

MONITORAMENTO AMBIENTAL DE DATACENTERS: AUTOMATIZANDO O PROCESSO DE IMPLANTAÇÃO DO AMBIENTE DE IOT E DISPONIBILIZANDO AS INFORMAÇÕES PARA ACESSO REMOTO

Leandro de Meirelles¹
Eugenio Sper de Almeida²

¹Aluno do CST em Análise e Desenvolvimento de Sistemas; e-mail: leandro.meirelles@fatec.sp.gov.br

²Professor da Fatec Cruzeiro – Prof. Waldomiro May; e-mail: eugenio.almeida@fatec.sp.gov.br

Área do Conhecimento: Sistemas de Computação.

Palavras-chave: IoT. Datacenter. Automatização. Container.

INTRODUÇÃO

O bolsista PIBITI CPS/CNPq Guilherme da Silva Donizetti (Processo: 146177/2020-7) desenvolveu um sistema de monitoramento ambiental de datacenters. O projeto focou em disponibilizar dados em tempo real, assim como proporcionar acesso ao histórico da temperatura e umidade. Este sistema integrou diversas ferramentas, resultando no envio, armazenamento e visualização de dados provenientes de sensores baseados em IoT (MADAKAM, 2015) alocados pelo datacenter. Identificamos algumas limitações no sistema atual:

- Necessidade de instalação e inicialização manual das ferramentas, uma vez que o sistema executa diretamente sobre o sistema operacional da máquina local;
- Todas as dependências das ferramentas devem ser contempladas em versões específicas, assim como devem ser compatíveis com outras propriedades e configurações do sistema operacional, gerando uma necessidade periódica de verificação manual do estado das dependências;
- O fluxo de dados (Workflow) que define as etapas deste sistema de IoT consiste em coleta, transmissão, armazenamento e visualização dos dados ambientais. Cada ferramenta é parte deste Workflow e inicializá-las em uma ordem incorreta pode acarretar no mal funcionamento do sistema.

Em um aplicações de IoT, um SGBD necessita suportar a geração de um enorme volume de dados que nem sempre se relacionam. Segundo Nasar et al. (2019), os sistemas NoSQL são ideais para IoT devido a sua escalabilidade horizontal e a capacidade de consumir dados rapidamente. Continuous Integration/Continuous Delivery (CI/CD) é um método para entregar aplicações com frequência aos clientes através da automação nas etapas do desenvolvimento de aplicações, permitindo solucionar problemas que a integração de novos códigos pode causar às equipes de operações e desenvolvimento (LAUKKANEN et al. 2017). Entre as ferramentas de processo de automação podemos citar o Docker, o Git e o Ansible. O Docker é uma ferramenta que encapsula o processo de criação de um artefato distribuído para qualquer aplicação em escala e em qualquer ambiente e racionaliza o fluxo de trabalho (JUNIOR, 2020). O Docker baseia-se na criação, configuração e gerenciamento de containers, que permitem o funcionamento dos processos em redes isoladas (MONTEIRO, 2017). O Git é um tipo de Sistema de Controle de Versão (VCS) cujo objetivo é registrar todas as mudanças feitas pelos desenvolvedores de determinado código/projeto. Ansible é uma ferramenta de automação destinada principalmente à implantação de aplicativos, atualizações de servidores, provisionamento de nuvem, gerenciamento de configuração e orquestração de serviços (Meijer et al., 2022). Utiliza-se o Ansible para modelar toda a infraestrutura de TI porque se concentra na maneira como os diferentes sistemas interagem, em vez de gerenciá-los um por um. As configurações são representadas como manuais (“playbooks”) que podem ser implantados em uma ou várias máquinas ao mesmo tempo.

OBJETIVOS

O objetivo atual do projeto é eliminar essas limitações, implementar novas funcionalidades e disponibilizar acesso aos clientes, utilizando tecnologias de containers e automatização. Para isso, padronizou-se as configurações de ferramentas e automatizouse o processo de implantação do ambiente de IoT. A implementação de mecanismos para implantar as ferramentas de IoT de forma automática e unificada elimina a necessidade de instalação manual do sistema e possibilita a utilização desta solução de IoT em um ambiente de produção.

METODOLOGIA

Para o desenvolvimento do projeto foram utilizadas as ferramentas Python (3.8.5), broker MQTT Mosquitto (1.6.12), Node-Red (1.2.2), InfluxDB (1.8), Grafana (8.2.1), Node JS (8), Docker (20.10.12), Docker Compose (3.2) e Ansible (2.9). Essas ferramentas são executadas em containers separados que são conectados por uma rede interna, evitando a possibilidade de que um erro em uma das ferramentas afete a outra. O *Docker* possibilita a criação dos containers. Neste projeto usamos "volumes", arquivos onde são armazenados o conteúdo dos containers, para o armazenamento da configuração e o conteúdo das ferramentas. Os dados de configuração dos serviços, os arquivos de criação dos containers e o script que simula os dados dos sensores encontram-se no arquivo "ansible.yml", responsável pela automação da instalação do *Docker*, suas dependências e inicialização dos containers. Ele permite o gerenciamento da configuração e a implantação de aplicações de uma maneira bastante simples. Toda a configuração para a instalação do sistema de IoT encontra-se no Github (https://github.com/eugenioalmeida/Datacenter_IoT), onde pode ser baixada e instalada utilizando o Ansible. A Figura 1 apresenta o fluxo dos dados (*Workflow*) do sistema de IoT com suas ferramentas encapsuladas em containers *Docker*.

