

Příloha 1 - Uživatelská dokumentace serveru

A

Serverová aplikace dokáže pomocí definovaného API propojit programy ovládající různá zařízení. Interakci zajišťuje přes sdílené úložiště dat. Je schopná

- nahrávat data,
- mazat data,
- vracet data,
- upravovat data,
- hlásit změny
- a další.

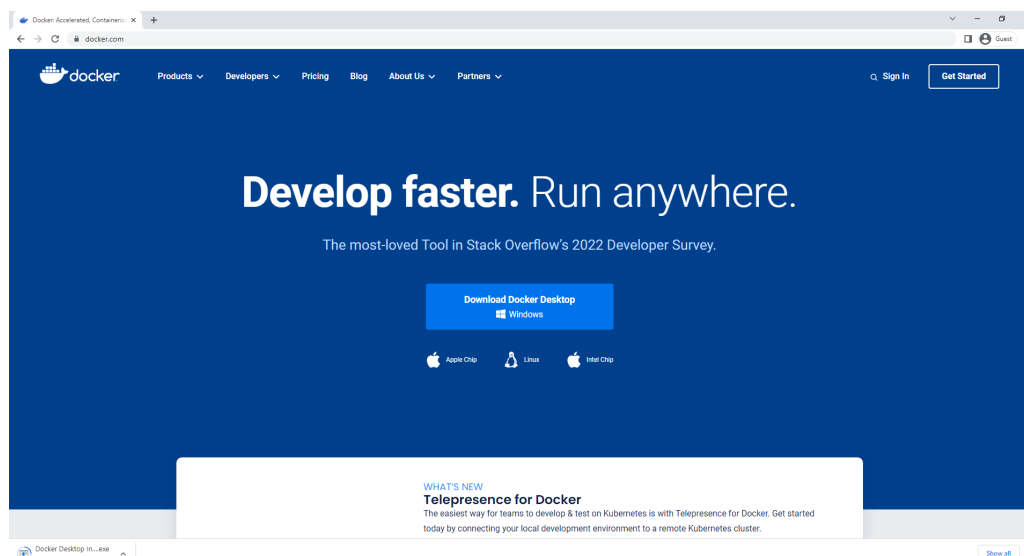
Server je distribuován jako zip archiv se zdrojovým kódem včetně nástrojů potřebných pro sestavení aplikace. Archiv je nutné rozbalit. Serverovou aplikaci je možné sestavit a spustit v Docker kontejneru.

A.1 Docker

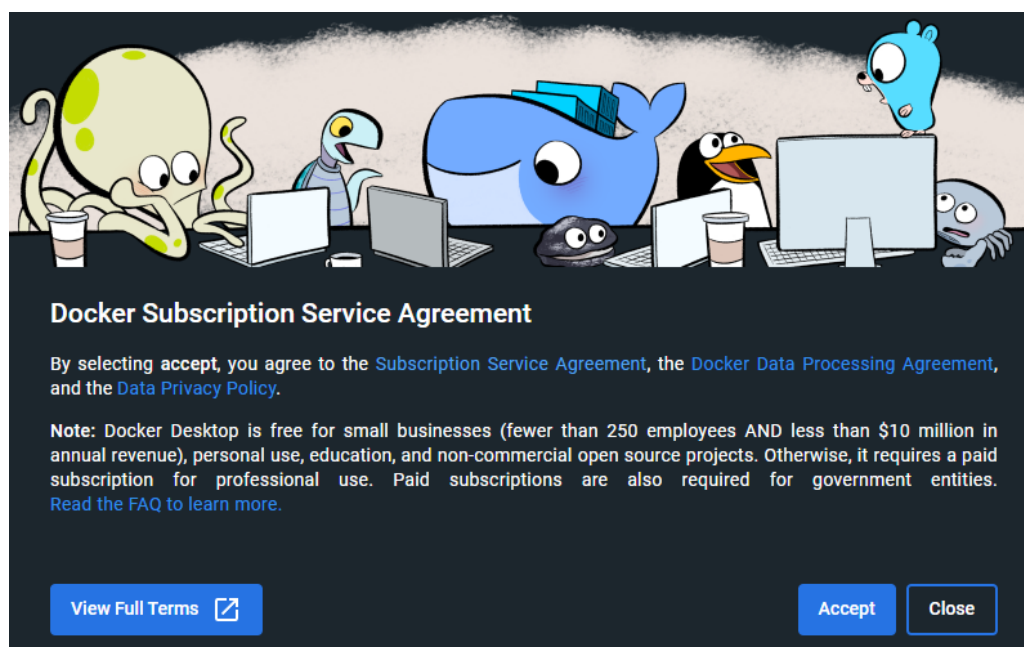
Serverová aplikace je primárně navržena pro běh v Dockeru. Pokud není Docker na počítači nainstalovaný, je nutné ho nejprve stáhnout. Instalační program je dostupný na stránkách <https://www.docker.com/>. Pro osobní, edukativní nebo open-source použití je Docker zdarma, takže není potřeba mít obavy o licenční podmínky.

