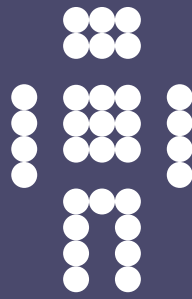


CFEngine



CFEngine Nova Technical Supplement

CFEngine Enterprise Documentation
for version 2.1

CFEngine

Table of Contents

1	Introduction to CFEngine Nova.....	1
1.1	What are enterprise versions of CFEngine?	1
1.2	What's new in Nova 2.1?	1
1.3	About the CFEngine architecture	2
1.3.1	CFEngine is agent based software.....	2
1.3.2	Single point of coordination	2
1.3.3	Requirements and scalability of CFEngine Nova	4
1.3.4	Phases of the server life-cycle	4
1.4	Nova Commercial Enhancements	5
1.4.1	Productivity enhancements	5
1.4.2	Functional enhancements.....	6
1.4.3	Reporting enhancements	6
1.4.4	Documentation enhancements.....	7
1.4.5	Knowledge Management in Nova	7
2	Mission Portal	9
2.1	Mission Status.....	10
2.2	Mission Engineering	10
2.3	Mission Planning	12
2.4	Mission Library.....	13
2.5	Reports in the Mission Portal.....	13
2.6	Should monitoring and configuration be separate?	14
2.7	CMDB and CFEngine.....	14
2.8	Decentralized data collection in CFEngine.....	14
3	Business Integration	17
3.1	Business goals and the Service Catalogue	17
3.2	Business value reports.....	18
3.3	Simple insight into IT compliance.....	19

1 Introduction to CFEngine Nova

CFEngine Nova is a commercially licensed version of the core CFEngine software¹ with enterprise library extensions. All of the documentation for CFEngine 3 applies to CFEngine Nova. This document is a supplement describing features particular to CFEngine Nova.

The aim of CFEngine Nova is to offer a knowledge-enhanced framework for configuration management that goes beyond mere technical configuration to support the needs of businesses. Features include compliance management, reporting and business integration, and tools for handling necessary complexity. CFEngine Nova has features to support Cloud Computing for public and private clouds, as well as greater integration facilities with database resources.

You should use this brief guide to CFEngine Nova's features in concert with the CFEngine Concept Guide and CFEngine Reference Manual, available from the CFEngine website, or in the 'docs' directory of your Nova hub installation.

1.1 What are enterprise versions of CFEngine?

CFEngine Enterprise versions contain features designed to both extend and simplify the use of CFEngine in enterprise scenarios. This includes quick setup, business alignment features and improved self-documentation, ultimately providing a single framework for self-healing, hands-free automation with integrated knowledge management. Each extended feature has been carefully designed to meet a specific need in the server life-cycle, replacing cumbersome or insecure technologies currently available in datacentre products. CFEngine Nova adds the following capabilities:

- Scalable management of complex, federated environments.
- Powerful Web-based graphical user interface
- Lightweight reporting engine.
- Distributed orchestration of tasks.
- Virtualization control (using libvirt).
- Simplified policy writing with Content-Driven Policies.
- Integrated Knowledge Management.
- Bundle managed services.
- Provides a simple window onto IT Operations for Business.

Of course, as a Nova user, you have access to all of the features of the CFEngine Community Edition, and access to its on-line and community resources.

1.2 What's new in Nova 2.1?

CFEngine Nova 2.1 is a major upgrade of CFEngine's commercial software, including the latest community core and many improvements to the language interface. Some of the major features in Nova 2.1 include:

- A new browsable Web interface (Mission Portal) utilizing client-side Javascript, allows you to see policy and state side by side.

¹ Major version 3

- Finder, viewer, editor model for application interface.
- Expanded monitoring on Linux and Solaris (cf-agents will report more basic data, such as load and cpu, "vital signs").
- Support for multiple release environments, for example staging and production from one Hub.
- External authentication for Mission Portal (LDAP, OpenLDAP, Active Directory)
- User class system with individual accounts, logins and management (Admin, Developer, Manager, Faculty, etc.).
- Collaboration tools: Store notes, publish messages/status per user account, activity log.
- Searchable reporting and analytics.
- Improved Copernicus Knowledge Map, providing searchable on-line documentation, integrated with the policy browser.
- Simplified management through Content-Driven Policies.
- Improved data transportation protocol efficiency (not backwards compatible but will fallback to legacy protocol if not supported on both sides)
- Save searches

In addition to these features, Nova 2.1 is built around our extensive bug-reporting and self-diagnostic test suite that has allowed us to improve the reliability and functionality of the software for the most demanding environments.

1.3 About the CFEngine architecture

1.3.1 CFEngine is agent based software

CFEngine is agent based software. It resides on and runs processes on each individual computer under its management. That means you do not need to grant any security credentials for login to CFEngine. Instead, for normal operation, CFEngine runs in privileged 'root' or 'Administrator' mode to get access to system resources and makes these available safely to authorized inquiries.

A CFEngine installation is thus required on every machine you want to manage: client and server, desktop or blade. Typically, you will single out one machine to be a *policy server* or *hub*. In very large networks of many thousands of machines, you might need several policy servers, i.e. several hubs.

1.3.2 Single point of coordination

The default CFEngine Nova architecture uses a single hub or policy server to publish changes of policy and to aggregate knowledge about the environment, but you can set up as many as you like to manage different parts of your organization independently. The CFEngine technology is not centralized by nature. Most users choose to centralize updating of policy and report aggregation for convenience however.

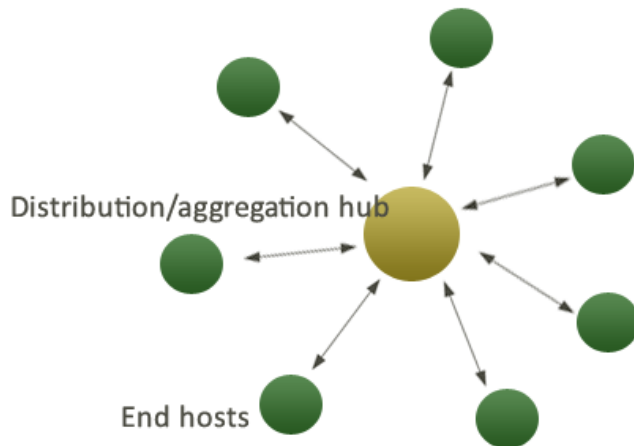


Figure: A policy server or 'hub' is implemented in CFEngine Nova as a simple solution that will scale for most sites 'out of the box'.

If you operate CFEngine Nova in its default mode, the hub acts as a server from which every other client machine can access policy updates. It also acts as a collector, aggregating summary information from each machine and weaving it into a knowledge map about the datacentre.

