

Sentinel DB 360 – Kurulum Kılavuzu (3 Senaryo, Çoklu Linux)

Sürüm: v1.7.11 · **Tarih:** 26 Mayıs 2026 (13 sprintlik rakip parity backlog'u tamam – 126 unit test, 0 schema migration) **Hedef kitle:** Elinde tek bir Linux makinesi olan kullanıcı – Ubuntu / Debian / RHEL / Rocky / Alma / CentOS / Fedora / openSUSE / Arch / Alpine. Her senaryoda **tek komutluk** kurulum.

İçindekiler

1. Hangi senaryo sana uyar?
2. Ön koşullar
3. ACR erişim hazırlığı (3 senaryo için ortak)
4. Senaryo A – DMC kendi `demo.sentineldb360.com` showcase'ini kur
5. Senaryo B – Müşteri kendi ortamında Demo / Showcase kur
6. Senaryo C – Müşteri Boş Sentinel kur (kendi DB'leri ekleyecek)
7. Linux dağıtımına göre özel notlar
8. Operasyonel komutlar – log, restart, backup, upgrade
9. Sorun giderme – Troubleshooting
10. Air-gap / kapalı ağ kurulumu (image bundle)

1. Hangi senaryo sana uyar?

Senaryo	Kim için	Ne içerir	Tek-komut
A – DMC Demo	DMC'nin kendi <code>demo.sentineldb360.com</code> showcase'i	4 motor + sample DB + traffic-gen + auto demo-login	<code>curl ... demo.sh \\ sudo bash</code>
B – Müşteri Demo	Müşteri kendi VM'inde aynı showcase	A ile birebir aynı (paket aynı, sadece domain değişir)	<code>curl ... demo.sh \\ sudo bash</code>
C – Müşteri Boş Sentinel	Müşteri production / pilot ortamı	Sıfır data, sıfır source – UI'dan kendi DB'lerini ekler	<code>curl ... install.sh \\ sudo bash</code>

Karar matrisi: - "Sentinel'i tanımak / pre-sales demo göstermek" → **A** veya **B** - "Kendi production DB'lerimi izlemek istiyorum" → **C**

A ve B aynı `demo.sh` script'ini kullanır. C farklı (`install.sh`).

2. Ön koşullar

Donanım

Senaryo	CPU	RAM	Disk
A/B (showcase)	4 vCPU	16 GB	60 GB SSD
C (boş prod)	2 vCPU (4 önerilen)	4 GB (8 GB önerilen)	40 GB SSD

Yazılım

- **Linux** (aşağıdaki dağıtımlardan biri — script otomatik tespit eder)
- **root / sudo** yetkisi (Docker kurulumu için)
- **İnternet erişimi** (ACR + Docker Hub'a pull için) — air-gap ortam için Bölüm 10
- **Açık port 80** (HTTP) ve 443 (HTTPS varsa)

Script ne yapar otomatik?

- OS tespiti (Ubuntu/Debian/RHEL/Rocky/Alma/Fedora/openSUSE/Arch/Alpine)
- Docker + Docker Compose plugin yoksa **otomatik kurulum**
- Azure CLI (`az`) yoksa otomatik kurulum
- ACR login (interaktif veya service principal)
- `docker-compose.yml` + `nginx.conf` + `.env` örneğini `sentineldb360.com/install/...` 'den **indirir**
- `.env` için secret'ları (JWT, SECRET_KEY, APP_ENC_KEY, vb.) **otomatik üretir** ve `chmod 600` ile yazar
- Firewall (ufw / firewalld) varsa 80+443 otomatik açar
- `docker compose pull` (image'lar ACR'dan, ~2.5 GB ilk seferde)
- `docker compose up -d`
- Health check + smoke test
- (Demo) Sample data bootstrap

Kurulum süresi: Docker zaten varsa ~5dk, Docker yoksa ~10dk.

