

2024

Revolutionary Cloud-Native
Network Architecture for Next-
Gen ISP Services: Design,
Implementation and Analysis

פרויקט מימוש וניתוח רשת
תקשורת מבוססת תוכנה
לספקי ISP

סתיו איזיק – Stav Aizik

תומר פיליפ – Tomer Philip

הפקולטה להנדסת חשמל, אלקטרוניקה ותקשורת (50076) – תשפ"ד

תוכן

1.	תקציר	4
1.1.	תיאור הבעיה, התוצרים, המסקנות ונושאי הספר	4
1.2.	ABSTRACT	5
1.3.	רשימת טבלאות	5
1.4.	רשימת סימנים וקיצורים	5
2.	פרק 1 : מבוא	9
2.1.	רקע כללי	9
2.2.	תיאור הבעיה	10
2.3.	הפתרון	11
2.4.	מבנה ספר הפרויקט	12
3.	פרק 2 : רקע תיאורטי	13
3.1.	מבוא לרשתות הנתונים של ספקי התקשורת (ISPs)	13
3.2.	וירטואליזציה של עולמות הרשת NFV	17
3.3.	שירותי רשת תוכנתיים מבוססי טכנולוגיית ענן CNF (Cloud-Native Network Functions)	18
3.4.	וירטואליזציה Virtualization	19
3.5.	תשתית הענן (Cloud infrastructure)	19
4.	פרק 3 : תוכן הפרויקט, הצגת האתגרים והפתרונות	25
4.1.	פתיחה	25
4.2.	ניתוח האתגרים המרכזיים איתם מתמודדים ספקי התקשורת בארכיטקטורות הרשת מסורתיות מבוססות חומרה	25
4.3.	ניתוח והשוואת החלופות להנגשת שירותי רשת על ידי ספקי התקשורת (ISP): מהשיטה המסורתית לגישה מבוססת וירטואליזציה (VNF) ועד לטכנולוגיה מבוססת לענן (CNF)	31
4.4.	תהליך בחירת התשתית המתאימה להצגת הנתבים והרכיבים התוכנתיים וביצוע הבדיקות בפרויקט	36
5.	פרק 4 : תכנון ומימוש הפרויקט	41
4.5.	תכנון ומימוש ארכיטקטורת רשת ISP מבוססת טכנולוגיות CNF הכוללות מדידות	42
4.6.	תכנון ומימוש התשתית העננית המארכת את הפרויקט	70
4.7.	שלב המימוש לתשתית העננית המארכת את הפרויקט	73

84.....	פרק 5 - מסקנות והצעות עבודה להמשך	.6
85.....	ביבליוגרפיה	.7

1. תקציר

1.1. תיאור הבעיה, התוצרים, המסקנות ונושאי הספר.

בפרויקט זה תוכננה ויושמה ארכיטקטורת רשת חדשנית ומבוזרת עבור ספקי שירותי אינטרנט (ISP), המתבססת על טכנולוגיות מודרניות של וירטואליזציה בענן, כגון NFV ו-CNF. הפרויקט נולד מתוך צורך מהותי של ספקי התקשורת להתאים את תשתיות הרשת שלהם לעידן הדיגיטלי המודרני, המאופיין בדרישות גבוהות של גמישות, סקאלביליות וזמינות. טכנולוגיות מסורתיות, המתבססות על חומרה ייעודית לכל שירות, הוכחו כמוגבלות ואינן מסוגלות לספק מענה לצרכים המשתנים של השוק לעולמות 5G וה-IoT ולעלייה בכמות המשתמשים. הפרויקט התמקד בפיתוח ארכיטקטורה מבוססת ענן אשר תאפשר ל-ISP לספק ללקוחותיהם שירותי רשת בעזרתה הוכחנו את בשלות הטכנולוגיה על ידי מימוש פרוטוקולי ניתוב מתקדמים (כגון IS-IS, GRE, MPLS) . היכולת להשתמש בטכנולוגיות כמו VRF ו-MPLS VPN אפשרה לנו לספק הפרדה לוגית מלאה בין לקוחות וישומים, תוך שמירה על אבטחת מידע גבוהה. באמצעות השימוש בקונטיינרים ותשתיות Cloud-Native, התאפשרה גמישות יוצאת דופן בפריסת השירותים, עם יכולת להרחיב את השירותים או לצמצמם במהירות ובאופן אוטומטי בהתאם לעומסים ולצרכים. זאת בנוסף ליתרונות של ניתוק החומרה והתוכנה ביניהם Vendor Lock-in/הארכיטקטורה שהוקמה מתפרסת על גבי סביבת ענן ציבורית של Amazon AWS, בעזרתה הקמנו שני אתרים של ספק תקשורת בארה"ב המחוברים ביניהם בארכיטקטורה גמישה ואמינה. במהלך הפרויקט, נבחנו ביצועי הרשת באמצעות מדידות RTT, Jitter ו-Throughput, והתקבלו תוצאות מרשימות שהעידו על חיבור יציב ואיכותי בין אתרים מרוחקים גיאוגרפית, עם זמני תגובה נמוכים במיוחד, קצבי תעבורה מרשימים ואפס אחוזי אובדן מנות. המסקנה היא שטכנולוגיית CNF מציעה לספקי התקשורת פתרון מהפכני, המספק יכולת הסתגלות לשינויים מהירים בשוק, צמצום עלויות תפעוליות, ויכולת לספק שירותים מותאמים אישית ברמה גבוהה. תשתית זו היא אבן דרך לעבר רשתות תקשורת אוטונומיות, המנוהלות בצורה חכמה באמצעות שילוב טכנולוגיות AI, ומציעה עתיד מבטיח לספקי התקשורת בעידן ה-IoT וה-5G.

1.2. ABSTRACT

In this project, we designed and implemented an innovative and distributed network architecture tailored for Internet Service Providers (ISPs), utilizing modern cloud-based virtualization technologies such as NFV and CNF. This project emerged from a fundamental need for ISPs to adapt their network infrastructures to the demands of the digital age, which is characterized by high requirements for flexibility, scalability, and availability.

Traditional technologies, which rely on dedicated hardware for each service, have proven limited and incapable of meeting the evolving market demands driven by 5G, IoT, and the increase in user traffic. This project focused on developing a cloud-native architecture enabling ISPs to deliver network services with unprecedented flexibility and efficiency. We demonstrated the maturity of CNF technology by implementing advanced routing protocols (such as IS-IS, GRE, and MPLS). Furthermore, leveraging technologies like VRF and MPLS VPN allowed us to ensure complete logical isolation between clients and applications while maintaining a high level of information security.

By utilizing containers and cloud-native infrastructures, we achieved exceptional flexibility in service deployment, with the ability to expand or reduce services quickly and automatically based on load and demand. This also brought the advantage of decoupling hardware and software, effectively addressing the challenge of Vendor Lock-in. The established architecture was deployed on Amazon AWS's public cloud environment, through which we set up two ISP sites in the United States, interconnected with a resilient and flexible architecture.

Throughout the project, we evaluated network performance through measurements of RTT, Jitter, and Throughput, yielding impressive results that indicated a stable and high-quality connection between geographically distant sites, with extremely low latency, high throughput rates, and zero packet loss.

The conclusion is that CNF technology offers a revolutionary solution for ISPs the ability to adapt rapidly to market changes, reduce operational costs, and providing a significant represents deliver highly customized services. This infrastructure milestone towards autonomous communication networks that are intelligently managed through AI technologies, presenting a promising future for ISPs in the IoT and 5G era.

1.3. רשימת טבלאות

מספר טבלה	תיאור	עמוד
טבלה מס' 1	טבלת סימנים וקיצורים	
טבלה מס' 2	השוואה בין טכנולוגיות	

1.4. רשימת סימנים וקיצורים

קיצור	הסבר
-------	------

ספק שירותי אינטרנט המספק קישוריות לאינטרנט ולרשתות תקשורת פרטיות עבור ארגונים ויחידים	ISP (Internet Service Provider)
יצרנית ציוד תקשורת מהמובילות בעולם, המספקת פתרונות לרשתות ארגוניות, ספקי שירות ופרויקטים טכנולוגיים. מוצריה כוללים נתבים, מתגים, מערכות ניהול רשת, פתרונות אבטחת סייבר ועוד	Cisco
רכיב רשת המשמש לניהול תעבורת נתונים בין מכשירים ברשת מקומית (LAN) המתג מחבר בין מכשירים כמו מחשבים ושרתים ומעביר את חבילות הנתונים למכשיר הנכון בהתאם לכתובת ה-MAC של כל מכשיר	מתג
רכיב רשת האחראי לניתוב חבילות נתונים בין רשתות שונות, באמצעות פרוטוקולי ניתוב מתקדמים כמו BGP ו-OSPF	נתב
נתב וירטואלי, המותאם לסביבות ענן וקונטיינרים, ומאפשר פריסה גמישה של יכולות ניתוב ואבטחת רשת מתקדמות בסביבות וירטואליות ובענן, תוך שמירה על ביצועים גבוהים.	Xrd
טכנולוגיה לניתוב תעבורה ברשתות תקשורת מבוססת תיוג חבילות נתונים (Labels) משמשת ליעול ניתוב התעבורה ולשיפור ביצועי הרשת.	MPLS (Multiprotocol Label Switching)
טכנולוגיה המאפשרת ניתוב והפרדה לוגית בין רשתות שונות על גבי אותה תשתית פיזית, תוך יצירת טבלאות ניתוב נפרדות לכל לקוח.	VRF (Virtual Routing and Forwarding)
פרוטוקול ניתוב חיצוני המשמש לניתוב תעבורה בין רשתות שונות או בין ארגונים שונים על גבי רשת האינטרנט.	BGP (Border Gateway Protocol)
פרוטוקול ניתוב פנימי דינמי המשמש להפצת נתבים בתוך ארגון, מבוסס על בחירת הנתיב הקצר ביותר.	OSPF (Open Shortest Path First)
מדד לזמן הנדרש לחבילה לצאת ממקור מסוים, להגיע ליעד ולחזור. ערך נמוך של RTT מצביע על ביצועים טובים של רשת.	RTT (Round Trip Time)
שיעור השגיאות ברמת הביטים. משמש למדידת איכות ואמינות התקשורת ברשת. ככל שה-BER נמוך יותר, כך איכות התקשורת גבוהה יותר.	BER (Bit Error Rate)

ממד לכמות הנתונים שניתן להעביר דרך מערכת תקשורת בפרק זמן נתון, בדרך כלל נמדד בביטים לשנייה.(bps)	Throughput (קצב העברה)
זמן העיכוב הכולל שלוקח לחבילת נתונים לנוע ממקור ליעד. השיהוי נמדד ב- msec ביישומים רגישים לזמן כמו VoIP ו-וידאו.	Latency (שיהוי)
פלטפורמה וירטואלית לסימולציה ואמולציה של רשתות. משמשת לאימון ולבדיקות של טופולוגיות רשת מורכבות בצורה וירטואלית לפני מימוש פיזי.	EVE-NG (Emulated Virtual Environment)
נתב הנמצא בקצה רשת הספק ומחבר בין רשתות הלקוחות לרשת הליבה של ספק השירות. אחראי על ניתוב תעבורה לנתבים של ספקים אחרים או ללקוחות הקצה.	PE (Provider Edge Router)
נתב הנמצא בקצה רשת הלקוח ומחבר את רשת הלקוח לרשת הספק. נתבים אלו מנהלים את התעבורה בתוך רשתות הלקוחות.	CE (Customer Edge Router)
נתב מרכזי ברשת הספק המספק ניתוב מהיר ויעיל של תעבורה ברשת	Core Router (נתב ליבה)
גישה לניהול רשתות תקשורת המבוססת על ניתוק השליטה ברשת מהחומרה והעברתה לשכבת תוכנה. מאפשרת גמישות רבה יותר בניהול ניתוב תעבורה.	SDN (Software-Defined Networking)
וירטואליזציה של פונקציות רשת, המאפשרת הפעלת פונקציות רשת כמו ניתוב, חומות אש והצפנה על חומרה כללית ולא על חומרה ייעודית.	NFV (Network Function Virtualization)
כלי לדימוי תנאים רשתיים כמו השהיה, אובדן חבילות, ושינויים בקצב העברה. משמש לבדיקת ביצועי רשת בתנאים מגוונים.	NETem
מנגנון לזיהוי שגיאות בחבילת נתונים. ה-CRC- מאפשר בדיקה האם חבילה הגיעה ליעדה בצורה תקינה או שנגרמה שגיאה במהלך השידור.	CRC (Cyclic Redundancy Check)
מצב שבו חלק מהחבילות שנשלחו ברשת אינן מגיעות ליעדן. מצב זה משפיע על ביצועי הרשת ואיכות התקשורת.	Packet Loss (אובדן חבילות)

מדד לשינויים בזיהוי ברשת, Jitter מצביע על חוסר יציבות ברשת ועלול להשפיע לרעה על יישומים הדורשים רציפות בתקשורת.	Jitter (השתנות השיהוי)
הגודל המרבי של חבילת נתונים שניתן להעביר בפרוטוקול רשת, MTU נמוך עלול לגרום לפיצול חבילות נתונים.	MTU (Maximum Transmission Unit)
טכנולוגיה לניהול תעבורה ברשת המבטיחה שרמת השירות הנדרשת ניתנת ליישומים קריטיים על ידי הקצאת רוחב פס, ניתוב עדיפויות לתעבורה, והבטחת זמני תגובה נמוכים. QoS מאפשרת לרשת לתת עדיפות לתעבורה כמו וידאו, קול ויישומים קריטיים בזמן אמת, על פני תעבורה פחות קריטית כמו גלישה באינטרנט או העברת קבצים, ובכך לשפר את ביצועי הרשת ולמנוע עיכובים ואובדן חבילות במקרים של עומסים.	QoS (Quality of Service)
POP, או Point of Presence (נקודת נוכחות), הוא מיקום גיאוגרפי שבו ספק שירותי אינטרנט (ISP) או רשת תקשורת אחרת מחזיקים ציוד, כמו שרתים ונתבים, כדי לספק שירותי חיבור ורשת ללקוחות באזור מסוים. המונח מתייחס למרכז פיזי המאפשר חיבור מקומי לרשת רחבה יותר, וכולל רכיבי רשת ותשתיות המספקות שירותי אינטרנט, נתונים, או קול באותו אזור.	POP
Command-Line Interface (ממשק שורת פקודה), הוא ממשק למשתמש המאפשר אינטראקציה עם מערכת ההפעלה או עם תוכנה מסוימת באמצעות הקלדת פקודות טקסט. במקום להשתמש בממשק גרפי הכולל כפתורים ותפריטים, המשתמש מזין פקודות בשורת פקודה ומקבל תגובה מהמערכת בטקסט.	CLI

טבלה מס' 2

2. פרק 1 : מבוא

2.1. רקע כללי

במהלך השנים האחרונות, חלה התקדמות משמעותית בעולם הטכנולוגי ובטכנולוגיות התקשורת, מה שהביא לעלייה דרמטית בצרכים ובדרישות מהמערכות השונות.

בעבר השימושים הטכנולוגים היו פחות מורכבים ושימשו בעיקר עבור חיבור בין מחשבים ושירותים בתוך משרדים ומתקנים ארגוניים, התקשורת נשענה על תשתיות חומרה פיזיות, הדרישות הטכנולוגיות כללו בעיקרן שליחת דואר אלקטרוני, גישה למידע בסיסי, שיתוף קבצים ורוב התקשורת הארגונית נשענה על בסיס תשתיות מקומיות.

לעומת זאת, בעידן הנוכחי הטכנולוגיה מתפתחת במהירות ומשמשת לשירותים נרחבים ומורכבים יותר הכוללים גישה ליישומים וקבצים בעננים, שירותי IoT – מכשירים רבים כמו חיישנים, מצלמות, רכבים חכמים ועוד רכיבי קצה רבים שמחוברים לרשתות, שירותי וידאו, משחקי אונליין ושירותי AI למשתמשים.

שירותים אלו שפירטנו ושירותים מורכבים נוספים דורשים חיבור מהיר ואמין, ורוחבי פס גבוהים כדי להבטיח גישה מיידית לנתונים מכל מקום בעולם. התקדמות זו דורשת ניהול רוחבי פס בצורה אופטימלית, ושימוש בטכנולוגיות חדשות בכדי להבטיח חווית משתמש חלקה ואמינה, נושא שהפך להיות קריטי ביותר בתקופה זו.

ספקי שירותי האינטרנט ISPs אחראים לחבר לקוחות פרטיים ועסקיים לרשת האינטרנט העולמית ולספק להם מגוון שירותי תקשורת כמו קישוריות IP, קישוריות אופטית מבוססת סיבים, תקשורת אלחוטית וסלולר בטכנולוגיות שונות.

כיום, רוב ספקי התקשורת מסתמכים על פתרונות חומרה מסורתיים לניהול ותפעול הרשתות שלהם, כמו נתבים, מתגים ומוצרים אופטיים.

בנוסף, בעידן שבו מתקפות סייבר מתוחכמות הולכות ומתגברות, על ספקי התקשורת להבטיח שהרשתות שהם מספקים עמידות בפני איומים חיצוניים ויכולות להתמודד עם ניסיונות חדירה ושיבוש השירותים. הדרישות לאבטחת מידע אינן מסתכמות רק בהגנה מפני מתקפות אלא גם ביכולת להבטיח את המשך תפקוד הרשת תחת עומסי תעבורה כבדים, במיוחד לאור הגידול בשימוש ביישומים עתירי משאבים.

2.2. תיאור הבעיה

הבעיה שהפרויקט מנסה לפתור היא הקושי של ספקי שירותי אינטרנט (ISP) לספק תשתיות רשת גמישות, מהירות ויעילות, העומדות בדרישות המודרניות של סקילביליות ועמידה ברמת שירות (SLA) גבוהה. הטכנולוגיות המסורתיות, המבוססות על חומרה ייעודית (Appliance), יוצרות זמני אספקה ארוכים, עלויות תחזוקה גבוהות, ותלות בספקים בודדים (Vendor Lock-In). עם ההתקדמות הטכנולוגית ספקי האינטרנט נדרשים להתמודד מול האתגרים הבאים :

חוסר גמישות והתאמה לסביבות מודרניות :

תשתיות רשת מבוססות חומרה פיזית מוגבלות בגמישותן , כל שינוי בתצורת הרשת – כמו הוספת רכיבים חדשים שדרוגים והתאמות ברשת דורשת עבודה פיזית באתר ההתקדמות הטכנולוגית לסביבות ענן מחייבת גמישות ותאימות אוטומטית לשינויים , תשתית מבוססת חומרה אינה מתאימה לניהול והרחבה מהירים בהתאם לביקוש , דבר שגורם לעיכובים .

TIME TO MARKET (TTM)

ברשתות ISP מסורתיות הוא זמני אספקת שירותים חדשים ללקוח ארוכים ולא עומדות בקצבי השינוי המהירים בשוק. במערכות המבוססות על חומרה ייעודית, תהליך רכישת, פריסת והתקנת החומרה דורש זמן רב, בשל הצורך במלאי רכיבים פיזיים, משלוח, התקנה וקונפיגורציה. תלות בתהליך הרכש, בשרשראות אספקה ובכוח אדם מובילה לעיכובים משמעותיים ביכולת של הספק להרחיב תשתיות, להוסיף שירותים חדשים או לשדרג שירותים קיימים.

• ניהול ומעקב לוגיסטי

תחזוקת תשתיות רשת המבוססות על חומרה ייעודית מצריכה ניהול מלאי מורכב של חלקי חילוף וציוד לוגיסטי נוסף. בכדי להבטיח שרידות וגיבוי, יש צורך ברכיבים ייחודיים ומודולים ספציפיים שמספקים הוונדורים השונים, שמטרתם לקשר בין רכיבי הרשת. כל רכיב חומרה דורש אחסון, תחזוקה, ומעקב מתמיד אחר תהליכים כמו עדכונים, החלפות ותיקונים, מה שמעלה את רמת המורכבות הלוגיסטית והעלות התפעולית של הספק.

• עלויות גבוהות ותחזוקה מורכבת

רכישת ציוד רשת פיזי, תחזוקתו ושדרוגו כרוכים בעלויות כבדות. במקרים רבים, במיוחד בסביבות גדולות, נדרשת השקעה משמעותית בציוד פיזי ייעודי, בשטח ריצפה רב, קירור ובכוח אדם לתחזוקה.

בנוסף לכך התשתיות המסורתיות דורשות תחזוקה יקרה ומורכבת הכוללת תיקונים, שדרוגים והחלפת רכיבים כאשר הם מגיעים למצב של End of Life (EOL) או End of Support (EOS). תהליכים אלו מצריכים הגעה פיזית לאתרי הלקוחות להתקנה, קונפיגורציה ראשונית, ושדרוגים שוטפים, מה שמוביל לעלויות תפעול גבוהות ולסיכון מוגבר לשגיאות אנוש. בנוסף, יש צורך לבצע תכנון מראש של תהליך השדרוג והחלפת רכיבים ישנים, הכולל סילוק החומרה הישנה, דבר שמגביר את המורכבות הלוגיסטית והתפעולית.

• צריכה וניהול משאבים

במערכות מסורתיות, כל שירות דורש מכונה פיזית ייעודית, מה שמוביל לצריכת משאבים גבוהה במיוחד. ריבוי המכונות הפיזיות מצריך שטח רב בארונות תקשורת, כמו גם משאבי חשמל וקירור משמעותיים לתמיכה בפעולתן התקינה. כתוצאה מכך, עלויות התחזוקה והאחזקה עולות באופן משמעותי, בנוסף, השימוש הרב במשאבי חשמל וקירור לא רק מייקר את התפעול, אלא גם מגביר את ההשפעה השלילית על הסביבה בשל צריכה אנרגטית גבוהה ולא יעילה.

מדובר באתגרים משמעותיים, שכן יש להבטיח שתעבורת הנתונים תתנהל בצורה יעילה ומאובטחת, תוך מינימום עיכובים ומקסימום ביצועים. יש לוודא שהרשת מסוגלת להתמודד עם עומסים כבדים מבלי לפגוע ברציפות השירות ובביצועים, במיוחד ביישומים קריטיים הדורשים מענה בזמן אמת. לבסוף, יש להטמיע מנגנוני אבטחה מתקדמים שיגנו על תעבורת המידע וימנעו איומים חיצוניים וזליגת מידע בין הרשתות השונות.

2.3. הפתרון

בעידן המודרני קיימות טכנולוגיות מתקדמות, כמו NFV ו-CNF שיכולות לספק פתרונות לאתגרים הללו ולהקל על סוגיות התפעול והתחזוקה בארכיטקטורות IT של ספקי תקשורת. טכנולוגיות אלה מאפשרות מעבר מחומרה ייעודית (appliance) לפתרונות וירטואליים וענניים, ומציעות גמישות רבה יותר בניהול הרשת, סקלביליות גבוהה ויכולת פריסה מהירה של שירותים חדשים. בפרויקט זה נבחן את היכולת של טכנולוגיות NFV ו-CNF לספק מענה לאותם אתגרים, ונבדוק האם הן יעילות כמו רכיבי החומרה הייעודיים במונחים של ביצועים, תחזוקה ועלויות תפעול.

מטרת הפרויקט היא להוכיח את היכולת ליישם טכנולוגיות רשת מבוססות תוכנה עבור ספקי שירותי אינטרנט (ISP) באמצעות שימוש ב-SDN ו-NFV. בפרויקט זה, נשים דגש על יישום רכיבי תקשורת תוכנותיים, בפרט נתבים אפליקטיביים המנהלים ומופעלים על גבי תשתית קונטיינרים בטכנולוגיות מתקדמות כמו CNF (Cloud Native Network Functions) או באמצעות מכונות וירטואליות (VM).

הפרויקט ישאף לבחון פרמטרים כמו:

- זמן פריסת שירותי התקשורת
- יכולת תפעול, ניהול ושדרוג רכיבים
- אוטומציה של תהליכים, עמידות בפני עומסי תעבורה ואבטחת מידע.
- חקר ביצועים ע"י מדידות של פרמטרים חשובים כמו latency, throughput, packet loss, ו-jitter ונבחן את יכולת המערכת להתמודד עם מתקפות סייבר ודרישות אבטחת מידע.

כאשר שאלת המחקר שלנו הינה :

האם ניתן להקים רשת תקשורת ISP מתקדמת ויעילה, המספקת גמישות, סקלבליות ואבטחת מידע גבוהה, באמצעות טכנולוגיות-קצה מתקדמות מבוססות תוכנה כמו CNF ו-NFV?

תוך השגת ביצועים מיטביים, שיפור בתחזוקה והפחתת עלויות בהשוואה לתשתיות מסורתיות מבוססות חומרה.

האם ניתן להקים רשת יעילה ומתקדמת המספקת גמישות, סקלבליות ואבטחת מידע בעזרת גישה מהפכנית של ומתקדמת טכנולוגיות רשת מבוססות תוכנה יעילות כמו רכיבי רשת יעודים , בהתייחסות לביצועים , תחזוקה ועלויות.

2.4. מבנה ספר הפרויקט

המסמך מחולק למספר פרקים עיקריים שיתארו את כל שלבי הפרויקט :

- תקציר
- Abstract
- רשימות וטבלאות
- פרק 1 : מבוא
- פרק 2 : רקע תאורטי
- פרק 3 : תוכן הפרויקט , הצגת אתגרים , פתרונות ובחינת חלופות
- פרק 4 : תכנון ומימוש
- פרק 5 : מסקנות והצעות להמשך
- פרק 6 : ביבליוגרפיה ונספחים

3. פרק 2 : רקע תיאורטי

3.1. מבוא לרשתות הנתונים של ספקי התקשורת (ISPs)

רשת (Internet Service Provider) ISP היא תשתית המאפשרת לספק שירותי אינטרנט ולחבר לקוחות לאינטרנט או לרשתות פרטיות. ספקי שירותים אלה אחראים לניהול תעבורת נתונים בין הלקוחות לבין ספקי שירות אחרים, תחנות קצה ורשתות פנימיות, תוך שימוש בטכנולוגיות ופרוטוקולי ניתוב שונים לניהול יעיל ואמין של התעבורה. ספקי תקשורת מסורתיים מפעילים ארכיטקטורת רשת נתונים מורכבת, המשרתת מגוון רחב של לקוחות על בסיס נתבים ופרוטוקולי ניתוב כמו OSPF ו-ISIS. רשת זו מתבססת על ניתוב מנות בין נתבים על פי פרוטוקולי ניתוב, ובעזרת מימוש טכנולוגיות כמו VRF, MPLS (Multiprotocol Label Switching) וקישורי VPN.

3.1.1. פרוטוקולי TCP/IP הבסיס לרשת התקשורת

בבסיס כל רשת תקשורת מודרנית עומדים שני פרוטוקולים מרכזיים: IP (Internet Protocol) ו-TCP (Transmission Control Protocol). שני הפרוטוקולים הללו, המרכיבים את חבילת הפרוטוקולים הידועה בשם TCP/IP, משמשים כעקרונות הבסיס של העברת מידע ברשתות תקשורת, וממלאים תפקיד חיוני בארכיטקטורת ספקי שירותי אינטרנט (ISP).

פרוטוקול IP (Internet Protocol)

IP הוא פרוטוקול הניתוב המרכזי שמאפשר העברת מנות מידע (packets) ברשתות תקשורת שונות, ומהווה את הבסיס לפעולתו של האינטרנט. תפקידו של IP הוא להגדיר כתובות ייחודיות לכל התקן ברשת ולנהל את הניתוב של מנות המידע בין מכשירים שונים, כך שיגיעו ליעדן באופן תקין.

ספקי שירותי אינטרנט (ISP) משתמשים בפרוטוקולי IP ו-TCP כדי לספק שירותי אינטרנט ללקוחותיהם. פרוטוקול IP ממלא תפקיד מרכזי בהקצאת כתובות IP ללקוחות, בניתוב התעבורה בין הרשתות השונות, ובניהול הקישוריות בין הלקוחות לאינטרנט. כתובות ה-IP שמוקצות ללקוחות ה-ISP מאפשרות להם להתחבר לאינטרנט ולתקשר עם מערכות אחרות בכל רחבי העולם.

כתובת IP היא מזהה ייחודי לכל מכשיר או תחנה ברשת. כיום קיימות שתי גרסאות של IP:

IPv4: משתמש במערך של 32 ביטים לכתובת ומאפשר כ-4.3 מיליארד כתובות שונות.

IPv6: פותח כדי לתת מענה למחסור בכתובות, ומשתמש במערך של 128 ביטים, מה שמאפשר מספר עצום של כתובות.

מנגנון הניתוב: פרוטוקול IP פועל על פי עקרון של ניתוב מנות מידע (packet switching), שבו המידע המחולק לחבילות קטנות משודר ברשת ונע דרך נתבים (routers) עד שהוא מגיע ליעדו. לכל חבילת מידע מצורפת כתובת IP של היעד וכתובת ה-IP של השולח. הנתבים לאורך הדרך מנתבים את המנות לפי המידע שנמצא בראש החבילה, תוך שימוש בפרוטוקולי ניתוב כגון OSPF ו-BGP כדי לקבוע את המסלול הטוב ביותר.

פרוטוקול TCP (Transmission Control Protocol)

TCP הוא פרוטוקול תקשורת אמין האחראי לניהול קשרי התקשורת בין המחשבים ברשת ולהבטחת העברת נתונים שלמה ובסדר הנכון. פועל בשכבת התעבורה (transport layer) של מודל ה-OSI והוא מסייע במקרים בהם נדרשת תקשורת אמינה, כגון גלישה באינטרנט, העברת קבצים, דואר אלקטרוני וכדומה.

מנגנון Retransmissions: כאשר TCP מזהה שאחת המנות לא התקבלה בהצלחה או לא אושרה בזמן על ידי הנמען, הוא מבצע שידור חוזר של אותה מנה.

מנגנון Congestion Window (Cwnd): זהו פרמטר ש-TCP משתמש בו כדי לשלוט בכמות הנתונים שניתן לשלוח לפני קבלת אישור מהצד השני. גודל ה-Cwnd משתנה בהתאם לתנאי הוא גדל כאשר הקישור יציב וקטן במקרה של עומס יתר או איבוד מנות.

פרוטוקול UDP (User Datagram Protocol)

הוא פרוטוקול תקשורת connection less בשכבת התעבורה, המתמקד בהעברת נתונים מהירה וללא בקרת שגיאות או הבטחת סדר הגעת המידע. בשונה מ-TCP, UDP אינו מבצע בקרת אמינות ואינו דורש הקמת חיבור, דבר המאפשר שידור נתונים בתצורת datagrams ללא אישור הגעה. מאפיינים אלו הופכים את UDP לפרוטוקול מהיר ויעיל מבחינת עומס רשת, אך לא אמין בהגעה ליעד. הפרוטוקול מתאים במיוחד ליישומים רגישים לזמן כמו שידורי וידאו, קול (VoIP) ומשחקים מקוונים, בהם מהירות התגובה קריטית.

