PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

GERENCIAMENTO DE REDES COM O ZABBIX COM ESTUDO DE CASO NA DISPONIBILIDADE DE UM SERVIÇO *WEB*

GABRIEL CARDOSO BARBOSA

GOIÂNIA
2022

GABRIEL CARDOSO BARBOSA

GERENCIAMENTO DE REDES COM O ZABBIX COM ESTUDO DE CASO NA DISPONIBILIDADE DE UM SERVIÇO *WEB*

Trabalho de conclusão de curso apresentado à Escola Politécnica, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientadora: Profª Ma. Angélica da Silva Nunes

GOIÂNIA

2022

GABRIEL CARDOSO BARBOSA

GERENCIAMENTO DE REDES COM O ZABBIX COM ESTUDO DE CASO NA DISPONIBILIDADE DE UM SERVIÇO *WEB*

Este Trabalho de Conclusão de Curso julgado adequado para obtenção o título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica, da Pontifícia Universidade Católica de Goiás, em ____/____/____.

_____

Prof. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

Banca examinadora:

_____

Orientadora: Profª Ma. Angélica da Silva Nunes

_____

Prof. Me. Rafael Leal Martins

_____

Prof. Me. Wilmar Oliveira de Queiroz

GOIÂNIA
2021

Dedico este trabalho a minha família, por me apoiarem nos estudos e minha formação. A minha irmã, por todo carinho e força, que me ajudou ao longo da vida.

AGRADECIMENTO

A professora Ma. Angélica da Silva Nunes pela orientação e dedicação para a elaboração deste trabalho.

Ao João Xavier da Silva Neto por diversas ajudas ao longo do trabalho.

Aos meus amigos Wellington Alves dos Santos, Walcy Santos Rezende Rios, Vanessa Ribeiro Nakamura e Karolliny Dourado Nascimento por estarem comigo em vários momentos.

“Nunca tenha certeza de nada. A sabedoria começa com a dúvida.”

Freud

RESUMO

O foco deste trabalho é estudar a disponibilidade de um serviço *Web*. Nele é levantado os prejuízos que falhas na disponibilidade podem trazer e é ressaltado a importância de se ter um gerenciamento de rede adequado. Foi utilizada a ferramenta Zabbix para monitorar, alertar, notificar, entre outros, com o objetivo de auxiliar o administrador de rede na tomada de decisões para diversos casos em que uma falha na disponibilidade venha a ocorrer. Através de máquinas virtuais, foram levantados dois servidores: o Servidor Zabbix e o Servidor *Web*. O servidor Zabbix tem a responsabilidade de mostrar uma *Interface Web* para gerenciamento centralizado, recolher e guardar as informações da máquina remota no seu banco de dados e, por fim, processar os dados coletados a fim de alertar o gerenciador em casos de falha. O Servidor *Web* tem o trabalho de disponibilizar a página *Web* através do servidor Apache. Os resultados obtidos mostraram que a ferramenta Zabbix é competente e realiza o monitoramento de diversos dados que impactam diretamente e indiretamente na disponibilidade do serviço *Web*, emitindo os alertas para casos em que houve falhas, notificando o administrador através do Gmail e, finalmente, restabelecendo o serviço automaticamente quando identificada falha no processo.

Palavras-Chave: Disponibilidade *Web*, Zabbix, Gerenciamento de redes

ABSTRACT

The purpose of this work is to study the availability of a Web service. In it, the damage that failures in availability can bring is raised and the importance of having an adequate network management is highlighted. The Zabbix tool was used to monitor, alert, notify, among others, in order to assist the network administrator in making decisions for several cases in which a failure in availability may occur. Through virtual machines, two servers were raised: the Zabbix Server and the Web Server. The Zabbix server is responsible for displaying a Web Interface for centralized management, collecting and storing information from the remote machine in its database and, finally, processing the collected data in order to alert the manager in cases of failure. The Web Server does the job of making the web page available through the Apache server. The results obtained showed that the Zabbix tool is competent and performs the monitoring of various data that directly and indirectly impact the availability of the Web service, issuing alerts for cases in which there were failures, notifying the administrator through Gmail and, finally, restoring the service automatically when a process failure is identified.

Keywords: *Web* Availability, Zabbix, Network Management

LISTA DE FIGURAS

LISTA DE SIGLAS

ASN.1	Notação de Sintaxe Abstrata Um – *Abstract Syntax Notation One*
DNS	Sistema de Nomes de Domínios – *Domain Name System*
HTTP	Protocolo de Transferência de Hipertexto – *HyperText Transfer Protocol*
HTTPD	Protocolo de Transferência de Hipertexto Daemon – *Hypertext Transfer Protocol daemon*.
HTTPS	Protocolo de Transferência de Hipertexto Seguro – *Hypertext Transfer Protocol Secure*
CPU	Unidade Central de Processamento – *Central Processing Unit*
I/O	Entrada e Saída – *Input*/*Output*
IP	Protocolo de *Internet* – *Internet Protocol*
ISO	Organização Internacional de Normalização – *International Organization for Standardization*
MB	*Megabyte*
MIB	Base de Informações de Gerenciamento – *Management Information Base*
NMS	Sistema de gerenciamento de rede - *Network Management System*
NOC	Centro de Operações da Rede – *Network Operations Center*
OID	Identificador de Objeto – *Object Identifier*
OSI	Interconexão de Sistemas Abertos – *Open System Interconnection*
PDU	Unidades de Dados de Protocolos – *Protocol Data Units*
RAM	Memória de Acesso Aleatório – *Random Access Memory*
RFC	Pedido de Comentários – *Request For Comments*
RPC	Chamada Remota de Procedimento – *Remote Procedure Call*
SGBD	Sistema Gerenciador de Banco de Dados – *Data Base Management System*
SMI	Estrutura de Informações de Gerenciamento – *Structure of Management Information*
SMS	Serviço de Mensagens Curtas – *Short Message Service*
SNMP	Protocolo Simples de Gerência de Rede – *Simple Network Management Protocol*
TI	Tecnologia da Informação
UDP	Protocolo de Datagramas de Usuário – *User Datagram Protocol*
URL	Localizador Uniforme de Recursos – *Uniform Resource Locator*
USM	Modelo de Segurança Baseado em Usuário – *User-Based Security Model*

VACM Modelo de Controle de Acesso Baseado em Visões – *View Based Access Control Model*

SUMÁRIO

1 INTRODUÇÃO

O crescimento exponencial de redes provocou uma série de necessidades das empresas. Dado que, em redes grandes e complexas, a atividade de gerenciamento de rede se torna complicada. Isto porque há necessidade de gerenciamento em diversas áreas, como o controle de atividades e monitoramento em recursos como *modem*, roteadores, servidores entre outros, incluindo recursos lógicos, buscando confiabilidade, performance e segurança. (SANTOS et al., 2015)

Falhas nas redes acarretam a perda financeira e a indisponibilidade parcial ou total de prestação de serviços. Elas são agravadas quanto mais complexa uma rede se torna, isto é, quando há um número grande de computadores. Desta forma, há necessidade de uma metodologia para auxiliar no isolamento e teste dos problemas.

O monitoramento de rede bem implementado traz diversos benefícios. Ele torna a correção de problemas mais fácil ao apontar os problemas e identificar os dispositivos defeituosos. É possível prever problemas antes deles acontecerem, por exemplo, o disco de um servidor quase cheio. Ainda neste exemplo, o monitoramento irá permitir que a equipe de Tecnologia da Informação - TI invista eficientemente onde é mais necessário no ambiente, neste caso, no aumento de espaço disponível.

O Zabbix é uma ferramenta para o gerenciamento de rede com a capacidade de escalonar para grandes ambientes. A ferramenta pode medir o desempenho e a disponibilidade dos dispositivos, oferece *interface Web*, facilitando o gerenciamento e a visualização dos dados.

Sobre o que é o Zabbix segundo a sua documentação, pode-se afirmar que:

> O Zabbix é um software que monitora numerosos parâmetros de rede, a saúde e integridade de servidores, máquinas virtuais, aplicações, serviços, banco de dados, websites, a nuvem e muito mais. O Zabbix usa um mecanismo flexível de notificação que permite aos usuários configurar alertas baseados em e-mail para praticamente qualquer evento. Isso permite uma resposta rápida para problemas do servidor. O Zabbix oferece um excelente recurso de relatórios e visualização de dados baseados em dados armazenados. Isso torna o Zabbix ideal para gerenciamento de capacidade. (ZABBIX SIA, 2022, p. 1).

Criado por Alexei Vladishev em 1998, com sua primeira versão estável lançada em abril de 2001, o Zabbix é um *software* de monitoramento cujo código é aberto. O *software* permite

a monitoração de vários parâmetros de uma rede, servidores, máquinas virtuais, serviços, *sites* e mais. (CODATA, 2018).

A Zabbix SIA COMPANY foi fundada em 2015 e é a empresa responsável pelo suporte técnico do produto, dentre outras reponsabilidades.

O Zabbix tem como ponto alto sua alta disponibilidade pois possui um *frontend* baseado em *Web*. Os relatórios e as estatísticas podem ser acessados remotamente. A facilidade de visualização das informações se estende já que é possível o acesso a dados previamente armazenados.

Justifica-se pesquisar este assunto porque a indisponibilidade de um serviço pode trazer, além da grande perda financeira a uma empresa, consequências de cunho legal. O *software* Zabbix é usado nesse trabalho com o objetivo de verificar os benefícios de seu uso no monitoramento de um servidor *Web* para identificar quaisquer falhas que afetem a disponibilidade do serviço.

Diante do contexto, esta pesquisa pretende responder a seguinte questão:

- Quais os benefícios que o Zabbix pode trazer para a disponibilidade de um Servidor *Web*?

1.1 Objetivo Geral

Realizar o monitoramento de um servidor *Web* com foco em sua disponibilidade através do Zabbix.

1.2 Objetivos Específicos

- Identificar incidentes que prejudicam a disponibilidade do serviço *Web*;
- Aplicar solução de resposta adequada para tratativa dos erros no ambiente;
- Guardar o histórico das informações coletadas para possibilitar análise dos dispositivos e serviços implementados;
- Aprofundar o conhecimento sobre o *software* Zabbix.

1.3 Procedimentos Metodológicos

Esta pesquisa quanto a sua natureza é um resumo de assunto pois agrega, analisa e aborda informações já publicadas. Resumos de assunto buscam sistematizar uma área de conhecimento (WAZLAWICK, 2014).

Quanto aos objetivos é uma pesquisa explicativa pois busca causas e explicações para os dados observados, em outras palavras, os fatores determinantes desses dados.

Quanto aos procedimentos técnicos é uma pesquisa experimental, pois é feita a manipulação de variáveis ou parâmetros do ambiente de rede, permitindo a medição e análise de resultados consequentes.

1.4 Estrutura da monografia

O trabalho está estruturado da seguinte estrutura:

No capítulo 2 é apresentada a base teórica sobre gerenciamento de redes, em que é explicado sua estrutura, a base de informações de gerenciamento e a estrutura de informações de gerenciamento.

No capítulo 3 é explicado sobre o protocolo usado na comunicação de rede, suas capacidades, seus modos de operação, as mensagens enviadas por ele e suas versões.

O capítulo 4 descreve a ferramenta Zabbix, suas características, requisitos, componentes, suas funcionalidades e principais capacidades.

No capítulo 5 o ambiente criado é detalhado, mostrando seus componentes principais, as configurações implementadas no Zabbix e a página criada com o servidor Apache.

No capítulo 6 são realizadas simulações no ambiente, mostrando quais objetivos as simulações visam alcançarem e como o ambiente se comportou nelas.

O capítulo 7 apresenta as considerações finais deste trabalho.

2 GERENCIAMENTO DE REDES

Esse capítulo retrata sobre a definição de gerenciamento de rede, porque ela é relevante, o que ela garante em uma rede. Mostra como o gerenciamento modela uma rede, indica e explica suas áreas, apresenta a estrutura com os componentes internos, informa definições do banco de dados responsável por guardar informações pertinentes ao gerenciamento e, por fim, indica a linguagem de definição dos dados.

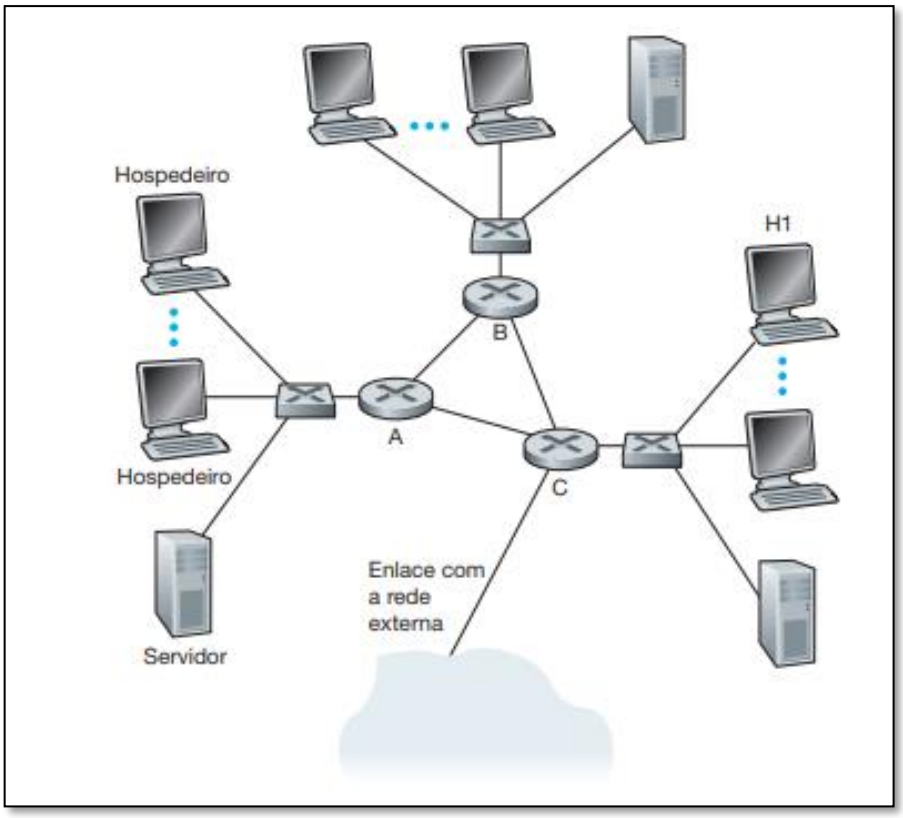
2.1 Gerenciamento de rede

As redes de computadores têm muitos elementos que interagem entre si. É comum que, em uma rede, problemas comecem a ocorrer, defeitos, elementos da rede mal configurados ou recursos da rede utilizados em excesso. Desta forma, surge a necessidade de gerenciar os elementos da rede com propósito de garantir uma série de recursos, tais como: confiabilidade, segurança, disponibilidade, monitoramento e análise dos elementos ativos da rede etc. (KUROSE, 2013). A Figura 1 ilustra uma rede simples que contém computadores (hospedeiro), *switches*, roteadores e servidores.

Sobre o gerenciamento de rede, Saydam diz que:

> Gerenciamento de rede inclui a implementação, a integração e a coordenação de elementos de *hardware*, *software* e humanos, para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de elementos, para satisfazer às exigências operacionais, de desempenho e de qualidade de serviço a um custo razoável. (SAYDAM, 1996, p. 345).

Figura 1 – Cenário de uma rede simples



Fonte: Kurose, 2013

2.2 Modelo do gerenciamento International Organization for Standardization

De acordo com Kurose (2013), a Organização Internacional de Normalização (*International Organization for Standardization* - ISO) criou um modelo de gerenciamento de rede. Este modelo define cinco áreas de gerenciamento de rede:

- Gerenciamento de desempenho: tem como meta quantificar, medir, analisar e controlar o desempenho da rede. Desta forma é possível avaliar o desempenho dos componentes de uma rede e da qualidade do serviço. Permite que usuários da rede possam saber o tempo médio de resposta e confiabilidade dos recursos da rede;
- Gerenciamento de falhas: sua finalidade é registrar, detectar e reagir às condições de falha da rede. Permite a resolução rápida e segura de algum problema, bem como a notificação aos usuários;
- Gerenciamento de configuração: permite que o administrador saiba quais dispositivos fazem parte da rede e suas configurações de *hardware* e *software*. Informa a condição dos componentes e recursos da rede aos usuários. Tem objetivo de automatizar ações em certos componentes;

- Gerenciamento de contabilização: permite que o administrador da rede especifique, registre e controle o acesso de usuários e dispositivos aos recursos da rede. Permite o administrador determinar os custos relacionados ao uso da rede;
- Gerenciamento de segurança: sua meta é controlar o acesso aos recursos da rede de acordo com alguma política. Garante a proteção dos recursos de rede e das informações dos usuários.

2.3 Estrutura do gerenciamento

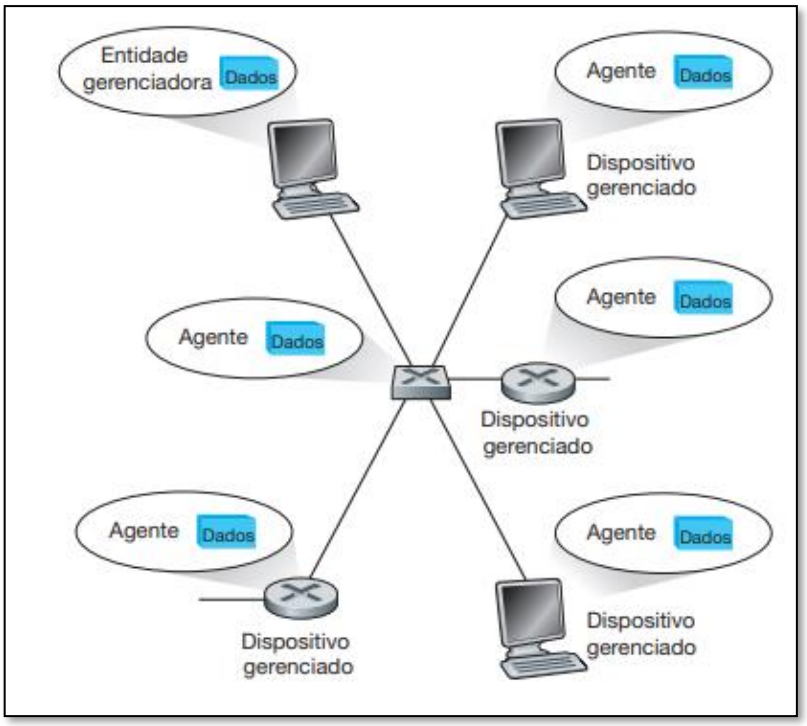
A construção da infraestrutura do gerenciamento de rede é realizada com o conjunto de três componentes principais. São eles a entidade gerenciadora, o dispositivo gerenciado e um protocolo de gerenciamento. Elementos secundários, mas que compõem a estrutura, são a Base de Informações de Gerenciamento (*Management Information Base* – MIB) e o agente de gerenciamento.

