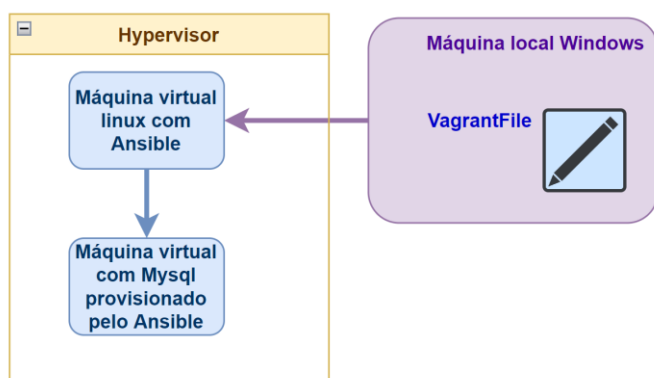




Ansible em windows – Ansible

Outra opção mais popular de **provisionador** é o **Ansible**, diferente do **puppet** o **Ansible** pode ser executar o script de provisionamento diretamente na **máquina HOST** que vai se conectar com a **máquina virtual** via **SSH** por baixo dos panos.

O problema é que o **Ansible** **NÃO FUNCIONA** em **windows**, ele roda apenas em sistemas **UNIX** (**Linux e mac**), pra contornar esse problema é **necessário o uso de uma máquina virtual** que tenha o **Ansible** instalado e a **partir dessa máquina virtual** provisionar outras **máquinas virtuais**:



No **Vagrantfile** eu vou adicionar a **máquina virtual que terá o Ansible** e uma **máquina virtual** que **vai ser provisionada** pela **máquina com o Ansible**:

```
config.vm.define "ansible" do |ansible|
  ansible.vm.network "public_network",
    ip: "192.168.100.44", bridge: "Realtek PCIe GbE Family Controller #2"
end

config.vm.define "mysqlserver" do |mysqlserver|
  mysqlserver.vm.network "public_network",
    ip: "192.168.100.43", bridge: "Realtek PCIe GbE Family Controller #2"
end
```

A **máquina virtual** que contém o **Ansible** precisa que ele **seja instalado**, então obviamente temos que fazer isso com **provisionamento via shell**:

```

config.vm.define "ansible" do |ansible|
  ansible.vm.network "public_network",
    ip: "192.168.100.44", bridge: "Realtek PCIe GbE Family Controller #2"

  ansible.vm.provision "shell",
    inline: ["apt-get update && \
             apt-get install -y software-properties-common && \
             add-apt-repository --yes --update ppa:ansible/ansible && \
             apt-get install -y ansible"]
end

vagrant@vagrant: ~
vagrant@vagrant:~$ ansible-playbook --version
ansible-playbook 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/vagrant/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible-playbook
  python version = 2.7.15+ (default, Nov 27 2018, 23:36:35) [GCC 7.3.0]
vagrant@vagrant:~$

```

Como dito anteriormente, o **Ansible** conecta as máquinas via SSH por baixo dos panos, portanto a máquina que possui o **Ansible** precisa ter a **chave privada** para acessar e provisionar a máquina virtual que precisa conter a **chave pública**.

Já que temos as chaves criadas, precisamos usar o shell para copiar as devidas chaves para suas respectivas máquinas. A **chave pública** será copiada para o **arquivo de authorized keys** na máquina virtual que eu quero acessar:

```

config.vm.define "mysqlserver" do |mysqlserver|
  mysqlserver.vm.network "public_network",
    ip: "192.168.100.43", bridge: "Realtek PCIe GbE Family Controller #2"

  mysqlserver.vm.provision "shell",
    inline: ["cat /vagrant/configs/id_bionic.pub >> .ssh/authorized_keys"]
end

```

A **chave privada** vai precisar ser copiada para a máquina que contém o **Ansible**, também via provisionamento de shell. Esse arquivo que contém a chave privada tem que ter sua **permissão mudada para leitura e escrita** só para **usuário**, outra configuração que precisa ser **mudada é o usuário em si para esse arquivo, a máquina por padrão tem o root mas precisamos que o usuário seja o vagrant**:

```

config.vm.define "ansible" do |ansible|
  ansible.vm.network "public_network",
    ip: "192.168.100.44", bridge: "Realtek PCIe GbE Family Controller #2"

  ansible.vm.provision "shell",
    inline: ["apt-get update && \
             apt-get install -y software-properties-common && \
             add-apt-repository --yes --update ppa:ansible/ansible && \
             apt-get install -y ansible"]

  ansible.vm.provision "shell",
    inline: ["cp /vagrant/id_bionic /home/vagrant/ && \
             chmod 600 /home/vagrant/id_bionic && \
             sudo chown vagrant:vagrant id_bionic"]
end

vagrant@vagrant:~$ pwd
/home/vagrant
vagrant@vagrant:~$ ls
id_bionic
vagrant@vagrant:~$ ls -al
total 44
drwxr-xr-x 5 vagrant vagrant 4096 Feb 11 11:45 .
drwxr-xr-x 3 root root 4096 Aug 15 2019 ..
-rw-r--r-- 1 vagrant vagrant 220 Aug 15 2019 .bash_logout
-rw-r--r-- 1 vagrant vagrant 3771 Aug 15 2019 .bashrc
drwx----- 2 vagrant vagrant 4096 Aug 15 2019 .cache
drwx----- 3 vagrant vagrant 4096 Aug 15 2019 .gnupg
-rw----- 1 vagrant vagrant 2610 Feb 11 11:45 id_bionic
-rw-r--r-- 1 vagrant vagrant 807 Aug 15 2019 .profile
drwx----- 2 vagrant root 4096 Feb 11 11:43 .ssh
-rw-r--r-- 1 vagrant vagrant 0 Aug 15 2019 .sudo_as_admin_successful
-rw-r--r-- 1 vagrant vagrant 6 Aug 15 2019 .vbox_version
-rw-r--r-- 1 root root 180 Aug 15 2019 .wget-hsts
vagrant@vagrant:~$

```

Com isso as configurações de preparações das máquinas está feita.