

# **Zabbix Advanced**

## **Aula 09: Zabbix Proxy e Monitoramento Distribuído**

**Arquitetura, Instalação, Configuração e Troubleshooting**

**4Linux - Curso Avançado**

# Agenda do Dia

## 1. Fundamentos do Zabbix Proxy

- O que é, quando usar, arquitetura

## 2. Active vs Passive Proxy

- Diferenças, casos de uso, trade-offs

## 3. Instalação e Configuração

- Passo a passo, banco de dados, integração

# Agenda do Dia (cont.)

## 4. Associação de Hosts ao Proxy

- Via GUI, via API, estratégias

## 5. Sincronização e Performance

- Cache, buffers, troubleshooting

## 6. Estratégias para Ambientes Distribuídos

- Filiais, cloud, DMZ

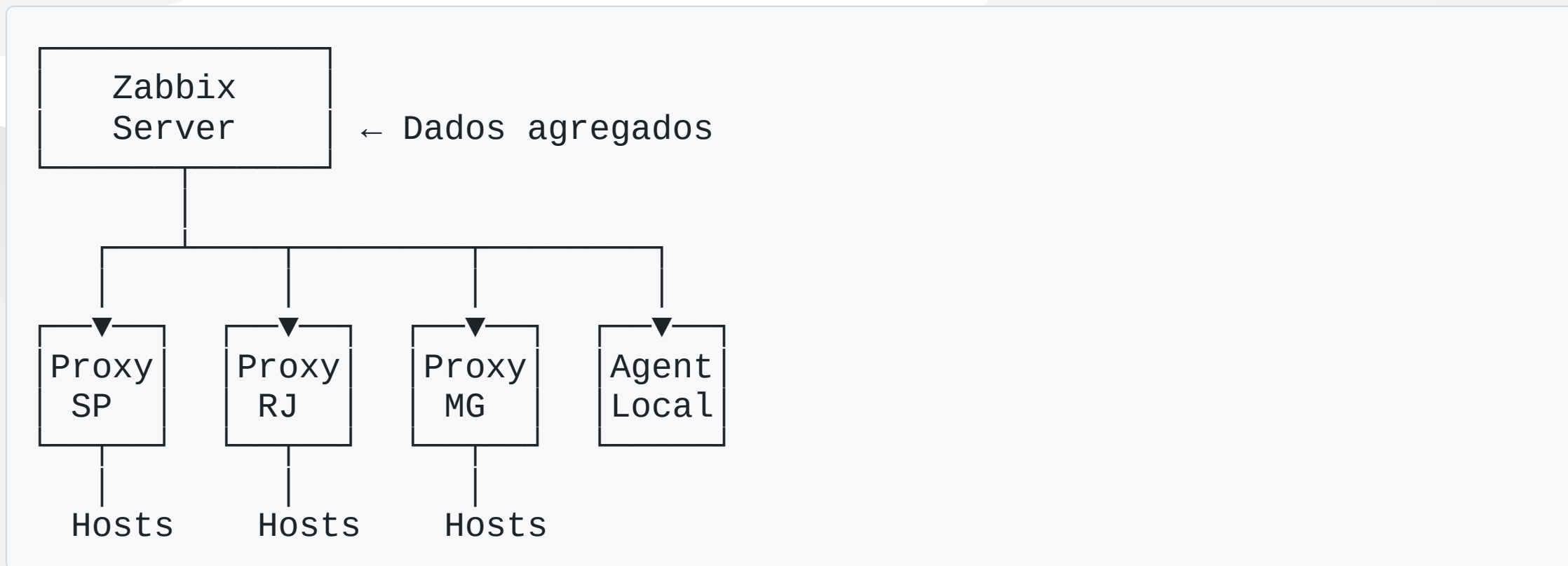
## 7. Laboratórios Práticos

# PARTE 1

## Fundamentos do Zabbix Proxy

# O Que É Zabbix Proxy?

**Proxy** = Coletor intermediário entre agents e Zabbix Server



**Função:** Coletar dados localmente e enviar ao Server central

# Por Que Usar Zabbix Proxy?

## Casos de uso principais:

### 1. Monitoramento distribuído geograficamente

- Filiais, escritórios remotos, datacenters regionais

### 2. Reduzir latência de rede

- Proxy local → coleta rápida
- Envio ao Server → batch periódico

### 3. Confiabilidade

- Proxy continua coletando se link com Server cair
- Buffer local de dados

### 4. Segurança/DMZ

- Apenas Proxy exposto, agents internos

# Quando NÃO Usar Proxy

## ✗ Não usar proxy se:

- Poucos hosts (< 100) na mesma rede do Server
- Latência já é baixa (< 10ms)
- Complexidade adicional não justifica benefício
- Orçamento de hardware limitado

## ✓ Usar Server direto:

- Ambiente pequeno/médio (< 500 hosts)
- Hosts na mesma LAN
- Administração centralizada simples



# Arquitetura: Server vs Proxy

| Componente                | Zabbix Server               | Zabbix Proxy                    |
|---------------------------|-----------------------------|---------------------------------|
| Banco de dados            | MySQL/PostgreSQL (completo) | SQLite/MySQL/PostgreSQL (cache) |
| Frontend                  | ✓ Sim (interface web)       | ✗ Não                           |
| Coleta de dados           | ✓ Sim                       | ✓ Sim                           |
| Processar triggers        | ✓ Sim                       | ✗ Não                           |
| Enviar alertas            | ✓ Sim                       | ✗ Não                           |
| Armazenamento longo prazo | ✓ Sim                       | ✗ Não                           |

# Fluxo de Dados com Proxy

1. Agent coleta métrica (ex: CPU 80%)  
↓
2. Proxy recebe dado e armazena em cache local (SQLite)  
↓
3. Proxy envia dados ao Server (batch a cada X segundos)  
↓
4. Server armazena em DB, processa triggers  
↓
5. Server gera alerta se trigger disparar  
↓
6. Server envia notificação via action

**Importante:** Alertas sempre vêm do Server, nunca do Proxy

# Vantagens do Proxy

## ✓ Performance:

- Coleta local rápida (baixa latência)
- Reduz carga no Zabbix Server

## ✓ Confiabilidade:

- Buffer local de até N horas de dados
- Sobrevive a falhas temporárias de rede

### ✓ Segurança:

- Proxy em DMZ, Server em rede interna
- Reduz exposição de agents

### ✓ Escalabilidade:

- Cada proxy pode monitorar milhares de hosts
- Server central agrega múltiplos proxies

# Desvantagens do Proxy

## ✗ Complexidade:

- Mais servidores para gerenciar
- Configuração adicional

## ✗ Custo:

- Hardware/VMs adicionais
- Manutenção extra

## ✗ Latência de alertas:

- Delay entre coleta (proxy) e alerta (server)
- Depende do intervalo de sincronização

## ✗ Troubleshooting:

- Mais pontos de falha
- Logs em múltiplas máquinas

# Quantos Hosts por Proxy?

## Capacidade típica:

| Configuração Proxy | Hosts Monitorados | NVPS*   |
|--------------------|-------------------|---------|
| 2 CPU, 4GB RAM     | ~500              | ~5,000  |
| 4 CPU, 8GB RAM     | ~2,000            | ~20,000 |
| 8 CPU, 16GB RAM    | ~5,000            | ~50,000 |

**NVPS = New Values Per Second (valores novos/segundo)**

## Fatores que afetam:

- Intervalo de coleta (1m vs 5m)
- Tipo de items (agent vs SNMP vs IPMI)
- Network bandwidth disponível

# PARTE 2

## Active vs Passive Proxy



# Active Proxy vs Passive Proxy

Diferença fundamental: QUEM inicia a conexão?

