הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

המעבדה למערכות תוכנה מרושתות

מחשוב ענן של אמאזון - AWS

חוברת הכנה

קומה 3 חדר 375

החדר נמצא במסדרון בין פישבך למאייר

דרישות קדם: מומלץ לקחת את הקורס "רשתות ואינטרנט 1" או "מבוא לרשתות מחשבים" לפני הניסוי או במקביל אליו. חובה קורס תכנות.

חשוב: יש למלא טופס משוב בתום הניסוי!!!

המאחר ביותר מ – 15 דקות לא יורשה לבצע את הניסוי!

המעבדה למערכות תוכנה מרושתות

רועי מטרני - טל. :3220 דוא"ל: roym@ee

בטיחות

כללי:
הנחיות הבטיחות מובאות לידיעת הסטודנטים כאמצעי למניעת תאונות בעת ביצוע ניסויים ופעילות במעבדה למערכות תוכנה. מטרתן להפנות תשומת לב לסיכונים הכרוכים בפעילויות המעבדה, כדי למנוע סבל לאדם ונזק לציוד.
אנא קראו הנחיות אלו בעיון ופעלו בהתאם להן.

מסגרת הבטיחות במעבדה:

- אין לקיים ניסויים במעבדה ללא קבלת ציון עובר בקורס הבטיחות של מעבדות ההתמחות באלקטרוניקה (שהינו מקצוע קדם למעבדה זו).
- לפני התחלת הניסויים יש להתייצב בפני מדריך הקבוצה לקבלת תדריך ראשוני הנחיות בטיחות.
- אין לקיים ניסויים במעבדה ללא השגחת מדריך ללא אישור מראש.

מדריך הקבוצה אחראי להסדרים בתחום פעילותך במעבדה; נהג על פי הוראותיו.

עשה ואל תעשה:

- יש לידע את המדריך או את צוות המעבדה על מצב מסוכן וליקויים במעבדה או בסביבתה הקרובה.
- לא תיעשה במזיד ובלי סיבה סבירה, פעולה העלולה לסכן את הנוכחים במעבדה.
- אסור להשתמש לרעה בכל אמצעי או התקן שסופק או הותקן במעבדה.
- היאבקות, קטטה והשתטות אסורים. מעשי קונדס מעוררים לפעמים צחוק אך הם עלולים לגרום לתאונה.
- אין להשתמש בתוך המעבדה בסמים או במשקאות אלכוהוליים, או להיות תחת השפעתם.
- אין לעשן במעבדה ואין להכניס דברי מאכל או משקה.
- בסיום העבודה יש להשאיר את השולחן נקי ומסודר.

בטיחות חשמל:

- בחלק משולחנות המעבדה מותקנים בתי תקע ("שקעים") אשר ציוד המעבדה מוזן מהם. אין להפעיל ציוד המוזן מבית תקע פגום.
- אין להשתמש בציוד המוזן דרך פתילים ("כבלים גמישים") אשר הבידוד שלהם פגום או אשר התקע שלהם אינו מחוזק כראוי.
- אסור לתקן או לפרק ציוד חשמלי כולל החלפת נתיכים המותקנים בתוך הציוד; יש להשאיר זאת לטפול הגורם המוסמך.

- אין לגעת בארון החשמל המרכזי, אלא בעת חירום וזאת - לצורך ניתוק המפסק הראשי.

מפסקי לחיצה לשעת חירום:

- במעבדה ישנם מפסקים ראשיים להפסקת אספקת החשמל. זהה את מקומם.
- בעת חירום יש להפעיל מפסקי החשמל הראשיים.

בטיחות אש, החייאה ועזרה ראשונה:

- במעבדה ממוקמים מטפי כיבוי אש זהה את מקומם.
- אין להפעיל את המטפים, אלא בעת חירום ובמידה והמדריכים וגורמים מקצועיים אחרים במעבדה אינם יכולים לפעול.
-

יציאות חירום:

- בארוע חירום הדורש פינוי, כגון שריפה, יש להתפנות מיד מהמעבדה.

דיווח בעת אירוע חירום:

- יש לדווח **מידית** למדריך ולאיש סגל המעבדה.
- המדריך או איש סגל המעבדה ידווחו מיידית לקצין הביטחון בטלפון; 2740, 2222.
- **במידה ואין הם יכולים** לעשות כך, ידווח אחד הסטודנטים לקצין הביטחון.

לפי הוראת קצין הביטחון, או כאשר אין יכולת לדווח לקצין הביטחון, יש לדווח, לפי הצורך:

משטרה 100,

מגן דוד אדום 101,

מכבי אש 102,

גורמי בטיחות ו/או ביטחון אחרים.

- בנוסף לכך יש לדווח ליחידת סגן המנמ"פ לעניני בטיחות; 3033, 2146/7.
- בהמשך, יש לדווח לאחראי משק ותחזוקה; 4776

לסיום, יש לדווח ל:

מהנדס המעבדה (טל. 3220)

- בעת הצורך ניתן להודיע במקום למהנדס המעבדה לטכנאי המעבדה.

תדריך כללי

מטרת הניסוי

מחשוב הענן תופס תאוצה גדולה בשנים האחרונות, וקרוב לוודאי שבעתיד הלא רחוק כל מהנדס ישתמש בענן לצרכי מחקר ופיתוח בצורה זו או אחרת. הניסוי בא לעשות הכירות עם הטכנולוגיה הזו. שירות הענן שנבחר לצורך כך הוא של אמאזון – AWS, משום שהם מובילי שוק ופורצי דרך בכל הקשור בענן.

סביבת העבודה בניסוי

על מנת שתוכלו לבצע מטלות בניסוי כמשתמשים רגילים, אתם מתבקשים לפתוח חשבון באמאזון. החשבון הוא חשבון מיוחד לסטודנטים ממוסדות אקדמיים נבחרים ברחבי העולם, אשר נותן אפשרות לשימוש חינם במגוון רחב של שירותים, ובנוסף קרדיט של 100 דולר עבור החלקים שהם לא בחינם.

כחלק מתהליך ההרשמה אתם תתבקשו להכניס מספר כרטיס אשראי. מהנסיון שצברנו עד כה, הניסוי אינו כרוך כלל בתשלום, וגם הקרדיט שקיבלתם בהרשמה יישמר במלואו או כמעט במלואו.

ככל שניתן, את הניסוי אתם תבצעו על המחשבים האישיים שלכם. במהלך הניסוי יעמוד לרשותכם עוד מחשב במעבדה שיישמש אתכם כמסך שני. כך תוכלו לעבוד במקביל על קריאת ההוראות וכתיבת הדו"ח על מחשב אחד, בעוד הדפדפן עם המימשק ל- AWS ירוץ על המחשב השני.

