



VisionHive

DEVOPS TOOLS & CLOUD COMPUTING

João Victor Michaeli - 555678
Larissa Muniz - 557197
Henrique Garcia - 558062

Links

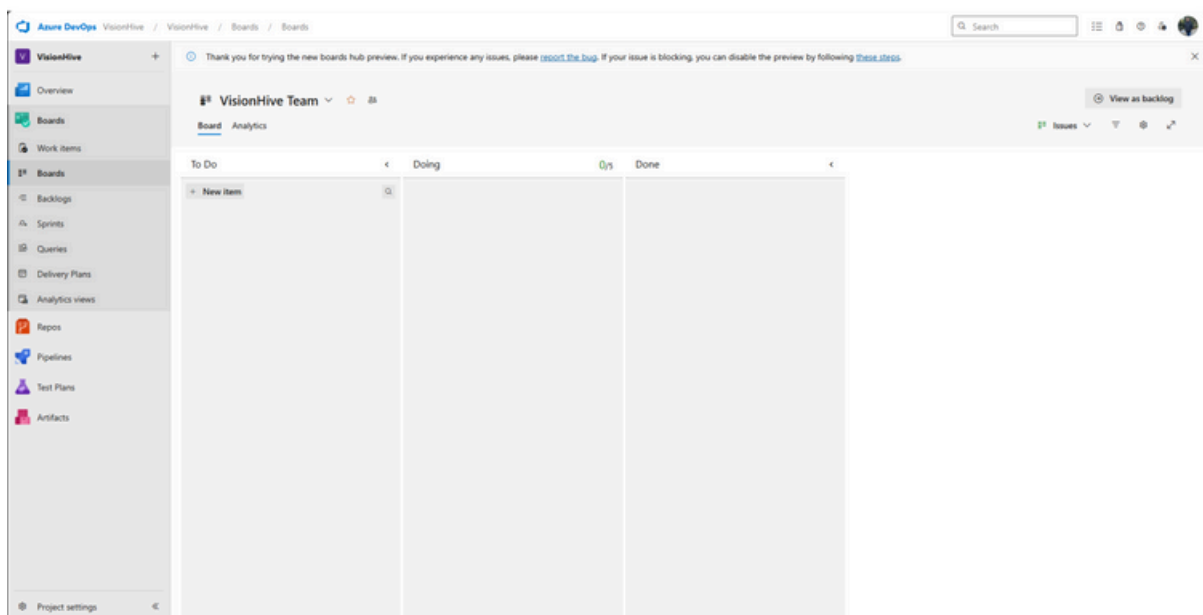
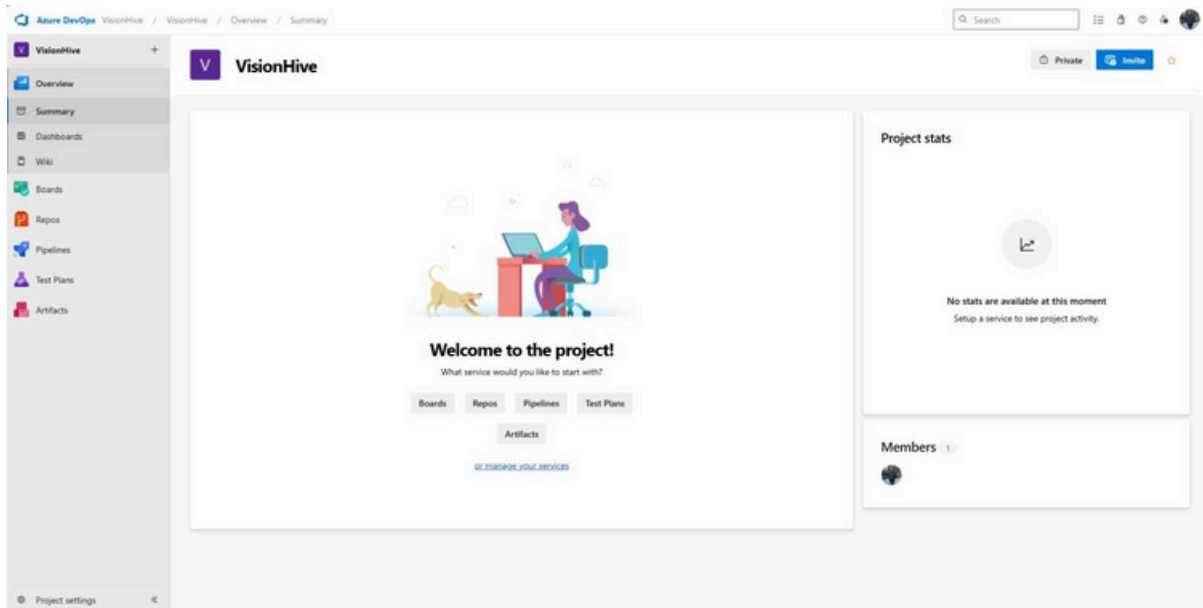
**Link para o vídeo da
criação da Vm**

