How To Integrate Slack With Nagios Network Analyzer 2026

Purpose

This document describes how to integrate Slack with your Nagios Network Analyzer instance.

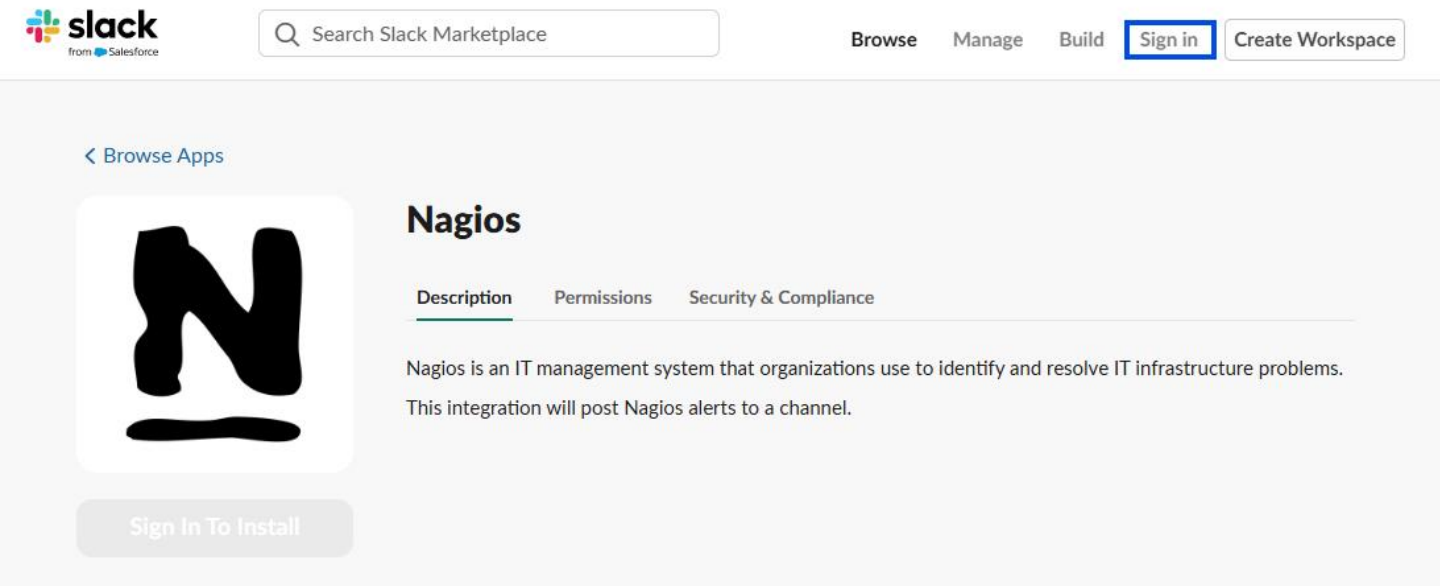
Overview

In Slack you have Channels that receive notifications. The Slack API allows you to target these channels by using the channel name, like NNA. This is used to send notifications from Nagios Network Analyzer to Slack. This documentation will create a Nagios Network Analyzer Command that will be used by Checks to send alerts to a Slack Room called NNA.

