נושא:	**מסמך ארכיטקטורה באירוח ענן למרפאות פרטיות – קליניקל**
עורך אחרון:	Yaron Ben Shalom
תאריך עדכון אחרון:	03:30:00 17/12/2020
גרסה:	0.4

תוכן עניינים

תוכן עניינים 2
טבלת שינויים 3
טבלת מסמכים קשורים 3
כללי 4
ארכיטקטורה [חוז-ד-3] 4
הקצאת משאבים 6
אבטחת מידע 7
עקרונות מנחים 7
פיתוח מאובטח 7
הזדהות ומדיניות סיסמאות 9
Best practices לשירותי AWS 9
הגנות ברמת תשתיות 10
הערכת סיכונים 11
טבלת תקשורת 13
חלוקת תפקידים בניהול שירות הענן 14
הצפנה [חוז-ד-3.5][פרט-B.3][פרט-4.12] 14
הצפנת מידע במנוחה (Data at rest) 15
הצפנת מידע בתנועה (Data in transit) 15
העלאת קבצים 15
פרטיות 16
הפרדה בין לקוחות [פרט-6.C.1] 16
שרידות 16
שרידות ברמת השירותים (DRP) 16
גיבוי 16
ניטור [חוז-ד-4] 17
לוגים 17
לוגים ברמת השירות 17
לוגים אפליקטיביים 17
לוג מעקב (Audit) [פרט-2/3.C.1] 17
ניטור שירותים 17
ניטור אבטחת מידע 18
תחזוקה 18
התקנת עדכונים [פרט-iv.D.3] 18
אוטומציה 18
ניהול מדיניות מרכזית 18
תמיכה 18

טבלת שינויים

גרסה	עורך	תאריך תחילת עריכה	תוכן
0.1	ירון בן שלום	8.6.2020	טיוטה ראשונה
0.2	ירון בן שלום	28.6.2020	טיוטה שניה
0.3	ירון בן שלום	30.6.2020	התייחסות למסמך פרטיות
0.4	ירון בן שלום	6.7.2020	סגירת נושאים פתוחים

טבלת מסמכים קשורים

שם וקישור	כותרת	קיצור	עורך	תאריך עדכון
חוזר מחשוב ענן להערות ציבור	חוזר ראש חטיבת רגולציה, בריאות דיגיטלית ומחשוב	[חוז-סעיף]	משרד הבריאות	9.7.2019
Privacy by Design_14052020	Privacy by Design Principles	[פרט-סעיף]	יועצת פרטיות	14.5.2020

כללי

מערכת קליניקל היא מערכת מבוססת טכנולוגיות Web המספקת יכולות ניהול תיק רפואי.
המערכת מבוססת מוצר Open Source בשם OpenEMR וכוללת פיתוחים ותיקונים רבים שבוצעו על ידי צוות הפיתוח במטריקס.
באופן מסורתי המערכת הותקנה בסביבת On premise לטובת מרפאות שנמצאות באחריות וניהול (ברמה כזו או אחרת) של משרד הבריאות.
כעת הצוות מבצע התאמות של המערכת לטובת מרפאות פרטיות, גם בצד האפליקטיבי וגם בצד התשתיתי.
בצד התשתיות הוחלט להקים מופע של המערכת על בסיס שירות ענן מתאים כדי לעמוד במטרות המרכזיות הבאות:

- חשיפה של המערכת ללקוחות קצה בכל רחבי הארץ
- אפשרות גידול קלה במקרה של צרכנים מרובים
- אבטחת מידע ברמה גבוהה
- שרידות ויתירות ברמה גבוהה
- שמירה על עקרונות הגנת הפרטיות
- אפשרות לניידות סבירה בין שירות הענן של AWS להתקנה On premise
- בניה מחדש (Refactoring) של רכיבי מערכת תבוצע במידה סבירה בכפוף לדרישות הרצת המערכת על בסיס תשתיות ענן

ארכיטקטורה [חוז-ד-3]

הנחות בסיס:

- כלל רכיבי המערכת בענן AWS יאורחו ב- eu-west-1 (Ireland) [פרט-5.C.1]
  - יש לציין שאירלנד היא חברה באיחוד האירופי [חוז-א-7.3]
- כל רכיבי המערכת יחולקו ל- Resource groups ויתויגו בהתאם
- רכיב האפליקציה המרכזי ימשיך להיות מבוסס Apache וייארז בתוך docker container אך לא תבוצע שבירה פנימית ל- Micro services
- תתווסף תמיכה בשמירת מסמכים במערכת על בסיס שירות AWS S3
  - במידת הצורך ניתן יהיה לעשות שימוש באפשרות הקיימת של שמירת מסמכים במסד נתונים CouchDB
  - לא תהיה מחיקת קבצים בפועל אלא שימוש ב- Delete markers על בסיס Versioned objects
- כיום התוצרים במערכת נבנים ידנית על בסיס CLI Scripts
  - יש לשקול מעבר לבניה בצורה אוטומטית על בסיס Jenkins
- הפרדת מתחמי המערכת תבוצע באמצעות הגדרת VPC [פרט-iii.D.3]
  - ניתן לשקול ריכוז של השליטה בכלל התעבורה (וביחוד זו היוצאת) באמצעות שירות Firewall Manager
- הנחיות בנוגע ל- Containers
  - מערכת ההפעלה בבסיס ה- Container images תהיה Alpine 3.x
  - בסיס ה- Image יהיה זה של OpenEMR הרשמי
  - ה- Container images ישמרו ב- Container registry של AWS ויתויגו בהתאם לסביבות וגרסאות
  - התקנה לתוך EKS תבוצע באמצעות Code Pipelines
  - תוצרים נלווים כדוגמת מבנה מסד נתונים יותקנו באמצעות SQL Scripts שיורצו בתוך ה- Container
  - הגדרות סביבה יועברו כ- Environment variables

