Purpose

This document describes how to configure inputs in Nagios Log Server.

What Are Inputs?

Inputs allow Nagios Log Server to collect data from various places, like TCP/UDP ports, SNMP Traps, Unix sockets, long running command pipes, etc.

Input Configuration Location

Nagios Log Server is a cluster oriented application that uses Logstash to receive and process logs. The base configuration provided with a default installation of Nagios Log Server has all the inputs defined as part of the Global Config. Global Config is an easy way to set up the same Logstash configuration on all your instances. To access the configuration, navigate to **Configure > Global (All Instances) > Global Config.**

Nagios®

Input Configuration Options

On the **Global Config** page there are two main tables named **Inputs** and **Filters** (and a third table called **Outputs** which is hidden). **Inputs**, **Filters** and **Outputs** are all used by Logstash to process incoming log data and do something with it, which normally is to store it in the Elasticsearch database. This document will focus on **Inputs**.

In the **Inputs** table there are several pre-configured inputs that come as part of Nagios Log Sever, these are called **blocks**. The blocks have several icons which are explained as follows.

Active Active/ **Inactive** Inactive

- o Each input block is Active, as indicated by the green **Active** box next to it. Clicking this box marks the input as Inactive, as indicated by the gray **Inactive** box. Inactive inputs will not be included in the running configuration the next time it is applied, allowing you to save inputs for future use without immediately applying them. Clicking the box again toggles the input back to Active.

Rename

- o This allows you to rename the block.

Open / Close

- o Clicking the **Open** icon will expand the block to show the input configuration in a text field. Once expanded, the **Open** icon changes to the **Closed** icon, which can be clicked to collapse the block. For more details, see the Input Structure section of this document.

Copy

- o This will create a duplicate of that block allowing you to easily create a new input based off the configuration of an existing input.

Remove

- o This will delete a block when it is no longer required.

Any changes you make will not be saved until you click the **Save** button. If you navigate away from the page before saving, all of your changes will be lost.
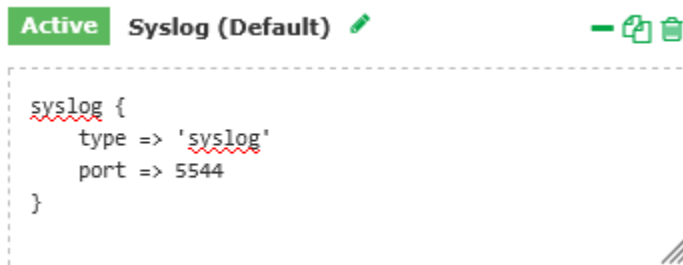
Input Structure

The inputs are a structured format like this:

```
<plugin>  {
    <config_option>  =>  <config_value>
    <config_option>  =>  <config_value>
}
```

`<plugin>` is the name of plugin that you wish to enable. Each plugin can have multiple `<config_option>` defined, depending on your requirements. Logstash includes a wide variety of plugin types, here is an example:

The example is using the `syslog` plugin that is configured by default in Nagios Log Server.

The purpose of showing you this example is to demonstrate how an input requires log data to come from "somewhere". In the example, the log data is coming via the network on port `5544` (TCP or UDP).

In the example you can see the option `type` has the value of `'syslog'`. This will label any logs coming in as `syslog` so you can easily manage them through the dashboard and queries. It also provides the ability for filters and outputs to work with this log data based on the value of the `type` field.

Another example is the `tcp` plugin configured for receiving Windows Event Logs. This is configured by default in Nagios Log Server:

The first option is `type` and this has the value of `'eventlog'`. This will label any logs coming in as `eventlog` so you can easily manage them through the dashboard and queries.

The next option is `port`, and this has the value of `3515`. This enables the Logstash listener to receive TCP network traffic on port `3515` (because this is the `tcp` plugin, it does not listen for UDP traffic).

The next option is `codec`, and this has the value of `json`. The `json` option has additional configuration options that can be defined.

The purpose of this example was to demonstrate that inputs can have a sub-set of configuration settings available, it allows for very flexible configurations.