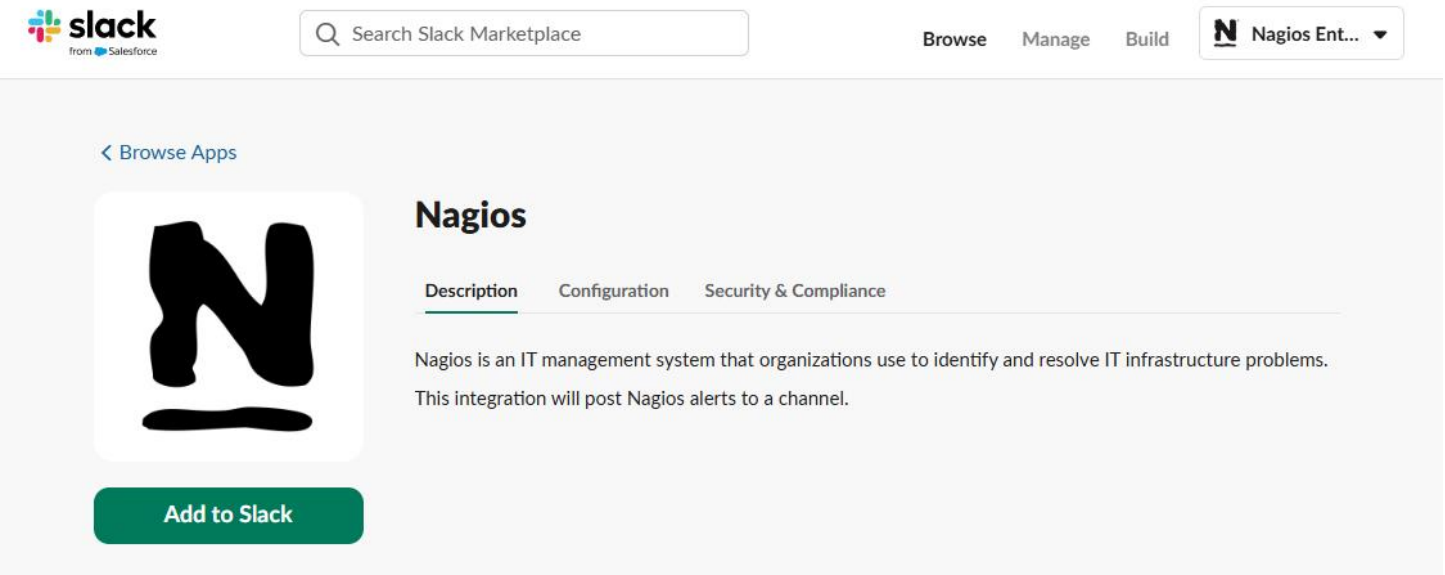
Install Nagios App In Slack

The first step is to install the Nagios App into Slack. Open your web browser to https://slack.com/apps/A0F81R747-nagios.

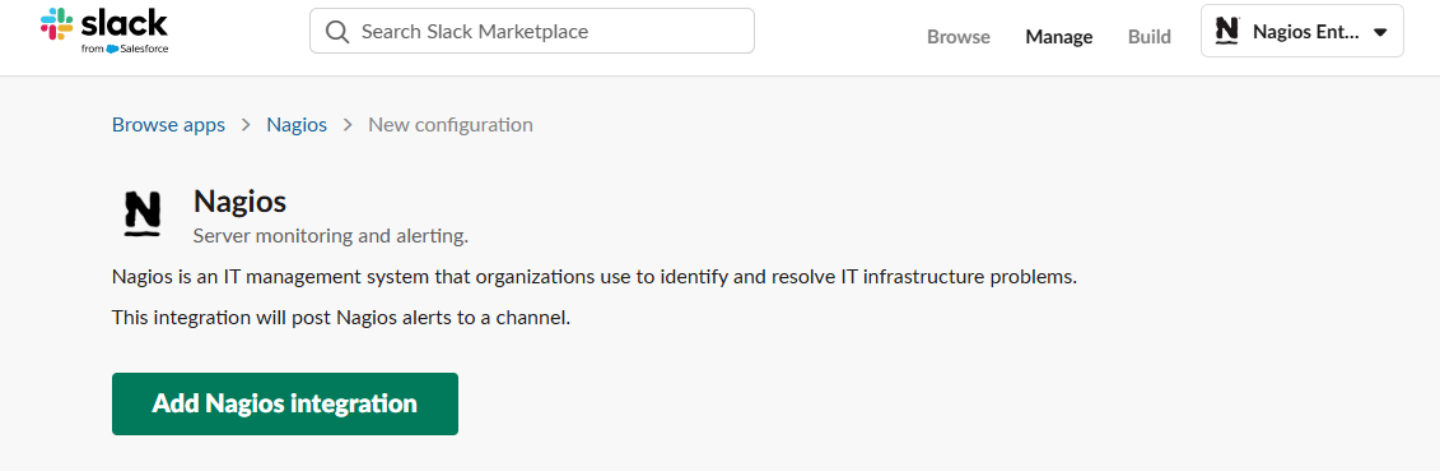
1. If you are not signed in to Slack, you will see a page similar to the following. Click **Sign in** and follow instructions to sign in to Slack.



