ekenfHow To Install Nagios 4 and Monitor Your Servers on Ubuntu 14.04

### Introduction

In this tutorial, we will cover the installation of Nagios 4, a very popular open source monitoring system, on Ubuntu 14.04. We will cover some basic configuration, so you will be able to monitor host resources via the web interface. We will also utilize the Nagios Remote Plugin Executor (NRPE), that will be installed as an agent on remote hosts, to monitor their local resources.

Nagios is useful for keeping an inventory of your servers, and making sure your critical services are up and running. Using a monitoring system, like Nagios, is an essential tool for any production server environment.

## Prerequisites

To follow this tutorial, you must have superuser privileges on the Ubuntu 14.04 server that will run Nagios. Ideally, you will be using a non-root user with superuser privileges. If you need help setting that up, follow the steps 1 through 3 in this tutorial: [Initial Server Setup with Ubuntu 14.04](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04).

A LAMP stack is also required. Follow this tutorial if you need to set that up: [How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 14.04](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04).

This tutorial assumes that your server has private networking enabled. If it doesn't, just replace all the references to private IP addresses with public IP addresses.

Now that we have the prerequisites sorted out, let's move on to getting Nagios 4 installed.

## Install Nagios 4

This section will cover how to install Nagios 4 on your monitoring server. You only need to complete this section once.

### Create Nagios User and Group

We must create a user and group that will run the Nagios process. Create a "nagios" user and "nagcmd" group, then add the user to the group with these commands:

* sudo useradd nagios
* sudo groupadd nagcmd
* sudo usermod -a -G nagcmd nagios

### Install Build Dependencies

Because we are building Nagios Core from source, we must install a few development libraries that will allow us to complete the build. While we're at it, we will also install apache2-utils, which will be used to set up the Nagios web interface.

First, update your apt-get package lists:

* sudo apt-get update

Then install the required packages:

* sudo apt-get install build-essential libgd2-xpm-dev openssl libssl-dev xinetd apache2-utils unzip

Let's install Nagios now.

### Install Nagios Core

Download the source code for the latest stable release of Nagios Core. Go to the [Nagios downloads page](http://www.nagios.org/download/core-stay-informed), and click the **Skip to download** link below the form. Copy the link address for the latest stable release so you can download it to your Nagios server.

At the time of this writing, the latest stable release is Nagios 4.1.1. Download it to your home directory with curl:

cd ~

curl -L -O https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz

Extract the Nagios archive with this command:

* tar xvf nagios-\*.tar.gz

Then change to the extracted directory:

* cd nagios-\*

Before building Nagios, we must configure it. If you want to configure it to use postfix (which you can install with apt-get), add --with-mail=/usr/sbin/sendmail to the following command:

* ./configure --with-nagios-group=nagios --with-command-group=nagcmd

Now compile Nagios with this command:

* make all

Now we can run these make commands to install Nagios, init scripts, and sample configuration files:

* sudo make install
* sudo make install-commandmode
* sudo make install-init
* sudo make install-config
* sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf

In order to issue external commands via the web interface to Nagios, we must add the web server user, www-data, to the nagcmd group:

* sudo usermod -G nagcmd www-data

### Install Nagios Plugins

Find the latest release of Nagios Plugins here: [Nagios Plugins Download](http://nagios-plugins.org/download/?C=M;O=D). Copy the link address for the latest version, and copy the link address so you can download it to your Nagios server.

At the time of this writing, the latest version is Nagios Plugins 2.1.1. Download it to your home directory with curl:

cd ~

curl -L -O http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz

Extract Nagios Plugins archive with this command:

* tar xvf nagios-plugins-\*.tar.gz

Then change to the extracted directory:

* cd nagios-plugins-\*

Before building Nagios Plugins, we must configure it. Use this command:

* ./configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl

Now compile Nagios Plugins with this command:

* make

Then install it with this command:

* sudo make install

### Install NRPE

Find the source code for the latest stable release of NRPE at the [NRPE downloads page](http://sourceforge.net/projects/nagios/files/nrpe-2.x/). Download the latest version to your Nagios server.

