

Centreon Web Application Analytics Documentation

Release 1.1.0

Centreon

April 26, 2016

Centreon Web Application Analytics is an OpenSource Nagios plugin under GPL license. This plugin is used to check a website with user experience. It is based on the product [Selenium](#).

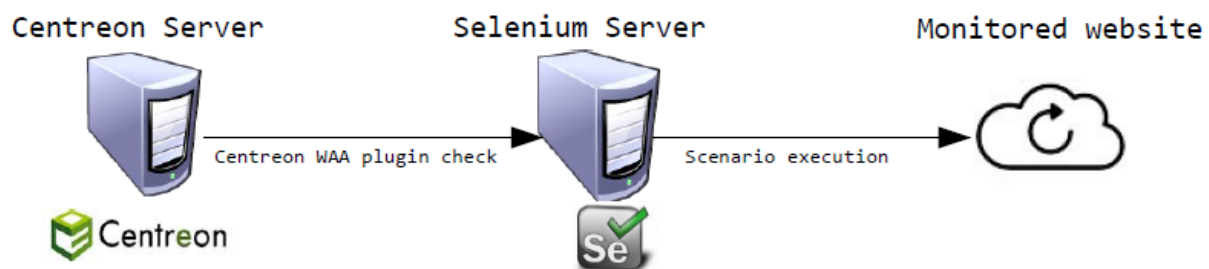
Contents:

How it works

Centreon Waa uses a second server on which the Selenium server and the Firefox browser are running.

When a check is executed, there are three steps:

- Centreon-engine executes the check_centreon_waa plugin
- This check connects to the Selenium server and asks for scenario actions
- The Selenium server executes the actions within a Firefox and send the result (execution duration and number of successful step/number of total steps).



How selenium works is described [here](#).

Installation

2.1 Prerequisites

Note: Centreon advises to install the Selenium server on a dedicated poller for web scenarios monitoring and not on an existing monitoring poller. This allows to do not impact monitoring of hosts and services.

Hardware requirements for the server:

- 6 vCPU 3 GHz
- 8 Gb RAM
- a minimum filesystem space of 30 Gb

Software requirements:

- A monitoring server with Centreon Engine
- Firefox minimum version or Iceweasel 27.0.1
- Selenium version 2.40

2.2 From packages

2.2.1 Selenium server

This server could be installed on a **monitoring poller**. If you plan to deploy huge number or complex scenarios, we strongly recommend to use a **dedicated server** to run Selenium.

If you are using Centreon Enterprise Server. Type the following command to install Selenium-server and all the dependencies needed

```
# yum install centreon-selenium-server
```

To start Selenium and xorg-x11-server-Xvfb services use the following commands

```
# /etc/init.d/xvfb start
# chkconfig --add xvfb
# chkconfig --level 2345 xvfb on
# /etc/init.d/selenium start
# chkconfig --add selenium
# chkconfig --level 2345 selenium on
```

2.2.2 Centreon Selenium Plugin

If you are using Centreon Enterprise Server with Plugins Pack repository. Type the following commands to install the plugin and associated host/service templates

```
# yum install ces-packs-applications-selenium
# yum install ces-plugins-applications-selenium
```

Note: If you do not have the plugin-packs license, please follow installation steps described above in the “From sources >

2.3 From sources

Init scripts and variables files needed for the Selenium server are provided in this repository : <https://github.com/centreon/centreon-web-application-analytics> They are NOT needed when using RPM packaging.

2.3.1 Desktop installation

The desktop is for creating scenarios with Selenium IDE.

Installation:

- Start Firefox
- Go to *Selenium download page* <<http://seleniumhq.org/download/>>
- In section *Selenium IDE*, download the last release of Selenium IDE
- Validate the XPI
- Restart Firefox

After the restart of Firefox, you can find the Selenium IDE in “Tools > Selenium IDE”

2.3.2 Selenium server installation

This server runs the Selenium RC server which drives the Firefox browser.

