מה עשיתי?

להוסיף לדיפלוימנט

---

apiVersion: apps/v1

kind: Deployment

metadata:

  name: jupyterlab-gui-single

  labels:

    app: jupyterlab-gui-single

spec:

  replicas: 1

  selector:

    matchLabels:

      app: jupyterlab-gui-single

  template:

    metadata:

      labels:

        app: jupyterlab-gui-single

    spec:

      securityContext:

        runAsUser: 0  # רוץ כרוט זמנית לצורכי התקנה

      containers:

      - name: jupyterlab

        image: jupyter/base-notebook:latest

        env:

        - name: DISPLAY

          value: ":1"

        ports:

        - name: notebook

          containerPort: 8888

        command: ["/bin/bash", "-c"]

        args:

          - |

            apt-get update && \

            apt-get install -y xvfb x11-apps libx11-dev --no-install-recommends && \

            Xvfb :1 -screen 0 1024x768x16 & \

            start-notebook.sh --NotebookApp.token='' --NotebookApp.password='' --ip=0.0.0.0

        resources:

          requests:

            memory: "512Mi"

            cpu: "250m"

          limits:

            memory: "1Gi"

            cpu: "500m"

---

apiVersion: v1

kind: Service

metadata:

  name: jupyterlab-service-single

  labels:

    app: jupyterlab-gui-single

spec:

  selector:

    app: jupyterlab-gui-single

  ports:

  - name: notebook

    port: 8888

    targetPort: 8888

  type: NodePort

kubectl port-forward pod/jupyterlab-gui-single-586bb46687-z694s 2222:22

1 apt update && apt install -y openssh-server

2 mkdir -p /run/sshd

3 echo 'root:root' | chpasswd # קובע סיסמה לרוט

4 echo 'root:root' | chpasswd

5 /usr/sbin/sshd -D

6 useradd -m newuser

7 passwd newuser

8 mkdir -p /home/newuser/.ssh

9 chown newuser:newuser /home/newuser/.ssh

10 chmod 700 /home/newuser/.ssh

11 usermod -aG sudo newuser

12 /usr/sbin/sshd -D

13 /usr/sbin/sshd -D &

14 echo "X11Forwarding yes" >> /etc/ssh/sshd\_config

ssh -X newuser@localhost -p 2222

SSSD (System Security Services Daemon) הוא שירות בלינוקס שמאפשר למערכות להשתמש במקורות זהות חיצוניים כמו LDAP, Active Directory, Kerberos, FreeIPA ועוד.

באמצעות SSSD + LDAP, ניתן לאפשר אימות משתמשים ב-SSH עם האישורים שלהם מה-Directory (ללא צורך ביצירת חשבונות לוקליים לכל משתמש).

איך זה עובד?

כשמשתמש מנסה להתחבר לשרת דרך SSH:

הבקשה עוברת ל-SSSD, שבודק האם המשתמש קיים במטמון המקומי.

אם המשתמש לא קיים במטמון, SSSD שולח בקשה ל-שרת ה-LDAP.

אם האישורים תקפים, המשתמש מאומת ומקבל גישה למערכת.

התקנה והגדרה של SSSD לאימות SSH דרך LDAP

1. התקנת החבילות הדרושות

בשרת הלינוקס שלך, התקיני את SSSD והלקוח של LDAP:

bash

Copy

Edit

sudo apt update && sudo apt install -y sssd libnss-sss libpam-sss ldap-utils

(במערכות RedHat/Fedora אפשר להשתמש ב- dnf install sssd sssd-ldap).

2. הגדרת SSSD לעבודה עם LDAP

צרי קובץ קונפיגורציה:

bash

Copy

Edit

sudo nano /etc/sssd/sssd.conf

והוסיפי את התוכן הבא (התאימי לשרת ה-LDAP שלך):

ini

Copy

Edit

[sssd]

services = nss, pam

config\_file\_version = 2

domains = example.com

[domain/example.com]

id\_provider = ldap

auth\_provider = ldap

ldap\_uri = ldap://your-ldap-server

ldap\_search\_base = dc=example,dc=com

ldap\_tls\_reqcert = never

cache\_credentials = true

enumerate = false

# הגדרת מפות UID/GID

ldap\_id\_use\_start\_tls = true

ldap\_user\_uid\_number = uidNumber

ldap\_user\_gid\_number = gidNumber

שמרי את הקובץ וודאי שההרשאות מאובטחות:

bash

Copy

Edit

sudo chmod 600 /etc/sssd/sssd.conf

3. הפעלת השירות ובדיקת חיבור ל-LDAP

bash

Copy

Edit

sudo systemctl restart sssd

sudo systemctl enable sssd

sudo sssctl user-show username

אם המשתמש קיים ב-LDAP, תראי את פרטיו.

4. קביעת אימות SSH דרך LDAP

שני דברים נדרשים:

לאפשר התחברות דרך PAM

ודאי שהשורות הבאות קיימות ב-/etc/pam.d/common-auth:

cpp

Copy

Edit

auth [success=1 default=ignore] pam\_sss.so

לאפשר חיפוש משתמשים דרך NSS

ודאי שב-/etc/nsswitch.conf יש:

makefile

Copy

Edit

passwd: files sss

group: files sss

shadow: files sss

5. בדיקת התחברות SSH דרך LDAP

כעת, נסי להתחבר לשרת דרך SSH עם שם משתמש שנמצא ב-LDAP:

bash

Copy

Edit

ssh username@your-server

אם הכול הוגדר נכון, האימות יתבצע דרך LDAP.

יתרונות של שימוש ב-SSSD לאימות SSH דרך LDAP

✅ אין צורך בניהול משתמשים מקומיים – המשתמשים מאוחסנים ב-LDAP ולא בשרת.

✅ ניהול הרשאות מרכזי – אפשר לקבוע חוקים לפי קבוצות ב-LDAP.

✅ תמיכה ב-SSO (Single Sign-On) – אפשר לשלב Kerberos לאימות ללא סיסמה.

✅ ביצועים טובים עם מטמון – גם אם שרת ה-LDAP לא זמין, SSSD יכול לאפשר התחברות עם נתונים ששמר.

אם את רוצה להעמיק, אני יכולה לעזור להוסיף MFA (אימות דו-שלבי) או אינטגרציה עם Active Directory