PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO

GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

GERENCIAMENTO DE REDES COM ZABBIX

PAULO HENRYCK MARTINS SILVA

GOIÂNIA

2021

PAULO HENRYCK MARTINS SILVA

GERENCIAMENTO DE REDES COM ZABBIX

Trabalho de Conclusão de Curso apresentado por Paulo Henryck Martins Silva à Pontifícia Universidade Católica de Goiás como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação.

Orientadora: Profª Ms. Angélica da Silva Nunes.

GOIÂNIA

2021

PAULO HENRYCK MARTINS SILVA

GERENCIAMENTO DE REDES COM ZABBIX

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia da Computação em: ____/____/____.

_____

Profª. Ms Ludmilla Reis Pinheiro dos Santos.
Coordenadora de Trabalho de Conclusão de Curso

_____

Orientadora: Profª. Ms. Angélica da Silva Nunes

Banca Examinadora:

_____

Prof°. Ms. Rafael Leal Martins

_____

Profª. Dr. Solange da Silva

GOIÂNIA

2021

“Os dois dias mais importantes da sua vida são: O dia em que você nasceu, e o dia em que você descobre o porquê.”

Mark Twain

RESUMO

Este trabalho possui a proposta de estudar uma ferramenta de Gerenciamento de redes, que auxilie o administrador a obter informações fundamentais para manter o perfeito funcionamento da rede, deste modo minimiza a falta de controle sob o ambiente, melhorando a detecção de problema e maximizando o funcionamento da rede. O Zabbix foi a ferramenta escolhida para auxiliar a administração da rede, pois dispõe de diversos recursos, como coleta de dados em tempo real, alertas de erros ocorridos nos dispositivos gerenciados, relatórios, integração com outras ferramentas etc. Deste modo o que é visto neste trabalho é a implementação e configuração da ferramenta usada para o Gerenciamento de redes, bem como sua integração com outras ferramentas.

Palavra-Chave: Zabbix, Gerenciamento de redes, Proxy.

ABSTRACT

This work has the proposal of studying a network management tool, which helps the administrator to obtain fundamental information to maintain the perfect functioning of the network, thus minimizing the lack of control over the environment, improving problem detection and maximizing the functioning from the Web. Zabbix was the tool chosen to assist the administration of the network, as it has several resources, such as data collection in real time, alerts of errors occurred in the managed devices, reports, integration with other tools, etc. Thus, what will be seen in this work is the implementation and configuration of the tool used for network management, as well as its integration with other tools.

Keyword: Zabbix, Network management, Proxy.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AWS - Amazon *Web Services*

CPU - Unidade Central de Processamento - *Central Processing Unit*

DNS - Sistema de Nomes de Domínios - *Domain Name System*

GB - Giga*byte*

GHz - Giga*Hertz*

HTTP - Protocolo de Transferência de Hipertexto - *Hypertext Transfer Protocol*

HTTPS - Protocolo de Transferência de Hipertexto Seguro - *Hypertext Transfer Protocol Secure*

IP - Protocolo de Internet - *Internet Protocol*

IP/DNS - Protocolo de Internet/Sistema de Nomes de Domínios - *Internet Protocol/Domain Name System*

ISO - Organização Internacional para Padronização - *International Organization for Standardization*

JSON RPC - Notação de Objetos JavaScript e Chamada Remota de Procedimento - *JavaScript Object Notation Remote Procedure Call*

LAN – Rede de área local – *Local Area Network*

MB - Mega*byte*

MIB - Base de Informações de Gerenciamento - *Management Information Base*

NOC - Estação Central de Gerenciamento - *Network Operations Center*

OID - Identificado de Objeto - *Object Identifier*

PDU - Unidade de Dados de Protocolo - *Protocol Data Units*

PHP – *Hypertext Preprocessor*

RAM - Memória de Acesso Randômico - *Random Access Memory*

SGBD - Sistema de Gerenciamento de Banco de Dados

SLAS – Acordo de Nível de Serviço - *Service Level Agreement*

SNMP- Protocolo Simples de Gerenciamento de redes - *Simple Network Management Protocol*

TCP/IP - Protocolo de Controle de Transmissão/Protocolo de Internet – *Transmission Control Protocol/Internet Protocol*

UDP - Protocolo de datagrama do usuário - *User Datagram Protocol*

URL – Localizador Uniforme de Recursos - *Uniform Resource Locator*

USP – Universidade de São Paulo

VM - Máquinas Virtuais – *Virtual Machine*

WAN - Rede de longa distância - *Wide Area Network*

Sumário

1 INTRODUÇÃO

Nos princípios das redes de computadores, quando ainda eram produto de pesquisa, em vez de infraestrutura utilizada por centenas ou milhares de pessoas todos os dias, o termo "Gerenciamento de redes" era algo nunca ouvido antes. No caso de ocorrência de um problema na rede, era realizado alguns testes como *ping*, para desta forma encontrar a origem do problema. Com crescimento da *Internet* pública e da *intranet* privada e a transição de pequenas redes para grandes infraestruturas globais, mais funções precisam ser gerenciadas sistemicamente, o número crescente de componentes de *hardware* e *software* também se tornaram mais importantes (KUROSE e ROSS, 2013).

> 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, apud KUROSE, 2013, p. 558).

A infraestrutura tornou-se indispensável, assim sendo é uma tarefa crítica, ou seja, não pode ter seu serviço interrompido. Ter as ferramentas de Gerenciamento de rede adequadas, mesmo em um cenário de rede simples, é fundamental para o administrador, no qual se beneficia ao descobrir de forma rápida os problemas que venham a surgir.

Há muitas ferramentas de Gerenciamento de rede disponíveis no mercado, tanto *hardware* pago como livre. O Zabbix é uma ferramenta desenvolvida para profissionais envolvidos em Gerenciamento de redes, é um *software* livre código aberto, gratuito, é bastante utilizado no mercado europeu e brasileiro. Oferece bastante flexibilidade, e não se limita apenas a itens de monitoramento nativo, é possível incrementar *scripts*, itens personalizados e realizar integração com outros *softwares*, assim torna-se mais completo e profissional.

O Zabbix foi criado em 1998 por Alexei Vladishev, quando trabalhava como administrador de redes em uma instituição bancária na Letônia, pois não era satisfeito com os sistemas de monitoramentos disponíveis na época. Em 2001 foi lançado a primeira licença do Zabbix sob versão 0.1 *alpha.* Já em 2004 foi lançado a sua versão estável, a 1.0. Em 2005, vendo a necessidade de tratar a ferramenta de forma mais profissional, foi instituída a criação

da empresa Zabbix SIA. A partir de 2006, foi evoluindo até a versão conhecida hoje, alcançando 800 mil *downloads* no ano de 2012 (LIMA, 2014).

A plataforma Zabbix é amplamente divulgada, segundo a comunidade Zabbix Brasil, existem diversos casos de sucesso na utilização da ferramenta no mercado brasileiro por empresas, organizações e universidades como Banco Central, CAIXA, DATASUS de alguns hospitais, Universidade de São Paulo (USP), entre outras.

O Zabbix possuí três componentes principais em sua arquitetura Zabbix *Server*, Zabbix *Proxy* e Zabbix *Agent*. O Zabbix *Server* é o componente principal da solução, que gerência a coleta e recebimentos das informações, calcula a partir de expressões lógicas o estado das *triggers*, e envia notificações aos usuários. Os agente e *proxy* da rede, enviam a ele os dados sobre *performance*, disponibilidade e integridade dos serviços e recursos monitorados (ZABBIX, 2021).

O Zabbix P*roxy* possibilita monitorar ambientes remotos, onde há a necessidade de se ter diversos dispositivos monitorados. Se esse monitoramento for realizado diretamente pelo servidor pode ocorrer percas de informações, até mesmo sobrecarregas no servidor. Deste modo, em um ambiente de rede com *proxy* a coleta de dados é realizada de forma distribuída, salvando localmente as informações, que posteriormente são enviados ao servidor, assim diminuindo o fluxo e perca de informações.

O Zabbix *Agent* é responsável por monitorar ativamente os recursos e aplicações locais em um dispositivo, como por exemplo utilização de Memória de Acesso Randômico - Random Access Memory (RAM), disco rígido, estatísticas de uso do processador, *interface* de rede, entre outras, é compatível com as principais plataformas disponíveis no mercado.

Justifica-se a pesquisar este assunto, a busca por explorar os benefícios da utilização de uma ferramenta para Gerenciamento da rede, bem como sua integração com outras ferramentas, para uma melhor administração do ambiente de rede, auxiliando na identificação e correção problemas de maneira ágil.

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

- Quais os benefícios o Zabbix pode trazer para o Gerenciamento de redes?

1.1 Objetivo Geral

Utilizar o Zabbix com objetivo de ter controle de atividade e monitoramento do uso dos recursos do ambiente de rede, resultando em maior eficiência e produtividade.

1.2 Objetivo Específicos

Implementar a ferramenta Zabbix, tendo como foco atingir os seguintes objetivos:

- Monitorar a *performance* e disponibilidade de redes, aplicações e recursos;
- Coletar e armazenar informações da infraestrutura, com objetivo de ter histórico do sistema para possíveis análise;
- Obter relatórios para acompanhamento do ambiente;
- Enviar alertas em caso o sistema não esteja de acordo com os parâmetros estabelecidos.

1.3 Metodologia

Esta pesquisa quanto a sua natureza é um resumo de assunto, pois irá reunir, analisar e discutir conhecimentos e informações já publicadas, desta forma o projeto demonstra a aplicação de métodos científicos.

Quanto aos objetivos é uma pesquisa explicativa. Segundo Wazilawick (2014), esta pesquisa é mais completa por analisar os dados observados, busca suas causas e explicações, ou seja, os fatores determinantes desses dados. Desta forma, o projeto visa coletar dados e analisar, assim podendo ter conhecimento sobre o ambiente da infraestrutura.

Quanto aos procedimentos técnicos é uma pesquisa experimental, realizando a manipulação na quantidade e qualidade de uma ou mais variáveis no ambiente de rede, o qual permite o estudo da relação entre causas do evento, podendo assim controlar e avaliar.

1.4 Estrutura da monografia

A estrutura deste trabalho é a seguinte:

No capítulo 2 é apresentado a fundamentação teórica sobre o Gerenciamento de redes e seus principais elementos.

No capítulo 3 é definido o que é o Protocolo Simples de gerenciamento de redes - *Simple Network Management Protocol* (SNMP), seus dois principais modos de operações, as diferenças e modo de configurações das versões, como é o funcionamento da base de informação de Gerenciamento e o método utilizado para troca de mensagens no protocolo SNMP.

No capítulo 4 são descritas as ferramentas utilizadas para o Gerenciamento, será apresentado o Zabbix, pfSense e Grafana.

No capítulo 5 descreve o ambiente utilizado para implantação do servidor Zabbix, página *web* utilizada para acessar os serviços e o Sistema de Nomes de Domínios - *Domain Name System* (DNS).

No capítulo 6 aborda a implantação do ambiente de Gerenciamento de rede em um sistema de colete de dados em tempo real.

No capítulo 7 apresenta as considerações finais e os problemas encontradas durante a implementação do ambiente.

2 GERENCIAMENTO DE REDES

2.1 Gerenciamento de redes

O Gerenciamento de rede inclui a integração, instalação e coordenação dos elementos de *hardware* e *software*. Possui como objetivo monitorar, testar, inspecionar, avaliar, configurar, analisar, controlar a rede e os recursos desses elementos para atingir requisitos em tempo real, desempenho operacional e qualidade de serviços a um custo razoável. (KUROSE, 2013).

As ferramentas de Gerenciamento de redes são recursos necessários para realizar o monitoramento, controle e gerenciar os recursos da rede. Há muitos cenários e o administrador de redes se beneficia da utilização de ferramentas de gerência de redes, como:

- Detecção de falha: uma *interface* de algum roteador que esteja apresentando falhas, deste modo pode tomar decisões de substituição ou reparo de modo mais rapidamente;
- O monitoramento de hospedeiros: trabalhar de forma preventiva, identificando o problema antes que ele ocorra;
- Monitoramento de tráfego: permite que os recursos e serviços possam ser corretamente fornecidos e distribuídos a todos os *hosts* dentro das regras definidas para a rede;
- Detecção de mudanças rápidas em tabela de roteamento: mudanças frequentes na tabela de roteamento podem indicar roteamento instável ou configuração incorreta no roteador;
- Monitorar Acordo de Nível de Serviço - *Service Level Agreement* (SLAs): é um contrato estabelecendo padrões de desempenho para a prestação do serviço, como latência, vazão, disponibilidade do serviço (interrupção do serviço) e requisitos de notificação para interrupção do serviço;
- Detectar e monitorar invasão em sua rede: havendo comportamento anormal na rede monitorada, será realizado alerta ao administrador.

2.2 Modelo de Gerenciamento da ISO

Segundo Kurose e Ross, a Organização Internacional para Padronização - *International Organization for Standardization* (ISO) define cinco áreas de Gerenciamento de rede:

- Gerenciamento de Desempenho: tem como objetivo é medir, analisar, informar e controlar o desempenho da rede, permitindo o controle da qualidade do serviço;
- Gerenciamento de Falhas: sua meta é detectar, registrar e reagir às condições de falha da rede;
- Gerenciamento de Configuração: permite que o administrador saiba quais dispositivos fazem parte da rede e suas configurações de *hardware* e *software*;
- Gerenciamento de Contabilização: permite que o administrador especifique, registre e controle o acesso de usuários e dispositivos aos recursos da rede, determinando assim os custos associados ao seu uso;
- Gerenciamento de Segurança: seu propósito é realizar o controle aos recursos da rede de acordo com a política de segurança definida.

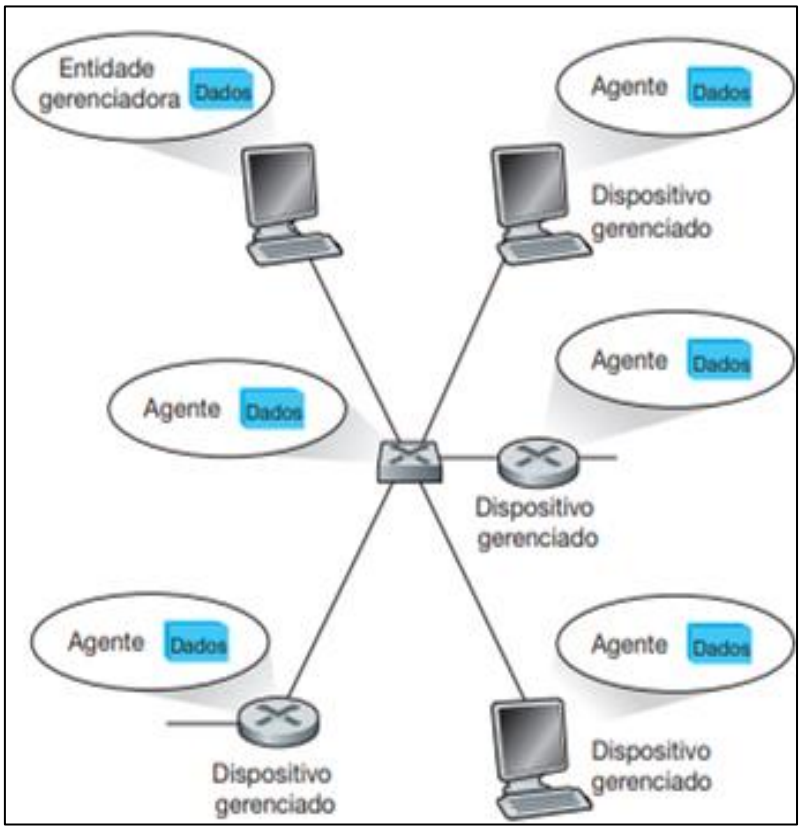
2.3 Arquitetura de Gerenciamento

A Figura 1 mostra os principais componentes da arquitetura de gerenciamento de rede, a entidade gerenciadora é o programa executado em uma Estação Central de Gerenciamento - *Network Operations Center* (NOC), responsável pelo controle e coleta dos dados gerenciados.

O dispositivo gerenciado são equipamento localizados na rede gerenciada como roteador, *hub*, impressora, modem e assim por diante incluindo *hardware*. No interior pode haver objetos gerenciados como elementos de *hardware* e *hardware*, como por exemplo *interface* de rede, processador, memória, disco rígido ou tarefas executadas. As informações relacionadas aos objetos gerenciados são coletadas e armazenados dentro da Base de Informações de Gerenciamento - *Management Information Base* (MIB). Também existe um agente de Gerenciamento de rede, responsável por realizar a comunicação entre a entidade gerenciadora e o dispositivo gerenciado, e executando ações enviadas pela entidade gerenciadora.

O protocolo de gerenciamento de rede é responsável por toda troca de informações entre a entidade gerenciadora e o agente de gerenciamento de rede dos dispositivos gerenciados, portanto ele é uma ferramenta e um canal de comunicação, desta forma permite monitorar, testar, consultar, configurar, analisar e controlar a rede.

Figura 1 - Principais componentes de gerenciamento de rede.



Fonte: KUROSE, 2013

3 PROTOCOLO SNMP

3.1 O que é o SNMP?

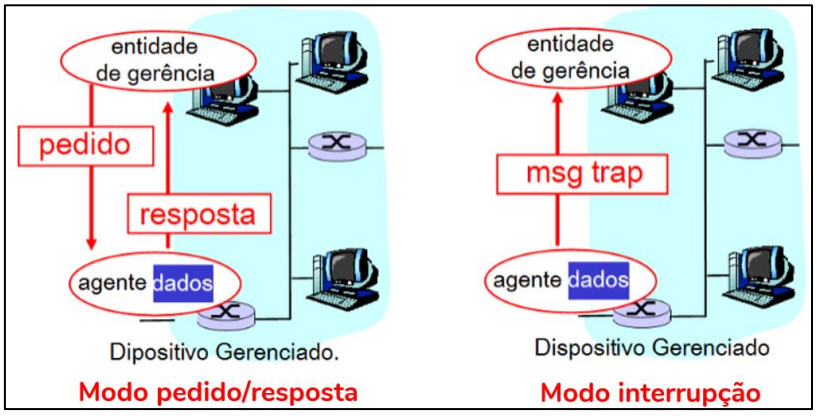
Para atender o aumento da demanda por um padrão para gerenciar dispositivos de Protocolo de *Internet* - *Internet Protocol* (IP), foi lançado em 1988 o SNMP, com o objetivo de oferecer um conjunto simples de operações que permite ter acesso remoto aos dispositivos gerênciados (MAURO; SCHMIDT, 2001).

O SNMP é o padrão de protocolo de gerenciamento de rede mais popular, padrão aberto adotado por diferentes fabricantes e operadoras, no qual possibilita ao administrador de rede obter informações de dispositivos ou componentes, como utilização de *interface* de rede, até mesmo realizar modificações, reiniciar ou desabilitar. Pode se gerenciar qualquer dispositivo com protocolo SNMP sendo ele *hardware* ou até mesmo *software*.

3.2 Modos de operações do SNMP

A Figura 2 apresenta dois modos de operações do protocolo SNMP, o primeiro é o modo pedido/resposta, em que a entidade de gerência envia um pedido para o agente de um dispositivo gerenciado, no qual é realizado uma ação, e envia a resposta do pedido a entidade de gerência com essa resposta pode ser alguma informação desejada, mensagem de confirmação ou erro. O segundo modo de operação é o modo interrupção, em que o agente do dispositivo gerenciado envia uma mensagem *trap* que é usada para notificar a entidade gerenciadora de situação eventual que causou alteração nos valores dos objetos MIB.

Figura 2 - Modos de operações do SNMP



Fonte: KUROSE, 2013

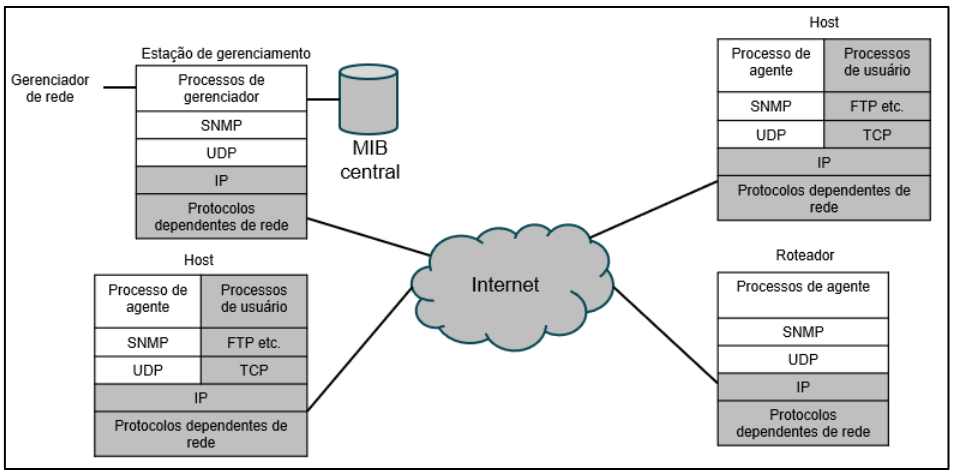
3.3 Versões do SNMP

O SNMP foi criado para servir como ferramenta de gerenciamento de redes, operando em nível de aplicação da arquitetura Protocolo de Controle de Transmissão/Protocolo de Internet – *Transmission Control Protocol/Internet Protocol* (TCP/IP), utilizando como transporte o protocolo Protocolo de Datagrama do Usuário - *User Datagram Protocol* (UDP). Desde então, vem sendo aprimorado expandindo para uso em diversos tipos de ambientes de rede. Existem três versões do SNMP e ao logo deste tópico será abordado a particularidades de cada um e suas diferenças.

3.4 Diferença entre SNMPv1 e SNMPv2

A Figura 3 ilustra uma configuração do SNMPv1 em que a uma estação de Gerenciamento independente, um processo de gerenciamento controla o acesso a MIB central e fornece uma *interface* com gerenciador de rede. Ainda na figura pode-se ver que o agente de cada dispositivo deve ter implementado o SNMP, UDP e IP. As partes sombreadas representam o ambiente operacional e as não sombreadas fornece suporte para função de gerenciamento de rede.

Figura 3 - Configuração do SNMPv1.



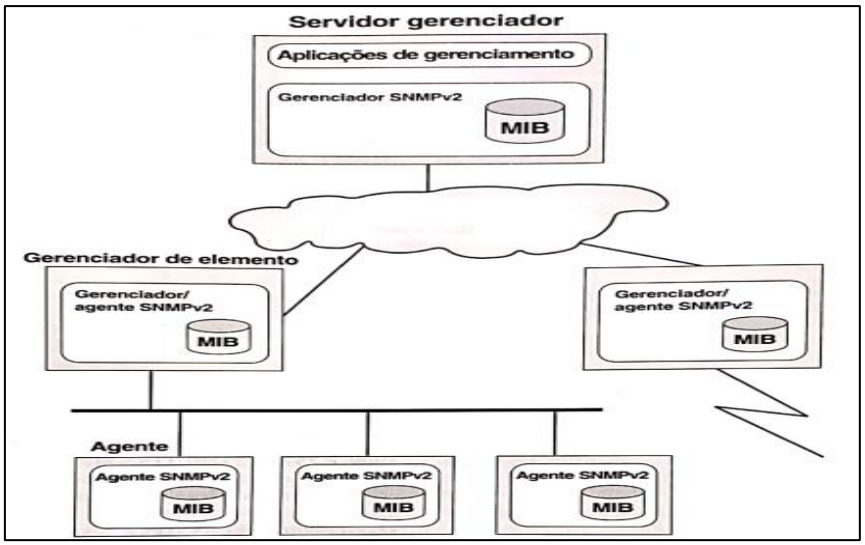
Fonte: STALLINGS, 2005

O SNMPv2 aceita uma estratégia de gerenciamento altamente centralizada ou uma estratégia distribuída, onde pode haver mais de uma estação de gerenciamento. Deste modo,

possibilita uma hierarquia de gerentes, assim fornece redundância ou simplesmente divide as responsabilidades em uma grande rede.

A Figura 4 ilustra a configuração do SNMPv2, que possui um servidor gerenciador, gerenciadores de elemento que podem assumir o papel de gerenciador e agente, e logo abaixo agentes. O servidor gerenciador deseje informações de um agente que é remoto a ele. Neste caso ele deve solicitar a informação ao gerenciador de elemento que realiza a função de agente *proxy*, que assume o papel de gerenciador para acessar informações no agente remoto ao servidor gerenciador, e depois realizar a solicitação assume novamente papel de agente para passar as informações para o servidor gerenciador.