מבנה הניסוי

לניסוי יש 2 חלקים, אשר בכל חלק אתם תקבלו משימות שונות. חלקו הראשון של הניסוי מקנה את הכלים הבסיסיים של ענן. החלק השני נותן דגש על כלים קצת יותר מתקדמים, ונשען על הניסיון שצברתם בחלק הראשון.

מכיוון שמטרת הניסוי הינה להביא אתכם למצב בו אתם יכולים באופן עצמאי להשתמש ב-AWS, אתם תידרשו לבצע חלק מהמטלות כמשימות הכנה. לכל חלק יש את משימות ההכנה שלו, וכדאי לבצע את משימות ההכנה של החלק השני רק אחרי המפגש הראשון של הניסוי.

דרישות קדם

כדי להצליח בניסוי דרוש נסיון בתחומים הבאים:

- רשתות: הכירות עם מבנה רשתות IP, subnets, TCP, ניתוב בסיסי. קורס רשתות או ניסוי TCP/IP או ידע כללי מקביל מספיקים
- לינוקס: עבודה בטרמינל של לינוקס ופקודות בסיסיות. ממ"ת מספיק.
- שפות תכנות. אנחנו נשתמש בניסוי בפייתון, אבל נסיון בכל שפה עילית (Java, C++) מספק.

חומר רקע

TCP/IP

חבילת פרוטוקולי האינטרנט. מי שלא מכיר מקורסים אחרים, נא לקרוא בחוברת ההכנה של ניסוי TCP/IP את החלק הראשון.

כתובות IP פרטיות וציבוריות

כתובת IP פרטית היא כתובת IP שלא ניתנת לניתוב באינטרנט, ומשמשת רק להתקשרות בתוך הרשת המקומית. מי שלא מכיר את הנושא, נא לקרוא כאן:

http://whatismyipaddress.com/private-ip

CIDR

CIDR היא השיטה שבה נקבע מרחב הכתובות של רשת לפי subnet mask, ולא לפי שיטת ה-classes המקורית והנושנה. היום כל האינטרנט עובד עם CIDR. מי שלא מכיר, נא לקרוא כאן:

http://whatismyipaddress.com/cidr

Application Layer

זוהי השכבה הגבוהה ביותר במודל השכבות של האינטרנט. דוגמא לפרוטוקולים בשכבה זו: HTTP, POP3, FTP.

לפי תקינה של ארגון IANA, לכל פרוטוקול יש פורט TCP שמזוהה איתו. למשל פורט 80 שייך ל-HTTP ופורט 22 שייך לפרוטוקול SSH.

פתיחת חשבון באמאזון

כדי לפתוח חשבון סטודנט באמאזון, היכנסו לכאן:

https://aws.amazon.com/education/awseducate/

בחרו את האופציה של סטודנטים ותמלאו את הטופס.

בחלק האמצעי של הטופס, תחת הכותרת **Please choose one option for accessing AWS**:

אל תבחרו Starter Account.

למי שיש חשבון ב-AWS, בשדה שבו מבקשים Account ID הכניסו את מספר החשבון הקיים שלכם. למי שאין עדיין חשבון, לחצו על הלינק לפתיחת חשבון, ופתחו חשבון רגיל ב-AWS.

תזדהו בשמכם האמיתי וכסטודנטים בטכניון, כולל אימייל טכניוני. רק כך יאשרו אתכם עם כל ההטבות. האישור עלול לקח כמה ימים, לכן היערכו מראש.

מחשוב ענן - מבוא

המונח ענן יכול להתייחס למגוון של שרותים שונים, בהקשרים שונים. הניסוי הזה מתמקד בסוג הקלאסי של הענן, שנקרא גם **Infrastructure as a Service** או בקיצור **IaaS**. מעתה והלאה כאשר מוזכר מחשוב ענן, המדובר הוא על הסוג הספציפי הזה. על סוגים אחרים של מחשוב ענן אתם מוזמנים לקרוא בזמנכם החופשי באינטרנט.

מחשוב ענן הינו תשתית מחשוב, אליה ניתן להגיע דרך האינטרנט, וממנה ניתן לרכוש משאבי מחשוב שונים, ובראשם כח עיבוד (compute), ושרותי אחסון (Storage).

המודל של הענן מאפשר ללקוחות לגשת למצבור משותף של משאבים, להשתמש במשאבים לפי דרישה, ולשחרר אותם כאשר אין צורך בהם יותר. התשלום מתבצע לפי היקף השימוש ואורך השימוש.

ללקוחות של הענן אין צורך לדעת כיצד בנויה התשתית הפיזית ממנה מורכב הענן. התשתית אותה מקבל הלקוח הינה וירטואלית, ואף על פי שבסופו של דבר המשאבים אותם הלקוח מקבל הם פיזים, הגישה למשאבים פיזים אלו עוברת דרך כמה שכבות של אבסטרקציה.

שרותי הענן של אמאזון

אמאזון הינה פורצת הדרך ומובילת השוק של מחשוב ענן. המחזור השנתי של פעילות הענן הגיע ב-2018 ל-25 מליארד דולר, שהם כ- 30% מנתח השוק.

תשתית הענן מפוזרת בחוות שרתים ענקיות הנקראות גם **Data Center** (או בקיצור **DC**). נכון ל-2019, חוות השרתים נמצאות ב-20 אזורים שונים בעולם. בכל איזור כזה יש בין 2 ל-5 חוות שרתים הנמצאות במקומות נפרדים אחד מהשני, כך שבמקרה של אסון טבע או כל דבר אחר שיכול לפגוע בחוות השרתים, רק אזור אחד ייפגע. כל איזור כזה נקרא **Availability Zone** וקיימת תשתית תקשורת מאד מהירה בין Availability Zones שונים באותו איזור. בכל Availability Zone יכול להיות DC אחד או יותר.

ניתן לראות מפה של פיזור האתרים השונים כאן: https://aws.amazon.com/about-aws/global-infrastructure/

בגלל קצב הגידול המסחרר של אמאזון, החברה, בדומה לגוגל, מייצרת את השרתים של עצמה, כך שהם מתאימים לצרכיה ומשתלמים יותר כלכלית.

לסיכום, הנה כמה מספרים שיעזרו לנו להבין את סדרי הגודל. המספרים נכונים לשנת 2019, וכאמור קצב הגידול מבטיח שתוך זמן קצר הם לא יהיו מעודכנים.

- 20 אזורים
- 61 Availability Zones
- 3-5 מליון שרתים (לפי הערכות שונות)
- 55 אלף עד 80 אלף שרתים בכל Data Center
- קיבולת הרשת היוצאת מכל DC היא בערך 100Tbps.

