

# Relatório De Atividades Do Estágio

Hartur Barreto Brito

Junho de 2015

## 1 Servidor *SVN* Com Acesso Via http

### 1.1 Instalando os pacotes

Inicialmente deve-se instalar os pacotes necessários para realizar o acesso ao servidor *SVN* via http.

Para isso, devem ser utilizados os seguintes comandos:

```
sudo apt-get install subversion
sudo apt-get install apache2
sudo apt-get install libapache2-svn
sudo apt-get install apache2-utils
```

### 1.2 Configurando *Apache2* para acessar o *SVN* via http

Com os pacotes instalados, o *Apache2* precisa ser configurado para que possa ser realizado o acesso ao servidor.

Para configurar o *Apache2*, o arquivo “dav\_svn.conf” deve ser alterado, sendo este acessado da seguinte maneira:

```
sudo pico /etc/apache2/mods-enabled/dav_svn.conf
```

Nesse arquivo, devem ser deletados os # no início das seguintes linhas:

```
<Location /svn>
...
DAV svn
...
SVNParentPath /var/lib/svn
...
AuthType Basic
AuthName "Subversion Repository"
AuthUserFile /etc/apache2/dav_svn.passwd
...
<LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
```

```
</LimitExcept>
...
</Location>
```

Com isso, essas linhas deixarão de ser apenas comentários e passarão a fazer parte da configuração do *Apache2*.

Com essas configurações, será possível realizar um *checkout* sem a utilização de um usuário válido. Entretanto, para que seja possível realizar comandos de *import*, *commit*, dentre outros comandos, será necessário efetuar o *login* com um usuário do *SVN*. Para ver como criar um novo usuário, ir para sessão 1.4.

### 1.3 Criando Repositório

Após configurar o *Apache2*, deverá ser criado um repositório. Para isso, foi deve ser utilizado o seguinte comando:

```
sudo svnadmin create /var/lib/svn/myProject
```

Note que o repositório foi criado na mesma pasta que foi definida como sendo a pasta do servidor *SVN* na variável *SVNParentPath* configurada no arquivo “*dav\_svn.conf*”.

### 1.4 Adicionando Usuários ao *SVN*

Para criar o primeiro usuário, o seguinte comando deve ser utilizado:

```
sudo htpasswd -cm /etc/apache2/dav_svn.passwd USERNAME
```

A *flag -c* irá criar o arquivo caso ele não exista, e sobrescrever caso exista. Para criar os outros usuários, será necessário utilizar apenas a *flag -m*, o que faz com que seja adicionado um novo usuário ao servidor. Dessa maneira, o comando deverá ser escrito como:

```
sudo htpasswd -m /etc/apache2/dav_svn.passwd USERNAME
```

A execução de qualquer um dos dois comandos irá solicitar que seja escrita a senha do novo usuário de acesso ao servidor *SVN*.

### 1.5 Atribuindo Permissões

Após a criação do repositório, é necessário permitir que esse diretório seja alterado via *http*. Para adicionar as permissões, basta utilizar os seguintes comandos:

```
sudo chown -R www-data:www-data /var/lib/svn/myProject
sudo chmod -R g+rws /var/lib/svn/myProject
```

As permissões de acesso devem ser estabelecidas para cada um dos repositórios criados.

Após seguir esses passos, o *Apache2* deve ser reiniciado com o seguinte comando:

```
sudo service apache2 restart
```

Com isso, o servidor *SVN* com acesso via **http** já deverá estar funcionando. Para testar, basta tentar acessar o link `http://localhost/svn/myProject` da sua máquina. Para acessar o servidor de outra máquina será necessário conhecer o *ip* do servidor (que pode ser obtido utilizando o comando `ifconfig`). Com ele, basta acessar `http://ServerIp/svn/myProject`.