<https://youtu.be/qW7178f7yGY>

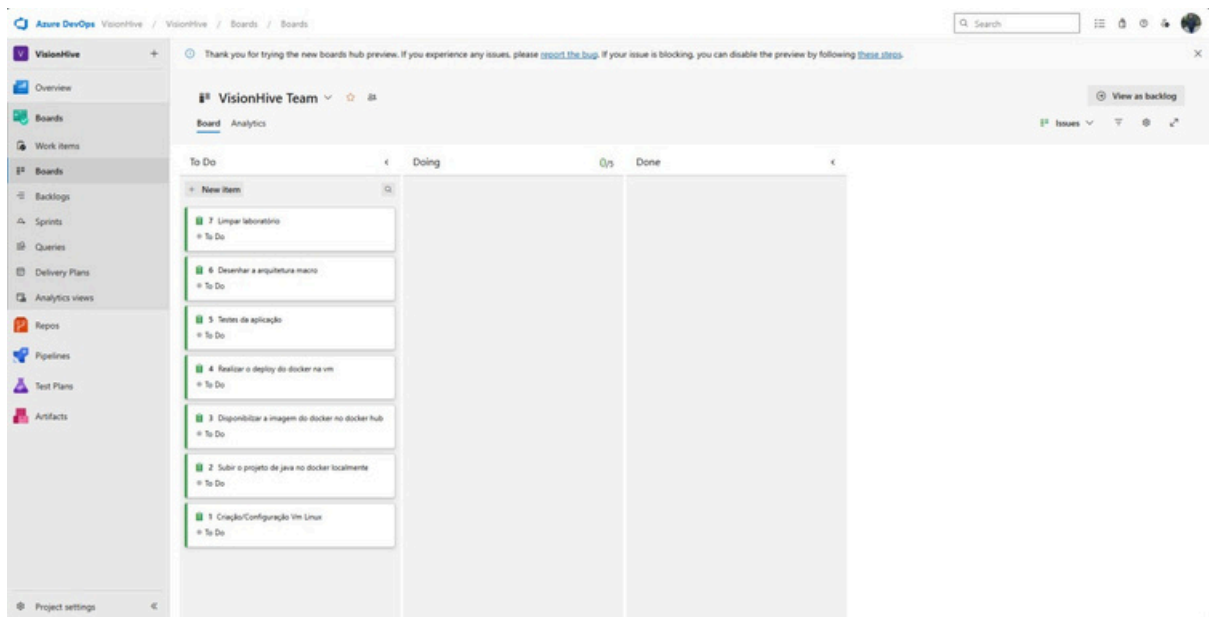
**Link do repositório
github**

<https://github.com/JoaoMichaeli/VisionHive-DevOps>

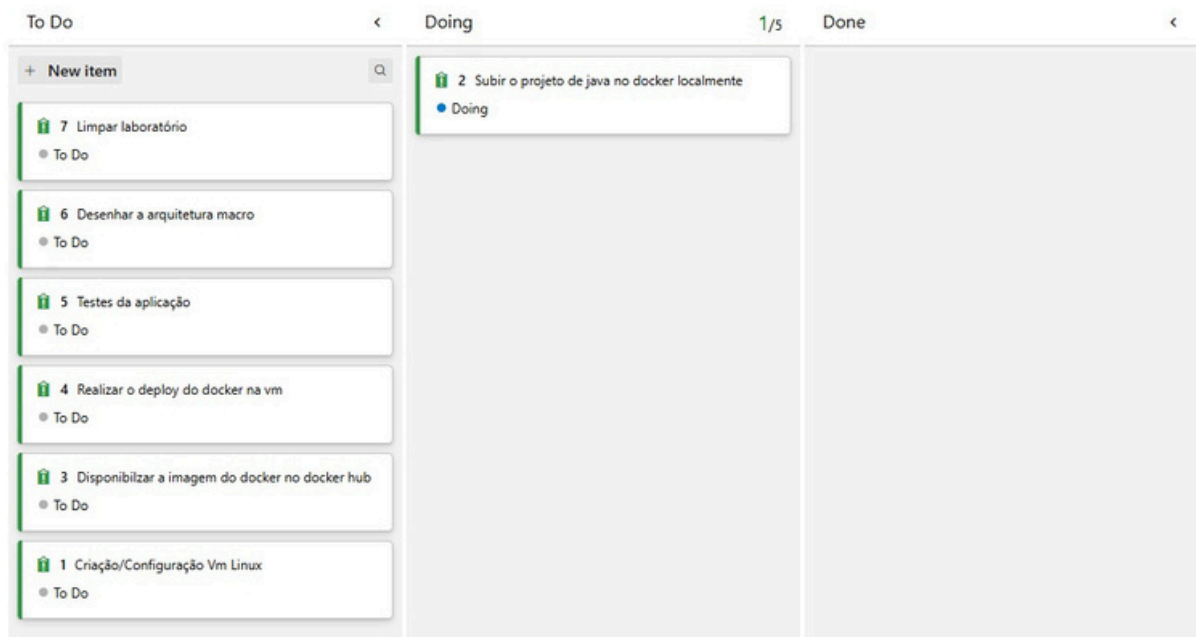
Criação do Azure board para melhor organização



Preenchimento da lista To Do



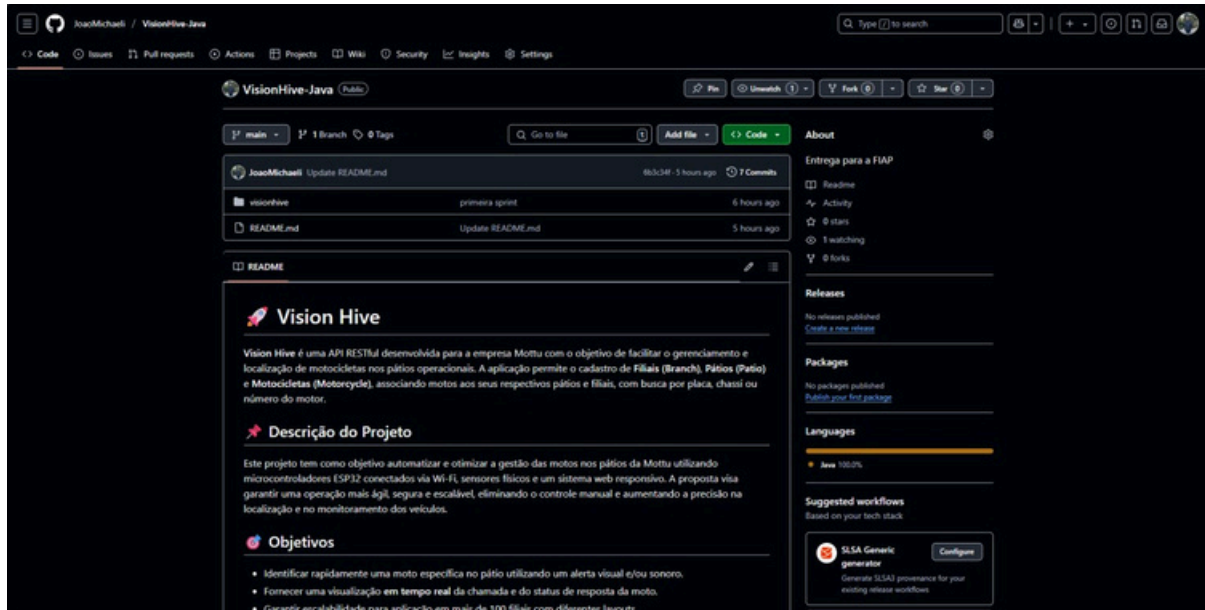
Primeiro vamos subir o java no docker local, para posteriormente subir no docker hub



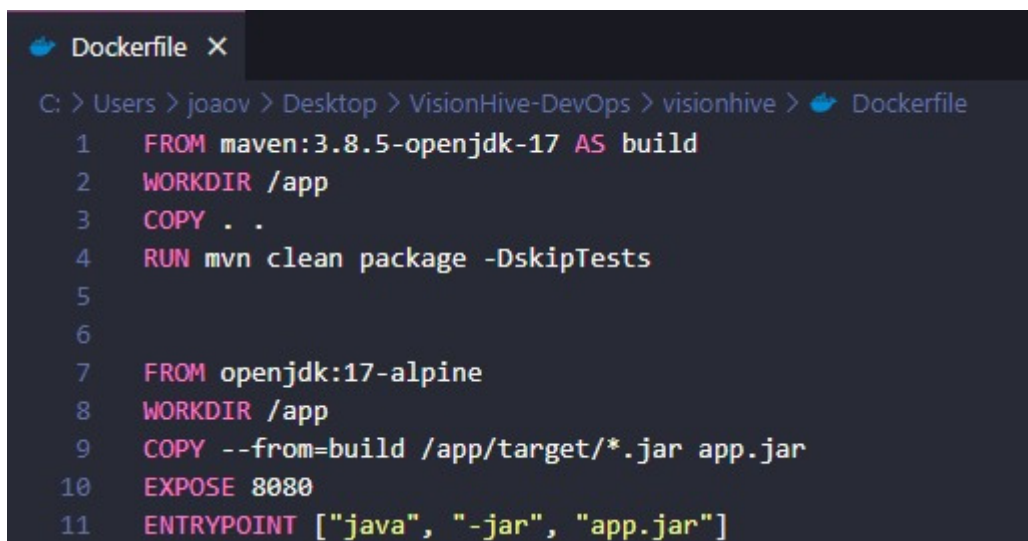
Criar o Dockerfile

O projeto java está nesse repositório do github:

<https://github.com/JoaoMichaeli/VisionHive-Java>



Dockerfile do projeto



Criar e testar a imagem Docker localmente

```
Terminal
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive> docker build -t joaomichaeli/visionhive-java:v1 .

[+] Building 17.9s (15/15) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 280B
=> [internal] load metadata for docker.io/library/openjdk:17-alpine
=> [internal] load metadata for docker.io/library/maven:3.8.5-openjdk-17
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [auth] library/maven:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [build 1/4] FROM docker.io/library/maven:3.8.5-openjdk-17@sha256:3a9c30b3af6278a8ae0007d3a3bf00fff80ec3ed
=> => resolve docker.io/library/maven:3.8.5-openjdk-17@sha256:3a9c30b3af6278a8ae0007d3a3bf00fff80ec3ed7ae4eb
=> [stage-1 1/3] FROM docker.io/library/openjdk:17-alpine@sha256:4b6abae565492dbe9e7a894137c966a748515423890
=> => resolve docker.io/library/openjdk:17-alpine@sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e9
=> [internal] load build context
=> => transferring context: 55.80kB
=> CACHED [stage-1 2/3] WORKDIR /app
=> CACHED [build 2/4] WORKDIR /app
=> [build 3/4] COPY .
=> [build 4/4] RUN mvn clean package -DskipTests
=> [stage-1 3/3] COPY --from=build /app/target/*.jar app.jar

=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:2ad7e5b3c8569518b474b305fe4e0a3d12d16f1340bde32130dcd46436c40d52
=> => exporting config sha256:1ec13dd6250f14552f903adee4dcce834d6eadbb4bd7c12791d456e820d53573
=> => exporting attestation manifest sha256:7f48ffd913268aacc79806528b0e69965cf88f3aef78d9be89020c7aff1c1ec5
=> => exporting manifest list sha256:9a82dd24a38e6529b03d0de549390be0e6476fe229cda4c4e09db14e92b7434c
=> => naming to docker.io/joaomichaeli/visionhive-java:v1
=> => unpacking to docker.io/joaomichaeli/visionhive-java:v1

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/y9kb6nmvfznbdb1hqw59eta1
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive>
```

