AWS

AWS הוא public cloud של חברת amzon. קיימים אצלה data centers ברחבי העולם, וחלקם עדיין בשלבי הקמה.

AWS מספקת מגוון רחב של שירותים, הכוללים databasים, שרתי אפליקציה ועוד…

ניתן להתחבר אליו באמצעות ה-free tier שהוא מספק, עם זאת שירותים נוספים יעלו תשלום כלשהו.

[EC2 - Elastic Compute Cloud](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html)

מדובר ב-service הנפוץ ביותר של AWS, המתעסק ב-VMs שנקראים למעשה instances.

באמצעותו ניתן לנהל את ה-instances השונים ומרכיביהם.

**תכונות עיקריות:**

* מאפשר ניהול של servers ב-amazon cloud.
* ניתן להגדיל ולהקטין בקלות את המשאבים הדרושים לנו (סקלביליות)
* ניתן לשלב אותו עם serviceים אחרים של AWS, לדוגמא: S3

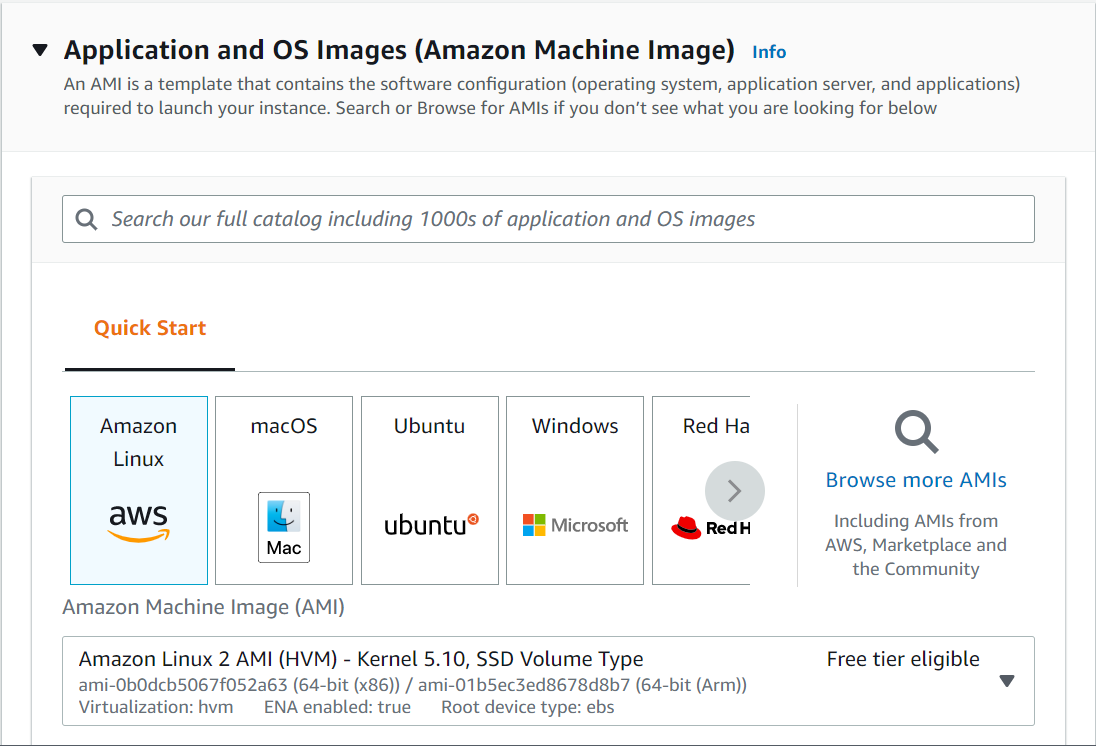
**תשלום:**

קיימות 4 דרכים לתשלום על service של EC2:

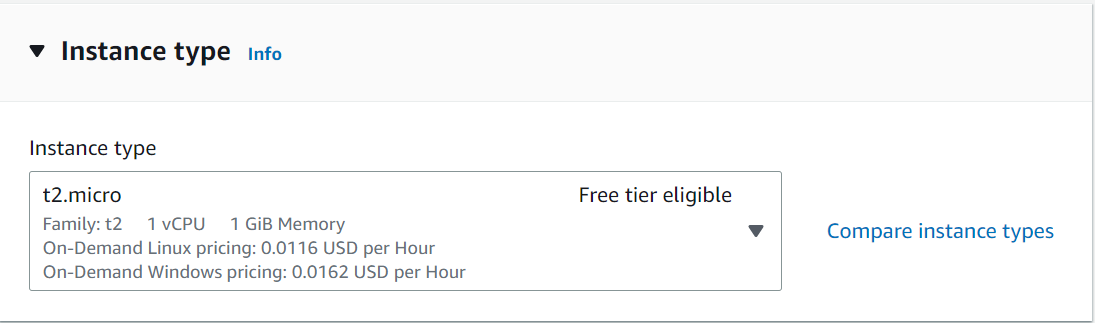
* תשלום לפי שעה
* תשלום מראש לשנה עד 3 שנים
* הצעת מחיר אישית על service של EC2 שלא נעשה בה שימוש - החיסרון, ברגע שמישהו מציע מחיר גבוה יותר, המכונה שלך תימחק
* קנייה של שרת פיזי המוקדש אישית לך/לחברה