For a single hub configuration, the figure below shows a normal process approach to managing policy. Policy is edited and developed at a Policy Definition Point, outside of normal production environment. This can be done using the specialized editor embedded in CFEngine Nova, or it can be done using any text editor of your choice.

Edits are made in conjunction with a version control repository², which should be used to document the *reasons* for changes to policy³. When a change has been tested and approved, it will be copied manually to the policy dispatch point on one or more distribution servers. All other machines will then download policy updates from that single location according to their own schedule.

² CFEngine supports integration with Subversion through its Mission Portal, but any versioning system can of course be used.

³ CFEngine and version control will document *what* the changes are, but what is usually missing from user documentation is an explanation of the reasoning behind the change. This is most valuable when trying to diagnose and debug changes later.

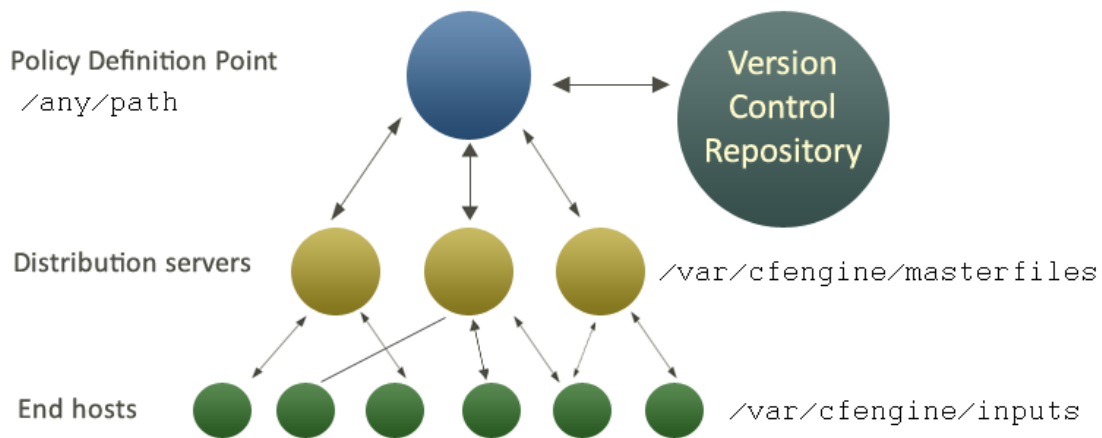


Figure: Policy coordinated from a central root location is implemented in a distributed manner at every leaf node.

1.3.3 Requirements and scalability of CFEngine Nova

The default architecture and configuration skeleton is expected to scale to a few thousand hosts with a dedicated policy hub. Your hub machine should have at least 2 GB of memory and a modern processor. You will need about 2 MB of disk storage for every machine under CFEngine's management.

CFEngine will scale to very large systems, with tens of thousands of machines. This is only possible because of its decentralized agent-based operation. CFEngine encourages the federation of policy in complex environments, so that local domain experts manage what they know and baseline policies can be handled generally for all. We also encourage a practice of small adjustments, as opposed to large risky redesigns of operational policy.

CFEngine's recommended practice is to encourage small incremental changes to policy, not to save up changes into 'big project roll-outs'. This strategy lowers the risk of error and offers improved scalability.

CFEngine will operate autonomously (hands-free) in a network, under your guidance. If your site is large (thousands of servers) you should spend some time discussing with CFEngine experts how to tune this description to your environment as *scale* requires you to have more infrastructure, and a potentially more complicated configuration. The essence of any CFEngine deployment is the same.

1.3.4 Phases of the server life-cycle

The four phases in managing systems, summarized in the Build, Deploy, Manage, Audit (BDMA) framework originated with a model of system management based on transactional changes ('roll-out'). CFEngine's conception of management is some different, as transaction processing is not a good model for highly distributed systems, but we can use this template to see how CFEngine works differently.

Build CFEngine is often used together with other tools to build new machines, whether virtual or real. You integrate it into the 'kickstart' or 'jumpstart' process, using PXE network booting, etc, by including the CFEngine software in the initial build and executing CFEngine once at the end of the installation.

To complete the installation of your system, you create a set of promises about the system state, or policy-template of proposed promises. When systems keep these promises, they will function seamlessly as planned.

Deploy Deploying software or changes really means implementing the policy that has been decided. In CFEngine you simply publish your proposed policy and the machines can adjust accordingly. Each installed agent is capable of implementing policies and maintaining them over time without further assistance.

Manage Once a decision is made, unplanned events will occur. Such incidents usually set off alarms and humans rush to make new transactions to repair them. In CFEngine, the autonomous agent manages the system, and you only have to deal with rare events that cannot be dealt with automatically.

Audit

Promises are assured by design in CFEngine and maintained automatically, so the main worry when using CFEngine is whether or not you have coded conflicting intentions. The compliance of machines with the policy template is easily measured by CFEngine, using its model-oriented approach. This can be viewed from the web-browsable interface.

1.4 Nova Commercial Enhancements

The enhancement provided by CFEngine Nova fall into a number categories:

- Business transparency and reporting enhancements, providing insight into IT operations for both Business and IT. CFEngine Nova connects the dots between high level business goals and low level configuration, supporting Knowledge Management for ITIL processes.
- Productivity enhancements – easier getting started and making changes.
- Continuity and repair related enhancements.
- Full integrated Knowledge Management.
- Special support for operating system features and virtualization, such as Solaris zones and the libvirt interface.
- Native support for the Windows platform, with access to the configuration of the Windows registry, processes and file access control lists.

1.4.1 Productivity enhancements

Features that are designed to make it easier to work with CFEngine on a day to day basis.

- Simple automatic installation (one command per system)
- A template library of managed services for common platforms.
- Auto-analyzing system knowledge console
- Policy mining and semantic knowledge representation
- Auto-classification of systems
- Performance and Service Level reports
- On- and off-line syntax look-up.
- Integration with LDAP directory services.

The most important productivity tool is the CFEngine Mission Portal, which contains an interface for browsing Mission status and interacting with Mission planning and engineering. Get an overview of compliance status and business value, see the current state of system repair, plan and make policy changes, or browse the Knowledge Map to increase your understanding of your system and policies.

A library of managed services for common Linux operating systems is provided as part of the knowledge base. This makes setting up specialized service nodes a simple process of matching hosts to services from a required list.

The ability to look up syntax on the fly, from the web or the command line, can save experienced users considerable time. Using the tools for database interaction, the process of setting up a CFEngine knowledge base at your organization is fully automated and you only have to think about local customizations you want to make.

LDAP query functions have been added for integration with LDAP services or Active Directory information, allowing a single point of definition for system lists and identity management.

1.4.2 Functional enhancements

- Redundant hub functionality for mission critical operations
- Database verification and editing including SQL (Mysql,Postgres)
- MS Registry management and repair
- File-system ACL security management (Linux, Windows, Solaris)
- Extended monitoring probes and system classification capabilities.

These include the ability to interact with and repair popular SQL databases (currently MySQL and PostgreSQL), as well as embedded system databases such as the Microsoft System Registry, defining, validating, scanning or repairing their tabular structure and the data within them.

- Access remote CFEngine variables (like a simple directory service)
- LDAP directory service integration
- Role based access control on policy execution by class.

Access Control List (ACL) support for Linux is now added for pinpoint accuracy in file permission security. CFEngine Nova's ACL support includes a completely new generic CFEngine model for ACLs that will translate across multiple platforms so that users can as closely as possible translate identical requirements across multiple platforms with different implementations. Native ACLs are also supported.

Fault tolerant features have been added in Nova to functions for retrieval of data from network connections. It is risky to rely on data from a network when configuring hosts. If the network connection could fail, erroneous data might be written to a local configuration. Nova adds local caching to network results and works opportunistically to provide the latest known values; however, look-up functions, such as LDAP retrieval of `remotescalar` data will not fail.

1.4.3 Reporting enhancements

Nova can provide a wider range of system reports about information collected by CFEngine on performance, security and state. This includes the ability to perform custom system discovery, and log the data into a variety of special reports.

Discovery and measurement promises are made through CFEngine's lightweight custom monitoring capabilities.

- HTML, XML, CSV, PDF report generation
- Service and performance level reports
- Reliability reports
- Deep history analysis and visualization, through knowledge base.
- Policy dependency and impact analysis reporting

Information can be extracted in HTML, XML, PDF, and text formats (e.g. CSV) for easy integration with other presentation tools, or for direct viewing through the knowledge console.

As part of policy-writing, CFEngine Nova allows you to track dependencies on policy items. The knowledge agent can take this information, analyze it and present it as part of an overview of the system (see the chapter on Knowledge Management for more information). This enables virtual impact analyses to be inspected.

1.4.4 Documentation enhancements

Additional documentation is provided for Nova users through program built-in help, additional manuals and integration of syntax information into the CFEngine Knowledge Console.

An annoying aspect of any software is the need to browse through a manual to find quick answers to questions. Ideally one would only look at manuals during a learning phase; thereafter we want to see examples and summaries of what we basically already know but cannot keep in our heads.

System administrators often prefer to work in the command line and find the need to go to a manual a distraction. Indeed, if the network is non-functional or we are working off-line, instant command line help is a great bonus. Other users prefer the point and click of a familiar web interface. CFEngine Nova provides both these options to users to provide quick answers.

Of the two, a web interface is clearly the most flexible. Far more information can be browsed on the web than is practical with a simple text interface. However, of the two, the command line interface is by far the fastest way to get answers to simple questions of syntax. CFEngine's knowledge agent knows CFEngine's syntax tree and can summarize it at the keyword level.

```
atlas$ ./cf-know --syntax link_from
Constraint link_from (of promise type files) has possible values:

  link_from ~ defined in a separate body, with elements

  source           ~ ()
  link_type        ~ (symlink,hardlink,relative,absolute,none)
  copy_patterns    ~ ()
  when_no_source   ~ (force,delete,nop)
  link_children    ~ (true,false,yes,no,on,off)
  when_linking_children ~ (override_file,if_no_such_file)

Description: A set of patterns that should be copied and synchronized instead of linked
```

1.4.5 Knowledge Management in Nova

The future of datacentre management lies in a more complete model of knowledge and transparency for processes and resources. CFEngine Nova provides a platform for this through:

- Auto-generated central knowledge console.
- Access to CFEngine's company support knowledge base.
- Tie-in knowledge base with local documentation.
- Dependency mapping between promises in CFEngine, with integration into the semantic
- Trace high level policies to low level policies

2 Mission Portal

Knowledge management has become a new focus in IT management, and it is a core focus at CFEngine. Comprehending the growing complexity of IT operations is one of the main challenges in IT today. CFEngine Nova has a number of key features for knowledge management, including automated documentation, report generation and associative inference. The CFEngine Mission Portal is the centerpiece of user interaction with CFEngine Nova. It offers insight into three main areas:

- Operations and performance.
- Business and compliance.
- Organizational knowledge and library.

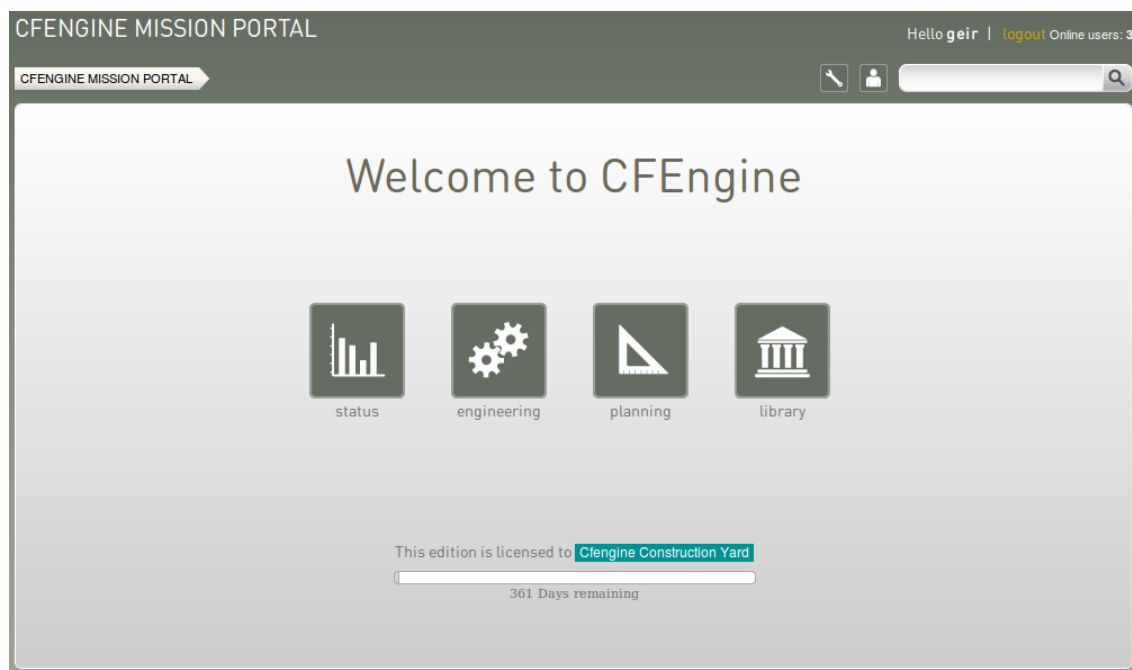


Figure: The mission portal

The Mission Portal uses a finder, viewer, editor model as an application interface, applicable in four main categories:

- Mission Status: a top level overview of compliance status and business value
- Mission Engineering: a place to see the current state of system repair
- Mission Planning: a place to plan and make policy changes
- Mission Library: a knowledge bank that connects information together

Each of these categories is a beginning from which you can refine your overview and search through information.

