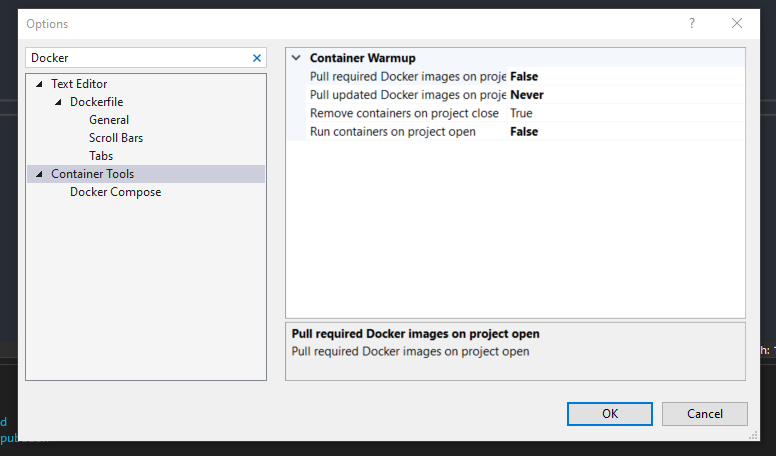
Micro services Architecture

**Notes:-**

**1-once you make clean solution all the docker containers are removed from the docker**

**(to make /allow debugging for attaching docker image)**



You can see that the docker is supported on the Docker Section on launcherSetting.json

{

"iisSettings": {

"windowsAuthentication": false,

"anonymousAuthentication": true,

"iisExpress": {

"applicationUrl": "http://localhost:46448",

"sslPort": 0

}

},

"$schema": "http://json.schemastore.org/launchsettings.json",

"profiles": {

"IIS Express": {

"commandName": "IISExpress",

"launchBrowser": true,

"launchUrl": "swagger",

"environmentVariables": {

"ASPNETCORE\_ENVIRONMENT": "Development"

}

},

"Catalog.API": {

"commandName": "Project",

"launchBrowser": true,

"launchUrl": "swagger",

"environmentVariables": {

"ASPNETCORE\_ENVIRONMENT": "Development"

},

"dotnetRunMessages": "true",

"applicationUrl": "http://localhost:5000"

},

"Docker": {

"commandName": "Docker",

"launchBrowser": true,

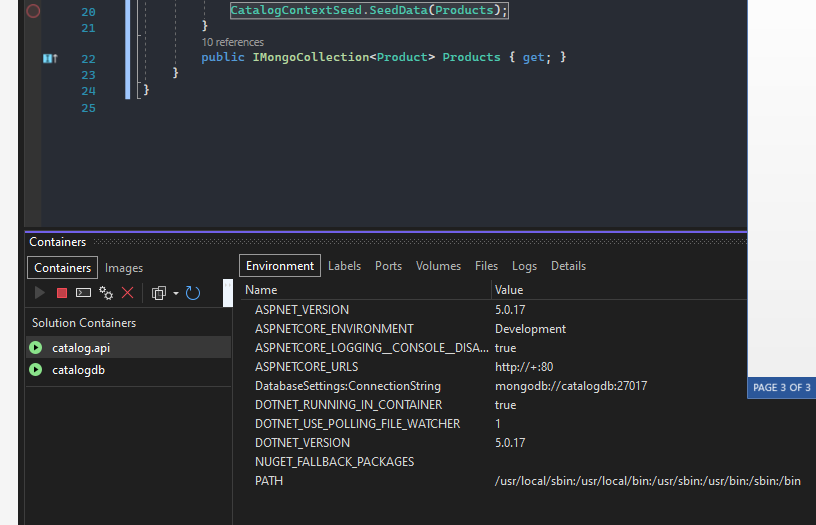
"launchUrl": "{Scheme}://{ServiceHost}:{ServicePort}/swagger",

"publishAllPorts": true

}

}

}

You can see that on window > Containers options you can see the containers generated   