A entidade gerenciadora é uma aplicação que roda em uma estação central na *Network Operations Center* (NOC). É o centro da atividade. Responsável pela coleta e controle, processamento e análise das informações de gerenciamento de rede.

Um dispositivo gerenciado é um equipamento de rede (hospedeiro, roteador, *switch*, *hub*, *bridge*, impressora ou *modem*) que se encontra em uma rede gerenciada. Em seu interior existem diversos objetos gerenciados, elementos de *hardware* e *software*, como por exemplo, uma placa de *interface* de rede, processador, memória ou protocolo de roteamento intra-domínio. A MIB guarda as informações associadas aos objetos gerenciados, podendo disponibilizar essas informações para a entidade gerenciadora. Por fim, o agente de gerenciamento, que é um processo executado em cada dispositivo gerenciado, é responsável por realizar a comunicação com a entidade gerenciadora e executar ações no dispositivo.

O último componente principal da estrutura de gerenciamento de rede é o protocolo de gerenciamento de rede. Ele é responsável por realizar a comunicação entre a entidade gerenciadora e o agente que reside nos dispositivos gerenciados. É através dele que é possível monitorar, consultar, configurar, avaliar etc. a rede. Pode-se afirmar que o protocolo é uma ferramenta com a qual o administrador pode gerenciar. A Figura 2 mostra uma rede simples onde está sendo aplicado o gerenciamento de rede com alguns dos componentes citados.

Figura 2 – Principais componentes da estrutura de gerenciamento



Fonte: Kurose, 2013

2.4 Base de Informações de Gerenciamento (MIB)

Segundo Kurose (2013), a MIB pode ser entendida como um banco virtual que guarda informações cujos valores refletem o estado da rede. As informações guardadas podem ser consultadas ou definidas por um gerenciador através de mensagens do Protocolo Simples de Gerenciamento de Redes (*Simple Network Management Protocol* - SNMP) ao agente de gerenciamento de rede citado na seção 2.2. Objetos gerenciados são definidos pela Estrutura de Informações de Gerenciamento (*Structure of Management Information* – SMI) através da construção OBJECT-TYPE e agrupados em módulos MIB com a construção MODULE-IDENTITY.

A construção OBJECT-TYPE define quatro cláusulas e seu objetivo é especificar o nome do objeto, seu tipo, sua forma de acesso, o status e uma descrição textual. A Figura 3 ilustra a construção de um objeto gerenciado definida pela linguagem SMI.

Figura 3 - Construção OBJECT-TYPE do objeto *ipSystemStatsInDelivers*

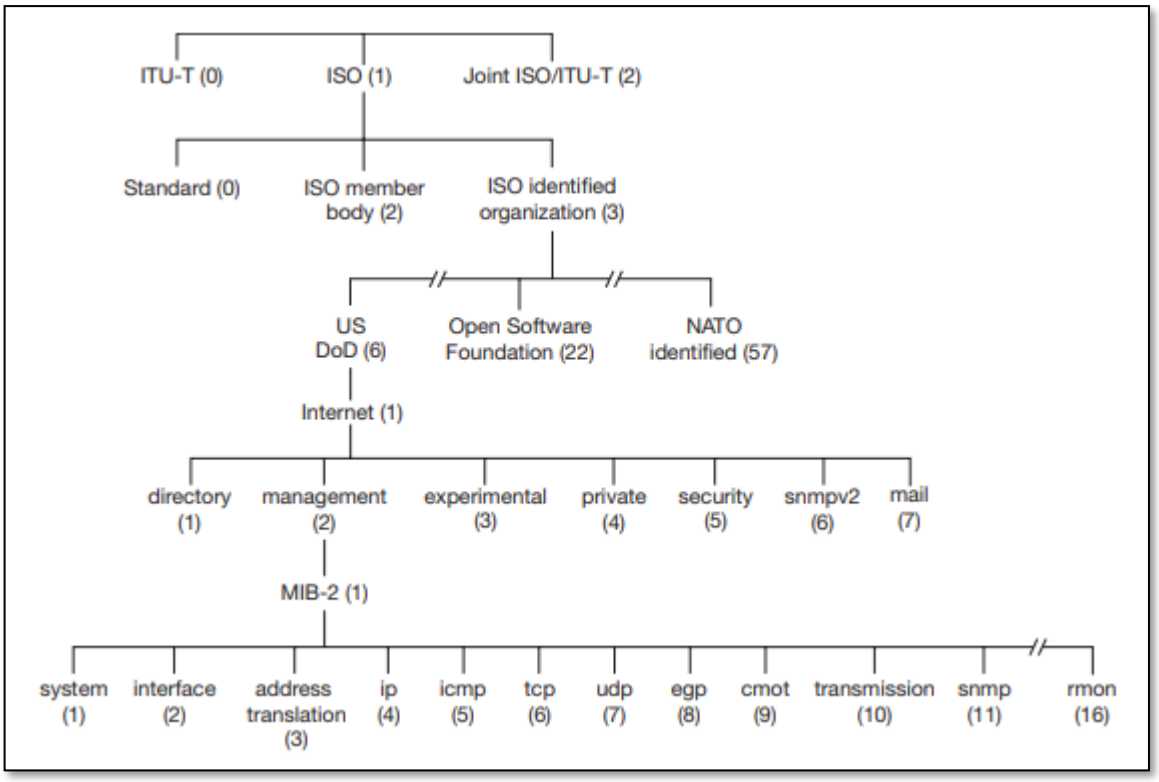
```
ipSystemStatsInDelivers OBJECT-TYPE
       SYNTAX Counter32
       MAX-ACCESS read-only
       STATUS current
       DESCRIPTION
              "The total number of datagrams successfully
              delivered to IPuser-protocols (including ICMP).

              When tracking interface statistics, the counter
              of the interface to which these datagrams were
              addressed is incremented. This interface might
              not be the same as the input interface for
              some of the datagrams.

              Discontinuities in the value of this counter can
              occur at re-initialization of the management
              system, and at other times as indicated by the
              value of ipSystemStatsDiscontinuityTime."
       ::= { ipSystemStatsEntry 18 }
```

Fonte: Kurose, 2013

A construção MODULE-IDENTITY, segundo Kurose (2013), permite o agrupamento de objetos relacionados. Sua documentação contém descreve informações para contato com o autor do módulo, a data da última atualização, histórico de revisões e descrição sobre o módulo.

A MIB II, definida no Pedido Para Comentários (*Request For Comments)* 1213 (RFC 1213, 1991), é organizada de forma hierárquica através da estrutura de uma árvore que por sua vez é especificada na linguagem de descrição de *interface* padrão Número de Sintaxe Abstrata (*Abstract Syntax Notation One* - ASN.1). Cada objeto da árvore possui um identificador próprio, denominado Identificador de Objeto – *Object Identifier* (OID), e pode ser especificado através do caminho da raiz até a "folha". A Figura 4 demonstra a disposição de alguns objetos na hierarquia.

Figura 4 - Árvore de identificadores de objetos ASN.1



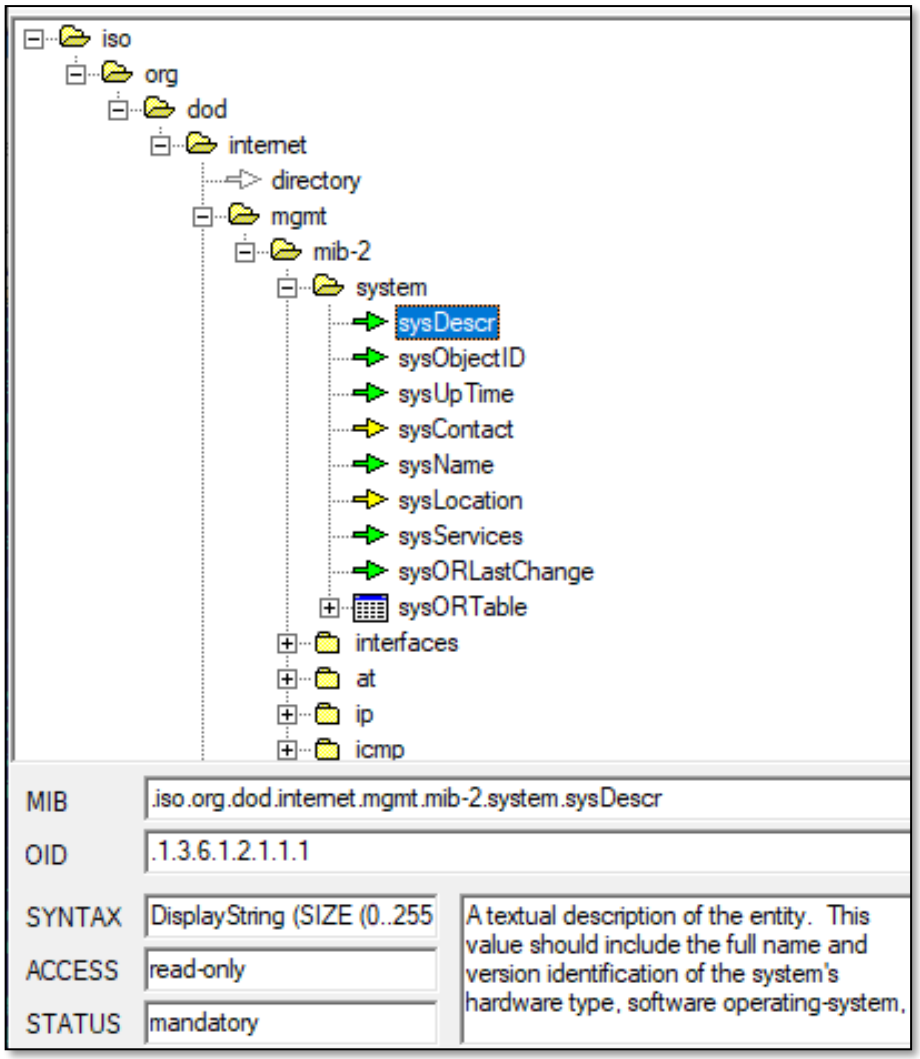
Fonte: Kurose, 2013

2.5 Estrutura de Informações de Gerenciamento (SMI)

A SMI é a linguagem que permite especificar os objetos gerenciados listados na MIB. Ela assegura a sintaxe e a semântica dos dados e evita ambiguidade, além de possibilitar o agrupamento de objetos em subconjuntos. A RFC 2578 (RFC 2578, 1999) documenta a SMIv2.

Segundo as construções da SMI: OBJECT-TYPE e a MODULE-IDENTITY. A Figura 5 exemplifica a implementação da especificação do objeto *sysDescr* no aplicativo *desktop Host*Monitor em que é descrito seu OID, sintaxe, acesso, *status* e descrição. (Kurose, 2013).

Figura 5 - Construção do objeto MIB *sysDescr*



Fonte: Recorte capturado de *Host* Monitor, 2021

3 PROTOCOLO SNMP

Este capítulo trata sobre o Protocolo SNMP, abordando sobre a sua criação, suas capacidades, os tipos de mensagens e evolução com o tempo.

3.1 Introdução

O SNMP é um dos protocolos mais utilizados para auxiliar na gerência de rede. O SNMP teve sua origem na RFC 1067 (RFC 1067, 1988) em 1988. Conforme necessidades tecnológicas, evoluiu em segurança e performance por algumas versões, estando atualmente na versão 3. É um protocolo da camada de aplicação, especificamente a camada de número sete modelo Interconexão de Sistemas Abertos (*Open Systems Interconnection* - OSI) que utiliza usualmente a porta 161 do protocolo de transporte UDP (TELCO MANAGER, s.d.).

3.2 Capacidades do SNMP

As capacidades principais do protocolo SNMP se reúnem nas três seguintes:

- *Get*: Permite que a entidade gerenciadora colete as informações MIB de um dispositivo gerenciado;
- *Set*: Permite que a entidade gerenciadora defina o valor de objetos MIB de um dispositivo gerenciado;
- *Notify*: Permite que um agente de gerenciamento envie mensagens assíncronas para a entidade gerenciadora na ocorrência de uma situação excepcional.

O protocolo SNMP tem duas principais utilizações, são elas o modo comando-resposta e a mensagem *trap*. As figuras 6 e 7 mostram, respectivamente, as duas utilizações do protocolo SNMP.

Figura 6 - Modo comando-resposta

Fonte: Autoria própria, adaptado de Kurose, 2013

O modo comando/resposta ocorre quando a entidade gerenciadora envia uma requisição para um agente de gerenciamento que, executa a ação localmente no dispositivo gerenciado e, por fim, retorna uma resposta para a entidade gerenciadora.

Figura 7 - Mensagem *Trap*

Fonte: Autoria própria, adaptado de Kurose, 2013

As mensagens *trap* são acionadas no evento de uma situação excepcional definida pela entidade gerenciadora previamente. Um evento excepcional pode ser a interrupção de algum serviço, a utilização excessiva de algum elemento, entre outros. Essas mensagens são caracterizadas por não serem solicitadas, isto é, são assíncronas.

3.3 Mensagens SNMP

O protocolo SNMP define sete mensagens que implementam suas capacidades. Estas mensagens são conhecidas como *Protocol Data Units* (PDU), são elas:

- *GetRequest*;

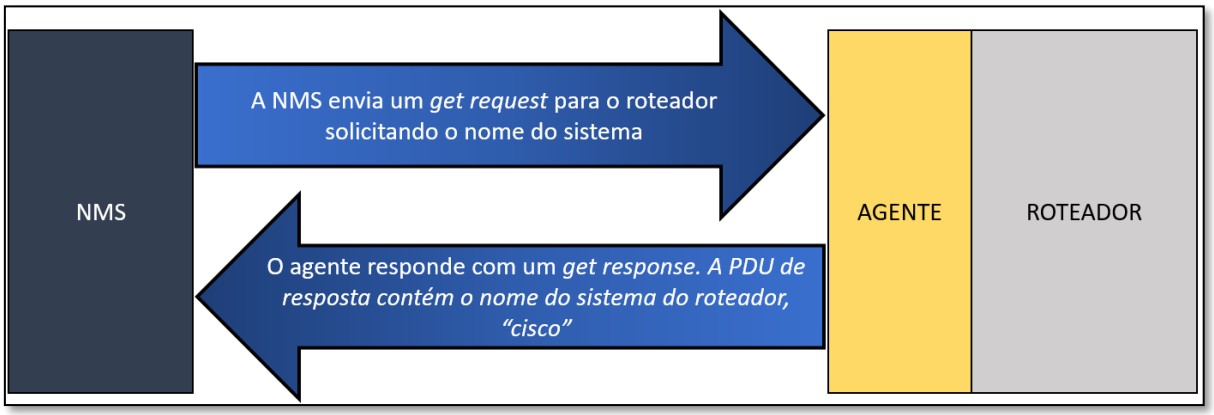
- *GetNextRequest*;
- *GetBulkRequest*;
- *Response*;
- *SetRequest*;
- *InformRequest*;
- *Trap*.

Nota-se que o SNMPv1 não implementa as mensagens *GetBulkRequest* e *InformRequest*. Esses importantes PDUs foram implementadas a partir da segunda versão do protocolo. (Stallings, 2005).

3.3.1 GetRequest

O comando *GetRequest* permite ao administrador receber o valor de um ou mais objetos MIB. Esse comando é enviado a partir de uma entidade gerenciadora para um agente. Nele, inclui-se uma lista de um ou mais objetos MIB que serão requisitados. O agente responde esta PDU com uma PDU Response. Essa PDU Response contém o identificador e o valor de todos os objetos requisitados. Caso algum problema ocorra na busca de um objeto, o identificador e código de erro serão retornados para ele. Essa operação é ilustrada na Figura 8.

Figura 8 – Operação *Get Request*



Fonte: Autoria própria, adaptado de Mauro, 2001

3.3.2 GetNextRequest

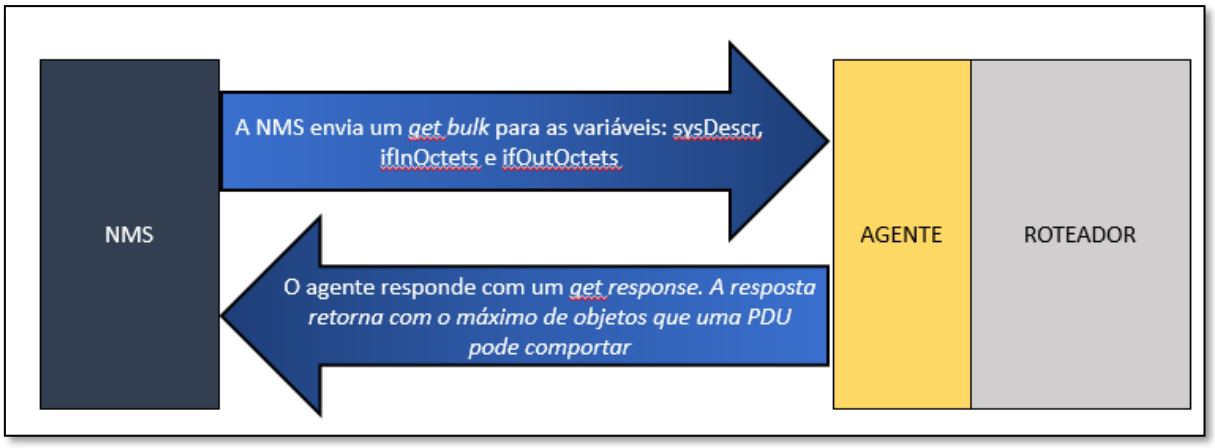
O comando *GetNextRequest* permite ao administrador receber o valor de um ou mais objetos MIB. Esse comando é enviado a partir de uma entidade gerenciadora para um agente. Nele, um valor é recuperado para o próximo objeto na ordem lexicográfica. O agente responde esta PDU com uma PDU Response quantas vezes necessária até que se percorra toda a MIB do

dispositivo. Semelhante a PDU *GetRequest*, a PDU *Response* contém o identificador e o valor de todos os objetos requisitados. Esse comando permite ao gerenciador descobrir o escopo do dispositivo, retornando todos os objetos gerenciados.

3.3.3 GetBulkRequest

O comando *GetBulkRequest* permite ao administrador receber um grande bloco de objetos MIB. Esse comando é enviado a partir de uma entidade gerenciadora para um agente. A operação solicita que o agente envie o máximo de dados que é possível transportar em uma PDU Response. Esse comando evita a sobrecarga que poderia ocorrer no uso de múltiplos *GetRequest* ou *GetNextRequest*. Essa operação é ilustrada na Figura 9.

