How to Monitor JMX in Nagios XI 2024

Purpose

This document will cover how to monitor Java application servers using the `check_jmx` plugin within Nagios XI, for users to be notified when java applications are not functioning properly.

Prerequisites

This document assumes you have the following:

- A remote server running JMX
- Nagios XI server with a network route to the JMX server

The monitoring plugin used in this documentation will be executed on the remote server. This means that the NRPE agent will need to be installed on your remote server. Follow our Linux Agent installation document below:

Installing The XI Linux Agent

The documentation will assume that you have installed NRPE before continuing.

Editing Files

In many steps of this documentation, you will be required to edit files. This documentation will use the vi text *editor*. When using the vi editor:

- To make changes press **i** on the keyboard first to enter insert mode
- Press **Esc** to exit insert mode
- When you have finished, save the changes in vi by typing `:wq` and press **Enter**

check_jmx Plugin Overview

This document will specifically cover the configuration of the `check_jmx` plugin, which can be downloaded from:

https://github.com/WillPlatnick/jmxquery

The check_jmx plugin can:

- Monitor standard Java JMX implementation by exposing memory, threads, OS, and garbage collector parameters
- Monitor Tomcat's multiple parameters such as requests, processing time, threads, etc
- Monitor Spring framework by exposing Java beans parameters to JMX

- Expose any attributes to JMX by declaration or explicitly
- Monitor any java object/attribute accessible through JMX, including jboss objects and beans

Installing The Plugin

Establish a terminal session to your JMX server and execute the following commands to download and install the plugin:

```
cd /usr/local/nagios/libexec/
wget -O check_jmx "https://raw.-
githubusercontent.com/WillPlatnick/jmxquery/master/plugin/check_jmx"
wget -O jmxquery.jar "https://-
github.com/WillPlatnick/jmxquery/raw/master/plugin/jmxquery.jar"
chmod +x *jmx*
chown nagios:nagios *jmx*
```

Test The Plugin

In your terminal session execute the following commands to test the plugin (it's a long command that wraps over three lines):

```
./check_jmx -U service:jmx:rmi:///jndi/rmi://localhost:<port>/jmxrmi -O
java.lang:type=Memory -A HeapMemoryUsage -K used -I HeapMemoryUsage -J used
-vvvv -w 4248302272 -c 5498760192
```

The expected output of the above command should be similar to:

```
JMX OK - HeapMemoryUsage.used=79112000 |
HeapMemoryUsage.usedd=79112000,committed=954204160;init=964689920;max=954204
160;used=79112000
```

You will have noticed that the command had the following:

```
localhost:<port>
```

You will need to replace the port your JMX application uses, in all future examples port 7199 will be used.

Configure NRPE

For Nagios XI to execute this plugin, you need to define a command for this plugin in the `nrpe.cfg` on the JMX server. Edit the `/usr/local/nagios/etc/nrpe.cfg` file by executing the following command:

```
vi /usr/local/nagios/etc/nrpe.cfg
```

Add the following line to the end of the file:

```
command[check_jmx]=/usr/local/nagios/libexec/check_jmx $ARG1$
```

After saving these changes, restart the xinetd service on the JMX Server (or the nrpe service if you compiled from source) by running the following command:

```
service xinetd restart
```

Now to test the check from the Nagios XI server. Establish a terminal session to your Nagios XI server and execute the following command, making sure to replace `<jmx_server_ip>` with the IP address of your JMX server (it's a long command that wraps over four lines):

```
/usr/local/nagios/libexec/check_nrpe -H <jmx_server_ip> -c check_jmx -a '-U
service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O java.lang:type=Memory
-A
HeapMemoryUsage -K used -I HeapMemoryUsage -J used -vvvv -w 4248302272 -c
5498760192'
```

You should see output like:

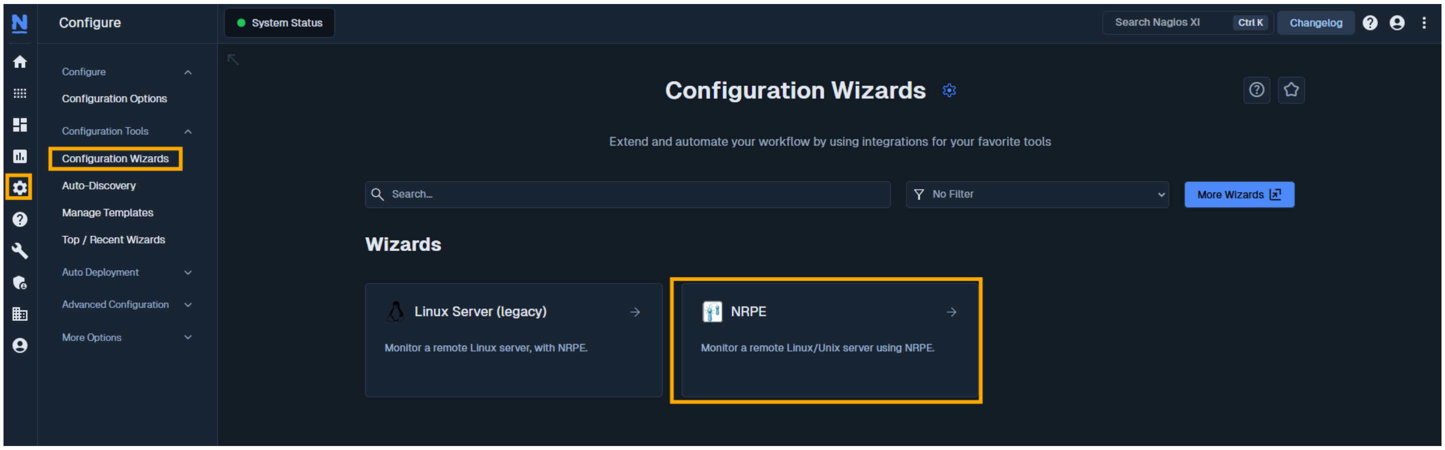
```
JMX OK - HeapMemoryUsage.used=94443776 |
HeapMemoryUsage.usedd=94443776,committed=954204160;init=964689920;max=954204
160;used=94443776
```

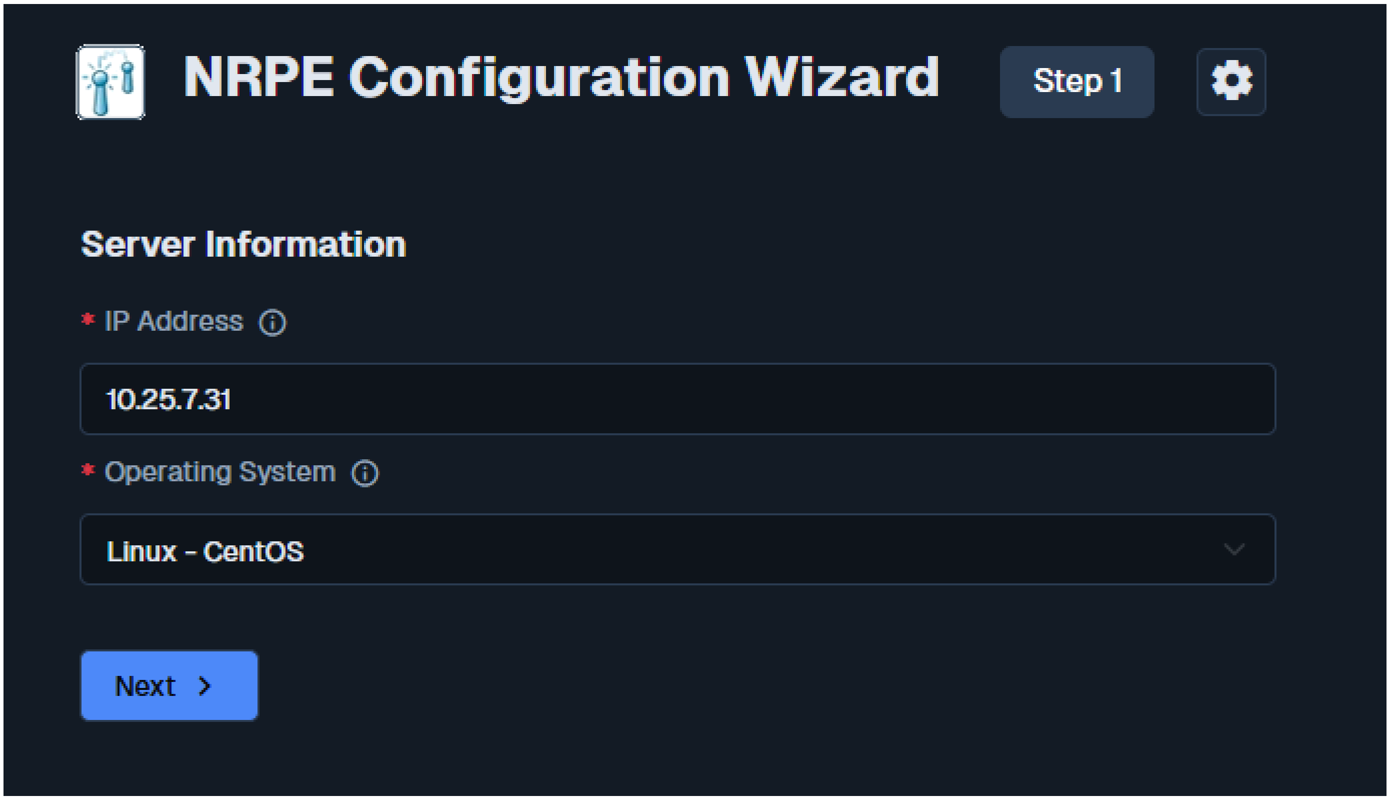
Now that NRPE has been configured correctly on your JMX server the next step is to create the monitoring configuration in Nagios XI.