על מנת להפוך את העניין לקצת יותר מוחשי, נציג כמה דוגמאות למשאבים ושרותים אותם ניתן לקבל.

שרותי מחשוב – EC2

Elastic Compute Cloud או בקיצור **EC2** - שרותי מחשוב הענן של AWS. מאפשרים להריץ מכונות וירטואליות בגדלים וסוגים שונים.

המכונה הכי חלשה (Nano) הינה בעלת ליבה אחת, 1 גיגה זיכרון RAM, 8 גיגה דיסק וחיבור רשת באיכות נמוכה.

המכונה הכי חזקה (32xlarge) מורכבת מ-64 ליבות, 1952 גיגה זיכרון RAM וחיבור רשת של 25 גיגה.

בין לבין יש כ-9 דרגות של חוזק, וכמה עשרות אפשרויות לשחק עם כמות המשאבים של כל מכונה, כתלות במטלות אותן המכונה צריכה לבצע. למשל עבור שרת שמבצע חישובים במאטלב נגדיר מכונה עם יותר ליבות ויותר זיכרון, אבל לא נשקיע באחסון.

המכונות מסוגלות להריץ מערכות הפעלה שונות והתשלום עליהן הוא לפי שעות שימוש. ניתן להריץ מכונות EC2 באזורים גאוגרפיים שונים בעולם.

שרותי אחסון

לאמאזון יש כמה סוגים של שרותי אחסון. הנה 2 לדוגמא:

Simple Storage Service או בקיצור **S3** – צורת האחסון הפשוטה ביותר. אמאזון מחייבת את הלקוחות ב-15 סנט לגיגה אחסון, אבל גובה גם על שרותי רשת כאשר קוראים וכותבים לשטח האחסון. לקוחות גדולים מקבלים הנחת כמות.

אמאזון מבטיחה 99.999999999 אחוז עמידות מפני אובדן מידע, ו-99.99 אחוז זמינות של המידע.

האחסון מתבצע ע"י הגדרת buckets.

Amazon Glacier – זהו שרות אחסון נוסף של אמאזון, המיועד יותר לאחסון לטווח ארוך, והתמחור הוא בהתאם. מחיר האחסון הוא פחות מ-1 סנט לגיגה, הכתיבה היא בחינם, אולם הקריאה היא יקרה מאד. סוג זה של אחסון נועד להחליף את השימוש בטייפ מגנטי או דיסק אופטי בהם לקוחות משתמשים כיום לגיבוי.

שרותי רשת

הרשת היא אחת משלושת עמודי התווך של הענן, ביחד עם המחשוב והאחסון. אולם שרותי הרשת החשובים ביותר הינם אלו המובלעים בתשתית הענן, ומאפשרים קישוריות טובה בין הלקוחות לענן, ובין המכונות הוירטואליות השונות של הלקוח בענן עצמו.

בכל זאת, נזכיר שרות אחד של אמאזון הקשור לרשת - **Virtual Private Cloud** או בקיצור **VPC**.

שרות VPC מאפשר ללקוח לבנות רשתות מקומיות בתוך הענן. רשתות אלו בעלות מאפיינים זהים לרשת מקומית רגילה, כלומר חולקים את אותו מרחב כתובות IP, מתקשרים עם רשתות אחרות דרך נתב, וניתן להתקין Firewall, או רכיבי רשת אחרים בנקודות היציאה של הרשת לעולם החיצון.

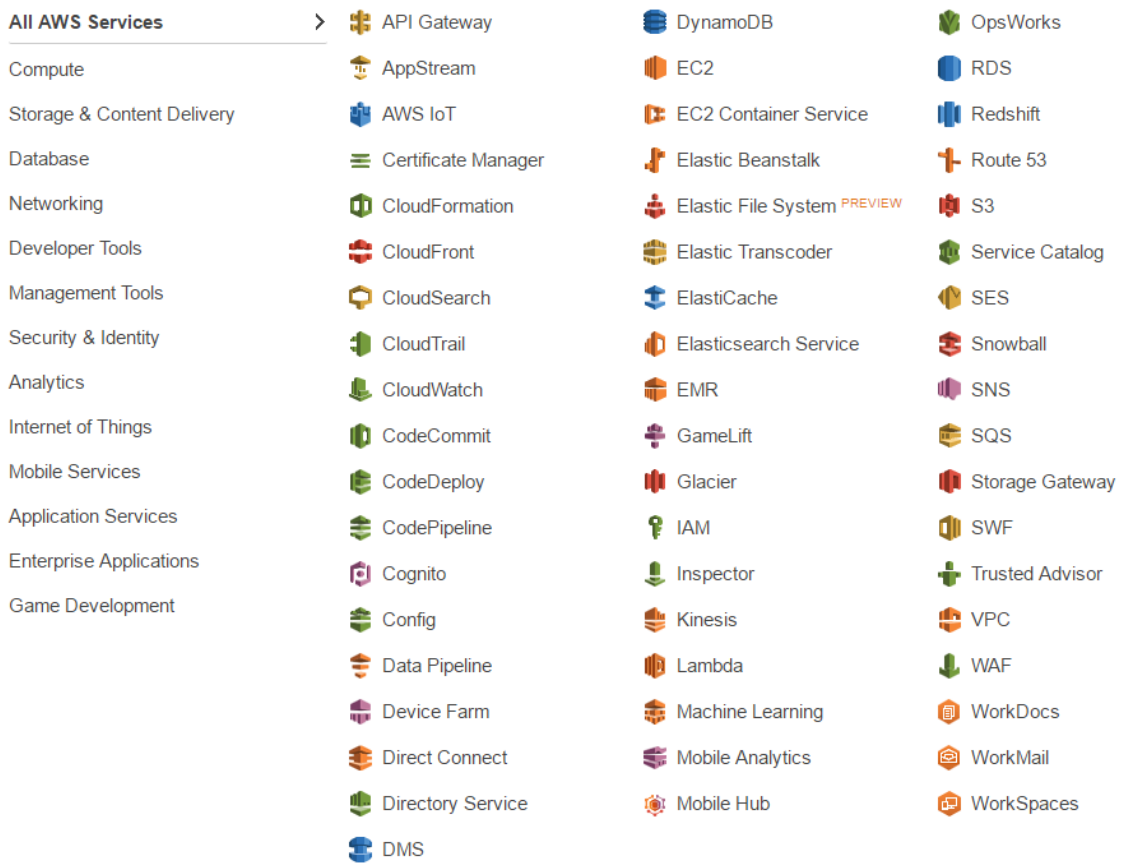
ניתן להגדיר רשת ציבורית או רשת פרטית. לרשת ציבורית ניתן להגיע מהעולם החיצון, והיא מיועדת למשל להתקנן שרתי אינטרנט. רשת פרטית היא לא נגישה מהעולם החיצון. ניתן להגיע לשרת ברשת פרטית, דרך שרת ברשת ציבורית. אנחנו נתרגל זאת במהלך המעבדה.

