v7.n12.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

MONITORAMENTO DE INFRAESTRUTURA DE TI COM ZABBIX E GRAFANA: UM ESTUDO DE CASO EM AMBIENTE VIRTUALIZADO NA AWS

IT INFRASTRUCTURE MONITORING WITH ZABBIX AND GRAFANA: A CASE STUDY IN A VIRTUALIZED AWS ENVIRONMENT

MONITOREO DE INFRAESTRUCTURA DE TI CON ZABBIX Y GRAFANA: UN ESTUDIO DE CASO EN UN ENTORNO VIRTUALIZADO EN AWS

Gabriel Bomfim Mota da Silva[1], Felipe Diniz Dallilo[2]

https://doi.org/10.47820/recima21.v7i12.9678

PUBLICADO: 12/2026

RESUMO

Este trabalho apresenta um estudo de caso sobre o monitoramento de infraestrutura de TI com Zabbix e Grafana, tendo o MySQL como repositório de dados. O ambiente foi montado na AWS com duas instâncias EC2 (EC2 1 e EC2 2) com CentOS Stream: a EC2 1 executa o Zabbix Server e o Grafana, e a EC2 2 hospeda o MySQL e o Zabbix Agent. Foram coletadas métricas do sistema operacional e do banco de dados, exibidas em *dashboards* do Grafana, e os alertas do Zabbix foram enviados ao Telegram. Os resultados observados indicam que a coleta, a visualização e a notificação funcionaram no ambiente testado. A redução do tempo médio de reparo (MTTR) não foi medida e permanece como hipótese.

PALAVRAS-CHAVE: Monitoramento de infraestrutura. Zabbix. Grafana. MySQL. Estudo de caso.

ABSTRACT

This paper presents a case study on IT infrastructure monitoring with Zabbix and Grafana, using MySQL as the data repository. The environment was built on AWS with two EC2 instances (EC2 1 and EC2 2) running CentOS Stream: EC2 1 runs Zabbix Server and Grafana, and EC2 2 hosts MySQL and Zabbix Agent. Operating system and database metrics were collected and displayed on Grafana dashboards, and Zabbix alerts were sent to Telegram. The observed results indicate that collection, visualization, and notification worked in the tested environment. The reduction in mean time to repair (MTTR) was not measured and remains a hypothesis.

KEYWORDS: *Infrastructure monitoring. Zabbix. Grafana. MySQL. Case study.*

RESUMEN

Este trabajo presenta un estudio de caso sobre el monitoreo de infraestructura de TI con Zabbix y Grafana, con MySQL como repositorio de datos. El entorno se montó en AWS con dos instancias EC2 (EC2 1 y EC2 2) con CentOS Stream: la EC2 1 ejecuta Zabbix Server y Grafana, y la EC2 2 aloja MySQL y Zabbix Agent. Se recopilaron métricas del sistema operativo y de la base de datos, mostradas en dashboards de Grafana, y las alertas de Zabbix se enviaron a Telegram. Los resultados observados indican que la recolección, la visualización y la notificación funcionaron en el entorno probado. La reducción del tiempo medio de reparación (MTTR) no fue medida y sigue como hipótesis.

[1] Graduando em Sistemas de Informação pela Universidade de Araraquara (UNIARA), Araraquara, SP, Brasil, com foco em monitoramento de infraestrutura de TI.

[2] Graduado em Processamento de Dados pela Faculdade de Tecnologia (FATEC), em 2009. Mestre em Engenharia de *Software* pela Universidade de São Paulo (USP), em 2014. Possui MBAs em Gestão de Pessoas, Gestão de Projetos e Data Science Analytics. Docente do curso de Sistemas de Informação da Universidade de Araraquara (UNIARA), Araraquara, SP, Brasil, e orientador deste trabalho.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

PALABRAS CLAVE: *Monitoreo de infraestructura. Zabbix. Grafana. MySQL. Estudio de caso.*

1. INTRODUÇÃO

Atualmente, muitas empresas dependem de serviços digitais para executar suas atividades. Nesse cenário, a infraestrutura de Tecnologia da Informação (TI), composta por servidores, dispositivos de rede e sistemas de banco de dados, é fundamental para garantir a disponibilidade e o desempenho desses serviços. A ausência de mecanismos eficientes de monitoramento pode dificultar a identificação de falhas e gargalos, ocasionando interrupções nas operações e prejuízos financeiros. Por isso, este trabalho aborda o monitoramento da infraestrutura de TI, com foco em práticas que permitam uma atuação preventiva diante de possíveis incidentes.

Este trabalho tem como objetivo implementar uma solução de monitoramento e visualização de infraestrutura com as ferramentas de código aberto (*open source*) Zabbix e Grafana, tendo o MySQL como repositório das informações coletadas. Para isso, foram realizadas a instalação e a configuração do ambiente de monitoramento, a coleta de métricas do sistema operacional e do banco de dados e a construção de *dashboards* para acompanhamento dos recursos monitorados.

A escolha do tema está relacionada à necessidade de reduzir períodos de indisponibilidade em sistemas essenciais para as empresas. Além disso, o uso de ferramentas *open source* amplamente adotadas pode viabilizar uma estratégia de monitoramento sem os custos de licenciamento de *software* proprietário.

Busca-se analisar de que maneira um sistema centralizado de monitoramento pode auxiliar na identificação mais rápida de falhas em ambientes de infraestrutura de TI. Parte-se da hipótese de que a integração entre o Zabbix, responsável pela coleta das métricas, e o Grafana, utilizado para a visualização, favorece uma resposta mais ágil aos incidentes e contribui para a redução do tempo médio de reparo (*Mean Time To Repair*, MTTR). Essa hipótese não foi testada de forma quantitativa, como detalhado na Metodologia.

O estudo combinou pesquisa bibliográfica e um estudo de caso em ambiente de laboratório, com um servidor CentOS Stream para o Zabbix Server e o Grafana e uma segunda instância para o MySQL, ambos com agentes de monitoramento que enviam métricas pela rede. Os resultados foram avaliados por meio dos painéis do Grafana e das notificações recebidas pelo Telegram.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

2. REFERENCIAL TEÓRICO

2.1. O monitoramento de infraestrutura de TI

O monitoramento é um processo contínuo de coleta e análise de dados sobre o estado de funcionamento de componentes de *hardware* e *software*. Barbosa (2020) apresenta o Zabbix como um sistema de gerenciamento de redes que contribui para a integridade dos ambientes de tecnologia e para o planejamento estratégico das empresas. A análise do histórico das métricas em gráficos permite identificar tendências de uso dos recursos e antecipar necessidades de expansão.

2.1.1. Monitoramento versus observabilidade

Os termos monitoramento e observabilidade costumam ser usados como sinônimos, mas descrevem práticas diferentes e complementares. O monitoramento acompanha métricas definidas de antemão e dispara alertas quando um limite configurado pelo administrador é ultrapassado, o que funciona bem para problemas já conhecidos (HACKONE, 2024). A observabilidade combina métricas, *logs* e rastreamentos (*traces*) para que a equipe consiga deduzir o que ocorre dentro de um sistema a partir dos dados que ele emite, o que ajuda a investigar falhas imprevistas (HACKONE, 2024; TEBALDI, 2019).

Tebaldi (2019) chega a uma conclusão semelhante: o monitoramento tradicional avisa quando algo deu errado, enquanto a observabilidade ajuda a entender o motivo. A solução implementada neste trabalho aproxima-se mais do monitoramento, pois depende de itens e *triggers* configurados previamente no Zabbix e detecta sobretudo falhas previstas pelo administrador. A reunião das métricas em *dashboards* do Grafana facilita o cruzamento de indicadores, o que se aproxima da observabilidade apenas de forma parcial.

