

## מעבדה בניתוח והצגת נתונים (094295)

### אביב תשפ"א 2021

#### DevOps Guide

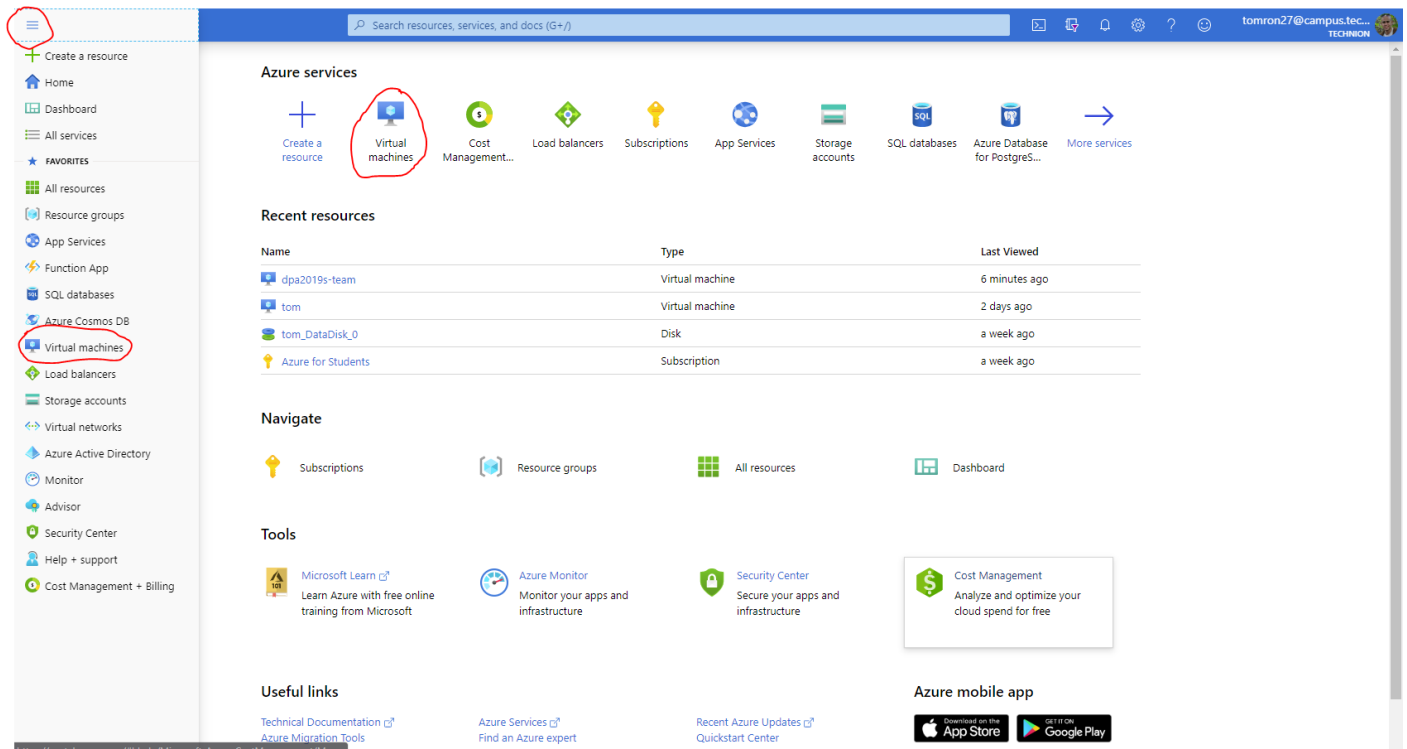
#### חיבור ל-Azure VM

משאבי חישוב בקורס ינוהלו ע"ג מכונות וירטואליות (VM) בענן Microsoft Azure.

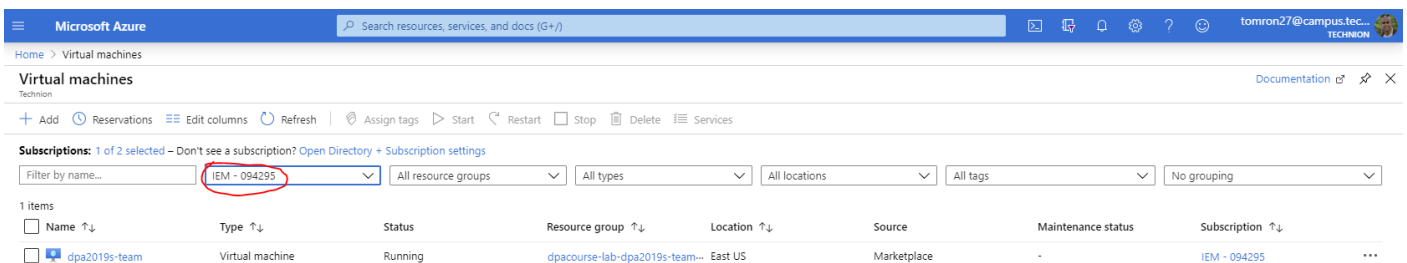
1. כניסה לפורטל: <https://portal.azure.com/>

שם משתמש וסיסמה טכניונים.

