

Monitoring WebSite with Zabbix

Contents

- [1 Introduction](#)
- [2 Adding a web site to the monitoring](#)
- [3 Setting up web site monitoring graphs](#)
- [4 Website monitoring with authorization](#)
- [5 Site unavailability alert](#)
- [6 Conclusion](#)

To monitor the web site, we will use the standard zabbix functionality. Here are the parameters for which we will observe:

- Availability.
- Response time in milliseconds.
- Access speed.
- Work authorization on the site.

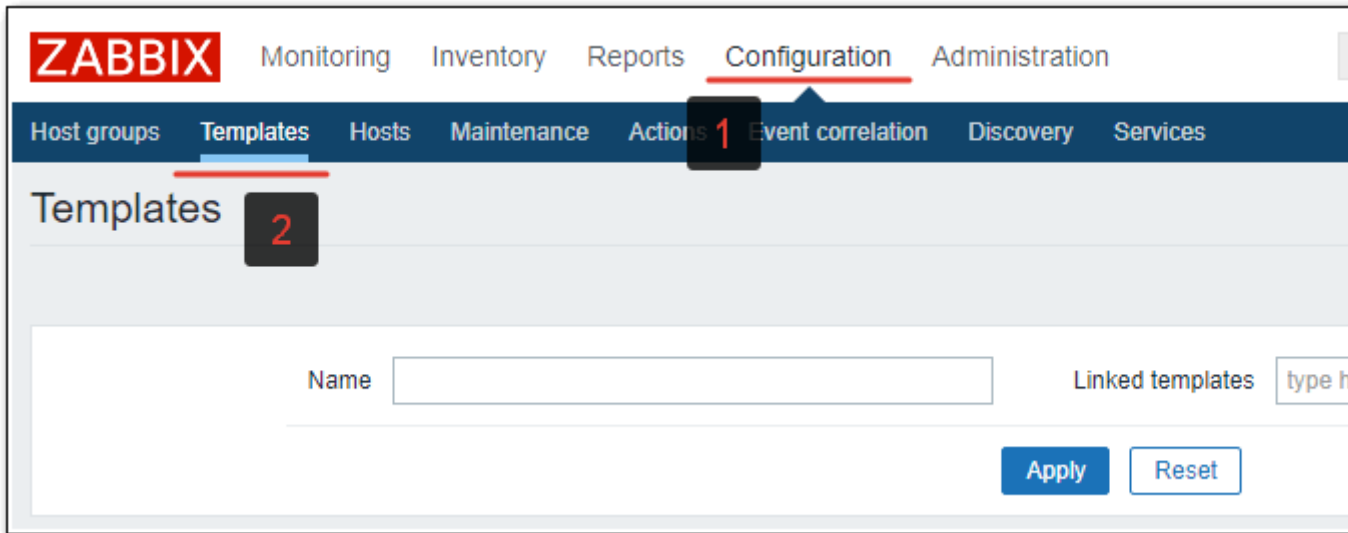
To do this, we will perform the following sequence of actions:

- Create a template for monitoring sites.
- Configure Web scenarios.
- Create graphs.
- Add triggers to check the availability and download speed of the site.

