

# מבחן: פריסת אפליקציה ל-Streamlit

---

## ה Disk On Key מרפייה

יוסף מ"מ ערבה מצא במילואים האחרונים ברפיה disk on key, לאחר שבו מהמילואים, יוסף בדק את הcornon (כמובן שמשמעותו הגנה הוא השתמש במחשב ישן, חסר ערך, ללא מידע שלא מחובר לרשת) במקום לגנות מפתחות לביטוקין (כפי שהוא קיווה), התברר שהcornon מכיל מספר קבוע טקסט, ללא שמות אינפורטטיביים, המכילים נתונים טבליים.

באחד הקבצים יוסף הצליח לזהות אוסף קואורדינטות של מיקומים גיאוגרפיים (קווי אורך וגובה) ולהרדרתו התברר שמדובר באוסף נקודות שמרכזן הוא בכניסה נמרוד.

יוסף האחראי, העביר את cornon הנידיד ליד הגורם המודיעיני הכל בכיר שהוא מכיר - מני. מני לאחר בירורים בטור היחידה, החליט להעביר את הנתונים להמשר ניתוח לצוות אנליסטים גאונים - אלו שיודעים לזהות דפוסים בתוך סדרות של מספרים נתונים. כאיש פיתוח, מני בנה באופן基础上ן Docker-CLI המציג את הנתונים הגלומיים, ויזואлизיה שלהם, ויזואлизיה של הקואורדינטות על גבי מפה.

תפקידם לבנות ולהציג התהיליך פריסת Kubernetes, שיועבר לצוות פעולה ותשתיות של הענן הסודי של היחידה. במסגרת התהיליך עלייכם להגדיר קבוע דוקר עבור האפליקציה, מניפסטים עבור קוברים וקובץ המכיל את הפיקודות CLI שהשתמשתם בהם, כל התוכנות צרכות לעלות לגיטהאב באופן מסודר ובחולקה לבראנצ'ים. כך שנוכל להעביר את כל התהיליך באופן מסודר ומתועד לצוות פעולה ותשתיות

במבחן זה תבחןנו על פריסת אפליקציה (Python Streamlit) על גבי Kubernetes באמצעות Minikube. התרגיל משלב את הידע שלכם ב-Docker עם העולם של Kubernetes האפליקציה הנדרשת יושבת בגיטהאב (לינק בהמשך)

### חומר עזר מותר:

**אין לשימוש בחומר שנייתן בכיתה או קבצים אישיים.**

<https://minikube.sigs.k8s.io/docs/handbook/>

<https://octopus.com/devops/kubernetes-deployments/kubernetes-yaml-types/>

<https://medium.com/geekculture/cheatsheet-for-kubernetes-minikube-kubectl-5500ffd2f0d5>

## דרישות מוקדמות

לפני שמתחלים, תווידאו ש모תקנים אצלכם:

- Docker Desktop
- Minikube
- kubectl
- חשבון ב-Docker Hub (אישי)
- Git
- חשבון אישי ב-GitHub

## מה להגיש?

צרו פרויקט גיט עם הקבצים הבאים:

- .1 - קובץ Dockerfile
- .2 - קובץ Pod-Pod.yaml
- .3 - קובץ Kubernetes Service.yaml
- .4 - קובץ טקסט עם כל פקודות CLI שהרצה (מהשלב הראשון ועד האחרון)
- .5 - צילום מסך של האפליקציה פועלת בדף (כולל ה-URL בשורת הכתובת)

שים לב מצופה שכל קובץ יהיה בבראנס נפרד עם קובץ MD הכלול הסביר על מהות ומטרת הקובץ ורק לבסוף יש לאחד את הכל ל main .  
יש להעלות את הפרויקט לגיטהאב ולהדיבק את הקישור למטלת ההגשה במודול.

## שלב 1: הכנת הסביבה

### 1.1 הורדת הקוד מ-GitHub

זהו האפליקציה שעלייכם לפרסום - הורידו את הקוד מגיטהאב.  
לא לשכוח לשמר את פקודות ה CLI הרלוונטיות להורדת קבצים מ GitHub ב commands.txt

[https://github.com/yosefanaliza/streamlit\\_minikube\\_app](https://github.com/yosefanaliza/streamlit_minikube_app)

### 1.2 הפעלת Minikube

לא לשכוח לשמר את פקודות ה CLI הרלוונטיות להפעלת מיניקיוב ובדיקה סטטוס ראשוןית ב commands.txt

## **שלב 2: בניית Docker Image (ניקוד 27%)**

### **2.1 כתיבת Dockerfile עבור האפליקציה**

בגיט יש להשתמש בבראנץ נפרד הכלל את הקובץ Docker וקובץ הסבר פשוט לגביו.  
לא לשכוח לשמר את פקודות ה CLI הרלוונטיות לגיט ב txt.commands.