Imagem criada

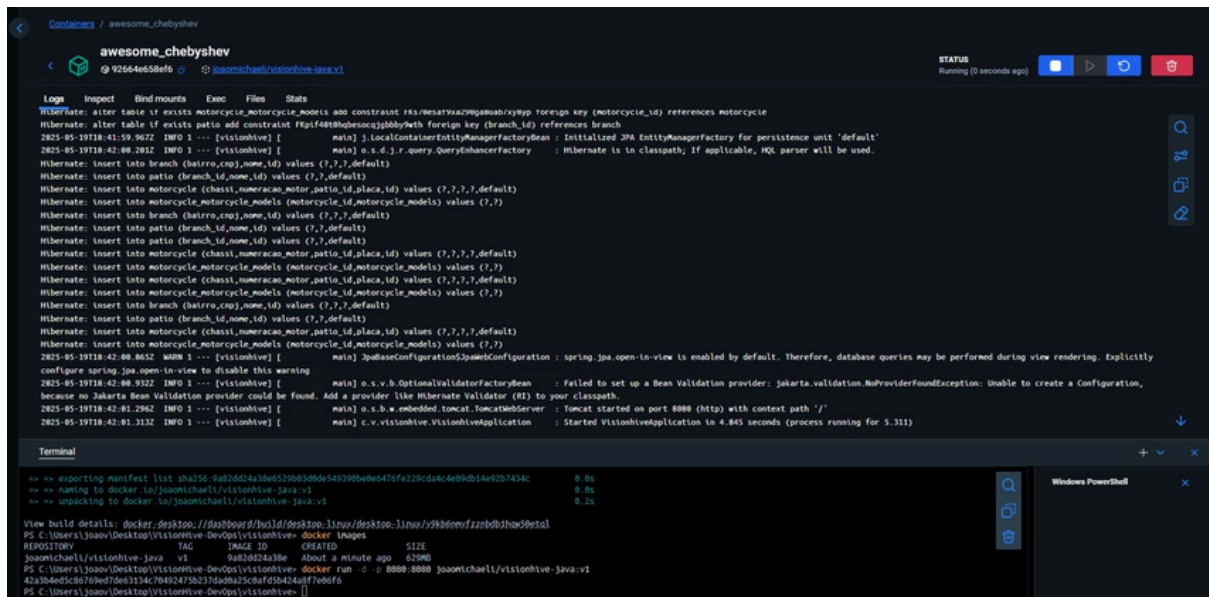
```
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive> docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
joaomichaeli/visionhive-java  v1         9a82dd24a38e  About a minute ago  629MB
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive>
```

Images [Give feedback](#) [Learn more](#)

View and manage your local and Docker Hub images. [Learn more](#)

	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	joaomichaeli/visionhive-java	v1	9a82dd24a38e	3 seconds ago	629.24 MB	▶ ⋮ 🗑️

Como podemos ver, o projeto já está funcionando localmente



```
awesome_chebyshev
@ 92664ed58ef6  java:1

Logs  Inspect  Bind mounts  Exec  Files  Stats

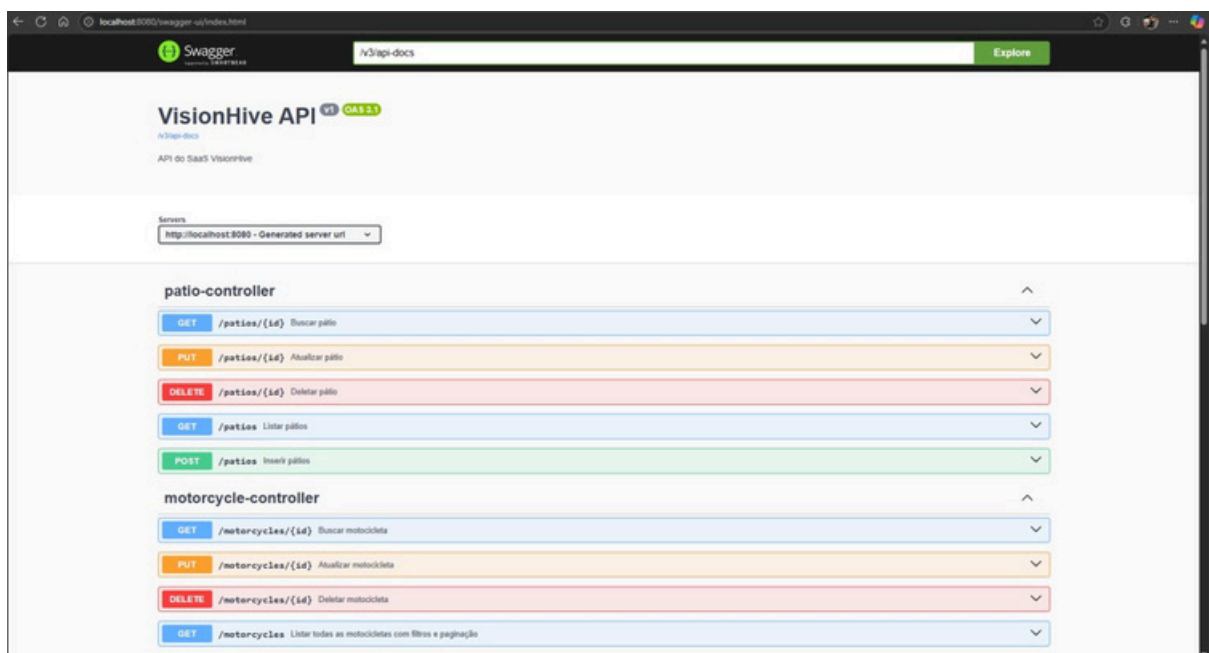
WARN: alter table if exists motorcycle_models add constraint FK5f488b0e0c93bb07b6 foreign key (branch_id) references branch
2825-05-19T18:42:59.962Z INFO 1 --- [visionhive] [ main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2825-05-19T18:42:59.281Z INFO 1 --- [visionhive] [ main] o.s.d.j.r.query.QueryHintsFactory : Hibernate is in classpath; if applicable, HQ parser will be used.
Hibernate: insert into branch (bairro,cnpj,nome,id) values (7,7,7,default)
Hibernate: insert into patio (branch_id,nome,id) values (7,7,default)
Hibernate: insert into motorcycle (chassi,numeracao,motor_patio_id,placa,id) values (7,7,7,7,default)
Hibernate: insert into motorcycle_models (motorcycle_id,motorcycle_models) values (7,7)
Hibernate: insert into branch (bairro,cnpj,nome,id) values (7,7,7,default)
Hibernate: insert into patio (branch_id,nome,id) values (7,7,default)
Hibernate: insert into motorcycle (chassi,numeracao,motor_patio_id,placa,id) values (7,7,7,7,default)
Hibernate: insert into motorcycle_models (motorcycle_id,motorcycle_models) values (7,7)
Hibernate: insert into motorcycle_models (motorcycle_id,motorcycle_models) values (7,7)
Hibernate: insert into branch (bairro,cnpj,nome,id) values (7,7,7,default)
Hibernate: insert into patio (branch_id,nome,id) values (7,7,default)
Hibernate: insert into motorcycle (chassi,numeracao,motor_patio_id,placa,id) values (7,7,7,7,default)
Hibernate: insert into motorcycle_models (motorcycle_id,motorcycle_models) values (7,7)
2825-05-19T18:42:59.865Z INFO 1 --- [visionhive] [ main] 3pbaseConfiguration5pbaseConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2825-05-19T18:42:59.932Z INFO 1 --- [visionhive] [ main] o.s.v.b.OptionalValidatorFactoryBean : Failed to set up a Bean Validation provider: jakarta.validation.spi.ProviderException: Unable to create a Configuration, because no Jakarta Bean Validation provider could be found. Add a provider like Hibernate Validator (RI) to your classpath.
2825-05-19T18:42:59.294Z INFO 1 --- [visionhive] [ main] o.s.w.embedded.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
2825-05-19T18:42:59.312Z INFO 1 --- [visionhive] [ main] c.v.visionhive.VisionhiveApplication : Started VisionhiveApplication in 4.845 seconds (process running for 5.311)

Terminal

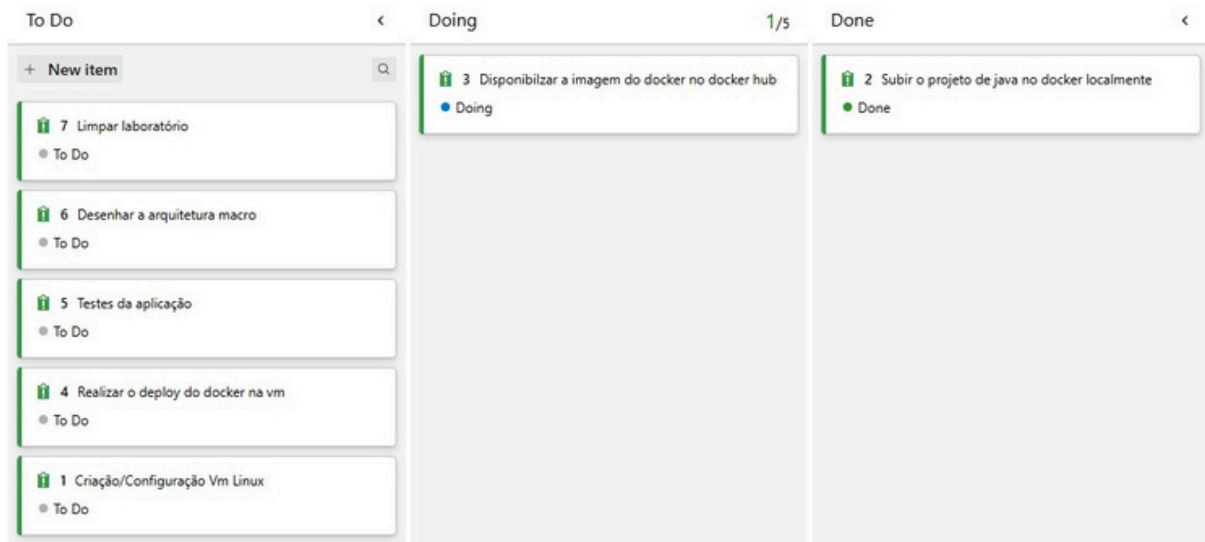
==> exporting Manifest list sha256:9a826d24a38e579b03d0d5493909eb647ef225cd4c4e9db14e2b7434c 0.0s
==> naming to docker.io/joamichaelli/visionhive-java-v1 0.0s
==> unpacking to docker.io/joamichaelli/visionhive-java-v1 0.2s

View build details: docker desktop://ashBoard/build/desktop.linux/desktop.linux/v9b6devfzr0b3hwe0beta1
PS C:\Users\joao\Desktop\Visionhive> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
joamichaelli/visionhive-java v1 9a826d24a38e About a minute ago 620MB
PS C:\Users\joao\Desktop\Visionhive> docker run -d -p 8080:8080 joamichaelli/visionhive-java:v1
42a3b4e5c8679ed7de03134c76492475237da0a25c4fd5b424a8f7e0f6
PS C:\Users\joao\Desktop\Visionhive> docker ps
PS C:\Users\joao\Desktop\Visionhive> docker logs -f 42a3b4e5c8679ed7de03134c76492475237da0a25c4fd5b424a8f7e0f6
```

