

Acquia Hosting Optimization Checklist

This checklist provides Drupal administrators with a series of review points to ensure resources are being effectively used, and that deployments on Acquia hosting perform with optimal efficiency. In the event that customers do not possess all of the necessary expertise to conduct the remediations in full, Acquia can provide a brief services engagements to ensure hosting preparation is effective and complete.

Note: This checklist is ordered by priority of impact.

☐ Drupal caching: pages, blocks, views, etc.

Explanation:	Employing Drupal's built-in page caching system can dramatically reduce database load by reducing the number of queries required to build pages for anonymous users, and to cache specific data for authenticated users. Caching should be enabled for Pages, Blocks, Panels, Views and any custom data.
Discovery:	Review the performance settings page.
Remediation:	At minimum, enable database caching for pages. Acquia Hosting runs best when using the Pressflow release of Drupal which allows pages to be cached with Varnish.

☐ CSS & JavaScript aggregation

Explanation:	Enabling aggregation in Drupal decreases HTTP requests per page.
Discovery:	Check with Firebug or YSlow!
Remediation:	Enable aggregation in Drupal's Performance settings page.

☐ PHP memory usage

Explanation:	The amount of memory consumed when Apache and PHP build pages limits the number of concurrent pages that any single server can generate.
Discovery:	Use the Devel module's performance logging capabilities to display page timer and memory usage.
Remediation:	Acquia Hosting uses APC, for the opcode cache. Disable unused modules.

❑ MySQL queries per page

Explanation:	Overly complex pages can generate an enormous number of MySQL queries, which can slow page building and limit scalability.
Discovery:	Use the Devel module's performance logging capabilities to display page timer, query log.
Remediation:	Decrease page complexity, optimize queries and utilize Drupal's cache API to store custom data for faster reuse.

❑ MySQL slow query logs

Explanation:	MySQL's slow query logs should be reviewed to discover database queries that are taking excessive time to complete, which can indicate improper SQL query construction or PHP code.
Discovery:	Tools such as Maatkit can process query logs to identify and report slow queries.
Remediation:	Optimization of Views queries and custom code.

❑ Core hacked?

Explanation:	Changes to Drupal's core PHP, JavaScript and CSS files can adversely affect performance and behavior of the site, and seriously degrade the ability to support and maintain the site post-launch.
Discovery:	Acquia's 'drush check' script will examine the site's document root(s) to find any core files that have been modified.
Remediation:	Suggest methods to achieve goals without hacking core through proper use of Drupal's APIs.

❑ Module selection and activation

Explanation:	Drupal's page building performance can be dramatically affected by the choice of modules used, custom module code and the number of modules enabled.
Discovery:	Review the Modules page, and potentially review custom module code.
Remediation:	Disable unneeded modules to prevent inclusion by PHP at runtime.

❑ 3rd party scripts and styles

Explanation:	Failing to use Drupal's drupal_add_css and drupal_add_js API's can lead to extra HTTP requests, and can cause inconsistent behavior of user interface elements.
Discovery:	After enabling CSS and JavaScript aggregation, check with Firebug or YSlow!
Remediation:	Identify any scripts that are being included directly from themes or modules via HTML. Add via proper hooks.

□ Use of asynchronous requests

Explanation:	Using asynchronous JavaScript (AJAX or AHAH) can slow page build performance to end user, and if used without care can cause server performance to be degraded.
Discovery:	Check with Firebug and YSlow, identify requested files that are not cacheable.
Remediation:	Ensure that Drupal-generated feeds (i.e. from Views, RSS) are cached at both database and Varnish. Limit number of overall AJAX calls on particular pages.

□ Gzip compression

Explanation:	Enabling compression of files decreases bandwidth utilization and improves user perception of site performance.
Discovery:	YSlow! can report page elements that are not being served compressed.
Remediation:	Enable compression using Apache, Varnish or PHP. This is available in Acquia Hosting.

□ Core search in use

Explanation:	Drupal's core search performs queries against the MySQL database, which for high traffic sites can stress the database server.
Discovery:	Check the Drupal Modules page.
Remediation:	Use Acquia Search, which removes search load from MySQL and offers feature and usability improvements.

□ Enabled Apache & PHP modules

Explanation:	Apache and PHP modules to enable must be selected with care to avoid consumption of excess memory and CPU resources.
Discovery:	Review phpinfo() output and review httpd.conf and any .htaccess files.
Remediation:	Acquia Hosting has an optimized list of Apache and PHP modules. If your site needs an additional set of modules let our support team know.

□ Representative page inspection

Explanation:	Pages that aggregate large amounts of data or require complex rendering or queries should be examined to determine any performance deficits.
Discovery:	Examine key pages from the client side with Firebug and YSlow!, and from backend with Devel, top, etc. Detailed direct analysis can be done with PHP profiling tools like X-debug and Webgrind.
Remediation:	Varies.

□ Load testing

Explanation:	Load testing realistic use cases should demonstrate how sites will perform in production, and indicate areas for adjustments to code to improve performance and scalability.
Discovery:	Acquia recommends load testing generation services from Soasta, or with tools like JMeter. Acquia can provide analysis of load tests on your Acquia Hosting infrastructure.
Remediation:	Varies.

□ Views query performance

Explanation:	The views module is powerful and very useful in site development. Through mis-configuration, Views queries can be created which cause excessive load on the database.
Discovery:	"EXPLAIN SELECT ..." on queries generated by views can reveal inefficient queries.
Remediation:	Review database slow query logs to see if any views need optimization.

□ Monitor server load

Explanation:	Excessive resource consumption limits scalability of sites, and can require an increase in hardware and support costs.
Discovery:	Monitor server CPU, memory and disk I/O load with tools such as Nagios, top, etc. A load test may be required to accurately reveal issues.
Remediation:	Acquia's load testing analysis service will report how your infrastructure responds to a load generated by your team or Acquia's load testing service partners.

□ Database engine

Explanation:	Use of InnoDB improves database performance and reliability.
Discovery:	"SHOW TABLE STATUS" query will identify engine used for all database tables.
Remediation:	Acquia Hosting import script can make this configuration change when data is imported. InnoDB should be used with performance improvements in Pressflow.

Disclaimer: This is a checklist with general recommendations. Your Drupal site is a custom site designed to meet your organization's business goals and specific recommendations may not apply. Where optimizations are not possible or not preferred additional hardware and support costs may be necessary.