Install and Configure Chef Workstation, Chef Server, and a Chef Node

Table of Contents

Description

This lab guide provides step-by-step instructions for setting up a Chef ecosystem, including the installation and configuration of Chef Workstation, Chef Server, and a Chef Node. Chef is a powerful tool for automating system configurations, infrastructure management, and application deployments. By the end of this guide, you will have a functional Chef environment ready for configuration management tasks.

Problem Statement

Automating configuration management across multiple servers can be challenging without the right tools. Chef provides a streamlined way to define infrastructure as code, making server management efficient, consistent, and scalable. In this lab, we will:

- Install Chef Workstation to manage our Chef configurations.
- Configure Chef Server as the central repository for all Chef configurations.
- Set up Chef Node, which will be managed by Chef Server.

Prerequisites

- 64 bit operating system
- CPU Virtualization enabled

In order to virtualize a 64 bit operating system, one must also be running a 64 bit operating system. Most importantly, the CPU itself must support hardware virtualization extensions and this must be enabled in the

BIOS/EFI. Most modern processors support virtualization extensions in the form of VT-x (Intel) or AMD-V (AMD).

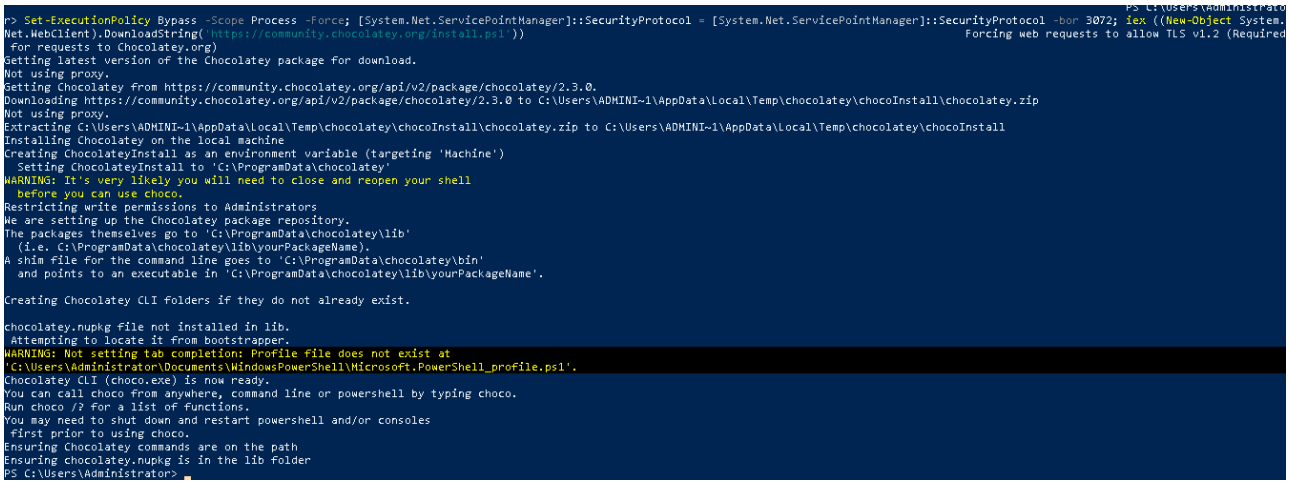
Implementation Steps

Step-1: Install Chocolatey, Virtual Box and Vagrant

1. Install Chocolatey

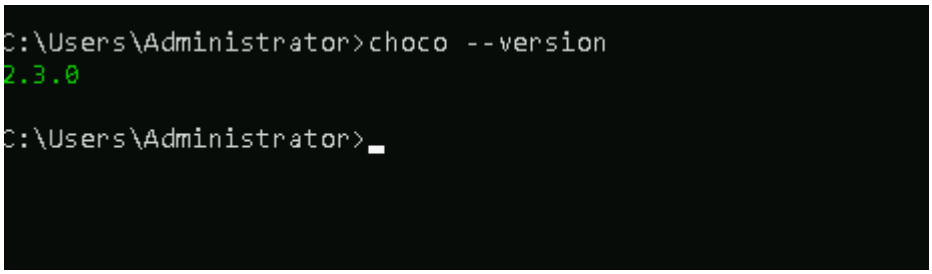
- To install Chocolatey, run the following command from the command line or from PowerShell:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-
Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/insta
ll.ps1'))
```



- Verify the installation by running the following command in cmd

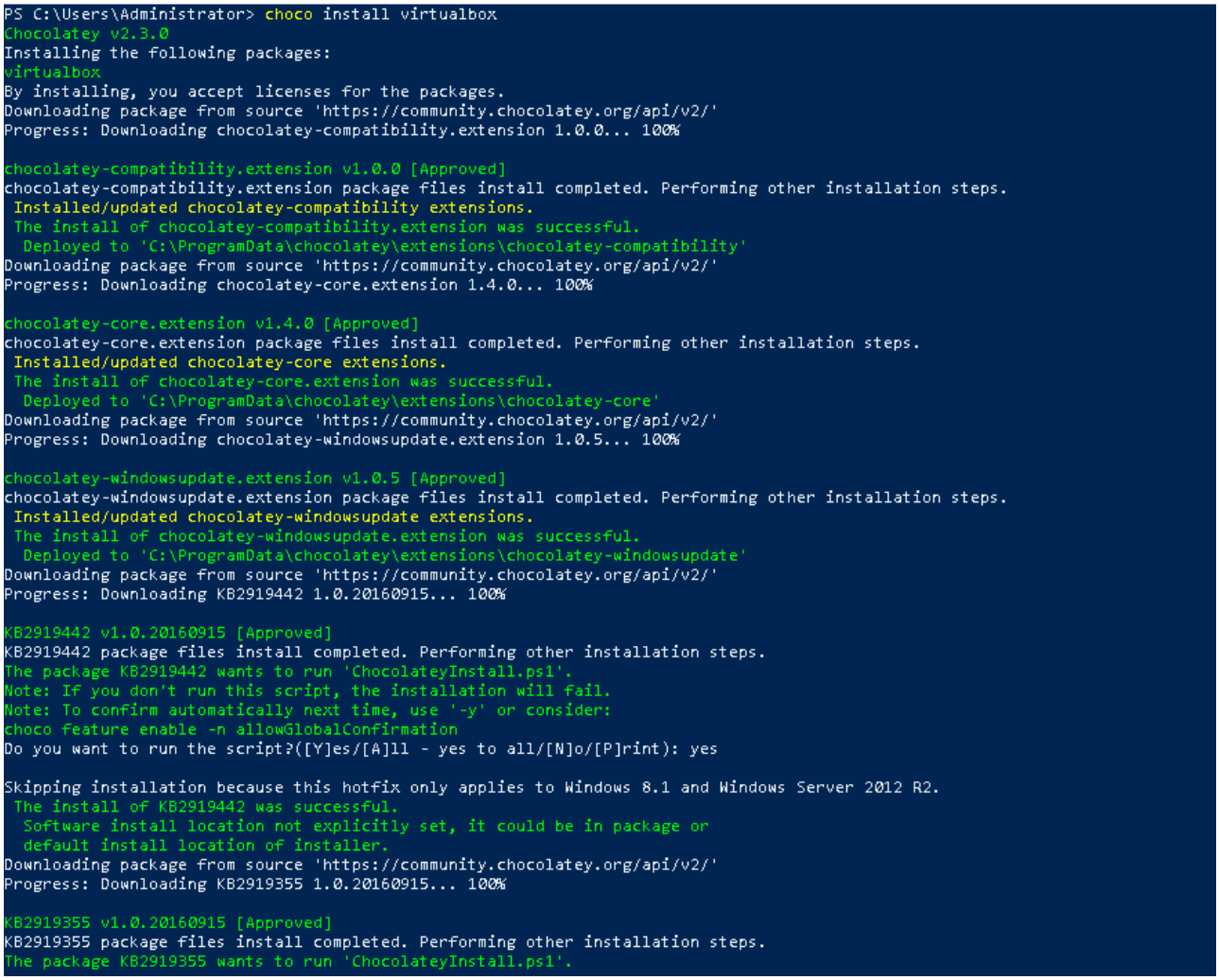
```
choco --version
```



2. Install Virtual Box

- VirtualBox is a hypervisor that lets you run virtual machines on your local workstation.
- To install Virtual Box, run the following command from the command line or from PowerShell:

