

**LIGA DE ENSINO DO RIOGRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
CURSO DE ESPECIALIZAÇÃO EM REDES DE COMPUTADORES**

WERNECK BEZERRA COSTA

**MONITORAMENTO DE REDE: VISÃO EXTERNA COM ABORDAGEM DE NUVEM
E SOFTWARE LIVRE ZABBIX**

NATAL/RN

2013

WERNECK BEZERRA COSTA

**MONITORAMENTO DE REDE: VISÃO EXTERNA COM ABORDAGEM DE NUVEM
E SOFTWARE LIVRE ZABBIX**

Monografia apresentada como exigência do Curso de Especialização de Redes de Computadores do Centro Universitário do Rio Grande do Norte para obtenção do título de Especialista em Redes de Computadores.

Orientador: Aluizio Ferreira da Rocha Neto, Msc.

NATAL/RN

2013

WERNECK BEZERRA COSTA

**MONITORAMENTO DE REDE: VISÃO EXTERNA COM ABORDAGEM DE NUVEM
E SOFTWARE LIVRE ZABBIX**

Monografia apresentada como exigência do Curso de Especialização de Redes de Computadores do Centro Universitário do Rio Grande do Norte para obtenção do título de Especialista em Redes de Computadores.

Aprovado em: ____/____/____

BANCA EXAMINADORA

Aluízio Ferreira da Rocha Neto (Msc.)
Orientador

Membro examinador

Membro examinador

Dedico esta obra aos meus colegas de trabalho na SINFO, uma família “adotada” com quem passo mais tempo do que com minha família de nascença. Dedico também aos Colegas de Profissão, que muitas vezes não são reconhecidos pelos esforços mas que mesmo assim, continuam sendo Pragmáticos e satisfeitos em fazer bem o seu papel.

AGRADECIMENTOS

Agradeço a Deus pelas possibilidades que tem colocando à minha frente. Agradeço à minha Esposa Cícera Moura, pelo apoio na luta diária e por me dar a mão sempre que precisei. Também por me entender e incentivar nos momentos de produção deste trabalho.

Agradeço ao meu Pai Washington Luiz Costa e meu irmão Wanner Marvin Costa pela força e bons momentos.

Agradeço às pessoas que estão ao meu redor no dia a dia, os amigos, colegas de trabalho como Júlio Lima, Denyson Falcão, Nicole Rieckmann, Jefferson Igor, Bruno Augusto, Fabrício Pires, Wal' Hença Marques, Alexsandro Justino, Anderson Silva, Cícero Basílio e demais colegas não citados. Uma das frases, dita por alguém a quem não me recordo o nome, que incentivou este trabalho foi: “Aqui está tudo normal. O problema deve ser aí!”.

A estes colegas agradeço primeiramente pelas lições de caráter e bem servir, também pelas lições técnicas nas quais demonstraram sempre o desprendimento pessoal ao me ensinar. Não posso deixar de agradecer-los também, pelos momentos de descontração que tanto aliviaram a tensão do expediente.

Agradeço aos colegas Bruno Anacleto e Edmilson Júnior pelo interesse em meu trabalho e em utiliza-lo no seu dia a dia.

RESUMO

O conhecimento do estado dos serviços oferecidos via Internet é de total valia para as empresas que os oferecem. Caso os serviços não funcionem, os provedores precisam de *Feedbacks* caso estes não sejam dados por um de seus clientes. Para se ter uma real sensação da situação dos serviços, foi idealizado e configurado um monitoramento com o software Zabbix, que levou em conta a presença de um Processo Proxy localizado externamente em uma das redes da *Amazon AWS*. Este Proxy foi utilizado pelo Servidor Zabbix como ponte que testou e coletou dados relacionados a alguns servidores de Portais WEB e serviços DNS no domínio de um provedor de serviço fictício, porém com demandas comuns a provedores Reais. Com esta prova de conceitos, foi possível verificar que o monitoramento com o Proxy Zabbix utilizando a Computação em Nuvem, complementa o tradicional pois foi possível acompanhar o estado dos Servidores monitorados de forma automatizada e não intrusiva, complementando o monitoramento executado internamente.

Palavras-chave: Monitoramento. Zabbix. Proxy. Computação em Nuvem.

ABSTRACT

The knowledge of the internet services state have full value for companies that offer them. If these services do not work, these Providers need feedbacks that are not given by one of his clients. To get a real feel of the situation of services has been designed and configured monitoring with Zabbix software, which took into account the presence of a Proxy Process located externally in a network of Amazon AWS. This Proxy has been used by the Server as a bridge which tested and collected data related to some DNS services servers and web portals services in the field of a fictional service provider, but with the ordinary providers demands. With this proof of concept, we found that monitoring with Zabbix Proxy using Cloud Computing, complements traditional therefore was possible to monitor the status of servers in an automated way unobtrusive, complementing the monitoring performed internally.

Key-words: Monitoring, Zabbix. Proxy. Cloud Computing.

LISTA DE FIGURAS

Figura 1: Exemplo de virtualização.	16
Figura 2: Site da Amazon AWS.....	18
Figura 3: Serviços oferecidos pela AWS.	19
Figura 4: Console de Gerenciamento AWS.....	22
Figura 5: Recursos disponíveis no Nível AWS gratuito.	24
Figura 6: Acesso ao portal de cadastro AWS (botão em evidência).	25
Figura 7: Um exemplo de árvore OID.....	29
Figura 8: Exemplo de interação entre os componentes do Modelo SNMP.	30
Figura 9: Alguns exemplos de dispositivos monitorados através do SNMP.....	31
Figura 10: Logomarca Zabbix.....	35
Figura 11: Componentes Zabbix e métodos de obtenção de dados.	38
Figura 12: Exemplos de gráficos gerados pelo Zabbix.....	39
Figura 13: Exemplo de gráfico Customizado.....	40
Figura 14: Exemplo de Network Map (Mapa de Rede).	40
Figura 15: Estrutura interna, de borda e externa.....	44
Figura 16: "Caminho" feito pelo Cliente versus Equipe de Suporte.....	46
Figura 17: Forma mais convencional para o Proxy Zabbix.	49
Figura 18: Monitorando de forma alternativa com Zabbix Proxy.	50
Figura 19: Adicionando um Proxy para requisições remotas.	52
Figura 20: Cadastro de informações do Proxy.	52
Figura 21: Status do Proxy depois de Adicionado.....	52
Figura 22: Script para monitoramento via consultas DNS.	54
Figura 23: Grupos criados para os servidores.....	54
Figura 24 - Template "Applications Only".....	55
Figura 25: Templates para servidores Monitorados	56
Figura 26: Configuração de Host.....	56
Figura 27: Primeiro item cadastrado para Template DNS.....	57
Figura 28: Segundo item cadastrado para Template DNS.....	58
Figura 29: Terceiro item cadastrado para Template DNS	59
Figura 30: Itens cadastrados para Template WEB.....	60
Figura 31: Itens HTTP/HTTPS para o Template WEB	60

Figura 32: Escolha do servidor para o <i>Web Scenario</i>	61
Figura 33: Criação de Web Scenario – Aba Scenario	61
Figura 34: Criação de Web Scenarios – Aba Steps	62
Figura 35: Dados adicionais para WebMail	62
Figura 36: Dados obtidos pelos itens monitorados (WEB).	63
Figura 37: Dados obtidos pelos itens monitorados (DNS).....	64
Figura 38: Resumo da verificação via Web Scenario – Servidor WEB 01	65
Figura 39: Velocidade de Download - Web Scenario	65
Figura 40: Tempo para Download - Web Scenario	66

LISTA DE TABELAS

Tabela 1: Tipos de instâncias EC2 (de uso Geral)	20
Tabela 2: Configuração da instância oferecida no Nível Gratuito.....	23
Tabela 3: Versões, datas, RFCs e observações do protocolo SNMP.	32
Tabela 4: Itens a serem monitorados nos servidores da EnterpriseWEB.	48

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	OBJETIVOS.....	14
1.1.1	<i>Objetivos Específicos.....</i>	<i>14</i>
1.2	JUSTIFICATIVA	14
2	REFERENCIAL TEÓRICO.....	15
2.1	DATACENTER VIRTUALIZAÇÃO E CLOUD COMPUTING	15
2.1.1	<i>Amazon WEB Services (AWS).....</i>	<i>17</i>
2.1.1.1	Serviços	19
2.1.1.1.1	EC2	19
2.1.1.1.2	RDS.....	20
2.1.1.1.3	EBS	21
2.1.1.2	Console de gerenciamento	21
2.1.1.3	Nível gratuito de serviços.....	22
2.1.1.4	Precificação	25
2.2	MONITORAMENTO DE REDES	25
2.2.1	<i>Modelo SNMP.....</i>	<i>26</i>
2.2.1.1	Gerentes SNMP	28
2.2.1.2	Agentes SNMP.....	28
2.2.1.3	MIB.....	28
2.2.1.4	Operações SNMP.....	31
2.2.1.5	Versões.....	32
2.2.2	<i>Baseados em Agentes Próprios.....</i>	<i>32</i>
2.2.3	<i>Outros (comandos auxiliares de sistema).....</i>	<i>33</i>
2.2.4	<i>Monitoramento na Prática</i>	<i>33</i>
2.3	ZABBIX.....	34
2.3.1	<i>História</i>	<i>35</i>
2.3.2	<i>Licenciamento.....</i>	<i>35</i>
2.3.3	<i>Arquitetura</i>	<i>36</i>
2.3.4	<i>Suporte visual</i>	<i>38</i>
2.3.4.1	Elementos simples	39
2.3.4.2	Elementos compostos	41
3	DESENVOLVIMENTO	42
3.1	CONTEXTUALIZAÇÃO DO AMBIENTE	42
3.1.1	<i>Estrutura Interna</i>	<i>42</i>
3.1.2	<i>Estrutura de Borda.....</i>	<i>43</i>

3.1.3	<i>Estrutura Externa</i>	43
3.1.4	<i>Dependência Entre as Estruturas</i>	44
3.2	PERCEPÇÃO DOS SERVIÇOS PELOS CLIENTES.....	45
3.3	MONITORAMENTO UTILIZANDO A NUVEM	46
3.3.1	<i>Provedor a Ser Monitorado</i>	48
3.4	MONITORAMENTO PROPOSTO.....	49
3.4.1	<i>Instalação e Configurações</i>	51
3.4.2	<i>Execução</i>	52
3.4.2.1	<i>Criação dos Host Groups</i>	54
3.4.2.2	<i>Criação dos Templates</i>	54
3.4.2.3	<i>Cadastro dos Hosts</i>	56
3.4.2.4	<i>Itens monitorados</i>	57
3.4.3	<i>Testes e itens visuais</i>	63
4	CONCLUSÃO.....	67
5	CONSIDERAÇÕES FINAIS.....	68
	REFERENCIAL BIBLIOGRÁFICO	69
	ANEXO 01: INSTALAÇÃO SIMPLIFICADA DO ZABBIX SERVER	73
	ANEXO 02: INSTALAÇÃO DO ZABBIX PROXY	76
	ANEXO 03: SCRIPT “PORT_CHECK.SH”	79

1 INTRODUÇÃO

Para maioria das pessoas, a internet é vista como uma grande rede sem começo ou fim. Na verdade se trata de uma grande variedade de dispositivos conectados, formando pequenas aglomerados, que juntos crescem em complexidade, formando múltiplas redes. Hoje, a definição de rede está muito além do famoso “terminais processadores conectados...”.

A rede mundial vive momentos de franca expansão, onde todos os tipos de dispositivos podem “conversar”. A escassez dos endereços IPv4 lançou a necessidade de adoção do IPv6, que fez com que conceitos fossem criados (ou lembrados) como a internet das coisas e internet industrial, por exemplo. Outro ponto forte é a rápida adoção das pessoas às tecnologias pessoais, como *SmartWatches*, *Tablets* e *SmartPhones*.

Os dispositivos e as pessoas demandam serviços. Neste contexto se verifica que a tarefa de monitorar/gerenciar estes serviços se torna complexa e impossível ser feita de forma manual. Este monitoramento exige visão antecipada das diversas situações, antes mesmo que o cliente ou usuário final venha a perceber que o serviço demandado não esteja disponível.

A indisponibilidade gerada pela parada de um destes serviços é um grande risco para as organizações, pois eles podem ser sua presença digital, levando o usuário a ter uma má impressão sobre a empresa, mediante o descontentamento com algo que, muitas vezes, foge do controle destas organizações.

As técnicas de gerência de redes estão modernas. Muitas vezes incluem abordagens interativas onde ao perceber que existe certo problema, o sistema de monitoramento avisa aos administradores de rede por meio de comunicação eletrônica sobre o ocorrido. Esta comunicação por vezes se dá através de Mensagens de Celular, Mensagens via Chat Instantâneo ou mesmo utilizando VoIP para chamadas telefônicas com mensagens pré-gravadas, porém geralmente estas verificações são executadas no âmbito interno.

Para que a solução fique completa, falta um método para obter a mesma sensação que o cliente tem quando acessa os serviços oferecidos por um provedor. Este trabalho se propõe a completar este “arsenal” provendo uma maneira para que,

utilizando a Computação em Nuvem, seja possível ter uma visão externa das condições dos serviços prestados aos usuários externos de um provedor de serviços de internet, se estendendo a qualquer empresa que deseje fazer o mesmo.

1.1 OBJETIVOS

O objetivo deste trabalho é apresentar uma solução que complemente a abordagem de monitoramento de serviços de rede feita atualmente, objetivando alcançar a visão do cliente externo com relação ao funcionamento dos serviços oferecidos pelo provedor para que, em momentos de problemas, seja possível antecipar o conhecimento e correção tendo o foco em minimizar a perda de credibilidade com o cliente final.

1.1.1 Objetivos Específicos

Para atingir o objetivo geral deste trabalho, as seguintes etapas foram seguidas:

- Estudar o atual estado do monitoramento de Redes
- Estudar o modelo de computação em nuvem
- Implementar uma solução que complementasse o atual cenário de monitoramento

1.2 JUSTIFICATIVA

Atualmente todas as organizações, com ou sem fins lucrativos, dependem das Tecnologias de Informação e Comunicação em seus processos do dia a dia. Seja a Tecnologia somente um meio para executá-las ou até mesmo o pilar de sustentação desta, é bastante importante que se tenha uma visão do estado atual dos serviços que dependam de meios tecnológicos para existir ou se manter.

Em vários casos, a imagem destas organizações pode estar ligada à sua presença digital, através dos serviços providos pela mesma a certo público.

Neste contexto, não é fácil dispor de uma forma de monitoramento destes serviços, de forma a se integrar com tecnologias de monitoramento existentes e quando existem, ficam caras, pois funcionam no modelo de comércio baseado em módulos, onde cada novo módulo tem um custo adicional, diversas vezes cobrado por item a ser monitorado.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os principais conceitos e tecnologias que são a base para o entendimento da solução proposta no Capítulo 3.

2.1 DATACENTER VIRTUALIZAÇÃO E CLOUD COMPUTING

Segundo (Arregoces & Portolani, 2004), os *Data Centers* hospedam recursos computacionais críticos em um ambiente controlado e sob gerenciamento centralizado, o que permite às empresas operarem todo o tempo ou de acordo com o que seus negócios demandam. Estes recursos computacionais incluem servidores centrais, Servidores de aplicações, rede otimizada e armazenamento de dados.

Um *Data Center* atual pode ser formado por uma infraestrutura centralizada que visa, dentre outras coisas, atender aos seguintes requisitos:

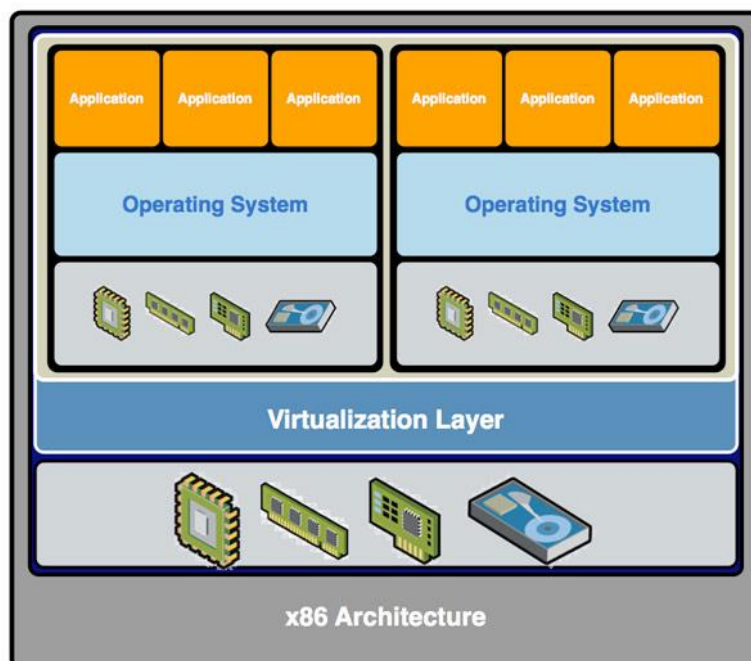
- Centralização de poder computacional;
- Centralização de Armazenamento de dados;
- Diminuição do custo de manutenção;
- Suporte às aplicações do negócio no tempo do negócio (tempo de atividade);
- Resiliência e/ou redundância para Hardware e Software.

Os *Data Centers* dependem e levam consigo grandes estruturas de Rede para a comunicação entre os nós internos e dos nós internos com o “mundo”. Estas estruturas são aproveitadas ao máximo, visto que o tráfego colocado tende a ser o mais próximo possível de sua capacidade total.

Segundo (Josyula, Orr, & Page, 2012) Virtualização é a criação de uma versão virtual de alguma coisa, como um Sistema Operacional, Servidor, dispositivo de armazenamento ou dispositivo de rede. Virtualização de servidor altera as regras quebrando o modelo tradicional, onde um Servidor Físico hospeda um único Sistema operacional, para um modelo onde são criadas várias máquinas virtuais no servidor físico, através da utilização de tecnologias de *Hyper-visors* (Hipervisores).

Um esquema de virtualização pode ser verificado na Figura 1.

Figura 1: Exemplo de virtualização.



Fonte: http://www.gruppen.com.br/virtualizacao/img/virtualizacao_alocacaoderecursos.jpg

Virtualização se tornou uma necessidade técnica nos dias atuais, e a tendência é a continuidade desta realidade, pois, quando bem implementada, possibilita muito benefícios, destacando-se os seguintes:

- Acesso sobdemanda a Servidores, Rede e Armazenamento de dados;
- Economia de energia;
- Redução da ocupação física;
- Redução de do investimento e custos operacionais.

A junção de Data Centers com Virtualização, fez surgir um novo modelo de operação chamado *Cloud Computing* (computação em nuvem).

Segundo o (NIST, 2012), Cloud computing é um modelo que permite acesso conveniente via rede, a um conjunto compartilhado de recursos de computação (rede, servidores, armazenamento, aplicações e serviços) que podem ser provisionados e lançados de forma ágil, com o mínimo de esforço ou interação com o provedor de serviços.

Em resumo a (NIST, 2012), este modelo em nuvem promove disponibilidade e é composto por cinco características essenciais:

- Serviços em modo "Self-service" e sobdemanda;
- Acesso a Redes de Banda Larga;

- Separação de recursos;
- Rápida Elasticidade;
- Serviços mensuráveis.

