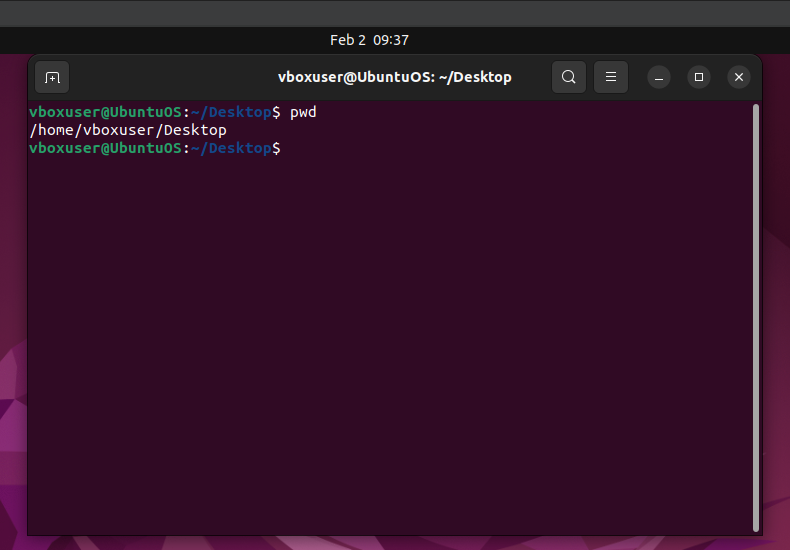


מגישים :

נדב ראובנס – 315819623

רון סיסו – 316398460

הרצאה 1 – Introduction :



הפקודה pwd (print working directory), כפי שניתן להבין משמה, מדפיסה את הpath המלא של התיקייה הנוכחית בה אנו נמצאים (working directory).

הרצאה 2 – Linux :

שקף 10:

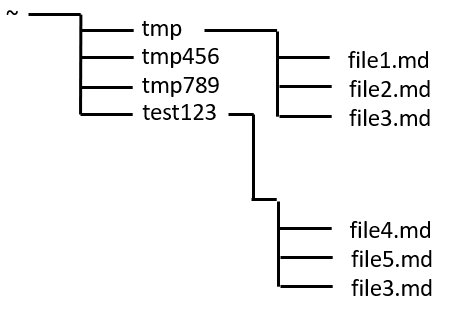
ניתן לראות מצילום המסך הקודם כי:

Username = vboxuser

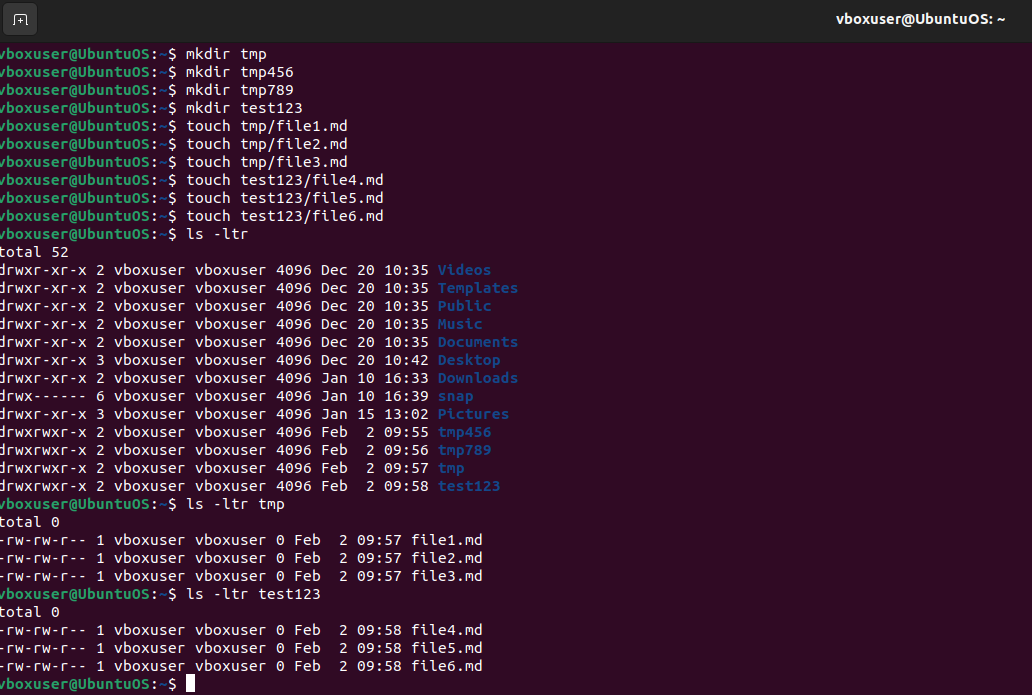
Machine name = UbuntuOS

Current directory = home/vboxuser/Desktop

שקף 26:



ניצור את המבנה הבא תחת הroot folder :



באמצעות פקודת mkdir יצרנו את כל התיקיות במבנה. לאחר מכן באמצעות פקודת touch יצרנו את הקבצים של המבנה בpath המתאים.

לבסוף באמצעות ls הראנו את רשימת הקבצים והתיקיות כדי לבדוק כי אכן נוצר המבנה הדרוש.

שקף 35:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטיראשית נבצע את הפקודות ln,cat, man כפי שהן מופיעות במצגת :

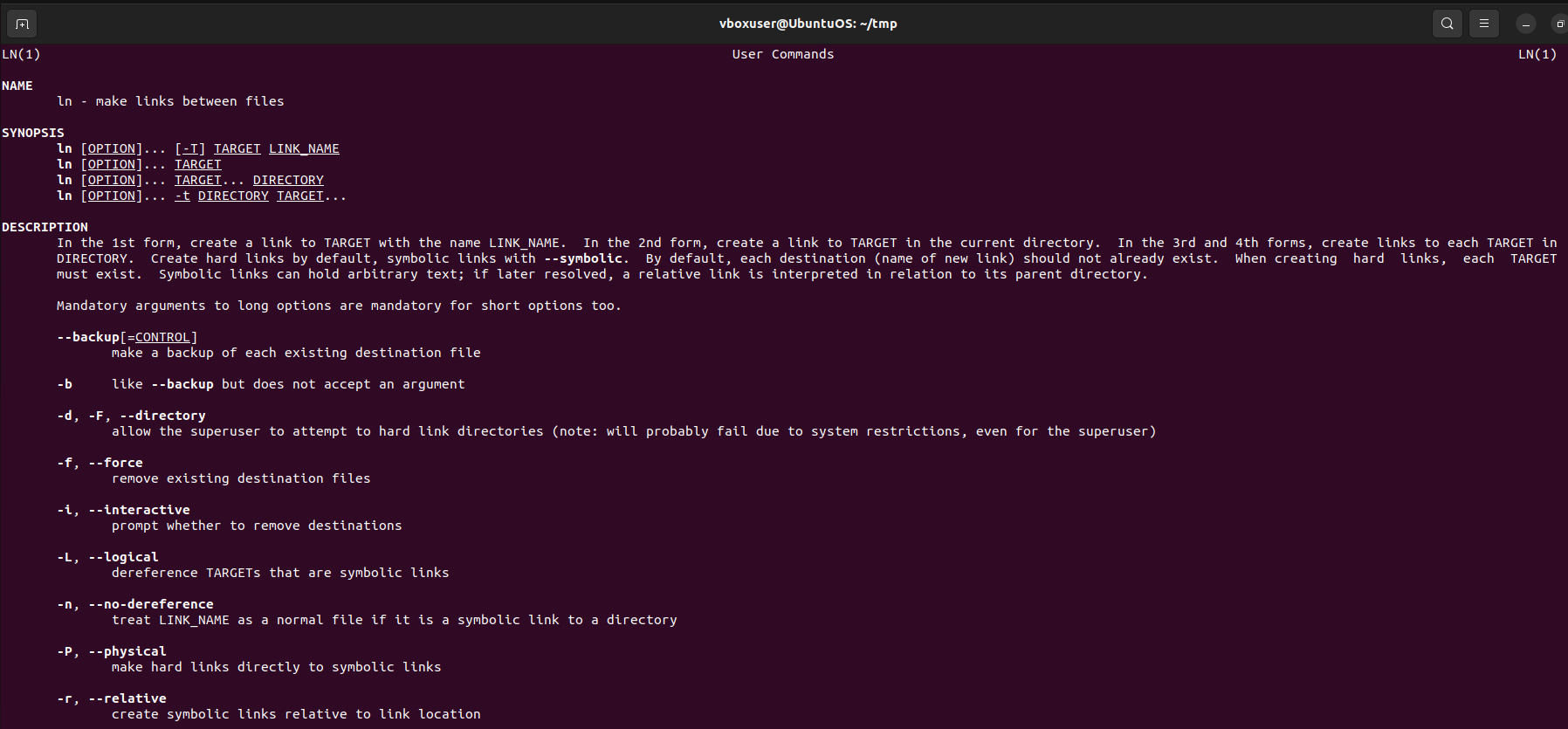
תמונה שמכילה טקסט, חשמל, צילום מסך, תוכנה