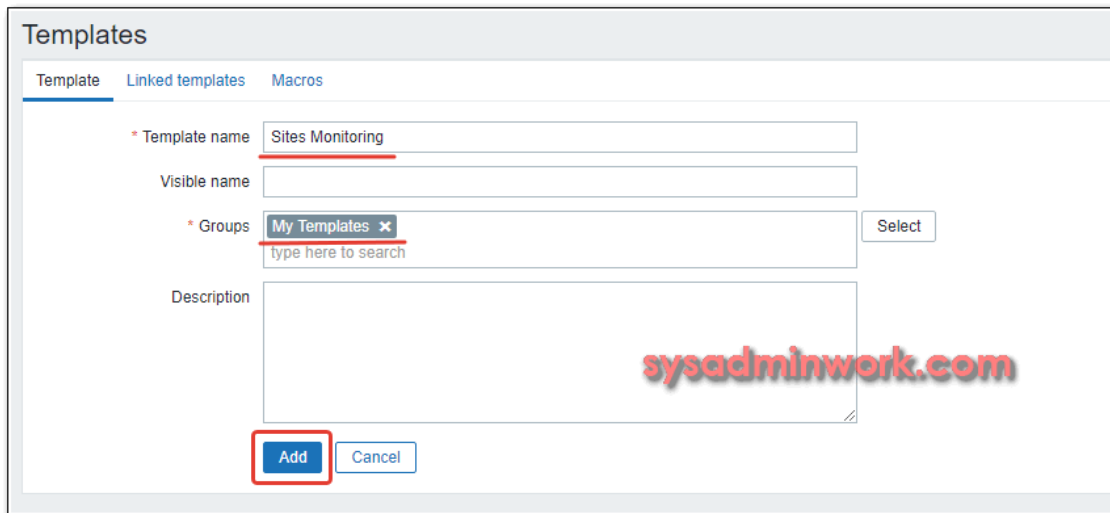
We proceed to setting up monitoring. We will use only the standard functionality available after installation. There will be no additional user parameters or scripts. If you do not have your own monitoring server, I recommend post on this site – [install and configure zabbix](#).

Adding a web site to the monitoring

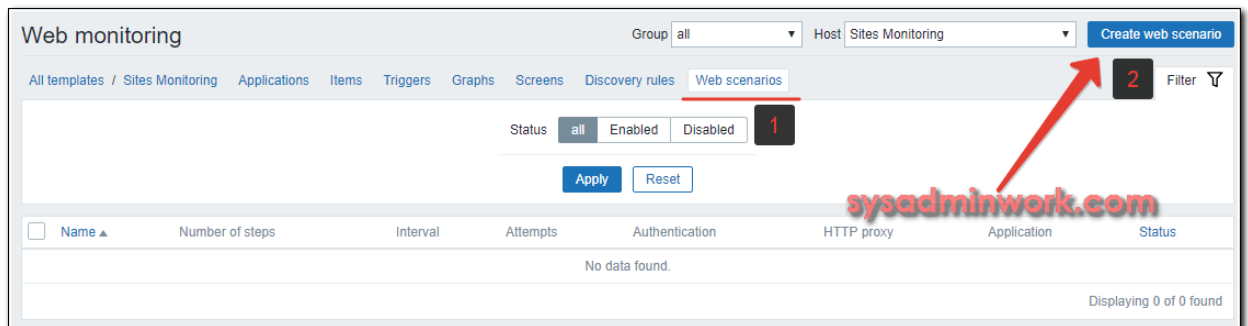
The easiest way to connect a site to monitoring is to add its check on an already existing host. In this case there is one big disadvantage – if you want to enable this monitoring from another host, or simply transfer to another server, then it will be difficult to do so. It is much more easy to monitor sites and everything connected with it, set up in a separate template. So go to the section **Configuration -> Templates** and create a new template.



A standard form for creating a template opens. Enter the name of the template, where the settings for monitoring sites will be, and add it to some group.



Open this template. Go to the **Web Scenarios** tab and add a new web scenario to monitor the site.



Fill in the basic parameters of the script. As a name, I usually specify the address of the site. In my example it will be github.com. Set the name of the application for monitoring sites for easy sorting of items related to the sites, the interval of testing and the number of connection attempts.

Scenario **Steps** Authentication

* Name

Application No applications found.

New application

* Update interval

* Attempts

Agent

HTTP proxy

Variables

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Add](#)

Headers

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Add](#)

Enabled

[Add](#) [Cancel](#)

After that, go to the Steps tab and add the verification step.

Scenario **Steps** Authentication

1 * Steps **2**

[Add](#)

[Add](#) [Cancel](#)

Then I specify the parameters of the step.

Step of web scenario ✕

* Name

* URL

Query fields

Name		Value
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>
		Remove
Add		

Post type

Post fields

Name		Value
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>
		Remove
Add		

Variables

Name		Value
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>
		Remove
Add		

Headers

Name		Value
<input type="text" value="name"/>	⇒	<input type="text" value="value"/>
		Remove
Add		

Follow redirects

Retrieve only headers

* Timeout

Required string

Required status codes

Let me explain each parameter:

- *Name* – the name of the step. In this case, the main page of the site will be checked, therefore I call the step index. This is not important, but I recommend giving names meaningful, so that later it would be convenient to operate with names, for example, in triggers.
- *URL* – the address of the page to be checked.
- *Required string* – the string on the page that zabbix will look for. I took the string from the footer site. If zabbix find it on the page, it will assume that the site is all right. If not, it will be an error.
- *Required status codes* – the required response code. I specify 200. If zabbix receives some other code in response from the web server, the check has failed.

