

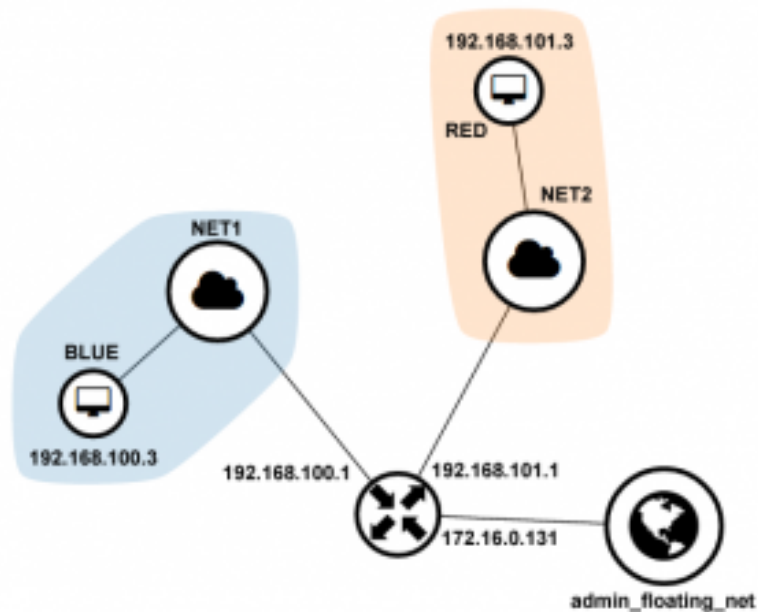
Configurando Redes no OpenStack com o Neutron

Ao criar uma instância no OpenStack é necessário informar uma rede onde ela irá funcionar. Dependendo da aplicação da instância, esta rede pode ser completamente isolada (as instâncias dentro desta rede só poderão se comunicar entre si), compartilhada com outras redes através de roteadores (instâncias se comunicam com instâncias em outras redes), com acesso para a internet (as instâncias podem acessar a internet), com permissão de acesso externo (tráfego externo ao OpenStack, como SSH, HTTP, etc, pode acessar as instâncias) ou qualquer combinação destas. Todas as instâncias recebem um endereço IP da rede fixa quando iniciam. Esta rede é completamente isolada do resto da estrutura até que você informe o contrário. Segundo o [site da OpenStack](#):

... fixed IPs are allocated dynamically by the nova-network component (agora Neutron) when instances boot up. There is no way to tell OpenStack to assign a specific fixed IP to an instance.

O primeiro passo é definir a infraestrutura de rede necessária para suas aplicações. Uma vez feito isso, pode-se criá-las utilizando o Horizon. É possível acessar o Neutron usando a console, mas não está no escopo deste tutorial.

Para o exemplo, vou criar duas redes (192.168.100.0/24 e 192.168.101.0/24). As instâncias nestas redes poderão se comunicar entre si, poderão acessar a internet e receber conexões externas. Estou usando a versão Mitaka [instalada em ambiente VirtualBox com o Mirantis Fuel](#). Após a instalação eu removi o roteador e a rede interna que vieram por padrão. Deixe apenas a rede externa **"admin_floating_net"**.



Proposta de rede

A primeira rede será nomeada **NET1** e a segunda **NET2**. Cada uma delas precisará de uma sub-rede, as quais vou denominar respectivamente **SUBNET1** e **SUBNET2**. Configurei SUBNET1 com a rede 192.168.100.0/0 e SUBNET2 com a rede 192.168.101.0/24.

Redes

<div> <div>Filtrar</div> <div>+</div> <div>Criar Rede</div> <div>Excluir Redes</div> </div>							
<input type="checkbox"/>	NOME	SUB-REDES ASSOCIADAS	COMPARTILHADO	EXTERNO	STATUS	ESTADO DE ADMIN	ACTIONS
<input type="checkbox"/>	admin_floating_net	admin_floating_net_subnet 172.16.0.0/24	Não	Sim	Ativo	UP	<div> <div>Editar Rede</div> <div></div> </div>
<input type="checkbox"/>	NET2	SUBNET2 192.168.101.0/24	Não	Não	Ativo	UP	<div> <div>Editar Rede</div> <div></div> </div>
<input type="checkbox"/>	NET1	SUBNET1 192.168.100.0/24	Não	Não	Ativo	UP	<div> <div>Editar Rede</div> <div></div> </div>

Displaying 3 items

Deixe ativado o DHCP para que as instâncias recebam automaticamente um endereço IP.

Criar Rede

[Rede](#)[Sub-rede](#)[Detalhes da Sub-rede](#)☒ **Habilitar DHCP**

Especifique atributos adicionais para a sub-rede.

Pools de Alocação ?

Deixe DHCP ativado

Na configuração da rede, você verá as “portas” ou endereços IP alocados. Na imagem abaixo da rede NET2, pode-se ver o DHCP (fornecido internamente) usando o IP 192.168.101.2 e uma instância que eu iniciei usando o IP 192.168.101.3. Perceba que não há um *gateway* definido, logo esta rede não possui rota de saída.

Visão Geral de Rede

Nome	NET2
ID	7cea665a-080f-4532-9e28-a340b837ec2c
ID do Projeto	1a001870a5de43fead59106c50cf1d29
Status	Ativo
Estado de Admin	UP
Compartilhado	Não
Rede Externa	Não
MTU	1500
Rede do Provedor	Tipo de Rede: vlan Rede Física: physnet2 ID de Segmentação: 1004

Sub-Redes

[+ Criar Sub-rede](#)[Excluir Sub-redes](#)

<input type="checkbox"/> NOME	ENDEREÇO DE REDE	VERSÃO DO IP	IP DO GATEWAY	ACTIONS
<input type="checkbox"/> SUBNET2	192.168.101.0/24	IPv4		Editar Sub-rede

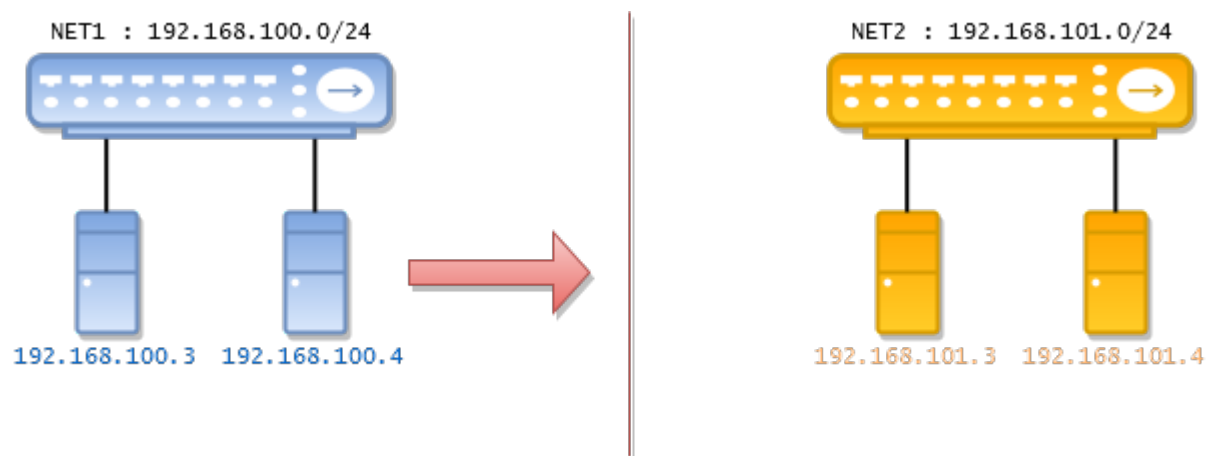
Displaying 1 item

Portas

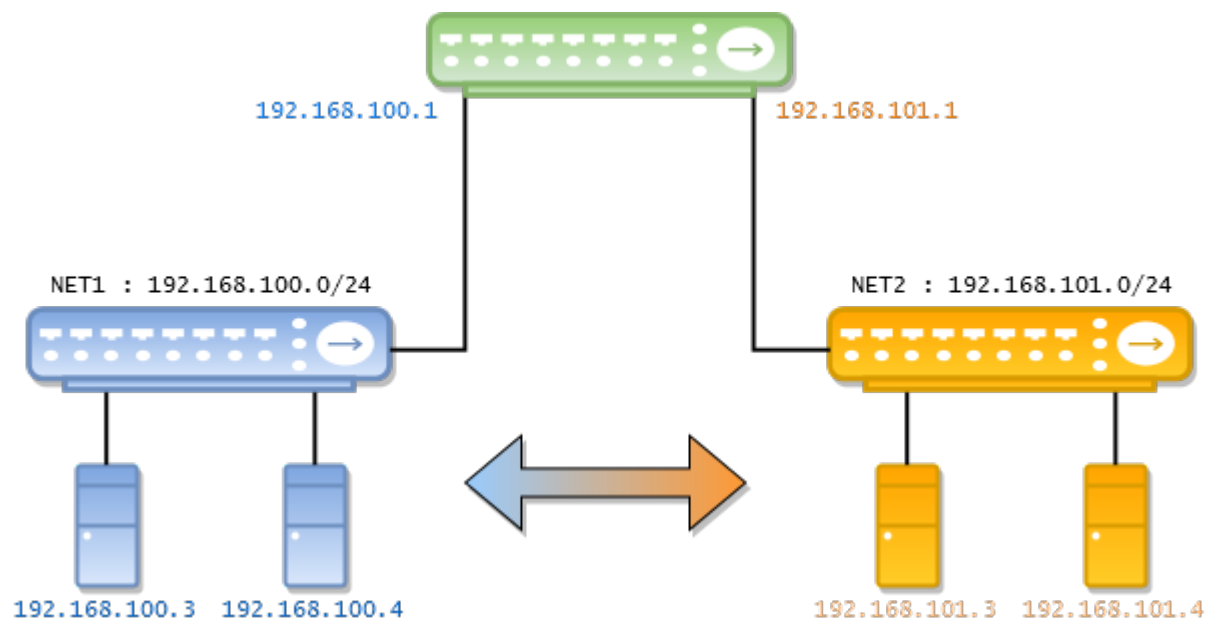
NOME	IPS FIXOS	DISPOSITIVO ANEXADO	STATUS	ESTADO DE ADMIN	ACTIONS
(4328b56b-f781)	192.168.101.2	network:dhcp	Ativo	UP	Editar Porta
(6ee56826-890f)	192.168.101.3	compute:nova	Ativo	UP	Editar Porta

Na configuração atual, as instâncias de uma rede não podem acessar as de outras

redes nem receber conexões externas. Também não podem acessar a internet.



Para que as redes possam se comunicar, é necessário criar um roteador e conectá-lo em ambas. Este roteador, como os roteadores reais, terá duas portas conectadas, uma em cada rede. Então eu criei um roteador chamado “**VR**” (na imagem abaixo ele pode ser visto na cor verde).



Crie um novo roteador, forneça um nome e salve. Clique no link do nome dele e

selecione “Adicionar Interface”. Faça isso para as duas redes que criamos (NET1 e NET2). No Dashboard ele deverá ser configurado da seguinte forma:

Roteadores / VR

Limpar Gateway

Visão Geral

Interfaces

Rotas Estáticas

+ Adicionar Interface

Excluir Interfaces

<input type="checkbox"/>	NOME	IPS FIXOS	STATUS	TIPO	ESTADO DE ADMIN	ACTIONS
<input type="checkbox"/>	(a9d179d6-6bda)	192.168.100.1	Ativo	Interface Interna	UP	Excluir Interface
<input type="checkbox"/>	(d0904d7d-5239)	192.168.101.1	Ativo	Interface Interna	UP	Excluir Interface

Displaying 3 items

Se você verificar as configurações das redes novamente verá que o roteador aparecerá usando as portas com o primeiro endereço IP da faixa de cada rede (como todo *gateway* normalmente faz).

