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

GERÊNCIA DE REDES COM ZABBIX EM AMBIENTE NUVEM

PEDRO HENRIQUE RIBEIRO DANTAS

GOIÂNIA
2024

PEDRO HENRIQUE RIBEIRO DANTAS

GERÊNCIA DE REDES COM ZABBIX EM AMBIENTE NUVEM

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como requisito para obtenção do título de Bacharel em Engenharia de Computação.

Orientadora:
Profª. Ma. Angélica da Silva Nunes

Banca Examinadora:
Prof. Me. Rafael Leal Martins
Prof. Me. Wilmar Oliveria de Queiroz

GOIÂNIA
2024

PEDRO HENRIQUE RIBEIRO DANTAS

GERÊNCIA DE REDES COM ZABBIX EM AMBIENTE NUVEM

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como requisito para obtenção do título de Bacharel em Engenharia de Computação, aprovado em_____/_____/__.

_____

Orientadora: Profª. Ma. Angélica da Silva Nunes

_____

Profº Me. Rafael Leal Martins

_____

Prof. Me. Wilmar Oliveria de Queiroz

Goiânia
2024

RESUMO

Este trabalho visa aprender e implementar uma ferramenta de administração de redes que capacite os administradores a obterem componentes essenciais para manter a saúde da rede, minimizando adversidades e promovendo um alto nível de performance e desempenho. O Zabbix foi escolhido como a ferramenta central deste estudo devido à sua ampla gama de recursos que simplificam o controle da infraestrutura de rede. O trabalho aborda não apenas a instalação e configuração do Zabbix em uma infraestrutura em nuvem, mas também a integração com o Telegram para envio de notificações aos usuários. Além disso, será utilizado o Grafana para visualização avançada das métricas monitoradas. A infraestrutura em nuvem selecionada inclui cinco *hosts* que serão monitorados para avaliar o desempenho e a disponibilidade dos recursos de rede de maneira eficaz e detalhada.

Palavras-Chave: Administração de Redes, Zabbix, Computação em Nuvem, Grafana, Telegram.

ABSTRACT

This work aims to learn and implement a network administration tool that enables administrators to obtain essential components to maintain network health, minimizing adversities and promoting a high level of performance and performance. Zabbix was chosen as the central tool for this study due to its wide range of features that simplify control of network infrastructure. The work covers not only the installation and configuration of Zabbix in a cloud infrastructure, but also the integration with Telegram to send notifications to users. In addition, Grafana will be used for advanced visualization of monitored metrics. Specific cloud infrastructure includes hosts that will be monitored to assess the performance and availability of network resources in an effective and specific manner.

Keywords: Network Administration, Zabbix, Cloud Computing, Grafana, Telegram.

LISTA DE FIGURAS

Figura 1: Comunicação da arquitetura da rede 18
Figura 2: Visão geral do gerenciamento de rede 20
Figura 3: Processo de comandos SNMP 22
Figura 4: Principais provedores de nuvem 26
Figura 5: Responsabilidades para cada tipo de serviço 28
Figura 6: Arquitetura do Zabbix 30
Figura 7: Autenticação no *site* da ZeroTier 34
Figura 8: Painel de rede ZeroTier 35
Figura 9: Topologia do ambiente 36
Figura 10: Especificações da VM 37
Figura 11: Nome do domínio 38
Figura 12: Valores referentes ao tempo que o domínio fica ativado 38
Figura 13: Painel de status do domínio 39
Figura 14:Informações do titular do domínio 39
Figura 15:Contatos do responsável pelo domínio 40
Figura 16:Painel de informação e configuração do DNS 40
Figura 17:Painel de configuração DNS no Registo.br 41
Figura 18:Grupos de segurança da instância AWS 42
Figura 19: Opção para adicionar a rede 43
Figura 20: Tela de adição da rede 43
Figura 21:Painel de *hosts* adicionados a rede 44
Figura 22: Processo de criação do bot 45
Figura 23: Continuação do processo de criação do bot 46
Figura 24: Processo de escolha do nome do bot 47
Figura 25: Token de acesso a API 48
Figura 26: Mensagem teste enviada para testar o funcionamento do bot 49
Figura 27: Informações recebidas da mensagem de teste enviada ao bot 50
Figura 28: Grupo criado com o bot 51
Figura 29: Mensagem teste enviada para obter a ID do grupo de alertas 52

Figura 30: Comando para obter a ID do grupo com o bot 52
Figura 31: Resultado do comando executado 53
Figura 32:Aba de usuários 53
Figura 33:Janela de configuração de mídia do usuário 54
Figura 34: Menu de funções do Zabbix 55
Figura 35: Tela de configuração do Telegram no Zabbix 55
Figura 36:Menu de funções do Zabbix 56
Figura 37: *Triggers* adicionadas para os *hosts* monitorados 57
Figura 38:Tela de configuração de mensagem de incidente 58
Figura 39: Tela de configuração de recuperação de incidente 59
Figura 40: Tela de configuração de atualização de incidente 60
Figura 41: Comando de instalação do *plugin* no Zabbix *Server* 61
Figura 42: Tela de configuração do *plugin* 61
Figura 43: Página inicial da conta Cloudflare 62
Figura 44:Configuração do DNS no Cloudflare 62
Figura 45:Tela de instalação do Zabbix Agent 63
Figura 46: Tela de configuração do *host* SRV-GYN 63
Figura 47: Tela de configuração do *host* SrvPorangatu 64
Figura 48: Tela de configuração do *host* SrvPelotas 64
Figura 49: Tela de configuração do *host* G 65
Figura 50: Tela de configuração do *host* Y 65
Figura 51: *Dashboard* do *host* Y 66
Figura 52: *Dashboard* do *host* G 66
Figura 53: *Dashboard* do *host* SRV-GYN 66
Figura 54: *Dashboard* do *host* SrvPorangatu 67
Figura 55: *Dashboard* do *host* SrvPelotas 67
Figura 56: Gráfico filtrado por período do *host* SRV-GYN 68
Figura 57: Mapa central da rede 69
Figura 58: Mapa dos *hosts* localizados em Goiânia 70
Figura 59: Mapa do *host* localizado em Pelotas 71
Figura 60: Mapa do *host* localizado em Porangatu 72

Figura 61: Painel de monitoramento do *host* Y 73
Figura 62: Painel de monitoramento do *host* SrvPelotas 73
Figura 63: Painel de monitoramento do *host* SrvPorangatu 73
Figura 64: Painel de monitoramento do *host* SRV-GYN 74
Figura 65: Painel de monitoramento do *host* G 74
Figura 66: Alertas enviados ao Telegram pelo Zabbix 75

LISTA DE QUADROS E TABELAS

Quadro 1: Pré-requisitos Zabbix..... 24

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface* – Interface de Programação de Aplicação
AWS	*Amazon Web Services*
DDoS	*Distributed Denial of Service* – Ataque Distribuído de Negação de Serviço
DNS	*Domain Name System* – Sistema de Nome de Domínio
EC2	*Elastic Compute Cloud*
GCP	*Google Cloud Plataform*
IaaS	*Infrastructure as a Service* – Infraestrutura como Serviço
IP	*Internet Protocol* – Protocolo de Internet
ISO	*International Organization for Standardization* – Organização Internacional para Padronização
ISO/IEC	*International Organization for Standardization/International Electrotechnical Commission* – Organização Internacional para Padronização/Comissão Internacional de Eletrotécnica
HTTP	*Hypertext Transfer Protocol* – Protocolo de Transferência de Hipertexto
HTTPS	*HyperText Transfer Protocol over SSL* - Protocolo de Transferência de Hipertexto sobre SSL
MIB	*Management Information Base* – Base de Informações de Gerenciamento
NCC	*Network Control Center* - Centro de Controle de Rede
NIC.BR	Núcleo de Informação e Coordenação do Ponto BR
NMA	*Network Management Application* - Aplicação de Gerenciamento de Rede
NME	*Network Management Entity* – Entidade de Gerenciamento de Rede
PaaS	*Plataform as a Service* – Plataforma como Serviço
PDU	*Protocol Data Unit* – Unidade de Dados de Protocolo
SaaS	*Software as a Service* – *Software* como Serviço
SLA	*Sevice Level Agreement*, Acordo de Nível de Serviço
SMI	*Management Information Structure* – Estrutura de Informações de Gerenciamento

SNMP	*Simple Network Management Protocol* – Protocolo Simples de Gerenciamento de Rede
SSL	*Secure Sockets Layer* – Camada de Soquete Seguro
RAM	*Random Acess Memory* – Memória de Acesso Randômico
TCP	*Transmission Control Protocol* – Protocolo de Controle da Transmissão
TCP/IP	*Transmission Control Protocol/Internet Protocol* – Protocolo de Controle da Transmissão/Protocolo de *Internet*
TI	*Technology of Information*, Tecnologia da Informação
UDP	*User Datagram Protocol* – Protocolo de Datagrama de Usuário
URL	*Uniform Resource Locator* - Localizador Uniforme de Recursos
VM	*Virtual Machines – M*áquinas Virtuais
VCPU	*Virtual Central Process Unit* - Unidade de Processo Central Virtual
VPN	*Virtual Private Network* - Rede Privada Virtual

SUMÁRIO

1 INTRODUÇÃO 13
1.1 Objetivo 14
1.1.1 Objetivo Geral 14
1.1.2 Objetivos Específicos 14
1.2 Metodologia 14
1.3 Estrutura da Monografia 15

2 GERENCIAMENTO DE REDES 16
2.1 Conceito 16
2.2 Modelo de gerenciamento 17
2.3 Arquitetura de gerenciamento 18

3 FERRAMENTAS UTILIZADAS 23
3.1 Zabbix 23
3.1.1 Apresentação da ferramenta 23
3.1.2 Pré-requisitos 24
3.1.3 Instalação na nuvem 25
3.1.4 Arquitetura do Zabbix 29
3.1.5 Nomenclaturas 31
3.1.6 Coleta de dados 33
3.2 Grafana 33
3.3 ZeroTier 34

4 DESCRIÇÃO DO EXPERIMENTO 36
4.1 Topologia do ambiente 36
4.2 Descrição do ambiente de implementação 37
4.2.1 Servidor nuvem 37
4.2.2 Domínio *Web* 37
4.2.3 Serviço DNS 41
4.3 Configuração do ambiente 42

5 TESTES DO AMBIENTE E RESULTADOS 68
5.1.1 Zabbix 68
5.1.2 Grafana 72

5.1.3 Telegram 74
5.1.4 Análise dos resultados 76

6 CONSIDERAÇÕES FINAIS 77
6.1 Sugestões de trabalhos futuros 78

REFERÊNCIAS 79

1 INTRODUÇÃO

As redes de computadores eram consideradas apenas ferramentas de pesquisa por volta das décadas de 1970 e 1980 e, por conseguinte, o termo “gerenciamento de rede” ainda não era conhecido. Se houvesse algum transtorno na rede, eram efetuadas tentativas de comandos de teste como *ping* e similares, para localizar a fonte do problema. Conforme a *Internet* foi evoluindo, mostrou-se a carência de ter recursos de administração de rede que contribuem com o gerente a mantê-la com uma boa taxa de serviço (KUROSE; ROSS, 2013).

As redes tornaram-se indispensáveis, e são consideradas uma tarefa de nível crítico, ou seja, não deve apresentar indisponibilidade de seu serviço. Ter em mãos recursos apropriados para a administração de redes, embora que o contexto de rede seja simples, traz grandes benefícios para o gerente na busca e resolução de problemas.

Há muitos sistemas de gerência de rede no mercado, pagos ou gratuitos. O Zabbix, por exemplo, é um *software* criado para profissionais que atuam com gerenciamento de redes, além de ser gratuito, é *open source*, ou seja, o usuário pode configurar a ferramenta conforme suas necessidades.

De acordo uma pesquisa feita pela Inmetrics em parceria com a TGT ISG e Unicamp divulgada em julho de 2023, o Zabbix é o principal *software* utilizado para monitoração e observabilidade dentro das empresas. A pesquisa entrevistou várias organizações brasileiras, e 54% delas são de grande porte, 21% de médio e 25% de pequeno porte e 42% dessas empresas colocaram o Zabbix como ferramenta líder de monitoração (INMETRICS, 2023).

O Zabbix foi desenvolvido em 1998, por Alexei Vladishev, devido a sua insatisfação com os sistemas de monitoramento que estava trabalhando naquele momento. Em 2001 foi lançada sua primeira versão 0.1 alpha, e em 2004 foi lançada a versão definitiva 1.0. Em 2005 foi fundada a empresa Zabbix SAI, a partir disso a ferramenta foi evoluindo e expandido no mercado, atingindo a marca de 800.000 *downloads* em 2012 (LIMA, 2014).

Justifica-se aprofundar este assunto, pelo fato de o *software* ser amplamente utilizado dentro das organizações brasileiras e estudar as vantagens da utilização dessa ferramenta no campo de gerência de rede, assim como a sua união com aplicações de comunicação, objetivando obter o controle dessa infraestrutura, tornando mais ágil a definição e resolução de problemas.

Diante desse cenário, este trabalho objetiva responder a seguinte questão:

Como realizar o gerenciamento de um ambiente em nuvem através do Zabbix?

1.1 Objetivo

Este trabalho tem como objetivo implantar o Zabbix em uma infraestrutura em nuvem, monitorando a *performance*, coletando informações, gerando relatórios e enviando alertas ao administrador.

1.1.1 Objetivo Geral

Implantar o Zabbix em uma máquina virtual da *Amazon Web Services* (AWS) para monitoramento de agentes distribuídos em várias localidades.

1.1.2 Objetivos Específicos

Implantar o Zabbix, buscando atingir os seguintes objetivos:

- Supervisionar o desempenho dos recursos;
- Reunir e arquivar informações do ambiente, visando analisar o histórico das coletas do sistema;
- Obter relatórios, objetivando acompanhar o ambiente;
- Notificar o usuário em caso de indisponibilidade ou dos parâmetros fora da configuração definida.

1.2 Metodologia

Este trabalho, segundo sua natureza, é um resumo de assunto, pois reúne, analisa e discute fundamentos e referências que já foram publicadas acerca do tema. Portanto, a pesquisa é classificada dessa forma, pois sistematiza uma área de conhecimento, indicando sua evolução histórica e estado (WAZLAWICK, 2014).

No que se refere a objetivos é uma pesquisa explicativa, este é um estudo mais completo pois visa averiguar as informações examinadas, buscar suas razões, ou seja,

determinar os fatores a partir dos dados (WAZLAWICK, 2014). Portanto o projeto busca ter o conhecimento do ambiente de infraestrutura.

No que se refere a métodos técnicos é uma pesquisa experimental, pois controla a quantidade e qualidade de variáveis experimentais da infraestrutura. Possibilitando o estudo das razões do evento, dessa forma, permite controlar e avaliar (WAZLAWICK, 2014).

A pesquisa foi realizada em um ambiente de testes na AWS, utilizando o Zabbix para monitoramento, Grafana para visualização de dados, Telegram para envio de alertas e ZeroTier VPN para acesso seguro e comunicação entre *hosts*. O controle rigoroso das variáveis experimentais permitiu um estudo detalhado dos eventos na infraestrutura. Foram avaliadas métricas como utilização de CPU, memória e disco, uso de banda, taxa de disponibilidade de serviços etc. A eficácia dos alertas foi analisada com base na precisão e rapidez de notificação. Este processo buscou obter uma visão detalhada do desempenho da infraestrutura em nuvem, permitindo a identificação e correção proativa de problemas.

1.3 Estrutura da Monografia

Este trabalho segue a seguinte estrutura:

O capítulo 2 apresenta a base teórica, bem como seu conceito, modelo e arquitetura de gerenciamento sobre Gerenciamento de redes.

O capítulo 3 apresenta o Zabbix e aborda seus pré-requisitos, instalação na nuvem, vantagens e desvantagens de uso, arquitetura, nomenclatura e coleta de dados. Além disso apresenta também outras duas ferramentas utilizadas para o desenvolvimento deste trabalho, Grafana e ZeroTier.

O capítulo 4 explica o ambiente utilizado, apresentando sua topologia, a instalação do servidor e configurações necessárias para incrementar a segurança no acesso a interface *web.*

O capítulo 5 mostra os testes realizados no ambiente e os resultados decorrentes destes testes no trabalho.

O capítulo 6 expõe a conclusão deste trabalho, abordando a importância do tema, as dificuldades encontradas, contribuições para a sociedade e sugestão de trabalhos futuros.

2 GERENCIAMENTO DE REDES

Esse capítulo aborda sobre a gerência de redes, bem como seu modelo de gerenciamento e sua arquitetura.

2.1 Conceito

O gerenciamento de redes abrange a implantação e organização de elementos da rede, físicos ou lógicos, para supervisionar, testificar, examinar, ajustar, analisar, ponderar e coordenar os recursos da rede, com a finalidade de atingir na medida do possível um bom nível de desempenho e qualidade de serviço, visando ter uma despesa moderada (KUROSE, 2013).

Com o passar do tempo a gerência de redes passou a ganhar mais importância dentro das empresas. Na década de 80 as redes eram limitadas apenas ao setor de *Technology of Information*, Tecnologia da Informação (TI). Com a evolução e crescimento da rede, esta não se restringe apenas ao setor de TI, abrange todos os setores das organizações podendo o gerenciamento ser feito pela própria empresa ou terceirizado objetivando o seu bom funcionamento.

Para assegurar uma boa taxa de serviço da rede, o administrador pode contar com um grupo de ferramentas integradas que o auxilie no monitoramento e controle. Este grupo é denominado de sistema de gerência de redes, que conta com *interfaces* com informações da rede e um conjunto de comandos.

Os recursos de gerência de redes passaram ser utilizadas e há muitas situações em que o administrador se beneficia do seu uso:

- Detecção de falhas: detecta falhas em *interfaces* da rede, o administrador detecta problemas com antecedência na *interface* e fazendo o ajuste, evitando assim que o usuário entre em contato reportando um problema;
- Monitoramento de hospedeiros: verifica de maneira ativa os hospedeiros da rede se estão ativos e operacionais;
- Monitoração de tráfego: possibilita que os recursos e serviços da rede sejam entregues precisamente a todos os *hosts*;

- Detecção de mudanças rápidas em tabelas de roteamento: o administrador verifica que devido a alterações constantes na tabela de roteamento, um roteador pode apresentar configuração instável ou incorreta;
- Monitoração de *Service Level Agreement* – Acordo de Nível de Serviço (SLA): é um contrato no qual são estabelecidos os padrões de desempenho para o serviço prestado. Alguns desses SLAs inclui: disponibilidade de serviço, latência, vazão e requisitos para a notificação da ocorrência de serviço interrompido;
- Detecção de invasão: alerta o administrador caso surja uma anormalidade na rede, podendo ser um hospedeiro que está navegando por um tráfego suspeito ou um elemento da rede que está apresentando alguma anormalidade (KUROSE, 2013).

2.2 Modelo de gerenciamento

Segundo a *International Organization for Standardization/International Electrotechnical Commission*, Organização Internacional para Padronização/Comissão Internacional de Eletrotécnica (ISO/IEC) 7498 (1984 apud Lopes, 2002), a gerência de redes inclui 5 áreas funcionais que foram definidas pela *International Organization for Standardization,* Organização Internacional para Padronização (ISO):

- Configuração: é nessa área que é feita a configuração inicial da rede, definição da topologia, manutenção e monitoração de mudanças e sua estrutura física e lógica;
- Falhas: nessa área é feita a detecção, diagnóstico e correção de falhas na rede. Quando bem planejada, pode evitar a ocorrência de falhas no futuro;
- Desempenho: nessa área é feito o monitoramento e análise do desempenho da rede para identificar problemas, permitindo o planejamento de capacidade da rede;
- Segurança: protege, monitora e detecta os elementos da rede para que não haja violações na política de segurança;
- Contabilidade: contabiliza e verifica os recursos da rede utilizados pelos usuários, considerando a divisão de contas feita por usuários ou grupos de usuários.

2.3 Arquitetura de gerenciamento

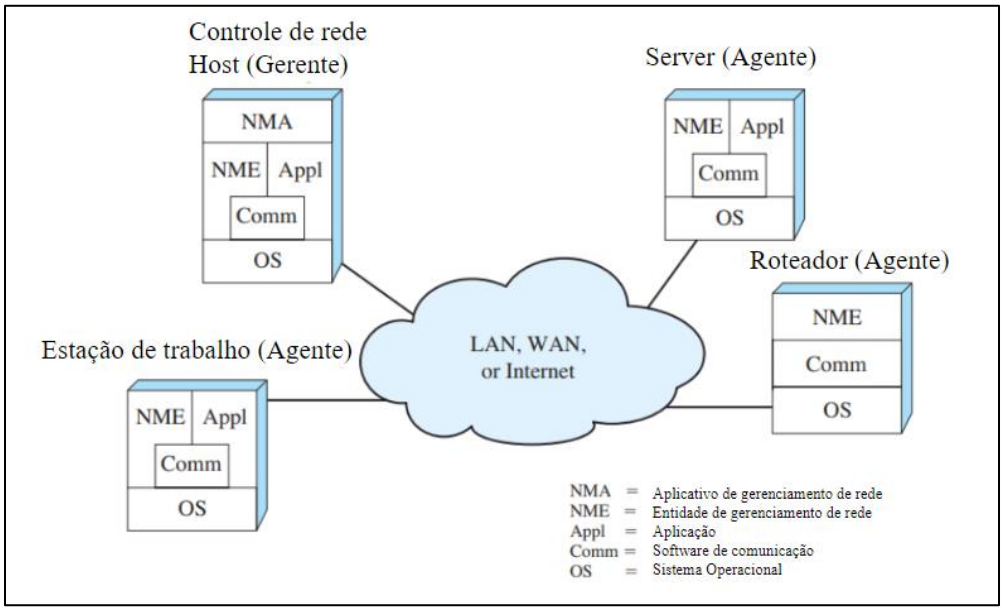
Envolve componentes importantes na rede, dentro dessa arquitetura existe um sistema de ferramentas para monitoramento e administração de rede, que é incorporado nos seguintes sentidos:

- Uma única *interface* de operador com um conjunto de comandos que executa a maioria ou todas as tarefas de gerenciamento de rede;
- Uma quantidade mínima de equipamento separado. Ou seja, a maioria do *hardware* e *software* necessário para o gerenciamento de rede está incorporada no equipamento do usuário existente (STALLINGS, 2005).

O sistema de gerenciamento, consiste em adições incrementais de *hardware* e *software* implementadas entre os componentes de redes existentes. O *software* usado para executar tarefas de gerenciamento está contido nos *hosts* e processadores de comunicações. Ele olha a rede inteira como uma arquitetura unificada com identificações atribuídas a cada ponto, atributos específicos de cada elemento e enlace conhecido no sistema. Dessa forma, ativa os elementos da rede, fornecendo informações de estado e feedback para o *Networking Control Center* - Centro de Controle de Rede (NCC) (STALLINGS, 2005).

A Figura 1 mostra alguns dos componentes da arquitetura.

Figura 1: Comunicação da arquitetura da rede



Fonte: STALLINGS (2005).

Cada nó da rede possui um aglomerado de *softwares* que se dedicam a tarefa de gerenciar a rede, referido na Figura 1 como uma *Networking Management Entity* – Entidade de Gerenciamento de Rede (NME), que tem as seguintes tarefas:

- Coletar estatísticas sobre comunicações e atividades relacionadas à rede;
- Armazenar estatísticas localmente;
- Responder aos comandos do centro de controle de rede;
- Enviar mensagens para o NCC quando as condições locais sofrerem uma alteração importante (STALLINGS, 2005).

Um *host* da rede é escolhido como gerenciador. Além da NME, o gerenciador incorpora um *software* chamado *Networking Management Application* -Aplicação de Gerenciamento de Rede (NMA). A NMA possui uma *interface* para permitir que um usuário autorizado gerencie a rede. Essa *interface* responde aos comandos do usuário exibindo informações e permite que ele emita comandos às NMEs que estão na rede. A comunicação entre os componentes da rede é feita por um protocolo de gerenciamento de rede.

Cada nó da rede inclui uma NME e é chamada de agente. Os agentes possuem os sistemas que os usuários utilizam. Como está representado na Figura 1 o *host* que controla a rede pode se comunicar com outros NMEs e para manter a alta disponibilidade dessa função são utilizados um ou mais *hosts* para administrar a rede.

