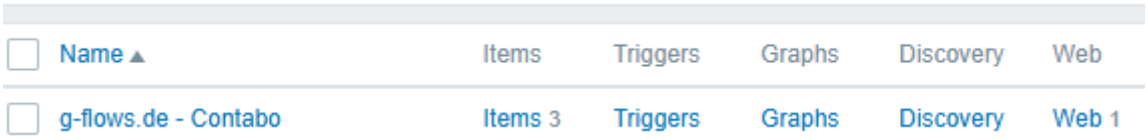
Web-Überwachung mit Web Scenarios in Zabbix 7.0+

Anstatt HTTP-Agent oder externe Skripte zu nutzen, richten wir ein Web Scenario ein, das:

- ✔ Den HTTP-Statuscode überwacht (200 OK oder Fehlercodes 403, 500 usw.)
- ✔ Die Ladezeit (Response time) misst
- ✔ Die Download-Geschwindigkeit (Download speed) speichert
- ✔ Automatisch ausgeführt wird, ohne manuelle Anpassungen

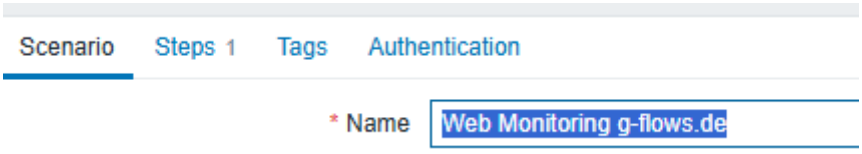


1️⃣ Web Scenario für "https://g-flows.de" erstellen

- Gehe zu Data collection → Hosts
- Wähle host (z.B. g-flows.de)
- Gehe zu Spalte Web → Create Web Scenario
  - Fülle folgende Werte aus:
  - Name: Web Monitoring g-flows.de
  - Update interval: 30s (Wie oft das Szenario laufen soll)
  - Retries/Attempts: 3 (Falls die Website nicht erreichbar ist, wird es erneut versucht)
  - Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) (Damit sich Zabbix wie ein echter Browser verhält) oder Chrome in neuester Variante
  - Variables: (Leer lassen)
  - Headers: (Leer lassen)
  - Required status codes: 200 (Damit Zabbix erwartet, dass die Seite mit HTTP 200 OK antwortet)
- 💾 Speichern!

2️⃣ Schritte für das Web-Szenario (Web Steps) hinzufügen

Öffne das gerade erstellte Web Monitoring g-flows.de



- Gehe zu Steps → Add
- Fülle folgende Werte aus:
- Name: Check Homepage
- URL: https://g-flows.de
- Timeout: 10s
- Required string: (Leer lassen oder Welcome falls ein spezifischer Text auf der Seite vorhanden ist)
- Required status codes: 200
- Retrieve mode: Body (Falls du nur den Body checken willst)
- 💾 Speichern!

3️⃣ Automatisch erfasste Web Monitoring Items

Sobald das Web Scenario gespeichert ist, erstellt Zabbix automatisch folgende Überwachungswerte:

□□ Allgemeine Werte für das gesamte Web-Szenario

- web.test.fail[g-flows.de,Web Monitoring g-flows.de] → Gibt die Nummer des fehlgeschlagenen Schritts zurück (falls Fehler)
- web.test.error[g-flows.de,Web Monitoring g-flows.de] → Letzte Fehlermeldung

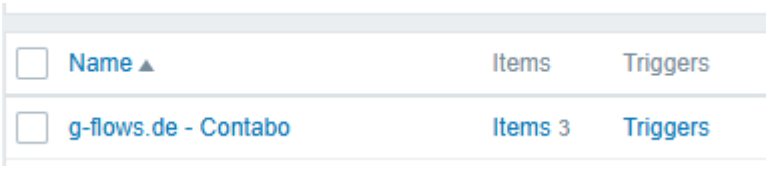
□□ Werte für den Schritt "Check Homepage"

- web.test.time[g-flows.de,Web Monitoring g-flows.de,Check Homepage] → Ladezeit der Seite (Response time)
- web.test.rspcode[g-flows.de,Web Monitoring g-flows.de,Check Homepage] → HTTP-Statuscode (z. B. 200, 404, 500)
- web.test.in[g-flows.de,Web Monitoring g-flows.de,Check Homepage] → Eingehende Download-Geschwindigkeit

□□ Diese Werte sind jetzt automatisch verfügbar!

4□ Trigger für Alarme einrichten

4.1 Alarm: Webserver antwortet nicht (200 OK fehlt)



Gehe zu Triggers → Create trigger

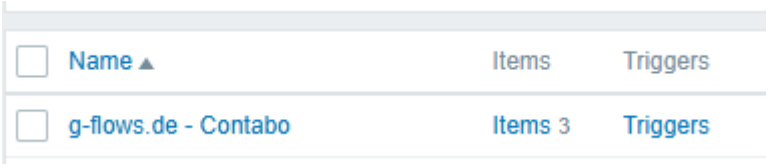
Fülle folgende Werte aus:

- Name: Webserver g-flows.de nicht erreichbar
- Severity: High
- Expression:blast(/g-flows.de/web.test.rspcode[g-flows.de,Web Monitoring g-flows.de,Check Homepage])<>200

□□ Speichern!

4.2 Alarm: Webserver lädt zu langsam (> 2s)

Gehe zu Triggers → Create trigger



Fülle folgende Werte aus:

Name: Webserver g-flows.de lädt zu langsam

Severity: Warning

Expression: last(/g-flows.de/web.test.time[g-flows.de,Web Monitoring g-flows.de,Check Homepage])>2

□□ Speichern!

Fazit – Warum ist Web Monitoring die beste Lösung?

- ✔ Web Scenarios sind speziell für Webseiten-Monitoring optimiert
- ✔ Kein kompliziertes Preprocessing oder externe Skripte nötig!
- ✔ Erfasst HTTP-Status, Ladezeit und Download-Speed automatisch
- ✔ Funktioniert auch mit Weiterleitungen und verschlüsselten HTTPS-Seiten
- ✔ Trigger für langsame Ladezeiten oder Fehler sind einfach konfigurierbar
- ✔ Mehrere Webseiten-Checks in einem einzigen Web Scenario möglich

➡ Fazit: Diese Methode ist die beste & einfachste Lösung für dein Zabbix Web Monitoring!

Revision #1
Created 2025-02-26 18:40:33 UTC by Gerd
Updated 2025-02-26 19:01:21 UTC by Gerd