Três modelos de serviço:

- Cloud Software como um Serviço (SaaS);
- Cloud Platform como um Serviço (PaaS);
- Cloud Infrastructure como um Serviço (IaaS).

E quatro modelos de Implantação:

- Nuvem Privada;
- Nuvem comunitária;
- Nuvem pública e;
- Nuvem Híbrida.

Tecnologias-chave para o funcionamento de Cloud, incluem: Redes rápidas mundiais, servidores poderosos e baratos e virtualização de alta-performance para hardware padrão de mercado.

O modelo de computação em nuvem promove a diminuição massiva dos custos combinada com o crescimento da agilidade na TI. É considerado crítico ao governo e iniciativas privadas, começar a adotar estas tecnologias em resposta às dificuldades financeiras. No entanto, a tecnologia de computação em Nuvem desafia as abordagens tradicionais dos Data Centers e desenho/manutenção das aplicações empresariais. Apesar da Computação em Nuvem estar se tornando popular, segurança, interoperabilidade e portabilidade, continuam sendo citadas como maiores barreiras para adoção em massa.

2.1.1 Amazon WEB Services (AWS)

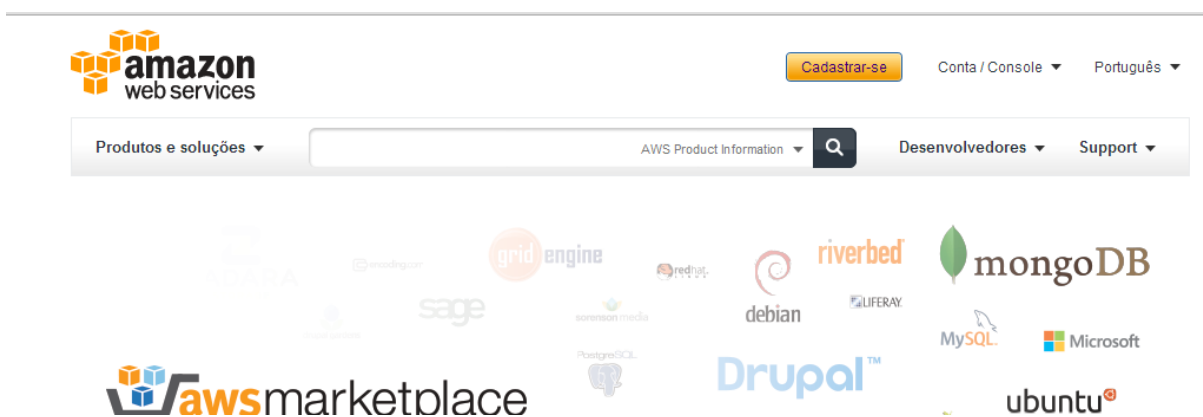
Com base em (Amazon AWS), Amazon AWS é um conjunto de serviços que provê uma plataforma de Computação em Nuvem com flexibilidade, boa relação custo-benefício, escalabilidade e facilidade de uso, que se adéqua a qualquer área (pesquisa, uso educacional, individual ou empresarial pra empresas e qualquer tamanho).

De acordo com (Varia & Mathew, 2013), a Amazon (como maior empresa de venda e distribuição para o Varejo), havia investido uma década e alguns milhões de

dólares pra construir e gerenciar uma estrutura de T.I confiável, eficiente e de larga-escala para apoiar as próprias operações. Em 2005, estava sendo formulado o serviço AWS, com a intenção de que outras empresas pudessem tirar vantagens deste know-how e estrutura. Com isso descobriu um novo mercado. O serviço foi realmente lançado em 2006 e vem ganhando muitos adeptos ao longo deste tempo.

O endereço para acesso à *homepage* é <http://aws.amazon.com>. A Figura 2 mostra o *Site* da empresa.

Figura 2: Site da Amazon AWS.



É possível verificar os serviços oferecidos pela AWS, acessando ao endereço <http://aws.amazon.com/products/>.

Todos os serviços oferecidos pela AWS estão disponíveis e podem ser facilmente acessados pela internet, utilizando um portal de Gerenciamento.

A AWS visa atender as demandas dos clientes, servindo seus recursos de forma “Elástica”. Segundo (Veras, p. 13), Elasticidade é o poder para dimensionar recursos computacionais, diminuindo ou expandindo-os, facilmente e com o mínimo de Atrito.

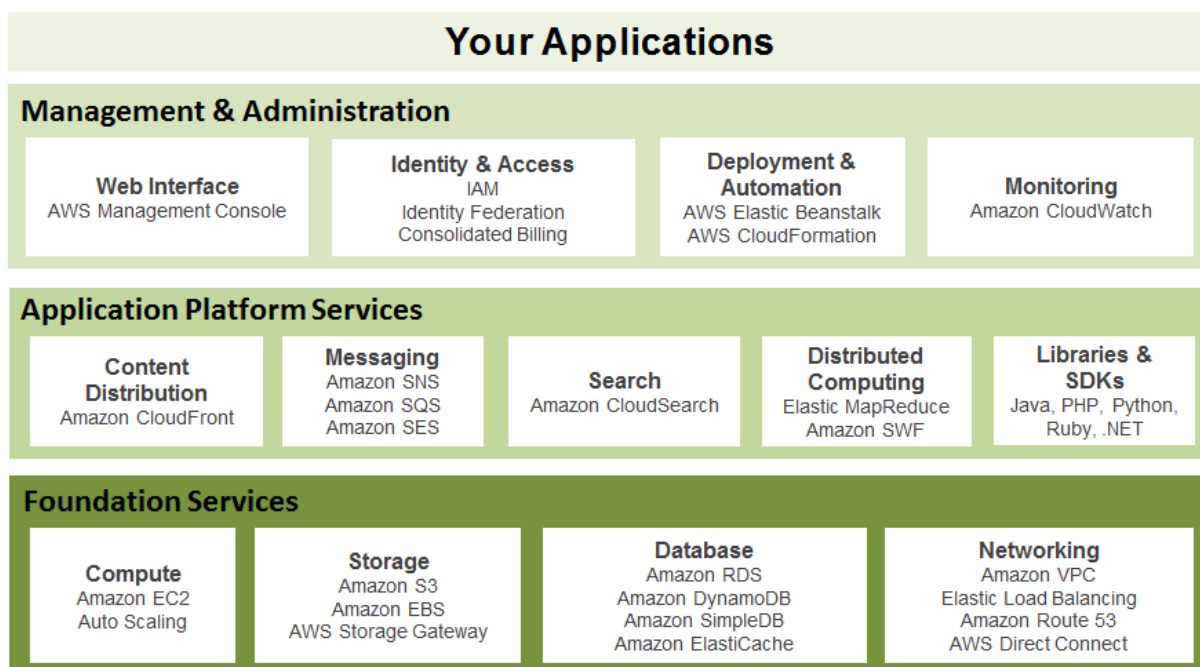
A elasticidade pode ser vista como “computação sobdemanda”, onde o cliente pode, como certa facilidade, alterar os recursos de sua Nuvem sem que o provedor de serviços interfira diretamente nesta tarefa.

A AWS oferece uma gama de serviços, visando ser genérica ao ponto de atender qualquer demanda Tecnológica de qualquer cliente. Isto porque toda a estrutura de TI converge para o lançamento e aplicações, que serviram organizações e pessoas.

2.1.1.1 Serviços

Na Figura 3, é possível verificar os serviços oferecidos pela AWS (em Junho/2013).

Figura 3: Serviços oferecidos pela AWS.



Fonte: Adaptada de (Amazon AWS, 2013).

Apesar da gama de serviços e suas diferentes aplicações, cabe aqui ressaltar alguns que foram utilizados ao longo deste trabalho:

- **EC2**: Elastic Compute Cloud;
- **RDS**: Relational Database Service;
- **EBS**: Elastic Block Storage.

2.1.1.1.1 EC2

Segundo (Amazon AWS, 2013), O Amazon Elastic Compute Cloud (Amazon EC2) é um serviço da Web que fornece uma capacidade de computação redimensionável na nuvem.

De forma complementar, segundo (Veras, p. 119) (...) ele é um Web Service adquirido e pago sob demanda. (...) O servidor convencional é substituído por uma instância virtual. (...) Permite às empresas clientes e aos

desenvolvedores utilizar recursos de processamento, de acordo do aplicativo e pagar pelo uso.

O serviço EC2 é contratado por instâncias. Cada instância representa os recursos de um servidor, como processador e memória.

Segundo (Veras, p. 119), O EC2 reduz para minutos o tempo necessário para lançar ou reiniciar novas instâncias, permitindo dimensionar a capacidade, pra cima ou para baixo, à medida que necessidades computacionais mudam.

As instâncias em si são algo como um “esqueleto”, pois não possuem sistema operacional. Para que seja dada “vida” a um servidor, é preciso unir o par instância + Imagem (AMI – *Amazon Machine Image*). A AMI é uma imagem pré configurada de sistema operacional e que pode ser disponibilizada pela AWS ou por terceiros.

Os tipos de instâncias (de uso Geral) podem ser verificadas na Tabela 1.

Tabela 1: Tipos de instâncias EC2 (de uso Geral)

Família de instâncias	Tipo de instância	Arquitetura de processador	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GB)	Otimizada para EBS disponível	Desempenho de rede
Uso geral	m1.small	32 bits ou 64 bits	1	1	1,7	1 x 160	-	Baixo
Uso geral	m1.medium	32 bits ou 64 bits	1	2	3,75	1 x 410	-	Moderado
Uso geral	m1.large	64 bits	2	4	7,5	2 x 420	Sim	Moderado
Uso geral	m1.xlarge	64 bits	4	8	15	4 x 420	Sim	Alto
Uso geral	m3.xlarge	64 bits	4	13	15	Somente EBS	Sim	Moderado
Uso geral	m3.2xlarge	64 bits	8	26	30	Somente EBS	Sim	Alto

Fonte: Adaptada de (Amazon AWS - Instance Types)

2.1.1.1.2 RDS

Segundo (Amazon AWS) O Amazon Relational Database Service (Amazon RDS) é um serviço da Web que facilita a configuração, a operação e o escalonamento de um banco de dados relacional na nuvem. Ele fornece uma capacidade econômica e reescalável enquanto gerencia tarefas de administração do banco de dados que consomem tempo (...). O RDS fornece (...) acesso às capacidades de um mecanismo de banco de dados familiar como MySQL, Oracle ou Microsoft SQL Server. Isso significa que o código, os aplicativos e as ferramentas utilizados hoje com os bancos de dados existentes podem ser usados com o Amazon RDS.

A utilização do RDS pode ser feita tanto por instâncias EC2 quanto por máquinas que estiverem fora da AWS, bastando configurar a área de segurança da instância RDS.

2.1.1.1.3 EBS

Dependendo do tipo da instância EC2, pode ser necessária a adição de novos dispositivos de armazenamento. Estes dispositivos são provisionados através dos EBS.

Segundo (Amazon AWS - EBS) O Amazon Elastic Block Store (EBS) fornece volumes de armazenamento em bloco para uso com instâncias do Amazon EC2. Os volumes do Amazon EBS são conectados à rede e persistem independentemente da vida útil de uma instância. O Amazon EBS fornece volumes de armazenamento altamente disponíveis, confiáveis e previsíveis que podem ser conectados a uma instância do Amazon EC2 em execução e expostos como um dispositivo dentro da instância. O Amazon EBS é particularmente adequado para aplicativos que exigem um banco de dados, um sistema de arquivos ou acesso a armazenamento de blocos brutos.

Dentre as características do serviço EBS, vale ressaltar as seguintes:

- Os volumes podem ter de 1 *GigaByte* a 1 *TeraByte*;
- O desempenho (em relação à Leitura/Escrita) é configurável através da configuração de IOPS (In/Out Per Seconds – Entradas/Saídas por segundo);
- Para a instância EC2 o EBS aparece como um disco conectado diretamente e visível ao sistema operacional.

2.1.1.2 Console de gerenciamento

Para administrar os recursos disponíveis na AWS, existe uma plataforma que centraliza todos os serviços e permite gerência total às soluções AWS. Esta demanda, segundo o (NIST, p. 2), faz parte das características essenciais à Computação em Nuvem:

Autoatendimento sobdemanda. Um cliente pode, de forma unilateral, provisionar capacidades computacionais, como tempo de processamento e

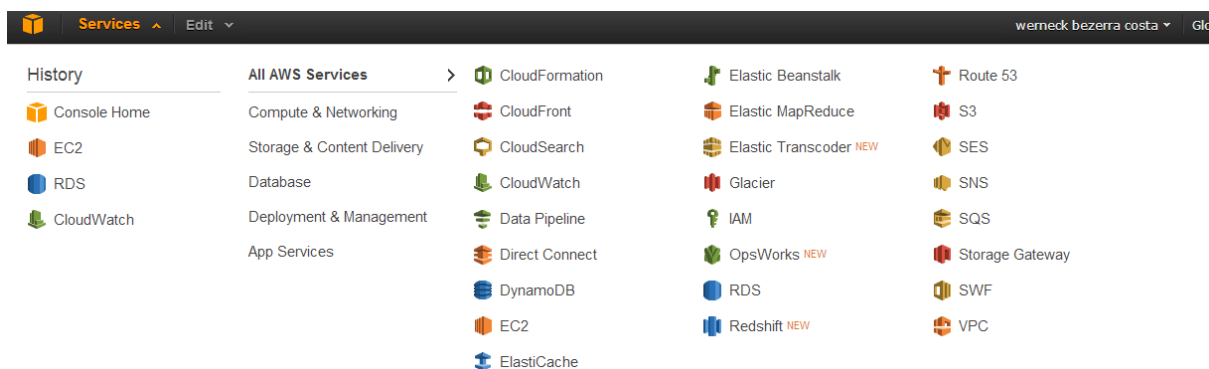
armazenamento em rede, quando necessário, de forma automatizada, sem que se faça necessária a interação *humano-humano*, com o provedor de serviços.

A AWS oferece este tipo de gerência através de seu *Console de Gerenciamento (Management Console)*. Segundo (Amazon AWS - Management Console), O AWS Management Console oferece uma interface web simples para a Amazon Web Services.

Para entrar no sistema de gerenciamento unificado é preciso possuir uma conta na AWS. Esta conta pode ser no modo Gratuito.

Através da Figura 4, é possível verificar a interface do Console de Gerenciamento AWS.

Figura 4: Console de Gerenciamento AWS.



Fonte: Acesso ao Console de Gerenciamento

2.1.1.3 Nível gratuito de serviços

A AWS fornece um nível gratuito de serviços para que o cliente interessado possa ter um primeiro contato com a plataforma e valide as suas funções básicas.

Segundo (Amazon AWS - Free) para ajudar os novos clientes da AWS a dar os primeiros passos na nuvem, a AWS está apresentando um nível de uso gratuito. O nível gratuito pode ser usado para tudo o que você quiser executar na nuvem: iniciar novos aplicativos, testar aplicativos existentes na nuvem ou simplesmente obter experiência prática com a AWS.

A configuração da instância EC2 para o nível gratuito pode ser visto na Tabela 2.

Tabela 2: Configuração da instância oferecida no Nível Gratuito.

Família de instâncias	Tipo de instância	Arquitetura de processador	vCPU	ECU	Memória (GiB)	Armazenamento da instância (GB)	Otimizada para EBS disponível	Desempenho de rede
Microinstâncias	t1.micro	32 bits ou 64 bits	1	Variável *6	0,615	Somente EBS	-	Muito baixo

Fonte: Adaptada de (Amazon AWS - Instance Types)

Segundo (Veras, p. 108), o uso gratuito da AWS é disponibilizado por um ano após a conta AWS ser aberta. Além da instância de servidor virtual (EC2), o nível gratuito oferece uma pequena parte de todos os serviços disponíveis no “cardápio” AWS, porém cada um com suas limitações. A Figura 5 mostra os limites dos serviços, para o nível gratuito (os mais utilizados).

Figura 5: Recursos disponíveis no Nível AWS gratuito.

Nível de uso gratuito da AWS (por mês):	
Elastic Compute Cloud (EC2) 750 horas do Amazon EC2 para Linux† Uso de Microinstância (613 MB de memória e suporte à plataforma de 32 e 64 bits) – horas suficientes para executar continuamente todos os meses* 750 horas do Amazon EC2 para Microsoft Windows Server‡ Uso de Microinstância (613 MB de memória e suporte à plataforma de 32 e 64 bits) – horas suficientes para executar continuamente todos os meses* 750 horas de Elastic Load Balancer além de 15 GB de processamento de dados* 30 GB de Amazon Elastic Block Storage, além de 2 milhões de E/S e 1 GB de armazenamento de snapshot*	Simple Workflow (SWF) 1.000 execuções de fluxo de trabalho do Amazon SWF podem ser iniciadas gratuitamente. Um total de 10.000 tarefas de atividade, sinais, temporizadores e marcadores e 30.000 dias de trabalho efetivo também podem ser utilizados gratuitamente**
Simple Storage Service (S3) 5 GB de armazenamento padrão do Amazon S3, 20.000 solicitações GET e 2.000 solicitações PUT*	Simple Queue Service (SQS) e Simple Notification Service (SNS) 1.000.000 de solicitações de Amazon Simple Queue Service** 1.000.000 de solicitações, 100.000 notificações de HTTP e mil notificações de e-mail para Amazon Simple Notification Service**
DynamoDB Armazenamento de 100 MB, cinco unidades de capacidade de gravação e dez unidades de capacidade de leitura para o Amazon DynamoDB**	Amazon Elastic Transcoder 20 minutos de transcodificação SD ou 10 minutos de transcodificação HD**
Relational Database Service (RDS) 750 horas de Microinstâncias de banco de dados Single-AZ do Amazon RDS para execução de MySQL, Oracle BYOL ou SQL Server (executando o SQL Server Express Edition) – horas suficientes para executar uma Instância de banco de dados continuamente todos os meses* 20 GB de armazenamento de banco de dados 10 milhões de E/S 20 GB de armazenamento de backup para seus backups de banco de dados automatizados e quaisquer snapshots de banco de dados iniciados por usuário	CloudWatch 10 métricas do Amazon Cloudwatch, 10 alarmes e 1.000.000 solicitações de API**
	Transferência de dados 15 GB de largura de banda para fora agregado em todos os serviços AWS*
	Data Pipeline 3 pré-condições de baixa frequência executando na AWS por mês* 5 pré-condições de baixa frequência executando na AWS por mês*
	ElastiCache 750 horas do Amazon ElastiCache – horas suficientes para executar um nó de cache continuamente todos os meses.*

Fonte: (Amazon AWS - Free).

A lista completa (separada por área de atuação) pode ser verificada em (Amazon AWS - Free).

Sobre a validade da oferta e cobranças, segundo (Amazon AWS - Free Tier FAQs), funciona da seguinte forma:

O nível de uso gratuito da AWS expirará em 12 meses, a partir da data em que você se cadastrou. Quando o uso gratuito expirar ou se o uso do

aplicativo ultrapassar os níveis de uso gratuito, você simplesmente pagará taxas de serviço padrão e de acordo com o uso (...).

