Managing Plugins in Nagios XI 2024

Purpose

This document describes how to manage plugins on your Nagios XI system, including: finding and installing new plugins, defining commands, and using them in your services. Plugins are how Nagios talks to the devices you monitor, they are instrumental to extending the functionality of Nagios XI.

Locate Your Plugin

There are thousands of community-created Nagios plugins available for download at the Nagios Exchange.

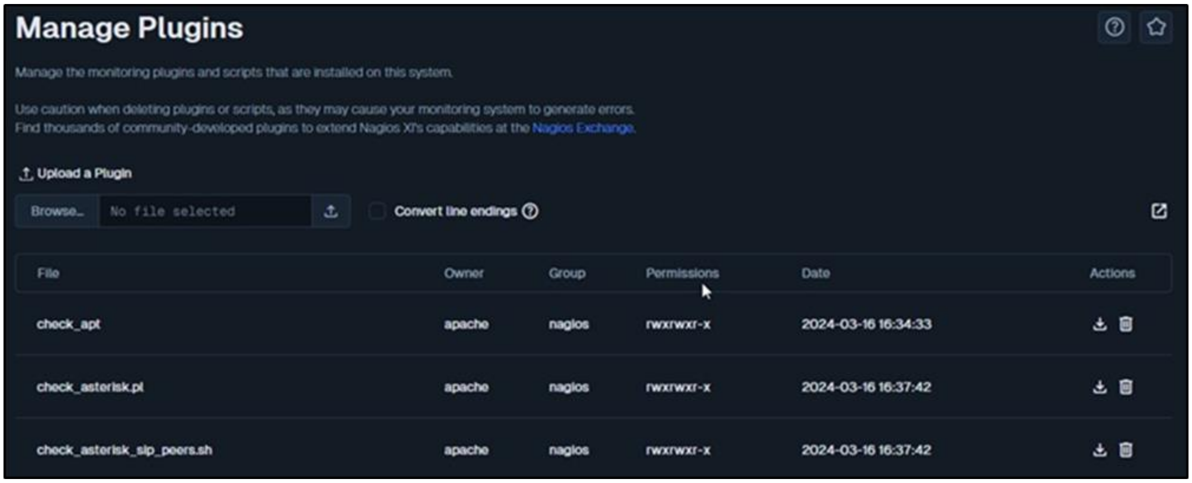
You can also find instructions for creating your own plugin here: Nagios Plugins Development Guidelines

This guide is going to use the `countdown_to_date` plugin as the example in this documentation, it can be downloaded here: countdown_to_date download

Download the `countdown_to_date.zip` file and extract it. The `countdown_to_date.php` file that was in the zip file will be used in the next step.

Install Your Plugin

1. Installing a new plugin in Nagios XI is very simple. Navigate to **Admin > System Extensions > Manage Plugins**.



Manage Plugins

Manage the monitoring plugins and scripts that are installed on this system.

Use caution when deleting plugins or scripts, as they may cause your monitoring system to generate errors.
Find thousands of community-developed plugins to extend Nagios XI's capabilities at the Nagios Exchange.

File	Owner	Group	Permissions	Date	Actions
check_apt	apache	nagios	rwxrwxr-x	2024-03-16 16:34:33	
check_asterisk.pl	apache	nagios	rwxrwxr-x	2024-03-16 16:37:42	
check_asterisk_sip_peers.sh	apache	nagios	rwxrwxr-x	2024-03-16 16:37:42	

2. Click the **Browse** button to select the new plugin you want to install.
3. Select the plugin file `countdown_to_date.php` that you previously downloaded using the window that appears and click **Open**.

4. The **Convert line endings** check box will convert the plugin's line endings to UNIX line endings.
   - Sometimes the files saved on a Windows computer will not work on the Nagios XI server, this check box fixes the problem
   - This process will not break already UNIX-formatted files
5. Click the **Upload Plugin** button to upload the plugin.
6. Once the plugin is uploaded, you will receive a message saying it was installed and it will appear in the list of plugins.

