



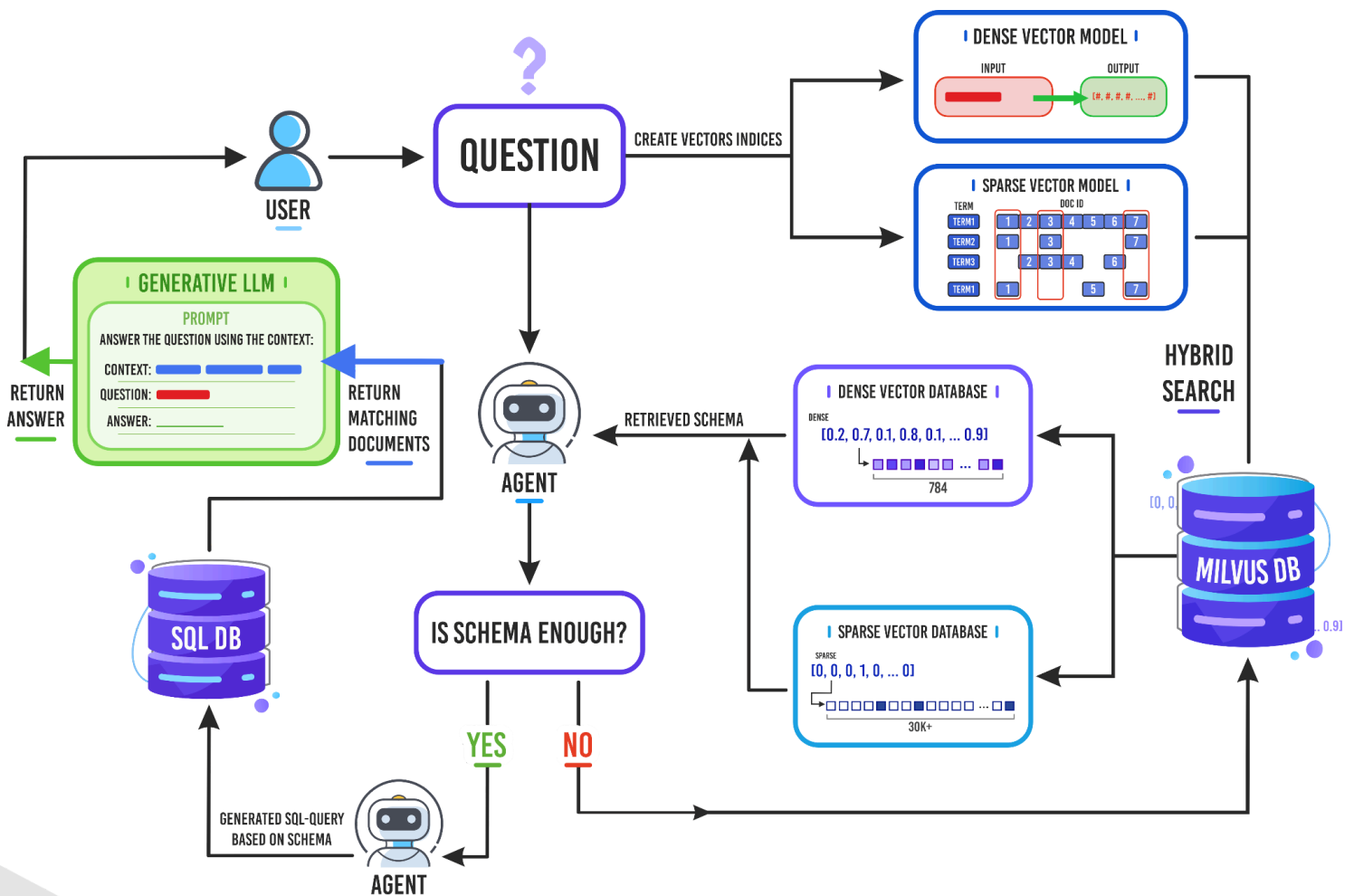
DOKUMENTACJA ARCHITEKTURY ROZWIĄZANIA W AZURE

● Opis rozwiązania

To rozwiązanie przedstawia architekturę aplikacji AI Chatbot w chmurze Azure. Aplikacja składa się z następujących komponentów:

- Aplikacja frontendowa Next.js
- Aplikacja backendowa FastAPI
- Baza danych Milvus
- Usługa etcd do przechowywania konfiguracji
- Usługa MinIO do przechowywania danych

Komponenty te są wdrażane w Azure przy użyciu usług Azure App Service dla aplikacji Next.js i FastAPI oraz Azure Container Instances dla Milvus, etcd i MinIO.