Figura 9 - Operação *GetBulkRequest*

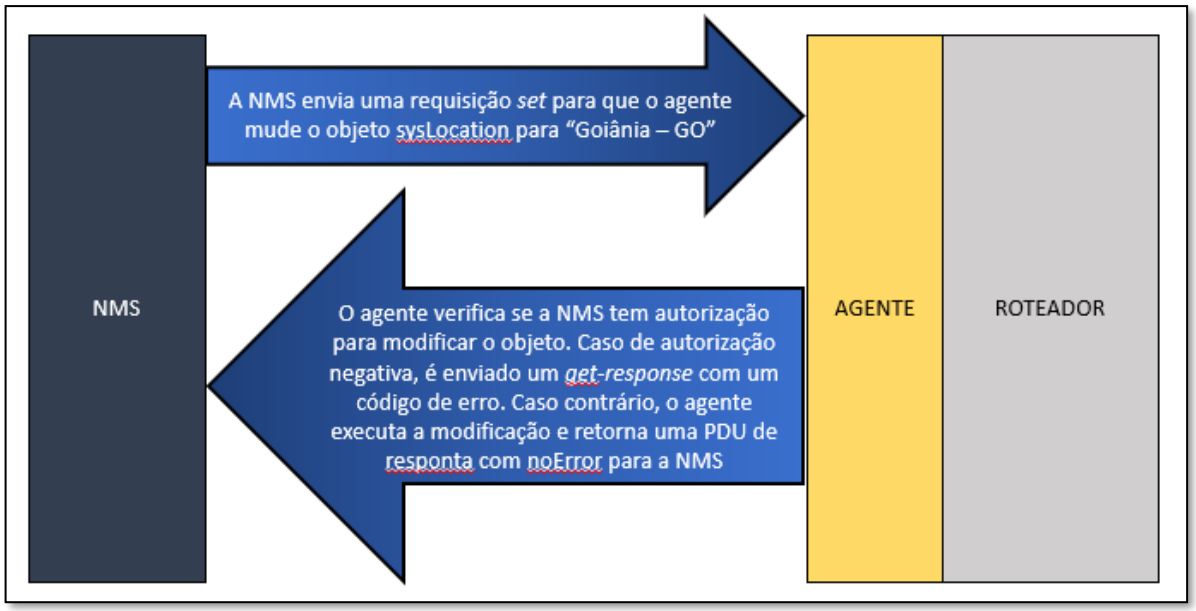


Fonte: Autoria própria, adaptado de Mauro, 2001

3.3.4 SetRequest

O comando *SetRequest*, ilustrado pela Figura 10, permite que o administrador defina o valor de objetos MIB em um dispositivo na rede. Nesta operação a entidade gerenciadora envia o comando para um agente. O agente realiza as ações no dispositivo e, então, retorna uma PDU *Response* confirmando a(s) alteração(ões) do(s) valor(es).

Figura 10 - Operação *SetRequest*

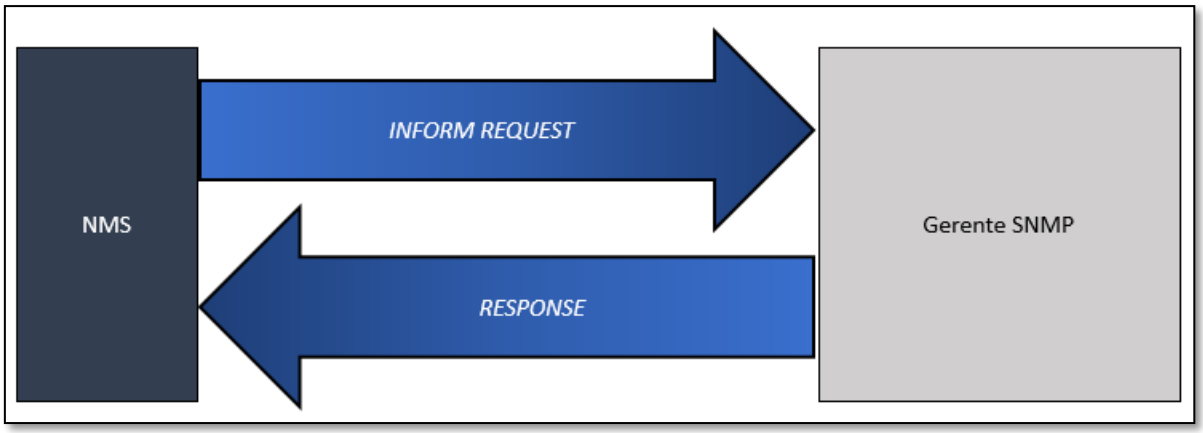


Fonte: Autoria própria, adaptado de Mauro, 2001

3.3.5 InformRequest

A Figura 11 demonstra o comando *InformRequest*, que é usado quando uma entidade gerenciadora se comunica com outra entidade gerenciadora. Nesta comunicação, uma entidade envia informação MIB para uma entidade receptora, que retorna uma PDU Response confirmando o recebimento da PDU *InformRequest*.

Figura 11 - Operação *Inform Request*



Fonte: Autoria própria, adaptado de Cisco, 2013

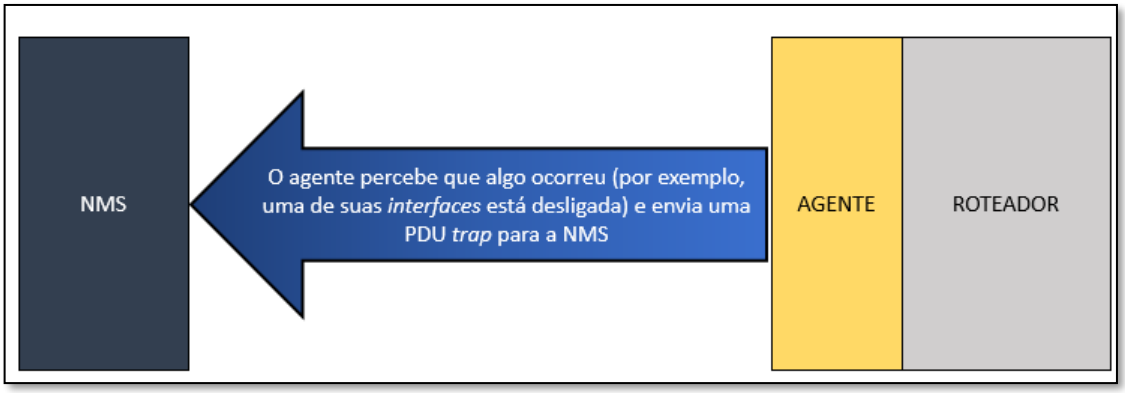
3.3.6 Response

Diferente dos comandos vistos, o *Response* é uma mensagem que pode conter os valores de variáveis em resposta a uma requisição ou confirmação de uma mudança realizada pela entidade gerenciadora.

3.3.7 Trap

As *Traps* são mensagens geradas assincronamente. Elas são causadas a partir de um agente e enviadas para uma entidade gerenciadora. O propósito desta mensagem é notificar o gerenciador de um evento extraordinário em algum valor dos objetos MIB. A operação é ilustrada pela Figura 12.

Figura 12 - Operação *Trap*

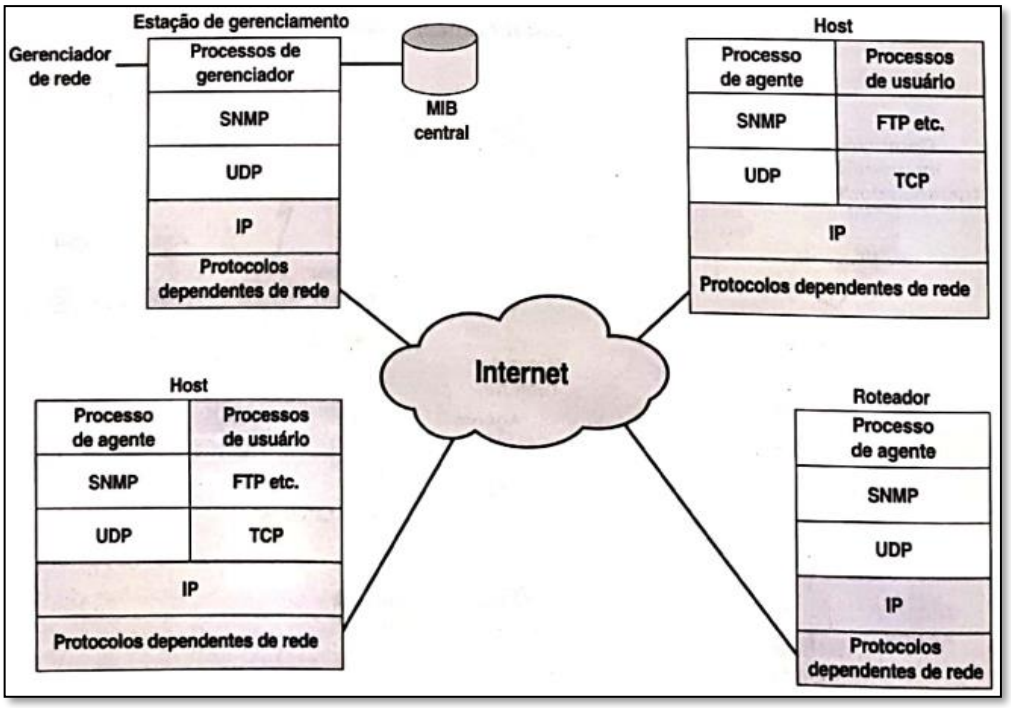


Fonte: Adaptado de Mauro, 2001

3.4 SNMPv1

A primeira versão do protocolo SNMP é definida, principalmente, através dos RFC's: RFC 1155 (RFC 1155, 1990), RFC 1157 (RFC 1157, 1990), RFC 1212 (RFC 1212, 1991) e RFC 1213 (RFC 1213, 1991). Nesta configuração, o protocolo contém as PDU's *Get*, *Getnext*, *Getresponse*, *Set e Trap*, abordadas nos sub-tópicos da seção 3.3. Um exemplo de sua possível configuração é demonstrado através da Figura 13.

Figura 13 - Configuração do SNMPv1



Fonte: Stallings, 2005

Como modelo de segurança, é usado o nome de comunidade. Em sua essência, os nomes de comunidades são senhas (*strings*), as quais permitem atividades entre os gerentes e agentes. Existem três comunidades: *read-only*, *read-write* e *trap*. Todas as três são configuradas nos agentes e permitem diferentes funções. A comunidade *read-only* autoriza apenas a função de leitura de valores nos dispositivos. Consequentemente, *read-write* libera as funções de escrita e leitura de dados, permitindo também execução de certas funções como redefinição de *interface*s ou configurações em algum dispositivo. Por fim, a comunidade *trap* concede o recebimento de mensagens assíncronas *trap* do agente.

Boas práticas devem ser levadas em consideração no uso desse modelo inseguro. Por ser, em sua essência, senhas, deve-se evitar senhas fracas ou comuns usadas em ataques de dicionário. A insegurança deste modelo se deve ao fato de que as mensagens não são criptografadas, isto é, qualquer *sniffing* na rede pode interceptar a mensagem em texto simples e, assim, obter a senha e consequentemente algum acesso.

Segundo Mauro e Schmidt (2001), pode-se minimizar a chance de ataque através de configurações no *Firewall*. Esta medida de proteção se trata da limitação do tráfego apenas entre os *hosts* conhecidos, liberando, por exemplo, o tráfego UDP na porta 161 somente se a solicitação vier de uma estação de gerenciamento.

O SNMPv1 possui certas deficiências como a não adequabilidade em coletar grande quantidade de dados, escalabilidade limitada e insegurança. A primeira versão não permite a

comunicação entre gerentes, não permite a configuração remota dos agentes e as mensagens têm característica de serem atômicas, em que ou se obtém total sucesso ou total fracasso nas operações.

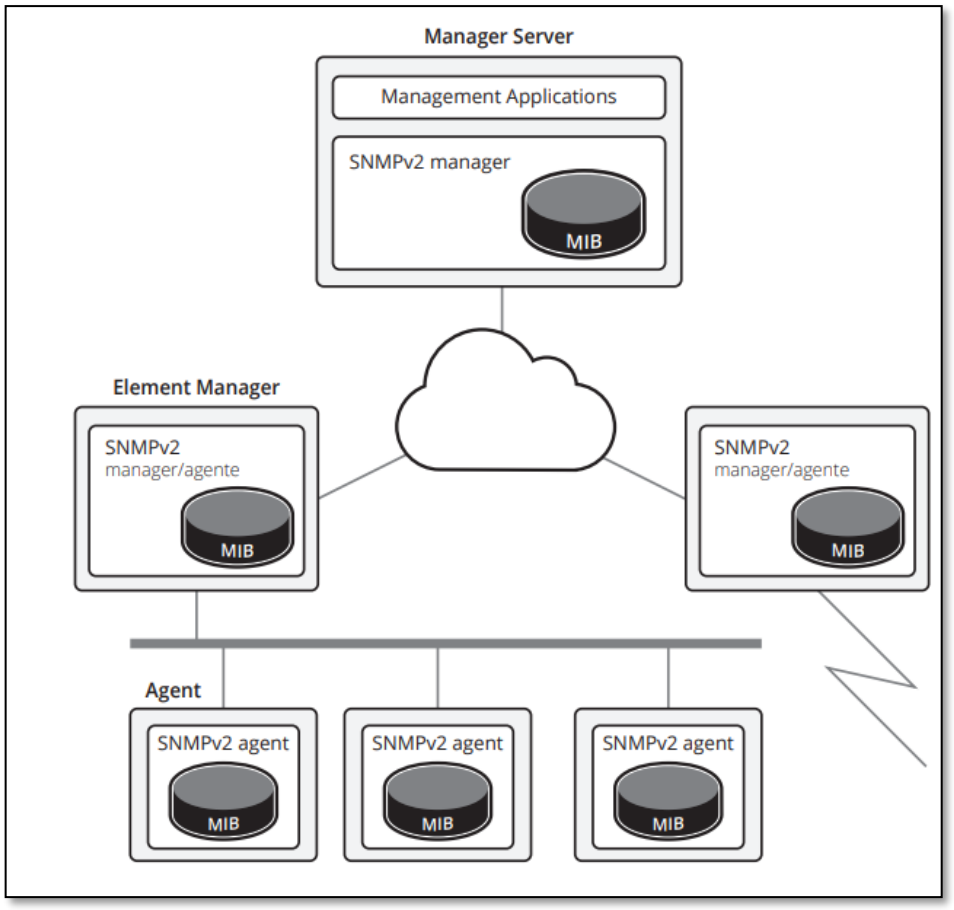
3.5 SNMPv2

Em 1993 foi lançada a segunda versão do protocolo SNMP, esta que não obteve sucesso em seu lançamento. Consequentemente, três anos depois (1996) foi lançada a versão definitiva do protocolo, isto é, a SNMPv2c que é o SNMPv2 baseado em comunidades. Nesta nova revisão do protocolo, novas PDUs's foram implementadas: *Getbulk*, *Inform*, *Report*, *Getbulkresponse* e *Report*.

O SNMPv2 possui algumas vantagens em relação ao seu antecessor, tais como:

- SMI aprimorada: a SMIv2 possui maior variedade de tipos de dados e melhor manipulação de tabelas;
- Melhor coleta: o protocolo introduziu uma nova PDU, que é capaz de obter uma grande quantidade de dados;
- Descentralização: a comunicação entre gerentes se tornou possível. Desta forma, viabiliza gerentes intermediadores em uma arquitetura de rede. A Figura 14 ilustra uma possível configuração descentralizada do protocolo SNMPv2.

Figura 14 - Gerenciamento Descentralizado SNMPv2



Fonte: Santos, 2015

Porém, pode-se observar a ausência de mecanismos eficazes de segurança pois a nova revisão continuou com o modelo baseado em comunidades, tal qual seu antecessor utilizava.

3.6 SNMPv3

O mais novo protocolo teve seu desenvolvimento iniciado em 1998 e uma de suas mudanças mais importantes foi a melhoria trazida na segurança. O SNMPv3 garante compatibilidade com as versões anteriores e uma arquitetura modular. De acordo com Kurose (2013), a segurança SNMPv3 provê proteção contra os ataques de reprodução, fornece autenticação, criptografia e controle de acesso.

Pode-se definir três principais serviços para a terceira versão do protocolo, são eles o mecanismo de autenticação e o serviço de privacidade no Modelo de Segurança Baseado em Usuário (*User-Based Security Model* – USM), sendo o terceiro serviço o Modelo de Controle de Acesso Baseado em Visões (*View-based Access Control Model* - VACM). O USM é responsável pela autenticação e criptografia dos pacotes SNMP, enquanto o VACM é responsável pela administração do acesso aos dados na MIB.

4 ZABBIX

Neste capítulo a ferramenta principal deste trabalho é abordada, o Zabbix. São descritas as suas características, os requisitos para seu uso, o que ele oferece e suas principais funcionalidades. A importância desta ferramenta se dá principalmente na sua robustez, oferecendo diversas opções ao gerente de rede.

4.1 Características

Segundo a Rede Nacional de Ensino e Pesquisa (SANTOS, 2015), o Zabbix tem diversas características que justificam seu uso. É um *software* escalável que já foi testado com 5 mil dispositivos. Possui monitoramento em tempo real de desempenho, disponibilidade e integridade, além de flexibilidade de alerta pois a ferramenta possibilita o envio de notificações através do Serviço de Mensagens Curtas (*Short Message Service* - SMS), *e-mail*, avisos sonoros e comandos remotos. O *software* de monitoramento possui várias outras características como a segurança, administração eficaz, uso de agentes extensíveis em diversas plataformas e mais vantagens. Logo, ele se mostra uma ferramenta competente para monitoramento e com diversas soluções.

4.2 Pré-requisitos

Os requerimentos para instalação do Zabbix são de 128 *Megabyte* (MB) de Memória de Acesso Aleatório – *Random Access Memory* (RAM) e de pelo menos 256 MB de espaço em disco. Alerta-se que o espaço em disco tende a aumentar conforme o número de *hosts* e de parâmetros a serem monitorados cresce (Zabbix SIA, 2022). A Figura 15 mostra exemplos de *hardware* que comportam o Zabbix e suas configurações:

Figura 15 - Configurações de *hardware* para o Zabbix

Nome	Plataforma	CPU/Memória	SGDB	Hosts Monitorados
Small	CentOS	Virtual Appliance	MySQL InnoDB	100
Medium	CentOS	2 CPU cores/2GB	MySQL InnoDB	500
Large	RedHat Enterprise Linux	4 CPU cores/8GB	RAID10 MySQL InnoDB ou PostgreSQL	>1000
Very large	RedHat Enterprise Linux	8 CPU cores/16GB	RAID10 rápido MySQL InnoDB ou PostgreSQL	>10000

Fonte: Zabbix SIA, 2022

O Zabbix SIA (ZABBIX SIA, 2022) recomenda o uso do sistema UNIX por ser o único capaz de entregar competentemente requisitos de *performance*, tolerância a falha e resiliência.