2. Once you are signed in, you will see a page similar to the following. Click **Add to Slack**.

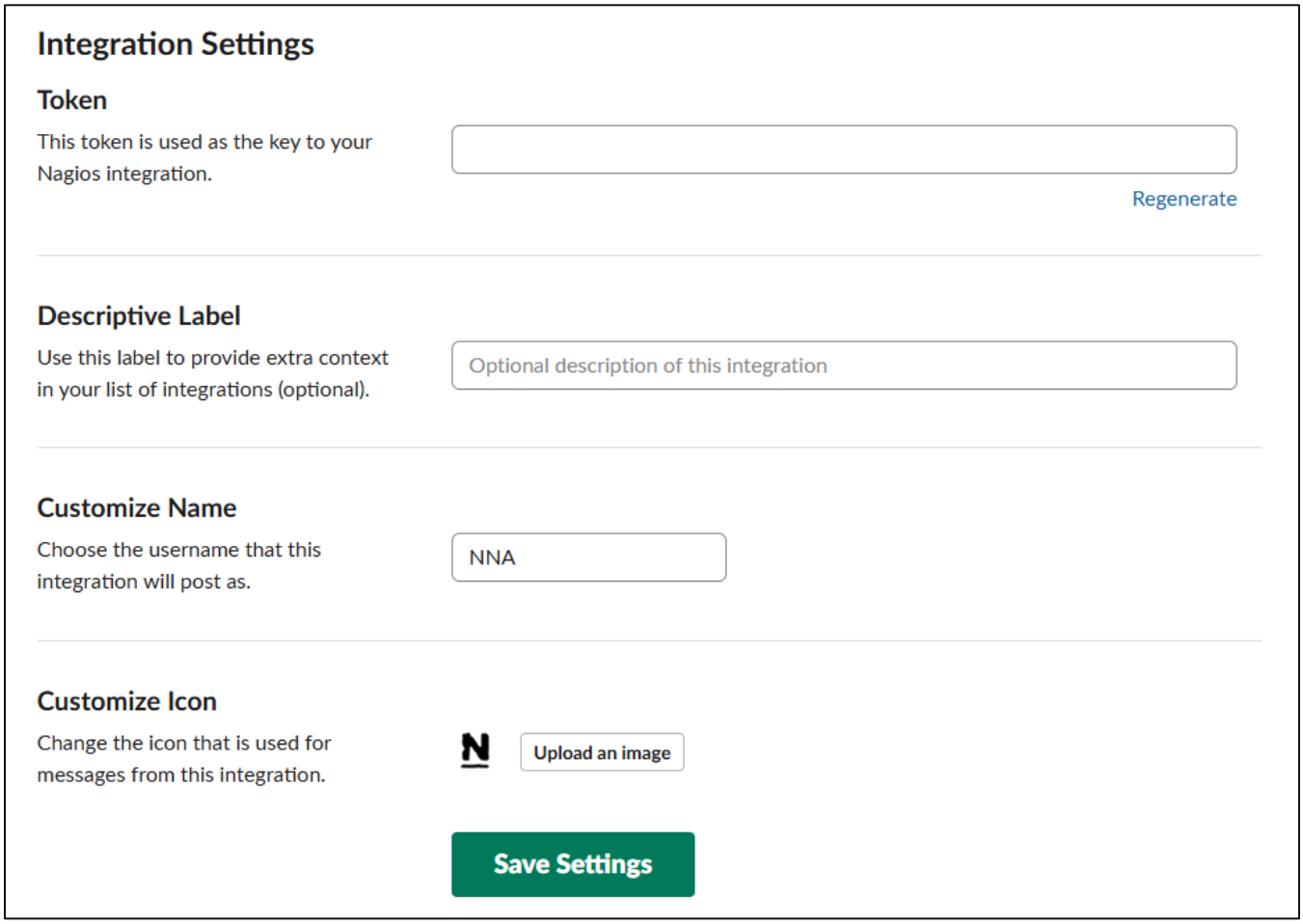


3. Click **Add Nagios Integration**



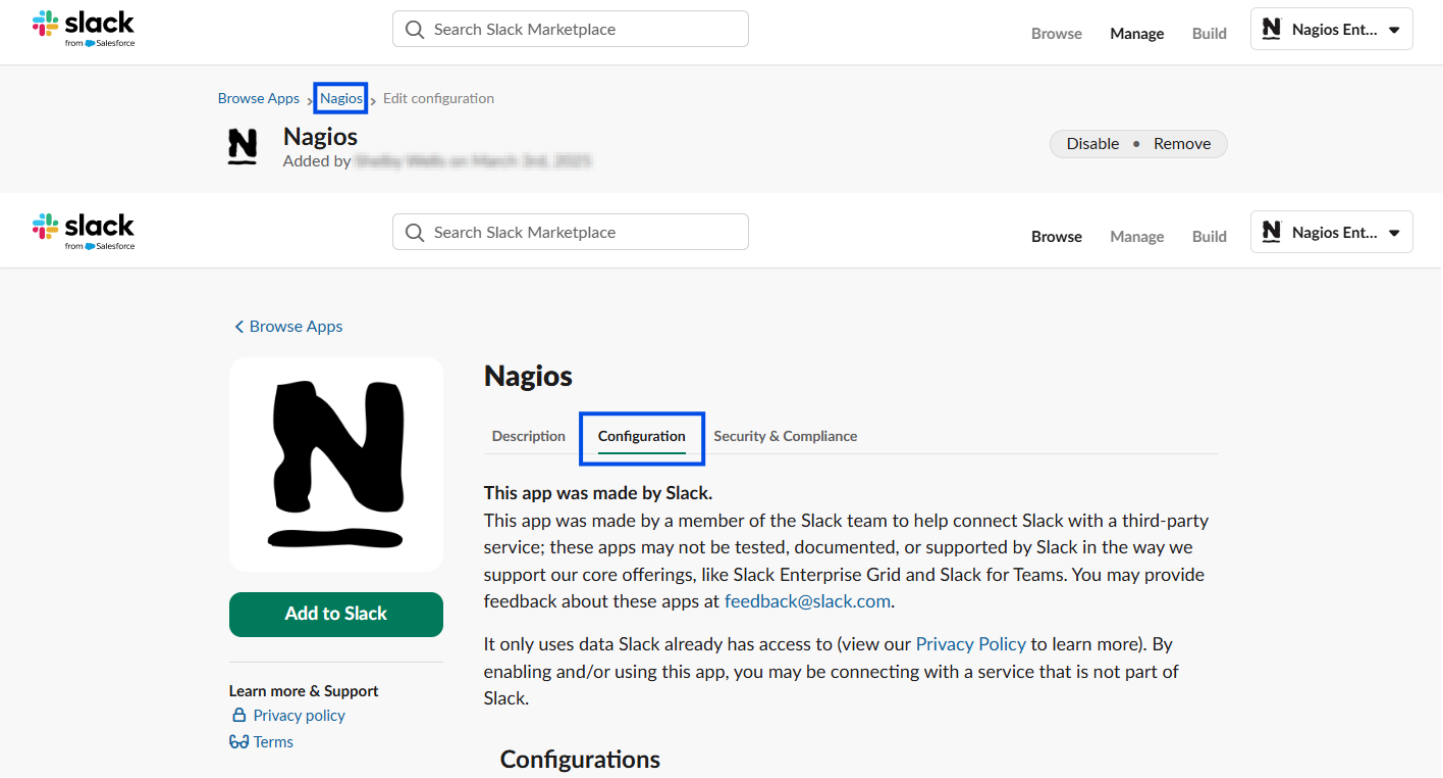
4. The next page is where you configure the app. The first section is the **Setup Instructions**, please do not follow these as this document will provide you with instructions specific to Nagios Network Analyzer.

