

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

FLÁVIO BORGES DE LIMA

**IMPLANTAÇÃO DE CLUSTER KUBERNETES EM COMPUTADORES COM
PODER COMPUTACIONAL LIMITADO**

GUARAPUAVA

2024

FLÁVIO BORGES DE LIMA

**IMPLANTAÇÃO DE CLUSTER KUBERNETES EM COMPUTADORES COM
PODER COMPUTACIONAL LIMITADO**

Clusters for Computer Network with Limited Computational Power

Proposta de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Hermano Pereira

Coorientador: Prof^a. Dr^a. Sediane Carmem Lunardi Hernandes

GUARAPUAVA

2024



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RESUMO

Este trabalho tem como objetivo a implantação de um cluster Kubernetes em computadores com poder computacional limitado, visando a otimização de recursos e a criação de uma infraestrutura escalável e eficiente para ambientes com hardware modesto. A justificativa para a realização deste estudo se dá pela crescente demanda por soluções de computação distribuída em máquinas com restrições de desempenho, comum em cenários de baixo custo ou reutilização de equipamentos antigos. A metodologia adotada envolve a construção de um cluster com três computadores, sendo um dedicado ao plano de controle e os outros dois como nós, todos interconectados por uma rede local Rede de Área Local, do inglês *local area network* (LAN) com Protocolo de Internet, do inglês *Internet Protocol* (IP) estáticos. O gerenciamento do cluster será feito por um computador adicional, sem sobrecarregar os nós. O estudo também incluirá a realização de testes de desempenho, considerando métricas como o uso de memória Memória de acesso aleatório, do inglês *Random Access Memory* (RAM) e processamento, antes e depois da implantação do cluster e durante a execução de tarefas. Além disso, serão analisadas diferentes versões do Kubernetes e distribuições de Linux para determinar as melhores combinações de configuração para um ambiente de hardware limitado. Espera-se que os resultados obtenham insights valiosos sobre como configurar e otimizar clusters Kubernetes em máquinas com recursos reduzidos, possibilitando uma operação eficiente e escalável. A conclusão do trabalho fornecerá uma base teórica e prática para a aplicação de clusters em hardware com baixa capacidade computacional, destacando as melhores práticas e configurações para garantir a performance adequada em diferentes cenários.

Palavras-chave: kubernetes; clusters; recursos limitados; computação distribuída; otimização de desempenho.

LISTA DE FIGURAS

Figura 1 – Rede a ser criada.	12
Figura 2 – Cluster a ser criado.	13

LISTA DE ABREVIATURAS E SIGLAS

Siglas

CPU	Unidade de Processamento Central, do inglês <i>Central Processing Unit</i>
HA	Alta Disponibilidade, do inglês <i>High Availability</i>
HPC	Computação de Alta Performance, do inglês <i>High-Performance Computing</i>
IBM	Corporação Internacional de Máquinas, do inglês <i>International Business Machines</i>
IoT	Internet das Coisas, do inglês <i>Internet of Things</i>
IP	Protocolo de Internet, do inglês <i>Internet Protocol</i>
LAN	Rede de Área Local, do inglês <i>local area network</i>
NAT	tradução de endereços de rede , do inglês <i>Network address translation</i>
OS	Sistema Operacional, do inglês <i>Operating System</i>
RAM	Memória de acesso aleatório, do inglês <i>Random Access Memory</i>
SSH	"Concha Segura" em tradução direta , do inglês <i>Secure Shell</i>
TCC	Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Objetivos	5
1.1.1	Objetivo geral	5
1.1.2	Objetivos específicos	5
1.2	Justificativa	6
1.3	Estrutura do trabalho	6
2	REFERENCIAL TEÓRICO	7
2.1	Clusters	7
2.2	Contêineres	7
2.2.1	Orquestração de contêineres	8
2.3	Kubernetes	8
2.3.1	Distribuições do Kubernetes	9
2.4	Computadores com poder computacional limitado	9
2.5	Distribuições de Linux	9
3	TRABALHOS RELACIONADOS	11
4	PROPOSTA	12
4.1	Estrutura da rede	12
4.2	Estrutura do cluster	13
4.3	Testes	13
	REFERÊNCIAS	15

1 INTRODUÇÃO

O conceito de clusters tem se tornado fundamental em ambientes que exigem processamento distribuído (MONTEIRO *et al.*, 2021). Um cluster é formado por um conjunto de computadores que trabalham de forma colaborativa, aumentando a capacidade de processamento ao distribuir tarefas entre os vários computadores desse cluster (IBM, 2024). Isso permite que sistemas computacionais com recursos limitados, como computadores antigos, possam ser reaproveitados em configurações que maximizam o desempenho do sistema como um todo.

