

UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES
CAMPUS DE ERECHIM
DEPARTAMENTO DE ENGENHARIAS E CIÊNCIA DA COMPUTAÇÃO
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOÃO VITOR VERONESE VIEIRA
KELWIN KOMKA
VINICIUS EMANOEL ANDRADE

GerenciaDocker:
Sistema para gerenciar contêineres

ERECHIM - RS
2019

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Seção 2	1
2	DESCRIÇÃO DO PROBLEMA	2
2.1	Justificativa	2
3	OBJETIVOS	3
3.1	Objetivo Geral	3
3.2	Objetivos Específicos	3
4	DESCRIÇÃO DA IMPLEMENTAÇÃO	4
4.1	Adaptive DSD	4
4.1.1	Funcionamento	4
5	CONCLUSÃO	5
	REFERÊNCIAS	6

LISTA DE ILUSTRAÇÕES

1 INTRODUÇÃO

Introdução *aqui*.

1.1 Seção 2

Outro Exemplo.

2 DESCRIÇÃO DO PROBLEMA

Em virtude da dinamicidade e agilidade necessárias em tarefas comuns para uma empresa de *software*, tal como a disponibilização de aplicações para clientes ou mesmo a configuração de ambiente para novos colaboradores na equipe de desenvolvimento, criou-se o conceito técnico de *containerização*. De modo resumido, esse conceito pode ser descrito, segundo (FERNANDES, 2018), como “o processo de distribuir uma aplicação de *software* de maneira compartimentada, portátil e autossuficiente”. Isto é, uma forma de criar um ambiente completo de qualquer aplicação desenvolvida e “empacotá-lo”, para posteriormente distribuí-lo e utilizá-lo.

Dentro desse cenário e tendo em vista os diversos benefícios que essa prática traz aos seus utilizadores, diversas ferramentas foram criadas. No entanto, um *software* em específico acabou destacando-se como o mais utilizado quando se deseja implantar essa tecnologia. O ***docker*** permite o gerenciamento completo de todos os *containers* criados e, devido às suas funcionalidades, ganhou notoriedade na comunidade.

No entanto, a utilização diária dessa ferramenta, geralmente realizada através de um terminal, pode se tornar uma tarefa desnecessária e até mesmo complicada, principalmente para um profissional iniciante, pois seu ambiente pode conter diversas especificidades (tal como vários *containers* executando em paralelo) que, quando se está aprendendo a utilizar a ferramenta, podem ser difíceis de serem implementadas.

Além disso, vale ressaltar que, se necessário, o usuário deve controlar a rede (*docker network*) em que os *containers* estão sendo executados, o que acaba gerando ainda mais dificuldades. Com isso, fica nítido que, apesar de ser um recurso que proporciona inúmeras vantagens aos usuários, ainda existe uma barreira de adoção à essa tecnologia, principalmente em um contexto organizacional, pois o profissional que se dispõe a aprender essa nova ferramenta, precisará conciliar esse aprendizado com a realização de todas as demais atividades tradicionais que ele é encarregado.

2.1 Justificativa

Baseando-se nos motivos descritos no capítulo 2 e com o desejo de aprender mais sobre essa moderna ferramenta, o grupo considerou que um monitor *web* que abstraísse essas dificuldades de gerenciamento em uma interface intuitiva e amigável para o usuário seria, além de uma ferramenta útil para os profissionais que se enquadram nessa situação, um bom assunto para ser o tema deste projeto.

3 OBJETIVOS

3.1 Objetivo Geral

- Criar uma ferramenta que possibilite gerenciar os *containers* em execução na máquina

3.2 Objetivos Específicos

- Tornar a ferramenta flexível, permitindo que o usuário escolha o sistema operacional do *contêiner* (com 4 opções)
- Construir uma interface amigável e intuitiva para o usuário
- Executar corretamente o algoritmo (*Adaptive-DSD*), que detectará falha nos *containers*

4 DESCRIÇÃO DA IMPLEMENTAÇÃO

4.1 Adaptive DSD

O Adaptive DSD (*Adaptive Distributed System-Level Diagnosis*) é um algoritmo para diagnóstico em redes completamente conectadas. Onde seu funcionamento é ao mesmo tempo adaptativo e distribuído. Foi desenvolvido para que cada máquina que possua o algoritmo em execução possa realizar o teste e também ser testada por outras máquinas na rede. É caracterizado como adaptativo por não depender e nem restringir o número de máquinas na rede, necessitando, no mínimo uma máquina para o teste. Para a execução dos testes, não é levado em consideração falhas na rede, pois o objetivo deste algoritmo é, testar o processamento ou funcionamento específico de um processo na máquina.

4.1.1 Funcionamento

O algoritmo possui duas listas, que possuem de tamanho o número de máquina conectadas à rede, as listas são: o vetor TESTED_UP, que irá guardar na posição da máquina atual, qual o índice da máquina testada que possui funcionamento normal; o vetor STATE, que armazena o estado das máquinas, tendo inicialmente o valor FALHO para todas e, caso uma máquina tenha seu funcionamento confirmado, esta receberá o valor NORMAL no vetor. A cada rodada o os vetores são atualizados e enviados às outras máquinas na rede.

Na primeira rodada, uma máquina irá iniciar o teste seguindo a lista de máquina existentes e disponíveis na rede. Esta máquina irá percorrer a lista de máquinas fará uma requisição de teste à próxima máquina da lista. Caso a máquina a ser testada retornar uma resposta de funcionamento correto, a máquina que está realizando o teste recebe os dados e envia à máquina testada, que por sua vez irá executar o mesmo processo com a máquina seguinte, até que todas as máquinas tenham sido testadas. Por outro lado, se a máquina testada, retornar algum erro, será marcada como falha, e a máquina que está testando irá testar a próxima máquina da lista, até encontrar outra máquina com funcionamento normal ou até que a lista de máquinas disponíveis acabe.

A segunda rodada será para atualizar as informações de todas as máquinas na rede sobre o estado de funcionamento de cada máquina. Inicialmente a primeira máquina com funcionamento normal irá verificar na lista de máquinas, qual a próxima máquina funcionando, e irá enviar os dados da rede. Ao receber os dados da rede, a máquina receptora irá prosseguir com a distribuição de informações.

5 CONCLUSÃO

Conclusão aqui.

REFERÊNCIAS

FERNANDES, A. **O que é Containerização de aplicação?** 2018. Disponível em: <<https://vertigo.com.br/o-que-e-containerizacao-de-aplicacao/>>. Acesso em: 28 de setembro de 2019. Citado na página 2.