מעבדה במחשוב ענן – חלק א'

במסגרת ההכנה כבר התוודעתם אל החלקים הבסיסים ביותר בענן.

המפגש הראשון של הניסוי מהווה המשך ישיר להכנה, וגם בו נתמקד במעבר על מספר שירותים בסיסים שמוצעים לנו ע"י AWS, ובכללם פיזור מחשבים על פני כמה רשתות, Load Balancing, ניטור נתונים ועוד.

חלקו האחרון של מפגש זה יהווה מיני סיכום של מה שהועבר עד כה, ובו תריצו שרתי אינטרנט שמנצלים את כל השירותים שהיכרנו.

הנחיות לביצוע הניסוי

הניסוי הוא באופיו יותר מעשי ופחות תיאורטי. לפיכך יהיו לכם הרבה מטלות ביצוע, שדרכם תוכלו להבין את היתרונות והאפשרויות של העבודה בסביבת ענן.

לכן, הדוח ישקף בעיקר את העבודה שעשיתם, ולא יכיל הרבה שאלות תיאורטיות.

אתם מתבקשים:

- לקרוא למדריך בסיום כל תרגיל כדי להראות לו את מה שעשיתם,
- לספק צילומי מסך של העבודה שלכם. בסוף כל תרגיל יוזכר אילו צילומי מסך צריך לצרף לדו"ח.

חיבור רשת

יש רשתות WiFi שחומת האש שלהן חוסמת חלק מהתעבורה לענן. עדיף מאד לבצע את הניסוי עם חיבור חוטי למחשב, ואם אין כזה זמין, אפשר להשתמש ברשת eewifi.

תרגיל 1: תרגיל VIRTUAL PRIVATE CLOUD

בתרגיל זה נבנה Virtual Private Cloud (או בקיצור VPC).

מבוא

ניתן לקרוא עוד על VPC כאן:

https://aws.amazon.com/vpc/

VPC מאפשר לנו לתחום חלק מהתשתית שייצרנו בענן בתוך תת-רשת וירטואלית פרטית. תת הרשת הזו מתנהגת בדיוק כמו תת רשת פיזית, כלומר אוסף מחשבים עם מרחב כתובות IP, עם נתב ביציאה מהרשת שיכול לשמש גם שרת NAT, חומת אש וכו'.

ישנם שני סוגים עיקריים של תתי רשת: public ו-private.

לתת רשת public יש קישוריות לרשת האינטרנט. בדרך כלל נמקם כאן שרתי front-end, כלומר כאלו שאנחנו רוצים שיגיעו אליהם מהעולם החיצון. כמובן שגם כאן נחיל כללים על הרשת באמצעות security groups, כך שאם השרת הוא למשל שרת HTTP, לא נאפשר גישה לשום פרוטוקול אחר.

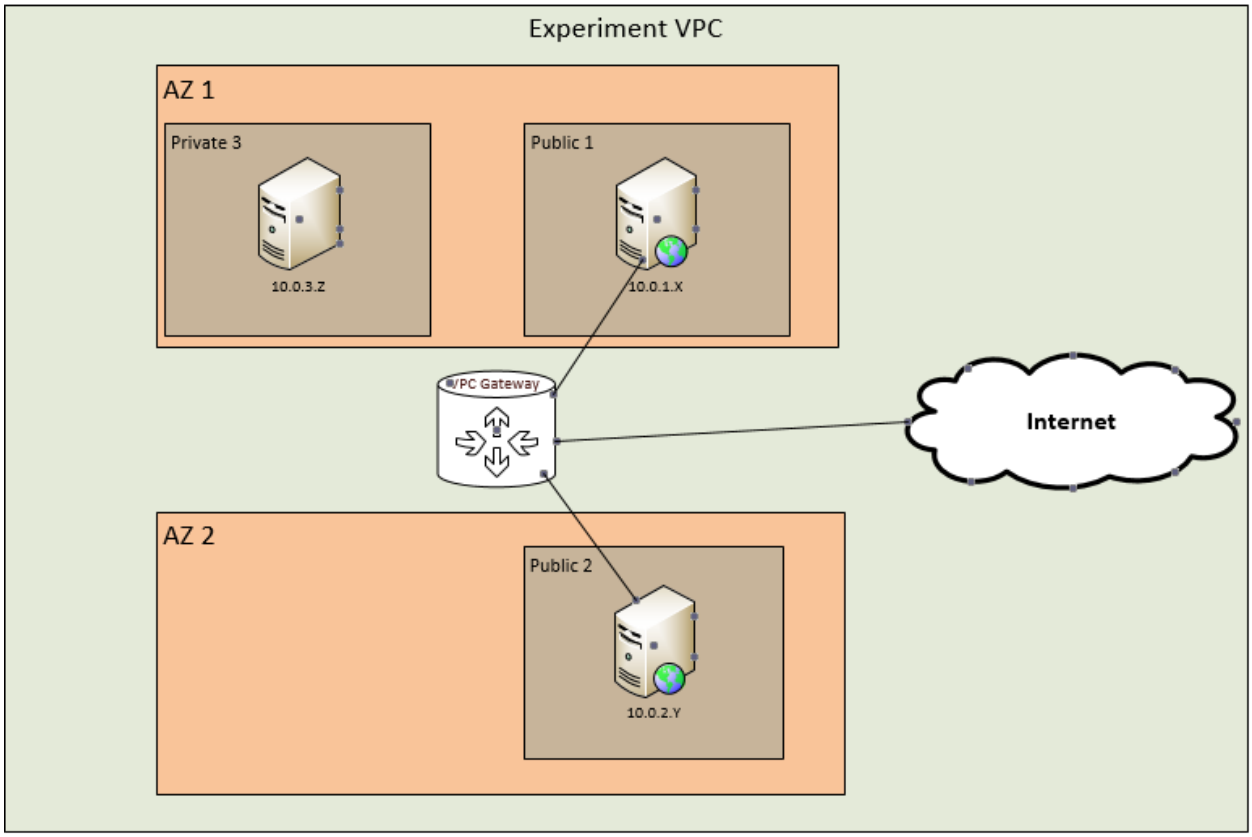
תת רשת private מקושרת לתתי רשת אחרים ב-VPC, אבל לה אין גישה ישירה לאינטרנט. כאן ממקמים בדרך כלל שרתי backend, שהם שרתים שעוסקים בעיבוד המידע ולא אמורים להיות פתוחים לציבור הרחב.