### **2.2 בניית ה-Image**

לא לשכוח לשמר את פקודות ה CLI הרלוונטיות לבניית התמונה בדוקר ב txt.commands.

### **2.3 בדיקה מקומית**

יש להריץ את הקונטינר בעזרת דוקר ולבזק שכאן ניתן להתחבר אליו ולראות את הדashboard בדף.  
לא לשכוח לשמר את פקודות ה CLI להרצת קונטינר בדוקר וסטוטס ב txt.commands.

### **2.4 העלה ל-Docker Hub**

לא לשכוח לשמר את פקודות ה CLI הרלוונטיות לגישה לדוקר האב - תיוג, דחיפה ווילדייזיה ב txt.commands.

## **שלב 3: פריסת Docker Hub (ניקוד 53%)**

### **3.1 כתיבת קובץ yaml.pod**

בגיט יש להשתמש בבראנץ נפרד הכלל את הקובץ yaml.pod וקובץ הסבר פשוט לגביו.  
לא לשכוח לשמר את פקודות ה CLI הרלוונטיות לגיט ב txt.commands.

### **3.2 כתיבת קובץ yaml.service**

בגיט יש להשתמש בבראנץ נפרד הכלל את הקובץ service.yaml וקובץ הסבר פשוט לגביו.  
לא לשכוח לשמר את פקודות ה CLI הרלוונטיות לגיט ב txt.commands.

## **שלב 4: הרצת האפליקציה ב-Minikube (ניקוד 15%)**

### **4.1 יישום (app) הקבצים**

לא לשכוח לשמר את פקודות ה CLI הרלוונטיות ליישום הקבצים ב txt.commands.

### **4.2 בדיקת סטוטוס**

לא לשכוח לשמר את פקודות ה CLI הרלוונטיות לבדיקה סטוטוס ב txt.commands.

### **4.3 גישה לאפליקציה**

בגיט יש להשתמש בבראנץ נפרד הכלל את הקובץ screenshot.png (כולל ה- URL בשורת הכתובת)  
לא לשכוח לשמר את פקודות ה CLI הרלוונטיות להרצת האפליקציה ולגיט ב txt.commands.

## שלב 5: בדיקות נוספות (ניקוד 5%)

### 5.1 צפיה בלוגים

לא לשכוח לשמר את פקודות הרלוונטיות לבדיקת לוגים ה CLI ב txt.commands

### 5.2 מידע מפורט על ה-Pod

לא לשכоч לשמר את פקודות ה CLI לקבלת מידע מפורט על ה POD ב txt.commands

## טיפים חשובים

- שימושו לב לשם המשתמש ב-Docker Hub - תווודאו שהוא תואם בכל מקום
- בדקו **syntax** - טעויות קטנות ב-YAML יכולות לגרום לכשלים
- השימושו ב-**kubectl describe** - זה עוזר לזהות בעיות
- קראו הودעות שגיאה Kubernetes נותן הודעות שגיאה מפורטות
- אל תשחחו את **--server.address=0.0.0.0** - בלעדיו Streamlit לא יהיה נגיש מחוץ ל-container (המושג נקרא bind address והוא מציין מאייזה IP ניתן לגשת לשרתת | 0.0.0.0 מאפשר גישה מכל IP)

פקודת ההרצה עם הקונפיגורציה הנ"ל:  
`streamlit run your_app.py --server.port 8080 --server.address=0.0.0.0`

## שאלות נפוצות

### ש: ה-Pod תקוע ב-ImagePullBackOff

ת: בדקו ש-image name נכון ושהעליתם אותו לדocker Hub. נס Pod kubectl describe pod pod-name לפרטיטם.

### ש: לא מצליח להיכנס לאפליקציה

ת: ודאו שה-Service מסוג NodePort והשתמשו ב-**minikube service streamlit-service** URL -- לקבלת ה-URL הנכון.

### ש: האפליקציה לא מציגה נתונים

ת: בדקו את הלוגים של ה-Pod עם **kubectl logs <pod-name>**

בצלחה!