At the time of this writing, the latest release is 2.15. Download it to your home directory with curl:

* cd ~
* curl -L -O http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz

Extract the NRPE archive with this command:

* tar xvf nrpe-\*.tar.gz

Then change to the extracted directory:

* cd nrpe-\*

Configure NRPE with these commands:

* ./configure --enable-command-args --with-nagios-user=nagios --with-nagios-group=nagios --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86\_64-linux-gnu

Now build and install NRPE and its xinetd startup script with these commands:

* make all
* sudo make install
* sudo make install-xinetd
* sudo make install-daemon-config

Open the xinetd startup script in an editor:

* sudo vi /etc/xinetd.d/nrpe

Modify the only\_from line by adding the private IP address of the your Nagios server to the end (substitute in the actual IP address of your server):

only\_from = 127.0.0.1 10.132.224.168

Save and exit. Only the Nagios server will be allowed to communicate with NRPE.

Restart the xinetd service to start NRPE:

* sudo service xinetd restart

Now that Nagios 4 is installed, we need to configure it.

## Configure Nagios

Now let's perform the initial Nagios configuration. You only need to perform this section once, on your Nagios server.

### Organize Nagios Configuration

Open the main Nagios configuration file in your favorite text editor. We'll use vi to edit the file:

sudo vi /usr/local/nagios/etc/nagios.cfg

Now find an uncomment this line by deleting the #:

#cfg\_dir=/usr/local/nagios/etc/servers

Save and exit.

Now create the directory that will store the configuration file for each server that you will monitor:

sudo mkdir /usr/local/nagios/etc/servers

### Configure Nagios Contacts

Open the Nagios contacts configuration in your favorite text editor. We'll use vi to edit the file:

sudo vi /usr/local/nagios/etc/objects/contacts.cfg

Find the email directive, and replace its value (the highlighted part) with your own email address:

email nagios@localhost ; <<\*\*\*\*\* CHANGE THIS TO YOUR EMAIL ADDRESS \*\*\*\*\*\*

Save and exit.

### Configure check\_nrpe Command

Let's add a new command to our Nagios configuration:

* sudo vi /usr/local/nagios/etc/objects/commands.cfg

Add the following to the end of the file:

define command{

command\_name check\_nrpe

command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

}

Save and exit. This allows you to use the check\_nrpe command in your Nagios service definitions.

### Configure Apache

Enable the Apache rewrite and cgi modules:

sudo a2enmod rewrite

sudo a2enmod cgi

Use htpasswd to create an admin user, called "nagiosadmin", that can access the Nagios web interface:

sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

Enter a password at the prompt. Remember this password, as you will need it to access the Nagios web interface.

**Note:** If you create a user that is not named "nagiosadmin", you will need to edit /usr/local/nagios/etc/cgi.cfg and change all the "nagiosadmin" references to the user you created.

Now create a symbolic link of nagios.conf to the sites-enabled directory:

sudo ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/

Nagios is ready to be started. Let's do that, and restart Apache:

sudo service nagios start

sudo service apache2 restart

To enable Nagios to start on server boot, run this command:

sudo ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios

#### Optional: Restrict Access by IP Address

If you want to restrict the IP addresses that can access the Nagios web interface, you will want to edit the Apache configuration file:

sudo vi /etc/apache2/sites-available/nagios.conf

Find and comment the following two lines by adding # symbols in front of them:

Order allow,deny

Allow from all

Then uncomment the following lines, by deleting the # symbols, and add the IP addresses or ranges (space delimited) that you want to allow to in the Allow from line:

# Order deny,allow

# Deny from all

# Allow from 127.0.0.1

As these lines will appear twice in the configuration file, so you will need to perform these steps once more.

Save and exit.

Now restart Apache to put the change into effect:

sudo service nagios restart

sudo service apache2 restart

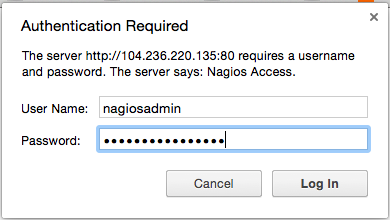
Nagios is now running, so let's try and log in.

