



MicroK8s

Trabalho Laboratorial nº 2 (TL2)
Laboratório de Tecnologias de Informação

EI 2022/23

Grupo G14

Rostyslav Romanyshyn nº 2200668

Rodrigo Ferreira nº 2201702

Leiria, maio de 2023



Resumo

O presente relatório tem por objetivo apresentar e descrever de forma detalhada o trabalho que foi desenvolvido na criação de uma aplicação para gestão de Kubernetes com a ajuda da API do microK8s.

Abstract

This report aims to present and describe in detail the work that was developed in the creation of an application for managing Kubernetes with the help of the microK8s API.

Índice

Trata-se de um elemento **obrigatório**. Nota: **o índice nunca figura no índice**.

Resumo.....	iii
Abstract.....	iv
Lista de Figuras.....	vii
1. Introdução	1
2. Fundamentação Teórica	2
3. Bibliotecas usadas no Python	3
4. Kubernetes	4
5. Criação do cenário microK8S	5
5.1. Criação de 3 máquinas virtuais no VMware	5
5.1.1. Criar uma máquina virtual:.....	5
5.1.2. Tipo de configuração	5
5.1.3. Hardware compatível.....	6
5.1.4. Tipo de instalação.....	6
5.1.5. Guest Operation System	7
5.1.6. Nome da máquina virtual	7
5.1.7. Quantidade de processadores virtuais	8
5.1.8. Quantidade de memória RAM.....	8
5.1.9. Opções de rede.....	9
5.1.10. tipo de disco virtual	9
5.1.11. Tipo do disco	10
5.1.12. Escolher o disco virtual	10
5.1.13. Tamanho do disco.....	11
5.1.14. Onde o disco vai ser guardado	11
5.1.15. Instalação do MicroK8s.....	12
6. Ações disponibilizadas no projeto	13
6.1. Controlar mais do que um dispositivo com a mesma aplicação.....	13

6.2.	Dashboard com as informações do cluster.....	14
6.3.	Nodes	15
6.4.	Namespaces	16
6.4.1.	Listar Namespaces.....	16
6.4.2.	Criar Namespaces.....	16
6.4.3.	Eliminar Namespaces	17
6.5.	Pods.....	18
6.5.1.	Listar Pods.....	18
6.5.2.	Criar Pods.....	18
6.5.3.	Eliminar Pods	19
6.6.	Deployments.....	20
6.6.1.	Listar Deployments	20
6.6.2.	Criar Deployments	20
6.6.3.	Eliminar Deployments.....	21
6.7.	Services	22
6.7.1.	Listar Services	22
6.7.2.	Criar Service.....	22
6.7.3.	Eliminal Service	23
7.	Conclusão	25
Bibliografia.....	Error! Bookmark not defined.	

Lista de Figuras

Elemento a figurar, **quando aplicável**.

Figura 3.1-3-Bibliotecas	3
Figura 4.1-Kubernetes	4
Figura 5.1-Criar uma máquina virtual	5
Figura 5.2-Tipo de configuração	5
Figura 5.3-Hardware compatível	6
Figura 5.4-Tipo de instalação	6
Figura 5.5-Guest Operation System	7
Figura 5.6-Nome da máquina virtual	7
Figura 5.7-Quantidade de processadores virtuais	8
Figura 5.8-memória RAM	8
Figura 5.9-Opções de rede.....	9
Figura 5.10-tipo de disco virtual.....	9
Figura 5.11-Tipo do disco	10
Figura 5.12-disco virtual.....	10
Figura 5.13-Tamanho do disco	11
Figura 5.14-Onde o disco vai ser guardado.....	11
Figura 6.1-IP/Port	13
Figura 6.2-Dashboard	14
Figura 6.3 - Informação dos Nodes	15
Figura 6.4-Namespaces	16
Figura 6.5-Criar Namespaces	16
Figura 6.6-Eliminar Namespaces.....	17
Figura 6.7-Listar Pods	18
Figura 6.8-Criar Pods	18
Figura 6.9-Eliminar Pods.....	19
Figura 6.10-Listar Pods	20

Figura 6.11-Criar Deployments.....	20
Figura 6.12-Eliminar Deployments.....	21
Figura 6.-Listar Services	22
Figura 6.14-Criar Service.....	22

1. Introdução

O presente trabalho foi realizado no âmbito da unidade curricular de Laboratório de Tecnologias de Informação, do 3º ano do curso de Engenharia Informática, na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Para a implementação, foi necessário a implementação de um cenário Kubernetes com 1 master e 2 workers e o desenvolvimento de uma aplicação para gestão de Kubernetes através da API microK8s com a ajuda do Python para comunicar com o dispositivo através da sua API REST.

2. Fundamentação Teórica

Para a obtenção de conhecimento das técnicas de desenvolvimento referentes às tecnologias abordadas neste projeto, foram utilizados meios de pesquisa, com o objetivo de reunir conhecimento sobre as tecnologias nas áreas de programação e desenvolvimento.

As tecnologias pesquisadas foram:



- **Python:** Python é uma linguagem de programação que, segundo Van Rossum (2003), é multi-paradigma e gerenciada por uma organização sem fins lucrativos. O Python foi lançado em 1991 e a sua estrutura permite uma comunicação sólida com o equipamento, além de ótimos atributos para aplicações no ramo da matemática, eletrônica e física. Esta linguagem foi utilizada neste projeto para programar os componentes ligados ao Raspberry PI.



- **Postman:** Postman é uma ferramenta que dá suporte à documentação das requisições feitas pela API. Ele possui ambiente para a documentação, execução de testes de APIs e requisições em geral.



- **Kubrnets:** O Kubernetes é uma plataforma de código aberto para automatizar, implantar, escalar e gerenciar aplicativos em contêineres. Ele fornece um ambiente robusto para a orquestração de containers, permitindo que as organizações gerenciem eficientemente suas cargas de trabalho em ambientes de produção.

3. Bibliotecas usadas no Python

Usamos a biblioteca tkinter, para a construção das interfaces gráficas que iram disponibilizar a criação de páginas, botões, e labels, e as bibliotecas matplotlib e numpy para criar gráficos.