Como essa é a primeira entrega da solução em java, ainda não temos integração web gráfica, então para testarmos o deploy será utilizado via swagger



Após o sucesso do teste, vou subir a imagem no docker hub para o envio posterior a vm.



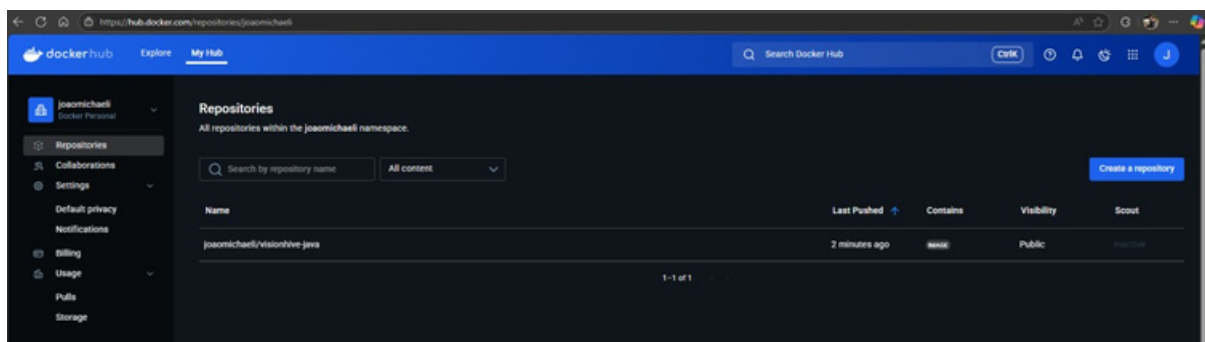
Agora vamos realizar a disponibilidade da imagem local para o hub

```
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive> docker login
Authenticating with existing credentials... [Username: joaomichaeli]

Info → To login with a different account, run 'docker logout' followed by 'docker login'

Login Succeeded
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive> docker push joaomichaeli/visionhive-java:v1
The push refers to repository [docker.io/joaomichaeli/visionhive-java]
c3b66361bb95: Pushed
7992fe636f76: Pushed
d8d715783b80: Pushed
53c9466125e4: Pushed
5843afab3874: Pushed
d902d246dc1f: Pushed
v1: digest: sha256:9a82dd24a38e6529b03d0de549390be0e6476fe229cda4c4e09db14e92b7434c size: 856
PS C:\Users\joaov\Desktop\VisionHive-DevOps\visionhive>
```

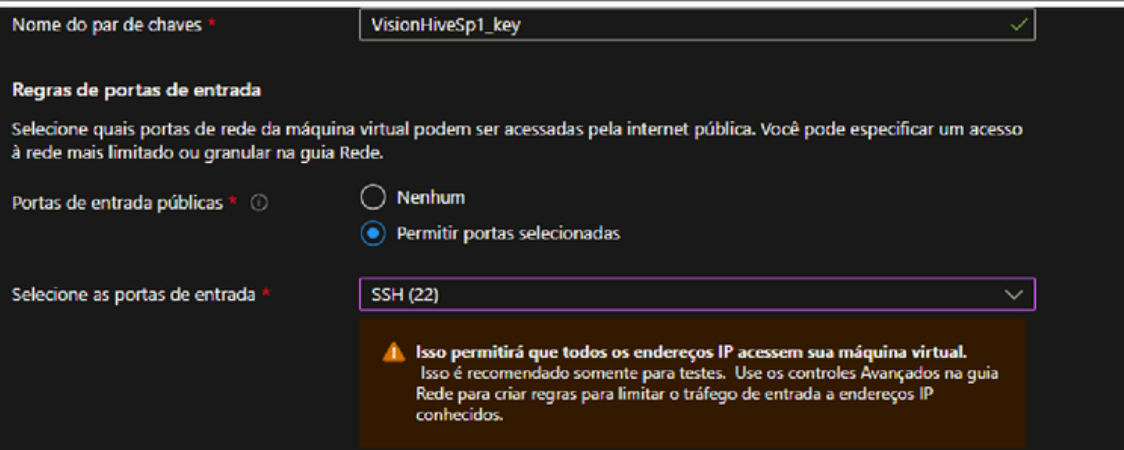
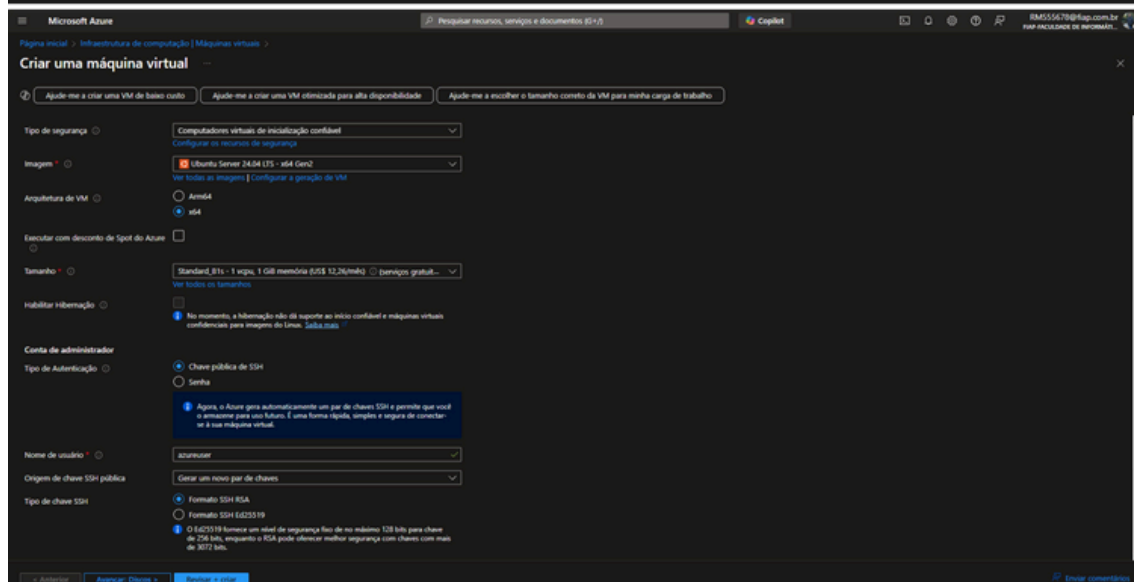
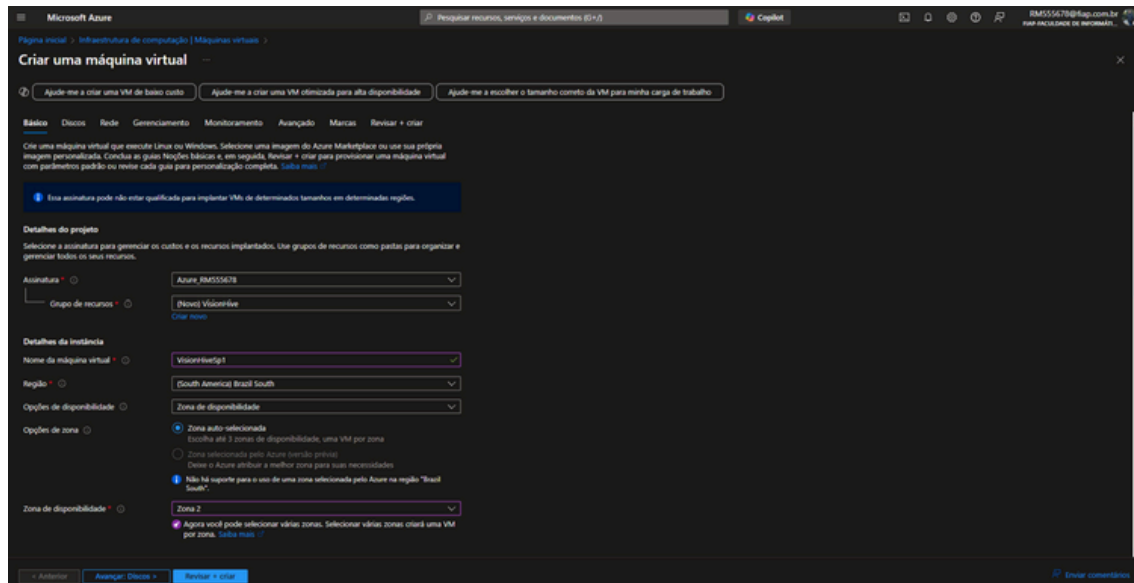
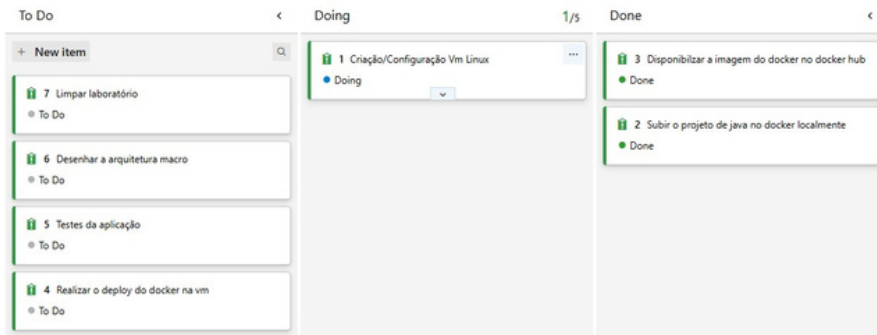
Imagem postada no hub