Aplicações web, por exemplo, podem requerer que o processamento seja distribuído permitindo que o armazenamento e o processamento de dados seja dividido entre os vários computadores em uma rede ou cluster de computadores. De forma mais abrangente, aplicações que seguem o paradigma cliente-servidor ou peer-to-peer podem se beneficiar do uso de clusters de computadores para o processamento distribuído.

Com tecnologias como containers, que isolam aplicações e garantem a portabilidade entre diferentes sistemas, e o Kubernetes, que orquestra e gerencia esses containers (CNCF, 2024b), torna-se possível utilizar clusters Kubernetes para diversas finalidades. Kubernetes pode ser utilizado de diferentes formas, como mencionado em (MONTEIRO *et al.*, 2021), p. 154, "em seu cluster próprio (de forma local), em um sistema de arquitetura híbrida ou até mesmo em qualquer provedor de computação na nuvem pública.". Um cluster de computadores é considerado um sistema de arquitetura híbrido. Logo, um cluster Kubernetes pode ser implantado considerando essa arquitetura.

Desta forma, este projeto busca explorar como um cluster Kubernetes pode ser implementado em máquinas com poder computacional limitado permitindo que seus usuários aproveitem o sistema para qualquer tipo de aplicação desejada, maximizando o uso dos recursos disponíveis.

1.1 Objetivos

Os objetivos deste trabalho se dividem em objetivos gerais e específicos.

1.1.1 Objetivo geral

Implantar um cluster Kubernetes utilizando computadores com poder computacional limitado.

1.1.2 Objetivos específicos

Como objetivos específicos deste trabalho tem-se:

- Montar uma rede de computadores utilizando computadores com capacidade computacional limitada.
- Implantar um cluster Kubernetes na rede de computadores criada.
- Avaliar o desempenho do cluster Kubernetes criado, considerando suas limitações computacionais, utilizando uma aplicação simples que requeira processamento distribuído.
- Documentar o processo de configuração e implantação do cluster Kubernetes.

1.2 Justificativa

Com o avanço da tecnologia, muitos computadores tornam-se obsoletos e são descartados, apesar de ainda possuírem capacidade de processamento significativa para determinados tipos de aplicações. A necessidade de soluções econômicas e sustentáveis para reaproveitar esses recursos é crescente, especialmente em ambientes acadêmicos, onde o acesso a hardware de alto desempenho é limitado.

A implementação de clusters Kubernetes em computadores com poder computacional limitado, os quais podem ser agrupados em um cluster de computadores, oferece uma alternativa sustentável para o uso desses equipamentos. Isso permite que através da distribuição de carga de trabalho, esses computadores trabalhem de forma colaborativa, aumentando sua eficiência. Além disso, o uso de containers e orquestração com Kubernetes proporciona flexibilidade e escalabilidade, permitindo que o cluster seja utilizado para vários fins, desde projetos acadêmicos até experimentos práticos.

Assim, este projeto se justifica, portanto, pela relevância de explorar soluções tecnológicas sustentáveis, reutilizando hardware ultrapassado, e pela necessidade de oferecer um ambiente prático, acessível e adaptável para estudantes e profissionais que desejam testar, desenvolver ou implementar suas aplicações em um cluster de computadores de baixo custo.

1.3 Estrutura do trabalho

O trabalho está dividido como segue. O primeiro capítulo é a Introdução, em que o problema é contextualizado e são apresentados os objetivos do projeto, justificativa para o seu desenvolvimento e a estrutura geral do trabalho. O segundo capítulo é o Referencial Teórico, o qual aborda os principais conceitos e tecnologias utilizados no projeto, como clusters, containers, Kubernetes e Docker, além de discutir os desafios de uso de hardware ultrapassado em ambientes distribuídos. O capítulo 3 descreve os trabalhos relacionados e no capítulo 4 a proposta deste TCC é apresentada. Por fim, tem-se as referências bibliográficas.

