UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

JEAN MARCELO BONASSA

IMPLANTANÇÃO DE MONITORAMENTO DO AMBIENTE DE UM DATA CENTER VIA ZABBIX COM INTEGRAÇÃO PARA ENVIO DE ALERTAS VIA TELEGRAM

MONOGRAFIA DE ESPECIALIZAÇÃO

PATO BRANCO
2018

JEAN MARCELO BONASSA

IMPLANTAÇÃO DE MONITORAMENTO DO AMBIENTE DE UM DATA CENTER VIA ZABBIX COM INTEGRAÇÃO PARA ENVIO DE ALERTAS VIA TELEGRAM

Monografia de especialização apresentada ao III Curso de Especialização em Redes de Computadores – Configuração e Gerenciamento de Servidores e Equipamentos de Rede, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Dr. Fábio Favarim

PATO BRANCO
2018

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
III Curso de Especialização em Redes de Computadores

TERMO DE APROVAÇÃO

IMPLANTAÇÃO DE MONITORAMENTO DO AMBIENTE DE UM DATA CENTER VIA ZABBIX COM INTEGRAÇÃO PARA ENVIO DE ALERTAS VIA TELEGRAM

por

Jean Marcelo Bonassa

Esta monografia foi apresentada às 11h00min do dia 24 de novembro de 2018, como requisito parcial para obtenção do título de ESPECIALISTA, no III Curso de Especialização em Redes de Computadores – Configuração e Gerenciamento de Servidores e Equipamentos de Redes, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho **aprovado**.

Prof. Dr. Fábio Favarim
Orientador / UTFPR-PB

Prof. M. Eng. Anderson Luiz Fernandes
Faculdade Mater Dei

Prof. Dr. Eden Ricardo Dosciatti
UTFPR-PB

Prof. Dr. Fábio Favarim
Coordenador do III Curso de Especialização
em Redes de Computadores

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

BONASSA, Jean Marcelo. Implantação de monitoramento do ambiente de um Data Center via Zabbix com integração para envio de alertas via telegram. 2018. 94 f. Monografia (Especialização em Redes de Computadores) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

A identificação de incidentes em ativos de rede é fundamental para garantir a disponibilidade do serviço em um *Data Center*. A melhor forma para se identificar esses incidentes ou até mesmo prevê-los antes que aconteçam é através do monitoramento dessa infraestrutura. Este trabalho tem como objetivo implantar o monitoramento de toda a infraestrutura do *Data Center* com a ferramenta Zabbix e integrá-la para o envio de alertas de incidentes aos administradores de rede utilizando o aplicativo Telegram. Possibilitando a rápida notificação aos administradores da rede sobre incidentes de rede, possibilitando ações rápidas de correção para assim reduzir as indisponibilidades de serviços do *Data Center* e ter uma visão mais ampla sobre todos os aspectos da infraestrutura em questão.

Palavras-chave: Zabbix. Monitoramento. Telegram. Integração. Data Center. Incidentes.

ABSTRACT

BONASSA, Jean Marcelo. Deployment of monitoring of the Data Center environment via Zabbix with integration to send alerts via telegram. 2018. 94 f. Monografia (Especialização em Redes de Computadores) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

Identifying incidents in network assets is critical to ensuring the availability of the service in a Data Center. The best way to identify these incidents or even predict them before they happen is by monitoring that infrastructure. This work aims to implement the monitoring of the entire Data Center infrastructure with Zabbix and integrate it to send incident alerts to network administrators using the Telegram application. Enabling quick notification to network administrators of network incidents, enabling quick correction action to thereby reduce Data Center service unavailability and have a better view of all aspects of the infrastructure in question.

Keywords: Zabbix. Monitoring. Telegram. Integration. Data Center. Incidents.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE GRÁFICOS

LISTA DE SIGLAS

API	*Application Programming Interface*
IETF	*Internet Engineering Task Force*
IP	*Internet Protocol*
MIB	*Management Information Base*
NAGIOS	*Nagios Ain't Gonna Insist On Sainthood*
NGFW	*Next-Generation Firewall*
OID	*Object Identifier*
OS	*Operational System*
RFC	*Request For Comment*
SaaS	*Software as a Service*
SMS	*Short Message Service*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
SSD	*Solid State Drive*
SSH	*Secure Shell*
TI	Tecnologia da Informação
UDP	*User Datagram Protocol*
VoIP	*Voice over Internet Protocol*
VPN	*Virtual Private Network*
ZaaS	*Zenoss as a Service*

SUMÁRIO

1 INTRODUÇÃO

Nas últimas décadas a tecnologia da informação vem crescendo exponencialmente, esse crescimento acelerado faz com que cada vez mais a infraestrutura por trás dessas tecnologias se torne mais complexa e torna-se cada vez mais necessária a identificação mais eficaz de qualquer problema nessa infraestrutura, de forma a prover melhor estabilidade e disponibilidade dos serviços fornecidos por essas tecnologias.

O monitoramento em tempo real da infraestrutura e seus ativos (*switches*, roteadores, servidores, *storages*) torna-se indispensável, para que se possa obter, de forma rápida, precisa e confiável, dados e informações sobre incidentes relacionados a esses ativos, tornando muito mais eficaz a resposta e possibilitando que o administrador da infraestrutura tenha uma visão geral sobre a alocação de recursos e necessidade de expansão da rede (BENINI; DAIBERT, 2011).

Tal monitoramento se torna indispensável para *Data Centers* pois estes provêm infraestrutura como serviço e, um dos principais fatores que determinam a confiabilidade de um *Data Center,* é sua disponibilidade.

Desta forma, um dos fatores mais importantes para um *Data Center,* é possuir um monitoramento eficaz, que monitore todos os ativos essenciais e serviços vitais para o correto funcionamento do ambiente, garantindo assim a notificação imediata de incidentes para os administradores da infraestrutura, e até mesmo para a tomada de ações corretivas automatizadas, configuradas no próprio serviço de monitoramento.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Implantar o monitoramento da infraestrutura de redes e servidores de um *Data Center* utilizando a ferramenta de monitoramento Zabbix e sua integração com o aplicativo Telegram para envio de alertas aos administradores da rede.

1.1.2 Objetivos Específicos

- Reduzir o tempo de detecção de incidentes e anomalias;

- Monitorar toda a infraestrutura de redes de forma a identificar possíveis incidentes ainda antes de acontecerem;
- Possibilitar a reação rápida para correção de problemas que sejam detectados pela ferramenta através de notificações personalizadas a todos os envolvidos;
- Prover melhor eficiência e disponibilidade no serviço prestado pelo *Data Center* através do monitoramento personalizado para o cenário em questão;
- Reduzir a quantidade de suporte necessário a clientes devido a incidentes e anomalias de ativos de rede do *Data Center*.

1.2 JUSTIFICATIVA

Prover um monitoramento completo do ambiente de um *Data Center* de forma a reduzir o suporte necessário a clientes que sejam impactados por algum incidente ou anomalia de ativos de rede do *Data Center*, reduzindo também custos financeiros gerados por esses incidentes, como por exemplo, um servidor que esteja sofrendo com problemas de aquecimento só seria identificado quando o equipamento parasse de funcionar, sem o monitoramento.

Junto deste monitoramento e, tão importante quanto, é necessário que as pessoas envolvidas sejam notificadas, correta e rapidamente, sobre os incidentes e anomalias, para que as correções necessárias sejam executadas de forma mais rápida e eficiente.

Uma das grandes vantagens desse monitoramento proposto, é que a ferramenta Zabbix, possui código aberto, sendo possível personalizar e integrar a outras ferramentas, como por exemplo a ferramenta Telegram, para envio de alertas via mensagens de texto.

O uso de uma ferramenta de código aberto, que além de reduzir custos, é vantajoso pela ampla documentação sobre a ferramenta em fóruns. Como é uma ferramenta muito utilizada em todo o mundo possui *templates* de monitoramento (conjunto de configurações de componentes que podem ser monitorados) já criados para praticamente todos tipos de ativos de rede que se queira monitorar. Caso não exista é possível a criação de *templates* personalizados. Além disso, a ferramenta poder ser personalizada por completa para atender os requisitos da organização.

O presente trabalho, além de resolver os problemas acima elencados, visa fornecer um estudo de caso sobre a implantação de uma ferramenta de monitoramento de ativos de rede de código aberto no ambiente de um *Data Center*. Por abranger vários aspectos da utilização da

ferramenta pode ser utilizado como base de conhecimento para a futura implantação desta ferramenta em outros ambientes empresariais. Bem como servir de referencial para estudos sobre ferramentas para monitoramento e administração de redes empresariais.

1.3 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos, dos quais este é o primeiro e apresenta a o contexto do trabalho, incluindo os objetivos e a justificativa para a realização do mesmo.

O Capítulo 2 apresenta o referencial teórico. O referencial teórico apresenta os principais conceitos relacionados a gerenciamento de redes, assim como apresenta algumas ferramentas destinadas ao monitoramento de redes.

No Capítulo 3 são apresentados os materiais (softwares e equipamentos) utilizados, assim como a metodologia empregada no desenvolvimento deste trabalho.

O Capítulo 4 contém o resultado do desenvolvimento deste trabalho.

No Capítulo 5 são apresentadas as considerações finais do trabalho.

2 REFERENCIAL TEÓRICO

2.1 ATIVOS DE REDES

Segundo Agasus (2018), Ativos de rede são equipamentos específicos que permitem estruturar uma rede de computadores, conectando as máquinas da empresa umas às outras e também conectando a organização à internet. Ativos de redes, podem ser vistos como todos os equipamentos de uma rede de computadores que recebem energia, e seu funcionamento impacta diretamente no correto funcionamento do ambiente de uma rede de computadores. Vários equipamentos podem ser considerados ativos de rede como por exemplo: Servidores, *storages*, roteadores, *switches* e *firewalls*.

2.1.1 Servidores

Servidores podem ser divididos em 2 categorias distintas, servidores físicos e servidores virtuais. Os servidores físicos, como o nome já diz, são servidores que existem fisicamente, São *hardwares* de alta capacidade de processamento e de memória, capazes de rodar um sistema operacional comum como, por exemplo, um Microsoft Windows Server, para ser usado de maneira convencional, dedicado apenas para um propósito. O que difere um servidor de um computador normal, são principalmente suas características de desempenho e principalmente de confiabilidade e disponibilidade. Também pode ser instalado em um servidor físico uma virtualização, como, por exemplo, um *Hypervisor* VMWare, e dentro desse servidor rodar diversos servidores virtuais, estes servidores virtuais utilizam os recursos de hardware do servidor físico de forma compartilhada. A Figura 1 apresenta um exemplo de servidores físicos.

Figura 1 - Exemplo de servidores físicos (Dell PowerEdge R730)

Segundo Lucena (2016), "Quando falamos em virtualização de servidores estamos falando em dividir os recursos de um hardware (servidor físico) em diversos servidores virtuais que podem ser usados com finalidades diferentes."

Um servidor virtual, ou virtualizado, é um servidor que não possui um hardware físico especificamente dele, é um servidor criado dentro de um *Hypervisor*, instalado em um servidor físico. Um único servidor físico pode conter diversos servidores virtualizados dentro dele, mas não ultrapassando seu limite de hardware físico.

"O *Hypervisor* simula, em cada divisão do hardware, uma máquina virtual. Cada máquina virtual possui capacidades diferentes conforme suas próprias necessidades. Assim, uma delas pode ter mais memória, a outra mais espaço em disco e outra mais do processador." (LUCENA, 2016).

Um exemplo que pode ser usado para demonstrar como se relacionam servidores físicos e virtuais é o uso de diversos servidores virtuais com sistemas operacionais distintos em um *Hypervisor* VMware, instalado em um servidor físico de alta capacidade de processamento e memória. A Figura 2 demonstra uma virtualização VMWare com vários servidores Virtuais distintos instalada em um servidor físico Dell PowerEdge R730. Por propósitos de segurança da informação, nomes e endereços foram ocultados na figura.

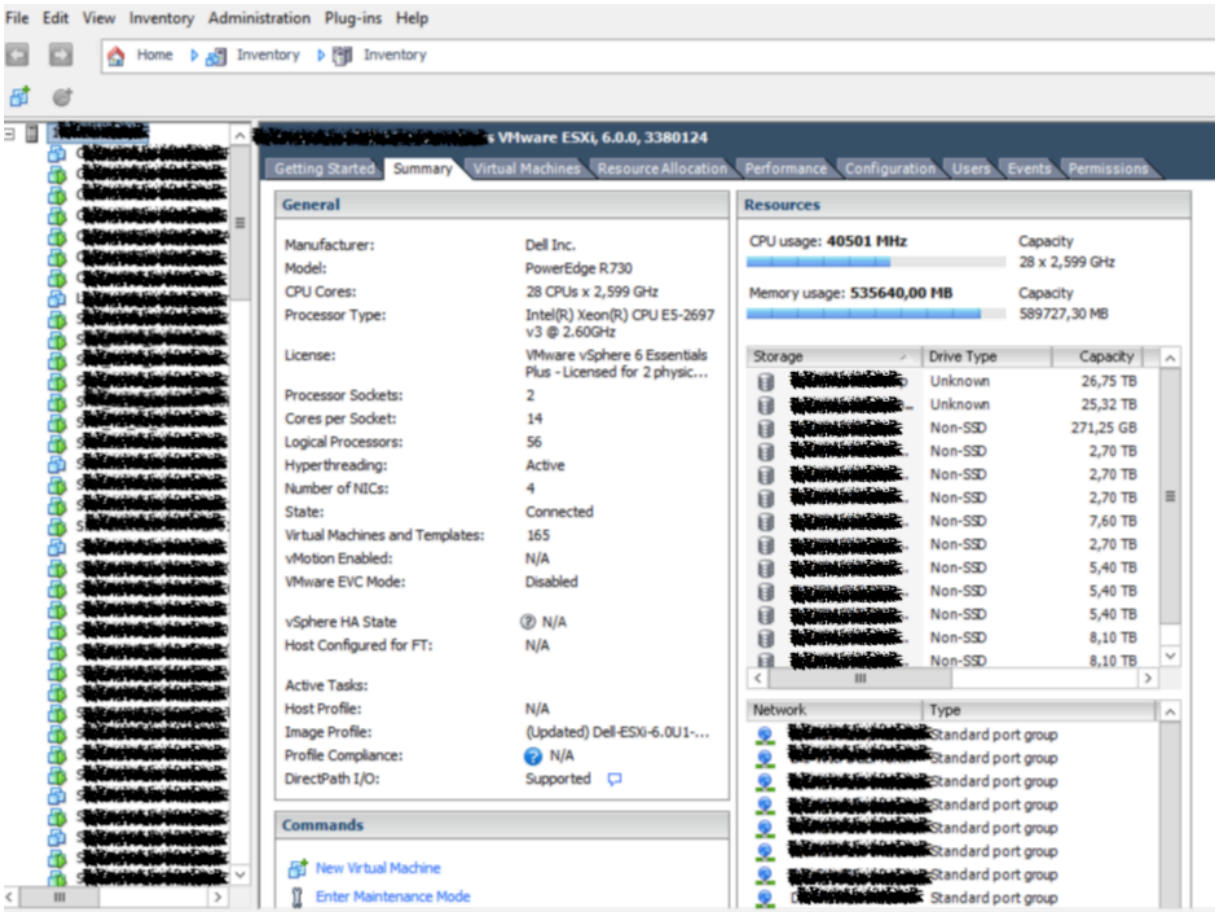


Figura 2 - Exemplo de uma virtualização VMWare rodando em um servidor fisico DELL PowerEdge R730

Na Figura 2 também é possível notar as informações de hardware do servidor físico, e o número de servidores virtuais que estão instanciados dentro desse servidor, no exemplo em questão, o servidor possui 56 processadores lógicos, e 575.90 gigabytes de memória RAM, e 165 servidores virtuais.

2.1.2 Storages

Storages são servidores dedicados especificamente para o armazenamento de dados, normalmente são utilizados em conjunto com servidores *Hypervisors* que possuem alta capacidade de processamento e memória, porém não possuem uma capacidade de armazenamento equivalente, Já os *storages* normalmente possuem alta capacidade de armazenamento e menos recursos de processamento e de memória, esses que por sua vez são destinados apenas para o uso de seu sistema operacional, e gerenciamento de discos.

> Storages são dispositivos projetados especificamente para armazenamento de dados, onde através de uma conexão via rede, você pode conectar seu(s) servidor(es) a um storage, facilitando assim a

> expansão da capacidade de armazenamento sem impacto na produção, garantindo maior flexibilidade e confiabilidade no armazenamento (LOCAWEB, 2017).

A Figura 3 apresenta uma *Storage* IBM Storwize V3700, em um ambiente de produção.

Figura 3 - Exemplo de uma Storage (IBM Storwize V3700)

Na Figura 3 a *storage* em questão armazena aproximadamente 35 terabytes de dados, de mais de 500 servidores virtuais, e possui 72 discos, separados em vários arranjos de discos (RAID).

2.1.3 Roteadores

Um roteador é um equipamento de redes de computadores que serve para interconectar múltiplas redes. O roteador faz o encaminhamento dos pacotes enviados pela rede para seu destino adequado. Cada pacote recebido pelo roteador tem um endereço de destino, o qual é utilizado pelo roteador para encaminhar para a rede de destino, de acordo com regras pré-definidas. Basicamente, um roteador funciona como um mapa da rede, direcionando por onde cada pacote deve seguir para chegar ao destino desejado. A Figura 4 apresenta exemplo de roteadores.

Figura 4 - Exemplo de Roteadores (CISCO 800 Series)

Os roteadores apresentados na Figura 4 são utilizados para a comunicação via VPN entre mais de cinco mil lojas, de uma grande rede internacional de supermercados. Apenas um desses quatro roteadores fica ativo, outros dois estão como redundância em caso de falha do primeiro, e o último roteador é utilizado para a comunicação de um ambiente de testes do mesmo cliente. Dessas mais de cinco mil lojas pelo mundo, aproximadamente 400 delas que ficam no Brasil utilizam os serviços do *Data Center* em questão, esses equipamentos se encontram na sala de conectorização do *Data Center*.

2.1.4 Switches

Um *switch* é um equipamento utilizado para expandir a capacidade de portas para as conexões em uma rede de computadores, semelhantemente a um *hub*, porém, um *switch* não é apenas uma extensão de portas de rede, um *switch* é capaz de transmitir vários pacotes

simultaneamente através de suas portas, enquanto um *hub* possui a capacidade de apenas um pacote por vez.

> Os *switches* são aparelhos bastante semelhantes aos *hubs*, tendo como principal diferença a forma como transmitem dados entre os computadores. Enquanto *hubs* reúnem o tráfego em somente uma via, um *switch* cria uma série de canais exclusivos em que os dados do computador de origem são recebidos somente pela máquina destino. Com isso, a rede não fica mais congestionada com o fluxo de informações e é possível estabelecer uma série de conexões paralelas sem nenhum problema. (GUGELMIN, 2011).

Existem também *switches* gerenciáveis que também podem exercer as funções de um roteador, também conhecidos como *switches* de camada 3, realizando o roteamento de uma rede diretamente no *switch*, sem a necessidade de um outro aparelho. A Figura 5 apresenta um exemplo de um desses *switches* que possui funções de roteamento.

Figura 5 - Exemplo de um switch de camada 3, modelo HP A5120-48G.

2.1.5 Firewalls

Um *firewall* é um equipamento utilizado como uma barreira de segurança, entre uma rede interna e o restante da Internet, um *firewall* monitora todo o tráfego de entrada e saída de uma rede de computadores e decide, o que passa e o que é bloqueado, conforme suas politicas pré-definidas.

Atualmente os *firewalls* vem se modernizando de forma a combater as novas ameaças que surgem a cada dia na Internet, com isso as novas tecnologias levaram ao surgimento do que é conhecido como *Next-Generation Firewall* (NGFW) ou *Firewalls* de próxima geração, que são capazes de combater ativamente ameaças de *malwares* avançados e ataques em nível de aplicação. Segundo Cisco (2018) "Os *firewalls* evoluíram para além da simples filtragem de pacotes e inspeção *stateful*. A maioria das empresas está implantando *firewall* de próxima geração para bloquear ameaças modernas, como *malware* avançado e ataques na camada da aplicação." Um *firewall* pode ser um *hardware* com *software* integrado, ou apenas um *software* instalado em um servidor físico ou virtual. A Figura 6 apresenta um exemplo de um *firewall* em *applience* físico.

Figura 6 - Exemplo de Firewall físico (Fortinet Fortigate 500E)

O *firewall* apresentado na Figura 6 trabalha de forma redundante, em caso de falha total de um dos equipamentos, o outro assume de forma completamente transparente sem nem mesmo causar a perda de conexões, neste *firewall* existem aproximadamente dez mil conexões ativas segundo o painel de monitoramento do próprio *firewall*, o *firewall* em questão encontra-se na sala de conectorização do *Data Center*.

2.2 MONITORAMENTO DE REDES

O monitoramento de ativos de rede está associado a diversos aspectos do gerenciamento de redes de computadores, como configuração, controle e relatórios, esses aspectos são primordiais para o administrador da rede ter uma visão geral do ambiente, bem como provisionar recursos, e mensurar o crescimento do ambiente.

"Até mesmo os equipamentos de última geração e os softwares mais atualizados não garantem sistemas imunes a erros. Por isso, qualquer sistema crítico para um negócio deve ser monitorado constantemente para evitar interrupções que prejudiquem sua utilização" (CONCEPTI, 2017).

Existem diversas formas para se monitorar um ambiente de rede e seus ativos, grande parte dos equipamentos hoje em dia possuem um sistema de monitoramento interno, vários deles inclusive, podem ser programados para enviar notificações sobre incidentes. Porém para ambientes significativamente grandes, muitas vezes se faz necessário o uso de uma ferramenta centralizada para monitorar todos os ativos de rede de um ambiente, para que se possa através de apenas uma ferramenta, monitorar: Servidores físicos, servidores virtuais (com diversos sistemas operacionais distintos), *storages*, *switches*, roteadores, *firewalls*, *no-breaks*, geradores de energia, conectividade de *links* de dados entre outros. Programando o envio de alertas sobre incidentes em uma única ferramenta, de forma centralizada e padronizada.

O que torna o monitoramento de tantos equipamentos tão distintos entre si possível é o uso do protocolo SNMP (*Simple Network Management Protocol*), pelas ferramentas de monitoramento, além do monitoramento através de agentes instalados diretamente no sistema operacional.

2.2.1 O protocolo SNMP

O protocolo SNMP, acrônimo para: *Simple Network Management Protocol* (Protocolo Simples para monitoramento de Redes) hoje em dia é um dos protocolos mais utilizados para o monitoramento de redes.

O protocolo SNMP foi descrito na RFC 1067, 1988. RFC é a sigla para *Request For Comment*. Em tradução livre para português "Pedido de comentário". Que é um documento técnico da IETF (*Internet Engineering Task Force*) em português, Força Tarefa de Engenharia da Internet. Que especifica padrões implementados e usados na web (OPSERVICES, 2017).

O Padrão SNMP foi criado para facilitar o monitoramento dos mais diversos tipos de dispositivos em uma rede, sendo possível monitorar praticamente qualquer dispositivo de rede gerenciável.

O protocolo SNMP, por padrão, utiliza a porta 161 UDP (*User Datagram Protocol*). A comunicação do protocolo SNMP se dá através de requisições (*GET*) e envio de solicitações de alterações (*SET*). Esses comandos são enviados de um servidor, que em SNMP é chamado de Gerente, para o dispositivo a ser monitorado, que em SNMP é considerado como Agente.

2.2.2 MIBs

O protocolo SNMP é capaz de monitorar os mais diversos tipos de dispositivos distintos, porém no protocolo SNMP não estão especificados quais informações de um equipamento serão monitorados, por possuir uma arquitetura expansível as informações a respeito dos itens que podem ser gerenciados em cada equipamento são obtidas graças as MIBs (*Management Information Base*) ou base de Informações de Gerenciamento.

Existem diversos tipos de MIBs, cada fabricante, ou até mesmo cada equipamento, pode possuir uma MIB especifica para seu gerenciamento.

A MIB de um equipamento é o banco de dados de objetos (itens) que podem ser gerenciados ou monitorados de um determinado equipamento, organizado de forma hierárquica. Cada um desses objetos dentro da MIB e suas respectivas categorias são identificados na árvore hierárquica da MIB por um número de OID (*Object Identifier*) ou Identificador de Objeto. A Figura 7 apresenta um exemplo de hierarquia básica da MIB, essa por sua vez pode se estender por mais diversos níveis hierárquicos, dependendo do equipamento a que se refere.

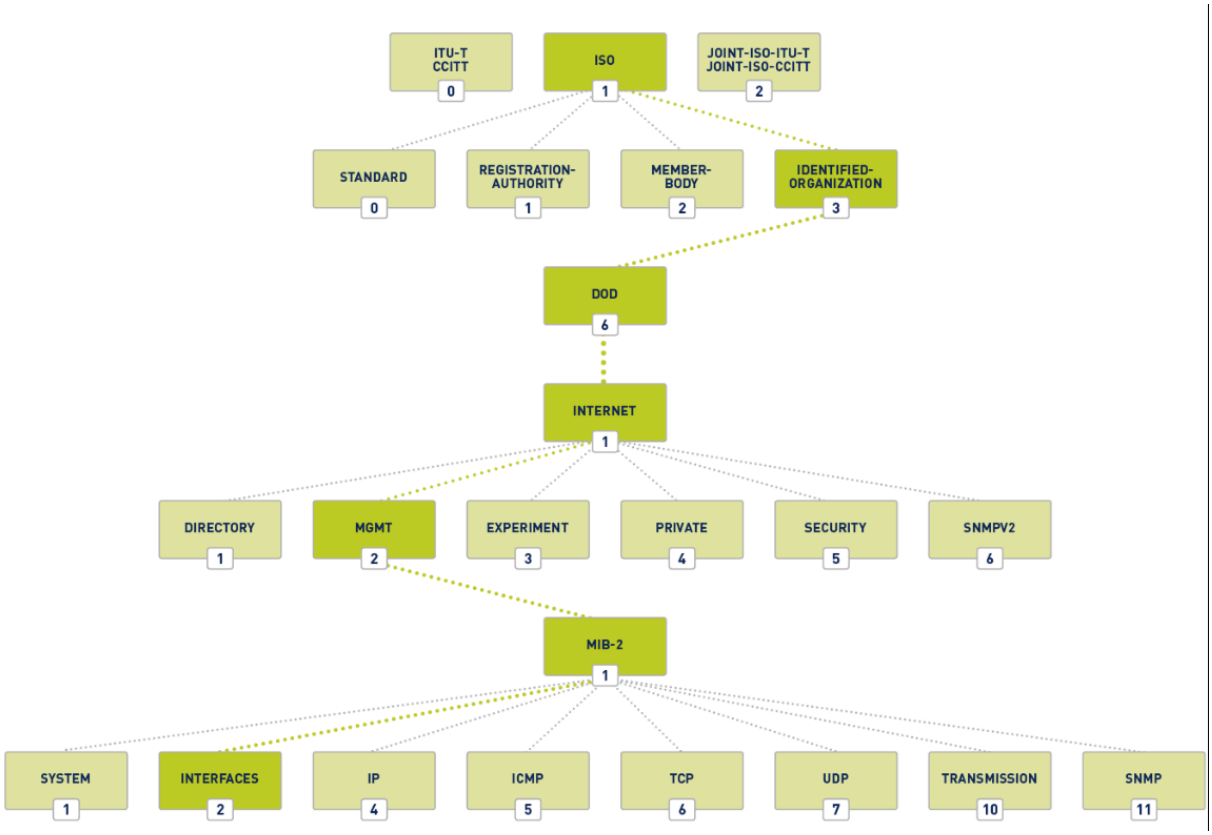


Figura 7 - Exemplo de árvore hierárquica básica da MIB
Fonte: Paessler The network monitoring company

2.2.3 Protocolos SNMPv2 e SNMPv3

Os protocolos SNMPv2 e SNMPv3, são melhoramentos do protocolo SNMP, e são descritos respectivamente na RFC 1901 e na RFC 2571.

