How To Monitor Docker Containers With Nagios XI 5

Purpose

This document describes how to use the Docker Configuration Wizard to monitor your containers' status and resource usage with Nagios XI. If you are using the more recent Nagios XI 2024, refer to the updated document.

Target Audience

This document is intended for use by Nagios Administrators and end users who want to monitor their Docker containers. Note that at the moment, we do not support Docker for Windows or Docker for Mac. These may be added in a future update.

Overview

The Docker configuration wizard allows two methods for monitoring Docker. It is highly recommended to make use of Docker's Remote API, if this is not possible then a plugin can be executed on the Docker server using the Nagios Cross-Platform Agent (NCPA). Either method requires some prerequisite steps to be followed first which are outlined below.

Using Docker Remote API

If possible, it is highly recommended to make use of Docker's built-in cURL API by binding the docker socket to a TCP port. At the time of this writing, this is most easily done by adding an additional host to the docker startup command. You can test the connection to the TCP port by executing the following command from your Nagios XI server inside a terminal session (replace `ip` and `port` with the relevant values for your docker server):

```
curl -f -g http://ip:port/containers/json?all=true
```

You can also test this by clicking the **Populate Containers/Networks** button on the first page of the Docker configuration wizard, after entering the relevant information. If the list successfully populates, or the command above returns a JSON object **other than** `{"message": "page not found"},` please proceed to the Docker Configuration Wizard section of this document.

Using NCPA

If you're not able to bind the Docker daemon to a TCP port, you will need to install NCPA on your Docker machine. The NCPA download link is available from the Docker Configuration wizard or in the Installing NCPA documentation.

Once installed you will need to download the check_docker.py plugin to the NCPA's plugins folder. The plugin can be downloaded directly from the Nagios XI server, in the following commands replace `xi_address` with the IP address of your Nagios XI server. In a terminal session on the Docker server execute the following commands:

```
cd /usr/local/ncpa/plugins/
wget http://xi_address/nagiosxi/includes/configwizards/docker/plugins/check_docker.py
```

You will also need to add the nagios user to the docker group, this will enable the nagios user to read/write to the docker socket, which is necessary for the `check_docker.py` plugin to function. In the same terminal session execute the following command:

```
usermod -a -G docker nagios
```

You will need to modify the NCPA configuration file and the sudoers file as well.

Edit the ncpa.cfg file with vi or your preferred editor. The default location for the ncpa.cfg file is:

```
/usr/local/ncpa/etc/ncpa.cfg
```

Edit the ncpa.cfg file, find the line that starts with **run_with_sudo**. Turn on sudo for the check_docker.py plugin, by adding the name of the plugin (check_docker.py) to this configuration key: **run_with_sudo**.

```
run_with_sudo = check_docker.py
```

Edit the **sudoers** file using **visudo**, or the appropriate tool for your operating system.

```
visudo
```

Add the following to the end of the sudoers file. This allows NCPA to run the Docker plugin with sudo and without a password prompt.

```
nagios ALL = NOPASSWD:/usr/bin/python3 /usr/local/ncpa/plugins/check_docker.py *
```

On the Docker server, change the ownership of the plugin.

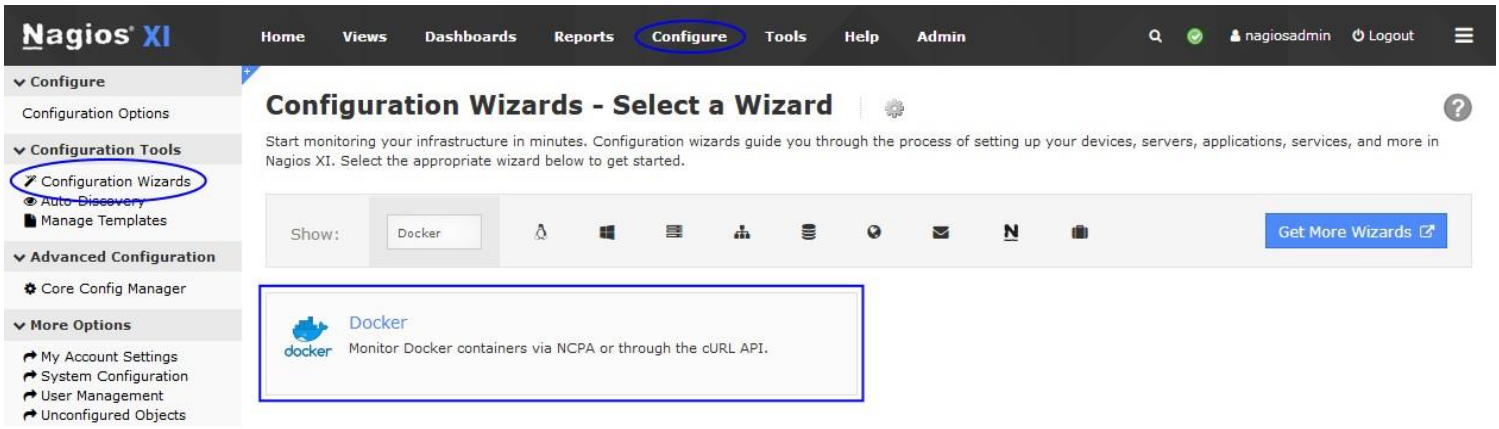
```
chown nagios:nagios /usr/local/ncpa/plugins/check_docker.py
```