5. Scroll down to the **Integration Settings** section. Take note of the value in the **Token** field, this will be required further on. In the **Customize Name** field you can see that **NNA** has been typed.



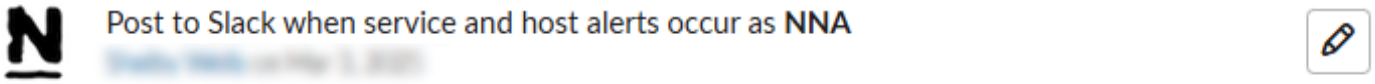
6. Click the **Save Settings** button after making the required changes.

7. Navigate to **Nagios > Configuration**



8. Once saved a new entry will appear under your **Configurations**.

   Click the pencil icon to edit the configuration or to view the **Token** again if you forgot it.

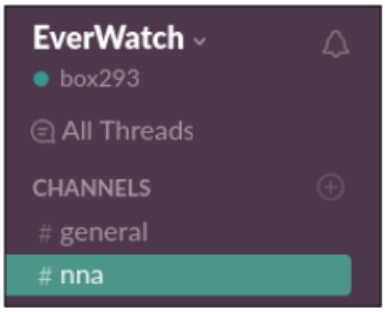


You have finished with the Slack web page; you can leave it open if you like as you may need to return here to get the **Token** if you forget it.

Create NNA Channel In Slack

1. Open the Slack application. Click the plus (**+**) icon in the left sidebar and select **Channel**.
2. On the **Create a channel** screen, **Blank channel** is the default, select **Next**.
3. On the **Channel details** screen, enter a **Channel name** (e.g., **nna**) and select the **Visibility** as either **Public** or **Private**.
4. Click **Create**.

Here you can see the new Channel now exists.



This completes the steps required in Slack. Leave the application open in case you need to return to it after configuring Nagios XI.

Configure Nagios Network Analyzer

The next step is to configure Nagios Network Analyzer. This includes:

- Installing Prerequisites
- Installing Slack Integration Script
- Create Wrapper Script
- Create Commands
- Create Check

Installing Prerequisites

Open a terminal session to your Nagios Network Analyzer server as the root user. Execute the following command to install the prerequisites:

RHEL | CentOS | Oracle Linux

```
yum install -y perl-libwww-perl perl-Crypt-SSLeay perl-LWP-Protocol-https
```

Debian | Ubuntu

```
apt-get install -y libwww-perl libcrypt-ssleay-perl liblwp-protocol-https-perl
```

Wait while they are installed. Leave this terminal session as you'll need it in the following step.

Installing Slack Integration Script

Execute the following commands to download the Slack Integration script:

```
cd /usr/local/nagiosna/scripts/
wget -O "slack_nagios.pl"
https://raw.github.com/tinyspeck/servicesexamples/master/nagios.pl
chmod 0775 slack_nagios.pl
chown nna:nnacmd slack_nagios.pl
```

NOTE: the wget command and the URL are one command and must be entered together on the same line.

The next step is to edit the script and define your Slack `domain` and `token`.

Execute the following command to open the script in vi:

```
vi slack_nagios.p1
```

When using the vi editor, to make changes press `i` on the keyboard first to enter insert mode. Press **Esc** to exit insert mode.

Find these lines:

```
my $opt_domain = "foo.slack.com"; # Your team's domain
my $opt_token = ""; # The token from your Nagios services page
```

Tip: Type `:66` and press **Enter** to go directly to these lines.

The first line needs to be your team's Slack `domain`, this was provided when you signed in to your team on the Slack web page.

The second line is the `token` that was generated when you added the Slack Integration on the Slack web page.

Make the required changes to these two lines.