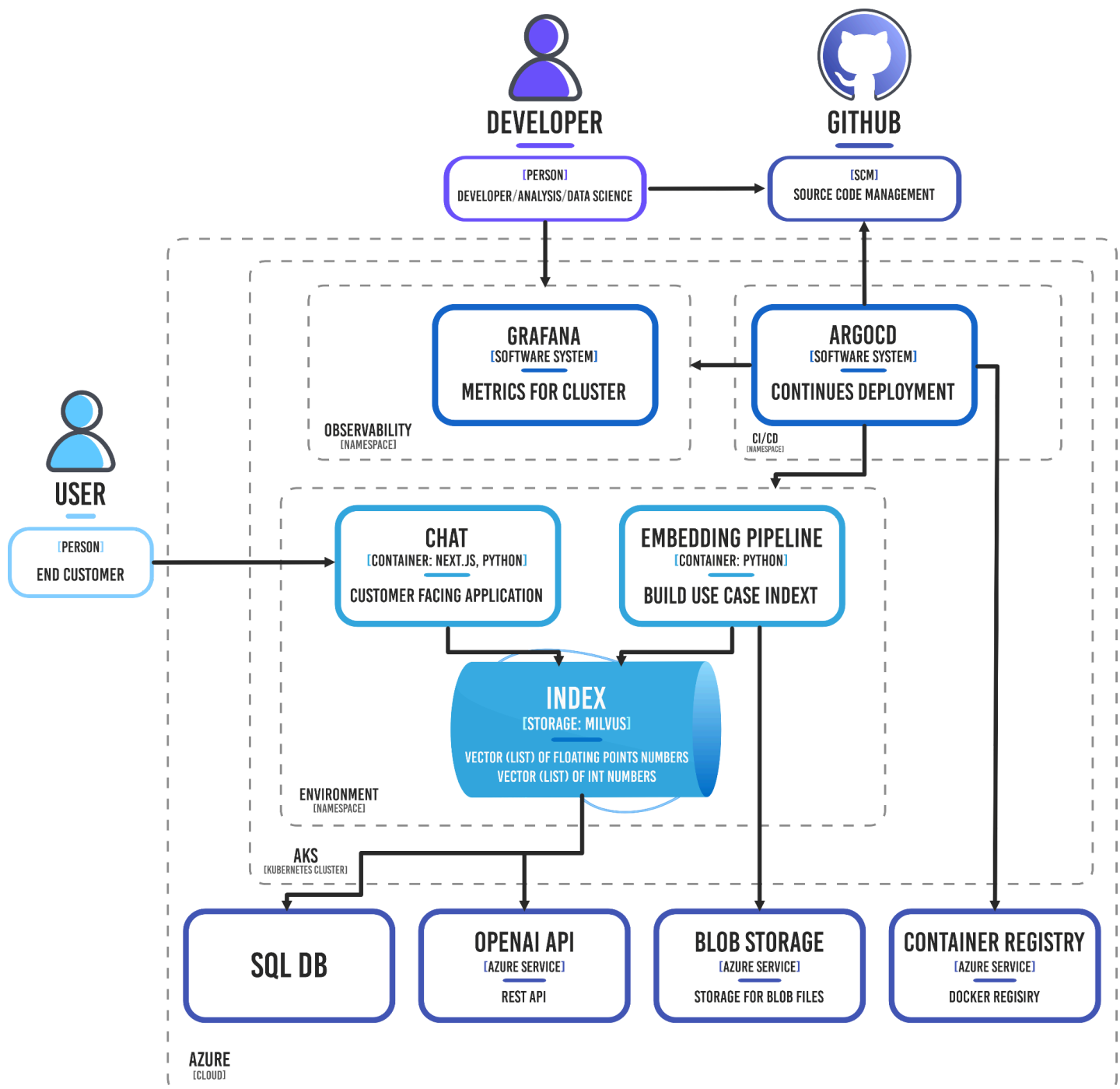
Dane są przechowywane w Azure File Shares, co zapewnia trwałość danych.