- Container images בגרסאות יציבות יפורסמו גם ב- docker hub לטובת קידום תוצרי ה- Open source של הפרויקט
- איסוף לוגים יבוצע על בסיס CloudWatch/FluentD

- גישה ניהולית לסביבת הענן תתאפשר רק באמצעות VPN
- ארכיטקטורת המערכת מתבססת על רכיבים שלחלקם ישנה מקבילה מקובלת גם מחוץ לענן [פרט-6.C.1][פרט-4.9]:
  - MariaDB = RDS for MariaDB
  - Kubernetes = EKS

  בעוד לאחרים יש חלופה יישומית:
  - CouchDB <= S3
  - Cellact ,Twilio <= SNS ואחרים
  - RabbitMQ <= SQS
  - ELK <= CloudWatch, CloudTrail
  - nginx + modSecurity <= WAF

להלן תרשים ארכיטקטורה עבור סביבת הייצור:

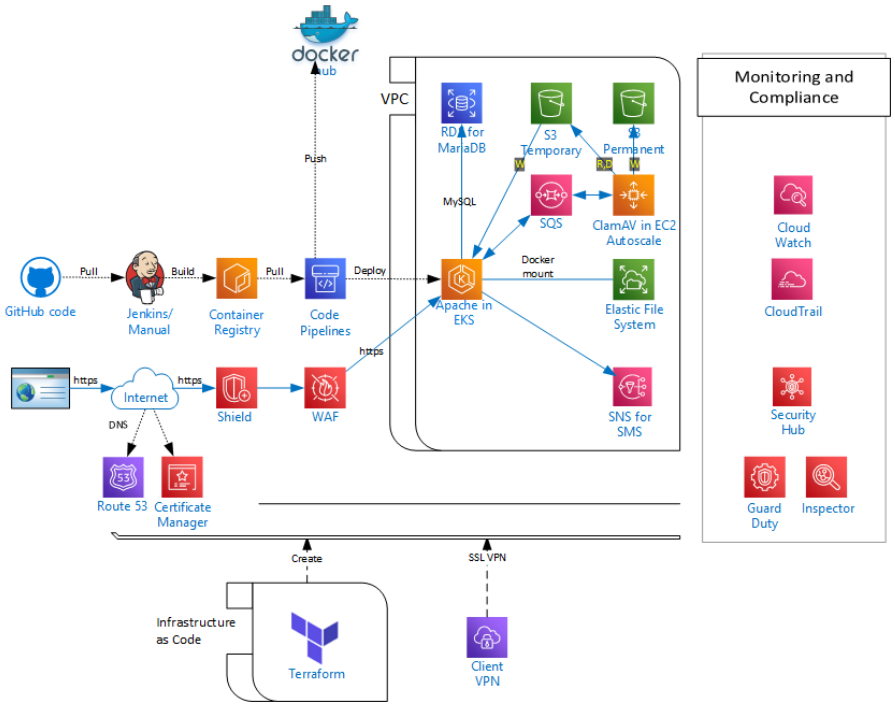


Figure1

הקצאת משאבים

שירות	Tier/Plan/Type	מהות
AWS Certificate Manager	Free	ניהול תעודות SSL/TLS פומביות
AWS Client VPN	Paid	חיבור מאובטח של מנהלי מערכת למשאבי ענן
AWS CloudWatch	Always Free (!) מעבר ל- Paid כאשר נפח הלוגים עולה על 5GB (dashboards ,logs)	איסוף לוגים מכל רכיבי המערכת ניטור ותצוגות דוחות על בסיס הלוגים
AWS CloudTrail	Free (!) מעבר ל- Paid כאשר צריך יותר מ- 90 ימים של היסטורית אירועים	מעקב, ניטור וניהול תאימות של חשבון AWS
AWS CodePipeline	Always Free (!) מעבר ל- Paid כאשר צריך יותר מ- Pipeline אחד פעיל בחודש	מנגנון CD לניהול התקנות של Releases
AWS Container Registry	Free (!) מעבר ל- Paid כאשר צריך יותר מ- MB500 ולאחר 12 חודשים	מאגר Docker (container) images
AWS EC2 Auto scale	Paid (!) נדרש לטובת כוח עיבוד גבוה יותר ולאחר 12 חודשים	כוח עיבוד גמיש לטובת סריקת אנטי וירוס
AWS Elastic File System	Paid (!) נדרש לאחר 12 חודשים	ניהול מערכת קבצים לטובת Persistence של מידע
AWS GuardDuty	Paid	שירות זיהוי סיכוני אבטחת מידע (Threat detection)
AWS Inspector	Paid	שירות הערכת רמת אבטחת מידע בשוטף
AWS Managed Kubernetes service	Paid (SSD ,t3.medium)	תזמון וניהול Containers על בסיס Kubernetes
AWS RDS for MariaDB	Multi-AZ (10.4.x ,SSD ,db.t3.2xlarge)	שירות מסד נתונים מנוהל על בסיס מנוע MariaDB
AWS Route 53	Paid	שירות DNS
AWS Security Hub	Paid	איחוד וניהול מרכזי של כלל מנגנוני אבטחת המידע ב- AWS