Instalační program provede celou instalaci programu Docker Desktop. Popis instalace v tomto dokumentu je závislý na platformě Windows, pokud instalace probíhá na jinou platformu, bližší popis je k nahlédnutí na stránce <https://docs.docker.com/desktop/>.

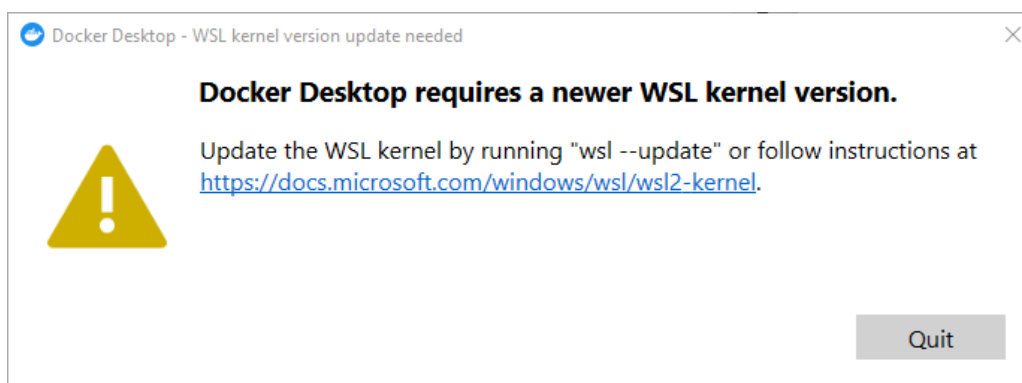
V průběhu instalace je vyžadováno provést restart operačního systému. Po něm stačí pouze potvrdit obchodní podmínky (viz obr. A.2) a může se Docker využívat.



Obrázek A.1: Stažení Docker Desktop



Obrázek A.2: Obchodní podmínky Docker Desktop



Obrázek A.3: Chybějící WSL

Po instalaci dojde k automatickému spuštění. Může se stát, že se ukáže chybová hláška o nenainstalovaném nebo zastaralém WSL jádře jako na obr. A.3.

Nainstalování Subsystem Windows pro Linux na Windows 10 verze 2004 a výš (Build 19041 a výš) nebo Windows 11 se provádí přes příkazovou řádku cmd.exe v režimu administrátora. V ní stačí spustit příkaz `wsl --install`. Konzolový výpis A.1 představuje ukázkový běh příkazu. Po dokončení instalace WSL lze Docker spustit. V případě starší verze Windows je instalace složitější a vyžaduje stažení instalačního souboru na adrese <https://learn.microsoft.com/en-us/windows/wsl/install-manual>.

Výpis A.1: Příkaz na instalaci WSL

```

1 C:\WINDOWS\system32>wsl --install
2 Installing: Virtual Machine Platform
3 Virtual Machine Platform has been installed.
4 Installing: Windows Subsystem for Linux
5 Windows Subsystem for Linux has been installed.
6 Downloading: WSL Kernel
7 Installing: WSL Kernel
8 WSL Kernel has been installed.
9 Downloading: Ubuntu
10 The requested operation is successful. Changes will not be
    effective until the system is rebooted.
11
12 C:\WINDOWS\system32>
```

A.1.1 Docker compose

Uvnitř adresářové struktury je k dispozici soubor `ZCU.TechnologyLab.VirtualWorld-Server/docker-compose.yml`. Obsahuje příkazy, které popisují způsob sestavení a

popis vlastností prostředí pro běh aplikace v Dockeru. S docker compose se pracuje v příkazové řádce. Příkazy automaticky použijí nejbližší docker-compose soubor.

