Manual de Instalação de Servidores LAMP para Hospedagem de Moodle (Badiu)

Paulo Angelo Alves Resende

Abril de 2017

Sumário

1 Instruções para uso deste guia **2**

2 Comentários e sugestões sobre o roteiro usado pela Badiu **2**

3 Configurações gerais **2**

4 Configurar uso de memória do SO **3**

5 Configurar análise de *logs* automatizada **3**

6 Redirecionar mensagens de e-mail para conta no GMail **3**

7 Rever regras de FW **4**

8 Configurar Fail2Ban **5**

9 Configurações de segurança no PHP **5**

10 Configurações de segurança no Apache **5**

11 Tuning PHP **6**

12 Configurar Apache para ambiente Moodle **6**

13 Tuning Apache **7**

14 Instalar e configurar sumarizador de logs para Apache **7**

15 Tuning MySQL **8**

16 Instalar e configurar sumarizador de *logs* para MySQL **9**

17 Referências **10**

1 Instruções para uso deste guia

1. Execute antes os passos definidos pelo manual criado pelo Lino.
2. Execute todos os comandos abaixo como usuário `root`.
3. Este documento é feito na linguagem de marcação *MarkDown*, que permite, além da fácil leitura do TXT, a geração de PDF, HTML, etc. Para saber como exportar para outros formatos, veja a ferramenta `pandoc` http://pandoc.org. Como exemplo, digite o comando abaixo para gerar um PDF.

   ```
   $ pandoc -V lang=brazilian --toc roteiro.md -o roteiro.pdf
   ```

2 Comentários e sugestões sobre o roteiro usado pela Badiu

1. Não vejo muitas vantagens no uso do repositório DotDeb para os pacotes PHP. O Debian Jessie já tem uma versão atualizada do PHP que é satisfatória.
2. Configure `"vm.swappiness = 10"` apenas quando se tem muita memória (no caso, mais de 64GB). Não é necessário em outro caso, o Linux já faz uma boa utilização da memória *swap* com a configuração padrão.
3. Outra forma de configurar o fuso horário é escolher `América` e depois `Sao_Paulo` após executar o comando abaixo.

   ```
   # dpkg-reconfigure tzdata
   ```

4. O módulo `userdir` do Apache é útil para ambiente de testes, porém não recomendado para ambientes de produção.

3 Configurações gerais

1. Instalar aplicações úteis. A aplicação tmux é um emulador de terminais muito bom, permite abrir várias abas além de não deixar cair a sessão SSH.

   ```
   # apt-get install vim tmux less
   ```

2. Instalar `ntpdate`, para sincronização de horário.

   ```
   # apt-get install ntpdate
   ```

3. Atualizar horário.

   ```
   # /usr/sbin/ntpdate-debian
   ```

4. Configurar para executar atualização de horário após reboot. Editar o /etc/rc.local.

   ```
   # vim /etc/rc.local
   ```

5. Adicionar a linha abaixo antes de **“exit 0”**.

   ```
   /usr/sbin/ntpdate-debian &
   ```

6. Digitar o comando abaixo e escolher o `vim.basic` como editor padrão do sistema.

   ```
   # update-alternatives --config editor
   ```

4 Configurar uso de memória do SO

1. Aumentar os limites de utilização de memória compartilhada. Necessário para tuning de PostgreSQL e outras aplicações que requerem compartilhamento de memória entre *forks*. A configuração abaixo é suficiente em muitos casos. Incluir as linhas abaixo no final do arquivo `"/etc/sysctl.conf"`.

   ```
   kernel.shmmax=133868503040
   kernel.shmmni=300
   kernel.shmall=133868503040
   kernel.sem=250 32000 32 300
   ```

2. Para não precisar reiniciar o servidor, basta digitar os comandos abaixo.

   ```
   # sysctl kernel.shmmax="133868503040"
   # sysctl kernel.shmmni="300"
   # sysctl kernel.shmall="133868503040"
   # sysctl kernel.sem="250 32000 32 300"
   ```

5 Configurar análise de *logs* automatizada

A aplicação *logwatch* possui várias funcionalidades para análise e sumarização automática de *logs* no sistema. A aplicação é executada pelo `crontab` e envia um e-mail periódico contendo informações úteis sobre os *logs*.