Figura 4 - Configuração SNMPv2



Fonte: STALLINGS, 2005

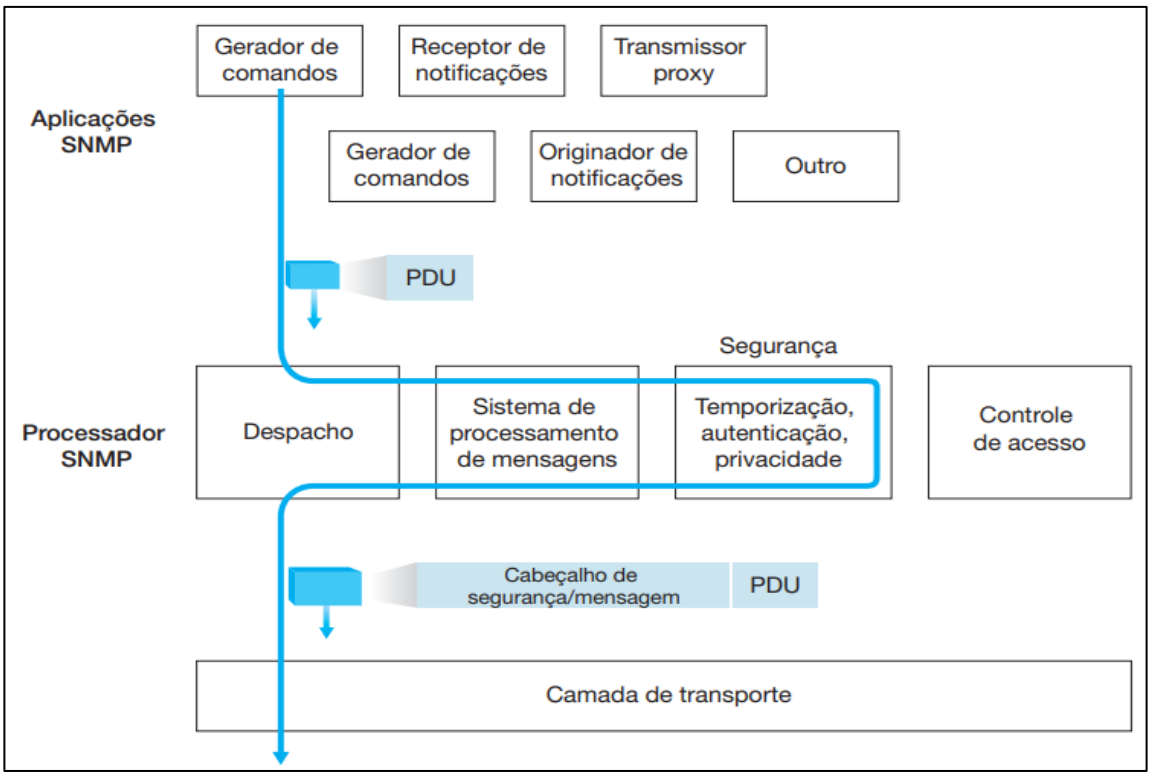
3.5 SNMPv3 segurança

Para resolver problemas de segurança do SNMPv1 e SNMPv2, foi lançado em 1998 o SNMPv3, considerado uma versão aprimorada do SNMPv2 com melhorias de segurança e capacidade de administração. Seu modelo de segurança é mais robusto, diferente das versões anteriores que utilizavam esquema baseado em comunidades que são *string* de texto puro, todas as mensagens do SNMPv3 possuem parâmetros de segurança, os quais são chaves que possuem o objetivo de garantir autenticação e a privacidade na comunicação entre agentes e gerentes.

O SNMPv3 fornece três serviços importantes, o primeiro é a autenticação que garante que a mensagem não foi alterada em trânsito, atrasada ou repetida. O segundo é a privacidade que permite que gerenciadores e agentes criptografem mensagens. O terceiro é o controle de acessos que permite configurar agentes para fornecer níveis diferentes de acesso a MIB do agente para diferentes gerenciadores.

A Figura 5 ilustra a segurança e administração SNMPv3 em que uma Unidade de Dados de Protocolo - *Protocol Data Units (PDU)* é gerada pela aplicação gerador de comandos, que entra no módulo de despacho e determina a versão do SNMP. Logo adiante, a PDU passa pelo sistema de processamento de mensagens, onde é incluído cabeçalho de mensagem, identificação única e informações sobre tamanho. Posteriormente é enviado para módulo de segurança e controle de acesso que criptografa e autêntica a mensagem. Após finalizar todos os passos a PDU é encaminhada para a camada de transporte, e é enviada para o destinatário.

Figura 5 - Segurança e administração SNMPv3



Fonte: KUROSE, 2013

3.6 Base de Informações de Gerenciamento MIB

A MIB é um banco de dados que armazena objetos gerenciados no qual os valores, coletivamente, representam o estado atual de dispositivos conectados à rede, essas informações são utilizadas no gerenciamento da rede e podem ser solicitadas pela entidade gerenciadora.

A estrutura básica de objetos da MIB é ilustrada pela Figura 6, em que cada objeto possui um nome, tipo, valor, forma de acesso *status* e descrição do objeto.

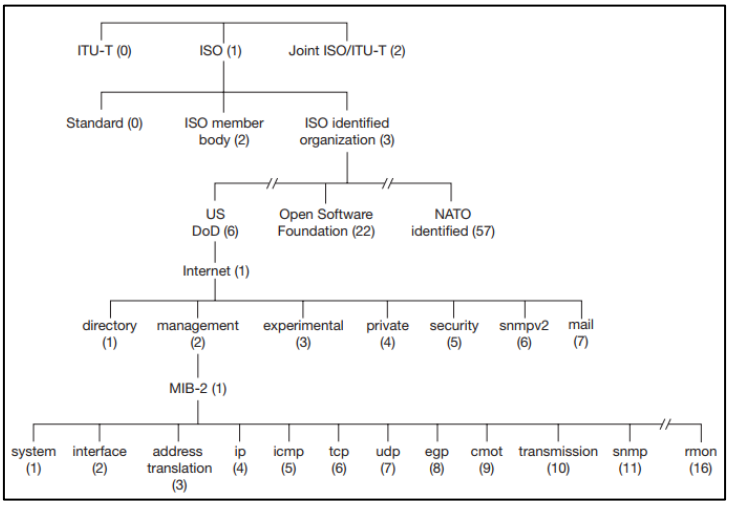
Figura 6 - Estrutura básica de objetos da MIB

```
<name> OBJECT-TYPE
        SYNTAX <datatype>
        ACCESS <read-only, read-write, write-only ou not-accessible>
        STATUS <mandatory, optional ou obsolete>
        DESCRIPTION
                "Descrição em texto explicando o objeto."
        ::= { <OID exclusiva que define este objeto> }
```

Fonte: Elaborado pelo autor

A arquitetura da MIB é organizada em forma de árvore, definida na ISO ASN.1. Deste modo, organiza todas as informações, como ilustrado na Figura 7, cada parte da árvore é um nó que contém um Identificador de Objeto - *Object Identifier* (OID) sequência de números separados por pontos, e uma breve descrição textual com a descrição do nó rotulado.

Figura 7 - Árvore hierárquica definida pela ISO ASN.1



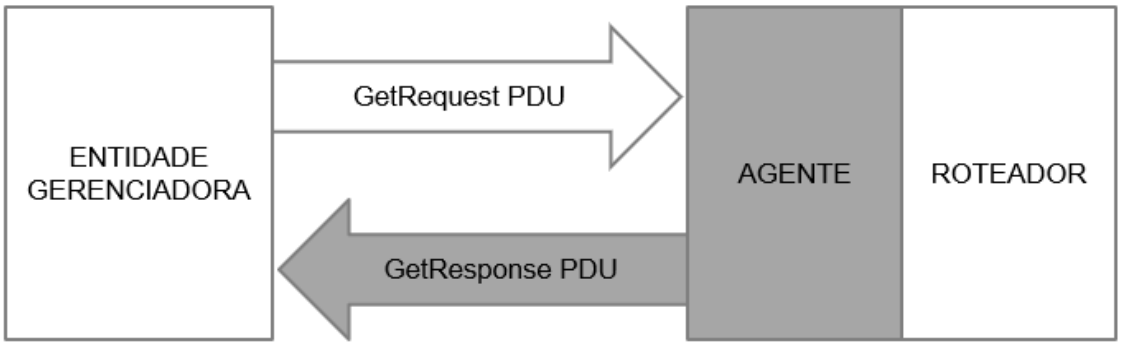
Fonte: STALLINGS, 2005

3.7 Tipos de PDUs

Os gerenciadores e agentes utilizam o formato de mensagem PDU para enviar e receber informações, são definidos sete tipos, descritos a seguir:

O *GetRequest* é utilizado para solicitar um único objeto por vez na MIB, como demonstrado na Figura 8 é uma solicitação entre gerente e agente, onde a entidade gerenciadora envia uma requisição *GetRequest* PDU solicitando um objeto da MIB, o agente recebe a requisição realiza a ação e envia o objeto solicitado pela requisição *GetResponse*.

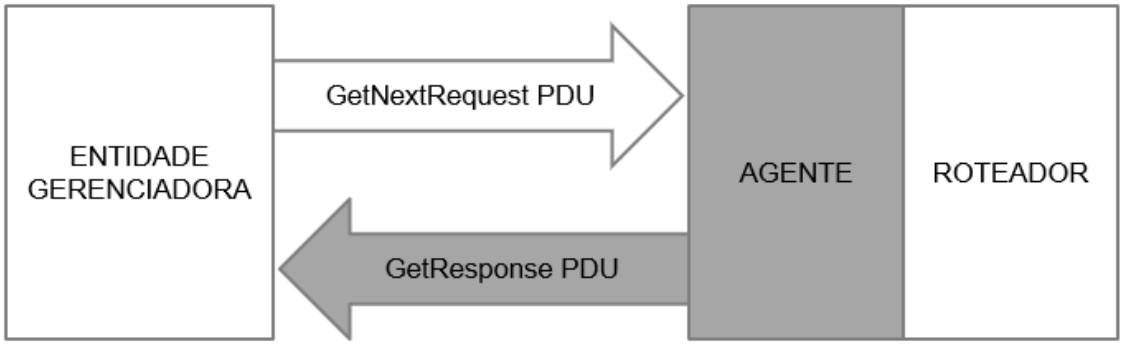
Figura 8 - Modo de operação PDU *GetRequest*



Fonte: Autoria própria, adaptado de MAURO, 2001

A operação *GetNextRequest* permite recuperar um grupo de objetos da MIB, e realizado uma sequência de comandos, como ilustrado na Figura 9 a entidade gerenciadora envia o comando para o agente que realiza a ação e envia um *GetResponse* com os objetos solicitados, quando a entidade gerenciadora recebe a resposta do comando emitido recentemente, ela envia outro, assim sucessivamente até o agente retornar um erro, deste modo sinalizando que chegou ao final da MIB e não possui mais objetos a serem obtidos.

Figura 9 - Modo de operação PDU *GetNextRequest*



Fonte: Autoria própria, adaptado de MAURO, 2001

A Figura 10 apresenta o comando *GetNextRequest* PDU no sistema operacional UNIX, comando esse chamado *snmpwalk*, logo em seguida e colocado o nome do dispositivo a ser consultado, tipo de acesso e o grupo que se deseja obter os valores no caso o *system*, por fim agente atende a requisição e retorna os valores.

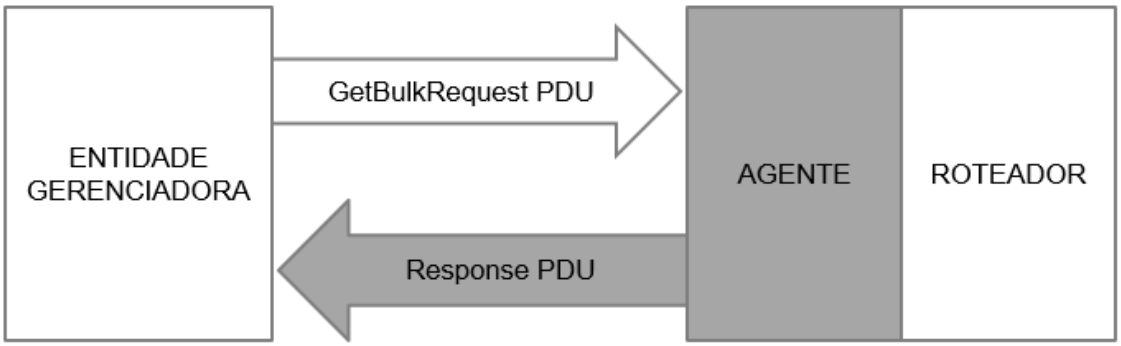
Figura 10 - Comando *GetNextRequest* no UNIX

```
$ snmpwalk cisco.ora.com public system
system.sysDescr.0 = "Cisco Internetwork Operating System Software
..IOS (tm) 2500 Software (C2500-I-L), Version 11.2(5), RELEASE
SOFTWARE (fc1)..Copyright (c) 1986-1997 by cisco Systems, Inc...
Compiled Mon 31-Mar-97 19:53 by ckralik"
system.sysObjectID.0 = OID: enterprises.9.1.19
system.sysUpTime.0 = Timeticks: (27210723) 3 days, 3:35:07.23
system.sysContact.0 = ""
system.sysName.0 = "cisco.ora.com"
system.sysLocation.0 = ""
system.sysServices.0 = 6
```

Fonte: MAURO, 2001

O *GetBulkRequest* é uma operação que permite recuperar um grande bloco de uma tabela, deste modo evita sobrecargas ocorridas pelo envio de múltiplas mensagens por operações como *GetNextRequest* e *GetRequest*. Na Figura 11 ilustra a entidade gerenciadora enviando o comando para o agente e após o agente realizar a ação e enviado um *GetResponse* com os valores solicitados.

Figura 11 - Modo de Operação *GetBulkRequest*



Fonte: Autoria própria, adaptado de MAURO, 2001

Para melhor entendimento da operação *GetBulkResquest*, a Figura 12 mostra um exemplo no sistema operacional UNIX, onde é digitado o comando *snmpbulkget*, como *GetBulkResquest* é um comando do SNMPv2 é colocado *-v2c* para indicar a versão do SNMP, logo após *-B 1 3* que defini o número máximo de repetições do comando, a diante e colocar o nome do dispositivo consultado e tipo de acesso, e por fim as tabelas que se deseja obter os valores.

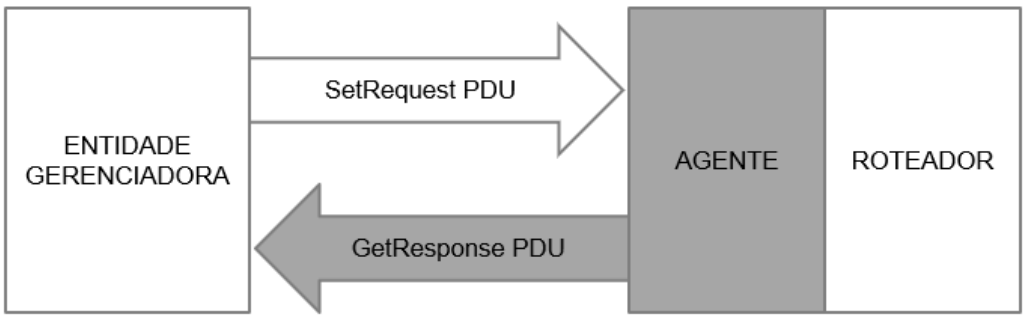
Figura 12 - Comando *GetBulkRequest* no UNIX

```
$ snmpbulkget -v2c -B 1 3 linux.ora.com public sysDescr ifInOctets
ifOutOctets
system.sysDescr.0 = "Linux linux 2.2.5-15 #3 Thu May 27 19:33:18 EDT 1999
i686"
interfaces.ifTable.ifEntry.ifInOctets.1 = 70840
interfaces.ifTable.ifEntry.ifOutOctets.1 = 70840
interfaces.ifTable.ifEntry.ifInOctets.2 = 143548020
interfaces.ifTable.ifEntry.ifOutOctets.2 = 111725152
interfaces.ifTable.ifEntry.ifInOctets.3 = 0
interfaces.ifTable.ifEntry.ifOutOctets.3 = 0
```

Fonte: MAURO, 2001

A operação *SetRequest* como ilustrado a Figura 13, a entidade gerenciadora envia o comando da operação para o agente de um dispositivo gerenciado que estabelece o valor de um ou mais objetos da MIB. Deste modo, a operação pode ser utilizada para modificar o valor de um objeto ou criar um novo.

Figura 13 - Modo de Operação *SetRequest*



Fonte: Autoria própria, adaptado de MAURO, 2001

A Figura 14 ilustra uma solicitação no sistema operacional UNIX utilizado o *SetRequest*, onde é digitado o comando *snmpset*, nome do dispositivo, tipo de acesso, variável a ser alterada, e o novo valor.

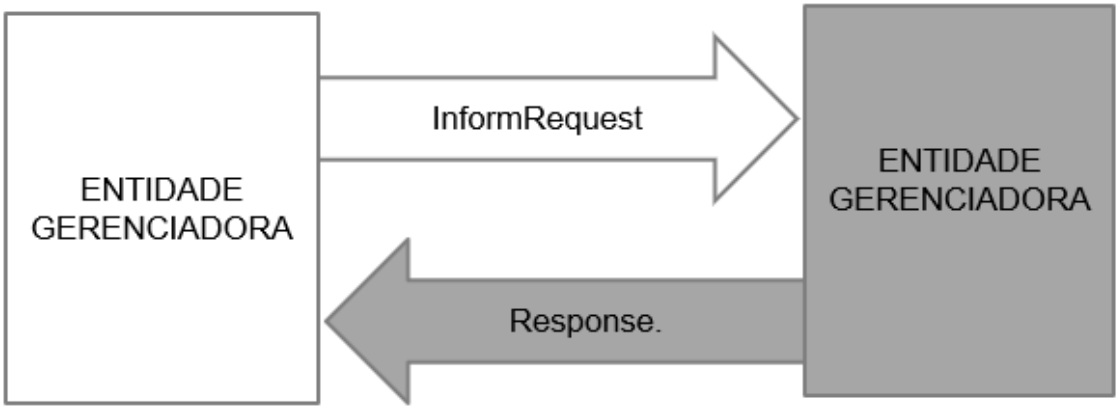
Figura 14 - Comando *GetBulkRequest* no UNIX

```
$ snmpset cisco.ora.com private system.sysLocation.0 s "Atlanta, GA"
system.sysLocation.0 = "Atlanta, GA"
```

Fonte: MAURO, 2001

O PDU *InformRequest* estabelece comunicação entre dois gerentes como apresenta na Figura 15, onde entidade gerenciadora envia o comando para outra entidade gerenciadora solicitando informação MIB remotas a entidade remetente, então a entidade receptora envia um *GetResponse* com as informações MIB.

Figura 15 - Modo de Operação *InformRequest*

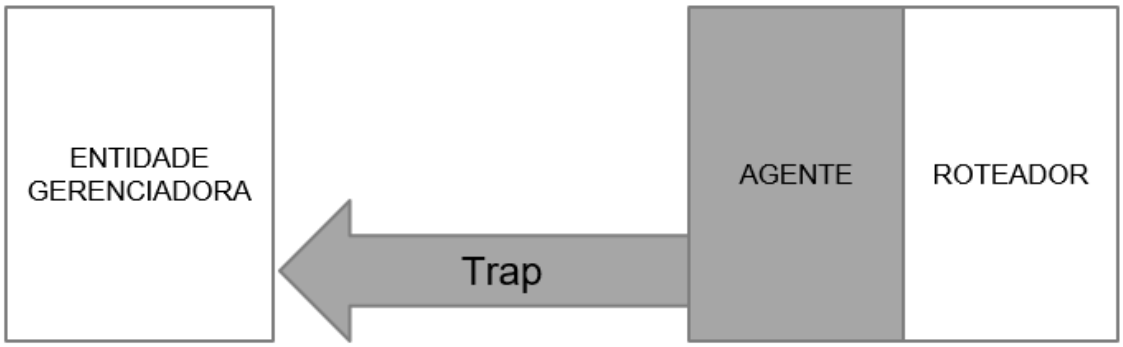


Fonte: Autoria própria, adaptado de MAURO, 2001

A operação *Response* ou *GetResponse* não é propriamente um comando, é uma mensagem contendo valores como reposta de uma requisição ou uma confirmação de uma mudança pela entidade gerenciadora.

Diferente dos demais operações a *Trap* é enviada pelo agente para entidade de gerência como ilustrado na Figura 16, o comando é enviado para notificar que uma situação eventual que causou alteração nos valores dos objetos MIB. Essa situação eventual pode ter sido ocasionada por uma *interface* de rede inoperante, perda de conexão com dispositivo, dentre os eventos que possam vir acontecer.

Figura 16 - Modo de Operação *Trap*



Fonte: Autoria própria, adaptado de MAURO, 2001

4 FERRAMENTAS UTILIZADAS

Este capítulo descreve as ferramentas utilizadas no gerenciamento da rede. As ferramentas têm o propósito facilitar a coleta de dados, gerar alertas, relatórios, proporcionar ao gerente de rede uma melhor visualização do ambiente e facilitar a identificação de problemas.

4.1 Zabbix

O Zabbix é um *software* de monitoramento distribuído de código aberto, criado por Alexei Vladishev em abril de 2001. A ferramenta possibilita monitorar vários parâmetros da rede, saúde e integridade dos servidores e agentes.

A ferramenta dispõe de um mecanismo de notificação que permite aos usuários configurar alertas onde pode ser enviado por mensageiros como Telegram, WhatsApp, Slack, *e-mail* e muitos outros para notificar sobre praticamente qualquer tipo de evento, deste modo permite uma reação rápida aos problemas. Oferece também a visualização de dados com base nos dados armazenados durante o monitoramento.

4.1.1 Pré-requisitos

Para instalação do Zabbix o *hardware* mínimo recomendado é 128 Mega*byte* (MB) de memória física e 256 MB de espaço em disco, no entanto, a quantidade de recurso necessária varia de acordo com a necessidade do ambiente ao qual a ferramenta está sendo implementada, o espaço em disco deve ser planejado acordo com a quantidade de *host* monitorados e o tempo de armazenamento dos dados coletados. O banco de dados pode exigir uma quantidade significativa de recursos de processamento e memória física dependendo da quantidade de parâmetros monitorados e do Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado.

A Tabela 1 fornece a configuração recomendada dependendo do ambiente onde a ferramentas está sendo implementada, indicando a plataforma, *hardware*, SGBD e quantidade de *host* monitorados sugeridos.

Tabela 1 - Pré-requisitos para instalação Zabbix

Nome	**Plataforma**	**CPU/Memória RAM**	**SGBD**	***Hosts* monitorados**
Pequeno	CentOS	Dispositivo Virtual	MySQL InnoDB	100
Médio	CentOs	2 CPU cores/2GB RAM	MySQL InnoDB	500
Grande	RedHat Enterprise Linux	4 CPU cores/8GB RAM	RAID10 MySQL InnoDB ou PostgreSQL	>1000
Muito grande	RedHat Enterprise Linux	8 CPU cores/16GB RAM	RAID10 rápido MySQL InnoDB ou PostgreSQL	>10000

Fonte: Autoria própria, adaptado de Zabbix SIA, s.d.

4.1.2 Nomeclaturas

O Zabbix possui alguns termos utilizados para definir os componentes que fazem parte de sua composição:

- *Host*: dispositivo de rede monitorado através de Protocolo de Internet/Sistema de Nomes de Domínios - *Internet Protocol/Domain Name System* (IP/DNS);
- Grupo de *hosts*: agrupamento lógico de *hosts*, utilizado para definir regras de acesso a diferentes grupos de usuários;
- Evento: ocorrência de alguma eventualidade que merece atenção do usuário, como por exemplo a mudança de estado em uma *trigger* ou *host*;
- Ação: reação pré-definida de um evento;
- Escalonamento: cenário personalizado para execução de operação de ações, como uma sequência de envio de comandos remotos e notificações;

- Notificação: mensagem enviada através da mídia, sobre um evento ocorrido;
- *Template*: grupamento de itens, *triggers*, regras de descobertas, gráficos, telas, cenários *web*, prontos para utilização a um *host* ou a um conjunto de *hosts*;
- Aplicação: união de itens em um agrupamento lógico;
- Cenário *web*: requisições Protocolo de Transferência de Hipertexto - *Hypertext Transfer Protocol* (HTTP) ou Protocolo de Transferência de Hipertexto Seguro - *Hypertext Transfer Protocol Secure* (HTTPS), utilizadas para monitoramento de páginas *web*.
- *Front-End*: *Interface web* utilizada na configuração e acesso a *interface* de monitoramento;
- API-Zabbix: permite a partir do protocolo Notação de Objetos JavaScript e Chamada Remota de Procedimento - *JavaScript Object Notation Remote Procedure Call* (JSON RPC) a atualização, criação e busca de objetos Zabbix, como itens, gráficos, *hosts* e outros, ou realizar qualquer outra tarefa personalizada.

4.1.3 Coleta de dados

A coleta de dados dos dispositivos conectado a rede torna possível o Gerenciamento, o Zabbix dispõe diversas formas de monitoramento e coleta de dados, como:

- Agente Zabbix: é uma aplicação instalada no dispositivo a ser gerenciado, que possui capacidade de realizar o monitoramento dos recursos e aplicações como, disco, memória RAM, processador, *interface* de rede entre outras, em uma eventualidade onde dispositivo gerenciado apresente alguma falha em seus recursos, é emito um alerta. O Agente Zabbix concentra as informações local do dispositivo gerenciado monitorado para imediato envio ao servidor de Gerenciamento ou dependendo da configuração ao Zabbix *Proxy*. A aplicação é compatível com diversos sistemas operacionais como, HP-UX, FreeBS, Mac OS X, Solaris, Android, Windows, IBM AIX, NetBSD e OpenBSD;

- SNMP: o Zabbix possui suporte ao protocolo SNMP, onde possibilita monitorar e coletar dados dos dispositivos de rede que tenham suporte ao protocolo em seu *firmware*.

