Cisco Cheetsheet

Cisco CLI Reference, Howtos, and Tools

Quick Navigtation

Quick Snippets & Scripts

1. Intialize
2. Basic Config
3. Assign Static IP to Interface
4. Snippet: Enable Router DHCP Server
5. Snippet: Enable Switch DHCP Server
6. Nuking (ROMMON, Password Recovery, etc)
7. Howto: File Transfer Over Console (linux / xmodem)
8. Access Console over USB on Linux

General Sections

- Basic Networking

```
+ [Basic Setup](#setup)
+ [Interfaces](#interfaces)
+ [DHCP](#dhcp)
```

* Intermediate Networking

```
  + [VLANs](#vlans)
  + [Trunks](#trunks)
  + [Etherchannel](#etherchannel)
  + [Dynamic Trunking Protocol (DTP)](#dtp-dynamic-trunking-protocol)
  + [Routing](#routing)
  + [Spanning Tree Protocol](#spaning-tree-protocol)
* [Advanced Networking](#advanced-networking)
  + [OSPFv2](#ospfv2)
* [How To's](#how-tos)
  + [FTP Server Usage](#ftp-server-usage)
  + [Access Console over USB on Linux](#access-console-over-usb-on-linux)
* [Tools](#tools)
```

Full Navigation

- Basic Networking
  - Setup
    - Intialize

        - Basic Switch Config
        - Basic Router Config
        - Basic Config with Password Security
        - Basic Security
        - Configure SSH
        - Set Clock
        - Basic Hardening (Work Needed)
        - Backup config over FTP
        - Backup config over console
        - Restore Config
        - Nuking (ROMMON, Password Recovery, etc)
    - Interfaces
        - Interface Selection
        - Assign Static IP to Interface
        - Interface Ranges
    - Interface Verification
        - Remove IP Addresses
    - Console Port
        - Change Console Baudrate
    - DHCP
        - Snippet: Enable Router DHCP Server
        - Snippet: Enable Switch DHCP Server
        - Create DHCP Pool
        - DHCP Verification
        - Disable DHCP
        - Re-enabled DHCP
        - Create VLAN DHCP
        - Verify DHCP Pool
        - Delete DHCP Pool
- Intermediate Networking
    - VLANs
        - VLAN Creation
        - Port Assignment
        - IP Assignemnt
        - Verification
        - Voice and Data VLAN
        - Management VLAN
        - Delete VLANS on file
        - Delete VLANS in memory
        - Inter-VLAN Routing
    - Trunks
        - Create multi-switch vlan trunk
        - Trunk Verification
    - EtherChannel
        - Configure EtherChannel
        - Verify EtherChannel
    - DTP (Dynamic Trunking Protocol)
        - Configure DTP
        - Disable DTP
        - Verify DTP
- Advanced Networking

    - OSPFv2
        - OSPF Router IDs
            - All Commands
            - Enable router OSPF process
            - Configure Loopback
            - Configure OSPF Router ID
            - Modify OSPF router ID
        - OSPF - Point-to-Point Networks
            - Network Command Syntax
            - Configure OSPF With Network Command
            - Use Entire Gigabit Interfaces
            - Configure OSPF with `ip ospf`
            - OSPF Passive Interfaces
            - Find Designated Router and Backup
            - Change OSPF from Broadcast to Point-to-Point
            - Loopback and P2P Networks
        - Multiaccess OSPF Networks
            - Configure OSPF Priority
        - Modifying Single Area OSPF
            - Adjusting Reference Bandwidth
            - Manually Set OSPF Link Cost
            - Show OSPF Hello Packet Intervals
            - Set OSPF Hello Packet Intervals
            - Set OSPF Dead Interval
        - OSPF Default Routes
            - Propogate Default Route
            - Verify Propogated Default Route
        - Verify Single-Area OSPF
            - Verify OSPF Neighbors
            - Verify OSPF Protocols
            - Verify OSPF Process Info
            - Verify OSPF Interface Setting
- How To's
    - FTP Server Usage
    - Install Packet Tracer on Fedora Workstation
    - Console Access with `minicom` on Linux
    - Configure Serial Port with `stty` on Linux
- Tools
    - Subnetting/Calcuation
        - ipcalc (*nix)
        - sipcalc (*nix)
        - whatmask (*nix)
          ## Basic Networking

Setup

Intialize

These commands wipe all config and reboot the device

```
erase startup-config
delete vlan.dat
reload
```

Note: Remeber to say “no” to saving running config on reload. If you say yes, running config will be saved and you wont be working with fresh config on reload.

Basic Switch Config

```
configure terminal
no ip domain-lookup
hostname S1
line console 0
logging synchronous
exit
banner motd $ Authorized Access Only! And Godzilla will beat Kong any day $
exit
copy running-config startup-config
```

Basic Router Config

```
configure terminal
no ip domain-lookup
hostname R1
line console 0
logging synchronous
exit
banner motd $ Authorized Access Only! And Godzilla will beat Kong any day $
exit
copy running-config startup-config
```

Basic Config with Password Security

pastable

```
configure terminal
no ip domain-lookup
hostname R1
line console 0
logging synchronous
exit
banner motd $ Authorized Access Only! And Godzilla will beat Kong any day $
exit
```

```
copy running-config startup-config
conf t
enable secret class
line console 0
password cisco
login
exit
line vty 0 4
password cisco
login
exit
service password-encryption
end
copy running-config startup-config
```

Basic Security

```

conf t
enable secret class
line console 0
password cisco
login
exit
line vty 0 4
password cisco
login
exit
service password-encryption
end
```

Configure SSH

```

show ip ssh
conf t
ip domain-name cisco.com
crypto key generate rsa

username admin secret ccna
line vty 0 15
transport input ssh
login local
exit
ip ssh version 2
exit
```

Set Clock

Show Clock

```
show clock
```

Sets clock to eastern US time

```
clock timezone EST -5
```

Revert to Default Timezone

```
no clock timezone
```

Basic Hardening (Work Needed)

```
conf t
! Logout timer
!
line con 0
 exec-timeout 5
line vty 0 4
 exec-timeout 5

exit

ip ssh time-out 60
ip ssh authentication-retries 3
end
```

Backup config over FTP

Using included FTP server

```
copy running-config startup-config
copy startup-config ftp://192.168.1.10/config.txt
```

Backup config over console

coming soon

Restore Config

```
copy ftp://192.168.1.10/config.txt running-config
```

Nuking (ROMMON, Password Recovery, etc)

Perform a Boot Interupt to Recover a lost or unknown password

WARNING: This operation will delete all current config on the device

1. Ensure Console Cable is connected at 9600 Baudrate
2. Backup config if you need
3. Unplug Power
4. Wait for a few seconds
5. Re-insert the power cord to the switch
6. Within 15 seconds, hold the `Mode` button until the green flashing light flashes amber and then returns to flashing green. Release the `Mode` button.
7. Something like the following should display:

   ```
   initialize the flash file system, and finish loading the operating
   system software# flash_init load_helper boot
   ```

8. Run `flash_init`
9. Run `copy flash:config.text flash:config.text.old`
10. Run `boot`

    The device should now boot with no config and grant you access to it.

Interfaces

Interface Selection

Assign and IP address to a port

```
conf t
int f0/1
ip addr 192.168.10.11 255.255.255.0
end
```

Assign Static IP to Interface

```
cont t
int g0/0
```

```
ip addr 10.0.0.10 255.255.255.0
```

Interface Ranges

Assign and IP address to a port

```
conf t
int f0/1
ip addr 192.168.10.11 255.255.255.0
end
```

Select Single Range and Assign to a VLAN

```
conf t
int range f0/1-12
switchport mode access
switch access vlan 10
end
```

```
conf t
int range f0/13-24
switchport mode access
switchport access vlan 20
end
```

Select Multiple Interface Ranges and Move to a VLAN

```
conf t
int range f0/1-4,g0/1,f0/16-20
switchport mode access
switchport access vlan 10
end
```

Interface Verification

```
show ip interface brief
```

or

```
show ip int br
```

Remove IP Addresses

```
conf t
int f0/1
no ip addr
end
```

Console Port

Change Console Baudrate

```
conf t
line con 0
speed 115200
end
```

```
conf t
line con 0
speed 9600
end
```

DHCP

Snippet: Enable Router DHCP Server

This snippet configures a DHCP Server on R1 and will hand out IPs on the `10.0.0.1/24` network. Great for using an FTP Server with.

```
conf t
ip domain name cisco.com
ip dhcp excluded-address 10.0.0.1
ip dhcp pool test
network 10.0.0.0 255.255.255.0
default-router 10.0.0.1
end
```

Snippet: Enable Switch DHCP Server

```
ip dhcp pool test
network 10.0.0.0 255.255.255.0
domain-name cisco.com
```

```
default-router 10.0.0.1
dns-server 10.0.0.1
lease 4
ip dhcp snooping
ip dhcp-server 10.0.0.3
interface vlan 1
ip address 10.0.0.3
```

Create DHCP Pool

Workaround for CCNA labs at Liberty University since we can't change the LAB IP addresses

```
conf t
ip domain name cisco.com
ip dhcp excluded-address 10.0.0.1
ip dhcp pool managementpool
network 10.0.0.1 255.255.255.0
default-router 10.0.0.1
end
```

```
conf t
ip dhcp excluded-address 192.168.10.1
ip dhcp excluded-address 192.168.10.254
ip dhcp pool office-pool-1
network 192.168.10.0 255.255.255.0
default-router 192.168.10.1
dns-server 192.168.5.5
domain-name linux.org
end
```

DHCP Verification

```
show running-config | section dhcp
show ip dhcp binding
show ip dhcp server statistics
```

Disable DHCP

```
conf t
no service dhcp
end
```

Re-enabled DHCP

```
conf t
service dhcp
end
```

Create VLAN DHCP

Creates a Seperate DHCP Pool for each VLAN

Create VLANS

```
conf t
vlan 10
name Management
vlan 20
name Sales
vlan 30
name Operations
end
```

Configure SVI's and IP Address

VLAN	IP Address	
——	—————	——–
10	192.168.10.254	
20	192.168.20.254	192.168.20.1
30	192.168.30.254	192.168.30.1

```
conf t
int vlan 10
ip address 192.168.10.254 255.255.255.0
ip default-gateway 192.168.10.1
no shut

int vlan 20
ip address 192.168.20.254 255.255.255.0
ip default-gateway 192.168.20.1
no shut

int vlan 30
ip address 192.168.30.254 255.255.255.0
ip default-gateway 192.168.30.1
no shut
end
```

Add interfaces to VLANS, 8 ports per vlan

```
conf t
int range f0/1-7
switchport mode access
switchport access vlan 10

int range f0/8-15
switchport mode access
switchport access vlan 20

int range f0/16-24
switchport mode access
switchport access vlan 30
end
```

Create DHCP Pools for each vlan

```
conf t
ip domain name cisco.com
ip dhcp excluded-address 192.168.10.1
ip dhcp pool vlan10pool
network 192.168.10.0 255.255.255.0
default-router 192.168.10.1
import all


ip dhcp excluded-address 192.168.20.1
ip dhcp pool vlan20pool
network 192.168.20.0 255.255.255.0
default-router 192.168.20.1
import all

ip dhcp excluded-address 192.168.30.1
ip dhcp pool vlan30pool
network 192.168.30.0 255.255.255.0
default-router 192.168.30.1
import all
end
```

Now when a device plugs into a port `f0/4` for instance and performs a DHCP request, it should get an IP like `192.168.10.3` because it is plugged into the ports assigned to VLAN 10

Verify DHCP Pool

```
show ip dhcp pool
```

Delete DHCP Pool

```

```

```
conf t
no ip dhcp pool managementpool
end
```

Intermediate Networking

VLANs

VLAN Creation

```
conf t
vlan 10
name Faculty
exit
```

```
conf t
vlan 20
name Students
exit
```

Port Assignment

```
conf t
interface range Fa0/1-12
switchport mode access
switchport access vlan 10
end
```

```
conf t interface range Fa0/13-24
switchport mode access
switchport access vlan 20
end
```

```
conf t
interface Gi0/1
switchport mode access
switchport access vlan 99
end
```

IP Assignemnt

```
```

```
cont t
int vlan 99
ip address 10.0.0.1 255.255.255.0
end
```

Verification

```
show vlan brief
```

Voice and Data VLAN

Assuming Data on VLAN 10, Voice on VLAN 20

```
conf t
int Fa0/4
switchport mode access
switchport access vlan 10
switchport voice vlan 20
end
```

Management VLAN

```
conf t
vlan 99
name Management
exit
interface Fa0/24
switchport mode access
switchport access vlan 99
exit
int vlan 99
ip addr 10.0.0.1 255.255.255.0
end
```

Delete VLANS on file

```
delete vlan.dat
```

Delete VLANS in memory

Warning: Make sure you move ports to another vlan or the will be unsable

```
```

```
conf t
no vlan 10
no vlan 20
end
```

Inter-VLAN Routing

Creates multiple sub-interfaces on a router port to enable inter-vlan routing.

Note: `encapsulation dot1q` must be called on a sub interface before an IP can be assigned to it.

```
conf t
interface G0/0/1.10
description Default Gateway for VLAN 10
encapsulation dot1Q 10
ip add 192.168.10.1 255.255.255.0
exit

interface G0/0/1.20
description Default Gateway for VLAN 20
encapsulation dot1Q 20
ip addr 192.168.20.1 255.255.255.0
exit

interface G0/0/1.99
description Default Gateway for VLAN 99
encapsulation dot1Q 99
ip addr 192.168.99.1 255.255.255.0
exit

interface G0/0/1
description Trunk link to S1
no shut
end
```

Trunks

Create multi-switch vlan trunk

S1

```
conf t
interface Gi0/1
description Trunk Line to S2 Gi0/1
switchport mode trunk
```

```
switchport trunk native vlan 99
switchport trunk allowed vlan 99
end
```

Note: Remember to set the native vlan (to 99 for instance) on each switch in the trunk so you don't get a native vlan mismatch warning

Trunk Verification

```
show interface trunk
show interface g0/1 switchport
```

EtherChannel

Etherchannel protocols LACP and PAgP configure multiple physical interfaces and links to act as one logical one. You can configure up to 8 ports to act as a single link. This increases bandwidth and improves redundancy.

Note: `mode active` sets the etherchannel group to use the LACP protocol

Configure EtherChannel

Configure etherchannel between two switches connected with two ethernet cables.

```
conf t
int range f0/1-2
channel-group 1 mode active
exit
int port-channel 1
switchport mode trunk
switchport trunk allowed vlan 1,2,20
```

Verify EtherChannel

```
show interfaces trunk
show etherchannel summary
```

DTP (Dynamic Trunking Protocol)

Configure DTP

```
conf t
int gi0/1
switchport mode dynamic auto
end
```

or

```
conf t
int gi0/1
switchport mode dynamic desirable
end
```

Disable DTP

Usefull for connecting to devices that don't support Cisco propietary DTP or creating a static trunk

```
conf t
int gi0/1
switchport mode trunk
switchport nonegotiate
end
```

Verify DTP

```
show dtp interface gi0/1
```

Advanced Networking

OSPFv2

OSPF Router IDs

All Commands

```
show ip ospf neighbor
show ip ospf database
```

Enable router OSPF process

Starting Mode: Global, Non-enabled

```
enable
conf t
router ospf 10
```

Configure Loopback

```
enable
conf t
interface Loopback 1
ip addr 1.1.1.1 255.255.255.255
end
```

Configure OSPF Router ID

replace 1.1.1.1 with desired id

```
conf t
router ospf 10
router-id 1.1.1.1
end
```

Modify OSPF router ID

Prompt confirmation with 'y' needed

```
conf t
router ospf 10
router-id 1.1.1.2
end
clear ip ospf process
```

Verify

```
show ip proto | include Router ID
```

OSPF - Point-to-Point Networks

Network Command Syntax

```
Router(config-router)# network network-address wildcard-mask area area-id
```

Configure OSPF With Network Command

The following configures a trianngle of 3 routers connected to each other as an OSPF point to point network.

```
conf t
router ospf 10
network 10.10.1.0 0.0.0.255 area 0
network 10.10.1.4 0.0.0.3 area 0
network 10.10.1.12 0.0.0.3 area 0
end
```

Use Entire Gigabit Interfaces

```
conf t
router ospf 10
network 10.10.1.1 0.0.0.0 area 0
network 10.10.1.5 0.0.0.0 area 0
network 10.10.1.14 0.0.0.0 area 0
end
```

Configure OSPF with `ip ospf`

Configure OSPF directly on the interfaces rather with with the network command.

Syntax: `Router(config-if)# ip ospf <process-id> area <area-id>`

```
R1(config)# router ospf 10
R1(config-router)# no network 10.10.1.1 0.0.0.0 area 0
R1(config-router)# no network 10.1.1.5 0.0.0.0 area 0
R1(config-router)# no network 10.1.1.14 0.0.0.0 area 0
R1(config-router)# interface GigabitEthernet 0/0/0
R1(config-if)# ip ospf 10 area 0
R1(config-if)# interface GigabitEthernet 0/0/1
R1(config-if)# ip ospf 10 area 0
R1(config-if)# interface Loopback 0
R1(config-if)# ip ospf 10 area 0
R1(config-if)#
```

OSPF Passive Interfaces

```
conf t
router ospf 10
passive-interface loopback 0
end
```

```
conf t
router ospf 10
passive-interface Gi0/0/0
end
```

Find Designated Router and Backup

```
show ip ospf interface GigabitEthernet 0/0/0
```

Change OSPF from Broadcast to Point-to-Point

```
conf t
interface GigabitEthernet 0/0/0
ip ospf network point-to-point
```

Loopback and P2P Networks

Loobacks can be used to simulate real LAN networks

```
conf t
interface Loopback 0
ip ospf network point-to-point
```

```
show ip route | include 10.10.1
```

Multiaccess OSPF Networks

Configure OSPF Priority

```
conf t
int g0/0/1
ip ospf priority 255
end
```

Where 255 can be values from 0 to 255 with higher numbers making the router to be elected DR.

Modifying Single Area OSPF

Adjusting Reference Bandwidth

```
Router# router ospf 10
Router(config-router) auto-cost reference bandwidth 1000
```

Where 1000 is the speed of the link in Mpbs Common Values: 10, 100, 1000

Manually Set OSPF Link Cost

```
conf t
int g0/0/1
ip ospf cost 25
interface l0
ip ospf cost 15
end
```

Show OSPF Hello Packet Intervals

```
show ip ospf int g0/0/1
```

Set OSPF Hello Packet Intervals

```
Router(config-if)# ip ospf hello-interval <seconds>
```

```
conf t
int g0/0/1
ip ospf hello-interval 30
end
```

Note: dead-interval automatically gets set as `hello-interval * 4`

Set OSPF Dead Interval

OSPF Default Routes

Propogate Default Route

```
conf t
ip route 0.0.0.0 0.0.0.0 loopback 1
router ospf 10
default-information originate
```

Verify Propogated Default Route

```
show ip route | begin Gateway
```

Verify Single-Area OSPF

Verify OSPF Neighbors

```
show ip ospf neighbor
```

Verify OSPF Protocols

```
show ip protocols
```

Verify OSPF Process Info

```
show ip ospf
```

Verify OSPF Interface Setting

```
show ip ospf int g0/0/1
show ip ospf int brief
```

Where `g0/0/1` is the interface you was to see OSPF information on.

```
conf t
int g0/0/1
ip ospf dead-interval 100
end
```

How To's

FTP Server Usage

1. Clone the repo:

   ```
   git clone https://github.com/grplyler/cisco-utils
   ```
   2. Install python requirements (for ftp server):

   ```
   pip install -r requirements.txt
   ```
   3. Run python ftp_server.py

   ```
   python3 ftp_server.py
   ```
   4. Pull a script onto a network device (WARNING: Backup to avoid any losses)

   ```
   Switch#> copy ftp://192.168.1.10/sw_base.txt running-config
   ```
   Replace 192.168.1.10 with the IP of the computer connected to the switch or router.

Install Packet Tracer on Fedora Workstation

(Credit for this howto goes to `philpinch` from the Fedora Forums)

1. Log into the Fedora GNOME Desktop

Remove old version of PacketTracer (if necessary):

```
rm -rf /opt/pt
rm -rf /usr/share/applications/cisco-pt7.desktop
rm -rf /usr/share/applications/cisco-ptsa7.desktop
rm -rf /usr/share/icons/hicolor/48x48/apps/pt7.png
```

2. Download from the netacad web site the PacketTracer*730*amd64.deb package.

Open a terminal :

```
mkdir -p tmp/pt730
```

copy the PacketTracer*730*amd64.deb package to tmp/pt730

4. We’re going to extract the deb file in this folder:

```
cd tmp/pt730
ar -xv PacketTracer_730_amd64.deb
mkdir control
tar -C control -Jxf control.tar.xz
mkdir data
```

```
tar -C data -Jxf data.tar.xz
```