After filling in all the parameters, click Add to add a step and then Add again to add the verification script itself. It should turn out like this.

Web monitoring

Group: all Host: Sites Monitoring [Create web scenario](#)

All templates / Sites Monitoring Applications 1 Items Triggers Graphs Screens Discovery rules Web scenarios 1 Filter

Status: all Enabled Disabled

[Apply](#) [Reset](#)

<input type="checkbox"/>	Name ▲	Number of steps	Interval	Attempts	Authentication	HTTP proxy	Application	Status
<input type="checkbox"/>	github.com	1	1m	3	None	No	Sites Monitoring	Enabled

0 selected [Enable](#) [Disable](#) [Clear history](#) [Delete](#) Displaying 1 of 1 found

The simplest site availability check is done. Next we need to attach this template to some host, so that the actual checks start. I will attach the template to the zabbix server itself. To do this, go to **Configuration -> Hosts**, select Zabbix Server and attach the template created earlier to it.

Hosts

All hosts / Zabbix server Enabled ZBX SNMP JMX IPMI Applications 15 Items 84 Triggers 54 Graphs 13 Discovery rules 5 Web scenarios 1

Host Templates IPMI Macros Host inventory Encryption

Linked templates

Name	Action
Domain Expiration	Unlink Unlink and clear
Remote RDP	Unlink Unlink and clear
<u>Sites Monitoring</u>	Unlink
SSH Auth RPM	Unlink Unlink and clear
SSL Cert Expiration	Unlink Unlink and clear
Template App Zabbix Server	Unlink Unlink and clear
Template OS Linux	Unlink Unlink and clear