Para dar início à utilização do nível gratuito de serviços AWS, é preciso acessar o *site* <http://aws.amazon.com/pt/free> e criar uma conta. Será necessário possuir um cartão de crédito com presença internacional.

A Figura 6 mostra o primeiro passo do cadastro: o acesso ao portal.

Figura 6: Acesso ao portal de cadastro AWS (botão em evidência).



Fonte: (Amazon AWS - Free).

2.1.1.4 Precificação

Como visto, a AWS oferece este nível gratuito de uso com limitações, estas limitação podem ser de tempo de uso (doze meses) ou de uso de recursos. Caso a primeira sentença seja atingida e se deseje continuar a utilizar os serviços, é possível calcular, de forma antecipada, o valor para uma instância compatível com a versão *Free* oferecida inicialmente.

Segundo (Amazon AWS - Pricing Overview, p. 10), é possível utilizar a Calculadora mensal simples AWS para estimar o custo de uso de todos os serviços AWS. É possível acessar a calculadora através do endereço <http://calculator.s3.amazonaws.com/calc5.html>.

2.2 MONITORAMENTO DE REDES

Monitorar significa, em linhas gerais, verificar, acompanhar, reunir dados sobre estados e situações, sejam elas normais ou anômalas. Na área de tecnologia da informação, à medida que seus serviços criam uma dependência maior no dia-a-

dia das pessoas, a importância do monitoramento de dispositivos e sistemas cresce, na tentativa de garantir que os equipamentos e sistemas estejam funcionando corretamente pelo máximo tempo possível.

Em redes de computadores o monitoramento vai desde o acompanhamento do estado de seus Ativos (Switchs, Roteadores, Firewalls, Equipamentos Wireless, dentre outros), à reunião de dados sobre utilização de recursos, visando, por exemplo, o gerenciamento/planejamento de capacidade e melhoria.

Em resumo à citação de (Nicolletti, Lopes, & Sauvé, p. 17), o objetivo da Gerência (monitoramento) de Redes é monitorar e controlar os elementos da rede (sejam eles físicos ou lógicos), assegurando certo nível de qualidade de serviço.

O monitoramento não possui uma forma exclusiva de ser executado. Esta realidade se dá pelo fato de que existem diferentes abordagens de monitoramento, umas mais práticas e simples funcionando em praticamente todos os sistemas ou equipamentos com baixos custos, outras mais complexas exigindo alto poder de processamento e recursos avançados com custos consideráveis. O que existe atualmente são melhores práticas e indicações, baseadas em RFCs, que norteiam os *Administradores de Redes* no caminho do monitoramento.

Do ponto de vista operacional, o monitoramento possui diferentes abordagens relacionadas a níveis de necessidade e complexidade. Este trabalho não pretende ser uma referência sobre modelos de gerenciamento, por tanto, apesar da existência de outros modelos, serão abordados somente os que seguem.

2.2.1 Modelo SNMP

Os protocolos de monitoramento e gerenciamento são parte da estratégia de manutenção dos ambientes físico e lógico das redes para a fluência e continuidade das aplicações que dependam destes ambientes.

Com base em (Comer, p. 342), é possível verificar que os protocolos de gerenciamento usados atualmente, nasceram quando as redes eram homogêneas, tendo como ponto forte a capacidade de serem gerenciados dependendo somente da camada de enlace, ou seja, preveniam-se contra possíveis falhas nas camadas superiores. Nos dias atuais, a heterogeneidade é verificada em todos os níveis que envolvem as redes: camada física, enlace, aplicações e equipamentos. Tendo em

vista esta realidade, é preciso a adoção de protocolos simples e genéricos, com interoperabilidade entre equipamentos diferentes, que afetem minimamente o tráfego das redes e com baixa demanda por processamento nos ativos ou hosts que o apliquem.

Em resumo a (RFC1157, p. 5) O SNMP (Simple Network Management Protocol) é definido pelo IETF como sendo o protocolo padrão para gerenciamento das redes de internet (Internet Standard Management Framework) por explicitar a simplicidade de gerenciamento. Segundo (Mauro & Schmidt, 2005), o SNMP foi proposto em 1988 devido à crescente demanda por padronização do gerenciamento em dispositivos IP. Possibilita a seus usuários, com um conjunto simples de operações, monitorarem remotamente estes dispositivos.

Atuando na camada de Aplicação, o SNMP não possui a necessidade de estar fisicamente na mesma rede dos equipamentos gerenciados, pois se utiliza da camada de Rede para acesso aos alvos monitorados.

Em resumo a (RFC1157, p. 3), o Modelo para gerenciamento em que o SNMP está baseado, consiste em:

- Estrutura e identificação das informações de gerenciamento (*Management Information*): Descrevem como os objetos da base de dados (MIB) são definidos;
- Base de gerenciamento de informações para redes IP: Descrevem os objetos gerenciáveis contidos na MIB; e,
- O SNMP: Definição do protocolo usado para gerenciar estes objetos.

Segundo (Melhores práticas para gerência de redes de computadores, p17), a arquitetura geral dos sistemas de gerência de redes apresenta quatro componentes básicos: elementos gerenciados (Agentes), estações de gerência (Gerente), protocolos de gerência (SNMP) e base de dados com informações de gerência (MIB).

Em relação à arquitetura, o SNMP segue o modelo cliente-servidor, dividido em Agente e Gerente. Apesar da similaridade e até mesmo em certo ponto a proximidade de sinônimos, aqui o Agente não é o cliente, mas sim o servidor.

2.2.1.1 Gerentes SNMP

Com base em (Mauro & Schmidt, 2005) um gerente SNMP é uma estação de gerenciamento (usada, em geral, para armazenamento dos dados coletados), executando um tipo de Software que manipula as tarefas de gerenciamento da rede. Este é visto como o cliente, consultando os dispositivos monitorados (servidores ou ativos de rede), sobre certa informação pertinente à sua gerência.

Segundo (Comer, p. 343), Um gerente chama o software cliente no host local, e especifica um agente com o qual deseja se comunicar. Após contatar o agente, o cliente envia consultas para obter informações ou comandos para mudar condições no agente contatado. O software que implementa o gerente serve de interface entre o gerente Humano e a estação de gerenciamento, para chegar ao gerenciamento da rede propriamente dito.

2.2.1.2 Agentes SNMP

O Agente é a parte servidora do SNMP. Ele responde às consultas feitas pelos Gerentes SNMP, podendo ainda trabalhar com mensagens assíncronas (através de *Traps*).

Segundo (Mauro & Schmidt, 2005), o agente pode ser um programa separado (algo como um *Daemon* em sistemas *NIX), ou incorporado ao Sistema Operacional, como no caso de um roteador Cisco, ou ainda em um nível mais baixo como em um sistema operacional básico que controla um *UPS*.

O agente provê informações sobre si, como por exemplo, o estado de suas interfaces de rede, a quantidade de *Bits* trafegados entre suas conexões, taxa de erros, tempo de atividade (*uptime*), dentre outras.

Os Agentes ou Gerentes, não especificam quais dados podem ser acessados ou o formato dos mesmos. Para isto existe outra especificação: a MIB.

2.2.1.3 MIB

Em resumo a (Comer, p. 344), a MIB (*Management Information Base*) é uma base de dados com os itens disponíveis ao gerente e especificação das possíveis operações de acesso a estes itens.

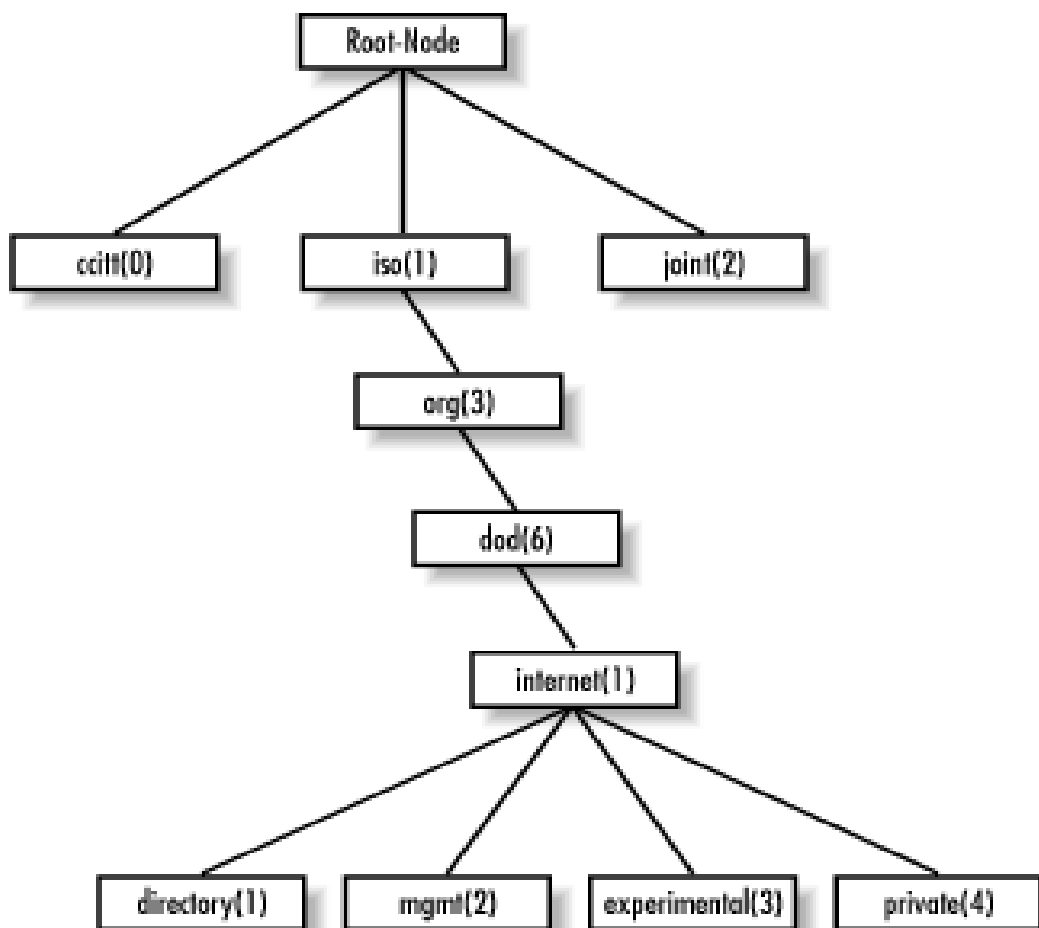
Segundo (RNP, 1997) A MIB é uma base de dados, cuja estrutura é especificada pelo padrão SMI (...). Ela pode ser caracterizada como um

banco de dados ativo, o que possibilita que os valores das suas variáveis sejam não só recuperados, como também alterados. Cada agente deve manter sua própria instância da MIB, relacionada com os objetos que estão sendo gerenciados sob o seu domínio.

O padrão SMI implementa um modelo de acesso aos dados, baseado em OID (*Object Identifier*). Segundo (Comer, p. 346), Um nome de um objeto na hierarquia (OID) é a sequência de rótulos numéricos, ao longo de um caminho, da raiz até o objeto (como é uma árvore).

Um exemplo de árvore identificadora de OIDs, poder ser verificada através da Figura 7.

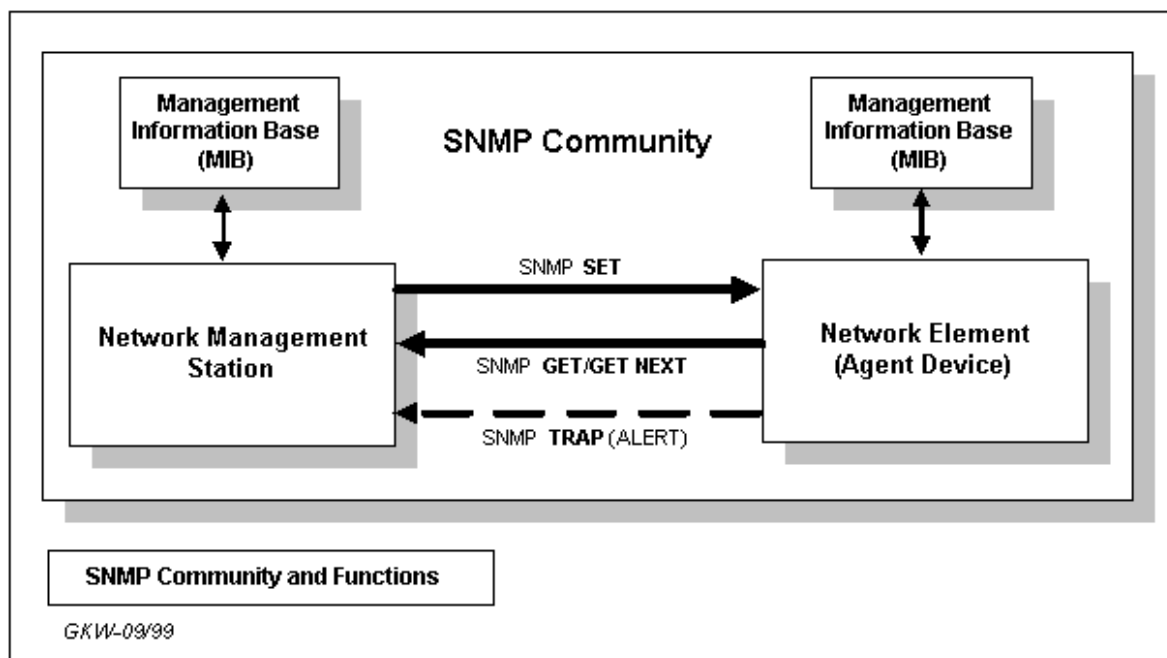
Figura 7: Um exemplo de árvore OID.



Fonte: (Mauro & Schmidt, 2005)

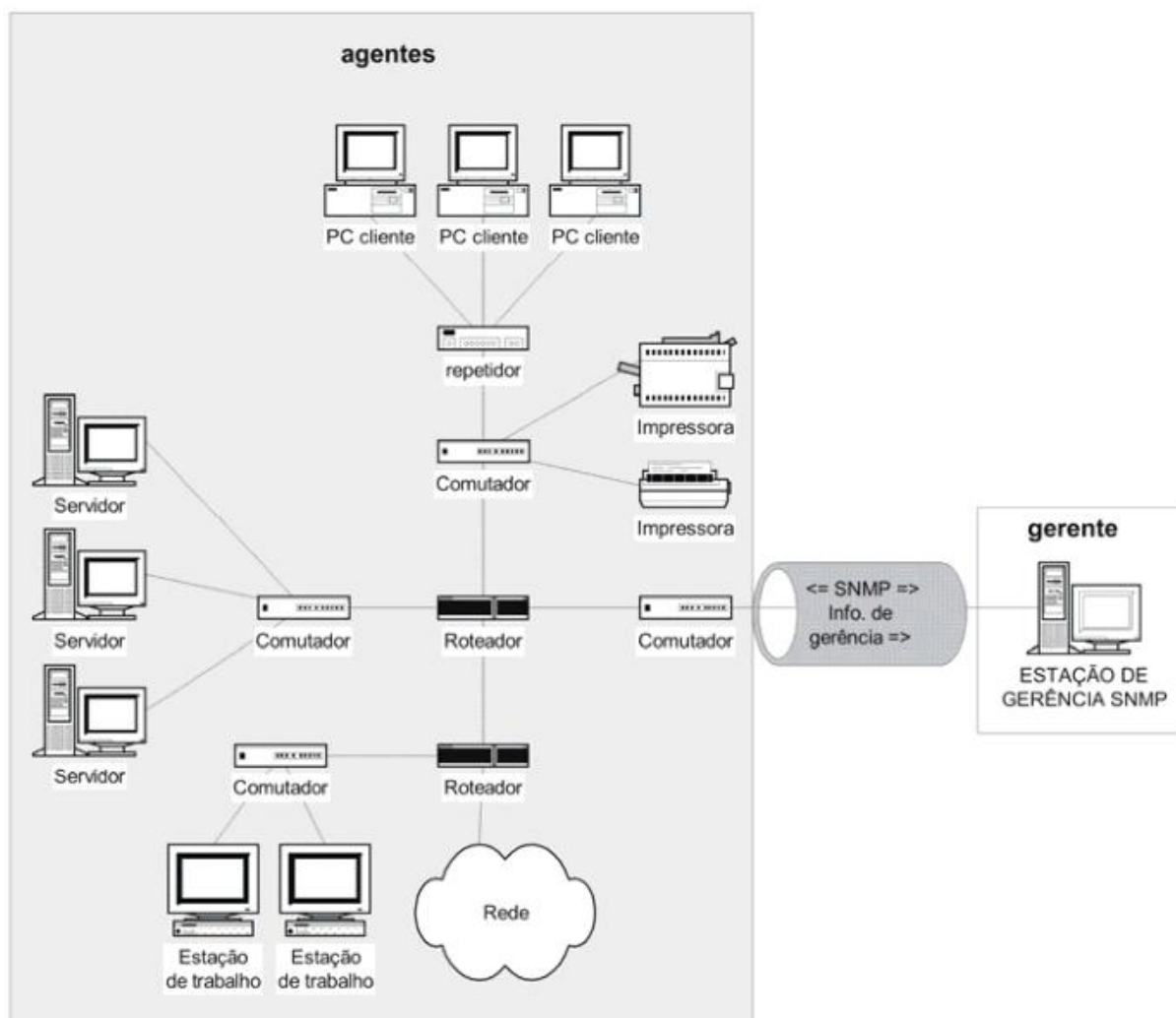
Os três itens comentados interagem no sentido de colocar em prática o conceito por trás do Modelo SNMP. Um exemplo da interação entre Gerente, agente e MIB pode ser verificado através da Figura 8.

Figura 8: Exemplo de interação entre os componentes do Modelo SNMP.



Por ter se tornado, ao longo dos anos, uma abordagem padrão para monitoramento de redes, o Modelo SNMP está presente em diversas implementações de Equipamentos e Sistemas. A Figura 9 demonstra um exemplo da gama de dispositivos monitorados por este Modelo.

Figura 9: Alguns exemplos de dispositivos monitorados através do SNMP.



Fonte: (Nicolletti, Lopes, & Sauv , p. 18)

2.2.1.4 Opera  es SNMP

Segundo a (RFC1157, p. 6), o protocolo SNMP possui tr s opera  es b sicas:

- get: esta  o gerente obt m por   es de dados na esta  o que implementa o agente;
- set: esta  o gerente configura valores na esta  o que implementa o agente;
- trap: esta  o que implementa o agente, envia dados n  solicitados   esta  o gerente (geralmente servindo como Alerta).

As demais intera   es s o baseadas nas principais.

2.2.1.5 Versões

O SNMP nasceu e cresceu com a missão de ser o protocolo padrão para gerência da internet. Possui atualmente três versões. Através da Tabela 3, é possível verificar detalhes sobre as três versões disponíveis do protocolo SNMP.

Tabela 3: Versões, datas, RFCs e observações do protocolo SNMP.