3.1.2 פרוטוקולים וטכנולוגיות בארכיטקטורת הרשת

IS-IS (Intermediate System to Intermediate System)

הוא פרוטוקול ניתוב פנימי (IGP) המיועד לניתוב בתוך תחומים (Domains) ולא בין מערכות אוטונומיות (AS). הפרוטוקול מוגדר על פי תקן ISO ומותאם לתמוך גם ב-IP. מבוסס Link State ובונה טופולוגיה מלאה של הרשת תוך שימוש באלגוריתם Dijkstra לחישוב המסלולים הקצרים ביותר בין צמתי הרשת.

מנגנון פעולה: IS-IS פועל בשכבה 3 של מודל OSI ומבצע חילופי מידע על מצב הקישורים (LSAs) בין נתבים, כך שכל נתב מחזיק עותק מעודכן של טופולוגיית הרשת. הוא משתמש במבנה היררכי עם רמות (Levels). קיימות בו שתי רמות ניתוב Level1 לניתוב בתוך אזור מקומי בלבד ו-Level2 לניתוב בין אזורים שונים, עם טופולוגיה של כל הרשת מה שייחודי לאזור ה-Backbone כאשר המטרה בחלוקה היא לשפר ביצועים. בפרויקט שלנו, IS-IS מוגדר לפעול ב-Level 2 בלבד, מה שמאפשר ניתוב מבוזר בין אזורים באופן יעיל.

יישום בפרויקט: בארכיטקטורת הפרויקט הקמנו IS-IS בין הנתבים XRd-1 ו-XRd-2 כחלק מהטופולוגיה הפנימית של הרשת. הבחירה ב-IS-IS נובעת מגמישותו בניהול תעבורה בסביבות מבוזרות, מניעת לולאות, והתאוששות מהירה במצבי שינוי. IS-IS פועל בשילוב עם GRE ליצירת מנהרות מאובטחות VPNs ותמיכה בחלוקת עומסים ושרידות בין הנתבים.

GRE (Generic Routing Encapsulation)

GRE הוא פרוטוקול עטיפה (Encapsulation) המאפשר יצירת מנהרות וירטואליות להעברת תעבורה מרובת פרוטוקולים על גבי רשתות IP. הוא פותח על ידי Cisco וכיום הוא נמצא בשימוש נרחב ליצירת VPNs קישוריות וירטואליות בין רשתות מרוחקות על גבי רשת האינטרנט או רשתות IP פרטיות. פרוטוקול GRE מאפשר יצירת "מנהרה" שבה עוטפים את המידע בתצורה שמאפשרת מעבר בטוח על גבי הרשת, כאשר הנתונים עצמם נשמרים בתצורה המקורית שלהם.

מנגנון פעולה: כל חבילת מידע מועברת כשהיא עוטפת את המידע המקורי במעטפת GRE, המכילה את כתובת המקור והיעד החדשה עבור המנהרה. באופן זה, התעבורה הופכת לשקופה (Transparent) עבור הרשת שמעבירה אותה, כך שכתובות ה-IP המקוריות מוסתרות במהלך ההעברה. כיוון ש-GRE הוא פרוטוקול כללי, ניתן להשתמש בו להעברת מגוון סוגי פרוטוקולים על גבי רשתות IP.

מימוש בפרויקט: ה-GRE משמש ליצירת שני מנהרות וירטואליות בין הנתבים XRd-1 באזור Alpha ו-XRd-2 באזור Beta. כל מנהרה מייצגת צינור תקשורת וירטואלי נפרד, ומאפשרת העברת נתונים בין האזורים בצורה שקופה ומאובטחת מבלי שהרשת הפיזית החיצונית של הספק תתערב או תפריע לרשת הלקוח. מאפשר גם חלוקת עומסים בין הנתבים, מה שמאפשר ניצול גמיש ויעיל יותר של משאבי הרשת.

בפרויקט שלנו, השילוב בין GRE ו-IS-IS נבחר כדי ליצור ארכיטקטורה מאובטחת, גמישה ואופטימלית לניהול תעבורה בין שני אזורים הרשת – Alpha ו-Beta (מפורט בהמשך) מאפשר יצירת שתי מנהרות וירטואליות בין הנתבים XRd-1 ו-XRd-2, שמספקות יתירות, חלוקת עומסים, וגיבוי אוטומטי במקרה של כשל באחת המנהרות. כל מנהרה יכולה לתמוך בעומסים או לגבות את השנייה בעת הצורך בזכות פרוטוקול IS-IS המנהל את הניתוב באופן דינמי, מתעדכן בזמן אמת על מצבי הקישוריות במנהרות, ובוחר את הניתוב האופטימלי עבור כל חבילה, מה שמבטיח ביצועים מקסימליים ותגובה מהירה לשינויים בתעבורה.

תפקיד ה-MPLS (Multiprotocol Label Switching) ברשתות תקשורת

פרוטוקול לניתוב מהיר שמאפשר להאיץ את תהליך העברת הנתונים ברשתות באמצעות מנגנון של תיוג (Labeling). בניגוד לניתוב מבוסס IP, שבו כל נתב בודק את כתובת ה-IP בשכבה ה-3 ומחליט לאיזה יעד לשלוח את החבילה, ב-MPLS החבילות מקבלות תג (Label) כבר בתחילת דרכן, והנתבים משתמשים בתג הזה כדי להעביר את החבילה במהירות ליעדה. MPLS מאפשר קיצור זמנים בתהליך הניתוב, מפחית עומסים על הנתבים, ומייעל את הניתוב בתשתיות מורכבות.

MPLS משמש בעיקר ברשתות גדולות של ספקי שירותי תקשורת (ISP) ובארגונים גדולים, ומספק יתרון חשוב של ניתוב מבוסס איכות שירות (QoS), מה שמאפשר להגדיר עדיפויות שונות לחבילות בהתאם לסוג התעבורה, כמו וידאו או קבצים קריטיים. בנוסף, MPLS תומך בתצורות רשת מגוונות כמו VPN ו-VRF, המאפשרות הפרדה בין תעבורות שונות באותה תשתית פיזית, תוך שמירה על אבטחה ופרטיות.

מימוש בפרויקט: בפרויקט שלנו, MPLS ממומש כדי להבטיח ניתוב מהיר ואמין בין האזורים Alpha ו-Beta דרך הנתבים XRd-1 ו-XRd-2. השימוש ב-MPLS מספק לנו אפשרות ליצור נתבים מוגדרים מראש בין אזורי הרשת, מה שמיעיל את זמן העברת התעבורה ומפחית עומסים. במקביל, משמש בסיס לטכנולוגיות כמו VRF ו-L3VPN אותן יישמנו לאבטחת הלקוח.

מנגנון VRF (Virtual Routing and Forwarding) ברשתות תקשורת

VRF הוא מנגנון שנמצא בשימוש נרחב בסביבות רשת מבוססות MPLS, המאפשר לנתבים לנהל מספר טבלאות ניתוב מבודדות באותו נתב פיזי. באמצעות VRF, ניתן ליצור רשתות פרטיות וירטואליות (VPNs) עבור לקוחות שונים, מה שמספק הפרדה מלאה בין תעבורה שונות באותה תשתית, תוך שמירה על אבטחת המידע. כל VRF "מרגיש" כאילו הוא הלקוח היחיד בתשתית.

מימוש בפרויקט: בפרויקט שלנו, VRF ממלא תפקיד מרכזי בהפרדת התעבורה של הלקוחות לבין התעבורה של ספק התקשורת המכילה לקוחות נוספים. הקמנו את הרשת הפרטית בעזרת GRE ו-ISIS וחיברנו את הלקוחות באזורים Alpha ו-Beta של VRF ייעודי להם בשם "nfs". השימוש ב-VRF מאפשר לנו ליצור תצורות ניתוב מבודדות עבור כל אחד מהאזורים, תוך שמירה על אבטחת מידע גבוהה ופרטיות.

L3VPN (Layer 3 Virtual Private Network)

שירות ניתוב ברמת שכבה 3 המיועד לספק הפרדה וירטואלית בין רשתות של לקוחות שונים. טכנולוגיית L3VPN, המבוססת על ניתוב ב-VRF ומעבר על גבי MPLS, מאפשרת יצירת רשת פרטית מאובטחת המנוהלת על גבי אותה תשתית פיזית. באמצעות L3VPN ניתן להקים רשתות וירטואליות המופרדות אחת מהשנייה ובכך לשמור על פרטיות ולהגביר את אבטחת הרשת.

מימוש בפרויקט: L3VPN ממומש בעזרת "nfs" VRF כדי לתמוך בתעבורת לקוחות פרטית המועברת על גבי ה-GRE Tunnels המחברים בין XRd-1 ל-XRd-2. כל תעבורה המועברת ב-VPN מנותבת בתוך ה-VRF, ומאפשרת לרשת להישאר מבודדת ומאובטחת. בפרויקט זה, L3VPN מהווה שכבת אבטחה נוספת המגנה על התעבורה בין אזורי הרשת Alpha ו-Beta, כך שתהיה מבודדת, מוגנת ונגישה אך ורק לנקודות הקצה הרלוונטיות.

3.1.3. דוגמה לארכיטקטורת רשת של ספק תקשורת

רשתות של ספקי שירותי אינטרנט (ISP) בנויים בארכיטקטורה רב-שכבתית שמטרתה לספק קישוריות אמינה, ביצועים גבוהים ויכולת גמישות למשתמשי קצה ולארגונים גדולים.

שכבת הגישה (Access Layer) – שכבה זו מספקת חיבוריות ישירה ללקוחות דרך אמצעי גישה כמו סיבים אופטיים, Wi-Fi ועוד. מטרת שכבת הגישה היא לאפשר חיבור של משתמשים פרטיים ועסקיים לרשת ISP.

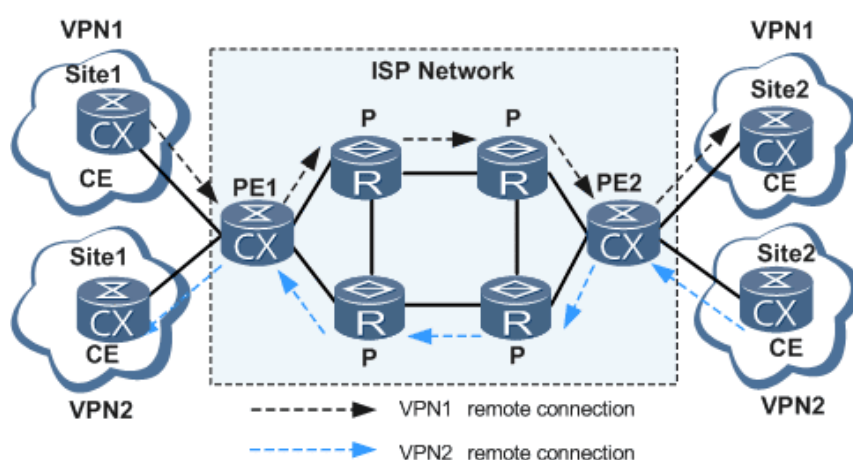
שכבת ההפצה (Distribution Layer) – שכבה זו מחברת בין שכבת הגישה לשכבת הליבה ומבצעת תפקידים כגון ניתוב, ניהול מדיניות, ואבטחת מידע. בשכבה זו נעשה שימוש בפרוטוקולי ניתוב פנימיים כמו IS-IS ובמבנים מבוזרים לניהול עומסים ולהפחתת השפעת תקלות על הרשת.

שכבת הליבה (Core Layer) – זו שכבת עמוד השדרה של ה-ISP, אשר נועדה לספק קישוריות מהירה ואמינה בין כל אזורי הרשת. בשכבה זו נעשה שימוש במיתוג וניתוב מהירים, ולעיתים קרובות בטכנולוגיות MPLS (Multi-Protocol Label Switching) כדי לנתב תעבורה באופן אופטימלי, תוך גמישות וחלוקת עומסים בין אזורים.

(Provider Router) P – נתבי ספק שממוקמים בתוך רשת ISP, אינם אחראים על חיבור ישיר ללקוחות אלא לניתוב פנימי של תעבורה בתוך הספק.

(Provider Edge Router) PE – נתבים הממוקמים בקצה רשת ה-ISP ומתחברים לרשתות הלקוחות ואחראים לניתוב תעבורת הלקוח אל רשת הספק.

(Customer Edge Router) CE – נתבי קצה הממוקמים ברשתות הלקוחות ומחברים את הרשתות של הלקוחות לרשת ה-ISP.



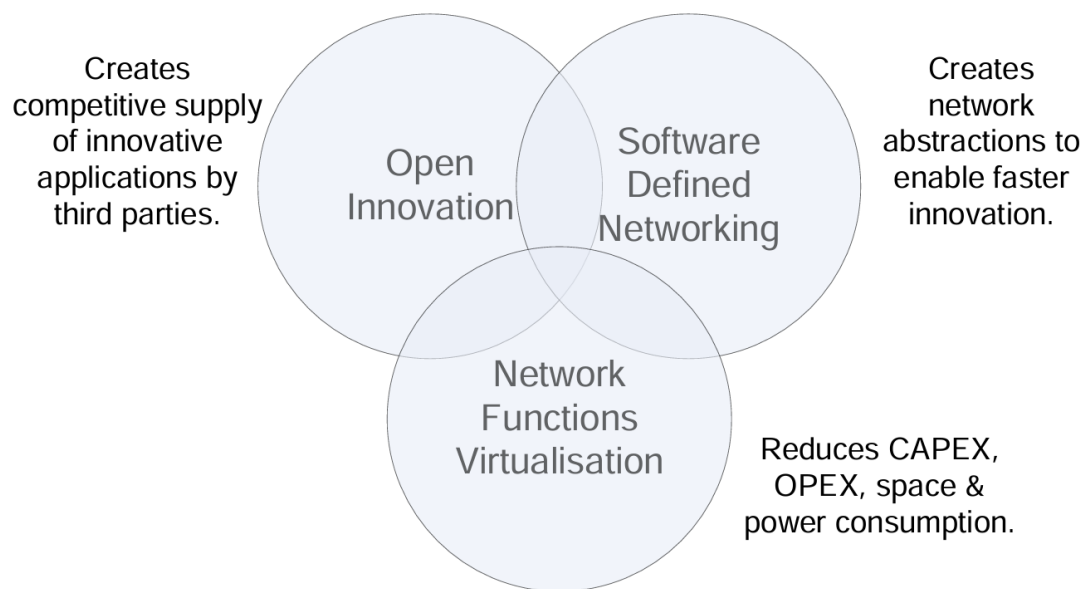
3.2. וירטואליזציה של עולמות הרשת NFV

3.2.1. גישת ה-NFV (Network Function Virtualization) וירטואליזציה של רשתות

וירטואליזציה של פונקציות רשת (NFV) היא גישה לניהול תשתיות רשת, שבה פונקציות רשת מסורתיות (כמו נתבים, חומות אש, מאזני עומסים, NAT, DNS ועוד) מיושמות בסביבה וירטואלית פתוחה ומבוססת תקנים, במקום על גבי חומרה ייעודית וספציפית ליצרן. הרעיון המרכזי של NFV הוא להפריד את פונקציות הרשת מחומרה ייעודית, ולהריץ אותן כרכיבי תוכנה על תשתיות רשת וירטואליות מלאות. כך, NFV עושה שימוש בטכנולוגיות וירטואליזציה סטנדרטיות לצורך אופטימיזציה וחדשנות בשירותי רשת, מה שמקל על ספקי שירותי תקשורת בניהול והפצת שירותים ברשתות שלהם.

באמצעות גישה זו, ספקי שירות יכולים להפחית משמעותית את עלויות ההון (CAPEX) ואת עלויות התפעול (OPEX) על ידי שימוש בחומרה כללית סטנדרטית (כמו שרתים ומתגים שאינם ייעודיים), במקום ציוד חומרה ייעודי ויקר. בנוסף, NFV מאפשרת פיתוח והפצה מהירים וגמישים יותר של שירותי רשת מבוססי תוכנה, מה שמקדם חדשנות בשוק. בנוסף משלים את הגישה של

ניתוב מוגדר תוכנה (SDN), אך ניתן ליישם כל אחד מהם בנפרד. בעוד ש-SDN מציע פשטות וניהול גמיש של משאבי רשת בשכבות הנמוכות (L2-L4) NFV מציע גמישות וניידות בשכבות הגבוהות יותר (L4-L7), ובכך מקדם את יכולת ההימנעות מתלות ביצרן ספציפי (vendor lock-in). בשילוב שתי הגישות, ניתן להפיק יתרונות נוספים בשירותי רשת עבור ספקי שירות, שכן הן מקלות זו על יישומה של זו ומשפרות את התועלת הכללית ברשתות תקשורת. לשם קידום מימוש ה-NFV, הוקמה קבוצת מפרט תעשייתי (ISG) בשם "Network Functions Virtualization" בחסות המכון האירופי לתקני תקשורת (ETSI).



3.3. שירותי רשת תוכנתיים מבוססי טכנולוגיית ענן CNF (Cloud-Native Network Functions)

הוא מונח המתאר יישום של פונקציות רשת (Network Functions) כקונטיינרים במקום כמכונות וירטואליות (VMs), בניגוד לגישה המסורתית של NFV (Network Function Virtualization). הרעיון מאחורי CNF הוא להפעיל רכיבי רשת – כמו נתבים, חומות אש, מאזני עומסים וכו' – כקונטיינרים בסביבה מבוססת ענן (כמו Kubernetes), ובכך להשיג יעילות גבוהה יותר וגמישות בניהול משאבים.

CNF התפתח כחלק מהמגמה לאמץ ארכיטקטורות ענן וטכנולוגיות קונטיינרים בתשתיות רשתות מודרניות. בעוד NFV מתבסס על וירטואליזציה מסורתית להרצת פונקציות רשת כ-VMs, CNF מנצל את היתרונות של סביבות קונטיינרים ומערכות ניהול כמו Kubernetes. כך, CNF מספק גישה קלילה יותר לניהול רשת, ומאפשר לספקי שירות להפעיל פונקציות רשת בניידות וגמישות מקסימלית.

בפרויקט זה, כל רכיבי הרשת המרכזיים – XRD (נתב וירטואלי מבוסס Cisco IOS XR).

ה-XRDs (XRd-1 ו-XRd-2) מתפקדים כנתבים וירטואליים כקונטיינרים, מה שמאפשר ניהול וגמישות בהפעלה ובתחזוקה.

PEER ו-CNF – הינם התקני קצה המדמים את "הלקוחות" גם הם מופעלים כקונטיינרים. משמשים כנקודות קצה לבדיקות עומסים מדידות וניתוח תעבורה.

3.4. וירטואליזציה Virtualization

3.4.1. וירטואליזציה (Virtualization) היא טכנולוגיה שמאפשרת להריץ מספר מערכות הפעלה ותוכנות על גבי מחשב פיזי יחיד, באמצעות יצירה של "מכונות וירטואליות" (VM - Virtual Machines).

כל מכונה וירטואלית (VM) פועלת כמחשב עצמאי עם מערכת הפעלה, יישומים, ונתונים, כשהיא מבודדת ממכונות וירטואליות אחרות שרצות על אותו שרת פיזי.

3.4.2. מושגים מרכזיים בוירטואליזציה

VM - Virtual Machine - מכונה וירטואלית היא סביבת מחשב וירטואלית המדמה מחשב אמיתי. כל VM יכולה להריץ מערכת הפעלה ותוכנות כאילו היא מחשב עצמאי, ומוגדרת על ידי מספר משאבים כמו זיכרון, מעבד ואחסון שמוקצים לה דרך ה**Hypervisor**.

Hypervisor - התוכנה שמנהלת את המכונות הווירטואליות על גבי שרת פיזי. הוא אחראי על הקצאת משאבים בין ה-VMs ועל הבידוד ביניהן.

Encapsulation

קפסולציה היא תכונה של VM שבה כל הנתונים והתצורה של המכונה הווירטואלית נשמרים בקובץ אחד או מספר קבצים. קפסולציה מאפשרת העברה קלה של VM בין שרתים שונים, ויוצרת "קפסולה" הכוללת את כל התלויות של ה-VM.

Isolation

בידוד הוא תכונה חשובה בוירטואליזציה, בה כל VM מופרדת מהשאר ומערכת ההפעלה שלה אינה יכולה להשפיע על מערכות הפעלה אחרות. כך, תקלה או תקיפה באחת מהמכונות לא תשפיע על שאר המכונות הווירטואליות או על השרת הפיזי.

3.5. תשתית הענן (Cloud infrastructure)

ענן הוא מערכת טכנולוגית המספקת למשתמשים גישה לשירותים, תוכנות, ומשאבים שונים דרך האינטרנט, מבלי צורך בהתקנתם או תחזוקתם במחשבים מקומיים. במקום לשמור ולאחסן נתונים על מחשבים פרטיים או על שרתים מקומיים, המידע והתוכנות נשמרים במרכזי נתונים מרוחקים, והשירותים ניתנים דרך האינטרנט.

3.5.1. מאפיינים עיקריים של הענן

גישה מרחוק: המשתמשים יכולים לגשת לקבצים, תוכנות ומשאבים אחרים מכל מקום בעולם, כל עוד יש להם חיבור לאינטרנט.

שימוש לפי צריכה (Pay-as-you-go): תשלום מתבצע לפי כמות השימוש, כך שמשתמשים לא צריכים לשלם על משאבים שאינם משתמשים בהם. אפשר להגדיל ולהקטין את כמות המשאבים הנדרשים בהתאם לצורך.

גמישות וסקלאביליות: הענן מאפשר להגדיל ולהקטין משאבים במהירות, מה שמסייע לארגונים להתמודד עם שינויים בעומסי העבודה מבלי לרכוש ולהתקין חומרה פיזית.

שירותים מנוהלים: ספקי ענן דואגים לתחזוקה, אבטחה ועדכונים של התשתית, כך שהמשתמשים יכולים להתמקד בצרכים העסקיים או האישיים שלהם מבלי לנהל את כל המערכת.

3.5.2. סוגי עננים

ענן ציבורי: מופעל ונמצא בבעלות ספק ענן חיצוני כמו אמזון, גוגל מייקרוסופט ומספק שירותים למספר משתמשים וארגונים.

ענן פרטי: מופעל על ידי ארגון לשימוש פנימי בלבד, ומנוהל לרוב במרכז הנתונים של הארגון.

ענן היברידי: משלב את היתרונות של ענן ציבורי ופרטי, כאשר חלק מהיישומים והנתונים נשמרים בענן הפרטי וחלקם בענן הציבורי.

3.5.3. סוגי שירותים בענן

תשתית כשירות (IaaS - Infrastructure as a Service)

מספקת למשתמשים גישה לתשתיות מחשוב בסיסיות כמו שרתים, אחסון ורשתות. המשתמש יכול להקים ולנהל את מערכות ההפעלה והיישומים שלו.

פלטפורמה כשירות (PaaS - Platform as a Service)

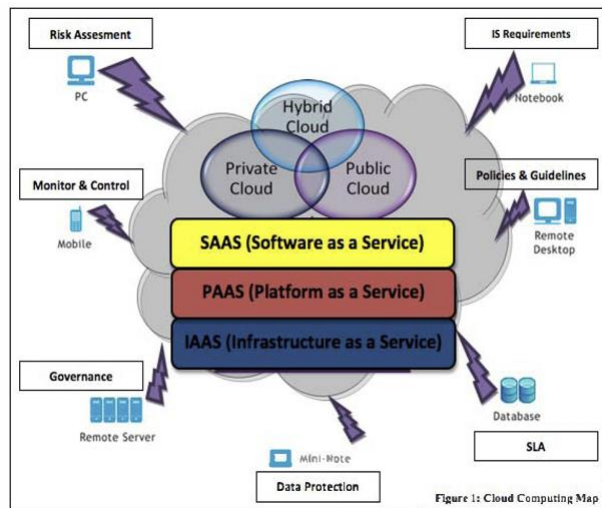
מספקת סביבה לפיתוח והרצה של יישומים מבלי שהמשתמש יצטרך לנהל את התשתית או מערכת ההפעלה.

תוכנה כשירות (SaaS - Software as a Service)

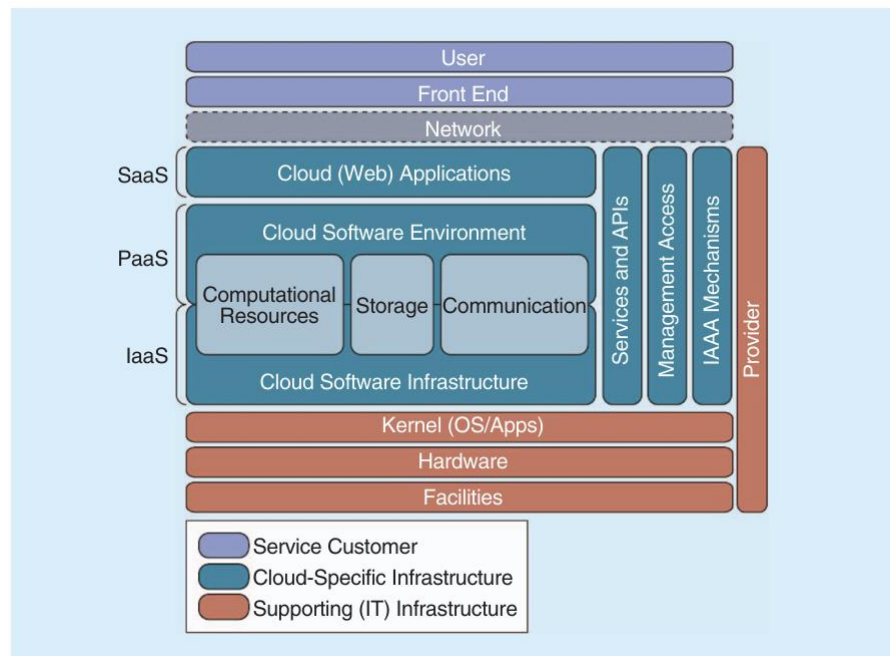
מאפשרת שימוש בתוכנות דרך האינטרנט ללא התקנה מקומית. השירות מנוהל במלואו על ידי ספק הענן, והמשתמשים מקבלים גישה לתוכנה לפי דרישה.

פונקציה כשירות (FaaS - Function as a Service)

מודל זה נקרא גם "מחשוב ללא שרת" (Serverless Computing) "שבו קוד רץ כתגובה לאירועים, מבלי צורך בניהול שרתים כלל.



X. מפת תצורות ושירותי ענן



X. ארכיטקטורת תשתית ענן

3.5.4. קונטיינרים (Kubernetes)

Container כטכנולוגיה:

קונטיינר (Container) מספק מעטפת וירטואלית רזה להרצה של אפליקציות יחד עם הדרישות הנלוות להרצה כמו ספריות וקבצים. טכנולוגיית הקונטיינרים מאפשרת להריץ את היישום באופן אחיד בכל סביבה – מהמחשב האישי ועד למערכות ענן – ללא תלות במערכת ההפעלה של השרת המארז. השימוש בקונטיינרים מאפשר להימנע מתלות בסביבה או במערכת ההפעלה, שכן כל קונטיינר כולל עותק מבודד של היישום והמשאבים הדרושים לו מה שמספק ניידות גבוהה, יעילות בניצול משאבים, וקיצור זמני עלייה בהשוואה למכונות וירטואליות (VMs).

בעקבות יתרונות אלו, טכנולוגיית הקונטיינרים הפכה לסטנדרט בתעשייה עבור פיתוח, בדיקה ופריסה של יישומים, במיוחד בסביבות של מיקרו-שירותים וענן.

קוברנטיס (Kubernetes) היא פלטפורמה לניהול ואוטומציה של יישומים מבוססי containers המאפשרת לפרוס, לנהל ולהריץ יישומים בצורה יעילה ומבוזרת בסביבות מחשוב שונות. קוברנטיס נוצרה על ידי גוגל והועברה לפרויקט קוד פתוח, וכיום היא מנוהלת על ידי CloudNative Computing Foundation (CNCF).

מושגים ואבני בניין:

1. **Container** - יחידת הקוד הקטנה ביותר, המשלבת את כל הדרוש להפעלת יישום או שירות (כמו קוד, ספריות, ותלויות).
2. **Pod** - היחידה הבסיסית ביותר לניהול בקוברנטיס, ומורכבת מ**Container** אחד או יותר הרצים יחד. הפוד משתף כתובות IP ונפחים עם כל **Containers** שבתוכו.
3. **Node** - שרת יחיד שבו רצים הפודים. נוד יכול להיות שרת פיזי או וירטואלי ועליו רץ Agent שמנהל את פעולת ה **Containers** ומאפשר את תקשורתם עם רכיבי קוברנטיס נוספים.
4. **Cluster** – מקבץ של **Nodes** המריץ את יישומי הקונטיינרים בצורה מתואמת. קוברנטיס מנהלת את ה-Cluster כיחידה אחת ומאפשרת ניהול מרכזי לכל הפודים וה-**Nodes** בו.
5. **Service** - אובייקט שמייצג קבוצת פודים עם אותה פונקציונליות ומספק גישה יציבה אליהם דרך כתובת IP קבועה (Virtual IP).
6. **Replica Set** - רכיב שמוודא מספר עותקים רצויים של פוד מסוים, ובכך דואג לשמור על יציבות היישום במקרה של כשל.
7. **Deployment** - מספק כלים לניהול ולפריסה של יישומים, ומאפשר לבצע עדכונים ושינויים בגרסה של היישום בצורה חלקה.

8. **Control Plane** - רכיב השולט על ניהול כלל המשאבים והקצאות בקלסטר. הוא מכיל מספר שירותים חשובים כמו API Server המאפשר תקשורת בין כל רכיבי המערכת, ו-Scheduler שמקצה משאבים בצורה אופטימלית.

בקוברנטיס יש פתרונות לאתגרים שנוצרים כאשר עובדים עם containers כגון ניהול תצורה, העברה בין שרתים, ואיזון עומסים. היא מציעה מערכת לניהול containers בצורה שתומכת בסקלabilיות, יציבות, וזמינות גבוהה, וכך מתאימה לארגונים הרוצים לייעל את ניהול היישומים שלהם בצורה אוטומטית.

היתרונות בשימוש בקוברנטיס:

1. **סקלabilיות**: מאפשרת הגדלה והקטנה של מספר הפודים בהתאם לצורך ואוטומטית.
2. **התאוששות מכשל**: במידה ופוד או נוד נופלים, קוברנטיס מבצעת התאוששות אוטומטית ומשחזרת את היישום.
3. **איזון עומסים**: מאפשרת איזון עומסים בין המיכלים השונים ומייעלת את חלוקת המשאבים.
4. **ניהול גרסאות**: מאפשרת ביצוע עדכונים ושדרוגים בצורה הדרגתית ובטוחה.
5. **גמישות ויכולת ניידות**: ניתן לפרוס את המערכת בסביבות ענן שונות או בסביבות מקומיות.

