

# 1. نصب و راه اندازی داکر و WLS

ابتدا نرم افزار docker را از لینک زیر دانلود و نصب کنید. مراحل نصب به شرح زیر میباشد.

<https://www.docker.com/products/docker-desktop>

Installing Docker Desktop 4.31.1 (153621)

## Configuration

- ☒ Use WSL 2 instead of Hyper-V (recommended)
- ☒ Add shortcut to desktop

Installing Docker Desktop 4.31.1 (153621)

## Docker Desktop 4.31.1

Unpacking files...

```
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/dvdp.iso
Unpacking file: resources/config/options.json
Unpacking file: resources/components/Version.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/gigignore
Unpacking file: InstallerCLI.pdb
Unpacking file: InstallerCLI.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/vk_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/vsutil.vbs
Unpacking file: frontend/resources/regedit/vbs/regUtil.vbs
Unpacking file: frontend/resources/regedit/vbs/regPutValue.wsf
Unpacking file: frontend/resources/regedit/vbs/regListenStream.wsf
Unpacking file: frontend/resources/regedit/vbs/regList.wsf
Unpacking file: frontend/resources/regedit/vbs/regDeleteValue.wsf
Unpacking file: frontend/resources/regedit/vbs/regDeleteKey.wsf
Unpacking file: frontend/resources/regedit/vbs/regCreateKey.wsf
Unpacking file: frontend/resources/regedit/vbs/JsonSafeTest.wsf
Unpacking file: frontend/resources/regedit/vbs/ArchitectureSpecificRegistry.vbs
Unpacking file: frontend/resources/regedit/vbs/ArchitectureAgnosticRegistry.vbs
Unpacking file: frontend/resources/LICENSE.rtf
Unpacking file: frontend/resources/dist/OSS-LICENSES
Unpacking file: frontend/resources/assets/app/icon.png
Unpacking file: frontend/resources/assets/app/icon.ico
Unpacking file: frontend/resources/assets/app/icon.16x16.png
Unpacking file: frontend/resources/app.asar.unpacked/worker/conoutSocketWorker.js.map
Unpacking file: frontend/resources/app.asar.unpacked/worker/conoutSocketWorker.js
Unpacking file: frontend/resources/app.asar.unpacked/shared/conout.js.map
Unpacking file: frontend/resources/app.asar.unpacked/shared/conout.js
Unpacking file: frontend/resources/app.asar.unpacked/conpty_console_list_agent.js
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/vimpty.lib
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/vimpty.ipdb
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/vimpty.iobj
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/vimpty.exe
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/vimpty-agent.ipdb
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/vimpty-agent.iobj
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/pty.node
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/pty.ipdb
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/pty.iobj
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/conpty_console_list.node
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/conpty_console_list.ipdb
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/conpty_console_list.iobj
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/conpty.node
Unpacking file: frontend/resources/app.asar.unpacked/build/Release/conpty.ipdb
```

## Docker Desktop 4.31.1

Installing...

