**DevOps**

**פרויקט סוף**

שמות מגישים:

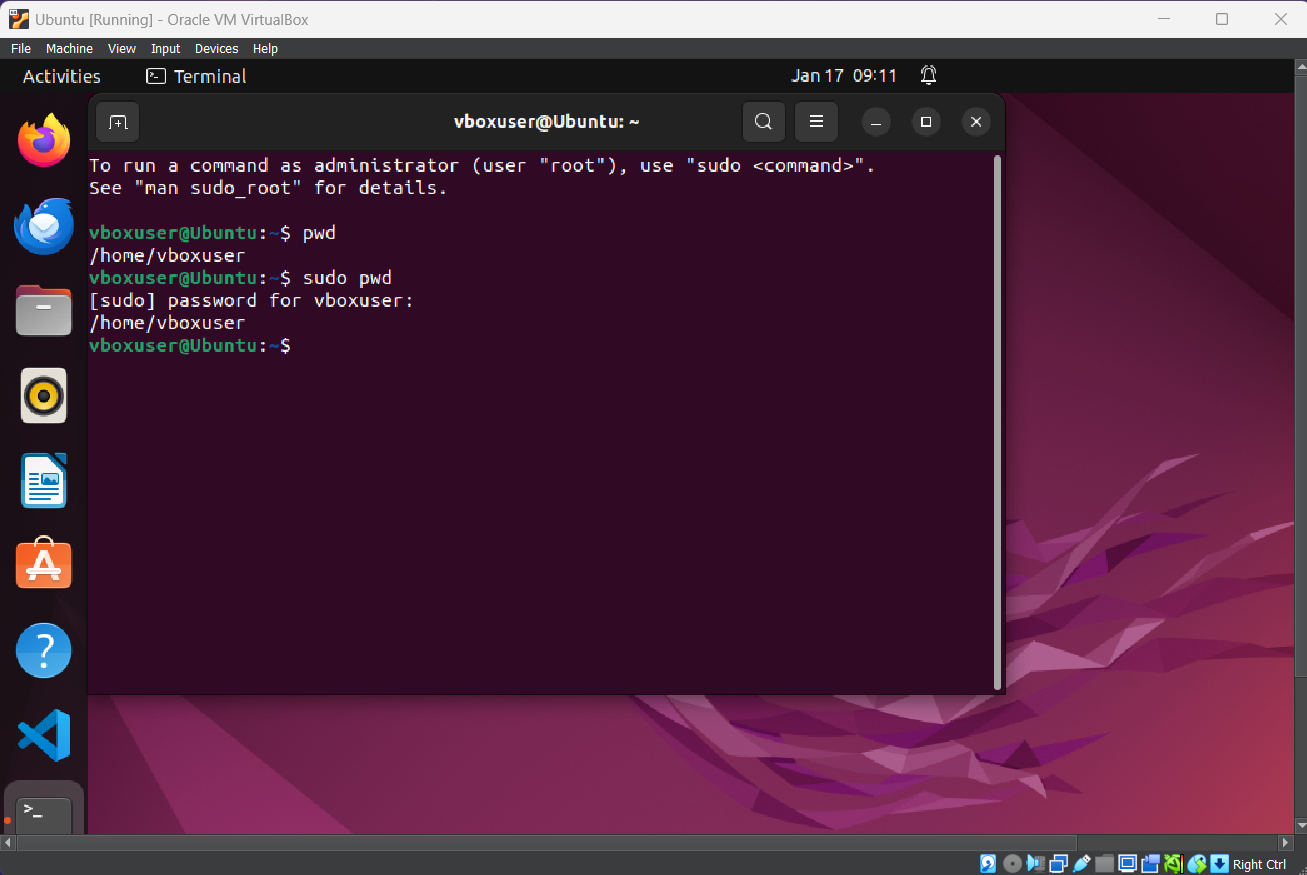
שם: עומר זומרשטיין שם: יוני אזרף

ת.ז: 316439876 ת.ז: 209459239

טלפון: 054-429-9999 טלפון: 052-648-8924

אימייל: [omers354@gmail.com](mailto:omers354@gmail.com) אימייל: [yoni4600@gmail.com](mailto:yoni4600@gmail.com)

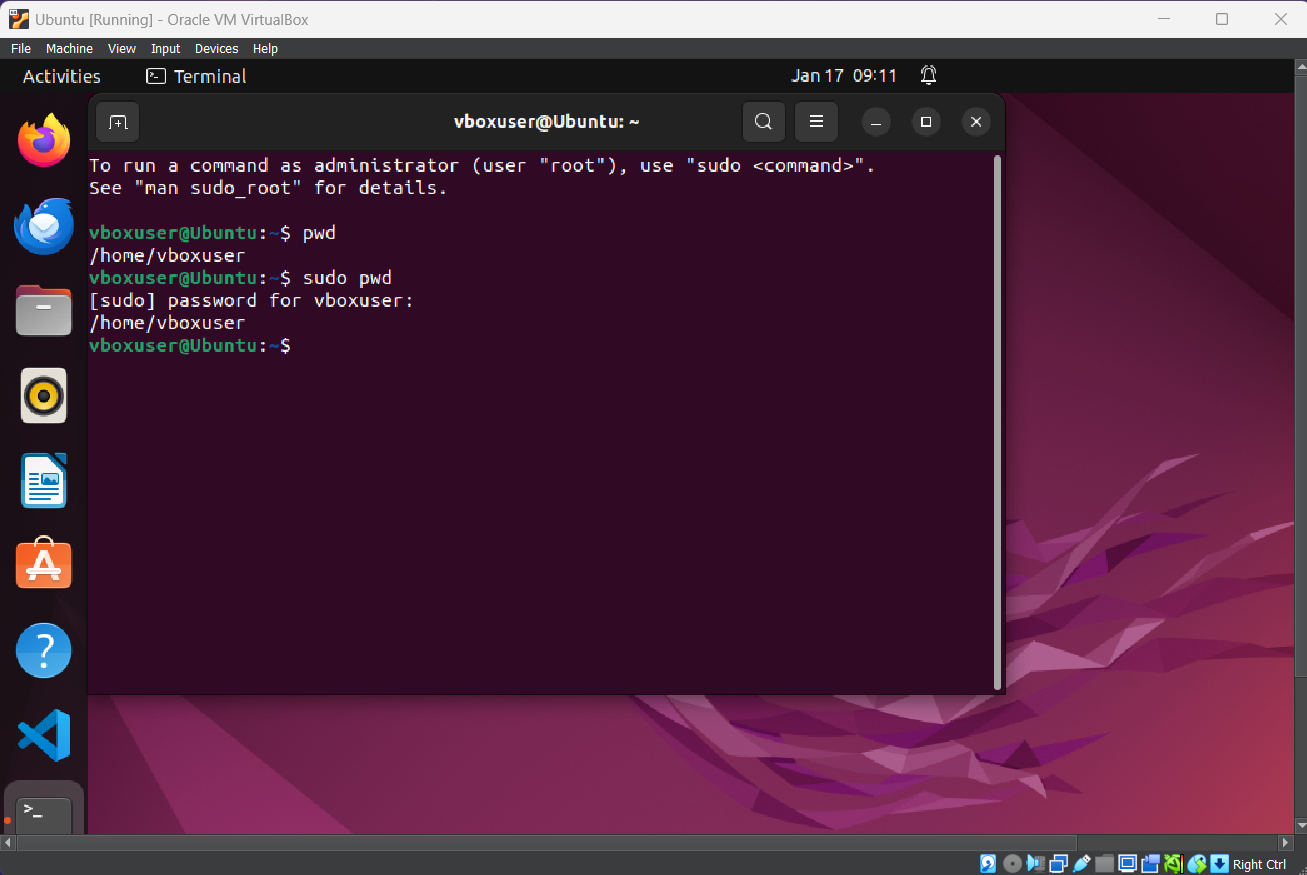
הרצאה 1: Introduction – שקף 34



הפקודה pwd הורצה – ניתן לראות כי מקבלים חזרה את התיקייה עליה אנו נמצאים כרגע.