When you have finished, save the changes in vi by hitting **Esc** and typing:

```
:wq
```

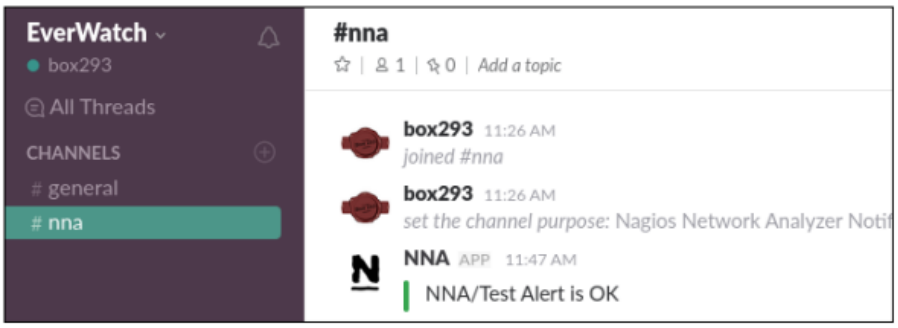
and press **Enter**.

You can test that it works by executing the following command. Note that this is a single long command, not three separate commands, so be sure to copy and paste the entire command into your terminal:

```
./slack_nagios.pl -field slack_channel="#nna" -field HOSTALIAS="NNA" -field\
 SERVICEDESC="Test Alert" -field SERVICESTATE="OK" -field SERVICEOUTPUT=\
"This is a test alert" -field NOTIFICATIONTYPE="RECOVERY"
```

You'll see a lot of output generated in the terminal window; the end result should be `ok`.

Most importantly you should see it appear in the Slack **#nna** Channel.



If this works, then you have correctly installed the Slack Integration script.

If this does not work, please review the output in the terminal session as it should provide an error explaining why.

The `slack_nagios.pl` script is written for Nagios Core; however, it will work fine with Nagios Network Analyzer. We are using service fields because the alert states generated by Nagios Network Analyzer match the service states in Nagios Core.

Create Wrapper Script

Nagios Network Analyzer has restrictions on the length of the command you can create. This means you need to create a wrapper script that will execute the `slack_nagios.pl` script. Additionally, the output generated by Nagios Network Analyzer checks looks like this:

```
flows on Firewall Public with filter `dst port 9914` is 0 | flows=0;1:;1:;0
```

The ` and | characters in this output cause issues with the wrapper script. With that in mind the wrapper script will remove these characters to allow the integration to work. Execute the following command to create a new script called `slack_nagios_wrapper.sh` using vi:

```
vi slack_nagios_wrapper.sh
```

Paste the following into the new file. Note that the `slack` line is a single long command, not three separate commands, so be sure to copy and paste the entire command into your terminal:

```
#!/bin/sh
channel=$1
sourcename="$2"
state=$3
output=$4
output="${output//\`}"
output="${output//\|}"
bin="/usr/local/nagiosna/scripts/slack_nagios.pl"

slack="${bin} -field slack_channel=\"${channel}\" -field HOSTALIAS=\"NNA\" \
-field SERVICEDESC=\"${sourcename}\" -field SERVICESTATE=\"${state}\" \
-field SERVICEOUTPUT=\"${output}\" -field NOTIFICATIONTYPE=\"${state}\""

eval $slack
```

When you have finished, save the changes in vi by hitting **Esc** and typing:

```
:wq
```

and press **Enter**.

You need to define the permissions for the script with the following commands:

```
chown nna:nnacmd slack_nagios_wrapper.sh
chmod 755 slack_nagios_wrapper.sh
```

You can now test that it works by executing the following command:

```
./slack_nagios_wrapper.sh '#nna' 'Test Alert' 'OK' 'This is a test alert'
```

You'll see a lot of output generated in the terminal window; the end result should be `ok`.

Most importantly you should see it appear in the Slack **#nna** channel, it should be identical to the previous test you performed.

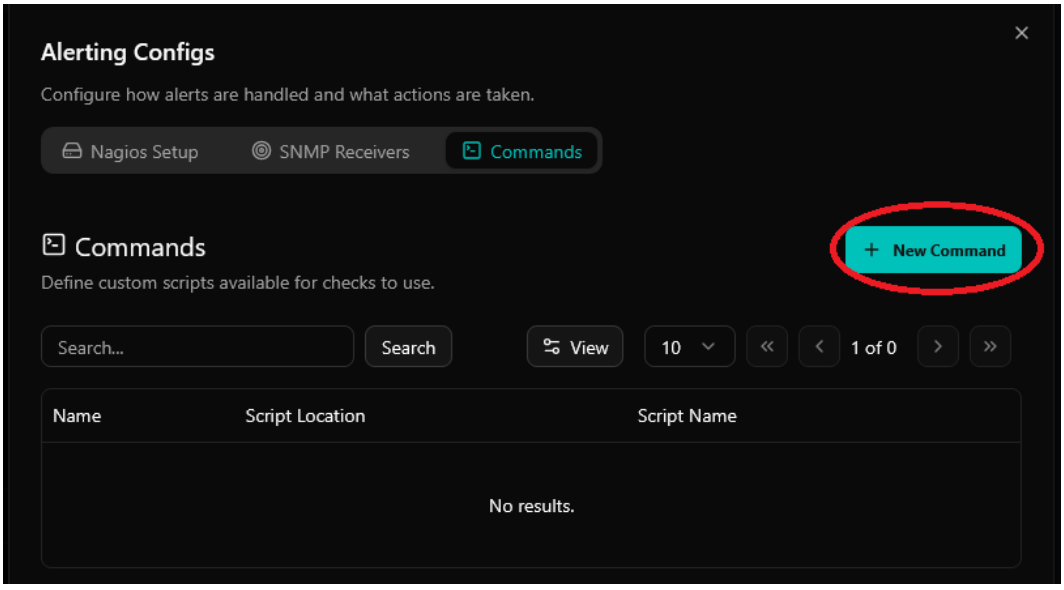
If this works, then you have correctly created the `slack_nagios.pl` wrapper script.