4.1.4 Tipos de Agente Zabbix

O Agente Zabbix é capaz de realizar verificações de forma passiva ou ativa, na verificação passiva o agente responde a uma requisição de informações, quando o servidor ou o *Proxy* Zabbix realiza a requisição dos dados em um momento quando for necessário e o agente responde a requisição solicitada.

Na verificação ativa o agente recebe primeiro uma lista de itens a ser monitorados e o intervalo entre as coletas, deste modo, permite o monitoramento mesmo que o servidor Zabbix ou *Proxy* Zabbix esteja indisponível, enviando os dados posteriormente quando o mesmo estiver disponível novamente.

4.1.5 Item

Os itens são utilizados para receber dados de um *host*, ao realizar a configuração de um *host*, é necessário adicionar itens para iniciar a coletas dos dados. Cada item possui métrica individual, é possível adicionar diversos itens de uma só vez realizando a criação de um *template* com os itens pré-definidos e realizar associação *template* ao *host* desejado.

4.1.6 Mídia

O Zabbix disponibiliza serviço de envio de mensagens contendo incidentes a partir de mídias criadas e devidamente configuradas. Desta forma, no momento em que ocorre o disparo de uma *trigger* ou um simples alerta de mudança no *host*, a atualização de pacotes disponíveis, entre outras, é enviada uma mensagem ao usuário o notificando, desta maneira mesmo não estando observando os painéis do NOC o ele é notificado do incidente.

4.1.7 Trigger

As *triggers* são constituídas por expressões lógicas, responsáveis por analisar os dados coletados pelos itens e representar o estado do sistema relacionado a eles. É possível delimitar um limite aceitável de dados, deste modo quando o dado obtido estiver fora dos limites delimitado a *trigger* é acionada, mudando estado para “INCIDENTE”. Cada vez que o servidor Zabbix recebe um novo valor que afeta a expressão, é recalculado o estado (expressão lógica) da *trigger*.

A *trigger* pode ter os seguintes estados:

Tabela 2 - Estados atribuídos a *trigger*

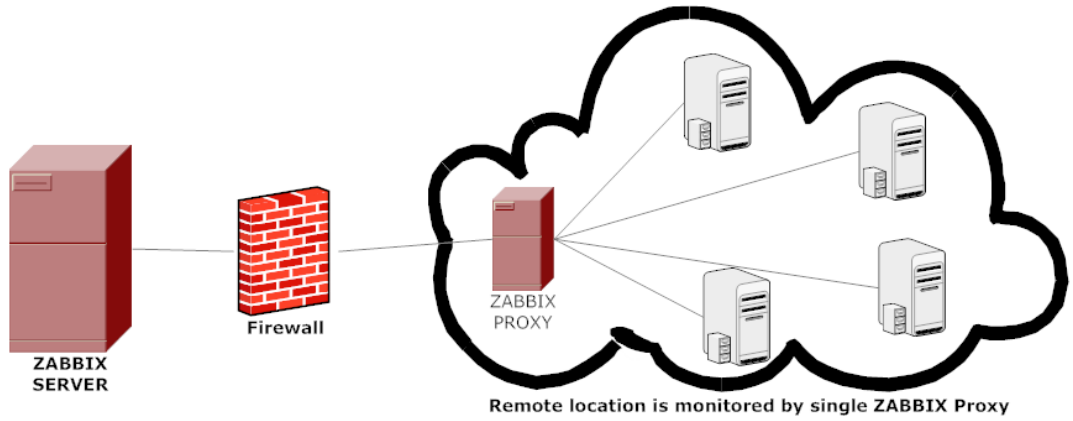
Valor	**Descrição**
OK	Este é o estado normal do gatilho. Nas versões anteriores do Zabbix, era chamado de "*FALSE* / FALSO".
INCIDENTE	Geralmente indica que algo aconteceu. Por exemplo, "CPU *Load*" é muito alto. Nas versões anteriores do Zabbix, era chamado de "*TRUE* / VERDADEIRO".

Fonte: Autoria própria, adaptado de Zabbix SIA, s.d.

4.1.8 Zabbix Proxy

O Zabbix *Proxy* é um processo que consegue receber dados coletados de um ou mais dispositivos gerenciados monitorados e posteriormente enviar ao servidor Zabbix. A Figura 17 demonstra o esquema de funcionamento do Zabbix *Proxy*, no qual funciona em nome do servidor em que, aos olhos do agente monitorado o *proxy* se torna o servidor. Todos os dados recebidos são armazenados em *buffer* temporariamente e transmitidos ao servidor ao qual o *proxy* pertence, e serão excluídos em seguida do *buffer*.

Figura 17 - Topologia de rede utilizando *proxy*



Fonte: Zabbix SIA, s.d.

O *proxy* é uma solução ideal para monitoramento centralizado geograficamente dispersos e para realização de gerenciamento remotos. O *proxy* necessita apenas de uma conexão com o servidor, isso torna mais fácil configurar regras de *Firewall*. (Zabbix SIA, s.d.)

O Zabby *Proxy* é ideal para ser usado para:

- Monitoramento de regiões remotas;
- Monitoramento de regiões com conexão instável;
- Desafogar o Servidor Zabbix, reduzindo a carga de processamento, quando milhares de dispositivos estão sendo monitorados;
- Simplifica a manutenção do monitoramento distribuído.

4.2 pfSense

O pfSense é uma distribuição de *firewall* de rede gratuita e código livre, baseado no sistema operacional FreeBSD, a aplicação dispõe de um sistema de pacotes que possibilita a instalação de *softwares* criados por terceiros, deste modo é capaz de fornecer a mesma funcionalidade ou até mais que *firewalls* comuns.

A distribuição inclui uma *interface web* para realização de configuração de seus componentes, como regras de *firewall*, instalação de pacotes, verificação de estados dos serviços, *prompt* de comando e visualização de tráfego na *interface* de rede.

4.2.1 Pré-requisitos

O requisito mínimo para instalação do pfSense é 500Mhz de CPU, 512 MB de memória RAM e disco rígido com espaço de 1 Gibabyte (GB), mas o recomendado é 1 GigaHertz(GHz) de CPU, 1 GB de memória RAM e disco rígido com espaço de 1 GB. Observe-se que o requisito mínimo não é adequado para todos os ambientes, pois em uma grande ambiente com centena ou milhares de conexões a quantidade de CPU, memória RAM e espaço de disco rígido deve ser maior.

4.2.2 Sistema de pacotes

O pfSense possibilita a instalação de pacotes de terceiros a partir do seu sistema de pacotes, que possibilita fornecer a mesma funcionalidade ou até mais que *firewalls* comuns. Para o trabalho em questão foi utilizado os pacotes Zabbix Agent 5.2 e Zabbix Proxy 5.2, deste modo possibilita a distribuição além de realizar trabalho como *firewall*, também fazer como agente e *proxy*.

A instalação dos pacotes ocorre de modo rápido sem necessidade de nenhum conhecimento do usuário em UNIX, ou de usar a linha de comando para nada e nem realizar edição manual.

4.3 Grafana

O Grafana é um *software* de aplicação *web* e código aberto, iniciado em 2014 por Torkel Ödegaard, se tornou um dos projetos mais populares no GitHub. O projeto permite a visualização, consulta, registros e alerta de métricas em painéis criados de forma personalizadas pelo usuário, que permite uma melhor visualização das informações no NOC.

Para realizar a integração entre o Grafana e Zabbix é necessário o uso de um *plugin* desenvolvido pelo russo Alexander Zobnin. Para fazer uso do *plugin* a versão do Zabbix deve ser a partir da 2.0. Com a instalação e configuração do *plugin* é possível acessar a base de dados do Zabbix, e criar painéis personalizados.

5 DESCRIÇÃO DO AMBIENTE IMPLEMENTAÇÃO

Neste capítulo é descrito a construção do ambiente para implementação do trabalho como, ambiente utilizado para criação do servidor Zabbix e página *web* utilizado para acessar a *interface* do Zabbix e Grafana, sem necessidade de acesso direto ao servidor.

5.1 Servidor em nuvem

O servidor Zabbix foi implementado utilizando o serviço da *Amazon Web Services* (AWS), é uma plataforma disponibiliza serviços de computação em nuvem lançado em 2007. O AWS apresenta diversos produtos, o escolhido para implementação foi *Amazon Elastic Compute Cloud* também chamado de EC2, que fornece capacidade computacional redimensionável por meios de Máquinas Virtuais – *Virtual Machine* (VM) ou instancias, disponibilizando um serviço *web* com capacidade computacional seguro e redimensionável. Com a utilização do EC2 pode-se definir e configurar o sistema operacional e os aplicativos que serão executados na instância. O procedimento de criação e configuração da instâancia está descrito no Apêndice D.

No AWS instâncias são servidores virtuais ou VM em nuvem, são disponibilizadas diferentes combinações de capacidade de memória, CPU, armazenamento e rede, tipo o escolhido foi a t2.micro, gratuita por um ano e possui recursos suficientes para execução do trabalho, com as seguintes especificações descritas na Tabela 3:

Tabela 3 - Especificação instância t2.micro

Família	**Tipo**	**vCPUs**	**Memória RAM**	**Desempenho de rede**
T2	t2.micro	1 Núcleo de 2.5 Ghz	1 GB	Baixo a moderado

Fonte: Autoria própria, adaptado de Amazon Web Services, 2021

5.2 Domínio *Web*

Um domínio *web* foi criado para não ter a necessidade de acessar a *interface* web do Zabbix e Grafana diretamente o servidor. Para criação do domínio foi escolhido o provedor Freenom, onde é ofertado diferentes tipos de domínio de forma gratuita por um ano, com direito a renovação sem necessidade de pagamento.

O domínio possui algumas limitações, como não é registrado no seu nome, o usuário que registrou aparece apenas como utilizador do nome de domínio, e a Freenom como licenciado, e não pode ser transferido para outro usuário.

5.3 Servidor DNS

Para realizar acessos a *interface web* do Zabbix e Grafana a partir do domínio registrado é necessário a utilização de um serviço DNS que é responsável por realizar a tradução e localização para o número do IP do *site* requisitado no navegador.

O Cloudflare foi o serviço de DNS escolhido, que é disponibilizado uma versão gratuita suficiente para realização do trabalho com diversos recursos como proteção de ataque que mitigar ataques apresentando um desafio computacional, gerenciamento do DNS onde é possível adicionar registros, serviço de armazenamento de *cache*, e etc.

6 DESCRIÇÃO DO EXPERIMENTO

Esse capítulo descreve a implementação de uma rede centralizada com monitoramento distribuído, utilizando as ferramentas e *softwares* mencionados nos capítulos 4 e 5, bem como a demonstração das configurações utilizadas para integração. O objetivo desta implementação é criar um ambiente de gerenciamento que possibilite a coleta de dados úteis para o monitoramento, tal como realizar um monitoramento distribuído, desta forma em uma grande rede a estação de gerenciamento não fica sobrecarregada, pois distribui sua tarefa com outros estações de gerenciamento intermediárias.

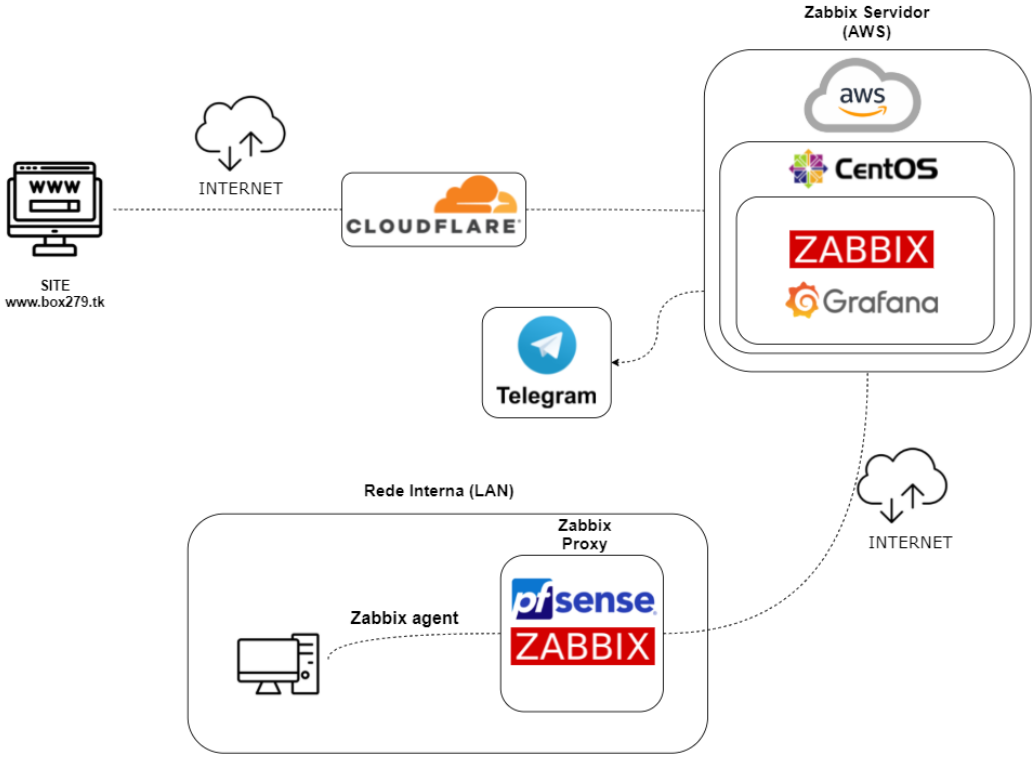
6.1 Topologia do ambiente

A Figura 18 demonstra a topologia do ambiente de rede gerenciado, e possui o agente Zabbix que realiza o envio de informação para o Zabbix *Proxy* que realiza o papel de *firewall* e gerente intermediário. Posteriormente ele envia os dados coletados para o Servidor *Zabbix* que está na nuvem, para que os dados sejam tratados e armazenados. As notificações das *triggers* além de aparecer na *interface* da ferramenta, são enviadas para o administrado por Telegram.

Para realizar o acesso *a interface web* das ferramentas, os usuários não necessitam de acessar o servidor diretamente, o acesso pode ser realizado através de uma página *web* criada, onde seus acessos passam pelos servidores DNS do Cloudflare e posteriormente para o Servidor Zabbix. Configurações de segurança foram realizadas no servidor AWS, no qual não é aceito acesso ao servidor de IPs diferente do DNS Cloudflare e a empresa ao qual o Zabbix *Proxy* está implementado.

A aplicação Grafana instalado no servidor da AWS, é utilizada para criação de painéis personalizados, contendo gráficos e informações relevantes ao administrador da rede a partir do NOC.

Figura 18 - Topologia da rede gerenciada



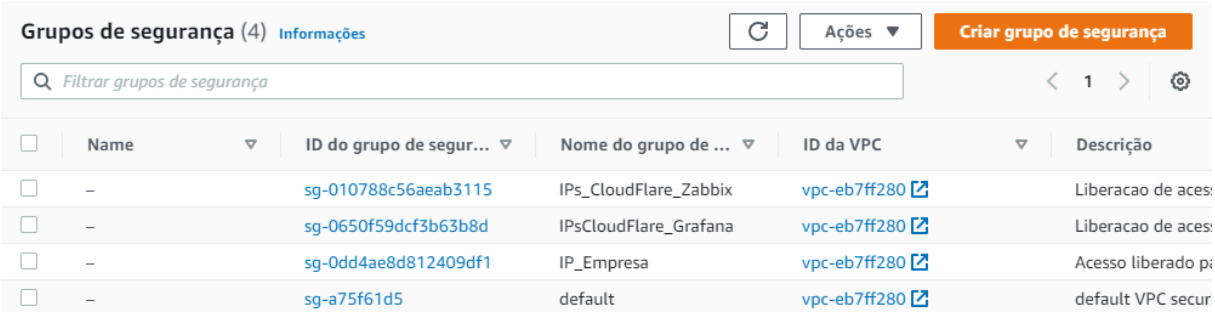
Fonte: Autoria própria, adaptado de FERNANDO, 2019

6.2 Configuração do ambiente

6.2.1 Servidor AWS

Para a segurança do servidor AWS foram configurados grupos de segurança, no grupo IPs_CloudFlare_Zabbix foi configurado as regras de entrada para aceitar apenas acessos de IPs com origem da Cloudflare e protocolo do tipo HTTP porta 80, assim permitindo acesso ao Zabbix, a mesma configuração foi realizada no grupo IPsCloudFlare_Grafana exceto o protocolo e porta, que foi configurado o protocolo TCP porta 2095 para permitir acesso ao Grafana. Por fim foi criado o grupo de segurança IP_empresa, no qual foi adicionado na regra de entrada o IP local da empresa. Deste modo, acessos ao servidor poderão ser realizados apenas pelo IP da empresa ou através da página *web* criada a qual passa pelos DNS do Cloudflare.

Figura 19 - Configuração de segurança da instancia



Fonte: Tela de captura do AWS com conteúdo desenvolvido pela autora deste trabalho

6.2.2 Serviço Web

Foi realizada a vinculação entre domino *web* a plataforma Cloudflare, no qual foi atribuído servidores DNS autoritativos, conforme mostrado na Figura 20. Esses servidores foram configurados como DNS do domínio para que seja realizado o apontamento para a plataforma da Cloudflare, sendo assim possível usufruir dos serviços disponibilizados.

Figura 20 - Servidores DNS atribuidos

Fonte: Tela de captura do Cloudflare com conteúdo desenvolvido pela autora deste trabalho

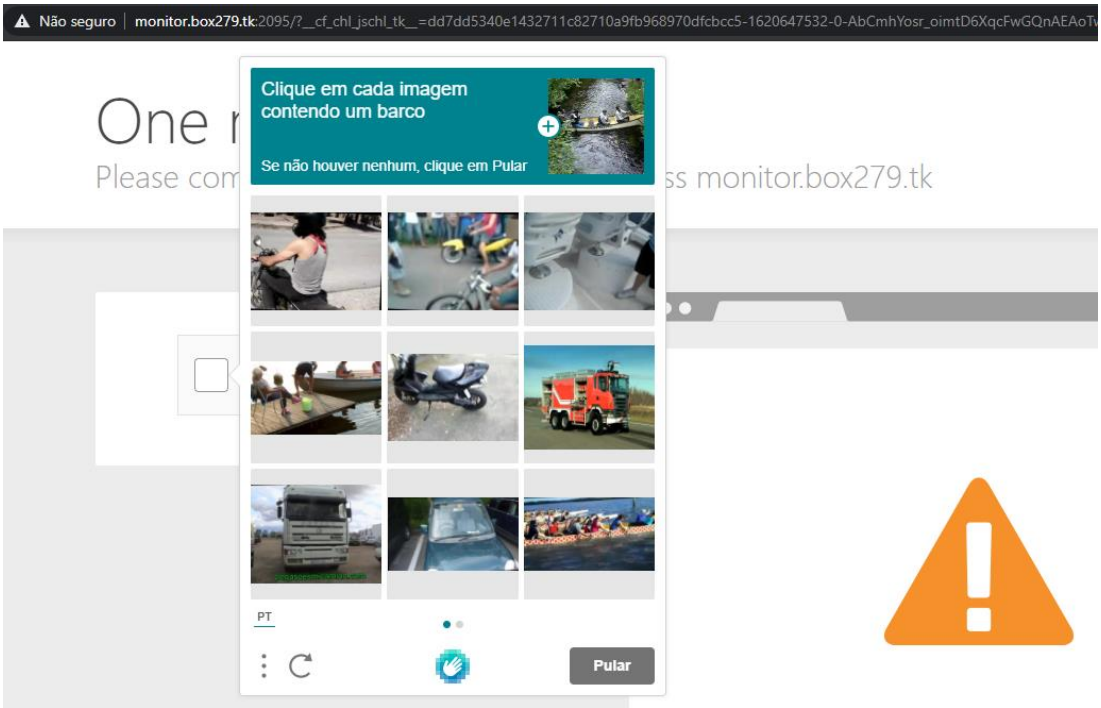
A Figura 21 ilustra a configuração da área de Gerenciamento do DNS na plataforma da Cloudflare, foi realizado o registro com nome “monitor”, criando um subdomínio apontando para o IP do servidor AWS, sendo possível realizar acesso aos serviços disponíveis a partir da Localizador Uniforme de Recursos - Uniform Resource Locator (URL) http://monitor.box279.tk/NOME_DO_ SERVIÇO_OU_PORTA.

Figura 21 - Registro no DNS

Fonte: Tela de captura do CLOUDCLARE com conteúdo desenvolvido pela autora deste trabalho

Para segurança do servidor e domínio, foi ativado o serviço “Modo Sob Ataque”, no qual quando o *site* está sob um ataque de negação de serviço ou recebendo acessos suspeitos, os visitantes são forçados a realizar um desafio para provar tal acesso não esteja sendo realizado por robôs, como ilustrado na Figura 22.

Figura 22 - Desafio proposto ao visitante

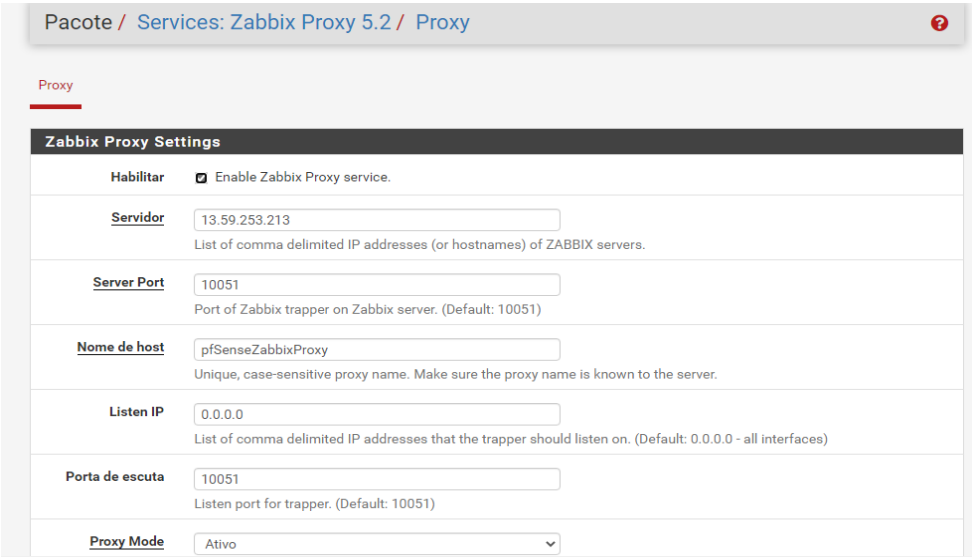


Fonte: Tela de captura do CloudFlare com conteúdo desenvolvido pela autora deste trabalho

6.2.3 pfSense

Para utilizar o pfSense como os serviço de *proxy* e agente do Zabbix, foi necessário a instalação de dois pacotes zabbix-agent52 e zabbix-proxy52. Após instalados os pacotes, foi realizado a configuração dos mesmos, no qual no Zabbix *Proxy* foi inserido o IP do servidor Zabbix, o *hostname* do *proxy* e o modo do *proxy* (ativo), conforme mostrado na Figura 23.

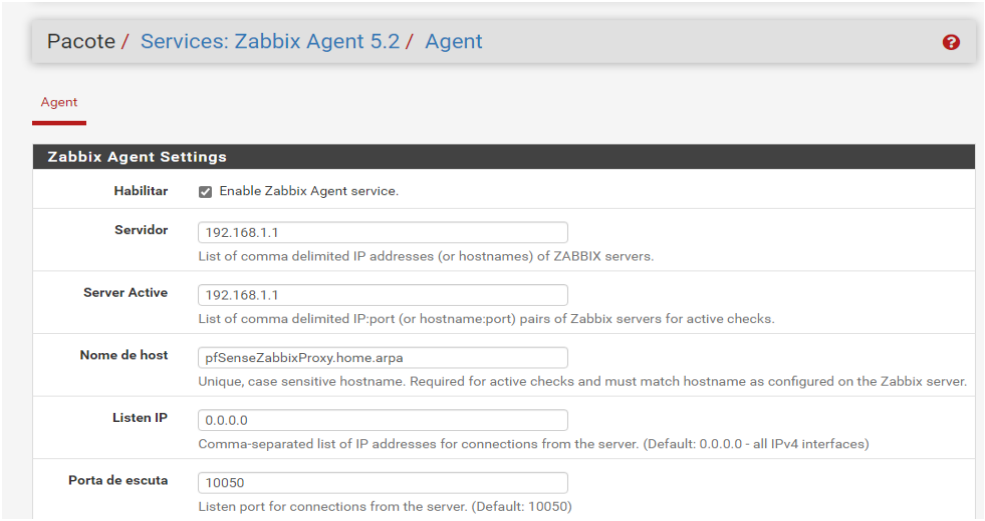
Figura 23 - Configuração Zabbix Proxy no pfSense



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Para configurar o Zabbix *Agent* foi inserido o IP do servidor local e seu *hostname*, o agente foi configurado de forma ativo, conforme ilustrado na Figura 24.

