



Eher unbekannt, doch leistungsstark: Zabbix



Die Featureliste liest sich wie das Ergebnis einer Kreuzung der bekanntesten Open-Source-Monitore in einem unbekanntem Produkt. Denn was viele Konkurrenten erst mit Unterstützung durch Third-Party-Software schaffen, das bringt Zabbix gleich mit.

Daniel Dörrer, Bogdan Taru, Markus Feilner

Bis vor einigen Monaten war Zabbix (1) nur Eingeweihten ein Begriff. In letzter Zeit aber gewinnt das sechs Jahre alte Open-Source-Überwachungstool an Popularität, ist es doch angestiegen, dem Platzhirsch Nagios das Revier streitig zu machen.

Dabei gibt seit zwei Jahren die Firma SIA Zabbix mit Sitz in der lettischen Hauptstadt Riga Rückendeckung. Sie bietet professionellem Support und Unterstützung bei Entwicklung und Test. Ob das Produkt dadurch tatsächlich schon dem Anspruch gerecht wird, die besten Features der Konkurrenz zu vereinen, zeigt der Test auf den folgenden Seiten.

Informationsbeschaffung

Bereits beim Sammeln von Informationen, der Grundfunktion jeder Überwachungssoftware, gibt sich Zabbix kaum eine Blöße. Es beherrscht alle gängigen Methoden: aktive und passive, solche, die sich auf Netzwerkprotokolle stützen, und natürlich auch SNMP. (Mit dem SNMP-Protokoll beschäftigt sich ein weiterer Artikel dieser Ausgabe.)

Direkt nach der Installation stehen bereits rund 100 eingebaute Tests zur Verfügung, mit denen sich grundlegende Funktionen und Ressourcen wie die CPU-Last oder der Speicherverbrauch beobachten lassen. Für viele dieser Tests kommt ein so genannter Agent zum Einsatz, der entweder auf dem zentralen Server oder dem überwachten Client als Informationsbeschaffer tätig ist. In manchen Fällen sammelt er permanent Informationen, gibt sie aber nicht ungefragt

weiter. In anderen ist er auch selbstständig tätig und kann etwa aus eigener Initiative eine Logdatei regelmäßig nach bestimmten Mustern durchforsten.

Dieser Agent fühlt sich auf vielen Unix- und Windows-Plattformen heimisch, so unter Linux, Solaris, HP-UX, AIX, FreeBSD, OpenBSD, MacOS X, Tru64/OSF1, Windows NT4.0, Windows 2000, Windows 2003 oder Windows XP.

Sollte der Agent für eine bestimmte Überwachungsaufgabe noch nicht geschult sein, kann man ihm leicht mit einem eigenen Perl- oder Shellskript oder einem C-Modul auf die Sprünge helfen. Zukünftige Zabbix-Versionen sollen sogar mit einer Diensterkennung ausgerüstet sein, mit deren Hilfe er automatisch die wichtigsten Services ins Auge fassen könnte.

Geräte im Netz, die SNMP sprechen, kann Zabbix direkt befragen, denn es beherrscht das Esperanto der Devices ebenfalls. Daneben kann die Software zahlreiche Netzwerkdienste testen, indem sie sich als zugehöriger Client ausgibt und eine Konversation beginnt. Die Antwort des Gegenübers verrät in der Regel viel über seinen Gesundheitszustand. Solche Checks sind – wie die SNMP-Abfragen – natürlich nicht auf das LAN beschränkt.

Zentrallager für Daten

Alle Beobachtungswerte gelangen in eine zentrale Datenbank, von wo aus sie zum Beispiel in vielfältig konfigurierbare Diagramme einfließen. Möglich sind Kurzzeitdarstellungen und Statusübersichten ebenso wie Langzeitstatistiken, wie

