



UNIVERSITÀ DEGLI STUDI DI CATANIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA
CORSO DI LAUREA MAGISTRALE IN INFORMATICA

Nicolò Di Marco
Andrea Filippo Salemi

SpicyLand

RELAZIONE FINALE DI SISTEMI CLOUD & LABORATORIO

Anno Accademico 2023 – 2024

Indice

Indice	2
Introduzione	3
1. Tecnologie usate	4
1.1 - .NET 8	4
1.2 - Entity Framework.	5
1.3 - Docker	6
1.3.1 - Comandi utilizzati su Powershell	7
1.4 - AKS	8
1.4.1 - Comandi utilizzati su Powershell	9
2. Grafico Progettuale	12
3. Immagini Risultato progetto	13
4. Link utili.....	17
5. Conclusioni	17

Introduzione

Nel nostro progetto, per il corso di Sistemi Cloud e Laboratorio, vi illustreremo lo sviluppo e la realizzazione di una web application innovativa per la gestione di un Fast-food. Abbiamo scelto di sviluppare questa soluzione utilizzando le tecnologie più recenti e performanti per garantire scalabilità, affidabilità e facilità d'uso. L'obiettivo principale è stato creare una piattaforma efficiente e user-friendly per la gestione delle operazioni quotidiane di un Fast-food attraverso la raccolta delle ordinazioni dei clienti, e promuovere o mostrare loro delle novità sul nostro modello di sistema Fast-Food. All'interno della nostra web app, vi è anche una sezione dedicata al ristoratore, il quale potrà gestire in autonomia il menù da presentare ai clienti con la possibilità di attivare la funzione "piatto del mese", mostrare e creare delle news e vedere le ordinazioni giornaliere in tempo reale.

1. Tecnologie usate

In questo capitolo verranno descritte le principali tecnologie utilizzate per lo sviluppo del progetto di sistemi cloud. Le tecnologie trattate sono:

- **.NET Core 8**: il framework scelto per lo sviluppo della nostra web app, grazie alla sua robustezza e flessibilità.;
- **Entity Framework**: per la gestione dell'interazione con il database, facilitando le operazioni CRUD;
- **Docker**: per containerizzare le nostre applicazioni, garantendo consistenza e portabilità tra i diversi ambienti di sviluppo e produzione;
- **AKS (Azure Kubernetes Service)**: per orchestrare l'infrastruttura basata su microservizi, garantendo un deployment efficiente e scalabile.

Per ciascuna di esse verranno forniti cenni teorici e un'analisi del loro utilizzo nel progetto.

1.1 - .NET 8

.NET Core 8 è una piattaforma di sviluppo open-source e cross-platform sviluppata da Microsoft, progettata per lo sviluppo di applicazioni moderne, cloud-based ad elevate prestazioni. Una delle principali caratteristiche di .NET Core è la sua modularità e la capacità di eseguire su diversi sistemi operativi, inclusi Windows, MacOS e Linux.

Nel nostro progetto, .NET Core 8 è stato utilizzato come framework principale per lo sviluppo della web application **SpicyLand**, attuando la metodologia **MVC(Model-View-Controller)**.

Il pattern MVC è una metodologia di sviluppo software che separa la nostra web app in tre componenti principali:

- **Model:** gestisce la logica dell'applicazione relativa ai dati e alle regole di business;
- **View:** responsabile della presentazione dei dati all'utente;
- **Controller:** intermediario tra il Model e la View, gestisce le interazioni dell'utente, elabora i dati attraverso il Model e restituisce la View appropriata.

Questa separazione consente una migliore organizzazione del codice, una maggiore facilità di manutenzione e una più chiara distinzione delle responsabilità all'interno dell'applicazione.

Tale metodologia è stata impiegata per strutturare l'applicazione in modo che le responsabilità siano distinte tra loro, facilitando così lo sviluppo, il testing e la manutenzione del codice. Questo approccio ha permesso di sviluppare un'applicazione scalabile e facilmente estensibile per future implementazioni.

1.2 - Entity Framework.

Entity Framework è un ORM (Object-Relational Mapper) sviluppato da Microsoft che permette agli sviluppatori di lavorare con un database utilizzando oggetti .NET, eliminando la necessità di scrivere gran parte del codice SQL manualmente. Questo strumento facilita la gestione dei dati attraverso operazioni CRUD (Create, Read, Update, Delete) su database relazionali.

Entity Framework è stato utilizzato per interagire con il database **SQL Server**; ciò ha permesso di mappare i dati del database a modelli C# utilizzando il paradigma del codice first. L'adozione di Entity Framework ha semplificato notevolmente la gestione dei dati, consentendo di incentrarsi prevalentemente sulla logica di business senza preoccuparsi troppo delle complessità delle query SQL.

SQL Server è un sistema di gestione di database relazionali sviluppato da Microsoft. È noto per le sue capacità di gestione dati sicure, affidabili e performanti ed è utilizzato in molte applicazioni aziendali per memorizzare e gestire grandi quantità di dati.

Nel nostro progetto, abbiamo scelto di utilizzare SQL Server per la sua affidabilità e le sue capacità di gestione dei dati, garantendo un'archiviazione sicura ed efficiente.