## 1.6 Modificando Nome do Repositório

Caso seja necessário modificar o nome do repositório, basta acessar o diretório onde ele foi criado ("`/var/lib/svn`") e utilizar os seguintes comandos:

```
mv old_repo_name new_repo_name
svnadmin setuuid new_repo_name
```

## 2 Configurando IP Fixo

Para fazer com que o servidor tenha um IP fixo, facilitando assim o acesso à ele, o arquivo `interfaces` deve ser modificado, sendo este acessado a partir do comando:

```
sudo pico /etc/network/interfaces
```

As seguintes linhas devem ser escritas no arquivo:

```
auto eth1
iface eth1 inet static
    address 150.161.49.126
    netmask 255.255.255.0
    gateway 150.161.49.254
    dns-nameservers 150.161.49.254
    dns-search lan
```

## 3 Configuração de *Path-Based File Access*

Para permitir que apenas alguns usuários possam ler e/ou escrever em alguns diretórios do repositório, o arquivo "`dav_svn.conf`" deve ser alterado, sendo ele acessado pelo comando:

```
nano /etc/apache2/mods-enabled/dav_svn.conf
```

retirando o `#` do início das linhas:

```
...
<IfModule mod_authz_svn.c>
    AuthSVNAccessFile /etc/apache2/dav_svn.authz
</IfModule>
...
```

Com essa alteração, será ativado o módulo de *Path-Based File Access*.

O arquivo “dav\_svn.authz”, encontrado no diretório “/etc/apache2/”, deve ser alterado seguindo o modelo:

```
[repo:/some/dir]
    user1 = rw
    user2 = r
[/]
    * = r
```

No qual:

- **rw**: Adiciona permissão de leitura e escrita ao usuário;
- **r**: Adiciona permissão de leitura ao usuário;
- **[repo:/some/dir]**: Referencia algum diretório do repositório “**repo**”;
- **[/]**: Referencia todos os repositórios;
- **\* = r**: Permite que qualquer pessoa tenha acesso ao menos com permissão de leitura à todos os repositórios.

## 4 Instalação do *Bugzilla*

### 4.1 Instalações Necessárias

Primeiramente devem ser instalados os pacotes necessários com o comando (Obs: O comando está quebrado em mais de uma linha, mas ele consiste em apenas 1 comando, que deverá ser escrito na mesma linha):

```
apt-get install apache2 mysql-server libappconfig-perl libdate-calc-perl
libtemplate-perl libmime-perl build-essential libdatettime-timezone-perl
libdatettime-perl libemail-sender-perl libemail-mime-perl
libemail-mime-modifier-perl libdbi-perl libdbd-mysql-perl libcgi-pm-perl
libmath-random-isaac-perl libmath-random-isaac-xs-perl apache2-mpm-prefork
libapache2-mod-perl2 libapache2-mod-perl2-dev libchart-perl libxml-perl
libxml-twig-perl perlmagick libgd-graph-perl libtemplate-plugin-gd-perl
libsoap-lite-perl libhtml-scrubber-perl libjson-rpc-perl libdaemon-generic-perl
libtheschwartz-perl libtest-taint-perl libauthen-radius-perl libfile-slurp-perl
libencode-detect-perl libmodule-build-perl libnet-ldap-perl libauthen-sasl-perl
libtemplate-perl-doc libfile-mimeinfo-perl libhtml-formattext-withlinks-perl
libgd-dev lynx-cur python-sphinx
```

Em seguida, deve ser baixada uma versão do *Bugzilla* clonando diretamente do *git*, na pasta `/var/www/html`. Para que isso seja feito, deve-se inicialmente acessar o diretório utilizando o comando:

```
cd /var/www/html
```

E em seguida, clonar a versão do *Bugzilla* com o comando (Obs: O comando está quebrado em mais de uma linha, mas ele consiste em apenas 1 comando, que deverá ser escrito na mesma linha):

```
git clone --branch release-4.4-stable https://git.mozilla.org/bugzilla  
/bugzilla bugzilla
```

## 4.2 Configuração do *MySQL*

Para realizar as configurações necessárias no *MySQL* para o correto funcionamento do *Bugzilla*, o arquivo `/etc/mysql/my.cnf` deve ser alterado da seguinte forma:

- Adicionar a linha `ft_min_word_len=2` após a linha `[mysqld]` (localizado na linha 32);
- Alterar o valor da variável `max_allowed_packet` de `16M` para `100M` (localizado na linha 52).

