

ДОДАТОК А

Презентація



Дослідження методів розробки хмарних інтеграторів e- commerce додатків

Виконавець: ст. гр. ІПЗмзд 17-1 Ворона М. О.
Керівник: доц. Лановий О.Ф.

Досліджуємі питання

1. Постановка задачі на дослідження та огляд існуючих рішень
2. Аналіз типових вимог до e-commerce системи
3. Порівняння монолітної та мікросервісної архітектур
4. Огляд хмарних інтеграторів у якості інструменту для побудови Мікросервісу
5. Створення дескриптору розгортання у GCP для одного з мікросервісів

Мета виконання дипломної роботи

Метою роботи є аналіз існуючих архітектурних стилей та підходів до розробки складних систем електронної комерції та використання хмарних інтеграторів у процесі розробки.

3

Функціональні області у е-commerce системі



4

Монолітна архітектура



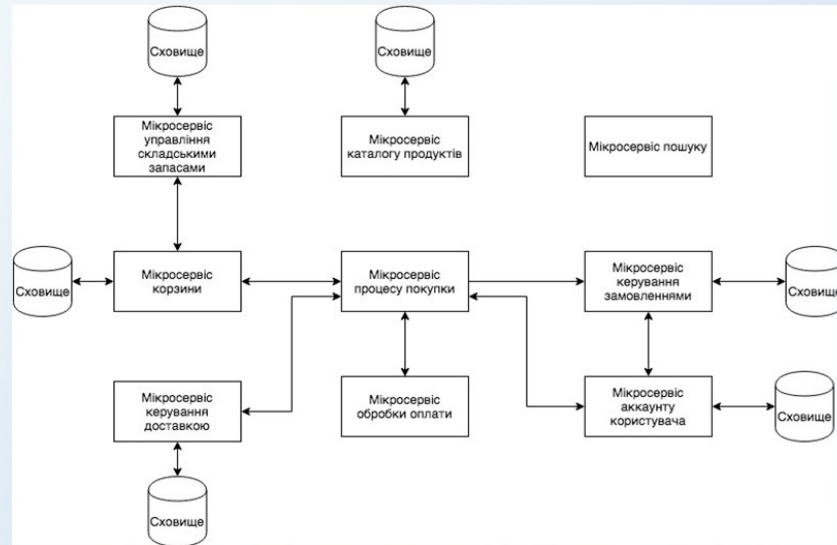
5

Мікросервісна архітектура



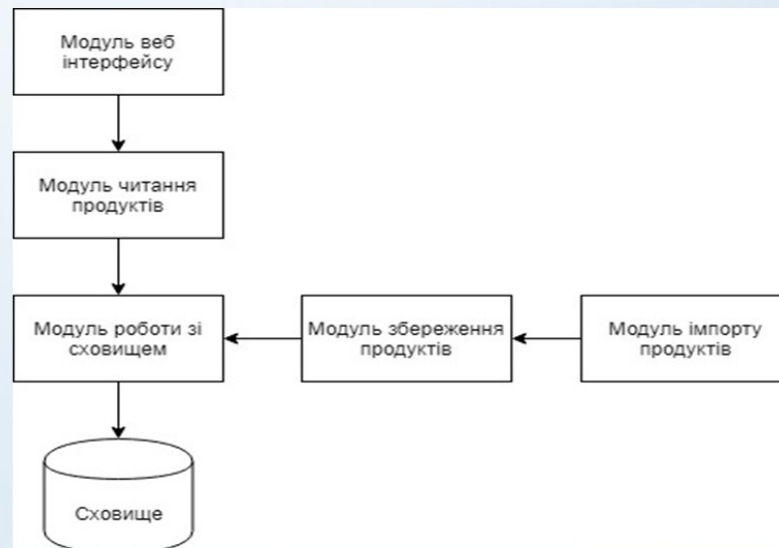
6

Мікросервісна архітектура e-commerce системи



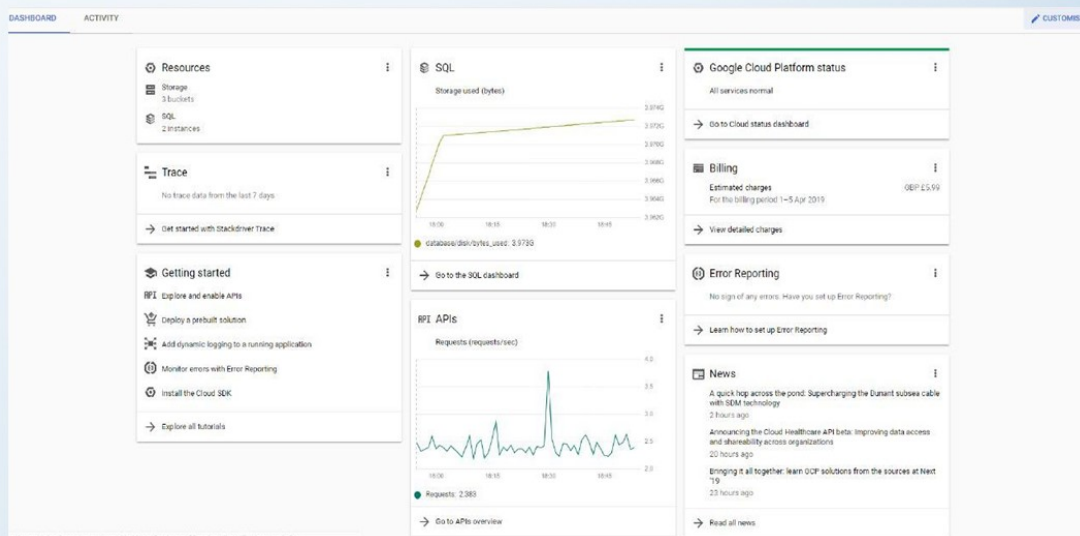
7

Структурна схема мікросервісу каталогу продуктів

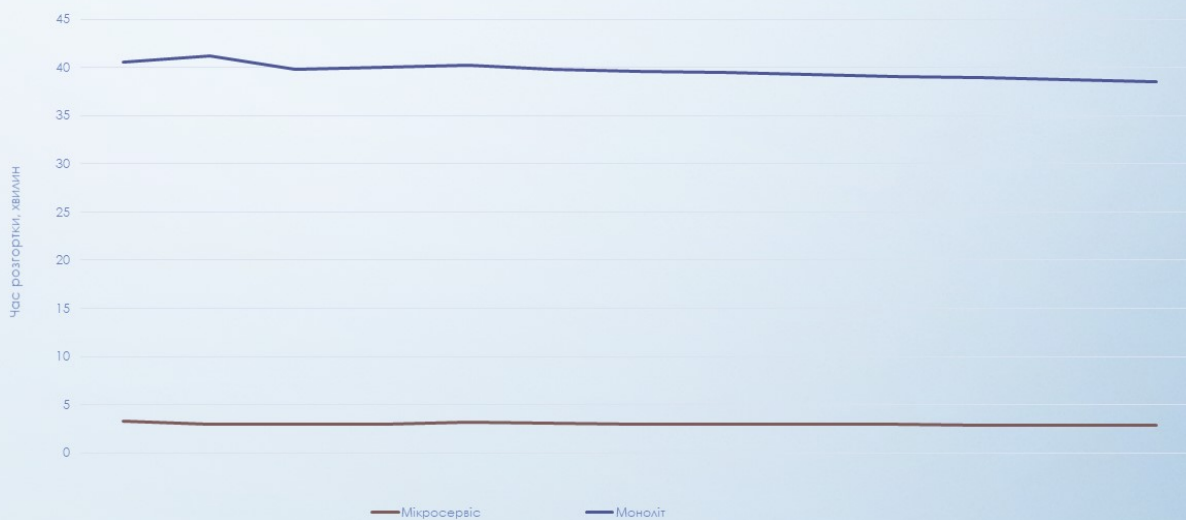


8

Веб консоль хмарного інструменту Google Cloud Platform



Порівняльній характеристика часу розгортання



Критерії порівняння мікросервісу та моноліту

- Тривалість релізного циклу;
- Стабільність системи;
- Вартість підтримки;
- Масштабуємість;
- Простота тестування;
- Тривалість розгортання;
- Складність впровадження на початковому етапі.