1.3 - Docker

Docker è una piattaforma di containerizzazione che permette di automatizzare il deployment di applicazioni all'interno di container leggeri e portabili. I container Docker includono tutto il necessario per eseguire un'applicazione: codice, Runtime, librerie e dipendenze. Questo garantisce che l'applicazione possa essere eseguita in modo consistente nei vari ambienti di sviluppo, testing e produzione.

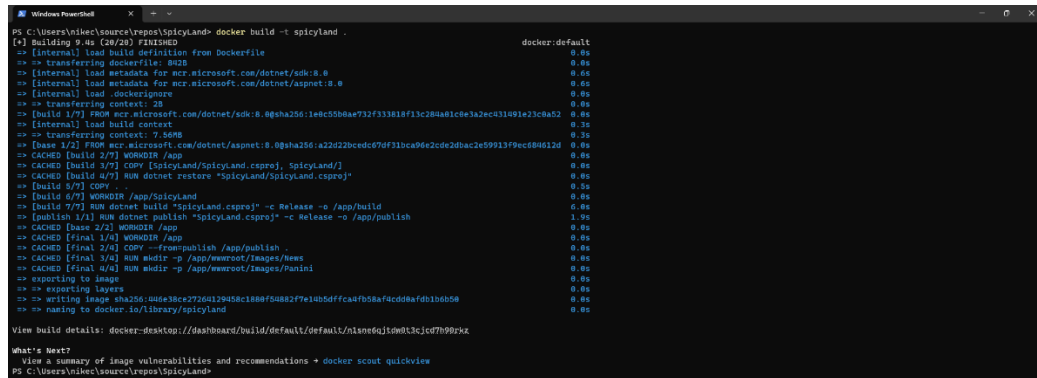
Dunque, abbiamo utilizzato Docker per la creazione di due container, ***spicyland*** e ***dbspicyland***, al fine di testare la comunicazione tra database e web app. Questo ha semplificato il processo di deployment, garantendo che l'applicazione possa essere eseguita in modo uniforme su qualsiasi macchina con Docker installato. La containerizzazione ha anche facilitato il testing e l'isolamento dei vari componenti, migliorando la sicurezza e la scalabilità del sistema.

Successivamente, abbiamo trasformato i container in immagini e rilasciato le varie versioni, all'interno di **Docker-Hub**.

Docker-Hub è un servizio di registry di container che permette di condividere e distribuire immagini Docker; ciò ha facilitato la distribuzione delle nostre immagini containerizzate permettendoci la gestione sul Cluster AKS.

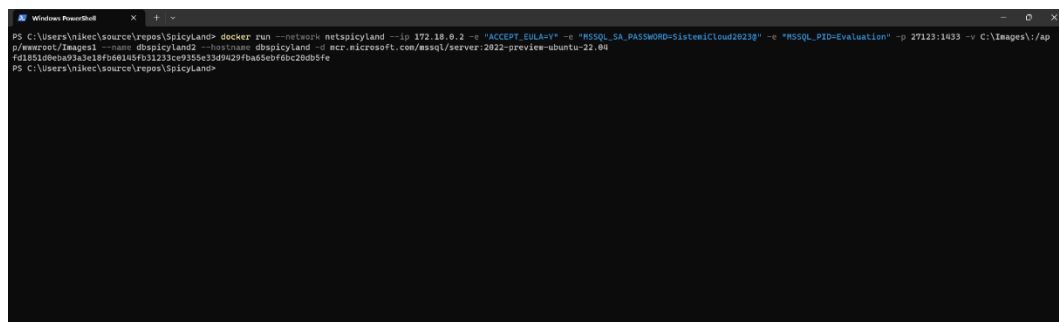
1.3.1 - Comandi utilizzati su Powershell

1. **docker build -t <nome_immagine> .** : è il comando che serve per creare un'immagine Docker a partire da un file Dockerfile presente nella directory corrente (indicata dal punto .);



```
PS C:\Users\nikec\source\repos\SpicyLand> docker build -t spicyland .
[+] Building 9.4s (20/20) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:8.0
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0
=> [internal] load .dockerignore
=> [internal] transfer context: .
=> [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:8.0@sha256:1e4c55b0a732f331818f1c28b0a81c0e3a2ec31401e23c0a52
=> [internal] load build context
=> [internal] transfer context: 7.56MB
=> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:8.0@sha256:a22d22bcdc670f31bca96e2cde2dbac2e59913f9ec080412d
=> CACHED [build 2/7] WORKDIR /app
=> CACHED [build 3/7] COPY [SpicyLand/SpicyLand.csproj, SpicyLand/]
=> CACHED [build 4/7] RUN dotnet restore "SpicyLand/SpicyLand.csproj"
=> [build 5/7] COPY
=> [build 6/7] WORKDIR /app/SpicyLand
=> [build 7/7] RUN dotnet build "SpicyLand.csproj" -c Release -o /app/build
=> [publish 1/1] RUN dotnet publish "SpicyLand.csproj" -c Release -o /app/publish
=> CACHED [base 2/2] WORKDIR /app
=> CACHED [final 1/4] WORKDIR /app
=> CACHED [final 2/4] COPY --from=publish /app/publish
=> CACHED [final 3/4] RUN mkdir -p /app/wwwroot/Images/news
=> CACHED [final 4/4] RUN mkdir -p /app/wwwroot/Images/Panini
=> exporting layers
=> writing image sha256:01d630ce2726d129d58c1880f50827fe14b5dffc4f058efcd0d04fd1b0b0e
=> naming to docker.io/library/spicyland
View build details: docker-desktop://dashboards/build/default/default/150e6e1d0d13cfc7b98rsk
what's next?
View a summary of image vulnerabilities and recommendations + docker scout quickview
PS C:\Users\nikec\source\repos\SpicyLand>
```

