

# MikroTik



## DYNAMIC DNS & CLOUD FEATURE

Manila- Philippines MUM

December 2<sup>nd</sup> 2014

MANI RAISSDANA

M.IT.S Co. [WWW.MITS-CO.COM](http://WWW.MITS-CO.COM)

# MANI RAISSDANA



- ▣ MikroTik Certified Trainer (since 2011)
- ▣ M.IT.S Co Founder & CTO  
(MikroTik Sales & Training Partner)
- ▣ Own a WISP (MikroTik Wireless Platform)

# Dynamic DNS & Cloud Feature

## Table of Contents

- ▣ Dynamic DNS Tool
- ▣ DDNS update Script for changeip.com
- ▣ DDNS update Script for DynDNS
- ▣ DDNS update Script for DNSoMatic
- ▣ MikroTik Cloud Service

# Dynamic DNS Update Tool

A way to keep domain name pointing to Dynamic IP

```
[admin@MikroTik] tool> dns-update
```

- ▣ Sends DNS update request to name server
- ▣ DNS update could be secured
- ▣ **hmac-md5** is the only supported Algorithm

# Dynamic DNS Update Tool

Important to know:

- ▣ Works only with BIND Server
- ▣ System Clock should be differ less than 5 Mins

# BIND

## Software Package

Berkeley Internet Name Domain

- ▣ Open Source Software Package (DNS Server)
- ▣ Developed by Internet Software Consortium (ISC)
- ▣ Most widely used DNS software



# Example

BIND Server: 1.1.1.1

Zone: example.com

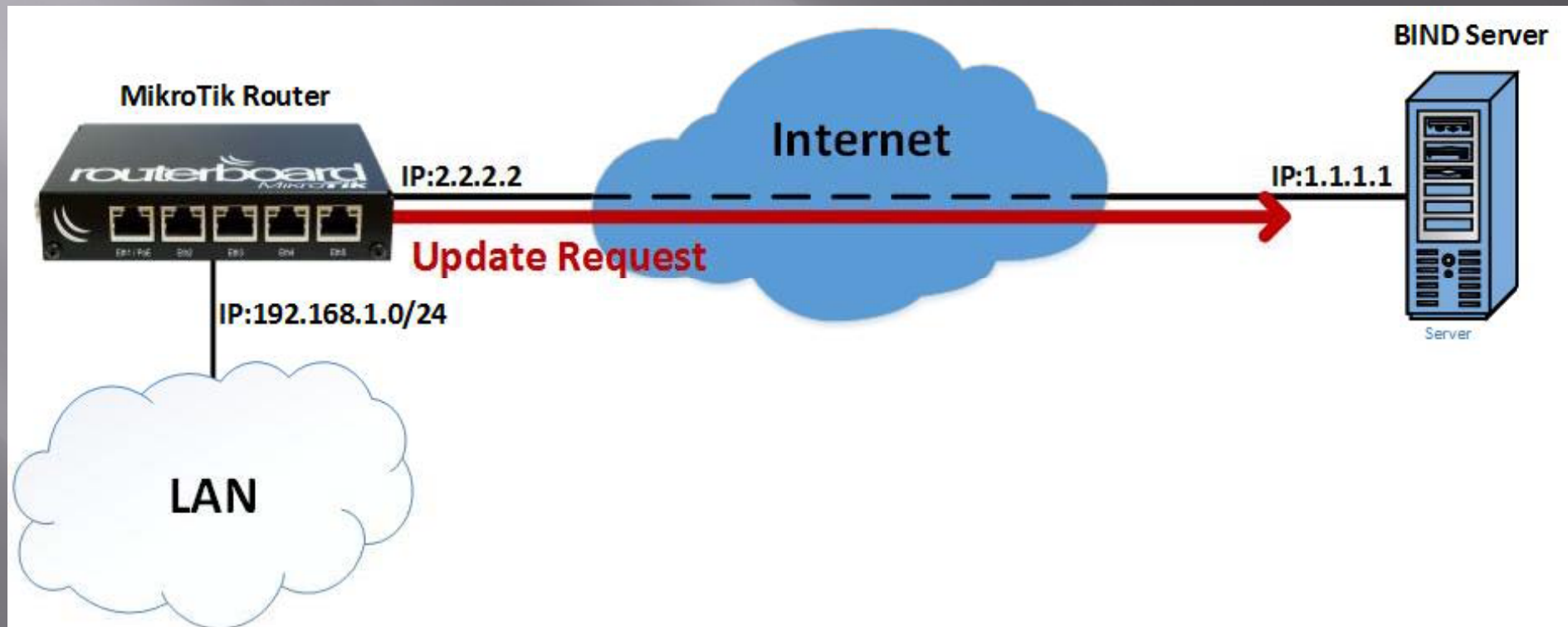
Name: mydomain

Address: 2.2.2.2

key-name: test

Key: update

```
[admin@MikroTik] tool> dns-update dns-server=1.1.1.1 name=mydomain \  
zone=example.com address=2.2.2.2 key-name=test key=update
```