2.2. A ferramenta Zabbix

O Zabbix é uma solução de monitoramento distribuído de código aberto e de nível empresarial, que acompanha parâmetros de rede e a saúde e a integridade de servidores, máquinas virtuais, aplicações, serviços e bancos de dados. Segundo a documentação oficial (ZABBIX, [s.d.]), seus componentes principais são o servidor central (Zabbix Server), responsável pelo processamento, pelas coletas e pela cópia de dados em *cache* por meio de processos internos, o banco de dados, que armazena as informações, e a interface web, usada no gerenciamento.

O Zabbix Agent é instalado nos alvos de monitoramento para acompanhar recursos locais e aplicações e enviar os dados coletados ao Zabbix Server (ZABBIX, [s.d.]), como uso de CPU, memória, rede e disco, além de processos e *logs* do sistema.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

2.2.1. Processos internos do Zabbix Server

O Zabbix Server é composto por processos especializados, cada um responsável por uma etapa da coleta, do processamento e do armazenamento das métricas (ZABBIX, [s.d.]). Na inicialização, o servidor lê o arquivo zabbix_server.conf, que define o tamanho dos *caches*, o número de processos coletores e as credenciais do banco de dados; um erro de sintaxe nesse arquivo impede o início do serviço. Em seguida, aloca os *caches* e valida a versão do esquema do banco, executando a atualização (*upgrade*) automática quando necessário.

A coleta é feita por dois tipos de processo. Os *pollers* conectam-se aos *hosts* monitorados e solicitam o valor de cada métrica, o que corresponde aos agentes passivos. Os *trappers* ficam à escuta na porta 10051 e recebem os dados enviados por agentes ativos, *proxies* ou pelo utilitário zabbix_sender. Ambos consultam o *configuration cache* para saber quais itens coletar.

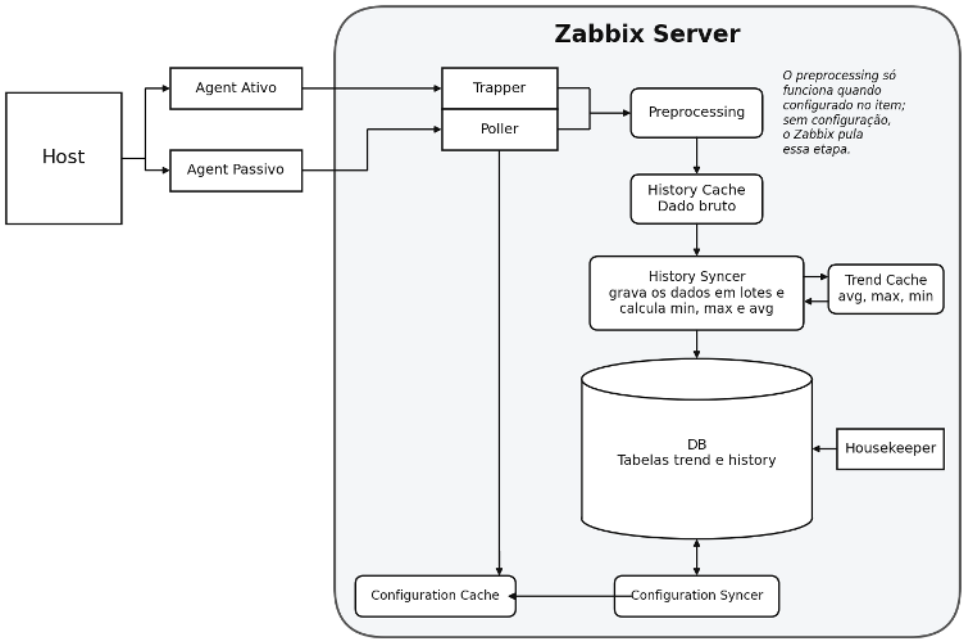
Depois da coleta, os valores passam pelo *preprocessing*, que aplica as regras definidas no item, como cálculo de *delta*, expressões regulares ou *scripts*, quando existem. Os dados seguem para o *history cache*, um *buffer* do qual o *history syncer* os retira e os grava no banco em lotes, o que evita gravações individuais. Se o banco não acompanha o ritmo de gravação, o *cache* se acumula, o que é registrado no *log* do servidor e pode interromper a coleta. Um resumo da última hora (mínimo, média e máximo) é mantido no *trend cache* e gravado na tabela de tendências, usada nos gráficos dos itens.

O *configuration syncer* mantém em memória (*configuration cache*) uma cópia dos *hosts*, itens, *templates* e *triggers*, atualizada conforme o parâmetro CacheUpdateFrequency do zabbix_server.conf, o que evita consultas constantes ao banco. O *housekeeper* remove dados de histórico, eventos e tendências que ultrapassam o período de retenção. O servidor mantém ainda o *value cache*, com valores recentes de histórico usados na avaliação de *triggers* e em consultas, e o *VMware cache*, com os dados coletados da API do VMware.

A Figura 1 resume esse fluxo. O agente ativo envia os dados ao *Trapper*, e os dados do agente passivo são buscados pelo *Poller*. Os valores passam pelo *Preprocessing* (apenas nos itens com regra configurada) e seguem para o *History Cache*, de onde o *History Syncer* os grava em lotes no banco, com apoio do *Trend Cache* para mínimo, média e máximo. O *Housekeeper* remove dados antigos, e o *Configuration Syncer* copia as configurações do banco para o *Configuration Cache*, consultado pelos coletores. As setas indicam o sentido dos dados e das consultas.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 1. Arquitetura interna do Zabbix



Fonte: Elaborado pelos autores (2026).

A Figura 2 mostra a tela de dados recentes do Zabbix para o *host* Firewall BeOnUp, usado como exemplo de item com histórico. O item *ICMP response time* mede o tempo de resposta a pacotes ICMP. A linha informa a última coleta (há 1 min 39 s), o último valor (15,03 ms), a variação (+7,46 ms) e a *tag Application: ICMP*. O *link Graph*, destacado em vermelho, abre o gráfico do histórico, mostrado na Figura 3.