O protocolo *Simple Network Management Protocol*, Protocolo Simples de Gerenciamento de Rede (SNMP) foi desenvolvido para servir de ferramenta de redes e inter-redes operando o *Transmission Control Protocol/Internet Protocol* – Protocolo de Controle da Transmissão/Protocolo de *Internet* (TCP/IP). Para Stallings (2005), esse protocolo possui alguns conceitos básicos e inclui os seguintes elementos-chave:

- Estação de gerenciamento, ou gerenciador;
- Agente;
- Base de informações de gerenciamento;
- Protocolo de gerenciamento de rede.

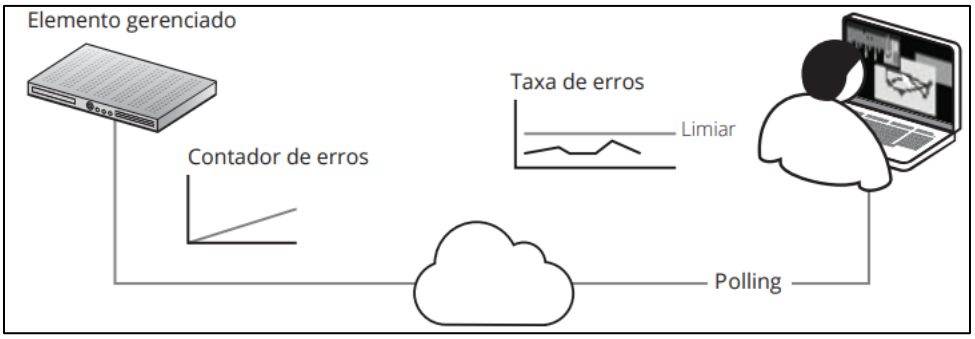
O gerenciador age como a *interface* entre o administrador e a ferramenta de administração de rede, o gerenciador deve ter no mínimo:

- Um conjunto de aplicações de gerenciamento para análise de dados, recuperação de falhas etc.;

- Uma *interface* com o usuário pela qual o administrador pode monitorar e controlar a rede;
- A capacidade de traduzir as necessidades do administrador no monitoramento e controle reais dos elementos na rede;
- Uma base de dados de gerenciamento de rede extraídas dos bancos de dados de todas as entidades gerenciadas na rede (STALLINGS, 2005).

Os agentes de gerenciamento são os elementos ativos na rede, como *hosts*, roteadores, hubs, impressoras etc. Esses agentes respondem as requisições de dados e ações a partir de um gerenciador, enviando informações importantes mesmo quando não é solicitado. Esse processo de troca de informações enviadas sob demanda ou periodicamente é chamado de *polling* (STALLINGS, 2005). A Figura 2 mostra esse processo.

Figura 2: Visão geral do gerenciamento de rede



Fonte: SANTOS et al. (2015).

Existem objetos na rede que são os recursos gerenciados na rede. São variáveis de dados que representam um aspecto do agente. A coleção de informações dos objetos é chamada de *Management Information Base,* Base de Informações de Gerenciamento (MIB) (STALLINGS, 2005).

Segundo Stallings (2005), a MIB atua como um grupo de pontos de acesso no agente para o gerenciador. O gerenciador pode recuperar o valor dos objetos da MIB e pode fazer com que uma ação ocorra em um agente. Dessa forma, a administração de redes permite mudar os elementos de um agente, modificando o valor de suas variáveis.

A estação e agentes de gerenciamento são vinculadas a um protocolo de gerenciamentos rede. O protocolo usado para administração das redes TCP/IP é o SNMP,

para as redes TCP/IP no modelo OSI é usada uma versão melhorada do SNMP, chamada de SNMPv2, que possui os seguintes comandos:

- *Get*: permite que o gerenciador recupere o valor dos objetos no agente;
- *Set*: permite que o gerenciador defina o valor dos objetos no agente;
- *Notify*: Permite que um agente envie notificações não solicitadas ao gerenciador sobre eventos importantes (STALLINGS, 2005).

O SNMP define uma MIB contendo variáveis e tabelas para objetos, além de especificar um protocolo para que um gerente possa obter e definir essas variáveis da MIB. Isso permite que um agente envie notificações não solicitadas, conhecidas como *traps*. (STALLINGS, 2005).

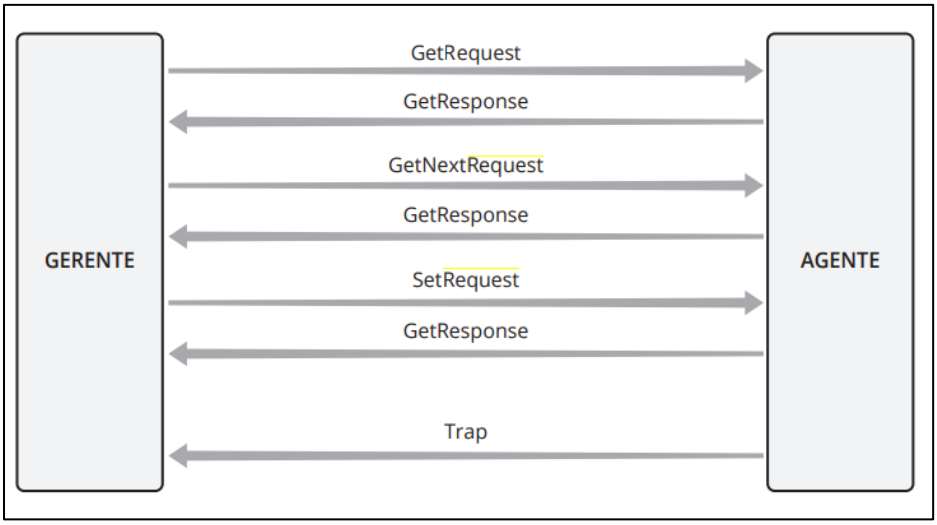
O protocolo SNPMv2 surgiu da necessidade de solucionar as deficiências da versão anterior, algumas delas eram deficiências funcionais e falta de mecanismos de proteção da rede. É implementado sobre *User Datagram Protocol*, Protocolo de Datagrama de Usuário (UDP). Sua essência é ter um protocolo usado para trocar informações de gerenciamento que define a estrutura dessas informações e os tipos de dados permitidos. Essa definição é chamada de *Management Information Structure*, Estrutura de Informações de Gerenciamento (SMI). O SMI é representado como uma linguagem que objetiva definir dados de gerenciamento (STALLINGS, 2005).

O SNMPv2 fornece um mecanismo que transfere dados entre gerenciador e agente. Essa troca é feita através de mensagens, que consistem em um *wrapper* de mensagem externo e uma *Protocol Data Unit*, Unidade de Dados de Protocolo (PDU) interna. Existem 7 tipos de mensagens que são transportadas pela PDU, as principais são:

- *GetRequest*: requisição para capturar o valor de uma variável;
- *GetNextRequest*: requisição para capturar o valor da próxima variável;
- *SetRequest*: requisição de mudança de um valor de uma variável feita pelo gerente;
- *Trap*: notificação do agente ao gerente, comunicando o acontecimento de um evento predeterminado;
- *Response*: mensagem de resposta do agente para as solicitações *Get* e *Set*; (STALLINGS, 2005; RNP, 2015).

A Figura 3 expõe como é o processo desses comandos entre gerente e agente.

Figura 3: Processo de comandos SNMP



Fonte:SANTOS et al. (2015).

3 FERRAMENTAS UTILIZADAS

Este capítulo aborda sobre a ferramenta Zabbix, tal qual sua apresentação, pré-requisitos, instalação na nuvem, arquitetura, nomenclaturas e coleta de dados. Além disso, são descritas as ferramentas Grafana e Zero Tier que complementaram a implementação feita.

3.1 Zabbix

O Zabbix é uma ferramenta essencial no contexto do gerenciamento de redes, um campo que envolve a supervisão, manutenção e otimização de uma rede de computadores para assegurar sua eficiência, segurança e desempenho contínuo. No gerenciamento de redes, a capacidade de monitorar em tempo real o estado de diversos componentes, como servidores, dispositivos de rede, aplicações e serviços, é crucial.

O Zabbix se destaca ao proporcionar uma visão abrangente e detalhada da infraestrutura de TI, coletando e analisando dados de performance e utilização de recursos. Essa visibilidade permite aos administradores identificarem rapidamente problemas potenciais, diagnosticar falhas e implementar soluções antes que afetem os usuários finais. Além disso, com suas capacidades avançadas de alertas e geração de relatórios, o Zabbix não apenas detecta anomalias, mas também facilita a tomada de decisões estratégicas, contribuindo significativamente para a eficiência operacional e a resiliência das redes corporativas.

3.1.1 Apresentação da ferramenta

O Zabbix se integra perfeitamente ao modelo e à arquitetura de gerenciamento de redes já descrita no capítulo 2 deste trabalho, que tradicionalmente segue uma abordagem hierárquica e modular para garantir uma administração eficiente e escalável. Este modelo é composto por camadas distintas de monitoramento, controle e gestão, no qual cada camada desempenha funções específicas que contribuem para a visão holística do desempenho da rede. Na arquitetura de gerenciamento de redes, o Zabbix atua como uma ferramenta centralizada na camada de monitoramento, coletando dados de diversos agentes distribuídos pela rede. Esses agentes, instalados em servidores e dispositivos de rede, reportam métricas

críticas de desempenho e estado, que são agregadas e analisadas pelo servidor central do Zabbix.

Segundo seu *site* oficial, o Zabbix é um *software* que visa monitorar numerosos parâmetros de rede, a saúde e integridade de servidores, *Virtual Machines*, Máquinas Virtuais (VM's), aplicações, serviços, banco de dados, *websites*, nuvem etc. É uma ferramenta *open source*, ou seja, o usuário pode adequar o *software* conforme o uso. Possui um mecanismo flexível de notificação, no qual o usuário pode configurar alertas para uma variedade de eventos via *e-mail*. Possibilitando ao usuário uma resposta rápida para os problemas do servidor. O Zabbix também oferece um recurso de relatórios e visualização de dados armazenados, tornando a ferramenta ideal para gerenciamento de capacidade.

O Zabbix suporta tanto o modelo "*pooling*" quanto "*trapping*". Os relatórios, estatísticas e padrões de ajustes são visualizados através de um *frontend* via *web.* Essa *interface* é encarregada de assegurar que os *status* da rede e a intangibilidade dos servidores sejam monitorados de qualquer localização. Apesar de ser gratuito, atende desde pequenas empresas com poucos servidores até grandes empresas com milhares de servidores.

3.1.2 Pré-requisitos

O *site* oficial do Zabbix fornece os pré-requisitos do *software*, conforme o Quadro 1.

Quadro 1: Pré-requisitos Zabbix

Tamanho da instalação	Métricas monitoradas	CPU/vCPU cores	Memória (GB)	Base de dados	Amazon EC2
Pequeno	1000	2	8	MySQL Server, Percona Server, MariaDB, PostgreSQL	M6i.large/m6g.large
Médio	10000	4	16	MySQL Server, Percona Server, MariaDB, PostgreSQL	M6i.xlarge/m6g.xlarge

Continua...

Tamanho da instalação	Métricas monitoradas	CPU/vCPU cores	Memória (GB)	Base de dados	Amazon EC2
Grande	100000	16	64	MySQL Server, Percona Server, MariaDB, PostgreSQL, Oracle	M6i.4xlarge/m6g,4x large
Muito Grande	1000000	32	96	MySQL Server, Percona Server, MariaDB, PostgreSQL, Oracle	M6i.8xlarge/m6g.8x large

Fonte: Zabbix (2023).

3.1.3 Instalação na nuvem

Este tópico aborda sobre nuvem, bem como seu conceito, principais provedores, tipos de nuvem, tipos de serviço, comparativo de uso, vantagens e dificuldades.

3.1.3.1 Conceito de nuvem

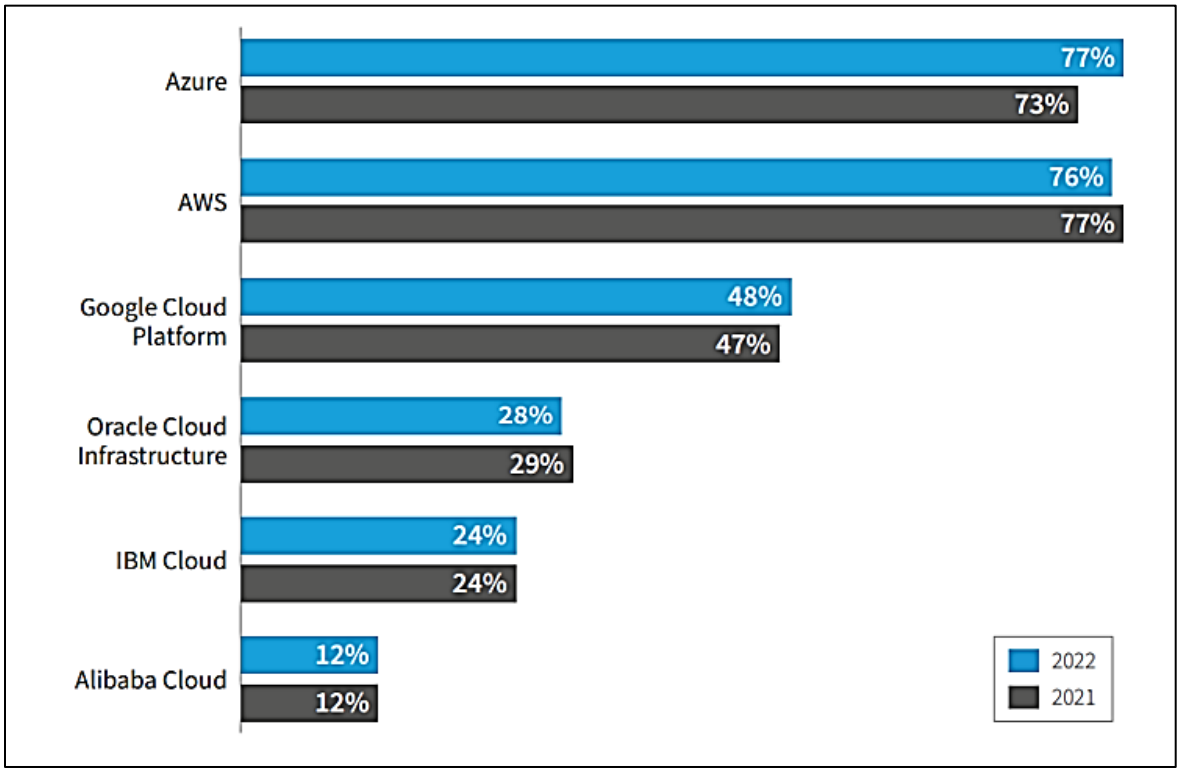
A computação em nuvem é definida como um modelo de acesso ubíquo, ou seja, está presente ao mesmo tempo e em todo lugar, sob demanda, para um conjunto de recursos computacionais compartilhados e configuráveis, que podem ser destinados e liberados com o mínimo de esforço de gerenciamento ou contato com o provedor de serviços (NIST, 2011 apud OPUS, 2015).

3.1.3.2 Principais provedores de serviços em nuvem

De acordo com Flexera (2022), empresa de tecnologia Norte Americana, os principais provedores em nuvem são: AWS, Microsoft Azure e Google Cloud Plataform (GCP). Esses três provedores possuem uma larga vantagem no mercado de serviços em nuvem e juntos representam mais de 50% de uso, conforme é exibido na Figura 4 que mostra um gráfico com os principais provedores do mercado. Com o aumento da adoção de estratégias multicloud

pelas empresas, onde utilizam mais de um provedor de nuvem simultaneamente, essa tendência justifica as porcentagens acima de 70% para AWS e Azure, refletindo a preferência por múltiplas plataformas para distribuir cargas de trabalho e mitigar riscos de dependência exclusiva de um único fornecedor.

Figura 4: Principais provedores de nuvem



Fonte: Flexera (2022).

Conforme mostra a Figura 4, a nuvem da *Microsoft* liderou o mercado no ano de 2022, logo após vieram a AWS e Google *Cloud Platform* (GCP).

3.1.3.3 Tipos de nuvem

No que se refere à implementação de um serviço em nuvem, deve-se pensar em qual tipo de nuvem esse serviço deve ser alocado. Existem três tipos de nuvem:

- Pública: tem como característica compartilhar recursos computacionais com diversos clientes e o controle das instâncias, VM's e recursos ficam por sob

responsabilidade do provedor. Um exemplo de serviço de nuvem pública é o Gmail;

- Privada: tem como característica ter seus recursos computacionais isolados dos utilizados por outras empresas. Geralmente os clientes que usam esse tipo de nuvem são do governo de algum país, por exemplo: governo dos Estados Unidos e governo do Brasil;
- Híbrida: tem como característica tem recursos compartilhados com outros clientes, porém possui uma parte desses recursos isolados como foi mencionado na nuvem privada. Esse tipo de nuvem é o mais utilizado no mercado (OPUS, 2015).

3.1.3.4 Tipos de serviço de nuvem

A nuvem possui três tipos de modelo de serviço:

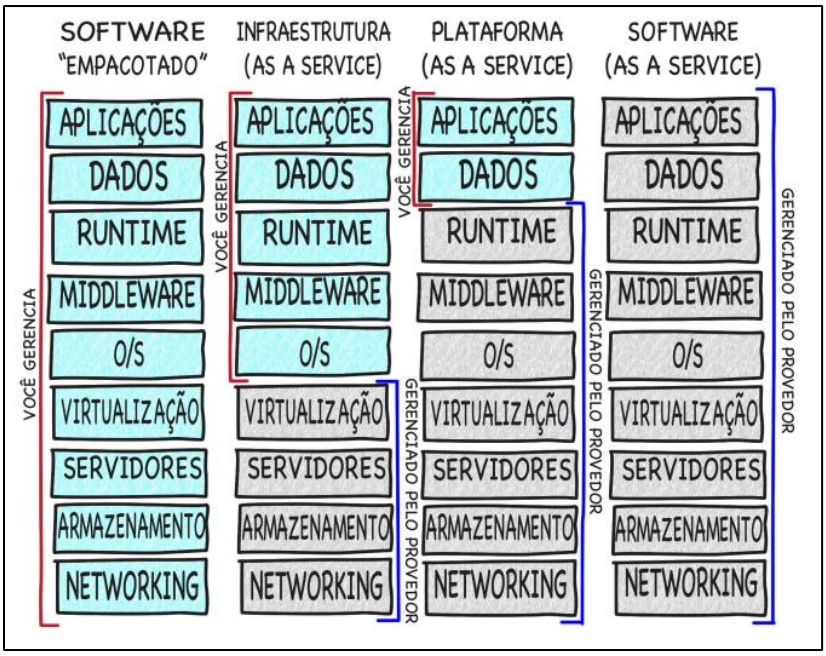
- *Infrastructure as a Service* – Infraestrutura como Serviço (IaaS): nesse tipo de serviço o cliente fica responsável por gerenciar parte dos recursos alocados, enquanto o provedor de nuvem gerencia a outra parte desses recursos, ou seja, é feita uma divisão de responsabilidades, alguns exemplos desse modelo de serviço são: AWS EC2, *Microsoft Azure Virtual Machines* e *Google Compute Engine*;
- *Plataform as a Service* – Plataforma como Serviço (PaaS): nesse tipo de serviço o cliente tem a responsabilidade de gerenciar apenas as aplicações e o dados, o restante dos recursos ficam encarregados ao provedor. Esse tipo de serviço é mais utilizado por desenvolvedores que estão criando *sotfwares*;
- *Software as a Service* – *Software* como Serviço (SaaS): nesse tipo de serviço o cliente não tem nenhuma responsabilidade sobre os recursos, tudo fica por conta do provedor. Exemplos desse modelo de serviço são: Dropbox, Trello, Zoom etc. (OPUS, 2015).

A Figura 5 mostra quais são as responsabilidades entre o cliente e o provedor para cada tipo de serviço e faz uma comparação entre os três tipos de serviço da nuvem e o *software* gerenciado pelo cliente. O *software* “empacotado” faz referência ao cliente tendo toda a responsabilidade sobre ele, desde *networking* a aplicações, optando pela infraestrutura como serviço, o provedor de nuvem fica responsável pelos servidores, virtualização,

armazenamento e *networking*, o cliente fica responsável pelo sistema operacional, middlewares, o período em que ficará ativo, dados e aplicações.

A plataforma como serviço o provedor tem mais responsabilidades, além das citadas na infraestrutura como serviço, o provedor cuida também do sistema operacional, middlewares e tempo que ficará ativo. Por fim, no software como serviço o provedor é encarregado por toda aplicação, plataforma e infraestrutura.

Figura 5: Responsabilidades para cada tipo de serviço



Fonte: OPUS (2015).

3.1.3.5 Vantagens e dificuldades do uso da nuvem

Alguns fatores contribuem para o incentivo de uso da nuvem, dentre eles têm algumas vantagens da computação em nuvem em relação ao modelo tradicional:

- Redução dos investimentos iniciais e eliminação dos custos de manutenção, segurança, eletricidade, espaço e outros que seriam necessários;
- Elasticidade e escalabilidade, ou seja, a capacidade de se ajustar dinamicamente à demanda, esticando ou encolhendo a capacidade de recursos computacionais;
- Agilidade para colocar novas aplicações no ar;
- Rapidez na implementação, incluindo tempo na aprovação de novas iniciativas.

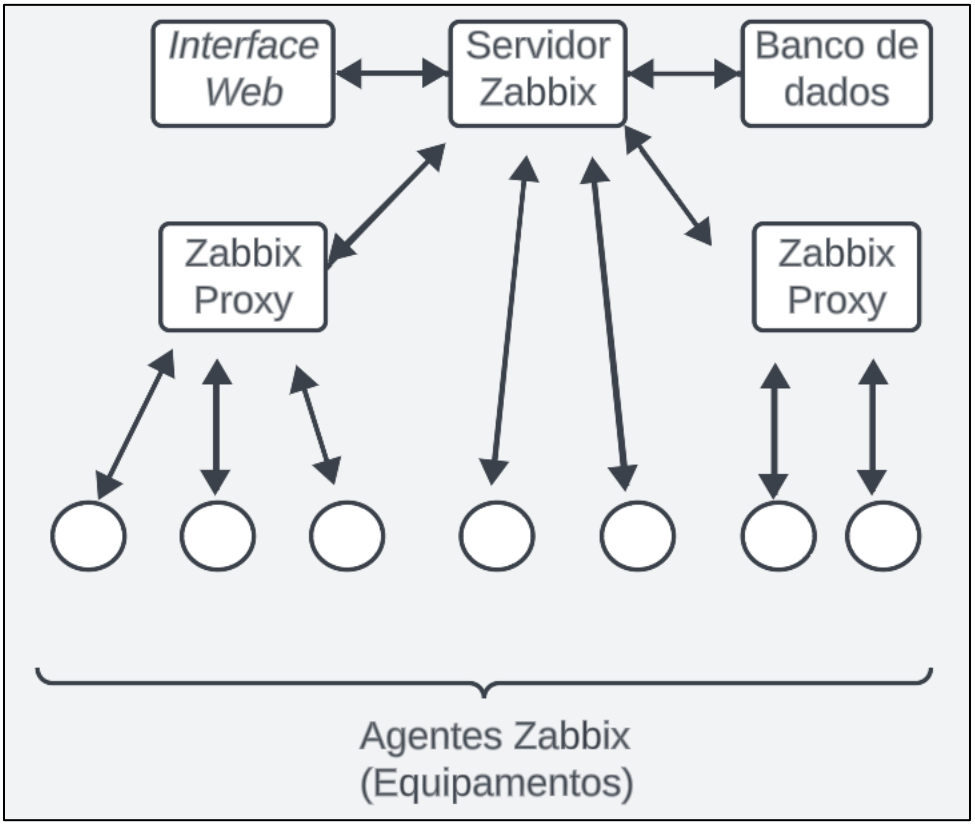
Em relação aos fatores que dificultam a adoção da Computação em Nuvem, pode-se citar:

- Necessidade de melhor integração entre os sistemas que rodam na nuvem e os sistemas que rodam internamente na organização;
- Exige acesso estável a *Internet*, com banda de comunicação adequada para o nível de uso;
- Resistência da equipe interna, que considera que o serviço aumenta o nível de complexidade do trabalho;
- Resistência de gestores de TI, que teme perder o controle sobre o ambiente operacional e perda de sua importância dentro da organização;
- Aspectos legais e de segurança, pois os gestores querem saber onde ficam alocados esses recursos fisicamente e quais a práticas legais da jurisdição desse local;
- Reações negativas em relação ao termo Computação em Nuvem (OPUS, 2015).