1. **Interakcja użytkownika:** Użytkownik wprowadza pytanie, które jest przekazywane do systemu przez interfejs użytkownika oparty na technologii Next.js.
2. **Agent i indeksowanie wektorowe:** Agent (komponent zarządzający zapytaniami) generuje poprawione zapytanie, które następnie przetwarzane jest na wektory pozwalając na znalezienie najbardziej dopasowanej tabeli zaindeksowanej w wektorowej bazie danych.
3. **Ocena adekwatności schematu:** Agent ocenia, czy odnaleziony schemat jest wystarczający do wygenerowania odpowiedzi na pytanie użytkownika. Jeśli tak to korzystając z silnika wyszukiwania hybrydowego odnajduje pasujące schematy oraz ich relacje z docelowej bazy danych.
4. **Dostęp do bazy danych Milvus:** W przypadku, gdy schemat nie jest kompletny lub jest nie wystarczający (co do zapytania użytkownika), Agent przeprowadza dalsze zapytania do bazy Milvus, która efektywnie zarządza zarówno gęstymi jak i rzadkimi wektorami (semantyczne oraz oparte na słowach kluczowych wektory).
5. **Generowanie zapytania SQL:** Jeżeli schemat jest wystarczający, na jego podstawie generowane jest zapytanie SQL, które jest następnie wykonywane w bazie danych SQL w celu uzyskania konkretnej odpowiedzi. Jeśli wystąpi błąd agent ponowi próbę z wykorzystaniem wewnętrznej krytyki ReAct*.
6. **Hybrydowe przeszukiwanie:** Za pomocą hybrydowych technik wyszukiwania, takich jak wektorów gęstych i rzadkich wektorów, system wyszukuje najbardziej odpowiednie dane do formułowania *sql-query*/odpowiedzi.
7. **Formułowanie odpowiedzi:** Po uzyskaniu wymaganych danych, silnik AI (Generative LLM) formułuje odpowiedź, która jest następnie zwracana użytkownikowi.

*Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, & Yuan Cao. (2023). *ReAct: Synergizing Reasoning and Acting in Language Models*.

• Architektura

Poniżej przedstawiamy architekturę rozwiązania zaimplementowanego w środowisku chmurowym Azure, zaprojektowanego z myślą o zapewnieniu efektywnej i skalowalnej obsługi zapytań użytkowników poprzez interfejs czatu.



Interfejs Użytkownika

Aplikacja Czatowa - stanowi front-end systemu, zapewniając interaktywny interfejs dla użytkowników końcowych. Rozwijana z wykorzystaniem technologii Next.js i Python, aplikacja działa w kontenerach w ramach Azure Kubernetes Service (AKS), co umożliwia elastyczne zarządzanie zasobami i skalowanie.

Środowisko Deweloperskie i Narzędzia

Grafana - platforma monitorowania dostarcza kluczowe metryki dotyczących wydajności klastra, co jest istotne dla ciągłego doskonalenia systemu i szybkiego identyfikowania ewentualnych problemów.

ArgoCD - system wdrażania ciągłego integruje się bezpośrednio z GitHubem, umożliwiając automatyzację procesu aktualizacji i wdrożeń aplikacji w środowisku produkcyjnym.

GitHub - jako centralny system zarządzania kodem źródłowym, GitHub jest źródłem dla wszystkich zmian w kodzie, które następnie są automatycznie wdrażane do produkcji.

Proces Przetwarzania Danych

Pipeline Embeddingów - ten komponent jest odpowiedzialny za opracowanie i utrzymanie modeli wektorowych (zarówno *sparse* jak i *dense*), co pozwala na proste generowanie indeksów oraz bezproblemową reindeksację w przypadku pojawienia się nowych danych (np. nowe tabele w bazie).

Indeksacja Danych - z wykorzystaniem Milvus, system oferuje przechowywanie wektorów obu rodzajów wektorów, które umożliwiają szybkie przeszukiwanie danych i odnajdywanie najbardziej adekwatnych informacji.