1. Instalar a aplicação de análise de *logs*.

   ```
   # apt-get install logwatch
   ```

2. Configurar *logwatch*.

   ```
   # echo "Format = html">> /etc/logwatch/conf/logwatch.conf
   ```

3. Para testar, digite o comando abaixo após a configuração do redirecionamento dos e-mails do usuário `root` para conta no GMail. Apenas será enviado e-mail se houver informações nos *logs* a serem processadas.

   ```
   # /usr/sbin/logwatch --output mail
   ```

6 Redirecionar mensagens de e-mail para conta no GMail

1. Instalar o *postfix* e aplicações úteis para e-mails.

   ```
   # apt-get install postfix mailutils
   ```

2. Adicionar em `/etc/mailname` um nome que o DNS aponte para o IP do servidor e o reverso do IP aponte para esse nome.

   ```
   # echo "nome.dominio.com.br"> /etc/mailname
   ```

3. Configurar o redirecionamento.

   ```
   # echo "root: reddhatt@gmail.com" >> /etc/aliases
   # newaliases
   ```

4. Reiniciar o `postfix`.

   ```
   # service postfix restart
   ```

5. Testar, digitando o comando abaixo e alguma mensagem depois. Para finalizar, basta digitar control+D.

   ```
   # mail -s Teste root
   ```

7 Rever regras de FW

Configuração de *firewalling* depende muito do ambiente, divisão dos servidores e uso de cada máquina. O ideal é liberar estritamente o que precisa utilizar e bloquear o restante.

1. Nesse sentido, a configuração utilizada pode estar boa. Apenas mudaria para a abaixo.

```
# iptables -F INPUT
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -p tcp -m multiport --dports 80,62834 -j ACCEPT
# iptables -A INPUT -j DROP
```

2. É interessante ter um IP totalmente liberado para acessar o servidor, que pode ser utilizado em situações de emergência. Para isso, pode-se utilizar o comando abaixo antes do `"DROP"`, substituindo `"IP_SAGRADO"` por um IP que se tenha acesso.

```
# iptables -A INPUT -s IP_SAGRADO -j ACCEPT
```

3. Deve-se liberar o IP do servidor Zabbix. Para isso, pode-se utilizar o comando abaixo antes do `"DROP"`.

```
# iptables -A INPUT -s IP_SERVER_ZABBIX -p tcp --dport 10050 -j ACCEPT
```

4. É recomendado liberar pacotes ICMP que são utilizados em controles nas conexões TCP/IP. Para isso, insira a linha abaixo antes do `"DROP"`.

```
# iptables -A INPUT -p icmp -m limit --limit 5/second -j ACCEPT
```

5. Dependendo do ambiente e da equipe, pode-se fazer *logs* de algumas tentativas suspeitas. Entretanto, isso não adianta muito se não houver alguém que monitore e tome alguma medida útil, que muita das vezes dá trabalho. Vale ressaltar também que fazer *logs* pode sobrecarregar o servidor em alguns casos. Neste caso, bastaria inserir a linha abaixo imediatamente antes do "DROP"‘. Considerando o contexto atual, recomendo *logs* apenas para depuração.

```
# iptables -A INPUT -m limit --limit 1/second -j LOG --log-level info --log-prefix "Pkt
Dropped"
```

6. É bastante útil bloquear os acessos do servidor para a Internet. Isso porque alguns ataques iniciam fazendo o servidor baixar os arquivos maliciosos do atacante para depois executá-los. Entretanto, é trabalhoso definir quais acessos o servidor precisa fazer para funcionar, como acessar os repositórios de pacote, resolver nomes DNS e outras ações. Pode-se considerar regras semelhantes às abaixo. Entretanto sempre será necessário testar antes de colocar em produção!

```
# iptables -F OUTPUT
# iptables -A OUTPUT -o lo -j ACCEPT
# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A OUTPUT -p tcp -m multiport --dports 80,443,10051 -j ACCEPT
# iptables -A OUTPUT -p udp -m multiport --dports 53 -j ACCEPT
# iptables -A OUTPUT -p tcp -m multiport --dports 25 -j ACCEPT
# iptables -A OUTPUT -j DROP
```

