סדנת Kubernetes

השלמה טכנולוגית 003



לפני שמתחילים

**מטרות הסדנה**

* הבנת מטרת ושירותי הKubernetes

**מבנה הסדנה**

כאן נרשום את סדר הסדנה

* הכרת Kubernetes ומטרתו
* שימוש בסיסי בKubernetes
* הכרת המושג Ingress
* הכרת המושג Pods

**לפני שמתחילים וודאו שעשיתם את הדברים הבאים:**

* התקנות - שירות Kubernetes מותקן ופועל.
* הכירות בסיסית עם Docker.

אז עכשיו כשהכל מוכן– בואו נצא לדרך!

מה זה Kubernetes?

Kubernetes היא פלטפורמת קוד פתוח לניהול עומסים ולשירותי תכולות (Containers).

קצת מושגים בסיסיים:

Kubectl - שורת הפקודה של Kubernetes. מאפשר להריץ פקודות בKubernetes.

Container - יחידת תוכנה האורזת קוד (והתלויות שלו) כך שאפליקציה רצה באופן מהיר ויעיל. בנוסף, במקרה של תקלה באחד הPodים, הContainer יכול להחליף אותו באחר זהה בקלות.

Cluster - שתי מכונות(וירטואליות או פיזיות) או יותר המחוברות יחדיו על מנת לעבוד כיחידה אחת.

Pod - קבוצה של Container אחד או יותר. בכל Pod, משאבי התקשורת והזיכרון משותפים לכלל הContainerים .

Service - דרך מופשטת לחשיפת אפליקציה הפועלת על סט Podים כשירות רשת.

Ingress – עצם API(ממשק הנגשת שירותים) שנותן גישה לService בתוך הCluster. בד"כ, עצם זה ייתן גישה בפרוטוקולים HTTP וHTTPS . כלומר, שמשתמש מהאינטרנט רוצה לגשת לService כלשהו בCluster, הIngress מאפשר לו את הגישה הזו.

Scaling- יצירת podים זהים נוספים שהשירות ייגש אליהם במקביל. נבצע scale כשנרצה שבזמני עומס בגישה לשירות, הservice לא יקרוס. ע"י scaling נקבל מספר podים זהים שישמשו את הservice.

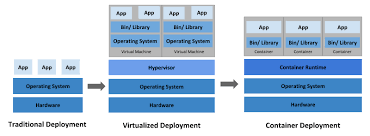
Docker Image - קובץ המורכב ממס' שכבות שמשומש על מנת להריץ קוד בContainer.

רקע על הנושא:

בTraditional Deployment(הפריסה המסורתית, השלב הראשון),על כל שרת פיזי, שכלל בתוכו חומרה ומערכת הפעלה, רצו מס' אפליקציות.

לאחר מכן, בVirtualized Deployment(פריסה וירטואלית, השלב השני), על כל חומרה ומערכת הפעלה, התקיימה שכבת HyperVisor - ווירטואליזציה. שכבה זו ניהלה את הVMים שנכללו בחומרה. בכל VM נכלל מערכת הפעלה משלו, מערכת קבצים ומס' אפליקציות.

בContainer Deployment(פריסת מכולות, השלב השלישי), השכבות התחתונות כוללות חומרה, מערכת הפעלה ומנהל זמני ריצה(בהקבלה לשכבת הווירטואליזציה ,בVirtualized Deployment). על בסיס השכבות התחתונות, ישבו מס' קונטיינרים שלכל אחד מהם יש מערכת קבצים נפרדת ואפליקציה נפרדת.

פלטפורמת הKubernetes מספקת מעטפת בעלת מגוון שירותים לContainer Deployment.

מה נוכל לעשות עם Kubernetes?

Kubernetes מספקת ללקוחותיה מס' שירותים, שביניהם: שירות איזון עומסים בין קונטיינרים, שירות ביזור משאבים, שירות rollback אוטומטי, שירות עלייה אוטומטית, שירותי אבטחת מידע ועוד..

סדנה מעשית

**תרגיל – Container deployment to Kubernetes**

**בתרגיל זה ניצור שירות שיציג למשתמש 'Hello world'.**

**דרישות פתיחה: image לדוגמא**

1. ניצור Pod. נעשה זאת בעזרת הפקודה הבאה:

**kubectl create deployment hello-node --image=gcr.io/google-samples/hello-app:1.0**

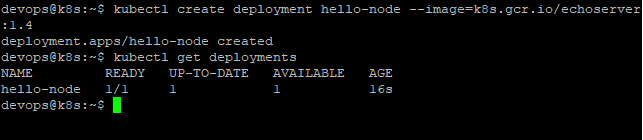
נסביר את הגורמים בפקודה:

* + Kubectl: הצהרה על פקודה של Kubectl.
  + create deployment: יצירת ה-pod, הרמתו והצהרה על מצבו הנוכחי (בריצה במקרה שלנו).
  + hello-node: השם שבחרנו לpod.
  + Image--: אופציה של הפקודה create, לה נציין את השם של ה-image להרצה ב-pod שיצרנו **(gcr.io/google-samples/hello-app:1.0).**

1. נבדוק שהDeployments נוצרו באמצעות הפקודה :

**Kubectl get deployments**

נסביר את הגורמים בפקודה:

* + Kubectl: הצהרה על פקודה של Kubectl.
  + get deployments: מחזיר רשימת deploymentים קיימים.  
    

1. נבנה שירות (service), באמצעות הפקודה:

**kubectl expose deployment <podname> --type=LoadBalancer --port=8080**

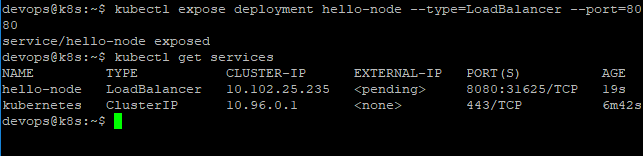
נסביר את הגורמים בפקודה:

* + Kubectl: הצהרה על פקודה של Kubectl.
  + expose deployment: חשיפת ה-pod כשירות (service) של Kubernetes.
  + type--: סוג השירות שאנחנו רוצים שKubernetes יבצע(בדוגמא זו, סוג השירות הוא מאזן עומסים)
  + port--: מס' הport שאנחנו רוצים שדרכו יגשו לservice.

1. נבדוק שהService נוצר באמצעות הפקודה :

**Kubectl get services**

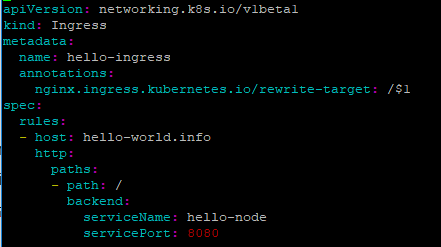
נסביר את הגורמים בפקודה:

* + Kubectl: הצהרה על פקודה של Kubectl.
  + get services: מחזיר רשימת serviceים קיימים.  
    

1. ניצור ingress שיאפשר לנו לגשת לשירות שבנינו:

ניצור קובץ example-ingress.yaml

נרשום בפנים את השורות הבאות:



כעת, ניצור את מקור הIngress ע"י הפקודה:

**kubectl apply –f example-ingress.yaml**

נקבל את הפלט הבא:

**ingress.networking.k8s.io/hello-ingress created**

נשתמש בפקודה **kubectl get ingress** כדי לוודא שנקבעה כתובת IP.

ונוסיף לקובץ etc/hosts/ את השורה הבאה:

ip> hello-world.info>

ובמקום <ip> נכתוב את הIP שקיבלנו מהפעולה הקודמת.

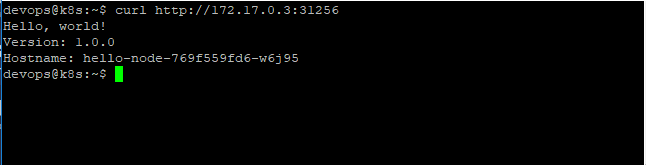
1. ניגש לשירות שבנינו:

במקרה שלנו אין דפדפן מותקן על המכונה ולכן נשתמש בפקודה הבאה:

**curl hello-world.info**

כאשר הIP הוא זה שה-ingress סיפק לנו.

* + curl: תחליף לשימוש בדפדפן(http client) לגישה לשירות שיצרנו.

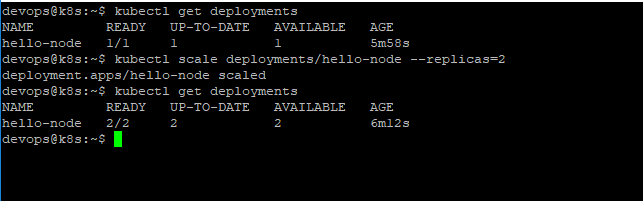


1. נבצע scale לhello-node ל-2 podים שישמשו את הservice:

**kubectl scale deployments/hello-node –replicas=2**

נסביר את הגורמים בפקודה:

* + deployments/hello-node: המקום במערכת בו שמור הdeployment לו אנחנו עושים scaling.
  + replicas--: מספר הpodים שנרצה.



מקורות מידע נוספים

* [**https://kubernetes.io/**](https://kubernetes.io/)
* [**https://www.redhat.com/en/topics/containers/what-is-kubernetes**](https://www.redhat.com/en/topics/containers/what-is-kubernetes)
* [**https://dzone.com/articles/a-beginners-guide-to-kubernetes**](https://dzone.com/articles/a-beginners-guide-to-kubernetes)
* [**https://enterprisersproject.com/article/2020/4/kubernetes-everything-you-need-know**](https://enterprisersproject.com/article/2020/4/kubernetes-everything-you-need-know)
* [**https://thenewstack.io/kubernetes-ingress-for-beginners/**](https://thenewstack.io/kubernetes-ingress-for-beginners/)