



**UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
CURSO DE GRADUAÇÃO EM REDES DE COMPUTADORES**

LUIZ ALVES DE AGUIAR FILHO

**KLE: UMA FERRAMENTA PARA CRIAÇÃO DE AMBIENTES DE APRENDIZADO
PARA O KUBERNETES**

QUIXADÁ

2025

LUIZ ALVES DE AGUIAR FILHO

KLE: UMA FERRAMENTA PARA CRIAÇÃO DE AMBIENTES DE APRENDIZADO PARA
O KUBERNETES

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Redes de Computadores
do da Universidade Federal do Ceará
Campus Quixadá, como requisito parcial à
obtenção do grau de tecnólogo em Redes de
Computadores.

Orientador: Prof. Dr. João Marcelo Uchôa de
Alencar.

QUIXADÁ

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A23k Aguiar Filho, Luiz Alves de.

KLE: uma ferramenta para criação de ambientes de aprendizado para o Kubernetes / Luiz Alves de Aguiar Filho. – 2025.

39 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2025.

Orientação: Prof. Dr. João Marcelo Uchôa de Alencar.

1. Educação em TI. 2. Kubernetes. 3. Contêineres. 4. Automação. 5. Nuvem. I. Título.

CDD 004.6

LUIZ ALVES DE AGUIAR FILHO

**KLE: UMA FERRAMENTA PARA CRIAÇÃO DE AMBIENTES DE APRENDIZADO PARA
O KUBERNETES**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Redes de Computadores
do da Universidade Federal do Ceará
Campus Quixadá, como requisito parcial à
obtenção do grau de tecnólogo em Redes de
Computadores.

Aprovada em: 27 fevereiro 2025

BANCA EXAMINADORA

Prof. Dr. João Marcelo Uchôa de
Alencar (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Michel Sales Bonfim
Universidade Federal do Ceará - (UFC)

Prof. Dr. Jeandro de Mesquita Bezerra
Universidade Federal do Ceará - (UFC)

Agradeço a Deus, minha mãe, meus irmãos, namorada e familiares que sempre me motivaram e me deram apoio.

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por ter me concedido forças para persistir e continuar tentando, mesmo quando nada parecia dar certo. Não foi fácil. Logo após a primeira semana de aula do meu curso, veio a pandemia, que me privou de muitas vivências que poderiam ter enriquecido ainda mais minha formação profissional. Após o retorno das aulas, percorria todas as noites pouco mais de 40 km, perdendo constantemente o início e o fim das aulas para conseguir pegar o ônibus.

Gostaria de expressar minha gratidão à minha mãe, Marlene Venâncio, que sempre me apoiou e incentivou, sendo, em muitos momentos, minha motivação para não desistir. Mesmo com suas limitações financeiras, ela me proporcionou os meios necessários para concluir este curso. Aos meus irmãos, em especial à minha irmã Rayane Kevina, que sempre esteve ao meu lado e foi minha conselheira nos momentos turbulentos que enfrentei. À minha namorada, Raionara Laurindo, que, mesmo antes de nosso relacionamento, esteve presente, fazendo-me acreditar que eu seria capaz.

Agradeço a todos os professores que integraram esta jornada, servindo como fontes de inspiração constante.

Meus profundos agradecimentos ao meu orientador, Dr. João Marcelo Uchôa de Alencar, por dedicar seu tempo à minha orientação e por esclarecer minhas dúvidas. Admiro-o profundamente, não apenas como orientador, mas também como um dos melhores docentes que tive a honra de conhecer. Muito obrigado.

Por fim, agradeço aos colegas que tive o prazer de conhecer durante essa trajetória: pelas noites de videochamadas para estudos, pelas conversas sobre o curso, pelas risadas e momentos divertidos. Não mantive contato com todos, mas cada um de vocês foi fundamental para minha jornada.

RESUMO

Atualmente, um número significativo de empresas utiliza contêineres em suas plataformas de desenvolvimento e implantação de aplicações. Com o crescimento exponencial no uso dessas tecnologias, torna-se imprescindível a adoção de sistemas especializados de gerenciamento – cenário em que o Kubernetes consolidou-se como principal solução do mercado, sendo amplamente empregado por organizações globais. Essa demanda por expertise na tecnologia revela a necessidade de capacitação profissional, especialmente em ambientes acadêmicos. Entretanto, a heterogeneidade e defasagem de infraestrutura física nas instituições de ensino dificultam a prática didática, limitando o acesso a recursos adequados para estudos práticos. Diante desse desafio, a computação em nuvem surge como alternativa viável, oferecendo infraestrutura padronizada e escalável. No entanto, a configuração manual de clusters Kubernetes nesses ambientes envolve processos complexos e repetitivos, exigindo conhecimentos técnicos que aumentam a curva de aprendizado. Para resolver essa lacuna, propôs-se o desenvolvimento de uma ferramenta de linha de comando (CLI) que automatiza a criação de clusters Kubernetes em múltiplas nuvens públicas, reduzindo custos e simplificando o provisionamento de ambientes educacionais. A solução foi implementada utilizando tecnologias como Terraform (para provisionamento de recursos), Ansible (automação de configurações) e kubeadm (inicialização do cluster), integrando-se a provedores de nuvem como AWS e Azure por meio de programas educacionais (AWS Academy e Azure for Students), que garantem acesso gratuito a recursos. Para validação, conduziu-se um experimento online onde primeiramente foram recrutados participantes via questionário (Google Forms) que testaram a ferramenta em conjunto com um cenário de estudos práticos. Os resultados, coletados por meio de novo questionário, indicaram alta satisfação dos participantes do teste. Como resultado final, obteve-se uma ferramenta funcional, disponibilizada em repositório público no GitHub, que permite a criação ágil de clusters Kubernetes em ambientes educacionais, eliminando custos e barreiras técnicas. A solução demonstra ser um recurso valioso para instituições de ensino, alinhando-se às demandas por capacitação prática em tecnologias emergentes.

Palavras-chave: contêineres; Kubernetes; nuvem; IaC; automação; educação em TI.

ABSTRACT

Currently, a significant number of companies use containers in their application development and deployment platforms. With the exponential growth in the adoption of these technologies, the implementation of specialized management systems has become essential—a scenario where Kubernetes has emerged as the leading market solution, widely adopted by global organizations. This demand for expertise in the technology highlights the need for professional training, particularly in academic settings. However, the heterogeneity and obsolescence of physical infrastructure in educational institutions hinder hands-on teaching practices, limiting access to adequate resources for practical studies. Faced with this challenge, cloud computing emerges as a viable alternative, offering standardized and scalable infrastructure. However, the manual configuration of Kubernetes clusters in these environments involves complex and repetitive processes, requiring technical knowledge that steepens the learning curve. To address this gap, a command-line interface (CLI) tool was proposed to automate the creation of Kubernetes clusters across multiple public clouds, reducing costs and simplifying the provisioning of educational environments. The solution was implemented using technologies such as Terraform (for resource provisioning), Ansible (configuration automation), and kubeadm (cluster initialization), integrated with cloud providers like AWS and Azure through educational programs (AWS Academy and Azure for Students), which provide free access to resources. For validation, an online experiment was conducted where participants were first recruited via a questionnaire (Google Forms) to test the tool alongside a practical study scenario. Results, collected through a follow-up questionnaire, indicated high satisfaction among test participants. As a final outcome, a functional tool was developed and made available in a public GitHub repository, enabling rapid creation of Kubernetes clusters in educational environments while eliminating costs and technical barriers. The solution proves to be a valuable resource for academic institutions, aligning with the growing demand for practical training in emerging technologies.

Keywords: containers; Kubernetes; cloud; IaC; automation; IT education.

LISTA DE FIGURAS

Figura 1 – Virtualização vs Contêiners	12
Figura 2 – Arquitetura do Kubernetes	14
Figura 3 – Fluxograma da Metodologia	21
Figura 4 – Arquitetura da Ferramenta	22
Figura 5 – CLI da Ferramenta	23
Figura 6 – CLI para escolha do provedor	23
Figura 7 – Exemplo de arquivo Terraform	24
Figura 8 – Playbook Ansible.	25
Figura 9 – Cenário de Estudos.	26
Figura 10 – Questão 1 - Qual curso você estuda??	28
Figura 11 – Questão 2 - Qual seu semestre atual?	28
Figura 12 – Questão 3 - Você possui acesso a uma máquina Linux?	29
Figura 13 – Questão 1 - Qual provedor de nuvem você utilizou?	29
Figura 14 – Questão 2 - Você já possuía alguma experiência prévia com o Kubernetes? .	30
Figura 15 – Questão 3 - Você já utilizou alguma ferramenta de provisionamento de recursos?	30
Figura 16 – Questão 4 - Você conseguiu executar a ferramenta?	31
Figura 17 – Questão 7 - Você conseguiu realizar o cenário de estudos?	31
Figura 18 – Questão 5 - Em que escala você avalia o nível de dificuldade em utilizar a ferramenta?	32
Figura 19 – Questão 6 - Em que escala está seu nível de satisfação com a ferramenta? .	32
Figura 20 – Questão 8 - Em que escala houve bugs durante a utilização da ferramenta? .	33
Figura 21 – Questão 10 - Como você avalia o tempo de criação da infraestrutura pela ferramenta?	33
Figura 22 – Questão 9 - Você acredita que a ferramenta facilitaria seus estudos de Kubernetes?	34
Figura 23 – Questão 13 - Em que escala você indicaria esta ferramenta para alguém está querendo estudar o Kubernetes?	34
Figura 24 – Questão 11 - Você considerou a descrição de utilização da ferramenta confusa? .	35
Figura 25 – Questão 12 - Você considerou a descrição do cenário de estudo confusa? . .	35

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	10
1.1.1	<i>Objetivo geral</i>	10
1.1.2	<i>Objetivos específicos</i>	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Contêiner	12
2.2	Kubernetes	13
2.3	Infraestrutura como Código	16
3	TRABALHOS RELACIONADOS	18
3.1	On-demand virtual research environments using microservices	18
3.2	Kubitect - a Solution for On-premise Cluster Deployment	18
3.3	DVL : Uma Ferramenta para Criação de Laboratórios Práticos de Disciplinas da Área de TI Utilizando Virtualização Baseada em Contêineres	19
3.4	Análise Comparativa.	20
4	PROCEDIMENTOS METODOLÓGICOS	21
4.1	Levantamento dos provedores de nuvem.	21
4.2	Arquitetura da solução	22
4.2.1	<i>Interface de linha de comando</i>	22
4.2.2	<i>Arquivo Terraform</i>	23
4.2.3	<i>Playbook Ansible</i>	24
4.3	Roteiro de Estudo	25
5	EXPERIMENTOS	27
5.1	Descrição dos Experimentos	27
5.2	Analise do questionário de inscrição	27
5.3	Analise do questionário de avaliação da ferramenta	29
5.4	Discussão Final	34
6	CONCLUSÃO	37
	REFERÊNCIAS	38

1 INTRODUÇÃO

Contêiner é o agrupamento de uma aplicação juntamente com suas dependências, que compartilham o *kernel* do sistema operacional do *host*, ou seja, da máquina (virtual ou física) na qual está executando (Vitalino; Castro, 2018). A utilização de contêineres traz consigo uma série de benefícios como portabilidade, melhor utilização de recursos, facilidade de simulação de ambientes, entre outros. Devido a esses e outros benefícios, várias empresas têm adotado contêineres como plataforma de desenvolvimento e implantação de suas aplicações.