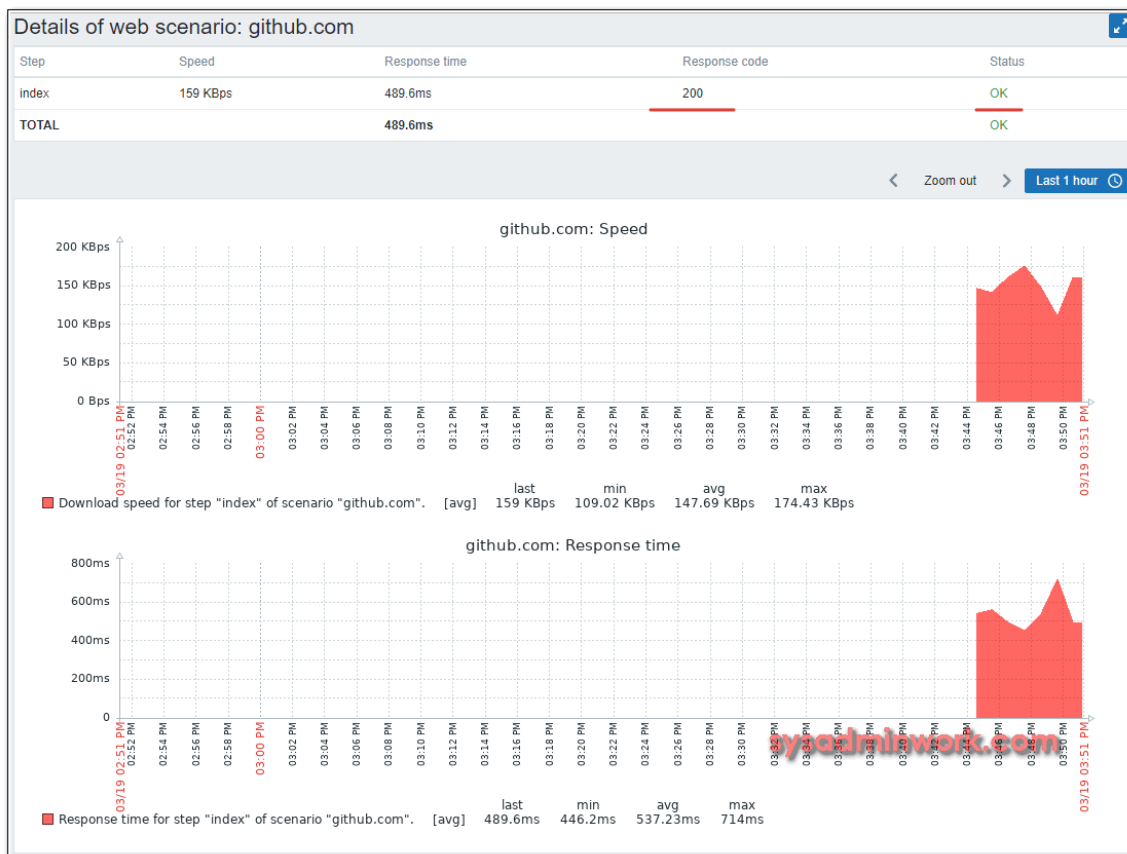
Link new templates

type here to search [Select](#)

[Add](#)

[Update](#) [Clone](#) [Full clone](#) [Delete](#) [Cancel](#)

We wait a few minutes and go to the section **Monitoring -> Web** to watch the results of monitoring the site github.com.



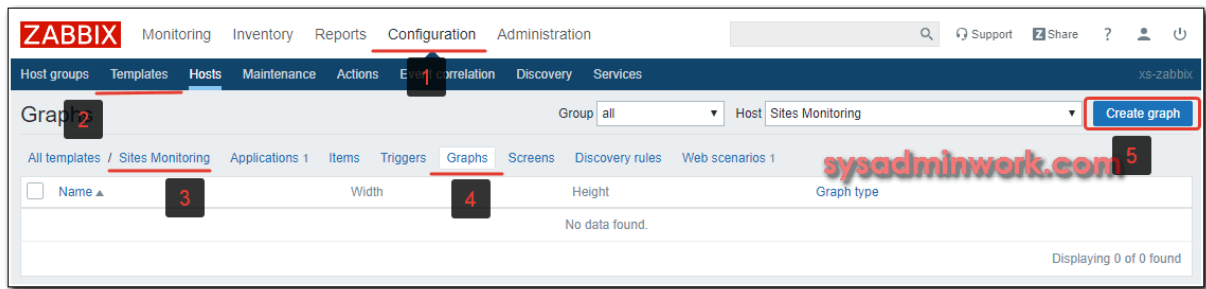
The response code 200, the search string was found, which confirms Status OK. Here is graphics site download speed and response time. For more information about monitoring the specified site can be found in **Latest Data**.

Host groups	Name	Last check	Last value	Change
Zabbix server	Sites Monitoring (6 items)			
<input type="checkbox"/>	Download speed for scenario "github.com".	03/19/2019 04:10:52 PM	115.91 KBps	+1.82 KBps Graph
<input type="checkbox"/>	Download speed for step "index" of scenario "github.com".	03/19/2019 04:10:52 PM	115.91 KBps	+1.82 KBps Graph
<input type="checkbox"/>	Failed step of scenario "github.com".	03/19/2019 04:10:52 PM	0	Graph
<input type="checkbox"/>	Last error message of scenario "github.com".			History
<input type="checkbox"/>	Response code for step "index" of scenario "github.com".	03/19/2019 04:10:52 PM	200	Graph
<input type="checkbox"/>	Response time for step "index" of scenario "github.com".	03/19/2019 04:10:52 PM	671.6ms	-10ms Graph

