**CYBER FORZA**

A reference document used to Monitoring Windows Server Using Nagios

Revision: Draft specification 1.0

July 2017

**Cyber Forza Inc. Confidential and Proprietary**

Copyright © 2017, Cyber Forza Incorporation. All rights reserved.

This document contains information that is considered proprietary and the property of Cyber Forza Inc. The content of this document may not be used in any way, duplicated, or distributed by any means without the express written permission of Cyber Forza Inc

Unless otherwise agreed in writing, by Cyber Forza Inc., The Cyber Forza products are not designed or intended for any application in which the failure of the product could create a situation where personal injury or death may occur.

Cyber Forza Inc. may change specifications and product description at any time, without notice.

Contact Cyber Forza Inc. to obtain latest specification.

# Contents

1 [1 Revision History](#_1_Revision_History)……………………………………………………………….…….4

[2 Quick Start Guide](#_2_Quick_Start) ……………………………………………………………….…… 5

[2.1 System Specification](#_2.1_System_Specification)….………………………………………………………… … 5

[2.2 Support Information](#_2.2__Support)...…………………………………....…………………………5

[2.3 Getting Started](#_2.3__Getting)………………………………….………….………………………. 5

[2.3.1 Installing the Windows Agent](#_2.3.1_Installing_the) …………………………………………………….6

[3 Nagios Server](#_3__) …………….…………………………………………………………10

[3.1 Configuration in the Nagios server](#_3.1_Configuration_in).……………………………………………….. 10

[3.2 Prerequisites and examples](#_3.2_Prerequisites_and) ………………………………………..……………….10

3.3 [3.3 Password Protection](#_3.3_Password_Protection)………….………………………………….…………….. 13

|  |  |  |
| --- | --- | --- |
| Date | Revision No. | Description |
| 12/07/17 | Draft Release 1.0 | Initial version |
|  |  |  |

# 1 Revision History

# 2 Quick Start Guide

# 2.1 System Specification

*Table 1: System Configuration*

|  |
| --- |
| * Ubuntu 16.04.1 LTS (Linux Kernel 4.4) |
| * System RAM: 3911776 KB |
| * Hard Disk space: 21.1 GB |
| * OS type: 64-bit |
| * Memory: 3.7 GiB |
| * Processor: Intel Core 2 Duo CPU E8400 @ 3.00GHz x 2 |

# 2.2 Support Information

* Please contact the Cyber Forza Inc. team for support

# 2.3 Getting Started

Monitoring private services or attributes of a Windows machine requires that you install an agent on it. This agent acts as a proxy between the Nagios plugin that does the monitoring and the actual service or attribute of the Windows machine.

To be able to monitoruig Windows server using Nagios we need to Install a Window Agent:

NSClient++ is the most widely used

we will be installing the NSClient++ addon on the Windows machine and using the *check\_nt* plugin to communicate with the NSClient++ addon.

# 2.3.1 Installing the Windows Agent

The NSClient++ agent must be installed on the target Windows machine and configured before Nagios can

monitor system metrics, services, processes, or performance data on the target machine using the

Windows Server configuration wizard.

We download the NSClient++ from <http://nsclient.org>

*NSCP-0.5.0.62-x64.msi* Windows Installer.

