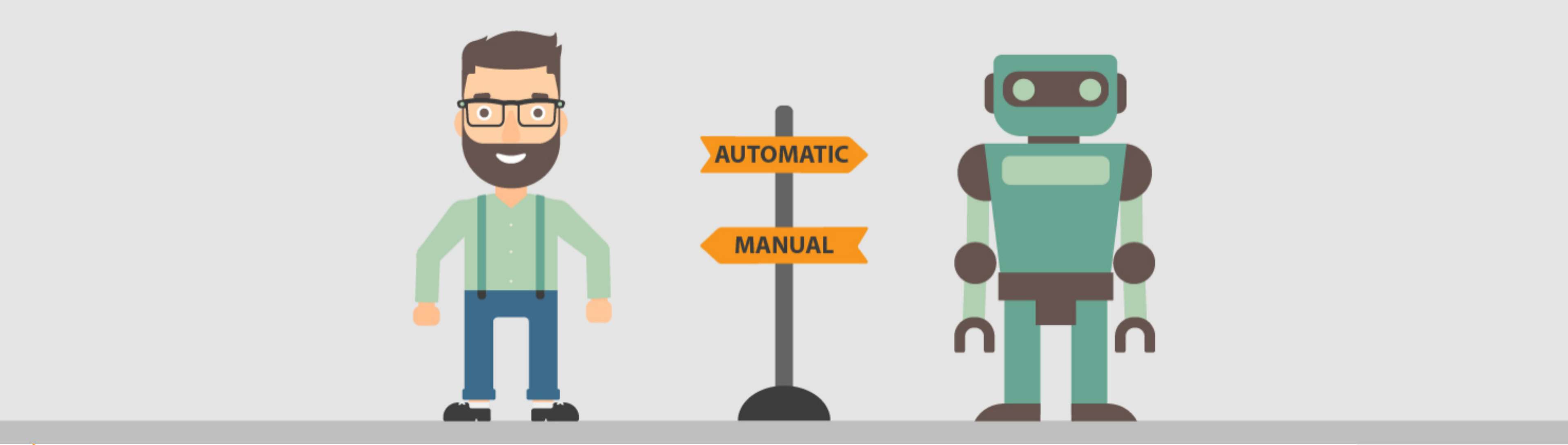


Zero Touch Provisioning using Python & Mikrotik API

Ahmad Rosid Komarudin, TR0574

Network Engineer & SDN Developer, PT Network Data Sistem

What are we going to talk about?

- Introduction to Network Automation
- General problem
- How to solve it
- Mikrotik scripting
- Python for networking
- SSH vs API
- DEMO TIME!!!

Introduction to Network Automation

- Network Automation is a methodology where software automatically configures, provisions, manages and tests network devices

General Problem

General Problem

- We have thousand mikrotik devices
- We need to configure identik parameters in all of mikrotik devices, such us SNMP community, ntp client, firewall rule, basic security, etc

General Problem

- Need many peoples work together in a few days to configure thousand of devices.
- Need to pay extra for many peoples who doing that job

General Problem

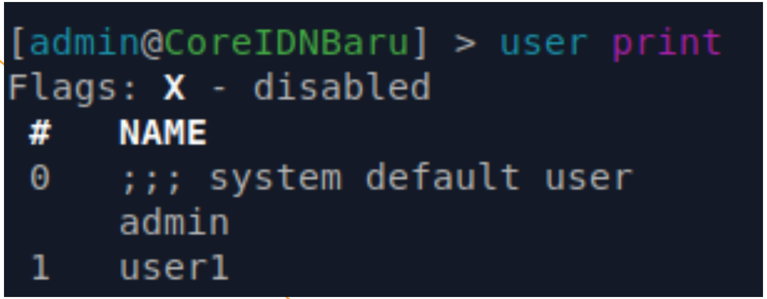
- **Human error** is a big enemy

Why I can’t ssh to my router??