2.1 Mission Status

Mission status is a high level summary of how well the entire system is behaving. Charts show the business value of the promises kept/not kept as well as host status. Each host is classified into red, yellow, green and blue for a quick overview of their level of compliance.

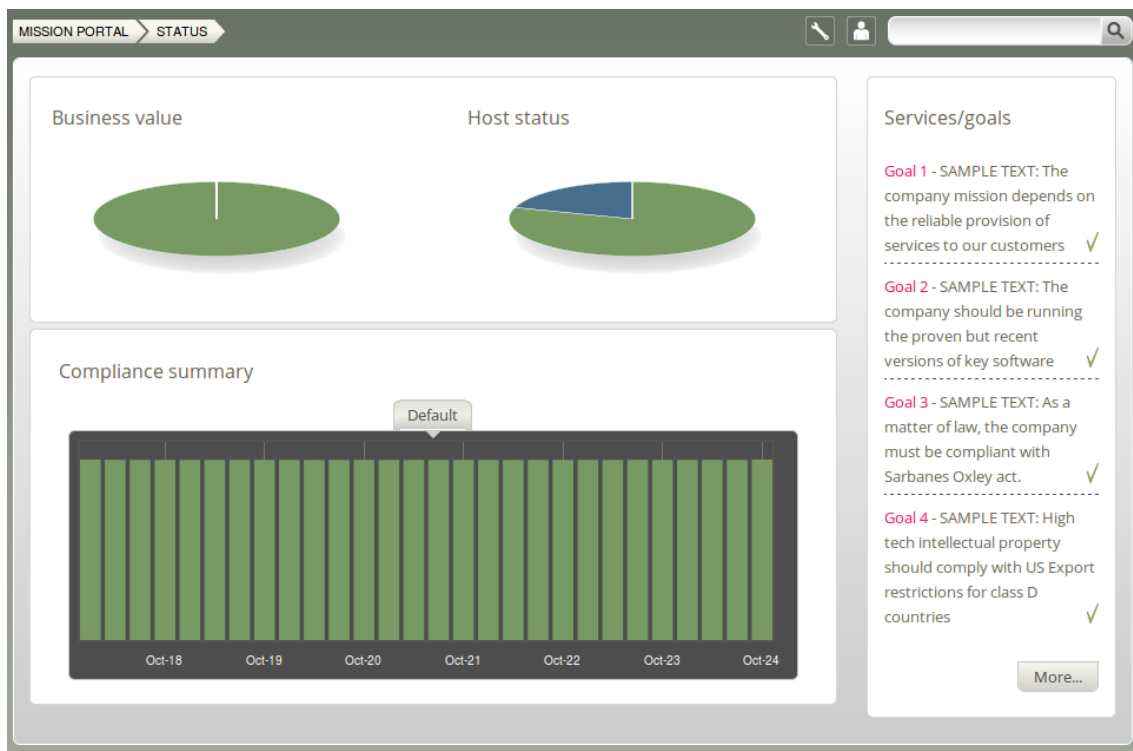


Figure: The status of IT operations.

2.2 Mission Engineering

Mission engineering illustrates the state of the system in relation to the desired state at all scales. Zoom in to specific areas and examine the impact of promises, query data, and extract reports using the finder functions.

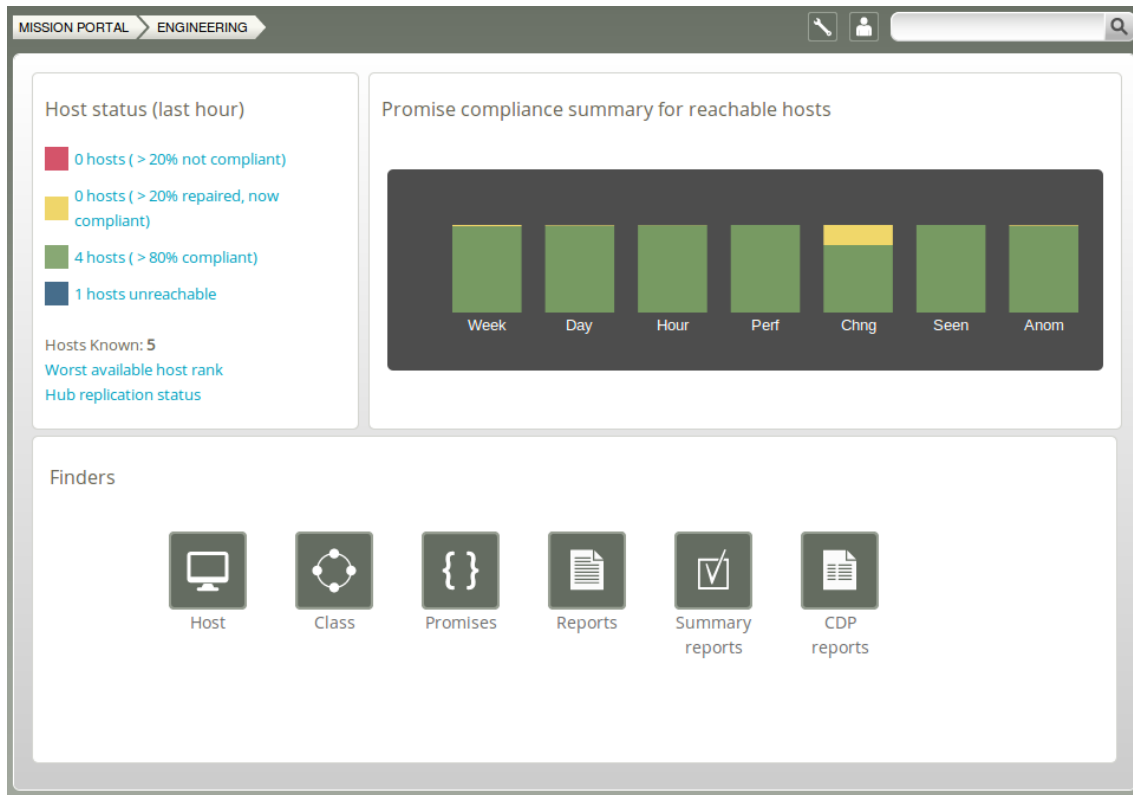


Figure: Mission Engineering

Clicking further on a host takes you to detailed performance data as measured by CFEngine's monitoring daemon. We call this view pseudo-real time monitoring data, because there is always some delay between when data were measured and when they become available for you to view. Most tools simply hide this delay, but CFEngine tells you precisely when data were last updated, providing crucial information about system latencies.

