# C-002
# Regarding Configuration of Ubuntu Server for Development

Vincent W. Finley*

June 2025

Hosting a software development environment on a remote headless server increases opportunity for developer collaboration.

Setup differences between remote Ubuntu server and desktop installations are described herein. Manual configuration of Netplan is investigated.

## 1 Introduction

The previous article[1] described how to setup an Ubuntu desktop distribution as a VM (Virtual Machine) and access it from VSCode (Visual Studio Code).

While it may make sense to install a desktop distro locally on a laptop, it makes less sense to install it on a remote shared development environment. Why? Because desktop distributions provide a GUI (Graphical User Interface) that need either local console access or a remote desktop client.

Sure, installing a desktop distro works fine if you are using the desktop GUI on a laptop, and the laptop is being used as a terminal. The single lucky user currently logged into the desktop GUI at the console will have benefit of the physical: keyboard, mouse and video display. Other users, logged in via remote desktop connections, will experience a slow awkward desktop session.

So, it would make more sense for users to:

- Install a desktop distro locally on their laptops.
- Use the local desktop GUI as only as a terminal.
- Connect remotely to a server running a server distro.
- Perform any work on the server.

Typically a server distro gets installed on fast hardware with vast resources. When installed on powerful server hardware, users can experience fast compute and compile cycles. The time to



[1] C-001: Regarding Setup of a Development Environment, https://github.com/vwfinley/regarding/blob/main/C/C-001/C-001.pdf

perform work on a powerful server can be significantly less than doing it locally on an underpowered laptop.

The other advantage of server distros is they tend to be smaller than desktop distros. How can this be? Server distros do not include all the graphical packages and libraries needed to display a desktop GUI. Less packages equals smaller footprint and easier maintenance.

## 2 Solution

You are probably working on either a laptop or desktop PC running Microsoft Windows. This was the same situation encountered in the previous article.

Microsoft Windows already has a desktop. As in the previous article, we will use the Windows desktop the as a "terminal" that can connect to remote systems.

However this time instead of installing Ubuntu Desktop in a VM, we will install Ubuntu Server in the VM. Ubuntu Server will act as our remote development system.

We can describe Ubuntu Server as "headless" because it does not include a GUI. A headless server installation is more appropriate place for teams of developers to collaborate.

Okay, so what's our strategy to setup a remote development environment? We will:

- Follow many of the steps found in the previous article.
- Install an Ubuntu server distro rather than the Ubuntu Desktop distro.
- Make a few changes that are unique to the Server distro.
- Connect VSCode to the remote Ubuntu Server instance.
- Complete the environment setup.

## 3 Ubuntu Server

Previously, our guest OS (Operating System) was Ubuntu 22.04.5 LTS **Desktop**. This time around our guest OS will be Ubuntu 22.04.5 LTS **Server**.

Let's download Ubuntu 22.04.5 LTS **Server** for AMD64 architecture from the Ubuntu project website. Or download directly from here:

```
https://ubuntu.com/download/server/thank-you?version=22.04.5
    &architecture=amd64&lts=true
```

In VirtualBox, or the hypervisor you are using, follow sections 1 thru 4 of the previous article.

Instead of naming the VM "alpha" we will name it "bravo". Also, when asked how much space to dedicate to storage increase storage to 40Gb or more. Why? Because we will be installing Docker, and our containers are hungry for storage.

Detailed instructions follow.

# 4 Installation of Ubuntu Server

In VirtualBox, or the hypervisor you are using:

1. Click Add from the menubar to add a new VM.
2. Give the VM a name (for example "bravo") and select the ISO image, in this case ubuntu-22.04.5-live-server-amd64.iso
3. Optionally check "Skip Unattended installation", or not. Check the box for a quick installation, or uncheck it to fine tune your Ubuntu configuration.
4. Expand the Hard Disk panel and Increase the hard disk size to match your development needs.
5. Expand the Hardware tab and increase the processors to at least two CPUs.

## 4.1 Attended Installation

If you checked the "Skip Unattended installation" box.