If this does not work, please review the wrapper script to ensure it has been typed/pasted correctly.

This completes all the steps required in the terminal session. You can now close it, as the remaining steps will be performed in Nagios Network Analyzer.

Create Commands

The commands are how the Nagios Network Analyzer checks send notifications. Open your web

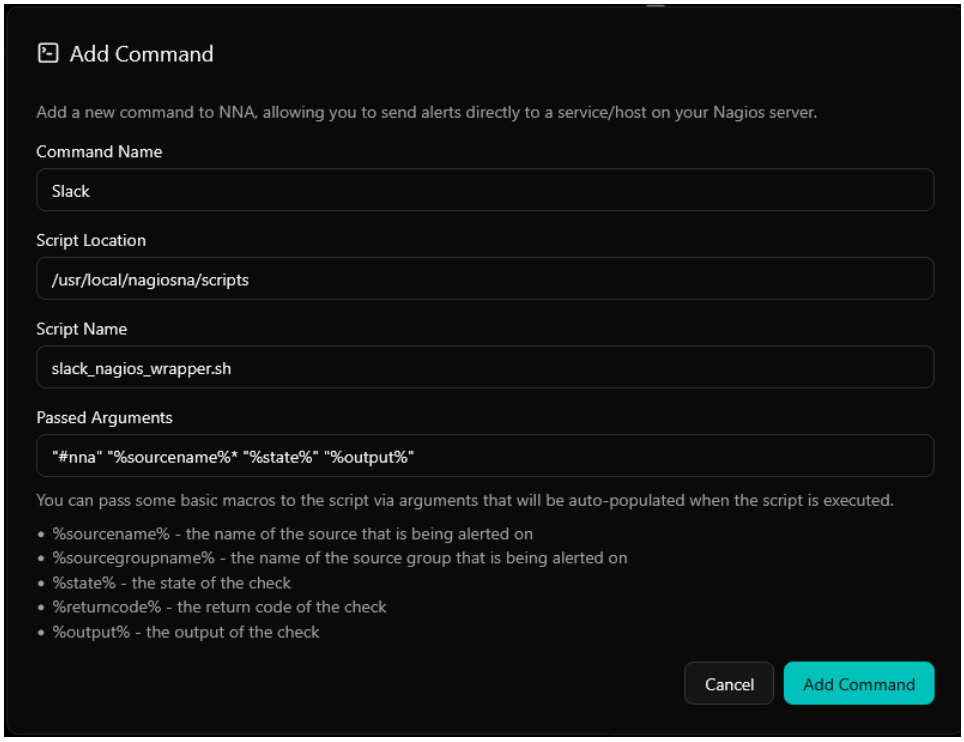


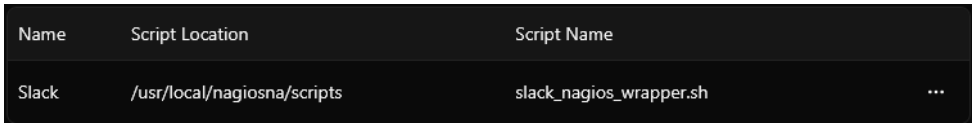
browser to Nagios Network Analyzer and navigate to **Alerting > Alerting Configs** (at the top right)

You will be presented with a list of fields to populate for creating a new command.

- **Name:** `Slack`
- **Script Location:** `/usr/local/nagiosna/scripts`
- **Script Name:** `slack_nagios_wrapper.sh`
- **Passed Arguments:** `"#nna" "%sourcename%" "%state%" "%output%"`
- Click the **Add Command** button once you have populated the fields.

