

Procedimento

Servidor Apache com PHP e Tomcat

Autor: Sandro Venezuela <sandro@linux2business.com.br>



Atribuição – Uso não-comercial – Compartilhamento pela mesma licença 2.5 Brasil

Você pode:



Copiar, distribuir, exibir e executar a obra.

Sob as seguintes condições:



Atribuição: Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.



Uso não-comercial: Você não pode utilizar esta obra com finalidades comerciais



Compartilhamento pela mesma licença: Se você alterar, transformar ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

A reprodução do material contido neste tutorial é permitido desde que se incluam os créditos ao autor e a frase: “**Reproduzido da Linux2Business — www.linux2business.com.br**” em local visível.

Índice

Versão.....	4
Objetivo.....	5
CentOS.....	6
Instalação.....	6
Configuração.....	6
Serviços desnecessários.....	6
Desabilitar Ctrl-Alt-Del.....	7
Desabilitar Terminais.....	7
Desabilitar Acesso Local para Usuário root.....	7
Desabilitar Acesso SSH para Usuário root.....	8
SNMP.....	8
SUDO.....	8
Apache.....	10
PHP.....	12
Tomcat.....	13

Versão

Criado/Alterado	Data	Versão
Sandro Venezuela	15/04/2011	V1.0
Sandro Venezuela	15/06/2011	V1.1

Objetivo

Apresentar os procedimentos de instalação e configuração do servidor *Apache* com *PHP* e *Tomcat*., incluindo a configuração de domínios virtuais e também a integração do *Tomcat* com o *Apache*.

CentOS

Instalação

Iniciar o servidor através da unidade de CD/DVD com a mídia do *CentOS 5*. A instalação deve ocorrer sempre no idioma *English*.

Na configuração do fuso horário deve-se marcar sempre a opção *UTC* para definição da data e hora.

O particionamento do disco deve obedecer a seguinte configuração:

Partição	Ponto de Montagem	Tamanho
/dev/sda1	/	1 GB
/dev/sda2	/usr	4 GB
/dev/sda3	swap	1 GB
/dev/sda5 (LVM)	/home	1 GB
	/tmp	1 GB
	/var	10 a 30 GB
	/var/www	>10

Obs.: A partição */var/www* deve ter o tamanho necessário para armazenar todos os arquivos relacionados ao site e também as aplicações *Java*.

Normalmente a instalação de um servidor é realizada com a quantidade mínima de pacotes, onde para realizar tal configuração no *CentOS* é necessário selecionar a opção *Customize Now* e desmarcar todos os grupos de pacotes.

Obs.: Somente é possível realizar a instalação mínima através da instalação gráfica.

Deve ser criado o usuário *sysadmin*, para administração do servidor e com isto evitar o uso do usuário *root*.

Após a instalação, caso sejam necessários alguns pacotes extras, como *nmap*, *tcpdump*, *crontabs*, etc, estes devem ser instalados através do comando **YUM**.

Configuração

Serviços desnecessários

Desabilitando os serviços desnecessários

```
# chkconfig haldaemon off
# chkconfig kudzu off
# chkconfig mcstrans off
# chkconfig messagebus off
# chkconfig netfs off
# chkconfig restorecond off
```

Desabilitar Ctrl-Alt-Del

Editar o arquivo `/etc/inittab`, comentando a seguinte linha:

```
# what to do when CTRL-ALT-DEL is pressed
# ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

Para habilitar a alteração, execute o comando:

```
# init q
```

Desabilitar Terminais

Editar o arquivo `/etc/inittab`, comentando a seguinte linha, em negrito:

```
...
# for ARGO UPS
sh:12345:powerfail:/sbin/shutdown -h now THE POWER IS FAILING

# getty-programs for the normal runlevels
# <id>:<runlevels>:<action>:<process>
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
# 3:2345:respawn:/sbin/mingetty tty3
# 4:2345:respawn:/sbin/mingetty tty4
# 5:2345:respawn:/sbin/mingetty tty5
# 6:2345:respawn:/sbin/mingetty tty6
#
#S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102
#cons:12345:respawn:/sbin/smart_agetty -L 38400 console
...
```

Normalmente devem ser permitidos somente 2 terminais, acessíveis localmente através das teclas `Alt+F1` e `Alt+F2`. Se for necessário mais terminais, basta habilitar, descomentando o terminal correspondente.

Para habilitar a alteração, execute o comando:

```
# init q
```

Desabilitar Acesso Local para Usuário root

Por padrão, não deve ser permitido o acesso local para o usuário `root`. Para bloquear este acesso, remova todas as linhas do arquivo `/etc/securetty`, conforme apresentado abaixo:

```
# cp -p /etc/securetty /etc/securetty.default
# cat /dev/null > /etc/securetty
```

Obs.: Este procedimento SOMENTE deve ser realizado após a criação de pelo menos um usuário, normalmente criado no momento da instalação.

Desabilitar Acesso SSH para Usuário root

Por padrão, não deve ser permitido o acesso via *SSH* para o usuário *root*. Para bloquear este acesso é necessário incluir ou alterar as seguintes linhas no arquivo `/etc/ssh/sshd_config`, conforme apresentado abaixo:

```
PermitRootLogin no  
AllowUsers sysadmin
```

Para que as alterações sejam ativadas é preciso reiniciar o serviço *SSH*:

```
# service sshd restart
```

Obs.: Este procedimento SOMENTE deve ser realizado após a criação de pelo menos um usuário, normalmente criado no momento da instalação.

SNMP

Obs.: Este serviço somente deve ser instalado e habilitado se o servidor for monitorado via *SNMP*, senão, esta etapa pode ser desconsiderada.

Para o serviço de monitoramento do servidor, devemos instalar o pacote `net-snmp` através do **YUM**. Em seguida, deve-se criar o arquivo `snmpd.conf`, no diretório `/etc/snmp`, com o seguinte conteúdo:

```
com2sec local 127.0.0.1/32 private  
com2sec local 192.168.0.39/32 linux2business  
  
group MyROGroup v1 local  
group MyROGroup v2c local  
group MyROGroup usm local  
  
view all included .1 80  
  
access MyROGroup "" any noauth exact all none none  
  
syslocation Linux2Business  
syscontact System Admin <sysadmin@linux2business.com.br>
```

Obs.: O endereço IP 192.168.0.39 deve ser substituído pelo endereço do seu servidor de monitoramento via *SNMP*.

Por fim, devemos iniciar o serviço *SNMP*:

```
# service snmpd start
```

E habilitar para que o serviço seja sempre iniciado junto com o sistema operacional:

```
# chkconfig snmpd on
```

SUDO

Para esta funcionalidade, deve-se instalar o pacote `sudo` através do **YUM**.

Com o comando `visudo`, que altera o arquivo `/etc/sudoers`, devemos adicionar os seguintes parâmetros para o usuário `sysadmin`:

```
# visudo
(Incluir ao final do arquivo)
# SysAdmin User
sysadmin ALL = NOPASSWD: /sbin/reboot,
                    /sbin/service httpd restart,
                    /sbin/service tomcat restart
```

Com a configuração acima o usuário `sysadmin` terá o “poder” de reiniciar o servidor e reiniciar os serviços *Apache* e *Tomcat*.

Outros comandos podem ser configurados, porém devem estar de acordo com a política de TI da empresa.

Apache

Instalar os pacotes `httpd` e `mod_ssl` utilizando o **YUM**:

```
# yum install httpd mod_ssl
```

A configuração do Apache ocorre através do arquivo `/etc/httpd/conf/httpd.conf`, onde somente os parâmetros abaixo foram alterados ou incluídos no arquivo original:

Arquivo `/etc/httpd/conf/httpd.conf`:

```
(Alterar os seguintes parâmetros)
ServerAdmin sysadmin@linux2business.br
ServerName www.linux2business.br
ServerTokens OS
ServerSignature Off
```

Para habilitar o SSL, antes é necessário instalar o pacote `make`, através do **YUM**:

```
# yum install make
```

Obs.: Se não for possível instalar o pacote `make` é possível criar os certificados através da referência abaixo:

- How to Create Self-Signed SSL Certificates with OpenSSL
http://www.xenocafe.com/tutorials/linux/centos/openssl/self_signed_certificates/index.php

Em seguida, deve-se executar o procedimento abaixo:

```
# cd /etc/pki/tls/certs
# make server.crt
(Responder as perguntas...)
[enter your private key password]: Senha para criacao do certificado
[enter your two character country code]: BR
[enter your full state or province name]: Sao Paulo
[enter your city name]: Santo Andre
[enter your company name]: Linux2Business
[enter your organizational unit or leave it blank]: TI
[enter your common name or fqdn]: www.linux2business.br
[enter your admin email address]: sysadmin@linux2business.br
```

Para remover a senha da chave privada, evitando assim que o Apache sempre pergunte a senha ao iniciar, deve-se executar o comando abaixo:

```
# cd /etc/pki/tls/certs
# openssl rsa -in server.key -out ../private/server.key
[enter your private key password]: Informar a senha criada acima
```

Por fim, o arquivo `/etc/httpd/conf.d/ssl.conf` deve ser alterado, conforme abaixo:

Arquivo `/etc/httpd/conf/httpd.conf`:

```
(Alterar os seguintes parâmetros)
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/private/server.key
```

Depois de tudo configurado, o serviço deve ser iniciado e habilitado para que seja iniciado junto com o sistema operacional:

```
# service httpd restart
# chkconfig httpd on
```

Para verificar se o *Apache* está funcionando corretamente, deve-se executar o comando:

```
# netstat -ntlp | grep http
tcp  0  0  :::80  :::*  LISTEN  <Numero do PID>/httpd
tcp  0  0  :::443  :::*  LISTEN  <Numero do PID>/httpd
```

Finalizando a configuração do *Apache*, a seguir está a configuração para criação dos domínios virtuais *www.linux2business.br* e *www.linux2business.org.br*. Deve-se criar o arquivo *vhosts.conf* no diretório */etc/httpd/conf.d* com o seguinte conteúdo:

Arquivo */etc/httpd/conf.d/vhosts.conf*:

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerAdmin sysadmin@linux2business.br
    DocumentRoot /var/www/html/www.linux2business.br
    ServerName www.linux2business.br
    ErrorLog logs/www.linux2business.br-error_log
    CustomLog logs/www.linux2business.br-access_log common
</VirtualHost>

<VirtualHost *:80>
    ServerAdmin sysadmin@linux2business.org.br
    DocumentRoot /var/www/html/www.linux2business.org.br
    ServerName www.linux2business.org.br
    ErrorLog logs/www.linux2business.org.br-error_log
    CustomLog logs/www.linux2business.org.br-access_log common
</VirtualHost>
```

Por fim, devem ser criados os diretórios *www.linux2business.br* e *www.linux2business.org.br* dentro do diretório */var/www/html*.

PHP

Instalar os pacotes `php` e `php-pear` utilizando o **YUM**:

```
# yum install php php-pear
```

Obs.: Se for necessário instalar outros módulos do PHP, como `php-gd`, `php-ldap`, etc, basta incluir o nome do pacote no comando acima.

Em seguida, o Apache deve ser reiniciado:

```
# service httpd restart
```

Para verificar se tudo está funcionando corretamente, deve ser criado o arquivo `/var/www/html/phpinfo.php` com o seguinte conteúdo:

Arquivo `/var/www/html/phpinfo.php`:

```
<?php
    phpinfo();
?>
```

Acessar o endereço `http://nome_do_servidor/phpinfo.php`, onde as informações sobre o PHP serão disponibilizadas.

Tomcat

Instalar os pacotes `tomcat5` e `tomcat5-admin-webapps` utilizando o **YUM**:

```
# yum install tomcat5 tomcat-admin-webapps
```

Obs.: O pacote `tomcat5-webapps` somente deve ser instalado em ambiente de desenvolvimento e teste, nunca em ambiente de produção.

Em seguida, o Java deve ser atualizado para a versão 1.6.0, pois a versão padrão é 1.4.2.

Obs.: Os procedimentos abaixo foram seguidos da seguinte referência:

- [HowTo Install Java on CentOS 4 and CentOS 5](http://wiki.centos.org/HowTos/JavaOnCentOS)
<http://wiki.centos.org/HowTos/JavaOnCentOS>

Para realizar a atualização, primeiro deve-se instalar o pacote `rpm-build`, através do **YUM**:

```
# yum install rpm-build
```

Depois, deve-se copiar o arquivo `java-1.6.0-sun-1.6.0.14-1jpp.nosrc.rpm` do endereço <http://mirrors.dotsrc.org/jpackage/5.0/generic/non-free/SRPMS/>:

```
# wget http://mirrors.dotsrc.org/jpackage/5.0/generic/non-free/SRPMS/java-1.6.0.14-1jpp.nosrc.rpm
```

Em seguida, deve-se copiar o *Java JDK 1.6 update 14* do site Sun JDK Archive (<http://java.sun.com/products/archive/>) escolhendo a plataforma “Linux” ou “Linux x64” e a extensão `.BIN` do arquivo.

Este arquivo deve ser copiado para o diretório `/usr/src/redhat/SOURCES`:

```
# mv jdk-6u14-linux-i586.bin /usr/src/redhat/SOURCES
```

Por fim, deve-se executar o comando abaixo para a criação dos pacotes do Java:

```
# rpmbuild --rebuild java-1.6.0-sun-1.6.0.14-1jpp.nosrc.rpm
```

Serão criados os seguintes pacotes no diretório `/usr/src/redhat/RPMS/i586`:

- `java-1.6.0-sun-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-devel-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-src-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-demo-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-plugin-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-fonts-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-alsa-1.6.0.14-1jpp.i586.rpm`
- `java-1.6.0-sun-jdbc-1.6.0.14-1jpp.i586.rpm`

Por fim, deve-se instalar os pacotes, conforme procedimento abaixo:

```
# rpm -Uvh java-1.6.0-sun-1.6.0.14-1jpp.i586.rpm \  
> java-1.6.0-sun-devel-1.6.0.14-1jpp.i586.rpm \  
> java-1.6.0-sun-plugin-1.6.0.14-1jpp.i586.rpm
```

Para verificar, deve-se executar o comando:

```
# java -version
java version "1.6.0_14"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.6.0_14-b08)
Java HotSpot(TM) Client VM (build 14.0-b16, mixed mode, sharing)
```

Se a resposta não for a exibida acima, então execute o comando:

```
# alternatives --config java
```

Obs.: Você pode verificar os links criados no diretório `/etc/alternatives`, principalmente os links do `java` e `java_sdk`, que devem referenciar a versão 1.6.0 do Java.