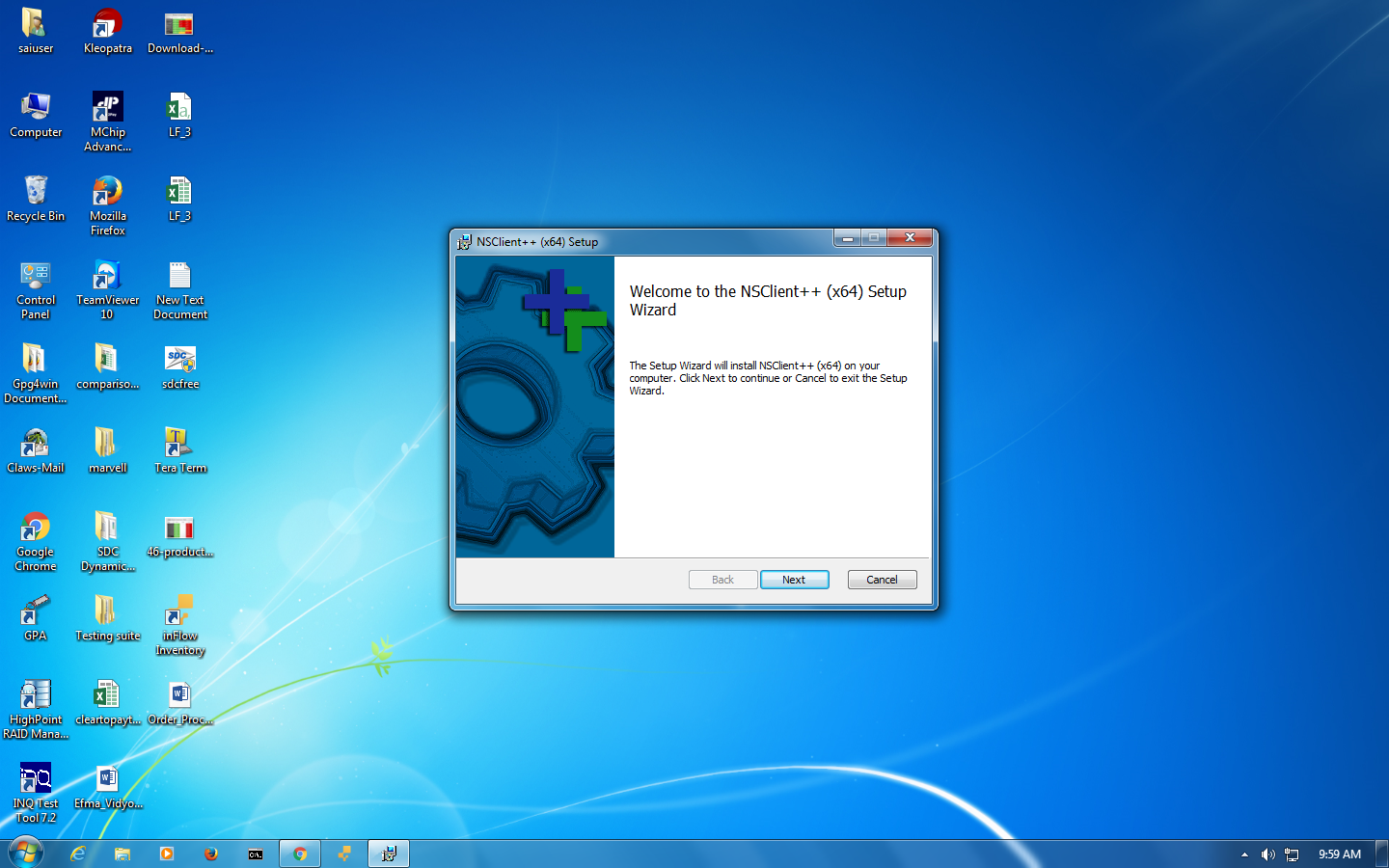


Figure1: NSClient++ Wizard

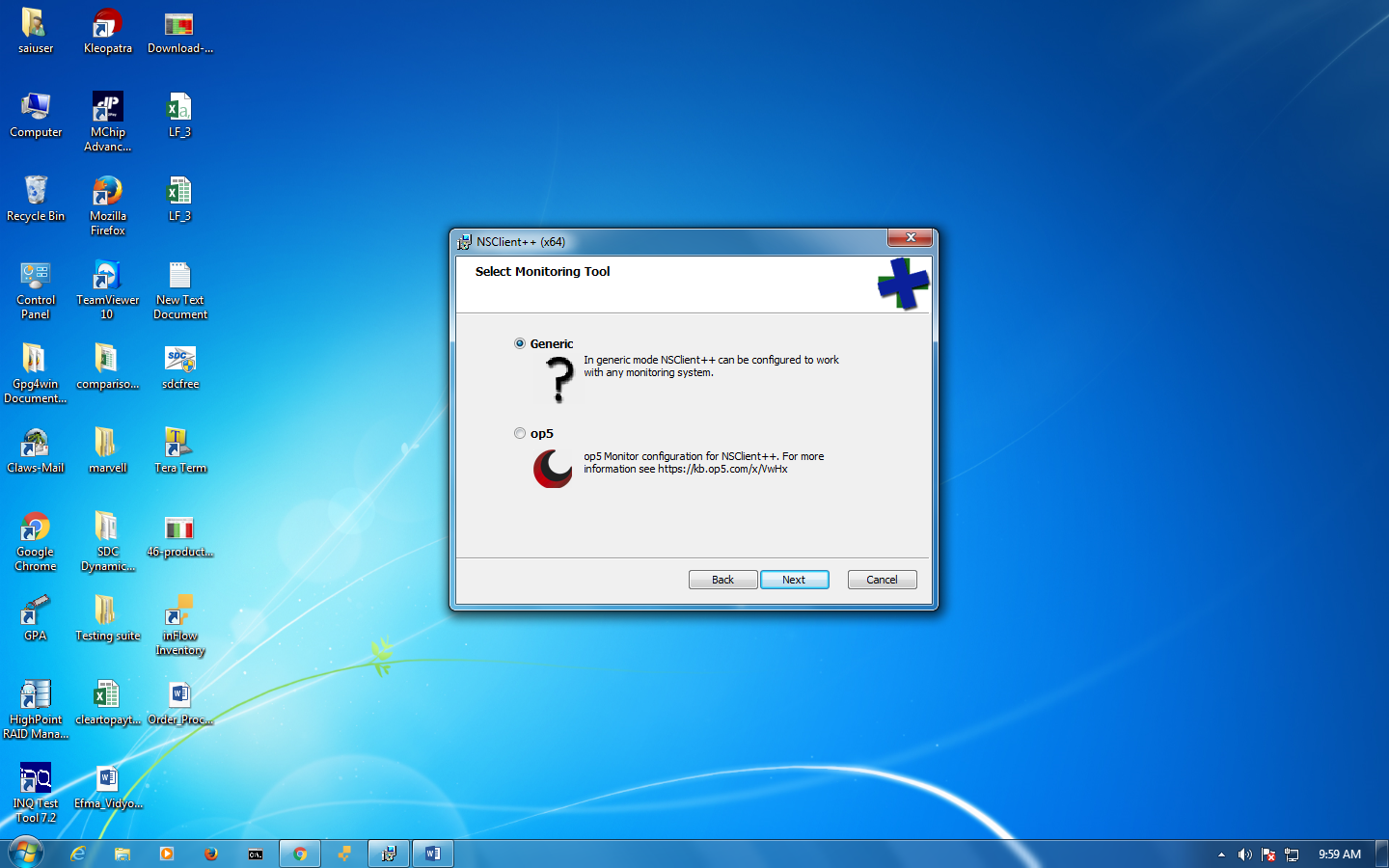


Figure2: We chose Generic option

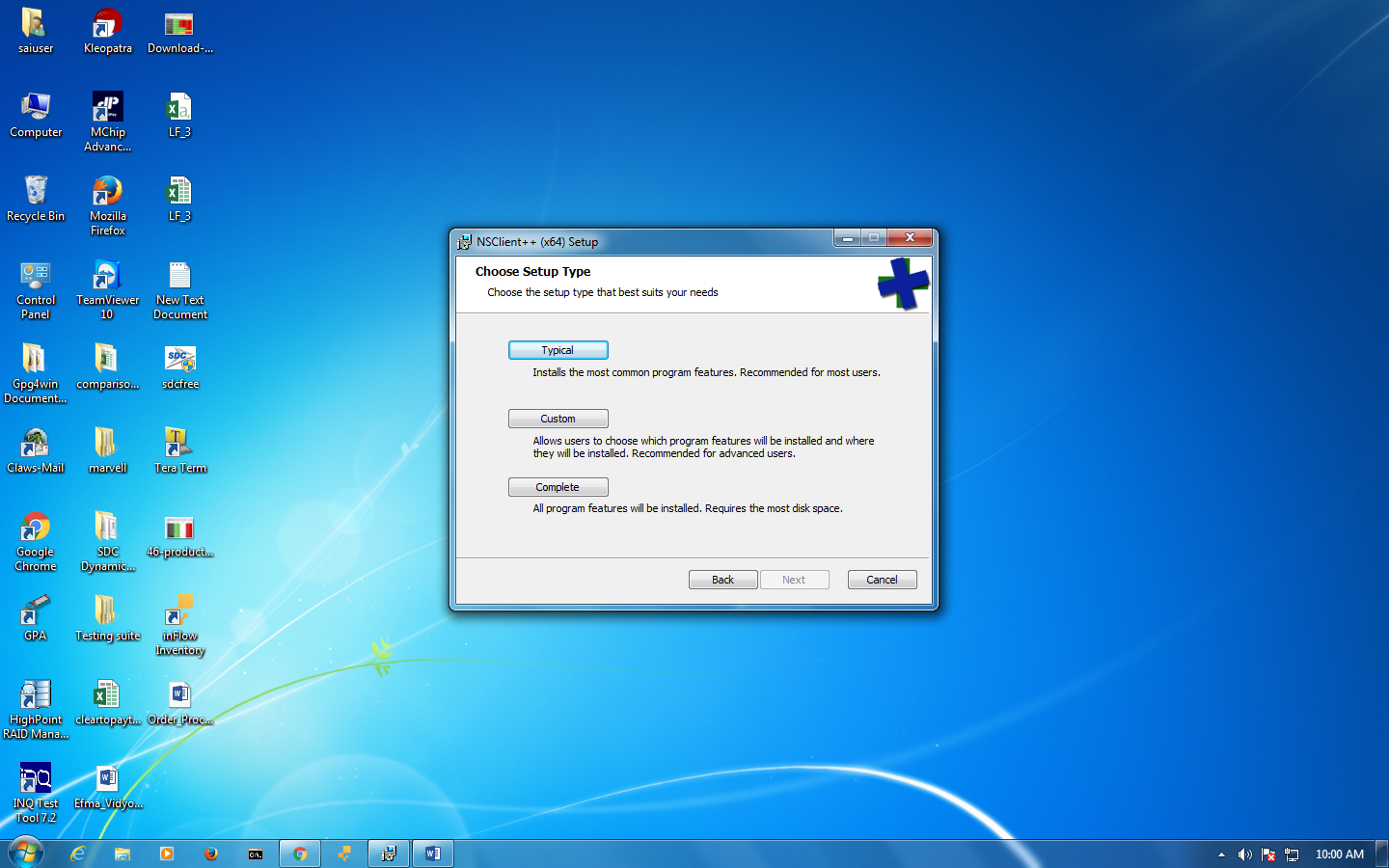


Figure3: We chose Typical option

Enter the IP address(s) of the Nagios XI server(s) in the

Allowed hosts box. In this example you can see two addresses have been added (10.25.5.11 and 10.25.5.12 separated by a comma).

Enter a password in the Password box that is required for communication between the Nagios server and the Windows machine (used by check\_nt plugin).

Enable common check plugins = Checked

Enable nsclient server (check\_nt) = Checked

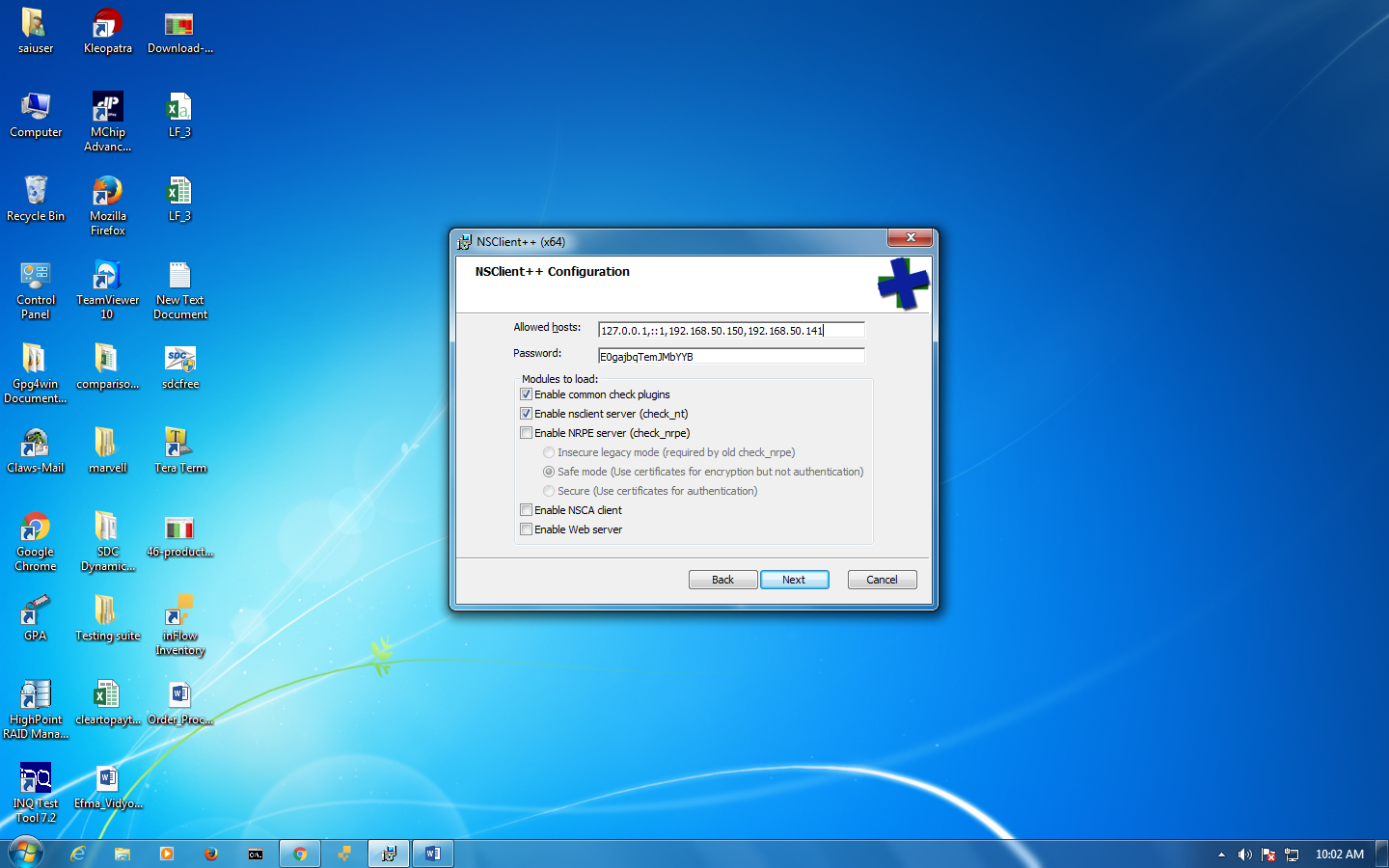


Figure 4: Click Next to continue

Click the Install button on the next screen to begin the installation up to you finished it.

In the file *nsclient.ini* in the line:

*allowed\_hosts*

you will need to add the IP address of your Nagios Server.

You will need verify that NSClient++ (x64) is running as Service in your Windows server.

In the Nagios Serve a few configuration tasks have already been done for you:

A *check\_nt* command definition has been added to the *commands.cfg* file. This allows you to use the *check\_nt* plugin to monitor Window services.

