

Documentação de Configuração do Zabbix Agent para Coleta de Métricas Hyper-V

Autor: Nilo Trigueiro – www.linkedin.com/in/nilotrigueiro

Introdução

Esta documentação detalha a configuração necessária do Zabbix Agent para coletar métricas avançadas de ambientes Hyper-V. Os códigos fornecidos permitem a monitorização de VMs, uso de memória e disco, tempo de atividade e informações do hardware físico, além de possibilitar a descoberta automática de VMs (Low-Level Discovery – LLD).

1. Inserção dos Códigos no `zabbix_agentd.conf`

Para que as coletas de métricas funcionem corretamente, os códigos a seguir devem ser adicionados ao arquivo de configuração do Zabbix Agent (`zabbix_agentd.conf`).
Recomendo inserir ao final do arquivo para melhor organização e manutenção.

```
### Parâmetro alterado e documentado por Nilo em 28-05-2025

# -----
#  Configuração básica
# -----

# Permite que o agente Zabbix execute comandos personalizados (como scripts
PowerShell)
UnsafeUserParameters=1

# -----
#  Métricas gerais de VMs Hyper-V
# -----

# Retorna o número total de máquinas virtuais configuradas no Hyper-V
UserParameter=hyperv.vm.total,powershell -Command "(Get-VM).Count"

# Retorna o número de VMs atualmente ligadas (estado = 'Running')
UserParameter=hyperv.vm.running,powershell -Command "(Get-VM | Where-
```

```
Object { $_.State -eq 'Running' }).Count"
```

```
# Retorna o número de VMs atualmente desligadas (estado = 'Off')
```

```
UserParameter=hyperv.vm.off,powershell -Command "(Get-VM | Where-Object { $_.State -eq 'Off' }).Count"
```

```
# -----
```

```
# Métricas de memória por VM
```

```
# -----
```

```
# Retorna a quantidade de memória atribuída à VM especificada (em GB)
```

```
UserParameter=hyperv.vm.memory_assigned[*],powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\hyperv_memory_assigned.ps1" -vm "$1"
```

```
# Retorna a quantidade de memória usada pela VM (em tempo real, em GB)
```

```
UserParameter=hyperv.vm.memory_used[*],powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\hyperv_memory_used.ps1" -vm "$1"
```

```
# -----
```

```
# Métricas de disco por VM
```

```
# -----
```

```
# Retorna a soma do tamanho total de disco alocado para a VM (em GB)
```

```
UserParameter=hyperv.vm.disk_total[*],powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\hyperv_disk_total.ps1" -vm "$1"
```

```
# Retorna o espaço de disco atualmente usado pela VM (em bytes; requer VM ligada e Integration Services ativos)
```

```
UserParameter=hyperv.vm.disk_used[*],powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\hyperv_disk_used.ps1" -vm "$1"
```

```
# -----
```

```
# Tempo de atividade por VM
```

```
# -----
```

```
# Retorna o tempo de atividade da VM em segundos
```

```
UserParameter=hyperv.vm.uptime[*],powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\hyperv_uptime.ps1" "$1"
```

```
# -----
```

```
# Descoberta automática (LLD)
```

```
# -----
```

```
# Executa um script PowerShell que retorna um JSON com as VMs para LLD no Zabbix
```

```
UserParameter=hyperv.vms.discovery,powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\hyperv_vms_discovery.ps1"
```

```

# -----
#   Status de execução da VM
# -----

# Retorna o status atual da VM (Running = 1, Off = 0, Paused = 2, outros = 99)
UserParameter=hyperv.vm.status[*],powershell -NoProfile -ExecutionPolicy Bypass
-File "C:\Scripts\hyperv_vm_status.ps1" -vm "$1"

# -----
#   Informações do host físico
# -----

# Retorna dados do servidor físico: modelo, fabricante, memória total, CPU, etc.
UserParameter=server.hardware.info,powershell -NoProfile -ExecutionPolicy
Bypass -File "C:\Scripts\get_server_hardware_info.ps1"

# Retorna detalhes dos módulos de memória física: quantidade, tipo (DDR),
tamanho e velocidade
UserParameter=custom.memory.info,powershell -NoProfile -ExecutionPolicy
Bypass -File "C:\Scripts\GetMemoryInfo.ps1"

# -----
#   Processadores atribuídos às VMs
# -----

# Retorna o número de vCPUs atribuídas a cada VM
UserParameter=hyperv.vm.cpu.count,powershell -NoProfile -ExecutionPolicy
Bypass -File "C:\Scripts\hyperv_cpu_count.ps1"

