**Installation of Minikube in a VirtualBox Appliance using Vagrant**

In this workshop we are going to install Minikube in a VirtualBox Appliance using Vagrant.

## Prerequisites

- Have VirtualBox installed (<https://www.virtualbox.org/wiki/Download_Old_Builds_5_2>), version 5.2.24

- Have Vagrant installed (<https://www.vagrantup.com/downloads.html>), version 2.2.3

*The Vagrant VirtualBox provider is compatible with VirtualBox versions 4.0.x, 4.1.x, 4.2.x, 4.3.x, 5.0.x, 5.1.x, 5.2.x, and 6.0.x. Other versions are unsupported and the provider will display an error message. Please note that beta and pre-release versions of VirtualBox are not supported and may not be well-behaved.*

# Let’s get started

Create a new directory and add the file **Vagrantfile** with the following contents:

Vagrant.configure("2") do |config|

config.vm.box = "ubuntu/xenial64"

config.vm.define "minikube" do |kube|

config.vm.provider "virtualbox" do |vb|

vb.memory = "4096"

vb.cpus = "4"

end

config.vm.network "forwarded\_port",

guest: 8001,

host: 8001,

auto\_correct: true

config.vm.network "forwarded\_port",

guest: 8443,

host: 8443,

auto\_correct: true

config.vm.network "forwarded\_port",

guest: 3306,

host: 3306,

auto\_correct: true

config.vm.network "forwarded\_port",

guest: 9000,

host: 9000,

auto\_correct: true

args = []

config.vm.provision "shell",

path: "scripts/kubectl.sh",

args: args

args = []

config.vm.provision "shell",

path: "scripts/docker.sh",

args: args

args = []

config.vm.provision "shell",

path: "scripts/minikube.sh",

args: args

args = []

config.vm.provision "shell",

path: "scripts/kubernetes.sh",

args: args

end

end

Add to the same directory a subdirectory called **scripts**.

Add to the **scripts** directory a file **kubectl.sh** with the following contents:

#!/bin/bash

echo "Begin installing kubectl"

apt-get update -qq && apt-get install -qqy apt-transport-https

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" > /etc/apt/sources.list.d/kubectl.list

apt-get update -qq

apt-get install -qqy kubectl

echo "End installing kubectl"

Add to the **scripts** directory a file **docker.sh** with the following contents:

#!/bin/bash

echo "Begin installing docker"

sudo apt-get remove docker docker-engine docker.io

sudo apt-get install apt-transport-https

sudo apt-get install ca-certificates

sudo apt-get install curl

sudo apt-get install software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

sudo apt-get update -qq

#sudo apt-get install -yqq docker-ce #By default, it installs the latest version of docker

sudo apt-get install -yqq docker-ce=18.06.1~ce~3-0~ubuntu

usermod -aG docker vagrant

sudo docker run hello-world

echo "End installing docker"

Add to the **scripts** directory a file **minikube.sh** with the following contents:

#!/bin/bash

echo "Begin downloading minikube"

curl -Lo minikube https://storage.googleapis.com/minikube/releases/v0.32.0/minikube-linux-amd64

chmod +x minikube

sudo cp minikube /usr/local/bin/

rm minikube

echo "End downloading minikube"

Add to the **scripts** directory a file **kubernetes.sh** with the following contents:

#!/bin/bash

echo "Begin setting up and starting kubernetes (k8s)"

sudo apt-get update -qq

sudo apt-get install ebtables

sudo apt-get install socat

sudo minikube start --vm-driver none

echo "Get Information About the Nodes in a Cluster"

sudo cp /etc/kubernetes/admin.conf $HOME

sudo chown $(id -u):$(id -g) $HOME/admin.conf

export KUBECONFIG=$HOME/admin.conf

kubectl get nodes

echo "Display cluster info"

kubectl cluster-info

#sudo minikube dashboard –url

kubectl proxy --address='0.0.0.0' --port=8001

echo "End setting up and starting kubernetes (k8s)"

## Create the VirtualBox Appliance via Vagrant

Go to the directory with the **Vagrantfile** and **scripts** subdirectory and open a Windows Command Prompt (cmd).

Install plugins:

vagrant plugin install vagrant-disksize

Installing the 'vagrant-disksize' plugin. This can take a few minutes...

Installed the plugin 'vagrant-disksize (0.1.3)'!

vagrant plugin install vagrant-docker-compose

Installing the 'vagrant-docker-compose' plugin. This can take a few minutes...

Installed the plugin 'vagrant-docker-compose (1.3.0)'!

vagrant plugin install vagrant-vbguest

Installing the 'vagrant-vbguest' plugin. This can take a few minutes...