# 3 Nagios Server

# 3.1 Configuration in the Nagios server

A Windows server host template (called windows-server) has already been created in the *templates.cfg* file. This allows you to add new Windows host definitions in a simple manner.

The above-mentioned config files can be found in the */usr/local/nagios/etc/objects/* directory. You can modify the definitions in these and other definitions to suit your needs better if you'd like.

# 3.2 Prerequisites and examples

The first time you configure Nagios Core to monitor a Windows machine, you should do the following:

Edit the main Nagios Core config file.

*vi /usr/local/nagios/etc/nagios.cfg*

Remove the leading pound (#) sign from the following line in the main configuration file:

*#cfg\_file=/usr/local/nagios/etc/objects/windows.cfg*

Save the file and exit.

Now it's time to define some object definitions in your Nagios Core configuration files in order to monitor the new Windows machine.

Open the windows.cfg file for editing.

*vi /usr/local/nagios/etc/objects/windows.cfg*

Add a new host definition for the Windows machine that you're going to monitor. If this is the \*first\* Windows machine you're monitoring, you can simply modify the sample host definition in *windows.cfg.* Change the host\_name, alias, and address fields to appropriate values for the Windows box.

*define host {*

*use windows-server ; Inherit default values from a Windows server template (make sure you keep this line!)*

*host\_name winserver*

*alias My Windows Server*

*address 192.168.50.150*

}

Now you can add some service definitions (to the same configuration file) in order to tell Nagios Core to monitor different aspects of the Windows machine. If this is the \*first\* Windows machine you're monitoring, you can simply modify the sample service definitions in *windows.cfg*.

You will found the following services already defined in that file:

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description NSClient++ Version*

*check\_command check\_nt!CLIENTVERSION*

*}*

The following service definition to monitor the uptime of the Windows server.

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description Uptime*

*check\_command check\_nt!UPTIME*

*}*

The following service definition to monitor the CPU utilization on the Windows server and generate a CRITICAL alert if the 5-minute CPU load is 90% or more or a WARNING alert if the 5-minute load is 80% or greater.

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description CPU Load*

*check\_command check\_nt!CPULOAD!-l 5,80,90*

*}*

The following service definition to monitor memory usage on the Windows server and generate a CRITICAL alert if memory usage is 90% or more or a WARNING alert if memory usage is 80% or greater.

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description Memory Usage*

*check\_command check\_nt!MEMUSE!-w 80 -c 90*

*}*

The following service definition to monitor usage of *the C:\* drive on the Windows server and generate a CRITICAL alert if disk usage is 90% or more or a WARNING alert if disk usage is 80% or greater.

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description C:\ Drive Space*

*check\_command check\_nt!USEDDISKSPACE!-l c -w 80 -c 90*

*}*

The following service definition to monitor the W3SVC service state on the Windows machine and generate a CRITICAL alert if the service is stopped.

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description W3SVC*

*check\_command check\_nt!SERVICESTATE!-d SHOWALL -l W3SVC*

*}*

The following service definition to monitor the Explorer.exe process on the Windows machine and generate a CRITICAL alert if the process is not running.

*define service {*

*use generic-service*

*host\_name winserver*

*service\_description Explorer*

*check\_command check\_nt!PROCSTATE!-d SHOWALL -l Explorer.exe*

*}*

# 3.3 Password Protection

If you specified a password in the NSClient++ configuration file on the Windows machine, you'll need to modify the *check\_nt* command definition to include the password. Open the *commands.cfg* file for editing.

*vi /usr/local/nagios/etc/objects/commands.cfg*

Change the definition of the check\_nt command to include the "-s <PASSWORD>" argument (where PASSWORD is the password you specified on the Windows machine) like this:

*define command {*

*command\_name check\_nt*

*command\_line $USER1$/check\_nt -H $HOSTADDRESS$ -p 12489 -s PASSWORD -v $ARG1$ $ARG2$*

*}*

Save the file.

Restart nagios.