Visão Geral de Rede

Nome

NET2

ID

7cea665a-080f-4532-9e28-a340b837ec2c

ID do Projeto

1a001870a5de43fead59106c50cf1d29

Status

Ativo

Estado de Admin

UP

Compartilhado

Não

Rede Externa

Não

MTU

1500

Rede do Provedor

Tipo de Rede: vlan
Rede Física: physnet2
ID de Segmentação: 1004

Sub-Redes

+ Criar Sub-rede

Excluir Sub-redes

<input type="checkbox"/>	NOME	ENDEREÇO DE REDE	VERSÃO DO IP	IP DO GATEWAY	ACTIONS
<input type="checkbox"/>	SUBNET2	192.168.101.0/24	IPv4	192.168.101.1	Editar Sub-rede

Displaying 1 item

Portas

NOME	IPS FIXOS	DISPOSITIVO ANEXADO	STATUS	ESTADO DE ADMIN	ACTIONS
(d0904d7d-5239)	192.168.101.1	network:router_interface	Ativo	UP	Editar Porta
(6ee56826-890f)	192.168.101.3	compute:nova	Ativo	UP	Editar Porta
(4328b56b-f781)	192.168.101.2	network:dhcp	Ativo	UP	Editar Porta

Perceba que no exemplo da figura acima (NET2) o roteador aparece usando o

endereço IP 192.168.101.1. Na rede NET1 ele usa o endereço IP 192.168.100.1. Isso já permitirá que as instâncias possam se comunicar com instâncias de redes diferentes, mas elas não podem receber conexões externas nem acessar a internet. Para que as instâncias acessem a internet, é necessário conectar o roteador com a rede externa “**admin_floating_net**”, que veio configurada por padrão na instalação do Fuel. Esta rede externa, no caso da instalação em ambiente VirtualBox pelo Fuel, é criada por uma das interfaces *Host Only* do VirtualBox e é visível pela máquina host. Para conectar o roteador com a rede externa, selecione o roteador e clique em “Configurar Gateway”.

Roteadores

<div>Filtrar </div>						<div>+ Criar Roteador</div>	<div>Excluir Roteadores</div>
<input type="checkbox"/>	NOME	STATUS	REDE EXTERNA	ESTADO DE ADMIN		ACTIONS	
<input type="checkbox"/>	VR	Ativo	-	CIMA		Configurar Gateway	

Roteador sem Rede Externa

Selecione a rede externa na caixa de diálogo que irá surgir e salve. Somente as redes marcadas como “externa” na lista de redes é que vão aparecer nesta caixa. Eu havia apagado a rede externa padrão criada pela instalação do Fuel e tentei criar uma nova, mas me enrolei em alguma coisa e ela não funcionou. Até que eu possa dizer com certeza como criar uma rede externa, mantenha ela no lugar para evitar problemas.

Redes

<div>Filtrar </div>								<div>+ Criar Rede</div>	<div>Excluir Redes</div>
<input type="checkbox"/>	NOME	SUB-REDES ASSOCIADAS	COMPARTILHADO	EXTERNO	STATUS	ESTADO DE ADMIN	ACTIONS		
<input type="checkbox"/>	admin_floating_net	admin_floating_net_subnet 172.16.0.0/24	Não	 Sim	Ativo	UP	Editar Rede		
<input type="checkbox"/>	NET2	SUBNET2 192.168.101.0/24	Não	Não	Ativo	UP	Editar Rede		
<input type="checkbox"/>	NET1	SUBNET1 192.168.100.0/24	Não	Não	Ativo	UP	Editar Rede		

Displaying 3 items

Detalhe da Rede Externa

Agora o roteador vai mostrar a rota para a rede externa. Note que ele recebeu mais uma interface com um endereço IP válido para a rede externa (172.16.0.131).

Roteadores / VR

Limpar Gateway

Visão Geral

Interfaces

Rotas Estáticas

+ Adicionar Interface

Excluir Interfaces

<input type="checkbox"/>	NOME	IPS FIXOS	STATUS	TIPO	ESTADO DE ADMIN	ACTIONS
<input type="checkbox"/>	(0f2a7fb2-8e0d)	172.16.0.131	Ativo	Gateway Externo	UP	Excluir Interface
<input type="checkbox"/>	(a9d179d6-6bda)	192.168.100.1	Ativo	Interface Interna	UP	Excluir Interface
<input type="checkbox"/>	(d0904d7d-5239)	192.168.101.1	Ativo	Interface Interna	UP	Excluir Interface

Displaying 3 items

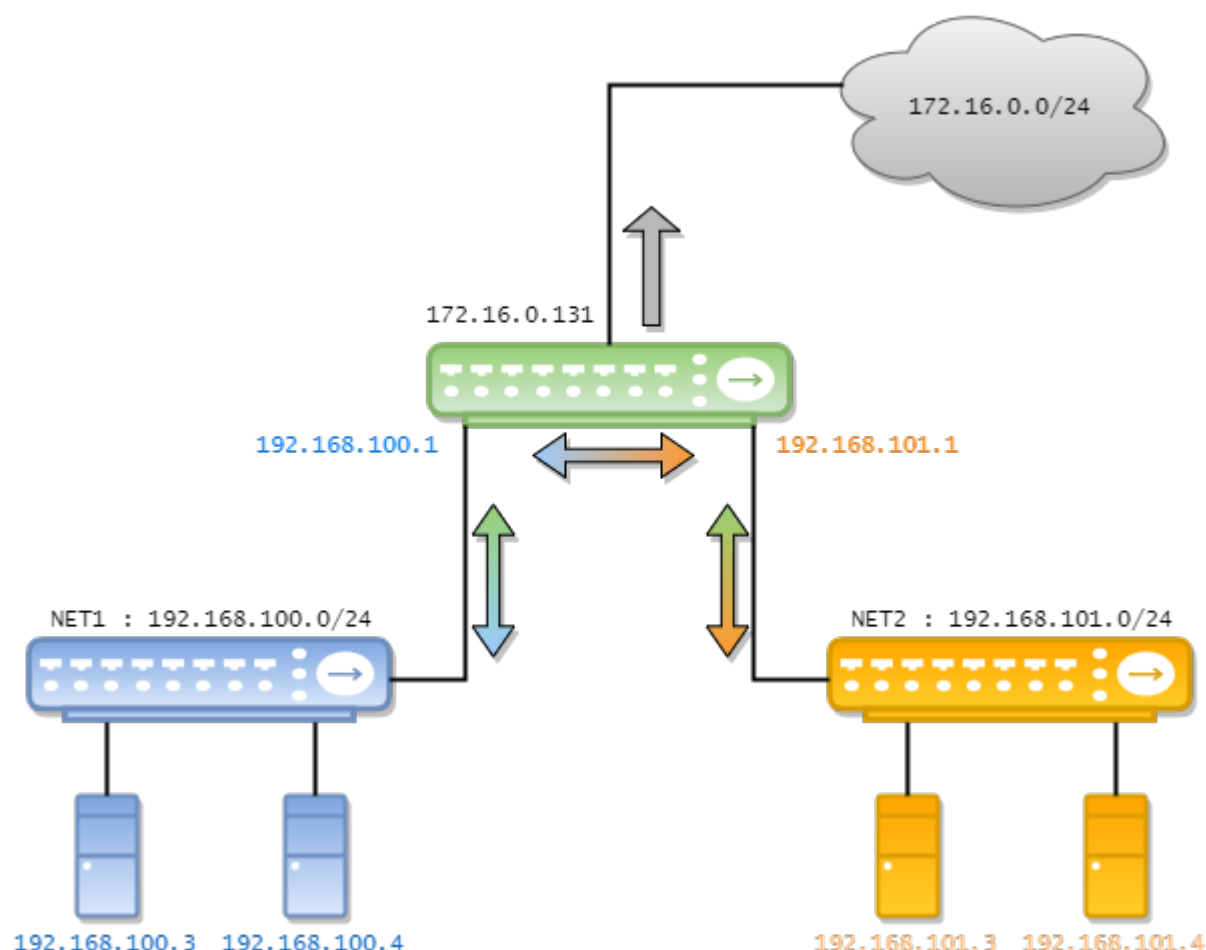
Na máquina host (a máquina física que executa o VirtualBox) já é possível ver que este endereço IP existe.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [versão 6.3.9600]
(c) 2013 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Magno>ping 172.16.0.131

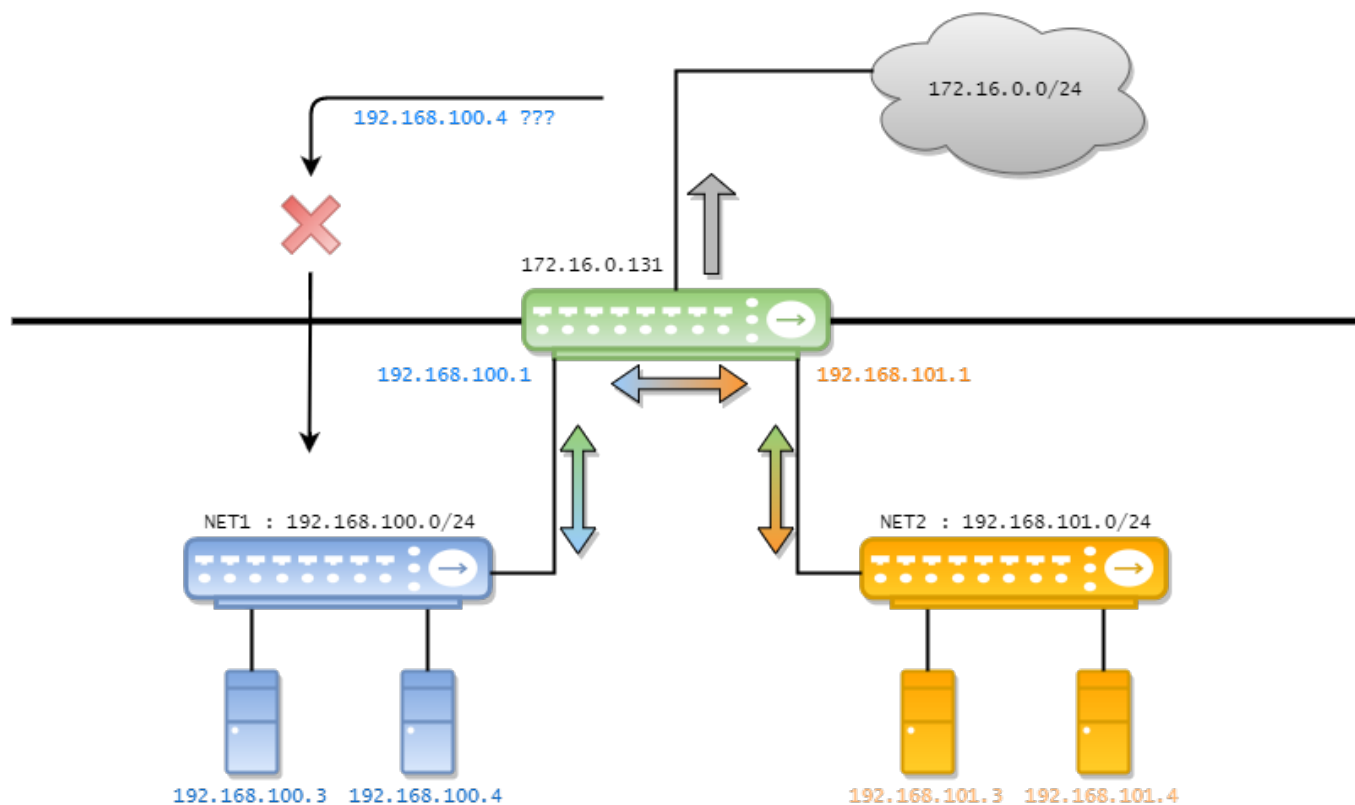
Disparando 172.16.0.131 com 32 bytes de dados:
Resposta de 172.16.0.131: bytes=32 tempo=2ms TTL=64
Resposta de 172.16.0.131: bytes=32 tempo=2ms TTL=64
Resposta de 172.16.0.131: bytes=32 tempo=2ms TTL=64
Resposta de 172.16.0.131: bytes=32 tempo=2ms TTL=64

