**הרצאה 5: Docker**

**שקף 29:**

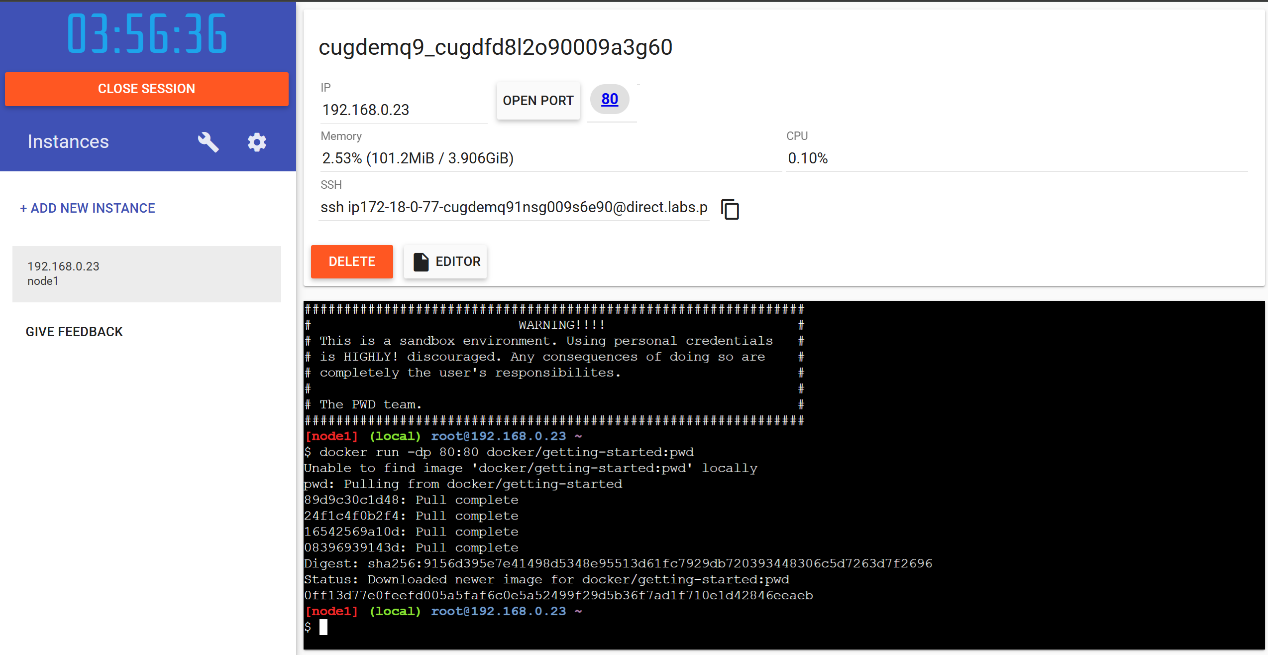
**פירוט המשימה:**

* כניסה לאתר <https://labs.play-with-docker.com/>

לחיצה על add new instance

הרצת הפקודות הבאות:

* % docker run -dp 80:80 docker/getting-started:pwd
* Docker run: Docker command to create and start new container from image
* -dp
* -d: detached mode. Runs the container in background
* -p: publish ports. Mapping a port on the host to a port in the container



מה מתרחש:

חיפוש התמונה מקומית

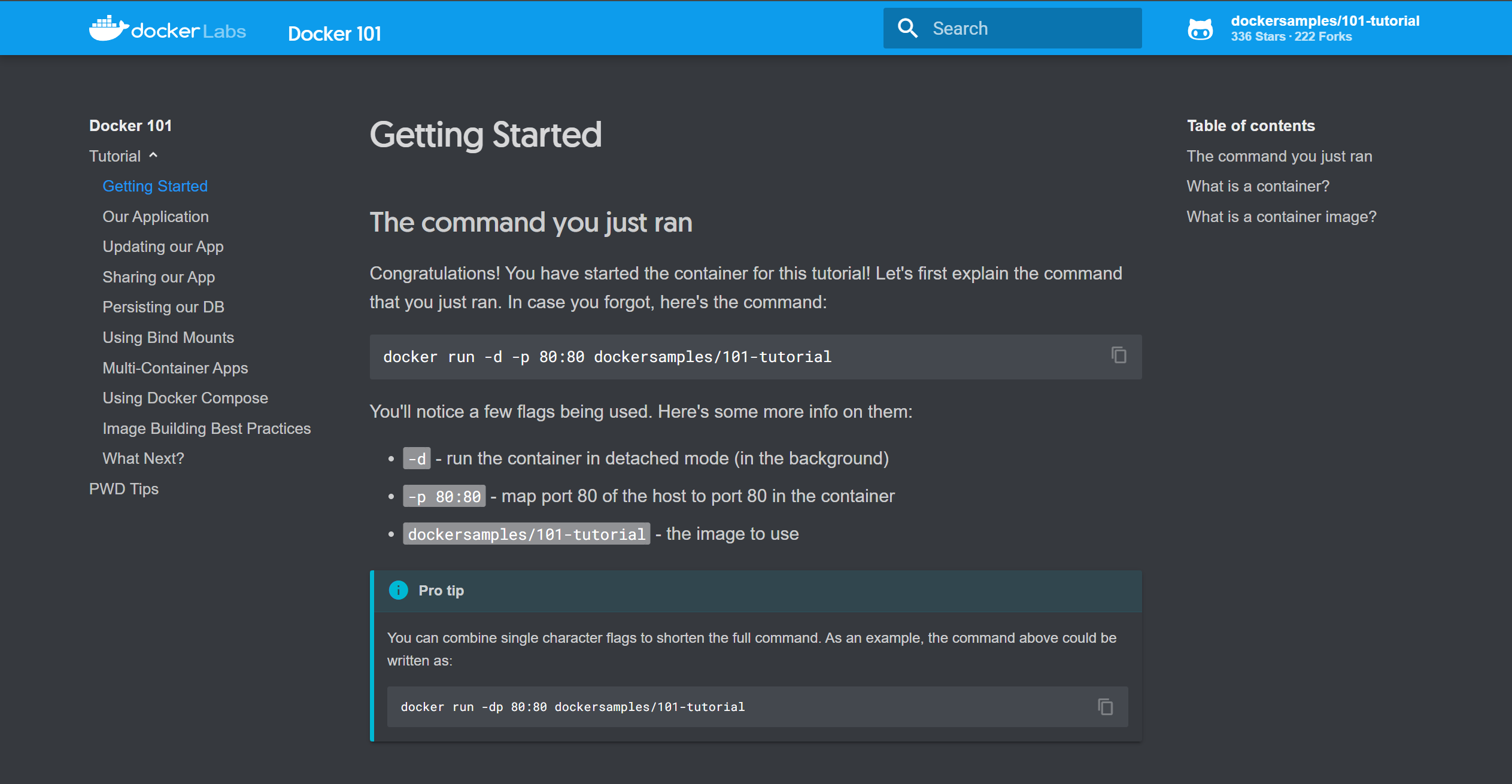
משיכת התמונה מהמאגר של Docker

התקדמות הורדת חלקי התמונה

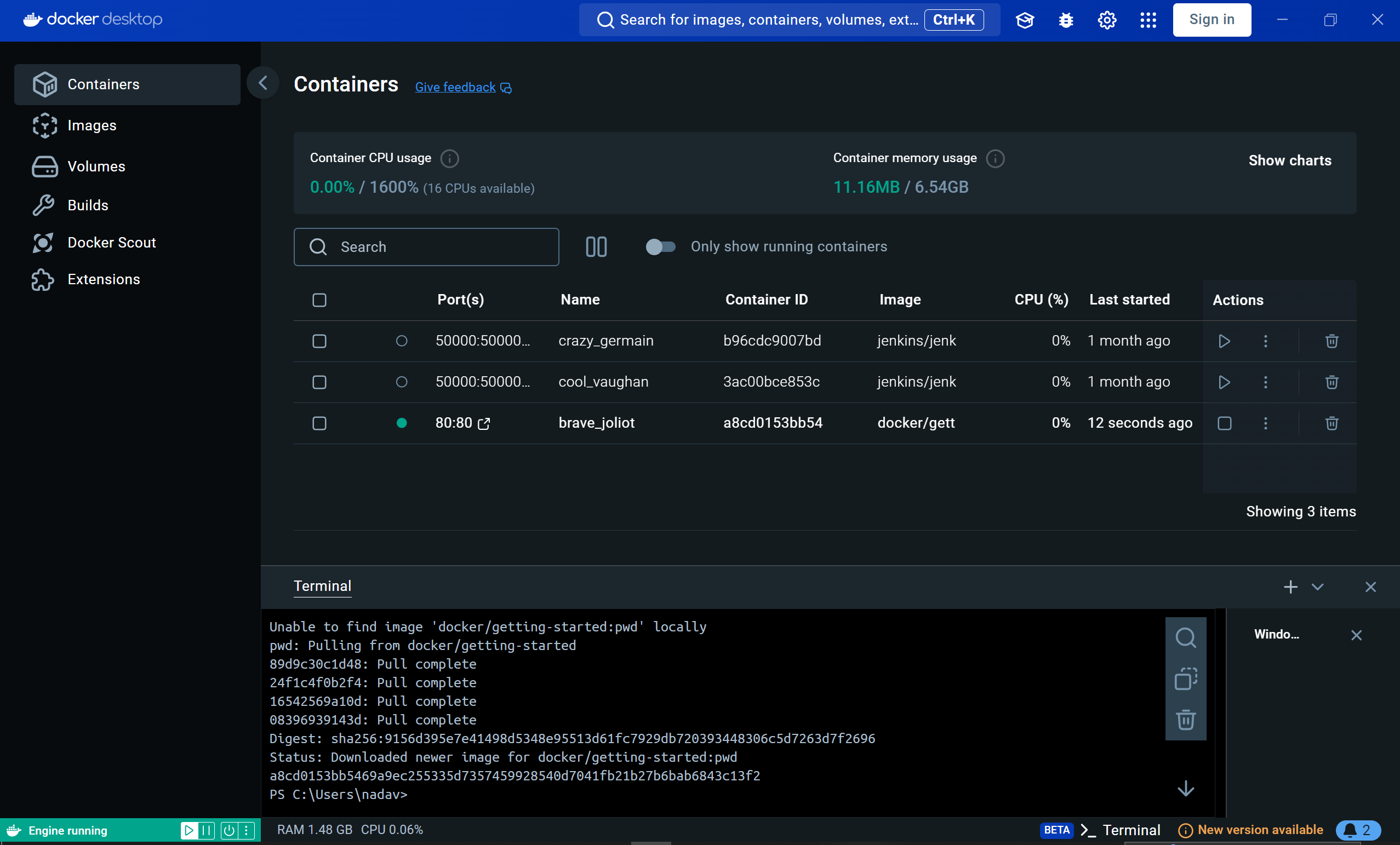
אימות התמונה לאחר הורדה

הורדה הושלמה בהצלחה

יצירת קונטיינר חדש והרצה

אפשר להשתמש בdocker ps כדי לראות את הקונטיינרים שרצים 

**שקף 33:**



נריץ את הפקודה docker run -dp 80:80 docker/getting-started:pwd מדוקר דסקטופ

נפתח את הקישור <http://localhost> ונקבל

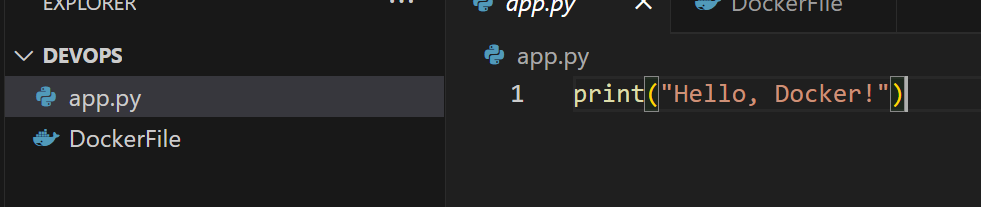


**שקף 45:**

ליצירת קונטיינר Docker שמריץ סקריפט פייתון פשוט, תחילה נצור קובץ בשם app.py ונוסיף אליו את השורה print("Hello, Docker!"), ולאחר מכן נשמור את הקובץ. כעת, נצור קובץ נוסף בשם Dockerfile באותה תיקייה, וכתוב בו את ההוראות הבאות FROM python:3.10-slim לשימוש בתמונה בסיסית של פייתון, COPY app.py /app.py כדי להעתיק את הקובץ לתוך הקונטיינר, ו- CMD ["python", "/app.py"] כדי לקבוע את הפקודה שתופעל בעת הרצת הקונטיינר. לאחר מכן, יש לבנות את התמונה באמצעות הפקודה docker build -t hello-docker . ולוודא שהתמונה נבנתה בהצלחה.

כדי להריץ את הקונטיינר ולהציג את הפלט, השתמש בפקודה docker run hello-docker, ואם הכול תקין, יוצג הפלט "Hello, Docker!".

קוד app.py:

