

Tema 5 Cloud Computing

Rosca Victor-Ciprian 3B3

Mai 2024

1 Studiu de Caz în contextul realității internaționale/naționale

DataLogix este un sistem de monitorizare și raportare care rulează local pe mașina clientului, colectând și analizând în mod regulat datele despre performanța și utilizarea serverului. Acest proiect oferă o soluție eficientă pentru managementul și monitorizarea serverelor, ajutând companiile să își optimizeze resursele și să își îmbunătățească securitatea și performanța sistemelor informatice.

Scopul principal al proiectului este de a dezvolta un sistem automatizat de monitorizare a serverului, care să colecteze informații relevante despre procese, gradul de încărcare, utilizatorii conectați, serviciile oferite și tipurile de conexiuni. Aceste date sunt apoi prelucrate pentru a genera rapoarte periodice care să ofere o imagine clară și detaliată asupra performanței și utilizării serverului.

Caracteristici cheie:

Colectarea Automată de Date: DataLogix monitorizează în mod regulat serverul și colectează informații despre procese, gradul de încărcare, utilizatorii conectați, serviciile oferite și tipurile de conexiuni.

Generarea de rapoarte: Pe baza datelor colectate, DataLogix generează rapoarte periodice care includ informații despre încărcarea memoriei, numărul de conexiuni ale fiecărui utilizator și numărul de conexiuni pentru fiecare serviciu pe intervale de timp

Integrarea cu Blob Storage: După generarea rapoartelor, un trigger declanșează încărcarea fișierului de text pe un blob storage, unde datele pot fi stocate într-un mod sigur și scalabil.

Interpretarea de Catre Compania de IT Support: Datele stocate în blob storage sunt apoi interpretate de către o companie specializată în servicii de IT support, care oferă clienților săi o înțelegere mai profundă a performanței și utilizării serverului, precum și recomandări pentru optimizare și îmbunătățire.

Beneficii:

Monitorizare Eficientă: ServerMonitor oferă o monitorizare automatizată și continuă a serverului, reducând nevoia de intervenție manuală

Optimizare a Performanței: Rapoartele periodice generat de ServerMonitor oferă informații valoroase despre utilizarea resurselor serverului, permițând companiilor să își optimizeze infrastructura și să își îmbunătățească performanța

Securitate si Scalabilitate: Stocarea datelor în blob storage oferă un nivel ridicat de securitate și scalabilitate, asigurând că informațiile sunt protejate și accesibile în orice moment

DataLogix reprezintă o soluție inovatoare și eficientă pentru managementul și monitorizarea serverelor, oferind beneficii pentru companii de toate dimensiunile și industriile. Prin automatizarea procesului de colectare și analiză a datelor, acest proiect contribuie la îmbunătățirea performanței și eficienței operaționale a sistemelor informatice.

2 Detalii Tehnice

Arhitectură: DataLogix este o soluție de monitorizare a serverelor care utilizează o arhitectură distribuită și include două componente principale:

Agentul de Monitorizare: Un agent software care rulează pe fiecare server individual și colectează date despre performanță și utilizare

Platforma Cloud: O platformă centralizată în cloud care primește datele de la agenți și utilizează tehnologii moderne de stocare și procesare a datelor, cum ar fi serviciile de Blob Storage pentru stocarea datelor

Instrumente tehnice:

Agentul de Monitorizare: Acesta este dezvoltat utilizând limbajul de programare C++ și Python pentru performanță și eficiență

Abordări de Marketing:

Diferențiere prin performanță: DataLogix se diferențiază prin performanța și eficiența sa. De asemenea, Agentul de monitorizare are un impact minim asupra resurselor serverului

Securitate și Conformitate: Cheile secrete sunt stocate în mod sigur în Key Vault. Acest lucru asigură că datele sunt criptate atât în tranzit, cât și în repaus, garantând confidențialitatea și securitatea acestora

3 Tehnologii Folosite

3.1 TCP - Transmission Control Protocol

Am folosit TCP(Transmission Control Protocol) deoarece toti biti sunt livrati corecti, in ordine si mascheaza tranzitia pachetelor pierdute. TCP este un protocol de comunicare ce efectueaza o conectare intre doua puncte terminale, definite de o adresa IP si de un port. Astfel, TCP se ocupa de monitorizarea pachetelor pentru a se asigura ca ajung in ordine si ca nu sunt corupte in drum spre destinatie; pachetele pot fi recuperate si retrimise catre destinatar (lucru imposibil in cadrul UDP). Proiectul se numeste DataLogix.

3.2 Baza de date SQL

Pentru a stoca datele (procesele, gradul de incarcare, utilizatorii conectati, servicii oferite si tipuri de conexiuni) si utiliza, voi folosi o baza de date MySQL. Aceasta varianta de stocare a datelor este potrivita acestui proiect, deoarece datele pe care trebuie sa le foloseasca aplicatia se incadreaza bine in modelul relational. Astfel, fiecare linie din tabel va reprezenta o noua intrare, in vreme ce fiecare coloana (atribut) va cuprinde o anumita informatie despre user: PID, SESSION, USERNAME, PASSWORD). Astfel, se vor popula tabelele si vom modela datele din baza de date.

3.3 Azure service Key Vaults

Utilizez Key Vaults pentru a stoca și gestiona în mod sigur o cheie secretă necesară pentru autentificare, permițând apoi încărcarea conținutului folosind aceea cheie pentru a accesa o funcție Azure.

3.4 Azure service DevOps

Folosim Azure DevOps pentru a automatiza procesul de deploy al funcției trigger către medii Azure.

3.5 Azure service Function App

Folosim o funcție Azure pentru a automatiza procesul de încărcare a unui fișier text pe un container de blob-uri în Azure Storage. Această funcție este proiectată pentru a fi declanșată de un eveniment specific, cum ar fi o cerere HTTP POST către o rută specifică.

Atunci când funcția este declanșată, aceasta primește conținutul fișierului text ca parte a corpului cererii HTTP. Apoi, utilizează cheia de acces a contului de stocare Azure, stocată în variabila de mediu STORAGE ACCOUNT KEY, pentru a autentifica și a se conecta la contul de stocare.

După ce autentificarea este reușită, funcția creează un client pentru serviciul de blob-uri Azure folosind biblioteca Azure SDK for Python. Acest client este utilizat pentru a obține un client de blob specific pentru a urca conținutul fișierului text pe un container de blob-uri.

Funcția generează o cale unică pentru fișierul blob, folosind data și ora curentă. În final, funcția încarcă conținutul fișierului text pe locația specificată în containerul de blob-uri și returnează un răspuns HTTP pentru a indica că încărcarea a fost realizată cu succes.

3.6 Azure Blob Storage

Serviciul Azure Blob Storage este utilizat pentru a stoca și gestiona în mod eficient fișierele text care conțin datele primite de la serverul nostru.