**הרכיבים של EC2:**

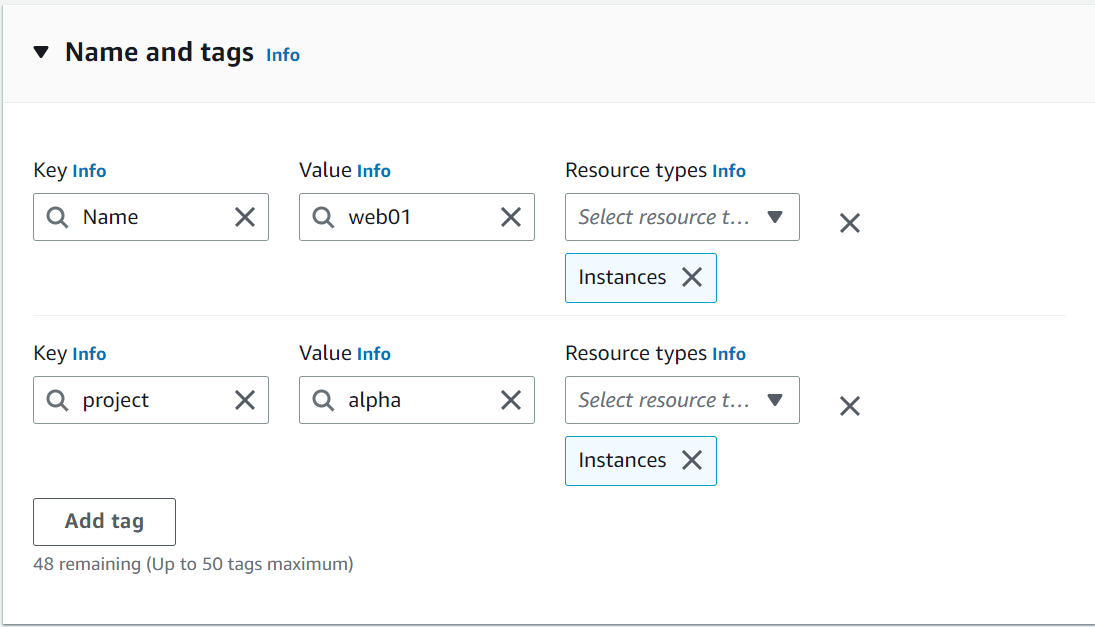
* AMI - Amazon Machine Image - בדומה לDockerfile או ל-Vagrantfile, ל-EC2 יש את ה-AMI, שהוא מכיל את האינפורמציה הדרושה על מנת להקים instance - הכוונה ל-server ב-amazon cloud.



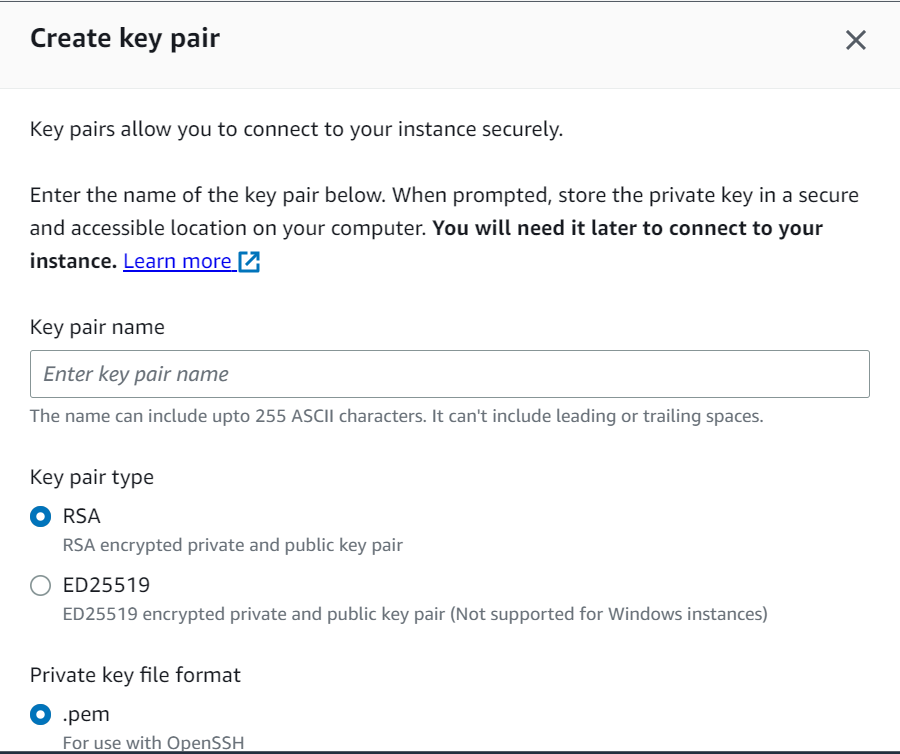
* [Instance Type](https://aws.amazon.com/ec2/instance-types/) - רכיב זה מציין את כמות החומרה הנדרשת עבור ה-instance שלנו, כמות CPU, RAM וכו'



* EBS - Amazon Elastic Block Store - רכיב זה מהווה את ה-storage של ה-instance. קיים storage של בערך 8 ג'יגה עבור linux ו-30 ג'יגה עבור windows, בנוסף ה-EC2 יכול לעשות שימוש עם serviceים נוספים של storage, מעבר לרכיב הספציפי הזה.
* security group - מהווה מעין firewall של ה-instance, אשר ששולט בתקשורת היוצאת והנכנסת שלו. לכל SG ניתן להגדיר roles השייכים לו.
* tags - ניתן לעשות שימוש ב-tags על מנת למפות בקלות רכיבים ב-EC2, ניתן לתת tags לפי קטגוריות כמו שם, פרוייקט וכו'



* keys - על מנת להתחבר לinstance יש לעשות שימוש ב-public key. ב-linux ניצור key ונתחבר באמצעות SSH, עבור windows אנו נשתמש ב-key על מנת ליצור סיסמא להתחברות.

