

Monitoring Drupal using Nagios integration

Khalid Baheyeldin

February 19, 2009

<http://2bits.com>

The logo for '2bits' features the number '2' in a bright green color and the word 'bits' in a dark grey color. The letters are stylized with a 3D effect, including a drop shadow and a slight reflection below them. The '2' is a simple, rounded shape, while the 'bits' are in a sans-serif font with a slight curve at the top of the 'i' and 't'.



Agenda



- Why monitoring?
- Nagios
 - What is Nagios?
 - What can it monitor?
- Nagios / Drupal Integration
 - Base features
 - Performance Integration
 - API
- Discussion





Why Monitoring?



- Murphy's Law
- Know of issues as they happen
- Act on them sooner
- Minimize downtime





Before Monitoring



- We all start with emails from cron scripts
 - Then it does not scale well ...
 - Too much noise in your inbox ...





Nagios



- GPL licensed
- Mature
- Monitoring platform
- Alerts by email, pager, and even IM (Jabber)
- Multiple alerts to different people
- Soft and hard states (makes sure it is an error)
- Escalation
- Can manage thousands of servers
- Web site at <http://www.nagios.org/>





Nagios states



- Each monitored “item” sends a state
- Represented by colors
 - **OK**
 - **Warning**
 - **Critical**
 - Unknown





Drupal



- How can Nagios help Drupal?
- Answer: New Module
 - <http://drupal.org/project/nagios>
 - Drupal 5.x and 6.x





What is monitored?



Many things in Drupal ...





Core and Modules



- New releases, as they are available on drupal.org
- Requires the update module in Drupal 6.x
- Or `update_status` on Drupal 5.x
- Core releases
- Contrib
- Security releases!





Cron



- Cron is essential for normal running of a Drupal site
- Cleans logs
- Checks for releases
- Many other things
- We check that cron is running regularly
 - 60 minutes default, can be adjusted





Permissions



- “update.php” should not be accessible to other than user 1
- “settings.php” should NOT be writable by web server
- “files” directory should be writable by the web server





Database schema



- When you install new modules, they may change the database structure (schema)
- Ever installed a new module and forgot to run `update.php`?





Requirements



- We monitor anything that goes in:
 - /admin/{logs,reports}/status





Performance Logging



- Started as an independent project by 2bits.com
- Now part of Devel (5.x, 6.x and 7.x, latest in -dev)
- Aims at collecting info for analysis of performance
 - Which pages use most queries
 - Which pages use most time to generate
 - Average and maximums
 - Logs to database (dev/test) or APC (ok for live sites)
 - Can be combined with stress testing (ab/siege)





Performance & Nagios



- Performance Logging initially done just like other things (i.e. alerts)
 - Page execution time
 - Memory per page
- Proved to be too noisy
 - Too many emails for transient problems
 - “Crying wolf”
- Planning a new approach, with no alerts on performance, but providing the data





Configuration



- Need to get Nagios running first
 - Not trivial, but not rocket science
- Documentated (for Nagios 2.11) in README.txt
- Need to install a Nagios Plugin (included with the module)
- Need a unique ID, to keep away prying eyes





API



- Simple and powerful API
- A couple of hooks
- Documented in README.txt
- Example in devel/performance.module
- You can add alerts on whatever you want in Drupal





Demo



Let us see it in action ...





Discussion



Questions?

Comments?

