

ZABBIX ESX-SERVER PER COMMON INFORMATION MODEL CIM UEBERWACHEN

[Druckversion](#)

Inhaltsverzeichnis [\[Verbergen\]](#)

[1 ESX CIM-API](#)

[2 CIM-API mit Zabbix abfragen](#)

[2.1 User im ESX- oder ESXi-Server einrichten](#)

[2.2 Python pywbem installieren](#)

[2.3 Hardwarestatus per CIM auslesen](#)

[3 ESX-Hardwareüberwachung in Zabbix einrichten](#)

[3.1 Discovery-Regel einrichten](#)

[3.2 Item-Prototypen einrichten](#)

[3.3 Cronjob einrichten](#)

[3.4 Trigger-Prototypen einrichten](#)

ESX CIM-API

ESX- und ESXi-Server verfügen über ein [Common Information Model API](#), welches es ermöglicht, den Systemstatus per HTTPS auszulesen.

Für ESXi-Server ist die CIM-API die einzige Möglichkeit, den Systemstatus aktiv abzufragen, da in der ESXi-Variante keine volle SNMP-Unterstützung zur Verfügung steht.

Über die CIM-API kann Zabbix all die Informationen auslesen, welche Sie auch über den vSphere-Client als Sensorstatus einsehen können. Der vSphere-Client und der Virtual Center Server nutzen ebenfalls die CIM-API.

Ob über die CIM-API alle Statusinformationen der Hardware exportiert werden, hängt von der Hardware und den in ESX integrierten CIM-Providern ab. Erst die passenden CIM-Provider ermöglichen dem ESX(i)-Server, die Hardwareinformationen auszulesen und in einer einheitlichen Form darzustellen. Für die gängige und von VMware zertifizierte Hardware sind schon viele CIM-Provider in den ESX(i)-Server integriert. Für einige Server oder Hardwarekomponenten müssen aber CIM-Provider nachinstalliert werden.

Sensor	Status
IBM IBM System x3650 -[644727G]-	✓ Normal
Prozessoren	✓ Normal
Temperatur	✓ Normal
Andere	✓ Normal
Spannung	✓ Normal
Lüftung	✓ Normal
Betrieb	✓ Normal
Softwarekomponenten	✓ Normal
System	✓ Normal
Watchdog	✓ Normal
Speicher	✓ Normal
Disk or Disk Bay 8 Drive 8 Status ...	✓ Normal
Disk or Disk Bay 7 Drive 7 Status ...	✓ Normal
Disk or Disk Bay 6 Drive 6 Status ...	✓ Normal
Disk or Disk Bay 5 Drive 5 Status ...	✓ Normal
Disk or Disk Bay 4 Drive 4 Status ...	✓ Normal
Disk or Disk Bay 3 Drive 3 Status ...	✓ Normal
Disk or Disk Bay 2 Drive 2 Status ...	✓ Normal
Disk or Disk Bay 1 Drive 1 Status ...	✓ Normal

Übersicht Hardwarestatus im Vsphere-Client, die auch Zabbix auslesen kann

vmware vSphere Web Client

Administrator@VSPHERE.LOCAL

esx-srv6 Aktionen

Erste Schritte Übersicht **Überwachen** Verwalten Verwandte Objekte

Probleme Leistung Protokoll-Browser Aufgaben Ereignisse **Hardwarestatus**

CIM-Daten Sensoren

Sensoren BIOS-Hersteller: HP, BIOS-Version: P89
 Alarme und Warnu... Modell: ProLiant DL380 Gen9, Seriennummer: CZJ5490N0W, Tag: 23.0, Asset Tag: unknown
 Systemereignispro... 3 Alarme 1 Warnung von 489 Sensoren.