Commented [YBS1]: לבדוק במסמך פרטיות

שירות	Tier/Plan/Type	מהות
AWS Shield	Standard/Free (!) מעבר ל- Advanced כאשר נדרשות יכולות מתקדמות	שירות הגנה בפני DDoS
Amazon Simple Notification Service	Paid per SMS	שליחת SMS לטובת MFA
AWS Simple Queue Service	Always Free (!) מעבר ל- Paid כאשר צריך יותר מ- M1 בקשות בחודש	שירות Message queue
AWS Simple Storage	Intelligent – Tiering	אחסון קבצים מותאם לרמת שימוש לא ידועה מראש ומעבר אוטומטי בין רמות אחסון שונות
AWS Virtual Private Cloud	Paid	הפרדה (Isolation) של משאבי הענן של המערכת
AWS Web Application Firewall	Paid (rules ,web ACLs)	הגנה על יישומי Web

הערות:

- מיטוב עלויות (Cost optimization), למשל על בסיס הקצאת משאבים מראש לשנה או שימוש בתוכניות חסכון, יבוצע קרוב לבנית הארכיטקטורה בסביבת הייצור
- במקומות בהם צוינה הקצאת המשאבים במספרים יש להבין כי מדובר בהערכה בלבד;
הקצאת משאבים מדויקת יותר תערך בכפוף להגדרת ביצועים נדרשים וביצוע בדיקות ביצועים תחת עומס.
בנוסף, הקצאת המשאבים תשופר בשוטף בכפוף לנתונים שיחשפו באמצעות מנגנוני הניטור השונים.

אבטחת מידע

עקרונות מנחים

פיתוח מאובטח

- כללי
  - פיתוח
    - תהליכי הפיתוח יבוצעו בהתאם לנוהל הפיתוח המאובטח של מטריקס בגרסתו העדכנית הנמצאת כאן
  - הקשחה [פרט-ii.D.3]
    - תשתית OpenEMR תוקשח על פי המלצות הקהילה ובנוסף:
      - Mode - Do Not Show SQL Queries = true
      - Idle Session Timeout Seconds = 300
      - Permit unsalted passwords = false
      - במידת הצורך נתמוך גם ב- Timeout כאשר המשתמש אינו Idle [פרט-5.1]
      - במידת הצורך תבוצע הצפנה של PHP Sessions [פרט-7.C.1]
    - התקנת גרסאות רכיבים עדכניות
      - תשתית הפתרון היא גרסת OpenEMR 5.0.2 שהיא הגרסה העדכנית ביותר של מוצר הבסיס. הקהילה עובדת בימים אלו על גרסה 5.0.3.

עדכון גרסאות עתידיות יבוצע בהתאמה לתכולת הגרסאות המשוחררות על ידי הקהילה.

- הפרדת רכיבים
  - תבוצע הפרדת משימות לרכיבים ייעודיים מומלצת גם כדי למטב את תחזוקת המערכת וגם מבחינת אבטחת המידע בהתאם למתואר בשרטוט הארכיטקטורה.
    מכוון שהמערכת לא תעבור Refactoring מלא, חלק מן המשימות במערכת ימשיכו להיות מבוצעות על ידי אותו רכיב.
    הכוונה בעיקר לשרת האפליקטיבי (מבוסס Apache) שממשיך לבצע את המשימות הבאות:
    - מימוש מערכת הניהול בגרסה הישנה (Apache/PHP)
    - מימוש שכבת שירותים לטובת מערכת הניהול בגרסה החדשה (Apache/PHP)
    - אירוח קבצי מערכת הניהול בגרסה החדשה (Apache/React)
- חשבונות גישה
  - כל אחד מן החשבונות במערכת צריך להיות מוגבל למשימה ייעודית וצריך לקבל את ההרשאות המינימליות הנדרשות לביצוע המשימה (Least privilege), תוך מתן דגש על:
    - חשבונות ניהוליים של מנהלי מערכת בסביבת הענן (ראו טבלת " חלוקת תפקידים בניהול שירות הענן")
    - חשבונות שירותים
      - גישה למסד הנתונים
        - שם משתמש ייעודי וסיסמה קשה
      - גישה ל- S3
        - שימוש ב- Policies מתאימים על בסיס IAM

