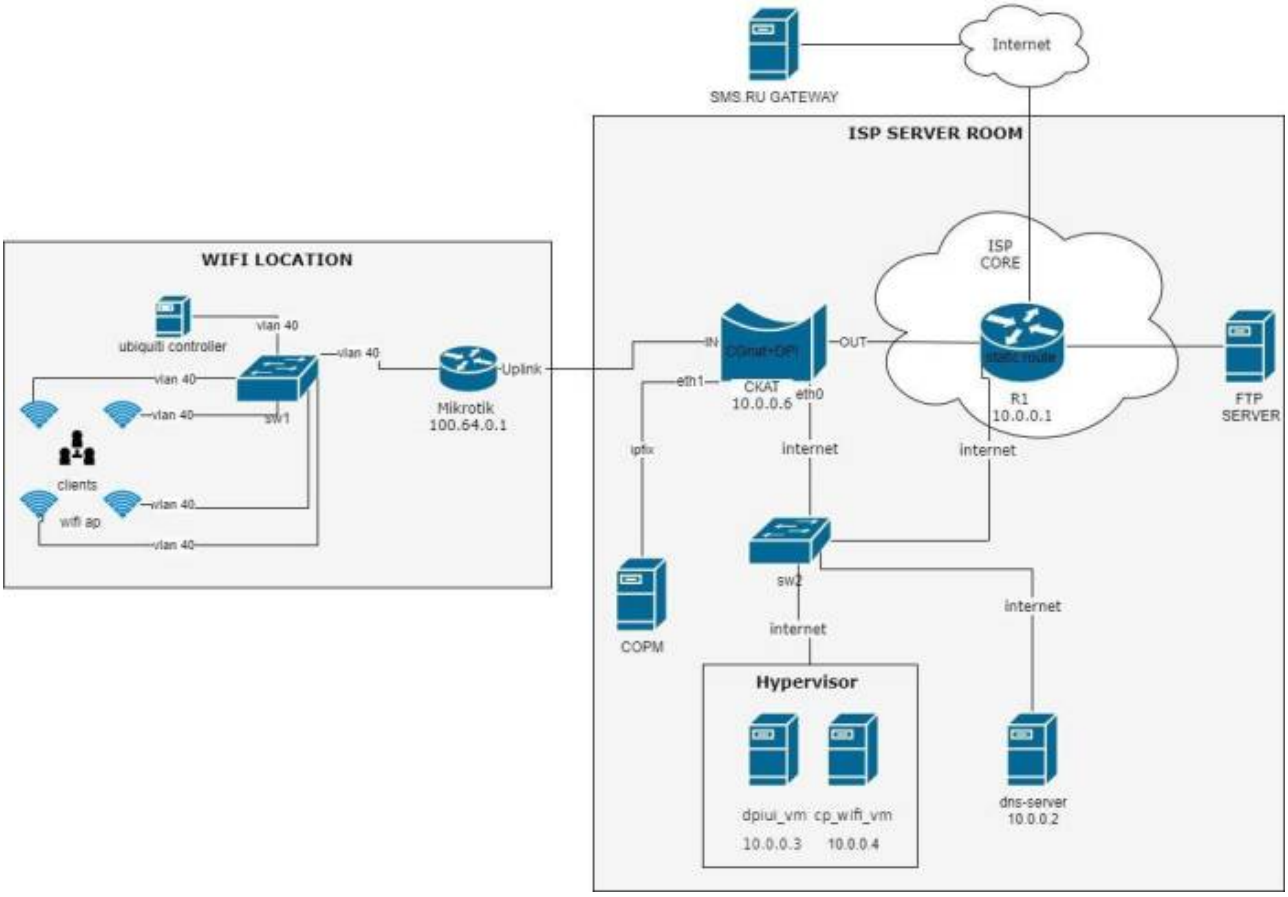
Содержание

Configuring GUI, SSG, and WiFi HotSpot with Session Management Enabled

Network Topology

1. Connect the equipment according to the network topology.



Authorization Sequence

1. The subscriber connects to the WiFi network
2. A welcome page appears informing the subscriber to open a browser and identify themselves
3. The subscriber opens a browser, and upon navigating to any URL, they are redirected to the identification page
4. The subscriber enters their phone number and requests an access code
5. The access code is sent to the phone number via SMS
6. The subscriber enters the received access code
7. Session cookies are recorded on the subscriber's device for a specified period, and the subscriber is redirected to the requested URL

