

Масово програмиране на Mikrotik устройства

Съдържание

- За мен
- Python и SSH
 - Особенности при комуникация през ssh
 - ssh с име и парола или ssh със сертификат
 - Изпълнение на последователни команди през ssh
 - Управление през ssh
 - Управление през API
 - Управление през REST API
- ANSIBLE
- Semaphore

За мен

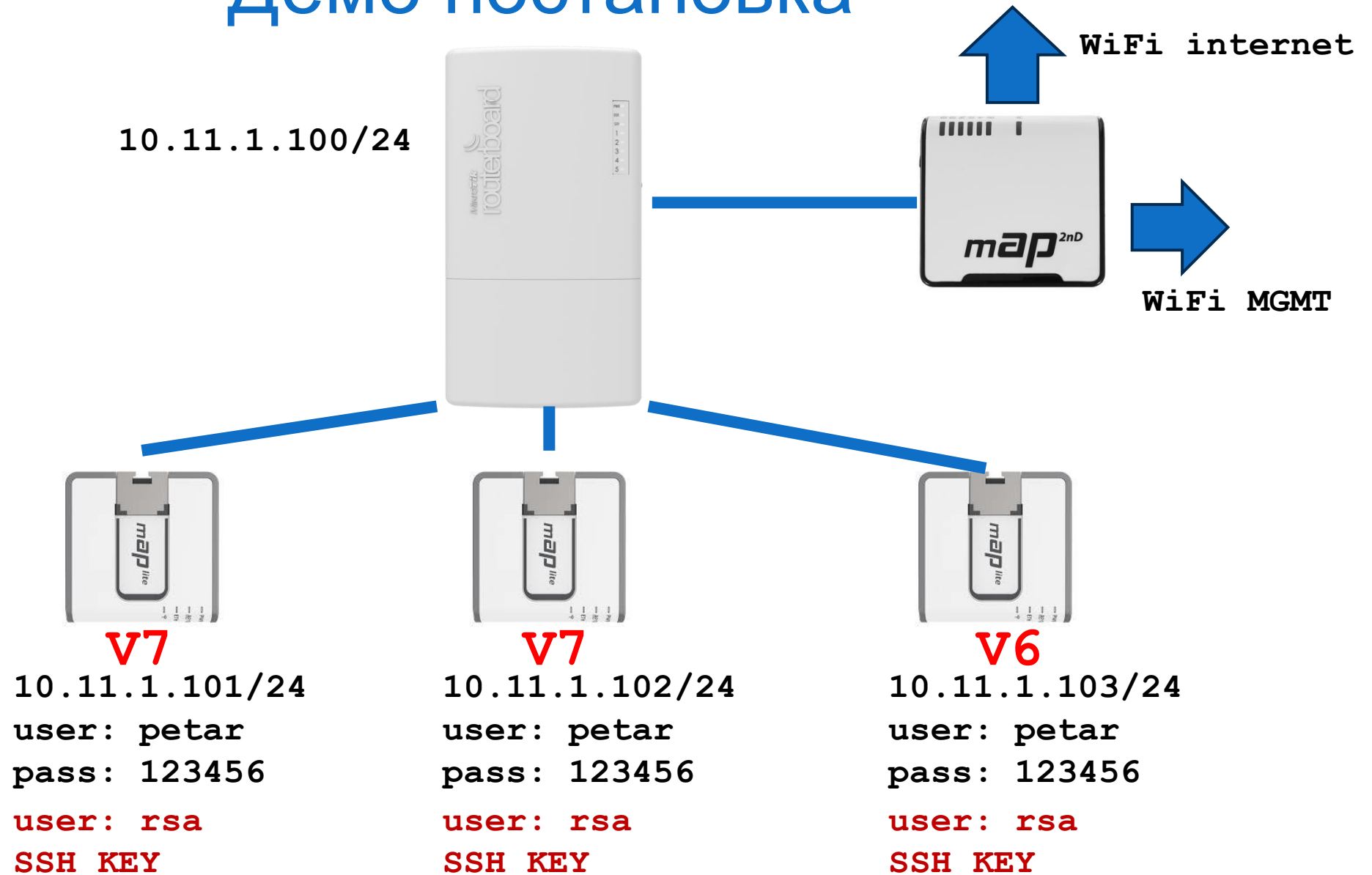
- Петър Цонев
- Занимавам се с мрежи и комуникации от 2006
- В отбора на Mikrotik от 2014
- CCNA, CCNP, CCNA SECURITY, MTCNA, MTCWE, MTCRE, MTCTCE, MTCUME, MTCIPv6E, MTCINE, MTCWE, MTCWE, MTCWE
- СЕН
- Булгартрансгаз ЕАД