1. Click Finish
2. In VirtualBox, select the VM you just created and click the Start button on the menubar.
3. After the VM starts, the GNU GRUB bootloader prompt will appear.
4. From the prompt select, "Try or Install Ubuntu Server", and hit Enter.
5. Be patient while Ubuntu starts.
6. The installer process will prompt you for information.
7. When prompted select your language and hit Enter.
8. When prompted about updating the installer select "Continue without updating" and hit Enter.
9. When prompted select your keyboard configuration and Done, then hit Enter.
10. When prompted to "Choose type of installation" select the Default Ubuntu Server installation, then Done and hit Enter.
11. When prompted for Network configuration, select the Defaults, then Done and hit Enter.
12. When prompted for Proxy configuration, enter a proxy address if necessary for your network, otherwise leave it blank. Select Done and hit Enter.
13. When prompted for Mirror configuration, wait for mirror location to pass tests, then select Done and hit Enter.
14. When prompted for "Guided Storage configuration" leave the defaults, select Done and hit Enter.
15. When prompted for "Storage configuration" leave the defaults, select Done and hit Enter.
16. When prompted to "Confirm Destructive Action" select Continue and hit Enter.
17. When prompted for "Profile Configuration" populate the fields, select Done and hit Enter.

18. When prompted to "Upgrade to Ubuntu Pro" select "Skip for now", select Continue and hit Enter.
19. When prompted for "SSH configuration", select "Install OpenSSH Server", then select Done and hit Enter.
20. When prompted for "Featured Server Snaps", leave all blank, then select Done and hit Enter.
21. Be patient while the system installs (30+ minutes).
22. When installation has completed, select "Reboot Now" and hit Enter.
23. In you get a message "Failed to unmount CDROM", hit Enter.
24. After system start up, login at the prompt.
25. Once you've logged in, verify you can reach the internet by entering:

```
$ ping google.com
```

26. Shutdown the VM by entering:

```
$ sudo shutdown now
```

### 4.2 Unattended Installation

If you left the "Skip Unattended installation" unchecked:

1. Expand the Unattended Install panel.
2. Set your user name and a password you will remember.
3. Optionally check install in background.
4. Optionally set the domain name.
5. Click Finish.
6. Wait while the VM boots Ubuntu and begins the installation cycle.
7. Be patient, wait for about 30 minutes while installation completes.
8. It may not be obvious installation has completed. You may need to occasionally hit the enter key to see the login prompt.
9. When the login prompt appears, try logging into the VM with the username and password you had previously set.
10. Once you've logged in, verify you can reach the internet by entering:

```
$ ping google.com
```

11. Shutdown the VM by entering:

```
$ sudo shutdown now
```

## 5 Network Setup

Section 6 of the previous article provided instructions for setting up the network. Those steps assumed a GUI since we were working with a desktop version of Ubuntu.

This time around we are working with a headless server version of Ubuntu. So, we will need to configure the network manually by editing text files.

Let's start by doing section 6.1 from the previous article to setup a second virtual ethernet adapter as a bridged network adapter.

You may recall previously, in Ubuntu Desktop the second virtual ethernet adapter is automatically setup to receive an IP (Internet Protocol) address from your local DHCP (Dynamic Host Configuration Protocol) server. However, in Ubuntu Server we need to manually setup this second adapter with an IP address.

We'll manually setup the second adapter to receive a dynamic IP address from the local DHCP server. We will do this to determine which network on the host (our laptop) to use for our bravo VM.

Once we know which network to use we'll manually set a static IP address. Setting a static IP address makes sense on a server because any clients need a reliable unchanging address to locate the server.

Let's start by discovering the virtual network adapters available to our VM. At the commandline on the VM enter:

```
$ ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
    ...
    ...
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state ...
    link/ether 08:00:27:31:cd:75 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 79566sec preferred_lft 79566sec
    ...
    ...
    ...
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state ...
    link/ether 08:00:27:c3:8e:b1 brd ff:ff:ff:ff:ff:ff
```

You should see output that looks similar to what is shown above. We are interested in the two vitual adapters created by VirtualBox. Above they start with the prefix "enp0s".

The vitual network adapter with ID=enp0s3 should have an IP4 address already set by VirtualBox. This adapter was defined by the first adapter tab in VirtualBox. In this case the address is 10.0.2.15 and the address was automatically assigned by DNS. VirtualBox has it setup for NAT

(Netork Address Translation). This adapter will be primarilly used for outbound connections to the internet from the VM, and inbound responses back to the VM.

You defined the second virtual network adapter with ID=enp0s8 on the second adapter tab in VirtualBox. Remember you had defined this adapter in VirtualBox with Bridge Host configruation. You'll notice that adapter enp0s8 has no IP address assigned, yet.

## 5.1 Netplan Configuration

This subsection summarizes setup of the network. It is based upon detailed information found on the internet[2].