```
choco install virtualbox
```



- Verify the installation by running the following command in cmd

```
virtualbox --version
```

2. Install Vagrant

- Vagrant manages hypervisors such as VirtualBox and makes it easy to distribute pre-packaged virtual machines, known as “boxes”.
- To install Vagrant, run the following command from the command line or from PowerShell:

```
choco install Vagrant
```

```
PS C:\Users\Administrator> choco install vagrant
Chocolatey v2.3.0
Installing the following packages:
vagrant
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading vagrant 2.4.2... 100%

vagrant v2.4.2 [Approved]
vagrant package files install completed. Performing other installation steps.
The package vagrant wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): yes

Downloading vagrant 64 bit
  from 'https://releases.hashicorp.com/vagrant/2.4.2/vagrant_2.4.2_windows_amd64.msi'
Progress: 100% - Completed download of C:\Users\Administrator\AppData\Local\Temp\chocolatey\vagrant\2.4.2\vagrant_2.4.2_windows_amd64.msi (250.51 MB).
Download of vagrant_2.4.2_windows_amd64.msi (250.51 MB) completed.
Hashes match.
Installing vagrant...
vagrant has been installed.
Updating installed plugins...
```

- Verify the installation by running the following command in cmd

```
vagrant --version
```

```
C:\Users\Administrator>vagrant --version
Vagrant 2.4.2

C:\Users\Administrator>
```

Step-2: Install Chef workstation

- To install Chef Workstation, run the following command from the command line or from PowerShell:

```
choco install chef-workstation
```

```
PS C:\Users\Administrator> choco install chef-workstation
Chocolatey v2.3.0
Installing the following packages:
chef-workstation
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading chef-workstation 24.8.1068... 100%

chef-workstation v24.8.1068 [Approved]
chef-workstation package files install completed. Performing other installation steps.
The package chef-workstation wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): yes
```

- This package includes Chef Infra Client, Chef InSpec, Test Kitchen, Cookstyle, and a variety of useful tools for the Chef ecosystem.

- Verify the installation by running the following command in cmd

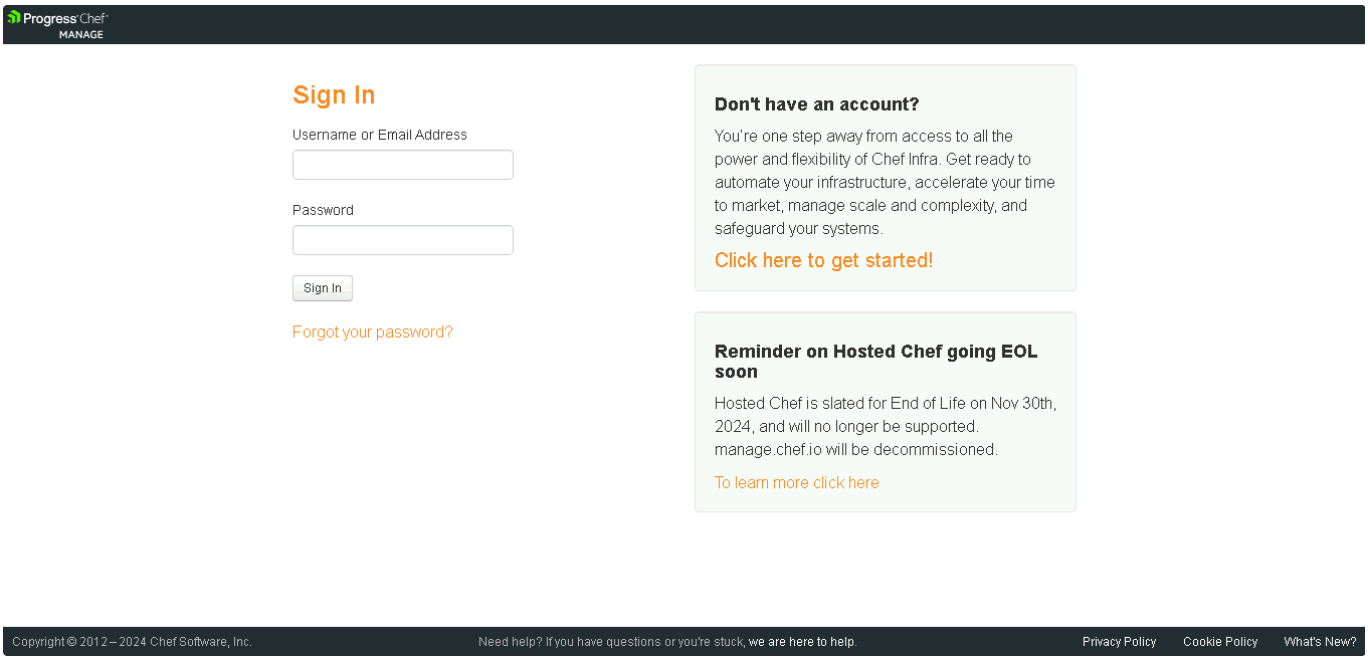
```
chef --version
```

```
C:\Users\Administrator>chef --version

Chef Workstation version: 24.8.1068
Chef Infra Client version: 18.5.0
Chef InSpec version: 5.22.55
Chef CLI version: 5.6.14
Chef Habitat version: 1.6.1041
Test Kitchen version: 3.6.0
Cookstyle version: 7.32.8