1. Přesunout aktivní složku příkazové řádky do složky ZCUTechnologyLab.VirtualWorldServer příkazem `cd`
2. Vytvořit a sestavit kontejner v dockeru příkazem `docker compose create`
3. Spustit kontejner se serverem příkazem `docker compose start`

Výpis A.2: Příkaz na vytvoření kontejneru

```
1 C:\server\ZCU.TechnologyLab.VirtualWorldServer>docker compose create
2 [+] Running 0/0
3 - zcu.technologylab.virtualworldserver Pulling 0.1s
4 [+] Building 3.9s (5/17)
5 => [internal] load metadata for
   mcr.microsoft.com/dotnet/sdk:6.0 0.4s
6 [+] Building 28.4s (18/18) FINISHED
7 => [internal] load build definition from Dockerfile 0.1s
8 => => transferring dockerfile: 971B 0.0s
9 => [internal] load .dockerignore 0.0s
10 => => transferring context: 382B 0.0s
11 => [internal] load metadata for
   mcr.microsoft.com/dotnet/sdk:6.0 0.4s
12 => [internal] load metadata for
   mcr.microsoft.com/dotnet/aspnet:6.0 0.5s
13 => [internal] load build context 0.2s
14 => => transferring context: 72.71kB 0.2s
15 => [build 1/7] FROM
   mcr.microsoft.com/dotnet/sdk:6.0@sha256:a3bbff689a86ba7f3d
   dcee5089a729b20e20e3b4dbfb9d0a43b 14.9s
16
17 => => resolve
   mcr.microsoft.com/dotnet/sdk:6.0@sha256:a3bbff689a86ba7f3d
   dcee5089a729b20e20e3b4dbfb9d0a43bb3284d9 0.0s
18
19
20 ... hashování a extrakce dat jsou z výpisu vynechány
21
22 => [base 1/2] FROM
   mcr.microsoft.com/dotnet/aspnet:6.0@sha256:f76f95813a87d71
   1928c6b02335614f328cc1c9c53315137fd 7.1s
23
24 => => resolve
   mcr.microsoft.com/dotnet/aspnet:6.0@sha256:f76f95813a87d71
   1928c6b02335614f328cc1c9c53315137fd64ea1 0.0s
25
26
27 ... hashování a extrakce dat jsou z výpisu vynechány
28
29 => [base 2/2] WORKDIR /app 1.6s
```

```

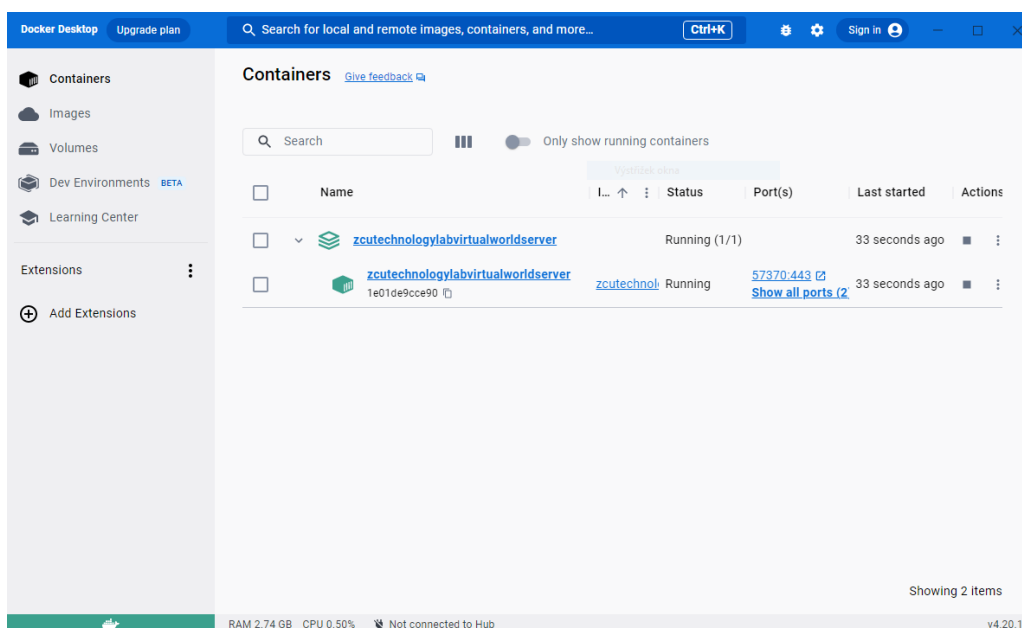
30 => [final 1/2] WORKDIR /app 0.1s
31 => [build 2/7] WORKDIR /src 1.5s
32 => [build 3/7] COPY
    [ZCU.TechnologyLab.VirtualWorldServer/ZCU.TechnologyLab.Vi
33 rtualWorldServer.csproj, ZCU.Techn 0.1s
34 => [build 4/7] RUN dotnet restore
    "ZCU.TechnologyLab.VirtualWorldServer/ZCU.TechnologyLab.Vi
35 rtualWorldServer.csp 4.4s
36 => [build 5/7] COPY . . 0.1s
37 => [build 6/7] WORKDIR
    /src/ZCU.TechnologyLab.VirtualWorldServer 0.1s
38 => [build 7/7] RUN dotnet build
    "ZCU.TechnologyLab.VirtualWorldServer.csproj" -c Release
    -o /app/build 4.3s
39 => [publish 1/1] RUN dotnet publish
    "ZCU.TechnologyLab.VirtualWorldServer.csproj" -c Release
    -o /app/publish 2.4s
40 => [final 2/2] COPY --from=publish /app/publish . 0.1s
41 => exporting to image 0.1s
42 => => exporting layers 0.1s
43 => => writing image
    sha256:3cf6ceeced767b120937ad7463a7e9f757780ca4fea34142d76
44 43d15cbafc562 0.0s
45 => => naming to
    docker.io/library/zcutechnologylabvirtualworldserver 0.0s
46 [+] Running 2/2
47 ✓ Network zcutechnologylabvirtualworldserver_default
    Cre... 0.1s
48 ✓ Container zcutechnologylabvirtualworldserver
    Created 0.1s
49 C:\server\ZCU.TechnologyLab.VirtualWorldServer>docker compose start
50 [+] Running 1/1
51 ✓ Container zcutechnologylabvirtualworldserver Started 0.4s
52 C:\server\ZCU.TechnologyLab.VirtualWorldServer>