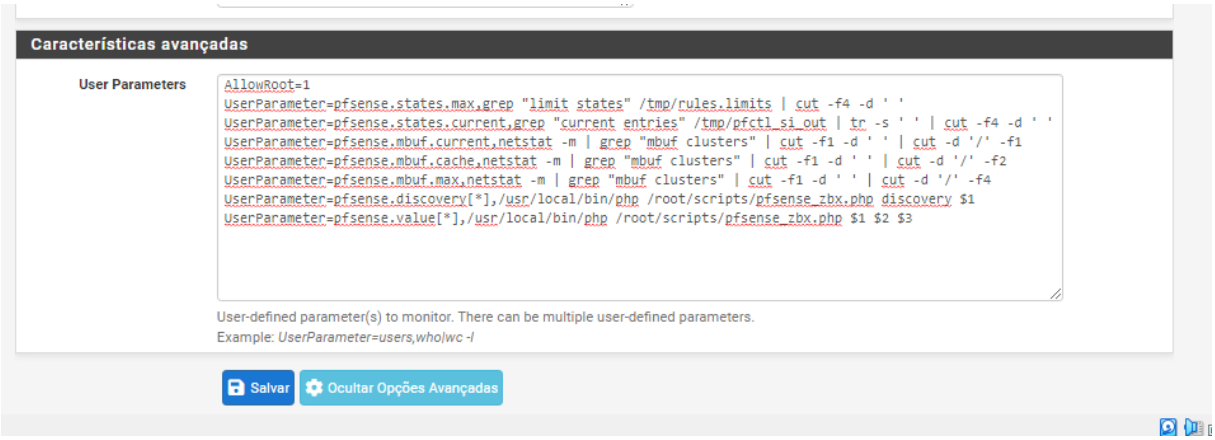
Figura 24 - Configuração Zabbix *Agent* pfSense



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Após realizar as configurações na *interface web* do pfSense, foi necessário adicionado o *proxy* a configuração do Zabbix *Server*, com o objetivo de receber as informações coletadas. Para obter dados mais detalhados do *proxy* foi utilizado um *template* personalizado desenvolvido pelo italiano Ricaardo Bicelli e disponível na plataforma GitHub. Em que é possível obter melhores informações sobre o pfSense como utilização das *interface*s de rede, disponibilidade dos serviços, alerta de atualização de versão do pfSense e pacotes. Para utilizar o *template* foi necessário a realização do *download* e importar o mesmo para a ferramenta Zabbix.e após importar o *template*, foi necessário incluir os seguintes parâmetros na configuração do pacote Zabbix Agent no pfSense, conforme apresentado na Figura 25.

Figura 25 - Paramentros Zabbix *Agent* no pfSense



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Para que as coletas sejam realizadas no pfSense é necessário realizar a inclusão de um *script Hypertext Preprocessor* (PHP) na sua configuração, no qual foi criado uma pasta com nome *scripts* e um arquivo chamado pfsense_zbx.php. a criação foi utilizando o *prompt* de comando disponível na *interface web* e os seguintes comandos.

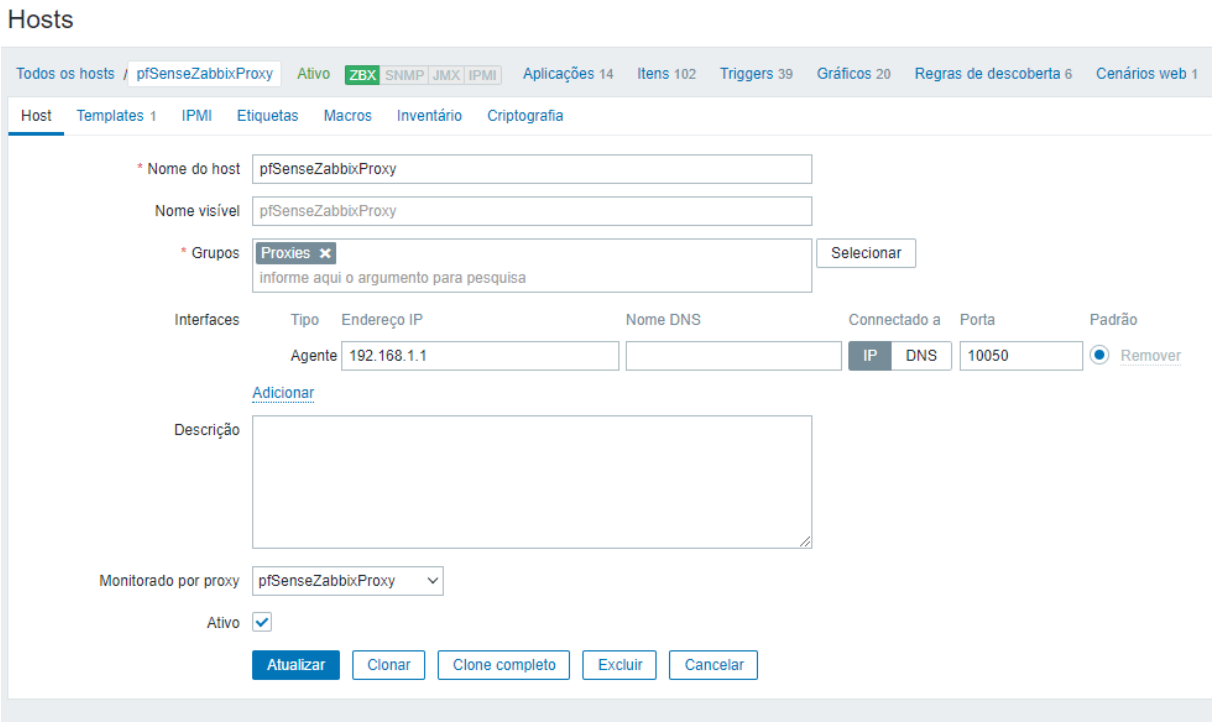
Para que as coletas de dados sejam realizadas no pfSense é necessário realizar a inclusão de um *script* PHP em sua configuração, no qual é preciso criar uma pasta com nome *scripts* e um arquivo chamado pfsense_zbx.php. Sua *interface web* disponibiliza um *prompt* de comando e os seguintes comandos para utilizados para o ato:

```
mkdir /root/scripts
```

```
curl -o /root/scripts/pfsense_zbx.php
https://raw.githubusercontent.com/rbicelli/pfsense-zabbix-
template/master/pfsense_zbx.php
```

Logo após a criação e configuração do *template*, foi realizado a criação do *host*, e foi inserido o nome, grupo de *hosts* ao qual pertence, tipo de *interface*, IP da *interface* e o *proxy* que realiza o monitoramento, mostrado na Figura 26. Na aba *Templates* foi escolhido o *template* personalizado já importado anteriormente.

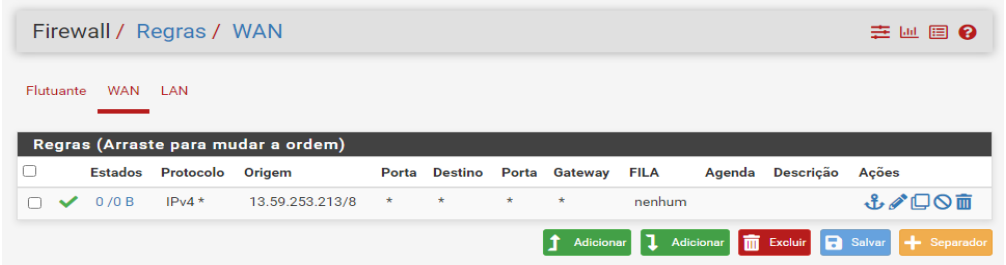
Figura 26 - Cadastro *host* pfSenseZabbixProxy



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Em sua configuração de *Firewall*, foi incluído no regras de Rede de Longa Distância - *Wide Area Network (*WAN) que aceite apenas acesso do IP pertencente ao servidor AWS, quanto a regras de Rede de Área Local - *Local Area Networks* (LAN) foi configurada para realizar todos os acessos, deste modo possibilita os dispositivos gerenciados acesso à *Internet*, como mostrado na Figura 27.

Figura 27 - Regras de WAN Firewall



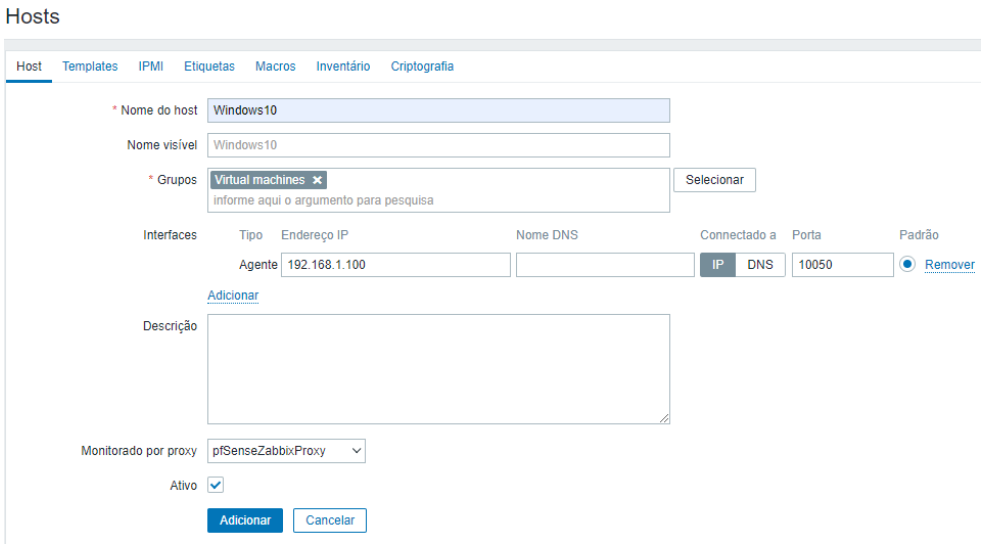
Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

6.2.4 Zabbix Agent

No dispositivo gerenciado foi realizado a instalação do pacote Zabbix Agent disponível no *site* oficial do Zabbix, configurado como agente do tipo passivo enviando os dados coletados para o servidor *proxy*.

Na *interface web* do Zabbix foi realizado a criação de um *host* para o agente, nomeado Windows10, o incluído no grupo de *hosts Virtual machines*, configurado na *interface* do tipo agente e monitorado pelo servidor *proxy*. Na aba *templates* foi adicionado o *template Windows by* Zabbix *agente*, como mostrado na Figura 28.

Figura 28 - Cadastro *host* Windows10



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

6.2.5 Integração Grafana com Zabbix

Para realizar a integração entre o Grafana com Zabbix foi realizado a instalação e configuração do *plugin* Zabbix na aplicação do Grafana, criado por Alezander Zobnin, permite a partir dele a configuração de acesso ao banco de dados do Zabbix Server e desta maneira obter informações que posteriormente possam ser utilizados para criação de painéis que possam ser utilizados no NOC, para efetuar instalação foi utilizado o seguinte de comando no Zabbix *Server*.

```
grafana-cli plugins install alexanderzobnin-zabbix-app
```

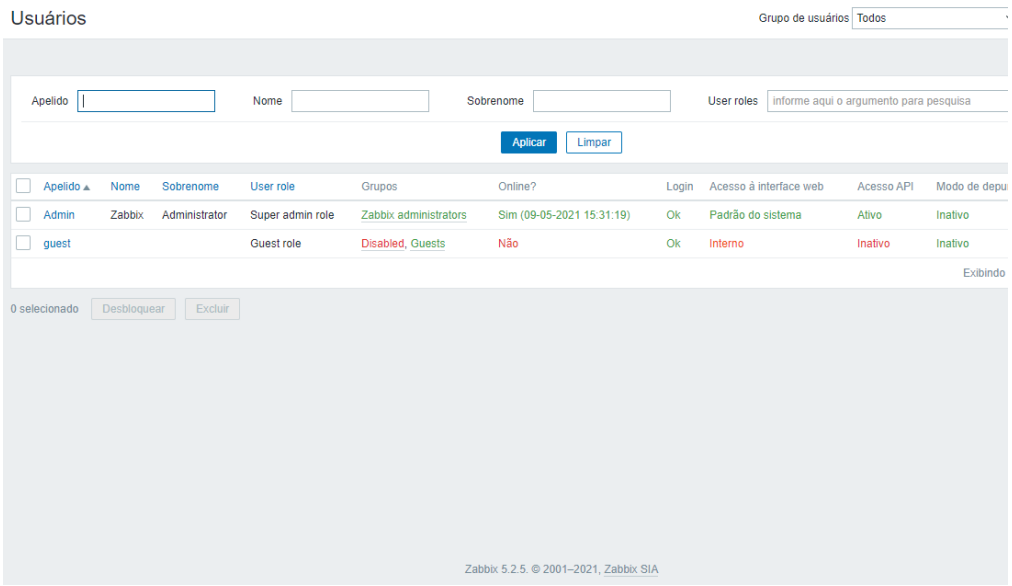
Após a instalação do *plugin*, foi necessário habilitar o mesmo pela *interface web* do Grafana, e posteriormente realizar a configuração no qual os campos URL, *Username*, *Password* e Zabbix *Version* foram alterados com seguintes dados.

```
URL = http://localhost/zabbix/api_jsonrpc.php

Username = zabbix

Password = zabbix

Zabbix Version = 5.x
```

6.2.6 Integração Zabbix com Telegram

Na *interface web* no Zabbix no menu “Administração”, opção “Usuários” e foi escolhido o usuário desejado Admin, conforme demonstrado na Figura 29.

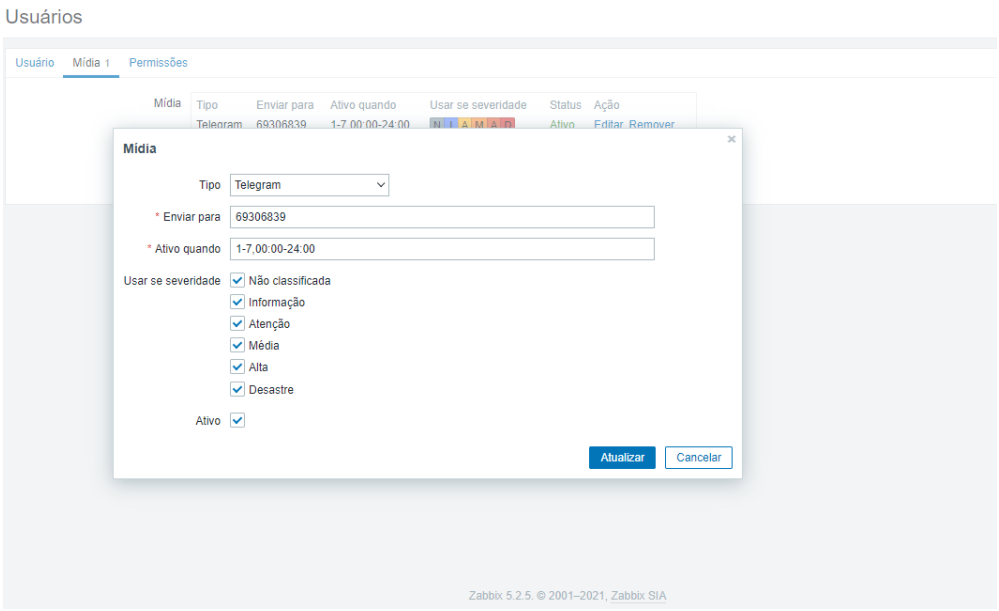
Figura 29 - Opção "Usuarios" na ferramenta Zabbix



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Logo após escolher o usuário, foi necessário acessar a aba “Mídia”, selecionar a opção “Adicionar”, e escolher o Tipo Telegram, insira a identificação do usuário para qual as mensagens devem ser enviadas, conforme ilustrado na Figura 30 e por fim adicionado.

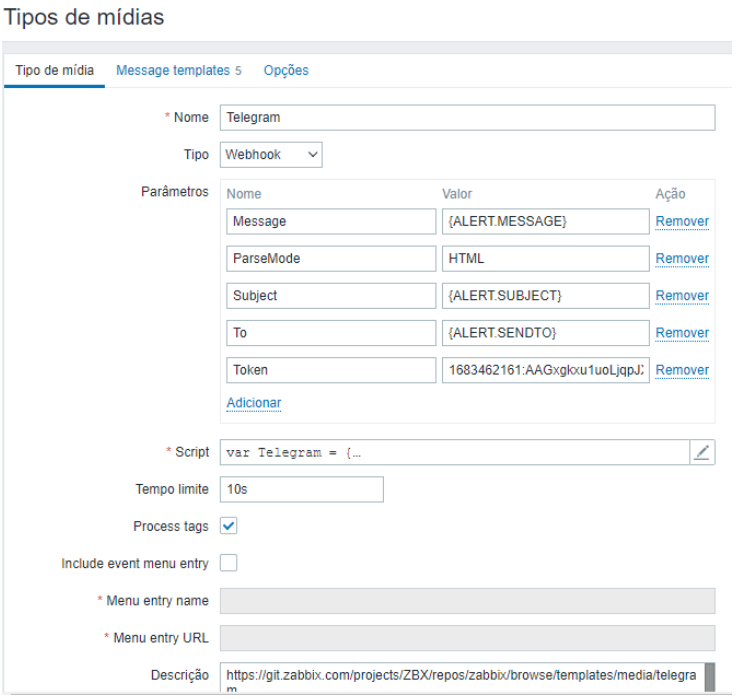
Figura 30 - Configuração mídia



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Para que o *bot* realize o envio das mensagens foi preciso adicionar o seu *Token*, foi necessário acessar o menu “Administração”, opção “Tipos de mídias”, posteriormente selecionar Telegram e no campo “Parâmetros” adicionar o *Token* conforme mostrado na Figura 31.

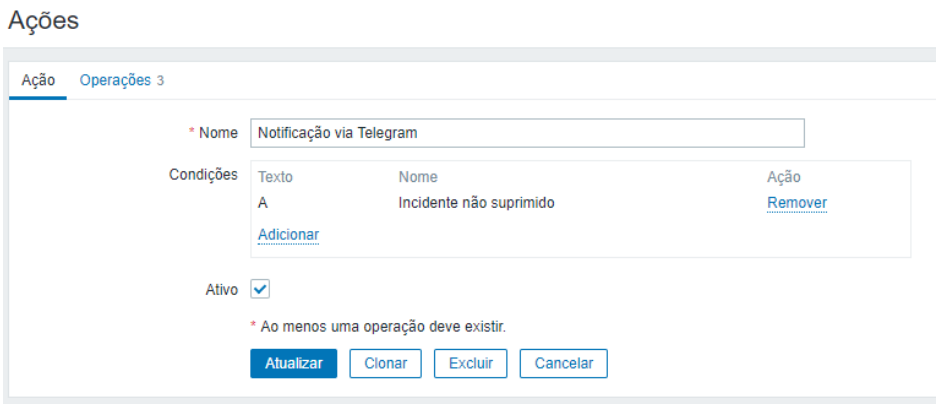
Figura 31 - Configuração "Tipos de mídias"



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Posteriormente foi preciso criar a ação que irá realizar os envios dos alertas, para isso foi necessário acessar o menu “Configuração”, opção “Ações” e selecionar a opção “Criar ações”, logo após colocado um nome a ação “Notificar via Telegram”, conforme mostrado na Figura 32, em “Condição” seleciona em “Adicionar” e selecione “Incidente suprimido” e marcando a opção “Não”.

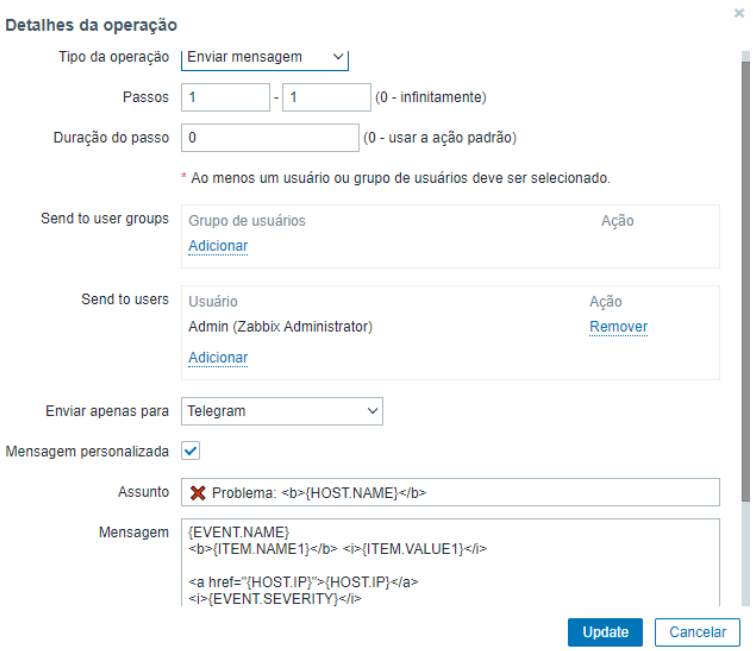
Figura 32 - Configuração de ações



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Após ter sido criado a ação foi necessário em seguida ir na aba “Operações” e no primeiro item chamado “Operações” e clicar em “Adicionar”, e em Enviar para usuário foi escolhido Admin, “Enviar apenas para” selecione Telegram, marcar “mensagem personalizada”, e foi preenchido o assunto e mensagem será enviado como ilustrado na Figura 33.

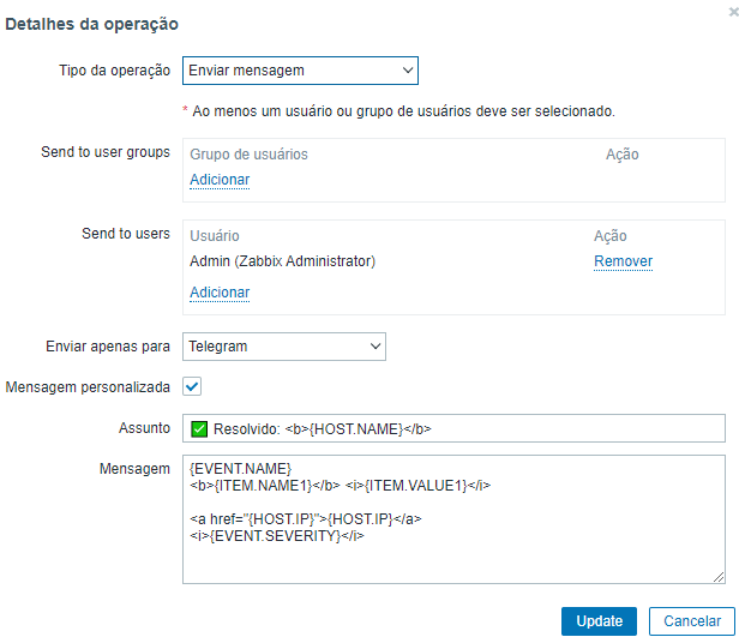
Figura 33 - Configuração de Operações



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Em “Operações de recuperação” se repete o mesmo processo anterior, exceto a mensagem, conforme mostrado Figura 34.

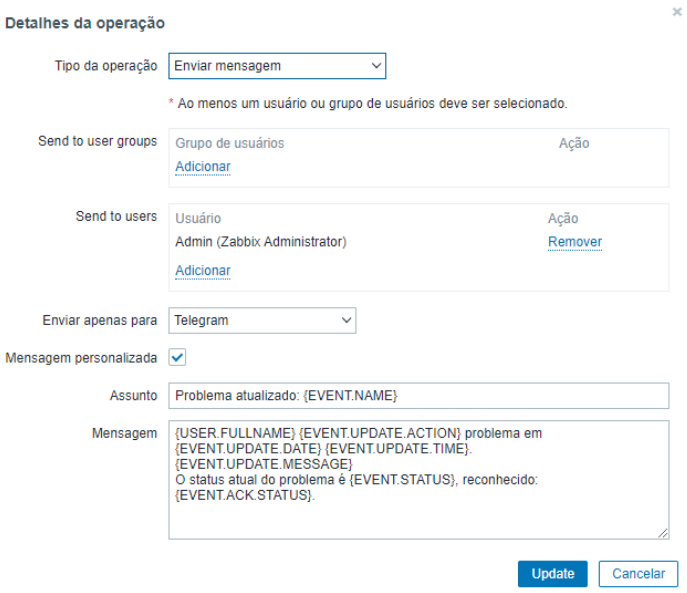
Figura 34 - Configuração de Operações de recuperação



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

No item “Operações de atualização” também foi preenchido igualmente, exceto a mensagem, conforme mostrado na Figura 35.

Figura 35 - Configuração de operações de atualização



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

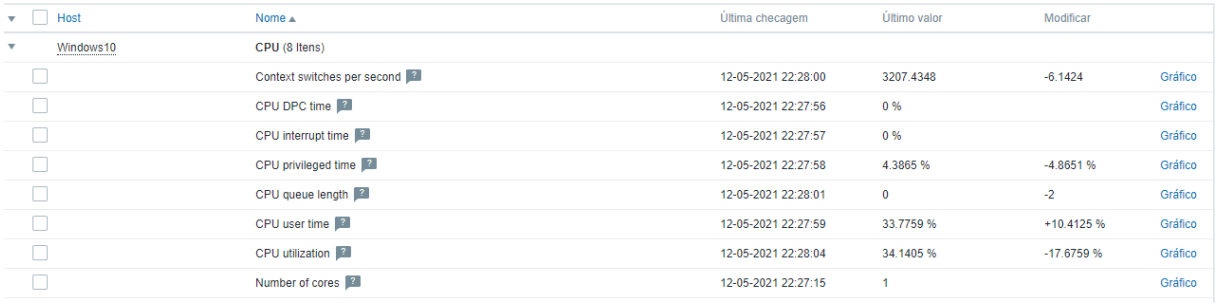
6.3 Utilização do ambiente

Após a realizar as configurações das ferramentas utilizadas no Gerenciamento de rede, foi possível visualizar o comportamento e utilização dos serviços e seus parâmetros, assim sendo plausível entender os funcionamentos e suas contribuições no Gerenciamento de rede.