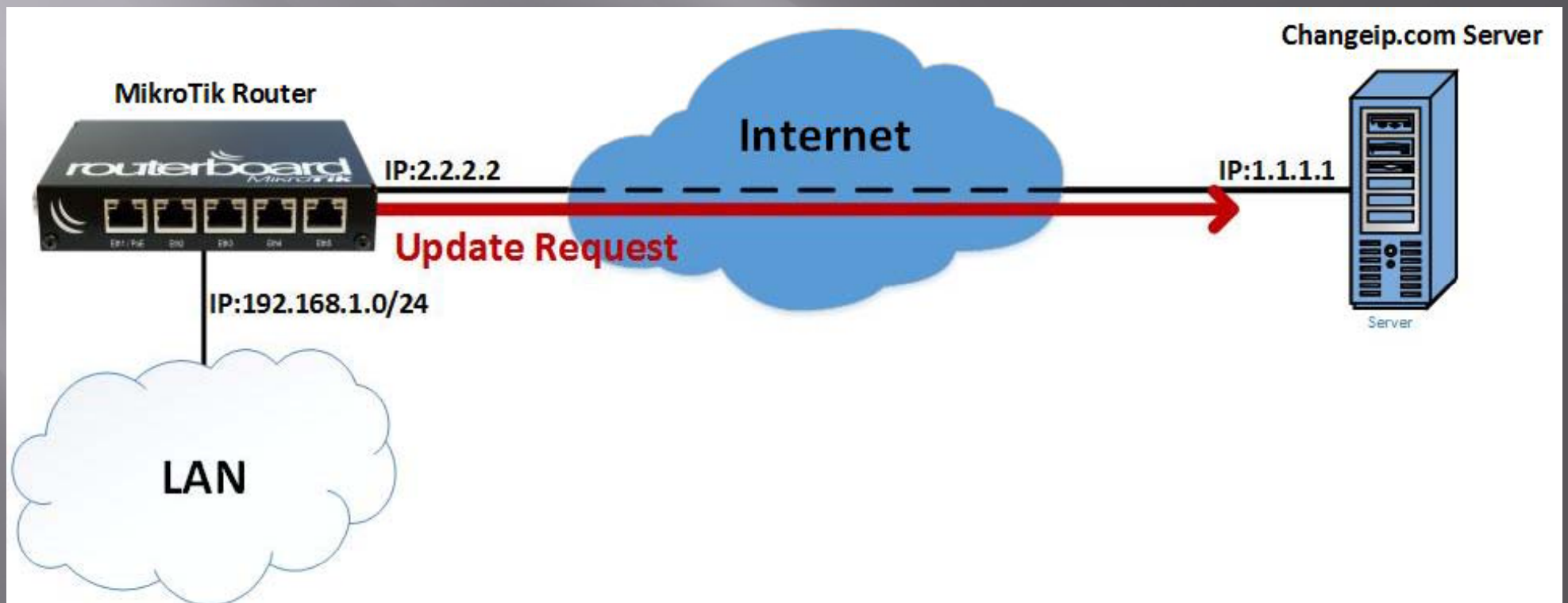
# DDNS update Script for changeip.com

- ▣ Enhanced DDNS Solutions
- ▣ Free solutions as well as Prepaid
- ▣ Set scheduler to run Script once in a while



# DDNS update Script for changeip.com

- ▣ Editing of Global Variables is needed
- ▣ Scheduler also can set manually
- ▣ Behind NAT Script is also available