```

A.1.2 Práce s kontejnerem

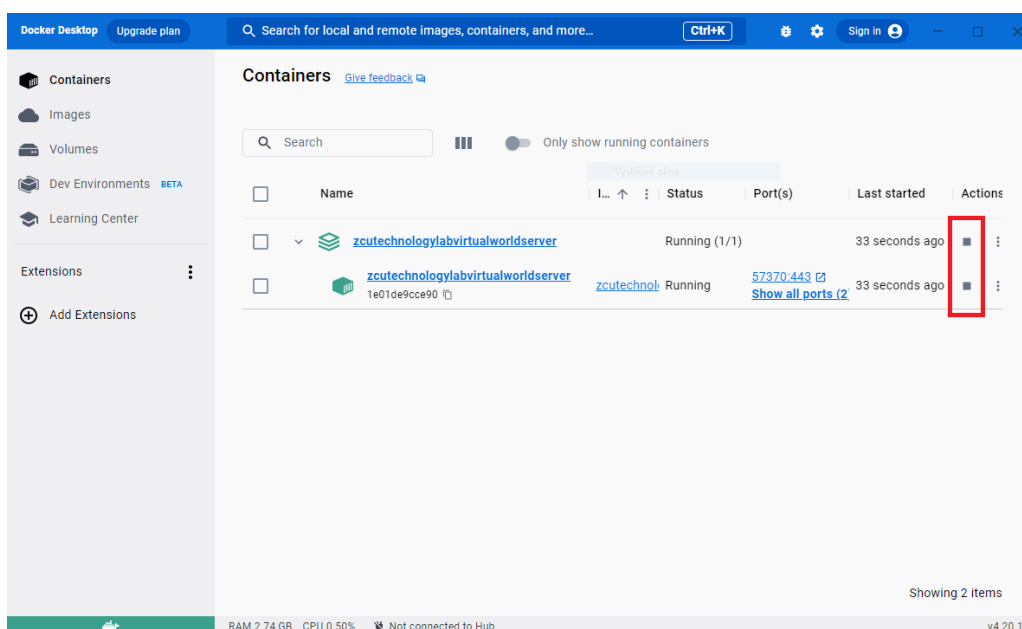
Kromě příkazové řádky lze přistupovat ke kontejneru také z aplikace Docker Desktop. Na hlavní obrazovce A.4 lze vidět seznam nainstalovaných kontejnerů. Běžící kontejner je možné zastavit tlačítkem se symbolem čtverce v aplikaci. Na obr. A.5 jsou tlačítka vyznačena. Na stejném místě se nalézá také tlačítko pro spuštění kontejneru, které se zobrazuje, když je kontejner zastavený.