As seguintes plataformas foram testadas para o uso da ferramenta: Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris, Windows nas versões 2000, Server 2003, XP, Vista, Server 2008, 7, 8, Server 2012 (Apenas Zabbix *Agent*).

4.3 Componentes do Zabbix

Esse *software* é composto por uma variedade de componentes. Para melhor entendimento de sua composição, abaixo a lista com os termos:

- *Host*: dispositivo de rede monitorado através de seu Protocolo de *Internet – Internet Protocol* (IP) / Sistema de Nomes de Domínios – *Domain Name System* (DNS);
- Grupo de *hosts*: agrupamento lógico de *hosts* ou *templates* que são utilizados para as definições de acesso dos diferentes grupos de usuários;
- Item: dado que se deseja receber de um *host*;
- *Trigger*: expressão lógica usada para definir a ocorrência de incidentes;
- Evento: ocorrência de algum evento que mereça atenção. Por exemplo, o acionamento de um incidente devido a uma *trigger*;
- Ação: reação pré-definida a um evento. Por exemplo, o envio de uma notificação;
- Escalonamento: sequência de envio de notificações ou execução de comandos remotos;
- Mídia: um canal de distribuição usado para entregar uma notificação (mensagem sobre algum evento);
- Comando remoto: um comando pré-definido que é automaticamente executado quando determinada condição ocorrer em um *host* monitorado;
- *Template*: um conjunto de entidades (itens, *triggers*, gráficos, telas, aplicações, regras, cenários *Web*). Sua função é acelerar a criação de um perfil de monitoração reutilizável e tornar mais fácil as mudanças em massa dos perfis de monitoração em vários *hosts*.
- Aplicação: grupo lógico de itens;
- Cenário *Web*: são requisições Protocolo de Transferência de Hipertexto – *Hypertext Transfer Protocol* (HTTP), ou Protocolo de Transferência de Hipertexto Seguro – *Hypertext Transfer Protocol Secure* (HTTPS), utilizadas para verificar a disponibilidade de uma página *Web*;
- *Front-end*: *interface Web* utilizada para a configuração e acesso ao ambiente Zabbix;

- API Zabbix: permite a criação, atualização e recebimento de objetos Zabbix ou execução de alguma tarefa personalizada através da Chamada Remota de Procedimento – *Remote Procedure Call* (RPC);
- Servidor Zabbix: componente central do Zabbix que é responsável por realizar o monitoramento, interação com os *proxies* e agentes, o cálculo das mudanças de estado nas *triggers*, envio das notificações e controle do repositório central de dados;
- Agente Zabbix: componente instalado nos *hosts*, usado para monitorar ativamente seus recursos e aplicações;
- *Proxy* Zabbix: componente com capacidade de realizar a coleta de dados no lugar do Servidor Zabbix, distribuindo a carga de processamento (ZABBIX SIA, 2022).

4.4 Funcionalidades

Segundo a documentação do Zabbix, a ferramenta é uma solução de monitoração integrada, provendo diversos recursos em um único pacote. A listagem a seguir descreve grande parte de suas funcionalidades (ZABBIX SIA, 2022):

- Coleta de dados: verificação de disponibilidade e desempenho; suporte ao SNMP; intervalos personalizáveis para coleta de dados; o servidor, *proxy* ou agentes podem realizar a coleta;
- Definição de limites: definição da configuração dos gatilhos;
- Alertas configuráveis: envio de notificações em diversas mídias diferentes (*e-mail*, Telegram, Discord...) e a possibilidade de envio de comandos remotos;
- Gráficos em tempo real: itens monitorados têm seus valores armazenados e podem gerar gráficos;
- *Web monitoring capabilities*: permite executar uma sequência de passos em um *site*;
- Opções de visualização: possibilita o ajuste dos gráficos com vários itens; mapa de rede; emissão de relatórios;
- Histórico e armazenamento de dados: um banco de dados guarda as informações; histórico customizável;
- Configuração simplificada: *hosts* podem ser incluídos em *templates*, facilitando e agilizando o monitoramento;
- Descoberta de rede: descobre automaticamente os dispositivos na rede;
- API Zabbix: *interface* programável para atualizações em massa e integração com ferramentas de terceiros;

- Sistema de permissões: autenticação dos usuários; limitação de permissão para determinadas funções para determinados de usuários;
- Arquitetura de agente expansível: instalação abrange diversos sistemas operacionais;

4.5 Checagem simples

A checagem simples é um tipo de verificação que retorna “sim” ou “não”. Normalmente é usada nos casos em que a instalação do agente Zabbix não é possível ou em situações de monitoramento de serviços em execução, como por exemplo catracas, que são equipamentos que não possuem suporte ao protocolo SNMP.

4.6 Servidor Zabbix

O servidor é responsável pelos principais papeis na ferramenta. Suas atividades são a coleta e recebimento dos dados, cálculo do estado das dos gatilhos e o envio de notificações para os usuários. Os dados enviados pelo agente Zabbix são recebidos pelo servidor Zabbix.

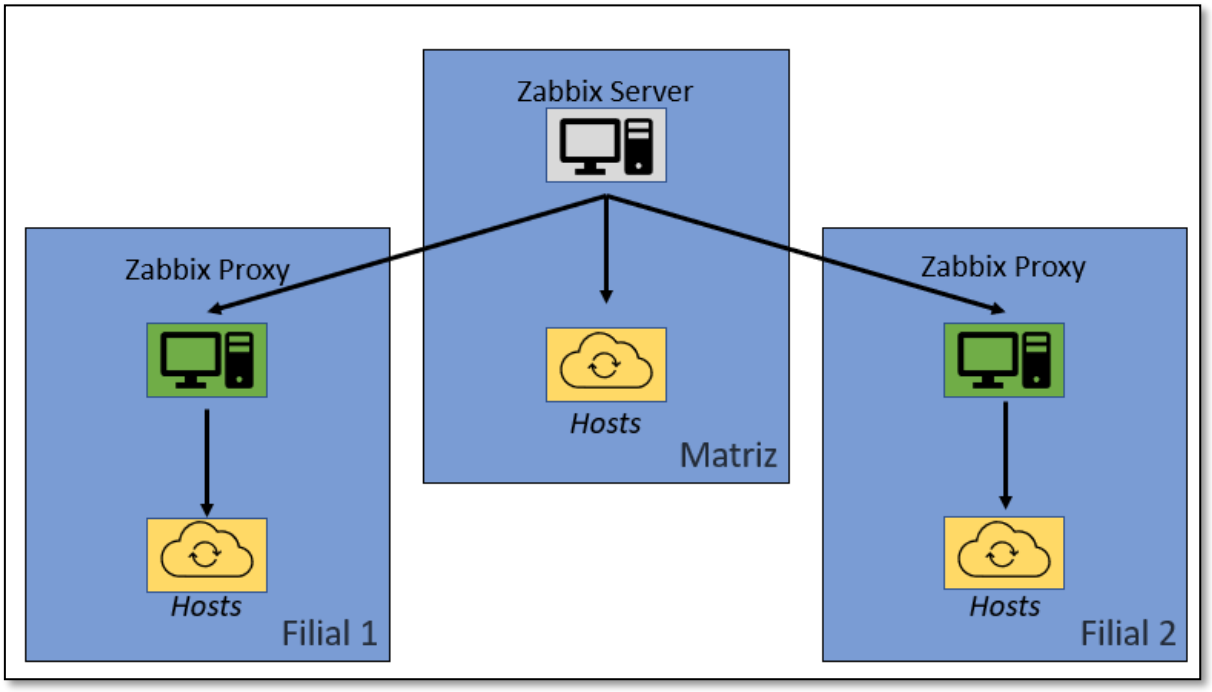
O servidor Zabbix gerencia um repositório central para configuração, estatísticas e armazenamento de dados operacionais. Dentre suas capacidades, ele pode realizar varreduras remotas de *hosts.*

A solução Zabbix é separada em três componentes, o Zabbix Server, *Web interface* e *database*. Todas as informações de configuração de monitoramento são armazenadas no banco de dados, tanto o Servidor quanto a *Interface Web* do Zabbix interagem com o Sistema de Gerenciamento de Banco de Dados (*Data Base Management* System – SGBD). (ZABBIX SIA, 2022)

4.7 Zabbix *Proxy*

O Zabbix *Proxy* é um processo que recebe dados de determinados *hosts* e os envia para o servidor Zabbix. A Figura 16 ilustra um ambiente que utiliza deste recurso. É possível notar que entre o Zabbix *Server* e os *Devices* (*hosts*) há o Zabbix *Proxy*. De acordo com a imagem, diversos *devices* estão sendo monitorados pelo processo, isto é, o Zabbix *Proxy* age como um servidor Zabbix para estes dispositivos, e o servidor Zabbix em si busca as informações direto dos *Proxy*. (Zabbix SIA, 2022)

Figura 16 - Exemplo de uso do Zabbix *Proxy*



Fonte: Autoria própria, adaptado de Zabbix SIA, 2022

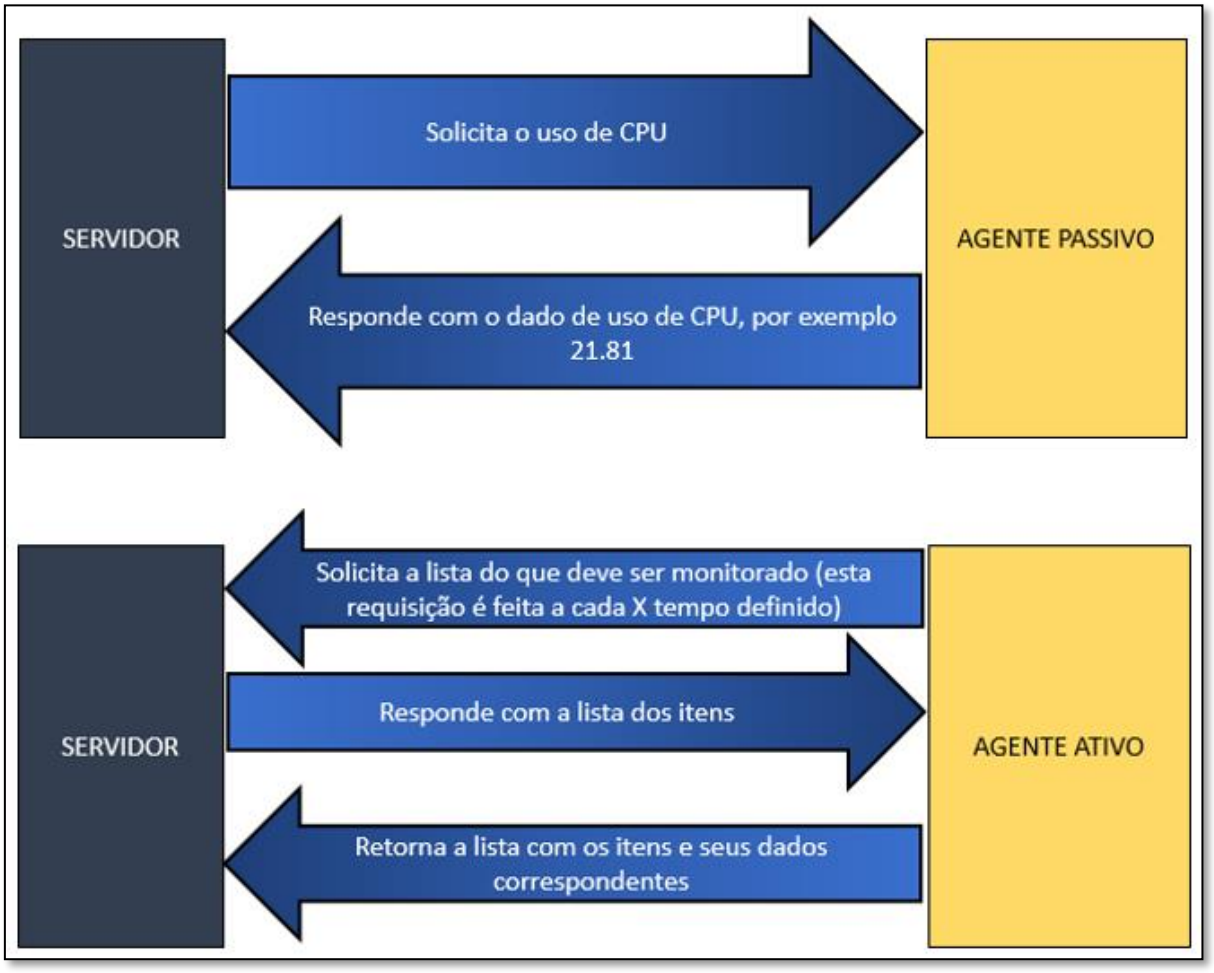
A utilização deste componente pode trazer benefícios para uma empresa. Dado que é possível haver um cenário no qual os dispositivos a serem monitorados estão distantes geograficamente. Um servidor *proxy* nestes locais acarreta uma distribuição da carga que normalmente iria toda para o servidor Zabbix. Isto ocorre porque o uso de processador e a Entrada e Saída (*Input/Output* - I/O) de dados diminui quando a coleta é realizada pelos *Proxies*.

4.8 Agente Zabbix

O agente Zabbix é um processo que roda no dispositivo a ser monitorado. Este processo coleta e guarda dados sobre o dispositivo, como o processador, uso de memória, espaço de armazenamento em disco etc. Os dados guardados são enviados para o servidor Zabbix ou para o *proxy* Zabbix.

Existem dois tipos de configuração de verificação nos agentes, a verificação passiva e a verificação ativa. No primeiro caso, o agente espera por uma requisição realizada pelo servidor Zabbix ou *proxy* Zabbix e a responde com o envio dos dados requisitados. Na verificação ativa, o agente recebe uma lista com os itens que devem ser monitorados e o intervalo da coleta de cada um deles. Desta forma, o agente pode realizar a coleta dos dados mesmo sem conexão com o servidor Zabbix, e enviar as informações posteriormente. A Figura 17 ilustra os dois tipos de verificação.

Figura 17 - Agente passivo e agente ativo



Fonte: Autoria própria, adaptado de Mauro 2001

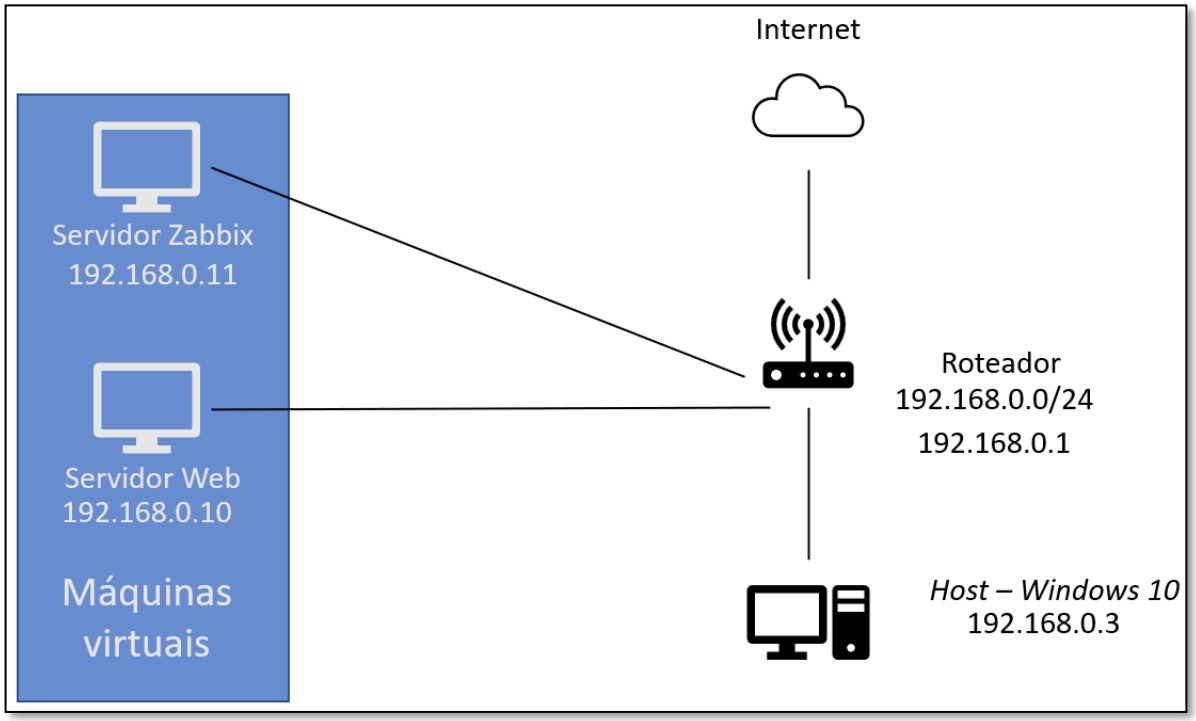
5 AMBIENTE DOS TESTES DE DISPONIBILIDADE DO SERVIDOR *WEB*

Neste capítulo são abordados a criação do ambiente e os testes práticos realizados. É descrito seu propósito, quais são os seus componentes, as configurações e o que ele proporciona. Para demonstrar como as ferramentas e soluções foram aplicadas, esse capítulo foi dividido em seções com as principais funcionalidades implementadas, cada qual descrita e ilustrada.

5.1 Sobre o ambiente

O propósito do ambiente é realizar o monitoramento de um serviço *Web* com foco em sua disponibilidade. O ambiente é formado por duas máquinas virtuais, a primeira é responsável pelo servidor Zabbix *All in One*, que integra um banco de dados e *interface Web*, e o segundo sistema roda o serviço *Web* através do servidor Apache. Ambas as máquinas têm o sistema operacional CentOS Stream 8 e são virtualizadas pelo *software* Oracle VM VirtualBox, programa que permite simular diferentes sistemas e/ou ambientes. A primeira máquina recebe o nome de CentOS 8 WEB SERVER (Servidor Apache) e a segunda máquina recebe o nome CentOS 8 ZABBIX SERVER (Servidor Zabbix). O nome foi criado neste padrão para facilitar a identificação de seu sistema operacional, versão e responsabilidade. Há uma máquina *desktop* que é utilizada para executar os dois servidores através do *software* de virtualização mencionado e para demonstrar a *interface Web* da ferramenta de monitoramento. Por fim, o último dispositivo da rede, o roteador, tem a função de prover a conexão entre os componentes da rede através de encaminhamento de pacotes de dados. A Figura 18 demonstra o ambiente descrito.

