1. Gerar Imagem API

TechChallengeFiapFase4\Contact:

docker build --no-cache -t contactcreateapiv1 -f APIs/CreateAPI/Dockerfile .

docker build --no-cache -t contactreadapiv1 -f APIs/ReadAPI/Dockerfile .

docker build --no-cache -t contactupdateapiv1 -f APIs/UpdateAPI/Dockerfile .

docker build --no-cache -t contactdeleteapiv1 -f APIs/DeleteAPI/Dockerfile .

1. Gerar Imagem Worker

TechChallengeFiapFase4\Contact:

docker build --no-cache -t createworkerv1 -f Workers/CreateWorker/Dockerfile .

docker build --no-cache -t updateworkerv1 -f Workers/UpdateWorker/Dockerfile .

docker build --no-cache -t deleteworkerv1 -f Workers/DeleteWorker/Dockerfile .

1. Instalando o Helm: <https://helm.sh/pt/docs/intro/install/>
2. Instalando o Rabittmq: IntalacaoRabbitMQ.txt

<https://artifacthub.io/packages/helm/bitnami/rabbitmq>

<https://www.youtube.com/watch?v=LcgvK4mvmTk>

IntalacaoRabbitMQ (Comandos para serem executados no terminal):

helm repo add bitnami https://charts.bitnami.com/bitnami

helm repo update

helm upgrade --install rabbitmq --set auth.username=guest --set auth.password=guest bitnami/rabbitmq

Para acessar a interface:

* Execute no terminal: kubectl port-forward --namespace default svc/rabbitmq 15672:15672
* Acesse <http://127.0.0.1:15672/>
* Utilize o usuário e senha cadastrado acima (guest / guest)

1. Pod/Service SQL (acessar pasta Pipelines):

kubectl apply -f .\Deployments\deploy-sqlserver.yaml

kubectl apply -f .\Services\svc-sqlserver.yaml

1. Pod Api (acessar pasta Pipelines):

kubectl apply -f .\Deployments\deploy-createapi.yaml

kubectl apply -f .\Deployments\deploy-readapi.yaml

kubectl apply -f .\Deployments\deploy-updateapi.yaml

1. Pod Worker (acessar pasta Pipelines):

kubectl apply -f .\Deployments\deploy-createworker.yaml

kubectl apply -f .\Deployments\deploy-updateworker.yaml

1. Service Api (acessar pasta Pipelines):

kubectl apply -f .\Services\svc-createapi.yaml

kubectl apply -f .\Services\svc-readapi.yaml

kubectl apply -f .\Services\svc-updateapi.yaml

1. Service Worker (acessar pasta Pipelines):

kubectl apply -f .\Services\svc-createworker.yaml

kubectl apply -f .\Services\svc-updateworker.yaml

Acessar swagger API:

* Executar commando kubectl get services
* Verificar em qual porta o serviço desejado está exposto para fora do cluster
* Acessar o swagger com <http://localhost:PORTA/swagger/index.html> (ex: <http://localhost:30000/swagger/index.html>)

1. Instalando o Prometheus:

<https://artifacthub.io/packages/helm/prometheus-community/prometheus>

<https://www.youtube.com/watch?v=I16rbYe7ULg&t=1268s>

<https://github.com/cirolini/prometheus-curso-monitoring/blob/main/07_kubernetes/01_install_prometheus_on_k8s.md>

Comandos para serem executados no terminal:

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

helm repo update

helm install prometheus prometheus-community/prometheus

kubectl --namespace default port-forward $POD\_NAME 9090



1. Instalando o Grafana:

<https://artifacthub.io/packages/helm/grafana/grafana>

<https://www.youtube.com/watch?v=I16rbYe7ULg&t=1268s>

<https://github.com/grafana/helm-charts/blob/main/charts/grafana/README.md>

Comandos para serem executados no terminal:

helm repo add grafana https://grafana.github.io/helm-charts

helm repo update

helm install grafana grafana/Grafana

kubectl --namespace default port-forward $POD\_NAME 3000

Recuperar Senha (Ambiente Linux):

kubectl get secret --namespace default grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo



