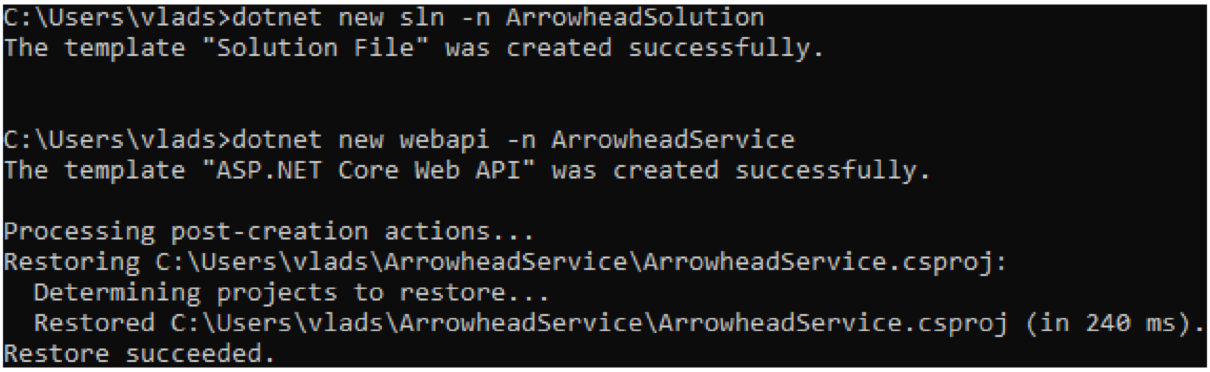
I tried to make some applications to familiarize myself with Arrowhead. I say "tried" because I had quite a few struggles with it.

First, I tried to create a simple service provider and a service consumer in C#.

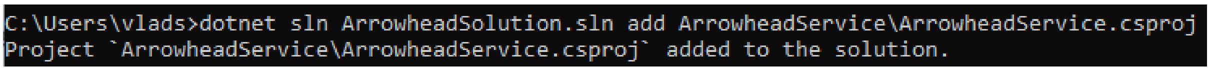
I downloaded and installed the latest versions of .NET Core SDK and Visual Studio Code from the official websites.

I opened Command Prompt and navigated to the directory where I wanted to create my project. I created a new solution and an ASP.NET Core Web API project by running the following commands:



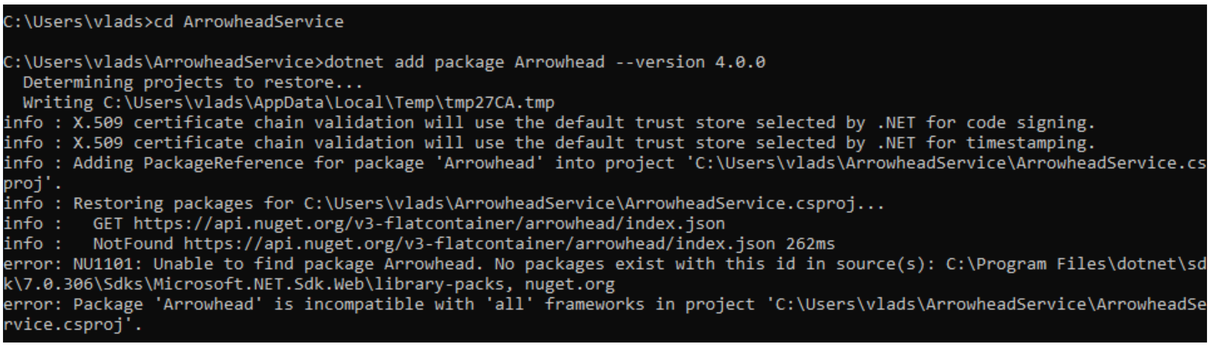
```
C:\Users\vlads>dotnet new sln -n ArrowheadSolution
The template "Solution File" was created successfully.


C:\Users\vlads>dotnet new webapi -n ArrowheadService
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring C:\Users\vlads\ArrowheadService\ArrowheadService.csproj:
  Determining projects to restore...
  Restored C:\Users\vlads\ArrowheadService\ArrowheadService.csproj (in 240 ms).
Restore succeeded.
```