Test Your Plugin From The Command Line

The next step is to test that the plugin works and understand what arguments the plugin allows. This is easiest to test using a terminal session. Plugins are located in the `/usr/local/nagios/libexec` directory.

```
cd /usr/local/Nagios/libexec
```

Establish a terminal session to your Nagios XI server as the root user and execute the following command:

```
./countdown_to_date.php —help
```

I'm not going to paste the full output here however at the top of the output is the usage:

```
Usage:     countdown_to_date.php —date <date> --warning <wdays> --crit-ical <cdys>
```

So with that in mind, lets test using some real values:

```
./ countdown_to_date.php —date 2024-03-19 —warning 60 —critical 30
```

This command will produce results which should be similar to the following:

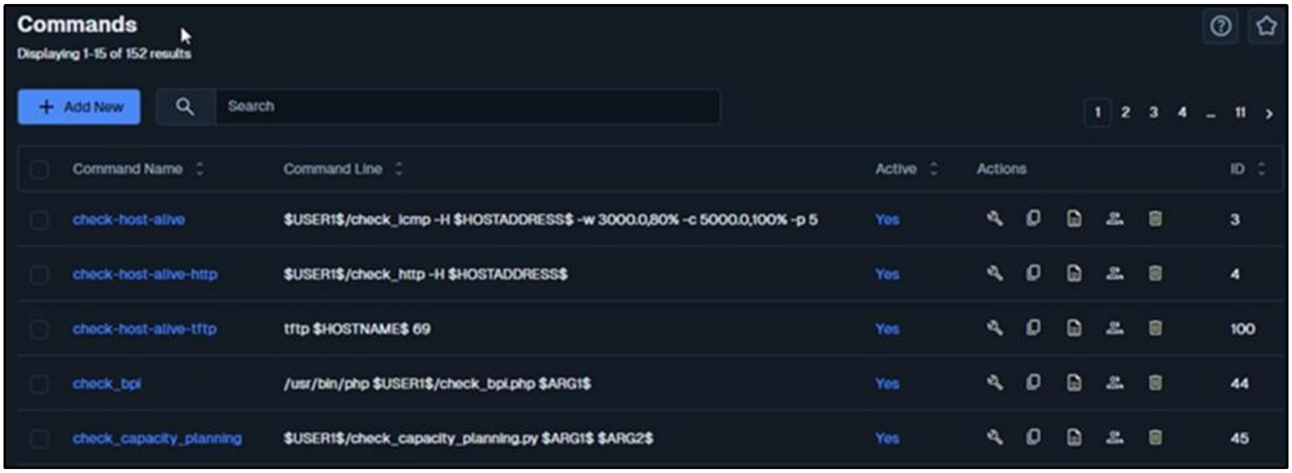
```
Critical: 1 days until 2024-03-19
```

That example tested the plugin and helped us determine what arguments are required, `--date, --warning`, and `--critical`. The next step will show you how to define a command for the plugin.

Define A Command

Now that your plugin is available and you know the command line arguments to run it, you should define a command in Nagios XI.

1. In the Nagios XI web interface navigate to **Configure > Core Config Manager > Commands**.



	Command Name	Command Line	Active	Actions	ID
□	check-host-alive	$USER1$/check_icmp -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -p 5	Yes		3
□	check-host-alive-http	$USER1$/check_http -H $HOSTADDRESS$	Yes		4
□	check-host-alive-tftp	tftp $HOSTNAME$ 69	Yes		100
□	check_bpi	/usr/bin/php $USER1$/check_bpi.php $ARG1$	Yes		44
□	check_capacity_planning	$USER1$/check_capacity_planning.py $ARG1$ $ARG2$	Yes		45

