מדריך התחברות לHPC עם VSCODE ושליחת JOB

ניתן להתחבר לHPC רק ממחשבי כיתת הלימוד במעבדה או בהתחברות למדריך VPN

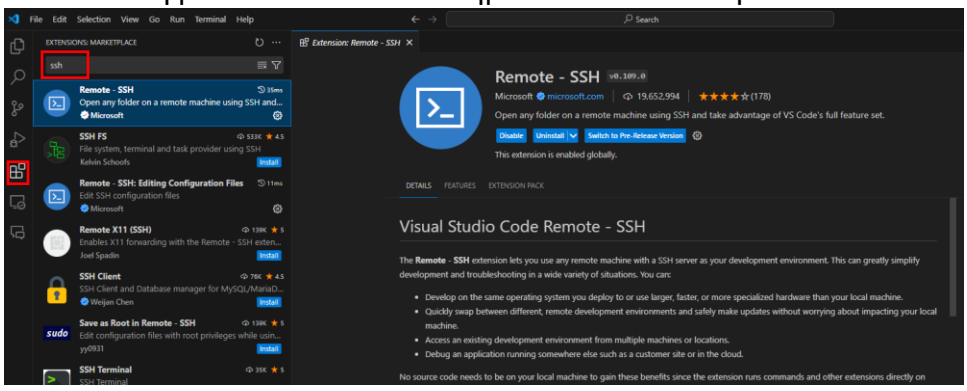
שם המשתמש והסיסמא הם פרטי ההתחברות שלכם במכללה

במדריך זה אנחנו

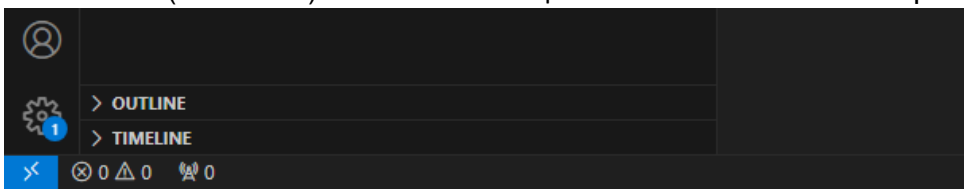
- נגדיר חיבור של HPC-MASTER ב VSCODE
- נקים סביבה וירטואלית בCONDA ונתקין את הספריות הרלוונטיות
- נערוך קובץ SBATCH להגדרת ה JOB
- נשלח את הJOB ונצפה בתוצאה
- נעבור על פקודות SLURM בסיסיות לצפייה וניהול JOBS

הגדרת חיבור של HPC-MASTER בVSCODE

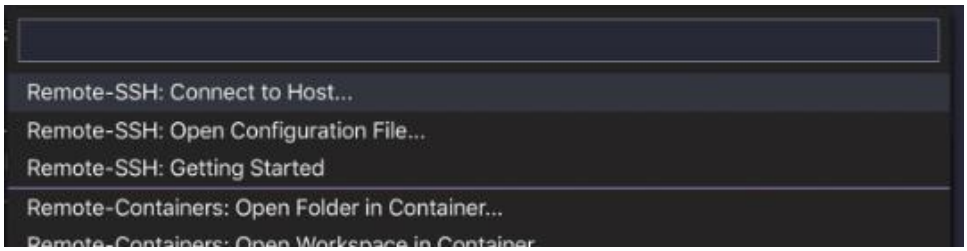
1. נתחיל בלוודא שתוסף Remote - SSH מותקן בVSCODE ואם לא אז נתקין



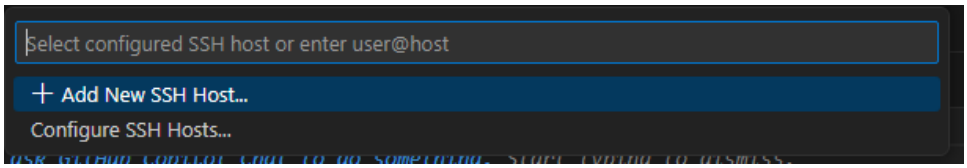
2. נלחץ על כפתור open remote window בצד שמאל למטה (כפתור כחול)



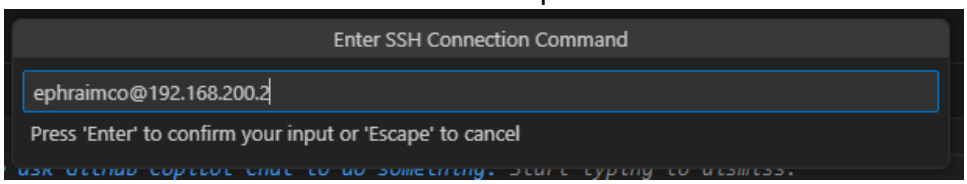
3. בתפריט שיפתח יש לבחור את האפשרות **Connect to Host**