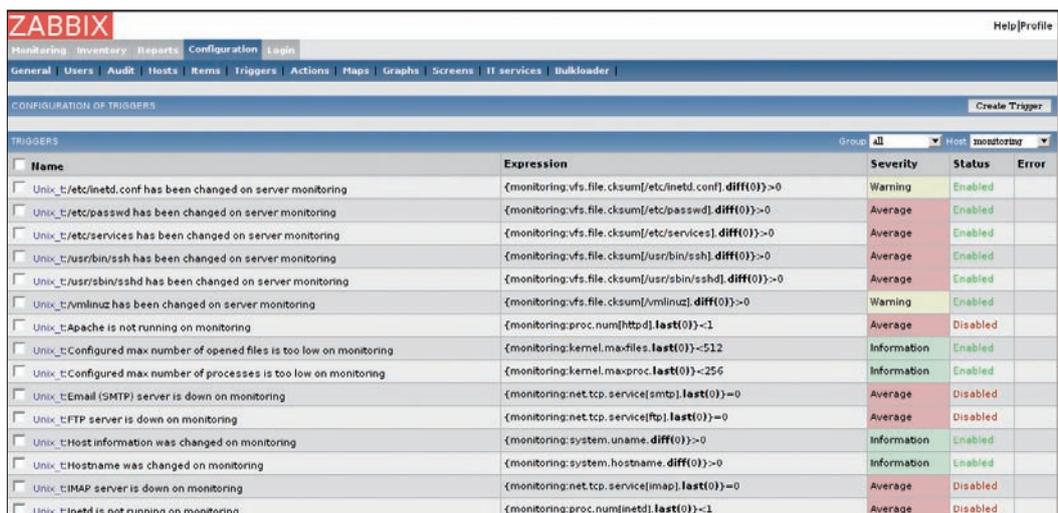


Abbildung 1: Ein übersichtliches Webfrontend bildet die Zabbix-Zentrale, zuständig für Konfiguration und Kommunikation mit dem Anwender. Im Bild: die Liste der überwachten Parameter.

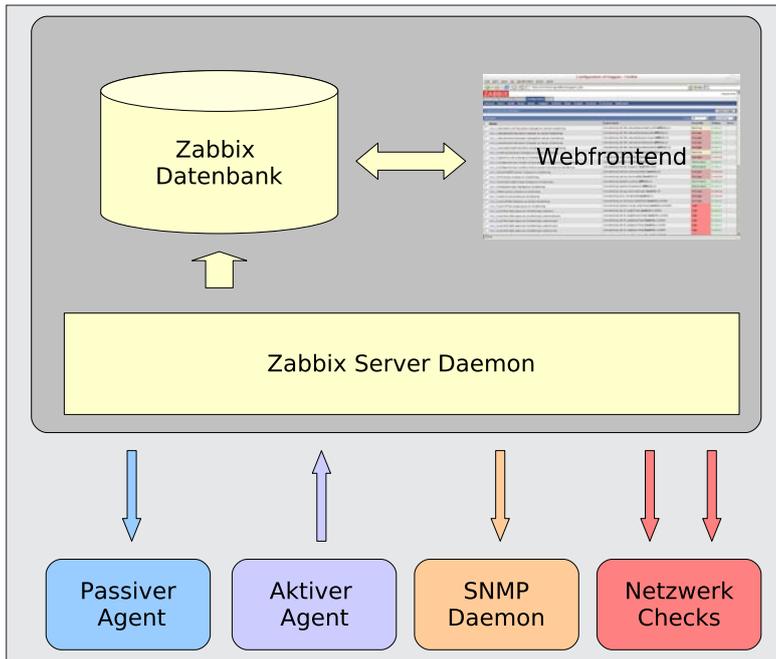


Abbildung 2: Zabbix sammelt Informationen über verschiedene Kanäle, neben seinen Agenten verwendet es SNMP oder zapft Netzwerkprotokolle an.

man sie beispielsweise für die Kapazitätsplanung benötigt.

Eine besondere Spezialität von Zabbix sind Diagramme, in denen sich die Darstellung mehrerer Werte (Items) überlagert. Die **Abbildung 3** zeigt beispielhaft eine Grafik mit dem aktuellen Verlauf der Prozessorauslastung während der letzten Stunde in den Kategorien idle, user, nice und system.

Der Dialog »Configuration | Graphs | Create Graph« ist der Ausgangspunkt beim Erstellen einer solchen Grafik. Über seine Schaltfläche »Edit« lassen sich beliebige Items aus einem Drop-down-Menü wählen.

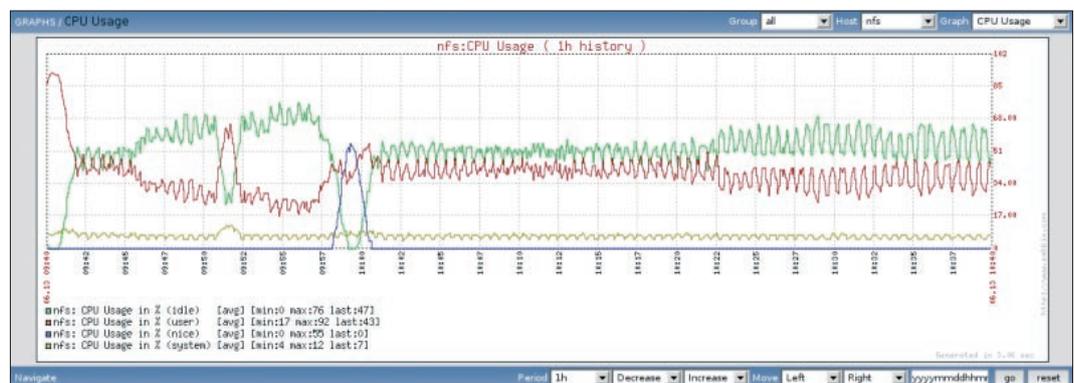


Abbildung 3: Verschiedene Zeitreihen in einem Diagramm sind eine Spezialität von Zabbix.