Versão	Data de Homologação	RFCs envolvidas*	Forma de acesso a dados	Operações
V1	1990	1157	SNMP MIB View (texto claro)	get, get-next, get-response, set-request e trap.
V2c	1996	1901 à 1908, 3416, 3417	Comunidades (communities) texto claro.	OpV1 + get-bulk
V3	1999	2570, 3410, 3411, 3412, 3413, 3414, 3415, 3416, 3417, 3584, 3826, 5343	VIEW-Based: Integridade de dados, autenticação da origem e confiabilidade baseadas em criptografia, autenticação e mensagens tipo HASH.	Mesmas da V2c

Fonte: SNMP Research (SNMP Research INC)

* Certas RFCs citadas, podem não ser o descritivo do protocolo, mas estão associadas ao mesmo de forma complementar.

Uma observação importante é que mecanismos de segurança foram propostos desde a primeira versão do SNMP (RFC2570-Sec, 1999), mas que, por diversos motivos, não foram implementados pelos fabricantes.

2.2.2 Baseados em Agentes Próprios

O modelo SNMP serviu de base para que novas ferramentas de monitoramento fossem criadas. Estas ferramentas não seguem os padrões e RFCs comentadas nas páginas anteriores sobre SNMP, apenas sua ideia principal: uma estação de Gerência, coleta dados de um Agente remoto com base em métodos próprios e acessando informação gerenciáveis (em geral, coletadas em tempo real).

Com esta abordagem não é padrão (como o SNMP), fabricantes de *Hardware* ou *Software* dificilmente as implementam em seus produtos, ficando a cargo (quando possível) dos *Administradores de Redes*.

2.2.3 Outros (comandos auxiliares de sistema)

De acordo com (Nicolletti, Lopes, & Sauv , p. 19), al m do Sistema de Ger ncia de Redes, outras ferramentas nos auxiliam a gerenciar uma rede. Dentre elas encontram-se analisadores de protocolos e outras ferramentas mais simples como os comandos *ping*, *traceroute* e *netstat*, dispon veis em v rios sistemas operacionais.

Estes comandos n o s o espec ficos de esta  es gerenciadoras. Eles est o dispon veis nos Sistemas Operacionais clientes de Rede, como por exemplo, Microsoft Windows XP, Microsoft Windows 7 e 8 (e suas variantes), Linux Ubuntu 12.04 Desktop, Fedora Core 18 e demais distribui  es.

Independente da abordagem escolhida, existem Softwares destinados a este fim; s o os NMS (*Network Management Systems – Sistema de Gerenciamento de Redes*). Os que se destacam s o os que possibilitam o uso de v rias abordagens das citadas   cima (umas complementando as outras).

2.2.4 Monitoramento na Pr tica

A abordagem que se utiliza do Modelo SNMP n o   muito usual se aplicada de forma manual e isolada, pois nem todos os dados puros (que chegam atrav s das consultas unit rias) s o suficientemente informativos se vistos isoladamente. Ela   geralmente amparada por Sistemas de Gerenciamento de Redes (NMS).

Baseando-se em (Nicolletti, Lopes, & Sauv , 2003, p. 17),   poss vel perceber que este tipo de sistema captura e processa dados de monitoramento, oferecendo ao *NETAdmin* acesso a uma interface que consolida informa  es sobre a rede. Pode oferecer tamb m, um conjunto de comandos que s o usados para executar certas tarefas da ger ncia da rede como, por exemplo, testes de resposta, rotas, tradu  o de endere os.

O mercado de TIC   um imenso mar de oportunidades. Estas oportunidades se estendem tamb m    rea de monitoramento/ger ncia de redes. Com a

diversidade de produtos (pagos, gratuitos e livres) se torna árduo o trabalho em escolher um que atenda as necessidades diversas desta crescente área. Estas necessidades de gerência podem ser elencadas, não de forma a esgota-las, na seguinte lista:

- Suporte a múltiplos fabricantes;
- Suporte a diferentes abordagens de monitoramento;
- Suporte a diferentes métodos de monitoramento;
- Facilidade nas operações;
- Suporte a interação visual para ações ágeis;
- Bom suporte a plataformas de alertas;
- Facilidade de manutenção;
- Atualização que atendam as novas demandas.

Além destes pontos, é preciso pensar que como a maioria das ferramentas é incompatível entre si (não trocam dados diretamente), a ferramenta escolhida provavelmente será utilizada por um longo tempo.

2.3 ZABBIX

Com base em (Zabbix SIA - About, 2012) e (Olups, 2010), o Zabbix é uma Solução para monitoramento distribuído, com código aberto (*OpenSource*) e nível empresarial (no sentido de seriedade e capacidade).

Segundo tradução livre de (Zabbix SIA - About, p. About), Zabbix é um software que monitora inúmeros parâmetros de uma rede além da “saúde” e integridade de servidores. Zabbix utiliza um mecanismo de notificações flexível, que permite ao usuário configurar alertas baseados em e-mails para, virtualmente, qualquer evento. Isto permite reações rápidas contra problemas (...). o Zabbix oferece excelentes relatórios e a possibilidade de interação visual com dados armazenados previamente (...). Todos os relatórios e estatísticas, bem como parâmetros de configuração, são acessados através de uma interface WEB, possibilitando o acesso à partir de qualquer lugar.

A logomarca do Software pode ser vista na Figura 10.

Figura 10: Logomarca Zabbix.



Fonte: (Zabbix SIA - Homepage, 2013)

Com bases na experiência do autor na utilização e pesquisa em Softwares Livre para Monitoramento (citando [Nagios](#) e [Cacti](#), por exemplo), foi escolhido o [Zabbix](#) pela junção dos aspectos julgados como mais importantes em um sistema de monitoramento: facilidade de administração, ótimo suporte a ambientes heterogêneos e múltiplas abordagens de monitoramento. Um trabalho que executa uma comparação de diversas ferramentas de monitoramento, levantando itens interessantes, pode ser visto em (Black, 2008). Outro trabalho que levanta as características entre as citadas ferramentas é (Majewski, 2009).

2.3.1 História

A partir de informações disponíveis em (Zabbix SIA - Presentation, 2012), (Wikipedia INC, 2013) e (Zabbix SIA - News, 2004-2013), é possível verificar que o Zabbix teve seus primeiros momentos de desenvolvimento em meados de 1998. Criado por *Alexei Vladishev* (<http://lv.linkedin.com/pub/alexei-vladishev/7/273/8a1>) como um projeto interno em um Banco ao qual o mesmo trabalhou, teve sua primeira versão estável lançada em Março de 2004. Na época de confecção desta seção do trabalho, a versão estável era 2.0.6 e, segundo (Zabbix SIA - News), foi disponibilizada em 22/04/2013.

2.3.2 Licenciamento

Segundo (Zabbix SIA - Licence), a licença seguida pelo Zabbix é a GNU General Public License versão 2 (GPLv2). Segundo (Free Software Foundation, 1992) A licença, dentre outras coisas, prevê aos usuários do Zabbix:

- Executa-lo livremente;
- Copiar e distribuir seu código fonte;
- Modificar sua cópia (em qualquer porção), e distribuí-la;
- Você não pode alterar as licenças de programas baseados no principal.

2.3.3 Arquitetura

Segundo (Zabbix SIA - Overview, 2012) e (Olups, p. 9), a solução Zabbix é composta por cinco elementos:

- **Server (Servidor):** É o componente central da arquitetura para o qual agentes reportam dados de disponibilidade e estatísticas. Nele, ficam guardadas todas as configurações e dados estatísticos e operacionais. O componente servidor está envolvido principalmente em coletar e gravar dados, disponibilizar mecanismos de Alerta e visualização de dados.
- **Database Storage (Armazenamento em banco de dados):** Local “físico” para o armazenamento de dados. Componente Banco de Dados Relacional que armazena e relaciona os dados em tabelas.
- **WEB Interface (interface HTML):** *Interface WEB* provida para dar acesso ao Servidor Zabbix, a partir de qualquer lugar e a qualquer dispositivo que entenda HTML.
- **Proxy:** Componente opcional para monitoramento distribuído. O Proxy pode coletar dados de disponibilidade e performance à favor de um Servidor Zabbix. Pode ser benéfico ao servidor Zabbix, distribuir a carga de monitoramento entre vários Proxys. Além disso, com o Proxy é possível monitorar ambientes onde a segurança é mais restrita ao ponto de não ser permitido acesso as configurações do Firewall, (restrições com relação a portas de Entrada/Saída). Esta parte do Zabbix, pode trabalhar de forma Ativa, onde ele busca o servidor, verifica que Hosts/itens devem ser consultados e se torna o “alimentador” do servidor, ou de forma Passiva, onde o servidor é quem se conecta e passa a solicitar, de tempos em tempos, os dados de monitoramento. O Proxy não substitui o servidor, uma vez que, segundo (Zabbix SIA - Distributed Monitoring), não suporta alguns itens-chave disponíveis no servidor.
- **Agent (Agente):** Agentes Zabbix são instalados para monitorar Servidores de rede, coletando dados de seus recursos e aplicações locais, e enviando-os para o servidor Zabbix. O agente monitora itens.

Estes itens seguem a lógica dos das MIBs do SNMP, porém ainda não fazem parte de um padrão globalmente adotado pelo mercado (apesar de ter a licença GPL). Porém, o agente estende muito a capacidade de monitoramento, pois proporciona aos Administradores de Redes a possibilidade de customizar itens e ações executadas diretamente no sistema operacional monitorado.

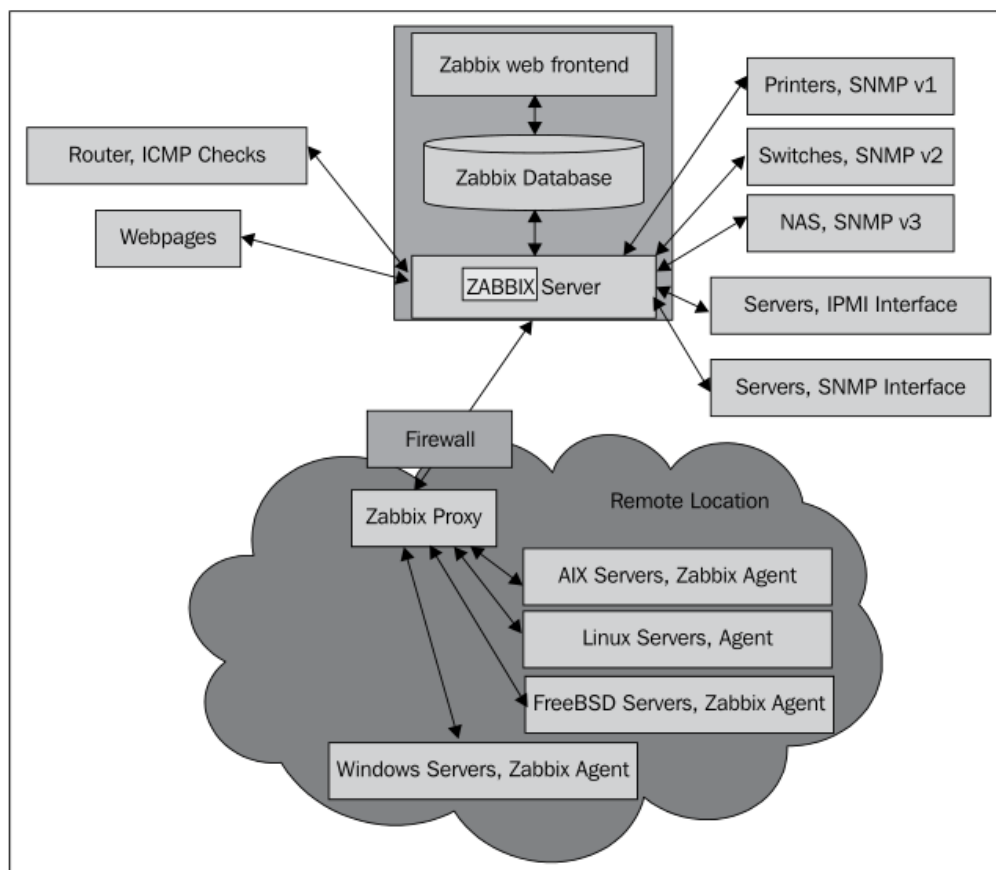
Estes componentes, apesar de serem listados separadamente, podem estar instalados no mesmo equipamento, depende do tipo (e tamanho) da instalação e sua arquitetura de monitoramento. O exemplo mais comum é ter o Server, Interface e Banco de Dados no mesmo servidor Físico.

O Servidor Zabbix pode obter dados dos alvos a serem monitorados através de diversos meios. Segundo (Zabbix SIA - Features, 2011) e (Olups, p. 9), estes são alguns dos métodos de coleta de dados:

- Via SNMP, IPMI, e monitoramento JMX (Java);
- Checagem customizada (através do uso de programação com Scripts);
- Instalação de agentes nativos em sistemas operacionais Baseados em UNIX ou Microsoft Windows;
- Monitoramento WEB, verificando os códigos de resposta às requisições feitas a servidores HTTP;
- Através de dados passados por agentes e Proxys Zabbix.

A Figura 11 dá uma ideia visual resumida sobre as características comentadas à cima.

Figura 11: Componentes Zabbix e métodos de obtenção de dados.



Fonte: (Olups, p. 9)

Os ativos monitorados podem pertencer a redes de qualquer dimensão e localidade, sejam LANs, CANs ou WANs.

2.3.4 Suporte visual

Como visto nos itens anteriores, o Zabbix possui componentes responsáveis por obter e gravar dados. Estes dados podem ter importância a curto, médio ou longo prazo (dependendo da avaliação dos Administradores de Redes). Porém estas informações estão armazenadas em tabelas de banco de dados, em um formato humanamente compreensível, porém não organizado (do ponto de vista de facilitação visual).

Para que seja possível a compreensão destes dados, o Zabbix oferece Elementos visuais para facilitar a **percepção** de eventos e entendimento das **alterações** no ambiente monitorado, além de permitir a visualização dos dados ao longo de períodos de tempo.

Segundo (Zabbix SIA - Visualizations), como formas de visualização de dados, o Zabbix oferece Gráficos (simples e customizados), Mapas de rede, Telas (Screens) e Slide-Shows. Segundo (Olups, p. 222), os Elementos visuais podem ser separados em duas categorias: Simples e Compostos.

2.3.4.1 Elementos simples

Gráficos Simples: itens de visualizações pontuais e direcionados a um item monitorado, como por exemplo, o consumo de um recurso em específico (como processador ou Memória). Não exige configuração, sendo gerados por padrão à partir da coleta de dados pelo servidor. Um exemplo pode ser visto através da Figura 12.

Gráficos Customizados: elementos que permitem a adição de mais de uma fonte de dados (memória e processador, por exemplo). Além da combinação de itens, pode ser alterada a forma de apresentação dos dados (cores, tipo de gráfico, posição e etc). Um exemplo pode ser visto através da Figura 13.

Mapas de Rede: Recurso que combina a representação visual em ícones, para os itens monitorados em uma rede. Os mapas são configurados manualmente, e auxiliam na hora de entender onde ou como os ativos estão ligados (verificando linhas de dependência entre eles, por exemplo). Em um mapa é possível configurar condições de reação onde os itens mudam sua forma visual quando há uma mudança em seu estado (por exemplo, quando o ativo parou de responder às requisições). Um exemplo pode ser verificado através da Figura 14.

Figura 12: Exemplos de gráficos gerados pelo Zabbix.

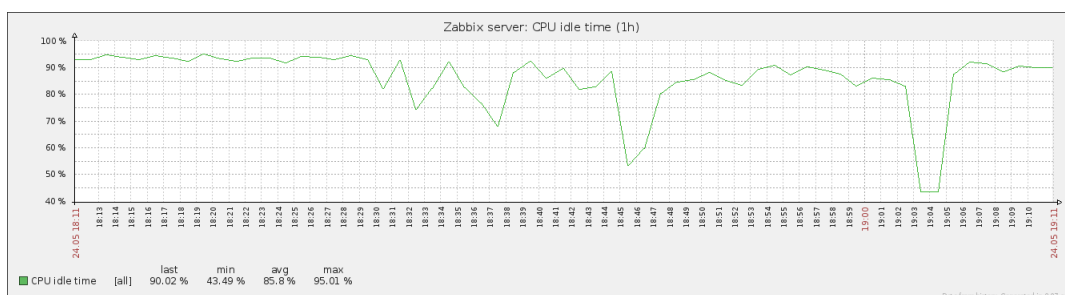


Figura 13: Exemplo de gráfico Customizado

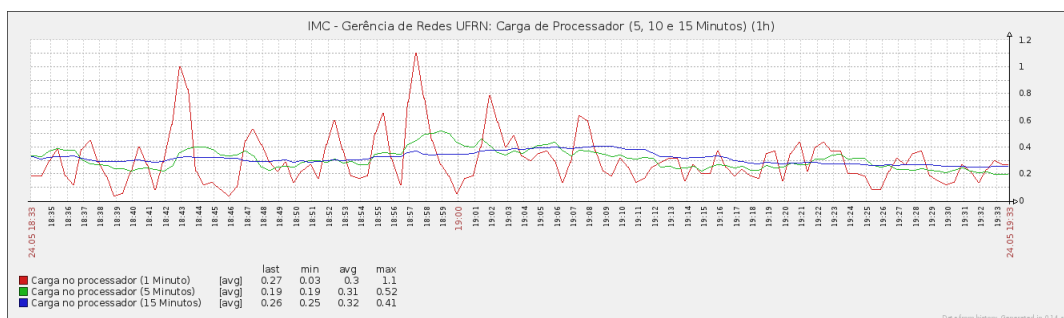
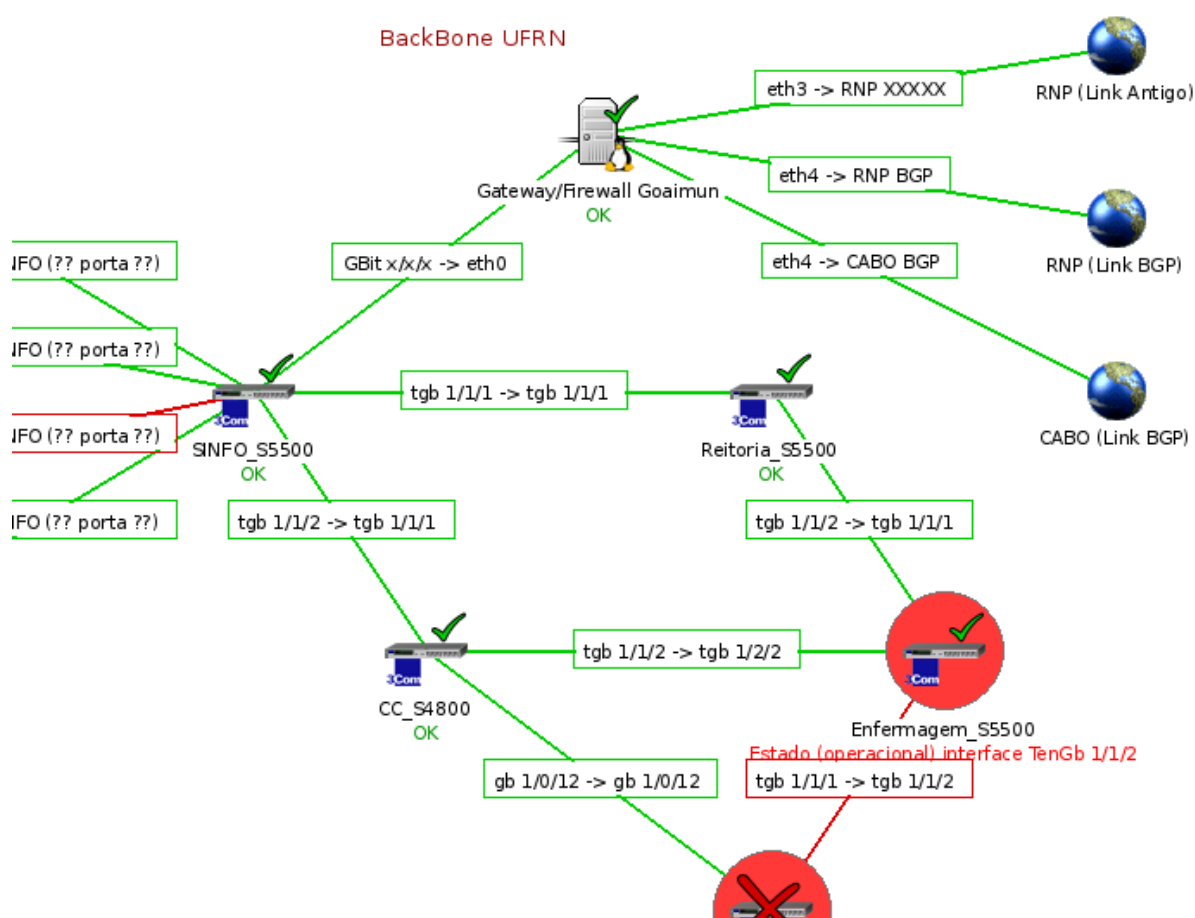


Figura 14: Exemplo de Network Map (Mapa de Rede).



