

## 4 IPMI checks

### Overview

You can monitor the health and availability of Intelligent Platform Management Interface (IPMI) devices in Zabbix.

To perform IPMI checks Zabbix server must be initially [configured](#) with IPMI support.

IPMI is a standardized interface for remote “lights-out” or “out-of-band” management of computer systems. It allows to monitor hardware status directly from the so-called “out-of-band” management cards, independently from the operating system or whether the machine is powered on at all.

Zabbix IPMI monitoring works only for devices having IPMI support (HP iLO, DELL DRAC, IBM RSA, Sun SSP, etc).

See also [known issues](#) for IPMI checks.

### Configuration

#### Host configuration

A host must be configured to process IPMI checks. An IPMI interface must be added, with the respective IP and port numbers, and IPMI authentication parameters must be defined.

See the [configuration of hosts](#) for more details.

#### Server configuration

By default, the Zabbix server is not configured to start any IPMI pollers, thus any added IPMI items won't work. To change this, open the Zabbix server configuration file ([zabbix\\_server.conf](#)) as root and look for the following line:

```
# StartIPMIPollers=0
```

Uncomment it and set poller count to, say, 3, so that it reads:

```
StartIPMIPollers=3
```

Save the file and restart `zabbix_server` afterwards.

#### Item configuration

When [configuring an item](#) on a host level:

- For *Host interface* select the IPMI IP and port

- Select 'IPMI agent' as the *Type*
- Specify the *IPMI sensor* (for example 'FAN MOD 1A RPM' on Dell Poweredge)
- Enter an item [key](#) that is unique within the host (say, ipmi.fan.rpm)
- Select the respective type of information ('Numeric (float)' in this case, for discrete sensors - 'Numeric (unsigned)'), units (most likely 'rpm') and any other required item attributes

## Timeout and session termination

IPMI message timeouts and retry counts are defined in OpenIPMI library. Due to the current design of OpenIPMI, it is not possible to make these values configurable in Zabbix, neither on interface nor item level.

IPMI session inactivity timeout for LAN is 60 +/-3 seconds. Currently it is not possible to implement periodic sending of Activate Session command with OpenIPMI. If there are no IPMI item checks from Zabbix to a particular BMC for more than the session timeout configured in BMC then the next IPMI check after the timeout expires will time out due to individual message timeouts, retries or receive error. After that a new session is opened and a full rescan of the BMC is initiated. If you want to avoid unnecessary rescans of the BMC it is advised to set the IPMI item polling interval below the IPMI session inactivity timeout configured in BMC.

## Notes on IPMI discrete sensors

To find sensors on a host start Zabbix server with **DebugLevel=4** enabled. Wait a few minutes and find sensor discovery records in Zabbix server logfile:

```
$ grep 'Added sensor' zabbix_server.log
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:7 id:'CATERR' reading_type:0x3 ('discrete_state') type:0x7
('processor') full_name:'(r0.32.3.0).CATERR'
8358:20130318:111122.170 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:15 id:'CPU Therm Trip' reading_type:0x3 ('discrete_state') type:0x1
('temperature') full_name:'(7.1).CPU Therm Trip'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:17 id:'System Event Log' reading_type:0x6f ('sensor specific')
type:0x10 ('event_logging_disabled') full_name:'(7.1).System Event Log'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:17 id:'PhysicalSecurity' reading_type:0x6f ('sensor specific')
type:0x5 ('physical_security') full_name:'(23.1).PhysicalSecurity'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:14 id:'IPMI Watchdog' reading_type:0x6f ('sensor specific') type:0x23
('watchdog_2') full_name:'(7.7).IPMI Watchdog'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:16 id:'Power Unit Stat' reading_type:0x6f ('sensor specific') type:0x9
('power_unit') full_name:'(21.1).Power Unit Stat'
8358:20130318:111122.171 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:16 id:'P1 Therm Ctrl %' reading_type:0x1 ('threshold') type:0x1
('temperature') full_name:'(3.1).P1 Therm Ctrl %'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0
```

```
id_sz:16 id:'P1 Therm Margin' reading_type:0x1 ('threshold') type:0x1
('temperature') full_name:'(3.2).P1 Therm Margin'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:13 id:'System Fan 2' reading_type:0x1 ('threshold') type:0x4 ('fan')
full_name:'(29.1).System Fan 2'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:13 id:'System Fan 3' reading_type:0x1 ('threshold') type:0x4 ('fan')
full_name:'(29.1).System Fan 3'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:14 id:'P1 Mem Margin' reading_type:0x1 ('threshold') type:0x1
('temperature') full_name:'(7.6).P1 Mem Margin'
8358:20130318:111122.172 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:17 id:'Front Panel Temp' reading_type:0x1 ('threshold') type:0x1
('temperature') full_name:'(7.6).Front Panel Temp'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:15 id:'Baseboard Temp' reading_type:0x1 ('threshold') type:0x1
('temperature') full_name:'(7.6).Baseboard Temp'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:9 id:'BB +5.0V' reading_type:0x1 ('threshold') type:0x2 ('voltage')
full_name:'(7.1).BB +5.0V'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:14 id:'BB +3.3V STBY' reading_type:0x1 ('threshold') type:0x2
('voltage') full_name:'(7.1).BB +3.3V STBY'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:9 id:'BB +3.3V' reading_type:0x1 ('threshold') type:0x2 ('voltage')
full_name:'(7.1).BB +3.3V'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:17 id:'BB +1.5V P1 DDR3' reading_type:0x1 ('threshold') type:0x2
('voltage') full_name:'(7.1).BB +1.5V P1 DDR3'
8358:20130318:111122.173 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:17 id:'BB +1.1V P1 Vccp' reading_type:0x1 ('threshold') type:0x2
('voltage') full_name:'(7.1).BB +1.1V P1 Vccp'
8358:20130318:111122.174 Added sensor: host:'192.168.1.12:623' id_type:0
id_sz:14 id:'BB +1.05V PCH' reading_type:0x1 ('threshold') type:0x2
('voltage') full_name:'(7.1).BB +1.05V PCH'
```