After that you should be able to see the windows server , as I show you in the next image. In this case we keep the generic name winserver:

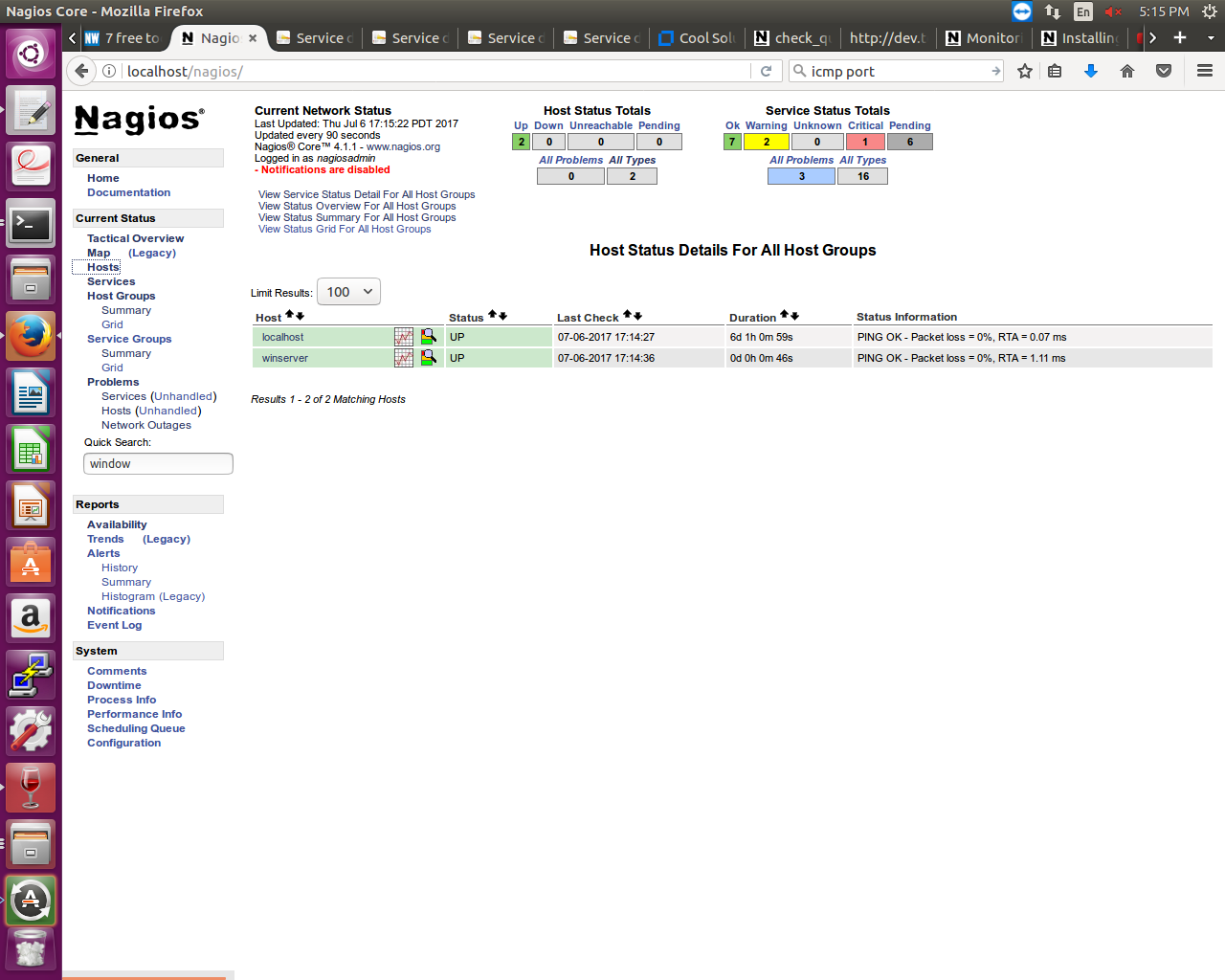


Figure 5: Nagios web Interface where showed up the windows server being monitored