Nagios XI 2024 Best Practices

Purpose

This document describes Nagios XI 2024 Best Practices.

Target Audience

This document is intended for use by Nagios XI Administrators as a guide to understand how to get the most out of Nagios XI. It covers a range of topics such as how to get the most out of XI, things you wish you knew, and configuration practices.

Concept

Strictly speaking, “best practices” is a flexible concept. Depending on your environment and your needs, some topics will not apply, and other topics may be exactly what is needed. The information in this talk reflects Nagios XI deployments in the wild.

Additionally, monitoring is not all about metrics and thresholds, it can also be helpful to ensure standards are enforced. If a setting gets changed you can be notified about it instead of having to track it down during one of your troubleshooting adventures.

Nagios XI License Entitlements

Each XI license entitles 3 instances:

- Production
- Test & Dev (T&D)
- Disaster Recovery (DR)

License activation is tied to the IP Address of each XI host. The caveat to this is that only your Production system can be used for actual monitoring. We are flexible and understand your need to actively have a test system that runs alongside production. This allows you to implement new checks into production with confidence that they will work.

Monitoring Nagios XI

- How would you know your XI server died?
  - Nagios XI Server Monitoring Wizard
- DR instance monitors production instance
  - Production instance is UP & HEALTHY
- Production Instance monitors DR instance
  - DR instance is UP & HEALTHY

The most important part of a monitoring system is knowing that it is working. Nagios XI comes with a "Nagios XI Server" monitoring wizard to ensure everything is "OK." However, there is no point in monitoring itself if it's down you are not going to hear about it.

Utilize your DR instance to monitor the production instance. This way, if the production instance goes down, you'll receive alerts about it. The same applies for monitoring the DR instance, make sure production monitors the DR instance to make sure it's healthy. By using the "Nagios XI Server" wizard to monitor the other instance, you can have confidence knowing that when something goes wrong, you will really hear about it.

Monitoring the Nagios XI "localhost"

Services you should ideally be monitoring:

- `crond`, `httpd`, `MySQL`, `ndo2db`, `npcd`, and `ntpd`
- `snmptrapd` and `snmptt` are not present until you follow the XI SNMP Trap procedure.
- The plugin called `check_init_service` that is part of the `linux-nrpe-agent.tar.gz` package can be used to monitor these services.
  - Requires the following line in /etc/sudoers

```
nagios ALL=NOPASSWD: /usr/local/nagios/libexec/check_init_service
```

  - Your command definition needs to be:

```
sudo $USER1$/check_init_service $ARG1$
```

File counts - check these folders to make sure the temp files are being processed:

- NPCD Perfdata spool directory

```
/usr/local/nagios/var/spool/perfdata/
```

- xidpe spool directory

```
/usr/local/nagios/var/spool/xidpe/
```

- Check the results folder.

```
/usr/local/nagios/var/spool/checkresults/
```

- snmptt spool folder

```
/var/spool/snmptt/
```

Note: These locations can vary if a RAMDisk has been implemented

- Use the **Folder Watch** wizard to create services to monitor these folders.

Has the nagios user account expired?

In some customer installations, it's possible that the nagios user account expires. This isn't always that obvious to troubleshoot, so checking that it hasn't expired is a good precautionary measure.

- **check_pass_expire.pl** is the plugin you can use (download here)

  File permissions changes required:

  ```
  setfacl -m u:nagios:r-- /etc/shadow
  ```

  Plugin also needs **line 23** changed.

  From:

  ```
  use lib ".";
  ```

  To:

  ```
  use lib "/usr/local/nagios/libexec";
  ```

Root mailbox size

If you are not familiar with Linux, then you do not know about the system mailbox. This is a local mail system on linux servers where messages are sometimes sent.

Certain components used in Nagios XI such as MRTG will send messages to this mailbox when it has a problem. An incorrect MRTG configuration can cause a message to be sent every five minutes as this is when MRTG runs. That's about 288 messages a day. Over time the root mailbox can grow to GB in size causing issues. This plugin can be used to monitor the root mailbox and let you know when it gets too big.