Kartierte Informationen

Zabbix kann darüber hinaus Statusinformationen mit Karten (Maps) verknüpfen, die die Infrastruktur entweder physisch oder logisch abbilden. Diese Karten lassen sich zudem so verschachteln, dass man von einer Übersichts-darstellung in immer detailliertere Ansichten hinein absteigen kann. Eine Karte als Ausgangspunkt beschleunigt erfahrungsgemäß die Fehlersuche, weil sie dem Administrator einen schnellen Überblick verschafft. Dabei lässt sich der Zugriff auf bestimmte Zoom-Stufen oder Bereiche an die Berechtigungen bestimmter Nutzer knüpfen.

Die Karten lassen sich komplett über das Frontend im Browser erstellen. Alles nötige findet sich im Dialog »Configuration | Maps | Create Map«. Hier legt der Administrator eine neue Karte an, benennt sie und platziert anschließend via »Add Element« beliebige von Zabbix überwachte Objekte. Auch die Verbindungen zwischen den Hosts sind sowohl darstell- wie auch überwachbar.

Templates

Beim schnellen Integrieren neuer Hosts in das Überwachungssystem helfen Vorlagen dem Administrator. Diese Templates für Web-, Application-, Datenbank- oder Mailserver oder auch Desktoprechner legen fest, welche Dienste oder Systemparameter zu überwachen, welche Checks dafür auszuführen sind und in welchen Grafiken Zabbix die Werte darstellt. Auch für ganze Hostgruppen stehen Templates zur Verfügung, so dass Administratoren Serverfarmen oder Cluster mit wenigen Mausklicks hinzufügen und in gemeinsamen Grafiken überwachen können.

Dashboards für jeden

Ein besonders interessantes Feature von Zabbix stellen die so genannten Screens dar. Das sind Ansichten, in denen sich beliebige Tabellen, Listen, Karten oder Diagramme auf einer Bildschirmseite kombinieren lassen. Über »Configuration | Screens | Create Screen« kann der Administrator die gewünschten Beobachtungspunkte für eine auf seine Bedürfnisse zugeschnittene Ansicht auswählen.

Zabbix beinhaltet wie jede Software seiner Art Algorithmen, die die Beobachtungsergebnisse mit bestimmten Grenzwerten vergleichen, die auch in Service Level Agreements (SLAs) festgeschrieben sein können. Bei einer Verletzung dieser Limits, die sich auf Fehlerraten, Ressourcenauslastung oder auch Performancekennwerte beziehen können, löst Zabbix über einen Triggermechanismus beliebige Aktionen aus. Auf diese Weise lässt sich sowohl die Alarmierung steuern wie auch jede andere Aktion starten, die beispielsweise zur Schadensbegrenzung oder Selbstheilung sinnvoll erscheinen mag. Zusätzlich kann der Admin Zeitfenster festlegen, in denen ein bestimmter Trigger reagieren oder pausieren soll.

Die Standardkonfiguration überwacht bereits direkt nach der Installation so viele Items, dass es oft sinnvoll erscheint, zuerst die nicht benötigten zu deaktivieren (unter »Configuration | Items«). Dasselbe lässt sich auch schon auf der Ebene der Vorlagen einstellen. Dazu wählt man als Gruppe »Templates« und als Host das entsprechende Template.

Zabbix-Installation

Die Installation des Zabbix-Servers und seiner Agenten stellt den Administrator vor keine großen Probleme. Allerdings muss er dafür überwiegend auf der Kommandozeile operieren, wogegen sich die folgende Konfiguration dann fast vollständig in der Browser-GUI erledigen lässt. Im Beispiel kam ein Ubuntu Server LTS 6.0.6 (installiert mit LAMP-Option) zum Einsatz, bei einem weiteren Agenten ein Red Hat Enterprise Linux 4.

Damit die Installation gelingt, sollte allerdings eine Reihe von Voraussetzungen erfüllt sein (siehe **Tabelle 1**).

Außerdem empfiehlt sich das Anlegen eines Accounts für den Benutzer »zabbix«. Und schließlich sollte auch die Datenbank fertig und arbeitsfähig eingerichtet sein, bevor man die

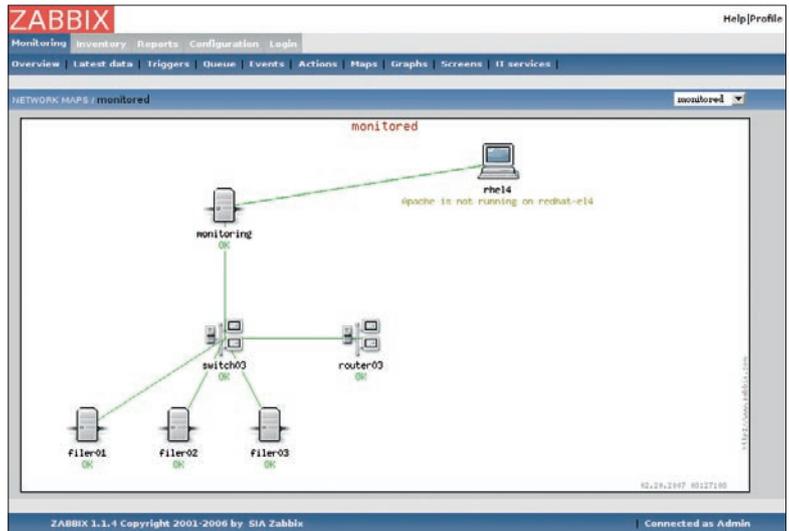


Abbildung 4: Netzwerkdarstellungen, die den Status der überwachten Geräte reflektieren, lassen sich direkt in Zabbix entwerfen.

