

K8S OpenShift – מבחן – Terrorists API

פרטי המבחן

משך זמן: 4 שעות

ניקוד כולל: 100 נקודות

הגשה: קישור לרפזיטורי GitHub שמוגש למטרלה הרלוונטיות במודול

תרחיש

כמה חודשים עברו אחרי שיפור התחליל לעובד ביחד עם המוסד כדי לאוצר מוחבים ימ"ח שינם וצה"ל ביחד עם חברת רפאל פיתחו מל"ט חדש ויחודי שיודיע להATABד על מטרה ספציפי ולהשל את המטרה. הבעיה היא שיש רק מספר מוגבל של מל"טים שאפשר לפתח מחדש لكن נוצר דרישת למצוא את המוחבים הקיימים מסוכנים שעליהם יופעל המיל"טים.

אתם בונים מיקרו-שירותים לקליטת נתוניים של מוחבים עבור ייחדות המודיעין של צה"ל. השירותים מקבל קובץ CSV המכיל רשומות של מוחבים, מעבד את הנתוניים באמצעות pandas, מדרג אותם לפי רמת מסוכנות, מחזיר את 5 המוחבים המסוכנים ביותר, ושומר את התוצאות ב-MongoDB. כל המערכת חייבת להיות ניתנת לפריסה ב-OpenShift עם **3 רפליקות של MongoDB** ו-**3 רפליקות של ה-API**.

חומר עזר מותרים:

אין להשתמש בקוד עבר שכתבתם או חיפוש בגוגל אלא במקורות הבאים:

[W3 Schools](#)

[GeeksforGeeks](#)

[Real Python](#)

(שימושelman במנוע חיפוש פנימי ולא באמצעות גוגל)

official docs:

[Docker Docs](#)

[Kubernetes Documentation](#)

[OpenShift Container Platform | 4.20 | Red Hat Documentation](#)

YAML - [The YAML Validator](#)

[pandas 2.3.3 documentation](#)

[Tutorial - User Guide - FastAPI](#)

- HTTP Post request with a csv file:

[How to write a FAST API function taking .csv file and making some preprocessing in pandas - Stack Overflow](#)

[Models - Pydantic Validation](#)

חלק 1: דרישות אפליקציית FastAPI

1.1 הגדרת Endpoint

יש ליצור אפליקציה FastAPI עם **endpoint אחד בלבד**:

Endpoint:
POST /top-threats

פורמט הבקשה:

- Content-Type: multipart/form-data •
 file: שם השדה •
 CSV: סוג הקובץ •
-

1.2 תהליכי העבודה (Processing Pipeline)

יש למש את השלבים הבאים **בדיווק לפי הסדר**:

1. קריית קובץ CSV pandas DataFrame של pandas
2. מילוי לפי danger_rate בסדר יורד
3. בחירת 5 השורות העליונות
4. ולידציה והמרה באמצעות מודל Pydantic כך שיישארו רק השדות:
 - name (מחוץ)
 - location (מחוץ)
 - danger_rate (מספר שלם, 10–1)

1.3 פורמט התגובה

יש להחזיר JSON מבנה המדוקן הבא:

```
{  
  "count": 5,  
  "top": [  
    {"name": "Mohammad Abu Maniac", "location": "Kuan Yunes", "danger_rate": 10},  
    {"name": "Hassan Balboah", "location": "Gaza City", "danger_rate": 9},  
    {"name": "Ahmed Aiben Imashlo", "location": "Rafah", "danger_rate": 8},  
    {"name": "Sharif Hammudi", "location": "Sagaiyah", "danger_rate": 7},  
    {"name": "Zoran Mamdani", "location": "Lebanon City", "danger_rate": 6}  
  ]  
}
```

חשיבות:

- אם לאחר העיבוד קיימות פחות מ-5 שורות תקינות – יש להחזיר את מספר הרשומות המקוריים
 - יש לעדכן את השדה `count` בהתאם למספר הרשומות שהוחזר בפועל
-

1.4 שמירה ב-MongoDB

לאחר יצירת התגובה:

1. יש להכניס (Insert) את כל הרשומות מתוך המערך `top` ל-**Collection**: `top_threats`
2. שם האוסף (Collection): `top_threats`
3. מבנה המסמך:

```
{  
  "name": "Alpha",  
  "location": "Metropolis",  
  "danger_rate": 10  
}
```

1.5 משתני סביבה

אפליקציית API **חייבת** לקרוא את המשתני הסביבה הבאים:

תיאור	ערך לדוגמה	משתנה
MongoDB שמו של הפוד הראשון של hostname	<code>mongo-0.mongo</code>	<code>MONGO_HOST</code>
פורט MongoDB	<code>27017</code>	<code>MONGO_PORT</code>
שם משתמש root של MongoDB	<code>admin</code>	<code>MONGO_USERNAME</code>
סיסמת root של MongoDB	<code>secretpass</code>	<code>MONGO_PASSWORD</code>
שם בסיס הנתונים threat_db		<code>MONGO_DB</code>
בסיס הנתונים לאימות admin		<code>MONGO_AUTH_SOURCE</code>

1.6 טיפול בשגיאות

יש למשתמש טיפול נכון בשגיאות עם קוד HTTP מתאימים:

הודעת תגובה	קוד HTTP	מצב שגיאה
{"detail": "No file provided"}	400	לא סופק קובץ
{"detail": "Invalid CSV file"}	400	הקובץ אינו CSV / שגיאת פרטסור
detail": "Database" } {"unavailable	503	MongoDB אינו נגיש
פירוט שגיאות ולידציה של Pydantic	422	פורמט נתונים שגוי

1.7 רשימת טכנולוגיות

- גרסה Python: 3.11 ומעלה
- ספריות נדרשות:
 - fastapi
 - pandas
 - pydantic
 - pymongo
 - python-multipart (להעלאת קבצים)
 - (שרות uvicorn)

חלק 2: דרישות Docker

2.1 קבצים נדרשים

שורש הרפזיטורי חייב לכלול:

- Dockerfile •
- dockerignore. •
- gitignore. •

דרישות (חובה) 2.2 Dockerfile

דרישות חובה:

- `Image` בסיס: `python:3.11-slim` או דומה
 - `マイコム קוד האפליקציה: app/`
 - `포트 חשוף: 8000`
 - `פקודת הפעלה: הרצת unicorn על 0.0.0.0:8000`
 - `העתקת requirements.txt`
-

קובץ 2.3 dockerignore

חייב להחריג לפחות:

- `__pycache__/`
 - `*.pyc`
 - `.git/`
 - `k8s/`
 - `.gitignore`
 - `README.md`
 - `screenshots/`
-

חלק 3: דרישות פריסה ב-OpenShift

כל קבצי ה-YAML של Kubernetes חייבים להיות פרוסים על OpenShift Cluster.

MongoDB StatefulSet 3.1

קובץ: k8s/mongo-statefulset.yaml

הגדרות בסיסיות:

מאפיין	ערך	שם
	mongo	
	3	מספר רפליקות
	mongo	שם השירות
	docker.io/library/mongo:7	Image
	27017	פורט קונטינר

:Labels

מפתח	ערך	שם
	mongo	app
	database	tier

משתני סביבה:

ערך	שם משתנה
admin	MONGO_INITDB_ROOT_USERNAME
secretpass	MONGO_INITDB_ROOT_PASSWORD
threat_db	MONGO_INITDB_DATABASE

:**(Persistent Volume Claim (volumeClaimTemplates**

מופיע	ערך	מופיע
mongo-data	Claim	שם ה-
ReadWriteOnce	Access Mode	
1Gi	בקשה אחסן	

:**Volume Mount**

מופיע	ערך	מופיע
mongo-data	Volume	שם
data/db/	Mount	נתיב

MongoDB 3.2 שירות

קובץ: k8s/mongo-service.yaml

הגדרות:

מופיע	ערך	מופיע
mongo		שם
ClusterIP		וירטואלי
(None (headless	ClusterIP	
27017		פורט
27017	Target Port	

:Labels

מפתח	ערך
app	mongo
tier	database

:Selector

מפתח	ערך
app	mongo
tier	database

3.3 פריסת API

קובץ: `k8s/api-deployment.yaml`

הגדרות בסיסיות:

מאפיין	ערך
שם	threat-api
רפליקות	3
Image	<code>your-dockerhub-username>/threat-api:v1></code>
פורט קונטינר	8000

:Labels

מפתח	ערך
app	threat-api
tier	backend

משתני סביבה:

ערך	שם משתנה
mongo-0.mongo	MONGO_HOST
27017	MONGO_PORT
admin	MONGO_USERNAME
secretpass	MONGO_PASSWORD
threat_db	MONGO_DB
admin	MONGO_AUTH_SOURCE

3.4 שירות API

קובץ: k8s/api-service.yaml

הגדרות:

מופיע	שם
threat-api-svc	
NodePort	טאג
8000	פורט
8000	Target Port
30080	Node Port

:Labels

מפתח	ערך
threat-a	app
pi	
backend	tier

:Selector

מפתח ערך

threat-a app
pi

backend tier

חלק 4: מבנה הרפוזיטורי

הרפוזיטורי חייב להיות במבנה הבא בדיק:

```
└── main.py
└── models.py
└── db.py
└── requirements.txt
└── k8s/
    ├── mongo-statefulset.yaml
    ├── mongo-service.yaml
    ├── api-deployment.yaml
    └── api-service.yaml
└── screenshots/
    ├── pods.png
    ├── openshift-ui.png
    └── api-request.png
└── commands.txt
└── Dockerfile
└── .dockerignore
└── .gitignore
└── README.md
```

חלק 5: דרישות קובץ Commands

יש ליצור קובץ בשם `commands.txt` שיפוריע בענף הראשי של הרפזיטורי המכיל את כל הפקודות הדרשיות לבנייה ולפריסה של האפליקציה.
הפקודות חייבות לעבוד בראף כאשר מרכיבים אותן לפי הסדר.

חשיבות:

אין לכלול פקודות `git`. ופקודות של מערכת הפעלה (כגון ייצרת תיקיות וכו')
יש לכלול רק פקודות Docker ו-`oc` (OpenShift).

חלק 6: תהליכי עבודה עם Git

6.1 נדרשים Branches

הרפזיטורי חייב להכיל:

1. `main` – פתרון סופי עובד
2. `feature/app` – קוד אפליקציית API
3. `Kubernetes` – קבצי `k8s`

6.2 תהליכי פיתוח

1. ייצרת רפזיטורי ו-`git commit` ראשון על `main`
 2. ייצרת `feature/app` branch
 - פיתוח אפליקציית FastAPI
 - תכופים commits עם הודעות ממשמעותו
 - מיזוג ל-`main` בסיום branch
 3. ייצרת `feature/k8s` branch
 - פיתוח Kubernetes manifests של
 - תכופים commits
 - מיזוג ל-`main` בסיום branch
-

חלק 7: דרישות צילומי מסך

יש לכלול 3 צילומי מסך בתיקיית [screenshots](#) המאשרים שהאפליקציה קבוצה רצה ב-OpenShift

7.1 צילום 1: סטטוס פודים (pods.png)

צילום פלט הפקודה:

```
oc get pods
```

חיב להציג:

- כל פודי 2 mongo-0, mongo-1, mongo-2 במצב Running (MongoDB)
- כל פודי ה-Redis במצב Running API
- כל Jobs Init הסתיימו בהצלחה

7.2 צילום 2: ממושך (openshift-ui.png)

צילום ממושך ה-Web Console של OpenShift

חיב להציג:

- תצוגת Workloads או Topology
- StatefulSets וDeployments גלויים
- אינדיקציה ברורה שהמשאבים רצים

7.3 צילום 3: בקשה (api-request.png)

צילום של בקשה HTTP מוצלחת ל-Route של OpenShift

חיב להציג:

- URL מלא של ה-Route
- שיטת הבקשה (POST ל-[/top-threats](#))
- תגובה עם 5 הרשומות בפורמט JSON
- קוד סטטוס HTTP 200

כליים מותרים:

.דפדן עם REST Client, Postman, curl

חלק 8: דרישות README.md

קובץ README.md חייב לכלול:

Threat Analysis Microservice #

שם הסטודנט:* [שם מלא]

תעודת זהות / מזהה סטודנט:* [ID]

קורס / כיתה: [שם הקורס]

הוראות הגשה

1. יצרת רפזיטורי GitHub בשם **terrorists-api-k8s**
2. דחיפת הפתרון המלא ל-branch **main**
3. וידוא שכל ה-branches נדחפו לריום (main, feature/app, feature/k8s)
4. הגשת קישור לרפזיטורי דרך מערכת הקורס

הוראות חשובות

- יש לבדוק היטב לפני ההגשה – פתרון שלא עובד יגרור הורדת נקודות משמעותית
- יש לבצע commits תקופים עם הودעת משמעותית
- יש לוודא שכל קבצי YAML תקין ומעוצבים נכון
- יש לוודא שכל משתני הסביבה מוגדרים נכון הן בטבלאות והן בפריסות

בצלחה!