

פרויקט גמר- DevConnect

ע"י : יהודית הורוביץ

מייל: yehudit.horowitz@grunitech.com

תאריך: 07-09/11/2023

פלטפורמה: GCP

תוכן העניינים:

חלק א':

- בניית קבצי: init.sh, delete.sh
- בניית התמונה והרצתה:
- רישום לאתר
- בניין התמונה מחדש עם ווליום
- כניסה לאתר
- יצירת רפוזיטורי ודחיפת התמונה לגיטהאב

חלק ב':

- יצירת artifact עם שם המשתמש:
- יצירת cluster עפ"י הגדרות נדרשות:
- בניית קובץ deploy.sh
- דחיפת התמונה ל-artifact ע"י הרצת הקובץ deploy.sh
- יצירת namespace מוגן production
- דחיפה של הגרסה המתוקנת:
- פריסה מחדש:

חלק ג':

- הרמת מכונה וירטואלית עפ"י ההגדרות הנדרשות:
- פתיחת תיקיה חדשה בשם jenkins_lab המכילה Dockerfile:
- קובץ deploy.sh:
- התחברות למכונה שלי ומשיכת התמונה:
- יצירת ווליום והרצת התמונה בווליום:
- יצירת מפתח SSH והוספה בגיטהאב:
- יצירת PIPELINE והגדרתו:
- כתיבת הקוד pipeline: (העתקתי לvs code בשביל להעתיק בצורה מסודרת לכאן)
- הרצה סופית (build now):
- הארכיטקטורה של הפרויקט:
- התקלות שנתקלתי בהם:

רקע כללי:

מה זה DJANGO:

[/https://www.djangoproject.com](https://www.djangoproject.com)

פרויקט גמר- DevConnect

https://www.w3schools.com/django/django_intro.php

https://www.youtube.com/watch?v=t_p4ZyAYyaY

מה זה SQLITE:

<https://www.sqlite.org/index.html>

<https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite>

<https://www.sqlite.org/about.html>

הסבר כללי על הפרויקט:

בפרויקט זה בעצם עשיתי פריסה לאפליקציה DevConnect דרך תהליך פיתוח עד לשלב ההגשה ללקוח. בשלב הראשון: ביצעתי דוקריזציה לאפליקציה ואז בניתי תמונה וקונטיינר שירוך על כל מכונה שהיא ללא צורך בהתקנת תלויות, ואז יצרתי אוטומציה לאתחול האפליקציה ומחיקת רכיבי דוקר מיותרים, ובסוף יצרתי רפוזיטורי והעלתי את הקוד לגיטהאב בשלב השני ביצעתי deploy- ופריסה של האפליקציה על גבי הענן. העלתי את התמונה ל- repository artifact, חשפתי אותה ב-deployment על גבי cluster GKE ואז בעצם התחיל מחזור חיים של פיתוח אפליקציה - העלאה מחדש ו-roll out בשלב האחרון יצרתי pipeline של CI/CD בעזרת jenkins שמכיל כמה שלבים, משיכת התמונה והרצתה ובסוף לאחר שהכל הצליח דחיפת התמונה מחדש לארטיפקט ופריסת הפרוייקט מחדש.

פתרון הפרויקט שלב אחר שלב

חלק א':

בשלב הזה הדרישות היו לעשות לאתר דוקריזציה, לבנות ולהריץ את התמונה, בצורה כזאת שיתאפשר להירשם לאתר ואז למחוק את הקונטיינר ועדיין לעשות login לאתר מבלי להירשם מחדש (ע"י שימוש בווליום) ובסופו של דבר ליצור רפוזיטורי בגיטהאב ולדחוף לשם את התמונה.

דוקריזציה: requirements, dockerfile

Dockerfile:

```
# Use an official Python runtime as a parent image
FROM python:3.9

# Set environment variables for Python and Django
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

# Set the working directory inside the container
WORKDIR /app

# Copy the requirements file into the container at /app
COPY requirements.txt /app/
```

פרויקט גמר- DevConnect

```
# Install any needed packages specified in requirements.txt
RUN pip install -r requirements.txt

# Install django-staticfiles
RUN pip install django-staticfiles

# Copy the current directory contents into the container at /app
COPY . /app/

# Make port 8000 available to the world outside this container
EXPOSE 8000

# Run Django migrations and start the development server
CMD python manage.py makemigrations && \
    python manage.py migrate && \
    python manage.py runserver 0.0.0.0:8000
```

בקובץ זה פירטתי את החבילות הנדרשות לפרויקט
(ואח"כ מריצים אותם בדוקר פייל בפקודה: `RUN pip install -r requirements.txt`)

Requirement.txt:

```
django_web_app > requirements.txt
1 Django==4.2.7
2 django-crispy-forms==2.1.0
3 Pillow==10.1.0
4
```

● בניית קבצי: `init.sh`, `delete.sh`

יצרתי קובץ `init.sh` שבונה את התמונה ומריץ אותה:

Init.sh:

```
settings.py  delete.sh  init.sh  requirements.txt
django_web_app > $ init.sh
1 #!/bin/bash
2
3 # Build the image using the Dockerfile
4 docker build -t final_project .
5
6 # Run the container with volumes
7 docker run -d -p 8000:8000 -v my-django-app-data:/code/media -v my-django-db-data:/var/lib/postgresql/data final_project
```

פרויקט גמר- DevConnect

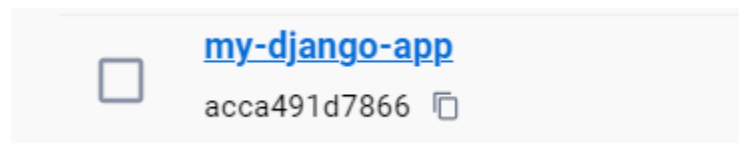
יצרתי קובץ delete.sh שמוחק את התמונה מהדוקר:

:Delete.sh

```
settings.py delete.sh requirements.txt
django_web_app > $ delete.sh
1  #!/bin/bash
2
3  docker rm $(docker ps -a -q)
```

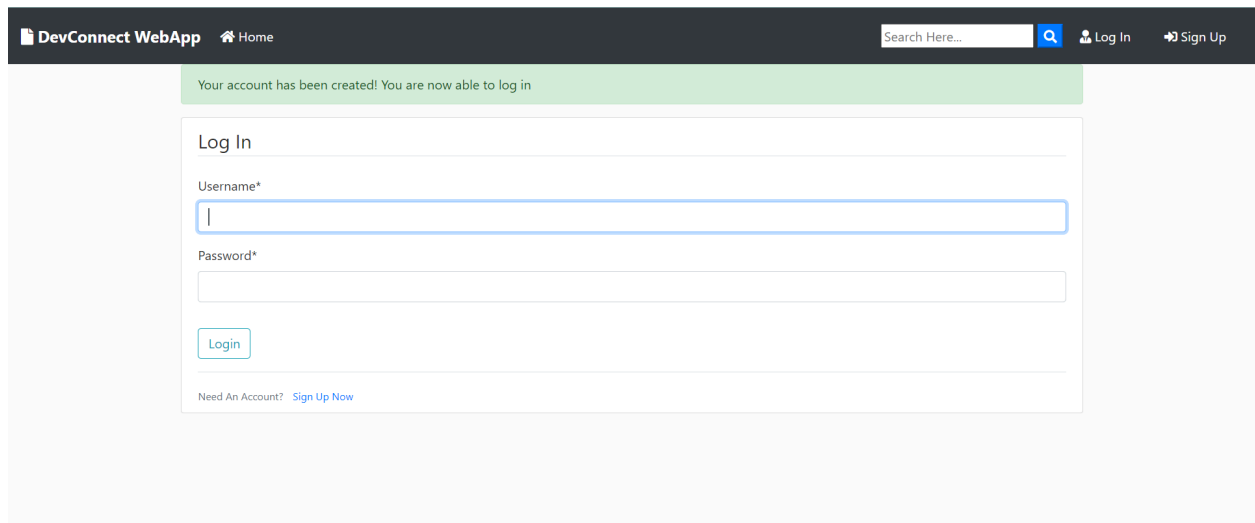
• בניית התמונה והרצתה:

הרצתי את הקובץ init.sh שבונה ומריץ את התמונה



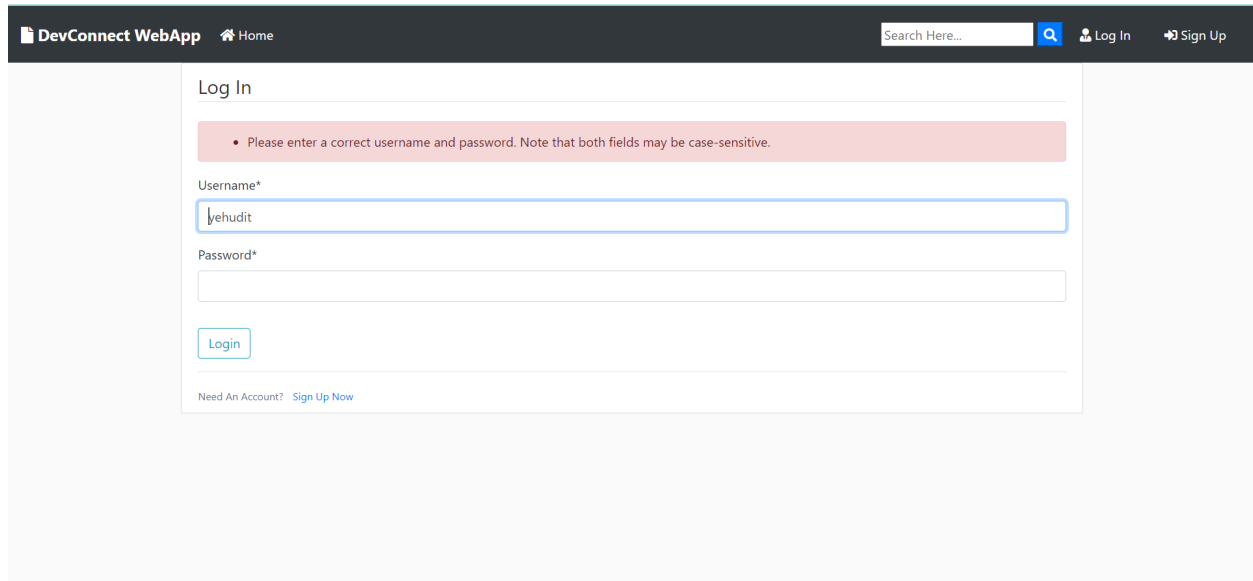
• רישום לאתר

כעת ביצעתי sign up לאתר:



לאחר מחיקת הקונטיינר ניסיתי להיכנס שוב מחדש עם השם משתמש שלי וזה לא נתן לי כי עשיתי עם ווליום.

פרויקט גמר- DevConnect



בניין התמונה מחדש עם ווליום

כדי להפוך את התמונה שלי ל"ציבה" בניתי מחדש את התמונה בווליום והרצתי את הקונטיינר:

```
yehud@DESKTOP-J0SE1KJ MINGW64 ~/Downloads/DevConnect Django CICD project/DevConnect Django CICD project/Django-WebApp (main)
$ docker volume create my-django-volume
my-django-volume

0f1e15cfed255c3300d0a34fa6fc989bedb0ff4da99ce10c0cf01c0d5e8e33d9
PS C:\Users\yehud\Downloads\DevConnect Django CICD project\DevConnect Django CICD project\Django-WebApp\django_web_app> docker run -d -p 8000:8000
0 -v my-django-volume:/app my-django-app
91e02a92623e25e63ce1c560c77fe242985f228e824240e4229f033f2ee6e83a
```

למה לבנות בווליום:

בהקשר של Docker, volume הוא מיקום אחסון מתמשך שקיים מחוץ לקונטיינר. Volumes שימושיים לאחסון נתונים שצריכים להימשך גם אם הקונטיינר נעצר או הוסר. ב-Dockerfile, ההוראה VOLUME משמשת לציון נקודת הרכבה עבור volume בתוך הקונטיינר.

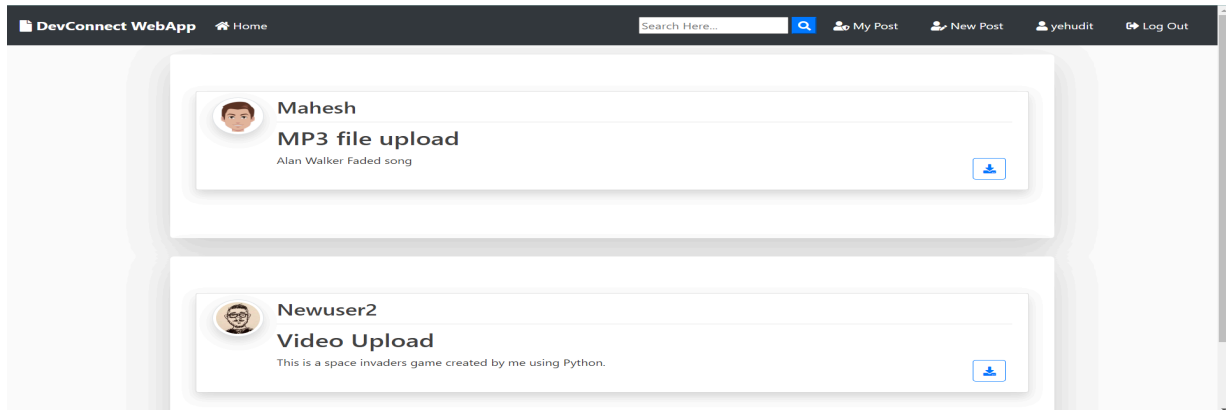
מקור:

<https://www.tutorialspoint.com/what-is-the-purpose-of-volume-in-dockerfile#:~:text=In%20the%20context%20of%20Docker,a%20volume%20within%20the%20container>