2.3 Mission Planning

Make changes to policies, goals determined by promises and implement specific tactics to achieve the desired state. Interact with data, approve changes and anomalies. Get an overview of users logged on to the Mission Portal, as well as their current activity.

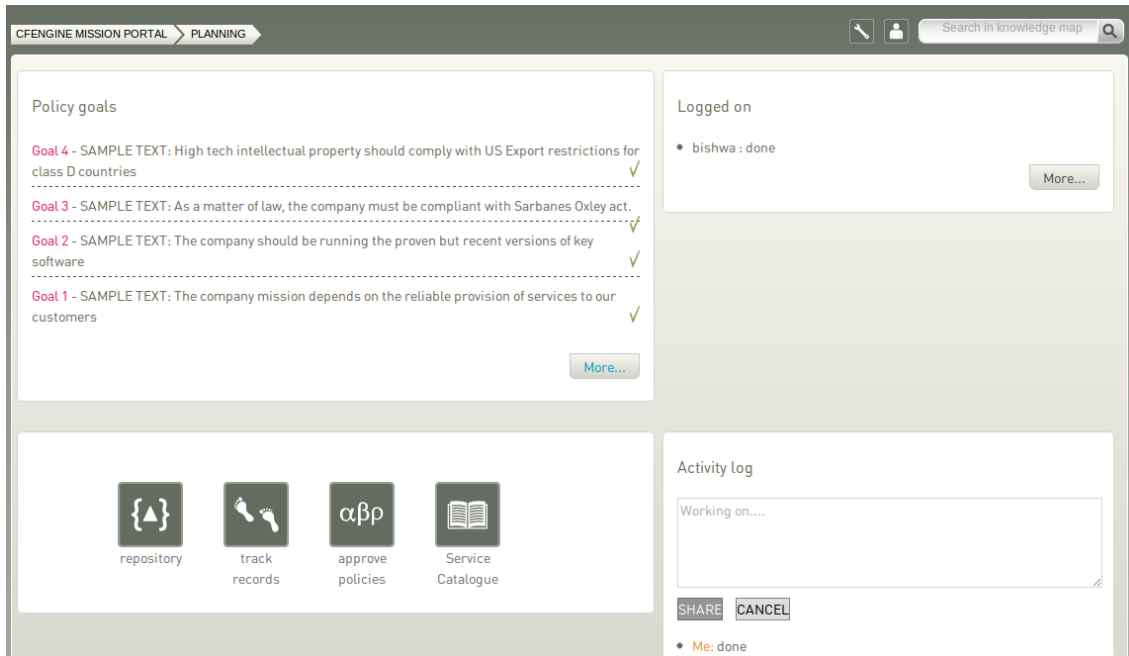


Figure: Mission Planning

2.4 Mission Library

The Library contains finders for documents, topics, a notes archive, and (external) link to the CFEngine community. The topic finder will lead to the Knowledge Map, a useful reference and learning tool. It is designed to show relationships between subjects of importance and provides an index of the information in the knowledge base. Also useful for brainstorming and self-learning.

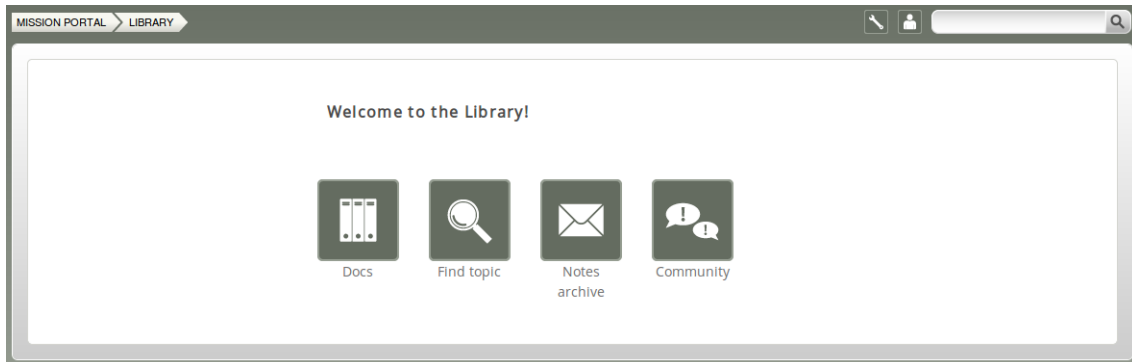


Figure: Mission Library

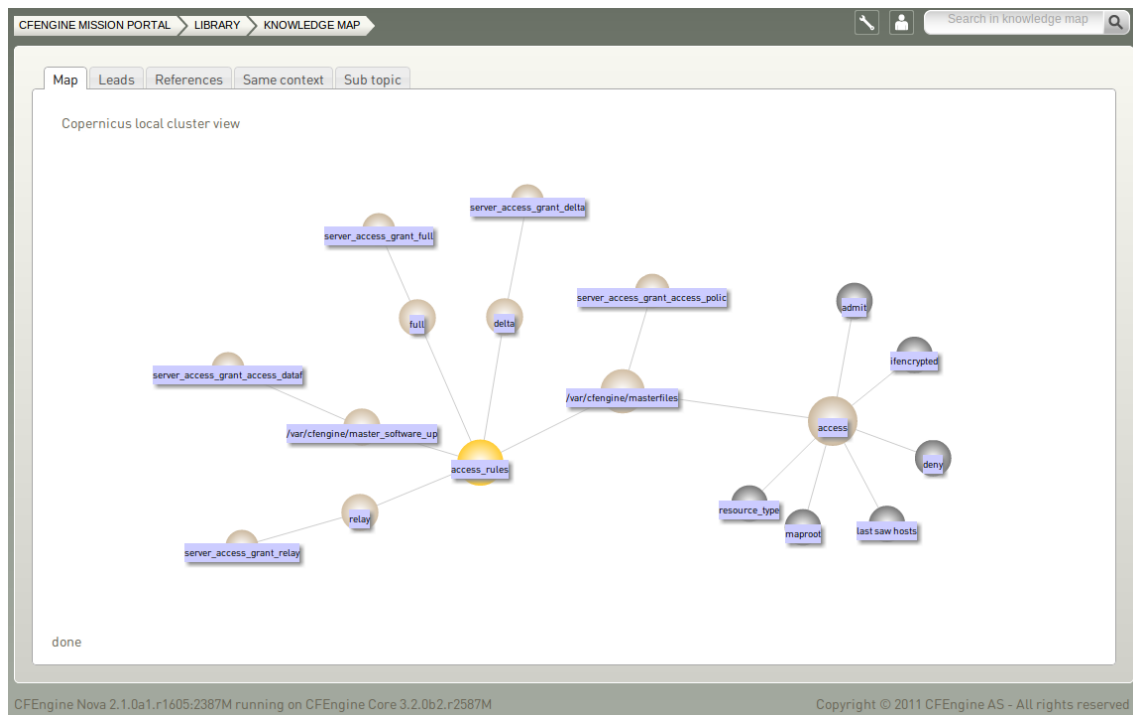


Figure: Topics viewer (Knowledge map)

2.5 Reports in the Mission Portal

A significant capability of CFEngine Nova over previous versions of CFEngine is the existence of automated system reporting. CFEngine collects history, state and change data about computers and ties them together. The CFEngine strategy is to replace conventional CMDBs with a more scalable and

flexible approach to information mining over the coming years. Commercial versions of CFEngine are designed to bring state of the art methods to the problem of information management for IT operations.

