Multi Node Kubernetes Cluster with Vagrant, VirtualBox and Kubeadm

Raj Rajaratnam Follow
Sep 6, 2018 · 4 min read

TL;DR, Go grab my Vagrantfile at https://github.com/ecomm-integration-ballerina/kubernetes-cluster and run `vagrant up` to bring a 3 node Kubernetes cluster up and running in VirtualBox.

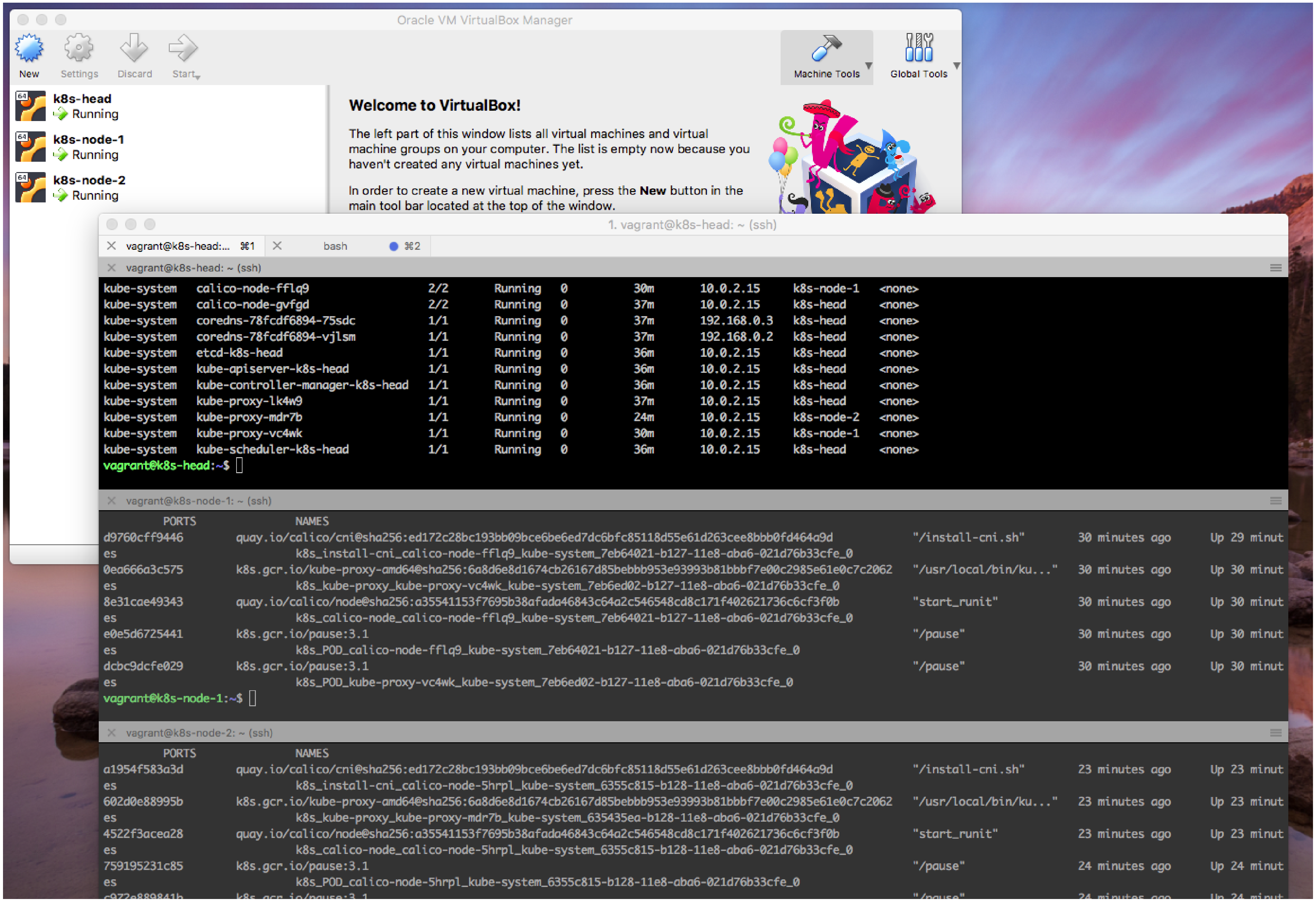
. . .

I have been trying to setup a multi node Kubernetes cluster to try out Ballerina integrations at scale in containers.

There are many third party tools to setup a Kubernetes cluster. I've looked at two options, Minikube and Kubeadm. Minikube doesn't support multi node setup, so I picked Kubeadm. Kubeadm is still in beta, do not use it in production.