Quando uma aplicação necessita de muitos contêineres para sua execução, seu gerenciamento se torna bastante complexo, desse modo surgem os orquestradores de contêineres. O Kubernetes é um sistema de orquestração de contêineres de código aberto, que se tornou o padrão *de facto* da indústria. Ele realiza o gerenciamento de aplicações conteinerizadas em vários *hosts* para implantação, monitoramento e escalabilidade de contêineres. Originalmente criado pelo Google, foi doado em março de 2016 para a *Cloud Native Computing Foundation* (CNCF). (Kubernetes, 2019)

Para organizações com um número crescente de serviços e aplicações, a gerência de *clusters* Kubernetes torna-se uma atividade constante (Muschko, 2022). De acordo com *CNCF Annual Survey 2021* (CNCF, 2021) 96% das instituições ouvidas na pesquisa estão usando ou avaliando o Kubernetes, e adicionalmente 93% dos respondentes estão atualmente usando ou planejando usar contêineres em produção.

Sendo assim, o estudo de tais ferramentas pode ser visto como bastante proveitoso e importante, uma vez que há uma demanda considerável por profissionais que possuam conhecimento sobre tais tecnologias. Desse modo, seu ensino no ambiente universitário deve ser fomentado, porém, os professores enfrentam uma dificuldade significativa no que diz respeito a grande heterogeneidade e defasagem de *hardware* disponível para os estudantes (SILVA *et al.*, 2018). A nuvem surge como alternativa, onde o provedor de nuvem ficará responsável pelo gerenciamento de toda a infraestrutura física e camada de virtualização, oferecendo aos usuários uma mesma infraestrutura final, que poderá ser adquirida como serviço pelos usuários de acordo com suas necessidades (Mušić; Fortuna, 2022).

Provedores de nuvem promovem períodos de testes gratuitos e outros benefícios para estudantes, como o projeto AWS Academy¹. No entanto, tal cenário traz consigo a necessidade do estudante aprender a utilizar o ambiente de nuvem escolhido para poder realizar o *deploy* do

¹ <https://aws.amazon.com/pt/training/awacademy/>

ambiente Kubernetes, caso ele queira utilizar mais de um ambiente para aproveitar os benefícios oferecidos por estes, ele terá que realizar esse processo várias vezes, além disso, embora a implantação seja parte importante do processo de aprendizagem do Kubernetes, não é o único aspecto a ser estudado. Após o aluno ter aprendido o processo de implantação do Kubernetes, ele comece o estudo dos outros aspectos, como *workloads*, *scheduling*, *services*, resolução de problemas e outros. É necessário que ele possua uma maneira fácil e rápida de implantar o ambiente no qual realizara os estudos, de modo que diminua o esforço repetitivo, reduzindo a chance de erros e maximizando o tempo de estudo. Nesse contexto vislumbramos algumas questões de pesquisa.

- Entre as opções de provedores de nuvem, quais são as melhores opções a serem utilizadas pelo professor e estudantes para o aprendizado de Kubernetes?
- É possível desenvolver uma ferramenta de linha de comando, de fácil interação, que automatize a implantação de *clusters* Kubernetes em múltiplas nuvens públicas, garantindo usabilidade satisfatória para alunos e professores em ambientes educacionais?

Nesse sentido, o objetivo do trabalho é propor uma ferramenta para a criação de ambientes virtuais de aprendizado para Kubernetes em provedores de nuvem distintos e que seja de fácil utilização pelo usuário, visando as melhores opções de custo disponíveis entre os provedores de nuvem a serem analisados no decorrer do projeto.

1.1 Objetivos

1.1.1 *Objetivo geral*

O objetivo do trabalho é propor uma ferramenta de linha de comando para a criação de ambientes virtuais de aprendizado para Kubernetes que possa ser utilizada para prover a infraestrutura em múltiplas nuvens públicas a serem escolhidas pelo usuário, visando à redução de custos e facilidade de implantação do *cluster* e acesso a benefícios fornecidos pelo provedor de nuvem.

1.1.2 *Objetivos específicos*

- a) Realizar o levantamento de plataformas de nuvem que ofereçam períodos de testes gratuitos.
- b) Projetar e desenvolver a ferramenta baseado nas plataformas de nuvem levantadas anteriormente.

ormente, para automação da implantação do *cluster* Kubernetes que será utilizado para estudos.

- c) Realizar uma avaliação de usabilidade da ferramenta desenvolvida.

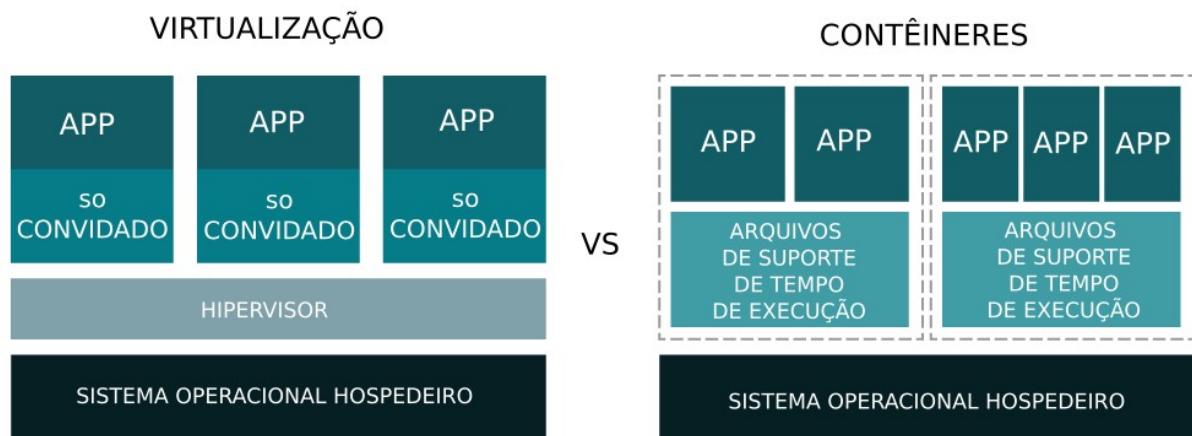
2 FUNDAMENTAÇÃO TEÓRICA

2.1 Contêiner

Um contêiner é o empacotamento de uma aplicação juntamente com suas dependências, executando por meio do compartilhamento do *kernel* de um SO (Sistema Operacional) hospedeiro (Vitalino; Castro, 2018). Para fazer o compartilhamento do *Kernel*, os contêineres recorrem a características já presentes no SO hospedeiro como, por exemplo, *namespaces* e *cgroups* do *kernel* Linux, tais características podem ser utilizadas para controlar os recursos disponibilizados ao contêiner por parte do hospedeiro.

Enquanto as máquinas virtuais realizam um processo de virtualização total onde por meio de um *hypervisor* todo o hardware é abstraído, desse modo é possível executar vários sistemas operacionais diferentes sobre um mesmo *host* hospedeiro (REDHAT, 2022b). Os contêineres, por outro lado, compartilham o *Kernel* do sistema operacional hospedeiro, desse modo não há necessidade de uma camada extra para abstração do *hardware*, uma vez que todos os contêineres compartilham os mesmos recursos disponíveis (Vitalino; Castro, 2018).

Figura 1 – Virtualização vs Contêiners



Fonte: Adaptado de (REDHAT, 2022b)

Devido ao fato dos contêineres executarem por meio do compartilhamento do *Kernel* do SO hospedeiro, eles podem ter uma menor flexibilidade, por exemplo, um contêiner Linux não pode ser executado sobre o *Kernel* do Windows (Turnbull, 2014). A despeito disso apresentam uma maior leveza e integração com o sistema operacional se comparados as máquinas virtuais.

O conceito de contêineres não é recente, já era possível trabalhar com tal conceito no Linux a bastante tempo, porém essa tarefa era complexa e requeria um grande conhecimento

técnico por parte do usuário, varias adições foram feitas ao *Kernel Linux* visando melhorias nesse cenário, mas foi com o surgimento do Docker¹ que os contêineres se popularizaram.

O Docker é um software que simplifica os processos de criação e gerenciamento de contêineres, adicionando uma camada de abstração que simplifica o uso desses contêineres ao abstrair as operações de baixo nível. De forma básica o Docker pode ser definido em dois componentes principais: Docker Cliente e o Docker Engine. A interação do usuário é realizada com o Docker Cliente geralmente por meio de uma interface de linha de comando ou arquivos de configuração, que permitem ao usuário executar comandos e configurar o ambiente de forma simples e intuitiva. Por sua vez, o Docker Engine é a parte do Docker responsável por realizar as tarefas mais complexas, o Docker cliente irá realizar o contato com o Docker Engine por meio de uma API, e este irá executar as ações necessárias para criar, gerenciar e executar os contêineres conforme as solicitações do usuário. Essa arquitetura cliente-servidor do Docker permite que os usuários utilizem a ferramenta de forma simplificada, concentrando-se apenas nas tarefas que desejam realizar, enquanto o Docker Engine cuida dos detalhes técnicos e operacionais.

O número de contêineres utilizados em uma infraestrutura pode aumentar rapidamente trazendo consigo uma certa complexidade, visando facilitar essa complexidade foram criadas ferramentas para orquestração. O Kubernetes é uma opção de código aberto criada pelo Google, sendo hoje a ferramenta mais popular para orquestração de contêineres.

2.2 Kubernetes

O Kubernetes é um orquestrador de contêineres desenvolvido inicialmente no Google a partir de uma vasta experiência em gerenciar aplicações baseadas em conteinerização, foi posteriormente doado a CNCF no ano de 2016. Se trata de um sistema de código aberto responsável por realizar o *deploy*, controle e escalonamento de contêineres (Kubernetes, 2019).