3.5.5 התבססות על מחקרים

בפרויקט זה, נערכה סקירת ספרות מקיפה המתמקדת בטכנולוגיות NFV (Network Function Virtualization) ו-CNF (Cloud Native Functions) לצד תשתיות הענן והפונקציות הווירטואליות. סקירה זו שימשה כבסיס לפיתוח ארכיטקטורת רשת מודרנית עבור ספקי שירותי אינטרנט (ISP), המאפשרת פריסת שירותים רשתיים גמישים, יעילים וסקלביילים. בסקירה זו מודגש הערך של CNF, המייצג שלב מתקדם במעבר מתשתיות וירטואליות מבוססות VM (מכונות וירטואליות) לארכיטקטורות מבוססות קונטיינרים.

רקע וטכנולוגיות NFV ו-CNF
NFV היא טכנולוגיה המספקת יכולת וירטואליזציה לתפקודי רשת (VNFs) על גבי תשתיות גנריות, במטרה לשפר את הגמישות ולהפחית את העלויות של פריסת רשת. מחקרים מצביעים על כך שהמימוש של NFV אפשר לספקי שירותי תקשורת להציע שירותים מגוונים תוך ניצול משאבי רשת בצורה יעילה. אולם, עם המעבר לשירותים מורכבים יותר ולעומסי עבודה מגוונים בעידן ה-5G ו-VNF, IoT, מוגבל מבחינת הסקלabilיות והביצועים בהשוואה ל-CNF.

הטכנולוגיה של CNF, המבוססת על קונטיינרים, נחשבת לפתרון יעיל יותר בסביבות רשת ענן. CNF מאפשר שימוש במיקרו-שירותים גמישים המותאמים לסביבות קונטיינרים, המאפשרים גמישות, אוטומציה ויכולת להרחבת משאבים בצורה דינמית. כפי שמצוין במאמר "VNF and CNF Placement in 5G: Recent Advances and Future Trends", השימוש ב-CNF מספק "יתרון משמעותי בסביבות 5G בשל האפשרות ליעול משאבי רשת והתמודדות עם עומסי עבודה כבדים".

המודרניזציה בטכנולוגיות CNF
יתרונות הטכנולוגיה של CNF מתבטאים בפריסת פונקציות רשת בצורה מודולרית על גבי ענן, מה שמאפשר לספקי התקשורת לספק שירותים במהירות רבה יותר ולשפר את חווית המשתמש באמצעות יכולת התאמה גבוהה של משאבים. Kubernetes, לדוגמה, היא מערכת הניהול המובילה לניהול קונטיינרים ומאפשרת לספקי שירותי תקשורת לנהל ולספק שירותים בקלות תוך שמירה על שרידות וביצועים גבוהים גם בעת שינויי עומסים על הרשת.

בכדי להבין בצורה רחבה את יתרונות הענן, המושגים והתצורות השונות, בחרנו לקרוא מספר מאמרים ב-IEEE, בחלק זה נפרט על 2 מאמרים מרכזיים שעזרו לנו להעמיק את הידע בתחום ולהביט בנקודת מבט רחבה יותר על מגמות עתידיות, ההיסטוריה, הארכיטקטורה ולהבין את היתרונות והחסרונות שיש ליכולות אלו להציע.

המאמר הראשון שנפרט עליו : Cloud Computing -Concepts, Architecture and Challenges
2012 International Conference on Computing, Electronics and Electrical Technologies
[ICCEET]

מאמר זה פורסם ב-2012, מציג את ההיסטוריה של תשתיות ענן, מושגים בתחום וארכיטקטורות במאמר זה התבססנו על ידע תאורטי בנושאי תצורות ענן, שירותי ענן הארכיטקטורות המופיעות ובנוסף התבססנו על היתרונות והחסרונות המופיעים במאמר כאשר היתרונות כוללים : ניהול קל, הפחתת עלויות, שירותים יציבים, גיבוי נתונים ו"מחשוב ירוק"(הפחתת פסולת אלקטרונית), לעומת החסרונות שבעיקרן כוללים דרישות אבטחת מידע חזקות, תלות בספקי ענן ועלויות תפעול שעלולות לגדול בשימוש מתמשך וצרכים משתנים.

מאמר נוסף שהעמקנו בידע המפורט בו הוא המאמר : Cloud Computing: Architecture, Vision, Challenges, Opportunities, and Emerging Trends
2023 International Conference on Computing, Communication, and Intelligent Systems
(ICCCIS)

מאמר זה פורסם בשנת 2023 וכולל ארכיטקטורות ומידע עדכני בתחום הענן, נעזרנו בו בכדי להבין את מבנה מחשוב הענן, ההזדמנויות והחזון, שירותים שונים בענן, יתרונות וחסרונות שכמו שבמאמר הקודם כללו גם הן ביתרונות את יכולות הגמישות ויכולת ההרחבה והגדילה, חיסכון בעלויות, זמינות ואמינות ומערכות גיבוי חזקות ובנוסף גם במאמר זה החסרונות היו צורך באבטחת מידע ופרטיות, תלות בספקי ענן וצורך בניהול משאבים הדוק בכדי למנוע בזבז כספי.

4. פרק 3 : תוכן הפרויקט, הצגת האתגרים והפתרונות

4.1. פתיחה

פרק זה עוסק בניתוח האתגרים המרכזיים איתם מתמודדים ספקי התקשורת בארכיטקטורות הרשת המסורתיות המבוססות על חומרה ייעודית. נבצע ניתוח מעמיק של כל אתגר, ונבחן כיצד טכנולוגיות ה-NFV (Network Function Virtualization) וה-CNF (Cloud-Native Network Functions) מסוגלות לספק מענה לאתגרים אלו. נדון בשאלה האם פתרונות ענן אלו בשלים דיים כדי לשמש תשתית לספקי שירותי אינטרנט (ISP), והאם הם עומדים בדרישות הטכניות להקמת רשת המסוגלת לחבר לקוחות שונים ולספק רמת שירות נדרשת (SLA).

לאחר מכן נבחן את השיטות השונות ונבצע ניתוח חלופות להנגשת שירותי רשת על ידי ספקי תקשורת (ISP) ונסקור את ההתפתחות שעברה התעשייה מהשיטה המסורתית לגישות וירטואליזציה מתקדמות. נעבור דרך המודל המסורתי, שהתבסס על חומרה ייעודית, לעבר וירטואליזציה באמצעות VNF (פונקציות רשת וירטואליות), ועד לגישת ה-CNF (פונקציות רשת מבוססות ענן) המתקדמת. נעמוד על יתרונותיה של כל שיטה, ונבחן כיצד התמורות בתחום הווירטואליזציה והמעבר לתשתיות ענן מודרניות שינו את היכולות של ספקי התקשורת להציע שירותים גמישים, יעילים ומותאמים לעידן המודרני של רשתות תקשורת. סקירה זו תספק תובנות לגבי הדרך בה כל טכנולוגיה משפיעה על שיפור השירותים והמענה לצרכים ההולכים וגוברים של השוק.

לבסוף גם נסקור את תהליך בחירת התשתית המתאימה להרצת הנתבים התוכנתיים ולביצוע הבדיקות בפרויקט. האפשרויות שעמדו לפנינו כללו תשתיות שונות, כמו תשתית וירטואלית קלאסית על גבי שרתים פיזיים (NFV), ענן פרטי עם תמיכה בקוברנטיס לניהול עומסים וגמישות משופרת, או תשתיות ענן ציבוריות המציעות שירותי קונטיינר ומחשוב מבוססי ענן. ביצענו השוואה גם להרצה של הארכיטקטורה על נתבים מבוססי חומרה. לאחר בחינת היתרונות והחסרונות של כל אפשרות, בחרנו להריץ את הפרויקט על גבי תשתית ענן ציבורית של Amazon (AWS) אשר הציעה מציעה גמישות מקסימלית, ביצועים גבוהים, אפשרות להרחבת משאבים מהירה ואבטחה מותאמת לסטנדרטים גבוהים.

4.2. ניתוח האתגרים המרכזיים איתם מתמודדים ספקי

התקשורת בארכיטקטורות הרשת מסורתיות מבוססות חומרה

ארכיטקטורות מסורתיות של ספקי תקשורת דורשות חומרה ייעודית עבור כל פונקציית רשת, דבר שגורם לעלויות גבוהות, עיכובים בפריסת שירותים חדשים, ותלות בספקים בודדים (VENDOR LOCK IN).

טכנולוגיה זו אינה מותאמת לעידן שבו גמישות, סקאלביליות ויכולת הסתגלות מהירה לצרכים משתנים הם תנאים הכרחיים להצלחה. הדרישות הגדלות משירותי תקשורת מהירים ואמינים יותר, כמו IOT ו-5G והצורך בתמיכה במספר רב של לקוחות תוך שמירה על SLA מטרת פרק זה

היא לבחון את האתגרים המרכזיים ולהציע פתרונות חדשניים באמצעות טכנולוגיות NFV מבוססת CNF.

4.2.1. אתגר טכני: האם ניתן לממש ארכיטקטורות מורכבות ומתקדמות של ספקי תקשורת ISP בעזרת נתבים מבוססי תוכנה?

אתגר - ISPs נדרשים לספק קישוריות רציפה, גמישה ומאובטחת בין סניפים מבוזרים גיאוגרפית של ארגונים, תוך שמירה על זמני תגובה נמוכים וביצועים גבוהים. ניהול תעבורה בין סניפים מרוחקים תוך שמירה על הפרדה לוגית בין תעבורת הלקוחות מהווה אתגר משמעותי, במיוחד ברשתות מסורתיות שמתמודדות עם עומסים כבדים וסיכון לזליגת מידע.

במבנה מסורתי, רשתות נוטות להיות נוקשות ופחות גמישות, מה שמקשה על התאמה מהירה לצרכים משתנים של לקוחות. כמו כן, עומס כבד יכול להוביל לזמני תגובה ארוכים ולירידה בביצועים – שני פרמטרים בעייתיים במיוחד בארכיטקטורות מבוזרות כמו אלו ש-ISP נדרשים לתמוך בהן.

פתרון - בפרויקט שלנו, מימשנו ארכיטקטורת רשת מתקדמת עבור ספקי תקשורת (ISP) אשר מתמודדת עם אתגרי הקישוריות, האבטחה, והביצועים הנדרשים לניהול תעבורה בין סניפים מבוזרים גיאוגרפית של לקוחות. הארכיטקטורה עושה שימוש משולב בפרוטוקולים מרכזיים כגון IS-IS, GRE, MPLS, BGP, VRF ו-L3VPN, כל אחד מהם תורם לרשת גמישה ועמידה. IS-IS משמש לניהול ניתוב דינמי ואופטימלי, GRE יוצר מנהרות וירטואליות שמאפשרות העברת תעבורה מאובטחת ומבודדת, MPLS ו-BGP מסייעים בניהול ניתוב יעיל ומבוקר עם ביצועים גבוהים, ואילו VRF ו-L3VPN מספקים הפרדה לוגית מלאה בין תעבורות לקוחות שונים, תוך שמירה על אבטחת מידע מלאה. שילוב זה יוצר פתרון המותאם לצרכים המודרניים של ISPs, ומספק תשתית רשת אמינה, מאובטחת וגמישה המותאמת לעומסים גבוהים ודרישות משתנות. החברות המובילות בשוק בתחום (Juniper, Cisco) טוענות שהפתרונות מבוססי התוכנה בשלים דיו למימוש טכנולוגיות מתקדם ועל כן ביצענו מימוש מורכב וערכנו בדיקות מקיפות.

4.2.2. אתגר טכני: האם NFV ו-CNF מסוגלים לספק תשתית ל-ISP ברמת השירות SLA הנדרשת מהספקים?

אתגר - אחד האתגרים המרכזיים בפריסה של פתרונות NFV ו-CNF הוא השאלה האם הטכנולוגיות הללו בשלות מספיק לספק תשתית יציבה ואמינה לספקי תקשורת, בעיקר בהקשר של רשתות MPLS. רשתות אלו נדרשות לחבר לקוחות שונים בצורה מאובטחת, עם ביצועים גבוהים ועמידה בדרישות SLA.

SLA (Service Level Agreement) זו התחייבות מצד ספק שירות לשמור על רמת שירות מוגדרת מראש, הכוללת פרמטרים כגון זמינות, זמן תגובה ורמת ביצועים. עבור ספקי תקשורת, SLA גבוה הוא קריטי להבטחת שביעות רצון הלקוחות ולמניעת קנסות.

פתרון - תשתית NFV מספקת את הגמישות הנדרשת להקמת תשתית מבוססת ענן העומדת בדרישות ה- SLA. הפריסה הווירטואלית מאפשרת לספק שירותי MPLS תוך שמירה על ביצועים

גבוהים, באמצעות שימוש בטכנולוגיות כמו Kubernetes לניהול משאבים בצורה אוטומטית ודינמית. בנוסף, ניתן לבצע בדיקות SLA מתמשכות, למדוד ביצועים בזמן אמת, ולהתאים את המשאבים בהתאם לצרכי הלקוח. היכולת לבצע התאמות ושדרוגים במהירות באמצעות גמישות ה CNF מקלה על עמידה בדרישות הביצועים של רשתות MPLS מודרניות. אף על פי שמדובר בטכנולוגיה חדשנית ישנם מספר חבירות מובילות כמו CISCO וJUNIPER הטוענות שמסוגלות להנגיש שירותי ISP בצורה מודרנית ועל כן נבצע חקר ביצועיים ומדידות פרמטרים הנדסיים לבדיקת היתכנות פתרון זה.

4.2.3. אתגר טכני – הבטחת הפרדה לוגית ושמירה על מבנה הרשת של הלקוחות

אתגר - ארגונים רבים, קיים צורך לשמור על מבנה רשת פנימי עקבי, כולל כתובות IP, סגמנטים ותצורות קיימות, כדי למנוע שיבושים בפעילות העסקית השוטפת ולהבטיח תאימות עם תשתיות קיימות. כל שינוי במבנה הרשת עלול להוביל לבעיות תאימות, לשיבוש תהליכים עסקיים קריטיים ולפגיעה בתפקוד התקין של מערכות חיוניות בארגון. בנוסף, ספקי שירותי אינטרנט (ISP) נדרשים להציע פתרון ניתוב יעיל ובטוח עבור מספר לקוחות על גבי אותה תשתית פיזית, תוך שמירה על הפרדה מלאה בין תעבורות הלקוחות ושמירה על אבטחת מידע גבוהה.

פתרון – יישום MPLS VPN עם VRF בפרויקט

בפרויקט שלנו, נעשה שימוש בטכנולוגיית MPLS VPN, המאפשרת ללקוחות לשמור על כתובות IP וסגמנטים קיימים ברשת שלהם, ללא צורך בהתאמות ושינויים ברשת הפנימית של הלקוח. באמצעות טכנולוגיית VRF (Virtual Routing and Forwarding), כל תעבורת רשת עבור לקוח מסוים מנותבת בנפרד, בהתבסס על מדיניות מותאמת אישית, כך שמובטחת הפרדה מלאה בין הלקוחות. בפרויקט, יישום ה-VRF מבוצע על כל אחד מהנתבים הווירטואליים (XRd-1 ו-XRd-2), וכך ניתן להבטיח שלכל לקוח יש "רשת פרטית וירטואלית" משלו על גבי התשתית הפיזית המשותפת. השילוב של MPLS עם VRF יוצר פתרון המותאם לצרכי הארגון, ומאפשר לספק הפרדה לוגית מלאה בין הלקוחות תוך שמירה על ביצועים גבוהים ועמידה בדרישות אבטחת מידע קפדניות.

4.2.4. אתגר טכני – אבטחת מידע ושמירה על פרטיות במערכות מרובות-לקוחות

אתגר - להבטיח שמירה על אבטחת מידע ופרטיות הלקוחות, במיוחד לאור העובדה שתשתית הרשת המשותפת מאפשרת גישה למספר לקוחות על גבי אותה פלטפורמה פיזית. ככל שהארכיטקטורה מבוססת על תשתיות משותפות, הסיכון לזליגת מידע והאפשרות למתקפות סייבר גוברים. ספקי שירותים (ISPs) מחויבים להבטיח שהמידע של כל לקוח יישמר מבודד ובטוח, ללא סכנה של חדירה לא מורשית או חשיפת נתונים ללקוחות אחרים.

פתרון - שימוש ב-VRF וב-GRE להבטחת הפרדה ואבטחה לוגית בתעבורת הרשת

בפרויקט שלנו, יישמנו את טכנולוגיית ה-VRF (Virtual Routing and Forwarding), המאפשרת יצירת טבלאות ניתוב נפרדות עבור כל לקוח על גבי אותה תשתית פיזית. כל VRF פועל כמערכת ניתוב עצמאית, ומספק הפרדה לוגית מלאה לכל לקוח, כך שהתעבורה שלו אינה מתערבבת עם

תעבורת לקוחות אחרים. טכנולוגיה זו מאפשרת ללקוחות לשמור על פרטיות מוחלטת של התעבורה שלהם.

בנוסף, נעשה שימוש במנהרות GRE (Generic Routing Encapsulation) ליצירת שכבת אבטחה נוספת בתקשורת בין נקודות הקצה המרוחקות. המנהרות מאבטחות את התעבורה על ידי עטיפת חבילות המידע והעברתן בצורה מוצפנת בין הסניפים והמרכזים של כל לקוח, ובכך מונעות גישה לא מורשית ומקטינות את הסיכון לזליגת מידע בין רשתות הלקוחות השונות. השילוב בין VRF ו-GRE מספק פתרון אבטחה חזק, שמאפשר ללקוחות לעבוד בבטחה על גבי תשתית משותפת מבלי לפגוע בפרטיות ובביצועים.

4.2.5. אתגר עסקי טכנולוגי ברשת המסורתית - זמני אספקה ארוכים לתשתיות חדשות

אתגר - בשימוש במערכות רשת מסורתיות המבוססות על חומרה ייעודית, תהליך רכישת החומרה, פריסתה והתקנתה הוא תהליך ממושך. כאשר עולה הצורך להוסיף שירות חדש או להרחיב את תשתית הרשת, נדרש מלאי של רכיבים פיזיים, משלוח הציוד לאתר, התקנה, וקונפיגורציה של הציוד בשטח. תהליך זה יוצר תלות רבה בשרשראות אספקה ובכוח אדם מיומן, מה שמוביל לעיכובים משמעותיים ביכולת של הספק להציע שירותים חדשים ללקוחות או לשדרג שירותים קיימים בזמן. כך, נוצר קושי במתן מענה מהיר לדרישות השוק ולהתאמת הרשת לשינויים בצרכים העסקיים.

פתרון - פריסת שירותים מהירה וגמישה באמצעות טכנולוגיות וירטואליזציה וענן תהליך הפריסה מהיר יותר. אין צורך בהמתנה לרכש חומרה ייעודית, שכן פונקציות הרשת מופעלות כתוכנה על גבי תשתיות מחשוב גנריות, אותם נדרש להתקין באתר הלקוח פעם אחת בלבד ולא עבור כל הגדלת שירות. כך ניתן להגדיל את השירותים באופן כמעט מיידי ומרחוק ולפרוס אותם בטכנולוגית ענן אצל אתר הלקוח. זה מאפשר לספקים להשיק שירותים חדשים במהירות רבה ולשדרג את הרשת בהתאם לדרישות השוק.

4.2.6. אתגר עסקי טכנולוגי – תחזוקת תשתית ה-ISP המסורתית היא יקרה ומורכבת

תשתיות מסורתיות מצריכות תחזוקה יקרה ומורכבת, הכוללת תיקונים, שדרוגים והחלפת רכיבים כאשר הם מגיעים לסוף חייהם (End of Life - EOL) או לסוף התמיכה (End of Support - EOS). כל תהליך תחזוקה או שדרוג מצריך הגעה פיזית לאתרי הלקוח, מה שמגדיל את העלויות התפעוליות ומעלה את הסיכון לשגיאות אנוש במהלך ביצוע הפעולות. תלות זו בציוד פיזי ובצוות בשטח מהווה אתגר משמעותי לספקי שירותי אינטרנט (ISP) המבקשים לשמור על זמינות גבוהה ורציפות תפעולית.

פתרון - הטכנולוגיה מאפשרת פתרון גמיש וחסכוני יותר לאתגרי התחזוקה. באמצעות NFV ו-CNF, ניתן לבצע שדרוגים ותיקונים מרחוק ללא צורך בהגעה פיזית לאתר הלקוח. פונקציות רשת וירטואליות (VNFs) ופונקציות מבוססות ענן (CNFs) מאפשרות ניהול מרכזי ואוטומטי של תהליכי תחזוקה ושדרוג, תוך שימוש בממשקי ניהול מתקדמים. כך ניתן לפרוס גרסאות חדשות של שירותים באופן מודולרי וללא השבתה.

4.2.7. אתגר עסקי טכנולוגי – יכולת מוגבלת לשינויים והתאמה לצרכים הטכנולוגיים

המשתנים במהירות

מערכות חומרה מסורתיות נוטות להיות קשיחות ובלתי גמישות, שכן כל רכיב חומרה מיועד לספק שירות מסוים ואינו ניתן לשינוי או לשדרוג מרחוק. חוסר הגמישות הזה מקשה על התאמת התשתית לשינויים מהירים בשוק, כמו גם להוספת טכנולוגיות חדשות. כל הוספת שירות, כמו שירותי מובייל מתקדמים, דור 5, (5G) אינטרנט של הדברים (IoT) או רכיבי אבטחת מידע, מצריכה התקנה של ציוד פיזי חדש, הכרוך בתיאום עם ספקי ציוד ובתלות משמעותית בשרשראות האספקה. מצב זה יוצר עיכובים בתגובה לשוק ומשפיע על יכולת ההטמעה של טכנולוגיות חדשות במהירות הנדרשת.

בנוסף, מערכות חומרה מסורתיות מתקשות להתמודד עם שינויים דינמיים בעומסים. לדוגמה, בעת שידור של אירוע ספורט גדול, ספקי התקשורת נדרשים לתמוך בעלייה חדה בביקוש לרוחב פס ובצורכי השירות. במערכות המסורתיות, הגדלת המשאבים מצריכה התקנת ציוד פיזי נוסף שאינו ניתן להסרה או צמצום כשהעומסים חוזרים למצבם הרגיל, מה שמוביל לעומס יתר בתשתיות או לבזבז משאבים כאשר הצרכים משתנים.

פתרון באמצעות באמצעות טכנולוגיות NFV ו- CNF ניתן ליצור ארכיטקטורת רשת גמישה ודינמית שמאפשרת התאמה מהירה לשינויים ולצרכים משתנים. פונקציות רשת מבוססות קונטיינרים (CNF), כגון ראוטרים, חומות אש ושירותים נוספים, מופעלות כאפליקציות תוכנה גמישות המבוססות על קונטיינרים. מבנה זה מאפשר לפרוס שירותים בקצב מהיר יותר, תוך התאמה אוטומטית לעומסים משתנים. היכולת לבצע סקלבליות אוטומטית (auto-scale) ולשכפל שירותים לפי דרישה, מאפשרת לספקי התקשורת להתרחב בעת עומסים ולהקטין משאבים בשעות רגיעה, תוך חיסכון בעלויות תפעול.

לדוגמה, בעת שידור משחק ספורט חשוב, ניתן להרחיב את השירותים על ידי יצירת עותקים נוספים של פונקציות רשת וירטואליות, שיפעלו לצידם של השירותים הקיימים ויבטיחו מענה לביקושים מוגברים. עם תום האירוע וחזרת הביקושים לרמתם הנורמלית, ניתן לצמצם את השירותים בקלות, כך שהמערכת תישאר יעילה, גמישה וחسכונית.

4.2.8. אתגר עסקי טכנולוגי בארכיטקטורות מסורתיות – Vendor Lock-In (נעילת ספקים)

אתגר - בארכיטקטורות מסורתיות, ספקי תקשורת נדרשים לרכוש חומרה ייעודית מיצרנים מסוימים כדי לספק שירותי רשת. מצב זה יוצר בעיית "נעילת ספקים" (Vendor Lock-In), שבו ספקי התקשורת תלויים בספק חומרה יחיד או במספר מוגבל של יצרנים עבור כל תשתית הרשת שלהם. התלות הזו נובעת מכך שמערכות החומרה המסורתיות מותאמות באופן ייחודי לשירותים או פונקציות רשת מסוימות, דבר המקשה על אינטגרציה עם רכיבי רשת מיצרנים אחרים. מעבר ליצרן אחר בתשתיות אלו כרוך בהחלפה פיזית מורכבת של ציוד ומחייב מיגרציה מסובכת, מה שמגביל את יכולת הספק להתרחב, לשנות או לשדרג את הרשת בהתאם לדרישות הטכנולוגיה המתחדשות והצרכים העסקיים המשתנים.

פתרון - שימוש בטכנולוגיות CNF שבהן פונקציות הרשת מיושמות כתוכנה ומופעלות על גבי תשתיות מחשוב גנריות, מאפשר לספקי התקשורת להשתחרר מתלות בספק חומרה אחד. CNF מציעה גישה מודולרית וגמישה, שבה שירותים שונים יכולים לפעול על אותה תשתית מחשוב גנרית וניתנים להפעלה של יצרנים שונים ללא תלות בחומרה ייעודית. כך, כאשר רכיב מסוים מגיע לסוף

חיו (End of Life - EOL) או לסוף תקופת התמיכה (End of Support - EOS), ניתן בקלות לשדרג את השירותים על ידי עדכוני תוכנה או מעבר לשירות של יצרן אחר, מבלי לשנות את התשתית הפיזית. יתרה מזאת, פתרון זה מאפשר לספקי התקשורת גמישות גבוהה במעבר בין ספקים, שדרוגי תוכנה והתאמה מהירה לשינויים טכנולוגיים בשוק, כל זאת תוך שימור השקעתם בתשתיות המחשוב הקיימות.

4.2.9. אתגר עסקי טכנולוגי בארכיטקטורות מסורתיות – יעילות וצריכת משאבים

אתגר - במערכות מסורתיות, כל שירות רשת מופעל על גבי מכונה פיזית ייעודית, מה שמוביל לצריכת משאבים גבוהה ולחוסר יעילות בשימוש בתשתיות. ריבוי מכונות פיזיות מצריך שטח רב בארונות תקשורת, משאבי חשמל וקירור משמעותיים, ותפעול מורכב הדורש תחזוקה שוטפת והקצאת משאבים רבים. כתוצאה מכך, עלויות התחזוקה וההחזקה עולות באופן משמעותי, והצריכה האנרגטית הגבוהה פוגעת הן בתקציב התפעולי והן בסביבה, עקב פליטת חום מוגברת ודרישות אנרגיה לא יעילות.

פתרון - מעבר לארכיטקטורות מבוססות ענן ולטכנולוגיות וירטואליזציה מתקדמות כגון NFV ו-CNF מאפשר לצמצם באופן ניכר את צריכת המשאבים ואת ההשפעה הסביבתית השלילית. CNF מאפשר להריץ פונקציות רשת רבות על תשתית וירטואלית משותפת, במקום לדרוש מכונה פיזית נפרדת לכל שירות. בזכות כך, נוצרת אופטימיזציה בשימוש בשטח ובמשאבי חשמל וקירור, מה שמפחית את העלויות התפעוליות הכלליות. בנוסף, טכנולוגיות קונטיינרים וכלים לניהול תשתיות מתקדמים כמו Kubernetes מספקים סקלביליות דינמית, שמאפשרת להגדיל או להקטין את המשאבים המוקצים בהתאם לעומס. כך מתאפשרת יעילות גבוהה ותפעול חסכוני באנרגיה, תוך עמידה בדרישות טכניות מתקדמות ושיפור הקיימות הסביבתית.

בעיות התשתית המסורתיות של ספקי התקשורת, כמו זמני אספקה ארוכים, תחזוקה יקרה וחוסר גמישות, יחד עם הדרישות הארכיטקטוניות הטכניות מקבלות פתרונות באמצעות טכנולוגיות NFV ו-CNF. טכנולוגיות אלו מאפשרות מעבר לארכיטקטורות רשת מודרניות, גמישות ודינמיות, העומדות בדרישות הטכניות להקמת רשתות MPLS תוך עמידה בדרישות SLA מחמירות.

4.3. ניתוח והשוואת החלופות להנגשת שירותי רשת על ידי ספקי

התקשורת (ISP) : מהשיטה המסורתית לגישה מבוססת

וירטואליזציה (VNF) ועד לטכנולוגיה מבוססת לענן (CNF)

ההנגשה של שירותי תקשורת על ידי ספקי שירותי האינטרנט (ISP) עברה שלושה שלבים עיקריים – מהשיטה המסורתית שהתבססה על חומרה ייעודית, דרך גישה הווירטואליזציה (Virtualized Network Functions - VNF), ועד לשימוש בטכנולוגיות ענן (Cloud-Native Network Functions - CNF). המאפשרות גמישות, יעילות וסקלביליות רבה יותר. בחלק זה נסקור את שלושת הגישות המרכזיות, נעמוד על יתרונותיהן ומגבלותיהן, ונראה כיצד הן עיצבו את האפשרויות המודרניות של ספקי התקשורת.

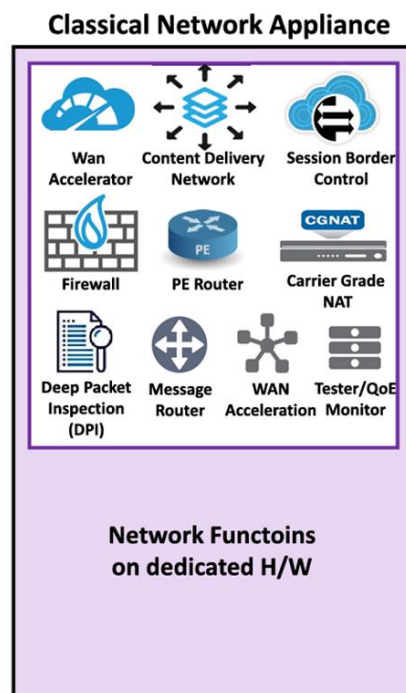
4.3.1. השיטה המסורתית: מכונה פיזית לכל שירות

בעבר, כל שירות רשת שסופק על ידי ה-ISP – כגון נתבים, מתגים, שירותי מובייל וסלולר, שירותי אבטחת מידע, וחלוקת עומסים – הופעל על גבי מכונה פיזית ייעודית. ארכיטקטורה זו כללה רכיבי חומרה ותוכנה מותאמים, שנבנו באופן ייחודי עבור הפונקציה המסוימת של כל שירות. גישה זו הביאה עמה מספר אתגרים משמעותיים:

תהליכי רכש ואספקה ארוכים: כל שירות דרש אפיון תכנון, רכישה ואספקה של חומרה מותאמת, מה שהאט את יכולת ה-ISP להציע שירותים חדשים או לשדרג את הקיימים.