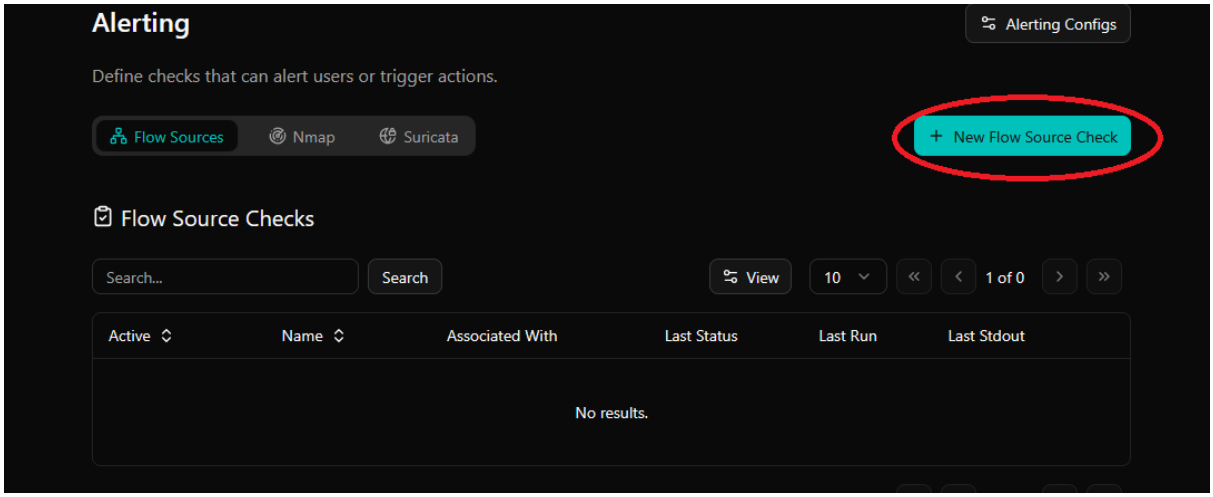
Here is a screenshot of the command that has been created:





Create Check

The last step is to define an alert so Slack receives notifications. In the following example, an alert is created to notify when a source has not received flow data on its designated port. Click the **Alerting** menu item, then click the **New Flow Source Check** button.

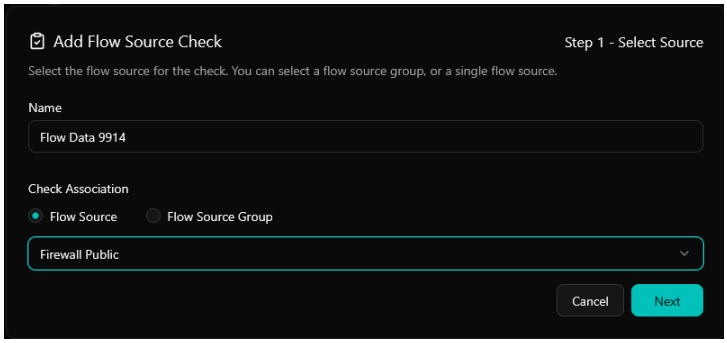


Step 1 – Select Source

Enter a name for the check for management and organizational purposes. The name can only contain whitespaces and alphanumeric characters.

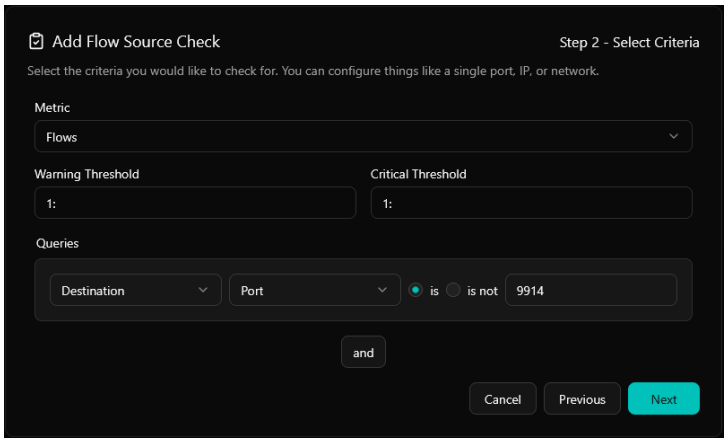
Select **Source** or **Sourcegroup**. This is the source the check will get values from

Click the **Next** button to proceed.



Step 2 – Select Criteria

Metric: Choose the metric to check against. If you want a packet count, pick **Packets**. If you want total bytes, pick **Bytes**. There are multiple traffic metrics available, and this setting specifies which one will be checked.



Warning threshold / Critical threshold

After extracting a value from the selected metric, Network Analyzer evaluates it against these thresholds to determine if the state is **WARNING, CRITICAL,** or **OK**. In this example, setting a **Critical threshold** of 1: means that if less than 1 is received, the check will enter a **CRITICAL** state, indicating that no flows were received.

More detailed information on thresholds is explained in the Nagios Threshold Values section of this document.

The bottom half of the screen is how you filter what data the check is looking at, which allows granularity.

In this example, **Flows** is the type of traffic being analyzed. The filter criteria used is:

- **Destination**: Specifies the direction of the flow traffic being examined.
- **Port**: This check is testing to make sure flow data is actually being received, seeing as the flow data is received on a port then this makes it easy to check.
- **Is:** Defines the operation; in this case, verifying that the **destination port is 9914**.
- **9914:** The specific port number being monitored.

Based on those selections, if no flow data is received it will be in a **CRITICAL** state, otherwise it will be **OK**.