**Може ли с
МИКРОТИК да се
направи ...?**

**Не само че го може, но
го може по няколко
начина**

Мрежова ТОПОЛОГИЯ

Демо постановка



PYTHON + Paramiko



SSH Достъп user/pass

```
import paramiko
ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh.connect(hostname='10.11.1.101',username='petar',password='123456',
look_for_keys=False)
```


SSH Достъп със сертификат

```
ssh-keygen -t rsa -b 2048 -C rsa
```

```
petar@ubuntu:~/python/mikrotik$ ll ~/.ssh/  
total 24  
drwx-----  2 petar petar 4096 апр  6 15:18 ./  
drwxr-x--- 21 petar petar 4096 апр 11 14:22 ../  
-rw-----  1 petar petar 1823 апр  6 15:18 id_rsa  
-rw-r--r--  1 petar petar  394 апр  6 15:18 id_rsa.pub  
-rw-r--r--  1 petar petar 1326 апр  7 09:35 known_hosts  
-rw-r--r--  1 petar petar  884 мар 12 19:49 known_hosts.original
```

```
>>> import paramiko  
>>> ssh = paramiko.SSHClient()  
>>> ssh.load_system_host_keys()  
>>> ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())  
>>> ssh.connect(hostname =  
'10.11.1.101', username='rsa', key_filename='/home/petar/.ssh/id_rsa')  
>>>
```

SSH Достъп със сертификат

Разлика между ROS V6 и V7

```
>>> ssh.connect(hostname =
'10.11.1.101',username='rsa',key_filename='/home/petar/.ssh/id_rsa')
>>>
>>> ssh.connect(hostname =
'10.11.1.103',username='rsa',key_filename='/home/petar/.ssh/id_rsa')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/python3/dist-packages/paramiko/client.py", line 435, in connect
    self._auth(
  File "/usr/lib/python3/dist-packages/paramiko/client.py", line 766, in _auth
    raise saved_exception
  File "/usr/lib/python3/dist-packages/paramiko/client.py", line 742, in _auth
    self._transport.auth_publickey(username, key)
  File "/usr/lib/python3/dist-packages/paramiko/transport.py", line 1624, in
auth_publickey
    raise SSHException("No existing session")
paramiko.ssh_exception.SSHException: No existing session
```

Особенности
при работа
с PYTHON и
SSH

Изпълнение на единични команди: Редактиране на System Note

```
/system note set note="Hello from NETCAMP 2023" show-at-login=yes
```

```
import paramiko
ssh = paramiko.SSHClient()
ssh.load_system_host_keys()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh.connect(hostname =
'10.11.1.101', username='petar',
password='123456', allow_agent=False,
look_for_keys=False, timeout=10)

ssh.exec_command ('/system note set note="Hello
from NETCAMP 2023" show-at-login=yes')
```

SSH достъп

Изпълнение на команди с извеждане на резултат:

```
>>> stdin, stdout, stderr = ssh.exec_command('ip  
address print')  
>>> print(stdout.read().decode('ascii'))  
Columns: ADDRESS, NETWORK, INTERFACE  
# ADDRESS          NETWORK          INTERFACE  
0 10.11.1.101/24    10.11.1.0       ether1
```

Изпълнение на последователни команди

ВАРИАНТ 1:

Добавяне на 2 правила във firewall

```
/ip/firewall/filter/ add chain=input action=accept  
/ip/firewall/filter/ add chain=output action=accept
```

```
import paramiko  
ssh = paramiko.SSHClient()  
ssh.load_system_host_keys()  
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())  
ssh.connect(hostname = '10.11.1.101', username='petar', password='123456',  
allow_agent=False, look_for_keys=False, timeout=10)  
  
ssh.exec_command ('/ip/firewall/filter/ add chain=input action=accept')  
ssh.exec_command ('/ip/firewall/filter/ add chain=output action=accept')
```