פרויקט גמר- DevConnect

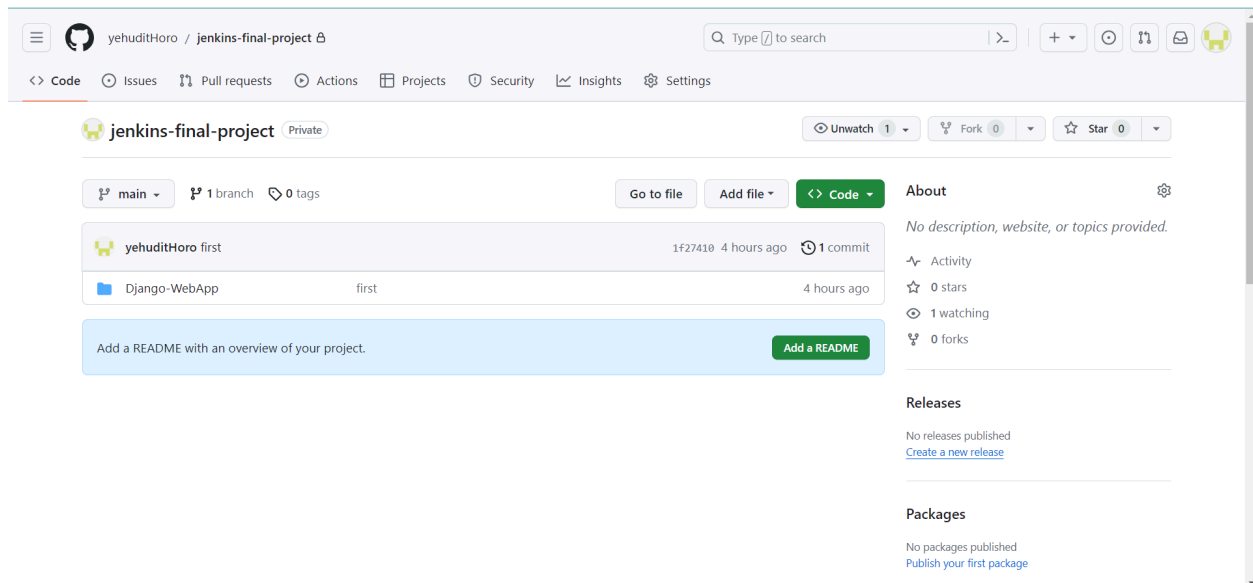
• כניסה לאתר

ואז ניסיתי לעשות log in עם השם משתמש שיצרתי מקודם (לפני המחיקה), והצלחתי:



• יצירת רפוזיטורי ודחיפת התמונה לגיטהאב

יצרתי רפוזיטורי פרטי בגיטהאב ודחפתי לשם את הפרויקט:

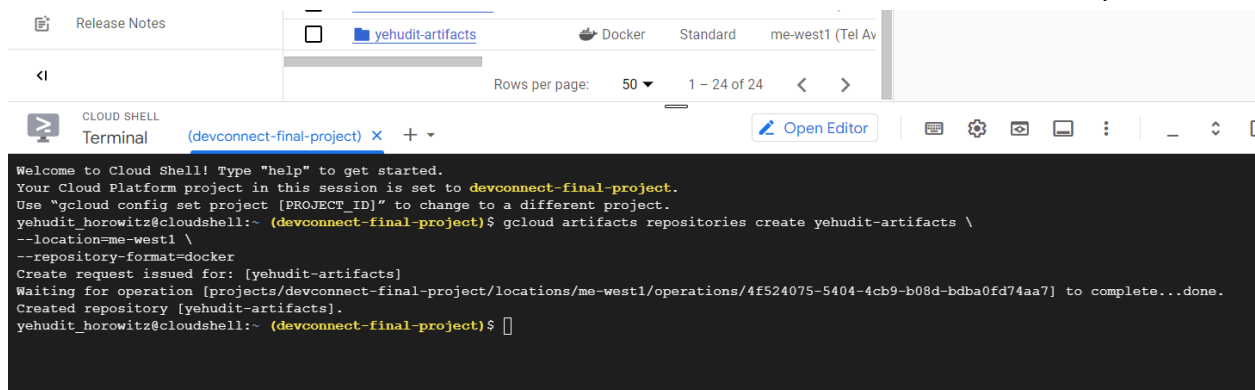


פרויקט גמר- DevConnect

חלק ב':

- יצירת artifact עם שם המשתמש:

יצרתי ארטיפקט עם השם שלי:



```
Release Notes
yehudit-artifacts
Docker Standard me-west1 (Tel Av)
Rows per page: 50 1 - 24 of 24
CLOUD SHELL
Terminal (devconnect-final-project) x + Open Editor
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to devconnect-final-project.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
yehudit_horowitz@cloudshell:~ (devconnect-final-project)$ gcloud artifacts repositories create yehudit-artifacts \
--location=me-west1 \
--repository-format=docker
Create request issued for: [yehudit-artifacts]
Waiting for operation [projects/devconnect-final-project/locations/me-west1/operations/4f524075-5404-4cb9-b08d-bdba0fd74aa7] to complete...done.
Created repository [yehudit-artifacts].
yehudit_horowitz@cloudshell:~ (devconnect-final-project)$
```

- יצירת cluster עפ"י הגדרות נדרשות:

ואז יצרתי קלסטר לפי ההגדרות:

1. Go to the Kubernetes Engine Console
2. Click the Create cluster button
3. In the Cluster name field, enter `yehudit-cluster`
4. In the Location section, select Zonal
5. In the Zone section, select `me-west1-b`
6. In the Machine type section, select `e2-micro`
7. In the Node count section, enter `1`
8. In the Service account section, select `DevOps-sa`
9. In the Boot disk size section, enter `10GB`
10. In the Node taints section, click the Add node taint button
11. In the Key field, enter `NO_EXECUTE`
12. In the Value field, enter `webapp:mywebapp`
13. Click the Add button
14. Review the cluster configuration and click the Create button

| | | | | | | | |
|--------------------------|-------------------------------------|---------------------------------|------------|---|---|------|---|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | yehudit-cluster | me-west1-a | 1 | 2 | 1 GB | — |
|--------------------------|-------------------------------------|---------------------------------|------------|---|---|------|---|

פרויקט גמר- DevConnect

מה זה taints:

Node taints משמשים לסימון nodes עם תכונות או דרישות ספציפיות. Taints משמשים למניעת הפעלת pods על nodes שאינם עומדים בקריטריונים מסוימים.

"NO_EXECUTE" מציין שלא ניתן לתזמן pods ללא סבילות מתאימה בצמתים עם taint כזה. במקרה שלנו, ה-taint הוא "webapp" והווליום הוא "mywebapp". המשמעות היא שרק pods עם סבילות תואמת ל-taint עם מפתח "webapp" וערך "mywebapp" יוכלו לפעול על pods עם taint זה.

על ידי החלת taints על nodes, אני בעצם יכולה להבטיח שרק nodes ספציפיים, עם התנאים המתאימים מתוזמנים על אותם nodes. זה עוזר בהפרדת עומסי עבודה ובהבטחה שיישומים קריטיים או מיוחדים יפעלו על nodes ייעודיים.

מקור:

<https://blog.kubecost.com/blog/kubernetes-taints/#:~:text=Taints%20are%20a%20Kubernetes%20node,benefits%20of%20taints%20and%20tolerations.>

<https://cloud.google.com/kubernetes-engine/docs/how-to/node-taints>

• בניית קובץ deploy.sh

הוספתי קובץ deploy.sh:

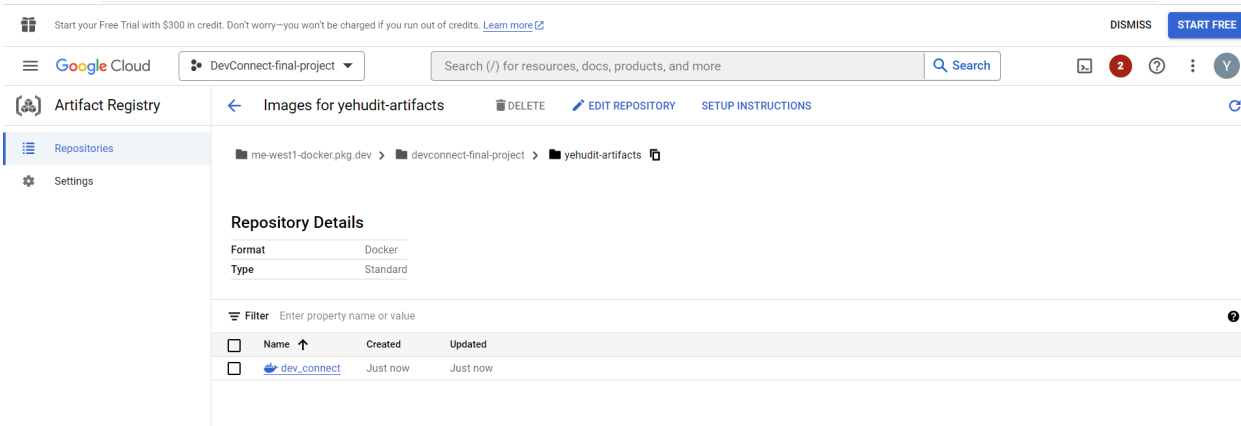
```
Django-WebApp > django_web_app > $ deploy.sh
1  #!/bin/bash
2
3  # Prompt the user to enter the version
4  echo "Which version of the application would you like to push?"
5  read -p "Enter the version: " VERSION
6
7  # Set the image name and tag
8  IMAGE_NAME="django"
9  IMAGE_TAG=${IMAGE_NAME}:${VERSION}
10
11 # Set the artifact repository URL
12 ARTIFACT_REGISTRY="me-west1-docker.pkg.dev/devconnect-final-project/yehudit-artifacts/dev_connect"
13
14 # Build the Docker image
15 docker build -t ${IMAGE_TAG} .
16
17 # Authenticate with Google Cloud (already configured with user account)
18 docker tag ${IMAGE_TAG} ${ARTIFACT_REGISTRY}:${VERSION}
19 docker push ${ARTIFACT_REGISTRY}:${VERSION}
20
21 # Clean up: remove the local Docker image (optional)
22 docker image rm ${ARTIFACT_REGISTRY}:${VERSION}
23
```

הסבר על הקובץ:

הקובץ מקבל מהמשתמש את מספר הגרסה, בונה את התמונה ודוחף אותה לארטיפקט.

פרויקט גמר- DevConnect

- דחיפת התמונה ל-artifact ע"י הרצת הקובץ deploy.sh



- יצירת namespace מסוג production

עכשיו יצרתי namespace מסוג production בעזרת הקבצים/ הפקודות הבאות:

```
kubectl create namespace production
```

עכשיו פרסתי את האפליקציה שלי בעזרת הקבצים הבאים:

nano app-service.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: devconnect-app-deployment-yehudit
spec:
  replicas: 1
  selector:
    matchLabels:
      app: devconnect-app
  template:
    metadata:
      labels:
        app: devconnect-app
    spec:
      containers:
        - name: devconnect-app
```

פרויקט גמר- DevConnect

```
image:
me-west1-docker.pkg.dev/devconnect-final-project/yehudit-artifacts/dev_connect:0.0.1
ports:
- containerPort: 8000
tolerations:
- key: "webapp"
  operator: "Equal"
  value: "mywebapp"
  effect: "NoExecute"
```

הסבר על הקובץ:

בעזרת הפקודות האלה בעצם הרמתי apiVersion מסוג: Deployment עם רפליקה (העתק) אחת... משכתי את התמונה: dev_connect:0.0.1 ונתתי הרשאה רק לפודים עם התווית: devconnect-app והרצתי לפורט 8000 והגדרתי tolerations שרק פודים שמכילים את התנאים המתאימים יוכלו לרוץ (בהתאם ל-taints שהגדרתי בקלסטר)

מקור:

<https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>

<https://www.armosec.io/blog/kubernetes-deployment-and-service/>

nano service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: app-service
  namespace: production
spec:
  selector:
    app: devconnect-app
  ports:
    - protocol: TCP
      port: 8000
      targetPort: 8000
  type: LoadBalancer
```

הסבר על הקובץ:

פקודות אלו מגדירות שירות Kubernetes בשם app-service שחושף יציאה 8000 בכתובת ה-IP החיצונית של הקלסטר. השירות מעביר תעבורה ל-pods שיש להם את אפליקציית התווית: devconnect-app. במקרה זה, ה-service מעביר תעבורה ל-pods שיש להם את אפליקציית התווית: devconnect-app. הפודים מאזינים בפורט 8000. לכן, כאשר לקוח שולח בקשה לכתובת ה-IP של השירות וליציאה 8000, ה-load balancer יפיץ את הבקשה לאחד הפודים שמאזינים בפורט 8000.

מקור:

<https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/>

פרויקט גמר- DevConnect

```
yehudit_horowitz@cloudshell:~ (devconnect-final-project) $ kubectl apply -f app-service.yaml
deployment.apps/app-yehudit created
yehudit_horowitz@cloudshell:~ (devconnect-final-project) $ nano service.yaml
yehudit_horowitz@cloudshell:~ (devconnect-final-project) $ kubectl apply -f service.yaml
service/app-service created
```

והם אכן נוצרו:

| | | | | | | |
|--------------------------|---|----|------------------------|-----------------------------------|------------|--|
| <input type="checkbox"/> | devconnect-app-deployment-yehudit | OK | Deployment | 1/1 | production | yehudit-cluster |
| <input type="checkbox"/> | app-service | OK | External load balancer | 34.83.53.223:8000 | 1/1 | production yehudit-cluster |

כעת נתקלתי בשיגאה, הכתובת IP לא מורשית עפ"י ההגדרות בקובץ setting.py:

DisallowedHost at /

Invalid HTTP_HOST header: '34.83.53.223:8000'. You may need to add '34.83.53.223' to ALLOWED_HOSTS.

Request Method: GET
Request URL: http://34.83.53.223:8000/
Django Version: 3.2.5
Exception Type: DisallowedHost
Exception Value: Invalid HTTP_HOST header: '34.83.53.223:8000'. You may need to add '34.83.53.223' to ALLOWED_HOSTS.
Exception Location: /usr/local/lib/python3.9/site-packages/django/http/request.py, line 149, in get_host
Python Executable: /usr/local/bin/python
Python Version: 3.9.18
Python Path: ['/app', '/usr/local/lib/python3.9.zip', '/usr/local/lib/python3.9', '/usr/local/lib/python3.9/lib-dynload', '/usr/local/lib/python3.9/site-packages']
Server time: Wed, 08 Nov 2023 12:01:25 +0000

Traceback [Switch to copy-and-paste view](#)

/usr/local/lib/python3.9/site-packages/django/core/handlers/exception.py, line 47, in inner

47. response = get_response(request)

► Local vars

/usr/local/lib/python3.9/site-packages/django/utils/deprecation.py, line 116, in __call__

116. response = self.process_request(request)

► Local vars

/usr/local/lib/python3.9/site-packages/django/middleware/common.py, line 48, in process_request

48. host = request.get_host()

► Local vars

/usr/local/lib/python3.9/site-packages/django/http/request.py, line 149, in get_host

149. raise DisallowedHost(msg)

► Local vars

ולכן הוספתי בקובץ Setting.py:

```
8 ALLOWED_HOSTS = ['*']
9
```

מקור:

[django - Invalid HTTP_HOST header: '127.0.0.1:8000'. You may need to add '127.0.0.1' to ALLOWED_HOSTS - Stack Overflow](#)

פרויקט גמר- DevConnect

דחיפה של הגירסה המתוקנת:

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

DISMISS [START FREE](#)

Google Cloud

DevConnect-final-project

arti

Search

Artifact Registry

Digests for dev_connect

DELETE

SETUP INSTRUCTIONS

Repositories

Settings

me-west1-docker.pkg.dev > devconnect-final-project > yehudit-artifacts > dev_connect

Filter Enter property name or value

| Name | Description | Tags | Created | Updated | |
|---------------|-------------|--------|-------------|-------------|--|
| 81dc585dc0d06 | | v0.0.2 | Just now | Just now | |
| d4211813b119 | | 0.0.1 | 3 hours ago | 3 hours ago | |

Release Notes

פריסה מחדש:

עכשיו זה עובד!!