See!! ssh service is blocked by your firewall!

General Problem

- **Miscommunication** is daily habit



```
[admin@CoreIDNBaru] > user print
Flags: X - disabled
 #   NAME
 0   ;;; system default user
     admin
 1   user1
```

Who create new user in my router??!!!

Who else if not him?!!

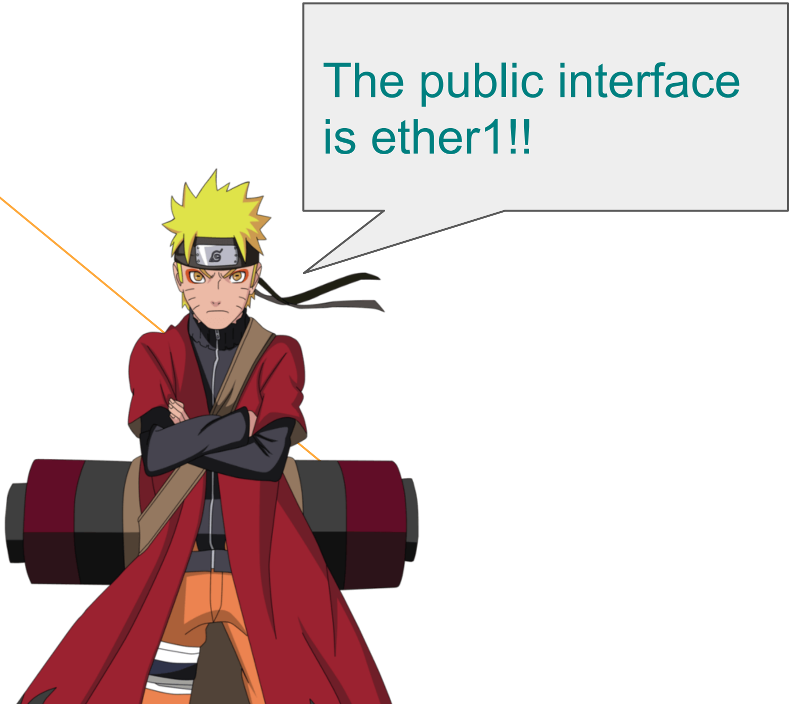
General Problem

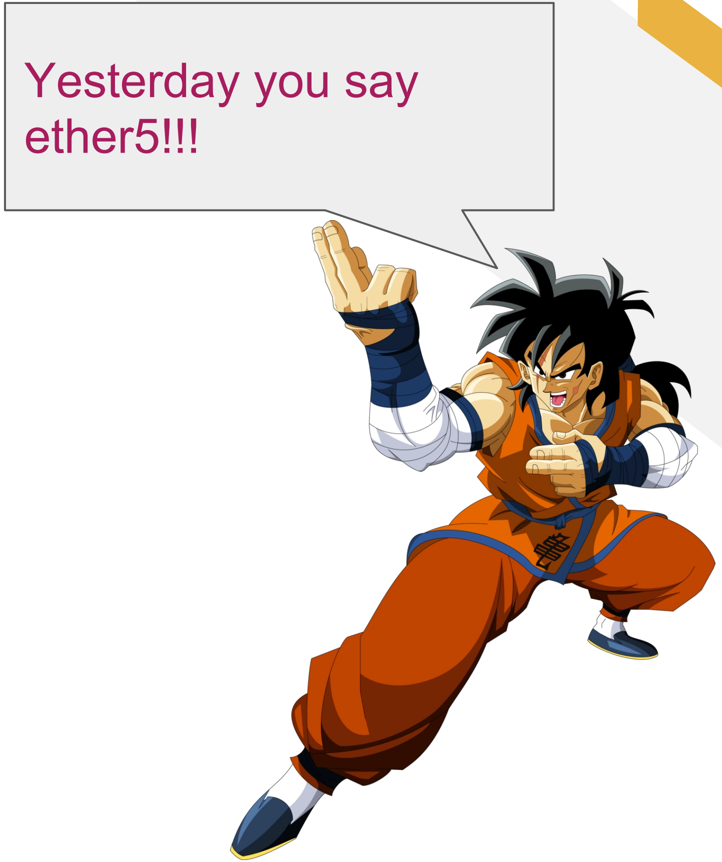
- People will feel bored when doing repititive jobs. When People bored, the jobs will not completed perfectly

When this jobs done!!!