ניהול לוגיסטי מורכב: תחזוקת הציוד, אחסון מלאי של חלקי חילוף וניהול רציף של רכיבים פיזיים הגבירו את סיבוכיות התפעול.

עלויות תפעול גבוהות: הצורך בחומרה ייעודית לכל שירות יצר עלויות תפעול גבוהות, כולל צריכת חשמל, קירור ושדרוגים תכופים.



4.3.2. המעבר לווירטואליזציה בעולם הרשת NFV ו Virtualized Network Functions (VNFs)

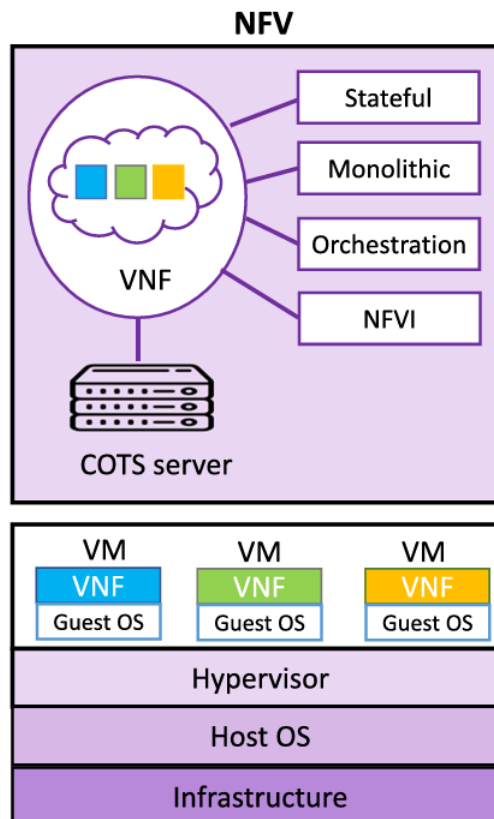
במהלך שנות ה-2010, חברות הטכנולוגיה החלו לאמץ את גישת הווירטואליזציה. המודל החדש, שהתבסס על פונקציות רשת וירטואליות (VNF), אפשר הרצה של מספר שירותים על גבי חומרה פיזית סטנדרטית, מה שצמצם את הצורך ברכישת חומרה ייעודית עבור כל שירות.

באמצעות NFV, ספקי תקשורת יכלו להפעיל פונקציות רשת על תשתיות מחשוב גנריות (COTS - Commercial Off-The-Shelf), דבר שהפחית משמעותית את התלות בחומרה ייעודית. טכנולוגיה זו מאפשרת פריסה, ניהול ותחזוקה של פונקציות רשת בצורה דינמית ומותאמת אישית לדרישות משתנות של לקוחות וספקי שירותי אינטרנט (ISP).

הפחתת תלות בחומרה פיזית: הווירטואליזציה אפשרה לספקי התקשורת להפחית את העלויות באמצעות הפעלת פונקציות רשת כתוכנה, דבר שהקל על השגת יעילות גבוהה יותר.

גמישות וניידות משופרת: המודל הווירטואלי אפשר גמישות רבה יותר ותגובה מהירה לשינויים.

למרות היתרונות, המודל הווירטואלי הציג מגבלות, בעיקר כשנדרש מעבר מהיר בין שירותים או התאמה לעומסים משתנים. מכונות וירטואליות (VMs) היו כבדות יחסית, והווירטואליזציה עצמה לא תמיד אפשרה סקלביליות נדרשת בזמן אמת בעיקר בשל הצורך בניהול שכבת מערכת ההפעלה הנפרדת לכל מכונה וירטואלית. למרות שהמודל החדש הציג שיפור מהשיטה המסורתית, היה לו קושי לתמוך בדרישות גבוהות של גמישות ודינמיות אך הניח את היסודות והתניע את השינוי למעבר לטכנולוגיות וגישות חדשות.



איור 4- בתיאור ניתן לראות תשתית מחשוב אחודה מבוססת חומרה, מעליה מונגשים שירותי רשת בתצורת מכונה וירטואלית VM

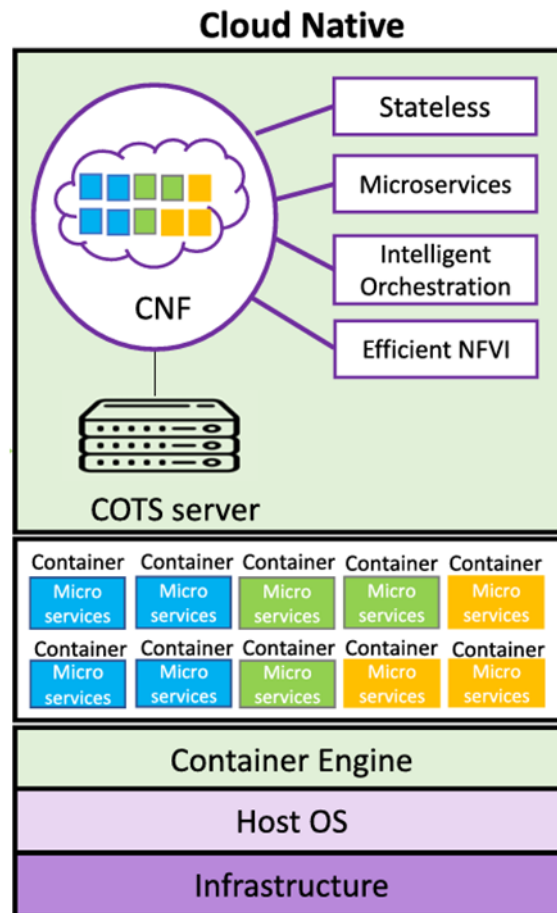
4.3.3. המעבר לפונקציות רשת מבוססות ענן (- Cloud-Native Network Functions (CNF

CNF מהווה את הדור הבא בתחום וירטואליזציית הרשתות, בשונה מ-VNF ה-CNF מבוססת על קונטיינרים (מעטפת וירטואלית להרצה תוכנה) ובניית מתוך עקרונות של Cloud native . קונטיינרים מאפשרים פריסה מהירה יותר של פונקציות רשת, עם צריכת משאבים מופחתת בהשוואה למכונות וירטואליות. יתרונות אלה תורמים לשיפור הביצועים והגמישות של תשתיות הרשת בצורה משמעותית.

פריסה מהירה וגמישה: CNF רזה יותר מ-VNF, מה שמאפשר לפרוס שירותים בצורה מהירה וגמישה, המותאמת לעומסים משתנים.

שיפור בביצועים ובסקלבליות: השימוש בקונטיינרים מביא להפחתה בזמן תגובה latency ובצריכת המשאבים, מה שמאפשר תפקוד יעיל יותר של הרשת.

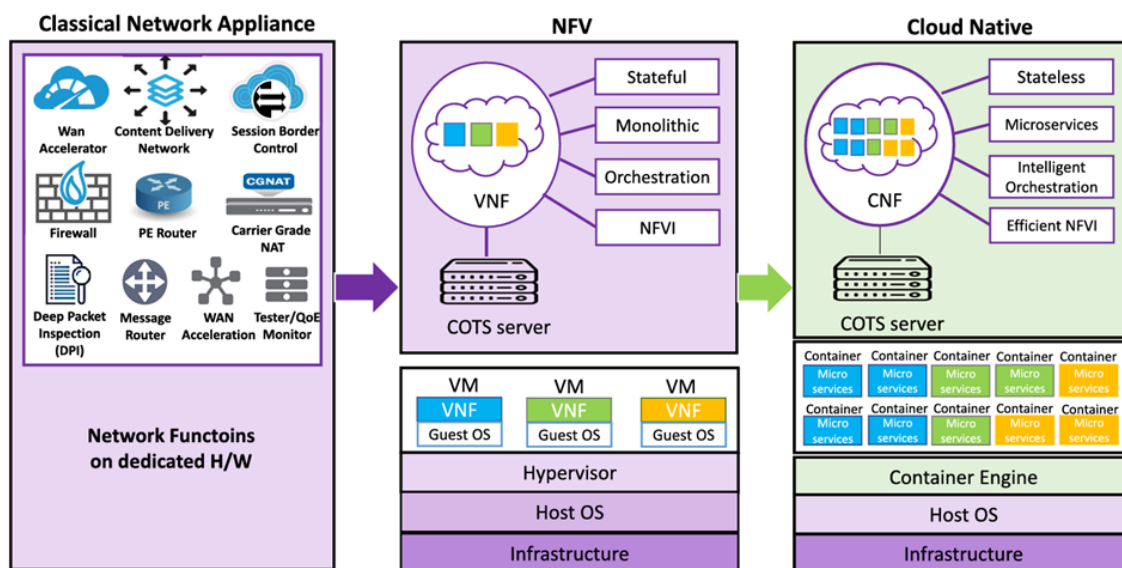
מודולריות והתאמה לדרישות המודרניות: היכולת לפרק אפליקציות למיקרו-שירותים מקלה על ניהול, תחזוקה ושדרוגים של המערכת, תוך הפחתת השבתות.



איור 5 - בתיאור זה ניתן לראות את השירותים אותם מנגיש ספק התקשורת מונגשות על גבי תשתית חומרתית המכילה מערכת הפעלה ומנוע להפעלת קונטיינרים, בתוכם יופעלו אפליקציות מודרניות הבנויות במיקרו סרוויסים שתפקידם לתת יכולות תקשורת שונות.

4.3.4. תוצאות מחקר מאת IEEE בנושא חשיבות הטכנולוגיה

מחקרים עדכניים בתחום, כדוגמת המאמר [VNF and CNF Placement in 5G: Recent Journals & Magazine | IEEE Xplore Advances and Future Trends | IEEE](#) מדגישים את היתרונות המשמעותיים של המעבר לשימוש ב-CNF (פונקציות רשת מבוססות ענן) בעידן ה-5G. המעבר ל-CNF מאפשר חיסכון במשאבים ושיפור משמעותי בשרידות וביכולת להתמודד עם עומסים גבוהים, מאפיינים שהם חיוניים לרשתות תקשורת מודרניות. כפי שמתאר המאמר, "המעבר ל-CNF מספק יעילות רבה יותר בניצול משאבים על ידי יישום שירותים רבים יותר על אותו שרת באמצעות מבנה מיקרו-שירותים וקונטיינרים". בנוסף, מערכת Kubernetes, המהווה פתרון נפוץ לניהול קונטיינרים, מאפשרת לספקי תקשורת לנהל ולהרחיב את הרשת בצורה גמישה ומודולרית, תוך שימוש באוטומציה ובתהליכי פריסה וניהול אוטומטיים, אשר משפרים את ניהול המשאבים והתעבורה ברשת.



איור 6 – הטנספורמציה שעברו שירותי התקשורת המסורתיים ועד לטכנולוגיות המודרניות, מארכיטקטורת מונולית למיקרו סרוויסים.

סקירה זו ממחישה את האבולוציה שעברה תעשיית התקשורת - מהשיטה המסורתית שהתבססה על חומרה ייעודית לכל שירות, למעבר למודל וירטואלי באמצעות VNF, ועד לגישה המתקדמת של CNF המבוססת ענן. כל שלב בהתפתחות זו מציג קפיצת דרך טכנולוגית, שמאפשרת לספקי התקשורת להציע שירותים גמישים, יעילים ובעלי יכולת גידול דינמית המתאימה לדרישות המודרניות של רשתות תקשורת. טכנולוגיות CNF מבוססות ענן מסמנות את השלב המתקדם ביותר בתהליך זה, והן מהוות פתרון אידיאלי לצרכים המודרניים של רשתות תקשורת בעידן הנוכחי והעתיד.

4.4. תהליך בחירת התשתית המתאימה להרצת הנתבים

והרכיבים התוכנתיים וביצוע הבדיקות בפרויקט

בחלק זה נדרשנו לבחור התשתית המתאימה להרצת הנתבים התוכנתיים ולביצוע הבדיקות. מכיוון שמדובר בארכיטקטורה חדשנית הכוללת שימוש בקונטיינרים, פונקציות רשת וירטואליות (VNF) ותמיכה במיקרו-שירותים, נדרשה תשתית שתוכל לספק גמישות, אמינות וביצועים גבוהים. בחירת תשתית מתאימה הייתה קריטית להצלחת הפרויקט, שכן כל אפשרות תשתית מביאה איתה יתרונות וחסרונות שונים מבחינת רמת הביצועים, אפשרויות הגמישות, המורכבות התפעולית, והעלויות הכרוכות בה.

בנוסף, ביצענו השוואה של טכנולוגיות הפריסה של פונקציות הרשת אותן תיארונו במהלך הפרויקט, הרצה כחומרה פיזית, בצורה וירטואלית ובצורה עננית מבוססת קונטיינרים.

4.4.1. בחינת הטכנולוגיה להרצת פונקציות הרשת - טבלה המסכמת את ההשוואה בין

טכנולוגיות החומרה, הווירטואליזציה והענן

טבלה מס' 2 : השוואה בין טכנולוגיות

מאפיין	רכיבי חומרה פיזיים	ענן וקונטנטיס (קונטיינרים)	ווירטואליזציה (VM)
אבטחה ובידוד	בידוד מלא בין שירותים	בידוד חלקי לוגי בין קונטיינרים	בידוד חלקי לוגי עם שיתוף משאבי חומרה
משאבים	ניצול מלא של החומרה עבור שירות בודד	רזה במשאבים ואינו מכיל מערכת הפעלה מלאה	כבד, דורש מערכת הפעלה לכל VM
זמן אתחול	דקות	שניות	דקות
סקלאביליות ואוטומציה	מוגבל, דורש הוספת חומרה פיזית	סקלאביליות ואוטומציה מתקדמת ומהירה	סקלאביליות נמוכה יחסית וניהול ידני
שרידות וגיבוי	אין, תלוי בהוספת חומרה נוספת	שרידות וזמינות גבוהה	שרידות גבוהה אך התאוששות וזמן תגובה ארוך
התאמה לשימוש	שירותים עם דרישות עיבוד גבוהות במיוחד	אפליקציות מודרניות ומודלריות לרבות CNF	שירותים הנדרשים בגמישות אך בזמינות נמוכה

4.4.2. תהליך בחירת תשתית עבודה – הצגת החלופות והניתוח

תוכנת סימולציה - שימוש בתוכנת סימולציה היה מאפשר לנו לדמות את ההתנהגות של הנתבים בסביבה וירטואלית. המדמה תשתית ענן כמו container-lab או EVE-NG, אשר הייתה מספקת הדמיה להתנהגות הנתבים.

יתרונות: כלי הסימולציה מאפשרים ניתוח ראשוני של התנהגות הפרוטוקולים והבדיקות, ללא צורך בהשקעה במשאבים פיזיים או תשתיות ענן.

חסרונות: למרות שמדובר בפתרון זול ונוח, הסימולציה אינה מספקת ביצועים ומדידות מציאותיות. היא לא מסוגלת לשקף באופן מלא את ההתנהגות של תשתית רשת אמיתית

במיקומים גיאוגרפיים שונים, ולא ניתן להפיק ממנה נתונים מדויקים על איכות הרשת בסביבות מגוונות. בנוסף, יש מגבלות ברמת הריאליזם של עומסי תעבורה ותגובות בין אתרים מרוחקים, שמדמות רק באופן חלקי את תשתית הרשת של ספקי תקשורת.

הקמת תשתית עננית פרטית (OnPrem) - על שרתים פיזיים במעבדה מאובטחת עם התקנת מערכת ההפעלה ופלטפורמות ענן להרצת הנתבים אשר הייתה מציעה גמישות פנימית אך דרשה משאבים רבים ולא אפשרה גישה לעבודה מרחוק על גבי האינטרנט. בנוסף, לא אפשרה פריסה במקומות פיזיים אמיתיים בעלי מרחק גיאוגרפי ברחבי העולם לשם מדידות איכות הרשת.

יתרונות: תשתית עננית פרטית הייתה מאפשרת גמישות פנימית ושליטה מלאה בתשתיות, כולל אפשרות להתאמה אישית לצורכי הפרויקט והפעלת רכיבי ענן והרצת הנתבים על שרתים פיזיים במעבדה מאובטחת.

חסרונות: הקמת תשתית עננית פרטית במעבדה דורשת משאבים רבים, כולל עלויות גבוהות של רכישת חומרה והקמת שרתים, תפעול ואחזקה. כמו כן, פתרון זה אינו מאפשר גישה לעבודה מרחוק באמצעות האינטרנט, דבר שמגביל את יכולת הצוות לעבוד באופן נייד וגמיש. בנוסף, תשתית OnPrem אינה מאפשרת פריסה במקומות גיאוגרפיים שונים בעולם, דבר החיוני למדידת איכות רשת גלובלית והפקת נתונים אמין אודות השפעת המרחק על הביצועים.

שימוש בתשתית ענן ציבורית - דוגמת AWS, Azure או Google Cloud, המאפשרת פריסה אמיתית של תשתיות קונטיינרים והפקת מדידות מבוססות ביצועים. תשתיות אלו פרוסות ברחבי העולם וכך ניתנה האפשרות לממש אתרים של ספק תקשורת במקומות גיאוגרפיים שונים, לצד עבודה וחיבור לסביבה מרחוק ללא צורך בהגעה פיזית למעבדה.

יתרונות: תשתית ענן ציבורית מאפשרת פריסה אמיתית של תשתיות קונטיינרים ונתבים בסביבת ענן מבוזרת, עם אפשרות גישה מרחוק מכל מקום בעולם. פתרון זה אפשר לנו לממש אתרים של ספק תקשורת במיקומים גיאוגרפיים שונים, דבר המספק מענה מושלם לביצוע מדידות רשת באתרים מרוחקים והפקת מדידות איכותיות. בנוסף, הפלטפורמות של AWS, Azure ו-Google Cloud מציעות ניהול קל של משאבים, סקלרוביליות, אוטומציה ואבטחה ברמה גבוהה.

חסרונות: העלות של שירותי ענן ציבורי יכולה להיות גבוהה, במיוחד כאשר יש צורך במשאבים רבים לאורך זמן. יחד עם זאת, בשל העובדה שהתשתיות מנוהלות ומנוטרות על ידי ספקי הענן, יש צורך בתלות בספקי צד שלישי בנוגע לניהול האבטחה ולזמינות השרתים.

4.4.3. הפלטפורמה שנבחרה להרצת הפרויקט – AWS שירותי ענן

לאחר בחינה מעמיקה של כל אחת מהאפשרויות, בחרנו להשתמש בפתרון של תשתית ענן ציבורית עם ספק AWS. הבחירה נעשתה לאור היתרונות הרבים שמספקת AWS: פריסה גיאוגרפית רחבה, יכולת גישה מרחוק, ניהול מתקדם של קונטיינרים באמצעות Kubernetes, גמישות והתרחבות מהירה בהתאם לצרכים. הבחירה בתשתית ענן ציבורית אפשרה לנו להימנע מהצורך

להקים תשתית פיזית במעבדה, ולהתרכז באפשרויות הווירטואליזציה וביישום הרשת במסגרת הענן, תוך חיסכון במשאבים ובזמן.

בנוסף, באמצעות AWS התאפשר לנו לבצע את כל הבדיקות והמדידות הנדרשות בפרויקט תוך שימוש בתשתיות גלובליות אמיתיות, דבר שאינו אפשרי בתשתית סימולציה או תשתית עננית פרטית.

IAM (Identity and Access Management)

מאפשר לנהל בצורה מאובטחת גישה לשירותים ולמשאבים בAWS

IAM הינו מנגנון לניהול הרשאות וגישה, אשר מגדיר מי רשאי לגשת לכל אחד מהשירותים שבמערכת – כולל EKS, EC2 ושירותי רשת נוספים. בשירות זה ניתן להגדיר משתמשים, קבוצות ותפקידים, להעניק הרשאות ולהגדיר מדיניות גישה. השירות מאפשר שליטה מדויקת על מי יכול לגשת לאיזה משאב ובאילו תנאים.

תכונות עיקריות של : IAM

- (1) **ניהול משתמשים** : יצירת משתמשים בודדים עבור כל משתמש בארגון והגדרת הרשאות גישה פרטניות לפי הצורך.
- (2) **קבוצות (Groups)** : יצירת קבוצות משתמשים, כך שניתן להחיל מדיניות אחידה לקבוצה שלמה, כמו צוותי פיתוח או צוותי תמיכה.
- (3) **מדיניות (Policies)** : הגדרה של אילו פעולות מותרות ואילו משאבים ניתנים לגישה. מדיניות מאפשרת קביעת כללים מותאמים אישית לכל משתמש, קבוצה או תפקיד.
- (4) **תפקידים (Roles)** : תפקידים מאפשרים הענקת גישה למשאבים על בסיס זמני או מוגדר מראש, שימושי במיוחד למערכות חיצוניות או לשירותים בתוך AWS שנדרשים להרשאות.

Amazon S3

בפרויקט זה השתמשנו ב Amazon S3 שהוא שירות אחסון ענן אמין ונגיש, שמאפשר לשמור ולשלוח נתונים באופן מאובטח ובעל קיבולת גבוהה. בפרויקט זה הוא משמש כמיקום אחסון לתבניות CloudFormation שכוללות את הגדרות התשתית השונות הדרושות לפריסה. שימוש זה מאפשר ניהול גמיש ואוטומטי של משאבים על ידי הפעלת תבניות CloudFormation מכתובת URL ישירה.

התבניות מאוחסנות בפורמט YAML או JSON ומכילות את ההגדרות הדרושות ליצירת התשתיות, כמו תתי-רשתות, VPC, Security Groups, ועוד.

תבנית CloudFormation ליצירת VPC כוללת את ההגדרות הבאות :

תתי-רשתות ציבוריות ופרטיות Public / Private Subnets

הגדרות Internet Gateway עבור גישה לאינטרנט

NAT Gateway שמאפשר גישה חיצונית לתתי-רשתות פרטיות

Amazon ECR (Elastic Container Registry)

שירות של AWS לאחסון, ניהול, ושיתוף של Container Images מספק מאגרים (Registry) מאובטחים שבהם ניתן לשמור Images ולהשתמש בהן לפריסת יישומים בסביבות מבוססות קונטיינרים, כמו Amazon EKS.

הוא נועד במיוחד לאפשר משיכות (pull) של Container Images

שירות זה ניתן לשילוב עם שירות ה IAM להגדרת הרשאות גישה לכל repository ו images המאוחסנים בו. ניתן להגדיר הרשאות גישה ליחידים, צוותים, ושירותים שונים.

אבטחת מידע בשירות זה :

ECR כולל אפשרות לבצע **סריקות אבטחה אוטומטיות** לתמונות המאוחסנות, לזיהוי חולשות אבטחה שעלולות להשפיע על היישומים המופעלים בקונטיינרים.

התמונות מוצפנות במנוחה (encryption at rest) באמצעות שירות ההצפנה של AWS ומוצפנות במהלך ההעברה (encryption in transit).

VPC (Virtual Private Cloud)

הוא שירות של Amazon Web Services (AWS) המאפשר ליצור רשת וירטואלית פרטית ומבודדת בענן, שבה ניתן לנהל את כל תשתיות הרשת, כולל כתובות IP, תתי-רשתות, שערי ניתוב וחומות אש. VPC מספק לארגונים את היכולת לשלוט באופן מלא על תעבורת הרשת שלהם, לנהל גישה פנימית וחיצונית, ולהגדיר את מבנה הרשת לפי הצרכים הספציפיים של הפרויקט.

EKS

EKS (Elastic Kubernetes Service) הוא שירות מנוהל של AWS שמאפשר להקים ולנהל אשכולות Kubernetes בענן.

הוא מנהל את רכיבי ה-Control Plane של Kubernetes, כגון תזמון קונטיינרים, ניהול משאבים, וביצוע עדכונים ותחזוקה, כך שהמשתמשים לא צריכים לנהל אותם בעצמם.

Worker Nodes

Worker Nodes הם מכונות EC2 שמשמשות להרצת קונטיינרים בפועל בתוך אשכול Kubernetes. ה-Worker Nodes מקבלים הוראות מה ה-Control Plane ומבצעים את משימות הרצת הפודים – (Pods) – קבוצות של קונטיינרים שמריצות יישומים.

הקשר בין EKS ל-Worker Nodes

- EKS מספק את ה-Control Plane שמנהל את האשכול כולו ומחליט על חלוקת המשאבים.
- Worker Nodes מתחברים ל-Control Plane של EKS ומבצעים את המשימות בפועל על פי הוראותיו.
- ה-Control Plane שולח פודים להרצה ב-Worker Nodes, אשר מספקים את כוח העיבוד והזיכרון עבורם.
- יחד, EKS ו-Worker Nodes יוצרים אשכול Kubernetes מנוהל בענן, שבו ה-Control Plane מנוהל על ידי AWS, והמשאבים מנוהלים על ידי ה-Worker Nodes.

5. פרק 4 : תכנון ומימוש הפרויקט

הקמת התשתית בפרויקט זה התבצעה בשני חלקים : תחילה תוכננה והוקמה סביבת הענן המארחת ב-AWS, אשר מהווה את התשתית הפיזית והווירטואלית שעליה תרוץ הרשת. לאחר מכן, על גבי תשתית זו, נבנתה ארכיטקטורת הרשת של ספק שירותי האינטרנט (ISP) מבוססת CNF, הכוללת רכיבים וירטואליים ותצורה מתקדמת, שמטרתה לספק שירותים מותאמים, יעילים ומאובטחים ללקוחות.

הארכיטקטורה הרשתית כוללת שימוש בפרוטוקולים מתקדמים דוגמת GRE, IS-IS, VRF, ו-L3VPN, ומאפשרת ניתוב גמיש ומאובטח בין אזורים גיאוגרפיים שונים. הארכיטקטורה מספקת בידוד מלא של תעבורת המידע בין הלקוחות, יחד עם יכולת הרחבה מיידית לכל שינוי בדרישות הרשת.

בפרקים הבאים נציג תחילה את תכנון הארכיטקטורה הרשתית, פריסת רכיבי ה-CNF, וביצוע המדידות ההנדסיות לאימות ביצועי הרשת והשרידות שלה. לאחר מכן נתאר את תכנון סביבת הענן המארחת ב-AWS, הכולל את ההיבטים התשתיתיים הנדרשים לתמיכה במערך רשת מתקדם זה.

4.5. תכנון ומימוש ארכיטקטורת רשת ISP מבוססת טכנולוגיות

CNF הכוללות מדידות

4.5.1. תכנון ומימוש ארכיטקטורת רשת לספקי ISP בטכנולוגיה מודרנית

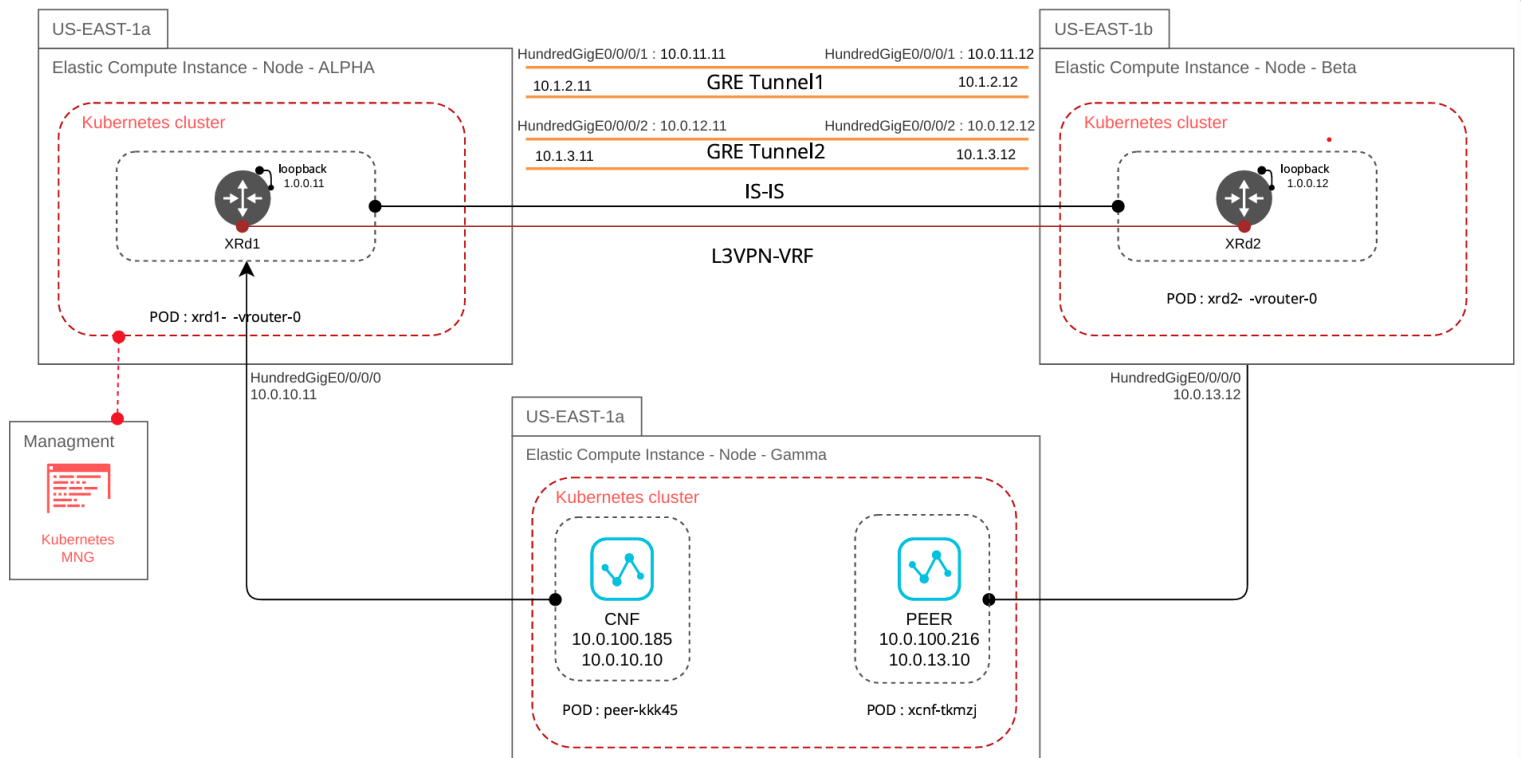
בפרויקט זה מתוארת ארכיטקטורת רשת וירטואלית מתקדמת המבוססת על סביבת ענן וטכנולוגיות מודרניות המגדירות את הדור החדש של טכנולוגיות NFV והווירטואליזציה של הרשתות.