שרותים נוספים

אמאזון מציעה עשרות שרותים נוספים, אשר לעיתים קרובות דרושים בענן. ע"י כך אמאזון חוסכים ללקוחות שלהם לא רק את כל הטירחה שכרוכה בניהול חומרה, אלא גם את הטירחה והידע הנחוצים להתקנות של רכיבי תוכנה. שרותים כאלו לדוגמא:

- מסדי נתונים Database
- איסוף סטטיסטיקות על שרתים (כמות פניות, נפח תעבורה וכו')
- כלי ניהול קונפיגורציית תוכנה.

כדי לשטוף את העיניים, הנה המסך שמראה את כל השרותים. ללמוד את כולם זו משימה די בלתי אפשרית (וגם קצת מיותרת), אם כי ישנם קורסים שמכינים מהנדסים להסמכה כמהנדסי ענן של אמאזון, שמלמדים חלק גדול מהרשימה. בכל אופן אנחנו ניגע בחלק קטן מאד מכל זה, רק כדי להבין את הרעיון.



ניהול חשבון AWS בניסוי ובכלל

אמאזון לא עובדים לשם שמיים, ומרוויחים לא רע מהשרות שלהם. ב-2018 ההכנסות שלהם מפעילות ה-AWS היו כ-25 מיליארד דולר.

עיקר ההכנסות שלהם הם לא מאנשים פרטיים, אלא מחברות שמעבירות פעילות מחדרי שרתים פרטיים (בעלי עלות אחזקה מאד גבוהה) לענן.

על מנת להישאר דומיננטים בשוק, AWS מציעים תכניות לסטודנטים ממוסדות מוכרים (הטכניון הוא אחד מהם) שילמדו את הפלטפורמה ללא עלות, ובכך יגבר הסיכוי שישתמשו בה בבוא היום במקומות העבודה שלהם.

תכנית ההטבות לסטודנטים מתחלקת ל-2 ערוצים:

- שימוש חינם בשרותים, עם מגבלה חודשית לכמות השימוש. למשל חודש הרצה של מכונה וירטואלית, 2000 שמירות לאחסון, 20,000 קריאות מאחסון וכך הלאה. פרטים בדף aws.amazon.com/free
- מתן קרדיט חד פעמי של 100 דולר עבור כל שימוש שהוא לא חינמי. את קוד ההטבה עבור הקרדיט הזה תקבלו במייל ההרשמה.

עם כל זאת, אתם תתבקשו להכניס מספר כרטיס אשראי, למקרה שתחרגו מההטבה שניתנה לכם. זה לא יקרה במהלך הניסוי, ולמעשה תנצלו חלק קטן, אם בכלל, מההטבה של 100 דולר. את רוב הניסוי תבצעו על שרותים שניתנים חינם, ומכיוון שהניסוי לוקח כמה שעות, לא תגיעו למגבלה חודשית באף קטגוריה.

התקנות במחשב האישי

על מנת להפעיל שרתים בענן מתוך המחשב שלכם, יש צורך בכמה פעולות הכנה, וכן בהבנת האופן בו מתבצע החיבור מרחוק.

שלב זה יכלול:

1. התקנת תוכנה להתחברות מרחוק - MobaXterm
2. הגדרת מפתח שמאפשר לעשות לוגין למחשב ללא סיסמא

כניסה ראשונה ל-AWS

הסעיף הבא הינו הפעם הראשונה שבו תידרשו להיכנס לשרות של AWS. לכן יש צורך לחדד כמה דברים לפני כן.

הדף הראשי של AWS נמצא כאן: http://aws.amazon.com. כאשר אתם נכנסים אליו, אתם צריכים להכניס שם משתמש וסיסמא. בשלב הזה כבר אמור להיות לכם שם משתמש טכניוני, שמזכה אתכם בהטבות ומאפשר לכם לבצע את הניסוי ללא עלות.

בצעו sign in.

מימוש הטבה ובחירת איזור

אחרי שאתם מחוברים, תראו בר תפריטים בצבע אפור כהה. בחלק הימני שלו מופיע שם החשבון שלכם, ומימינו האיזור אליו התחברתם.

אנחנו נעבוד באיזור US East 1 שנקרא גם N.Virginia. אם זה לא האיזור שרשום לכם, החליפו ל-US-East1.

לאחר מכן, הקליקו על שם החשבון, ובחרו Billing and Cost Management.

במסך שמופיע, מצד שמאל יש עוד תפריט. בחרו Credits (מופיע יחסית למטה). הכניסו את קוד ההטבה שקיבלתם באימייל, ולחצו על Redeem.

ניווט במימשק ה-AWS

במהלך הניסוי אנחנו נשתמש בכמה שרותים שונים של AWS. לשם הנוחיות, מומלץ לפתוח כל שירות בלשונית אחרת, על מנת שלא תצטרכו לנווט הלוך ושוב כל הזמן כדי לחזור לדפים שכבר הייתם בהם.

המעבר לשירות מסויים מתבצע ע"י לחיצה על Services בבר התפריטים הראשי, יחסית משמאל. מכאן ההגעה לשירות הנכון היא די אינטואיטיבית.

ברגע שבחרתם שרות מסויים, אתם מגיעים למסך הראשי של השירות. למסך זה יש תפריט מצד שמאל שמנווט לחלקים שונים בתוך השירות. אנחנו נקרא לו "**תפריט השירות**".

SSH

Secure Shell (בראשי תיבות: SSH) הוא פרוטוקול לתקשורת מחשבים המאפשר ביצוע פעולות על מחשב מרוחק לאחר תהליך הזדהות (login). הוא נועד לאפשר תקשורת מאובטחת ומוצפנת בין שני מחשבים לא תלויים ברשתות לא מאובטחות. SSH פועל מעל TCP, והפורט הסטנדרטי שלו הוא 22.

תהליך הלוגין ב-SSH יכול להתבצע ב-2 אופנים. הראשון והיותר מוכר – בעזרת שם משתמש וסיסמא. השני, בו אנחנו נשתמש, הוא באמצעות החלפת מפתחות Key Pair. בשיטה הזו השרת בענן מחזיק מפתח אחד הנקרא מפתח ציבורי, והמחשב שלכם ממנו אתם מתחברים מחזיק מפתח שנקרא מפתח פרטי. במהלך הלוגין השרת מצפין מסרים בעזרת המפתח הציבורי, והמחשב שלכם מפענח אותם בעזרת המפתח הפרטי. כאן לא נדרשת סיסמא, אבל כן נדרש שם משתמש.