Os mapas, apesar de serem montados com vários elementos unitários, são considerados elementos simples, pois o tipo de item adicionado é um só: ativos de rede. Além disso, os itens simples (tanto Gráficos quanto mapas) são exibidos de forma separada e apenas um por vez.

2.3.4.2 Elementos compostos

Apesar do nível de informações que é possível se obter com os Elementos simples, nos casos práticos é necessário se ter uma visão simultânea de várias áreas do ambiente monitorado. Segundo (Olups, p. 250), um exemplo seria a visão simultânea do Mapa da rede, de um gráfico com o registro do consumo do *Link* principal de internet e uma lista dos problemas atuais na rede.

O recurso que agrupa vários itens simples é chamado de Screens (telas, em tradução livre).

Segundo (Zabbix SIA - Visualisation (Screens)), nos *Screens* Zabbix, é possível agrupar informação proveniente de várias fontes em um ponto único para visualização rápida. Montar as telas é simples e intuitivo.

Os Screens podem acumular vários itens simples para monitoramento, porém para que seja útil o agrupamento destes itens, é preciso ter um meio de vê-los em seu formato expandido e confortável. Uma boa maneira seria utilizar uma TV de um tamanho à partir de 32', para apresentação de Screens de maneira visualmente mais confortável.

3 DESENVOLVIMENTO

3.1 CONTEXTUALIZAÇÃO DO AMBIENTE

Um ator que fomenta o crescimento da internet, através do oferecimento de aplicações e/ou serviços, é o Provedor de Serviços de Internet (ISP – *Internet Service Provider*). Estes provedores comportam e mantêm uma estrutura tecnológica em seus domínios (físicos e lógicos) com a pretensão de atender a uma comunidade de usuários (clientes), sejam eles pagantes diretos pelos serviços oferecidos, ou usuários de bens públicos nas esferas municipal, estadual ou federal (pagando através do recolhimento de tributos e contribuições afins).

Entendamos aqui **Serviços** como tecnologias que compreendam (total ou parcialmente):

- Acesso à internet: O provedor fornece a conexão entre o cliente e a internet;
- Conteúdo: WEB Sites, Blogs, portais e correlatos;
- Serviços Básicos de Rede: DNS, DHCP, NTP, FTP e congêneres;
- Acesso remoto: VPN, VNC, RDP e congêneres;
- Intercomunicação pessoal: *Instant Messaging*, *VoIP*, E-mail, Vídeo Conferências, telepresença e etc.

Neste sentido as novas demandas de gerenciamento transpassam a simplicidade imposta pelo modelo SNMP. Não se trata mais de obter informações sobre o estado ou consumo de determinada parte de um sistema em separado, mas sim observar comportamentos inadequados antes que o cliente final venha a sentir seus efeitos. Estes efeitos podem potencializar grandes perdas para o provedor, desde a perda de credibilidade até grandes perdas financeiras.

Os provedores de serviço dependem de, de forma macro, três estruturas: a interna, a borda e a externa.

3.1.1 Estrutura Interna

A estrutura interna trata dos componentes de rede dos quais o provedor tem domínio completo, ou seja, são os *Switchs*, Roteadores (no âmbito de LAN, CAN ou MAN), servidores e demais ativos de rede. Juntos, estes ativos fornecem a quem

vem de fora, o acesso aos Serviços hospedados internamente, ou ainda, fornecem caminhos de saída da rede interna para a internet caso o provedor ofereça o acesso à internet como um serviço.

3.1.2 Estrutura de Borda

É a fronteira entre a rede interna e a externa. A borda pode contar com serviços de Roteamento exterior, o qual envia e recebe informações sobre suas Rotas ou redes vizinhas conectadas. Estes serviços de roteamento não estão presentes em todos os provedores de serviço, mas podem estar naqueles que são Sistemas Autônomos (AS – *Autonomous Systems*).

Além de serviços de Roteamento, a borda também é composta por dispositivos de segurança perimetral (como o Firewalls, IPSs/IDSs e UTMs) que responde por:

- Limitar o acesso, de quem está fora da rede, a serviços internos públicos;
- Registrar/Bloquear ações que visem atacar a disponibilidade, confiabilidade ou integridade dos serviços oferecidos;
- Direcionar aos ativos corretos as requisições externas por serviços internos;
- Separar redes em limites de acesso e função, dividindo-as em (por exemplo) WAN, DMZ e LAN.

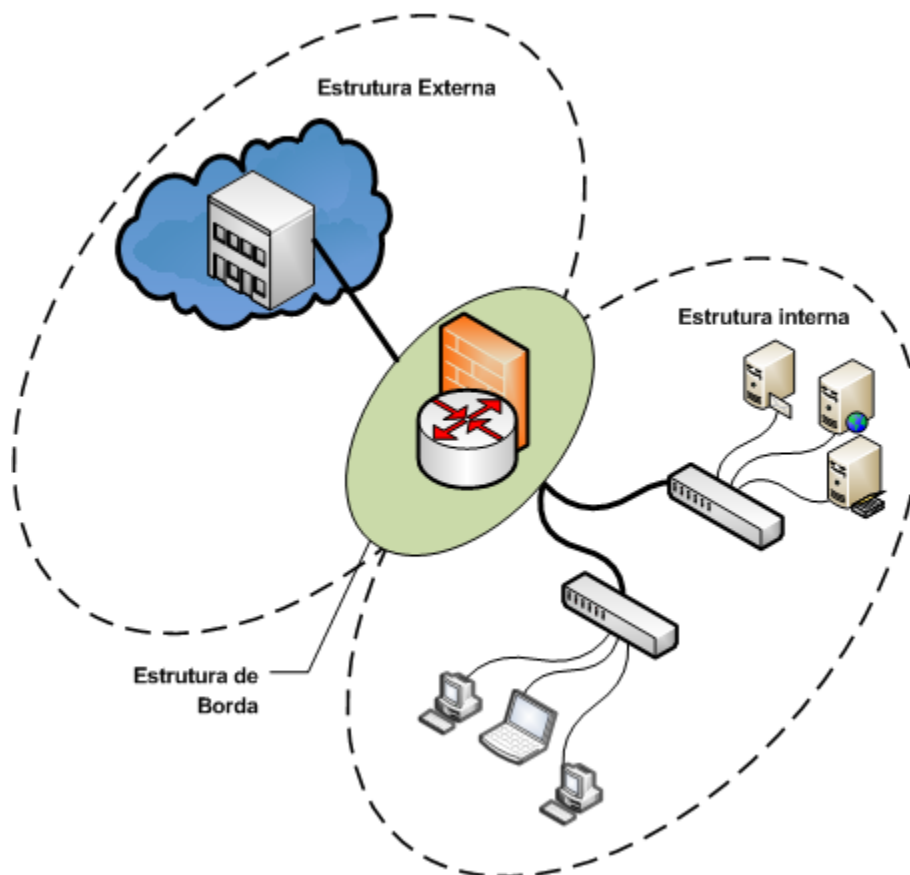
Nesta parte da estrutura, o provedor ainda possui a responsabilidade e poder sobre a maior parte dos equipamentos. O controle não é total devido ao fato de que as informações advindas da rede externa (com relação a Rotas, por exemplo), serem controlada “pela outra ponta” e por vezes chegarem com dados incorretos.

3.1.3 Estrutura Externa

Esta parte da estrutura é controlada pelo “provedor do provedor”, ou seja, para que o Provedor de serviços ofereça estes serviços à comunidade, como comentado, ele precisa de uma estrutura de nível superior que o forneça acesso ao *Backbone* de telecomunicações. Neste nível, o provedor usa a sua Estrutura de Borda para se comunicar com o provedor de Backbone.

Esta é uma parte da estrutura que foge totalmente do controle do provedor de serviços. A Figura 15 pode dar uma ideia sobre a posição das estruturas comentadas.

Figura 15: Estrutura interna, de borda e externa.



Fonte: O Autor.

3.1.4 Dependência Entre as Estruturas

Fica clara a dependência que uma estrutura exerce sobre a outra no sentido de que se a estrutura de Borda não estiver disponível ou com algum problema de configuração, os serviços não serão publicados e os clientes não os acessarão. Da mesma forma acontece se algum ativo interno estiver inacessível, impedindo o caminho até um servidor que hospede páginas WEB, por exemplo.

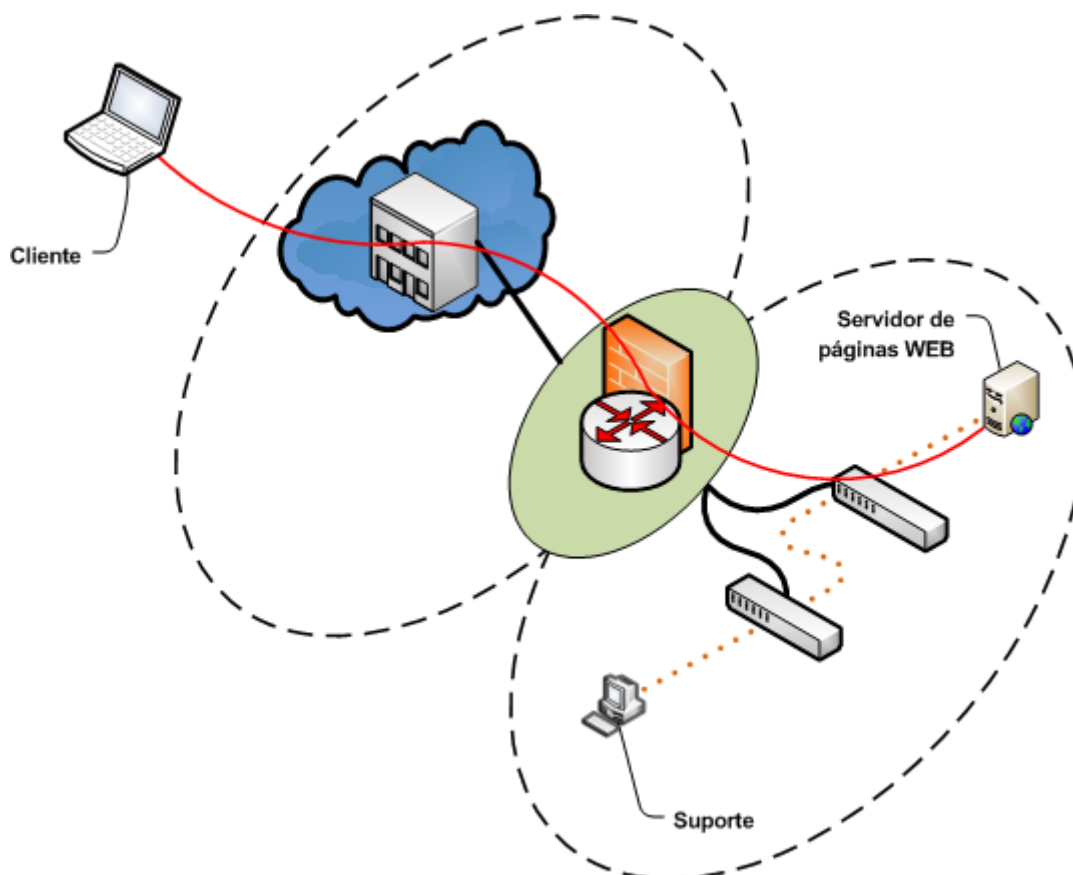
3.2 PERCEPÇÃO DOS SERVIÇOS PELOS CLIENTES

No momento em que um cliente percebe que o serviço demandado não está acessível, normalmente ele entra em contato com o Suporte oferecido pelo provedor de serviços para buscar a correção ou a informação a respeito do problema.

É muito comum que estas equipes de Suporte deem respostas aos clientes com base em testes internos (mesmo para serviços acessíveis externamente). Muitas vezes, estas respostas são algo como “estou testando, e está tudo OK... O problema deve estar aí!”. Naquele momento o cliente não tem a mesma sensação que o suporte, pois os recursos utilizados por ele, para acessar o serviço oferecido pelo provedor, podem estar tendo respostas diferentes das que a equipe de Suporte recebeu e repassou.

Isto acontece porque, em geral, as equipes de suporte estão fisicamente na mesma estrutura interna e a poucos “saltos lógicos” das redes às quais estão hospedados os serviços acessados pelos clientes. Ou seja, no contexto das estruturas, a equipe de suporte vem diretamente da estrutura interna e o cliente vem da estrutura externa. Isto causa um falso-positivo de bom funcionamento, visto que internamente as coisas dificilmente não estarão funcionando. Através da Figura 16, é possível visualizar este comportamento.

Figura 16: "Caminho" feito pelo Cliente versus Equipe de Suporte.



Fonte: O Autor.

Para dar uma melhor resposta ao cliente é importante que a equipe de suporte tenha a visibilidade dos serviços de rede não só internamente. A equipe precisa perceber se os serviços oferecidos estão conseguindo sair da rede interna do provedor de serviços, completando o caminho como se chegassem ao cliente.

Nota-se que certas partes envolvidas no acesso a estes serviços, não dependem do provedor de serviços, sendo de responsabilidade do provedor de *Backbone* ou de estruturas superiores, todavia impactam diretamente os clientes do provedor de serviços. Neste ponto, o suporte talvez não possa ajudar o cliente a resolver, mas, o mínimo que o suporte precisa prestar é informação sobre prováveis pontos de falha.

3.3 MONITORAMENTO UTILIZANDO A NUVEM

É preciso destacar a importância do *Feedback* externo, pois remete à percepção que os Clientes tem sobre a disponibilidade dos serviços oferecidos pelo

provedor. Porém, não é desejável que este *Feedback* seja dado por um cliente Real pois isto significaria insatisfação.

Neste sentido, é notável a proximidade entre o Zabbix e as necessidades reais do mercado de monitoramento. Ele possui componentes fracamente-acoplados possibilitando divisão dos mesmos em instalações distribuídas e conta ainda com a possibilidade de se utilizar várias abordagens de monitoramento combinadas.

Para a execução deste trabalho foi traçado o perfil de um provedor de Serviços, com sua topologia e principais necessidades de monitoramento. E para executar este monitoramento foi utilizada a ferramenta Zabbix com o diferencial de executar os procedimentos de testes utilizando um Zabbix Proxy baseado na Nuvem pública (*Public Cloud Computing*).

Este trabalho não pretende ser referência para instalação, prova de conceito ou funcionalidades básicas do Software Zabbix, pois não se vê a necessidade de provar o quanto eficaz é a ferramenta. Para este fim, já existem trabalhos como (PUSKA, 2011), (BONOMO, 2006) e o manual oficial (Zabbix SIA - Instalation, 2012).

Para a montagem e execução do trabalho, foi idealizado um provedor privado de serviços de internet que atende a um público diretamente pagante, a exemplo de tantos provedores disponíveis atualmente. Seus serviços serão trabalhados de forma genérica, no intuito de demonstrar que esta abordagem pode ser aplicada em qualquer provedor, independente de seu tipo ou tamanho. O provedor fantasia foi denominado **EnterpriseWEB**.

Para a instalação do Proxy externo, foi utilizada uma instância AWS no plano Gratuito, com a seguinte configuração:

- Um servidor EC2 com Imagem do sistema operacional Linux Ubuntu 12.04 LTS, imagem AMI *ubuntu-precise-12.04-amd64-server-20130124*;
- Um volume EBS de 10GB, apresentado ao servidor para instalação do Sistema Operacional;
- Um banco de Dados RDS com 5GB, utilizando a Engine *MySQL 5.5.27*;

3.3.1 Provedor a Ser Monitorado

A EnterpriseWEB é uma empresa privada que atua como Provedor de serviços via internet nas modalidades:

- Hospedagem de conteúdo: WEB Sites, Blogs, portais, *Wikis*, e-commerce e correlatos (HTTP/HTTPS);
- Serviços Base de Rede: DNS, NTP e FTP;
- Intercomunicação pessoal: e-mail.

Seus clientes diretos são empresas que locam sua plataforma no formato “tudo em um”, contando com um pacote de serviços (Servidor WEB, e-mail e FTP) que ofereçam uma espécie de infraestrutura pronta e fácil de manusear, basicamente uma pilha *LAMP* (Linux + Apache + MySQL + PHP).

Não se trata de Servidores Virtuais Privados (*vps*) diretamente, mas sim de conjuntos de aplicações e serviços compartilhados essenciais à criação de Sites e portais. Foi abstraído o serviço de Banco de Dados (*SGDB*), por ser um serviço que, apesar de impactar diretamente no funcionamento da maioria dos Sites e Portais atuais, não está diretamente visível ao cliente final.

Para os serviços supracitados, o monitoramento compreenderá os itens presentes na Tabela 4.

Tabela 4: Itens a serem monitorados nos servidores da EnterpriseWEB.