Next, I added the project to the solution:



```
C:\Users\vlads>dotnet sln ArrowheadSolution.sln add ArrowheadService\ArrowheadService.csproj
Project `ArrowheadService\ArrowheadService.csproj` added to the solution.
```

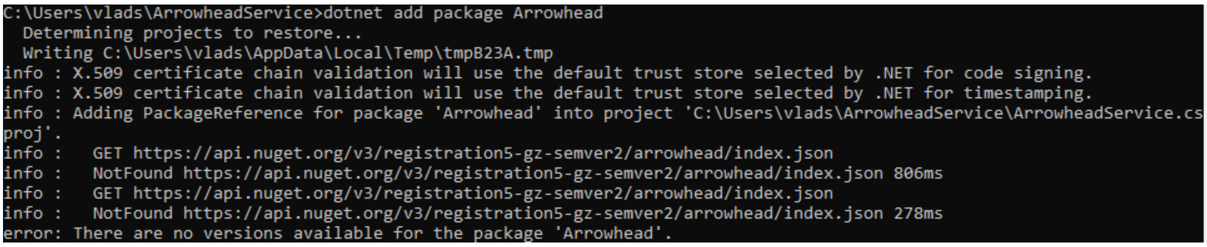
When I wanted to Install the Arrowhead NuGet package i received the next errors:



```
C:\Users\vlads>cd ArrowheadService

C:\Users\vlads\ArrowheadService>dotnet add package Arrowhead --version 4.0.0
  Determining projects to restore...
  Writing C:\Users\vlads\AppData\Local\Temp\tmp27CA.tmp
info : X.509 certificate chain validation will use the default trust store selected by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store selected by .NET for timestamping.
info : Adding PackageReference for package 'Arrowhead' into project 'C:\Users\vlads\ArrowheadService\ArrowheadService.cs
proj'.
info : Restoring packages for C:\Users\vlads\ArrowheadService\ArrowheadService.csproj...
info :   GET https://api.nuget.org/v3-flatcontainer/arrowhead/index.json
info :   NotFound https://api.nuget.org/v3-flatcontainer/arrowhead/index.json 262ms
error: NU1101: Unable to find package Arrowhead. No packages exist with this id in source(s): C:\Program Files\dotnet\sd
k\7.0.306\Sdks\Microsoft.NET.Sdk.Web\library-packs, nuget.org
error: Package 'Arrowhead' is incompatible with 'all' frameworks in project 'C:\Users\vlads\ArrowheadService\ArrowheadSe
rvice.csproj'.
```

It seems that the error message I encountered indicates that the Arrowhead NuGet package is not available in the configured package sources, and it is not compatible with my project's target frameworks.

