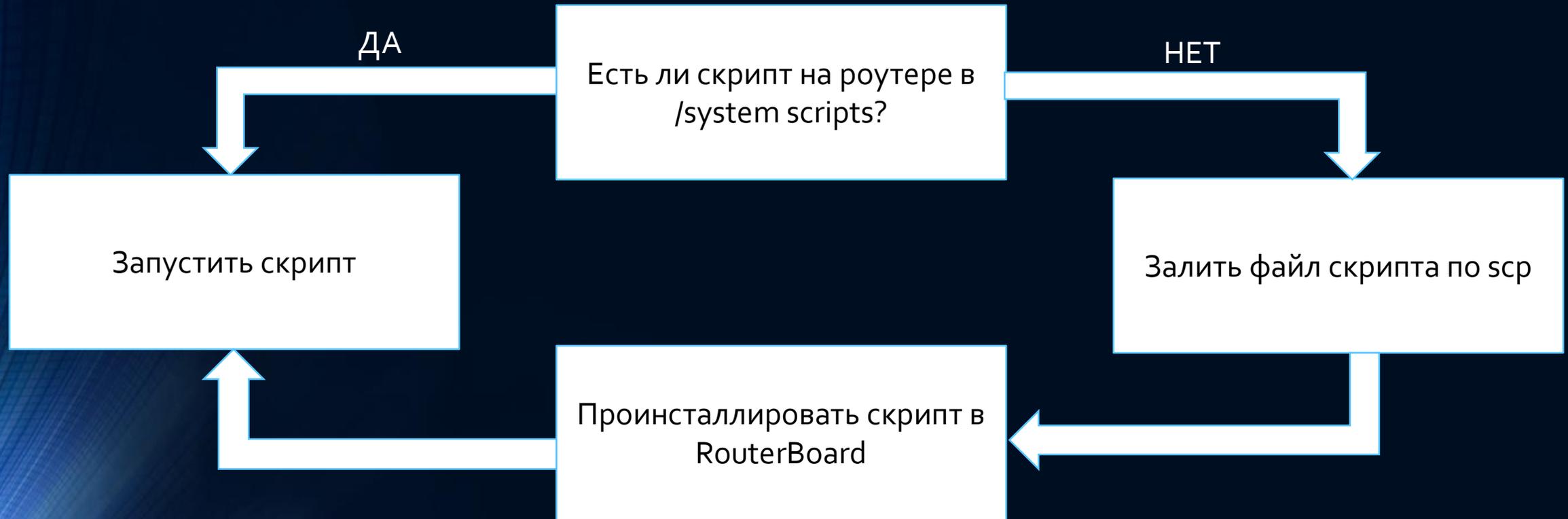


Оповещение о проблемных конфигурациях на email или syslog средствами скрипта



Распространяем, устанавливаем и запускаем скрипт через Ansible.

# Контроль за конфигурацией RouterOS > Пример алгоритма загрузки и запуска скрипта



# Контроль за конфигурацией RouterOS > Пример

## Пример файла roles/scrip-route-distance-check/tasks/main.yml

```
---
- name: Check for script exists
  command: bash -c "cat roles/scrip-route-distance-check/files/check.rsc | ssh -T {{inventory_hostname}} -p {{ansible_ssh_port}}"
  register: RouteScriptExists

- name: if script not exist upload it
  command: "scp -P {{ansible_ssh_port}} roles/scrip-route-distance-check/files/RouteCheck.rsc {{ansible_ssh_user}}@{{inventory_hostname}}:/RouteCheck.rsc"
  when: RouteScriptExists.stdout != "EXISTS"

- name: if script not exist install it
  command: "ssh -T -p {{ansible_ssh_port}} {{ansible_ssh_user}}@{{inventory_hostname}} '/import file=RouteCheck.rsc'"
  when: RouteScriptExists.stdout != "EXISTS"

- name: Run Route check script if it already exists
  command: "ssh -T -p {{ansible_ssh_port}} {{ansible_ssh_user}}@{{inventory_hostname}} '/system script run RouteCheck'"
  when: RouteScriptExists.stdout == "EXISTS"
```

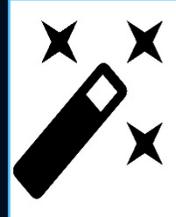
## Пример файла roles/scrip-route-distance-check/files/check.rsc

```
:if ([/system script find name=RouteCheck] != "") do={:put "EXISTS"}
```

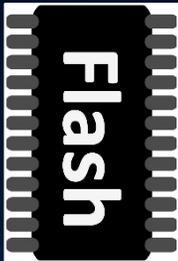
# Контроль за конфигурацией RouterOS > Какие задачи по контролю можно решать?