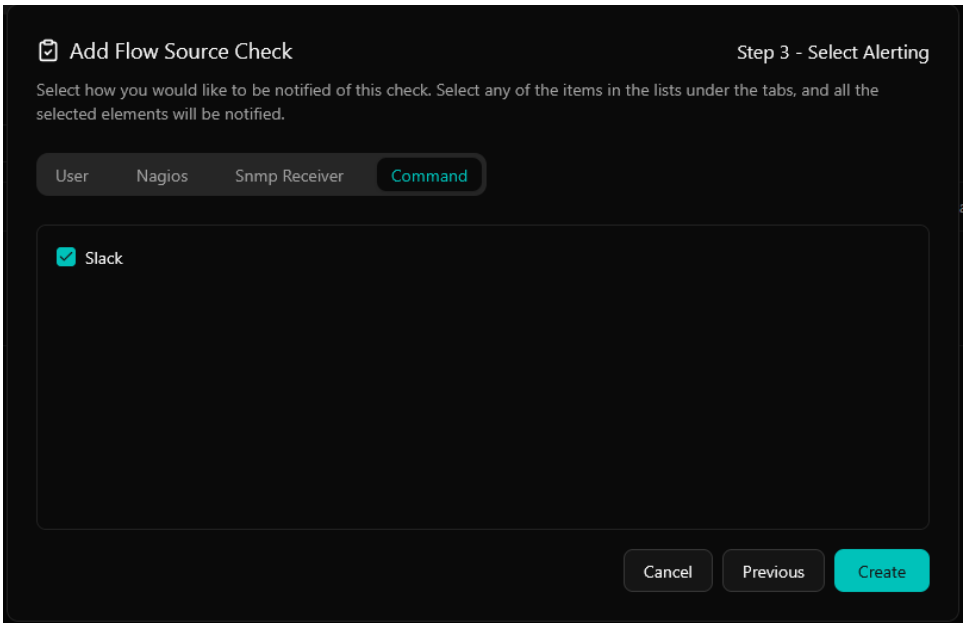
You can specify as many of these filters as you would like by clicking the **And** button. This will add a new box where you can specify additional filters. Please note that it is a Boolean AND, where the traffic must meet all specifications that are chosen for the check to be used.

Click the **Next** button to proceed

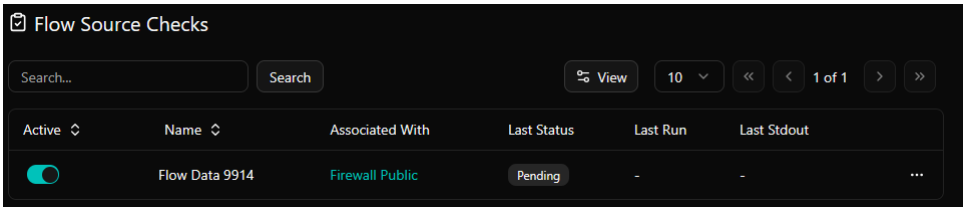
Step 3 – Select Alerting Methods

In this example, the **Commands** tab has been selected and the **Slack** command that was previously defined has been selected.

Click the **Create** button to create the alert.

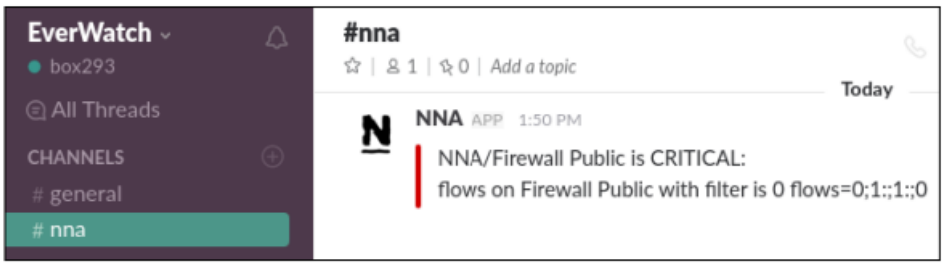


The check will be created and will appear on the screen in a pending state:



The checks run every five minutes, and it's important to understand that each time the check runs, it will send a notification to Slack. Slack will receive the check results based on the **WARNING** and **CRITICAL** thresholds you defined. The notification will include the state of the check (**OK**, **WARNING**, or **CRITICAL**).

Here is an example of the check when it had a **CRITICAL** state:



You can see how Slack has a horizontal red line for the **CRITICAL** notification.

You have now successfully configured Nagios Network Analyzer to send notifications to Slack.

Nagios Threshold Values

Nagios Thresholds can be complicated to initially understand, but once you grasp them, they can be very powerful. Documentation on Nagios thresholds is available here:

https://nagios-plugins.org/doc/guidelines.html#THRESHOLDFORMAT

The Nagios Threshold standards were designed with many different use cases, including those where negative numbers are valid values. However, in the case of Nagios Network Analyzer, the alert value being tested will always be 0 or greater (no negative numbers are involved).

Finishing Up

This completes the documentation on how to integrate Slack with your Nagios Network Analyzer. 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