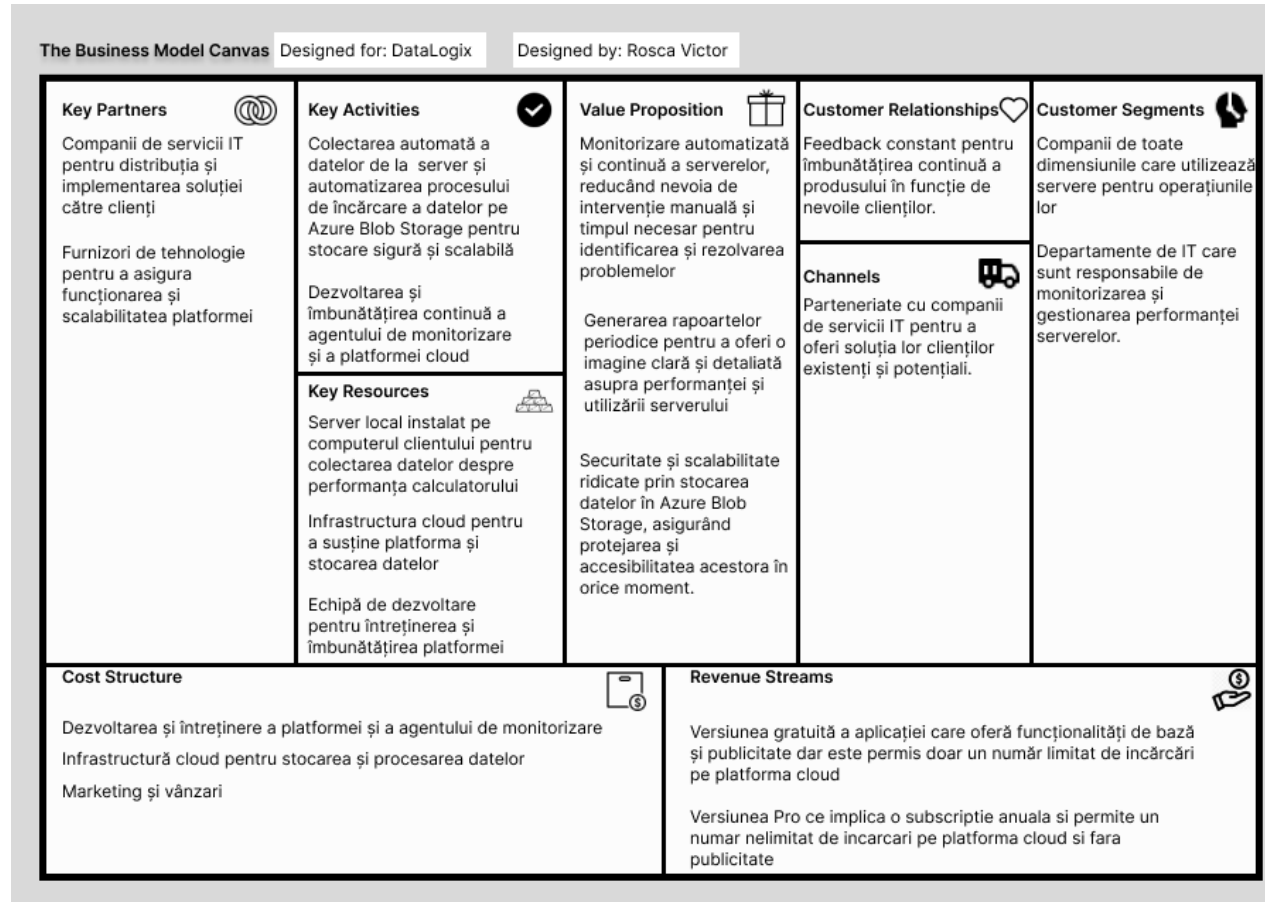
3.7 Azure Data Factory

În Azure Data Factory, declanșăm manual un trigger pentru a executa un flux de date. După această acțiune, în panoul de monitorizare a execuției, se vede numele pipeline-ului care a fost declanșat, precum și o mențiune că declanșarea a fost făcută manual.

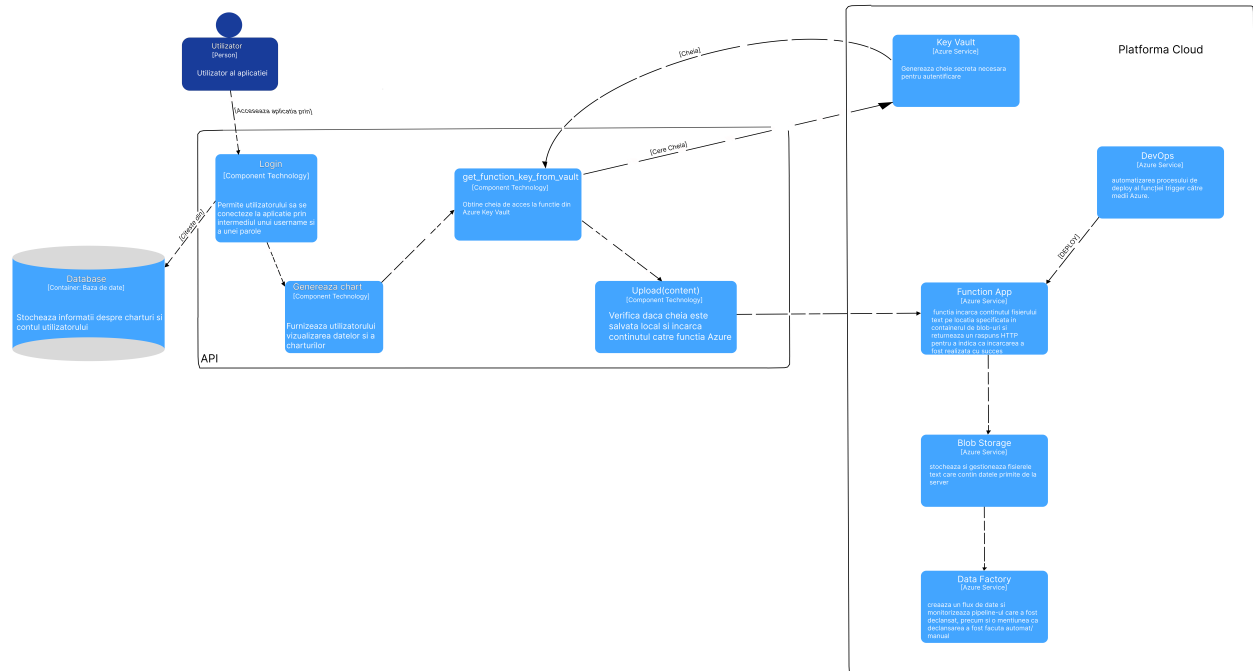
De asemenea, putem să verificăm în Azure DevOps, în secțiunea de pipelines, unde vom vedea că execuția pipeline-ului a fost marcată ca succes, confirmând că fluxul de date a fost rulat cu succes.

