



Kaline Marculino.
Obede Oliveira.
Odin Miguel.

Sumário

- 1.** Resumo;
- 2.** Introdução;
- 3.** Objetivos;
- 4.** Desenvolvimento;
 - 0.1.** Sistema de Virtualização;
 - 0.2.** S.O e instalação do S.O;
 - 0.3.** Automação e Atualização(Shell Script);
 - 0.4.** Serviço SSH;
 - 0.5.** Filtro de pacotes iptables);
 - 0.6.** Gerência de Permissão(sudo);
 - 0.7.** Implantação do P&PII;
- 5.** Dificuldades e Melhorias;
- 6.** Conclusão;
- 7.** Referências;

1. Resumo

Vai ser feita uma abordagem técnica sobre a implantação e implementação de servidores, virtualização na matéria de implantação e administração de serviços web, visto como podemos chegar a ser bons profissionais em administração de servidores em ambiente de virtualização, com um foco maior na prática, mas sempre não deixando de lado a teoria para entendermos todos os conceitos e termos, para ao final ser implementado no projeto na matéria de projeto e prática II.

2. Introdução

Relatório das aulas práticas, onde visa mostrar conceitos teóricos e práticos aplicados no âmbito acadêmico, buscando a exploração do desenvolvimento realizado, desde a sala de aula até a prática na implementação dos conceitos apresentados em sala de aula na implementação prática no projeto, com a introdução a sistemas de virtualização com a instalação do sistema operacional sobre a virtualização em uma ambiente de produção real, com a automatização de atualização do sistema, incluindo serviços SSH(Permite a você acessar virtualmente o servidor), com às políticas de filtro de pacotes iptables) tendo uma gerência de permissão, visto todo esse âmbito, implementando todo o conteúdo ministrado na disciplina de implantação e administração de serviços web na disciplina de projeto e prática II.

3. Objetivos

Realizar a implementação de alocar a aplicação da disciplina de projeto e prática II, para deixar o serviço disponível para visualização de tal. Mostrando o aprendizado adquirido na disciplina de projeto e prática II, com todas às especificações.

4. Desenvolvimento

0.1. Sistema de Virtualização

A virtualização permite que um servidor físico rode diversas aplicações em diferentes sistemas operacionais, otimizando a utilização dos recursos de processamento e memória. Atuando como peça-chave no modelo de computação em nuvem (cloud computing), está, dependendo do funcionamento de um conjunto de datacenters dinâmicos que para funcionar sob demanda precisam da virtualização como tecnologia essencial. Tendo como alguns benefícios o melhor aproveitamento da infraestrutura existente, um parque de máquinas menor, o gerenciamento centralizado, implementação mais rápida, uso de sistemas legados(sistemas antigos), segurança e confiabilidade, migração e ampliação mais fácil...

0.2. S.O e instalação do S.O

Passo 1: Inicializar o computador usando o disco de instalação, quando aparecer a primeira tela, selecionar a opção “install”.

Passo 2: Em seguida selecionar a linguagem que será usada no processo de instalação e clique no botão “Continue”.

Passo 3: Caso apareça uma mensagem informando que a tradução na linguagem que você escolheu está incompleta, clique em “Sim” e confirme clicando no botão “Continuar”.

Passo 4: Em seguida, selecione a sua localidade para começar a configurar o fuso horário do seu PC.

Passo 5: Na tela de configuração do teclado, escolher o mapa de teclado a ser usado, no nosso caso, Português Brasileiro.

Passo 6: Na tela de configurar rede, digite o nome que será usado pela máquina.

Passo 7: Em configurar usuários e senhas, digite uma senha para o usuário root e repita ela no campo abaixo, Lembre-se de criar uma senha forte, que não seja muito óbvia e que contenha no mínimo oito caracteres, com números e letras.

Passo 8: Na tela de configurar usuários e senhas, digite o nome do usuário que usará o sistema.

Passo 9: Na tela seguinte ainda em configurar usuários e senhas, crie a senha de usuário, seguindo os mesmos critérios da criada para o root.

Passo 10: Criado o usuário, é hora de escolher o fuso horário de sua localidade;

Passo 11: Na hora de particionar, selecione a opção “Assistido – usar o disco inteiro” e clique no botão “Continuar”;

Passo 12: A seguir, selecione o disco onde será instalado o sistema;

Passo 13: Agora clique na primeira opção e depois no botão “Continuar”;

Passo 14: Finalmente, clique em “Finalizar o particionamento e escrever as mudanças no disco” e em seguida, no botão “Continuar”;

Passo 15: Na próxima tela, apenas clique na opção “Sim” e no botão “Continuar”;

Passo 16: Na tela seguinte, se quiser adicionar mais um disco de instalação, marque a opção “Sim”. Caso contrário, clique em “Não”. Por fim, clique no botão “Continuar”;