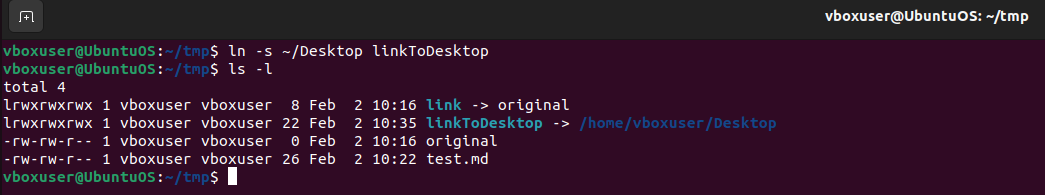
התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטינבצע הדפסת documentation של הפקודות ln וmv באמצעות הפקודה man:

תמונה שמכילה טקסט, צילום מסך

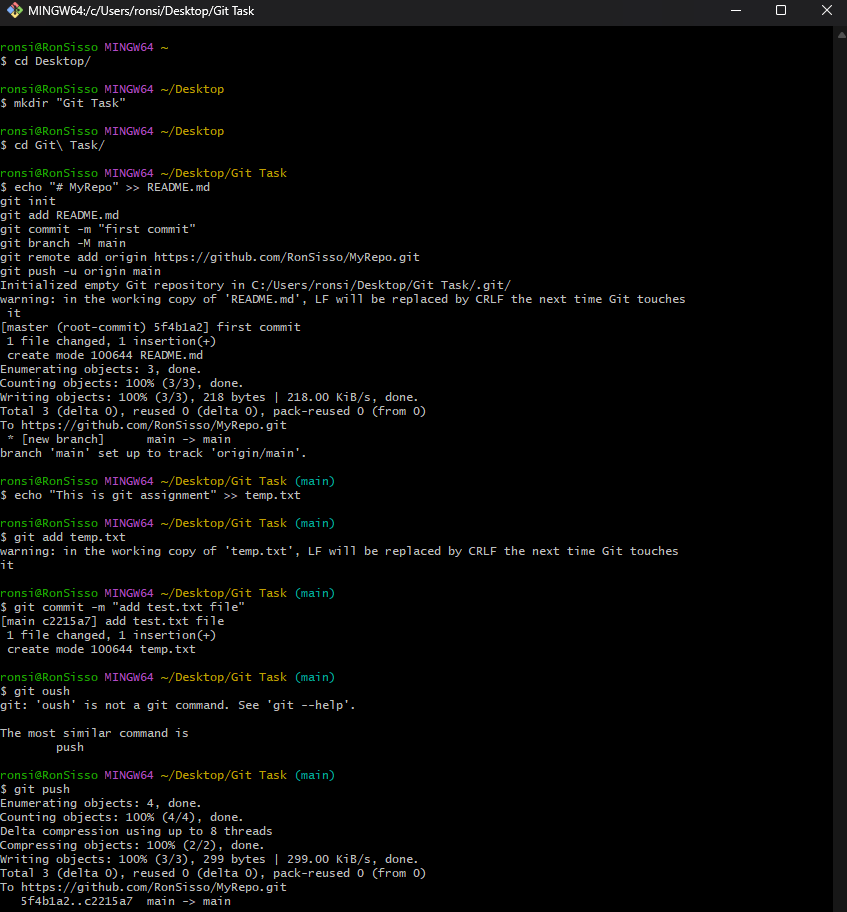
התיאור נוצר באופן אוטומטי

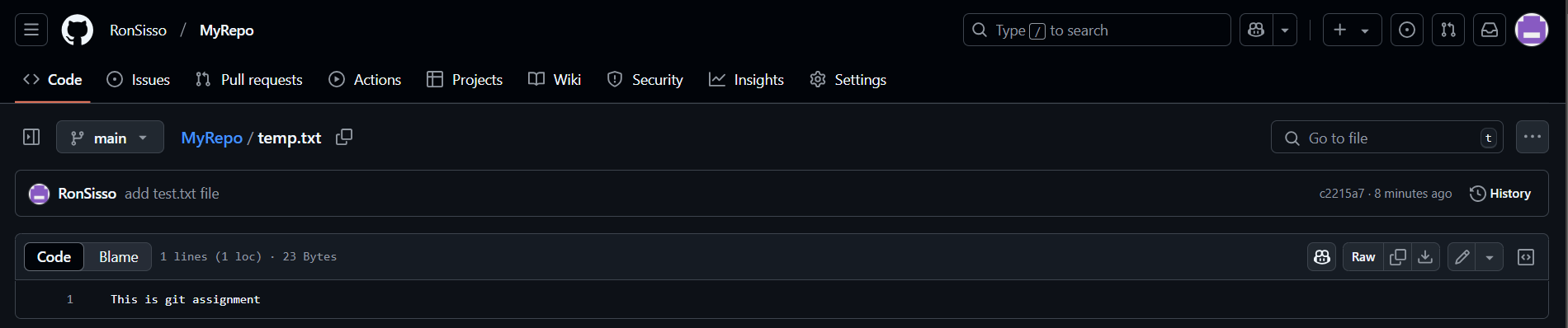
נבצע soft link לתיקיית הDesktop :