DevConnect WebApp

Home

Search Here...

Log In

Sign Up

Mahesh

MP3 file upload

Alan Walker Faded song

Download

Newuser2

Video Upload

This is a space invaders game created by me using Python.

Download

פרויקט גמר- DevConnect

חלק ג':

- הרמת מכונה וירטואלית עפ"י ההגדרות הנדרשות:

Create a new instance:

- On the Compute Engine page, click Create Instance.
 - In the Name field, enter yehudit-jenkins.
 - In the Region field, select me-west1 (Tel-Aviv).
 - In the Machine type field, select e2-medium (2 vCPU, 1 core, 4 GB memory).
- e. In the Boot disk section, select Standard persistent disk and then click Change.
 - In the Size (GB) field, enter 10.
 - In the Filesystem field, select ext4.
 - Click OK to save your changes.
 - f. In the Boot disk section, click Create.

Assign a service account:

- In the Service accounts section, select DevOps-sa.
- c. Click Done.

Enable Docker:

- In the Management, Security, and Disk section, click Edit.
 - Expand the Startup scripts section.
- c. Paste the following script into the Startup script field:
 - Bash

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo systemctl start docker
sudo systemctl enable docker
```

| | | | | | | | |
|--------------------------|-------------------------------------|-----------------|------------|-----------------------|-----------------------|-------|---|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | yehudit-jenkins | me-west1-a | 10.208.0.37 (nic0) | 34.165.82.2 (nic0) | SSH ▾ | ⋮ |
|--------------------------|-------------------------------------|-----------------|------------|-----------------------|-----------------------|-------|---|

פרויקט גמר- DevConnect

• פתיחת תיקיה חדשה בשם jenkins_lab המכילה Dockerfile:

```
Dockerfile > ...
1 FROM jenkins/jenkins:lts-jdk17
2 USER root
3
4 # Give jenkins access to docker
5 RUN groupadd -g 997 docker
6 RUN gpasswd -a jenkins docker
7
8 # Install docker
9 RUN apt-get update && apt-get install -y docker.io
10
11 # Install docker-compose
12 RUN curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-
13 RUN chmod +x /usr/local/bin/docker-compose
14
15 # Install Docker CLI
16 RUN apt-get update && \
17     apt-get install -y apt-transport-https ca-certificates curl software-properties-common && \
18     curl -fsSL https://get.docker.com | sh && \
19     apt-get clean
20
21 # Install Google Cloud SDK
22 # RUN apt-get install -y google-cloud-sdk
23 RUN apt-get update && apt-get install -y curl gnupg
24 RUN echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] http://packages.cloud.google.com/apt cloud-sdk main" | tee -a /etc/apt/sou
25 RUN curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key --keyring /usr/share/keyrings/cloud.google.gpg add -
26 RUN apt-get update && apt-get install -y google-cloud-sdk
27
28
29 # Make sure the 'jenkins' user can run Docker commands
30 RUN usermod -aG docker jenkins
31
32 EXPOSE 8080
33
34 USER jenkins
```

פרויקט גמר- DevConnect

• קובץ deploy.sh :

```
$ deploy.sh
1  #!/bin/bash
2  # Prompt the user to enter the version
3  echo "Which version of the application would you like to push?"
4  read -p "Enter the version: " VERSION
5  # Set the image name and tag
6  IMAGE_NAME="with_jenkins"
7  IMAGE_TAG=${IMAGE_NAME}:${VERSION}
8  VM_INSTANCE="yehudit-jenkins"
9  # GCP Artifact Registry details
10 PROJECT_ID="devconnect-final-project"
11 GCP_REGION="me-west1"
12 # Set the artifact repository URL
13 ARTIFACT_REGISTRY="me-west1-docker.pkg.dev/devconnect-final-project/yehudit-artifacts/with_jenkins"
14 # Authenticate with GCP using the gcloud CLI
15 gcloud config configurations activate final-project
16 gcloud auth login --no-launch-browser
17 gcloud config set project $PROJECT_ID
18 gcloud auth configure-docker me-west1-docker.pkg.dev
19 # Configure the gcloud CLI for your project
20 gcloud config set project "$PROJECT_ID"
21 # Configure the gcloud CLI for the Artifact Registry repository
22 gcloud artifacts repositories configure-docker "yehudit-artifacts" \--location="$GCP_REGION"
23 # Build the Docker image
24 docker build -t ${ARTIFACT_REGISTRY} .
25 # Authenticate with GCP using the docker credential helper
26 gcloud auth print-access-token | docker login -u oauth2accesstoken --password-stdin "https://me-west1-docker.pkg.dev"
27 docker tag ${IMAGE_NAME} ${ARTIFACT_REGISTRY}:${VERSION}
28 # Push the Docker image to the GCP Artifact Registry repository
29 docker push ${ARTIFACT_REGISTRY}:${VERSION}
30 gcloud compute ssh "$VM_INSTANCE" --zone="me-west1-a" --command="docker
31 run -d -p 8080:8080 $ARTIFACT_REGISTRY/.$VERSION"
32 # Clean up: Remove the local image (optional)
33 docker rmi ${ARTIFACT_REGISTRY}:${VERSION}
34 # Log out (optional)
35 docker logout
36 # Revoke the gcloud CLI authorization (optional)
37 gcloud auth revoke
```

הרצתי את הקובץ וזה עבד:



[←](#) Images for yehudit-artifacts [DELETE](#) [EDIT REPOSITORY](#) [SETUP INSTRUCTIONS](#)

me-west1-docker.pkg.dev > devconnect-project > yehudit-artifacts

Repository Details

| | |
|--------|----------|
| Format | Docker |
| Type | Standard |

Filter Enter property name or value

| <input type="checkbox"/> | Name ↑ | Created | Updated |
|--------------------------|--|--------------|--------------|
| <input type="checkbox"/> |  dev_connect | 1 hour ago | 1 hour ago |
| <input type="checkbox"/> |  with_jenkins | 1 minute ago | 1 minute ago |

פרויקט גמר- DevConnect

• התחברות למכונה שלי ומשיכת התמונה:



SSH-in-browser

↑ UPLOAD FILE

↓ DOWNLOAD FILE



```
Linux yehudit-jenkins 5.10.0-26-cloud-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
yehudit_horowitz@yehudit-jenkins:~$ gcloud auth configure-docker me-west1-docker.pkg.dev
```

```
Adding credentials for: me-west1-docker.pkg.dev
```

```
After update, the following will be written to your Docker config file located at
```

```
[/home/yehudit_horowitz/.docker/config.json]:
```

```
{  
  "credHelpers": {  
    "me-west1-docker.pkg.dev": "gcloud"  
  }  
}
```

```
Do you want to continue (Y/n)? y
```

```
Docker configuration file updated.
```

```
yehudit_horowitz@yehudit-jenkins:~$ docker pull me-west1-docker.pkg.dev/devconnect-project/yehudit-artifacts/wi  
th_jenkins:v0.0.1
```