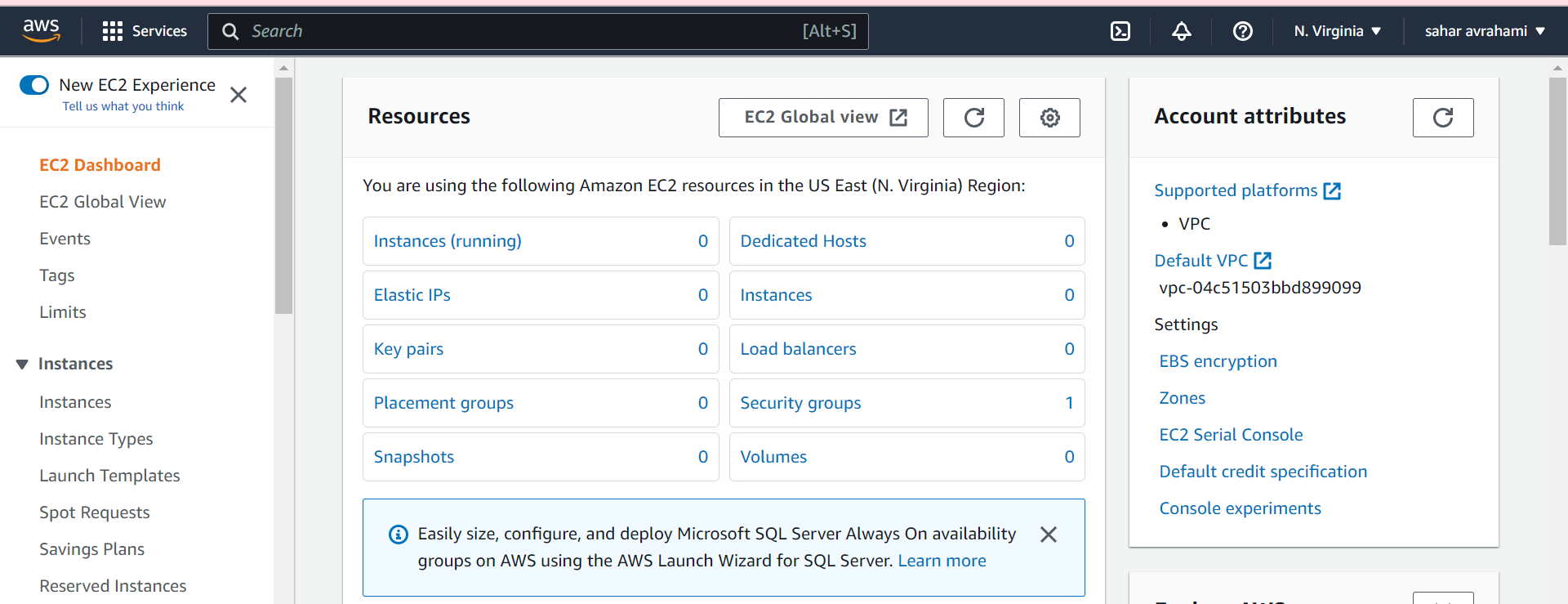


* Elastic IP’s - מדובר ב-IP סטטי, כלומר, גם לאחר כיבוי ה-instance. ה-IP, לא יעלם, אלא ישאר קבוע. ניתן ליצור עד חמישה כאלה, אם נרצה עוד נצטרך לרכוש את האופציה הזו מ-AWS.

**שלבים להקמת EC2:**

1. נבחר AMI
2. נבחר instance type
3. נגדיר את ה-instance
4. נוסיף storage
5. נוסיף tags
6. נגדיר security group
7. נבדוק תקינות

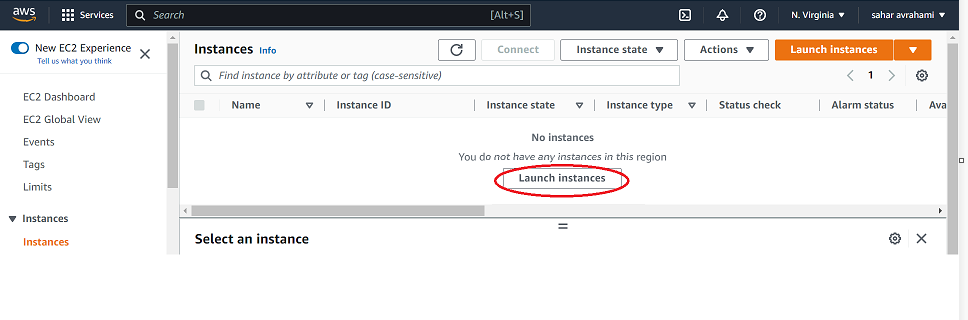
**EC2 Dashboard:**

****

*כאן ניתן לראות את כלל המשאבים של ה-EC2 שקיימים אצלנו*

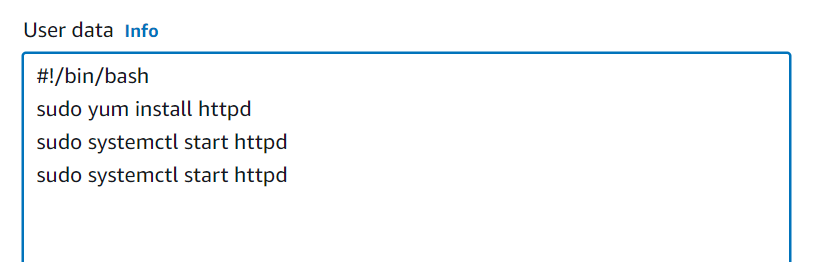
**יצירת instance:**

על מנת ליצור instance יש להיכנס אל הטאב instnaces וללחוץ על הכפתור lunch instance