Além disso, para que o *Bugzilla* tenha acesso aos bancos do servidor *MySQL*, devem ser adicionados os privilégios necessários ao usuário do *Bugzilla* (**bugs**) no *MySQL* com o seguinte comando (Obs: O comando está quebrado em mais de uma linha, mas ele consiste em apenas 1 comando, que deverá ser escrito na mesma linha):

```
mysql -u root -p -e "GRANT ALL PRIVILEGES ON bugs.* TO bugs@localhost  
IDENTIFIED BY 'YOUR_PASSWORD'"
```

Após realizar essas modificações, o *MySQL* deve ser reiniciado utilizando o seguinte comando:

```
service mysql restart
```

## 4.3 Configurações do *Bugzilla*

Para realizar as configurações iniciais do *Bugzilla*, o arquivo `/etc/apache2/sites-available/bugzilla.conf` deve ser alterado, criando-o caso não exista, adicionando as seguintes linhas:

```
ServerName localhost
```

```
<Directory /var/www/html/bugzilla>  
    AddHandler cgi-script .cgi  
    Options +ExecCGI  
    DirectoryIndex index.cgi index.html  
    AllowOverride Limit FileInfo Indexes Options  
</Directory>
```

E então, o *Bugzilla* deve ser ativado utilizando os comandos:

```
a2ensite bugzilla
a2enmod cgi headers expires
```

Esses comandos mostraram como saída no **terminal** que o site do *Bugzilla* e os módulos `cgi`, `headers` e `expires` foram ativados.

Para concluir a ativação, é necessário reiniciar o *Apache2* com o seguinte comando:

```
service apache2 restart
```

Após isso, alguns *scripts*, que são encontrados no diretório onde foi baixado o *Bugzilla* (`/var/www/html/bugzilla`), devem ser executados. Esses *scripts* irão verificar a existência dos módulos necessários para o funcionamento do *Bugzilla*.

Primeiramente, deve-se acessar o diretório, utilizando o comando:

```
cd /var/www/html/bugzilla
```

Então, o *script* `checksetup.pl` deve ser executado, utilizando o comando:

```
./checksetup.pl
```

Esse comando irá mostrar que alguns módulos estão faltando. Para instalá-los, basta utilizar o comando:

```
/usr/bin/perl install-module.pl --all
```

Após a instalação dos módulos, o `checksetup.pl` deve ser executado novamente.

Com essa segunda execução, o arquivo `localconfig` será criado. Ele deverá ser modificado da seguinte forma:

- Utilizar o comando:

```
nano localconfig
```

- Alterar as linhas:

```
...
@webservergroup = 'apache'
...
$db_pass = ''
...
```

para:

```
...
@webservergroup = 'www-data'
...
$db_pass = 'YOUR_PASSWORD'
...
```

Após modificar o arquivo `localconfig`, o *script* `checksetup.pl` deverá ser executado novamente. Nessa execução serão solicitados email, nome e senha do administrador do *Bugzilla*.

Por fim, para testar se tudo está funcionando, o script `testserver.pl` deverá ser executado com o comando:

```
./testserver.pl http://localhost/bugzilla
```

Esse comando terá como saída no **terminal** a informação dos módulos estão funcionando, mostrando **TEST-OK** e, caso ainda haja algum problema, irá informar o que pode ser a causa dele.

## 4.4 Configurações Pelo Link

Com o *Bugzilla* funcionando e online, precisamos realizar as últimas configurações necessárias. Para isso, basta acessar `http://localhost/bugzilla/`, realizar o login com o usuário administrador, ir na aba *Administration* e então clicar em *Parameters*.

Nessa página, o primeiro e o último campo devem ser alterados. Caso tenha seguido o tutorial, o campo de *urlbase* deverá conter `http://SERVER_IP/bugzilla/`, e o campo de *cookiepath* deverá conter `/bugzilla/`.

Com isso o, *Bugzilla* estará completamente funcional.

## 5 Bibliografia

- Ubuntu Wiki
- Ubuntu Help
- nixCraft - Linux and Unix tutorials for new and seasoned sysadmin.
- Bugzilla Quick Start