As of the time of this writing (Thomas 2020-02-26T10:34:58Z), as a result of this study, I have concluded that openlitespeed is not ready for enterprise use, and even worse, it did not live up to its promise of making it easy to get a django website running. The documentation is poor, they seem to want config files to be written only by their panel so they left them undocumented, they assume nothing will go wong (but it did), and the panel comes up unsecure. As a pre-spun instance on GCP it only appears with Ubuntu.

**Review of docs**

<https://openlitespeed.org/kb/>

Their ‘knowledge base’ docs just explain how to put common stuff into their panel. There is little information that increases a person’s understanding of how litespeed works, how it interfaces with a django application, or even what is in their config files.

It might be that litespeed is not faster than other servers except when the cache feature is turned on, but we probably can’t turn it on, because, they say, it doesn’t work with dynamic websites.

**Server administration commands.**

These are specific to litespeed, and not to django of course. Start stop the sever etc.

<https://openlitespeed.org/kb/command-references-for-administration/>

/usr/local/lsws/bin/lswsctrl help

Graceful restart needed after changing configuration files:

/usr/local/lsws/bin/lswsctrl restart

**Configuration**

Configuration files may be in XML or plain text. This is set by command (see above section). Looks like they default to plain text.

This variable appears often, and is used on the panel:

$SERVER\_ROOT (on the rt-website-server-1 as of today: /usr/local/lsws)

**The panel**

<ip address>:7080

Before opening the panel the firewall must allow the port:

> ufw allow from 1.2.3.4 to any port 7080

or:

> ufw allow 7080

To remove the firewall rule:

> ufw delete allow 7080

The panel renders in the browser through plain http even after the security protocol is checked. Attempts to make it secure have just resulted in the panel not being accessible. The litespeed developer community did not respond to questions on how to do this. So I will attempt to use config files.

**Aps and Modules**

Aps are things like word press. Modules are things like PHP. They mention a module ‘ols1clk’ that handles a number of modules at once. Neither Django nor Python are mentioned in either the modules or aps docs. There is a place on the panel for listing these.

**Cache**

We are not using the cache feature.

… and that’s the whole of what is on the ‘knowledge base’.