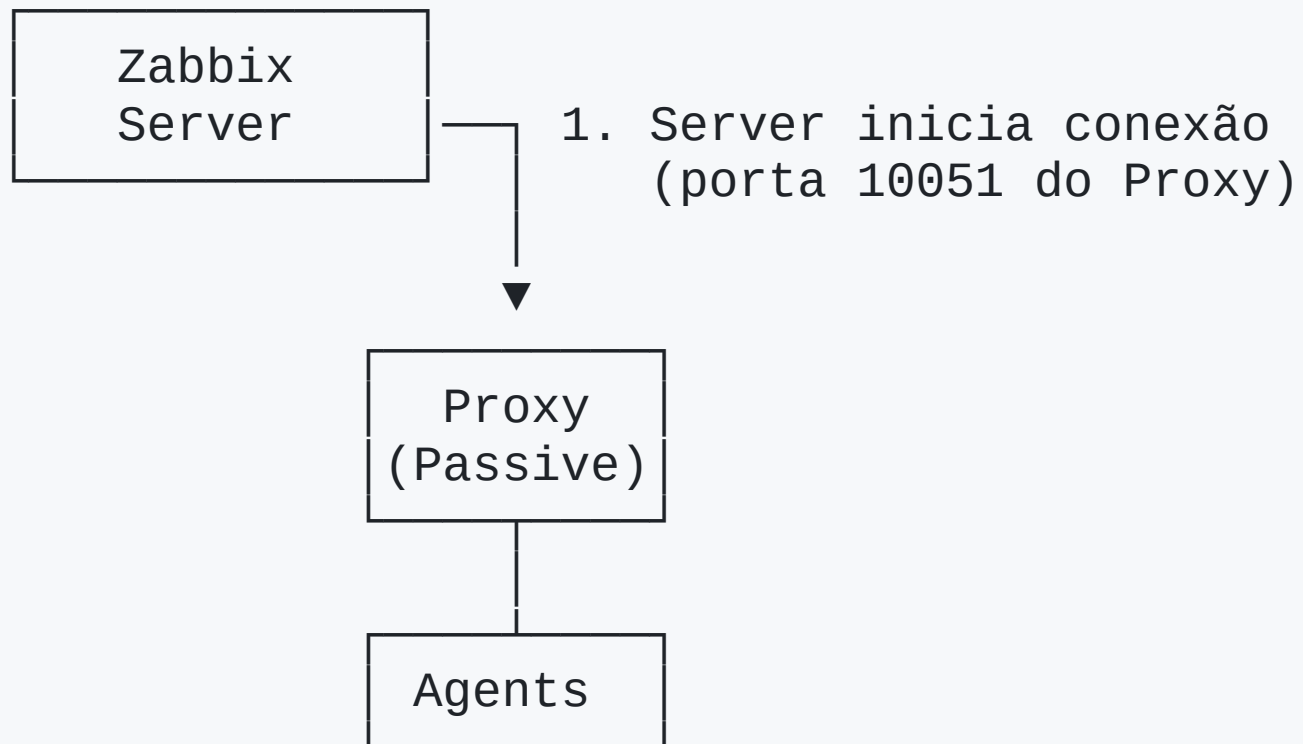
## Passive Proxy (padrão):

- Server conecta no Proxy (polling)
- Server: "Proxy, me envie os dados"
- Proxy: "Aqui estão"

## Active Proxy:

- Proxy conecta no Server (push)
- Proxy: "Server, tenho dados novos"
- Server: "OK, recebido"

# Passive Proxy: Arquitetura



**Configuração:** `ProxyMode=0` (passive)

# Passive Proxy: Quando Usar

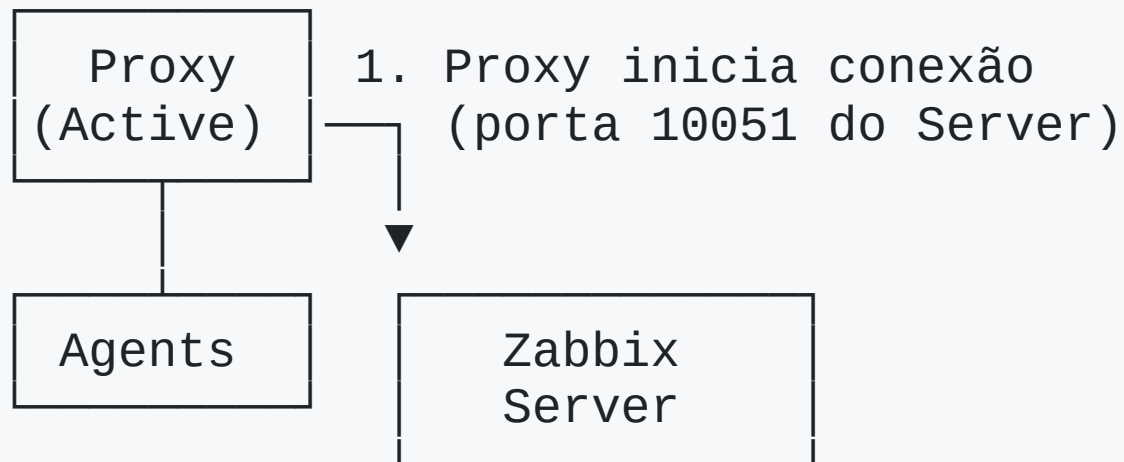
## ✓ Usar Passive Proxy quando:

- Proxy está em rede acessível do Server
- Firewall permite Server → Proxy (porta 10051)
- Controle centralizado (Server "puxa" dados)
- Ambiente corporativo tradicional

**Exemplo:** Filial com link dedicado ao datacenter central

```
Datacenter Central (Server)
      ↓ (conecta)
Filial SP (Proxy Passive)
```

