Universidade dos Açores

Informática - Redes e Multimédia

Administração de Sistemas e de Redes

Serviço de redes para empresa local

Grupo:
Helder Correia - 20102556
Miguel Luís - 20122604

Professora:
Ibéria Medeiros
iberia@uac.pt

Projecto Final

6 de Junho de 2013

Conteúdo

1 | Solução

A solução encontrada para a empresa Imbcc, SA está representada no diagrama da figura 1.1.

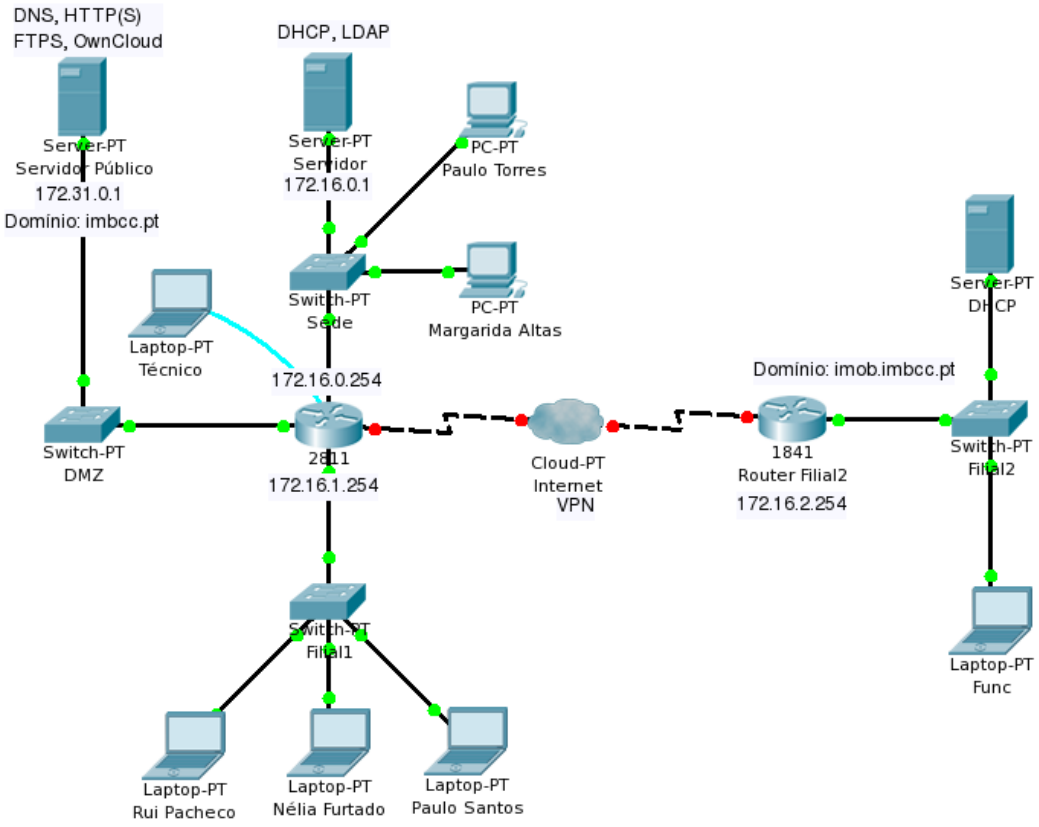

Figura 1.1: Topologia de rede para implementação da solução.

Para poupar em equipamentos, o router na Sede deve suportar 4 interfaces de rede separadas, nem que seja através de sub-interfaces. O router e dois servidores ficam no armário de equipamentos da Sede. Um servidor é privado, para uso interno enquanto o outro deverá ser acessível externamente, ambos em redes separadas. Uma terceira interface leva um cabo de rede ao piso inferior até a um switch que liga os computadores da Filial 1. Para ligação segura entre a Filial 2 e o resto da empresa no edifício da Sede será usada uma ligação VPN gerida de forma transparente pelo ISP local, que aproveita a linha de acesso à Internet.