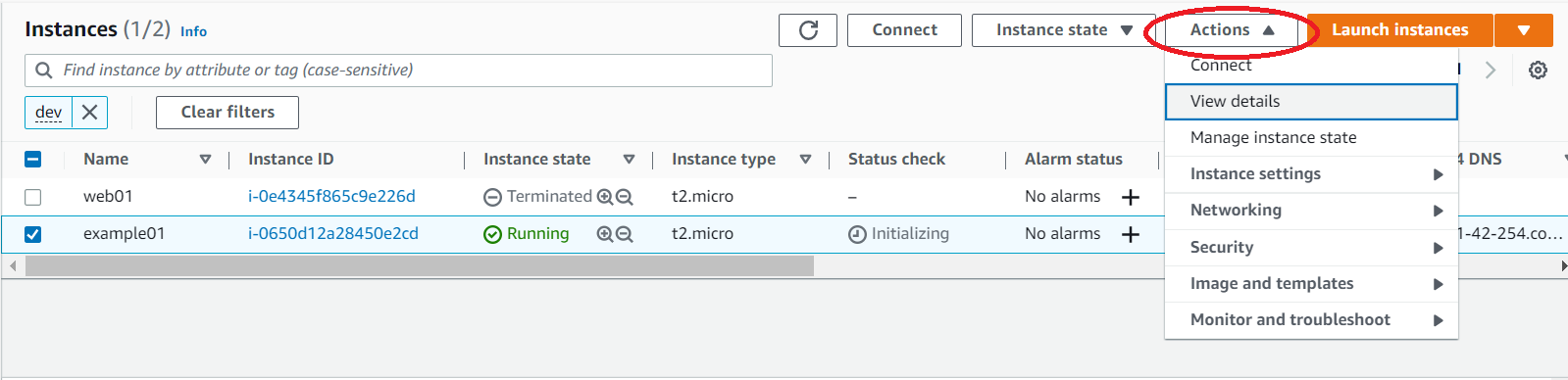


בעת יצירת ה-instance אנו נעבור על השלבים שתוארו קודם לכן. בנוסף, קיימים כל מיני סוגים של advanced details שניתן לקבוע לכל instance.

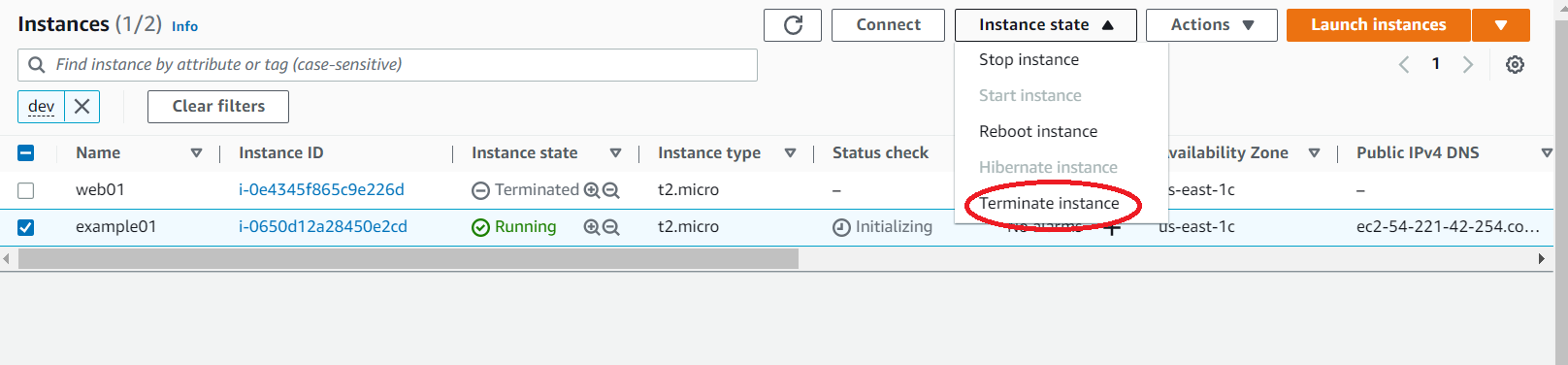
אחד מהם הוא user data, שם ניתן לכתוב שם פקודות שירוצו בעת עליית ה-instance, יש לכתוב את הפקודות לפי סוג ה-instance המתאים, למשל: ל-linux יש לכתוב ב-bash.



לאחר יצירת ה-instance, יהיה ניתן לבצע עליו מגוון רחב של פעולות, חלקם ניתן לבצע בזמן הריצה שלו וחלקם ניתן לבצע רק כאשר הוא מכובה.



על מנת למחוק instance יש לבחור באופציה: terminate instance



[AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)

למעשה מדובר ב-command line של AWS.

כלומר, כל פעולה שנרצה לבצע דרך ה-GUI בצורה גרפית. נוכל לבצע גם באמצעות הרצת פקודות ב-CLI.

מצורף [PDF](https://drive.google.com/open?id=1mqc_nVQwL0ItXiJvGwTZ-cRBG9ZyyLjy) עם הדרך להתקנת ה-CLI, האופן בו יוצרים יוזר עם ההרשאות המתאימות על מנת להריץ פקודות ב-CLI ולהתחבר לענן הנוכחי.

בנוסף, אין צורך לזכור בעל פה את כל הפקודות האפשריות ב-AWS CLI, מכיוון שקיימים המון לכך.

[באתר שלהם](https://awscli.amazonaws.com/v2/documentation/api/latest/index.html) מפורטות כל הפקודות שניתן לבצע, באופן מאוד קל ונגיש.

EBS - Elastic Block Storage

מדובר למעשה ב-virtual hard disk של ה-EC2 instance (נקראים גם פשוט volumes)

זהו אחסון פשוט מסוג block storage, אשר אחראי להרצת EC2 instances, לאחסון מידע מה-DB ועוד

כאשר אנו נגדיר EBS, אנו נגדיר לו גם availability zone. חשוב לשים לב שה-instance שאנו נרצה לקשר אליו את ה-volume של ESB חייב להיות עם אותו AZ.

