

4 Verificações IPMI

Visão geral

Você pode monitorar a saúde e disponibilidade de seus dispositivos através da 'Intelligent Platform Management Interface' (IPMI).

Para executar verificações IPMI o Zabbix Server/Proxy precisa ser [configurado](#) com suporte ao IPMI.

O IPMI é uma interface padronizada para monitoração independente através da interface secundária de gerenciamento do servidor. Ele permite monitorar o status do hardware diretamente através de cartões de gerenciamento secundários, de forma independente do sistema operacional ou, até mesmo, sem o servidor estar ligado.

O monitoramento do Zabbix através de IPMI funcionará com dispositivos que possuem este suporte (HP iLO, DELL DRAC, IBM RSA, Sun SSP, etc).

Consulte [problemas conhecidos de IPMI](#) para maiores informações.

Configuração

Configuração do Host

Para que a monitoração através do IPMI fique funcional, o host deverá ser configurado. Deverá ser adicionada uma interface IPMI com os parâmetros de localização e autenticação (IP, porta, etc).

Para maiores detalhes consulte: [configuração dos hosts](#).

Configuração do Servidor

Por padrão o Zabbix Server não é configurado para iniciar os coletores IPMI, logo os itens IPMI adicionados não irão funcionar. Para modificar isso, você deverá alterar o arquivo de configuração do Zabbix Server ([zabbix_server.conf](#)) e localizar a linha a seguir:

```
# StartIPMIPollers=0
```

Remova o comentário e defina a quantidade de coletores (poolers), poderia ser algo como o exemplo a seguir:

```
StartIPMIPollers=3
```

Save the file and restart zabbix_server afterwards.

Configuração do Item

Quando estiver [configurando o item](#) no nível do host:

- Defina o *Tipo* do item como sendo 'Agente IPMI'
- Selecione no campo *Interface do host* uma interface IPMI
- Defina qual *Sensor IPMI* deverá ser coletado (por exemplo 'FAN MOD 1A RPM' em um Dell Poweredge)
- Informe uma *chave* que seja única no host (por exemplo: ipmi.fan.rpm)
- Selecione o respectivo tipo de informação ('Numérico (fracionário)' neste caso, para sensores discretos - 'Numérico (inteiro sem sinal)'), unidades (provavelmente 'rpm') e outros atributos necessários

Limite de tempo (timeout) e fim de sessão

As mensagens de 'timeout' do IPMI e contadores de novas tentativas são definidos na biblioteca OpenIPMI. A implementação atual do OpenIPMI não permite que o Zabbix defina estes valores no nível de interface ou de item.

O 'timeout' de sessão IPMI para a LAN é de 60 segundos (com margem de erro de +/-3 segundos). Atualmente não é possível definir o envio periódico do comando de sessão ativa com o OpenIPMI. Se não existirem verificações IPMI do Zabbix para um BMC em particular, por tempo superior ao 'timeout' configurado no BMC, a próxima verificação IPMI solicitada irá receber mensagem de 'timeout', tentar novamente ou erro. Após isso será iniciada uma nova sessão e uma verificação completa do BMC. Se você quiser evitar verificações desnecessárias do BMC é recomendável que o tempo em 'pooling' do item seja inferior ao 'timeout' configurado no BMC.

Para localizar os sensores em um host, inicie o Zabbix Server com o nível de log para **debug** (DebugLevel=4). **Aguarde alguns minutos e localize os registros de descoberta de sensores no arquivo de log: === Observações sobre sensores discretos no IPMI === \$ 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' Para decodificar os tipos e estados dos sensores IPMI, baixe uma cópia da especificação do IPMI 2.0 em