Setting Up Virtual Machines (VM)

1. Create two virtual machines with the following minimum specifications:
   - VM `dpiui_vm` – Equipment prerequisites
   - VM `cp_wifi_vm` – Equipment prerequisites
2. Install the OS on both virtual machines. Choose minimal installation during setup.
   After OS installation, open the console and install packages on both VMs: first

   ```
   yum install epel-release
   ```

   and then:

   ```
   yum install nano tcpdump openssh-server openssh-clients
   ```

3. Disable SELinux on both VMs:
   - Edit the file `/etc/sysconfig/selinux`
   - Set the parameter `SELINUX=disabled` and reboot the VM

Installing and Configuring dpiui_vm

1. Install DPIUI on `dpiui_vm` following the instructions
2. Configure the network on both VMs and SSG:

   ```
   BOOTPROTO=static
   ONBOOT=yes
   IPADDR=10.0.0.x
   NETMASK=255.255.255.0
   GATEWAY=10.0.0.1
   DNS1=10.0.0.2
   ```

   IPADDR — specify for each host according to the scheme (or use your own addressing).

3. Log in to the GUI and add both VMs and SSG in the "EQUIPMENT" section, following the instructions:

Installing and Configuring cp_wifi_vm

1. Install the `wifi_hotspot` package on the `cp_wifi_vm` VM following the instructions:
2. Edit the configuration file for Hotspot:

   ```
   nano /var/www/html/wifi_hotspot/backend/.env
   ```

   Change/add only these lines:

   1. **AAA_HOTSPOT_IP – 10.0.0.4**
      NAS server address, IPv4/IPv6, if unknown — 0.0.0.0
   2. **AAA_HOTSPOT_PORT – 0**
      NAS server port, number, if unknown — 0

3. **AAA_HOTSPOT_ID – 2**
   Network access point ID, integer between 0 and 1000, must be filled in for public WiFi access points, corresponds to the access point ID in field 1 from the access point export
4. **AAA_EXPORT_ENABLED=1**
   Enable AAA export
5. **AUTH_CODE_LENGTH=4**
   Change the number of characters in the SMS authorization code

If the parameter AUTH_CODE_LENGTH is set, then in the file `/var/www/html/wifi_hotspot/frontend/env.js` set the value:

```
AppEnv.AuthCodePlaceHolder = "0000";
```

Finally, run the command:

```
php /var/www/html/wifi_hotspot/backend/artisan queue:restart
```

Installing and Configuring dhcp-isc on cp_wifi_vm

1. Install the `dhcp-isc` package:

   ```
   yum install dhcp expect
   ```

2. Configure the static ARP scripts and the `dhcpd.conf` configuration file:
   - First, the `dhcpd` configuration file:

     ```
     nano /etc/dhcp/dhcpd.conf
     ```

     Set your values for `option domain-name` and `option ntp-servers`!

     ```
     ddns-update-style none;
     authoritative;
     db-time-format local;
     log-facility local7;

     subnet 100.64.0.0 netmask 255.255.252.0 {
       range 100.64.0.3 100.64.3.254;
       default-lease-time 600;
       max-lease-time 600;
       option subnet-mask 255.255.252.0;
       option broadcast-address 100.64.3.255;
       option routers 100.64.0.1;
       option ntp-servers <ntp-server>;
       option domain-name-servers 10.0.0.2;
       option domain-name "name.local";

