



# ת"ע צוות מידגארד 2023

DevOps is Dead, Long Live Platform Engineering

- נעשה מחקר מעמיק על מנת לברר מה הידע והיכולות הנדרשים מצוות תוכנה על מנת להצליח להשלים את המעבר ל OpenShift.
- הצוות התנסה וחקר כלים אשר מזוהים עם DevOps.
- שאיפת הצוות היא לעבוד עם צוותי תוכנה בתהליך רציף שבו עוזרים להם להתאים את הארכיטקטורה שלהם לסביבה שהיא Cloud-Native
- הגישה הנוכחית - להעלות כמה שיותר אפליקציות של מפתחים במערך על OpenShift, תוך מעקב צמוד של היכולות שלהם.



# הלקוחות הקיימים כרגע

באופן כללי, לקוחות מתחלקים לשניים:

רחוקים ממודרניזציה - אלו לקוחות אשר יחסית רחוקים מעבודה מודרנית.

קרובים למודרניזציה - אלו לקוחות שעובדים בצורה דומה יחסית לאזרחות.

שני הסוגים האלו מביאים איתם אתגרים שונים בכל הקשור למעבר לתשתית, ובשני המקרים אתגרים אלו מקשים עלינו מאוד לייצר לקוחות במהירות.



## הפערים בגישה הנוכחית

- **תלויות רבות** - מסתמכת יותר מדי על שיתוף הפעולה של צוותים אחרים.
- **צוותים לא עצמאיים** - אנחנו לא מייצרים צוותי פיתוח עצמאיים.
- **הצוות אינו בשל לזה** - על מנת להתקדם בגישה זו, לצוות דרושות יכולות שכרגע אין לו.
- **מחסור בכוח אדם** - אין מספיק אנשים בשביל לפתח כלים ולחזק את היכולות על OpenShift, ולתת בו זמנית מענה רציף לדרישות של צוותים שונים.
- **לא מייצרת מספיק ערך** - ההערכות שסופקו קובעות שנדרשים 90 ימים להעביר צוות לתשתית החדשה. אלה רק ארבעה צוותים בשנה.



DevOps is Dead, Long Live



**Platform  
Engineering**



# הגישה המוצעת - Platform Engineering

השלב הבא של רעיון ה DevOps כעת דוהר קדימה באזרחות, Platform Engineering. לפי הגישה הזו, משימתנו כצוות (וכמדור) היא לפתח כלים ולהציע שירותים שיהפכו מפתחים לעצמאיים, ויעזרו להם לבצע את עבודתם במדויק מבלי להיות תלויים באף אחד חוץ מבעצמם.

מבחינה פרקטית, עלינו לשדרג אפליקציות ושירותים קיימים ולהקים חדשים, אשר ירוצו על ה OpenShift, ולהציע אותם למפתחים. נבחר פרויקטים שרלוונטיים לכמות גדולה של מפתחים, ונשווק אותם כפרוייקטים אשר מתאפשרים בזכות OpenShift. בכך, שלב א' של הפרויקט מתמקד אך ורק ביכולותיו הטכנולוגיות של הצוות. אנחנו לא צריכים להסתמך על הידע הטכנולוגי או שיתוף פעולה גבוה מצוותים אחרים.



## היתרונות של הגישה הזו

- **מייצרת לקוחות מהר** - בתחילת הדרך אנחנו מחפשים safe, quick wins.
- **תלויים רק בעצמנו** - הצוות תלוי אך ורק ביכולותיו המקצועיות על מנת לשחרר כלים ושירותים ללקוחות.
- **מייצרים יכולות לצוותים** - נייצר כלים אשר מעניקים לצוותי פיתוח עצמאות וכוח
- **מבוססת על יכולות קיימות** - כל המשימות הקרובות מתכתבות היטב עם יכולות שכבר קיימות בצוות.
- **גמישות גבוהה** - בכל שלב בפרויקט קיימת האופציה לשקול מחדש את הצעד הבא, לפי צורך של לקוחות ומידע חדש שאנו עשויים לקבל
- **מתאים לכוח האדם הנוכחי** - המשימות הקרובות מתאימות לכוח האדם שיש לנו כרגע.

# איך נראית השנה הקרובה?

## מתכוננים ל OpenShift

עם התחזית ש OpenShift יגיע ל Production בסוף הרבעון הראשון, אנחנו יכולים לעבוד על כלים שירוצו עליו מייד עם הקמתו.