Figura 18 - Arquitetura do ambiente



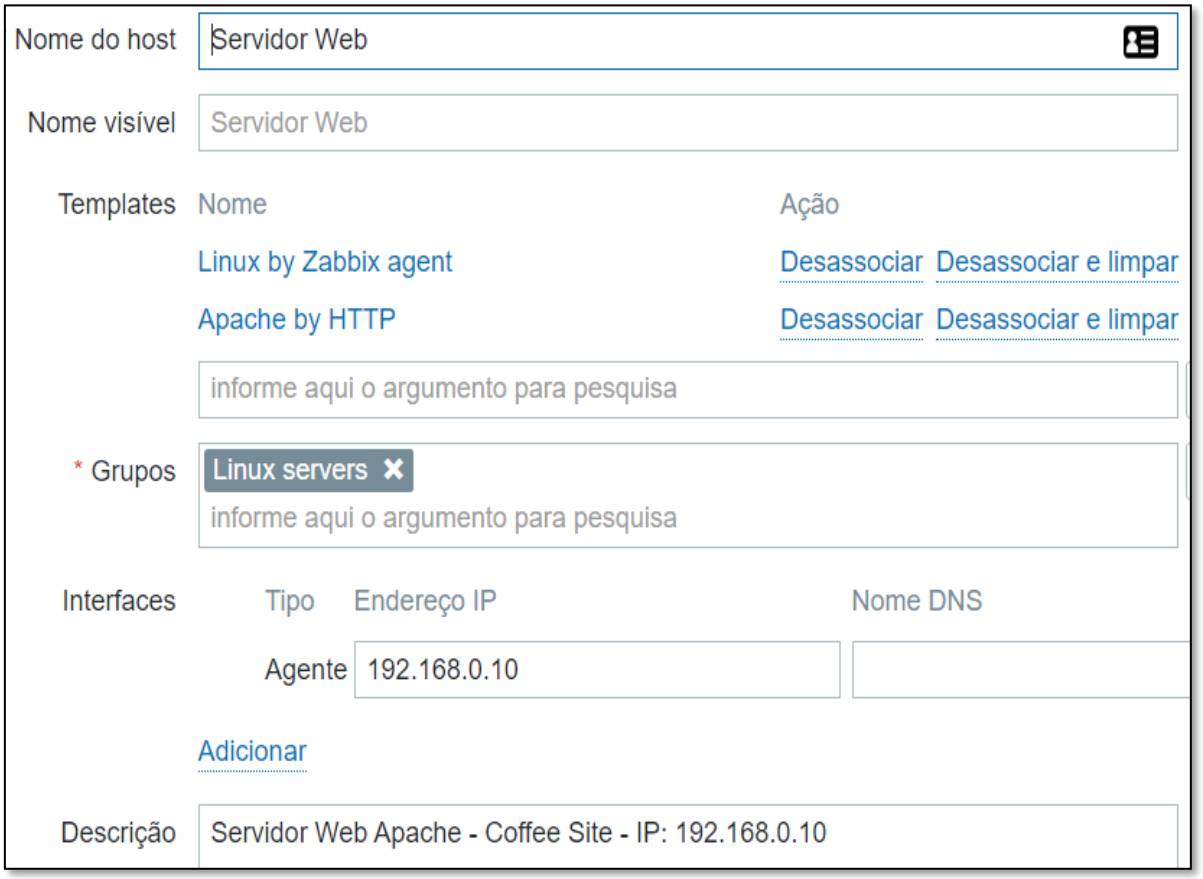
Fonte: Autoria própria

Visando melhor facilidade em manutenção e instalação, a topologia implementada foi do tipo estrela. Este tipo de topologia pode trazer benefícios como modificações sem complexidade nos dispositivos do ambiente, seja uma atualização, adição ou substituição de algum componente, ou facilidade ao buscar solução para alguma falha. Como desvantagem, a falha no dispositivo principal interrompe a comunicação do ambiente.

5.2 Monitoramento do servidor *Web*

O ambiente foi montado para verificar a disponibilidade de um serviço *Web* e aplicar medidas quando houver sua indisponibilidade. Desta forma, teve-se a preocupação de monitorar os serviços, portas e o próprio servidor para identificar ou antecipar falhas. A Figura 19 ilustra o *host* referente ao servidor *Web*.

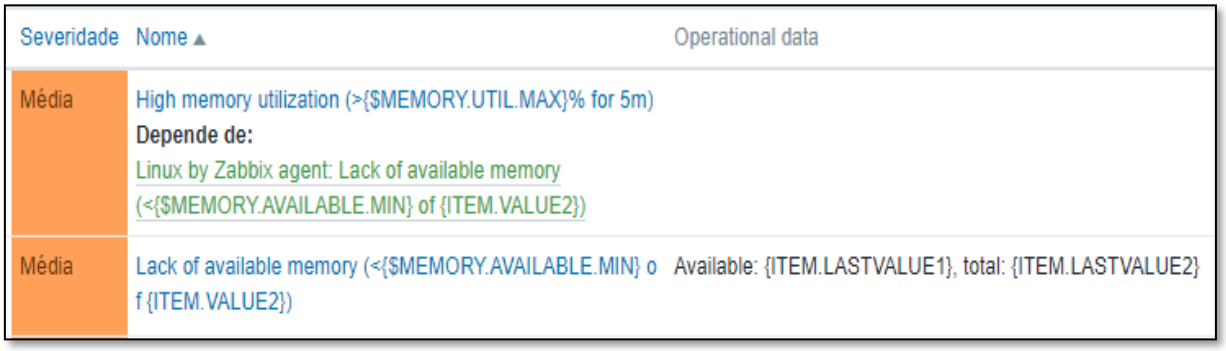
Figura 19 - *Host* Servidor Web



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

No Zabbix, o *host* Servidor *Web* está integrado nas *templates* "*Linux by Zabbix agent*" que possui diversos itens e *triggers* que habilitam o monitoramento do próprio servidor. Esta primeira *template* monitora itens do sistema como a CPU (uso do CPU, tempo ocioso entre outros), a memória RAM (total de memória, total em uso entre outros) e outros itens, como por exemplo, o agente do Zabbix. A *template Linux by Zabbix agent* possui *triggers*, ilustradas pela Figura 20 que são ativadas quando há alguma falha, isto é, alertas são emitidos quando o uso da Unidade Central de Processamento (*Central Processing Unit* - CPU) ou memória está acima de um padrão definido. Desta forma pode-se identificar que o servidor está trabalhando com alta carga. Em uma empresa, esta alta carga deve ser avaliada para identificar a origem deste problema. Caso a demanda esteja crescendo, pode ocorrer que a máquina não consiga atender maiores cargas.

Figura 20 - Exemplo de Triggers



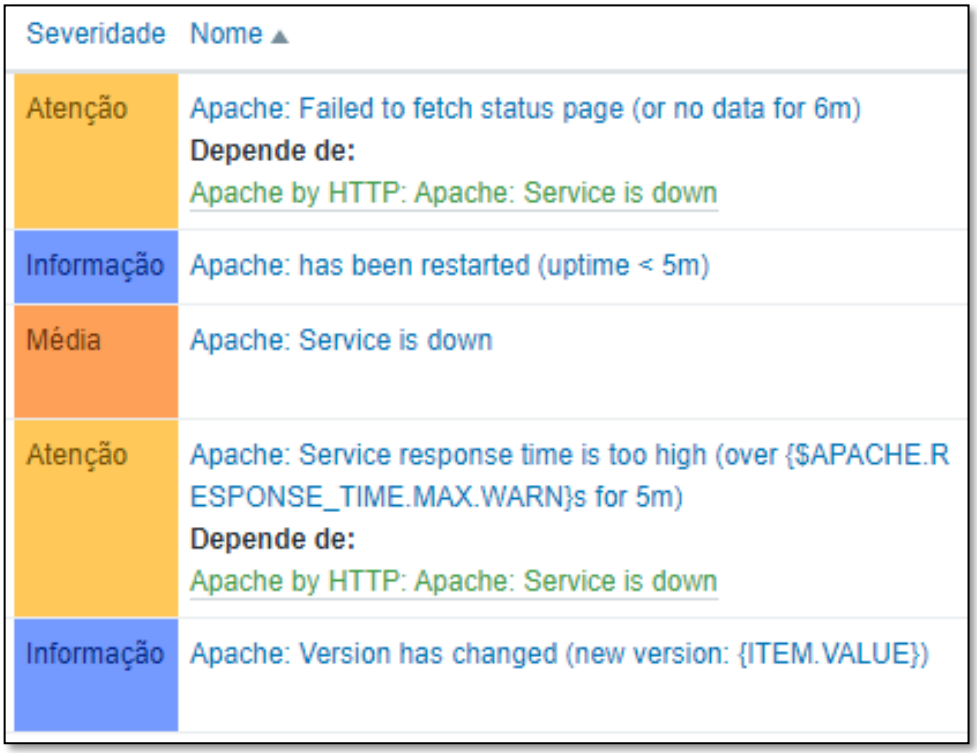
Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

5.3 Monitoramento do serviço Apache

Para monitorar o serviço Apache, que foi implementado com o propósito de colocar a página *Web* no ar, o *host* Servidor *Web* inclui a *Template Apache by HTTP*. Esta *Template* adiciona itens que monitoram o número de acessos, tempo ativo do serviço, a versão entre outras métricas. Seus gatilhos, ou *Triggers*, disparam em cinco ocasiões que são demonstradas pela Figura 21. Esses cinco gatilhos são ativados quando:

1. Falha na coleta de dados pelo Apache por 6 minutos seguidos;
2. Valor de *uptime* do Apache for menor do que 5 minutos;
3. Serviço Apache está fora do ar;
4. Tempo de resposta do Apache for maior do que a variável configurada por 5 minutos seguidos;
5. Versão do Apache sofreu mudanças.

Figura 21 - Autoria própria

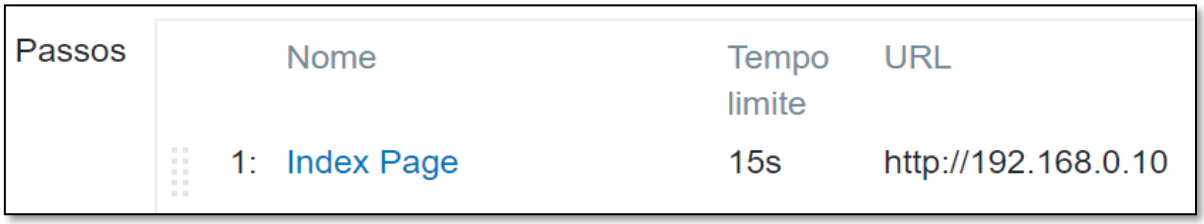


Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

5.4 Cenário *Web*

Por fim, a terceira configuração realizada no servidor *Web* foi o monitoramento *Web* denominado "Cenário *Web*", ilustrado pela Figura 22. Neste cenário, foi configurado um passo para verificar o Localizador Uniforme de Recursos (*Uniform Resource Locator* - URL) que dá acesso à página *Web* através de requisições HTTP. Podem ser configurados mais passos. O Zabbix segue a ordem deles e faz as requisições. Por padrão, um cenário *Web* coleta dados de velocidade média de *download*, número de falhas nas requisições (passos) e a última mensagem de erro.

Figura 22 - Cenário *Web*



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

A Figura 23 demonstra a coleta dos dados realizada pelo cenário *Web*, em que temos 6 itens sendo coletados. São eles:

1. Velocidade de *download* do cenário *Web*;
2. Velocidade de *download* do primeiro passo de cenário *Web*;
3. Valor de passo com falha do cenário *Web*;
4. Última mensagem de erro do cenário *Web*;
5. Código de resposta do primeiro passo do cenário *Web;*
6. Tempo de resposta do primeiro passo do cenário *Web.*

Figura 23 - Itens de Cenário *Web*

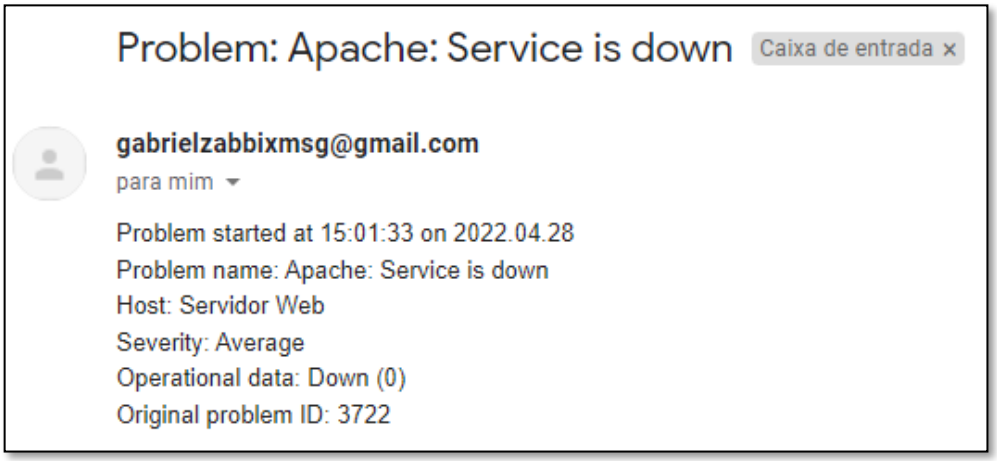
Host	Nome ▲	Última checagem	Último valor
Servidor Web	Download speed for scenario "Coffee Shop".	22s	14.97 MBps
Servidor Web	Download speed for step "Index Page" of scenario "Coffee Shop".	22s	14.97 MBps
Servidor Web	Failed step of scenario "Coffee Shop".	22s	0
Servidor Web	Last error message of scenario "Coffee Shop".	18h 29m 21s	Couldn't connect to serv...
Servidor Web	Response code for step "Index Page" of scenario "Coffee Shop".	22s	200
Servidor Web	Response time for step "Index Page" of scenario "Coffee Shop".	22s	1.69ms

Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

5.5 Alertas via *e-mail*

Além dos alertas que são mostrados através do *Dashboard* na seção de “Incidentes” do próprio Zabbix, a ferramenta permite configurações que possibilitam o envio destas mensagens para outras plataformas. Para este ambiente, foi configurado o envio de *e-mail*s para ajudar na visualização das falhas do ambiente. A Figura 24 ilustra um alerta de severidade do tipo “média” que foi emitido pelo Zabbix e enviado para o *e-mail* gabrielzabbixmsg@gmail.com. Este alerta indica que o serviço Apache está fora do ar.

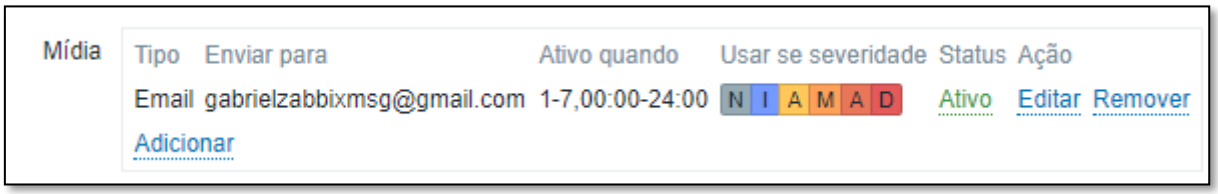
Figura 24 - *E-mail* de alerta



Fonte: imagem capturada pelo autor desse trabalho de Gmail, 2022.

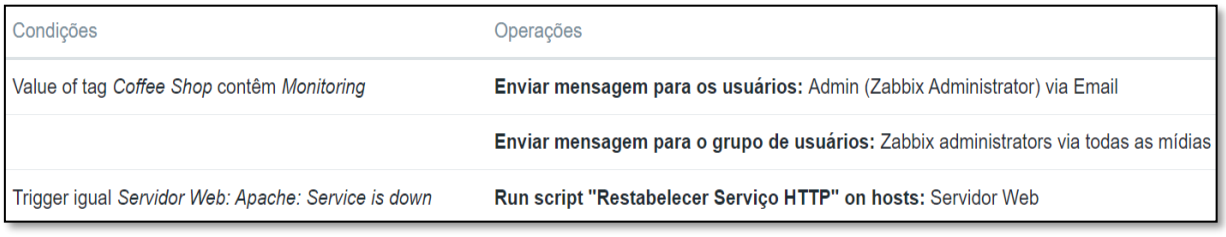
Para o funcionamento do envio de *e-mail*s, foi adicionado a mídia no usuário Admin do Zabbix, tal qual a Figura 25 ilustra. Essa figura demonstra que a mídia do tipo "*Email*" estará disponível todos os dias da semana e para todas as severidades do Zabbix para o usuário cadastrado, neste caso o usuário Admin. Além da adição da mídia, é necessário configurar uma ação em "*Trigger actions"* para habilitar o envio das mensagens mediante condições. A Figura 26 demonstra, resumidamente, como foi configurada a ação do gatilho. Especificamente, na sua primeira linha pode-se observar que a operação de envio do e-mail será realizada pelo administrador do Zabbix quando a condição for atendida, isto é, quando a *tag* do gatilho for "Coffee Shop" contendo o valor "*Monitoring*".

Figura 25 - Mídia *e-mail*



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

Figura 26 - Ação do gatilho



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

5.6 *Script* para restabelecimento de serviço

A ferramenta Zabbix permite a criação de *scripts* que rodam comandos remotos através do agente Zabbix, do *proxy* ou para o próprio servidor Zabbix. Desta forma, foi criado um *script* que roda quando o serviço Apache está com estado "*down*". Durante o monitoramento do Servidor *Web*, caso seja identificado uma falha no serviço referente a página *Web*, o *script* tenta enviar um comando remoto para o servidor que executa um comando para reinicialização do serviço Protocolo de Transferência de Hipertexto Daemon – *Hypertext Transfer Protocol Daemon* (HTTPD).

Em *Scripts* dentro de Administração encontra-se os *scripts* do Zabbix. Com o propósito de restabelecer o serviço HTTPD, o *script* demonstrado pela Figura 27 foi criado. Assim como o envio de *e-mail*s, é necessário configurar ações através de gatilhos para que o *script* seja enviado ao servidor remoto. Esta configuração é realizada em "*Trigger actions"*, ilustrada pela Figura 27. Na imagem, pode-se observar que o restabelecimento do serviço HTTP será realizado executando o comando "sudo systemctl restart httpd.service" através da ação "Restabelecer Serviço HTTP", ativada quando o Zabbix diagnostica a queda do serviço Apache.