Při kliknutí na vyznačený kontejner v obr. A.7 se zobrazí logovací obrazovka A.8. Jedná se o dobré místo na kontrolu funkčnosti serveru v případě nějakého neočekávaného chování. V pozdější části dokumentace bude práce s logy ukázána

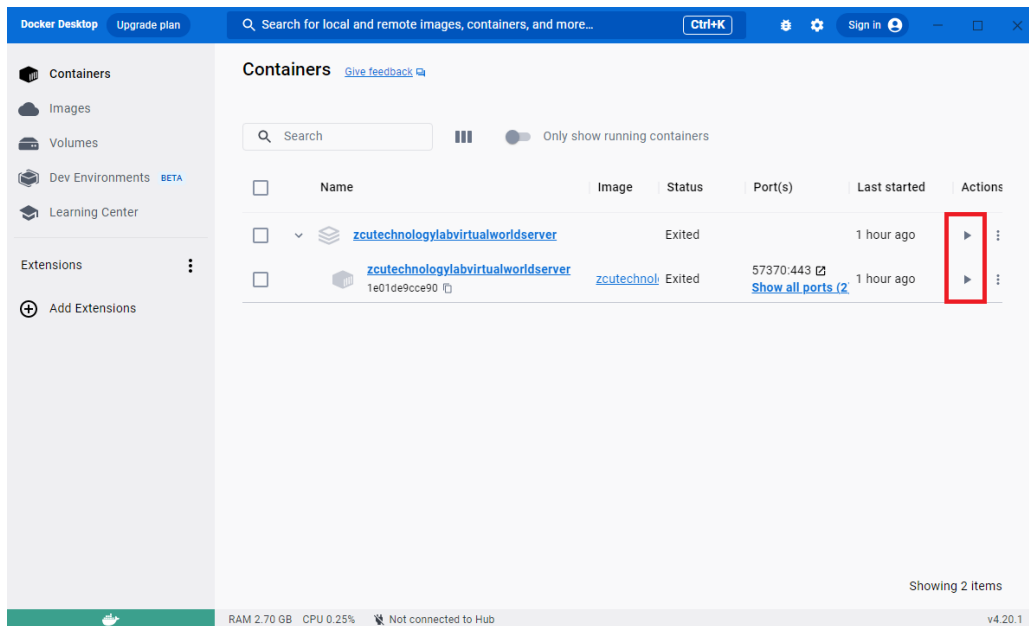


Obrázek A.4: Seznam kontejnerů v Docker Desktop

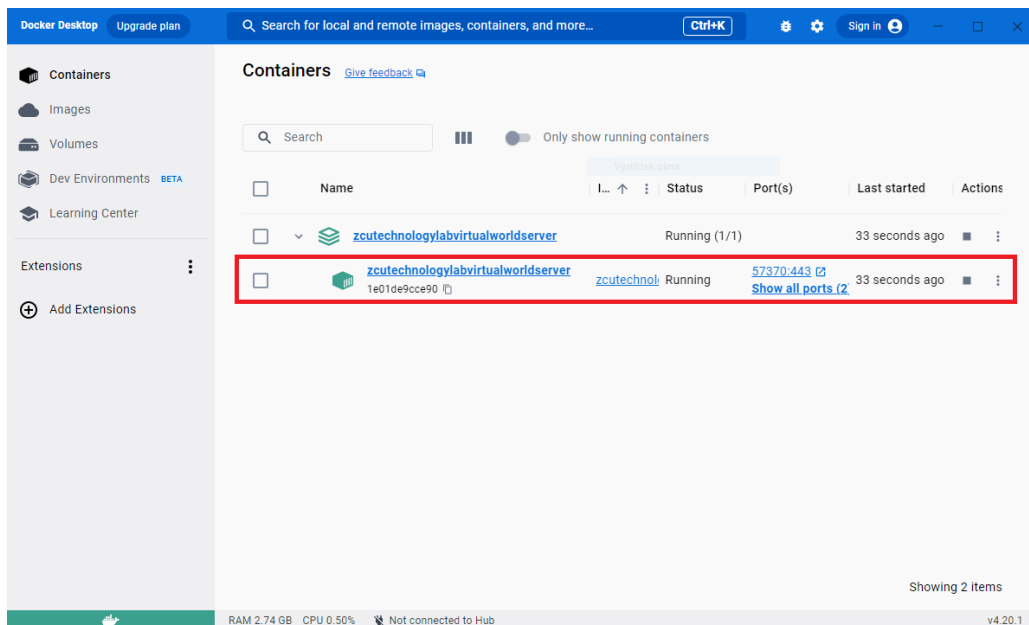
blíže. Významná je také obrazovka s vlastnostmi kontejneru pod záložkou „Inspect“. K nahlédnutí na obr. A.9 jsou například verze jazyka a frameworku, nastavené porty nebo adresářová struktura uvnitř kontejneru.