הרשת שלנו מתפרסת על 2 אזורים (AZ), ובהמשך נעלה שירות על 2 האיזורים הללו, כדי להבטיח זמינות.

מה נעשה בתרגיל

אנו ניצור VPC, ומתחתיו 3 תתי רשתות, 2 ציבוריות ואחת פרטית. 2 הרשתות הציבוריות תהיינה ב-Availability Zones שונים. נגדיר נתב אשר מחבר בין תתי הרשתות הציבוריים לבין האינטרנט. ניצור מחשב אחד בכל אחד מה-subnets, ונראה כיצד ניתן לתקשר ביניהם.

הנה ציור של הרשתות שיתקבלו בסוף התרגיל:



ה-VPC שנייצר עכשיו ישמש אתכם במהלך שני המפגשים של הניסוי.

יצירת ה-VPC

1. לחצו על תפריט Services ובחרו VPC

למרות הפיתוי, ועל מנת להבין יותר טוב את מרכיבי הרשת, לא נשתמש הפעם באשף, אלא נגדיר הכל ידנית.

2. בתפריט מצד שמאל, בחרו Your VPCs
3. לחצו על הכפתור הכחול Create VPC
4. תנו לו שם, וב-CIDR Block רישמו 10.0.0.0/16. לסיום לחצו על Yes, Create.

כרגע הגדרנו VPC של מרחב כתובות מאד גדול, של כל כתובת שהיא 10.0.X.Y. עתה נגדיר כמה תתי רשת תחת ה-VPC הזה. את תתי הרשת נגדיר בכמה AZ שונים. זו ההמלצה של אמאזון, לפזר שרתים על כמה AZ כדי לוודא זמינות למקרה שמשהו קורה ל-AZ ספציפי. שימו לב שעדיין נעבוד באותו region.

5. בתפריט מצד שמאל, בחרו Subnets, ולחצו על Create Subnet.
6. תנו לו את השם "public 1" , שייכו אותו ל-VPC החדש, בחרו AZ כלשהו (תזכרו את הבחירה שלכם) ותנו CIDR 10.0.1.0/24. לבסוף לחצו על Yes, Create.

ה-CIDR הזה מוכל ב-CIDR של ה-VPC. זה חשוב על מנת שתהיה תקשורת בין תתי רשת שונים באותו VPC.

7. חזרו על הפעולה, הפעם עם תת רשת 10.0.2.0/24 והשם public 2. כאשר אתם בוחרים AZ, בחרו אחד אחר ממה שבחרתם עבור תת הרשת public 1.
8. חזרו על הפעולה פעם שלישית, הפעם עם תת רשת 10.0.3.0/24 והשם private 3. ה-AZ לא חשוב.

אחרי שסיימנו להגדיר תתי רשתות, נגדיר Internet Gateway, אשר יאפשר גישה לאינטרנט לשרתים בעלי כתובת public עבור ה-VPC שלנו.

9. בחרו בתפריט משמאל Internet Gateways, ולחצו על הכפתור הכחול Create Internet Gateway.
10. תנו לו שם, ולחצו Yes, Create.
11. בחרו אותו, לחצו על הכפתור הימני ובתפריט לחצו Attach to VPC. שייכו אותו ל-VPC שהגדרנו בתחילת התרגיל.

לכל תת רשת ניתן להגדיר טבלת ניתוב, שהיא מקבילה לטבלת ניתוב של נתב היציאה של תת רשת בעולם האמיתי. אנחנו לא חייבים להגדיר ספציפית לכל תת רשת טבלת ניתוב, ובמקרה שאנחנו בוחרים לא לעשות זאת, תת הרשת משתמשת בטבלת הניתוב של ה-VPC, אשר מוגדרת אוטומטית.

12. בתפריט מצד שמאל, בחרו Route Table. כרגע יש עבור ה-VPC שלכם רק את טבלת הניתוב הדיפולטית, עם ניתוב מקומי לרשת 10.0.0.0/16. הטבלה הזו טובה לרשת הפרטית אבל לא לרשתות הציבוריות. נוסיף עוד טבלת ניתוב לרשתות הציבוריות.
13. לחצו על Create Route Table ובחרו את ה-VPC שלכם. קיראו לטבלה Public Route.
14. לחצו על לשונית Routes. כרגע יש רק ניתוב מקומי. לחצו על Edit, והוסיפו ניתוב. ה-Destination הוא 0.0.0.0/0 (שזה מסמל Default Gateway), כ-Target הכניסו את ה-Internet Gateway שזה עתה הגדרנו. לסיום לחצו על Save.
15. כעת נעביר את תתי הרשתות הציבוריות לטבלת הניתוב החדשה. חיזרו ל-subnet בחרו את תת הרשת, ותחת הלשונית Route Table בצעו את השינוי ע"י לחיצה על Edit ובחירת הטבלה ב-Change to.

בשלב הזה יש רשת מוכנה לפי מה שרצינו להגדיר. בואו ונגדיר שרתים ונראה שיש ביניהם קישוריות. בשלב ראשון נגדיר 2 שרתים ציבוריים עם כתובת ציבורית, כפי שעשינו בתרגיל 1. אחר כך נגדיר שרת ברשת הפרטית.

16. הגדירו 2 שרתי EC2 ציבוריים. הנה כמה דגשים להגדרת השרת:
17. שלב 3: בשדה Network, בחרו את ה- VPC שלנו
18. שלב 3: בשדה Subnet בחרו פעם אחת את Public 1 ובפעם השניה את Public 2.
19. שלב 3: לא לשכוח לאפשר Public IP.
20. שלב 5: קראו לשרתים Srv 1 ו-Srv 2.
21. שלב 6: בנוסף ל-Rule של SSH שכבר מוגדר לכם, הוסיפו עוד שורה לאותו security group של השרת שמאפשר ICMP מכל מחשב (שוב, לא רצוי אבל נסלח באופן זמני ויושמד בסוף הניסוי).
22. הגדירו שרת גם עבור Subnet 3, עם ההבדל היחיד שבשלב 3 לא צריך לתת לו Public IP, כי אנחנו לא מעוניינים לגשת אליו מבחוץ. גם אליו צריך לאפשר גישה ב-ICMP.