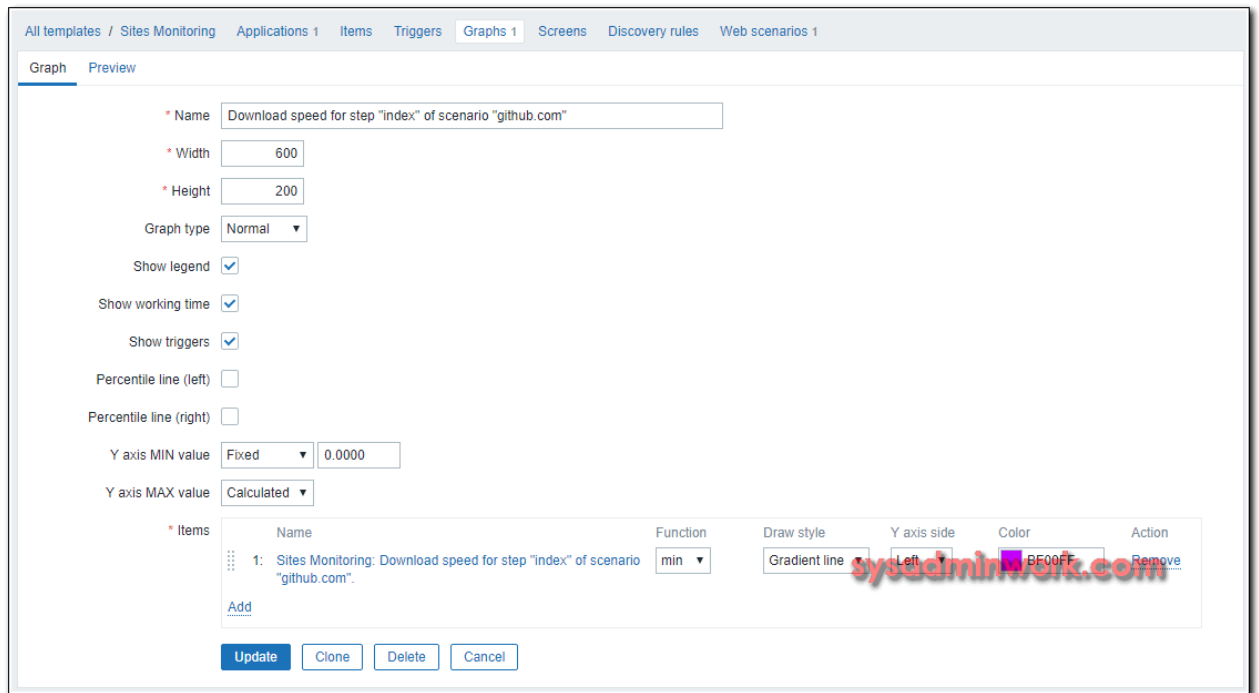
The value of the parameter *Failed step of scenario "github.com"* equal to 0 means that all steps to check the site were completed without errors. If you have several steps and one of them is finished with an error, there will be a number of this step. That is, in general, all that is not 0, these are some problems. Later we will use it in the trigger. In the meantime, add a couple of graphs to the template, which can then be used in dashboards.

Setting up web site monitoring graphs

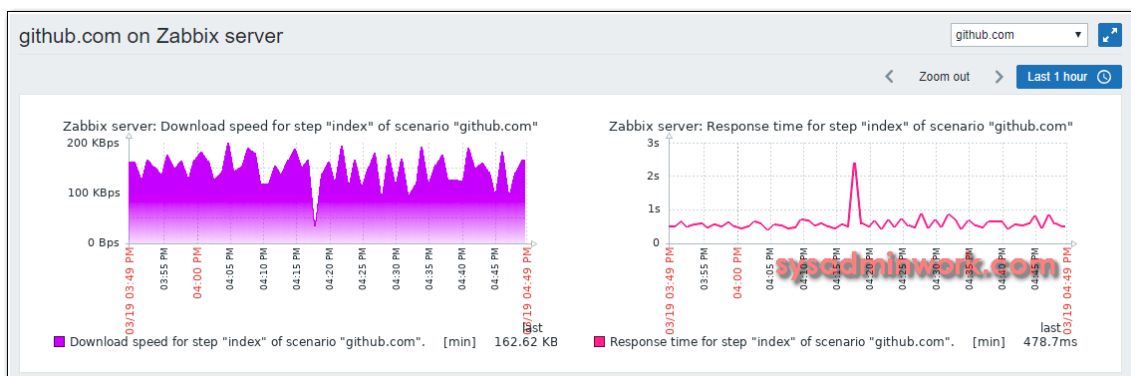
We return to our template and go to the Graphs section. Create a new graph.



Add a graph of the download speed for the main page of the site.



Similarly, you can add a site response time. I once added both of these graphics to Screen. It should turn out like this.