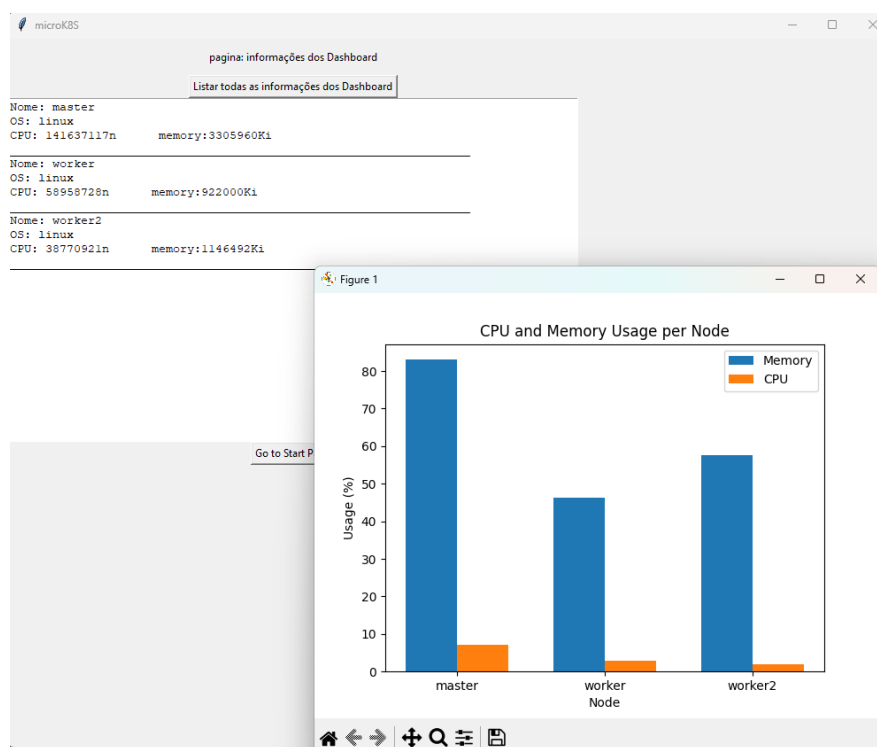


Figura 3.1-3-Bibliotecas

4. Kubernetes

	MICROK8S	K3S	MINIKUBE
CNCF certified	✓	✓	✓
Vanilla Kubernetes	✓	–	✓
Architecture support	x86, ARM64, s390x, POWER9	x86, ARM64, ARMhf	x86, ARM64, ARMv7, ppc64, s390x
Enterprise support	✓	✓	–
Single-node support	✓	✓	✓
Multi-node cluster support	✓	✓	–
Automatic high availability	✓	✓	–
Automatic updates	✓	✓	–
Memory requirements	540 MB	512 MB	644 MB
Add-on functionality	✓	–	✓
Container runtime	containerd, kata	CRI-O	Docker, containerd, CRI-O
Networking	Calico, Cilium, CoreDNS, Traefik, NGINX, Ambassador, Multus, MetalLB	Flannel, CoreDNS, Traefik, Canal, Klipper	Calico, Cilium, Flannel, ingress, DNS, Kindnet
Default storage options	Hostpath storage, OpenEBS, Ceph	Hostpath storage, Longhorn	Hostpath storage
GPU acceleration	✓	✓	✓

Figura 4.1-Kubernetes

Decidimos escolher o microK8S pois é o que tem mais funcionalidades, diferente dos seus concorrentes, e também sendo uma distribuição leve do Kubernetes pronta para produção e que pode ser executado em qualquer sistema operacional.

No geral, o microK8s é um pouco mais complicado de usar do que o K3s ou o Minikube, principalmente porque possui uma arquitetura modular e executa apenas um conjunto mínimo de serviços por padrão.

Concluindo:

- Minikube é o mais fácil de usar, mas não é uma solução de nível de produção.
- K3s é a distribuição leve de nível de produção mais fácil.
- O MicroK8s oferece o maior grau de controle, mas é um pouco mais difícil de instalar e configurar do que as outras distribuições.

5. Criação do cenário microK8S

5.1. Criação de 3 máquinas virtuais no VMware

5.1.1. Criar uma máquina virtual:

No menu principal do VMware Workstation, clique em "File" (Arquivo) e selecione "New Virtual Machine" (Nova Máquina Virtual). Uma janela será aberta.

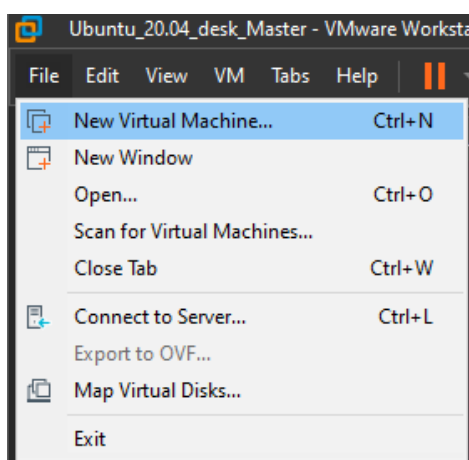


Figura 5.1-Criar uma máquina virtual

5.1.2. Tipo de configuração

Aqui vamos escolher entre várias opções de configuração, como "Typical" ou "Custom". Para este guia, vamos escolher a opção "Custom", que simplifica o processo.

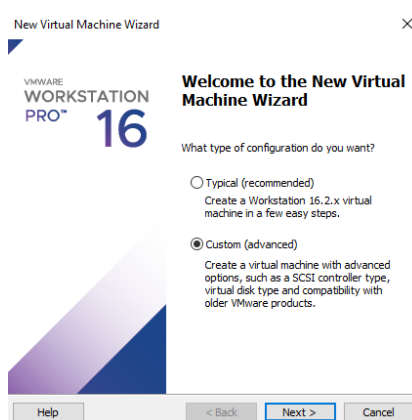


Figura 5.2-Tipo de configuração

5.1.3. Hardware compatível

Selecione o hardware compatível com sua máquina virtual. Selecione a opção desejada e clique em "Next" (Próximo).