כרגע אמורים להיות לכם 3 שרתים רצים. ל-2 מהם יש 2 כתובות. כתובת אחת ברשת 10.0.0.0 שהיא כתובת פרטית, כתלות ב-VPC שלכם, וכתובת נוספת ציבורית, בד"כ מתחילה במספר 54 או 52. הכתובת הציבורית לא באמת מקונפגת על השרת, אלא AWS יודע למפות תעבורה אל ומהכתובת הציבורית לשרת שהגדרתם.

23. הריצו ping:
    - מהמחשב שלכם ל-2 השרתים בכתובת הציבורית
    - מאחד מהשרתים הציבורים לשרת הציבורי השני בכתובת הציבורית
    - מאותו שרת ל-2 האחרים בכתובת הפרטית.

התחברות ב-SSH לשרת הפרטי

מכיוון שלשרת הפרטי אין כתובת אינטרנט, אתם לא יכולים להגיע אליו ישירות מהמחשב שלכם. הפיתרון הוא פשוט: התחברות ב-SSH לאחד המחשבים הציבוריים, וממנו התחברות ב-SSH למחשב הפרטי. לצורך כך, צריך להעביר את המפתח של ה-SSH למחשב הציבורי. באופן כללי, מומלץ מאד להשתמש במפתחות נפרדים, כדי לא להיות במצב בו אחד מהשרתים שלכם מחזיק את המפתח לכל הענן שלכם. אנחנו נחסוך את הטרחה לאור העובדה שהמחשבים האלו יסיימו את חייהם בתוך מספר דקות.

24. העתיקו את המפתח מהמחשב האישי שלכם את השרת. תמקמו אותו בתיקיית הבית.
    - משתמשי Windows יכולים לגרור את קובץ המפתח לשרת.
    - כל השאר, הריצו scp מתוך טרמינל.
25. התחברו אל המחשב הפרטי דרך אחד מהמחשבים הציבוריים. הפעם כולם משתמשים בשיטה הלינוקסית. חיזרו לחוברת ההכנה ועקבו אחר ההוראות להתחברות ב-ssh למשתמשי לינוקס.

הערה למשתמשי MobaXterm: ניתן לדלג בעתיד על התהליך ע"י שימוש בלשונית Network settings. במקרה שלנו היינו מגדירים ישר את השרת הפרטי עם הכתובת הפרטית כשרת היעד, ואת השרת הציבור שדרכו עוברים כ-Gateway SSH server.

להגשה

שאלה מס' 1.1 — כמה subnets מהסוג שכבר הגדרנו אפשר להגדיר ל-VPC שלכם, בהנחה שמ-Mask נשאר 24 ביט?

שאלה מס' 1.2 — אם נרצה להגדיר מספר כפול מזה של subnets, האם יש דרך? אם כן, מהי?

שאלה מס' 1.3 — נניח שאני רוצה להגדיר עוד VPC אחד. האם אפשר להגדיר אותו עם אותו מרחב כתובות כמו ה-VPC הראשון?

שאלה מס' 1.4 — האם ניתן לגשת לאינטרנט ממחשב בתת הרשת הפרטית, ללא קונפיגורציות נוספות? אם אתם לא בטוחים, נסו מתוך הטרמינל להוריד קובץ משרת ע"י הפקודה `wget www.example.com`

שאלה מס' 1.5 — אם נגדיר מחשב בתת הרשת הפרטית עם כתובת ציבורית, מה לדעתכם יקרה אם ננסה להתחבר אליו מרחוק?

צילומי מסך

שאלה מס' 1.6 רשימת ה-subnets

שאלה מס' 1.7 ה-Route table של ה-VPC

שאלה מס' 1.8 2 הפינגים מהמחשב שלכם ל-2 המכונות ב-public.

סיום התרגיל

26. מחקו את שלושת השרתים שייצרתם בתרגיל הזה. כדי למחוק שרת, קליק ימני, בחרו בתפריט:

Instance State->Terminate

תרגיל 2: תרגיל ELASTIC LOAD BALANCING

מבוא

אפשר לקרוא יותר בהרחבה על ELB כאן: http://aws.amazon.com/elasticloadbalancing/

Elastic Load Balancing מפזר באופן אוטומטי תעבורה נכנסת לעבר מספר שרתים. הוא מאפשר לנותן השרות לספק עמידות גדולה יותר מפני תקלות, ומגדיל את קיבולת השרות. ELB מוודא את תקינות השרתים, וישלח תעבורה רק לאלו אשר במצב תקין. ELB יודע להגדיל את הקיבולת של עצמו במקרה של תעבורה נכנסת גדולה מאד, וגם יודע להתממשק לשירות אחר, אשר יגדיל את מספר מחשבי ה-EC2 במידה ואלו הקיימים לא עומדים בעומס השירות.