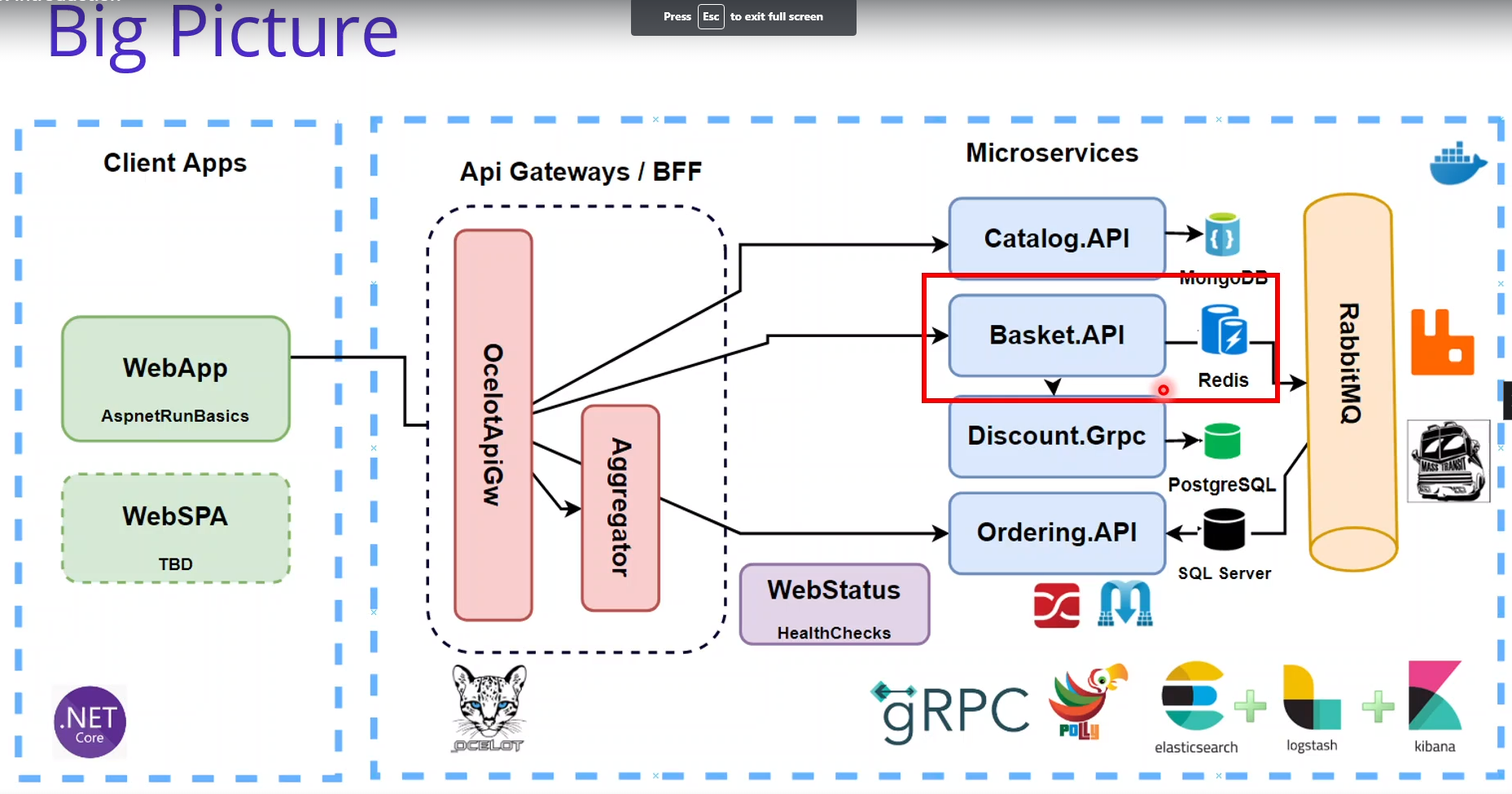
**Installing And Configuring Mongo Client**

Notes:-

1-in order to see mongo data on tool , there is called mongo client which is used as dashboard for mongo datagbase

docker run -d -p 3000:3000 mongoclient/mongoclient

**Basket Microservices**



Redis in Basket Microservices

A-Open Source NOSQL Database

B-Remote Dictionary Server

C-Key-Value Paris

D-Data Structure Server

E-Externaly Fast

F-Save Data Both on Ram And on The Dissk

G-its working on single instance , on RAM you need huge size ram

On Docker Compose > right click > open developer powershell and type the following commands