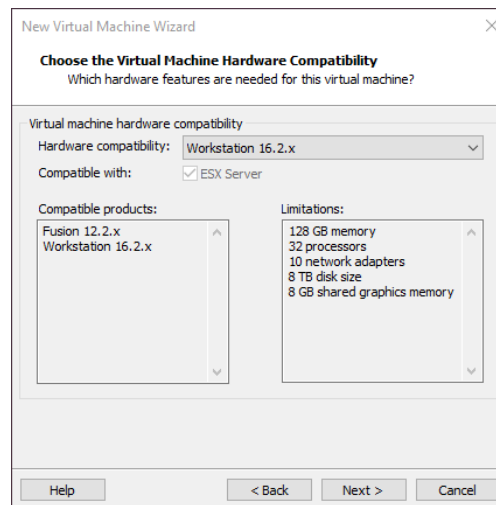


Figura 5.3-Hardware compatível

5.1.4. Tipo de instalação

Agora selecionar o tipo de instalação do sistema operacional. Escolha a opção "I will install the operating system later" e clique em "Next".

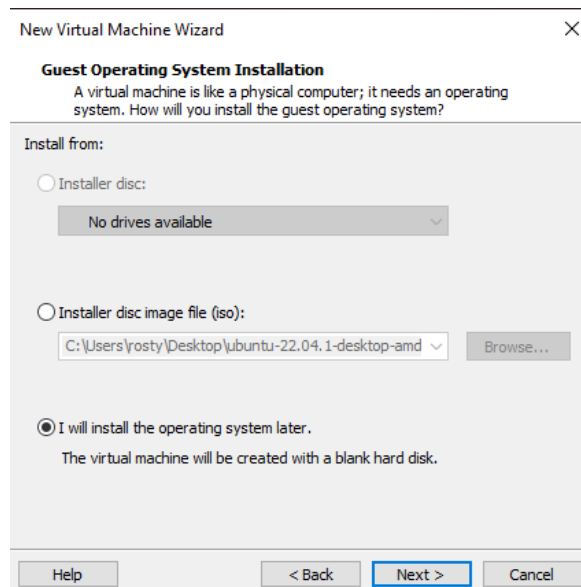


Figura 5.4-Tipo de instalação

5.1.5. Guest Operation System

Neste passo seleccionar Linux pois vamos instalar ubuntu de seguida clicar em Next.

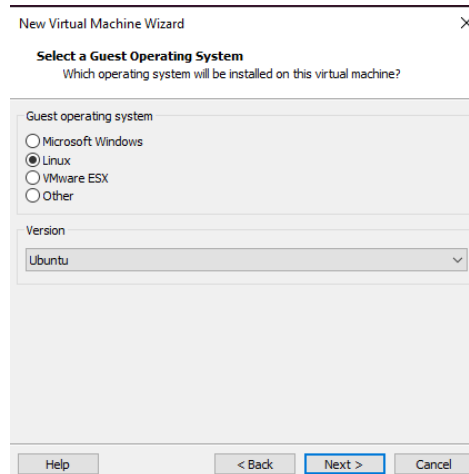


Figura 5.5-Guest Operation System

5.1.6. Nome da máquina virtual

Digite um nome para sua máquina virtual e especifique o local onde deseja salvá-la.

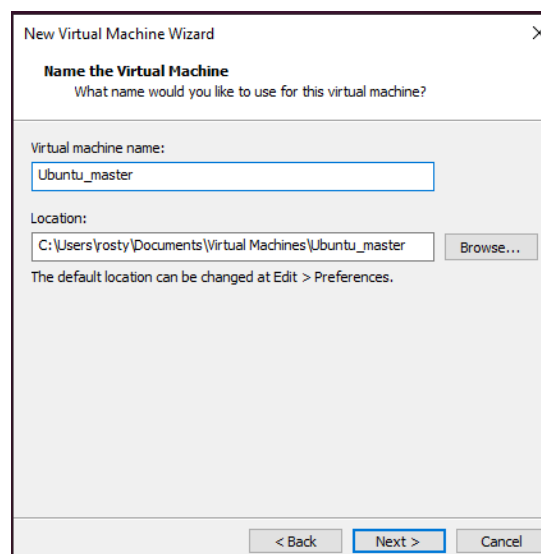


Figura 5.6-Nome da máquina virtual

5.1.7. Quantidade de processadores virtuais

Defina o número de processadores virtuais que você deseja alocar para a máquina virtual.

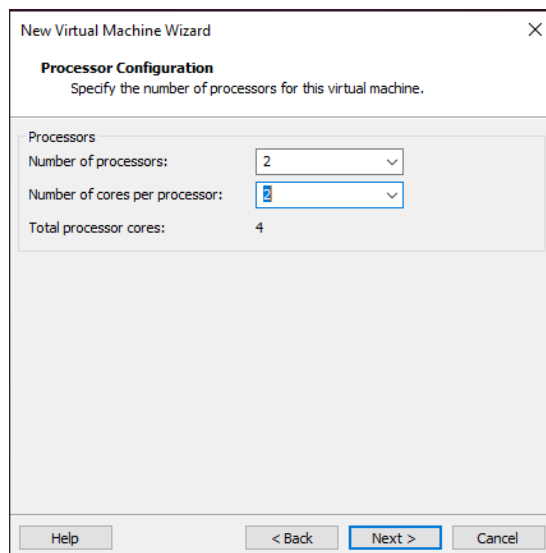


Figura 5.7-Quantidade de processadores virtuais

5.1.8. Quantidade de memória RAM

Ajuste a quantidade de memória RAM que você deseja alocar para a máquina virtual.