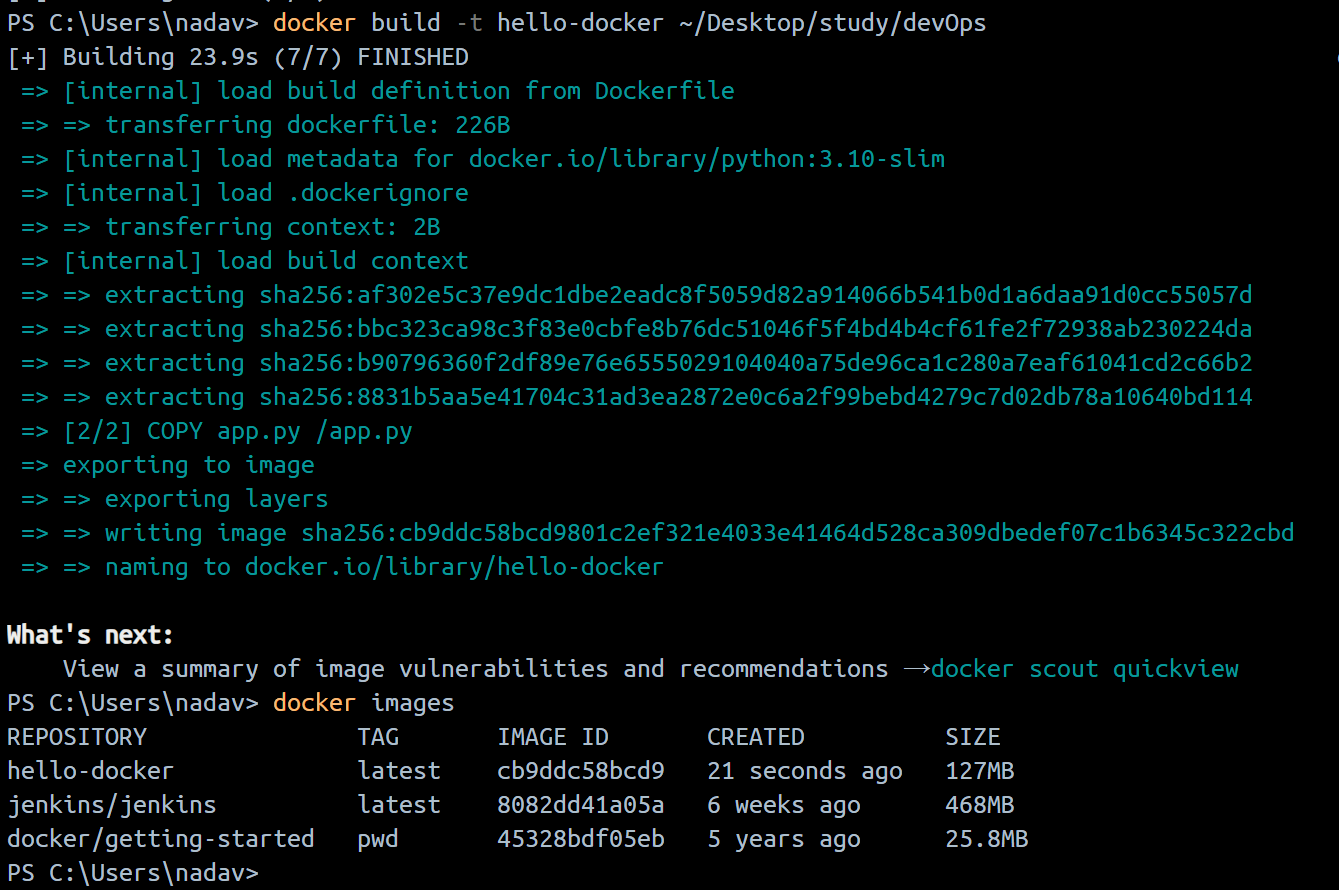


קובץ Docker:

תמונה שמכילה טקסט, צילום מסך, גופן, תוכנה

התיאור נוצר באופן אוטומטי

Image build:



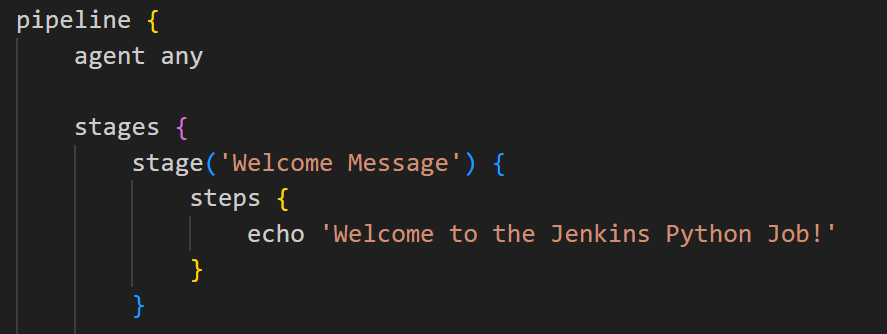
הרצת הקונטיינר:

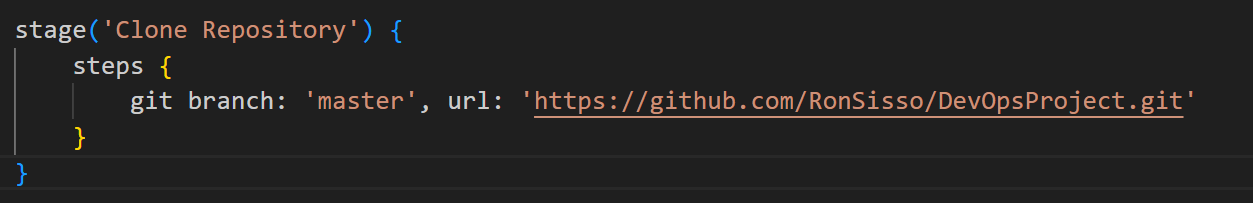
תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

**הרצאה 6:**

**שקף 19:**

* Create Jenkinks Job, which do the following:
* 1. Print welcome message
* 
* 2. Installs python & verify python installation with “python --version"
* תמונה שמכילה טקסט, צילום מסך

  התיאור נוצר באופן אוטומטי
* 3. Check-out your repository from master branch
* 4. Runs unit-tests on your code
* תמונה שמכילה צילום מסך, גופן, קו, טקסט

  התיאור נוצר באופן אוטומטי
* 5. Sends email for every run result

תמונה שמכילה טקסט, צילום מסך, גופן, תוכנה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