Fetching: micromachine-2.0.0.gem (100%)

Fetching: vagrant-vbguest-0.17.2.gem (100%)

Installed the plugin 'vagrant-vbguest (0.17.2)'!

Update the box for the current Vagrant environment, if there are updates available:

vagrant box update

==> minikube: Checking for updates to 'ubuntu/xenial64'

minikube: Latest installed version: 20190123.0.0

minikube: Version constraints:

minikube: Provider: virtualbox

==> minikube: Box 'ubuntu/xenial64' (v20190123.0.0) is running the latest version.

Build the VirtualBox Appliance:

vagrant up

Bringing machine 'minikube' up with 'virtualbox' provider...

==> minikube: Importing base box 'ubuntu/xenial64'...

==> minikube: Matching MAC address for NAT networking...

==> minikube: Checking if box 'ubuntu/xenial64' is up to date...

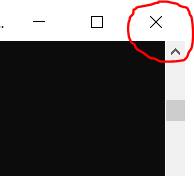
==> minikube: Setting the name of the VM: omgeving\_minikube\_1548405649622\_64179

…

When after some time you see the line below, the build is done:

minikube: Starting to serve on [::]:8001

You can now close the Windows Command Prompt, using the X on the top right:



Just to check the appliance is made correctly, open Oracle VM VirtualBox Manager.

The name of the appliance contains the name of the **directory** (with the Vagrantfile) and also **minikube**.



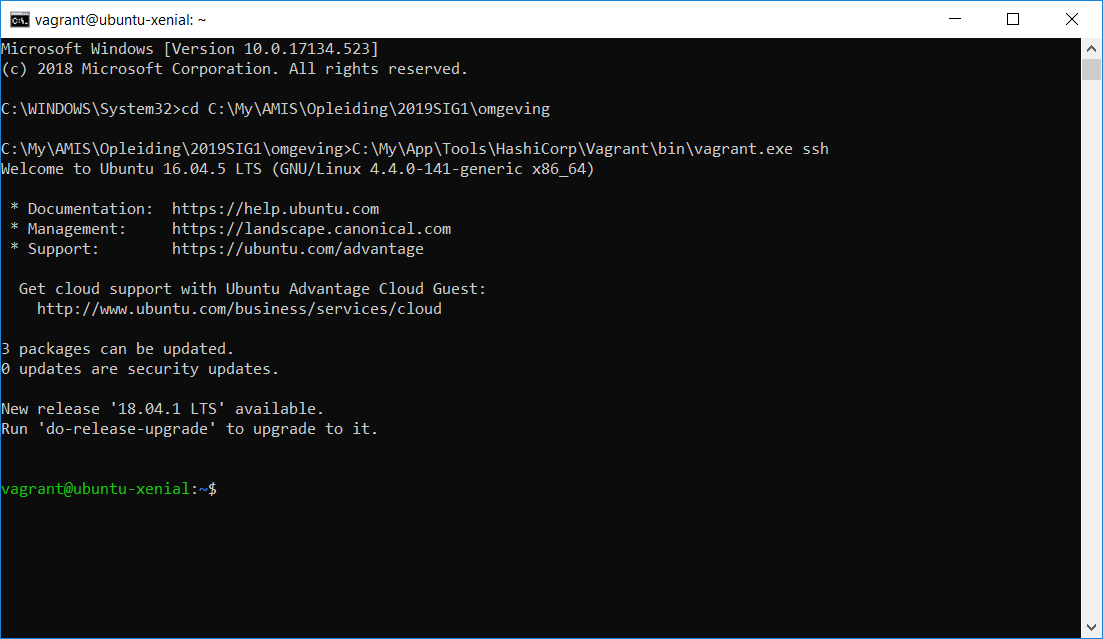
As you can see, the appliance is running.

## Open a Windows Command Prompt to access linux (within the VirtualBox Appliance)

Go to the directory with the **Vagrantfile** and **scripts** subdirectory and open a Windows Command Prompt (cmd) to access linux (within the VirtualBox Appliance) via ssh:

vagrant ssh

Use this Command Prompt for example for building a Docker Image.