Passo 17: Se quiser baixar pacotes diretamente da Internet, marque a opção “Sim”. Do contrário, clique em “Não”. Depois, clique no botão “Continuar”;

Passo 18: Confirme ou não o fornecimento de informações anônimas sobre os pacotes mais utilizados e então clique no botão “Continuar”;

Passo 19: Nesta etapa, selecione os servidores e ambientes gráficos que você quer usar, e clique no botão “Continuar”;

Passo 20: Na tela de instalação do GRUB, marque a opção “Sim” e clique no botão “Continuar”;

Passo 21: Na próxima tela, selecione o disco onde ficará o GRUB;

Passo 22: Em poucos segundos, você será informado que a instalação terminou. Clique no botão “Continuar” e aguarde o sistema ser reiniciado.

É instalado diretamente no servidor físico o VMWARE ESXI, permitindo que ele seja dividido em vários servidores lógicos chamados de máquinas virtuais. Os clientes

podem usar o ESXI com o VSphere Hypervisor gratuito ou como parte da edição paga do VSphere. Foi usado o sistema operacional debian 8 para S.O 32 ou 64 para virtualização, dado o nome do usuário para a máquina e uma senha padrão para o acesso ao sistema de virtualização do VSphere, nas especificações tendo uma interface, com 8GB, adicionar um cd/dvd.

0.3. Automação e Atualização(Shell Script)

Para ter um ganho no tempo e um melhor desempenho do servidor, sem necessariamente precisar para o sistema para sua atualização, onde é feita em um determinado horário definido pelo administrador do sistema, para não “atrapalhar” no serviço. Assim automatizando para um melhor aproveitamento e evitar o desgaste de fazer essa atualização periodicamente.

Primeiramente é criado um diretório no sistema operacional já aberto e logado(Debian 8), depois já dentro dele, cria o arquivo de automação, onde é colocado nele o script, primeiramente para atualizar o sistema, depois atualizando os componentes do hardware e do software

Script

```
# !/bin/bash
# script de atualização do S.O
# Obede Oliveira - obede.silva3@gmail.com
# 28/05/2017
```

```
apt-get update >> /etc/crontab
apt-get upgrade -y
apt-get dist-upgrade
```

Dado este script, no upgrade, quando têm a mensagem de continuar, ele automaticamente vai aceitar a continuação, clicando automaticamente em y(yes).

O outro script é o de ativar o script de automação, também automatizando a hora de ser atualizado o sistema para não atrapalhar o ambiente de produção.

Primeiramente entrara em /etc para depois abrir com o nano ou o editor que preferir o script crontab, já disponibilizado por padrão pelo sistema operacional(Debian 8), adicionando a tal, o horário de atualização do sistema de primeira, depois o usuário que poderá dar está permissão e por último o caminho do script de atualização. Como mostrado logo abaixo:

```
# cd etc
/etc# nano crontab
# agora veremos o script adicionado
```

m(minuto)	h(hora)	dom	mon	dow	user	command
00	3	*	*	*	root	/root/home/diretório onde está o script/script

No caso, para a atualização ser feita às 3 da manhã, com a permissão ao root ou seja o administrador do sistema, dando o caminho do script para atualização como já falado.

0.4. Serviço SSH

O SSH é um protocolo que permite a você acessar virtualmente o servidor como se você estivesse em um terminal (no prompt do DOS, por exemplo). No caso, usamos para autenticação do acesso do usuário, tendo as chaves de acesso, uma chave pública e outra privada, primeiramente usamos o programa Putty key para gerar as chaves, enviando para a pasta do tal, tendo as chaves para poder gerá las, feito isso, logamos, criando um usuário para entrar no sistema, quando entrar no sistema, criamos um diretório para às chaves, em seguida:

```
# ssh-keygen -f nome da key(chave do usuário) -c [email]
# senha: ****
```

usamos o win SCP(Cliente SFTP e FTP, que permite acessar, transferir e manipular arquivos remotamente).

cp [chave]_key* /root/home/nome do diretório para onde a chave estará indo (copiamos a chave para o diretório que queremos manipular).

chown [nome do novo usuário] /caminho para aplicá lo (Para dar permissão ao usuário que foi criado para acessar o sistema com a permissão de superusuário).

```
# cd /etc/ssh/
```

nano sshd_config (entramos no script de configuração do ssh, para alterar para dar o restante da permissão que falta para o usuário).

alteramos o %h= para definir o diretório home do usuário, o PasswordAuthenticion, e mudando de yes para no a permiteLogin(permissão de login) para no.