2 REFERENCIAL TEÓRICO

Neste capítulo, serão apresentados os conceitos fundamentais relacionados ao desenvolvimento do trabalho, incluindo clusters, contêineres, orquestração de containers e Kubernetes, com foco na implementação de clusters Kubernetes em máquinas com poder computacional limitado.

2.1 Clusters

Segundo a Corporação Internacional de Máquinas, do inglês *International Business Machines* (IBM), clusters são aglomerados de computadores que trabalham juntos para executar tarefas, distribuindo a carga de trabalho entre os computadores que formam a cluster(IBM, 2024).

Cada computador em um cluster é chamado de nó (*node*), e cada nó é formado por um Sistema Operacional, do inglês *Operating System* (OS) e um *middleware*, que é responsável por gerenciar a comunicação e a execução de tarefas no cluster. A estrutura de um cluster pode variar entre um único nó até milhares de nós, geralmente conectados por uma LAN (IBM, 2024).

Os clusters podem ser divididos em dois grupos principais:

- Computação de Alta Performance, do inglês *High-Performance Computing* (HPC): Projetados para executar tarefas que precisam de um grande poder computacional, como simulações científicas ou processamento de grande volume de dados.
- Alta Disponibilidade, do inglês *High Availability* (HA): Desenvolvidos para garantir que os serviços e recursos permaneçam acessíveis, mesmo em caso de falhas em um ou mais nós, geralmente sendo utilizados na hospedagem de aplicações.

Neste Trabalho de Conclusão de Curso (TCC), será utilizado um cluster de HA, gerenciado pelo Kubernetes, que será explorado em capítulos posteriores. A escolha do Kubernetes foi feita com base na facilidade de adicionar novos nós (CNCF, 2024b) e na grande flexibilidade fornecida pelos contêineres (CLOUD, 2024a), que são gerenciados pelo Kubernetes.

2.2 Contêineres

Como descrito por (CLOUD, 2024a), "os contêineres são pacotes de software que contêm todos os elementos necessários para serem executados em qualquer ambiente". Os contêineres utilizam o mesmo kernel que a máquina hospedeira e, além disso, podem compartilhar camadas de sistema de arquivos, o que permite que múltiplos contêineres usem as mesmas dependências sem a necessidade de duplicá-las, poupando assim recursos computacionais como memória e armazenamento.

Algumas características fundamentais dos contêineres incluem a fácil e confiável recriação de um mesmo contêiner, o bom isolamento entre as aplicações que estão sendo executadas dentro de contêineres distintos em uma mesma máquina, e a leveza em comparação com máquinas virtuais (CLOUD, 2024a). O isolamento é uma grande vantagem dos contêineres, garantindo que as aplicações executadas em contêineres diferentes não interfiram uma nas outras, apesar de estarem rodando no mesmo Sistema Operacional e compartilhando o mesmo kernel. Isso oferece segurança, evita conflitos de dependências entre os diferentes ambientes de execução e gera uma facilidade em hospedar aplicações completamente distintas em um mesmo ambiente (CLOUD, 2024a).

Essas características tornam os contêineres ideais para maximizar a utilização do hardware disponível. Graças à sua leveza e flexibilidade, é possível executar múltiplos contêineres em uma única máquina ou distribuir a carga de trabalho entre contêineres alocados em diferentes máquinas, otimizando a eficiência e a escalabilidade dos recursos computacionais.

2.2.1 Orquestração de contêineres

De acordo com o Google Cloud (CLOUD, 2024b), a orquestração de contêineres refere-se à automação do gerenciamento de recursos e do ciclo de vida dos contêineres. Essa prática desempenha um papel crucial na garantia da eficiência e escalabilidade em ambientes com múltiplos contêineres, seja em sistemas independentes ou operando em clusters.

Neste TCC, o Kubernetes será empregado como a principal ferramenta de orquestração. Sua implementação permitirá a gestão eficiente dos recursos computacionais disponíveis, bem como a distribuição equilibrada da carga de trabalho entre as máquinas do cluster, maximizando a utilização de hardware limitado.