Komponenty Infrastrukturalne

Azure Kubernetes Service (AKS) - hostuje kluczowe komponenty systemu, w tym aplikację czatową i mechanizm osadzeń, dostarczając niezbędne zasoby obliczeniowe oraz umożliwiając efektywne skalowanie.

Baza Danych SQL - zarządzana przez Azure, baza danych SQL, pozwala na dostęp do zbioru produktów wystawione przez klienta, skonfigurowana dla Agenta na read-only.

OpenAI API - dostępne poprzez Azure, API OpenAI umożliwia integrację zaawansowanych funkcji AI do przetwarzania i analizy zapytań użytkownika.

Blob Storage - służy jako magazyn dla dużych plików oraz metadanych, zarządzany przez Azure Blob Storage, zapewnia bezpieczny i trwały storage dla danych systemu.

Rejestr Kontenerów - Azure Container Registry przechowuje wszystkie obrazy Docker używane w systemie, wspierając ich wersjonowanie i dostępność.

Uwaga

Ostatecznie nasza aplikacja została wystawiona na VM z wykorzystaniem serwisu Azure. - może plan był inny ale najważniejsze że w końcu się udało :D. Ruch to VM kierowany przy użyciu load balancera, z portu 80=> 3000. Maszynie wirtualnej usunęliśmy IP publiczne i VM przyjmuje do siebie jedynie ruch z load balancera przy użyciu taga balancera w Azure

Uruchomienie aplikacji działa poprzez zalogowanie się przez SSH do maszyny wirtualnej na platformie azure, a następnie poprzez przejście do repozytorium uruchamiamy komendę docker compose up.

Bezpieczeństwo

Plan minimalny:

- W podstawowej konfiguracji bezpieczeństwa kluczową koncepcją jest segmentacja sieci. Elementy architektury, które są niezbędne do ekspozycji na świat zewnętrzny, będą dostępne publicznie, jednak większość komunikacji międzyserwisowej powinna być przeprowadzana w obrębie prywatnej sieci wirtualnej (VNet). Takie podejście minimalizuje powierzchnię ataku i chroni wewnętrzne zasoby przed nieautoryzowanym dostępem.
- Sekrety i certyfikaty, kluczowe dla uwierzytelniania i szyfrowania, będą zarządzane i przechowywane z wykorzystaniem usługi Azure Key Vault, co zapewnia ich bezpieczeństwo oraz umożliwia centralne zarządzanie i monitorowanie dostępu do tych krytycznych zasobów.
- Kontrola dostępu oparta na rolach (RBAC) w Kubernetes pozwala na ściśle określenie, które zasoby są dostępne dla poszczególnych użytkowników i usług, zwiększając bezpieczeństwo i zapobiegając nadmiernym uprawnieniom.
- Zaimplementowany zostanie również firewall na poziomie sieci, który będzie nadzorował cały ruch wchodzący i wychodzący, umożliwiając tylko uprzednio zdefiniowane i zabezpieczone połączenia.

Plan optymalny:


- Rozbudowując plan minimalny, plan optymalny uwzględnia zaawansowane rozwiązania zwiększające poziom bezpieczeństwa.
- Bastion host zostanie użyty jako bezpieczny i monitorowany punkt dostępu do wewnętrznych zasobów sieciowych. Dostęp do bastion hosta będzie możliwy tylko z

wykorzystaniem sieci VPN, co zapewni dodatkowe szyfrowanie ruchu i izolację od potencjalnych zagrożeń z internetu.

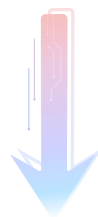
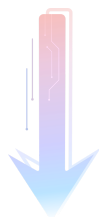
- Wirtualna sieć prywatna (VPN) posłuży jako tunel do bezpiecznej komunikacji między użytkownikami zdalnymi a wewnętrznymi zasobami sieci, ograniczając ryzyko przechwycenia danych w transporcie.
- Dodatkowo, wdrożony zostanie Web Application Firewall (WAF), który będzie chronił aplikacje internetowe przed atakami na poziomie warstwy aplikacji, takimi jak *SQL-injection* czy *XSS*.