Figura 2. Item ICMP response time

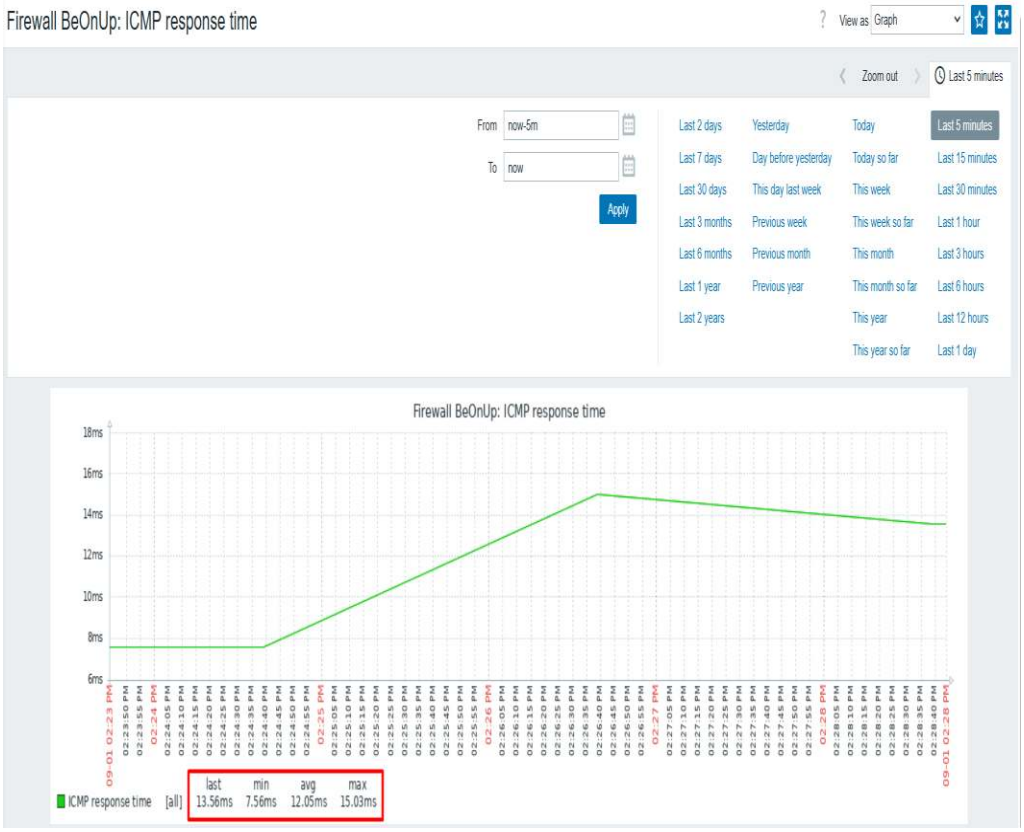


Fonte: Elaborado pelos autores (2026).

A Figura 3 mostra o gráfico do item nos últimos 5 minutos, das 14h23 às 14h28 de 1º de setembro. O tempo de resposta parte de cerca de 7,5 ms, sobe até cerca de 15 ms às 14h26min40s e cai para 13,56 ms. O quadro vermelho resume o período: último valor de 13,56 ms, mínimo de 7,56 ms, média de 12,05 ms e máximo de 15,03 ms. O gráfico evidencia que o Zabbix guarda o histórico de cada item e permite consultá-lo em outras janelas de tempo (*Last 15 minutes*, *Last 1 hour*, *Last 1 day* e outras).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 3. Gráfico do item ICMP response time



Fonte: Elaborado pelos autores (2026).

2.3. Sistema gerenciador de banco de dados MySQL

Como repositório central das métricas, o MySQL atua como sistema gerenciador de banco de dados (SGBD) de código aberto baseado na linguagem SQL (ORACLE CORPORATION, [s.d.]), mantendo o histórico de métricas e o inventário dos *hosts*. Entre os bancos de dados compatíveis com o Zabbix, está o MySQL (ZABBIX, [s.d.]).

2.4. Visualização de dados com Grafana

O Zabbix oferece recursos próprios de visualização, como mapas e *dashboards*, e o Grafana permite consultar dados de diferentes fontes e montar *dashboards* personalizados (GRAFANA LABS, [s.d.]). No ambiente estudado, a conexão entre as duas ferramentas é feita pelo *plugin* alexanderzobnin-zabbix-app (Figura 5). Borges (2025) integrou o Zabbix, para a coleta, o MySQL, para o armazenamento histórico, e o Grafana, para a visualização, e obteve um *dashboard* interativo com gráficos de fácil interpretação, que apoia decisões técnicas rápidas.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

3. METODOLOGIA

Quanto à natureza, esta pesquisa classifica-se como aplicada: ela parte de um problema concreto, a dificuldade de identificar falhas com agilidade em ambientes de infraestrutura de TI, e busca uma solução prática para esse problema (GIL, 2002). Quanto aos objetivos gerais, tem caráter exploratório, já que procura maior familiaridade com o tema da observabilidade de infraestrutura, tornando-o mais claro e permitindo formular hipóteses que possam ser testadas.

Quanto aos procedimentos técnicos, o trabalho é um estudo de caso apoiado em pesquisa bibliográfica. O caso estudado é um ambiente de monitoramento montado na nuvem, composto por Zabbix, Grafana e MySQL, analisado em profundidade dentro de um contexto delimitado (GIL, 2002). A parte bibliográfica envolveu o levantamento da documentação oficial das ferramentas e de trabalhos acadêmicos sobre monitoramento de infraestrutura, que serviram de base para as escolhas técnicas. A parte prática consistiu em montar um ambiente controlado na AWS e observar, nos painéis do Grafana, como as métricas coletadas pelo Zabbix aparecem e se isso favorece a identificação rápida de falhas, hipótese central deste trabalho.

A escolha da Amazon Web Services (AWS) como provedor de nuvem tem uma razão prática: os recursos podem ser criados e destruídos rapidamente, o que facilita repetir os testes quantas vezes for preciso. Também pesou o fato de a AWS ser amplamente usada em ambientes corporativos reais, o que aproxima o laboratório de um cenário de produção. O sistema operacional escolhido, CentOS Stream, pertence à família Red Hat Enterprise Linux, para a qual o Zabbix 6.0 LTS oferece pacotes oficiais de instalação (ZABBIX, [s.d.]), e é usado em servidores corporativos.

As métricas monitoradas foram escolhidas por serem indicadores usuais da saúde de uma infraestrutura: uso de CPU e memória, disponibilidade dos serviços e desempenho do banco de dados. A seção 3.3 detalha o que foi coletado.

A validação do ambiente foi qualitativa. Ela consistiu em conferir se as métricas coletadas pelo Zabbix chegavam aos painéis do Grafana e se esses painéis refletiam o estado dos serviços, como o status dos agentes, as respostas HTTP e os indicadores do MySQL. Também se verificou se os alertas gerados pelo Zabbix chegavam ao Telegram. Medir o tempo médio de reparo (MTTR) em ambiente de produção ficou fora do escopo, ponto retomado na seção 4.3.

3.1. Objeto de estudo e critérios de avaliação

Este estudo de caso avalia uma solução de monitoramento específica: o Zabbix como coletor de métricas, o MySQL como repositório e o Grafana como camada de visualização, instalados em duas instâncias EC2. A avaliação busca responder se essa combinação permite identificar falhas com mais rapidez e, para isso, observa quatro pontos:

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

- a coleta distribuída: se o Zabbix Server, em uma instância, consegue coletar métricas do sistema operacional e do MySQL que está em outra instância;
- a integração entre as ferramentas: se o Grafana consulta os dados do Zabbix pela API nativa e os exibe em painéis;
- a legibilidade dos painéis: se o estado dos serviços, o uso de memória e disco, as respostas HTTP e os comandos executados no banco podem ser lidos rapidamente;
- a cobertura dos gatilhos críticos: se as condições previstas para cada métrica (serviço fora do ar, excesso de conexões, CPU alta e latência elevada) correspondem a situações que o administrador precisa conhecer antes que virem indisponibilidade.

Os painéis que sustentam essa avaliação são apresentados na seção 4.1. Os alertas enviados pelo Telegram, descritos na seção 4.2, complementam a avaliação, e a análise dos resultados fica na seção 4.3.