2.3 Kubernetes

O Kubernetes, também conhecido como k8s, é uma plataforma de orquestração de contêineres desenvolvida pela Google, a qual foi projetada para automatizar a implantação, o dimensionamento e a operação de contêineres. Ele permite a criação de clusters de HA, compostos por um ou mais planos de controle e um ou mais nós (CNCF, 2024a).

Os planos de controle são responsáveis por manter o estado desejado do cluster dentro de parâmetros definidos pelo administrador, como a quantidade de cópias de um contêiner de uma aplicação ou a escolha do nó mais adequado para executar uma aplicação, avaliando os recursos disponíveis (CNCF, 2024a).

Os nós são responsáveis por executar um ou mais pods. Os pods são as menores unidades de trabalho no Kubernetes, e cada um pode conter um ou mais contêineres que compartilham recursos como rede e armazenamento (CNCF, 2024c).

No contexto deste projeto, o Kubernetes desempenhará um papel fundamental na distribuição da carga de trabalho entre as máquinas do cluster, garantindo o uso eficiente do hardware disponível e maximizando o aproveitamento dos recursos computacionais.

2.3.1 Distribuições do Kubernetes

Assim como outros diversos projetos de código aberto o Kubernetes tem diversas variações que foram criadas com base no projeto. E algumas dessas variações se especializaram na diminuição do consumo de recursos como memória RAM e processamento da Unidade de Processamento Central, do inglês *Central Processing Unit* (CPU), podendo serem citadas:

- O *k3s*, desenvolvido para ambientes de Internet das Coisas, do inglês *Internet of Things* (IoT), é otimizado para ser implantado em uma ampla gama de arquiteturas de sistemas (LABS, 2024).
- O *MicroK8s* é uma versão simplificada e otimizada do Kubernetes, projetada para permitir a criação de clusters com facilidade e rapidez (CANONICAL, 2024).
- O *k0s* é uma variante *bare metal* do Kubernetes, ideal para ser utilizado em IoT, com flexibilidade para ser implantado em diversos ambientes (K0S PROJECT, 2024).

Neste TCC as diferentes variantes do Kubernetes serão implantadas nos sistemas operacionais disponíveis e escolhidos na seção 2.5. Após implementação do cluster, serão realizados testes para avaliar o consumo individual de recursos de RAM e CPU dos computadores, com o objetivo de comparar a performance de cada variação do Kubernetes em máquinas com recursos computacionais limitados.

2.4 Computadores com poder computacional limitado

O conceito de computadores com capacidade computacional limitada pode variar dependendo do contexto de uso. Para os propósitos deste TCC, define-se como máquinas com recursos modestos em relação aos requisitos típicos de aplicações modernas. Nesta definição, incluem-se computadores que possuem entre 2 e 4 gigabytes de memória RAM e processadores com 1 à 4 núcleos de CPU, características que restringem sua capacidade de executar tarefas intensivas em recursos.

2.5 Distribuições de Linux

O Linux é OS de código aberto (THE LINUX FOUNDATION, 2024), e como diversos projetos de código aberto, existem diversas distribuições de Linux, cada uma com suas característi-

cas. Como todo OS, o Linux administra os recursos computacionais da máquina, o que tem um grande impacto no desempenho do mesmo.

No contexto desse trabalho serão utilizadas versões mais leves do Linux, em que recursos como interface gráfica e aplicativos utilitários não estarão presentes, uma vez que não são necessários para o funcionamento do cluster. Porém, ainda será necessário que o OS seja compatível com as versões do Kubernetes e tenha uma ampla documentação. Sendo assim, duas escolhas se destacam:

- Linux Alpine: o Linux Alpine foi projetado para ser simples, seguro e eficiente, conseguindo operar nos mais diversos ambientes sem consumir uma grande quantidade de recursos computacionais (ALPINE LINUX, 2024).
- Debian: o Debian é a base para algumas das distribuições mais utilizadas de Linux, incluindo o Ubuntu. Isso faz com que o Debian tenha uma imensa compatibilidade de pacotes, extensa documentação e comunidade bastante ativa (DEBIAN, 2024).

Neste trabalho será feita uma comparação entre as duas distribuições de Linux executando as variantes do Kubernetes escolhidas na subseção 2.3.1 disponíveis para o OS, os cenários escolhidos foram: antes da implantação do cluster, depois da implantação do cluster e com o cluster implantado durante a execução de tarefas.