Monitoring-Lösung installiert. Dafür legt der Installateur zunächst eine leere Datenbank an – im Beispiel unter MySQL etwa mittels »create database zabbix;«. Danach kommt das mitgelieferte Create-Skript für die jeweilige Datenbank zum Zug, das die nötigen Datenbankobjekte erzeugt. Unterhalb des Installationsverzeichnis findet es sich in »create/mysql«:

```
$ cat schema.sql | mysql -u root zabbix
$ cd ../data/
$ cat data.sql | mysql -u root -p zabbix
$ cat images.sql | mysql -u root -p
```

Dann kann es losgehen. Weil sich Zabbix im Moment stürmisch weiterentwickelt, empfiehlt

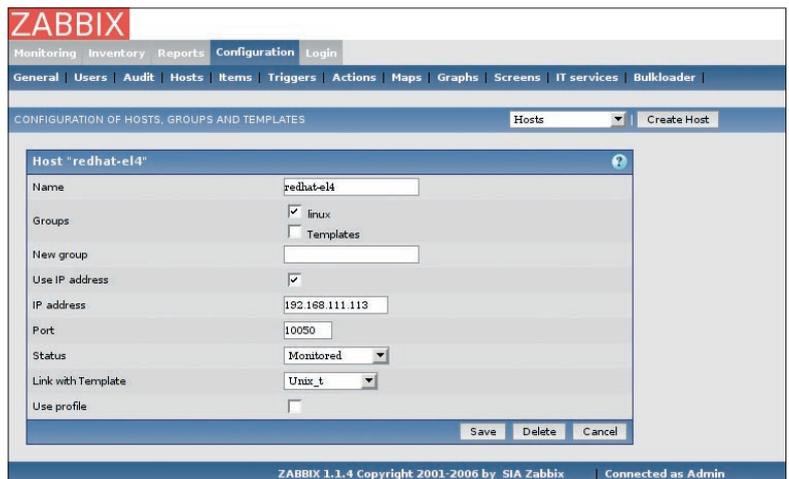


Abbildung 5: Neue Hosts lassen sich auf der Grundlage von Templates leicht integrieren.

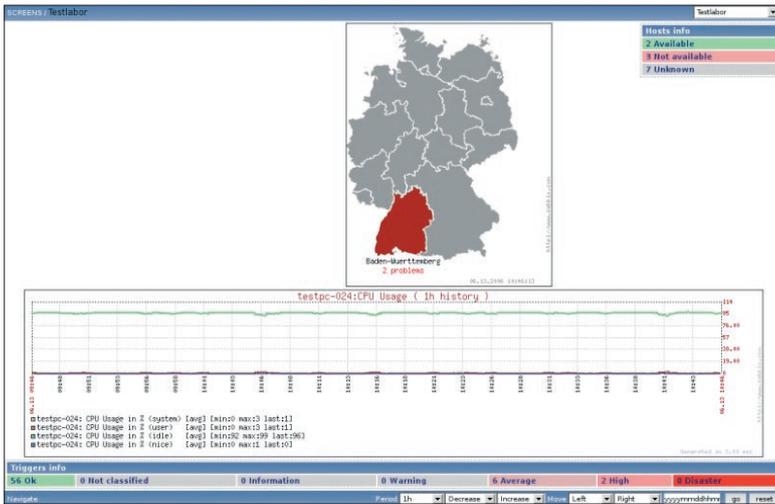


Abbildung 6: Bildschirmdarstellungen – so genannte Screens – lassen sich in Zabbix frei zusammenstellen. Das Beispiel zeigt eine Karte, die signalisiert, dass im Bundesland Baden-Württemberg zwei Probleme aufgetreten sind. Darunter wurde ein CPU-Last-Graph platziert.

sich die Installation aus den Quellen. Der Quellcode steht auf der Zabbix-Homepage (1) zum Download bereit. Das Configure-Skript benötigt folgende Optionen:

```
$ ./configure \
--enable-server \
--enable-agent \
--with-mysql \
--with-net-snmp
```

Der Parameter »--with-net-snmp« ist optional. Der Parameter »--with-mysql« hängt von der Datenbank ab, die als Backend verwendet wird. Im Beispiel soll der Zabbix-Server sich auf dem lokalen Rechner selbst überwachen. Dafür braucht es außerdem einen Agenten, also »--enable-agent«. Nach »configure« kompletieren »make« und »make install« den Dreisatz ohne Besonderheiten.