You will then need to restart the machine for the group changes to take affect.

Please proceed to the Docker Configuration Wizard section of this document

Docker Configuration Wizard

The Docker Configuration Wizard communicates with your Docker installation through the Docker UNIX socket. Each check will retrieve the relevant metrics from your Docker installation and compare them to thresholds you set in the wizard. To begin using the wizard, navigate via the top bar to **Configure > Configuration Wizards** and select the **Docker** wizard. In the following screenshot you can see how the search field allows you to quickly find a wizard.



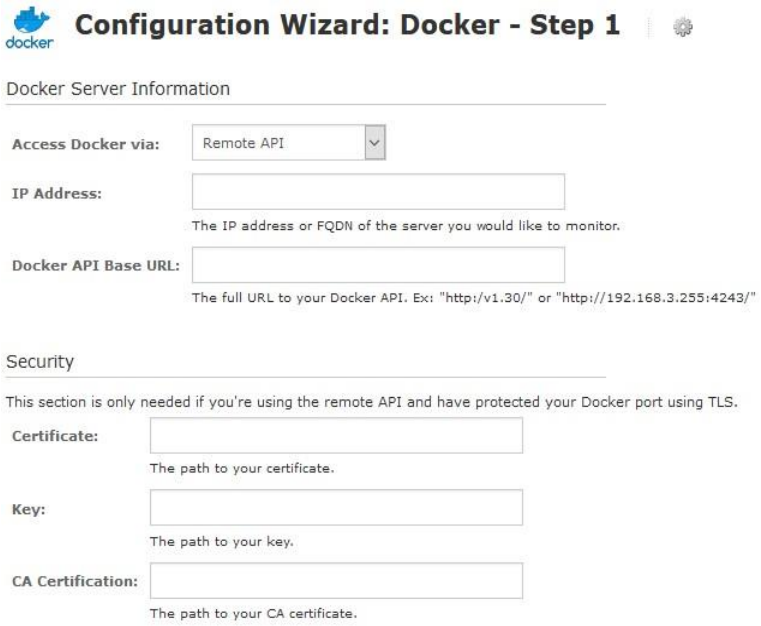
Step 1 is split up into two sections, **Docker Server Information** and **Checks to Run**. The **Docker Server Information** section has different options depending on how you are accessing Docker.

Remote Agent (NCPA)

- **IP Address** is the IP address of the machine which is running Docker.
- **NCPA Listener Port** is the port that NCPA is configured to listen to.
- **NCPA Token** is the Token that allows access to NCPA.
- **Docker Socke**t is the location of the Docker socket, normally `/var/run/docker.sock`
- **Docker API Base URL** is the URL to access Docker, this will normally be closely related to your API version, i.e. `http:/v1.30/` for an installation running API version 1.30

Remote API

- **IP Address** is the IP address of the machine which is running Docker
- **Docker API Base URL** is the URL to access your Docker API, i.e. `http://ip:port/`