6.3.1 Zabbix

Após a configuração do ambiente, foi possível visualizar o comportamento da rede. No menu “Monitoramento” opção “Dados Recentes” é possível visualizar o nome do *host*, últimos valores coletados pela ferramenta, o momento da coleta e a taxa de variação dos valores obtidos ao longo da coleta, conforme ilustrado na Figura 36.

Figura 36 - Dados recentes

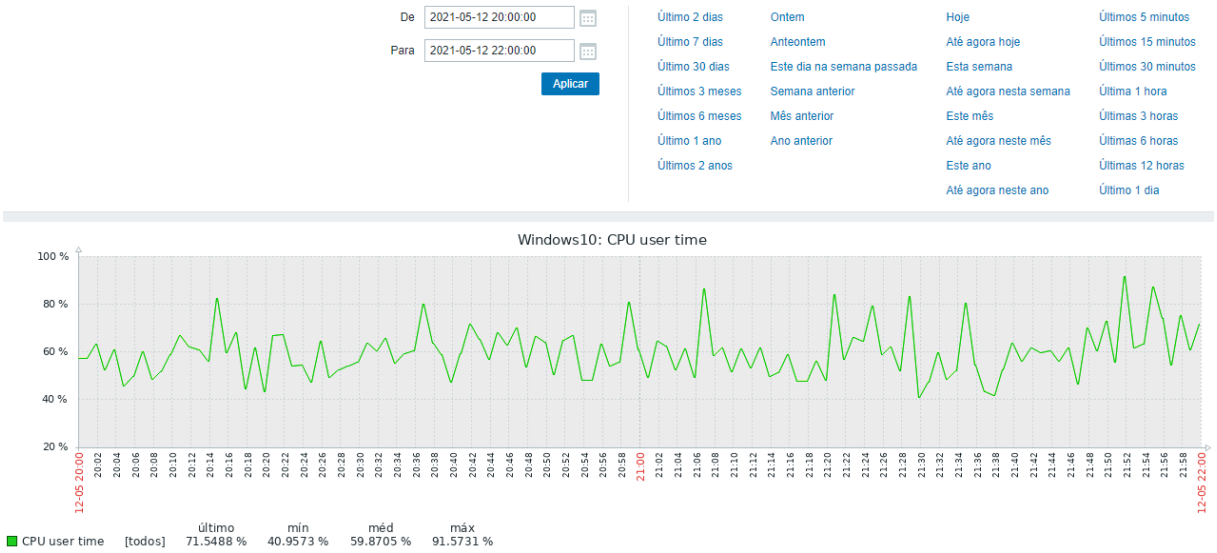


Host	Nome ▲	Última checagem	Último valor	Modificar	
Windows10	**CPU** (8 Itens)				
	Context switches per second	12-05-2021 22:28:00	3207.4348	-6.1424	Gráfico
	CPU DPC time	12-05-2021 22:27:56	0 %		Gráfico
	CPU interrupt time	12-05-2021 22:27:57	0 %		Gráfico
	CPU privileged time	12-05-2021 22:27:58	4.3865 %	-4.8651 %	Gráfico
	CPU queue length	12-05-2021 22:28:01	0	-2	Gráfico
	CPU user time	12-05-2021 22:27:59	33.7759 %	+10.4125 %	Gráfico
	CPU utilization	12-05-2021 22:28:04	34.1405 %	-17.6759 %	Gráfico
	Number of cores	12-05-2021 22:27:15	1		Gráfico

Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Ainda na opção “dados recentes” é possível a visualização em gráfico dos valores coletados, para isso é clicado na Opção “Gráfico”, e ilustra a tela presente na Figura 37, nela é possível filtrar os dados de acordo com período de coleta, na parte inferior do gráfico é disponível informações sobre a média, mínimo, máximo e último valor obtido.

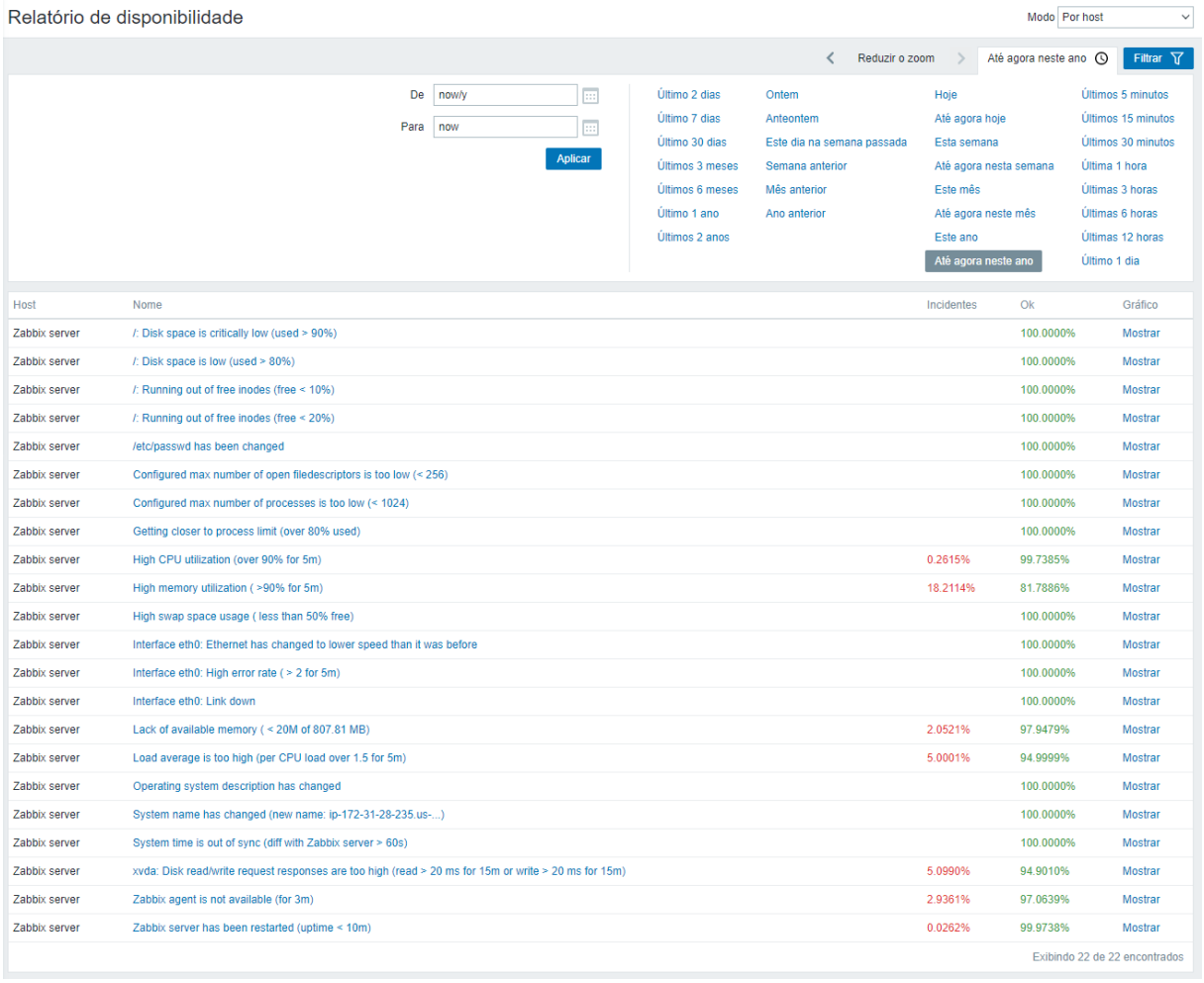
Figura 37 - Gráfico dados coletados



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

No menu “Relatórios” tem-se a opção “Relatório de disponibilidade”, no qual é possível visualizar a porcentagem de tempo que cada *trigger* passou no estado de OK/INCIDENTE, deste modo, é fácil determinar a disponibilidade dos elementos no ambiente. Conforme ilustrado na Figura 38, pode ser observado que a utilização da memória RAM no servidor ficou com a utilização acima de noventa porcento em dezoito por cento do tempo, essa informação não é uma boa notícia. Isso indica que o servidor em quase um quinto do seu tempo de utilização trabalhando com toda sua capacidade de memória RAM, assim o administrador da rede pode identificar a necessidade de aquisição aumento de sua capacidade.

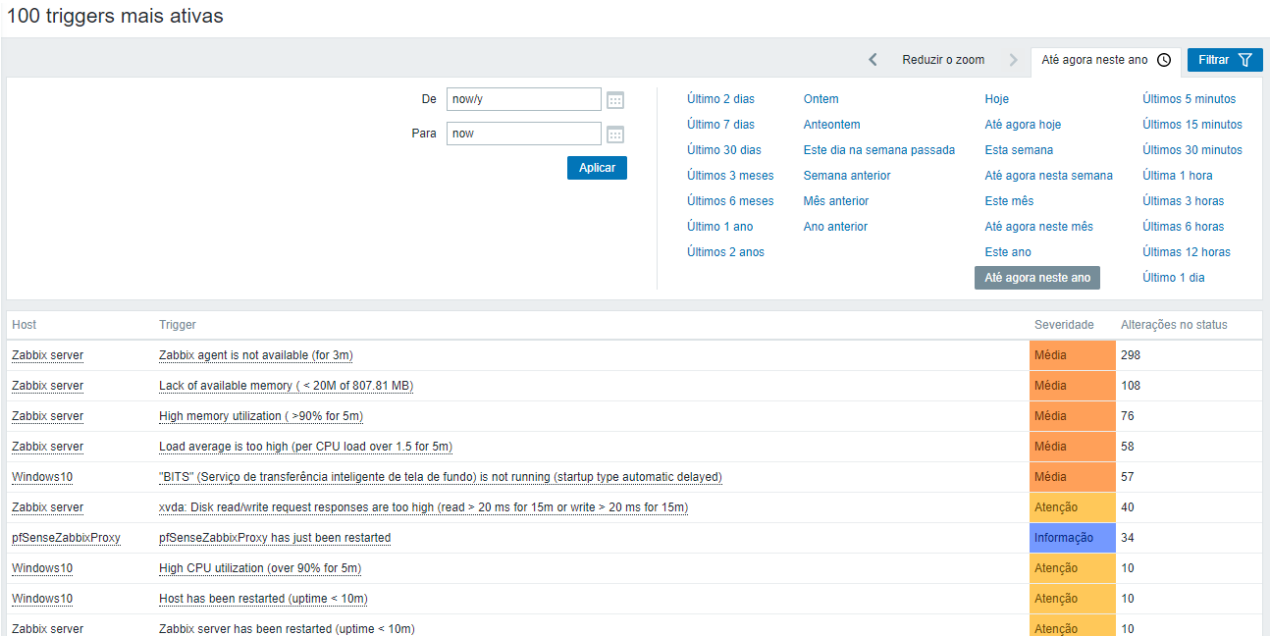
Figura 38 - Relatório de disponibilidade



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Ainda no menu “Relatórios” tem-se a opção “Top 100 de triggers”, no qual é possível visualizar às *triggers* que mais mudaram de estados em determinado período escolhido, assim é permitido realizar análise sobre o ambiente. Na Figura 39 é mostrado o relatório correspondente ao ambiente, em que é possível visualizar nome do *host*, *trigger* disparada, severidade e a quantidade de mudanças no estado.

Figura 39 - Top 100 triggers

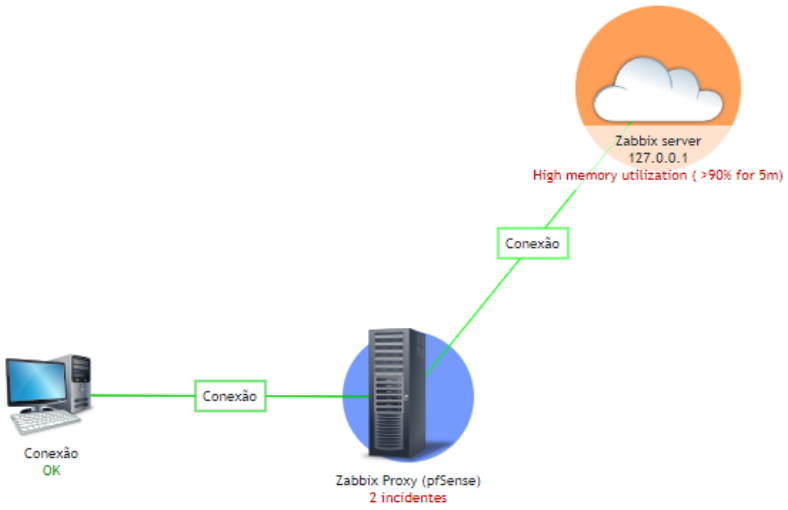


Host	Trigger	Severidade	Alterações no status
Zabbix server	Zabbix agent is not available (for 3m)	Média	298
Zabbix server	Lack of available memory (< 20M of 807.81 MB)	Média	108
Zabbix server	High memory utilization (>90% for 5m)	Média	76
Zabbix server	Load average is too high (per CPU load over 1.5 for 5m)	Média	58
Windows10	"BITS" (Serviço de transferência inteligente de tela de fundo) is not running (startup type automatic delayed)	Média	57
Zabbix server	xvda: Disk read/write request responses are too high (read > 20 ms for 15m or write > 20 ms for 15m)	Atenção	40
pfSenseZabbixProxy	pfSenseZabbixProxy has just been restarted	Informação	34
Windows10	High CPU utilization (over 90% for 5m)	Atenção	10
Windows10	Host has been restarted (uptime < 10m)	Atenção	10
Zabbix server	Zabbix server has been restarted (uptime < 10m)	Atenção	10

Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

A Figura 40 ilustra o mapa criado utilizando a ferramenta Zabbix, em que é adicionado os *hosts* cadastrados, sendo assim possível obtém-se uma visualização ampla do ambiente, na qual é mostrado o estado de cada *host* caso ocorra algum incidente, um círculo correspondente a cor da severidade é formada no entorno e descrito a quantidade ou no caso de apenas um é informado o nome.

Figura 40 - Mapa da rede

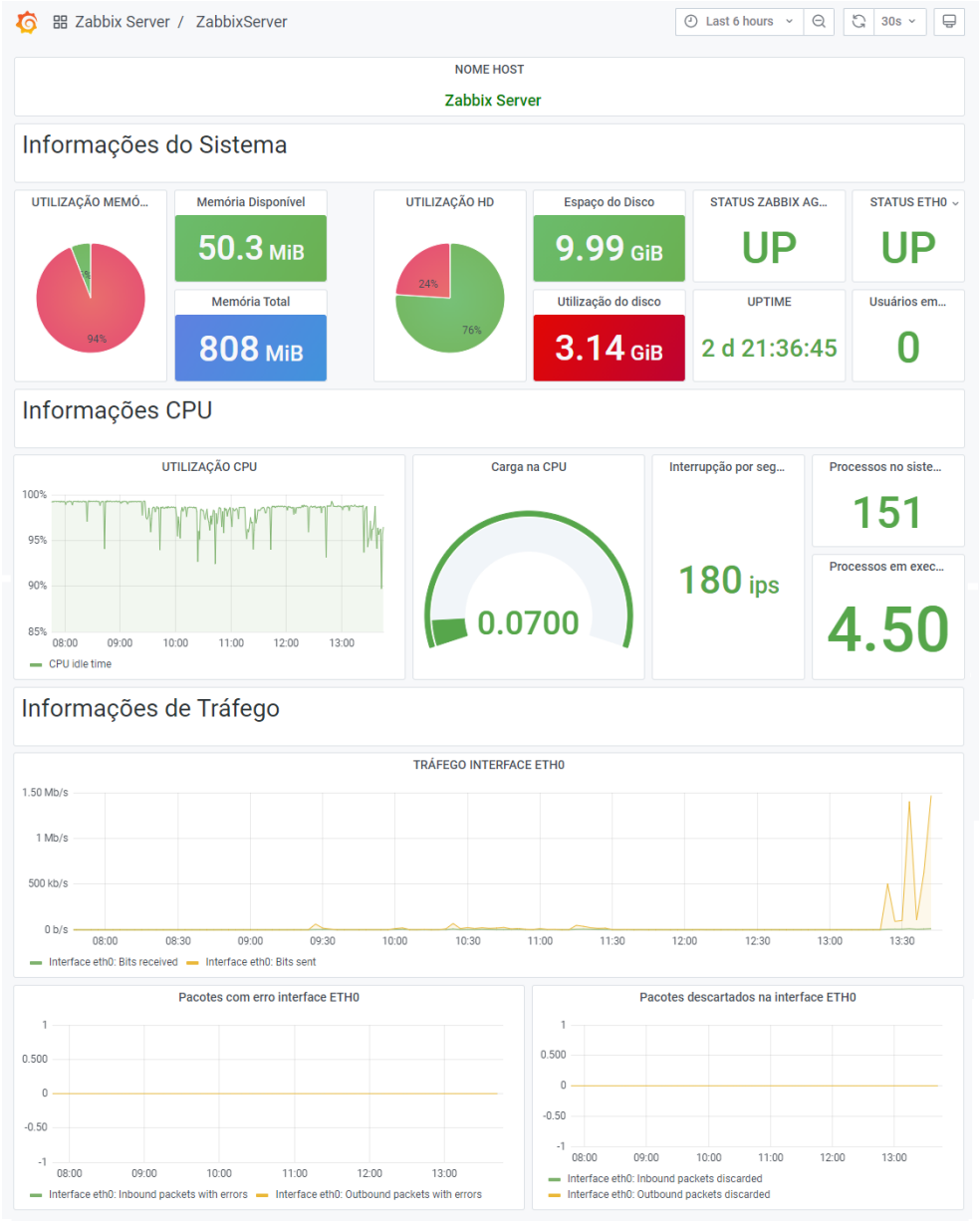


Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

6.3.2 Grafana

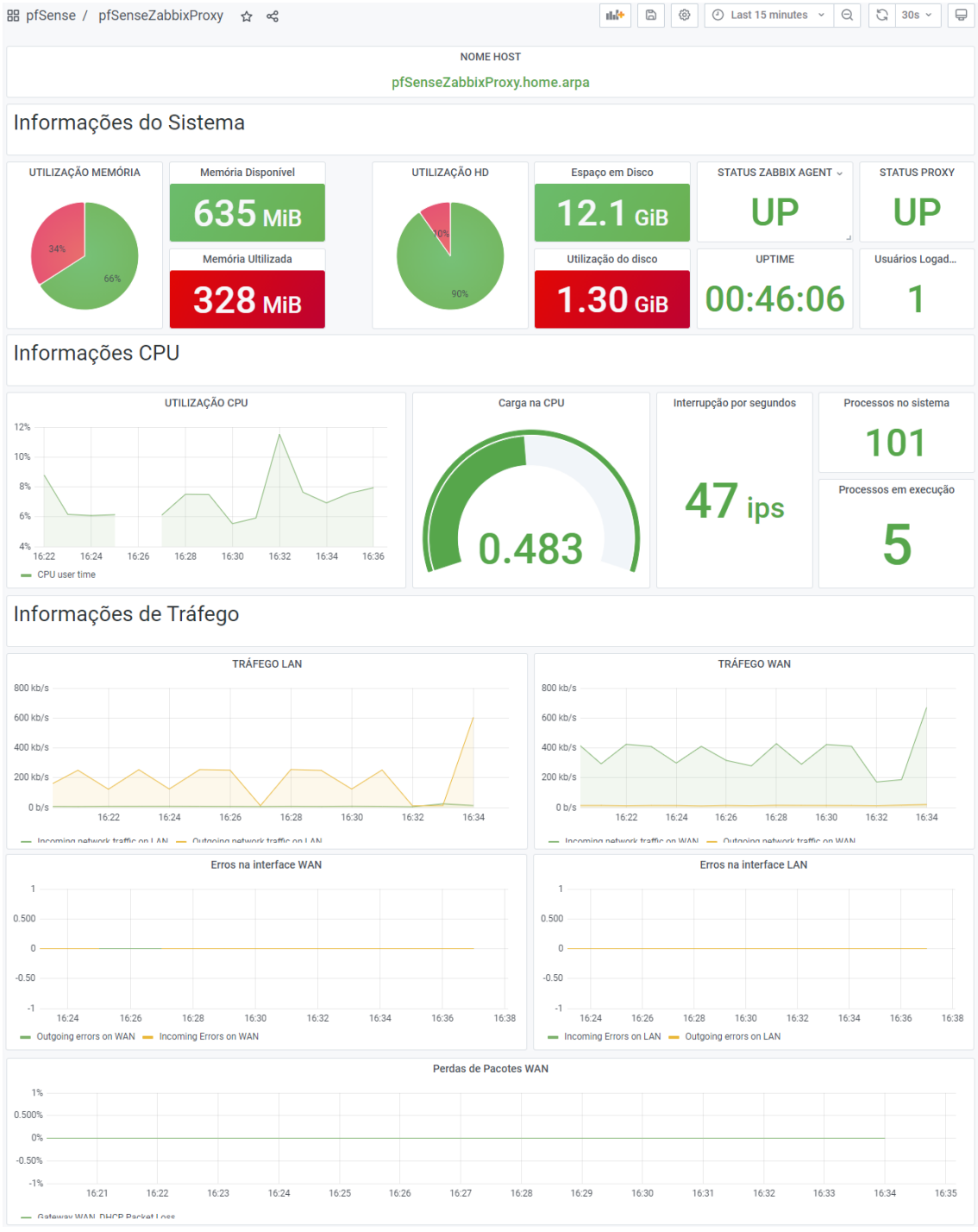
No Grafana foi realizado a criação de painéis também chamados de *dashboard*, cada *host* possui um painel contendo informações relevantes como utilização da memória RAM, CPU, disco rígido, *interface* de rede, bem como quantidade de erros na *interface* e disponibilidade do *host* as Figuras 41, 42 e 43 mostra os painéis criados para o Zabbix *Server*, Zabbix *Proxy* e o Dispositivo gerenciado, respectivamente.