בכניסה לפורטל נראה מסך דומה:



2. יש לבחור ב-Virtual Machines. נראה מסך דומה (לרוב שיש פילטר לפי הקורס שלנו):



3. יש לבחור במכונה שהוקצתה לכם. לדוגמה נבחר ב-dpa2019s-team:

Microsoft Azure

Home > Virtual machines > dpa2019s-team

Virtual machines

Search (Ctrl+/)

Connect Start Restart Stop Capture Delete Refresh

Advisor (1 of 1): Use availability sets for improved fault tolerance →

Resource group (change) : dpacourse-lab-dpa2019s-team-592854

Status : **Stopped (deallocated)**

Location : East US

Subscription (change) : IEM - 094295

Subscription ID : fb7fcadc-4320-4131-9ffe-34237a351365

Computer name : (start VM to view)

Operating system : Linux

Size : Standard F4s\_v2 (4 vcpus, 8 GiB memory)

Tags (change) : Click here to add tags

Azure Spot : N/A

Public IP address : dpa2019s-team

Private IP address : 10.0.0.4

Public IP address (IPv6) : -

Private IP address (IPv6) : -

Virtual network/subnet : DtlDpacourse-lab/DtlDpacourse-labSubnet

DNS name : dpa2019s-team.eastus.cloudapp.azure.com

Show data for last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

CPU (average)

Percentage CPU (Avg) dpa2019s-team: 0.57%

Network (total)

Network In Total (Sum) dpa2019s-team: 401.56 MB

Network Out Total (Sum) dpa2019s-team: 6.99 MB

4. אם המכונה במצב (Stopped (deallocated) נלחץ על Start. אחרת אפשר לעבור לשלב הבא.  
בעבור מס' דקות נקבל התראה שהמכונה עלתה:

Microsoft Azure

Home > Virtual machines > dpa2019s-team

Virtual machines

Search (Ctrl+/)

Connect Start Restart Stop Capture Delete Refresh

Advisor (1 of 1): Use availability sets for improved fault tolerance →

Resource group (change) : dpacourse-lab-dpa2019s-team-592854

Status : **Running**

Location : East US

Subscription (change) : IEM - 094295

Subscription ID : fb7fcadc-4320-4131-9ffe-34237a351365

Computer name : (not available)

Operating system : Linux

Size : Standard F4s\_v2 (4 vcpus, 8 GiB memory)

Tags (change) : Click here to add tags

Show data for last: 1 hour 6 hours 12 hours 1 day 7 days

CPU (average)

Percentage CPU (Avg) dpa2019s-team: 0.55%

Network (total)

Network In Total (Sum) dpa2019s-team: 402.2 MB

Network Out Total (Sum) dpa2019s-team: 7.1 MB

Notifications

More events in the activity log →

Dismiss all

Started virtual machine

Successfully started virtual machine 'dpa2019s-team'.

a few seconds ago

Successfully stopped virtual machine

Successfully stopped the virtual machine 'dpa2019s-team'.

4 minutes ago

## 5. בעת נלחץ על SSH->Connect:

The screenshot shows the Azure portal interface for a virtual machine. On the left, the 'Connect' tab is selected, and the 'SSH' option is highlighted. The main panel displays the following details:

- Connect** (selected), Start, Restart, Stop, Capture, Delete, Refresh
- SSH** (highlighted in the sidebar)
- RDP**: [course-lab-dpa2019s-team-592854](#)
- Bastion**: [ing](#)
- Location**: East US
- Subscription** (change): [IEM - 094295](#)
- Subscription ID**: fb7fcadc-4320-4131-9ffe-34237a351365
- Computer name**: (not available)
- Operating system**: Linux
- Size**: Standard F4s\_v2 (4 vcpus, 8 GiB memory)
- Tags** (change): [Click here to add tags](#)
- Azure Spot**: N/A
- Public IP address**: 23.100.27.104
- Private IP address**: 10.0.0.4
- Public IP address (IPv6)**: -
- Private IP address (IPv6)**: -
- Virtual network/subnet**: [DtlDpacourse-lab/DtlDpacourse-labSubnet](#)
- DNS name**: [dpa2019s-team.eastus.cloudapp.azure.com](#)

## 6. בחלון שנפתח נקבל:

The screenshot shows the 'Connect' window for the virtual machine. The 'SSH' tab is active, and the instructions for connecting via SSH are displayed:

- Connect via SSH with client**
- 1. Open the client of your choice, e.g. [PuTTY](#) or [other clients](#).
- 2. Ensure you have read-only access to the private key.
 