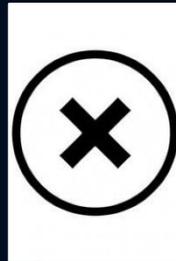
Контроль  
корректности  
настройки Firewall



Автоматическое  
назначение dhcp static  
по hostname



Мониторинг  
состояния flash  
роутера



Отлов известных  
ошибок в  
конфигурации



На что фантазии  
хватит ;)

# **Автоматизация настройки типового оборудования**

# Автоматизация настройки типового оборудования >

## Настройка оборудования по шаблону

Ansible дает возможность подготовить шаблон конфигурации роутера, а далее генерировать конфиг роутера на основании шаблона и переменных.

Шаблон представляет из себя обычный файл конфигурации RouterOS, но с подстановкой переменных там, где это необходимо.

Переменная должна быть заключена в двойные фигурные скобки.

Значение для переменной можно задать разными способами:

- Файл `role/defaults/main.yml`
- Файл `role/vars/main.yml`
- Индивидуально для хоста в `host_vars/_hostname_`
- Для группы хостов в `group_vars/_groupname_`
- Взять из `playbook`
- Взять из `hosts`
- Передать при запуске `ansible-playbook`
- etc

### Файл с переменными

```
---
ssid: "exampleNet"
key: "SuperSectet"
frequency: "2437"
txpower: "14"
```

### Фрагмент шаблона

```
/interface wireless security-profiles
add authentication-types=wpa2-psk mode=dynamic-keys name="{{ssid}}" \
    wpa2-pre-shared-key="{{key}}"

/interface wireless
set [ find default-name=wlan1 ] band=2ghz-b country=russia disabled=no \
    distance=indoors frequency="{{frequency}}" frequency-mode=regulatory-
domain \
    keepalive-frames=disabled mode=ap-bridge security-profile="{{ssid}}" ssid=\
    "{{ssid}}" tx-power="{{txpower}}" tx-power-mode=all-rates-fixed
```

# Автоматизация настройки типового оборудования >

## Настройка оборудования по шаблону

Для написания шаблонов используется язык шаблонов jinja2

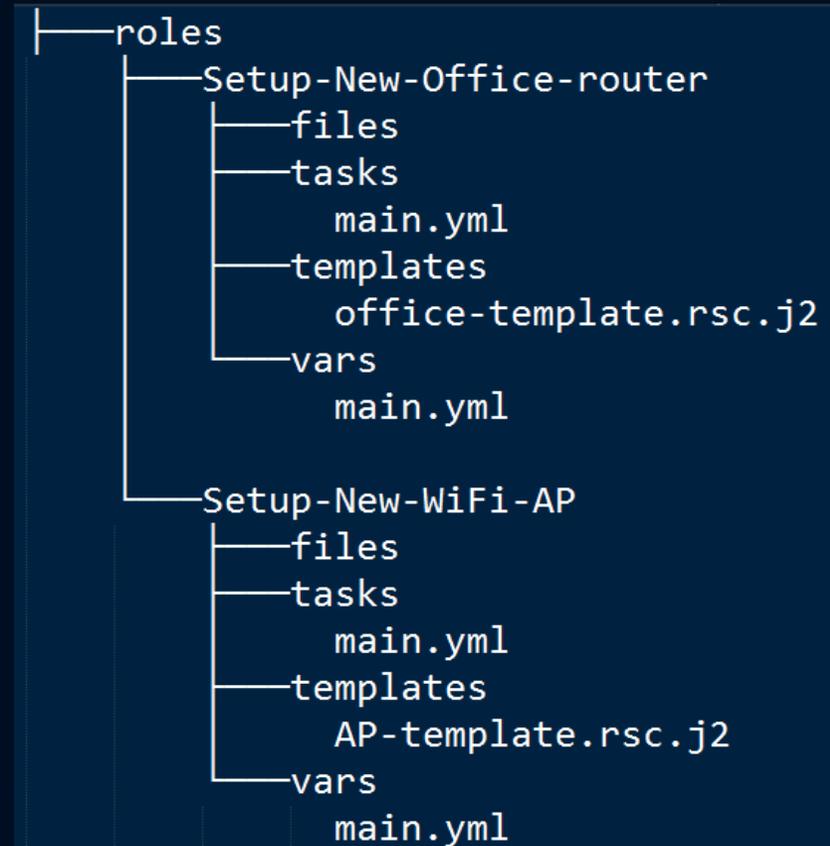
Шаблоны должны располагаться в каталоге templates внутри каталога роли.

