

4 Проверки IPMI

Обзор

Вы можете наблюдать за состоянием и доступностью устройств Intelligent Platform Management Interface (IPMI) в Zabbix. Для выполнения проверок по IPMI Zabbix сервер должен быть изначально [сконфигурирован](#) с поддержкой IPMI.

IPMI стандартизованный интерфейс для удаленного управления “lights-out” или “out-of-band” компьютерными системами. Он позволяет наблюдать за состоянием аппаратного обеспечения напрямую с так называемых карт управления “out-of-band”, независимо от операционной системы или же от наличия питания на машине.

Zabbix IPMI мониторинг работает только с устройствами имеющими поддержку IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP и т.п.).

Начиная с Zabbix 3.4, добавлен новый процесс IPMI менеджер, который выполняет распределение проверок IPMI между IPMI поллерами. Теперь, узел сети всегда опрашивается одним и тем же IPMI поллером, таким образом снижается количество открытых подключений к BMC контроллерам. Благодаря этим изменениям можно безопасно увеличивать количество IPMI поллеров, не беспокоясь о перегрузке BMC контроллеров. Процесс IPMI менеджер автоматически запускается, если запускается хотя бы один IPMI поллер.

Смотрите также [известные проблемы](#) по IPMI проверкам.

Настройка

Настройка узла сети

Узел сети необходимо настроить для обработки проверок IPMI. Необходимо добавить интерфейс IPMI, с соответствующими IP адресом и номером порта, и задать параметры аутентификации IPMI.

Смотрите [настройку узлов сети](#) для получения более подробных сведений.

Настройка сервера

По умолчанию, Zabbix сервер не запускает IPMI поллеры, таким образом любые добавленные элементы данных IPMI не будут работать. Чтобы изменить это, откройте файл конфигурации ([zabbix_server.conf](#)) Zabbix сервера из под root и найдите следующую строку:

```
# StartIPMIPollers=0
```

Раскомментируйте эту строку и задайте количество поллеров, скажем, равное 3 так, чтобы строка была следующей:

StartIPMIPollers=3

Сохраните файл и затем перезапустите `zabbix_server`.

Настройка элемента данных

Для [настройки элемента данных](#) на уровне узла сети:

- Для *Интерфейса узла сети* выберите IPMI IP и порт
- Выберите 'IPMI агент' как *Тип*
- Укажите *IPMI датчик* (например 'FAN MOD 1A RPM' на Dell Poweredge). По умолчанию, необходимо указать ID датчика. Также имеется возможность использования префиксов до самого значения:
 - `id`: - чтобы указать ID датчика;
 - `name`: - чтобы указать полное имя датчика. Эта опция может быть полезна в ситуациях, когда датчики можно отличить только указав полное имя.
- Введите [ключ](#) элемента данных, уникальный в пределах узла сети (скажем, `ipmi.fan.rpm`)
- Выберите соответствующий тип информации ('Числовой (с плавающей точкой)' в этом случае, для дискретных датчиков - 'Числовой (целый)'), единицы измерения (скорее всего 'rpm') и любые другие требуемые атрибуты элемента данных.

Время ожидания и завершение сессии

Время ожидания IPMI сообщений и количества попыток определены в библиотеке OpenIPMI. В связи с текущим дизайном OpenIPMI, невозможно сделать эти значения настраиваемыми из Zabbix, ни на уровне интерфейса, ни на уровне элемента данных.

Время ожидания неактивности IPMI сессии для LAN равняется 60 +/-3 секунд. В настоящее время невозможно реализовать периодическую отправку команды Активации Сессии в OpenIPMI. Если проверки IPMI элементов данных выполняются от Zabbix к конкретному BMC в течении более чем время ожидания сессии, настроенное в BMC, тогда следующая проверка IPMI после того, как истечет время ожидания, приведет к отдельным сообщениям о превышении времени ожидания, попыток или ошибке при получении. После того, как откроется новая сессия и сделано полное повторное сканирование BMC. Если вы хотите избежать лишних сканирований BMC, рекомендуется установить интервал опроса IPMI элементов данных ниже времени ожидания неактивности IPMI сессии, которое настраивается в BMC.