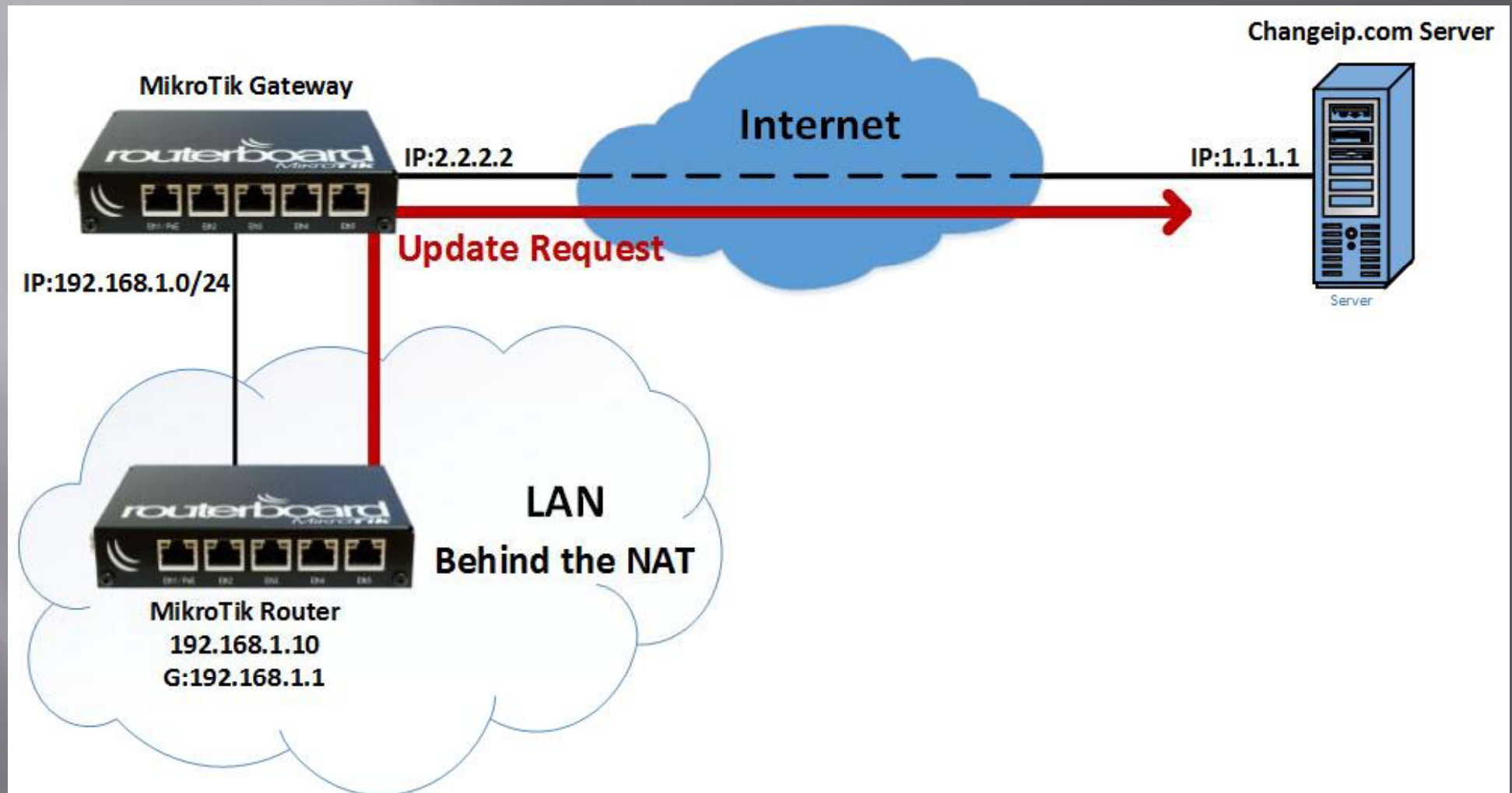


```

add name=HomingBeacon-Simple policy=\
    \n#\r\
    \n# CONFIGURATION FIELD DEFINITIONS:\r\
    \n# ddnsuser:  ChangeIP.com user id.\r\
    \n# ddnspass:  ChangeIP.com password.\r\
    \n# ddns host:  hostname (www.example.com) to update.\r\
    \n# ddnsinterface:  Enter a list of interface names - case sensitive.\r\
    \n:global ddnsuser \"ChangeIPUserID\" \r\
    \n:global ddnspass \"PASSWORD\" \r\
    \n:global ddns host \"MyRouterHostname.example.org\" \r\
    \n:global ddnsinterface \"ether1\" \r\
    \n:global ddns system (\"mt-\" . [/system package get [/system package find\
    \_name=system] version] ) \r\
    \n:global ddns ip [ /ip address get [/ip address find interface=\$ddnsinter\
    face] address ] \r\
    \n:global ddns lastip \r\
    \n:if ([ :len [/interface find name=\$ddnsinterface]] = 0 ) do={ :log info \
    \"DDNS: No interface named \$ddnsinterface, please check configuration.\" \
    \n:if ([ :typeof \$ddns lastip ] = \"nothing\" ) do={ :global ddns lastip 0.\
    0.0.0/0 } \r\
    \n:if ([ :typeof \$ddns ip ] = \"nothing\" ) do={ \r\
    \n:log info (\"DDNS: No ip address present on \" . \$ddnsinterface . \", p\
    lease check.\") \r\
    \n} else={ \r\
    \n :if (\$ddns ip != \$ddns lastip) do={ \r\
    \n :log info \"DDNS: Sending UPDATE!\" \r\
    \n :log info [ :put [/tool dns-update name=\$ddns host address=:pick \$\
    ddns ip 0 [ :find \$ddns ip \"/\" ] ] key-name=\$ddnsuser key=\$ddnspass ] ] \r\
    \n :global ddns lastip \$ddns ip \r\
    \n } else={ \r\
    \n :log info \"DDNS: No changes necessary.\" \r\
    \n } \r\
    \n} \"
/system scheduler
add comment="" disabled=no interval=5m name=DDNS on-event=\
    HomingBeacon-Simple policy=read,write,policy,test,password start-time=\
    startup

```

# Behind the NAT Scenario



```
/system script
add name=DDNS policy=\
  \n#\r\
  \n:local user \"user\" \r\
  \n:local pass \"pass\" \r\
  \n:local host \"host\" \r\
  \n#####\r\
  \n#####\r\
  \n:global lastwanip;\r\
  \n:if ([ :typeof \$lastwanip ] = \"nothing\" ) do={ :global lastwanip 0.0.0.0
};\r\
  \n:local wanip [:resolve \$host];\r\
  \n:if ( \$wanip != \$lastwanip ) do={\r\
    \n\t/tool fetch mode=http address=\"checkip.dyndns.org\" src-path=\"/\" dst-
path=\"/dyndns.checkip.html\" \r\
    \n\t:local result [/file get dyndns.checkip.html contents]\r\
    \n\t:local resultLen [:len \$result]\r\
    \n\t:local startLoc [:find \$result \": \" -1]\r\
    \n\t:set startLoc (\$startLoc + 2)\r\
    \n\t:local endLoc [:find \$result \"</body>\" -1]\r\
    \n\t:local currentIP [:pick \$result \$startLoc \$endLoc]\r\
    \n\t:set lastwanip \$currentIP;\r\
    \n\t:put [/tool dns-update name=\$host address=\$currentIP key-name=\$user
key=\$pass ]\r\
    \n}
/system scheduler
add comment="" disabled=no interval=5m name=DDNS on-event=\
  HomingBeacon-Simple policy=read,write,policy,test,password start-time=\
  startup
```