Estatísticas do Ping para 172.16.0.131:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
    perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 2ms, Máximo = 2ms, Média = 2ms
C:\Users\Magno>_
```

Nossa infraestrutura está um pouco melhor agora. As instâncias das duas redes podem se comunicar e acessar a internet.



Porém elas não podem receber nenhuma conexão do mundo exterior. Isso seria muito útil quando você cria um servidor de banco de dados, HTTP, ou até mesmo se você precisar um acesso SSH ou FTP para suas instâncias. Na minha opinião, um acesso SSH é indispensável. as instâncias não podem receber conexão do mundo exterior porque elas possuem endereços válidos apenas para suas redes. É parecido com o que acontece na sua casa: seu computador possui endereços válidos apenas para sua rede interna. O endereço IP que conta para o mundo exterior (internet) é o do seu modem ADSL. Não importa quantos computadores você tenha em casa, todos acessam a Internet usando o IP do seu modem. No nosso caso, todas as instâncias acessam a rede externa usando o IP externo do nosso roteador (172.16.0.131). Não há como acessar uma instância dentro da rede interna dessa forma porque aquele endereçamento (192.168.aaa.bbb) não é reconhecido pela rede externa.

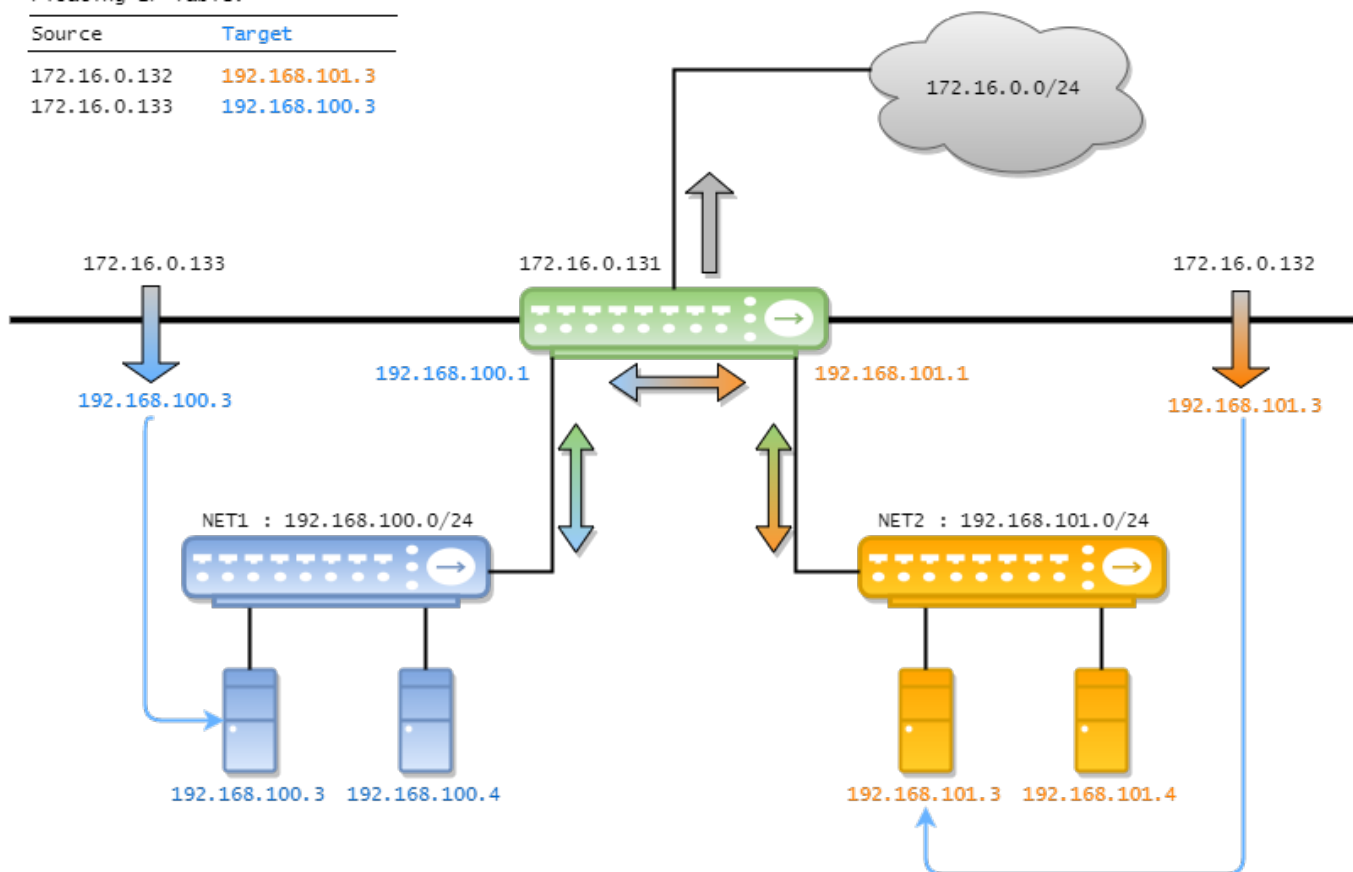


Nós precisamos dar um endereço válido na rede externa para as instâncias, assim, quando desejarmos acessar uma determinada instância dentro da rede interna, usaremos o IP da rede externa. O Neutron usará este mapeamento para encontrar o destino do acesso. É aí que entram os “*floating IPs*”. Os *floating IPs* nada mais são do que mapeamentos de endereços IP da rede externa para endereços IP da rede interna. Cada instância que receber um *floating IP* passa a ser reconhecida na rede externa usando este endereço. Você encontrará uma boa explicação sobre [floating IPs aqui](#).

Nossa rede deverá ficar assim no final (o roteador “**VR**” está em verde):

Floating IP Table:

Source	Target
172.16.0.132	192.168.101.3
172.16.0.133	192.168.100.3



Para designar um floating IP para uma instância, primeiro você deverá iniciá-la e depois selecionar “Associar IP flutuante” no combo lateral na lista de instâncias. O resultado final ficará como na imagem abaixo (lista de instâncias):

Instâncias

Nome da Instância = <input type="text"/>								
<input type="checkbox"/>	NOME DA INSTÂNCIA	NOME DA IMAGEM	ENDEREÇO IP	TAMANHO	PAR DE CHAVES	STATUS	ZONA DE DISPONIBILIDADE	TAREFA
<input type="checkbox"/>	RED	TestVM	192.168.101.3 IPs Flutuantes: 172.16.0.132	m1.tiny	-	Desligar	nova	Nenhum
<input type="checkbox"/>	BLUE	TestVM	192.168.100.3 IPs Flutuantes: 172.16.0.133	m1.tiny	-	Desligar	nova	Nenhum

Displaying 2 items

Mas ainda não basta. Ainda é necessário um passo adicional: configurar as

permissões de acesso. Nas permissões de acesso é possível informar quais portas e protocolos estão liberados na máquina de destino do mapeamento. Isso possibilita um maior controle sobre o acesso externo.

Vá em “Acesso e Segurança”. Você vai encontrar o grupo “*default*”, que é designado para a instância caso você não informe nada. É claro que você poderá criar seu próprio grupo e definir ele para a instância no momento de sua criação. Mas vamos ficar com o padrão mesmo. Clique em “Gerenciar Regras”. No meu caso eu resolvi tocar bandalha e permiti acesso geral, mas você pode (e deve) ser mais seletivo. As opções são intuitivas.

Acesso e Segurança / Gerenciar Regras de Grupo de Segurança: default (d8cbf89f-8ce4-468a-9533-26b16daeedad)

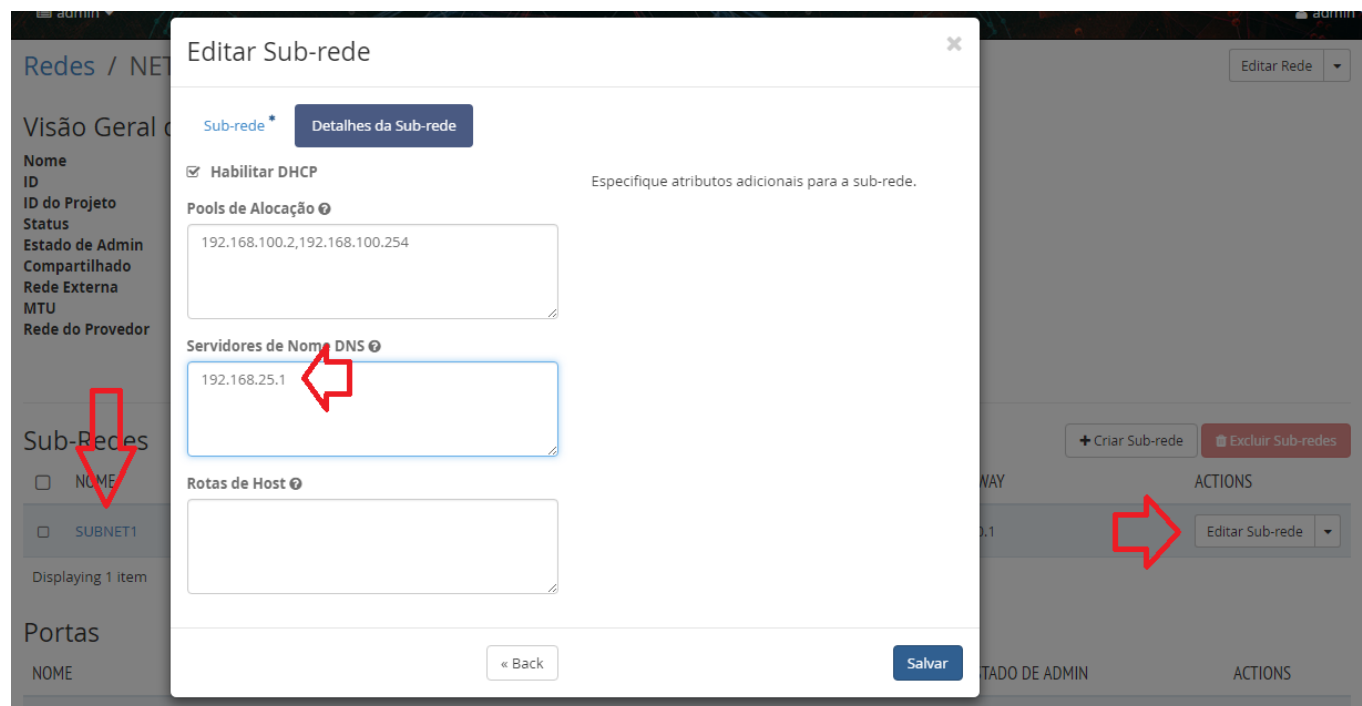
							+ Adicionar Regra	Excluir Regras
<input type="checkbox"/>	DIREÇÃO	TIPO ETHER	PROTOCOLO IP	FAIXA DE PORTAS	PREFIXO DO IP REMOTO	GRUPO DE SEGURANÇA REMOTO	ACTIONS	
<input type="checkbox"/>	Ingresso	IPv4	Qualquer	Qualquer	-	default	Excluir Regra	
<input type="checkbox"/>	Egresso	IPv4	Qualquer	Qualquer	0.0.0.0/0	-	Excluir Regra	
<input type="checkbox"/>	Ingresso	IPv6	Qualquer	Qualquer	-	default	Excluir Regra	
<input type="checkbox"/>	Egresso	IPv6	Qualquer	Qualquer	::/0	-	Excluir Regra	
<input type="checkbox"/>	Ingresso	IPv4	ICMP	Qualquer	172.16.0.0/24	-	Excluir Regra	
<input type="checkbox"/>	Ingresso	IPv4	TCP	1 - 65535	172.16.0.0/24	-	Excluir Regra	
<input type="checkbox"/>	Ingresso	IPv4	UDP	1 - 65535	172.16.0.0/24	-	Excluir Regra	

Displaying 7 items

Perceba que eu selecionei “Ingresso”. Permiti protocolos ICMP, TCP e UDP em todas as portas vindos da rede 172.16.0.0/24 (nossa rede externa). Isso certamente é uma boa fonte de problemas, pois libera todo tipo de acesso (inclusive o PING) para suas instâncias. Mas Deus ajuda. Caso você não tenha tanta fé assim, use o bom senso e escolha as opções de acordo com a necessidade.

Mas ainda não acabou. Quando eu acessei uma das instâncias usando o SSH, pude perceber que ela realmente acessava a rede externa, contanto que eu só usasse endereços IP. No caso de usar nomes (ping `www.google.com`) não funcionava. Percebi que o DNS estava designado para o gateway da instância, no caso o roteador “VR”. Precisei então modificar o DNS para o meu modem ADSL (ou para um DNS público como o do Google). É certo que existem outras formas, mas a mais conveniente pra mim foi mudar o DNS na sub-rede. Fui na configuração das redes (*NET1* e *NET2*) e cliquei em “Editar Sub-rede”. Em “Servidores de Nomes DNS” eu simplesmente coloquei o IP do meu modem ADSL e pronto. Ao salvar pude retornar para a console da instância e ver que ela já sabia resolver nomes.

Magicamente meu arquivo “*resolv.conf*” foi modificado para o IP do meu modem. Na mesma tela você poderá definir a faixa de alocação de endereços IP para sua rede em “*Pools de Alocação*”.



É isso. Fico por aqui deixando as referências onde consegui material para este post (menos o DNS que eu tive que bater cabeça).

Referências:

[Configuring Floating IP addresses for Networking in OpenStack Public and Private Clouds.](#)

[\[VIDEO\] : Introduction to OpenStack Neutron](#) (David Mahler)

[\[VIDEO\] : Introduction to OpenStack Neutron](#) (Assaf Muller)

[Networking in too much detail — RDO](#)

instalando o OpenStack com o DevStack

Após gastar uns 15 dias [brincando com o Mirantis Fuel](#), resolvi instalar o OpenStack usando o DevStack.

A instalação é a mais simples de todas. Basta baixar os arquivos do repositório e executar o instalador. Eu segui exatamente desta forma, como em quase todos os exemplos que vi na internet. O problema é que o resultado é uma instalação extremamente pobre. Dá para ver umas poucas opções, iniciar umas instâncias e não é possível ver o Neutron funcionando (configurar redes e roteadores). Já estava acostumado a ver o Neutron e o Murano, então não achei legal fazer um *test-drive* em algo tão limitado.

Por sorte inventaram o Google. Após algumas buscas, descobri umas dicas para incrementar minha instalação. Primeiro baixei o DevStack do repositório. Informei o *branch* **mitaka** só para garantir.

Meu usuário padrão possui permissão para sudo. Fui para pasta home dele, mas sem sudo.