2. Click the **Add New** button and you will need to provide the following details:

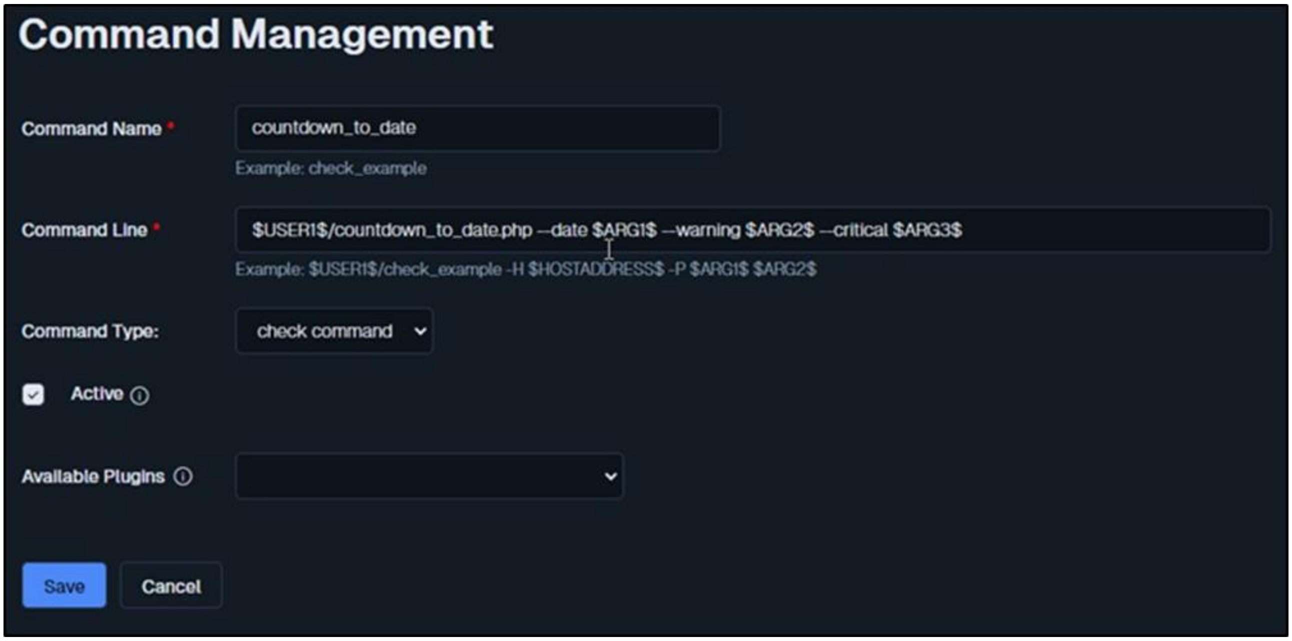
   `Command Name: countdown_to_date`

   Command Line: `$USER1$/countdown_to_date.php --date $ARG1$ --warning $ARG2$ --critical $ARG3$`

   `Command Type: check command`

3. Make sure the **Active** box is checked.

4. The final command definition should resemble the screenshot:



Command Management

Command Name * countdown_to_date

Example: check_example

Command Line * $USER1$/countdown_to_date.php --date $ARG1$ --warning $ARG2$ --critical $ARG3$

Example: $USER1$/check_example -H $HOSTADDRESS$ -P $ARG1$ $ARG2$

Command Type: check command

☑ Active

Available Plugins