- The plugin is **box293_check_mbox.pl** (download here)

  Required file permission changes:

  ```
  setfacl -m u:nagios:r-- /var/spool/mail/root
  ```

MySQL / MariaDB Databases

If the tables are crashed and go undetected, this can have a severe impact on the system, and you may not be storing important data which may cause strange problems.

- **box293_check_mysql_table_status** is the plugin you can use to check this:

  https://exchange.nagios.org/directory/Plugins/Databases/MySQL/box293_check_mysql_table_status/details

- Also, another problem can occur if the database engine runs on a different timezone to the local system.
- **box293_check_mysql_date** is the plugin you can use to check this:

  https://exchange.nagios.org/directory/Plugins/Databases/MySQL/box293_check_mysql_date/details

Overall Load

- This service is included by default in Nagios XI.
- `check_load` is the plugin included with Nagios XI.

Memory Free – Physical

- Make sure your XI server does not run out of memory.
- The plugin is `check_memory.sh`. https://exchange.nagios.org/directory/Plugins/Operating-Systems/Linux/check_memory-2Esh/details

Swap Usage

- If the system runs out of physical memory and starts swapping to disk, the system performance will be impacted.
- This service is included by default in Nagios XI
- `check_swap` is the plugin included with Nagios XI.

Disk Free

- Disk free space is very important.
- This service is included by default in Nagios XI however it only monitors `/`
- If you have different volumes mounted, then you should be monitoring each one of these.
- `check_disk` is the plugin included with Nagios XI.

Date and Timezone

Make sure your XI server has its time zone correctly defined.

- Configure Time zone.
- **Admin > System Config > System Settings**

It can be the source of confusing problems such as:

- Apply configuration throwing errors.
- Performance data not being processed.

When using NTP make sure it's synchronized with a trusted time source like **pool.ntp.org**. If this is a virtual machine, **do not sync** its time with the hypervisor (ESXi, HyperV etc.)!

CPU

Having the right amount of CPU cores is important but so is the speed of those cores. Not all plugins and processes are multi-threaded, so a higher speed CPU is going to benefit a 3.4GHz CPU will do a lot more than a 2.2GHz one. Refer to the Nagios XI Hardware Requirements guide.

Memory

How much memory do you need on an XI system? When all the hosts and services in XI are healthy, the amount of memory used is far less compared to a major system outage. When XI fires off event handlers they consume memory, if there is a major outage and a lot of event handlers are being executed, a lot of memory is being consumed. It doesn't take long for 6GB of memory to be used. You should have at least 50% more memory than needed. Refer to the Nagios XI Hardware Requirements guide.

RAM Disk

Configuring Nagios XI with a RAM Disk is highly recommended as the number of monitored objects increases. The more things you are monitoring the more disk I/O occurs. By directing this traffic to a RAM Disk, the time it takes for that I/O operation to complete is drastically faster.

- Reduces disk I/O & load.
- Speeds up processing of performance data.
- Speeds up processing of spooled check results.
- Speeds up nagios restarts.

Refer to the Utilizing a RAM disk procedure.

Solid State Disk (SSD)

Greatly improves overall performance.

- Compliments RAM Disk
- Helps read/writes with:
  - Logs
  - Database
  - Performance Graphs
  - Reports

SSD vs RAID

RAID allows for much larger disk capacities than SSD can provide, however it would be hard for a spinning disk RAID set to beat the performance of SSD.

Keep in mind if you implement SSD, you should implement RAID1 sets for redundancy purposes.

rrdcached

rrdcached is a way of accumulating the received performance data and then processing it in a batch job. It helps with larger installations and can reduce I/O; however, it can also result in performance graphs lagging behind the real time results.

Refer to the Using rrdcached with Nagios XI procedure

Offloaded MySQL / MariaDB

On larger installations there can be a lot more data being written on the databases, which in turn can result in a lot of CPU usage directed away from actual monitoring.