5. Copy PacketTracer files to install it:

```
cd data
cp -r usr /
cp -r opt /
```

6. Configure Gnome Environment:

```
sudo xdg-desktop-menu install /usr/share/applications/cisco-pt7.desktop
sudo xdg-desktop-menu install /usr/share/applications/cisco-ptsa7.desktop
sudo update-mime-database /usr/share/mime
sudo gtk-update-icon-cache --force --ignore-theme-index
/usr/share/icons/gnome
sudo xdg-mime default cisco-ptsa7.desktop x-scheme-handler/pttp
ln -sf /opt/pt/packettracer /usr/local/bin/packettracer
```

Console Access with `minicom` on Linux

under construction

Configure Serial Port with `stty` on Linux

Set the default configuration with stty to cisco console default, 9600 bps, 8N1, no flow control:

```
stty -F /dev/ttyUSB0 9600 litout -crtscts
```

or:

```
stty -F /dev/ttyUSB0 cs8 -parenb -cstopb -echo raw speed 9600

 # What the arguments mean:
 #   cs8:     8 data bits
 #   -parenb: No parity (because of the '-')
 #   -cstopb: 1 stop bit (because of the '-')
 #   -echo: Without this option, Linux will sometimes automatically send
back
 #          any received characters, even if you are just reading from the
serial
 #          port with a command like 'cat'. Some terminals will print codes
 #          like "^B" when receiving back a character like ASCII ETX (hex
03).
 ```
```
```

```
### Console Access with Screen on Linux

For this you will need a USB console cable. These can be picked up
on amazon for about $9-$12.

1. Connect your the USB console cable from the computers usb port to the
cisco RJ-45 console port.

2. Install the `screen` program if you dont already have it.
```

apt install screen

```


3. Find the USB device.

If its the first USB serial device you plugged in, it should be
`/dev/ttyUSB0`. The second one should be `/dev/ttyUSB1`, etc.

You can verify with with `ls /dev | grep USB`

4. Run `screen`

You will need root access.
```

screen /dev/ttyUSB0

```


Running with a specific baudrate.
```

screen /dev/ttyUSB0 9600 screen /dev/ttyUSB0 115200

```


To exit screen, hit `Ctrl-a`, `Ctrl-d`

If you have trouble with the connection, e.g. it lags or is funky, cisco
serial connections require the following settings by default:

- `9600` baud
- `8` data bits
- `no` parity
- `1` stop bit
- `no` flow control

To do that exactly with screen:
```

screen /dev/ttyS0 9600,cs8,-parenb,-cstopb,-hupcl screen /dev/ttyS0 19200,cs8,-parenb,-cstopb,-hupcl screen /dev/ttyS0 115200,cs8,-parenb,-cstopb,-hupcl

```
```

```
With `odd` parity:
```

screen /dev/ttyS0 9600,cs8,parenb,parodd,-cstopb,-hupcl

```
With `even` parity:
```

screen /dev/ttyS0 9600,cs8,parenb,-parodd,-cstopb,-hupcl

```
See more details at
[http://www.noah.org/wiki/Screen_notes](http://www.noah.org/wiki/Screen_notes)

### Linux File Transfer Over Console (minicom / xmodem)

_Howto comming soon!_

### Windows File Transfer Over Console ( HyperTerminal / xmodem)

_Howto comming soon!_

## Tools

### Subnetting/Calcuation

#### ipcalc (*nix)

Debian/Ubuntu
```

apt install ipcalc

```
Fedora
```

dnf install ipcalc

```

```
yum install ipcalc
```

**sipcalc (*nix)**

Debian/Ubuntu

```

```
apt install sipcalc
```

or replace `apt` with your package manager

whatmask (*nix)

Example Usage

```
$ whatmask 10.0.1.12/30

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
           TCP/IP NETWORK INFORMATION
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
IP Entered = ..................: 10.0.1.12
CIDR = ........................: /30
Netmask = .....................: 255.255.255.252
Netmask (hex) = ...............: 0xfffffffc
Wildcard Bits = ...............: 0.0.0.3
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Network Address = .............: 10.0.1.12
Broadcast Address = ...........: 10.0.1.15
Usable IP Addresses = .........: 2
First Usable IP Address = .....: 10.0.1.13
Last Usable IP Address = ......: 10.0.1.14
```

Install

Debian/Ubuntu

```
apt install whatmask
```

or replace `apt` with your package manager