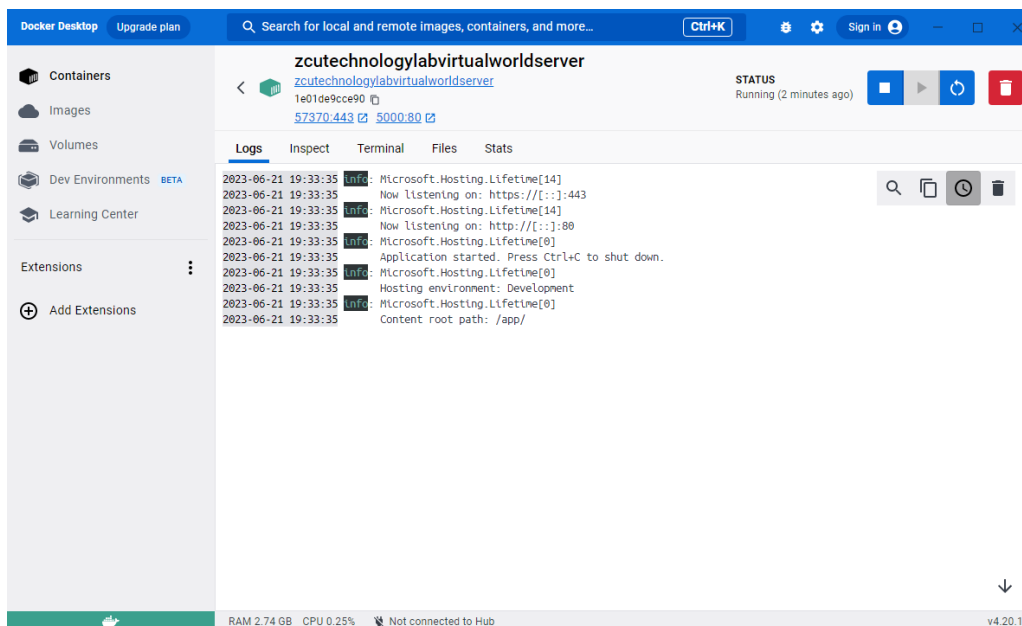
Obrázek A.5: Zastavení kontejneru v Docker Desktop