C:\Users\Administrator>
```

Step-3: Create an organization for server setup

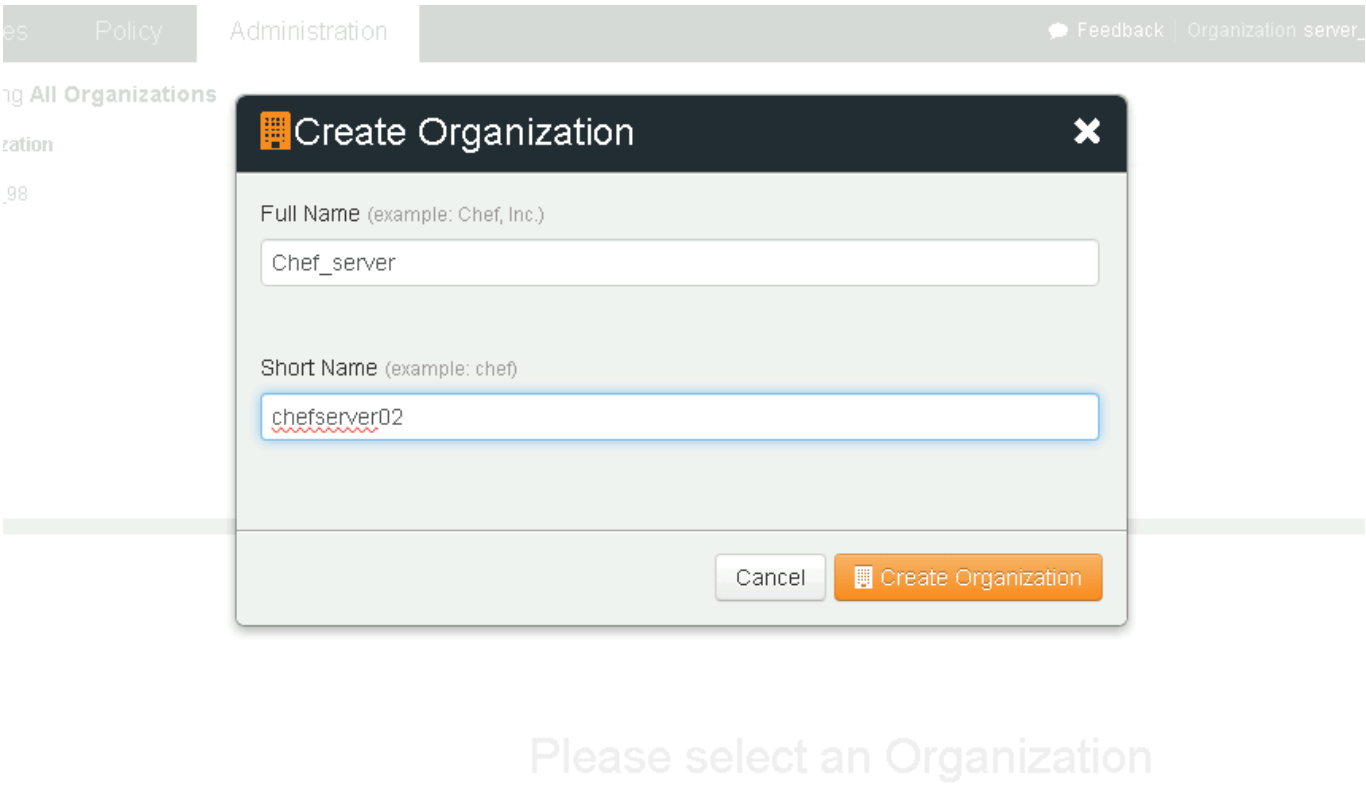
1. Sign Up for Manage Chef

- Go to Manage Chef in your web browser.
- Click **Sign Up** or **Get Started**.
- Enter your email address and follow the on-screen instructions to complete the sign-up process. You may be required to verify your email address.



2. Log in and Create an Organization

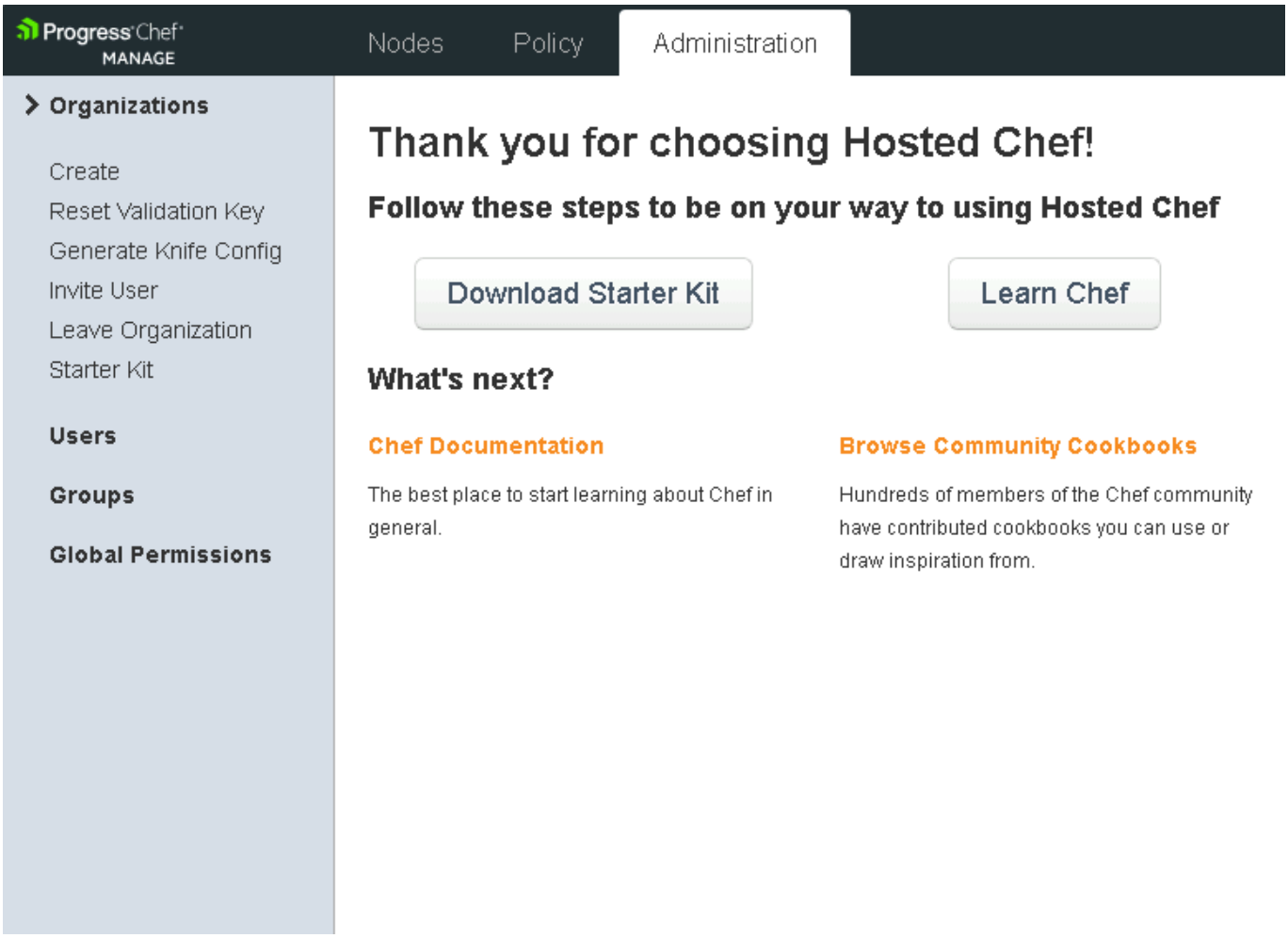
- Once signed up, log in to Manage Chef.
- After logging in, navigate to the **Organizations** section.
- Click on **Create Organization**.
- Provide the required details:
  - **Organization Name**: Choose a name for your organization.
  - **Short Name**: Enter a short, unique identifier for the organization.
- Complete any additional steps to confirm and create the organization.



3. Download the Starter Kit

- After creating the organization, navigate to the **Starter Kit** section (usually under your organization settings).
- Click **Download Starter Kit**. The kit includes essential configuration files for Chef, like `knife.rb`, which you'll need to manage nodes.
- Follow any on-screen prompts to confirm the download.
- Extract the files

The starter kit should now be downloaded to your system, and you can proceed with configuring Chef on your workstation using the kit files.



Step-4: Generate your Cookbook

- cd to chef-start\chef-repo\cookbooks and open command promt

1. Creating a Cookbook

- In order to keep our example as simple as possible let's create a chef-infra cookbook to automate the installation and management of the Git distributed version control tool.
- run the following command to generate a cookbook

