

Vagrant, k3s and VirtualBox

Adrian Kosmaczewski

2021-09-23

Last weekend I decided to learn Vagrant to build a simple k3s Kubernetes cluster on top of a set of VirtualBox virtual machines.

The result of those explorations is now available on my GitLab for you to use and enjoy.

The `Vagrantfile` simply launches 4 VMs based on Alpine 3.14; one for the k3s “server” (the control plane) and three agents (aka worker nodes). The `kubeconfig` file is copied to a shared folder, so that one can easily `kubectl` or even better `k9s` on the new cluster, to install applications or do whatever you want.

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

```
server_ip = "192.168.33.10"
```

```
agents = { "agent1" => "192.168.33.11",  
           "agent2" => "192.168.33.12",  
           "agent3" => "192.168.33.13" }
```

```
# Extra parameters in INSTALL_K3S_EXEC variable because of  
# K3s picking up the wrong interface when starting server and agent  
# https://github.com/alexellis/k3sup/issues/306
```

```
server_script = <<-SHELL
```

```
  sudo -i
```

```
  apk add curl
```

```
  export INSTALL_K3S_EXEC="--bind-address=#{server_ip} --node-external-ip=#{server_ip} --"
```

```
  curl -sL https://get.k3s.io | sh -
```

```
  echo "Sleeping for 5 seconds to wait for k3s to start"
```

```
  sleep 5
```

```
  cp /var/lib/rancher/k3s/server/token /vagrant_shared
```

```
  cp /etc/rancher/k3s/k3s.yaml /vagrant_shared
```

```
SHELL
```

```

agent_script = <<-SHELL
  sudo -i
  apk add curl
  export K3S_TOKEN_FILE=/vagrant_shared/token
  export K3S_URL=https://#{server_ip}:6443
  export INSTALL_K3S_EXEC="--flannel-iface=eth1"
  curl -sL https://get.k3s.io | sh -
SHELL

Vagrant.configure("2") do |config|
  config.vm.box = "generic/alpine314"

  config.vm.define "server", primary: true do |server|
    server.vm.network "private_network", ip: server_ip
    server.vm.synced_folder "./shared", "/vagrant_shared"
    server.vm.hostname = "server"
    server.vm.provider "virtualbox" do |vb|
      vb.memory = "2048"
      vb.cpus = "2"
    end
    server.vm.provision "shell", inline: server_script
  end

  agents.each do |agent_name, agent_ip|
    config.vm.define agent_name do |agent|
      agent.vm.network "private_network", ip: agent_ip
      agent.vm.synced_folder "./shared", "/vagrant_shared"
      agent.vm.hostname = agent_name
      agent.vm.provider "virtualbox" do |vb|
        vb.memory = "1024"
        vb.cpus = "1"
      end
      agent.vm.provision "shell", inline: agent_script
    end
  end
end

```

As usual, a `vagrant up` and a `vagrant destroy -f` will simply build and tear down the whole setup, as usual.

In the course of these explorations, I found this bug which basically makes k3s pick the first network interface in the VM, instead of using the one we want; a few command line arguments did the trick.