3.2. Arquitetura do ambiente de testes distribuído

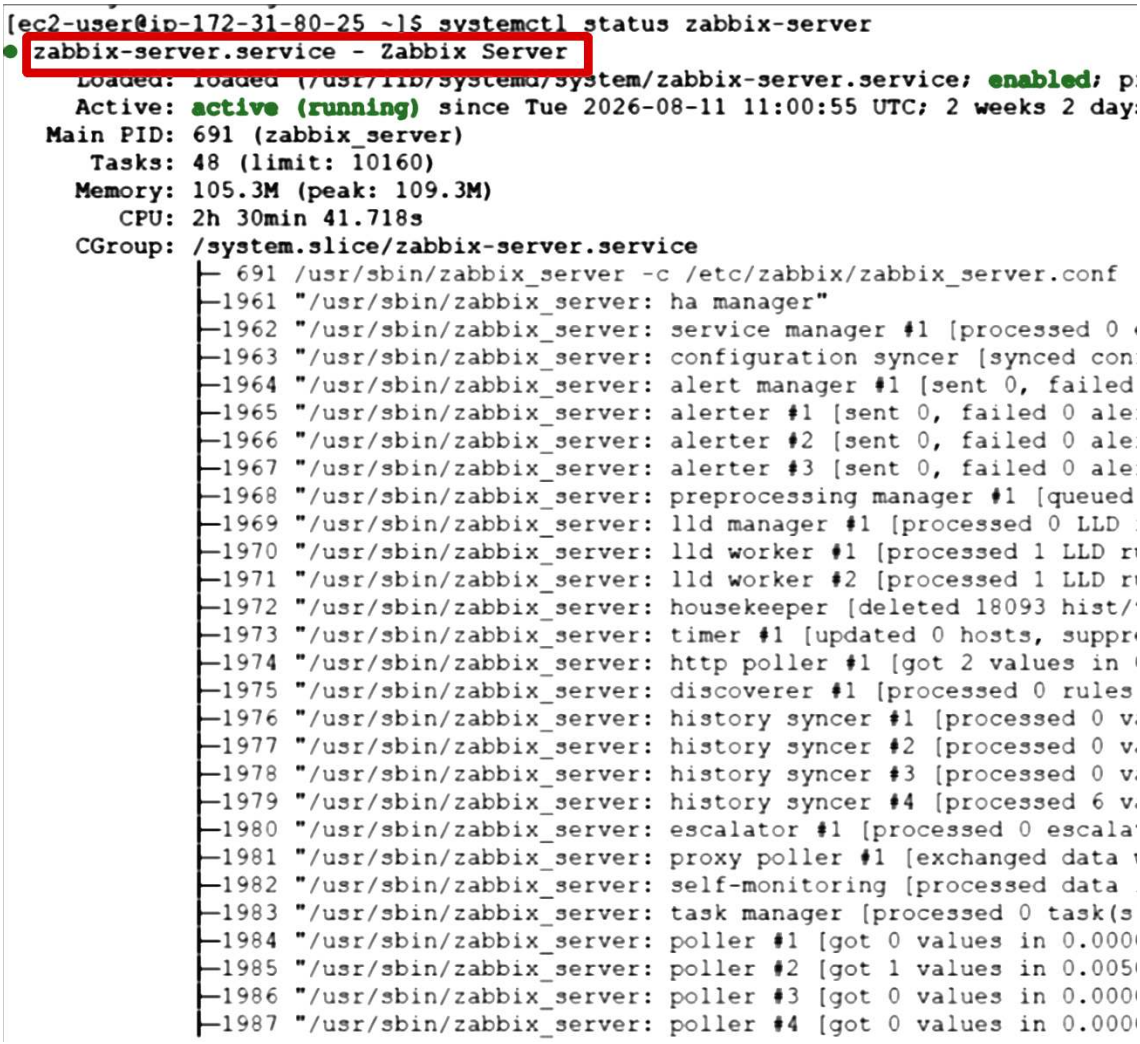
Para a realização dos testes, foram criadas duas instâncias EC2 na nuvem da Amazon Web Services (AWS). As instâncias permanecem isoladas logicamente, mas conectadas pela mesma VPC (Rede Virtual Privada) com liberação de regras de IP e de portas para permitir a comunicação entre os serviços. A distribuição de recursos, papéis e serviços foi definida conforme descrito a seguir:

- EC2 1 (Servidor de Aplicação de Monitoramento): Instanciada com o sistema operacional CentOS Stream, configurada como uma t3.small (família t3) com 2 vCPUs e 2 GB de memória RAM. Essa EC2 é dedicada exclusivamente à execução do ecossistema de gerenciamento, hospedando o Zabbix Server (responsável pelo motor de coleta e regras de checagem) e o Grafana (responsável pela visualização).

A Figura 4 foi obtida com o comando *systemctl status zabbix-server* na EC2 1 e confirma que o Zabbix Server estava em execução: o serviço está habilitado (*enabled*) para iniciar com o sistema e consta como *active (running)* desde 11 de agosto de 2026. A saída informa o PID principal (691), 48 tarefas, 105,3 MB de memória (pico de 109,3 MB) e 2 h 30 min de CPU acumulada. A árvore *CGroup* lista os processos filhos descritos na seção 2.2.1, como *configuration syncer*, *alerters*, *preprocessing manager*, *housekeeper*, *history syncers* e *pollers*.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 4. Zabbix Server em execução

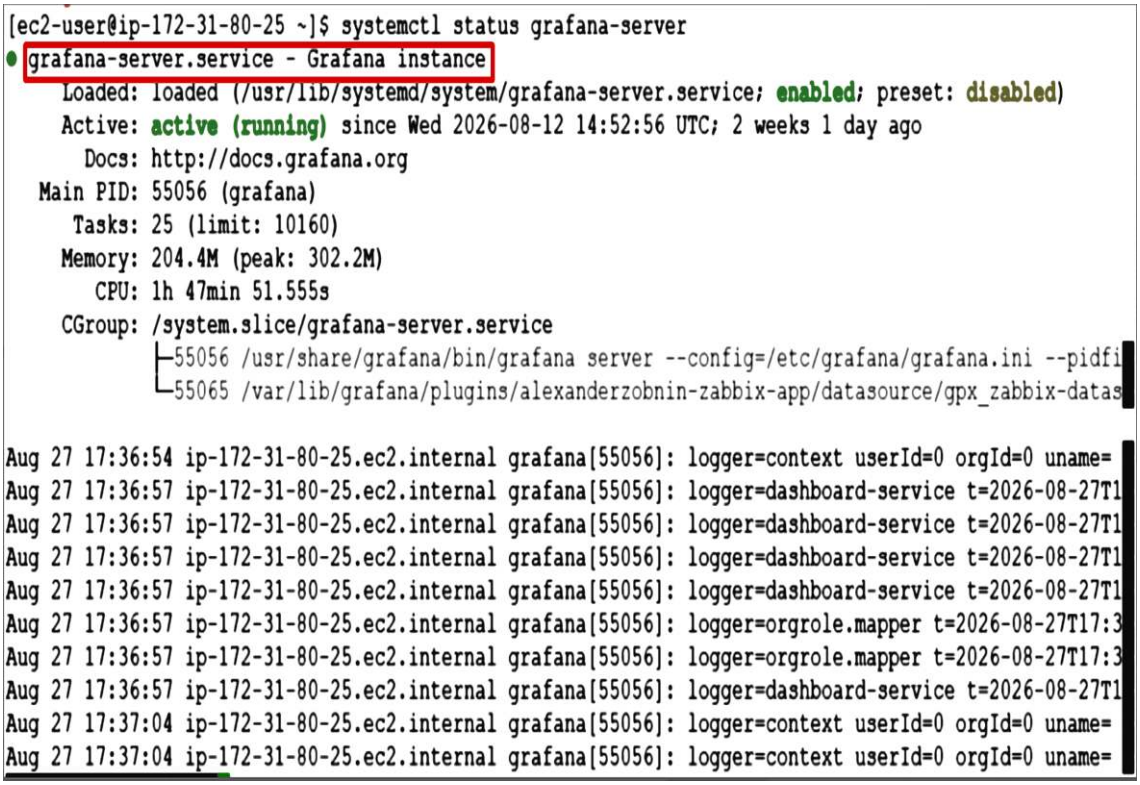


Fonte: Elaborado pelos autores (2026).