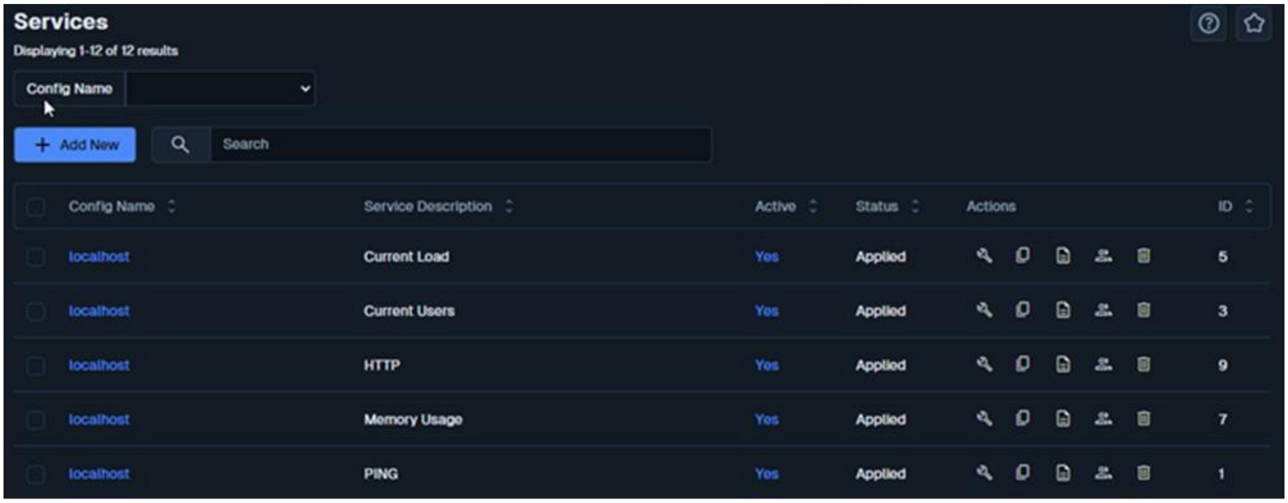
5. Click **Save** when finished.
    a. `$USER1$` references the directory `/usr/local/nagios/libexec` from the resources.cfg file. This is the default path for plugins and scripts in Nagios XI.
    b. The `$ARG1$, $ARG2$` and `$ARG3$` are macros that are a placeholder for an argument you will specify in the service that uses this command.
    c. Macros allow you to use the same command in different services and each service can provide different values. This is one of the key features of Nagios XI that allows for flexible and dynamic monitoring configurations.
6. Click the **Apply Configuration** button to commit these changes to Nagios XI.

