# Solution M6: Nagios

In the following solution, we will create a two-node setup - one node for Nagios and another one for Docker

On **docker** host we will run a simple web-enabled container which will then be monitored in the **nagios** host

Please note, that this is just one possible solution out of many

## Steps

Our **Vagrantfile** can look like this

# -\*- mode: ruby -\*-

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  config.ssh.insert\_key = false

  config.vm.define "nagios" do |nagios|

    nagios.vm.box="shekeriev/centos-8-minimal"

    nagios.vm.hostname = "nagios.dob.lab"

    nagios.vm.network "private\_network", ip: "192.168.99.100"

    nagios.vm.network "forwarded\_port", guest: 80, host: 8081, auto\_correct: true

    nagios.vm.synced\_folder "vagrant/nagios/", "/vagrant"

    nagios.vm.provision "shell", path: "vagrant/nagios.sh"

  end

  config.vm.define "docker" do |docker|

    docker.vm.box="shekeriev/centos-8-minimal"

    docker.vm.hostname = "docker.dob.lab"

    docker.vm.network "private\_network", ip: "192.168.99.101"

    docker.vm.network "forwarded\_port", guest: 80, host: 8082, auto\_correct: true

    docker.vm.synced\_folder "vagrant/docker/", "/vagrant"

    docker.vm.provision "ansible\_local" do |ansible|

      ansible.become = true

      ansible.install\_mode = :default

      ansible.playbook = "playbook-docker.yml"

      ansible.galaxy\_role\_file = "requirements-docker.yml"

      ansible.galaxy\_roles\_path = "/etc/ansible/roles"

      ansible.galaxy\_command = "sudo ansible-galaxy install --role-file=%{role\_file} --roles-path=%{roles\_path} --force"

    end

    docker.vm.provision "shell", path: "vagrant/docker.sh"

  end

end

We will mix the provisioning mechanisms

The **nagios** node will be provisioned with **shell** only

Complementary provisioning file for **nagios** is called **nagios.sh** and has the following content

#!/bin/bash

echo "\* Add hosts ..."

echo "192.168.99.100 nagios.dob.lab nagios" >> /etc/hosts

echo "192.168.99.101 docker.dob.lab docker" >> /etc/hosts

echo "\* SELinux in permisive mode ..."

setenforce 0

echo "\* Prepare the repositories ..."

dnf install -y epel-release

dnf config-manager --set-enabled powertools

echo "\* Install Nagios ..."

dnf install -y nagios nagios-plugins-all nagios-plugins-nrpe

echo "\* Set password (Password1) for nagiosadmin ..."

htpasswd -b /etc/nagios/passwd nagiosadmin Password1

echo "\* Add some nice logos ..."

cp /vagrant/logos/\*.png /usr/share/nagios/html/images/logos

echo "\* Add/change Nagios configuration files ..."

cp /vagrant/cfg/localhost.cfg /etc/nagios/objects/localhost.cfg

mkdir -p /etc/nagios/conf.d

echo "cfg\_dir=/etc/nagios/conf.d" | tee -a /etc/nagios/nagios.cfg

cp /vagrant/custom/\* /etc/nagios/conf.d

echo "\* Start and enable Nagios ..."

systemctl enable nagios

systemctl start nagios

echo "\* Start and enable Apache HTTP ..."

systemctl enable httpd

systemctl start httpd

echo "\* Firewall - open port 80 ..."

firewall-cmd --add-port=80/tcp --permanent

firewall-cmd --reload

The **docker** host will be provisioned with **local Ansible** mainly

To achieve this, first we download a 3rd party role by stating it in the **requirements-docker.yml** file

# from galaxy

- src: geerlingguy.docker

The actual playbook contains the following

---

- hosts: all

  roles:

    - geerlingguy.docker

  tasks:

    - name: Add nagios host

      lineinfile:

        path: /etc/hosts

        line: '192.168.99.100 nagios.dob.lab nagios'

    - name: Add docker host

      lineinfile:

        path: /etc/hosts

        line: '192.168.99.101 docker.dob.lab docker'

    - name: Install EPEL Repository

      dnf:

        name=epel-release

        state=present

    - name: Install NRPE

      dnf:

        name=nrpe,nagios-plugins-nrpe,nagios-plugins-all

        enablerepo=powertools

        state=present

    - name: Substitute nrpe.cfg

      copy:

        src=/vagrant/nrpe.cfg

        dest=/etc/nagios/nrpe.cfg

    - name: Add nrpe to sudoers.d

      copy:

        src=/vagrant/nrpe.sudo

        dest=/etc/sudoers.d/nrpe

    - name: Add nagios to sudoers.d

      copy:

        src=/vagrant/nagios.sudo

        dest=/etc/sudoers.d/nagios

    - name: Adding user vagrant to docker group

      user: name=vagrant

        groups=docker

        append=yes

    - name: Adding user nrpe to docker group

      user: name=nrpe

        groups=docker

        append=yes

    - name: Adding user nagios to docker group

      user: name=nagios

        groups=docker

        append=yes

    - name: Copy additional docker plugin

      copy:

        src=/vagrant/check\_docker\_container.sh

        dest=/usr/lib64/nagios/plugins/check\_docker\_container.sh

    - name: Turn on execute permission for check\_docker\_container.sh

      file:

        path=/usr/lib64/nagios/plugins/check\_docker\_container.sh

        mode="a+x"

    - name: Enable and start NRPE

      service:

        name=nrpe

        state=started

        enabled=true

    - name: Enable HTTP communication

      firewalld:

        service: http

        permanent: true

        state: enabled

        immediate: true

    - name: Enable NRPE (5666/tcp) port

      firewalld:

        port: 5666/tcp

        permanent: true

        state: enabled

        immediate: true

In addition to **docker**, the **NRPE** plugin is installed and configured

And the final touch to start the container that is going to be monitored, we will do with **shell**

#!/bin/bash

echo "\* Start a NGINX container ..."

docker container run -d --name dob-http -p 80:80 -v /vagrant/web:/usr/share/nginx/html nginx

You should pay attention to the register of the container name and what is set in the service

As usual, the whole solution is run with

**vagrant up**