Konfiguration

Im Anschluss an das Übersetzen trägt der Installateur die nötigen Ports in »/etc/services« ein.

```
zabbix_agent 10050/tcp
zabbix_trap 10051/tcp
```

Die Konfigurationsdateien des Zabbix-Servers (»zabbix_server.conf«) und des Agenten (»zabbix-agentd.

conf«) gelangen leider nicht automatisch nach »/etc«, allerdings finden sich im Installationsverzeichnis im Unterordner »misc/conf/« Vorlagen mit einer Beispielkonfiguration. Sie lassen sich einfach nach »/etc/zabbix« kopieren. In der Konfigurationsdatei des Servers »zabbix_server.conf« sind folgende Einträge anzupassen:

```
ListenIP=<Server IP>
DBHost=<DB-Server>
DBName=zabbix
DBUser=<DB-User>
DBPassword=*****
```

Anschließend starten die Befehle

```
zabbix@monitoring:/etc/zabbix$ ?
/usr/local/bin/zabbix_server
zabbix@monitoring:/etc/zabbix$ ?
/usr/local/bin/zabbix_agentd
```

den Zabbix-Server und seinen Agenten. Das Kommando »zabbix_agentd« startet dabei den Zabbix-Agenten als Stand-alone Daemon, das Programm »zabbix_agent« ist für den Start des Agenten über den »inetd« verfügbar. Wer sich davon überzeugen möchte, dass Server und Agent tatsächlich laufen, schaut in die Protokolldateien – standardmäßig liegen sie in »/tmp«, lassen sich aber genauso gut in das passendere »/var/log«-Verzeichnis konfigurieren. Passende Init-Skripte für den automatischen Start liegen im Unterverzeichnis »misc/init.d/« für die meisten Distributionen bereit.

Konfiguration des Frontends und Aufruf

Der Code des Webfrontends von Zabbix ist komplett in PHP geschrieben und muss vom Installateur in das Document-Root des Webservers bewegt werden.

Apache ab 1.3.12
MySQL ab 3.22 oder PostgreSQL ab 7.0.2
MySQL- oder PostgreSQL-Header und -Libraries
PHP ab 4.0 oder 5.0
PHP GD- oder GD2-Modul
PHP MySQL- oder PostgreSQL-Modul
NET-SNMP oder UCD-SNMP-Header und -Libraries
OpenSSL-Header und -Libraries
GNU Make
GNU C Compiler

```
zabbix@monitoring:/var/www$ sudo cp -R \
/home/zabbix/install/zabbix-1.1.6/\
frontends/php/ ./zabbix
```

Danach ist ihm die verwendete Datenbank über seine Konfigurationsdatei bekannt zu machen:

```
zabbix@monitoring:/var/www/zabbix$ \
sudo vim include/db.inc.php
//$DB_TYPE = "ORACLE";
//$DB_TYPE = "POSTGRESQL";
$DB_TYPE = "MYSQL";
$DB_SERVER = "DB-Host";
```

In dieser Zeile sollte der Admin für dieses Beispiel »DB-Host« durch »localhost« (oder die IP des Datenbankservers) ersetzen. Die folgenden Zeilen enthalten die Zugangsdaten für den Datenbankserver:

```
$DB_DATABASE = "zabbix";
$DB_USER = "DB-User";
$DB_PASSWORD = "*****";
```

Danach ist das Frontend bereits startklar, man erreicht es über die URL [<http://Monitoring-Host/zabbix>]. Die Anmeldung ist mit dem Standardbenutzer »admin« ohne Passwort möglich.

Einschleusen eines Agenten

Solange es keinen Host überwacht, schaut das Webfrontend allerdings noch recht leer aus. Die Installation der Agenten erfolgt in analogen Schritten, allerdings reicht hier »configure --enable-agent« aus. Auf dem Client ist nach dem Übersetzen lediglich in der Konfigurationsdatei »zabbix_agentd.conf« der Server einzutragen. Das gelingt mit »Server=192.168.111.249«. Die Adresse muss mit dem Wert des Listen-Parameters der Server-Konfiguration übereinstimmen. Nun startet der Zabbix-Agent nach »/usr/local/bin/zabbix_agentd«.

Windows-Agenten

Der Zabbix-Agent für Windows-Systeme »ZabbixW32.exe« verwendet den Performance Data Helper PDH, um Informationen unter Windows 2000 und XP zu sammeln. In Windows NT 4.0 ist PDH nicht enthalten, ein Artikel aus der Microsoft Knowledge Base (2) beschreibt die nachträgliche Installation. WMI (Windows Management Instrumentation) wird offensichtlich noch nicht unterstützt, ebenso fehlt ein Client für Vista.