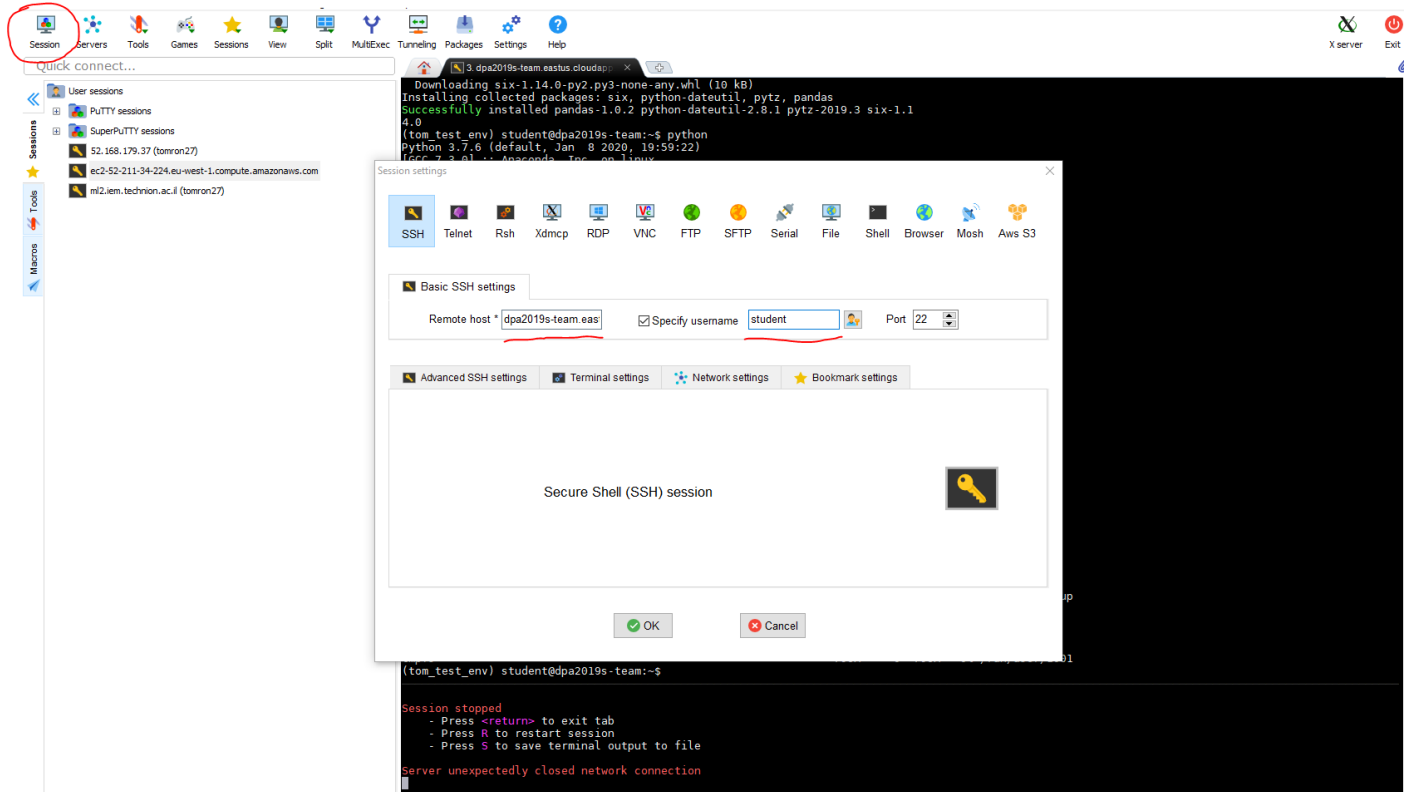
```
chmod 400 vmadmin.pem
```
- 3. Provide a path to your SSH private key file.
 

```
Private key path: ~/.ssh/vmadmin
```
- 4. Run the example command below to connect to your VM.
 

```
ssh -i <private key path> vmadmin@dpa2019s-team.eastus.cloudapp.azure.com
```
- Can't connect?**
  - [Test your connection](#)
  - [Troubleshoot SSH connectivity issues](#)

