Nagios XI - Hardware Requirements - Baseline Testing

Article Number: 523 | Rating: 1/5 from 1 votes | Last Updated: Mon, Jan 1, 2018 at 10:18 PM

Overview

This KB article shows you how to setup a Nagios XI server to perform baseline testing. Baseline testing allows you to determine if an XI server is capable of handling the amount of cl

It is important to identify that every customer's monitoring setup is different depending on the type of monitoring being performed. One customer may have SNMP checks while anothe server depending on the CPU, memory and disk I/O utilized by the plugin.

The purpose of the following guide is to remove the type of plugin being used from the equation, in order to focus purely on the "moving parts" that make up a Nagios XI server and t performed is to simulate a major outage that will cause all of the checks to go into a DOWN / WARNING / CRITICAL / UNKNOWN state. This will test if your XI server is still responsive

This guide is for Nagios XI 5 onward. XI 5 introduced an API which this guide utilizes, hence XI 5 is a requirement.

Deploy XI Server

This KB article uses the latest available VMware 64-bit virtual machine (Nagios XI 5.3.1 at the time of this writing).

If you are not going to be running Nagios XI in a VMware virtual machine you should install XI on the hardware or virtualization platform that you will be using in production.

Once your XI server is up and running you can proceed to the next step.

Disable Flap Detection

Flap detection allows Nagios to detect hosts and services that are "flapping". Flapping occurs when a service or host changes state too frequently, resulting in a storm of problem an

In a testing environment this is a variable that will produce inconsistent results. With this in mind, disabling flap detection globally will ensure you have a consistent testing method. To

- Navigate to **Configure** > **Core Config Manager**
- Click **CCM Admin** -> **Core Configs**, (the **General [nagios.cfg]** tab will be selected)
    - Find the directive `enable_flap_detection=`
    - Change it to `enable_flap_detection=0`
    - Click **Save Changes**
- Navigate to the **Admin** menu
- **System Information** -> **System Status**
    - Find the component **XI System Component Status**
    - For **Monitoring Engine** click the **gear** icon and click **Restart**

Nagios XI is now running with flap detection disabled. If you wish to re-enable it afterwards for production use, follow the steps above, instead changing `enable_flap_detection`

Create Test Plugin

The cornerstone of this baseline test is to use a plugin that detects if a file exists on the Nagios XI server. If it finds the file it will report an OK status and if it doesn't it will report a CR

In addition, the plugin will return a string of 1024 random characters for the status output and performance data to allow graphs to be generated (a single data source of a random r

This test plugin is simple in the sense that it allows you to delete the file it is looking for, which instantly starts the major outage. You can then re-create the file to stop the major outa

The test plugin is what will be used for the service checks.

Host checks will use the standard ICMP check (ping) to determine UP / DOWN status. A simple firewall rule will allow us to simulate the hosts being down and hence cause an outage.

Open an SSH session as the root user to your Nagios XI server.

Execute this command:

```
vi /usr/local/nagios/libexec/check_file_test.sh
```

This opens the vi text editor.

Press `i` on the keyboard to go into insert mode.

Paste the following text into the SSH session:

```
#!/bin/sh
test_file='/tmp/test_file.txt'
random_text=$(tr -dc 'a-zA-Z0-9 ' < /dev/urandom | head -c 1024)
if [ -a $test_file ]
then
    perfdata=$[ ( $RANDOM % 100 )  + 1 ]
    echo "File does exist and this is a lot of output $random_text|ds1=$perfdata"
    exit 0
```

```
    exit 0
else
    echo "File does NOT exist and this is a lot of output $random_text|ds1=0"
    exit 2
fi
```

Press **Escape** on the keyboard to exit insert mode.

Type **:wq** and then press Enter (this will save the file and exit vi).

Execute these commands:

```
chown nagios:nagios /usr/local/nagios/libexec/check_file_test.sh
chmod +x /usr/local/nagios/libexec/check_file_test.sh
touch /tmp/test_file.txt
```

This completes creating the plugin.