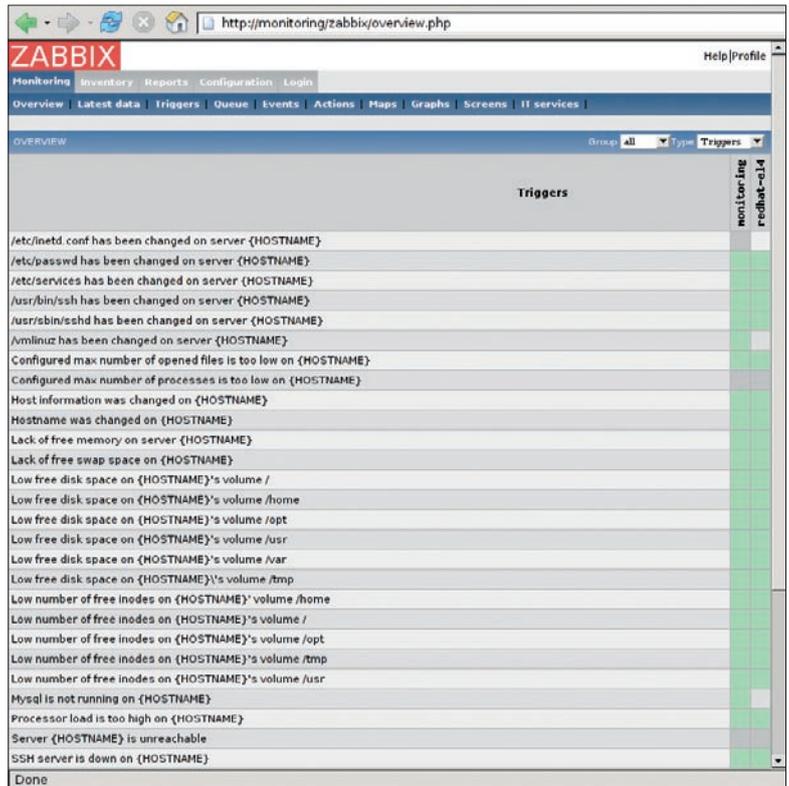


Abbildung 7: Eine Statusübersicht vermittelt einen raschen Überblick.

Die Konfiguration des Zabbix-Agenten vollzieht sich auch unter Windows in einer Konfigurationsdatei mit identischer Syntax wie unter Linux/Unix, eine Liste der spezifischen Windows-Parameter gibt es auf der Zabbix-Website unter (3). »ZabbixW32.exe« ist ein reines Befehlszeilentool, das zum Beispiel einen Syntaxcheck der Konfigurationsdatei ausführen oder als Quelle für das Windows Event Log dienen kann. Auf Wunsch integriert sich der Client in die Windows-Dienstverwaltung und wird von ihr automatisch gestartet und angehalten.

Zabbix konfigurieren

Im Webfrontend ist der Zabbix-Server über die Arbeitsaufnahme seines Agenten zu unterrichten. Das geschieht über »Login | Configuration | Hosts | Create Host«. Bereits unmittelbar danach sind dank der automatischen Checks zahlreiche Beobachtungswerte abfragbar. Ein Klick auf den Reiter »Monitoring« bestätigt das. Ganz rechts in dieser Liste der Trigger befindet sich jeweils eine Spalte für jeden überwachten Host. Weitere Einstellungen erfordern von nun an kein Editieren von Konfigurationsdateien mehr, sondern nur noch die Menüs der GUI. ▶

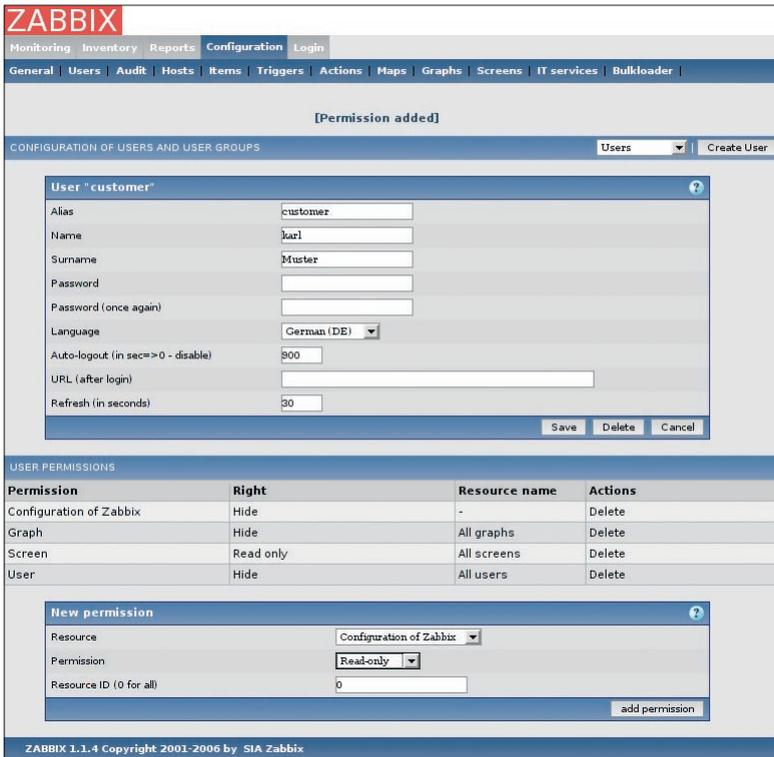


Abbildung 8: Die Benutzerverwaltung von Zabbix erlaubt ein fein abgestuftes Berechtigungskonzept.