11

Висновки

- Було проаналізовано різні підходи у проектуванні системи електронної комерції з точки зору складності розробки та підтримки, швидкості релізного циклу, швидкості та вартості розгортання;
- Складена порівняльна характеристика монолітної та мікросервісної архітектури, використовуючи метод лінійної аддитивної згортки з ваговими коефіцієнтами;
- Модифіковано метод для побудови великих систем електронної комерції на основі мікросервісної архітектури.
- Запропонований метод дозволяє спростити процес розробки та пришвидшити релізний цикл великих систем електронної комерції.

12

Дякую за увагу!

ДОДАТОК Б

Файли розгортання мікросервісу за допомогою Kubernetes

```
kind: Service
apiVersion: v1
metadata:
  name: product-catalog
  namespace: diploma
  labels:
    app: product-catalog
    environment: stubs
    serviceExposition: internal
spec:
  type: ClusterIP
  selector:
    app: product-catalog
    environment: stubs
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: http-proxy
```

```
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: product-catalog
  namespace: diploma
spec:
  replicas: 2
  minReadySeconds: 10
  selector:
    matchLabels:
      app: product-catalog
      environment: stubs
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 0
  template:
    metadata:
      labels:
        app: product-catalog
```



```

    environment: stubs
spec:
  restartPolicy: Always
  terminationGracePeriodSeconds: 30
  automountServiceAccountToken: false
  securityContext:
    fsGroup: 101
    runAsNonRoot: true
  containers:
  - name: product-catalog
    image: ${GOOGLE_IMAGE_TAG}:${PRODUCT_API_VERSION}
    imagePullPolicy: Always
    ports:
    - containerPort: 8080
      name: http
    resources:
      requests:
        memory: "500Mi"
        cpu: "1000m"
      limits:
        memory: "1000Mi"
        cpu: "1000m"
    securityContext:
      readOnlyRootFilesystem: true
      runAsUser: 1001
    env:
    - name: DB_USER
      valueFrom:
        secretKeyRef:
          name: product-catalog-mysql-inst-1
          key: user
    - name: DB_PASSWORD
      valueFrom:
        secretKeyRef:
          name: product-catalog-mysql-inst-1
          key: password
    envFrom:
    - configMapRef:
        name: product-catalog-env-config
  readinessProbe:
    httpGet:
      path: /api/v1/admin/health
      port: 8080
    initialDelaySeconds: 60
    periodSeconds: 10

```

```

        failureThreshold: 10
livenessProbe:
  httpGet:
    path: /api/v1/admin/health
    port: 8080
    initialDelaySeconds: 60
    periodSeconds: 10
    failureThreshold: 10
- name: cloudsql-proxy
  image: gcr.io/cloudsql-docker/gce-proxy:1.11
  command: ["/cloud_sql_proxy",
    "-instances=product-flex:eu-west1:
product-catalog-mysql-inst-1=tcp:3306",
    "-
credential_file=/secrets/cloudsql/credentials"]
  securityContext:
    runAsUser: 2 # non-root user
    allowPrivilegeEscalation: false
  volumeMounts:
    - name: product-catalog-mysql-inst-1-credentials
      mountPath: /secrets/cloudsql
      readOnly: true
volumes:
  - name: product-catalog-mysql-inst-1-credentials
    secret:
      secretName: product-catalog-mysql-inst-1-
credentials
  - name: product-catalog-mysql-inst-1-credentials
    secret:
      secretName: product-catalog-mysql-inst-1-
credentials
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: product-catalog-env-config
  namespace: diploma
data:
  OTHER_SERVICE_URL: other-service:80

```