Deploying component: Use WSL 2 instead of Hyper-V (recommended)  
Deploying component: Add user to docker-users group  
Deploying component: Create docker-users group  
Installing components  
Unpacking file: System.Xml.XPath.XDocument.dll  
Unpacking file: System.Xml.XPath.dll  
Unpacking file: System.Xml.XmlSerializer.dll  
Unpacking file: System.Xml.XmlDocument.dll  
Unpacking file: System.Xml.XDocument.dll  
Unpacking file: System.Xml.ReaderWriter.dll  
Unpacking file: System.Web.Http.Owin.dll  
Unpacking file: System.Web.Http.dll  
Unpacking file: System.ValueTuple.dll  
Unpacking file: System.Threading.Timer.dll  
Unpacking file: System.Threading.ThreadPool.dll  
Unpacking file: System.Threading.Thread.dll  
Unpacking file: System.Threading.Tasks.Parallel.dll  
Unpacking file: System.Threading.Tasks.Extensions.dll  
Unpacking file: System.Threading.Tasks.dll  
Unpacking file: System.Threading.Overlapped.dll  
Unpacking file: System.Threading.dll  
Unpacking file: System.Text.RegularExpressions.dll  
Unpacking file: System.Text.Encoding.Extensions.dll  
Unpacking file: System.Text.Encoding.dll  
Unpacking file: System.Security.SecureString.dll  
Unpacking file: System.Security.Principal.dll  
Unpacking file: System.Security.Cryptography.X509Certificates.dll  
Unpacking file: System.Security.Cryptography.Primitives.dll  
Unpacking file: System.Security.Cryptography.Encoding.dll  
Unpacking file: System.Security.Cryptography.Csp.dll  
Unpacking file: System.Security.Cryptography.Algorithms.dll  
Unpacking file: System.Security.Claims.dll  
Unpacking file: System.Runtime.Serialization.Xml.dll  
Unpacking file: System.Runtime.Serialization.Primitives.dll  
Unpacking file: System.Runtime.Serialization.Json.dll  
Unpacking file: System.Runtime.Serialization.Formatters.dll  
Unpacking file: System.Runtime.Numerics.dll  
Unpacking file: System.Runtime.InteropServices.RuntimeInformation.dll  
Unpacking file: System.Runtime.InteropServices.dll  
Unpacking file: System.Runtime.Handles.dll  
Unpacking file: System.Runtime.Extensions.dll  
Unpacking file: System.Runtime.dll  
Unpacking file: System.Runtime.CompilerServices.VisualC.dll  
Unpacking file: System.Runtime.CompilerServices.Unsafe.dll  
Unpacking file: System.Resources.Writer.dll  
Unpacking file: System.Resources.ResourceManager.dll  
Unpacking file: System.Resources.Reader.dll  
Unpacking file: System.Reflection.Primitives.dll

## Docker Desktop 4.31.1

Installation succeeded

You must restart Windows to complete installation.

Close and restart



## Docker Subscription Service Agreement

By selecting **accept**, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), and the [Data Privacy Policy](#).

Commercial use of Docker Desktop at a company of more than 250 employees OR more than \$10 million in annual revenue requires a paid subscription (Pro, Team, or Business). [See subscription details](#)

[View Full Terms](#)



**Accept**

**Close**



## Finish setting up Docker Desktop

version 4.31.1 (153621)

Complete the installation of Docker Desktop.