3. ACR erişim hazırlığı (3 senaryo için ortak)

DMC Azure Container Registry'sinden (`dmcacr22819`) image çekmek için bir kez **giriş** yapılır. 3 yöntem:

3.1. Browser-based (interactive — DMC kendisi için)

```
az login
az acr login --name dmcacr22819
```

3.2. Service Principal (müşteri için — DMC sağlar)

DMC sana üç bilgi gönderir: `client-id` , `client-secret` , `tenant-id` . Şununla giriş yap:

```
az login --service-principal \  
  --username <client-id> \  
  --password <client-secret> \  
  --tenant <tenant-id>  
az acr login --name dmcacr22819
```

Script ayrıca SENTINEL_ACR_USERNAME + SENTINEL_ACR_PASSWORD env'lerini de okuyabilir — bu durumda az login adımı gerekmez: `bash sudo SENTINEL_ACR_USERNAME=<client-id> SENTINEL_ACR_PASSWORD=<secret> \ bash demo.sh`

3.3. Managed Identity (Azure VM ise)

```
az login --identity  
az acr login --name dmcacr22819
```

Eğer az login adımı **kurulumdan önce** yapılmamışsa, script seni durduracak ve seçenekleri gösterecektir. Login sonrası scripti yeniden çalıştır.

4. Senaryo A — DMC kendi demo.sentineldb360.com showcase'ini kur

Hedef kitle: DMC ekibi · DMC'nin pre-sales / showcase ortamı.

Adım 1 — VM hazırla (Azure örneği)

```
RG=rgSentinelDB360_demo; LOC=westeurope; VM=sentineldb360-demo  
  
az group create -n $RG -l $LOC  
az vm create \  
  --resource-group $RG --name $VM \  
  --image Ubuntu2204 --size Standard_B4ms \  
  --admin-username azureuser --generate-ssh-keys \  
  --public-ip-sku Standard --storage-sku Premium_LRS \  
  --os-disk-size-gb 60 --nsg-rule SSH --assign-identity  
  
az vm open-port -g $RG -n $VM --port 80 --priority 1010  
az vm open-port -g $RG -n $VM --port 443 --priority 1020  
  
# ACR AcrPull rolü VM'in managed identity'sine  
PRINCIPAL_ID=$(az vm show -g $RG -n $VM --query identity.principalId -o tsv)  
ACR_ID=$(az acr show --name dmcacr22819 --query id -o tsv)  
az role assignment create --assignee $PRINCIPAL_ID --scope $ACR_ID --role AcrPull
```

```
VM_IP=$(az vm show -d -g $RG -n $VM --query publicIps -o tsv)
echo "VM IP: $VM_IP"
```

Cloudflare DNS: A record `demo.sentineldb360.com` → `<VM_IP>` (Proxied önerilir).

Adım 2 – VM'e SSH + ACR login

```
ssh azureuser@$VM_IP
# VM içinde:
az login --identity
az acr login --name dmcacr22819
```

Adım 3 – Tek-komut kurulum

```
curl -fsSL https://sentineldb360.com/install/demo.sh | sudo bash
```

Script otomatik: - OS tespit + Docker + Azure CLI kurulum (gerekirse) - `/opt/sentinel-demo` dizinine compose + nginx + env indirme - 14 image pull (~2.5 GB) - 13 servis `up -d` (mongo, redis, backend, frontend, collector, ingestor, advisor, demo-mssql, demo-postgres, demo-mysql, demo-mongo, traffic-gen, nginx + sentinel-bootstrap) - `sentinel-bootstrap` container otomatik tetiklenir → admin user + 4 demo source + seed - Smoke test

Domain set etmek istersen:

```
curl -fsSL https://sentineldb360.com/install/demo.sh \
| sudo SENTINEL_DOMAIN=demo.sentineldb360.com bash
```

Adım 4 – TLS (opsiyonel)

A) Cloudflare proxied (en kolay): Cloudflare DNS → A record proxied. SSL Cloudflare termination.