At this point you have a command defined; in the next step you will be shown how to assign the command to a service.

Add A Service

For the purpose of keeping things simple, we are going to create a service for the Nagios XI locahost. There are already existing services for the localhost so we'll copy an existing service to use it to create the new service. By copying an existing service it allows us to focus just on the command part of the service, we won't need to focus on all the options as they will already be defined the service being copied.

1. Navigate to **Monitoring > Services** and search for localhost.



Services

Displaying 1-12 of 12 results

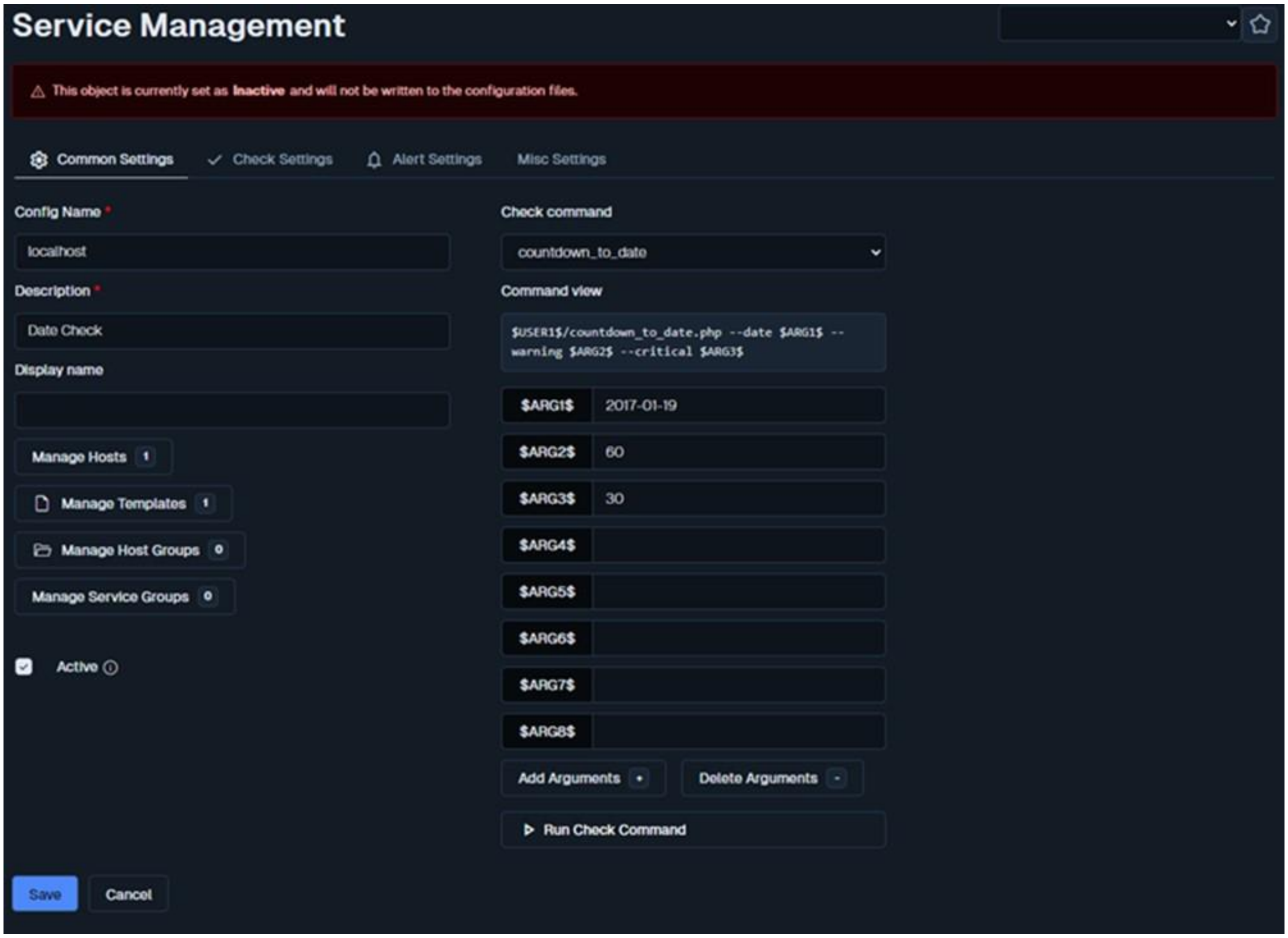
	Config Name	Service Description	Active	Status	Actions	ID
□	localhost	Current Load	Yes	Applied		5
□	localhost	Current Users	Yes	Applied		3
□	localhost	HTTP	Yes	Applied		9
□	localhost	Memory Usage	Yes	Applied		7
□	localhost	PING	Yes	Applied		1