O Kubernetes atua em uma camada que somente faz a orquestração dos contêineres, desse modo não é papel do Kubernetes construir ou iniciar imagens de contêineres, para isso ele precisa de um ambiente de execução de contêineres que irá executar tarefas de baixo nível como iniciar ou finalizar contêineres, enquanto isso o Kubernetes vai atuar numa camada mais alta, tomando decisões de orquestração e certificando-se que o *cluster* mantenha seu estado (Poulton, 2023). O Kubernetes fornece uma interface padrão definida como *Container Runtime Interface* (CRI) responsável pela comunicação com o ambiente de execução de contêineres,

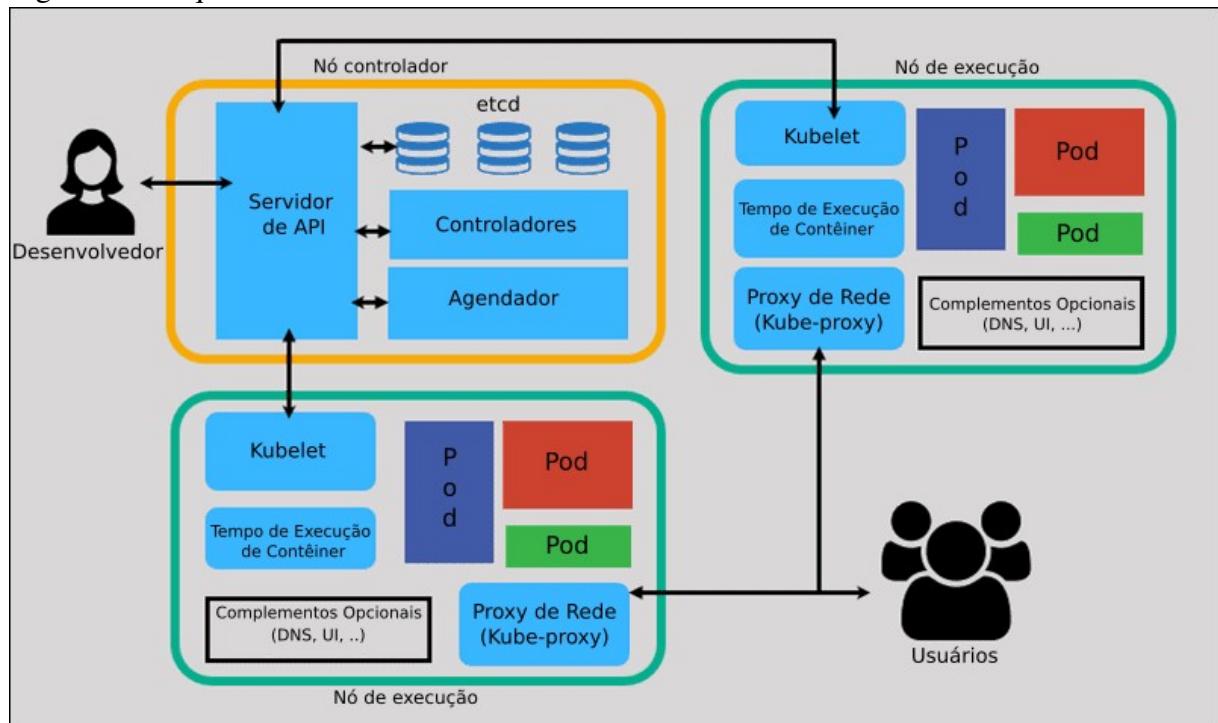
¹ <https://www.docker.com/>

desse modo você pode usar diferentes ambientes de execução de contêineres como o *Docker*, *CRI-O*, *containerd* entre outros.

Ao realizar um *deploy* do Kubernetes você irá passar a gerenciar um *cluster*, no qual teremos diversas máquinas funcionando como uma só, por meio de um modelo controle/execução, nessa estrutura o nó de controle que deverá sempre existir será responsável pelo gerenciamento do *cluster*, por meio da tarefa de realizar o controle dos nós de execução que irão ser responsáveis pela execução de fato das aplicações. A Figura 2 apresenta os principais componentes presentes na arquitetura do Kubernetes.

Os nós de controle são responsáveis pela execução de serviços que servem para o gerenciamento do *cluster*, podemos ter um único nó, funcionamento como nó de controle e de execução, mas para ambientes de produção nos quais se busca uma alta disponibilidade isso não é recomendado, sendo o ideal que isso não ocorra, onde teríamos um ou mais nós exclusivos para controle, e outros para execução, desse modo os nós de controle se concentram somente na tarefa de gerenciamento (Poulton, 2023). São componentes do *control plane*:

Figura 2 – Arquitetura do Kubernetes



Fonte: Adaptado de (Kaplarevic, 2019).

- a) Servidor de API: É uma Interface de Programação de Aplicação do Kubernetes. É utilizada para as comunicações externas e internas do *cluster* (Kaplarevic, 2019).
- b) *etcd*: É um banco de dados distribuídos responsável pelo armazenamento das informações

do *cluster* e seu estado. Por meio dele o nó de controle consegue manter o histórico das ações e o estado do *cluster*.

- c) Controlador: É responsável por executar vários controladores que se certificam que o cluster se encontra no estado desejado (Poulton, 2023).
- d) Agendador: É o componente responsável por definir em quais nós de execução novas tarefas serão executadas. Isso é feito por meio do ranqueamento dos nós disponíveis, que considera parâmetros como a quantidade de recursos disponíveis no nó, portas disponíveis, entre outros. O nó com melhor ranqueamento será escolhido para executar a tarefa. (Poulton, 2023).

Os nós de execução são os nós do *cluster* nos quais as tarefas serão executadas de fato, são componentes desses nós:

- a) *Kubelet*: É um agente do *cluster* executado em todos os nós *workers*, sendo responsável pelo gerenciamento dos pods de seu nó. Ele irá executar as instruções fornecidas pelo nó de controle, para manter o *cluster* no estado desejado pelo *controller*.
- b) Ambiente de Execução de Contêineres: É o ambiente responsável por executar e gerenciar os contêineres. Irá realizar funções básicas como início e término de contêineres, e gerenciar todo o ciclo de vida do contêiner, como gerencia de imagens, volumes e outros. O Tempo de execução de contêineres usado por padrão pelo Kubernetes é o containerd.
- c) *kube-proxy*: É um componente do *cluster* que fica responsável pela rede local do *cluster*, realizando funções de roteamento entre os nós do *cluster*, garantido que as informações da rede cheguem ao destino correto.

O Kubernetes possui conceitos que auxiliam o administrador a ter um melhor entendimento do seu funcionamento, alguns deles são:

- a) *Pod*: É a menor estrutura do Kubernetes. São uma maneira de envelopar os contêineres para sua utilização no Kubernetes. Um *Pod* pode conter um ou mais contêineres, sendo que todos compartilharam os mesmos recursos, como CPU memória, endereço de rede e outros.
- b) *Deployment*: É um recurso do Kubernetes utilizado para prover atualizações para *Pod*. Podemos especificar passar para um *deployment* por meio de arquivos do tipo json ou yaml, a definição das características e o estado desejado para um determinado *Pod*, onde o papel do *deployment* será manter esse estado e características.
- c) *Service*: é a maneira pela qual disponibilizamos o acesso a um *Pod*.

- d) *Namespace*: O conceito de *namespaces* permite que no qual podemos isolar grupos de recursos em um mesmo *cluster*, desse modo algumas aplicações podem executar sem saber da existência de outras.

2.3 Infraestrutura como Código

A infraestrutura como código (IaC) é o conceito de automatizar o processo de provisionamento da infraestrutura por meio da substituição de processos manuais por códigos que contêm as especificações da infraestrutura criada. Por meio desses códigos, é possível ter uma documentação mais precisa da infraestrutura. O versionamento também é uma parte importante, uma vez que os arquivos gerados devem ser tratados como qualquer outro projeto de *software*, evitando alterações manuais não comunicadas e garantindo o provisionamento do mesmo ambiente todas às vezes (REDHAT, 2022a).

A computação em nuvem veio para simplificar o provisionamento de infraestruturas. Por meio dela, tornou-se mais fácil e rápido criar e adicionar componentes a uma infraestrutura específica. A nuvem também oferece flexibilidade, permitindo mudanças frequentes. No entanto, essa flexibilidade pode trazer consigo maior complexidade e desafios no gerenciamento da infraestrutura.

Sem a IaC, o processo de gerenciamento da infraestrutura teria que ser feito de forma totalmente manual. Com o advento da computação em nuvem, torna-se cada vez mais complexo acompanhar e escalar a infraestrutura de maneira eficiente. Sem uma ferramenta de IaC, o processo de gerenciamento da infraestrutura pode se tornar desorganizado e frágil.

A IaC traz consigo uma série de benefícios que a tornam uma abordagem indispensável para o gerenciamento eficiente e escalável da infraestrutura na computação em nuvem:

- Tarefas que antes seriam realizadas de forma totalmente manual, podem ser automatizadas por ferramentas de IaC, o que resulta numa redução de custos bem como agiliza o processo de implantação e expansão de infraestruturas.
- O código é desenvolvido como qualquer outro *software*, permitindo que seja revisado e testado antes da implantação, reduzindo a ocorrência de erros.
- Por meio da IaC, também é possível reduzir os desvios de configuração, uma vez que o código-fonte, juntamente com o versionamento, pode ser utilizado como um registro garantindo que todas as alterações sejam documentadas.

Para implantar o processo de automação por meio da utilização de IaC, é necessário

ter acesso remoto ao ambiente onde a infraestrutura será implantada. A maioria dos provedores de nuvem disponibiliza APIs para que os usuários possam acessar e instanciar seus recursos, os quais também podem ser utilizados por ferramentas de gerenciamento de configuração para automatizar o processo de implantação. Essas ferramentas podem consistir em um conjunto de *scripts* que automatizam tarefas específicas, o que pode tornar o processo complexo. No entanto, atualmente existem várias ferramentas de código aberto que facilitam esse processo, permitindo que as configurações sejam escritas em linguagens de marcação como JSON ou YAML. Exemplos de tais ferramentas incluem o Ansible², Terraform³, entre outras.

² <https://github.com/ansible/ansible>

³ <https://www.terraform.io/>

3 TRABALHOS RELACIONADOS

Esta seção apresenta uma visão geral dos trabalhos relacionados com o projeto proposto.

3.1 On-demand virtual research environments using microservices

Em Capuccini *et al.* (2019) os autores buscam propor uma metodologia para implantar *Virtual Research Environments* (VRE) sob demanda na nuvem utilizando microsserviços. Busca-se encontrar uma maneira de implantar a infraestrutura de modo simples, mas que ainda forneça um bom nível de configuração, além disso deve ser possível executar os VRE em múltiplos provedores de *cloud*. A metodologia é desenvolvida a partir da ideia de ambientes sob demanda e de curta duração, desse modo a disponibilidade não é tão importante, enquanto a velocidade de implantação é crucial. A partir disso os autores apresentam a ferramenta kubeNow, uma plataforma *open-source* para instanciação de VREs sob-demanda. A arquitetura adotada pelos autores para implantação do VREs sob-demanda é composta por três camadas: provedor de *cloud* que irá gerenciar e fornecer os recursos virtuais a nível de infraestrutura; orquestrador e microsserviços. O kubeNow é construído com uma camada sobre *softwares* já adotados como Docker, Kubernetes, GlusterFS, Traefik, Cloudflare, Ansible e Packer. A implantação automatizada fornecida pelo kubeNow, está disponível por meio de uma Command Line Interface (CLI), que visa tornar transparente os serviços de nuvem, de modo que os VRE possam ser implementados sem a necessidade de realizar tarefas muito complexas.

Assim como em Capuccini *et al.* (2019) visamos desenvolver uma ferramenta que possa ser de fácil utilização, além de que os ambientes deverão ser sob-demanda, de modo que o usuário possa destruir e recriá-los conforme a utilização, buscando com isso uma redução de custos. Diferentemente de Capuccini *et al.* (2019) nos não fornecemos uma infraestrutura a ser utilizada por pesquisadores, mas sim para estudantes.