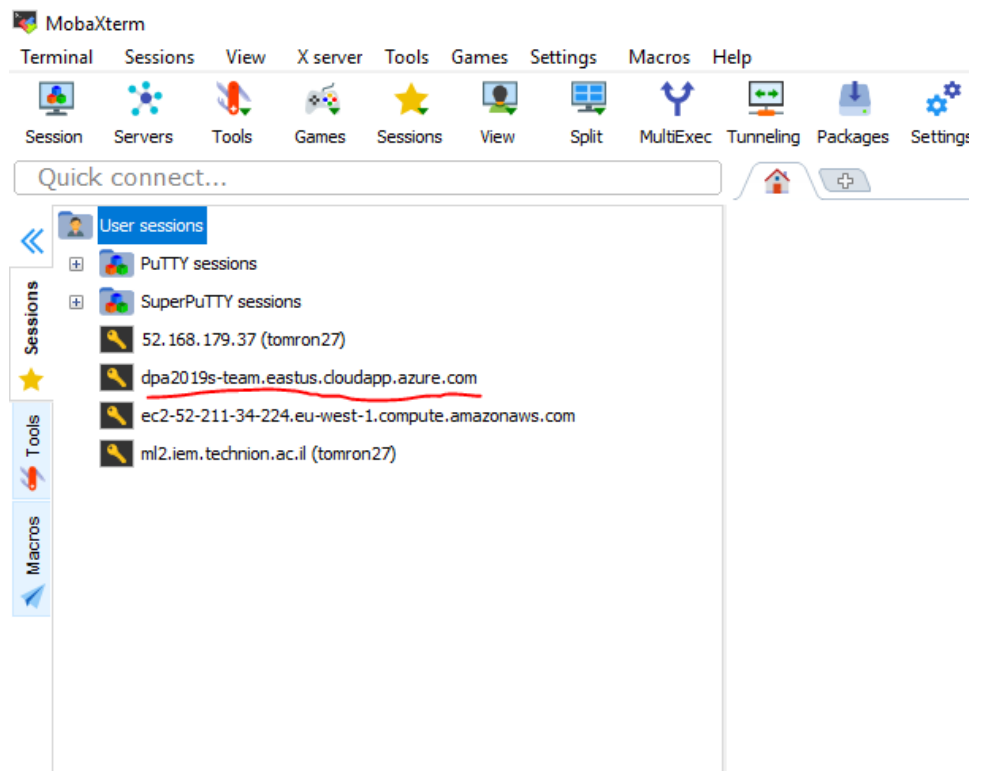
## 7. נעתיק את כתובת המכונה - מסומן בצהוב (החלק שאחרי ה-@). אין צורך לספק מפתח SSH.

8. ניגש ל SSH Client שברשותנו (משתמשי ווינדוס - מומלץ להשתמש ב-MobaXterm החינמי)  
וניצור SSH Session חדש:

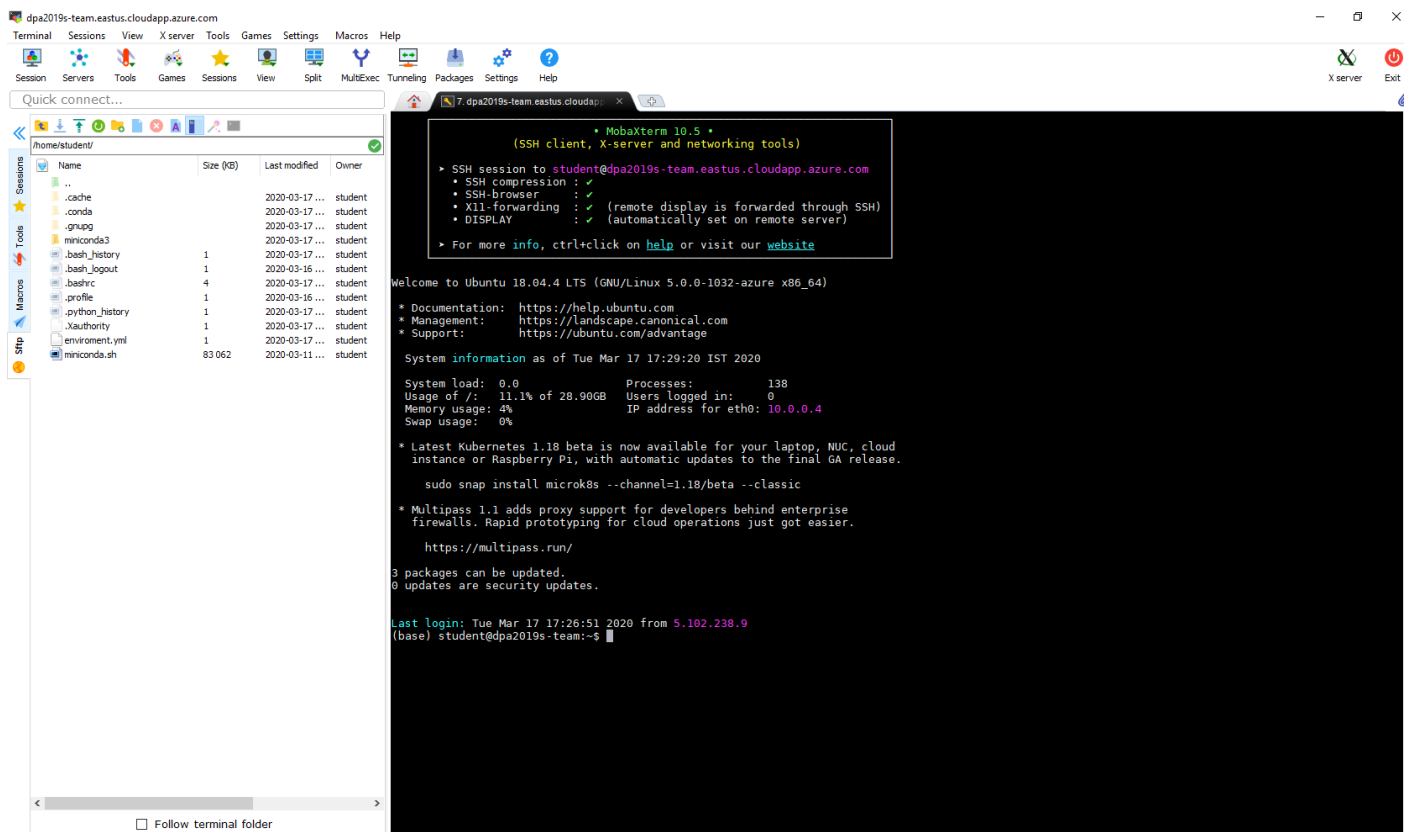


9. בשדה של Remote Host נכניס את כתובת המכונה ותחת Username את שם המשתמש שלכם. OK.