       on commit {
           set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
           set ClientMac = concat (
           suffix (concat ("0", binary-to-ascii (16, 8, "",
     ```

```
substring(hardware,1,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,2,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,3,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,4,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,5,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,6,1))),2));
      log(concat("Request: IP: ", ClientIP, " Mac: ", ClientMac));

 execute("/usr/local/etc/dhcpd/clients_add_drop.sh", "add",
ClientIP, ClientMac);}
  on release {
      set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
      set ClientMac = concat (
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,1,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,2,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,3,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,4,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,5,1))),2), ":",
      suffix (concat ("0", binary-to-ascii (16, 8, "",
substring(hardware,6,1))),2));

      log(concat("Release: IP: ", ClientIP, " Mac: ", ClientMac));
      execute("/usr/local/etc/dhcpd/clients_add_drop.sh",
"drop_rls", ClientIP, ClientMac);}
  on expiry {
      set ClientIP = binary-to-ascii(10, 8, ".", leased-address);
      log(concat("Timeout: IP: ", ClientIP));
      execute("/usr/local/etc/dhcpd/clients_add_drop.sh",
"drop_exp", ClientIP);}
}
subnet 10.0.0.0 netmask 255.255.255.0 {
}
```

Create directories and change their permissions:

```
mkdir /usr/local/etc/dhcpd/ && chown dhcpd:dhcpd
/usr/local/etc/dhcpd/

touch /usr/local/etc/dhcpd/clients_add_drop_mysql.sh && touch
/usr/local/etc/dhcpd/clients_add_drop.sh
&& chown dpiacc:dpiacc /usr/local/etc/dhcpd/*
```

```
chmod 755 /usr/local/etc/dhcpd/

chmod 755 /usr/local/etc/dhcpd/*
```

Next, copy the following script to `/usr/local/etc/dhcpd/clients_add_drop.sh`:

```
#!/usr/bin/expect -f

set METHOD [lindex $argv 0]
set IP_ADDR [lindex $argv 1]
set MAC_ADDR [lindex $argv 2]
set MAC_ADDR [string toupper $MAC_ADDR]
#Client interface on Mikrotik:
set INT_ClIENT "vWifi"
set status 0


#Recording dhcp-lease (start and end) in the Hotspot database
spawn /usr/local/etc/dhcpd/./clients_add_drop_mysql.sh "$METHOD"
"$IP_ADDR" "$MAC_ADDR"


expect "end_mysql";


#Connecting to the router
spawn ssh -i /usr/local/etc/dhcpd/.ssh/id_rsa admin+t@100.64.0.1 -
oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev/null
expect {
    "password:" {send "\n";}
    "timeout" {set status 1;}
    ">" {}
}
if { $METHOD == "add" && $status == 0} {
send "ip arp add address=$IP_ADDR mac-address=$MAC_ADDR
interface=$INT_ClIENT\r";
expect ">";


send "ip firewall address-list remove \[find address=$IP_ADDR
list=DROP_CLIENTS\]\r";
expect ">";
send "log info \"ADD: $IP_ADDR -- $MAC_ADDR\"\r";
expect ">"
send "quit\r";
expect eof
} elseif { $METHOD == "drop_rls" && $status == 0} {
send "ip arp remove \[find mac-address=$MAC_ADDR\]\r";
expect ">";
send "ip firewall address-list add address=$IP_ADDR
```

```
list=DROP_CLIENTS\r";
expect ">";
send "log info \"DROP_RLS: $IP_ADDR -- $MAC_ADDR\"\r";
expect ">"
send "quit\r";
expect eof
} elseif { $METHOD == "drop_exp" && $status == 0} {
send "ip arp remove \[find address=$IP_ADDR\]\r";
expect ">";
send "ip firewall address-list add address=$IP_ADDR
list=DROP_CLIENTS\r";
expect ">";
send "log info \"DROP_EXP: $IP_ADDR\"\r";
expect ">"
send "quit\r";
expect eof
} elseif {$status == 0} {
send "quit\r";

expect eof
exit 1;
}

set status 0

#Connecting to SSG and adding static subscriber record
spawn ssh -i /usr/local/etc/dhcpd/.ssh/id_rsa dpisu@10.0.0.6 -
oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev/null

expect {
    "password" {send "\r"}
    "timeout" {set status 1; exit 4}
    "\$" {}
}
if {$status == 0} {
send "/var/dpiui2/add_captive_portal_auth_ivstar.sh $IP_ADDR\r"
expect "\$"
send "exit\r";
expect eof
}
```