Get API Key

Sections in this document will use commands in the SSH session to create objects in Nagios XI using the new XI 5 backend API. The API requires a key to be used to authenticate the

To obtain the key:

- Log into Nagios XI as **nagiosadmin**
- In the top right corner click the username **nagiosadmin**
    - You will be directed to the Account Information screen
    - Here you will see an API key, for example **tokunpg7**
    - You will need to use this key in several sections of the guide - copy and paste it into a text editor so it's easy to reference
    - This guide will reference the API key as **XXXXXX**

Create Initial Objects

The testing baseline determines that for every **1** host object there will be **9** service objects.

The host object will use a template which adds it to a hostgroup.

The nine services will all use the hostgroup in the service definition. Hence every time a new host is added it will receive the nine service objects. This allows for a scalable testing en

1 host object + **9** service objects = **10** checks in total per host.

These steps will create the objects described above.

Create Hostgroup

The following command requires the API key:

```
curl -XPOST "http://localhost/nagiosxi/api/v1/config/hostgroup?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&alias=all_t
```

Now open the Nagios XI web interface and navigate to **Configure** menu -> **Core Config Manager**

Create Service Command

- **Commands** > **>_ Commands**
    - Click the **+ Add New** button
        - Command Name: `check_file_test`
        - Command Line: `$USER1$/check_file_test.sh`
        - Check the **Active** box
        - Click **Save**

You will continue using the Core Config Manager in the next step.

Create Host Template

- **Templates** -> **Host Templates**

- Search for the template `linux-server`
- Click the **Copy** icon to make a copy of it
- Click the new copy called `linux-server_copy_1`
  - Template Name: `testing_host_template`
  - Click the **Manage Hostgroups** button
    - Select `all_testing_hosts` in the left pane
    - Click the **Add Selected** button
    - Click **Close**
- Check the **Active** box
- Click the **Check Settings** tab
  - Check interval: `5`
  - Retry interval: `1`
  - Max check attempts: `5`
- Click **Save**

You will continue using the Core Config Manager in the next step.

Create Service Template

- **Templates** -> **Service Templates**
  - Search for the template `local-service`
  - Click the **Copy** icon to make a copy of it
  - Click the new copy called `local-service_copy_1`
    - Template Name: `testing_service_template`
  - Check command:
    - Select **`check_file_test`** from the drop down list
  - Check the **Active** box
  - Click the **Check Settings** tab
    - Check interval: `5`
    - Retry interval: `1`
    - Max check attempts: `5`
  - Click **Save**

You will continue using the Core Config Manager in the next step.

Apply Configuration

- **Quick Tools** -> **Apply Configuration**
- Click the **Apply Configuration** button

Once the Apply Configuration process has completed you can proceed to the next step.

Create The 1st Host

The following command requires the API key:

```
curl -XPOST "http://localhost/nagiosxi/api/v1/config/host?apikey=XXXXXX&pretty=1" -d "host_name=HOST_1&address=127.0.0.1&use=testing_ho
```

Create The 9 Services

The following commands require the API key:

```
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
```

```
curl -XPOST "http://localhost/nagiosxi/api/v1/config/service?apikey=XXXXXX&pretty=1" -d "hostgroup_name=all_testing_hosts&service_descr
```

Apply Config

The following command requires the API key:

```
curl -XGET "http://localhost/nagiosxi/api/v1/system/applyconfig?apikey=XXXXXX&pretty=1"
```

This completes creating the initial objects. The next section discusses making a script to create **X** amount of host objects which in turn will create the services for each host.

Add More Hosts

As explained earlier, for every **1** host object there will be **9** service objects which is **10** checks in total.

A little mathematics is required to determine how many hosts need to be created to create the total amount of objects.