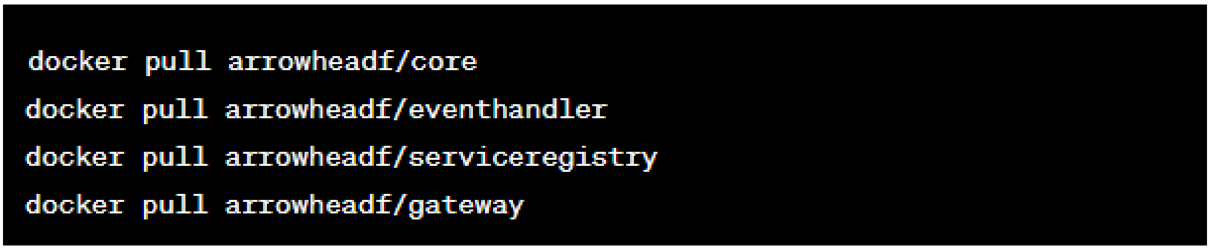
So instead of specifying a specific version, I tried installing the latest version of the Arrowhead NuGet package available at the time I tried the installation and I received the next error:



```
C:\Users\vlads\ArrowheadService>dotnet add package Arrowhead
  Determining projects to restore...
  Writing C:\Users\vlads\AppData\Local\Temp\tmpB23A.tmp
info : X.509 certificate chain validation will use the default trust store selected by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store selected by .NET for timestamping.
info : Adding PackageReference for package 'Arrowhead' into project 'C:\Users\vlads\ArrowheadService\ArrowheadService.cs
proj'.
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/arrowhead/index.json
info :   NotFound https://api.nuget.org/v3/registration5-gz-semver2/arrowhead/index.json 806ms
info :   GET https://api.nuget.org/v3/registration5-gz-semver2/arrowhead/index.json
info :   NotFound https://api.nuget.org/v3/registration5-gz-semver2/arrowhead/index.json 278ms
error: There are no versions available for the package 'Arrowhead'.
```

I gave up and moved on to the next application:

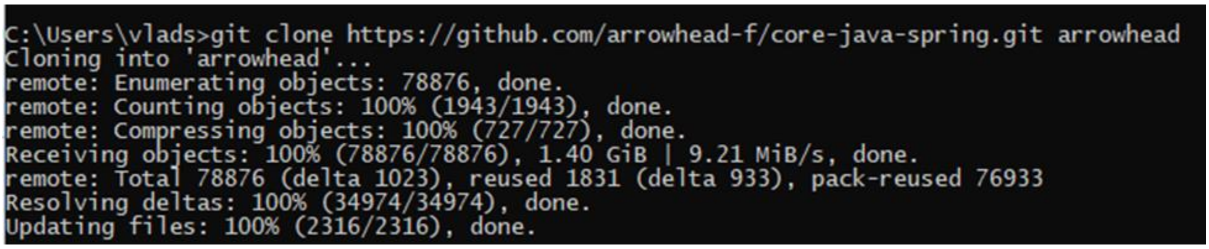
First, I wanted to pull Arrowhead Docker images:



```
docker pull arrowheadf/core
docker pull arrowheadf/eventhandler
docker pull arrowheadf/serviceregistry
docker pull arrowheadf/gateway
```

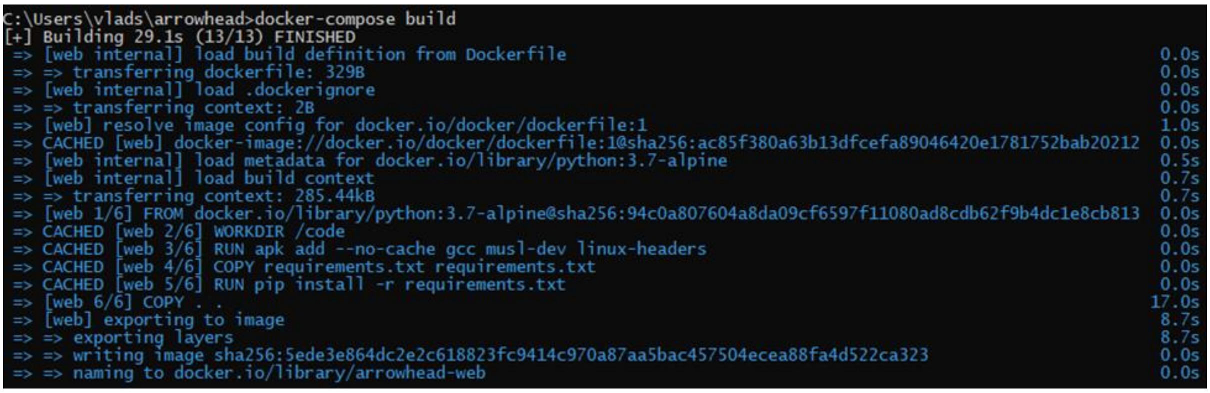
But I encountered errors again, the error message suggests that this repository either does not exist or requires authentication. So I tried to set up Arrowhead using Docker without specific images.