- ניהול קוד
  - כלל הקוד ינוהל בשירות GitHub התומך בפרוטוקול git המבוזר לניהול פרויקטים של תוכנה בצורה יעילה ואמינה [פרט-i.E.3]
  - Pull requests
    - הגדרת ה- Branches הרלוונטיים ב- Github כ- Protected והגדרת מדיניות שדורשת
      - At least 1 approving review
      - Strict status checks
- סריקת קוד כחלק מתהליך הפיתוח היא תהליך מומלץ כדי להסיר קוד בעייתי או בלתי מאובטח מוקדם ככל שניתן. ישנם שני סוגי סריקת קוד מרכזיים:
  - סריקת קוד סטטית (Static code analysis)
    - ניתן לעשות שימוש בכלי קוד פתוח ללא עלות רישוי כדוגמת SonarQube
    - ניתן לעשות שימוש בשירות ענן כדוגמת Fortify On Demand שעשויים להיות זמינים גם כשירות וגם כהתקנה On premise
  - סריקת פגיעות של רכיבי קוד פתוח (Open source vulnerability scan) [פרט-i.D.3]
    - ניתן לעשות שימוש בכלי קוד פתוח ללא עלות רישוי כדוגמת Snyk (מתאים לפרויקטי קוד פתוח)
    - ניתן לעשות שימוש בגרסאות בתשלום של Snyk או מתחרים כדוגמת Whitesource
    - בנוסף לאמרו לעיל יעשה שימוש ב- ב- Image scanning המובנה ב- ECR
- בהיות כלל התוצרים של המערכת חשופים כקוד פתוח לעולם כולו, ניתן יהיה לקבל משוב ממשתמשי קצה ובעלי עניין אחרים לגבי רמת האבטחה של המערכת על בסיס GitHub issues [פרט-vi.D.3]

הזדהות ומדיניות סיסמאות

- כל פעולה במערכת, בין אם היא ב- Backoffice הישן או העדכני, תדרוש הזדהות [פרט-1.i.A.3]
- אוכלוסיית המשתמשים שמורה במסד הנתונים וההזדהות מבוצעת אפליקטיבית
  - בשכבת ה- Backoffice באמצעות Session
  - בשכבת ה- API באמצעות Short lived tokens שעברו Hash
- המערכת מיישמת מנגנון ACL ושיוך לתפקידים [פרט-2.i.A.3]
  - התפקידים עשויים להשתנות בהתאם לסוג מרפאה
  - כל משתמש ישויך רק לתפקידים הרלוונטיים עבורו [פרט-ii.A.3]
  - מספר המשתמשים הניהוליים יצומצם למינימום [פרט-iii.A.3]
- סיסמאות משתמשים יוצפנו במסד הנתונים באמצעות Bcrypt או Argon
- מדיניות סיסמאות [פרט-v.A.3]
  - Require strong passwords = true
  - Minimum password length = 8
  - Maximum Failed Login Attempts = 3
  - לא יעשה שימוש ב- Password expiration
  - Idle session timeout seconds = 600
  - לא תהיה תמיכה בתרחיש טיפול עצמאי ב"שכחתי סיסמה"
  - כישלונות בהזדהות נרשמים ללוג המערכת [פרט-vii.A.3]
- המערכת תדרוש MFA באמצעות SMS [פרט-C.1.1][פרט-vi.A.3]
  - כאמור לעיל, יעשה שימוש בשירות SNS של AWS
  - כיום SMS נחשב למנגנון פחות מאובטח משום שניתן לעקוף אותו באמצעות Social engineering או החלפת SIM. ניתן לשקול החלפה של מנגנון כזה במנגנון מבוסס יישום כדוגמת Google Authenticator.
- גם ניהול שירות הענן יוגדר לדרוש הזדהות באמצעות MFA
  - יש תמיכה מובנת במגוון Devices כולל Virtual devices

BEST PRACTICES לשירותי AWS

- שימוש בחוקים של AWS Security Hub לקביעת מדיניות אבטחת מדינה שנבדקת באופן שוטף
- Security Best Practices for Amazon RDS
- Amazon S3 Preventative Security Best Practices
- Amazon EKS Best Practices Guide for Security

הגנות ברמת תשתיות

הגנות ברמת תשתית התקשורת תסופקנה באמצעות 3 מרכיבים מרכזיים:

- Security groups – הגדרות תקשורת מותרת ברמת ה- VPC
- AWS WAF – We application Firewall להגנה על מערכות Web החשופות לאינטרנט תוך שימוש בחוקים מובנים עבור, בין השאר: [פרט-iii.F.3]
  - חסימת תעבורה מאזורי IP אסורים
  - הגנה בפני SQL Injection
  - הגנה בפני התקפות XSS
  - הגנה בפני HTTP Flooding
- AWS Shield – הגנה בפני התקפות DDoS
- הפרדת סביבות
  - לא יעשה שימוש חוזר במשתמשים וסיסמאות זהות בין סביבות שונות
  - מידע אישי או רגיש מסביבות גבוהות (ייצור) לא ימצא בסביבות נמוכות [פרט-4.6]

הערכת סיכונים

להלן הסיכונים המרכזיים ביישום המערכת המדוברת בסביבת ענן:

- חשיפת מידע [חוז-ב-1]
  - כתוצאה מהפרדה לא יעילה בין לקוחות הענן
    - ההפרדה תבוצע בשתי רמות
      - הפרדה אפליקטיבית ברמת סוג המרפאה: כל סוג מרפאה יופעל על בסיס Container שונים תוך שימוש ב:
        - כתובות גישה שונות
        - חשבונות שירות שונים
        - מפתחות הצפנה שונים
      - הפרדת מידע ברמת מסד הנתונים: כל מרפאה תופעל על בסיס מסד נתונים נפרד
    - פרטים נוספים בסעיף "הפרדה בין לקוחות"
  - לגופים ממשלתיים מחוץ לישראל
    - כמצוין לעיל, המערכת תאורח במדינות שהן חלק מן האיחוד האירופי
    - מדיניות AWS היא לידע את הלקוחות לפני חשיפת כל מידע בצורה הזו
  - כתוצאה מהעברת מידע שנותר בסביבת הענן לאחר סיום התקשרות
    - כל המידע הרגיש מוצפן At rest כמתואר בסעיף "הצפנת מידע במנוחה (DATA AT REST)"
    - מדיה שהגיע לסיום חייה מושמדת בהתאם ל- NIS 800-88
    - מידע בחשבון AWS שנסגר נמחק לאחר 90 יום
  - מניעת חשיפת מידע על ידי עובדי אחראי סביבת הענן תיושם ברמת הסכם העבודה בינו לבין משרד הבריאות. בנוסף הקפדה על הפרדת תפקידים ומזעור כמות המנהלים עם גישה בהרשאות גבוהות לסביבה תקטין את הסיכון לחשיפה כזו של מידע רגיש.
  - על ידי גישה למכשירי קצה [פרט-B.3]
    - כלל המידע במערכת נשמר ב- Backend בלבד וב- Cache מקומי בדפדפנים ישמר מידע בלתי רגיש ובלתי מזוהה אישית בלבד
- אובדן או שיבוש מידע
  - בצד ספק שירות הענן [חוז-ב-2]
    - המידע הרגיש, השמור ב- Backend, יוצפן At rest כאמור לעיל
    - ספק שירות הענן AWS עומד בתקנים רבים שמבטיחים רמת אבטחת מידע גבוהה ולכן הסיכוי לשיבוש מידע בצורה זדונית נמוך מאד
    - ניתן ליישם גיבוי מחוץ לסביבת הענן המידית במספר צורות:
      - העתקת Database snapshots ל- Region אחר
        - ניתן לבצע ידנית ב- Console או על ידי אוטומציה של CLI או API
      - העתקת S3 buckets ל- Region אחר
        - מוגדר על בסיס הגדרות השירות (XML)
      - שימוש בשירות AWS Backup לגיבוי מרכזי של שירותים כדוגמת RDS ו- EFS כולל תמיכה בגיבוי ל- Region אחר
      - שימוש בכלי צד שלישי כדוגמת Veritas Backup Exec לגיבוי Hybrid בין סביבת הענן לסביבת On premise
  - בצד היישום [פרט-4.7]
    - ניתן לשלב פתרונות DLP לתוך EKS, למשל
      - NeuVector AWS EKS and ECS Container Security
      - Nightfall
- אובדן זמינות המידע [חוז-ב-3][פרט-C.3]

- תשתית הענן AWS בנויה בצורה מיטבית ומאפשרת יישום מערכות עם שרידות וזמינות ברמה הגבוהה ביותר
- ספק שירות הענן AWS מבטיח רמת SLA גבוהה לכלל השירותים שלו ולכן הסיכוי לאובדן זמינות למידע נמוך מאד; בנוסף המערכת תבנה על בסיס עקרון No-single-point-of-failure כדי להבטיח שכשל נקודתי בכל רכיב לא ישבית אותה
- תמיכה בזמינות המערכת מחוץ לגבולות ה- Region אפשרית על בסיס תכנון מראש של תמיכה בריבוי Regions ותלויה בפרמטרים כגון רמת הזמינות הנדרשת מן ה- Region השני, רמת העדכניות של המידע השמור בו, כמות הפעולות הנדרשות במקרה של מעבר בין Regions וכדומה.
  - ברמת IaC ניתן לעשות שימוש ב- Terraform Modules
  - ברמת RDS אפשר להתבסס על Read replicas ובמקרה הצורך לבצע Promotion
  - ברמת S3 נושא ה- Replication נידון בסעיף "אובדן או שיבוש מידע"
- הגנה בפני DDoS תסופק על ידי AWS Shield
- הקצאת משאבים
  - ההקצאה הראשונית תבוצע על בסיס Sizing מוערך על בסיס פרמטרי שימוש צפויים
  - מימוש של פתרונות Auto scaling על בסיס יכולות AWS EKS יוודאו כי הקצאת המשאבים לשירותי האפליקציה במערכת תותאם אוטומטית על פי השימוש בה
    - ניתן לשקול שימוש ב- Storage auto scaling עבור RDS במידה ורמת השימוש באחסון בשכבת מסד הנתונים לא ניתנת להערכה מראש
  - ניטור מתמשך של המערכת והגדרת התראות מתאימות יוודא כי מנהלי המערכת מודעים לצוואר בקבוק מבעוד מועד
- אמינות החיבור לאינטרנט מכל אתר לקוח תהיה באחריות הלקוח עצמו

- ספק שירות הענן (AWS) [חוז-ג]
  - AWS מפעילה את מרכזי המחשוב שלה ברמה תואמת ל- Tier III
  - AWS תומכת בתקנים הבינלאומיים הבאים
    - חובת עמידה:
      - ISO-9001
      - ISO-27018
      - ISO-27017
      - ISO-27001
      - ISO-27701
      - ISO-20000
      - AICPA
      - PCI
      - CSA
      - SOC-2
      - CCPA
    - יכולת תמיכה:
      - HIPAA
      - GDPR
      - ASIP HDS
    - מוצר הבסיס OpenEMR עומד בתקן ONC Ambulatory EHR Certification
  - AWS תומכת במגוון רחב של Regions וניתן להקצות שירותים שונים ב- Regions שונים על פי בחירת הלקוח
  - AWS תומכת ב- 74 אזורי זמינות (Availability zones)