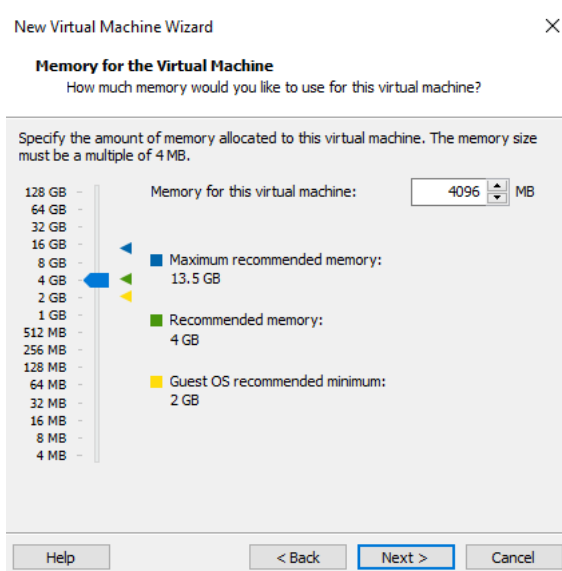


Figura 5.8-memória RAM

5.1.9. Opções de rede

Selecione as opções de rede para a máquina virtual. Clique em NAT e de seguida em "Next" (Próximo) quando terminar.

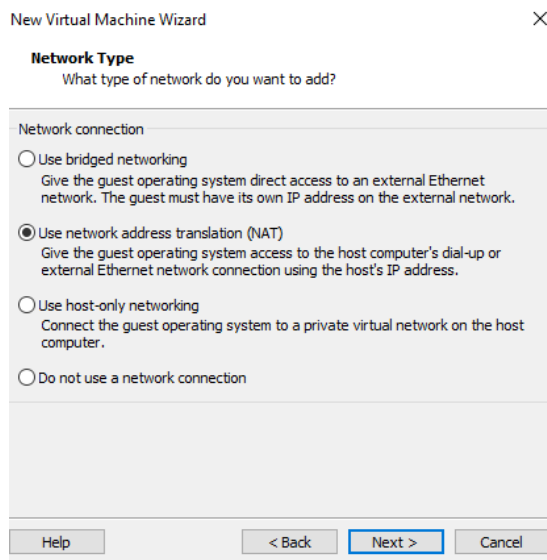


Figura 5.9-Opções de rede

5.1.10. tipo de disco virtual

Selecione o tipo de disco virtual que você deseja criar para a máquina virtual e de seguida clique em Next.

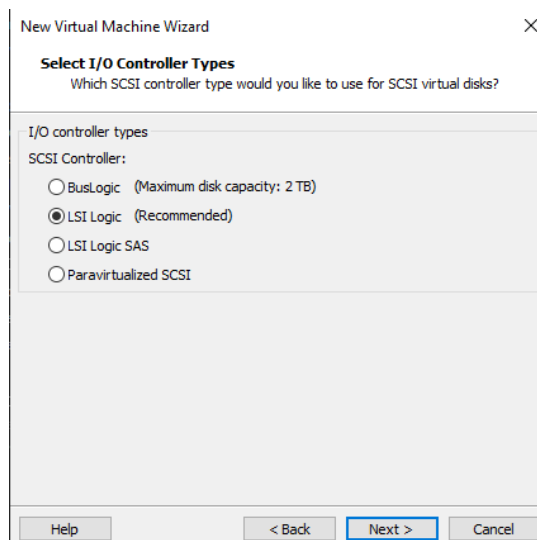


Figura 5.10-tipo de disco virtual

5.1.11. Tipo do disco

Defina o tipo do disco neste passo vai ser SCSI e clique em Next.

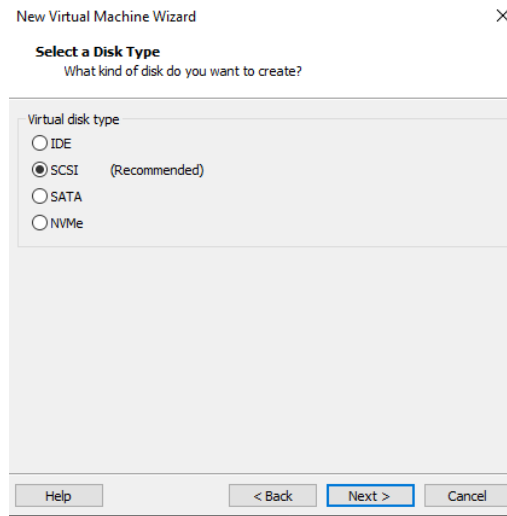


Figura 5.11-Tipo do disco

5.1.12. Escolher o disco virtual

Escolher criar um disco novo e clique em Next.

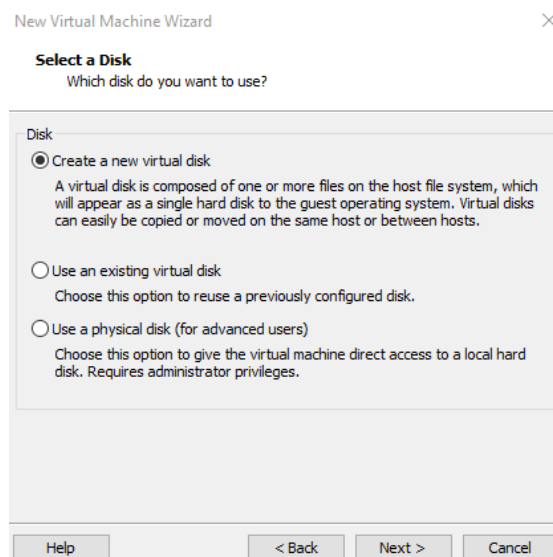


Figura 5.12-disco virtual

5.1.13. Tamanho do disco

Defina o tamanho do disco virtual para a máquina virtual. Insira o tamanho desejado e clique em Next.

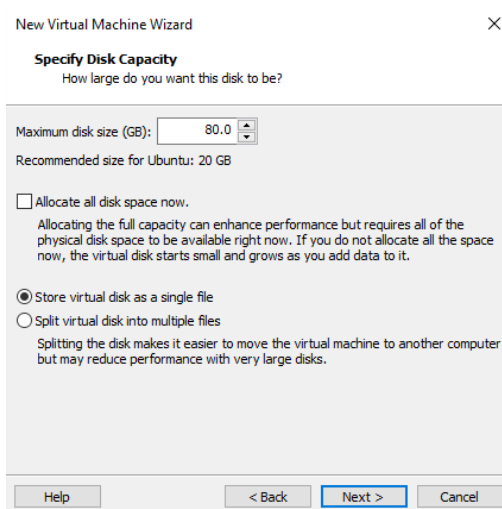


Figura 5.13-Tamanho do disco

5.1.14. Onde o disco vai ser guardado

Defina a localização do disco onde ele vai ser guardado na máquina física e pressionar Next por fim clicar no Finish.