The network manager used by Ubuntu is called Netplan[3]. It is published by Canonical[4], the company that maintains Ubuntu. Netplan is responsible for configuring any network adapters and getting an IP address assigned to them. Netplan uses one or more yaml files to configure how it will setup adapters.

Open the file /etc/netplan/50-cloud-init.yaml in an editor:

```
$ sudo vi /etc/netplan/50-cloud-init.yaml
```

Edit the file to add configuration for adapter enp0s8. Set it up to get its IP address from the DHCP server. Save the file and quit the editor.

```
network:
    ethernets:
        enp0s3:
            dhcp4: true
        enp0s8:
            dhcp4: true
    version: 2
```

Now restart netplan to see the changes applied.

[2]https://linuxconfig.org/how-to-configure-static-ip-address-on-ubuntu-18-04-bionic-beaver-linux
[3]https://netplan.io
[4]https://canonical.com

```
$ sudo netplan apply
$ ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    ...
    ...
    ...
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state ...
    link/ether 08:00:27:31:cd:75 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
    ...
    ...
    ...
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state ...
    link/ether 08:00:27:c3:8e:b1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.117/24 metric 100 brd 10.0.0.255 scope global dynamic enp0s8
    ...
    ...
    ...
```

### 5.2 Sanity Check

Notice that adapter enp0s8 has an IP address assigned? In the example above the IP address and CIDR (Classless Inter-Domain Routing) block is 10.0.0.117/24. This tells us that our bravo VM is connected to the Microsoft Windows host's network 10.0.0.X via adapter enp0s8.

To test everything is working okay, ping address 10.0.0.117 from the Windows host on the laptop.

In Windows open a CMD (Command) shell session and enter:

```
C:\> ping 10.0.0.117
```

You should see a reply from the bravo VM. Now write down the IP address that you pinged (10.0.0.117), you will need it for the next section.

## 6 cloud-init

Reboot our VM named "bravo". Log back in and view the /etc/netplan/50-cloud-init.yaml file.

You will see the change we made in the previous section was removed and now a notice appears at the top of the file. The notice says "...To disable cloud-init's network configuration capabilities...".

cloud-init[5] is a system for customizing the initialization of Cloud hosted operating systems. Like Netplan, it was written and published by Canonical.

[5] https://cloud-init.io/

cloud-init assumes Ubuntu Server will be running on a cloud platform. It wants to setup the server's network to align with a cloud hosting environment.

Though the Ubuntu Server distro we are using has Cloud capabilities baked-in, we are not interested in running on a cloud platform. So we will disable cloud-init's automatic network setup, and then manually setup our network.

### 6.1 Disable cloud-init network setup

To disable cloud-init's network setup, let's create a file at /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg containing the following line: "network: {config: disabled}".

Better yet, run the following script at the command line:

```
$ echo "network: {config: disabled}" \
    | sudo tee /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg > /dev/null
```

### 6.2 Set Static IP

Okay, again edit the /etc/netplan/50-cloud-init.yaml file as sudo.

This time change it so that virtual adapter enp0s8 has:

- dhcp4 set to false.
- the static IP address, you wrote down in the previous section (10.0.0.117/24), explicitly set.

Save the the /etc/netplan/50-cloud-init.yaml file file. It should appear as below.

```
network:
    ethernets:
        enp0s3:
            dhcp4: true
        enp0s8:
            dhcp4: false
            addresses:
                - 10.0.0.117/24
    version: 2
```

Now reboot the VM. When the VM is back up, ping the static IP address (10.0.0.117) from the laptop Windows host CMD shell. You should see a reply to the ping.

## 7 Other Setup

For the Ubuntu Server distro, the remaining steps to set up a development environment are same as for the Ubuntu Desktop distro.

To complete the setup, follow sections from the previous article:

- Section 7.2: Use the commandline to update software on the VM.
- Section 8: Save the VM's name and static IP address in the hosts file.

- Section 10 thru 17: Finish setting up the development environment.

## 8 Conclusion

Configuring Ubuntu Server as a remote development environment involves many of the same steps described in the previous article.

The main differences between using Ubuntu server vs. desktop are related to:

- Server distro installation procedures.
- Manual configuration of Netplan.
- Override of cloud-init's network auto-setup.

Choosing a server distro as development system reduces maintenance cost while increasing developer collaboration. It also facilitates relocation of the development environment to other hypervisor platforms when needed.