Q2

Q1

## מתחילים לצבור לקוחות

הרבה יותר דגש על עבודה מול הלקוחות, הנגשת הכלים שלנו לציבור הרחב תוך שמירה על מהימנות

## משתמשים בידע שרכשנו

העבודה מתפתחת בהתאם לצרכים שוטפים של לקוחות, כמו גם חיזוק היכולות בראייה עתידית. הזדמנות טוב לקבל החלטות ולשקול צעדים

Q3

Q4

## מפתחים עצמאיים יותר ויותר

בשלב הזה אנחנו שואפים להתחיל להגיש למפתחים כלים אשר יאפשרו להם להריץ את האפליקציות שלהם על ה OpenShift



## משימות לרבעון הראשון - בחירת הכלי לפי צרכי הלקוח



Knative -  
Serverless  
Functions



Automaton -  
Ops Delivery  
for Souf



Kubevirt -  
Virtual  
Machines on  
K8S



Runner as a  
Service



## מדד לבחירת הכלים המתאימים ביותר לעבודה

- דוקמנטציה קיימת ומועילה
- מהי כמות הלקוחות ביחידה שזקוקים לכלי?
- מהו מחיר ההטמעה שלו? כמה זמן הוא ייקח?
- מהו הערך שהוא נותן ליחידה?
- כמה הוא מתכתב עם השאיפה שלנו ל Platform Engineering?
- מהי דרישת הזמינות שלו? כמה רחב מרחב הטעויות שלנו?
- תמיכה מ RedHat



# מה אנחנו דורשים מכלי לפני שהוא עולה ל Production?

- תהליך שלם אוטומטי לחלוטין לשדרוג הגרסא שלה (כולל הלבנות אוטומטיות לגרסאות מבחוץ)
- GitOps לחלוטין
- ידע רווח בצוות לגבי ניהול הכלי + דוקומנטציה מפורטת
- קיים על גבי סביבת Staging בצורה זהה ל Production. קיים בסביבת Dev.
- נמצא כחלק מ Monitoring.
- עבר בדיקות עומס לפי הציפיות.
- הוגדר היטב מהו אופן השימוש של צוותים בו, ובאילו דרישות אנחנו יודעים לעמוד, כמו גם ווידוא שלא מנסים להשתמש בו מעבר לדרישות אלו.
- שם שימשוך לקוחות!



# איך משווקים את הכלים למפתחים?

הגיע הזמן שלמדור ה DevOps יהיה פורטל משל עצמו, אליו ייפנו לקוחות אשר ירצו להשתמש בשירותים שלנו, לשמוע מה אנחנו מתכננים או אפילו מה חדש בעולם ה DevOps

לזה בדיוק הקימו את האפליקציה Backstage, אפליקציה שנועדה להיות פלטפורמה שלמה לכלים שמוגשים למפתחים בארגון.



## האתגרים לחצי השנה הקרובה

- חיזוק שיתוף הפעולה עם מדור תשתיות בכלל וצוות Platforms בפרט.
- עלינו לחזק את ההיכרות שלנו עם צרכים של כל צוותי הפיתוח ביחידה (לא רק אלו שהעבודה העיקרית שלהם נמצאת בתוך היחידה), בהתאם לכלים החדשים שאנחנו שואפים לשווק.
- GitLab דורש תשתית חדשה, אשר צריכה להתקיים (בשאיפה) על גבי קלאסטר OpenShift נפרד.
- Keycloak הוא עדיין Technical Debt שאנחנו שואפים להיפטר ממנו.
- קבלת החלטה לגבי Rancher, האם מעבירים הכל ל- OpenShift או שומרים עליו רץ כדי להגביר את האמינות שקיימת במערכת ה OpenShift?



# החזון לקראת הרבעון הרביעי ושנת 2024

אחרי כמעט שנה שבה עבדנו באופן צמוד לדרישות הלקוח ופתרנו דרישות מסובכות על גבי Kubernetes, יוכל הצוות להתקדם לשלב הבא, בו הוא מרחיב את יכולות ה- OpenShift עצמו מהבסיס, ומאפשר לאנשים להעלות את האפליקציות שלהם על גבי הפלטפורמה.

אנחנו לא מגבילים שום דבר אצל המפתחים, כל אחד מוזמן להיעזר בכלים שלנו, ואם ירצה להקים את הכלים שלו בעצמו כראות עיניו, הוא מוזמן לעשות זאת.

תשתית ה OpenShift תהיה מהימנה ומוכרת, אנשים רבים ישתמשו בה למטרות Production ועוד רבים ירצו להיעזר בה ובכלים אשר קיימים עליה על מנת לחזק את יכולותיהם.