מה נעשה בתרגיל

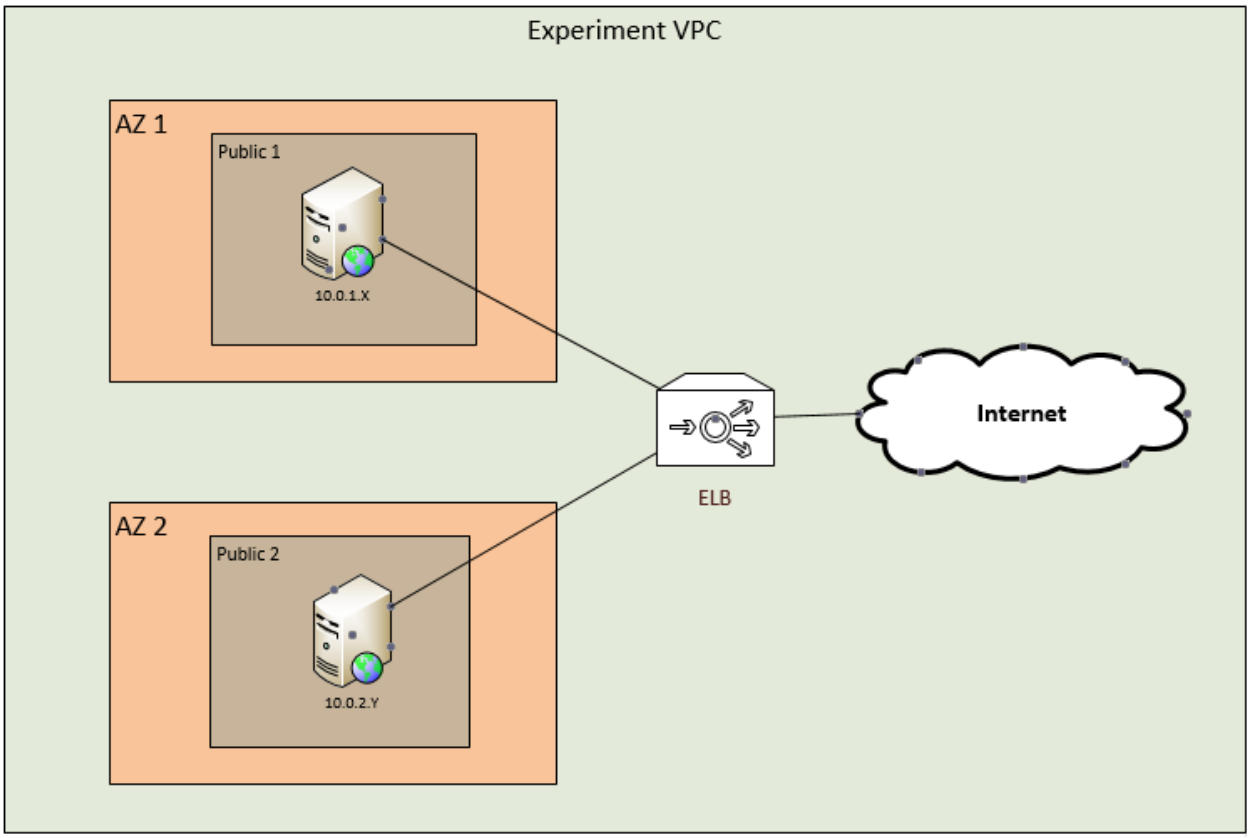
בתרגיל זה נלמד על העקרונות של Load Balancer והאופן שבו משתמשים בו בענן.

אנחנו נשתמש בשרות **Elastic Load Balancing** (או בקיצור **ELB**) אשר יחלק עומס בין כמה שרתי EC2. אנחנו נריץ אפליקציה מאד פשוטה על כל אחד מהשרתים ונראה כיצד ה-ELB מפזר את העומס ביניהם כאשר נפנה לאפליקציות בעזרת הדפדפן שלנו.

בשלב ראשון נעלה 2 מחשבי EC2, נגדיר עליהם שרת Web ונוודא שאנחנו יכולים להגיע לכל אחד מהשרתים באופן עצמאי. בשלב שני נגדיר את ה-ELB ונוסיף אליו את שני המחשבים, ואז במקום לפנות ישירות לשרתים נפנה ל-ELB ונראה כיצד כל פעם הבקשה הולכת לשרת אחר. לבסוף, נציץ בסטטיסטיקות אותן מספק ה-ELB ב-CloudWatch.

אנחנו נשתמש באותם VPC ו-subnets שהגדרנו בתרגיל הקודם.

הקונפיגורציה שלנו נראית כך :



הגדרת SECURITY GROUP

כבר יצא לכם להגדיר כמה security groups בתרגילים הקודמים. אם לא הפגנתם ערנות יתרה, כנראה שלכל אחד מהשרתים שהגדרתם עד כה יש security group משלו, בשם launch-wizard-<n>.

הפעם נרצה להיות יותר מסודרים, ולהגדיר באופן מפורש security group ולהשתמש בו לכל אורך התרגיל הזה.

1. בחרו (או הישארו) בשירות EC2
2. בתפריט השמאלי, תחת Network & Security, בחרו Security Groups
3. לחצו על הכפתור הכחול למעלה, Create Security Group
4. תנו לו שם
5. אפשרו תעבורת כניסה ל-HTTP Port 80, ICMP, SSH
6. לחצו על Create.

הרצת שרתי ה-WEB

כאמור, נתחיל בהגדרה והרצה של 2 שרתי WEB מעל EC2

הגדירו 2 שרתי EC2 כפי שעשינו בתרגיל הקודם, על בסיס אותו AMI, אבל עם כמה שינויים:

1. בשלב 3: קשרו כל אחד מהשרתים לשני ה-public subnets שלכם, שרת לכל subnet.

2. לא לשכוח לתת שם לשרת ב- Step 5
3. ב- step 6, בחרו ב-Select an existing security group
4. בחרו ב-SG החדש שלכם

התקנת שרת אינטרנט במכונות החדשות

אחרי שהשרתים מוגדרים, צריך להתקין עליהם שרת אינטרנט:

5. sudo yum install httpd
6. sudo service httpd start

בשלב הזה ניצור קובץ טקסט. אל הקובץ הזה נגיע בעזרת הדפדפן. שם הקובץ צריך להיות זהה לחלוטין בכל שרת, אבל התוכן שלו צריך להיות שונה. באופן זה, כאשר ניגש לקובץ לדך ה-ELB, נקבל בכל פעם ערכים שונים.

בכל אחד מהשרתים:

7. פיתחו חלון טרמינל, ועיברו לתיקיה /var/www/html
8. צרו קובץ חדש, וכיתבו בו משהו, כאמור שונה בין המכונות. השתמשו בהרשאות sudo, אחרת לא תצליחו לשמור את הקובץ.
9. שימרו את הקובץ בשם nssl.txt

אחרי שסיימתם והרצתם את המחשבים בואו נבדוק שאתם יכולים לגשת אליהם בעזרת הדפדפן.
עבור כל שרת:

10. מיצאו את כתובת ה-public IP שלו
11. פיתחו דף חדש בדפדפן וכיתבו http://<public ip>/nssl.txt

שימו לב לכך שכל שרת מציג תוכן שונה. כך תוכלו לזהות איזה שרת מבין השניים עונה לכם כל פעם, כאשר נחבר אותם ל-LB.