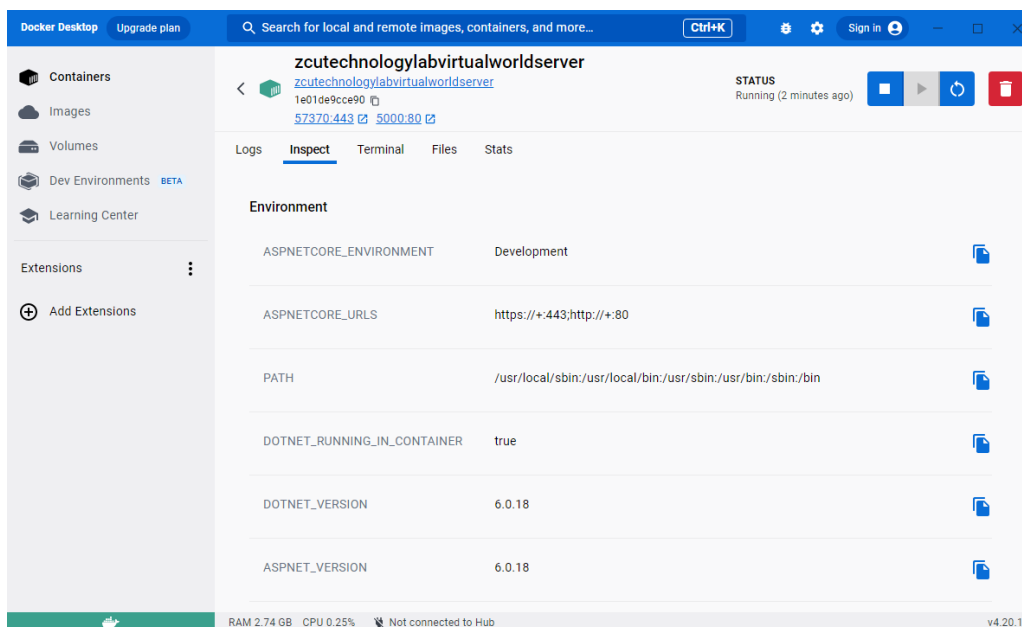
Obrázek A.6: Spuštění kontejneru v Docker Desktop



Obrázek A.7: Výběr kontejneru v Docker Desktop



Obrázek A.8: Logy v Docker Desktop



Obrázek A.9: Vlastnosti kontejneru v Docker Desktop

A.2 Komunikační API

Server vystavuje komunikační rozhraní veřejně. Nástrojem Swagger je možné na rozhraní nahlédnout a vykoušet odeslání dotazů na server. Na stejném zařízení, kde běží Docker, otevřete prohlížeč a do adresy zadejte `http://localhost:5000/swagger`. Pokud je vše správně nastaveno, měla by se otevřít webová stránka se Swaggerem na obr. A.10.

Otevřete POST `/api/objects`, zde uvidíte formát objektu v těle zprávy a možnost vyzkoušet dotaz. Klikněte na „Try it out“ a nastavte hodnoty na

Zdrojový kód A.3: Testovací JSON

```

1 {
2   "Name": "testWorldObject",
3   "Position": {
4     "X": 1,
5     "Y": 2,
6     "Z": 3
7   },
8   "Rotation": {
9     "X": 4,
10    "Y": 5,
11    "Z": 6
12  },
13  "Scale": {
14    "X": 7,
15    "Y": 8,
16    "Z": 9
17  },
18  "Type": "Bitmap",
19  "Properties": {
20  }
21 }
```

Klikněte na tlačítko „Execute“. Jděte do Docker Desktop, otevřete logy a zkontrolujte, že výpis v A.4 odpovídá.