קיימים מספר סוגים של ESB, כאשר כל אחד מהם מיועד לשם מטרה אחרת:

* SSD - אחסון פשוט, נעשה בו שימוש ברוב המקרים.
* IOPS - משומש עבור DBים גדולים
* Throughput Optimized HD - משומש עבור Big DATA
* Cold HDD - משומש עבור file server
* Magnetic - האחסון האיטי והפשוט ביותר, נעשה בו שימוש בעיקר עבור גיבויים וארכובים

*הערה:*

*על מנת שנוכל להשתמש ב-EBS ב-free tier, יש לאחסן* ***בסך הכל*** *עד 30 ג'יגה*

בנוסף ל-volumes, קיימת גם אופציה של snapshots

שזה למעשה גיבוי של ה-volume הנוכחי.

נוכל ליצור מ-snapshot כלשהו volume חדש, נרצה להשתמש ביכולת זו על מנת לקבל בחזרה מידע שאבד לנו

על מנת להגדיל volume כלשהו, או על מנת לשנות את ה-AZ שלו.

[ELB - Elastic Load Balancer](https://docs.aws.amazon.com/elasticloadbalancing/)

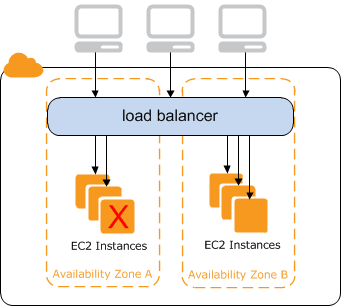
בארכיטקטורה בה יש לנו מספר גדול של instances, לדוגמא: web servers, ונרצה לנתב את המשתמשים שלנו לכמה instanceים על מנת שלא ניצור עומס על instance אחד וכל המשתמשים יתחברו רק אליו.

כאן בא לידי ביטוי ה-ELB.

ל-ELB קיים frontend port, איתו הוא יאזין, בנוסף קיים לו גם backend port, אשר מבוסס על ה-backend port שקיים ב-servers הנוכחים.

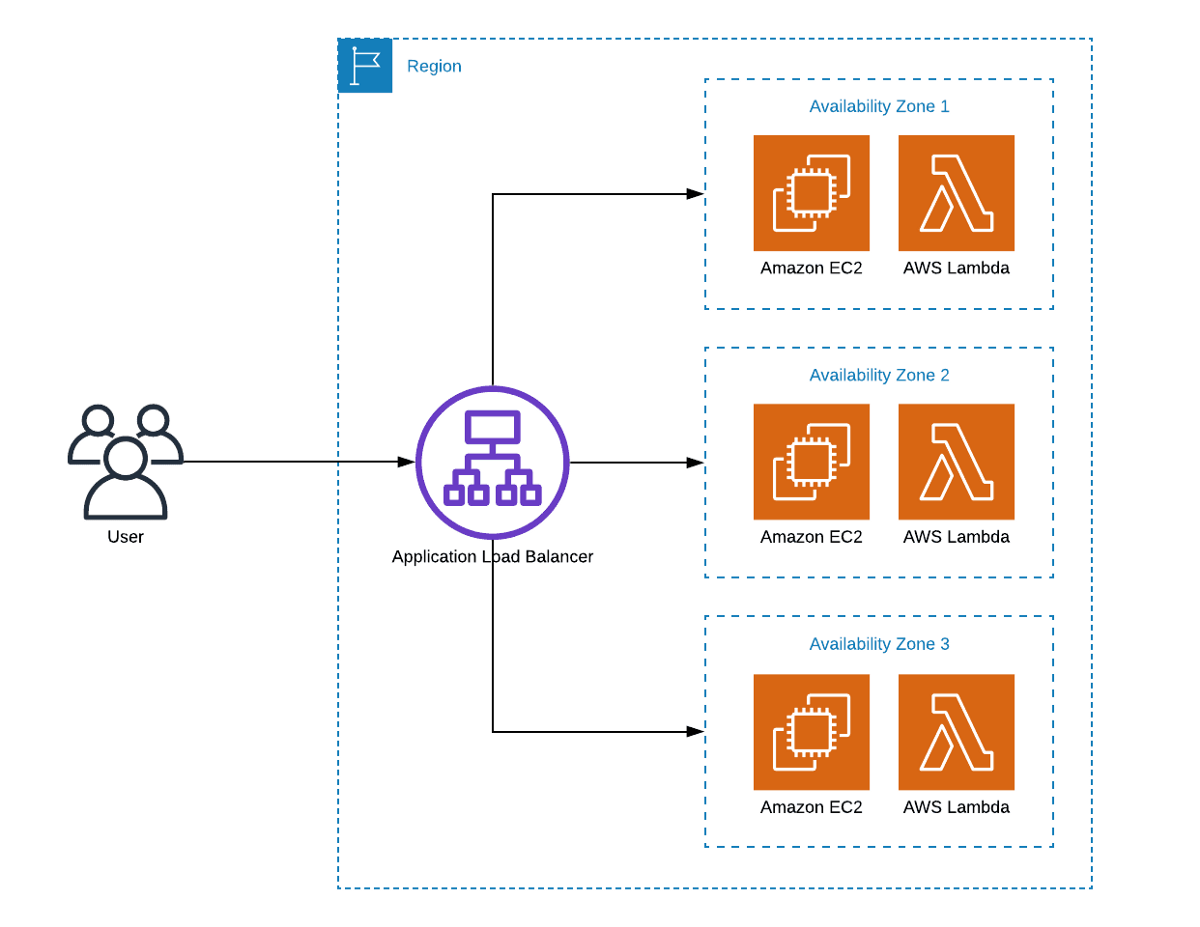
**סוגי Load Balancers:**

* Classic Load Balancer - הסוג הפופולארי ביותר של ELB - הוא מנתב את תעבורת הרשת על בסיס שכבת הרשת או שכבת האפליקציה (במודל 7 השכבות). סוג זה אידיאלי עבור תעבורה בין מספר EC2 instances.