3.2 Kubitect - a Solution for On-premise Cluster Deployment

Em Mušić e Fortuna (2022) é apresentado uma ferramenta *open-source* que objetiva automatizar o processo de implantação de clusters Kubernetes *on-premise*. Como motivação os autores utilizaram a ausência de ferramentas que possibilitem a fácil criação de pequenos ambientes Kubernetes localmente, sendo que as opções disponíveis no mercado focam em

ambientes de larga escala. Uma das dificuldades enfrentadas é a preparação do ambiente, como criação de máquinas virtuais, resolução de dependências e outras que são abstraídas do usuário final pelos provedores de *cloud*. A ferramenta possui um arquivo de configuração em linguagem declarativa que será utilizado pelo usuário para configuração do seu ambiente, após isso o usuário fará uso do Kubitect, ferramenta de linha de comando que criará o ambiente a partir do arquivo de configuração. A ferramenta assume que os *hosts* já estão pre-configurados, com um *hypervisor* e uma API de virtualização. Somente é suportado como API de virtualização *Libvirt*, como *hypervisor* pode ser utilizado qualquer um que seja suportado pela API *Libvirt*. A solução é implementada em *Golang* com a biblioteca *Cobra*, porém também são utilizados Ansible para criação de diretórios e geração de arquivos requeridos, Terraform para providenciar a infraestrutura virtual, Cloudinit para configuração das máquinas virtuais, kubernetes como orquestrador de contêineres e o *Kubespray* como ferramenta para configuração dos *clusters* Kubernetes.

Assim como em Mušić e Fortuna (2022) planejamos desenvolver uma ferramenta que automatize o processo de implantação do *cluster* Kubernetes, sendo estes destinados a pequenos ambientes de estudo. Diferentemente de Mušić e Fortuna (2022), nossa ferramenta não visa o ambiente *on-premise*.

3.3 DVL : Uma Ferramenta para Criação de Laboratórios Práticos de Disciplinas da Área de TI Utilizando Virtualização Baseada em Contêineres

Em SILVA *et al.* (2018) é abordado a virtualização de contêineres como alternativa para suprir a falta de recursos computacionais necessários para realização de laboratórios práticos apresentados por muitas instituições de ensino. A partir dessa premissa é apresentado a ferramenta DVL (Docker Virtual Lab), uma solução para automatização de laboratórios virtuais a partir da utilização de contêineres Docker. A ferramenta possui uma interface Web pela qual será feita a interação para criação dos laboratórios virtuais pelo usuário professor. O processo de criação de tais laboratórios será feito automatizadamente por meio de *Scripts Shell*. O laboratório consiste em cenários, que como definido pelo autor são um conjunto de contêineres destinados a um determinado usuário, a ferramenta também conta com um banco de dados MySQL para armazenar as informações pertinentes como usuários e outras. O acesso dos usuários alunos aos cenários instanciados é realizado por meio acesso remoto (SSH). As funções fornecidas pela ferramenta são a instanciação de cenários onde o professor poderá instanciar cenários já

disponíveis na ferramenta, após a instanciação o professor terá a possibilidade de gerenciar os cenários e o professor poderá também realizar a criação de novos cenários elaborados por ele próprio.

Assim como em SILVA *et al.* (2018) nosso trabalho planeja fornecer uma ferramenta para uso estudantil. Diferentemente de SILVA *et al.* (2018) nós utilizaremos Kubernetes, sendo que o nosso foco será somente no fornecimento de ambientes de estudo para o aprendizado do Kubernetes, e não em ambientes diversos do estudo de redes de computadores.

3.4 Análise Comparativa.

O Quadro 1 apresenta uma análise comparativa entre a nossa proposta e os 3 trabalhos relacionados apresentados anteriormente em quatro pontos, sendo: (i) se o trabalho tem como foco o uso educacional; (ii) se apresenta uma CLI; (iii) se usa o Kubernetes como ferramenta de desenvolvimento; e (iv) se pode ser utilizado em diferentes *clouds*.

Quadro 1 – Análise comparativa entre os trabalhos relacionados e a nossa proposta.

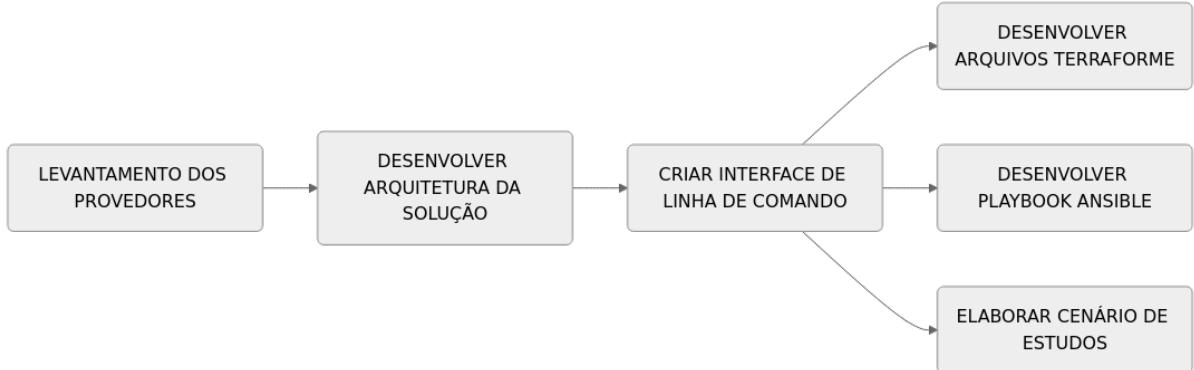
Trabalhos	Estudantil	Automação	Kubernetes	Cloud
(Capuccini <i>et al.</i> , 2019)	Não	Sim	Sim	Sim
(Mušić; Fortuna, 2022)	Não	Sim	Sim	Não
(SILVA <i>et al.</i> , 2018)	Sim	Sim	Não	Não
Trabalho proposto	Sim	Sim	Sim	Sim

Fonte: Elaborado pelo autor

4 PROCEDIMENTOS METODOLÓGICOS

A Figura 3 apresenta o fluxograma da metodologia seguida para a realização deste trabalho.

Figura 3 – Fluxograma da Metodologia



Fonte: Elaborado pelo autor

4.1 Levantamento dos provedores de nuvem.

Na Tabela 1 temos uma lista de alguns ambientes de nuvem que oferecem um período de teste gratuito e benefícios para estudantes. Esse levantamento foi feito com auxílio do repositório do GitHub (learnk8s, 2023).

Tabela 1 – Levantamento de ambientes de *cloud Free Trial*

Plataforma	Créditos	Período	Requer Cartão
AWS	US\$ 50,00	6 meses	Não
Azure	US\$ 100,00	12 meses	Não
GCP	US\$ 300,00	3 meses	Sim
Digital Ocean	US\$ 200,00	2 meses	Sim
Linode	US\$ 100,00	2 meses	Sim
OCI	US\$ 300,00	1 mês	Sim
IBM Cloud	US\$ 200,00	1 mês	Não

Fonte: elaborada pelo autor.

A partir da Tabela 1, foram escolhidas as nuvens da AWS e Azure para a realização deste trabalho, focando no uso estudantil. A ausência da necessidade de cartão de crédito foi um fator determinante nessa escolha. Além disso, ambas as nuvens possuem projetos voltados para estudantes: o AWS Academy¹ da AWS e o Microsoft Azure for Students² da Azure. Assim

¹ <https://aws.amazon.com/pt/training/awsacademy/>

² <https://azure.microsoft.com/pt-br/free/students>

como em (Capuccini *et al.*, 2019), é necessário que a nuvem forneça certos serviços para ser usada neste projeto, incluindo serviços de computação, Internet, armazenamento em bloco e uma API para acesso a todos esses serviços.

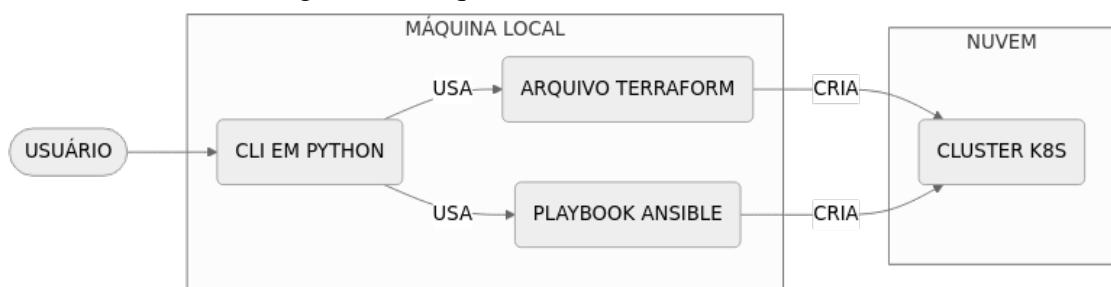
4.2 Arquitetura da solução

Uma visão geral da arquitetura da ferramenta pode ser observada na Figura 4. A solução é composta por três camadas principais:

- Interface de linha de comando (CLI): Desenvolvida em Python, permite ao usuário interagir com a ferramenta por meio de comandos intuitivos (ex: criação/destruição de *clusters*).
- Automação de infraestrutura: Utiliza arquivos Terraform para provisionamento de recursos em nuvem (ex: instâncias EC2 na AWS, máquinas virtuais no Azure) e *playbooks* Ansible para configuração automatizada dos nós.
- Orquestração do cluster: Baseia-se no kubeadm para inicialização de *clusters* Kubernetes autogerenciados, priorizando o contato direto com componentes do *control plane* (API Server, etcd, Scheduler, Controller Manager).

A opção por *clusters* autogerenciados, em vez de serviços gerenciados como EKS ou AKS, dará ao estudante um conhecimento mais profundo sobre o Kubernetes por meio da exposição a cenários práticos de troubleshooting e configuração manual, competências exigidas em certificações como o CKA. Essa abordagem também simula ambientes produtivos heterogêneos, onde o autogerenciamento é frequentemente requisitado.

Figura 4 – Arquitetura da Ferramenta



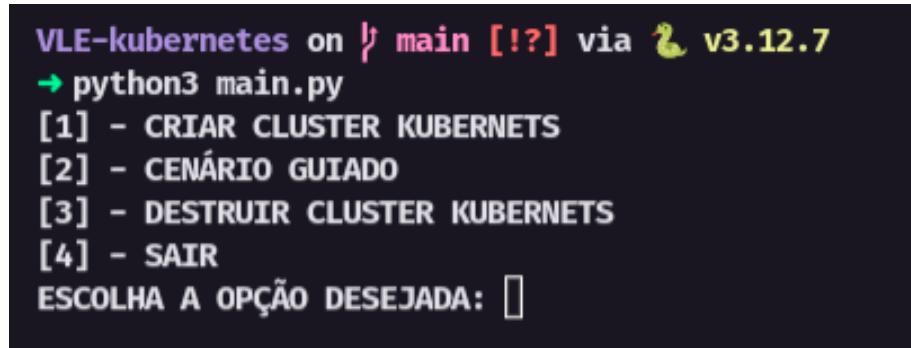
Fonte: Elaborado pelo autor

4.2.1 Interface de linha de comando

A Figura 5 ilustra a interface de linha de comando à qual o usuário tem acesso , essa interface foi desenvolvida em Python e por meio dela o usuário irá ter acesso às principais

funções ofertadas pela ferramenta, de forma fácil e intuitiva.