Create Nagios Monitoring Objects

In this example you will use the **NRPE Configuration Wizard** which will create the host and service objects.

Navigate via the top menu bar to **Configure > Run** a configuring wizard, and select the **NRPE** wizard. In the following screenshot you can see how the search field allows you to quickly find a wizard.





Server Information

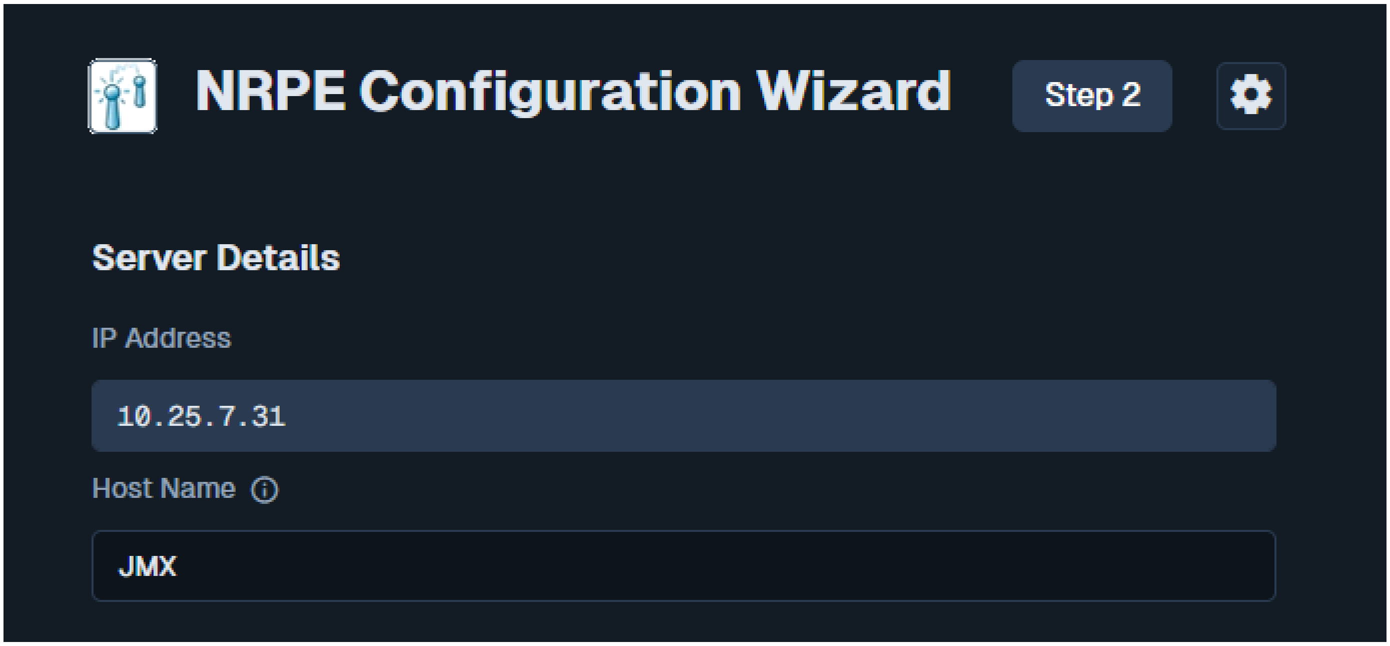
* IP Address

10.25.7.31

* Operating System

Linux - CentOS

1. On **Step 1** you will be asked to supply the **IP Address** of the JMX server.
    a. You will also have to select the **Operating System**; in this case it is CentOS.
    b. Click **Next** to progress to **step 2**.