Conforme afirma Teleco (2018), O protocolo SNMPv2 surgiu para suprir algumas deficiências do protocolo SNMP, como por exemplo o gerenciamento de redes descentralizadas, a possibilidade de transferência de grandes blocos de informação e sua comunicação baseada na *community*.

O protocolo SNMPv3 por sua vez, veio para suprir uma grande deficiência dos protocolos SNMPv1 e SNMPv2, seus requisitos de segurança. Segundo Teleco (2018), Apesar do grande esforço do IETF, ainda não foi possível no SNMPv2 se chegar a um consenso a respeito do padrão de segurança a ser usado no SNMP, e para se reparar essa falha grupos independentes começaram a trabalhar na melhoria de segurança do SNMPv2, e então em 1998, o grupo IETF SNMPv3 publicou uma série de documentos definindo a estrutura

para incorporar características de segurança junto a manter as funcionalidades totais do SNMPv1 e SNMPv2 e segurança em redes e controle de acesso.

2.3 FERRAMENTAS PARA MONITORAMENTO DE REDES

Existem diversas ferramentas que podem ser utilizadas para o monitoramento e gerenciamento de redes de computadores e seus ativos, por conta desse amplo mercado de ferramentas existem diversas opções para atender as mais diversas necessidades e os mais diversos ambientes, de acordo com as necessidades de monitoramento.

2.3.1 Nagios

Uma das mais populares ferramentas para monitoramento e gerenciamento de redes, o Nagios já foi a ferramenta mais utilizada para isso.

Em 1996 Ethan Gastlad criou uma aplicação em MS-DOS para monitoramento de ping nos servidores Novell Netware e enviar páginas numéricas. E em 1998, planejando entrar no mercado de monitoramento e gerenciamento de servidores Gastlad começou a desenvolver um novo projeto baseado em seu projeto anterior, mas desenhado para rodar sobre o Linux. Então em 1999 Ethan Liberou seu trabalho como um projeto de código aberto sob o nome de NetSaint. Em 2002, Devido aos problemas que a marca registrada com o nome "NetSaint" podia trazer a longo prazo, Gastlad decidiu renomear seu projeto para Nagios, um acrônimo recursivo para "*Nagios Ain't Gonna Insist On Sainthood*" ou seja, Nagios não vai insistir na santidade. Com o sucesso do projeto, em 2016 o Nagios superou o número de 7.500.000 (sete milhões e quinhentos mil) downloads diretamente do site SourceForge.net (NAGIOS, 2018).

Apesar de ter atingido um número muito alto de *downloads*, atualmente o Nagios tem caído em desuso devido a ter mantido por todos esses anos algumas características que não agradam alguns gestores de TI, como, por exemplo, todas suas configurações serem baseadas em arquivos texto, e não ter apresentado grandes inovações.

O Nagios monitora toda sua infraestrutura de TI, para garantir que sistemas, aplicações, serviços, e processos de negócios, estejam funcionando corretamente. O Nagios cresceu para incluir milhares de projetos, e muitos *plug-ins* desenvolvidos pela comunidade através do mundo. Nagios é oficialmente patrocinado pela Nagios Enterprises. É um produto estável, porém pode ser muito difícil de se configurar e manter, por usuários que não tenham muita experiência com o Nagios. (OPENTICA, 2016).

2.3.2 Zenoss

Zenos é o nome de uma empresa que desenvolve soluções híbridas para monitoramento analítico de TI, para ambientes em *cloud*, virtuais ou físicos de TI. Zenoss também é o nome da aplicação desenvolvida pela empresa. Zenoss era considerada uma empresa de software comercial de código aberto, porém a empresa atualmente tem se dedicado principalmente a entrega continua de sua plataforma hibrida de monitoramento de TI, e também a sua plataforma em *cloud*, que trabalha como Software as a Service (SaaS), em português, software como um serviço, conhecida como ZaaS, Zenoss as a Service, as quais não são mais gratuitas, diferentemente do Zenoss Core, que se mantem como plataforma de código aberto. (ROUSE, 2018).

Um dos grandes atrativos do Zenoss é sua forma de trabalho, por ser um sistema *agentless* (sem agente), com o Zenoss não existe a necessidade de se instalar aplicativo no computador ou servidor que será monitorado, toda sua configuração é realizada no console de gerenciamento do servidor Zenoss.

Uma das desvantagens do Zenoss com relação a outras ferramentas como Nagios, e Zabbix, é sua popularidade, por possuírem maior popularidade, as comunidades de desenvolvedores, fóruns e documentações das outras duas ferramentas são bem maiores, apesar de sua instalação e configuração ser considerada fácil, essas desvantagens com relação ao número de adeptos pode causar maiores dificuldades ao gestor do ambiente quando surjam algum problema.

2.3.3 Zabbix

O Zabbix foi criado por Alexei Vladishev em 1998, em 2001 sua primeira versão foi disponibilizada, Zabbix v1.0aplha, e em 2005, Alexei fundou a Zabbix SAI, empresa que até os dias atuais desenvolve ativamente e suporta o Zabbix. O Zabbix é uma ferramenta que monitora diversos parâmetros da rede e a saúde e integridade de servidores, e utiliza um sistema de notificações flexível, que permite ao utilizador configurar praticamente qualquer forma de envio de alertas, e para qualquer tipo de evento. Também oferece uma ótima capacidade de visualização dos dados coletados, que são armazenados em um banco de dados, provendo assim uma boa capacidade para ser utilizado no planejamento de expansão do ambiente. (ZABBIX, 2016).

Uma característica muito atrativa do Zabbix é sua capacidade de monitorar praticamente qualquer equipamento, graças a gigantesca quantidade de *templates* de monitoramento desenvolvidos por sua grande comunidade de usuários.

Outra característica que atrai muitos gestores de redes a utilizar o Zabbix é sua incrível capacidade de integração com diversas ferramentas, como por exemplo, integração com a ferramenta Grafana, para geração de gráficos ainda mais elaborados que os gráficos padrões do Zabbix, integração com serviços de envio de SMS, Serviços de e-mail SMTP, e também integração com aplicações VoIP e aplicativos de mensagens instantâneas, como por exemplo o Telegram.

Apesar de seus desenvolvimentos terem ocorrido praticamente ao mesmo tempo, no início o Nagios tinha muito mais aceitação do publico com relação ao Zabbix, cenário esse que tem se invertido com o passar dos anos, conforme mostra a Figura 8, desde 2004 o Zabbix vem crescendo continuamente na procura do publico, enquanto o Nagios, apesar de ter tido uma alta procura no início, vem decaindo rapidamente, devido aos problemas apresentados anteriormente. Atualmente o Zabbix é a ferramenta mais procurada nos termos de busca do Google, como mostra a análise do Google Trends Analysis.

Figura 8 - Comparação entre os termos de pesquisa relacionados as Ferramentas Zabbix, Zenoss e Nagios
Fonte: Google Trends Analysis

A respeito dos dados apresentados na Figura 8, os números representam o interesse de pesquisa relativo ao ponto mais alto no gráfico de uma determinada região em um dado período. Um valor de 100 representa o pico de popularidade de um termo. Um valor de 50

significa que o termo teve metade da popularidade. Uma pontuação de 0 significa que não haviam dados suficientes sobre o termo.

Apesar de sua consideravelmente complexa configuração e parametrização, o Zabbix foi considerado a ferramenta mais adequada para o desenvolvimento deste projeto, devido as características do projeto, a robustez da ferramenta e as necessidades de integração junto a outras ferramentas.

3 MATERIAIS E METODOLOGIA

A ênfase deste capítulo está em reportar as ferramentas e o método que foram empregados para a implantação do monitoramento do Zabbix, com a integração para o envio de alertas através do aplicativo Telegram no ambiente de um *Data Center*.

3.1 MATERIAIS

Foram utilizados para a implantação do projeto, além das referências bibliográficas, as ferramentas abaixo:

- Aplicação Zabbix Server 3.0: Servidor central de monitoramento;
- Aplicação Zabbix Agent 3.0: Agente de monitoramento do Zabbix que será instalado nos servidores, inclusive no servidor onde será instalado o Zabbix Server;
- Aplicação Zabbix Proxy 3.0: Aplicação Zabbix utilizada como um Proxy, para realizar o monitoramento em servidores e equipamentos que não tenham comunicação direta com o Servidor Zabbix Server;
- Banco de dados MySQL: Servidor de banco de dados utilizado pelo Zabbix Server;
- Aplicativo Telegram 5.0.17: Aplicativo que será utilizado para o envio de alertas para os gestores da rede do *Data Center*;
- Telegram BotFather: Sistema de gerenciamento e criação de Bots para automatização da ferramenta Telegram.
- Servidor Virtual para a instalação do Zabbix Server: Para a instalação do Zabbix Server estimou-se um servidor virtual, com Sistema Operacional Ubuntu Server, e os seguintes recursos de hardware, 8 vCPUs, 24 Gigabytes de memória RAM, duas unidades de disco, uma unidade de 20 Gigabytes para uso apenas para o Sistema Operacional Linux, e outra unidade de 170 Gigabytes para utilização exclusiva para o banco de dados do servidor Zabbix Server, ambas as unidades de disco utilizadas em Storages com discos de estado sólido (Solid State Disc – SSD) de alto desempenho. O servidor virtual foi criado utilizando o virtualizador VMWare.

3.1.1 Zabbix Server

Zabbix Server é o processo central do software Zabbix. O Zabbix Server controla a coleta e recebimento dos dados, realiza o processamento das triggers e o envio de alertas aos usuários. O Zabbix Server é o componente central para onde os agentes e proxies enviam seus dados, de monitoramento (ZABBIX, 2018). Todas as configurações a respeito de *template*s de monitoramento, criação de gráficos e gestão geral da ferramenta é realizada no Zabbix Server.

3.1.2 Zabbix Agent

Zabbix Agent é a aplicação do Zabbix que é instalada no servidor ou computador que será monitorado, O Zabbix Agent monitora dados de memória, carga de processamento, trafego em interfaces de rede e uso de disco. O Zabbix Agent obtém localmente os dados operacionais do sistema e reporta esses dados para o Zabbix Server, ou para o Zabbix Proxy (ZABBIX, 2018).

3.1.3 Zabbix Proxy

Zabbix Proxy é um processo que pode coletar dados de monitoramento de um ou mais dispositivos monitorados, e envia as informações para o Zabbix Server. Essencialmente funcionando em nome do Zabbix Server, Na visão do agente monitorado, o Zabbix Proxy passa a ser o Zabbix Server, assim o Zabbix Agent envia os dados para o Zabbix Proxy, onde esses são armazenados temporariamente, e enviados para o Zabbix Server. (ZABBIX, 2018).

3.1.4 Telegram

Telegram é um aplicativo de troca de mensagens instantâneas amplamente utilizado pelo mundo, apesar de seu concorrente direto (WhatsApp) ser mais difundido em grande parte do mundo, o Telegram possui recursos de integração que não existem nativamente para o WhatsApp, além de ser um aplicativo de código aberto, enquanto o WhatsApp possui código proprietário. Um dos recursos diferenciados do Telegram, é o Telegram BotFather, que é um *bot* do aplicativo, um *bot* (derivado da palavra *robot,* robô em tradução livre) do telegrama, é uma conta especial que não requer um número de telefone para sua criação, como um robô esses *bots* possuem a capacidade de interagir com o usuário, enviando mensagens e recebendo comandos. O BotFather é utilizado para criar autonomamente *bots* de automação dentro do

aplicativo. Esse recurso será utilizado para realizar a integração entre o Zabbix e o Telegram e será explicado mais detalhadamente no Capítulo 4.

3.2 METODOLOGIA

Nessa seção está descrita a metodologia utilizada para implantação do monitoramento do ambiente de um *Data Center* via Zabbix com integração para envio de alertas via Telegram.

3.2.1 Levantamento Bibliográfico

O principal objetivo do levantamento bibliográfico, foi ampliar os conhecimentos a respeito de ativos de rede, gerenciamento de redes e aplicações para esse fim, com o intuito de atender os objetivos propostos da melhor forma possível.

3.2.2 Levantamento do Ambiente

Nessa etapa foi realizado o levantamento completo de todo o ambiente do *Data Center* para elencar quais ativos de rede precisam ser monitorados, e estimar tempos necessários para a implantação da ferramenta.

Foram identificados 750 Servidores virtuais, 12 Servidores Físicos, 5 *Storages*, 20 *switches* e 8 *No-breaks*.

3.2.3 Levantamento de Itens e Ajustes dos *Templates* de Monitoramento

Após realizar o levantamento de toda a infraestrutura do *Data Center*, foi necessário realizar o levantamento de todos os itens que precisam ser monitorados prioritariamente. Em conversa com a diretoria responsável pelo setor, identificamos que o monitoramento seria mais eficaz se destinado apenas aos itens mais críticos para o ambiente, sendo assim os itens que serão monitorados são:

- Uso de processador;
- Uso de memória;
- Uso de disco;
- Trafego de rede;

- *Uptime* do computador;

Os itens acima levantados, se aplicam a todos os servidores físicos, servidores virtuais e *switches* monitorados, tanto via agente, quanto para os monitorados via protocolo SNMP, com a adição apenas do monitoramento da disponibilidade do agente do Zabbix nos servidores monitorados via agente. Na sequência foram realizadas alterações nos *templates* de monitoramento padrão, para abranger os itens necessários para o monitoramento satisfatório do ambiente em questão.

3.2.4 Instalação e configuração do Zabbix, Server, Proxies e Agentes

Nessa etapa do trabalho foi realizada a instalação do servidor do Zabbix e sua configuração, bem como ajustes nos parâmetros e configurações do Zabbix Server. Instalação de 55 (cinquenta e cinco) servidores do Zabbix Proxy. E por fim a instalação do Zabbix Agent em todos os 750 Servidores virtuais. Esses procedimentos estão melhor descritos no Capítulo 4 deste trabalho.

3.2.5 Ajustes finos das configurações

Após configurados todos os serviços necessários, Servidor Zabbix, Proxies e Agentes, foi verificado o desempenho do servidor do Zabbix, através de seus próprios gráficos de desempenho, e realizados ajustes para melhorar o desempenho, bem como garantir a disponibilidade do serviço, esses ajustes estão descritos no Capítulo 4.

3.2.6 Integração com Telegram

Na última etapa das configurações foram realizadas as configurações e testes necessários para se utilizar o aplicativo Telegram como meio para o envio de alertas da ferramenta.

3.2.7 Documentação dos procedimentos

A última, mas muito importante etapa do desenvolvimento desse trabalho, foi a documentação de todos os procedimentos realizados, a fim de servir como base de estudo para futuras implantações da ferramenta em outros ambientes, bem como para que seja possível

que outros administradores da rede do *Data Center* em questão consigam manter esse monitoramento.

4 RESULTADOS

Nesse capítulo são apresentados os procedimentos de instalação e configuração da ferramenta Zabbix e os procedimentos de implantação de sua integração com o Telegram.

4.1 CENÁRIO ATUAL E PROBLEMAS ENFRENTADOS

A infraestrutura atual do *Data Center* possui 750 servidores virtuais, com sistemas operacionais diversos, dependendo a finalidade do servidor, todos esses servidores estão virtualizados em 12 servidores físicos, com *hardwares* das marcas Dell, IBM, HP e Nutanix; 5 *Storages* para armazenamento, das marcas HP, IBM e SuperMicro; 8 *No-Breaks* de grande porte com módulos auxiliares de bateria das marcas Emerson e APC; e 20 *Switches* de grande porte gerenciáveis da marca HP.

Esse número elevado de ativos de rede gera diversos incidentes diariamente, como por exemplo, um servidor virtual de banco de dados fica sem espaço de armazenamento disponível não sendo possível assim a gravação ou alteração de dados no banco de dados. Atualmente, sem possuir um monitoramento desses recursos, só são identificados problemas, após eles já terem causado um grande impacto, como no exemplo citado a parada do servidor de banco de dados, causando a parada total do sistema de clientes, gerando assim uma indisponibilidade de serviços do *Data Center*.

Esse tipo de incidente gera muitos problemas para a empresa, sendo assim foi feita a proposta para a empresa, da implantação de um sistema de monitoramento do ambiente, esse com isso foi realizada a comparação entre as ferramentas Nagios, Zabbix e Zenoss, demonstradas na Seção 2.3 do presente trabalho.

Uma grande peculiaridade que existe no ambiente em questão é que 500 desses servidores virtuais são de clientes distintos, divididos em 55 sub-redes isoladas completamente entre si, para que seja possível o monitoramento de todos os servidores dentro dessas redes isoladas, sem quebrar o isolamento dessas redes, foi optado pelo uso do Zabbix Proxy em todos os servidores de firewall independentes dessas redes, pois é o único ponto que já possui comunicação com todas as máquinas da sub-rede de cada cliente e ao mesmo tempo com o Zabbix Server.

4.2 INSTALAÇÃO DO ZABBIX SERVER

O processo de instalação do Zabbix Server via pacotes não é em si muito complicado, porém deve ser tomado o cuidado de executar todos os procedimentos na ordem correta, para evitar erros e falta de configurações necessárias.

Para a instalação do Zabbix Server no ambiente do *Data Center* em questão foi utilizado um servidor virtual com sistema operacional Ubuntu Server versão 16.04 LTS por ser um servidor virtualizado, seus recursos de hardware são escaláveis, não sendo relevante no momento.

4.2.1 Instalação e configuração via pacotes

O passo inicial para realizar a instalação do Zabbix Server foi baixar os arquivos necessários para instalação do Zabbix Server, nesse pacote também estão disponíveis as outras ferramentas necessárias para a instalação.

Conectado como root ao servidor onde foi instalado o Zabbix Server, o comando abaixo foi utilizado para fazer o *download* do pacote de instalação do zabbix da versão 3.0 para o sistema operacional Ubuntu 16.04 (Xenial):

```
# wget https://repo.zabbix.com/zabbix/3.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.0-2+xenial_all.deb
```

Também existem diversos outros pacotes para instalação em outros sistemas operacionais, ou versões diferentes do Zabbix, e podem ser encontrados em: https://www.zabbix.com/download.

Após baixar o pacote de instalação foi necessário instalá-lo como repositório, utilizando o comado abaixo:

```
# dpkg -i zabbix-release_3.0-2+xenial_all.deb
```

Extraídos os arquivos do pacote, foi preciso atualizar o índice dos repositórios do *apt*, utilizando o comando:

```
# apt update
```

Concluída a atualização do índice dos repositórios, foi possível instalar o Zabbix Server com seu banco de dados mysql, a interface do Zabbix em php e o Zabbix agente para que seja possível ao Zabbix Server fazer o monitoramento do próprio servidor.

```
# apt install zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

Concluída a instalação das aplicações necessárias, foi realizado as configurações básicas do banco de dados mysql.

Primeiro passo, foi necessário fazer login pela primeira vez no banco de dados e definir a senha, utilizando o comando:

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

Foi solicitada senha para o banco de dados do Zabbix, essa senha será utilizada na configuração do Zabbix, Senha foi utilizado para ocultar a senha utilizada nas configurações do Zabbix no ambiente do *Data Center* devido aos critérios de segurança da empresa.

```
Enter password:Senha
```

Configurada a senha do servidor de banco de dados, foi necessário criar a base de dados do Zabbix, através do comando *create database*

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

Criada a base de dados, foi necessário dar ao Zabbix os privilégios para seu correto funcionamento com o banco de dados mysql. Para isso, foi utilizado o comando abaixo, novamente Senha foi utilizado para ocultar a senha utilizada na configuração.

```
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'Senha';
```

Após configurar os privilégios necessários ao Zabbix no banco de dados, foi preciso executar o *script* SQL para a criação da estrutura do banco de dados do Zabbix, executando o comando abaixo:

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

Após criar a estrutura do banco de dados, foi necessário configurar a senha do banco de dados no arquivo de configurações do Zabbix Server, para isso foi preciso editar o arquivo /etc/zabbix/zabbix_server.conf e, dentro dele, no campo DBPassword informar a senha configurada anteriormente para o banco de dados, conforme exibido na Figura 9.

```
### Option: DBPassword
#       Database password. Ignored for SQLite.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=Senha
```

Figura 9 - Configuração de senha no arquivo de configuração do Zabbix Server

O próximo passo foi configurar o fuso horário dentro das configurações do apache, editando o arquivo /etc/zabbix/apache.conf, descomentando o campo **php_value date.timezone** e definindo o mesmo para **America/São_Paulo**, conforme exibido na Figura 10.

```
<IfModule mod_php5.c>
    php_value max_execution_time 300
    php_value memory_limit 128M
    php_value post_max_size 16M
    php_value upload_max_filesize 2M
    php_value max_input_time 300
    php_value max_input_vars 10000
    php_value always_populate_raw_post_data -1
    php_value date.timezone America/Sao_Paulo
</IfModule>
<IfModule mod_php7.c>
    php_value max_execution_time 300
    php_value memory_limit 128M
    php_value post_max_size 16M
    php_value upload_max_filesize 2M
    php_value max_input_time 300
    php_value max_input_vars 10000
    php_value always_populate_raw_post_data -1
    php_value date.timezone America/Sao_Paulo
</IfModule>
```

Figura 10 - Configuração de parâmetros do apache para o Zabbix

Agora a configuração inicial do Zabbix Server está feita, após isso foi necessário reiniciar os serviços

```
# systemctl restart zabbix-server zabbix-agent apache2
```

Também foi necessário configurar para que esses serviços sejam iniciados automaticamente junto com o Linux com o comando systemctl enable