## Accessing the Nagios Web Interface

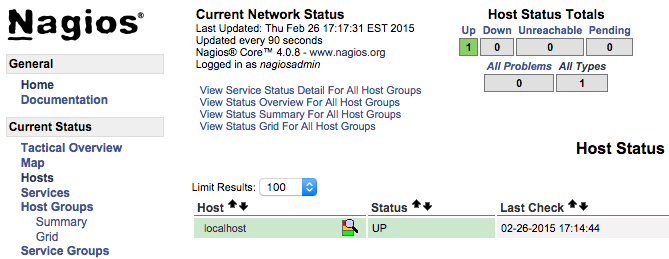
Open your favorite web browser, and go to your Nagios server (substitute the IP address or hostname for the highlighted part):

http://nagios\_server\_public\_ip/nagios

Because we configured Apache to use htpasswd, you must enter the login credentials that you created earlier. We used "nagiosadmin" as the username:



After authenticating, you will be see the default Nagios home page. Click on the **Hosts** link, in the left navigation bar, to see which hosts Nagios is monitoring:



As you can see, Nagios is monitoring only "localhost", or itself.

Let's monitor another host with Nagios!

## Monitor a Host with NRPE

In this section, we'll show you how to add a new host to Nagios, so it will be monitored. Repeat this section for each server you wish to monitor.

On a server that you want to monitor, update apt-get:

sudo apt-get update

Now install Nagios Plugins and NRPE:

sudo apt-get install nagios-plugins nagios-nrpe-server

### Configure Allowed Hosts

Now, let's update the NRPE configuration file. Open it in your favorite editor (we're using vi):

sudo vi /etc/nagios/nrpe.cfg

Find the allowed\_hosts directive, and add the private IP address of your Nagios server to the comma-delimited list (substitute it in place of the highlighted example):

allowed\_hosts=127.0.0.1,10.132.224.168

Save and exit. This configures NRPE to accept requests from your Nagios server, via its private IP address.

### Configure Allowed NRPE Commands

Look up the name of your root filesystem (because it is one of the items we want to monitor):

df -h /

We will be using the filesystem name in the NRPE configuration to monitor your disk usage (it is probably /dev/vda). Now open nrpe.cfg for editing:

sudo vi /etc/nagios/nrpe.cfg

The NRPE configuration file is very long and full of comments. There are a few lines that you will need to find and modify:

* **server\_address**: Set to the private IP address of this host
* **allowed\_hosts**: Set to the private IP address of your Nagios server
* **command[check\_hda1]**: Change /dev/hda1 to whatever your root filesystem is called

The three aforementioned lines should look like this (substitute the appropriate values):

server\_address=client\_private\_IP

allowed\_hosts=nagios\_server\_private\_IP

command[check\_hda1]=/usr/lib/nagios/plugins/check\_disk -w 20% -c 10% -p /dev/vda

Note that there are several other "commands" defined in this file that will run if the Nagios server is configured to use them. Also note that NRPE will be listening on port 5666 because server\_port=5666 is set. If you have any firewalls blocking that port, be sure to open it to your Nagios server.

Save and quit.

### Restart NRPE

Restart NRPE to put the change into effect:

sudo service nagios-nrpe-server restart

Once you are done installing and configuring NRPE on the hosts that you want to monitor, you will have to add these hosts to your Nagios server configuration before it will start monitoring them.

### Add Host to Nagios Configuration

On your Nagios server, create a new configuration file for each of the remote hosts that you want to monitor in /usr/local/nagios/etc/servers/. Replace the highlighted word, "yourhost", with the name of your host:

sudo vi /usr/local/nagios/etc/servers/yourhost.cfg

Add in the following host definition, replacing the host\_name value with your remote hostname ("web-1" in the example), the alias value with a description of the host, and the address value with the private IP address of the remote host:

define host {

use linux-server

host\_name yourhost

alias My first Apache server

address 10.132.234.52

max\_check\_attempts 5

check\_period 24x7

notification\_interval 30

notification\_period 24x7

}

With the configuration file above, Nagios will only monitor if the host is up or down. If this is sufficient for you, save and exit then restart Nagios. If you want to monitor particular services, read on.

