

Guide - How to configure an apache load balancer



☆

Your Rating: ☆☆☆☆☆ Results: ★★★★★☆☆ 153 rates

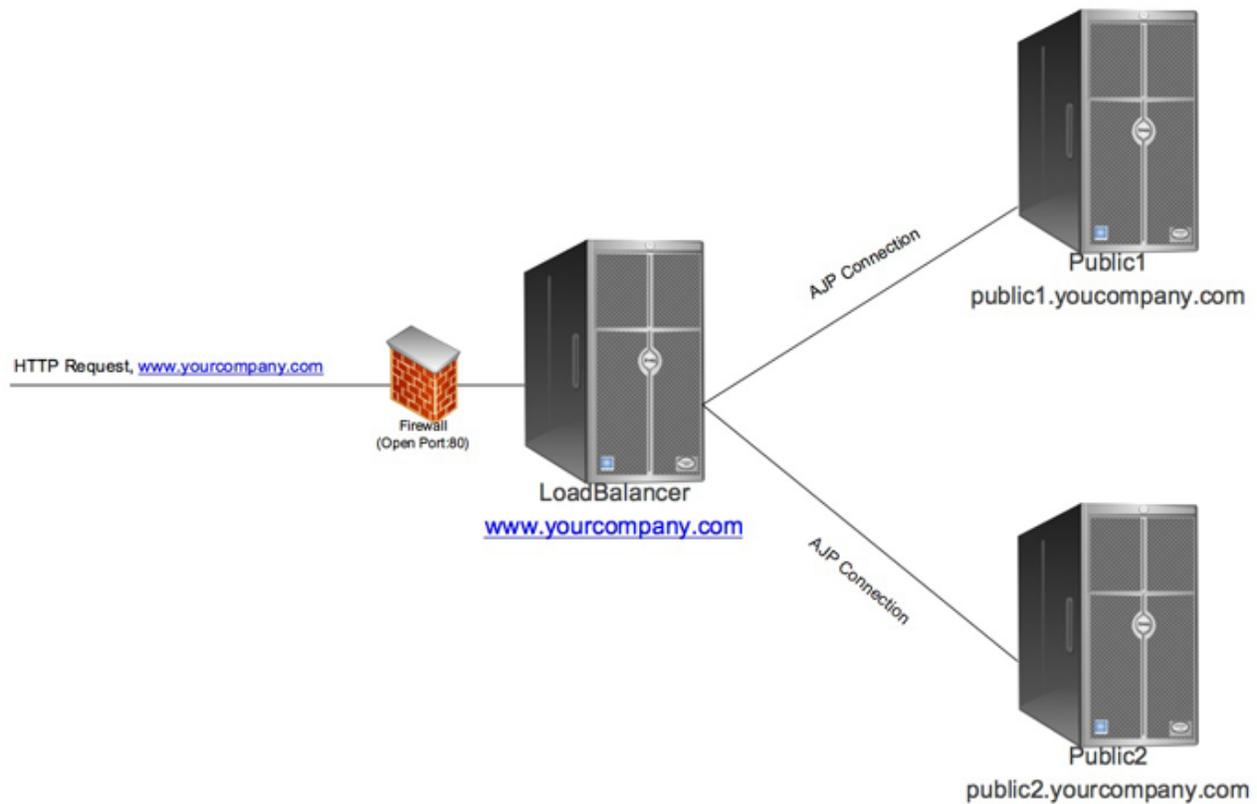
At a certain amount of traffic, or a certain need on availability, you might consider using multiple public instances. Most likely those instances are on different servers as well. This guide will illustrate how to setup a load-balanced system using three different servers, where one acts as the load-balancer (using apache for splitting the requests) and the two remaining servers host the Magnolia public instances.

Sticky session will ensure that one visitor generally will be handled by the same server over the lifetime of a session. If you want to access the public instances' adminCentral (or have public user registration or any other User Generated Content module installed) you want to use sticky sessions.

- [Server setup](#)
- [Load-balancer](#)
- [Public1 / Public2](#)
- [Done](#)

Server setup

The layout may look something like this (we will refer to these names through the rest of the guide).