- Offloading to a separate server will remove this CPU usage from your monitoring server.
- Of course, make sure you monitor the offloaded server!
    - Disk/CPU/Memory/Tables/Service
    - Refer to earlier notes about services.
- Refer to the Offloading MySQL to Remote Server procedure

Disaster Recovery

Define what is important to you in a disaster. Once you have clearly defined goals and outcomes you can plan appropriately and test.

These presentations cover DR options:

High Availability and Failover Solutions for Nagios XI

Jeremy Rust & Devin Vance: Scaling Across Data Centers Using High Availability

Backups

Have you scheduled your backups in Nagios XI?

- **Admin > System Backups**
- Schedule backups of Nagios XI
  - Location can be local, FTP, SSH
  - Remote location recommended. Storing them on storage that is not local to the XI file system is important - make sure you can get to your backups if your XI server dies.
- Manual Backups
  - Local Backup Archives via Admin menu
  - `/usr/local/nagiosxi/scripts/backup_xi.sh`

Restoring Backups

The Backing Up And Restoring Nagios XI procedure is very straight-forward and allows for a full recovery of your Nagios XI system.

Another effective use of it is to migrate XI from one server to another.

Service Dependencies

When a host goes down, the services still get executed and can result in services in an unknown or critical state. Nagios suppresses any notifications however they still appear critical in the interface.

Sometimes a host can be up, but the monitoring agent can be down. An example of this is an NRPE agent. By using service dependencies, if the master service goes down, you prevent notifications from being sent OR prevent checks from being executed. Either option simply pushes the next check or notification to the next interval.

Disable Service Checks

A new feature of Nagios Core 4.1.0 (included in XI 5 onwards) is a configuration directive called.

```
host_down_disable_service_checks
```

- Best described as automatic service dependencies on their own hosts.
- System wide setting, this applies across the board, it is not granular.
- Defined in /usr/local/nagios/etc/nagios.cfg.

```
host_down_disable_service_checks=1
```

    - Can also be defined via **CCM > CCM Admin > Core Configs.**

- Restart nagios monitoring engine or Apply Configuration for setting to take effect.
- Can reduce the load on the XI host as plugins will not be executed if this host is down.
- Keep in mind that if the host is down then any defined service dependencies will be ignored.

Check Intervals - Be Realistic

It can be quite easy to set up your monitoring with the same intervals across the board. This can lead to peaks and troughs in load on the XI server as a lot of checks can occur in the same time windows.

Think about what you are monitoring and how often do you really need to check it. Something like disk usage rarely runs out quickly, you can monitor this every hour and be confident you will be notified about the free disk space running low in a reasonable time.

- Does it need to be checked every 5 minutes?
- Disk Free Space – every 60 minutes perhaps?
- Too long = no performance data
    - An interval that is more than four hours apart
- However, if you are going to make it every hour, why not every 58 minutes or 61 minutes? Try to spread the load out a bit.
- Different intervals to spread the load.
    - 3, 5, or 7 minute intervals.
    - 58, 60, or 62 minute intervals.

Notification & Check Intervals

Sometimes larger check intervals can have an adverse effect on notification intervals. The monitoring engine determines if it should send a notification every time a check result is received.

Due to how the internal scheduling works, you might fall short of the notification window by a small time period like 20 seconds. This means it might be another 15 minutes until the next check is run, that's when the notification will be sent, e.g. 15-minute check and 60-minute notification.

- Internal scheduling may cause 14min 55sec to pass, 4 x 14:55 = 59min 40sec … it's < 60min!
- Notification not sent until 75min!
- Scheduling is geared +/- to reduce load!

Use Hostgroups

Using hostgroups in your service definitions is one of the most powerful features of Nagios.

Common services have the same threshold for all hosts. Instead of creating individual services for each host you monitor, a service can be assigned to multiple hosts using a hostgroup. What this means is that you only need to have the service defined once, and when you want to tweak the thresholds, you only need to change it in one location and all hosts will receive the updated thresholds.

If you have a host group called windows_servers, whenever you add a new windows server, it is just a matter of adding that server to the hostgroup and that host gets that bunch of common checks. This is great for consistent monitoring; ensuring standards are applied and reducing management overhead.