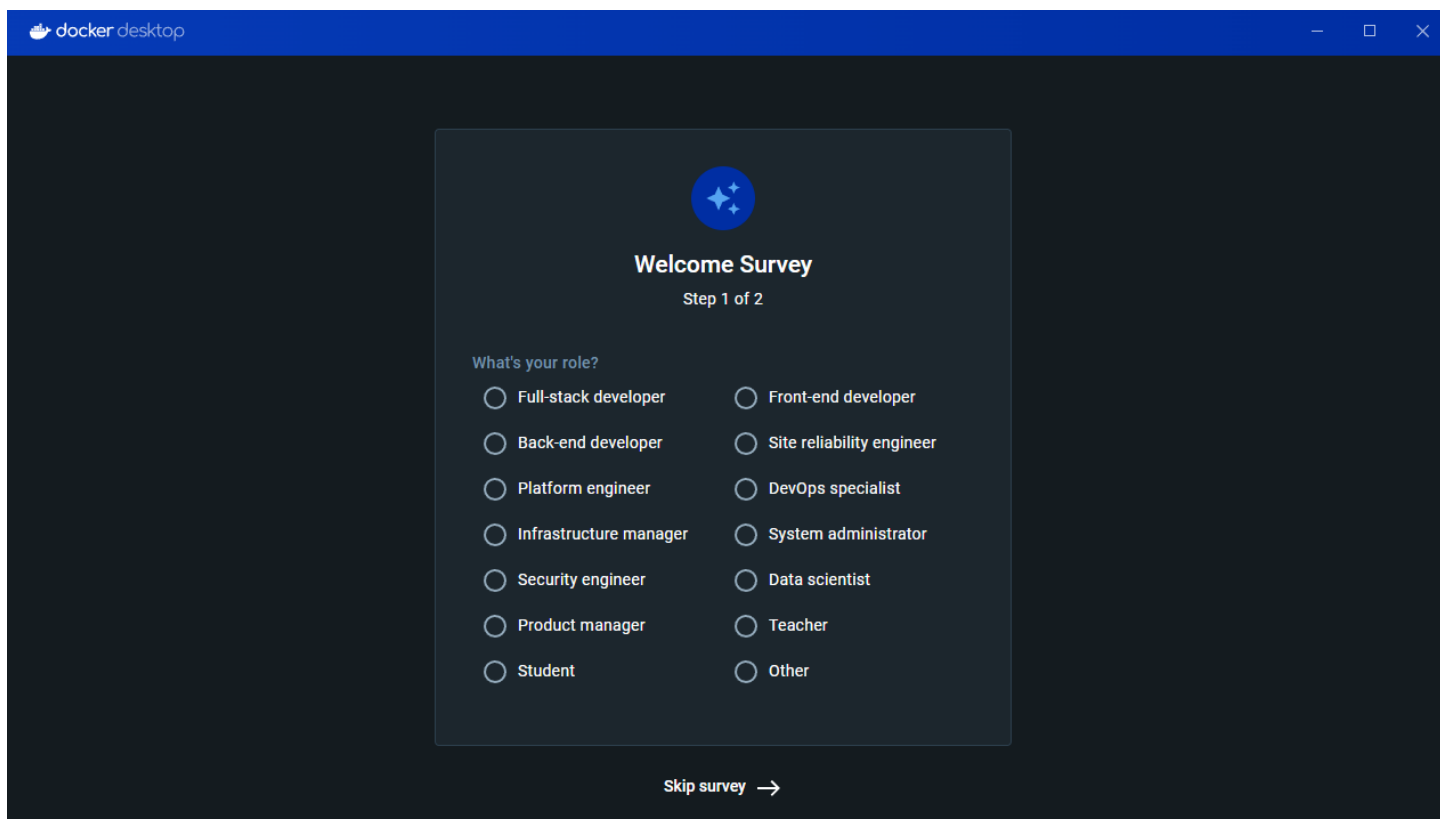
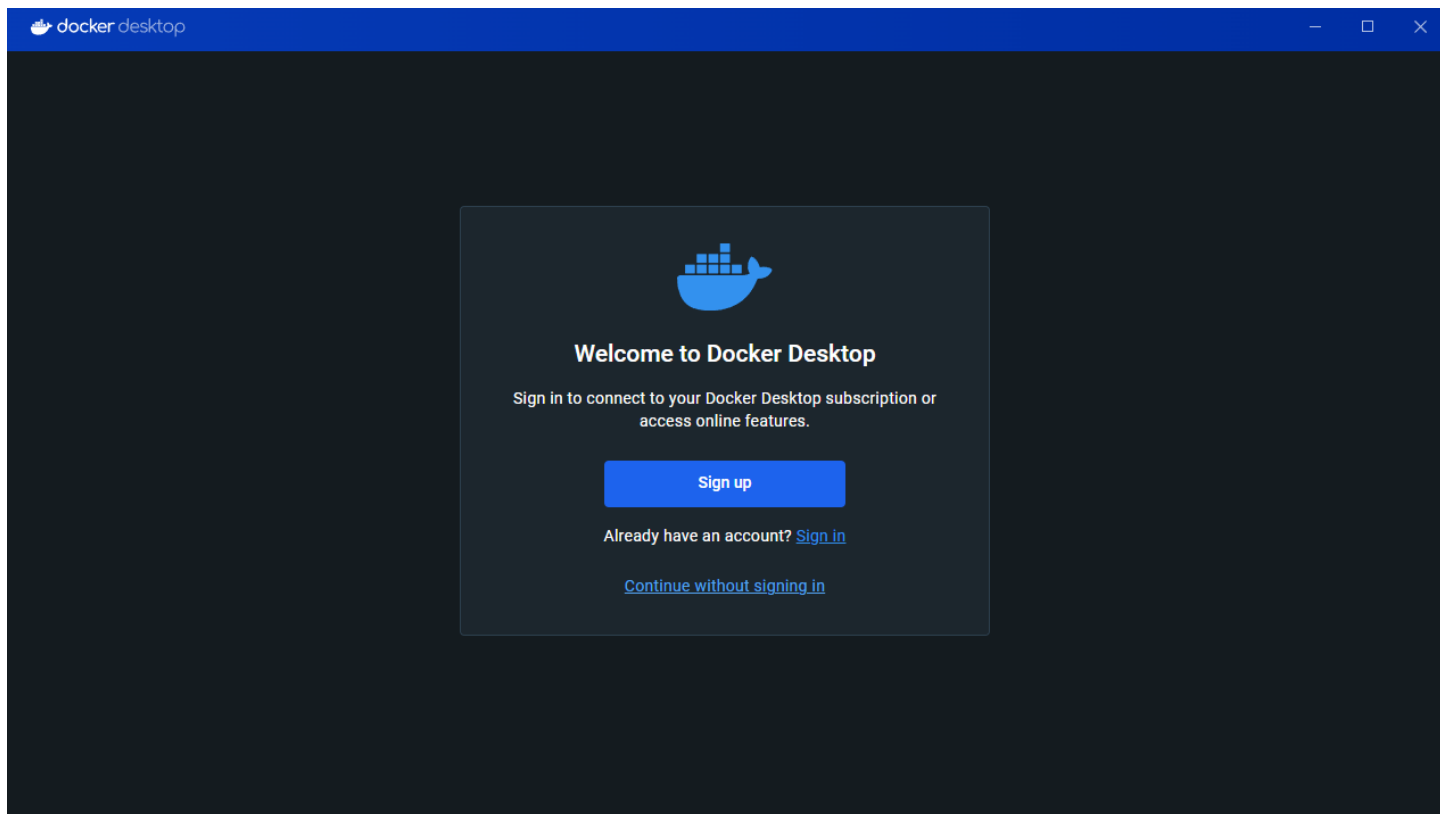
☒ **Use recommended settings (requires administrator password)**