Docker pull redis

Docier run –d –p 6379:6379 –name aspnetrun-redis redis

**We Will Use Repository Layer Architecture Which is**

**A-Presentaion Layer**

**B-Service Layer (Business Logic Layer)**

**C-Data Access Layer**

Catalog API

SPA

API Gateway

Basket API

Order API

External Micro services Pattern

API Gateway

Resilient Communication

Publisher / Subscriber and Event Driver

**The Reason of using Redis Cache is the benefit of the distributed caching which is speed up the applicaiton**

Contianer Management With Portainer

**Notes:-**

**1-Open Source**

**2-Managing Container-based software application**

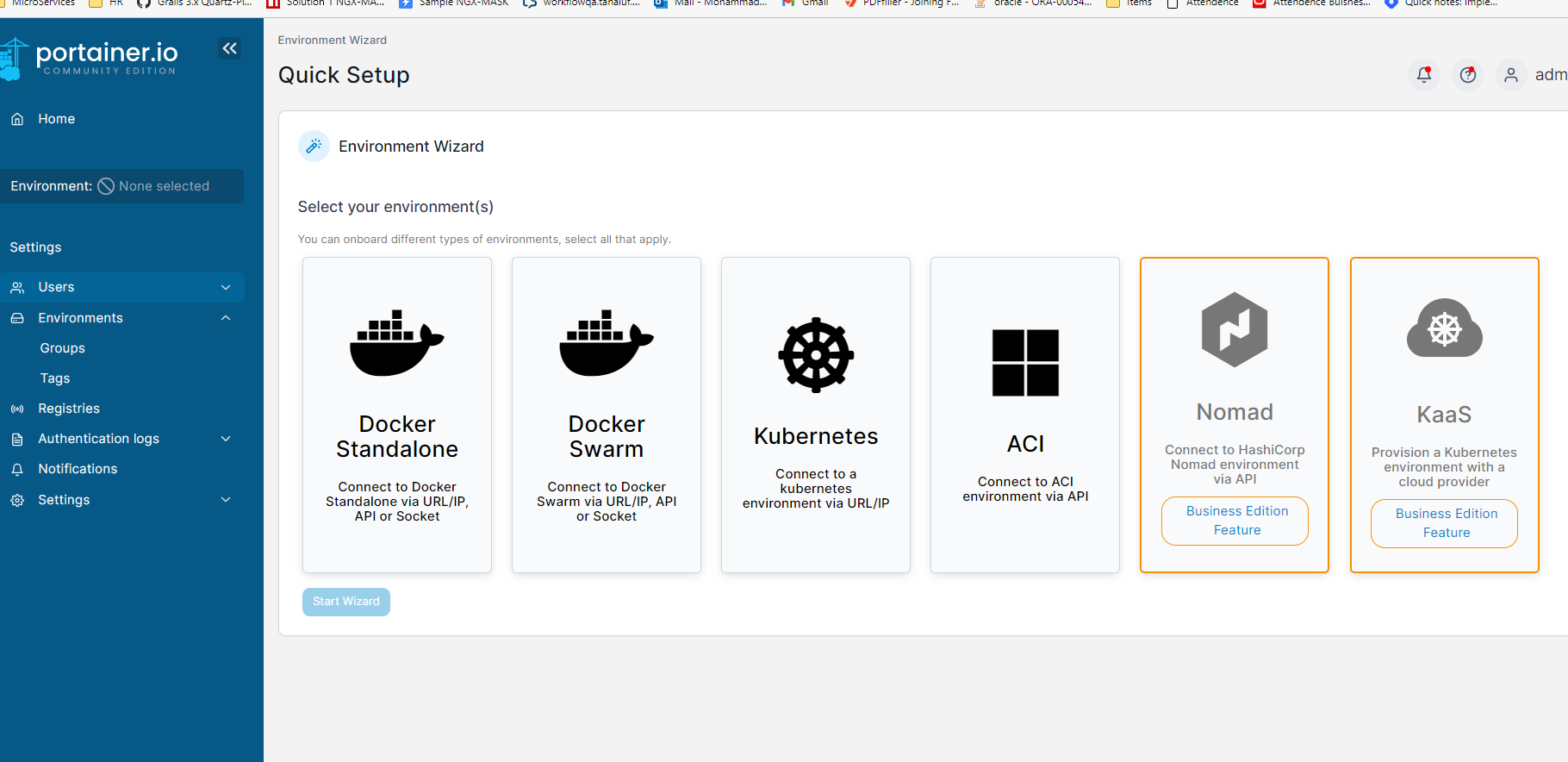
**3-kubernetes , Docker , Docker Swarm , Azure ACI and Edge Enviroemnts**