7. A configuração de *firewall* completa fica conforme abaixo.

```
# iptables -F INPUT
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A INPUT -p tcp -m multiport --dports 80,62834 -j ACCEPT
# iptables -A INPUT -s IP_SAGRADO -j ACCEPT
# iptables -A INPUT -s IP_SERVER_ZABBIX -p tcp --dport 10050 -j ACCEPT
# iptables -A INPUT -p icmp -m limit --limit 5/second -j ACCEPT
```

```
# # LOG recomendado apenas para depuração
# # iptables -A INPUT -m limit --limit 1/second -j LOG --log-level
  #  info --log-prefix "Pkt Dropped"
# iptables -A INPUT -j DROP
# iptables -F OUTPUT
# iptables -A OUTPUT -o lo -j ACCEPT
# iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# iptables -A OUTPUT -p tcp -m multiport --dports 80,443,10051 -j ACCEPT
# iptables -A OUTPUT -p udp -m multiport --dports 53 -j ACCEPT
# iptables -A OUTPUT -j DROP
```

8. A criação de regras na tabela `FORWARD` só é aplicável quando o firewall protege outras máquinas e não apenas a si própria.

9. Antes de salvar as regras, verifique se ainda possui acesso ao servidor via SSH. Abrindo nova conexão SSH e mantendo a anterior, que pode ainda ser usada em caso de problemas.

10. Finalmente, salvar as regras que estão vigentes para o próximo boot.

    ```
    # iptables-save > /etc/iptables.rules
    ```

8 Configurar Fail2Ban

1. O serviço mais importante a ser considerado na configuração do Fail2Ban é mesmo o `sshd`, que já está previsto no guia a ser executado antes deste. Se é utilizado serviço FTP (o que não é recomendado), deve-se habilitar o Fail2Ban para esse serviço também. Caso tenha alguma pasta HTTP com autenticação, pode-se habilitar.

9 Configurações de segurança no PHP

A extensão Suhosin implementa vários recursos para bloquear alguns ataques conhecidos do PHP.

1. Adicionar o repositório do Suhosin.

   ```
   # echo "deb http://repo.suhosin.org/ debian-jessie main">> /etc/apt/sources.list.d/suhosin.list
   ```

2. Adicionar chave do repositório no chaveiro do APT-GET.

   ```
   # wget -qO - https://sektioneins.de/files/repository.asc | apt-key add -
   ```

3. Atualizar lista de respositórios.

   ```
   # apt-get update
   ```

4. Instalar extensão Suhosin.

   ```
   # apt-get install php5-suhosin-extension
   ```

10 Configurações de segurança no Apache

1. Manter a configurações realizadas em `/etc/apache2/conf-available/security.conf`.

2. Adicionar as opções `-Indexes"` e `-MultiView"` nas diretivas `Directory`, como no exemplo abaixo.

   ```
   <Directory /var/www/html>
     Options -Indexes -MultiView
   </Directory>
   ```

3. Dentro das respectivas diretivas `<VirtualHost xxx>`, pode-se inserir a variável `"php_admin_value open_basedir"` para restringir a ação do PHP nos diretórios que seguem. Com isso, os programas PHP não têm acesso a outras pastas que não as listadas. Isso não é necessário para o Moodle, porém é recomendado para aplicações em que não se conheça bem a segurança do código.

```
# Manter as informações abaixo em um única linha!
php_admin_value open_basedir /usr/share/php5:/usr/share/php:
                             /tmp:/usr/share/php:/var/www
```

4. Reiniciar Apache.

```
# service apache2 restart
```

5. O Moodle já é bem seguro. Para outras aplicações, pode-se considerar a utilização do Apache em jaula. Quando isso é feito, aparecem vários ajustes que devem ser feitos para as aplicações funcionarem corretamente. Na prática é necessário inserir alguns arquivos na jaula, à medida que são necessários. Mesmo com essas questões, em muitos casos aumenta muito a segurança do sistema. Os procedimentos são simples e estão documentados em https://www.howtoforge.com/tutorial/chrooting-apache-2.4-with-mod_unixd-on-debian-8-jessie/

11 Tuning PHP

1. Configurar as variáveis abaixo no arquivo `/etc/php5/apache2/php.ini`, em suas respectivas seções.

```
max_execution_time = 300
max_input_time = 600
memory_limit = 96M
post_max_size = 300M
upload_max_filesize = 300M
date.timezone = America/Sao_Paulo
```

2. Adicionar as variáveis no arquivo `/etc/php5/apache2/php.ini`, na seção do `opcache`, conforme abaixo.

```
[opcache] ; ESSA LINHA JÁ DEVE ESTAR LÁ !
opcache.enable = 1
opcache.memory_consumption = 128
opcache.max_accelerated_files = 4000
opcache.revalidate_freq = 60

; Required for Moodle
opcache.use_cwd = 1
opcache.validate_timestamps = 1
opcache.save_comments = 1
opcache.enable_file_override = 0
```