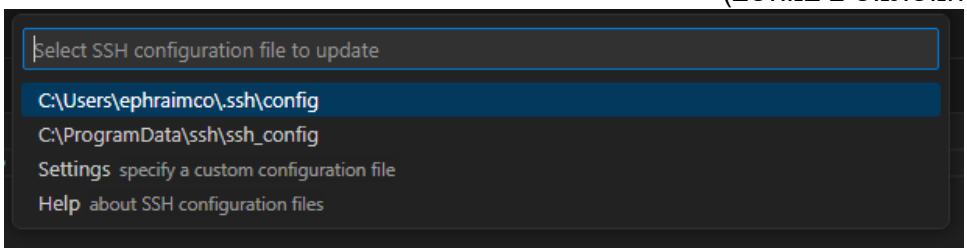
4. בחרו באפשרות הבאה **Add New SSH Host**



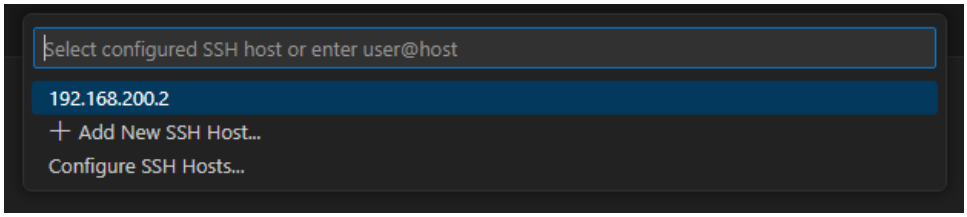
5. בחלון הבא יש להקליד את שם המשתמש וכתובת השרת בפורמט user_name@host_ip כתובת שרת הMASTER הוא **192.168.200.2** וללחוץ ENTER



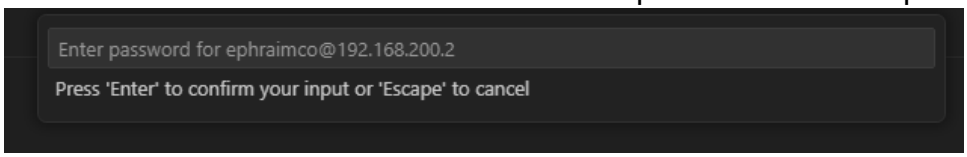
6. בחר את מיקום שמירת קונפיגורצית SSH (האפשרות הראשונה היא רק למשתמש, והשניה היא לכל המשתמשים במחשב)



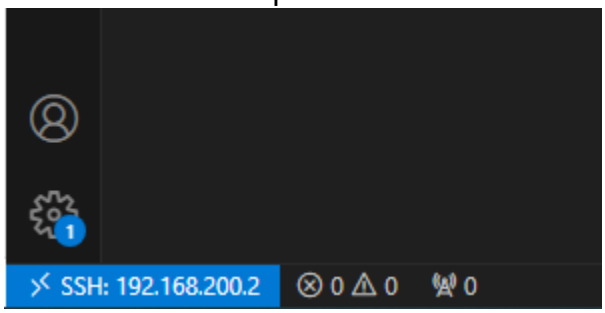
7. ועכשיו יש לחזור על סעיפים 2 ו 3 והפעם יופיע השרת שהגדרנו יש ללחוץ עליו



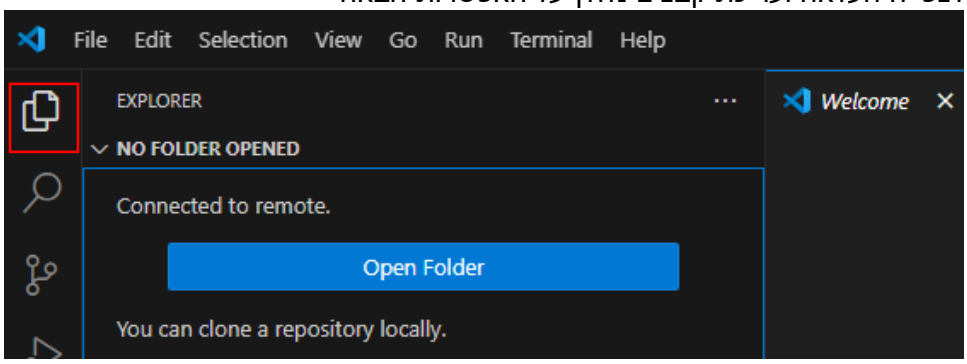
8. בחלון החדש שיפתח יש להזין את הסיסמא שלכם ואז ENTER