# Active Proxy: Arquitetura



**Configuração:** `ProxyMode=1` (active)

# Active Proxy: Quando Usar

## ✓ Usar Active Proxy quando:

- Proxy está atrás de NAT/firewall
- Server NÃO consegue conectar no Proxy
- Proxy em cloud pública (IP dinâmico)
- Múltiplos proxies remotos (escritórios home office)

## Exemplo: Proxy em AWS com IP público dinâmico

```
AWS EC2 (Proxy Active, NAT)
      ↓ (conecta)
Datacenter On-Premise (Server)
```

# Comparação: Active vs Passive

| Aspecto      | Passive Proxy                 | Active Proxy                  |
|--------------|-------------------------------|-------------------------------|
| Quem conecta | Server → Proxy                | Proxy → Server                |
| Porta aberta | 10051 no Proxy                | 10051 no Server               |
| NAT/Firewall | Server precisa alcançar Proxy | Proxy precisa alcançar Server |
| Controle     | Server puxa dados             | Proxy empurra dados           |
| IP dinâmico  | ❌ Não funciona                | ✅ Funciona                    |
| Complexidade | ★ Simples                     | ★★ Moderada                   |
| Latência     | Depende do polling            | Mais imediato                 |

# Configurando Passive Proxy

No arquivo `/etc/zabbix/zabbix_proxy.conf` :

```
# Modo Passive
ProxyMode=0

# Endereço do Zabbix Server (para receber conexões de configuração)
Server=192.168.1.10

# Porta do proxy (Server conectará aqui)
ListenPort=10051

# Hostname (deve coincidir com o cadastro no Server)
Hostname=proxy-sp-01

# Banco de dados
DBName=/var/lib/zabbix/zabbix_proxy.db
DBUser=zabbix
```

# Configurando Active Proxy

No arquivo `/etc/zabbix/zabbix_proxy.conf` :

```
# Modo Active
ProxyMode=1

# Endereço do Zabbix Server (Proxy conectará aqui)
Server=192.168.1.10

# Porta do Server
ServerPort=10051

# Hostname (deve coincidir com o cadastro no Server)
Hostname=proxy-rj-01

# Intervalo de heartbeat (segundos)
HeartbeatFrequency=60

# Banco de dados
DBName=/var/lib/zabbix/zabbix_proxy.db
```



# Heartbeat: Active Proxy

**Active Proxy envia "heartbeat" periódico ao Server:**

```
HeartbeatFrequency=60 # A cada 60 segundos
```

**Heartbeat contém:**

- "Estou vivo"
- Quantidade de dados em buffer
- Estatísticas de performance

## **Server usa heartbeat para:**

- Detectar proxy offline
- Monitorar saúde do proxy
- Trigger automática se proxy parar de responder

# PARTE 3

## Instalação e Configuração

# Pré-requisitos

## Hardware recomendado (proxy médio):

- CPU: 4 cores
- RAM: 8 GB
- Disco: 50 GB SSD (para banco local)
- Network: 100 Mbps+

## Software:

- SO: Ubuntu 22.04 LTS / RHEL 9 / Debian 12
- Banco: SQLite (padrão) ou MySQL/PostgreSQL
- Firewall: Porta 10051 configurada

# Instalação: Ubuntu 22.04

## 1. Adicionar repositório oficial:

```
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ubuntu22.04_all.deb  
  
sudo dpkg -i zabbix-release_latest+ubuntu22.04_all.deb  
sudo apt update
```

## 2. Instalar Zabbix Proxy:

```
# Com SQLite (recomendado para < 1000 hosts)  
sudo apt install zabbix-proxy-sqlite3  
  
# OU com MySQL (para > 1000 hosts)  
sudo apt install zabbix-proxy-mysql zabbix-sql-scripts
```

# Configuração: SQLite (Simples)

## 3. Editar configuração:

```
sudo nano /etc/zabbix/zabbix_proxy.conf
```

## Parâmetros essenciais:

```
Server=192.168.1.10      # IP do Zabbix Server
Hostname=proxy-sp-01     # Nome único
ProxyMode=0              # 0=passive, 1=active

# Banco SQLite (automático)
DBName=/var/lib/zabbix/zabbix_proxy.db

# Timeout e buffers
Timeout=4
LogSlowQueries=3000
```

# Configuração: MySQL (Performance)

## 3a. Criar banco de dados:

```
sudo mysql -u root -p
```

```
CREATE DATABASE zabbix_proxy CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;  
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'password';  
GRANT ALL PRIVILEGES ON zabbix_proxy.* TO 'zabbix'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

## 3b. Importar schema:

```
zcat /usr/share/zabbix-sql-scripts/mysql/proxy.sql.gz | \  
mysql -u zabbix -p zabbix_proxy
```

# Configuração: Proxy com MySQL

## 3c. Configurar proxy:

```
Server=192.168.1.10  
Hostname=proxy-sp-01  
ProxyMode=0
```

```
# MySQL
```

```
DBHost=localhost  
DBName=zabbix_proxy  
DBUser=zabbix  
DBPassword=password
```

```
# Cache e buffers (ajustar conforme necessário)
```

```
CacheSize=128M  
HistoryCacheSize=64M  
HistoryIndexCacheSize=32M
```



# Iniciar Zabbix Proxy

## 4. Habilitar e iniciar serviço:

```
sudo systemctl enable zabbix-proxy  
sudo systemctl start zabbix-proxy
```

## 5. Verificar status:

```
sudo systemctl status zabbix-proxy  
  
# Deve mostrar: active (running)
```

## 6. Verificar logs:

```
sudo tail -f /var/log/zabbix/zabbix_proxy.log
```

```
# Procurar por:
```

```
# "server #1 started [main process]"
```

```
# "server #2 started [configuration syncer]"
```

# Cadastrar Proxy no Zabbix Server

## 7. No frontend do Zabbix:

Administration → Proxies → Create proxy

Name: proxy-sp-01

← Deve ser EXATAMENTE igual ao Hostname

Proxy mode: Active

← Ou Passive, conforme configurado

Proxy address: 192.168.1.20

← Apenas para Passive mode  
(vazio para Active)

Encryption:

Connections to proxy: No encryption

Connections from proxy: No encryption

**⚠ IMPORTANTE:** Nome no Server = Hostname no zabbix\_proxy.conf

## Verificar Conexão Proxy Server

8. Aguardar sincronização inicial (~1 minuto)

9. Verificar status:

Administration → Proxies

Proxy: proxy-sp-01

Status: (deve mostrar ícone verde)

Last seen: < 1 minute ago ← Confirma conexão OK

Hosts: 0 ← Ainda sem hosts associados

Items: 0

Required performance (vps): 0

## Se "Last seen" não atualiza:

- Verificar firewall (porta 10051)
- Verificar logs do proxy e server
- Verificar Hostname matching

# Troubleshooting: Proxy Não Conecta

## Problema 1: Hostname mismatch