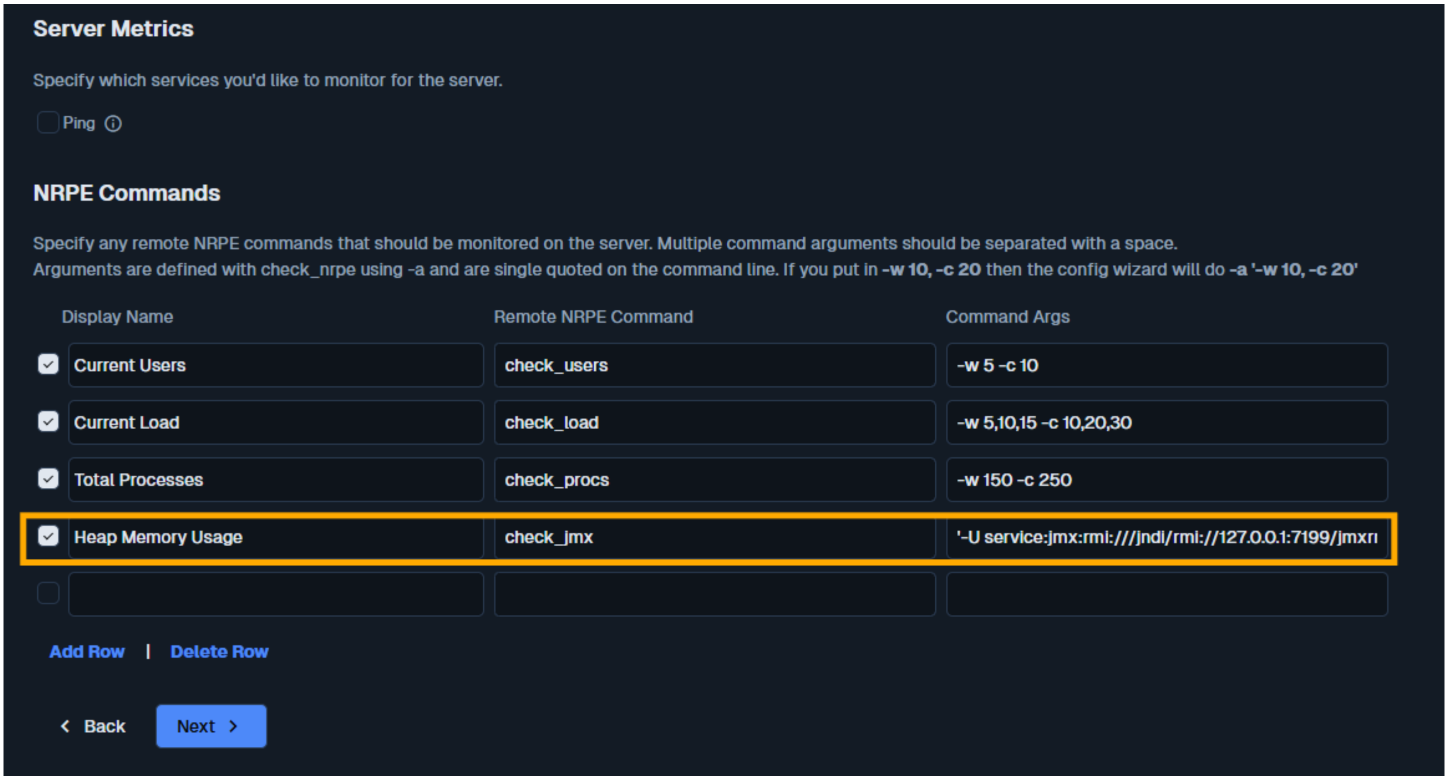
Server Details

IP Address: 10.25.7.31

Host Name: JMX

2. On **step 2** you will configure all the options for monitoring.
    a. To start off with make sure a valid **Host Name** has been entered.
    b. The **NRPE Agent** section can be ignored because you have already installed it.
    c. The NRPE wizard allows you to specify which **NRPE commands** should be executed and monitored and what display name (service description) should be associated with each command.
    d. In the screenshot below you can see the command has been defined for the **check_jmx** check
    e. In the **Command Args** field the following has been typed (the field is too short to display it all):

```
'-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O java.lang:type=Memory -A
HeapMemoryUsage -K used -I HeapMemoryUsage -J used -vvvv -w 4248302272 -c
5498760192'
```



Server Metrics

Specify which services you'd like to monitor for the server.

□ Ping

NRPE Commands

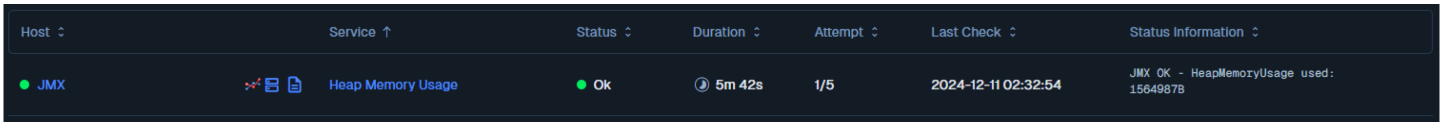
Specify any remote NRPE commands that should be monitored on the server. Multiple command arguments should be separated with a space.
Arguments are defined with check_nrpe using -a and are single quoted on the command line. If you put in **-w 10, -c 20** then the config wizard will do **-a '-w 10, -c 20'**

	Display Name	Remote NRPE Command	Command Args
☑	Current Users	check_users	-w 5 -c 10
☑	Current Load	check_load	-w 5,10,15 -c 10,20,30
☑	Total Processes	check_procs	-w 150 -c 250
☑	Heap Memory Usage	check_jmx	'-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrı
□			

3. Click **Next** and then complete the wizard by choosing the required options in **Step 3 – Step 5**.