Adding An Input

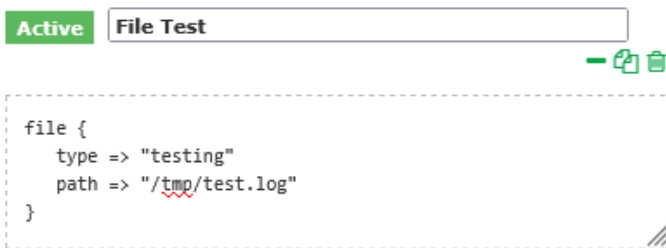1. On the **Global Config** page, click the **Add Input** drop-down list and select **Custom**.
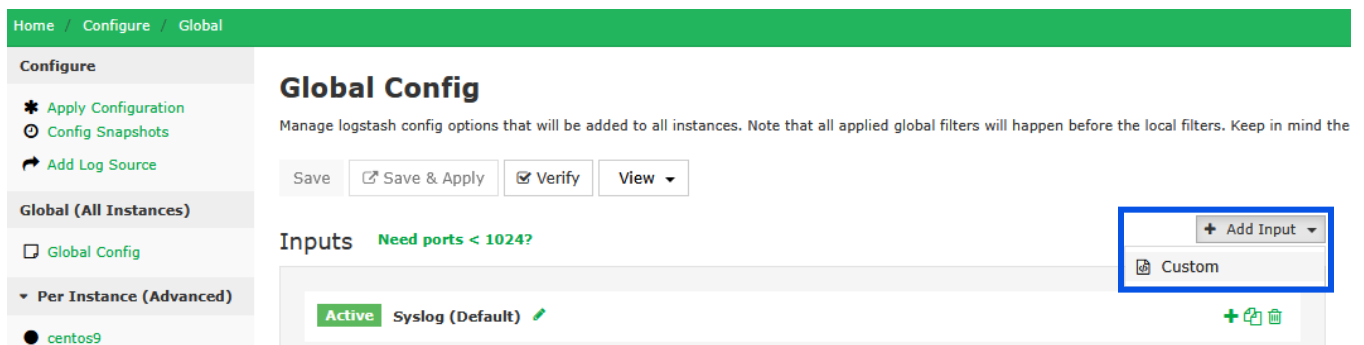
2. A new block appears at the bottom of the **Inputs** table.
   o Type a descriptive name for the input.
   o In the text field you will need to define the input configuration. Here is a basic example for a local file on the Nagios Log Server machine itself:

```
file {
type -> "testing"
path => "/tmp/test.log"
}
```



3. Once you have finished, click the **Save** button.

For the new input to become active you will need to Apply Configuration, however, it is recommended that you Verify the configuration first. The next step will be to Apply Configuration to put this new input into the running configuration so it can be tested and demonstrated.

## Apply Configuration

To apply the configuration, navigate to **Configure > Configure > Apply Configuration**. Click the **Apply** button. A popup will appear, click the **Yes, Apply Now** button.
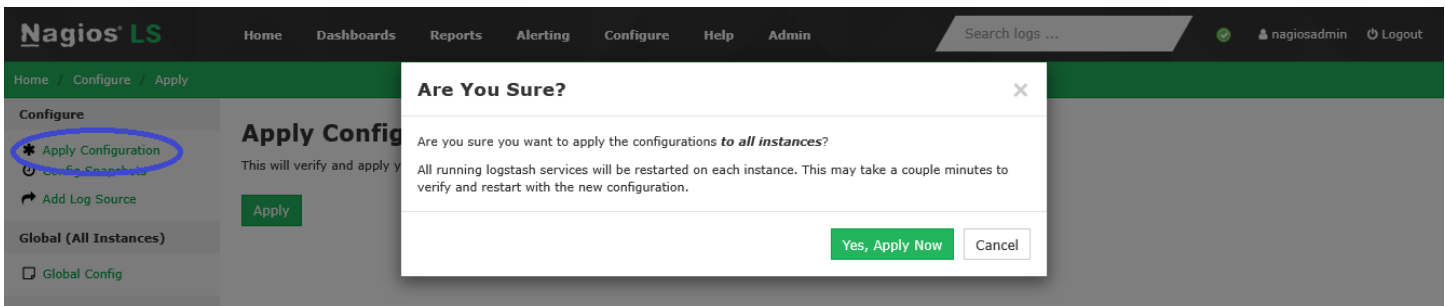


Wait while the configuration is applied. When it is completed, the screen will refresh with a completed message. Please proceed to the Test Input section below.