```
# No proxy
grep "^Hostname=" /etc/zabbix/zabbix_proxy.conf
# Output: Hostname=proxy-sp-01

# No Server GUI
Administration → Proxies → [Verificar nome exato]
# Deve ser IDÊNTICO (case-sensitive!)
```

## Problema 2: Firewall

```
# No Server, testar conectividade (Passive proxy)
telnet 192.168.1.20 10051

# No Proxy, testar conectividade (Active proxy)
telnet 192.168.1.10 10051
```

# Troubleshooting: Logs do Proxy

Verificar logs em tempo real:

```
sudo tail -f /var/log/zabbix/zabbix_proxy.log
```

Mensagens importantes:

✓ Sucesso:

```
server #1 started [configuration syncer #1]  
received configuration data from server
```

✗ Falha (Passive):

```
cannot send list of active checks to [192.168.1.10]: host [proxy-sp-01] not found
```

✗ Falha (Active):

# **PARTE 4**

## **Associação de Hosts ao Proxy**



# Associando Host ao Proxy (GUI)

## Opção 1: Host novo

Data collection → Hosts → Create host

Host name: web-sp-01

Groups: Web servers

Interfaces:

Agent: 192.168.10.100:10050

Monitored by proxy: proxy-sp-01 ← SELECIONAR AQUI

Templates: Linux by Zabbix agent

**Host agora será monitorado via proxy, não direto pelo Server**

# Associando Host Existente ao Proxy

## Opção 2: Migrar host existente

Data collection → Hosts → [Selecionar host] → Edit

Monitored by proxy: (no proxy) → proxy-sp-01

↓  
Salvar

### Efeito:

- Próxima coleta será via proxy
- Server para de coletar diretamente
- Histórico anterior é mantido

# Associação em Massa (GUI)

Para múltiplos hosts:

Data collection → Hosts

1. Filtrar hosts (ex: grupo "Filial SP")
2. Selecionar múltiplos (checkbox)
3. Mass update
4. Proxy: proxy-sp-01
5. Update

**Resultado:** Todos os hosts selecionados migrados para o proxy

# Associação via API

**Criar host via API já associado ao proxy:**

```
#!/bin/bash
# create_host_with_proxy.sh

curl -X POST https://zabbix.example.com/api_jsonrpc.php \
-H "Content-Type: application/json-rpc" \
-d '{
  "jsonrpc": "2.0",
  "method": "host.create",
  "params": {
    "host": "web-sp-02",
    "groups": [{"groupid": "2"}],
    "interfaces": [{
      "type": 1,
      "main": 1,
      "ip": "192.168.10.101",
      "dns": "",
      "port": "10050"
    }],
    "proxy_hostid": "10001" ← ID do proxy (ver abaixo)
  },
  "auth": "AUTH_TOKEN",
  "id": 1
}'
```

# Descobrir Proxy ID (API)

Listar proxies para obter proxy\_hostid:

```
curl -X POST https://zabbix.example.com/api_jsonrpc.php \
-H "Content-Type: application/json-rpc" \
-d '{
  "jsonrpc": "2.0",
  "method": "proxy.get",
  "params": {
    "output": ["proxyid", "host"]
  },
  "auth": "AUTH_TOKEN",
  "id": 1
}'
```

## Resposta:

```
{  
  "result": [  
    {"proxyid": "10001", "host": "proxy-sp-01"},  
    {"proxyid": "10002", "host": "proxy-rj-01"}  
  ]  
}
```

# Estratégia: Proxy por Grupo

## Organização lógica:

```
Host Group: Filial SP → Proxy: proxy-sp-01  
Host Group: Filial RJ → Proxy: proxy-rj-01  
Host Group: Filial MG → Proxy: proxy-mg-01  
Host Group: Datacenter → (no proxy, Server direto)
```

## Vantagem:

- Fácil identificar qual proxy monitora o quê
- Relatórios por região
- Troubleshooting segmentado

# Verificar Hosts no Proxy

## No frontend:

Administration → Proxies

Proxy: proxy-sp-01

Hosts: 150 ← Quantidade de hosts

Items: 12,450 ← Quantidade de items

Required performance: 207 vps ← NVPS esperado

## Clicar no nome do proxy → aba "Hosts":

- Lista completa de hosts monitorados
- Status de cada host
- Templates aplicados



# **PARTE 5**

## **Sincronização e Performance**

# Como Funciona a Sincronização

## Fluxo de configuração (Server → Proxy):

1. Admin altera config no Server (adiciona item, trigger, etc)  
↓
2. Server marca config como "dirty"  
↓
3. Proxy solicita config (ProxyConfigFrequency=3600s)  
↓
4. Server envia configuração completa  
↓
5. Proxy aplica nova configuração

## Fluxo de dados (Proxy → Server):

1. Proxy coleta dados dos agents  
↓
2. Armazena em cache local (SQLite/MySQL)  
↓
3. Envia batch ao Server (ProxyDataFrequency=1s)  
↓
4. Server armazena em DB principal

# Parâmetros de Sincronização

No `zabbix_proxy.conf`:

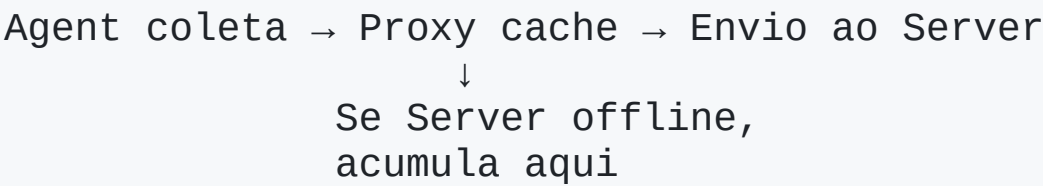
```
# Frequência de sincronização de configuração (segundos)
ProxyConfigFrequency=3600      # Padrão: 1 hora
                                # Menor = config mais atualizada
                                # Maior = menos carga

# Frequência de envio de dados (segundos)
ProxyDataFrequency=1           # Padrão: 1 segundo
                                # Menor = alertas mais rápidos
                                # Maior = menos carga de rede

# Modo offline: quanto tempo manter dados em cache
ProxyOfflineBuffer=1           # Padrão: 1 hora (em horas)
                                # Máximo: 720 (30 dias)
```

# Cache e Buffers: Conceitos

## Cache local do proxy:



## Tipos de cache:

| Cache                 | Função                      | Tamanho Padrão |
|-----------------------|-----------------------------|----------------|
| ConfigCacheSize       | Configuração (hosts, items) | 32M            |
| HistoryCacheSize      | Dados históricos            | 16M            |
| HistoryIndexCacheSize | Índice do histórico         | 4M             |
| TrendCacheSize        | Trends (não usado em proxy) | N/A            |

# Ajustando Cache (Alta Performance)

Para proxy com 2000+ hosts:

```
# /etc/zabbix/zabbix_proxy.conf

# Configuração de hosts e items
ConfigCacheSize=128M          # Padrão: 32M

# Buffer de dados históricos
HistoryCacheSize=256M        # Padrão: 16M

# Índice para acesso rápido
HistoryIndexCacheSize=64M    # Padrão: 4M

# Database cache (se usar MySQL)
CacheSize=256M               # Padrão: 8M

# Timeout de queries
Timeout=10                   # Padrão: 3s
DBSocket=/var/run/mysqld/mysqld.sock
```

# Processos do Proxy: Tuning

Ajustar quantidade de processos paralelos:

```
# Pollers: coletam dados dos agents
StartPollers=10                # Padrão: 5

# IPMI pollers
StartIPMIPollers=2             # Padrão: 0

# Pollers unreachable: hosts down
StartPollersUnreachable=3     # Padrão: 1

# Trappers: recebem dados (agent active)
StartTrappers=10              # Padrão: 5

# HTTP pollers: HTTP agent items
StartHTTPPollers=5            # Padrão: 1
```

# Monitorar Performance do Proxy

## Item interno do Zabbix:

Data collection → Hosts → Zabbix server (ou criar template)

Items internos úteis:

`zabbix[proxy,<proxy_name>,delay]`

→ Delay médio entre coleta (proxy) e recebimento (server)

`zabbix[proxy,<proxy_name>,lastaccess]`

→ Timestamp da última comunicação



## Criar trigger:

Name: Proxy delay is high

Expression:

```
last(/Zabbix server/zabbix[proxy,proxy-sp-01,delay])>300
```

Severity: Warning

Description: Delay > 5 minutos entre proxy e server

# Proxy: Interno Self-Monitoring

## Proxy pode se auto-monitorar:

Data collection → Hosts → Create host

Host name: proxy-sp-01-self

Interfaces: Agent 127.0.0.1:10050

Monitored by proxy: (no proxy) ← Server monitora diretamente

Templates: Zabbix proxy health

## Métricas coletadas:

- Utilização de cache
- Queue de items
- CPU/RAM do proxy
- Espaço em disco (banco SQLite/MySQL)

# Troubleshooting: Delay Alto

**Sintoma:** Dados demoram muito para aparecer no Server

**Diagnóstico:**

```
# 1. Verificar queue do proxy  
tail -f /var/log/zabbix/zabbix_proxy.log | grep queue  
  
# Procurar por:  
# "history queue size: 15000" ← Alto = problema
```

## Causas comuns:

1. **ProxyDataFrequency muito alto** → Reduzir para 1
2. **Network bandwidth insuficiente** → Aumentar link
3. **Cache cheio** → Aumentar HistoryCacheSize
4. **Server lento** → Otimizar Server (outro tópico)

# Troubleshooting: Proxy Offline

**Sintoma:** Proxy perde conexão com Server

**O que acontece:**

1. Proxy detecta que Server está inacessível  
↓
2. Continua coletando dados dos agents  
↓
3. Armazena em cache local (ProxyOfflineBuffer)  
↓
4. Quando Server volta, envia dados acumulados

**Limite:** ProxyOfflineBuffer=1 (hora) → Dados mais antigos são descartados

**Aumentar buffer:**

# PARTE 6

## Estratégias para Ambientes Distribuídos

# Cenário 1: Filiais Geográficas

## Problema:

- Empresa com 10 filiais
- Cada filial: 50-200 hosts
- Link WAN lento (10 Mbps)

## Solução:

Datacenter Central (Server)