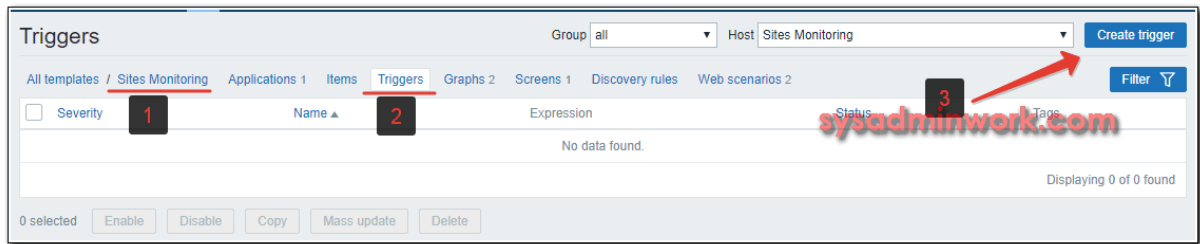
For more usefull visualizations, it is better to use **Dashboards**.

Site unavailability alert

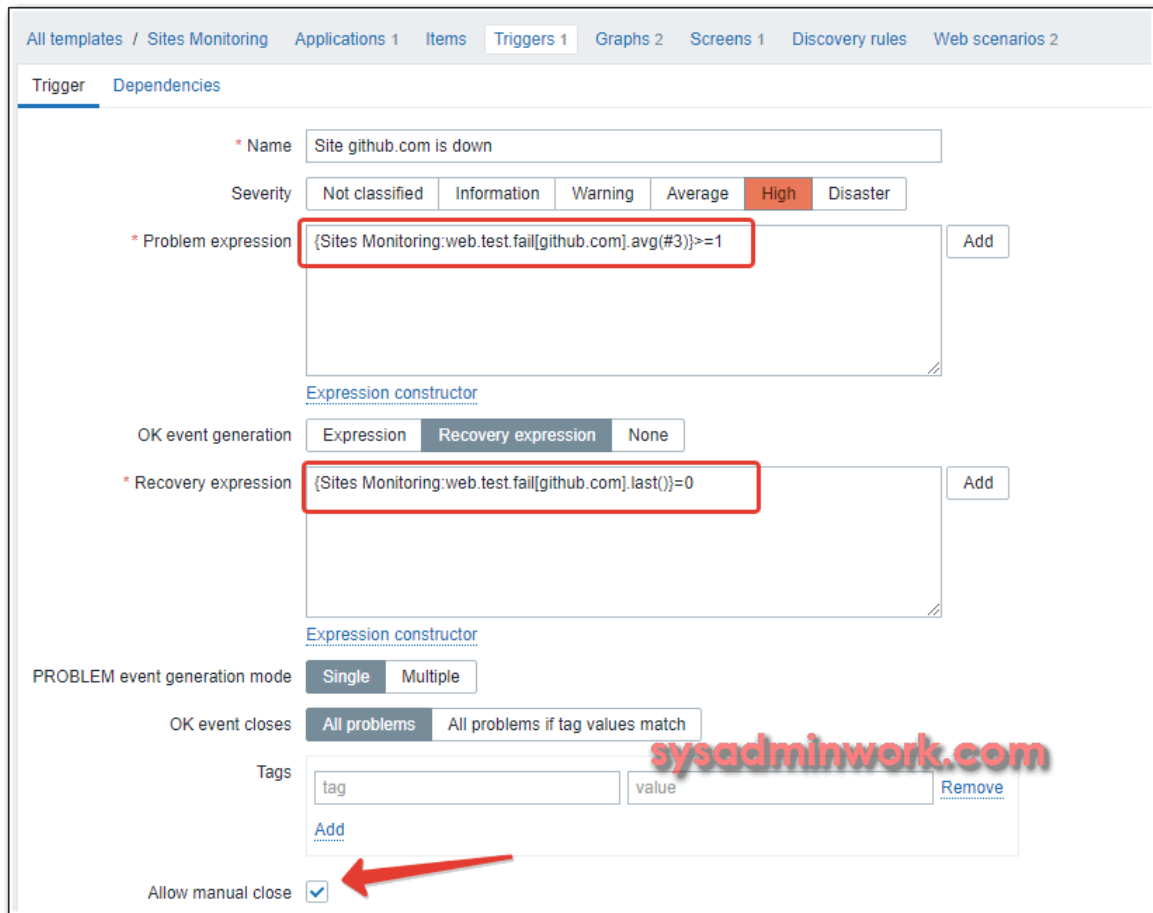
Let's set up notifications about problems on the site. I offer 2 types of alerts:

- Low site access speed.
- The site availability.

We go, as usual in the original template, on the Triggers tab and add a new one.



I propose such a trigger condition to determine the inaccessibility of the site. If the average value of the last 3 checks is greater than or equal to one, then the site unavailability warning is triggered.



When 0 goes in all checks, everything is in order. The trigger will work only if all 3 last checks are not zero. In my example, Failed step can be either 0 or 1, where 1 is the number of the failed step. If you have several steps, then the second step or the third step may be a failure. That is, the value can be greater than 1. But in any case, if the last 3 values in a row are strictly not 0, then the trigger is triggered. The recovery operation is very simple. If the last check without error, that is, the code is 0, then we consider that the site is already working.

To test the trigger, it is enough to add a line to the /etc/hosts file on the zabbix server:

```
127.0.0.1 github.com
```


and wait 3 minutes to get 3 unsuccessful checks. After that, you should have been sent a notification about the inaccessibility of the site. I got this:



Next, we check the server response time. Here everyone is free to tune in as it seems to him more correct and convenient. I use such a scheme. I take the average response time of the site and multiply it by 3. Then I look at the last 7 checks. If in 5 checks among these seven there were values higher than the tripled average response time, then I consider that the site slows down and it is necessary to send a notification. It is a bit confusing, but in practice such a scheme I have recommended myself well without false positives. However, if there are real problems, I see them. We make the trigger.

Trigger Dependencies

* Name

Severity

* Problem expression

[Expression constructor](#)

OK event generation

* Recovery expression

[Expression constructor](#)

PROBLEM event generation mode

OK event closes

Tags

[Add](#)

Allow manual close

URL

Description

Enabled

Condition recovery – in the last three requests, two or more were faster than the tripled average access time. Expression text to copy:

```
{Sites Monitoring:web.test.time[github.com,index,resp].count(#7,1.5,\"ge\")}>4
{Sites Monitoring:web.test.time[github.com,index,resp].count(#3,1.5,\"lt\")}>1
```

Here 1.5 is the response time in seconds. You can check it in **Latest Data**.

The screenshot shows the Zabbix web interface. The top navigation bar includes 'Monitoring', 'Inventory', 'Reports', 'Configuration', and 'Administration'. Below it, a secondary navigation bar has 'Dashboard', 'Problems', 'Overview', 'Web', 'Latest data', 'Graphs', 'Screens', 'Maps', 'Discovery', and 'Services'. The main content area displays a table titled 'Zabbix server: Response time for step "index" of scenario "github.com"'. The table has two columns: 'Timestamp' and 'Value'. The 'Value' column is highlighted with a red box. The data shows response times ranging from 0.5445 to 0.5633 seconds. A watermark 'sysadminwork.com' is visible on the right side of the table.

Timestamp	Value
03/19/2019 09:00:51 PM	0.5445
03/19/2019 08:59:51 PM	0.5874
03/19/2019 08:58:50 PM	0.5936
03/19/2019 08:57:49 PM	0.5614
03/19/2019 08:56:48 PM	0.9305
03/19/2019 08:55:47 PM	0.5633

In conclusion, I leave my template that I created for writing the article. You can copy and edit to adapt it for your sites. It is faster than being from scratch. Template exported from zabbix version 4.0 – [sites_monitoring.xml](#)

That's all, web site monitoring works, authorization is checked, the site unavailability notification is configured. For completeness, you can create a Screen or Dashboard with the output of all the necessary parameters on one screen. Its settings will already depend on the specific situation and the data that you have. For example, if you have configured web server monitoring, then you can place a number of graphs of its download and parameters of access to the site. There you can also add the load of the server itself on the processor and memory and display a graph of the use of the network interface.

Zabbix is very flexible and allows you to customize everything for every taste and under any requirements.