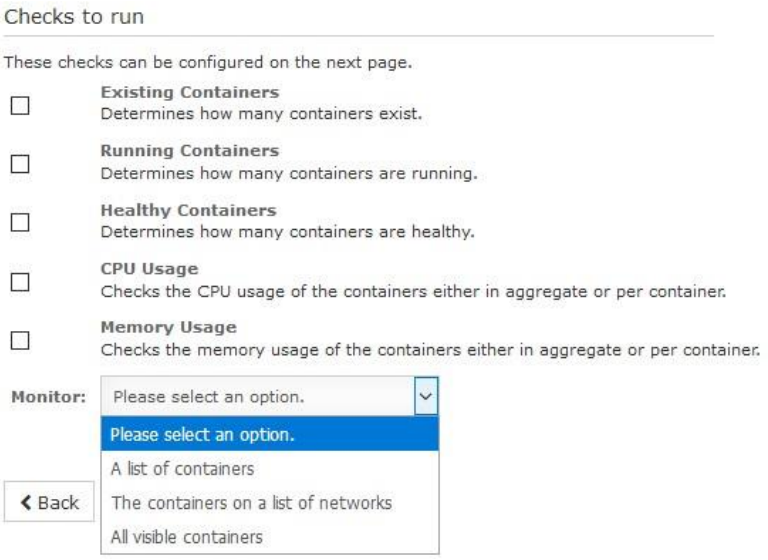


Security

- The security section will be shown when you have selected the **Remote API** access method, these are only required if you have configured Docker with TLS for additional security. The three options available need to be populated with the locations of the relevant files on your Nagios XI server.

Checks to Run

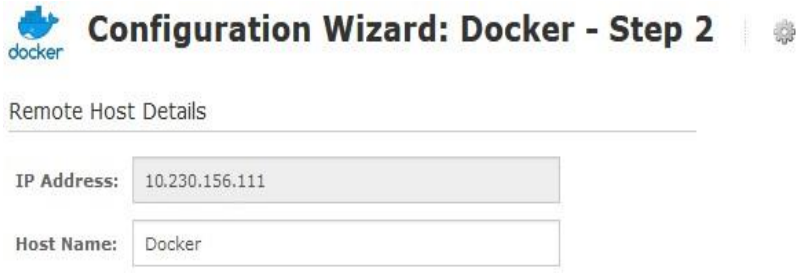
- This section provides a list of monitoring options that you will need to select before proceeding to Step 2.
- The options **A list of containers** and **The containers on a list of networks** both display the **Populate Container/Network List** button. Clicking the button will provide a list of containers that will be used in Step 2 of the wizard.



After making all your selections click **Next** to proceed to **Step 2**.

The choices presented to you in **Step 2** will depend on the checks you selected in **Step 1**.

In **Remote Host Details** you have the choice of defining the **Host Name** to your requirements. All the services created by this wizard will be assigned to this newly created host.

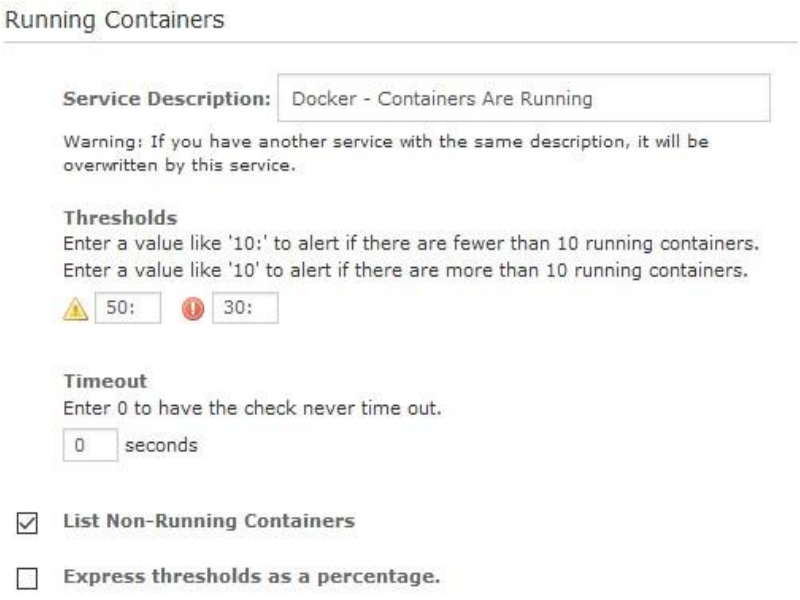


Existing Containers (if the section is present)