   To finish up, click on **Finish** in the final step of the wizard. This will create new hosts and services and begin monitoring.

   Once the wizard applies the configuration, click the **View status details** for xxxxx link to see the new host and service that was created.



Host	Service	Status	Duration	Attempt	Last Check	Status Information
● JMX	Heap Memory Usage	● Ok	5m 42s	1/5	2024-12-11 02:32:54	JMX OK - HeapMemoryUsage used: 1564987B

You are now monitoring your JMX server.

Note: In the wizard, in the Command Args field, pay special attention to how the entire argument is surrounded by single quotes.

```
'-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O
java.lang:type=Memory -A HeapMemoryUsage -K used -I HeapMemoryUsage -J used
-vvvv -w 4248302272 -c 5498760192'
```

Using check_jmx With Authenticated JMX Servers

Some JMX servers require authentication, the examples so far have assumed the

local host is allowed to connect to the JMX server without authentication. The check_

jmx plugin allows you to provide credentials using the -username and -password arguments. Here is an example when testing from the Nagios XI server:

```
/usr/local/nagios/libexec/check_nrpe -H <jmx_server_ip> -c check_jmx -a '-U
service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O java.lang:type=Memory
-A
HeapMemoryUsage -K used -I HeapMemoryUsage -J used -vvvv -w 4248302272 -c
5498760192 -username admin -password welcome123'
```

Additional check_jmx Argument Configurations

Below you will find a few different examples of the types of checks that can be done

with `check_jmx`. Your JMX server may or may not support some of the examples but

will most likely support many more options.

Note: The `-w xxx` and `-c yyy` threshold values need to be replaced with values relative to the object being monitored.