יצירת LOAD BALANCER

עכשיו כשיש לנו שני שרתים, נייצר Load Balancer שיהווה כתובת אחת לגישה ל-2 השרתים.

12. ב-AWS Console תחת EC2 Dashboard, הקליקו על Load Balancing->Load Balancer (זה ממוקם יחסית למטה)
13. לחצו על כפתור Create Load Balancer. בחרו Classic Load Balancer.
14. שלב 1: תנו שם ל-LB, ובחרו את ה-VPC שלכם.
15. שלב 1: הפרוטוקול צריך להישאר כפי שהוא. אנחנו רוצים לקבל תעבורה בפורט 80 ולהעביר אותה לשרתי EC2 גם כן בפורט 80.
16. שלב 1: בחלק ה-subnets העבירו את כל ה-public subnets ל-selected subnets ע"י לחיצה על הפלוס.
17. שלב 2: השתמשו ב-security group שהגדרתם מקודם
18. שלב 4: שנו את ה-ping path ל-/nssl.txt (מחקו את index.html). השלב הזה קובע כיצד ה-ELB יבצע בדיקת תקינות לשרת. שרת שמחזיר HTTP 200 ייחשב זמין.
19. שלב 5: הוסיפו את שני השרתים שהגדרתם מוקדם יותר
20. סיימו את התהליך

בשלב זה AWS מעלה את ה-ELB, ואחר כך, בהתאם להגדרות, יוצר קשר עם שרתי ה-WEB ומבצע בדיקות תקינות מולם. כל זה עלול לקחת 2-3 דקות. ניתן לצפות בהתקדמות ע"י לחיצה על ה-ELB.

21. לחצו על ה-ELB מתוך הרשימה, עיברו ללשונית Instances והמתינו עד שהשרתים עוברים למצב In Service.

22. כאשר 2 השרתים במצב In Service, עיברו ללשונית Description.
23. העתיקו את הכתובת ה-DNS של ה-ELB.
24. הדביקו את הכתובת הדף חדש של דפדפן, כולל שם הקובץ. רפרשו את הדפדפן כמה פעמים. אם ה-ELB עובד כהלכה, אתם אמורים לראות את 2 הערכים מתחלפים ביניהם.

להגשה

שאלה מס' 2.1 נניח שהשרת שלנו משרת גם HTTPS על פורט 443. מה נצטרך לשנות כדי לאפשר גישה כזו?

שאלה מס' 2.2 ה-ELB שלנו מעביר תעבורה מפורט 80 לפורט 80. האם ניתן להעביר גם תעבורה מפורט מסויים לפורט במספר אחר?

צילומי מסך

שאלה מס' 2.3 מסך ה-ELB, כאשר בחלקו התחתון מסומנת לשונית Instances

שאלה מס' 2.4 דפדפן מכוון לדף ה-ELB

ניטור ב-CLOUDWATCH

אנחנו ננטר את כמות השרתים שמחוברים ל-ELB. בשלב ראשון ניצור גרף שמראה את מספר החיבורים, ולאחר מכן נגדיר התראה שפועלת כאשר מספר החיבורים קטן מ-2, ננתק מחשב אחד ונראה את ההתראה משוגרת.

יצירת SNS עבור ההתראה

1. בעזרת שירות SNS, צרו topic חדש וקראו לו ELB_Alert.
2. צרו עבור הטופיק subscription מסוג SMS, עם מספר הנייד שלכם. לא לשכוח להתחיל ב-972. למשל, עבור הטלפון 052-3456789 הכניסו 972523456789

יצירת התראה

3. תחת CloudWatch היכנסו ל-Alarms
4. לחצו על Create Alarm
5. בחרו מתוך ELB את Per-ELB Metrics
6. שלב 1: בחרו את HealthyHostCount, ובאיזור הגרף שנו את קצב הדגימה לדקה אחת.
7. שלב 2: הגדירו שם ותיאור להתראה, ובתנאי הגדירו שההתראה תפעל כאשר מספר ההוסטים קטן מ-2.
8. חיזרו ל-ELB ונתקו את אחד השרתים ממנו.
9. חיזרו ל-CloudWatch. האם קיבלתם התראה?
10. חיזרו לגרף, תראו את הנפילה מ-2 ל-1 וצלמו אותו.

להגשה

שאלה מס' 2.5 כמה זמן יכול לקחת עד שמגיעה התראה?

צילומי מסך

המסך הראשי של CloudWatch, עם הגרף הקטן של ה-Alarm.

תרגיל 3: AUTOSCALE

מבוא

ניתן לקרוא עוד על Auto Scaling כאן:

https://aws.amazon.com/documentation/autoscaling/

אחד מהמנגנונים החשובים שמאפשרים לנו לשמור על זמינות השרות שלנו, היא היכולת לגדול בזמן עומס ולקטון בחזרה כאשר העומס יורד. המנגנון הזה נקרא AutoScale והוא נמצא תחת השירות של EC2.

המנגנון עובד כך: אנחנו מגדירים אירוע אחד או יותר שנותן אינדיקציה על עומס. כאשר האירוע קורה, תוגדר מכונת EC2 חדשה, מתוך Image (AMI) ידוע מראש. התהליך נקרא Scale Out.

בנוסף, אנחנו מגדירים אירוע שנותן אינדיקציה על ירידת העומס. כאשר אירוע כזה קורה, אחת מהמכונות תפסיק את עבודתה. התהליך נקרא Scale In.

על מנת שלא תהיה השתוללות של מכונות שעולות ויורדות במקרה שאנחנו סובבים סביב הסף, ההגדרות צריכות להיות חכמות. למשל הסף הנמוך הוא נמוך בצורה משמעותית מהסף הגבוה. כמו כן, יש מינימום זמן בין אירוע לאירוע, כך שלא יקרה מצב של העלאה היסטרית של מכונות (אלא אם רוצים את זה).