Изпълнение на последователни команди

ВАРИАНТ 2:

Добавяне на 2 правила във firewall

```
/ip/firewall/filter/  
add chain=input action=accept  
add chain=output action=accept
```

```
import paramiko  
ssh = paramiko.SSHClient()  
ssh.load_system_host_keys()  
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())  
ssh.connect(hostname='10.11.1.101', username='petar', password='123456',  
allow_agent=False, look_for_keys=False, timeout=10)
```

```
ssh.exec_command('/ip/firewall/filter/')  
ssh.exec_command('add chain=input action=accept')  
ssh.exec_command('add chain=output action=accept')
```

Изпълнение на последователни команди

ВАРИАНТ 3 – изпълняване на 1 ред:

Добавяне на 2 правила във firewall

```
/ip/firewall/filter/  
add chain=input action=accept  
add chain=output action=accept
```

```
import paramiko  
ssh = paramiko.SSHClient()  
ssh.load_system_host_keys()  
ssh.connect(hostname = '10.11.1.101', username='petar', password='123456',  
allow_agent=False, look_for_keys=False, timeout=10)  
  
ssh.exec_command ('/ip/firewall/filter/;add chain=input action=accept;add  
chain=output action=accept')
```


Изпълнение на по-сложни команди

Особености при отваряне на ssh сесия: пример с *place-before*

```
ssh.exec_command('/ip firewall filter add action=accept chain=forward  
dst-address=1.1.1.1 src-address=1.1.1.1 place-before=0')  
ssh.exec_command('/ip firewall filter add action=accept chain=forward  
dst-address=1.1.1.2 src-address=1.1.1.2 place-before=0')  
ssh.exec_command('/ip firewall filter add action=accept chain=forward  
dst-address=1.1.1.3 src-address=1.1.1.3 place-before=0')  
  
ssh.exec_command('/ip firewall filter print')  
ssh.exec_command('/ip firewall filter add action=accept chain=forward  
dst-address=1.1.1.4 src-address=1.1.1.4 place-before=0')
```

MIKROTIK

API

APY.PY

<https://help.mikrotik.com/docs/display/ROS/Python3+Example>

```
python3 api.py 10.11.1.101 petar 123456
```

```
/ip/address/print
<<< /ip/address/print
<<<
>>> !re
>>> =.id=*2
>>> =address=10.11.1.101/24
>>> =network=10.11.1.0
>>> =interface=ether1
>>> =actual-interface=ether1
>>> =invalid=false
>>> =dynamic=false
>>> =disabled=false
>>>
>>> !done
>>>
```

Библиотека `ros_api`

https://github.com/LaiArturs/RouterOS_API

```
import ros_api
router = ros_api.Api('10.11.1.101',
user='petar', password='123456', port=8728)
r=router.talk('/ip/address/print')
print(r)
[{'id': '*2', 'address': '10.11.1.101/24',
'network': '10.11.1.0', 'interface': 'ether1',
'actual-interface': 'ether1', 'invalid':
'false', 'dynamic': 'false', 'disabled':
'false'}]
```

ros_api

https://github.com/LaiArturs/RouterOS_API

```
r=router.talk(['/system/note/set',  
'=note=Hello from API'])
```

```
r=router.talk(['/system/note/set',  
'=note=Hello from API', '=show-at-login=yes'])
```

Проблеми с arі:

ВНИМАНИЕ! Какъв ще е резултатът от следващата команда?

```
>>> r=router.talk([' /ip/firewall/filter/add',  
'=chain=input', '=action=accept', '=place-  
before=0', '=comment="MY TOP RULE" '])
```

- Море от библиотеки за python – всяка с различен синтаксис
- В командите (без print) няма find