And copy the following script to `/usr/local/etc/dhcpd/clients_add_drop_mysql.sh` for adding `dhcp-lease` data to the Hotspot database:

```
#!/bin/bash
METHOD=$1
IP_ADDR=$2
MAC_ADDR=$3

MYSQL_CONNECT_LEASEDB="mysql -u root -pvasexperts -Dwifi_hotspot -
```

```
h 127.0.0.1"

if [ "$METHOD" = "add" ]; then
    echo "insert into hotspot_aaa(TYPE,MAC,IP)
values("1",\""$MAC_ADDR"\",\""$IP_ADDR"\");" |
$MYSQL_CONNECT_LEASEDB
elif
   [ "$METHOD" = "drop_rls" ]; then
    echo "insert into hotspot_aaa(TYPE,MAC,IP)
values("2",\""$MAC_ADDR"\",\""$IP_ADDR"\");" |
$MYSQL_CONNECT_LEASEDB

elif
   [ "$METHOD" = "drop_exp" ]; then
    echo "insert into hotspot_aaa(TYPE,MAC,IP)
values("2",\"""\",\""$IP_ADDR"\");" | $MYSQL_CONNECT_LEASEDB
fi

echo "end mysql"
```

Enable the dhcpd server and add a firewall rule:

```
systemctl enable dhcpd
systemctl start dhcpd
firewall-cmd --permanent –add-service=dhcp
firewall-cmd --reload
```

3. Create a script for transferring the session file to FTP:

```
mkdir  /srv/aaa/
mkdir  /srv/aaa/processed/
mkdir  /srv/aaa/script/
touch /srv/aaa/script/script.sh
```

Copy the content into `/srv/aaa/script/script.sh`:

```
#!/bin/bash

FTP_ADDR="<ip ftp>"
FTP_USER="<user ftp>"
FTP_PASS="<password ftp>"

#Directory with AAA Hotspot
DIR="/var/www/html/wifi_hotspot/backend/storage/aaa_events"

ls $DIR | while read f; do
    curl --user $FTP_USER:$FTP_PASS --upload-file $DIR

/$f ftp://$FTP_ADDR/ISP/aaa/  > /dev/null 2>&1
    mv $DIR/$f /srv/aaa/processed
```

and add to cron:

```
crontab -e
*/5 * * * * /srv/aaa/script/script.sh
```

4. Create an SSH key pair:

```
mkdir usr/local/etc/dhcpd/.ssh && cd usr/local/etc/dhcpd/.ssh
ssh-keygen -t rsa
```

Leave the passphrase empty.
Attention! Transfer id.pub to SSG (10.0.0.6) and Mikrotik (100.64.0.1)!

- SSG (10.0.0.6): transfer the file via SSH to SSG and add it to `authorized_keys`

```
cat id.pub >>  ~/.ssh/authorized_keys
```

- Mikrotik (100.64.0.1): transfer the file via SSH or through the Web interface and import it:

```
user ssh-keys import public-key-file=id.pub user=admin
```

Configuring SSG

1. Configure the DB for users on SSG:

```
nano /etc/dpi/fastdpi.conf
udr=1
```

2. Set up filtering based on the federal list:

```
black_list_sm=0
federal_black_list=1
#redirect to page
black_list_redirect=http://block.lan/
```