Commented [YBS2]: @אלונה, שימי לב שאין כרגע אזכור של עמידה בתקן הזה במסמכי AWS

- ניתן לעשות שימוש ב- AWS Artifact לקבל Compliance reports ומידע על Agreements רלוונטיים
- ניתן לבצע בדיקות חדירות מול שירותי AWS מסוימים על פי המדיניות הפומבית
- מידע על נושאי אבטחת מידע מפורסם ב- Security Bulletins

- מיפוי המידע [חוז-ד-1]
  - המידע הרפואי במערכות המדוברות מסווג כחסוי
  - ממשקים חיצוניים
    - לא מתוכננים בשלב זה ממשקים למערכות חיצוניות
  - הרשאות ותפקידים בתוך המערכת
    - המערכת מיישמת מנגנון הרשאות וחלוקת תפקידים פנימי ותמשיך לעשות שימוש במנגנון זה גם בסביבת הענן
- ניתוח סיכונים [חוז-ד-2]
  - במידת הצורך ניתן לבצע תהליך מלא של ניתוח סיכונים בסיוע גוף שמתמחה בתחום
  - במידה ויבוצע תהליך כזה, יש לסקור את ממצאיו וליישם טיפול מתאים בכל אחד מהם בהתאם לחומרתו ובתיאום עם הגוף שביצע את הניתוח
- בחינת הפתרון בתחום אבטחת המידע [חוז-ד-5]
  - אחראי סביבת הענן יבצע Review של הארכיטקטורה בסיוע גוף מתמחה בתחום אבטחת המידע
  - אחראי סביבת הענן יבקש ביצוע של Penetration test עבור המערכת לפני חשיפתה בפני משתמשי הקצה על ידי גוף מתמחה בתחום אבטחת המידע
    - במידת הצורך אותו גוף יבצע Penetration test חוזר כל 18 חודש [חוז-ד-7.3]
- מעבר לסביבת ייצור [חוז-ד-6]
  - אחראי סביבת הענן יבצע הדרכה של משתמשי קצה בהתאם לנדרש ובכפוף להסכם שלו עם משרד הבריאות

טבלת תקשורת

מקור	יעד	מטרה	Port	PrivateLink
דפדפן (אינטרנט)	מערכת	גלישה לממשק Web של המערכת והפעלת Backend APIs משכבת React	443	
דפדפן (אינטרנט)	Route 53	שאילתות DNS	53	
GitHub	Jenkins	שליפת קוד לבניית תוצרים	443	
Jenkins	Container Registry	בנית ושמירת Container images	443	כן
Apache container	RDS	גישה למסד הנתונים	3306	
Apache container	SQS	שימוש ב- Message queue	443	כן
Apache container	S3	שימוש ב- Simple storage	443	VPC endpoint
מערכת	Cloudtrail	ניטור		כן
מערכת	CloudWatch	איסוף לוגים	Agent	כן

הערה: ניתן לשקול הגבלת תעבורה מן האינטרנט לכתובות IP ספציפיות של מרפאות כדי להגביר את האבטחה על המערכת. הגדרה כזו תידרש להיות חלק מתהליך ההקמה של מרפאה חדשה. [פרט-viii.A.3]

חלוקת תפקידים בניהול שירות הענן

הפרדת התפקידים בניהול של שירות הענן (Console access) באה לצמצם את הסיכון הנובע מהענקת הרשאות גורפות למספר גדול של משתמשים ניהוליים.

תפקיד	סוגי משאבים רלוונטיים	Policies	אוכלוסיית משתמשים
מנהל מערכת	הכל	AdministratorAccess	צוות תשתיות
מנהל תשתיות	CloudWatch, CloudTrail	CloudWatchFullAccess AWSCloudTrailFullAccess	צוות תשתיות
מנהל פיתוח	CloudWatch, CloudTrail, RDS, S3, EFS, ECR, CodePipeline	CloudWatchReadOnlyAccess, AWSCloudTrailReadOnlyAccess, AmazonS3ReadOnlyAccess, AmazonRDSReadOnlyAccess, AmazonECS_FullAccess, AmazonEC2FullAccess, AmazonElasticFileSystemFullAccess, AWSCodePipelineFullAccess, AmazonEC2ContainerRegistryPowerUser	ראש צוות פיתוח
אחראי אבטחת מידע	Security Hub, GuardDuty, Inspector, WAF	AWSSecurityHubFullAccess AmazonGuardDutyFullAccess AWSWAFFullAccess AWSSecurityHubFullAccess	צוות אבטחת מידע
מנהל מאגרי מידע	RDS, S3	AmazonS3FullAccess, AmazonRDSFullAccess	צוות תשתיות
מנהל אפליקציה	EKS, EC2, EFS	AmazonECS_FullAccess, AmazonEC2FullAccess AmazonElasticFileSystemFullAccess	צוות תשתיות
מנהל תקשורת	VPC, Route 53, SQS	AmazonVPCFullAccess, AmazonRoute53FullAccess, AmazonSQSFullAccess	צוות תשתיות
מנהל DevOps	ECR, CodePipeline	AWSCodePipelineFullAccess, AmazonEC2ContainerRegistryFullAccess	צוות DevOps

הערות:

1. מומלץ לייצר Custom roles כדי לאגד Policies רלוונטיות
2. יש לשקול שימוש ב- Service linked roles במידת האפשר
3. מומלץ ליישם את ההמלצות השונות בכל הקשור ל- IAM

