

דף הוראות התקנה:

1. יצירת סביבת עבודה ב-AWS Cloud9-

1. בפורטל, AWS, נווט ל **AWS Cloud9 → Create environment**
2. שם IDE: Microservices
3. Instance type: t3.small, מערכת הפעלה Amazon Linux 2
4. VPC: LabVPC, Subnet: Public Subnet1
5. צור ופתח את ה-IDE

2. הגדרת מפתחות גישה

```
bash
CopyEdit
#אם טרם עשיתם: configure
aws configure
#הכנסו Access Key, Secret Key, Region=us-east-1
```

3. העתקת קוד המקור של האפליקציה המונוליתית

1. הורדו את קובץ המפתח (labsuser.pem) מהלוח ותעלו אותו ל-Cloud9
2. העניקו הרשאות:

```
bash
CopyEdit
chmod 400 ~/environment/labsuser.pem
```
3. צרו תיקייה זמנית:

```
bash
CopyEdit
mkdir -p ~/environment/temp
```
4. סטו את ה-IP הפרטי של השרת (EC2 console) MonolithicAppServer
5. העתקו את הקוד:

```
bash
CopyEdit
scp -r -i ~/environment/labsuser.pem ec2-user@<MONO_PRIV_IP>:/home/ec2-user/resources/codebase_partner/* ~/environment/temp/
```

4. יצירת מבנה הפרויקט למיקרו-סרוויסים

```
bash
CopyEdit
cd ~/environment
mkdir microservices
cd microservices
mkdir customer employee
#העתקו את כל הקבצים מתוך temp לתוך כל תיקייה:
cp -R ~/environment/temp/* customer/
cp -R ~/environment/temp/* employee/
rm -rf ~/environment/temp
```

5. התחלת Git ו-CodeCommit

```
bash
CopyEdit
cd ~/environment/microservices
git init
git checkout -b dev
git add .
git commit -m "initial monolith copy for customer & employee"
#יצירת רפוזיטורי CodeCommit בשם microservices
git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/microservices
git push -u origin dev
```

6. התקנת Docker ובדיקת תוכנה

```
bash
CopyEdit
#בדקו התקנה:
docker --version
#אם לא מותקן:
```

```
sudo yum update -y
sudo amazon-linux-extras install docker -y
sudo service docker start
sudo usermod -a -G docker ec2-user
הערה: ייתכן שתצטרכו להתנתק ולהתחבר מחדש ל-Cloud9.
```

7. הגדרת משתני סביבה לחיבור ל-RDS.

```
bash
CopyEdit
#בחרו בנתוני חיבור מתוך app/config/config.js
export APP_DB_HOST=$(grep 'APP_DB_HOST' customer/app/config/config.js | cut -d '"' -f2)
echo $APP_DB_HOST
```

8. בניית והרצת התכולות המקומיות

לקוח (Customer)

```
bash
CopyEdit
cd ~/environment/microservices/customer
#בנייה
docker build -t customer .
#ריצה על פורט 8080
docker run -d --name customer_1 -p 8080:8080 -e APP_DB_HOST=$APP_DB_HOST customer
• גשו בכתובת http://<Cloud9_Public_IP>:8080 :
```

עובד (Employee)

```
bash
CopyEdit
cd ~/environment/microservices/employee
docker build -t employee .
docker run -d --name employee_1 -p 8081:8081 -e APP_DB_HOST=$APP_DB_HOST employee
```

- גשו בכתובת `http://<Cloud9_Public_IP>:8081/admin/suppliers`

9. העלאת תמונות ל-AWS ECR

bash

CopyEdit

#קבלת מספר חשבון

```
export ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
```

#התחברות ל-AWS ECR

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin $ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com
```

#יצירת רפוזיטוריות ב Console או CLI

```
aws ecr create-repository --repository-name customer
```

```
aws ecr create-repository --repository-name employee
```

#תיוג ודחירה

```
docker tag customer:latest $ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/customer:latest
```

```
docker tag employee:latest $ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/employee:latest
```

```
docker push $ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/customer:latest
```

```
docker push $ACCOUNT_ID.dkr.ecr.us-east-1.amazonaws.com/employee:latest
```

10. המשך הפרויקט

לאחר התקנת סביבת העבודה ובדיקת המיקרו-סרוויסים המקומיים, יש להמשיך בשלבים הבאים:

- יצירת ECS Cluster (Fargate)
- הגדרת ALB ו-Target Groups
- יצירת CodeCommit לרכיבי פריסה (taskdefs & AppSpec)
- הקמת שירותי ECS ו-CodePipeline & CodeDeploy