הארכיטקטורה פותחה עבור ספקי שירותי אינטרנט (ISP) וכוללת שימוש בפרוטוקולים מתקדמים כגון GRE, IS-IS, VRF, ו-L3VPN ומציעה גמישות מירבית ואבטחה ללקוחות, תוך בידוד מלא של תעבורת המידע בין הלקוחות שונים. ספקי ה-ISP יכולים להשתמש בארכיטקטורה זו כדי להציע שירותים מותאמים אישית כגון רשתות VPN פרטיות, תעבורה מאובטחת, והפרדת משאבים בצורה לוגית. במידה ולקוח מצטרף או מתנתק, המערכת ניתנת להרחבה מיידית, ללא צורך בשינויים ברמת החומרה או זמן השבתה, מה שמקל על ניהול והפעלה.

ביישום המודרני, כל רכיב בתצורת הרשת ממומש על פלטפורמות ענן דוגמת AWS. עם שילוב יכולות מתקדמות של תשתיות וירטואליזציה ו-Kubernetes המאפשרות עמידות גבוהה, גמישות בפריסה וניהול השירותים, בניית יכולות שרידות וגיבוי, ביצועים גבוהים, מדרגיות וניהול פשוט של מחזור החיים של הרכיב. השימוש ב-Cisco XRd בתור Vrouter מתקדם המבוסס Micro-services ומותאם לסביבות ענן גמישות. המוצר בשלבי פיתוח מוקדמים וקיבלנו את הרישיון והתוכנה להפעלה שלו בשיתוף פעולה עם חברת Cisco.

לאחר הקמת תשתית IT מלאה בענן לטובת הרצת סביבת הרשת, הקמנו רכיבי ניתוב מבוססי ענן (CNFs) ב-2 אזורים גאוגרפיים שונים ברחבי ארצות הברית (US EAST) בעזרת תשתית מחשוב של ספקית הענן AMAZON. הרכיבים מדמים 2 אתרים של ספק תקשורת ISP המחוברים ביניהם בארכיטקטורה לוגית ופיזית המקשרת בין לקוחות בצורה יעילה מתקדמת ומאובטחת.

הארכיטקטורה מורכבת משלושה אזורים מרכזיים: Alpha ו-Beta עם נתבים וירטואליים



(XRd-2 ו-XRd-1) המאפשרים ניתוב תעבורה מאובטח וגמיש. אזור Gamma כולל יחידות Alpine Linux המשמשות כ-CNF ו-Peer, ומאפשר בדיקות סימולציה ושליטה על תפקוד הרשת.

4.5.1.1. רכיבי המערכת:

XRd1 – נתב תוכנתי מבית CISCO

XRd-1 הוא נתב וירטואלי הממוקם באזור Alpha המתאר את אתר ספק התקשורת A ומשמש כקצה ראשון של ה GRE TUNNEL ניהול ניתוב בעזרת IS-IS, ומימוש L3VPN לצורך יצירת הפרדה בין לקוחות שונים.

הנתב הנתב מותקן על גבי שרת EC2 בתצורת M5 המספקת ביצועים חזקים ויכולת ניהול עומסים לניהול תעבורה רשתית.

הנתב נפרס בתור אפליקציה מודרנית Cloud Native גמישה ומהירה המותאמת לריצה בסביבות ענן בתור קונטיינרים (POD).

כתובות IP :

כתובת ה Loopback : 1.0.0.11/32

משמשת ככתובת לולאה פנימית לנתב XRd-1, כתובת זו משמשת עבור תהליכים פנימיים, ניהול וניתוב באמצעות פרוטוקול IS-IS.

כתובת ה Tunnel-ip1 : 10.1.2.11/24

מקור ה-Tunnel ב-XRd-1 בכתובת 10.1.2.11 והיעד בצד XRd-2 הוא 10.1.2.12 ה GRE Tunnel מספק קישור עיקרי להעברת נתונים בין שני הנתבים דרך.

כתובת ה Tunnel-ip2 : 10.1.3.11/24

מקור ה-Tunnel ב-XRd-1 בכתובת 10.0.12.11 והיעד בצד XRd-2 הוא 10.0.12.12 ה Tunnel זה מספק קישור נוסף לשרידות וחלוקת עומס.

2 ממשקי 100G לכיוון הנתב XRd2

10.0.11.11 : HundredGigE0/0/0/1

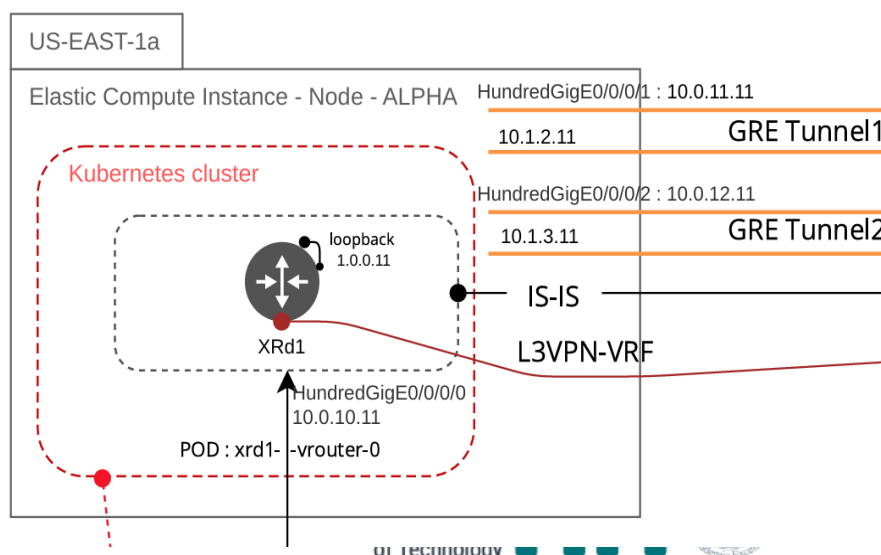
ממשק המחובר ישירות ל XRd-2 באמצעות GRE ומספק נתיב קישור ראשי להעברת תעבורה ברשת בין Alpha ל Beta.

10.0.12.11 : HundredGigE0/0/0/2

ממשק המחובר ישירות ל XRd-2 באמצעות GRE עבור יצירת קישור נוסף עם מטרות שרידות וחלוקת עומסים.

ממשק 100G לכיוון יחידת הבדיקה CNF 10.0.10.11 : HundredGigE0/0/0/0 (ראה מטה)

חיבור פיזי לסביבת Gamma המוקצה עבור רשת ה VRF-בשם "nfs". ממשק זה מאפשר גישה לנקודות הקצה ברשת לטובת בדיקות תעבורה, ביצועים ואבטחה.



XRd2 – נתב תוכנתי מבית CISCO

XRd-2 הוא נתב וירטואלי הממוקם באזור **Beta** המתאר את אתר ספק התקשורת B ומשמש כקצה השני של ה GRE TUNNEL. ניהול ניתוב בעזרת IS-IS, ומימוש L3VPN לצורך יצירת הפרדה בין לקוחות שונים.

הנתב מותקן על גבי שרת EC2 בתצורת M5 המספקת ביצועים חזקים ויכולת ניהול עומסים לניהול תעבורה רשתית.

הנתב נפרס בתור אפליקציה מודרנית Cloud Native גמישה ומחירה המותאמת לריצה בסביבות ענן בתור קונטיינרים (POD).

כתובות IP :

כתובת ה Loopback : 1.0.0.12/32

כתובת לולאה פנימית המשמשת את הנתב XRd-2. משמשת ב-IS-IS עבור ניתוב פנימי, ניהול וניטור.

כתובת ה Tunnel-ip1 : 10.1.2.12/24

מקור ה-Tunnel ב-XRd-2 בכתובת 10.1.2.12 והיעד בצד XRd-1 הוא 10.1.2.11. ה GRE Tunnel מספק קישור עיקרי להעברת נתונים בין שני הנתבים דרך.

כתובת ה Tunnel-ip2 : 10.1.3.12/24

מקור ה-Tunnel ב-XRd-2 בכתובת 10.1.3.12 והיעד בצד XRd-2 הוא 10.1.3.11. ה Tunnel זה מספק קישור נוסף לשרידות וחלוקת עומס.

2 ממשקי 100G לכיוון הנתב XRd2

HundredGigE0/0/0/1 : 10.0.11.12

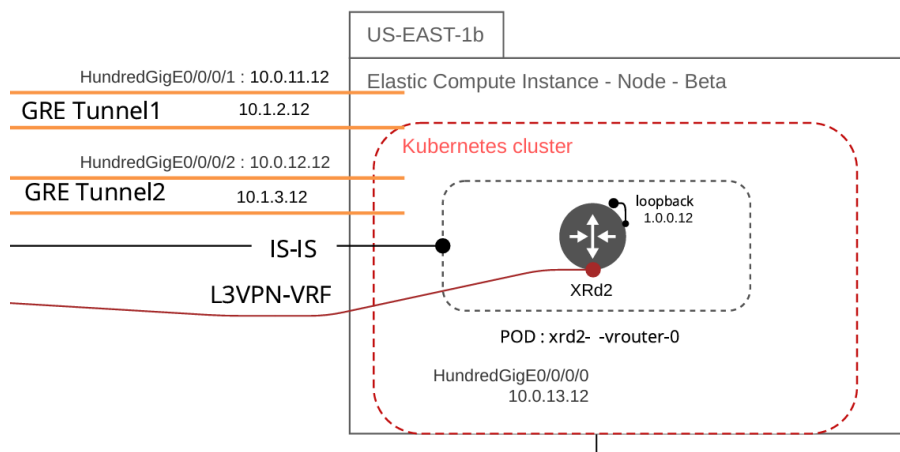
ממשק המחובר ישירות ל XRd-2 באמצעות GRE ומספק נתיב קישור ראשי להעברת תעבורה ברשת בין Alpha ל Beta.

HundredGigE0/0/0/2 : 10.0.12.12

ממשק המחובר ישירות ל XRd-2 באמצעות GRE עבור יצירת קישור נוסף עם מטרות שרידות וחלוקת עומסים.

ממשק 100G לכיוון יחידת הבדיקה CNF 10.0.13.12 : HundredGigE0/0/0/0 (ראה מטה)

חיבור פיזי לסביבת Gamma המוקצה עבור רשת ה VRF-בשם "nfs". ממשק זה מאפשר גישה לנקודות הקצה ברשת לטובת בדיקות תעבורה, ביצועים ואבטחה



התקני קצה למדידות ובדיקות

התקני קצה עם יכולות רשת לבדיקות ומדידות

באזור **Gamma** בענן מיקמנו פונקציות רשת (Network Functions) לצורכי בדיקות תעבורה והערכת הביצועים של הרשת. מכיל שתי יחידות Alpine Linux הפועלות כ-CNF, פודים המריצים מערכת הפעלה של לינוקס המכילים ממשקי רשת ותקשורת ומאפשרים להריץ Tests.

התקן CNF – התקן ראשון לצורכי בדיקות תעבורה והערכת הביצועים של הרשת המבוסס קונטיינר Alpine Linux ופועל כ-CNF (Cloud Network Function). התקן זה מחובר לנתב XRd1 ובעזרתו נבצע בדיקות.

כתובות IP:

eth0: 10.0.100.185/32 כתובת

net1: 10.0.10.10/24 כתובת

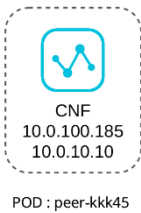
1. **התקן PEER** – התקן שני נוסף לצורכי בדיקות תעבורה והערכת הביצועים של הרשת המבוסס קונטיינר Alpine Linux ופועל כ-CNF (Cloud Network Function). התקן זה מחובר לנתב XRd2 ובעזרתו נבצע בדיקות מול התקן CNF.

כתובות IP:

eth0: 10.0.100.216/32 כתובת

net1: 10.0.13.10/24 כתובת

כל אחד מהאזורים הללו פועל על גבי שרתי EC2 מסוג M5 באמזון AWS, ומספק תשתית וירטואלית מתקדמת לניהול תעבורת רשת בין אזורים שונים ברשת ה-ISP.



4.5.1.2. USE CASE : ארכיטקטורה של ספק תקשורת ISP ב 2 מיקומים

גיאוגרפיה שונים

במקרה של ספק שירותי אינטרנט (ISP), ארכיטקטורה זו מתארת 2 אתרים של ספק אינטרנט המיועדים להנגשת שירותי תקשורת ללקוחות בפריסה גיאוגרפית גלובלית. הספק יכול להקים את האתרים שלו בקרבה פיזית ללקוח שיוכל להנות משיהוי נמוך וביצועים גבוהים, תשתית הענן מספקת לספק התקשורת את החופש והגמישות להקים את האתרים שלו בכל נקודת POP (point-of-presence) של ספק הענן הפרוסים ברחבי העולם.

ב USE CASE זה מתוארת ארכיטקטורת רשת נפוצה המנגישה שירות L3VPN ללקוח בעל 2 סניפים, שירות זה מונגש בארכ' לוגית המשלבת פרוטוקולי ניתוב דינמיים ISIS, טכנולוגיית MPLS/Segment-Routing וקישוריות GRE המאפשרת ליצור זוג מינהרות לוגיות (כניסה אחת ויציאה אחת על גבי רשת מרובת אלמנטים) המאפשרת גמישות מירבית וחלוקה מבודדת של משאבים ללקוחות השונים.

אל אתרי הספק ISP שהקמנו חיברנו "לקוחות" המתוארות כהתקני קצה איתם ביצענו מדידות של ביצועי הרשת. כלל האלמנטים שארכיטקטורה הוקמו בצורה אמיתית (מצורפת מטה קבלת תשלום על שימוש במשאבי מחשוב של חברת אמזון) והוכחו יכולת כי ניתן לממש ארכ' מתקדמת ומודרנית מבוססת קונטיינרים ושירותי תקשורת מבוססים טכנולוגיות CNF.

גמישות מירבית בחלוקת משאבים: השימוש בתשתית בצורה וירטואלית בענן AWS לצד השילוב עם VRF ו-L3VPN, מאפשר לספק שירותים מותאמים אישית בקלות ולשנות את הקצאת המשאבים לפי דרישה באופן מהיר ודינמי.

הפרדה ואבטחת תעבורה בין לקוחות: פרוטוקולי VRF ו-L3VPN מבטיחים שכל לקוח יקבל רשת פרטית ומבודדת, מה שמונע ערבוב בין תעבורת לקוחות ומעניק רמת אבטחה גבוהה בתוך סביבת הרשת המשותפת.

שרידות ואמינות (Redundancy): השימוש בשני GRE Tunnels בין XRd-1 ו-XRd-2 מאפשר המשכיות אוטומטית של תעבורה במקרה של כשל באחד ה-Tunnels, ובכך מונע השבתות ומשפר את רמת השרידות של הרשת. בנוסף, אנו מייצרים שרידות ועמידות לנתבים המבטיחים התאוששות מהירה בעת כשל.

ניהול ניתוב יעיל ואיזון עומסים (Load Balancing): פרוטוקול IS-IS מבצע ניתוב דינמי בהתאם לעומסי התעבורה הקיימים בכל Tunnel, ומאפשר שימוש אופטימלי בשני הנתבים במקביל או מעבר אוטומטי לנתיב חלופי במקרה של עומס או תקלה.

הרחבה וגמישות (Scalability): הארכיטקטורה מבוססת הענן מאפשרת להוסיף רכיבי רשת חדשים או להרחיב את הקיימים בקלות, מה שמאפשר להרחיב את השירות במהירות בהתאם לצרכים המשתנים של הלקוחות וללא הגבלה פיזית.

4.5.1.3. הארכיטקטורה הלוגית של המערכת

הארכיטקטורה מתבססת על מספר פרוטוקולים מרכזיים שמאפשרים לה לבצע ניתוב יעיל, מאובטח וגמיש בתצורה מבוצרת. הפרוטוקולים שבשימוש כוללים:

IS-IS (Intermediate System to Intermediate System)

IS-IS הוא פרוטוקול ניתוב דינמי (Dynamic Routing Protocol) הפועל ברמת שכבה 3 (Layer 3) של מודל OSI. תפקידו המרכזי הוא לאפשר ניתוב אופטימלי של תעבורה בין הנתבים XRd-1 ו-XRd-2. הפרוטוקול מוגדר ברמת Level-2 בלבד, ומספק ניהול ניתוב אוטומטי על גבי שני ה-GRE Tunnels.

GRE (Generic Routing Encapsulation)

פרוטוקול GRE הוא פרוטוקול עטיפה (Encapsulation) המאפשר יצירת מנהרה וירטואלית המובילה תעבורת IP בצורה שקופה על גבי רשתות מרוחקות. בפרויקט זה, GRE משמש ליצירת שני מנהרות (Tunnels) בין הנתבים XRd-1 באזור Alpha ו-XRd-2 באזור Beta. כל מנהרה מייצגת צינור תקשורת וירטואלי המאפשר נידוד תעבורה בצורה מאובטחת ואפקטיבית, תוך כדי שמירה על גמישות בניהול וחלוקת הניתוב בין האזורים.

מימוש GRE בפרויקט

שימוש בשני GRE Tunnels (tunnel-ip1 ו-tunnel-ip2) בין הנתבים XRd-1 ו-XRd-2 מאפשרת מימוש של עמידות (Redundancy) ושרידות (Resiliency), ומספקת תמיכה מלאה בחלוקת עומסים (Load Balancing) ע"י יצירת זוג "מנהרות" לוגיות בין הנתבים הפועלות במקביל.

Tunnel 1 : כתובת מקור בממשק בנתב ב-XRd-1 היא 10.0.11.11, ומיועדת לכתובת יעד 10.0.11.12 ב-XRd-2.

כתובת IP פנימית של ה-TUNNEL : 10.1.2.11 בצד XRd-1 ו-10.1.2.12 בצד XRd-2. ממשק זה נועד לתמוך בתעבורה העיקרית בין הנתבים.

Tunnel 2 : כתובת מקור בממשק בנתב ב-XRd-1 היא 10.0.12.11 וכתובת יעד 10.0.12.12 ב-XRd-2.

כתובת IP פנימית של ה-TUNNEL : 10.1.3.11 בצד XRd-1 ו-10.1.3.12 בצד XRd-2. ממשק זה מספק שרידות (Redundancy) ומשמש כגיבוי.

כל Tunnel פועל עצמאית, כך שהמנהרה הראשונה (tunnel-ip1) משמשת כנתיב עיקרי להעברת תעבורה, בעוד המנהרה השנייה (tunnel-ip2) משמשת כנתיב גיבוי.

עמידות ושרידות: במקרה של כשל במנהרה הראשונה, IS-IS מעביר את התעבורה אוטומטית למנהרה השנייה, וכך נמנעת השבתה.

חלוקת עומסים: כאשר שתי המנהרות זמינות, ניתן לפצל את התעבורה ביניהן, מה שמאפשר לרשת להתמודד עם עומסים גבוהים בצורה יעילה.

תפעול ואבטחה: השימוש בכתובות IP פנימיות עבור כל Tunnel מהווה רובד נוסף של הפרדה בין התשתית של הספק ללקוח המשתמש בה.

VRF (Virtual Routing and Forwarding) ו-L3VPN

כל אחד מהפרוטוקולים ממלא תפקיד קריטי בתשתית הרשת, ומספק פתרונות ברמות שונות של אבטחה, חלוקת עומסים, ניתוב ועמידות.

Virtual Routing and Forwarding (VRF) היא טכנולוגיית ניתוב המאפשרת הפרדת טבלאות ניתוב (Routing Tables) בתוך מכשיר רשת בודד, ובכך מספקת אפשרות ליצירת רשתות

לוגיות מבודדות על גבי אותה תשתית פיזית. למעשה, כל VRF מייצג "רשת פרטית וירטואלית" המוקצית ללקוח או ליישום מסוים, תוך שמירה על תעבורת מידע מבודדת ובטוחה. VRF מסייע להבטיח שבנתב יחיד, תעבורת רשת של לקוחות שונים תנותב ללא הפרעה זה לזה, באופן התורם לביצועי הרשת ולמניעת זליגת מידע בין רשתות.

מימוש VRF בפרויקט

בפרויקט זה ה-VRF מוגדר בשם "nfs" ונמצא בשימוש על גבי ממשקים מסוימים בנתבים XRd-1 ו-XRd-2, כגון ממשקי GRE Tunnels וממשקי Ethernet מרכזיים (HundredGigE0/0/0/0) ו-HundredGigE0/0/0/1. כתוצאה מכך, כל תעבורה המגיעה לממשקים אלו מנותבת דרך VRF "nfs" ומבודדת משאר התעבורה ברשת. הפרדה זו מאפשרת לתעבורת הרשת לעבור בצורה מאובטחת ובלתי תלויה בין רשתות הלקוחות.

הבחירה ב-VRF תואמת את הצורך במימוש מספר רשתות ללקוחות כמו אצל ספקי התקשורת (ISP), שם יש צורך בהפרדה ברמת הניתוב עבור כל לקוח, כדי למנוע השפעה הדדית ולהבטיח פרטיות.

בפרויקט זה, התצורה תומכת בבידוד התעבורה הנכנסת והיוצאת על גבי ה-GRE Tunnels בין הנתבים, ומשמשת כבסיס לאבטחה ולהפרדה לוגית ברשת.

יתרונות השימוש ב-VRF בפרויקט

בידוד תעבורה: כל תעבורה של לקוח מנותבת בטבלת ניתוב נפרדת, מה שמונע אפשרות זליגה לרשתות אחרות.

גמישות ניהולית: VRF מאפשר להוסיף רשתות חדשות ללקוחות נוספים ללא צורך בשינוי פיזי במבנה הרשת של הספק.

L3VPN (Layer 3 Virtual Private Network)

L3VPN הוא שירות ניתוב ברמת שכבה 3 המיועד לספק הפרדה וירטואלית בין רשתות של לקוחות שונים. טכנולוגיית L3VPN, המבוססת על ניתוב ב-VRF ומעבר על גבי MPLS, מאפשרת יצירת רשת פרטית מאובטחת המנוהלת על גבי אותה תשתית פיזית. באמצעות L3VPN ניתן להקים רשתות וירטואליות המופרדות אחת מהשנייה ובכך לשמור על פרטיות התעבורה ולהגביר את אבטחת הרשת.

בפרויקט הנוכחי, L3VPN ממומש בעזרת "nfs" VRF כדי לתמוך בתעבורת לקוחות פרטית המועברת על גבי ה-GRE Tunnels המחברים בין XRd-1 ל-XRd-2. כל תעבורה המועברת ב-VPN מנותבת בתוך ה-VRF, ומאפשרת לרשת להישאר מבודדת ומאובטחת. בפרויקט זה, L3VPN מהווה שכבת אבטחה נוספת המגנה על התעבורה בין אזורי הרשת Alpha ו-Beta, כך שתהיה מבודדת, מוגנת ונגישה אך ורק לנקודות הקצה הרלוונטיות.

4.5.1.4. מימוש הארכיטקטורה וביצוע בדיקות תקינות ומדידות הנדסיות

הקדמה

המערכת בפרויקט כוללת שני נתבים וירטואליים, XRd-1 ו-XRd-2, אשר מחוברים באמצעות GRE Tunnel ומפעילים פרוטוקולי ניתוב מתקדמים כדי להבטיח תקשורת אמינה ויעילה. בנוסף, ישנם שני התקני קצה CNF, ו- Peer הרצים בתור PODS ומדמים את תעבורת הרשת ומאפשרים בדיקות ביצועים שונות. המטרה העיקרית היא לבצע בדיקות איכות לרשת ולמדוד פרמטרים קריטיים כגון אובדן חבילות (Packet Loss), זמן סבב לפקטה (Round Trip Time), שיהוי (Latency), רוחב פס (Throughput), ו-שיעור שגיאות (Bit Error Rate - BER).

- קונפיגורציות מלאות של הנתבים מצורפים מטה בתור קבצים דיגיטליים תהליך הגישה אל הנתבים והתקני הקצה והרצת הבדיקות למדידת איכות הרשת

1. הכנה והתחברות אל קלאסטר הקוברנטיס (EKS) ב-AWS:

```
aws eks update-kubeconfig --name xrd-terraform-2d601e34 --region us-east-1
```

```
kubectl get svc
```

פלט :

```
root@DESKTOP-KAUMP6H:~/xrd-eks/xrd-terraform# kubectl get svc
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes           ClusterIP     172.20.0.1    <none>         443/TCP    31m
```

2. הצגת כל הפודים במערכת :

```
kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
cnf-b88f4bdd5-tkmzj	1/1	Running	0	25m	10.0.100.185	ip-10-0-100-13.ec2.internal	<none>	<none>
peer-76d759b69c-kkk45	1/1	Running	0	25m	10.0.100.216	ip-10-0-100-13.ec2.internal	<none>	<none>
xrd1-xrd-vrouter-0	1/1	Running	0	25m	10.0.100.196	ip-10-0-100-11.ec2.internal	<none>	<none>
xrd2-xrd-vrouter-0	1/1	Running	0	25m	10.0.100.224	ip-10-0-100-12.ec2.internal	<none>	<none>

כל הפודים במערכת (כולל CNF, XRd-1, XRd-2, ו-Peer) צריכים להיות במצב "Running". הטבלה שתקבל צריכה לכלול את כתובות ה-IP של כל פוד, אשר ישמשו אותנו בבדיקות הקישוריות הבאות.

3. התחברות אל ממשק ה-CLI לקינפוג בתוך הנתבים הווירטואליים והתחברות עם שם משתמש וסיסמא לחיבור מאובטח:

```
Kubectl exec -it xrd1-xrd-vrouter-0 -- xr
```

```
root@DESKTOP-KAUMP6H:~/xrd-eks/xrd-terraform# kubectl exec -it xrd1-xrd-vrouter-0 -- xr
User Access Verification
Username: cisco
Password:
RP/0/RP0/CPU0:xrd1#
```

```
Kubectl exec -it xrd2-xrd-vrouter-0 -- xr
```

```
root@DESKTOP-KAUMP6H:~/xrd-eks/xrd-terraform# kubectl exec -it xrd2-xrd-vrouter-0 -- xr
User Access Verification
Username: cisco
Password:
RP/0/RP0/CPU0:xrd2#
```

4.5.1.5. ביצוע בדיקות קישוריות ותקינות פרוטוקולים - פרמטרים, כלי בדיקה

ותוצאות רצויות

בדיקת קישוריות בין הנתבים XRd-1 ל-XRd-2

בבדיקה זו נבדוק את הקישוריות בין הנתב XRd-1 לנתב XRd-2 על ידי ביצוע סדרת פינגים לכתובות ה-IP השונות של XRd-2 מממשקי XRd-1. בדיקה זו תוודא את תקינות החיבור והנתבים בין שני הנתבים, כולל ממשקי ה-GRE Tunnel והכתובות הלוגיות והפיזיות.

Ping לכתובת ה-Loopback של XRd-2 :

ping 1.0.0.12

```
RP/0/RP0/CPU0:xrd1#ping 1.0.0.12
Sat Oct 26 20:26:36.773 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.0.0.12 timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

כל החבילות הגיעו ליעדן עם זמן RTT ממוצע של 2ms מה שמעיד על קישוריות תקינה.

Ping לכתובת Tunnel Interface של XRd-2 (tunnel-ip1) :

ping 10.1.2.12

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
RP/0/RP0/CPU0:xrd1#ping 10.1.2.12
Sat Oct 26 20:26:48.719 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.12 timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/3/4 ms
```

החבילות הגיעו בהצלחה עם זמן RTT ממוצע של 3ms, מה שמצביע על תפקוד תקין של ה-GRE Tunnel הראשון.

Ping לכתובת של ממשק HundredGigE0/0/0/1 של XRd-2 :

ping 10.0.11.12

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/3/4 ms
RP/0/RP0/CPU0:xrd1#ping 10.0.11.12
Sat Oct 26 20:26:56.051 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.11.12 timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

Ping לכתובת של ממשק HundredGigE0/0/0/2 של XRd-2 :

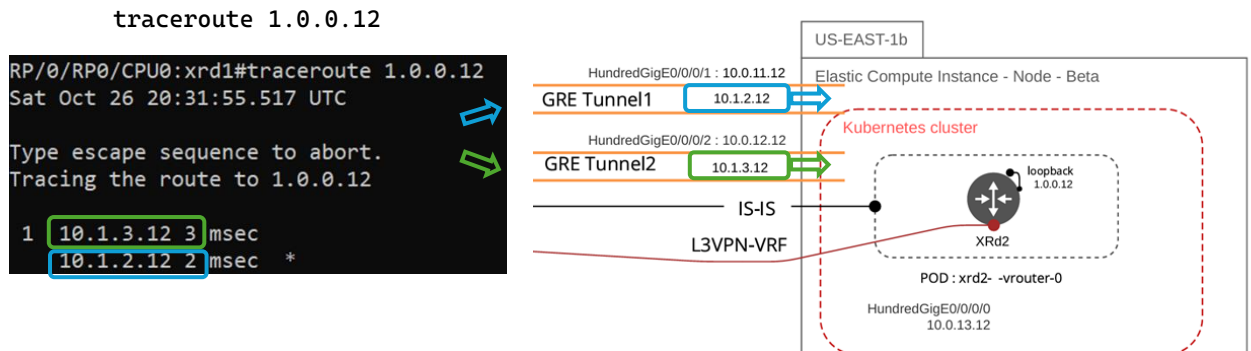
ping 10.0.12.12

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
RP/0/RP0/CPU0:xrd1#ping 10.0.12.12
Sat Oct 26 20:27:03.798 UTC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.12.12 timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 3/3/4 ms
```

כל החבילות הגיעו עם RTT ממוצע של 2/3ms מה שמעיד על קישוריות תקינה דרך 2 הממשקים הפיזיים של XRd-2.

בדיקת הנתבים בהם עברו החבילות והוכחת השימוש בגיבוי וחלוקת עומסים:

מטרת הבדיקה היא לשלוח פקודת traceroute אל הכתובת Loopback של נתב XRd-2 שהיא 1.0.0.12 מהנתב XRd-1. הפקודה הזו תאפשר לעקוב אחרי המסלול שעוברת חבילה מהרגע שהיא יוצאת ממכשיר המקור ועד שהיא מגיעה ליעד. מה שמאפשר לנו לראות את הנתבים שעוברת החבילה, כולל כל התקנים המתווכים בדרך (hops) ואת זמן השיהוי בכל שלב.



שלב ראשון : כתובת ה-IP היא 10.1.3.12, והשיהוי (הזמן שלוקח לחבילה להגיע ל-hop זה) הוא 3 מילישניות (msec). כתובת זו שייכת ל TUNNEL1 אל נתב XRd2.

שלב שני : כתובת ה-IP היא 10.1.2.12, והשיהוי הוא 2 מילישניות. כתובת זו שייכת ל TUNNEL2 אל נתב XRd2.