שימוש ב-SSH (מחשבי Windows בלבד)

למשתמשי ווינדוס, אנחנו נעבוד עם תוכנת MobaXterm אשר אמורה להיות מוכרת למי שעשה קורס ממ"ת.

התקינו את הגירסה החינמית מהלינק הזה:

http://mobaxterm.mobatek.net/download.html

יצירת Key Pair (כל מערכות הפעלה)

בשלב ראשון נייצר key pair בענן. תפקידו, כאמור, לאפשר לנו להתחבר לשרתים ללא צורך בסיסמא. המפתח שניצור עכשיו ישרת אותנו במהלך כל הניסוי. לא יהיה צורך אמיתי לחזור על התהליך הזה בהמשך.

1. היכנסו לדף הראשי של AWS בכתובת http://aws.amazon.com
2. תחת Services בחרו EC2
3. בתפריט השירות משמאל, תחת Network and Security, בחרו Key Pairs.
4. לחצו על הכפתור של Create Key Pair, ותנו שם למפתח שלכם.
5. המפתח נוצר, הענן שומר אצלו את המפתח הציבורי, ובמקביל מוריד בעזרת הדפדפן קובץ עם שם המפתח שלכם וסיומת pem. זהו המפתח הפרטי.

חשוב לציין שקובץ ה-pem לא נשמר כלל בענן, ולמעשה העותק היחיד של הקובץ הוא זה שכרגע ירד למחשב שלכם. לכן, אם המפתח הפרטי הולך לאיבוד, אין שום דרך לשחזר אותו, וכל שרת וירטואלי שהגישה אליו היתה דרך מפתח זה, הופך להיות לא נגיש.

כאן בא לסיומו תהליך יצירת key pair. אנחנו נשתמש בו כבר בתרגיל הראשון.

תרגיל 1: שירות EC2

מטרת התרגיל

בתרגיל זה נגדיר ונריץ לראשונה מחשב וירטואלי בענן. נלמד כיצד להעלות מחשבי EC2, ואיך ליצור security groups ו-key pairs שיאפשרו לנו לגשת את המכונות האלו מרחוק בעזרת פרוטוקול SSH. בסיום התרגיל יהיה לכם מחשב וירטואלי רץ בענן. תוכלו לעשות לוגין למחשב וכן לגשת אליו ב-ping.

מבוא

Elastic Compute Cloud או בקיצור **EC2** - שרותי מחשוב הענן של AWS. מאפשרים להריץ מכונות וירטואליות בגדלים וסוגים שונים.

המוטיבציה להשתמש בשירות כזה היא ברורה: כל מי שצריך משאבי מחשוב באופן זמני, יכול לקבל אותם רק לזמן שהוא צריך, מבלי להתעסק עם רכש, התקנות, פינוי שטח לעבודה וכו'. וברגע שהצורך נגמר, אפשר לשחרר את המשאבים האלו.

הנה כמה דוגמאות לצורך במערכת כזו:

הדוגמא הקלאסית היא אתר קניות. יש אתרי קניות בהם התעבורה גדולה פי 10 ויותר בתקופות של כמה שבועות בשנה (לפני חגים או בימי מבצע) מאשר בשאר ימות השנה. אם אתר כזה מחזיק את כל החומרה שלו בעצמו, 90% מהציוד שלו יהיה מיותר כ-11 חדשים בשנה. מה שיגרור הוצאות מיותרות. במקום זה האתר יכול להשתמש בענן, לגדול אוטומטית כאשר יש לחץ, ולקטון אוטומטית כאשר הלחץ יורד.

דוגמא אחרת, יותר קרובה אלינו, נניח שאני רוצה להריץ סימולצית מאטלב מאד גדולה למשך 5 ימים עם 10 מחשבים חזקים. כל מחשב כזה עולה בענן בערך חצי דולר לשעה. כלומר העלות הכוללת שלי היא באיזור ה-300 דולר. גם אם זה נשמע הרבה, עדיין זה יותר זול מלהשקיע 20 אלף דולר בשרתים, להוסיף תשתית מיזוג, אל-פסק, ארונות, למצוא להם מקום ולתחזק אותם.

דוגמא אחרונה, נניח שאני כותב אפליקציה Web, ומחשב אחד יספיק לי מבחינת קיבולת. שני אתגרים שקיימים בשלב ה-deployment הינם:

- היכן למקם את השרת כך שיהיה זמין לעולם, אבל לא יסכן מבחינת אבטחת רשת את המחשבים שנמצאים לידו
- איך להבטיח זמינות של השרת, כלומר שהוא תמיד יהיה דלוק ומחובר לרשת

גם כאן, הענן יפתור לי את שתי הבעיות. הגדרות נכונות של ה-EC2 (כפי שנראה מיד) נותנות לי מענה לסעיף הראשון, והבטחת הזמינות של AWS עומדת על 99.99% מהזמן, מה שפתור לחלוטין את הבעיה השניה מבלי שאצטרך בכלל לעשות משהו בנידון.

עבודה באיזור (Region) מסויים

אפשר להגדיר מכונה וירטואלית בכל אחד מהאיזורים. אולם ברגע שהגדרנו מחשב באיזור מסויים, הוא זמין רק שם, ולא ניתן לראות את המחשב כאשר עובדים באיזור אחר. אנחנו נעבוד לאורך כל הניסוי באזור us-east-1, שלפעמים נקרא גם US Standard ולפעמים נקרא N. Virginia.

Application Machine Images (AMIs)

כאשר מעלים מחשב וירטואלי, אין צורך להתקין מערכת הפעלה. אמאזון מספקת מספר גדול של images מוכנים, שכוללים מערכת הפעלה, ובמקרים מסויימים גם תוכנות נוספות בהתאם למטרות של המחשב הוירטואלי. למשל, מי שרוצה להעלות מחשב בשביל Database, יכול להשתמש באימאג' עם MSSQL Server על ווינדוס או MySQL על לינוקס וכך הלאה, ובכך לחסוך זמן רב על התקנות.

עקרונית סוג ה-image לא משפיע על גודל המכונה, אבל אם בוחרים אחד עם תוכנות כבדות, ברור מאליו שמכונה בגודל מינימאלי לא תצליח לסחוב אותו.

ה- Image בו נשתמש בתרגילים הבאים נקרא **Amazon Linux AMI**. זהו מחשב על מערכת הפעלה לינוקס (מבוסס RedHat למי שמכיר). שם המשתמש הוא ec2-user. סיסמא, כאמור לא צריכים.

