JSend NSCA 2.0.0

API User Guide

Contents

Overview

JSend NSCA 2.0.0 is a Java API for sending Nagios[1] Passive Checks to the Nagios NSCA add on.

By using the API, you can easily integrate your Java applications into a Nagios monitored environment thereby notifying Nagios of problems and issues during the running of your application.

Background

JSend NSCA was developed as the company I'm working for uses Nagios to monitor applications and servers. For existing applications written is Perl and c, there were options available to send passive checks but for Java application, the option available was to shell out and execute the send_nsca command line tool.

Although send_nsca worked in this manner, it's a bit ugly and we preferred having the code within our applications for better performance, cleanliness, maintainability etc.

A search revealed a few potential options such as the NagiosAppender for log4j but in the end we settled on writing our own client. This client is currently in use thus proving the feasibility of the approach.

On the back of this, I decided to write JSend NSCA from the ground up as an exercise in TDD and thought I would make it available as an open source project so other developers can benefit from the functionality.

[1] For information on Nagios and NSCA see http://www.nagios.org/

API Quick Start

OK, so you want to send passive checks from your java code. This quick start will hopefully have you sending checks in no time.

Pre-requisites

You need to have Nagios and NSCA set up. This is beyond the scope of this document so refer to the documentation available on the Nagios web site.

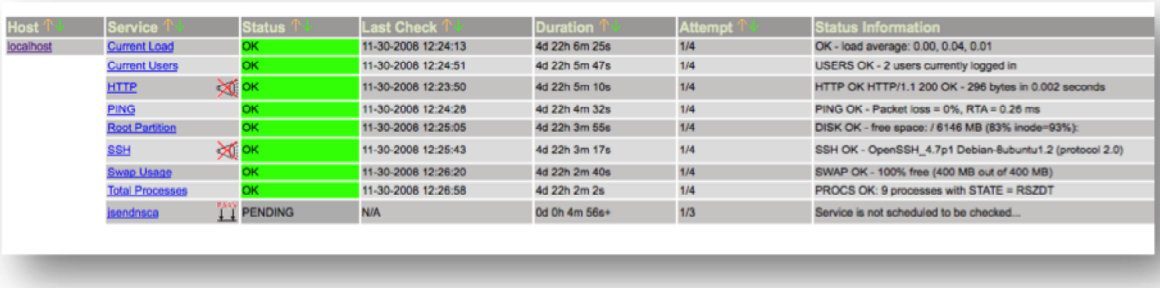
For this quick start, add the following passive check to the localhost.cfg. I will assume Nagios is set up locally, if not make a note of the IP or hostname for use later on.

```
define service{
        use                     template-nsca-service
        host_name               localhost
        service_description     jsendnsca
        check_command           check_dummy
}
```

And restart nagios as shown below

```
raj@ubuntu-server:/usr/local/nagios/etc/objects$ sudo /etc/init.d/nagios restart
Running configuration check...done.
Stopping nagios: done.
Starting nagios: done.
```

Now check localhost on Nagios to see the jsendnsca service as shown below



Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-30-2008 12:24:13	4d 22h 6m 25s	1/4	OK - load average: 0.00, 0.04, 0.01
	Current Users	OK	11-30-2008 12:24:51	4d 22h 5m 47s	1/4	USERS OK - 2 users currently logged in
	HTTP	OK	11-30-2008 12:23:50	4d 22h 5m 10s	1/4	HTTP OK HTTP/1.1 200 OK - 296 bytes in 0.002 seconds
	PING	OK	11-30-2008 12:24:28	4d 22h 4m 32s	1/4	PING OK - Packet loss = 0%, RTA = 0.26 ms
	Root Partition	OK	11-30-2008 12:25:05	4d 22h 3m 55s	1/4	DISK OK - free space: / 6146 MB (83% inode=93%):
	SSH	OK	11-30-2008 12:25:43	4d 22h 3m 17s	1/4	SSH OK - OpenSSH_4.7p1 Debian-8ubuntu1.2 (protocol 2.0)
	Swap Usage	OK	11-30-2008 12:26:20	4d 22h 2m 40s	1/4	SWAP OK - 100% free (400 MB out of 400 MB)
	Total Processes	OK	11-30-2008 12:26:58	4d 22h 2m 2s	1/4	PROCS OK: 9 processes with STATE = RSZDT
	jsendnsca	PENDING	N/A	0d 0h 4m 56s+	1/3	Service is not scheduled to be checked...

OK, so now we have our service set up, we can move onto the code.