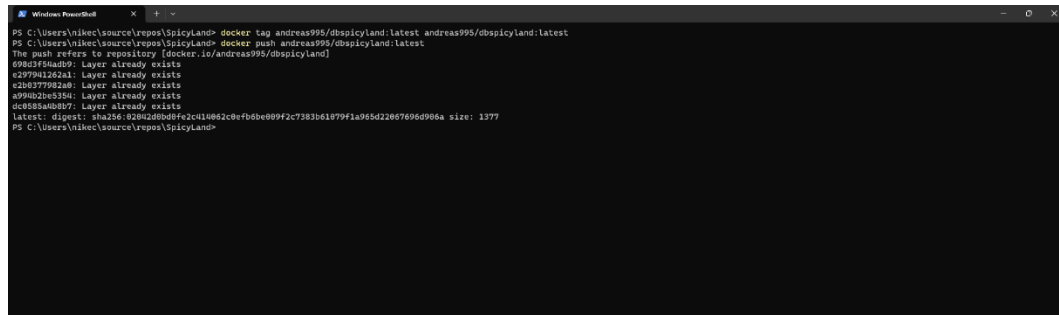
2. **docker run -d -p 8080:80 <nome_immagine>** : è il comando che serve per eseguire il container Docker partendo da un'immagine;
3. **docker run --network netspicyland --ip 172.18.0.2 -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=XXXXXXXXX@" -e "MSSQL_PID=Evaluation" -p 27123:1433 -v C:\Images\:/app/wwwroot/Images1 --name dbspicyland --hostname dbspicyland -d mcr.microsoft.com/mssql/server:2022-preview-ubuntu-22.04**: è il comando che serve per eseguire un container Docker utilizzando un'immagine specifica, configurando la rete netspicyland.



```
PS C:\Users\nikec\source\repos\SpicyLand> docker run --network netspicyland --ip 172.18.0.2 -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=SistemiCloud2023!" -e "MSSQL_PID=Evaluation" -p 27123:1433 -v C:\Images\:/app/wwwroot/Images1 --name dbspicyland2 --hostname dbspicyland -d mcr.microsoft.com/mssql/server:2022-preview-ubuntu-22.04
fd1851d0eba93a3e18f6d145f31233ce935e33d9d29fba55ebf60c26db5fe
PS C:\Users\nikec\source\repos\SpicyLand>
```

4. **docker tag spicyland andreas995/spicyland:latest** : è il comando che serve per creare un nuovo tag (alias) per un'immagine Docker esistente per poterlo infine caricare su Docker-Hub;

5. **docker push nome_utente/nome_repository:latest:** è il comando che serve per eseguire un push della nostra immagine dentro Docker-Hub.



```
PS C:\Users\nikec\source\repos\SpicyLand> docker tag andreas995/dbspicyland:latest andreas995/dbspicyland:latest
PS C:\Users\nikec\source\repos\SpicyLand> docker push andreas995/dbspicyland:latest
The push refers to repository [docker.io/andreas995/dbspicyland]
098d3f54ad09: Layer already exists
c29799d1202a1: Layer already exists
c3b6377982a8: Layer already exists
d99021e4355d: Layer already exists
dc8088a4b0b7: Layer already exists
latest: digest: sha256:02842d0bd0fe2c410861c0efb6be089f2c7383b61079f1a965d22067696d906a size: 1377
PS C:\Users\nikec\source\repos\SpicyLand>
```

1.4 - AKS

Azure Kubernetes Service (AKS) è un servizio gestito di orchestrazione di container fornito da Microsoft Azure basato su Kubernetes. Kubernetes è una piattaforma open-source per l'automazione del deployment, scaling e gestione di applicazioni containerizzate. AKS semplifica la gestione di Kubernetes offrendolo come servizio cloud, riducendo la complessità operativa e migliorando la scalabilità delle applicazioni containerizzate.

AKS è stato utilizzato per controllare i container Docker. Questo ha permesso di distribuire la nostra web app in modo scalabile e affidabile, gestendo automaticamente il bilanciamento del carico, la scalabilità orizzontale e il monitoraggio delle risorse. L'uso di AKS ha semplificato notevolmente la gestione dell'infrastruttura cloud, permettendoci di concentrarci maggiormente sullo sviluppo e l'ottimizzazione della nostra web app. Per interagire con AKS, abbiamo optato l'uso di una connessione tramite PowerShell, in modo da poter gestire i comandi Kubernetes localmente.

Abbiamo notato che, l'uso di PowerShell per interagire con AKS offre dei vantaggi, quali:

- **Automazione dei Processi:** abbiamo creato uno script PowerShell per automatizzare il deployment e la gestione delle applicazioni su AKS, migliorando l'efficienza operativa;
- **Integrazione con l'Infrastruttura Esistente:** PowerShell è stato integrato con i nostri strumenti di gestione e CI/CD (Continuous Integration/Continuous Deployment), per una gestione fluida delle risorse;
- **Facilità di Accesso e Gestione:** la possibilità di eseguire comandi Kubernetes localmente ha reso più agevole la gestione dei cluster, senza dover accedere continuamente alla console web di Azure.