MIKROTIK

REST API

<https://help.mikrotik.com/docs/display/ROS/REST+API>

MIKROTIK REST API

http, https

get, put, delete, post, patch

HTTP	ROS	Описание
DELETE	remove	Изтриване на запис
GET	print	Извличане на записи
PATCH	set	Update на запис
POST		Универсална команда
PUT	add	Добавяне на запис

mktssh_uni.py

```
#####  
# Program for mass config Mikrotik routers #  
# usage: python3 mktssh_uni.py -i <ip address file>#  
# -s <command file> #  
# FOR HELP: python3 mktssh_uni.py -h #  
#-----#  
# Petar Tsonev: petar.gsm@gmail.com #  
#####
```

config.cfg

```
savelogfile=TRUE
logfile=/home/petar/python/mikrotik/log/
execType=ssh_1row
#execType=ssh
#execType=rsc
#execType=api
#execType=api_rsc
# 0 - no debug
# 1 - simple debug
# 2 - debug info messages
# 3 - debug all
debug=0
#key_filename=/home/petar/.ssh/id_rsa
username=petar
password=123456
```

netcamp_ip.txt

```
10.11.1.101  
10.11.1.102  
10.11.1.103
```

setnote.rsc

```
#####  
# Set systm note  
#####  
/system note set note="Hello from NETCAMP 2023" show-at-  
login=yes
```

Редактиране на System Note

```
python3 mktssh_uni.py -i ipaddress/netcamp_ip.txt -s  
scripts/setnote.rsc
```

Start program

Send to IP	10.11.1.101	1 of 3...	OK!
Send to IP	10.11.1.102	2 of 3...	OK!
Send to IP	10.11.1.103	3 of 3...	OK!

Четене на System Note

```
/system note print
```

```
python3 mktssh_uni.py -i ipaddress/netcamp_ip.txt -s  
scripts/getnote.rsc
```

```
cat log/getnote.txt_2023-07-14_152110
```

```
#####
```

```
IP address 10.11.1.101:OK! show-at-login: yes  
note: Hello from NETCAMP 2023
```

```
IP address 10.11.1.102:OK! show-at-login: yes  
note: Hello from NETCAMP 2023
```

```
IP address 10.11.1.103:OK! show-at-login: yes  
note: Hello from NETCAMP 2023
```

Добавяне на 1 правило във RAW

```
/ip firewall raw
remove [find comment ="Enable ICMP type 3 code 0"]
:if ([:len [/ip firewall raw find]] = 0 ) do={
    add action=accept chain=prerouting icmp-options=3:1
protocol=icmp comment="Enable ICMP type 3 code 0"
} else={
    add action=accept chain=prerouting icmp-options=3:1
protocol=icmp place-before=0 comment="Enable ICMP type 3 code
0"
}
```

Start program

Send to IP	10.11.1.101	1 of 3...	OK!
Send to IP	10.11.1.102	2 of 3...	OK!
Send to IP	10.11.1.103	3 of 3...	OK!

Резултат при липса на комуникация

```
ipaddress/netcamp_ip_104.txt
```

```
10.11.1.101
```

```
10.11.1.102
```

```
10.11.1.103
```

```
10.11.1.104
```

```
Send to IP      10.11.1.101      1 of 4...      OK!
```

```
Send to IP      10.11.1.102      2 of 4...      OK!
```

```
Send to IP      10.11.1.103      3 of 4...      OK!
```

```
Send to IP      10.11.1.104      4 of 4...
```

```
Authentication error for 10.11.1.104 or other communication error
```

```
[Errno None] Unable to connect to port 22 on 10.11.1.104
```

```
log/IPError__2023-08-21_174145
```

```
10.11.1.104
```



ANSIBLE



Red Hat



Red Hat
Ansible Automation
Platform



ANSIBLE
Community
edition

A Documentation	
System modules	riak – This module handles some common Riak operations
Utilities modules	rocketchat – Send notifications to Rocket Chat
Web Infrastructure modules	rollbar_deployment – Notify Rollbar about app deployments
Windows modules	route53 – add or delete entries in Amazons Route53 DNS service
Playbook Keywords	route53_facts – Retrieves route53 details using AWS methods
Ansible Galaxy	route53_health_check – add or delete health-checks in Amazons Route53 DNS service
Return Values	
Ansible Configuration Settings	
Controlling how Ansible behaves: precedence rules	