3 TRABALHOS RELACIONADOS

Diversos estudos abordam a utilização de Kubernetes em ambientes com recursos limitados, destacando-se pela adaptação de arquiteturas para permitir a operação eficiente em hardware de baixo custo. Estes trabalhos são relevantes para este TCC, pois apresentam alternativas e abordagens que se alinham com a proposta de implantar um cluster Kubernetes em computadores com poder computacional limitado.

Um estudo realizado por Silva (SILVA, 2022), aborda a implementação de soluções computacionais em ambientes com hardware modesto utilizando tecnologias como o Kubernetes para otimização de recursos em sistemas com capacidade computacional limitada. Esse trabalho é particularmente interessante, pois explora as mesmas questões relacionadas ao uso de clusters Kubernetes em máquinas de baixo custo, oferecendo insights sobre os desafios de desempenho e eficiência.

Além disso, o trabalho de *Programming Group* (GROUP, 2023) explora diferentes distribuições de Kubernetes otimizadas para ambientes com recursos limitados, como o k3s, k0s e microk8s, destacando a importância dessas versões leves para a implementação de clusters em hardware modesto. Essa pesquisa complementa a proposta deste TCC, já que o uso de distribuições otimizadas pode ser uma solução crucial para garantir o funcionamento eficiente do cluster, mesmo em máquinas com pouca memória RAM e capacidade de processamento.

Outro estudo relevante é o de *Learn Fast Make Things* (THINGS, 2023), que discute como transformar hardware antigo em clusters Kubernetes, utilizando recursos limitados de maneira eficiente. Esse trabalho se aproxima diretamente da proposta deste projeto, ao sugerir métodos de aproveitamento de hardware obsoleto para a criação de clusters, alinhando-se à proposta de utilizar máquinas com poder computacional limitado para implementação de soluções escaláveis.

Esses trabalhos fornecem uma base teórica e prática sólida, que orienta a escolha de tecnologias e estratégias para a implementação de clusters Kubernetes em hardware com recursos limitados, contribuindo significativamente para o desenvolvimento deste TCC.