Feito isso, saímos do script e vamos para /home/diretório do usuário, criando um diretório para o ssh e depois copiando a chave

```
# cp [chave]_key.pub  
# cp /home/diretorio do usuário/ [chave]_key.pub .ssh/authorized.key
```

Todas chaves públicas, onde cópia às chaves para o ssh de autorização, em seguida mudando o usuário para ter a permissão.

```
# chown usuário.novo usuário /home/usuario/ -k
```

0.5. Filtro de pacotes(iptables)

Objetivo de adicionar restrições de acesso, permitindo a criação de regras de firewall e NATs(iptables), para estas foram usadas chains de restrições para conexões NEW(novas), ESTABLISHED(estabelecida), RELATED(inicia uma nova conexão para uma já existente). Política usada:

```
# iptables -A INPUT -m state --state NEW, ESTABLISHED,RELATED -j ACCEPT
```

Seguido desta política, é criado um script para primeiramente salvar e restaurar qualquer alteração nas regras de cheins e iptables, está, sendo relatada logo abaixo: Baseado nestes comandos:

```
iptables-save > /etc/regras-iptables  
iptables-restore > /etc/regras-iptables
```

Onde é direcionado para o diretório /etc e para a pasta regras-iptables. Feito isso, é adicionada às regras de iptables, primeiramente, é liberada a porta do http(porta:80) para saída do serviço:

```
# iptables -I OUTPUT -p tcp --sport 80 -j ACCEPT
```

Depois foi adicionado a regra para o ssh(porta:22) tanto para entrada, quanto para saída do serviço:

```
# iptables -A INPUT -p tcp --dport 22 -j ACCEPT  
# iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Seguindo, por fim foi feita a liberação da regra localmente no servidor, com o endereço da máquina virtualizada no caso:

```
# iptables -A INPUT -d 127.0.0.1(localhost) -j ACCEPT  
# iptables -A OUTPUT -s 127.0.0.1 -j ACCEPT
```

Por fim, para uma segurança e confiabilidade de todas as ativação do firewall, ativamos o firewall:

nano firewall

Conferimos se está tudo certo, depois deixamos executável:

chmod +x firewall.sh

./firewall.sh.start

Por fim foi ligado(startado) o serviço de firewall.

0.6 . Gerenciamento de permissão (sudo)

Configuramos o servidor de acesso remoto SSH para realizar autenticação de usuários unicamente através do método publickey. Era obrigatório adicionar um usuário para o administrador [nome], no servidor no qual administramos.

Usuário: ramon

Chave pública de acesso: que foi enviada em anexo por e-mail.

Onde no sistema de virtualização foi criado um diretório e dentro uma pasta para ser alocado essa chave e ser feita a configuração para permitir o acesso de um novo administrador e como já relatado no ponto(0.4) para o acesso do administrador local.

0.7. Implantação do P&PII

Como objetivo desta atividade, foi ter o projeto alocado no servidor, disponibilizado na web. Serão instalados alguns requisitos para montagem do ambiente de desenvolvimento e trabalho com a framework utilizada, no caso CakePHP, primeiramente o github(plataforma de hospedagem de código para controle de versão e colaboração), para poder ser feito o upload do projeto, seguindo os passos:

Primeiramente atualizando o sistema:

apt-get update

Depois baixando o apache2 para trabalhar com o serviço para web:

apt-get install apache2 php5

Seguindo, é instalado o php na versão 5, mas sempre bom verificar se já têm instalado por padrão na máquina na qual trabalha:

apt-get install php5-intl

Habilitando o módulo rewrite do apache:

a2enmod rewrite

Restartando o apache2:

service apache2 restart

Instalação da base de dados, no caso o mysql:

apt-get install mysql-server php5-mysql

Instalação do curl, usado para envio de documentos ou arquivos para o servidor:

apt-get install curl

Instalação do composer para instalar o composer.phar, onde será instalado no projeto:

curl -s https://getcomposer.org/installer | php

Instalação do git para baixar o projeto:

apt-get install git

cd /var/www/

Já dentro da pasta mostrada acima, é feito clone do projeto para o s.o.:

git clone <https://github.com/odinAsgardian/biblioteca>

#cd biblioteca/

Dentro da pasta do projeto, dar o composer install para vir o restante das pastas como /vendor, arquivos, app.php para configuração do banco de dados:

php composer.phar install

cd ..

Instalação da versão do desenvolvimento:

chown www-data.www-data biblioteca/ -R

É dado o caminho do projeto até a pasta webroot no DocumentRoot para o apache2 reconhecer onde está o projeto:

nano /etc/apache2/sites-available/000-default.conf


```
192.168.25.33 - PuTTY

<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    #DocumentRoot /var/www/html
    DocumentRoot /var/www/biblioteca/webroot

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
/etc/apache2/sites-available/000-default.conf (END)
```

Para terminar a configuração é feito alguns procedimentos para aceitar as requisições do apache2 ao projeto:

nano /etc/apache2/apache2.conf

