# הטמעת מערכת SOX ב-Azure — מדריך למנהל ה-IT

מסמך זה מיועד לצוות ה-IT/DevOps להקמת סביבת ייצור (Production) ב-Azure. הוא נגזר מ-`SPEC.md` ו-`DECISIONS.md` (Azure, PostgreSQL+RLS, Entra ID, Graph, Blob). יש לקרוא יחד עם פרק האבטחה ב-`SPEC.md`.

> **סטטוס היישום:** האפליקציה בנויה ובדוקה (backend FastAPI, frontend React), עם שכבות הפשטה להחלפה לפרודקשן: אחסון (Local↔Azure Blob), מייל (Console↔Graph), AI (Heuristic↔Anthropic). ההחלפה מתבצעת דרך משתני סביבה בלבד — אין צורך בשינוי קוד.

---

## 1. ארכיטקטורת יעד

```
                    ┌──────────────────────────────────────┐
משתמשים  ──────►    │  Azure Front Door + WAF (OWASP, TLS)  │
                    └──────────────────────────────────────┘
                                   │ HTTPS בלבד
                    ┌──────────────┴──────────────┐
                    ▼                             ▼
┌──────────────────┐                 ┌──────────────────┐
│ Frontend (Static Web │              │ Backend (Container  │
│ App / nginx container)│  /api ──►   │ App / App Service)  │
└──────────────────┘                 └──────────────────┘
                                              │ Private Endpoint
                    ┌─────────────────────────┼──────────────┐
                    ▼                         ▼              ▼
              ┌──────────────────┐  ┌──────────────────┐  ┌──────────────┐
              │ PostgreSQL Flexible │  │ Blob Storage      │  │ Key Vault     │
              │ Server (+ RLS)     │  │ (ראיות, CMK)      │  │ (secrets/keys) │
              └──────────────────┘  └──────────────────┘  └──────────────┘
                    ▲                                          ▲
                    │ Entra ID (SSO יועצים) · Microsoft Graph (מייל)     │
                    └──────────────────────────────────────────────────┘
```

---

## 2. משאבי Azure נדרשים

| משאב | שירות מומלץ | תפקיד |
|---|---|---|
| Compute (Backend) | **Azure Container Apps** (או App Service for Containers) | מריץ את ה-API |
| Compute (Frontend) | **Azure Static Web Apps** (או Container Apps עם nginx) | מגיש את ה-UI |
| Database | **Azure Database for PostgreSQL — Flexible Server** (v16) | נתונים + RLS |
| אחסון קבצים | **Azure Blob Storage** (Container `sox-evidence`) | ראיות/אסמכתאות |
| סודות/מפתחות | **Azure Key Vault** | JWT secret, DB password, Graph/Anthropic keys, CMK |
| זהות | **Microsoft Entra ID** (App Registration) | SSO ליועצים + הרשאות Graph |

| משאב | שירות מומלץ | תפקיד |
|---|---|---|
| מייל | **Microsoft Graph** (אותה App Registration) | שליחה/קליטה של דוא"ל |
| הגנה | **Azure Front Door + WAF** | TLS, OWASP Top 10, חסימת IP |
| ניטור | **Application Insights + Log Analytics** | לוגים, מטריקות, התראות |
| רישום קונטיינרים | **Azure Container Registry (ACR)** | אחסון image-ים |

> כל המשאבים באותו **Region** (למשל West Europe) ובאותה **Resource Group**, עם **Private Endpoints** ל-DB/Blob/Key-Vault (ללא חשיפה ציבורית).