In CFEngine Nova 2 reports are summarized and collected automatically by the `cf-hub` program. This is a daemon which runs only on the system hub or policy server. Its sole function is to connect to all registered systems and build a searchable summary of its running state, in as efficient a manner as possible.

In addition to the automated hub, reports may be generated on each individual host using the `cf-report` program. This generates reports in a variety of formats, either for direct viewing (in HTML) or for import/export to other software systems, using XML, CSV or PDF output formats.

2.6 Should monitoring and configuration be separate?

The traditional view of IT operations is that configuration, monitoring and reporting a three different things that should not be joined. Traditionally, all three have been independent centralized processes. This view has emerged historically, but it has a major problem.

Monitoring as an independent activity is inherently non-scalable. When numbers of hosts grow beyond a few thousands, centralized monitoring schemes fail to manage the information. Tying configuration (and therefore repair) to monitoring at the host level is essential for the effective management of large and distributed data facilities. CFEngine foresaw this need in 1998, with its Computer Immunology initiative, and continues to develop this strategy.

CFEngine's approach is to focus on scalability. The commercial editions of CFEngine provide what meaningful information they can in a manner that can be scaled to tens of thousands of machines.

2.7 CMDB and CFEngine

A Configuration Management Database (CMDB), sometimes called a Change and Configuration Management Database (CCMDB), is a repository of information about hardware and its expected state. CMDBs are basically an outgrowth of *inventory databases*. They were embraced by many IT companies as a plausible solution to configuration management, in the absence of an alternative.

A CMDB records so-called *configuration items* (CI), which record technical, ownership and proposed state data. A CMDB does not generally record actual state – this is left for monitoring software.

The term CMDB originates from the IT Infrastructure Library (ITIL) framework, where it is given a prominent role in system planning and verification. The CMDB is by definition a centralized repository.

CFEngine deviates from the classical view of a CMDB to modernize the concept for modern scaling requirements. Configuration management requires something more sophisticated than a database to describe system patterns. The CMDB applies a *brute force* approach to collection and searching of system data that does not scale well, and requires large and expensive resources to manage.

CFEngine works with a highly compressed description of system properties that is based on category classification rather than an exhaustive inventory of computers. Being distributed in operation, CFEngine can also enforce policy on state, without brute force imaging.

2.8 Decentralized data collection in CFEngine

CFEngine is designed to scale to vast numbers of machines. It does so because it is fundamentally decentralized. Data about system state are recorded *in situ* and are not transmitted over the network

directly. CFEngine summarizes and compresses system information before making it available for central aggregation and analysis. This means that high resolution data are available where they count, processing is decentralized, and inexpensive summaries may be compared and mined for correlations.

CFEngine's distributed architecture means that no data are lost if the network fails temporarily. CFEngine uses the network *opportunistically*, but it is not reliant on it for real-time operations.

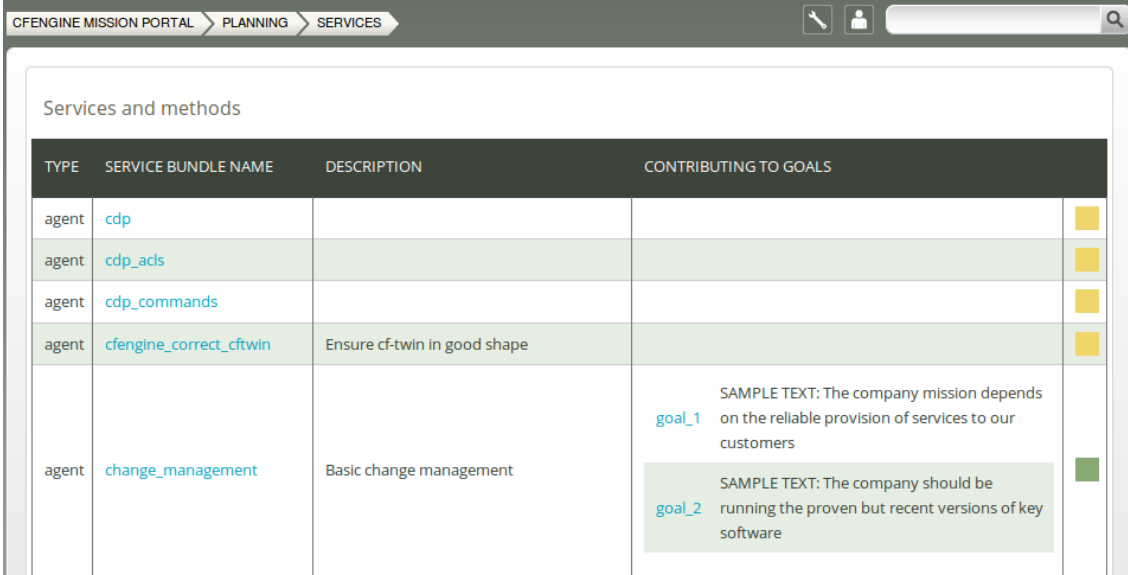
3 Business Integration

3.1 Business goals and the Service Catalogue

Trust and confidence form just one of the pillars of business; agility and creativity make up the others. CFEngine Nova brings features that make it possible to connect the dots meaningfully between high level business goals and low level configuration implementation. This enables different layers of an organization to gain insight into how the system is being managed, in relation to the strategic goals.

- CFEngine Nova's rich repertoire of configuration capabilities allows model-based integration of systems, attending to the unique requirements of each business service. Off-the-shelf does not have to mean bland and generic. Customization is CFEngine's forte. It is the glue holding applications together.
- Nova's speed and use of patterns allows many small changes to be made very quickly.

Nova provides appropriate insight into IT operations through explanation, reporting and visualization of policy, allowing business and IT to gather around information of mutual interest. Of course, CFEngine is not going to make IT experts of business staff, or vice versa. However, business and IT work best together when they can communicate effectively. Our aim with Nova is to give business heads just the right level of insight into technical IT operations to be assured that their needs are being met¹. Similarly, Nova makes business goals available to IT staff so that they can feel connected and responsible for the strategic goals of the organization.