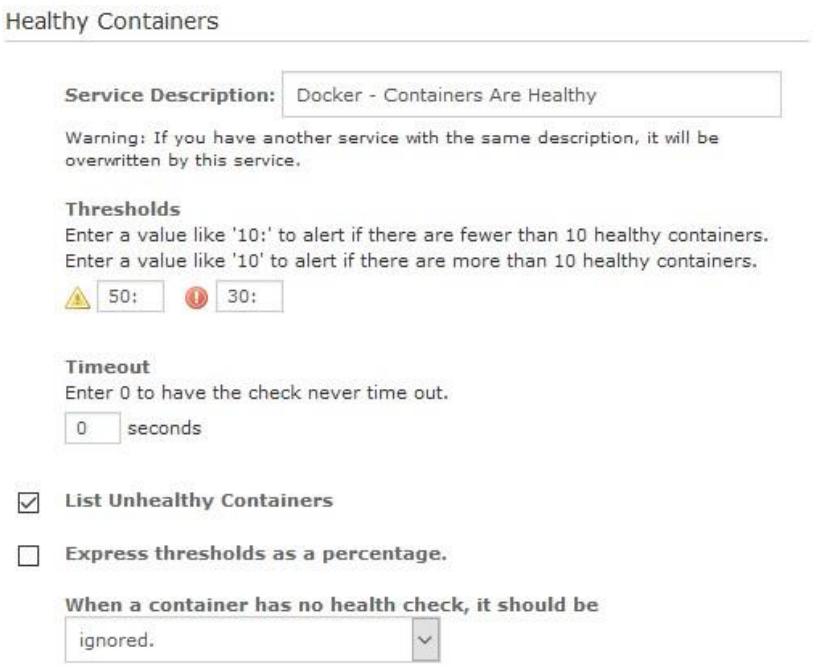
- **Service Description** is the name you will see associated with this check
- **Thresholds** are the normal nagios thresholds
- **Timeout** will tell the check how long it has to complete before returning **UNKNOWN**

Running Containers (if the section is present)

- **Service Description** is the name you will see associated with this check
- **Thresholds** are the normal nagios thresholds
- **Timeout** will tell the check how long it has to complete before returning **UNKNOWN**
- **List Non-Running Containers** will tell the check to give you a list of containers that aren't running in the service output
- **Express thresholds as a percentage** will tell the check to treat your entered thresholds as a percentage, and to output the percent of containers that are running out of those selected, rather than a count.

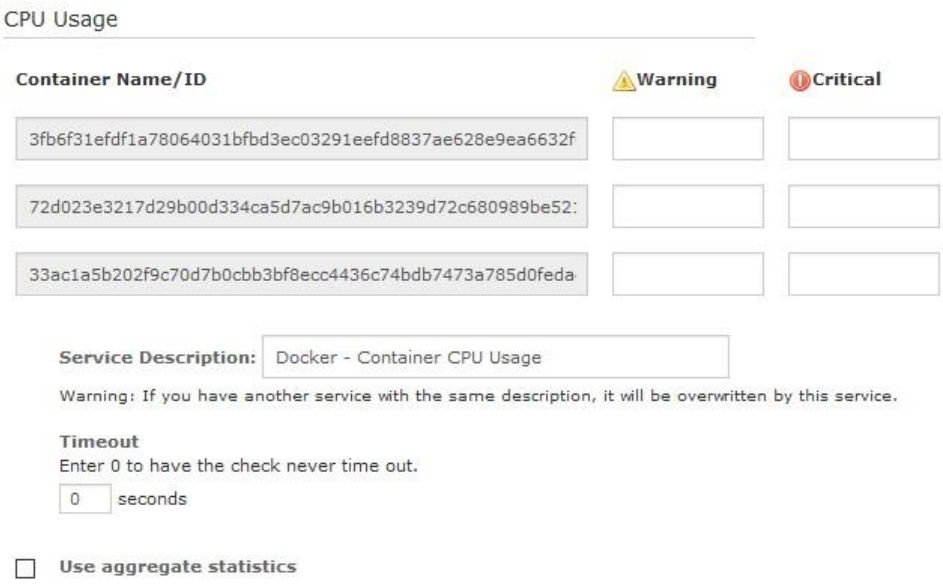


Healthy Containers (if the section is present)



- **Service Description** is the name you will see associated with this check.
- **Thresholds** are the normal nagios thresholds
- **Timeout** will tell the check how long it has to complete before returning **UNKNOWN**.
- **List Unhealthy Containers** will tell the check to give you a list of containers that aren't healthy in the service output. • **When a container has no health check...** will tell the check how to treat containers that have no **healthcheck** specified. It will default to excluding them from the total count, but if you prefer, you can have these automatically counted as healthy or unhealthy.