```
v0.0.1: Pulling from devconnect-project/yehudit-artifacts/with_jenkins
```

```
0a9573503463: Pull complete
```

```
e7dfbe47a424: Pull complete
```

```
63a7b790559d: Pull complete
```

```
de38613af57f: Pull complete
```

```
5e2a9b06c446: Pull complete
```

```
aaaca84f7bc7: Pull complete
```

```
b7d3a74d9ac1: Pull complete
```

```
c1655dbb8674: Pull complete
```

```
8517c106c345: Pull complete
```

```
6c94ede8b808: Pull complete
```

```
f94c7e55f771: Pull complete
```

```
8f558801c051: Pull complete
```

```
d86c8359b81a: Pull complete
```

• יצירת ווליום והרצת התמונה בווליום:

```
me-west1-docker.pkg.dev/devconnect-project/yehudit-artifacts/with_jenkins:v0.0.1
```

```
yehudit_horowitz@yehudit-jenkins:~$ docker volume create jenkins_vol
```

```
jenkins_vol
```

```
yehudit_horowitz@yehudit-jenkins:~$ docker run -it -p 8080:8080 -p 50000:50000 -v /var/run/docker.sock:/var/run  
/docker.sock -v jenkins_home:/var/jenkins_home me-west1-docker.pkg.dev/devconnect-project/yehudit-artifacts/wit  
h_jenkins:v0.0.1
```

```
Running from: /usr/share/jenkins/jenkins.war
```

```
webroot: /var/jenkins_home/war
```

```
2023-11-12 10:16:05.868+0000 [id=1] INFO winstone.Logger#logInternal: Beginning extraction from war file
```

```
2023-11-12 10:16:07.253+0000 [id=1] WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPat
```

```
h
```

```
2023-11-12 10:16:07.380+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: jetty-10.0.17; built:
```

```
2023-10-02T04:04:10.314Z; git: a0f5f05abaa6c3aabb7c3d35f10a6f412ab8b05f; jvm 17.0.8.1+1
```

```
2023-11-12 10:16:07.787+0000 [id=1] INFO o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Suppor
```

```
t for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
```

```
2023-11-12 10:16:07.864+0000 [id=1] INFO o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=n
```

```
ode0
```

```
2023-11-12 10:16:08.808+0000 [id=1] INFO hudson.WebAppMain#contextInitialized: Jenkins home directory: /
```

```
var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
```


פרויקט גמר- DevConnect

```
SSH-in-browser
[Icons] [Keyboard] [Settings]
[Upload File] [Download File]

2023-11-12 10:16:09.100+0000 [id=1] INFO o.e.j.server.AbstractConnector#doStart: Started ServerConnector
@5e01a982{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2023-11-12 10:16:09.129+0000 [id=1] INFO org.eclipse.jetty.server.Server#doStart: Started Server@784b990
c{STARTING}[10.0.17,sto=0] @4279ms
2023-11-12 10:16:09.140+0000 [id=25] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: c
ontrolPort=disabled
2023-11-12 10:16:09.503+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Started initialization
2023-11-12 10:16:09.523+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2023-11-12 10:16:11.095+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2023-11-12 10:16:11.106+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2023-11-12 10:16:11.120+0000 [id=32] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extension
s
2023-11-12 10:16:11.440+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2023-11-12 10:16:11.442+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2023-11-12 10:16:11.442+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2023-11-12 10:16:11.444+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all j
obs updated
2023-11-12 10:16:11.500+0000 [id=46] INFO hudson.util.Retrier#start: Attempt #1 to do the action check up
dates server
2023-11-12 10:16:12.422+0000 [id=33] INFO jenkins.install.SetupWizard#init:

*****
*****
*****

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

e0fce6b93bd64eefad2a3bae8f0304c5

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*****
*****
*****

█
```

כניסה לג'נקינס:

Getting Started


Getting Started

| | | | | |
|----------------|--------------------------|-------------------------------------|------------------------|------------------------|
| ✓ Folders | ✓ OWASP Markup Formatter | ⌚ Build Timeout | ⌚ Credentials Binding | ** Ionicons API |
| ⌚ Timestampmer | ⌚ Workspace Cleanup | ⌚ Ant | ⌚ Gradle | Folders |
| ⌚ Pipeline | ⌚ GitHub Branch Source | ⌚ Pipeline: GitHub Groovy Libraries | ⌚ Pipeline: Stage View | OWASP Markup Formatter |
| ⌚ Git | ⌚ SSH Build Agents | ⌚ Matrix Authorization Strategy | ⌚ PAM Authentication | ** Structs |
| ⌚ LDAP | ⌚ Email Extension | ⌚ Mailer | | |

** - required dependency

Jenkins 2.414.3

פרויקט גמר- DevConnect

 Jenkins

חיפוש (CTRL+K)

Yehudit Horowitz

התנתק

Dashboard

חדש

אנשים

היסטוריית בניה

Manage Jenkins

My Views

תור בניות

אין בניות מחוינות בחור

מצב מרכז הבניות

לא פעיל

לא פעיל

Jenkins! ברוך הבא ל

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project


Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

 Jenkins

חיפוש (CTRL+K)

Yehudit Horowitz


התנתק


Dashboard > All


Enter an item name


my pipeline


» Required field

 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

 **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

 **Branch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

פרויקט גמר- DevConnect

• יצירת מפתח SSH והוספה בגיטהאב:

```
jenkins@29adc73227c9:/$ ssh-keygen -t rsa -b 4096 -C "yehuditush@outlook.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:gswgNiDjJNvYI0RV/OGI20ieCUjRS30m+HjFSsN0xug yehuditush@outlook.com
The key's randomart image is:
+--[RSA 4096]-----+
|X+..X..|
|*X.* & *|
|o..X & .|
|l = E O o|
|+ ..o S |
|o..      |
|o o      |
|l +      |
|         |
+----[SHA256]-----+
jenkins@29adc73227c9:/$ cat /var/jenkins_home/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAQDAmWnnF/brU2JkWRyToDX5ERY7MLGAp7sOLd9cjUwqEBqkvJCVurC7cZJyqS8nBLWVquD9lpH1rp18m/PFdkankUNX1yKHVf1g718V58WnQ+D9FFzAR2ukmbs17Ha1f7J04m++WyOPjzstRL611itNaD
zGuftzP98CCAT2wOwceXcN1CF1KN0GQbksy7W3m2mJ9obwKpwtDf5KedGRi+UAq//zIDua34g5A3Sote88NvFTPcaG8aOPG6Y7ImmUThVwoIXOj1gaD2dPldOttaaA/yx1ldQCzXp4CjDxVewPCy6ubJwO8CKFacXWRaCcs4hwVLTd1mCYQsJ2XGc4
1zOUP75IM3Bgv9HcL1189xfCLQTBGtvd6bir91ay2zf8y6na1238Nvf+3zTig2ahxPu8o8IQRfqpHpydJ0KFOO3M/Z2bgnaH9POV3Gw7cYMPJlcaUvdawqZ0e9jC1MxU9H+Gh9hTAs53XCG3mW1TSj+iaU7RRd8w/ocM2Uc6Z4b7Uy7hUImSZQ3ohG5DE7qP8Od2ej4Jr
y18eK4mC9rBaldC474jmsEU+yp5Dfkl15c8se3SSFu/omjBkRwFkwLSK+Ap1Dth6cXJ2YcuF/U8WfuXRZR8KpQmJih9F9VoMhoY4k3yzL7+9mILNve18ORD8AEM4Jv51Q== yehuditush@outlook.com
jenkins@29adc73227c9:/$
```

Settings

Search

yehuditHoro (yehuditHoro)

Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Add new SSH Key

Title

pipeline_key

Key type

Authentication Key

Key

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDAQDAmWnnF/brU2JkWRyToDX5ERY7MLGAp7sOLd9cjUwqEBqkvJCVurC7cZJyqS8nBLWVquD9lpH1rp18m/P
FdkankUNX1yKHVf1g718V58WnQ+D9FFzAR2ukmbs17Ha1f7J04m++WyOPjzstRL611InaDzGufzP98CCAT2w2OWceXcN1CF1KN0GQbksy7W3m2mJ
9obwKpwtDf5KedGRi+UAq//zIDua34g5A3Sote88NvFTPcaG8aOPG6Y7ImmUThVwoIXOj1gaD2dPldOttaaA/yx1ldQCzXp4CjDxVewPCy6ubJwO8CK
FacXWRaCcs4hwVLTd1mCYQsJ2XGc41zOUP75IM3Bgv9HcL1189xfCLQTBGtvd6bir91ay2zf8y6na1238Nvf+3zTig2ahxPu8o8IQRfqpHpydJ0KFOO3M/
Z2bgnaH9POV3Gw7cYMPJlcaUvdawqZ0e9jC1MxU9H+Gh9hTAs53XCG3mW1TSj+iaU7RRd8w/ocM2Uc6Z4b7Uy7hUImSZQ3ohG5DE7qP8Od2ej4Jr
y18eK4mC9rBaldC474jmsEU+yp5Dfkl15c8se3SSFu/omjBkRwFkwLSK+Ap1Dth6cXJ2YcuF/U8WfuXRZR8KpQmJih9F9VoMhoY4k3yzL7+9mILNve1
SOKDR8AEM4Jv51Q== yehuditush@outlook.com

Add SSH key

pipeline_key

SHA256: gswgNiDjJNvYI0RV/OGI20ieCUjRS30m+HjFSsN0xug

Added on Nov 12, 2023

Never used — Read/write

Delete

Credentials that should be available irrespective of domain specification to requirements matching.

| ID | Name | Kind | Description |
|----------------|---------|-------------------------------|-------------|
| devconnect-app | jenkins | SSH Username with private key | |

פרויקט גמר- DevConnect

• יצירת PIPELINE והגדרתו:

General

Enabled 

תאור

Plain text [תצוגה מקדימה](#)

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url ?

git@github.com:yehuditHoro/jenkins-final-project.git/

מתקדם ▾

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2
3   agent any
4
5   environment {
6
7     DOCKERFILE_PATH = 'Dockerfile' // Path to your Dockerfile
8
9     IMAGE_NAME = 'app'
10
11     IMAGE_TAG = 'test'
12
13     GCP_PROJECT = 'DevConnect-project'
14
15     ARTIFACT_REGISTRY_REGION = 'us-central1'
16
17   }
```

☒ Use Groovy Sandbox ?

[Pipeline Syntax](#)

Save

Apply

הסבר כללי:

בשלב הזה הייתי צריכה ליצור מפתח SSH ולקשר אותו לגיטהאב ולג'נקינס (שככה הוא יוכל למשוך את התמונה) ואז ביצירת הפייפליין שמתי את הקישור של ה-SSH לגיטהאב וקישרתי את המפתח שייצרתי. (פירוט על שגיאה שנתקלתי כאן בהמשך הקובץ)

פרויקט גמר- DevConnect

- כתיבת הקוד pipeline: (העתקתי לvs code בשביל להעתיק בצורה מסודרת לכאן)

```
pipeline {
  agent any
  environment {
    DOCKERFILE_PATH = 'Dockerfile' // Path to your Dockerfile
    IMAGE_NAME = 'app'
    IMAGE_TAG = 'test'
    GCP_PROJECT = 'DevConnect-project'
    ARTIFACT_REGISTRY_REGION = 'us-central1'
    ARTIFACT_REGISTRY_REPO =
'me-west1-docker.pkg.dev/devconnect-project/yehudit-artifacts'
  }
  stages {
    stage('Checkout') {
      steps {
        checkout([class: 'GitSCM',
                  branches: [[name: 'main']], // Replace with your
branch
                  userRemoteConfigs: [[url:
'git@github.com:yehuditHoro/jenkins-final-project.git',
                  credentialsId: 'devconnect-app']]])
      }
    }
    stage('Build and Push Docker Image') {
      steps {
        script {
          dir("Django-WebApp/django_web_app") {
            def gitCommitMessage = sh(script: 'git log -1
--pretty=%B', returnStdout: true).trim()
            def containerName = "django_app-${gitCommitMessage}"
            // Stop and remove the existing container if it exists
            sh "docker stop ${containerName} || true"
            sh "docker rm ${containerName} || true"
            // Build and run the new container
            sh "docker build -t django_app:${gitCommitMessage} ."
            sh "docker run -d --name ${containerName} -p 8000:8000
django_app:${gitCommitMessage}"
          }
        }
      }
    }
  }
}
```

פרויקט גמר- DevConnect

```
    }
  }
  stage('Test Application') {
    steps {
      script {
        dir("Django-WebApp/django_web_app"){
          echo 'Running Django tests...'
          def gitCommitMessage = sh(script: 'git log -1
--pretty=%B', returnStdout: true).trim()
          def containerName = "django_app-${gitCommitMessage}"
          // Run Django tests within the Docker container
          sh "docker exec ${containerName} python manage.py
test"

          // // Define the URL of your app
        }
      }
    }
  }
  stage('Push Docker Image to Artifact Registry') {
    steps {
      script {
        dir("DevConnect Django CICD") {
          echo "Build and Push Docker Image..."
          def gitCommitMessage = sh(script: 'git log -1
--pretty=%B', returnStdout: true).trim()
          def containerName =
"django_app-${gitCommitMessage}"
          def imageTag =
"me-west1-docker.pkg.dev/devconnect-project/yehudit-artifacts/dev_connect:
${gitCommitMessage}"
          sh "docker tag django_app:${gitCommitMessage}
${imageTag}"

          // Authenticate with Google Cloud (if not
already authenticated)
          sh "gcloud auth configure-docker
me-west1-docker.pkg.dev"

          // Push the Docker image to Artifact Registry
          sh "docker push ${imageTag}"
        }
      }
    }
  }
}
```

פרויקט גמר- DevConnect

```
}  
}  
}  
  
}
```

מקור:

<https://www.jenkins.io/doc/book/pipeline/syntax>

<https://www.youtube.com/watch?v=TiTrcFEsj7A>

הרצה סופית (build now):

ובסופו של דבר לאחר שינויים בקוד בהתאם לשגיאות שקיבלתי, הקוד רץ ועובד תקין!



הישורים הבועים

הסבר על ה-pipeline:

פייפליין הוא סדרה של משימות או תהליכים שנועדו לבצע פעולה מוגדרת, בפייפליין אנחנו מגדירים שלבים שבכל שלב יש פקודות שונות שעליו לבצע. כאשר אנחנו מריצים את הפייפליין הוא מתחיל שלב אחרי שלב ואם הוא נתקע בשלב מסוים הוא לא ממשיך אל שאר השלבים, במקרה שלנו בשלב הראשון הוא מושך את הגרסה האחרונה של התמונה מגיטהב, בשלב השני הוא בונה ודוחף את התמונה, בשלישי הוא מריץ את הניסוי ואם הוא אכן מצליח, ברביעי הוא דוחף את התמונה לארטיפקט.