Security Groups

לפני שנגדיר את המחשב הראשון חשוב לזכור: האינטרנט מלא אנשים רעים. חלקם אוהבים לתקוף אנשים אחרים, וחלקם אוהבים לתקוף מחשבים של אחרים. זו אחריות שלנו להגן על עצמינו. מחשבי אמאזון מהווים

מטרה די קורצת, מכיוון שמדובר על שרתים בטווח כתובות IP מצומצם יחסית, אשר חלק גדול מהם לא מתחבאים מאחורי Firewall.

כדי להגן על עצמינו אנחנו צריכים לקבוע כללים ברורים למי מותר לתקשר עם המחשב שלנו ולמי אסור. הכללים האלו מוגדרים לפי פרוטוקול (שממופה בדרך כלל ל-TCP Port ספציפי), והמקור של התעבורה.

לדוגמא, מחשב שיהיה נגיש בפרוטוקול SSH רק מרשת הטכניון ויהיה נגיש בפרוטוקול FTP לציבור הרחב יגדיר הגדרות security כאלה:

פרוטוקול	טווח פורטים	מקור התעבורה
SSH	TCP 22	רשת 132.68.0.0/16
FTP	TCP 20-21	כל מקום (רשת 0.0.0.0/0)

כל ישות בענן בעלת כתובת אינטרנט, למשל מחשב כמו שנראה מיד, או ישויות אחרות שנראה בניסוי עצמו, מגדירה הרשאות חיבור אליה בעזרת security group. אנחנו נלמד תוך כדי הניסוי לנהל בצורה נכונה security groups שונים.

הפעלת מכונת לינוקס

1. היכנסו לדף הבית
2. תחת Services בחרו EC2
3. לחצו על Launch Instance
4. בשלב זה הגעתם לאשף הגדרת שרת. האשף הזה מורכב מ-7 שלבים. בחלק העליון של הדף ניתן לראות את כל השלבים, ומתחת לזה באיזה שלב אתם נמצאים כרגע. המבנה הזה חוזר על עצמו גם בהגדרות אחרות. על מנת להקל על ההתמצאות של ההוראות כאן למטה, בכל הוראה נתייחס לשלב בו אנו נמצאים כרגע.
5. שלב 1: בחרו את ה-AMI בשם - **Amazon Linux AMI**

עתה נבחר גודל של מכונה, כפי שהוסבר מקודם.

6. שלב 2: בחרו t2.micro (הערך הדיפולטיבי והחינמי לסטודנטים).
7. לחצו על Next: Configure Instance Details

בהמשך נתעכב יותר על השלב הזה. כרגע נסתפק בכך שנתן לשרת כתובת public, על מנת שנוכל לתקשר איתו ישירות מהעולם החיצון.

8. שלב 3: שנו את השדה Auto Assign Public IP ל-Enable

9. שלב 3: לחצו על Next: Add Storage
כרגע לא נעשה כלום בשלב הזה.

10. שלב 4: לחצו על Next: Add Tags
11. שלב 5: לחצו על Add Tag. עבור Key כתבו את המילה Name , הכניסו את השם שאתם רוצים לתת למכונה ב-Value.
12. שלב 5: לחצו על Next: Configure Security Group

כדי שתהיה גישה גם ל-ping, צריך לאפשר גישה ל-ICMP

13. שלב 6:לחצו על Add Rule

14. הוסיפו כניסה של All ICMP – Ipv4, כאשר ה-source הוא Anywhere (עקרונית רעיון גרוע אבל בשביל התרגיל הראשון מותר לנו).
15. שלב 6: לחצו על Review and Launch ובדף הבא (שלב 7) על Launch
בשלב הזה עולה חלון בו אתם צריכים לבחור Key Pair. אם יצרתם key pair אחד, הוא כבר יופיע בחלון ב-drop down list התחתון.
16. סמנו וי בהודעה בתחתית החלון ולחצו על Launch Instance. מופיע חלון הודעה גדול מאד, ובפינה הימנית תחתונה כפתור View Instances. לחצו עליו כדי לעבור אל הרשימה של המחשבים שהרצתם. כרגע אמור להיות רק מחשב אחד ברשימה. אם הלכתם לאיבוד, אתם יכולים להגיע לאותה רשימה ע"י בחירת Services->EC2, ולחיצה על Instances->Instances בתפריט השירות.
17. מבנה הדף של רשימת ה-Instances דומה קצת לאאוטלוק. כאשר בוחרים instance, ניתן לראות פרטים עליו בחלק התחתון של המסך. אתם מוזמנים להסתכל ולהתרשם.
18. חפשו את הערך public IP של השרת שהגדרתם. הוא מופיע גם ברשימה למעלה וגם תחת הלשונית Description. לכתובת הזו ננסה להתחבר.
19. בשלב ראשון בצעו ping מה-PC שלכם (עשו זאת מתוך חלון command) לכתובת הזו. מכיוון שהרשינו ICMP, אתם צריכים לקבל תשובה.
20. בשלב הזה נתחבר ב-ssh למכונה. ההוראות כאן הן שונות בין משתמשי וונדוס לאחרים. עיקבו אחרי ההוראות שמתאימות לכם. זיכרו ששם המשתמש של המכונה הוא ממש "**ec2-user**". השתמשו בשם משתמש זה ואל תחליפו אותו בהוראות עם שם המשתמש שלכם.

משתמשי Windows בלבד

21. הריצו MobaXterm. לחצו על Session (הכפתור השמאלי בסרגל הכלים), ובחלון שנפתח לחצו על SSH. תחת Remote host כיתבו את הכתובת הציבורית של השרת שלכם. הדליקו את Specify username ורשמו בשדה שלו "ec2-user".
22. לחצו על הלשונית של Advanced SSH settings. סמנו Use Private key, ובחרו את קובץ ה-pem שהורדתם בשלב יצירת ה-key pair.
23. לחצו על OK. אם הכל הלך כשורה, מזל טוב, אתם בתוך השרת החדש שלכם.