3.1.4 Arquitetura do Zabbix

O Zabbix possui arquitetura no modelo *three-tier,* que faz uma abordagem em três camadas, que são: a aplicação, o banco de dados e a *interface web.* A Figura 6 exibe essa arquitetura.

Figura 6: Arquitetura do Zabbix



Fonte: Adaptado de INSTITUTO METRÓPOLE DIGITAL, [s. d.]

A aplicação representa o *back-end* e é encarregada de recolher os dados nos ativos da rede. O banco de dados é delegado a arquivar os dados recolhidos pelo *back-end* e apresentá-las ao *front-end*. Essa camada é representada pela base de dados. A *interface web* é representada pelo *front-end*, e é encarregada de dar acesso as informações de monitoramento aos administradores e fornecer informações para as aplicações que utilizam *Application Programming Interface* – Interface de Programação de Aplicação (API) do Zabbix (LIMA, 2014).

Dentro dessa arquitetura, existem três elementos que representam o *back-end.* São eles:

- Zabbix *Server*: nele todos os agentes se reportam ao *back-end*, que tem a função de armazenar as informações recolhidas na base de dados. Esses dados ficam acessíveis através do *front-end.;*

- Zabbix Proxy: é um elemento opcional e é um *host* responsável por fazer a coleta em cliente remotos, o *Server* não depende dele para funcionar. É um empilhador de informações que faz a coleta dos clientes na rede remota para o *Server*. Feita a coleta desses dados, o *Proxy* consolida os dados e transfere pacotes com todas as informações ao *Server*. O *hardware* utilizado pelo *Proxy* não precisa dos mesmos requisitos que o *Server* e sua manutenção é praticamente zero;
- Zabbix *Agent*: é o cliente que se reporta para o Zabbix Server ou para o Zabbix Proxy. Foi desenvolvido para consumir poucos recursos computacionais e não impactar o ambiente monitorado. Também pode ser visualizado com agentes externos, tais como: SNMP, IPMI e SSH (LIMA, 2014).

3.1.4.1 Tipos de agentes

O Zabbix possui dois tipos de agentes, o passivo e o ativo. O agente é passivo quando ocorre uma conexão entre o servidor e o *host* para recolher informações solicitadas (LIMA, 2014). Essa coleta é feita periodicamente e o *host* responde a solicitação.

O agente é ativo quando o *host* possui um rol de itens que precisam ser enviadas ao servidor. Essas coletas serão monitoradas e enviadas durante intervalo de tempo. Dessa forma mesmo que o servidor esteja indisponível, ele pode receber os dados coletados quando voltar a estar disponível (LIMA, 2014).

3.1.5 Nomenclaturas

Este tópico aborda sobre algumas nomenclaturas que são importantes para compreender melhor sobre o monitoramento do Zabbix.

3.1.5.1 Item

É a fonte de informação que o Zabbix usa para coletar dados com o objetivo de retornar uma métrica. Alguns desses tipos são:

- Agente Zabbix (passivo): a consulta é realizada pelo servidor;
- Agente Zabbix (ativo): os dados são processados pelo agente e transmitidos para o servidor;

- Monitoramento simples: executado pelo servidor;
- Arquivos de *log*: arquivos de *log* dos sistemas *Unix-like* e *Event Viewer* do Windows;
- Banco de dados: estáticas de base dados através do *query (*LIMA, 2014).

3.1.5.2 Mídia

É uma tarefa de encaminhamento de mensagens disponibilizado pelo Zabbix que notificam as ocorrências com base em mídias criadas e configuradas. São cinco tipos de mídia:

- *E-mail*;
- Jabber;
- SMS;
- *Script*;
- *Ez Texting* (serviço pago) (LIMA, 2014).

Esse serviço possibilita que quando a *trigger* é acionada, uma modificação no *host*, updates disponíveis, dentre outros eventos, uma mensagem é enviada notificando o administrador da rede, ainda que este esteja longe do painel da NCC, ele é comunicado do evento.

3.1.5.3 *Trigger*

É uma expressão lógica, uma regra que é avaliada a cada coleta de um item (LIMA, 2014). Esse item chega com um valor que é analisado e se estiver vinculado a uma *trigger* o Zabbix toma uma decisão em concordância com a expressão lógica configurada.

A partir dessa situação, pode-se ter alertas com alguns níveis de severidade, que representam o nível de criticidade do ambiente monitorado. São seis níveis de criticidade:

- Não classificada;
- Informação;
- Atenção;
- Média;
- Alta;

- Desastre (LIMA, 2014).

3.1.5.4 Mapa

A visualização por mapas serve para se analisar um conjunto de *hosts* separados por dois ou mais *links*. O cenário mais comum de uso seria entre uma matriz e suas filiais, em que pode ser monitorado o *link* entre essas unidades, além de alertar os principais sistemas e serviços. O uso de mapas é simples, e pode ser personalizado com imagens de fundo representando cidades, estados ou países para mostrar a interligação dos *links* entres os *hosts* da rede (LIMA, 2014).

3.1.6 Coleta de dados

O Zabbix possui inúmeras medidas para recolher informações dos equipamentos ativos da rede. Essa coleta torna o gerenciamento seja mais eficiente. As principais formas de coleta são:

- Agente Zabbix: é instalado nos dispositivos que foram gerenciados, realiza a supervisão dos recursos e aplicações como, memória *Random Acess Memory*, Memória de Acesso Randômico (RAM), *interface* de rede, disco, processador. Quando ocorrer algum tipo de falha nesse dispositivo é emitido um alerta pelo agente. O agente depende dos parâmetros do *Proxy* e centraliza os dados do dispositivo monitorado para o *Server*;
- Agente SNMP: o Zabbix por possuir suporte ao SNMP, permite que ocorra a supervisão e recolhe as informações dos equipamentos da rede, essa coleta só poderá ser feita se o dispositivo possuir suporte a esse protocolo em seu *firmware*.

3.2 Grafana

O Grafana é um *software* de código aberto que permite ao usuário visualizar, consultar, alertar e explorar métricas. Teve início em 2014 por seu criador Torkel Ödegaard devido a sua insatisfação com algumas funcionalidades de outro *software*, o Kibana. Segundo Torkel (2019), uma de suas insatisfações com o Kibana era o ruído visual do painel de visualização, que deixava o usuário distraído. Em 2014 a versão 1.0.0 foi lançada e nela o

usuário podia criar painéis e editar com facilidade, dessa forma, o usuário recebia um *feedback* visual à medida que as edições fossem feitas.

O Grafana está entre os principais *softwares* de observabilidade do mercado, possuindo mais de 42 milhões de *plugins* baixados em 2023 para fazer a integração com outros *softwares*. Para integrar o Grafana com o Zabbix é necessário usar o *plugin* Grafana-Zabbix. O requisito mínimo para uso é a versão 4.0 do Zabbix. Após instalar e configurar o *plugin*, a base de dados do Zabbix fica acessível ao Grafana e, dessa forma, painéis personalizados podem ser criados (GRAFANA LABS, 2024).

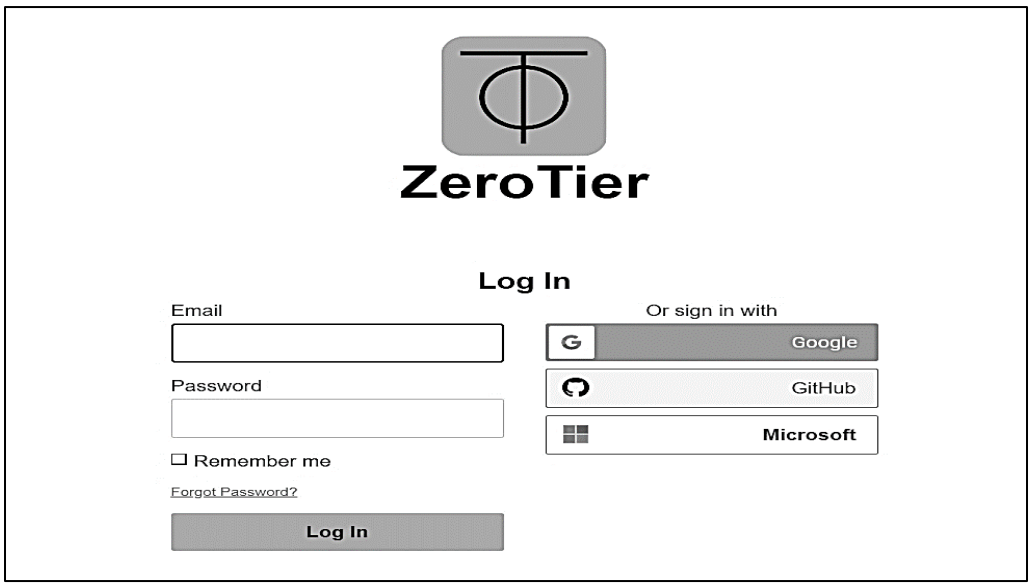
3.3 ZeroTier

Segundo seu *site* oficial, o ZeroTier é um *software* que permite criar uma *Virtual Private Network* – Rede Privada Virtual (VPN) que possibilita conectar diretamente dispositivos do mundo todo. Oferece uma solução rápida e flexível para implantar e manter seguras redes *Zero Trust*. Foi fundado em 2015 por Adam Lerymenko e atualmente possuí mais de 3 milhões de dispositivos conectados.

Para conectar o Zabbix Server com as máquinas monitoradas, foi criado uma VPN incluindo o servidor e todas as máquinas monitoradas na mesma rede. Dessa forma, a partir dessa conexão foi possível coletar os dados de cada agente instalado. Para criar a VPN é necessário seguir alguns passos, que são mostrados a seguir.

Primeiramente é necessário criar uma conta no *site* da ZeroTier e se autenticar conforme mostra a Figura 7.

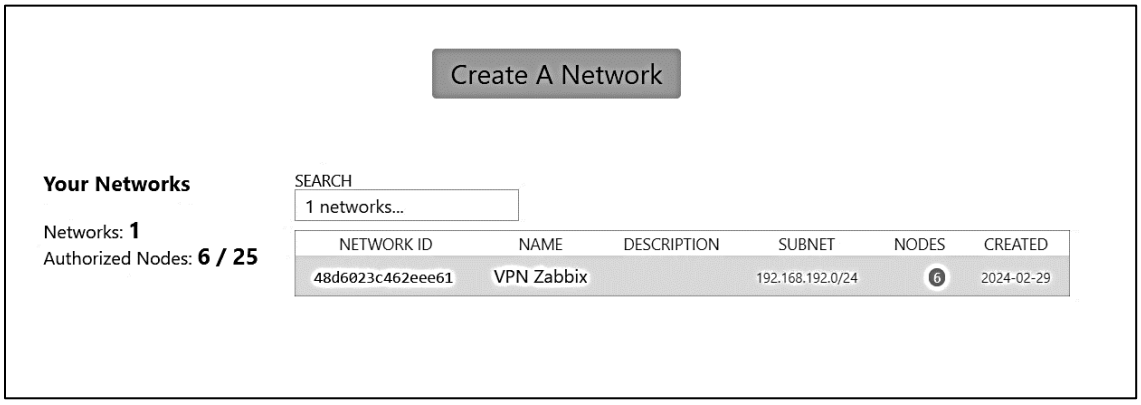
Figura 7: Autenticação no *site* da ZeroTier



Fonte: Tela capturada pelo autor desse trabalho de ZeroTier (2024).

Em seguida é necessário criar uma rede. Para tanto clica-se no botão “Create A Network” para gerar a rede, conforme mostra a Figura 8.

Figura 8: Painel de rede ZeroTier



Fonte: Tela capturada pelo autor deste trabalho de ZeroTier (2024).

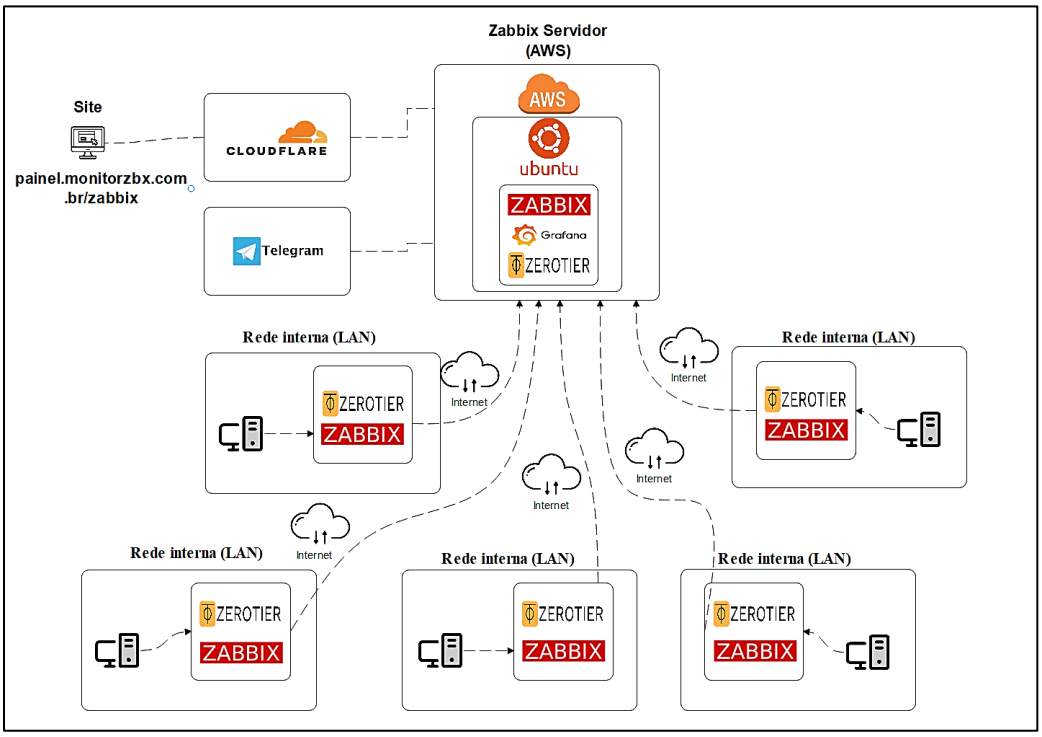
4 DESCRIÇÃO DO EXPERIMENTO

O experimento deste trabalho consiste em monitorar uma infraestrutura de rede em nuvem, aplicando os conceitos relacionados a administração de rede e do *software* mencionados no Capítulo 2 desse trabalho. Foi feita a configuração da infraestrutura, instalação das ferramentas abordadas no Capítulo 3 desse trabalho e posteriormente a integração das ferramentas Telegram e Grafana que objetiva facilitar a visualização de informações ao usuário. O ambiente escolhido monitorado foi o serviço EC2 da AWS. Este provedor de nuvem foi escolhido pelo fato de estar entre os três mais utilizados no mundo. No total foram cinco *hosts* físicos monitorados, que pertencem a pessoas que se dispuseram a colaborar com este trabalho. Três deles se localizam na cidade de Goiânia, Goiás, um na cidade de Porangatu, Goiás e outro na cidade de Pelotas, Rio Grande do Sul.

4.1 Topologia do ambiente

A Figura 9 apresenta a infraestrutura que foi gerenciada ao longo do trabalho.

Figura 9: Topologia do ambiente



Fonte: Autoria Própria (2024).

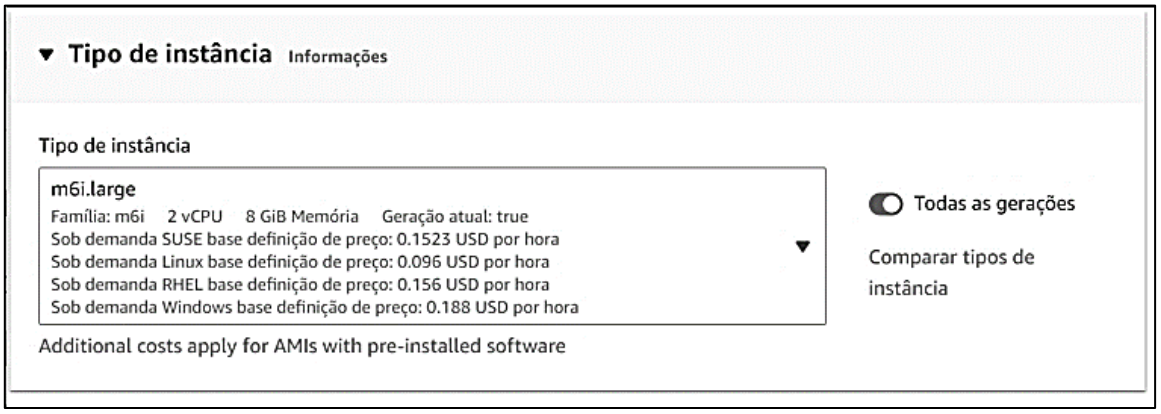
4.2 Descrição do ambiente de implementação

Este tópico aborda sobre o ambiente de implementação, tal qual o servidor nuvem, domínio *web* e serviço *Domain Name System* – Sistema de nome de domínio (DNS).

4.2.1 Servidor nuvem

A implementação do servidor Zabbix foi feita no provedor de nuvem *Amazon Web Services* (AWS), que foi lançado em 2006 e dispões de vários serviços de *cloud computing.* O serviço utilizado para a execução do trabalho foi o *Amazon Elastic Computing* (EC2), que proporciona recursos computacionais com alta elasticidade através de *Virtual Machines* que são chamadas de instâncias. Esse serviço também permite ao usuário configurar o sistema operacional usado, a quantidade de *Virtual Central Process Unit* - Unidade de Processo Central Virtual (VCPU), o tamanho do disco rígido e memória RAM. A VM utilizada como servidor do Zabbix foi a m6i.large, a Figura 10 mostra as especificações dessa VM. Posteriormente foram configurados os meios de acesso a essa VM e as etapas de instalação da ferramenta de gerenciamento de rede.

Figura 10: Especificações da VM



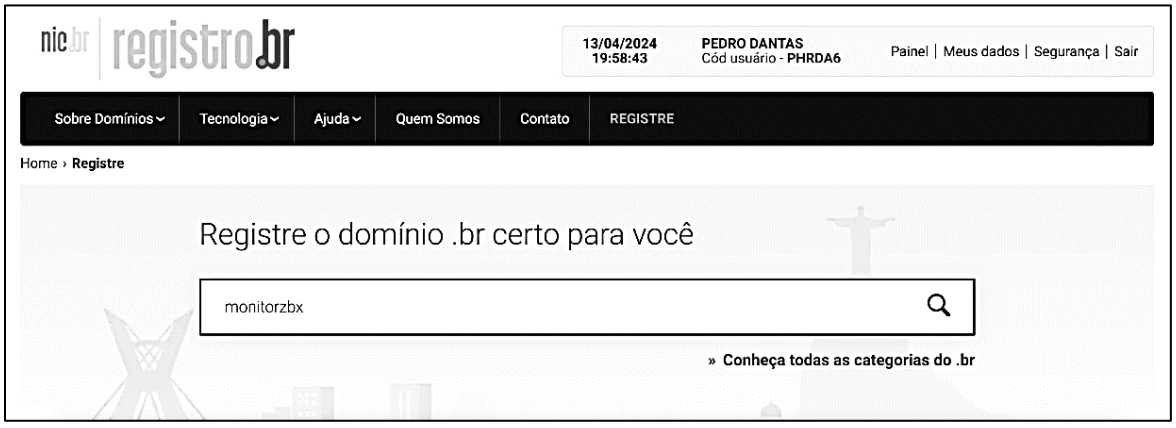
Fonte: AWS (2023).

4.2.2 Domínio *Web*

Para facilitar o acesso ao painel do Zabbix foi criado um domínio usando o provedor Registro.br. e foi escolhido por fazer parte do Núcleo de Informação e Coordenação do Ponto

BR (NIC.BR). Para registrar um domínio nesse provedor, o usuário precisa escolher um nome que esteja disponível e logo em seguida é necessário escolher o tempo que esse domínio fica ativo. As Figuras 11 e 12 ilustram esses dois primeiros passos a serem seguidos.

Figura 11: Nome do domínio



Fonte: Tela capturada pelo autor deste trabalho de Registro.br, 2024

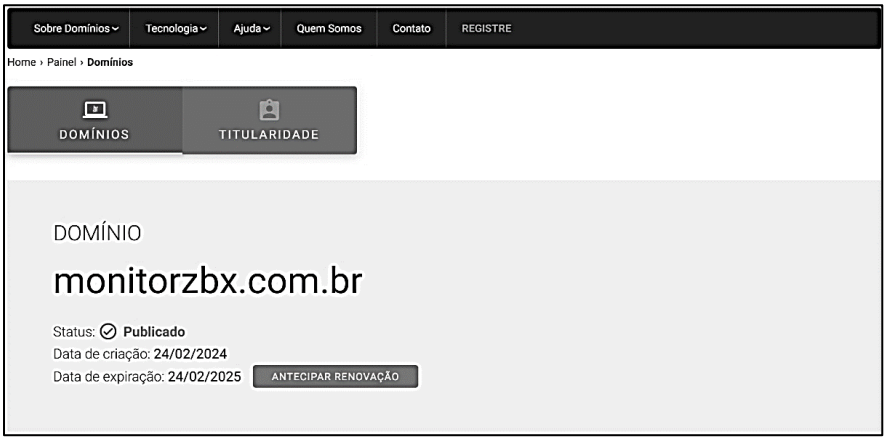
Figura 12: Valores referentes ao tempo que o domínio fica ativado

Fonte: Tela capturada pelo autor deste trabalho de Registro.br (2024).

Após executar os passos já mencionados, o usuário deve aguardar a validação do seu domínio por parte do registro.br e, após a validação o painel é atualizado e é possível

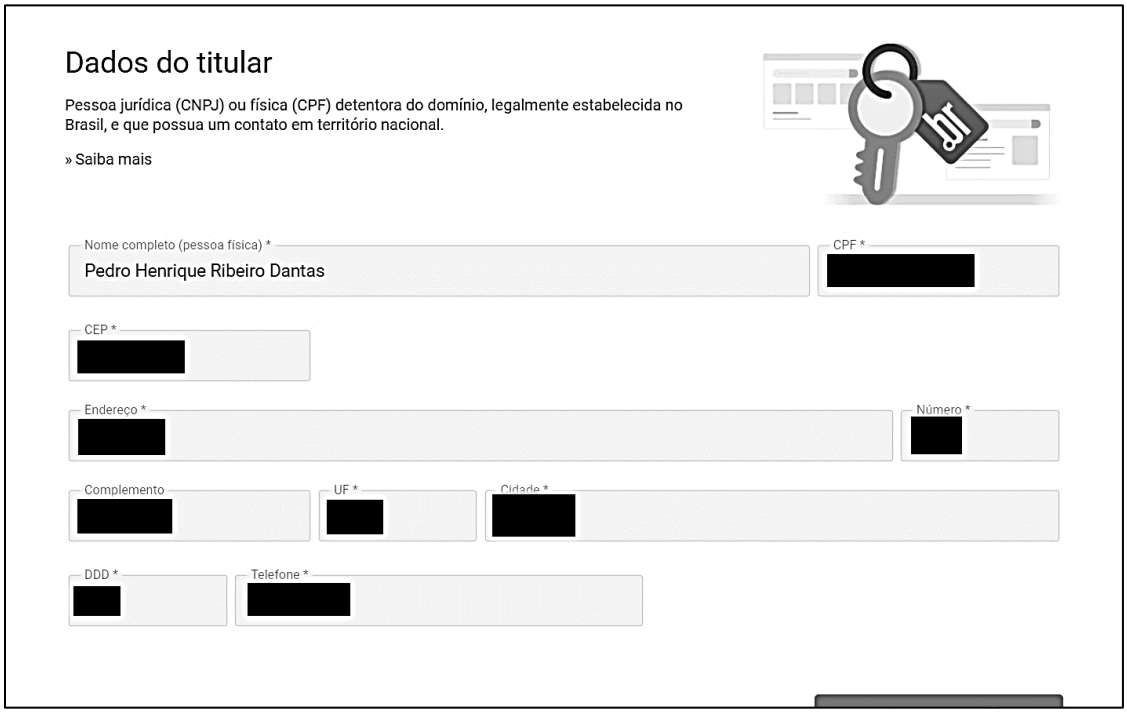
visualizar o domínio registrado, os dados do titular, contatos com o responsável e funções para troca de DNS. As Figuras 13,14,15 e 16 ilustram essas informações.