Para além do DHCP, o servidor da Sede serve também o serviço de LDAP de forma centralizada e privada (interno). O servidor da rede pública (rede DMZ), contém os serviços de resolução de nomes (DNS), páginas web (HTTP e HTTPS), FTPS para upload dos ficheiros para a web, e é onde fica o ownCloud para que seja possível usar o serviço externamente.

O ownCloud deve ter ligação segura por TLS com o servidor LDAP para a autenticação dos utilizadores da empresa. Todos os computadores devem ter um servidor de SSH com conexões limitadas apenas à equipa técnica para acesso remoto.

1.1 DHCP

O servidor de DHCP na Sede serve 3 redes: a `sede`, a `dmz` e a `filial1`. Um DCHP relay fica activo no router e assim consegue estender o domínio até às outras redes. O servidor principal da DMZ podia receber o seu IP dinamicamente com reserva de atribuição de IP mas o melhor é não depender do outro servidor privado. Assim o servidor da DMZ consegue funcionar de forma independente e também facilita a gestão de serviços como o DNS.

1.2 DNS

O DNS suporta duas vistas. Uma interna e outra externa. A vista interna dá um nome a todos os computadores da empresa. Regra geral, na rede DMZ os servidores têm o nome `serv$`, onde `$` representa o número do último octeto do IP (e.g. 172.31.0.5 corresponde a `serv5`). Na Sede e na Filial 2, tendo ambas um servidor privado, têm o nome de `server`, i.e., `server.imbcc.pt` e `server.imob.imbcc.pt` respectivamente.

Para as máquinas que recebem o IP por DHCP, têm o nome `dhcp[rede]$`. O número opcional da rede é para distinguir as máquinas da Sede (com id 0) das máquinas da Filial 1 (com id 1) uma vez que pertencem ao mesmo domínio. Ou seja, sendo que as gamas de IPs são 60-99, o primeiro cliente DHCP da Sede, Filial 1 e Filial 2 são respectivamente `dhcp060`, `dhcp160` e `dhcp60.imob`.

Para fazer o subdomínio da Filial 2 (imob) optou-se não por criar uma zona dedicada mas sim usar o ficheiro de zona principal para facilitar a gestão. Assim cada máquina da Filial 2 tem o sufixo de `.imob` em frente a cada nome.

Cada nome adicional da mesma máquina é configurado com um registo CNAME na rede interna ou um registo A na rede DMZ. Assim, usando por exemplo a ferramenta `dig` para questionar o subdomínio `www` será retornado na resposta o CNAME para serv1 internamente, mas externamente retornará o IP (registo A) do próprio servidor. Na DMZ apenas se usa o CNAME se o serviço for mesmo análogo (como `dns` ser outro nome para `ns1`).

Para a tradução reversa de nomes (registos PTR), externamente o servidor da DMZ é visto como `ns1` principalmente, mas internamente o IP é traduzido para `serv1`.

A configuração de forwarders facilita a gestão da rede porque as máquinas apenas precisam do servidor DNS da DMZ. A vista *external* tem a opção *recursion* ligada apenas durante a fase de testes para que apenas seja pre-

ciso o servidor DNS da DMZ na máquina ligada externamente. Porém em produção essa opção deve ser desligada para não oferecer serviços de DNS gratuitamente ao público geral.

Optou-se por não implementar um servidor de DNS secundário para poupar em equipamento.

1.3 HTTP

Para os serviços web optou-se por usar o Apache 2, com configuração por *virtual hosts*. Para o site principal da empresa optou-se por preferir o subdomínio `www` portanto acessos ao endereço `http://imbcc.pt` são redireccionados para `http://www.imbcc.pt`. É também neste domínio principal que estão configuradas as homepages dos vendedores imobiliários acedendo por exemplo ao endereço `http://www.imbcc.pt/~nelia/` para aceder à página da Nélia Furtado.

Com a excepção do serviço ownCloud, há apenas mais um site HTTPS para ligação segura: `https://clientes.imbcc.pt`. Os acessos a este endereço com http serão redireccionados automaticamente para a versão segura. Todos os outros acessos a um https que não esteja configurado serão redireccioandos para a sua versão http.