Примечания о дискретных датчиках IPMI

Для поиска датчиков на узле сети запустите Zabbix сервер с включенным **DebugLevel=4**. Подождите пару минут и найдите записи об обнаруженных датчиках в журнале Zabbix сервера:

```
$ 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'
```

Для расшифровки типов датчиков IPMI и их состояния, скачайте экземпляр спецификаций IPMI 2.0 с <http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-specifications.html> (Во время подготовки документации новейшим документом был <http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/second-gen-interface-spec-v2.pdf>)

Начнем с параметра “тип_чтения”(“reading_type”). Для расшифровки кода “reading_type” используйте раздел “Table 42-1, Event/Reading Type Code Ranges” из спецификации. Большинство датчиков из нашего примера имеют “reading_type:0x1” означающих “порог” датчика. “Table 42-3, Sensor Type Codes” показывает, что “type:0x1” - температурный датчик, “type:0x2” - датчик напряжения, “type:0x4” - датчик частоты вращения вентилятора системы охлаждения и так далее. Пороговые датчики иногда называют “аналоговыми” датчиками, так как они измеряют непрерывные параметры, такие как температуру, напряжение, частоту вращения в минуту.

Другой пример - датчик с “reading_type:0x3”. “Table 42-1, Event/Reading Type Code Ranges” говорит, что считываемые типы кодов 02h-0Ch это “Общий Дискретный” датчик. Дискретные датчики имеют до 15 возможных состояний (другими словами - до 15 значимых бит). К примеру, для датчика 'CATERR' с “type:0x7” “Table 42-3, Sensor Type Codes” показывает, что этот тип обозначает “Процессор” и значение отдельных бит: 00h (наименьший значимый бит) - IERR(внутренняя ошибка процессора), 01h - Перегрев процессора и т.д.

Есть в нашем примере несколько датчиков с “reading_type:0x6f”. Для этих датчиков “Table 42-1, Event/Reading Type Code Ranges” советует использовать “Table 42-3, Sensor Type Codes” для расшифровки значений бит. Например, датчик 'Power Unit Stat' имеет тип “type:0x9”, который означает “Блок питания”. Смещение 00h означает “Выключено/Обесточено”. Другими словами, если младший значимый бит равен 1, то сервер выключен. Для проверки этого бита можно воспользоваться функция **band** с маской 1. Выражение триггера может выглядеть следующим образом

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

для предупреждения о выключенном сервере.

Заметки о именах дискретных датчиков в OpenIPMI-2.0.16, 2.0.17, 2.0.18 и 2.0.19

Имена дискретных датчиков в OpenIPMI-2.0.16, 2.0.17 и 2.0.18 зачастую имеют дополнительный символ “0” (или какую-то другую цифру или символ), присоединенный к концу имени. Например, тогда как ipmitool и OpenIPMI-2.0.19 отображают имена датчиков как “PhysicalSecurity” или “CATERR”, в OpenIPMI-2.0.16, 2.0.17 и 2.0.18 следующее именование “PhysicalSecurity0” или “CATERR0”, соответственно.

При настройке элемента данных IPMI для Zabbix сервера использующего OpenIPMI-2.0.16, 2.0.17

и 2.0.18, добавьте к их именам "0" в поле *IPMI датчик* для элементов данных IPMI агента. Когда ваш Zabbix сервер будет обновлен в новом Linux дистрибутиве, который использует OpenIPMI-2.0.19 (или более позднюю), элементы данных с такими IPMI дискретными датчиками перейдут в "NOT SUPPORTED". Вам потребуется изменить их имена *IPMI датчик* (удалить с конца '0') и подождать некоторое время пока они станут "Активированными" снова.

Заметки о одновременной доступности пороговых и дискретных датчиках

Некоторые IPMI агенты предоставляют одновременно пороговые и дискретные датчики под одним именем. В версиях Zabbix до 2.2.8, выбирался первый полученный датчик. Начиная с версии 2.2.8, предпочтение всегда отдается пороговому датчику.

Примечания о завершении соединений

Если IPMI проверки не выполняются (по любой из причин: все элементы данных IPMI деактивированы/не поддерживаются на узле сети, сам узел сети деактивирован/удален, узел сети в обслуживании и так далее), то в этом случае Zabbix сервер/прокси продолжит сбор данных по IPMI с этого узла сети до перезагрузки сервера/прокси.

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/ru/manual/config/items/itemtypes/ipmi>

Last update: **2018/07/12 08:49**