Aktualisieren Sensoren zurücksetzen Daten exportieren Filter

Sensor	Status	Details
Prozessor	Normal	
Arbeitsspeicher	Normal	
PCI		
Lüftung	Normal	
Spannung	Normal	
Temperatur	Normal	
Betrieb	3 Alarme 1 Warnung	
Power Supply 1	Normal	
Power Supply 2	Alarm	
Power Supply 4	Normal	
Power Supply 3 Power Supplies		RedundancyStatus: Keine Redundanz
Power Supply 3 Power Supplies	Warnung	Aktueller Status: Redundancy lost
Power Supply 2 PS 2 Presence		Aktueller Status: Device inserted/present
Power Supply 1 PS 1 Presence		Aktueller Status: Device inserted/present
Power Supply 2 PS 2 Output	Normal	Lesen: 0 Watt
Power Supply 2 Power Supply 2: Presence detected		Lesen: 0 Watt
Power Supply 2 Power Supply 2: Failure status	Alarm	Lesen: 0 Watt
Power Supply 2 Power Supply 2: Power Supply AC lost	Alarm	Lesen: 0 Watt

31 Elemente

Ab VMware ESX6 wird der Hardware-Status nur noch im Webclient angezeigt.

CIM-API mit Zabbix abfragen

User im ESX- oder ESXi-Server einrichten

Richten Sie über den vSphere-Client einen neuen User ein, mit dem sich Zabbix zum ESX-Server verbinden kann. Der User muss der Gruppe root angehören.

Neuen Benutzer hinzufügen

Benutzerinformationen

Benutzer: zabbix UID:

Benutzername: zabbix

Benutzername und UID sind optional

Kennwort eingeben

Kennwort:

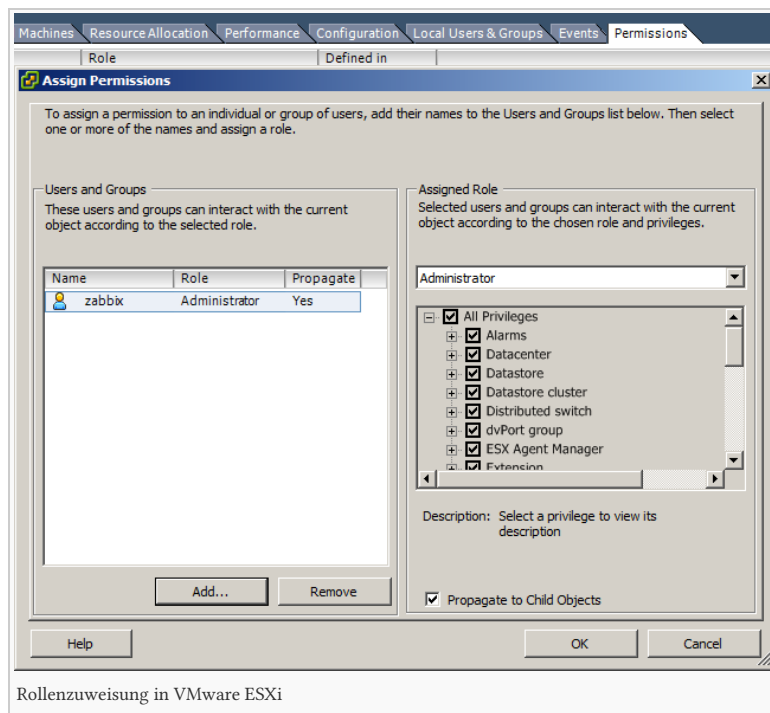
Bestätigen:

Gruppenmitgliedschaft

Gruppe: root

Einen Benutzer für das Monitoring anlegen

Ab ESXi5.1 gibt es keine Gruppen mehr. Wählen Sie den Tab "Permissions" an, nachdem Sie den neuen Benutzer angelegt haben. Weisen Sie dann dem Benutzer die Rolle "Administrator" zu.



Rollenzuweisung in VMware ESXi

Warnung: Leider können nur Benutzer der Gruppe root bzw. Benutzer mit Administratorrolle den Sensorstatus auslesen. Wählen Sie also ein sicheres Passwort und achten Sie darauf, dass nicht alle Nutzer Ihres Monitoring-Servers das Passwort aus dem externen Skript auslesen können.

Python pywbem installieren

Damit Zabbix die CIM-API abfragen kann, muss ein externes Skript installiert werden, welches per Zabbix-Sender die Daten der CIM-API an den Zabbix-Server schickt.