- Total checks = Number Of Hosts x 10
- 100 checks = 10 hosts
- 1,000 checks = 100 hosts
- 5,000 checks = 500 hosts
- 10,000 checks = 1,000 hosts
- 20,000 checks = 2,000 hosts

Follow these steps to create a script which can create the amount of hosts you require.

NOTE:

- The script has **two** lines that require the API key
- The second line in the script needs to be adjusted so it creates the correct amount of host objects you require
    - `for p in {2..10}`
    - In that example `10` hosts = `100` checks
    - It starts at number `2` because the first host object has already been created

Execute this command:

```
vi /tmp/add_hosts.sh
```

This opens the vi text editor.

Press **i** on the keyboard to go into insert mode.

Paste the following text into the SSH session:

```
#!/bin/sh
for p in {2..10}
do
        host_name="HOST_$p"
        curl -XPOST "http://localhost/nagiosxi/api/v1/config/host?apikey=XXXXXX&pretty=1" -d "host_name=$host_name&address=127.0.0.1&us
done
curl -XGET "http://localhost/nagiosxi/api/v1/system/applyconfig?apikey=XXXXXX&pretty=1"
```

Press **Escape** on the keyboard to exit insert mode.

Type **:wq** and then press Enter (this will save the file and exit vi).

Execute this command:

```
chmod +x /tmp/add_hosts.sh
```

The script is ready to be run to create the the host objects, execute this command:

```
/tmp/add_hosts.sh
```

Depending on how many host objects you configured it to create will determine how long the script will take to run. Realistically it shouldn't take any longer than 30 seconds.

Once the script has completed running you can go to the XI GUI and you'll see the new host and services in a PENDING state, transitioning into an UP/OK state as they are checked.

Simulate A Major Outage

There are two commands you will execute to simulate a major outage:

- Create a temporary firewall rule to simulate the hosts being down and hence cause an outage (ICMP drop)
- Delete the test file the `check_file_test.sh` plugin is looking for causing the services to go critical

Execute these commands:

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j DROP
rm -f /tmp/test_file.txt
```

You will need to wait for all the host and services to go into a HARD state, this will ensure global event handlers are run, it might take 5 - 10 minutes for this to occur. You should view t

Observe

While Nagios XI is dealing with this major outage there are several commands that you can execute to observe the status/health of your Nagios XI server.

```
top
free -m
icps -q
iotop
```

The main two considerations are CPU and memory usage. If your Nagios XI server is not handling the major outage, you'll need to add more resources to your Nagios XI server and

Running Nagios XI as a virtual machine allows you to easily add more resources. If you're intending on purchasing a physical server to run Nagios XI you should consider testing it in a

Return To Normal Operations

There are two commands you will execute to recover from the "major outage":

- Restart the firewall to discard the temporary ICMP drop rule that was created
- Re-create the test file the `check_file_test.sh` plugin is looking for

Execute these commands:

```
service iptables restart
touch /tmp/test_file.txt
```

You will need to wait for all the host and services to return to an UP / OK state. You should view the **Tactical Overview** page to see where the services are up to.

Other Testing

You may be wondering how you can simulate other "real" outages in your environment.

As demonstrated earlier, using a firewall rule to drop outbound traffic is an easy way to simulate an outage. You could create rules that:

- Drop outbound SNMP traffic
- Drop outbound NRPE traffic
- Drop outbound check_nt traffic
- Drop outbound WMI traffic
- Drop outbound HTTP/HTTPS traffic
- Drop outbound traffic to an entire subnet
- Drop inbound traffic from an entire subnet
- Drop inbound traffic from external workers (like Mod-Gearman)

Final Thoughts

For any support related questions please visit the Nagios Support Forums at:

http://support.nagios.com/forum/

Posted by: **tlea** - Sun, Jul 17, 2016 at 11:40 PM. This article has been viewed 1061 times.

Online URL: https://support.nagios.com/kb/article/nagios-xi-hardware-requirements-baseline-testing-523.html