

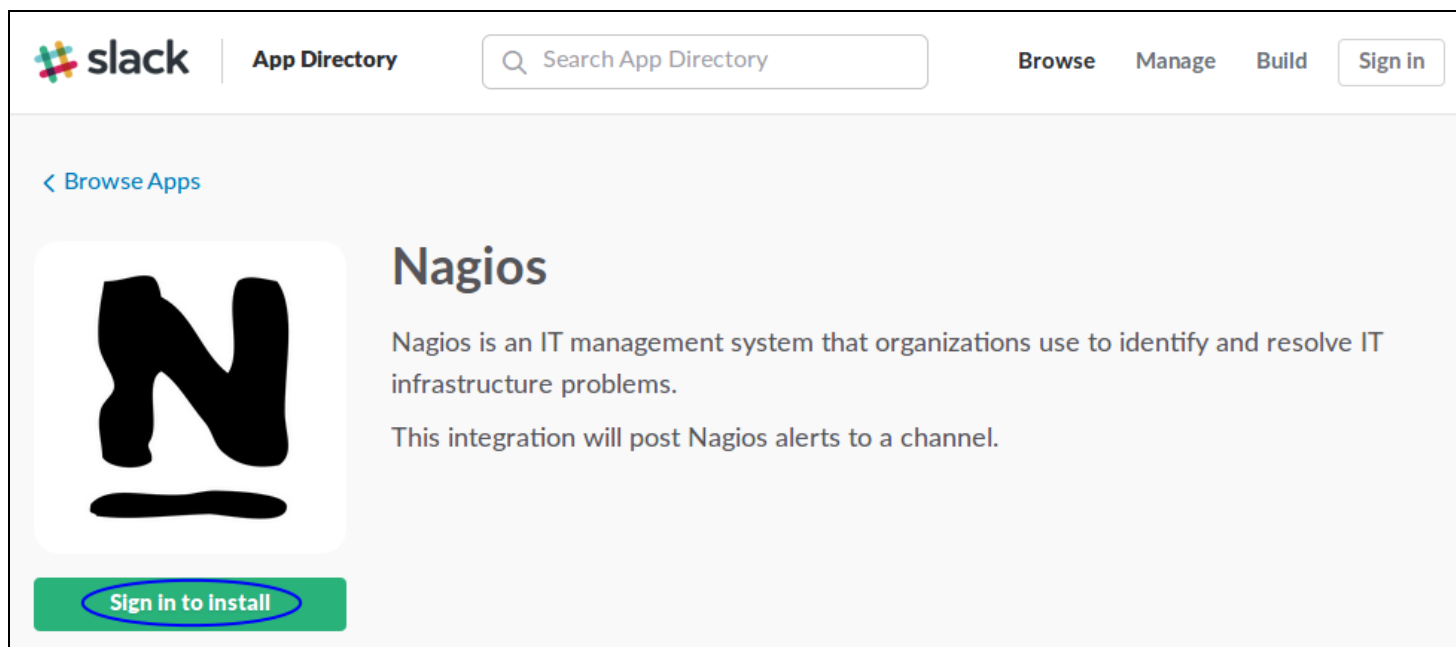
Integrating Slack with NNA 2024

Overview

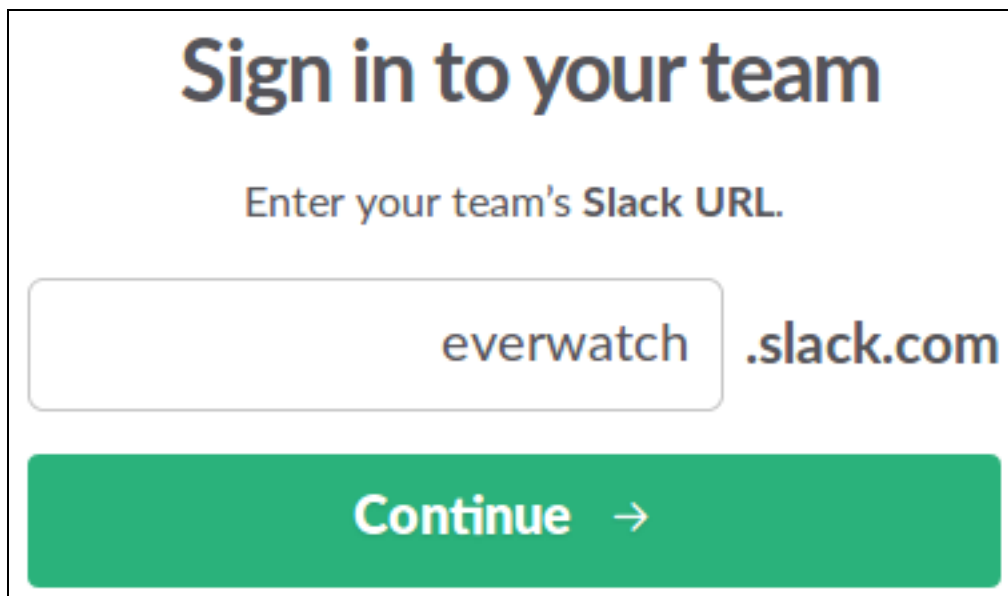
In Slack you have Channels that receive notifications. The Slack API allows you to target these channels by using the channel name, like NNA. This is used to send notifications from Nagios Network Analyzer to Slack. This documentation will create a Nagios Network Analyzer Command that will be used by 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> and you should see a page similar to the following screenshot:



1. Click the Sign in to install button.



The screenshot shows a white rectangular box with a thin black border. At the top, the text "Sign in to your team" is displayed in a large, bold, dark blue font. Below this, the instruction "Enter your team's Slack URL." is written in a smaller, dark blue font. A text input field with rounded corners contains the text "everwatch" in a dark blue font. To the right of the input field, the text ".slack.com" is displayed in a dark blue font. Below the input field and the ".slack.com" text is a large, solid green button with rounded corners. The button contains the text "Continue" in a white font, followed by a white right-pointing arrow.

You will need to provide your team's Slack URL.