A Figura 5 mostra a mesma verificação para o Grafana, com o comando *systemctl status grafana-server*. O serviço consta como *active (running)* desde 12 de agosto de 2026, com 25 tarefas, 204,4 MB de memória (pico de 302,2 MB) e cerca de 1 h 47 min de CPU. A árvore de processos inclui o *plugin* alexanderzobnin-zabbix-app, que permite ao Grafana usar o Zabbix como fonte de dados, e os registros finais (*logs*) mostram o atendimento de requisições de painéis (*dashboard-service*). Juntas, as Figuras 4 e 5 indicam que os dois componentes da EC2 1 estavam em execução.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 5. Grafana Server em execução



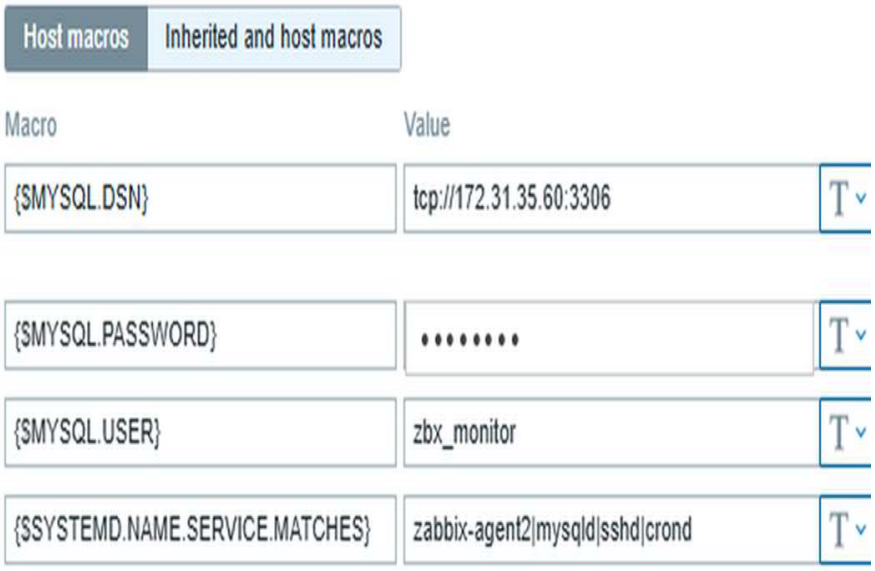
Fonte: Elaborado pelos autores (2026).

– EC2 2 (Servidor de Banco de Dados - *Host* Alvo): Instanciada também com o sistema operacional CentOS Stream, configurada como uma t3.small (família t3) com 2 vCPUs e 2 GB de memória RAM. Essa EC2 atua como o repositório centralizado do ambiente, hospedando o SGBD MySQL (que armazena a base de dados do Zabbix). Simultaneamente, esta máquina executa o Zabbix Agent para que sua infraestrutura e a performance do próprio banco de dados sejam monitoradas ativamente.

A Figura 6 mostra a aba *Macros* do *host* Banco_zabbixserver. Macros são variáveis usadas pelo *template* de monitoramento do MySQL para a conexão com o banco. Foram configuradas quatro: {$MYSQL.DSN}, com o endereço e a porta (protocolo tcp, porta 3306); {$MYSQL.USER}, com o usuário zbx_monitor; {$MYSQL.PASSWORD}, com a senha, ocultada na imagem; e {$SYSTEMD.NAME.SERVICE.MATCHES}, que filtra os serviços do *systemd* acompanhados (zabbix-agent2, mysqld, sshd e crond). Assim, o Zabbix consulta o MySQL da EC2 2 e verifica o estado desses serviços.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 6. Configuração de macros para conexão com o MySQL



Fonte: Elaborado pelos autores (2026).

A Figura 7 lista itens coletados do MySQL no *host* Banco_zabbixserver, todos com a *tag component: connections*. Cada linha é uma métrica, com nome, última coleta, último valor e variação. Os itens tratam de conexões: clientes e conexões abortadas por segundo, erros de conexão por causa (*accept*, *internal*, *max connections*, *peer address*, *select* e *tcpwrap*), conexões por segundo e máximo de conexões usadas. Na captura, os contadores de erro e de aborto estavam em 0. As conexões por segundo estavam em 0,4454 (variação de -0,01045) e o máximo de conexões simultâneas usadas era 31. Esses itens sustentam o acompanhamento de conexões descrito na seção 3.3.

Figura 7. Itens do MySQL monitorados pelo Zabbix

Host	Name	Last check	Last value	Change	Tags
Banco_zabbixserver	MySQL: Aborted clients per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Aborted connections per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connection errors accept per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connection errors internal per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connection errors max connections per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connection errors peer address per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connection errors select per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connection errors tcpwrap per second	31s	0		component: connections
Banco_zabbixserver	MySQL: Connections per second	31s	0.4454	-0.01045	component: connections
Banco_zabbixserver	MySQL: Max used connections	21m 31s	31		component: connections

Fonte: Elaborado pelos autores (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

3.3. Definição de métricas e coleta de dados do SGBD

A extração de informações é distribuída: o Zabbix Server, na EC2 1, conecta-se à EC2 2 para coletar as métricas do sistema operacional e os indicadores de desempenho do MySQL (por consultas de verificação e monitoramento de processos). As métricas e os gatilhos críticos previstos são:

- Status do serviço MySQL (EC2 2): indica se o banco está em execução, com painel de status. Gatilho crítico: serviço fora do ar.
- Número de conexões ativas (EC2 2): exibido em gráfico de linha. Gatilho crítico: conexões acima de 80% do limite permitido.
- Volume de consultas por segundo (EC2 2): exibido como série temporal. Gatilho crítico: queda abrupta no número de consultas.
- Uso de CPU e memória RAM (EC2 2): exibido em gráfico de área. Gatilho crítico: consumo de CPU acima de 85% por 5 minutos.
- Latência de rede entre a EC2 1 e a EC2 2: exibida como série temporal. Gatilho crítico: tempo de resposta acima de 200 ms.