```
# systemctl enable zabbix-server zabbix-agent apache2
```

4.2.2 Configuração do acesso web do Zabbix

Para realizar a configuração inicial do acesso web do servidor Zabbix pela primeira vez, foi preciso acessar o Zabbix no navegador, usando o endereço IP do servidor Zabbix http://IP_zabbix_server/zabbix (IP_zabbix_server foi utilizado para ocultar o endereço IP do servidor Zabbix do *Data Center*, por motivos de segurança). Enquanto o servidor não tiver sido configurado, é exibida a tela de configuração do acesso web conforme exibido na Figura 11.

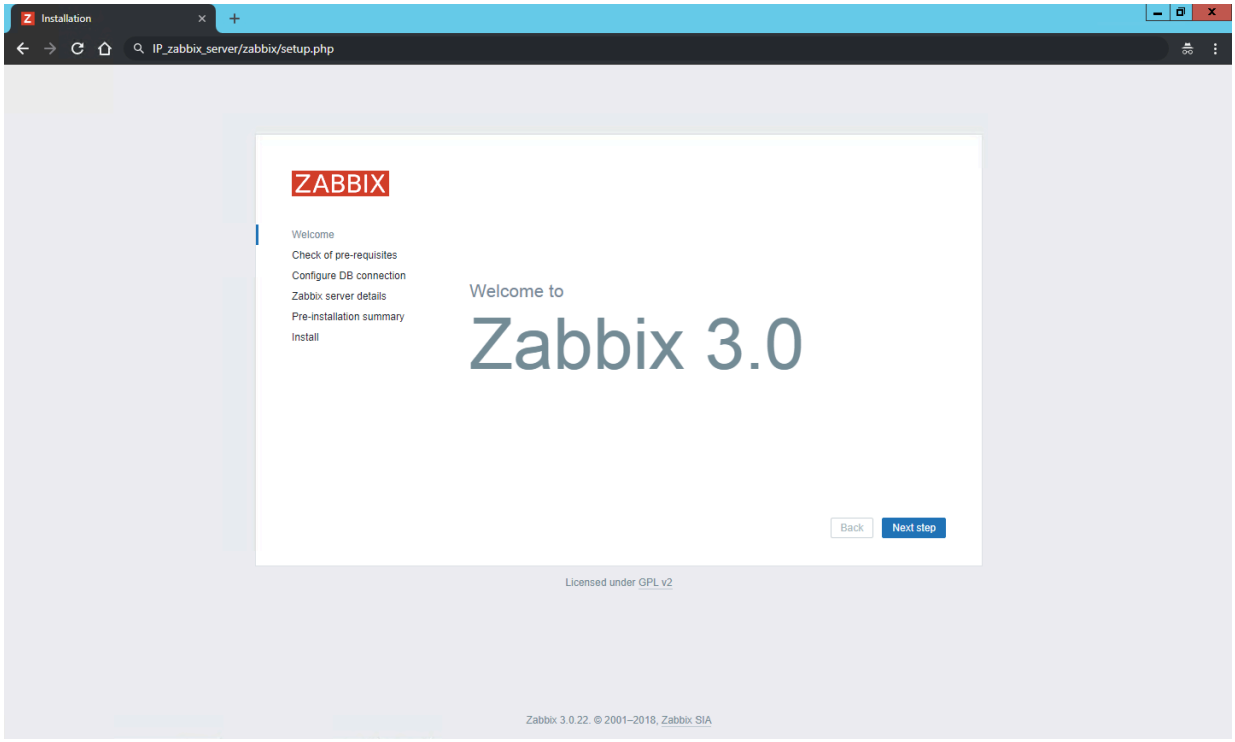


Figura 11 - Tela inicial de configuração do acesso web do Zabbix

A primeira tela da configuração da interface web do Zabbix é apenas uma tela de boas vindas, após clicar em *Next step* (Próxima etapa), foram exibidos os requisitos necessários para a instalação, caso exista algum requisito que não fosse atendido, seria exibido nessa tela, com todos os requisitos de instalação atendidos, foi habilitado o botão *Next step*, conforme demonstrado na Figura 12.

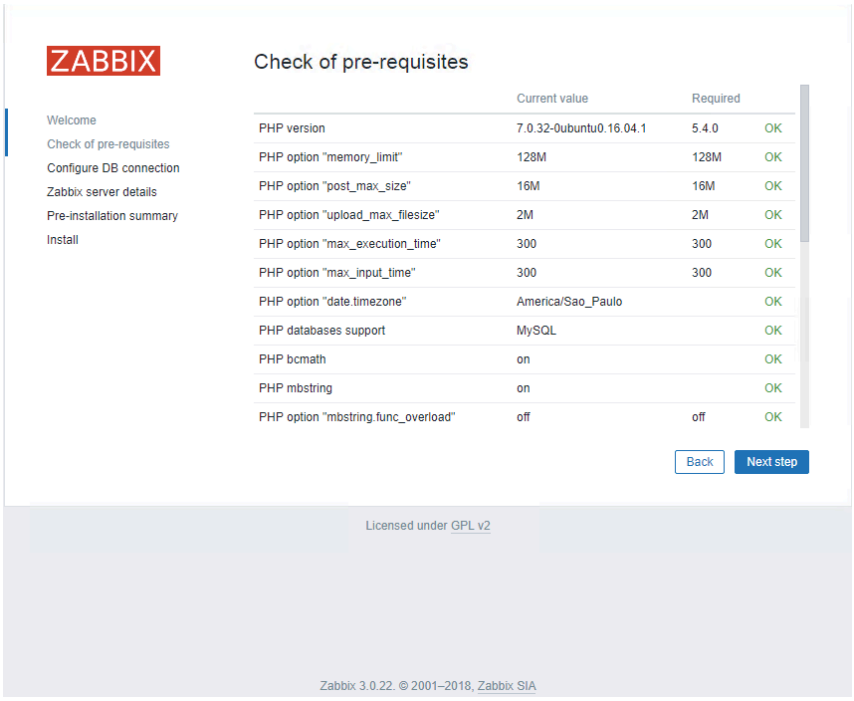


Figura 12 - Tela de requisitos de instalação do Zabbix Server

Na próxima tela de configuração, foram requisitados os dados de conexão com o banco de dados do Zabbix, nessa tela é necessário preencher corretamente os dados de conexão com o banco de dados, utilizando a senha que foi criada anteriormente. Conforme exibido na Figura 13. Os dados de conexão do servidor Zabbix foram ocultados na Figura 13 devido as normas de segurança da empresa.

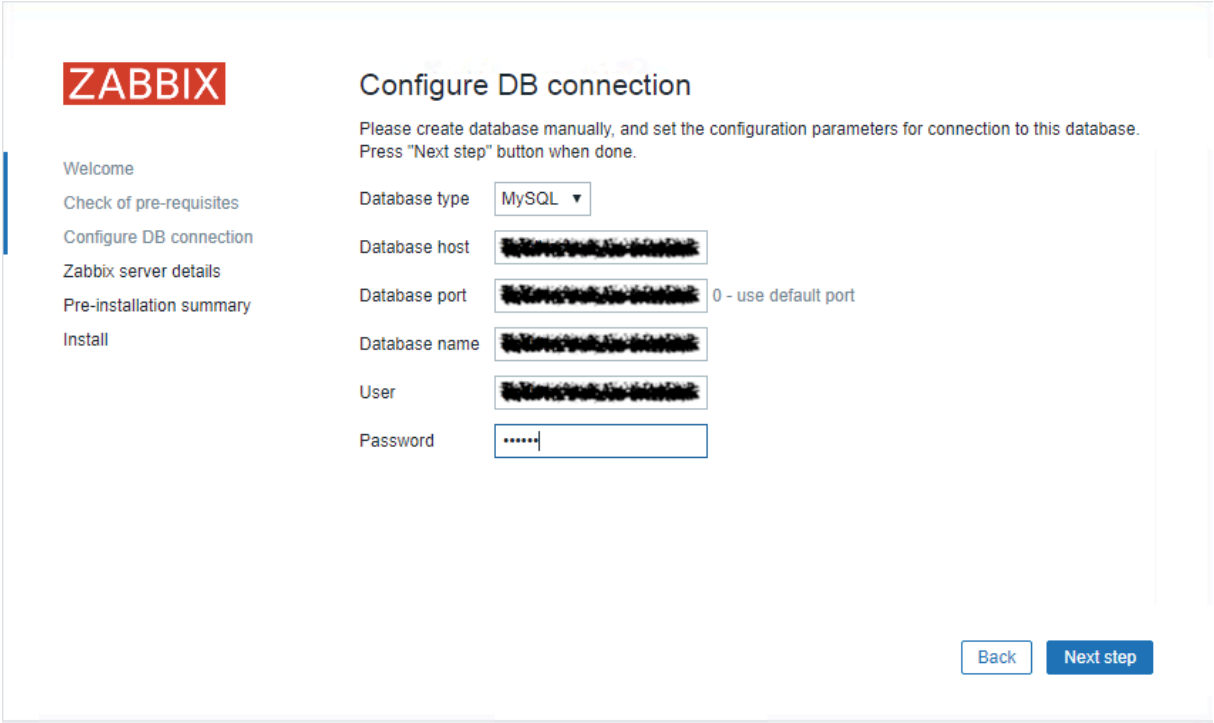


Figura 13 - Configuração da conexão com o banco de dados do Zabbix

Após corretamente preenchidos os dados de conexão do banco de dados, ao clicar em *Next step* foi exibida a tela para configuração do endereço e porta do Zabbix Server, conforme exibido na Figura 14. Novamente os dados de conexão do servidor foram ocultados devido as normas de segurança da empresa.

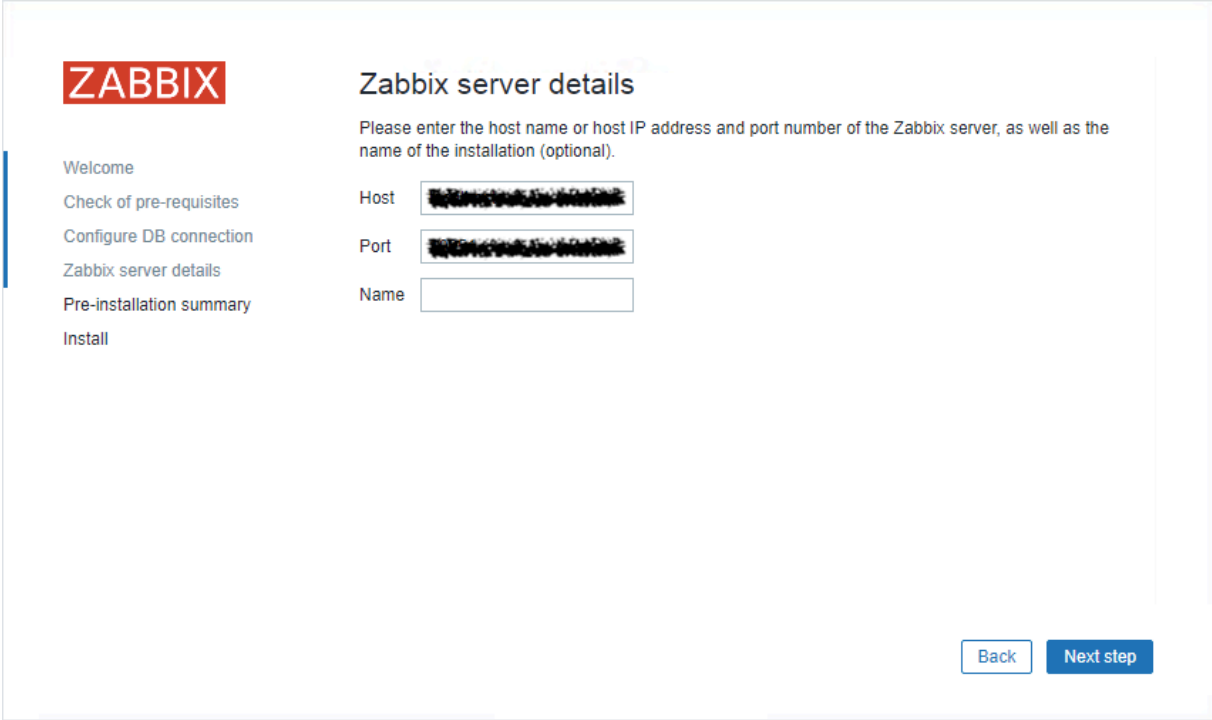


Figura 14 - Tela de configuração da conexão do Zabbix Server.

Após informar os dados corretos de configuração do Zabbix Server e clicar em *Next Step*, foi exibida uma tela de confirmação com os dados informados do banco de dados, e das configurações do Zabbix. Conforme exibido na Figura 15, novamente esses dados foram ocultados devido normas de segurança da informação do *Data Center*.

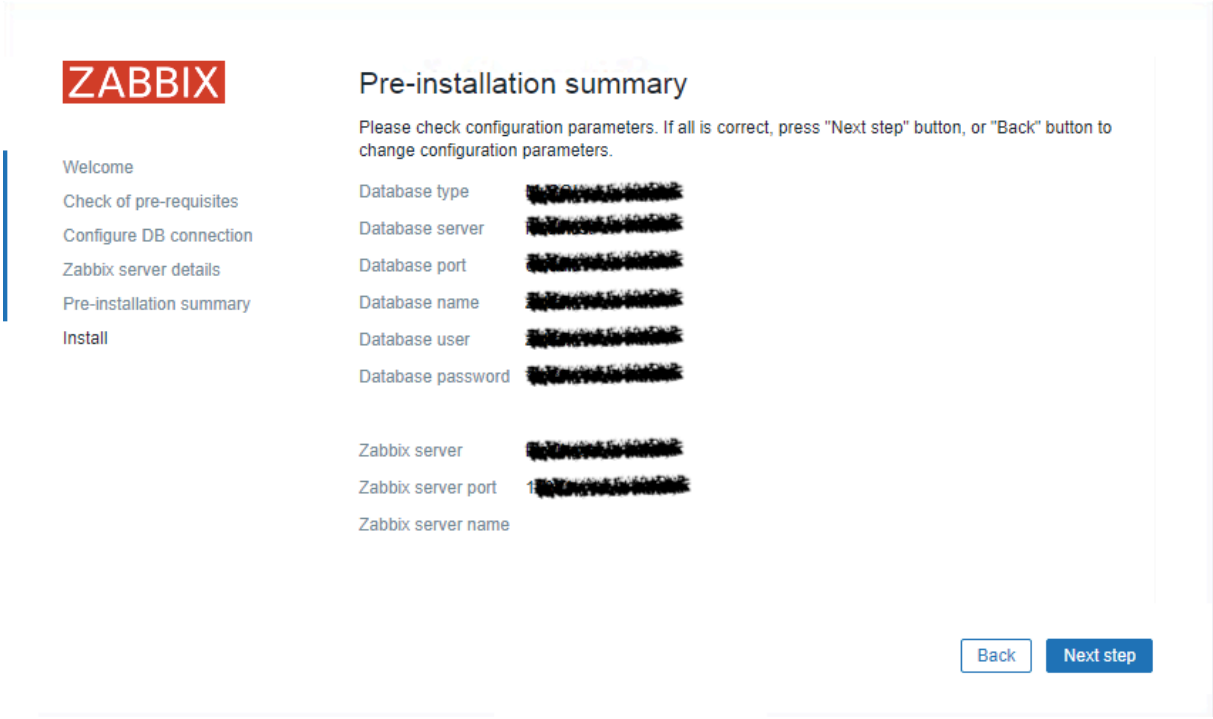


Figura 15 - Tela de confirmação dos dados de conexão do banco de dados e Zabbix Server

Após confirmados os dados informados, ao clicar em *Next step* foi exibida a tela de conclusão da configuração. Conforme exibido na Figura 16. Após clicar no botão *Finish*, o usuário é redirecionado diretamente para a página de login na interface web do Zabbix.

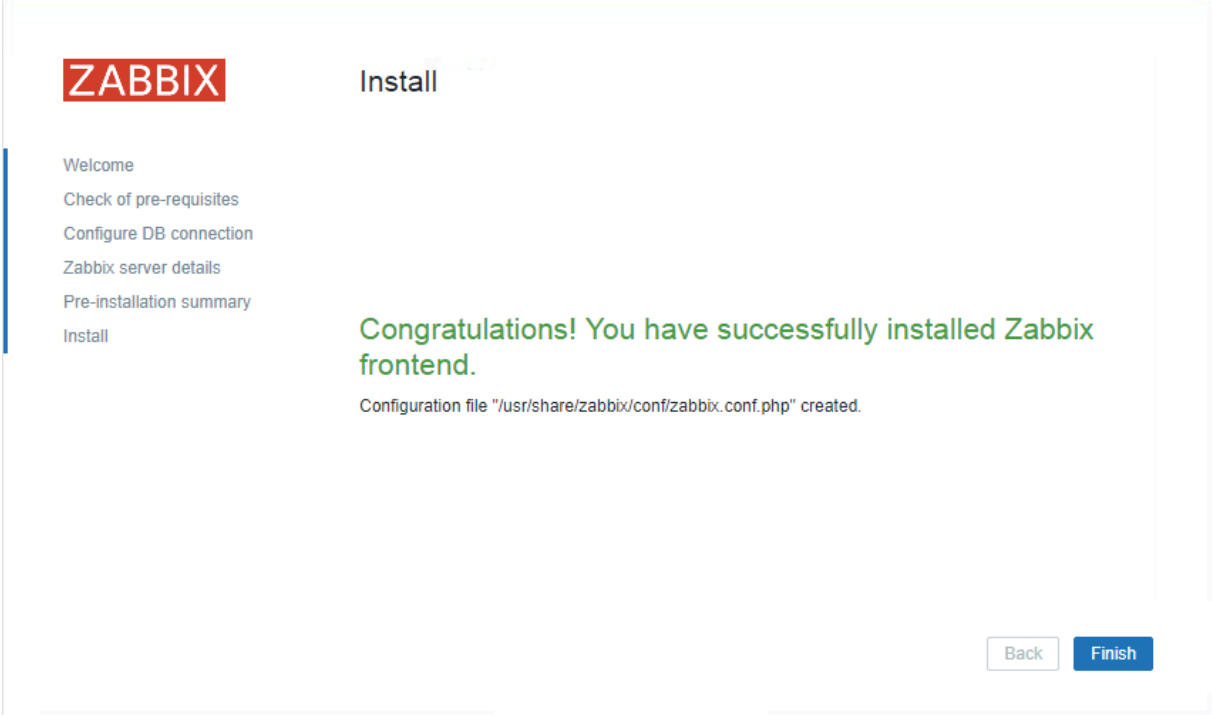


Figura 16 - Tela de conclusão da instalação da interface Web do Zabbix

Uma etapa fundamental após configurar a interface de gerenciamento web do Zabbix é a configuração dos usuários de acesso, o usuário padrão da interface web do Zabbix é "Admin" com a senha "zabbix". A primeira ação executada foi realizar a alteração dessa senha para uma senha complexa para melhor garantir a segurança dos dados presentes no Zabbix.

4.3 INSTALAÇÃO E CONFIGURAÇÃO DO ZABBIX PROXY

Uma característica peculiar do *Data Center* em questão, é que para cada cliente, existe um ambiente completamente isolados uns dos outros, utilizando servidores Linux como *firewalls* individuais virtuais, através dos estudos levantados neste trabalho, foi identificado como uma alternativa viável para ser possível monitorar esses servidores, sem quebrar o isolamento de rede.

Instalando nos servidores virtuais de *firewall* o Zabbix Proxy é possível monitorar todos os servidores que estão em cada sub-rede individual atrás desses servidores de *firewall*. Todos os firewalls em questão utilizam o sistema operacional Linux Ubuntu 14.04 LTS.

Assim como seu funcionamento, a instalação e configuração do Zabbix Proxy é bem simples, neste trabalho foi utilizada a versão 3.0 do Zabbix Proxy, para o sistema operacional Linux Ubuntu 14.04 LTS (Trusty).

A primeira etapa da instalação do Zabbix Proxy, foi fazer o *download* do arquivo do repositório de instalação do Zabbix 3.0 para o sistema operacional Linux Ubuntu 14.04 LTS (Trusty).

```
# wget http://repo.zabbix.com/zabbix/3.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_3.0-2+trusty_all.deb
```

A etapa seguinte foi instalar o repositório, utilizando o comando:

```
# dpkg -i zabbix-release_3.0-2+trusty_all.deb
```

Após instalar o repositório é muito importante atualizar o índice dos repositórios do apt-get, utilizando o comando:

```
# apt-get update
```

Com o repositório necessário instalado, e os índices dos repositórios atualizados, foi realizada a instalação do Zabbix Proxy, utilizando o banco SQLite3, para isso foi utilizado o comando:

```
# apt-get install zabbix-proxy-sqlite3
```

Após instalado o Zabbix Proxy, foi preciso criar o diretório para o banco de dados do proxy, e definir o usuário “zabbix” como dono do diretório, utilizando os comandos:

```
# mkdir /var/lib/zabbix
# chown zabbix. -R /var/lib/zabbix/
```

Após criado o diretório do banco de dados do Zabbix Proxy, foi necessário editar o arquivo de configuração do Zabbix Proxy (/etc/zabbix/zabbix_proxy.conf), para os servidores do Zabbix Proxy do *Data Center* em questão, foram utilizadas as configurações demonstradas na Figura 17, apenas sendo alterado o campo *Hostname*.

```
ProxyMode=0
Server=[IP DO ZABBIX SERVER]
Hostname=[Hostname do Servidor Firewall]
LogFile=/var/log/zabbix/zabbix_proxy.log
LogFileSize=0
DebugLevel=3
PidFile=/var/run/zabbix/zabbix_proxy.pid
DBName=/var/lib/zabbix/zabbix.db
DBUser=zabbix
ProxyOfflineBuffer=24
ConfigFrequency=60
DataSenderFrequency=10
StartTrappers=20
```

Figura 17 - Arquivo de configuração do Zabbix Proxy

Após salvar o arquivo de configuração já configurado, foi necessário iniciar o serviço do Zabbix Proxy pela primeira vez, utilizando o comando:

```
# /etc/init.d/zabbix-proxy start
```

Após iniciar o Zabbix Proxy, foi necessário apenas adicionar um novo proxy ao Zabbix. Acessando o Zabbix no navegador, clicando em **Administração>Proxies**, depois clicando em **Criar proxy**. Adicionado o Hostname informado no arquivo de configuração do Zabbix Proxy e selecionando o tipo de proxy como **Ativo**, selecionados os servidores virtuais monitorados por ele, e clicado em **Adicionar**, em alguns segundos foi detectado o novo proxy e as máquinas apontadas a ele coletarão seus dados. Essa configuração é demonstrada na Figura 18. Os dados de nomes de Servidores configurados foram ocultados por questão de segurança.

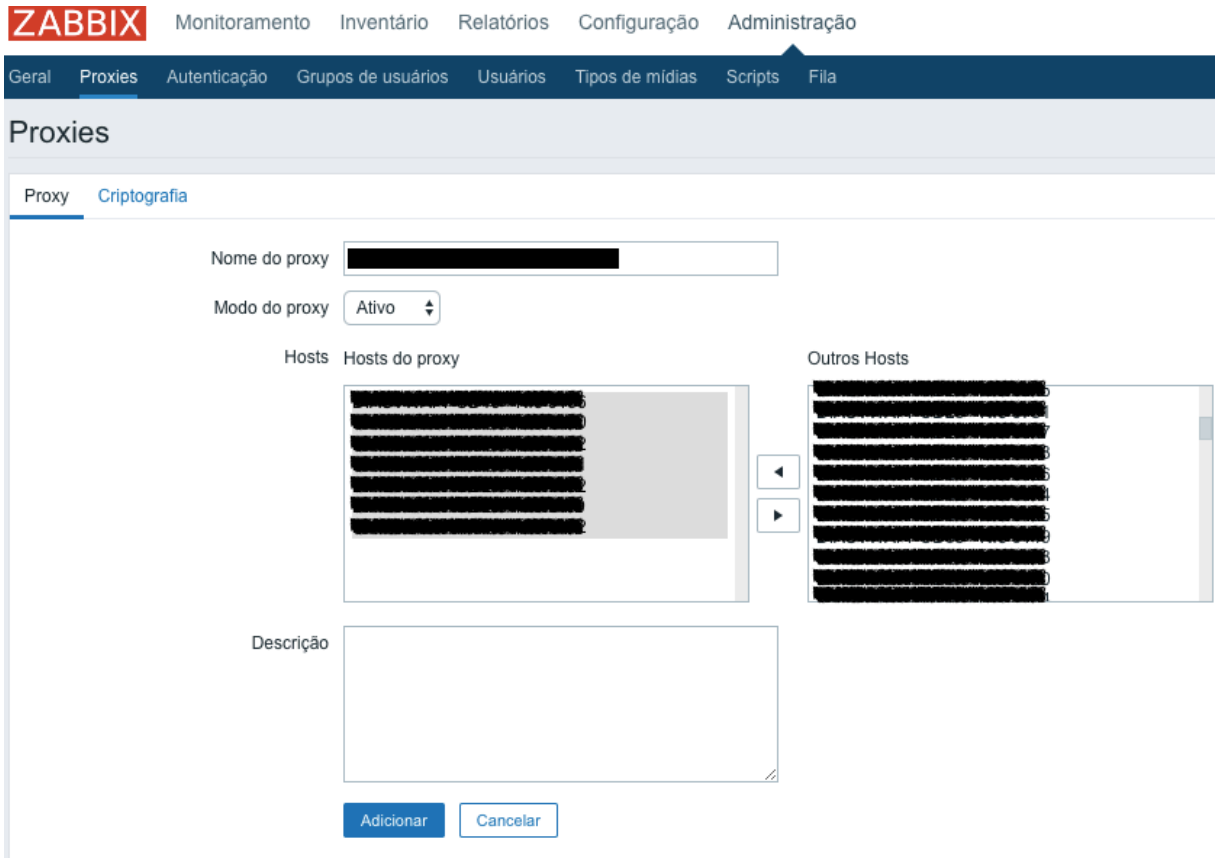


Figura 18 - Configuração do servidor do Zabbix Proxy no Zabbix Server

4.4 CONFIGURAÇÃO DOS TEMPLATES DE MONITORAMENTO

A etapa mais complexa e importante do presente trabalho é a configuração e testes dos *templates* de monitoramento para cada tipo de ativo da rede a ser monitorado no *Data Center*.

4.4.1 Configuração dos templates para servidores virtuais Windows e Linux

O Zabbix vem por padrão com *templates* para o monitoramento de servidores com sistemas operacionais Windows e Linux, porém para atender o monitoramento da infraestrutura, da forma definida pela direção da empresa, foram necessários ajustes nos *templates* de monitoramento, dos sistemas operacionais Linux e Windows.

4.4.1.1 Configuração do template de monitoramento para sistemas operacionais Windows

O *template* padrão de monitoramento para servidores Windows chamado "*Template OS Windows*" possui 19 itens que estão listados na Figura 19.

Nome ▲	Triggers	Chave
Template App Zabbix Agent: Agent ping	Triggers 1	agent.ping
Average disk read queue length		perf_counter[\234(_Total)\1402]
Average disk write queue length		perf_counter[\234(_Total)\1404]
File read bytes per second		perf_counter[\2\16]
File write bytes per second		perf_counter[\2\18]
Free memory	Triggers 1	vm.memory.size[free]
Free swap space		system.swap.size[,free]
Free virtual memory, in %	Triggers 1	vm.vmemory.size[pavailable]
Template App Zabbix Agent: Host name of zabbix_agentd running	Triggers 1	agent.hostname
Number of processes	Triggers 1	proc.num[]
Number of threads		perf_counter[\2\250]
Processor load (1 min average)	Triggers 1	system.cpu.load[percpu,avg1]
Processor load (5 min average)		system.cpu.load[percpu,avg5]
Processor load (15 min average)		system.cpu.load[percpu,avg15]
System information	Triggers 1	system.uname
System uptime	Triggers 1	system.uptime
Total memory		vm.memory.size[total]
Total swap space		system.swap.size[,total]
Template App Zabbix Agent: Version of zabbix_agent(d) running	Triggers 1	agent.version

Figura 19 - Itens do template padrão de sistemas operacionais Windows

Baseado nos itens descritos na Figura 19, o *template* padrão de monitoramento para sistemas operacionais Windows, possui 9 *triggers*, que estão demonstradas na Figura 20.

Severidade	Nome ▲	Expressão
Média	Host information was changed on {HOST.NAME}	{Template OS Windows:system.uname.**diff**(0)}>0
Informação	Template App Zabbix Agent: Host name of zabbix_agentd was changed on {HOST.NAME}	{Template OS Windows:agent.hostname.**diff**(0)}>0
Média	Lack of available virtual memory on server {HOST.NAME}	({TRIGGER.VALUE}=0 and {Template OS Windows:vm.vmemory.size[pavailable].**min**(10m)}<10) or ({TRIGGER.VALUE}=1 and {Template OS Windows:vm.vmemory.size[pavailable].**min**(10m)}<20)
Média	Lack of free memory on server {HOST.NAME}	{Template OS Windows:vm.memory.size[free].**last**(0)}<10000
Média	Processor load is too high on {HOST.NAME}	{Template OS Windows:system.cpu.load[percpu,avg1].**avg**(5m)}>5
Média	Too many processes on {HOST.NAME}	{Template OS Windows:proc.num[].**avg**(5m)}>300
Informação	Template App Zabbix Agent: Version of zabbix_agent(d) was changed on {HOST.NAME}	{Template OS Windows:agent.version.**diff**(0)}>0
Média	Template App Zabbix Agent: Zabbix agent on {HOST.NAME} is unreachable for 5 minutes	{Template OS Windows:agent.ping.**nodata**(5m)}=1
Média	{HOST.NAME} has just been restarted	{Template OS Windows:system.uptime.**change**(0)}<0

Figura 20 - Triggers do template padrão de sistemas operacionais Windows

O *template* de monitoramento padrão também possui duas regras de descoberta automática de itens, destinadas para a descoberta de unidades de disco, chamada "*Mounted filesystem discovery*" e para a descoberta de interfaces de rede, chamada "*Network interface Discovery*". Os protótipos de itens padrões das regras de descoberta de unidades de disco e de interfaces de rede estão demonstrados nas figuras 21 e 22, respectivamente.

Nome ▲	Chave
Free disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},free]
Free disk space on {#FSNAME} (percentage)	vfs.fs.size[{#FSNAME},pfree]
Total disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},total]
Used disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},used]

Figura 21 - Protótipos de itens da regra de descoberta de unidades de disco do template OS Windows

Nome ▲	Chave
Incoming network traffic on {#IFNAME}	net.if.in[{#IFNAME}]
Outgoing network traffic on {#IFNAME}	net.if.out[{#IFNAME}]

Figura 22 - Protótipos de itens da regra de descoberta de interfaces de rede do template OS Windows

A regra de descoberta de unidades de disco, por padrão possui um protótipo de *trigger*, demonstrado na Figura 23.

Severidade	Nome ▲	Expressão
Atenção	Free disk space is less than 20% on volume {#FSNAME}	{Template OS Windows:vfs.fs.size[{#FSNAME},pfree].last(0)}<20

Figura 23 - Protótipo de trigger da regra de descoberta de unidades de disco do template OS Windows

O *template* padrão para o sistema operacional Windows, precisou de diversas modificações, como a remoção de itens e *triggers* desnecessários e a adição de novos itens, *triggers* e protótipos de *triggers*. O objetivo dessas modificações foi atender os seguintes requisitos de itens a serem monitorados:

- Carga do processador (*Load*);
- Tempo de atividade do sistema (*Uptime*);
- Uso de memória (em bytes e em valor percentual);
- Tamanho total da memória do servidor;
- Uso de processador (valor percentual);
- Número de sessões do *Microsoft Terminal Services* (Total / Ativas / Inativas);
- Conectividade com o Zabbix Server (disponibilidade do Zabbix Agent).

Também atender os seguintes itens de regra de descobertas a serem monitorados:

- Espaço livre por disco (em bytes e em valor percentual);
- Espaço utilizado em disco (em bytes);
- Espaço total alocado ao disco (em bytes);
- Tráfego de entrada/saída por interface de rede.

Após realizadas as alterações no *template*, os itens do *template* ficaram configurados conforme apresentado na Figura 24.

Nome	Triggers	Chave
Processor load (5 min average)		system.cpu.load[percpu,avg5]
Processor load (1 min average)		system.cpu.load[percpu,avg1]
System uptime	Triggers 1	system.uptime
Memory Usage %	Triggers 1	vm.memory.size[pused]
Memory Usage		vm.memory.size[used]
Total memory		vm.memory.size[total]
Processor load (15 min average)		system.cpu.load[percpu,avg15]
Uso de Processador %	Triggers 1	perf_counter[\238(_Total)\6]
Version of zabbix_agent(d) running		agent.version
Agent ping	Triggers 1	agent.ping
Total de Sessões ativas de TS		perf_counter["\Serviços de Terminal\Sessões Ativas"]
Total de Sessões Inativas de TS		perf_counter["\Serviços de Terminal\Sessões Inativas"]
Total de Sessões de TS		perf_counter["\Serviços de Terminal\Total de Sessões"]
Host name of zabbix_agentd running		agent.hostname

Figura 24 - Novos itens do template para sistemas operacionais Windows

Baseados nos novos itens também foram criadas novas **triggers** para a geração de incidentes relacionados a esses itens. Essas *triggers* estão demonstradas na Figura 25.

Severidade	Nome	Expressão
Desastre	Uso de Memoria muito alto no Servidor: {HOST.NAME}	{Template OS Windows New:vm.memory.size[pused].**avg**(5m)}>=99
Média	O Servidor {HOST.NAME} Reiniciou	{Template OS Windows New:system.uptime.**change**(0)}<0
Desastre	Processamento muito alto no servidor: {HOST.NAME}	{Template OS Windows New:perf_counter[\238(_Total)\6].**avg**(5m)}>=99
Desastre	Zabbix indisponível por 5 Minutos no Servidor: {HOST.NAME}	{Template OS Windows New:agent.ping.**nodata**(5m)}=1

Figura 25 - Novas *triggers* do template para sistemas operacionais Windows

Os itens das regras de descobertas de interfaces de rede e unidades de disco não precisaram ser alterados, pois já estavam conforme o solicitado pela diretoria do *Data Center*. Porém os protótipos de *triggers* para a regra de descoberta de unidades de disco, precisaram ser revisados sendo necessária a alteração da *trigger* existente, e também a criação de outro protótipo de *trigger*, os protótipos de *triggers* após a alteração, são demonstrados na Figura 26.

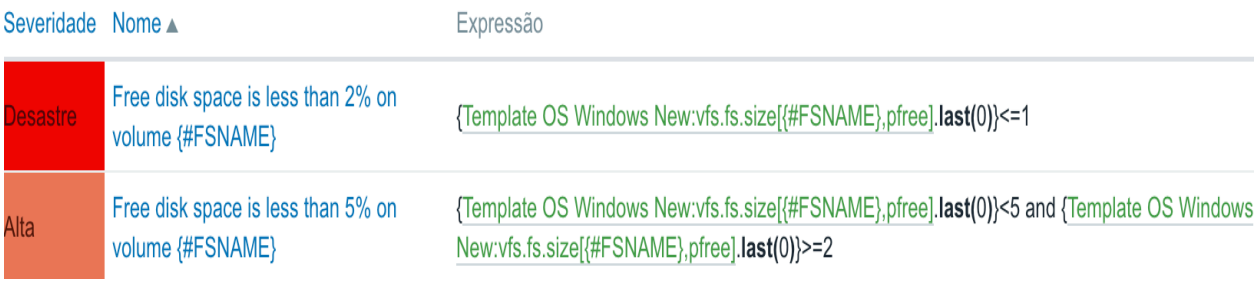


Figura 26 - Novos protótipos de triggers do template OS windows.

Após concluir as alterações no *template*, foram realizados testes em um servidor virtual Windows utilizado para testes internos, testadas todos os itens e *triggers*, e todos funcionaram corretamente.

4.4.1.2 Configuração do template de monitoramento para sistemas operacionais Linux

O *template* padrão de monitoramento para servidores Linux, chamado de "*Template OS Linux*", possui 32 itens padrão, que são apresentados na Figura 27.

Nome ▲	Triggers	Chave
Template App Zabbix Agent: Agent ping	Triggers 1	agent.ping
Available memory	Triggers 1	vm.memory.size[available]
Checksum of /etc/passwd	Triggers 1	vfs.file.cksum[/etc/passwd]
Context switches per second		system.cpu.switches
CPU steal time		system.cpu.util[,steal]
CPU softirq time		system.cpu.util[,softirq]
CPU interrupt time		system.cpu.util[,interrupt]
CPU idle time		system.cpu.util[,idle]
CPU user time		system.cpu.util[,user]
CPU nice time		system.cpu.util[,nice]
CPU system time		system.cpu.util[,system]
CPU iowait time	Triggers 1	system.cpu.util[,iowait]
Free swap space		system.swap.size[,free]
Free swap space in %	Triggers 1	system.swap.size[,pfree]
Host boot time		system.boottime
Host local time		system.localtime
Host name	Triggers 1	system.hostname
Template App Zabbix Agent: Host name of zabbix_agentd running	Triggers 1	agent.hostname
Interrupts per second		system.cpu.intr
Maximum number of opened files	Triggers 1	kernel.maxfiles
Maximum number of processes	Triggers 1	kernel.maxproc
Number of logged in users		system.users.num
Number of processes	Triggers 1	proc.num[]
Number of running processes	Triggers 1	proc.num[,,run]
Processor load (1 min average per core)	Triggers 1	system.cpu.load[percpu,avg1]
Processor load (5 min average per core)		system.cpu.load[percpu,avg5]
Processor load (15 min average per core)		system.cpu.load[percpu,avg15]
System information	Triggers 1	system.uname
System uptime	Triggers 1	system.uptime
Total memory		vm.memory.size[total]
Total swap space		system.swap.size[,total]
Template App Zabbix Agent: Version of zabbix_agent(d) running	Triggers 1	agent.version

Figura 27 - Itens do template padrão de sistemas operacionais Linux.

Baseado nos itens apresentados na Figura 27, o *template* para sistemas operacionais Linux possui 15 triggers padrões, que são apresentadas na Figura 28.

Severidade	Nome ▲	Expressão
Atenção	/etc/passwd has been changed on {HOST.NAME}	{Template OS Linux:vfs.file.cksum[/etc/passwd].**diff**(0)}>0
Informação	Configured max number of opened files is too low on {HOST.NAME}	{Template OS Linux:kernel.maxfiles.**last**(0)}<1024
Informação	Configured max number of processes is too low on {HOST.NAME}	{Template OS Linux:kernel.maxproc.**last**(0)}<256
Atenção	Disk I/O is overloaded on {HOST.NAME}	{Template OS Linux:system.cpu.util[,iowait].**avg**(5m)}>20
Informação	Host information was changed on {HOST.NAME}	{Template OS Linux:system.uname.**diff**(0)}>0
Informação	Template App Zabbix Agent: Host name of zabbix_agentd was changed on {HOST.NAME}	{Template OS Linux:agent.hostname.**diff**(0)}>0
Informação	Hostname was changed on {HOST.NAME}	{Template OS Linux:system.hostname.**diff**(0)}>0
Média	Lack of available memory on server {HOST.NAME}	{Template OS Linux:vm.memory.size[available].**last**(0)}<20M
Atenção	Lack of free swap space on {HOST.NAME}	{Template OS Linux:system.swap.size[,pfree].**last**(0)}<50
Atenção	Processor load is too high on {HOST.NAME}	{Template OS Linux:system.cpu.load[percpu,avg1].**avg**(5m)}>5
Atenção	Too many processes on {HOST.NAME}	{Template OS Linux:proc.num[].**avg**(5m)}>300
Atenção	Too many processes running on {HOST.NAME}	{Template OS Linux:proc.num[,,run].**avg**(5m)}>30
Informação	Template App Zabbix Agent: Version of zabbix_agent(d) was changed on {HOST.NAME}	{Template OS Linux:agent.version.**diff**(0)}>0
Média	Template App Zabbix Agent: Zabbix agent on {HOST.NAME} is unreachable for 5 minutes	{Template OS Linux:agent.ping.**nodata**(5m)}=1
Informação	{HOST.NAME} has just been restarted	{Template OS Linux:system.uptime.**change**(0)}<0

Figura 28 - Triggers do template padrão para sistemas operacionais Linux

O *template* de monitoramento padrão Linux também possuí duas regras de descoberta automática de itens, destinadas para a descoberta de unidades de disco, chamada "*Mounted filesystem discovery*" e para a descoberta de interfaces de rede, chamada "*Network interface Discovery*". Os protótipos de itens padrões das regras de descoberta de unidades de disco e de interfaces de rede estão demonstrados nas Figuras 29 e 30 respectivamente.

Nome ▲	Chave
Free disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},free]
Free disk space on {#FSNAME} (percentage)	vfs.fs.size[{#FSNAME},pfree]
Free inodes on {#FSNAME} (percentage)	vfs.fs.inode[{#FSNAME},pfree]
Total disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},total]
Used disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},used]

Figura 29 - Protótipos de itens da regra de descoberta de unidades de disco do Template OS Linux

Nome ▲	Chave
Incoming network traffic on {#IFNAME}	net.if.in[{#IFNAME}]
Outgoing network traffic on {#IFNAME}	net.if.out[{#IFNAME}]

Figura 30 - Protótipos de itens da regra de descoberta de interfaces de rede do Template OS Linux

Na regra de descoberta de unidades de disco do *template* de sistemas operacionais Linux também existem 2 protótipos de *triggers* padrões, apresentados na Figura 31.

Severidade	Nome ▲	Expressão
Atenção	Free disk space is less than 20% on volume {#FSNAME}	{Template OS Linux:vfs.fs.size[{#FSNAME},pfree].last(0)}<20
Atenção	Free inodes is less than 20% on volume {#FSNAME}	{Template OS Linux:vfs.fs.inode[{#FSNAME},pfree].last(0)}<20

Figura 31 - Protótipos de triggers da regra de descoberta de unidades de disco do template OS Linux

O *template* padrão para os sistemas operacionais linux, precisou de diversas modificações, como a remoção de itens e *triggers* desnecessários e a adição de novos itens, *triggers* e protótipos de *triggers*. O objetivo dessas modificações foi atender os seguintes requisitos de itens a serem monitorados:

- Carga do processador (*Load*);
- Tempo de atividade do sistema (*Uptime*);
- Uso de memória (em bytes e em valor percentual);
- Tamanho total da memória do servidor;
- Uso de processador (valor percentual);

- Informações gerais do sistema (número de usuários conectados, mudanças nas informações gerais do sistema);
- Conectividade com o Zabbix Server (disponibilidade do Zabbix Agent).

Também atender os seguintes itens de regra de descobertas a serem monitorados:

- Espaço livre por disco (em bytes e em valor percentual);
- Espaço utilizado em disco (em bytes);
- Espaço total alocado ao disco (em bytes);
- Tráfego de entrada/saída por interface de rede.

Após realizadas as alterações necessárias no *template* de monitoramento para sistemas operacionais Linux, o *template* ficou com apenas 28 itens, demonstrados na Figura 32.

Nome	Triggers	Chave
Free swap space in %	Triggers 1	system.swap.size[,pfree]
Total swap space		system.swap.size[,total]
Free swap space		system.swap.size[,free]
Host local time		system.localtime
CPU user time		system.cpu.util[,user]
Host name	Triggers 1	system.hostname
System information	Triggers 1	system.uname
System uptime	Triggers 1	system.uptime
Total memory		vm.memory.size[total]
Memory Usage		vm.memory.size[used]
Memory Usage %	Triggers 1	vm.memory.size[pused]
Checksum of /etc/passwd	Triggers 1	vfs.file.cksum[/etc/passwd]
Number of logged in users		system.users.num
CPU system time		system.cpu.util[,system]
CPU steal time		system.cpu.util[,steal]
Processor load (15 min average per core)		system.cpu.load[percpu,avg15]
Processor load (1 min average per core)	Triggers 1	system.cpu.load[percpu,avg1]
Host boot time		system.boottime
Version of zabbix_agent(d) running	Triggers 1	agent.version
Agent ping	Triggers 1	agent.ping
Processor load (5 min average per core)		system.cpu.load[percpu,avg5]
Processor load		system.cpu.load[percpu]
CPU nice time		system.cpu.util[,nice]
CPU softirq time		system.cpu.util[,softirq]
CPU iowait time	Triggers 1	system.cpu.util[,iowait]
CPU interrupt time		system.cpu.util[,interrupt]
CPU idle time	Triggers 1	system.cpu.util[,idle]
Host name of zabbix_agentd running	Triggers 1	agent.hostname

Figura 32 - Novos itens do template para sistemas operacionais Linux

Baseados nos novos itens do *template* de monitoramento para servidores com sistema operacional Linux, foram realizadas alterações em suas *triggers*, para realizar alertas de forma mais assertiva para os responsáveis pelo servidor. As novas configurações de *triggers* para o *template OS Linux* são apresentadas na Figura 33.

Severidade	Nome	Expressão
Informação	{HOST.NAME} has just been restarted	{Template OS Linux New:system.uptime.**change**(0)}<0
Informação	/etc/passwd has been changed on {HOST.NAME}	{Template OS Linux New:vfs.file.cksum[/etc/passwd].**diff**(0)}>0
Desastre	Uso de Memoria muito alto no Servidor: {HOST.NAME}	{Template OS Linux New:vm.memory.size[pused].**avg**(5m)}>=99
Desastre	Processamento muito alto no servidor: {HOST.NAME}	{Template OS Linux New:system.cpu.util[,idle].**avg**(5m)}<=1
Informação	Host information was changed on {HOST.NAME}	{Template OS Linux New:system.uname.**diff**(0)}>0
Informação	Hostname was changed on {HOST.NAME}	{Template OS Linux New:system.hostname.**diff**(0)}>0
Atenção	Processor load is too high on {HOST.NAME}	{Template OS Linux New:system.cpu.load[percpu,avg1].**avg**(5m)}>5
Desastre	Zabbix agent on {HOST.NAME} is unreachable for 5 minutes	{Template OS Linux New:agent.ping.**nodata**(5m)}=1
Informação	Host name of zabbix_agentd was changed on {HOST.NAME}	{Template OS Linux New:agent.hostname.**diff**(0)}>0
Informação	Version of zabbix_agent(d) was changed on {HOST.NAME}	{Template OS Linux New:agent.version.**diff**(0)}>0

Figura 33 - Novas triggers do Template OS Linux

Os itens da regra de descoberta de interfaces de rede (*Network interface Discovery*) não precisaram ser alterados pois já atendiam os requisitos necessários, porém, para a regra de descoberta de unidades de disco (*Mounted filesystem Discovery*) foi removido um item desnecessário, os itens da regra de descoberta de unidades de disco para sistemas operacionais Linux estão descritos na Figura 34.

Nome ▲	Chave
Free disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},free]
Free disk space on {#FSNAME} (percentage)	vfs.fs.size[{#FSNAME},pfree]
Total disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},total]
Used disk space on {#FSNAME}	vfs.fs.size[{#FSNAME},used]

Figura 34 - Novos itens da regra de descoberta de unidades de disco para sistemas operacionais Linux

Ainda para a regra de descoberta de unidades de disco, foram realizadas alterações nos protótipos de *triggers* baseadas nos itens descritos na Figura 34, os novos protótipos de triggers são apresentados na Figura 35.

Severidade	Nome ▲	Expressão
Desastre	Free disk space is less than 1% on volume {#FSNAME}	{Template OS Linux New:vfs.fs.size[{#FSNAME},pfree].**last**(0)}<1
Alta	Free disk space is less than 5% on volume {#FSNAME}	{Template OS Linux New:vfs.fs.size[{#FSNAME},pfree].**last**(0)}<5 and {Template OS Linux New:vfs.fs.size[{#FSNAME},pfree].**last**(0)}>=1

Figura 35 - Novos protótipos de triggers da regra de descoberta de unidades de disco do template OS Linux

Após serem realizadas as alterações no *template* para sistemas operacionais Linux, o mesmo foi configurado em um servidor de testes com sistema operacional Suse Linux Enterprise Server 11 SP4, em que foram executados diversos testes, para conferir se os dados indicados pelos itens no Zabbix estavam corretos com relação aos dados de desempenho do servidor e também realizados testes das triggers, para verificar se todos os alertas configurados estavam gerando os incidentes de forma correta.

Após concluídos todos os testes tanto para o *template* para sistemas operacionais Windows, quanto para sistemas Linux, os *templates* passaram a ser os novos *templates* de produção, mantendo os *templates* antigos apenas como um backup

Os principais testes realizados foram:

- Validar se os dados monitorados pelos itens eram os mesmos apresentados internamente no servidor (por exemplo o percentual de memória utilizado em determinado momento de tempo).
- Realizar testes de triggers (como por exemplo, simular um pico de uso de processador no servidor, e validar se a trigger de alerta de uso de processador muito elevado emitia o alerta corretamente).

Após as validações serem concluídas o processo seguinte foi a instalação do Zabbix Agent em todos os servidores virtuais. Os procedimentos de instalação do Zabbix Agent em servidores Windows e Linux são apresentados nos apêndices A e B respectivamente.

4.4.2 Monitoramento de servidores físicos usando SNMP

No ambiente do *Data Center* são utilizados 11 servidores físicos, das marcas DELL, HP e IBM todos utilizados como virtualizadores VMWare ESXi. Todos esses servidores são monitorados através de um único *template* de monitoramento baseado no protocolo SNMP.

Por conta da virtualização VMWare funcionar com base em um sistema operacional Linux, para o monitoramento desses servidores físicos, foi utilizado o *template* padrão do Zabbix para monitoramento de servidores Linux via SNMP chamado "*Template* SNMP OS Linux".

Para realizar o monitoramento do Servidor físico primeiro foi necessário ativar e configurar o protocolo SNMP do servidor, mas antes foi necessário ativar o acesso SSH (*Secure Shell*) ao servidor. Acessando a interface do VMWare, e clicando em ***Configuration > Security Profile*** e em ***Services*** clicando em "***Properties..***", selecionado o serviço **SSH** e clicando em "***Options...***", selecionando "***Start and stop with host***" e clicando em ***Start*** para iniciar o serviço, em seguida confirmadas as alterações. Na Figura 36 é exibida a tela de inicialização do serviço de SSH do VMWare.

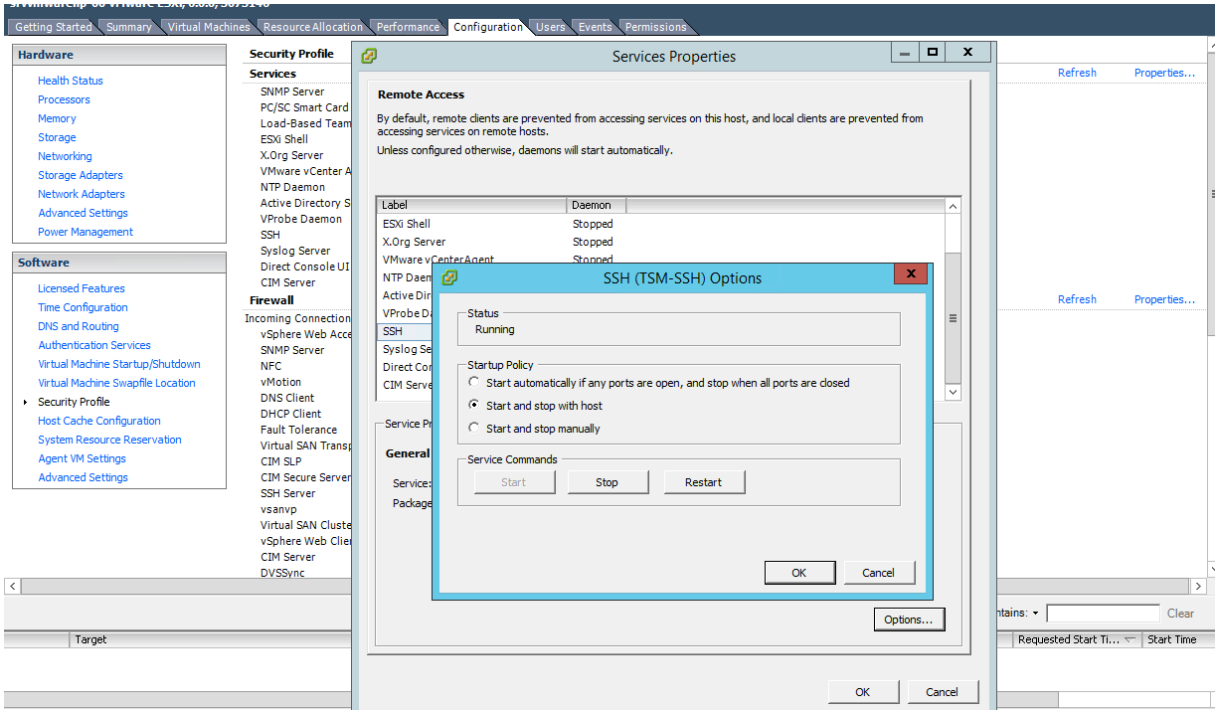


Figura 36 - Tela de inicialização do serviço de SSH do VMWare ESXi.

Após configurado o acesso SSH, foi realizado o acesso SSH ao servidor e configurado as informações do servidor SNMP. Para configurar o nome da *community*, foi usado o comando:

```
# esxcli system snmp set --communities NOMECOMMUNITY
```

Alterando o valor "**NOMECOMMUNITY**" pelo nome da *community* SNMP que foi utilizado no ambiente.

Na sequência configurar o contato responsável pelo equipamento, usando o comando abaixo, substituindo os dados entre parênteses pelos dados corretos do contato responsável pelo equipamento.

```
# esxcli system snmp set --syscontact= "NOMECONTATO <EMAILCONTATO>"
```

Após isso foi informado o local físico onde esse equipamento está instalado, utilizando o comando abaixo, substituindo apenas os dados entre parênteses pelo local correto onde o equipamento está fisicamente instalado.

```
# esxcli system snmp set --syslocation="LOCALDOEQUIPAMENTO"
```

A última parte da configuração de SNMP no servidor VMWare, foi ativar a comunicação SNMP, para isso é utilizado o comando:

```
# esxcli system snmp set --enable true
```

Após ser ativado o protocolo SNMP, já foi possível scanear os itens monitorados via SNMP, utilizando o SNMPWalk:

```
# snmpwalk -v2c -c NOMECOMUNNITY IPDOSERVIDOR
```

Após ser testado o escaneamento dos itens SNPM do servidor, foi realizada a configuração do servidor físico como um *Host* no Zabbix. Na tela de configuração de um novo ***Host***, foram configurados os seguintes parâmetros:

- **Nome do *Host*:** Configurado o *hostname* do servidor físico.
- **Nome Visivel:** Configurado um nome amigável para facilitar a identificação do servidor.
- **Grupos:** Configurado um grupo em que foram adicionados todos os servidores físicos.
- **Interfaces SNMP:** Configurado o IP do servidor e porta do SNMP

Na Figura 37 é exibido um exemplo das configurações realizadas ocultando as informações sensíveis do *Data Center* por motivos de segurança.

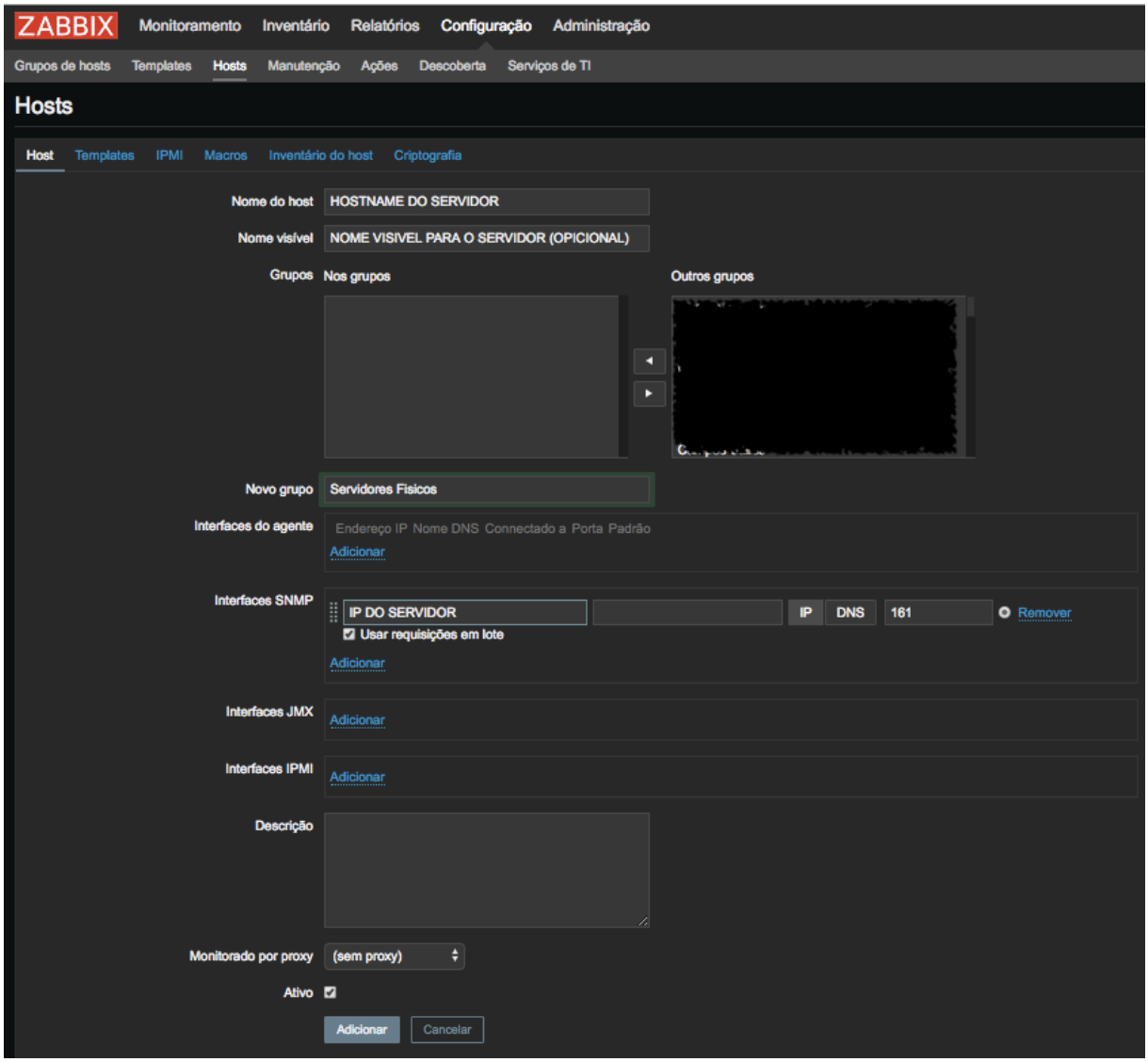


Figura 37 - Configuração do Host físico no Zabbix

Na aba ***Templates*** foram associados os *templates* "*Template SNMP OS Linux*" e o *template* para monitoramento ICMP Ping conforme mostrado na Figura 38.

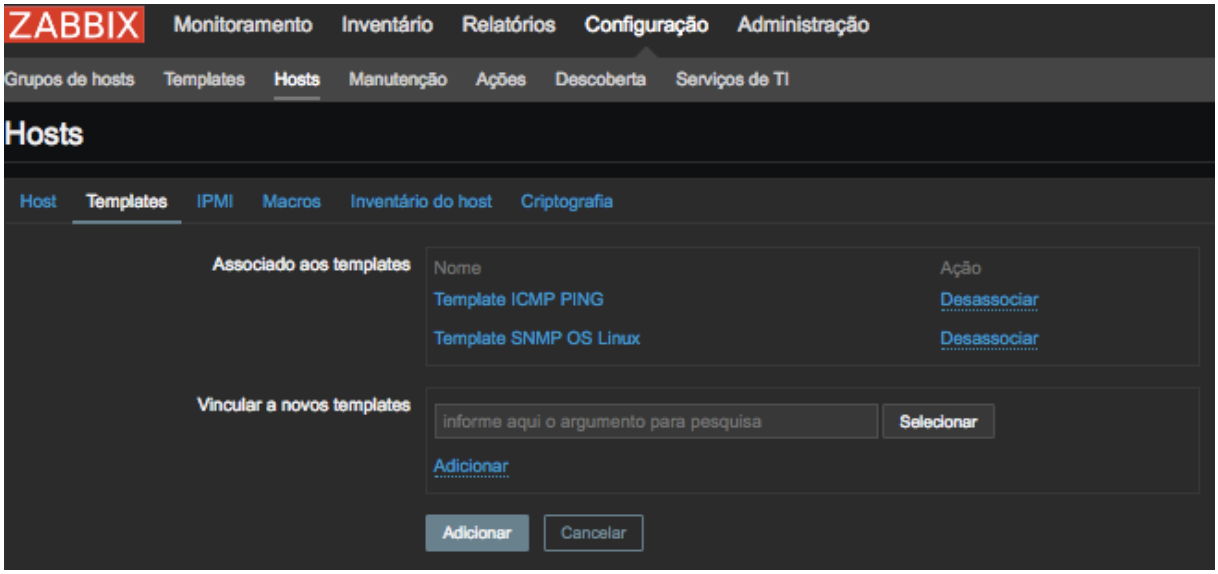


Figura 38 - Templates associados ao servidor físico.

Por fim, na aba **Macros** foi configurado o macro SNMP necessário para o *template*, o macro utilizado pelo *Template SNMP OS Linux* é o **{$SNMP_COMMUNITY}**, e o valor que precisa ser informado nele é o nome da *community* configurado no servidor, após isso foi confirmada a adição do servidor ao Zabbix. Na Figura 39 é exibido o exemplo da configuração realizada para o Macro.

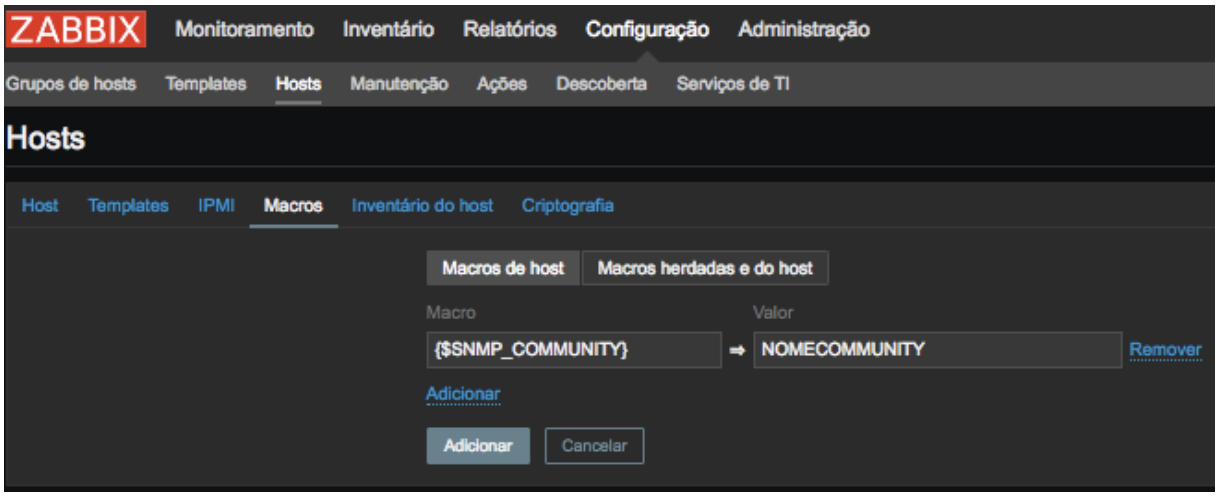


Figura 39 - Configuração de Macros para o monitoramento de Servidores físicos

Com relação aos itens a ser monitorados, foi realizada apenas uma alteração no *template* de monitoramento ICMP Ping, alterando a severidade da trigger de alerta de indisponibilidade do servidor via Ping, da severidade Média para Desastre. As Figuras 40 e 41 demonstram respectivamente como eram as configurações padrão e como ficaram as triggers do ***Template ICMP Ping***.

Severidade	Nome	Expressão
Atenção	Response time is too high on {HOST.NAME} **Depende de:** Template ICMP PING: {HOST.NAME} is unavailable by ICMP	{Template ICMP PING:icmppingsec.**avg**(5m)}>0.15
Atenção	Ping loss is too high on {HOST.NAME} **Depende de:** Template ICMP PING: {HOST.NAME} is unavailable by ICMP	{Template ICMP PING:icmppingloss.**min**(5m)}>20
Média	{HOST.NAME} is unavailable by ICMP	{Template ICMP PING:icmpping.**max**(#3)}=0

Figura 40 - Configuração padrão de Triggers do Template ICMP Ping

Severidade	Nome	Expressão
Atenção	Response time is too high on {HOST.NAME} **Depende de:** Template ICMP PING: {HOST.NAME} is unavailable by ICMP	{Template ICMP PING:icmppingsec.**avg**(5m)}>0.15
Atenção	Ping loss is too high on {HOST.NAME} **Depende de:** Template ICMP PING: {HOST.NAME} is unavailable by ICMP	{Template ICMP PING:icmppingloss.**min**(5m)}>20
Desastre	{HOST.NAME} is unavailable by ICMP	{Template ICMP PING:icmpping.**max**(#3)}=0

Figura 41 - Nova configuração de Triggers do Template ICMP Ping

Essa alteração da severidade da *trigger* de indisponibilidade teve como objetivo gerar um alerta de criticidade máxima caso qualquer ativo de rede passe a estar indisponível via Ping.

4.4.3 Monitoramento de Switches gerenciáveis através de SNMPv3

Um ativo muito importante para ser monitorado no *Data Center* em questão são seus *switches* gerenciáveis. Um fator que facilita bastante esse processo é que os *switches* em questão são padronizados sendo *switches* HP A5120, o primeiro passo foi encontrar um *template* para monitoramento desse modelo de equipamentos, devido não existir nenhum *template* padrão no Zabbix para esse equipamento. No Fórum Zabbix *Share* é possível encontrar diversos *templates* para muitos tipos de equipamentos, nesse fórum foi possível encontrar um *template* de monitoramento para switches HP A5120 do autor Jakub Samek e está disponível abertamente para download no fórum Zabbix Share: https://share.zabbix.com/network_devices/cat-hp/hp-a5120. Na publicação do *template* o autor aponta que o mesmo utiliza os seguintes macros para seu correto funcionamento:

- Macro: {$SNMP_V3_USER} Valor: Usuário de acesso SNMP
- Macro: {$SNMP_V3_AUTHPASSPHRASE} Valor: Senha de autenticação SNMP
- Macro: {$SNMP_V3_PRIVPASSPHRASE} Valor: Senha *privacy* SNMP

O *template* em questão chama-se "***Template* HP A5120**" e possui como padrão os itens descritos na Figura 42.

Nome ▲	Triggers	Chave
CPU usage	Triggers 2	switch.cpu
Device contact details		sysContact
Device description		sysDescr
Device location		sysLocation
Device name		sysName
External Power Supply 1		Ext.Power.Supply
Fan 1		fan1.status
Internal Power Supply 1		Int.Power.Supply1
Memory usage		switch.memory
Power Supply Sensor		Power.Supply.Sensor
Switch Temperature	Triggers 1	switch.temp
SysUptime	Triggers 1	SysUptime

Figura 42 - Itens padrão do Template HP A5120

O *Template* HP A5120 padrão também possui 4 triggers por padrão, que estão demonstradas na Figura 43.

Severidade	Nome ▲	Expressão
Alta	CPU usage on {HOST.HOST} > 80%	{Template HP A5120:switch.cpu.**last()**}>80
Alta	Memory usage on {HOST.HOST} > 80%	{Template HP A5120:switch.cpu.**last()**}>80
Alta	Temperature on {HOST.HOST} is > 45	({TRIGGER.VALUE}=0 and {Template HP A5120:switch.temp.**last()**}>45) or ({TRIGGER.VALUE}=1 and {Template HP A5120:switch.temp.**last()**}>40)
Informação	{HOSTNAME} Has been restarted	{Template HP A5120:SysUptime.**last**(0)}<1000

Figura 43 - Triggers padrão do Template HP A5120

O *template* em questão também vem com uma regra de descoberta por padrão, chamada "*Network interface erros"*, essa regra possuí apenas um protótipo de item padrão, demonstrado na Figura 44.

Nome ▲	Chave
CRC errors on interface {#SNMPVALUE}	CRC.Errors[{#SNMPVALUE}]

Figura 44 - Protótipos de itens padrão do Template HP A5120

Para atender os requisitos levantados pela diretoria para monitoramento desses equipamentos, foi preciso realizar algumas alterações no *template*, foram realizados apenas ajustes nas *triggers* do *template* padrão, as novas *triggers* do *template* estão demonstradas na Figura 45.

Severidade	Nome	Expressão
Alta	Temperature on {HOST.HOST} is > 45	({TRIGGER.VALUE}=0 and {Template HP A5120 com Interfaces:switch.temp.**last()**}>45) or ({TRIGGER.VALUE}=1 and {Template HP A5120 com Interfaces:switch.temp.**last()**}>40)
Alta	Memory usage on {HOST.HOST} > 90%	{Template HP A5120 com Interfaces:switch.cpu.**last()**}>90
Alta	CPU usage on {HOST.HOST} > 90%	{Template HP A5120 com Interfaces:switch.cpu.**last()**}>90
Informação	{HOSTNAME} Has been restarted	{Template HP A5120 com Interfaces:SysUptime.**last**(0)}<1000

Figura 45 - Novas triggers do Template HP A5120

Também foi removida a regra de descoberta "*Network interface erros"*.

Para monitorar os tráfegos de rede em todas as interfaces dos switches foi também utilizado o *template* de monitoramento de interfaces de rede via SNMP padrão do Zabbix, chamado "***Template* SNMP Interfaces**". O *Template* SNMP Interfaces utiliza SNMPv2, por tanto utiliza como autenticação apenas o nome da *Community* SNMP para monitorar as

interfaces do equipamento, esse parâmetro é repassado através do macro: {$SNMP_COMMUNITY}. Este *template* se baseia apenas em uma regra de descoberta referente a interfaces de rede e possui 8 protótipos de itens padrão que não precisaram ser alterados, e estão demonstrados na Figura 46.

Nome ▲	Chave
Admin status of interface {#SNMPVALUE}	ifAdminStatus[{#SNMPVALUE}]
Alias of interface {#SNMPVALUE}	ifAlias[{#SNMPVALUE}]
Description of interface {#SNMPVALUE}	ifDescr[{#SNMPVALUE}]
Inbound errors on interface {#SNMPVALUE}	ifInErrors[{#SNMPVALUE}]
Incoming traffic on interface {#SNMPVALUE}	ifInOctets[{#SNMPVALUE}]
Operational status of interface {#SNMPVALUE}	ifOperStatus[{#SNMPVALUE}]
Outbound errors on interface {#SNMPVALUE}	ifOutErrors[{#SNMPVALUE}]
Outgoing traffic on interface {#SNMPVALUE}	ifOutOctets[{#SNMPVALUE}]

Figura 46 - Protótipos de Itens do Template SNMP Interfaces

Esse *template* também possui apenas 1 protótipo de trigger, demonstrado na Figura 47.

Severidade	Nome ▲	Expressão
Informação	interface {#SNMPVALUE} changed status	{Template SNMP Interfaces:ifOperStatus[{#SNMPVALUE}].diff(0)}=1

Figura 47 - Protótipo de triggers do Template SNMP Interfaces

A etapa seguinte foi configurar o protocolo SNMP diretamente nos *switches* os comandos utilizados na configuração estão descritos abaixo.

Configurado o Grupo para o SNMPv3, onde GrupoSNMPv3 é o nome do grupo:

```
snmp-agent group v3 GrupoSNMPv3 privacy
```

Configurado o usuário, seu respectivo grupo, o modo de autenticação e senha e o modo de privacy e a senha, onde:

- Usuário: UsuarioSNMPv3
- Grupo: GrupoSNMPv3
- Modo de autenticação: SHA
- Senha de autenticação: SenhaAutenticacao
- Modo *privacy*: AES

- Senha *privacy*: SenhaPrivacy

Dessa forma foi utilizado o comando:

```
snmp-agent usm-user v3 UsuarioSNMPv3 GrupoSNMPv3 authentication-mode sha SenhaAutenticacao privacy-mode aes128 SenhaPrivacy
```

Após configurado o SNMP foi configurado cada switch no Zabbix configurando os macros do *Template* HP A5120 e do *Template* SNMP Interfaces. Conforme demonstrado na Figura 48.

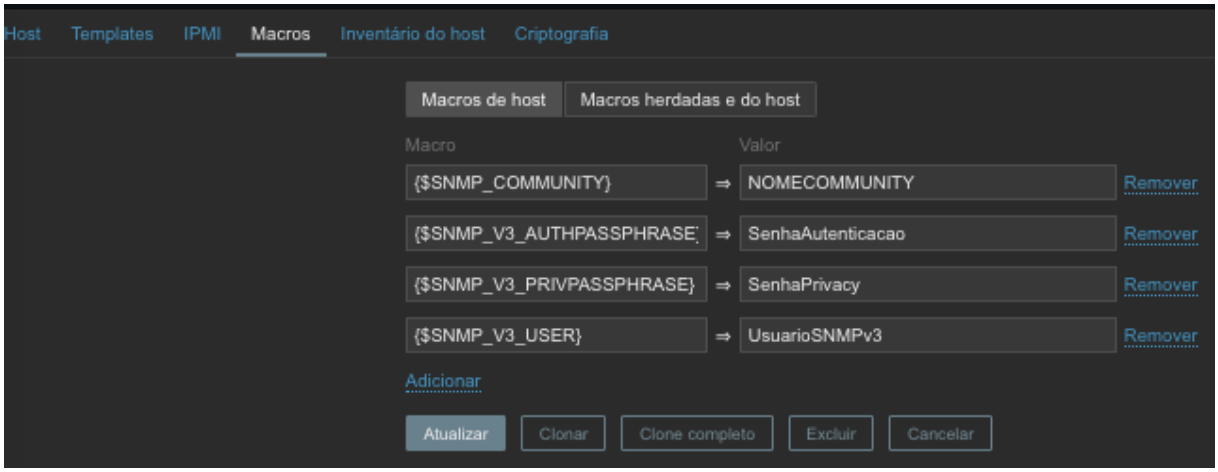


Figura 48 - Macros configurados para o monitoramento de Switches HP A5120

Após configurados todos os switches passaram a ser monitorados através do Zabbix segundo os itens dos *templates Template* HP A5120 e *Template* SNMP *Interfaces*, descritos anteriormente.

4.5 NOTIFICAÇÃO DE ALERTAS DE INCIDENTES NO ZABBIX VIA E-MAIL

Após terem sidos configurados todos os ativos de rede relevantes no Zabbix foi preciso realizar a configuração das notificações dos alertas de incidentes do Zabbix.

Por padrão o Zabbix já suporta o envio de notificações por E-Mail, sendo necessário apenas algumas configurações simples. A primeira etapa, foi realizar a configuração do Tipo de Midia do sistema de envio de E-Mails, com as configurações do SMTP da organização, o exemplo dessas configurações é demonstrado na Figura 49. Os dados sensíveis da organização foram ocultados devido as normas de segurança das informações implementadas na empresa.

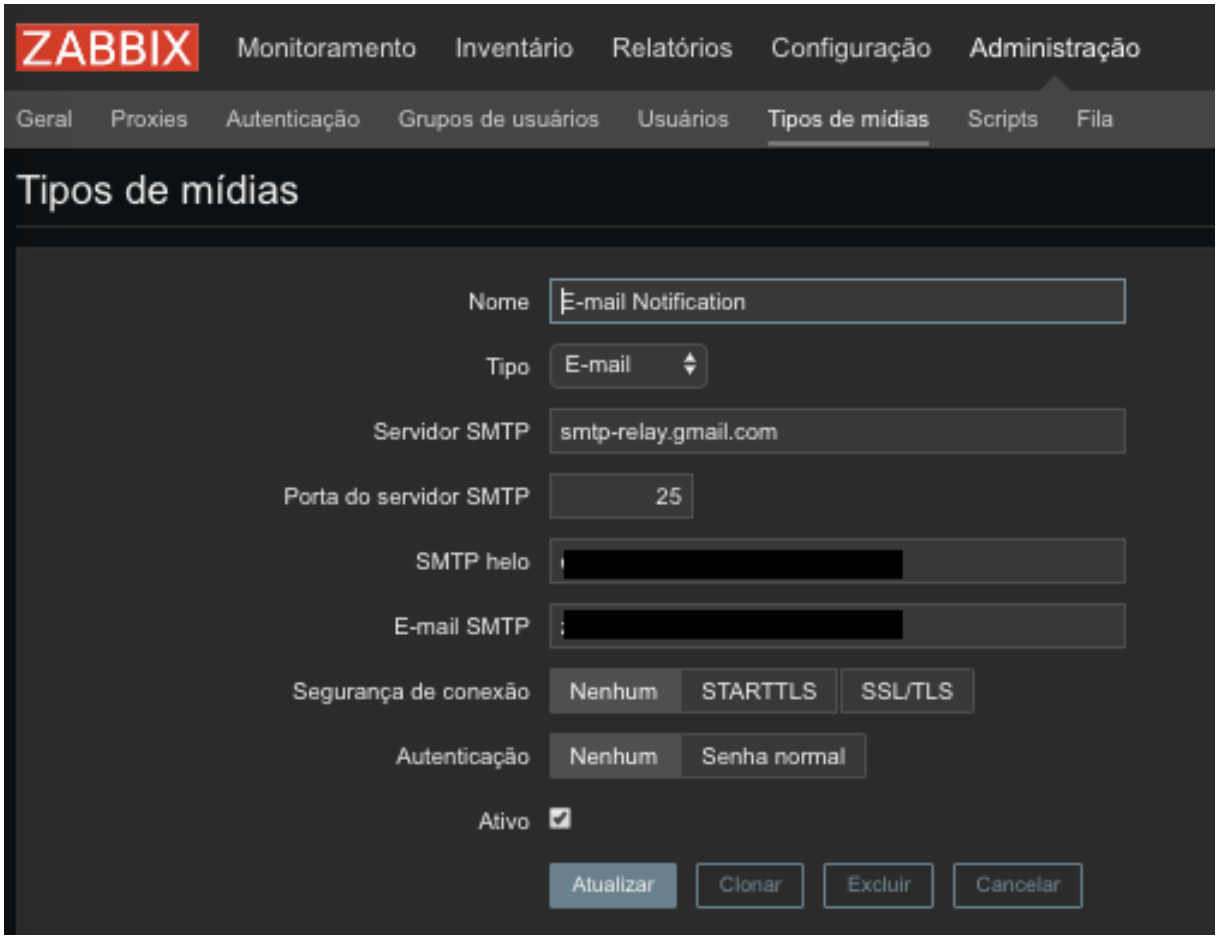


Figura 49 - Configurações do tipo de midia E-mail Notification

Após terem sido realizadas as configurações de SMTP corretamente para o tipo de mídia , é necessário configurar uma Ação para o envio de alertas utilizando esse tipo de mídia, definindo as condições necessárias para que essa ação seja realizada.

Para o envio de e.mails E-Mails aos administradores de rede da empresa, foi criado uma ação chamada "**E-MAIL - Administrators**" e configurada a mesma para enviar mensagens utilizando o tipo de mídia "E-mail Notification" para os membros do grupo "Zabbix Administrators" caso ocorra algum incidente de severidade "Alta" ou "Desastre" as configurações desta ação estão demonstradas nas figuras 50, 51 e 52.

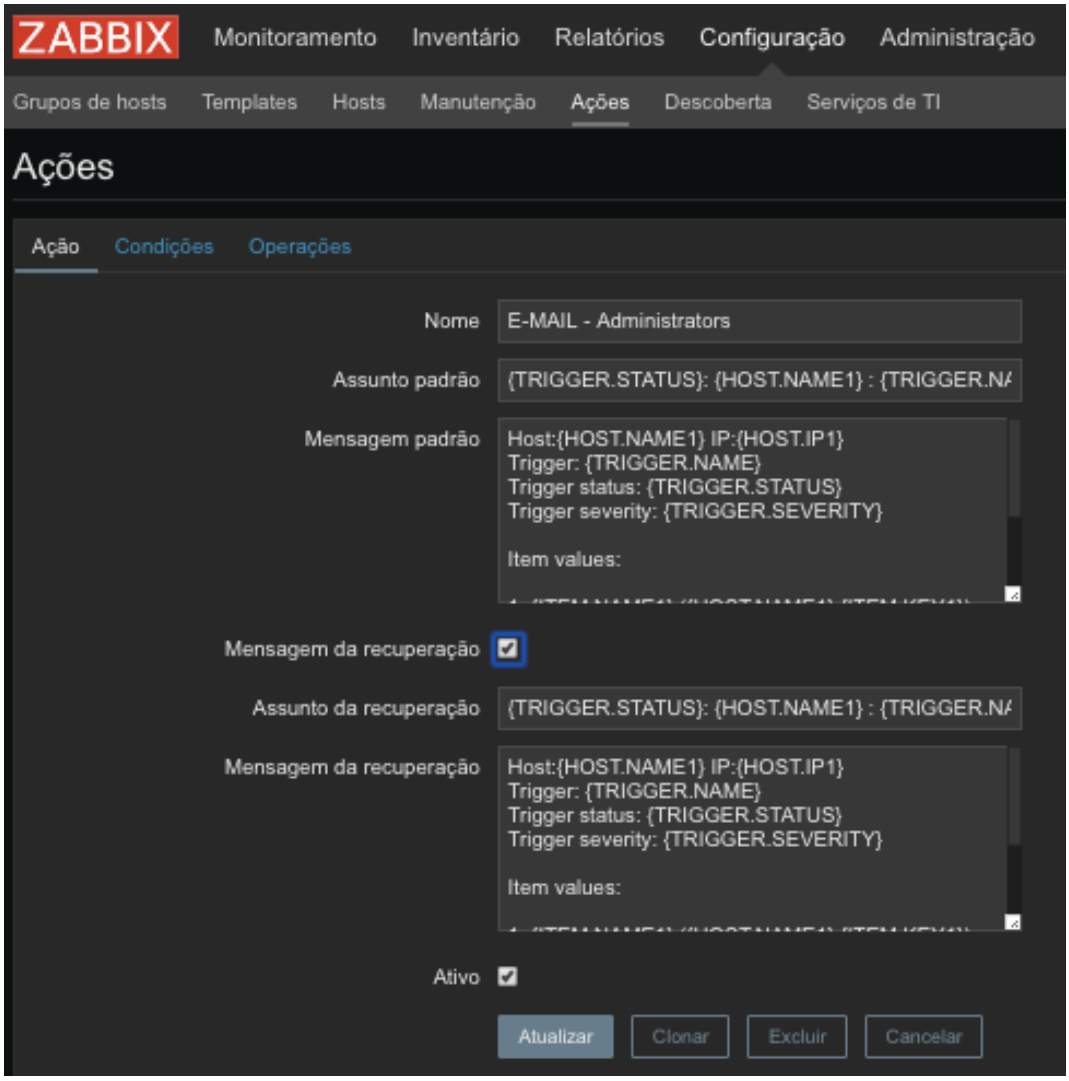


Figura 50 - Configuração da Ação E-MAIL – Administrators

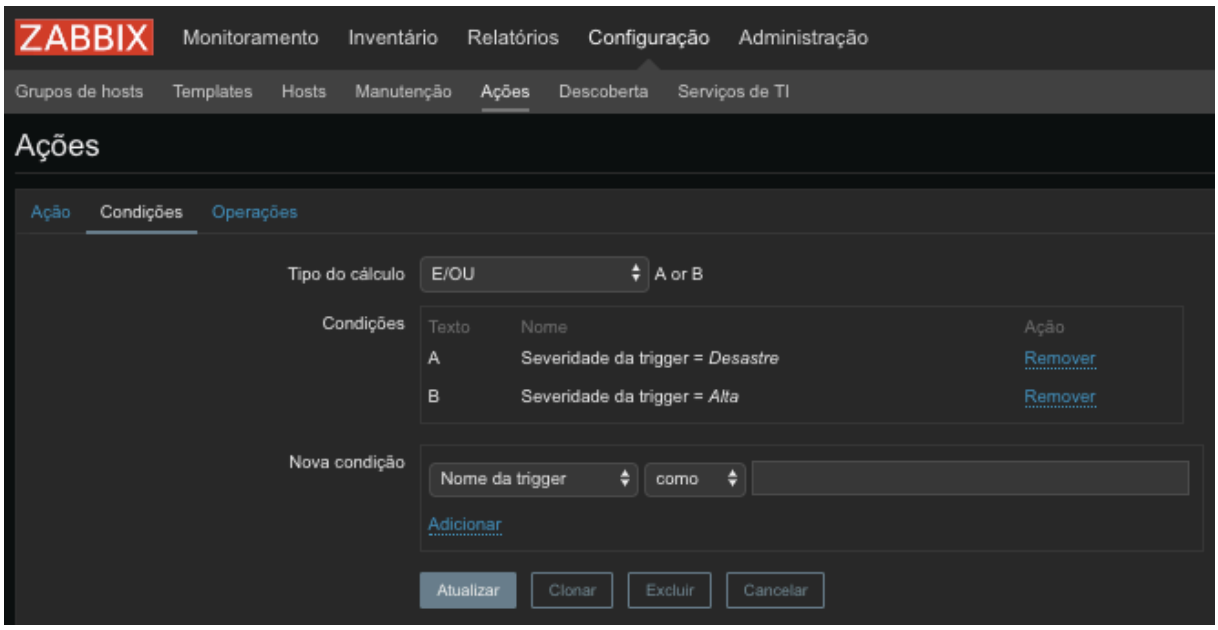


Figura 51 - Condições da Ação E-MAIL – Administrators

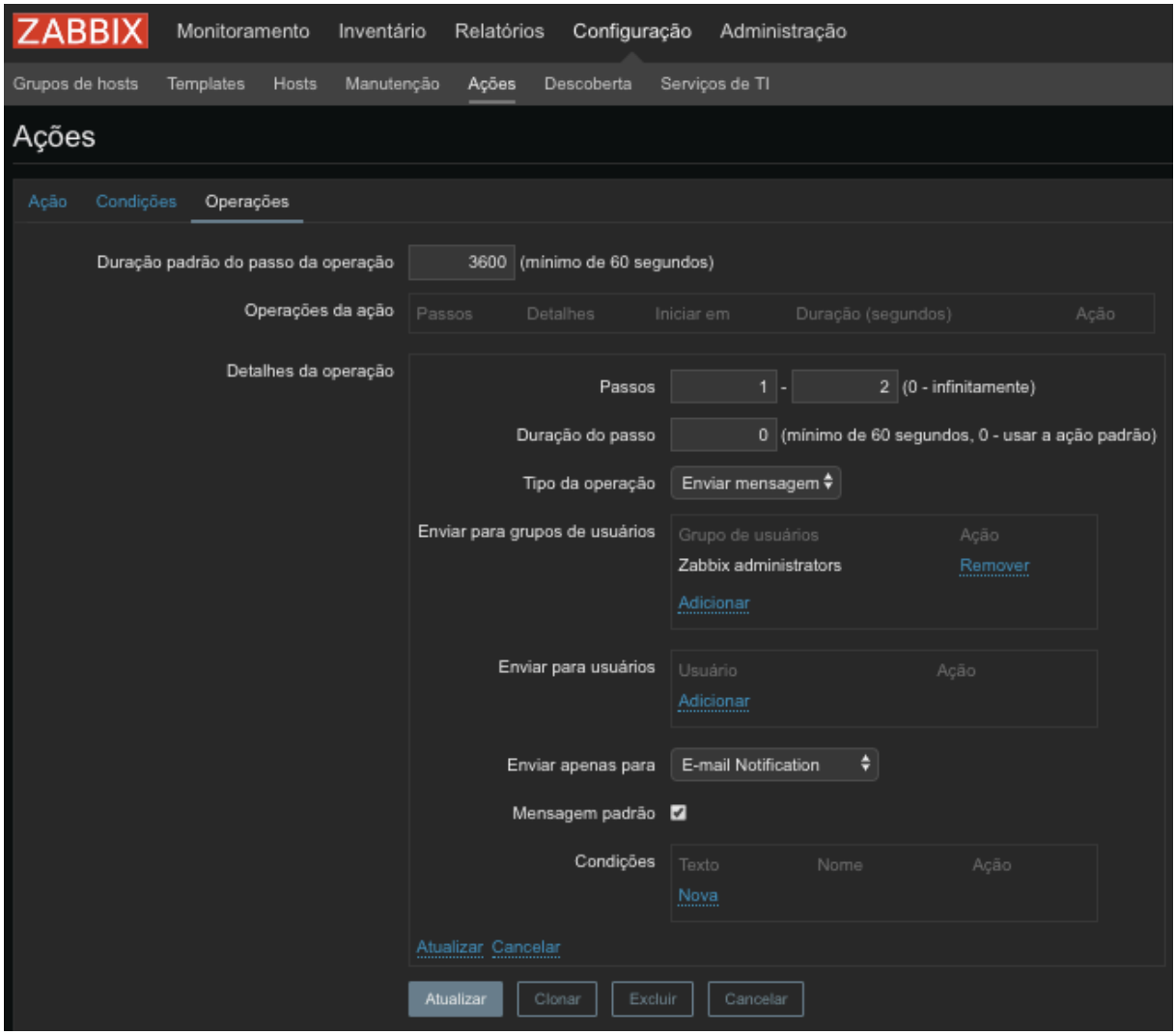


Figura 52 - Operações da ação E-MAIL – Administrator

A última etapa da configuração das notificações via E-mail é configurar o tipo de mídia "*E-mail Notification*" para os usuários do grupo "*Zabbix Administrators*" a Figura 53 representa a configuração em questão realizada para um dos usuários.

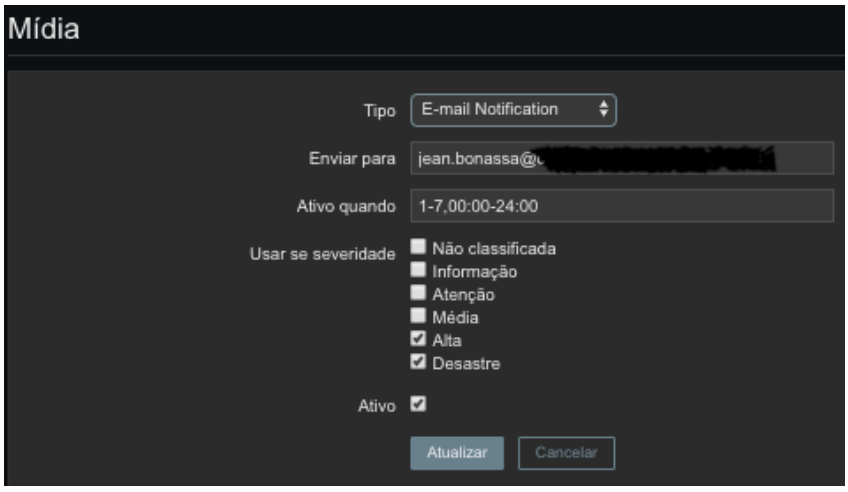


Figura 53 - Configuração da Midia E-mail Notification no usuário do Zabbix

Após terem sido realizadas as configurações foi realizado um teste simulando um incidente em um servidor de teste para validar o correto envio de e-mails dos incidentes e também do envio do E-mail de recuperação do incidente. A Figura 54 demonstra o e-mail enviado pelo Zabbix notificando o incidente da simulação em questão.

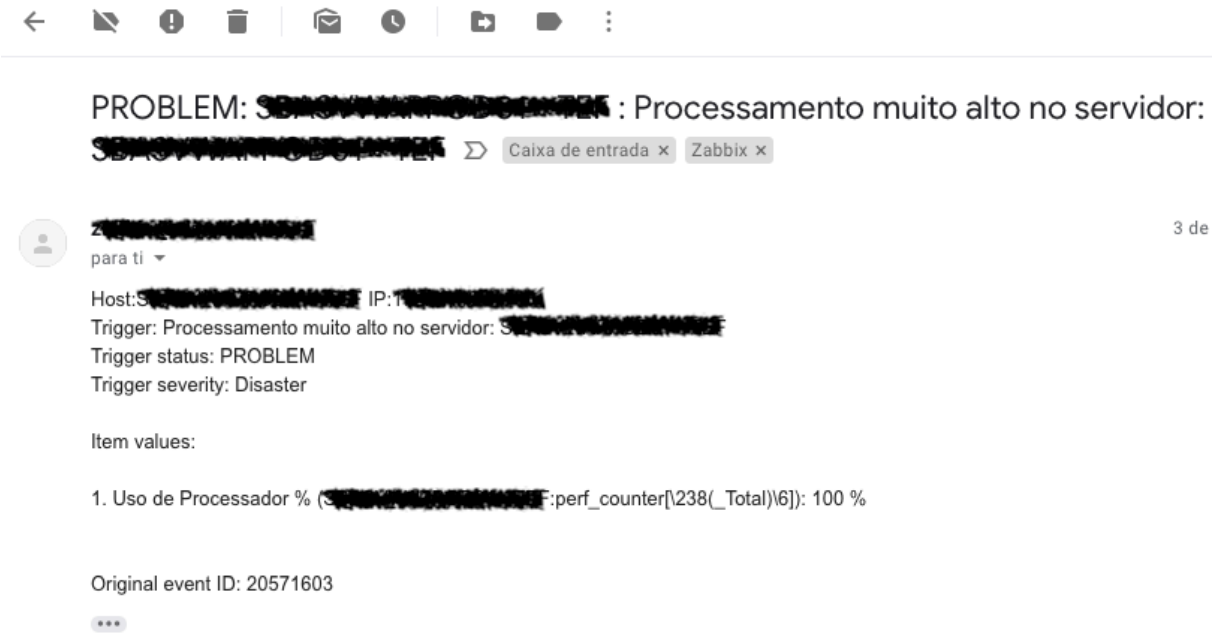


Figura 54 - E-mail de notificação de incidente do Zabbix

Logo em seguida o incidente de teste foi corrigido, gerando assim um E-mail de recuperação, mostrado na Figura 55.

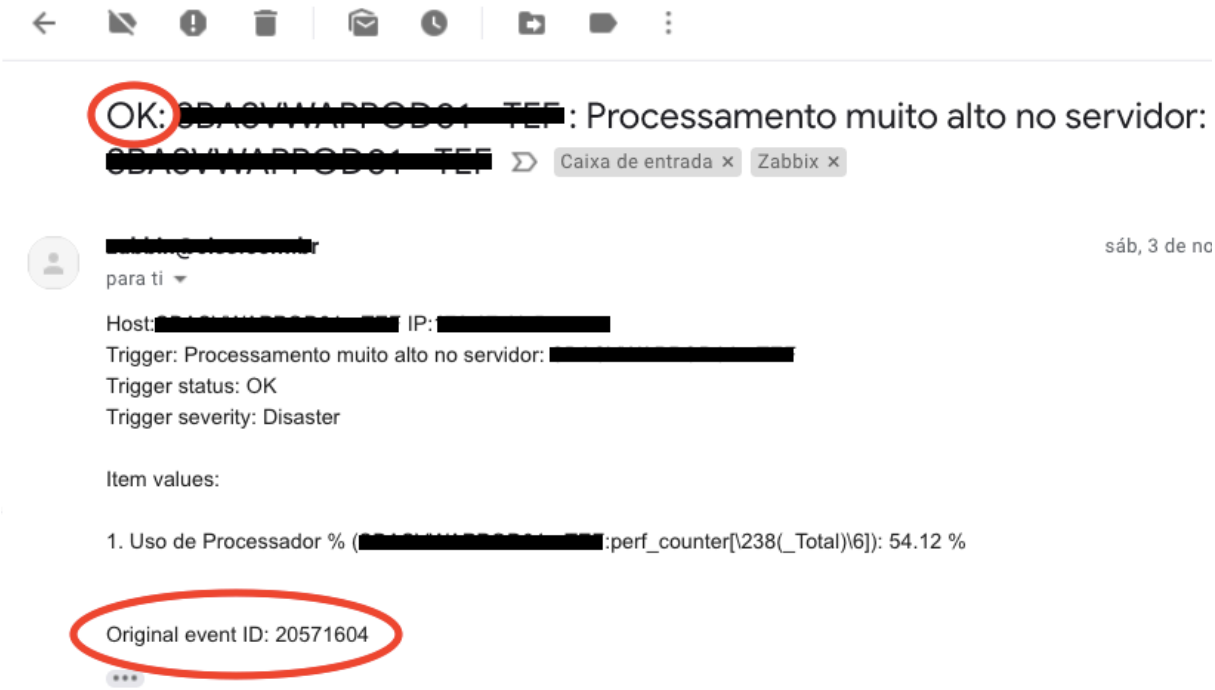


Figura 55 - E-mail de recuperação de incidente do Zabbix

Nota-se nas figuras 54 e 55 que o campo "*Original event ID*:" (Identificador do evento original) é o mesmo, tanto no E-mail de notificação do incidente, quanto no E-mail de recuperação. Esse dado refere-se ao identificador interno do Zabbix referente a cada incidente gerado no Zabbix. Também é possível notar que o primeiro campo do título da mensagem representa o *status* da *trigger* ou seja, caso seja um E-mail de notificação de um incidente, o título do E-mail inicia com "*PROBLEM*" (Problema) conforme na Figura 54. e caso seja uma mensagem de recuperação o título do E-mail inicia com "OK" conforme na Figura 55. Os dados referentes a *Hostname*, IP e o E-mail utilizado pelo Zabbix da empresa foram ocultados devido as normas de segurança da informação definidas pela empresa.

4.6 INTEGRAÇÃO COM FERRAMENTA FRESHSERVICE PARA ABERTURA DE TICKETS SOBRE INCIDENTES

O *Data Center* em questão utiliza o sistema Freshservice para gestão de *tickets* de suporte, essa ferramenta possui a capacidade de abertura de *tickets* de incidentes enviados via E-mail. Dito isso foi julgado viável realizar a configuração para abertura de *tickets* para a equipe técnica do *Data Center*, diretamente pelo Zabbix, utilizando o E-mail enviado pelo Zabbix para notificar um incidente, sendo assim foi realizada a configuração de um endereço de E-mail para o qual, ao enviar um E-mail o seja aberto um novo *ticket* na ferramenta Freshservice e foi configurado uma ação no Zabbix para enviar um novo E-mail para cada novo incidente ao endereço configurado na ferramenta Freshservice e abrir um ticket, com os dados do incidente. A configuração utilizada para a ação de abertura de *tickets* é descrita na Figura 56.

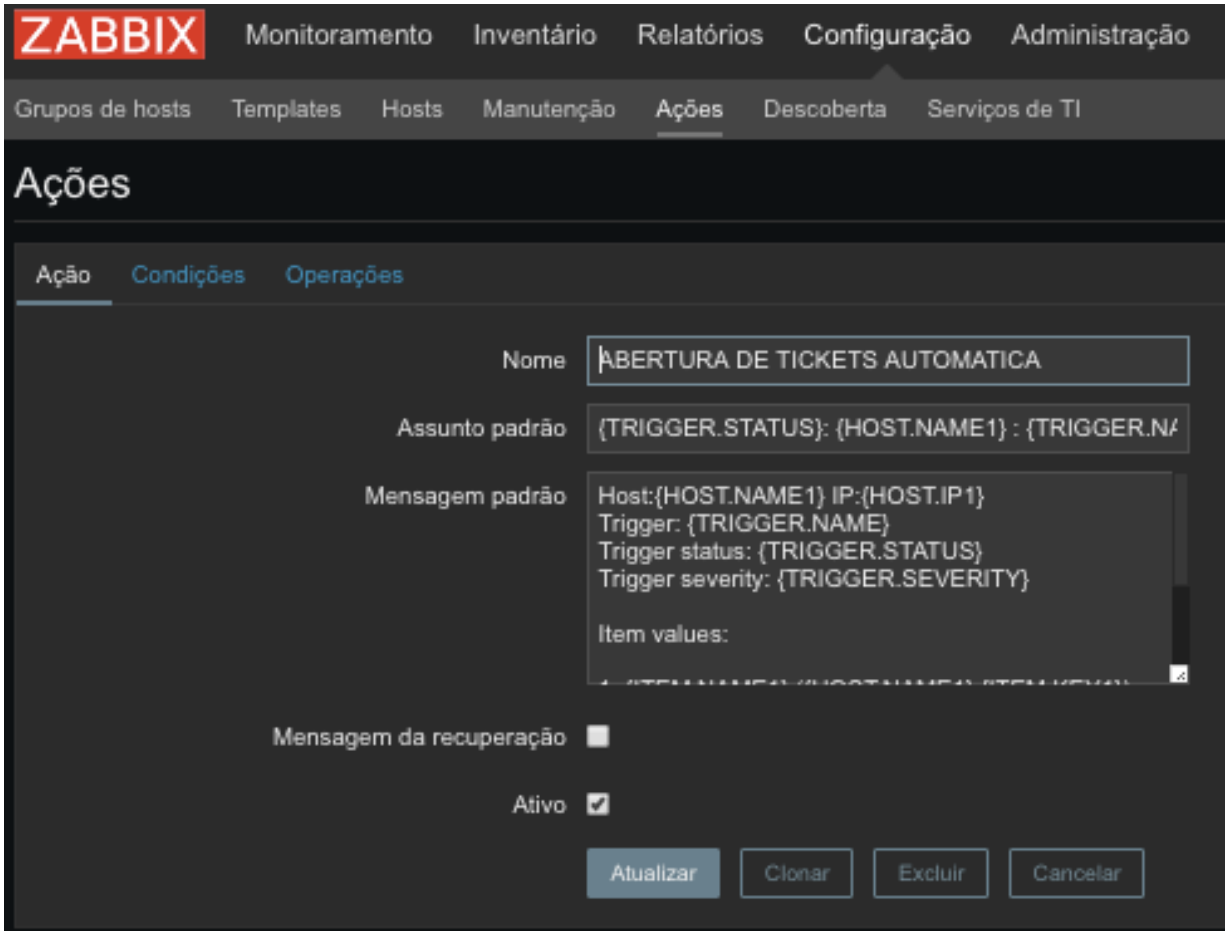


Figura 56 - Ação para a abertura de tickets automática na ferramenta Freshservice

Para a ação de abertura de *tickets* foi configurado que não seja enviada a mensagem de recuperação, pois se não seria aberto um novo *ticket* com a mensagem de recuperação. O que não é interessante para o contexto de trabalho do *Data Center*.

Para essa ação de abertura automática de *tickets* foi configurado para que sejam abertos apenas *tickets* referentes a incidentes de severidades Alta e Desastre, conforme pode ser visualizado na Figura 57.

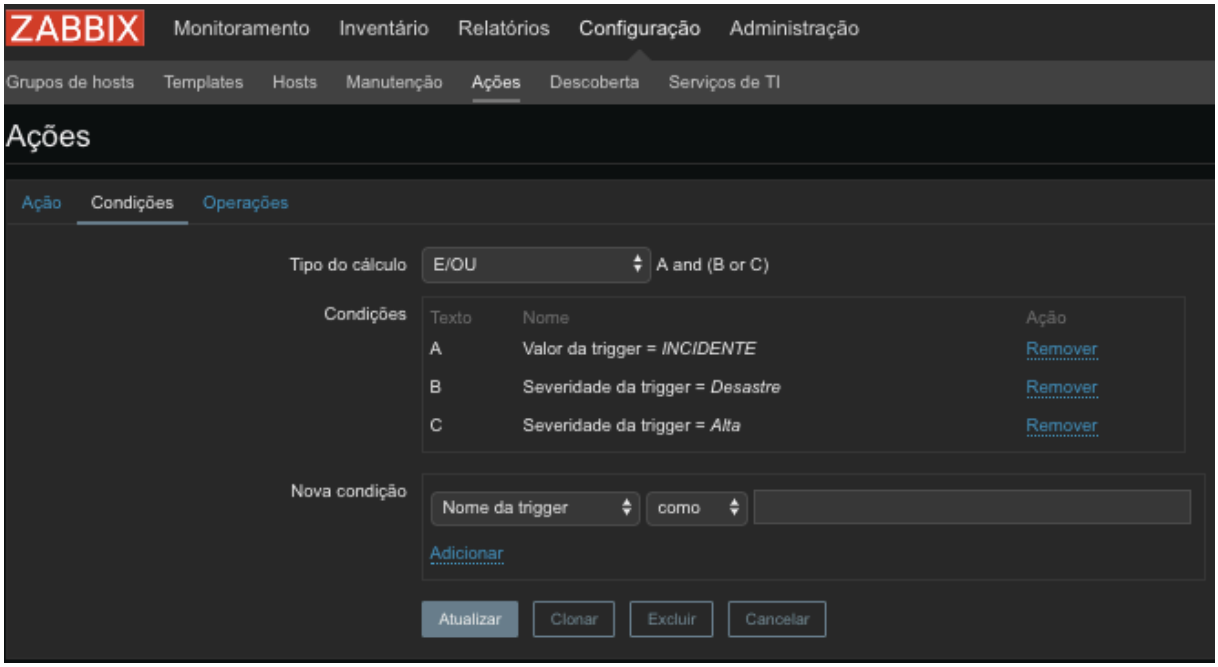


Figura 57 - Condições da Ação para abertura automática de tickets

Também foi configurada a operação para enviar a notificação dos incidentes dessa ação para um usuário chamado “Tickets”, e, nesse usuário, foi configurado o e-mail para onde são enviados os tickets utilizando o tipo de mídia “E-mail Notification” semelhantemente ao descrito na Figura 53. A operação configurada para a ação de abertura de tickets está descrita na Figura 58.

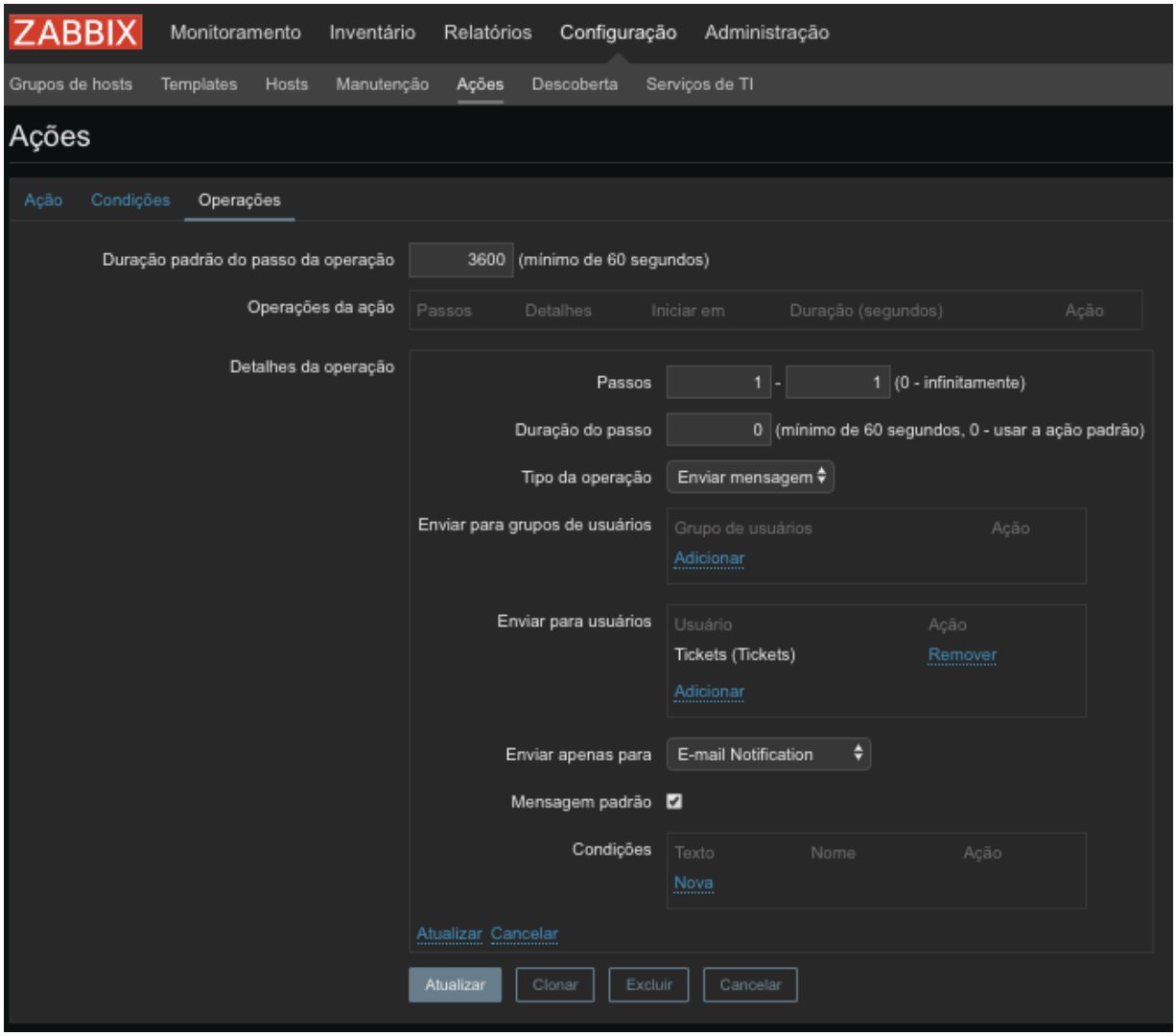


Figura 58 - Operação configurada para a ação de abertura de tickets

Após terem sido concluídas as configurações no Zabbix e ativada a abertura de tickets por e-mail no Freshservice foi realizado um teste para validar se o funcionamento da abertura de tickets estava correto. A tela do painel do ticket, criado para o teste, é exibida na Figura 59.

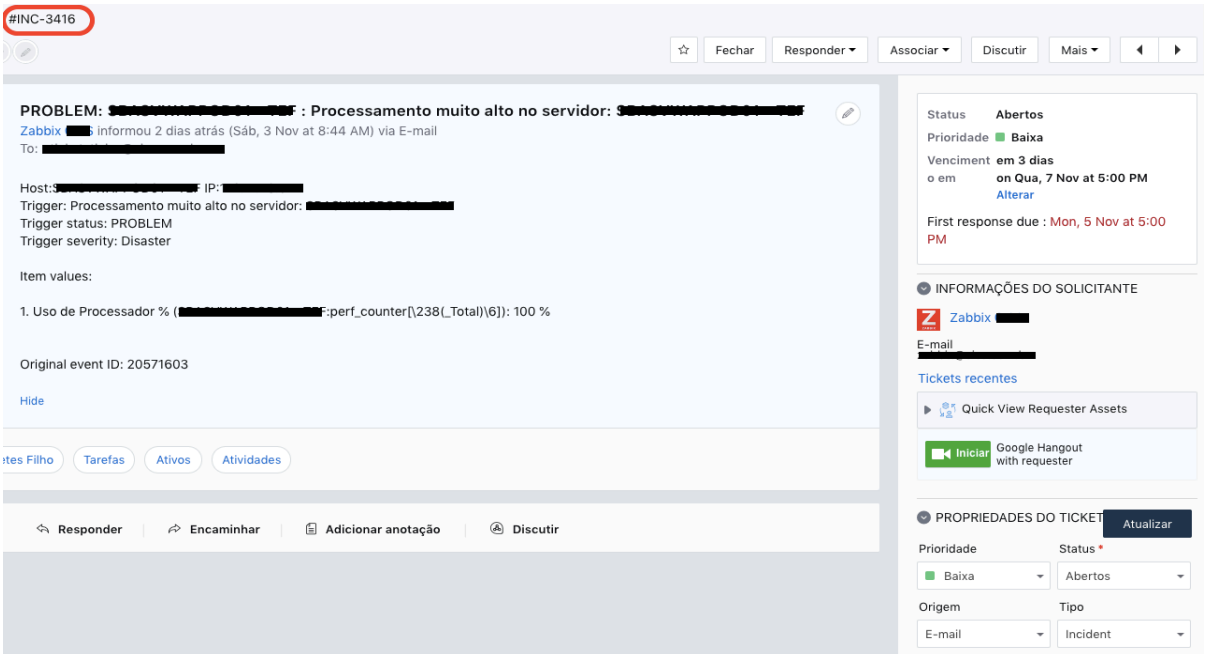


Figura 59 - Ticket de teste da integração com o Freshservice

Os dados referentes a *Hostname*, IP e o e-mail utilizado pelo Zabbix da empresa foram ocultados devido as normas de segurança da informação definidas pela empresa.

4.7 INTEGRAÇÃO COM O TELEGRAM PARA ENVIO DE ALERTAS

Para atender uma grande necessidade do *Data Center*, foi preciso encontrar uma forma de enviar notificações aos administradores de rede de forma personalizada, em qualquer lugar que estejam, anteriormente em outras aplicações havia sido utilizado o envio de mensagens via SMS, porém a integração com o sistema de SMS da empresa não se mostrou confiável. Além da telefonia móvel ter um custo mais elevado. Por esse fato e graças aos administradores de rede da empresa sempre terem acesso a uma conexão de dados disponível, foi identificado como melhor solução enviar essas notificações por aplicativos de mensagens instantâneas, e assim se chegou ao aplicativo Telegram, que através das pesquisas realizadas no Capítulo 3 do presente trabalho foi julgado a melhor ferramenta para realizar essa integração.

O primeiro passo necessário para realizar essa integração, foi criar um *bot* no Telegram, que será o responsável pelo envio das mensagens para os administradores da rede. Para essa criação existe o Telegram BotFather, que é um *bot* padrão do Telegram para a criação de outros *Bots* para automações.

Para criar o *bot*, foi necessário através do aplicativo Telegram, iniciar uma conversação com o BotFather e efetuar os comandos para a criação de um novo *bot* para o Zabbix da

empresa, os comandos realizados estão demonstrados nas figuras 60 e 61, alguns dados sigilosos da empresa foram ocultados por questões de segurança da informação.

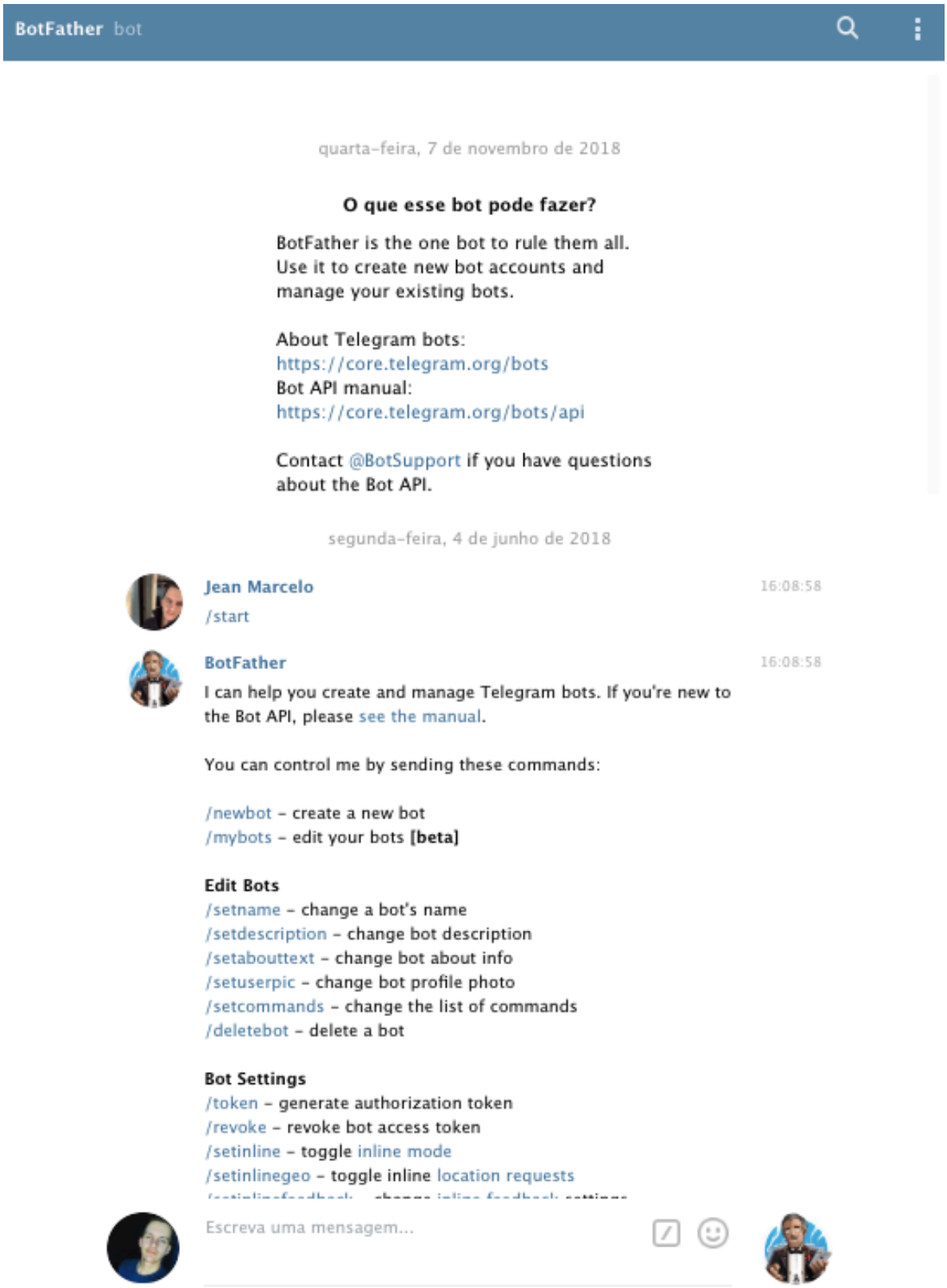


Figura 60 - Criação do bot para a integração com o Zabbix parte 1

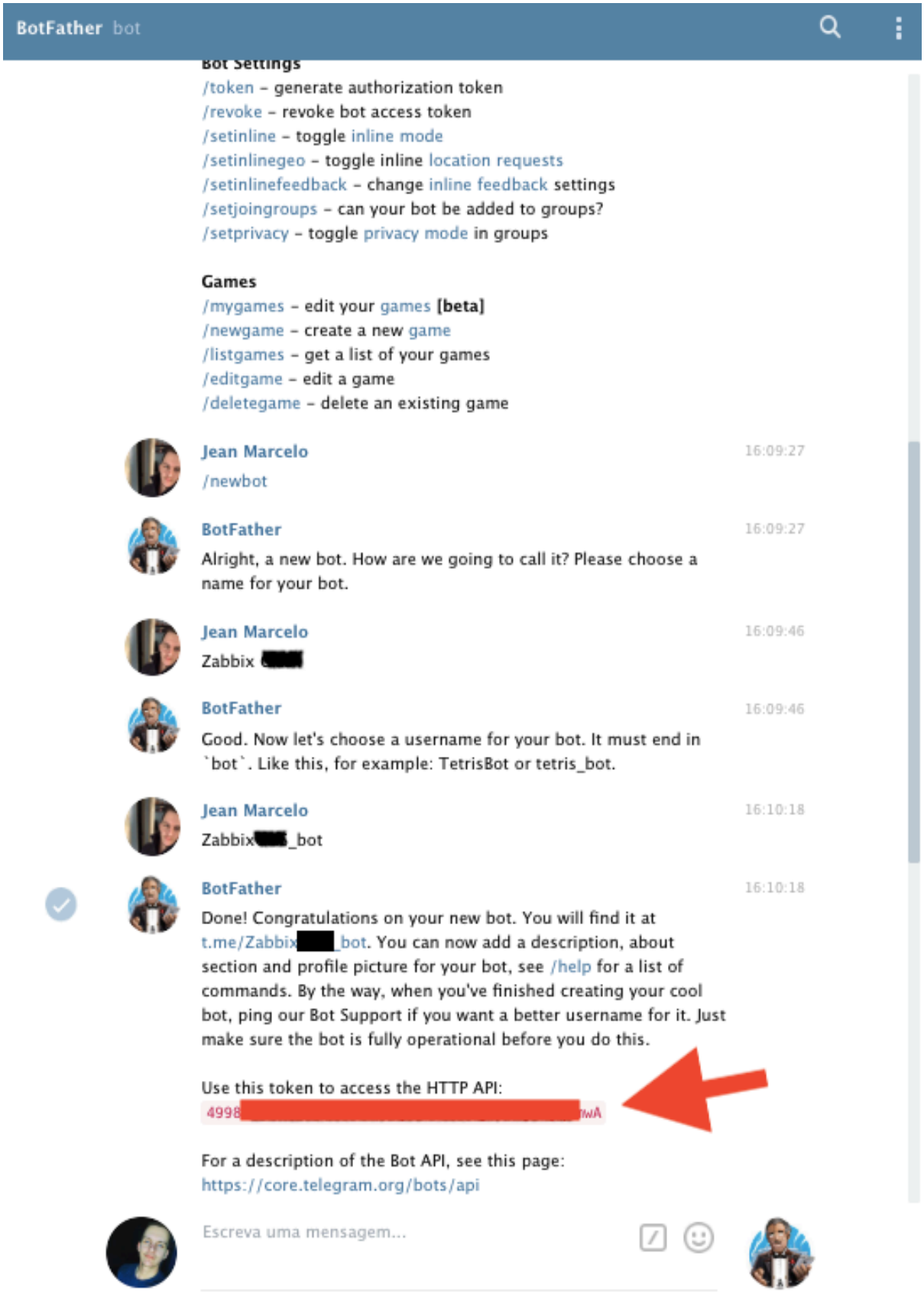


Figura 61 - Criação do bot para a integração com o Zabbix parte 2

Após realizar a criação do *bot* para realizar essa integração, por não ser uma integração que vem disponível na versão padrão do Zabbix, se fez necessário a utilização de *scripts* para

realizar o envio de mensagens pelo aplicativo externo, então foi necessário configurar o diretório de onde o Zabbix Server faria a leitura dos *scripts*, editando o arquivo de configuração e realizando a seguinte configuração descrita na Figura 62.

```
### Option: AlertScriptsPath
#       Full path to location of custom alert scripts.
#       Default depends on compilation options.
#
# Mandatory: no
# Default:
# AlertScriptsPath=${datadir}/zabbix/alertscripts

AlertScriptsPath=/usr/lib/zabbix/alertscripts
```

Figura 62 - Configuração do diretório de scripts no Zabbix Server

Após configurar o diretório na configuração do Zabbix, foi preciso reiniciar o serviço do Zabbix Server para aplicar as configurações. Após isso, foi necessário efetuar a instalação dos scripts necessários para a integração. Dentro do diretório configurado para os scripts informado no arquivo de configuração, foram utilizados os comandos listados na Figura 63 para baixar os scripts necessários.

```
wget https://goo.gl/JB2PSy -O telegram-notify.sh
wget https://goo.gl/tAhWvL -O telegram-getUpdates.sh
```

Figura 63 - Comandos utilizados para baixar os scripts de integração com o Telegram

Após baixar os scripts foi preciso dar permissão para que esses scripts possam ser executados, e alterar o proprietário dos arquivos para o usuário do Zabbix, para isso foram usados os seguintes comandos:

```
# chmod +x telegram-*.sh
# chown zabbix: telegram-*.sh
```

Alterar as permissões dos scripts, foi preciso configurar o bot criado nos scripts, para isso foi preciso criar um arquivo de texto dentro dessa mesma pasta, chamado "**botinfo.txt**" e dentro desse arquivo colar apenas o valor do hash correspondente ao bot criado, esse hash está representado em vermelho na Figura 61. A Figura 64 mostra como ficou a configuração do arquivo **botinfo.txt** nesse caso.

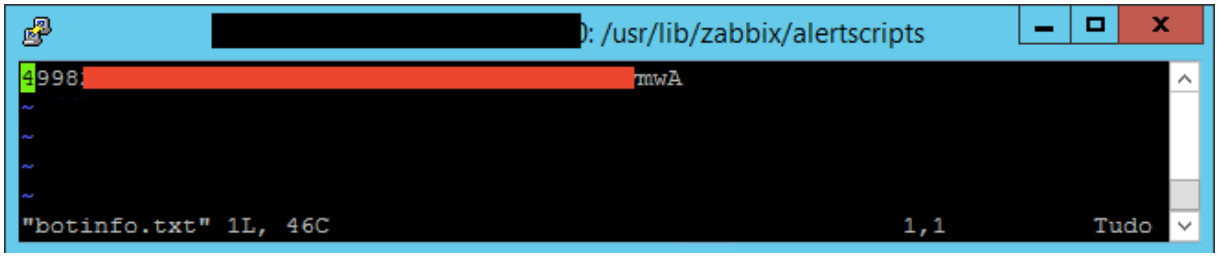


Figura 64 - Configuração do arquivo botinfo.txt

A configuração do arquivo **botinfo.txt** é necessária para definir o código do bot criado, pois esse código é consumido pela API do Telegram, o uso dessa API é feito através dos dois scripts baixados, "**telegram-getUpdates.sh**" e "**telegram-notify.sh**".

O Script "telegram-getUpdates.sh" é utilizado para se obter os códigos dos usuários do Telegram que enviam mensagens para o bot configurado. Esse código é necessário para configurar o tipo de mídia para os usuários, é o identificador único do usuário no Telegram. Para pegar esse código, bastou entrar no aplicativo Telegram e iniciar uma conversa com o Bot criado, e enviar algumas mensagens para ele, conforme mostra a Figura 65.

Figura 65 - Conversa iniciada com o Bot criado

Essas mensagens enviadas para o bot servem para que utilizando o script "**telegram-getUpdates.sh**" o código de identificação do usuário que enviou as mensagens seja capturado, assim sendo possível utilizar esse código para futuramente configurá-lo como um tipo de mídia. A Figura 66 apresenta como é capturado o código do usuário do Telegram.

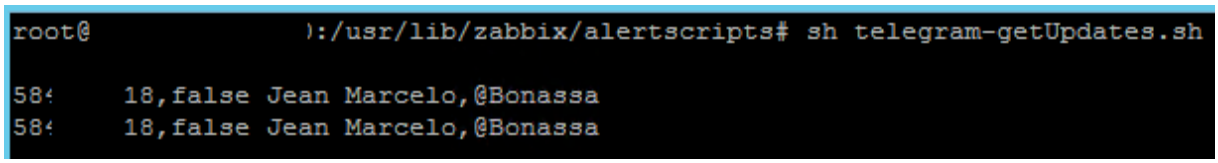


Figura 66 - Captura do código do Telegram do usuário

Após ter capturado o código do usuário do Telegram, foram realizados testes do envio de mensagens através do bot criado utilizando o script "**Telegram-notify.sh**" e passando por parâmetro, o código do usuário, o título da mensagem e o conteúdo da mensagem, conforme mostra a Figura 67.