Garbage Collection:

```
-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O java.lang:-
type=GarbageCollector,name=ConcurrentMarkSweep -A LastGcInfo -K duration -u
ms -vvvv -w xxx -c yyy
```

Thread Count:

```
-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O
java.lang:type=Threading -A
ThreadCount -w xxx -c yyy
```

Available Connections in Pool:

```
-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O jboss.jca:-
name=JmsXA,service=ManagedConnectionPool -A AvailableConnectionCount -w
xxx -c yyy
```

System Load:

```
-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O java.lang:-
type=OperatingSystem -A SystemLoadAverage -w xxx -c yyy
```

Regular Expression Matching:

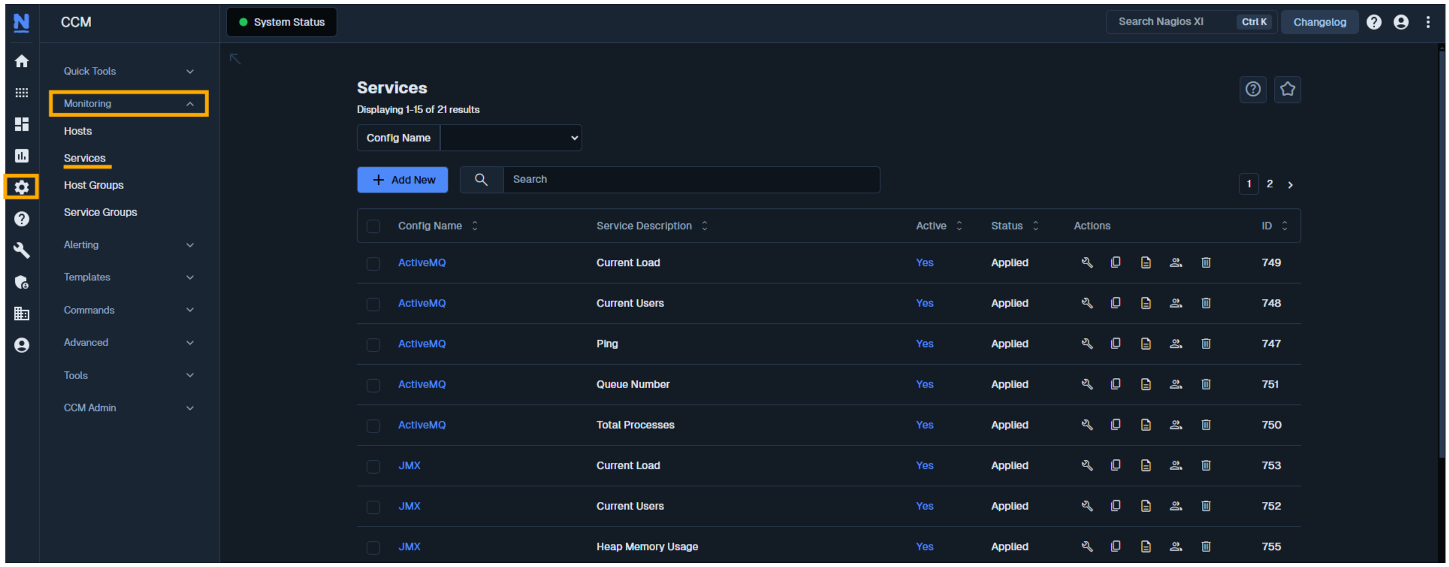
```
-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O
<object>:name=<Bean
name> -A Status -w <regular expression> -c <regular expression>
```

To add additional services, you can:

- Run the **NRPE Wizard** again
- Copy an existing service and change the required options

The following steps show you how to copy an existing service to create an additional service. This example will add the Thread Count check shown above.

1. In Nagios XI, navigate via the top menu bar to **Configure > Core Configuration Manager** and then **Monitoring** > **Services**.



