Optamos por utilizar o Docker e Docker Compose. Foi necessário a instalação do Docker, Docker Compose e conector de banco do .Net na VM.

ssh-keygen -t rsa -b 2048 -f ~/.ssh/id\_rsa\_azure

ssh-keygen -t rsa -b 2048 -f C:\Users\flavi\.ssh\id\_rsa\_azure

type C:\Users\<Seu\_Usuario>\.ssh\id\_rsa\_azure.pub

ssh -i ~/.ssh/id\_rsa\_azure [RM552421@52.168.55.195](mailto:RM552421@52.168.55.195)

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update

sudo apt-get install -y docker-ce

sudo systemctl status Docker

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose –version

mkdir ~/ReciclaKi-db

cd ~/ReciclaKi-db

nano docker-compose.yml

version: '3.1'

services:

db:

image: postgres

restart: always

environment:

POSTGRES\_USER: user

POSTGRES\_PASSWORD: password

POSTGRES\_DB: mydb

ports:

- "5432:5432"

sudo docker-compose up -d

sudo docker os

{

"ConnectionStrings": {

"DefaultConnection": "Host=ip-publico-da-vm;Port=5432;Username=user;Password=password;Database=mydb"

}

}

dotnet add package Npgsql.EntityFrameworkCore.PostgreSQL

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<ApplicationDbContext>(options =>

options.UseNpgsql(Configuration.GetConnectionString("DefaultConnection")));

services.AddControllers();

}