בנוסף להגדרה נכונה של ספים, צריך גם לתכנן נכון את קצב העלאת המכונות. קצב העלאה מתון יתן יותר יציבות למערכת, אבל קצב מתון מדי יכול לגרום לעומס יתר וירידה בשירות. הרבה מזה תלוי גם באופי השרות.

לדוגמא, נניח שיש אתר קניות, שכל ערב נפח התעבורה שלו הוא פי 8 מאשר בבוקר, אבל בתבנית קבועה ועליה הדרגתית שלוקחת כמה שעות (נניח משעה 5 אחר הצהריים עד 10 בערב). עליה מתונה, נניח שלא מאפשרת עליה של יותר ממכונה אחת כל 5 דקות תתן פיתרון הולם.

מצד שני, אתר חדשות, שעלול לספוג בעקבות אירוע מסויים עליה פתאומית של מאות אחוזים, לא יכול להרשות לעצמו התנהגות דומה, וצריך להיות מסוגל להכפיל את הקיבולת שלו בזמן קצר מאד.

החיסרון של קצב העלאה גבוה הוא שמערכת רגישה מדי עלולה להרים שרתים מיותרים, ובכך לבזבז משאבים. למשל ספייק של בקשות או התקפת DDOS יכולים לגרום לכך.

ההגדרה כוללת גם גבול עליון ותחתון של מספר המכונות, כך שאי אפשר לעלות מעל מקסימום מסויים.

שלבים בהגדרות

ההגדרה של המנגנון הזה היא קצת מורכבת, ולכן נעבור על סדר הדברים לפני שנתחיל לבצע את ההגדרות.

שמירת IMGAE

מכיוון שהמכונות עולות ויורדות אוטומטית, אנחנו צריכים לשמור לעצמנו Amazon Machine Image או בקיצור AMI כזה שיהיה מוכן לעבודה ללא מגע יד אדם. לשם כך, בדרך כלל, נעלה מכונה, נבצע עליה את ההתקנות, ההגדרות והבדיקות הרצויות, ונשמור אותה כ-AMI.

שלב זה כבר נעשה עבורכם, וה- Image שבו תשתמשו נקרא NSSL-AutoScale וניתן למצוא אותו תחת Public Images.

יצירת LAUNCH CONFIGURATION

שלב זה מאד דומה ליצירת מחשב EC2 בודד. כאן נגדיר את סוג ה-Image, גודל המכונה, VPC ושאר הדברים שאתם כבר מכירים מתרגילים קודמים בהם הגדרתם מחשבים.

יצירת AUTO SCALING GROUP

זהו למעשה השלב החשוב, בו מגדירים את התנאים בהם אנחנו רוצים להגדיל או להקטין את מספר המכונות. השלב הזה הוא קצת מבלבל וקל מאד לטעות בו.

יצירת ALARMS

כחלק משלב יצירת ה-AS Groups אנחנו נייצר Alarms, שמוגדרים בתוך שירות ה-CloudWatch (אותו שירות שבעזרתו כבר ראינו סטטיסטיקות).

מה נעשה בניסוי

בניסוי אנחנו נייצר עומס מלאכותי על הרשת בעזרת תוכנה תוכנת stress בה השתמשנו בהכנה. אנחנו ניצור מנגנון שמנטר את העומס על השרתים ומעלה את מספר המכונות כאשר העומס הוא גדול, עד למגבלה של 8מכונות. לאחר מכן נפסיק את העומס ונראה כיצד מספר המכונות קטן בחזרה.

על השרתים יהיה מותקן שרת Web פשוט, שמראה את כתובת ה-IP של השרת. בצורה זו נדע להבחין בין השרתים.

על מנת שתהיה לנו גישה לכל השרתים, נחבר אותם ל-Load Balancer בצורה כזו, שכל שרת שעולה מתחבר אוטומטית ל-LB. אנחנו נפנה לכתובת ה-LB ונקבל (בשאיפה) כל פעם כתובת של שרת אחר.

הגדרות

הגדרת ELASTIC IP

בהמשך, כאשר נריץ כמה שרתים, הם יהיו כולם מחוברים דרך Load Balancer, ולכן לא יזדקקו לכתובת חיצונית. אבל אנחנו כן צריכים גישה למחשב אחד, על מנת להתחבר אלו ב-SSH ולייצר עומס.

לשם כך נגדיר כתובת IP ציבורית, אשר בהמשך, כאשר המכונה תיווצר, נדביק לה את הכתובת הזו.

11. תחת שירות EC2 בחרו Elastic IP
12. לחצו על Allocate New Address
13. לחצו על Allocate

VPC

גם בתרגיל זה נעבוד עם תתי רשת הציבורים באותו VPC שהגדרתם בתחילת הניסוי.

שימוש ב- LOAD BALANCER

אנו נמשיך להשתמש ב-LB שהגדרנו בתרגיל הקודם. וודאו שכרגע אף שרת אינו מחובר אליו ושכל תתי הרשת הציבוריים שהגדרתם בתרגיל ה-VPC מחוברים אליו. אולם הבדיקה תיעשה מול קובץ שונה, מכיוון שבשרת החדש אין את הקובץ nssl.txt.

1. עבור ה-ELB שלכם, בלשונית Health Check שני את ה-Ping Path ל-/ip.html

יצירת LAUNCH CONFIGURATION

זה השלב בו מגדירים את מאפייני המכונה שתעבור Auto Scale. אנחנו נשתמש ב- Image מיוחד עבור הניסוי.

בשירות EC2, בחרו Auto Scaling -> Launch Configuration. לחצו על כפתור Create Auto Scaling Group, ובמסך הבא לחצו על הכפתור Create launch configuration.

2. שלב 1: תחת Community AMIs בחרו NSSL-AutoScale
3. שלב 2: הפעם ניקח מכונה יותר חזקה, t2.large.
4. שלב 3: תנו לקונפ' שם, אפשרו Monitoring.
5. שלב 3: אפשרו detailed monitoring
6. שלב 5: בחרו Security Group שכבר קיים, למשל זה של המחשב הבודד שייצרתם קודם.
7. בחרו את ה-key הרגיל שלכם

בסוף התהליך, ה- AWSיוביל אתכם ישירות למסך יצירת Auto Scaling Group.

יצירת AUTO SCALING GROUP

זהו השלב שבו אתם תקבעו קריטריונים לפיהם מכונות יעלו או ירדו. בשלב הזה יש בעיית ביצה ותרנגולת. את הקריטריונים, כאמור, קובעים מתוך Alarms, אולם ה-alarms האלו מנטרים פרמטרים של scaling groups. לכן ההגדרה צריכה להיעשות בד בבד.

1. שלב 1: תנו שם ל-AS group.
2. שלב 1: Group Size צריך להתחיל ממכונה אחת.
3. שלב 1: בחרו את ה-VPC שלכם, והוסיפו את כל תתי הרשת הציבוריים הזמינים עבור ה-VPC.
4. שלב 1: תחת Advanced Details אפשרו Load Balancing והוסיפו את ה- ELB שלכם.
5. שלב 1: תחת Advanced Details אפסו את ערך ה-Health Check Grace Period .
6. שלב 1: תחת Advanced Details אפשרו CloudWatch Detailed Monitoring.
7. שלב 2: לחצו על Use scaling policies. בתחתית המסגרת שמופיעה יש לינק. לחצו על הלינק. תופיע מסגרת יותר גדולה עם 2 חלקים: Increase Group Size ו-Decrease Group Size.

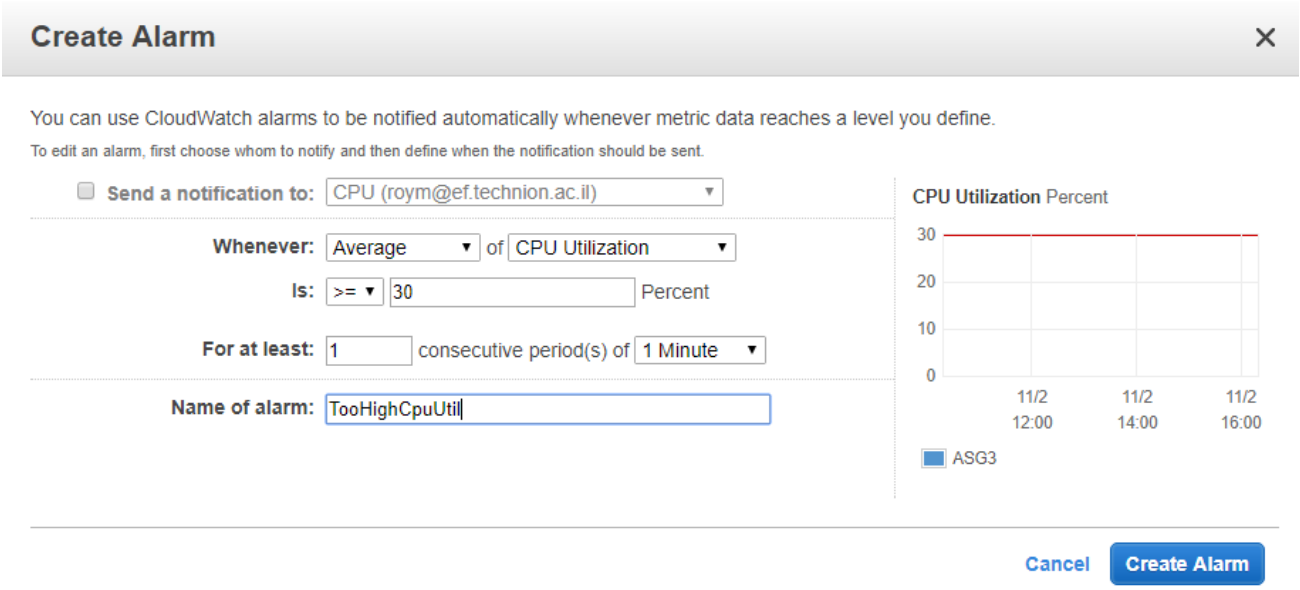
כאן נפתח לכם מסך עם הרבה אפשרויות.

- שורה אחת שמגדירה טווח מכונות

- איזור Increase Group Size שמגדיר מנגנון החלטה להעלאת מספר מכונות
- איזור Decrease Group Size שמגדיר מנגנון החלטה להורדת מספר מכונות

בכל אחד מ-2 האיזורים יש לינק להגדרת Alarm, בו נשתמש כדי להגדיר התרעה עבור ה- Auto Scaling Group שלנו.

8. שלב 2: טווח המכונות שלנו יהיה בין 1 ל-8
9. שלב 2: Increase Group Size: לחצו על Add new Alarm והגדירו Alarm לפי המסך הבא:



10. שלב 2: Increase Group Size: הגדירו הוספה של מכונה אחת בכל צעד.
11. שלב 2: Increase Group Size: Instance Need הכניסו ערך אפס ל-Warm-up.
12. שלב 2, Decrease Group Size: הגדירו Alarm באותו אופן, הפעם קטן מ-20 אחוז. שימו לב שכאן אתם צריכים להפוך את הסימן, ולשנות את שם ה-alarm למשהו יותר הגיוני, שמכיל את המילה low ולא את המילה high.
13. שלב 2, Decrease Group Size: בשורת Take the action הגדירו הסרה של מכונה אחת כל פעם.
14. שלב 4: הוסיפו תג שם. כל מכונה שתעלו, תופיע עם אותו שם.

15. סיימו את ההגדרות.

אם הכל עובד כשורה, מכיוון שמספר המכונות המינימלי הוא 1, באופן די מיידי עולה מכונה לאוויר. תנו לה דקה לעלות.

16. על מנת לוודא שהשרת שעלה כרגע לא ייעלם כשעומס יורד, חזרו למסך ה-Auto Scaling Group לחצו על לשונית Instances, בחרו את השרת שלכם, ותחת Actions אפשרו scale in protection.
17. כדי שתהיה לנו גישה למכונה ב-SSH, חיזרו לדף ה-Elastic IPs, ושייכו את הכתובת שהגדרתם לשרת שזה עתה עלה.

יצירת גרפים לניטור השרות

במסך CloudWatch הראשי אמורים להופיע 2 גרפים עבור ה-Alarms שהגדרתם תוך כדי הגדרת ה-Groups. מכיוון שהגרפים האלו לא כל כך ברורים, נגדיר Dashboard חדש עם גרף יותר טוב שמראה את מצב המכונות.