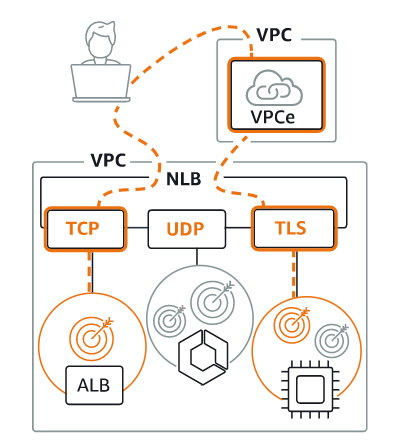
* Application Load Balancer - נעשה שימוש בסוג זה בעיקר אצל web servers, כיוון שהוא מתמחה בניתוב תעבורה ברמת שכבת האפליקציה (במודל 7 השכבות)

לדוגמא במצב בו משתמש ניגש לנ-URL מסויים, אז ה-load balancer יכול לנתב אותו ל-instance מסויים, וכאשר הוא ניגש ל-URL אחר, הוא יכול לנתב אותו ל-instance אחר.



* Network Load Balancer - הסוג היקר ביותר, שמתמחה בניתוב של מספר עצום של בקשות (מיליוני בקשות בשניה). בנוסף, סוג זה מתמחה בניתוב בקשות ברמת השכבה הרביעית במודל 7 השכבות.

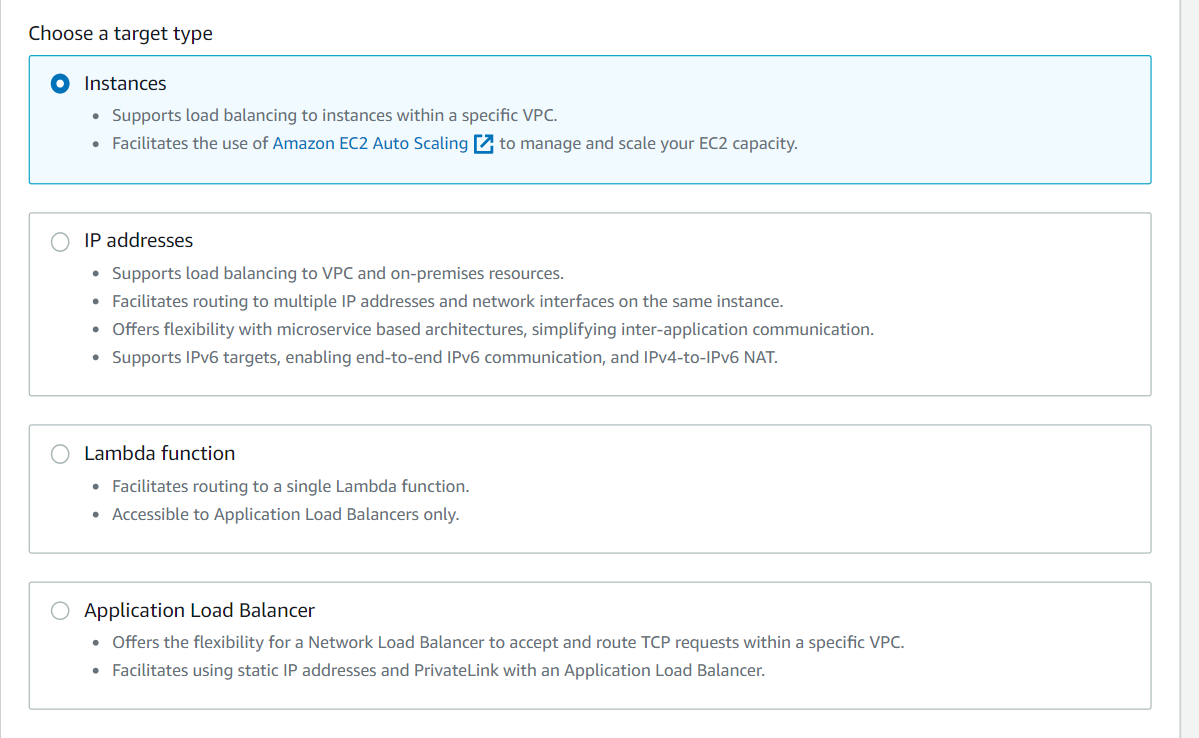
יתרון נוסף של סוג זה אל מול לשניים האחרים הוא שבו ניתן לעשות שימוש ב-static IP.



על מנת שנוכל להגדיר ELB, נצטרך להגדיר לפני כן אובייקט הנקרא: [**target group.**](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html)

כל target group מיועד עבור לנווט בקשות אל היעד שנבחר לו.

יעדים אופציונליים:



ברגע שנבחר ל-TG שלנו rule כלשהו, תעבורה תעבור באמצעות הפורט שנגדיר באופן סדיר, ותבדוק אם התעבורה תקינה (healthy) או לא (unhealthy). בנוסף, אנו נגדיר כל כמה זמן הוא יבצע את הבדיקה, כמה בדיקות נחשבות למצב תקין, כמה בדיקות נחשבות למצב לא תקין, איזה נתיב נרצה לבדוק וכו'...