3. Reiniciar Apache.

```
# service apache2 restart
```

12 Configurar Apache para ambiente Moodle

1. Atualmente os Moodles estão em diretórios de usuários no `/home`, o que não é recomendado pois:

   - impede que separe a partição do Moodle dos demais arquivos do sistema e usuários;
   - aumenta as chances de erros humanos;

- mistura de arquivos de usuário com arquivos do Moodle, o que pode dificultar a administração e backups; e
- aumento na complexidade da configuração do Apache.

O correto é colocar todos os arquivos `PHP` no `/var/www/html` e os arquivos dos `moodledata` em um diretório fora de `/var/www`.

Sugiro:

- Arquivos `PHP` do Moodle em `/var/www/html/NomeDoMoodle` e
- Arquivos do MoodleData em `/moodledata/NomedoMoodle`.

Nesse caso, o diretório `/moodledata` pode estar montado em uma partição separada, com formatação parametrizada.

13 Tuning Apache

A configuração desta seção considera um **servidor com 32GB de RAM e processador de 8 núcleos.**

1. Alterar a configuração do módulo `prefork` para a abaixo no arquivo `/etc/apache2/mods-enabled/mpm_prefork.conf`.

```
<IfModule mpm_prefork_module>
StartServers 20
MinSpareServers 20
MaxSpareServers 100
MaxRequestWorkers 600
MaxConnectionsPerChild 30
</IfModule>
```

2. Alterar a configuração do Apache como abaixo, no arquivo `/etc/apache2/apache2.conf`.

```
Timeout 300
ServerLimit 600
ServerName nomedoservidor  # Adicione um nome de servidor para evitar warnings.

KeepAlive Off
#MaxKeepAliveRequests 1 # Pode ser comentada
#KeepAliveTimeout 5     # Pode ser comentada
```

3. Reiniciar o Apache.

```
# service apache2 restart
```

14 Instalar e configurar sumarizador de logs para Apache

1. Alterar a retenção de *logs* do Apache para 30 dias. Basta alterar a variável abaixo no arquivo `/etc/logrotate.d/apache2`.

```
rotate 30
```

2. Copiar os arquivos de configuração padrão dos módulos `http` e `http-error` do *LogWatch*.

```
# cp /usr/share/logwatch/default.conf/services/http-error.conf
     /etc/logwatch/conf/services/
# cp /usr/share/logwatch/default.conf/services/http.conf
     /etc/logwatch/conf/services/
```

3. O correto é não fazer nenhuma alteração nesses arquivos em um primeiro momento e averiguar se os sumários terão informações úteis. Caso venha muitas informações que não representem riscos, exemplo mensagens de *warnings* e outras coisas, pode ser necessário fazer ajustes para filtrar informações que não são tão importantes.

   Como exemplo, no arquivo `/etc/logwatch/conf/services/http-error.conf` descomentar as linhas abaixo.

   ```
   $ignore_not_exist_all = Yes
   $ignore_not_exist_no_referer = Yes
   ```

   e no arquivo `/etc/logwatch/conf/services/http.conf` setar a variável abaixo para `1`.

   ```
   $HTTP_IGNORE_ERROR_HACKS = 1
   ```

4. Os e-mails do *LogWatch* serão enviados diariamente, quando houver *logs*, para o `root@localhost`, que é encaminhado para uma conta no GMail. Verifique se não caiu na caixa de SPAM.

15 Tuning MySQL

A configuração desta seção considera um **servidor com 32GB de RAM e processador de 8 núcleos.**