General Problem

- Non standard configuration

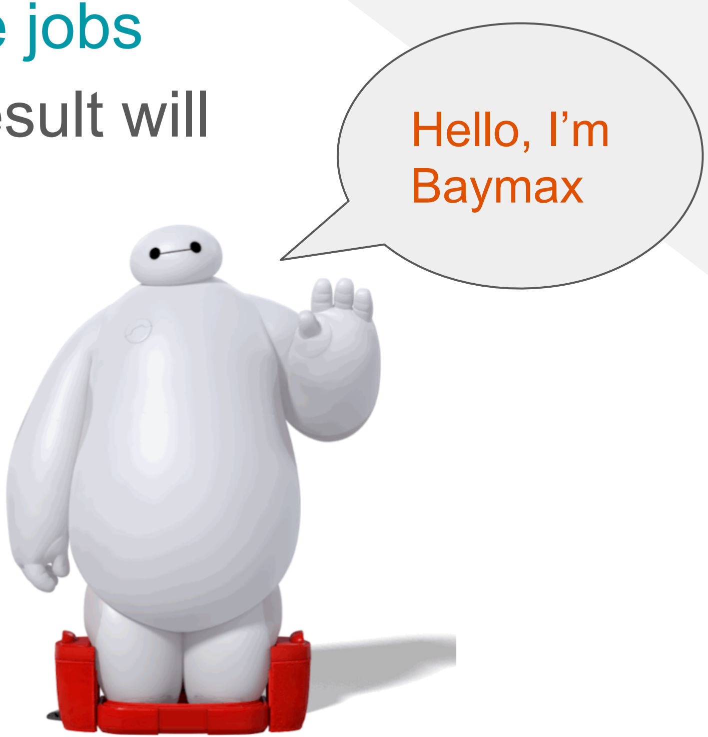




Solution!!!

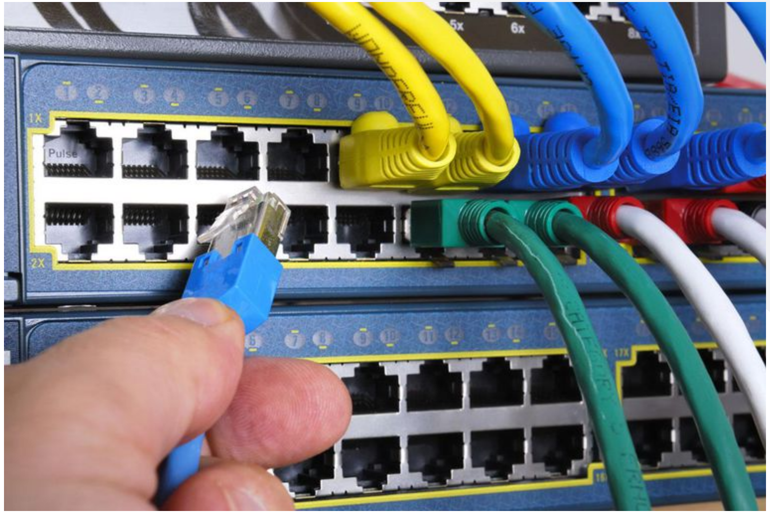
Solution

- Computer can doing repititive jobs without feel bored, and the result will be perfect!