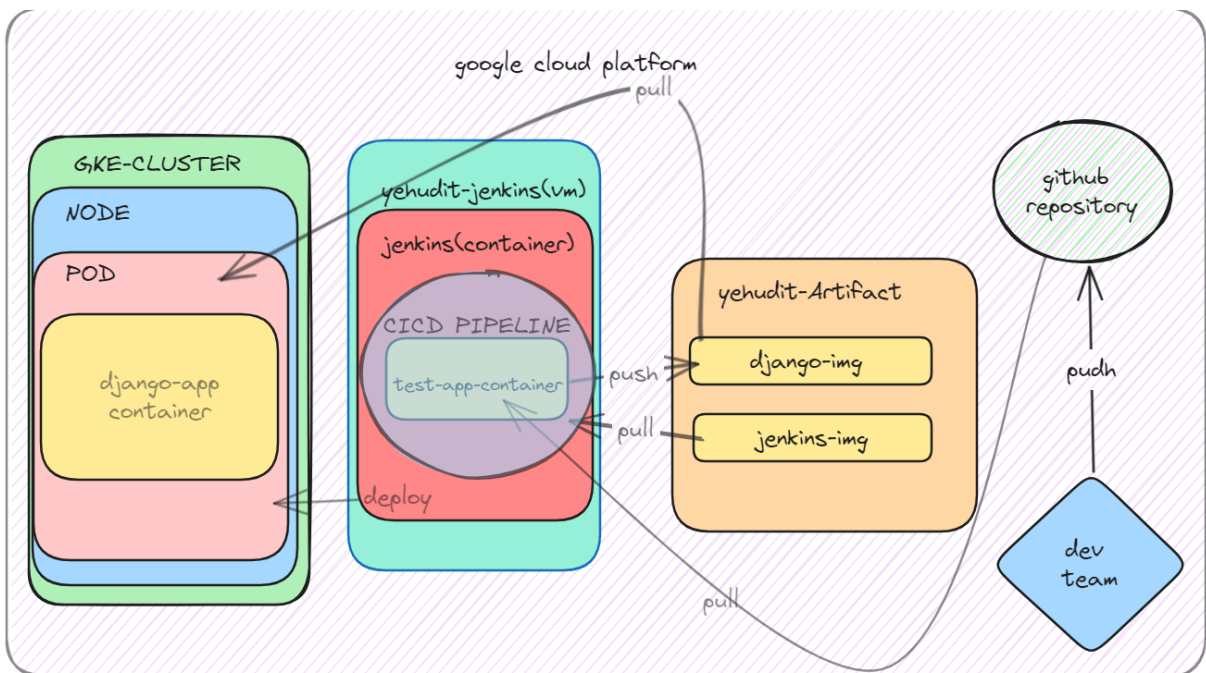
פרויקט גמר- DevConnect

| | | | | | | |
|--|--|-------------|--------|--------------|--------------|--|
| DevConnect-project | Search (/) for resources, docs, products, and more | Search | | | | |
| Digests for dev_connect | | | | | | |
| me-west1-docker.pkg.dev > devconnect-project > yehudit-artifacts > dev_connect | | | | | | |
| Filter Enter property name or value | | | | | | |
| | Name | Description | Tags | Created | Updated | |
| | 1af12d46f127 | | bla | Just now | Just now | |
| | d17c16187bc6 | | v0.0.1 | 12 hours ago | 12 hours ago | |

Built Nov 12, 2023, 10:58:12 PM
Created Nov 12, 2023, 11:08:02 PM
Updated Nov 12, 2023, 11:08:02 PM
Tags bla

עכשיו הרצתי שוב את הגרסה העדכנית והעלתי אותה (בעזרת ה deploy כמו בחלק ב' בשלב האחרון...)

• הארכיטקטורה של הפרויקט:



פרויקט גמר- DevConnect

התקלות שנתקלתי בהם:

* כשניסיתי להריץ את הקונטיינר הוא אמנם הריץ אבל נפל כשנכנסתי לאתר:

TemplateSyntaxError at /

'staticfiles' is not a registered tag library. Must be one of:

admin_list
admin_modify
admin_urls
cache
crispy_forms_field
crispy_forms_filters
crispy_forms_tags
crispy_forms_utils
i18n
l10n
log
static
tz

Request Method: GET

Request URL: http://localhost:8000/

Django Version: 3.2.5

Exception Type: TemplateSyntaxError

Exception Value: 'staticfiles' is not a registered tag library. Must be one of:

admin_list
admin_modify
admin_urls
cache
crispy_forms_field
crispy_forms_filters
crispy_forms_tags
crispy_forms_utils
i18n
l10n
log
static
tz

Exception Location: /usr/local/lib/python3.9/site-packages/django/template/defaulttags.py, line 1036, in find_library

Python Executable: /usr/local/bin/python

Python Version: 3.9.18

Python Path: ['/app',
'/usr/local/lib/python3.9.zip',
'/usr/local/lib/python3.9',

ולכן שיניתי בקובץ settings.py:

```
50  
57 TEMPLATES = [  
58     {  
59         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
60         'DIRS': [],  
61         'APP_DIRS': True,  
62         'OPTIONS': {  
63             'context_processors': [  
64                 'django.template.context_processors.debug',  
65                 'django.template.context_processors.request',  
66                 'django.contrib.auth.context_processors.auth',  
67                 'django.contrib.messages.context_processors.messages',  
68             ],  
69             'libraries': {  
70                 'staticfiles': 'django.templatetags.static',  
71             }  
72         },  
73     },  
74 ]  
75
```

```
STATIC_ROOT= os.path.join(BASE_DIR, '/static/'),
```

פרויקט גמר- DevConnect

*כשניסיתי לדחוף את התמונה לארטיפקט נתקלתי בבעיות הרשאה:

```
denied: Permission "artifactregistry.repositories.uploadArtifacts" denied on resource
"projects/devconnect-final-project/locations/me-west1/repositories/yehudit-artifacts" (or it may
(not exist
Untagged:
me-west1-docker.pkg.dev/devconnect-final-project/yehudit-artifacts/dev_connect:0.0.1
```

ניסיתי להריץ את הפקודות הבאות:
gcloud auth login
gcloud auth configure-docker

וזה עדיין לא עבד
ואז הרצתי את 2 הפקודות הבאות:
gcloud config unset auth/impersonate_service_account
gcloud auth login
gcloud config set project devconnect-final-project
עכשיו זה עבד!!
*כשהרמתי את ה-deployment הוא עשה לי את השגיאה הזאת:

The screenshot shows a Jenkins console output for a deployment. It includes a status bar at the top with a checkbox, the name 'app-yehudit', a red warning icon and text 'Does not have minimum availability', the word 'Deployment', '0/1', the environment 'production', and a link 'yehudit-cluster'. The main output text says 'בגלל שלא שמתי שמות תואמים בקבצים של ה- deployment ושל ה-load balancer שיניתי וזה עבד.'

כשיצרתי מפתח פרטי (בשביל הפייפליין), הוא זרק לי את השגיאה הזאת:

```
! Failed to connect to repository : Command "git ls-remote -h -- git@github.com:yehuditHoro/
stdout:
stderr: Load key "/tmp/jenkins-gitclient-ssh14711254630614589114.key": error in libcrypto
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

והפתרון היה להריץ את השורה הזאת בתוך הקונטיינר:

```
"git ls-remote -h -- git@github.com:yehuditHoro/jenkins-final-project.git HEAD"
```

וזה עדיין לא עזר. אז הרצתי את הפקודה הזאת (עדיין בתוך הקונטיינר):

```
cat /var/jenkins_home/.ssh/id_rsa.pub
```

ואז העתקתי את המפתח הארוך, ואז זה עבד!