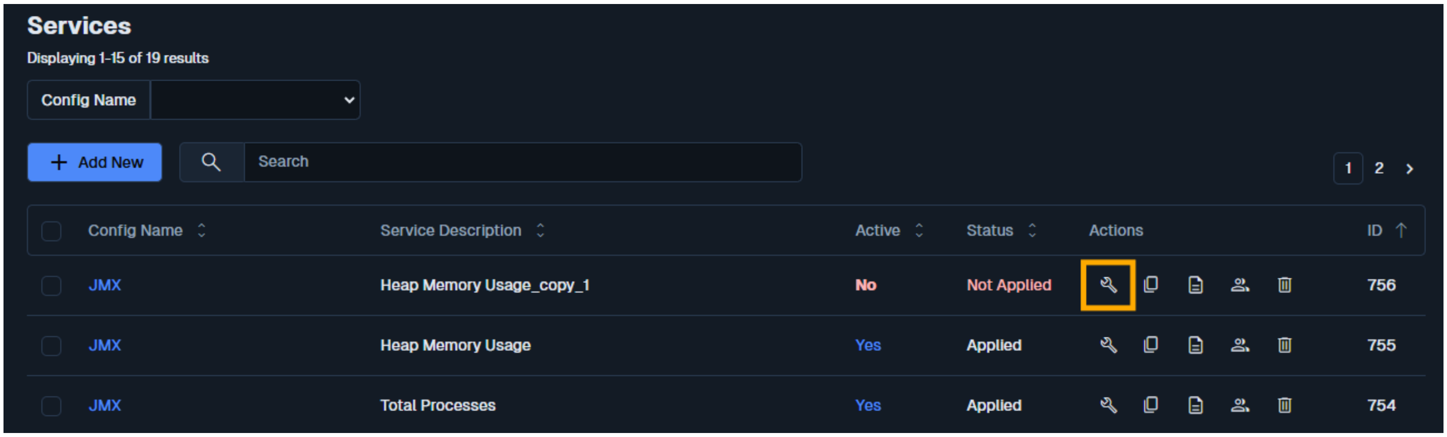
Services

Displaying 1-15 of 21 results

Config Name

Config Name	Service Description	Active	Status	Actions	ID
ActiveMQ	Current Load	Yes	Applied		749
ActiveMQ	Current Users	Yes	Applied		748
ActiveMQ	Ping	Yes	Applied		747
ActiveMQ	Queue Number	Yes	Applied		751
ActiveMQ	Total Processes	Yes	Applied		750
JMX	Current Load	Yes	Applied		753
JMX	Current Users	Yes	Applied		752
JMX	Heap Memory Usage	Yes	Applied		755

2. Click the **Copy** icon for the existing service. This will create a copy of the existing service and its name will be appended with _copy_1.