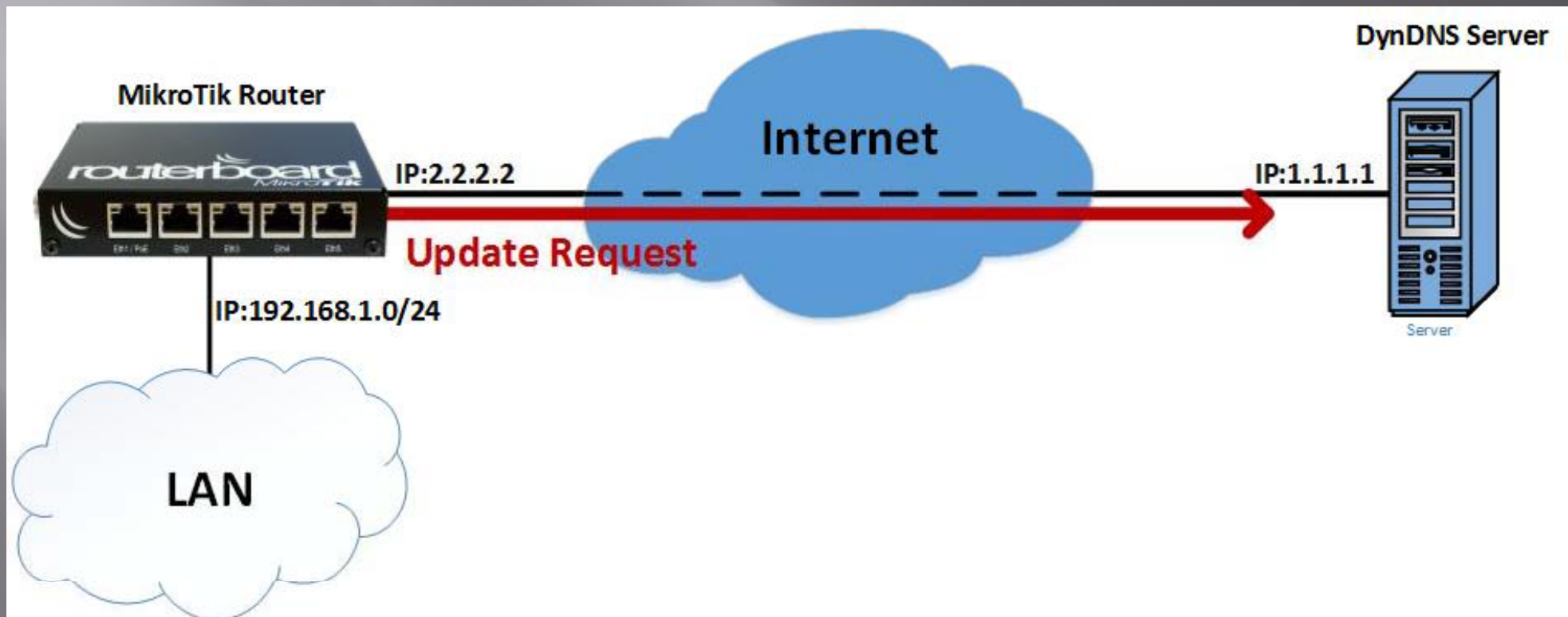


# DDNS update Script for DynDNS

- ▣ One of the advanced DDNS provider (Prepaid)
- ▣ Script is same as changeip with little difference
- ▣ Set scheduler to run Script once in a while

# DDNS update Script for DynDNS

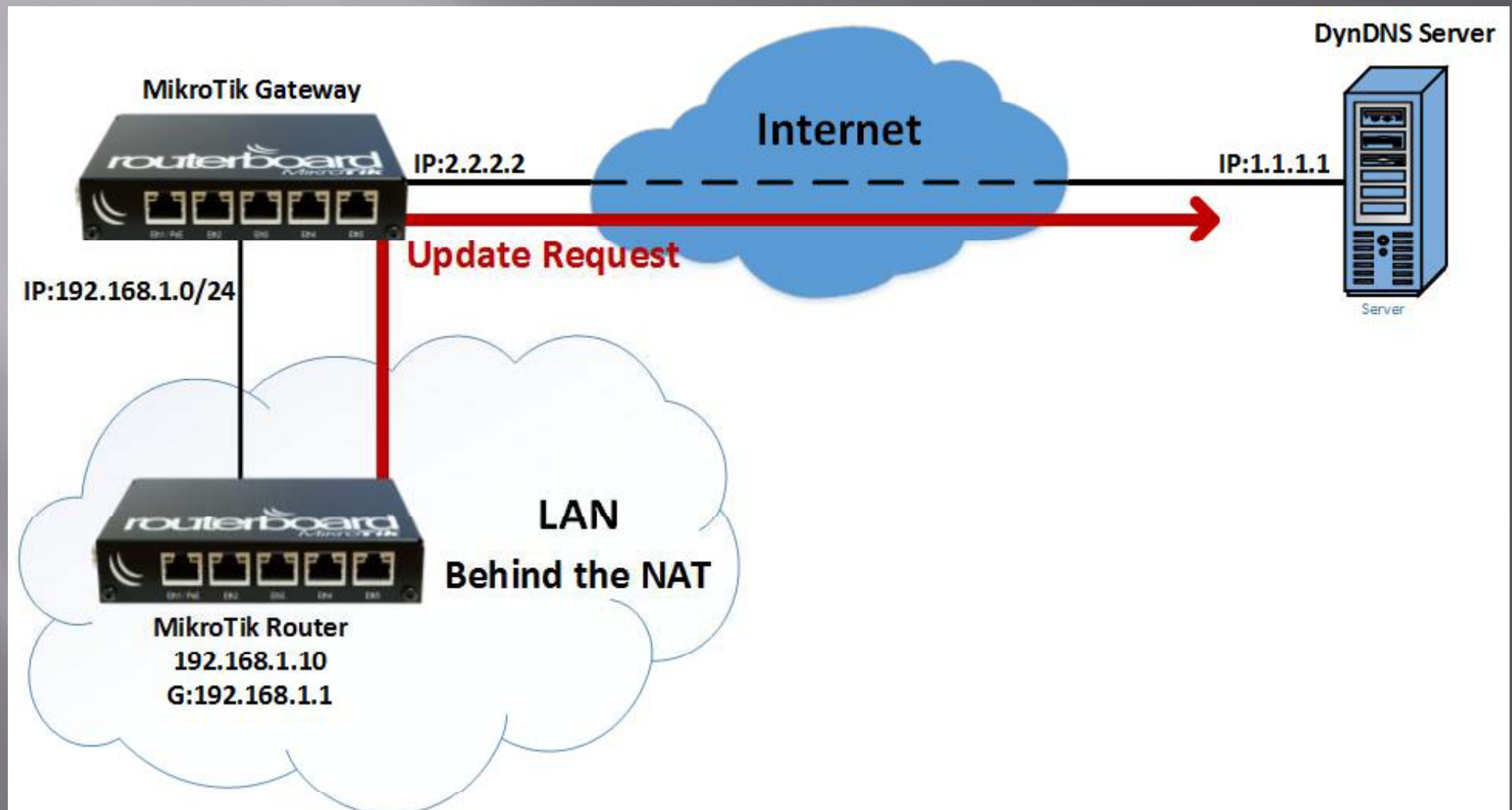
- ▣ Editing of Global Variables is needed
- ▣ Scheduler also can set manually
- ▣ Behind NAT Script is also available





```
/ # Define User Variables
:global ddnsuser "DYNDNSUSER"
:global ddnspass "DYNDNSPASS"
:global ddnshost "DYNDNSHOST"
# Define Global Variables
:global ddnsip
:global ddnslastip
:if ([ :typeof $ddnslastip ] = nil ) do={ :global ddnslastip "0" }
:global ddnsinterface
:global ddnssystem ("mt-" . [/system package get system version] )
# Define Local Variables
:local int
# Loop thru interfaces and look for ones containing
# default gateways without routing-marks
:foreach int in=[/ip route find dst-address=0.0.0.0/0 active=yes ] do={
  :if ([:typeof [/ip route get $int routing-mark ]] != str ) do={
    :global ddnsinterface [/ip route get $int interface]
  }
}
# Grab the current IP address on that interface.
:global ddnsip [ /ip address get [/ip address find interface=$ddnsinterface ] address ]
# Did we get an IP address to compare?
:if ([ :typeof $ddnsip ] = nil ) do={
  :log info ("DynDNS: No ip address present on " . $ddnsinterface . ", please check.")
} else={
  :if ($ddnsip != $ddnslastip) do={
    :log info "DynDNS: Sending UPDATE!"
  }
  :local str
"/nic/update?hostname=$ddnshost&myip=$ddnsip&wildcard=NOCHG&mx=NOCHG&backmx=NOCHG"
/tool fetch address=members.dyndns.org src-path=$str mode=http user=$ddnsuser \
password=$ddnspass dst-path=("/DynDNS.".$ddnshost)
/system scheduler add name=dynDNS interval=00:01 on-event="/system script run dynDns\r\n"
```

# Behind the NAT Scenario



```
# Set needed variables
:local username "YourUsername"
:local password "YourPassword"
:local hostname "nohostset"
:global dyndnsForce
:global previousIP
:log info ("UpdateDynDNS: username = $username")
:log info ("UpdateDynDNS: hostname = $hostname")
:log info ("UpdateDynDNS: previousIP = $previousIP")
# parse the current IP result
:local resultLen [:len $result]
:local startLoc [:find $result ": " -1]
:set startLoc ($startLoc + 2)
:local endLoc [:find $result "</body>" -1]
:local currentIP [:pick $result $startLoc $endLoc]
:log info "UpdateDynDNS: currentIP = $currentIP"
#:set dyndnsForce true
:if (($currentIP != $previousIP) || ($dyndnsForce = true)) do={
  :set dyndnsForce false
  :set previousIP $currentIP
  /tool fetch user=$username password=$password mode=http address="members.dyndns.org" src-
path="/nic/update?hostname=$hostname&myip=$currentIP" dst-path="/dyndns.txt"
  :local result [/file get dyndns.txt contents]
  :log info ("UpdateDynDNS: Dyndns update needed")
  :log info ("UpdateDynDNS: Dyndns Update Result: ".$result)
  :put ("Dyndns Update Result: ".$result)
} else={
  :log info ("UpdateDynDNS: No dyndns update needed")
}
/system scheduler add name=dynDNS interval=00:01 on-event="/system script run dynDns\r\n"
```

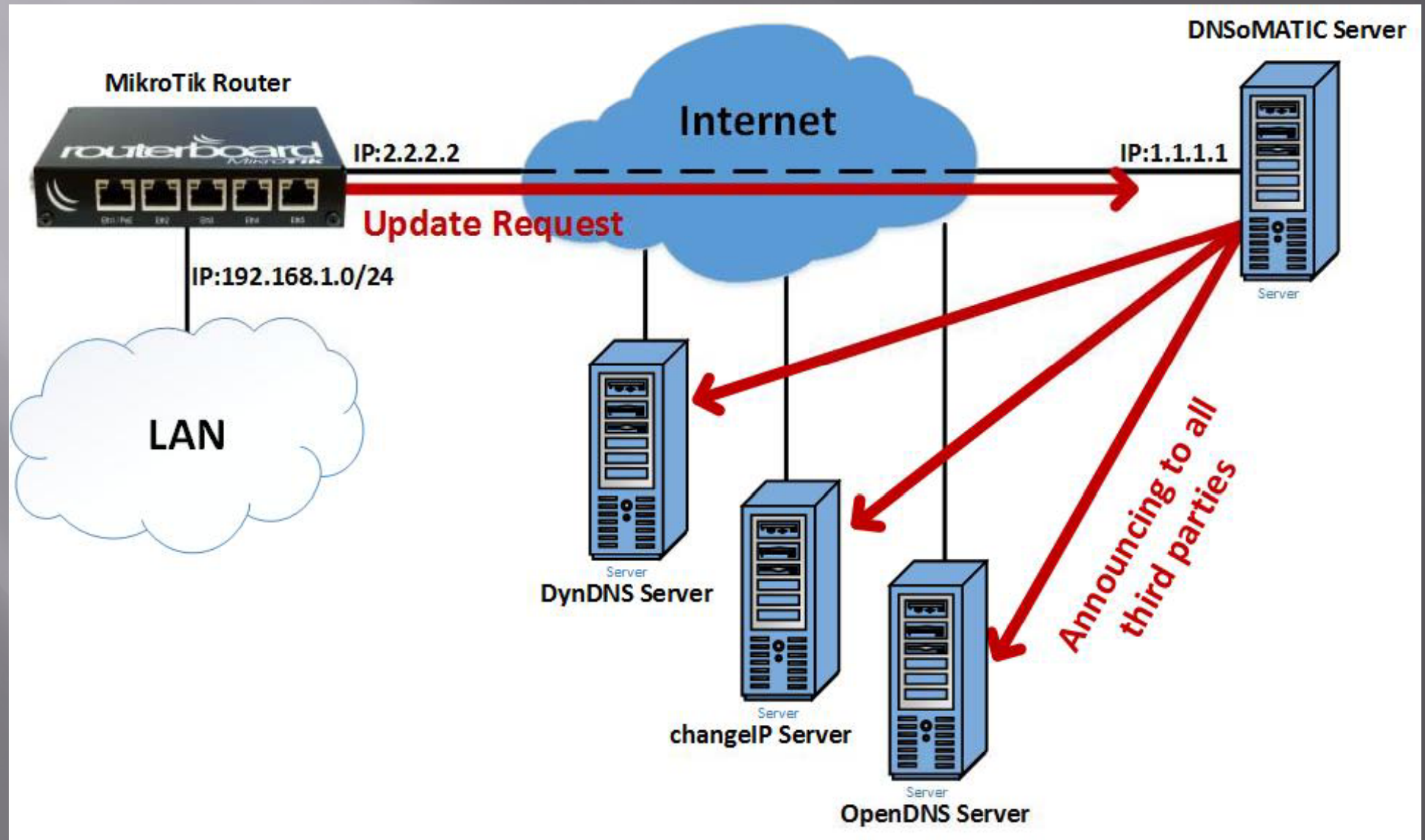
# DDNS update Script for DNSoMATIC

- ▣ Provides Free services
- ▣ Propagating DNS updates to third parties
- ▣ DynDNS, Changeip, OpenDNS and other 27 more
- ▣ Announce to all third parties with single update



# DDNS update Script for DNSoMATIC

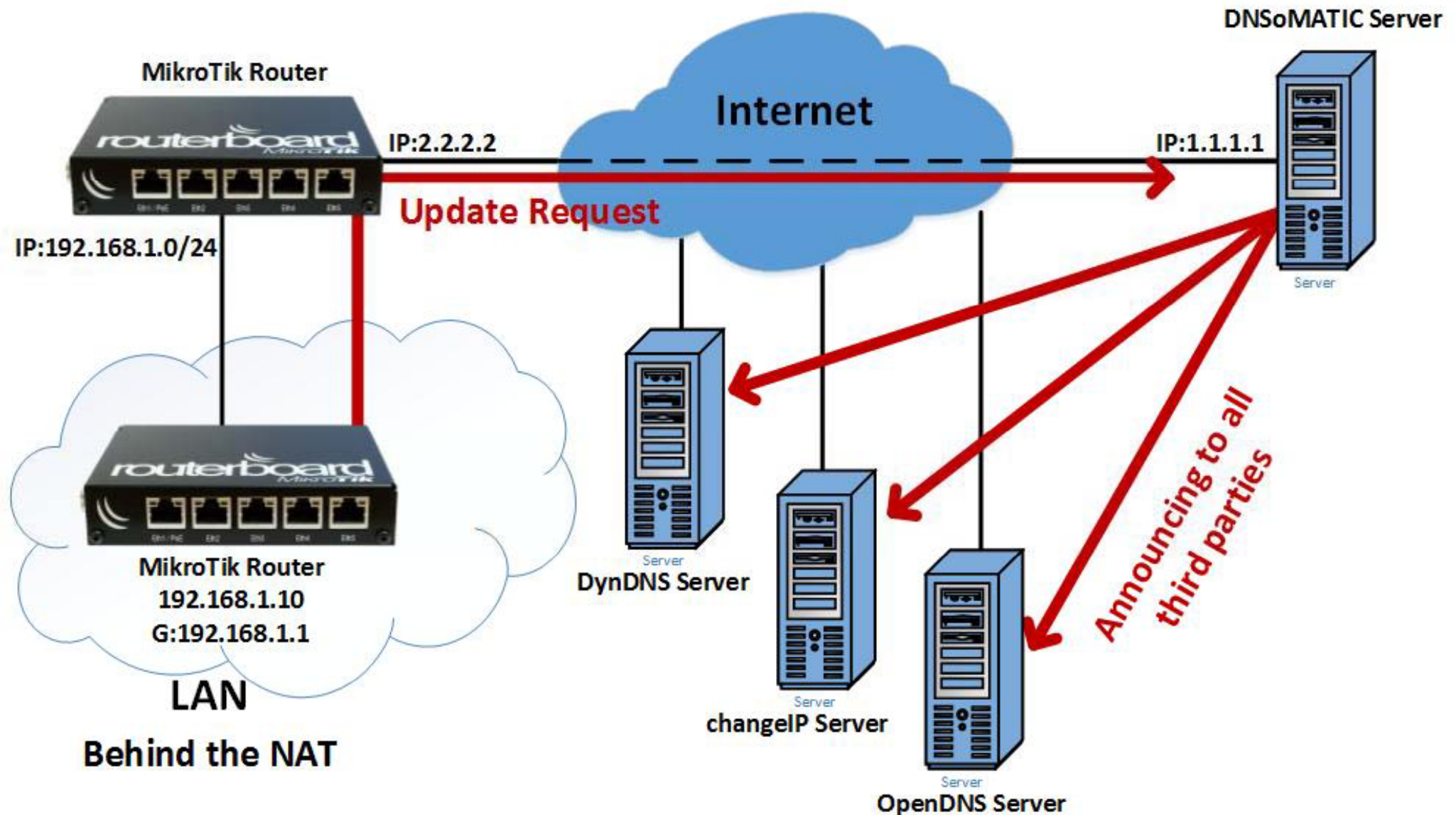
- ▣ Editing of Global Variables is needed
- ▣ Scheduler also can set manually



```
:global previousIP;
:if ([/interface get $inetinterface value-name=running]) do={
# Get the current IP on the interface
:local currentIP [/ip address get [find interface="$inetinterface" disabled=no] address];
# Strip the net mask off the IP address
:for i from=([len $currentIP] - 1) to=0 do={
:if ([pick $currentIP $i] = "/") do={
:set currentIP [pick $currentIP 0 $i]
}
}
:if ($currentIP != $previousIP) do={
:log info "DNSoMatic: Update needed"
:set previousIP $currentIP
# The update URL. Note the "\3F" is hex for question mark (?). Required since ? is a special character in
commands.
:local url
"http://updates.dnsomatic.com/nic/update\3Fmyip=$currentIP&wildcard=NOCHG&mx=NOCHG&backmx=
NOCHG"
:local matichostarray;
:set matichostarray [:toarray $matichost];
:foreach host in=$matichostarray do={
:log info "DNSoMatic: Sending update for $host"
/tool fetch url=($url . "&hostname=$host") user=$maticuser password=$maticpass mode=http dst-
path=("dnsomaticupdate-" . $host . ".txt")\
:log info "DNSoMatic: Host $host updated on DNSoMatic with IP $currentIP"
}
} else={
:log info "DNSoMatic: Previous IP $previousIP and current IP equal, no update need"
```



# Behind the NAT Scenario

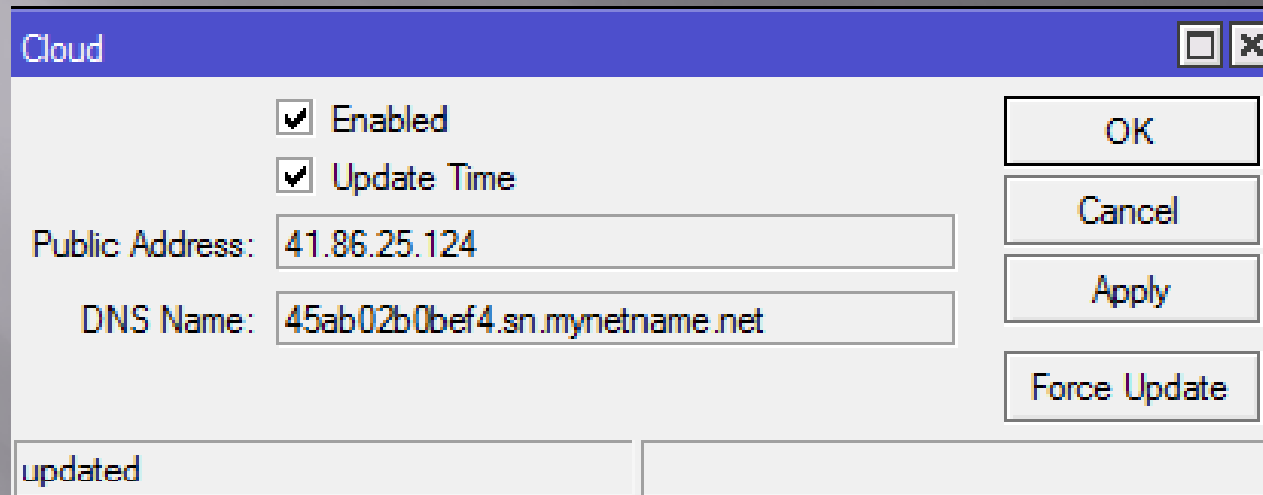


```
:global previousIP;
:log info "Fetching current IP"
# Get the current public IP using DNS-O-Matic service.
/tool fetch url="http://myip.dnsomatic.com/" mode=http dst-path=mypublicip.txt
# Read the current public IP into the currentIP variable.
:local currentIP [/file get mypublicip.txt contents]
:log info "Fetched current IP as $currentIP"
:if ($currentIP != $previousIP) do={
  :log info "DNSoMatic: Update needed" :set previousIP $currentIP # The update URL. Note the "\3F" is hex for
question mark (?). Required since ? is a special character in commands. :local url
"http://updates.dnsomatic.com/nic/update\3Fmyip=$currentIP&wildcard=NOCHG&mx=NOCHG&backmx=
NOCHG"
:local matichostarray;
:set matichostarray [:toarray $matichost];
:foreach host in=$matichostarray do={
  :log info "DNSoMatic: Sending update for $host"
/tool fetch url=($url . "&hostname=$host") user=$maticuser password=$maticpass mode=http dst-
path=("dnsomaticupdate-" . $host . ".txt")
:log info "DNSoMatic: Host $host updated on DNSoMatic with IP $currentIP"
}
} else={
:log info "DNSoMatic: Previous IP $previousIP and current IP equal, no update need"
```

# MikroTik Cloud Service

- ▣ Added from RouterOS V6.14
- ▣ DDNS Name Service for RouterBoards
- ▣ Useful when IP changes often
- ▣ Provides 2 Services:
  1. DDNS
  2. Approximate Time
- ▣ More features added later on

# MikroTik Cloud Service



The screenshot shows a 'Cloud' configuration window with a blue title bar. It contains several settings: 'Enabled' and 'Update Time' are checked. The 'Public Address' field contains '41.86.25.124' and the 'DNS Name' field contains '45ab02b0bef4.sn.mynetname.net'. On the right side, there are four buttons: 'OK', 'Cancel', 'Apply', and 'Force Update'. At the bottom left, there is a status field labeled 'updated'.

- ▣ If you wish to send your local IP address:

```
[admin@E-MAIN] /ip cloud advanced> set use-local-address=  
no yes  
[admin@E-MAIN] /ip cloud advanced> set use-local-address=yes
```

# MikroTik Cloud Service

Important to know:

- ▣ Such access should be accepted by firewall
- ▣ In case of having multiple Public IPs/Gateways, IP used for the update may not be as expected
- ▣ There is no any expire time for DDNS records



# Conclusion

- ▣ These are RouterOS new features
- ▣ MikroTik Developers working day by day to improve
- ▣ Your feedbacks highly looking forwarded



# MY CONTACT DETAILS

Official Phone: +98 (21) 88 400 717  
Private Cell: +98 (912) 149 7009 (Viber)  
International Cell: +37259431151  
Skype: mani\_raissdana  
[m.raissdana@mits-co.com](mailto:m.raissdana@mits-co.com)  
[raissdana.mani@gmail.com](mailto:raissdana.mani@gmail.com)  
[www.mits-co.com](http://www.mits-co.com)



mani\_raissdana



mikrotikiran @mani\_raissdana



Mani Raissdana



*Any Questions????????*

**ENJOY MUM**

**GOOD LUCK**