Für die Programmiersprache Python gibt es eine fertige Klassensammlung zum Abfragen von CIM-APIs per HTTP. Das Skript [check-esx-hardware.py](#) nutzt diese Bibliothek und liefert die Daten der CIM-API in einem für Zabbix-Sender passenden Format.

Nutzer von Ubuntu und CentOS/RHEL 7 können die benötigten Python-Bibliotheken über den Paketmanager mit dem Kommando `apt-get install python-pywbem` bzw. `yum install yum install pywbem` installieren.

Warnung: Nutzer von Debian 8 sollten pywbem nicht per `apt-get` und nicht per `pip` installieren. Aus nicht nachvollziehbaren Gründen ist das Debian-Paket python-wbem von python-epydoc abhängig, welches wiederum von Latex abhängig ist. Ein achtloses `apt-get install python-pywbem` würde neue Pakete in der Größe von ca 740 MB installieren. Wählen Sie besser die manuelle Installation von pywbem. Die aktuelle Version von pywbem, welche Sie mit pip installieren können, ist leider nicht kompatibel mit esx-check-skript.

Nutzer anderer Distributionen laden die Bibliotheken von [Sourceforge.net](http://sourceforge.net) herunter und installieren diese wie folgt:

```
apt-get install python-m2crypto
cd /tmp
wget http://vorboss.dl.sourceforge.net/project/pywbem/pywbem/pywbem-0.8.0/pywbem-0.8.0-dev.r728.zip
unzip pywbem-0.8.0-dev.r728.zip
cd pywbem-0.8.0-dev
python setup.py build
python setup.py install
```

Hardwarestatus per CIM auslesen

Laden Sie das Python-Skript [check-esx-hardware.py](#) herunter und kopieren Sie es in den Ordner, welchen Sie für die externen Skripte im Zabbix-Server konfiguriert haben. Machen Sie das Skript mit dem Kommando `chmod +x check-esx-hardware.py` ausführbar. Einen Überblick über die Optionen des Skriptes erhalten Sie mit dem üblichen Parameter `--help`.

```
root@monitoring:/usr/lib/zabbix/cron# ./check-esx-hardware.py --help
usage: check-esx-hardware.py [-h] [-u USER] -p PASSWORD -i HOST [-v] [-s] [-x]
                        [-d] [-z ZABBIX_HOSTNAME] [-n ITEM_NAME]
                        [-k DISCOVERY_ITEM_NAME]
```

Retrieve hardware health state from VMWare ESX(i) server

```
optional arguments:
-h, --help            Show this help message and exit.
-u USER, --user USER  Username for the connection to ESX(i) Server. If not
                        given, "root" is used.
-p PASSWORD, --password PASSWORD
                        Password for the connection to ESX(i) Server.
-i HOST, --host HOST  IP or Hostname of ESX(i) Server
-v, --verbose          Be more verbose.
-s, --show-inventory  Show the discovered items with formatted output.
-x, --zabbix-out       Print output formatted for piping to zabbix_sender.
-d, --item-discovery  Print json list for Zabbix low level item discovery.
-z ZABBIX_HOSTNAME, --zabbix-hostname ZABBIX_HOSTNAME
                        Name of host in Zabbix. If not given, host (-h/--host)
                        will be used.
-n ITEM_NAME, --item-name ITEM_NAME
                        Name of Item as configured in Zabbix Server. If not
                        given, "vmware.hardwarehealthstate" is used.
-k DISCOVERY_ITEM_NAME, --discovery-item-name DISCOVERY_ITEM_NAME
                        Name of Item for discovery as configured in Zabbix
                        Server. If not given, "vmware.hardwareitems" is used.
```