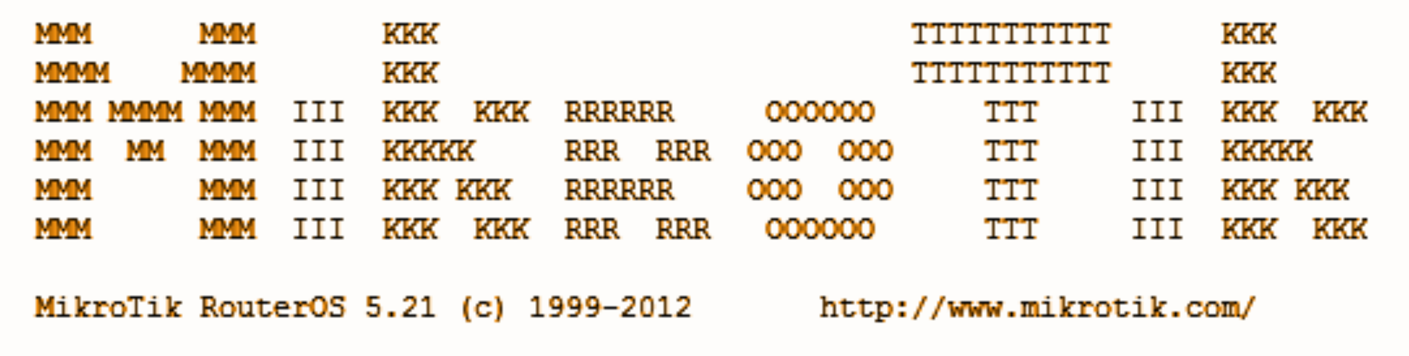
Solution

- We don’t need configure each devices manually, computer will do that for us! We should focus on jobs that can’t solved by computer



Mikrotik Scripting

- Used to auotomate simple stuff in single router



```
MMM      MMM       KKK                          TTTTTTTTTTT      KKK
MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
MMM MMMM MMM  III  KKK  KKK  RRRRRR     OOOOOO      TTT     III  KKK  KKK
MMM  MM  MMM  III  KKKKK     RRR  RRR  OOO  OOO     TTT     III  KKKKK
MMM      MMM  III  KKK KKK   RRRRRR    OOO  OOO     TTT     III  KKK KKK
MMM      MMM  III  KKK  KKK  RRR  RRR   OOOOOO      TTT     III  KKK  KKK

MikroTik RouterOS 5.21 (c) 1999-2012       http://www.mikrotik.com/
```

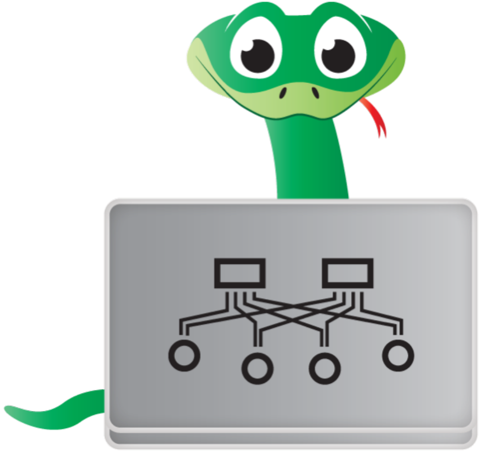
Mikrotik Scripting (example)

- Configure simple queue for target-address 192.168.1.100-192.168.1.200

```
:local x
:for x from 100 to 200 do={/queue simple
add target-address="192.168.1.$x"}
```

Python for Networking

- Used to auotomate advanced stuff in multiple router
- Easy to Learn



Python for Networking (example)

- Configure multiple queue in multiple router

```
ip_address = ["192.168.99.1","192.168.99.2","192.168.99.3"]

for ip in ip_address:
       ssh.connect(hostname=ip, username=user, password=passw)
       for x in range(100,200):
              ssh.exec_command("queue simple add
target="192.168.1.%s" % x)
```