Depois de tudo configurado, o serviço deve ser iniciado e habilitado para que seja iniciado junto com o sistema operacional:

```
# service tomcat5 start
# chkconfig tomcat5 on
```

Para verificar se o *Tomcat* está funcionando corretamente, deve-se executar o comando:

```
# netstat -ntlp | grep ":8080"
tcp 0 0 :::8080 :::* LISTEN <Numero do PID>/java
```

O próximo passo será a habilitação dos aplicativos *Manager* e *Web Admin* do *Tomcat*. O *Manager* permite administrar outras aplicações enquanto o servidor estiver em execução, como realizar *deploy*, *undeploy*, iniciar e parar uma aplicação.

O *Web Admin* permite administrar o servidor e aplicações individuais, como editar o arquivo `server.xml`.

Estas funções foram instaladas através do pacote `tomcat5-admin-webapps` e para criar um usuário com permissão de acessar estas aplicações, devemos realizar a seguinte alteração no arquivo `/etc/tomcat5/tomcat-users.xml`, acrescentando os parâmetros abaixo:

Arquivo `/etc/tomcat5/tomcat-users.xml`:

```
(Acrescentar)
<role rolename="manager"/>
<role rolename="admin"/>
<user username="admin" password="admin" roles="manager,admin"/>
```

Obs.: Na configuração acima foi criado o usuário `admin` com permissão de acesso em ambos aplicativos de administração, porém é possível criar um usuário para cada aplicativo.

Para que a configuração seja ativada é necessário reiniciar o *Tomcat*:

```
# service tomcat5 restart
```

Finalizando a configuração do *Tomcat*, deve-se integrá-lo ao *Apache* através do módulo *Proxy AJP*, configurado através do arquivo `/etc/httpd/conf.d/proxy_ajp.conf`, conforme abaixo:

Arquivo `/etc/httpd/conf.d/proxy_ajp.conf`:

```
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
ProxyPass /jsp-examples ajp://localhost:8009/jsp-examples
```

```
ProxyPass /servlets-examples ajp://localhost:8009/servlets-examples
```

Para que a configuração seja ativada é necessário reiniciar o *Apache*:

```
# service apache restart
```

Para verificar se tudo está funcionando corretamente, digite `http://localhost/jsp-examples` e deve aparecer a tela de exemplos de *JSP*. Outro teste pode ser o acesso a tela principal do *Tomcat*: `http://localhost:8080`.

Como configurações adicionais, pode-se habilitar os registros das ações do *Tomcat* pela válvula *AccessLog* ou pela biblioteca *Log4j*. No caso da primeira opção, basta editar o arquivo `/etc/tomcat5/server.xml` conforme abaixo:

Arquivo `/etc/tomcat5/server.xml`:

```
(Remover o comentário das linhas 348 e 352 deixando conforme abaixo)
...
<Valve className="org.apache.catalina.valves.AccessLogValve"
  directory="logs" prefix="localhost_access" suffix=".log"
  fileDateFormat="yyyy-MM-dd" pattern="common" resolveHosts="false"/>
...
```

Os registros do *Tomcat* serão armazenados no arquivo `localhost_access.<Data no formato ano-mês-dia>.log` dentro do diretório `/var/log/tomcat5`.

No caso do *Log4j*, primeiro deve-se instalar os pacotes `log4j` e `jakarta-commons-logging`, através do **YUM**. Em seguida, deve-se habilitar as bibliotecas no *Tomcat*:

```
# cd /usr/share/tomcat5/common/lib
# ln -s /usr/share/java/log4j.jar log4j.jar
# ln -s /usr/share/java/commons-logging.jar commons-logging.jar
```

Para finalizar a configuração, deve-se criar o arquivo `log4j.properties` no diretório `/usr/share/tomcat5/common/classes`, conforme abaixo:

Arquivo `/etc/tomcat5/server.xml`:

```
log4j.rootLogger=INFO, Tomcat
log4j.appender.Tomcat=org.apache.log4j.FileAppender
log4j.appender.Tomcat.File=$CATALINA_HOME/logs/tomcat.log
log4j.appender.Tomcat.layout=org.apache.log4j.PatternLayout
log4j.appender.Tomcat.layout.ConversionPattern=%d{dd/MM/yyyy HH:mm:ss} %p %c :
%m %n
```

Deve-se reiniciar o *Tomcat* para que as alterações tenham efeito.