```
# ./telegram-notify.sh 58[illegible]18 Titulo Mensagem
```

Figura 67 - Envio da mensagem de teste utilizando o script telegram-notify.sh

A configuração do script faz o envio de forma que o Título fica com a fonte em negrito, e o corpo da mensagem em fonte normal. A Figura 68 apresenta o exemplo das mensagens de teste recebidas via Telegram.

Figura 68 - Mensagem de teste recebida via Telegram

Após realizados os testes com sucesso foi criado um novo Tipo de mídia no Zabbix Server para o envio de mensagens de alerta utilizando o script para envio de mensagens via Telegram. A Figura 69 apresenta a configuração do tipo de mídia "*Telegram Integration*".

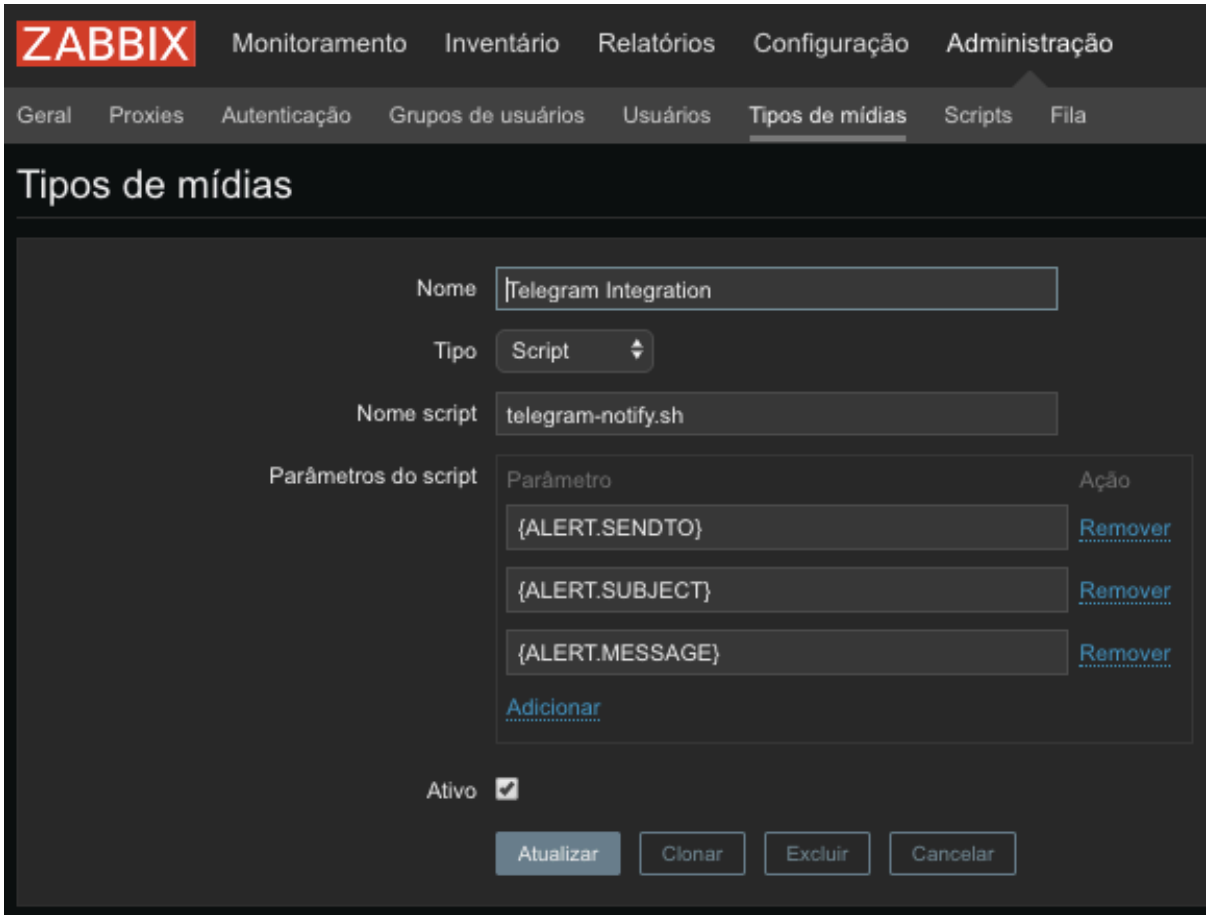


Figura 69 - Configuração do Tipo de mídia Telegram Integration

Após criar o Tipo de mídia no Zabbix, foi preciso criar um tipo de mídia para os usuários, onde foram apontados os códigos identificadores de cada usuário do Telegram que receberá notificações de incidentes do Zabbix. A Figura 70 demonstra a configuração do novo tipo de mídia para o usuário utilizado nos testes anteriores.

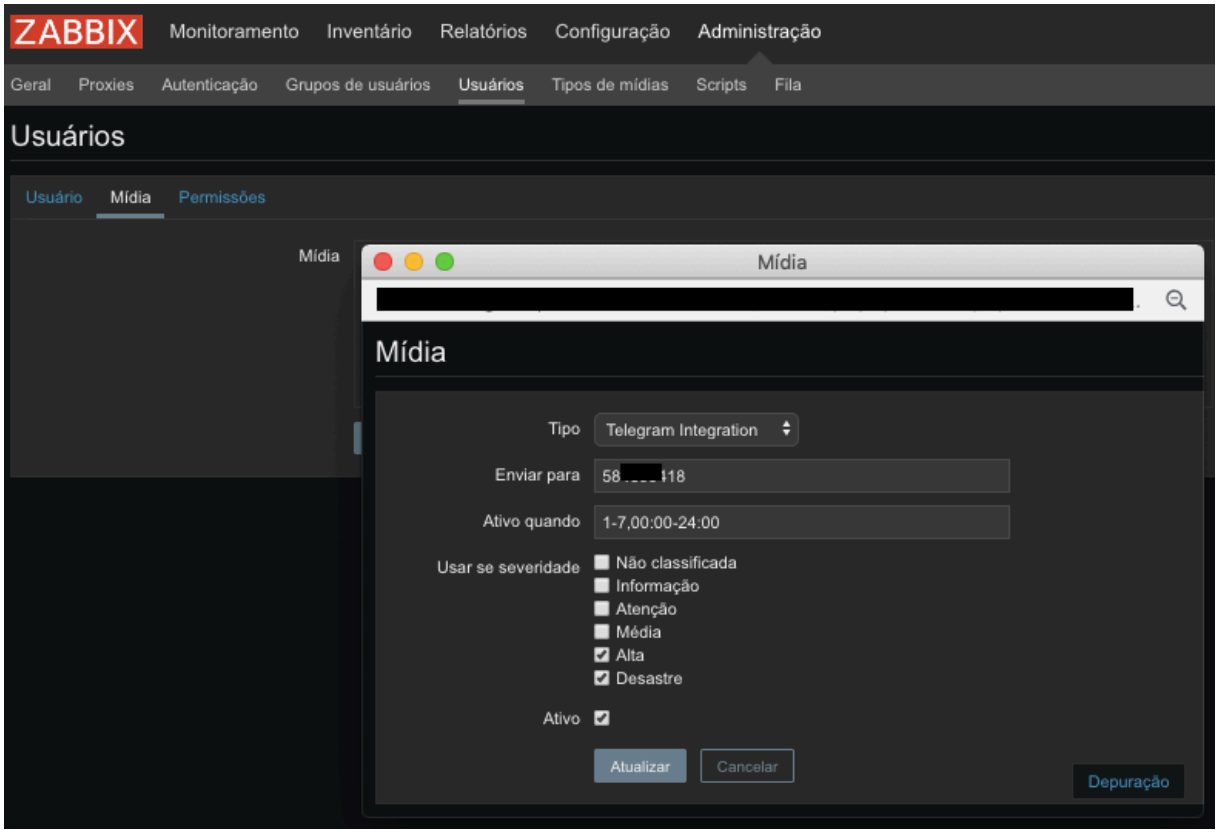


Figura 70 - Configuração do novo Tipo de mídia para o usuário.

Configurado o Tipo de mídia para o usuário, por fim foi criada uma ação para realizar o envio de notificações de incidentes do Zabbix via Telegram. Para isso foi criada uma nova ação chamada "*TELEGRAM NOTIFY*". A Figura 71 apresenta a configuração dessa ação.

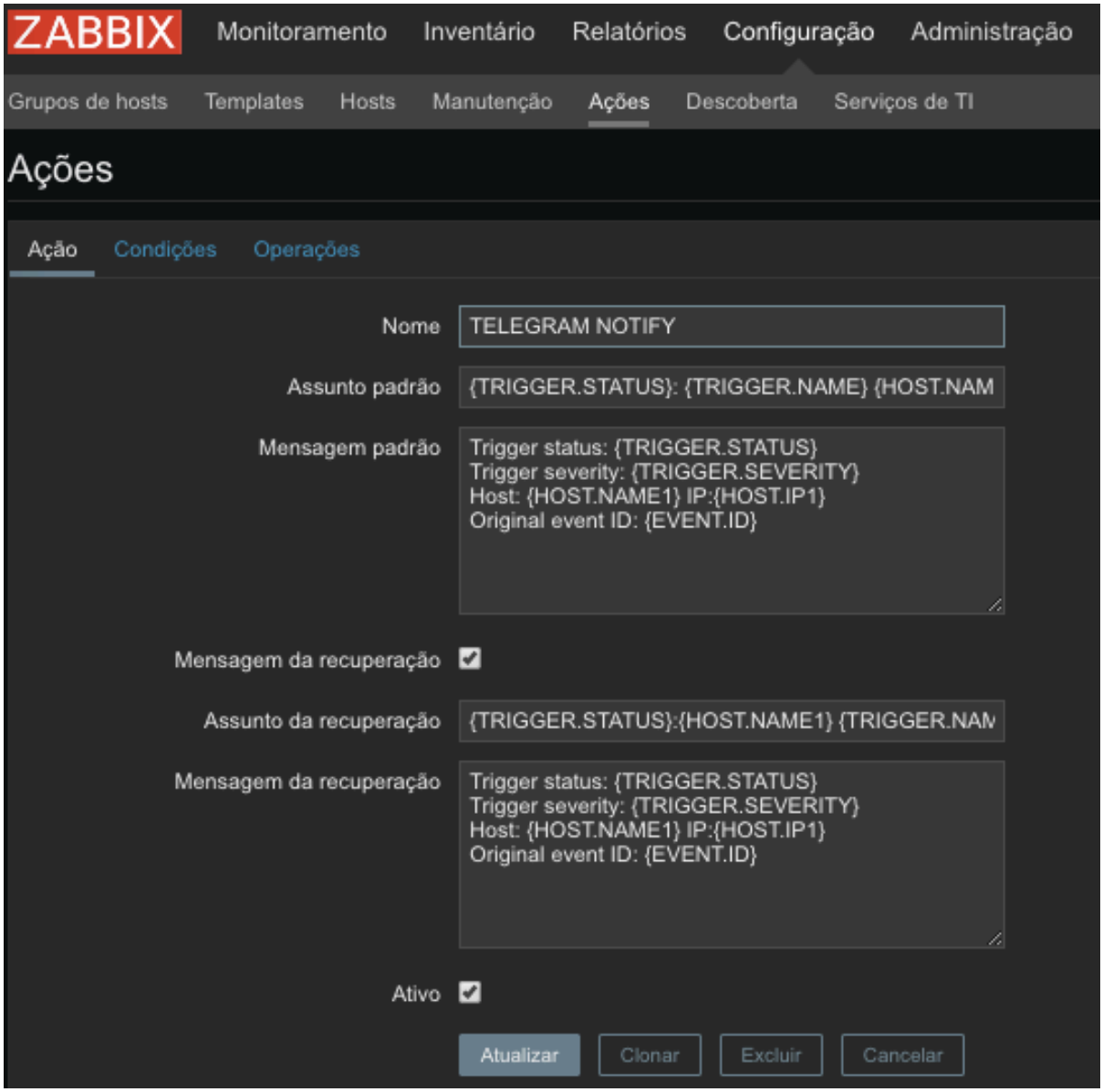


Figura 71 - Configuração da Ação para envio de alertas via Telegram.

Essa ação foi configurada para enviar alerta de qualquer incidente, sem discriminar o tipo de incidente ou severidade, pois a severidade mínima dos alertas já estava configurada no tipo de mídia cadastrado para cada usuário, então as condições dessa ação ficaram como mostra a Figura 72.

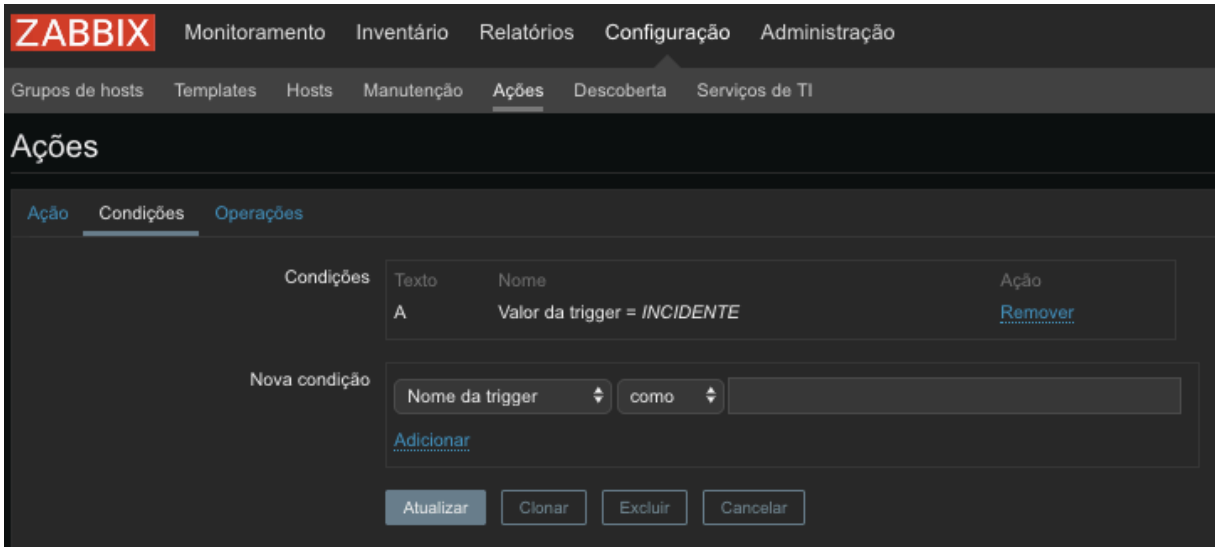


Figura 72 - Condições da ação para envio de alertas via Telegram

Por fim essa ação possui apenas uma operação, que é enviar uma mensagem utilizando o tipo de mídia "*Telegram Integration*" para os usuários do grupo "Zabbix Administrators" do Zabbix, aos quais já foi relacionado o código de identificação do usuário no Telegram para o envio de mensagens, conforme demonstrado um exemplo na Figura 70, que seriam no caso todos os membros da equipe técnica do *Data Center*. Essa configuração é demonstrada na Figura 73.

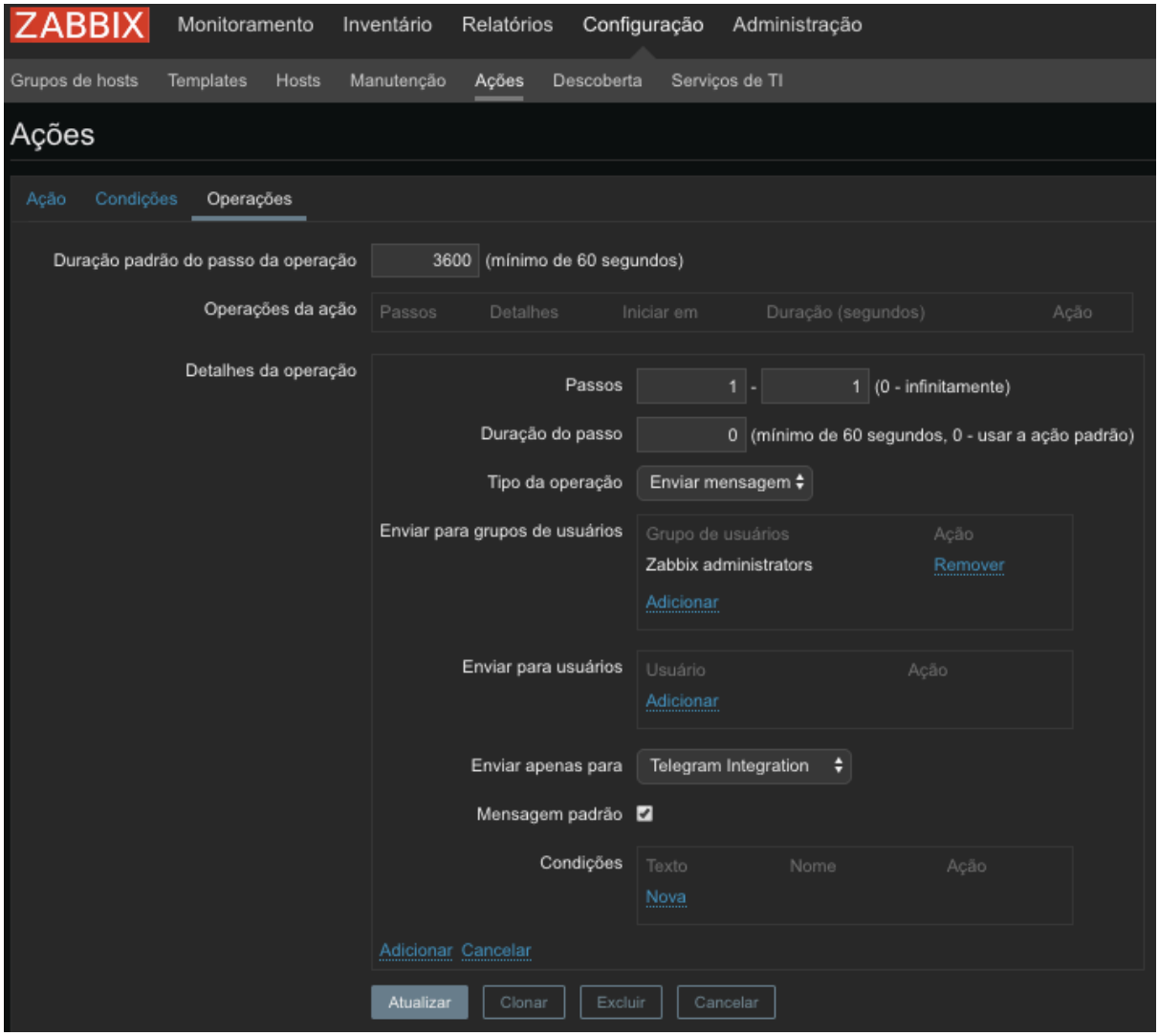


Figura 73 - Operações da Ação de envio de alertas via Telegram

Após concluídas todas as configurações do envio de alertas via Telegram, foram realizadas simulações de incidentes para testar o correto funcionamento da integração, os testes foram bem-sucedidos, sendo enviado o alerta via Telegram para todos os membros do grupo corretamente. A Figura 74 apresenta um exemplo de uma mensagem de notificação de um incidente simulado e também a mensagem de recuperação desse mesmo incidente.

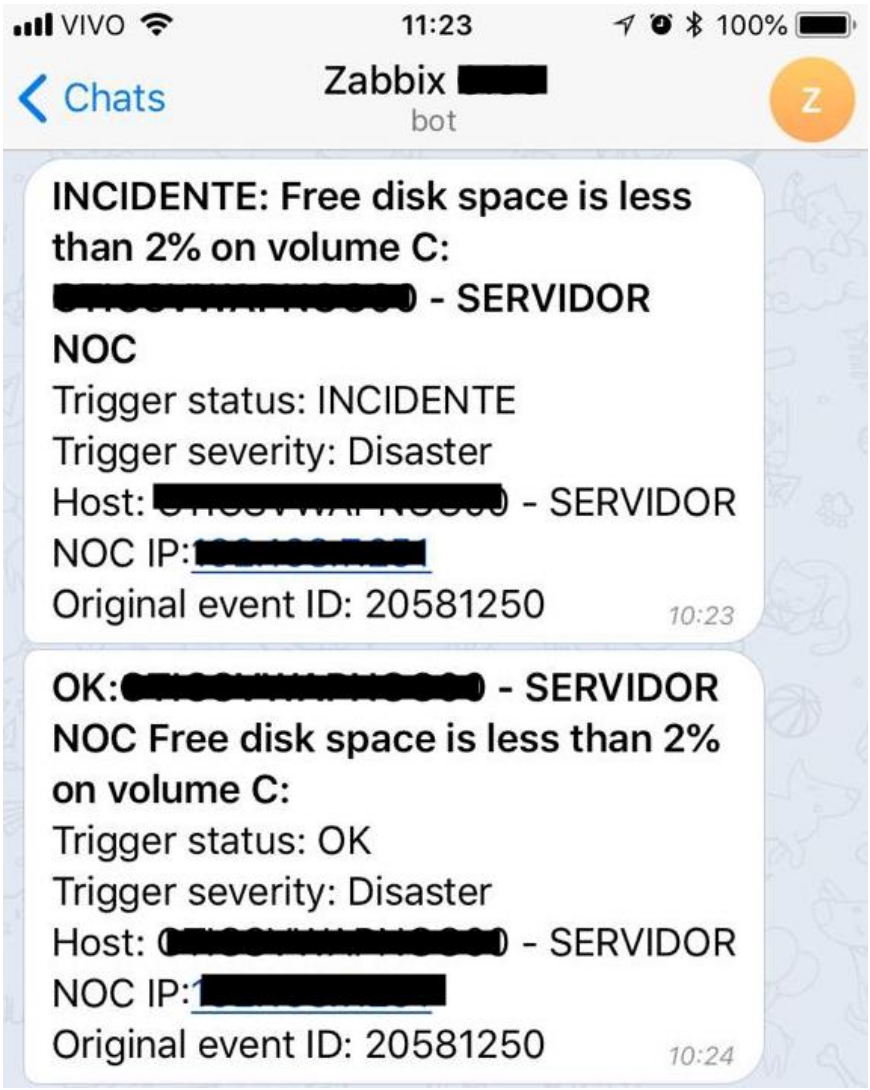


Figura 74 - Exemplo de mensagem de notificação de incidentes via Telegram

5 CONCLUSÃO

A utilização da ferramenta Zabbix para o monitoramento do ambiente completo do *Data Center* se mostrou muito útil, podendo ser considerada hoje, após implantada, completamente imprescindível para manter um ambiente tão grande e complexo. Com a utilização do Zabbix, de forma integrada ao envio de E-mails, abertura de tickets na ferramenta Freshservice e principalmente os envios de alertas em tempo real via Telegram, proporcionaram melhor visibilidade sobre os incidentes ocorridos, e foi possível assim reduzir consideravelmente problemas de indisponibilidade de servidores, esses problemas normalmente ocorriam devido a não haver um monitoramento do uso de recursos, principalmente de uso de disco por exemplo, que muitas vezes era identificado somente após os servidores ou serviços ficarem indisponíveis devido ao disco do servidor estar cheio. O Quadro 1 apresenta uma relação dos Incidentes identificados antes de ocorrer alguma indisponibilidade ao cliente, e os eventos de indisponibilidade total dos serviços devido algum incidente.

	Julho	Agosto	Setembro	Outubro	Novembro (Até o dia 10)
Eventos de Indisponibilidade	194	204	139	16	0
Incidentes Identificados antes de gerar alguma indisponibilidade	3	17	80	253	97

Quadro 1 - Eventos de indisponibilidade X Incidentes identificados

Os dados apresentados no Quadro 1 estão representados graficamente no Gráfico 1.

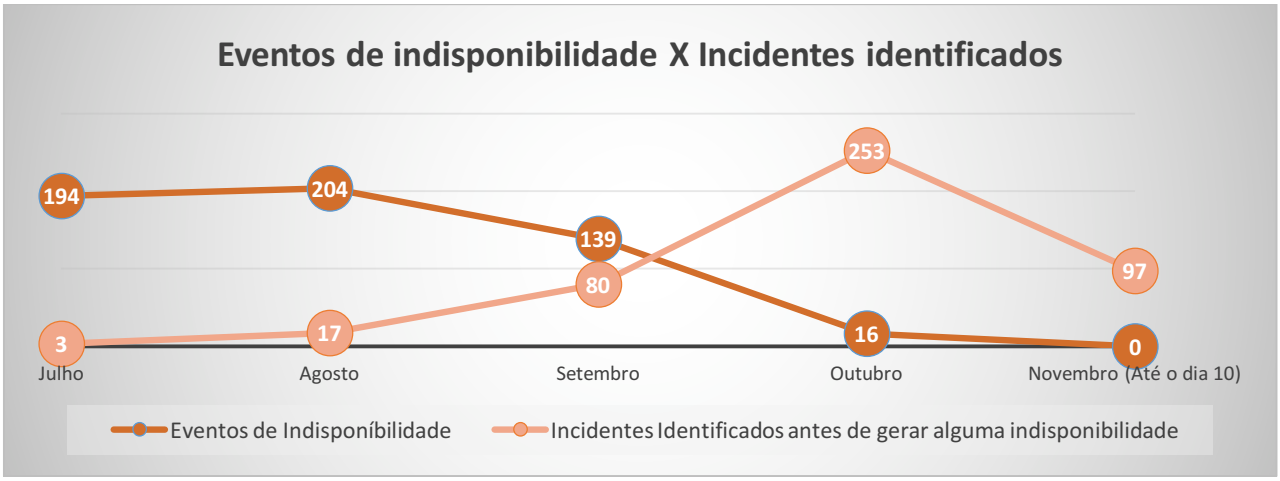


Gráfico 1 - Eventos de indisponibilidade X Incidentes identificados

Conforme demonstrado no Quadro 1 e no Gráfico 1 nos últimos meses durante a implantação do monitoramento via Zabbix, o número de eventos que causaram alguma indisponibilidade aos clientes finais foi reduzido drasticamente, conforme aumentou o número de incidentes identificados a tempo de serem resolvidos antes de causar indisponibilidades a esses clientes. Dessa houve uma grande melhora na disponibilidade do serviço prestado pelo *Data Center* atendendo assim os objetivos propostos para o presente trabalho.

O Zabbix se mostrou uma excelente ferramenta de monitoramento de redes para qualquer tipo de ambiente, pequeno, médio ou mesmo grandes e complexos ambientes. Seus maiores pontos fortes são sua robustez, capacidade de monitorar diversos tipos de ativos de redes via SNMP (versões 1, 2 e 3), também ser uma ferramenta de código aberto, com uma ampla comunidade de desenvolvimento, e por ser possível criar templates *templates* de monitoramento para qualquer coisa, tendo-se o conhecimento necessário, e principalmente suas capacidades de integrações com outras ferramentas para envios de alertas, como nesse trabalho foi utilizada a integração para envio de alertas personalizados via Telegram.

REFERÊNCIAS

AGASUS. **ATIVOS DE REDE: O QUE É E QUAIS SÃO OS MAIS IMPORTANTES PARA UMA EMPRESA?**. 2018. Agasus. Disponível em: <http://www.agasus.com.br/ativos-de-rede-o-que-e-e-quais-sao-os-mais-importantes-para-uma-empresa/>. Acesso em 18 Set. 2018.

BENINI, Renata Aparecida; DAIBERT, Marcelo Santos. **Monitoramento de Redes de Computadores - Artigo Revista Infra Magazine 1**. 2011. Devmedia. Disponível em: <https://www.devmedia.com.br/monitoramento-de-redes-de-computadores-artigo-revista-infra-magazine-1/20815>. Acesso em: 04 Ago. 2018.

CISCO. **O que é um firewall?**. 2018. Cisco. Disponível em: <https://www.cisco.com/c/pt_br/products/security/firewalls/what-is-a-firewall.html >. Acesso em 10 Set. 2018.

CONECPTI. **Gerenciamento e Monitoramento de Ativos de Redes**. 2017. ConcepTi. Disponível em: <http://concepticonsultoria.com.br/index.php/gerenciamento/gerenciamento-e-monitoramento-de-ativos-de-redes>. Acesso em: 08 Set. 2018.

GUGELMIN, Felipe. **Quais as diferenças entre hub, switch e roteador?**. 2011. Tecmundo. Disponível em: <https://www.tecmundo.com.br/roteador/9586-quais-as-diferencas-entre-hub-switch-e-roteador-.htm>. Acesso em: 10 Set. 2018**.**

LOCAWEB. **Storage - O que é e como obter**. 2017. Locaweb. Disponível em: <https://wiki.locaweb.com.br/pt-br/Storage_-_O_que_%C3%A9_e_como_obter>. Acesso em: 08 Set. 2018.

LUCENA, Felipe. **VIRTUALIZAÇÃO DE SERVIDORES: O QUE É E COMO FUNCIONA**. 2016. Diferencial TI. Disponível em: <https://blog.diferencialti.com.br/entenda-o-que-e-virtualizacao-de-servidores-e-como-funciona/>. Acesso em: 08 Set. 2018.

NAGIOS. **History of Nagios**. 2018. Nagios The Industry Standard In IT Infrastructure Monitoring. Disponível em: <https://www.nagios.org/about/history/>. Acesso em: 22 Set. 2018.

OPSERVICES. **Entenda o que é o protocolo SNMP e sua importância no monitoramento**. 2017. OpServices. Disponível em: <https://www.opservices.com.br/snmp/>. Acesso em: 13 Set. 2018.

OPENTICA. **Open Source Monitoring Tools**. 2016. Opentica. Disponível em: <http://opentica.com/en/2016/02/02/open-source-monitoring-tools/>. Acesso em: 22 Set. 2018

ROUSE, Margaret. **Zenoss**. 2018. TechTarget. Disponível em: <https://searchitoperations.techtarget.com/definition/Zenoss>. Acesso em 22 Set. 2018

TELECO. **Gerenciamento e Monitoramento de Rede II: Gerenciamento SNMP**. 2018. Teleco. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialgmredes2/pagina_2.asp>. Acesso em 20 Set. 2018.

ZABBIX. **Zabbix Documentation 1.8 - About**. 2016. Zabbix. Disponível em: <https://www.zabbix.com/documentation/1.8/manual/about>. Acesso em 22 Set. 2018.

ZABBIX. **Zabbix Documentation 3.0 - Agent**. 2018. Zabbix. Disponível em: <https://www.zabbix.com/documentation/3.0/manual/concepts/agent>. Acesso em 23 Set. 2018.

ZABBIX. **Zabbix Documentation 3.0 - Proxy**. 2018. Zabbix. Disponível em: <https://www.zabbix.com/documentation/3.0/manual/concepts/proxy>. Acesso em 23 Set. 2018.

ZABBIX. **Zabbix Documentation 3.0 - Server**. 2018. Zabbix. Disponível em: <https://www.zabbix.com/documentation/3.0/manual/concepts/server>. Acesso em 23 Set. 2018.

APÊNDICES

APENDICE A – INSTALAÇÃO DO ZABBIX AGENT EM SERVIDORES WINDOWS

Baixe o arquivo Zabbix Agent.zip disponível em: goo.gl/5b9Brf

1- Extraia os arquivos compactados e mova a pasta inteira para C:\Program Files

2- Edite o arquivo zabbix_agentd.conf que fica em: C:\Program Files\Zabbix Agent\
   - No campo **Server**, aponte o IP ou DNS do servidor Zabbix
   - No campo **ServerActive,** aponte o IP ou DNS do servidor Zabbix
   - No campo **Hostname**, informe o nome do servidor (preferencialmente o mesmo configurado no Windows).
   - Salve o arquivo e saia.

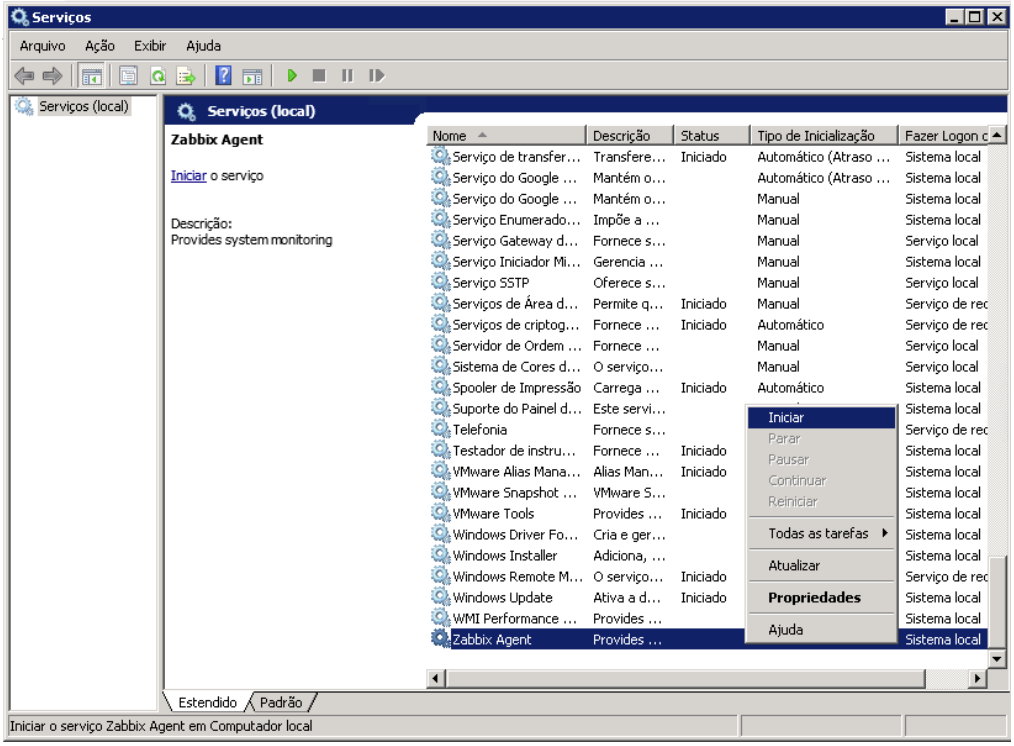
3- Abra o prompt de comandos do Windows, e acesse a pasta do Zabbix.

```
cd "\Program Files\Zabbix Agent"
```

4- Instale o Zabbix-Agent como um serviço utilizando o seguinte comando:

```
C:\Program Files\Zabbix Agent>zabbix_agentd.exe -c zabbix_agentd.conf -i
```

5- Acesse os Serviços do Windows, Localize o Serviço **Zabbix Agent** inicie o mesmo:



6- Acesse o Servidor do Zabbix: http://monitor.ciss.com.br/zabbix/

7- Clique em **Configurações > Hosts > Criar Host**

8- Preencha os campos com os dados fundamentais do servidor:
   a. Campo **Nome do host:** colocar o mesmo nome configurado no Hostname do arquivo de configuração do Zabbix Agent no servidor que está adicionando.
   b. Campo **Nome visível**: (OPCIONAL) colocar um nome amigável para o servidor (um nome mais simples para identificar a maquina.
   c. Campo **Endereço IP** dentro de **Interfaces do agente:** Coloque o endereço IP do servidor.
   d. No Campo **Grupos** > **Nos Grupos**, adicione os grupos pertinentes ao servidor.
   e. Caso seja monitorado através de um servidor proxy selecione o proxy correto no campo: **Monitorado por proxy**

9- Após preenchidos os campos acima citados, clique na aba **Templates**, e associe o *template* para hosts Windows e na sequencia clique em **Adicionar**.

APENDICE B – INSTALAÇÃO DO ZABBIX AGENT EM SERVIDORES LINUX

1 – Adicionar repositório ao servidor:
OBS: Validar versão do S.O:
Para SUSE Linux Enterprise Server 12 SP3:

```
# zypper addrepo
http://download.opensuse.org/repositories/server:/monitoring/SLE_12_SP3/server:monitoring.repo
```

Para outras versões do Linux consulte: https://www.zabbix.com/download_agents
Os passos seguintes são iguais indiferente da versão do SUSE Linux Enterprise.

2 – Instalar o Agente do Zabbix:

```
# zypper install zabbix-agent
```

3 – Editar arquivo de configuração do Zabbix:

- **Edite o arquivo /etc/zabbix/zabbix-agentd.conf**
- **Remova todas as configurações padrão do arquivo e cole o seguinte conteúdo:**

```
PidFile=/var/run/zabbix/zabbix-agentd.pid
LogFile=/var/log/zabbix/zabbix-agentd.log
Server=[IP DO SERVIDOR ZABBIX]
DebugLevel=3
EnableRemoteCommands=1
StartAgents=3
ServerActive=[IP DO SERVIDOR ZABBIX]
Hostname=[HOSTNAME]
Timeout=30
AllowRoot=1
```

  - **Altere o campo Hostname para o hostname do servidor em questão;**
  - **Altere os campos e Server e ServerActive para o IP do servidor do Zabbix.**
  - **Salve as alterações no arquivo.**

4 – Após alterado o arquivo de configuração, Inicie o serviço do Zabbix:

```
# /etc/init.d/zabbix-agentd start
```

5 – Adicionar o Zabbix a inicialização automática:

```
# echo /etc/init.d/zabbix-agentd start >> /etc/rc.d/boot.local
```

6 – Acesse o Servidor do Zabbix

7 – Clique em **Configurações > Hosts > Criar Host**

8 – Preencha os campos com os dados fundamentais do servidor:

f. Campo **Nome do host:** colocar o mesmo nome configurado no Hostname do arquivo de configuração do zabbix agent no servidor que está adicionando.
g. Campo **Nome visível**: (OPCIONAL) colocar um nome amigável para o servidor (um nome mais simples para identificar a maquina.
h. Campo **Endereço IP** dentro de **Interfaces do agente:** Coloque o endereço IP do servidor.
i. No Campo **Grupos** > **Nos Grupos**, adicione os grupos pertinentes ao servidor.
j. Caso seja monitorado através de um servidor proxy selecione o proxy correto no campo: **Monitorado por proxy**

9 – Preencha os campos com os dados fundamentais do servidor:

a. Campo **Nome do host:** colocar o mesmo nome configurado no Hostname do arquivo de configuração do Zabbix Agent no servidor que está adicionando.
b. Campo **Nome visível**: (OPCIONAL) colocar um nome amigável para o servidor (um nome mais simples para identificar a maquina.
c. Campo **Endereço IP** dentro de **Interfaces do agente:** Coloque o endereço IP do servidor.
d. No Campo **Grupos** > **Nos Grupos**, adicione os grupos pertinentes ao servidor.
e. Caso seja monitorado através de um servidor proxy selecione o proxy correto no campo: **Monitorado por proxy**

10 – Após preenchidos os campos acima citados, clique na aba **Templates**, e associe o *template* para hosts Linux e na sequencia clique em **Adicionar**.