**הרצאה 7:**

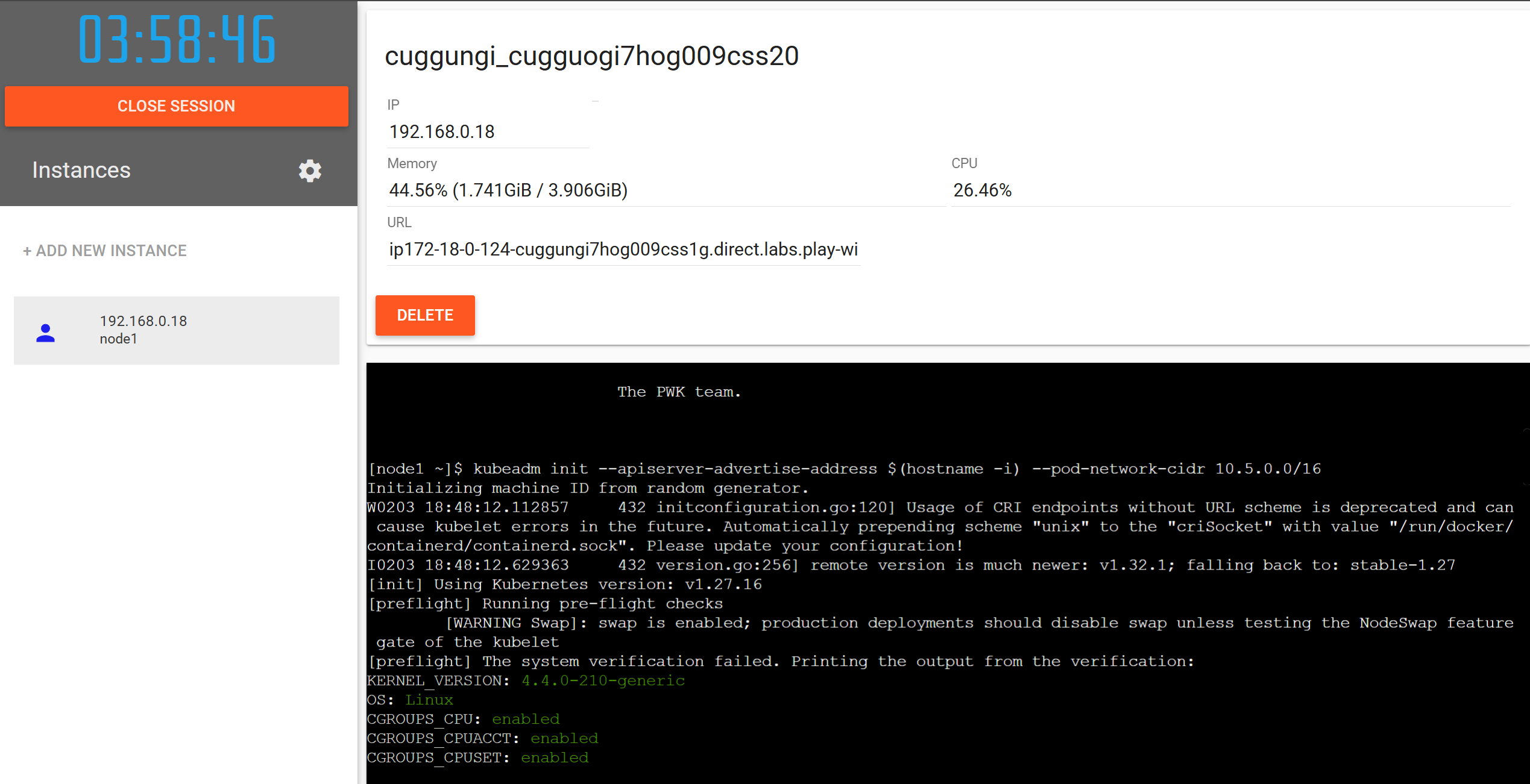
**שקף 31:**

ניצור את המאסטר נוד באמצעות הפקודה:

kubeadm init --apiserver-advertise-address $(hostname -i) --pod-network-cidr 10.5.0.0/16

configuration:

kubectl apply -f https://raw.githubusercontent.com/cloudnativelabs/kube-router/master/daemonset/kubeadm-kuberouter.yaml



תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

נצרף worker node :

kubeadm join 192.168.0.18:6443 --token 4m2kwf.xjxj687lx6dzvp2q \

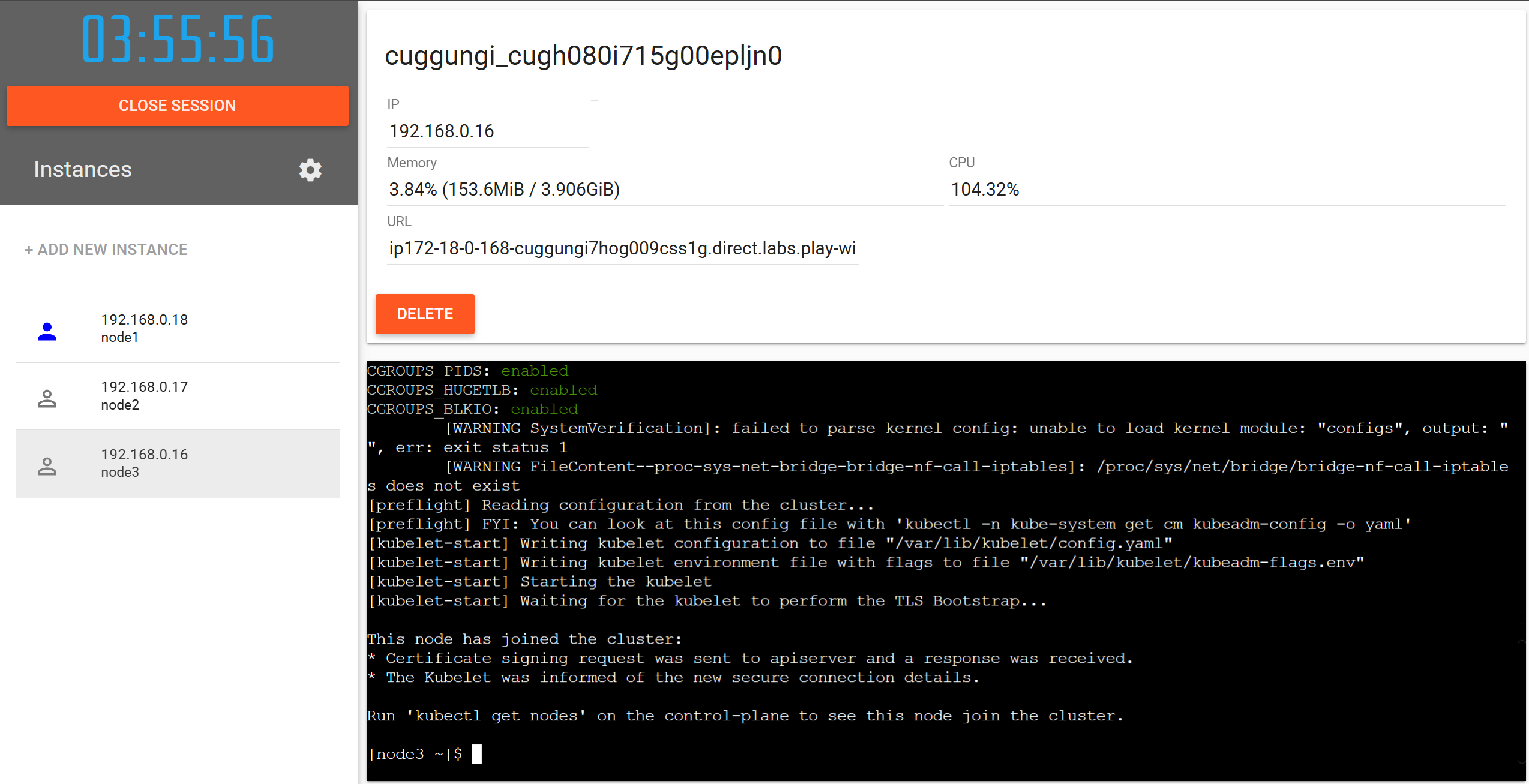
--discovery-token-ca-cert-hash sha256:66228517792379e1af37dce716d568a40f0db0e600ec5e133bd90defc7ded485