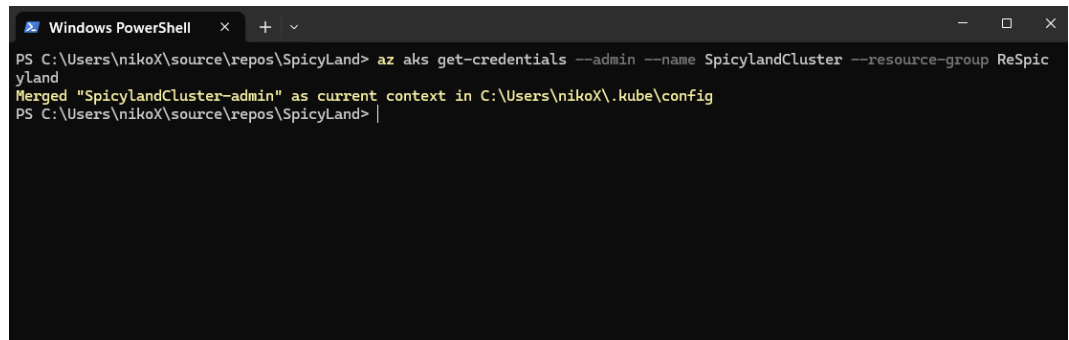
1.4.1 - Comandi utilizzati su Powershell

Di seguito i comandi utilizzati per la gestione del nostro Cluster dentro AKS:

1. **az login:** è il comando per l'autenticazione ad Azure mediante PowerShell;
2. **az aks create --resource-group ReSpicyland --name SpicylandCluster --node-count 2 --node-vm-size "Standard_B2s" --enable-addons monitoring --generate-ssh-keys:** è il comando utilizzato per la creazione del nostro Cluster (denominato SpicylandCluster) associato ad un gruppo di risorse denominato ReSpicyland.

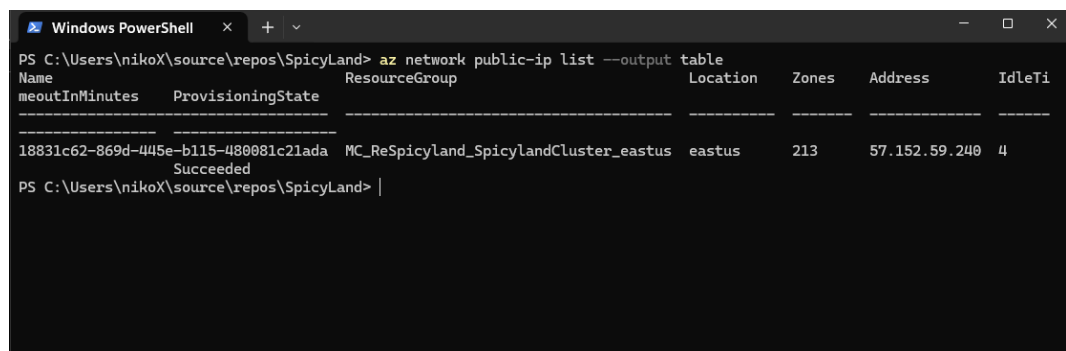
L'add-on **monitoring**, Abilita l'add-on di monitoraggio per il cluster, che consente di utilizzare Azure Monitor per raccogliere e visualizzare metriche e log dal cluster;

3. **az aks get-credentials --admin --name SpicylandCluster --resource-group ReSpicyland**: è il comando utilizzato per poter entrare dentro il cluster SpicylandCluster. Tale comando effettua un merge tra la config Kubernetes locale, con la config Kubernetes all'interno del Cluster;



```
Windows PowerShell
PS C:\Users\nikoX\source\repos\SpicyLand> az aks get-credentials --admin --name SpicylandCluster --resource-group ReSpicyland
Merged "SpicylandCluster-admin" as current context in C:\Users\nikoX\.kube\config
PS C:\Users\nikoX\source\repos\SpicyLand>
```

4. **az aks show --resource-group ReSpicyland --name SpicylandCluster --query "addonProfiles.httpApplicationRouting.config.HTTPApplicationRoutingZoneName" -o table**: tale comando serve per ottenere informazioni specifiche sul cluster AKS SpicylandCluster nel gruppo di risorse ReSpicyland, in particolare l'URL della zona DNS creata dall'add-on HTTP Application Routing. L'add-on HTTP Application Routing configura automaticamente una zona DNS e crea record DNS per i servizi ingress nel cluster.
5. **az network public-ip list --output table**: è il comando utilizzato per elencare tutti gli indirizzi IP pubblici all'interno di una determinata sottoscrizione di Azure.



```
Windows PowerShell
PS C:\Users\nikoX\source\repos\SpicyLand> az network public-ip list --output table
Name                               ResourceGroup      Location    Zones    Address          IdleTi
-----
18831c62-869d-445e-b115-480081c21ada MC_ReSpicyland_SpicyLandCluster_eastus eastus      213      57.152.59.240    4
PS C:\Users\nikoX\source\repos\SpicyLand>
```