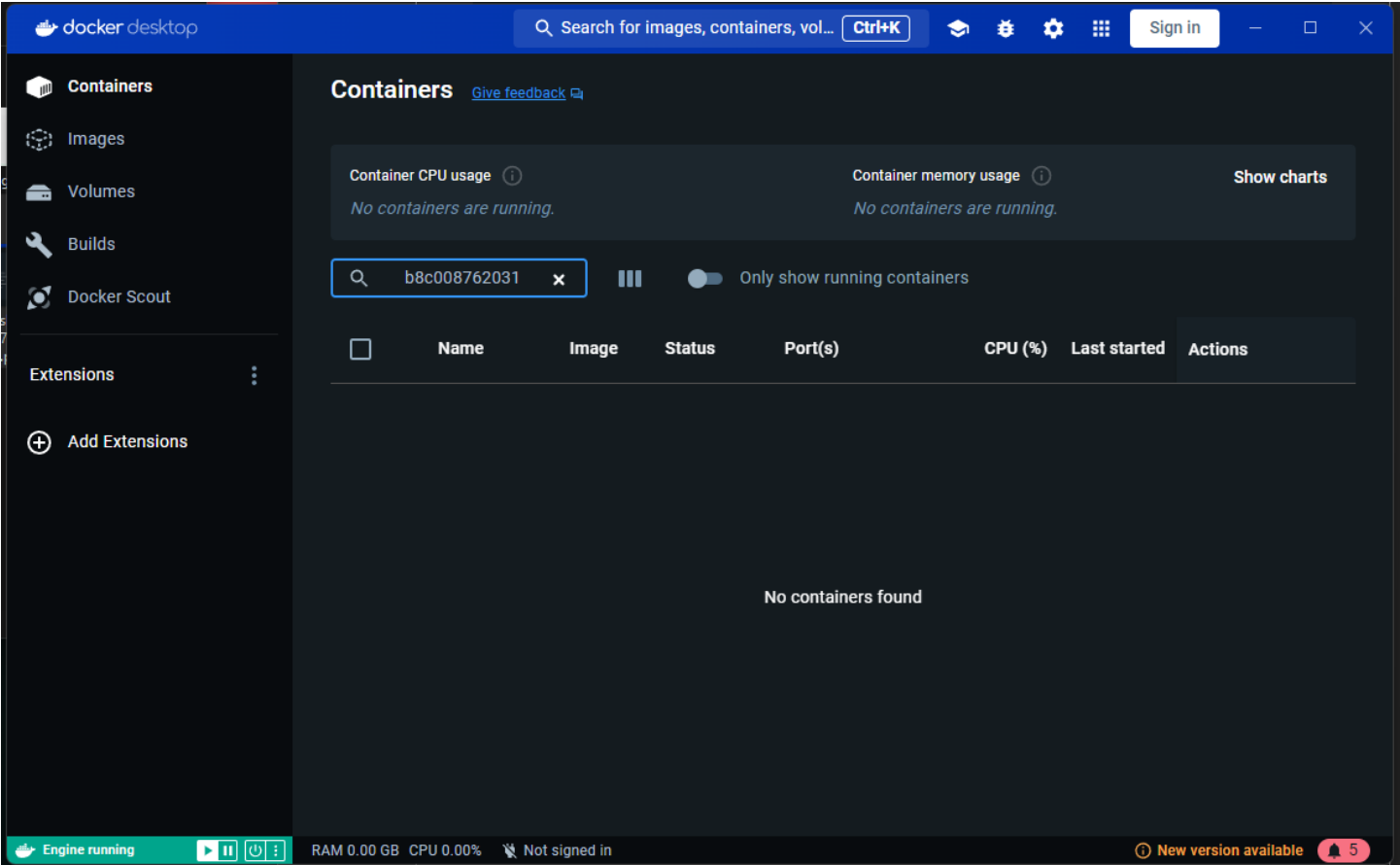
Docker Desktop automatically sets the necessary configurations that work for most developers.

