

Tier-2 System Administration: A Comprehensive Approach

John Bland (jbland@hep.ph.liv.ac.uk), Rob Fay (fay@hep.ph.liv.ac.uk),
Stephen Jones (sjones@hep.ph.liv.ac.uk), Mark Norman (m.j.norman@liv.ac.uk)

Management

With focus on the technical rather than the human aspects, this covers the tools and procedures used to administrate infrastructure, including definition and configuration of O/S and services.

- Provision and Configuration: Puppet, for package management, service configuration and control. Through Puppet extensions and external commands all aspects of system configuration can be controlled.
- Package Management: Yum. At Liverpool we also mirror regularly used repositories locally for performance and control reasons, as well as maintaining a local repository for site packages.
- Middleware Configuration: Yaim, Vomssnooper.
- Code Management: Subversion, Gitlab (local repositories and GitHub)

Communication

Communication is essential for coordinating and directing effort (both locally at a site and across the global community).

In conjunction with documentation, it provides the means for discussion, decision-making, and dissemination of information.

- Email: Reports, communication from the global community including required updates, tickets and other issues.
- Chat: IRC, Skype, Seevogh, Vidyo
- Issue-tracking: RT (bug track, tickets, workflow)
- Face to face: Discussions, formal meetings and informal conversations. Minutes and follow-ups (e.g. via email) to ensure information is not forgotten or lost.

Documentation

This ensures that information necessary to understand and hence administer the site is available both now and in the future.

The process draws upon the outputs of site-related communication and management, but requires additional infrastructure and procedures to render it comprehensive. It is all too common for vital documentation to be left until a later that never comes ("I'll do it when it's finished").

Local procedure at Liverpool aims to ensure a basic level of documentation is achieved in-line with most activities (comments in Puppet configuration, code, commits) and, by providing an easily accessible iterative interface for code, encourages documentation to be kept in pace with activity.

- Contributing tools: E-mail, RT, Puppet, Subversion, Gitlab, logs
- Document management system: Twiki

Introduction

It is easy to take an ad hoc approach to site administration, employing whatever tools seem appropriate at any given moment.

However, as a long-term strategy this can lead to constant "fire-fighting" resulting in loss of efficiency in terms of both system and human resources.

Inevitably this will result in periods of downtime. While such fire-fighting cannot be avoided entirely, it is by employing a comprehensive framework approach that we at Liverpool have sought to keep such disruption to a minimum.

Goals

The comprehensive approach should ensure that all administration activity and consequent effects on other activities is managed, not just by the individual but by all those with collective responsibility.

To this end, it is necessary to ensure an appropriate set of tools and procedures are in place to cover the following areas:

- Communication
- Management
- Monitoring
- Documentation
- Control
- Protection

Principles

To realise these objectives, the procedures and tools at our disposal need to be employed in the appropriate manner.

- Test changes (upgrades, new installations etc.) before full deployment where practical.
- Ensure changes have a rollback path where practical.
- Address fundamentals first; only optimise where time allows.
- Ensure that tools and procedures are sustainable in respect of demands on human and technical resources.
- Try to avoid becoming locked in to a solution; software, hardware or vendor

Monitoring

Monitoring covers the gathering of data, such that it can be visualised when required, and such that it can be used to generate appropriate alerts, allowing appropriate action to be taken to avoid downtime and/or reductions in efficiency.

Monitoring at Liverpool is roughly divided between the broad gathering of metrics, which can be observed to detect problems and known issues, and targeted testing which can raise alerts when necessary, both for general issues (nagios) and specific purpose (testnodes, for worker node fitness).

It is also necessary to monitor the environmental conditions on both the system and site level to detect e.g. overheating resulting from broken cooling.

- Metrics: Ganglia, Graphite
- Monitoring, Testing, Alerting: Nagios, Testnodes

Protection

Protection covers the measures taken to ensure the site's robustness in the face of failures and disasters. This includes both initial design and specification to ensure high availability, along with security procedures and backups.

Design and specification includes, where practical, putting in redundancy on a supply level (e.g. UPSes on two electrical feeds, network resilience) system level (e.g. disks, PSUs), and site level (e.g. fail-over servers).

Backups are taken regularly, and kept both off-system and off-site. The procedures used for this should inform recovery procedures (see: Documentation).

- Security: iptables, logwatch, pakiti
- Backups: rdiff-backup for space efficient incremental backups, off-site backups

Control

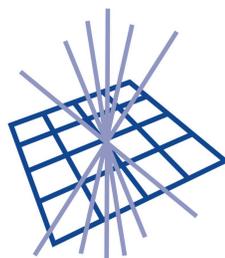
Control provides the means by which actions are ultimately carried out on the site, both directly by administrators, and indirectly, e.g. arising from monitoring and managements tools.

On anything beyond the smallest site, it is not practical to manage systems purely from the physical console. Consequently it is necessary to have the tools (together with the information, see: documentation) to manipulate the site hardware and environment as necessary. Given the inherent power of these tools, they must be clear to use (well-developed UI) and secure.

- Low-level system control: IPMI
- General system access: KVMoIP, SSH
- Parallel execution: Locally developed 'parallel' script (general alternatives would include ClusterShell)
- Other tools UIs: e.g. Torque's pbsnodes command



UNIVERSITY OF
LIVERPOOL



GridPP
UK Computing for Particle Physics