**Nagios**®

## Test Input

1. Using the input example created previously, these steps will show you how to confirm the input is working. Establish a terminal session to your Nagios Log Server instance and then execute the following command:

```
echo "This is a test log entry" >> /tmp/test.log
```

2. In Nagios Log Server, open the **Dashboards** page**,** click the **Advanced Search** drop-down link in the **Query** section, and perform the query `type:testing`. The query should return one result in the **ALL EVENTS** panel.
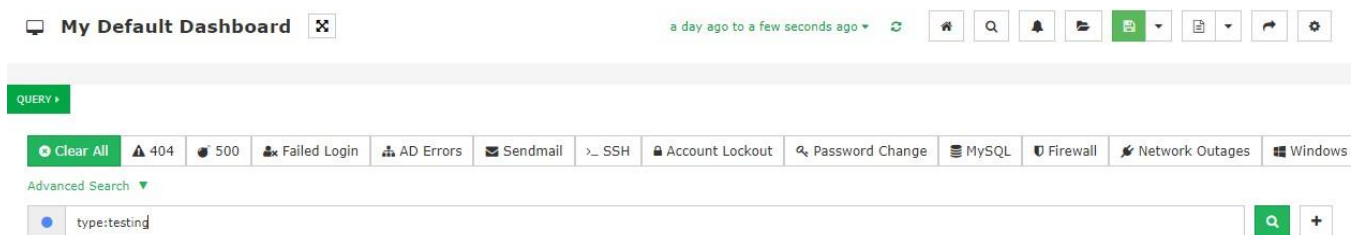
3. Clicking on the log entry will show you the full details about the entry. Here you can see that the `type` is `testing`, and the text has been stored in the `message` field.

Global Config Buttons

This section explains the buttons available on the **Global Config** page. Navigate to **Configure > Global (All Instances) > Global Config**.

Verify

The **Verify** button ensures that the current saved configuration is valid. It can be useful when updating your configurations before attempting to **Apply Configuration**.
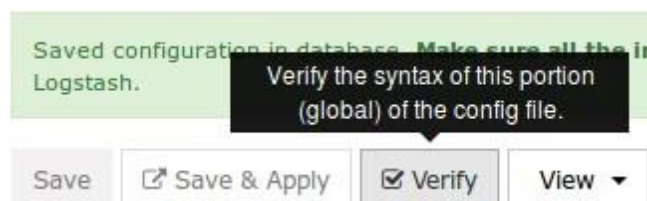
Wait while the configuration is verified.

If you do not receive a **Configuration is OK** message, then you will need to fix the problem before applying the configuration.

Save vs Save & Apply

There are two separate buttons, **Save** and **Save & Apply**.

Save allows you to make changes but not make the changes become immediately active. This allows you to navigate away from the **Configure** screen without losing your work.

Save & Apply will save your changes and then make the changes become immediately active. This can be helpful if you changed a simple option in your config that does not need to be verified.

View

The **View** button allows you to view the raw Logstash configuration. When you click the button, you have a choice of selecting an individual config or a combination of all the configs.
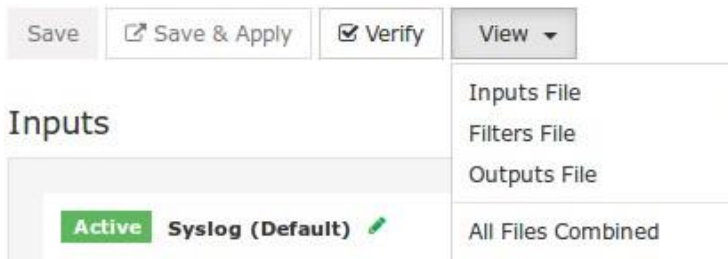
This will open a new popup window where the configuration can be viewed or copied.

Firewall Ports

If you add an input to Nagios Log Server that uses a network port to receive the data, then the local operating system firewall will also require a rule to be added to open that network port. A detailed explanation of this can be found in the Sending Syslog with SSL in NLS 2024 documentation. In that documentation, the TCP port 7778 needs to be opened, which can be done with the following commands:

RHEL | CentOS | Oracle Linux

```
firewall-cmd --zone=public --add-port=7778/tcp
firewall-cmd --zone=public --add-port=7778/tcp --permanent
```

Debian

The local firewall is not enabled on Debian by default and no steps are required here. If it is enabled then the commands are:

```
iptables -I INPUT -p tcp --destination-port 7778 -j ACCEPT
```

Ubuntu

The local firewall is not enabled on Ubuntu by default and no steps are required here. If it is enabled then the commands are:

```
sudo ufw allow 7778/tcp
sudo ufw reload
```

If you plan on creating the Input as part of the **Global Config**, you will need to create this firewall rule on all the instances in your Nagios Log Server cluster.

External Resources

The following resources provide additional information on Logstash:

Structure Of A Logstash Configuration File:

https://www.elastic.co/guide/en/logstash/2.4/configuration-file-structure.html

Input Plugins Available for Logstash:

https://www.elastic.co/guide/en/logstash/2.4/input-plugins.html

Finishing Up

This completes the documentation on configuring inputs in Nagios Log Server. 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

Nagios®