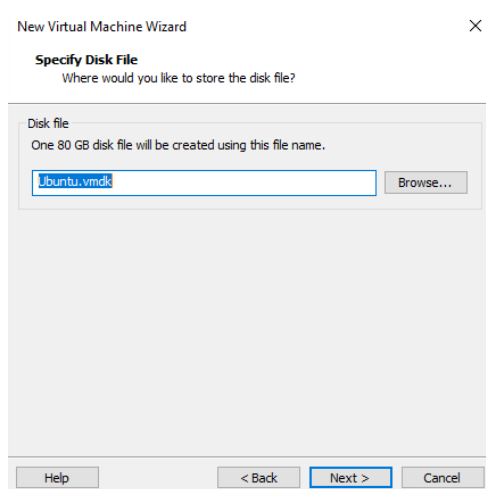


Figura 5.14-Onde o disco vai ser guardado

5.1.15. Instalação do MicroK8s

Depois de instalar as 3 máquinas virtuais temos de dar update nelas com o comando:

- Sudo apt-get update

De seguida podemos instalar o MicroK8s:

- sudo snap install microk8s --classic

Configurar o Master Node:

- hostnamectl set-hostname master

Configurar nos workers Nodes:

- hostnamectl set-hostname worker1
- hostnamectl set-hostname worker2

Configurar o Master Node:

- sudo microk8s add-node
- sudo nano /etc/hosts -> mudar o ficheiro e adicionar “IP worker1”
“ IP worker2”

Configurar nos workers Nodes:

- sudo microk8s join x.x.x.x:y/xptoabcd12345 --worker

Configurar o Master Node:

- apt-get install socat
- (sudo socat TCP-LISTEN:8443,reuseaddr,fork TCP:127.0.0.1:10443)&
- sudo microk8s kubectrl proxy --address=0.0.0.0 --accept-hosts=^* &

6. Ações disponibilizadas no projeto

existem os seguintes métodos:

- Uma dashboard com as informações do cluster
- Nodes: listar
- Namespaces: listar, criar e eliminar
- Pods: listar, criar e eliminar
- Deployments: listar, criar e eliminar
- Services: listar, criar e eliminar

6.1. Controlar mais do que um dispositivo com a mesma aplicação

Para controlar mais do que um microK8s acrescentamos na página principal 2 Labels, um para inserir o IP e o outro para a porta.



The screenshot shows the microK8S start page. At the top left is the microK8S logo. The main content area has a light gray background. It starts with the text "This is the start page". Below this, there are two input fields. The first is labeled "ip:" and contains the text "192.168.126.130". The second is labeled "port:" and contains the text "8001". Below these fields is a "Submit" button. At the bottom of the form, the URL "http://192.168.126.130:8001/" is displayed.

Figura 6.1-IP/Port

6.2. Dashboard com as informações do cluster

Para visualizar a Dashboard foi criada a página “Dashboard” que irá mostrar todos os recursos utilizados nos nodes.

Nessa tal página foi criado um botão que ao premir irá mostrar uma Dashboard como mostra na figura abaixo.