Com a imagem postada no docker hub, agora vamos criar e configurar a máquina virtual

<https://youtu.be/qW7178f7yGY>

Criar a maquina virtual



Precificação

Preço

1 X Standard B1s

por Microsoft

[Termos de uso](#) | [Política de privacidade](#)

Créditos de assinatura se aplicam ⓘ

0.0168USD/hora

Preços de outros tamanhos de VM

Implementação concluída

Microsoft Azure

Pesquisar recursos, serviços e documentos (G+/)

Página inicial >

CreateVm-canonical.ubuntu-24_04-lts-server-20250519161205 | Visão Geral ⓘ

Implantação

Pesquisar

Excluir Cancelar Reimplantar Baixar Atualizar

Visão Geral Entradas Saídas Modelo

A implantação foi concluída

Nome da implantação: CreateVm-canonical.ubuntu-24_04-lts-serve... Hora de início: 19/05/2025, 16:16:33

Assinatura: Azure_RM555678 ID de Correlação: d32e16b7-d000-4c67-8f50-5f7ec92b8405 ⓘ

Grupo de recursos: VisionHive

Detalhes de implantação

Próximas etapas

Configurar desligamento automático Recomendado

Monitorar dependências de rede, desempenho e integridade da VM Recomendado

Executar um script dentro da máquina virtual Recomendado

[Ir para o recurso](#) [Criar outra VM](#)

Enviar comentários

Conte-nos sobre sua experiência com a implantação

Criação das portas

Microsoft Azure

Pesquisar recursos, serviços e documentos (G+/)

Copilot

Página inicial > CreateVm-canonical.ubuntu-24_04-lts-server-20250519161205 | Visão Geral > VisionHiveSp1

Visão geral Log de atividade IAM (Controlador de acesso) Marcas Diagnóstico e resolução de problemas Visualizador de recursos Converter Rede

Configurações de rede

Balanciamento de carga Grupos de segurança de aplicativos Gerente de rede Configurações Disponibilidade + escala Segurança Backup + recuperação de desastres Operações Monitoramento Automação Ajuda

Adaptador de rede / Configuração de IP visionhive-sp1726_u2 (primário) / ipconfig1 (primário)

Fundamentos

Adaptador de rede : visionhive-sp1726_u2

Rede virtual / sub-rede : visionhive-sp1-vnet / default

Endereço IP público : 20.195.168.126

Endereço IP privado : 10.0.0.4

Regras de segurança administrat... : 0 (Configure)

Balancedadores de carga : 0 (Configure)

Grupos de segurança do aplicativo : 0 (Configure)

Grupo de segurança de rede : visionhive-sp1-nsg

Rede acelerada : Desabilitado

Regras de segurança eficazes : 0

Regras Recolher tudo

Grupo de segurança de rede visionhive-sp1-nsg (anexado à networkinterface visionhive-sp1726_u2)

Impacta 0 sub-redes, 1 interfaces de rede

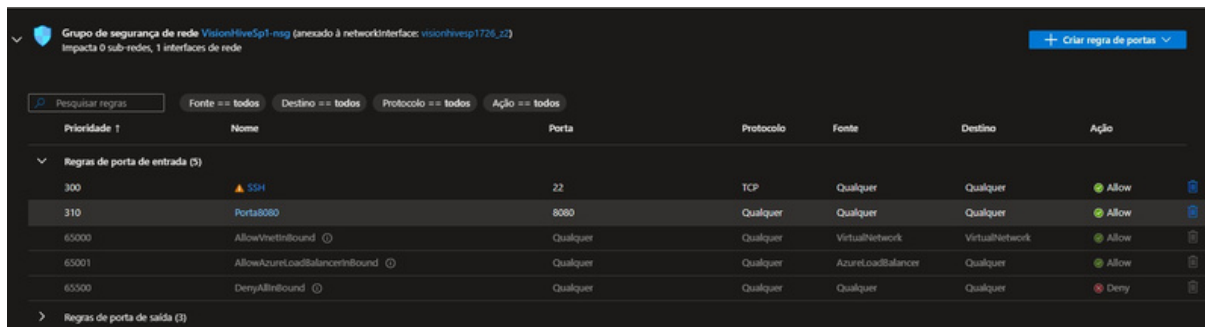
[+ Criar regra de portas](#)

Pesquisar regras

Fonte == todos Destino == todos Protocolo == todos Ação == todos

Prioridade ↑	Nome	Porta	Protocolo	Fonte	Destino	Ação
Regras de porta de entrada (4)						
300	SSH	22	TCP	Qualquer	Qualquer	Allow
40000	AllowInbound	Qualquer	Qualquer	VirtualNetwork	VirtualNetwork	Allow
40001	AllowAzureLoadBalancerInbound	Qualquer	Qualquer	AzureLoadBalancer	Qualquer	Allow
40000	DenyInbound	Qualquer	Qualquer	Qualquer	Qualquer	Deny
Regras de porta de saída (1)						

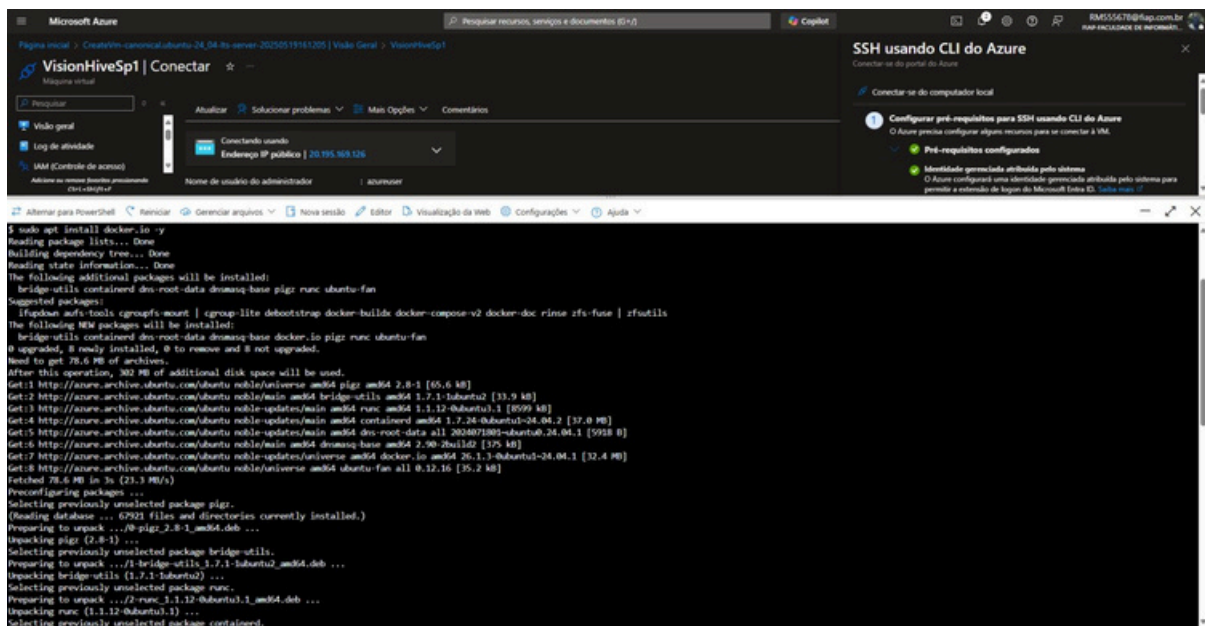
Como o java vai ser acessado pelo o swagger, vamos deixar somente a porta 8080 aberta