3. Set the default class:

```
class_order=0
```

4. Enable IPFIX export:
   - Configure the `eth1` interface: `nano /etc/sysconfig/network-scripts/ifcfg-eth1`

```
BOOTPROTO=none
ONBOOT=yes
IPADDR=<ip address>
PREFIX=24
```

```
netflow=8
netflow_dev=eth1
```

```
netflow_timeout=20
netflow_full_collector_type=2
netflow_full_collector=127.0.0.1:1500
netflow_passive_timeout=10
netflow_active_timeout=20
netflow_rate_limit=30
ipfix_dev=eth1

ipfix_tcp_collectors=<ip:port ipfix collectors>
ipfix_meta_tcp_collectors=<ip:port ipfix collectors>
ipfix_observation=127
ipfix_dns_tcp_collectors=<ip:port ipfix collectors>
ipfix_nat_udp_collectors=<ip:port ipfix collectors>
```

5. Minimize traffic in class 7:

```
tbf_class7=rate 1kbit
tbf_inbound_class7=rate 1kbit
```

6. Enable redirect to Captive portal: `cp_server=10.0.0.4 (ip cp)`
7. Disable NAT for private addresses: `nat_exclude_private=1`
8. Other SSG settings:

```
ctrl_port=29000
ctrl_dev=lo
scale_factor=1
num_threads=2
class_order=0
mem_tracking_flow=1500000
mem_tracking_ip=3000000
http_parse_reply=1
rlimit_fsize=32000000000
```

9. Replace the content of the script `/var/dpiui2/add_captive_portal_auth_ivstar.sh` with the following:

```
#!/bin/sh
fdpi_ctrl load --service 5 --profile.name='hotspot_white_list_profile'
--ip $1
fdpi_ctrl load --service 11 --profile.name='NAT_PUBLIC_WIFI' --ip $1
fdpi_ctrl load --policing --profile.name='wifi_hotspot_auth_policing' -
-ip $1
```

10. Add the public key for Hotspot access to SSG in the file `/home/dpisu/.ssh/authorized_keys`:

```
#!/bin/sh
fdpi_ctrl load --service 5 --profile.name='hotspot_white_list_profile'
--ip $1
fdpi_ctrl load --service 11 --profile.name='NAT_PUBLIC_WIFI' --ip $1
fdpi_ctrl load --policing --profile.name='wifi_hotspot_auth_policing' -
-ip $1
```

Save all changes in the file `/etc/dpi/fastdpi.conf` and perform a `reboot`.

11. Configure the `eth0` interface for access to Hotspot and DPIUI:

```
nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.0.0.6
PREFIX=24
DNS1=10.0.0.2
```

Configuring DPI and Hotspot via DPIUI

Configuring prioritization by protocols.

1. Go to the DPI Management tab → PROTOCOL PRIORITIZATION (DSCP) → Editor

- cs0 – what we pass through
- cs1 – what we throttle by tariff
- cs7 – what we globally throttle

```
Bittorrent cs7
default cs1
dns cs0
http cs0
https cs0
```

2. CG-NAT on SSG:
Go to the Services Management tab → Services → CGNAT
Create a profile:
Description: `NAT_WIFI`
Type: `CGNAT`
NAT IP pool: `<public ip>`
Number of TCP sessions: `1000` (per subscriber)
Number of UDP sessions: `1000` (per subscriber)

Hotspot Configuration:

1. Go to the Services Management tab → Hotspot
   Web server: `WiFi-Hotspot` (VM `cp_wifi_vm` previously set up in DPIUI)
   Captive portal URL: `https://10.0.0.4` (cp url)
   Session lifetime: `36000`
   Redirect URL: `https://google.ru` (redirect page after successful authorization)
2. Enable WiFi and SMS authorization
   SMS authorization through sms.ru service:
   Method: `Post`
   URL: `https://sms.ru/sms/send`
3. Body (From):

```
api_id  =  <id from sms.ru personal account>
to  =  [PHONE]
msg  =  Your code for WIFI: [CODE]
```

Hotspot Tariffs (in the editor):

