

Monitoring

Note: Any specific Tiki-Nagios documentation should be moved to <https://doc.tiki.org/Nagios>

Tiki Monitoring with Nagios

Assumptions

This text will assume you have enough knowledge and domain of Nagios and Linux, your Nagios is already deployed and it is in an external server (different where you have your Tiki deployment)

Nagios Configuration

First you need to copy the check_tiki scripts. You can download it [here](#). This script is written in PHP. Put it on a path you will remember, for example /usr/local/bin. You will need to call it manually to see the options. This document will use some defaults ones, but you will need to put your own values.

Look for your nagios.cfg file, open it and check what directory is pointing the variable cfg_dir. In this case we are going to use /etc/nagios/conf.d/

Create a file at /etc/nagios/conf.d/check_tiki.cfg with the following content:

```
define command{ command_name check_tiki_bcc command_line /usr/local/bin/check_tiki.php -u $ARG1$ -c bcc --bccwarn $ARG2$ --bcccrit $ARG3$ } define command{ command_name check_tiki_searchindex command_line /usr/local/bin/check_tiki.php -u $ARG1$ -c searchindex --sirwarn $ARG2$ --sircrit $ARG3$ } define command{ command_name check_tiki_db command_line /usr/local/bin/check_tiki.php -u $ARG1$ -c db }
```

Edit your server definition. Put the following in the proper directory. In our case, it is in the /etc/nagios/objects/servers/ directory. This is a template, your server definition may be much more complex than this.

```
define host{ use linux-server host_name trunk.notre.website alias trunk } define service{ use local-service host_name trunk.notre.website service_description HTTP check_command check_http!-H trunk.notre.website notifications_enabled 1 } define service{ use local-service host_name trunk.notre.website service_description TIKI Op Code Cache check_command check_tiki_bcc!http://trunk.notre.website/tiki-monitor.php!80!95 notifications_enabled 1 } define service{ use local-service host_name trunk.notre.website service_description TIKI Index check_command check_tiki_searchindex!http://trunk.notre.website/tiki-monitor.php!80000!95000 notifications_enabled 1 } define service{ use local-service host_name trunk.notre.website service_description TIKI Op Code Cache check_command check_tiki_db!http://trunk.notre.website/tiki-monitor.php notifications_enabled 1 }
```

As optional setup you can put your service dependencies. In this specific case, check_tiki_* depends on HTTP availability.

```
define servicedependency{ host_name trunk.notre.website service_description HTTP dependent_host_name trunk.notre.website dependent_service_description TIKI Op Code Cache execution_failure_criteria n notification_failure_criteria w,u,c } define servicedependency{ host_name trunk.notre.website service_description HTTP dependent_host_name trunk.notre.website dependent_service_description TIKI Index execution_failure_criteria n notification_failure_criteria w,u,c } define servicedependency{ host_name trunk.notre.website service_description HTTP dependent_host_name trunk.notre.website dependent_service_description TIKI DB execution_failure_criteria n notification_failure_criteria w,u,c }
```

WikiSuite: The most comprehensive and integrated Open Source enterprise solution.

You can check your configuration by executing:

```
nagios -v /etc/nagios/nagios.cfg
```

If there is an error, you will be notified. After that, you can restart nagios and new monitoring will start.