Testen Sie das Skript auf der Kommandozeile. Geben Sie den Hostnamen und das Passwort eines ESX-Servers an und lesen Sie den kompletten Hardwarestatus mit dem Parameter `--show-inventory` rekursiv aus.

```
root@monitoring:/usr/lib/zabbix/cron# ./check-esx-hardware.py -i esx1.example.local -p geheim --show-inventory
System Board 71                HealthState = 0
System Board 72                HealthState = 0
Add-in Card 111                HealthState = 0
Add-in Card 112                HealthState = 0
Add-in Card 113                HealthState = 0
Hardware Management Controller (Node 0) HealthState = 0
Processor 1 Temp                HealthState = 5
System Board 1 Pwr Consumption HealthState = 5
Power Supply 2 Voltage 2       HealthState = 5
Power Supply 1 Voltage 1       HealthState = 5
Power Supply 2 Current 2       HealthState = 5
Power Supply 1 Current 1       HealthState = 5
System Board 1 Sys Fan1 RPM     HealthState = 5
```

Geben Sie im Skript einen User und ein Passwort an, welchen Sie vorher per vSphere-Client auf dem ESX-Server eingerichtet haben. Die Nummern des HealthStates bedeuten

Nummer	Bewertung	Hinweis
0	OK	Unknown
5	OK	OK
10	Warning	Degraded
15	Warning	Minor
20	Critical	Major
25	Critical	Critical
30	Critical	Non-recoverable Error

ESX-Hardwareüberwachung in Zabbix einrichten

Discovery-Regel einrichten

Wenn Sie auf der Kommandozeile erfolgreich die CIM-API Ihres ESX- oder ESXi-Servers abfragen konnten, richten Sie nun die Items im Zabbix-Server ein. Dazu nutzen Sie die Funktion [Low-Level-Discovery](#) und Zabbix-Sender.

Wenn Sie das Skript mit der Option `-d` oder `--item-discovery` aufrufen, wird eine json-kodierte Liste der im ESX-Server erkannten Hardware erzeugt. Mit dieser Liste generiert der Zabbix-Server automatisch einzelne Items für jede Hardwarekomponente. (Die Liste im nachfolgenden Beispiel wurde gekürzt und leserlich formatiert.)

```
./check-esx-hardware.py -i esx1.example.com -p geheim --item-discovery
vmware.hardwareitems {
  "data": [{
    "#{ITEM_KEY}": "SystemBoard71",
    "#{ITEM_NAME}": "System Board 71"
  }, {
    "#{ITEM_KEY}": "SystemBoard72",
    "#{ITEM_NAME}": "System Board 72"
  }, {
    "#{ITEM_KEY}": "Add-inCard111",
    "#{ITEM_NAME}": "Add-in Card 111"
  }, {
    "#{ITEM_KEY}": "Add-inCard112",
    "#{ITEM_NAME}": "Add-in Card 112"
  }, {
    "#{ITEM_KEY}": "Add-inCard113",
    "#{ITEM_NAME}": "Add-in Card 113"
  }, {
    "#{ITEM_KEY}": "HardwareManagementController(Node0)",
    "#{ITEM_NAME}": "Hardware Management Controller (Node 0)"
  }, {
    "#{ITEM_KEY}": "Processor1Temp",
    "#{ITEM_NAME}": "Processor 1 Temp"
  }, {
    "#{ITEM_KEY}": "SystemBoard1PwrConsumption",
    "#{ITEM_NAME}": "System Board 1 Pwr Consumption"
  }, {
    "#{ITEM_KEY}": "PowerSupply2Voltage2",
    "#{ITEM_NAME}": "Power Supply 2 Voltage 2"
  }
}]
}
```

Legen Sie ein neues Template mit dem Namen „VMware Hardware Health“ an. Klicken Sie anschließend in der Zeile des neuen Templates auf den Link **Discovery** und legen Sie mit der Schaltfläche „Create Discovery Rule“ eine neue Regel mit folgenden Daten an:

Name	Hardware-Items
Type	Zabbix trapper
Key	vmware.hardwareitems
Keep lost resources period (in days)	30

CONFIGURATION OF DISCOVERY RULES

◀ Template list

Template: VMware Hardware Health

◀ Discovery list

Discovery: Hardware Items

Discovery rule

Name

Hardware Items

Type

Zabbix trapper

Key

vmware.hardwareitems

Keep lost resources period (in days)