4. RESULTADOS E DISCUSSÃO

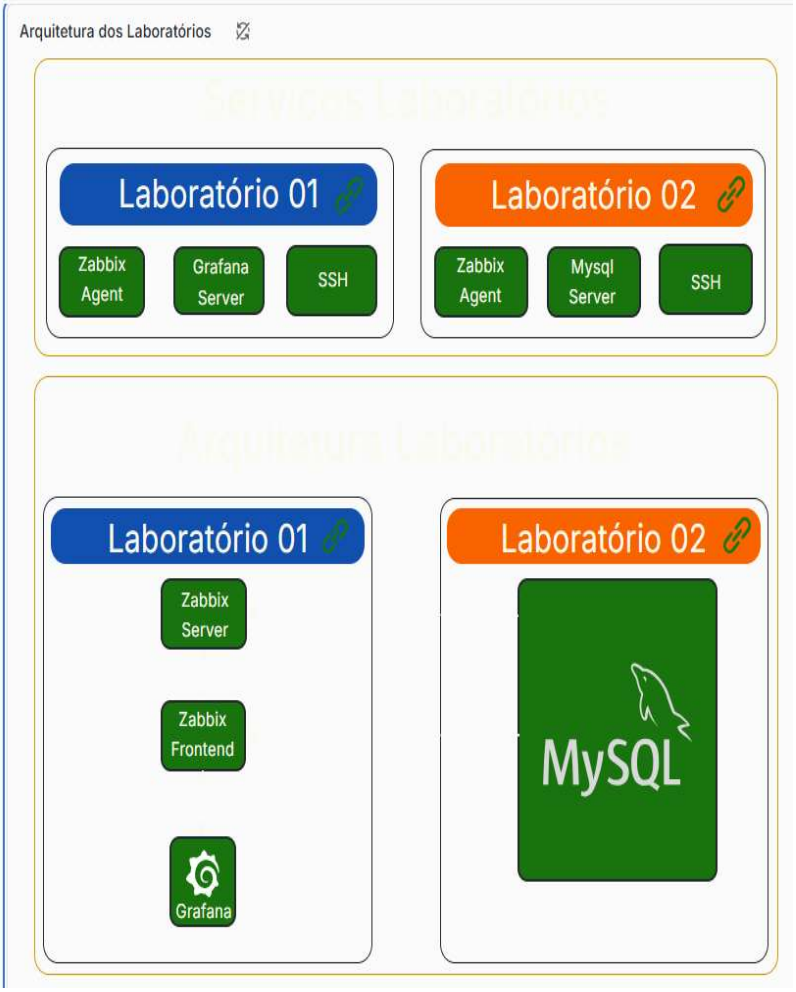
4.1. Integração e construção dos dashboards no Grafana

O Grafana consome a API do Zabbix para exibir, em *dashboards*, os dados históricos coletados e armazenados no MySQL, o que reúne em um só lugar o estado do banco de dados e o da própria aplicação de monitoramento.

A Figura 8 mostra o painel de arquitetura criado no Grafana como mapa do ambiente. A área superior (*Serviços Laboratórios*) lista os serviços de cada máquina: no Laboratório 01 (EC2 1), Zabbix Agent, Grafana Server e SSH; no Laboratório 02 (EC2 2), Zabbix Agent, MySQL Server e SSH, todos em verde (UP). A área inferior (*Arquitetura Laboratórios*) apresenta os componentes: Zabbix Server, Zabbix Frontend e Grafana no Laboratório 01, e o MySQL no Laboratório 02. Os rótulos azul e laranja distinguem as duas máquinas.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 8. Arquitetura dos Laboratórios 01 e 02

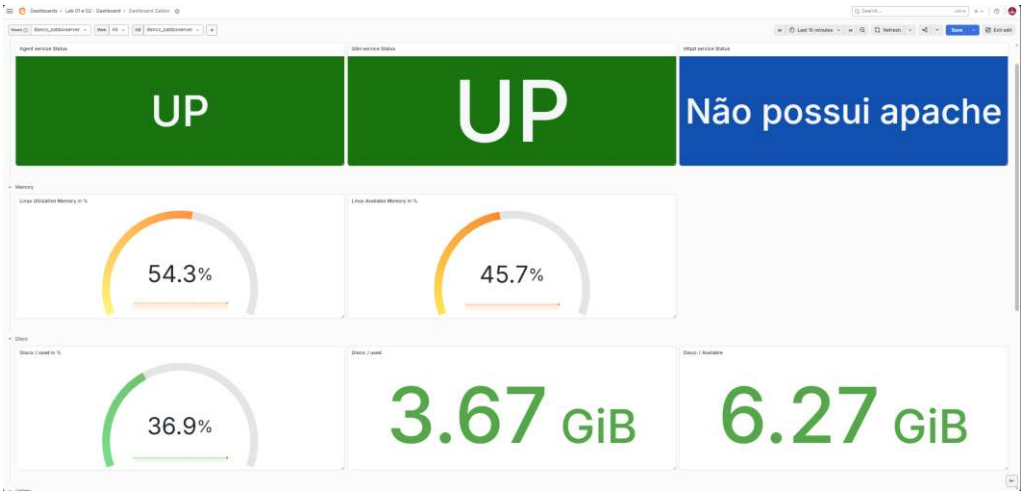


Fonte: Elaborado pelos autores (2026).

A Figura 9 mostra a aba *System* do *dashboard* Lab 01 e 02, filtrada para o *host* Banco_zabbixserver, com período de 15 minutos (*Last 15 minutes*). O primeiro grupo exibe o status de serviços: *Agent service Status* e *SSH service Status* aparecem em UP (verde), e *Httpd service Status* exibe a mensagem "Não possui apache" (azul), que indica que o *host* não executa o servidor web. O grupo *Memory* mostra 54,3% de memória em uso (*Linux Utilization Memory in %*) e 45,7% disponível. O grupo *Disco* mostra a partição raiz com 36,9% de uso, 3,67 GiB ocupados e 6,27 GiB livres, ou seja, cerca de 9,94 GiB no total. Com esses painéis, o administrador verifica rapidamente se os serviços do *host* estão ativos e quanto resta de memória e de disco; na captura, não há sinal de esgotamento de recursos.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

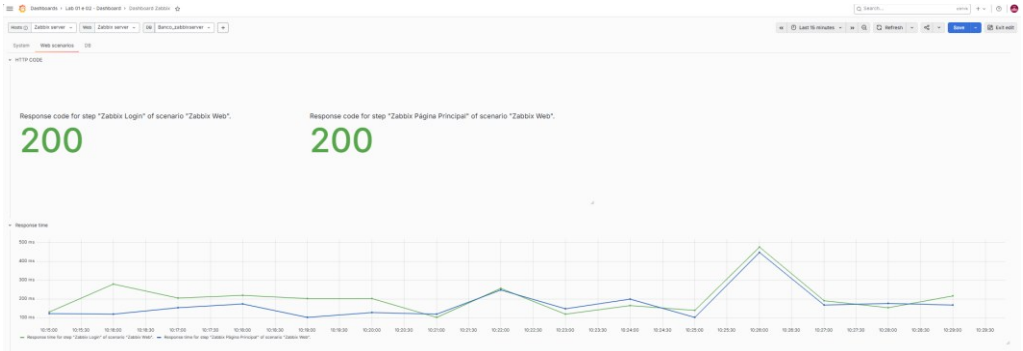
Figura 9. Dashboard System: visão geral do host Banco_zabbixserver



Fonte: Elaborado pelos autores (2026).

A Figura 10 mostra a aba *Web scenarios*, com o *host* Zabbix server selecionado. Um *web scenario* é uma verificação HTTP periódica dividida em etapas (*steps*). O cenário *Zabbix Web* tem as etapas *Zabbix Login* e *Zabbix Página Principal*, e ambas retornaram o código HTTP 200, que indica requisição atendida com sucesso. O gráfico *Response time* cobre os últimos 15 minutos, das 10h15 às 10h29: os valores ficaram, em geral, entre cerca de 100 ms e 300 ms, e as duas etapas subiram juntas até cerca de 480 ms por volta das 10h26, voltando em seguida ao patamar anterior. Como o aumento ocorreu nas duas etapas ao mesmo tempo, ele sugere lentidão geral do servidor naquele momento.