6. **kubectl apply -f <nome file yaml>**: è il comando utilizzato per applicare configurazioni Kubernetes da un file YAML. Questo file YAML definisce le risorse che si vogliono creare o aggiornare nel cluster Kubernetes, come: pod, deployment, servizi, ingressi, secret etc.;

7. **kubectl get pods:** è il comando utilizzato per elencare i pod in un cluster Kubernetes. Contengono uno o più container, che riguardano le applicazioni effettivamente eseguite;

```
Windows PowerShell
PS C:\Users\nikoX\source\repos\SpicyLand> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
dbspicyland-f649c9-27x9m            0/1     Evicted   0           2d2h
dbspicyland-f649c9-2xczp            0/1     Evicted   0           2d2h
dbspicyland-f649c9-5nxsj            0/1     Evicted   0           2d2h
dbspicyland-f649c9-8ngxr            0/1     Evicted   0           2d2h
dbspicyland-f649c9-bbwxm            0/1     Evicted   0           2d2h
dbspicyland-f649c9-cvqj9            0/1     Evicted   0           2d2h
dbspicyland-f649c9-jxv7q            0/1     Evicted   0           2d2h
dbspicyland-f649c9-kgzj8            1/1     Running   0           2d2h
dbspicyland-f649c9-m4sq6            0/1     Evicted   0           2d2h
dbspicyland-f649c9-n24sc            0/1     ContainerStatusUnknown 1           2d2h
dbspicyland-f649c9-trf8q            0/1     Evicted   0           2d2h
dbspicyland-f649c9-zkq2w            0/1     Evicted   0           2d2h
spicyland-56766f8bf4-wvdmf         1/1     Running   0           2d2h
PS C:\Users\nikoX\source\repos\SpicyLand>
```

8. **kubectl get svc:** è il comando che viene utilizzato per elencare i servizi (services) in un cluster Kubernetes. Tali servizi sono risorse che forniscono un punto di accesso per un insieme di pod, consentendo la scoperta e il bilanciamento del carico;

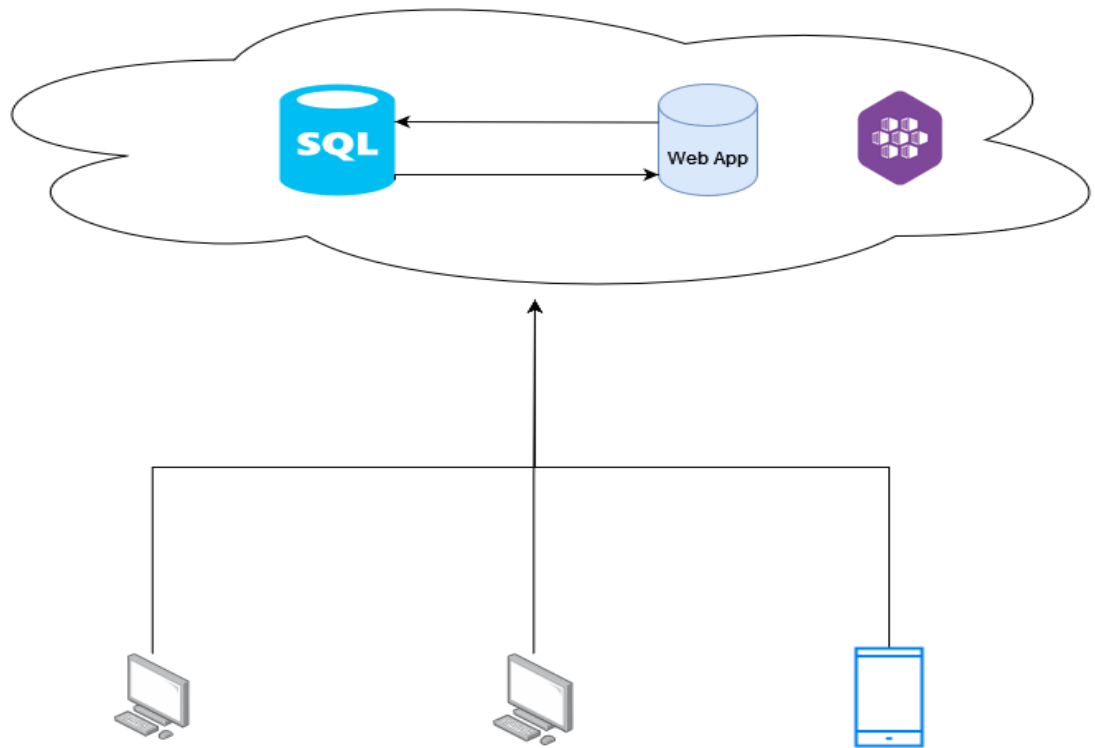
```
Windows PowerShell
PS C:\Users\nikoX\source\repos\SpicyLand> kubectl get svc
NAME      TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
dbspicyland LoadBalancer  10.0.98.66    57.152.59.240 1433:30600/TCP   2d2h
kubernetes ClusterIP      10.0.0.1      <none>         443/TCP          2d2h
spicyland LoadBalancer  10.0.20.157   57.152.59.240 80:32209/TCP     2d2h
PS C:\Users\nikoX\source\repos\SpicyLand>
```