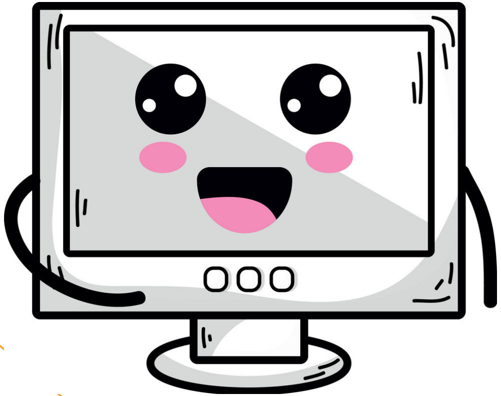
SSH vs API

SSH vs API

```
[admin@Mikrotik] > ip firewall nat print
Flags: X - disabled, I - invalid,

 0     ;;; masquerade hotspot network
       chain=srcnat action=masquerade
        src-address=10.10.10.0/24

 1     ;;; masquerade hotspot network
       chain=srcnat action=masquerade
       src-address=10.10.10.0/24
```

SSH is a human language, we happy to look the display like that. But computer don’t! Computer like display with “key” & “value” pair

SSH vs API



```
{
    "chain": "srcnat",
    "packets": 0,
    "bytes": 0,
    ".id": "*12",
    "invalid": false,
    "dynamic": false,
    "action": "masquerade",
    "src-address": "10.10.10.0/24"
}
```

API is a computer languange, Computer like display with “key” & “value” pair.

Automation using SSH

```
import paramiko
from getpass import getpass

ip_address = ["192.168.99.1","192.168.99.2","192.168.99.3"]

user = raw_input("Input username: ")
passw = getpass()
…

for ip in ip_address:
        ssh.connect(hostname=ip,username=user, password=passw)
        stdin,stdout,stderr = ssh.exec_command("ip address print")
        print stdout.read()
```

Automation using SSH

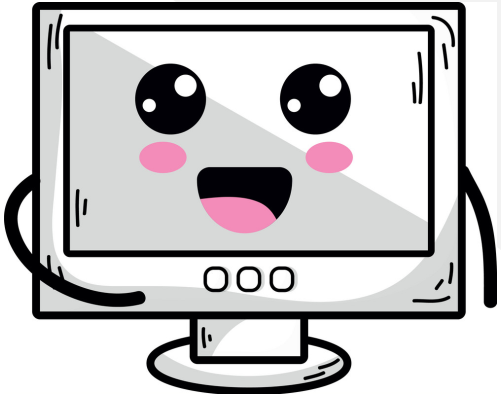
```
$ python mikrotik_ssh.py
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.99.1/24    192.168.99.0    ether4

Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.99.2/24    192.168.99.0    ether1

Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         INTERFACE
 0   192.168.99.3/24    192.168.99.0    ether3
```

Automation using API

```
from librouteros import connect
from getpass import getpass
import json

ip_address = ["192.168.99.1","192.168.99.2","192.168.99.3"]

user = raw_input("Input username: ")
passw = getpass()

for ip in ip_address:
       api = connect(username=user, password=passw, host=ip)
       ip_info = api(cmd="/ip/address/print")
       print json.dumps(ip_info, indent=3)
```

Automation using API

```
$ python mikrotik_api.py
[
    {
        "network": "192.168.99.0",
        "dynamic": false,
        "invalid": false,
        "disabled": false,
        "actual-interface": "ether4",
        ".id": "*1",
        "address": "192.168.99.1/24",
        "interface": "ether4"
    }
]
```