Load-balancer

This server will handle all HTTP requests from site visitors. As you might see, this means even though you run a load balanced system, using only a single load balancer means you still have a SPOF (single point of failure). It is also possible to configure an environment where yet another server will act as the fail-over load-balancer if the first one fails, but this is outside the scope of this guide.

To set up our load-balancer, we use the Apache web-server and its modules `mod_proxy`, `mod_proxy_ajp` and `mod_proxy_balancer`. These are part of most of the Apache web-server distributions.

First, create a virtual host handling the requests for your domain: `www.yourcompany.com`

```

<VirtualHost *:80>
  ServerName www.yourcompany.com
  ServerAlias yourcompany.com

  DocumentRoot /Library/WebServer/Documents

  ProxyRequests Off

  <Proxy *>
    Order deny,allow
    Allow from all
  </Proxy>

  ProxyPass /balancer-manager !
  ProxyPass / balancer://mycluster/ stickysession=JSESSIONID nofailover=On
  ProxyPassReverse / ajp://public1.yourcompany.com:8009
  ProxyPassReverse / ajp://public2.yourcompany.com:8009
  <Proxy balancer://mycluster>
    BalancerMember ajp://public1.yourcompany.com:8009 route=public1
    BalancerMember ajp://public2.yourcompany.com:8009 route=public2
    ProxySet lbmethod=byrequests
  </Proxy>

  <Location /balancer-manager>
    SetHandler balancer-manager
  </Location>

  <Directory "/Library/WebServer/Documents">
    AllowOverride AuthConfig
  </Directory>

</VirtualHost>

```

We exclude the path `balancer-manager` from the proxy, since we can manage our balanced members with the `balancer-manager` tool. The `balancer-manager` is part of the `mod_proxy_balancer` module. To secure its path, create the folder and secure it. The setup may slightly differ on your server, so use the following as a starting guide.

```

mkdir /Library/WebServer/Documents/balancer-manager
htpasswd -c /etc/apache2/conf/security/users administrator
cd /Library/WebServer/Documents/balancer-manager
touch .htaccess
vi .htaccess

```

```

AuthName "secured"
AuthType Basic
AuthUserFile users

require valid-user

```

Public1 / Public2

Let's look at the relevant configuration here to set up the load-balancer. Most likely you will also have an Apache web-server installed on this machines, as for accessing the author instance if located on one of this servers with a nice URL. Here we suggest to use a single application server (Tomcat or JBoss) for hosting one public instance. Make sure the AJP Port is set correctly to what you have defined in the virtual host configuration of the load-balancer (8009 as the default value used here).

If you want to change the AJP Port of your application server, this can be done here.

Tomcat: `TOMCAT_HOME/conf/server.xml`

JBoss: `JBOSS_HOME/server/default/deploy/jbossweb-tomcat55.sar/server.xml` (depending on your JBoss version)

```
<Connector port="8009" protocol="AJP/1.3" (...)>
```

Now in the same file as we configure the AJP Port `server.xml` we need to configure the `jvmRoute` for sticky sessions working correctly. Use the name defined in the virtual host configuration on load-balancer, the route value here separately for the two servers.

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="publicXY">
```

Done

That's basically it. Now you can set your DNS entry of `www.yourcompany.com` to your Load-Balancer's IP address and enjoy the comfort and security of a redundant Magnolia installation. If one of the public Magnolia servers is failing, `mod_proxy` on your load-balancer will automatically detect this and stop serving requests to that server.

You can test this by stopping tomcat on one of the machines. Your load-balancer Apache webserver `error_log` will show something like

```
[Tue Jul 28 18:17:35 2009] [error] proxy: AJP: failed to make connection to backend: public1.yourcompany.com  
[Tue Jul 28 18:17:36 2009] [error] ap_proxy_connect_backend disabling worker for (public1.yourcompany.com)
```

Also you could access the balancer-manager with <http://www.yourcompany.com/balancer-manager> to manually disable a worker. The balancer-manager also offers you an easy way to set different load factors for your servers.

Feedback? Use the comment features!