הרצאה 3 – Git :

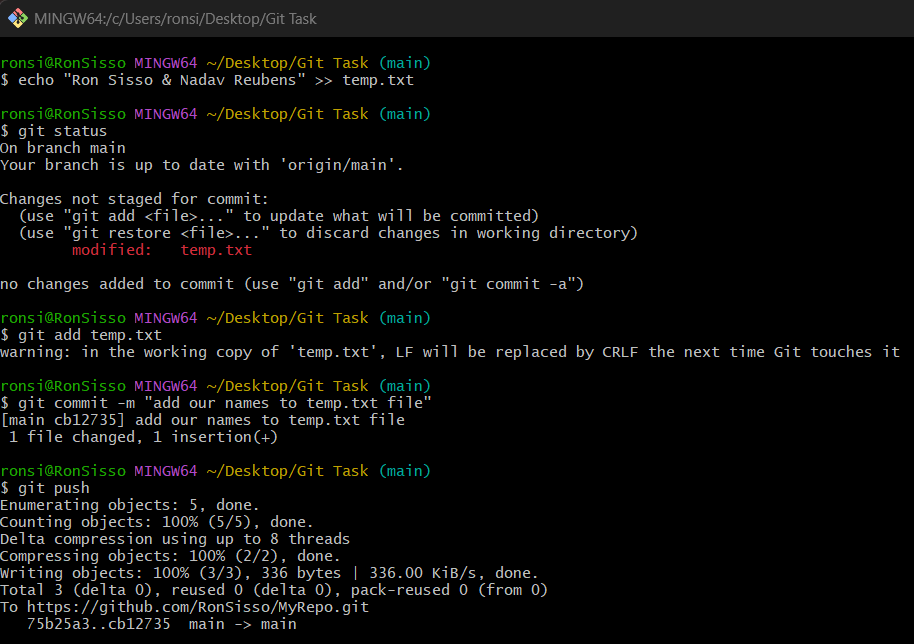
נבצע יצירת repository חדש בgithub ולאחר מכן נבצע CRUD כלומר

Create -> Read -> Update -> Delete על קובץ בשם temp.txt



נבצע קריאה של הקובץ בgithub :

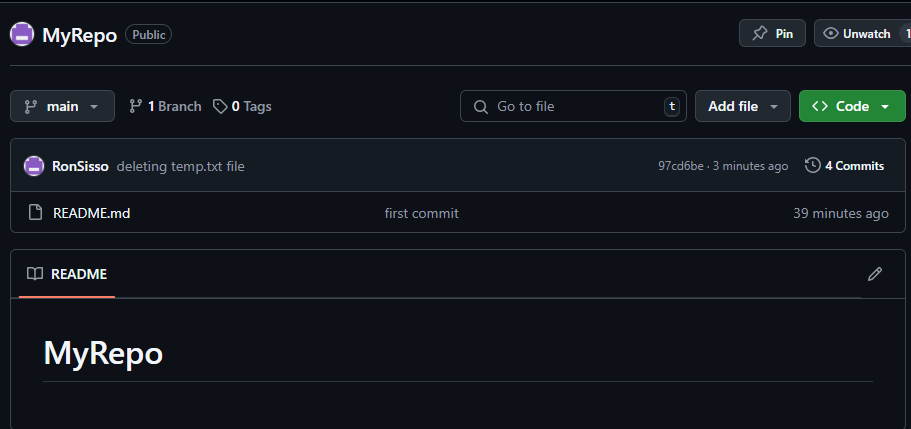
נבצע update על ידי הוספת תוכן נוסף לקובץ temp.txt. לאחר מכן נבצע commit וpush לrepository שלנו.

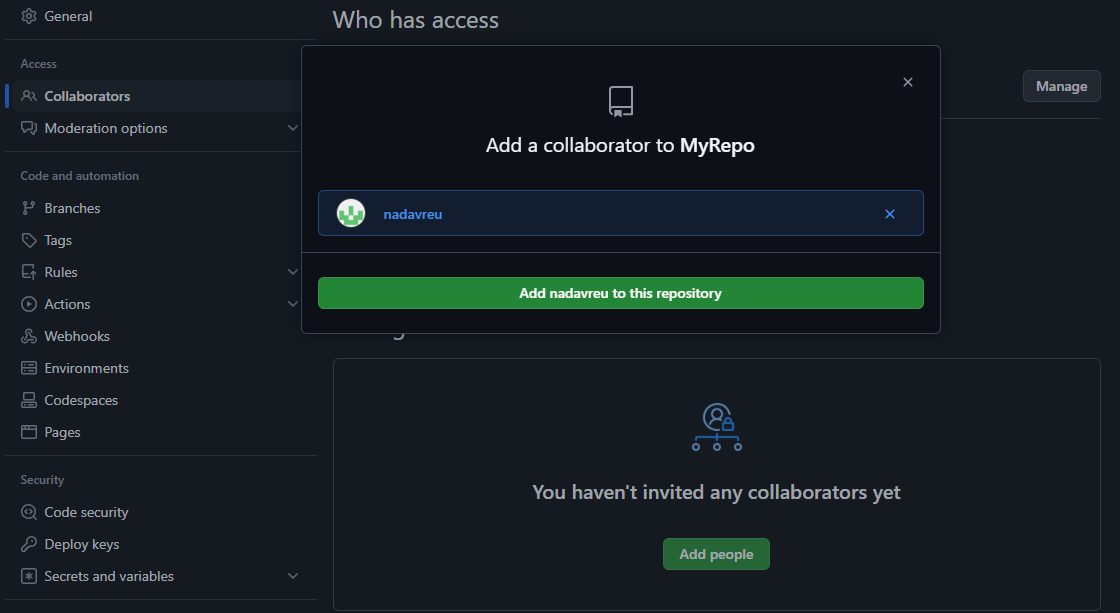


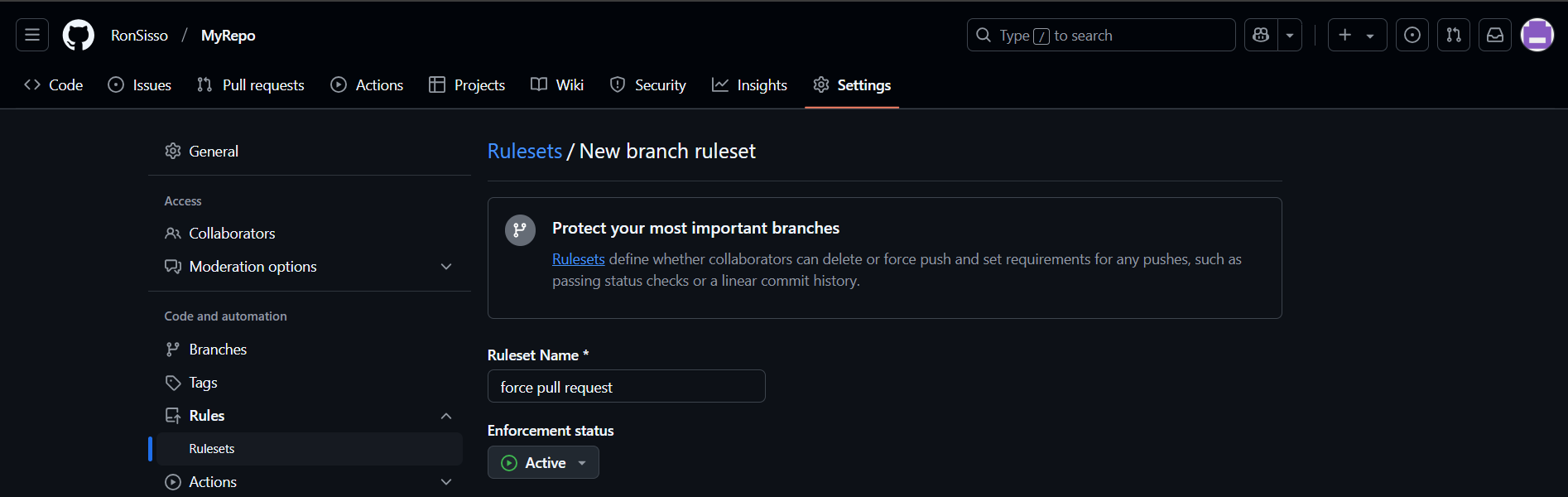
נבצע delete לקובץ temp.txt ע"י הפקודה git rm

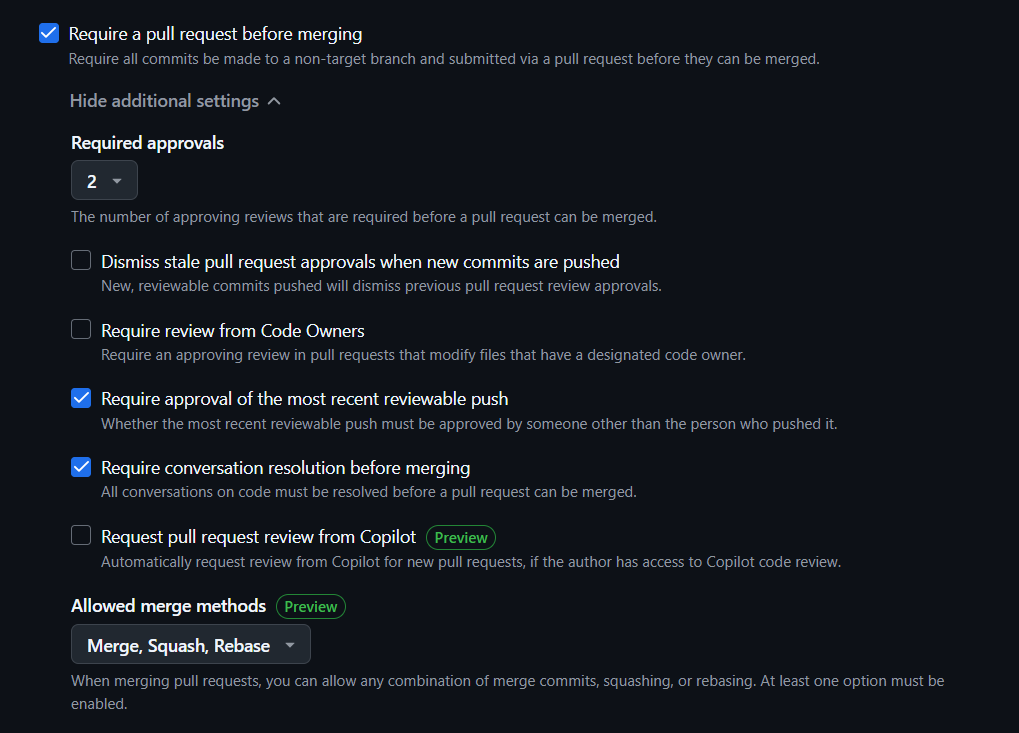
תמונה שמכילה טקסט, חשמל, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטי



נוסיף את חבר הצוות השני אל הrepository:

נבצע configure לrepository עבור force pull request :



Dependabot הוא כלי מובנה ב-GitHub שמעדכן אוטומטית את ה-dependencies בפרויקט. הוא מזהה גרסאות ישנות, מסייע לשמור על קוד מאובטח ומציע עדכונים דרך pull requests.

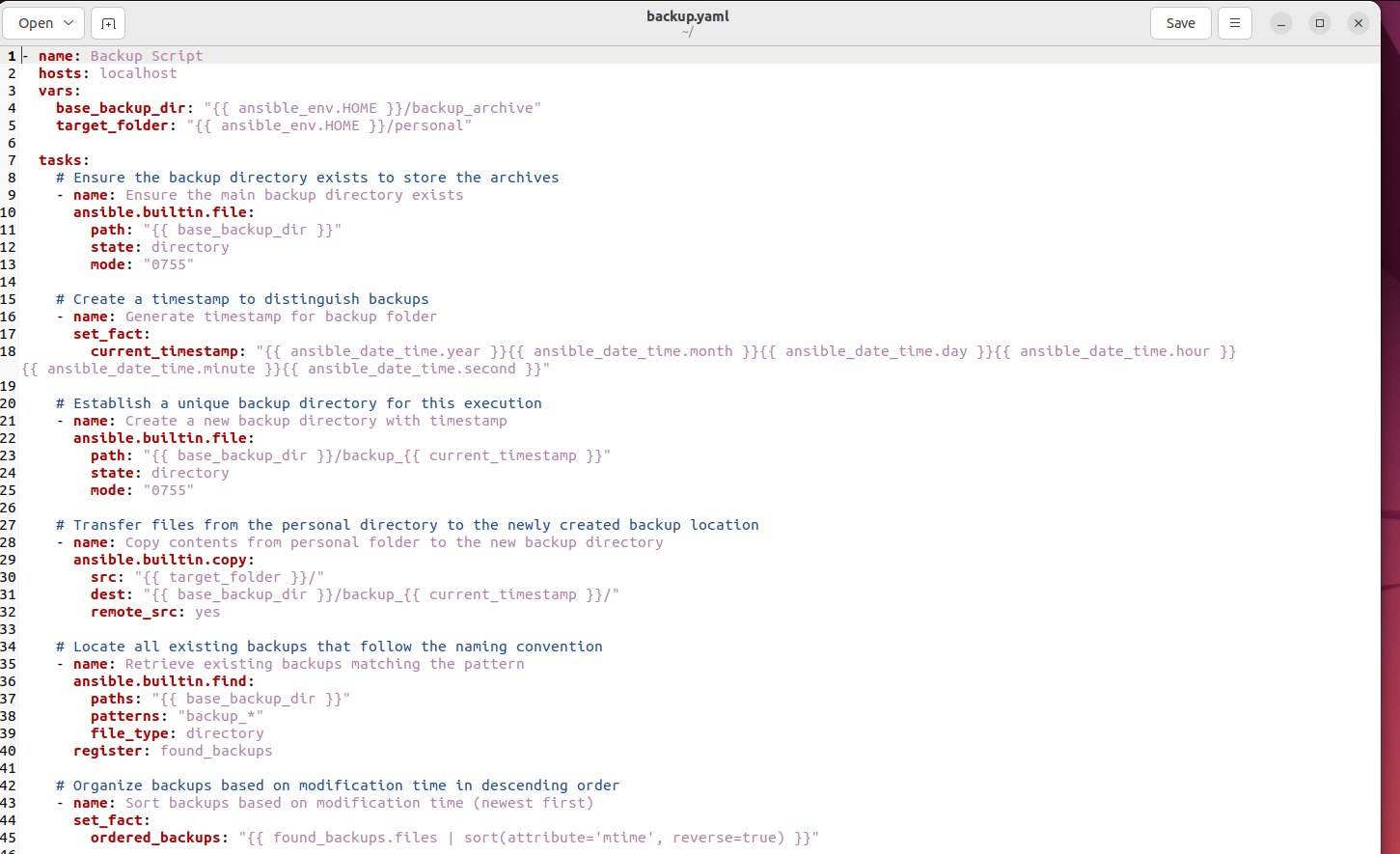
מה הצורך שלנו בDependabot ?

הוא שומר על הפרויקט שלנו מאובטח, מעודכן ויציב. הוא מזהה אוטומטית נקודות תורפה באבטחה ומציע עדכונים ל-dependencies, מה שחוסך זמן, מקטין סיכונים ומונע בעיות תאימות. במקום לעדכן חבילות ידנית, Dependabot עושה זאת אוטומטית ושומר על הפרויקט ברמה הגבוהה ביותר.

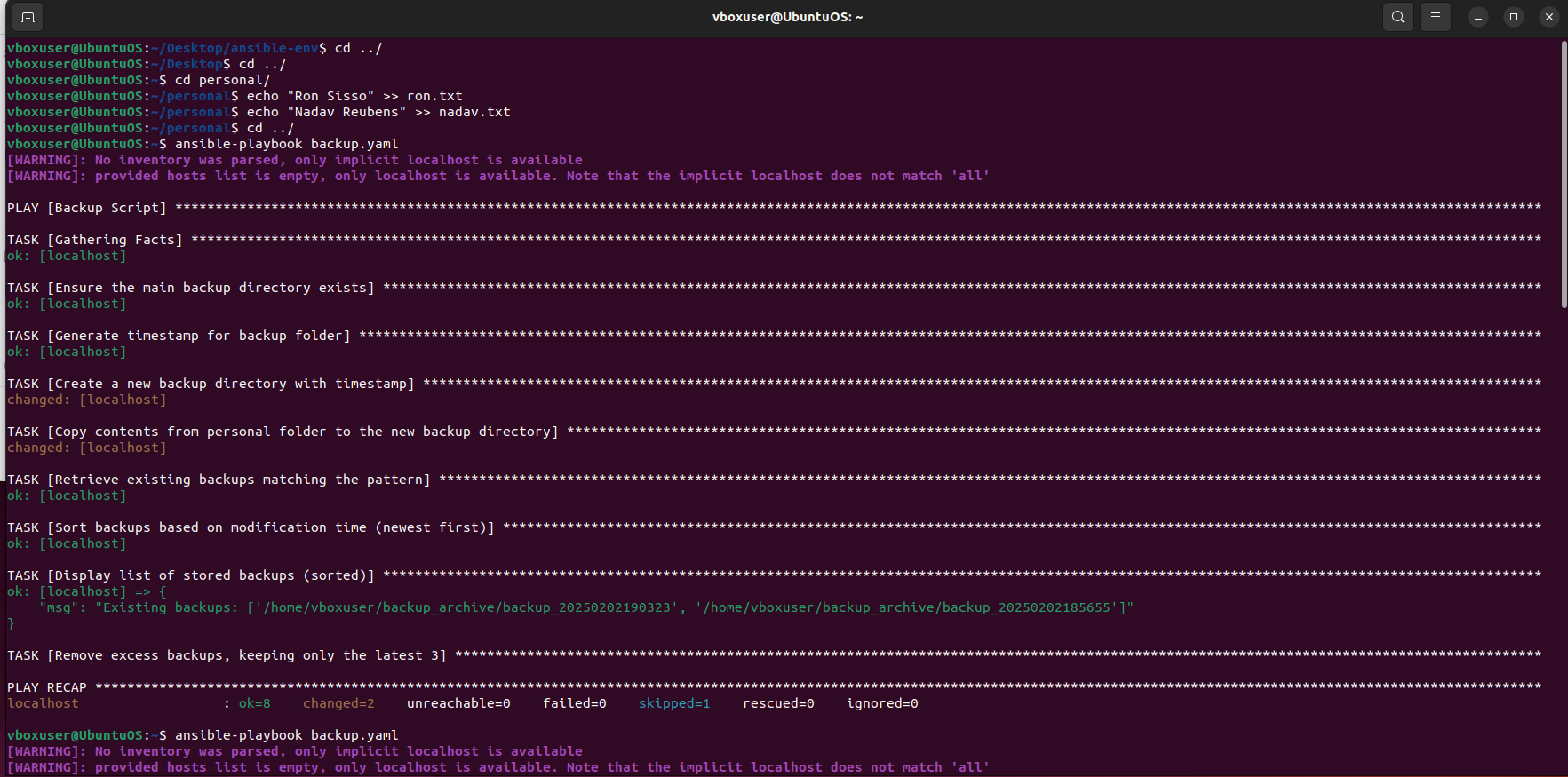
נשתמש ב Dependabot כאשר נרצה לוודא שהפרויקט שלנו מעודכן ובטוח מפני פרצות אבטחה. במיוחד, נשתמש בו כאשר GitHub מזהה פגיעות אבטחה ב dependenciesכאשר אנו רוצים לשמור על יציבות הקוד ולהימנע מבעיות תאימות, וכאשר הפרויקט שלנו כולל CI/CD או רץ בסביבת production שבה עדכונים שוטפים הם קריטיים. בנוסף Dependabot שימושי כאשר אנו רוצים לחסוך זמן על ידי עדכון אוטומטי של החבילות במקום תחזוקה ידנית.

Dependabot חיוני לכל פרויקט כדי לשמור על ביטחון, יציבות ותחזוקה קלה.

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטיהרצאה 4 – Ansible :

\* הקוד מצורף בצורה ברורה יותר כקובץ yaml בתיקיית הgit של הפרויקט שלנו.

הרצה מהterminal :