CPU Usage (if the section is present)

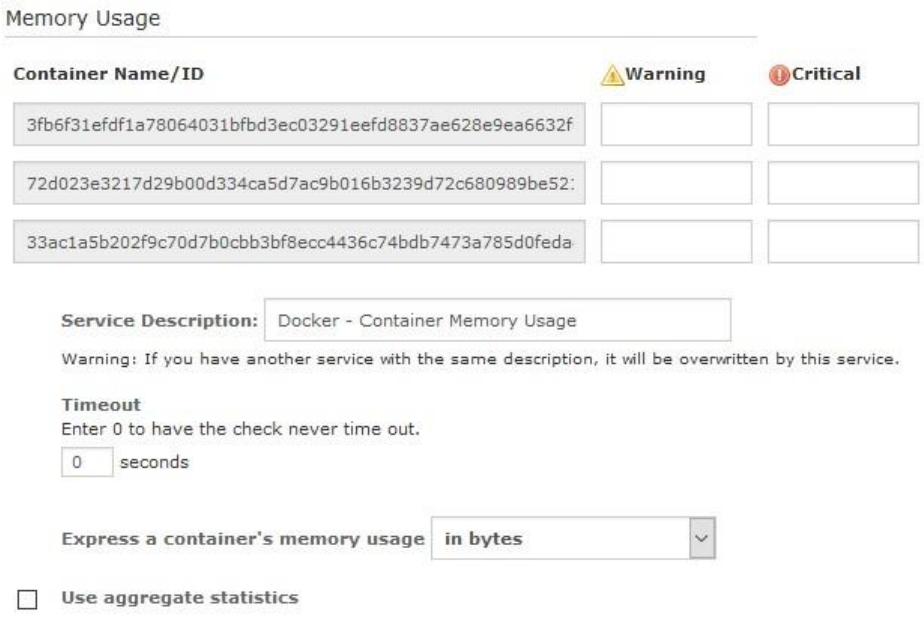


- There may be a table that shows up before the service description. If this is present, use these to specify individual (per-network or per-container) thresholds in the regular nagios format.
- A container's CPU Usage will always be collected as a percent of its host system's CPU Usage.
- **Service Description** is the name you will see associated with this check.
- **Timeout** will tell the check how long it has to complete before returning **UNKNOWN**.

- **List Containers that are outside of acceptable ranges** will tell the check to give you a list of containers that fail the check in the service output.
- **Use aggregate statistics** will allow you to set additional thresholds based on total and average CPU usage across all selected containers or networks. It will also allow you to discard the individual warning/critical thresholds if you choose.

For **Memory Usage** (if the section is present)

- There may be a table that shows up before the service description. If this is present, use these to specify individual (per-network or per-container) thresholds in the regular *nagios* format.
- A container's Memory Usage is considered to be equivalent to its resident set size.



- **Service Description** is the name you will see associated with this check.
- **Timeout** will tell the check how long it has to complete before returning **UNKNOWN**.
- **Express a container's memory usage** will let you determine whether the check should compare memory usage to a set quantity (in bytes), or to a percentage of its limit.
- **List Containers that are outside of acceptable ranges** will tell the check to give you a list of containers that fail the check in the service output.
- **Use aggregate statistics** will allow you to set additional thresholds based on total and average memory usage across all selected containers or networks. It will also allow you to discard the individual warning/critical thresholds if you choose.

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. Once the wizard applies the configuration, click the **View status details for *<your host>*** link to see the new services that have been created.

Host	Service	Status	Duration	Attempt	Last Check	Status Information
Docker	Docker - Container CPU Usage	Ok	N/A	1/5	2018-06-13 13:39:45	OK: total_usage returned OK (value 0.0%),
	Docker - Container Memory Usage	Critical	1m 43s	1/5	2018-06-13 13:39:45	CRITICAL: total_usage returned CRITICAL (value 688128.0B),
	Docker - Containers Are Healthy	Ok	N/A	1/5	2018-06-13 13:39:45	OK: 3 containers monitored, 3 have no health check, container IDs with no healthcheck: ['3fb6f31efdf1a78064031bfbd3ec03291eefd8837ae628e9ea6632f0c8273a79', '72d023e3217d29b00d334ca5d7ac9b016b3239d72c680989be52151102ddc6a3', '33ac1a5b202f9c70d7b0cbb3bf8ecc
	Docker - Containers Are Running	Ok	N/A	1/5	2018-06-13 13:39:45	OK: 1 running, 2 not running, IDs not running: ['72d023e3217d29b00d334ca5d7ac9b016b3239d72c680989be52151102ddc6a3', '33ac1a5b202f9c70d7b0cbb3bf8ecc4436c74bdb7473a785d0feda45fa83d66f']
	Docker - Containers Exist	Ok	N/A	1/5	2018-06-13 13:39:45	OK: 3 containers found

Finishing Up

This completes the documentation on **How To Monitor Docker Containers With 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