Jak wyszło w praktyce

Przed dalszym przeglądaniem dokumentu należy puścić w tle:

 Tu turu tu

MINDPULSE



Notifications



[More events in the activity log →](#)

[Dismiss all](#)

Deployment failed

Deployment to resource group 'RESOURCE_GR_9' failed.
Additional details from the underlying API that might be helpful: At least one resource deployment operation failed. Please list deployment operations for details. Please see <https://aka.ms/arm-deployment-operations> for usage details.

a few seconds ago

Deployment failed

Deployment to resource group 'RESOURCE_GR_9' failed.
Additional details from the underlying API that might be helpful: At least one resource deployment operation failed. Please list deployment operations for details. Please see <https://aka.ms/arm-deployment-operations> for usage details.

3 minutes ago

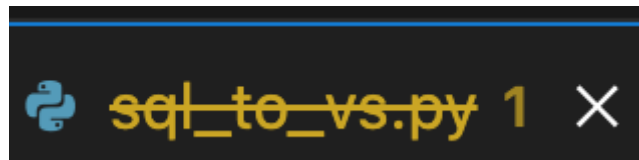
Deployment failed

Resource type 'Microsoft.ContainerInstance/containerGroups' container group quota 'StandardCores' exceeded in region 'francecentral'. Limit: '10', Usage: '10' Requested: '1'. [Click here for details](#)

Azure Server | Deploy Download Refresh

Resource type 'Microsoft.ContainerInstance/containerGroups' container group quota 'StandardCores' exceeded in region 'francecentral'. Limit: '10', Usage: '10' Requested: '1'. [Click here for details](#)

Resource type 'Microsoft.ContainerInstance/containerGroups' container group quota 'StandardCores' exceeded in region 'francecentral'. Limit: '10', Usage: '10' Requested: '1'. [Click here for details](#)



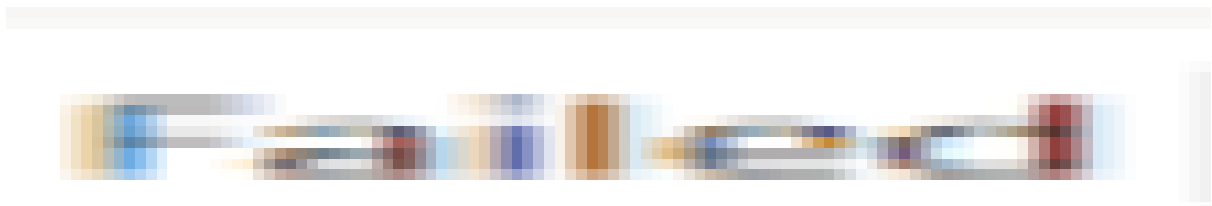
Search resources, services, and docs (G+)		
View template		
	Status	
net	✓	Succeeded
	!	Failed (Error details)
net	✓	Succeeded
	!	Failed (Error details)
net	✓	Succeeded
	✓	Succeeded
	✓	Succeeded
	!	Failed (Error details)
IC...	✓	Succeeded
	✓	Succeeded
	✓	Succeeded
	✗	Canceled
	✓	Succeeded
	✓	Succeeded
42...	✓	Succeeded

Vulnerabilities (110)

```
File "C:\Users\ml\PycharmProjects\bluesoft\accenture-hackathon\.env\lib\site-packages\pandas\io\common.py", line 873, in get_handle
    handle = open(
FileNotFoundError: [Errno 2] No such file or directory: '/Users/bartek/Documents/ai_persp/accenture-hackathon/backend/data/products.csv'
```

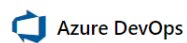
```
No such file or directory: '/Users/bartek/Documents/ai_persp/
```


Informacja o tym czy ryzyko zostało zaadresowane. (Tak/Nie)												
	Nie	Nie	Nie	Nie	Nie	Nie	Nie	Nie	Nie	Nie	Nie	Tak



AAAALEEEE...

nie ma miękkiej gry



robert.komar@accenturehackaton.onmicrosoft.com

**Taking you to your Azure
DevOps organization...**



Containers

Images

Volumes

Dev Environments BETA

Docker Scout

Learning center

Extensions

Add Extensions

damasosanoja/scout-demo:v1

8e60fea79666

CREATED 1 hour ago

SIZE 47.32 MB

Recommended fixes

Run

Advanced image analysis is provided by Docker Scout

Upgrade to continue to get access to guided vulnerability remediation and additional software supply chain features. [Learn more](#) and [upgrade](#).