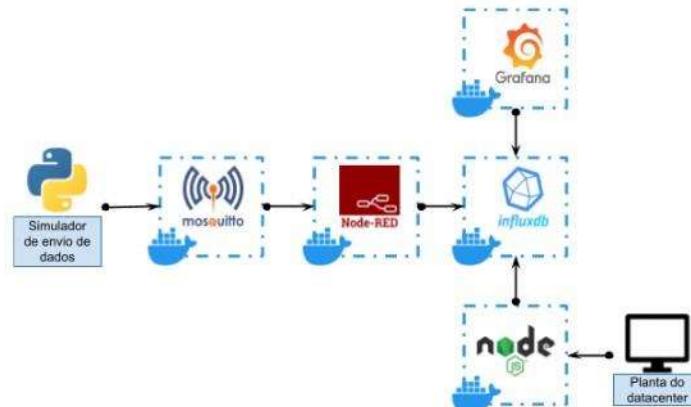


Figura 1: Workflow do projeto

RESULTADOS E DISCUSSÃO

O sistema atual recebe os dados de sensores, armazena e os apresenta em gráficos e na planta baixa do datacenter. A execução em containers, com a utilização do Docker, permite a manipulação de containers e padroniza a estrutura dos serviços, eliminando a necessidade de configuração manual. O orquestrador de containers Docker Compose cria e executa de forma coordenada os containers (levando em consideração a existência de uma relação de dependência entre alguns containers). Utiliza-se um arquivo YAML ("Yet another Markup Language") e um único comando para criar e iniciar todos os serviços da sua configuração. Todas as ferramentas que complementam o sistema precisam ser inicializadas em containers utilizando o Docker com suas dependências devidamente instaladas no servidor. Com o propósito de suprir esta necessidade sem exigir a instalação manual de cada dependência, o Ansible foi implementado no projeto. Resumindo, nesta aplicação a função do Ansible é executar os comandos para a instalação dos recursos necessários na máquina que pretende rodar o sistema. Ao término dessas etapas o sistema estará pronto para ser utilizado. Caso haja falha em uma dessas etapas, o processo é interrompido. Desta forma, o sistema poderá ser instalado utilizando um único comando, ao mesmo tempo que fornece informações sobre sua instalação. A implementação com Ansible elimina o trabalho

manual da instalação (propício a apresentar problemas). Além dos containers que realizam a obtenção, o transporte e o armazenamento dos dados existem outros dois containers que representam duas formas de visualizar os dados:

- **Grafana:** permite visualizar os dados através de gráficos permitindo o monitoramento ambiental do datacenter.
- **Planta do datacenter:** permite visualizar os dados ambientais em planta baixa do datacenter, no padrão da ANSI/TIA-942, através de página web desenvolvida em Node JS.

CONCLUSÕES

Este trabalho teve como objetivo automatizar do processo de implantação do ambiente de IoT para o monitoramento ambiental de datacenters. Nesta nova abordagem utilizou-se as tecnologias mais recentes de CI/CD, permitindo uma instalação que garanta o correto funcionamento do sistema de IoT. Utilizou-se as ferramentas Docker, Git e Ansible no processo de automatização. O código necessário para a replicação deste sistema encontrase disponibilizado na plataforma Github. O sistema de IoT, utilizando containers, mostrou-se capaz de capturar, armazenar e visualizar dados de sensores. Isto foi comprovado com a apresentação gráfica dos valores de temperatura e umidade em um Dashboard específico ou através de uma página web que apresenta esses valores diretamente na planta baixa do datacenter.

REFERÊNCIAS

JUNIOR, Fernando Costa Serra; LIMA, Bruno Seabra Nogueira Mendonça. TECNOLOGIA DOCKER. 2020.

LAUKKANEN, Eero; ITKONEN, Juha; LASSENIUS, Casper. Problems, causes and solutions when adopting continuous delivery - A systematic literature review. 2017.

MADAKAM, Somayya et al. Internet of Things (IoT): A literature review. 2015.

MEIJER, B., HOCHSTEIN, L., MOSER, R. Ansible: Up and Running. "O'Reilly Media, Inc.", 2022.
NASAR, Mohammad; KAUSAR, Mohammad Abu. Suitability of influxdb database for IoT applications. 2019.