1.4 Certificados

Os certificados são assinados por um CA *self-signed* pertencente à empresa. Esse CA é que assina todos os certificados que são precisos. Esta PKI foi criada usando o TinyCA. As passwords dos certificados (se existirem) é `imbcc`.

1.5 LDAP

Para a configuração do LDAP foi editado o ficheiro `/etc/openldap/slapd.d/cn=config/olcDatabase={2}bdb.ldif` para mudar o `olcSuffix` e `olcRootDN`, assim como o `olcRootPW` com uma password criada com `sldappasswd -s imbcc`. Também foi editado o ficheiro `olcDatabase={1}monitor.ldif` para actualizar o `base.dn`. O resultado está nos ficheiros de configuração entregues com o projecto.

Depois são adicionados as entradas ao directório usando o `ldapadd` e os ficheiros `.ldif` também entregues. A ordem é importante.

```
ldapadd -f [ficheiro].ldif -D [valor de olcRootDN] -w [password]
```

```
imbcc.ldif
users.ldif
groups.ldif
users/margarida.ldif
users/nelia.ldif
users/paulo.ldif
users/psantos.ldif
users/rui.ldif
groups/webdesign.ldif
groups/imob.ldif
```

Para a organização da árvore LDAP usou-se os *organizational units* para representar um tipo de objectos. Temos um para os utilizadores todos e outro para os grupos. Posteriormente pode ser criado mais um para os equipamentos por exemplo. Depois dentro do `ou=groups` temos os grupos do web designers e dos utilizadores da imobiliária. Cada grupo depois tem como membro os utilizadores adequados do `ou=users`. Este esquema pode ser visualizado com a figura 1.2.

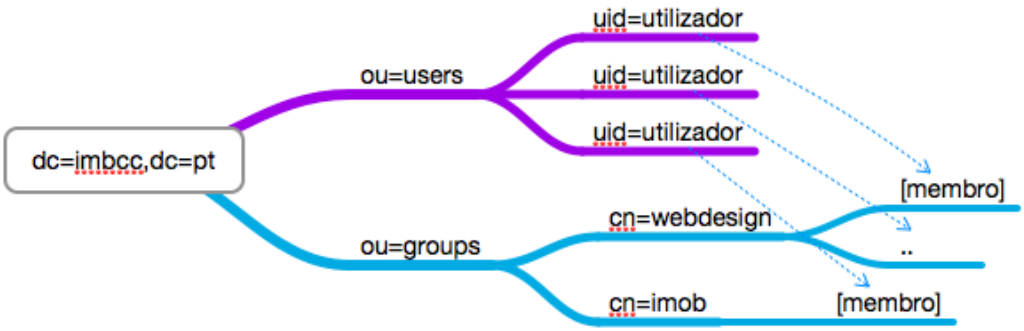

Figura 1.2: Esquema da árvore LDAP para esta empresa.

O nome mais descritivo dos grupos pode ser consultado no campo *description* que se encontra nos ficheiros `.ldif`. Para o `dn` dos utilizadores preferiu-se o `uid` (nome de utilizador) e o nome mais descritivo o campo `cn`. De notar que o utilizador `nelia` tem um campo `cn` adicional com o nome sem acento para que seja possível pesquisar por "Nelia" e não apenas "Nélia" através do ownCloud por exemplo.

1.6 ownCloud

O ownCloud pode ser acedido através do endereço `https://cloud.imbcc.pt` e devia ter uma ligação segura por TLS com o servidor LDAP para a autenticação dos utilizadores da empresa, mas isso não foi conseguido a tempo.

Para configurar a ligação por LDAP, após instalar a app adequada introduz-se as configurações da figura 1.3.

O servidor ldap está em `ldap.imbcc.pt` e usa-se o filtro `intOrgPerson` para os utilizadores e o filtro `groupOfNames` para os grupos. A seguir preenche-se a configuração avançada como na figura 1.4.

Nas configurações avançadas usa-se o campo `cn` para mostrar o nome completo da pessoa em vez do seu username e `description` para mostrar o nome humano do grupo em vez do nome máquina.