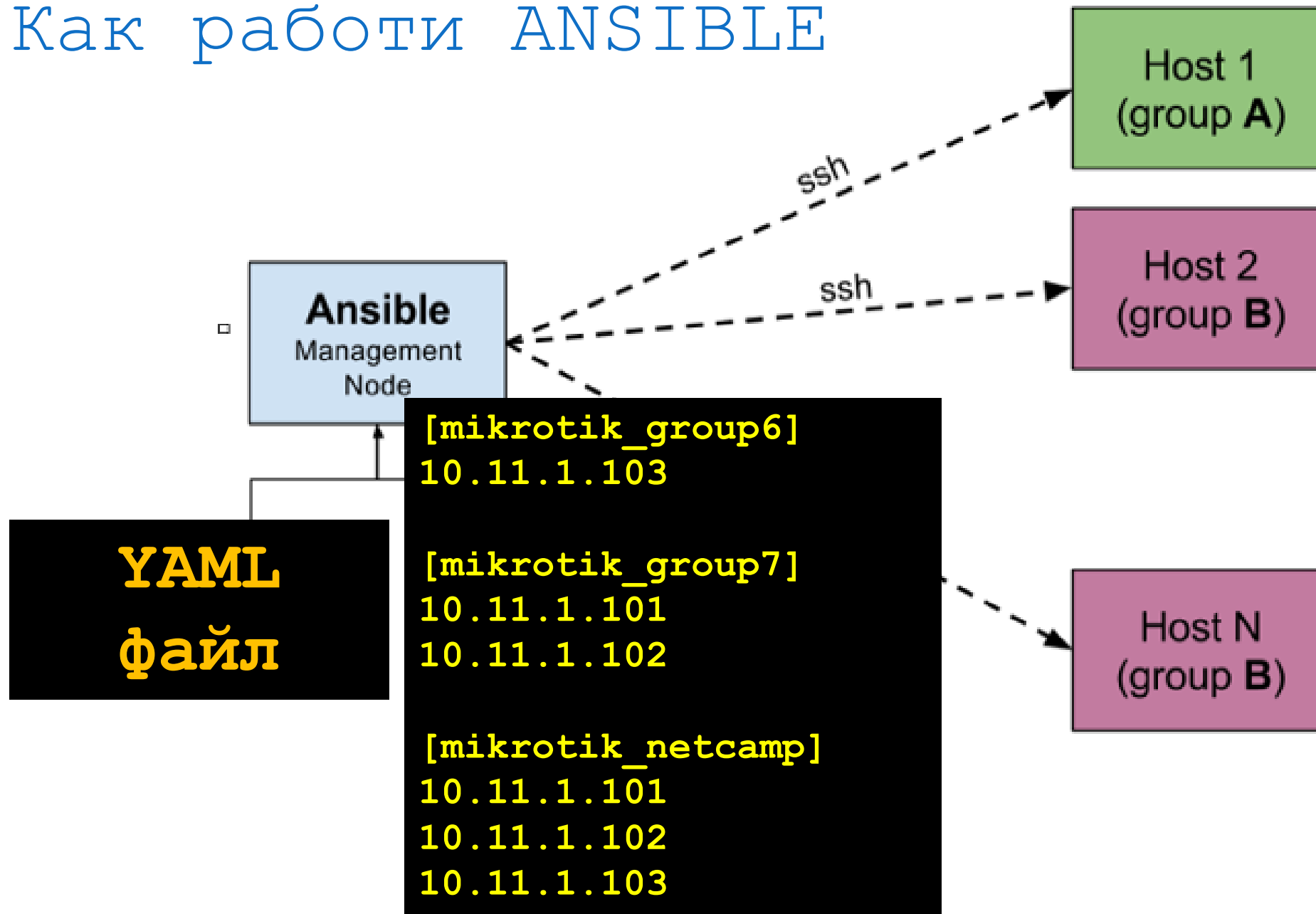
[routeros_command](#) – Run commands on remote devices running MikroTik RouterOS

[routeros_facts](#) – Collect facts from remote devices running MikroTik RouterOS

Special Variables	runit – Manage runit services
Red Hat Ansible Tower	s3_bucket – Manage S3 buckets in AWS, DigitalOcean, Ceph, Walrus and FakeS3
Roadmaps	s3_lifecycle – Manage s3 bucket lifecycle rules in AWS
Ansible Roadmap	s3_logging – Manage logging facility of an s3 bucket in AWS
Red Hat Ansible Automation Platform	s3_sync – Efficiently upload multiple files to S3