שני הכתובות מופיעות מכיוון שנמצאו שני נתבים אפשריים שונים שיכולים לשמש להעברת החבילה. הכתובת הראשונה מופיעה כתוצאה מהשימוש ב-Tunnel הראשון, והכתובת השנייה כתוצאה מהשימוש ב-Tunnel השני. השימוש בשני הכתובות 10.1.3.12 ו-10.1.2.12 מדגיש שקיימים שני GRE Tunnels בין XRd-1 ל-XRd-2 ושהם עובדים בצורה תקינה עם השהיות נמוכות. שני הנתבים מצביעים על חלוקת עומס אפשרית ושרידות, כך שבמקרה של כשל באחד הנתבים, החבילה עדיין תוכל להגיע ליעדה דרך הנתב החלופי.

התוצאות מראות קישוריות מלאה ותקינה בין הנתבים XRd-1 ו-XRd-2 בכל הכתובות שנבדקו. ה-RTT הממוצע נמוך ואין אובדן חבילות, מה שמצביע על יציבות ואמינות גבוהה של הקישוריות בין הנתבים. בדיקה זו מאשרת שה-GRE Tunnel פועל כמצופה ושהנתבים מתפקדים כראוי במבנה הרשת. כעת ניתן לעבור לבדיקת תקינות פרוטוקולים ולאחר מכן לבדיקות ביצועים.

4.5.1.6. המשך בדיקות מקדימות – תקינות ממשקים ופרוטוקולים חיוניים לספק התקשורת

בדיקת תצורת הממשקים של הנתב התוכנתי:

נוודא כי כל הממשקים ב-XRd-1 מוגדרים באופן תקין, פעילים, מוגדרים עליהם כתובות כך שיוכלו לנתב תעבורה.

RP/0/RP0/CPU0:xrd1(config)#do show ip interface brief

```
RP/0/RP0/CPU0:xrd1(config)#do show ip interface brief
Sat Oct 26 23:52:33.928 UTC
```

Interface	IP-Address	Status	Protocol	Vrf-Name
Loopback0	1.0.0.11	Up	Up	default
tunnel-ip1	10.1.2.11	Up	Up	default
tunnel-ip2	10.1.3.11	Up	Up	default
HundredGigE0/0/0/0	10.0.10.11	Up	Up	nfs
HundredGigE0/0/0/1	10.0.11.11	Up	Up	default
HundredGigE0/0/0/2	10.0.12.11	Up	Up	default

RP/0/RP0/CPU0:xrd1(config)#

ממשקים
לוגיים

ממשקים
"פיזיים"

VRF של חלקו

כל הממשקים הנדרשים פעילים, מה שמצביע על קישוריות פיזית ולוגית תקינה.

בדיקת שכנות בפרוטוקול IS-IS

נוודא כי הנתבים מזוהים אחד את השני בפרוטוקול הניתוב ISIS בטבלת השכנויות.

RP/0/RP0/CPU0:xrd1#show isis neighbors

```
RP/0/RP0/CPU0:xrd1#show isis neighbors
Sat Oct 26 20:00:31.803 UTC
```

IS-IS 1 neighbors:

System Id	Interface	SNPA	State	Holdtime	Type	IETF-NSF
xrd2	ti1	*PtoP*	Up	27	L2	Capable
xrd2	ti2	*PtoP*	Up	26	L2	Capable

Total neighbor count: 2

אנו רואים שכנויות של הנתבים דרך הממשקי הטאנלים, ב-ti1 ו-ti2 עם מצב L2 (Level-2) המעיד על הפצת עידכונים מלאים בין הנתבים כיוון שמהווים את Backbone של הרשת.

אימות ממשקי MPLS

נוודא כי הגדרות ה-MPLS על המנהרות GRE תקינות כדי שיוכלו ליצור תוויות MPLS להפרדת תעבורה וביצוע העברת חבילות בצורה מהירה.

RP/0/RP0/CPU0:xrd1#show mpls interfaces

```
RP/0/RP0/CPU0:xrd1#show mpls interfaces
Sat Oct 26 23:56:24.213 UTC
```

Interface	LDP	Tunnel	Static	Enabled
tunnel-ip1	No	No	No	Yes
tunnel-ip2	No	No	No	Yes

RP/0/RP0/CPU0:xrd1#

בשתי המנהרות (tunnel-ip1 ו-tunnel-ip2) מופעל MPLS, המאפשר VPNs בשכבה 3 ועטיפת חבילות בתוויות, דבר קריטי להבטחת העברת נתונים מאובטחת בסביבות ISP.

בדיקת טבלת העברות MPLS Forwarding

המטרה של בדיקת טבלת העברות MPLS היא לאמת את התוויות המוקצות ואת הנתיבים להכוונת תעבורה באמצעות המנהרות המוגדרות (tunnel-ip1 ו-tunnel-ip2). תוויות ה-MPLS מאפשרות הכוונת תעבורה בצורה יעילה, תוך שימוש במנגנון ניתוב סגמנטי לניהול עומסים, גיבוי וייעול ביצועים.

RP/0/RP0/CPU0:xrd1#show mpls forwarding

```
RP/0/RP0/CPU0:xrd1#show mpls forwarding
Sun Oct 27 00:00:26.914 UTC
```

Local Label	Outgoing Label	Prefix or ID	Outgoing Interface	Next Hop	Bytes Switched
16011	Aggregate	SR Pfx (idx 11)	default		0
16012	Pop	SR Pfx (idx 12)	ti1	10.1.2.12	22704
	Pop	SR Pfx (idx 12)	ti2	10.1.3.12	0
24000	Aggregate	nfs: Per-VRF Aggr[V] \	nfs		21672

תווית 16012 - מוגדרת כ-Pop ומייצגת Prefix עם אינדקס 12 ומשויכת לשני ממשקים :

ti1 עם כתובת ה-Next Hop של 10.1.2.12 – דרכה עברו 22,704 בתים, מה שמעיד על כך שהיא בשימוש פעיל.

ti2 עם כתובת ה-Next Hop של 10.1.3.12 – אין תעבורה שעברה דרכה, מה שמעיד על כך שהיא נמצאת כנתיב גיבוי או לחלוקת עומסים.

תווית 24000 - מוגדרת כ-"Aggregate" ומייצגת VRF לניהול תעבורת nfs, תחת הממשק nfs. דרכה עברו 21,672 בתים, מה שמעיד על שימוש יציב לתעבורת VRF ספציפית זו.

טבלת ה-MPLS מציגה מערכת ניתוב המבוססת על תוויות לניהול עומסים וגיבוי אוטומטי. המערכת מקצה תווית אחת למנהרות נפרדות (ti1 ו-ti2) לאותו יעד, כאשר ti2 משמשת כממשק גיבוי, המעיד על מדיניות ניתוב אוטומטית לבחירת הנתיב האופטימלי. בנוסף, תווית 24000 מספקת VRF נפרד לתעבורת nfs, המאפשר גמישות וניהול מאובטח לשירותים ייחודיים.

בדיקת פרוטוקול BGP

נועדה לאמת את תקינות החיבור בין נתבים בפרוטוקול BGP, ולהציג את מצב טבלת הניתוב והתקשורת בין השכנים (Neighbors).

RP/0/RP0/CPU0:xrd1#show bgp vpnv4 unicast summary

```
RP/0/RP0/CPU0:xrd1#show bgp vpnv4 unicast summary
Sat Oct 26 20:01:31.986 UTC
BGP router identifier 1.0.0.11, local AS number 100
BGP generic scan interval 60 secs
Non-stop routing is enabled
BGP table state: Active
Table ID: 0x0
BGP table nexthop route policy:
BGP main routing table version 10
BGP NSR Initial initsync version 4 (Reached)
BGP NSR/ISSU Sync-Group versions 0/0
BGP scan interval 60 secs

BGP is operating in STANDALONE mode.
```

Process Speaker	RcvTblVer	bRIB/RIB	LabelVer	ImportVer	SendTblVer	StandbyVer	
	10	10	10	10	10	10	0

Neighbor	Spk	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	St/PfxRcd
1.0.0.12	0	100	59	59	10	0	0	00:55:55	1

השכן 1.0.0.12 נמצא במצב Up כבר 55 דקות ו-55 שניות, מה שמעיד על חיבור יציב. הנתב והשכן מחוברים עם AS 100, ושולחים ומקבלים הודעות BGP בהצלחה. תפקיד ה-BGP (Border Gateway Protocol) הוא לנהל את הניתוב בין רכיבי הרשת באמצעות העברת טבלאות ניתוב בין (AS) Autonomous system שונות ניתוב בצורה דינמית ומבוססת מדיניות.

4.5.1.7. בדיקות – תקינות קישוריות בין התקני קצה PEER ו CNF

הבדיקות מבוצעות באמצעות שני התקני קצה וירטואליים, CNF ו-PEER, אשר פועלים כקונטיינרים על פלטפורמת Kubernetes. התקני קצה אלו מדמים תעבורה מרוחקת (2 אתרים שונים בצפון אמריקה) ומחוברים לנתבים הווירטואליים XRd-1 ו-XRd-2 באמצעות GRE Tunnel ופרוטוקולי ניתוב, דבר שמאפשר סימולציה של תעבורת רשת אמיתית בתנאים מבוקרים.

התקני הקצה והקונפיגורציה שלהם:

:CNF

המערכת מבוססת על פוד המריץ Alpine Linux (גרסה 64.5.10.226-214.880.amzn2.x86_64)

כתובות רשת:

10.0.100.185/32 – `eth0` – כתובת לצורך חיבור לתעבורה חיצונית.

10.0.10.10/24 – `net1` – כתובת פנימית לחיבורי תעבורה מבוססי GRE.

```
/ # uname -a
Linux cnf-b88f4bdd5-tkmzj 5.10.226-214.880.amzn2.x86_64 #1 SMP Tue Oct 8 16:18:15 UTC 2024 x86_64 Linux
/ # ipaddr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: eth0@if11: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 9001 qdisc noqueue state UP
    link/ether 1a:31:25:ea:0a:49 brd ff:ff:ff:ff:ff:ff
    inet 10.0.100.185/32 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::1831:25ff:feea:a49/64 scope link
        valid_lft forever preferred_lft forever
4: net1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 0a:ff:f2:73:a5:0f brd ff:ff:ff:ff:ff:ff
    inet 10.0.10.10/24 brd 10.0.10.255 scope global net1
        valid_lft forever preferred_lft forever
    inet6 fe80::8ff:f2ff:fe73:a50f/64 scope link
        valid_lft forever preferred_lft forever
```

התקן הקצה פועל כשרת ברשת ומאפשר בדיקות תעבורה וניתוח ביצועים. הקונפיגורציה שלו מאפשרת לו להאזין לבקשות של בדיקות רוחב פס מ-PEER ולבצע בדיקות latency, Jitter, ו-Packet Loss.

:PEER

המערכת נוספת מבוססת Alpine Linux (גרסה 5.10.226-214.880.amzn2.x86_64)

כתובות רשת:

eth0: 10.0.100.216/32 – כתובת לצורך חיבור לתעבורה חיצונית.

net1: 10.0.13.10/24 – כתובת פנימית לחיבורי תעבורה מבוססי GRE.

```
Linux peer-76d759b69c-kkk45 5.10.226-214.880.amzn2.x86_64 #1 SMP Tue Oct 8 16:18:15 UTC 2024 x86_64 Linux
/ # ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
3: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 9001 qdisc noqueue state UP
    link/ether 1e:71:06:95:d1:53 brd ff:ff:ff:ff:ff:ff
    inet 10.0.100.216/32 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::1c71:6ff:fe95:d153/64 scope link
        valid_lft forever preferred_lft forever
4: net1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 0a:ff:dd:da:52:6f brd ff:ff:ff:ff:ff:ff
    inet 10.0.13.10/24 brd 10.0.13.255 scope global net1
        valid_lft forever preferred_lft forever
    inet6 fe80::8ff:ddff:feda:526f/64 scope link
        valid_lft forever preferred_lft forever
```

התקן זה פועל כלקוח לרוב הבדיקות ומחובר ל-CNF לצורך מדידות ביצועים. ה-PEER שולח בקשות בדיקות ל-CNF במצב שרת על מנת לאסוף נתונים מדויקים בנוגע לאיכות הקישוריות בין רכיבי הרשת.

תצורת הבדיקות בין CNF ו-PEER

CNF ו-PEER משמשים כבסיס לבדיקות איכות ביצועי הרשת על ידי שימוש בכלים כמו ping ו-iperf3 והקלטת תעבורה עבור בדיקות שונות, כגון, Packet Loss, Round Trip Time (RTT), Latency, Jitter, Throughput, ו-Bit Error Rate (BER). הבדיקות המפורטות להלן נועדו לבחון את איכות החיבור, אמינות העברת הנתונים והעמידות של הרשת תחת עומסים ובתנאים שונים. הבדיקות מתבצעות כך ש-CNF משמש לרוב כנקודת קצה המקבלת את התעבורה, בעוד PEER הוא נקודת הקצה השולחת את התעבורה ויוזמת את הבדיקות.

4.5.1.8. ביצוע בדיקות הנדסיות - פרמטרים, כלי בדיקה, תוצאות רצויות ותוצאות שהתקבלו

1. בדיקת אובדן חבילות (Packet Loss) (מהתקן CNF ל PEER)

מדד המייצג את אחוז החבילות שלא הגיעו ליעדן מתוך כלל החבילות שנשלחו. מדד זה משקף את איכות הרשת והיכולת שלה להתמודד עם עומסים ולשמור על אמינות. תוצאה גבוהה של Packet Loss עשויה להעיד על בעיות באיכות הרשת או בהגדרות תשתית הניתוב.

$$100 \times \frac{\text{Number of lost packets}}{\text{Total packets Sent}} = \text{Packet Loss Rate}$$

כלי מדידה : ping

הפקודה ping היא כלי אבחון המשמש לבדוק קישוריות בין שני התקנים, למדידת זמן תגובה (RTT) ולבדיקת אובדן חבילות. הפקודה שולחת פקטות ICMP בגודל של 1500 בייטים (MTU) ליעד וממתינה לתגובה על כל אחת מהן, מה שמאפשר למדוד את זמן התגובה ואת יציבות הקישוריות.

פקודה :

```
ping -c 30 10.0.13.10
```

פקודה זו שולחת 30 פקטות מ-CNF לכתובת IP של PEER (10.0.13.10), ומודדת את זמן הגעתן לכל פקטה, כמו גם אובדן חבילות, אם קיים.

```
/ # ping -c 30 10.0.13.10
PING 10.0.13.10 (10.0.13.10): 56 data bytes
64 bytes from 10.0.13.10: seq=0 ttl=125 time=0.688 ms
64 bytes from 10.0.13.10: seq=1 ttl=125 time=0.626 ms
```

תוצאה רצויה :

אובדן חבילות (Packet Loss) של פחות מ-1%, המעיד על קישוריות יציבה בין CNF ל-PEER.

תוצאה שהתקבלה :

```
--- 10.0.13.10 ping statistics ---
30 packets transmitted, 30 packets received, 0% packet loss
round-trip min/avg/max = 0.596/0.669/1.963 ms
```

התוצאה מראה אובדן חבילות של 0%, כלומר כל הפקטות שנשלחו הגיעו ליעדן.

$$100 \times \frac{\text{Number of lost packets}(0)}{\text{Total packets Sent}(30)} = \text{Packet Loss Rate} = 0$$

נבצע את הבדיקה גם עבור פאקטות JUMBO-FRAMES

בבדיקה זו נבדקה תמיכת הרשת בחבילות גדולות (Jumbo Frames) בגודל 9000 bytes, זאת בכדי לוודא שיכולת הרשת לתמוך בעומסי תעבורה גדולים נשמרת, ללא בעיות קישוריות.

```
ping -c 30 -s 9000
```

הפקודה:

10.0.100.185

```
/ # ping -c 30 -s 9000 10.0.100.185
PING 10.0.100.185 (10.0.100.185): 9000 data bytes
9008 bytes from 10.0.100.185: seq=0 ttl=126 time=0.067 ms
9008 bytes from 10.0.100.185: seq=1 ttl=126 time=0.081 ms
```

הפקודה שולחת 30 פקטות בגודל 9000 bytes מ-PEER לכתובת ה-IP של CNF (10.0.100.185).

תוצאה שהתקבלה:

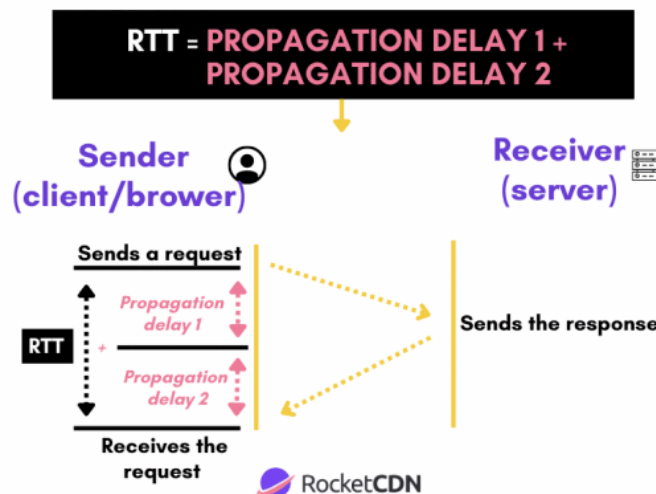
```
--- 10.0.100.185 ping statistics ---
30 packets transmitted, 30 packets received, 0% packet loss
round-trip min/avg/max = 0.063/0.070/0.106 ms
```

הבדיקה מראה שכל החבילות בגודל 9000 bytes הגיעו ליעדן, עם אובדן חבילות של 0% תוצאה זו מצביעה על כך שהרשת תומכת ב-Jumbo Frames בצורה מיטבית, ומאפשרת העברת נתונים בעומס גבוה ביעילות גבוהה.

2. בדיקת Round Trip Time (RTT) (מהתקן PEER ל CNF)

RTT מודד את הזמן שלוקח לחבילה לעבור מהמקור ליעד ולחזור, ומספק תובנות על עיכובים בקישוריות. ערכי RTT נמוכים חשובים במיוחד עבור אפליקציות רגישות לזמן תגובה כמו שיחות וידאו.

$$T_{\text{arrival}} - T_{\text{departure}} = \text{RTT}$$



כלי מדידה : ping

מודד RTT (Round Trip Time) באמצעות שליחת חבילות ICMP Echo Request לכתובת IP מסוימת, וממתין לתגובת Echo Reply מהיעד. זמן ה-RTT, המייצג את משך הזמן הלך ושוב של החבילה, מחושב על ידי מדידת ההפרש בין שליחת החבילה וקבלתה חזרה. כלי זה מספק

נתונים הכוללים ממוצע, ערך מינימלי ומקסימלי של זמן ה-RTT, ואחוז אובדן חבילות אם ישנו, מה שמאפשר להעריך את איכות הקישוריות והיציבות ברשת.

פקודה :

```
ping -c 30 10.0.10.10
```

פקודה זו שולחת 30 פקטות מ-PEER לכתובת IP של CNF (10.0.10.10), ומודדת את זמן הגעתן לכל פקטה, כמו גם אובדן חבילות, אם קיים.

```
Linux peer-76d759b69c-kkk45 5.10.226-214.880.amzn2.x86_64 #1
/ # ping -c 30 10.0.10.10
PING 10.0.10.10 (10.0.10.10): 56 data bytes
64 bytes from 10.0.10.10: seq=0 ttl=125 time=0.706 ms
64 bytes from 10.0.10.10: seq=1 ttl=125 time=0.648 ms
```

תוצאה רצויה :

זמן תגובה (RTT) ממוצע נמוך מ-20 מילישניות נחשב למצוין עבור חיבורים מקומיים, עד 100 מילישניות נחשב לטוב עבור חיבורים בינלאומיים ומצביע על חיבור מהיר ויעיל.

תוצאה שהתקבלה :

```
--- 10.0.10.10 ping statistics ---
30 packets transmitted, 30 packets received, 0% packet loss
round-trip min/avg/max = 0.603/0.631/0.706 ms
```

במידה זו נמדדו 30 חבילות שהועברו והתקבלו כולן, כלומר 0% אובדן חבילות, זמן התגובה ממוצע של 0.631 מילישניות, עם זמן תגובה מינימלי של 0.603 מילישניות ומקסימלי של 0.706 מילישניות. תוצאה זו מעידה על חיבור אמין ומהיר בין PEER ל-CNF, המתאפיין ביציבות ובביצועים גבוהים במיוחד, המאפשרים חוויית משתמש מיטבית גם ביישומים רגישים.

נבצע את הבדיקה גם עבור פאקטות JUMBO-FRAMES

נבדוק את ערך ה RTT בחבילות גדולות (Jumbo Frames) בגודל 9000 bytes, זאת בכדי לוודא שיכולת הרשת לתמוך בעומסי תעבורה גדולים נשמרת, ללא בעיות קישוריות.

```
ping -c 30 -s 9000
```

הפקודה :

10.0.100.185

```
/ # ping -c 30 -s 9000 10.0.100.185
PING 10.0.100.185 (10.0.100.185): 9000 data bytes
9008 bytes from 10.0.100.185: seq=0 ttl=126 time=0.067 ms
9008 bytes from 10.0.100.185: seq=1 ttl=126 time=0.081 ms
```

הפקודה שולחת 30 פקטות בגודל 9000 bytes מ-PEER לכתובת ה-IP של CNF (10.0.100.185).

תוצאה שהתקבלה :

```
--- 10.0.100.185 ping statistics ---
30 packets transmitted, 30 packets received, 0% packet loss
round-trip min/avg/max = 0.063/0.070/0.106 ms
```

בבדיקה הנוכחית נשלחו 30 חבילות בגודל 9000 בתים (Jumbo Frames) לכתובת 10.0.100.185 כולן התקבלו ללא אובדן (0% packet loss).

זמני ה-RTT שנמדדו היו: מינימלי של 0.063 מילישניות, ממוצע של 0.070 מילישניות ומקסימלי של 0.106 מילישניות שהם נמוכים במיוחד. בבדיקה הקודמת, נשלחו 30 חבילות בגודל סטנדרטי של 1500 בתים, עם זמני RTT ממוצעים של 0.631 מילישניות. הבדיקה הנוכחית מציגה זמני RTT נמוכים יותר, למרות גודל החבילות הגדול יותר.

3. בדיקת ערך השיהוי (Latency) (מהתקן CNF ל PEER)

השהיה (Latency) מודדת את הזמן שלוקח לחבילה לעבור מהמקור ליעד, ומספקת תובנות על עיכובים בקישוריות. ערכי השהיה נמוכים חשובים במיוחד עבור אפליקציות רגישות לזמן תגובה כמו שיחות וידאו.

$$\frac{RTT}{2} = \text{Latency}$$

כלי מדידה : ping

הפקודה ping היא כלי אבחון המשמש לבדוק קישוריות בין שני התקנים, למדידת זמן תגובה (RTT) ולבדיקת אובדן חבילות.

פקודה :

```
ping -c 30 10.0.13.10
```

פקודה זו שולחת 30 פקטות מ-CNF לכתובת IP של PEER (10.0.13.10), ומודדת את זמן הגעתן לכל פקטה, כמו גם אובדן חבילות, אם קיים.

```
/ # ping -c 30 10.0.13.10
PING 10.0.13.10 (10.0.13.10): 56 data bytes
64 bytes from 10.0.13.10: seq=0 ttl=125 time=0.688 ms
64 bytes from 10.0.13.10: seq=1 ttl=125 time=0.626 ms
```

תוצאה רצויה :

עבור ספקי תקשורת, השהיה ממוצעת של פחות מ-20 מילישניות נחשבת למצוינת, ומתחת ל-50 מילישניות נחשבת לטובה מאוד. לפי ה-FCC, ספקי אינטרנט נדרשים לספק שיהוי של עד 100 מילישניות ב-95% מהמדידות. [Performance Measures Testing - Universal Service Administrative Company](#)

תוצאה שהתקבלה :

```
--- 10.0.13.10 ping statistics ---
30 packets transmitted, 30 packets received, 0% packet loss
round-trip min/avg/max = 0.596/0.669/1.963 ms
```

חישוב השיהוי :

$$\text{Latency}(\min) = \frac{RTT(\text{avg})}{2} = \frac{0.596\text{ms}}{2} = 0.297\text{ms}$$

$$\text{Latency}(\max) = \frac{RTT(\text{avg})}{2} = \frac{1.963\text{ms}}{2} = 0.981\text{ms}$$

$$\text{Latency}(\text{avg}) = \frac{RTT(\text{avg})}{2} = \frac{0.669\text{ms}}{2} = 0.334\text{ms}$$

נבצע את הבדיקה גם עבור פאקטות JUMBO-FRAMES

נבדוק שיהיו בחבילות גדולות (Jumbo Frames) בגודל 9000 bytes בכדי לוודא שיכולת הרשת לתמוך בעומסי תעבורה גדולים נשמרת.

תוצאה:

```
--- 10.0.100.185 ping statistics ---
30 packets transmitted, 30 packets received, 0% packet loss
round-trip min/avg/max = 0.063/0.070/0.106 ms
```

$$\text{Latency}(\min) = \frac{\text{RTT}(\min)}{2} = \frac{0.063\text{ms}}{2} = 0.0315\text{ms}$$

$$\text{Latency}(\max) = \frac{\text{RTT}(\max)}{2} = \frac{0.106\text{ms}}{2} = 0.053\text{ms}$$

$$\text{Latency}(\text{avg}) = \frac{\text{RTT}(\text{avg})}{2} = \frac{0.07\text{ms}}{2} = 0.035\text{ms}$$

התוצאות מצביעות על רשת בעלת ביצועים גבוהים עם שיהוי נמוך מאוד, העומדת בדרישות ה-FCC לספקי אינטרנט. גם עם חבילות גדולות (Jumbo Frames), הרשת מציגה שיהוי נמוך, מה שמעיד על יכולת להתמודד עם עומסי תעבורה גדולים ביעילות.

4. מדידת רוחב הפס (Throughput) תעבורת TCP (מהתקן CNF ל PEER)

בבדיקה זו נמדדה מהירות רוחב הפס בין התקני הקצה CNF ו PEER-בסביבה וירטואלית, שמדמה רשת מבוססת שירותי ענן CNF ו PEER-הם פודים (Pods) שרצים בתשתית הקוברנטיס של אמזון AWS EKS (Elastic Kubernetes Service). הפודים מריצים מערכות הפעלה של לינוקס.

הכתובות שנעשה בהן שימוש בבדיקה הן 10.0.100.216 עבור PEER (לקוח) ו-10.0.100.185 עבור CNF (שרת).

$$\text{Throughput} = \frac{\text{Total Data Transferred}}{\text{Total Transfer Time}}$$

Total Data Transferred – כמות הנתונים שהועברה במהלך הבדיקה, נמדדת ב-Gigabytes או Megabytes.

Total Transfer Time – משך זמן הבדיקה (בדוגמה זו: 10 שניות)

מטרת הבדיקה:

לאמוד את יכולת העברת הנתונים המקסימלית (Throughput) בין ההתקנים PEER ל-CNf. לבדוק את יציבות החיבור באמצעות ניתוח מספר ההעברות החוזרות (Retransmissions). לוודא שהרשת עומדת בדרישות לרוחב פס גבוה, אשר נדרשות לתמיכה בתעבורה כבדה ויציבה על ידי ניתוח Congestion Window.

כלי הבדיקה: iperf3

כלי פתוח למדידת ביצועי רשת, המאפשר לבדוק את קצב העברת הנתונים המרבי בין התקן לקוח להתקן שרת. הוא מתאים למדידת רוחב פס הן עבור פרוטוקול UDP והן עבור פרוטוקול TCP.

iperf3 מציג את כמות הנתונים שהועברו בכל שנייה, את קצב העברת הנתונים (Bitrate) ואת מספר ההעברות החוזרות (Retransmissions), כל אלה משמשים להערכת איכות הרשת.

תהליך הבדיקה:

התקנה והפעלת הכלי בהתקן CNF בתצורת השרת המאזין לתעבורה:

```
Linux cnf-b88f4bdd5-tkmzj 5.10.226-214.880.amzn2.x86_64 #1 SMP Tue Oct 8 16:18:15 UTC 2024 x86_64 Linux
/ # iperf3 -s
-----
Server listening on 5201 (test #1)
-----
```

התקנה והפעלת הכלי בהתקן PEER בתצורת הלקוח המשדר את התעבורה:

פקודה:
iperf3 -c 10.0.100.185 -t 10

בפקודה זו מבצעת בדיקת Throughput למשך 10 שניות בין ה-PEER לכתובת ה-IP של ה-CNf (10.0.100.185), בבדיקה זו נעשה שימוש בפרוטוקול TCP.

```
Linux peer-76d759b69c-kkk45 5.10.226-214.880.amzn2.x86_64
/ # iperf3 -c 10.0.100.185 -t 10
```

תוצאה רצויה:

עבור ספקי אינטרנט (ISP), קצב העברת נתונים גבוה ויציב הוא חיוני. לפי ה-FCC, חיבורי פס רחב נחשבים למהירים אם הם מספקים לפחות 25 מגה-ביט לשנייה להורדה ו-3 מגה-ביט לשנייה להעלאה עבור כל לקוח, כאשר כל אתר משרת מספר גדול של לקוחות. עם זאת, ברשתות

ארגוניות או חיבורי Data Center, מצפים לקצבים גבוהים בהרבה, בהתאם לתשתית הקיימת. תוצאה רצויה בבדיקה זו היא קצב העברת נתונים של מעל 30 Gbps, עם מספר מינימלי של Retransmissions (מומלץ פחות מ-5), המעיד על חיבור יציב ויעיל.

תוצאה שהתקבלה:

```
Linux peer-76d759b69c-44445 5.10.226-214.880.amzn2.x86_64 #1 SMP Tue Oct 8 16:18:15 UTC 2024
# iperf3 -c 10.0.100.185 -t 10
Connecting to host 10.0.100.185, port 5201
[ 5] local 10.0.100.216 port 59400 connected to 10.0.100.185 port 5201
```

[ID]	Interval	Transfer	Bitrate	Retr	Cwnd
[5]	0.00-1.00 sec	4.51 GBytes	38.7 Gbits/sec	0	3.55 MBytes
[5]	1.00-2.00 sec	4.50 GBytes	38.7 Gbits/sec	0	3.55 MBytes
[5]	2.00-3.00 sec	4.23 GBytes	36.3 Gbits/sec	1	3.55 MBytes
[5]	3.00-4.00 sec	4.43 GBytes	38.1 Gbits/sec	0	3.55 MBytes
[5]	4.00-5.00 sec	4.28 GBytes	36.8 Gbits/sec	2	3.55 MBytes
[5]	5.00-6.00 sec	4.45 GBytes	38.3 Gbits/sec	0	3.55 MBytes
[5]	6.00-7.00 sec	4.52 GBytes	38.8 Gbits/sec	0	3.55 MBytes
[5]	7.00-8.00 sec	4.28 GBytes	36.7 Gbits/sec	0	3.55 MBytes
[5]	8.00-9.00 sec	4.51 GBytes	38.8 Gbits/sec	0	3.55 MBytes
[5]	9.00-10.00 sec	4.48 GBytes	38.5 Gbits/sec	0	3.55 MBytes

[ID]	Interval	Transfer	Bitrate	Retr	sender	receiver
[5]	0.00-10.00 sec	44.2 GBytes	38.0 Gbits/sec	3		
[5]	0.00-10.00 sec	44.2 GBytes	38.0 Gbits/sec			

סך הנתונים שהועברו: 44.2 גיגה-בייט.