I cloned the official Arrowhead repository from GitHub. Git is required to clone repositories from platforms like GitHub, including the Arrowhead registry. To resolve this issue, I needed to install Git on my system.



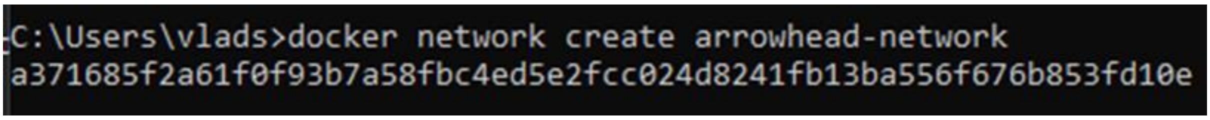
```
C:\Users\vlads>git clone https://github.com/arrowhead-f/core-java-spring.git arrowhead
Cloning into 'arrowhead'...
remote: Enumerating objects: 78876, done.
remote: Counting objects: 100% (1943/1943), done.
remote: Compressing objects: 100% (727/727), done.
Receiving objects: 100% (78876/78876), 1.40 GiB | 9.21 MiB/s, done.
remote: Total 78876 (delta 1023), reused 1831 (delta 933), pack-reused 76933
Resolving deltas: 100% (34974/34974), done.
Updating files: 100% (2316/2316), done.
```

I navigated to the cloned repository directory and build the Arrowhead services using Docker Compose:



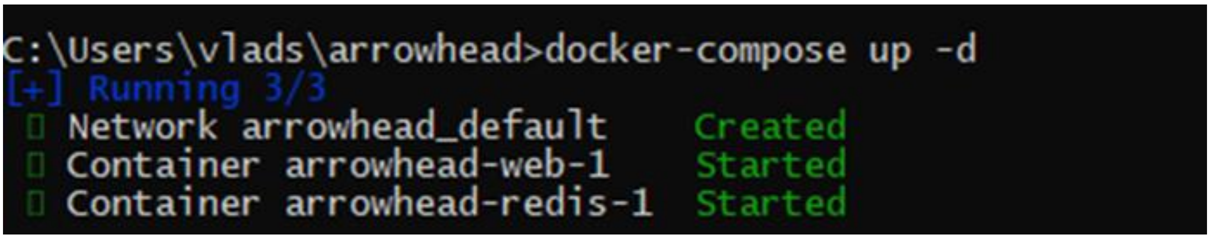
```
C:\Users\vlads\arrowhead>docker-compose build
[+] Building 29.1s (13/13) FINISHED
 => [web internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 329B                                                                               0.0s
 => [web internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                    0.0s
 => [web] resolve image config for docker.io/docker/dockerfile:1                                                   1.0s
 => CACHED [web] docker-image://docker.io/docker/dockerfile:1@sha256:ac85f380a63b13dfcefa89046420e1781752bab20212  0.0s
 => [web internal] load metadata for docker.io/library/python:3.7-alpine                                          0.5s
 => [web internal] load build context                                                                              0.7s
 => => transferring context: 285.44kB                                                                              0.7s
 => [web 1/6] FROM docker.io/library/python:3.7-alpine@sha256:94c0a807604a8da09cf6597f11080ad8cdb62f9b4dc1e8cb813  0.0s
 => CACHED [web 2/6] WORKDIR /code                                                                                 0.0s
 => CACHED [web 3/6] RUN apk add --no-cache gcc musl-dev linux-headers                                             0.0s
 => CACHED [web 4/6] COPY requirements.txt requirements.txt                                                        0.0s
 => CACHED [web 5/6] RUN pip install -r requirements.txt                                                           0.0s
 => [web 6/6] COPY . .                                                                                            17.0s
 => [web] exporting to image                                                                                       8.7s
 => => exporting layers                                                                                            8.7s
 => => writing image sha256:5ede3e864dc2e2c618823fc9414c970a87aa5bac457504ecea88fa4d522ca323                       0.0s
 => => naming to docker.io/library/arrowhead-web                                                                   0.0s
```