Prioridade	Nome	Porta	Protocolo	Fonte	Destino	Ação
300	SSH	22	TCP	Qualquer	Qualquer	Allow
310	Porta8080	8080	Qualquer	Qualquer	Qualquer	Allow
65000	AllowVnetInbound	Qualquer	Qualquer	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInbound	Qualquer	Qualquer	AzureLoadBalancer	Qualquer	Allow
65500	DenyAllInbound	Qualquer	Qualquer	Qualquer	Qualquer	Deny

Iremos utilizar o ssh do azure mesmo

Instalando o docker

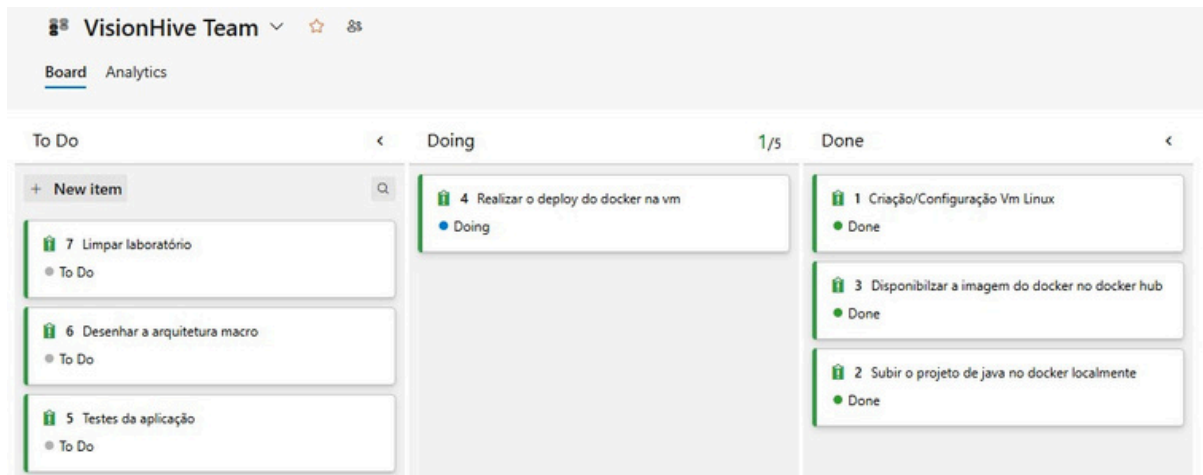


```
$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dracut-base pigz runc ubuntu-fan
Suggested packages:
  firestorm nfs-tools groupfs-mount | group-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dracut-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 8 not upgraded.
Need to get 78.6 MB of archives.
After this operation, 302 MB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.1.12-0ubuntu1.1 [8599 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.24-0ubuntu1~24.04.2 [37.0 MB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071800-ubuntu0.24.04.1 [5918 B]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 dracut-base amd64 2.90-2build2 [375 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 26.1.3-0ubuntu1~24.04.1 [32.4 MB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12-16 [35.2 kB]
Fetched 78.6 MB in 3s (23.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 67921 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.12-0ubuntu1.1_amd64.deb ...
Unpacking runc (1.1.12-0ubuntu1.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../4-containerd_1.7.24-0ubuntu1~24.04.2_amd64.deb ...
Unpacking containerd (1.7.24-0ubuntu1~24.04.2) ...
Selecting previously unselected package dns-root-data.
Preparing to unpack .../5-dns-root-data_2024071800-ubuntu0.24.04.1_all.deb ...
Unpacking dns-root-data (2024071800-ubuntu0.24.04.1) ...
Selecting previously unselected package dracut-base.
Preparing to unpack .../6-dracut-base_2.90-2build2_amd64.deb ...
Unpacking dracut-base (2.90-2build2) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../8-ubuntu-fan_0.12-16_all.deb ...
Unpacking ubuntu-fan (0.12-16) ...
```

Docker instalado na vm com a última versão

```
$ sudo systemctl start docker
$ sudo systemctl enable docker
$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

Agora vamos fazer o deploy para a vm

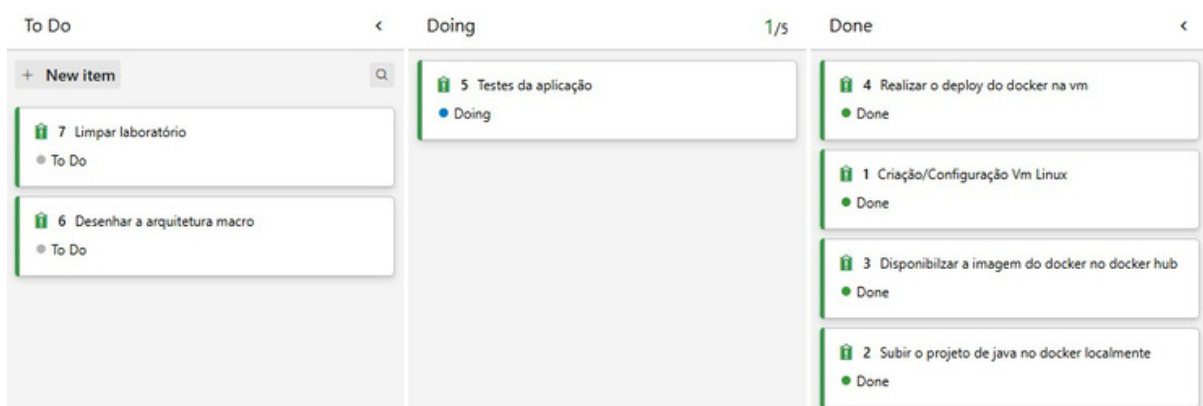


Fazendo o pull da imagem do docker

```
$ sudo docker pull joaomichaeli/visionhive-java:v1
v1: Pulling from joaomichaeli/visionhive-java
5843afab3874: Pull complete
53c9466125e4: Pull complete
d8d715783b80: Pull complete
d902d246dc1f: Pull complete
7992fe636f76: Pull complete
Digest: sha256:9a82dd24a38e6529b03d0de549390be0e6476fe229cda4c4e09db14e92b7434c
Status: Downloaded newer image for joaomichaeli/visionhive-java:v1
docker.io/joaomichaeli/visionhive-java:v1