Această sincronizare între Azure Data Factory și Azure DevOps ne permite să monitorizăm și să gestionăm eficient întregul proces de dezvoltare și execuție a fluxurilor de date.

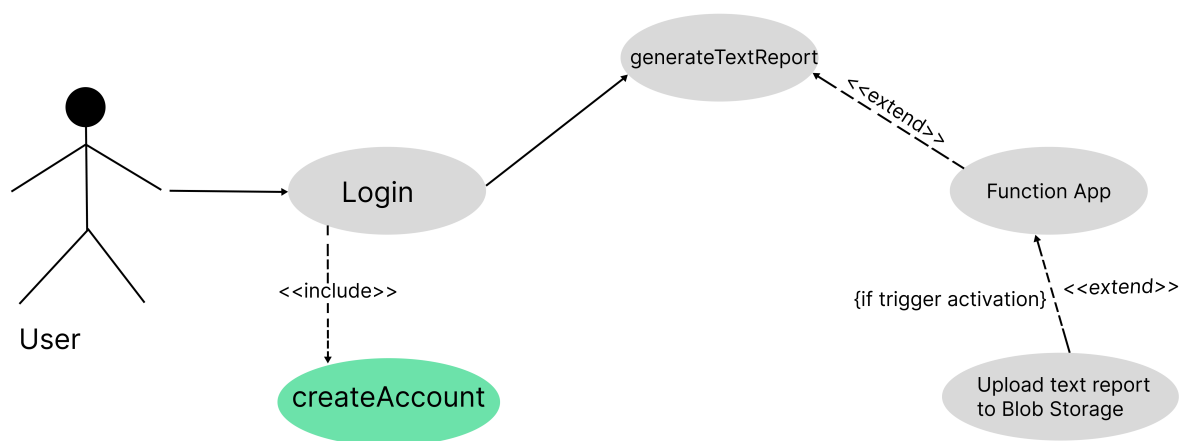
4 Business Canvas



5 Architectural Diagram



6 Use-Case Diagram



7 Documentarea API-urilor folosind SwaggerHub

openapi: 3.0.0

info:

title: DataLogix API

version: 1.0.0

paths:

/collect-data:

post:

summary: Endpoint pentru colectarea datelor serverului

description: Acest endpoint permite serverului să trimită
datele colectate către platforma DataLogix

requestBody:

required: true

```

        content:
          application/json:
            schema:
              $ref: '/components/schemas/DataPayload'

    responses:
      '200':
        description: Datele au fost trimise cu succes
      '400':
        description: Cererea este invalidă

  components:
    schemas:
      DataPayload:
        type: object
        properties:
          processes:
            type: array
            items:
              type: string
            description: Lista proceselor în execuție
          memory_load :
            type : number
            description : Gradul de ncrcare a memoriei
          users :
            type : array
            items :
              type : string
            description : Lista utilizatorilor conectați
          services :
            type : array
            items :

```

type : string
description : Lista serviciilor oferite de server
connections :
type : array
items :
type : string
description : Lista tipurilor de conexiuni

7.1 Documentarea fluxurilor de functionalitate

Fluxul de Funcționalitate: Colectarea Automată a Datelor

Scop: Scopul fluxului este de a colecta automat datele relevante despre server și de a le trimite către platforma cloud.

Cerințe inițiale:

- Serverul trebuie să monitorizeze în mod regulat procesele, gradul de încărcare a memoriei, utilizatorii conectați, serviciile oferite și tipurile de conexiuni.
- Datele trebuie să fie trimise către platforma cloud automat după generarea raportului.

Rezultat final: Datele colectate sunt trimise cu succes către platforma cloud.

Detalii tehnice:

- Tehnologia utilizată: TCP pentru comunicare între server și client.
- Integrări: Utilizarea Azure Blob Storage pentru stocarea datelor colectate.

Gestionarea Excepțiilor:

- Excepție 1: Eșec la trimiterea datelor către platforma DataLogix.

- Excepție 2: Cererea de colectare a datelor este invalidă.

Testare și Validare:

- Metode de testare: Testare manuală și testare automată folosind framework-uri specifice.
- Validare: Verificarea datelor trimise către platforma cloud și a rapoartelor generate.