2. Populate the field and then click Continue.

Sign in to EverWatch

everwatch.slack.com

Enter your email address and password.

Sign in

Keep me signed in [Forgot password?](#)

You will need to provide your credentials to proceed.

3. Populate the fields and then click Sign in.



Nagios

Server monitoring and alerting.

Nagios is an IT management system that organizations use to identify and resolve IT infrastructure problems.

This integration will post Nagios alerts to a channel.

[Add integration](#)



[Add Configuration](#)

4. Once you have signed in click the Add Configuration button.

5. You will be presented with a summary of the Nagios app. Click the Add integration button.
6. 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.

Integration Settings

Token

This token is used as the key to your Nagios integration.

[Regenerate](#)

Descriptive Label

Use this label to provide extra context in your list of integrations (optional).

Customize Name

Choose the username that this integration will post as.

Customize Icon

Change the icon that is used for messages from this integration.



or

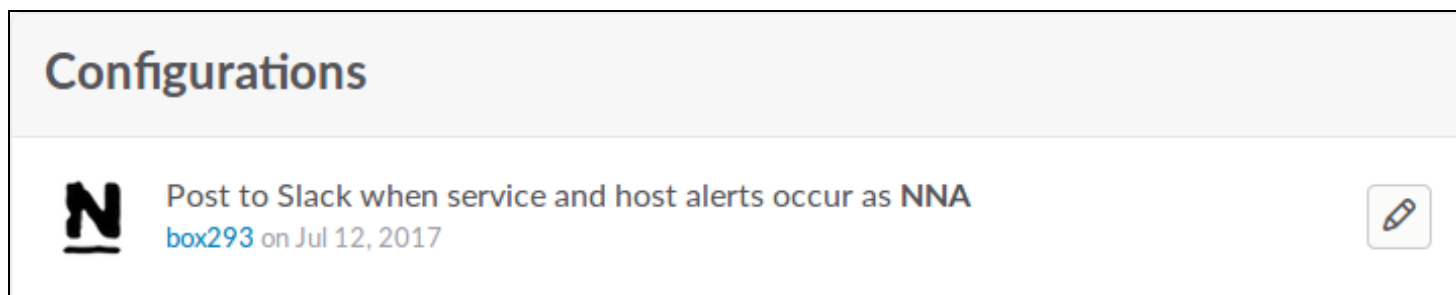
Preview Message

Here's what messages from this integration will look like in Slack.

