



## Integração do puppet com Vagrant – Puppet

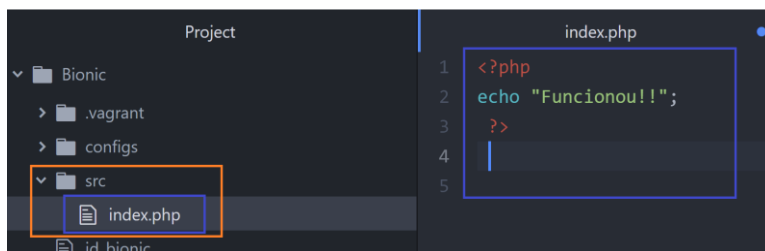
A integração do **puppet** com o **vagrant** é bem simples, primeiro temos que usar o **provisionamento do shell** para que a **máquina virtual** **INSTALE** o **puppet** para que **ela** rode o arquivo. Depois **basta especificar as propriedades de local do arquivo e o nome dele**:

```
config.vm.define "devweb" do |devweb|
  devweb.vm.network "forwarded_port", guest: 80, host: 8090
  devweb.vm.network "public_network",
    ip: "192.168.100.42", bridge: "Realtek PCIe GbE Family Controller #2"

  devweb.vm.provision "shell",
    inline: ["sudo apt-get update && apt-get install -y puppet"]

  devweb.vm.provision "puppet" do |puppet|
    puppet.manifests_path = "./configs/manifests"
    puppet.manifest_file = "devweb.pp"
  end
end
```

Pra ter algum conteúdo para se buscar, dá pra colocar uma página **php bem simples** que vai rodar na **máquina virtual**, mas **que será acessível para** o **host** através da conexão direta por **IP** ou por **port forwarding**:



Para que isso funcione eu preciso **definir o target (-t)** dessa **aplicação PHP** no arquivo do **puppet**, ou seja, como isso vai ser executado na **máquina virtual** ele precisa **buscar o código da pasta compartilhada**. Também defini que na porta **8888** ele vai aceitar conexão por IP de **qualquer máquina(0.0.0.0)**:

```
exec { ['apt-update']:
  command => '/usr/bin/apt-get update'
}

package { ['php7.2', 'php7.2-mysql']:
  require => Exec['apt-update'],
  ensure => installed
}

exec { 'run php-7':
  require => Package['php7.2'],
  command => '/usr/bin/php -S 0.0.0.0:8888 -t /vagrant/src &'
}
```

Também mudei a configuração de **portforwarding**, quando eu acessar a **porta 9091 do meu host** ele vai me **empurrar para a porta 8888 da minha máquina virtual (guest)** onde está rodando a página **PHP**:

```

config.vm.define "devweb" do |devweb|
  devweb.vm.network "forwarded_port", guest: 8888, host: 9091
  devweb.vm.network "public_network",
    ip: "192.168.100.42", bridge: "Realtek PCIe GbE Family Controller #2"

  devweb.vm.provision "shell",
    inline: "sudo apt-get update && apt-get install -y puppet"

  devweb.vm.provision "puppet" do |puppet|
    puppet.manifests_path = "./configs/manifests"
    puppet.manifest_file = "devweb.pp"
  end
end

```

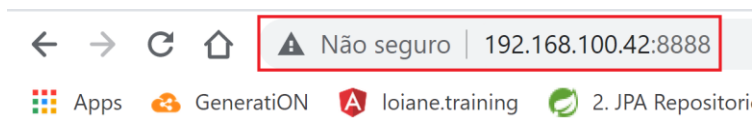
Hora de destruir e reconstruir a **máquina de webdev** e ver se está tudo ok, inclusive com o **PHP já rodando na porta 8888**:

```

vagrant@vagrant:~$ netstat -tnlp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:8888            0.0.0.0:*               LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
vagrant@vagrant:~$ ps aux | grep php
root      22678  0.0  1.6 274564 17132 ?        S   19:55   0:00 /usr/bin/php -S 0.0.0.0:8888 -t /vagrant/src
vagrant   22766  0.0  0.0 14428 1004 pts/0    S+  20:01   0:00 grep --color=auto php
vagrant@vagrant:~$

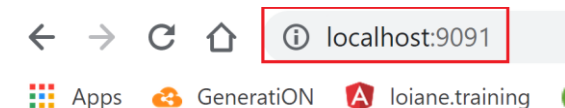
```

Conectando direto pelo **IP da máquina virtual**:



Funcionou!!

Conectando pelo **port forwarding**:



Funcionou!!