הצפנה [חוז-ד-3.5][פרט-B.3][פרט-4.12]

הצפנת מידע במנוחה (DATA AT REST)

- נתונים השמורים ב- MariaDB יוצפנו במנוחה באמצעות היכולות המובנת ב- RDS (AES-256) בניהול AWS KMS
  - יש לציין כי כאשר מופע RDS מוצפן, גם הלוגים והגיבויים שלו מוצפנים
- נתונים השמורים ב- S3 יוצפנו במנוחה באמצעות היכולת המובנת ב- S3 בניהול AWS KMS
- נתונים השמורים ב- File system יוצפנו במנוחה באמצעות היכולת המובנת ב- EFS
- לוגים השמורים ב- CloudTrail מוצפנים ברמת S3 Server Side Encryption
- במידת הצורך ניתן בכפוף לדרישת מרפאה מסוימת לממש שימוש במפתחות המנוהלים על ידי הלקוח
  - עשוי לדרוש פעילות הקמה של שירות מסד נתונים נפרד עבור אותה מרפאה
- במידת הצורך ניתן לממש הצפנה של מידע רגיש, הנשמר ב- PHP Session, באמצעות Session Handler מתאים

הצפנת מידע בתנועה (DATA IN TRANSIT)

- תקשורת מול RDS
  - תקשורת מול מסד הנתונים צריכה להיות מוגדרת לעשות שימוש ב- TLS
- תקשורת מול S3
  - תקשורת מול S3 מחייבת תמיכה ב- TLS 1.0 לפחות, אנחנו נתמוך ב- TLS 1.2
- תקשורת מול ECR
  - תקשורת מול ECR מחייבת תמיכה ב- TLS 1.0 לפחות, אנחנו נתמוך ב- TLS 1.2
- תקשורת מול EFS
  - תקשורת מול EFS תומכת ב- TLS 1.2
- תקשורת מן העולם
  - תקשורת מדפדפנים של משתמשי קצה תוצפן באמצעות TLS 1.2 על בסיס תעודה פומבית רשמית

העלאת קבצים

הנחות יסוד:

- המערכת תאפשר העלאת קבצים הנמצאים ב- White list בלבד (= Secure Upload Files with White List true)

קבצים המועלים ל- S3 יעברו את התהליך הבא:

- שמירת הקובץ ב- Bucket זמני
- סריקה באמצעות ClamAV על בסיס תהליך דומה למתואר בתבנית S3 VirusScan [פרט-i.F.3]
- קבצים תקינים יועברו ל- Bucket קבוע עם זיהוי זהה למקורי
- קבצים שאינם תקינים ימחקו

פרטיות

הפרדה בין לקוחות [פרט-6.C.1]

המערכת תיישם הפרדה בין לקוחות בנקודות הבאות:

- עבור כל ורטיקל (סוג מרפאה פרטית)
  - כתובת גישה חיצונית נפרדת
  - מופע שירות RDS נפרד
  - מופע שירות S3 נפרד
  - מופעי Micro service (Apache) נפרדים
- עבור כל מרפאה
  - מופע מסד נתונים נפרד
  - חוצץ S3 נפרד

שרידות

שרידות ברמת השירותים (DRP)

אנא ראו סעיף "אובדן זמינות המידע" תחת "הערכת סיכונים".

גיבוי

- גיבוי מסד נתונים (RDS)
  - יעשה שימוש במנגנון הגיבוי המובנה בשירות
  - גיבוי אוטומטי יבוצע במועדים (Backup windows) בהם צפויה פעילות מופחתת במערכת; מועד הגיבוי יקבע להיות 02:00
  - גיבויים ישמרו למשך 10 ימים
    - ניתן להאריך את תקופת השמירה (Retention) עד 35 ימים על בסיס היכולת המובנה של השירות
    - שמירת גיבויים למשך זמן ארוך תדרוש הפעלת או יישום רכיבים נוספים, נא לעיין בסעיף "אובדן או שיבוש מידע" לפרטים נוספים
  - גיבויים אוטומטיים הם מוצפנים
  - מכוון שלוג היישום נשמר במסד הנתונים, גם הלוג יגובה כחלק מגיבוי זה [פרט-4.2]
- גיבוי קבצים (S3)
  - בהתחשב בעובדה שאנחנו נעשה שימוש ב- Versioned objects עם Soft delete, אין צורך במימוש מנגנון גיבוי נוסף עבור הקבצים השמורים ב- S3
    - מומלץ לעשות שימוש ב- Lifecycle management ו- Object locks כדי לשלוט בתקופת השמירה של מידע השמור ב- S3
- גיבוי מחוץ לסביבת הענן המידית
  - נא לעיין בסעיף "אובדן או שיבוש מידע"

Commented [YBS3]: @אלונה – לבדוק האם נדרש גיבוי לזמן ארוך יותר

ניטור [חוז-ד-4]

לוגים

הנחיות כלליות

- לוגים הנאגרים ב- CloudTrail נשמרים כברירת מחדל ללא הגבלת זמן
  - ניתן להגדיר Lifecycle management rules ברמת S3 כדי לשנות את תקופת שמירת הלוגים [פרט-iv.E.3]
- שמירה על אמינות הלוגים תיאכף על ידי הפעלת Log integrity validation [פרט-C.4]