פקודה זו ניתן למצוא בפלט של יצירת המאסטר נוד

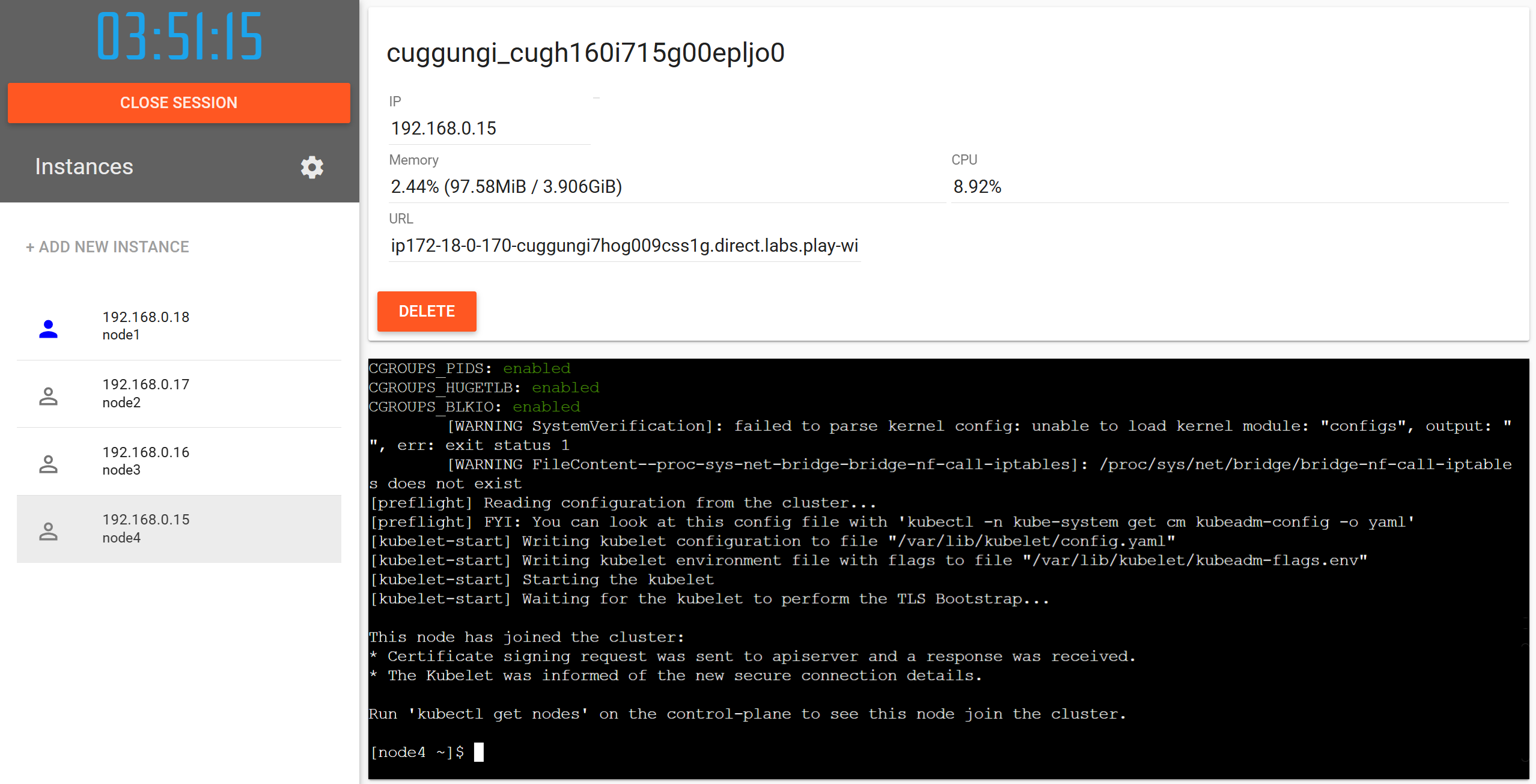
תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

נוסיף worker node שני



נוסיף worker node שלישי



נבדוק במאסטר נוד בעזרת:

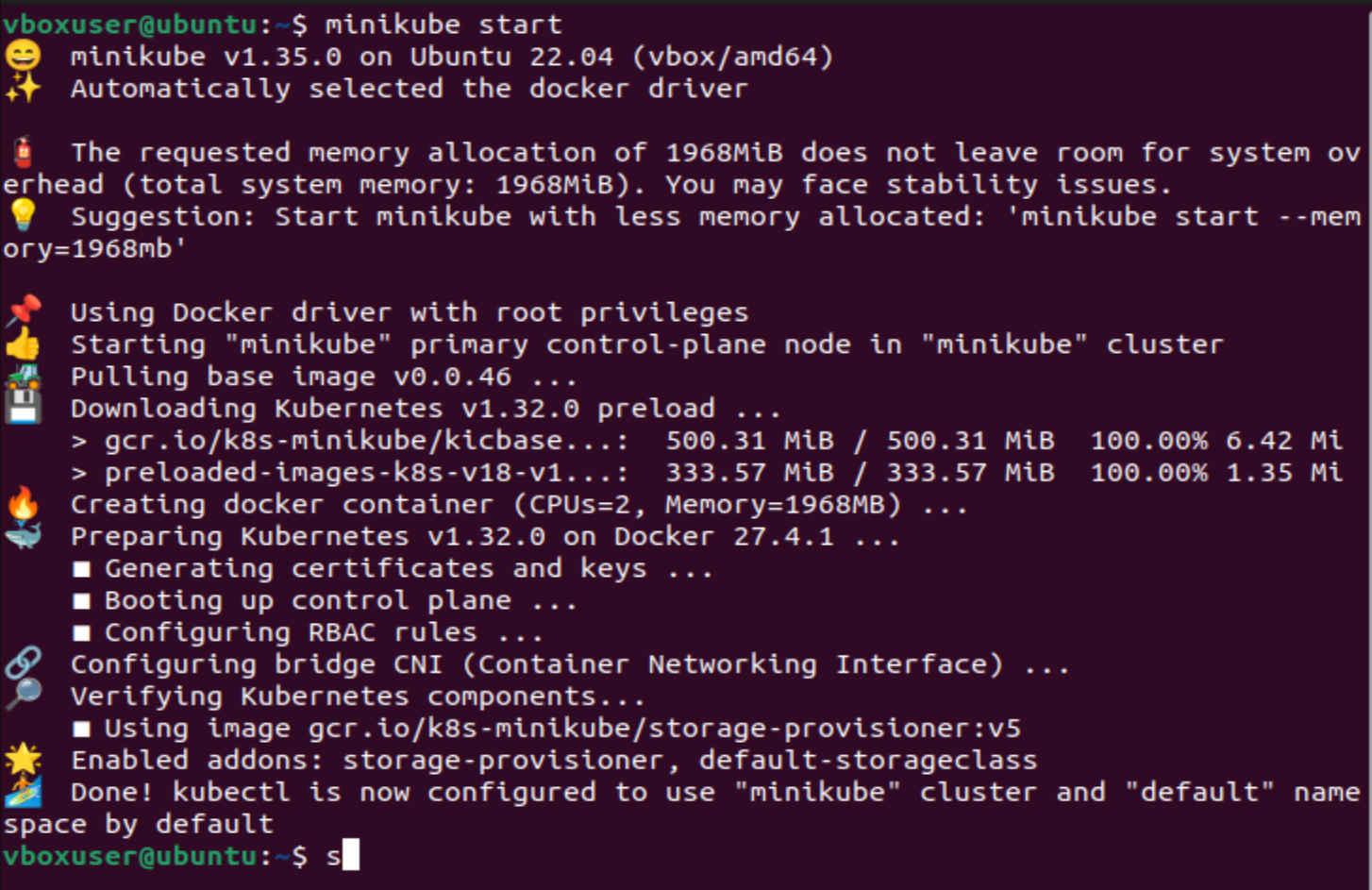
תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