---

## 3. שלבי הקמה (סדר מומלץ)

### 3.1 תשתית בסיס

1. צור **Resource Group** ו-**VNet** עם subnet ל-Container Apps ו-subnet ל-Private Endpoints.
2. הקם **PostgreSQL Flexible Server v16**:
3. Compute: התחל מ- `Standard_D2ds_v5` (אפשר להקטין).
4. **Private access (VNet integration)** — ללא public endpoint.
5. גיבויים: שמירה 35 ימים, **geo-redundant** (תומך RPO<1h, RTO<4h מ-SPEC).
6. הפעל `require_secure_transport = ON` (TLS חובה).
7. הקם **Storage Account** + Container `sox-evidence` :
8. Redundancy: GRS/ZRS. **Soft delete** + **versioning** + **immutability policy** (תומך retention 7 שנים — R1, וב"החלפה בלבד" של ראיות).
9. שקול **Customer Managed Keys (CMK)** דרך Key Vault (הצפנה במנוחה).
10. הקם **Key Vault** ושמור: `JWT_SECRET_KEY` , סיסמת DB, `ENTRA_CLIENT_SECRET` , `ANTHROPIC_API_KEY` (בשלב ב'), מפתחות CMK.

### 3.2 זהות (Entra ID) — ל-SSO וגם ל-Graph

1. **App Registration** חדש (Single/Multi-tenant לפי מדיניות).
2. **Redirect URI** (Web): `https://<frontend-domain>/api/v1/auth/sso/entra/callback` .
3. **API permissions** (Microsoft Graph, Application):
4. `Mail.Send` (שליחת מיילים מהמערכת)
5. `Mail.Read` / `Mail.ReadWrite` + **Graph change notifications** (קליטת מייל נכנס)
6. `User.Read` (פרופיל בסיסי) — Grant admin consent.
7. צור **Client Secret** ושמור ב-Key Vault.
8. רשום את הערכים: `ENTRA_TENANT_ID` , `ENTRA_CLIENT_ID` , `ENTRA_CLIENT_SECRET` .

### 3.3 בנייה ופריסה

1. בנה ודחוף image-ים ל-**ACR**: `bash az acr build -r <acr> -t sox-backend:latest -f backend/Dockerfile . az acr build -r <acr> -t sox-frontend:latest -f frontend/Dockerfile ./frontend`
2. פרוס **Container App** ל-backend עם משתני הסביבה (§4), עם **Managed Identity** לקריאת Key Vault.
3. פרוס **frontend** (Static Web App מ-build, או Container App עם ה-nginx image).
4. חבר **Front Door + WAF** מול ה-frontend, עם routing של `/api/*` ל-backend.

### 3.4 אתחול ראשוני

הרץ את סקריפט ה-bootstrap פעם אחת ליצירת ה-tenant והאדמין הראשון (chicken-and-egg — ראה `SPEC.md` ):

```
python -m app.scripts.bootstrap \
  --name "Entropy" --subdomain entropy \
```

```
  --admin-email admin@entropy.com --admin-password '<strong>' \
  --ad-tenant-id <entra_tenant_id>
```

> ב-Container Apps אפשר להריץ כ-**Job** חד-פעמי, או דרך `az containerapp exec`. המיגרציות (`alembic upgrade head`) רצות אוטומטית ב-entrypoint של ה-image.

---

## 4. משתני סביבה (Backend)

| משתנה | ערך לדוגמה / מקור | הערה |
|---|---|---|
| `DATABASE_URL` | `postgresql+asyncpg://user:***@<pg-host>:5432/sox` | מ-Key Vault |
| `JWT_SECRET_KEY` | (סוד חזק) | מ-Key Vault |
| `ENTRA_TENANT_ID` / `ENTRA_CLIENT_ID` / `ENTRA_CLIENT_SECRET` | Entra App | סוד מ-Key Vault |
| `ENTRA_REDIRECT_URI` | `https://<frontend>/api/v1/auth/sso/entra/callback` | |
| `AZURE_STORAGE_CONNECTION_STRING` | מחרוזת חיבור Blob | מפעיל אוטומטית את ה-AzureBlobStorage |
| `AZURE_BLOB_CONTAINER` | `sox-evidence` | |
| `EMAIL_PROVIDER` | `graph` | מפעיל את GraphSender |
| `EMAIL_FROM_ADDRESS` | `sox@entropy.com` | תיבת השליחה |
| `AI_PROVIDER` | `heuristic` (שלב א') → `anthropic` (שלב ב') | ראה R2 |
| `ANTHROPIC_API_KEY` | (שלב ב') | מ-Key Vault, אחרי DPA |
| `AI_MODEL` | `claude-sonnet-4-6` | |
| `DEFAULT_LOCALE` / `SUPPORTED_LOCALES` | `he` / `he,en` | |