Figura 27 - *Trigger actions*

Nome ▲	Scope	Usado nas ações	Tipo	Executar em	Comandos
Detect operating system	Manual host action		Script	Servidor (proxy)	`sudo /usr/bin/nmap -O {HOST.CONN}`
Ping	Manual host action		Script	Servidor (proxy)	`ping -c 3 {HOST.CONN}; case $? in [01]) true;; *) false;; esac`
Restabelecer Serviço HTTP	Action operation	Restabelecer Serviço HTTP	Script	Agente	`sudo systemctl restart httpd.service`
Traceroute	Manual host action		Script	Servidor (proxy)	`/bin/traceroute {HOST.CONN}`

Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

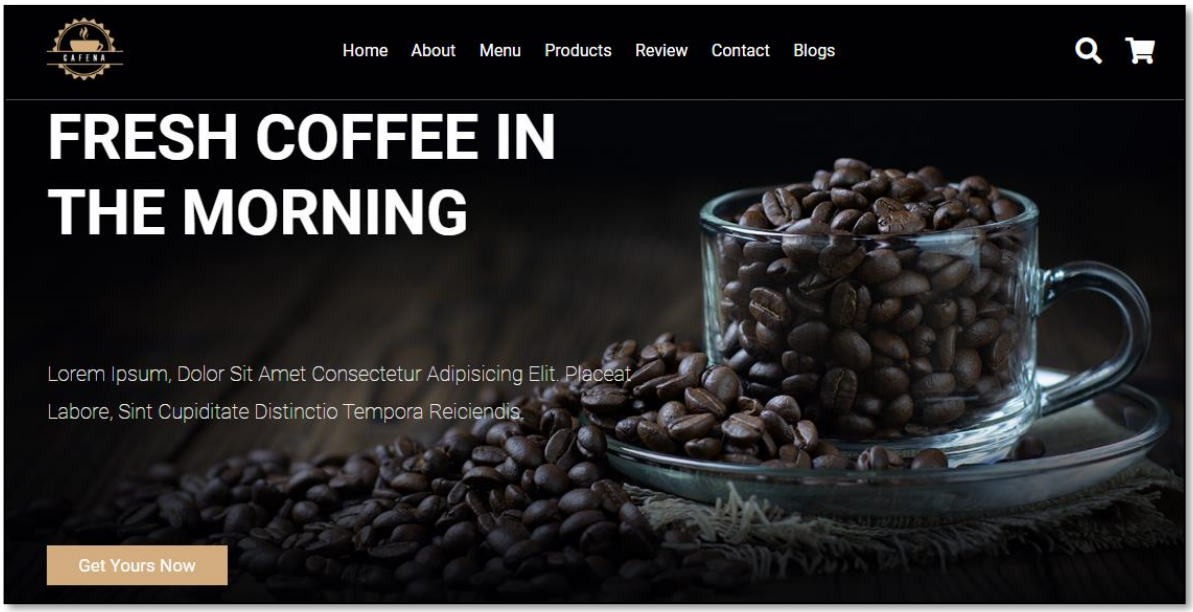
5.7 Servidor Apache

O servidor HTTP Apache foi criado em 1995 por Rob McCool e mantido pela Apache Software Foundation, é um servidor de código aberto que permite *site*s de *Internet* sejam disponibilizados. Seu uso é amplamente difundido no mundo, contendo cerca de 31.5% dos servidores ativos segundo pesquisa realizada pelo w3techs em 15 de maio de 2022. Dentre as vantagens de usar o Apache, as principais que foram levadas em consideração para usá-lo

foram: estabilidade, facilidade para configurá-lo, suporte para diversos sistemas operacionais e suporte amplo. Embora o produto da Apache Software Foundation tenha um problema conhecido como “c10k”, em que a partir de 10.000 conexões simultâneas há um problema de *performance*, no ambiente em questão, esta deficiência não é relevante. (W3TECHS, 2022)

A Figura 28 ilustra o *site* que foi hospedado no Servidor Apache. Esta página é monitorada pelo Servidor Zabbix. O objetivo dela é simular um serviço de compras *online* em que, caso haja indisponibilidade, as vendas dos produtos serão afetadas.

Figura 28 - Página *Web*



Fonte: Autoria própria

6 IMPLEMENTAÇÃO E TESTES DE DISPONIBILIDADE DO SERVIDOR *WEB*

Este capítulo trata sobre as simulações criadas para testar o ambiente criado, discorrendo sobre os objetivos a serem alcançados pelos cenários. Foram criados três cenários em que é observado o comportamento do ambiente em cada um deles, documentando as consequências de cada caso de falha.

6.1 Simulações

Com o propósito de testar como o Zabbix se comporta com relação a diversas falhas em relação a disponibilidade, foram criados casos de falha. Cada caso detalha os seguintes pontos:

- Qual falha ocorreu;
- Como esta falha afeta a disponibilidade do serviço *Web*;
- Como o Zabbix reagiu à falha.

As simulações realizadas foram pensadas em casos que poderiam ocorrer em um ambiente após um período de tempo. Em uma notícia de abril de 2022, a Conexis – Sindicato Nacional das Empresas de Telefonia e de Serviço Móvel, Celular e Pessoal, informou que mais de 4 milhões de metros de cabos de telecomunicações foram roubados ou furtados, que por consequência deixou mais de 6 milhões de clientes sem serviço em 2021. Uma interrupção no serviço de *Internet* por uma operadora afeta diretamente o serviço *Web* prestado por uma empresa. Desta forma, espera-se que as simulações criadas possam ajudar a identificar a origem das falhas que levam a indisponibilidade do serviço *Web*. (Conexis, 2022)

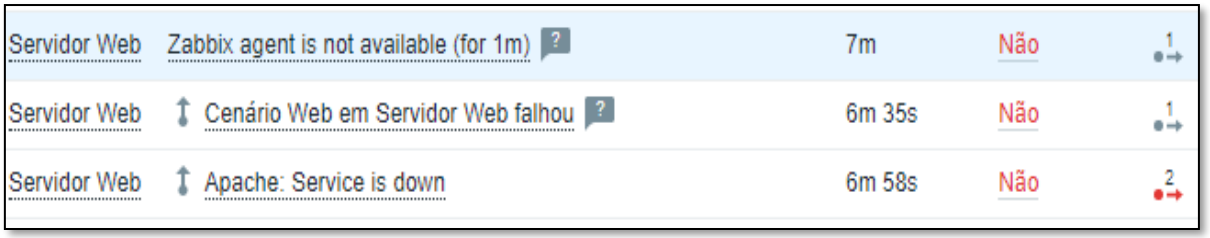
6.2 Incidente de perda de conexão com o servidor remoto

Para o primeiro teste no ambiente visando um possível caso de indisponibilidade, foi simulado uma queda de energia ou do serviço de *Internet*. A máquina virtual Servidor Web, responsável pela página *Web*, foi desligada através do *software* Oracle VM VirtualBox. Desta forma, há interrupção total do serviço e funcionamento da máquina enquanto o servidor Zabbix continua operante na máquina virtual Servidor Zabbix.

A Figura 29 demonstra os incidentes resultantes da falha de comunicação com o servidor. Pode-se observar que, os três gatilhos disparados foram: "*Zabbix agente is not available* (*for 1m*)", "Cenário *Web* em Servidor *Web* falhou" e "Apache: *Service is down*". A primeira *Trigger* indica que não há comunicação com o agente Zabbix no *host* Servidor *Web*, isto é, não necessariamente implica uma falha na disponibilidade do serviço *Web*, mas sim que houve uma falha na coleta de dados pelo agente Zabbix indicando uma falha no processo

zabbix-agent. O segundo gatilho, demonstrado pela Figura 29, informa que houve uma falha no Cenário *Web*, logo, o serviço *Web* foi interrompido. Por fim, o último alerta informa que o serviço Apache está inoperante.

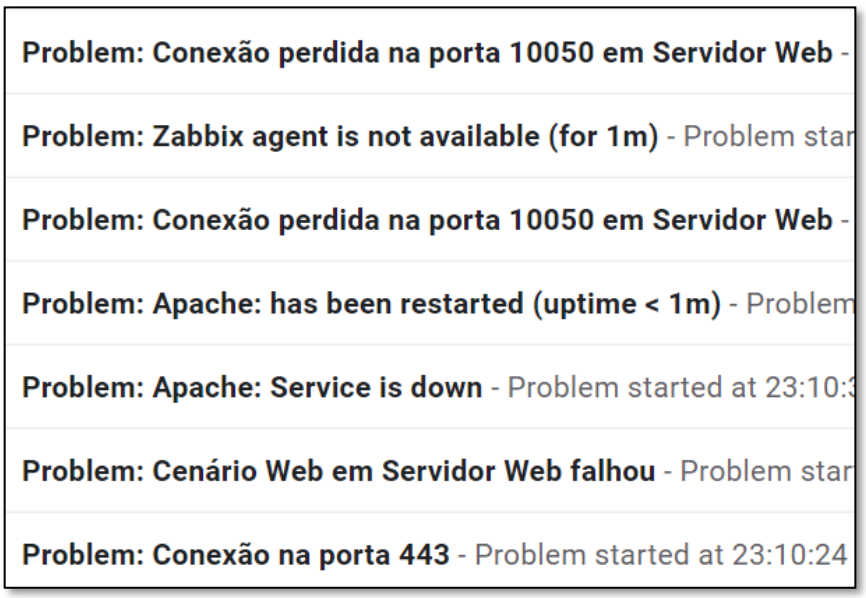
Figura 29 - Incidentes da primeira simulação



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

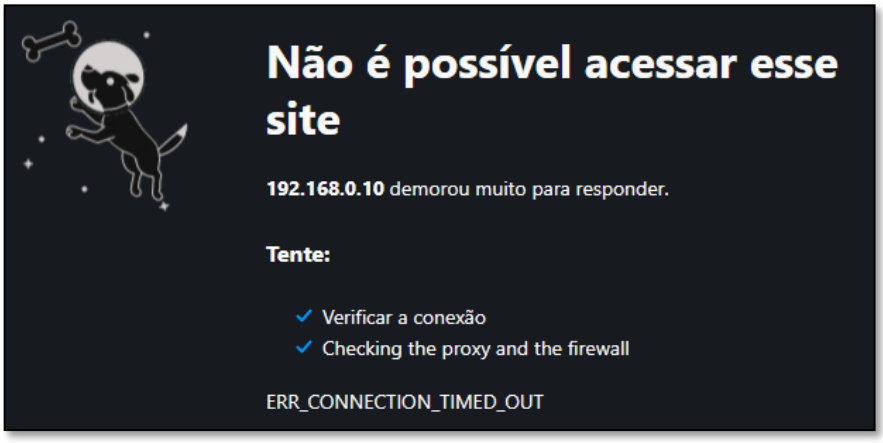
Além dos incidentes observados na Figura 29, pode-se verificar na quinta coluna, que para cada evento foi iniciada pelo menos uma ação. Os *e-mail*s foram enviados e podem ser observados pela Figura 30. Com o agente remoto inoperante, não foi possível realizar o *script* de reinicializar o serviço HTTPD remotamente. Ao realizar a tentativa de acesso à página *Web* a resposta é de falha, a Figura 31 demonstra a falha informada pelo navegador.

Figura 30 - *E-mails* de alerta



Fonte: imagem capturada pelo autor desse trabalho de Gmail, 2022.

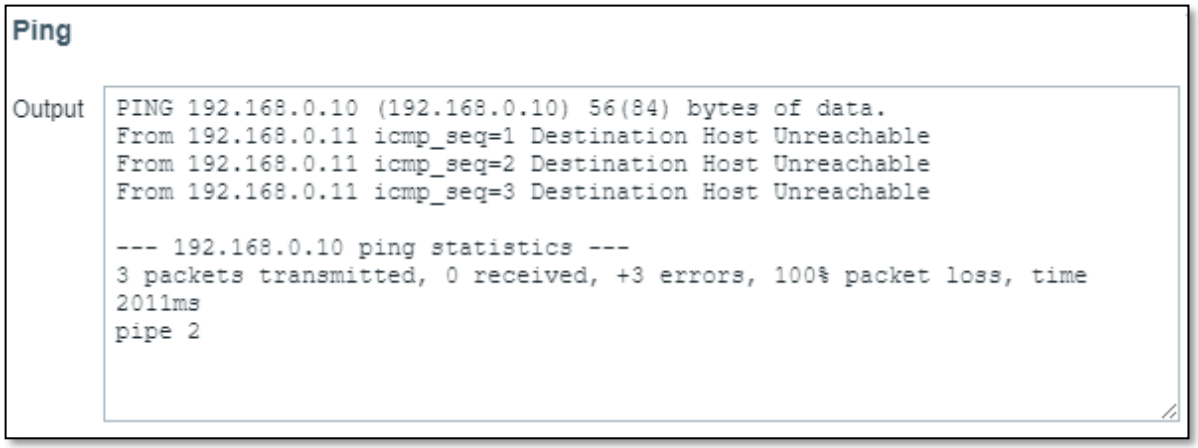
Figura 31 - Erro de acesso à página *Web*



Fonte: Recorte retirado do navegador Opera

Por fim, uma última tentativa de acesso a máquina foi através de um comando *ping*, que foi realizado pela própria ferramenta Zabbix e pode ser observado através da Figura 32. Como esperado, a mensagem de resposta informa que o *host* de destino está inalcançável após uma tentativa de envio de 3 pacotes transmitidos e todos com falha.

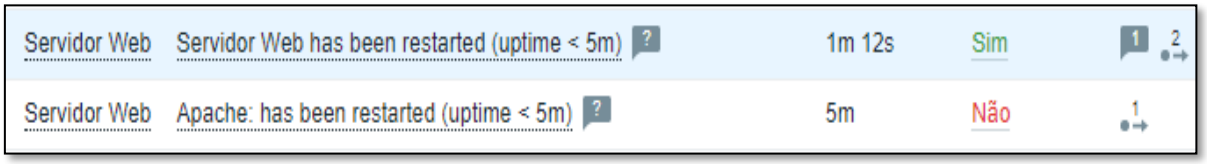
Figura 32 - Comando *ping*



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

Terminado os testes de indisponibilidade, a conexão com o servidor foi restabelecida e, como pode ser observado pela Figura 33, não há alertas de indisponibilidade. Os incidentes informados pela Figura 33, que podem ser observados na segunda coluna, informam o restabelecimento do serviço através das *triggers* “Servidor *Web has been restarted* (*uptime* < 5m)” e “Apache *has been restarted* (*uptime* < 5m)”.

Figura 33 - Restabelecimento do serviço

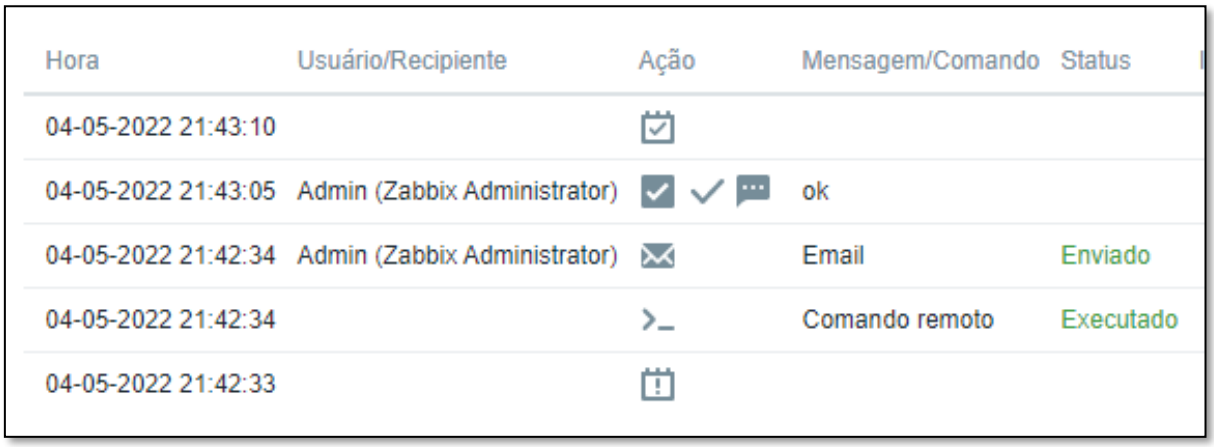


Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

6.3 Falha no processo HTTPD

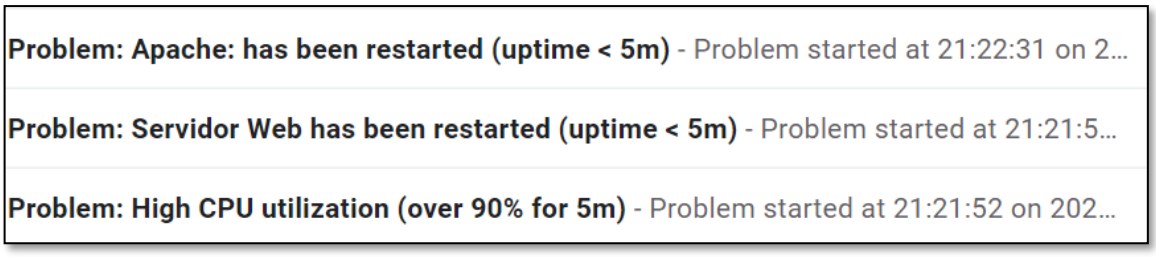
Para o segundo teste no ambiente, foi simulado uma queda no serviço HTTPD. Neste teste, devido a falha no Apache, a página *Web* ficou inoperante. Como pode ser observado pela Figura 34, houve três incidentes registrados no *host* Servidor *Web*. Por cerca de 1 minuto o Zabbix alertou que houve queda do serviço através das *Triggers* "*Apache: Service is down*" e "Cenário *Web* em Servidor *Web* falhou". O primeiro gatilho indica que o servidor Apache está inoperante, enquanto o segundo indica que o Cenário Web está recebendo falha ao tentar a conexão. Junto com estes alertas, foram realizadas duas ações: 1 - Enviar notificação via *e-mail*, demonstradas pelas Figura 34 e o recebimento das mensagens na Figura 35; 2 – Enviar um *script* remoto, que foi executado com sucesso e pode ser observado pela Figura 34 através das colunas "Mensagem/Comando" com valor "Comando remoto" e da coluna "Status" de valor "Executado".

Figura 34 - Incidentes HTTPD



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

Figura 35 - E-mails recebidos

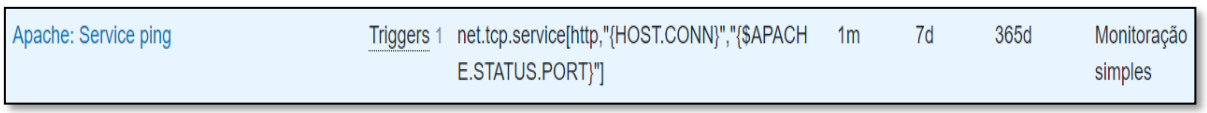


Fonte: imagem capturada pelo autor desse trabalho de Gmail, 2022.