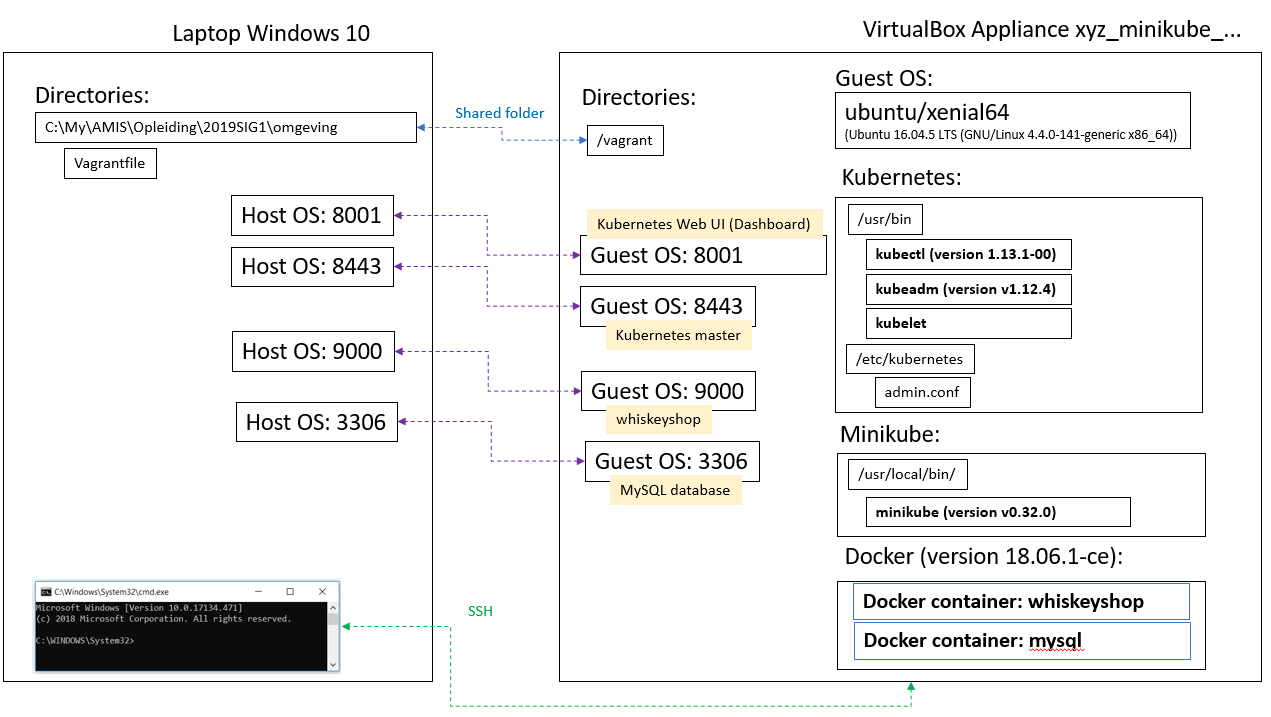
Navigate to the VirtualBox Appliance Shared Folder:



This is the **directory** with the Vagrantfile you created earlier.

## Overview of the VirtualBox Appliance

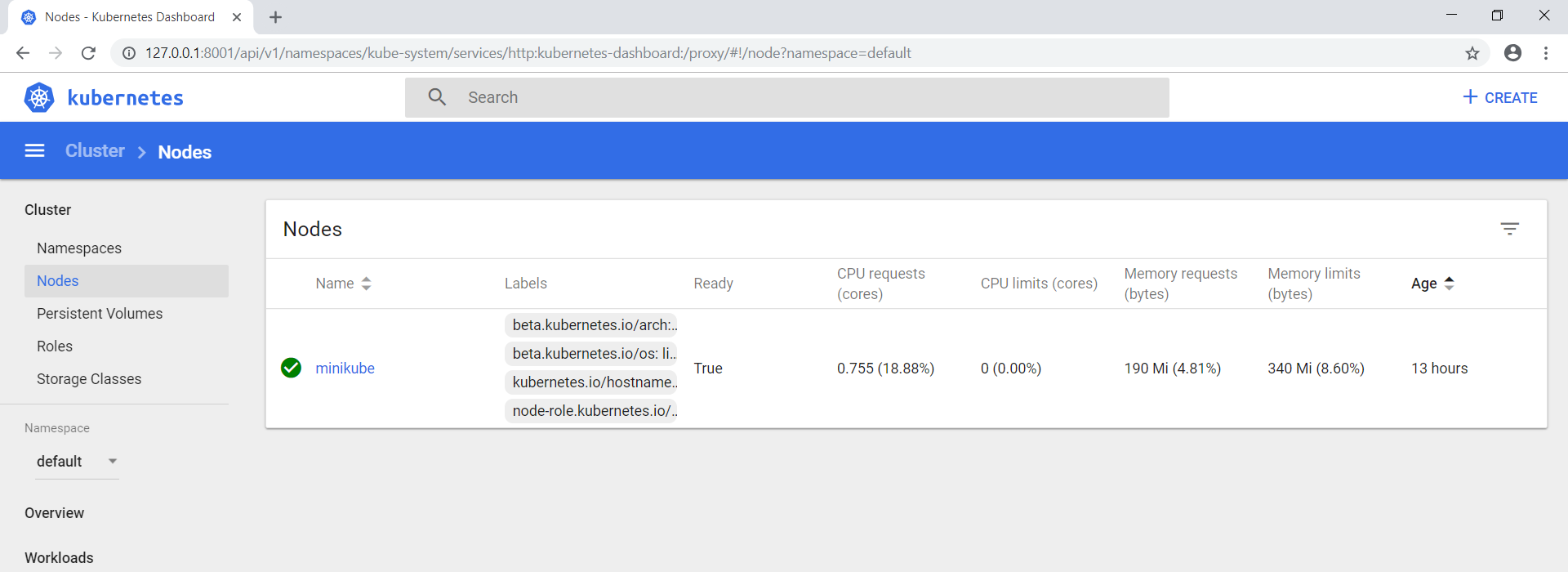
The VirtualBox Appliance contains the following:



## Kubernetes Web UI (Dashboard)

Open the Kubernetes Web UI (Dashboard) via:

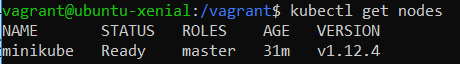
[http://127.0.0.1:8001/api/v1/namespaces/kube-system/services/http:kubernetes-dashboard:/proxy/#!/node?namespace=default](http://127.0.0.1:8001/api/v1/namespaces/kube-system/services/http:kubernetes-dashboard:/proxy/" \l "!/node?namespace=default)



There is a cluster with 1 Node called **minikube**.

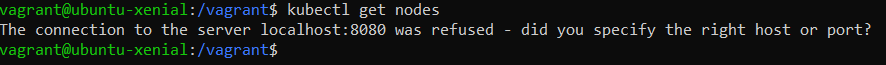
You can also check this from the linux Command Prompt:

kubectl get nodes



Now your minikube environment is ready for action.

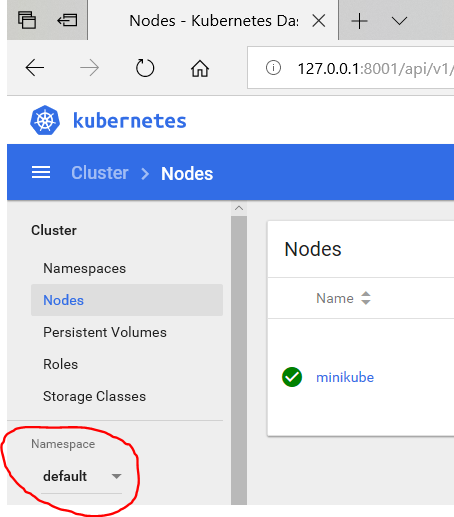
If you don’t get the answer above, you probably get the answer below.



For fixing this, see the paragraph: “The connection to the server localhost:8080 was refused - did you specify the right host or port?”.

Remark:

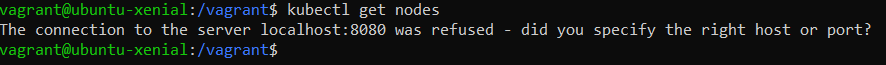
The Namespace that is used is: default



## The connection to the server localhost:8080 was refused - did you specify the right host or port?

Sometimes you get an error message when using **kubectl** from the linux Command Prompt. For example:

kubectl get nodes



Then you simply have to give the following commands:

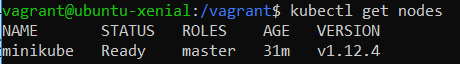
sudo cp /etc/kubernetes/admin.conf $HOME

sudo chown $(id -u):$(id -g) $HOME/admin.conf

export KUBECONFIG=$HOME/admin.conf

And try again:

kubectl get nodes



Remark:

After restarting the created VirtualBox Appliance, this problem occurs.

## Closing the VirtualBox Appliance

If you want to close the running VirtualBox Appliance, open Oracle VM VirtualBox Manager, select the appliance and from the menu choose: Machine | Close | Save State.

## Cleaning up the VirtualBox Appliance

If you want to clean up the created VirtualBox Appliance, you can this by opening a Windows Command Prompt (cmd) :

vagrant destroy

When you get the following question, answer: y

minikube: Are you sure you want to destroy the 'minikube' VM? [y/N]

==> minikube: Discarding saved state of VM...

==> minikube: Destroying VM and associated drives...