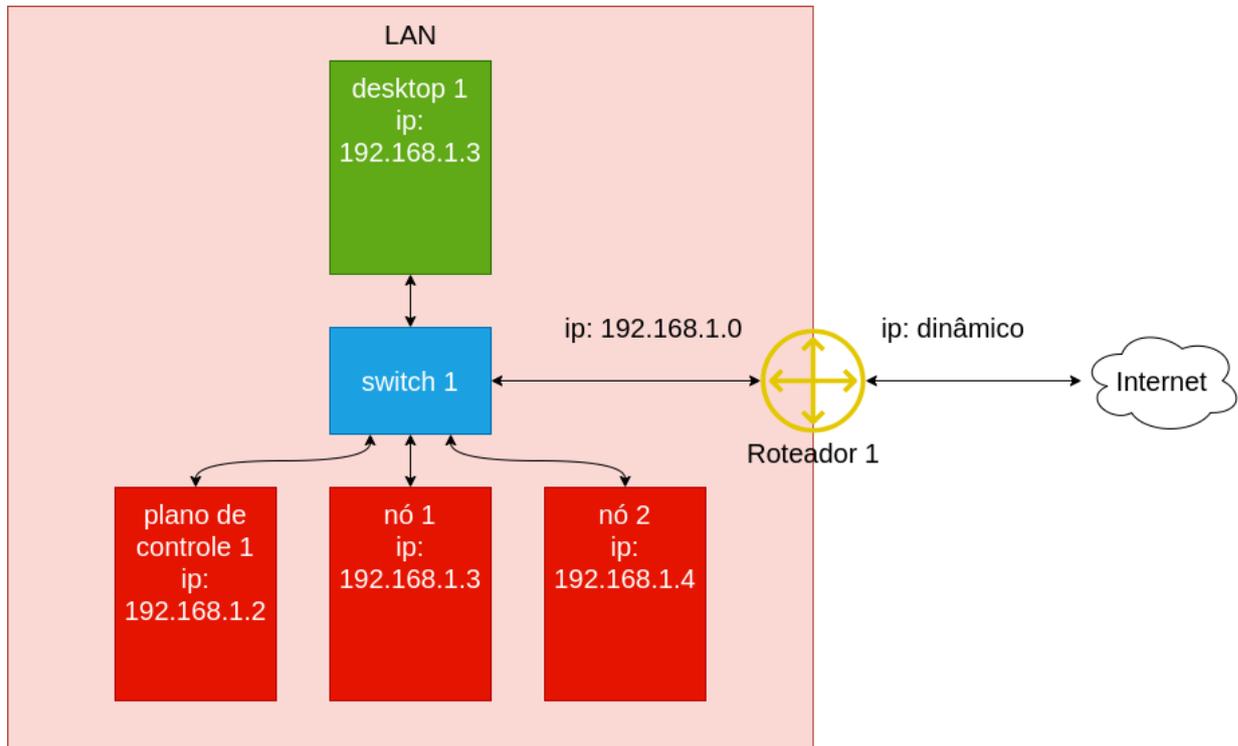
4 PROPOSTA

Nesta seção, será descrita a estrutura planejada para o cluster a ser implementado, incluindo os componentes de *hardware* e *software*, a configuração de rede e estrutura do cluster, além de possíveis métricas e testes.

4.1 Estrutura da rede

O cluster será composto por três computadores conectados por uma LAN configurada com endereços IP estáticos. A atribuição de IP estático garante estabilidade e previsibilidade nas conexões entre os computadores, características fundamentais para o funcionamento adequado de clusters.

Figura 1 – Rede a ser criada.



Fonte: Autoria própria (2024).

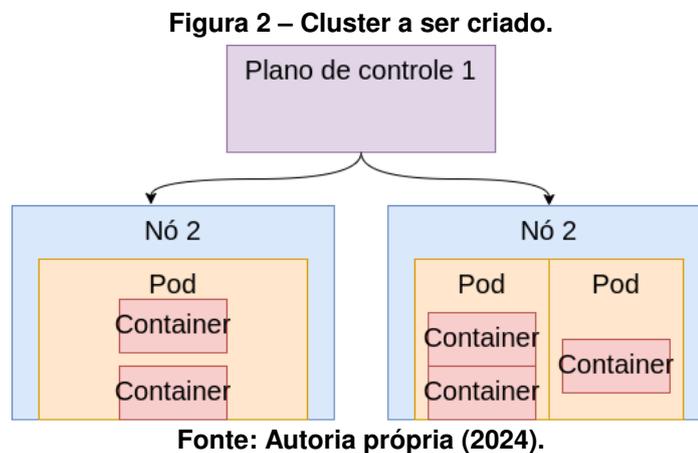
A Figura 1 apresenta a configuração de rede desejada. Os computadores em vermelho (plano de controle 1, nó 1 e nó 2) que estão marcados com um possível IP, que será atribuído para eles manualmente, são os computadores que vão formar o cluster, o qual está previsto para ser constituído por um plano de controle e dois nós. O plano de controle necessita de mais poder computacional do que os nós, então será selecionada a melhor máquina disponível. Esse computador já foi selecionado e possui 4 gigabytes de RAM e 4 núcleos de processador.

O computador destacado em verde (desktop 1) será utilizado para gerenciar o cluster. Ele estará configurado para acessar os nós por meio do "Concha Segura" em tradução direta

, do inglês *Secure Shell* (SSH), permitindo a configuração, coleta de métricas e realização de testes sem sobrecarregar os recursos computacionais das máquinas do cluster. Este computador não faz parte do cluster propriamente dito e poderá rodar qualquer sistema operacional, preferencialmente distribuições Linux como o Ubuntu.

A interligação dos componentes será realizada por meio de um switch (em azul), que conectará todos os dispositivos da LAN entre si e ao roteador (em amarelo). O roteador terá a função de fornecer acesso à Internet, essencial para o download de imagens de contêineres e outras operações dependentes da web, utilizando tradução de endereços de rede, do inglês *Network address translation* (NAT).

4.2 Estrutura do cluster



A Figura 2 é uma representação do cluster desconsiderando fatores como a rede e focando somente nas estruturas mais abstratas de uma cluster Kubernetes. Ou seja, o plano de controle, os nós, os pods e os contêineres.