I created a Docker network to allow the Arrowhead services to communicate with each other:

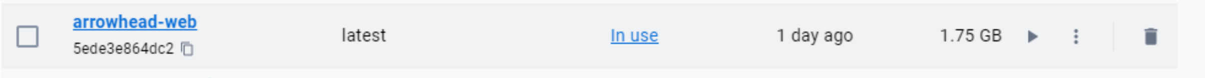


```
C:\Users\vlads>docker network create arrowhead-network
a371685f2a61f0f93b7a58fbc4ed5e2fcc024d8241fb13ba556f676b853fd10e
```

I started the Arrowhead services using Docker Compose within the created network:

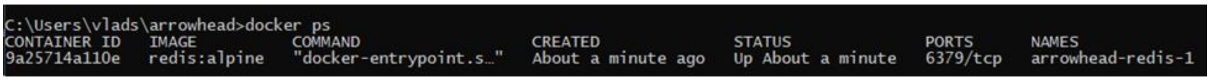


```
C:\Users\vlads\arrowhead>docker-compose up -d
[+] Running 3/3
 ✔ Network arrowhead_default     Created
 ✔ Container arrowhead-web-1     Started
 ✔ Container arrowhead-redis-1   Started
```



```
□  arrowhead-web        latest        In use        1 day ago        1.75 GB
   5ede3e864dc2
```

I verified that the Services are running:

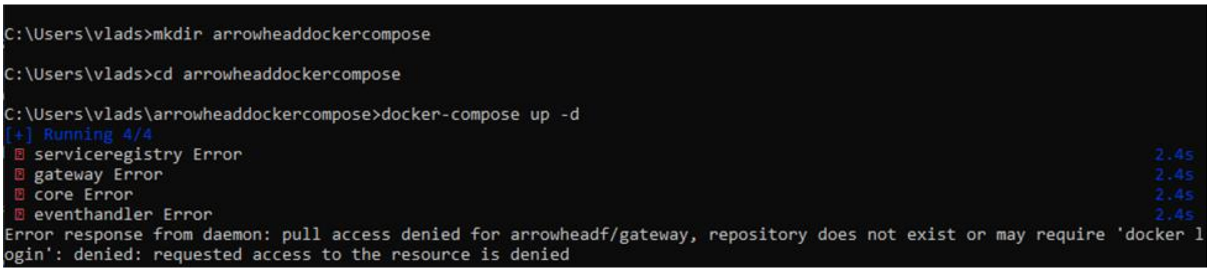


```
C:\Users\vlads\arrowhead>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS      NAMES
9a25714a110e   redis:alpine   "docker-entrypoint.s…"   About a minute ago   Up About a minute   6379/tcp   arrowhead-redis-1
```

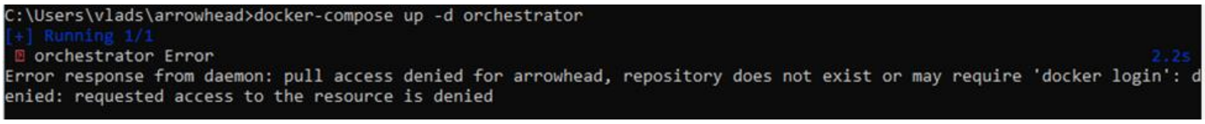
By building the Arrowhead services from the official source code, I avoided any specific image-related issues and had more control over the deployment process.

To access Arrowhead services, I need to make HTTP requests to the endpoints of the individual services.