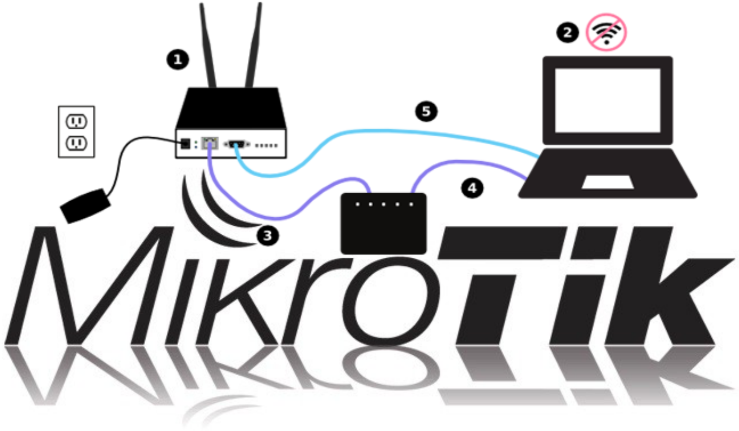
Use Case

Use case

- Security Vulnerability
  - Change Password
  - Change Winbox Port
  - Disable Unused Services
  - Setting Allowed from on Services

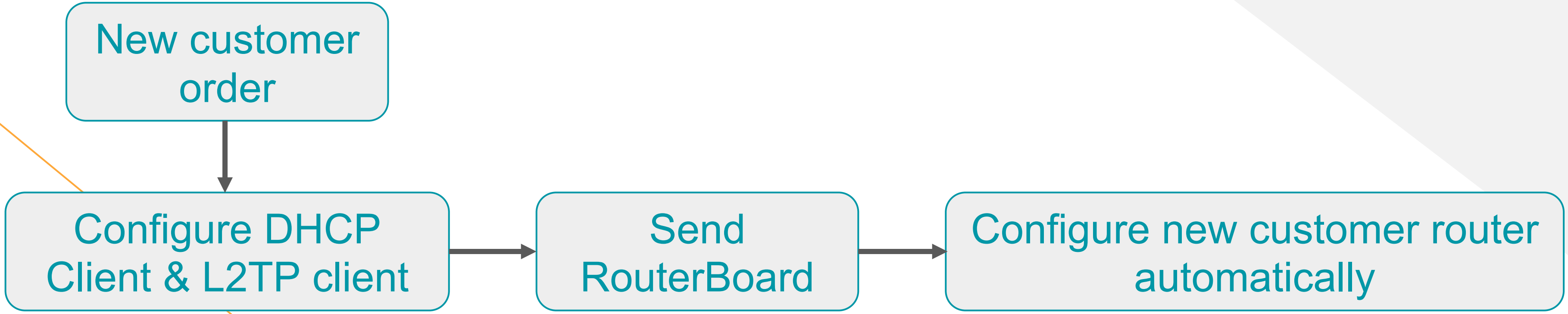
Use case

- Setup new customer in ISP
  - Same private IP
  - Same firewall rule
  - Same NAT rule
  - Same security rule