Figura 10. Dashboard Web Scenarios: monitoramento HTTP



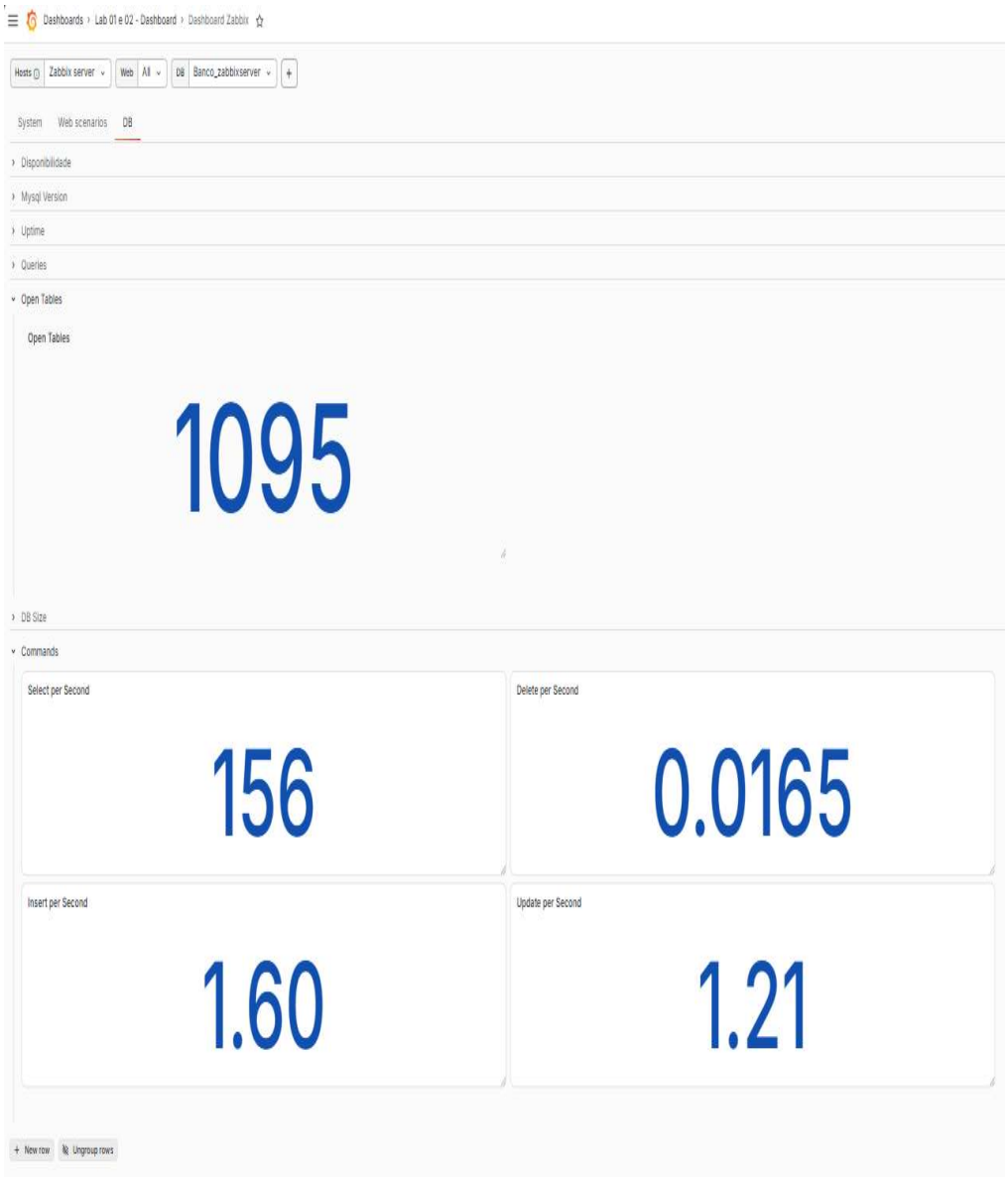
Fonte: Elaborado pelos autores (2026).

A Figura 11 mostra a aba *DB*, com o *host* Zabbix server e o banco Banco_zabbixserver selecionados e período de 15 minutos. As seções *Disponibilidade*, *MySQL Version*, *Uptime*,

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Queries e *DB Size* estão recolhidas, e as seções *Open Tables* e *Commands* estão abertas. O painel *Open Tables* exibe 1095, número de tabelas abertas pelo MySQL no momento da captura. O painel *Commands* exibe a taxa de comandos por segundo: 156 *SELECT*, 1,60 *INSERT*, 1,21 *UPDATE* e 0,0165 *DELETE*, o que indica carga predominantemente de leitura no período.

Figura 11. Dashboard DB: métricas de desempenho do MySQL



Fonte: Elaborado pelos autores (2026).

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

4.2. Mecanismo de alertas e notificação via Telegram

Além da coleta e do armazenamento de métricas, o Zabbix disponibiliza um mecanismo de notificação que permite alertar os responsáveis pela infraestrutura assim que um problema é identificado. Esse mecanismo é acionado por uma *trigger*, uma expressão lógica associada a um ou mais itens monitorados que, ao ser avaliada como verdadeira (por exemplo, quando o uso de CPU permanece acima de um limite por um determinado período), gera um evento de problema. A partir desse evento, uma *action* (ação) previamente configurada avalia condições como severidade, *host* ou tags para determinar quem deve ser notificado e por qual canal.

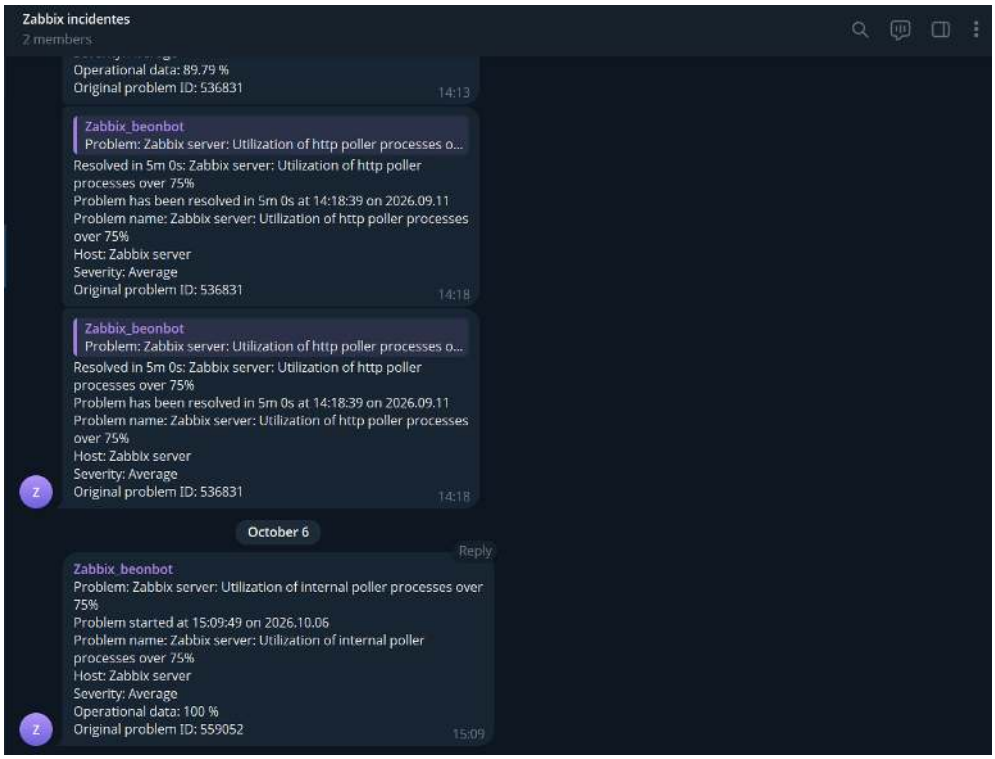
Entre os canais suportados, o Zabbix oferece *webhook* nativo para integração com serviços de mensagens como o Telegram (ZABBIX, [s.d.]). A integração depende de um *bot* criado no BotFather, que fornece um *token* único usado para autenticar as chamadas à API. No Zabbix, o *webhook* é cadastrado como *media type*, que armazena o *token* do *bot* e o *script* de formatação e envio da mensagem; cada usuário que deve receber alertas recebe uma mídia do tipo Telegram vinculada ao seu identificador de chat (*chat ID*), individual ou de grupo (TEBALDI, 2026).

