

## Contents

- 1 Ajaxterm - Web Based SSH Terminal Client
  - ◆ 1.1 To Start / Stop
  - ◆ 1.2 To Config
  - ◆ 1.3 To Access (from Localhost)
  - ◆ 1.4 Config Ajaxterm and Apache
  - ◆ 1.5 Why Ajaxterm
  - ◆ 1.6 Larger Terminal Window

## Ajaxterm - Web Based SSH Terminal Client

If you are behind a firewall, and only have internet access through a restrictive proxy server, you will have to use a Web Based SSH terminal client. Note: the performance is severely below that of Putty. I suggest you exhaust all possibilities of using any available Socks and Http Proxies from within Putty.

Ajaxterm ( <http://antony.lesuisse.org/qweb/trac/wiki/AjaxTerm> ) is in the upcoming Debian Etch. It is an alternative to Anyterm (discussed [here](#)) and Mindterm (this is in Debian Sarge Stable, however runs as a java applet). **How-to below works with etch, lenny and debian squeeze**

```
apt-get install ajaxterm
```

### To Start / Stop

```
/etc/init.d/ajaxterm start|stop
```

### To Config

The config file is: `/etc/default/ajaxterm` . You can change the default port been used. Default=8022.

The ajaxterm Debian docs are located at: `/usr/share/docs/ajaxterm`

Depending on your setup etc. you may have to do the following:

```
vi /etc/ssh/ssh_config
//uncomment: PasswordAuthentication yes
```

Otherwise you may see the following error:  
**Permission denied (publickey,password).**

### To Access (from Localhost)

```
telnet localhost 8022
```

## Ajaxterm

Browse in an Internet Window to <http://localhost:8022>

Note: By default, ajaxterm will only bind to localhost, on port 8022, and thus will not be accessible from any other computer apart from localhost. Ajaxterm only runs in http and is not secure.

To help alleviate these problems, we configure Apache to serve out Ajaxterm over https.

## Config Ajaxterm and Apache

Put the following in an Apache vhost file:

```
<VirtualHost *:80>
    ServerName ajaxterm.website.com
    Redirect 301 / https://ajaxterm.website.com
    CustomLog /var/log/apache2/access.log combined
    ErrorLog /var/log/apache2/error.log
</VirtualHost>

<VirtualHost *:443>
    ServerName ajaxterm.website.com
    HostnameLookups Double
    CustomLog /var/log/apache2/access.log combined env=!dontlog
    SetEnvIf Request_URI "^/u" dontlog
    ErrorLog /var/log/apache2/error.log
    LogLevel warn
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/apache.pem

    ProxyRequests Off
    <Proxy *>
        AuthUserFile /srv/ajaxterm/.htpasswd
        AuthName EnterPassword
        AuthType Basic
        require valid-user

        Order Deny,allow
        Allow from all
    </Proxy>
    ProxyPass / http://localhost:8022/
    ProxyPassReverse / http://localhost:8022/
</VirtualHost>
```

Enable the proxy and proxy\_html modules in apache.

```
ln -s /etc/apache2/mods-available/proxy.conf /etc/apache2/mods-enabled/proxy.conf
ln -s /etc/apache2/mods-available/proxy.load /etc/apache2/mods-enabled/proxy.load
ln -s /etc/apache2/mods-available/proxy_http.load /etc/apache2/mods-enabled/proxy_http.load
```

or

```
a2enmod proxy proxy_http
```

That should do it. Check auth.log and apache logs etc.

## Why Ajaxterm

The reasons I have chosen Ajaxterm over Anyterm is because Anyterm is not yet in Debian Packages. Anyterm is difficult to compile correctly. When I did manage to compile anyterm, I recieved apache threading errors ( see [Anyterm](#) ).

Although Mindterm is in Debian Sarge, it will not be in etch, as the free version of Mindterm does not support SSH v\_2. Also Mindterm requires a user/client to have a JRE, and may possibly require port 22 access open (which is the main problem).

Ajaxterm, while slow, works quickly and effectively out of the box with little work. Again, I have Ajaxterm on a secondary box which is not mission critical.

## Larger Terminal Window

To get ajaxterm using a larger terminal window within the browser, the following files need to be edited:

```
vi /usr/share/ajaxterm/ajaxterm.html
window.onload=function() {
    t=ajaxterm.Terminal("term",100,40);
};
</script>
```

```
vi /usr/share/ajaxterm/ajaxterm.py
def __init__(self,width=100,height=40):
```

You may have to restart ajaxterm for the above to work.