☐ **Use advanced settings**

You manually set your preferred configurations.

**Finish**





## 2. ساخت کانتینر مربوط به برنامه

- سپس در مسیر اصلی برنامه یک ترمینال باز کرده و دستور زیر را در آن اجرا میکنیم.

## **docker compose up --build**

- در نهایت خروجی زیر در ترمینال نمایش داده میشود:

```
C:\Users\PCMOO>cmdtop\gfs-main>docker compose up --build
```

```
[+] Building 0.8s (0/0) docker:default
[+] Building 0.5s (12/12) FINISHED
=> [server internal] load build definition from Dockerfile
=> transferring dockerfile: 1.72kB
[server] resolve image config for docker://docker.io/docker/dockerfile:1
[server] docker-image //docker.io/docker/dockerfile:1:b9ba256:e87caa74dc7b4d6cd829352bfca12591f3dba3ddc4285e19c7dcd13359f7cefd
=> resolve docker.io/docker/dockerfile:1:b9ba256:e87caa74dc7b4d6cd829352bfca12591f3dba3ddc4285e19c7dcd13359f7cefd
=> sha256:25d111eb93307e232a1211001304eefcfab0af4311f98508d21f772da95 1.26kB / 1.26kB
=> sha256:1d7e21b9ea3117824ebc8797e591ad18a37dff6affbd175f7db6c95aa532 12.09MB / 12.09MB
=> sha256:e87caa74dc7b4d6cd829352bfca12591f3dba3ddc4285e19c7dcd13359f7cefd 8.40kB / 8.40kB
=> sha256:dcc2273f8d8a29297ae2b7581f82089c976f0474bc313967c7e7bf493251a 402B / 402B
=> extracting sha256:1d7e21b9ea3117824ebc8797e591ad18a37dff6affbd175f7db6c95aa532
[server internal] load metadata for docker.io/library/python:3.12.1-slim
[server internal] <!-- deduplicate -->
=> transferring context: 705B
[server base 1/5] FROM docker.io/library/python:3.12.1-slim@sha256:a56ac5be6928c94f08b1fe09cdf3ba3edd4455241dfaf516a550804737573e
[server internal] load build context
=> transferring context: 17.19kB
[server base 2/5] WORKDIR /app
[server base 3/5] RUN adduser --disabled-password --gecos "" --home "/nonexistent" --shell "/sbin/nologin" --no-create-home --uid "10001" appuser
[server base 4/5] RUN --mount=type:cache,target=/root/.cache/pip --mount=type:bind,source=requirements.txt,target=requirements.txt python -m pip install -r requirements.txt
[server base 5/5] COPY . .
[server] exporting to image
=> exporting layers
=> writing image sha256:22b6d075c23f1e74fcdb112fb8fc7980493d5913c5295f18ca92c915b0aa1e
=> naming to docker.io/library/gfs-main-server
[*] Running 2/2
```

```
Network gfs-main_default Created
Container gfs-main-server-1 Created
```

```
Attaching to server-1
server-1 | self.status = 'fetching'
server-1 | 2024-07-05 18:11:29.709352
server-1 | 29240795
server-1 | https://nomads.ricep.noaa.gov/cgi-bin/filter_gfs_ops.pl?dir=x2Zgfs.20240705ZJ210N2F.atmos&file=gfs.t18z.pgrib2.0p25.F90S3k1_varmonGall_levmon&subregion&toplat=40&leftlon=40&rightlon=65&bottomlat=20
server-1 | res.status.code = 208
server-1 | downloaded: 7.08 MB
server-1 | self.status = 'convert grib'
server-1 | public/20240705/gfs.t18z.pgrib2.0p25.F900
server-1 | public/[[]:0~2][[:0~2])/download
server-1 | http://127.0.0.1:5080/download/?file-name=public/20240705/gfs.t18z
server-1 | created
server-1 | self.status = 'idle'
server-1 | * Serving Flask app 'src.server'
server-1 | * Debug mode: off
server-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
```

```
server-1 | * Running on all addresses (0.0.0.0)
server-1 | * Running on http://127.0.0.1:5080
server-1 | * Running on http://172.18.0.2:5080
server-1 | Press CTRL-C to quit
```

```
View In Docker Desktop  Enable Watch
```

```
12:13 PM 7/5/2024
```

- به یاد داشته باشید که حتماً دی ان اس 403.online بر روی نتورک سیستم تان فعال باشد.

دی ان اس های 403 به شرح زیر میباشند:

10.202.10.202

10.202.10.102

### 3. دریافت اطلاعات.

- برای آپدیت کردن اطلاعات ذخیره شده روی سرور باید یک http request با متد GET به api داخلی <http://127.0.0.1:5000/update/> ارسال شود.
- برای دریافت اطلاعات هوشناسی از طریق ریکوعست به api ی

```
"http://127.0.0.1:5000/download/?file-  
name=public/{:0>2}{:0>2}/gfs.xlsx".format(  
    self.date_instance.year,  
    self.date_instance.month,  
    self.date_instance.day  
)
```

در صورت وجود دریافت میشوند.