**NNA** APP 11:16 AM

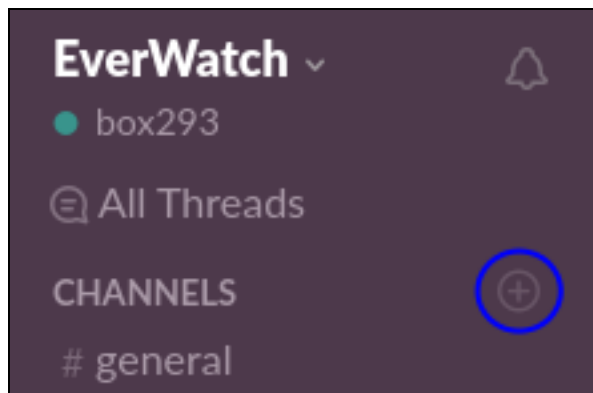
This is what messages from this service will look like in Slack.

7. Scroll down to the Integration Settings section. Take a 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.
8. Click the Save Settings button after making the required changes.



9. Once saved this will appear under your Configurations. You can click the pencil icon to edit it 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 Channels In Slack

1. Open the Slack application and next to the CHANNELS heading click the + icon.

Create a channel

Channels are where your team communicates. They're best when organized around a topic – #leads, for example.

Public Anyone on your team can view and join this channel.

Name

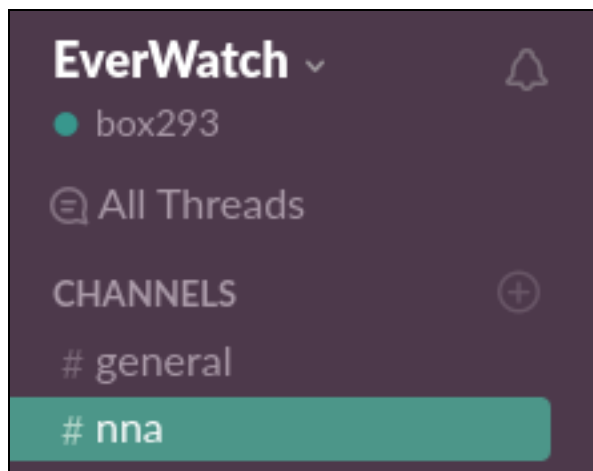
Names must be lowercase, without spaces or periods, and shorter than 22 characters.

Purpose (optional)

What's this channel about?

Send invites to: (optional)

2. On the Create a channel page provide a Name. In the screenshot to the right you can see the channel being created is called nna.
3. Optionally provide a Purpose.
4. Click the Create Channel button once you've populated the fields.



Here you can see the new Channel now exists.

This completes the steps required in Slack. Leave the application open as you'll want to return here once Nagios XI is configured.

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.githubusercontent.com/tinyspeck/services-examples/master/nagios.pl
```

```
chmod 0775 slack_nagios.pl
```

```
chown nna:nnacmd slack_nagios.pl
```

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.pl
```

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 into 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 typing:

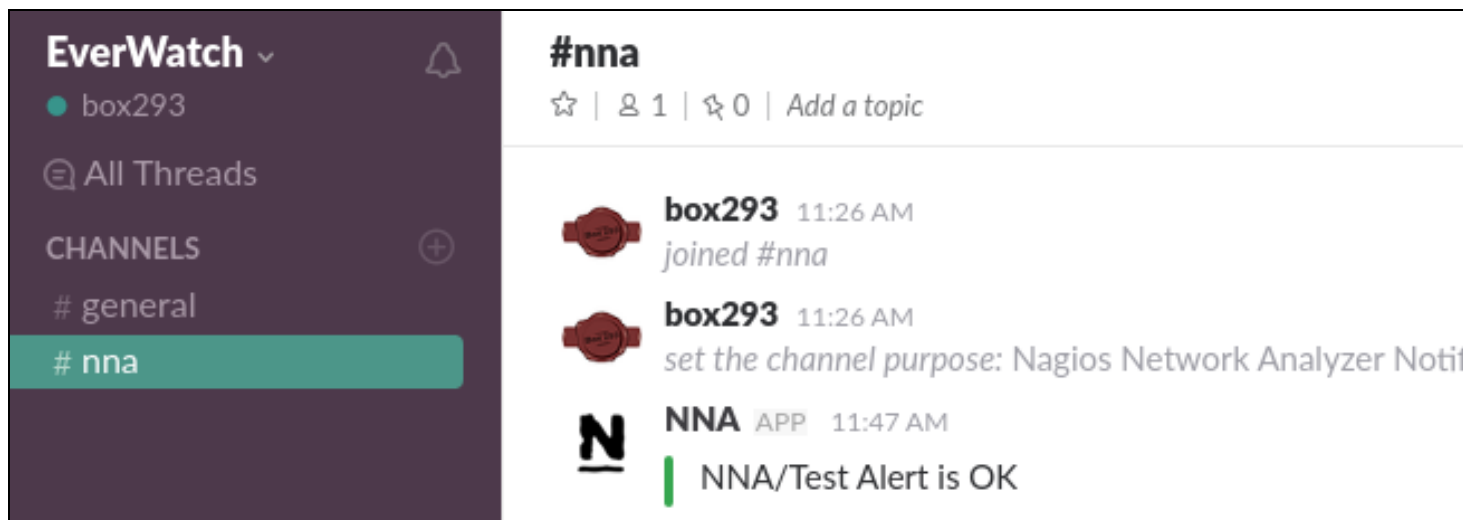
```
:wq
```

and press Enter.

You can test that it works by executing the following command (it's one long command that wraps over three lines):

```
./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 something 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:

```
#!/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
```



The slack line is one long line, it just wraps over three lines due to the length.

When you have finished, save the changes in vi by 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.

The completes all the steps required in the terminal session, you can close it now as the remaining steps will be performed through 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 and click the Commands tab.





Nagios
Network Analyzer™


Admin  nagiosadmin

Dashboard Sources Source Groups Views Reports Queries **Alerting** Help Administration Log Out

Alerting / Commands

Alerting

 Checks  Nagios Setup  SNMP Receivers  **Commands**

 >_ New Command

Name	Script Location	Script Name	Passed Arguments	Actions
No commands created.				

New Command ✕

Specify a script to run when an alert happens.

Name	<input type="text" value="Slack"/>
Script Location	<input type="text" value="/usr/local/nagiosna/scripts"/>
Script Name	<input type="text" value="slack_nagios_wrapper.sh"/>
Passed Arguments	<input %sourcename%\"="" \"%output%\""="" \"%state%\"="" type="text" value="'#nna' \"/>

You can pass some basic macros to the script via arguments that will be auto-populated when the script is executed.

- **%sourcename%** - the name of the source that is being alerted on
- **%state%** - the alert state (ok, warning, critical, unknown)
- **%returncode%** - the return code of the check (0 to 3)
- **%output%** - the full output of the check

Click the >_ New Command button.

Name: Slack

Script Location: /usr/local/nagiosna/scripts

Script Name: slack_nagios_wrapper.sh

Passed Arguments: Please see the full command on the following page.

Click the Create button once you have populated the fields.

The Passed Arguments field is shown below.