Add any of these service blocks for services you want to monitor. Note that the value of check\_command determines what will be monitored, including status threshold values. Here are some examples that you can add to your host's configuration file:

Ping:

define service {

use generic-service

host\_name yourhost

service\_description PING

check\_command check\_ping!100.0,20%!500.0,60%

}

SSH (notifications\_enabled set to 0 disables notifications for a service):

define service {

use generic-service

host\_name yourhost

service\_description SSH

check\_command check\_ssh

notifications\_enabled 0

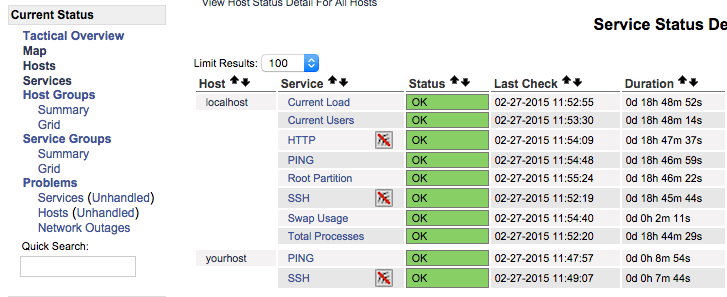
}

If you're not sure what use generic-service means, it is simply inheriting the values of a service template called "generic-service" that is defined by default.

Now save and quit. Reload your Nagios configuration to put any changes into effect:

sudo service nagios reload

Once you are done configuring Nagios to monitor all of your remote hosts, you should be set. Be sure to access your Nagios web interface, and check out the **Services** page to see all of your monitored hosts and services:



## Conclusion

Now that you monitoring your hosts and some of their services, you might want to spend some time to figure out which services are critical to you, so you can start monitoring those. You may also want to set up notifications so, for example, you receive an email when your disk utilization reaches a warning or critical threshold or your main website is down, so you can resolve the situation promptly or before a problem even occurs.

# Configure Nagios on Ubuntu :

# Part 1 - Nagios Server:

### Overview

Nagios is a popular server monitoring software suite which can help greatly improve the uptime of your application. In this guide, I'm going to show you how to install Nagios on Ubuntu 14.04.

You need a LAMP stack installed on your server to be able to use Nagios. If you would like to learn how you can do this on Ubuntu, please refer to the guide: [How to Install Apache, MySQL, and PHP on Ubuntu](https://www.vultr.com/docs/how-to-install-apache-mysql-and-php-on-ubuntu)

### Step 1: Installing prerequisites

Aside from a functioning LAMP stack, there are three prerequisites. Install them with apt-get:

apt-get install build-essential libgd2-xpm-dev apache2-utils

### Step 2: Creating a group and user

For Nagios, we need to create a group and user. First off, create the user:

useradd -m nagios

If you didn't give it a password yet, do this with the passwd command:

passwd nagios

We will now create the Nagios group:

groupadd nagcmd

Add the Nagios user to the Nagios group:

usermod -a -G nagcmd nagios

And add the www-data user to the Nagios group:

usermod -a -G nagcmd www-data

### Step 3: Downloading Nagios

Now, we can download Nagios. Install wget now, if you don't already have it installed:

apt-get install wget

Then, download the Nagios files:

wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-4.0.8.tar.gz

wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz

Untar it:

tar xzf nagios\*

And finally, navigate to the directory:

cd nagios\*

### Step 4: Installing Nagios

Now that our server is setup, we can compile and install Nagios.

./configure --with-command-group=nagcmd

make all

make install

make install-init

make install-config

make install-commandmode

make install-webconf

### Step 5: Configuring the Nagios web interface

Nagios features a web interface for easy management. For security purposes, you should limit who can access the web interface with either a firewall or a password-protected Apache configuration.

Restart Apache:

service apache2 restart

Start the Nagios service:

service nagios start

# Part 2 - Remote Host:

### Adding remote hosts

In the previous part of this tutorial, we configured a Nagios server. Now that Nagios is installed and running, it's time to configure a remote host that will be monitored by Nagios. SSH into any remote host that you want to be monitored.

### Step 1: Installing prerequisites

To monitor hosts, we need to add them to Nagios. By default, Nagios only monitors localhost (the server it's running on). We're going to add hosts that are part of our network to gain even more control. You will need to use the following instructions on all hosts that you want to monitor.

First, install nagios-plugins and nagios-nrpe-server:

apt-get install nagios-plugins nagios-nrpe-server

### Step 2: Configuring NRPE

Next, open the /etc/nagios/nrpe.cfg file. Replace the value of allowed\_hosts with 127.0.0.1,0.0.0.0 replacing the second IP with the IP address of the Nagios server.

We will now open the file /etc/nagios/nrpe.cfg and replace a couple of values.