2. We are going to copy the **Ping** service, click the **copy** icon in the **Actions** column for the **PING** service.
3. When the screen refreshes, click the copied service **localhost_copy_1** to edit it.
4. Update the service with the following details:

```
I       Config Name:
Localhost  l  Description
Date Check  l  Check
command:
countdown_to_date
l $ARG1$: 2017-01-19
I    $ARG2$: 60
l $ARG3$: 90
```

5. Make sure the **Active** box is checked.

6. The final command definition should resemble this screenshot:



Service Management

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

Config Name *: localhost

Description *: Date Check

Display name:

Check command: countdown_to_date

Command view:

```
$USER1$/countdown_to_date.php --date $ARG1$ --
warning $ARG2$ --critical $ARG3$
```

$ARG1$	2017-01-19
$ARG2$	60
$ARG3$	30
$ARG4$	
$ARG5$	
$ARG6$	
$ARG7$	
$ARG8$	

☑ Active

7. Click **Save** when finished and then click **Apply Configuration**.

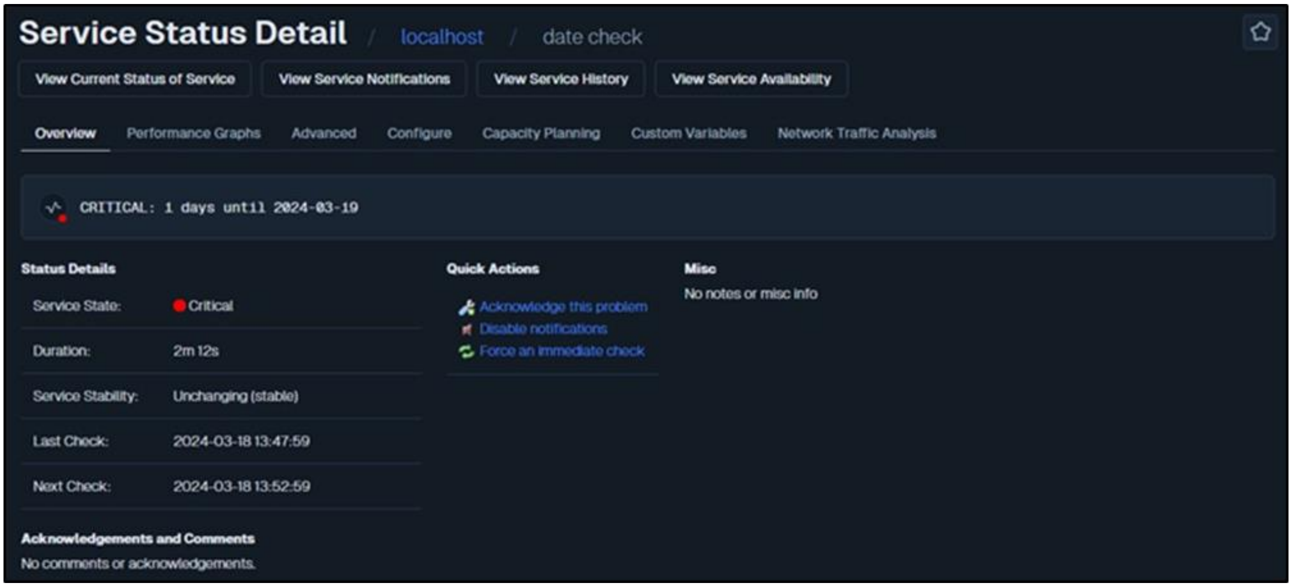
In the screenshot previously, you can see in the **Command View** field the command you created in the last step of this document. The `$ARGx$` macros refer to the fields underneath that allow you to provide the values you want to use for this particular service. For example you can see how the `--date` argument uses the `$ARG1$` macro which has the value 2017-01-19.

As explained earlier, macros allow you to use the same command in different services and each service can provide different values. This is one of the key features of Nagios XI that allows for flexible and dynamic monitoring configurations.

Verify Your Service Is Working

Now that you have created a service, you should check that it is working as expected. Use the **Search** field on the menu bar to search for localhost.

Click the **new service** to bring up the **Service Status Details** page.



Service Status Detail / localhost / date check

```
CRITICAL: 1 days until 2024-03-19
```

Status Details

Service State:	● Critical
Duration:	2m 12s
Service Stability:	Unchanging (stable)
Last Check:	2024-03-18 13:47:59
Next Check:	2024-03-18 13:52:59