The screenshot shows a web interface titled "CFENGINE MISSION PORTAL" with a navigation menu containing "PLANNING" and "SERVICES". Below the navigation is a search bar and user profile icon. The main content area is titled "Services and methods" and contains a table with the following data:

TYPE	SERVICE BUNDLE NAME	DESCRIPTION	CONTRIBUTING TO GOALS
agent	cdp		
agent	cdp_acls		
agent	cdp_commands		
agent	cfengine_correct_cftwin	Ensure cf-twin in good shape	
agent	change_management	Basic change management	<p>goal_1 SAMPLE TEXT: The company mission depends on the reliable provision of services to our customers</p> <p>goal_2 SAMPLE TEXT: The company should be running the proven but recent versions of key software</p>

A human-readable Service Catalogue generated from technical specifications shows what goals are being attended to automatically

¹ Often it is enough to be able to ask a relevant question and feel that one has a voice, while experts are taking care of the details.

The Service Catalogue is a concept that has been promoted as part of the IT Infrastructure Library (ITIL). Nova generates a service catalogue directly from a model of the knowledge about policy.

```
bundle agent service_catalogue
{
methods:

  "everyone" usebundle => time_management,
              comment => "Ensure clocks are synchronized";

  "everyone" usebundle => garbage_collection,
              comment => "Keep disks free of junk and rotate logs";

  "everyone" -> goal_3,

              usebundle => change_management,
              comment=> "Monitor change";

  "managed hosts" -> goal_1,

              usebundle => CFEngine_management,
              comment => "Management of CFEngine internals";

  "mail server" -> { "goal_3", "goal_1", "goal_2" }

              usebundle => app_mail_postfix,
              comment => "The mail delivery agent";
}
```

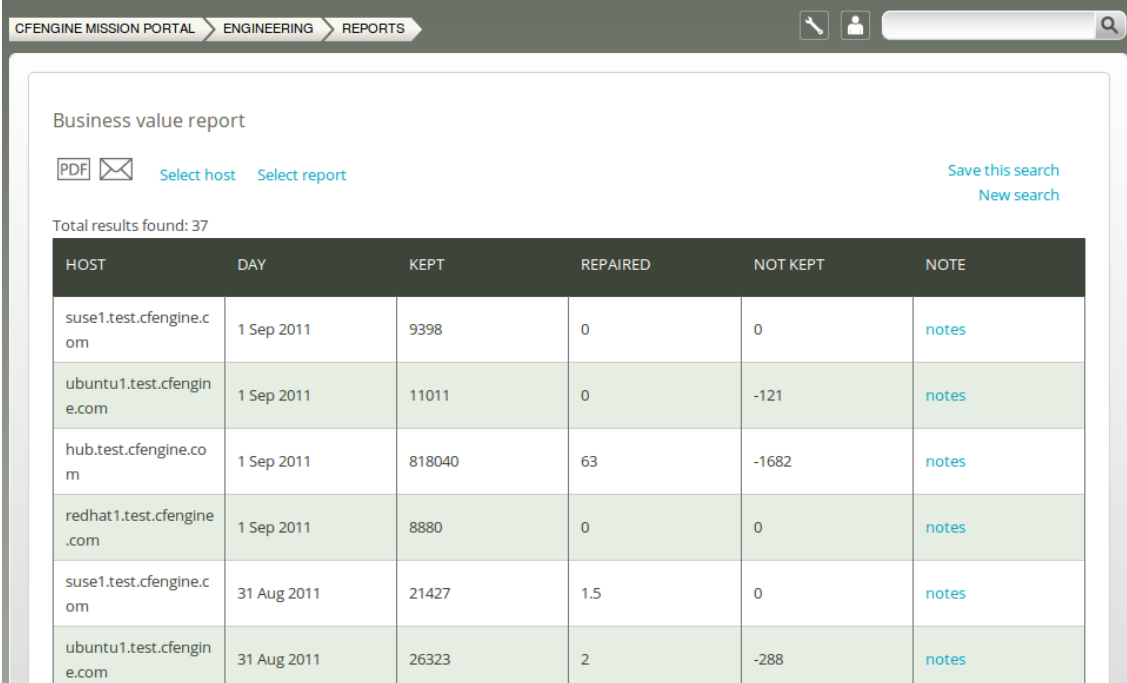
Technical IT administrators can document which promises or bundles of promises are directed towards specific goals by adding a simple reference. The Nova Knowledge Map will then build a narrative around these references and tie the goals to the services in the catalogue (See the Planning room in the Mission Portal).

3.2 Business value reports

One of the capabilities of CFEngine is to add business or organizational value to the configuration model. The notion of business value is not a clear concept, but a very simple approach to measuring value is to attach a monetary value to the outcome of promises.

The `value_kept` (default value 1), `value_repaired` (default value 0.5), `value_notkept` (default value -1) settings fall under CFEngine transaction logging and allow administrators to attach actual monetary (or other) values to promises kept, or issues repaired, or conversely measure the loss of non-compliance in dollar terms (choose your currency). This value is summed and recorded for each execution of CFEngine, and can be turned into graphs for your management reports.

An example of this report is shown below. The results are summed for each day and presented in column form.



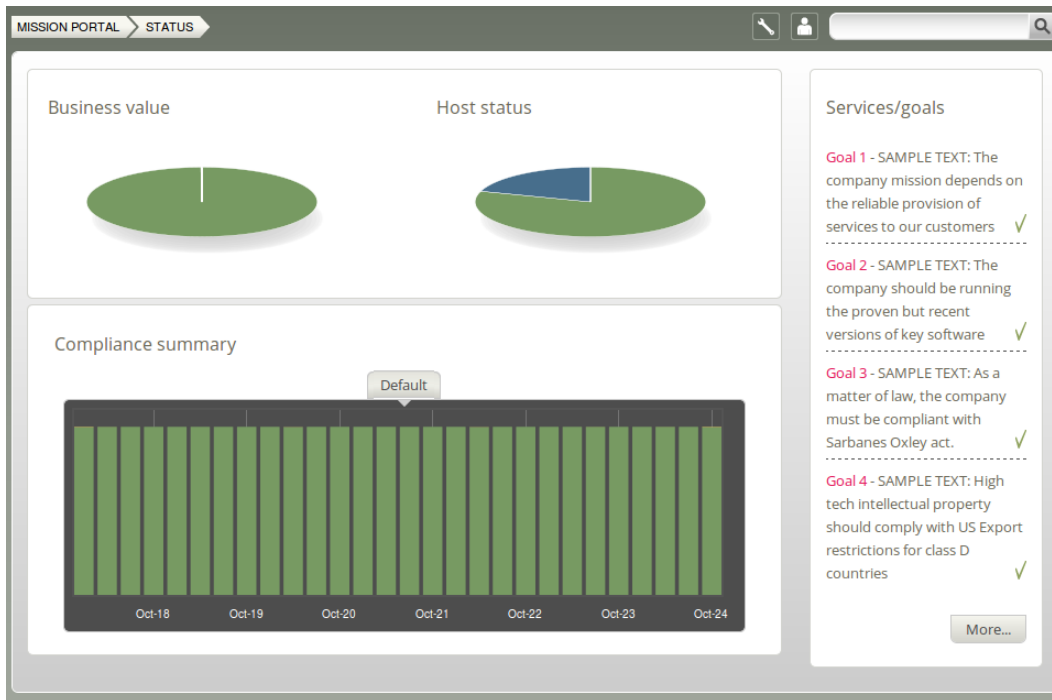
The screenshot shows a web interface for the CFENGINE MISSION PORTAL. The breadcrumb navigation is 'CFENGINE MISSION PORTAL > ENGINEERING > REPORTS'. The page title is 'Business value report'. There are links for 'PDF', 'Select host', and 'Select report'. On the right, there are links for 'Save this search' and 'New search'. Below the title, it says 'Total results found: 37'. A table displays the following data:

HOST	DAY	KEPT	REPAIRED	NOT KEPT	NOTE
suse1.test.cfengine.com	1 Sep 2011	9398	0	0	notes
ubuntu1.test.cfengine.com	1 Sep 2011	11011	0	-121	notes
hub.test.cfengine.com	1 Sep 2011	818040	63	-1682	notes
redhat1.test.cfengine.com	1 Sep 2011	8880	0	0	notes
suse1.test.cfengine.com	31 Aug 2011	21427	1.5	0	notes
ubuntu1.test.cfengine.com	31 Aug 2011	26323	2	-288	notes

Business value accounting can be entered into the model.

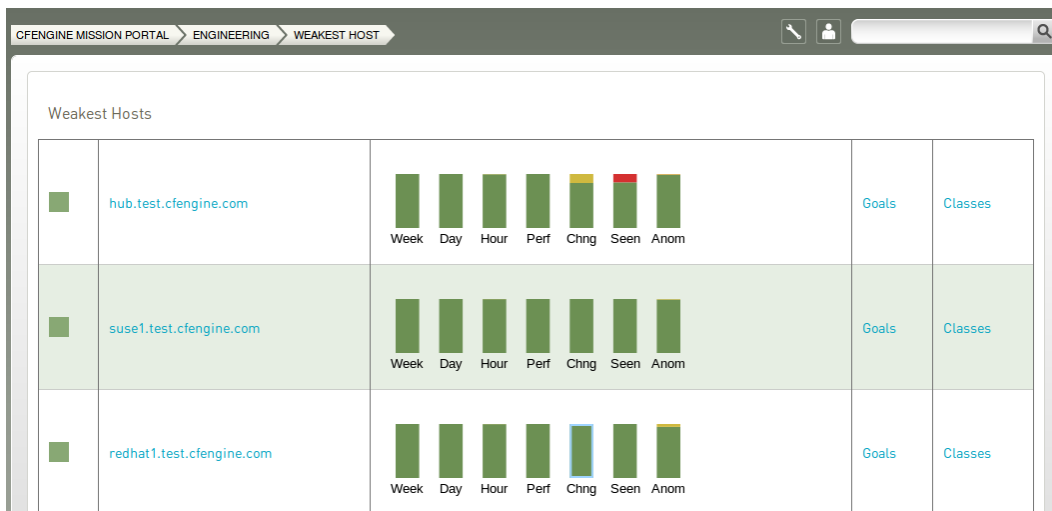
3.3 Simple insight into IT compliance

While business and IT can come together to decide what promises IT services should keep, a simple confirmation of that compliance helps business units to be assured



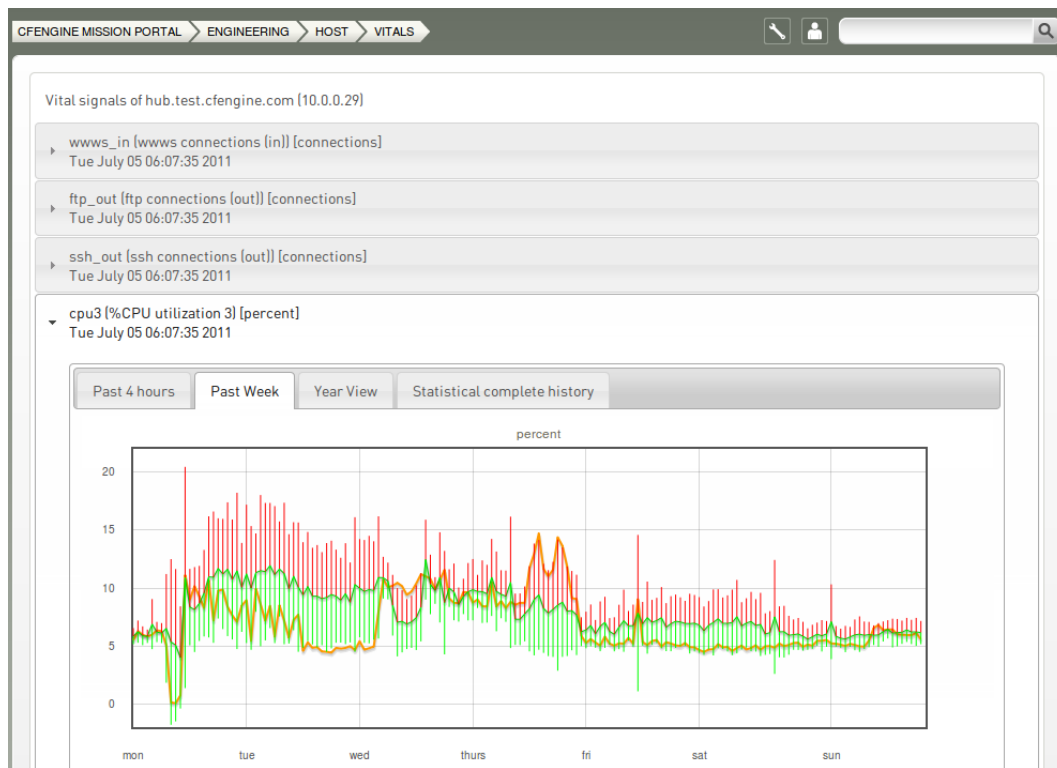
An overview of compliance

Increasing and graded levels of detail can be obtained to match the level of expertise of the viewer. Bar meters can represent more specific details.



Weakest hosts broken down by compliance, performance and change.

For system analysts, Nova can recognize and highlight trends and patterns of behavior that inform further strategic decision making.



Analysis of system performance over multiple time-scales can aid with resource deployment and future planning.

