# Oracle Virtual Box setup guide

One of the biggest drawbacks of Hyper-V is that its NAT network generates new IPs every time you reboot. And there’s that horrendous bug in it as well.

VirtualBox allows you to create a NAT network (for external web access) with fixed IPs and a Host-only network with fixed IPs (for host-to-VM-access), which is limited to the PC itself, making it useful for our cloned classroom setups.

## Step 1: Install VBox

Yeah, yeah, I know, but just for completeness.

## Step 2: Create a NAT Network

In VBox, go **File** | **Preferences** | **Network** | Add. This will add a **Nat Network** with a CIDR block of **10.0.2.0/24** (at least it did on my box). You can click the “Edit” button to check or change the range, just make sure that later steps reflect that range.

## Step 3: Create a host only network

In VBox, go **File** | **Host Network Manager** and **Create** a **Host-Only adapter**. On my box, this gave me an adapter named “VirtualBox Host-Only Ethernet Adapter” with a range of **172.17.1.1/24**, so that’s the range I’ll be using.

## Step 4: Create a scratch Bunts 20.04 VM

Download the live CD from <https://ubuntu.com/download/server> (unless you want a desktop version)

In VBox manager, go **New**

In the **Create Virtual Machine** dialog, give it a **name** (I went with “Bunts 20 Scratch” but knock yourself out).

**Type** is Linux (obvs).

And **version** is “Ubuntu (64-bit)” (also obvs).

Click **Next**.

Give it 2GB RAM (probably enough for our purposes, but you can always scale up if necessary). **Next**

Create a new hard disk, VDI is probably fine, **Dynamically allocated** is the preference to save space. 20GB is probably preferable to 10.

Give it 2 vCPUs (otherwise K8s won’t install)

## Step 5: Install Bunts

You’ve got a VM now. Select it and choose **Settings** | **Storage** and mount your ISO file (it’s the little button with a CD icon next to the drop-down list).

Now go **Network** and attach **adapter 1** to the **Nat Network** and **adapter 2** to the Host-only adapter.

OK, **Start** it.

### Install Ubuntu.

Change the language and keyboard to English UK (or whatever is right for you)

### Network connections

For **enp0s3** (I’m hoping these adapters will get the same ids reliably, but the first adapter the installer asks you to configure), change the network settings to **manual**. Subnet should be **10.0.2.0/24** (but check your Nat Network range). Address should be something memorable but high. I went with **10.0.2.55**. Gateway is **10.0.2.1** (but check your NAT network). Name servers **8.8.8.8.** Leave search domains blank.

For **enp0s8** (or the second adapter), change the settings to **manual**. Subnet should be (but check) **172.17.1.0/24** and address again something memorable, I went with **172.17.1.55**. Leave the other stuff blank.

**Done**.

No proxy (unless you’re using one, I guess).

Done continue until you get to

### Profile setup

Your name: **student** (for the LF courses, otherwise use your own name)

Your server’s name: **scratch** (or whatevs)

Username: **student**

Password: I went with P@$$w0rd, but whatevs.

Done.

No Bunts advantage token.

Install OpenSSH, but none of the other optional packages.

Go and get yourself a cuppa.

Once the installation is done, Reboot (press enter)

## Step 6: Baseline setup

### Add student to sudoers and don’t require a password.

sudo echo "student ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/student

### Disable swap

sudo swapoff –a

sudo rm /swap.img

Remove the following line from **/etc/fstab**:

/swap.img blah

I don’t think VM guest additions brings much to the party, but you might want to install it:

In the **Devices** menu, **insert guest additions CD image**

sudo mount /dev/cdrom /media

cd /media

sudo ./VBoxLinuxAdditions.run

Once that’s done,

sudo reboot now

## Step 7: Start Cloning

Right-click on your scratch machine and choose **Clone**.

Give it a name (I’m going with K8s Controller, but whatevs)

For **MAC Address Policy**, choose “Generate new...”

Choose a path or leave the default.

Choose “**Full clone**”.

Spin it up.

### Change hostname

sudo hostnamectl set-hostname k8scp # (or whatevs)

### Change network settings

sudo vim /etc/netplan/00-installer-config.yaml

Set the addresses for the two interfaces. I’ve gone with 10.0.2.10 and 172.17.1.10 but knock yourself out. FYI, I went with .10 for k8scp, .11 for worker, .14 for cp2, .15 for cp3 and .16 for proxy, but you do you.

# This is the network config written by 'subiquity'

network:

ethernets:

enp0s3:

addresses:

- 10.0.2.55/24 # <-- change this

gateway4: 10.0.2.1

nameservers:

addresses:

- 8.8.8.8

search: []

enp0s8:

addresses:

- 172.17.1.55/24 # <-- and this

version: 2

Repeat however many times you want to, doing “sudo shutdown now” once you’re done.

Alternatively, you could run a script. I’ve called this **config-host**:

#!/bin/bash

usage() { echo "Usage: $0 -h hostname -n nataddress -a host-only-address " 1>&2; exit 1; }

while getopts :h:n:a: flag

do

case $flag in

a) hostonlyaddress=${OPTARG};;

h) hostname=${OPTARG};;

n) nataddress=${OPTARG};;

?) usage;;

esac

done

sudo hostnamectl set-hostname $hostname

sudo sed -i s/10.0.2.55/$nataddress/g /etc/netplan/00-installer-config.yaml

sudo sed -i s/172.17.1.55/$hostonlyaddress/g /etc/netplan/00-installer-config.yaml

sudo reboot now

Which you can then execute from your host machine like this:

ssh student@scratch \

"bash -s" -- < ./config-host -h k8scp3 -n 10.0.2.15 -a 172.17.1.15

For instance

## Step 8: Edit your hosts file

I mean, your host machine’s hosts file, not the guest VMs.

On Linux / WSL,

sudo vim /etc/hosts

On Windows, run notepad as **Administrator** and open **C:\Windows\System32\drivers\etc\hosts**.

Add the 172 entries with hostnames, so on my machine, that’s:

172.17.1.10 k8scp

172.17.1.11 worker

172.17.1.14 k8scp2

Etc...

Start your VMs headless and then

ssh student@k8scp

## Step 9: Install K8s

Duh. Use the 10.0.2.\* addresses internally, i.e. for all things Kubernetes. Bonus marks for adding:

localAPIEndpoint:  
 advertiseAddress: 10.0.2.10  
 bindPort: 6443

To your kube-config.yaml

Once you've joined the worker node(s) to the cluster, edit the worker node's **/etc/systemd/system/kubelet.service.d/10-kubeadm.conf** file to include the following:

[Service]  
Environment="KUBELET\_KUBECONFIG\_ARGS=--bootstrap-kubeconfig=/etc/kubernetes/bootstrap-kubelet.conf --kubeconfig=/etc/kubernetes/kubelet.conf"  
Environment="KUBELET\_CONFIG\_ARGS=--config=/var/lib/kubelet/config.yaml"  
Environment="KUBELET\_EXTRA\_ARGS=--node-ip=10.0.2.11" # ADD THIS LINE HERE

… More kubelet stuff

There shouldn’t be a problem with this VBox build, but it will help with a Hyper-V build.

Use the 172.17.1.\* addresses as the cluster’s “Public IP address” (i.e. to browse to a NodePort) and for SSHing into, or, if you got your hosts file right, use the hostname.