```
graph TD; DC[Datacenter Central (Server)] --> P1[Proxy Filial SP (200 hosts)]; DC --> P2[Proxy Filial RJ (150 hosts)]; DC --> P3[Proxy Filial MG (100 hosts)]; DC --> P4[Proxy Filial BA (80 hosts)]; DC --> P5[...];
```

Proxy Filial SP (200 hosts)  
Proxy Filial RJ (150 hosts)  
Proxy Filial MG (100 hosts)  
Proxy Filial BA (80 hosts)  
...

## Vantagens:

- Coleta local rápida (LAN)
- Envia apenas ao Server comprimido
- Sobrevive a queda de link WAN



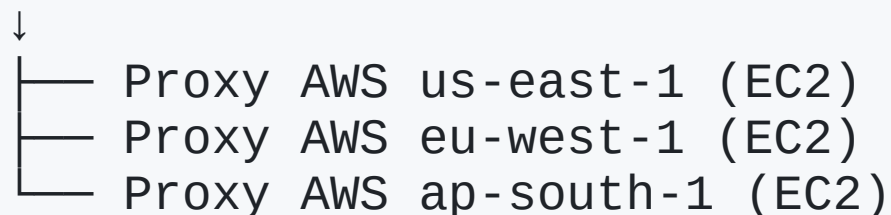
## Cenário 2: Cloud Multi-Region

### Problema:

- Aplicação distribuída em AWS (us-east, eu-west, ap-south)
- Latência alta entre regiões
- Custo de transferência cross-region

## Solução:

Datacenter On-Premise (Server)



## Configuração:

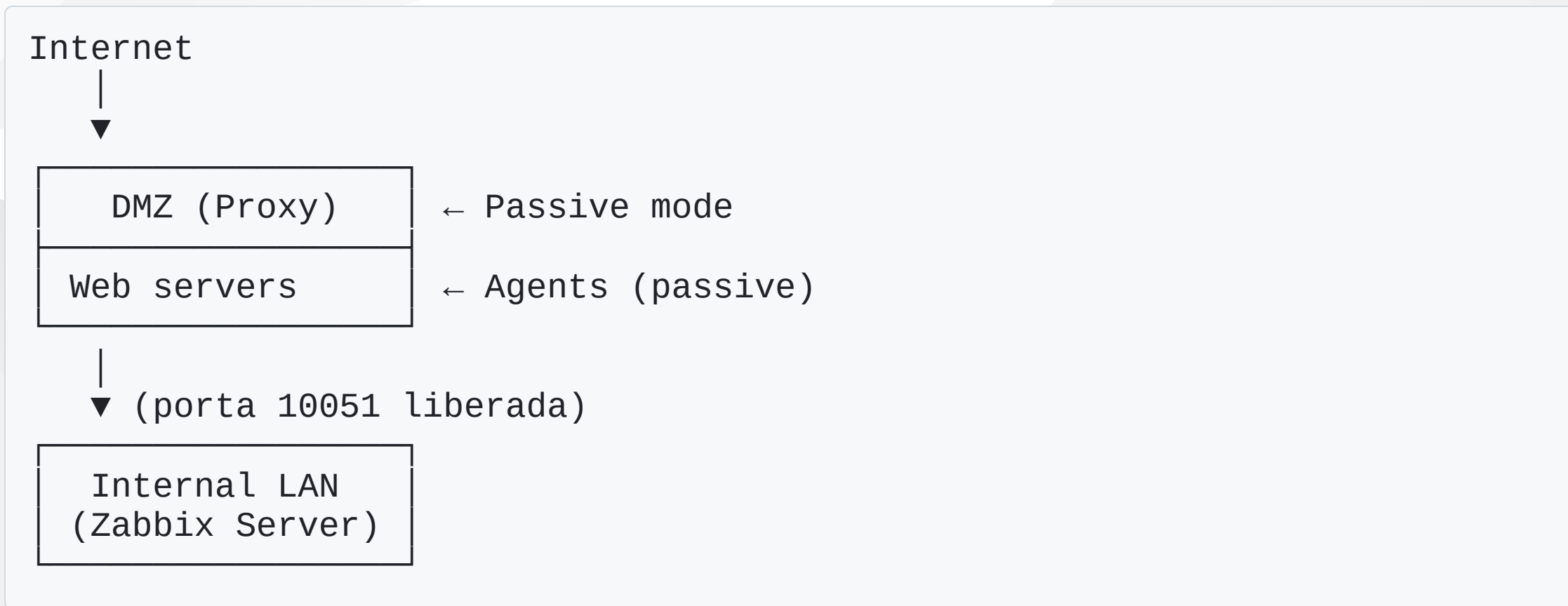
- Proxies em Active mode (NAT friendly)
- Cada proxy monitora instâncias EC2 da sua região
- Reduz cross-region traffic

## Cenário 3: DMZ e Segurança

### Problema:

- Monitorar servidores web em DMZ
- Firewall impede Server acessar DMZ diretamente
- Security team não autoriza agents abrirem porta para fora

## Solução:



## Cenário 4: Monitoramento Híbrido

### Problema:

- 80% hosts on-premise
- 20% hosts em cloud pública (Azure, AWS, GCP)
- Política: separar monitoramento por ambiente

## Solução:

Zabbix Server (On-Premise)



- Direct monitoring (on-premise hosts)
- Proxy Azure (active mode)
- Proxy AWS (active mode)
- Proxy GCP (active mode)

## Tag strategy:

- Tag: `environment:onprem`
- Tag: `environment:azure`
- Dashboards filtrados por tag

## Cenário 5: Clientes/Tenants Isolados

### Problema:

- MSP (Managed Service Provider) com múltiplos clientes
- Cada cliente quer isolamento
- Não quer compartilhar Server

## Solução:

Zabbix Server (MSP Datacenter)



- ├── Proxy Cliente A (rede do cliente)
- ├── Proxy Cliente B (rede do cliente)
- └── Proxy Cliente C (rede do cliente)

## Isolamento via:

- User groups separados
- Permissões por host group
- Proxies dedicados por cliente



# High Availability: Failover de Proxy

**Problema:** Proxy único = single point of failure

**Solução (Zabbix 7.0 não tem HA nativo de proxy):**

**Opção 1: Proxy redundante manual**

Filial SP:

- └─ Proxy SP-01 (principal) → 100 hosts
- └─ Proxy SP-02 (backup) → mesmos 100 hosts (desabilitado)

Em caso de falha: Habilitar SP-02 manualmente

## Opção 2: Keepalived + Virtual IP

```
VIP 192.168.1.99 (proxy-sp-vip)
├─ Proxy SP-01 (MASTER)
└─ Proxy SP-02 (BACKUP)
```

Keepalived gerencia failover automático

# Estimativa de Bandwidth: Proxy Server

## Fórmula aproximada:

$$\text{Bandwidth (Kbps)} = \text{NVPS} \times 100 \text{ bytes} \times 8 \text{ bits/byte} \div 1000$$

Exemplo:

$$1000 \text{ NVPS} \times 100 \times 8 \div 1000 = 800 \text{ Kbps } (\sim 1 \text{ Mbps})$$

## Recomendações:

| NVPS   | Bandwidth Recomendado |
|--------|-----------------------|
| 500    | 0.5 Mbps              |
| 1,000  | 1 Mbps                |
| 5,000  | 5 Mbps                |
| 10,000 | 10 Mbps               |

# **PARTE 7**

## **Laboratórios Práticos**

# Lab 1: Instalar Zabbix Proxy (SQLite)

**Objetivo:** Instalar proxy simples com SQLite

**Ambiente:**

- VM Ubuntu 22.04
- IP: 192.168.1.20
- Zabbix Server: 192.168.1.10

## 1. Instalar pacote:

```
wget https://repo.zabbix.com/zabbix/7.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ubuntu22.04_all.deb  
sudo dpkg -i zabbix-release_latest+ubuntu22.04_all.deb  
sudo apt update  
sudo apt install zabbix-proxy-sqlite3 -y
```

# Lab 1: Configuração

## 2. Editar configuração:

```
sudo nano /etc/zabbix/zabbix_proxy.conf
```

### Ajustar:

```
Server=192.168.1.10  
Hostname=proxy-lab-01  
ProxyMode=0  
DBName=/var/lib/zabbix/zabbix_proxy.db  
LogSlowQueries=3000
```

## 3. Iniciar serviço:

```
sudo systemctl enable zabbix-proxy  
sudo systemctl start zabbix-proxy  
sudo systemctl status zabbix-proxy
```

# Lab 1: Cadastrar no Server

## 4. No frontend (192.168.1.10):

Administration → Proxies → Create proxy

Name: proxy-lab-01

Proxy mode: Passive

Proxy address: 192.168.1.20

Port: 10051

## 5. Verificar conexão:

Administration → Proxies

proxy-lab-01

Status: (ícone verde)

Last seen: < 1 min

# Lab 2: Associar Host ao Proxy

**Objetivo:** Migrar host existente para proxy

## 1. Identificar host:

Data collection → Hosts

Host: web-server-01 (atualmente sem proxy)

## 2. Editar host:

Monitored by proxy: (no proxy) → proxy-lab-01

## 3. Salvar e aguardar 1 minuto

## 4. Verificar coleta:



## Lab 2: Verificar no Proxy

### 5. Verificar logs do proxy:

```
sudo tail -f /var/log/zabbix/zabbix_proxy.log | grep web-server-01
```

### Deve mostrar:

```
starting to poll items  
item "web-server-01:system.cpu.load[percpu,avg1]" became supported
```

### 6. Verificar estatísticas:

Administration → Proxies → proxy-lab-01

Hosts: 1

Items: ~50 (depende do template)

Required performance: ~1 vps

# Lab 3: Proxy Active Mode

**Objetivo:** Configurar proxy em modo ativo (NAT simulation)

## 1. Editar configuração:

```
sudo nano /etc/zabbix/zabbix_proxy.conf
```

## Alterar:

```
ProxyMode=0 → ProxyMode=1      # Active  
Server=192.168.1.10            # Mesmo parâmetro  
HeartbeatFrequency=60          # Adicionar
```

## 2. Reiniciar:

```
sudo systemctl restart zabbix-proxy
```

## Lab 3: Atualizar no Server

### 3. No frontend:

Administration → Proxies → proxy-lab-01 → Edit

Proxy mode: Passive → Active

Proxy address: (apagar, deixar vazio)

### 4. Salvar

### 5. Verificar logs do proxy:

```
sudo tail -f /var/log/zabbix/zabbix_proxy.log
```

## Deve mostrar:

```
sending heartbeat message  
heartbeat sent, heartbeat frequency is 60 sec
```

## Lab 4: Monitorar Performance do Proxy

**Objetivo:** Criar itens para monitorar saúde do proxy

### 1. Criar host "Zabbix Server" (se não existir):

Data collection → Hosts → Create host

Host name: Zabbix Server

Interfaces: (nenhuma, apenas internal items)

Templates: (nenhum)

## 2. Criar item de delay:

Items → Create item

Name: Proxy lab-01 delay

Type: Zabbix internal

Key: zabbix[proxy,proxy-lab-01,delay]

Type of information: Numeric (unsigned)

Units: s

Update interval: 1m

# Lab 4: Trigger de Delay

## 3. Criar trigger:

Name: Proxy lab-01 has high delay

Expression:

```
last(/Zabbix Server/zabbix[proxy,proxy-lab-01,delay])>300
```

Severity: Warning

Description: Delay entre proxy e server > 5 minutos

## 4. Testar:

```
# No proxy, simular sobrecarga  
stress --cpu 4 --timeout 300s
```

```
# Verificar se delay aumenta
```

# Lab 5: Proxy com MySQL

**Objetivo:** Migrar proxy de SQLite para MySQL (performance)

## 1. Instalar MySQL:

```
sudo apt install mariadb-server zabbix-proxy-mysql zabbix-sql-scripts -y
```

## 2. Criar banco:

```
sudo mysql
```

```
CREATE DATABASE zabbix_proxy CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;  
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'StrongPass123';  
GRANT ALL PRIVILEGES ON zabbix_proxy.* TO 'zabbix'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```



# Lab 5: Importar Schema

## 3. Importar schema:

```
zcat /usr/share/zabbix-sql-scripts/mysql/proxy.sql.gz | \
mysql -u zabbix -p zabbix_proxy
# Senha: StrongPass123
```

## 4. Parar proxy SQLite:

```
sudo systemctl stop zabbix-proxy
```

## 5. Backup config antiga:

```
sudo cp /etc/zabbix/zabbix_proxy.conf /etc/zabbix/zabbix_proxy.conf.sqlite.bak
```

# Lab 5: Configurar MySQL

## 6. Editar configuração:

```
sudo nano /etc/zabbix/zabbix_proxy.conf
```

### Alterar:

```
# Comentar SQLite
#DBName=/var/lib/zabbix/zabbix_proxy.db

# Adicionar MySQL
DBHost=localhost
DBName=zabbix_proxy
DBUser=zabbix
DBPassword=StrongPass123

# Aumentar cache
CacheSize=128M
HistoryCacheSize=64M
```

## Lab 5: Iniciar e Validar

### 7. Iniciar proxy com MySQL:

```
sudo systemctl start zabbix-proxy  
sudo systemctl status zabbix-proxy
```

### 8. Verificar logs:

```
sudo tail -f /var/log/zabbix/zabbix_proxy.log
```

### Deve mostrar:

```
database is working with MySQL  
server #1 started [main process]
```

### 9. Verificar conectividade no Server GUI (Last seen < 1 min)

# Lab 6: Tuning de Performance

**Objetivo:** Otimizar proxy para 1000+ hosts

## 1. Criar carga (simular 1000 hosts):

```
# Via API, criar 1000 hosts dummy associados ao proxy  
# (script fornecido separadamente)
```

## 2. Monitorar queue:

```
watch -n 5 'tail -20 /var/log/zabbix/zabbix_proxy.log | grep queue'
```

**Procurar por:**

```
history queue size: 5000 ← Alto = problema
```

# Lab 6: Ajustar Processos

## 3. Se queue alto, editar:

```
sudo nano /etc/zabbix/zabbix_proxy.conf
```

## Aumentar processos:

```
StartPollers=5 → 20  
StartTrappers=5 → 15  
StartHTTPPollers=1 → 5  
  
# Aumentar cache  
HistoryCacheSize=64M → 256M  
ConfigCacheSize=32M → 128M
```

## 4. Reiniciar:

```
sudo systemctl restart zabbix-proxy
```

## Lab 6: Validar Melhoria

### 5. Monitorar queue novamente:

```
watch -n 5 'tail -20 /var/log/zabbix/zabbix_proxy.log | grep queue'
```

### Deve reduzir:

```
history queue size: 500 ← Melhor!
```

### 6. Verificar delay no Server:

```
Item: zabbix[proxy,proxy-lab-01,delay]  
Value: < 60s ← Aceitável
```

# Lab 7: Failover Manual de Proxy

**Objetivo:** Simular falha de proxy e migrar hosts

**Cenário:**

- Proxy principal: proxy-lab-01
- Proxy backup: proxy-lab-02 (instalado previamente)

## 1. Listar hosts do proxy principal:

```
-- No banco do Zabbix Server  
SELECT hostid, host FROM hosts WHERE proxy_hostid = 10001;
```

# Lab 7: Migração em Massa

## 2. Via GUI (massa update):

Data collection → Hosts

Filter:

Monitored by proxy: proxy-lab-01

Select all → Mass update

Monitored by proxy: proxy-lab-02

Update

## 3. Verificar migração:

Administration → Proxies

proxy-lab-01: Hosts: 0 ← Migrados

proxy-lab-02: Hosts: 50 ← Recebeu todos



# Lab 7: Desligar Proxy Antigo

## 4. Desabilitar proxy antigo:

```
# No servidor proxy-lab-01  
sudo systemctl stop zabbix-proxy  
sudo systemctl disable zabbix-proxy
```

## 5. No Server GUI:

Administration → Proxies → proxy-lab-01 → Edit

Status: Disabled

## 6. Verificar que dados continuam chegando via proxy-lab-02

Monitoring → Latest data (hosts migrados)

# PARTE 8

## Troubleshooting Avançado

# Problema 1: Proxy Não Aparece no Server

**Sintoma:** Proxy criado no Server, mas "Never seen"

## Checklist:

### 1. Hostname matching:

```
# No proxy
grep "^Hostname=" /etc/zabbix/zabbix_proxy.conf

# No Server (GUI)
Administration → Proxies → [Nome do proxy]

# Devem ser IDÊNTICOS (case-sensitive)
```