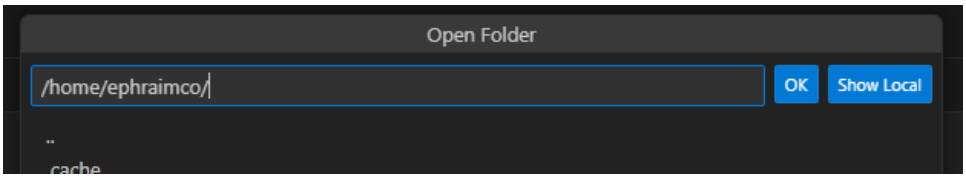
9. לאחר התחברות מוצלחת ניתן יהיה לראות את כתובת השרת שהתחברנו אליו



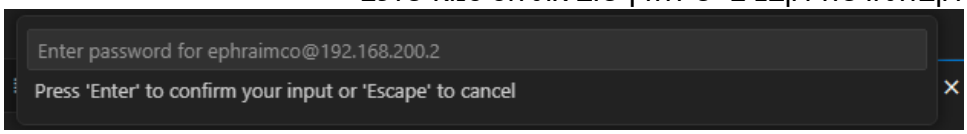
10. לצפייה העלאה ועריכת קבצים נלחץ על האפשרות הבאה



11. ואז OK



12. לקבלת גישה לקבצים יש להזין שוב את הסיסמא שלכם



הקמת סביבה וירטואלית בANACONDA והתקנת ספריות

1. ליצירת סביבה ווירטואלית יש להריץ את הפקודה **conda create --name env_name python=3.9**
   בפקודה זו אנו מגדירים סביבה בשם env_name וpython בגרסת 3.9
2. להפעלת הסביבה יש להריץ את הפקודה **conda activate env_name**
3. להתקנת ספריה ניתן להשתמש בconda או pip
4. אידיאלית כדאי לשמור את הספריות הנדרשות בקובץ txt/yml ע"י הרצת אחד מהאפשרויות
   4.1. conda list --export > environment.yml
   4.2. pip freeze > requirements.txt
5. להתקנה מקובץ
   5.1. conda install --file environment.yml
   5.2. r requirements.txt- pip install
6. לסגירת הסביבה הווירטואלית יש להריץ את הפקודה **conda deactivate**

הגדרת ה JOB

כדי להריץ JOB בקלאסטר הHPC ניצטרך לערוך קובץ SBATCH שבו יוגדר למערכת כל הפרמטרים הרלוונטיים עבור הJOB ופקודות ההרצה בתוך ה JOB

Partition ב-SLURM הוא קבוצה של משאבים (מחשבים\שרתים) בתוך מערכת הHPC. ניתן להשתמש ב-Partitions כדי לנהל גישה למשאבים, לקבוע קדימויות ולשפר ביצועים.

מעבדת תכנה

אפי כהן - 01/04/2024

```bash
#!/bin/bash

#SBATCH --partition main                ### Specify partition name where to run a job.
#SBATCH --time 0-02:00:00               ### Job running time limit. Make sure it is not exceeding the partition time li>
#SBATCH --job-name 'job_name'             ### Name of the job. replace my_job with your desired job name
#SBATCH --output job-%J.out             ### Output log for running job - %J is the job number variable
#SBATCH --mail-user=username@post.jce.ac.il        ### User's email for sending job status
#SBATCH --mail-type=ALL                ### Conditions when to send the email. ALL,BEGIN,END,FAIL,REQUEU,NONE

#SBATCH --gpus=1                ### number of GPUs, ask for more than 1 only if you can parallelize your code for multi>
##SBATCH --mem=32G              ### ammount of RAM memory
##SBATCH --cpus-per-task=6     ### number of CPU cores

### Print some data to output file ###
echo `date`
echo -e "\nSLURM_JOBID:\t\t" $SLURM_JOBID
echo -e "SLURM_JOB_NODELIST:\t" $SLURM_JOB_NODELIST "\n\n"

### Start you code below ####
module load anaconda                    ### load anaconda module (must present when working with conda environments)
source activate env_name                  ### activating environment, environment must be configured before running the j>
python -c "print('hello')"                          ### replace with your own command
```

ניתן להוריד את הדוגמא מהקישור המצורף

לקובץ יש 2 חלקים החלק הראשון הוא פרמטרים של SBATCH שבהם מוגדר הJOB והחלק השני זה מה שירוץ על הNODE לאחר ההקצאה והרצת הJOB

לפני כל פרמטר מופיע הסימן # אם ## אז הפרמטר בהערה והJOB ירוץ עם הפרמטר הדיפולטיבי

#SBATCH --partition	קבוצת שרתים בHPC שמהם המערכת יכולה לתת עבור הJOB, כדי לבדוק איזה Partitions זמינים עבורכם יש להריץ את הפקודה **sinfo** וכדי לקבל עוד נתונים על זמינות, זמן ריצה מקסימאלי, מספר השרתים ב Partition והסטטוס שלהם, יש להריץ **"sinfo -t% D% l% a% P% o**
#SBATCH –time	מגדיר את זמן הריצה המקסימאלי עבור הJOB כדי לבדוק זמן ריצה מקסימאלי ב partition ספציפי שבו תרצו להריץ את הJOB יש להריץ את הפקודה **sinfo -l% P% o "-partition_name p**
#SBATCH --job-name	בפרמטר הזה תוכלו להגדיר את השם של הJOB
#SBATCH --output	בפרמטר הזה יש להגדיר את מיקום ושם הקובץ שבו ישמר הOUTPUT של הJOB בדוגמא מופיע job-%J.out מה שייצור קובץ עם הID של הJOB למנוע דריסה של קובץ הOUTPUT בין הרצה להרצה
#SBATCH --mail-user	כדי לקבל התראות על JOB במייל יש להזין את כתובת המייל שלכם במכללה
#SBATCH --mail-type	בפרמטר הזה תוכלו להגדיר את סוג ההתראות שתרצו לקבל האפשרויות הם ALL, BEGIN, END, FAIL, REQUEU, NONE
#SBATCH --gpus	בהרצה רגילה יש להשאיר פרמטר זה על 1 ואם מוגדר 0 אז הכרטיס לא יהיה זמין, אם מריצים על NODE עם יותר מכרטיס אחד (כמו שרת הCUDA) אז יש לשנות למס' הכרטיסים הקיימים בNODE
#SBATCH --mem	פרמטר זה מגדיר את הRAM הנדרש עבור הJOB שימו לב שהפרמטר הוא כהערה ובהקצאה יתקבל RAM בהגדרה הדיפולטיבית עבור הpartition כדי לבדוק את הRAM הזמין עבורכם יש להדפיס את רשימת השרתים ב partition שלכם ולהריץ שאילתא על NODE ספציפי

	• להדפסת רשימת השרתים **sinfo -N% "o" -partition_name p** • לבדיקת הRAM על גבי NODE מהרשימה **scontrol show node=node_name | grep RealMemory**
`#SBATCH --cpus-per-task`	בפרמטר זה נגדיר את כמות הCORES שנקבל עבור הJOB וכדי לבדוק את הCORES הזמינים בNODE ספציפי ניתן להריץ **scontrol show node=node_name | grep CPUTot**

בחלק הבא מופעים פקודות הJOB שירוצו על הNODE לאחר ההקצאה

module load anaconda	פקודה זו מתאימה את משתני הסביבה כך שיכללו את הנתיבים הנחוצים לשימוש ב-Anaconda. לבדיקת המדולים הזמינים עבורכם יש להריץ את הפקודה **module avail**
source activate env_name	מפעילה סביבת Anaconda שצוינה, מאפשרת שימוש בחבילות ובספריות שהותקנו בסביבה זו. לבדיקת הסביבות שקימות אצלכם יש להריץ את הפקודה **conda env list**
python hello.py	הרצה של קובץ python

לפני הרצת הJOB יש להריץ conda deactivate אם אתם בתוך הסביבה

לסיום יש להריץ את הפקודה הבאה בTERMINAL בתיקיית קובץ הSBATCH והקוד שתרצו להריץ

sbatch hello.sbatch

```
[ephraimco@hpc-master hello_world]$ sbatch hello.sbatch
Submitted batch job 180
```

לאחר ההרצה תקבלו את הID של הJOB ששלחתם וישמר קובץ .OUT עם הID ובתוכו יהיה הOUTPUT של הקוד שרץ על הNODE שהוקצה

```
job-180.out ×
hello_world_tests > hello_world > job-180.out
1   Sun Mar 17 04:00:58 PM IST 2024
2
3   SLURM_JOBID:          180
4   SLURM_JOB_NODELIST:   HPC-RTX3070-01
5
6
7   hello
8
```

מעבדת תכנה

אפי כהן - 01/04/2024

פקודות SLURM בסיסיות לניהול וצפייה ב-JOBS

פקודה	תיאור
squeue	מציג את רשימת ה-JOBS בתור ההמתנה או בביצוע.
scancel JOB_ID	מבטל JOB בהתאם ל-ID שלו.
sacct	מספק מידע מפורט על JOBS שהופעלו, כולל מצב ריצה, זמן שימוש במשאבים, ועוד.
sinfo	מציג מידע על המצב הנוכחי של משאבי ה-HPC, כמו מספר הליבות הזמינות, זמן ההמתנה, וכו'
scontrol	מאפשרת ניהול והגדרת משתנים של מערכת SLURM, כולל פעולות על JOBS, ועוד.