

Basic Configuration & Management

Server Startup & Shutdown

Start OpenLiteSpeed:	Typically starts automatically on system boot. Check systemd or init.d scripts.
Stop OpenLiteSpeed:	Use the command: <code>service lsws stop</code> (on systems using systemd or init.d)
Restart OpenLiteSpeed:	Use the command: <code>service lsws restart</code>
Check Status:	Use the command: <code>service lsws status</code>
Graceful Restart:	In the WebAdmin console, this allows minimal downtime.
Configuration File Location:	<code>\$SERVER_ROOT/conf/httpd_config.conf</code>

WebAdmin Console Access

Access the WebAdmin console via a web browser using the server's IP address or domain name, followed by port <code>7080</code> .
Example: <code>https://your_server_ip:7080</code> or <code>https://your_domain.com:7080</code>
Default username is <code>admin</code> .
Initial password is the server's serial number, found in <code>\$SERVER_ROOT/serial.no</code>
It is highly recommended to change the default password immediately after logging in.

Basic WebAdmin Tasks

Adding a Virtual Host:	Navigate to <code>Virtual Hosts > Add Virtual Host</code> in the WebAdmin console.
Setting Document Root:	Within a virtual host's settings, configure the <code>Document Root</code> to point to your website's files.
Configuring Listeners:	Set up listeners to define which IP addresses and ports OpenLiteSpeed should listen on for incoming connections.
Enabling SSL:	Configure SSL certificates and enable HTTPS for secure connections under the virtual host settings.
Deploying a WordPress Site:	Use the built-in one-click WordPress installer for simplified setup.

Virtual Host Configuration

Virtual Host Basics

Virtual hosts allow you to host multiple websites on a single server. Each virtual host has its own configuration, document root, and settings.
Virtual host configuration files are located in <code>\$SERVER_ROOT/conf/vhosts/</code> .
Each virtual host directory contains a <code>vhconf.conf</code> file, which holds the specific settings for that virtual host.

Key Virtual Host Directives

<code>documentRoot</code>	Specifies the directory where the website's files are located. Example: <code>/var/www/example.com/public_html</code>
<code>domain</code>	Defines the domain names that this virtual host will respond to. Supports wildcards. Example: <code>example.com</code> or <code>*.example.com</code>
<code>enableWebDav</code>	Enables or disables WebDAV support for the virtual host. Example: <code>Yes</code> or <code>No</code>
<code>adminEmail</code>	Specifies email addresses for administrators who will receive error notifications. Example: <code>admin@example.com</code>
<code>errorLog</code>	Path to the error log file. Example: <code>/var/log/httpd/example.com_error.log</code>
<code>accessLog</code>	Path to the access log file. Example: <code>/var/log/httpd/example.com_access.log</code>

Context Configuration

Contexts define how OpenLiteSpeed handles requests for specific URLs or file types within a virtual host.
Common context types include: <ul style="list-style-type: none">Static Files: Serves static content like HTML, CSS, images, etc.External App: Proxies requests to an external application server (e.g., Node.js, Python).Script Handler: Executes scripts (e.g., PHP) to generate dynamic content.Rewrite: URL rewriting rules.
Contexts are defined within the <code>vhconf.conf</code> file using the <code>@context</code> directive.

Security & Optimization

Basic Security Measures

Keep OpenLiteSpeed up-to-date with the latest security patches.
Use a strong password for the WebAdmin console and change it regularly.
Enable ModSecurity for enhanced web application firewall protection.
Configure access control lists (ACLs) to restrict access to sensitive files and directories.
Disable unnecessary modules to reduce the attack surface.
Use SSL/TLS encryption to protect data in transit.

Performance Optimization

Enable Static File Cache:	Configure caching for static files (CSS, JavaScript, images) to reduce server load and improve page load times.
Enable Gzip Compression:	Compress responses to reduce bandwidth usage and improve page load times.
Use a Content Delivery Network (CDN):	Distribute static content across multiple servers to improve performance for users in different geographic locations.
Optimize Database Queries:	Ensure that database queries are efficient to reduce database load and improve application performance.
Tune PHP Settings:	Adjust PHP settings (e.g., memory_limit, max_execution_time) to optimize performance for PHP-based applications.
Use LiteSpeed Cache:	Leverage the LiteSpeed Cache plugin (for WordPress and other platforms) for advanced caching features.

Rate Limiting

Implement rate limiting to protect against denial-of-service (DoS) attacks and brute-force attacks.
Configure rate limiting rules based on IP address, URL, or other criteria.
Use the WebAdmin console to define rate limiting settings under the Security section.

Troubleshooting & Logging

Common Issues

500 Internal Server Error: Check server error logs for detailed information about the cause of the error.
403 Forbidden Error: Verify file permissions and access control settings.
404 Not Found Error: Ensure that the requested file or directory exists and that the document root is configured correctly.
Website Not Accessible: Check DNS settings, firewall rules, and listener configurations.
Slow Performance: Analyze server resource usage, optimize caching, and tune PHP settings.

Log File Locations

Server Error Log:	<code>\$SERVER_ROOT/logs/error.log</code>
Virtual Host Error Log:	<code>\$SERVER_ROOT/vhosts/<vhost_name>/logs/error.log</code>
Virtual Host Access Log:	<code>\$SERVER_ROOT/vhosts/<vhost_name>/logs/access.log</code>
WebAdmin Console Log:	<code>\$SERVER_ROOT/admin/logs/error.log</code>

Debugging Tools

Use the <code>lswsctl</code> command-line utility for managing OpenLiteSpeed.
The WebAdmin console provides various tools for monitoring server performance and troubleshooting issues.
Enable PHP error reporting to display errors in the browser or log them to a file.
Use debugging tools like Xdebug for advanced debugging of PHP applications.