

Projeto: Provisionamento Local de Kubernetes com Docker e Rancher

Aluno: Anderson Luiz Pancheniak (anderson.pancheniak@outlook.com)

Matrícula: 358342

Curso: MBA Cloud Engineering & Architecture

Disciplina: Gerenciamento de APIs

Professor: Claudio Eduardo de Oliveira

Este projeto visa o provisionamento de um cluster Kubernetes local utilizando Docker e Rancher, com a implantação automatizada via GitOps utilizando ArgoCD. O projeto também incorpora o Kong API Gateway para gerenciamento de APIs e um serviço de exemplo para validação da configuração.

Repositório GitHub: andersonluizp/fiap

> No repositório, encontra-se um vídeo explicativo detalhando todo o processo descrito abaixo.

Ferramentas Utilizadas

- **Docker:** v25.0.3
- **Rancher:** v2.9.3
- **WSL2:** Subsistema Windows para Linux
- **Postman:** v11.18.0
- **Kubectl CLI:** v1.29.1
- **ArgoCD:** v2.12.6+4dab5bd

Passo a Passo da Configuração

1. Inicializar Rancher em Contêiner Docker

Execute o Rancher como contêiner para gerenciar o Kubernetes:

```
docker run -d --restart=unless-stopped \
  -p 8080:80 -p 8443:443 \
  --privileged \
```

```
rancher/rancher:latest
```

Recupere a senha de bootstrap:

```
docker logs container-id 2>&1 | grep "Bootstrap Password:"
```

Dica: Substitua container-id pelo ID do contêiner Rancher.

2. Configurar o Kubeconfig para Uso do Kubectl

Copie o arquivo kubeconfig para acessar o cluster Kubernetes através dos comandos kubectl.

3. Instalar ArgoCD no Cluster

Crie o namespace do ArgoCD:

```
kubectl create namespace argocd
```

Aplique o manifesto para instalar o ArgoCD:

```
kubectl apply -f https://raw.githubusercontent.com/andersonluzp/fiap/refs/heads/main/MBA%20Cloud%20Engineering%20%26%20Archi
```

Recupere a senha de acesso inicial do ArgoCD:

```
kubectl get secret -n argocd argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d
```

Redirecione a porta para acessar o ArgoCD no navegador:

```
kubectl port-forward svc/argocd-server -n argocd 8888:80
```

Acesse o ArgoCD no navegador em <https://localhost:8888/>.

4. Configuração do Repositório para GitOps

Configure o repositório em Settings > Repositories dentro do ArgoCD para obter os manifestos de implantação: <https://github.com/andersonluzp/fiap.git>

5. Estrutura dos Manifestos Helm

Os manifestos estão organizados no formato de templates Helm nas seguintes pastas:

- **k8s/manifests**: Contém todos os manifestos para os recursos do cluster.
- **kong**: API Gateway Kong (v2.7).
- **hello-world**: Serviço de teste de comunicação com o Kong API Gateway.

6. Validação do Kong API Gateway

Redirecione o tráfego para acessar o serviço através do Kong API Gateway. Após a configuração bem-sucedida, uma resposta com statusCode 200 será retornada, e ao inspecionar os cabeçalhos, você verá as seguintes entradas:

Via: kong/2.7.2

X-Kong-Upstream-Latency: 2

X-Proxy-Latency: 2020

7. Apêndice

Imagens

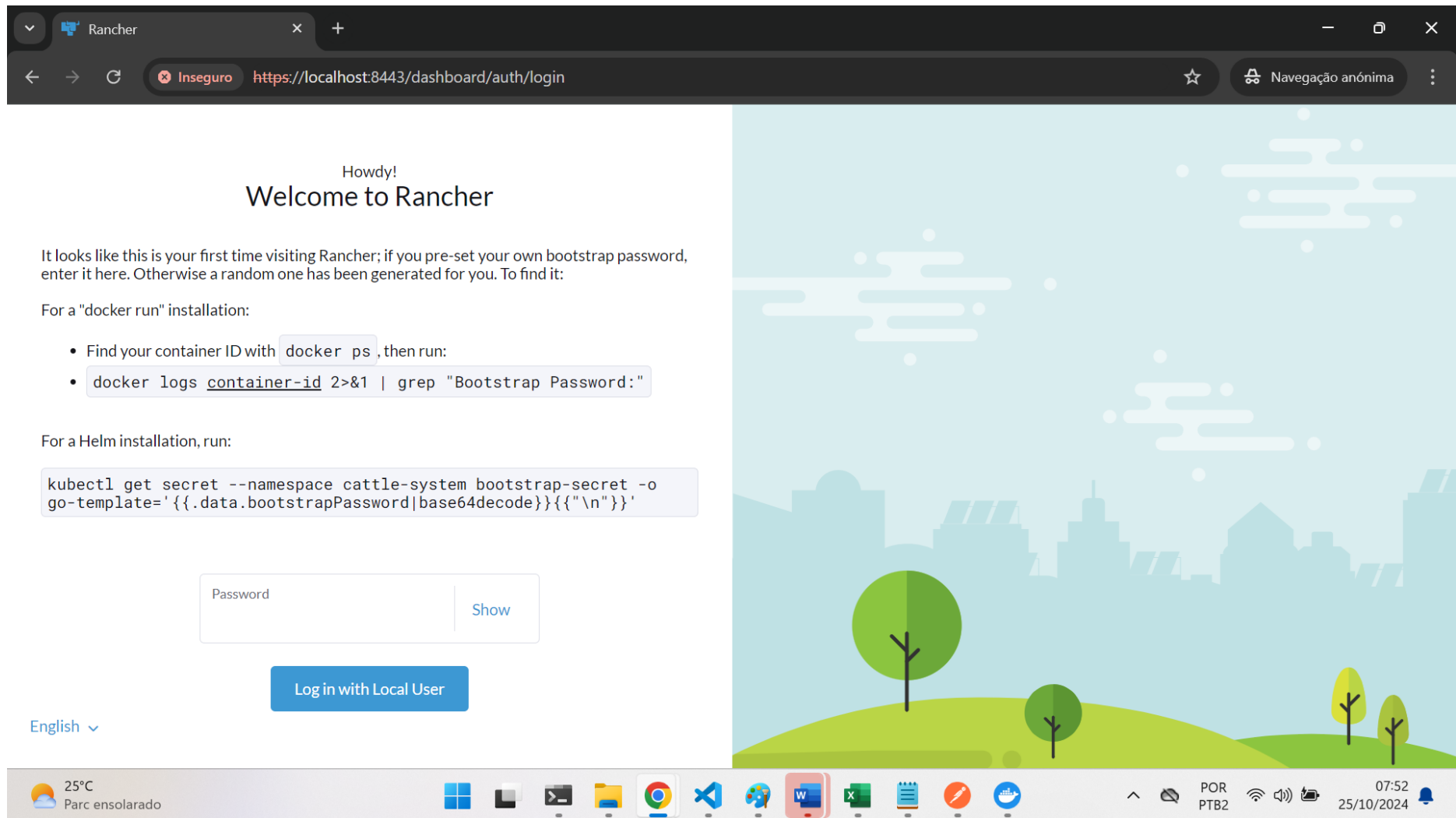


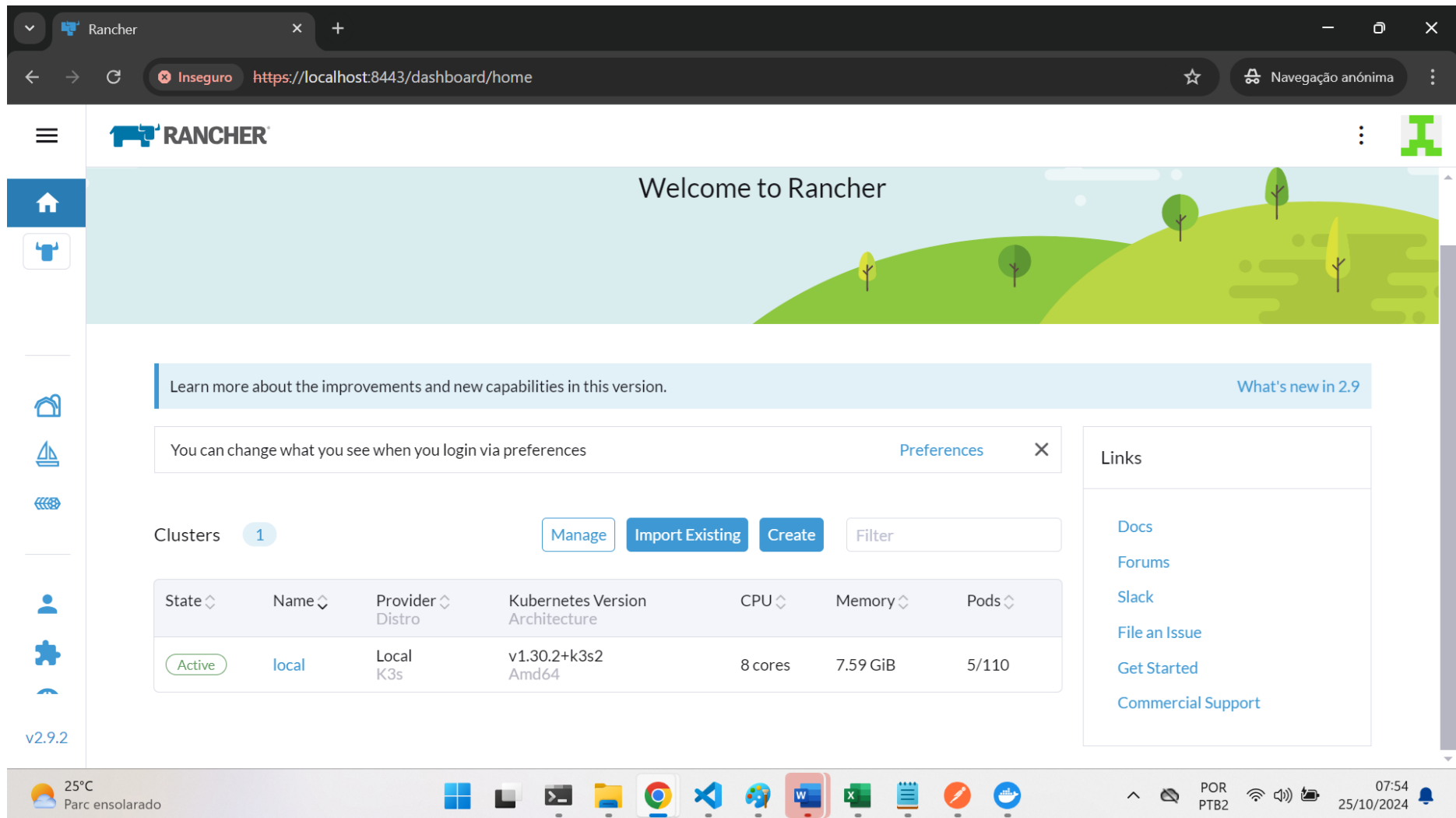
Figure 1: Rancher tela inicial

```
~/fiap/fiap
→ fiap git:(main) X docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
2b9191d9a29a   rancher/rancher:latest             "entrypoint.sh"         About a minute Up About a minute 0.0.0.0:8080->80/tcp, 0.0.0.0:8443->443/tcp hungry_bohr
→ fiap git:(main) X docker logs 2b 2>&1 | grep "Bootstrap Password:"
2024/10/25 10:52:18 [INFO] Bootstrap Password: k2xsnksjt77vd5x4j87xcx9qt2qt79f96bz7s2k7c6knkpdsctlqdv
→ fiap git:(main) X
```