9. **kubectl deployments:** è il comando utilizzato per elencare tutti i deployment nel namespace corrente di un cluster Kubernetes. I deployment in Kubernetes sono risorse che gestiscono il rilascio di applicazioni, consentendo di dichiarare lo stato desiderato per le applicazioni, eseguire aggiornamenti graduali, roll back e ridimensionamento;

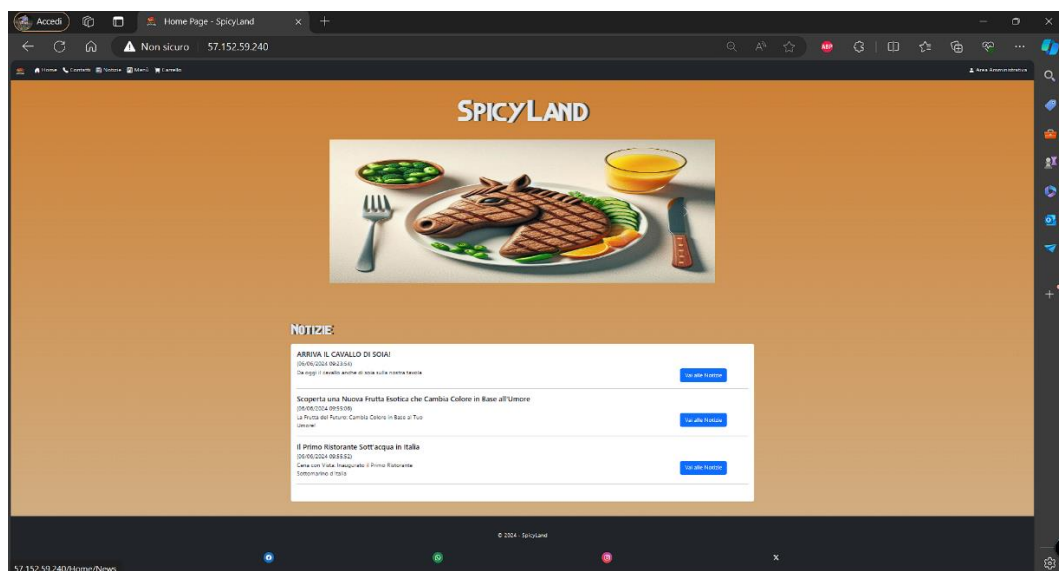
```
Windows PowerShell
PS C:\Users\nikoX\source\repos\SpicyLand> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
dbspicyland 1/1      1             1           2d2h
spicyland   1/1      1             1           2d2h
PS C:\Users\nikoX\source\repos\SpicyLand>
```

10. **kubectl logs <nome-pod>:** è il comando utilizzato per ottenere una visibilità chiara e dettagliata sul comportamento del pod specificato, facilitando la diagnosi e la risoluzione dei problemi.