Esse fluxo liga a detecção automática de um problema à resposta humana. É possível configurar escalonamento em múltiplos níveis, de modo que, se o alerta não for reconhecido em determinado tempo, a notificação seja reenviada ou direcionada a outros responsáveis (TEBALDI, 2026). Recomenda-se tratar o *token* do *bot* como credencial sensível e evitar dados confidenciais nas mensagens, pois os alertas costumam ser recebidos em grupos com vários participantes.

A Figura 12 mostra o grupo do Telegram *Zabbix incidentes*, com dois membros, no qual o *bot* Zabbix_beonbot publica os alertas do Zabbix. Cada mensagem traz o problema, a data e o horário de início ou de resolução, o nome do problema (*Problem name*), o *host* afetado (*Host*), a severidade (*Severity*), o dado operacional (*Operational data*) e o identificador original do problema (*Original problem ID*). Na parte superior, uma mensagem informa que o problema *Utilization of http poller processes over 75%*, do *host* Zabbix server, com severidade *Average*, foi resolvido em 5 minutos, às 14h18min39s de 11 de setembro de 2026; o identificador 536831 também consta na notificação de abertura, que registrava 89,79% de utilização. Na parte inferior, em 6 de outubro, outra mensagem registra o início, às 15h09min49s, do problema *Utilization of internal poller processes over 75%*, com utilização de 100%.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Figura 12. Notificações do Zabbix no Telegram



Fonte: Elaborado pelos autores (2026).

4.3. Discussão dos resultados

Os resultados observados mostram que a integração entre Zabbix, Grafana e MySQL funcionou no ambiente de laboratório. O Zabbix Server, na EC2 1, coletou métricas do próprio ambiente e do MySQL da EC2 2 (Figuras 4 a 8); o Grafana exibiu esses dados em *dashboards* de estado de serviços, memória, disco, respostas HTTP e comandos do banco (Figuras 9 a 11); e o Zabbix enviou alertas ao Telegram (Figura 12).

Os painéis reúnem em poucas telas informações que, de outro modo, exigiriam consultar cada servidor. O cenário web (Figura 10) acompanha a disponibilidade de páginas via HTTP e permite distinguir um serviço lento de um serviço fora do ar, e os comandos por segundo (Figura 11) permitem perceber mudanças no perfil de carga do MySQL. Reunir tipos diferentes de métrica na mesma tela aproxima a solução da observabilidade discutida na seção 2.1.1, ainda que de forma parcial.

Não foi demonstrado, porém, que essa integração reduz o MTTR. A hipótese permanece como resultado esperado: confirmá-la exigiria dados de incidentes em ambiente de produção, por um período mais longo, com e sem a solução, o que ficou fora do escopo do trabalho.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

Também não foi medido o tempo que a equipe leva para reagir aos alertas do Telegram. Os resultados valem, portanto, para o ambiente de laboratório com duas instâncias EC2 e não devem ser generalizados para ambientes corporativos reais.

5. CONSIDERAÇÕES FINAIS

Este trabalho implementou uma solução de monitoramento de infraestrutura de TI com Zabbix, Grafana e MySQL em duas instâncias EC2 na AWS, para avaliar como essa integração pode favorecer a identificação mais ágil de falhas. No ambiente testado, foi possível instalar e configurar o monitoramento, coletar métricas do sistema operacional e do banco de dados, exibi-las em *dashboards* do Grafana e receber alertas do Zabbix no Telegram. Esses são os benefícios efetivamente observados.

São benefícios apenas esperados a redução do tempo médio de reparo (MTTR) e o aumento da confiabilidade dos serviços, que não foram medidos. As limitações do estudo são a escala reduzida (duas instâncias e um serviço monitorado, o MySQL), a distância em relação a infraestruturas corporativas com dezenas ou centenas de *hosts* e a avaliação qualitativa, sem dados históricos de incidentes em produção para comparar cenários com e sem a solução.

Como trabalhos futuros, sugere-se medir o tempo de resposta da equipe aos alertas do Telegram, o que permitiria calcular o MTTR de forma quantitativa; incluir outros tipos de *host*, como contêineres e balanceadores de carga; avaliar *proxies* Zabbix em redes distribuídas; e incorporar *logs* e rastreamentos (*traces*) para aproximar a solução da observabilidade.

REFERÊNCIAS

BARBOSA, I. F. **Gestão da infraestrutura de TI com Zabbix**. 2020. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Informática para Negócios) – Faculdade de Tecnologia de São José do Rio Preto, São José do Rio Preto, 2020. Disponível em: https://ric.cps.sp.gov.br/handle/123456789/25481. Acesso em: 12 nov. 2026.

BORGES, L. M. R. **Dashboard interativo para monitoramento e apoio à gestão de indicadores de desempenho de infraestrutura de TI e bancos de dados**. 2025. Trabalho de Conclusão de Curso (Graduação em Tecnologias da Informação e Comunicação) – Universidade Federal de Santa Catarina, Campus Araranguá, Araranguá, 2025. Disponível em: https://repositorio.ufsc.br/handle/123456789/267647. Acesso em: 12 nov. 2026.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002.

GRAFANA LABS. **Grafana documentation**. [S. l.]: Grafana Labs, [s.d.]. Disponível em: https://grafana.com/docs/. Acesso em: 12 nov. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.

HACKONE. **Entendendo Observabilidade: A Evolução do Monitoramento**. [S. l.]: Hackone, 17 jul. 2024. Disponível em: https://hackone.com.br/blog/entendendo-observabilidade-a-evolucao-do-monitoramento/. Acesso em: 14 out. 2026.

ORACLE CORPORATION. **MySQL 8.0 Reference Manual**. [S. l.]: Oracle, [s.d.]. Disponível em: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html. Acesso em: 12 nov. 2026.

TEBALDI, P. **O que é Observabilidade? Guia Completo: Pilares, Ferramentas e Implementação**. [S. l.]: OpServices, 4 jul. 2019. Atualizado em: 30 set. 2026. Disponível em: https://www.opservices.com.br/o-que-e-observabilidade/. Acesso em: 14 out. 2026.

TEBALDI, P. **Notificações no Zabbix: do bot do Telegram ao escalonamento em 3 níveis**. [S. l.]: OpServices, 27 ago. 2026. Disponível em: https://www.opservices.com.br/zabbix-telegram/. Acesso em: 14 out. 2026.

ZABBIX. **Zabbix Documentation 6.0 LTS**. [S. l.]: Zabbix, [s.d.]. Disponível em: https://www.zabbix.com/documentation/6.0/en. Acesso em: 12 nov. 2026.

Este artigo é publicado em acesso aberto (Open Access) sob a licença Creative Commons Atribuição 4.0 Internacional (CC-BY), que permite uso, distribuição e reprodução irrestritos em qualquer meio, desde que o autor original e a fonte sejam creditados.