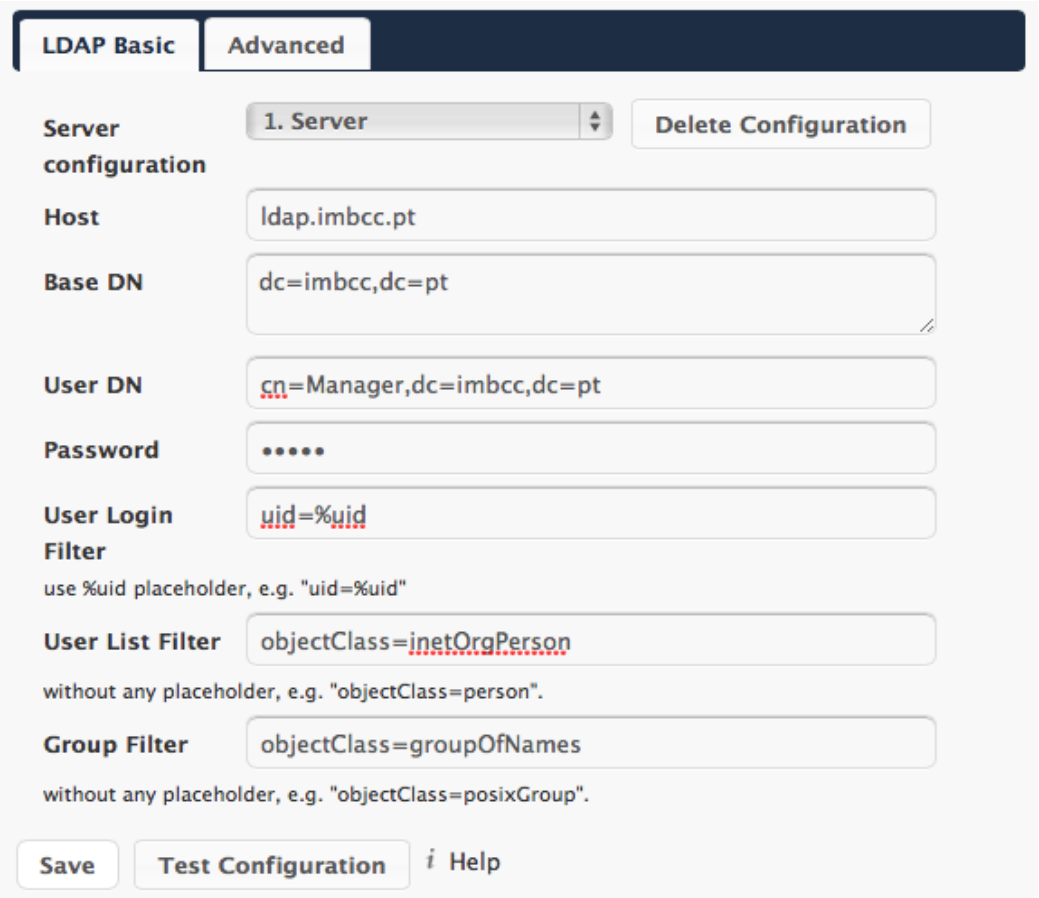

Figura 1.3: Configuração básica para o LDAP no ownCloud

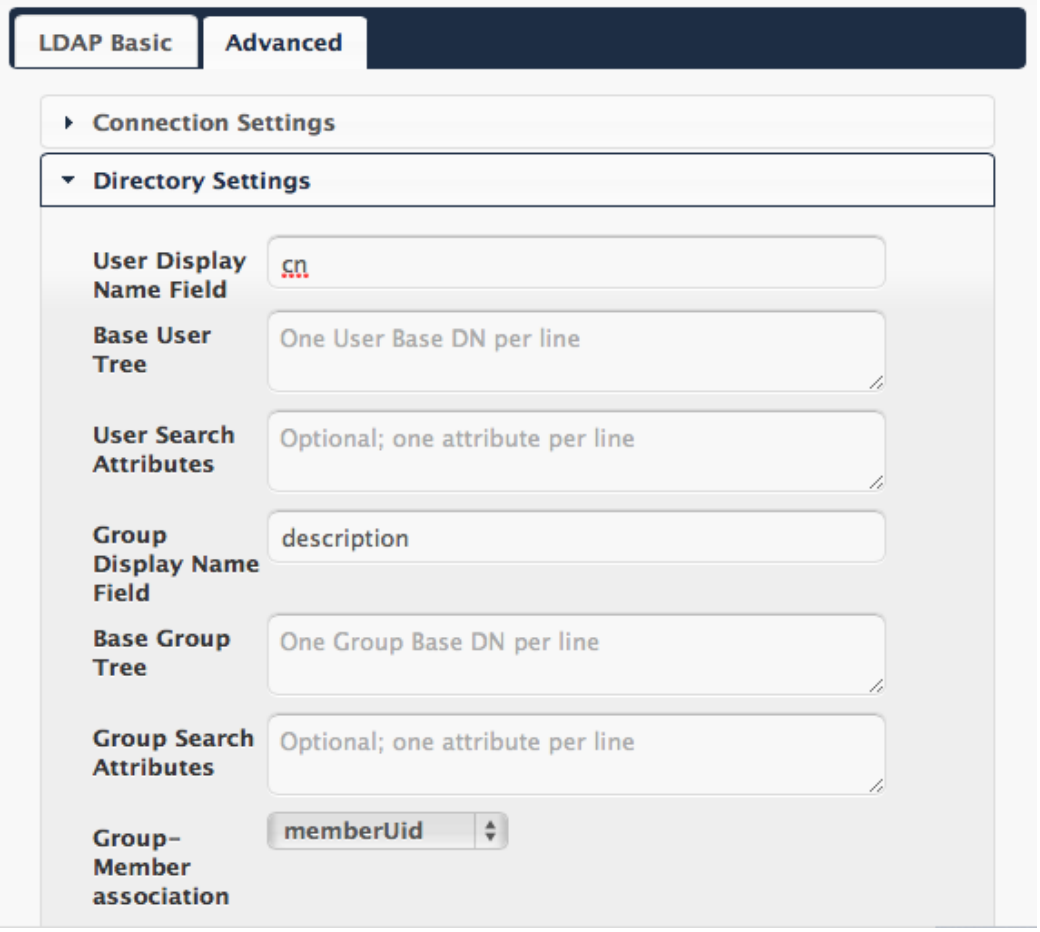

Figura 1.4: Configuração avançada para o LDAP no ownCloud

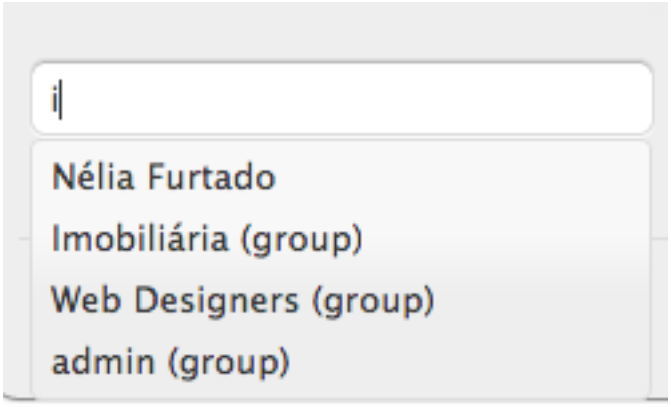

Figura 1.5: Resultado final quando se tenta partilhar um ficheiro.

O resultado final deve aparecer como na figura 1.5 ao tentar partilhar um ficheiro.

1.7 FTPS

Para se realizarem testes ao serviço de FTP seguro procedeu-se à utilização do programa lftp da seguinte forma:

```
lftp -d -u vagrant,vagrant -e "set ssl:verify-certificate no" ftp.imbcc.pt
```

2 | Simulação com Vagrant

Para testar a solução encontrada, foi usado o Vagrant [1] com VirtualBox, que permite automatizar e reproduzir uma infraestrutura virtual com facilidade. Como cada máquina gasta recursos da máquina *host* criou-se uma infraestrutura que pudesse simular a solução encontrada para a empresa, com o número mínimo de máquinas virtuais como se pode ver na figura 2.1.