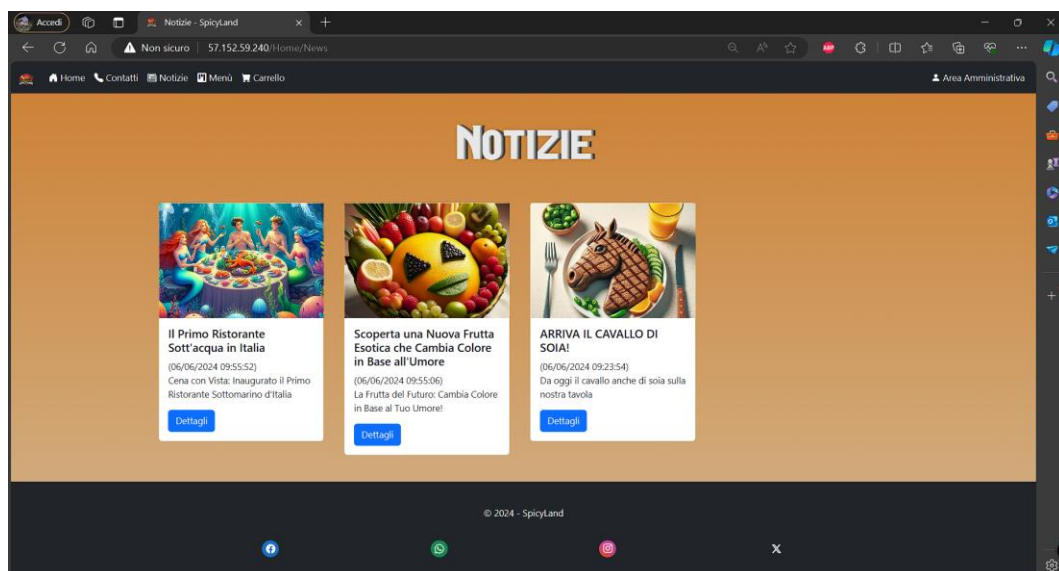
2. Grafico Progettuale



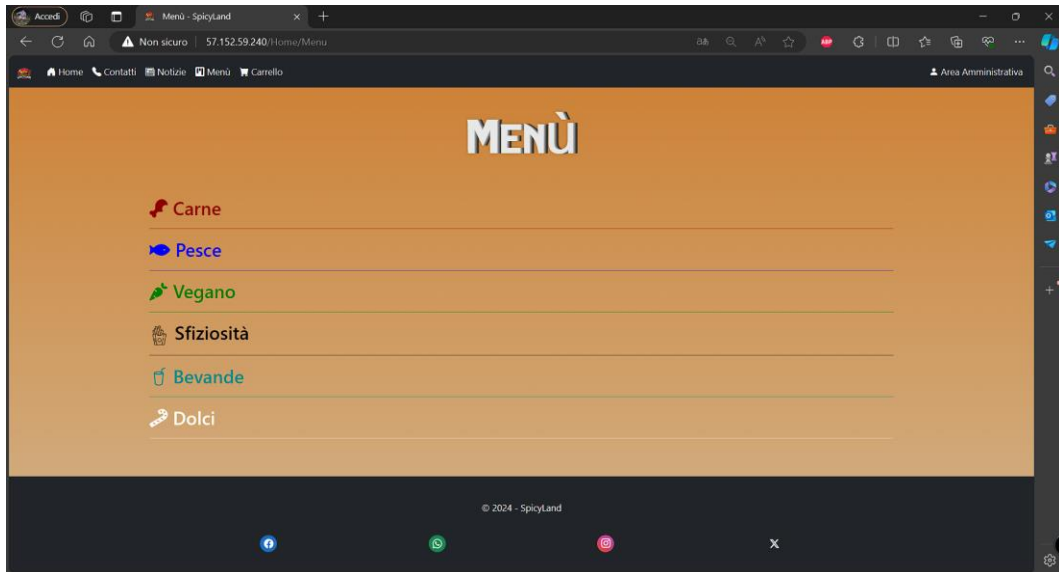
3. Immagini Risultato progetto



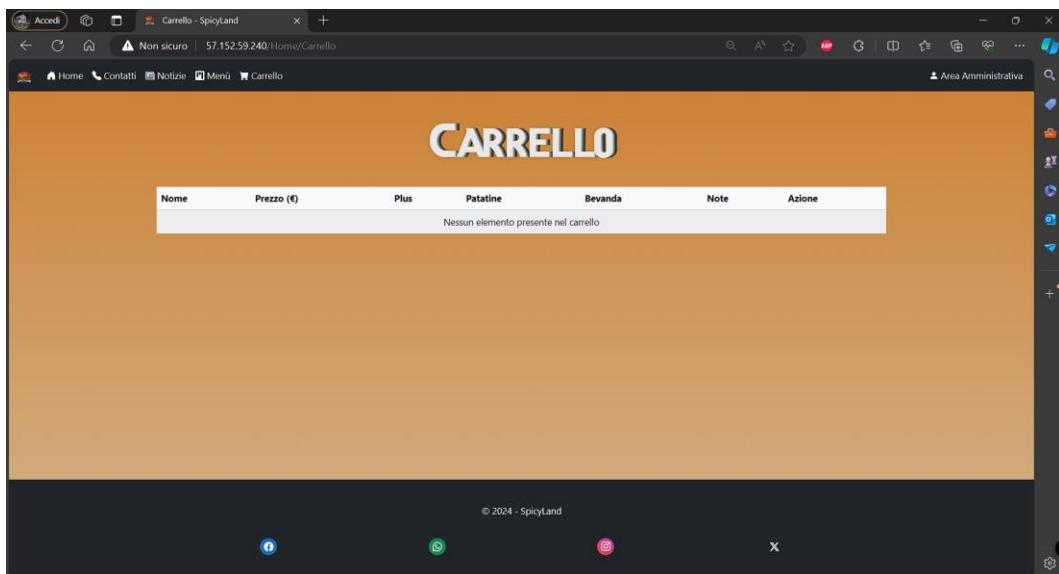
[Home Page]



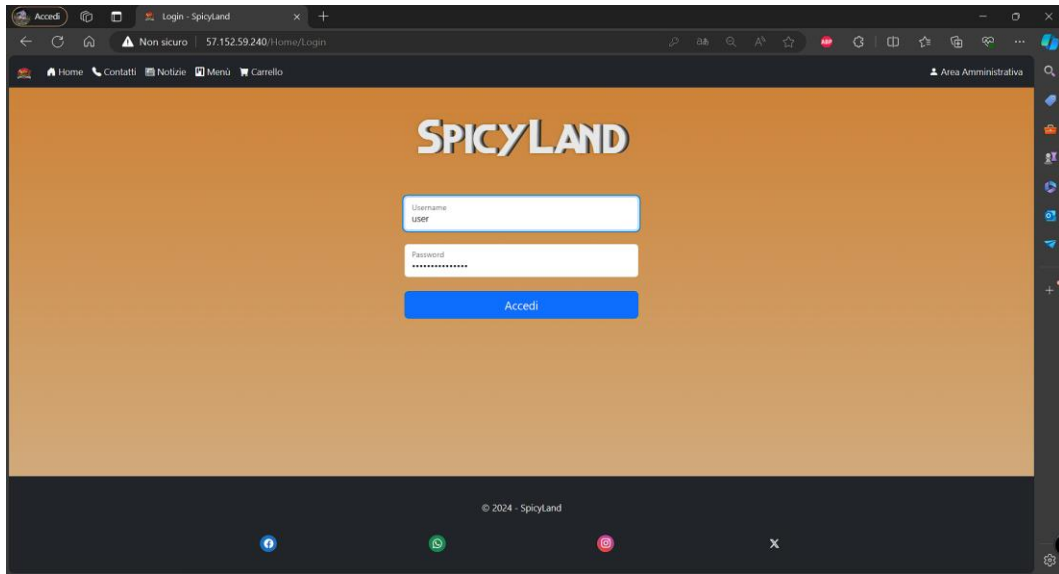
[Notizie]