```
"#nna" "%sourcename%" "%state%" "%output%"
```

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

Name	Script Location	Script Name	Passed Arguments	Actions
Slack	/usr/local/nagiosna/scripts	slack_nagios_wrapper.sh	"#nna" "%sourcename%" "%state%" "%output%"	Edit • Delete

Create Check

The last remaining step is to define an alert so Slack will receive notifications. The following example creates an alert that will notify if a source has no flow data received on the port that the flow data is being received on. Click the Checks tab and then click the New Check button.

The screenshot shows the Nagios web interface. At the top, there is a navigation bar with tabs: Dashboard, Sources, Source Groups, Views, Reports, Queries, Alerting (selected), Help, Administration, and Log Out. Below the navigation bar, the breadcrumb 'Alerting / Checks' is visible. The main heading is 'Alerting'. Underneath, there are four tabs: Checks (selected), Nagios Setup, SNMP Receivers, and Commands. A '+ New Check' button is circled in blue. Below the tabs is a table with the following columns: Name, Associated With, Last Status, Last Date Ran, Last Stdout, and Actions. The table currently contains the text 'No checks created.'

Step 1 - Select Source ✕

Please name the check for management: (Required)

Source Sourcegroup

Source

Firewall Public

View

No View

Step 1

You must enter a name for the check for management/organizational purposes. It can contain only whitespaces and alphanumeric characters.

Then you need to select your source or source group. This is the source the check will get values from. If you select a source, you can select a view to test against as well.

Click the Step Two button to proceed to Step 2.

Step 2 - Select Criteria ×

Analyze traffic for:

Flows

Warning threshold is:

1:

Critical threshold is:

1:

Where The:

Destination

Port

 is is not

(required)

9914

And

Cancel

◀ Step One

Step Three ▶

Step 2

Analyze traffic for - This is the metric you would like to get the number for to check against. If you want a packet count, pick Packets. If you want total bytes, pick Bytes. There are several other options, but the point is there are multiple dimensions to the traffic on your network, and this specifies which one will be checked.

Warning and Critical - Once Network Analyzer has extracted a number from the metric you selected, it will use these thresholds to determine if the number is in a WARNING, CRITICAL or OK state. In this example 1: means that if less than 1 was received then it will be in a CRITICAL state (meaning no flows were received).

More detailed information on thresholds is explained in the [Nagios Threshold Values](#) section of this document.

The bottom half of step 2 is how you filter what data the check is looking at, this allows granularity.

In the screenshot on the previous page, Flows is the type of traffic being analyzed. The filter criteria used is:

- Destination - The direction of the flow traffic being looked at
- Port - This check 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 - This is the operation, we want to make sure the destination port IS 9914
- 9914 - Here the port number 9914 has been defined

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 at the bottom. It 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.

Step 3 - Select Alerting Methods ✕

Select how you would like to be notified of these checks. Select any of the items in the lists under the tabs, and all the selected elements will be notified.

[Email Users](#) [Nagios](#) [SNMP Traps](#) **Commands**

Select local scripts or commands to be ran when the check happens. Hold ctrl and click to un-elect commands.

Slack

Click the Step Three button to proceed to Step 3.

Here you select the alerting method. In this example the Nagios tab has been selected and the Servicename/Hostname that was previously defined has been selected.

Click the Finish and Save button to create the alert.

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

Alerting

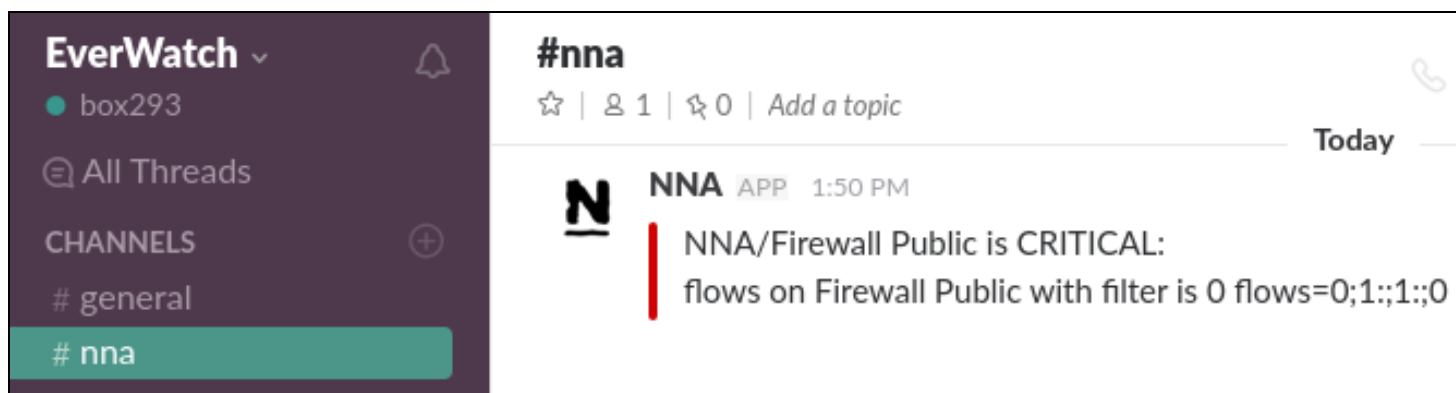
[Checks](#)
[Nagios Setup](#)
[SNMP Receivers](#)
[Commands](#)

[+ New Check](#)

Name	Associated With	Last Status	Last Date Ran	Last Stdout	Actions
Flow Data 9914	Firewall Public	PENDING	N/A		View / Edit • Delete

The checks are run every five minutes, it's important to understand that every five minutes the check will fire off a notification to Slack. Based on the WARNING and CRITICAL thresholds you defined on the check, Slack will receive the check results with the state from the check.

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, however once grasped 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, for example 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).