```
git clone https://git.openstack.org/openstack-dev/devstack -b  
stable/mitaka
```

Para a instalação padrão, bastaria agora executar

```
./stack.sh
```

porém eu preciso fazer algumas considerações antes. É possível usar um arquivo de configuração que é bem flexível e possibilita controlar diversos aspectos da instalação, mas é necessário criá-lo na pasta raiz do instalador (mesma pasta do stack.sh).

```
cd devstack
```

```
vim local.conf
```

É obrigatório iniciar o arquivo com a seguinte linha:

```
[[local|localrc]]
```

Na instalação padrão, o instalador irá pedir as senhas que serão posteriormente usadas no sistema. É possível definir estas senhas no arquivo de configuração.

```
ADMIN_PASSWORD=suasenha  
MYSQL_PASSWORD=suasenha  
RABBIT_PASSWORD=suasenha  
SERVICE_PASSWORD=suasenha  
SERVICE_TOKEN=xynMtSp9KS59SPHp  
SWIFT_HASH=JRBj7ukxMG4tckek
```

As senhas não precisam ser as mesmas.

Quando eu tentava instalar recebia um erro informando que não era possível encontrar a URL de autenticação e a instalação parava. Após procurar muito, encontrei a solução. Editei o arquivo *stack.sh* e localizei a parte onde são definidas as variáveis para acesso ao Keystone:

```
# Set up password auth credentials now that Keystone is  
bootstrapped  
export OS_IDENTITY_API_VERSION=3  
export OS_AUTH_URL=$KEYSTONE_AUTH_URI  
export OS_USERNAME=admin  
export OS_USER_DOMAIN_ID=default  
export OS_PASSWORD=$ADMIN_PASSWORD  
export OS_PROJECT_NAME=admin  
export OS_PROJECT_DOMAIN_ID=default  
export OS_REGION_NAME=$KEYSTONE_REGION_NAME
```

Comente as linhas:

```
# Set up password auth credentials now that Keystone is  
bootstrapped  
export OS_IDENTITY_API_VERSION=3  
export OS_AUTH_URL=$KEYSTONE_AUTH_URI  
export OS_USERNAME=admin  
# export OS_USER_DOMAIN_ID=default  
export OS_PASSWORD=$ADMIN_PASSWORD  
export OS_PROJECT_NAME=admin  
# export OS_PROJECT_DOMAIN_ID=default
```

```
export OS_REGION_NAME=$KEYSTONE_REGION_NAME
```

Agora eu preciso do Neutron e do Murano. Para isso eu usei a configuração que encontrei [aqui](#). Esta configuração ainda me brindou com o Sahara e o Trove que eu queria tanto! Agora eu pude executar o instalador.

```
./stack.sh
```

Isso demora! Obviamente as coisas não foram bem logo da primeira vez e precisei reinstalar tudo várias vezes. Dizem as más línguas que basta executar

```
./unstack.sh
```

para limpar a instalação e começar novamente. Eu tive alguns problemas com isso e resolvi apagar todo o diretório onde os arquivos são instalados:

```
./unstack.sh  
rm -R /opt/stack/*  
./stack.sh
```

Isso é bem radical e faz as coisas demorarem bem mais, porém garante uma reinstalação mais limpa.

O resultado é um ambiente *tudo-em-um*. Todos os serviços ficam disponíveis na mesma máquina. Isso é ótimo para testar, mas é péssimo se você quer entender a infra do OpenStack.

Não sei se é devido a minha máquina estar conectada à rede por uma interface wifi, mas quando reiniciei o servidor eu perdi o acesso ao Horizon (conexão recusada). Pensei que fosse devido às mexidas que fiz na configuração da rede e reinstalei a bagaça toda mais uma vez só para descobrir que o problema acontece mesmo sem eu mexer em nada. Mais uma vez não fiquei correndo atrás de solução. Não acho que vale a pena empregar tempo para consertar uma coisa que não vou usar.

Resumo: apesar de ser a única instalação que me deu o Trove, eu não gostei. Vou retomar minha instalação no VirtualBox Windows para poder seguir adiante com meus testes pois o caminho é longo. Só lamento o Fuel não instalar o Trove. Eles se amarraram com a Tesora (que é pago). Lamentável.

Mirantis Fuel OpenStack : Resumo

Após terminar de [instalar o OpenStack no VirtualBox em um Ubuntu Server usando o Fuel](#), eu resolvi postar algumas observações:

A [instalação no Windows](#) é de longe a mais prática, mas o desempenho não é dos melhores. Precisei comprar mais 8GB de memória para ficar com 16GB e poder ficar tranquilo.

A [instalação no Ubuntu Server](#) é um pouco chata, mas não é difícil. O desempenho é muito melhor, mas como não há interface gráfica, é preciso alguns macetes para poder acessar o Horizon e o painel do Fuel. Não pude acessar as instâncias das VM por causa do erro que dá quando tento colocar um IP flutuante. Não me incomodei muito em tentar resolver o erro, mas achei uns [vídeos excelentes](#) que explicam muito bem como o Neutron funciona. É capaz da solução estar ali.

A conclusão: se for para testar o OpenStack usando o Fuel, é muito melhor usar o Windows. Deixe o Linux para quando for jogar pra valer.

Apesar disso, vou testar o DevStack.

[Video] OpenStack Neutron

Encontrei estes excelentes vídeos explicativos sobre a infraestrutura de rede no OpenStack (Neutron):

Os direitos pertencem aos seus respectivos proprietários. Para mais detalhes acesse as respectivas páginas YouTube.

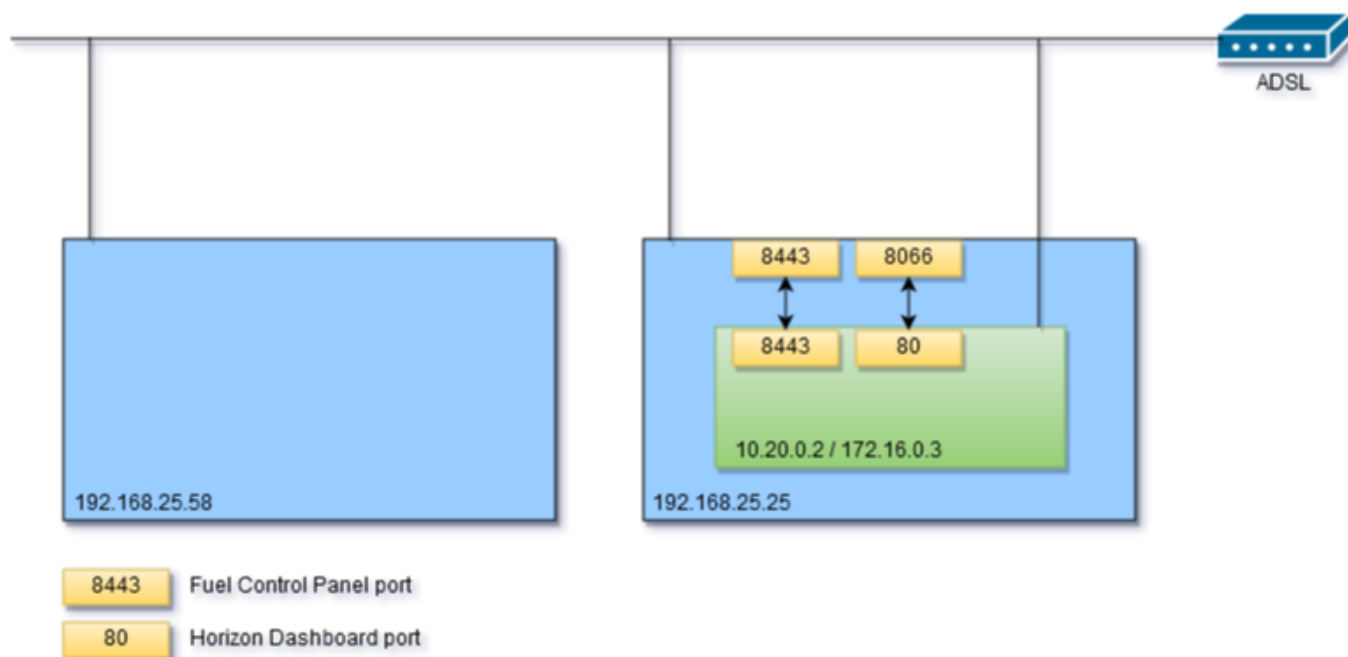
Deploy do OpenStack no Ubuntu Server

Após [instalar o Mirantis Fuel](#) em um ambiente virtualizado hospedado no Ubuntu Server, chegou a hora de instalar o OpenStack (novamente). O procedimento não é [diferente do anterior](#), hospedado em ambiente Windows. Esperei bastante tempo para que o processo terminasse.

No meio do caminho surgiu um problema: a instalação do Ubuntu no slave que estava destinado ao Cinder deu erro. Não consegui descobrir que tipo de erro foi, apesar de existir um sistema de log bem completo. Parti logo para a solução. Como a instalação nos outros dois nós foi tranquila e não tinha nenhuma opção de instalar somente no nó defeituoso, cliquei em “Deploy” (aba “Dashboard”) novamente. Achei que ia refazer tudo, mas ele teve a decência de instalar somente o nó que deu erro, sem mexer nos outros dois.

Após o Ubuntu ter sido “instalado com sucesso” nos 3 nós, o instalador iniciou o *deploy* do OpenStack. Nesse ponto você pode ir visitar seu tio na Noruega, porque vai demorar (pelo menos na minha máquina demorou).

Após o *deploy*, o link para o Horizon foi exibido. Mesmo problema de acesso para o painel do Fuel: o endereço do Horizon não é exposto para a minha rede local, então preciso redirecionar portas novamente. Nesse caso, como eu tenho um Apache instalado na máquina host ([O PHPVirtualBox lembra?](#)) preciso colocar o host para escutar em outra porta, mas enviar para a porta 80 da máquina onde está o Horizon.



Vou aplicar a mesma regra *iptables* da outra vez, mas trocando as portas. Para ser sincero, não lembro agora se é a mesma máquina que está usando os dois endereços (10.20.0.2 e 172.16.0.3). Vou verificar e edito o post depois. O certo é que a rede 10.20.0.x é a rede administrativa do Fuel (PXE). A 172.16.0.x é a pública e *floating* do OpenStack. Como estamos em um ambiente VirtualBox , é necessário criar esta ponte da rede externa da máquina host para a rede pública/*floating* do OpenStack. sempre que se desejar um acesso de fora para dentro (de dentro para fora ele já se resolve, pois já vi que os nós são capazes de acessar a internet pela minha rede doméstica).