10. ע"מ להתחבר למכונה יש ללחוץ על ה Session החדש שנוצר בצד שמאל:



11. בחלון ה Terminal תתבקשו להזין את הסיסמה שלכם.  
לאחר התחברות למכונה, חלון ה Terminal יראה בערך כך:



כעת המכונה מוכנה לשימוש.

## שימו לב:

מכונה שאינה פעילה (אף משתמש לא מחובר ואין בה אף Process רץ של משתמש) תכובה באופן אוטומטי לאחר 30 דקות של חוסר פעילות. על מנת להרים אותה מחדש יש לחזור על שלבים 1-4.

הרצת תהליכים ברקע עם פקודת `nohup` לא תגרוור השבתה ונחשבת ל-Process רץ של משתמש.

שטחי אחסון

הרצת הפקודה

`df -h`

תראה לנו את מחיצות האחסון במכונה:

```
student@dpa2019s-team:~$ df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                     3.9G         0   3.9G   0% /dev
tmpfs                     796M    680K   795M   1% /run
/dev/sda1                 29G     1.7G    28G   6% /
tmpfs                     3.9G         0   3.9G   0% /dev/shm
tmpfs                     5.0M         0   5.0M   0% /run/lock
tmpfs                     3.9G         0   3.9G   0% /sys/fs/cgroup
/dev/sda15               105M     3.6M   101M   4% /boot/efi
/dev/sdc1                  98G     61M    93G   1% /StudentData
//dpacoursedatastorage.file.core.windows.net/dpacoursedatastore 10G     64K    10G   1% /datashare
/dev/sdb1                 32G     49M    30G   1% /mnt
tmpfs                     796M         0   796M   0% /run/user/1001
student@dpa2019s-team:~$
```

שימו לב:

מחיצות לשימוש:

- מחיצת `"/mnt"` - מחיצת עבודה זמנית. נמחקת בכל פעם שהמכונה יורדת. אין סיבה להשתמש בה.
- מחיצת `"/"` - מכילה את ה **home directory** שלכם. מוגבלת ל-GB30. יש להקפיד לא למלא את המחיצה יותר מידה. במידה והמחיצה תתמלא המכונה עלולה לקרוס.

## התקנת Miniconda

המכונות הוירטואליות ב Azure מגיעות ריקות, ללא Python עליהן. באחריותכם להקים סביבת עבודה על המכונה. אנו נעבוד עם מנהל סביבה של Miniconda (גרסה מוקטנת של Anaconda).

1. לאחר התחברות למכונה, יש להקליד:

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O  
~/miniconda.sh
```

- פקודה זו תוריד את קובץ ההתקנה של miniconda.

2. לאחר מכן נקליד:

```
bash ~/miniconda.sh
```

- פקודה זו תתקין את miniconda.

- במהלך ההתקנה יש ללחוץ Enter או לכתוב "yes" כל פעם שמתבקש.

- במהלך ההתקנה תוכלו לגלול טקסט במהירות ע"י לחיצה על "q".

3. בעת יש להתנתק ולהתחבר ל-Session מחדש.

4. ע"מ לוודא שההתקנה הצליחה, נקליד:

```
conda list
```

ונקבל פלט דומה:

```
(base) student@dpa2019s-team:~$ conda list
# packages in environment at /home/student/miniconda3:
#
# Name                    Version                Build                Channel
#-----
libgcc_mutex              0.1                    main
asn1crypto                1.3.0                  py37_0
ca-certificates           2020.1.1               py37_0
certifi                   2019.11.28             py37_0
cffi                      1.14.0                 py37h2e261b9_0
chardet                   3.0.4                  py37_1003
conda                     4.8.2                  py37_0
conda-package-handling    1.6.0                  py37h7b6447c_0
cryptography              2.8                    py37h1ba5d50_0
idna                      2.8                    py37_0
ld_impl_linux-64          2.33.1                 h53a641e_7
libedit                   3.1.20181209           hc058e9b_0
libffi                    3.2.1                  hd88cf55_4
libgcc-ng                 9.1.0                  hdf63c60_0
libstdc++-ng              9.1.0                  hdf63c60_0
ncurses                   6.2                    he6710b0_0
openssl                   1.1.1d                 h7b6447c_4
pip                       20.0.2                 py37_1
pycosat                   0.6.3                  py37h7b6447c_0
pyparser                  2.19                   py37_0
pyopenssl                 19.1.0                 py37_0
pysocks                   1.7.1                  py37_0
python                    3.7.6                  h0371630_2
readline                  7.0                    h7b6447c_5
requests                  2.22.0                 py37_1
ruamel_yaml               0.15.87                py37h7b6447c_0
setuptools                 45.2.0                 py37_0
six                       1.14.0                 py37_0
sqlite                    3.31.1                 h7b6447c_0
tk                         8.6.8                  hbc83047_0
tqdm                      4.42.1                 py_0
urllib3                   1.25.8                 py37_0
wheel                     0.34.2                 py37_0
xz                         5.2.4                  h14c3975_4
yaml                      0.1.7                  had09818_2
zlib                      1.2.11                 h7b6447c_3
(base) student@dpa2019s-team:~$
```

בנוסף, ניתן לקרוא ל interpreter של פייתון ע"י הקלדת:

```
python
```

ולקבל פלט דומה:

```
(base) vmadmin@dpa2020s-0020:~$ python
Python 3.8.5 (default, Sep 4 2020, 07:30:14)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
```

יש לוודא כי הותקן Python <= 3.8 והמילה "Anaconda" מופיעה. ניתן לצאת ע"י לחיצה על ctrl+d.

## יצירת סביבה וירטואלית

לאחר התקנת Miniconda, ניתן כבר להשתמש ישירות בפיתון. עם זאת, אנו ניצור סביבה וירטואלית לכל תרגיל בית בנפרד. בדיקת תרגילי הבית תקים סביבה וירטואלית זהה לזו שתגדירו בשלבים הבאים.

### יצירת קובץ environment.yml

ישנן מספר דרכים ליצור סביבות וירטואליות תחת conda. אנו נשתמש בקובץ קונפיגורציה לעשות זאת. קובץ הקונפיגורציה מכיל את כל הפרטים לגבי

באמצעות עורך טקסט לבחירתכם, צרו קובץ בשם environment.yml בפורמט הבא (YAML):

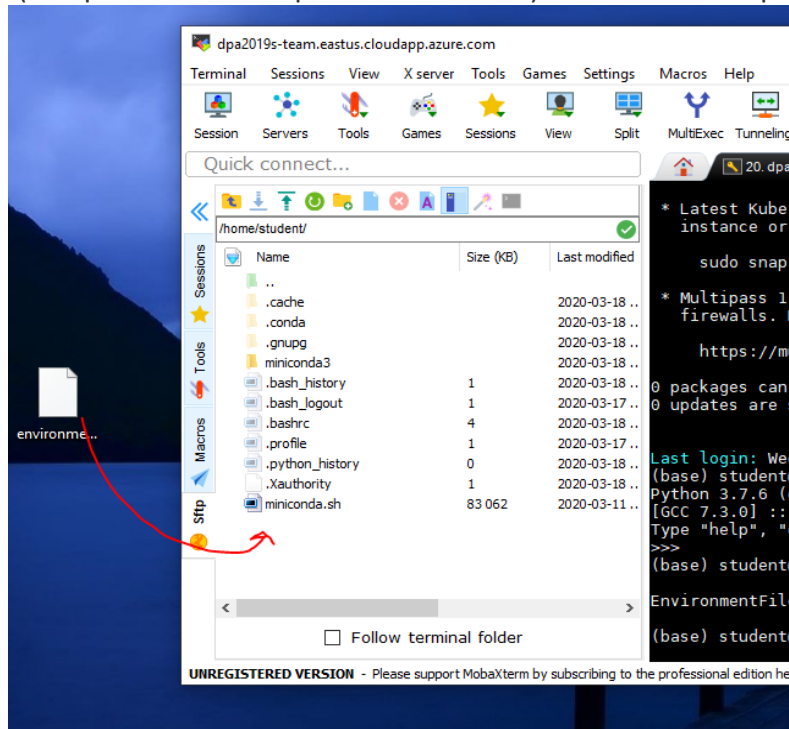
```
environment.yml
1  name: hwX_env
2  dependencies:
3    - python=3.8
4    - pip
5    - pip:
6      - numpy==1.20.2
7      - pandas==1.2.4
```

תחת שדה ה **name** ניתן שם לסביבה הוירטואלית.

תחת שדה ה **dependencies** יש לפרט באיזו גרסת פיתון הסביבה תשתמש ויש להתקין את מנהל החבילות pip עבור הסביבה. ניתן לפרט חבילות נוספות שיוותקנו ישירות ע"י מנהל הסביבה (conda) אך נתקין את רוב החבילות תחת מנהל החבילות pip.

תחת שדה ה pip נפרט את החבילות שנרצה להתקין בסביבה (שקול לקובץ requirements).

נשמור את הקובץ ונעתיק אותו למכונה שלנו (ב-MobaXterm ניתן לגרור לחלונית הקבצים).



לאחר ששמרנו את הקובץ במכונה, נקליד:

**conda env create -f environment.yml**

- פקודה זו תקים סביבה וירטואלית חדשה עם השם והתלויות שהגדרנו בקובץ environment.yml.



אחר יצירת הסביבה, נקליד:

**conda activate <env\_name>**

ונוודא שהסביבה הופעלה ע"י שינוי של ה prompt מ- (base) ל- (<env\_name>), לדוגמא:

```
(base) vmadmin@dpa2020s-0020:~$ conda activate hwX_env
(hwX_env) vmadmin@dpa2020s-0020:~$
```

כעת בכל פעם שאנו רוצים להפעיל את הסביבה שלנו (למשל אם התחברנו למכונה מחדש) נצטרך לקרוא ל `conda activate`. ניתן לגרום לסביבה שהגדרנו לעלות באופן אוטומטי בכל פעם שמתחברים למכונה ע"י עריכת קובץ ה `bashrc` (לא חובה). פרטים:

<https://askubuntu.com/questions/849470/how-do-i-activate-a-conda-environment-in-my-bashrc>

הערות חשובות:

1. הקובץ `environment.yml` רגיש לפורמט ולתחביר. יש להקפיד על סינטקס נכון (רווחים, איות, שימוש ב'=' עבור `conda` ו'==' עבור `pip`). קובץ עם שגיאות תחביר ייכשל בהקמת הסביבה.
2. מומלץ מאוד לציין באופן מפורש את גרסת החבילה שאנו מתקינים ב `pip` עם "==" . כאשר לא מציינים זאת, `pip` מוריד את הגרסה העדכנית ביותר. אם לא תצינו את גרסת החבילה, ייתכן שבעת בדיקת התרגיל תוקם סביבה עם גרסה שונה מזו המותקנת אצלכם.
3. ניתן להוסיף ספריות לסביבה אחרי הקמתה באמצעות פקודת "`pip install x`". עם זאת, **חייבים להוסיף את החבילה לקובץ ה `environment.yml`** על מנת שחבילה זו תותקן בעת בדיקת התרגיל שלכם.
4. במידה והקמת הסביבה נכשלה מסיבה כלשהי, ניתן למחוק את התיקייה שבה הסביבה הותקנה ע"י:

**rm -rf /home/student/miniconda3/envs/<env\_name>**

ולחזור על פקודת `env create` שוב.

5. פקודות `conda` שימושיות נוספות:

**conda deactivate**

- יציאה מסביבה פעילה (וחזרה ל-base)

**conda create -n <env\_name> python=3.8**

- יצירת סביבה ללא קובץ `environment.yml`

**conda remove -name <env\_name> --all**

- מחיקת סביבה (זהירות - ימחק את כל הספריות שהתקנתם בה)

שימו לב:

על אחריותכם לוודא שקבצי ה **environment.yml** עם כל הספריות הדרושות להרצת הקוד שלכם בעת ההגשה. לפני ההגשה מומלץ מאוד ליצור סביבה חדשה עם קובץ ה **environment.yml**, להריץ את הקוד ולוודא שאין שגיאות **import error** למינהן. כאמור, בדיקת התרגילים תיצור סביבה וירטואלית לפי קובץ ה **environment.yml** באופן אוטומטי. קוד שלא יוכל לרוץ עקב קונפיגורציה לא טובה של סביבת העבודה יגורר הורדת נקודות בתרגיל.

## הגשת תרגילי בית דרך GitHub

העבודה על תרגילי הבית תדרוש שימוש בסיסי במנוע VCS (Version Control System). אנו נדגים שימוש ב git ובשירות האחסון החינמי GitHub. העבודה עם VCS מאפשרת ניהול גרסאות קוד באופן מבוצר, נוח ובטיחותי. תוכלו לעבוד על הפרויקטים שלכם ממחשבים שונים ו/או ע"י אנשים שונים.

הגשת תרגילי הבית תעשה ע"י לינק ל repository שלכם ב-GitHub. כל תרגיל בית ידרוש repository נפרד.

## הרשמה ל-GitHub

במידה ואין לכם, עליכם לפתוח חשבון GitHub בקישור הבא:

<https://github.com/join?source=header-home>

מומלץ לבצע זאת עם המייל הטכניוני שלכם. יש ל-GitHub שירותי פרימיום לסטודנטים (למשל PyCharm Professional לשנתיים בחינם). ניתן גם לקשר חשבון קיים למייל הטכניוני שלכם. פרטים נוספים פה: <https://education.github.com/pack>

## יצירת repository ב-GitHub

תחת עמוד הבית שלכם ב-GitHub יש ללחוץ על New (או בקישור הבא: <https://github.com/new>)

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner: tomron27 / Repository name \*: test\_repo ✓ ←

Great repository names are short and memorable. Need inspiration? How about [furry-waddle?](#)

#### Description (optional)

This is a test repository for HW submission in 094295

☒ **Public** ←  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository ←

בחלון שנפתח יש:

- לתת ל repository שם אינפורמטיבי (לדוגמא: 094295\_hw1)
- לתת תיאור קצר ל repository (לא חובה)

- לוודא שה repository על מצב public
- ללחוץ על Create repository

כעת ה repository שלכם הוקם. ניתן להתחבר אליו מכל מחשב שיש בו git מותקן. המכוונה שלכם ב Azure מגיעה עם git מותקן מראש. להורדת git למחשב האישי שלכם (אין צורך ב GUI):

<https://git-scm.com/downloads>

לאחר שלחצנו Create repository, בחלון הבא נקבל הנחיות כיצד להתחבר ל repository שהקמתם:

tomron27 / test\_repo

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

**Quick setup — if you've done this kind of thing before**

Set up in Desktop or HTTPS SSH [https://github.com/tomron27/test\\_repo.git](https://github.com/tomron27/test_repo.git)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# test_repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/tomron27/test_repo.git
git push -u origin master
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/tomron27/test_repo.git
git push -u origin master
```