**שקף 50:**

נרצה לבצע בדיקות POST וGET בPOSTMAN לכתובת <http://localhost:7080/>

נריץ minikube



נצור פריסת Kubernetes עם תמונת השרת

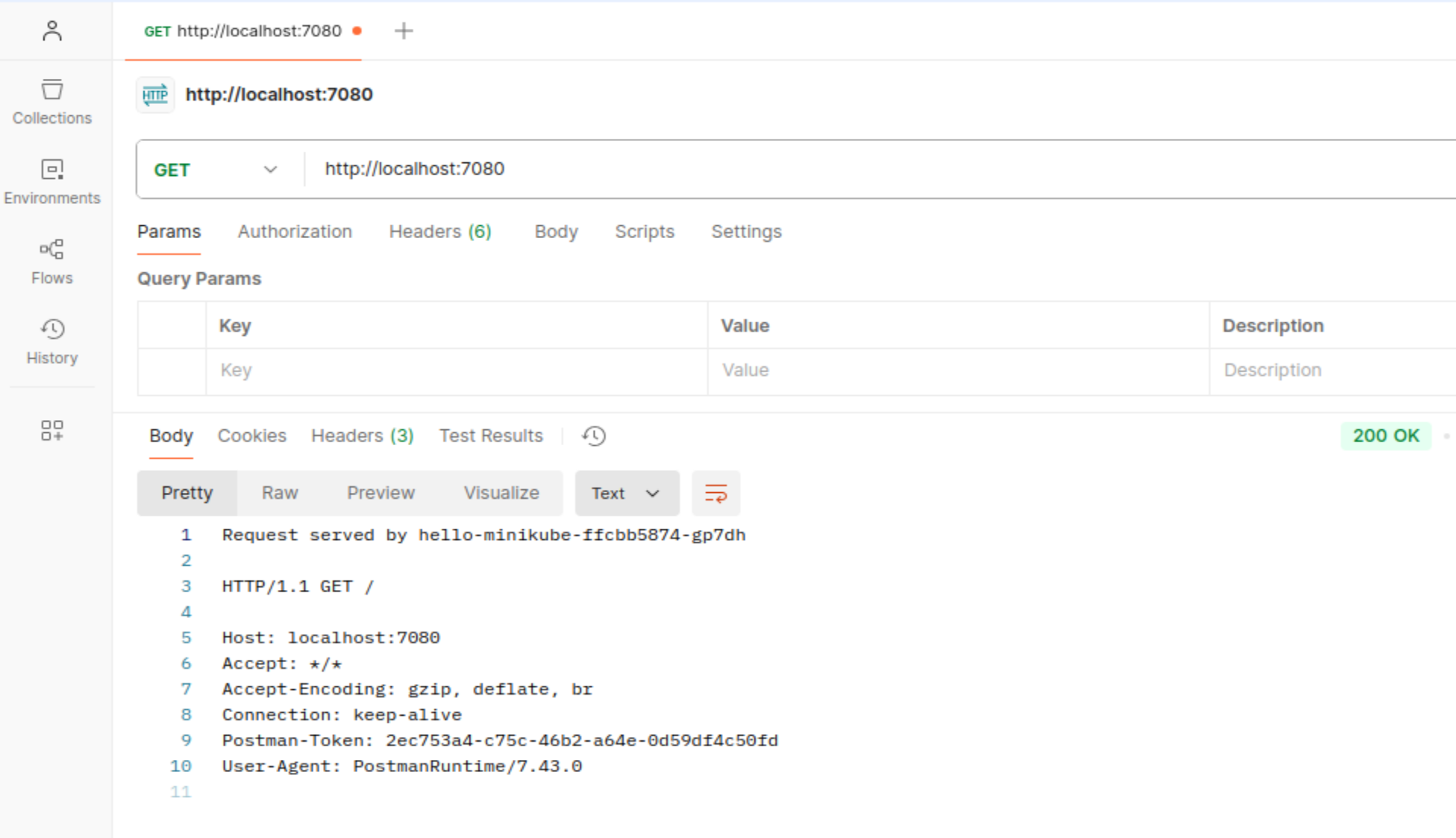
נחשוף את השירות עם NodePort בפורט 8080

ונבדוק פרטי השירות:

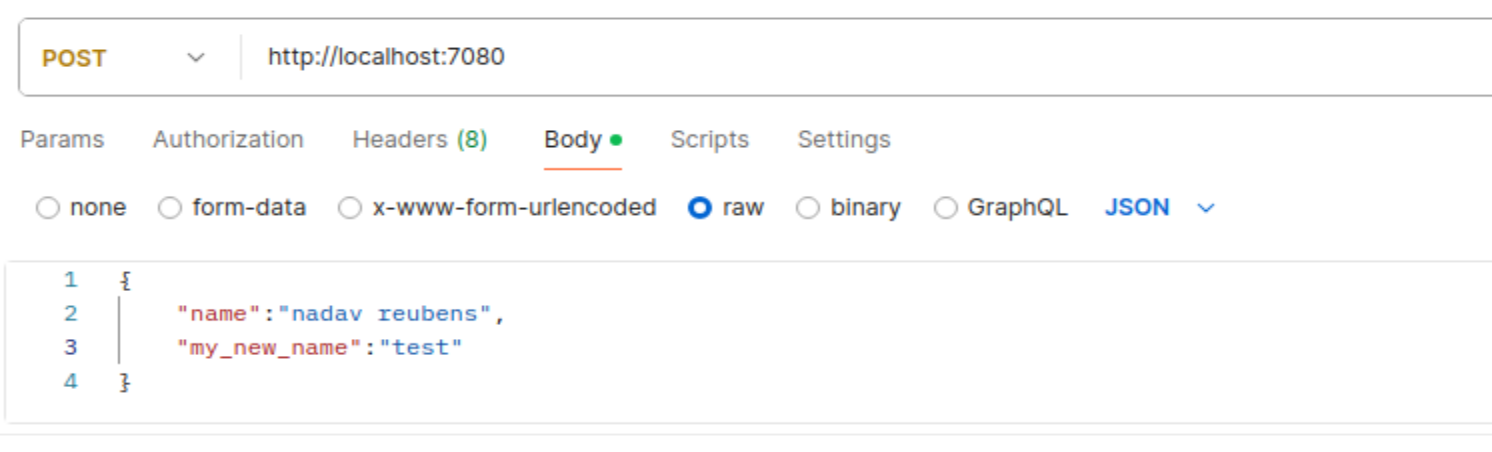
תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

נבצע קריאת GET:



נבצע קריאת POST:



תמונה שמכילה טקסט, גופן, צילום מסך, אלגברה

התיאור נוצר באופן אוטומטי

**שקף 51:**

פקודות לניהול קלאסטר



תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי