

# Other

- [Cisco Switch commands](#)
- [PowerTools Renamer](#)
- [Testing Multicasting](#)
- [Updating Ghost Server](#)
- [Replacing a XCP-NG Raid1 disk and growing the array](#)
- [Cloudflare Dynamic DNS IP Updater](#)
- [OpenDNS Linux IP Updater](#)

# Cisco Switch commands

## Assign IP

```
enable
configure terminal
interface vlan <No>
ip address <IP> <Mask>
no shutdown
exit
```

## Remove IP

```
enable
configure terminal
interface vlan <No>
no ip address <IP> <Mask>
exit
```

## Assign a Gateway

```
enable
configure terminal
ip default-gateway <ip>
exit
```

## Show interface IPs

```
show ip interface brief
```

## Backup Config

- Run tftpd64 on windows machine: [tftpd64.464.zip](http://tftpd64.464.zip) (make sure IPs are on same subnet)
- Make sure it can be reached i.e. the Windows firewall is not blocking it (must be able to get through "Public Network").

```
copy startup-config tftp:
```

For restore put tftp: first.

## To reboot the switch

```
enable  
reload now
```

## Remove a trunk

```
enable  
configure terminal  
interface GigabitEthernet1/0/13  
switchport trunk allowed vlan remove 20,30,50,100  
switchport mode access  
no switchport nonegotiate  
no switchport trunk allowed vlan none  
no switchport trunk native vlan 100
```

## Changing VLAN on port

```
enable  
configure terminal  
interface GigabitEthernet1/0/port  
switchport access vlan vlanNo  
switchport mode access  
end
```

## Check config and save to start-up config

```
show running-config interface GigabitEthernet1/0/port  
  
write memory
```

# PowerTools Renamer

Select all text

Replace with "Main-number"

# Testing Multicasting

## Multicast Address Basics

Multicast Group Address Range:

Valid range: 224.0.0.0 to 239.255.255.255 (Reserved for multicast traffic, not assigned to individual devices.)

Common Multicast Addresses:

- 224.0.0.1: All systems on the local network segment.
- 224.0.0.2: All routers on the local network segment.
- 239.255.255.250: Used by Universal Plug and Play (UPnP) and Simple Service Discovery Protocol (SSDP). This is common for smart devices like Google Home. You can pick any address in this range if you're testing, as long as it's not reserved for a specific purpose.

## How It Relates to Your Subnets

Multicast traffic is sent to the multicast group address (e.g., 239.255.255.250) and not directly to the subnet IP range (e.g., 192.168.10.0/24 or 192.168.20.0/24).

Devices that "subscribe" to the multicast group on either subnet will receive traffic sent to the multicast group address if multicast routing is configured properly (like IGMP Proxy in pfSense).

### To verify multicast routing between your subnets:

#### On Subnet 192.168.10.0:

Run a multicast listener for 239.255.255.250 on one device:

```
iperf -s -u -B 239.255.255.250 -i 1
```

#### On Subnet 192.168.20.0:

Send multicast traffic from another device:

```
iperf -c 239.255.255.250 -u -b 1M -t 10
```

If multicast routing is configured correctly, the listener on 192.168.10.0 should receive the traffic.

# Updating Ghost Server

Change to site directory.

call ghost update (not sudo)

# Replacing a XCP-NG Raid1 disk and growing the array

Check RAID status

```
cat /proc/mdstat
```

Identify RAID devices and members

```
mdadm --detail /dev/md0
```

If `/dev/md0` doesn't exist, check which md device is being used:

```
lsblk | grep md
```

Check physical disks and partitions

```
lsblk -o NAME,SIZE,TYPE,UUID,MOUNTPOINT
```

Check volume groups and logical volumes (if LVM is used)

```
vgs  
lvs
```

Check mount points and filesystems

```
df -h
```

For Lenovo m720q host:

```
cat /proc/mdstat # check everything is synced  
  
mdadm /dev/md127 --fail /dev/nvme0n1 # fail drive  
mdadm /dev/md127 --remove /dev/nvme0n1 # remove drive  
  
mdadm --detail /dev/md127 # double check
```

```
shutdown -h now # power off host
```

Swap disk out.

```
sgdisk -R=/dev/nvme0n1 /dev/sda # clone partiion layout of /dev/sda  
sgdisk -G /dev/nvme0n1 # Give it a new unique GPT GUID
```

This will mirror the partitions and metadata exactly. The commands assume that the new drive is still /dev/nvme0n1.

```
mdadm /dev/md127 --add /dev/nvme0n1 # add new disk to the RAID  
watch cat /proc/mdstat # check rebuild progress
```

Wait until [UU] is shown again and rebuild completes (could take 15-60+ minutes depending on system load and speed).

```
mdadm --grow /dev/md127 --size=max # grow the raid array to use the full disk  
mdadm --detail /dev/md127 # verify that its has been successful
```

Resize the Partition Table on /dev/md127

```
gdisk /dev/md127
```

- In `gdisk`:
  - `p` — print partition table (record start of partition 3)
  - `d` — delete partition 3
  - `n` — create new partition:
    - Partition number: 3
    - First sector: type the exact start sector (e.g., `75497472`)
    - Last sector: just press Enter to accept default (uses remaining space)
    - Hex code: default (`8e00` for LVM)
  - `w` — write and exit

Double-check the start sector of `md127p3` beforehand using `lsblk -o NAME,START,SIZE` or `gdisk`

```
pvresize /dev/md127p3 # resize the LVM PV
```

Expand Your Volume Group or Logical Volumes

```
vgs # verify free space  
lvextend -l +100%FREE /dev/VG_XenStorage-xxx/VHD-xxxxx #example of estending the volume  
resize2fs /dev/VG_XenStorage-xxx/VHD-xxxxx # grow filesystem if needed (EXT4)  
xfs_growfs /mount/point # (if XFS)
```

You can leave the free space in the VG for XenServer to manage VM disks.

## Final checks

```
mdadm --detail --scan >> /etc/mdadm.conf # update mdadm config
```

Double-check everything work;

```
cat /proc/mdstat  
vgs  
lvs  
df -h
```

## Other notes

- `VG_XenStorage-xxx`: **The Volume Group (VG) name**

```
vgs
```

- `VHD-xxxxxx`: **The Logical Volume (LV) name**

```
lvs # can add VG name
```

Crafted with the assistance of <https://chatgpt.com/share/68328b0c-5094-8001-b27a-d7a27a61cdfc>

# Cloudflare Dynamic DNS IP Updater

## Installation

```
git clone https://github.com/K0p1-Git/cloudflare-ddns-updater.git
```

## Usage

This script is used with crontab. Specify the frequency of execution through crontab.

```
# ┌────────── minute (0 - 59)
# | ┌────────── hour (0 - 23)
# | | ┌────────── day of the month (1 - 31)
# | | | ┌────────── month (1 - 12)
# | | | | ┌────────── day of the week (0 - 6) (Sunday to Saturday 7 is also Sunday on some
systems)
# | | | | | ┌────────── command to issue
# | | | | |
# | | | | |
# * * * * * /bin/bash {Location of the script}
```

### • Tested Environments:

- macOS Mojave version 10.14.6 (x86\_64)
- AlmaLinux 9.3 (Linux kernel: 5.14.0 | x86\_64)
- Debian Bullseye 11 (Linux kernel: 6.1.28 | aarch64)

## Contributing

Pull requests are welcome. For major changes, please open an issue first to discuss what you would like to change.

## Reference

This script was made with reference from [Keld Norman](

<https://www.youtube.com/watch?v=vSIBkH7sxos>) video.

## License

[MIT](<https://github.com/K0p1-Git/cloudflare-ddns-updater/blob/main/LICENSE>)

## Actual Script

```
#!/bin/bash
## change to "bin/sh" when necessary

## Add below to see logs
##set -x # Echo each command before execution
##exec > /tmp/cloudflare-debug.log 2>&1 # Redirect all output to a temp log
##echo "=== Script Started: $(date) ==="

auth_email="cloudflare@login.com" # The email used to login
'https://dash.cloudflare.com'
auth_method="global" # Set to "global" for Global API Key or
"token" for Scoped API Token
auth_key="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" # Your API Token or Global API Key
zone_identifier="yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy" # Can be found in the "Overview" tab of
your domain
record_name="sub.domain.com" # Which record you want to
be synced
ttl="3600" # Set the DNS TTL (seconds)
proxy="false" # Set the proxy to true or false
sitename="" # Title of site "Example Site"
slackchannel="" # Slack Channel #example
slackuri="" # URI for Slack WebHook
'https://hooks.slack.com/services/xxxxx'
discorduri="" # URI for Discord WebHook
'https://discordapp.com/api/webhooks/xxxxx'

#####
## Check if we have a public IP
#####
ipv4_regex='([01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.([01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.([01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.([01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])'
ip=$(curl -s -4 https://cloudflare.com/cdn-cgi/trace | grep -E '^ip'); ret=$?
if [[ ! $ret == 0 ]]; then # In the case that cloudflare failed to return an ip.
    # Attempt to get the ip from other websites.
    ip=$(curl -s https://api.ipify.org || curl -s https://ipv4.icanhazip.com)
else
    # Extract just the ip from the ip line from cloudflare.
```

```

    ip=$(echo $ip | sed -E "s/^ip=($ipv4_regex)/\1/")
fi

# Use regex to check for proper IPv4 format.
if [[ ! $ip =~ ^$ipv4_regex$ ]]; then
    logger -s "DDNS Updater: Failed to find a valid IP."
    exit 2
fi

#####
## Check and set the proper auth header
#####
if [[ "${auth_method}" == "global" ]]; then
    auth_header="X-Auth-Key:"
else
    auth_header="Authorization: Bearer"
fi

#####
## Seek for the A record
#####

logger "DDNS Updater: Check Initiated"
record=$(curl -s -X GET
"https://api.cloudflare.com/client/v4/zones/$zone_identifler/dns_records?type=A&name=$record_n
ame" \
        -H "X-Auth-Email: $auth_email" \
        -H "$auth_header $auth_key" \
        -H "Content-Type: application/json")

#####
## Check if the domain has an A record
#####
if [[ $record == *"\count\":0"* ]]; then
    logger -s "DDNS Updater: Record does not exist, perhaps create one first? (${ip} for
${record_name})"
    exit 1
fi

#####

```

```

## Get existing IP
#####
old_ip=$(echo "$record" | sed -E 's/.*"content":"([0-9]{1,3}\.){3}[0-9]{1,3}).*/\1/')
# Compare if they're the same
if [[ $ip == $old_ip ]]; then
    logger "DDNS Updater: IP ($ip) for ${record_name} has not changed."
    exit 0
fi

#####
## Set the record identifier from result
#####
record_identifier=$(echo "$record" | sed -E 's/.*"id":"([A-Za-z0-9_]+).*/\1/')

#####
## Change the IP@Cloudflare using the API
#####
update=$(curl -s -X PATCH
"https://api.cloudflare.com/client/v4/zones/$zone_identifier/dns_records/$record_identifier" \
    -H "X-Auth-Email: $auth_email" \
    -H "$auth_header $auth_key" \
    -H "Content-Type: application/json" \
    --data
"{\"type\": \"A\", \"name\": \"$record_name\", \"content\": \"$ip\", \"ttl\": \"$ttl\", \"proxied\": ${
proxy}}")

#####
## Report the status
#####
case "$update" in
*"\"success\":false"*)
    echo -e "DDNS Updater: $ip $record_name DDNS failed for $record_identifier ($ip). DUMPING
RESULTS:\n$update" | logger -s
    if [[ $slackuri != "" ]]; then
        curl -L -X POST $slackuri \
            --data-raw '{
                "channel": "'$slackchannel'",
                "text": "'$sitename' DDNS Update Failed: '$record_name': '$record_identifier'
($ip')."
            }'
    fi

```

```

fi
if [[ $discorduri != "" ]]; then
    curl -i -H "Accept: application/json" -H "Content-Type:application/json" -X POST \
    --data-raw '{
        "content" : ""'$sitename'" DDNS Update Failed: '$record_name': '$record_identifie'
('$ip')."
    }' $discorduri
fi
exit 1;;
*)
logger "DDNS Updater: $ip $record_name DDNS updated."
if [[ $slackuri != "" ]]; then
    curl -L -X POST $slackuri \
    --data-raw '{
        "channel": ""'$slackchannel'",
        "text" : ""'$sitename'" Updated: '$record_name''''''s'''''' new IP Address is '$ip'"
    }'
fi
if [[ $discorduri != "" ]]; then
    curl -i -H "Accept: application/json" -H "Content-Type:application/json" -X POST \
    --data-raw '{
        "content" : ""'$sitename'" Updated: '$record_name''''''s'''''' new IP Address is '$ip'"
    }' $discorduri
fi
exit 0;;
esac

```

# OpenDNS Linux IP Updater

Install **ddclient**

/etc/ddclient.conf

```
##  
## OpenDNS.com account-configuration  
##  
protocol=dyndns2  
use=web, web=myip.dnsomatic.com  
ssl=yes  
server=updates.opendns.com  
login=opendns_username  
password='opendns_password' # If there are any single-quotes in your password, put backslash (  
\\ ) before the single-quote to escape the character.  
opendns_network_label #can find the network label in the Settings Tab of the OpenDNS Dashboard
```

<https://support.opendns.com/hc/en-us/articles/227987727-Linux-IP-Updater-for-Dynamic-Networks>