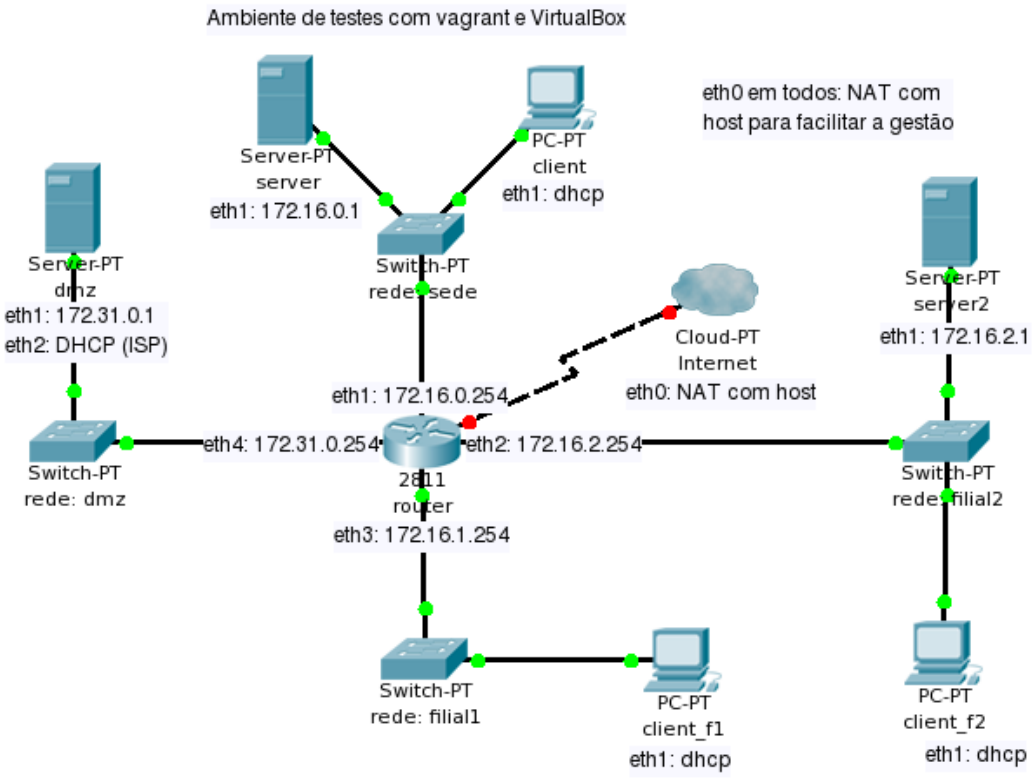

Figura 2.1: Topologia de rede para simulação com o VirtualBox.

O Vagrant[1] permite automatizar através de um simples ficheiro de configuração (`Vagrantfile`) a criação de várias máquinas virtuais através do VirtualBox. O VirtualBox em si é transparente e não é preciso correr o programa nem abrir nenhuma VM através da gráfica.

Escolheu-se apenas uma imagem base para todas as máquinas para poupar espaço em disco. O endereço está no ficheiro `Vagrantfile` entregue. A primeira vez que é corrido o comando `vagrant up` na pasta do projecto é feito o download dessa imagem para o disco e daí é feita uma cópia para cada máquina virtual. A box escolhida (imagem base) foi encontrada no site Vagrantbox.es [2], tem 416MB e as seguintes características: CentOS 6.4 i386 Minimal (VirtualBox Guest Additions 4.2.12, Chef 11.4.4, Puppet 3.1.1).

2.1 Particularidades

A ligação VPN entre a Sede e a Filial 2 pode ser simulada através de uma ligação directa. Em produção teriam apenas que ser adicionadas as rotas adequadas entre os dois lados da linha. A Internet é simulada através da interface `eth0` do router que está como *Host Only* (NAT). Para simular um IP dedicado para o servidor da DMZ é usada uma interface em *Bridge mode* (`eth2`) com o *host* que recebe o IP da nossa rede local.

O Vagrant por si só configura a primeira interface (`eth0`) em cada VM com NAT e redireccionamento para a porta 22 para que seja possível ligar individualmente por ssh a cada máquina o que facilita a gestão. Todas as outras são configuradas pelo ficheiro Vagrantfile. É portanto a interface `eth1` em modo *Internal network* (*intnet*) que simula as ligações locais usadas em produção. A excepção é o router que tem mais interfaces para poder interligar todas as redes.

O VirtualBox age como um switch entre interfaces *intnet* com o mesmo nome. Portanto para criar o switch da sede que se pode ver na figura 2.1, as interfaces `eth1` do servidor e do cliente estão configuradas para a *intnet* com

nome `sede`, assim como a mesma interface no router para funcionar como gateway para essa rede. É o mesmo para as restantes redes.

2.2 Provision com Chef

O Vagrant apenas gere as máquinas virtuais através do VirtualBox. Para configurar as máquinas é usado um método suportado pelo Vagrant que se chama *Provision*. Há várias formas de se fazer o provision, incluindo através de comandos shell. Porém preferiu-se o uso a ferramenta popular Chef [3].

Portanto quando o Vagrant activa uma máquina, depois da máquina iniciar é corrido o Chef para aplicar as configurações. Essas configurações devem ser escritas por forma a que seja possível fazer o provision tantas vezes quanto se queira para assegurar que a máquina está no estado que pretendemos.

2.3 Cookbooks

Para fazer o provision, o Chef usa *cookbooks* com uma ou mais receitas (*recipes*). A estrutura é simples.

Quando no Vagrantfile aparece `chef.add_recipe "dns"`, o ficheiro de configuração está em `cookbooks/dns/recipes/default.rb`. Se a *recipe* for do tipo `iptables::server` então em vez de `default.rb` é `server.rb`.

Quando uma receita usa um ficheiro (e.g. `cookbook_file`), ele encontra-se em `files/centos/[ficheiro]` dentro do cookbook em questão. A parte do "centos" pode ser substituida por outra plataforma ou pelo valor "default" para ser usado em todos os sistemas.

Este projecto foi feito a pensar no CentOS e não são fornecidas configurações para outras plataformas.

3 | Guia

É muito fácil correr este ambiente de simulação uma vez que está todo automatizado. Primeiro é preciso ter o Vagrant e o VirtualBox.

Depois de instalado o Vagrant na máquina *host*, através da linha de comandos na pasta que contém o ficheiro `Vagrantfile` corre-se o comando:

```
$ vagrant up
```

Este processo, após fazer o download da box base, leva no mínimo 30 minutos para clonar essa box, iniciar as 7 máquinas, configurar as interfaces, redireccionamento de portas, etc, e aplicar as configurações (provision) que envolve fazer download dos pacotes necessários (e.g. named, http).

É também possível fazer up apenas a uma máquina:

```
$ vagrant up dmz
```

Ou a todas as máquinas clientes usando uma expressão regular:

```
$ vagrant up /client*/
```

Nota: Os nomes das máquinas usadas pelo vagrant estão representados na figura 2.1.

Depois do primeiro `up` pode-se saltar o provision para ser mais rápido:

```
$ vagrant up server --no-provision
```

Só faz sentido usar o comando `up` quando uma máquina não está a correr. Para reiniciar usa-se o `reload` ou para encerrar usa-se o `halt`.

Também existe o `suspend` e o `resume` para colocar em estado de suspensão. Assim poupa-se RAM mas não espaço em disco.

No final pode-se destruir tudo com (máquinas criadas e box base):

```
$ vagrant destroy -f
$ vagrant box remove centos virtualbox
```

3.1 Usar o servidor DNS externamente

Durante o provision o IP público/externo atribuído à DMZ é actualizado automaticamente no servidor DNS para que seja possível acedê-lo fora do ambiente privado do VirtualBox.