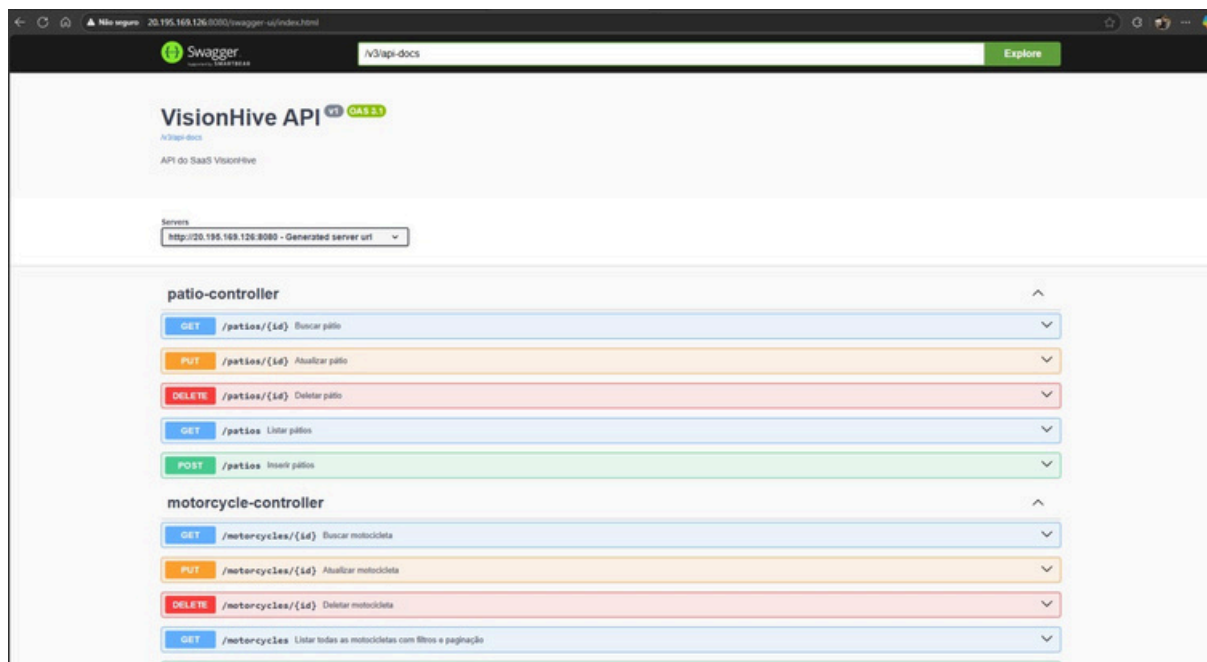
$ sudo docker run -d -p 8080:8080 joaomichaeli/visionhive-java:v1
ca62c090c390c341df6494a500c5e50d407f13af55772369cce7d951f2eb8d9a
```

Como havia demonstrado no começo, o java ainda não possui interface web, o acesso é somente do swagger

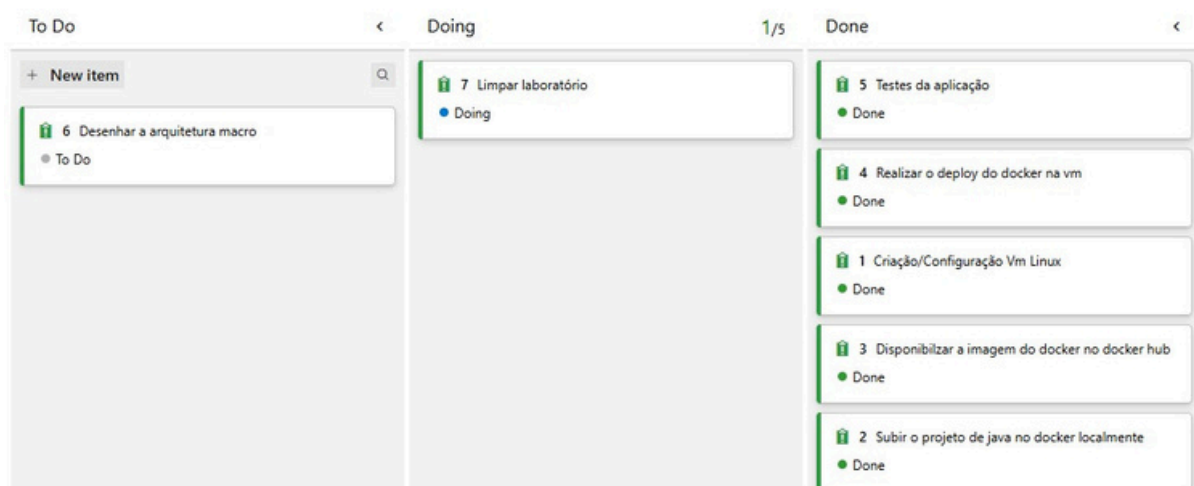


O swagger fica disponível no seguinte acesso:

http://<IP_VM>/swagger-ui/index.html

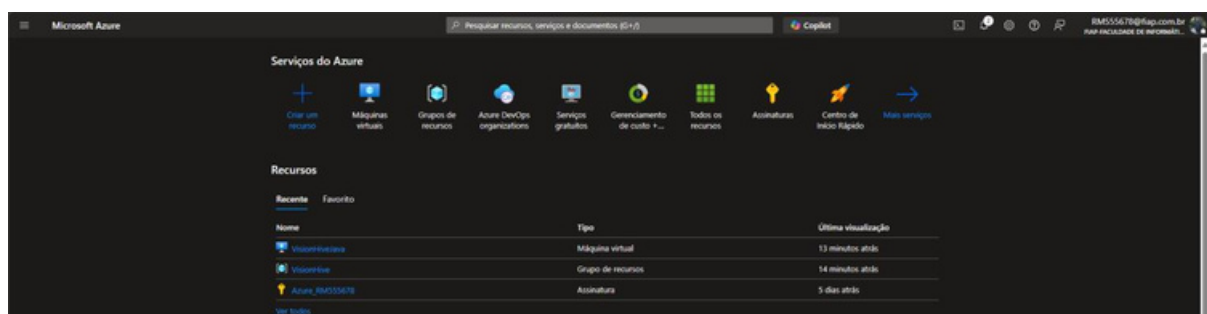


Agora é a limpeza do laboratório



Primeiro vamos limpar o Azure

Serviços criados



Excluir VisionHiveJava

Esta ação excluirá permanentemente esta máquina virtual.

Recurso a ser excluído	Tipo de recurso
<div><div></div><div>VisionHiveJava</div></div>	Máquina virtual

☒

Aplicar exclusão forçada ⓘ

☐

Esta máquina virtual pode ser excluída à força com segurança porque todos os seus recursos associados estão sendo excluídos.

Você também pode optar por excluir os recursos associados ao mesmo tempo. Os recursos que não são excluídos serão órfãos. Os recursos associados que estão sendo usados por outros recursos não são mostrados aqui.

Tipo de recurso associado	Quantida...	Excluir com VM