1. בשירות CouldWatch, בחרו Dashboards, ולחצו על Create Dashboard תנו לו שם דומה לשם ה- Auto Scale Group, מכיוון שהגרפים שהוא יציג יראו סטטיסטיקה על ה-Group.
2. לחצו על Add Widget ובחרו Line Graph.
3. נפתח מסך לבחירת קטגוריה. אנחנו מעוניינים במטריקות של EC2 עבור Auto Scaling Group. מתוך המטריקות האלו, נרצה סטטיסטיקות על CPU Utilization.
4. היכנסו ללשונית Graphed metrics ושנו את הדגימה של הגרף לדקה אחת.
5. לחצו על Create Widget, הרחיבו את הגרף כל שיהיה יותר ברור.
6. הגדירו בדומה, על אותו dashboard, עוד widget עם מטריקות של Auto Scaling. הוסיפו אליו את המטריקות In Service Instances ו-Desired Capacity. דאגו לכך ששני הגרפים יהיו בדיוק אחד מתחת לשני.
7. לחצו על Save Dashboard.

הגרף הראשון משחזר במדוייק את המדידה של ה-Alarm. כלומר, מודדים ממוצע על כל המכונות בקבוצה, כל דקה, פרטמר CPU Utilization. זה יעזור לנו להבי מתי מכונות אמורות לעלות או לרדת.

בדיקות

שלב הבדיקות חשוב, על מנת שלא יתבזבז זמן מיותר בניסוי.

בשלב זה כבר הגדרנו לא מעט דברים, וכדאי לוודא שעד כה דברים עובדים לפי הצפוי. ננסה להגיע לשרת ה-web במכונה שעלתה, דרך ה-Load Balancer שלנו, בדומה למה שעשינו בתרגיל הקודם.

1. היכנסו דרך דפדפן ללינק הזה: <ELB DNS name>/ip.html. קראו למדריך להראות לו מה קיבלתם.

כרגע המצב הוא שיש מכונה אחת של EC2, אשר עלתה אוטומטית, והגרף ב-CloudWatch מראה על נצילות נמוכה.

בשלב הבא אנחנו נתחבר למכונה, נפעיל את תוכנת ה-stress, דבר שיגרום ליצירת Alarm והפעלה אוטומטית של מכונות (Scale Out). המכונות הבאות שיעלו, יהיו בעצם גלמים. הם אמנם ייספרו בממוצע, אבל בפועל לא יבצעו שום פעולה אלא ינוחו.

לאחר שפעולת ה-stress תסתיים, מספר המכונות יירד (Scale In).

עכשיו אנחנו מוכנים לעבודה.

2. לפני שמתחילים בהרצה, אנא וודאו יחד עם המדריך ש:
   - ב-ELB Health Check ה-Ping Path נכון, ו-Healthy Threshold מכוון ל-2
   - הגרפים ב-Dachboard מכוונים כולם לאינטרוול של דקה
   - ה-Alarm בעליה מראה גדול מ-30, ומכוון לדגימה של דקה
   - ה-Alarm בירידה מראה קטן מ-20, ומכוון לדגימה של דקה
   - יש Scale in protection

הרצת עומס

1. מצאו את הכתובת הציבורית של המכונה, והתחברו אליה ב-SSH.
2. מתוך הטרמינל של המכונה המרוחקת, הפעילו את הפקודה stress –c 2 –t 3600
3. בשלב זה יש לנו הרבה כלים לבדוק אם משהו באמת קורה.
   - ב-CouldWatch אפשר לראות את הגרף שייצרנו ב-dashboard

- ב-Auto Scaling Group אפשר ללכת ללשונית ההסטוריה של הקבוצה שלנו ולראות מתי עולות ויורדות מכונות
- ב-EC2 אפשר גם כן לראות באופן כללי אילו מכונות למעלה.
- ב-ELB אפשר לראות אלו מכונות עוברות ל-In Service.

מומלץ מאד לפתוח 4 לשוניות שונות בדפדפן כדי לעקוב אחר הנעשה בכל השרותים האלו בו זמנית.

אם אין תזוזה תוך 2 דקות בגרפים וב-Group, כנראה שיש בעיה. קיראו למדריך.

4. אחרי שמכונות מתחילות לעלות, בדפדפן, תרפרשו את הדף עם הלינק ל-ELB שלכם. אתם אמורים לראות כל פעם כתובת IP שונה. קיראו למדריך להראות לו שזה עובד
5. חכו שהמערכת תתייצב, וגרף ה-CPU יראה קו מאוזן ל-3 דגימות. לפעמים המערכת לא מתייצבת. במקרה כזה, אחרי שהגרף הגיע פעמיים לנק' מקסימום והתחיל לרדת, קיראו למדריך.

להגשה

שאלה מס' 3.1 האם המערכת התייצבה? לכמה מכונות הגענו?

שאלה מס' 3.2 גרף ה-CPU היה על איזור ה-100 אחוז, ואז התחיל לרדת כאשר נוספו מכונות. למה?

שאלה מס' 3.3 כמה מכונות בפועל "נושאות בנטל", כלומר המעבד שלהם באמת עמוס?

שאלה מס' 3.4 עם הסף שנקבע עבור העלאת מכונה נוספת, כמה מכונות נדרשות כדי לעמוד בעומס?

6. הפסיקו את פעולת ה-Stress.

שאלה מס' 3.5 אחרי שמכונות סיימו לרדת, צלמו את ה-dashboard ואת ה-group history.

שאלה מס' 3.6 מה לדעתכם היה קורה אם לא היינו מסמנים scale in protection על השרת הראשון שעלה?

תרגיל 4: קיפול

זה השלב לעצור הכל, וגם למחוק את כל מה שכבר לא נשתמש בו יותר.

עיצרו ומיחקו את כל השרתים

מיחקו את כל ה-buckets

מיחקו את ה-ELB.

מיחקו את ה-security groups אשר לא בשימוש. הם לא נמחקים אוטומטית כאשר מוחקים שרתים, ונוטים להצטבר.

מיחקו את הגדרות ה-Auto Scale.