Warning: You must verify the compatibility between Firefox and Selenium server. This information is in Selenium server [Changelog](#). For example, if you have Firefox 10 or below, you must use Selenium server version 2.20.0 or below.

2.3.3 Java installation

The minimal Java version is 1.6.

On Debian:

```
# apt-get install sun-java6-jre sun-java6-bin
```

On CentOS or CES:

```
# yum install java-1.6.0-openjdk
```

For other installation, go to the *java site* <<http://www.java.com>> and download the JRE.

2.3.4 Virtual X server installation

The Selenium server must run a browser for executing scenarios. An X server must be installed.

For lighter installation, we will use the Framebuffer server (xvfb).

On Debian:

```
# apt-get install xvfb
```

On CentOS or CES:

```
# yum install xorg-x11-server-Xvfb
```

To start the server on boot, a script is available in the Git. To install this script, copy the `init-xvfb` for your distribution into `/etc/init.d` and the `default-xvfb` into `/etc/default`.

To activate this start options:

On Debian:

```
# chmod a+x /etc/init.d/xvfb
# update-rc.d xvfb defaults
# mkdir -p /usr/local/labkey/
```

On CentOS or CES:

```
# chmod a+x /etc/init.d/xvfb
# chkconfig --add xvfb
# mkdir -p /usr/local/labkey/
```

The configuration variables are:

- **X_SERVER_NUMBER** : The X display port
- **FBDIR** : The directory for cache framebuffer file

2.3.5 Browser installation

The browser must be a Firefox or Iceweasel.

On Debian:

```
# apt-get install iceweasel
```

On CentOS or CES:

```
# yum install firefox
```

2.3.6 Selenium server installation

The Selenium server is a JAR archive. We can download this archive from the [selenium download page](#) in the “Selenium Server” section. We copy the downloaded archive into a directory and make a symbolic link to make the upgrade easier.

Example:

```
# mkdir /opt/selenium
# cd /opt/selenium
# cp ~/selenium-server-standalone-version.jar /opt/selenium
# ln -sf selenium-server-standalone-version.jar selenium-server-standalone.jar
```

To start the server on boot, a script is available in the centreon waa source package. To install this script, copy the `init-selenium` from Git into `/etc/init.d` and the `default-selenium` into `/etc/default`.

To activate this start options:

On Debian:

```
# useradd -r -s /bin/bash -d /var/run/selenium -m selenium
# mkdir -p /var/log/selenium
# chown selenium: /var/log/selenium
# chmod a+x /etc/init.d/selenium
# update-rc.d selenium defaults
```

On CentOS or CES:

```
# useradd -r -s /bin/bash -d /var/run/selenium -m selenium
# mkdir -p /var/log/selenium
# chown selenium: /var/log/selenium
# chmod a+x /etc/init.d/selenium
# chkconfig --add selenium
```

The configuration variables are:

- **SELENIUM_LIB** : The path to the Selenium JAR
- **SELENIUM_PORT** : The listening port for Selenium server
- **SELENIUM_LOGDIR** : The log directory
- **SELENIUM_PID** : The path for PID file
- **SELENIUM_FFPROFILE** : The Firefox profile used to run the scenarios
- **X_DISPLAY** : The X display port

2.4 Centreon WAA Plugin

This check must be installed on the **monitoring server** (central or poller). We strongly recommend to use a **poller**

2.4.1 PERL requirements

The list of perl plugins:

- **Getopt::Long**
- **Time::HiRes**
- **XML::XPath**
- **WWW::Selenium**

On Debian:

```
# apt-get install libtest-www-selenium-perl
```

On CentOS or CES with epel repository:

```
# yum install perl-Test-WWW-Selenium perl-XML-XPath
```

With CPAN:

```
# cpan -i Getopt::Long Time::HiRes XML::XPath WWW::Selenium
```

2.4.2 Plugin installation

To install the plugin, it is necessary to get Centreon Plugins project.

```
# cd /tmp
# git clone http://git.centreon.com/centreon-plugins.git
# mv centreon-plugins/* /usr/lib/nagios/plugins/
```

2.4.3 Scenario directory

This check uses a Selenium scenario in HTML format, these scenarios are copied into a directory:

```
# mkdir /var/lib/centreon_waa
# chown centreon-engine:centreon-engine: /var/lib/centreon_waa
```


Check configuration

This part describes the configuration of the Centreon Web.