Figura 5 – CLI da Ferramenta

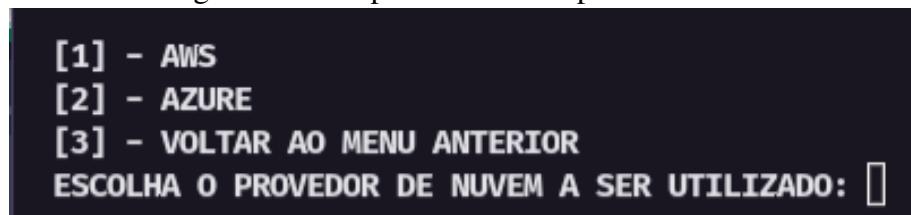


```
VLE-kubernetes on ⚡ main [!?] via 🐄 v3.12.7
→ python3 main.py
[1] - CRIAR CLUSTER KUBERNETS
[2] - CENÁRIO GUIADO
[3] - DESTRUIR CLUSTER KUBERNETS
[4] - SAIR
ESCOLHA A OPÇÃO DESEJADA: [
```

Fonte: Elaborado pelo autor

Por meio dessa interface o usuário poderá realizar a criação de um *cluster* Kubernetes simples na AWS ou Azure, que poderá usar como *playground* para o estudo do Kubernetes de forma prática e conveniente, ao optar pela criação do *cluster* o usuário poderá escolher entre os dois provedores de nuvem atualmente suportados pela ferramenta como mostrado na Figura 6. A interface também oferece a opção de acessar cenários guiados, que guiarão o usuário para obtenção de conhecimentos básicos sobre o Kubernetes por meio de atividades como configuração de um *cluster*, realização de *deploys* entre outras atividades. Essas atividades ajudarão o usuário a se familiarizar com o Kubernetes e seus comandos básicos. Além disso, a ferramenta permite destruir a infraestrutura criada, evitando cobranças financeiras indesejadas.

Figura 6 – CLI para escolha do provedor



```
[1] - AWS
[2] - AZURE
[3] - VOLTAR AO MENU ANTERIOR
ESCOLHA O PROVEDOR DE NUVEM A SER UTILIZADO: [
```

Fonte: Elaborado pelo autor

4.2.2 Arquivo Terraform

Ao realizar a seleção de uma das opções disponíveis na CLI o *Script Python* realizará por meio de um arquivo Terraform Figura 7 a opção desejada, no provedor de nuvem escolhido.

Este arquivo Terraform se conectará à nuvem de forma segura, utilizando as chaves de acesso fornecidas pelo ambiente. Há dois arquivos Terraform na ferramenta, que serão acionados pela CLI conforme a escolha do ambiente de nuvem selecionado pelo usuário. O arquivo da

Figura 7 – Exemplo de arquivo Terraform



```

1  terraform {
2      required_providers {
3          aws = {
4              source = "hashicorp/aws"
5              version = "~> 5.0"
6          }
7      }
8  }
9
10 provider "aws" {
11     region = "us-east-1" # Norte da Virginia
12 }
13
14 # locals
15

```

Fonte: Elaborado pelo autor

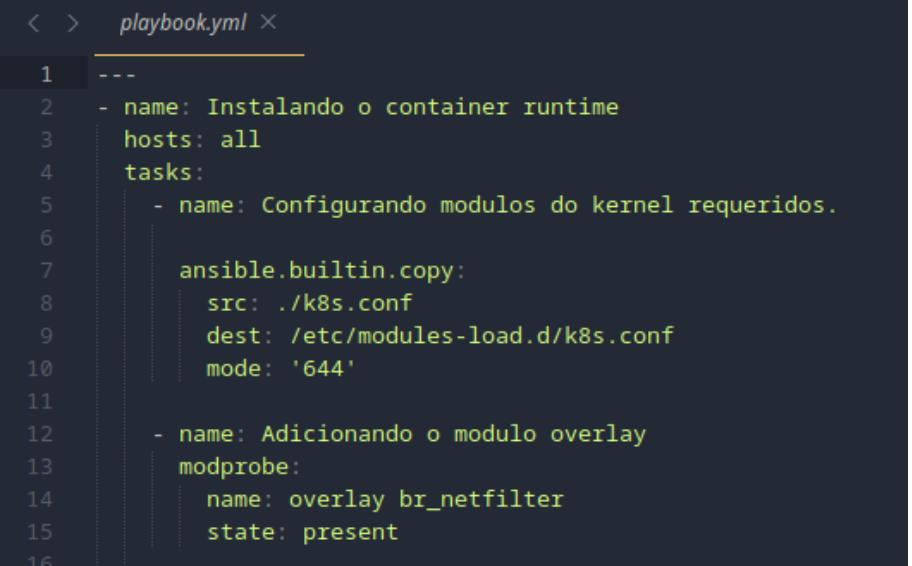
AWS irá utilizar o VPC e sub-rede padrão, e irá criar um Grupo de Segurança vinculado a esse VPC para controlar o acesso e garantir a comunicação eficaz dentro do VPC. Além disso, irá criar três instâncias EC2 do tipo *t2.medium* do Ubuntu 24.04 LTS com um armazenamento de 20GB cada. Já no arquivo da Azure será utilizada a Vnet e sub-rede padrão, assim como na AWS será configurado um grupo de segurança para controle de acesso e comunicação interna dentro da Vnet, além disso serão criadas três instâncias do tipo *Standard-B2ms* do Ubuntu 24.04 LTS com um armazenamento de 30GB cada.

4.2.3 *Playbook Ansible*

A execução do *Playbook Ansible* será realizada ao final do Arquivo Terraform. O *playbook*, como mostrado na Figura 8 desempenha um papel crucial, ele será responsável por realizar todo a configuração do *cluster* Kubernetes de forma automatizada, que é o principal objetivo deste trabalho.

Para isso ele irá acessar as máquinas criadas na etapa anterior pelo TerraForm, através do protocolo *Secure Shell* (SSH). Ao se conectar, ele irá executar por meio de *roles* todo o processo de configuração do *cluster* Kubeadm para aprendizado, envolvendo etapas como configuração das máquinas, instalação do contêiner *runtime*, configuração de *cgroup driver*, inicialização do *control-plane node* e adição dos *workers nodes*. Esse é um processo complexo e suscetível a erros; a utilização do *playbook* Ansible irá eliminar esses contratemplos, criando um

Figura 8 – Playbook Ansible.



```

1  ---
2  - name: Instalando o container runtime
3  hosts: all
4  tasks:
5      - name: Configurando modulos do kernel requeridos.
6
7      ansible.builtin.copy:
8          src: ./k8s.conf
9          dest: /etc/modules-load.d/k8s.conf
10         mode: '644'
11
12      - name: Adicionando o modulo overlay
13        modprobe:
14            name: overlay br_netfilter
15            state: present
16

```

Fonte: Elaborado pelo autor

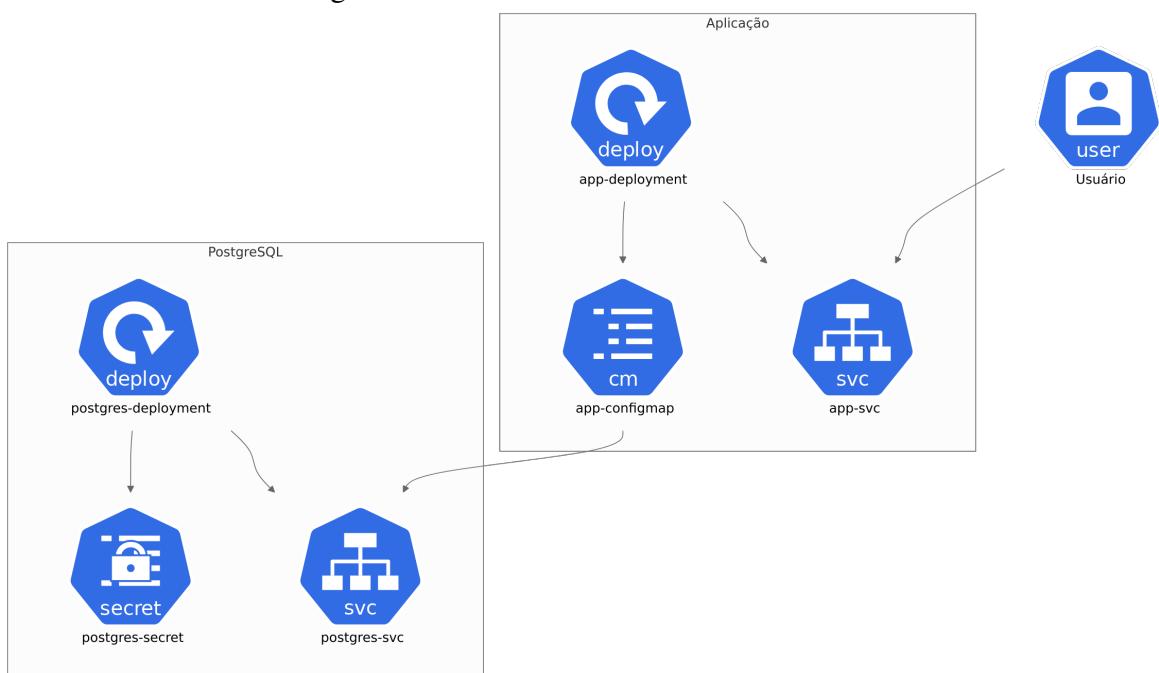
ambiente de aprendizado de forma rápida e eficiente.

4.3 Roteiro de Estudo

Para testar se o cluster fornecido pela ferramenta está funcional, foi estruturado um roteiro de estudos baseado em um tutorial guiado, disponibilizado através de uma página Markdown no GitHub, acessível via link na própria ferramenta. Este tutorial oferece uma visão abrangente sobre a configuração dos principais componentes do Kubernetes, como Deployments, ReplicaSets, Services, entre outros. O cenário proposto envolve a implementação de uma aplicação web simples, escrita em Python, para listagem de livros, que se conecta a um banco de dados PostgreSQL. A Figura 9 ilustra o cenário a ser desenvolvido pelo aluno.

Neste cenário, o aluno irá realizar o *deployment* do banco de dados PostgreSQL, utilizando um *secret* para armazenar informações sensíveis, como nome de usuário e senha. Adicionalmente, será criado um *service* para expor o banco de dados dentro do *cluster* Kubernetes. Um configmap será configurado para conter o *endpoint* do *service* do PostgreSQL, o qual será utilizado pelo *deployment* da aplicação para se conectar ao banco de dados. Além disso, será criado um *service* adicional para expor a aplicação para fora do cluster, permitindo o acesso externo à mesma. Todas essas configurações e a ferramenta em si podem ser visualizados por meio deste link.

Figura 9 – Cenário de Estudos.



Fonte: Elaborado pelo autor

5 EXPERIMENTOS

Neste Capítulo, descrevemos o processo utilizado para a realização dos experimentos e análise dos resultados obtidos.

5.1 Descrição dos Experimentos

Nesta etapa, realizamos um teste de usabilidade da ferramenta. Para isso, divulgamos um formulário do Google Forms com o intuito de recrutar participantes interessados em testar a ferramenta; além disso, o questionário também possuía algumas perguntas para traçar o perfil dos participantes. O questionário ficou disponível por uma semana, no período compreendido entre 31 de janeiro e 7 de fevereiro. Esse questionário teve um total de 21 respostas.

