

מדריך SSH מנוצב – GitHub & GitLab

עובדים רק עם SSH, כולל ניהול כמה חשבונות במחשב אחד והעברה למחשב חדש

תוכן עניינים

(2) הוספת המפתח הציבורי לחשבונות
(addkeys#)

(1) יצירת זוג מפתחות SSH (#keys)

(4) בדיקות חיבור (test#)

(3) קובץ ~/.ssh/config (#config)

(6) clone חדש (SSH) (repo-new#)

(5) עדכון ריפו קיים ל-SSH (#repo-existing)

(8) מעבר למחשב חדש (migrate#)

(7) הרשאות נכונות לקבצי SSH (#perms)

(9) שאלות נפוצות (faq#)

1) יצירת זוג מפתחות SSH

הפקודה יוצרת מפתח פרטי (נשאר אצלך בלבד) ומפתח ציבורי (שנותנים ל-GitHub/GitLab):

```
C -t ed25519 ssh-keygen -f ~/.ssh/id_github_personal -C "האימייל שלך"
```

טיפ: אפשר ללחוץ Enter פעמיים (ללא סיסמה), או להוסיף Passphrase לאבטחה חזקה יותר.

מה נוצר?

~/.ssh/id_github_personal – פרטי
~/.ssh/id_github_personal.pub – ציבורי

(2) הוספת המפתח הציבורי לחשבונות

GitLab

SSH Keys → Settings

הדבק את אותו התוכן (השורה שמתחילה ב--ssh
ed25519).

GitHub

SSH and GPG Keys → Settings

New SSH key

הדבק את התוכן של הקובץ הציבורי:

```
at ~/.ssh/id_github_personal.pub
```

אפשר לייצר גם מפתחות נפרדים לחשבון עבודה (למשל: id_github_work, id_gitlab_work).

(3) קובץ ~/.ssh/config

הקובץ הזה קובע איזה מפתח יישמש לכל Host (כינוי). כך תומכים בכמה חשבונות במחשב אחד.

```
# אישי GitHub
Host github-personal
HostName github.com
User git
IdentityFile ~/.ssh/id_github_personal
IdentitiesOnly yes
```

```
# אישי GitLab
Host gitlab-personal
HostName gitlab.com
User git
IdentityFile ~/.ssh/id_gitlab_personal
IdentitiesOnly yes
```

Host – שם כינוי מקומי (למשל github-personal) שתשתמש בו בפקודות SSH ו-Git.

IdentityFile – הנתיב למפתח הפרטי המתאים לחשבון הזה.

IdentitiesOnly yes – מכריח שימוש במפתח הזה בלבד (חשוב כשיש כמה מפתחות).

(4) בדיקות חיבור

בדוק שכל חשבון מזוהה כראוי:

```
ssh -T git@github-personal
ssh -T git@gitlab-personal
```

הודעת הצלחה צפויה: *...Hi <username>! You've successfully authenticated*

5) עדכון ריפו קיים ל-SSH

בתוך תיקיית הריפו הקיים:

```
git remote -v  
git remote set-url origin git@github-personal:USERNAME/REPO.git
```

אחריזה: `git push origin main`

6) clone חדש (SSH)

בחר SSH באתר וקח קישור בסגנון:

```
git clone git@github-personal:USERNAME/REPO.git
```

7) הרשאות נכונות לקבצי SSH

```
chmod 700 ~/.ssh  
*/chmod 600 ~/.ssh
```

אם ההרשאות פתוחות מדי, SSH יסרב להשתמש במפתח (בטיחות).

8) מעבר למחשב חדש

במחשב החדש

```
-xzvf ~/ssh_backup.tar.gz -C  
  chmod 700 ~/.ssh  
*/chmod 600 ~/.ssh
```

במחשב הישן

```
zvf ssh_backup.tar.gz ~/.ssh
```

כל עוד בקובץ config הנתיבים עם ~, אין תלות בשם המשתמש.

9) שאלות נפוצות

האם צריך Token?

לא. כשעובדים ב-SSH אין צורך ב-Tokens. Tokens רלוונטיים ל-HTTPS בלבד.

מה אם אני רואה אזהרת fingerprint?

בפעם הראשונה מול שרת חדש (GitHub/GitLab) הקלד **yes**. החתימה תישמר ב-
~/.ssh/known_hosts.

אפשר כמה חשבונות באותו אתר?

כן. צור מפתחות נפרדים והוסף בלוקים נפרדים בקובץ config עם שמות Host שונים.

מוכן להדפסה ל-PDF מתוך הדפדפן (Ctrl/Cmd + P) • מומלץ לבחור "רקעי צבעים" בהגדרות ההדפסה