## 2. Firewall:

```
# Passive: Server → Proxy (porta 10051)
telnet 192.168.1.20 10051

# Active: Proxy → Server (porta 10051)
telnet 192.168.1.10 10051
```

# Problema 1: Continuação

## 3. Logs do proxy:

```
sudo tail -100 /var/log/zabbix/zabbix_proxy.log | grep -i error
```

## Erros comuns:

```
cannot connect to [[192.168.1.10]:10051]
```

→ Active mode mas Server inacessível

```
received empty response from server
```

→ Hostname não encontrado no Server

```
database is down
```

→ Problema com SQLite/MySQL local

## Problema 2: Dados Atrasados (High Delay)

**Sintoma:** Dados levam 10+ minutos para aparecer no Server

**Diagnóstico:**

**1. Verificar queue do proxy:**

```
tail -f /var/log/zabbix/zabbix_proxy.log | grep "history queue size"
```

Se > 10000 → Queue alto

## 2. Causas:

- Cache insuficiente → Aumentar HistoryCacheSize
- Poucos pollers → Aumentar StartPollers
- Network lenta → Verificar bandwidth
- Server lento → Otimizar Server (fora do escopo)

## Problema 2: Solução

### 3. Ajustar proxy:

```
# /etc/zabbix/zabbix_proxy.conf

HistoryCacheSize=256M      # Era 16M
StartPollers=20            # Era 5
StartTrappers=15          # Era 5
ProxyDataFrequency=1       # Garantir 1s
```

### 4. Reiniciar:

```
sudo systemctl restart zabbix-proxy
```

### 5. Monitorar queue por 15 minutos:

Deve reduzir gradualmente



## Problema 3: Proxy Offline Buffer Cheio

**Sintoma:** Mensagem no log do proxy:

```
proxy buffer is full, oldest data will be dropped
```

**Causa:** Server esteve offline por tempo > ProxyOfflineBuffer

**Consequências:**

- Dados antigos são perdidos
- Gaps nos gráficos

**Solução preventiva:**

```
ProxyOfflineBuffer=24    # Aumentar de 1h para 24h
```

## Problema 4: "Database is locked" (SQLite)

**Sintoma:** Proxy lento, logs mostram:

```
database is locked  
[DATABASE IS LOCKED] ZBX_DB_STEP_1_RETRY
```

**Causa:** SQLite não suporta múltiplos writers bem

**Solução:** Migrar para MySQL/PostgreSQL

```
# Ver Lab 5 (Proxy com MySQL)  
sudo apt install zabbix-proxy-mysql  
# ... (passos de migração)
```

**Quando migrar:**

- 500 hosts

# Problema 5: Heartbeat Perdido (Active Proxy)

**Sintoma:** Active proxy mostra "red icon" no Server

**Diagnóstico:**

**1. Verificar logs do proxy:**

```
sudo tail -f /var/log/zabbix/zabbix_proxy.log | grep heartbeat
```

**Deve mostrar a cada HeartbeatFrequency segundos:**

```
sending heartbeat message  
heartbeat sent, heartbeat frequency is 60 sec
```

**Se não aparecer:**

```
cannot send heartbeat message: connection refused  
→ Server inacessível
```

# Problema 5: Solução

## 2. Verificar conectividade:

```
# No proxy  
telnet 192.168.1.10 10051
```

## 3. Se firewall:

```
# No Server  
sudo ufw allow 10051/tcp  
# Ou iptables  
sudo iptables -A INPUT -p tcp --dport 10051 -j ACCEPT
```

## 4. Verificar Server escutando:

```
# No Server  
sudo netstat -tulpn | grep 10051  
# Deve mostrar: zabbix_server listening
```

# Problema 6: Configuration Syncer Falha

**Sintoma:** Mudanças no Server não aplicam no proxy

**Logs do proxy:**

```
cannot obtain configuration data from server  
failed to get list of hosts
```

**Causa:** Permissões de DB ou Server rejeitando proxy

**Solução:**

**1. Verificar Hostname no Server:**

```
Administration → Proxies → [Verificar proxy existe e está enabled]
```

**2. Forçar reconfig:**

# Problema 7: Espaço em Disco Cheio

**Sintoma:** Proxy para de coletar, logs mostram:

```
cannot write to database: disk full
```

**Diagnóstico:**

```
df -h /var/lib/zabbix  
# Se > 90% → Problema
```

**Solução emergencial:**

```
# Limpar banco SQLite antigo  
sudo rm /var/lib/zabbix/zabbix_proxy.db-journal  
  
# Verificar tamanho do banco  
du -sh /var/lib/zabbix/zabbix_proxy.db
```

# Best Practices: Resumo

## ✓ DO:

- Usar proxy para filiais remotas (> 50 hosts)
- Usar Active mode para NAT/cloud
- Usar MySQL para > 500 hosts
- Monitorar delay e queue do proxy
- Configurar ProxyOfflineBuffer adequado
- Fazer backup regular do config
- Documentar topologia (qual proxy monitora o quê)

## ✗ DON'T:

- Usar proxy se latência já é baixa (< 10ms)
- Deixar SQLite para > 1000 hosts

# Recursos Adicionais

## Documentação oficial:

- [https://www.zabbix.com/documentation/7.0/en/manual/distributed\\_monitoring/proxies](https://www.zabbix.com/documentation/7.0/en/manual/distributed_monitoring/proxies)
- [https://www.zabbix.com/documentation/7.0/en/manual/appendix/config/zabbix\\_proxy](https://www.zabbix.com/documentation/7.0/en/manual/appendix/config/zabbix_proxy)

## Performance tuning:

- <https://www.zabbix.com/documentation/7.0/en/manual/installation/requirements>
- <https://blog.zabbix.com/zabbix-proxy-performance-tuning/>











## Community templates:

- Template for Zabbix Proxy monitoring
- <https://github.com/zabbix/community-templates>

# Revisão da Aula

## Aprendemos:

1.  Fundamentos do Zabbix Proxy (quando usar, arquitetura)
2.  Active vs Passive proxy (diferenças e casos de uso)
3.  Instalação com SQLite e MySQL
4.  Associação de hosts ao proxy (GUI e API)
5.  Sincronização e tuning de performance
6.  Estratégias para ambientes distribuídos
7.  7 laboratórios hands-on
8.  Troubleshooting avançado

# Próxima Aula

## Aula 10: Monitoramento Web e Aplicações

### Tópicos:

- Web scenarios avançados (multi-step, autenticação)
- HTTP agent para APIs REST
- Coleta de métricas customizadas via HTTP
- Análise de tempo de resposta por etapa
- Detecção de falhas e geração de alertas críticos

# Perguntas?

Obrigado!

**4Linux - Zabbix Advanced Course**