Caso não houvesse o *script* e o Apache continuasse inoperante, certamente haveria uma demora maior no tempo de solução do problema. Dado que o agente Zabbix continuou a coleta de dados sem problemas, uma conclusão viável neste cenário é de que o servidor *Web* continuou operante, mas houve interrupção no processo HTTP que resultou na falha da coleta dos dados do processo Apache e de Cenário *Web*. Assim, o administrador de rede teria um maior esforço para restabelecer o serviço, ainda que o Zabbix tenha orientado no diagnóstico do problema enfrentado no ambiente.

A indisponibilidade da página *Web* pode ser percebida pelo usuário final, mesmo com o *script* sendo enviado e restabelecendo o serviço. Isto ocorre porque primeiramente o Zabbix deve detectar a falha no serviço, em seguida enviar os *e-mail*s e executar o *script* ao emitir o alerta e, finalmente, a máquina remota recebe o comando através do agente Zabbix e o executa. De acordo com a configuração encontrada em "*Apache by HTTP*", que se encontra ilustrada pela Figura 36, têm-se uma previsão de que a página pode ficar até 1 minuto inoperante devido ao tempo de coleta de "Apache: *Service ping*", localizado na quarta coluna com valor "1m". A *trigger* que verifica a indisponibilidade do Apache monitora a camada de transporte através do protocolo TCP e porta de número 80.

Figura 36 - Serviço de *ping* no Apache



Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

Em diversos testes com a execução do *script* após a queda do Apache, obteve-se os resultados que demonstram o tempo em que o serviço ficou indisponível e são ilustrados pela Figura 37. A duração da indisponibilidade pode ser observada na coluna "Duração" e seus valores variam em torno de 30 segundos até um minuto em sua grande maioria. Em outros testes, o serviço voltou a ficar operante após cerca de 10 segundos. A grande variação de tempo é causada pelo intervalo demonstrado na Figura 36.

Figura 37 - Tempo de indisponibilidade

Host	Incidente	Duração
Servidor Web	Apache: Service is down	1m
Servidor Web	Apache: Service is down	1m
Servidor Web	Apache: Service is down	32s
Servidor Web	Apache: Service is down	1m
Servidor Web	Apache: Service is down	22s
Servidor Web	Apache: Service is down	37s
Servidor Web	Apache: Service is down	1m
Servidor Web	Apache: Service is down	47s
Servidor Web	Apache: Service is down	57s
Servidor Web	Apache: Service is down	1m
Servidor Web	Apache: Service is down	27s
Servidor Web	Apache: Service is down	37s

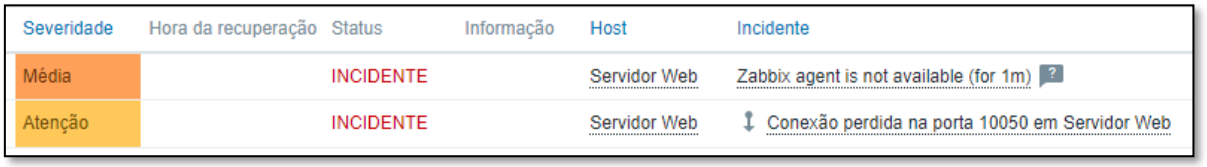
Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

6.4 Falha no agente Zabbix no servidor *Web*

Para o último caso de falha, foi testada uma parada no serviço do agente Zabbix, especificamente o serviço de nome “zabbix-*agent.service*”. Neste cenário, espera-se que a página *Web* não sofra de indisponibilidade e que o servidor *Web* continue operante.

O *dashboard* do Zabbix alertou sobre dois eventos, são eles a falha de conexão na porta 10050 e a indisponibilidade de seu agente no servidor remoto. A Figura 38 demonstra os alertas informados. Não houve indisponibilidade da página *Web* em nenhum momento deste teste, afirmando que os gatilhos corretos foram disparados. Embora a falha de comunicação na porta 10050 possa decorrer de outros problemas, como por exemplo, um possível bloqueio no *Firewall*, pode-se diagnosticar que a comunicação não ocorre devido a um problema principalmente no agente do Zabbix.

Figura 38 - Alertas sobre o agente Zabbix



Severidade	Hora da recuperação	Status	Informação	Host	Incidente
Média		INCIDENTE		Servidor Web	Zabbix agent is not available (for 1m)
Atenção		INCIDENTE		Servidor Web	Conexão perdida na porta 10050 em Servidor Web

Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

A importância deste teste pode ser observada na coleta de dados da máquina CentOS 8 WEB SERVER. Na Figura 39, é possível observar que os itens de componentes do sistema (implementados pela *Template Linux by Zabbix agent*) tem a sua coleta interrompida - A coluna "Última checagem" mostra que a última coleta foi realizada há 2 minutos e 48 segundos, quando o tempo deveria ser de no máximo 1 minuto. Enquanto os itens que dependem de outros serviços para coleta continuaram recebendo dados, por exemplo os itens dependentes da *Template Apache by HTTP*. Assim, foi possível diagnosticar a rede através de alertas que remontaram para a indisponibilidade parcial do servidor remoto.

Figura 39 - Itens dos componentes do sistema

Nome ▲	Última checagem
/: Free inodes in %	2m 48s
/: Space utilization	2m 46s
/: Total space	2m 44s
/: Used space	2m 42s
/boot: Free inodes in %	2m 47s
/boot: Space utilization	2m 45s
/boot: Total space	2m 43s
/boot: Used space	2m 41s
Apache: Bytes per request	48s
Apache: Bytes per second	48s
Apache: Connections async closing	48s
Apache: Connections async keep alive	48s
Apache: Connections async writing	48s
Apache: Connections total	48s
Apache: Get status	48s

Fonte: imagem capturada pelo autor desse trabalho de Zabbix, 2022.

A *Template* Apache *by* HTTP utilizada no *host* “Servidor *Web*” foi pensada para este caso de falha no agente. Uma segunda opção seria utilizar a *Template* Apache *by Zabbix agent*, mas, como observado nesta simulação, esta *template* falharia juntamente da *Template Linux by Zabbix agent* pois ambas dependeriam do funcionamento do agente Zabbix para a coleta. Logo, uma parada no agente na máquina remota acionaria gatilhos que remeteriam a falha na coleta de dados pelo Apache, o que pode influenciar o administrador da rede a diagnosticar a falha do serviço *Web*, quando não é o caso.

7 CONSIDERAÇÕES FINAIS

Neste trabalho, foi desenvolvido um ambiente simulado com o intuito de gerenciá-lo com foco na disponibilidade de um serviço *Web*. Foi utilizado a ferramenta Zabbix, agregado com conhecimentos em gerenciamento de rede, com o objetivo de verificar eventuais indisponibilidades do serviço, apontando a origem do problema e, possivelmente, restabelecer de forma automática com o auxílio de *scripts* o servidor Apache.

Devido ao aumento da complexidade de redes provocada pelo crescimento delas ao longo do tempo, tornou-se necessário a criação do gerenciamento de redes. O gerenciamento em uma rede deve ser capaz de monitorar, analisar, testar, configurar, entre outras atividades, com o intuito de auxiliar o administrador a tomar decisões de forma rápida e correta para qualquer eventual problema. Assim como um médico deve analisar os sintomas de um paciente para diagnosticá-lo e aplicar um tratamento, um gerente de rede, aliado a ferramentas, deve ser capaz de analisar uma rede, diagnosticá-la e aplicar correções, sejam elas reativas ou proativas.

O gerenciamento de redes tem uma arquitetura composta por três componentes principais, são eles a entidade gerenciadora, os dispositivos gerenciados e um protocolo de transporte. A entidade gerenciadora é o centro da atividade, responsável por atividades como coleta, processamento, análise e apresentação das informações. Os dispositivos gerenciados são os equipamentos que estão submetidos a coleta dos dados realizados pelo gerenciador. Por fim, o protocolo executa o transporte das informações, em outras palavras realiza a comunicação, entre os dispositivos gerenciados e a entidade gerenciadora.

O SNMP é o protocolo padrão utilizado para a transmissão dos dados no gerenciamento. É um protocolo com apenas duas operações com capacidade de transmitir desde coleta de dados até comandos na rede. Ao longo dos anos, ele se aperfeiçoou permitindo a distribuição do gerenciamento da rede e, em sua terceira versão, garantiu parâmetros de segurança importantes.

Para a criação do ambiente, foi utilizado o *software Oracle VM VirtualBox*. Com ele, foi possível a criação de duas máquinas virtuais, ambas rodando o sistema operacional CentOS Stream 8. Sendo a primeira máquina o servidor responsável pelo gerenciamento da rede, utilizando hospedando o servidor Zabbix *All in One*, e o segundo sistema responsável por executar o serviço *Web*, através do servidor Apache.

A ferramenta escolhida para a realização do gerenciamento do ambiente criado foi o Zabbix. Sua escolha se justifica devido a sua simplicidade de uso e as suas capacidades. Através de uma *interface Web*, é possível acessar e realizar todo o monitoramento da rede. Dentre as capacidades do Zabbix, ele permite a manipulação das configurações e integração com outras

ferramentas. Para a metodologia escolhida, esta ferramenta se provou muito útil visto que ela possui a capacidade de se adaptar as condições induzidas na simulação, bem como a disposição de um histórico com dados e incidentes.

Para alertar o administrador da rede, além dos eventos gerados no *dashboard* da ferramenta Zabbix, foi configurado o envio de notificações através do Gmail.

Foi criado um *script* para automatizar o processo de recuperação do serviço HTTPD para quando o Zabbix detectasse a queda dele. A vantagem dessa ação é realizar a contramedida o mais rápido possível, podendo ser realizada antes mesmo da percepção do usuário ou do administrador em uma eventual queda do serviço.

Com o objetivo de testar as diversas configurações aplicadas no Zabbix, dentre elas o monitoramento, alertas e ações, foram criados cenários de falha para investigar como a ferramenta se comportava e, principalmente, como isso contribuiu para a disponibilidade do serviço *Web*.

Cenários de falha são simulações de falhas que poderiam possivelmente ocorrer em um servidor *Web* no mundo real. Em um cenário de falha, um incidente é forçado no ambiente com o intuito de verificar se o Zabbix foi capaz de identificar onde ocorreu a falha, como ocorreu a recuperação, o que foi afetado, entre outras atividades realizadas pela ferramenta, passando pelos diversos alertas emitidos pelo Zabbix de forma a auxiliar o administrador a identificar os problemas e a tomar decisões.

O primeiro cenário de falha ocorreu com o desligamento do servidor *Web*. Para este caso, o Zabbix disparou diversos gatilhos que remeteram a falha tanto no seu agente instalado no servidor remoto quanto nas checagens simples. Visto que houve uma perda de conexão generalizada na máquina remota, concluiu-se que houve um desligamento ou queda no *link* de *Internet*.

O segundo caso de falha ocorreu com a falha no serviço HTTPD, responsável pelo funcionamento do servidor Apache. De imediato, o *software* alertou de falha no Cenário *Web* e no serviço apache. Conforme configurado previamente, a própria ferramenta executou um comando remoto no Servidor *Web*, restabelecendo com sucesso o serviço HTTPD. Logo, o Zabbix foi capaz de identificar onde ocorreu a falha e restabelecer o serviço automaticamente. Finalmente, como a falha foi centralizada no Apache, os dados não relacionados ao serviço HTTPD continuaram a serem recolhidos normalmente.

O último caso de falha foi o único não relacionado diretamente com a indisponibilidade do serviço *Web*. Nesta simulação, o agente Zabbix teve seu processo interrompido. Assim, a página *Web* continuou a operar normalmente, mas os dados referentes ao sistema operacional

tiveram sua coleta interrompida. Como observado nos testes realizados, a ferramenta alertou sobre a indisponibilidade do agente, mas a coleta dos dados referentes ao Apache continuou operando sem problemas. A consequência desse cenário normalmente será percebida a longo prazo quando ocorrer problemas relacionados a falta de memória, alto uso de processador, espaço de armazenamento cheio, entre outros. Logo, a importância desse teste se dá na prevenção de futuros problemas que possam vir a ocorrer.

Concluindo, este trabalho cumpriu seu objetivo, sendo capaz de detectar falhas que impactam diretamente ou indiretamente no serviço *Web*. Respondendo a questão de pesquisa, a capacidade da ferramenta Zabbix se provou capaz de atingir as metas, realizando um monitoramento competente, agregado a mecanismos de notificação e reação automática a eventos, possuindo ainda a possibilidade de integração com ferramentas de terceiros.

7.1 Sugestão de trabalhos futuros

- Integrar o Zabbix com outras ferramentas de notificação, como por exemplo, o Telegram ou o Discord;
- Implementar o certificado SSL visando maior segurança nas transações eletrônicas;
- Integrar o Zabbix com o Grafana para obter melhor visibilidade dos dados coletados.

REFERÊNCIAS

CISCO. **SNMP Configuration Guide, Cisco IOS XE Release 3S**. Disponível em: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/snmp/configuration/xe-3s/snmp-xe-3s-book/nm-snmp-cfg-snmp-support.html. Acesso em: 10 maio 2022.

CODATA. **CONHEÇA o Zabbix, *software* para ambientes de monitoramento de TI**. 8 out. 2018. Disponível em: https://codata.pb.gov.br/noticias-da-codata/conheca-o-zabbix-*software*-para-ambientes-de-monitoramento-de-ti#:~:text=História%20Zabbix-,O%20Zabbix%20é%20um%20*software*%20de%20monitoramento%20Open%20Source%20(totalmente,primeira%20versão%20estável%20da%20ferramenta. Acesso em: 14 maio 2022.

CONEXIS. **Mais de 4 milhões de metros de cabos de telecomunicações foram roubados ou furtados em 2021**. Abr. 2022. Disponível em: https://conexis.org.br/mais-de-4-milhoes-de-metros-de-cabos-de-telecomunicacoes-foram-roubados-ou-furtados-em-2021/. Acesso em: 10 maio 2022.

Advanced Host Monitor. Versão: 13.30. Advanced Network Software, 2022.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a *Internet*: uma abordagem top-down**. 6. ed. São Paulo: Pearson Education do Brasil, 2013. ISBN 978-85-430-1443-2.

MAURO, Douglas R.; SCHMIDT, Kevin J. **SNMP Essencial: Ajuda para os Administradores de Sistemas e de Redes**. 1. ed. Rio de Janeiro: Campus, 2001. 336 p. ISBN 978-8535208825.

ORACLE. **Oracle VM VirtualBox**. [S. l.], 2022. Disponível em: https://download.virtualbox.org/virtualbox/6.1.32/VirtualBox-6.1.32-149290-Win.exe. Acesso em: 21 maio 2022.

RFC 1067. **Simple Network Management Protocol**. Ago. 1998. Disponível em: https://datatracker.ietf.org/doc/html/rfc1067. Acesso em: 12 maio 2022.

RFC 1155. **Structure and identification of management information for TCP/IP-based *Internet*s**. Maio 1990. Disponível em: https://datatracker.ietf.org/doc/html/rfc1155. Acesso em: 12 maio 2022.

RFC 1157. **A Simple Network Management Protocol (SNMP)**. Maio 1990. Disponível em: <https://tools.ietf.org/html/rfc1157> Acesso em: 12 maio. 2022.

RFC 1212. **Concise MIB definitions**. Mar. 1991. Disponível em: https://datatracker.ietf.org/doc/html/rfc1212. Acesso em: 12 maio 2022.

RFC 1213. **Management Information Base for Network Management of TCP/IP-based *Internet*s: MIB-II**. Mar. 1991. Disponível em: https://datatracker.ietf.org/doc/html/rfc1213. Acesso em: 12 maio 2022.

RFC 2578. **Structure of Management Information Version 2 (SMIv2)**. Abr. 1999. Disponível em: https://datatracker.ietf.org/doc/html/rfc2578. Acesso em: 12 maio 2022.

SANTOS, Mauro Tapajós et al. **Gerência de Redes de Computadores**. 2. ed. Rio de Janeiro: Escola Superior de Redes, 2015. 320 p.

SAYDAM, T.; MAGEDANZ, T. **From Networks and Network Management into Service and Service Management**. Journal of Networks and System Management, v.4, n.4 (dez. 1996), p. 345–348.

STALLINGS, William et al. **Redes e Sistemas de Comunicação de Dados: Teoria e aplicação corporativas**. 5. ed. Rio de Janeiro: Elsevier, 2005. ISBN 85-352-1731-2.

TELCOMANAGER. **O que é SNMP**. Disponível em: https://www.telcomanager.com/blog/o-que-e-snmp/#:~:text=O%20SNMP%20ajuda%20o%20gestor,e%20memória%20de%20diversos%20dispositivos. Acesso em: 9 maio 2022.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. 2. ed. Rio de Janeiro: Elsevier, 2014. 145 p. ISBN 978-85-352-7782-1.

ZABBIX SIA. **Zabbix 6**. [S. l.], 2022. Disponível em: https://www.zabbix.com/br/download. Acesso em: 21 maio 2022.

ZABBIX SIA. **3 Funcionabilidades do Zabbix, 2022**. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual/introduction/features/>. Acesso em: 7 mai. 2022.

ZABBIX SIA. **5 *Triggers*, 2022**. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual/appendix/*triggers*/>. Acesso em: 7 mai. 2022.

ZABBIX SIA. **8 Monitoração *Web*, 2022**. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual/*Web*_monitoring>. Acesso em: 8 mai. 2022.

ZABBIX SIA. **5 Verificações simples, 2022**. Disponível em: <https://www.zabbix.com/documentation/current/pt/manual/config/items/itemtypes/simple_checks>. Acesso em: 8 mai. 2022.

ZABBIX SIA. **Zabbix Documentation 6.0**. Disponível em: https://www.zabbix.com/documentation/current/pt/manual. Acesso em: 8 mai. 2022.

ZABBIX SIA. **Zabbix 6.0: The Enterprise-Class Open Source Network**. [S. l.], 2022. Disponível em: https://www.zabbix.com/download?zabbix=6.0. Acesso em: 8 mai. 2022.

W3TECHS. **Usage Statistics and Market Share of Apache**, May 2022. 15 maio 2022. Disponível em: https://w3techs.com/technologies/details/ws-apache. Acesso em: 15 maio 2022.

