### **Opening Slide**

Introduction to Docker  
How Docker Revolutionized Development and Deployment

### **Slide 1: Context of the Presentation**

Title: Why Docker?

* **What is Docker?** - A brief introduction to Docker and its role in software development.
* **Why Docker?** - The fundamental problems Docker addresses
* **Virtual Machine VS Docker**- The Pros of Docker
* **Summary**

-מטרה של השיעור להבין לעומק למה דוקר זה כלי כל כך מהפכני בעולם הפיתוח

-נתחיל בלהבין מזה דוקר?

-החלק הכי חשוב בהתמודדות ולמידה של טכנולוגיה כלשהיא, היא להבין לשם מה היא נוצרה או איזה בעיה היא פותרת, נענה על השאלה, למה.

-נדבר על היתרונות של דוקר מול מכונה וירטואלית

-סיכום

### **Slide 2: What is Docker?**

Title: What is Docker?

* **Virtualization Programm**
* **Makes developing and deploying applications much easier**
* **Packages application with all the necessary dependencies, configuration, system tools and run time.**

-תוכנת וירטואליזציה שמקלה על פיתוח ודיפלוימנט.

-מקל על דיפלוימנט ופיתוח בכך שהוא דוחס את האפליקציה לתוך קונטיינר, משהו שמכיל את כל הדברים שהאפליקציה זקוקה בשביל לרוץ, כמו הקוד של האפליקציה, השירותים השונים והגדרות סביבה.

-דוגמה, להראות לפי התמונות איך הכל נכנס פנימה לקונטיינר

-קל להפיץ את הקונטיינר

### 

### **Slide 3: Life Before Docker**

* **Environment Inconsistency:** “It Works for me so why does it not work for you?” -Applications often malfunction when moved between different environments.
* **Difficult Application Deployment:** Traditionally, deploying an application involved manually installing all its dependencies on the target server.
* **Version Control Challenges:** Managing different versions of an application and its dependencies could be cumbersome

-בעבר נדרש בניית סביבת עבודה עבור כל מפתח על מערכות ההפעלה שלהם, דוגמה, פייטון בגרסה מסויימת או שירות מאגרי נתונים בגרסה מסויימת, ספריות ושירותים שונים וכדומה.

-מערכות הפעלה שונות דורשות צורות התקנה שונות, למשל ווינדוס ולינוקס

-יש המון שלבים בתהליך התקנה שכוללים הגדרות שונות, בנוסף יש שירותים שהתקינו והגדירו בשורות קוד רבות.

-כל השלבים הארוכים גורמים לסיכויים מאוד גבוהים לטעויות, לדברים שלא יעבדו כמו שצריך ושמשהו ישתבש

-ליצור סביבית עבודה הייתה עבודה סיזיפית במיוחד בסביבות יותר מורכבות

-תוכנת וואב פשוטה דורשת שירות מאגר נתונים, פייטון3, ספיירות פייטון שונות כמו pytest, flask ותתארו לכם מצב שאתם צריכים להתקין את כל הכלים האלה על המחשבים של עשרה מפתחים שונים.

-קושי נוסף, צריך גרסאות שונות של שירותים עבור עבודות שונות, יותר התנגשויות.

### **Slide 4: How Docker Solves the Problem**

Title: Docker to the Rescue

* Consistency Across Environments: Docker containers ensure the same environment from development to production.
* Simplified Deployment: Containers bundle the application and its dependencies
* Isolation and Dependencies: Each container has its own dependencies, avoiding conflicts.

-קונטיינר מכיל את כל מה שנדרש על מנת להריץ את האפליקקציה כולל הגדרות הסביבה, לכן אם יעבוד בסביסה אחת תעבוד גם באחרת.

-במקום שדיפלוייימנט ידרוש תהליך ארוך וסזיפי של הרמת סביבה מתאימה עובר הרצת האפליקציה, דוקר מאפשר הרצה בעזרת פקודה אחת פשוטה.

-כל קונטיינר מבודד מכיל את הדפנדסיס משלו, לכן מונע התנגשויות של גרסעות שונות בין דברים.

### **Slide 5: Docker vs. Virtual Machines**

Title: Docker vs. Virtual Machines

* Architecture:
  + VMs: Each VM runs on a hypervisor, with its own OS.
  + Docker: Containers share the host OS
* Resource Usage: Containers use fewer resources because they share the host OS kernel.
* Performance: Containers start and stop much faster than VMs.

-למה בעצם לא להישאר עם מכונה וירטואלית?

-דוקר מקבל את המשאבים שהוא זקוק ממערכת ההפעלה עליה נמצא לעומת מכונה וירטואלית שזקוקה למערכת הפעלה משלה

-מערכת הפעלהדורשת: זיכרון, משאבים, זמן פעולה.

-

### **Slide 6: Summary of Key Points**

Title: Summary

* Docker Packages application with all the necessary dependencies into a container
* long and complicated deployment process that led to many human errors.
* Docker Containers are easy to transfer isolated environments that are consistent and make deployment easier.
* Docker Container uses a host OS to run, so it's lighter, uses fewer resources, deploys faster, and is easily transportable compared to a VM.