```
chef generate cookbook git_cookbook
```

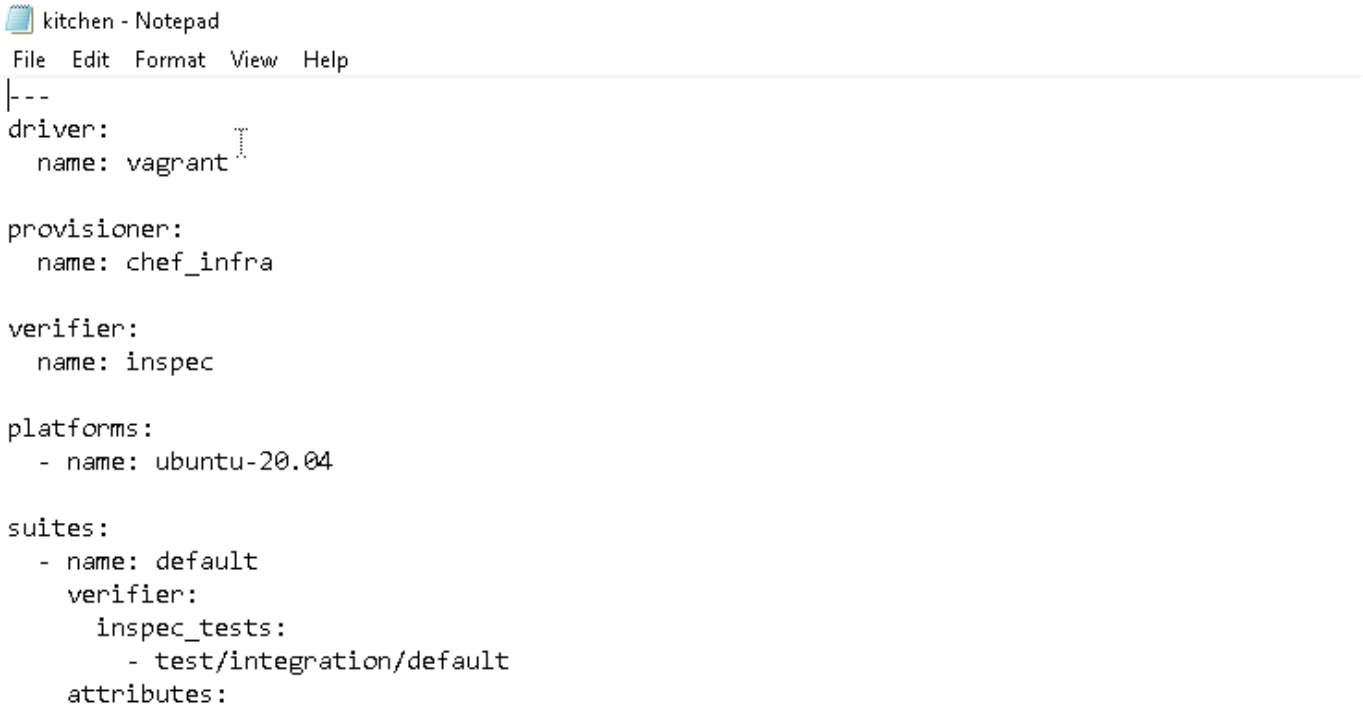
```
C:\Users\Administrator>chef generate cookbook git_cookbook
+----------------------------------------+
           Chef License Acceptance

Before you can continue, 3 product licenses
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

Licenses that need accepting:
  * Chef Workstation
  * Chef Infra Client
  * Chef InSpec

Do you accept the 3 product licenses (yes/no)?

> yes

Persisting 3 product licenses...
3 product licenses persisted.

+----------------------------------------+
Generating cookbook git_cookbook
- Ensuring correct cookbook content
- Committing cookbook files to git

Your cookbook is ready. Type `cd git_cookbook` to enter it.

There are several commands you can run to get started locally developing and testing your cookbook.

Why not start by writing an InSpec test? Tests for the default recipe are stored at:

test/integration/default/default_test.rb

If you'd prefer to dive right in, the default recipe can be found at:

recipes/default.rb


C:\Users\Administrator>
```

2. Configure kitchen.yam File for Node Configuration

- The `kitchen.yml` file is the main configuration file for **Test Kitchen**. It defines how to set up, run, and destroy instances for testing infrastructure code, primarily used for Chef cookbooks.
- cd to chef-start\chef-repo\cookbooks\git_cookbook
- Edit the `kitchen.yml` file

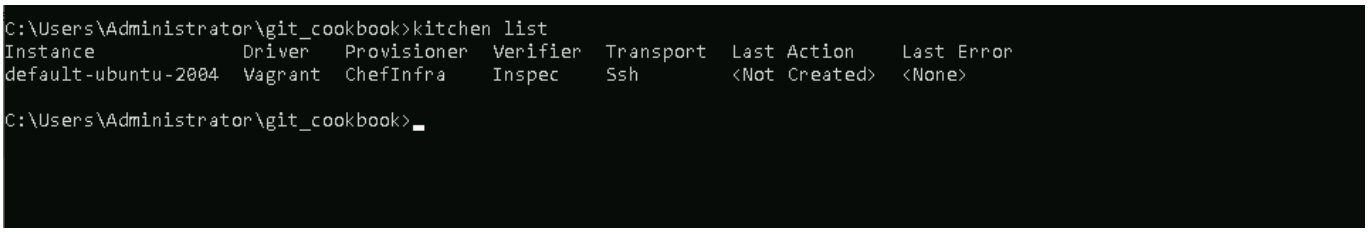
```
---
driver:
  name: vagrant

provisioner:
  name: chef_infra

verifier:
  name: inspec

platforms:
  - name: ubuntu-20.04

suites:
  - name: default
    verifier:
```

```
        inspec_tests:
          - test/integration/default
      attributes:
```



- Run the following command to verify the `kitchen list`