И иметь имя \*.j2

Ссылки по теме

[http://docs.ansible.com/ansible/template\\_module.html](http://docs.ansible.com/ansible/template_module.html)

[http://docs.ansible.com/ansible/playbooks\\_variables.html](http://docs.ansible.com/ansible/playbooks_variables.html)



# Автоматизация настройки типового оборудования > Процесс настройки нового роутера по шаблону

Генерируем конфигурацию



Заливаем пакеты на настраиваемый роутер



Заливаем конфиг на роутер



Делаем reset конфигурации с применением нового конфига.

# Автоматизация настройки типового оборудования > Пример роли. Файл roles/New-WiFi-AP-setup/tasks/main.yml

```
---  
- name: Generate AP config  
  template: src=wifi-template.rsc.j2 dest=roles/New-WiFi-AP-setup/files/tmp/{{inventory_hostname}}.rsc  
  
- name: Upload packages to router  
  command: "sshpass -p " scp -r -o StrictHostKeyChecking=no -o NumberOfPasswordPrompts=1 {{ item }}  
admin@{{inventory_hostname}}:/"  
  with_fileglob:  
    - roles/New-WiFi-AP-setup/files/pkg/*.*.npk  
  
- name: Upload config to router  
  command: "sshpass -p " scp -o StrictHostKeyChecking=no -o NumberOfPasswordPrompts=1 roles/New-WiFi-AP-  
setup/files/tmp/{{inventory_hostname}}.rsc admin@{{inventory_hostname}}:/setup.rsc"  
  
- name: Reset router and apply new config  
  command: bash -c "cat roles/New-WiFi-AP-setup/files/reset_and_reboot.rsc | sshpass -p " ssh -T -o  
NumberOfPasswordPrompts=1 admin@{{inventory_hostname}} -o StrictHostKeyChecking=no"
```

# Автоматизация настройки типового оборудования >

## Пример роли

---

Пример файла `roles/New-WiFi-AP-setup/files/reset_and_reboot.rsc`

```
:delay 5
/system reset-configuration no-defaults=yes skip-backup=yes run-after-reset="setup.rsc"
y
```

**Дополнительная информация**

# Дополнительная информация >

## Возможные проблемы с ssh соединением

---

Может возникнуть ситуация, когда оборудование приняло соединение и подвисло в этом состоянии. Команды не выполняет, но и ssh соединение не рвет.

Причины различны - от проблем с сетью до подвисаний самой RouterOS (увы и такое иногда случается ;-)

Все это может привести к «зависанию» процесса перенастройки оборудования. Ansible будет ждать ответа от управляемого хоста... Долго и безрезультатно

Минимизировать данную проблему можно путем настройки ssh клиента на сервере с Ansible

# Дополнительная информация >

## Настройка ssh клиента

### Пример файла ~/.ssh/config

```
#
Host *
  ServerAliveInterval 4
  ServerAliveCountMax 5
  ConnectTimeout 10
  NumberOfPasswordPrompts 0
  StrictHostKeyChecking yes
```

Так настроить ssh проще, чем писать ключи запуска ssh клиента в каждой роли

`ServerAliveInterval` – интервал в секундах между отправкой на сервер (роутер) alive-пакетов. Предотвращает обрыв соединения из-за неактивности.

`ServerAliveCountMax` – число неудачных попыток обмена alive-пакетами с сервером, после которых соединение будет разорвано

`ConnectTimeout` – время ожидания подключения к серверу ssh

`NumberOfPasswordPrompts` - число попыток ввода пароля.

`StrictHostKeyChecking` – проверка подлинности ssh сервера.

Есть и другие опции

`man ssh_config`

# Дополнительная информация > Отлов ошибок подключения

Может возникнуть ситуация, когда у роутера поменялся ключ ssh сервера (например, оборудование заменили)

Или по какой-то причине учетные данные Ansible не подходят.

В такой ситуации роутер никогда не будет настроен автоматически.

Нужно отлавливать такие ошибки и исправлять их.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now (man-in-the-middle attack)!

It is also possible that a host key has just been changed.

The fingerprint for the DSA key sent by the remote host is
a7:8d:27:ff:09:9d:a5:fe:87:e6:42:46:65:91:3f:63.

Please contact your system administrator.

Add correct host key in /home/mikroansible/.ssh/known_hosts to get rid of this message.
Offending DSA key in /home/mikroansible/.ssh/known_hosts:320
remove with: ssh-keygen -f "/home/mikroansible/.ssh/known_hosts" -R 10.27.54.254

Password authentication is disabled to avoid man-in-the-middle attacks.

Keyboard-interactive authentication is disabled to avoid man-in-the-middle attacks.
```

```
Received disconnect from 10.32.56.254
```

Спасибо за внимание!

Вопросы?