<http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-specifications.html> (No momento que escrevemos esta documentação a versão mais nova do documento era: <http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/second-generation-interface-spec-v2.pdf>) O primeiro parâmetro para começar é o “reading_type”. Use “Table 42-1, Event/Reading Type Code Ranges” existente nas especificações para decodificar o “reading_type”. A maioria dos sensores em nosso exemplo possuem “reading_type:0x1” que significa ser um sensor de “limiar”. “Table 42-3, Sensor Type Codes” demonstra que “type:0x1” significa um sensor de temperatura, “type:0x2” - um sensor de voltagem, “type:0x4” - ventilador (cooler/Fan) etc. Os sensores de limiar algumas vezes são chamados de sensores analógicos que medem continuamente parâmetros como temperatura, voltagem, rpm, etc. Outro exemplo - um sensor com “reading_type:0x3”. A tabela “Table 42-1, Event/Reading Type Code Ranges” informa que os códigos de leitura 02h-0Ch são sensores “Discretos e genéricos”. Os sensores discretos possuem até 15 estados possíveis (em outras palavras até 15 bits úteis). Por exemplo, para o sensor 'CATERR' com “type:0x7” a tabela “Table 42-3, Sensor Type Codes” demonstra que este tipo se refere ao “Processor” e o significado do bit menos significativo: 00h - IERR, 01h - Trilha térmica etc. Temos também alguns sensores com “reading_type:0x6f” em nosso exemplo. Para estes sensores a tabela “Table 42-1, Event/Reading Type Code Ranges” aconselha usar “Table 42-3, Sensor Type Codes” para decodificar os bits. Por exemplo, o sensor 'Power Unit Stat' tem o tipo “type:0x9” que significa “Unidade de força”. O offset 00h significa “PowerOff/Desligado”. Em outras palavras, o bit menos significativo é 1 se o servidor estiver desligado. Para testar este bit, a função `band**` com a máscara 1 poderá ser utilizada. Uma expressão de trigger poderia ser:

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

para avisar sobre problema de perda de força no servidor.

Observações sobre nomes de sensores discretos no OpenIPMI-2.0.16, 2.0.17, 2.0.18 e 2.0.19

Os nomes de sensores discretos no OpenIPMI-2.0.16, 2.0.17 e 2.0.18 often possuem um “0” adicional (ou outro dígito ou letra) adicionado ao final. Por exemplo, enquanto `ipmitool` e OpenIPMI-2.0.19 apresenta os nomes de sensores “PhysicalSecurity” ou “CATERR”, no OpenIPMI-2.0.16, 2.0.17 e 2.0.18 estes nomes são “PhysicalSecurity0” ou “CATERR0”, respectivamente.

Quando configurar um item IPMI no Zabbix Server para monitorar um dispositivo com OpenIPMI-2.0.16, 2.0.17 e 2.0.18, utilize estes nomes terminados com “0” no campo *Sensor IPMI* dos itens do tipo Agente IPMI. Quando seu Servidor Zabbix for atualizado para uma nova distribuição Linux, que utilize uma versão superior do OpenIPMI, tais itens passarão para o estado de “NÃO SUPORTADO”. Você terá que alterar o campo *Sensor IPMI* removendo o '0' no final e aguardar algum tempo para que a monitoração fique operacional novamente neles.

Observações sobre limites e sensores discretos

Alguns agentes IPMI proveem tanto sensores de limite quanto sensores discretos com o mesmo nome. No Zabbix anterior ao 2.2.8 e 2.4.3, o primeiro sensor que for localizado será utilizado. A partir das versões 2.2.8 e 2.4.3, a preferência sempre será para o sensor de limite.

Observações sobre finalização de conexão

Se a verificação IPMI não for executada (por qualquer razão: todos os itens IPMI com o estado desabilitado/não suportado, host desabilitado/excluído, host em manutenção etc.) a conexão IPMI será terminada no Zabbix Server/Proxy de 3 a 4 horas dependendo do momento em que o Zabbix Server/Proxy foi iniciado.

From:
<https://www.zabbix.com/documentation/3.0/> - **Zabbix Documentation 3.0**

Permanent link:
<https://www.zabbix.com/documentation/3.0/pt/manual/config/items/itemtypes/ipmi>

Last update: **2016/02/17 12:43**