```
kitchen list
```



- Configure virtual machine boot timeout
    - cd to C:\Users\Administrator.vagrant.d\boxes\bento-VAGRANTSLASH-ubuntu-20.04\202407.23.0\amd64\virtualbox
    - Add config.vm.boot_timeout = 600 to the `Vagrantfile`

Step-5: Create your Node

A Test Kitchen **Instance** is a combination of a **Suite** and a **Platform** as laid out in your `kitchen.yml` file. Test Kitchen has auto-named our only instance by combining the **Suite** name (**"default"**) and the **Platform** name (**"ubuntu-20.04"**) into a form that is safe for DNS and hostname records, namely **"default-ubuntu-2004"**.

Let's spin this **Instance** up to see what happens. We're going to be explicit and ask kitchen to only create the **default-ubuntu-2004** instance:

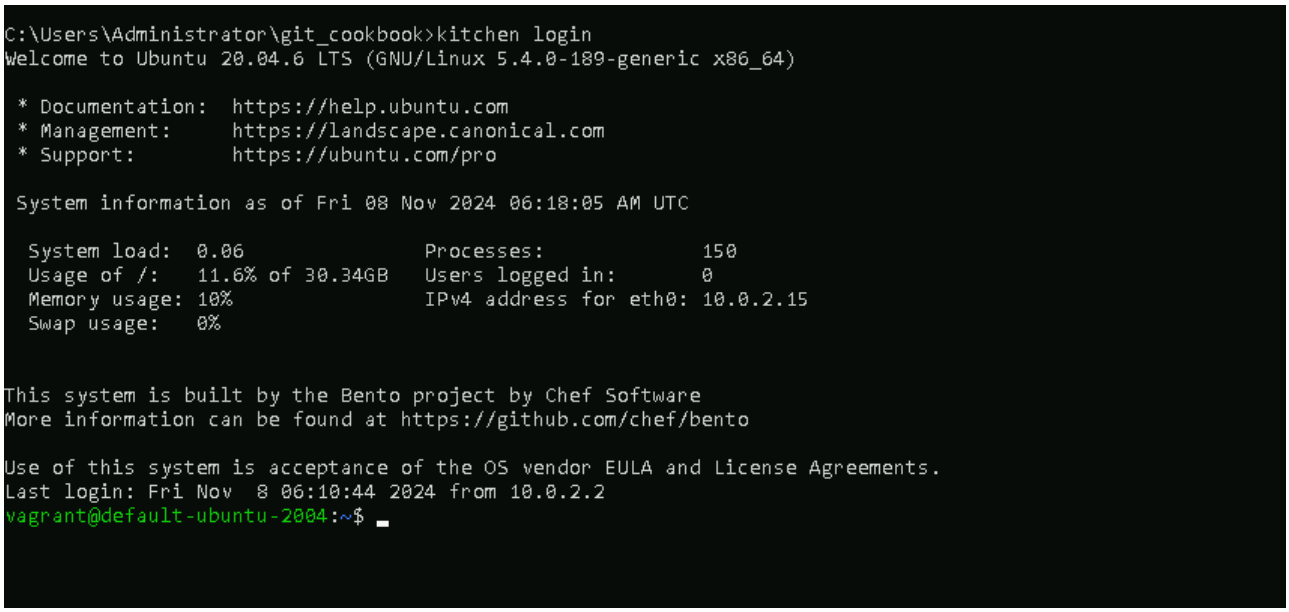
```
kitchen create default-ubuntu-2004
```

```
C:\Users\Administrator\git_cookbook>kitchen create default-ubuntu-2004
-----> Starting Test Kitchen (v3.6.0)
-----> Creating <default-ubuntu-2004>...
       Bringing machine 'default' up with 'virtualbox' provider...
       ==> default: Importing base box 'bento/ubuntu-20.04'...
==> default: Matching MAC address for NAT networking...
       ==> default: Checking if box 'bento/ubuntu-20.04' version '202407.23.0' is up to date...
       ==> default: Setting the name of the VM: kitchen-git_cookbook-default-ubuntu-2004-f65108d6-7cf9-48a7-b2d4-832077deb402
       ==> default: Clearing any previously set network interfaces...
       ==> default: Preparing network interfaces based on configuration...
           default: Adapter 1: nat
       ==> default: Forwarding ports...
           default: 22 (guest) => 2222 (host) (adapter 1)
       ==> default: Running 'pre-boot' VM customizations...
       ==> default: Booting VM...
       ==> default: Waiting for machine to boot. This may take a few minutes...
           default: SSH address: 127.0.0.1:2222
           default: SSH username: vagrant
           default: SSH auth method: private key
           default: Warning: Connection reset. Retrying...
           default: Warning: Connection aborted. Retrying...
           default:
           default: Vagrant insecure key detected. Vagrant will automatically replace
           default: this with a newly generated keypair for better security.
           default:
           default: Inserting generated public key within guest...
           default: Removing insecure key from the guest if it's present...
           default: Key inserted! Disconnecting and reconnecting using new SSH key...
       ==> default: Machine booted and ready!
       ==> default: Checking for guest additions in VM...
       ==> default: Setting hostname...
       ==> default: Mounting shared folders...
           default: C:/Users/Administrator/.kitchen/cache => /tmp/omnibus/cache
       ==> default: Machine not provisioned because `--no-provision` is specified.
       [SSH] Established
       Vagrant instance <default-ubuntu-2004> created.
       Finished creating <default-ubuntu-2004> (9m56.21s).
-----> Test Kitchen is finished. (10m1.39s)
```

1. Setup Node

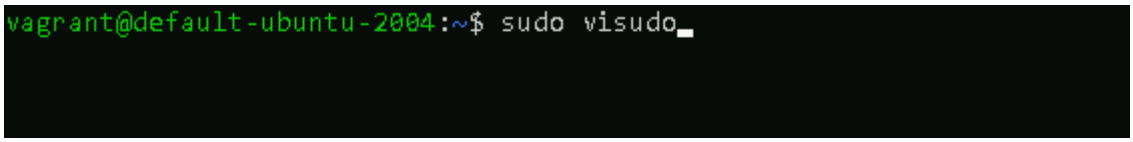
- Run the following command to login to your Node

```
kitchen login
```



- Run the following command to open the sudoers file safely

```
sudo visudo
```



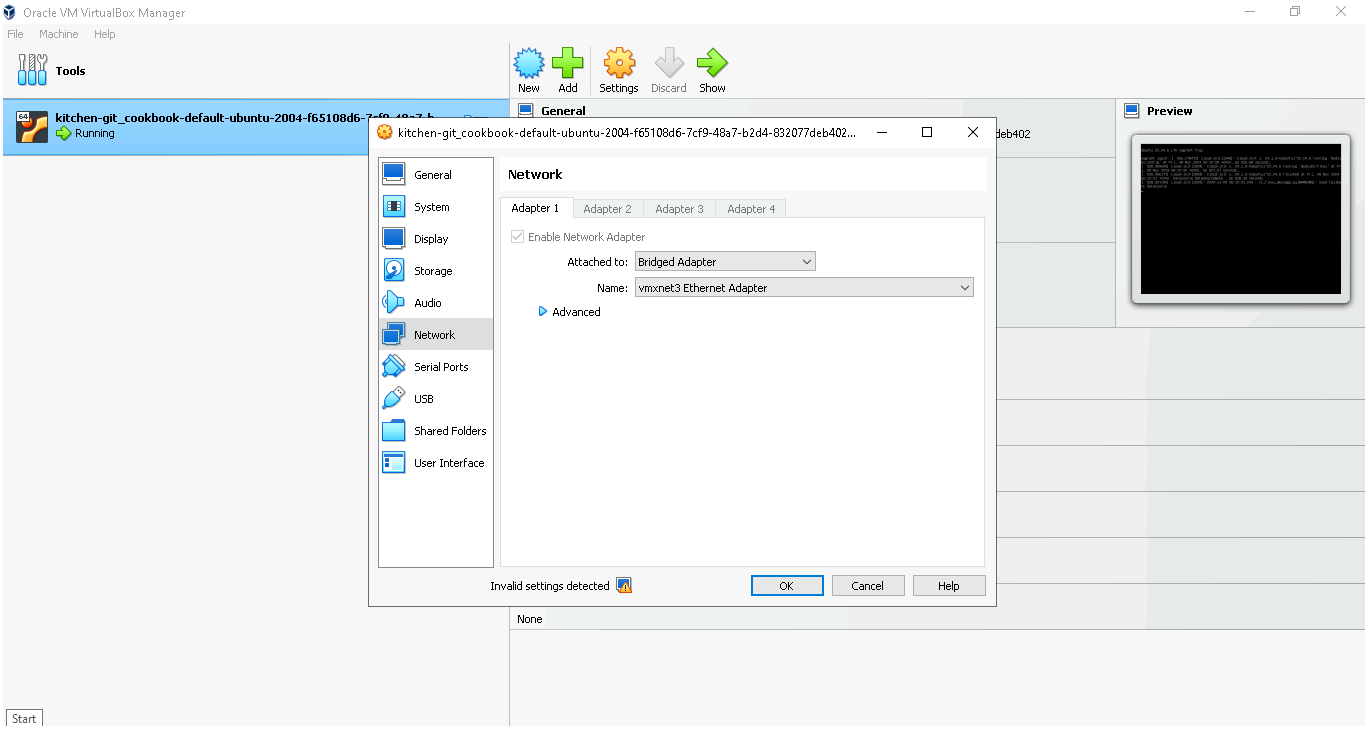
- Add the following at the end of sudoers file and `ctrl+x` to safely exit the file