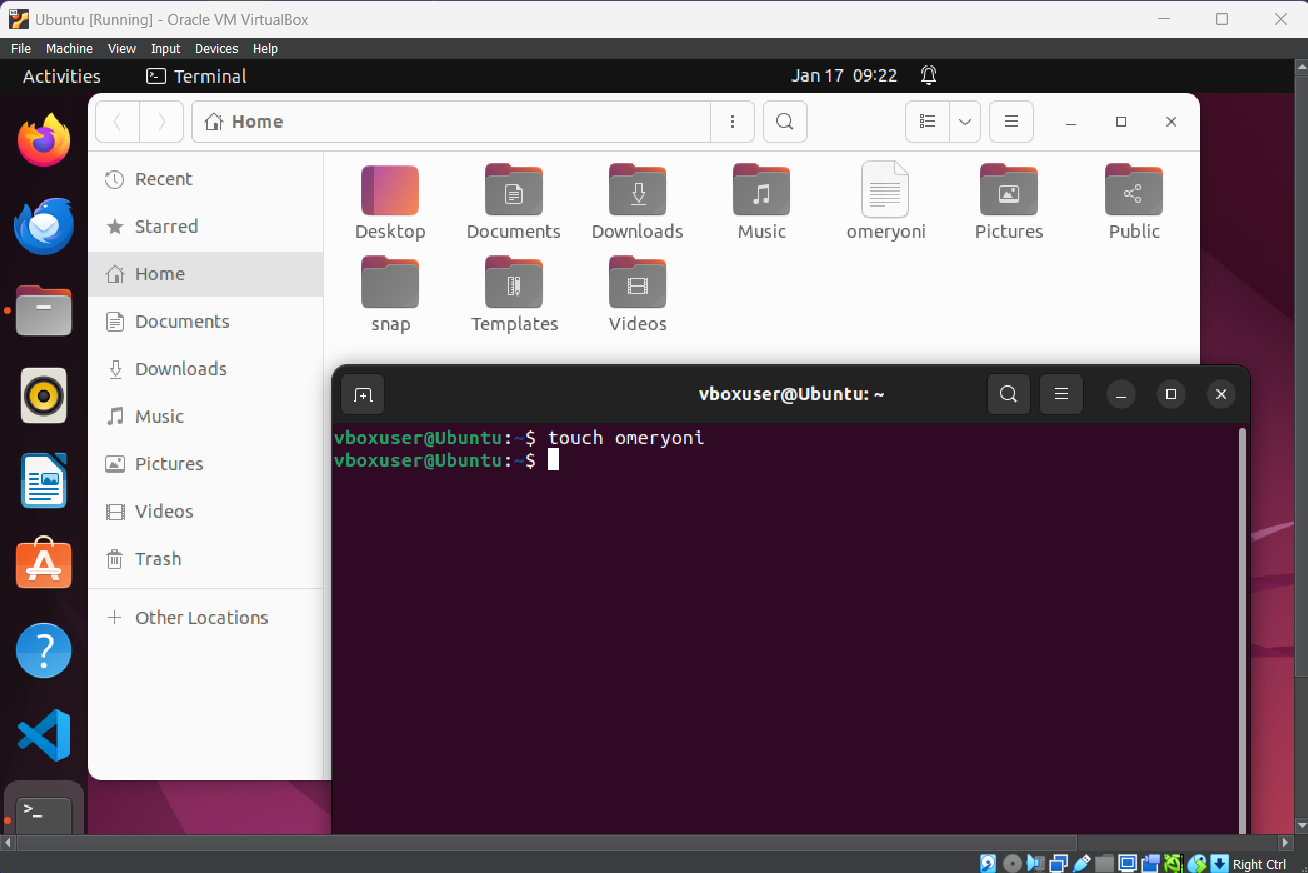
הרצאה 2: Linux – שקף 10,26,35

שקף 10



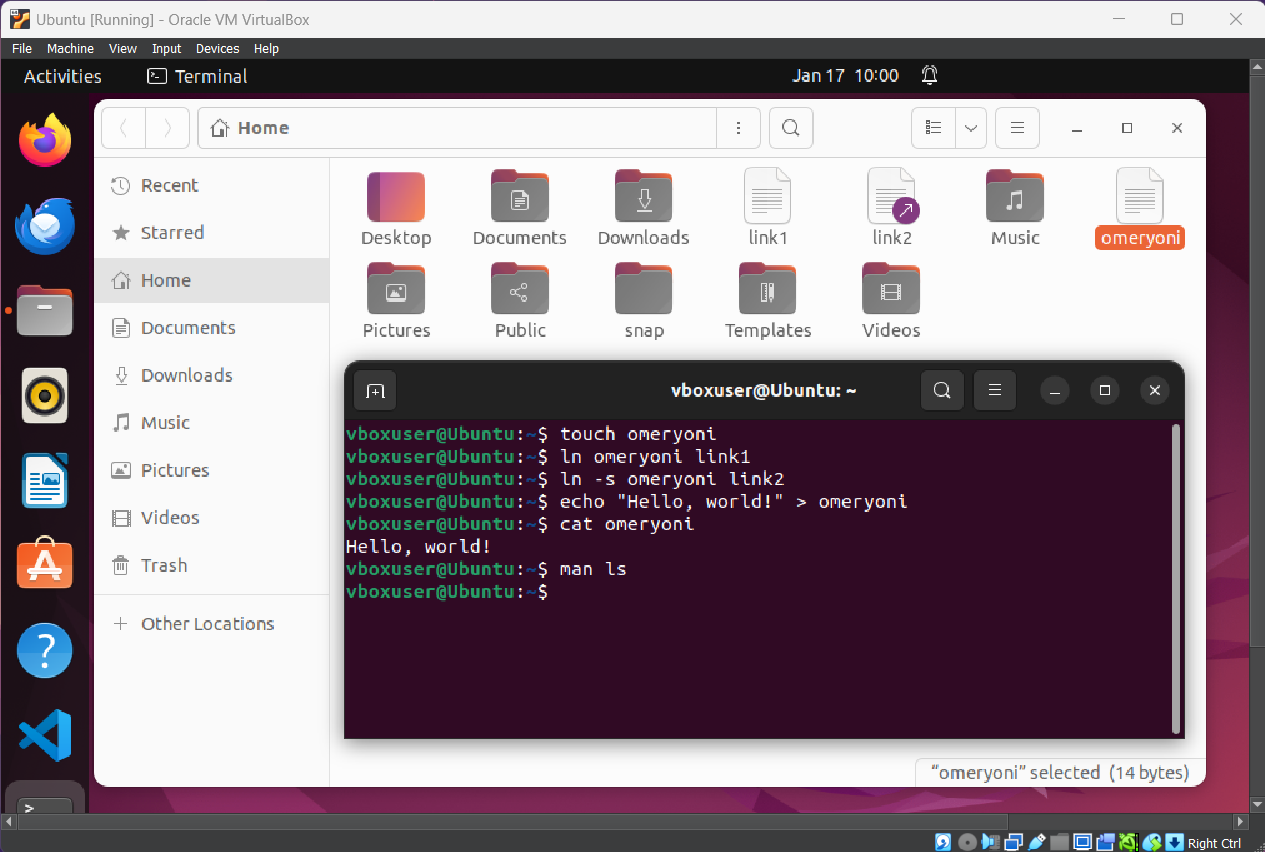
1. ה- Username הוא: vboxuser.
2. ה- Machine Name הוא: Ubuntu.
3. התיקייה הנוכחית היא: /home/vboxuser

שקף 26

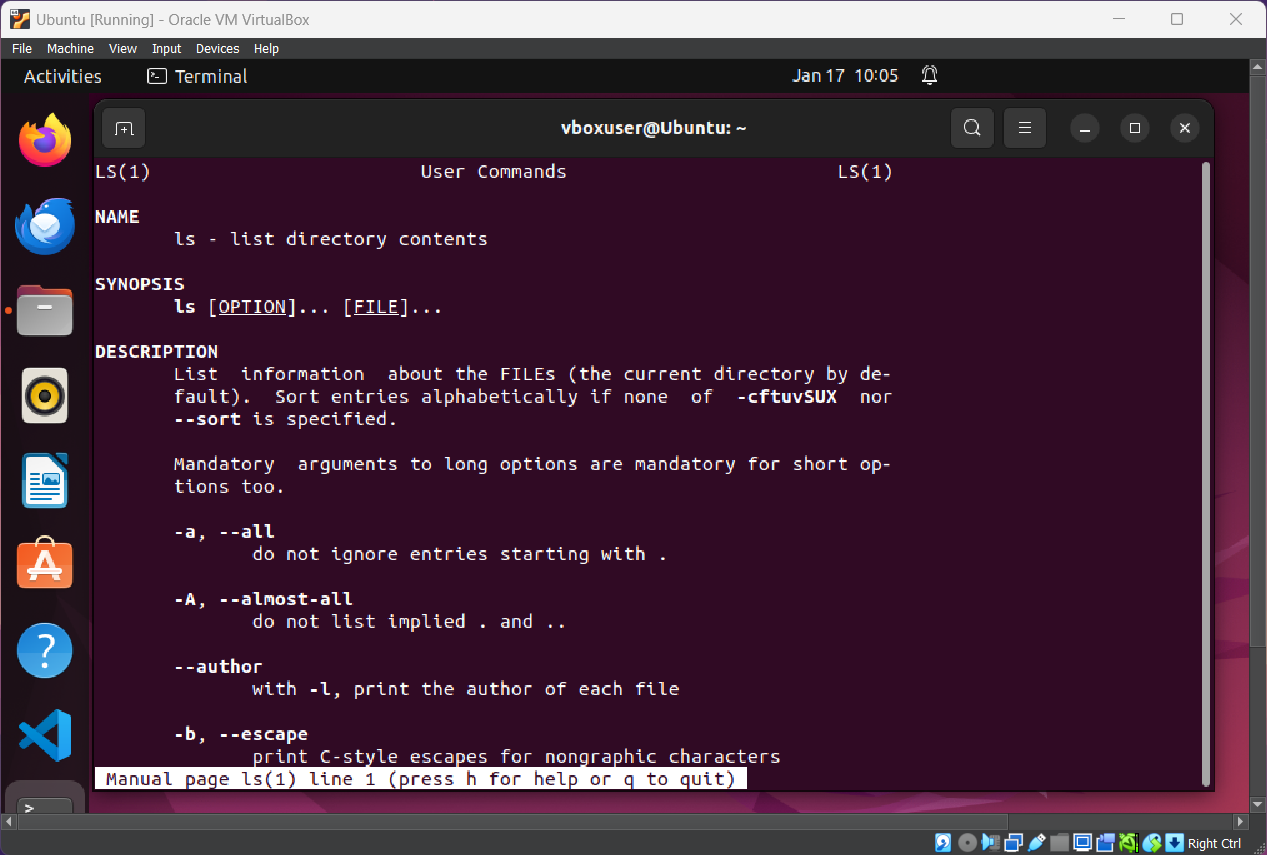


כפי שניתן לראות, נוצר קובץ חדש בשם omeryoni.

שקף 35



1. תחילה יצרנו את הקובץ omeryoni.
2. לאחר מכן השתמשנו בפקודה ln כדי ליצור לינקים בין קבצים.
3. Link1 – Hard Link
4. Link2 – Symbolic Link – מצביע לשם קובץ.
5. יצרו את הסטרינג “Hello, world!”, ושמנו אותו בקובץ שיצרנו (omeryoni).
6. ביצענו את הפקודה cat, ואז קיבלנו חזרה את התוכן בתוך הקובץ.
7. הפקודה man מגיעה מהמילה manual, כלומר הוראות לפקודות אחרות (ניתן לראות את ההוראות לדוגמה בתמונה הבאה).