**...or import code from another repository**

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

לפני שנחבר את ה repository למכוונה שלנו ב Azure, נבניס בה את פרטי החיבור ל GitHub שלנו. בחלון הטרמינל יש להקליד:

**`git config --global user.email "you@example.com"`**

**`git config --global user.name "Your Name"`**

- יש להבניס את המייל שאיתו נרשמתם ל GitHub ואת השם שלכם

כעת ניצור תיקייה חדשה:

**`mkdir <repo_name>`**

**`cd <repo_name>`**

ולאחר מכן נבצע את הפקודות המסומנות למעלה:

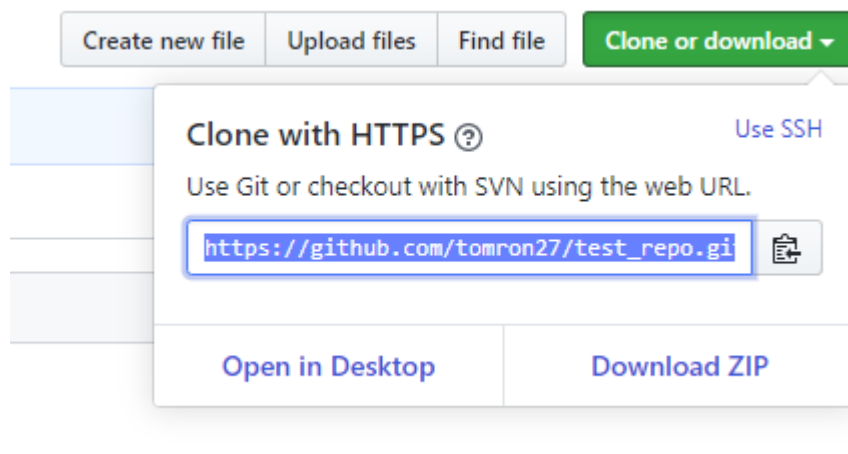
```
echo "#<repo_name>" >> README.md
git init
git add README.md
git remote add origin https://github.com/<username>/<repo_name>.git
git push -u origin master
```

הפלט יראה כך:

```
2020-03-11.. (test_env) student@dpa2019s-team:~$ mkdir test_repo
2020-03-18.. (test_env) student@dpa2019s-team:~$ cd test_repo
2020-03-18.. (test_env) student@dpa2019s-team:~/test_repo$ echo "#test_repo" >> README.md
(test_env) student@dpa2019s-team:~/test_repo$ git init
Initialized empty Git repository in /home/student/test_repo/.git/
(test_env) student@dpa2019s-team:~/test_repo$ git add README.md
(test_env) student@dpa2019s-team:~/test_repo$ git commit -m "first commit"
[master (root-commit) 57f135c] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
(test_env) student@dpa2019s-team:~/test_repo$ git remote add origin https://github.com/tomron27/test_repo.git
(test_env) student@dpa2019s-team:~/test_repo$ git push -u origin master
Username for 'https://github.com': tomron27
Password for 'https://tomron27@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 220 bytes | 220.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/tomron27/test_repo.git
* [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
(test_env) student@dpa2019s-team:~/test_repo$
```

שימו לב שלאחר פקודת ה push יש להזין את שם המשתמש והסיסמה שלכם ל GitHub (באדום).  
ע"מ לוודא שה push הצליח, תוכלו לפתוח את ה repository שלכם באתר של GitHub ולראות שנוצר  
בו קובץ README.md:

ע"מ להעתיק (clone) את ה repository שהגדרנו למחשב אחר, יש ללחוץ על Clone or download:



להעתיק את הקישור ולהריץ במחשב שתרצו:

***git clone <GitHub URL>***

הערות:

- מומלץ לבצע היכרות עם פקודות git הנפוצות: commit, push, pull, clone, merge  
<https://rogerdudler.github.io/git-guide/>
- לרוב סביבות העבודה יש אינטגרציה נוחה עם git, ניתן לבצע את רוב הפקודות דרך סביבת העבודה עצמה ולא דרך ה command line. בין היתר ל PyCharm:  
<https://www.jetbrains.com/help/pycharm/using-git-integration.html>

לכל שאלה או בעיה שעולה אתם מוזמנים להשתמש בפורום השאלות שבמודל.

בהצלחה!