```

2. Caminhos dos Scripts e Nomenclatura

Os caminhos dos arquivos de script PowerShell nos `UserParameter` foram definidos como `C:\Scripts`. Esta é uma escolha do autor para organização. Caso o ambiente de destino utilize um diretório diferente para armazenar os scripts, é **fundamental** que os caminhos sejam ajustados correspondentemente no arquivo `zabbix_agentd.conf`.

Sugiro manter os nomes dos arquivos conforme apresentados nos códigos (`hyperv_memory_assigned.ps1` , `hyperv_vms_discovery.ps1` , etc.), pois eles são descritivos e fazem sentido em relação às métricas que coletam, facilitando a identificação e manutenção.

3. Permissões do Zabbix Agent

Para garantir a coleta completa e correta de todas as métricas, especialmente aquelas que interagem com o Hyper-V e o sistema operacional, é crucial que o serviço do Zabbix Agent seja executado com uma conta que possua os privilégios necessários.

Recomenda-se configurar o Zabbix Agent para **fazer logon como um serviço (Log On As Service)** utilizando uma conta de usuário com permissões administrativas ou, no mínimo, com privilégios suficientes para acessar as informações do Hyper-V e executar scripts PowerShell.

4. Próximos Passos

Com a configuração do Zabbix Agent concluída, o próximo passo é trabalhar na interface do Zabbix para criar ou importar uma **template personalizada**. Esta template será responsável por definir os itens de coleta, triggers, gráficos e regras de descoberta de baixo nível (LLD) que utilizarão os `UserParameter` configurados neste documento para visualizar e alertar sobre as métricas do Hyper-V.

Local de Configuração

Para ativar as coletas, adicione os seguintes parâmetros personalizados no final do arquivo `zabbix_agentd.conf`, localizado normalmente em:

- **Windows:** `C:\Program Files\Zabbix Agent\zabbix_agentd.conf` (ou o diretório de instalação)

Permissões Necessárias

Para que os scripts funcionem corretamente, é essencial que o Zabbix Agent:

- Seja executado com uma conta de serviço com privilégios administrativos, preferencialmente com permissão de:
 - Acesso ao Hyper-V (`Get-VM` , `Get-VHD` , etc.)
 - Leitura de WMI (`Win32_ComputerSystem` , `Win32_PhysicalMemory` , etc.)
- Esteja configurado como:
 - "Logon as a service" nas propriedades do serviço do agente

- Com permissão de execução para scripts PowerShell (ExecutionPolicy Bypass)

Teste Local (Opcional)

Você pode testar localmente qualquer script antes de integrar ao Zabbix com:

```
powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Scripts\nome_do_script.ps1"  
-vm "NomeDaVM"
```

Próximo Passo: Integração no Zabbix (Frontend)

Com os parâmetros configurados no agente, o próximo passo é:

- Criar um template personalizado no Zabbix.
- Definir os itens de coleta usando as chaves definidas (hyperv.vm.memory_assigned , hyperv.vm.status , etc.).
- Configurar a LLD (descoberta automática) usando a chave hyperv.vms.discovery .
- Associar o template ao host correspondente no Zabbix.

Se desejar, posso gerar o template XML pronto para importação no Zabbix com os itens, triggers e discovery.

Conclusão

Com esses ajustes, seu Zabbix será capaz de coletar métricas detalhadas de servidores Hyper-V e suas VMs, incluindo:

- Estado das VMs
- Uso de CPU, memória e disco
- Informações do hardware físico
- Descoberta automática de VMs

Tudo isso usando scripts customizados, reutilizáveis e organizados, com desempenho e segurança controlados.