* Replace the value of server\_address to the private IP address of the host.
* Set allowed\_hosts to the private IP address of your Nagios server.
* Execute df -h /, copy the output, and put that as the value of command. It indicates your root file system.

Save the file when you are finished.

Now restart NRPE:

service nagios-nrpe-server restart

### Step 3: Adding the host to Nagios

Now that we've configured the host we're going to monitor, we need to switch back to our Nagios server and add the host to it. Open the following file with your favorite editor:

/usr/local/nagios/etc/servers/host.cfg

Use the following block as a template. Replace host with an appropriate name for your remote host, and update the host\_name, alias, and address values accordingly.

define host {

use linux-server

host\_name yourhost

alias My first Apache server

address 1.2.3.4

max\_check\_attempts 5

check\_period 24x7

notification\_interval 30

notification\_period 24x7

}

This will allow you to simply monitor whether the server is up or down. Now reload Nagios:

service nagios reload

# Server Monitoring With Nagios

### Capabilities

Nagios is recognized as the top solution to monitor servers in a variety of different ways.  Server monitoring is made easy in Nagios because of the flexibility to monitor your servers with and without agents.  With over 3500 different addons available to monitor your servers, the community at the Nagios Exchange have left no stone unturned.

Nagios is fully capable of monitoring Windows servers, Linux servers, Unix servers, Solaris, AIX, HP-UX, and Mac OS/X and more.

### Benefits

Implementing effective server monitoring with Nagios offers the following benefits:

* Increased server, services, process, and application availability
* Fast detection of network and server outages and protocol failures
* Fast detection of failed servers, services, processes and batch jobs

### Solutions

These Nagios solutions provide server monitoring capabilities and benefits:

* [Nagios XI](https://www.nagios.com/products/nagios-xi/)
* [Nagios Core](https://www.nagios.com/products/nagios-core/)

# Nagios XI:

## Benefits

### Comprehensive IT Infrastructure Monitoring

Provides monitoring of all mission-critical infrastructure components including applications, services, operating systems, network protocols, systems metrics, and network infrastructure. Hundreds of third-party addons provide for monitoring of virtually all in-house applications, services, and systems.

### Performance

The powerful Nagios Core 4 monitoring engine provides users with the highest degree of monitoring server performance. High-efficiency worker processes allow for nearly limitless scalability and monitoring effectiveness.

### Visibility

Provides a central view of your entire IT operations network and business processes. Powerful dashboards provide at-a-glance access to powerful monitoring information and third-party data. Views provide users with quick access to the information they find most useful.

### Proactive Planning & Awareness

Automated, integrated trending and capacity planning graphs allow organizations to plan for infrastructure upgrades before outdated systems catch them by surprise. Alerts are sent to IT staff, business stakeholders, and end-users via email or mobile text messages, providing them with outage details so they can start resolving issues immediately.

### Customizability

A powerful GUI provides for customization of layout, design, and preferences on a per-user basis, giving your customers and team members the flexibility they want.

### Ease of Use

Integrated web-based configuration interface lets admins hand out control of managing monitoring configuration, system settings, and more to end-users and team members easily. Configuration wizards guide users through the process of monitoring new devices, services, and applications – all without having to understand complex monitoring concepts.

### Multi-Tenant Capabilities

Multi-user access to web interface allows stakeholders to view relevant infrastructure status. User-specific views ensure clients only see the infrastructure components they’re authorized for. Advanced user management simplifies administration by allowing you to manage user accounts easily. Provision new user accounts with a few clicks and users automatically receive an email with their login credentials.

### Extendable Architecture

Multiple APIs provide for simple integration with in-house and third-party applications. Thousands of community-developed addons extend monitoring and native alerting functionality. Custom interface and addon developments are available to tailor Nagios XI to meet your organization’s exact needs. [Request a Quickstart For Nagios XI](https://www.nagios.com/services/quickstart/nagios-xi)

## Features:

**Powerful Monitoring Engine:** Nagios XI uses the powerful Nagios Core 4 monitoring engine to provide users with efficient, scalable monitoring.

**Updated Web Interface:** Your new dashboard provides a customizable high-level overview of hosts, services, and network devices.

**Advanced Graphs & More:** Administrators can easily view network incidents and resolve them before they become major catastrophes.

**Capacity Planning:** Automated, integrated trending and capacity planning graphs allow organizations to plan for upgrades.

**Configuration Wizards:** Fast Wizards! Simply enter the required information, and you’re up and monitoring with a few simple clicks.

**Infrastructure Management:** Improved Bulk Host Import, Autodiscovery, Auto Decommissioning, Mass Acknowledgment & much more!

**Configuration Snapshot:** Save your most recent configurations. Archive it. Revert back whenever you like. Never lose it again. Relax.

**Advanced User Management:** Easily setup and manage user accounts with only a few clicks then assign custom roles to ensure a secure environment.

## System Requirements

These figures represent the minimum requirements to run Nagios XI. To view more detailed guidelines [view our hardware requirements PDF](https://assets.nagios.com/downloads/nagiosxi/docs/Nagios-XI-Hardware-Requirements.pdf).

### Hard Drive

20 GB

### Memory

2 GB

### CPU

Dual core, 2.4 GHz

### Operating System

CentOS or Redhat Enterprise Linux (RHEL) versions 6, or 7

### Database

MySQL/MariaDB, plus PostgresQL if running versions less than XI 5 or if upgrading from a pre-5 version

## Pricing

### License Levels

Nagios XI is available in multiple different license levels: 100 Node through Unlimited Node. A node is anything with an IP address or domain that you would like to monitor (switches, routers, firewalls, workstations, and other network devices). There are no restrictions on the number of services that can be monitored. Each license includes twelve months of maintenance (upgrade entitlements) and email support.

### Standard Edition vs. Enterprise Edition

Nagios XI is available in two different editions: Standard Edition and an Enterprise Edition. The Enterprise Edition provides users with additional functionality and includes features that are designed to aid in large-scale configuration, forecasting, and scheduled reporting. Each license includes twelve months of maintenance (upgrade entitlements) and email support.

**Standard Edition**

**From $1,995**

* Easy Configuration Wizards
* GUI Configuration
* Advanced Reporting
* Enhanced Visualizations
* Custom User Dashboards
* Custom User Views
* Executive Summary Report
* Custom Actions
* Dashboard Deployment
* Notification Escalations

**Enterprise Edition**

**From $3,495**

* **Everything in Standard Edition Plus:**
* Scheduled Reports
* Capacity Planning Reports
* Web-Based Server Console Access
* Bulk-Modification Tools
* Audit Logging
* Notification Deployment
* SLA Reports
* Scheduled Pages
* Automated Host Decommissioning

### Nagios Core

# The open source industry standard in IT infrastructure monitoring and alerting

## Benefits

### Monitoring

IT staff configure Nagios to monitor critical IT infrastructure components, including system metrics, network protocols, applications, services, servers, and network infrastructure.

### Alerting

Nagios sends alerts when critical infrastructure components fail and recover, providing administrators with notice of important events. Alerts can be delivered via email, SMS, or custom script.

### Response

IT staff can acknowledge alerts and begin resolving outages and investigating security alerts immediately. Alerts can be escalated to different groups if alerts are not acknowledged in a timely manner.

### Reporting

Reports provide a historical record of outages, events, notifications, and alert response for later review. Availability reports help ensure your SLAs are being met.

### Maintenance

Scheduled downtime prevents alerts during scheduled maintenance and upgrade windows.

### Planning

Trending and capacity planning graphs and reports allow you to identify necessary infrastructure upgrades before failures occur.

## Features

**Comprehensive Monitoring:** Core provides monitoring of all mission-critical infrastructure components in your IT infrastructure.

**Visibility:** Gain a centralized view of your entire IT operations and review detailed status information through the web interface.

**Awareness:** Alerts with escalation capabilities are delivered to IT staff via email and SMS to ensure fast detection of outages.

**Problem Remediation:** Event handlers can automatically restart failed applications, servers, devices, and services when problems are found.

**Proactive Planning:** Trending and capacity planning extensions allow you to proactively plan for upgrades.

**Reporting:** Availability reports ensure SLAs are being met and historical reports provide record of critical information.

**Multi-Tenant Capabilities:** Multi-user access and user-specific views can be configured to ensure clients see specific information.

**Extendable Architecture:** Multiple API’s provide integration with in-house / third-party applications, and community-developed add-ons.