```
iptables -I FORWARD -d 172.16.0.3 -m comment --comment "Accept to forward Horizon DashBoard traffic" -m tcp -p tcp --dport 8066 -j ACCEPT
```

```
iptables -I FORWARD -m comment --comment "Accept to forward Horizon DashBoard return traffic" -s 172.16.0.3 -m tcp -p tcp --sport 80 -j ACCEPT
```

```
iptables -t nat -I PREROUTING -m tcp -p tcp --dport 8066 -m comment --comment "redirect pkts to Horizon Dashboard" -j DNAT --to-destination 172.16.0.3:80
```

```
iptables -t nat -I POSTROUTING -m comment --comment "NAT the src ip" -d 172.16.0.3 -o vboxnet1 -j MASQUERADE
```

Atualizei minha cópia do *iptables* para que isso não se perca ao reiniciar o servidor:

```
sudo iptables-save > /etc/iptables.conf
```

Feito isso, pude acessar o Horizon a partir do meu notebook, assim como faço com o painel de controle do Fuel.

<http://192.168.25.25:8066/>

No meu caso, tive um problema ao acessar o Horizon:

The server encountered an internal error or misconfiguration and was unable to complete your request.

Eu usei a Conexão de Área de Trabalho Remota do Windows para conectar no nó onde está o Horizon. Se não for a porta 5001 então tente outro. Não lembro de foi um bom palpite ou se eu verifiquei em algum lugar.:

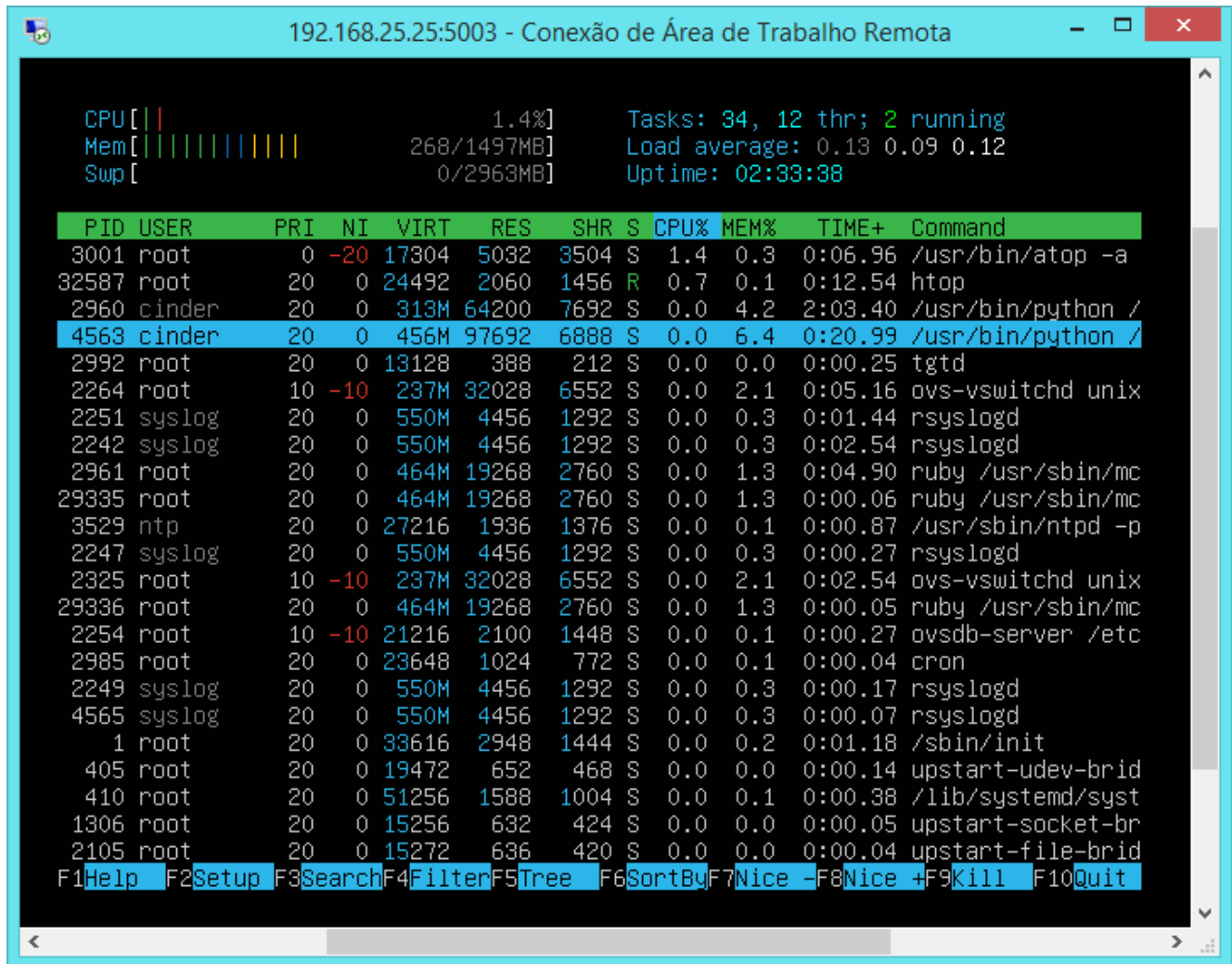
<IP_D0_HOST_VIRTUALBOX>:5001

Uma vez logado como root (senha **r00tme**) eu simplesmente reiniciei o Apache:

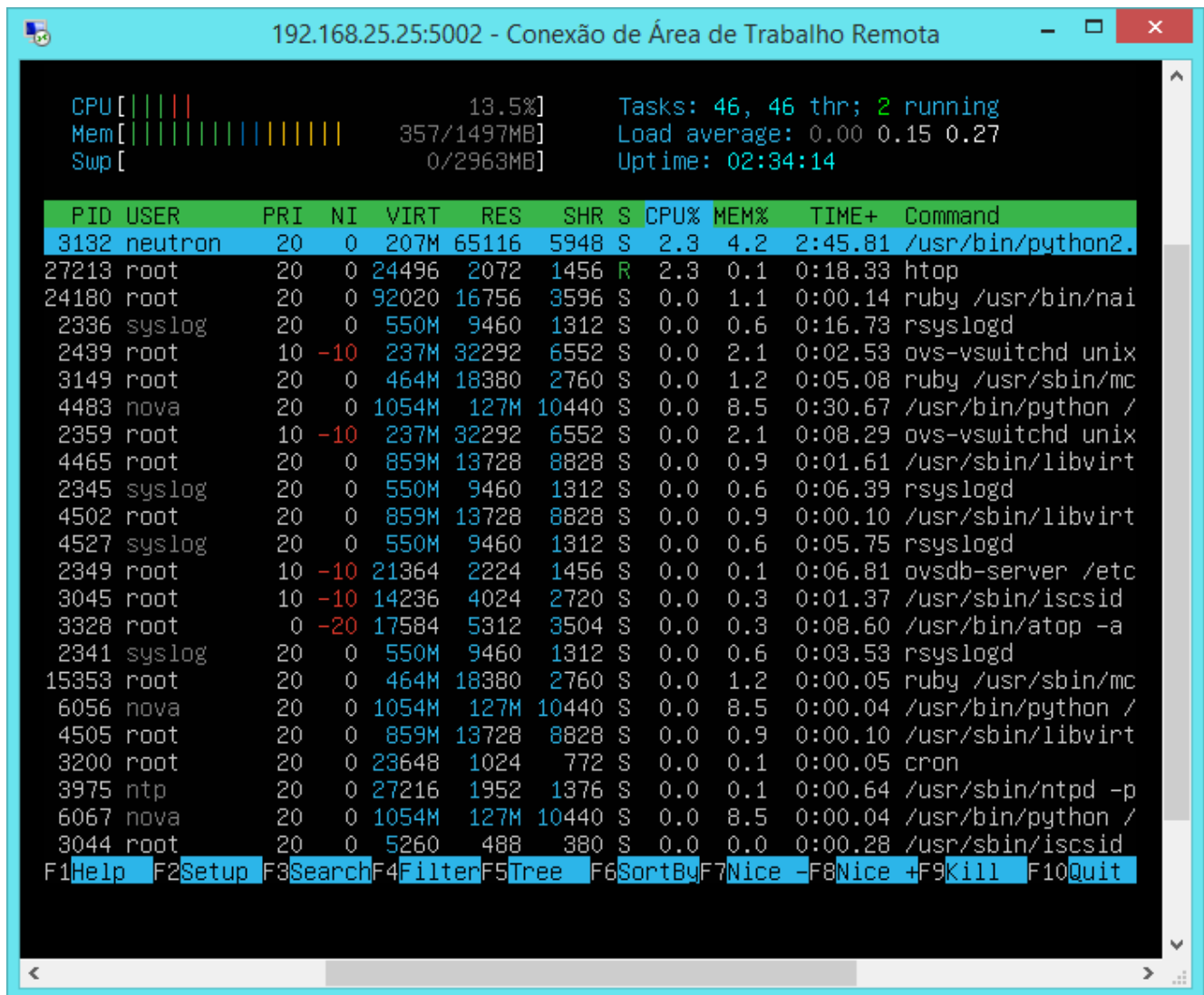
```
service apache2 restart
```

e tudo funcionou como deveria.

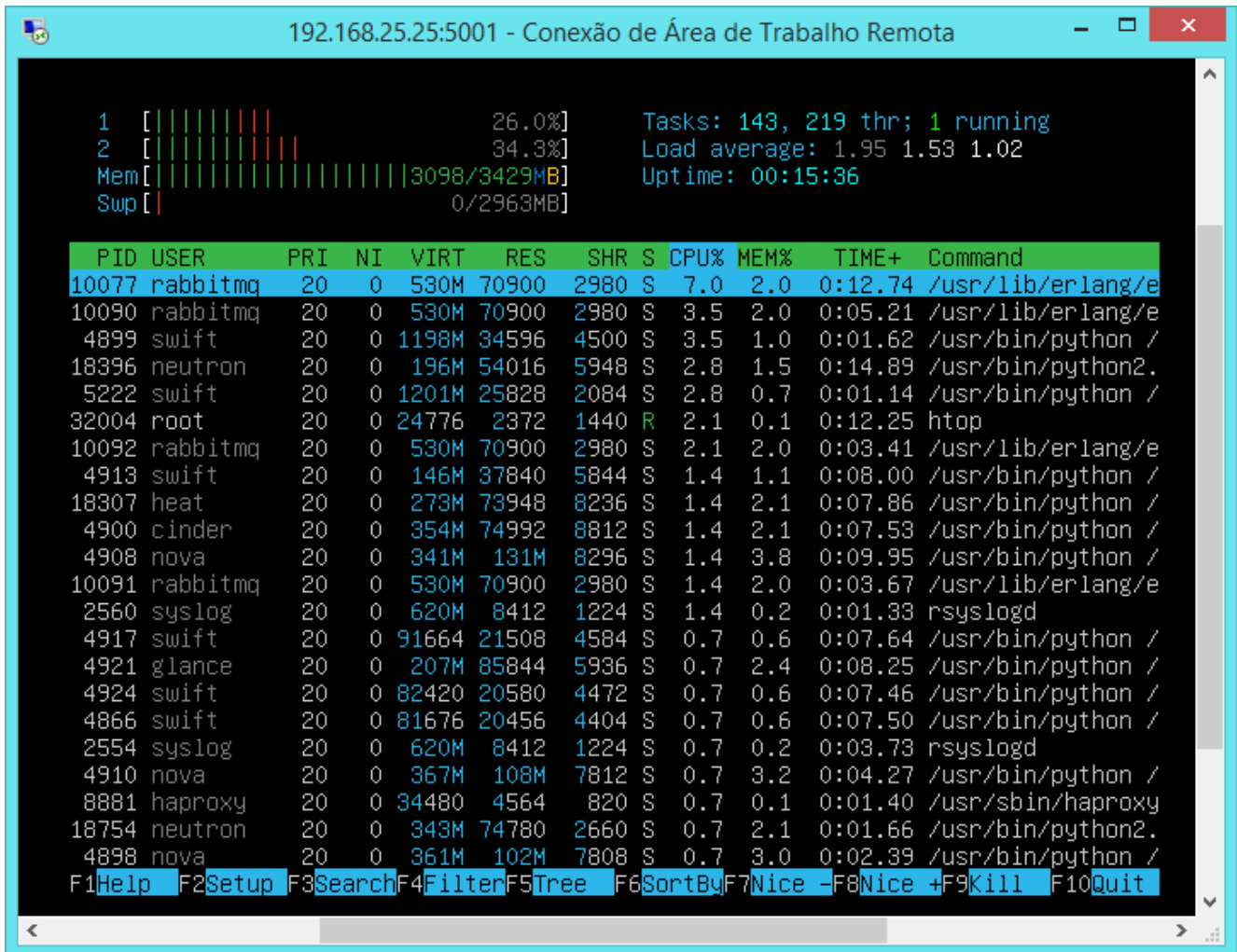
Só por curiosidade eu acessei cada máquina e executei o aplicativo htop do Ubuntu para monitorar a utilização de recursos. Percebi que o Controller estava com 1.3 GB de RAM (assim como todos os outros) mas estava paginando demais e consumindo toda a memória disponível. Apesar de não estar nos manuais, resolvi disponibilizar mais um processador para esta VM e também resolvi aumentar a memória para 3.5 GB de RAM. Após isso, reiniciei a VM do Controller e percebi que o acesso ao Horizon ficou mais fuido e que não estava paginando tanto. Monitorei também a memória disponível na máquina host para ver o impacto da minha modificação. Tudo dentro do normal, pelo menos sem executar nada no OpenStack.



Cinder



Compute



Controller

Após tudo isso, tentei executar uma instância. Escolhi a imagem "TestVM" que veio na instalação, tamanho m1.tiny.

Quando eu seleciono a rede "admin_internal_net" tudo vai bem. A máquina sobe, mas não consigo acessá-la de nenhum lugar (o IP da rede interna não é exposto). Quando eu escolho a rede "admin_floating_net" ocorre um erro:

No valid host was found. There are not enough hosts available.

Não sei bem o que é isso. Procurei pela internet, mas cada lugar dá uma solução diferente e nenhuma resolveu o problema. Aloquei alguns endereços IP flutuantes e criei uma chave, mas também não resolveu. Quando descobrir o que houve eu posto aqui.

No valid host was found. There are not enough hosts available.

File `"/usr/lib/python2.7/dist-`