30

Filter

Macro

Regexp

Allowed hosts

Description

Discovery-Regeln zum Anlegen einzelner Items pro Hardwarekomponente

Item-Prototypen einrichten

Nachdem Sie die Discovery-Regel angelegt haben, richten Sie einen „Item Prototype“ mit folgenden Daten ein:

Name	Health state of {#ITEM_NAME}
Type	Zabbix trapper
Key	vmware.hardwarehealthstate['{#ITEM_KEY}']
Type of Information	Numeric (unsigned)
Data type	Decimal

CONFIGURATION OF ITEM PROTOTYPES

◀ [Template list](#) **Template: VMWARE Hardware Health** ▶ [Discovery list](#) **Discovery: Hardware Items**

Item prototype

Name:

Type:

Key:

Type of information:

Data type:

Units:

Use custom multiplier: ☐

History storage period (in days):

Trend storage period (in days):

Store value:

Show value: [show value mappings](#)

Allowed hosts:

New application:

Applications:

Item-Prototyp für die VMwarehardwareüberwachung

Sobald Sie das Template einem Host zugewiesen haben, können Sie die Konfiguration testen. Dazu loggen Sie sich per SSH auf dem Zabbix-Server ein und rufen das zuvor beschriebene Skript wie folgt auf:

```
. /check-esx-hardware.py -i esx1.example.com -p geheim -d/zabbix_sender -z 127.0.0.1 -i - -vvv
```

Das Skript baut über das CIM-Interface eine Verbindung zum ESX-Server auf, ruft alle Hardware-Items ab, generiert eine json-kodierte Liste und sendet diese per Zabbix-Sender an den Zabbix-Server. Sollte der Host in Zabbix einen abweichenden Namen haben, können Sie diesen mit der Option `--zabbix-hostname "Mein ESX Server"` angeben. Sobald das Skript seine Arbeit ausgeführt hat, sollten Sie zahlreiche neue Items für den Host sehen. Klicken Sie auf [Configuration/Hosts](#) und in der Zeile des entsprechenden Hosts auf den Link [Items](#).

In der Liste erkennen Sie die von der Discovery-Regel automatisch angelegten Items an dem vorangestellten Namen der Regel, in diesem Beispiel „Hardware Items“:

Items	
Displaying 1 to 50 of 238 found	
◀ Host list Host: esx1.example.com Monitored Applications (4) Items (238) Triggers (184) Graphs (0) Discovery rules (4) Web scenarios (0)	
Name	Key
Hardware Items : Health state of Add-in Card 1 ROMB Battery 0 Failed	vmware.hardwarehealthstate['Add-inCard1ROMBBattery0Failed']
Hardware Items : Health state of Add-in Card 111	vmware.hardwarehealthstate['Add-inCard111']
Hardware Items : Health state of Add-in Card 112	vmware.hardwarehealthstate['Add-inCard112']
Hardware Items : Health state of Add-in Card 113	vmware.hardwarehealthstate['Add-inCard113']
Hardware Items : Health state of CPU1	vmware.hardwarehealthstate['CPU1']
Hardware Items : Health state of CPU1 Level-1 Cache	vmware.hardwarehealthstate['CPU1Level-1Cache']
Hardware Items : Health state of CPU1 Level-2 Cache	vmware.hardwarehealthstate['CPU1Level-2Cache']
Hardware Items : Health state of CPU1 Level-3 Cache	vmware.hardwarehealthstate['CPU1Level-3Cache']
Hardware Items : Health state of Disk Drive Bay 1 Cable SAS A 0 Config Error	vmware.hardwarehealthstate['DiskDriveBay1CableSASA0ConfigError']
Hardware Items : Health state of Disk Drive Bay 1 Cable SAS B 0 Config Error	vmware.hardwarehealthstate['DiskDriveBay1CableSASB0ConfigError']
Hardware Items : Health state of Disk Drive Bay 1 Power Cable 0 Config Error	vmware.hardwarehealthstate['DiskDriveBay1PowerCable0ConfigError']
Hardware Items : Health state of Disk Drive Bay 1 Signal Cable 0 Config Error	vmware.hardwarehealthstate['DiskDriveBay1SignalCable0ConfigError']
Hardware Items : Health state of Hardware Management Controller (Node 0)	vmware.hardwarehealthstate['HardwareManagementController(Node0)']
Hardware Items : Health state of IPMI SEL	vmware.hardwarehealthstate['IPMISEL']

Das Skript hat per Zabbix-Sender neue Items angelegt.

Heartbeat:

Das Skript `check-esx-hardware.py` gibt bei jedem Aufruf ein Item mit dem Key `vmware.hardwarehealthstate[heartbeat]` aus. Richten Sie im Template „VMware Hardware Health“ ein „normales“ Item vom Typ „Zabbix-Trapper“ ein, das mit jedem Lauf des Skriptes aktualisiert wird. Per Trigger prüfen Sie später, ob das Skript regelmäßig Daten liefert.

Cronjob einrichten

Wenn die ersten Tests auf der Kommandozeile erfolgreich waren und wenn der Zustand der Hardware im ESX-Server in den einzelnen Items aktualisiert wird, sollten Sie einen Cronjob einrichten, der das Skript regelmäßig aufruft. Nutzen Sie entweder das Kommando `crontab` oder legen Sie eine Datei in `/etc/cron.d/` an, die beispielsweise wie folgt aussehen könnte:

```
*/30 * * * * root /usr/lib/zabbix/cron/check-esx-hardware.py \
-i esx1.example.com -p geheim -d -x \
zabbix_sender -z 127.0.0.1 -i -
```

Trigger-Prototypen einrichten

Wenn das Anlegen der Items per Low-Level-Discovery erfolgreich war, sollten Sie als letzten Schritt einen Trigger einrichten. Wählen Sie erneut über [Configuration/Templates](#) das Template „VMware Health State“ aus und klicken Sie auf den Link „Discovery“. Legen Sie nun über den Link „Trigger Prototyp“ einen neuen Trigger-Prototypen an. Der ESX-Server kennzeichnet defekte Hardware mit den Status 0 oder 5. Legen Sie den Trigger-Prototypen mit folgenden

Werten an: Name:

Hardware Failure on {#ITEM_NAME}

Expression (ohne Zeilenumbruch):

{VMware Hardware Health:vmware.hardwarehealthstate[{#ITEM_KEY}].last()}<>5 and {VMware Hardware Health:vmware.hardwarehealthstate[{#ITEM_KEY}].last()}<>0

CONFIGURATION OF TRIGGER PROTOTYPES

« Template list

Template: VMware Hardware Health

« Discovery list

Discovery: Hardware Item

Trigger prototype

Name

Hardware Failure on {#ITEM_NAME}

Expression

{VMware Hardware Health:vmware.hardwarehealthstate[{#ITEM_KEY}].last()}<>5 and {VMware Hardware Health:vmware.hardwarehealthstate[{#ITEM_KEY}].last()}<>0

Add

Expression constructor

Multiple PROBLEM events generation

☐

Description

URL

Severity

Not classified

Information

Warning

Average

High

Enabled

☒

Trigger-Prototyp, der defekte Hardware in ESX-Servern erkennt

Heartbeat:

Berücksichtigen Sie auch den Fall, dass das Python-Skript keine Daten mehr liefert. Richten Sie im Template „VMware Health State“ einen Trigger ein, welcher auslöst, wenn der Zabbix-Server keine Daten vom Skript erhält. Verwenden Sie dazu die Expression `nodata()`, zum Beispiel `{VMware Hardware Health:vmware.hardwarehealthstate[heartbeat].nodata(1800)}=1`.

Kategorie

Zabbix

MEINE WERKZEUGE

Anmelden

LINKS

Blog

Zabbix Schulungen

INFORMATIONEN

Diese Seite wurde bisher 15.832 mal abgerufen.

Diese Seite wurde zuletzt am 12. April 2016 um 16:41 Uhr von [Thorsten Kramm](#) geändert.

Datenschutz

Über freie Dokumentation

Impressum