Figura 13: Painel de status do domínio



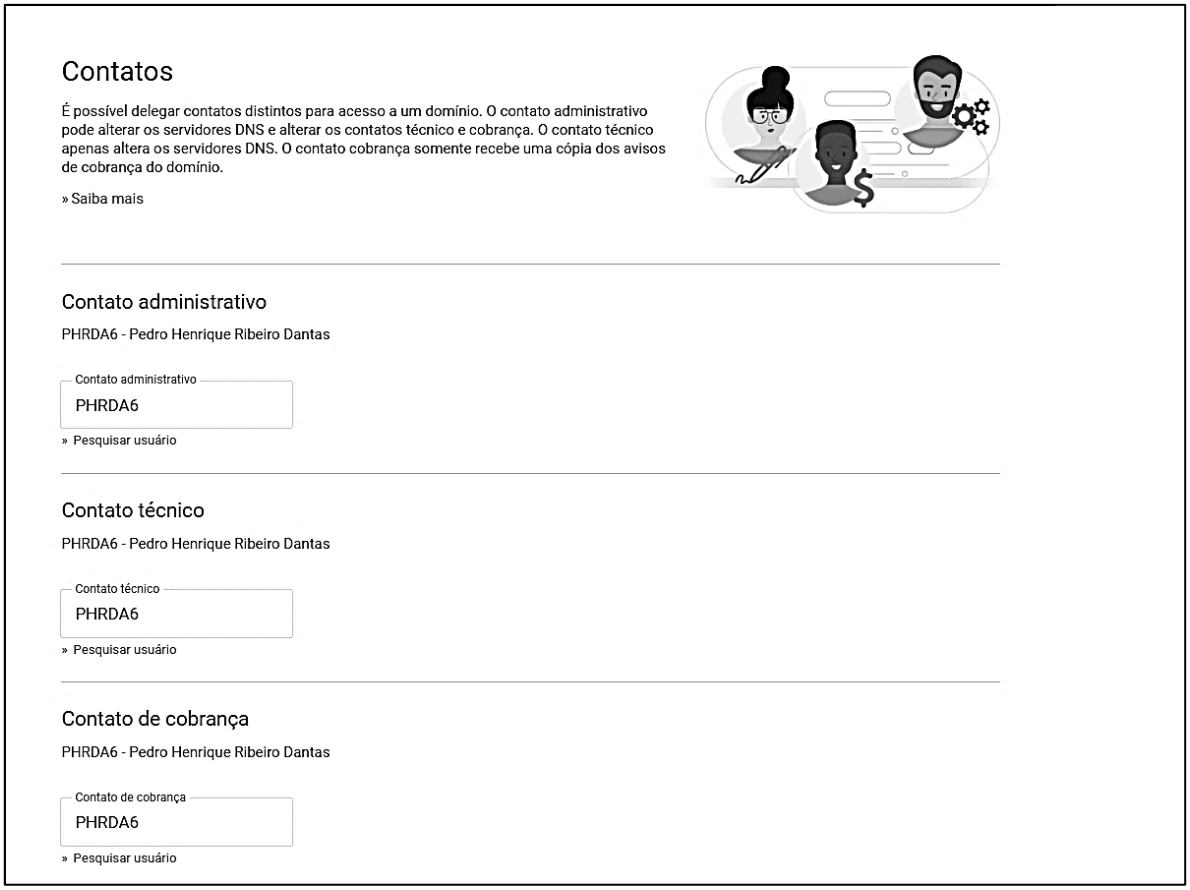
Fonte: Tela capturada pelo autor deste trabalho de Registro.br (2024).

Figura 14:Informações do titular do domínio



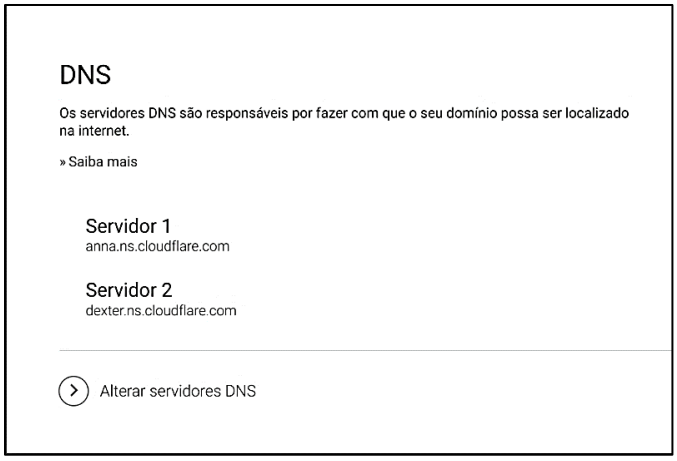
Fonte: Tela capturada pelo autor deste trabalho de Registro.br (2024).

Figura 15:Contatos do responsável pelo domínio



Fonte: Tela capturada pelo autor deste trabalho de Registro.br (2024).

Figura 16:Painel de informação e configuração do DNS



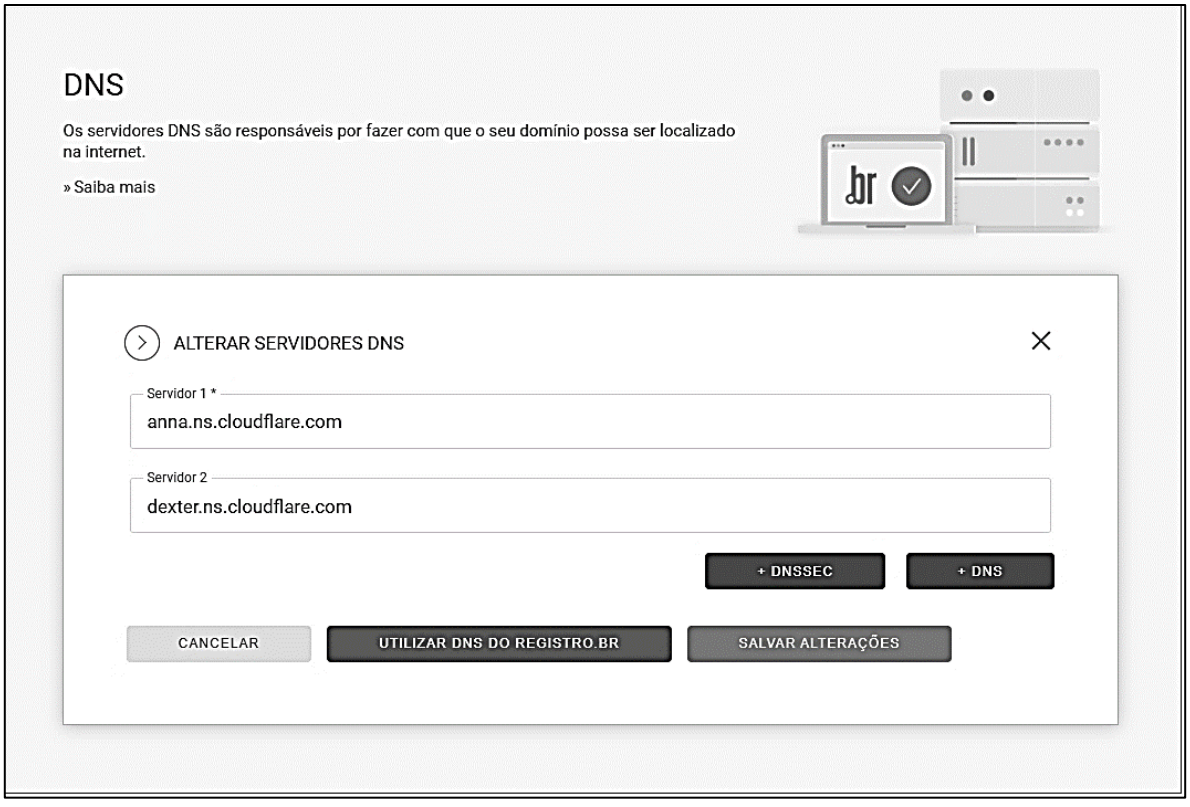
Fonte: Tela capturada pelo autor deste trabalho de Registro.br (2024).

4.2.3 Serviço DNS

Para facilitar o acesso ao painel do Zabbix foi utilizado um serviço de DNS que tem como objetivo converter as solicitações de nomes em endereço *Internet Protocol* – Protocolo de Internet (IP). O serviço escolhido foi o Cloudflare, que disponibiliza um plano gratuito e fornece recursos de proteção contra ataques e fornece um certificado *Secure Sockets Layer* - Camada de Soquete Seguro (SSL), que ajuda a proteger os dados e verifica a autenticidade do *site*.

Ao entrar na página inicial da conta, a plataforma mostra uma opção de adicionar o *site* que foi registrado pelo usuário e logo após adicionar o endereço a plataforma gera dois servidores DNS para serem alterados no Registro.br. Após fazer a alteração a provedora pede um prazo de 4 horas para efetuar a troca solicitada, conforme mostra a Figura 17.

Figura 17:Painel de configuração DNS no Registo.br



Fonte: Tela capturada pelo autor deste trabalho de Registro.br (2024).

4.3 Configuração do ambiente

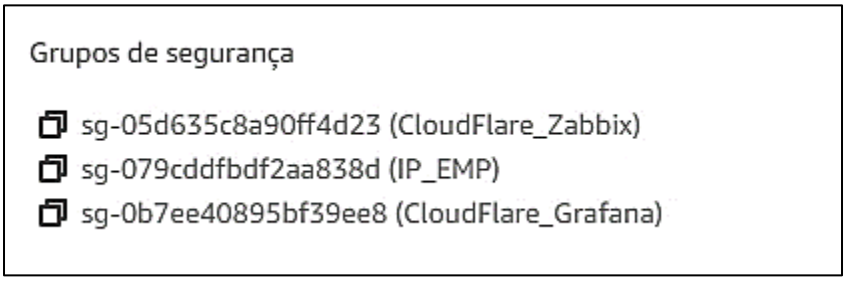
Este subtópico aborda as especificações da VM que foi utilizada como servidor do Zabbix, roteamento nos agentes, integração do Zabbix com o Telegram, integração com Grafana, serviço *web* e Zabbix Agent.

4.3.1.1 Servidor AWS

Na instância utilizada como servidor do Zabbix, foi necessário configurar três grupos de segurança visando proteger o acesso ao servidor. Cada requisição de acesso de usuário foi direcionada para a Cloudflare que acessa o servidor e envia a resposta e, dessa forma, evita que alguém acesse diretamente a instância.

O primeiro grupo criado foi o IP_EMP, que tem como objetivo permitir que uma estação tenha acesso a instância. O segundo grupo foi o CloudFlare_Zabbix que foi configurado para aceitar nas regras de entrada um conjunto de endereços IPs fornecidos pelo Cloudflare e protocolos do tipo *Hypertext Transfer Protocol,* Protocolo de Transferência de Hipertexto (HTTP) na porta 80 e *HyperText Transfer Protocol over SSL*, Protocolo de Transferência de Hipertexto sobre SSL (HTTPS) na porta 443. O terceiro grupo foi o CloudFlare_Grafana que foi configurado para aceitar nas regras de entrada o protocolo do tipo *Transmission Control Protocol*, Protocolo de Controle da Transmissão (TCP) para a porta 3000. A Figura 18 ilustram os grupos de segurança criados.

Figura 18:Grupos de segurança da instância AWS

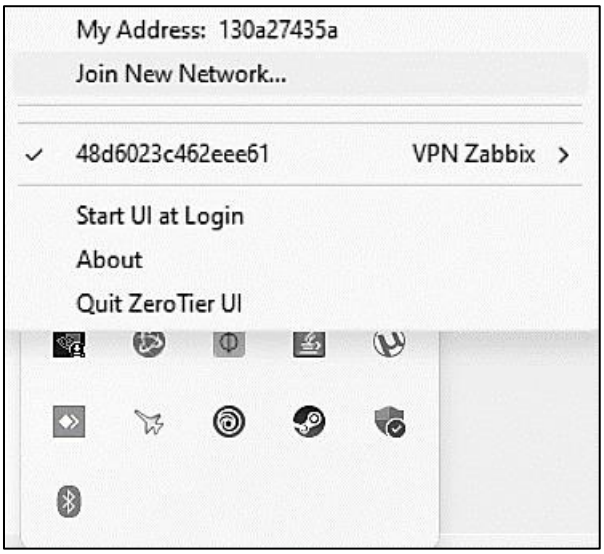


Fonte: Tela capturada pelo autor deste trabalho das configurações da instância alocada na AWS (2024).

4.3.1.2 Roteamento nos agentes

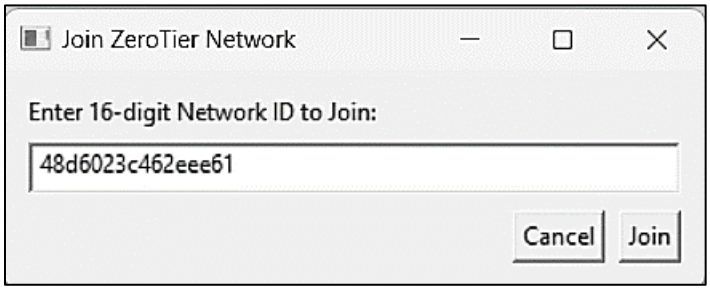
Para estabelecer comunicação boa e segura entre servidor e estações, foi criada uma VPN utilizando o ZeroTier para incluir cada *host* do ambiente. Foi necessário criar a VPN, pois não estava ocorrendo comunicação entre servidor e estações, após instalar e configurar o ZeroTier no servidor e em cada estação foi possível obter comunicação e coletar dados através do agente do Zabbix, as Figuras 19, 20 e 21 ilustram o processo de configuração em uma estação.

Figura 19: Opção para adicionar a rede



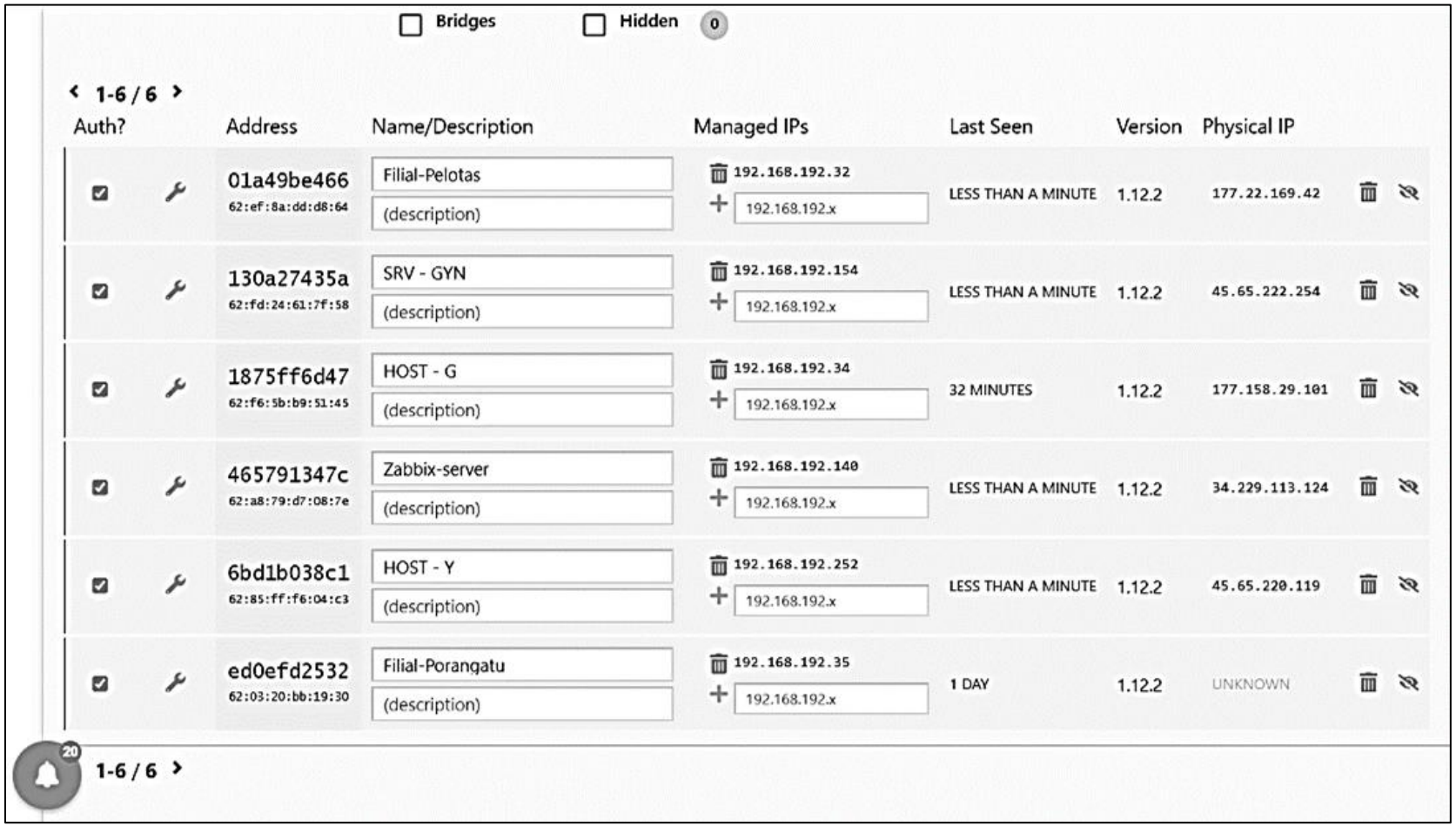
Fonte: Tela capturada pelo autor deste trabalho do programa ZeroTier (2024).

Figura 20: Tela de adição da rede



Fonte: Tela capturada pelo autor deste trabalho do programa ZeroTier (2024).

Figura 21:Painel de *hosts* adicionados a rede



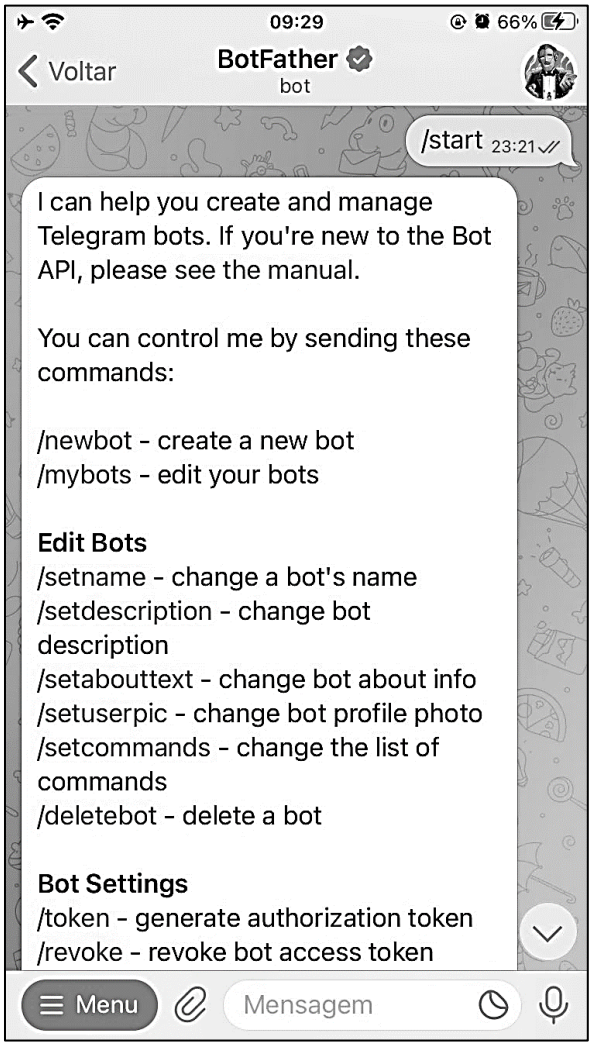
Fonte: Tela capturada pelo autor deste trabalho do painel de rede do *site* do ZeroTier (2024).

4.3.1.3 Integração do Zabbix com Telegram

Os incidentes Zabbix detectados e notificados ao usuário, são enviadas via Telegram. Cada notificação de incidente indica o nome do *host*, a data e a hora que ocorreu o incidente e o seu nível de severidade. As notificações de resolução do problema indicam o *host* onde ocorreu a resolução e o horário de recuperação.

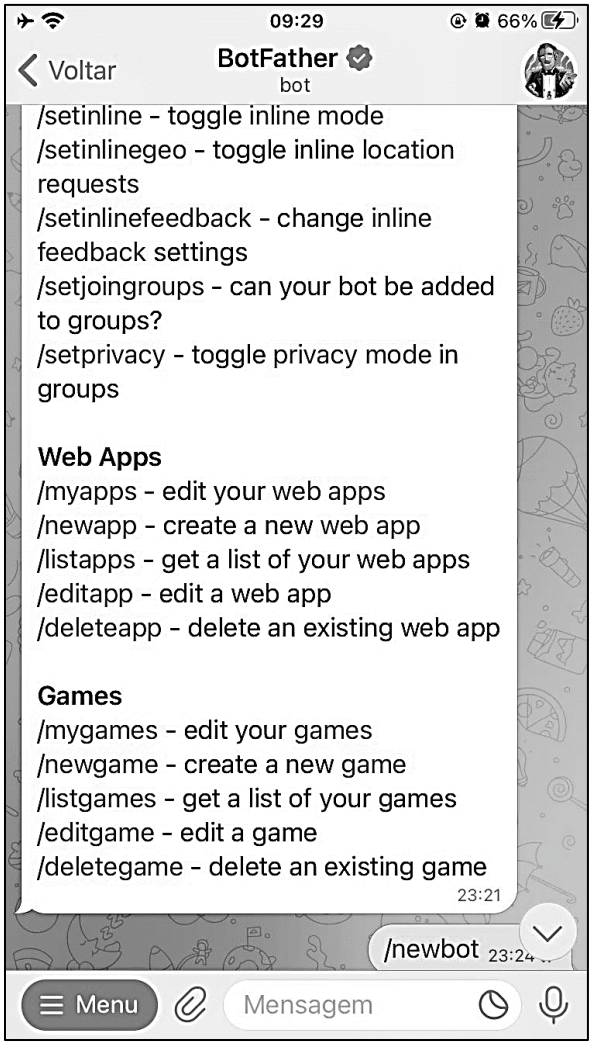
Desta forma, o usuário mesmo estando distante da *interface* de monitoramento, é notificado e pode tomar alguma providência independente de sua localização. Para integrar o Zabbix com o Telegram é necessário usar um *bot* que é responsável por criar a automação utilizada nesse projeto. As Figuras 22, 23, 24 e 25 ilustram o processo de criação do *bot*.

Figura 22: Processo de criação do bot



Fonte: Tela capturada pelo autor deste trabalho do Telegram (2024).

