- NAME
- FastCGI Deployment
  - Pros
    - Speed
    - App Server
    - Load-balancing
    - Multiple versions of the same app
    - Can run with threaded Apache
    - Widely supported.
  - Cons
    - More complex environment
  - Standalone FastCGI Server
- WEB SERVERS
  - Apache
  - nginx
  - lighttpd
  - Microsoft IIS
- AUTHORS
- COPYRIGHT

NAME

Catalyst::Manual::Deployment::FastCGI - Deploying Catalyst with FastCGI

FastCGI Deployment

FastCGI is a high-performance extension to CGI. It is suitable for production environments, and is the standard method for deploying Catalyst in shared hosting environments.

Pros

Speed

FastCGI performs equally as well as mod_perl. Don't let the 'CGI' fool you; your app runs as multiple persistent processes ready to receive connections from the web server.

App Server

When using external FastCGI servers, your application runs as a standalone application server. It may be restarted independently from the web server. This allows for a more robust environment and faster reload times when pushing new app changes. The frontend server can even be configured to display a friendly "down for maintenance" page while the application is restarting.

Load-balancing

You can launch your application on multiple backend servers and allow the frontend web server to perform load-balancing among all of them. And of course, if one goes down, your app continues to run.

Multiple versions of the same app

Each FastCGI application is a separate process, so you can run different versions of the same app on a single server.

Can run with threaded Apache

Since your app is not running inside of Apache, the faster mpm_worker module can be used without worrying about the thread safety of your application.

Widely supported.

FastCGI is compatible with many server implementations, not just Apache.

Cons

You may have to disable mod_deflate. If you experience page hangs with mod_fastcgi then remove deflate.load and deflate.conf from mods-enabled/

More complex environment

With FastCGI, there are more things to monitor and more processes running than when using mod_perl.

Standalone FastCGI Server

In server mode the application runs as a standalone server and accepts connections from a web server. The application can be on the same machine as the web server, on a remote machine, or even on multiple remote machines. Advantages of this method include running the Catalyst application as a different user than the web server, and the ability to set up a scalable server farm.

To start your application in server mode, install the FCGI::ProcManager module and then use the included fastcgi.pl script.

```
    $ script/myapp_fastcgi.pl -l /tmp/myapp.socket -n 5
```

Command line options for fastcgi.pl include:

```
    -d -daemon     Daemonize the server.
    -p -pidfile    Write a pidfile with the pid of the process manager.
    -l -listen     Listen on a socket path, hostname:port, or :port.
    -n -nproc      The number of processes started to handle requests.
```

See below for the specific web server configurations for using the external server.

WEB SERVERS

Any web server which supports FastCGI should work with Catalyst. Configuration recipes for well-known web servers are linked below, and we would welcome contributions from people deploying Catalyst on other web servers.

Apache

Catalyst::Manual::Deployment::Apache::FastCGI

nginx

Catalyst::Manual::Deployment::nginx::FastCGI

lighttpd

Catalyst::Manual::Deployment::lighttpd::FastCGI

Microsoft IIS

Catalyst::Manual::Deployment::IIS::FastCGI

AUTHORS

Catalyst Contributors, see Catalyst.pm

COPYRIGHT

This library is free software. You can redistribute it and/or modify it under the same terms as Perl itself.

syntax highlighting: no syntax highlighting ▼

120193 Uploads, 34929 Distributions
178154 Modules, 12986 Uploaders

hosted by YellowBot