Image hierarchy

FROM alpine:3, 3.14, 3.14.1, latest

ALL damasosanoja/scout-demo:v1

Layers (9)

0	ADD file:1a8fd1066485e1261462e689c1...	5.33 MB	
1	CMD ["/bin/sh"]	0 B	
2	ENV BLUEBIRD_WARNINGS=0 NODE_EN...	0 B	
3	RUN /bin/sh -c apk add --no-cache nodej...	39.61 MB	
4	COPY package.json ./ # buildkit	800 B	
5	RUN /bin/sh -c apk add --no-cache npm ...	2.27 MB	
6	COPY ./app # buildkit	110.02 KB	
7	CMD ["node" "/app/app.js"]	0 B	
8	EXPOSE map[3000/tcp:0]	0 B	

Images (2)

Vulnerabilities (27)

Packages (79)

[Give feedback](#)

Package or CVE name

Fixable packages

Reset filters

Package	Vulnerabilities
alpine/openssl 1.1.1k-r0	1 C 4 H
alpine/zlib 1.2.11-r3	1 C 1 H
alpine/busybox 1.33.1-r3	10 H 2 M
qs 6.7.0	1 H 0 M
express 4.17.1	1 H 0 M
alpine/libretls 3.3.3p1-r2	1 H 0 M
CVE-2022-0778	7.5 H

CVSS Score: 7.5
Affected range: <3.3.p1-r3
Fix version: 3.3.3p1-r3
Publish date: 2022-03-15

1-6 of 6

HOME > RESOURCE_GRP_9 >

MIND_LOAD

Load balancer

Search

Move Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Frontend IP configuration

Backend pools

Health probes

Load balancing rules

Inbound NAT rules

Outbound rules

Essentials

Resource group (move) : RESOURCE_GRP_9

Location : France Central

Subscription (move) : Accenture Hackaton Subscription

Subscription ID : c5c6f0ba-9990-4683-a443-6f3e265b053a

SKU : Standard

Tags (edit) : Add tags

See more

Backend pool : mind_pool (1 virtual machine)

Load balancing rule : rule_mind (Tcp/80 to Tcp/3000)

Health probe : health_mind (Tcp/3000)

NAT rules : 0 inbound

Tier : Regional

Configure high availability and scalability for your applications

Create highly-available and scalable applications in minutes by using built-in load balancing for cloud services and virtual machines. Azure Load Balancer supports TCP/UDP-based protocols and protocols used for real-time voice and video messaging applications. [Learn more](#)

mindpulse1234117_z1	Network Interface	France Central
mindpulse12701_z1	Network Interface	France Central
mindpulse12_OsDisk_1_a0ecfe31ef141ab8ac34e78763dd71b	Disk	France Central
nowe_sd	Public IP address	France Central

UDAŁO SIĘ!!!

więc warto się nie poddawać.

Następny Release

- Ze względu na to, że niestety czasu było za mało, żeby skończyć całą aplikację, backend niepoprawnie pobiera ścieżki po wrzuceniu na wirtualny obraz docker. (działa lokalnie)
W następnym release wrzucimy poprawkę, która odpowiednio nakieruje wszystkie ścieżki zawarte w pliku.
- Ponadto zadbamy o to, żeby poprawić architekturę systemu i wystawimy ją na container Instances albo na klastrze kubernetesa z wykorzystaniem ArgoCD.
- Dodamy również skrypt CRON, który raz dziennie będzie update'ował baze wektorową jeśli wystąpiły w bazie SQL jakieś zmiany.
- Rozbudowanie architektury vectorstore o metodyka RAPTOR, która zbierała wiele schem, wiele baz danych i znajdując po nich relacje łączyła je w klastry dają high overview całość pozwalając na skuteczniejsze wyszukiwanie adekwatnych tabel etc.