> **עיקרון:** אין סודות בקוד או ב-image. כולם ב-Key Vault, נטענים בזמן ריצה דרך Managed Identity.

---

## 5. רשת ואבטחה (תואם פרק האבטחה ב-SPEC)

- **HTTPS/TLS בלבד** — נאכף ב-Front Door; הפניית HTTP→HTTPS.
- **WAF** — מדיניות OWASP Top 10 + rate limiting + הגבלת IP (allow-list) במידת הצורך.
- **Private Endpoints** ל-PostgreSQL, Blob, Key Vault — אין גישה ציבורית.
- **RLS** — נאכף בקוד (כל שאילתה לפי `tenant_id` מה-JWT) **וגם** ברמת ה-DB (policies שנוצרות במיגרציות). אין צורך בהגדרה נוספת.
- **Audit log append-only** — נאכף ע"י trigger במיגרציה (אין UPDATE/DELETE).
- **Soft-delete + retention 7 שנים** — ברמת האפליקציה + immutability policy ב-Blob (R1).
- **Brute force / ניהול סשנים** — JWT קצר-מועד + refresh; שקול חסימה ב-WAF/Front Door.
- **גיבוי ושחזור** — PostgreSQL geo-redundant (RTO<4h, RPO<1h); בדיקות שחזור תקופתיות.
- **הפרדת סביבות** — Resource Groups נפרדים ל-Dev/Test/Prod.

- **PT (Penetration Test)** — לבצע לפני מסירת סביבה ללקוח (כפי שמוגדר ב-SPEC).

---

## 6. CI/CD (אופציונלי, מומלץ)

ה-CI הקיים ( `.github/workflows/ci.yml` ) מריץ lint + מיגרציות על PostgreSQL + טסטים (שער כיסוי 80%) + build ל-frontend. להוספת **CD**: 1. הוסף workflow שבונה ודוחף image-ים ל-ACR על merge ל- `main` . 2. עדכן את ה-Container App ל-image החדש ( `az containerapp update` ). 3. המיגרציות רצות אוטומטית ב-entrypoint בכל עליית גרסה (idempotent).

---

## 7. רולאאוט הדרגתי (תואם נספח SPEC)

| שלב | תוכן | תצורה |
|---|---|---|
| א' | שימוש פנימי ביועצים בלבד; מיילים יוצאים בלבד | `EMAIL_PROVIDER=graph` , ללא כניסת לקוחות |
| ב' | פיילוט עם מספר לקוחות; כניסה למערכת | פתיחת SSO/Local ללקוחות נבחרים |
| ג' | הרחבה לכלל הלקוחות | — |
| ד' | אוטומציה לבדיקת בקרות (AI) | `AI_PROVIDER=anthropic` (אחרי R2: DPA + data-residency) |
| ה' | כניסת רו"ח מבקר | — |

---

## 8. פריטים פתוחים לאישור לפני Production

| # | פריט | אחראי |
|---|---|---|
| R1 | אישור משפטי ל-retention 7 שנים + immutability policy | יועמ"ש |
| R2 | DPA עם Anthropic + הכרעת data-residency לפני הפעלת AI | IT + יועמ"ש |
| R3 | חתימת אדריכל על `SPEC.md` / `DECISIONS.md` | אדריכל |
| — | ביצוע PT וטיפול בממצאים קריטיים | IT Security |
| — | הגדרת CMK ו-immutability ב-Blob | IT |