ANEXO A – INSTALAÇÃO DO APACHE

Sistema operacional: CentOS Stream 8

Requisitos mínimos:

- Sistema operacional: GNU/Linux, Windows, MacOS
- Memória RAM: 4GB
- Espaço em disco: 10GB

Passo 1 – Instalar pacote HTTPD e digitar no Terminal:

```
sudo dnf install httpd
```

Passo 2 – Habilitar o serviço httpd no firewall

```
Terminal → sudo firewall-cmd --permanent –add-service=http
Terminal → sudo firewall-cmd --reload
```

Passo 3 – Adicionar o arquivo html no diretório /var/www/html/

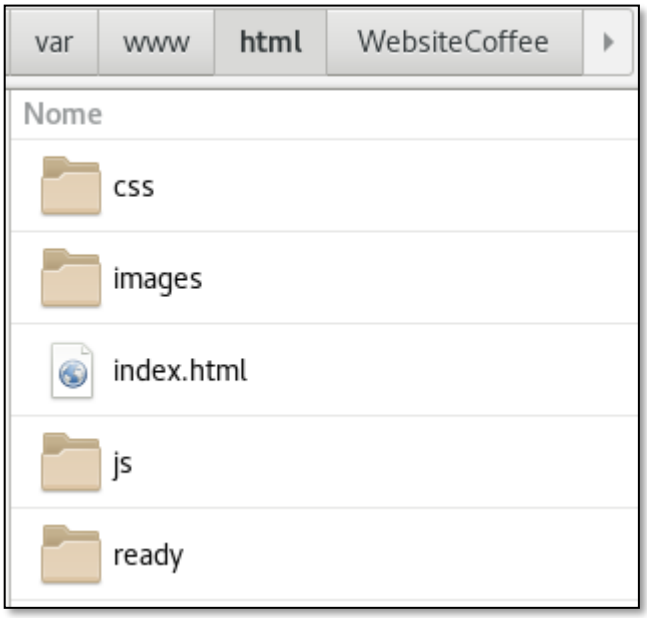


Figura A. 1 - Recorte de tela de CentOS Stream 8 com o arquivo html

Passo 4 – Habilitar o monitoramento de server-status

```
Terminal → cd /etc/httpd/
Terminal → vim conf.modules.d/00-base.conf
```

Passo 4.1 – Descomentar a linha a seguir do arquivo 00-base

```
LoadModule status_modules/mod_status.so
```

Passo 4.2 – Criar o bloco de texto no arquivo httpd.conf

```
Terminal → vim /etc/httpd/conf/httpd.conf
```

Adicionar o bloco de texto no arquivo httpd.conf:

```
<location /server-status>
      SetHandler server-status
      Order deny, allow
      Deny from all
      Allow from 192.168.0.11
</location>
```

ANEXO B – INSTALAÇÃO DO ZABBIX 6

Sistema operacional: CentOS Stream 8

Requisitos mínimos:

- Sistema operacional: GNU/Linux, Windows, MacOS
- Memória RAM: 128MB (Variável de acordo com o tamanho do ambiente)
- Espaço em disco: 256GB (Variável de acordo com o tamanho do ambiente)
- Banco de dados: MySQL/MariaDB

Passo 1 – Instalação do Zabbix 6

```
sudo rpm -Uvh
https://repo.zabbix.com/zabbix/6.0/rhel/8/x86_64/zabbix-
release-6.0-1.el8.noarch.rpm

dnf clean all

dnf -y install zabbix-server-mysql zabbix-Web-mysql zabbix-
apache-conf zabbix-sql-scripts zabbix-selinux-policy zabbix-
agent
```

Passo 2 – Instalação do banco de dados

```
curl -LsS -O
https://downloads.mariadb.com/MariaDB/mariadb_repo_setup

sudo bash mariadb_repo_setup --mariadb-server-version=10.6
dnf -y install mariadb-server && systemctl start mariadb &&
systemctl enable mariadb
```

Passo 3 – Configuração de senha do banco de dados

```
mariadb-secure-installation
```

```
Enter current password for root (enter for none): Press Enter
Switch to unix_socket authentication [Y/n] y
Change the root password? [Y/n] y
New password: <Enter root DB password>
Re-enter new password: <Repeat root DB password>
Remove anonymous users? [Y/n]: Y
Disallow root login remotely? [Y/n]: Y
Remove test database and access to it? [Y/n]:  Y
Reload privilege tables now? [Y/n]:  Y
```

Figura B. 1 – Demonstração da configuração da senha por Blog Best Monitoring Tools

Passo 4 – Criar o banco de dados

```
sudo mysql -uroot -p'rootDBpass' -e "create database zabbix
character set utf8mb4 collate utf8mb4_bin;"

sudo mysql -uroot -p'rootDBpass' -e "grant all privileges on
zabbix.* to zabbix@localhost identified by 'zabbixDBpass';"
```

Passo 5 – Importando dados iniciais

```
sudo vim /etc/zabbix/zabbix_server.conf
```

Adicionar a senha desejada no arquivo zabbix.server.conf

```
DBPassword=senhaDesejada
```

Passo 6 – Iniciar o servidor e o agente Zabbix

```
systemctl restart zabbix-server zabbix-agent
systemctl enable zabbix-server zabbix-agent
```

Passo 7 – Configurar o *firewall*

```
firewall-cmd --add-service={http,https} --permanent
firewall-cmd --add-port={10051/tcp,10050/tcp} --permanent
firewall-cmd –reload
```

ANEXO C – CONFIGURAÇÃO MÍDIA DO GMAIL

Passo 1 – Acessar “Tipos de mídias” dentro de “Administração” no *frontend* do Zabbix. É disponibilizada uma listagem com as mídias disponíveis pelo Zabbix.

Passo 2 – Editar a mídia “*Email*”, configurando os campos de acordo como a Figura C.2 demonstra.

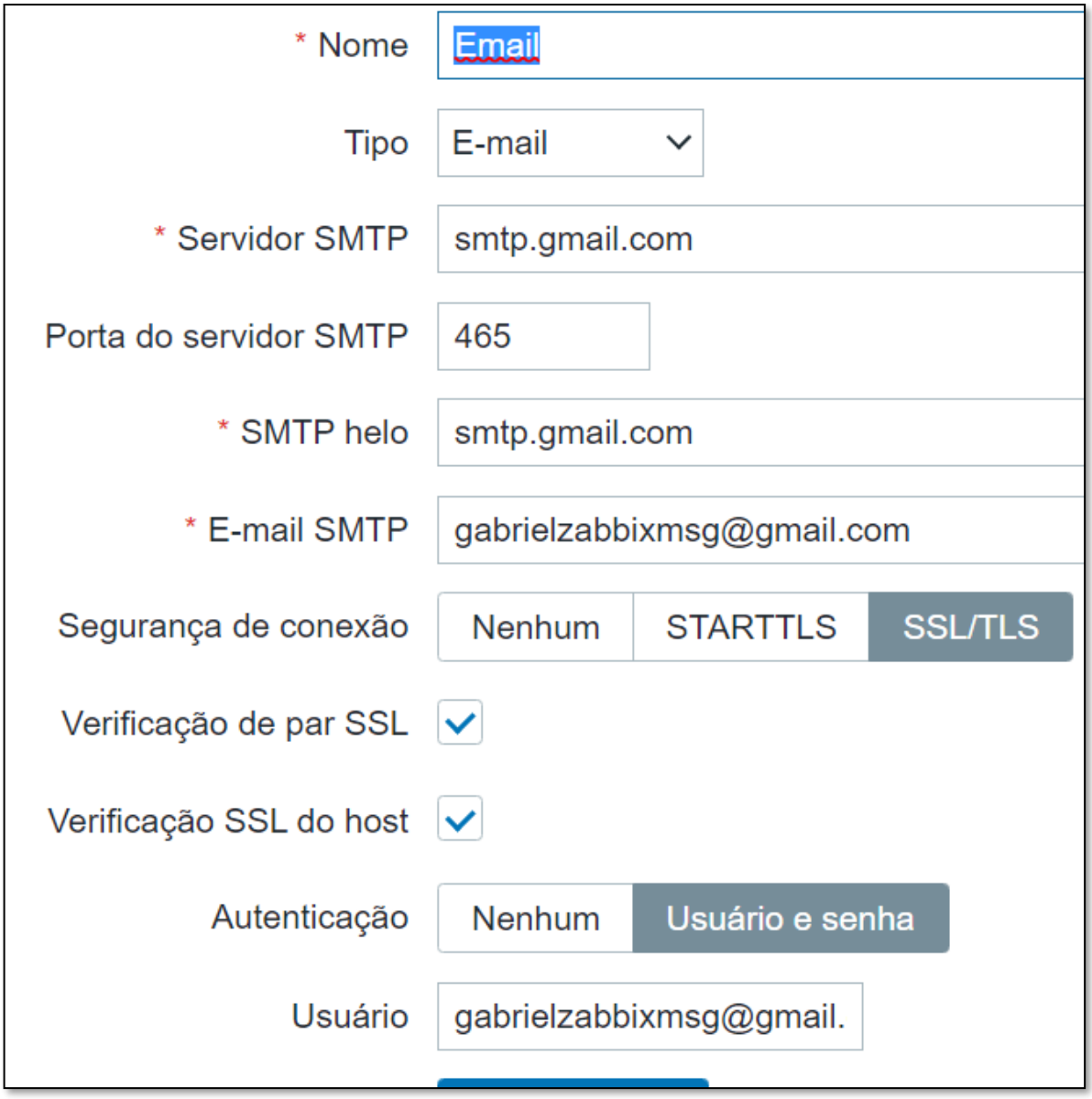


Figura C. 1 – Configuração do tipo de mídia Email

Passo 3 – Adicionar a mídia *Email* no usuário desejado

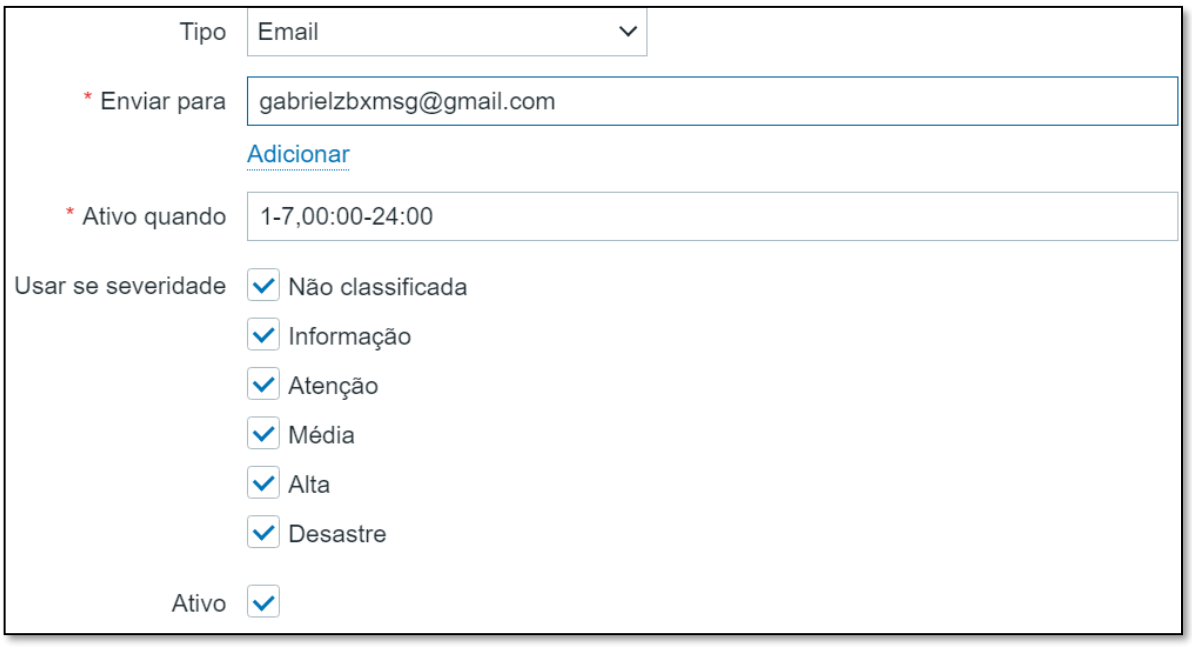


Figura C. 2 – Configurações da mídia no usuário

Passo 4 – Acessar “Trigger actions” em “Configuração -> Ações” e configurar uma nova ação para uma ou mais condições. A Figura C. 3 demonstra a condição imposta para ativar a ação, enquanto a Figura C. 4 ilustra a operação que será realizada para quando a condição for satisfeita.

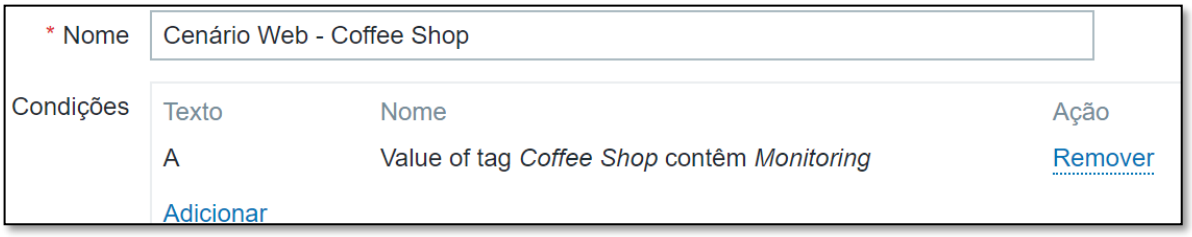


Figura C. 3 Exemplo de condições para ativação do gatilho

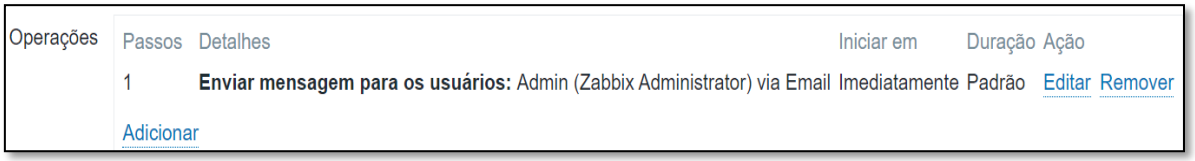


Figura C. 4 – Configurações da operação

ANEXO D – *SCRIPT* REMOTO

Passo 1 – Acessar "*Scripts*" em "Administração" no *frontend* do Zabbix e criar um *script*.

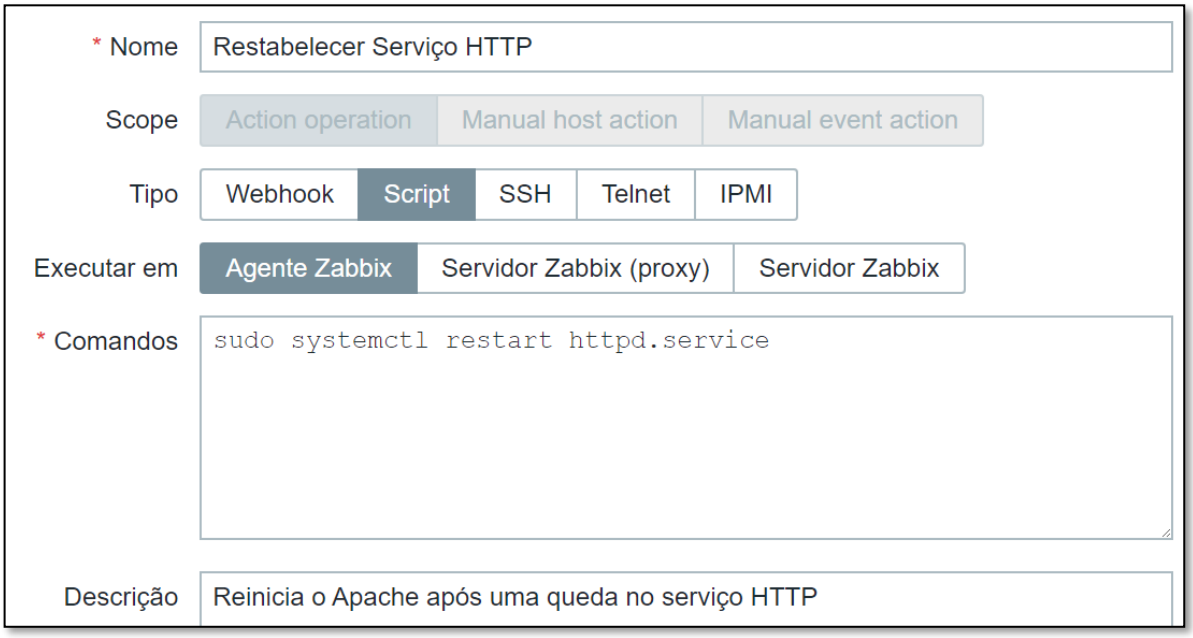


Figura D. 1 – Configurações do *script*

Passo 2 – Realizar o Passo 4 do Anexo C adaptando para este caso. As imagens Figura D. 2 e Figura D. 3 ilustram o procedimento.

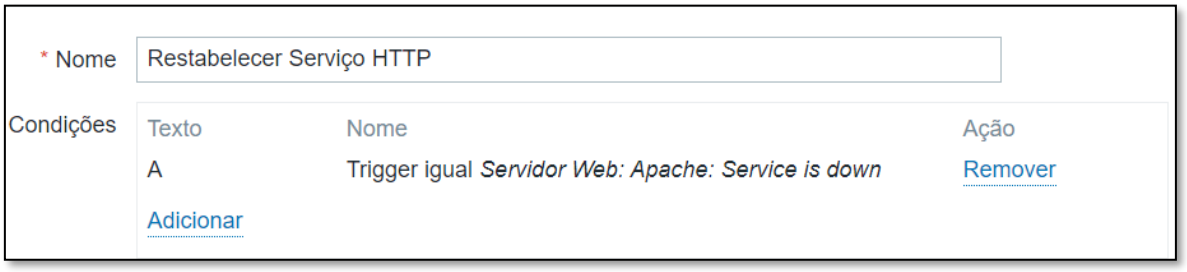


Figura D. 2 – Condições para execução do *script*

Figura D. 3 – Configurações da operação

Passo 3 – Por fim, acessar a máquina na qual o *script* será rodado

```
visudo
```

Adicionar a linha de texto:

```
zabbix ALL=NOPASSWD: ALL
```

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR
Av. Universitária, 1069 • Setor Universitário
Caixa Postal 88 • CEP 74605-010
Goiânia • Goiás • Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br • reitoria@pucgoias.edu.br

PUC GOIÁS

RESOLUÇÃO nº 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Gabriel Cardoso Barbosa, do Curso de Engenharia de Computação, matrícula 2017100330296 7, telefone: 62 99701-7995 e-mail gabriel.cardoso.barbosa@hotmail.com na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Gerenciamento de redes com o Zabbix com estudo de caso na disponibilidade de um serviço web, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 21 de junho de 2022.

Assinatura do autor: Gabriel Cardoso Barbosa

Nome completo do autor: Gabriel Cardoso Barbosa

Assinatura do professor–orientador:

Nome completo do professor-orientador: Angélica da Silva Nunes