O plano de controle será dedicado exclusivamente à administração dos recursos do cluster, incluindo o agendamento de cargas de trabalho, gerenciamento de estado e controle de nós.

Os pods e contêineres irão variar de teste para teste e de aplicação em aplicação, mas cada pod sempre vai ter um ou mais container e um nó pode ter vários pods ou nenhum pod.

4.3 Testes

Os testes planejados visam avaliar o desempenho e a eficiência do cluster em diferentes cenários: antes da implantação do cluster, depois da implantação do cluster e com o cluster implantado durante a execução de tarefas. Métricas como o uso de RAM e CPU serão coletadas em cada cenário.

Além disso, serão exploradas diferentes versões do Kubernetes, sempre que disponível, em conjunto com diversas distribuições de Linux. Isso permitirá a realização de comparações entre configurações distintas, avaliando o impacto das variações no desempenho do cluster ao executar as mesmas tarefas.

Esses testes fornecerão uma visão abrangente sobre a performance do cluster em máquinas com recursos computacionais limitados, ajudando a identificar as melhores combinações de configurações para esses ambientes.

REFERÊNCIAS

- ALPINE LINUX. **About | Alpine Linux**. 2024. Accessed: 20 Nov. 2024. Disponível em: <https://alpinelinux.org/about/>.
- CANONICAL. **MicroK8s Documentation**. 2024. Disponível em: <https://microk8s.io/docs>. Acesso em: 18 nov. 2024.
- CLOUD G. **What Are Containers?** 2024. Disponível em: <https://cloud.google.com/learn/what-are-containers>. Acesso em: 6 nov. 2024.
- CLOUD G. **What is container orchestration?** 2024. Disponível em: <https://cloud.google.com/discover/what-is-container-orchestration?hl=en>. Acesso em: 18 nov. 2024.
- CNCF. **Cluster Architecture**. Online, 2024. Disponível em: <https://kubernetes.io/docs/concepts/architecture/>. Acesso em: 18 nov. 2024.
- CNCF . **Overview**. Online, 2024. Disponível em: <https://kubernetes.io/docs/concepts/overview/>. Acesso em: 18 nov. 2024.
- CNCF . **Pods**. Online, 2024. Disponível em: <https://kubernetes.io/docs/concepts/workloads/pods/>. Acesso em: 20 nov. 2024.
- DEBIAN. **Debian – Reasons to use Debian**. 2024. Accessed: 20 Nov. 2024. Disponível em: https://www.debian.org/intro/why_debian.
- GROUP, P. **Lightweight Kubernetes Distributions**. 2023. Disponível em: https://programming-group.com/assets/pdf/papers/2023_Lightweight-Kubernetes-Distributions.pdf. Acesso em: 20 nov. 2024.
- IBM. **What is Cluster Computing?** 2024. Disponível em: <https://www.ibm.com/think/topics/cluster-computing>. Acesso em: 18 nov. 2024.
- K0S PROJECT. **K0s Documentation**. 2024. Disponível em: <https://k0sproject.io/>. Acesso em: 20 nov. 2024.
- LABS R. **K3s Documentation**. 2024. Disponível em: <https://k3s.io/>. Acesso em: 20 nov. 2024.
- MONTEIRO, E. R. *et al.* **DevOps**. SAGAH, 2021. 154 p. ISBN 9786556901725. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9786556901725/>. Acesso em: 6 nov. 2024.
- SILVA, J. **Título do Trabalho**. 2022. Trabalho de Conclusão de Curso (Curso de Ciência da Computação) – Universidade Federal de Pernambuco, Recife. Disponível em: https://www.cin.ufpe.br/~tg/2022-1/tg_CC/tg_pgr.pdf. Acesso em: 20 nov. 2024.
- THE LINUX FOUNDATION. **What is Linux?** 2024. Disponível em: <https://www.linux.com/what-is-linux/>. Acesso em: 20 nov. 2024.
- THINGS, L. F. M. **How to turn your old hardware into a Kubernetes cluster**. 2023. Disponível em: <https://learnfastmakethings.com/p/how-to-turn-your-old-hardware-into-a-kubernetes-cluster-129d17aa8704>. Acesso em: 20 nov. 2024.