```
packages/nova/conductor/manager.py", line 739, in
build_instances request_spec, filter_properties) File
"/usr/lib/python2.7/dist-packages/nova/scheduler/utils.py",
line 343, in wrapped return func(*args, **kwargs) File
"/usr/lib/python2.7/dist-
packages/nova/scheduler/client/__init__.py", line 52, in
select_destinations context, request_spec, filter_properties)
File
"/usr/lib/python2.7/dist-
packages/nova/scheduler/client/__init__.py", line 37, in
__run_method return getattr(self.instance, __name)(*args,
**kwargs) File
"/usr/lib/python2.7/dist-
packages/nova/scheduler/client/query.py", line 34, in
select_destinations context, request_spec, filter_properties)
File
"/usr/lib/python2.7/dist-
packages/nova/scheduler/rpcapi.py", line 120, in
select_destinations request_spec=request_spec,
filter_properties=filter_properties) File
"/usr/lib/python2.7/dist-
packages/oslo_messaging/rpc/client.py", line 158, in call
retry=self.retry) File
"/usr/lib/python2.7/dist-
packages/oslo_messaging/transport.py", line 90, in _send
timeout=timeout, retry=retry) File
"/usr/lib/python2.7/dist-
packages/oslo_messaging/_drivers/amqpdriver.py", line 533, in
send retry=retry) File
"/usr/lib/python2.7/dist-
packages/oslo_messaging/_drivers/amqpdriver.py", line 524, in
_send raise result
```

Edit:

[A solução para este erro é a seguinte](#): eu estava designando a instância para uma rede externa sem que ela tenha um IP válido na rede fixa (por desconhecimento meu na infraestrutura de rede do Neutron). Toda instância DEVE possuir um IP fixo na rede interna designado pelo Nova e só então receber um IP flutuante externo. A expressão *"not enough hosts available"* significa que ele não encontrou um IP fixo (host) disponível (available) da rede interna designado para a instância que possa ser mapeado para o IP da rede externa.

OpenStack: Mirantis Fuel no Ubuntu Server

Tendo conseguido [instalar uma versão do OpenStack usando o Mirantis Fuel](#) em um ambiente VirtualBox hospedado em uma máquina Windows, meu próximo passo foi fazer o mesmo em um ambiente VirtualBox hospedado em uma máquina Ubuntu 14.04.

Eu sei, já existe versão LTS mais recente. Me deixe em paz, panela velha é que faz comida boa.

Peguei um HD de 300 e poucos GB de um notebook velho que tinha aqui e pluguei na mesma máquina de antes (um Core i5 com 8GB de RAM). É o que o sistema oferece. Tive o cuidado de remover meu HD com o Windows. Só Jesus salva.

Com meu Ubuntu fresquinho, fiz a prece do recém-instalado:

```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

Instalando o VirtualBox com Interface Gráfica WEB:

Parti então para a instalação do VirtualBox. Para tanto, usei [este tutorial](#). Só modifiquei as versões usadas. Vou repetir os passos usados:

```
apt-get install build-essential dkms
```

Crie o seguinte arquivo:

```
vi /etc/apt/sources.list.d/virtualbox.list
```

e coloque nele o seguinte conteúdo:

```
deb http://download.virtualbox.org/virtualbox/debian trusty
contrib
```

Baixe e instale a chave:

```
wget -q http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O- | sudo apt-key add -
```

e instale o VirtualBox. A minha versão difere do tutorial. Para verificar a versão mais recente vá no site do VirtualBox e troque o número da versão de acordo.

```
sudo apt-get update && sudo apt-get install VirtualBox-5.1
```

Minha versão do Extension Pack também é diferente (vá para seu diretório *home*):

```
wget http://download.virtualbox.org/virtualbox/5.1.2/Oracle_VM_VirtualBox_Extension_Pack-5.1.2-108956.vbox-extpack
```

instale:

```
VBoxManage extpack install Oracle_VM_VirtualBox_Extension_Pack-5.1.2-108956.vbox-extpack
```

Esse próximo passo é importante. É necessário criar um usuário para acessar o VirtualBox pelo PHP. Eu não tinha me ligado nisso e quando instalei o OpenStack usei o meu usuário como root em modo sudo (*sudo -i*) então não pude ver as máquinas virtuais criadas. Tive que fazer tudo denovo do início. Use o mesmo usuário aqui e na instalação do OpenStack. Para instalar o OpenStack ele precisará ter poderes de root. Eu usei o usuário padrão da instalação do Ubuntu (primeiro usuário).

Neste ponto o tutorial manda criar um usuário, mas eu vou usar o meu usuário padrão do Ubuntu.

Verifique a instalação do VirtualBox:

```
sudo /etc/init.d/vboxdrv status
```

Caso algo esteja errado, execute:

```
sudo /etc/init.d/vboxdrv setup
```

É necessário um servidor Apache e o PHP. Se já tiver, pule esta etapa e verifique a pasta de destino se confere com sua instalação (*/var/www/html/*). Não esqueça também de conferir se os demais pacotes do PHP estão instalados:

```
sudo apt-get install apache2 php5 php5-common php-soap php5-gd
```

Baixei o PHPVirtualBox. Minha versão é diferente da do tutorial (vá para seu diretório *home*).

```
wget
```

```
http://sourceforge.net/projects/phpvirtualbox/files/phpvirtualbox-5.0-5.zip
```

Descompacte o arquivo. Se não tiver o unzip, o apt-get resolve seu problema.

```
unzip phpvirtualbox*.zip
```

Copiar o PHPVirtualBox para o servidor Apache.

```
sudo mv phpvirtualbox-5.0-5 /var/www/html/phpvirtualbox
```

Criar o arquivo de configuração e configurar o usuário que irá controlar o VirtualBox e suas máquinas. Precisa ser o mesmo que irá instalar o OpenStack (com poderes sudo).

```
sudo cp /var/www/html/phpvirtualbox/config.php-example  
/var/www/html/phpvirtualbox/config.php
```

```
sudo vi /var/www/html/phpvirtualbox/config.php
```

Modifique as seguintes linhas:

```
[...]  
var $username = 'magno';  
var $password = 'minhasenha';  
[...]
```

O próximo passo é informar ao VirtualBox este usuário. Crie o seguinte arquivo:

```
sudo vi /etc/default/virtualbox
```

e coloque a seguinte linha nele (claro que você vai trocar meu nome pelo usuário que existe na sua instalação):

```
VBOXWEB_USER=magno
```

Reinicie o VirtualBox WebService:

```
sudo /etc/init.d/vboxweb-service start
```

agora é só navegar para o seu novo servidor:

`http://<SEU_IP>/phpvirtualbox/`

Usuário: admin

Senha: admin

Na minha instalação eu recebi o seguinte erro após logar no painel do PhpVirtualBox:

```
This version of phpVirtualBox (5.0-5) is incompatible with
VirtualBox 5.1.2. You probably need to download the latest
phpVirtualBox 5.1-x.
```

O caso é que coloquei a versão mais recente do VirtualBox e do PhpVirtualBox mas elas não estavam ainda equiparadas. Mas até agora não tive maiores problemas com isso.