Note: If you use plugin-packs and have installed ces-packs-applications-selenium and ces-plugins-applications-selenium, please ignore these steps. You can directly use App-Selenium-WAA host template and check “Create service linked to template”.

3.1 Command configuration

The command is configured in the “Configuration > Commands > Checks”. To add a new command, click on the **Add** link.

We complete the form with the following information:

- Command Name : App-Selenium-Scenario
- Command Type : Check
- Command Line : `$USER1$/centreon_plugins.pl --plugin apps::selenium::plugin --mode scenario --selenium-hostname $_HOSTSELENIUMHOST$ --selenium-port $_HOSTSELENIUMPORT$ --directory $_HOSTSCENARIODIR$ --scenario $_SERVICESCENARIONAME$ --timeout $_SERVICETIMEOUT$ --warning $_SERVICEWARNING$ --critical $_SERVICECRITICAL$`

Configuration > Commands > Checks

Add a Command

Check

Command Name *

Command Type Notification Check Misc Discovery

Command Line *

```
$USER1$/centreon_plugins.pl --plugin apps::selenium::plugin --mode scenario --selenium-hostname
$_HOSTSELENIUMHOST$ --selenium-port $_HOSTSELENIUMPORT$ --directory $_HOSTSCENARIODIR$
--scenario $_SERVICESCENARIONAME$ --timeout $_SERVICETIMEOUT$ --warning
$_SERVICEWARNING$ --critical $_SERVICECRITICAL$
```

3.2 Service template configuration

To facilitate the service configuration, we configure a service template.

The service template is configured in the **Configuration => Services => Templates*** to add a new service template, click on the ****Add** link.

We complete the form with the following information:

- Alias : Scenario-Selenium
- Service Template Name : App-Selenium-Scenario-WAA
- Template Service Model : generic-service
- Check Command : App-Selenium-Scenario

Define the following macros:

- SCENARIO_NAME = @NAMEOFTHETEST@
- TIMEOUT = @CENTENGINE_TIMEOUT@
- WARNING = @WARNINGEXECUTIONTIME@
- CRITICAL = @CRITICALEXECUTIONTIME@

Configuration > Services > Templates

Service Configuration Relations Data Processing Service Extended Info

Modify a Service Template Model

General Information

Alias : Scenario-Selenium

Service Template Name : App-Selenium-Scenario-WAA

Service Template Model : generic-active-service

Service State

Is volatile : Yes No Default

Check Period : 24x7

Check Command : App-Selenium-Scenario

Args

Argument	Value	Example
No argument found for this command		

Max Check Attempts :

Normal Check Interval : * 60 seconds

Retry Check Interval : * 60 seconds

Active Checks Enabled : Yes No Default

Passive Checks Enabled : Yes No Default

Macros

Custom macros

Add a new entry +

Macro name : CRITICAL	Macro value : 30
Macro name : SCENARIO_NAME	Macro value : test
Macro name : TIMEOUT	Macro value : 40
Macro name : WARNING	Macro value : 20

3.3 Host configuration

Add a new Host through **Configuration => Hosts => Add**.