Servidor	Itens a monitorar	
	Porta(s)	Acesso
DNS	53 (UDP)	Consulta a um registro do domínio próprio
		Consulta a um registro de domínio (outros)
WEB	80/443 (TCP)	Homepage
		Pesquisa nas páginas subsequentes
		Acesso a Webmail

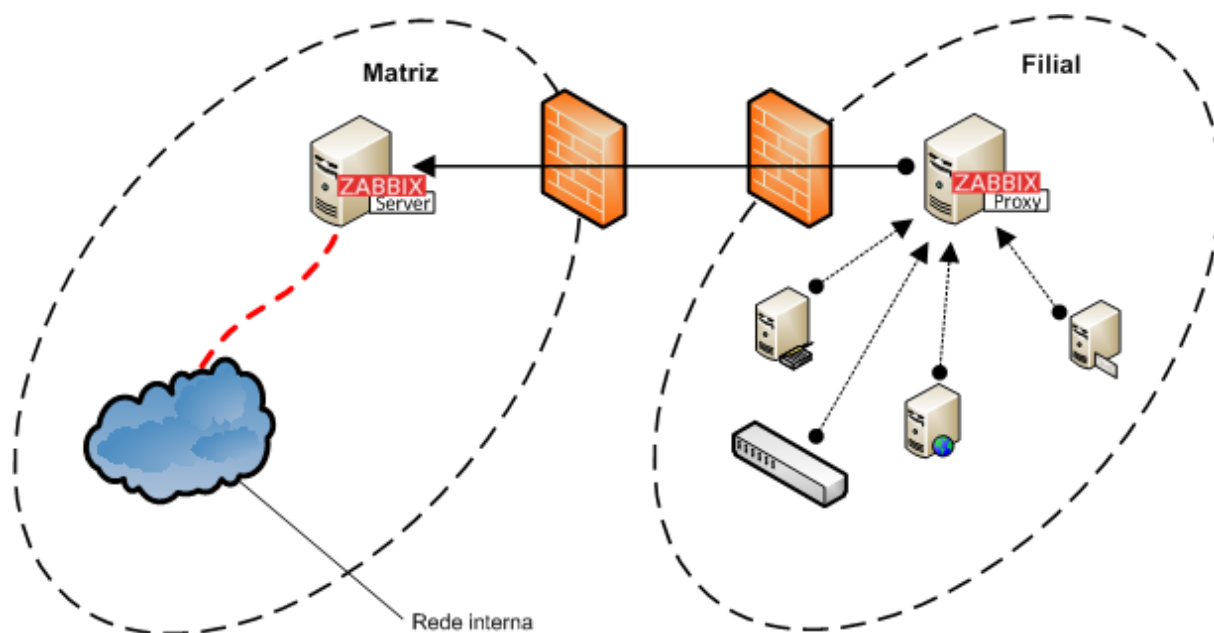
Fonte: O Autor.

3.4 MONITORAMENTO PROPOSTO

O principal objetivo do monitoramento proposto foi agir como um cliente externo, executando as mesmas funções que o cliente faria (ou a maioria delas), porém de forma automatizada, repetitiva e registrada. Com esta visão, a ideia é ter um monitoramento não intrusivo, sem a instalação de agentes Zabbix ou qualquer tipo de alteração nos serviços do ambiente (servidores monitorados). Apesar disso, os serviços precisam ser monitorados internamente, porém esta parte não foi abordada neste trabalho.

Para o monitoramento proposto neste trabalho, foi utilizado o Proxy Zabbix, porém de forma diferente do convencional. A Figura 17 dá uma melhor ideia de como é a maneira tradicional de se utilizar o Proxy.

Figura 17: Forma mais convencional para o Proxy Zabbix.



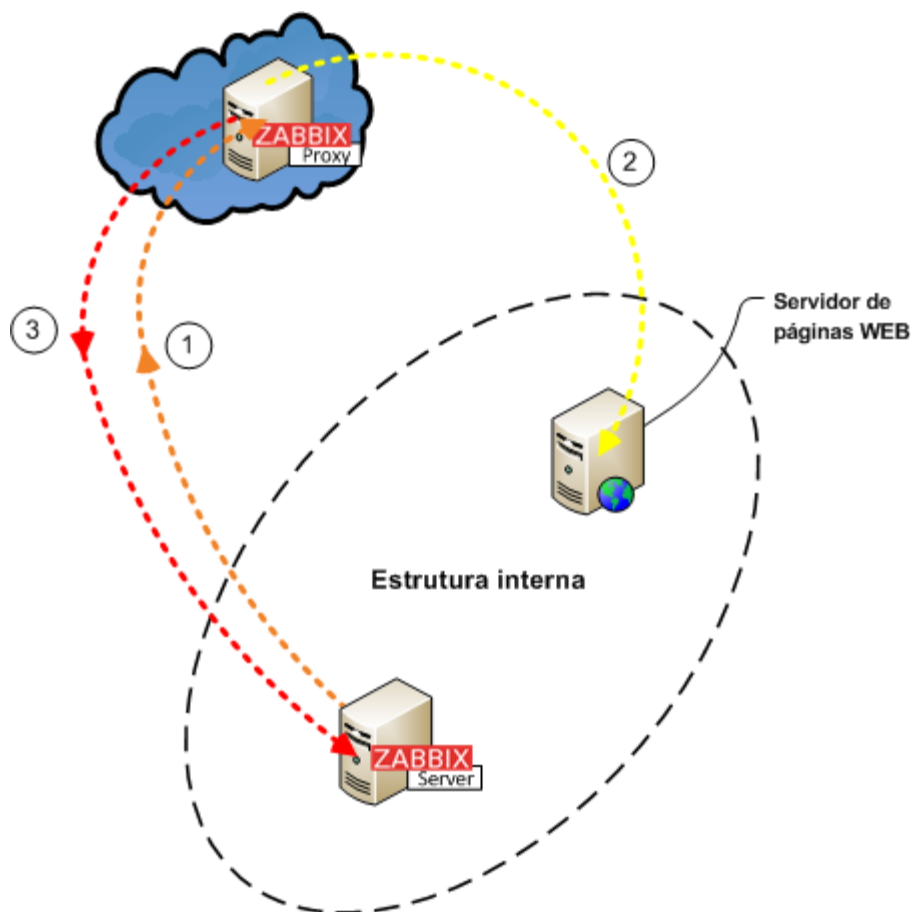
Fonte: O Autor.

A Figura 17 expressa o fluxo da informação em um ambiente monitorado de forma distribuída, onde um Servidor na Matriz recebe, processa e armazena as informações adquiridas pelo Proxy na Filial. O Proxy guarda as informações temporariamente, mas não possibilita a visualização diretamente (para isso, depende do servidor). As setas representam a origem e destino dos fluxos, mas não os limitam nas direções, pois o Proxy pode trabalhar de forma ativa ou passiva.

Neste trabalho, a proposta central, foi utilizar o Zabbix Proxy em um modelo diferente do convencional. A elasticidade e tipo de licença da ferramenta possibilita a adaptação de seu modo de trabalho visando atender as necessidades do interessado. Neste caso, foi alterada a forma padrão com que o Proxy trabalha para que, ao invés do monitoramento ser executado em locais remotos, ele monitorasse remotamente a localidade central. Na visão do provedor, a proposta foi monitorar a si mesmo, porém, no sentido “de fora pra dentro”, com o apoio da computação em Nuvem para que se tenha uma resposta realmente vinda de fora.

A Figura 17 representa a maneira como se pretende monitorar a condição externa dos serviços disponíveis.

Figura 18: Monitorando de forma alternativa com Zabbix Proxy.



Fonte: O Autor

Na Figura 18, as cores e números demonstram as fases, onde:

1. O Servidor envia ao Proxy a lista de Hosts e itens ao qual deseja obter dados (seta Laranja);

2. O Proxy recebe a lista, processa e inicia as verificações propostas (seta Amarela);
3. O Proxy retorna ao servidor as informações coletadas (seta Vermelha).

No caso apresentado na Figura 18, o Zabbix Proxy está na mesma posição que estaria um cliente externo, ou seja, fora da rede e percorrendo todas as estruturas (externa, borda e interna) para chegar até o serviço demandado.

Para esta abordagem, é necessário que o Zabbix Proxy esteja em uma rede logicamente distante, pois isso faz com que as verificações feitas na estrutura interna realmente tenham passado por diversos provedores e ASs diferentes, antes de chegar aos serviços oferecidos pelo provedor. Neste caso, foi utilizada a Estrutura da AWS (*Amazon Web Services*) para o posicionamento do Proxy.

3.4.1 Instalação e Configurações

O Proxy aqui foi nomeado como “cirrostratus.zabbixcloud.com.br”, fazendo alusão a um tipo de nuvem alta e uniforme.

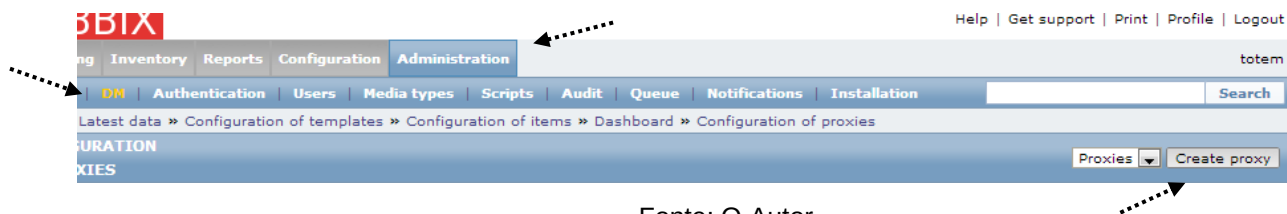
A instalação simplificada do servidor Zabbix foi efetuada seguindo os passos em **Anexo 01: Instalação simplificada do Zabbix Server**.

A instalação do Proxy foi efetuada seguindo os passos em **Anexo 02: Instalação do Zabbix Proxy** com as seguintes observações:

1. Se o Proxy utilizar o banco MySQL em uma máquina que já possua o Zabbix Server, será preciso criar um Banco separado (com usuário separado também);
2. O proxy estando como Ativo, produz mais informações nos *logs*;
3. É preciso verificar quanto tempo (em média), um script leva pra ser executado e traduzir isso na diretiva "Timeout" na configuração do zabbix_proxy.

Depois de instalado o processo Zabbix_proxy, foi preciso configura-lo no Servidor para que o mesmo pudesse ser utilizado como ponte para as requisições. Para tanto, foi preciso acessar o servidor via Navegador WEB e seguir o caminho *Administration -> DM -> Create Proxy*, como pode ser visto na Figura 19.

Figura 19: Adicionando um Proxy para requisições remotas.



Fonte: O Autor

Na tela seguinte, é preciso informar o endereço (IP ou FQDN) do Proxy, como pode ser verificado na Figura 20.

Figura 20: Cadastro de informações do Proxy.

Proxy name:

Proxy mode:

Hosts: Proxy hosts

Other hosts:

- BackBone-CConvivencia
- BackBone-CCS
- BackBone-Enfermagem
- BackBone-Reitoria
- BackBone-SINFO
- dns.ufrn.br
- dns.ufrnet.br
- mail-dns.ufrn.br
- ns1.ufrn.br
- ns2.ufrn.br

Buttons: Save, Clone, Delete, Cancel

Fonte: O Autor

Com o Proxy cadastrado, é possível verificar seu estado em *Administration* -> *DM*. A Figura 21 demonstra o *Status* do Proxy depois de adicionado (com algumas colunas omitidas).

Figura 21: Status do Proxy depois de Adicionado.

Proxies		
Displaying 1 to 1 of 1 found		
☐	Name	Mode
☐	cirrostratus.zabbixcloud.com.br	Active

Fonte: O Autor

3.4.2 Execução

Como se trata de um monitoramento não intrusivo, ou seja, sem a instalação de Agentes nos alvos monitorados, foi preciso adaptar os métodos de coleta de

dados. Para isso, foram criados Scripts em Linux Bash, utilizando ferramentas próprias do Sistema Operacional Linux (*Ubuntu 12.04 LTS*).

Entre as versões macro 1.8 e 2.0 do Zabbix, existem diferenças nos itens suportados. Uma delas é o item “net.udp.service” (do método *simple check*). Este item permite ao servidor Zabbix enviar e receber informações a portas UDP, na tentativa de verificar se elas estão abertas (ouvindo) e se estão trocando informações com os usuários interessados (serviços corretos ativos). Este item não é suportado nas versões 2.0.X, que inclui a versão utilizada neste trabalho (2.0.5).

No cenário do Provedor supracitado, é preciso monitorar uma porta UDP específica: porta 53 do servidor DNS. O monitoramento desta porta é de suma importância, já que é através dela que os clientes consultam o servidor autoritativo para o domínio DNS do provedor EnterpriseWEB, à respeito das Zonas cobertas pelo mesmo. O Script criado para este fim, foi nomeado como “port_check.sh”.

No **Anexo 03: Script “port_check.sh”**, é possível verificar o código fonte do Script. É interessante levantar alguns pontos nele:

1. Ele recebe como parâmetros o HOST, o PROTOCOLO (tcp ou udp) e a PORTA;
2. Faz testes de tipos nestes parâmetros (se válidos ou inválidos);
3. Retorna o estado para portas, informado pelo NMAP.

Outro item de extrema necessidade são as consultas DNS. É preciso saber se o servidor DNS do provedor responde por certos registros dos quais ele tem propriedade e isto não é feito pelos métodos padrão encontrados no Zabbix 2.0.5.

Para isso, foi criado o Script “dig.sh” executado através do método “External check”. O nome do Script copia o nome do utilitário de Linha de comando encontrado em Sistemas *Unix Like* (como o Linux).

Na Figura 22 é possível verificar o script criado para suprir a demanda de monitoramento via consultas DNS.

Figura 22: Script para monitoramento via consultas DNS.

```
#!/bin/bash
# Acolhimento de parâmetros:
# 1 - Servidor DNS
# 2 - Domínio a resolver
SRV_DNS=$1
DOMINIO=$2
TIPO="a"
OPCOES="+short"

#Execução:
RETORNO=`dig @$SRV_DNS $OPCOES -t $TIPO $DOMINIO|egrep -v '[A-Za-z]'`

if [ -z $RETORNO ]
then
    echo "NOT_FOUND"
else
    echo $RETORNO
fi
```

Fonte: O Autor

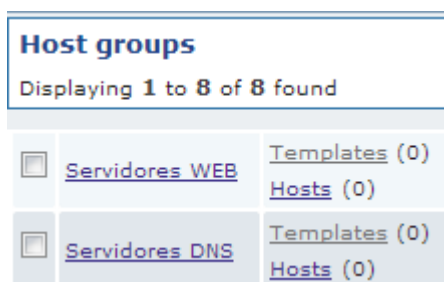
Uma observação é que os Scripts foram colocados no diretório padrão no Proxy: **/usr/local/share/zabbix/externalscripts/**.

3.4.2.1 Criação dos *Host Groups*

É interessante criar Grupos para que os servidores sejam separados por Tipos, como por exemplo, Servidores WEB, Servidores DNS e etc.

Para tanto, foi preciso acessar, na interface WEB do servidor Zabbix, o caminho *Configuration -> Host Groups -> Create host Group*. A Figura 23 mostra os grupos criados para os servidores a serem monitorados.

Figura 23: Grupos criados para os servidores.



Host groups		
Displaying 1 to 8 of 8 found		
☐	Servidores WEB	Templates (0) Hosts (0)
☐	Servidores DNS	Templates (0) Hosts (0)

Fonte: O Autor

Os grupos criados fazem referência ao tipo de servidor a serem monitorado.

3.4.2.2 Criação dos *Templates*

Em se tratando de um Provedor de serviços, vários sistemas servidores atendem uma mesma área. Isto se deve ao fato de que com o crescente número de

clientes, o provedor precisa distribuir ou balancear as cargas de acesso às aplicações de seus clientes.

Para facilitar a configuração de vários Hosts no Zabbix, utilizou-se dos *Templates*. Este tipo de objeto *container* é bastante útil por agrupar checagens comuns a vários Hosts. Isto significa que itens, *Triggers* e gráficos podem ser configurados em somente um local e que todos os Hosts ligados a este *Template* herdem estas configurações.

Templates podem herdar informações de outros *Templates*. Foi criado um *Template* somente para cadastrar *Applications* a fim de disponibiliza-las para os demais *Templates* customizados. Este foi criado como **Custom_Template_Applications_ONLY**. Nele, foram configuradas aplicações genéricas comuns a qualquer servidor: Aplicação, Caminho (Path), Disponibilidade, Geral, Página, Servidor, Serviço(s).

A Figura P mostra o *Template Applications_ONLY*.

Figura 24 - Template "Applications Only"

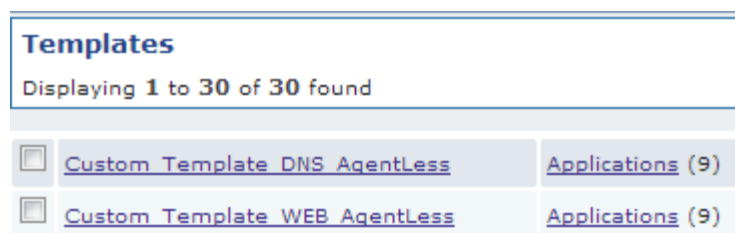


Fonte: O Autor

Os *Templates* foram criados seguindo o caminho *Configuration -> Templates -> Create Template*. Os nomes foram configurados como **Custom_Template_DNS_AgentLess** e **Custom_Template_WEB_AgentLess**.

A Figura 25 mostra os *Templates* depois de criados.

Figura 25: Templates para servidores Monitorados



Fonte: O Autor

3.4.2.3 Cadastro dos Hosts

Para criar os *Hosts* a serem monitorados, foi preciso acessar, através da interface WEB do Servidor Zabbix, o caminho *Configuration -> Hosts -> Create Host*.

Na Figura 26 é possível verificar as informações configuradas no *Host*.

Figura 26: Configuração de Host

The screenshot shows the 'Create Host' form in Zabbix. It includes fields for 'Host name' and 'Visible name', both set to 'Servidor DNS 01'. The 'Groups' section shows 'Servidores DNS' in the 'In groups' list and 'Zabbix servers' in the 'Other groups' list. The 'Agent interfaces' section shows a table with IP address '172.16.0.100', DNS name 'dns.enterpriseweb.local', and port '10050'. The 'Monitored by proxy' field is set to 'cirrostratus.zabbixcloud.com.br' and the 'Status' is set to 'Monitored'.

Host name	Visible name	Groups	Agent interfaces	Monitored by proxy	Status								
Servidor DNS 01	Servidor DNS 01	In groups: Servidores DNS Other groups: Zabbix servers	<table border="1"> <thead> <tr> <th>IP address</th> <th>DNS name</th> <th>Connect to</th> <th>Port</th> </tr> </thead> <tbody> <tr> <td>172.16.0.100</td> <td>dns.enterpriseweb.local</td> <td>IP DNS</td> <td>10050</td> </tr> </tbody> </table>	IP address	DNS name	Connect to	Port	172.16.0.100	dns.enterpriseweb.local	IP DNS	10050	cirrostratus.zabbixcloud.com.br	Monitored
IP address	DNS name	Connect to	Port										
172.16.0.100	dns.enterpriseweb.local	IP DNS	10050										

Fonte: O Autor

No contexto de monitoramento via Proxy, é preciso configurar os Hosts com uma informação adicional: *Monitored by Proxy*. Nesta configuração foi informado o Proxy usado para monitorar este Host. É possível notar que este Proxy é o mesmo que foi configurado anteriormente (*cirrostratus.zabbixcloud.com.br*).

Para fins de prova de conceito, o monitoramento foi composto por:

- Dois servidores WEB;
- Dois servidores DNS.

3.4.2.4 Itens monitorados

Utilizando-se dos *Templates*, foram configurados os itens conforme a Tabela 4. Para o *Template* Servidores DNS os itens cadastrados estão nas Figura 27, Figura 28 e Figura 29.