Após esse período, os interessados foram incluídos em um grupo do *WhatsApp* para esclarecimento de dúvidas e orientações sobre a condução do teste. Os participantes foram divididos aleatoriamente em dois grupos: um dos grupos contendo 11 participantes testou a utilização da ferramenta na AWS, enquanto o outro, contendo 10 participantes, realizou o mesmo processo na Azure. Todos receberam o link do Github da ferramenta, onde estão contidos todos os detalhes e passo a passo para utilização da ferramenta e o cenário guiado do roteiro de estudos. Para os participantes da AWS, criamos uma turma na plataforma AWS *Academy* e os inscritos foram automaticamente adicionados a ela. Já os integrantes do grupo da Azure precisaram criar uma conta no programa Azure for *Students*. O passo a passo para criação da conta na Azure for *Students* também está contido no link disponibilizado. O prazo para que os participantes realizassem o teste da ferramenta foi durante os dias oito de fevereiro a 14 de fevereiro.

Ao final do período de testes, os participantes responderam a um questionário de avaliação disponibilizado via Google Forms, proporcionando *feedback* sobre a experiência de uso e possíveis dificuldades na utilização da ferramenta. O Quadro 2 apresenta as perguntas contidas neste questionário.

5.2 Analise do questionário de inscrição

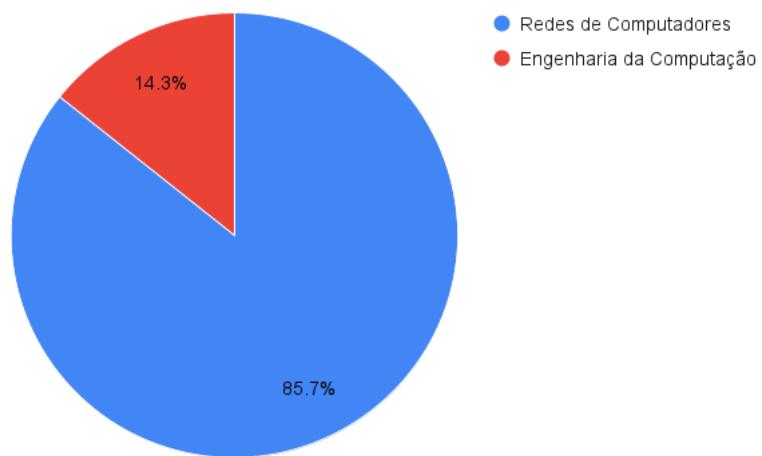
Nesta seção, analisamos os dados coletados por meio do questionário de inscrição para o teste da ferramenta.

As Questões 1, 2 e 3 tiveram como objetivo traçar o perfil dos usuários participantes do teste. Os resultados, apresentados nas Figuras 10, 11 e 12, revelaram as seguintes

características:

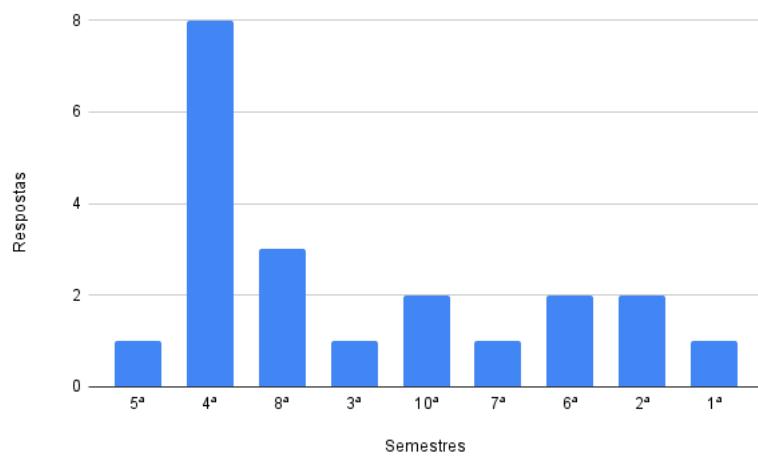
Conforme o apresentado no gráfico da Figura 10, observa-se que a maioria significativa dos inscritos (87,5%) pertence ao curso de Redes de Computadores. Já na Figura 11, constata-se que o maior contingente de participantes (8 indivíduos) está matriculado no 4º semestre de seus respectivos cursos. Por fim, a Figura 12 demonstra que 85,7% dos avaliados possuem acesso a uma máquina com sistema operacional Linux.

Figura 10 – **Questão 1** - Qual curso você estuda??



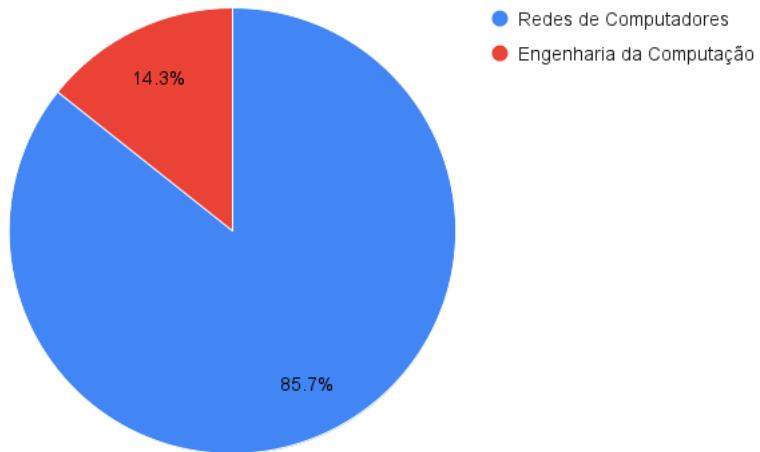
Fonte: Elaborado pelo autor

Figura 11 – **Questão 2** - Qual seu semestre atual?



Fonte: Elaborado pelo autor

Figura 12 – **Questão 3** - Você possui acesso a uma máquina Linux?



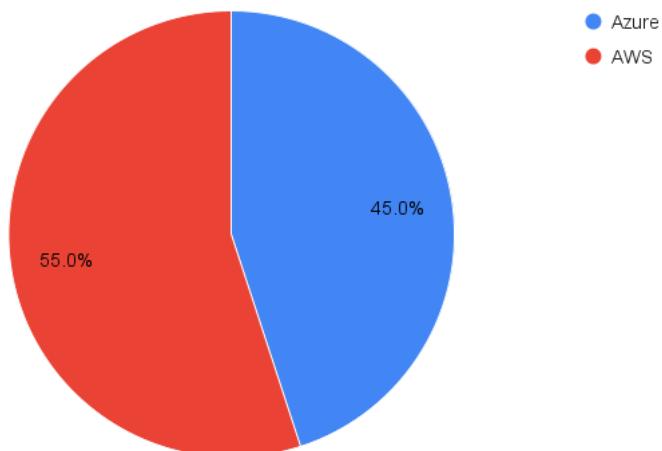
Fonte: Elaborado pelo autor

5.3 Analise do questionário de avaliação da ferramenta

Nesta seção, analisaremos os dados coletados por meio do questionário de avaliação da ferramenta. Vale ressaltar que apenas 20 participantes responderam a essa etapa do estudo.

A **Questão 1** foi elaborada para identificar quantos usuários dos grupos pré-definidos no estudo realizaram testes nos provedores de nuvem distintos suportados pela ferramenta. O gráfico contido na Figura 13 demonstra que 55% dos participantes realizaram o teste na AWS, enquanto 45% utilizaram a Azure para a execução dos testes. Revelando que todos os usuários do grupo da AWS realizaram o teste, enquanto um usuário do grupo da Azure não respondeu ao questionário.

Figura 13 – **Questão 1** - Qual provedor de nuvem você utilizou?

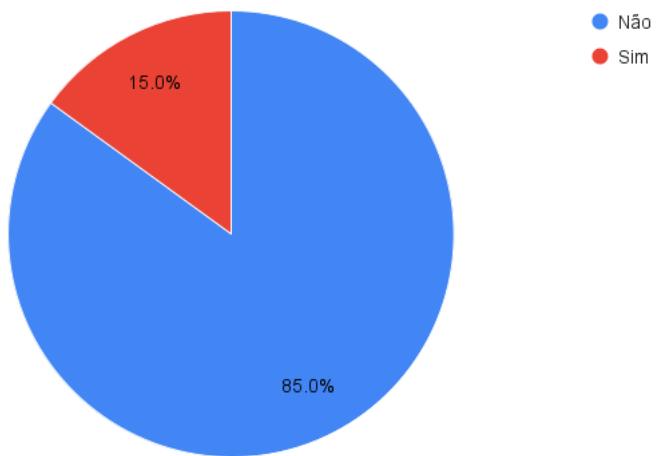


Fonte: Elaborado pelo autor

A **Questão 2** buscou verificar se os participantes tinham experiência prévia com o

uso do Kubernetes. Os dados exibidos no gráfico da Figura 14 revelam que apenas 15% dos avaliados já haviam utilizado a tecnologia, indicando um perfil majoritariamente iniciante nessa ferramenta.

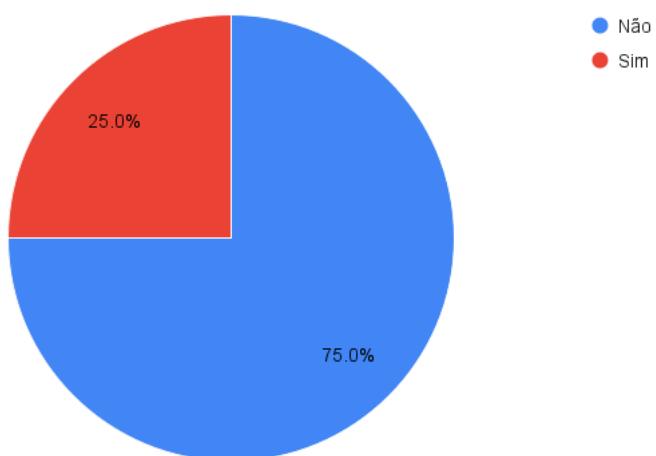
Figura 14 – **Questão 2** - Você já possuía alguma experiência prévia com o Kubernetes?



Fonte: Elaborado pelo autor

A **Questão 3** teve como objetivo identificar a familiaridade dos participantes com a utilização de ferramentas de provisionamento de recursos. No gráfico apresentado na Figura 15 podemos observar que a maioria (75%) dos participantes não possuía nenhum contato prévio com a utilização de ferramentas de provisionamento de recursos.

Figura 15 – **Questão 3** - Você já utilizou alguma ferramenta de provisionamento de recursos?