Figura 23: Continuação do processo de criação do bot

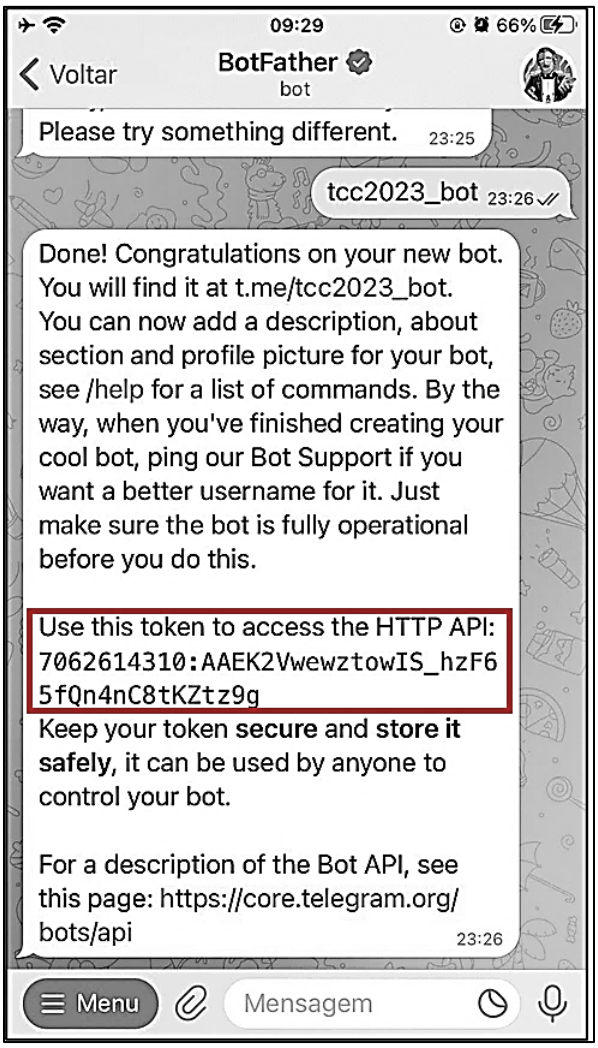


Fonte: Tela capturada pelo autor deste trabalho do Telegram (2024).

Figura 24: Processo de escolha do nome do bot

Fonte: Tela capturada pelo autor deste trabalho do Telegram (2024).

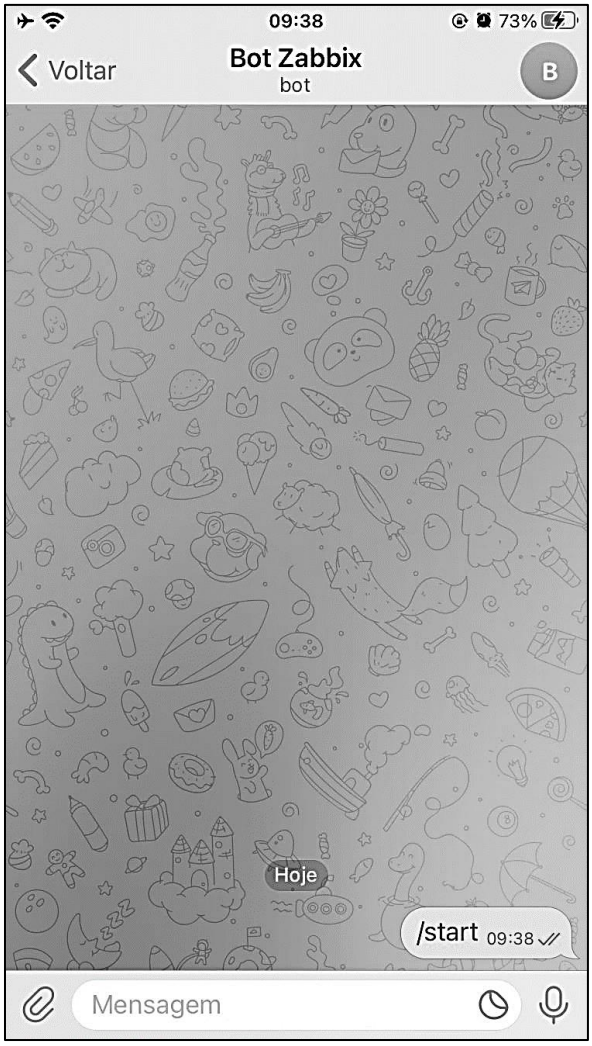
Figura 25: Token de acesso a API



Fonte: Tela capturada pelo autor deste trabalho do Telegram (2024).

Após criar o *bot* e gerar o *token* é necessário validar sua funcionalidade. O teste pode ser feito enviando uma mensagem para o *bot* e acessando a *Uniform Resource Locator* – Localizador Uniforme de recursos (URL) https://api.telegram.org/bot7062614310:AAEK2VwewztowIS_hzF65fQn4nC8tKZtz9g/getUpdates para verificar se a mensagem foi enviada. As Figuras 26 e 27 ilustram o teste feito.

Figura 26: Mensagem teste enviada para testar o funcionamento do bot



Fonte: Tela capturada pelo autor deste trabalho do Telegram (2024).

Figura 27: Informações recebidas da mensagem de teste enviada ao bot


```
{
  "ok": true,
  "result": [
    {
      "update_id": 182860628,
      "message": {
        "message_id": 2982,
        "from": {
          "id": 5472553918,
          "is_bot": false,
          "first_name": "Pedro",
          "last_name": "Henrique",
          "language_code": "pt-br"
        },
        "chat": {
          "id": 5472553918,
          "first_name": "Pedro",
          "last_name": "Henrique",
          "type": "private"
        },
        "date": 1715517505,
        "text": "/start",
        "entities": [
          {
            "offset": 0,
            "length": 6,
            "type": "bot_command"
          }
        ]
      }
    }
  ]
}
```

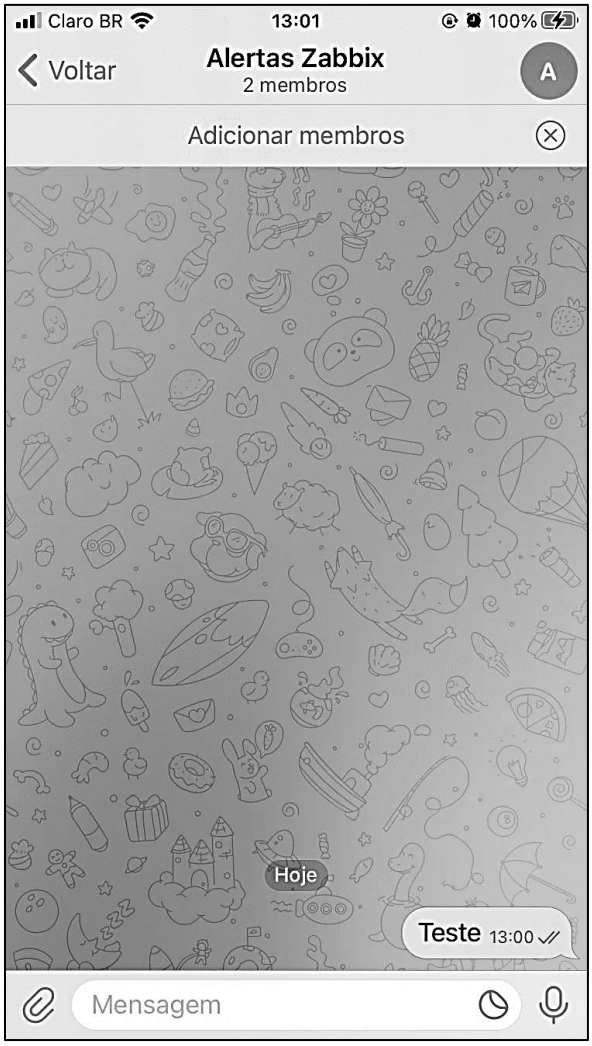

Fonte: Tela capturada pelo autor deste trabalho do Telegram (2024).

Feito os testes, os próximos passos foram selecionar o usuário que envia as informações coletadas, criar um grupo com o *bot* e configurar o Zabbix para enviar os alertas. Dentro do Zabbix na aba de usuários, foi selecionado o usuário que envia as notificações ao Telegram. Dentro do usuário é colocado a ID do grupo criado com o bot. As Figuras 28, 29, 30, 31, 32 e 33 ilustram o processo.

Figura 28: Grupo criado com o bot

Fonte: Tela capturada pelo autor deste trabalho

Figura 29: Mensagem teste enviada para obter a ID do grupo de alertas



Fonte: Tela capturada pelo autor deste trabalho

Figura 30: Comando para obter a ID do grupo com o bot

```
https://api.telegram.org/bot7062614310:AAEK2VwewztowIS_hzF65fQn4nC8tKZtz9g/getUpdates
```

Fonte: Tela capturada pelo autor deste trabalho

Figura 31: Resultado do comando executado

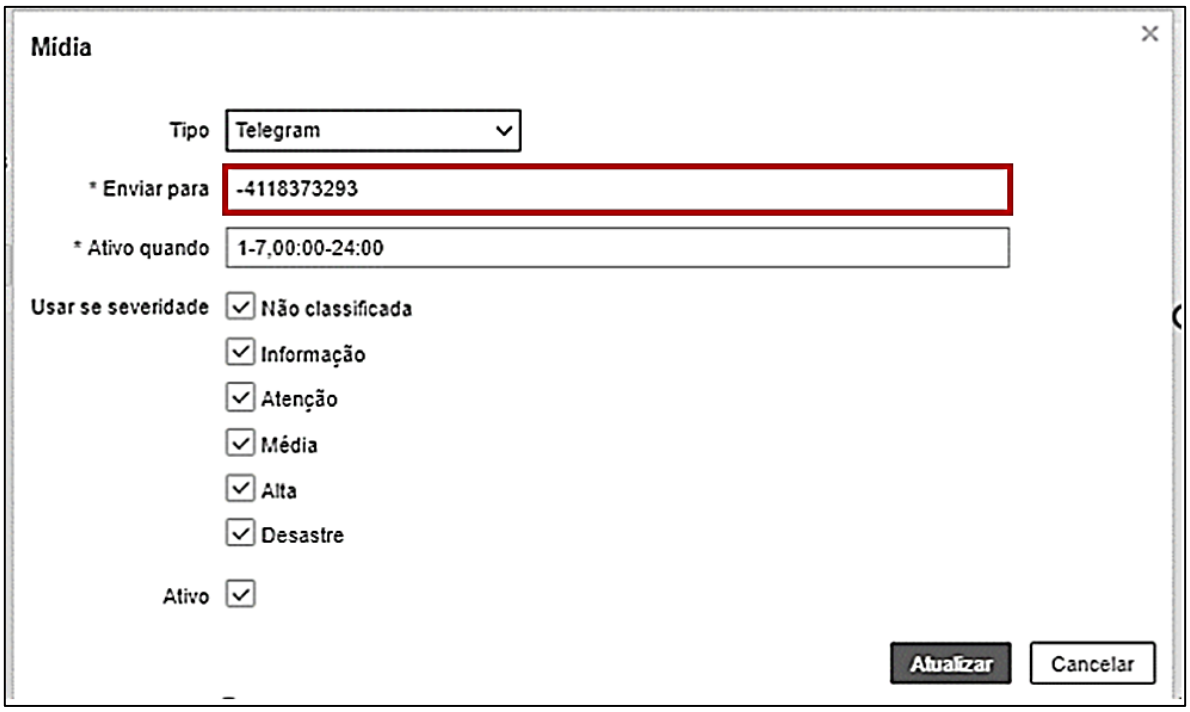

```
{
  "update_id": 182860629,
  "message": {
    "message_id": 2987,
    "from": {
      "id": 5472553918,
      "is_bot": false,
      "first_name": "Pedro",
      "last_name": "Henrique",
      "language_code": "pt-br"
    },
    "chat": {
      "id": -4118373293,
      "title": "Alertas Zabbix",
      "type": "group",
      "all_members_are_administrators": true
    },
    "date": 1715518947,
    "group_chat_created": true
  }
},
```


Fonte: Tela capturada pelo autor deste trabalho

Figura 32:Aba de usuários

Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Figura 33:Janela de configuração de mídia do usuário



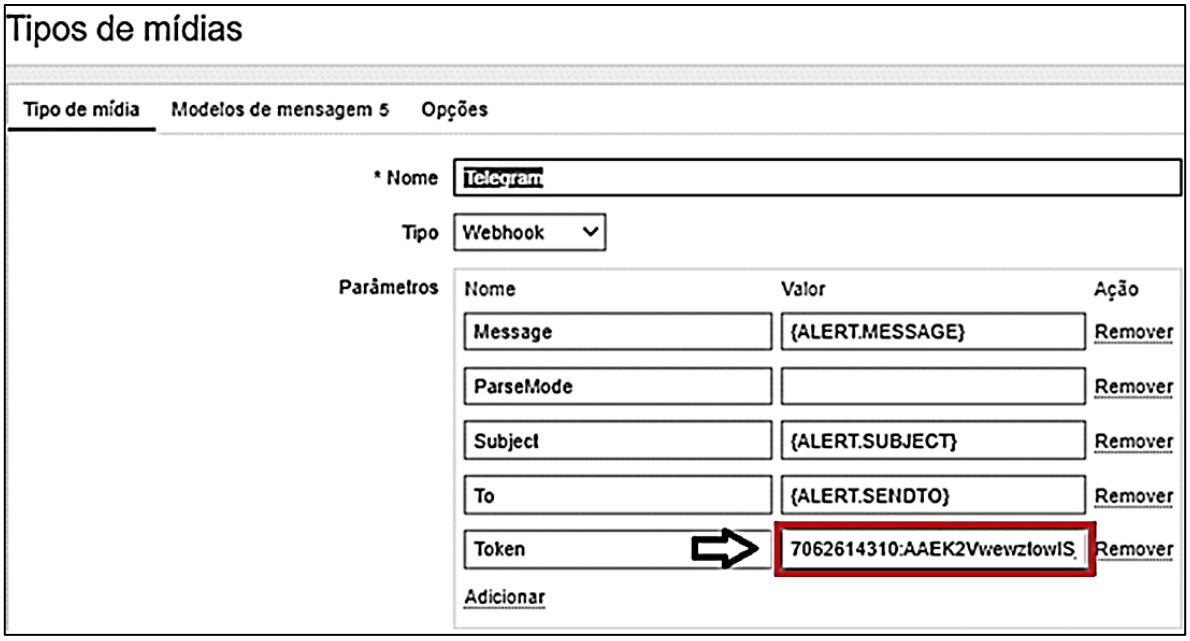
Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Após configurado o usuário, na barra de opções, foi selecionada a função “Alertas”, em seguida selecionou-se a função “Tipos de mídia”, por fim foi selecionado o Telegram e, após aberta sua janela de configuração, no campo “*Token*” foi inserido o *token* gerado ao criar o *bot*. As Figuras 34 e 35 ilustram o processo de configuração.

Figura 34: Menu de funções do Zabbix

Fonte: Tela capturada pelo autor deste trabalho do menu do Zabbix (2024).

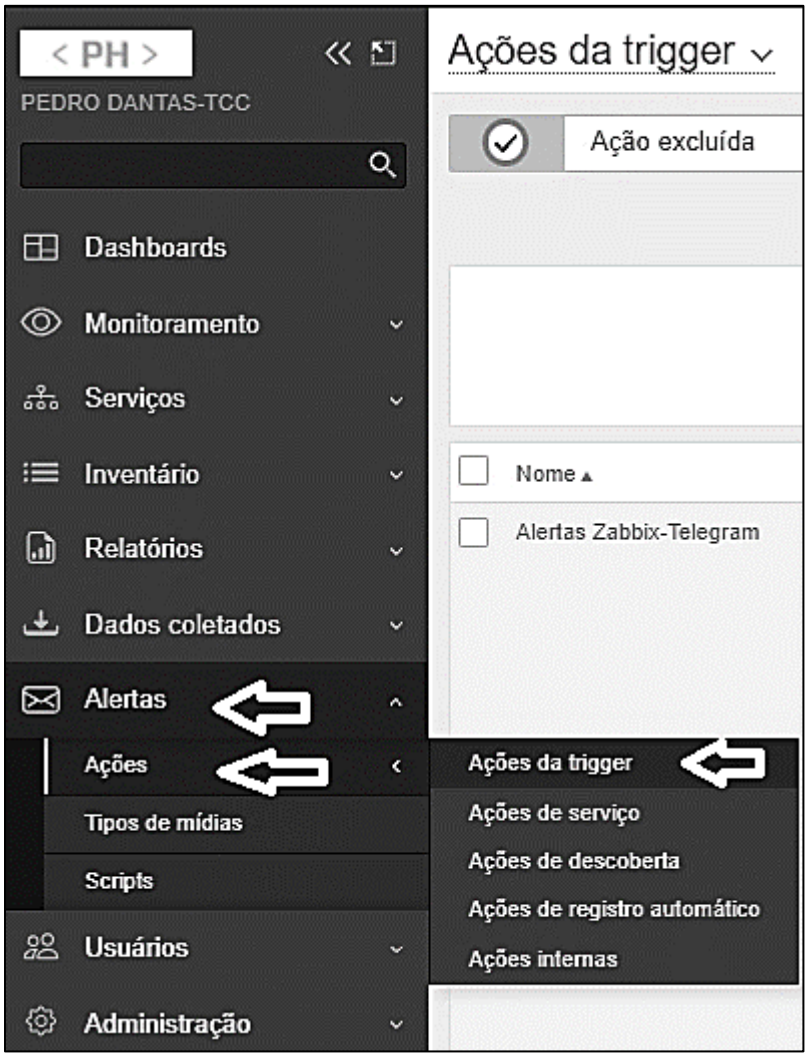
Figura 35: Tela de configuração do Telegram no Zabbix



Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Para finalizar, foi necessário fazer a configuração das *Triggers*, no menu do opções em “Alertas”, ao selecionar “Ações”, selecionou-se “Ações da *Trigger*”. No botão “Criar ação” foram definidas as triggers a serem monitoradas e configurou-se as mensagens de alerta para incidentes, recuperação e atualização de incidentes. As Figuras 36, 37, 38, 39 e 40 ilustram os processos.

Figura 36:Menu de funções do Zabbix



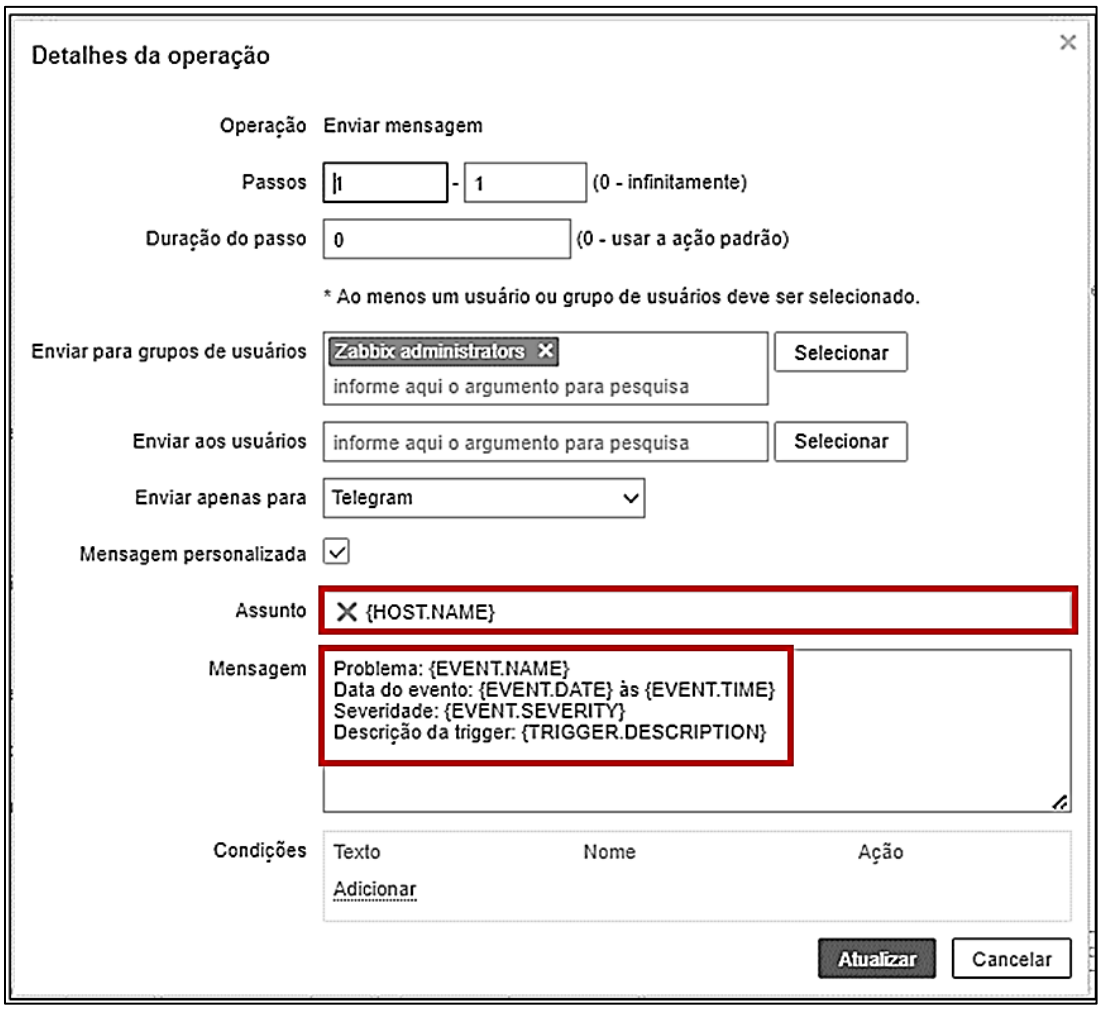
Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Figura 37: *Triggers* adicionadas para os *hosts* monitorados

Nome ▲	Condições
Alertas Zabbix-Telegram	Trigger igual *Windows by Zabbix agent: Windows: High CPU utilization* Trigger igual *Windows by Zabbix agent: Windows: High memory utilization* Trigger igual *Windows by Zabbix agent: Windows: Host has been restarted* Trigger igual *Windows by Zabbix agent: Windows: Number of free system page table entries is too low* Trigger igual *Windows by Zabbix agent: Windows: Operating system description has changed* Trigger igual *Windows by Zabbix agent: Windows: System name has changed* Trigger igual *Windows by Zabbix agent: Windows: System time is out of sync* Trigger igual *Windows by Zabbix agent: Windows: The Memory Pages/sec is too high* Trigger igual *Windows by Zabbix agent: Windows: Zabbix agent is not available* Trigger igual *SrvPelotas: (C:): Disk space is critically low* Trigger igual *SrvPelotas: (C:): Disk space is low* Trigger igual *HOST - Y: (C:): Disk space is critically low* Trigger igual *HOST - Y: (C:): Disk space is low* Trigger igual *SRV - GYN: (C:): Disk space is critically low* Trigger igual *SRV - GYN: (C:): Disk space is low* Trigger igual *SRV - GYN: Interface Realtek PCIe GbE Family Controller(Ethernet): High bandwidth usage* Trigger igual *SRV - GYN: Interface Realtek PCIe GbE Family Controller(Ethernet): Link down*

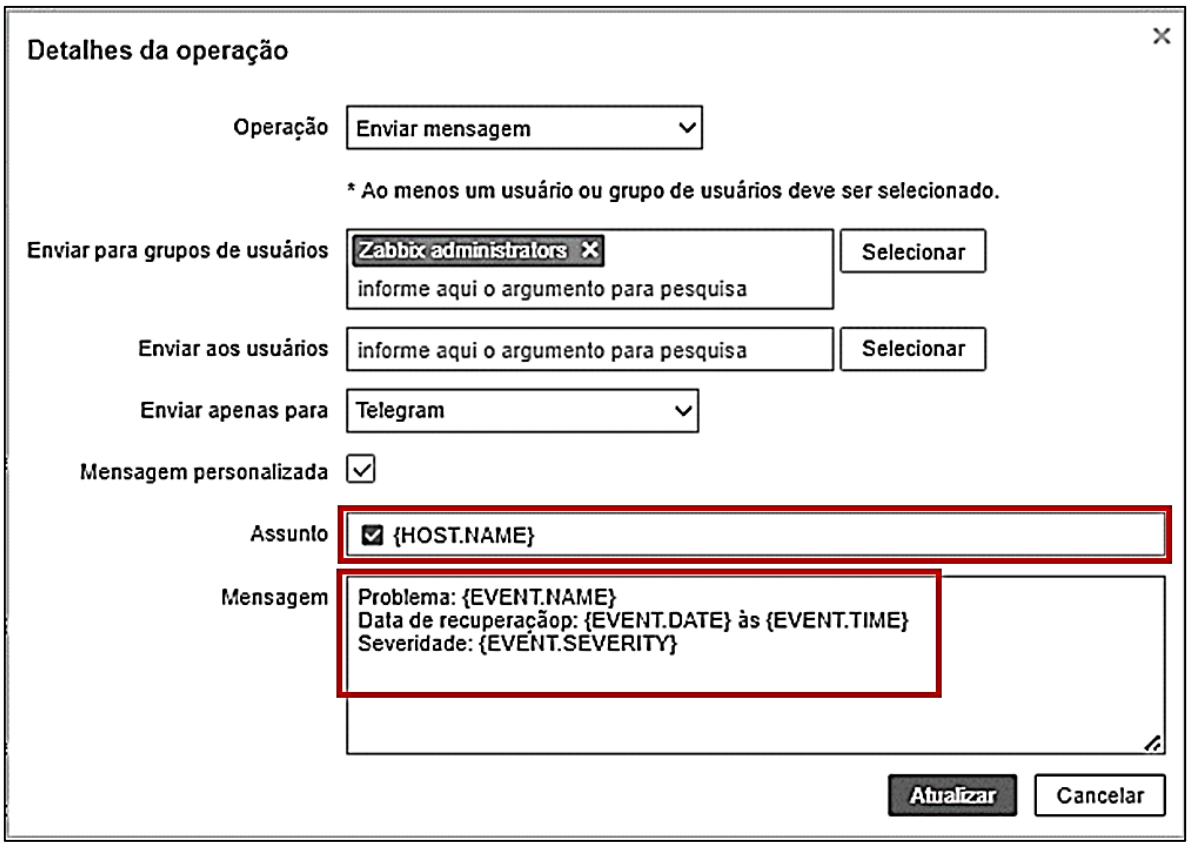
Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Figura 38:Tela de configuração de mensagem de incidente