25°C
Parc ensolarado

POR PTB2 07:53 25/10/2024

Figure 2: Rancher obtendo a senha de acesso



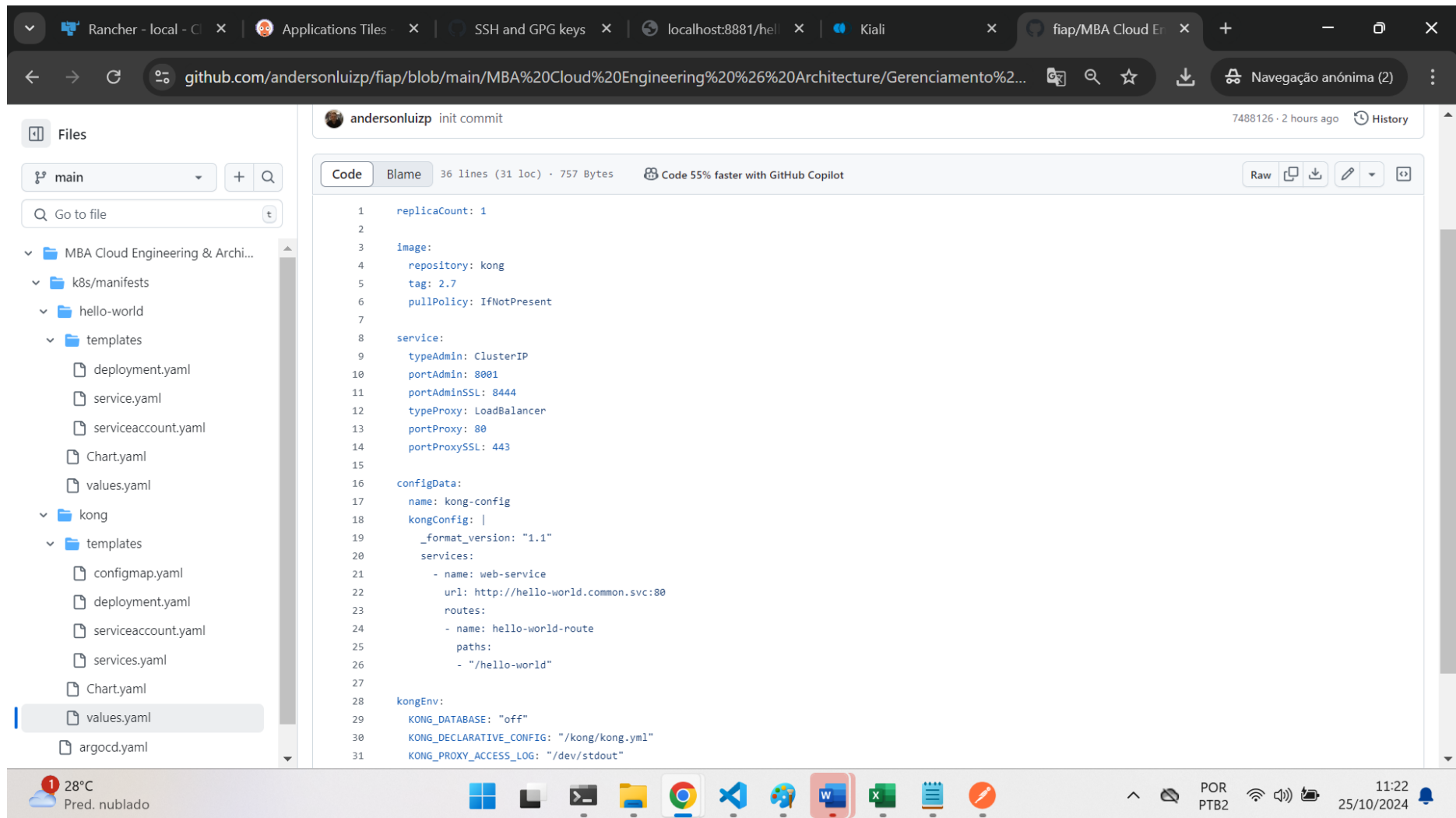


Figure 4: Repositório que contém os arquivos de manifestos Helm a serem utilizados

fiap/README.md at main · and... x | Rancher - local - kong-api-gate... x | kong-api-gateway - Application... x | raw.githubusercontent.com/de... x | +

Inseguro https://localhost:8888/applications/argocd/kong-api-gateway?view=tree&resource=

argo v2.12.0-4dab5bd

Applications

Settings

User Info

Documentation

NAME

NAME

KINDS

KINDS

SYNC STATUS

☐ Synced 5

☐ OutOfSync 0

HEALTH STATUS

☐ Healthy 5

☐ Progressing 0

☐ Degraded 0

☐ Suspended 0

☐ Missing 0

☐ Unknown 0

APPLICATION DETAILS TREE

Log out

DETAILS

DIFF

SYNC

SYNC STATUS

HISTORY AND ROLLBACK

DELETE

REFRESH

APP HEALTH

Healthy

SYNC STATUS

Synced to HEAD (f47cb6b)

Auto sync is enabled.

Author: Anderson Luiz Pancheniak <anderson.pancheniak@outlook.com>

Comment: init commit

LAST SYNC

Sync OK to 991c9d1

Succeeded an hour ago (Fri Oct 25 2024 14:58:05 GMT-0300)

Author: Anderson Luiz Pancheniak <anderson.pancheniak@outlook.com>

Comment: init commit

Diagram showing the deployment of Kong API Gateway components:

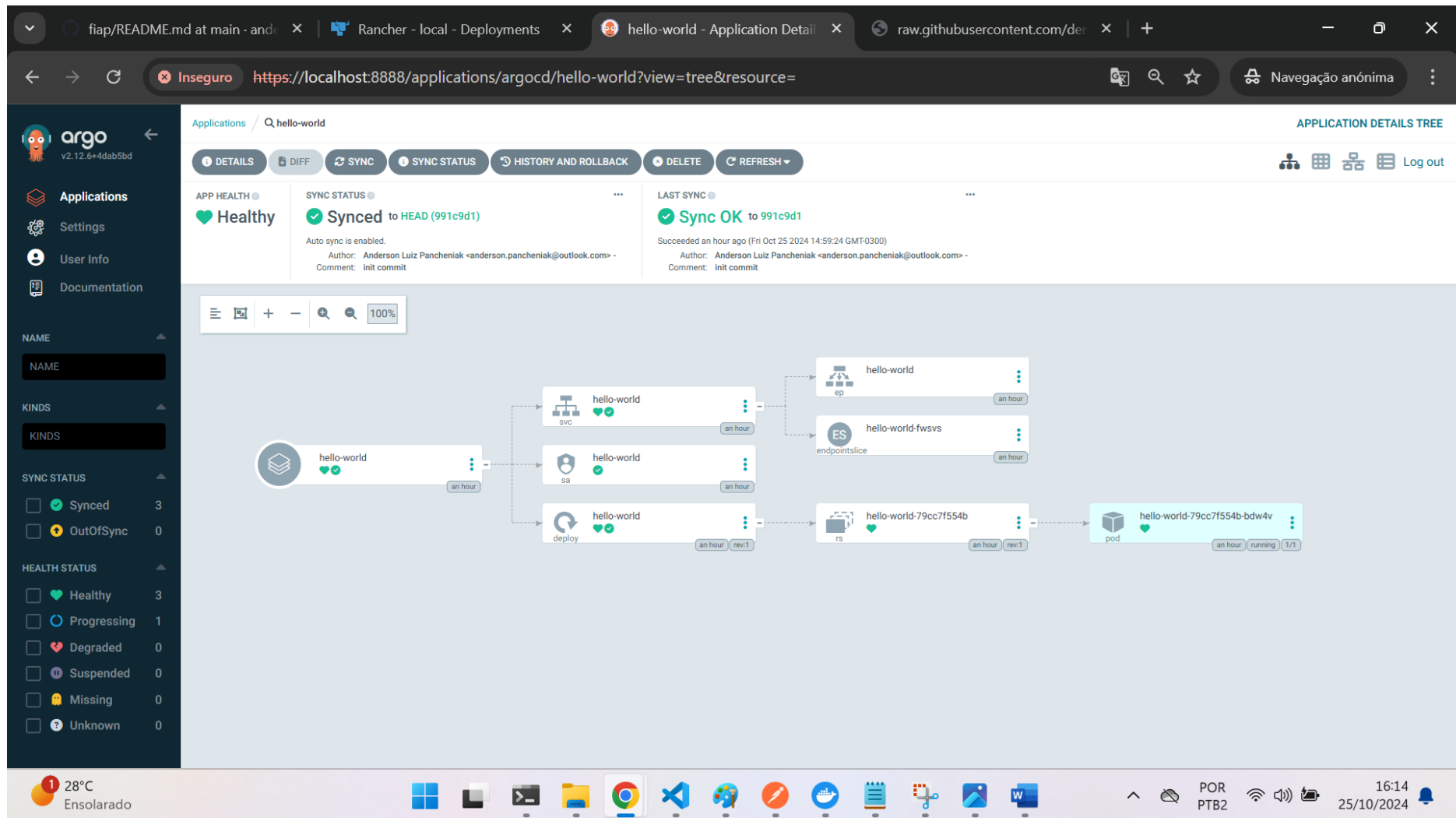
- kong-config (cm)
- kong-api-gateway-admin (api)
- kong-api-gateway-admin-x7kk2 (endpoint/slice)
- kong-api-gateway-proxy (api)
- kong-api-gateway-proxy-rtrfv (endpoint/slice)
- kong-api-gateway (api)
- kong-api-gateway-f664fc65d (api)
- kong-api-gateway-f664fc65d... (pod)

28°C Ensolarado

POR PTB2

16:16 25/10/2024

Figure 5: Argo implantação Kong Api Gateway



Rancher - local - Pods Applications Tiles - Argo CD fiap/MBA Cloud Engineering & +

Inseguro https://localhost:8443/dashboard/c/local/explorer/pod

local Only User Namespaces

Cluster > Workloads > CronJobs 0 DaemonSets 0 Deployments 8 Jobs 0 StatefulSets 1 Pods 9 Apps > Service Discovery > Storage > Policy > More Resources >

v2.9.2

Pods ☆

Download YAML Delete

Create

Filter

State	Name	Namespace	Image	Ready	Restarts	IP	Node	Age
Running	argocd-application-controller-0	argocd	quay.io/argoproj/argocd:v2.12.6	1/1	0	10.42.0.31	local-node	24 mins
Running	argocd-applicationset-controller-75d8c9495-zj4mx	argocd	quay.io/argoproj/argocd:v2.12.6	1/1	0	10.42.0.26	local-node	24 mins
Running	argocd-dex-server-7c9b44b9f9-v6kwg	argocd	ghcr.io/dexidp/dex:v2.38.0	1/1	0	10.42.0.25	local-node	24 mins
Running	argocd-notifications-controller-77f49c7745-2fhd6	argocd	quay.io/argoproj/argocd:v2.12.6	1/1	0	10.42.0.27	local-node	24 mins
Running	argocd-redis-575c96bc4f-ffztw	argocd	redis:7.0.15-alpine	1/1	0	10.42.0.28	local-node	24 mins
Running	argocd-repo-server-7f44b474d7-mwmpt	argocd	quay.io/argoproj/argocd:v2.12.6	1/1	0	10.42.0.29	local-node	24 mins
Running	argocd-server-5f4dd5d648-fwfkz	argocd	quay.io/argoproj/argocd:v2.12.6	1/1	0	10.42.0.30	local-node	24 mins
Running	hello-world-79cc7f554b-9lr87	common	nginxdemos/hello:plain-text	1/1	0	10.42.0.34	local-node	2.7 mins
Running	kong-api-gateway-f664fc65d-8ldnx	kong	kong:2.7	1/1	0	10.42.0.33	local-node	7 mins

25°C Parc ensolarado

POR PTB2 08:26 25/10/2024

Figure 7: Rancher visualização dos pods

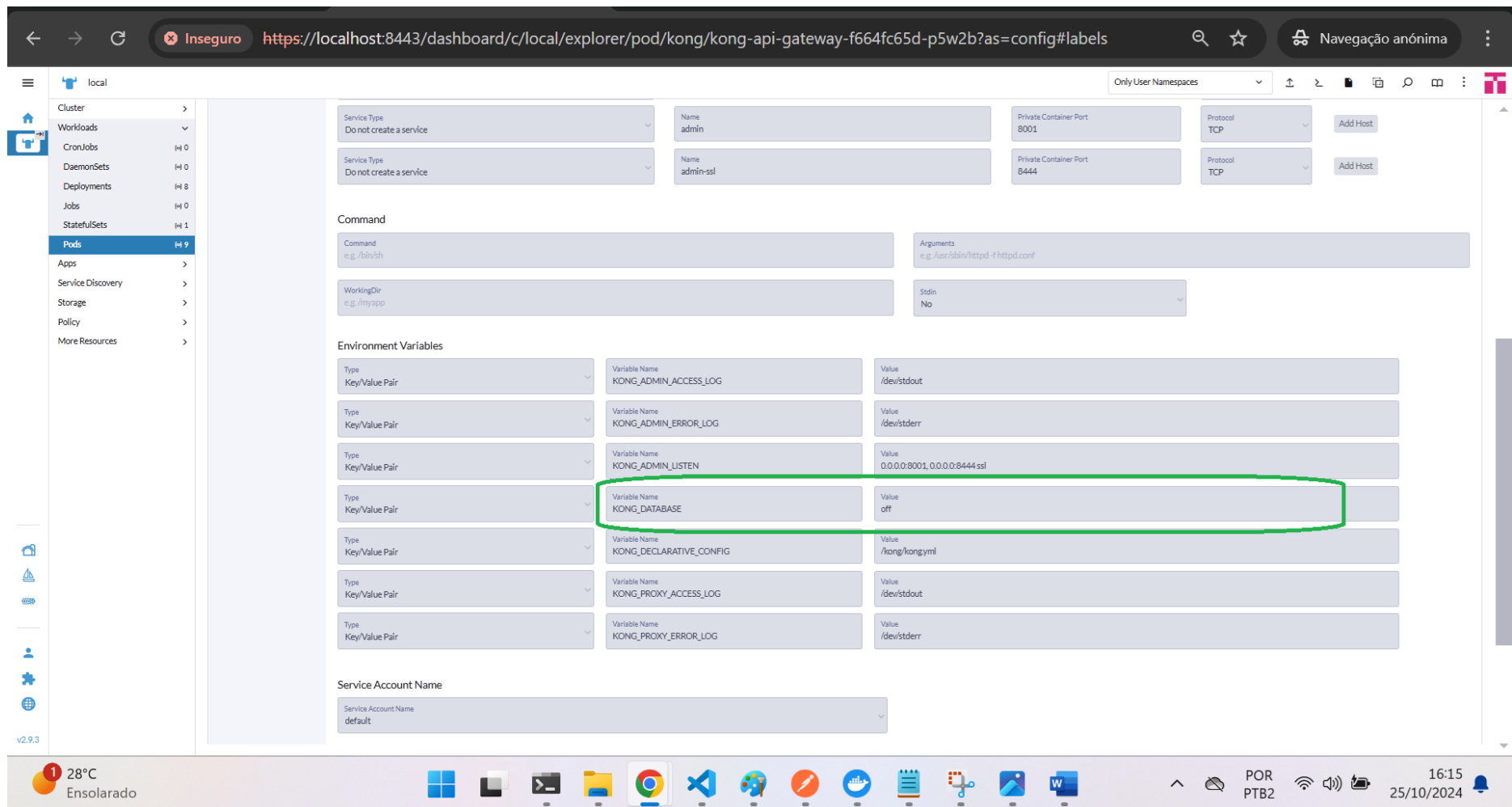


Figure 8: Kong Api Gateway variável de ambiente KONG_DATABASE off

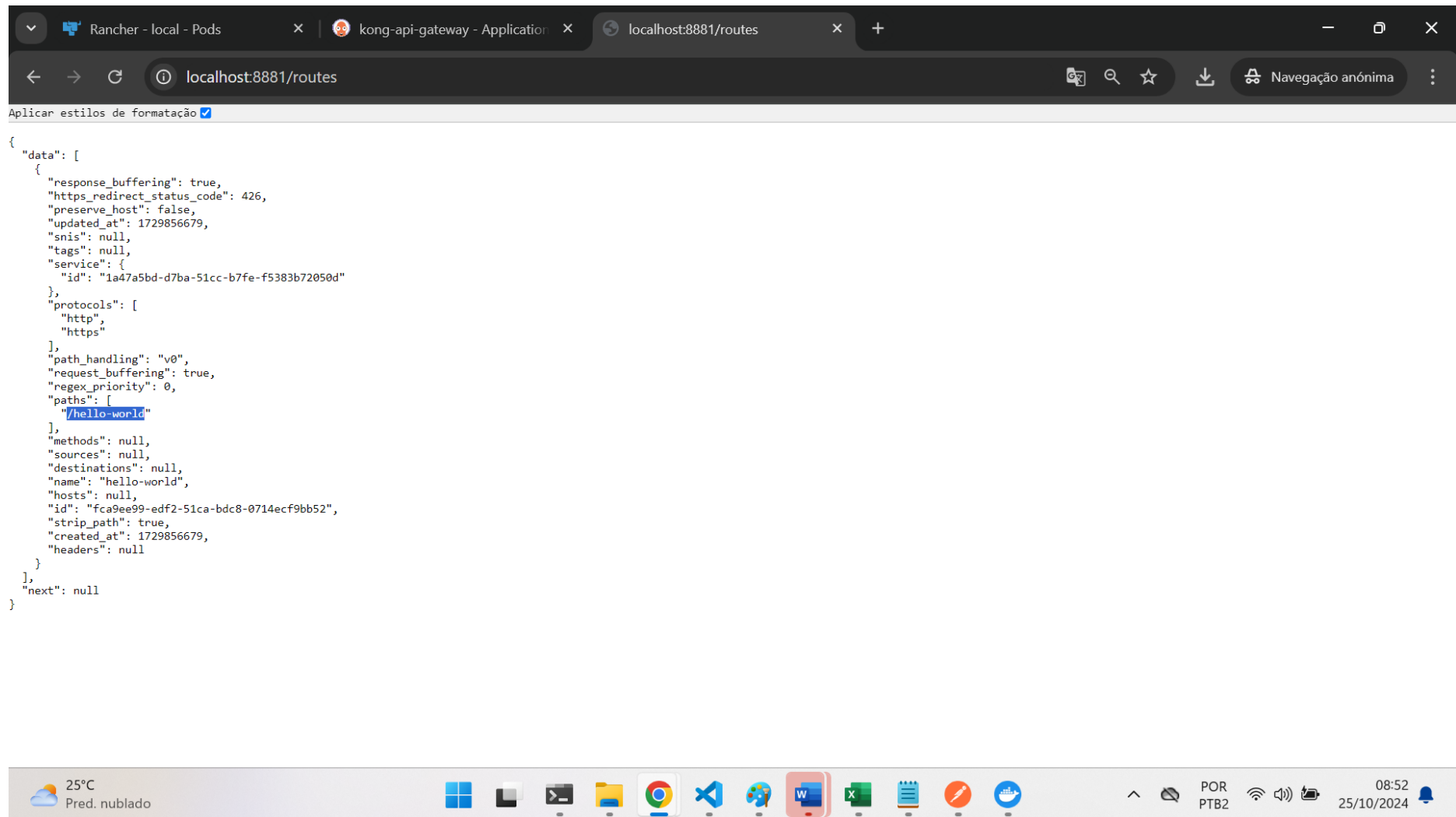
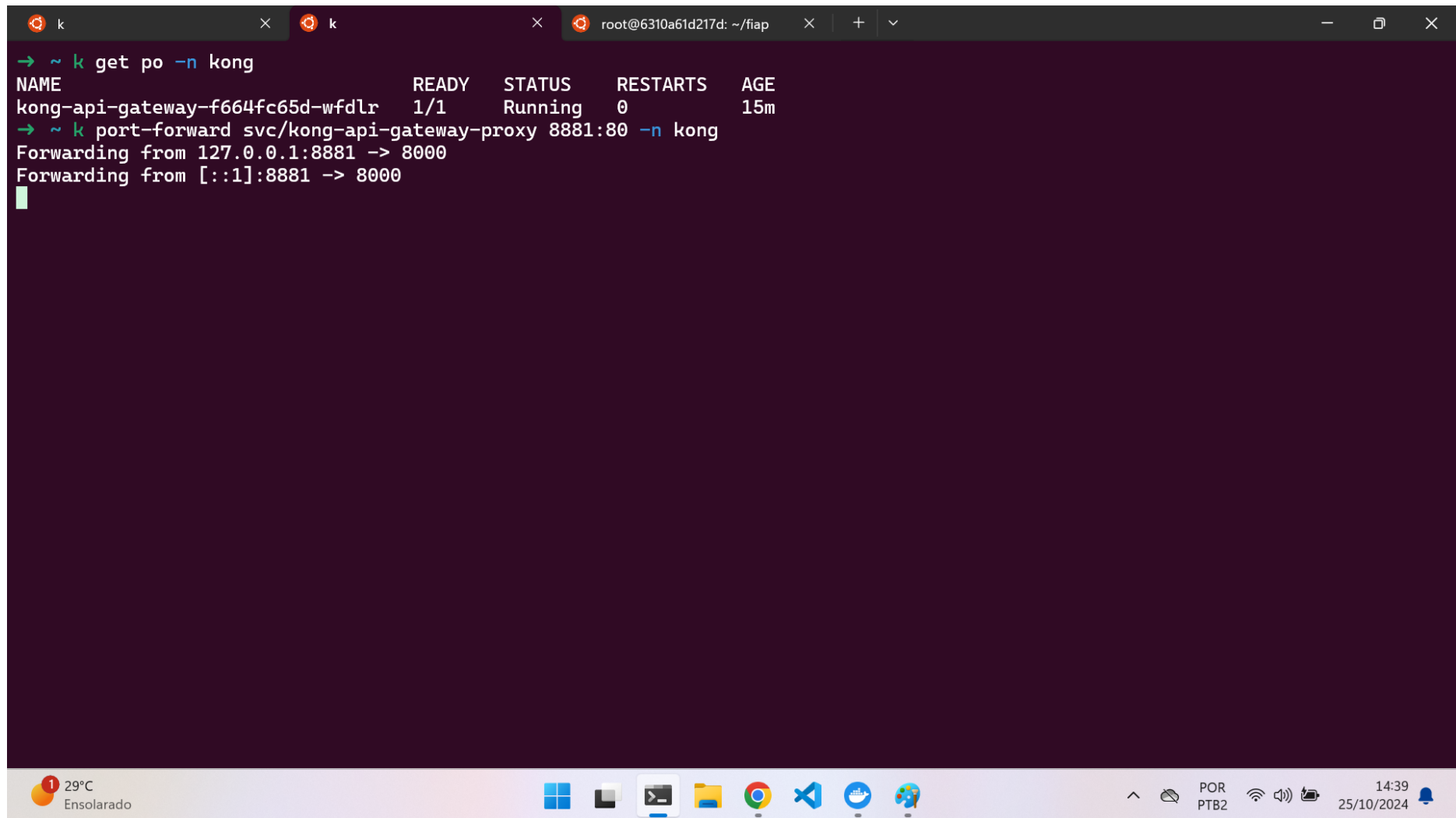


Figure 9: Kong Api Gateway rotas cadastradas



```
→ ~ k get po -n kong
NAME                                READY   STATUS    RESTARTS   AGE
kong-api-gateway-f664fc65d-wfdlr    1/1     Running   0           15m
→ ~ k port-forward svc/kong-api-gateway-proxy 8881:80 -n kong
Forwarding from 127.0.0.1:8881 -> 8000
Forwarding from [::1]:8881 -> 8000
```

The image shows a terminal window with three tabs. The active tab is titled 'root@6310a61d217d: ~/fiap'. The terminal output shows the execution of two Kubernetes commands. The first command, 'k get po -n kong', lists a pod named 'kong-api-gateway-f664fc65d-wfdlr' in the 'kong' namespace, which is in a 'Running' state. The second command, 'k port-forward svc/kong-api-gateway-proxy 8881:80 -n kong', initiates port forwarding from the host's ports 8881 and 80 to the pod's port 8000. The terminal window is part of a desktop environment with a taskbar at the bottom showing various application icons and system status information like temperature and time.