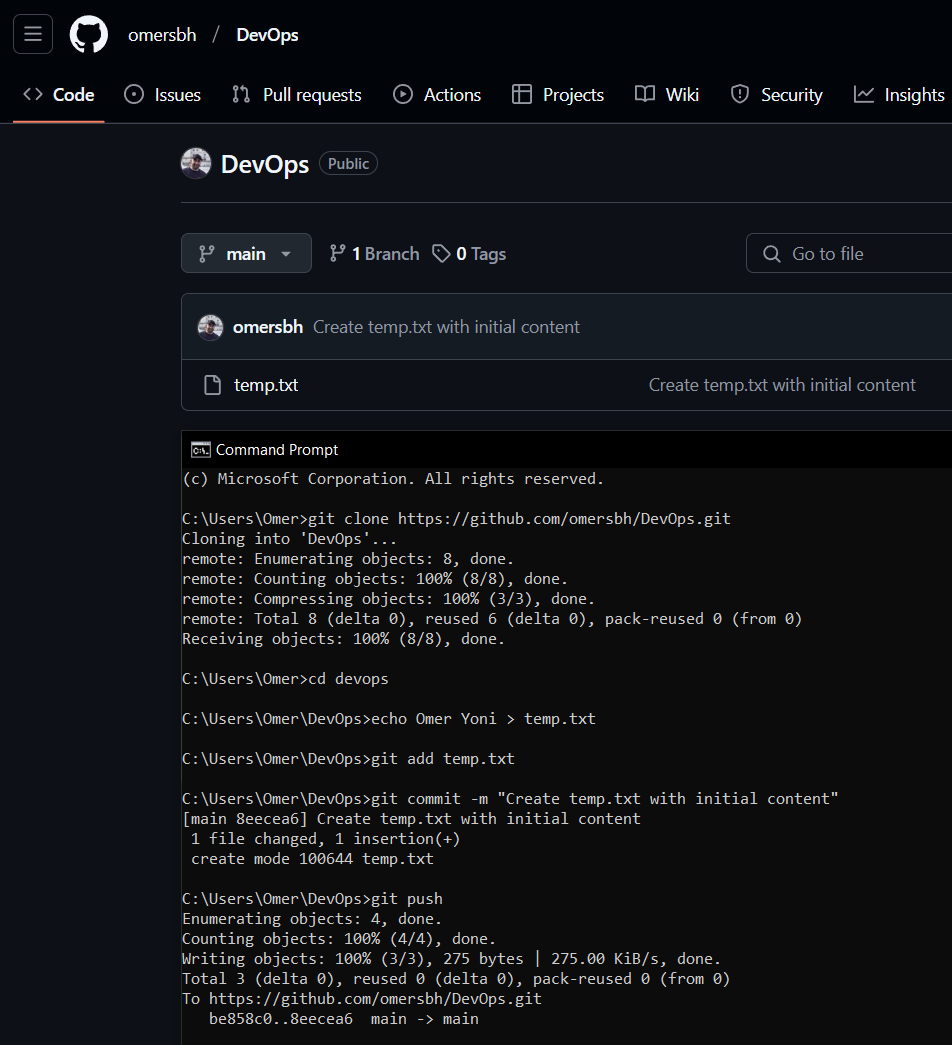


פקודת man – manual.

הרצאה 3: git – שקף 46

1. נבצע פעולות CRUD:

Create:

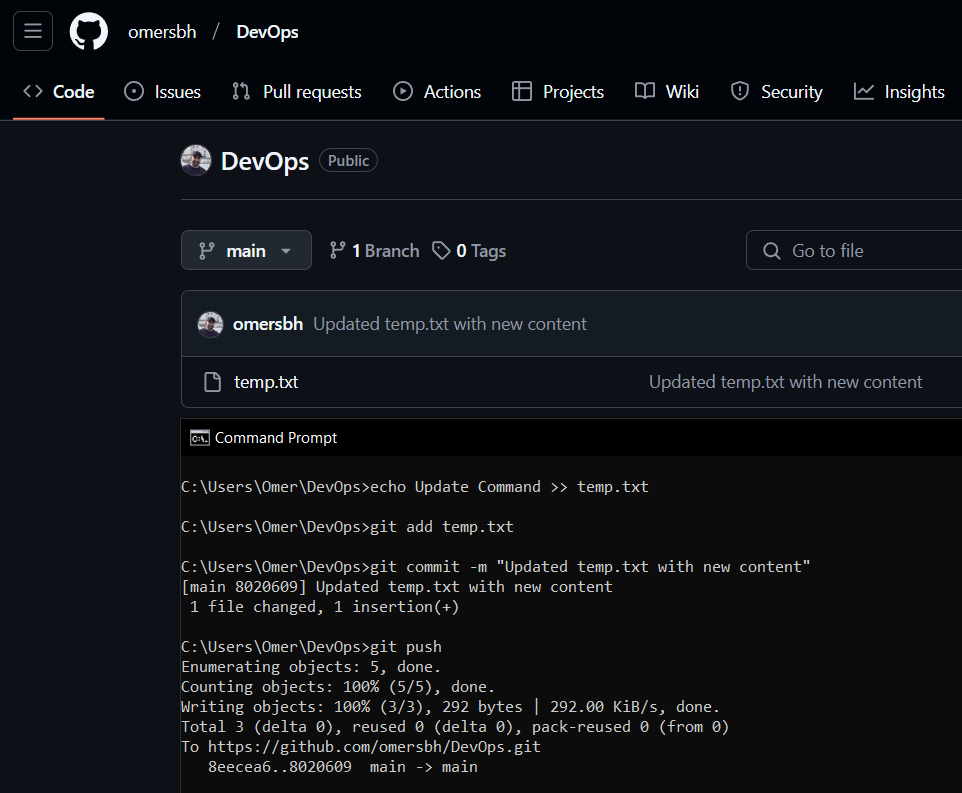


Read:

A screen shot of a computer

Description automatically generated

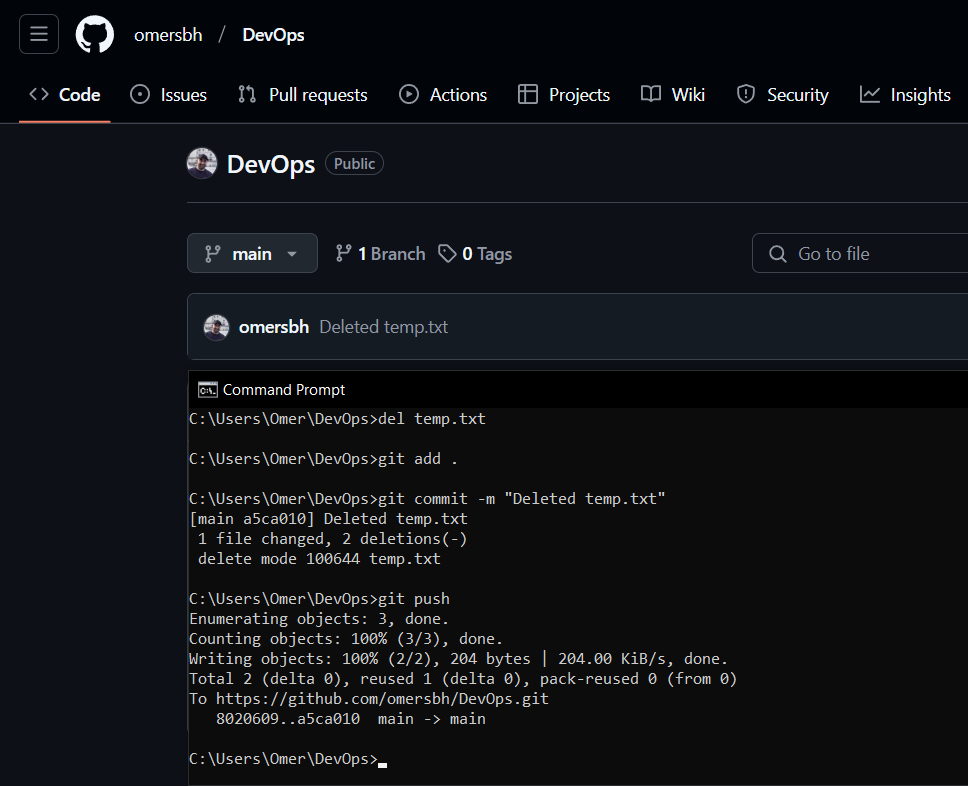
Update:



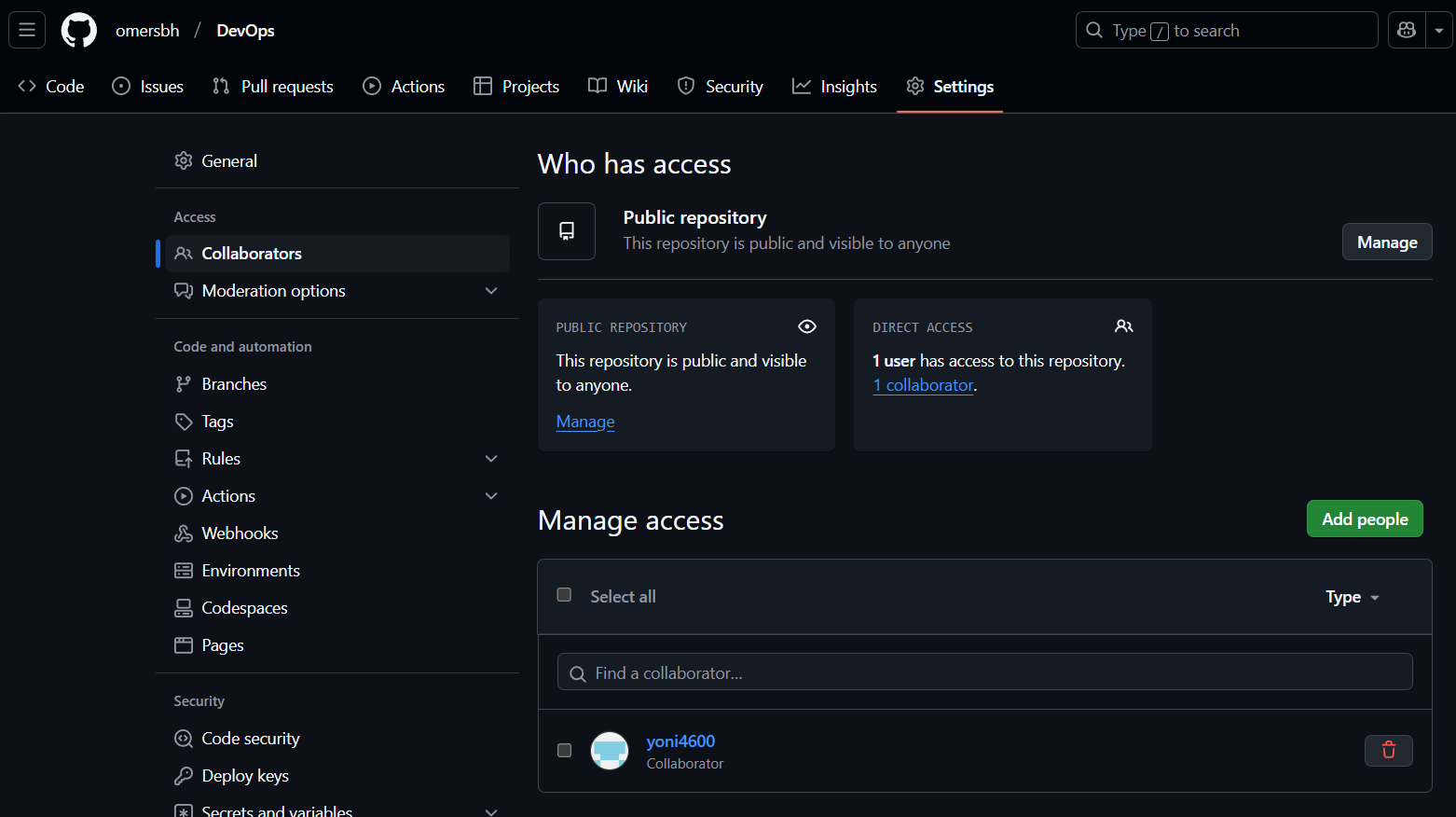
A black screen with white text

Description automatically generated

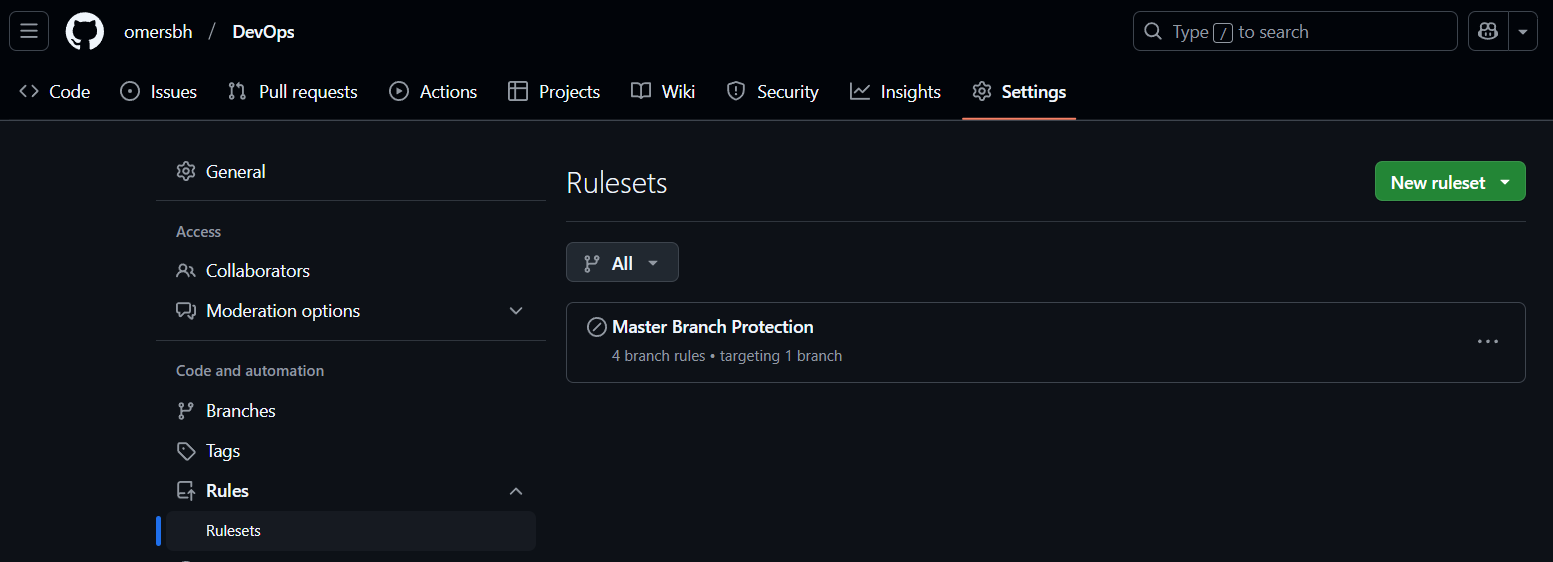
Delete:



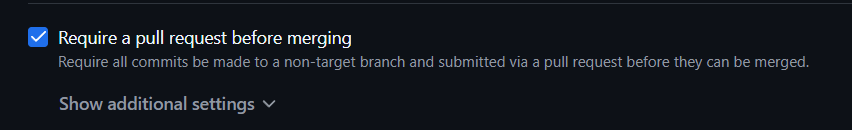
1. כעת, נוסיף את יוני לגיטהאב:



2 משתמשים צריכים לאשר Pull Request:



לדוגמה, אחד מה- Branch rules שהגדרנו:



1. Dependabot הוא כלי שמעדכן אוטומטית תלויות בפרויקט שלך ב-:GitHub
2. למה צריך אותו?

* אבטחה: מעדכן גרסאות עם תיקוני אבטחה.
* חסכון בזמן: אוטומטי, חוסך זמן בניהול עדכונים.
* אחריות: מונע בעיות תאימות בין תלויות.

1. מתי משתמשים בו?

* בפרויקטים שמכילים תלויות (כמו לדוגמה Node.js, Python).
* אם יש צורך בשמירת עדכונים שוטפים.

1. למה הוא חשוב?

* מונע בעיות אבטחה.
* אוטומטי, שומר על עקביות.
* מקטין את הצורך בניהול ידני של עדכונים.

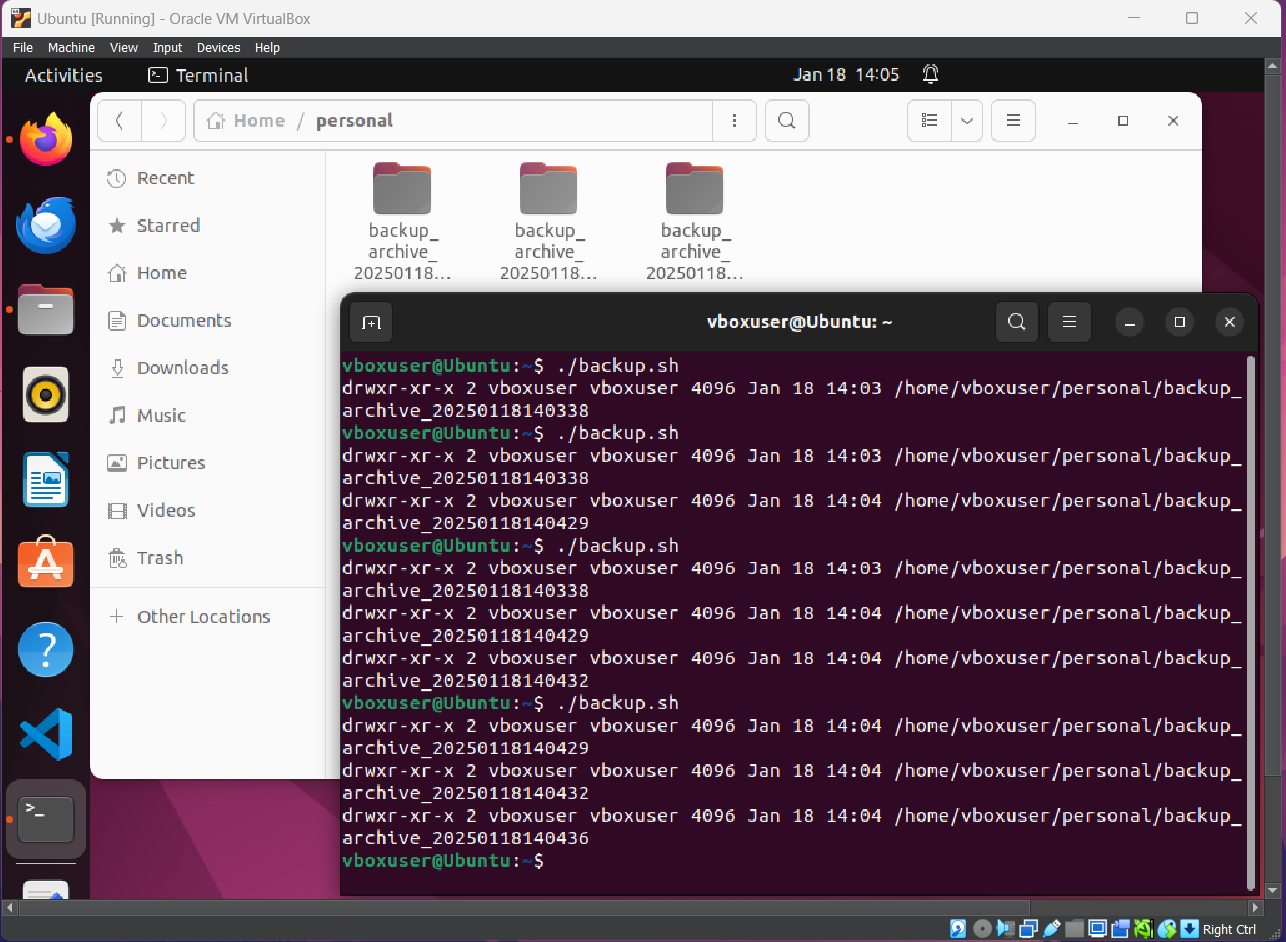
הרצאה 4: Ansible – שקף 47

Script:

A screenshot of a computer

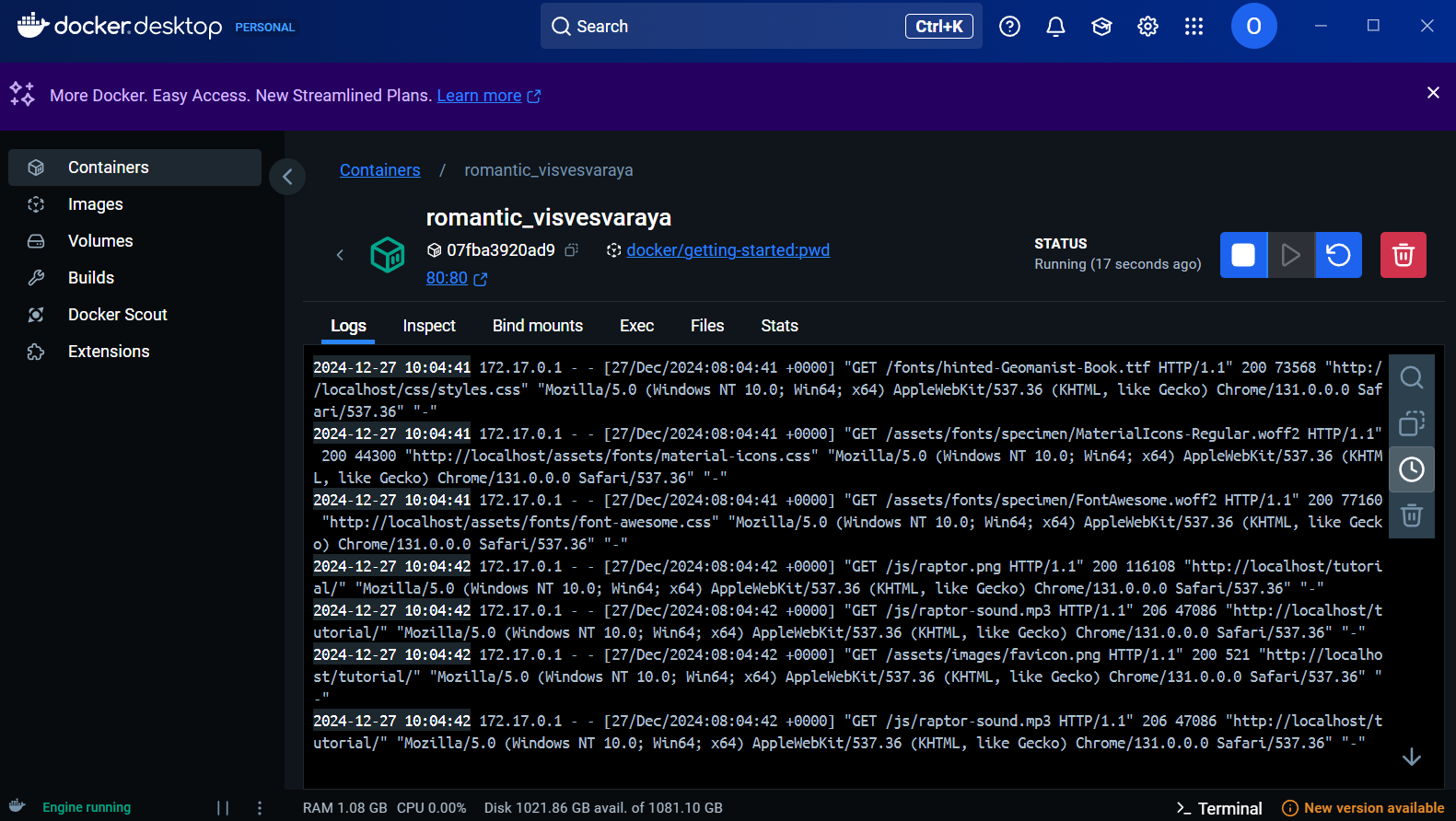
Description automatically generated

הרצה:

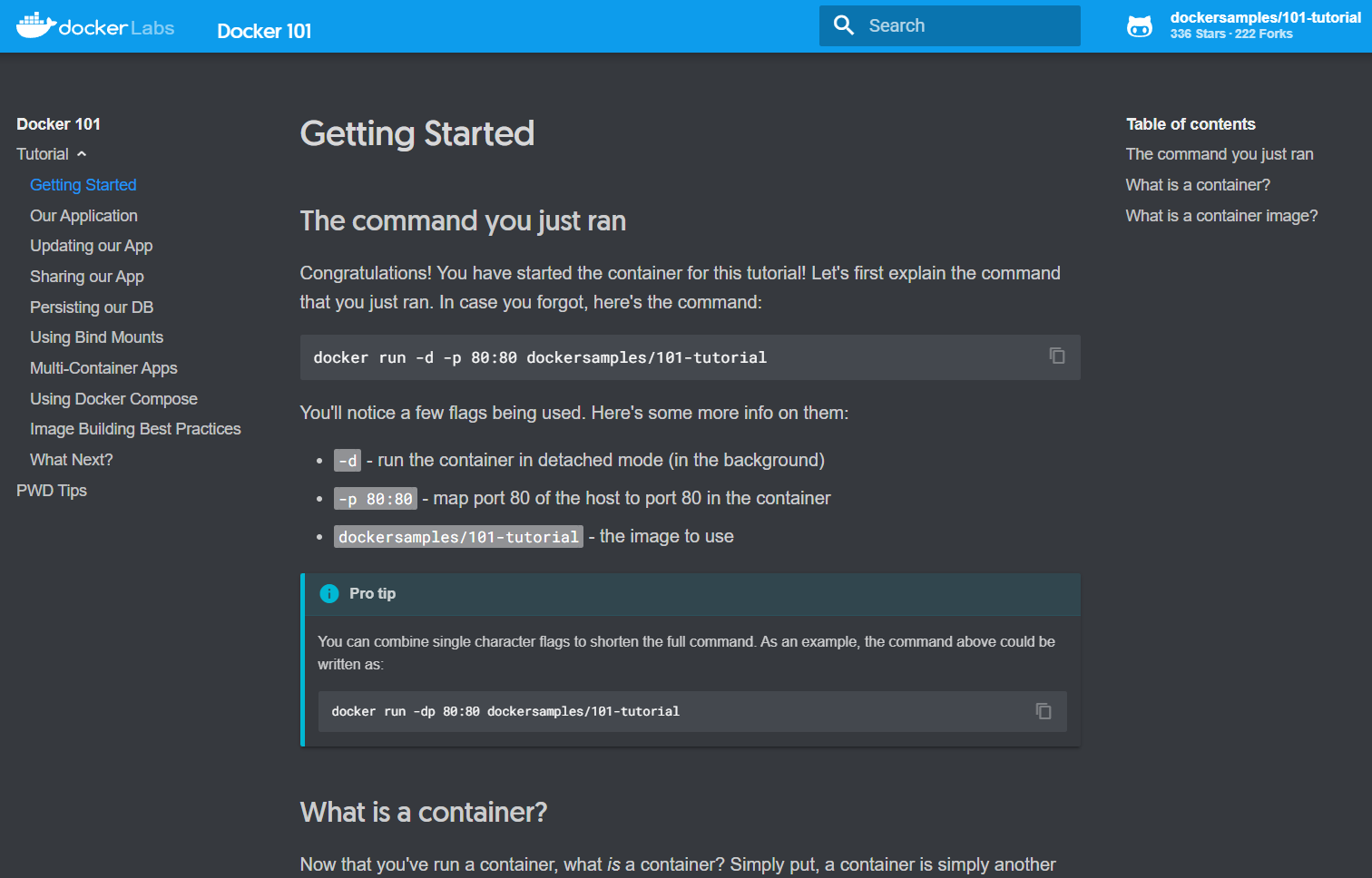


הרצאה 5: Docker – שקף 29,33,45

שקף 29

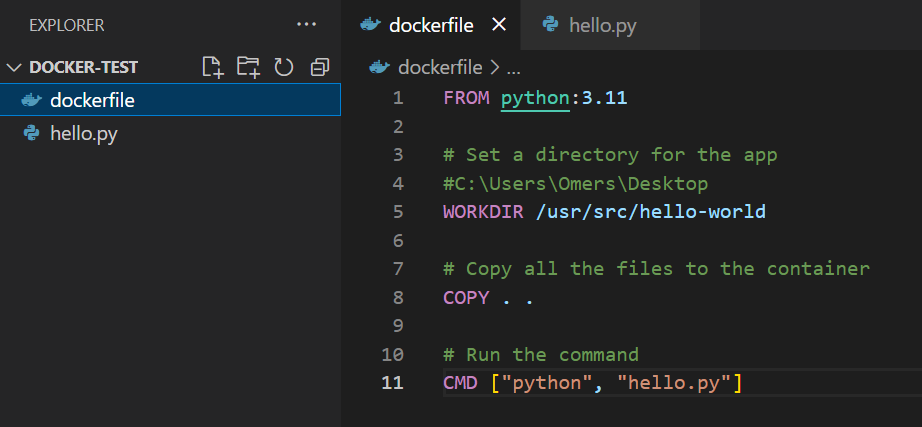


שקף 33:



שקף 45:

Visual Studio Code:



A screenshot of a computer

Description automatically generated

כפי שניתן לראות יצרנו קוד dockerfile פשוט, וגם קובץ hello.py עם פעולת הדפסה פשוטה שבהמשך נראה איך נריץ.

Image:

A screenshot of a computer

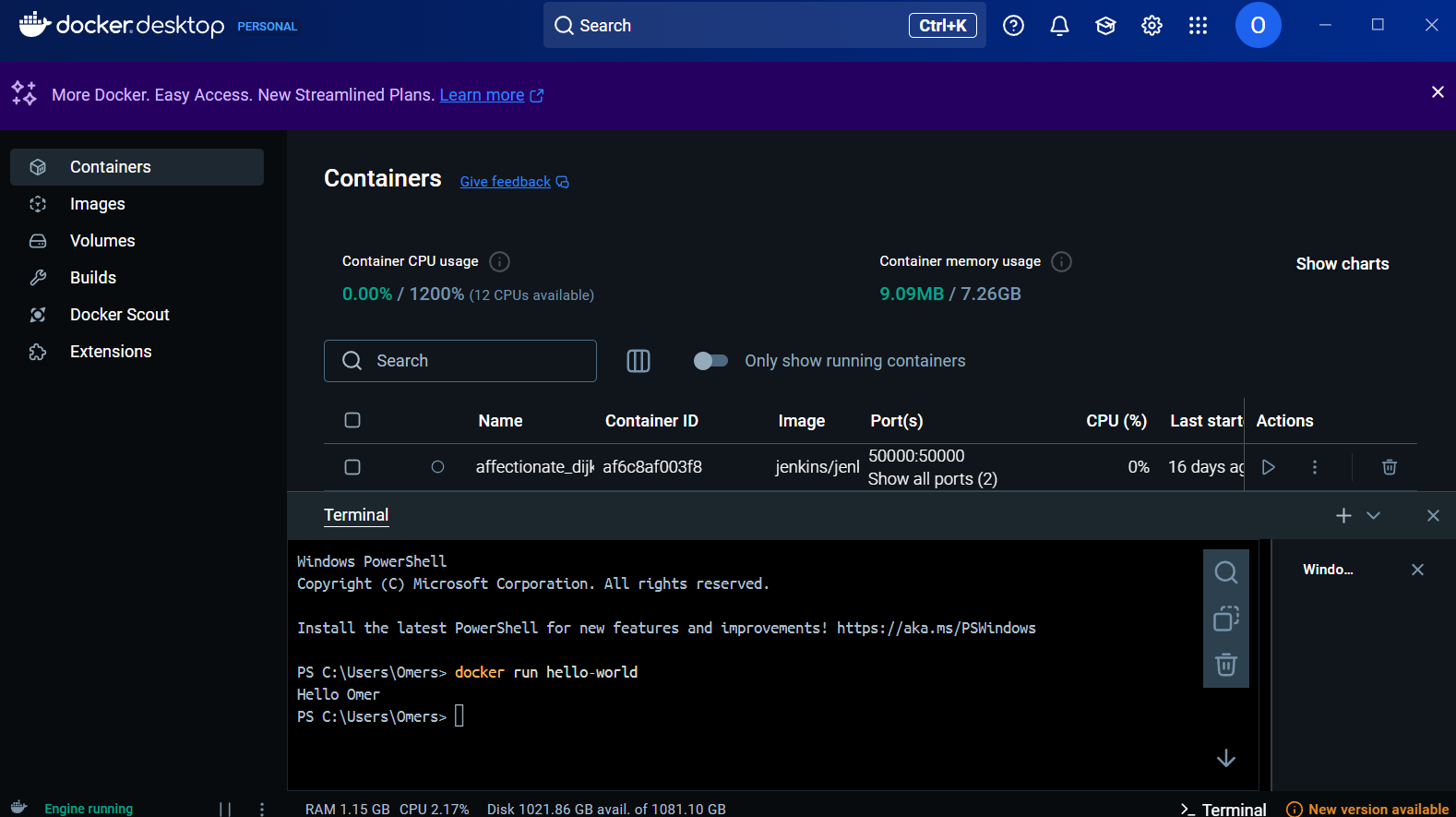
Description automatically generated

Build dockerfile:

A screenshot of a computer

Description automatically generated

Run in terminal:

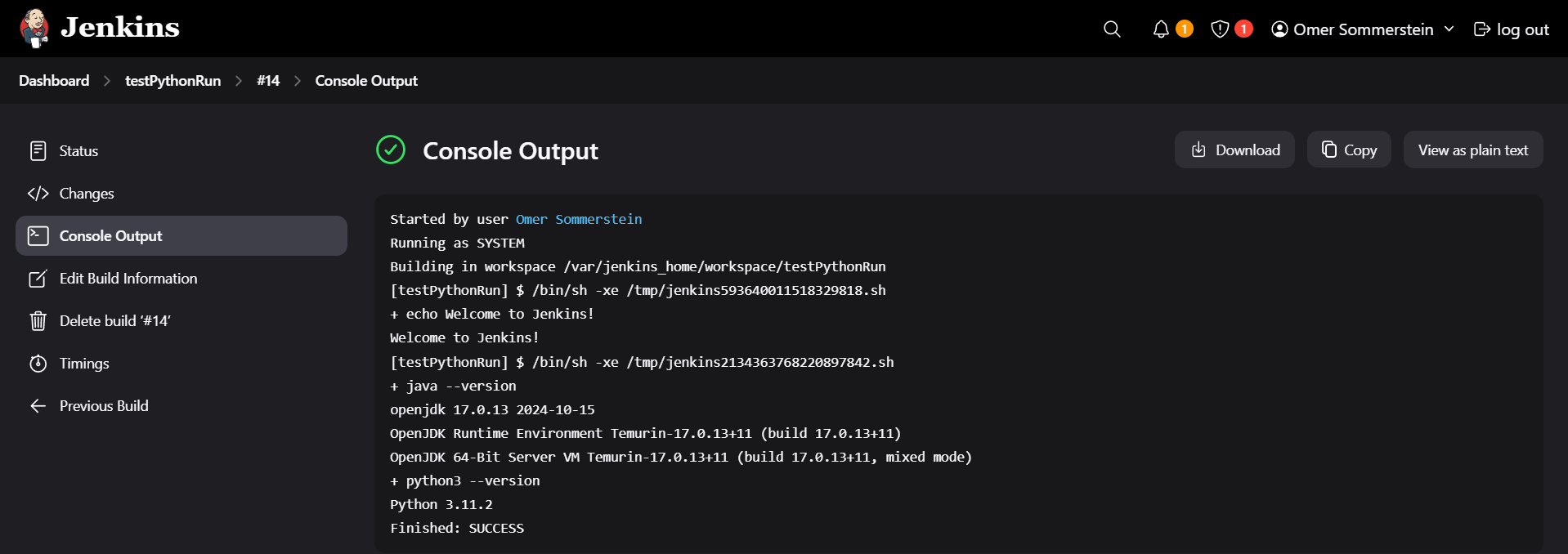


הרצת הקובץ hello-world כפי שראינו לעיל, שנותן את הפלט בתמונה הנ"ל.

הרצאה 6: Jenkins – שקף 19,20

1. Print welcome message:

2. Installs python & verify python installation with “python –version”:



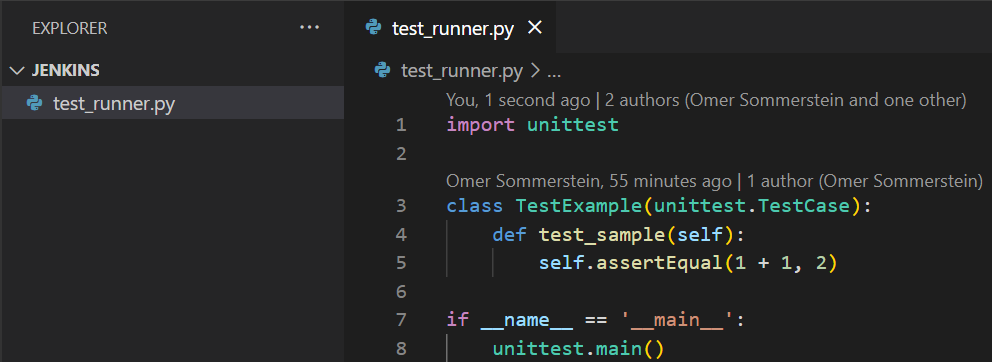
3. Check-out your repository from master branch:

A screenshot of a computer

Description automatically generated

יצרנו חשבון לגיט וביצענו PUSH של הקוד DUMMY.

4. Runs unit-tests on your code:

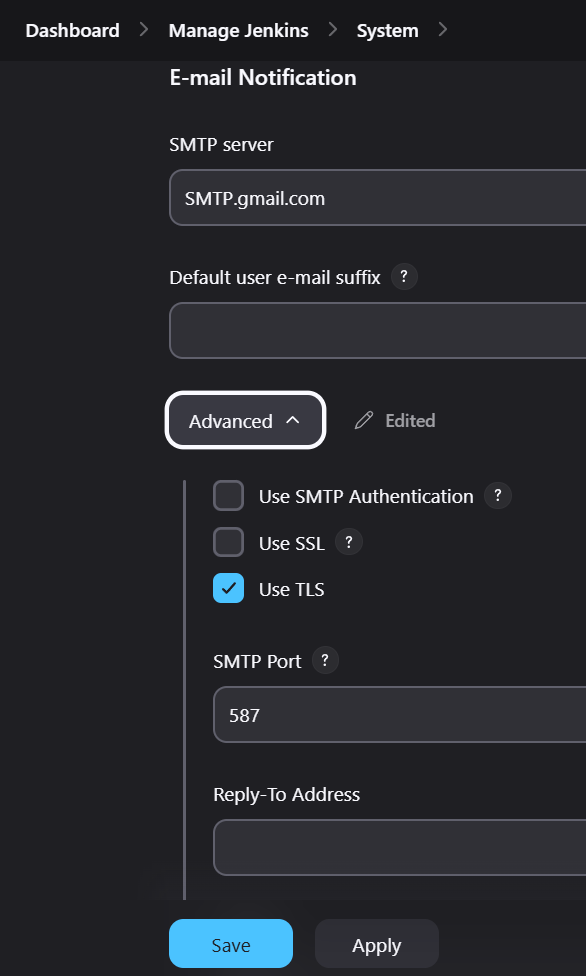


בקוד לעיל ניתן לראות כי בנינו UNIT TEST פשוט שבודק אם 1+1 ו- 2 הם שווים.

5. Sends email for every run result:

A screenshot of a computer

Description automatically generated



מכיוון שלא ניתן לשלוח מיילים מטעמי אבטחה של ג'ימייל, היינו צריכים להגדיר זאת ב- Jenkins ולספק מפתח ייחודי מג'ימייל לטובת ג'נקינס, דבר שלמעשה גוגל טוענים בעצמם שלא כדאי לעשות, אך זה אפשרי. לטעמי בדיקה ביצענו זאת.

A screenshot of a computer

Description automatically generated

הגדרנו את האימייל אליו ישלחו התוצאות.

A screenshot of a computer

Description automatically generated

הרצה, ניתן לראות שקיבלנו הודעה שזה הצליח.

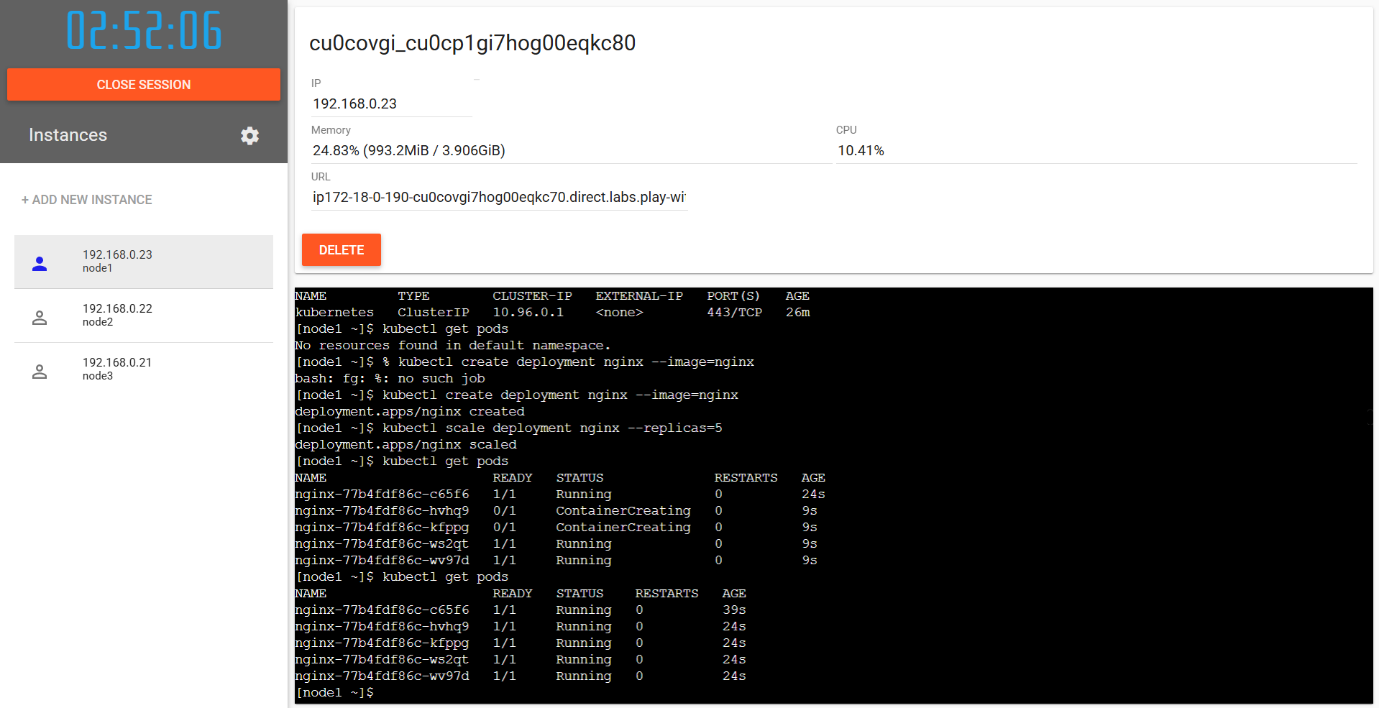
A screenshot of a computer

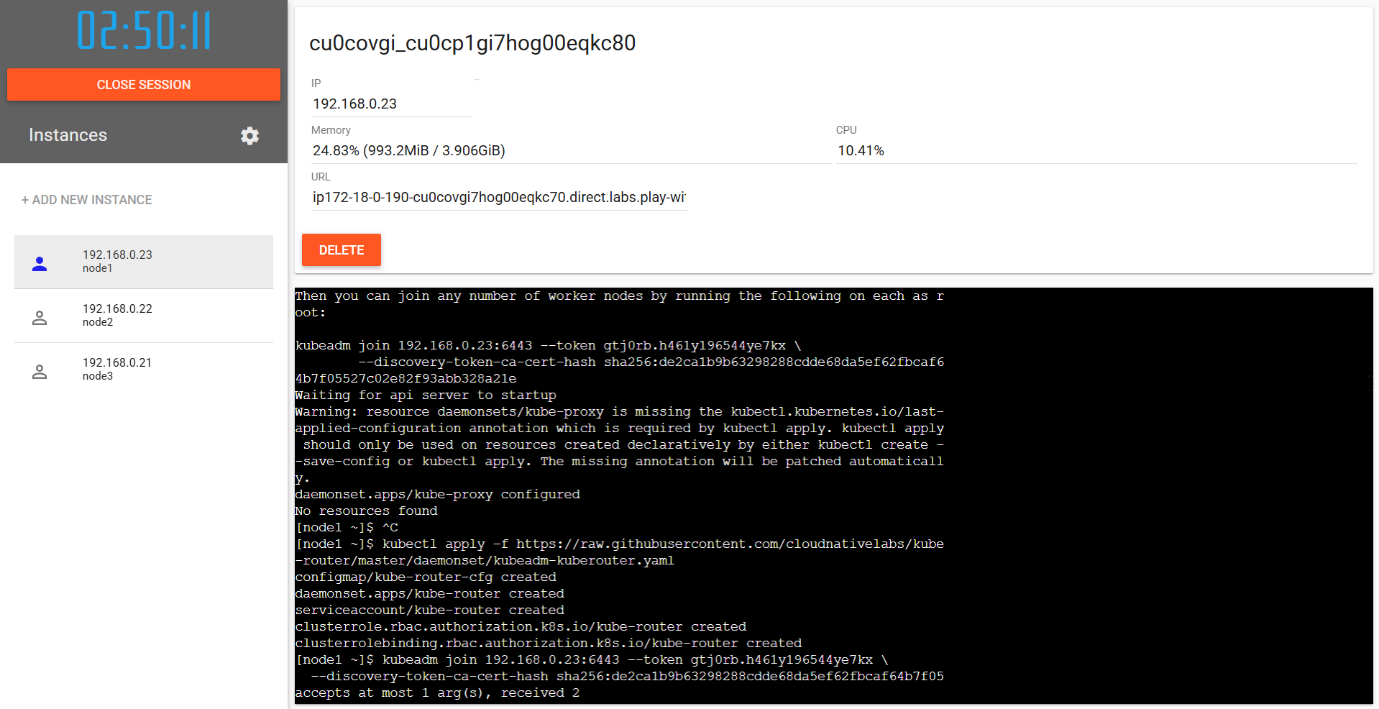
Description automatically generated

קבלת אימייל מ- Jenkins.

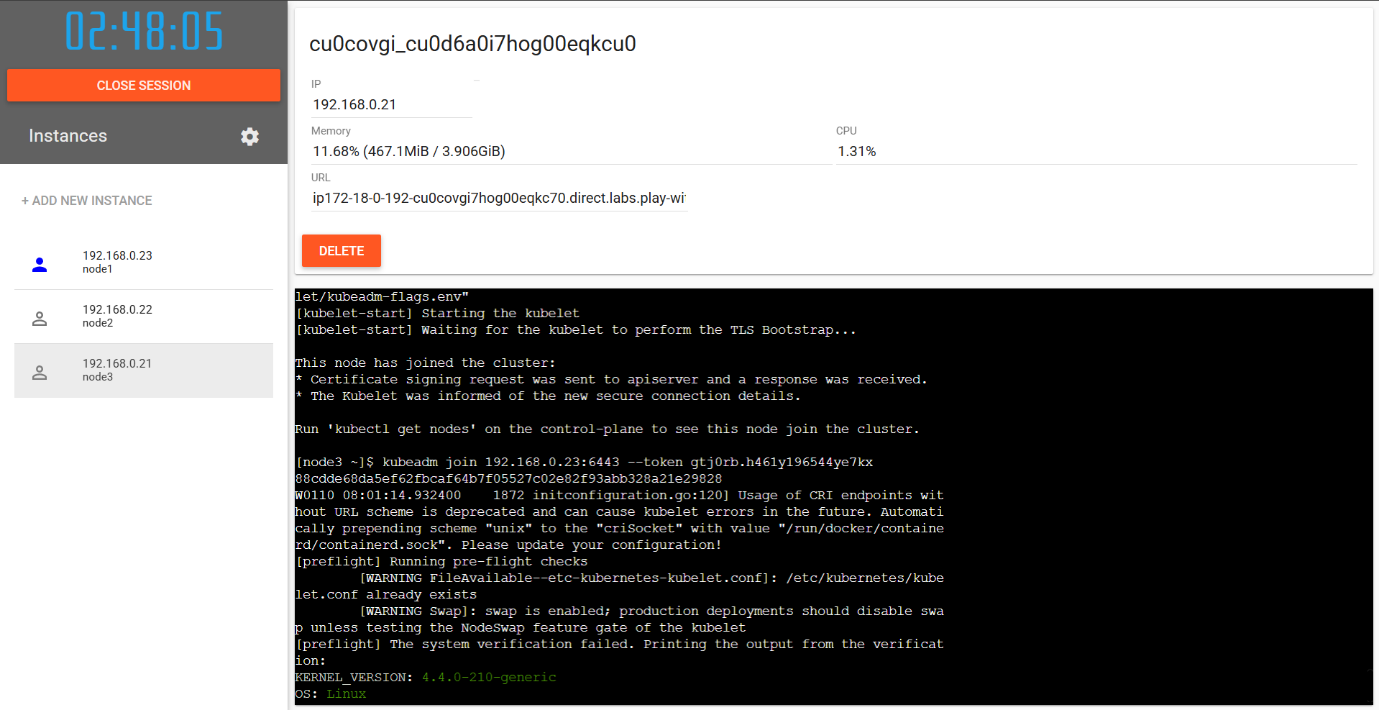
הרצאה 7: Kubernetes – שקף 31,50,51

שקף 31





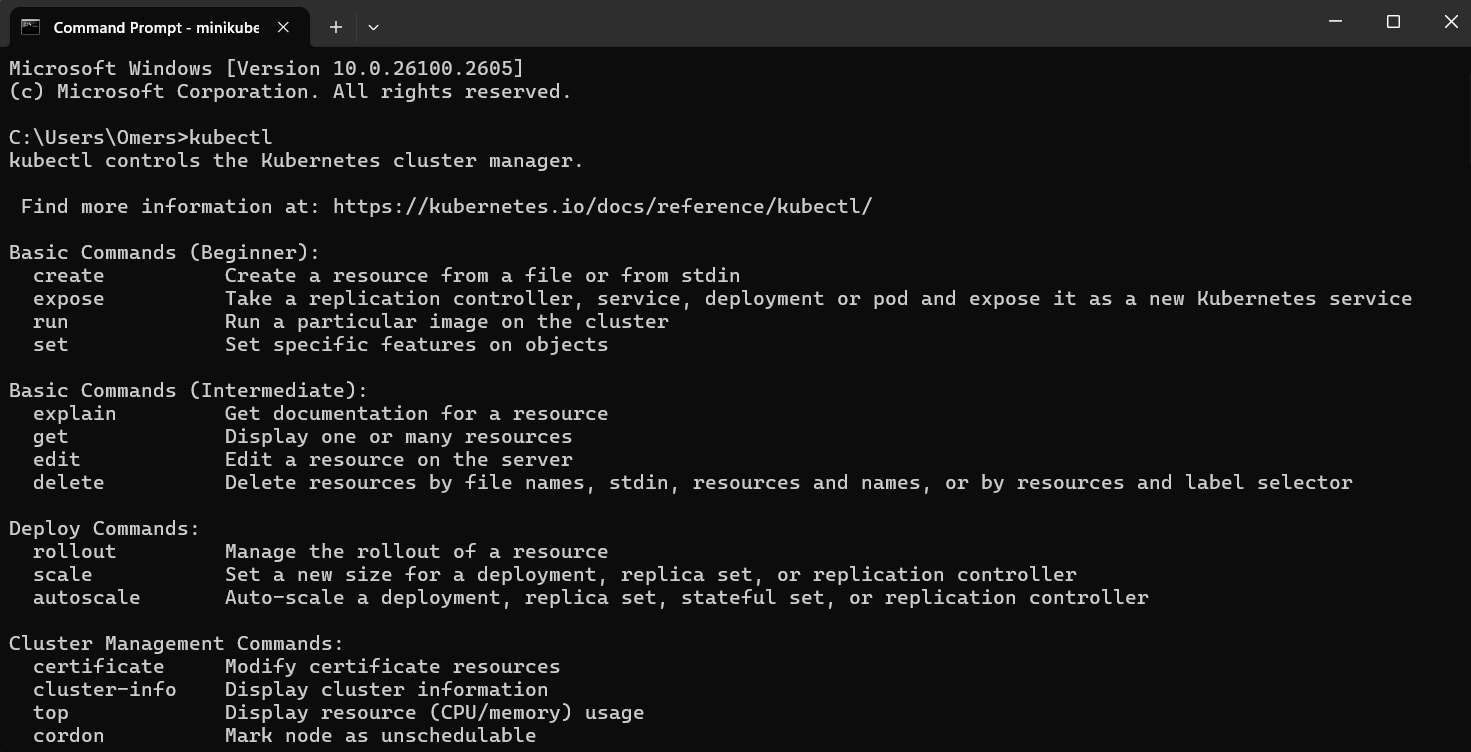






ניתן לראות כי יצרנו Cluster בהצלחה והוספנו 3 worker nodes כנדרש.

שקף 50,51



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

ביצענו את הקודים משקפים קודמים כדי לוודא שההתקנה בוצעה כנדרש וניתן לראות שהכל עובד כראוי.

A screenshot of a computer

Description automatically generated

איתחול minikube.

A screenshot of a computer

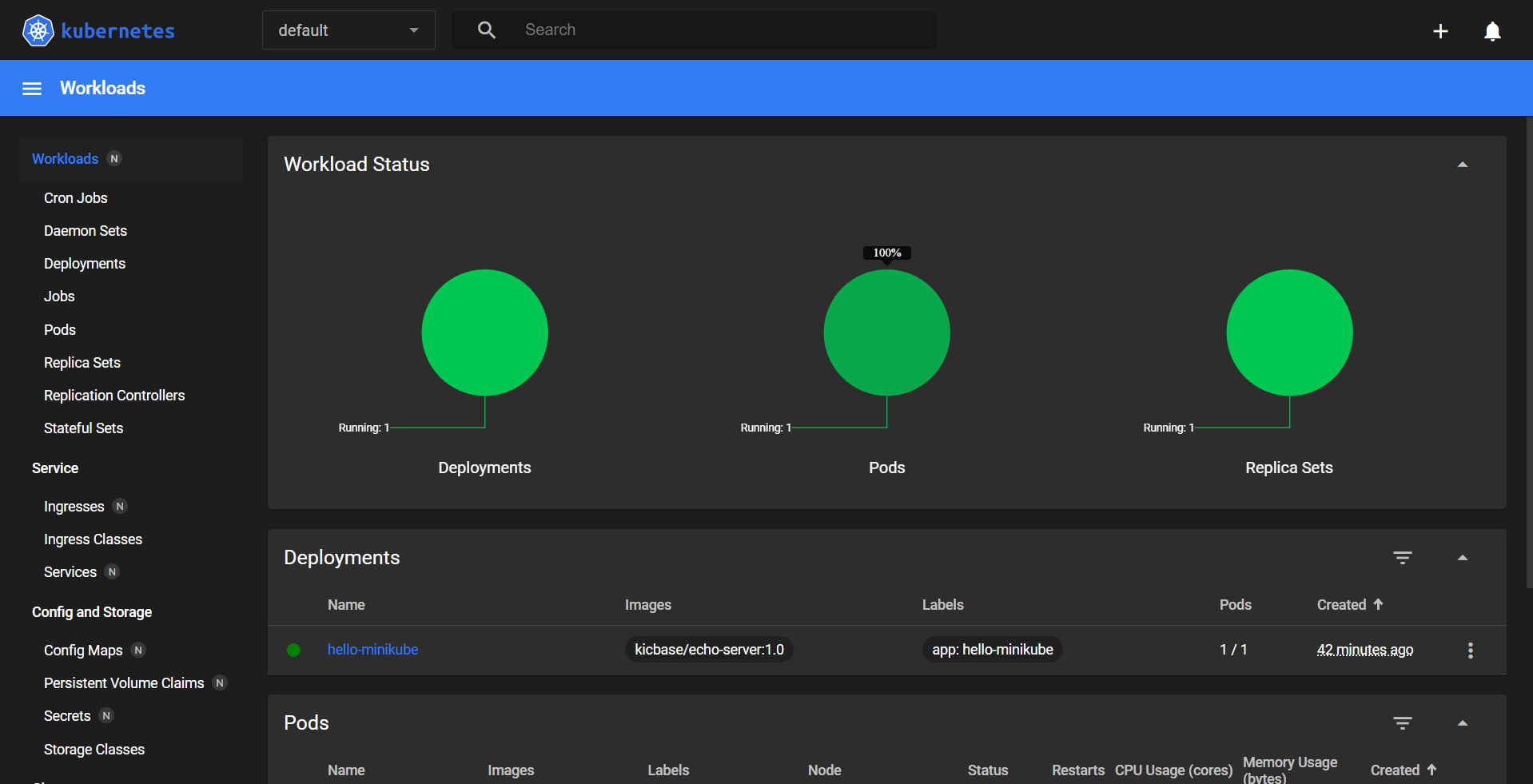
Description automatically generated

פקודת get nodes בוצעה ופקודת בדיקת גרסה בוצעה. בנוסף, פקודה אחרונה נותנת לנו רשימה של כל הpods ב- namespaces בתוך ה- Kubernetes cluster.

A computer screen with white text

Description automatically generated

הרצת הדאשבורד, מוביל אותנו לאתר בתמונה הבאה:

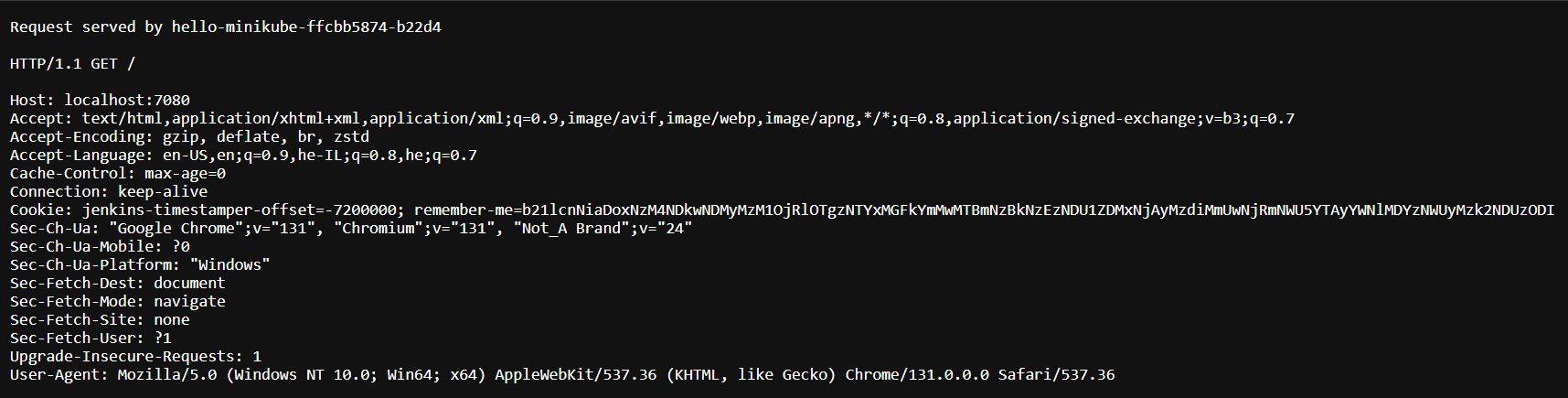


זה האתר שנפתח לנו של Kubernetes.

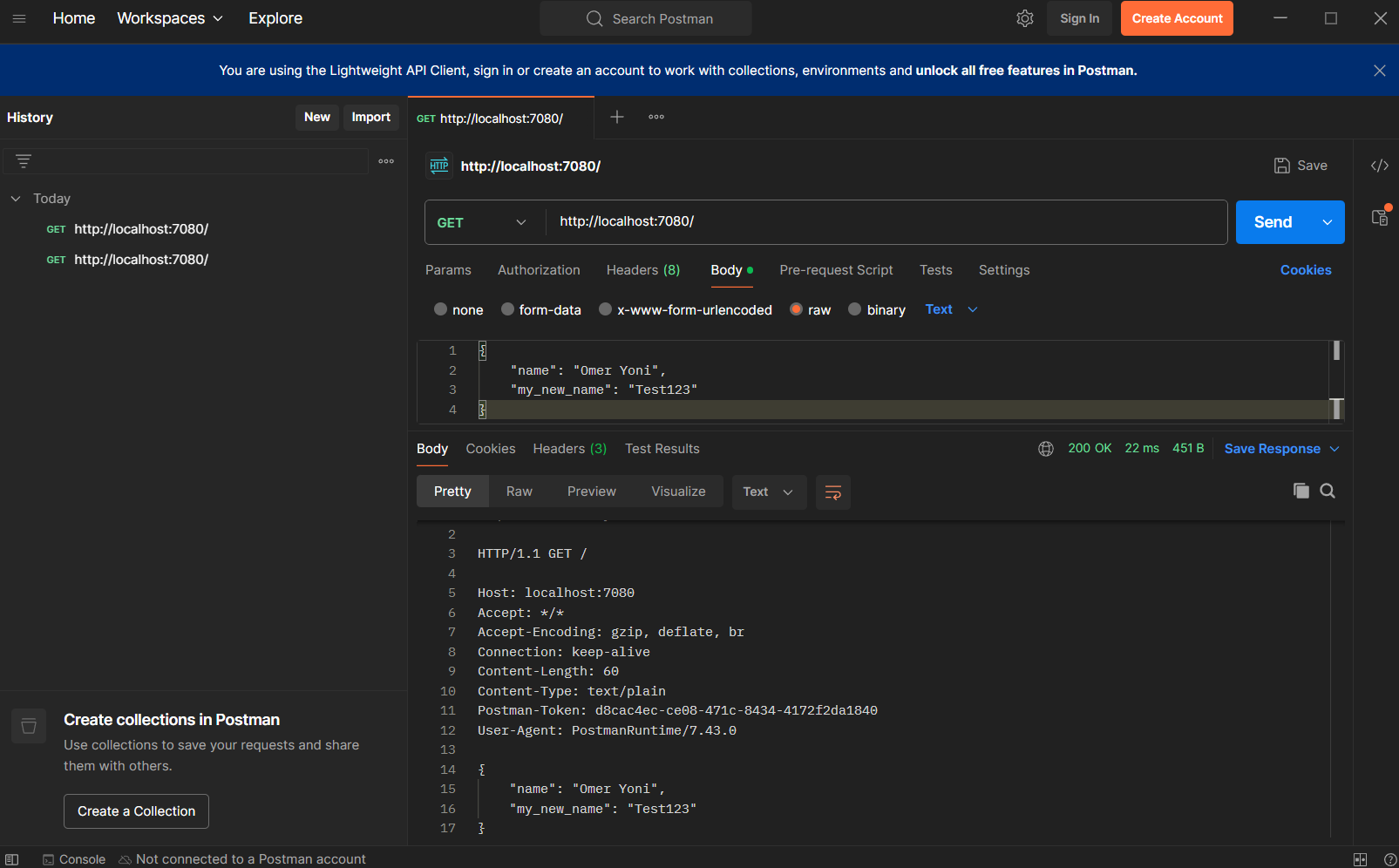
A screenshot of a computer

Description automatically generated

הרצת השירות וחשיפתו לפורט 8080.



גישה לשירות דרך הקישור https://localhost:7080/.



שימוש בתוכנה Postman, שעוזרת לנו לנהל דברים מאחורי הקלעים כמו פקודות CRUD פשוטות כפי שניתן לראות בתמונה לעיל. עשינו שליחה (Send) של מה שכתבנו, ועם פקודת GET קיבלנו את התוצאה הרצויה.