```
vagrant  ALL=(ALL) NOPASSWD:ALL
```

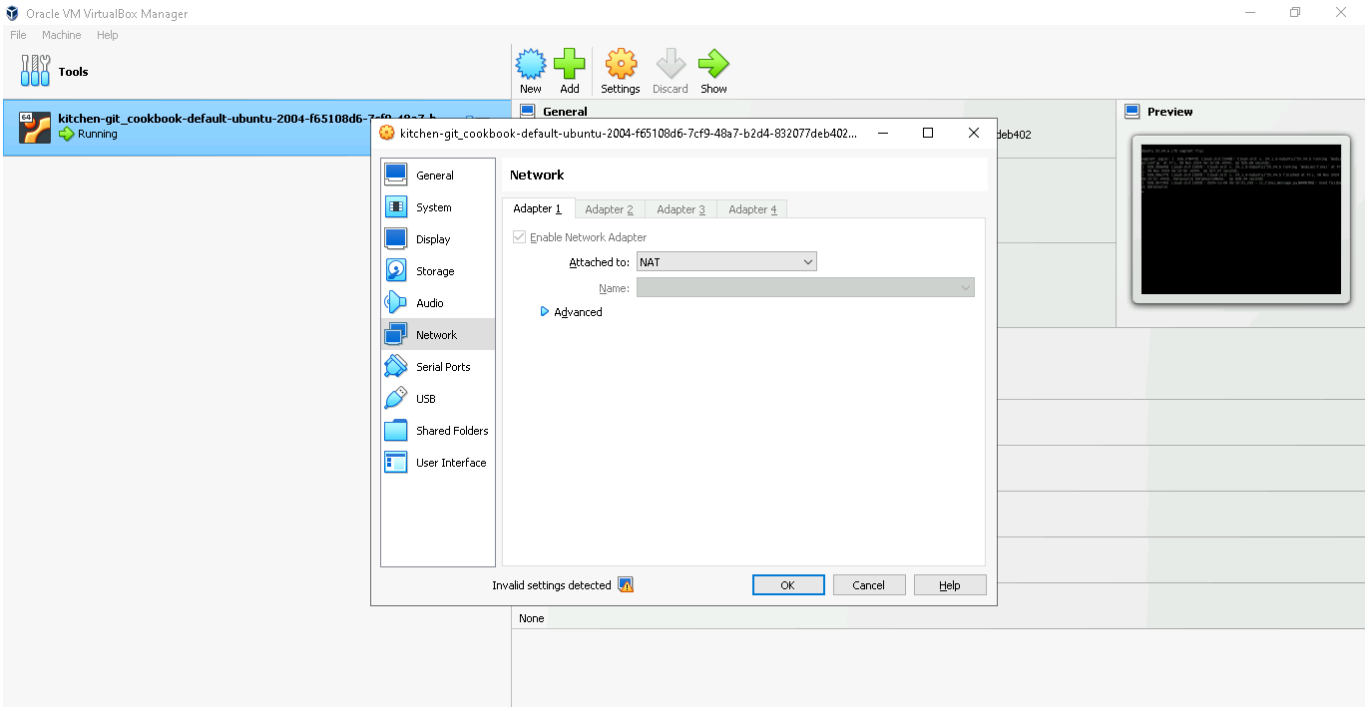
```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        exempt_group=sudo
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