לוגים ברמת השירות

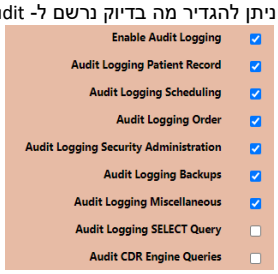
- כל השירותים שיהיו בשימוש מספקים לוגים מקומיים
- שירות CloudWatch logs יאסוף את הלוגים הללו ויאפשר תצוגה מרוכזת שלהם
  - יבוצע איסוף גם של VPC Flow logs באותה צורה
- גישה ללוגים תתאפשר בכפוף להגדרות בטבלת "חלוקת תפקידים בניהול שירות הענן" [פרט-iii.E.3]

לוגים אפליקטיביים

- לוגים אפליקטיביים ברמת Apache יאספו מרמת ה- Pod ל- AWS CloudTrail.
- במידה ויידרש מעקב מעמיק יותר אחר פעילות המערכת, ניתן לשלב איסוף נתונים לתוך CloudWatch באמצעות AWS SDK for PHP.
- גישה ללוגים אלו מתוך המערכת אפשרית רק למשתמש בתפקיד ניהולי
- הלוגים לא יכללו או יציגו מידע אישי או מידע רגיש אחר בצורה בלתי מוצפנת [פרט-4.1]

לוג מעקב (AUDIT) [פרט-2/3.C.1]

- המערכת אוגרת מידע מעקב כברירת מחדל למסד הנתונים שם הם נשמרים ללא הגבלת זמן
  - ניתן להגדיר מה בדיוק נרשם ל- Audit המערכות:



שינויי הגדרה של מנגנון ה- Audit נרשמים ל- Audit [פרט-4.i.A.3]

  - ניתן, במידת הצורך, להוסיף Audit ברמת השירות (RDS for MariaDB) באמצעות Audit plugin
- עבור S3 מעקב ייושם באמצעות Server logs
- עבור פעילות בחשבון מעקב ייושם באמצעות CloudTrail

ניטור שירותים

מעקב אחר זמינות, אמינות וביצועי השירותים השונים יסופק על ידי שירות AWS CloudWatch.
יש לשקול שימוש ב- Prometheus לניטור מעמיק של רכיבים בתוך AWS EKS.

Commented [YBS4]: @אלונה – לבדוק האם ידוע לכמה זמן נדרש לשמור

ניטור אבטחת מידע

איסוף ומעקב אחר נתונים הקשורים לרמת אבטחת המידע של המערכת ינוהל על ידי שירות AWS Security Hub.
התראות מתאימות עבור אירועים חריגים יוגדרו על בסיס השילוב בין Security Hub ל- Cloud Watch.
במידה ונדרשות התראות גם ברמת הגנת DDoS, יהיה צורך לשדרג את שריות AWS Shield לרמת Advanced.

להלן רשימת אירועים חריגים שמומלץ להקים עבורם התראה [פרט-5.2]:

- כישלונות בכניסה ליישום עד כדי נעילת משתמש
- ניסיון תקיפה DDoS
- ניסיון גישה ברשת שנחסם
- תוספת משתמש ניהולי לסביבת הענן או ליישום
- גישה בלתי מורשת לאזורים שונים ביישום

תחזוקה

התקנת עדכונים [פרט-IV.D.3]

- עדכוני AWS EKS יבוצעו בהתאם למסמך Updating an Amazon EKS cluster Kubernetes version
- עדכון גרסת Major של מנוע MariaDB תבוצע ידנית
- עדכוני היישום עצמו יבוצעו בהתאם למימוש תהליכי אוטומציה כמפורט בסעיף הבא

אוטומציה

להלן פירוט תהליכי האוטומציה במערכת:

- Infrastructure-as-code
  - הקמת תשתיות הענן תבוצע תוך יישום Infrastructure-as-code כדי להקל על תחזוקה ושינויים בצורה אוטומטית ועם מינימום פעילות ידנית כאשר התשתית ליישום זה תהיה Terraform
- Build pipelines
  - בניית תוצרים ובעיקר Docker images תבוצע בתחילה ידנית על בסיס Scripts קיימים
  - בהמשך יש לשאוף לאוטומציה של התהליך על בסיס Jenkins (אך ניתן לשקול חלופות כדוגמת CodeBuild) שישלוף גרסאות מ- Github ויבצע Build
- Deploy pipelines
  - הפצת תוצרים ובעיקר ל- EKS תבוצע באמצעות AWS CodePipelines

ניהול מדיניות מרכזית

- ניתן לעשות שימוש ב- AWS Organizations לניהול מרכזי של מדיניות משתמשים, הרשאות ושירותים, ביחוד כאשר ארגון נתון עושה שימוש במספר חשבונות AWS. במידה ויעשה שימוש בריבוי חשבונות (למשל להפרדה בין סביבות), יוגדרו Service control policies כדי לתחם את הפעילות המותרת באותם חשבונות.
- ניתן לעשות ב- AWS Config לביצוע הערכה ומעקב של הגדרות סביבת ענן AWS [פרט-ii.E.3]

תמיכה

- מומלץ ללוות את המערכת בתמיכה בתשלום של AWS, לפחות ברמת Developer (עלות מינימום של 29$ בחודש) ובעדיפות לרמת Business (עלות מינימום של 100$ בחודש)