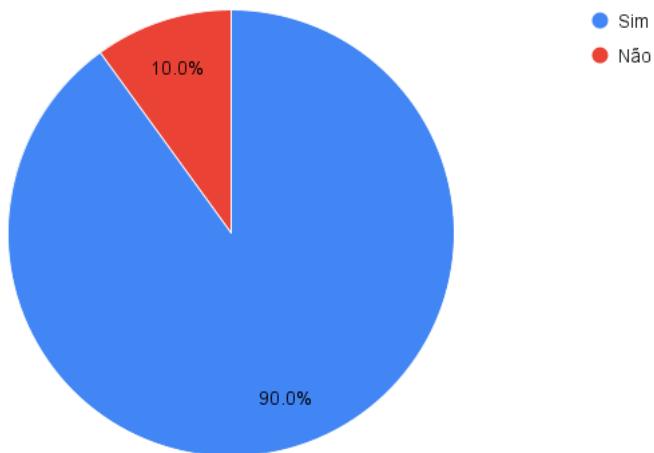


Fonte: Elaborado pelo autor

As **Questões 4 e 7** foram elaboradas com o objetivo de identificar se os usuários conseguiram realizar as duas principais atividades pretendidas no experimento, sendo elas a execução da ferramenta e realização do roteiro de estudo. Os resultados destas questões podem

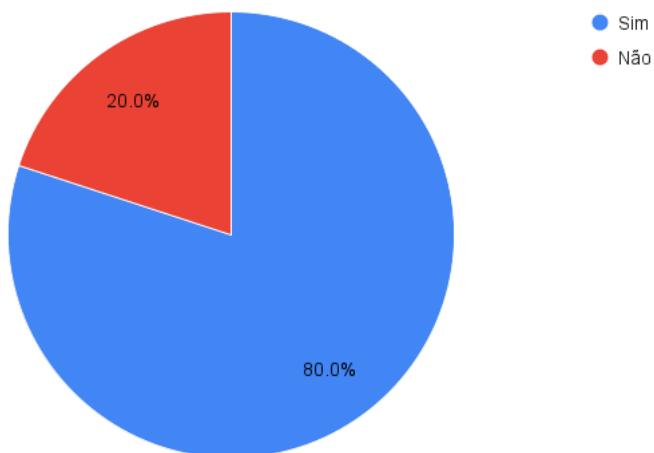
ser observados nas Figuras 16 e 17. O gráfico apresentado na Figura 16 revela que a grande maioria dos participantes (90%) conseguiu realizar a execução da ferramenta. Já os dados representados na Figura 17 demonstra que a maior parcela dos usuários (80%) conseguiu realizar o cenário de estudos.

Figura 16 – **Questão 4** - Você conseguiu executar a ferramenta?



Fonte: Elaborado pelo autor

Figura 17 – **Questão 7** - Você conseguiu realizar o cenário de estudos?

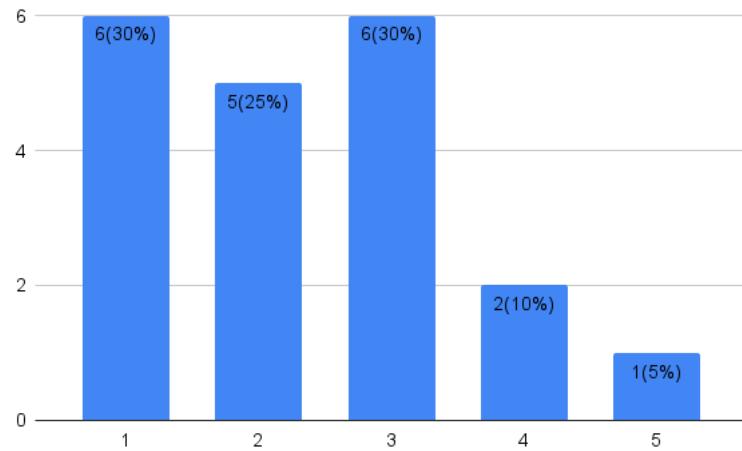


Fonte: Elaborado pelo autor

A **Questão 5** foi elaborada para avaliar a opinião dos usuários sobre a dificuldade de execução da ferramenta. O gráfico exibido na Figura 18 mostra que, em uma escala de 1 (Muito Fácil) a 5 (Muito Difícil), a maioria dos participantes (55%) avaliou o nível de dificuldade como Fácil ou Muito Fácil.

A **Questão 6** foi elaborada para avaliar o nível de satisfação dos usuários em relação à utilização da ferramenta. O gráfico exibido na Figura 19 mostra que, em uma escala de 1

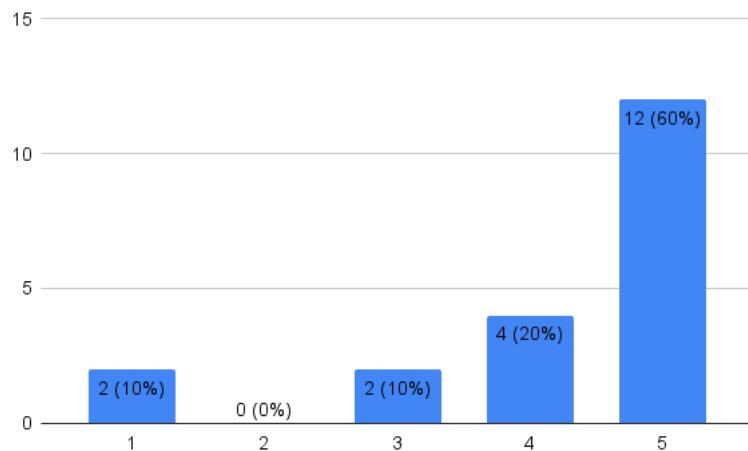
Figura 18 – **Questão 5** - Em que escala você avalia o nível de dificuldade em utilizar a ferramenta?



Fonte: Elaborado pelo autor

(Muito Insatisfeito) a 5 (Muito Satisfeito), a maioria dos participantes (60%) se apresentaram muito satisfeitos com o uso da ferramenta.

Figura 19 – **Questão 6** - Em que escala está seu nível de satisfação com a ferramenta?

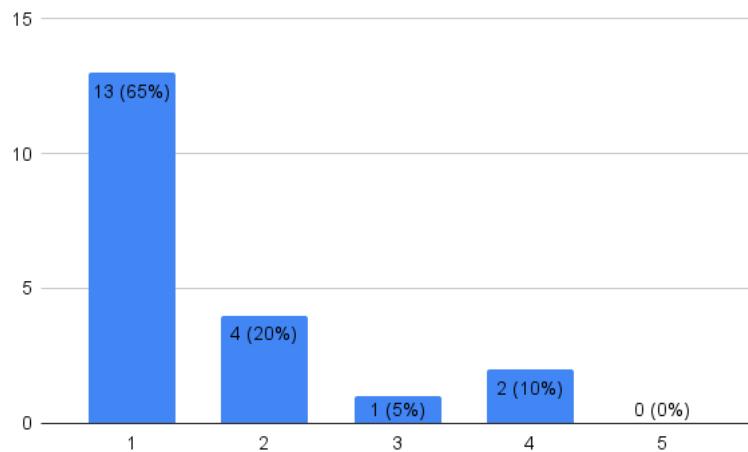


Fonte: Elaborado pelo autor

As **Questões 8 e 10** foram formuladas com o propósito de avaliar a opinião dos usuários em relação às condições de execução da ferramenta durante a atividade. Os resultados dessas avaliações podem ser conferidos nas Figuras 20 e 21. O gráfico apresentado na Figura 20, em uma escala de 1 (Nenhum Bug) a 5 (Inutilizável devido aos bugs), destaca que a maioria dos usuários (65%) não presenciou nenhum *bug* durante a execução da ferramenta. Na Figura 21 50% dos participantes considerou o tempo de criação da ferramenta como moderado.

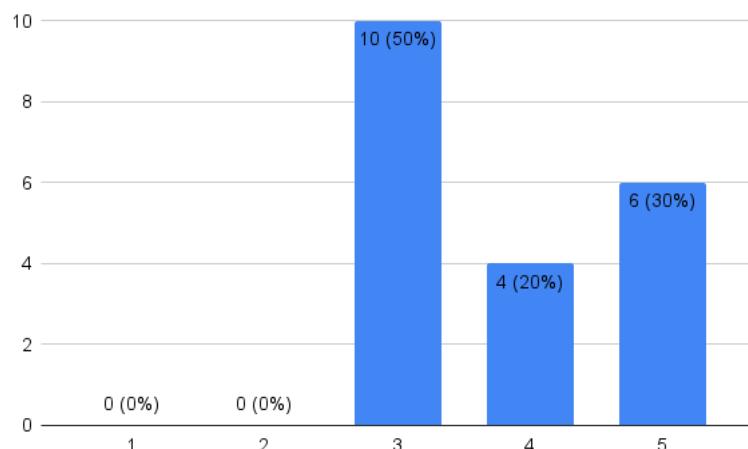
As **Questões 9 e 13** foram pensadas com o objetivo de avaliar a percepção dos usuários sobre os benefícios da ferramenta para o estudo do **Kubernetes**. Os resultados dessas

Figura 20 – **Questão 8** - Em que escala houve bugs durante a utilização da ferramenta?



Fonte: Elaborado pelo autor

Figura 21 – **Questão 10** - Como você avalia o tempo de criação da infraestrutura pela ferramenta?

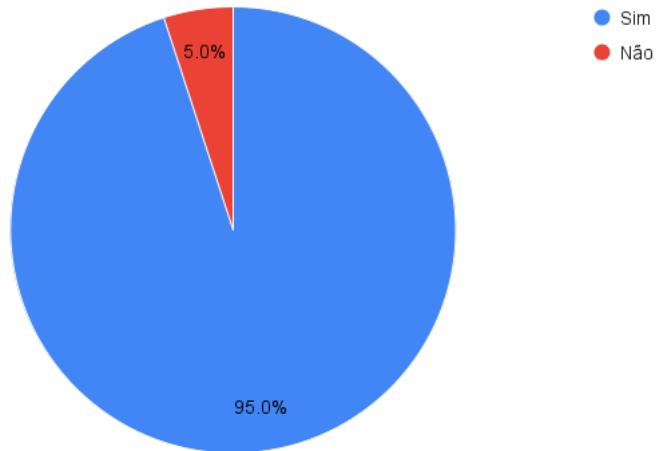


Fonte: Elaborado pelo autor

perguntas podem ser observados nas Figuras 22 e 23. Os dados apresentados na Figura 22 demonstram que a maioria dos respondentes (95%) considerou que a ferramenta auxiliaria seus estudos de kubernetes. Já no gráfico apresentado na Figura 23, em uma escala de 1 (certamente não indicaria) a 5 (certamente indicaria), 60% dos membros certamente indicariam a ferramenta para pessoas interessadas em estudar o kubernetes.

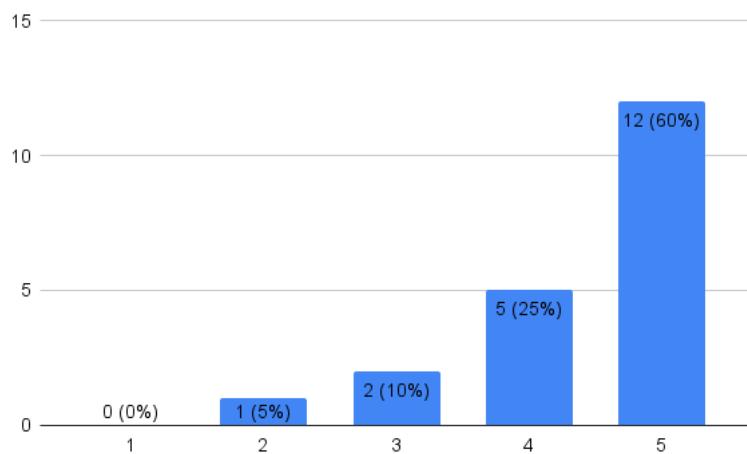
As **Questões 11 e 12** foram elaboradas objetivando obter a opinião dos alunos sobre os recursos disponibilizados para execução da atividade proposta. Os resultados podem ser observados nas Figuras 24 e 25. O gráfico representado na Figura 24 em uma escala de 1 (muito confusa) a 5 (muito clara) evidencia que a maioria dos integrantes da pesquisa (85%) considerou a descrição de utilização da ferramenta clara ou muito clara. Os resultados na Figura 25 avaliaram a opinião dos membros em relação à clareza do roteiro de estudo, nesse caso

Figura 22 – **Questão 9** - Você acredita que a ferramenta facilitaria seus estudos de Kubernetes?