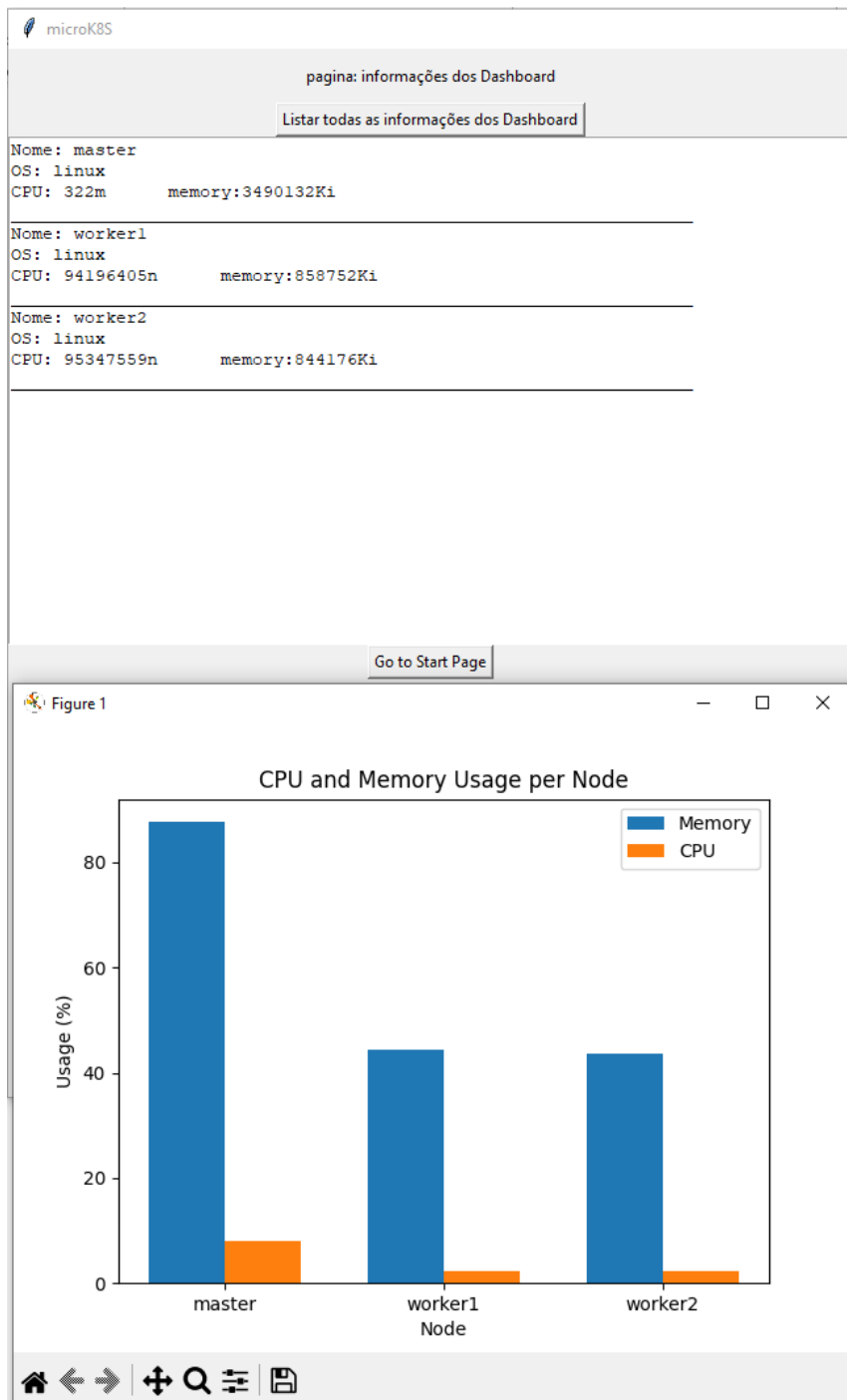


Figura 6.2-Dashboard

6.3.Nodes

Na página de informação dos nodes é possível visualizar as informações mais importantes dos nodes como o nome, número de cores, espaço de armazenamento e memória de cada node do cluster kubernetes.

pagina: informações dos Nodes

Listar todas as informações dos Nodes

Nome: worker2
Cpu: 2
Space: 10218772Ki = 9.745380401611328GB
RAM: 1988232Ki = 1.8961257934570312GB

Nome: worker
Cpu: 2
Space: 10218772Ki = 9.745380401611328GB
RAM: 1988232Ki = 1.8961257934570312GB

Nome: master
Cpu: 2
Space: 50770432Ki = 48.41845703125GB
RAM: 3982120Ki = 3.7976455688476562GB

Go to Start Page

Figura 6.3 - Informação dos Nodes

6.4.Namespaces

6.4.1.Listar Namespaces

Para listar todos os namespaces foi criada a página “Namespaces” que irá listar as informações básicas como o nome, a label, o estado e a data de criação.

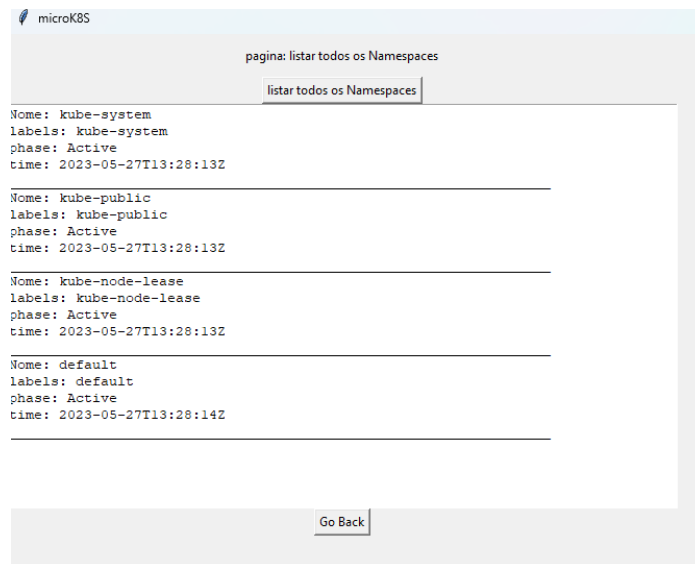


Figura 6.4-Namespaces

6.4.2.Criar Namespaces

Para criar os namespaces foi criada a página “criar Namespace” que irá criar um namespace depois de inserir na label um nome para tal.

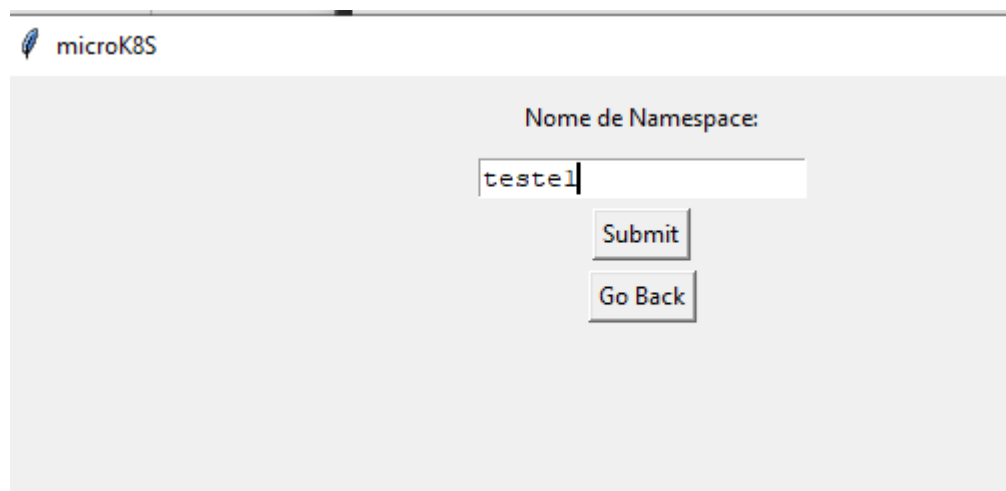
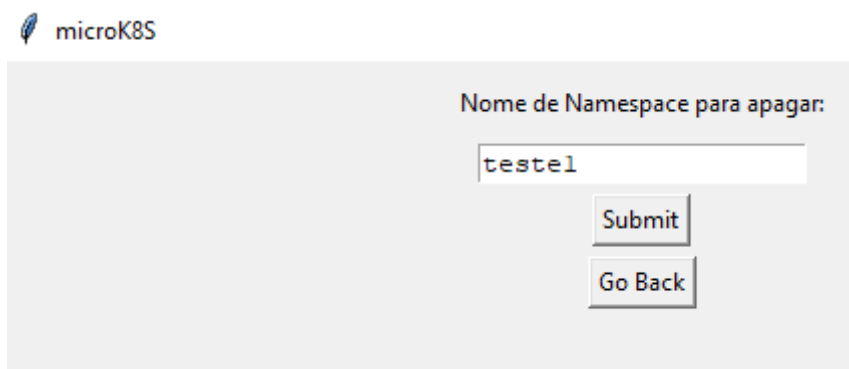