Figura 41 - Painel criando para Zabbix Server



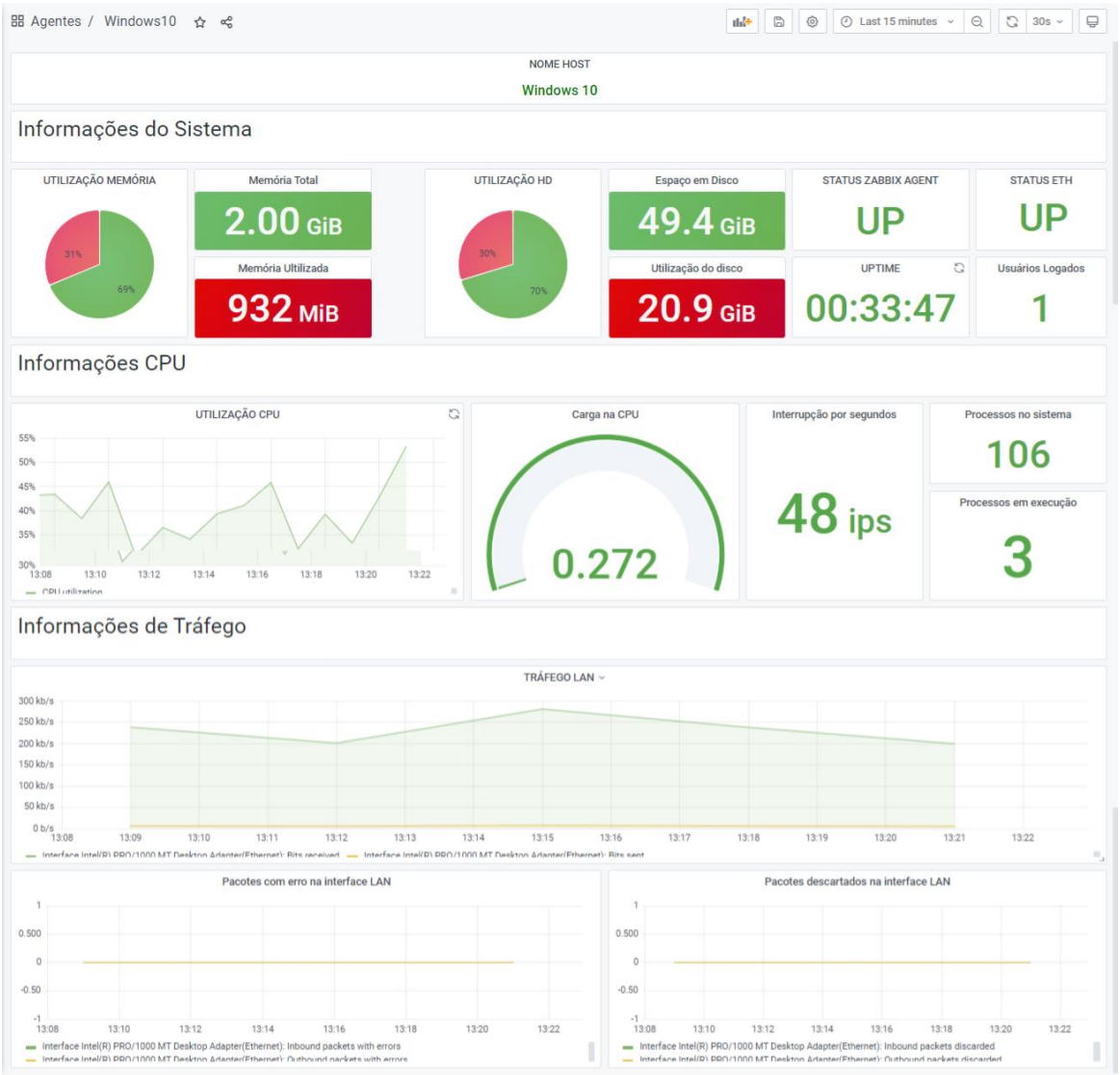
Fonte: Tela de captura do Grafana com conteúdo desenvolvido pela autora deste trabalho

Figura 42 - Painel criado para Zabbix Proxy



Fonte: Tela de captura do Grafana com conteúdo desenvolvido pela autora deste trabalho

Figura 43 - Painel criado para *host* Windows10

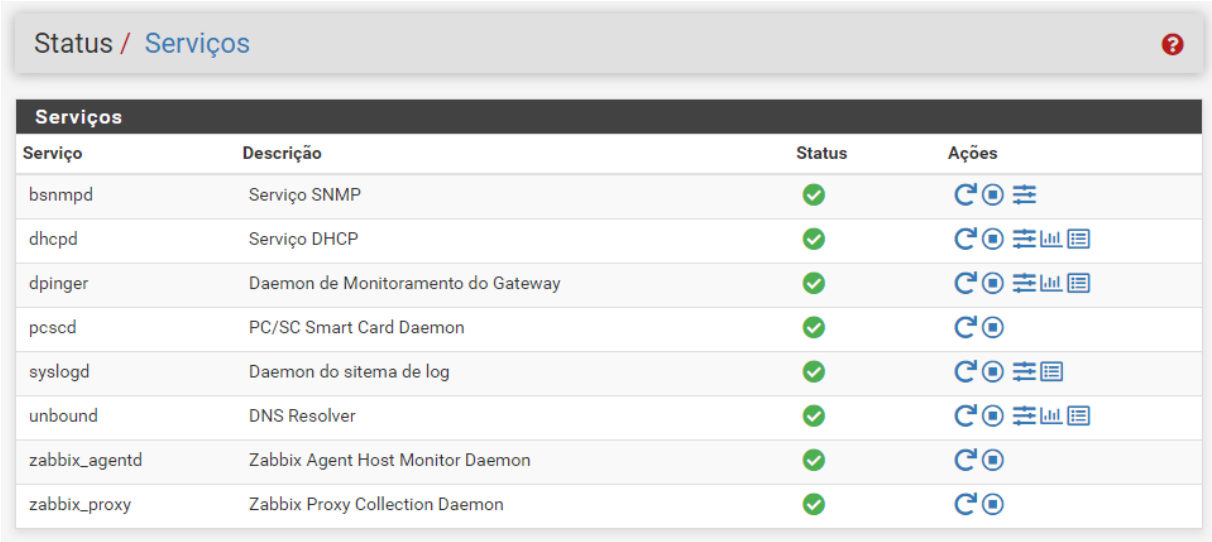


Fonte: Tela de captura do Grafana com conteúdo desenvolvido pela autora deste trabalho

6.3.3 pfSense

O pfSense foi utilizado como *firewall*, agente e *proxy*, conforme já mencionado em capítulos 4 e 6, por meio *interface web* é possível verificar o *status* dos serviços habilitados, conforme mostrado Figura 44 todos os serviços foram configurados corretamente, com *status* de funcionando.

Figura 44 - Status Serviços da distribuição pfSense

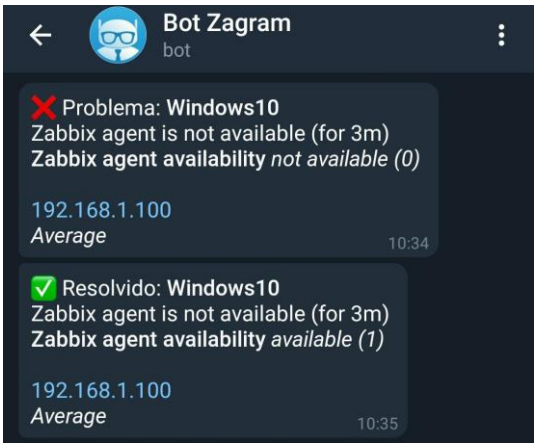


Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

6.3.4 Telegram

Todas as notificações de incidente que o Zabbix alerta, são enviados via Telegram, indicando o nome do *host*, descrição do problema e IP. Desta forma, auxilia o usuário caso não esteja observando ou acessando o NOC no momento, sendo notificado a qualquer momento. Desta forma ajuda ficar ciente de eventos que possam ocorrer com os dispositivos gerenciados e serviços utilizados. A Figura 45 apresenta como enviadas as mensagens pelo Zabbix ao usuário.

Figura 45 - Mensagem enviado pelo Zabbix



Fonte: Tela de captura do Telegram com conteúdo desenvolvido pela autora deste trabalho

7 CONCLUSÃO

O objetivo deste trabalho foi gerenciar um ambiente de rede, realizando a integração do Zabbix com diferentes ferramentas que felicitem a administração, identificação de incidentes, visualização de informações e distribuição da coleta de dados, facilidades essas são essenciais em um grande ambiente de rede, com muitos computadores o que pode tornar o gerenciamento complexo caso não haja a utilização de bons *hardware*s para maximiza a eficiente e aumentar a produtividade.

Ao longo do tempo houve aumento na demanda de dispositivos conectados à rede, deste modo surgiu a necessidade pelos administradores da utilização de ferramenta para os auxiliarem no Gerenciamento da rede e realizar tarefas como monitorar, testar, inspecionar, avaliar, configurar, analisar, controlar a rede e seus recursos, no qual o administrador poderá realizar tomadas de decisões de maneira ágeis, aumentando a eficiência e produtividade.

Uma rede sem gerenciamento pode ter problemas sérios como lentidão na entrega de serviços e sistemas, pois não há uma métrica ou *software* para realizar alerta caso a rede esteja sobrecarregada. Ainda, a não correção de incidentes com agilidade, ou previsão de incidentes ocasionam um custo operacional direto e indireto.

Há muitos casos em que as ferramentas de gerenciamento de rede podem auxiliar o administrador, como na identificação de falhas que venha a ocorrem em um dispositivo. Desta forma, a tomada de decisão ocorre de maneira mais rápida, além de monitorar o tráfego da rede, obtendo informação da utilização dos recursos e serviços.

O gerenciamento de rede é dividido em cinco áreas, Gerenciamento de desempenho, falhas, configuração, contabilização e segurança, todas elas buscando garantir um bom funcionamento da rede.

A arquitetura de um sistema Gerenciamento de rede é composto por três componentes principais, o primeiro deles é a entidade gerenciadora que é o centro da atividade, responsável por controlar a coleta, processamento, análise e apresentação das informações, nela é iniciado as ações para controlar o comportamento da rede. O segundo componente são os dispositivos gerenciados, os quais são qualquer dispositivo conectado à rede que tenha um IP válido. O terceiro componente é o protocolo de Gerenciamento de rede o qual é responsável de realizar a comunicação entre os dispositivos gerenciados e a entidade gerenciadora.

Um dos protocolos mais populares para Gerenciamento de rede é o SNMP, usado para transmitir informações e comandos entre a entidade gerenciadora e os dispositivos gerenciados, com o objetivo de monitorar dispositivos conectado à rede. O SNMP possui três versões até o momento da escrita deste trabalho, a versão recente se destaca com sua robustez na melhoria de segurança, algoritmos de criptografia e autenticação.

A partir da versão dois do SNMP foi possibilitado a organização da estrutura de rede de forma hierarquia, desta forma em uma rede com centenas ou milhares de computadores, é possível distribuir as tarefas entre entidades gerenciadoras intermediárias que realiza a coleta dos dados e posteriormente os enviam para entidade gerenciadora superior, desta forma é reduzido o tráfego gerado pelas coletas e não sobrecarrega a entidade gerenciadora superior.

O Zabbix foi a principal ferramenta escolhida por apresentar uma ampla possibilidade de coleta de dados, e permitir a integração com outras ferramentas como pfSense, Grafana e Telegram, deste modo enriquecendo o gerenciamento da rede. Foi possível visualizar na descrição do experimento que é possível realizar a integração com diversas ferramentas de gerenciamento centralizado, envio de notificações e visualizações dos dados coletados.

7.1 Dificuldades encontradas

Houve dificuldades durante o desenvolvimento deste trabalho, a qual é atribuída ao servidor AWS, no qual é disponibilizado apenas um núcleo de processamento na CPU e 1 GB de memória RAM, sendo insuficiente para manter um serviço de gerenciamento da rede por longos períodos utilizando diversos *software*. Ao longo do desenvolvimento de acordo com aumento de itens monitorados e *sotfware* utilizados, a memória RAM permanece com sua utilização acima de noventa por cento por um longo período e a CPU apresenta alguns picos de utilização, ocasionou em lentidão nos serviços implantados no servidor, bem como sua indisponibilidade em alguns momentos.

7.2 Sugestão de trabalhos futuros

Como sugestão para quem deseja-se continuar com este trabalho, sugere-se:

- Implementar agente Zabbix na plataforma Android;

- Realizar monitoramento de acesso à serviço de *streaming* no ambiente de rede;
- Criação de *script* que possa gerar uma ação quando disparado uma *trigger*, e reestabelecer automaticamente o serviço.

8 REFERÊNCIAS

AMAZON WEB SERVICES. **AWS**: Serviços de computação em nuvem. [S. l.], 2021. Disponível em: https://aws.amazon.com/pt/. Acesso em: 16 mar. 2021.

CLOUDFLARE, INC. **Cloudflare**: The Web Performance & Security Company. [S. l.], 2021. Disponível em: https://www.cloudflare.com/pt-br/. Acesso em: 16 mar. 2021.

ELECTRIC SHEEP FENCING. **PfSense Appliance Guidance**. [*S. l.*]. Disponível em: https://www.pfsense.org/products/. Acesso em: 19 mar. 2021.

ELECTRIC SHEEP FENCING. **PfSense 2.5.0**: World‘s Most Trusted Open Source Firewall. [S. l.], 2021. Disponível em: https://www.pfsense.org/download/. Acesso em: 31 mar. 2021.

FERNANDO, Luiz. **Monitoramento pfSense-Zabbix**, 17 dez. 2019. Disponível em: https://www.youtube.com/playlist?list=PL3Sj98RICiBGMRfU0PvWJodPGMPs8tetO. Acesso em: 1 mar. 2021.

GRAFANA LABS. **Grafana Documentation**. [*S. l.*]. Disponível em: https://grafana.com/docs/. Acesso em: 19 mar. 2021.

GRAFANA LABS. **Grafana 7.5.0**: The open observability platform. [S. l.], 25 mar. 2021. Disponível em: https://grafana.com/grafana/download/7.5.0. Acesso em: 31 mar. 2021.

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.

LIMA, Janssen Dos Reis *et al*. **Monitoramento de Redes com ZABBIX**: Monitore a saúde dos servidores e equipamentos de rede. 1. ed. Rio de Janeiro: Brasport, 2014. ISBN 978-85-7452-651-5.

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.

MICROSOFT. **Windows**: 10. [S. l.], 2021. Disponível em: https://www.microsoft.com/pt-br/software-download/windows10. Acesso em: 16 mar. 2021.

PESSOL, Luiz. **PfSense - instalação básica do pfSense 2.4.5**. [*S. l.*], 5 abr. 2020. Disponível em: https://youtu.be/ccxXuson7O0. Acesso em: 19 mar. 2021.

PUTTY.ORG. **PuTTY**: a free SSH and telnet client for Windows. [S. l.], 2021. Disponível em: https://www.putty.org/. Acesso em: 16 mar. 2021.

REMONTII, Rudimar. **Instalação do Zabbix 5 + notificações pelo Telegram nativo + Grafana 7 + Debian 10 Buster**. [*S. l.*], 22 maio 2020. Disponível em: https://blog.remontti.com.br/4348. Acesso em: 11 mar. 2021.

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

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.

TELEGRAM. **Telegram**: Messenger. 2.6.3. [S. l.], 2021. Disponível em: https://telegram.org/apps. Acesso em: 16 mar. 2021.

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 Documentation 5.2**. Disponível em: https://www.zabbix.com/documentation/current/pt/manual. Acesso em: 29 abr. 2021.

ZABBIX SIA. **Zabbix 5.2**: The Enterprise-Class Open Source Network. [S. l.], 2020. Disponível em: https://www.zabbix.com/download?zabbix=5.2. Acesso em: 01 nov. 2020.

APÊNDICE A - INSTALAÇÃO ZABBIX

Sistema Operacional utilizado: CentOS8

Pré-requisitos

- Servidor Web Apache
- PHP
- Servidor MySQL

Primeira etapa – Realizar atualização de pacotes do sistema.

```
yum update -y
```

Segunda etapa – Instação do repositório do Zabbix.

```
# rpm -Uvh https://repo.zabbix.com/zabbix/5.2/rhel/8/x86_64/zabbix-release-5.2-1.el8.noarch.rpm
# dnf clean all
```

Terceira etapa – Instalação do Zabbix Server, Fronted e Agente.

```
# dnf install zabbix-server-mysql zabbix-web-mysql zabbix-apache-conf zabbix-agent
```

Quarta etapa – Instalação do editor de texto Nano

```
yum install nano -y
```

Quinta etapa – Criar e configurar o banco de dados.

Login no banco de dados MySQL.

```
# mysql -uroot -p
```

Criação do banco de dados.

```
mysql> create database zabbix character set utf8 collate utf8_bin;
```

Criação de usuário e senha do banco de dados.

```
mysql> create user zabbix@localhost identified by 'password';
```

Conceder privilégios ao usuário criado anteriormente.

```
mysql> grant all privileges on zabbix.* to zabbix@localhost;
```

Sair do MySQL.

```
mysql> quit;
```

Importação do esquema e dados iniciais.

```
zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p
zabbix
```

Modificar o as configurações do Zabbix Server com os detalhes definidos na criação do banco de dados.

```
nano /etc/zabbix/zabbix_server.conf
```

Modificar os seguintes campos.

```
DBHost=localHost
DBName=zabbix
DBUser=zabbix
DBPassword=password
```

Alterar o fuso horário editando o arquivo zabbix.conf

```
nano /etc/httpd/conf.d/zabbix.conf
```

Modificar o seguinte campo “date.timezone” com a seguinte informação.

```
php_value date.timezone America/Sao Paulo
```

Inicie o servidor zabbix e os processos do agente.

```
systemctl restart zabbix-server zabbix-agent httpd php-fpm
systemctl enable zabbix-server zabbix-agent httpd php-fpm
```

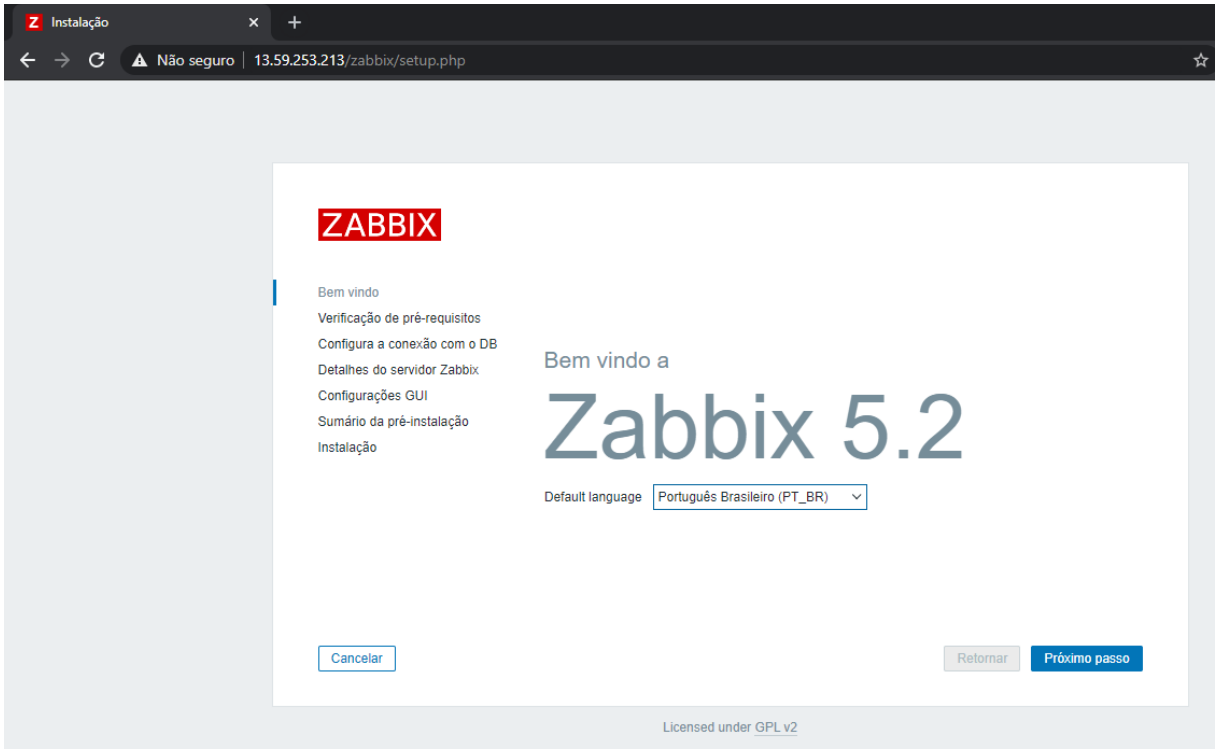
Sexta etapa – Configurar o Zabbix

O Zabbix poderá ser acessado usando o seguinte endereço URL:

http://ip_ou_nome_sevidor/zabbix

Será apresentada a página de boas-vindas contendo a informação da versão do Zabbix, conforme a Figura 46, clicar em próxima etapa.

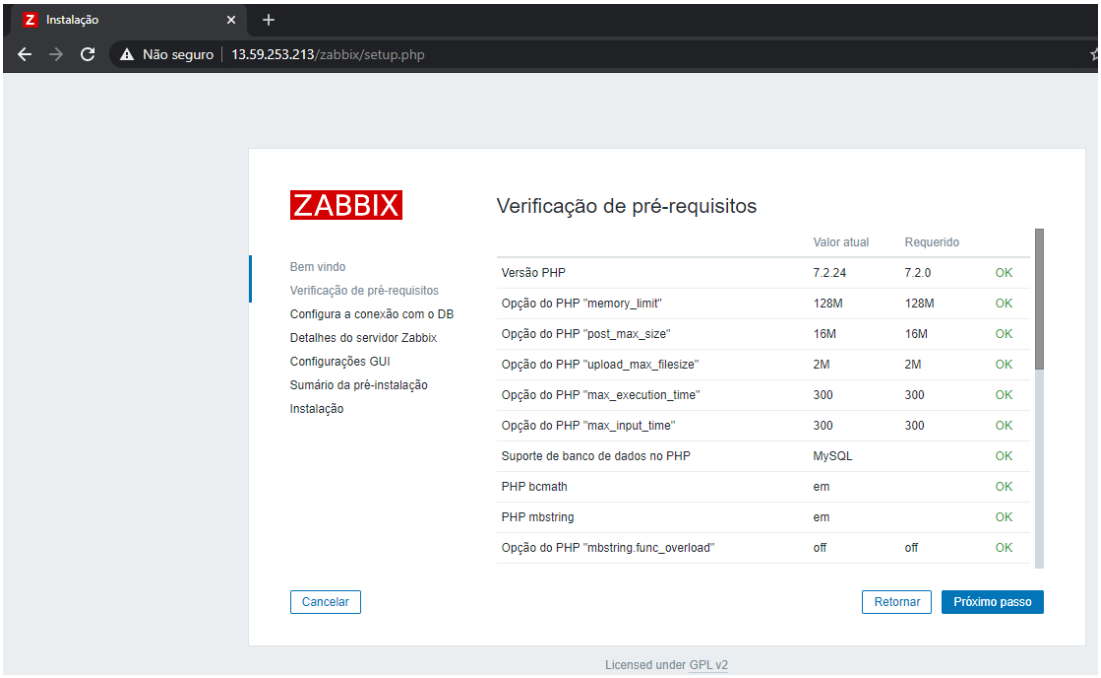
Figura 46 - Tela boas-vindas Zabbix



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Na próxima etapa conforme a Figura 47, apresentará informações dos pré-requisitos incluindo o seu status onde todas devem estar com 'OK', estando todos em conformidade clicar em próxima etapa.

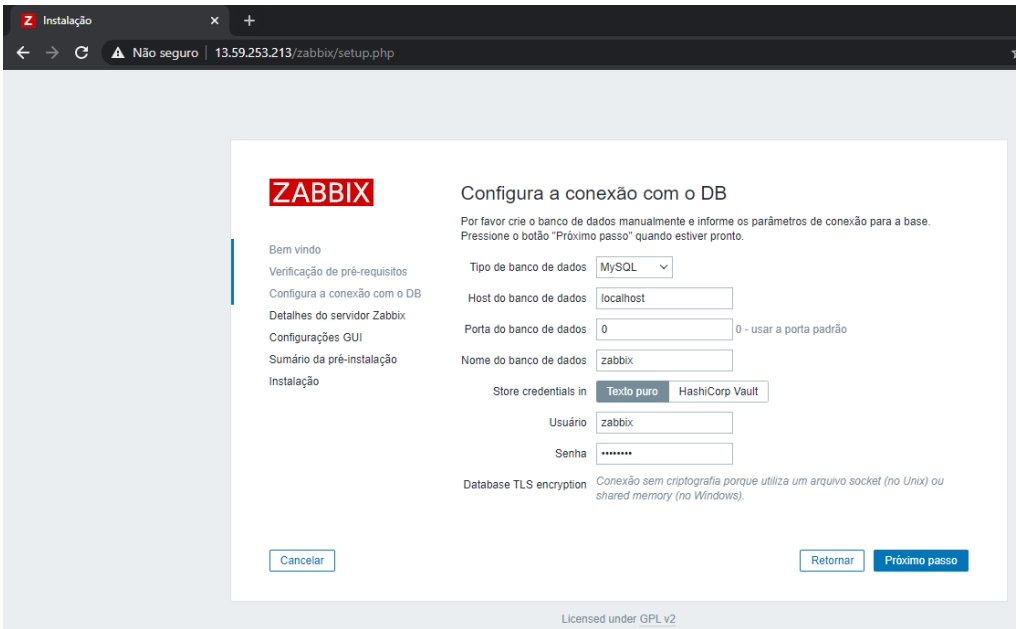
Figura 47 - Tela verificação de pré-requisitos



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Na próxima tela conforme a Figura 48, deve ser preenchido as configurações do banco de dados, conforme criado anteriormente na quinta etapa. Sendo inseridos os dados de forma correta clica em próximo passo.

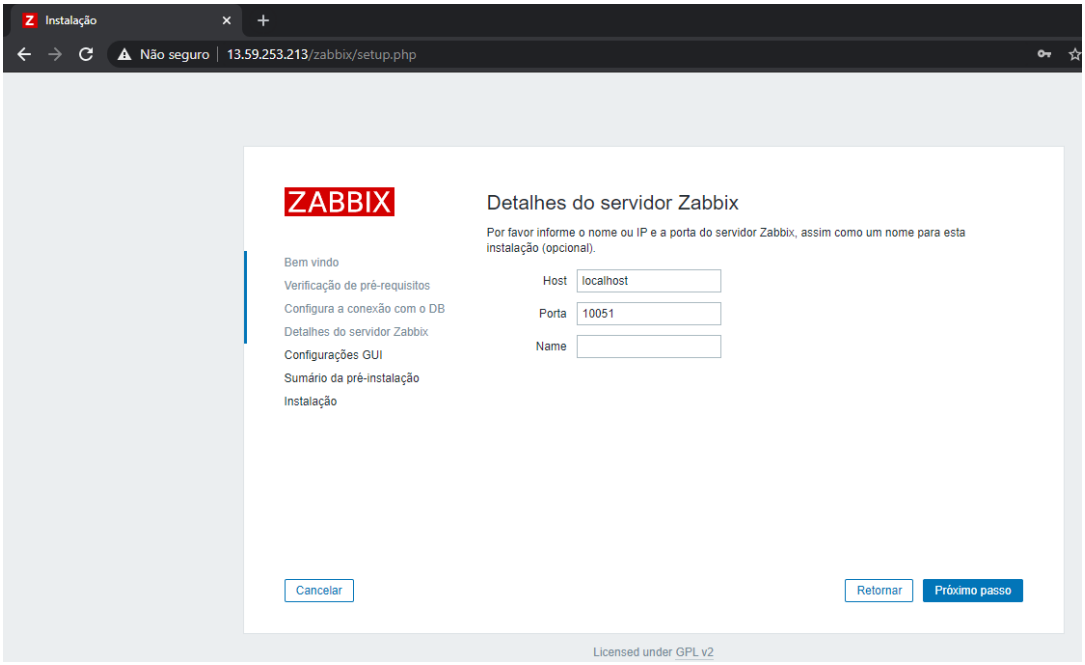
Figura 48 - Configuração do SGBD



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Na tela da Figura 49, é mostrado os detalhes do servidor Zabbix como, *host*, porta e *name*. Os preenchimentos dos campos são opcionais, clicar em próximo passo caso esteja conforme desejar.