1. Tariff for authorization:

```
htb_inbound_root=rate 5mbit ceil 5mbit burst 2500kbit cburst 2500kbit
htb_inbound_class0=rate 8bit ceil  5mbit burst 8bit cburst 2500kbit
htb_inbound_class1=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_inbound_class2=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_inbound_class3=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_inbound_class4=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_inbound_class5=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_inbound_class6=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_inbound_class7=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_root=rate 100kbit ceil 100kbit burst 50kbit cburst 50kbit
htb_class0=rate 8bit ceil  100kbit burst 8bit cburst 50kbit
htb_class1=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_class2=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_class3=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_class4=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_class5=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_class6=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_class7=rate 8bit ceil  8bit burst 8bit cburst 8bit
```

2. Tariff for free WiFi:

```
htb_inbound_root=rate 10mbit ceil 10mbit burst 5mbit cburst 5mbit
htb_inbound_class0=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class1=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class2=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class3=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class4=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class5=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class6=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_inbound_class7=rate 8bit ceil  8bit burst 8bit cburst 8bit
htb_root=rate 10mbit ceil 10mbit burst 5mbit cburst 5mbit
htb_class0=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class1=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class2=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class3=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class4=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class5=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class6=rate 8bit ceil  10mbit burst 8bit cburst 5mbit
htb_class7=rate 8bit ceil  8bit burst 8bit cburst 8bit
```

3. Services:
Go to Services Management, enable CGNAT, and select the NAT_WIFI profile.

4. Allow list:
Go to the Services Management tab → Services → Block lists and Allow lists.
Select the desired profile and create a list: `ip 10.0.0.4 (cp ip)`
If there is a DNS record for CP, add it as: `cn example.com`
Save settings through the interface.

Mikrotik Configuration (100.64.0.1)

1. Configure Client Interface on Mikrotik:
   Upgrade to Router OS 6.48.x

```
  /interface vlan
  add arp=reply-only arp-timeout=10m interface=sfp1 name=vWifi vlan-
id=40

  /ip settings
  set icmp-rate-limit=5 rp-filter=strict

  /ip address
  add address=100.64.0.1/22 interface=vWifi network=100.64.0.0

  /ip dhcp-relay
  add dhcp-server=10.0.0.4 disabled=no interface=vWifi local-
address=100.64.0.1 name=relay1

  /ip dns
  set servers=10.0.0.2

  /ip route
  add distance=1 dst-address=10.0.0.4/32 gateway=<specify gateway>
pref-src=100.64.0.1

  /system clock
  set time-zone-name=Europe/Moscow

  /system ntp client
  set enabled=yes primary-ntp=<specify ntp server>

  /tool bandwidth-server
  set authenticate=no enabled=no
```

2. Configure IP Connectivity between DHCP/Hotspot and Mikrotik

Unifi Network Configuration

1. Configure Ubiquiti Access Points:
   - Install Unifi Network on the server.
   - Configure DHCP to provide settings to the access points.

- If the access points and controller are in different subnets, specify option 43 in DHCP with the controller IP address in hex format.
  https://help.ui.com/hc/en-us/articles/204909754-UniFi-Device-Adoption-Methods-for-Remote-UniFi-Controllers
  Note: Switch to the old interface by toggling the switch in System Settings → New USER Interface.

1. **Configure Network and Additional Settings:**
   - Go to Settings → Network
     Create a new network with VLAN 40, name it `WiFi-Client`, set the gateway as `100.64.0.1/22`, and configure other options as desired.
   - Go to Settings → Guest Control
     In Pre-Authorization Access, specify the IP of the Hotspot (10.0.0.4).
   - Go to Settings → Wireless Networks
     - Create a WiFi network.
     - Open ADVANCED OPTIONS.
     - Enter any name/SSID.
     - Check Enabled.
     - Check Open.
     - Check Guest Policy.
     - Select `WiFi-Client` in Network.
     - Check Block LAN to WLAN Multicast and Broadcast Data.
     - Check Allow BSS Transition with WNM.
     - Check Block Tunneled Link Direct Setup (TDLS) connections.
     - Check Isolate stations on layer 2 (ethernet) level.
   - Click Save.