Cloudwatch

זהו service של AWS המיועד לניטור של כלל המשאבים בענן: CPU, RAM, network packets, billing ועוד…

בנוסף, ל-cloud watch יש אופציה לייצא alarms: התראה כאשר מתרחש אירוע מסוים, למשל כאשר ה-CPU של instance מסויים עובר את ה - 60%

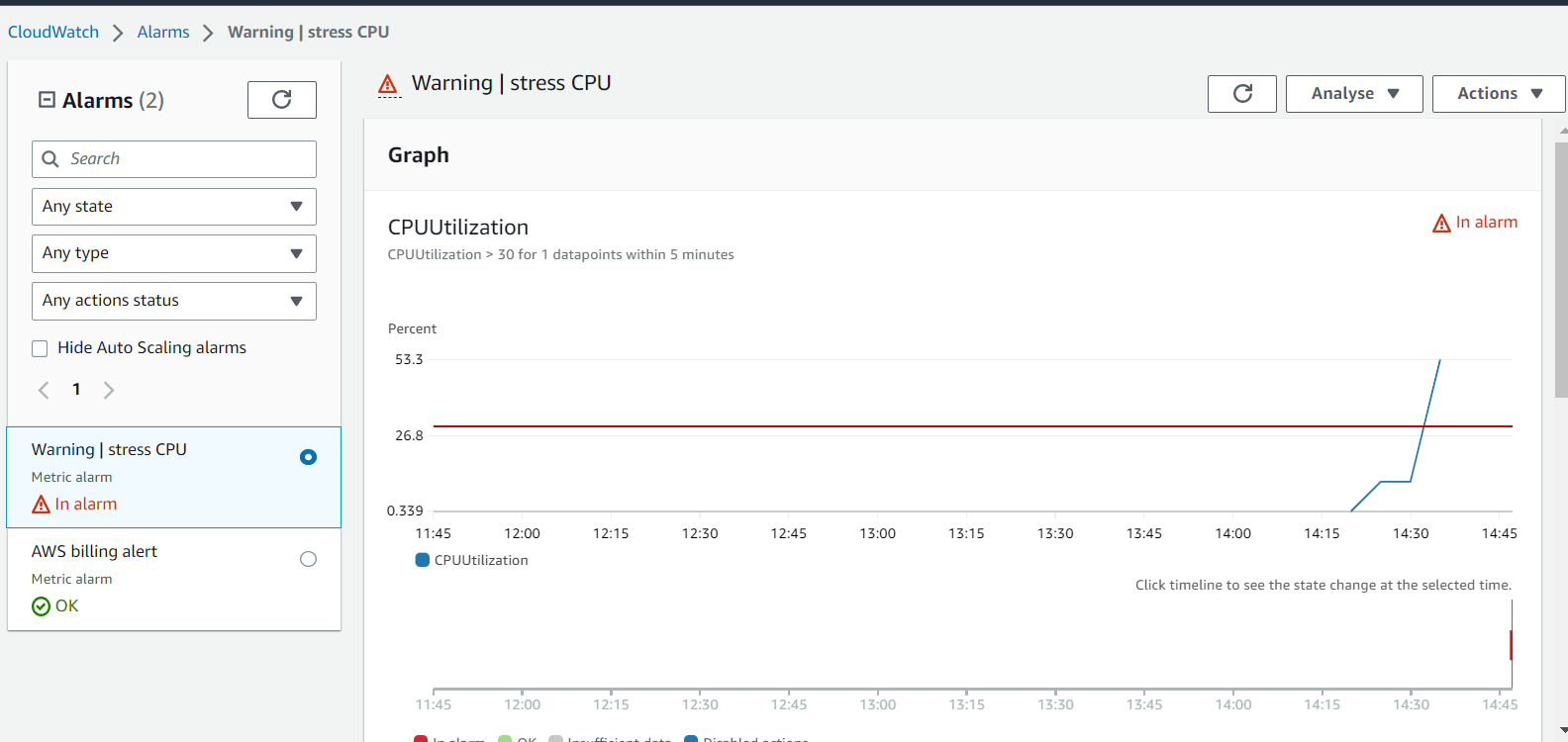
בנוסף לעניין הניטור, ל-cloudwatch יש יכולת של הצגת לוגים על אובייקטים ב-AWS.

ניתן להגדיר התרעה חדשה באמצעות:

cloudwatch → all alarms → create alarm

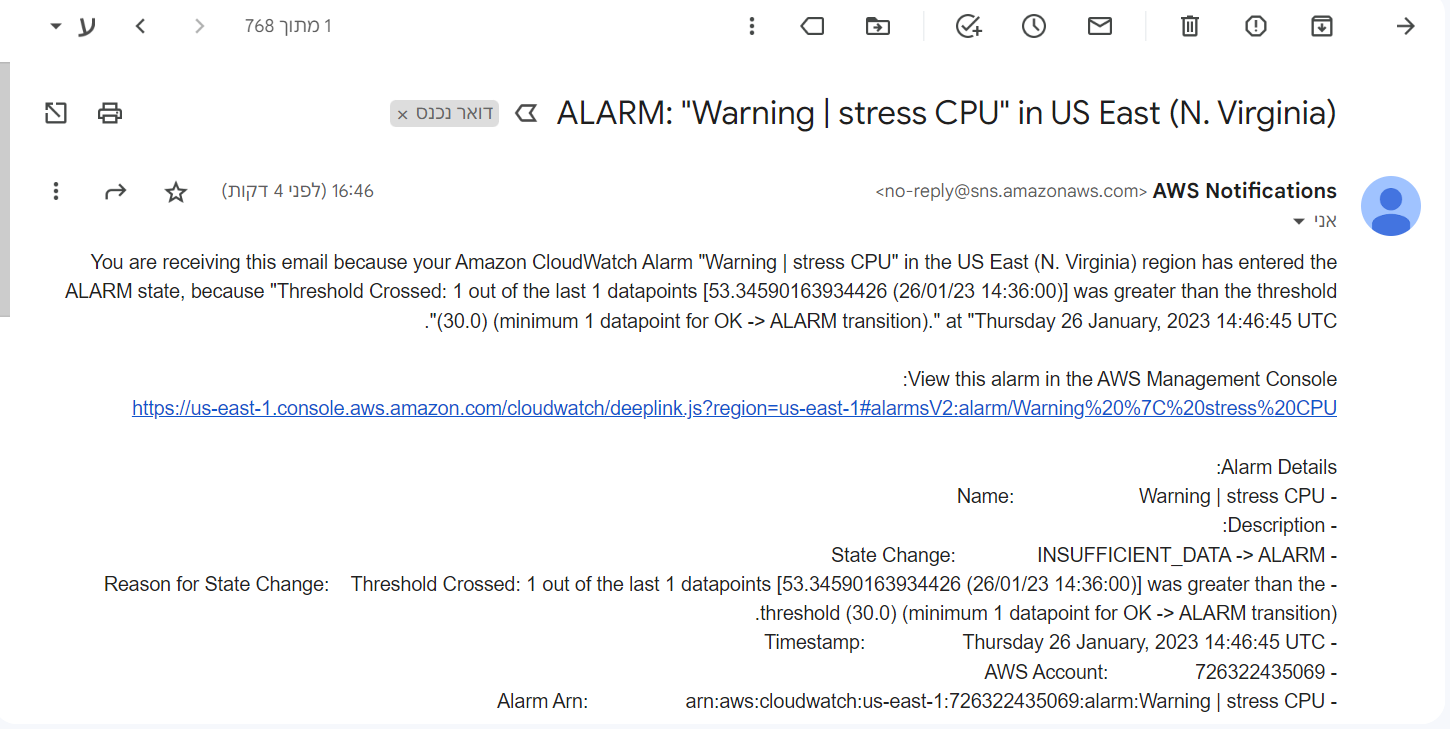
שם נכניס את הפרטים הרצויים לפי הצורך שלנו