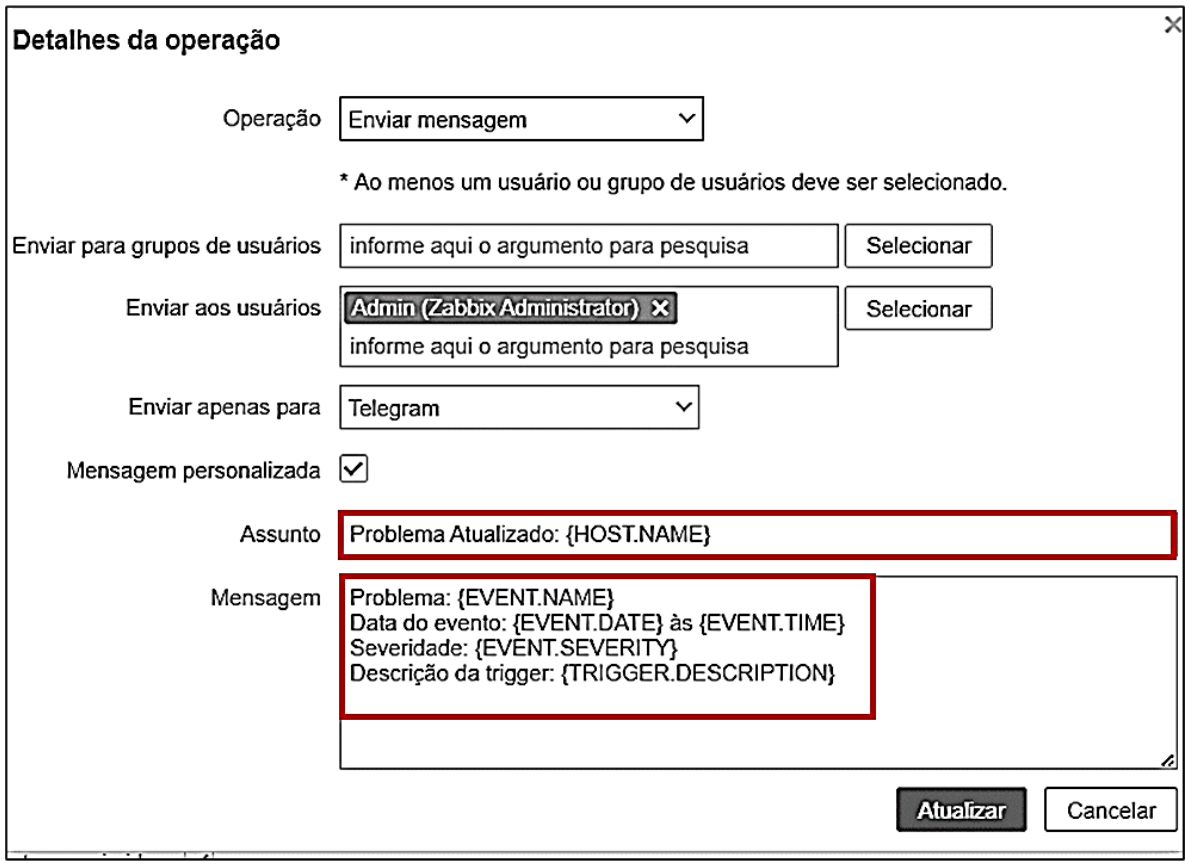
Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Figura 39: Tela de configuração de recuperação de incidente



Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

Figura 40: Tela de configuração de atualização de incidente

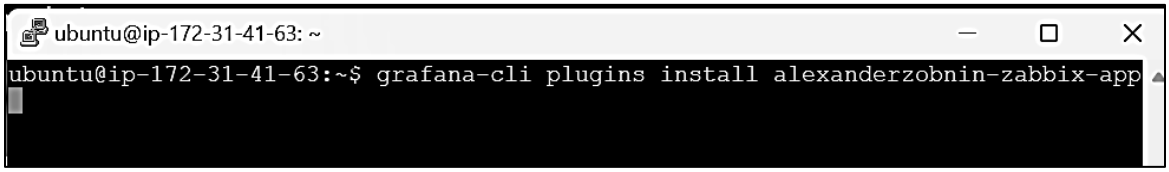


Fonte: Tela capturada pelo autor deste trabalho do Zabbix (2024).

4.3.1.4 Integração com Grafana

Para melhorar a visualização dos dados coletados, foi feita a integração entre o Zabbix e o Grafana. Para integrar, foi necessário instalar o *plugin* do Zabbix. Com a instalação feita, o acesso ao banco de dados do servidor foi liberado, e com as informações obtidas *dashboards* puderam ser criados para monitorar a *performance* do ambiente. A Figura 41 mostra o processo de instalação no Servidor do Zabbix.

Figura 41: Comando de instalação do *plugin* no Zabbix *Server*



Fonte: Tela capturada pelo autor deste trabalho (2024).

O próximo passo foi configurar o acesso ao banco de dados, informando a URL, Nome de usuário, senha e versão do Zabbix. A Figura 42 mostra os dados na *interface* do Grafana.

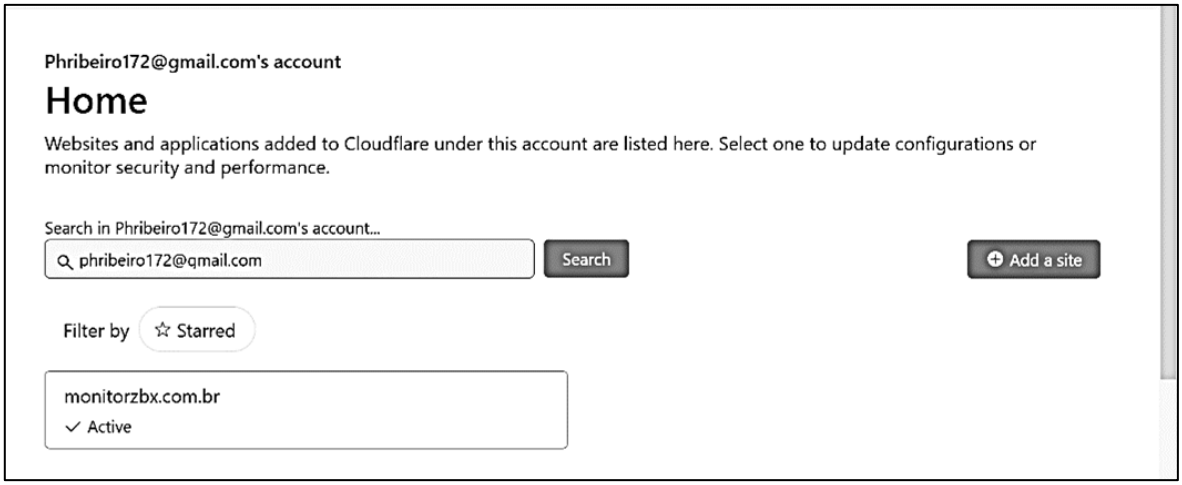
Figura 42: Tela de configuração do *plugin*

Fonte:Tela capturada pelo autor deste trabalho de Grafana (2024).

4.3.1.5 Serviço *Web*

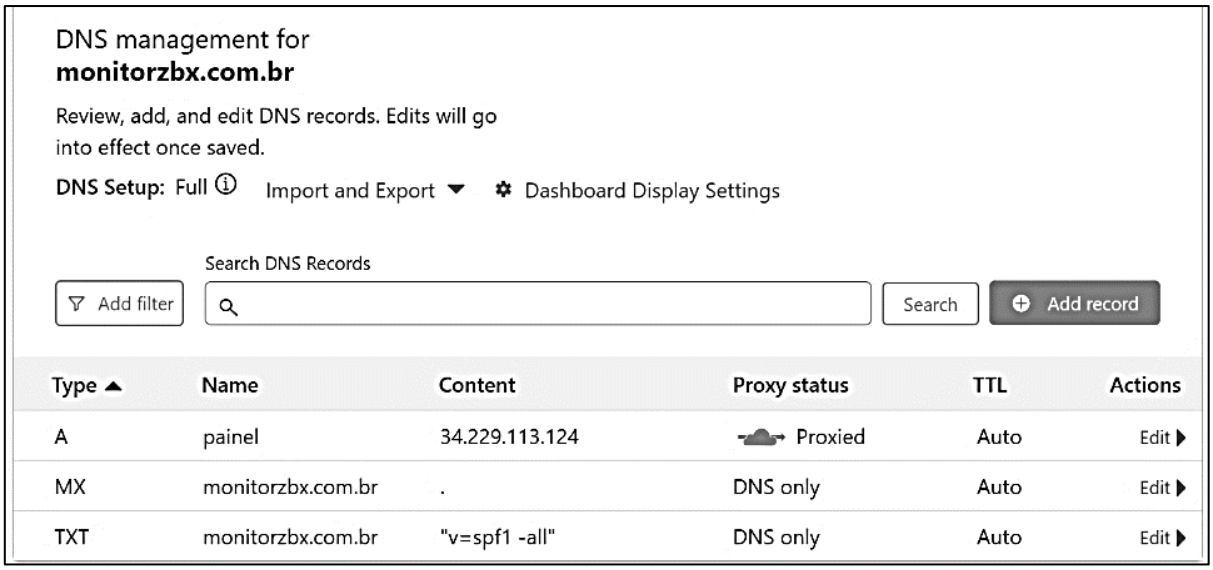
Após feita a alteração dos servidores DNS, conforme é mostrado na Figura 17, o *site* já pôde ser visualizado, ao entrar na página inicial da conta. Em seguida, foi feita a configuração de DNS dentro do Cloudflare para que, quando um acesso for solicitado seja feita a conversão do nome ao endereço IP público da instância criada na AWS. As Figuras 43 e 44 mostram o processo.

Figura 43: Página inicial da conta Cloudflare



Fonte: Tela capturada do *site* Cloudflare (2024).

Figura 44:Configuração do DNS no Cloudflare

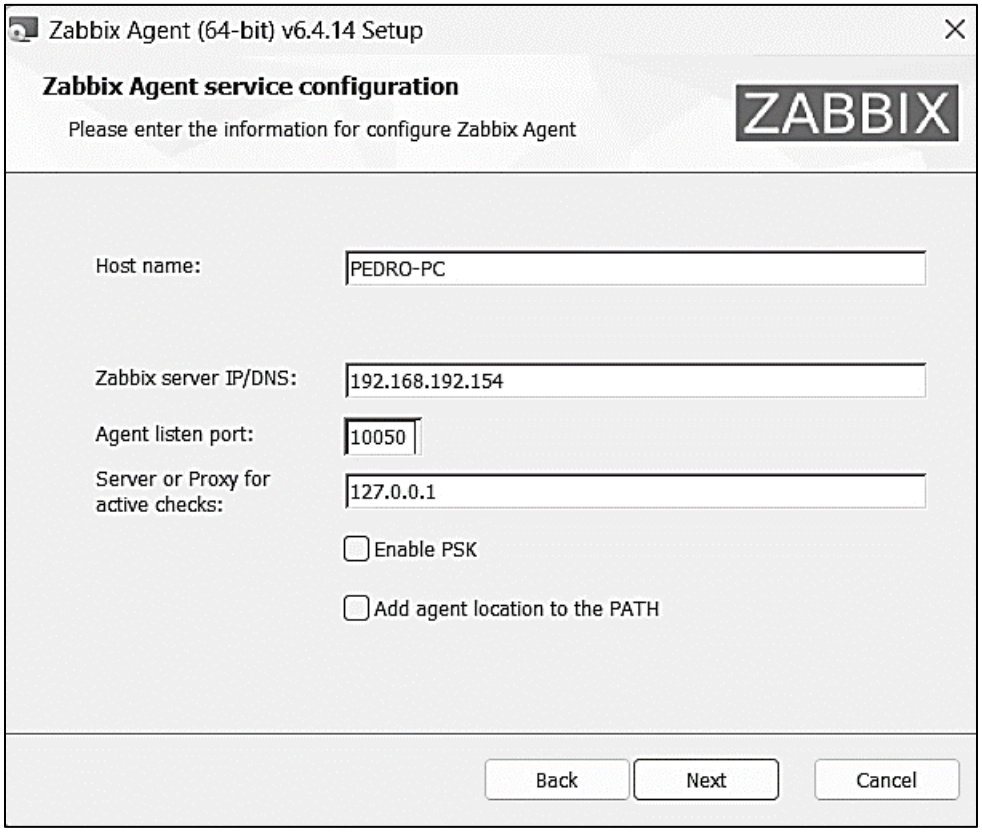


Fonte: Tela capturada do *site* Cloudflare (2024).

4.3.1.6 Zabbix Agent

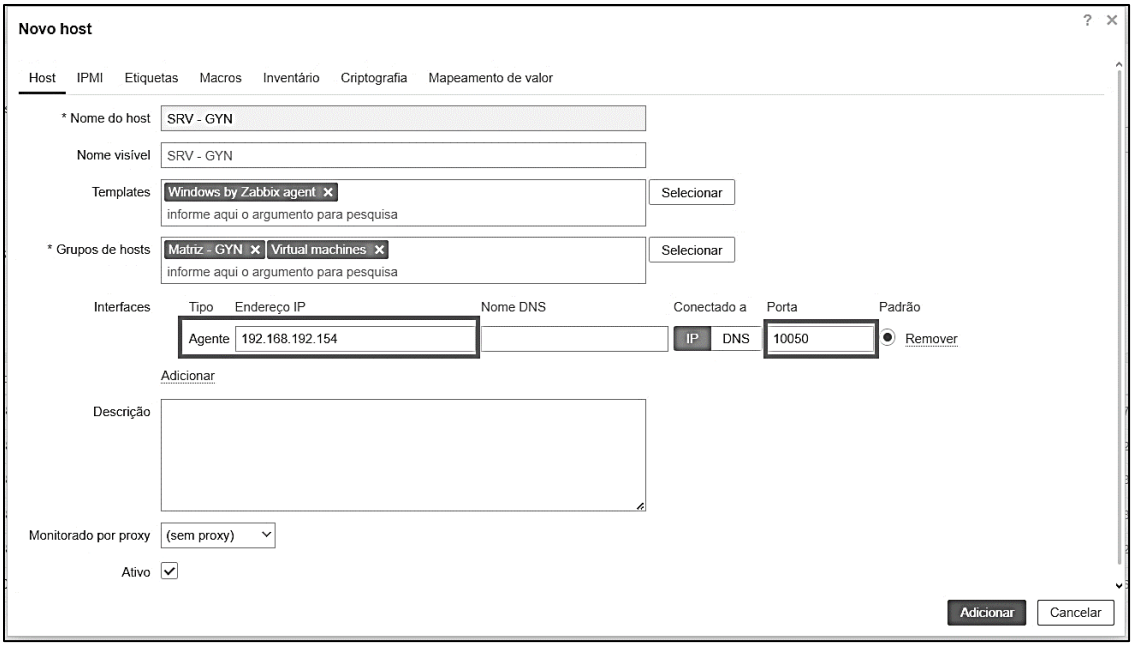
Para fazer a coleta dos dados de cada *host* monitorado, foi instalado o Zabbix Agent. No momento da instalação foi necessário colocar o endereço IP do servidor do Zabbix que faz o monitoramento do *host*. No Zabbix *Server* todos os *hosts* foram configurados de acordo com os IPs recebido pela VPN. As Figuras 45, 46, 47, 48, 49 e 50 mostram o processo de instalação e configuração dos *hosts* monitorados.

Figura 45:Tela de instalação do Zabbix Agent



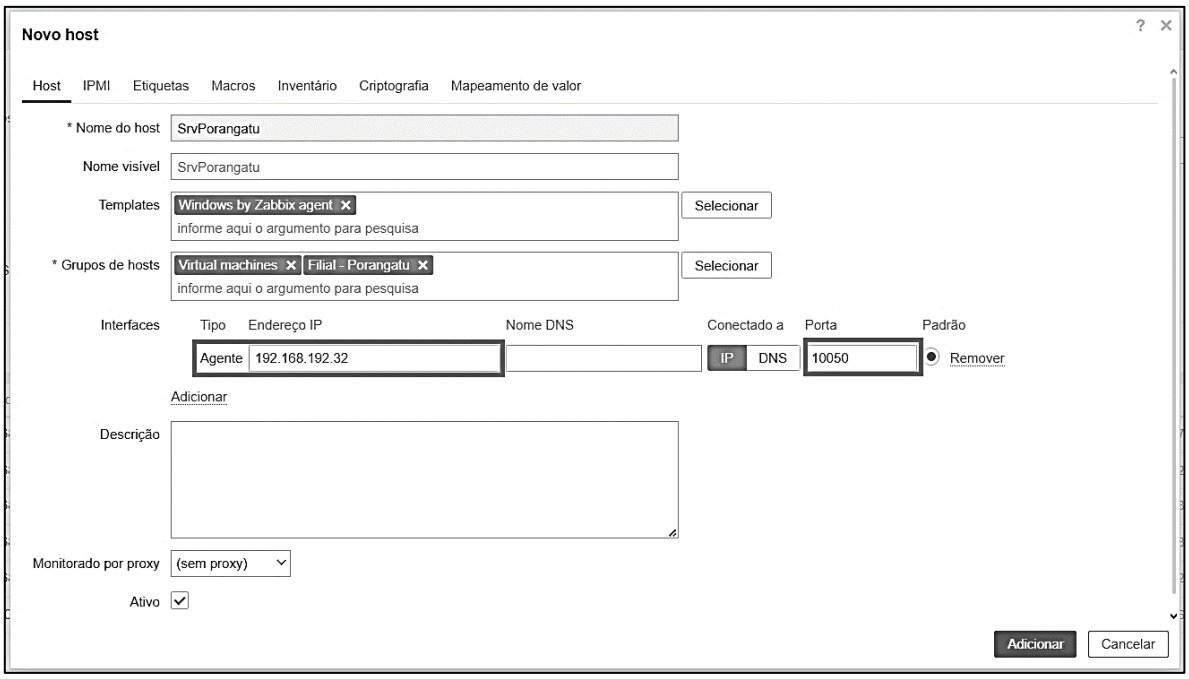
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Agent (2024).

Figura 46: Tela de configuração do *host* SRV-GYN



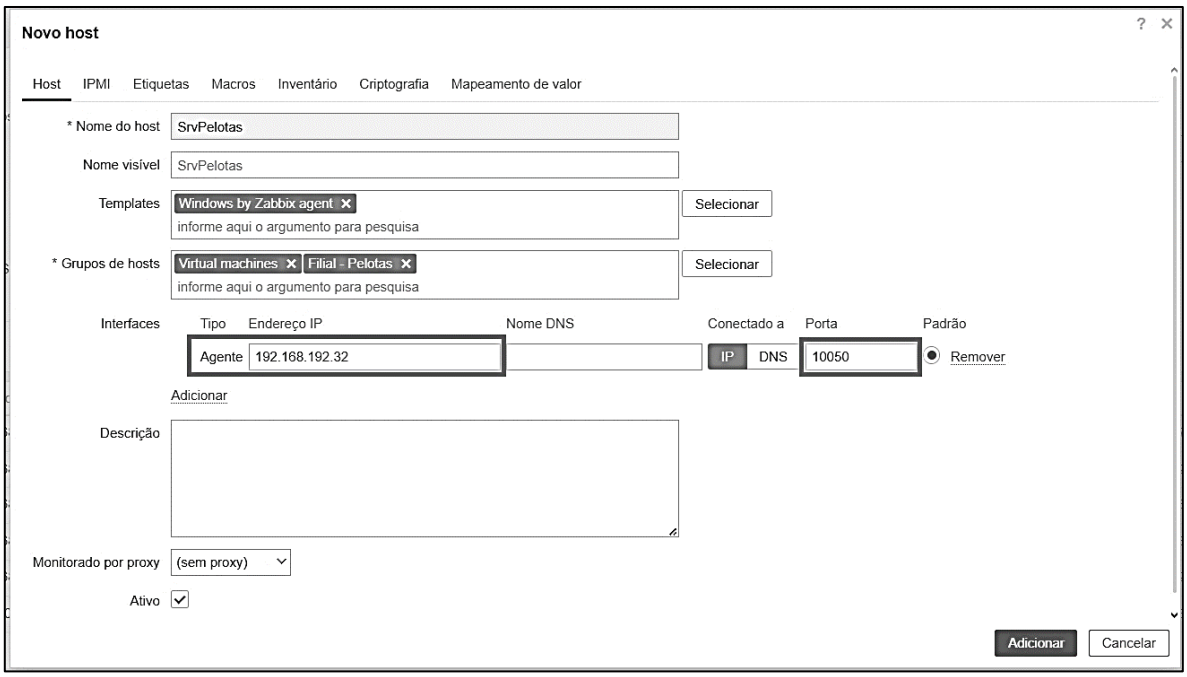
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 47: Tela de configuração do *host* SrvPorangatu



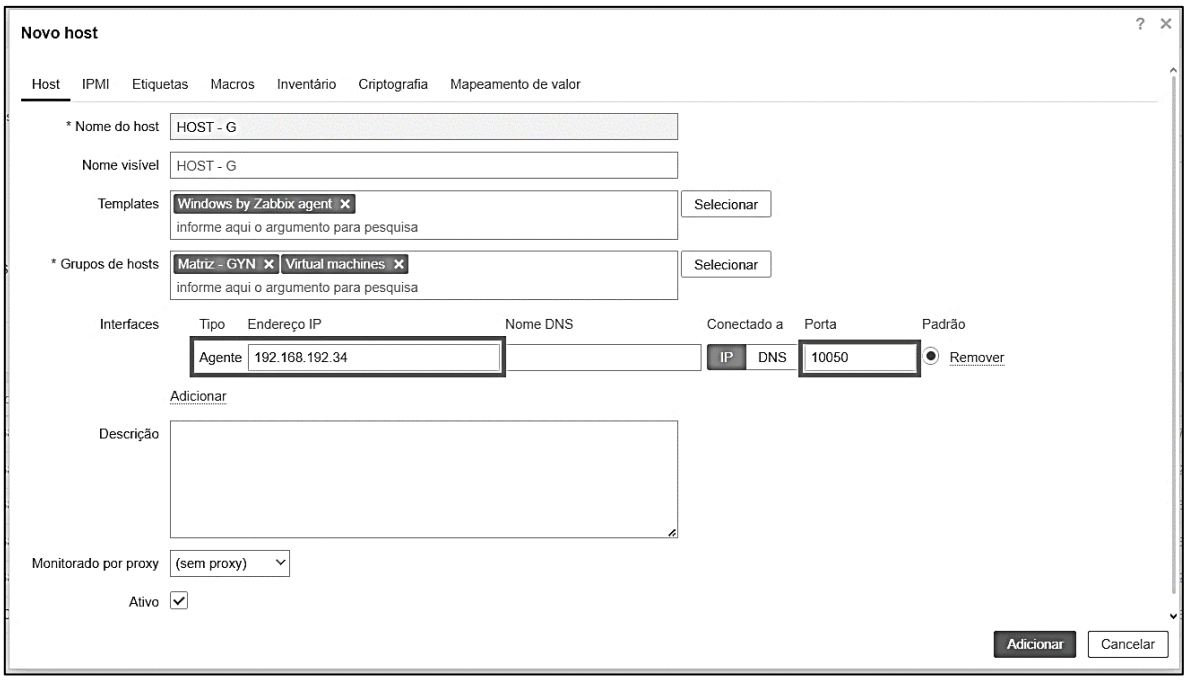
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 48: Tela de configuração do *host* SrvPelotas