I followed Kubeadm guide and setup a three node cluster (single master and two nodes) manually first and then automated in Vagrant because I can recreate the cluster easily later with no effort.

So here is how I did it. Try it, create issues in my repo and send pull requests.



Welcome to VirtualBox!

The left part of this window lists all virtual machines and virtual machine groups on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

vagrant@k8s-head: ~ (ssh)

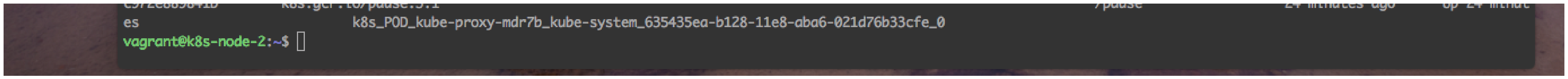
```
kube-system   calico-node-fflq9                   2/2     Running   0          30m     10.0.2.15     k8s-node-1   <none>
kube-system   calico-node-gvfgd                   2/2     Running   0          37m     10.0.2.15     k8s-head     <none>
kube-system   coredns-78fcdf6894-75sdc            1/1     Running   0          37m     192.168.0.3   k8s-head     <none>
kube-system   coredns-78fcdf6894-vjlsm            1/1     Running   0          37m     192.168.0.2   k8s-head     <none>
kube-system   etcd-k8s-head                       1/1     Running   0          36m     10.0.2.15     k8s-head     <none>
kube-system   kube-apiserver-k8s-head             1/1     Running   0          36m     10.0.2.15     k8s-head     <none>
kube-system   kube-controller-manager-k8s-head    1/1     Running   0          36m     10.0.2.15     k8s-head     <none>
kube-system   kube-proxy-lk4w9                    1/1     Running   0          37m     10.0.2.15     k8s-head     <none>
kube-system   kube-proxy-mdr7b                    1/1     Running   0          24m     10.0.2.15     k8s-node-2   <none>
kube-system   kube-proxy-vc4wk                    1/1     Running   0          30m     10.0.2.15     k8s-node-1   <none>
kube-system   kube-scheduler-k8s-head             1/1     Running   0          36m     10.0.2.15     k8s-head     <none>
vagrant@k8s-head:~$
```

vagrant@k8s-node-1: ~ (ssh)

```
         PORTS                  NAMES
d9760cff9446       quay.io/calico/cni@sha256:ed172c28bc193bb09bce6be6ed7dc6bfc85118d55e61d263cee8bbb0fd464a9d      "/install-cni.sh"       30 minutes ago       Up 29 minut
es                      k8s_install-cni_calico-node-fflq9_kube-system_7eb64021-b127-11e8-aba6-021d76b33cfe_0
0ea666a3c575       k8s.gcr.io/kube-proxy-amd64@sha256:6a8d6e8d1674cb26167d85bebbb953e93993b81bbbf7e00c2985e61e0c7c2062   "/usr/local/bin/ku..."   30 minutes ago       Up 30 minut
es                      k8s_kube-proxy_kube-proxy-vc4wk_kube-system_7eb6ed02-b127-11e8-aba6-021d76b33cfe_0
8e31cae49343       quay.io/calico/node@sha256:a35541153f7695b38afada46843c64a2c546548cd8c171f402621736c6cf3f0b      "start_runit"           30 minutes ago       Up 30 minut
es                      k8s_calico-node_calico-node-fflq9_kube-system_7eb64021-b127-11e8-aba6-021d76b33cfe_0
e0e5d6725441       k8s.gcr.io/pause:3.1                                                                             "/pause"                30 minutes ago       Up 30 minut
es                      k8s_POD_calico-node-fflq9_kube-system_7eb64021-b127-11e8-aba6-021d76b33cfe_0
dcbc9dcfe029       k8s.gcr.io/pause:3.1                                                                             "/pause"                30 minutes ago       Up 30 minut
es                      k8s_POD_kube-proxy-vc4wk_kube-system_7eb6ed02-b127-11e8-aba6-021d76b33cfe_0
vagrant@k8s-node-1:~$
```