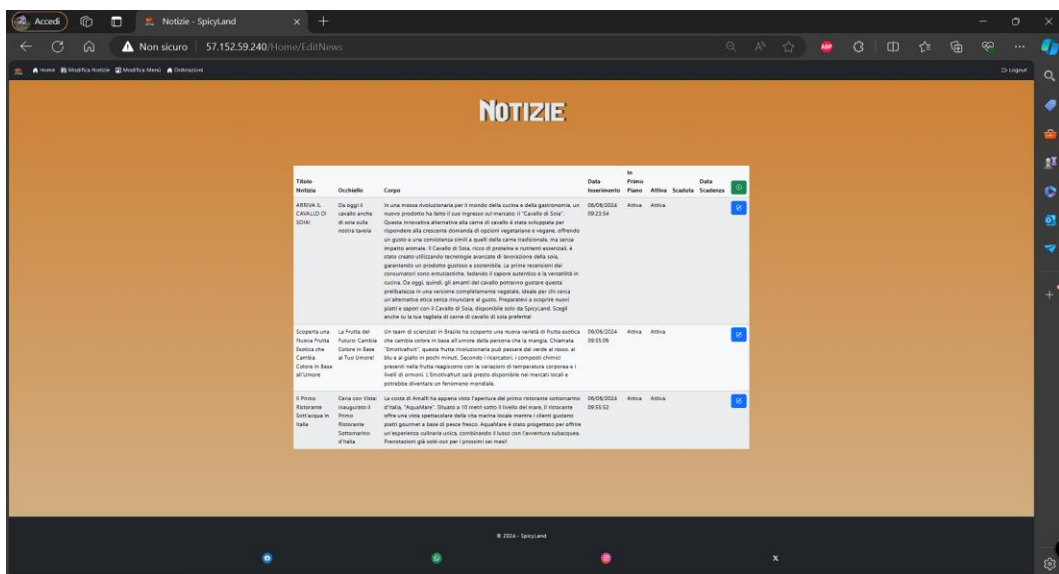
[Menù]



[Carrello]



[Login Area Amministrativa]



[Modifica Notizie]

Nome	Descrizione	Prezzo (€)	Piatto del Mese	In Menù	Data di Creazione	
Carne di Cavallo di Soia	Carne di Cavallo di Soia	€ 22.00	Piatto del mese	In menù	06/06/2024 08:48:44	
Cotoletta di pesce	Cotoletta di pesce	€ 15.00		In menù	06/06/2024 09:39:27	
Orata Arrosto	Orata Arrosto	€ 12.00		In menù	06/06/2024 09:41:25	
Gelato Artigianale	Gelato artigianale in cono	€ 2.30		In menù	06/06/2024 09:42:10	
Raviola Di Ricotta	Raviola di ricotta al forno con ricotta freschissima	€ 2.00		In menù	06/06/2024 09:42:51	
Carne di Cavallo originale	Quella vera e non vegana.	€ 5.00		In menù	06/06/2024 09:43:29	
Coca-Cola	Coca-cola da 33cl	€ 1.50		In menù	06/06/2024 09:43:59	
Patatine	Patatine classiche fritte	€ 1.50		In menù	06/06/2024 09:44:22	
Vertigo	Patatine twist	€ 3.00		In menù	06/06/2024 09:44:53	
Nachos	Patatine nachos messicane	€ 2.00		In menù	06/06/2024 09:45:27	

[Modifica Menù]

Cliente	Ordine n°.	Ordine	Quantità	Note	Status	Azione
Nessun ordinazione presente...						

© 2024 - SpicyLand

[Ordinazioni]

4. Link utili

1. [Home Page - SpicyLand](#)
2. <https://github.com/solutionsbytegenius/SpicyLand.git>

5. Conclusioni

Questo progetto è stato pensato per le piccole imprese di fast-food con l'obiettivo di facilitare il lavoro eliminando la gestione manuale degli ordini. In questo studio, quindi, abbiamo deciso di sviluppare una sola replica per il deployment. Tuttavia, è possibile aumentare il numero di repliche utilizzando il comando:

kubectl scale deployment <nome deployment> --replicas=<numero di repliche>

In alternativa, si può modificare il file ***KuberSpicyland.yaml***, impostando il numero desiderato di repliche nella sezione **replicas** all'interno del deployment.

In conclusione, il progetto risponde alle necessità immediate dei piccoli fast-food e offre anche una soluzione sostenibile e scalabile per il futuro. La facilità di gestione degli ordini e la possibilità di scalare il sistema secondo le esigenze rappresentano un valore aggiunto significativo, contribuendo a migliorare la soddisfazione dei clienti e la competitività del business. Con un'infrastruttura tecnologica robusta e flessibile, i gestori possono concentrarsi maggiormente sulla qualità del servizio e sull'innovazione, garantendo così un'esperienza positiva per i loro clienti.