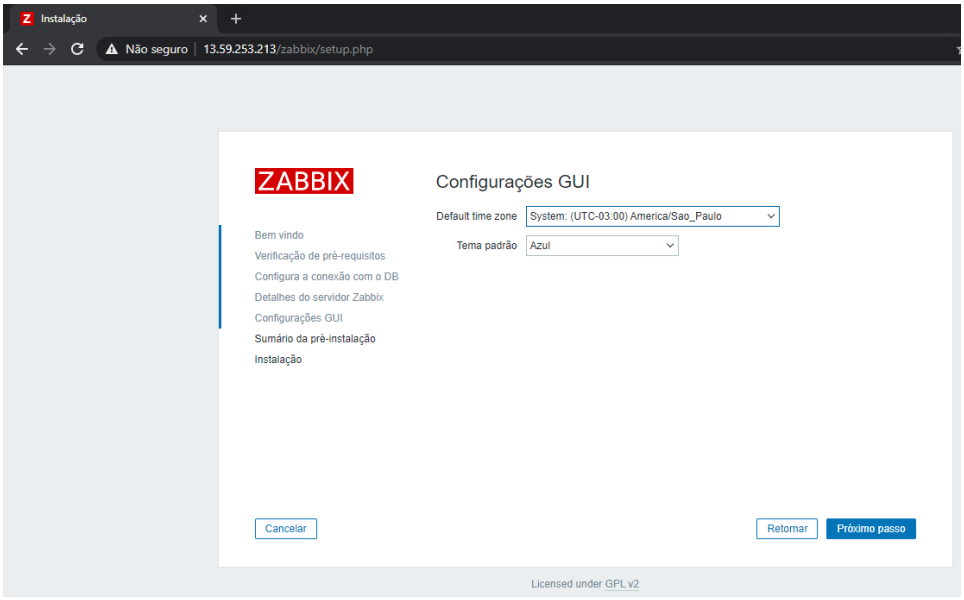
Figura 49 - Detalhes do servidor Zabbix



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Na próxima tela conforme a Figura 50, é exibido as configurações da *interface* gráfica do usuário e o fuso horário, pode-se alterar o tema como preferir, após definidos clicar em próximo passo.

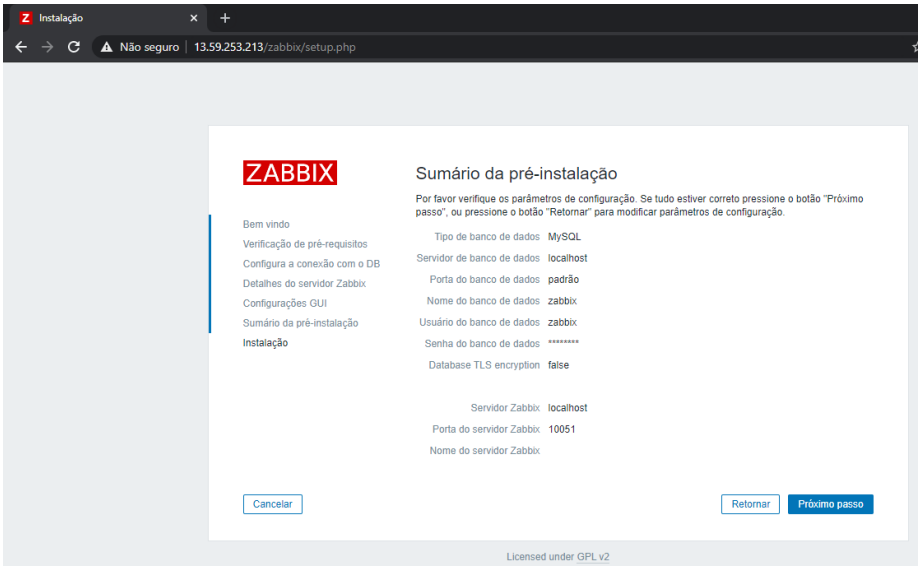
Figura 50 - Configuração da *interface*



Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

Por fim é exibido o sumário de pré-instalação para confirmar os parâmetros de configuração inseridos anteriormente, estando tudo correto, clicar em próximo passo, e a instalação é finalizada.

Figura 51 - Sumário da pré-instalação

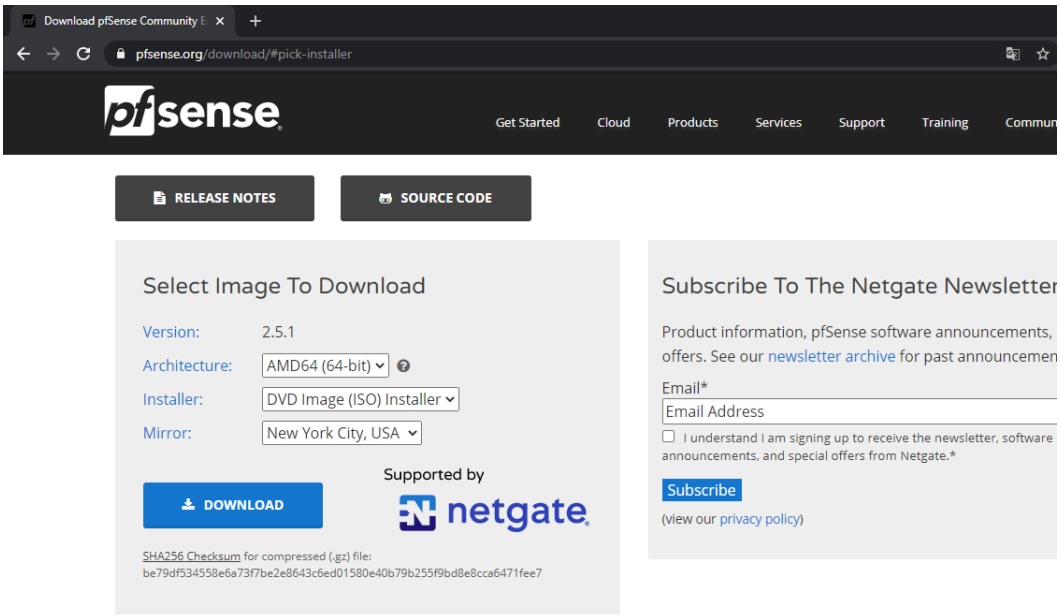


Fonte: Tela de captura do Zabbix com conteúdo desenvolvido pela autora deste trabalho

APENDICE B – INSTALAÇÃO PFSENSE

Para instalar o pfSense é necessário realizar o *download* da imagem no site oficial https://www.pfsense.org/, onde é necessário selecionar a arquitetura e a opção de instalador. Caso a instalação ocorra em ambiente virtual é recomendado selecionar a opção de instalador '*DVD Image (ISO) Installer*' conforme a Figura 52.

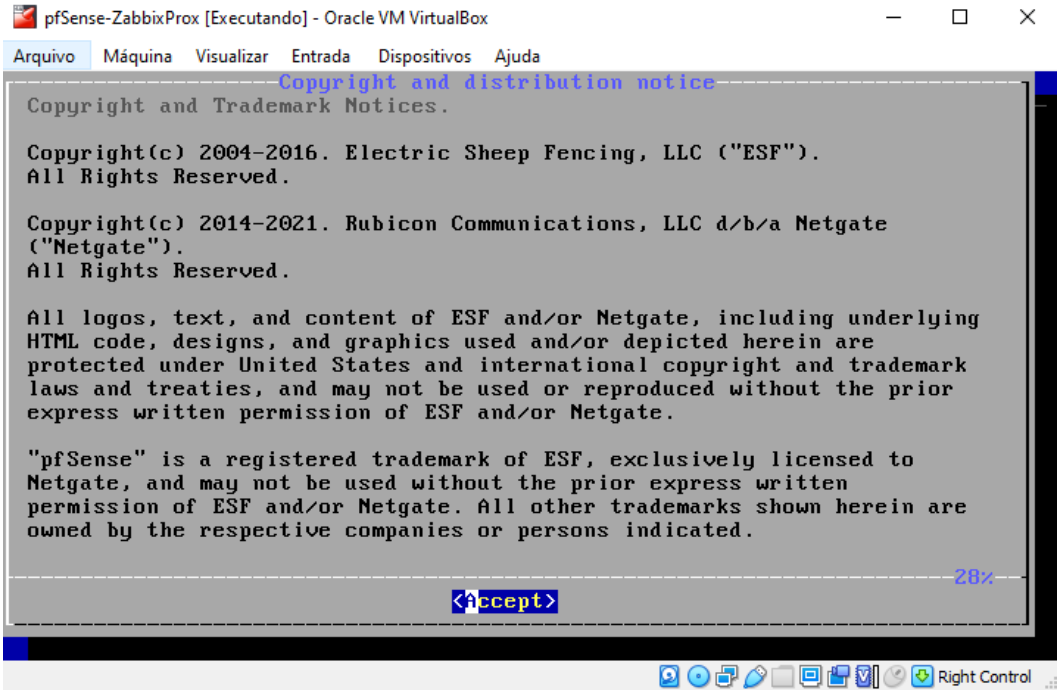
Figura 52 - página web oficial do pfSense



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Primeiro passo – Ao iniciar a máquina com a imagem, deve-se aceitar os termos do *hardware*, conforme a Figura 53.

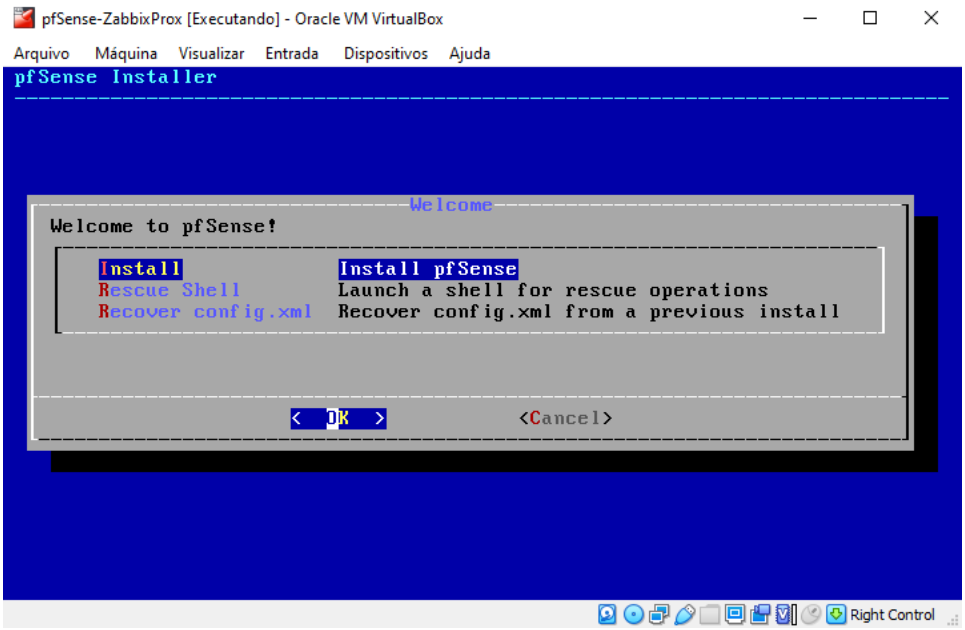
Figura 53 - Termos do serviço



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Segunda passo – Após aceitar os termos selecione a opção "*Install pfSense"* e clique em 'OK'

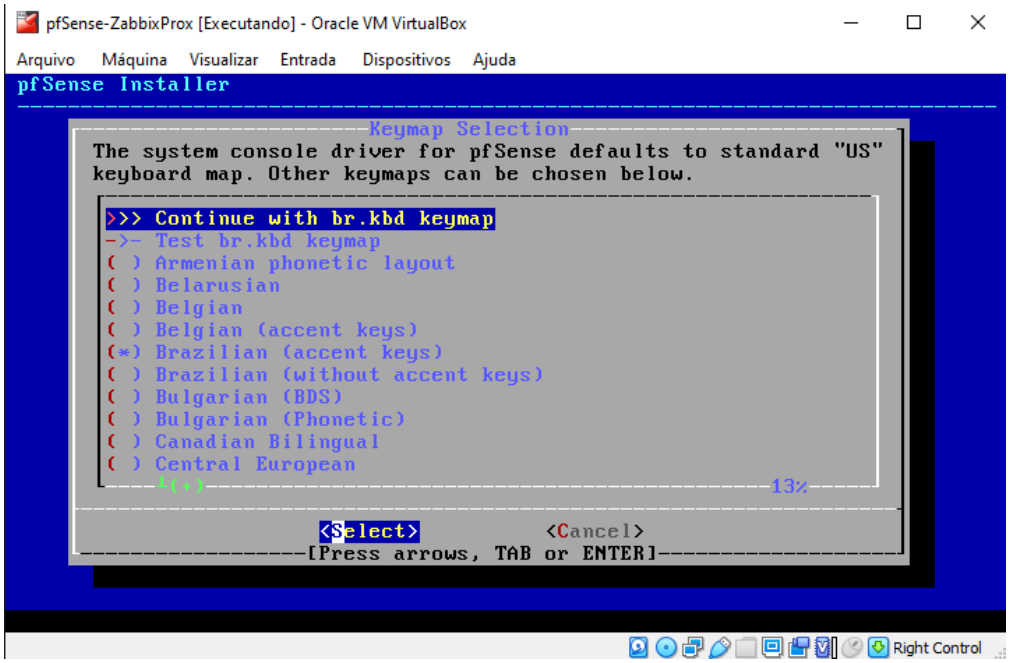
Figura 54 - Menu de instalação



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Terceiro passo – Selecione a linguagem do teclado, no caso de teclado brasileiro, selecione a opção *"Brazilian (accent keys)"* e posteriormente clique em "*continue with br.kbd keymap"*.

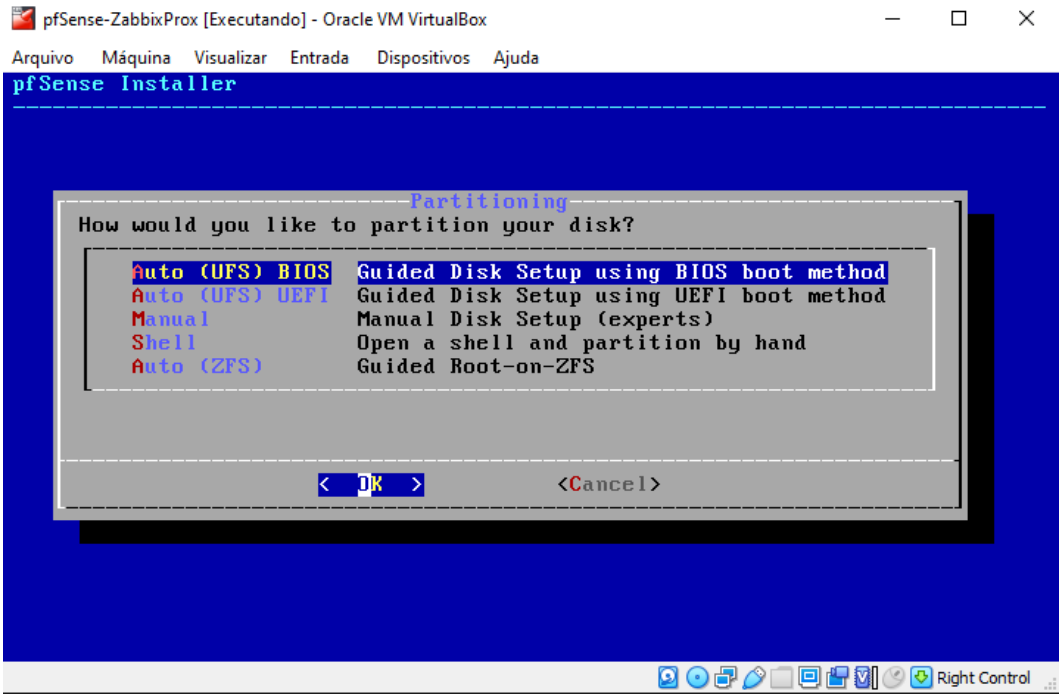
Figura 55 - Configuração do teclado



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Quarto passo – Selecione o tipo de partição deseja, o escolhido para desenvolvimento deste trabalho foi *Auto (UFS) BIOS.*

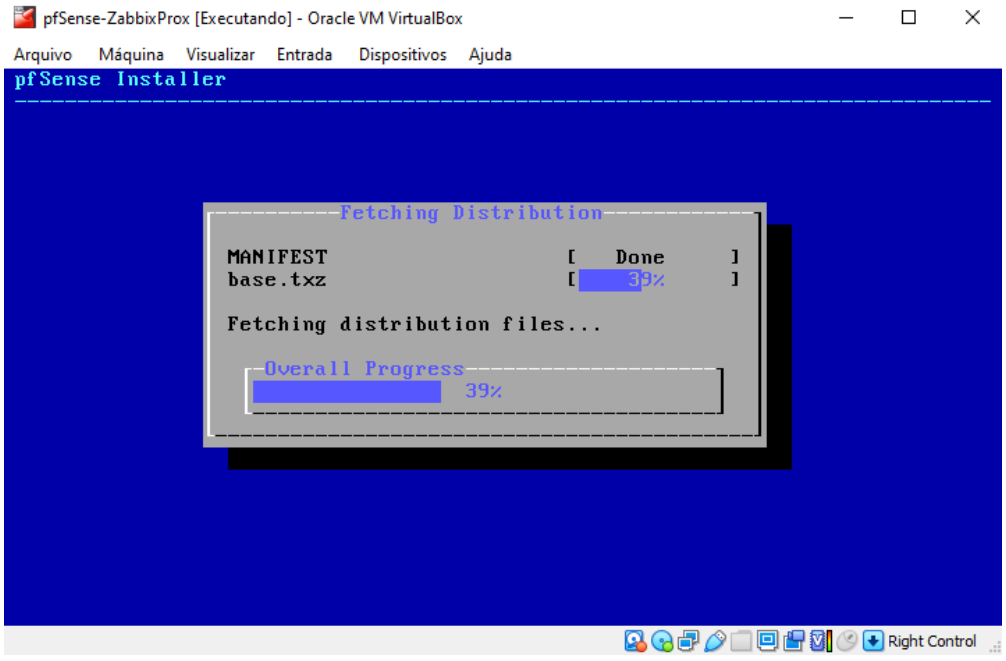
Figura 56 - Tipos de partição



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

A instalação do sistema irá prosseguir, conforme a Figura 57.

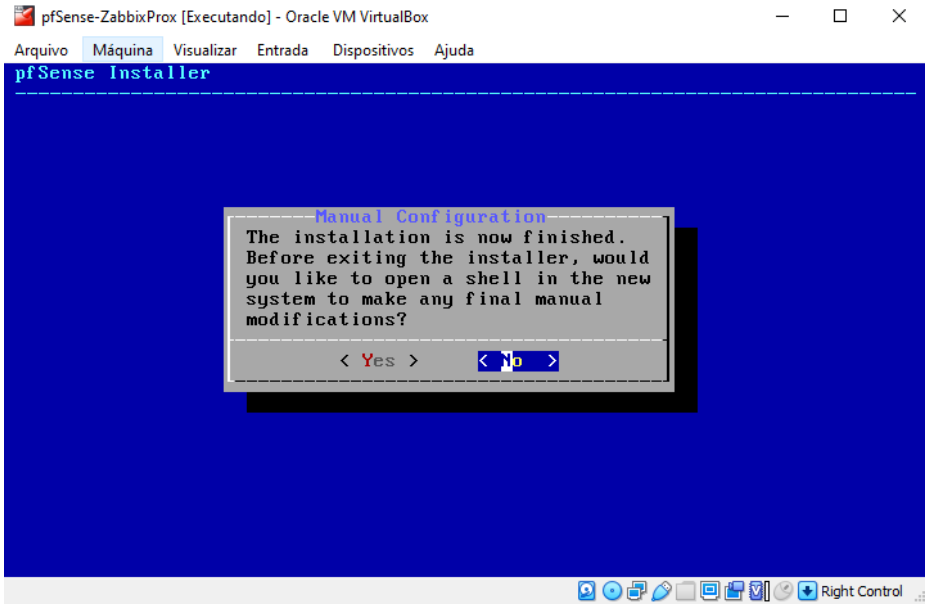
Figura 57 - Realizando instalação



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Quinto passo – Após a instalação é exibido a mensagem conforme a Figura 58, perguntando se deseja realizar alguma modificação, selecione "*No*" e logo depois *reboot* para reiniciar o sistema e concluir a instalação.

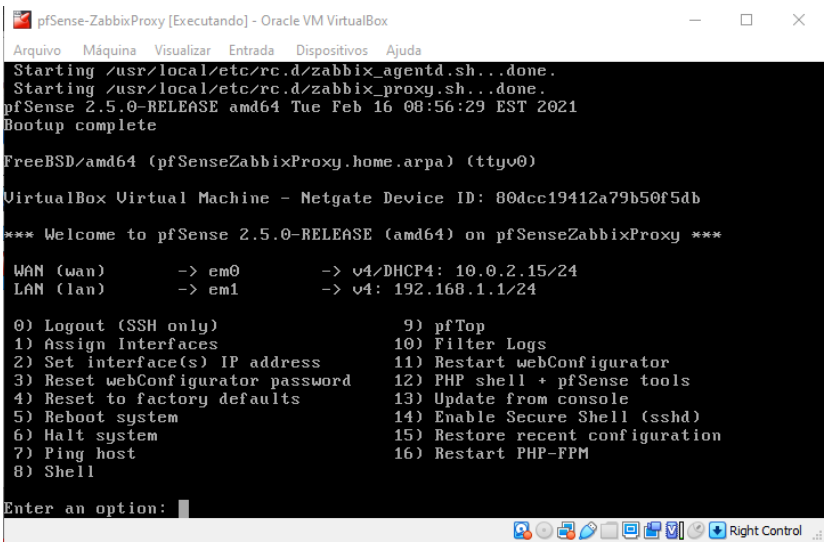
Figura 58 - Reiniciar sistema



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

Após reiniciar a instalação será concluída e iniciar o sistema mostrando o IP do mesmo, para acessar a *interface web* e realizar as configurações.

Figura 59 - tela inicial do pfSense no *prompt* de comando



Fonte: Tela de captura do pfSense com conteúdo desenvolvido pela autora deste trabalho

APENDICE C – INSTALAÇÃO GRAFANA

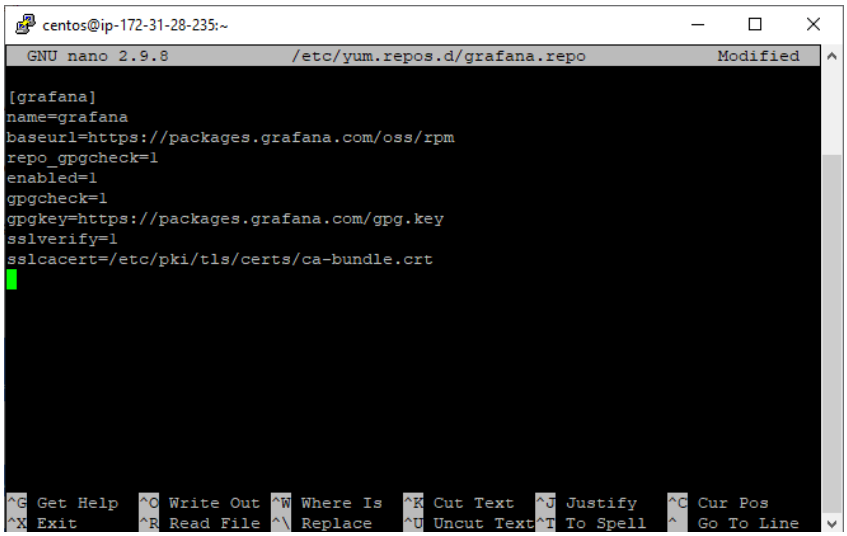
Primeiro passo – criação repositório do Grafana

Foi criado um novo arquivo no repositório do yum, para realizar a instalação.

```
sudo nano /etc/yum.repos.d/grafana.repo
```

No arquivo grafana.repo foi inserida as informações constante na Figura 60.

Figura 60 - Grafana.repo



Fonte: Tela de captura do Putty com conteúdo desenvolvido pela autora deste trabalho

Após a criação do repositório, foi realizado o comando repolist para atualizar a lista de repositório.

```
yum repolist
```

Segundo passo – Instalação Grafana

Para instalar o Grafana foi digitado o comando

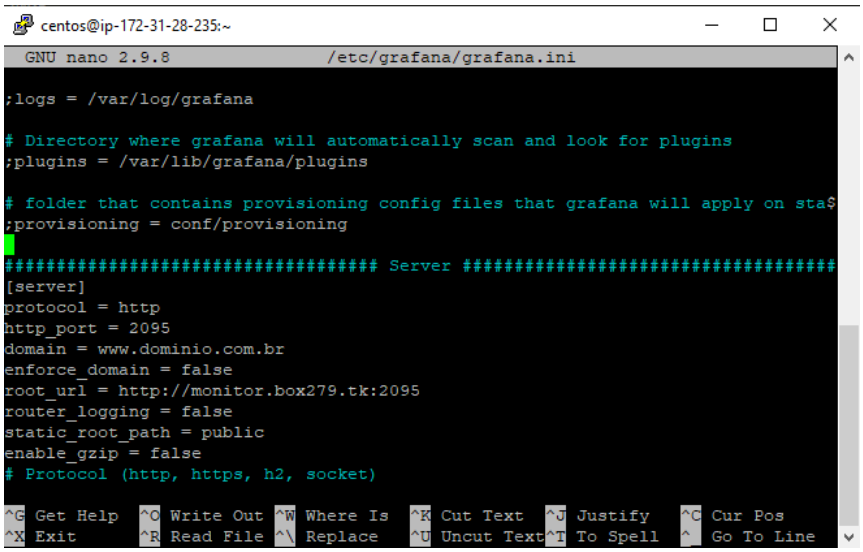
```
yum install grafana -y
```

Foi realizado a configuração do Grafana a partir do arquivo grafana.ini utilizando o comando

```
nano /etc/grafana/grafana.ini
```

No editor de texto Nano foi inserido as configurações de protocolo, porta e pagina de acesso, conforme a Figura 61.

Figura 61 - Configuração de comunicação.



Fonte: Tela de captura do Putty com conteúdo desenvolvido pela autora deste trabalho

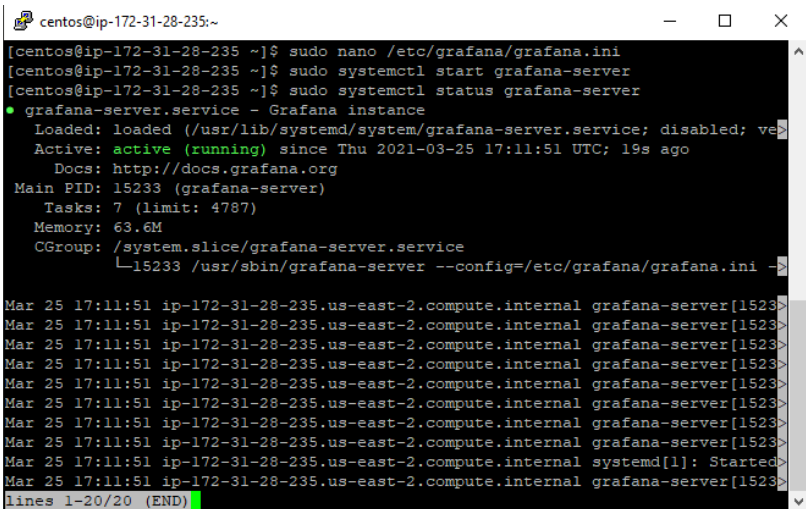
Posteriormente foi realizado a inicialização do Grafana e a verificação do seu status, utilizando os comandos abaixo e obtendo o resultado da Figura 62.

```
sudo systemctl start grafana-server
```

```
sudo systemctl status grafana-server
```

```
sudo systemctl enable grafana-server
```

Figura 62 - Status Grafana



Fonte: Tela de captura do Putty com conteúdo desenvolvido pela autora deste trabalho

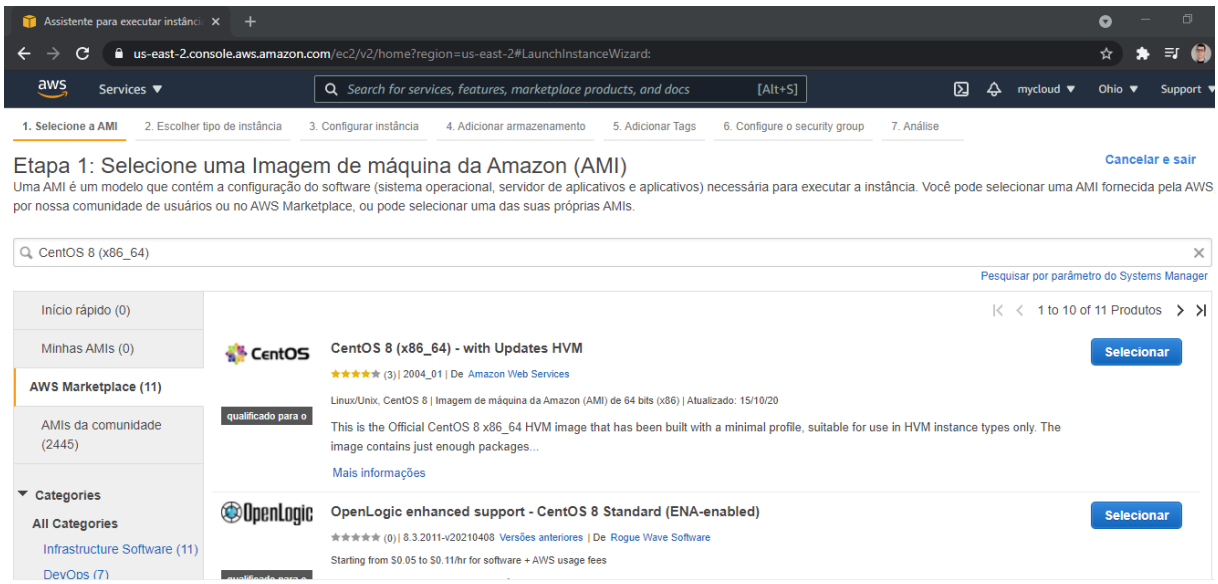
APENDICE D – CRIAÇÃO INSTÂNIA EC2

Para criar uma instância EC2 foi realizado cadastro na página web da Amazon AWS com endereço https://aws.amazon.com/pt/, feito o cadastro, no menu Serviços da AWS foi seleciona a opção EC2, feito isso o site redirecionou para outra a página, em qual foi selecionar opção Instância e posteriormente *Launch Instance*, feito isso realizou-se seguir as seguintes etapas para concluir a criação da instância.

Primeira etapa – Escolha da imagem para a instância.

Nesta etapa ocorreu a escolha da imagem a qual a instância utilizou, CentOS 8 (x86_64), conforme a Figura 63.

Figura 63 - Seleção imagem AWS

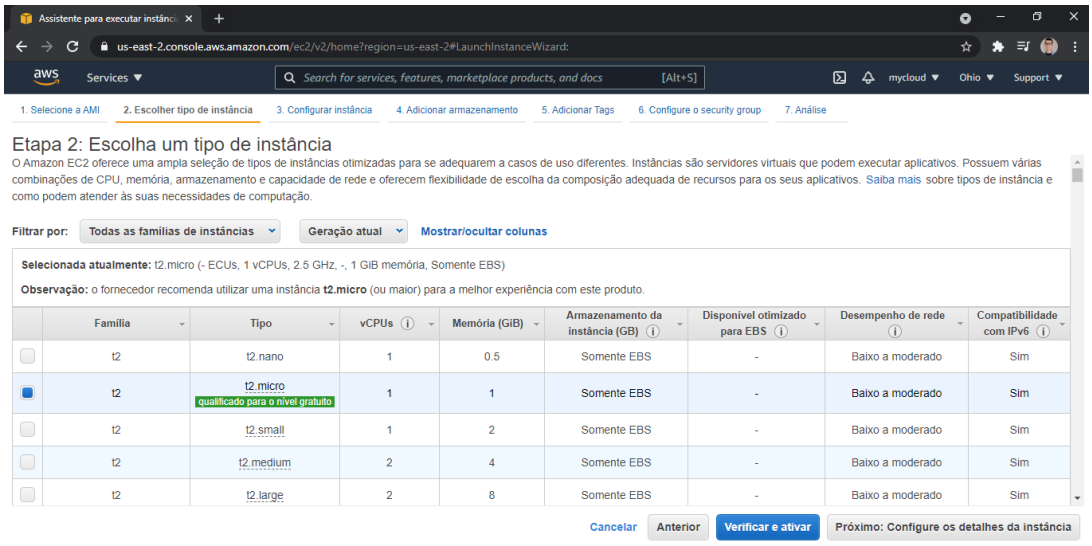


Fonte: Tela de captura do AWS com conteúdo desenvolvido pela autora deste trabalho

Segunda etapa – Escolha instância.

A Amazon AWS disponibiliza diversos tipos de instância com na Figura 64 conforme já discutido na tópico 5.1, nesta etapa foi selecionado t2.micro.

Figura 64 - Escolha tipo de instância



Fonte: Tela de captura do AWS com conteúdo desenvolvido pela autora deste trabalho

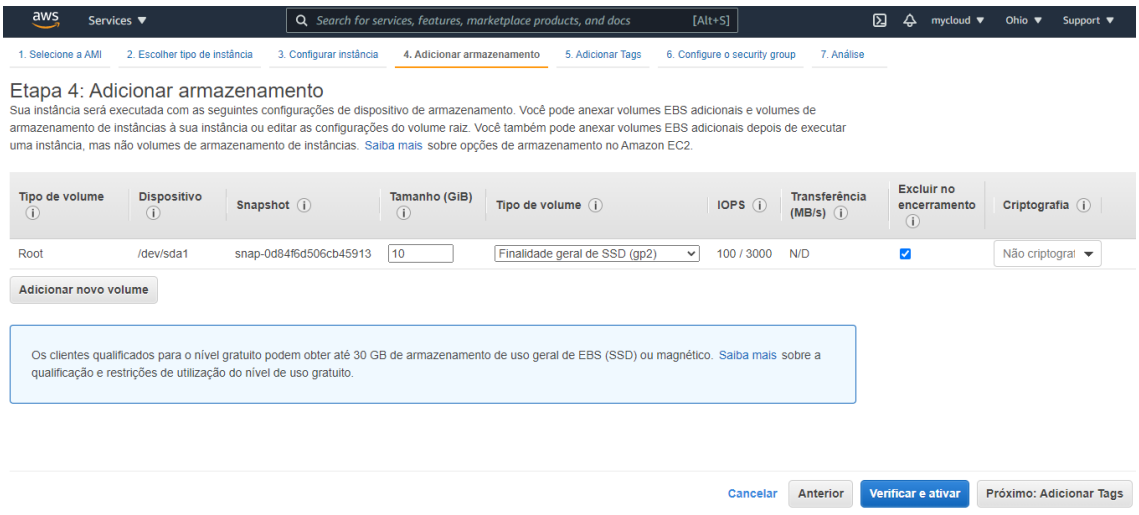
Terceira etapa – Configuração de detalhes.

Nesta etapa é possível realizar alteração de detalhes da configuração da instância, porém foi utilizado a configuração padrão.

Quarta etapa – Adicionar armazenamento.

Na quarta etapa foi selecionado Excluir no encerramento, para quando encerrado a instância seus dados sejam apagados.

Figura 65 - Configurações de armazenamento



Fonte: Tela de captura do AWS com conteúdo desenvolvido pela autora deste trabalho

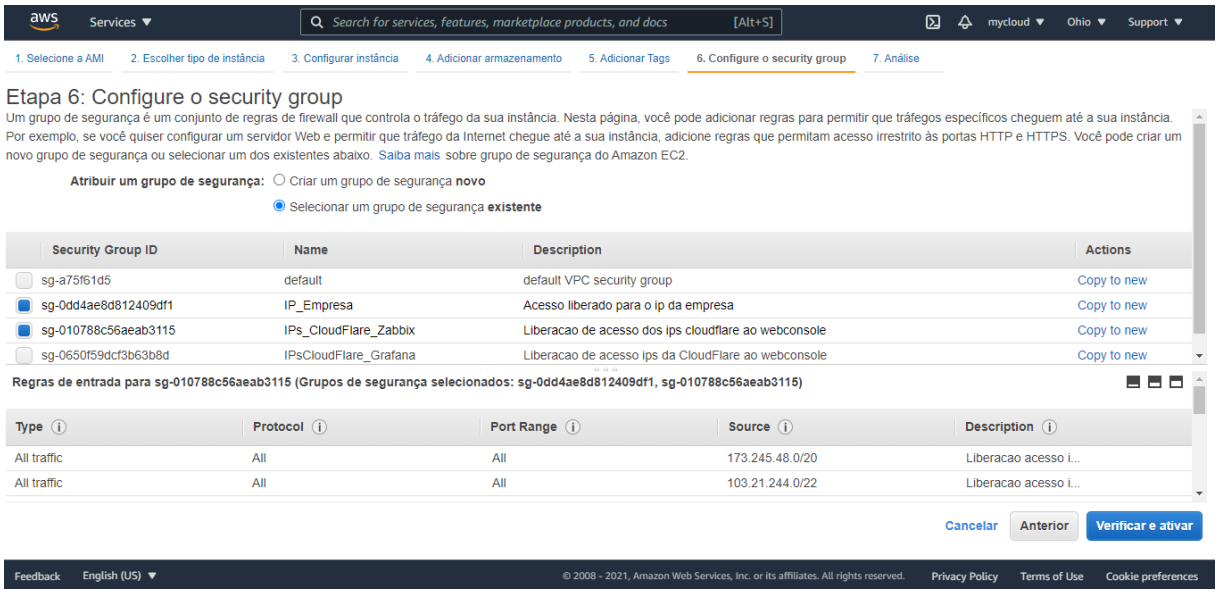
Quinta etapa – Adicionar Tags

Nenhuma tag foi utilizada, apenas avançado para o próximo passo.

Sexta etapa – Configuração do *security group*

Em atribuir um grupo de segurança foi escolhido selecionar um grupo de segurança existente, e selecionados posteriormente os grupos IP_Empresa e IPs_CloudFlare.

Figura 66 - Seleção dos grupos de segurança

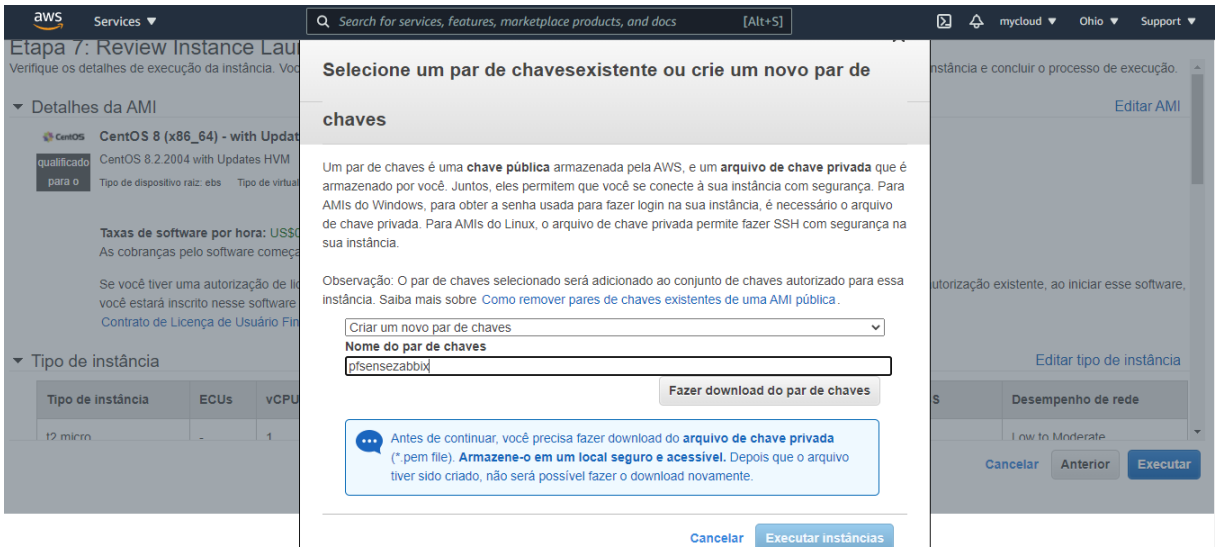


Fonte: Tela de captura do AWS com conteúdo desenvolvido pela autora deste trabalho

Sétima etapa – Review e inicialização da instância

Antes de concluir a criação e iniciar a instância foi criado um par de chaves e atribuído a instância, o nome escolhido foi pfsensezabbix, está chave é usada para realizar acesso via *Secure Shell* (SSH).

Figura 67 - Seleção par de chaves



Fonte: Tela de captura do AWS com conteúdo desenvolvido pela autora deste trabalho

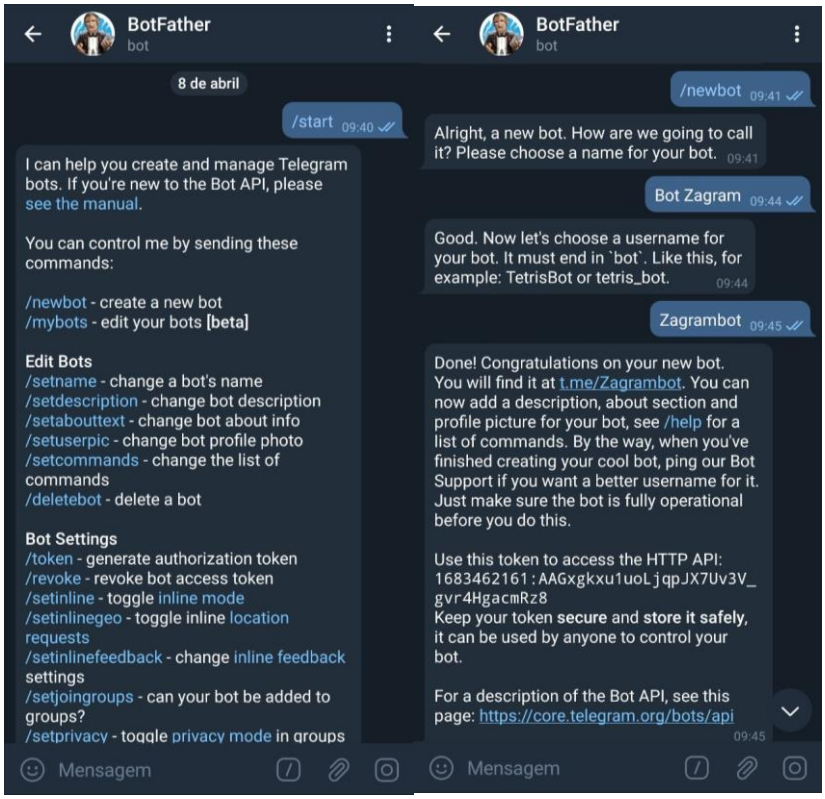
APENDICE E – CRIAÇÃO BOT NO TELEGRAM

Primeira etapa - Criando *bot* no Telegram

Na plataforma do Telegram deve-se procurar por @BotFather, em seguida iniciar a conversa com comando /start, conforme a Figura 68 e posteriormente digitar /newbot para iniciar a criação do *bot.*

Após o comando /newbot deve-se dá um nome ao *bot.* Exemplo “Bot Zagram”, logo após é necessário digitar nome de usuário para o *bot*, o mesmo deve terminar com *bot.* Exemplo “Zagrambot”. Por fim, será enviado o TOKEN conforme a Figura, que é necessário para realizar as configurações.

Figura 68 - Comandos criação bot



Fonte: Tela de captura do Telegram com conteúdo desenvolvido pela autora deste trabalho

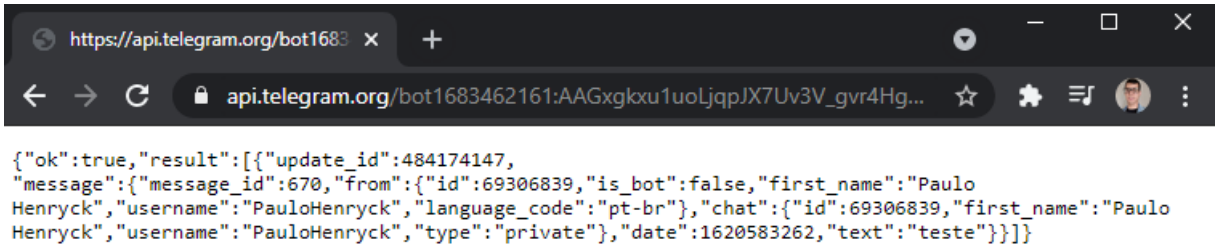
Logo após a criação do *bot*, inicie uma conversa com o mesmo com comando /start e envie uma mensagem por exemplo teste como demonstrado na Figura 69, feito isso abra o navegador e digite o endereço: https://api.telegram.org/botTOKEN/getUpdates para descobrir a identificação do usuário que enviou a mensagem.

Figura 69 - Iniciar conversa com bot

Fonte: Tela de captura do Telegram com conteúdo desenvolvido pela autora deste trabalho

O endereço digitado apresenta as informações sobre a mensagem enviada e o usuário, bem como sua identificação, conforme a Figura 70.

Figura 70 - Obtendo informações sobre o *bot* e usuário



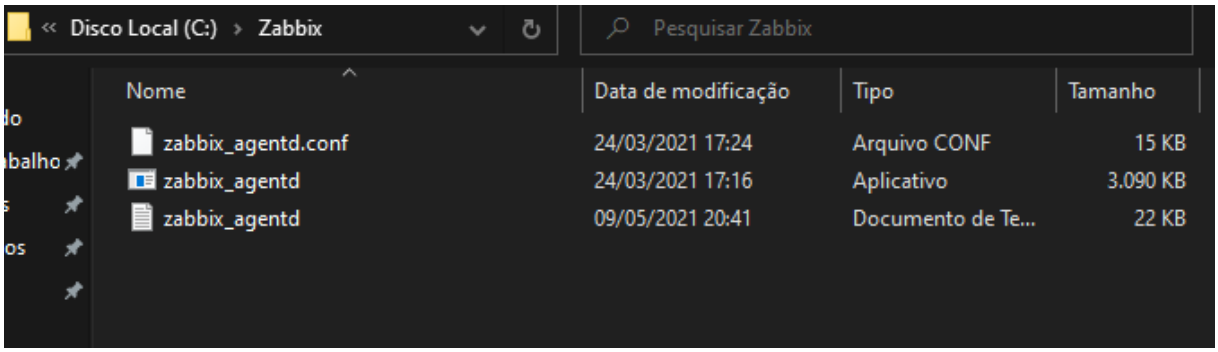
Fonte: Tela de captura do Telegram com conteúdo desenvolvido pela autora deste trabalho

Concluído a criação do *bot* e obtido as informações do usuário a qual deseja-se enviar as mensagens, agora deve-se configurar a mídia, meio pelo qual o Zabbix usa para enviar notificações.

APÊNDICE F – INSTALAÇÃO ZABBIX AGENTE WINDOWS

Para instalação do Zabbix Agent no Windows foi necessário realizar o download do pacote no site oficial do Zabbix, foi realizado a extração do pacote comprimido e criado uma pasta no disco local C: com nome Zabbix e adicionado os arquivos extraídos de acordo com a Figura.

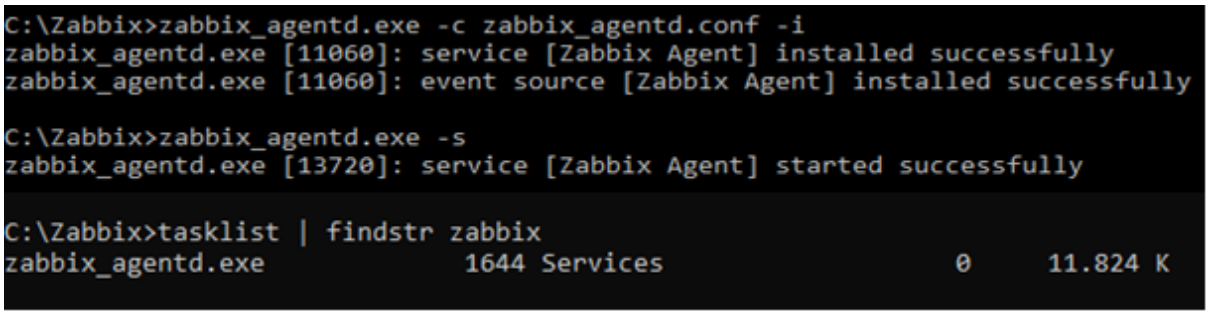
Figura 71 - Pacotes Zabbix Agent



Fonte: Tela de captura do Windows com conteúdo desenvolvido pela autora deste trabalho

Para realizar a instalação do agente Zabbix no Windows foi utilizado o prompt de comando e a partir dele acessado a pasta Zabbix criada no disco local C: e digitado o seguinte o comando `zabbix_agentd.exe -c zabbix_agent.conf -i, zabbix_agent.exe -s` para iniciar o serviço, e `task | findstr zabbix` verificar se o serviço está em execução, conforme a Figura 72.

Figura 72 - instalação pacotes Zabbix Agent



Fonte: Tela de captura do Windows com conteúdo desenvolvido pela autora deste trabalho

Para configuração do agente foi realizado algumas modificações no arquivo zabbix_agentd.conf nos campos como número IP do servidor e o nome do seu nome do *host*.

```
Server=192.168.1.1

Hostname=Windows10
```

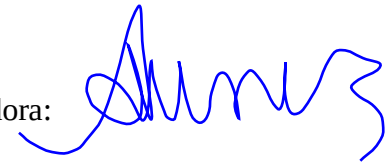
APÊNDICE G – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA

O estudante Paulo Henryck Martins Silva do Curso de Engenharia de Computação, matrícula 20151003304833, telefone: (62) 98247-7146, e-mail paulohenryck_@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 Zabbix, 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, 19 de Junho de 2021.

Assinatura do autor: 

Nome completo do autor: Paulo Henryck Martins Silva

Assinatura da professora-orientadora: 

Nome completo da professora-orientadora: Angélica da Silva Nunes