Mas tudo isso é desnecessário se você for fera na linha de comando do VirtualBox. Eu não sou e preciso muito da interface gráfica do VirtualBox. O mais legal disso é que você poderá pegar a console de qualquer máquina virtualizada usando o Console Remoto do Windows apontando para o IP da máquina host (sim da host, não da máquina virtual). O VirtualBox te dirá qual máquina está escutando em que porta).

Instalando o Fuel:

A partir daqui, o procedimento foi igual ao descrito [no primeiro post da série](#). Copiei a imagem ISO para a pasta *iso* do pacote de scripts e executei dessa vez a versão para 8GB, pois agora estou ostentando recursos.

Precisei informar ao script de instalação que agora não há mais interface gráfica no VirtualBox. Isso é muito importante, pois o script não vai conseguir iniciar as VM corretamente. Para tanto, editei o arquivo **config.sh**:

Localize a linha contendo o código a seguir:

```
# Set to 1 to run VirtualBox in headless mode
headless=0
```

e troque para

headless=1

Após isso, executei a instalação.

```
./launch_8GB.sh
```

Fiquei impressionado com a melhora no tempo de instalação. Incrivelmente mais rápido. Após tudo instalado, o sistema me informou o mesmo modo de acesso de antes:

`http://10.20.0.2:8443`

Só tem um probleminha: essa rede não está exposta para minha rede local e não posso modificar nada do ambiente de rede do Fuel sem danificar nada. A interface que controla a rede 10.20.0.0 é *host only*. Logo, só posso acessar o painel do Fuel a partir da máquina host e não tenho interface gráfica para isso.

Deixei uns dois dias esse problema de molho na minha cabeça e concluí que a solução não estava no VirtualBox, mas no próprio Ubuntu (e é uma solução até bem simples, se você sabe para onde olhar).

Resolvi criar um redirecionamento de portas da interface de rede da máquina host (que dá para minha rede local) para a interface de rede virtual que controla a rede 10.20.0.0. Na verdade, só tive a ideia porque tenho conhecimentos de infraestrutura de redes, mas não tinha a menor ideia de como fazer isso, pois não domino muito bem o *iptables* do linux. Tá bom, eu confesso: não entendo uma vírgula disso.

Recorri ao pessoal do [Unix StackExchange](https://unix.stackexchange.com/) e a resposta veio rápido. Primeiro postei a configuração das interfaces de rede da máquina host:

```
root@AKRAB:~# ifconfig
lo          Link encap:Loopback Local
            inet end.: 127.0.0.1  Masc:255.0.0.0
            endereço inet6: ::1/128 Escopo:Máquina
            UP LOOPBACK RUNNING  MTU:65536  Métrica:1
            pacotes RX:19685 erros:0 descartados:0 excesso:0
quadro:0
            Pacotes TX:19685 erros:0 descartados:0 excesso:0
portadora:0
            colisões:0 txqueuelen:0
            RX bytes:7674590 (7.6 MB) TX bytes:7674590 (7.6 MB)
```

```
vboxnet0      Link  encap:Ethernet      Endereço de HW
0a:00:27:00:00:00
                inet end.: 10.20.0.1    Bcast:10.20.0.255
Masc:255.255.255.0
                endereço inet6: fe80::800:27ff:fe00:0/64 Escopo:Link
                UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
                pacotes RX:0 erros:0 descartados:0 excesso:0
quadro:0
                Pacotes TX:167 erros:0 descartados:0 excesso:0
portadora:0
                colisões:0 txqueuelen:1000
                RX bytes:0 (0.0 B) TX bytes:22260 (22.2 KB)

vboxnet1      Link  encap:Ethernet      Endereço de HW
0a:00:27:00:00:01
                inet end.: 172.16.0.254  Bcast:172.16.0.255
Masc:255.255.255.0
                endereço inet6: fe80::800:27ff:fe00:1/64 Escopo:Link
                UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
                pacotes RX:0 erros:0 descartados:0 excesso:0
quadro:0
                Pacotes TX:437 erros:0 descartados:0 excesso:0
portadora:0
                colisões:0 txqueuelen:1000
                RX bytes:0 (0.0 B) TX bytes:137886 (137.8 KB)

vboxnet2      Link  encap:Ethernet      Endereço de HW
0a:00:27:00:00:02
                inet end.: 172.16.1.1    Bcast:172.16.1.255
Masc:255.255.255.0
                endereço inet6: fe80::800:27ff:fe00:2/64 Escopo:Link
                UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
                pacotes RX:0 erros:0 descartados:0 excesso:0
quadro:0
                Pacotes TX:464 erros:0 descartados:0 excesso:0
portadora:0
                colisões:0 txqueuelen:1000
                RX bytes:0 (0.0 B) TX bytes:150336 (150.3 KB)

wlan0         Link  encap:Ethernet      Endereço de HW
00:13:46:94:18:c1
```

```
        inet end.: 192.168.25.25   Bcast:192.168.25.255
Masc:255.255.255.0
        endereço inet6: fe80::213:46ff:fe94:18c1/64
Escopo:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
        pacotes RX:2354945 erros:0 descartados:4 excesso:0
quadro:0
        Pacotes TX:1237088 erros:0 descartados:0 excesso:0
portadora:0
        colisões:0 txqueuelen:1000
        RX bytes:3455421823 (3.4 GB) TX bytes:103231994
(103.2 MB)
```

```
root@AKRAB:~#
```

O que me interessa é a interface exposta na minha rede local (*wlan0*) e a interface que responde pela rede 10.20.0.0 (*vboxnet0*). Então a dica para o redirecionamento de pacotes que recebi foi:

```
iptables -I FORWARD -d 10.20.0.2 -m comment --comment "Accept
to forward Fuel DashBoard traffic" -m tcp -p tcp --dport 8443
-j ACCEPT
```

```
iptables -I FORWARD -m comment --comment "Accept to forward
Fuel DashBoard return traffic" -s 10.20.0.2 -m tcp -p tcp --
sport 8443 -j ACCEPT
```

```
iptables -t nat -I PREROUTING -m tcp -p tcp --dport 8443 -m
comment --comment "redirect pkts to virtual machine" -j DNAT -
-to-destination 10.20.0.2:8443
```

```
iptables -t nat -I POSTROUTING -m comment --comment "NAT the
src ip" -d 10.20.0.2 -o vboxnet0 -j MASQUERADE
```

Isso ainda não resolve o problema. É necessário informar ao linux que eu quero ativar o *IP Forwarding*:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ainda não basta. Ao reiniciar o servidor tudo isso será perdido. É necessário editar o arquivo

```
/etc/sysctl.conf
```

e remover o comentário da linha

```
net.ipv4.ip_forward=1
```

Ainda, o *iptables* vai evaporar também ao reiniciar o servidor. Para tornar permanente as regras de redirecionamento, é necessário gravar as regras atuais para um arquivo

```
sudo iptables-save > /etc/iptables.conf
```

e então editar o seguinte arquivo:

```
/etc/rc.local
```

e colocar o seguinte conteúdo nele (antes do “exit 0”):

```
# Load iptables rules from this file
iptables-restore < /etc/iptables.conf
```

Pronto. Agora foi possível acessar o painel de controle do Fuel através do endereço

```
https://<IP_DO_SERVIDOR>:8443
```

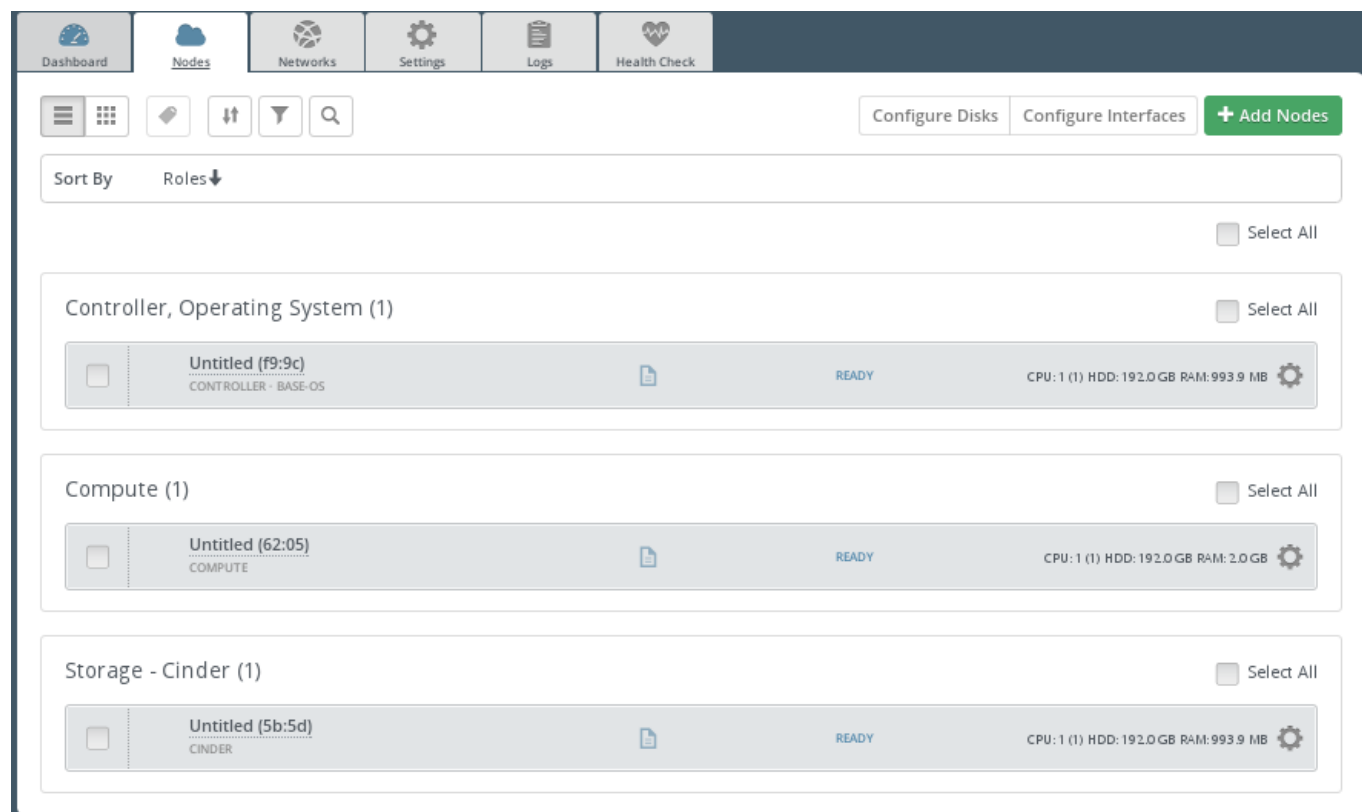
Ele irá redirecionar o acesso para a máquina virtual *master* no IP 10.20.0.2 na mesma porta.

Não sei se isso é algum problema da instalação do VirtualBox, mas o caso é que, ao reiniciar o servidor, minha interface *wlan0* sobe mas não conecta com a rede. Estou precisando executar um *ifdown* seguido de *ifup* em *wlan0* para ela conectar. Fica a dica se você tiver somente uma interface Wi-Fi e acontecer aí.

[No próximo post](#): e agora José?

OpenStack: Criação da Nuvem

Após [instalar o Mirantis Fuel em um VirtualBox no Windows](#), o próximo passo foi criar a minha nuvem. A minha primeira pergunta foi: o que fazer com as 3 máquinas virtuais que me foram entregues?



A resposta eu encontrei no arquivo **config.sh** (configuração mínima):

Uma configuração sem HA (Alta Disponibilidade) : 1 Controller e 1 Compute (a que sobra coloca como Controller se quiser).