vagrant@k8s-node-2: ~ (ssh)

```
         PORTS                  NAMES
a1954f583a3d       quay.io/calico/cni@sha256:ed172c28bc193bb09bce6be6ed7dc6bfc85118d55e61d263cee8bbb0fd464a9d      "/install-cni.sh"       23 minutes ago       Up 23 minut
es                      k8s_install-cni_calico-node-5hrpl_kube-system_6355c815-b128-11e8-aba6-021d76b33cfe_0
602d0e88995b       k8s.gcr.io/kube-proxy-amd64@sha256:6a8d6e8d1674cb26167d85bebbb953e93993b81bbbf7e00c2985e61e0c7c2062   "/usr/local/bin/ku..."   23 minutes ago       Up 23 minut
es                      k8s_kube-proxy_kube-proxy-mdr7b_kube-system_635435ea-b128-11e8-aba6-021d76b33cfe_0
4522f3acea28       quay.io/calico/node@sha256:a35541153f7695b38afada46843c64a2c546548cd8c171f402621736c6cf3f0b      "start_runit"           23 minutes ago       Up 23 minut
es                      k8s_calico-node_calico-node-5hrpl_kube-system_6355c815-b128-11e8-aba6-021d76b33cfe_0
759195231c85       k8s.gcr.io/pause:3.1                                                                             "/pause"                24 minutes ago       Up 24 minut
es                      k8s_POD_calico-node-5hrpl_kube-system_6355c815-b128-11e8-aba6-021d76b33cfe_0
c972e889841b       k8s.gcr.io/pause:3.1                                                                             "/pause"                24 minutes ago       Up 24 minut
```



```
es                  k8s_POD_kube-proxy-mdr7b_kube-system_635435ea-b128-11e8-aba6-021d76b33cfe_0
vagrant@k8s-node-2:~$ 
```

> *Note: Step-1 to Step-8 should be done in all the Kubernetes nodes (master + nodes). Step-9 to Step-12 should be done in Kubernetes master. Step-13 should be done in Kubernetes nodes.*

Step-1: Pick a base box for Kubernetes nodes

```
config.vm.box = "ubuntu/xenial64"
config.vm.box_version = "20180831.0.0"
```

Step-2: Setup a network

```
config.vm.network :private_network, ip: "192.168.205.10"
```

Step-3: Setup memory and CPUs

```
v.customize ["modifyvm", :id, "--memory", "2048"]
v.customize ["modifyvm", :id, "--cpus", "2"]
```

Step-4: Install Docker

I’m not going to use vagrant docker provisioning because it always installs the latest version, but Kubeadm requires 17.03 or older. So I’m going to use shell provisioning to install docker.

```
apt-get update
apt-get install -y apt-transport-https ca-certificates curl
software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
apt-key add -
add-apt-repository "deb
https://download.docker.com/linux/$(. /etc/os-release; echo
"$ID") $(lsb_release -cs) stable"
apt-get update && apt-get install -y docker-ce=$(apt-cache
madison docker-ce | grep 17.03 | head -1 | awk '{print $3}')
```

Step-5: Allow regular user to run docker commands

I’m adding the regular user (vagrant) to the docker user group so that I can run docker commands as vagrant user without sudo privileges.

```
usermod -aG docker vagrant
```

Step-6: Install Kubeadm, Kubelet and Kubctl

I’m using shell provisioning to install Kubeadm, Kubelet and Kubectl from the repositories.

```
    apt-get install -y apt-transport-https curl
    curl -s https://packages.cloud.google.com/apt/doc/apt-
```

```
key.gpg | apt-key add -
    cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
    deb http://apt.kubernetes.io/ kubernetes-xenial main
EOF
    apt-get update
    apt-get install -y kubelet kubeadm kubectl
    apt-mark hold kubelet kubeadm kubectl
```

Step-7: Kubelet requires swap to be disabled

Running following bash command will disable the swap in the current login session.

```
swapoff -a
```

Step-8: Disable swap permanently

Following change will keep swap off after reboot too

```
sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

Step-9: Initialize Kubernetes master

Following Kubeadm command will install and configure a Kubernetes master.

```
# ip of this box
IP_ADDR=`ifconfig enp0s8 | grep Mask | awk '{print $2}'| cut
-f2 -d:`
```

```
# install k8s master
HOST_NAME=$(hostname -s)
kubeadm init --apiserver-advertise-address=$IP_ADDR --
apiserver-cert-extra-sans=$IP_ADDR  --node-name $HOST_NAME -
-pod-network-cidr=192.168.0.0/16
```

Step-10: Allow regular user to run Kubectl commands

I'm going to copy the Kubectl credentials to vagrant user's home directory to be able to run Kubectl commands without sudo privileges.

```
#copying credentials to regular user - vagrant
sudo --user=vagrant mkdir -p /home/vagrant/.kube
cp -i /etc/kubernetes/admin.conf /home/vagrant/.kube/config
chown $(id -u vagrant):$(id -g vagrant)
/home/vagrant/.kube/config
```

Step-11: Install a Pod Network

We need to install a Pod Network to enable pod-to-pod communication in our Kubernetes cluster. There are many third party Pod Network projects such as Calico, Flannel and Weave. I've tried to use Weave, but looks like there are some bugs that prevent a Weave pod to be scheduled on Kubenetes nodes. Then I've tried Calico and it works fine so far.