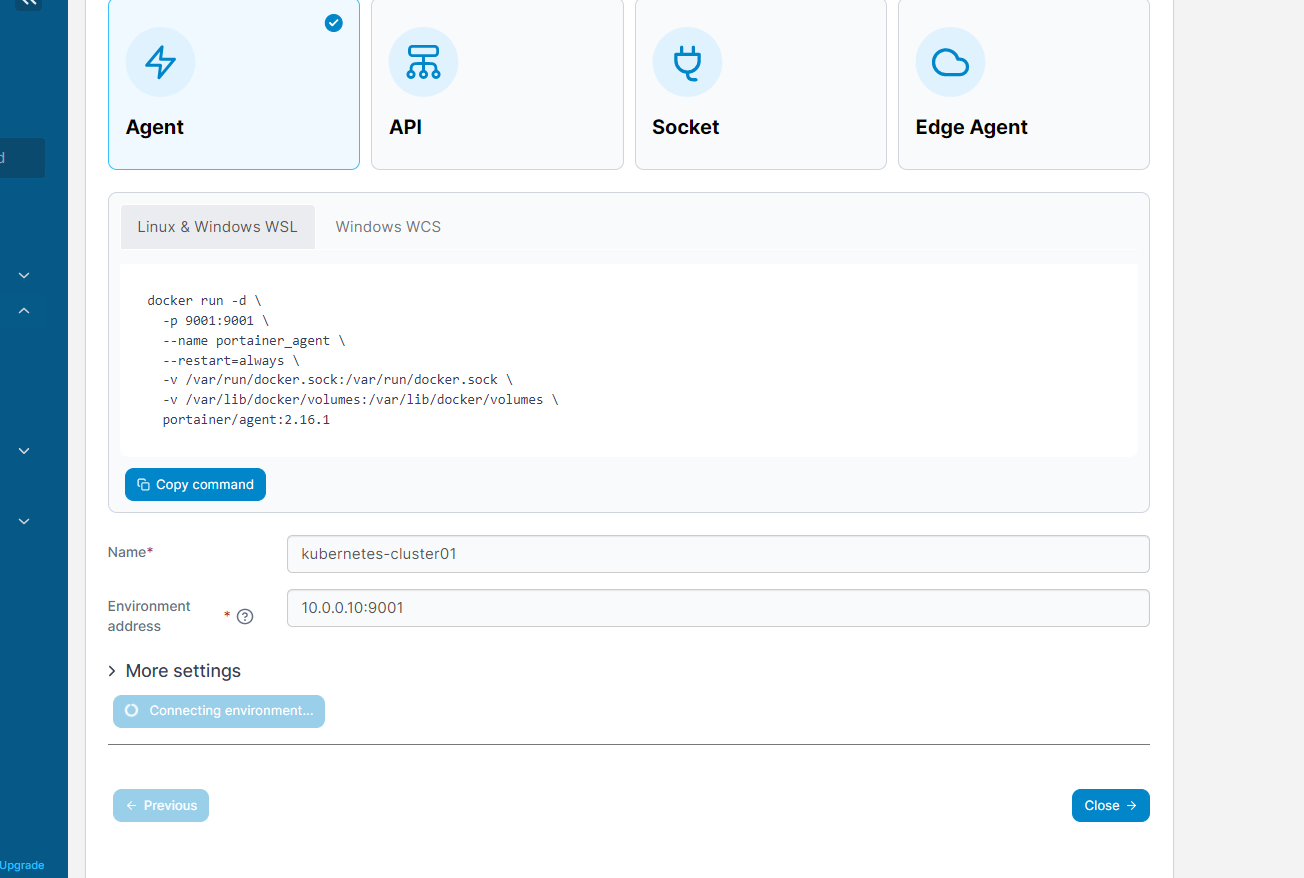
**4-Manage Enviroments , deploy applications , monitor app performance and triage problems**

(with portainer we can manage our container , image , volumes on new easy way)

First time it will not uplaod docker , becouse you have to create user admin with username and password)

Then you have to logout and then login and select the docker standalone





**PostGreess in Discount Microservices**

**Notes:-**

**Open-source relational database**

**Security , Sortability and scalability**

**Reliablity and flexibility**

**Successful and performance**

We will use pgAdmin which tool that enable as dasbhoard for the postgress database

We will using the following packages

Dapper ORM : is an object mapping Nuget Package for query operation to access the tables very fast without need of the Linq of Entity framework (its mixed between ADO.Net and Entity Framework)

Npgsql : is used for the Postgress database