Misc

No notes or misc info

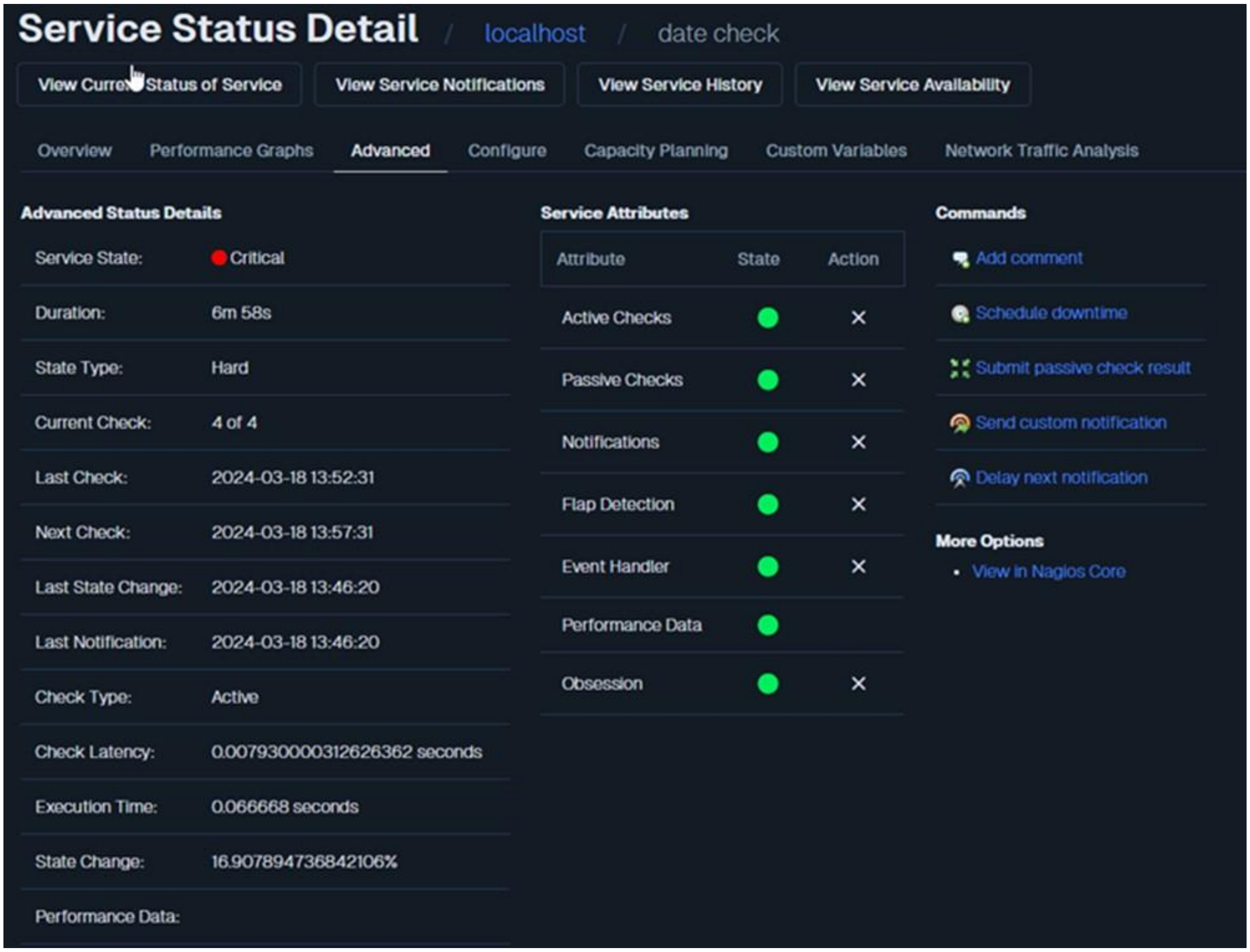
Acknowledgements and Comments

No comments or acknowledgements.

In this screenshot, you can see that the service has the same output that the test in a terminal session produced, so we know the service is correctly working. This means that the new plugin that was added to Nagios XI is working as expected.

Performance Data

If you expect that your plugin will generate performance graphs, you will want to verify that it is collecting performance data. To do so, click the **Advanced** tab after your plugin has run a check and make sure you see some output under **Performance Data**. If you don't see anything there, you will not be able to see performance graphs with this plugin.