vagrant  ALL=(ALL) NOPASSWD:ALL
```

Step-6: Change the Network Settings

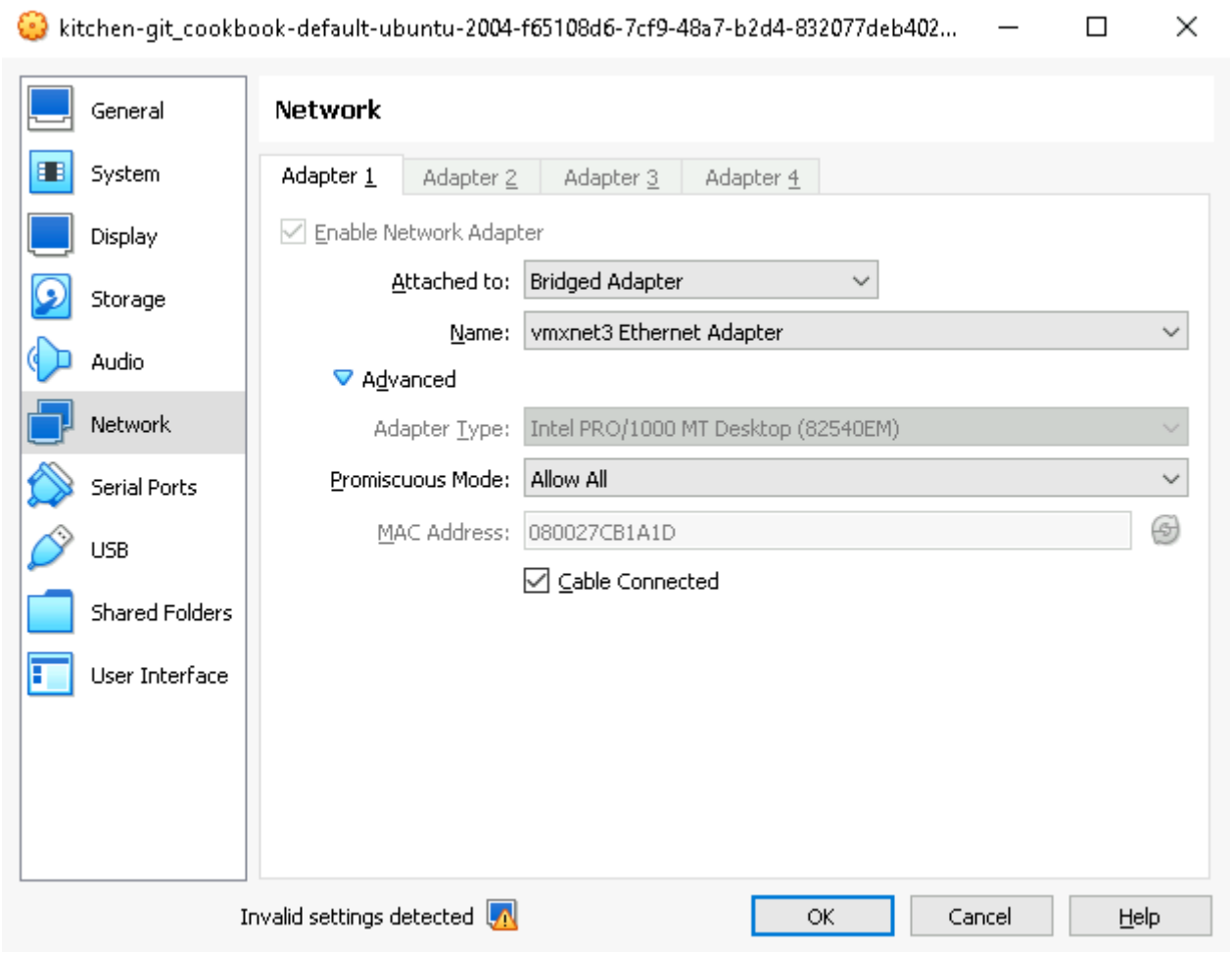
- Open your Virtual box and click on `Network`



- Change the Network settings from Attached to:`NAT` to Attached to:`Bridged Adapter`

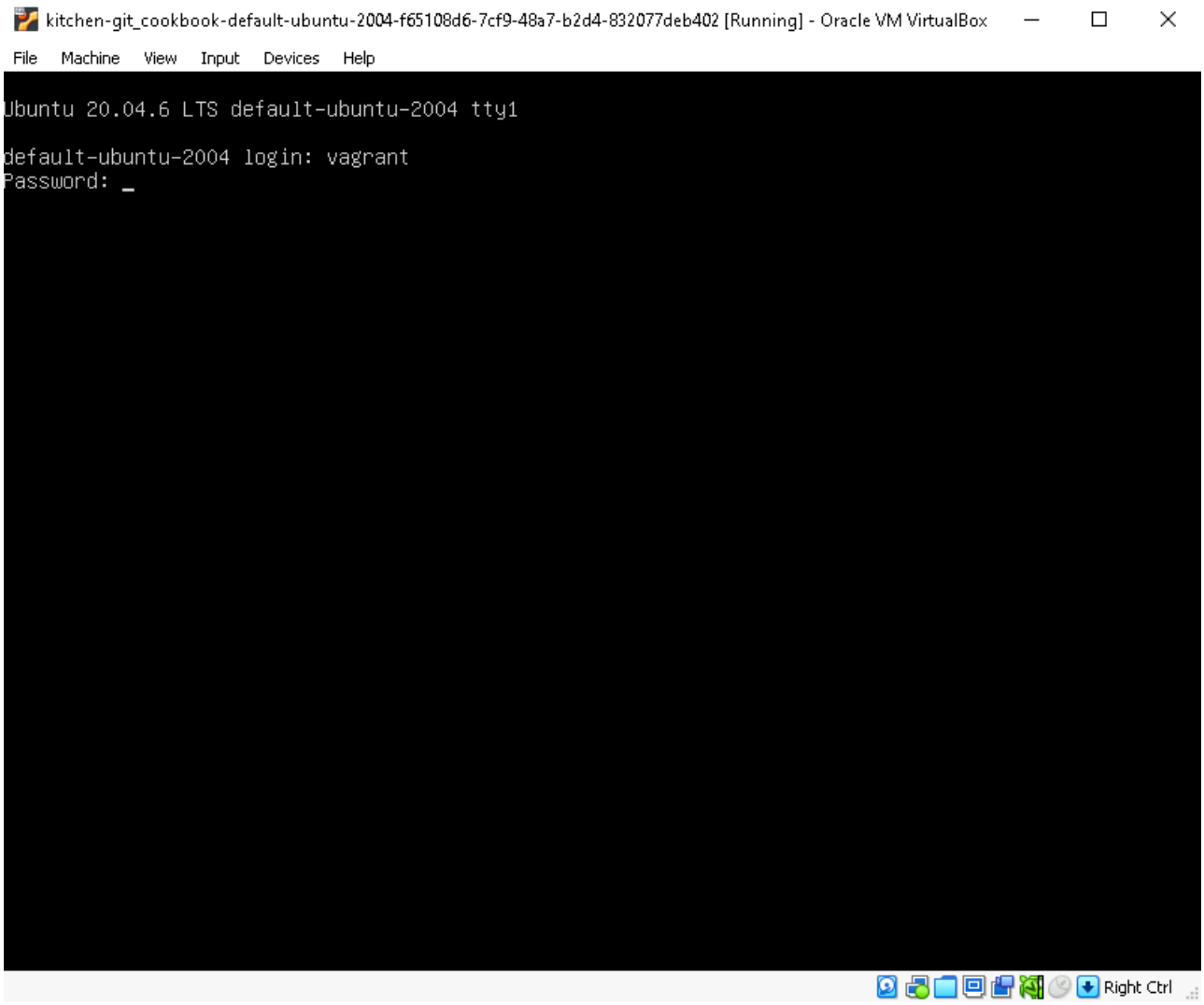


- Change the Promiscuous Mode: `Allow All` and click on `ok`



- Reboot your Node by stopping the system in virtual box and starting it again
- Once the system is Running, click on Show and login to your system with the following credetials
    - username- vagrant

- password- vagrant



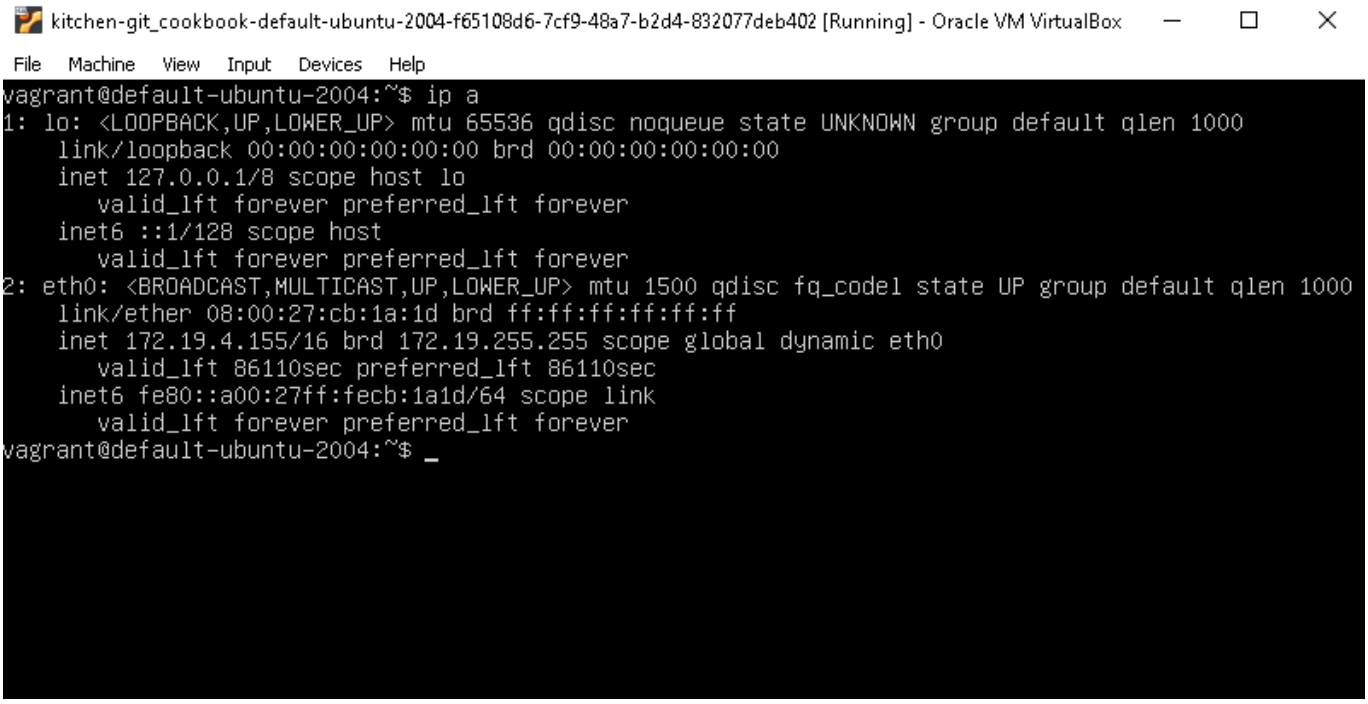
- Run the folllowing command to install net-tools

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