2833 модула за различни системи

Как работи ANSIBLE



YAML файл:

```
---  
- hosts: mikrotik_netcamp  
  connection: network_cli  
  gather_facts: no  
  
vars:  
  ansible_network_os: routeros  
  ansible_user: petar  
  ansible_password: 123456  
  
tasks:  
  - name: Set system note  
    routeros_command:  
      commands:  
        - /system note set note="Hello note" show-at-login=no
```

Изпълнение на файл:

```
ansible-playbook set_hello_note.yaml
```

```
PLAY [mikrotik_netcamp]
```

```
*****
```

```
TASK [Set system note]
```

```
*****
```

```
ok: [10.11.1.101]
```

```
ok: [10.11.1.102]
```

```
ok: [10.11.1.103]
```

```
PLAY RECAP
```

```
*****
```

```
10.11.1.101: ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

```
10.11.1.102: ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

```
10.11.1.103: ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

SEMAPHORE

SEMAPHORE

Username

admin

Password

●●●●●●

SIGN IN

[Don't have account or can't sign in?](#)

SEMAPHORE конфигуриране

The screenshot displays the Semaphore web interface. On the left is a dark sidebar with navigation items: Netcamp (selected), Dashboard, Task Templates, Inventory, Environment, Key Store, Repositories, and Team. The main content area is titled 'Key Store' and shows a table with one entry named 'petar'. A 'NEW KEY' button is in the top right. A 'New Key' modal dialog is open, featuring a 'Key Name' input field and a 'Type' dropdown menu with options: SSH Key, Login with password, and None. The table has an 'Actions' column with delete and edit icons.

Name	Actions
petar	

New Key

Key Name

Type

- SSH Key
- Login with password
- None

SEMAPHORE конфигуриране

The screenshot shows the Semaphore web interface. On the left is a sidebar with navigation items: Netcamp (selected), Dashboard, Task Templates, Inventory, Environment, Key Store, Repositories, and Team. The main content area is titled 'Inventory' and contains a table with columns: Name, Type, Path, and Actions. A 'NEW INVENTORY' button is in the top right. A modal window titled 'New Inventory' is open in the center, showing the following fields:

- Name:
- User Credentials:
- Sudo Credentials (Optional):
- Type:

Name	Type	Path	Actions
Ansible inventory	file	/etc/ansible/hosts	
Semaphore inventory	static	-	

SEMAPHORE конфигуриране

The screenshot displays the Semaphore web interface. On the left is a sidebar with navigation items: Netcamp, Dashboard, Task Templates, Inventory, Environment, Key Store, Repositories (selected), and Team. The main area shows a table of repositories with columns for Name, Git URL, SSH Key, and Actions. A modal window titled 'Edit Repository' is open in the center, showing the configuration for 'Netcamp REPO'. The 'URL or path' field is highlighted with a red circle and contains the text '/home/petar/semaphore/repo'. Below this field are radio buttons for 'git' (selected), 'ssh', 'https', 'file', and 'local: abs. path'. Other fields include 'Branch' (empty), 'Access Key' (set to 'petar'), and 'CANCEL' and 'SAVE' buttons at the bottom.

Name	Git URL	SSH Key	Actions
Netcamp REPO	/home/petar/semaphore/repo	petar	

Edit Repository

Name
Netcamp REPO

URL or path
/home/petar/semaphore/repo

git: ssh https file git local: abs. path

Branch
.....

Access Key
petar

CANCEL SAVE

SEMAPHORE конфигуриране

The screenshot displays the Semaphore web interface. On the left is a dark sidebar with navigation items: Netcamp (selected), Dashboard, Task Templates, Inventory, Environment (highlighted), Key Store, Repositories, and Team. At the bottom of the sidebar is a 'Dark Mode' toggle. The main content area shows an 'Environment' table with two rows: 'Empty env' and 'PetarPasswordRepo'. A modal window titled 'Edit Environment' is open over the 'PetarPasswordRepo' row. The modal contains the following fields and content:

- Environment Name:** PetarPasswordRepo
- Extra variables:** A code editor with the following JSON:

```
1 {
2   "ansible_network_os": "routeros",
3   "ansible_user": "petar",
4   "ansible_password": "123456"
5 }
```
- Environment variables:** A text input field with the placeholder text 'Enter env JSON...'
- Message:** An information icon followed by the text: 'Environment and extra variables must be valid JSON. Example: { "var_available_in_playbook_1": 1245, "var_available_in_playbook_2": "test" }'
- Buttons:** 'CANCEL' and 'SAVE' at the bottom right.

In the background, the 'Environment' table has an 'Actions' column with trash and edit icons for each row. A 'NEW ENVIRONMENT' button is located in the top right corner of the main content area.

SEMAPHORE конфигуриране

The screenshot displays the Semaphore CI web interface. On the left is a navigation sidebar with 'Task Templates' selected. The main area shows the configuration for a task named 'Set system note'. A modal window titled 'Edit Template' is open, showing fields for Name, Description, Playbook Filename, Inventory, Repository, Environment, and Vault Password. A 'Cron' field is also present with a tooltip. At the bottom of the modal are 'CANCEL' and 'SAVE' buttons. In the background, a table lists task templates with columns for Task ID, Version, Duration, and Actions.

Task ID	Version	Duration	Actions
#48	—	a few seconds	RERUN
#47	—	a few seconds	RERUN
#45	—	a few seconds	RERUN
#44	—	a few seconds	RERUN
#43	—	a few seconds	RERUN

Edit Template

TASK BUILD DEPLOY

Name *
Set system note

Description

Playbook Filename *
semaphore_set_hello_note.yaml

Inventory *
Ansible inventory

Repository *
Netcamp REPO

Environment *
PetarPasswordRepo

Vault Password
petar

Survey Variables
+ Add variable

View

Cron
I want to run a task by the cron only for for new commits of some repository
Read the [docs](#) to learn more about Cron.

Suppress success alerts


```
1 CLI Args (JSON array). Example:  
[  
  "-i",  
  "@myinventory.sh",  
  "--private-key=/there/id_rsa",  
  "-vvvv"  
]
```

Allow CLI args in Task

CANCEL SAVE

SEMAPHORE Изпълнение на команди

```
Set system note Semaphore repo v7 > Task #54 ×
```

 Success	Author petar	Started a few seconds ago	Duration a few seconds
---	-----------------	------------------------------	---------------------------

```
2:20:05 PM Task 54 added to queue
2:20:09 PM Preparing: 54
2:20:09 PM Prepare TaskRunner with template: Set system note Semaphore repo v7
2:20:09 PM installing static inventory
2:20:09 PM No collections/requirements.yml file found. Skip galaxy install process.
2:20:09 PM No roles/requirements.yml file found. Skip galaxy install process.
2:20:14 PM Started: 54
2:20:14 PM Run TaskRunner with template: Set system note Semaphore repo v7
2:20:14 PM [WARNING]: Could not match supplied host pattern, ignoring: mikrotik_netcamp
2:20:14 PM
2:20:14 PM PLAY [mikrotik_netcamp] *****
2:20:14 PM skipping: no hosts matched
2:20:14 PM [WARNING]: Could not match supplied host pattern, ignoring: mikrotik_v6
2:20:14 PM
2:20:14 PM PLAY [mikrotik_v6] *****
2:20:14 PM skipping: no hosts matched
2:20:14 PM
2:20:14 PM PLAY [mikrotik_v7] *****
2:20:14 PM
2:20:14 PM TASK [Set system note] *****
2:20:27 PM ok: [10.11.1.101]
2:20:27 PM ok: [10.11.1.102]
2:20:27 PM
2:20:27 PM PLAY RECAP *****
2:20:27 PM 10.11.1.101 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
2:20:27 PM 10.11.1.102 : ok=1 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
2:20:27 PM
```

Инсталиране на ANSIBLE

```
sudo apt update
sudo apt install software-properties-common
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Допълнително инсталиране на библиотеки за работа с MIKROTIK API

```
pip3 install ansible-pylibssh
pip3 install librouteros
```

/etc/ansible/ansible.cfg

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

- fatal: [10.11.1.213]: FAILED! => {"changed": false, "msg": "paramiko: The authenticity of host '10.11.1.213' can't be established.\n\nThe ssh-rsa key fingerprint is b'2f7a2a91c27194a202a09ae627a2eb7c'."}
- fatal: [10.11.1.208]: FAILED! => {"changed": false, "msg": "paramiko: The authenticity of host '10.11.1.208' can't be established.\n\nThe ssh-rsa key fingerprint is b'10abd2d95228a7926d6f9e7f0d4e416e'."}

Въпроси?



Благодаря за вниманието