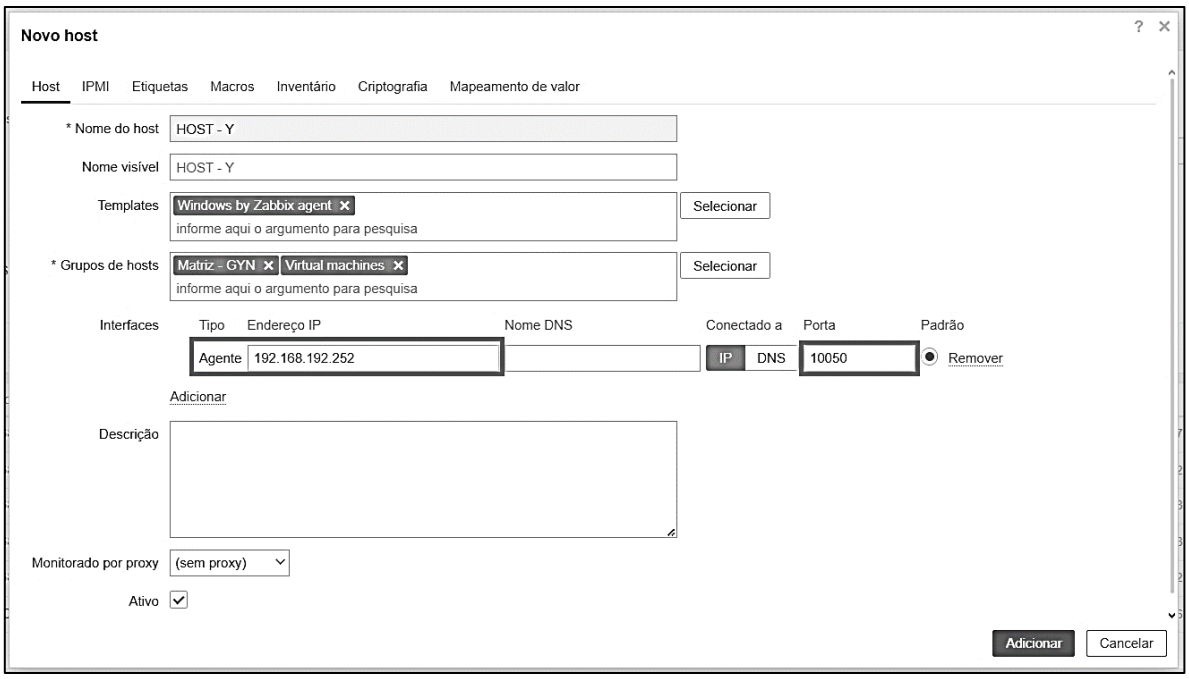
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 49: Tela de configuração do *host* G



Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

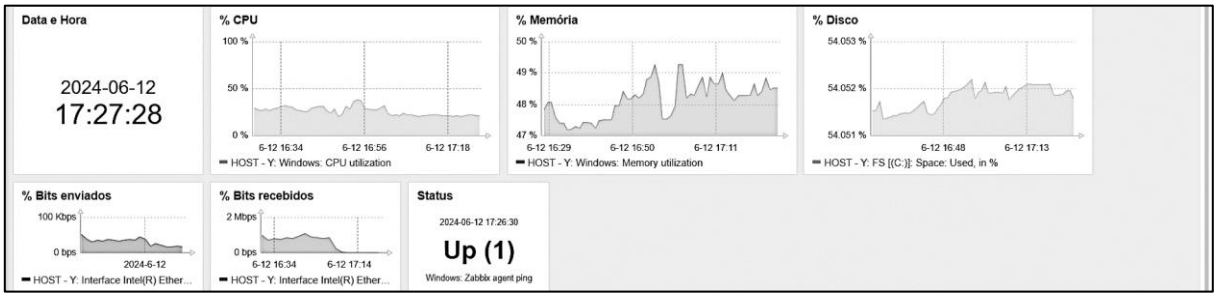
Figura 50: Tela de configuração do *host* Y



Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

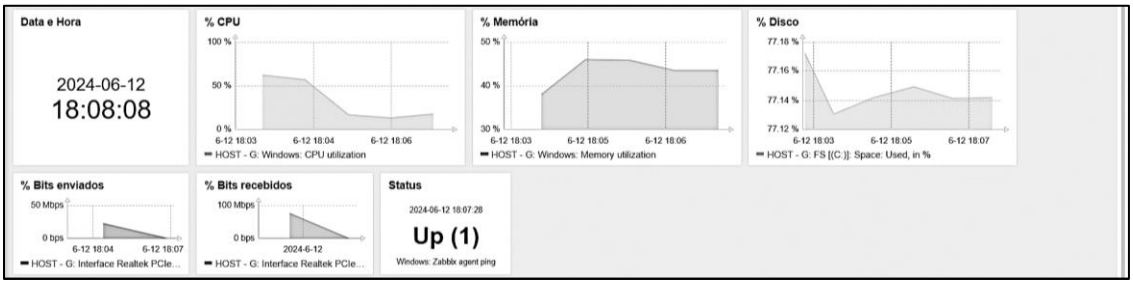
Após adicionar todos os *hosts*, foi criado *dashboards* com o objetivo exibir a coleta de dados como uso de CPU, memória RAM, rede e armazenamento de disco, que são elementos essenciais para o bom funcionamento de cada máquina. As Figuras 51, 52, 53, 54 e 55 ilustram os *dashboards*.

Figura 51: *Dashboard* do *host* Y



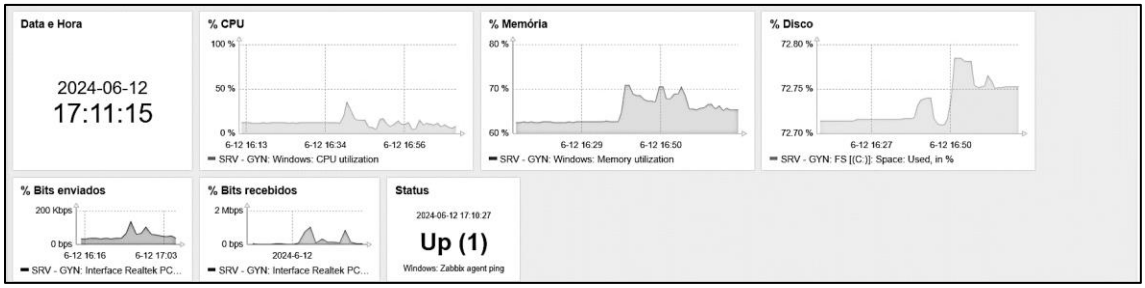
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 52: *Dashboard* do *host* G



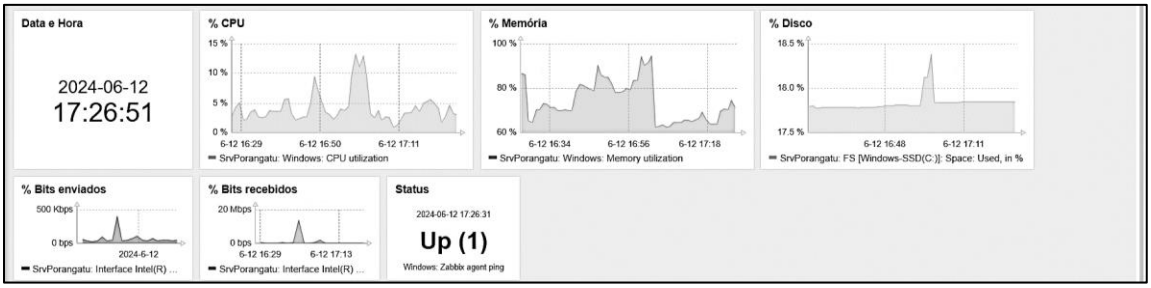
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 53: *Dashboard* do *host* SRV-GYN



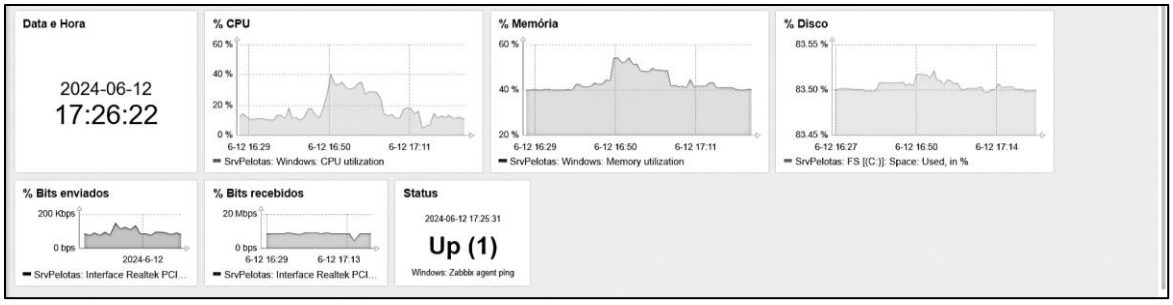
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 54: *Dashboard* do *host* SrvPorangatu



Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 55: *Dashboard* do *host* SrvPelotas



Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

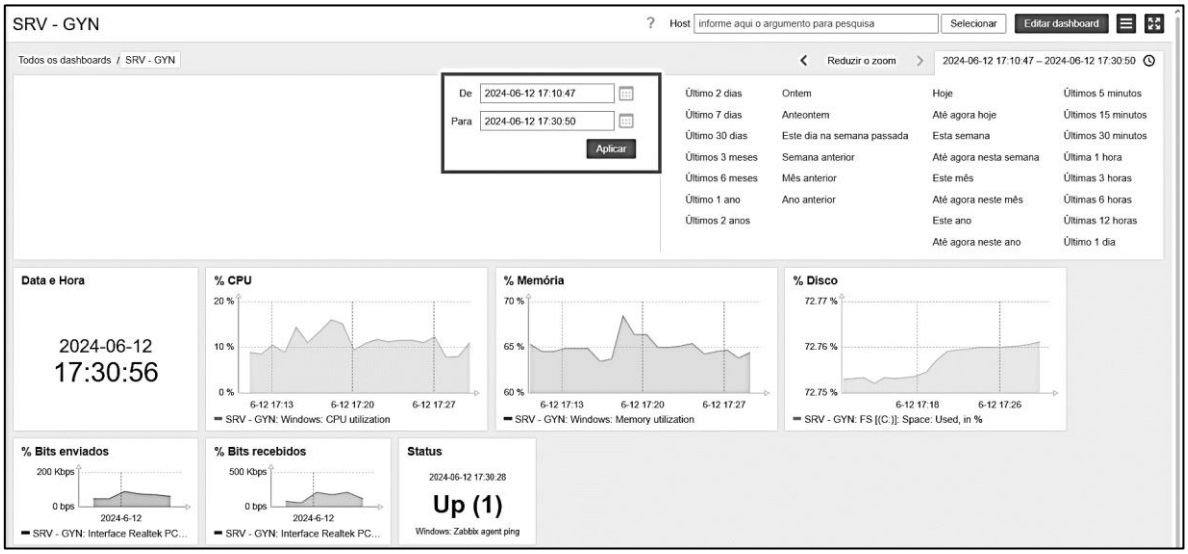
5 TESTES DO AMBIENTE E RESULTADOS

Este subtópico aborda os testes do ambiente e resultados no Zabbix, Grafana e Telegram que foram obtidos após as configurações das ferramentas mencionadas com o desenvolvimento do trabalho.

5.1.1 Zabbix

Com a configuração concluída no ambiente, foi possível monitorar todos os *hosts* através dos *dashboards* criados, através deles as coletas são exibidas nos gráficos e pode-se filtrar por período. A Figura 56 ilustra o filtro por período de um dos *hosts* monitorados.

Figura 56: Gráfico filtrado por período do *host* SRV-GYN

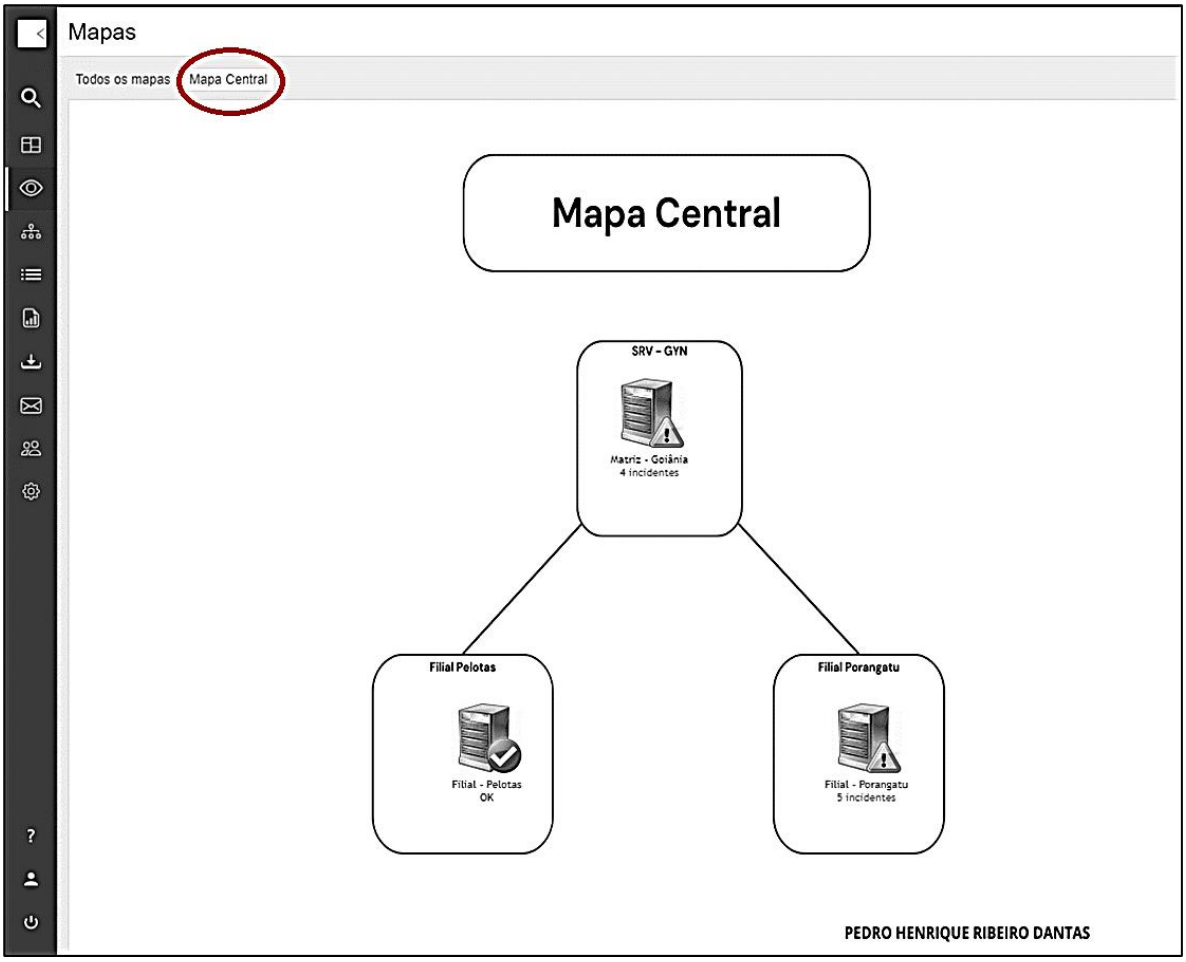


Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Outra funcionalidade utilizada foi um mapa de rede criado para a topologia do ambiente. Primeiro, foi criado um mapa central que indica cada área que o *host* está vinculado a cada ícone tem-se um submapa que mostra cada *host*. Em cada ícone foi adicionado 3 indicadores de estado da máquina. Caso esteja ocorrendo algum incidente no *host*, o ícone correspondente ao erro é simbolizado pela figura de alerta e apresenta uma legenda do erro. Caso seja mais de um, informa a quantidade de erros no *host*. Se não houver erros é

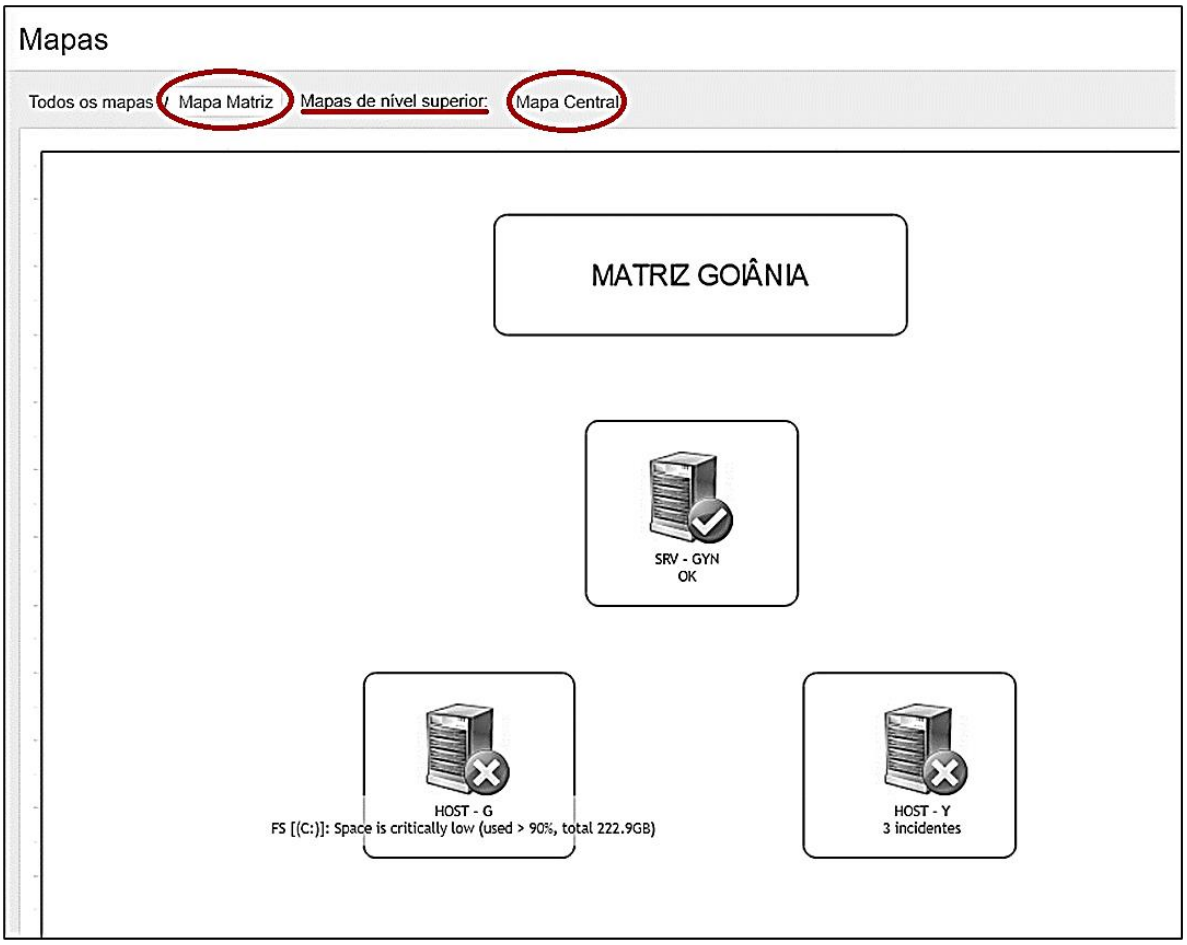
simbolizado pelo ícone positivo, conforme mostra a Figura 57. As Figuras 57, 58, 59 e 60 mostram cada mapa criado.

Figura 57: Mapa central da rede



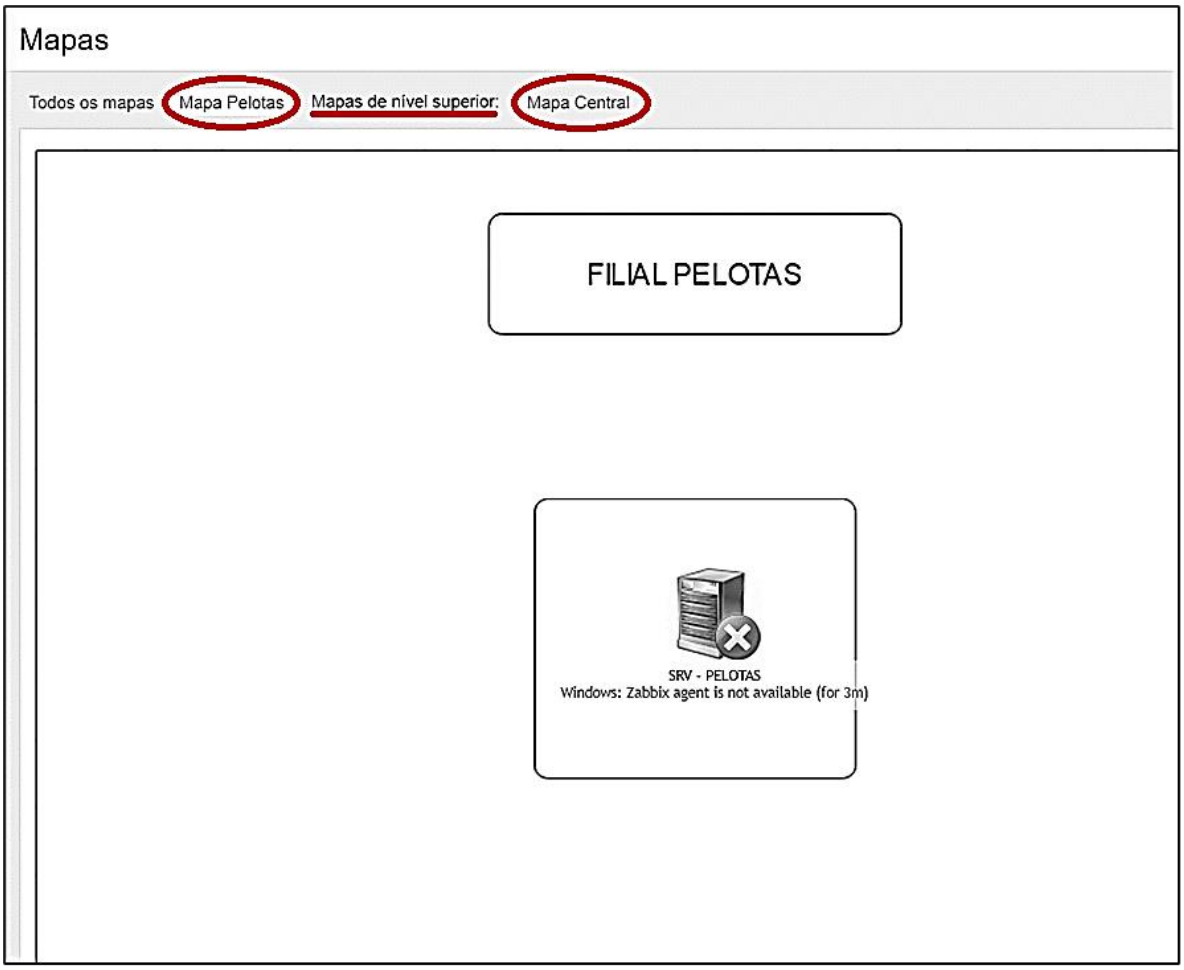
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 58: Mapa dos *hosts* localizados em Goiânia



