1 Evaluation of Existing IT Infastructure Monitoring Systems

This section will review current IT infrustructure monitoring systems and evaluate them on several points as follows:

- Support for timeseries monitoring and real time alerting
- How they can be configured to monitor custom metrics
- How are alert thresholds defined
- How configuration and custom code is delivered to nodes (if requried)
- How the user configures the system
- How dependencies are handled

1.1 Nagios

Timeseries monitoring and real time alerting Nagios is primarily focused at real time alerting and therefore has very little in the way of timeseries monitoring. Additional plugins are available which can be used to graph metrics over time but these cannot be used to make decisions on the status of a given system or service. All that is supported in terms of alerting on historical data is to refrain from alerting until a given condition has been observed in the previous n checks, there is no support for alerting based on trends in historical data. Supports basic display of changes in state of hosts/services over time but not individual metrics.

Support for custom metrics Nagios has support for custom metrics through the NRPE (Nagios Remote Plugin Executor) plugin. These plugins can be any sort of executable which prints out a message to represent the data read as well as a specific exit code which defines the status, for example "OK", "Critical" .etc

Alert threshold definition Thresholds for NRPE agents must be set on the remote server itself. These thresholds are passed into the remote plugin as an argument when it is executed and are used internally by the script to output the approprate alert level.

Code/Config delivery to nodes Nagios does not have any in built functionality to distribute configuration files or plugin code to remote nodes. In order to automate this, additional software such as Puppet would be required.

How the user configures the system Configuration for Nagios is primarily managed through text files stored on disk. Third party configuraiton tools are available to allow the system to be configured through a web interface. Configuration lives on both the Nagios server as well as on the machines being monitored.

How dependencies are handled Rigid tree - No way to define that a service/host is dependant on a given host OR another host being available. This reduces its usefulness in modern networks where redundancy and failover is commonplace. These are defined in config files that live on the Nagios server.

1.2 Icinga 2

Timeseries monitoring and real time alerting Like Nagios, Icinga's primary focus is around real time alerting however it has now introuced support for graphing performance metrics. No support for alerting based on trends in historical data beyond Nagios's idea of a state change changing to HARD once it has been observed n times.

Support for Custom Metrics Custom metrics can be written as scripts in any language as long as they echo a message to describe the status of what they are checkign and use a certain exit code to define the status e.g. "OK", "CRITICAL", "WARNING".etc.

Alert threshold definition Thresholds are passed into the check commands as command line arguments. Therefore they are defined on the machines being monitored individually.

Code/Config delivery to nodes Icinga 2 does not have any built-in mechanism to distribute code and config files to remote hosts. In order for this to be achieved, additional software such as Puppet would be required.

How the user configures the system Configuration is managed through configuration files stored on disk. Configuration files exist both on the Icinga server as well as the nodes being monitored.

How dependencies are handled Same as Nagios with a rigid tree. No way to define a host/service as being dependent on one of several different machines as is common in modern environments with redundancy/failover systems.