Demo Time

Flow Chart



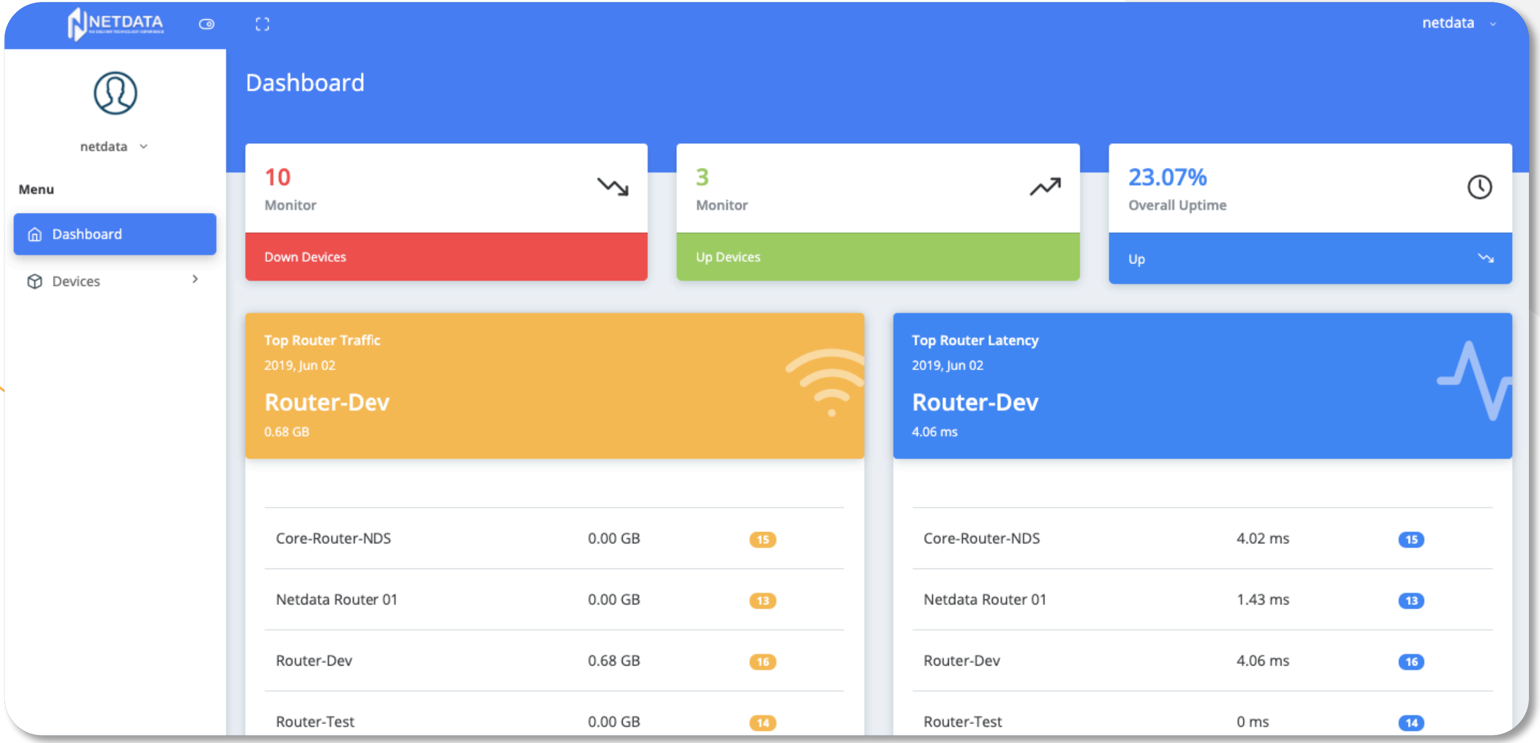
New customer order

Configure DHCP Client & L2TP client

Send RouterBoard

Configure new customer router automatically

Netdata SD-WAN Demo



NETDATA

netdata

netdata

Menu

Dashboard

Devices

Dashboard

10
Monitor
Down Devices

3
Monitor
Up Devices

23.07%
Overall Uptime
Up

Top Router Traffic
2019, Jun 02
Router-Dev
0.68 GB

Core-Router-NDS	0.00 GB	15
Netdata Router 01	0.00 GB	13
Router-Dev	0.68 GB	16
Router-Test	0.00 GB	14

Top Router Latency
2019, Jun 02
Router-Dev
4.06 ms

Core-Router-NDS	4.02 ms	15
Netdata Router 01	1.43 ms	13
Router-Dev	4.06 ms	16
Router-Test	0 ms	14

Question?

Further Reading

- Mikrotik Scripting
  https://wiki.mikrotik.com/wiki/Manual:Scripting
- Mikrotik API
  https://wiki.mikrotik.com/wiki/Manual:API
- Mikrotik Python
  https://wiki.mikrotik.com/wiki/Manual:API_Python3
- My Github
  https://github.com/arrosid

Got more question?
Stay in touch!

Ahmad Rosid Komarudin
info@nds.id

fb.com/ahmadrosidkomarudin

github.com/arrosid

@ahmadrosidkomarudin

linkedin.com/in/arrosid