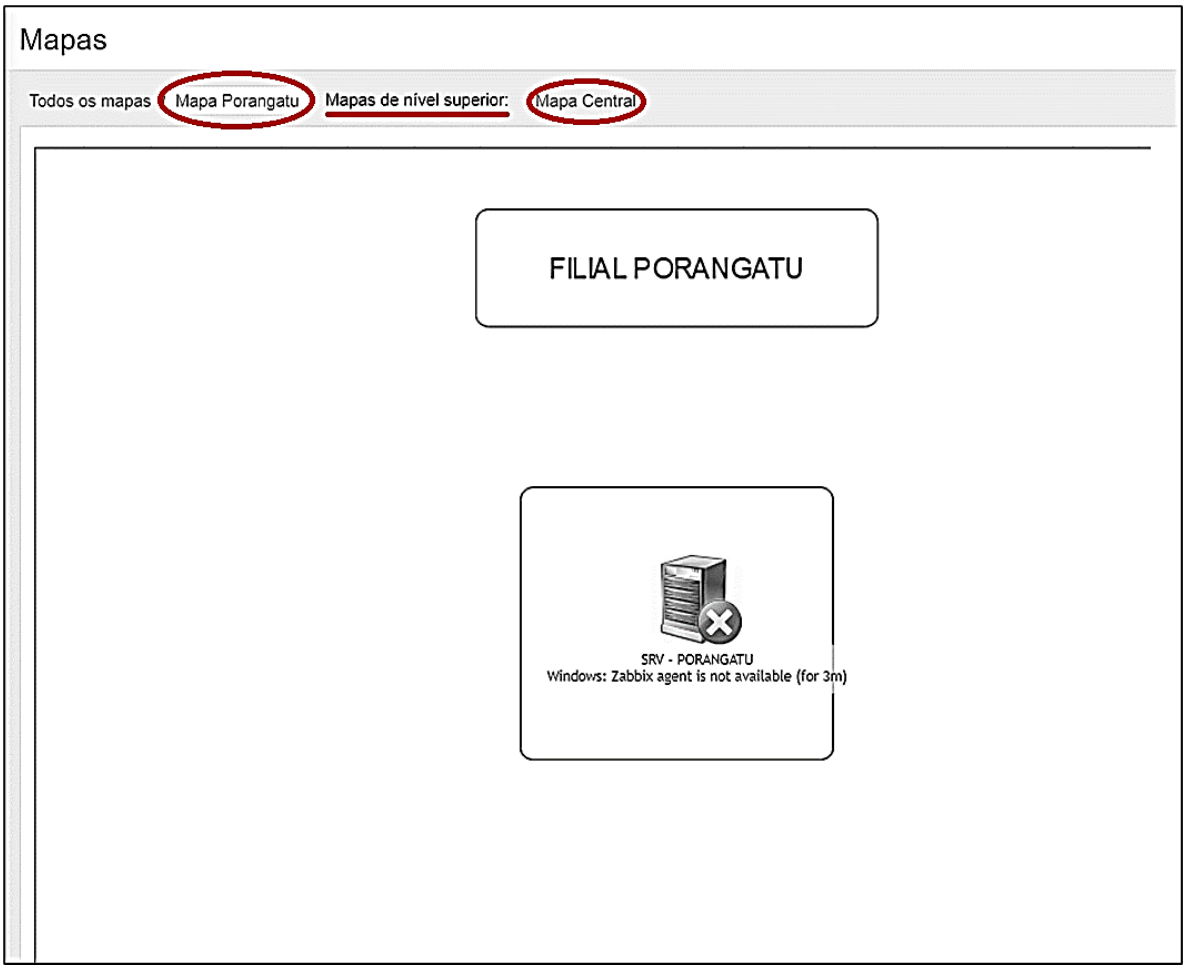
Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 59: Mapa do *host* localizado em Pelotas



Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

Figura 60: Mapa do *host* localizado em Porangatu

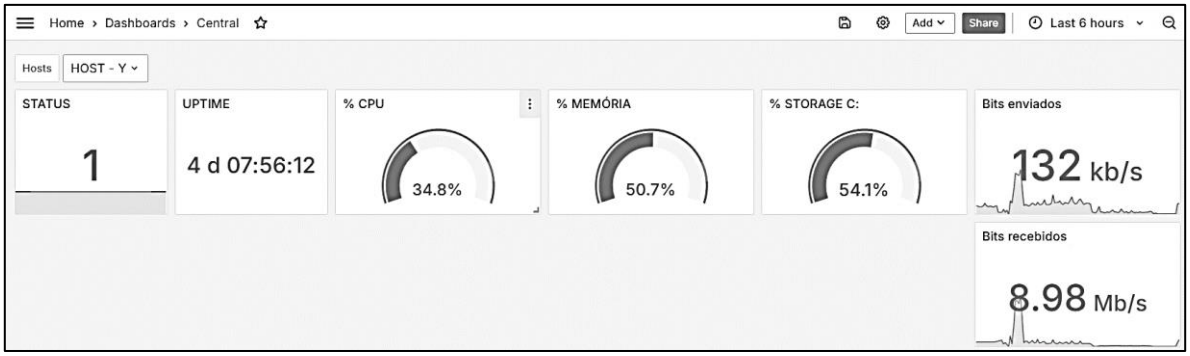


Fonte: Tela capturada pelo autor deste trabalho do Zabbix Server (2024).

5.1.2 Grafana

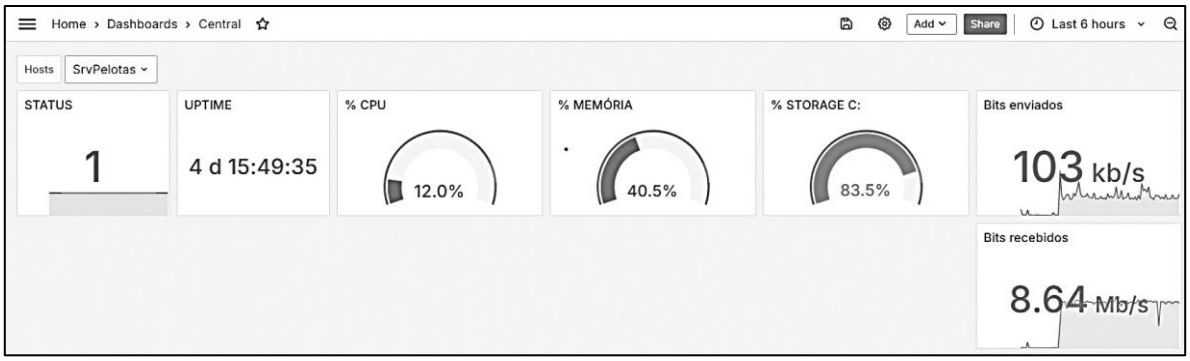
No Grafana foram criados *dashboards* mais intuitivos de cada *host*. Cada painel monitora as métricas essenciais como *status* do Zabbix Agent, Uptime, porcentagem de armazenamento de disco, uso de CPU e memória RAM e a taxa *bits* recebidos/enviados de cada máquina. As Figuras 61, 62, 63, 64 e 65 mostram os painéis que foram gerados para cada *host*.

Figura 61: Painel de monitoramento do *host* Y



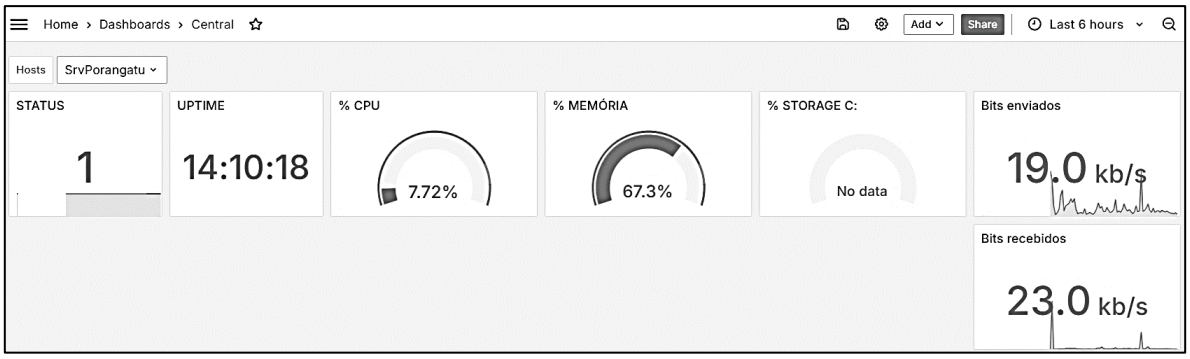
Fonte: Tela capturada pelo autor deste trabalho do Grafana (2024).

Figura 62: Painel de monitoramento do *host* SrvPelotas



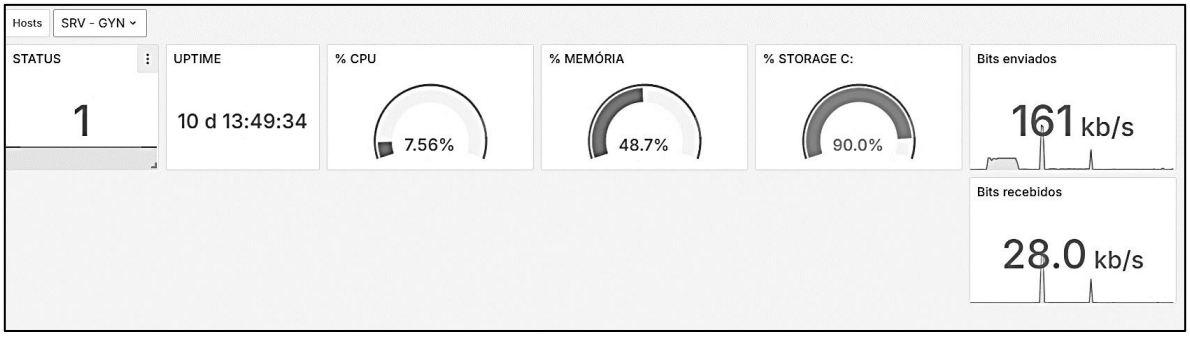
Fonte: Tela capturada pelo autor deste trabalho do Grafana (2024).

Figura 63: Painel de monitoramento do *host* SrvPorangatu



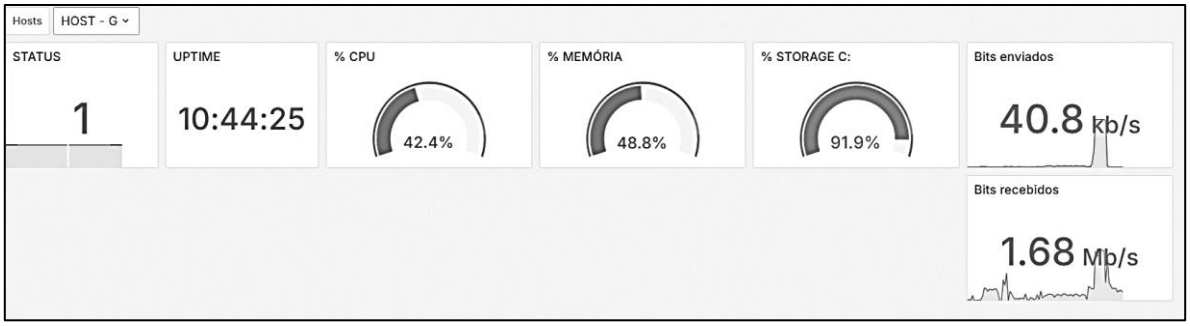
Fonte: Tela capturada pelo autor deste trabalho do Grafana (2024).

Figura 64: Painel de monitoramento do *host* SRV-GYN



Fonte: Tela capturada pelo autor deste trabalho do Grafana (2024).

Figura 65: Painel de monitoramento do *host* G

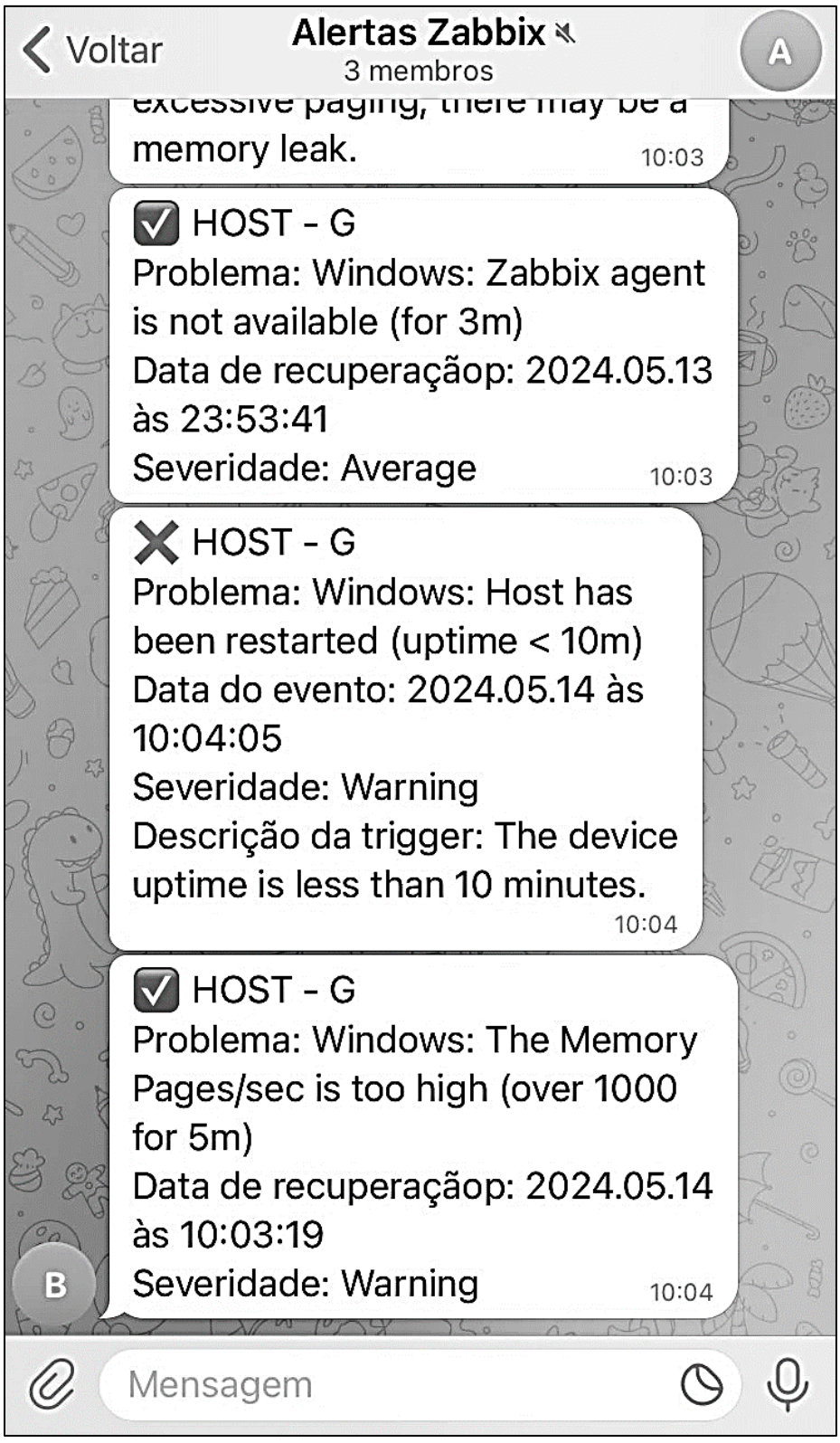


Fonte: Tela capturada pelo autor deste trabalho do Grafana (2024).

5.1.3 Telegram

No Telegram todo incidente identificado foi enviado ao grupo criado entre o usuário e *bot*. Os alertas foram configurados para enviar o nome do *host*, o problema ocorrido, a data do evento, a severidade e a descrição da *trigger* acionada. Esses alertas têm por objetivo facilitar o monitoramento do ambiente feito pelo administrador de redes. A Figura 66 mostra os alertas enviados do Zabbix para o usuário do Telegram.

Figura 66: Alertas enviados ao Telegram pelo Zabbix



Fonte: Tela capturado pelo autor deste trabalho do Telegram (2024).

5.1.4 Análise dos resultados

Durante o monitoramento contínuo da infraestrutura de TI utilizando o Zabbix, obteu-se *insights* valiosos a partir dos dados apresentados nos *dashboards*. As métricas essenciais, como utilização de CPU, memória e disco, bem como a disponibilidade do agente Zabbix e a taxa de *bits* enviados e recebidos, forneceram uma visão detalhada do desempenho dos servidores e dispositivos de rede. Essas informações foram crucialmente analisadas para garantir a operação contínua e eficiente dos sistemas.

Além das métricas básicas, a utilização de mapas no Zabbix revelou-se particularmente eficaz em cenários complexos, como o de uma matriz corporativa com várias filiais. Os mapas permitiram visualizar de forma intuitiva a interconexão entre os diferentes componentes da rede, identificando rapidamente falhas e otimizando a resolução de problemas. A capacidade de configurar alertas específicos para cada localidade através do Zabbix contribuiu significativamente para a redução do tempo de resposta a incidentes, garantindo a continuidade das operações em todas as unidades.

Ao complementar o monitoramento com o Grafana, observou-se melhorias na visualização e análise das mesmas métricas monitoradas pelo Zabbix. O Grafana oferece gráficos mais dinâmicos e personalizáveis, possibilitando uma análise mais profunda do comportamento dos recursos ao longo do tempo. Essa flexibilidade foi fundamental para entender tendências de uso, capacidade de resposta e planejamento de capacidade de maneira mais estratégica.

Por fim, a integração dos alertas do Zabbix com o Telegram mostrou-se altamente eficiente na detecção e resolução de falhas. A capacidade de receber notificações imediatas via Telegram permitiu uma resposta rápida a incidentes críticos, facilitando a intervenção proativa e minimizando o impacto operacional. Em resumo, a combinação do monitoramento detalhado pelo Zabbix, a visualização avançada pelo Grafana e a comunicação eficiente via Telegram proporcionaram um ambiente de gerenciamento de redes robusto, ágil e altamente eficaz para a organização.

6 CONSIDERAÇÕES FINAIS

O gerenciamento eficaz de redes desempenha um papel crucial na garantia da estabilidade, segurança e desempenho dos sistemas em um ambiente tecnológico em constante evolução. Este estudo, teve como foco o monitoramento detalhado de cinco hosts utilizando o servidor Zabbix, integrado ao Grafana para visualização avançada de métricas e ao Telegram para notificações instantâneas. A escolha do Zabbix como ferramenta central deve-se à sua capacidade robusta de monitorar em tempo real diversos aspectos da infraestrutura de rede, incluindo utilização de CPU, memória, disco, rede, entre outros.

A implementação dessas soluções não apenas facilitou o monitoramento contínuo dos *hosts*, mas também permitiu uma gestão proativa dos recursos de rede. Com o Grafana, pode-se criar *dashboards* personalizados que ofereceram uma visualização clara e intuitiva das métricas monitoradas pelo Zabbix, facilitando a identificação de tendências, picos de uso e potenciais gargalos de desempenho. Essa combinação de ferramentas proporcionou uma análise mais aprofundada e uma resposta mais ágil a problemas emergentes.

A integração com o Telegram também se mostrou extremamente eficaz ao fornecer alertas em tempo real sobre eventos críticos detectados pelo Zabbix. Isso permitiu uma resposta ágil a incidentes, minimizando o impacto sobre a disponibilidade e a confiabilidade dos serviços oferecidos. Além disso, a comunicação instantânea de notificações essenciais contribuiu significativamente para a eficiência operacional, garantindo uma resposta rápida e coordenada a qualquer anomalia na rede.

Durante a implementação, surgiram desafios como a configuração inicial dos agentes nos *hosts*, ajustes finos dos parâmetros de monitoramento e a otimização da *performance* geral do sistema. No entanto, esses obstáculos foram superados com um processo de resolução de problemas bem estruturado, baseado em análise cuidadosa dos dados coletados e na aplicação de ajustes específicos. A capacidade de diagnosticar e resolver esses problemas de forma eficaz demonstrou a robustez das ferramentas escolhidas e a expertise da equipe envolvida.

Os resultados obtidos têm implicações práticas significativas para a gestão de redes em ambientes corporativos. A implementação bem-sucedida do Zabbix, Grafana e Telegram não apenas melhorou a capacidade de monitoramento e resposta a incidentes, mas também

fortaleceu a infraestrutura de TI como um todo. A capacidade de antecipar problemas potenciais e implementar medidas corretivas de forma proativa não só aumenta a eficiência operacional, mas também contribui para a satisfação dos usuários finais ao garantir a disponibilidade contínua dos serviços.

6.1 Sugestões de trabalhos futuros

Para futuros trabalhos, recomenda-se explorar ainda mais as capacidades de automação e inteligência artificial nas ferramentas de monitoramento, visando aprimorar a detecção automática de anomalias e a previsão de necessidades de capacidade. Além disso, investigações adicionais podem focar na integração com novas tecnologias emergentes e na adaptação das soluções para ambientes de nuvem híbrida ou *multicloud*, ampliando assim o escopo e a eficácia do gerenciamento de redes em cenários cada vez mais complexos e dinâmicos.

REFERÊNCIAS

INMETRICS. Sumário Executivo de tecnologia. *In*: **O futuro da qualidade de *software* no Brasil**. [*S. l.*], 2023. Disponível em: https://lp.inmetrics.com.br/inscricao-estudo-tgt-pesquisa-qualidade. Acesso em: 16 nov. 2023.

INSTITUTO METRÓPOLE DIGITAL. **Aula 15 - Gerenciamento de Redes – Parte III**: Arquitetura do Zabbix. [S. l.], [s. d.]. Disponível em: https://materialpublic.imd.ufrn.br/curso/disciplina/4/57/15/8. Acesso em: 4 nov. 2023.

FLEXERA (Estados Unidos). State of the Cloud Report: Public cloud adoption is evolving. *In*: FLEXERA (Estados Unidos). **State of the Cloud Report**: Public cloud adoption is evolving. [*S. l.*], 2022. Disponível em: https://www.flexera.com/about-us/press-center/2022-state-of-the-cloud-report-by-flexera. Acesso em: 14 nov. 2023.

GRAFANA LABS. **Grafana Labs documentation: Installation**. [s. l.], [s. d.]. Disponível em: https://grafana.com/docs/plugins/alexanderzobnin-zabbix-app/latest/installation/. Acesso em: 10 abr. 2024.

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.

LOPES, Raquel Vigolvino. **Melhores Práticas para a Gerência de Redes de Computadores**. 2002. Dissertação (Pós-graduação em Informática) - Universidade Federal da Paraíba, [*S. l.*], 2002.

OPUS SOFTWARE; NIST (Estados Unidos). Instituto Nacional de Padrões e Tecnologia. Conceitos Fundamentais: O que é computação em nuvem?. *In*: OPUS SOFTWARE. **Computação em Nuvem**. 1. ed. São Paulo: Opus Software Com. e Repr. Ltda, 2015. cap. Não, p. 23-23.

SANTOS, Mauro Tapajós; TAROUCO, Liane; BERTHOLDO, Leandro; LIMA, Francisco Marcelo Marques de; VASCONCELLOS, Vanner. **Gerência de Redes de Computadores**. Rio de Janeiro: RNP/ESR, 2015.

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.

ZABBIX SIA**. Zabbix Documentation 6.4**. Disponível em: https://www.zabbix.com/documentation/current/pt/manual. Acesso em: 17 out. 2023.

ZEROTIER. **Global Decentralized Networking: About ZeroTier**. In: ZEROTIER. Global Decentralized Networking: About ZeroTier. [S. l.], 2024. Disponível em: https://www.zerotier.com/about/. Acesso em: 11 abr. 2024.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. 2ª ed. Elsevier Editora Ltda, 2014.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante PEDRO HENRIQUE RIBEIRO DANTAS do Curso de ENGENHARIA DE COMPUTAÇÃO, matrícula 20181003302697, telefone: xxx e-mail 20181003302697@pucgo.edu.br, 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 GERÊNCIA DE REDES COM ZABBIX EM AMBIENTE NUVEM, 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, 26 de junho de 2024.

Assinatura do(s) autor(es): Documento assinado digitalmente gov.br PEDRO HENRIQUE RIBEIRO DANTAS Data: 25/06/2024 10:21:26-0300 Verifique em https://validar.iti.gov.br

Nome completo do autor: PEDRO HENRIQUE RIBEIRO DANTAS

Assinatura do professor-orientador: Documento assinado digitalmente gov.br ANGELICA DA SILVA NUNES Data: 24/06/2024 16:48:02-0300 Verifique em https://validar.iti.gov.br

Nome completo do professor-orientador: ANGELICA DA SILVA NUNES