É fornecido um script que verifica e actualiza o IP no DNS caso este tenha mudado. Em todo o caso o IP actual é impresso no ecrã e pode ser usado para actualizar na lista de nameservers da máquina host para que se possa utilizar a resolução de nomes da nossa rede virtualizada.

Portanto para saber o IP da dmz:

```
$ ./dns.sh
Already up to date for IP 192.168.1.131!
```

No entanto se a máquina já tiver sido iniciada e esse IP mudou por algum motivo (e.g. deslocamento para outra rede), também dá para reiniciar a interface para que seja renovado o IP atribuido e actualizar o servidor DNS:

```
$ ./dns.sh -r
Determining IP information for eth2... done.
Recovering from backup... done.
Updating files... done.
Stopping named: .[  OK  ]
Starting named: [  OK  ]
Your nameserver is at 192.168.0.105
```

O script que faz esse update pode ser consultado em `cookbooks/dns/files/centos/dns-external.sh`.

Durante o update os ficheiros originais são guardados num backup (ficheiro terminado em ~). Esse ficheiro original depois é recuperado para fazer novo update simplesmente substituindo as referências ao IP interno pelo novo IP externo.

3.2 Acesso SSH às máquinas

Cada máquina pode ser acedida individualmente através do comando `vagrant ssh [vm]`. Esse acesso é criado automaticamente pelo Vagrant através da interface `eth0` em NAT com port forwarding que é gerido automaticamente.

Esse acesso facilita a gestão das máquinas mas para simular o que aconteceria em produção acede-se à dmz externamente e daí novo ssh para qualquer máquina interna. Uma vez que o Vagrant também cria em cada máquina automaticamente o utilizador `vagrant` com poderes administrativos, aproveitou-se esse utilizador na configuração dos servidores `sshd` para ser o único com acesso.

Para acesso root ao servidor interno:

```
$ ssh vagrant@imbcc.pt
vagrant@imbcc.pt's password:
[vagrant@dmz ~]$ ssh vagrant@server
vagrant@server's password:
[vagrant@server ~]# su -
Password:
[root@server ~]#
```

A password do utilizador `vagrant` e `root` é `vagrant`.

3.3 Pasta partilhada

O Vagrant configura em cada VM automaticamente uma pasta `/vagrant` que aponta para a pasta no host onde está o ficheiro `Vagrantfile`. Isso pode ser útil para transferir ficheiros de um lado para o outro e é o que usamos para exportar as configurações das máquinas.

3.4 Ordem de iniciação

A ordem pela qual se iniciam as máquinas tem importância uma vez que há dependências entre elas. A máquina `dmz` tem ligação própria à Internet e portanto pode ser usada independentemente, excepto pelo ownCloud que precisa do LDAP do `server`.

Todas as outras máquinas precisam do `router` para poderem aceder à Internet e da `dmz` para a resolução de nomes, incluindo o próprio `router`.

3.5 Problemas

Se uma máquina não conseguir aceder à Internet irá falhar na fase do *provision*. Se a máquina já tiver sido construída pode-se saltar essa fase com a opção `--no-provision` como visto no início do capítulo 3 Guia.

Também por vezes pode acontecer uma máquina não responder quando se encerra ou inicia. Pode aparecer um erro do tipo:

```
stderr: VBoxManage: error: The object is not ready
```

Nesses casos é preciso insistir.

Para erros durante a fase do provision pode-se fazer novo provision sem reiniciar a máquina:

```
$ vagrant provision server
```

4 | O que falta

Faltou fazer ligação segura por TLS entre o ownCloud e o servidor LDAP. Também faltou usar o LDAP com o FTPS para autorizar o grupo da imobiliária nos seus *userdirs* e os web designers à pasta `/var/www` e certificar que ele funciona bem no geral.

Referências

[1] Vagrant, *Development environments made easy.*, disponível em `http://www.vagrantup.com`

[2] Vagrantbox.es, *A list of boxes for Vagrant*, disponível em `http://www.vagrantbox.es`

[3] Chef, *infrastructure as code*, disponível em `http://www.opscode.com/chef/`