Figura 6.5-Criar Namespaces

6.4.3. Eliminar Namespaces

Para eliminar os namespaces foi criada a página “eliminar Namespaces” que irá eliminar o namespace inserido na label



microK8S

Nome de Namespace para apagar:

testel

Submit

Go Back

Figura 6.6-Eliminar Namespaces

6.5.Pods

6.5.1.Listar Pods

Para listar todos os Pods foi criada a página “Pods” que irá listar as informações básicas de cada Pod tais como o nome, imagem, node, estado e a data de criação

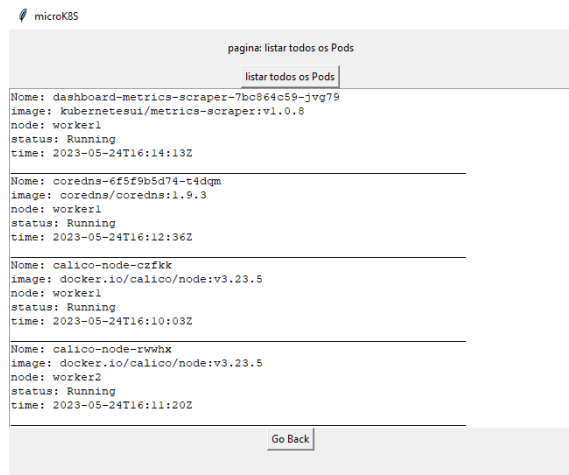


Figura 6.7-Listar Pods

6.5.2. Criar Pods

Para criar os Pods foi criada a página “criar Pods” que irá criar um Pod depois de inserir na label um Namespace, nome da imagem, versão, container port e host port

microK8S

Nome de Namespace:

testel

Nome da imagem para instalar:

nginx

versão:

(no caso de não meter nenhuma vai ser latest)

container port:

80

Host port:

8080

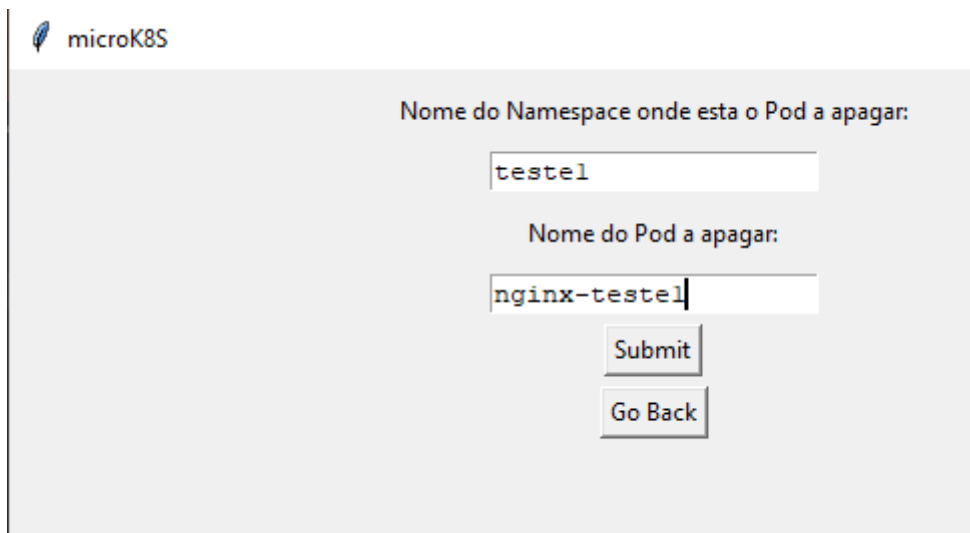
Submit

Go Back

Figura 6.8-Criar Pods

6.5.3. Eliminar Pods

Para eliminar um Pod foi criada a página “eliminar Pod” que irá eliminar um Pod para poder eliminar temos de inserir o namespace onde está o Pod e nome do Pod.



microK8S

Nome do Namespace onde esta o Pod a apagar:

testel

Nome do Pod a apagar:

nginx-testel

Submit

Go Back

Figura 6.9-Eliminar Pods

6.6.Deployments

6.6.1.Listar Deployments

Para listar todos os Deployments foi criada a página “Deployments” que irá listar as informações básicas de cada Deployment tais como o nome, imagem, pods e a data de criação.

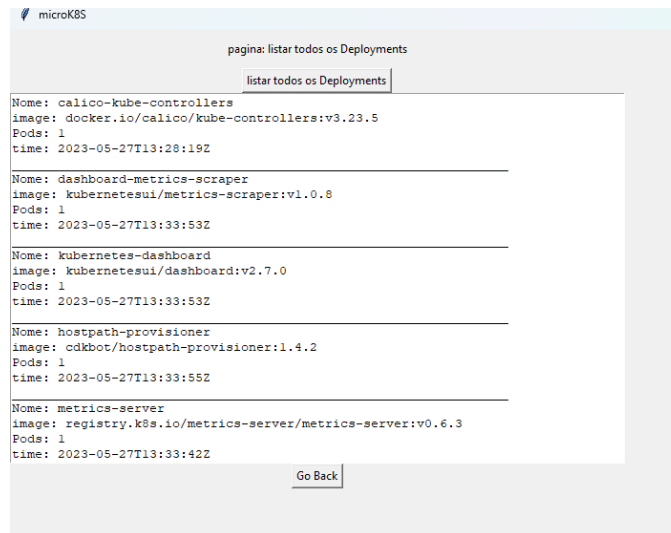


Figura 6.10-Listar Pods

6.6.2. Criar Deployments

Para criar os Deployments foi criada a página “criar Deployment” que irá criar um Deployment depois de inserir na label um Namespace existente, número de réplicas, imagem, versão, container port e host port.

microK8S

Nome de Namespace:

default

Numero de replicas:

1

Nome da imagem para instalar:

nginx

versão:
(no caso de não meter nenhuma vai ser latest)

container port:

80

Host port:

8080

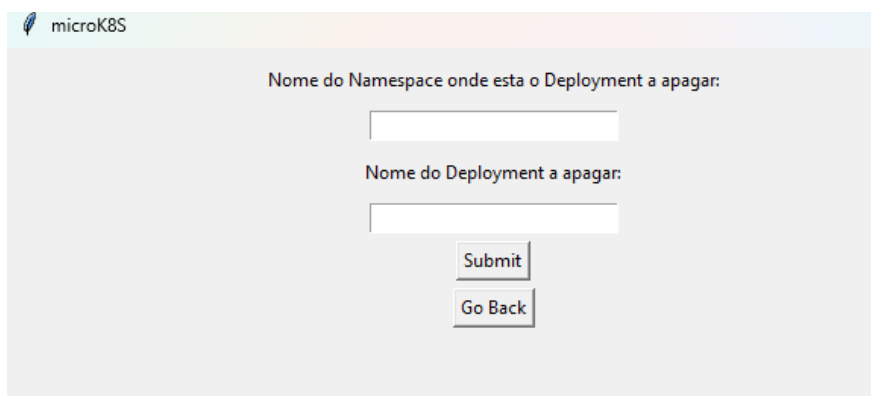
Submit

Go Back

Figura 6.11-Criar Deployments

6.6.3. Eliminar Deployments

Para eliminar um Deployment foi criada a página “eliminar Deployment” que irá eliminar um Deployment para poder eliminar temos de inserir o namespace onde está o Deployment e nome do Deployment.



The screenshot shows a web interface for 'microK8S'. It features a form with two input fields and two buttons. The first input field is labeled 'Nome do Namespace onde esta o Deployment a apagar:' and the second is labeled 'Nome do Deployment a apagar:'. Below the second input field are two buttons: 'Submit' and 'Go Back'.

Figura 6.12-Eliminar Deployments

6.7.Services

6.7.1.Listar Services

Para listar todos os Services foi criada a página “Services” que irá listar as informações básicas de cada Service tais como o nome, tipo, Internal Endpoint e a data de criação

6.7.2. Criar Service

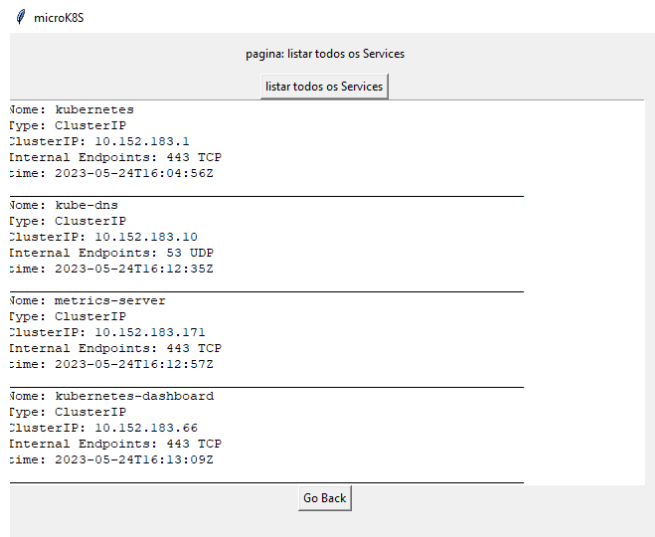


Figura 6.13-Listar Services

Para criar um Service foi criada a página “criar Service” que irá criar um Service depois de inserir na label um Nome do Service, Namespace, Protocolo, Port e Target port.

The screenshot shows the 'criar Service' form in the microK8S interface. The form has the following fields and values:

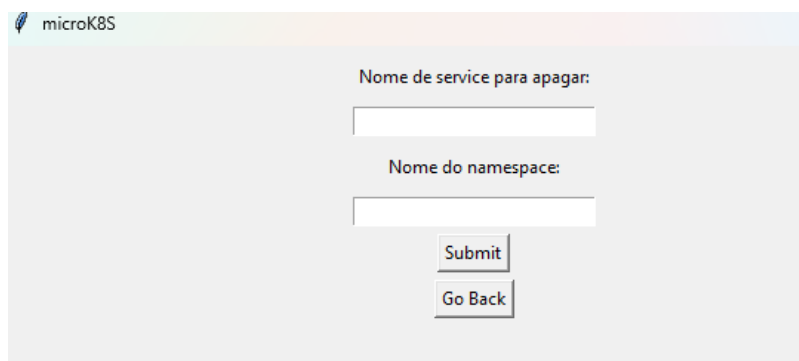
- Nome do Service: teste
- Namespace: default
- Protocol: TCP
- Port: 80
- Target Port: 8080

At the bottom of the form are two buttons: 'Submit' and 'Go Back'.

Figura 6.14-Criar Service

6.7.3. Eliminal Service

Para eliminar um Service foi criada a página “eliminar Service” que irá eliminar um Service, para poder eliminar temos de inserir o namespace onde está o Service e nome do Service.



The screenshot shows a web interface for microK8S. At the top left, there is a logo and the text 'microK8S'. The main content area has a light gray background. It contains two text input fields. The first field is labeled 'Nome de service para apagar:' and the second field is labeled 'Nome do namespace:'. Below these fields are two buttons: 'Submit' and 'Go Back'.

7. Análise crítica e proposta de melhorias

No âmbito do projeto foi feito um trabalho funcional, e que permite as mínimas configurações básicas em Kubernetes.

Como qualquer solução existe sempre uma margem de melhoria não sendo diferente com o nosso trabalho desenvolvido. Dentro das possíveis melhorias as que se destacam mais são a utilização de uma interface mais bonita e fácil de usar, ter a dashboard a dar update automaticamente e a opção de configurar mais campos em cada configuração.

8. Conclusão

Com base no que foi apresentado, pode-se concluir que o trabalho foi concluído com sucesso, tendo sido implementado um cenário Kubernetes e criado uma aplicação para controlo de Kubernetes funcional em Python que permite interagir com qualquer cenário Kubernetes `microk8s`.

O trabalho contribuiu para novos aprendizados e a consolidação dos mesmos tais como a consolidação de conhecimentos em criação de API Rest, Kubernetes, e a criação de uma aplicação para gerir kubernetes.