לדוגמא כאן, הגדרתי alarm שיתריע אם ה-CPU של instance עובר את ה-30% במשך ליותר מ-5 דקות:



ניתן לראות ממש באופן חזותי על פי הגרף את המצב הנוכחי של ה-instance

ברגע הופעת ה-alarm, אני אקבל מייל עם ההתרעה שהגדרתי:

האימייל הזה מפרט את כלל הפרטים של ה-alarm.

[EFS - Elastic File System](https://docs.aws.amazon.com/efs/latest/ug/whatisefs.html)

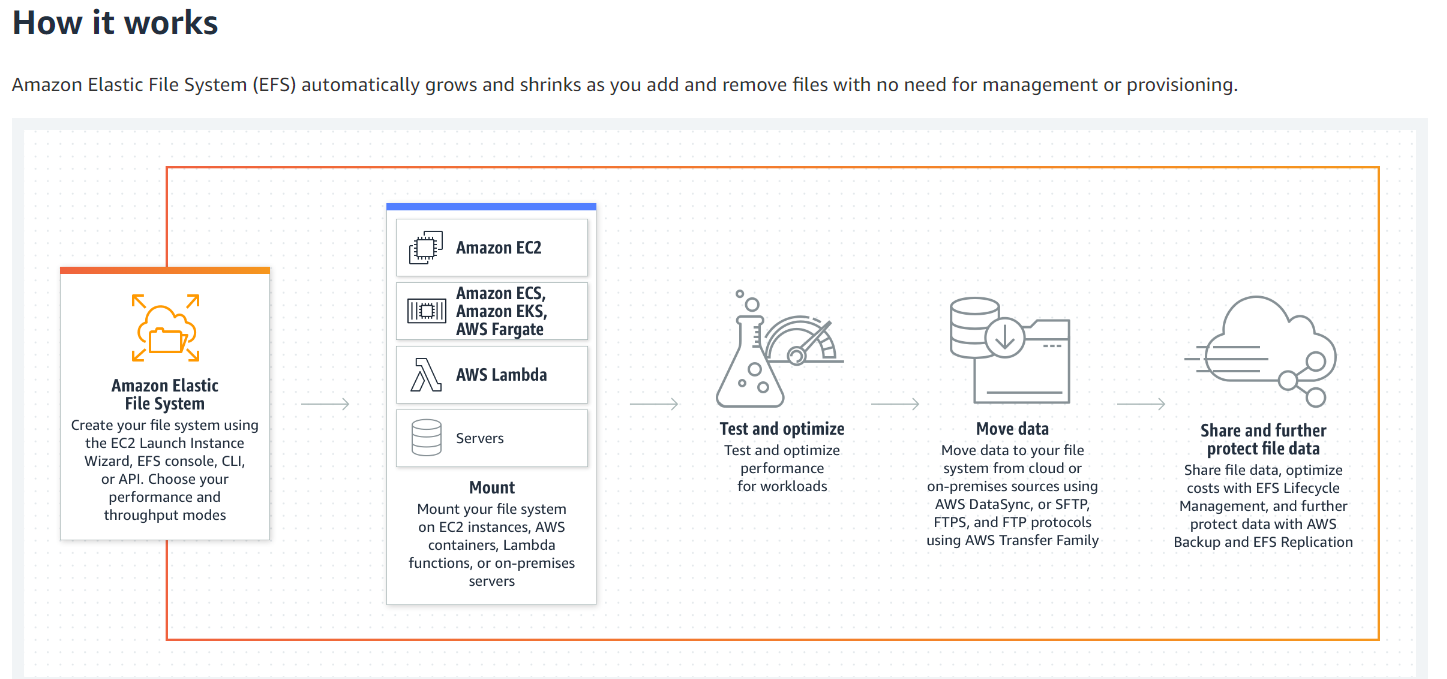
זהו service מאוד שימושי של AWS, שמהווה למעשה כ-shared storage ל-instanceים

לעומת EBS, שקיים אחד ממנו לכל instance, ניתן ליצור EFS אחד ולשתף אותו בין מספר instanceים (כמו שרשמתי מקודם: **shared** storage)

*הערה: service זה אינו רלוונטי ל-windows instances*

על מנת שנוכל ליצור EFS, יש ליצור קודם לכן security group - הוא מאפשר למעשה למי תהיה את היכולת לגשת ל-shared storage שניצור

*הערה: כשנצטרך לסמן בעת יצירת ה-EFS את ה-security group, נסמן אותו בכל ה-zones האפשריים*

**