```
vagrant@default-ubuntu-2004:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 196 kB of archives.
After this operation, 864 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 net-tools amd64 1.60+git20180626.aebd88e-1ubuntu1 [196 kB]
Fetched 196 kB in 7s (27.7 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 41439 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20180626.aebd88e-1ubuntu1_amd64.deb ...
Unpacking net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Setting up net-tools (1.60+git20180626.aebd88e-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
vagrant@default-ubuntu-2004:~$
```

- Check the ipconfig of your Node and Note down the etho2 ip: inet

```
ip a
```

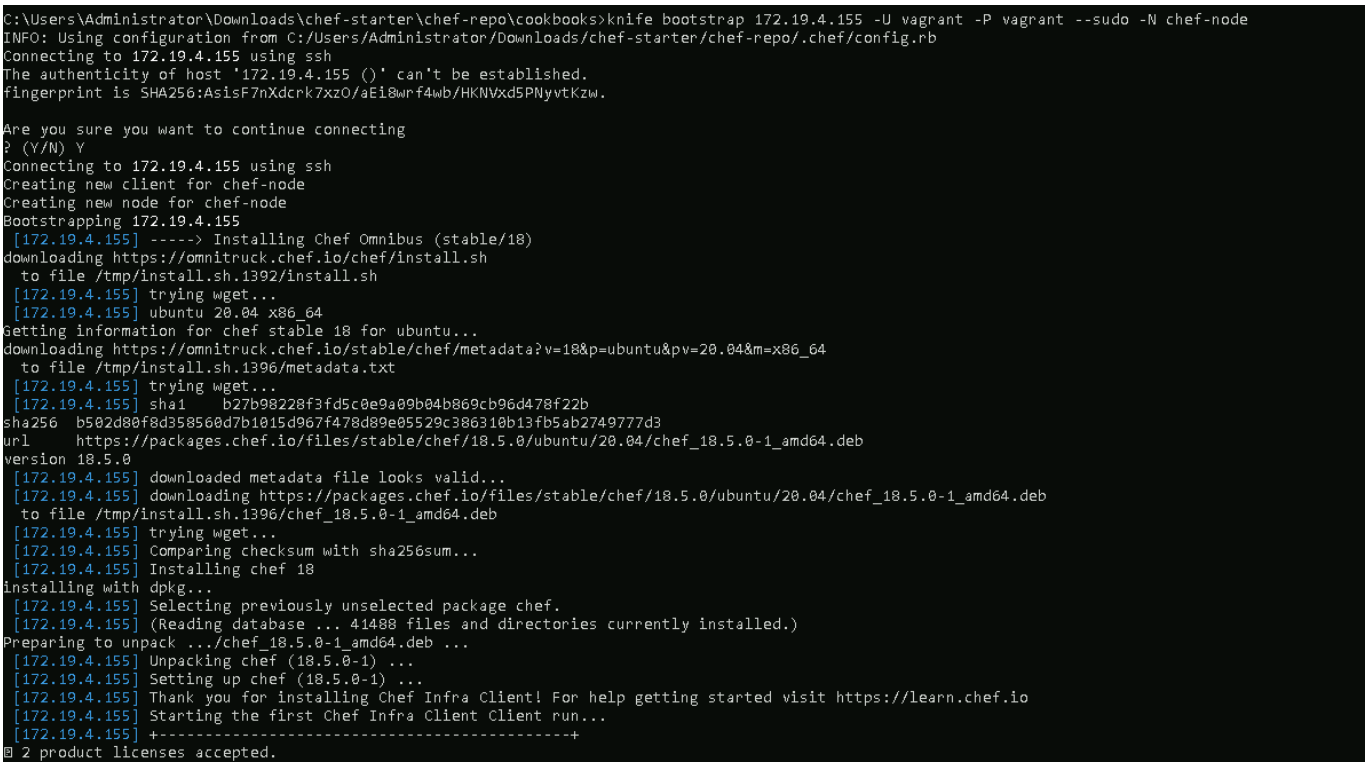


Step-7: Connect Server to your Node

Run the following command to connect server to your node

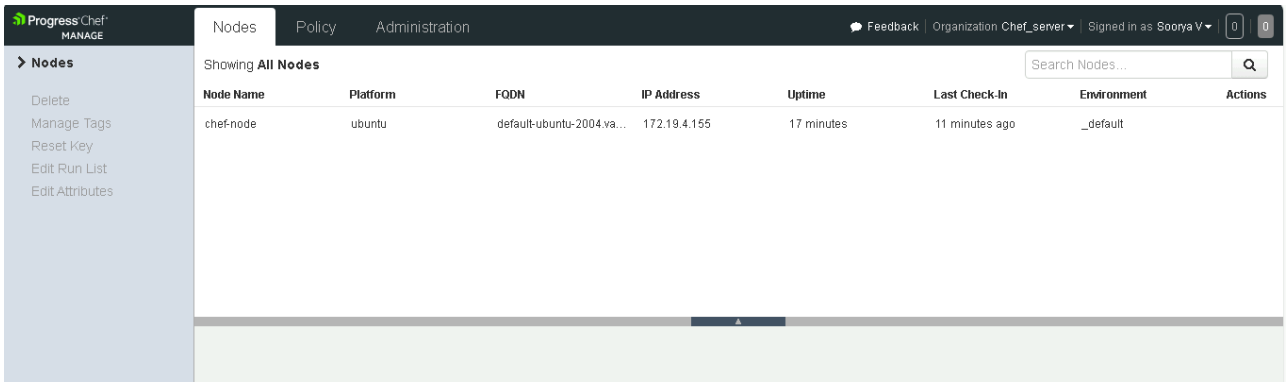
> Note: Replace the ip address in the command with your respective ip address

```
knife bootstrap 172.19.4.155 -U vagrant -P vagrant --sudo -N chef-node
```



- Verify your node

    1. Go to Manage Chef in your web browser.
    2. Click on your **Organization**.
    3. Click n **Nodes**.



References

- Official Chef Documentation: https://docs.chef.io/
- Setting Up Chef Server: https://docs.chef.io/server/
- Working with Chef Workstation: https://docs.chef.io/workstation/
- Chef Infra Client Overview: https://docs.chef.io/chef_client_overview/