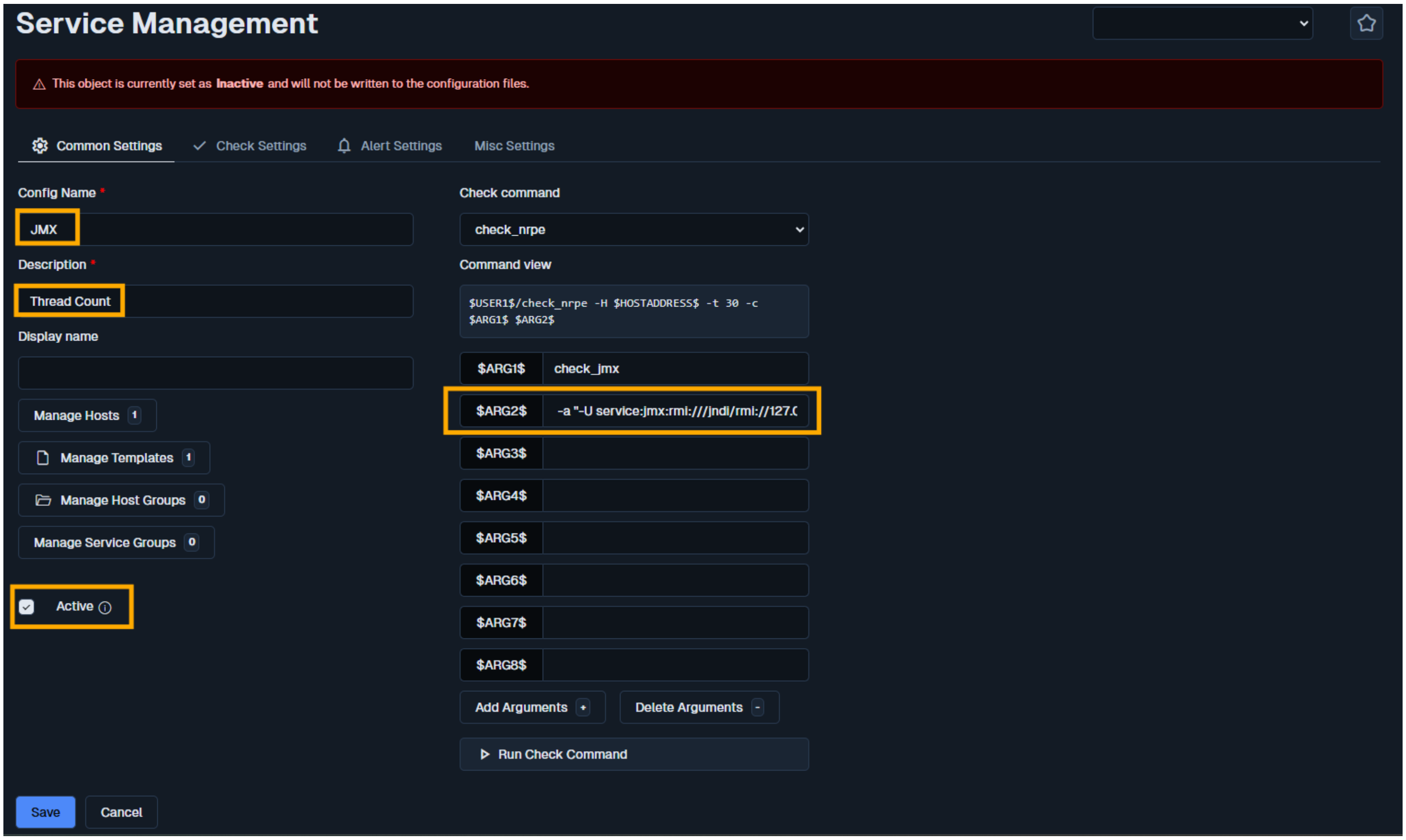
Services

Displaying 1-15 of 19 results

Config Name

Config Name	Service Description	Active	Status	Actions	ID
JMX	Heap Memory Usage_copy_1	No	Not Applied		756
JMX	Heap Memory Usage	Yes	Applied		755
JMX	Total Processes	Yes	Applied		754

3. Click the **Modify icon** to edit the copied service.



Service Management

⚠ This object is currently set as **Inactive** and will not be written to the configuration files.

Config Name *: JMX

Description *: Thread Count

Display name:

☑ Active

Check command: check_nrpe

Command view:

```
$USER1$/check_nrpe -H $HOSTADDRESS$ -t 30 -c
$ARG1$ $ARG2$
```

$ARG1$: check_jmx

$ARG2$: -a "-U service:jmx:rmi:///jndi/rmi://127.(

$ARG3$:

$ARG4$:

$ARG5$:

$ARG6$:

$ARG7$:

$ARG8$:

4. In the **Config Name** field remove the **'_copy_1'**.

5. Change the **Description** field to Thread Count.

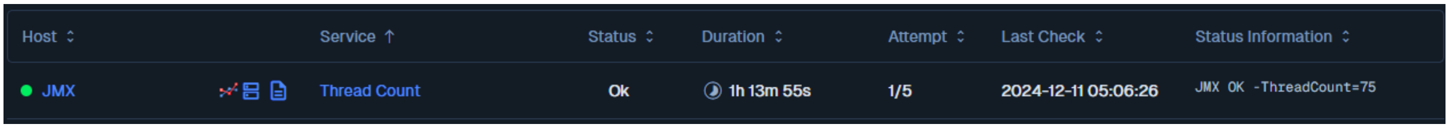
6. In the `$ARG2$` field change the text so that it correctly matches the check being performed, this is what is being used in this example:

```
-a '-U service:jmx:rmi:///jndi/rmi://127.0.0.1:7199/jmxrmi -O
java.lang:type=Threading -A ThreadCount -w 250 -c 500'
```

7. Finally make sure the **Active** checkbox is checked.

8. Click the **Save** button and then **Apply Configuration**.

Once the configuration has been applied you will have a new **Thread Count** service:



Host	Service	Status	Duration	Attempt	Last Check	Status Information
● JMX	Thread Count	Ok	1h 13m 55s	1/5	2024-12-11 05:06:26	JMX OK -ThreadCount=75

That is how you can use **Core Configuration Manager** to copy an existing service to create a new service.

More Information:

Using Configuration Wizards

Finishing Up

This completes the documentation on monitoring JMX in Nagios XI. 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