Fonte: Elaborado pelo autor

Figura 23 – **Questão 13** - Em que escala você indicaria esta ferramenta para alguém está querendo estudar o Kubernetes?



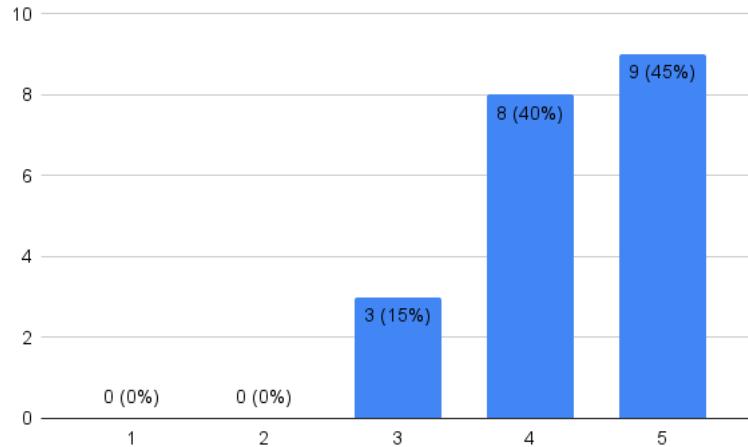
Fonte: Elaborado pelo autor

70% deles consideraram o roteiro de estudos claro ou muito claro.

5.4 Discussão Final

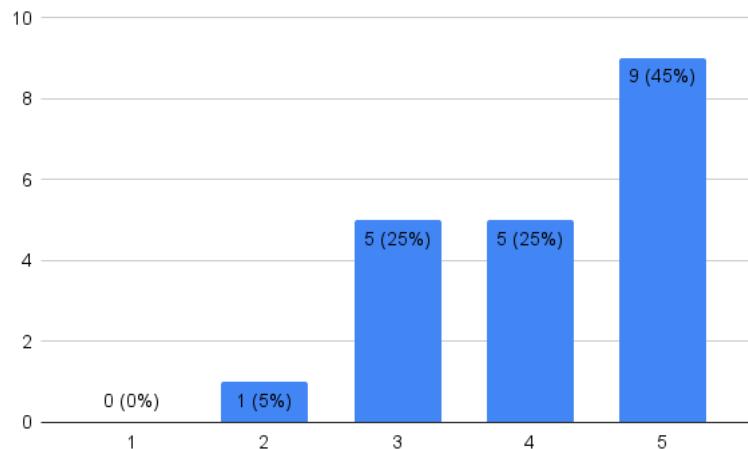
Analisamos os gráficos fornecidos pelos participantes e pudemos concluir que a ferramenta se encontra funcional nos dois ambientes de nuvem pretendidos pelo trabalho. Constatamos ainda que a maioria deles não possuía nenhuma experiência com as tecnologias utilizadas neste trabalho. Mesmo com essa falta de conhecimento, a grande maioria deles conseguiu executar a ferramenta e o roteiro de estudos propostos de forma satisfatória, o que pode ser observado pelo fato de a grande maioria ter considerado a ferramenta de fácil utilização.

Figura 24 – **Questão 11** -Você considerou a descrição de utilização da ferramenta confusa?



Fonte: Elaborado pelo autor

Figura 25 – **Questão 12** - Você considerou a descrição do cenário de estudo confusa?



Fonte: Elaborado pelo autor

Quadro 2 – Perguntas utilizadas no questionário de avaliação.

Número	Questões
1	Qual provedor de nuvem você utilizou??
2	Você já possuía alguma experiência prévia com o Kubernetes?
3	Você já utilizou alguma ferramenta de provisionamento de recursos?
4	Você conseguiu executar a ferramenta?
5	Em que escala você avalia o nível de dificuldade em utilizar o VLE?
6	Em que escala está seu nível de satisfação com a ferramenta?
7	Você conseguiu realizar o cenário de estudos?
8	Em que escala houve bugs durante a utilização da ferramenta?
9	Você acredita que a ferramenta facilitaria seus estudos de Kubernetes?
10	Como você avalia o tempo de criação da infraestrutura pela ferramenta?
11	Você considerou a descrição de utilização da ferramenta confusa?
12	Você considerou a descrição do cenário de estudo confusa?
13	Em que escala você indicaria esta ferramenta para alguém está querendo estudar o Kubernetes?

Fonte: Elaborado pelo autor

A avaliação dos usuários também foi positiva em relação a satisfação com a ferramenta, onde a maioria se demonstrou além de satisfeita, dispostos a indicar a ferramenta para

outros usuários que estiverem realizando o estudo do Kubernetes. A avaliação dos integrantes também foi positiva em relação aos recursos disponibilizados para utilização da ferramenta.

A ferramenta se demonstrou estável e com a maioria dos usuários não reportando nenhum bug durante sua utilização. Embora poucos usuários tenham relatados bugs metade dos usuários considerou o tempo de execução da ferramenta moderado, demonstrando aí um possível ponto de melhoria.

De modo geral a análise dos dados demonstrou que a ferramenta apresentou um índice de satisfação significativo.

6 CONCLUSÃO

Este trabalho desenvolveu uma ferramenta para automação da criação de clusters Kubernetes com o **kubeadm** em ambientes de nuvem, visando facilitar o estudo dessa tecnologia, líder de mercado em orquestração de contêineres. O objetivo central foi fornecer um ambiente totalmente gratuito por meio de programas educacionais, como o *AWS Academy* e o *Azure for Students*, direcionados ao público estudantil.

Para atingir o objetivo geral de propor a ferramenta, foram estabelecidos três objetivos específicos. O primeiro consistiu em identificar plataformas de nuvem com períodos de utilização gratuita. Verificou-se que a maioria dos provedores exigia registro de cartão de crédito mesmo para testes limitados, o que representou uma limitação significativa. Após análise, selecionaram-se dois provedores alinhados aos requisitos do projeto: a *Azure* (via programa *Azure for Students*) e a *AWS* (por meio do *AWS Academy*).

O segundo objetivo envolveu o projeto e o desenvolvimento da ferramenta. Nesta etapa, identificaram-se as soluções de Infraestrutura como Código (**IaC**) mais adequadas, optando-se pelo **Terraform** e pelo **Ansible**, devido à facilidade de compreensão, à flexibilidade e à ampla documentação disponível.

Por fim, realizou-se um teste de usabilidade da ferramenta por meio de avaliação online. A análise dos resultados permitiu concluir que, embora o protótipo apresente limitações técnicas e requeira melhorias incrementais, ele cumpriu sua função principal e foi considerado satisfatório pelos participantes do experimento.

Ao final, o objetivo do trabalho foi concluído com o desenvolvimento e validação da ferramenta. Dessa forma, ela está disponível gratuitamente no **GitHub** para alunos de universidades públicas, demonstrando a possibilidade de estudar **Kubernetes** sem custos, mesmo sem acesso a recursos de hardware específicos para um estudo satisfatório dessa tecnologia.

REFERÊNCIAS

AWS Academy. Disponível em: <https://aws.amazon.com/pt/training/awsacademy/>. Acesso em: 26 abr. 2023.

AZURE for Students. Disponível em: <https://azure.microsoft.com/pt-br/free/students/>. Acesso em: 26 abr. 2023.

CAPUCCINI, M.; LARSSON, A.; CARONE, M.; NOVELLA, J. A.; SADAWI, N.; GAO, J.; TOOR, S.; SPJUTH, O. On-demand virtual research environments using microservices. **PeerJ Computer Science**, PeerJ Inc., v. 5, p. e232, 2019.

CNCF. **CNCF Annual Survey 2021**. [S. l.], 2021. Disponível em: <https://www.cncf.io/reports/cncf-annual-survey-2021/>. Acesso em: 04 abr. 2023.

IBM. **IBM Cloud free tier**. Disponível em: <https://www.ibm.com/cloud/free>. Acesso em: 26 abr. 2023.

KAPLAREVIC, V. **Understanding Kubernetes Architecture with Diagrams**. 2019. Disponível em: <https://phoenixnap.com/kb/understanding-kubernetes-architecture-diagrams>. Acesso em: 14 mai. 2023.

KUBERNETES, T. Kubernetes. **Kubernetes. Retrieved May**, v. 24, p. 2019, 2019.

LEARNK8S. **Free Kubernetes**. 2023. Disponível em: <https://github.com/learnk8s/free-kubernetes>. Acesso em: 04 abr. 2023.

MUSCHKO, B. **Certified Kubernetes Administrator (CKA) Study Guide**. [S. l.]: "O'Reilly Media, Inc.", 2022.

MUŠIĆ, D.; FORTUNA, C. Kubitect-a solution for on-premise cluster deployment. In: IEEE. **2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)**. [S. l.], 2022. p. 273–278.

ORACLE. **Modo Gratuito da Oracle Cloud**. Disponível em: <https://www.oracle.com/br/cloud/free/>. Acesso em: 26 abr. 2023.

POULTON, N. **The kubernetes book**. [S. l.]: Nigel Poulton, 2023.

PROGRAMA gratuito do Google Cloud. Disponível em: <https://cloud.google.com/free/docs/free-cloud-features?hl=pt-br#free-trial>. Acesso em: 26 abr. 2023.

REDHAT. **O que é infraestrutura como código (IaC)?** 2022. Disponível em: <https://www.redhat.com/pt-br/topics/automation/what-is-infrastructure-as-code-iac>. Acesso em: 18 mai. 2023.

REDHAT. **O que é um container Linux?** 2022. Disponível em: <https://www.redhat.com/pt-br/topics/containers/whats-a-linux-container>. Acesso em: 07 mai. 2023.

SILVA, W. d.; REGO, P.; ALENCAR, J. d.; MATEUS, B. Dvl: Uma ferramenta para criação de laboratórios práticos de disciplinas da área de ti utilizando virtualização baseada em contêineres. In: SIMPOSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 36. **Anais Estendidos** [...]. Porto Alegre, RS: SBC, 2018.

TRY Digital Ocean. Disponível em: <https://try.digitalocean.com/cfe/>. Acesso em: 26 abr. 2023.

TRY Linode. Disponível em: <https://www.linode.com/lp/free-credit-100/>. Acesso em: 26 abr. 2023.

TURNBULL, J. **The Docker Book**:: Containerization is the new virtualization. [S. l.]: James Turnbull, 2014.

VITALINO, J. F. N.; CASTRO, M. A. N. C. **Descomplicando o Docker**. [S. l.]: Brasport, 2018.