I wanted to start the Arrowhead Orchestrator, but again I received errors:



```
C:\Users\vlads>mkdir arrowheaddockercompose

C:\Users\vlads>cd arrowheaddockercompose

C:\Users\vlads\arrowheaddockercompose>docker-compose up -d
[+] Running 4/4
 ✘ serviceregistry Error                                                                                  2.4s
 ✘ gateway Error                                                                                          2.4s
 ✘ core Error                                                                                             2.4s
 ✘ eventhandler Error                                                                                     2.4s
Error response from daemon: pull access denied for arrowheadf/gateway, repository does not exist or may require 'docker l
ogin': denied: requested access to the resource is denied
```



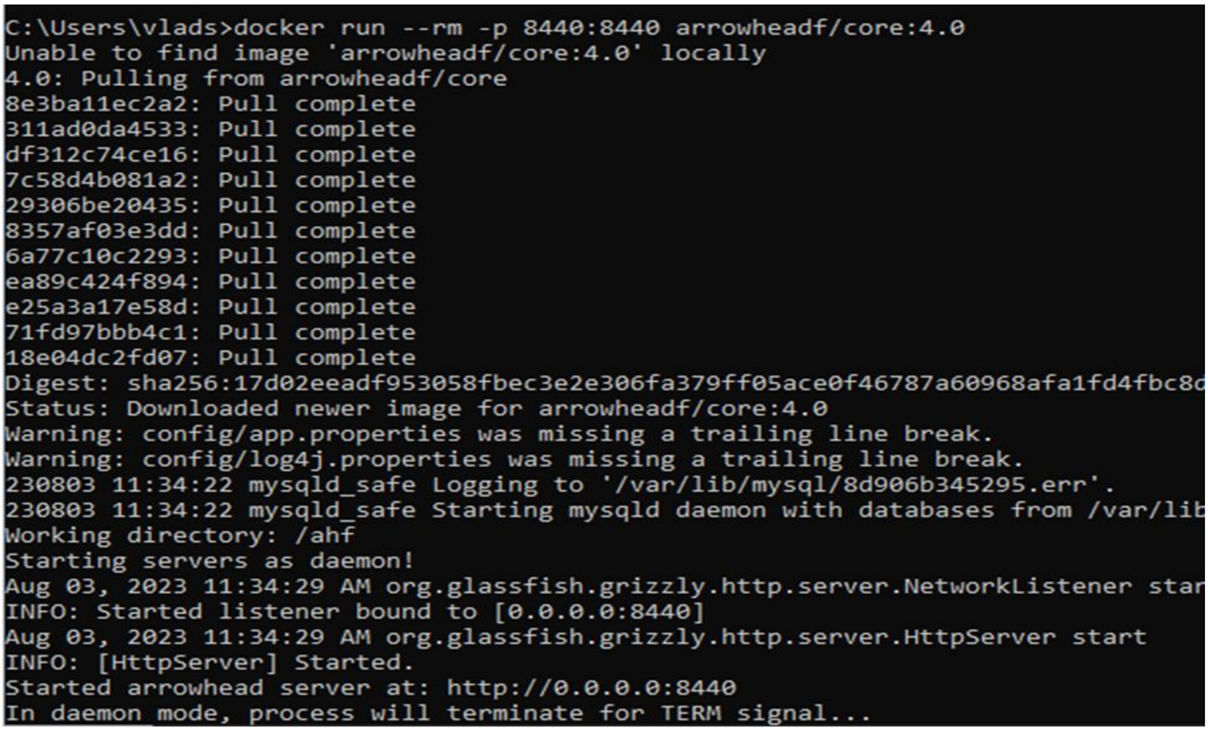
```
C:\Users\vlads\arrowhead>docker-compose up -d orchestrator
[+] Running 1/1
 ✘ orchestrator Error                                                                                      2.2s
Error response from daemon: pull access denied for arrowhead, repository does not exist or may require 'docker login': d
enied: requested access to the resource is denied
```

I managed to access the Service Registry Arrowhead Core System.