```
192.168.25.33 - PuTTY

LogLevel warn

# Include module configuration:
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf

# Include list of ports to listen on
Include ports.conf

# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    #AllowOverride None
    #AllowOverride All
    #Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Restartando o apache2 para ser reconhecida as alterações:

service apache2 restart

Por fim é feita a exportação do banco de dados do projeto para o mysql e configurado o arquivo da base de dados em config/app.php, colocando o username, password e nome da base de dados criada no mysql;

5. Dificuldades e Melhorias

Ao início foi observar o mundo do datacenter, entender toda sua estrutura em conjunto com isso a virtualização, teve uma dificuldade em sua compreensão ao início, mas logo foi resolvida com muita prática e estudo, quando começamos a prática de fato, dentro do sistema de virtualização em um máquina virtual(debian 8), ao início um pouco de dificuldade para entender sua estruturação e saber encontrar cada coisa, mas como amamos GNU/Linux, logo foi suprida está pequena dificuldade, depois foi mais questão de prática que realmente dificuldade, saber achar como fazer o que não sabíamos fazer, buscando fontes e mais fontes de conteúdo, como na hora da criação dos scripts de automação, buscando a estruturação de tal, para ter uma melhor melhoria e aperfeiçoar, dado que nossa aplicação ao início está em um datacenter de pequeno porte e não suportaria uma grande gama de acessos simultâneos, como a aplicação tende a crescer e ter muitos acessos, observamos em fazer restrição da quantidade de acesso, limitando uma gama de 100 usuários logados simultaneamente, além de dar uma maior segurança com outras restrições, regras e políticas de acesso, melhorar os scripts de automação do sistema, dando uma maior redundância ao s.o(sistema operacional) e futuramente, dada a escalabilidade do projeto, migrar para um servidor de grande porte.

6. Conclusão

Visto todo o decorrer das atividades desde seu início, foi de grande satisfação partilhar de um semestre de grande aprendizado e enriquecimento como profissionais da área, mas também, como pessoas com o trabalho em equipe, onde foi de suma importância para uma maior interatividade e agilidade da realização das tarefas. Por fim cada realização de uma tarefa era a confirmação de que o conteúdo ministrado e ensinado vai ser para o resto da vida, como profissionais de T.I, para entrar e competir por um espaço no mercado de trabalho.

7. Referências

https://www.centos.org/docs/5/html/Cluster_Logical_Volume_Manager/
<http://www.guiafoca.org/cgs/guia/iniciante/ch-perm.html>
<http://www.guiafoca.org/cgs/guia/iniciante/ch-perm.html#s-perm-chown>
<http://www.guiafoca.org/cgs/guia/iniciante/ch-perm.html#s-perm-chgrp>
<https://www.cert.br/docs/seg-adm-redes/seg-adm-redes.html>
<https://www.cert.br/docs/seg-adm-redes/seg-adm-redes.html>
<https://book.cakephp.org/2.0/en/installation/url-rewriting.html>
<https://concani3.wordpress.com/2011/05/09/configurando-o-apache2-no-debian/>
<http://labcisico.blogspot.com.br/2015/05/configuracao-do-servidor-web-apache-no-ht>
[ml](http://labcisico.blogspot.com.br/2015/05/configuracao-do-servidor-web-apache-no-ht)
<https://www.digitalocean.com/community/tutorials/como-configurar-apache-virtual-hosts-no-ubuntu-14-04-lts-pt>
<https://www.vivaolinux.com.br/topico/UbuntuBR/erro-de-superblock>
<http://www.balochini.com.br/como-reparar-sistemas-linux-com-fsck-paulo-d/>
<https://www.vivaolinux.com.br/artigo/Checando-seu-sistema-de-arquivos-com-fsck>
<https://www.vivaolinux.com.br/topico/Linux-Avancado/Sistema-de-arquivo-ext4-com-errosajudem>
<https://www.vivaolinux.com.br/artigo/Checando-seu-sistema-de-arquivos-com-fsck/>
<https://www.vivaolinux.com.br/topico/Servidores-Linux-para-iniciantes/superblock>
<https://www.vivaolinux.com.br/artigo/Montando-regras-iptables>
<http://sysadmin.linuxdescomplicado.com.br/2016/05/10-regras-do-firewall-iptables-que-todo-sysadmin-linux-deve-conhecer/>
<http://www.hardware.com.br/livros/linux-redes/resumo-das-regras-iptables.html>
<https://www.vivaolinux.com.br/dica/O-que-fazer-depois-de-instalar-o-Debian-8>
[https://www.vivaolinux.com.br/dica/Instalando-e-configurando-servidor-SSH-\(Ubuntu\)](https://www.vivaolinux.com.br/dica/Instalando-e-configurando-servidor-SSH-(Ubuntu))