זמן הבדיקה: 10 שניות.

קצב העברת הנתונים (Throughput): 38.0 גיגה-ביט לשנייה.

כדי לחשב את ה-Throughput, במונחי Gigabytes per Second נשתמש בנוסחה:

$$\text{Throughput (GBps)} = \frac{\text{Total Data Transferred}}{\text{Total Transfer Time}} = \frac{44.2 \text{ GB}}{10 \text{ Sec}} = 4.2 \frac{\text{GB}}{\text{Sec}}$$

כאשר מתרגמים זאת למונחים של Gigabits per Second בהם נהוג להשתמש בתעשייה:

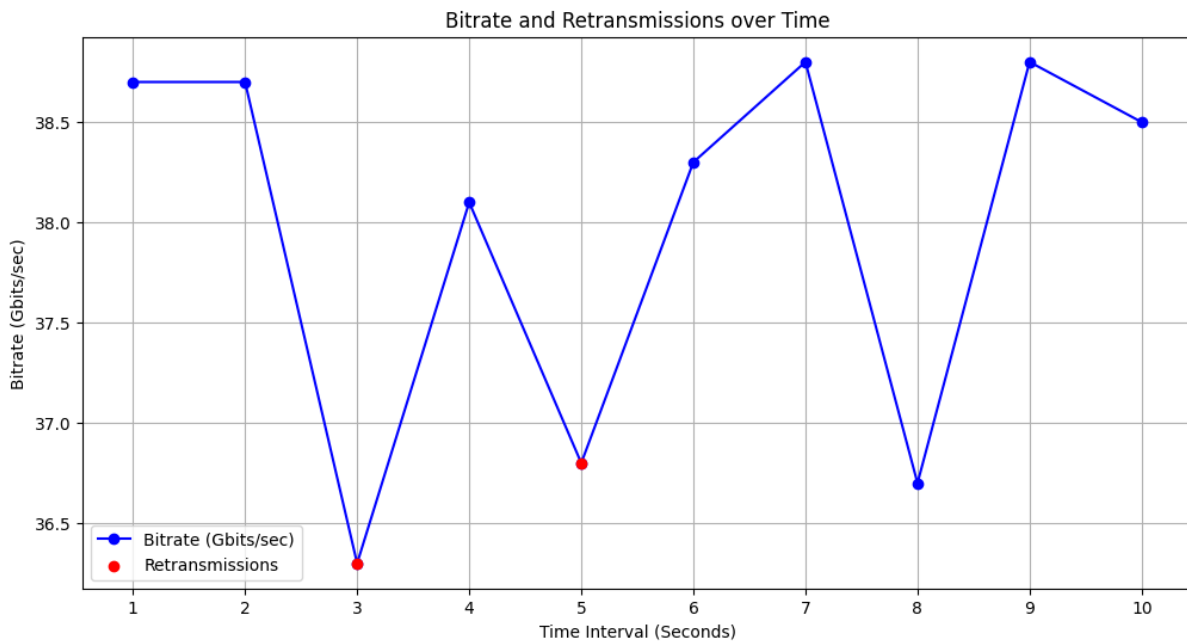
$$\text{Throughput (Gbps)} = 4.2 \frac{\text{GB}}{\text{Sec}} \times 8 = 35.36 (\text{Gbit/sec})$$

Throughput: קצב העברת הנתונים שנמדד במהלך הבדיקה הוא בממוצע 38.0 Gbps (גיגה-ביט לשנייה). תוצאה זו היא גבוהה מאוד ועומדת ביעדים שהוגדרו לתמיכה בתעבורה כבדה, מה שמעיד על יכולת הרשת להעביר נתונים בקצב גבוה באופן יציב.

Retransmissions: במהלך הבדיקה נרשמו 3 העברות חוזרות (Retransmissions) בלבד. כמות נמוכה זו של Retransmissions מעידה על חיבור יציב ומינימום בעיות בקישוריות בין PEER ל-CNF.

Congestion Window (Cwnd): הערך של חלון הגודש (Cwnd) נשאר קבוע על 3.55 MBytes במהלך הבדיקה, דבר המעיד על יציבות בקצב התעבורה וביכולת הרשת להתמודד עם העומס. (הרחבה על הפרמטר ברקע התיאורתי)

תיאור גרפי של תוצאות הבדיקה:



הקו הכחול מציין את קצב התעבורה בכל שנייה במהלך הבדיקה, ונע בין כ-36.3 ל-38.8 גיגה ביט לשנייה.

הנקודות האדומות המופיעות על קו קצב התעבורה מציינות שידורים חוזרים, אחד בין השנייה ה-2-3 ושניים נוספים בין השנייה ה-4-5.

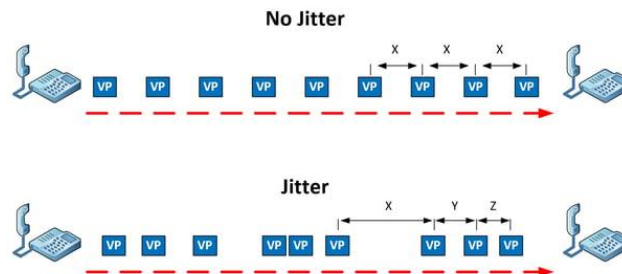
קצב העברת הנתונים הממוצע של 38.0 גיגה-ביט לשנייה מעיד על רשת בעלת קיבולת גבוהה במיוחד. תוצאה זו מתאימה לא רק לחיבורי אינטרנט רגילים אלא גם לרשתות מרכזי נתונים (Data Centers) ויישומים רגישים, בהם נדרשת יכולת תעבורה מהירה ואמינה. מספר השידורים החוזרים הנמוך (3 בלבד) מצביע על חיבור יציב עם מינימום שגיאות.

5. בדיקת Jitter ו-Packet Loss בפרוטוקול UDP באמצעות iperf3 בין התקני CNF ו-PEER

בדיקה זו מבוצעת כדי למדוד את Jitter (תנודת זמן) ואת Packet Loss (אובדן מנות) בתקשורת UDP בין שני התקני הקצה CNF ו-PEER בסביבה וירטואלית, שמדמה רשת מבוססת שירותי ענן CNF ו-PEER הם פודים (Pods) שרצים בתשתית הקוברנטיס של אמזון AWS EKS (Elastic Kubernetes Service). הפודים מריצים מערכות הפעלה של לינוקס והם ממוקמים במקומות גיאוגרפים שונים.

UDP (User Datagram Protocol) הוא פרוטוקול חסר חיבור (connectionless) שאינו מספק תיקוני שגיאות ברמת השידור ובקרת זרימה של תעבורה ולכן הוא מתאים במיוחד למדידות ביצועים ביישומים רגישי זמן, כמו שיחות VoIP ושידורי וידאו.

Jitter מוגדר כשונות בזמן ההגעה של מנות נתונים רצופות ברשת. לפי התקן RFC 4689 מחושב כהפרש המוחלט בין זמני ההעברה של שתי חבילות עוקבות באותו זרם נתונים. ככל ש-Jitter נמוך יותר, כך החיבור יציב יותר. ערך גבוה מדי עלול לגרום לעיכובים, ניתוקים, ולפגום באיכות של יישומי אודיו ווידאו, ולכן חשוב לשמור על ערכי Jitter נמוכים.



נוסחא לתיאור הפרמטר:

מוגדר כהפרש המוחלט בין זמני ההעברה של שתי חבילות עוקבות באותו הזרם.

$$\text{Jitter} = |\text{Latency}_A - \text{Latency}_B|$$

Latency_A – זמן ההשהייה של חבילה A.

Latency_B – זמן ההשהייה של חבילה B.

כלי הבדיקה: iperf3

המאפשר ביצוע בדיקות TCP ו-UDP לקבלת מדדים מדויקים של רוחב פס, תנודת זמן ואובדן מנות. במצב iperf3, UDP מספק מידע ייחודי על Jitter ועל Packet Loss, שני מדדים קריטיים להערכת איכות הרשת ביישומים הדורשים תעבורה רציפה.

תהליך הבדיקה:

במקרה זה, CNF מוגדר כשרת, בעוד PEER מוגדר כלקוח שמבצע את שליחת המנות. הבדיקה נמשכת 10 שניות עם קצב תעבורה של 1 מגה-ביט לשנייה. המדידה של Jitter ואובדן מנות מסייעת להעריך את איכות הקישוריות ואת יציבות התקשורת בין ההתקנים, בהתאם לתנאים אופייניים של ספקי תקשורת (ISP).

התקנה והפעלת הכלי בהתקן CNF בתצורת השרת המאזין לתעבורה:

```
Linux cnf-b88f4bdd5-tkmzj 5.10.226-214.880.amzn2.x86_64 #1 SMP Tue Oct 8 16:18:15 UTC 2024 x86_64 Linux
/ # iperf3 -s
-----
Server listening on 5201 (test #1)
-----
```

התקנה והפעלת הכלי בהתקן PEER בתצורת הלקוח המשדר את התעבורה:

iperf3 -c 10.0.100.185 -u -b 1M -t

פקודה:

10

```
/ # iperf3 -c 10.0.100.185 -u -b 1M -t 10
Connecting to host 10.0.100.185, port 5201
[ 5] local 10.0.100.216 port 60422 connected to 10.0.100.185 port 5201
[ ID] Interval      Transfer    Bitrate      Total Datagrams
[ 5] 0.00-1.00 sec  122 KBytes  1.00 Mbits/sec  14
[ 5] 1.00-2.00 sec  122 KBytes  1.00 Mbits/sec  14
```

בפקודה זו מבצעת בדיקת תעבורה בקצב קבוע של 1Mbit לשנייה, במשך 10 שניות בין ה-PEER לכתובת ה-IP של ה-CNF (10.0.100.185), בבדיקה זו נעשה שימוש בפרוטוקול UDP.

מטרת הבדיקה:

מדידת ה Jitter בתקשורת UDP כדי לוודא כי הרשת יכולה לתמוך ביישומים רגישים לזמן כמו שיחות VoIP, שידורי וידאו, ויישומי מדיה בזמן אמת.

תוצאה רצויה:

בקרב ספקי שירותי תקשורת (ISP) ישנם ערכים מומלצים למדדים אלה כדי להבטיח איכות שירות (QoS) גבוהה, בדרך כלל ערך נמוך מ-30 מילישניות (ms) נחשב לטוב עבור יישומים רגישים זמן.

תוצאה שהתקבלה:

ID	Interval	Transfer	Bitrate	Jitter	Lost/Total Datagrams
5	0.00-10.00 sec	1.19 MBytes	1.00 Mbits/sec	0.000 ms	0/140 (0%) sender
5	0.00-10.00 sec	1.19 MBytes	1.00 Mbits/sec	0.432 ms	0/140 (0%) receiver

מהבדיקה התקבל ערך Jitter של **0.432ms** ערך זה חושב על ידי בדיקה של ההפרשים בזמני ההגעה של כל זוג חבילות רצופות לאורך כל משך הבדיקה (10 שניות).

דוגמא לחישוב בנוסחא:

החבילה הראשונה הגיעה לאחר 10.000 מילישניות והחבילה השנייה לאחר 10.432 מילישניות

$$\text{Jitter} = |\text{Latency}_A - \text{Latency}_B| = |10.000\text{ms} - 10.432\text{ms}| = 0.432\text{ms}$$

במהלך הבדיקה, נמדדו כל זוג חבילות עוקבות, וההפרשים בין זמני ההגעה שלהן חושבו. ההפרש הממוצע בין כל זוגות החבילות הללו, הביא לתוצאה הסופית של 0.432 מילישניות.

התוצאות מראות שהחיבור בין PEER ל-CNF עומד בדרישות הגבוהות ביותר לאיכות חיבור רשת עם JITTER נמוך במיוחד של 0.432ms המעיד על יציבות ויכולת לתמוך ביישומים רגישים זמן כמו VoIP ושידורי וידאו ללא פגיעה בחוויית המשתמש.

6. שיעור שגיאת ביטים (Bit Error Rate - BER) בפרוטוקול UDP באמצעות כלי iperf3 בין התקני CNF ו-PEER

מדדת Bit Error Rate (BER) היא קריטית להערכת אמינות ודיוק התקשורת ברשתות בהן נדרשת רמת אמינות גבוהה, במיוחד כאשר מדובר ברשתות רגישות לשגיאות כמו רשתות של ספקי שירותי תקשורת.

מדד ה-BER משמש למדידת דיוק והאמינות של העברת הנתונים, אך קשה למדידה ישירה בתנאים רגילים של שידור. לכן, נעשה שימוש במדד – Packet Error Rate (PER) – אשר ניתן להמירו ל-BER בהתבסס על גודל החבילה ועל איכות הקישור.

הגדרות ושימוש ב-BER ו-PER:

Bit Error Rate (BER): מייצג את היחס בין מספר הביטים השגויים לבין סך כל הביטים שהועברו. מדד זה משקף את השיעור הכולל של טעויות שנוצרו במהלך ההעברה, ומשמש לאמידת אמינות הקישור.

$$BER = \frac{\text{Number of Bit Errors}}{\text{Total Number of Bits Transmitted}}$$

Packet Error Rate (PER): מוגדר כיחס בין מספר החבילות השגויות (אחרי תיקון שגיאות FEC) לבין סך כל החבילות שהתקבלו.

$$PER = \frac{\text{Number of Packet Drops}}{\text{Total Number of Packets Transmitted}}$$

הקשר בין BER ל-PER תלוי בגודל החבילה (n) ניתן להציג את היחס בין BER ל-PER באמצעות הנוסחה:

$$PER = (BER \cdot n)^n$$

כלי הבדיקה: iperf3

המאפשר ביצוע בדיקות TCP ו-UDP לקבלת מדדים מדויקים של רוחב פס, תנועת זמן ואובדן מנות. במצב iperf3, UDP מספק מידע ייחודי על Jitter ועל Packet Loss, שני מדדים קריטיים להערכת איכות הרשת ביישומים הדורשים תעבורה רציפה.

תהליך הבדיקה:

במקרה זה, CNF מוגדר כשרת, בעוד PEER מוגדר כלקוח שמבצע את שליחת המנות. הבדיקה נמשכת 10 שניות עם קצב תעבורה של 1 מגה-ביט לשנייה.

התקנה והפעלת הכלי בהתקן CNF בתצורת השרת המאזין לתעבורה:

```
Linux cnf-b88f4bdd5-tkmzj 5.10.226-214.880.amzn2.x86_64 #1 SMP Tue Oct 8 16:18:15 UTC 2024 x86_64 Linux
/ # iperf3 -s
-----
Server listening on 5201 (test #1)
-----
```

התקנה והפעלת הכלי בהתקן PEER בתצורת הלקוח המשדר את התעבורה:

`iperf3 -c 10.0.100.185 -u -b 1M -t 10`

פקודה:

```
/ # iperf3 -c 10.0.100.185 -u -b 1M -t 10
Connecting to host 10.0.100.185, port 5201
[ 5] local 10.0.100.216 port 60422 connected to 10.0.100.185 port 5201
[ ID] Interval           Transfer     Bitrate     Total Datagrams
[ 5] 0.00-1.00 sec      122 KBytes  1.00 Mbits/sec  14
[ 5] 1.00-2.00 sec      122 KBytes  1.00 Mbits/sec  14
```

בפקודה זו מבצעת בדיקת תעבורה בקצב קבוע של 1Mbit לשנייה, במשך 10 שניות בין ה-PEER לכתובת ה-IP של ה-CNF (10.0.100.185), בבדיקה זו נעשה שימוש בפרוטוקול UDP.

מטרת הבדיקה:

הבדיקה נועדה למדוד את ערכי BER ו-PER בין שני רכיבי רשת (PEER ו-CNF) ומהם להסיק על שיעור השגיאות ברשת.

תוצאה רצויה:

עבור ספקי שירותי אינטרנט, על פי תקנים כמו IEEE 802.16-2009 ערכי BER מתחת ל 10^{-6} נחשבים לטובים, ו-PER של עד 1% הוא קביל. יחד עם זאת, ביישומים קריטיים, השאיפה היא להגיע לערכי BER נמוכים יותר כמו 10^{-9} ואובדן מנות נמוך ככל האפשר, כלומר קרוב ל-0%.

תוצאה שהתקבלה:

ID	Interval	Transfer	Bitrate	Jitter	Lost/Total Datagrams
5	0.00-10.00 sec	1.19 MBytes	1.00 Mbits/sec	0.000 ms	0/140 (0%) sender
5	0.00-10.00 sec	1.19 MBytes	1.00 Mbits/sec	0.432 ms	0/140 (0%) receiver

במהלך הבדיקה לא נרשם אובדן מנות Packet Loss כלל (0%), מה שמעיד על כך שהרשת מתפקדת בצורה אמינה וללא שגיאות.

שיעור BER: מאחר שלא היו מנות שאבדו, ניתן להניח כי גם שיעור ה-BER נמוך ביותר או קרוב לאפס. בדרך כלל, ספקי תקשורת שואפים ל-BER של פחות מ- 10^{-6} וזהו סף מומלץ בתעשייה לשמירה על איכות חיבור גבוהה ואמינות מרבית. תוצאה זו מצביעה על כך שהרשת אכן עומדת בדרישות אלו.

מאחר והבדיקה תקינה נבצע סימולציה תיאורטית לחישוב ערך BER על פי ערך PER:

נניח ומתוך 140 חבילות שנשלחו 2 חבילות לא הגיעו וכי כל חבילה בגודל של 1500 ביטים.

חישוב ה-PER:

$$PER = \frac{\text{Number of Packet Drops}}{\text{Total Number of Packets Transmitted}} = \frac{2}{140} = 0.0143 \approx 1.43\%$$

חישוב BER: מאחר וכל חבילה מורכבת מ-1500 ביטים מספר הביטים השגויים:

$$\text{Number of Bit Errors} = 1500 \times 2 = 3000$$

$$\text{Total Number of Bits Transmitted} = 1500 \times 140 = 210000$$

חישוב ה-BER:

$$BER = \frac{\text{Number of Bit Errors}}{\text{Total Number of Bits Transmitted}} = \frac{3000}{210000} = 0.0143 \approx 1.43\%$$

חישוב ה-BER וה-PER הראה שיעור שגיאות משוער של 1.43%, ערך זה גבוה מהמקובל לספקי תקשורת ($BER < 10^{-6}$).

4.6. תכנון ומימוש התשתית העננית המארכת את הפרויקט

4.6.1. תוכנות ורישויים

בכדי להתחיל לעבוד על התשתית העננית נדרש להוריד מספר תוכנות וכלים שונים למחשב. מהעמקה שביצענו במספר מדריכים העוסקים בהקמת תשתיות ענן, הבנו שקיימים מספר כלים שנדרש מאיתנו להתקין לפני שאנחנו מתחילים לעבוד, הגרסאות אותן הורדנו והתקנו היו בהתאמה לתשתית לינוקס על מנת שנוכל להתחבר לענן ולהרים את התשתית בצורה נוחה יותר. אלו התוכנות שאותן נדרשנו להוריד בכדי להתחיל לעבוד :

- Ubuntu 20.04.6 LTS
- AWS CLI
- Kubectl
- Helm
- Skopeo
- Docker
- Git

כעת נפרט על הכלים שהשתמשנו :

Ubuntu 20.04.6 LTS

מערכת הפעלה מבוססת לינוקס

גרסת ה-LTS-מציינת תמיכה ארוכת טווח, גרסה זו מתאימה מאוד לשימוש בשרתים ובסביבות ענן בשל יציבותה ואבטחתה, והיא נפוצה בתשתיות IT ובסביבות פיתוח בחרנו בגרסה זו מהסיבה שהיא יציבה ולא חדשה מידי, קיימים עדכוני אבטחה קבועים וקיימת תאימות לכלים אחרים – כלומר תוכנות האחרות יודעות להתממשק איתה בצורה טובה.

AWS CLI (Command Line Interface)

הוא כלי שורת פקודה לניהול שירותי AWS הוא מאפשר למשתמשים לבצע פעולות ולקיים אינטראקציה עם שירותי AWS ישירות משורת הפקודה או מסקריפטים אוטומטיים, ללא צורך בממשק הווי של AWS כלי זה אידיאלי לשימושים כמו ניהול ענן, תפעול תשתיות ואוטומציה של תהליכים, במיוחד כאשר יש צורך בגמישות, מהירות ויעילות בתפעול.

Kubectl

הוא כלי שורת פקודה לניהול Kubernetes Cluster

הוא משמש לשליטה, ניהול וניטור של כל הרכיבים באשכול Kubernetes ומאפשר לבצע פריסות (Deployments), להגדיר שירותים (Services), לנהל קונפיגורציות ולהריץ בדיקות ובנוסף פעולות תפעול כמו הקמת Pods Nodes, ConfigMaps

Helm

הוא כלי לניהול חבילות (Package Manager) עבור Kubernetes שנועד לפשט את תהליך הפריסה, הניהול והעדכון של אפליקציות בסביבת Kubernetes זהו כלי יעיל במיוחד כשעובדים עם פרויקטים מורכבים הכוללים מספר רב של רכיבים.

Skopeo

הוא כלי לניהול Images של קונטיינרים. הוא מאפשר לבצע פעולות שונות על container image מבלי שיהיה צורך להפעיל מנוע קונטיינרים כגון Docker. באמצעות Skopeo אפשר לשכפל, לבדוק ולהעביר container image בין Registries בצורה קלה ויעילה.

Docker

הוא פלטפורמה לניהול קונטיינרים (Containers) שמאפשרת למפתחים ולמנהלי מערכת לבנות, לפרוס ולהריץ יישומים מבודדים בסביבות שונות.

Git

מערכת לניהול גרסאות של קוד, קבצי YAML, תבניות קוד וסקריפטים, נעזרנו בה בכדי לנהל ולהשתמש בקבצי קוד שרלוונטים לפרויקט שלנו.

רישוי :

xrd-vrouter-container-x86.24.2.2

גרסת קונטיינר של **Cisco IOS XRd vRouter** שמיועדת לפעול על פלטפורמות וירטואליות מבוססת על מערכת ההפעלה **Cisco IOS XR** גרסה זו מותאמת במיוחד לסביבות מבוססות ענן, תשתיות קונטיינרים, וסביבות מבוזרות כמו Kubernetes ו-Docker ומאפשרת להפעיל יכולות ניתוב מתקדמות מבלי צורך בחומרת נתב ייעודית.

4.6.2. תכנון טכני של שלבי המימוש

- 1) תכנון ובדיקת גישה של מספר משתמשים לפרויקט
 - תכנון הרשאות עבור 2 משתמשים לעבודה בתור ADMIN
 - בדיקת היכולת
- 2) קביעת מענה הגנה : הגדרת מדיניות אבטחה מקיפה שתכלול :
 - הגנה על התעבורה
 - Security Groups – הגבלות גישה
 - IAM – תכנון מדיניות ניהול זהויות והרשאות, תכנון גישה לEKS, לשרתים, לECR ולמשאבים נוספים.

- הצפנה על ידי פרוטוקולי רשת
 - (3) תכנון רשת : קביעת תצורת הרשת והגדרות VPC
 - קביעת כמות + כתובות של רשתות ציבוריות ופרטיות
 - תכנון גישה מאובטחת לאינטרנט : שימוש ב-NAT Gateway בכדי לאפשר גישה לאינטרנט ללא חשיפת רשתות פרטיות .
 - (4) תכנון אחסון ו-IMAGE ב-ECR
 - תכנון אחסון לטובת IMAGE הנדרשים
 - תכנון מערך האחסון
 - (5) תכנון עבור EKS Cluster :
 - קביעת גודל המשאבים שיוקצו לטובת הקלאסטר
 - קביעת מספר השרתים , region וכדי
 - (6) תכנון תצורת Worker Nodes :
 - תכנון ליצירת AMI - בחינת הגדרות עבור : Kernel , Hugepages , הגדרות קבצים, ותצורת Docker
 - קביעת משאבים עבור כל Worker Node
 - (7) תכנון תהליך הניהול ופריסת הרכיבים :
 - כתיבת קונפיגורציות וקבצי אוטומציות
 - תכנון עבודה לפי פרוטוקולים נדרשים
 - תכנון מערך ניטור לטובת התמודדות עם תקלות, עומסים ושגיאות
- 4.6.3. ארכיטקטורת ענן למימוש :



X. ארכיטקטורת ענן HLD לפרויקט

הסבר על הארכיטקטורה :

CloudFormation Stack : תשתית אוטומטית שמקימה את כל המשאבים בצורה מסודרת ומבוקרת.

ה Stack-יכול :

VPC : רשת פרטית וירטואלית הכוללת חיבור לאינטרנט ותתי רשתות פרטיות וציבוריות.

Security Groups : הגנה על הרשת באמצעות כללים מגבילים לתעבורה.

Subnets : תתי רשתות נפרדות עבור תעבורת ניהול ותעבורת נתונים.

AWS EKS : שירות Kubernetes מנוהל שמאפשר פריסה, ניהול ושדרוג של אפליקציות מבוססות קונטיינרים בצורה פשוטה ויעילה.

EC2 Worker Nodes : שרתים וירטואליים שמשמשים כצמתים באשכול Kubernetes ומריצים את הקונטיינרים של Cisco IOS XRd וכלים נוספים לניהול הרשת.

Cisco IOS XRd : רכיב רשת וירטואלי שמבצע ניתוב של תעבורה בין רכיבי המערכת ומספק יכולות ניהול תעבורה, הפרדת רשתות ותמיכה בתעבורה בין תתי הרשתות הפנימיים.

4.7. שלב המימוש לתשתית העננית המארכת את הפרויקט

4.7.1. סדר הכנת התשתית ל XRd:

- יצירת חוקות ופוליסות להרשאות IAM
- העלאת קבצים לאחסון S3 ו ECR
- יצירת VPC
- הגדרת EKS Cluster
- יצירת Worker Node
- שימוש בשירותי ניטור לפתרון תקלות

4.7.2. IAM:

בשלב זה עשינו 2 דברים עיקריים :

- יצרנו משתמשים בכדי לספק גישה לממשקי העבודה
- יצרנו ROLES ו Policy עבור EKS ו Worker Nodes לטובת גישה והרשאות לניהול.
- **יצירת משתמשים:**

כאשר מספר אנשים מנהלים פרויקט בתשתית וירטואלית נדרש להקים משתמש עבור כל אחד עם ההרשאות והשם תפקיד המתאים בכדי שיהיה מעקב על הפעולות המתבצעות .

פרויקט זה כולל 2 אנשים שמנהלים אותו ולכן הקמנו USER ROOT ובנוסף יצרנו משתמש נוסף עם הרשאות גישה וניהול להכל כלומר USER ADMIN , בכדי ששנינו נוכל להקים ולנהל את הפרויקט במידה שווה .

יצרנו משתמש ROOT עם מייל וסיסמה עבור תומר ויצרנו משתמש בשם STAV_HIT_Project עם מדיניות AdministratorAccess , בנוסף יצרנו STAV_HIT_Project_accessKeys לצורך התחברות דרך CLI .

Review

The following policies will be attached to this user. [Learn more](#)

User details

User name
STAV_HIT_Project

Permissions summary (1)

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

X. יצירת משתמש עם הרשאות ADMIN

[IAM](#) > [Users](#) > [STAV_HIT_Project](#) > Create access key

Step 1

[Access key best practices & alternatives](#)

Step 2 - optional

[Set description tag](#)

Step 3

Retrieve access keys

Retrieve access keys [Info](#)

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIADEVXYKU6UL7ITSHN	***** Show

X. יצירת מפתח גישה למשתמש לטובת ניהול דרך CLI

■ יצירת IAM ROLE עבור EKS Cluster :

יצרנו Role שמאפשר Control Plane של EKS לבצע פעולות ולגשת למשאבים הנדרשים. (פירוט הפעולות נמצא תחת הכותרת EKS)

יצרנו trust policy – מדיניות שקובעת אילו שירותים או משתמשים יכולים לקבל את ההרשאות של Role זה , במקרה זה קבענו שההרשאה להשתמש בRole זה הינה רק eks.amazonaws.com .

כדי לאפשר ל Control Plane - לנהל את הקלאסטר ולגשת למשאבים הדרושים לו,

צירפנו לתפקיד את מדיניות ההרשאות AmazonEKSClusterPolicy

פוליסה זו כוללת את כל ההרשאות הנדרשות ל Control Plane של EKS לניהול רכיבי הקלאסטר,

כגון יצירה וניהול של משאבים בKubernetes - כולל פודים (Pods), שירותים (Services), אחסון, ועוד.

מדיניות זו מאפשרת ל Control Plane גם לנהל רכיבי רשת כמו

ENI (Elastic Network Interfaces) ולבצע את תפקידי הניהול הנדרשים על מנת לשמור על פעילות ויציבות הקלאסטר.

- יצירת IAM Role עבור Worker Nodes :
- יצרנו Role עבור שרתי ה Worker Nodes בקלאסטר, שמאפשר להם לפעול במסגרת ה EKS Cluster (בעצם נותן הרשאות לEC2)
- יצרנו Trust Policy שמאפשרת לשירות EC2 להשתמש בRole זה.
- הצמדנו Policies כדי לאפשר ל Worker Nodes לפעול במסגרת הCluster הפוליסות שהגדרנו :

AmazonEKSWorkerNodePolicy

פוליסה זו מאפשרת ל Worker Nodes להתחבר לקלאסטר EKS ולהיות מנוהלים על ידי ה Control Plane של הקלאסטר.

היא כוללת את כל ההרשאות הבסיסיות הנדרשות כדי שהWorker Nodes- יוכלו לשתף פעולה עם שאר רכיבי הקלאסטר ולבצע פעולות בהתאם לצרכים של Kubernetes.

AmazonEC2ContainerRegistryReadOnly

פוליסה זו מעניקה ל Worker Nodes הרשאה לקרוא Container Images המאוחסנות ב Amazon ECR (Elastic Container Registry)

היא מאפשרת ל Worker Nodes למשוך images מ ECR שמחן נבנים הפודים (Pods) בקלאסטר.

AmazonEKS_CNI_Policy

פוליסה זו מספקת הרשאות ניהול רשת הנדרשות לצורך הקמת והפעלת הפודים ב Worker Nodes ומאפשרת להם ליצור ולנהל Elastic Network Interfaces (ENI) כדי לתמוך בתעבורת רשת בין הפודים לבין רכיבים אחרים בקלאסטר.

הפוליסה הזו חשובה במיוחד לתמיכה בתשתיות הרשת של Kubernetes ובניהול רשתות פרטיות במידת הצורך.

Amazon S3 4.7.3

בפרויקט השתמשנו ב S3 כמיקום אחסון לתבניות CloudFormation שכוללות את הגדרות התשתית השונות הדרושות לפריסה.

השתמשנו בשירות זה לפי השלבים הבאים :

יצרנו Bucket - מאגר אחסון לכל קבוצת קבצים שהיינו צריכים ,

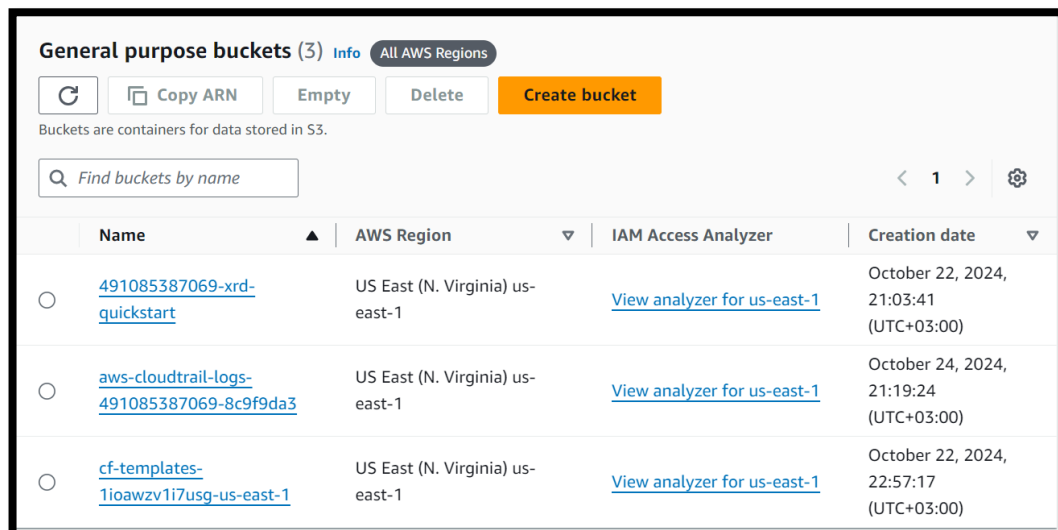
לכל Bucket שייכנו Region מסוים

הגדרנו הרשאות גישה בהתאם לצורך

העלנו קבצים

ולאחר מכן השתמשנו ב URL שנוצר לצורך גישה לאחסון .

כך נראה השירות דרך התחברות Console בפרויקט שלנו :

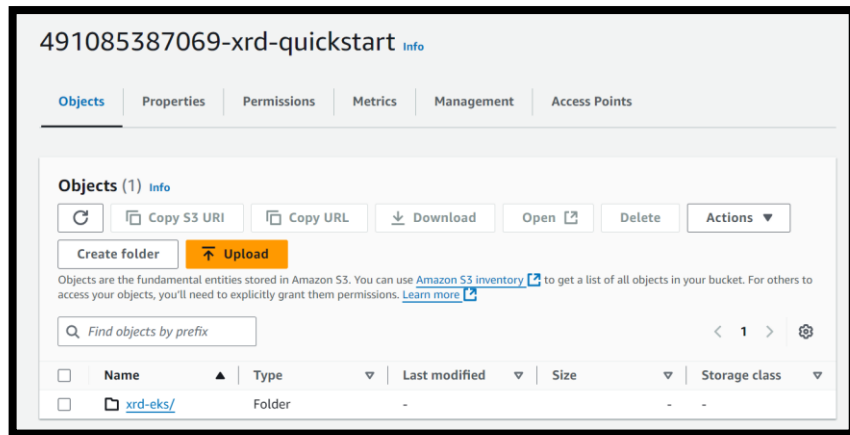


General purpose buckets (3) Info All AWS Regions				
	Copy ARN	Empty	Delete	Create bucket
Buckets are containers for data stored in S3.				
<input type="text" value="Find buckets by name"/>				
	Name ▲	AWS Region ▼	IAM Access Analyzer	Creation date ▼
<input type="radio"/>	491085387069-xrd-quickstart	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 22, 2024, 21:03:41 (UTC+03:00)
<input type="radio"/>	aws-cloudtrail-logs-491085387069-8c9f9da3	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 24, 2024, 21:19:24 (UTC+03:00)
<input type="radio"/>	cf-templates-1ioawzv1i7usg-us-east-1	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 22, 2024, 22:57:17 (UTC+03:00)

X.שירות אחסון , AWS Console

להלן הפרטים על כל אחד מה-Buckets המוצגים :

xrd-quickstart-491085387069

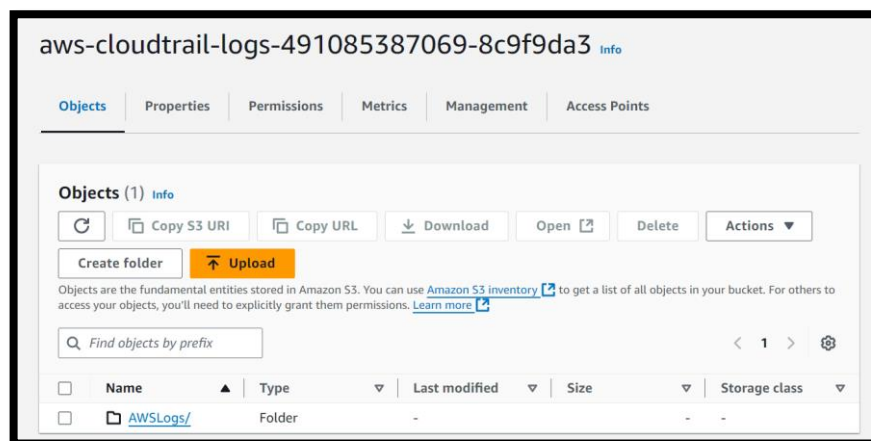


X. תצוגת Bucket 1 ב-AWS Console

אזור us-east-1 (N. Virginia) - US East (Region):

Bucket זה משמש כמאגר לקבצים הקשורים לפרויקט "XRd", עבור הגדרות ראשוניות.

aws-cloudtrail-logs-491085387069-8c9f9da3

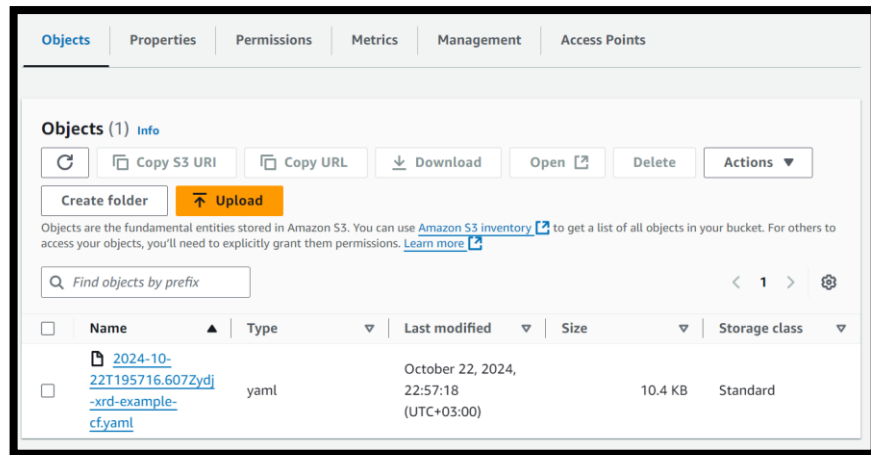


X. תצוגת Bucket 2 ב-AWS Console

אזור us-east-1 (N. Virginia) - US East (Region):

Bucket זה משמש לאחסון לוגים של AWS CloudTrail, שירות שמאפשר מעקב אחרי פעילויות וגישה לשירותי AWS, לצורך אבטחה וניטור, לוגים אלו משמשים גם לCloudWatch.

cf-templates-1ioawzv17iusg-us-east-1



X. תצוגת Bucket ב AWS Console

אזור us-east-1 (Region): US East (N. Virginia)

Bucket זה מיועד לתבניות CloudFormation (cf) לצורך ניהול והקמה של תשתיות בענן AWS.

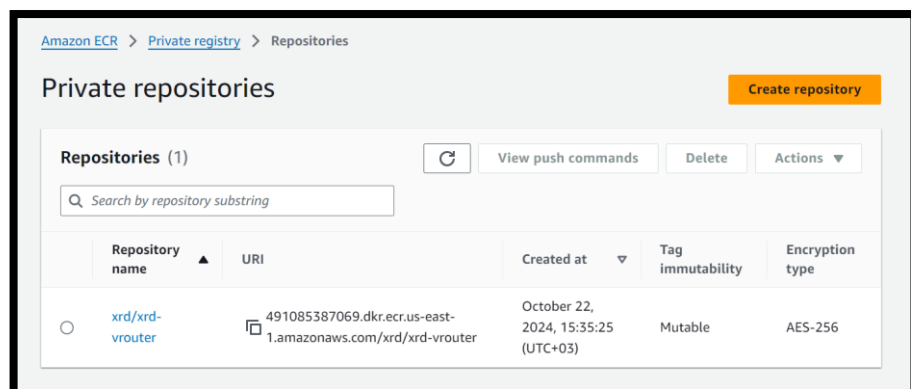
לכל אחד מה-Buckets יש קישור ל-IAM Access Analyzer עבור ניתוח גישה ובקרת אבטחה לאזור us-east-1, המאפשר לנתח מי יכול לגשת לנתונים המאוחסנים ב-Bucket בהתאם למדיניות הרשאות IAM.

ECR.4.7.4

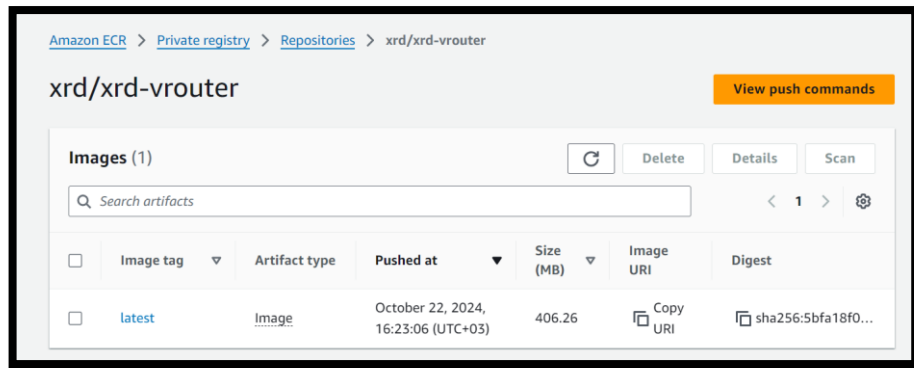
Amazon ECR (Elastic Container Registry)

בפרויקט זה השתמשנו בשירות ECR לטובת אחסון IMAGE של ה-Virtual Router כדי שהן יהיו זמינות לכלל הרכיבים בקלאסטר. שימוש ב-ECR מאפשר שליטה, ניהול גרסאות ואבטחת מידע גבוהות images.

להלן imagen שאחסנו ב-ECR



X. שירות Amazon ECR



X. אחסון image בRepositories

4.7.5 יצירת VPC

בשלב זה יצרנו רשת וירטואלית פרטית ומבודדת בענן,

השלב הבא ביצענו :

• יצירת VPC:

יצרנו VPC עם כתובת CIDR: 10.0.0.0/16.

שם ה-VPC: vpc-038f0b441b45a3df3.

• הוספת תתי-רשתות (Subnets) וקישורן אותם ל: Availability Zones

שני Availability Zones שבחרנו: us-east-1a ו us-east-1b.

יצירת שתי תתי-רשתות פרטיות :

Private Subnet A :

ב us-east-1a-

CIDR 10.0.100.0/24

ID: subnet-01fc42602a50e7d43.

Private Subnet B:

ב us-east-1b-

CIDR 10.0.101.0/24

ID: subnet-0b3ed6ab940c8527b.

יצירת שתי תתי-רשתות ציבוריות :

Public Subnet A :

us-east-1a

CIDR 10.0.200.0/24

ID: subnet-05086541c0ded93e4.

Public Subnet B :
us-east-1b
CIDR 10.0.201.0/24
ID: subnet-0723bf128ea640c2f.

• **לאחר מכן הגדרנו Internet Gateway שמאפשר גישה לאינטרנט :**

יצירת Internet Gateway עם ID: igw-04cae497d30f68e19 והצמדתו ל-VPC-

הגדרת: NAT Gateway

יצירת NAT Gateway בתת-רשת ציבורית, עם :

NAT ID: nat-06e5c0efe6175f460

Private IP: 10.0.200.234

Public IP: 54.81.177.255.

• **הגדרת: Routing Tables**

הגדרנו טבלת ניתוב בצורה הבאה :

הגדרת טבלאות ניתוב (Routing Tables) עבור תתי-הרשתות הציבוריות והפרטיות.
חיבור ה-Internet Gateway לטבלת הניתוב של תתי-הרשתות הציבוריות.
חיבור ה-NAT Gateway לטבלת הניתוב של תתי-הרשתות הפרטיות.

• **הגדרת: Security Group**

יצירת Security Group עם ID: sg-051e10bfdadbf9bb2.
הגדרת גישה ל-SSH ו-ICMP מ-0.0.0.0/0.

4.7.6 יצירת EKS ו-Worker nodes

בשלב זה השתמשנו ב-Role וה-Trust Policy שיצרנו עבור ה- EKS ,

יצרנו EKS והגדרנו Node Group עם הפרטים הבאים :

Name : xrd-terraform-2d601e34

region : us-east-1 (האזור שבו ייווצר ה-cluster.)

שיוך של הרשתות הפרטיות והציבוריות שיצרנו ב-VPC

(xrd-terraform : nodegroup-name שם קבוצת ה-Worker Nodes.)

(node-type : EC2 instances שישמשו כ-Worker Nodes.)

בעצם ברגע שהגדרנו את הפרטים הללו , יצרנו קלאסטר עם EC2 שעליהם נרים את ה- XRd.

יצירת EKS

The screenshot shows the AWS EKS Clusters page. At the top, there's a search bar and buttons for 'Refresh', 'Delete', and 'Add cluster'. Below is a table with columns: Cluster name, Status, Kubernetes version, Support period, Upgrade policy, and Created. One cluster is listed: 'xrd-terraform-2d601e34' with status 'Active', version '1.27', and support until 'July 24, 2025'.

Cluster name	Status	Kubernetes version	Support period	Upgrade policy	Created
xrd-terraform-2d601e34	Active	1.27	Extended support until July 24, 2025	Extended	4 hours ago

להחליף תמונה בהזדמנות

The screenshot shows the AWS Resources page for the US East (N. Virginia) Region. It displays a grid of resource counts for various EC2-related services.

Resources	
Instances (running)	4
Auto Scaling Groups	0
Capacity Reservations	0
Dedicated Hosts	0
Elastic IPs	4
Instances	4
Key pairs	2
Load balancers	0
Placement groups	1
Security groups	5
Snapshots	2
Volumes	7

The screenshot shows the AWS EC2 Instances page. It lists four running instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
xrd-terraform-...	i-0cfce32cce1d53b1d	Running	t3.nano	3/3 checks passed	View alarms +	us-east-1a
xrd-terraform-...	i-00bc22f65ddd5f351	Running	m5.large	3/3 checks passed	View alarms +	us-east-1a
xrd-terraform-...	i-052c9458d8b99538e	Running	m5.2xlarge	3/3 checks passed	View alarms +	us-east-1a
xrd-terraform-...	i-05469eb30e1e357db	Running	m5.2xlarge	3/3 checks passed	View alarms +	us-east-1a

4.7.7 Monitoring and logging

ניטור מתייחס לתהליך מעקב רציף אחר ביצועים ומצב המערכת. מטרת הניטור היא לספק תובנות ו"תמונת מצב" כללית על המדדי ביצועים, שימוש במשאבים ותקלות.

Logging מתייחס לתהליך של איסוף ואחסון של יומני רישום (Logs) שהם רישומים מפורטים של פעולות ואירועים שמתרחשים במערכת. לוגים מאפשרים להבין בצורה עמוקה מה התרחש במערכת בזמן מסוים, מה קרה בזמן תקלה, ואילו פעולות בוצעו.

חשיבות הניטור:

זיהוי מוקדם של בעיות: ניטור מסייע בזיהוי מוקדם של בעיות כמו עלייה חדה בשימוש במשאבים, שגיאות במערכת, או תקלות ברשת.

אופטימיזציה של משאבים: על ידי ניתוח נתונים לאורך זמן, ניתן להבין כיצד לנצל את המשאבים בצורה מיטבית ולמנוע בזבוז.

אוטומציה והתראות: בעזרת התראות אוטומטיות ניתן להגיב במהירות לתקלות ולעומסים.
חשיבות logging :

חקר תקלות (Troubleshooting): לוגים מאפשרים להבין מה קרה במערכת בזמן בעיה או תקלה.

מעקב אבטחה: רישום לוגים מאפשר לגלות פעילויות חריגות או גישה לא מורשית.

בקרה וביקורת: מאפשר תיעוד של פעולות משתמשים ושירותים לצורך בקרה ועמידה בתקני אבטחה ורגולציה.

קיימים 2 כלים שהשתמשנו בהם לטובת ניטור על התשתית ועל תקלות :

Amazon CloudWatch (1)

שירות הניטור העיקרי ב AWS-שמשפק נתוני ניטור בזמן אמת על מגוון משאבים כמו, EC2, RDS, EKS, ועוד.

CloudWatch מציג **מדדים (Metrics)** כגון שימוש ב CPU , תעבורת רשת, ועוד.

ניתן ליצור **התראות (Alarms)** שמופעלות כאשר ערך של מדד מסוים עובר סף מוגדר. לדוגמה, התראה שמופעלת כאשר שימוש ב CPU-עובר 80%.

CloudWatch מאפשר גם יצירת **Dashboards** לניטור כולל של מערכת מורכבת.

CloudWatch Logs מאפשר להגדיר **התראות על בסיס חיפושים בלוגים** כדי לקבל התראות כאשר מתרחשות שגיאות או אירועים חריגים.

AWS CloudTrail (2) :

CloudTrail מתמקד בלוגים של פעילויות API וניהול גישה. הוא רושם כל קריאה לשירותי AWS כולל מי ביצע את הפעולה, מתי ואילו משאבים הושפעו.

CloudTrail חשוב במיוחד לניטור אבטחה, כיוון שהוא מספק תיעוד של פעולות בקרה וביקורת.

X.דוגמא לרשימת היסטורית אירועים בשירות CloudTrail

CloudTrail

×

Dashboard

Event history

Insights

Lake

Dashboard

Query

Event data stores

Integrations

Trails

Settings

Pricing

Documentation

Forums

Event history (50+) info

Event history shows you the last 90 days of management events.

Lookup attributes

Read-only

Q false

×

Filter by date and time

<

1

2

...

>

⚙

<input type="checkbox"/>	Event name	Event time	User name	Event source	Resource type	Resource name
<input type="checkbox"/>	DeleteDashboards	October 27, 2024, 21:12:26 (UT...	STAV_HIT_Project	monitoring.amazonaws.com	-	-
<input type="checkbox"/>	PutDashboard	October 27, 2024, 21:05:00 (UT...	STAV_HIT_Project	monitoring.amazonaws.com	-	-
<input type="checkbox"/>	ConsoleLogin	October 27, 2024, 21:03:42 (UT...	STAV_HIT_Project	signin.amazonaws.com	-	-
<input type="checkbox"/>	ConsoleLogin	October 27, 2024, 09:38:17 (UT...	root	signin.amazonaws.com	-	-
<input type="checkbox"/>	DeleteVpc	October 27, 2024, 02:43:19 (UT...	ubuntu_tomer2	ec2.amazonaws.com	AWS::EC2::VPC	vpc-038f0b441b45a3..
<input type="checkbox"/>	DeleteSubnetCidrRes...	October 27, 2024, 02:43:18 (UT...	ubuntu_tomer2	ec2.amazonaws.com	-	-
<input type="checkbox"/>	DeleteSubnet	October 27, 2024, 02:43:18 (UT...	ubuntu_tomer2	ec2.amazonaws.com	AWS::EC2::Subnet	subnet-01fc42602a50..
<input type="checkbox"/>	DeleteSubnet	October 27, 2024, 02:43:18 (UT...	ubuntu_tomer2	ec2.amazonaws.com	AWS::EC2::Subnet	subnet-0b3ed6ab940c...

6. פרק 5 - מסקנות והצעות עבודה להמשך

הפרויקט שביצענו מדגים באופן מעשי את עוצמתה ויתרונותיה של טכנולוגיית ה-CNF (Cloud Native Functions) במתן מענה לאתגרים המודרניים של ספקי שירותי התקשורת (ISP).

התשתית שהקמנו משלבת טכנולוגיות וירטואליזציה מתקדמות מבוססת קונטיינרים, אשר מאפשרות להעניק שירותי רשת בצורה גמישה, סקלאבילית ומהירה – יכולות קריטיות במיוחד בעידן בו דרישות המשתמשים, היקפי התעבורה ואופי התקשורת צומחים ומשתנים במהירות.

במהלך הפרויקט, נבנתה ארכיטקטורה מורכבת המדמה רשת של ספק תקשורת אמיתי, תוך פריסה של רכיבים וירטואליים (XRd routers) על גבי תשתית ענן ציבורית מבוססת AWS. הפריסה בענן אפשרה לנו לממש אתרים גיאוגרפיים מרוחקים ברחבי ארה"ב, ולבחון את הפרוטוקולים ואת המדדים המבצעיים בתנאי שטח מציאותיים לגמרי. הוכחנו את בשלות הטכנולוגיה על ידי מימוש פרוטוקולי ניתוב מתקדמים (כגון GRE, IS-IS, ו-MPLS).

היכולת להשתמש בטכנולוגיות כמו VRF ו-MPLS VPN אפשרה לנו לספק הפרדה לוגית מלאה בין לקוחות ויישומים, תוך שמירה על רמת אבטחה גבוהה והגנה מפני זליגת מידע.

בעזרת תשתית קונטיינרים מבוססת Kubernetes, הצלחנו להקים סביבה רשתית אמينة, יעילה, ובעלת ביצועים גבוהים. השגנו יכולת להגדיל או להקטין את היקף השירותים בהתאם לעומסים משתנים (Auto-scaling) עם שרידות גבוהה.

ביצענו מדידות מדויקות והצגנו מדדים מרשימים של ביצועים גבוהים מאוד וזמני תגובה קצרים, תוך שימוש ברכיבים וירטואליים בלבד – ללא תלות בחומרה פיזית. תוצאה זו מוכיחה את יכולתה של ארכיטקטורה מבוססת CNF לתת מענה לדרישות המודרניות של ספקי תקשורת.

ארכיטקטורה זו מאפשרת לספקים להתמודד עם מגוון אתגרים עסקיים וטכנולוגיים כמו נעילת ספקים (Vendor Lock-In), שמנעה גמישות במעבר בין יצרנים שונים, התמודדות עם שינויים דינמיים בעומסים ועם הצורך המתגבר באוטומציה וביכולת התאמה מהירה לשינויים בשוק. המענה המתואר ומאפשרת להם להתאים את הרשת לצרכים דינמיים בצורה חלקה, תוך ניצול משאבים אופטימלי והקטנת העלויות התפעוליות.

כיוונים לעתיד: לקראת רשתות אוטונומיות בשילוב CNF ו-AI. המגמה העולמית בתשתיות תקשורת נעה לעבר רשתות אוטונומיות המשלבות טכנולוגיות וירטואליזציה מתקדמות, ובראשן CNF (Cloud-Native Network Functions), יחד עם יכולות AI ו-ML (למידת מכונה) כדי להתמודד עם המורכבויות והדרישות ההולכות וגדלות של עידן ה-G5 וה-G6. המעבר לטכנולוגיות כמו CNF, מאפשר לספקי תקשורת לנהל את הרשתות בצורה גמישה, דינמית ויעילה יותר, בזכות היכולת להטמעת יכולות AI ברשתות תוכנותיות אלו. שילוב זה מוביל ליצירת רשתות אוטונומיות, המסוגלות לפעול, לנטר, ולהתאים את עצמן באופן עצמאי בתנאים משתנים.

לסיכום הפרויקט מדגיש את הפוטנציאל העצום של טכנולוגיות CNF במימוש רשתות תקשורת מתקדמות, גמישות, ובעלות עלויות תפעול מופחתות. כך ספקי תקשורת יכולים להגיב בצורה מהירה ויעילה לצרכים המשתנים של השוק, להגדיל את יכולות האוטומציה ולהעניק חוויית משתמש איכותית המותאמת לעידן ה-IoT וה-G5.

הוכחת יכולת זו בארכיטקטורה שבנינו מהווה עדות ליתרונותיה של טכנולוגיית CNF. השילוב עם טכנולוגיות AI ו-ML תאפשר לספקי תקשורת להקים רשתות אוטונומיות ואינטליגנטיות המספקות התאמה מהירה לשינויים, חוסן גבוה ויכולת התאוששות מהירה מתקלות ובכך לשפר את רמת השירות ללקוחות וכל זאת תוך ניצול מיטבי של משאבי המחשוב.

7. ביבליוגרפיה

ביבליוגרפיה

Holistic Network Virtualization and Pervasive Network Intelligence for 6G

המאמר הזה סיפק תובנות יסודיות לגבי המעבר לוירטואליזציה של רשתות והחשיבות של בינה מלאכותית לשילוב ברשתות הדור ה-6G. הוא הדגיש את הצורך בשילוב של פונקציות רשת וירטואליות עם יכולות AI, תוך הגדרת מסגרת חזונית ליצירת רשתות אוטונומיות. רעיונות אלו היו בסיסיים בעיצוב ההבנה שלנו על דרישות הדור הבא וביסוס הארכיטקטורה בפרויקט שלנו. מקור: "Holistic Network Virtualization and Pervasive Network Intelligence for 6G" (IEEE, 2023)

Cloud Native 5G Virtual Network Functions: Design Principles and Use Cases

מסמך זה עסק בעקרונות עיצוב לפונקציות רשת מבוססות ענן (CNF) המותאמות במיוחד לרשתות 5G, תוך שימת דגש על גמישות וסקלאביליות המושגות באמצעות ארכיטקטורת קונטיינרים. הוא העניק לנו תובנות על שיטות עבודה מיטביות לפריסת CNF בענן, ותרם להחלטתנו להשתמש בתשתית AWS כדי לספק את הדרישות הסקלאביליות והגמישות להקמת רשת ISP.

מקור: "Cloud Native 5G Virtual Network Functions: Design Principles and Use

Cases

Network Function Virtualization (NFV) : תזה לתואר שני

התזה הזו הציעה סיקור מקיף על NFV, תוך פירוט של עקרונות יסודיים ושימושים מעשיים. היא עסקה במעבר מפונקציות מבוססות חומרה לפונקציות רשת וירטואליות על גבי חומרה סטנדרטית, וסיפקה הקשר חשוב להבנת המגבלות של טכנולוגיות ישנות שה-CNF נועד להתגבר עליהן. הדיון באתגרים וביתרונות של NFV סייע לנו להגדיר את הגישה המודרנית עבור ספקי תקשורת בפרויקט זה.

מקור: תזה של מנטנה - "Network Function Virtualization (NFV)"

VNF and CNF Placement in 5G: Recent Advances and Future Trends

המחקר הזה התמקד במיקומים אסטרטגיים וארכיטקטורות של פונקציות רשת וירטואליות (VNF) ופונקציות מבוססות ענן (CNF) במסגרת רשתות 5G, תוך הדגשת יתרונות תפעוליים של CNF על פני VNF. המחקר דן ביתרונות בתחום יעילות המשאבים והגמישות, שהיוו קריטיים להשגת ביצועים גבוהים וסקלאביליות בתוך רשת ISP מבוססת בפרויקט.

מקור: "VNF and CNF Placement in 5G: Recent Advances and Future Trends" (IEEE) ACP-WG-S מדידת שגיאות PER ו-BER

מסמך זה הציג מתודולוגיות טכניות למדידת Bit Error Rate (BER) ו-Packet Error Rate (PER), שהיו קריטיות לאימות ביצועי הרשת בפרויקט. התקנים עבור שיעורי השגיאות סייעו לנו לקבוע ספי ביצוע לאיכות הרשת ולמדידת אמינות ויעילות הארכיטקטורה שיצרנו.

מקור: "ACP-WG-S WP04 - Error Measurement in PER and BER" מדידת איכות השירות לשירותי אינטרנט ניידים

המחקר הזה סיפק מסגרת מפורטת למדידת איכות השירות (QoS) ברשתות ניידות, כולל מדדים חשובים כגון Jitter, שהיה ואובדן מנות. טכניקות המדידה שנדונו שימשו אותנו בגישתנו להערכת QoS לרשת מבוססת CNF, תוך התמקדות בהשגת השהיות מינימליות ורמות Jitter עקביות לביצועים מיטביים.

מקור: "Measuring Quality of Service for Mobile Internet Services" (2016, ICSITech) Spirent Communications White Paper : מדידה מדויקת של Jitter

נייר העמדה הזה הציע הסבר טכני על Jitter ועל השפעתו על ביצועי הרשת, כמו גם שיטות מדידה שונות. תובנות אלו היו חיוניות להבנת תפקידו של Jitter בחוויית המשתמש ולהבטחת עמידת הרשת בסטנדרטים קפדניים של ביצועים, במיוחד עבור יישומים רגישים להשהיה.

מקור: "Measuring Jitter Accurately" (Spirent Communications, 2007)
כל אחד מהמקורות הללו תרם תובנות תיאורטיות ומעשיות אשר תמכו בתכנון הארכיטקטורה, בגישת הפריסה ובאימות הביצועים של הפרויקט. השילוב של CNF במודל רשת ISP שלנו נשען על תובנות אקדמיות ותעשייתיות אלה, ומדגים את היכולת של תשתיות ענן קונטיינריות לספק מענה לצרכים המודרניים של ספקי תקשורת.

IEEE – Cloud_computing_-_concepts_architecture_and_challenges

IEEE – Cloud_Computing_Opportunities_and_Challenges

Cloud_Computing_Architecture_Vision_Challenges_Opportunities_and_Emerging_T

IEEE – rends

*קבצי קונפיגורציה של הרכיבים יצורפו באופן דיגיטלי