Figure 10: Kong Api Gateway redirecionando porta para acesso

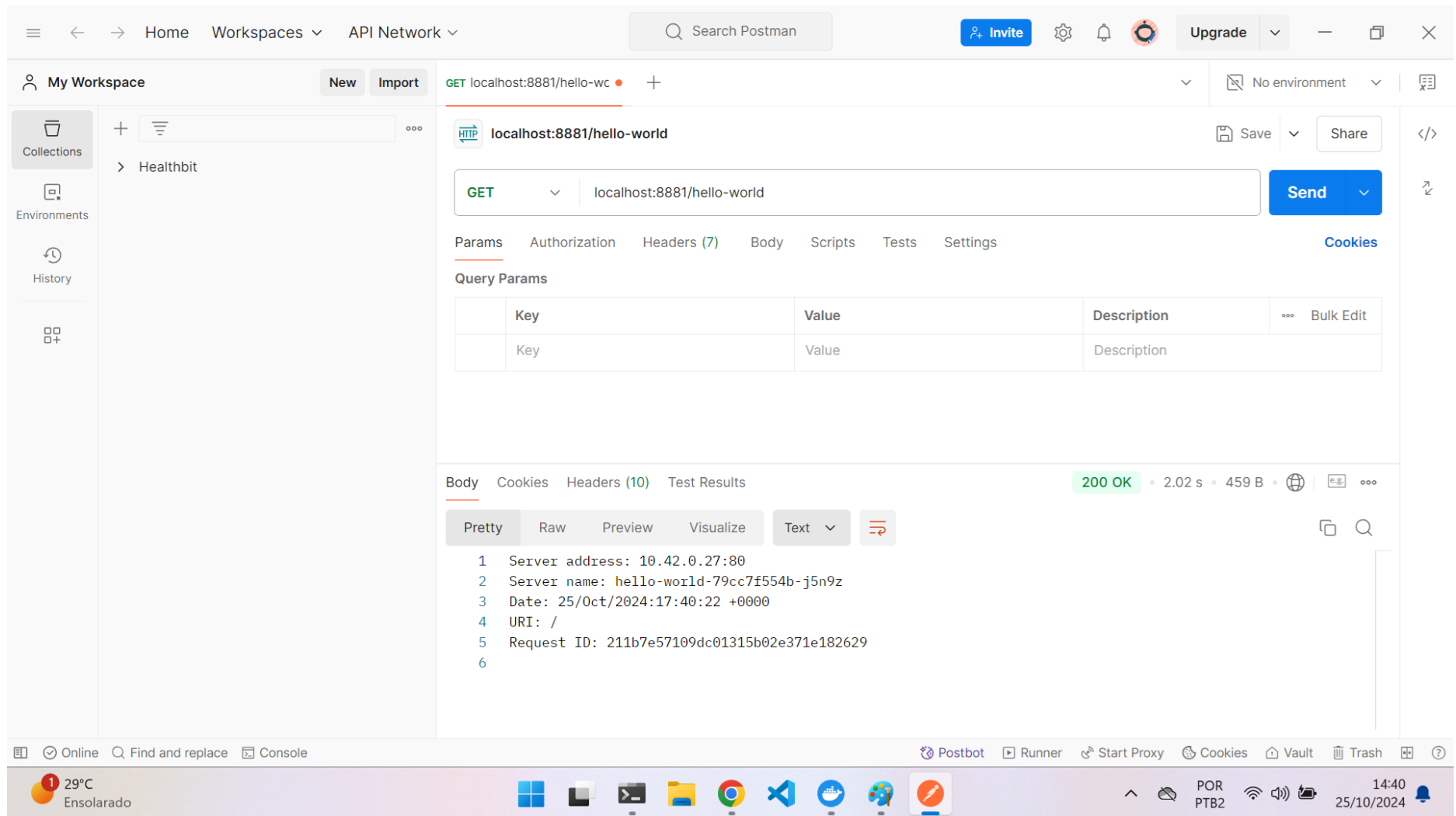


Figure 11: Postman requisição para rota cadastrada serviço Kong Api Gateway

Postman interface showing a successful GET request to `localhost:8881/hello-world`. The response status is **200 OK** with a response time of 2.02 s and a body size of 459 B.

The response headers are displayed in the Headers tab:

Key	Value	Description
Expires	Fri, 25 Oct 2024 17:40:21 GMT	
Cache-Control	no-cache	
X-Kong-Upstream-Latency	1	
X-Kong-Proxy-Latency	2020	
Via	kong/2.7.2	

The bottom of the image shows the Windows taskbar with the system clock displaying 14:41 on 25/10/2024.

Figure 12: Postman resultado da requisição para rota cadastrada serviço Kong Api Gateway