Alarmierung

Meldet ein Agent eine Störung, kann er den Administrator auf den üblichen Wegen benachrichtigen: über E-Mail, SMS, Pager oder beliebige externe Programme. Das Zabbix-Forum auf Sourceforge (4) enthält zahlreiche Anleitungen dazu; die Spanne der Möglichkeiten reicht von Audio Alerts bis zur Benachrichtigung über WinPopUps.

Als Grundlage für Alarme dienen Zabbix wahlweise auch kumulierte Daten oder die Ergebnisse von Berechnungen. Ein Administrator erhält in diesem Fall bereits eine E-Mail, wenn der benötigte Speicher oder die durchschnittliche CPU-Auslastung einer Gruppe von Hosts, zum Beispiel eines Clusters, einen vordefinierten Schwellwert übersteigt. Die Vitalität der gesamten Infrastruktur zeigt die zentrale Statusübersicht (Abbildung 7).

Daneben lassen sich die Beobachtungswerte auch individuell verknüpfen, und beispielsweise automatisch mit den Eckdaten von Service Level Agreements (SLAs) vergleichen. Auch ausführliche Reports und Zusammenfassungen generiert Zabbix auf Wunsch. Darüber hinaus bietet es umfangreiche Möglichkeiten der Dokumentation, Analyse und Inventarisierung der überwachten Systeme.

Redundante Checks und unnötige Alarme vermeidet Zabbix durch Abhängigkeiten, die der Administrator in der Konfiguration der Trigger einträgt. Fällt beispielsweise ein Router aus, so erkennt Zabbix automatisch, dass diverse Server dahinter nicht mehr erreichbar sind und führt die überflüssigen Checks und Alarmierungen nicht durch. Mit der Zuordnung der Trigger zu Fehlerkategorien von »Information« bis »Desaster« bietet Zabbix Eskalationsstufen mit angepassten Benachrichtigungsmethoden.

Zabbix-Benutzerverwaltung

Für den Einsatz in größeren Unternehmen nützlich ist die flexible Benutzer- und Rechteverwaltung von Zabbix. Auch sie kann der Administrator komplett im Webfrontend konfigurieren (Abbildung 8). Im oberen Teil der Abbildung sind die Daten eines neu angelegten Benutzers und einige Einstellungen zu sehen. In der unteren Hälfte dieses Dialoges stehen die User-Berechtigungen.

Der Administrator hat für den Beispiel-User offensichtlich sehr strenge Vorgaben gewählt: Der Benutzer darf nur Screens betrachten (»Read

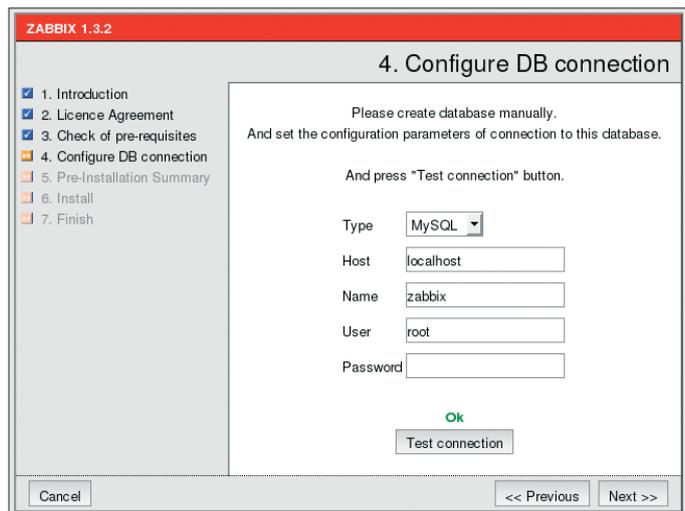


Abbildung 9: In der kommenden Zabbix-Version soll die Installation komplett in der Web-GUI ablaufen.

only«), die Konfiguration, Graphen und das Benutzermanagement bleiben vollständig vor ihm verborgen (»Hide«).

Der Dialog am unteren Ende des Bildschirms erstellt neue Berechtigungen (die verwendbaren Optionen sind »Add«, »Hide«, »Read-Write« und »Read-Only«). Durch die geschickte Verknüpfung von Ressource-IDs mit Berechtigungen kann der Administrator jeden User auf eine für ihn maßgeschneiderte Zabbix-Umgebung beschränken. Der Entzug von Ressourcen ist dabei umfassend und betrifft alle Elemente von Zabbix, einschließlich der Kommentare und Benachrichtigungen

Ausblick: die Beta-version

Neben der stabilen gibt es eine aktuelle Beta-version von Zabbix, die bereits einen Blick in die Zukunft gestattet. Ihre Installation verläuft komplett im Webfrontend und beinhaltet einen Systemcheck und die Einrichtung der Verbindung zum Datenbankserver, wie das bei anderen PHP-basierten Programmen auch der Fall ist.

Eines ihrer besonderen Features ist der Export und Import der Konfiguration von Objekten in oder aus XML-Dateien. Dadurch ließe sich beispielsweise ein Host bequem von einem zu einem anderen Zabbix-Server migrieren. Des Weiteren behoben die Programmierer zahlreiche Bugs, ergänzten die Unterstützung für SQLite3 und verbesserten viele weitere Details.

Auf der Roadmap steht auch, die in fast jedem Dialog verfügbaren Kommentarfelder im Webinterface nach und nach zu einem Trouble Ticket System auszubauen. Bisher bieten diese Felder immerhin die Möglichkeit, Fehler und Fehlerbehebung zu dokumentieren.

Das Engagement von SIA Zabbix spiegelt sich in umfangreichen, freien Dokumentationen mit Tutorials, Quick Start Guide und detaillierten Beschreibungen der Datenbankstrukturen wider. Mittlerweile sind auch Red Hat und Ubuntu auf Zabbix aufmerksam geworden, das Kompilieren der Quellen ist allerdings die einzige vollständige Installationsart und die einzige, die der Hersteller supportet. Anders als bei Nagios oder Cacti existieren keine offiziellen RPM- oder Debian-Pakete, die alle Features unterstützen.

Zusammenfassung

Zabbix macht einen sehr ausgereiften Eindruck, was angesichts des geringen Bekanntheitsgrades überrascht. Der Hersteller testet nach eigenen

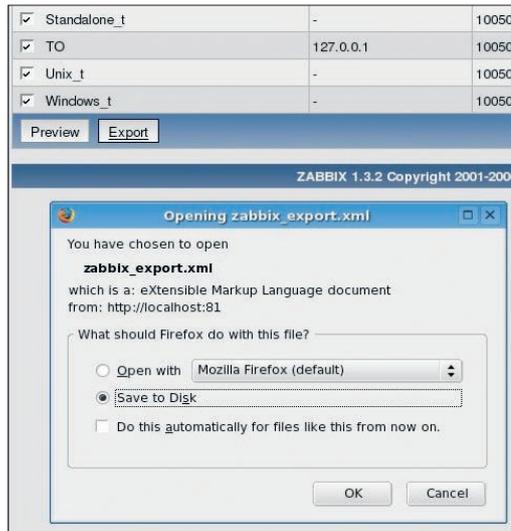


Abbildung 10: Beliebige Host-Daten lassen sich direkt aus dem Webfrontend als XML-File exportieren oder importieren.

Angaben regelmäßig mit mehr als 5000 Devices und Servern, er erhält Feedback aus zahlreichen Installationen in der Praxis und unterstützt die wachsende Community. Clients gibt es für fast alle Betriebssysteme. Der fehlende Vista-Client und die WMI-Unterstützung lassen sicher nicht lange auf sich warten. Der Ansatz, zur offenen Software kommerziellen Support anzubieten, hat sich in vielen anderen Projekten bewährt und dürfte Zabbix gerade für den Einsatz in Unternehmen interessant machen.

Das Webfrontend von Zabbix überzeugt durch eine einfache und verständliche Darstellung mit vielen Möglichkeiten. Zabbix kann Daten aus fast beliebigen Quellen sammeln und voneinander unabhängige Werte in gemeinsamen Karten, Screens oder Diagrammen anzeigen. Eine gelungene Benutzerverwaltung, umfangreiche Vorlagen und das gute Event-Management machen es zu einer ernstzunehmenden Alternative zu Nagios und Cacti. (mfe) ■■■

Infos

- (1) Zabbix-Homepage: (<http://www.zabbix.com/download.php>)
- (2) Installation PDH unter NT 4.0: (<http://support.microsoft.com/default.aspx?scid=kb;en-us;284996>)
- (3) Zabbix Windows-Agent: (http://www.zabbix.com/manual_config_w32.php)
- (4) Zabbix-Projektseite bei Sourceforge: (<https://sourceforge.net/projects/zabbix>)

Die Autoren



Bogdan Taru arbeitet als System-Engineer bei Thinking Objects Software GmbH. Er beschäftigt sich hauptsächlich mit den Themen AIX, Websphere und DB2.



Daniel Dörner arbeitet als System-Engineer bei Thinking Objects Software GmbH. Sein Arbeitsschwerpunkt ist Unix- und Linux-Administration.