Use Contactgroups

One of the most common support questions we get asked is how to add or remove a contact to or from a bunch of objects? If you don't have the enterprise edition license, then you don't get access to the bulk modification tool that allows you to do this.

However just changing just the contacts can lead to human error. It is very easy to make mistakes and before you know it a notification was not sent to the correct people.

Using a contact group is a much better method. It is so much easier to go in and add or remove a contact from a contactgroup and instantly all the objects that use this group will be updated. Even if there is only one member to a contact group it still makes administration so much easier.

If you've not activated the trial of enterprise edition, this is a great way make use of the bulk modification tool to implement contactgroups and remove the individual contacts. Once your standard is in place, administration will be so much easier.

Configuration Wizards

Configuration wizards are a great ice breaker for people who are new to nagios, you didn't need to learn how a plugin worked or how to create a command definition followed by service definitions, you just pointed and clicked.

The downside to wizards is that they create a lot of services. In a large scale monitoring environment, you might use services that are applied to hostgroups which reduces administrative overhead. Using wizards doesn't really apply to these environments however they are a great primer for setting up initial services, from there you can modify them in Configuration Manager.

- Pros
    - Great for getting up and running quickly
    - No need to learn how a plugin works
- Cons
    - Creates individual services
    - More work later when enforcing "standards"

Templates

Templates are powerful when used for the right purposes. A good example is how the XI Configuration Wizards use templates for the host objects. The host object template has a standard ICMP up/down check. This means if you ever wanted to change the thresholds, you could change the template and then all hosts using that template will get the updated check.

You can use multiple templates in a layering fashion. As Nagios core reads the object definition, it looks at the first template and obtains the settings. It then looks at the next template and layers those settings over the top of the previous settings. This continues and builds the final object. In the XI GUI, with multiple templates, the template at the bottom of the list will be overwritten by a template above it. Object directives can be set to inherit that setting from a template or ignore it. Other settings can be additive, like hosts, hostgroups, contacts, contactgroups.

For example, you might have a master template that defines the base settings all services should use. However, you have a bunch of service checks that require a specific time period. Create a separate template that uses this time period and put that template at the top of the chain. The final service object that is created will use the specific time period.

You can even create an empty template that uses a combination of other templates, this way you can use the master templates across all your objects and easily add / remove other templates to the master template, in turn reducing your administrative overhead. Be careful not to add more administrative overhead though.

User Macros – resources.cfg

User macros are a way of storing and referencing common items such as usernames and passwords.

Because you referencing the objects as a macro, the actual value is not visible in the object definitions.

It also allows special characters to be used like an exclamation mark. Normally when an exclamation mark is used in a command_name directive, its purpose is to split up the different arguments, so by storing it in a user macro it works around the problem.

```
$USER1$ = /usr/local/nagios/libexec
```

Custom Object Variables

Custom object variables are one of the lesser-known features of Nagios.

It allows you to define your own variables to use in your object definitions, this makes Nagios very flexible and very powerful.

A good example is if each windows host had its own custom check_nt password. What you can do is store that password in the host object and then from your service objects you can reference the password. It also means that you can still have just one command that can be used by many hosts, reducing administrative overhead.

- e.g. hosts have their own check_nt password.
- Define _CHECK_NT_PASSWORD in host object.
  - **Misc Settings tab > Manage Free Variables** button.
- In command definitions reference it as:

  ```
  $_HOSTCHECK_NT_PASSWORD$
  ```

MRTG Clean Configs

Your MRTG configs may be collecting more than you think.

In Nagios XI, the **Network Switch / Router wizard** uses MRTG to collect the monitoring data from the network device.

The configuration files for MRTG are created with the program cfgmaker. While you may have selected to only monitor a handful of ports from your network device, MRTG will collect data from all the interfaces. This creates extra network traffic and I/O.

```
/etc/mrtg/conf.d/*.cfg files
```

You can edit these MRTG config files and comment out the ports for which you do not need data to be collected for. Each port consists of about 37 lines in total.

Non-interfaces like VLANs, can also be commented out, unless you want to monitor them.

Finishing Up

This completes the documentation on Nagios XI 2024 Best Practices. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library