> <div><div></div><div>Disco de SO</div></div>	1	<input checked="" type="checkbox"/>
> <div><div></div><div>Interfaces de rede</div></div>	1	<input checked="" type="checkbox"/>
> <div><div></div><div>Endereços do IP públicos</div></div>	1	<input checked="" type="checkbox"/>

☒ Eu li e entendi que esta máquina virtual, bem como todos os recursos associados selecionados listados acima, serão excluídos.

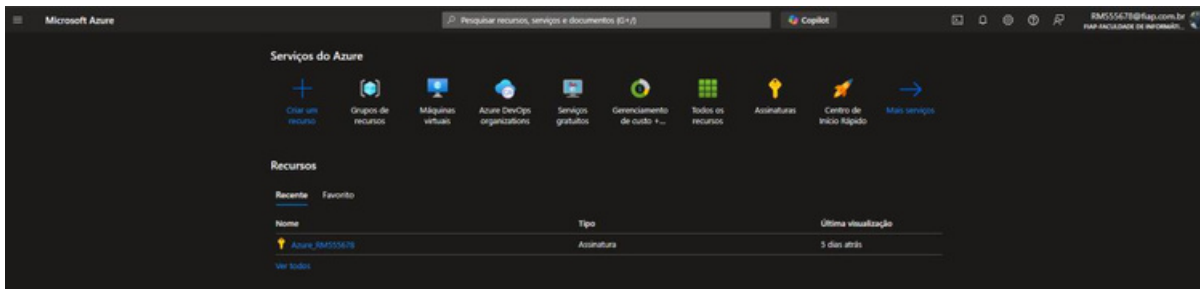
Excluir

Cancelar

Comentários

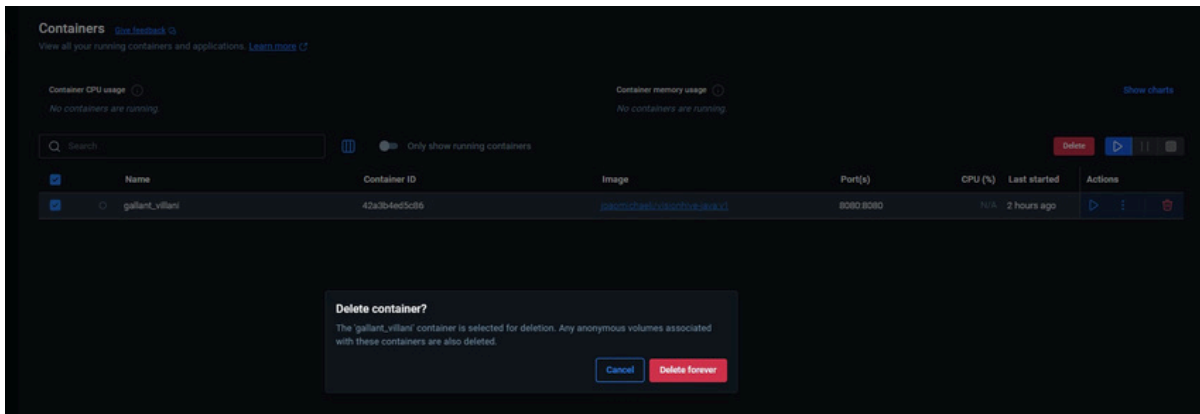
[illegible]

Serviços limpos

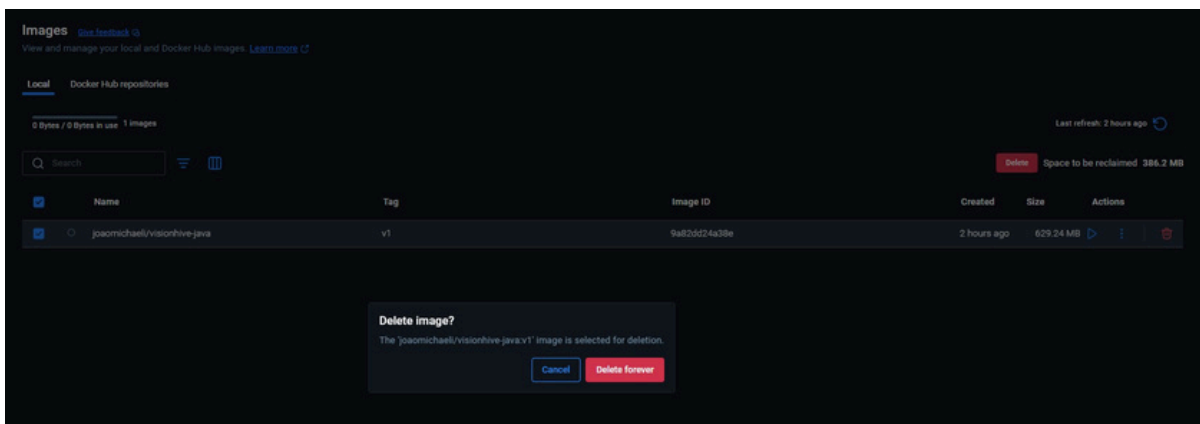


Agora vamos limpar o docker

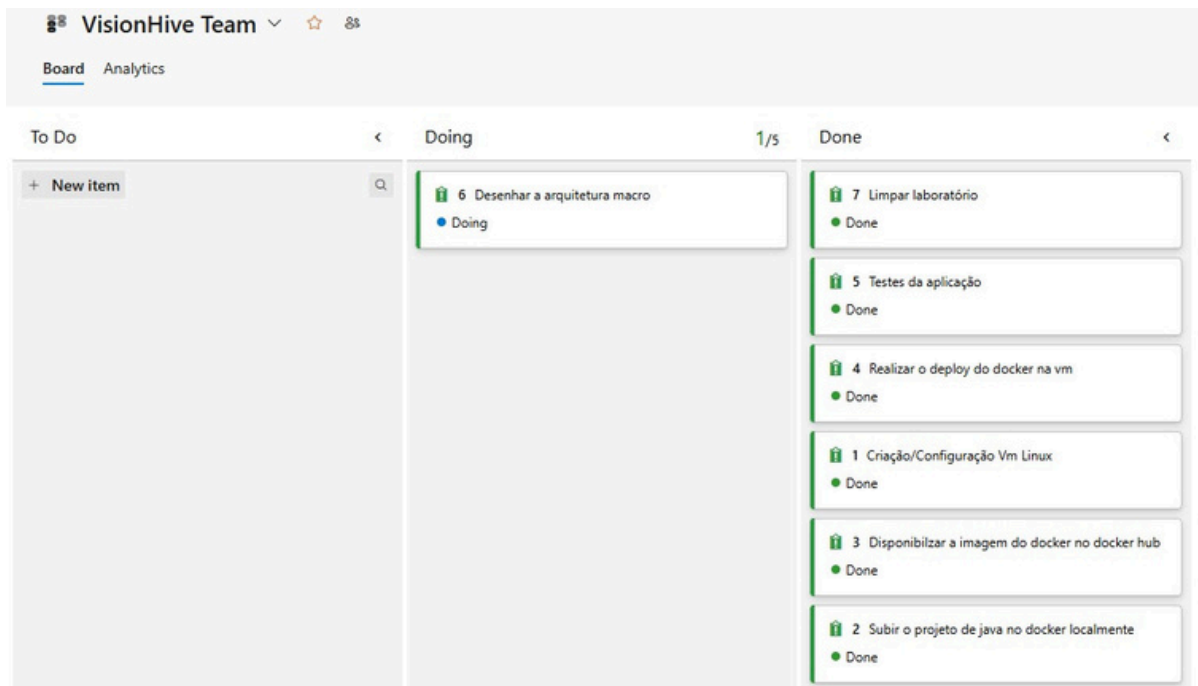
Excluindo o container



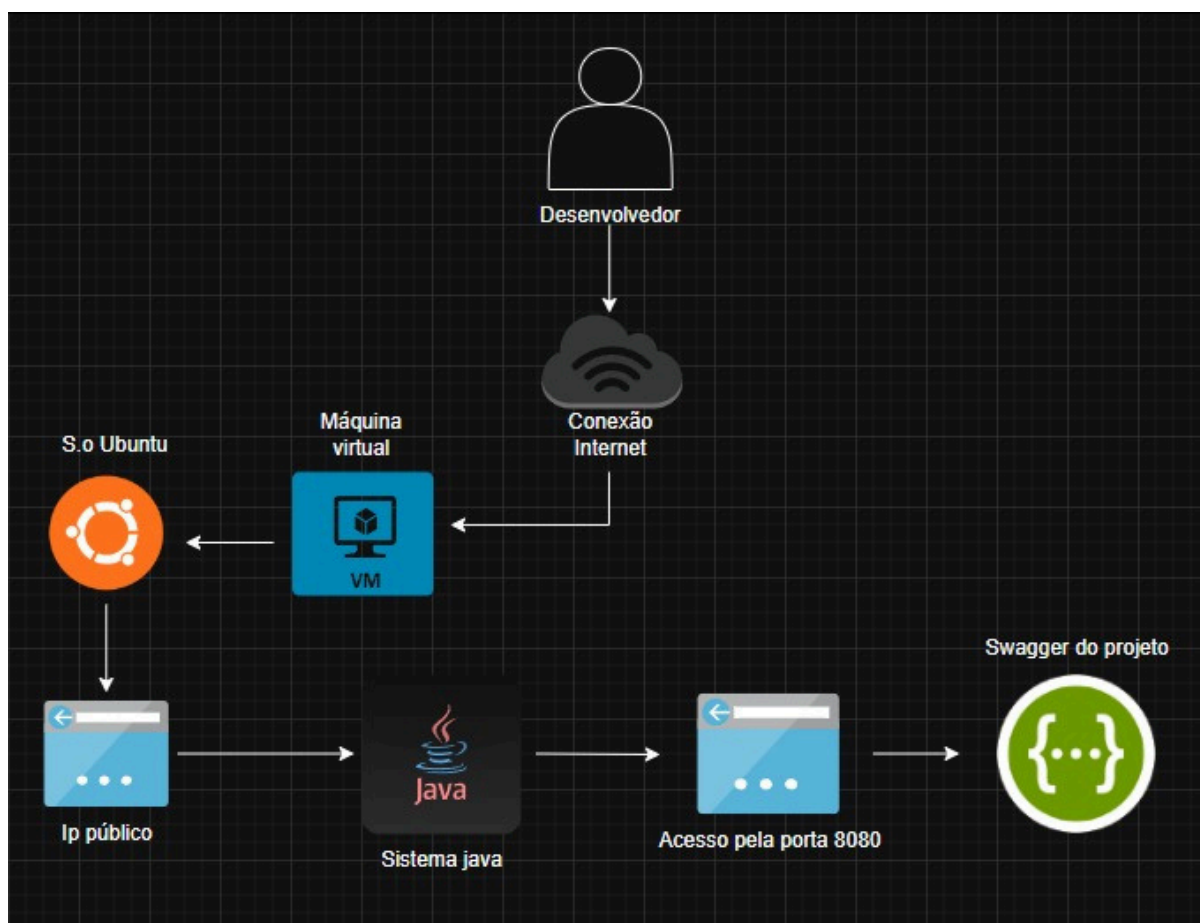
Excluindo a imagem



Agora vamos fazer o diagrama da solução



Arquitetura feita no Draw.io



Board finalizado

VisionHive Team

Board

Analytics

To Do

<

Doing

0/5

Done

<

+ New item

6

Desenhar a arquitetura macro

Done

7

Limpar laboratório

Done

5

Testes da aplicação

Done

4

Realizar o deploy do docker na vm

Done

1

Criação/Configuração Vm Linux

Done

3

Disponibilizar a imagem do docker no docker hub

Done

2

Subir o projeto de java no docker localmente

Done