Figura 27: Primeiro item cadastrado para Template DNS

Host	Custom_Template_DNS_AgentLess								
Name	SCRIPT Estado Porta DNS (UDP-53)								
Type	External check								
Key	port_check.sh[{HOST.CONN},udp,53]		Select						
Type of information	Text								
Update interval (in sec)	15								
Flexible intervals	<table border="1"> <thead> <tr> <th>Interval</th> <th>Period</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td colspan="3">No flexible intervals defined.</td> </tr> </tbody> </table>			Interval	Period	Action	No flexible intervals defined.		
Interval	Period	Action							
No flexible intervals defined.									
New flexible interval	Interval (in sec)	50	Period 1-7,00:00-24:00 Add						
Keep history (in days)	15								
New application									
Applications	-None- Aplicação Caminho (Path) Disponibilidade DNS Geral								
Populates host inventory field	-None-								
Description	Verificação do Estado da porta UDP 53.								
Status	Enabled								

Fonte: O Autor

O primeiro item verifica se a porta UDP/53 (padrão para o serviço DNS), está aceitando solicitações (se está aberta).

Figura 28: Segundo item cadastrado para Template DNS

Host	Custom_Template_DNS_AgentLess								
Name	Resposta externa para domínio próprio (\$2)								
Type	External check								
Key	dig.sh[{HOST.CONN}],www.enterpriseweb.local	Select							
Type of information	Text								
Update interval (in sec)	15								
Flexible intervals	<table border="1"> <thead> <tr> <th>Interval</th> <th>Period</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td colspan="3">No flexible intervals defined.</td> </tr> </tbody> </table>			Interval	Period	Action	No flexible intervals defined.		
Interval	Period	Action							
No flexible intervals defined.									
New flexible interval	Interval (in sec)	50	Period 1-7,00:00-24:00 Add						
Keep history (in days)	15								
New application									
Applications	-None- - Aplicação - Caminho (Path) - Disponibilidade - DNS - Geral								
Populates host inventory field	-None-								
Description	Resposta permitida para domínios de propriedade (requerido). Teste a capacidade do servidor DNS responder a requisições para os domínios dos quais ele tem propriedade.								
Status	Disabled								

Fonte: O Autor

O segundo item checa a capacidade de o servidor DNS do Provedor responder por consultas a um domínio de sua propriedade.

Figura 29: Terceiro item cadastrado para Template DNS

Host	Custom_Template_DNS_AgentLess								
Name	Resposta externa para outros domínios (\$2)								
Type	External check								
Key	dig.sh[{{HOST.CONN}},google.com]	Select							
Type of information	Text								
Update interval (in sec)	15								
Flexible intervals	<table border="1"> <thead> <tr> <th>Interval</th> <th>Period</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td colspan="3">No flexible intervals defined.</td> </tr> </tbody> </table>			Interval	Period	Action	No flexible intervals defined.		
Interval	Period	Action							
No flexible intervals defined.									
New flexible interval	Interval (in sec)	50	Period 1-7,00:00-24:00						
	Add								
Keep history (in days)	15								
New application									
Applications	Aplicação Caminho (Path) Disponibilidade DNS Geral								
Populates host inventory field	-None-								
Description	Resposta recursiva para domínios de não propriedade (Perigo). Teste a capacidade do servidor DNS responder a requisições para os domínios dos quais ele não tem propriedade. PERIGO: DNSs recursivos abertos, podem ser um problema muito grande.								
Status	Enabled								

Fonte: O Autor

O terceiro item é mais voltado à segurança. Ele verifica se externamente, o servidor DNS está respondendo a consultas relacionadas a outros domínios. Caso o servidor responda, representa um risco pois estará agindo de forma recursiva aberta, assim disponível a fazer parte de ataques ou mesmo servir a outros que não sejam os clientes do provedor.

Para o *Template* Servidores WEB os itens cadastrados estão na Figura 30.

Figura 30: Itens cadastrados para Template WEB

Host: Custom_Template_WEB_AgentLess

Name: Estado Porta HTTP (TCP-80)

Type: Simple check

Key: net.tcp.service[http,,] [Select](#)

Type of information: Numeric (unsigned)

Data type: Boolean

Update interval (in sec): 15

Flexible intervals:

Interval	Period	Action
No flexible intervals defined.		

New flexible interval: Interval (in sec) 50 Period 1-7,00:00-24:00 [Add](#)

Keep history (in days): 15

Keep trends (in days): 730

Show value: Service state [show value mappings](#)

New application:

Applications:

- None-
- Aplicação
- Caminho (Path)
- Disponibilidade
- Geral
- HTTP

Populates host inventory field: -None-

Description: Verificação da porta HTTP (TCP-80).

Status: Enabled

Fonte: O Autor

A Figura 30 verifica a porta padrão para o protocolo HTTP. O segundo item tem a mesma base que o primeiro, porém trocando o parâmetro de “http” para “https”. Ficando conforme a Figura 31.

Figura 31: Itens HTTP/HTTPS para o Template WEB

Template: [Custom_Template_WEB_AgentLess](#) [Applications](#) (9) [Items](#) (2) [Triggers](#) (0) [Graphs](#) (0) [Screens](#) (0) [Discovery rules](#) (0)

Name	Triggers	Key
Estado Porta HTTP (TCP-80)		net.tcp.service[http,,]
Estado Porta HTTPS (TCP-443)		net.tcp.service[https,,]

Fonte: O Autor

Para o monitoramento complementar da Tabela 4 foram necessários itens de monitoramento do tipo *Web Monitoring (Web Scenarios)*. Nas versões 2.0.x do Zabbix, os *Web Scenarios* não podem pertencer a *Templates*, ou seja, não podem

ser herdados e configurados como os itens dos subtópicos anteriores. Assim, para cada Host no Grupo “Servidores WEB”, foi preciso configurar um *Web Scenario* próprio. Estes itens foram cadastrados através do Menu *Configuration - > Web*.

A criação dos *Web Scenarios* busca os Hosts em seus grupos. Neste caso, foi selecionado o Grupo “Servidores WEB” e um de seus Hosts, como visto na Figura 32.

Figura 32: Escolha do servidor para o Web Scenario

Create scenario (select host first)

Group: Servidores WEB Host: all

all
Servidor WEB 01
Servidor WEB 02

Status

Enabled

Connected as 'werneck'

Fonte: O Autor

Para criar um novo *Web Scenario* foi preciso informar, na aba *Scenario*, os dados conforme a Figura 33.

Figura 33: Criação de Web Scenario – Aba Scenario

Application: Aplicação Select

Name: Homepage, pesquisa e logon

Authentication: None

Update interval (in sec): 20

Agent: Mozilla Firefox 8.0

Variables: {usuario}=teste
{senha}=senha123

Active: ☒

Save Clone Delete Cancel

Fonte: O Autor

A aba *Steps* é composta de uma sequência de passos, para validar o funcionamento de um Site (portal, blog, e-commerce e etc), como se fosse um cliente que navega. Ali, foram configurados os passos vistos na Figura 34.

Figura 34: Criação de Web Scenarios – Aba Steps

Steps	Name	Timeout	URL	Required	Status codes	Action
1:	Homepage	30 sec	http://www.entrepriseweb.local/html/index.php	.*organograma.*	200	Remove
2:	Pesquisa	30 sec	http://www.entrepriseweb.local/html/conteudo/cooperacoes/index.php	.*regulamentada	200	Remove
3:	Home Webmail	30 sec	https://www.entrepriseweb.local/webmail/src/login.php	Desenvolvimento	200	Remove
4:	Acesso sem senha	30 sec	https://www.entrepriseweb.local/webmail/src/redirect.php	ERRO	200	Remove
5:	Logon webmail	30 sec	https://www.entrepriseweb.local/webmail/src/redirect.php		302,200	Remove
6:	Verifica se Logado	30 sec	https://www.entrepriseweb.local/webmail/src/left_main.php	Pastas	200	Remove
7:	Logout Webmail	30 sec	https://www.entrepriseweb.local/webmail/src/signout.php	.*saiu.*	200	Remove
Add						

Fonte: O Autor

O item que merece atenção é o “Logon webmail”. Ele leva algumas configurações adicionais para que seja possível realmente se registrar no webmail. Estas informações podem ser vistas na Figura 35.

Figura 35: Dados adicionais para WebMail

Name	Logon webmail
URL	https://www.entrepriseweb.local/webmail/src/redirect.php
Post	login_username={usuario}&secretkey={senha}&submit=login
Timeout	30
Required string	
Required status codes	302,200

Fonte: O Autor

Os sete passos montados para o *Web Scenario* executam:

1. Acesso à *Homepage* do portal;
2. Acesso a uma página secundária (pesquisa em banco de dados);
3. Acesso à página do Webmail;
4. Tentativa de acesso a uma página interna do Webmail sem se registrar (aqui a resposta demanda é “ERRO”);
5. Logon real no Webmail;

6. Acesso a uma página interna estando logado;
7. Logout do webmail.

Em resumo, neste *Web Scenario* estão implícitos serviços como E-mail e Banco de dados. O acesso a certas partes do *frontend web* só será feito com sucesso se estes sistemas de *Backend* estiverem respondendo normalmente.

3.4.3 Testes e itens visuais

Ao finalizar a configuração dos servidores, foi possível verificar os dados colhidos pelos itens, através do Menu *Monitoring -> Lastest data*. Selecionado o Servidor, os dados aparecem em lista, como pode ser visto nas Figura 36 e Figura 37.

Figura 36: Dados obtidos pelos itens monitorados (WEB).

 <u>Name</u>	<u>Last check</u>	<u>Last value</u>
 HTTP (1 Item)		
Estado Porta HTTP (TCP-80)	18 Jun 2013 21:53:22	Up (1)
 HTTPS (1 Item)		
Estado Porta HTTPS (TCP-443)	18 Jun 2013 21:53:24	Up (1)
 Serviço(s) (2 Items)		
Estado Porta HTTP (TCP-80)	18 Jun 2013 21:53:22	Up (1)
Estado Porta HTTPS (TCP-443)	18 Jun 2013 21:53:24	Up (1)

Fonte: O Autor

É importante lembrar que a coluna *Last Value* na Figura 36 poderia mostrar *Down* ao invés de *Up* em duas situações:

- o servidor HTTP não ter sido iniciado (ou ter parado);
- existência de erros nas configurações do serviço;
- o Firewall ou outro equipamento de proteção na Borda, não repassar as conexões para o servidor HTTP interno.

Figura 37: Dados obtidos pelos itens monitorados (DNS)

☐ <u>Name</u>	<u>Last check</u>	<u>Last value</u>
☐ Aplicação (2 Items)		
Resposta externa para domínio próprio (enterpriseweb.local)	18 Jun 2013 22:03:05	172.16.0.101
Resposta externa para outros domínios (google.com)	18 Jun 2013 22:02:54	NOT_FOUND
☐ Disponibilidade (3 Items)		
SCRIPT Estado Porta DNS (UDP-53)	18 Jun 2013 22:02:54	open

Fonte: O Autor

Na Figura 37, o item SCRIPT Estado Porta DNS (UDP-53) informa o valor “Open”. Caso o valor fosse diferente deste, denotaria um problema no servidor DNS ou mesmo em equipamentos da Borda da rede, como comentado para o Servidor WEB.

Para a Figura 37, a coluna *Last value* mostra um endereço IP para o item “Resposta externa para domínio próprio (enterprisewab.local)”. Este IP é a prova de que o servidor DNS está respondendo positivamente por um domínio do qual é responsável. Caso a resposta fosse “NOT_FOUND”, denotaria um problema referente ao DNS, ligado aos motivos demonstrados anteriormente para o servidor WEB.

Além dos dados confirmados pelas Figura 36 e Figura 37, também foi possível acompanhar os dados colhidos pelo *Web Scenario*, já com gráficos montados automaticamente.

O monitoramento com *Web Scenarios* possibilita a verificação de:

- Velocidade de acesso à páginas Web;
- Tempo de resposta pra este acesso;
- Código de resposta HTTP¹;
- E status final baseado na String esperada.

A Figura 38 demonstra o resumo da verificação, após a sequência de passos do *Web Scenario*.

¹ [Códigos de Resposta HTTP](#)

Figura 38: Resumo da verificação via Web Scenario – Servidor WEB 01

DETAILS OF SCENARIO Homepage pesquisa e logon [19 Jun 2013 20:17:02]				
Step	Speed	Response time	Response code	Status
Homepage	11.04 KBps	1s 130ms	200	OK
Pesquisa	31.93 KBps	338ms	200	OK
Home Webmail	1.27 KBps	1s 859ms	200	OK
Acesso sem senha	3.84 KBps	323ms	200	OK
Logon webmail	510 Bps	648ms	200	OK
Verifica se Logado	6.73 KBps	345ms	200	OK
Logout Webmail	2.35 KBps	314ms	200	OK
TOTAL		4s 956ms		OK

Fonte: O Autor

O *Web Scenario* também produz gráficos que podem ser visualizados sob o tempo, dos dados resumidos na Figura 38. Estes gráficos podem ser vistos através das Figura 39 e Figura 40.

Figura 39: Velocidade de Download - Web Scenario

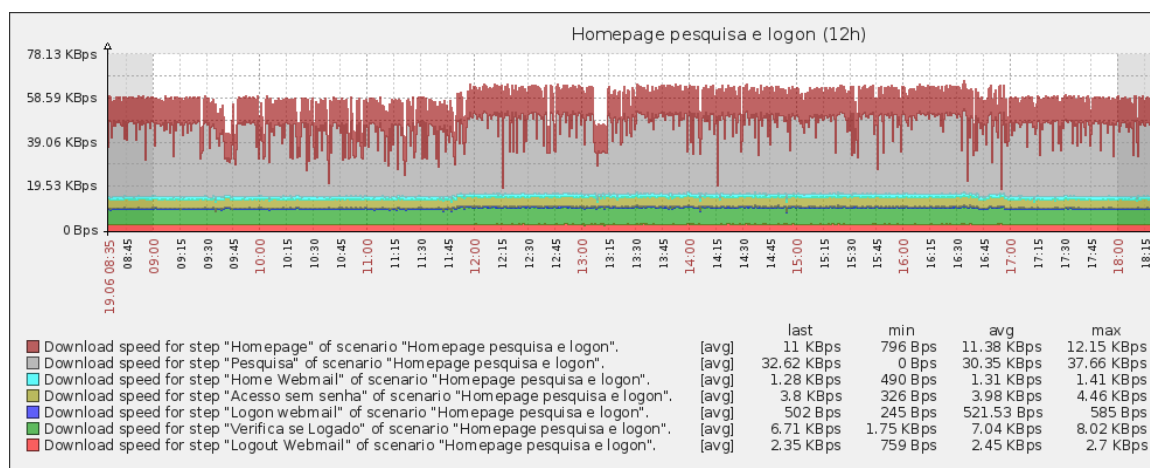
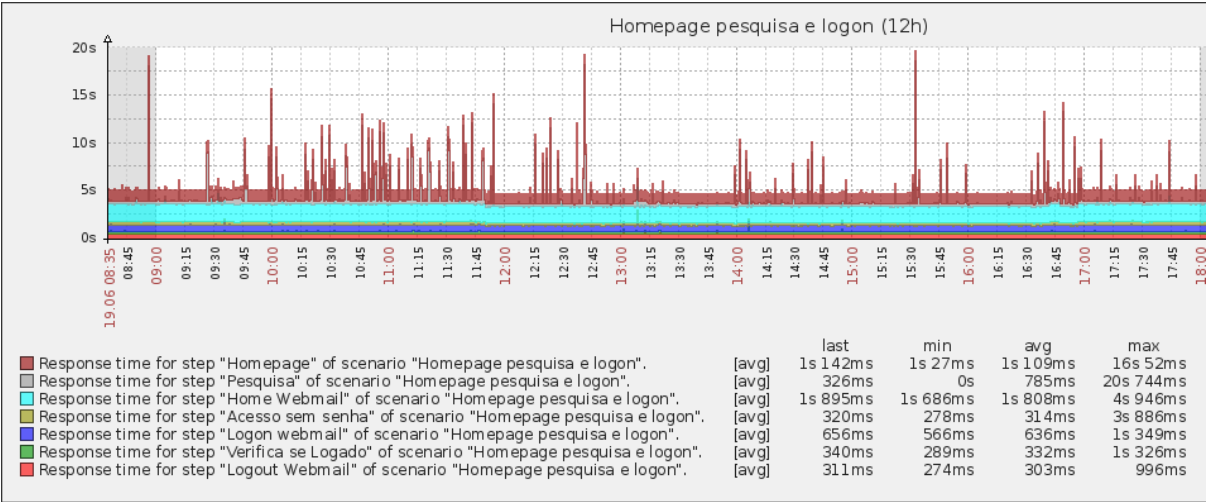


Figura 40: Tempo para Download - Web Scenario



4 CONCLUSÃO

O monitoramento de ativos e sistemas de rede é um item indispensável, porém na maior parte das vezes, pode ser complementado de modo a melhorá-lo.

A abordagem de monitoramento que foi proposta neste trabalho é complementar à atualmente praticada, pois alinha a visão interna (do suporte, por exemplo), à visão externa que é, geralmente, a dos verdadeiros usuários.

Além disto, esta abordagem permite averiguar se a ocorrência de certo problema está ligada diretamente ao destino final do acesso (um servidor WEB, por exemplo), ou a componentes que estejam entre o usuário externo e seu destino final (como Firewalls e outros componentes de segurança perimetral).

Utilizando as técnicas de monitoramento por porta e *Scripts*, é possível estender o monitoramento a Servidores e Sistemas com os mais variados fins, além disto é possível mesclar os itens verificados internamente com externos via Proxy, afim de unificar estas visões.

5 CONSIDERAÇÕES FINAIS

Configuração de itens como *Mapas*, *Triggers* e *Actions*, foram deixados de fora deste trabalho, pois o intuito real era demonstrar uma abordagem complementar ao monitoramento praticado atualmente.

É necessário comentar que foram encontrados alguns problemas na execução deste trabalho, os quais seguem à baixo:

- *Bug*: Ao cadastrar um *Web Scenario*, seu nome (*Name*) não pode conter vírgulas, pois os resultados não saem com as descrições corretas. Por exemplo: foi cadastrado inicialmente o nome “Homepage, pesquisa e logon” mas precisou-se retirar a vírgula, ficando “Homepage pesquisa e logon”;
- Se for utilizada a instância AWS t1.micro (plano gratuito) com execução de *Scripts* (como adotado neste trabalho), será preciso aumentar, no arquivo de configuração do Proxy, o parâmetro *Timeout* para valores próximos de seu máximo. Cada execução de Script pode levar certo tempo e se o Servidor não receber dados provenientes destas checagens, causará os famosos “vales” nos gráficos. O valor final, para este trabalho, ficou em 20.

Os arquivos utilizados neste trabalho, podem ser obtidos através do Link <http://goo.gl/L0yju>.

REFERENCIAL BIBLIOGRÁFICO

- Amazon AWS - EBS. (s.d.). **Amazon Elastic Block Store (EBS)**. Acesso em 14 de Junho de 2013, disponível em Amazon AWS: <http://aws.amazon.com/pt/ebs/>
- Amazon AWS - Free. (s.d.). **Nível de uso gratuito da AWS**. Acesso em 14 de Junho de 2013, disponível em Amazon - AWS: <http://aws.amazon.com/pt/free/>
- Amazon AWS - Free Tier (faq). (s.d.). **Amazon AWS Free - faq**. Acesso em 10 de Março de 2013, disponível em Perguntas frequentes do Nível de uso gratuito da AWS: <http://aws.amazon.com/pt/free/faqs/>
- Amazon AWS - Instance Types. (s.d.). **Instâncias do Amazon EC2**. Acesso em 14 de Junho de 2013, disponível em Amazon AWS: <http://aws.amazon.com/pt/ec2/instance-types/#instance-details>
- Amazon AWS - Management Console. (s.d.). **AWS Management Console**. Acesso em 10 de Janeiro de 2013, disponível em Amazon AWS: <http://aws.amazon.com/pt/console/>
- Amazon AWS - **Pricing Overview**. (Março de 2012). Acesso em 18 de Julho de 2013, disponível em Amazon AWS - Pricing: http://media.amazonwebservices.com/AWS_Pricing_Overview.pdf
- Amazon AWS. (10 de Janeiro de 2013). **What is Amazon Web Services?** Acesso em 10 de Abril de 2013, disponível em AWS Documents: <http://docs.aws.amazon.com/gettingstarted/latest/awsgsg-intro/intro.html>
- Amazon AWS. (s.d.). **Amazon Relational Database Service (Amazon RDS)**. Acesso em 14 de Junho de 2013, disponível em Amazon RDS: <http://aws.amazon.com/pt/rds/>
- Arregoces, M., & Portolani, M. (2004). **Data center Fundamentals**. Indianápolis: Cisco Press.
- Black, T. L. (Dezembro de 2008). **Comparação de Ferramentas de Gerência de Redes**. Acesso em 18 de Maio de 2013, disponível em <http://www.lume.ufrgs.br/bitstream/handle/10183/15986/000695315.pdf>
- BONOMO, E. (30 de Abril de 2006). **GERENCIAMENTO E MONITORAÇÃO DE REDES DE COMPUTADORES UTILIZANDO-SE ZABBIX**. Acesso em 10 de Maio de 2013, disponível em <http://www.ginix.ufla.br/files/mono-EsleyBonomo.pdf>

- Comer, D. E. (2001). **Redes de Computadores e Internet Traduzida de "Computer Networks and Internets with Internet" (2ª Edição ed.)**. Porto Alegre: Bookman.
- Free Software Foundation. (Junho de 1992). **GNU - General Public Licence Version 2**. Acesso em 10 de Janeiro de 2013, disponível em GPLv2: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
- Josyula, V., Orr, M., & Page, G. (2012). **Cloud Computing - Automating the Virtualized Data Center**. Indianapolis: Cisco Press.
- Majewski, R. (Novembro de 2009). **Sistema de Monitoração de Rede**. Acesso em 10 de Fevereiro de 2013, disponível em Programa de Pós-Graduação em Informática (PUCPR): <http://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08A/Roberto%20Majewski%20-%20Artigo.pdf>
- Mauro, D. R., & Schmidt, K. J. (2005). **Essential SNMP (2ª Edição ed.)**. O'Reilly.
- Nicolletti, S. P., Lopes, R. V., & Sauvé, P. J. (2003). **Melhores Práticas para Gerência de Redes de Computadores**. Rio de Janeiro: Campus.
- NIST. (2012, Maio 15). **The NIST Definition of Cloud Computing**. Acesso em Junho 15, 2013, from National Institute Of Standards and Technology: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- Olups, R. (2010). **Zabbix 1.8 Network Monitoring**. MUMBAI (Maharashtra/Índia): Packt Publishing.
- PUSKA, A. A. (2011). **Solução de Gerenciamento de Redes Utilizando o Sistema de Código Aberto: Zabbix**. Acesso em 18 de Maio de 2013, disponível em http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/395/1/CT_GESER_1_2011_04.pdf
- RFC1157. (Maio de 1990). **Network Working Group**. Acesso em 09 de Janeiro de 2012, disponível em <http://tools.ietf.org/html/rfc1157>
- RFC2570-Sec. (Abril de 1999). **RFC 2570 - Seção Segurança**. Acesso em 18 de Maio de 2013, disponível em <http://tools.ietf.org/html/rfc2570#section-3.4>
- RNP. (01 de Agosto de 1997). **RNP NewsGeneration**. Acesso em 07 de Fevereiro de 2012, disponível em <http://www.rnp.br/newsgen/9708/n3-2.html>
- SNMP Research INC. (n.d.). **SNMP RFCs**. Acesso em Maio 18, 2013, from http://www.snmp.com/protocol/snmp_rfcs.shtml

- Varia, J., & Mathew, S. (13 de Junho de 2013). **Amazon AWS Whitepaper**. Fonte: Overview of Amazon AWS: http://d36cz9buwru1tt.cloudfront.net/AWS_Overview.pdf
- Veras, M. (2013). **Arquitetura Corporativa de Nuvem: Amazon WEB Services (AWS)**. Rio de Janeiro: Brasport.
- Wikipedia INC. (2013, Abril 24). **Zabbix**. Acesso em Maio 10, 2013, from Wikipedia.org: <http://en.wikipedia.org/wiki/Zabbix>
- Zabbix SIA - About. (2012, Dezembro 10). **About Zabbix - What is Zabbix**. Acesso em Janeiro 10, 2013, from Wiki Zabbix: <https://www.zabbix.com/documentation/2.0/manual/introduction/about>
- Zabbix SIA - Distributed Monitoring. (28 de Setembro de 2012). **Distributed Monitoring - Proxies**. Acesso em 27 de Maio de 2013, disponível em Zabbix Wiki: https://www.zabbix.com/documentation/2.0/manual/distributed_monitoring/proxies
- Zabbix SIA - Features. (2011, Dezembro 28). **Zabbix Features**. Acesso em Janeiro 10, 2013, from Zabbix Wiki: <https://www.zabbix.com/documentation/2.0/manual/introduction/features>
- Zabbix SIA - Homepage. (10 de Julho de 2013). **Homepage**. Acesso em 15 de Julho de 2013, disponível em Zabbix SIA: <http://www.zabbix.com/>
- Zabbix SIA - Instalation. (2012, Fevereiro 20). **Zabbix 2.0 Instalation Manual**. Acesso em Abril 10, 2013, from Wiki Zabbix: <https://www.zabbix.com/documentation/2.0/manual/installation>
- Zabbix SIA - Licence. (2001-2013). **Zabbix SIA**. Acesso em Julho 13, 2013, from Zabbix.com: <http://www.zabbix.com/license.php>
- Zabbix SIA - News. (2004-2013). **Zabbix News**. Acesso em Junho 12, 2013, from Zabbix.com: <http://www.zabbix.com/news.php>
- Zabbix SIA - Overview. (2012, Fevereiro 4). **Zabbix SIA - Documentation**. Acesso em Janeiro 19, 2013, from Zabbix Overview: <https://www.zabbix.com/documentation/2.0/manual/introduction/overview>
- Zabbix SIA - Presentation. (10 de Dezembro de 2012). **Presentation**. Acesso em 10 de Março de 2013, disponível em Zabbix SIA - Homepage: <http://www.zabbix.com/presentation.php>

Zabbix SIA - Visualisation (Screens). (2013, Maio 28). **Zabbix Screens**. Acesso em Junho 16, 2013, from Zabbix Wiki: <https://www.zabbix.com/documentation/2.0/manual/config/visualisation/screens>

Zabbix SIA - Visualizations. (n.d.). **Zabbix SIA - Visualizations**. Acesso em Maio 24, 2013, from Wiki Zabbix: <https://www.zabbix.com/documentation/2.0/manual/config/visualisation>

Anexo 01: Instalação simplificada do Zabbix Server

OBS: Os comandos nas caixas de texto devem ser executados no Terminal Linux, tendo o usuário os poderes de *root*.

Criar um diretório temporário para os arquivos a serem extraídos e extraí-los:

```
cd /usr/src  
mkdir zabbix  
cd ./zabbix
```

Baixar o código fonte da página de Downloads do Zabbix:

```
wget http://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/2.0.5/zabbix-2.0.5.tar.gz/download -O zabbix-2.0.5.tar.gz
```

Extrair os arquivos:

```
tar xzv zabbix-2.0.5.tar.gz
```

Instalação de Pré-requisitos (dependências):

```
apt-get install build-essential apache2 php5 libapache2-mod-php5 php5-gd mysql-server libopenipmi0 fping libcurl3-gnutls snmp libmysqlclient-dev libcurl4-gnutls-dev libsnmp-dev libopenipmi-dev php5-mysql
```

Criação de grupo/usuário (com poderes mínimos):

```
addgroup --system zabbix  
adduser --system --ingroup zabbix --no-create-home zabbix
```

Criação da base de dados, usuário e permissões de acesso:

```
mysql -uroot -p (será solicitada a senha do root)  
mysql> CREATE DATABASE zabbix;  
mysql> CREATE USER 'zabbix_usr'@'localhost' IDENTIFIED BY 'senha';  
mysql> GRANT ALL ON zabbix.* TO 'zabbix'@'localhost';  
mysql> FLUSH PRIVILEGES;
```

Compilação do zabbix:

```
./configure --enable-server --enable-agent --with-mysql --with-net-snmp --with-libcurl --with-openipmi
```

Instalação do Zabbix:

```
make install
```

Inserção de dados no BD: Navegar até o diretório `/usr/src/zabbix/zabbix-2.0.5/database/mysql`, e executar:

```
mysql -uzabbix_usr -p zabbix < schema.sql
mysql -uzabbix_usr -p zabbix < images.sql
mysql -uzabbix_usr -p zabbix < data.sql
```

Criação de diretórios essenciais à instalação:

```
mkdir -p /var/{log,run}/zabbix
chown zabbix:zabbix /var/{log,run}/zabbix
```

Configurações do **`/usr/local/etc/zabbix_server.conf`**:

```
LogFile=/var/log/zabbix/server.log
PidFile=/var/run/zabbix/server.pid
DBHost=localhost
DBName=zabbix
DBUser=zabbix_usr
DBPassword=senha
FpingLocation=/usr/bin/fping
```

Configurações do **`/usr/local/etc/zabbix_agentd.conf`**:

```
LogFile=/var/log/zabbix/agentd.log
PidFile=/var/run/zabbix/agentd.pid
Server=127.0.0.1
```

Setar permissões no arquivo **`/usr/bin/fping`**:

```
chown root:zabbix /usr/bin/fping
chmod 4710 /usr/bin/fping
```

Copiar os arquivos de inicialização do `zabbix_server` e `zabbix_agentd` para `/etc/init.d`: no diretório `/misc/init.d/debian`, copiar os dois scripts (`zabbix-agent` e `zabbix-server`). Alterar as configurações dos scripts (*PID* e nome do *Daemon*), conforme configurações feitas nos arquivos **`.conf`**.

Iniciar o Zabbix junto com o sistema:

```
update-rc.d zabbix-server defaults
update-rc.d zabbix-server enable
```

Copiar os arquivos da interface WEB:

```
Cp -Rp /usr/src/zabbix/zabbix-2.0.5/frontends /var/www/zabbix
```

Acessar a máquina via `http://endereço-ip/zabbix` e seguir os passos da instalação (alterando o que for necessário no `php.ini`).

No fim, é solicitado que um arquivo de configuração seja criado com os dados de BD, host e etc. Em geral, o usuário do Processo Apache (`www-data`) não tem permissão de escrita no diretório `/zabbix`. Será preciso fazê-lo manualmente. Pode ser de duas formas: ou baixando para uma máquina intermediária e subindo para o servidor, ou baixando, copiando seu conteúdo e colando no arquivo correto no servidor.

Ao finalizar, iniciar o servidor:

```
/etc/init.d/zabbix-server start
```

Para ter certeza que deu certo, execute um `'ps aux'` e verifique se existem processos nominados `'zabbix_server'`. Checar também os logs em `/var/log/zabbix/server.log`, verificando se são demonstrados os passos para subir o servidor, semelhante ao que segue à baixo:

```
2494:20130610:165944.604 Starting Zabbix Server. Zabbix 2.0.5 (revision 33558).
```

```
2494:20130610:165944.605 ***** Enabled features *****
```

```
2494:20130610:165944.605 SNMP monitoring:      NO
```

```
2494:20130610:165944.605 IPMI monitoring:      NO
```

```
2494:20130610:165944.605 WEB monitoring:      YES
```

```
2494:20130610:165944.605 Jabber notifications: NO
```

```
2494:20130610:165944.605 Ez Texting notifications: YES
```

```
2494:20130610:165944.605 ODBC:                NO
```

```
2494:20130610:165944.605 SSH2 support:        NO
```

```
2494:20130610:165944.605 IPv6 support:        NO
```

```
2494:20130610:165944.605 *****
```

```
2505:20130610:165944.767 server #2 started [db watchdog #1]
```

```
2504:20130610:165944.768 server #1 started [configuration syncer #1]
```

```
2506:20130610:165944.769 server #3 started [poller #1]
```

```
2507:20130610:165944.771 server #4 started [poller #2]
```

```
2508:20130610:165944.775 server #5 started [poller #3]
```

```
2509:20130610:165944.777 server #6 started [poller #4]
```

```
2510:20130610:165944.782 server #7 started [poller #5]
```

Anexo 02: Instalação do Zabbix Proxy

OBS: Os comandos nas caixas de texto devem ser executados no Terminal Linux, tendo o usuário os poderes de *root*.

Criar um diretório temporário para os arquivos a serem extraídos e extraí-los:

```
cd /usr/src  
mkdir zabbix  
cd ./zabbix
```

Baixar o código fonte da página de Downloads do Zabbix:

```
wget http://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/2.0.5/zabbix-2.0.5.tar.gz/download -O zabbix-2.0.5.tar.gz
```

Extrair os arquivos:

```
tar xzv zabbix-2.0.5.tar.gz
```

Instalação de Pré-requisitos (dependências):

```
apt-get install build-essential libmysqlclient-dev libcurl4-gnutls-dev
```

Criação de grupo/usuário (com poderes mínimos):

```
addgroup --system zabbix  
adduser --system --ingroup zabbix --no-create-home zabbix
```

Criação da base de dados, usuário e permissões de acesso:

```
Criar instância RDS, setar permissões e executar sequência:  
mysql -h "host RDS" -uroot -p (será solicitada a senha do root)  
mysql> CREATE DATABASE zabbix_proxy;  
mysql> CREATE USER ' zabbix_prx '@ip do proxy' IDENTIFIED BY 'senha';  
mysql> GRANT ALL ON zabbix.* TO ' zabbix_prx '@ip do proxy';  
mysql> FLUSH PRIVILEGES;
```

Compilação do zabbix:

```
./configure --enable-proxy --with-mysql --with-libcurl
```

Instalação do Zabbix:

```
make install
```

Inserção de dados no BD: Navegar até o diretório `/usr/src/zabbix/zabbix-2.0.5/database/mysql`, e executar:

```
mysql -h "host RDS" -uzabbix_usr -p zabbix < schema.sql
```

Criação de diretórios essenciais à instalação:

```
mkdir -p /var/{log,run}/zabbix
chown zabbix:zabbix /var/{log,run}/zabbix
```

Configurações do **`/usr/local/etc/zabbix_proxy.conf`**:

```
ProxyMode=0
Server="Servidor zabbix"
ServerPort=10051
Hostname="Nome deste Proxy Zabbix"
LogFile=/var/log/zabbix/proxy.log
LogFileSize=15
PidFile=/var/run/zabbix/proxy.pid
DBHost="Host Amazon RDS"
DBName=zabbix_proxy
DBUser=zabbix_prx
DBPassword="Senha"
ProxyLocalBuffer=1
ProxyOfflineBuffer=2
ConfigFrequency=180
StartPingers=2
FpingLocation=/usr/bin/fping
Timeout=20
```

Setar permissões no arquivo **`/usr/bin/fping`**:

```
chown root:zabbix /usr/bin/fping
chmod 4710 /usr/bin/fping
```

Copiar o arquivo de inicialização do `zabbix_proxy` para `/etc/init.d`: no diretório `/usr/src/zabbix/zabbix-2.0.5/misc/init.d/debian`, copiar o script `zabbix-server` e alterar as configurações das variáveis para as seguintes:

```
NAME=zabbix_proxy
DAEMON=/usr/local/sbin/${NAME}
DESC="Zabbix Proxy daemon"
PID=/var/run/zabbix/proxy.pid
```

Iniciar o Zabbix junto com o sistema:

```
update-rc.d zabbix-proxy defaults
update-rc.d zabbix-proxy enable
```

Ao finalizar, iniciar o servidor:

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

Para ter certeza que deu certo, execute um 'ps aux' e verifique se existem processos nominados 'zabbix_proxy'. Verificar também os logs em /var/log/zabbix/proxy.log, verificando se são demonstrados os passos para subir o Proxy, semelhante ao que seguem à baixo:

```
10965:20130514:222253.925 Starting Zabbix Proxy (active) [nome do proxy].
Zabbix 2.0.5 (revision 33558).
10965:20130514:222253.925 **** Enabled features ****
10965:20130514:222253.925 SNMP monitoring:      NO
10965:20130514:222253.926 IPMI monitoring:      NO
10965:20130514:222253.926 WEB monitoring:      YES
10965:20130514:222253.926 ODBC:              NO
10965:20130514:222253.926 SSH2 support:       NO
10965:20130514:222253.926 IPv6 support:       NO
10965:20130514:222253.926 *****
10965:20130514:222253.926 In init_database_cache()
```

Anexo 03: Script “port_check.sh”

```
#!/bin/bash
# Script simples para checagem de portas TCP/UDP
# Parametros:
# HOST: Host a ser verificado. - # PROTO: Protocolo a ser utilizado na verificação - # PORTA: Porta a
ser verificada - # OP_PROTO: Será populado com a opção referente ao protocolo detectado
# PARAMS: Lista (customizável) de parâmetros extras passados ao NMAP
HOST=$1
PROTO=$2
PORTA=$3
OP_PROTO=
PARAMS='-PN'
saniti(){
    if [ -z "$HOST" ]
    then
        echo "ZBX_NOTSUPPORT_HOST"
        exit 100
    fi
    if [ -n "$PROTO" ]
    then
        case "$PROTO" in
            "UDP"|"udp")
                OP_PROTO='-sU'
                ;;
            "TCP"|"tcp")
                OP_PROTO='-sT'
                ;;
            *)
                echo "ZBX_NOTSUPPORT_PROTO"
                exit 101
                ;;
        esac
    else
        echo "ZBX_NOTSUPPORT_PROTO"
    fi
    if [ -z "$PORTA" ]
    then
        echo "ZBX_NOTSUPPORT_PORTA_VAZIO"
        exit 102
    else
        if [[ $PORTA =~ ^[0-9]{1,5}$ ]]
        then
            PORTA=$PORTA
        else
            echo "ZBX_NOTSUPPORT_PORTA_INCORRETA"
            exit 103
        fi
    fi
}
check_port(){
    saniti
    RESULT=`sudo nmap $PARAMS $OP_PROTO -p $PORTA $HOST|egrep '^[0-9]{1,5}.'|cut -d ' ' -f 2`
    echo $RESULT
}
check_port
```