Set up API

Download jsendnsca-2.0.0.tar.gz/zip unpack somewhere. The contents are listed below.

```
jsendnsca-core-1.1.1
-LICENSE-2.0.txt
-apidocs
-jsendnsca-2.0.0-javadoc.jar
-jsendnsca-2.0.0-sources.jar
-jsendnsca-2.0.0.jar
-lib
--commons-lang-2.4.jar
-pom.xml
-src
```

Add jsendnsca-2.0.0.jar and lib/commons-lang-2.4.jar to your class path so it can be referenced by your code.

Code to send passive check

To show sending of a passive check, we will build up a class to send the check

Set up a basic class with a main as below

```
package com.googlecode.jsendnsca.quickstart;

public class QuickStart {

        public static void main(String[] args) {
        }

}
```

Now lets build some settings for Nagios NSCA using the NagiosSettingsBuilder.

```
NagiosSettings settings = new NagiosSettingsBuilder()
    .withNagiosHost("nagiosHostNameOrIPAddress")
    .withPort(5667)
    .withEncryption(Encryption.XOR)
    .create();
```

Defaults are as follows so you only need to set what differs

Setting	Default value
Nagios Host	localhost
Port	5667
Password	password
Timeout	10000 ms
Connect Timeout	5000 ms
Encryptor	none

Now our settings have been created, we can create the passive check itself.

```
MessagePayload payload = new MessagePayloadBuilder()
    .withHostname("localhost")
    .withLevel(Level.OK)
    .withServiceName("jsendnsca")
    .withMessage("it works!")
    .create();
```

The message we have have constructed is being sent from localhost, is OK, for service jsendnsca and the message is “It works!”

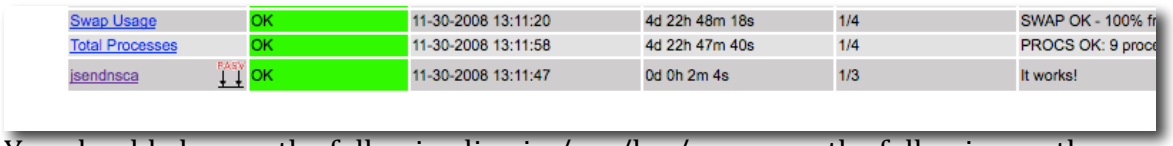
We are explicitly setting the hostname to localhost in this example which can be useful for testing when sending from different hosts (no need to configure each host on Nagios). Alternatively you can use .withLocalHostname() or .withCanonicalHostname to determine your short and fully qualified domain name respectively for you

To send the passive check we have created, we construct a NagiosPassiveCheckSender using our settings and send the MessagePayload.

```
NagiosPassiveCheckSender sender = new NagiosPassiveCheckSender(settings);

try {
    sender.send(payload);
} catch (NagiosException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

If you run this class now, it will send a passive check. After checking the service on Nagios, you should see the entry change as shown below



Swap Usage		OK	11-30-2008 13:11:20	4d 22h 48m 18s	1/4	SWAP OK - 100% fr
Total Processes		OK	11-30-2008 13:11:58	4d 22h 47m 40s	1/4	PROCS OK: 9 proc
jsendnsca	PASV	OK	11-30-2008 13:11:47	0d 0h 2m 4s	1/3	It works!

You should also see the following line in /var/log/messages the following on the Nagios host

```
Nov 30 13:11:48 ubuntu-server nagios: EXTERNAL COMMAND:
PROCESS_SERVICE_CHECK_RESULT;localhost;jsendnsca;0;It works!
```

Complete quick start code

```
package com.googlecode.jsendnsca.quickstart;

import java.io.IOException;

import com.googlecode.jsendnsca.Level;
import com.googlecode.jsendnsca.MessagePayload;
import com.googlecode.jsendnsca.NagiosException;
import com.googlecode.jsendnsca.NagiosPassiveCheckSender;
import com.googlecode.jsendnsca.NagiosSettings;
import com.googlecode.jsendnsca.builders.MessagePayloadBuilder;
import com.googlecode.jsendnsca.builders.NagiosSettingsBuilder;
import com.googlecode.jsendnsca.encryption.Encryption;

public class QuickStart {

    public static void main(String[] args) {
        NagiosSettings settings = new NagiosSettingsBuilder()
            .withNagiosHost("localhost")
            .withPort(5667)
            .withEncryption(Encryption.XOR)
            .create();

        MessagePayload payload = new MessagePayloadBuilder()
            .withHostname("hostname of machine sending check")
            .withLevel(Level.OK)
            .withServiceName("jsendnsca")
            .withMessage("It works!")
            .create();

        NagiosPassiveCheckSender sender = new NagiosPassiveCheckSender(settings);

        try {
            sender.send(payload);
        } catch (NagiosException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```