Uma configuração sem HA com Cinder : 1 Controller, 1 Compute e 1 Cinder

Uma configuração HA é necessário no mínimo 3 Controllers e 1 Compute.

Encontrei nesse arquivo algumas dicas de hardware, visto que uma das máquinas

tinha configuração de memória diferente, mas todas possuem disco igual. A máquina com memória diferente seria para quem?

Um Controller requer no mínimo 1.5 G de RAM

Um Compute requer no mínimo 1 G de RAM ou as instâncias de VM não rodam.

Se tiver um Cinder dedicado, este precisará de 768M e o Ceph, 1 G.

Recomendo a leitura do arquivo **config.sh** para obter informações valiosas sobre configuração.

De posse destas informações e do vídeo do primeiro post eu pude fazer meu primeiro *deploy* do OpenStack. Tentei um Controller, um Compute e um Cinder, como no vídeo.

Demorou bastante. Primeiro ele instala um Ubuntu em todos os *slaves* e depois faz a instalação dos programas necessários para rodar o OpenStack. Na primeira tentativa tive um problema com um dos nós: deu um erro após a instalação do Ubuntu porque aparentemente o nó ficou *offline* durante a instalação. Para não ter uma instalação contaminada eu apaguei todas as VM e fiz tudo novamente desde o início.

Ao final do processo eu tinha meu ambiente OpenStack instalado. O link para acessar o Horizon foi entregue na aba "*Dashboard*" do painel de controle do Fuel (imagem acima).

Pude brincar um pouco e até instanciar uma máquina, mas algumas perguntas logo apareceram: e depois? este ambiente é o bastante para algo sério? todo o ambiente está encapsulado na máquina host que executa o VirtualBox. como acessar alguma coisa de fora desta máquina se nenhuma interface de rede está exposta na minha rede doméstica? Estas perguntas reforçaram minha teoria de que o Mirantis Fuel, embora seja uma ferramenta excelente para simplificar o processo de instalação do OpenStack e oferecer uma boa oportunidade para ver funcionando sem precisar de um parque computacional completo, é apenas um ambiente de teste.

Também oferece uma oportunidade de estudar a complexa configuração de rede do OpenStack. Quando for criar um ambiente mais sério acompanhando os manuais oficiais, isso será de grande valia para saber quantas interfaces de rede cada máquina terá, qual será o papel de cada uma e onde vão se conectar nos diversos roteadores que serão usados para montar as diferentes redes.

Outra observação: não é possível tirar máximo proveito do ambiente hospedando-o em uma máquina Windows. Partirei para o próximo passo, que será instalar o VirtualBox em um servidor Ubuntu e instalar tudo novamente, assim poderei ter mais recursos de hardware disponíveis (minha máquina não é lá essas coisas). Depois de ficar satisfeito, partirei para a instalação do Trove, que infelizmente não vem no Fuel. Precisarei fazer isso “na mão”.

Após tudo isso, usarei meus conhecimentos em Java para acessar a API do OpenStack e manipular os recursos da nuvem programaticamente, pois tenho planos para criar um Sistema Gerenciador de Workflows Científicos nativo para o OpenStack.

Estou bem empolgado para acessar o Trove e o Swift usando Java, mas até lá o caminho me parece longo ainda. Como dizem, *toda grande jornada começa com o primeiro passo*.

OpenStack : Primeiros passos

Semana passada resolvi me aventurar a entender o OpenStack. Tudo começou quando meu amigo Ricardo Campisano do Laboratório Interinstitucional de e-Astronomia me perguntou se eu conhecia. Bastou isso para ativar minha curiosidade.

Primeiro tentei instalar com o AutoPilot do Ubuntu, mas a infraestrutura exigida não é para amadores. Precisava de alguma coisa mais prática e rápida.

Achei o [Mirantis Fuel](#). É perfeito para quem está começando. Futuramente vou

dar uma olhada a versão do [DevStack](#), que possui a opção de instalar o OpenStack diretamente em uma máquina física.

Do site <http://docs.mirantis.com>:

You can install Fuel on VirtualBox and use that to deploy a Mirantis OpenStack environment for demonstration and evaluation purposes. Mirantis provides scripts that create and configure all the VMs required for a test environment, including the Master node and Slave nodes.

Baixei a versão 8.0 poucos dias antes de lançarem a 9.0. Trata-se de um arquivo ISO e um arquivo ZIP contendo alguns scripts bash. Não tenho um ambiente Ubuntu com virtualbox pronto, mas felizmente as [instruções de instalação](#) permitem usar o cygwin.

Copiei o arquivo ISO para a pasta “iso” que veio junto com os scripts e coloquei tudo no cygwin. Após isso, executei o comando de instalação. Existem 3 versões do script de execução: normal, 8GB e 16GB. Rodei o normal, que promete 3 slaves e um master. O master não é propriamente o OpenStack, mas o controle de *deploy* do Fuel.

Não entrei em detalhes sobre as outras opções. Quando a gente ganha um brinquedo de natal quer logo sair brincando, então digitei:

```
./launch.sh
```

Esperei um bom tempo. Minha máquina é um Core i5 com 8GB de RAM. Não ficou tão pesado quanto eu achei, mas o OpenStack ainda não estava instalado, só o Fuel (que, a propósito, é algo como um “criador de nuvens”, permitindo criar uma infraestrutura OpenStack de acordo com suas necessidades).

Resolvi me apoiar em algum vídeo do Youtube, que possui uma vasta coleção de vídeos sobre o assunto. Escolhi este, que apesar de estar desatualizado, serviu.

Como prometido, recebi 3 máquinas slave e uma master.

O usuário de todas elas é root,
Senha: r00tme (zero no lugar da letra "o")

Fui instruído a acessar o painel de controle do Fuel na máquina master através do

endereço

<https://10.20.0.2:8443/>



Usuário: admin

Senha: admin

Percebi que as máquinas *slave* são apenas “cascas vazias” com boot PXE. Acredito que se eu iniciar uma máquina física com boot PXE na mesma rede ela vá fazer parte do cluster. Vou testar isso ainda. Eu clonei um dos nós para testar e ele foi acrescentado ao cluster. Obviamente eu resetei o endereço MAC dele e a descrição, que é o próprio endereço MAC. Apesar da tentação de ter mais um *slave*, apaguei a nova máquina, afinal, tartaruga não sobe em árvore. Se decidiram por apenas 3, que seja.

Por falar em rede, a infraestrutura de rede criada me deixou um pouco confuso. Procurei por material na internet que explique como cada máquina está posicionada na rede, mas não tive muita paciência em me aprofundar nisso por enquanto. Quando for passar para o mundo físico eu vou enfrentar isso de frente. Basta saber que foram criadas 3 novas interfaces de rede no VirtualBox: 2 Host Only e 1 NAT e nenhuma delas expõe a rede interna para a sua rede doméstica. Fiquei tentado a mudar uma das redes no arquivo de configuração para coincidir com minha rede doméstica, mas ele criou um gateway com o mesmo endereço do meu Modem ADSL e acabou tirando minha máquina da internet. Resolvi que deveria instalar o produto exatamente como manda o manual e parei de experimentar coisas antes de conhecer tudo. Apaguei as VM e instalei tudo novamente.

Pelo pouco que entendi, uma rede é a rede administrativa, outra é a rede pública e outra é a rede interna usada para comunicação do cluster e boot PXE.

Encontrei [este site com boas informações](#) sobre a rede e requerimentos:

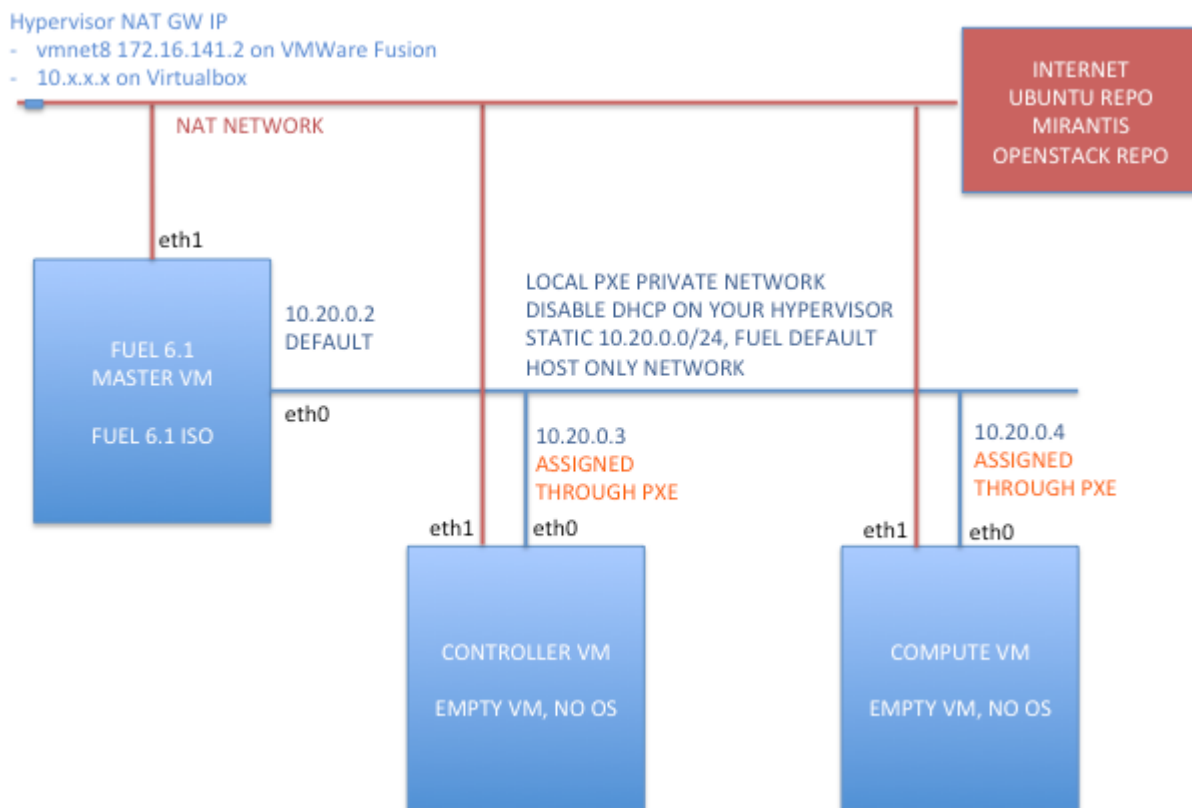
Hypervisor (Virtualbox, VMWare Fusion). This guide is based on VMWare Fusion on Mac (3 GHz i7, 16GB RAM), but the steps are similar for Virtualbox

Fuel ISO for Fuel Master VM, download from Mirantis Website (min 41 GB HDD), use 50GB VM Storage, 2 vCPU, 4GB RAM
2 network adapters: 1 for NAT, 1 for host to host/host-only
first boot on Fuel ISO

Node 1 -> for OpenStack Controller VM
50 GB space, 2vCPU, 2GB RAM
Empty VM with no OS, OS will be provisioned by Fuel Master, OpenStack role will be deployed by Fuel Master
2 network adapters: 1 for NAT, 1 for host to host/host-only
first boot through PXE on eth0

Node 2 -> for OpenStack Compute VM
50 GB space, 2vCPU, 2GB RAM
Empty VM with no OS, OS will be provisioned by Fuel Master, OpenStack role will be deployed by Fuel Master
2 network adapters: 1 for NAT, 1 for host to host/host-only
first boot through PXE on eth0

Connectivity from all 3 VM to Internet (Ubuntu Repo and Mirantis OpenStack Repo).



No [próximo post](#) vou continuar descrevendo a instalação de uma nuvem OpenStack.