# Ubuntu: Setup Static IP Address

Here are notes and steps to assign a static IP address for an Ubuntu host.

These steps are known to work with Ubuntu 22 and 24.

> NOTE: If you are doing for a Debian host, see this page: Debian: Setup Static IP Address

## Net Tools (ipconfig)

Install net tools, so we can use commands like: ipconfig

```
sudo apt install net-tools
```

## Static IP Address

Static addresses will be defined in a netplan file, configured below.

But, we need to do a few things, first:

- Get the Gateway IP
- Enable Adapters

## Get the Gateway IP

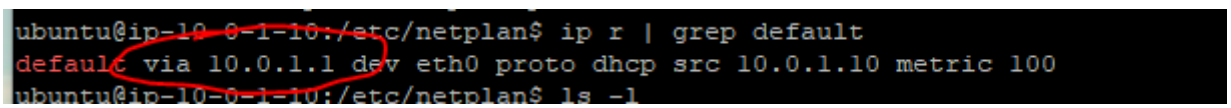
You will need to know the gateway address that your host will use.

If the host is not on the desired network, you will need to determine the gateway IP, manually. Or, you can join the host to the network, and do the following to determine it.

If the host is up, you can run this command to get the current default gateway assigned to it:

```
ip r | grep default
```

This command will return the default gateway address, like this:



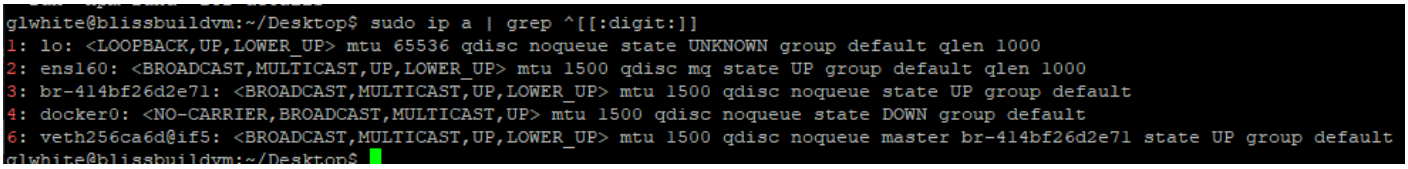
## Enable Adapters

If you added an adapter to the host, it may be in a down state.

To enable it, use this command to find the name:

```
sudo ip a | grep ^[[:digit:]]
```

The above will give a list, like this:




And, you can enable it with this:

```
sudo ifconfig eth1 up
```

> NOTE: Be sure to use the name of the nic, from the previous call.

## NetPlan YML

We will assign our static address, via NetPlan.

Open the netplan file, so you can edit the network config.
It will be located here:

`/etc/netplan/`

# Existing Cloud YML

If this folder contains a cloud yaml, you may have to disable cloud network config.
To do so, create a file at:

`/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`

And, give it this content:

`network: {config: disabled}`

Delete the existing cloud yml file, so it does not interfere with the config you will build, next.

Now, you can write a config yaml file that will be accepted through a reboot.

# Editing NetPlan YML

If a non-cloud netplan yml file exists, you may be able to edit it. The file on your host may have a different name, depending on OS version.
Check the filename before running this command.

Here’s the name of a valid netplan file that can be created, if one does not already exist.

`sudo nano /etc/netplan/01-network-manager-all.yaml`

Edit the file to reflect the following.
Be sure to use your ethernet adapter name (enps3s0), or whatever its name was discovered to be earlier.

Create the netplan file with the above name, `01-network-manager-all.yaml`

Set it up with your ether adapter names, static address, DNS, and gateway addresses.

The following are examples you can use:

```
network:
    version: 2
    renderer: networkd
    ethernets:
        enp3s0:
            addresses:
                - 10.10.10.2/24
            nameservers:
                search: [mydomain, otherdomain]
                addresses: [10.10.10.1, 1.1.1.1]
            routes:
                - to: default
                  via: 10.10.10.1
```

Here’s a working netplan file, in use on AWS, that has two NICs (public and private subnets):

```
# This sample has two network adapters, each in different subnets.
# The first NIC has a configured gateway and DNS.
# The second NIC is in a private subnet.
network:
  version: 2
  renderer: networkd
```

```
  ethernets:
    eth0:
      addresses:
        - 10.0.1.10/24
      nameservers:
        addresses: [127.0.0.53,8.8.8.8,8.8.4.4]
      routes:
        - to: default
          via: 10.0.1.1
    eth1:
      addresses:
        - 10.0.2.10/20
```

Create the Save changes to the file.

Make sure the created netplan file is only accessible by root, by doing this:

```
sudo chmod 600 ./01-network-manager-all.yaml
```

And, apply changes with:

```
sudo netplan apply
```

Check that connectivity exists through a reboot, to make sure settings persist.

---

Revision #4
Created 23 February 2025 01:26:07 by glwhite
Updated 7 May 2026 04:44:46 by glwhite