Výpis A.4: Logy

```

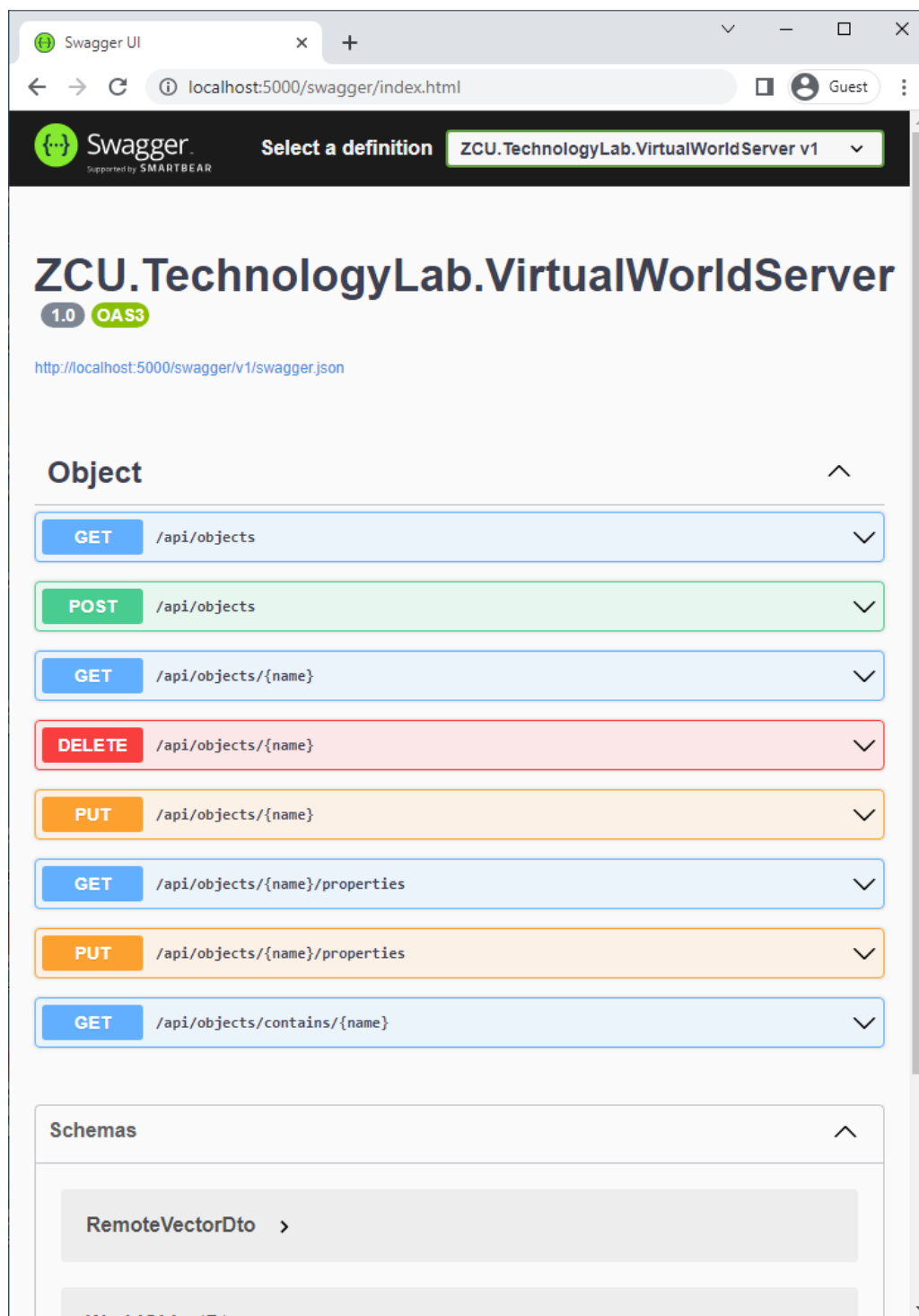
1 2023-06-22 00:41:40 info: Microsoft.Hosting.Lifetime[0]
2 2023-06-22 00:41:40      Application is shutting down...
3 2023-06-22 00:41:43 info: Microsoft.Hosting.Lifetime[14]
4 2023-06-22 00:41:43      Now listening on: https://[::]:443
5 2023-06-22 00:41:43 info: Microsoft.Hosting.Lifetime[14]
6 2023-06-22 00:41:43      Now listening on: http://[::]:80
7 2023-06-22 00:41:43 info: Microsoft.Hosting.Lifetime[0]
8 2023-06-22 00:41:43      Application started. Press Ctrl+C
      to shut down.
```

```
9 2023-06-22 00:41:43 info: Microsoft.Hosting.Lifetime[0]
10 2023-06-22 00:41:43      Hosting environment: Development
11 2023-06-22 00:41:43 info: Microsoft.Hosting.Lifetime[0]
12 2023-06-22 00:41:43      Content root path: /app/
13 2023-06-22 00:54:41 info: ZCU.TechnologyLab.VirtualWorldServ
14     er.Services.WorldObjectService[0]
15 2023-06-22 00:54:41      Adding object testWorldObject to
    the server
```

Logy potvrzují, že byl objekt na server přidáný. Nyní ho zkusíme získat zpět. Otevřete GET /api/objects/{name}. Klikněte na „Try it out“ a do parametru name zadejte testWorldObject. Klikněte na tlačítko „Execute“. Tělo odpovědi by mělo být stejné jako testovací JSON v ukázce A.3.

Smažte objekt ze serveru. Otevřete DELETE /api/objects/{name}. Klikněte na „Try it out“. Do jména zadejte testWorldObject. Klikněte na „Execute“ a zkontrolujte, že vrácený stavový kód je 200 a tělo odpovědi obsahuje text „Object removed“.

Můžete znovu zkontrolovat logy. Budou v nich všechny prováděné akce.



Obrázek A.10: Swagger

