

# Monitoring Tiki

Tiki's tiki-monitor.php script now supports enhanced options to facilitate integration with monitoring systems like Nagios, Zabbix, and Icinga. These enhancements include IP restrictions, token-based authentication, role-based data visibility, custom probes, and support for multiple HTTP methods with custom error codes.

## Nagios

- Ensure all necessary services/daemons are running. Use `check_procs` or `check_ssh` (to call `check_procs` on remote server)
  - Apache
  - MySQL (also use `check_mysql` to run test queries)
- Ensure adequate memory, cpu and disk space are available using `check_mem`, `check_load` and `check_disk`
- Ensure web site is healthy.
  - Use `check_http` to check a particular web page or
  - Use `check_webinject` to script a login and perform a more detailed site check
- Track site performance
  - Use `pnp4nagios` or `nagiosgraph` to capture the performance data from `check_http` or `check_webinject` to track response time data over time.

## Tiki Monitoring Script

The primary monitoring script, `tiki-monitor.php`, now supports multiple options to integrate seamlessly with external monitoring systems like Nagios, Zabbix, and Icinga.

### • IP Restriction:

- Define IP addresses allowed to access the monitoring script. Set the `monitoring_restricted_ips` preference with a comma-separated list of IPs, ensuring only specified addresses can initiate requests. This setting complements the existing `$tikiMonitorRestriction` option for backward compatibility.
- Example Configuration:

---

```
127.0.0.1,192.168.1.10
```

### • Token-Based Authentication:

- Set a `monitoring_token` preference to authenticate requests via a secure token. The token can be passed as:
  - A GET or POST parameter with the name `monitoring_token`
  - An HTTP header `X-Tiki-Monitoring-Token`
- Requests with an invalid token return HTTP 401, preventing unauthorized access.
- Token Usage Examples:
  - GET Parameter:

---

```
curl --location '<domain>/tiki-monitor.php?monitoring_token=your_token_here'
```

- POST Parameter:

```
curl --location '<domain>/tiki-monitor.php' --form  
'monitoring_token=your_token_here'
```

- HTTP header:

```
curl --location --head '<domain>/tiki-monitor.php' --header 'X-Tiki-Monitoring-Token:  
your_token_here'
```

### • Role-Based Data Visibility:

- Configure data visibility with the `monitoring_auth` preference to control which data is accessible to authenticated vs. public users.
- Define rules such as `OPCodeCache:auth` or `*:auth` to specify visibility at the key level in JSON output, allowing fine-grained control over what each role can see.
- **Two roles are supported:**
  - `public`: Default role with access to limited data.
  - `auth`: Role for authenticated users (admin or valid token). Users with `auth` can see both public and restricted data based on rule configuration.
  - Example Rule:

```
OPCodeCache:auth SearchIndexRebuildLast:public OpCodeStats.opcode_cache:public  
OpCodeStats:auth *:auth
```

- With these rules, public users would only see selected keys, while authenticated users would have broader access. If no custom rules are provided, defaults apply to retain backward compatibility.

### • Custom Probes:

- Use the `monitoring_probes` preference to define specific checks (probes) using Tiki's calculation syntax.
- Examples of probes:
  - Basic Comparison:

```
(less-than SearchIndexRebuildLast 10)
```

- Timestamp Comparison:

```
(less-than SearchIndexRebuildLast NOW)
```

- Equality Check:

```
(equals OpCodeStats.opcode_cache "OpCache")
```

- Sample JSON Output with Probes:

```
{ "Probes": { "result": "OK", "details": { "probe_1": "OK", "probe_2": "OK" } } }
```

- If probes fail, the result field will be set to FAIL, and each probe's status will be noted.

### • HTTP Methods and Custom Error Codes:

- The script supports HEAD, GET, and POST methods:
  - `HEAD`: Returns no body, useful for health checks.
  - `GET` and `POST`: Standard responses based on request.
- The parameter `monitoring_error_code` (passed as a GET/POST parameter or `X-Tiki-Monitoring-Error-Code` header) specifies a custom HTTP code (200-599) to be returned on probe failure.
- Examples of Custom Error Codes:
  - GET Request:

```
curl --location '<domain>/tiki-monitor.php?monitoring_error_code=400'
```

- POST Request:

```
curl --location '<domain>/tiki-monitor.php' --form 'monitoring_error_code=400'
```

- HEAD Request:

```
curl --location --head '<domain>/tiki-monitor.php' --header 'X-Tiki-Monitoring-Error-Code: 400'
```

## Configuration

- Configure the preferences within `tiki-admin.php?page=performance`
- Set IP restrictions with `monitoring_restricted_ips`.
- Enable token-based requests with `monitoring_token`.
- Define visibility rules with `monitor_rules`.
- Add probes in `monitor_probes`.

## Additional Monitoring Scripts

In addition to `tiki-monitor.php`, Tiki includes other monitoring scripts for system health checks:

- `check_tiki.php` – A basic health-check script.
- `check_tiki-new.php` – An enhanced version of `check_tiki.php` with extended checks.

## Code in Tiki for Zabbix, Nagios, Icinga, Shinken, etc.

- <https://gitlab.com/tikiwiki/tiki/-/blob/master/tiki-monitor.php>
- [https://gitlab.com/tikiwiki/tiki/-/blob/master/doc/devtools/check\\_tiki.php](https://gitlab.com/tikiwiki/tiki/-/blob/master/doc/devtools/check_tiki.php)
- [https://gitlab.com/tikiwiki/tiki/-/blob/master/doc/devtools/check\\_tiki-new.php](https://gitlab.com/tikiwiki/tiki/-/blob/master/doc/devtools/check_tiki-new.php)