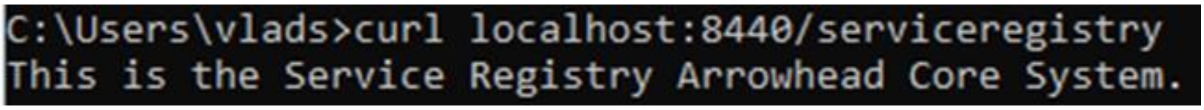
I cloned the Arrowhead Framework Docker repository from GitHub, navigated to the ahf-docker directory, ran the command “docker run --rm -p 8440:8440 arrowheadf/core:4.0” to start the Arrowhead Core 4.0-lw Docker container.

I tested the service by running the command “curl localhost:8440/serviceregistry”. This should respond with "This is the Service Registry Arrowhead Core System".

This allows me to quickly set up and run the Arrowhead Core 4.0-lw Docker container for development purposes without the need for connecting to an existing local cloud.



```
C:\Users\vlads>docker run --rm -p 8440:8440 arrowheadf/core:4.0
Unable to find image 'arrowheadf/core:4.0' locally
4.0: Pulling from arrowheadf/core
8e3ba11ec2a2: Pull complete
311ad0da4533: Pull complete
df312c74ce16: Pull complete
7c58d4b081a2: Pull complete
29306be20435: Pull complete
8357af03e3dd: Pull complete
6a77c10c2293: Pull complete
ea89c424f894: Pull complete
e25a3a17e58d: Pull complete
71fd97bbb4c1: Pull complete
18e04dc2fd07: Pull complete
Digest: sha256:17d02eeadf953058fbec3e2e306fa379ff05ace0f46787a60968afa1fd4fbc8d
Status: Downloaded newer image for arrowheadf/core:4.0
Warning: config/app.properties was missing a trailing line break.
Warning: config/log4j.properties was missing a trailing line break.
230803 11:34:22 mysqld_safe Logging to '/var/lib/mysql/8d906b345295.err'.
230803 11:34:22 mysqld_safe Starting mysqld daemon with databases from /var/lib
Working directory: /ahf
Starting servers as daemon!
Aug 03, 2023 11:34:29 AM org.glassfish.grizzly.http.server.NetworkListener star
INFO: Started listener bound to [0.0.0.0:8440]
Aug 03, 2023 11:34:29 AM org.glassfish.grizzly.http.server.HttpServer start
INFO: [HttpServer] Started.
Started arrowhead server at: http://0.0.0.0:8440
In daemon mode, process will terminate for TERM signal...
```



```
C:\Users\vlads>curl localhost:8440/serviceregistry
This is the Service Registry Arrowhead Core System.
```

Theory:

Arrowhead is an Industrial Internet of Things framework that aims to facilitate the development of interoperable and secure IoT systems for industrial automation and control. It was developed by the Arrowhead Tools project, funded by the European Union. The framework promotes the use of Service-Oriented Architecture (SOA) principles and defines a set of standardized communication protocols and interfaces. Arrowhead enables devices, systems, and services from different manufacturers to work together seamlessly in industrial environments.

Arrowhead focuses on ensuring security, scalability, and interoperability, making it suitable for a wide range of industrial applications. It emphasizes the concept of "System of Systems" (SoS), where multiple interconnected systems collaborate to achieve specific goals.

Docker is a popular open-source platform that allows developers to automate the deployment, scaling, and management of applications inside lightweight, portable containers. Containers provide a consistent environment for applications to run, ensuring that they work consistently across different development and production environments. Docker uses containerization technology to package applications and their dependencies into isolated containers, making them easily deployable and manageable.

Docker has revolutionized application deployment by making it easier to build, ship, and run applications across various platforms. It promotes the use of microservices architecture, which allows applications to be broken down into smaller, modular services that can be developed and deployed independently.

The **curl** command is used to transfer data to or from a server using various supported protocols like HTTP, HTTPS, FTP, FTPS, SCP, SFTP, and more. It is commonly used to download files, make API requests, or test web services directly from the command line.