משתמשי לינוקס ומק

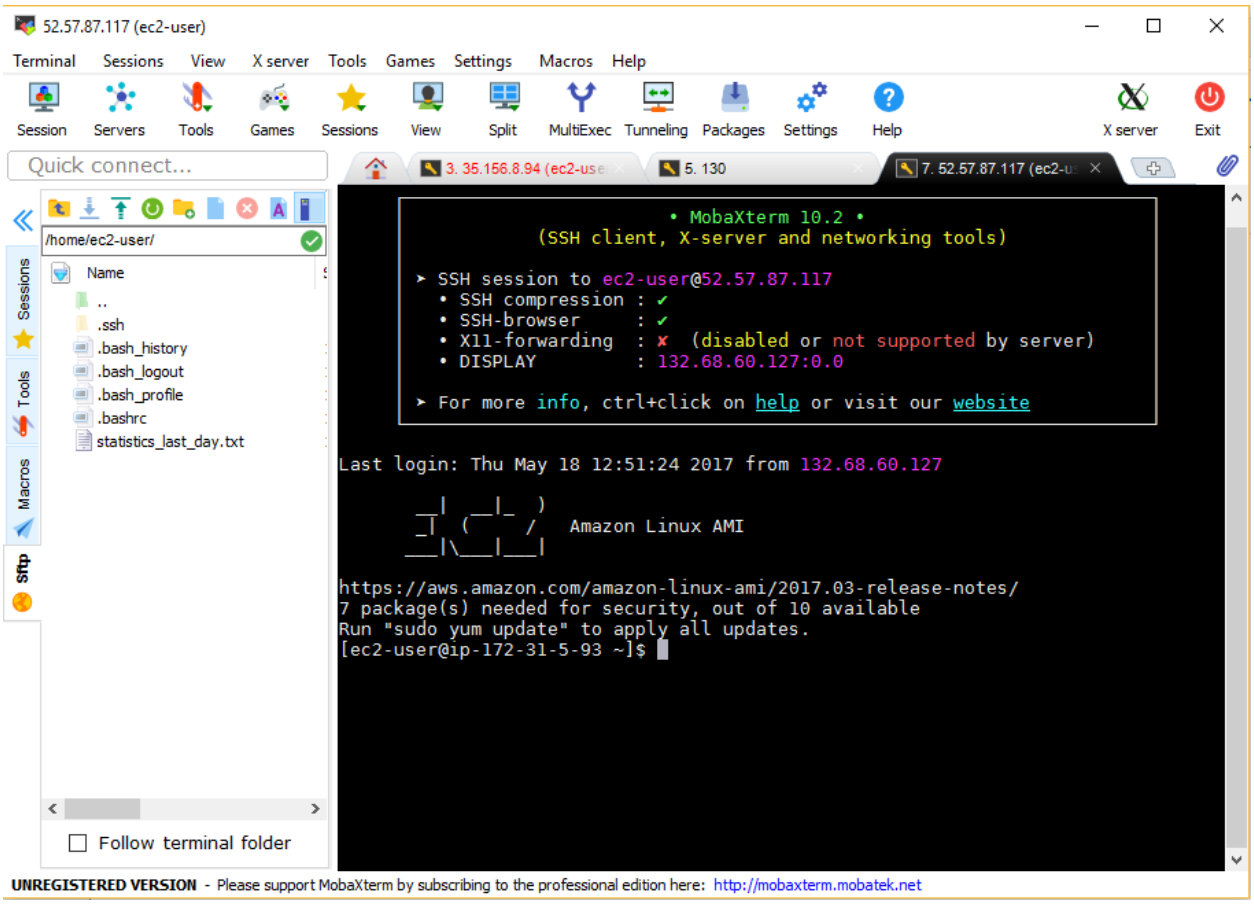
24. פיתחו טרמינל, ועיברו לתיקיה בה נמצא קובץ המפתח שלכם
25. שנו הרשאות לקובץ, לקריאה וכתיבה ע"י בעל הקובץ בלבד, בעזרת הפקודה:

```
chmod 600 <key file name>
```

26. רישמו את הפקודה:

```
ssh -i <key file name> ec2-user@<public IP>
```

חלון הטרמינל אמור להיראות כך:



עבודה עם קבצים

העלאה והורדה של קבצים

בהרבה מקרים כחלק מהעבודה שלכם תרצו להעלות קבצים לשרת שלכם (לדוגמא קוד) וגם להוריד קבצים מהשרת למחשב האישי שלכם (למשל תוצאות סימולציה או לוגים).

משתמשי לינוקס ומק

אנו נשתמש בפרוטוקול scp להעברת קבצים. פקודת scp מגיעה עם מערכת ההפעלה, והתחביר שלה משלב ssh ו-cp. גם כאן צריך את ה- key בתור פרמטר.

לדוגמא: העתקת קובץ מהמחשב שלי לתיקית MyFiles מתחת ל-home:

```
scp -i aws_key.pem TheFile ec2-user@52.53.54.55:~/MyFiles
```

גם הכיוון השני הולך, כלומר להעתיק מהמחשב הרחוק אלי:

```
scp -i aws_key.pem ec2-user@52.53.54.55:~/MyFiles/TheFile .
```

משתמשי Windows

החלק השמאלי בחלון של MobaXterm מראה את מערכת הקבצים בשרת אליו אתם מחוברים. ניתן לגרור קבצים מהמחשב שלכם לאזור זה, וכן מהאזור הזה למערכת הקבצים במידה ואתם רוצים להעתיק קובץ מהשרת המרוחק למחשב שלכם.

כחלק מהניסוי, אתם אמורים לדעת לערוך קבצים בלינוקס. האתגר במקרה זה הוא שאין לנו סביבה חלונאית עם עכבר וגלגלת, ולכן צריך להשתמש באמצעים פחות ידידותיים.

שתי תוכנות עריכה מקובלות הן nano ו-vim. מי שמכיר vim, זה העורך המומלץ. לאלו שזו ההתנסות הראשונה שלהם במוד כזה של עבודה, nano יותר אינטואיטיבי.

27. העתיקו קובץ טקסט כלשהו למחשב המרוחק.
28. השתמשו ב-nano כדי לערוך את קובץ הטקסט. הוסיפו שורות, מחקו כמה, ולבסוף שימרו את הקובץ.

מנגנוני בקרה וניהול

Cost Explorer

זהו שרות שמאפשר לכם מעקב אחר ההוצאות הכספיות שלכם. כרגע לא השתמשתם בחשבון מספיק כדי שזה יהיה מעניין להסתכל כאן, ונעשה זאת במהלך הניסוי. אולם בינתיים, צריך לאפשר את השירות זה כי לוקח עד 24 שעות מרגע שמקנפגים את האיפשור ועד שהוא מתחיל לעבוד.

1. לחצו על שם המשתמש שלכם למעלה מימין ובחרו Billing and Cost Management
2. לחצו על Cost Explorer. אם זו פעם ראשונה שאתם נכנסים לכאן, תקבלו מסך עם כפתור כחול של Enable Cost Explorer. לחצו עליו. אתם אמורים לקבל הודעה שהשירות יתחיל לעבוד בתוך 24 שעות.

שירות הודעות SNS

מבוא

שירות ההודעות של AWS נקרא Simple Notification Service או בקיצור SNS. השירות עובד במודל הנפוץ של העברת הודעות שנקרא publish-subscribe. במודל הזה קיימים publishers אשר שולחים הודעות, ו-subscribers אשר מקבלים הודעות. מה שמקשר ביניהם זה נושא ההודעה, שנקרא Topic.

נניח שיש topic בשם X. מי שמעניין אותו לקבל הודעות X נרשם כ-subscriber. מי ששולח הודעות X, לא באמת מודע לזהות מקבלי ההודעה, אבל הוא יודע שמי שההודעה מעניינת אותו, יקבל אותה.

על מנת לממש כזה מודל, צריך מערכת מתווכת (middleware) אשר תדע לרשום subscribers ולנתב הודעות לפי topic. מערכת מתווכת כזו נקראת Message Broker. ב-AWS, כמו שאנחנו כבר רגילים, עשו את העבודה בשבילנו, וה-AWS הוא ה-Message Broker. לכן כל שנותר לנו הוא להגדיר topics ו-subscribers (ועוד כמה פרטים נלווים).

התרגיל הטריוויאלי

1. ב-console בחרו בשירות SNS
2. לחצו על create topic
3. תנו לו שם HighCPU ו-Display name דומה.
4. לחצו על Create Subscription
5. בחרו בפרוטוקול e-mail, ורשמו את האימייל שלכם.
6. קיבלתם אימייל? אם לא, חפשו בתיקיית ספאם, אולי זה שם.
7. פיתחו את הודעת האימייל, אשרו את ההרשמה על מנת להמשיך לקבל התראות
8. לחצו על publish to topic, ושלחו אימייל. וודאו שקיבלתם גם את האימייל הנוסף.

שירות CloudWatch

שירות הניטור הבסיסי של AWS נקרא CloudWatch. בעזרתו ניתן לראות גרפי ביצועים, כולל CPU Utilization, תעבורת רשת ועוד. בנוסף, ניתן גם להגדיר התראות כאשר משאב עובר סף מסויים, וכן לכתוב קבצי לוג עבור קוד שאתם מכניסים לענן.

עוד על CloudWatch ניתן לקרוא כאן:

http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

לאורך הניסוי נשתמש לא מעט ב-CloudWatch. כרגע, לצורך ההדגמה, ננטר את ה-CPU Utilization של השרת שהגדרתם.

יצירת גרף

ניתן ליצור מערכת של גרפים שיראו ביצועים על הדברים החשובים. את הגרפים אפשר לראות במסכים שונים שנקראים Dashboard. בכל Dashboard אפשר להגדיר כמה גרפים, ובכל גרף אפשר להגדיר כמה מטריקות. לנו אין כל כך עם מה לעבוד כרגע, לכן נגדיר Dashboard אחד עם גרף אחד ומטריקה אחת.

1. היכנסו לשירות CloudWatch, ובתפריט השירות בחרו Dashboards.
2. לחצו על Create Dashboard, ותנו לו שם, למשל Preliminary1
3. תחת EC2 Metrics בחרו Per-Instance Metrics.
4. בחרו את המטריקה CPUUtilization, הורידו את הדגימה לדקה (בגרף שמופיע למטה) ולחצו Create Widget.

יצירת התראה

מי שמנהל כמה מאות שרתים, לא יכול להסתמך על התבוננות בגרפים, ולכן על כל פעולה חריגה כדאי להגדיר Alarm. ה-Alarm מגדיר חציה של סף מסויים עבור מטריקה שמעניינת אותנו. את ה- Alarm מחברים ל-SNS כדי שההתראה באמת תגיע לאנשהו.

המטריקות עליהם ניתן לייצר התראות הן אותן מטריקות עליהן ניתן לייצר גרפים ל-dashboards. מכיוון שיש הרבה כאלה, AWS מחלק אותם לפי קטגוריות. מציאת המטריקה הנכונה נראית התחילה כמו מציאת מחט בערימת שחת, אבל כשמבינים את הקטגוריות השונות זה נראה פחות נורא.

במוד החינמי הדגימה של מטריקה היא כל 5 דקות. גם אם מכוונים בדיקת התראה פעם בדקה, זה לא אומר שהוא דוגם את המטריקה כל דקה.

1. תחת CloudWatch בחרו Alarms בתפריט השירות
2. לחצו על Create Alarm
3. <u>שלב 1:</u> הגיעו שוב ל-CPUUtilization, כמו ביצירת הגרף.
4. <u>שלב 2:</u> הגדירו שם ותיאור להתראה, ובתנאי הגדירו שההתראה תפעל כאשר ה-CPUUtilization גדול מ-50%.
5. <u>שלב 2:</u> ב-Actions, הוסיפו את ה-Topic שהגדרתם, ולחצו על Create Alarm.

בדיקה

כדי לבדוק שהכל עובד, נעמיס על ה-CPU של המכונה הוירטואלית ונראה מה קורה.

1. התחברו לשרת ב-SSH.
2. הריצו את הפקודה הבאה כדי להתקין תוכנה פשוטה שמעמיסה על ה-CPU

```
sudo yum install stress
```

3. הריצו את הפקודה הבאה כדי להעמיס על השרת במשך 10 דקות

```
stress -c 10 --timeout 10m
```

האם קיבלתם אימייל? שמרו את האימייל וצלמו את הגרף מה-Dashboard, אחרי שה-CPU חוזר להיות נמוך.

סיום שלב ההכנה

ככלל, לא מומלץ להשאיר שרותים עובדים בענן אשר לא נמצאים בשימוש. לכן, בסיום ההכנה כדאי להפסיק את הריצה של כל מחשב וירטואלי שהקמתם.

הפסקת הריצה היא פעולה שונה מאשר מחיקת המחשב, ושקולה יותר לכיבוי שלו. בזמן שהוא מופסק הוא לא תופס ליבה וגם לא כתובת IP ציבורית. אם נרים את המחשב מחדש, נקבל כתובת ציבורית אחרת.

באופן זה אנחנו חוסכים עלויות (מכונה כבויה עולה הרבה פחות ממכונה דולקת) וגם מצמצמים חשיפה שלנו להתקפות.

1. היכנסו לשירות EC2, ולחצו בתפריט משמאל על Instances
2. עבור כל מכונה שעובדת לחצו לחצן ימני, Instance State->Stop

להגשה

אין צורך להגיש דבר במייל או בלאב אדמין. עליכם להכין את מה שרשום ברשימה, והמדריך יבדוק אתכם בתחילת הניסוי.

1. חשבון AWS פעיל (בלעדיו לא ניתן יהיה להתחיל את הניסוי)
2. Instance כבוי של EC2
3. SNS עם topic אחד לפחות.
4. אימייל של Alarm
5. גרף של ה-Dashboard.