Advanced Status Details

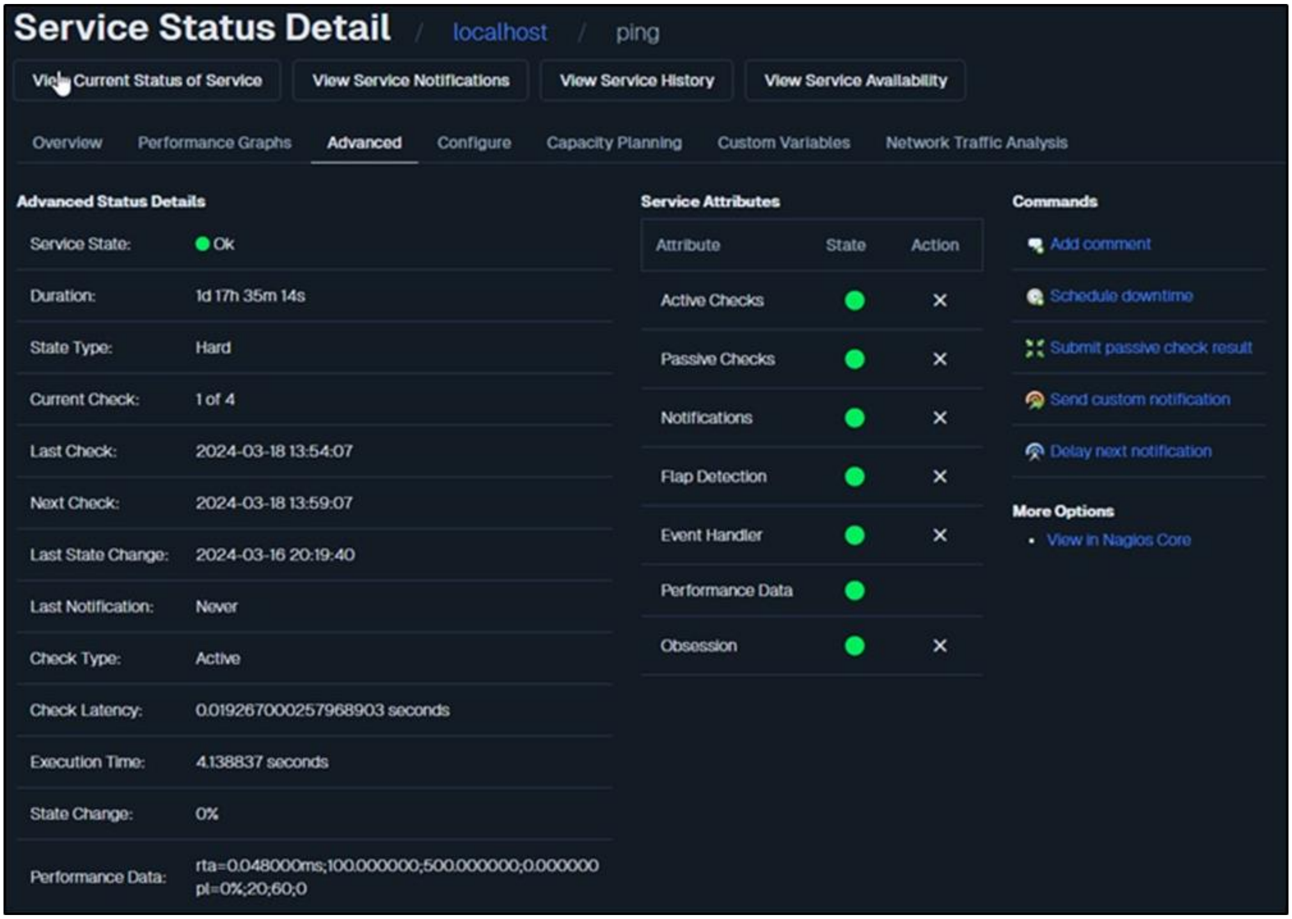
Service State:	● Critical
Duration:	6m 58s
State Type:	Hard
Current Check:	4 of 4
Last Check:	2024-03-18 13:52:31
Next Check:	2024-03-18 13:57:31
Last State Change:	2024-03-18 13:46:20
Last Notification:	2024-03-18 13:46:20
Check Type:	Active
Check Latency:	0.007930000312626362 seconds
Execution Time:	0.066668 seconds
State Change:	16.907894736842106%
Performance Data:	

Service Attributes

Attribute	State	Action
Active Checks	●	✕
Passive Checks	●	✕
Notifications	●	✕
Flap Detection	●	✕
Event Handler	●	✕
Performance Data	●	
Obsession	●	✕

The `countdown_to_date.php` plugin used in this example does not output performance.

After about 15 minutes (when enough data has been gathered) performance graphs will appear on the **Performance Graphs** tab.



Service Status Detail / localhost / ping

Advanced Status Details

Service State:	● Ok
Duration:	1d 17h 35m 14s
State Type:	Hard
Current Check:	1 of 4
Last Check:	2024-03-18 13:54:07
Next Check:	2024-03-18 13:59:07
Last State Change:	2024-03-16 20:19:40
Last Notification:	Never
Check Type:	Active
Check Latency:	0.019267000257968903 seconds
Execution Time:	4.138837 seconds
State Change:	0%

Performance Data:	rta=0.048000ms;100.000000;500.000000;0.000000
	pl=0%;20;60;0

Service Attributes

Attribute	State	Action
Active Checks	●	✕
Passive Checks	●	✕
Notifications	●	✕
Flap Detection	●	✕
Event Handler	●	✕
Performance Data	●	
Obsession	●	✕

Commands

More Options

The **PING** service is an example of performance data.

After about 15 minutes (when enough data has been gathered) performance graphs will appear on the **Performance Graphs** tab.

Finishing Up

This completes the documentation on managing plugins 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