1. Altere a configuração da seção `[mysqld]` do arquivo `/etc/mysql/my.cnf` conforme abaixo.

   ```
   [mysqld]

   user                    = mysql
   pid-file                = /var/run/mysqld/mysqld.pid
   socket                  = /var/run/mysqld/mysqld.sock
   port                    = 3306
   basedir                 = /usr
   datadir                 = /var/lib/mysql
   tmpdir                  = /tmp
   lc-messages-dir         = /usr/share/mysql
   bind-address            = 127.0.0.1
   log_error               = /var/log/mysql/error.log

   skip-external-locking
   skip-name-resolve

   max_connections         = 600
   wait_timeout            = 7200
   connect_timeout         = 10
   max_allowed_packet      = 16M
   expire_logs_days        = 10
   tmp_table_size          = 1G
   max_heap_table_size     = 512M
   bulk_insert_buffer_size = 64M

   thread_stack            = 256K
   thread_cache_size       = 20
   thread_concurrency      = 8

   query_cache_type        = 1
   query_cache_limit       = 256K
   query_cache_size        = 128M
   ```

```
myisam-recover-options  = BACKUP
max_binlog_size         = 100M
table_cache             = 1000
key_buffer              = 512M
myisam_sort_buffer_size = 16M
sort_buffer_size        = 16M
read_buffer_size        = 16M
read_rnd_buffer_size    = 8M
join_buffer_size        = 2G

innodb_buffer_pool_size = 10G
innodb_log_file_size    = 512M
innodb_file_per_table   = ON
innodb_flush_method     = O_DIRECT
innodb_buffer_pool_instances = 8

slow_query_log_file     = /var/log/mysql/mysql-slow.log
slow_query_log          = 1
long_query_time         = 5
log_queries_not_using_indexes = 1
log_error = /var/log/mysql/error.log
```

2. Pode ser necessário remover os *logs* transacionais do MySQL para a nova configuração. **Atenção: Apenas faça isso em instalação nova, pois pode-se perder dados.**

```
# service mysql stop
# rm /var/lib/mysql/ib_logfile0
# rm /var/lib/mysql/ib_logfile1
```

3. Reinicie o MySQL.

```
# service mysql restart
```

4. Verifique os *logs* para ver se ocorre algum erro.

```
# less /var/log/mysql/error.log
```

5. O uso de discos SSD para os arquivos do MySQL em `/var/lib/mysql` aumentam significativamente a performance!

6. É recomendado ter um servidor exclusivo para para banco de dados separado dos servidores Apache, pelas seguintes razões:

   - Simplifica a manutenção e atualização de programas;
   - O tuning do MySQL fica mais fácil ser feito, considerando que o servidor é todo do MySQL;
   - Pode-se ter nesse servidor apenas disco SSD, o que aumenta a performance consideravelmente;
   - Na execução, o MySQL não precisa concorrer com o Apache no uso dos recursos de hardware e chamadas a sistema;
   - O Apache é escalável facilmente, enquanto não é tão simples escalar o banco de dados. Assim, em situações de grande volume de acessos, é mais fácil organizar os servidores para atender à demanda.

16 Instalar e configurar sumarizador de *logs* para MySQL

1. Alterar a retenção de *logs* do MySQL para 15 dias. Basta alterar a variável abaixo no arquivo `/etc/logrotate.d/mysql-server`.

```
rotate 15
```

2. Copiar o arquivo de configuração padrão do módulos `mysql` do *LogWatch*.

   ```
   # cp /usr/share/logwatch/default.conf/services/mysql.conf
        /etc/logwatch/conf/services/
   ```

3. O correto é não fazer nenhuma alteração nesse arquivo em um primeiro momento e averiguar se os sumários terão informações úteis. Caso venha muitas informações que não representem riscos, pode ser necessário fazer ajustes para filtrar informações que não são tão importantes.

   Como exemplo, no arquivo `/etc/logwatch/conf/services/mysql.conf` alterar a variável abaixo para `Low`

   ```
   Detail = Low
   ```

4. Os e-mails do *LogWatch* serão enviados diariamente, quando houver *logs*, para o `root@localhost`, que é encaminhado para uma conta no GMail. Verifique se não caiu na caixa de SPAM.

17 Referências

Abaixo são relacionadas algumas referências que podem ajudar a entender o que é feito.

1. http://pandoc.org
2. http://ajmoreti.blogspot.com.br/2013/08/configurando-shmmax-e-shmall-para.html
3. https://n0where.net/how-does-it-work-iptables/
4. https://kevin.deldycke.com/2012/01/how-to-generate-pdf-markdown/
5. https://devops.profitbricks.com/tutorials/install-and-configure-logwatch/
6. https://en.wikipedia.org/wiki/Markdown
7. https://www.howtoforge.com/tutorial/chrooting-apache-2.4-with-mod_unixd-on-debian-8-jessie/
8. https://suhosin.org/
9. https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html
10. https://spin.atomicobject.com/2011/05/09/mysql-failed-registration-of-innodb-as-a-storage-engine/