```
# install Calico pod network addon
export KUBECONFIG=/etc/kubernetes/admin.conf
kubectl apply -f
https://docs.projectcalico.org/v3.1/getting-
started/kubernetes/installation/hosted/rbac-kdd.yaml
kubectl apply -f
```

```
https://docs.projectcalico.org/v3.1/getting-
started/kubernetes/installation/hosted/kubernetes-
datastore/calico-networking/1.7/calico.yaml
```

Step-12: Save the Kubeadm join commands

Our Kubernetes master should be up and running by now. We can join Kubernetes nodes to master using Kubeadm join command. You can run `kubeadm token create — print-join-command` in Kubernetes master to get the join command that should be executed in Kubernetes nodes.

I'm going to store the output of this command in a file in Kubernetes master. Later I'm going to download this file from inside Kubernetes nodes, execute and join the cluster.

```
kubeadm token create --print-join-command >>
/etc/kubeadm_join_cmd.sh
chmod +x /etc/kubeadm_join_cmd.sh
```

Step-13: Execute Kubeadm join command in Kubernetes nodes

What I'm doing here is to scp the above `kubeadm_join_cmd.sh` from master to nodes and execute it. I'm using `sshpass` to run prompt-less scp command in bash.

```
apt-get install -y sshpass
sshpass -p "vagrant" scp -o StrictHostKeyChecking=no
vagrant@192.168.205.10:/etc/kubeadm_join_cmd.sh .
sh ./kubeadm_join_cmd.sh
```

That’s all, your cluster should be up and running now.

Putting it all together

Here is my full Vagrantfile gist. But be sure check my repo to get the up-to-date Vagrant file. Run `vagrant up` to bring up a 3 node Kubernetes cluster.

```ruby
# -*- mode: ruby -*-
# vi: set ft=ruby :

servers = [
    {
        :name => "k8s-head",
        :type => "master",
        :box => "ubuntu/xenial64",
        :box_version => "20180831.0.0",
        :eth1 => "192.168.205.10",
        :mem => "2048",
        :cpu => "2"
    },
    {
        :name => "k8s-node-1",
        :type => "node",
        :box => "ubuntu/xenial64",
        :box_version => "20180831.0.0",
        :eth1 => "192.168.205.11",
        :mem => "2048",
        :cpu => "2"
    },
    {
        :name => "k8s-node-2",
        :type => "node",
        :box => "ubuntu/xenial64",
        :box_version => "20180831.0.0",
        :eth1 => "192.168.205.12",
        :mem => "2048",
        :cpu => "2"
    }
]

# This script to install k8s using kubeadm will get e
```

```
$configureBox = <<-SCRIPT

    # install docker v17.03
    # reason for not using docker provision is that i
    apt-get update
    apt-get install -y apt-transport-https ca-certifi
    curl -fsSL https://download.docker.com/linux/ubun
    add-apt-repository "deb https://download.docker.c
    apt-get update && apt-get install -y docker-ce=$(

    # run docker commands as vagrant user (sudo not r
    usermod -aG docker vagrant

    # install kubeadm
    apt-get install -y apt-transport-https curl
    curl -s https://packages.cloud.google.com/apt/doc
    cat <<EOF >/etc/apt/sources.list.d/kubernetes.lis
    deb http://apt.kubernetes.io/ kubernetes-xenial m
EOF
    apt-get update
    apt-get install -y kubelet kubeadm kubectl
    apt-mark hold kubelet kubeadm kubectl

    # kubelet requires swap off
    swapoff -a

    # keep swap off after reboot
    sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fsta

    # ip of this box
    IP_ADDR=`ifconfig enp0s8 | grep Mask | awk '{prin
    # set node-ip
    sudo sed -i "/^[^#]*KUBELET_EXTRA_ARGS=/c\KUBELET
    sudo systemctl restart kubelet
```

```
SCRIPT

$configureMaster = <<-SCRIPT
    echo "This is master"
    # ip of this box
    IP_ADDR=`ifconfig enp0s8 | grep Mask | awk '{prin

    # install k8s master
    HOST_NAME=$(hostname -s)
    kubeadm init --apiserver-advertise-address=$IP_AD

    #copying credentials to regular user - vagrant
    sudo --user=vagrant mkdir -p /home/vagrant/.kube
    cp -i /etc/kubernetes/admin.conf /home/vagrant/.k
    chown $(id -u vagrant):$(id -g vagrant) /home/vag

    # install Calico pod network addon
    export KUBECONFIG=/etc/kubernetes/admin.conf
    kubectl apply -f https://docs.projectcalico.org/v
    kubectl apply -f https://docs.projectcalico.org/v
```

🙏