Specify hostname, alias, IP address and select **generic-host** template. You must define following macros:

- SCENARIODIR = /var/lib/centreon_waa [Local directory containing scenario]
- SELENIUMHOST = 192.168.1.1 [Set the IP Address of your Selenium server]
- SELENIUMPORT = 4444 [Change according to your configuration]

You can create a host template if you want, as in the screenshot below:

The screenshot shows the 'Add a Host' configuration page in Centreon. The page is divided into several sections:

- General Information:** Host Name: App-Selenium-WAA, Alias: Host template to perform web test soe, IP Address / DNS: (empty), SNMP Community & Version: (empty), Monitored from: Central, Host Templates: generic-host.
- Host Check Properties:** Check Period: (empty), Check Command: (empty), Args: (empty), Max Check Attempts: (empty), Normal Check Interval: 60 seconds, Retry Check Interval: 60 seconds, Active Checks Enabled: (radio buttons for Yes, No, Default), Passive Checks Enabled: (radio buttons for Yes, No, Default).
- Macros:** Custom macros section with three entries:
 - Macro name: SELENIUMHOST, Macro value: 192.168.1.1, Password: (empty)
 - Macro name: SELENIUMPORT, Macro value: 4444, Password: (empty)
 - Macro name: SCENARIODIR, Macro value: /var/lib/centreon_waa, Password: (empty)

3.4 Service configuration

The service is configured in the “Configuration > Services > Service by hosts” to add a new service, click on the **Add** link.

We complete the form with the following information:

- Description : Scenario01 : The service name
- Service template : App-Selenium-Scenario-WAA
- The macro **SCENARIO** to specify the test associated to this service (you can override some of the macro defined in the service template if you want).

Tips and tricks

4.1 How to create a firefox profile

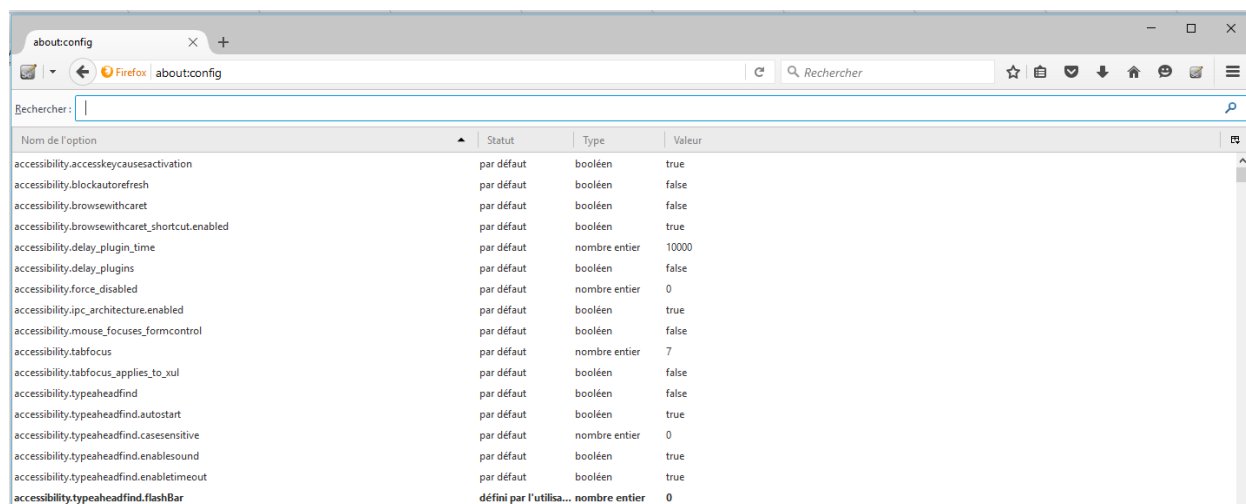
Use your firefox for create a profile. Add your information into this profile (add proxy configuration, add-ons...).

To optimize your firefox profile for Selenium. Open the “Run” windows and enter the following command

```
firefox -p
```

Create a new user profile and start Firefox with this new user. In the URL bar type :

```
about:config
```



Finally, enter set the following parameters to optimize your selenium profile

```
browser.cache.disk.capacity = 0
browser.cache.disk.enable = false
browser.cache.disk.smart_size.enabled = false
browser.cache.memory.enable = false
browser.cache.offline.enable = false
app.update.auto = false
app.update.enabled = false
app.update.silent = false
```

When the configuration is finished, you must copy the profile into a directory in Selenium server. You must remove cache directory from profile directory and verify the size of this profile. If the profile if to large, it can cause some

performance issues. For finish the configuration, you must modify the default script of Selenium for change the SELENIUM_FFPROFILE and restart the Selenium server.

4.2 Deactivate add ons auto update

The auto update block the start of firefox browser and cause a Nagios timeout. For deactivate the auto update, go to **Options > Advanced > Update** and uncheck the options in section **Automatically check for updates to**

4.3 Pause in scenario

If you have some latencies problem in your site, you can add some pause in your scenario and think to use clickAndWait action.

4.4 Provide details in case of an error in a scenario

When an scenario fails, it is not always easy to find the cause of the problem. Newer versions of the plugin will print the selenium command that failed, but it is not so readable.

To help find the cause of problems, it is possible to add text information before one or more selenium commands. In case of an error, the plugin will display this last information that was read in the scenario.

This feature can be used with the “echo” selenium command. This is an example of a scenario that fails:

The screenshot shows the Selenium IDE 2.9.0 interface. The main window displays a table of commands for a test case named 'scenario01'. The table has three columns: 'Commande', 'Cible', and 'Valeur'. The commands are as follows:

Commande	Cible	Valeur
open	/centreon/index.php	
type	name=useralias	admin
type	name=password	centreon
clickAndWait	name=submit	
click	link=Monitoring	
clickAndWait	link=Services	
type	id=host_search	camera
clickAndWait	link=All Services	

Below the table, the 'Commande' dropdown is set to 'open', the 'Cible' field contains '/centreon/index.php', and the 'Valeur' field is empty. The 'Log' window at the bottom shows the following output:

```
[info] Playing test case scenario01
[info] Executing: |open | /centreon/index.php | |
[info] Executing: |type | name=useralias | admin |
[info] Executing: |type | name=password | centreon |
[info] Executing: |clickAndWait | name=submit | |
[info] Executing: |click | link=Monitoring | |
[info] Executing: |clickAndWait | link=Services | |
[error] Element link=Services not found
[info] Test case failed
[info] Test suite completed: 1 played, 1 failed
```

Without modification of the scenario, the output of the plugin will be

CHECKWEB CRITICAL - clickAndWait link=Services - Execution time = 10.689183s Test Ok 5/8 |'time'=10

Now we can add some additional information in the scenario:

The screenshot shows the Selenium IDE 2.9.0 interface. The main window displays a test scenario named 'scenario01' with a table of commands. The table has three columns: 'Commande', 'Cible', and 'Valeur'. The commands are: 'open' (target: '/centreon/index.php'), 'type' (name: 'useralias', value: 'admin'), 'type' (name: 'password', value: 'centreon'), 'clickAndWait' (name: 'submit'), 'click' (link: 'Monitoring'), 'clickAndWait' (link: 'Services'), 'type' (id: 'host_search', value: 'camera'), and 'clickAndWait' (link: 'All Services'). The 'clickAndWait' command with 'link=Services' is highlighted in red. Below the table, there are input fields for 'Commande', 'Cible', and 'Valeur', with 'open', '/centreon/index.php', and an empty field respectively. The 'Log' window at the bottom shows the following output:

```
[info] Playing test case scenario01
[info] Executing: |open | /centreon/index.php | |
[info] Executing: |type | name=useralias | admin |
[info] Executing: |type | name=password | centreon |
[info] Executing: |clickAndWait | name=submit |
[info] Executing: |click | link=Monitoring | |
[info] Executing: |clickAndWait | link=Services | |
[error] Element link=Services not found
[info] Test case failed
[info] Test suite completed: 1 played, 1 failed
```

In this case, the output of the plugin will be

CHECKWEB CRITICAL - clickAndWait link=Services - Ouverture services - Execution time = 10.689183s Test

Note the extra "Ouverture services" text displayed, that corresponds to the echo command of the scenario