To decode IPMI sensor types and states, get a copy of IPMI 2.0 specifications at <http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-specifications.html> (At the time of writing the newest document was <http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/second-gen-interface-spec-v2.pdf>)

The first parameter to start with is “reading\_type”. Use “Table 42-1, Event/Reading Type Code Ranges” from the specifications to decode “reading\_type” code. Most of the sensors in our example have “reading\_type:0x1” which means “threshold” sensor. “Table 42-3, Sensor Type Codes” shows that “type:0x1” means temperature sensor, “type:0x2” - voltage sensor, “type:0x4” - Fan etc. Threshold sensors sometimes are called “analog” sensors as they measure continuous parameters like temperature, voltage, revolutions per minute.

Another example - a sensor with “reading\_type:0x3”. “Table 42-1, Event/Reading Type Code Ranges” says that reading type codes 02h-0Ch mean “Generic Discrete” sensor. Discrete sensors have up to

15 possible states (in other words - up to 15 meaningful bits). For example, for sensor 'CATERR' with "type:0x7" the "Table 42-3, Sensor Type Codes" shows that this type means "Processor" and the meaning of individual bits is: 00h (the least significant bit) - IERR, 01h - Thermal Trip etc.

There are few sensors with "reading\_type:0x6f" in our example. For these sensors the "Table 42-1, Event/Reading Type Code Ranges" advises to use "Table 42-3, Sensor Type Codes" for decoding meanings of bits. For example, sensor 'Power Unit Stat' has type "type:0x9" which means "Power Unit". Offset 00h means "PowerOff/Power Down". In other words if the least significant bit is 1, then server is powered off. To test this bit a function **band** with mask 1 can be used. The trigger expression could be like

```
{www.zabbix.com:Power Unit Stat.band(#1,1)}=1
```

to warn about a server power off.

#### Notes on discrete sensor names in OpenIPMI-2.0.16, 2.0.17, 2.0.18 and 2.0.19

Names of discrete sensors in OpenIPMI-2.0.16, 2.0.17 and 2.0.18 often have an additional "0" (or some other digit or letter) appended at the end. For example, while `ipmitool` and OpenIPMI-2.0.19 display sensor names as "PhysicalSecurity" or "CATERR", in OpenIPMI-2.0.16, 2.0.17 and 2.0.18 the names are "PhysicalSecurity0" or "CATERR0", respectively.

When configuring an IPMI item with Zabbix server using OpenIPMI-2.0.16, 2.0.17 and 2.0.18, use these names ending with "0" in the *IPMI sensor* field of IPMI agent items. When your Zabbix server is upgraded to a new Linux distribution, which uses OpenIPMI-2.0.19 (or later), items with these IPMI discrete sensors will become "NOT SUPPORTED". You have to change their *IPMI sensor* names (remove the '0' in the end) and wait for some time before they turn "Enabled" again.

#### Notes on threshold and discrete sensor simultaneous availability

Some IPMI agents provide both a threshold sensor and a discrete sensor under the same name. In Zabbix versions prior to 2.2.8 and 2.4.3, the first provided sensor was chosen. Since versions 2.2.8 and 2.4.3, preference is always given to the threshold sensor.

#### Notes on connection termination

If IPMI checks are not performed (by any reason: all host IPMI items disabled/notsupported, host disabled/deleted, host in maintenance etc.) Zabbix server/proxy will continue polling IPMI host until server/proxy restart.

From:

<https://www.zabbix.com/documentation/2.4/> - **Zabbix Documentation 2.4**

Permanent link:

<https://www.zabbix.com/documentation/2.4/manual/config/items/itemtypes/ipmi>

Last update: **2016/01/13 09:45**