B) Let's Encrypt origin:

```
sudo certbot certonly --standalone -d demo.sentineldb360.com
sudo cp /etc/letsencrypt/live/demo.sentineldb360.com/fullchain.pem /opt/sentinel-demo/
nginx/ssl/cert.pem
sudo cp /etc/letsencrypt/live/demo.sentineldb360.com/privkey.pem /opt/sentinel-demo/
nginx/ssl/key.pem
sudo docker compose -f /opt/sentinel-demo/docker-compose.demo.yml --env-file /opt/
sentinel-demo/.env.demo restart nginx
```

Adım 5 – Test

Browser: `http://<VM_IP>/` veya `https://demo.sentineldb360.com/` - Otomatik demo viewer login (kayıt formu yok) - 4 source aktif, traffic-gen sürekli yük üretiyor - NOC dashboard → CPU/Memory canlı trend (WebSocket) - Read-only: kill/save tuşlarına basınca toast "Demo modu — salt-okuma"

```
curl -I http://localhost/ # 200 OK
curl -I -X POST http://localhost/api/sources # 403 (read-only enforce)
sudo docker compose -f /opt/sentinel-demo/docker-compose.demo.yml ps
```

5. Senaryo B – Müşteri kendi ortamında Demo / Showcase

A ile aynı script + aynı imajlar. Müşteri kendi VM'inde showcase'i çalıştırır (Sentinel'i değerlendirme veya kendi pre-sales demosu için).

Tek farklar: - **VM'in hostingi:** müşteri kendi cloud (AWS/GCP/Azure/on-prem) - **ACR erişimi:** Service Principal (DMC sağlar) - **Domain:** müşterinin kendi (`demo.sirket.com.tr`)

Adım 1 – VM hazırla (herhangi cloud / on-prem)

Cloud	VM tipi
AWS	<code>t3.xlarge</code> (4 vCPU / 16 GB) – Ubuntu 22.04 AMI
GCP	<code>n2-standard-4</code> – Ubuntu 22.04
Azure	<code>Standard_B4ms</code> – Ubuntu 22.04
On-prem	KVM/VMware/Hyper-V – herhangi Linux dağıtımı

Disk 60 GB SSD, port 80 + 443 (ve 22 SSH).

Adım 2 – VM'e SSH + ACR login (Service Principal)

DMC'nin sağladığı `client-id`, `client-secret`, `tenant-id` ile:

```
ssh <user>@<vm-ip>
sudo az login --service-principal \
  --username <client-id-from-dmc> \
  --password <client-secret-from-dmc> \
  --tenant <tenant-id-from-dmc>
sudo az acr login --name dmcacr22819
```

Daha pratik: ACR credentials'i `demo.sh` 'a env olarak verebilirsin — bu durumda `az login` gerekmez:

```
bash curl -fsSL https://sentineldb360.com/install/demo.sh | \ sudo
SENTINEL_ACR_USERNAME=<client-id> \ SENTINEL_ACR_PASSWORD=<secret> \
SENTINEL_DOMAIN=demo.sirket.com.tr \ bash
```

Adım 3 – Tek-komut kurulum

```
curl -fsSL https://sentineldb360.com/install/demo.sh | \
sudo SENTINEL_DOMAIN=demo.sirket.com.tr bash
```

A ile aynı işlem (Docker kurulumu, image pull, compose up, bootstrap).

Adım 4 – DNS + TLS

A senaryosu ile aynı (Cloudflare proxied veya Let's Encrypt).

Adım 5 – Test

Browser: `http://<vm-ip>/` veya `https://demo.sirket.com.tr/` . A ile birebir aynı.

6. Senaryo C – Müşteri Boş Sentinel

Hedef kitle: Müşterinin canlı production veya pilot ortamı. **Sıfır data, sıfır source** – UI'dan kendi DB'lerini ekler. Bu adımlar **showcase değil, gerçek production**.

Adım 1 – VM hazırla

A senaryosundaki gibi (herhangi cloud), ama: - VM tipi `Standard_B2ms` (2 vCPU / 4 GB) yeterli – büyük müşteri ise `B4ms` - Disk 40 GB SSD (Mongo metadata için) - Port 80 + 443

Adım 2 – VM'e SSH + ACR login

A senaryosu ile aynı.

Adım 3 – Mongo data dizini

```
sudo mkdir -p /opt/sentineldb360/mongo-data
sudo chown -R 999:999 /opt/sentineldb360/mongo-data
```

Script bu adımı otomatik yapar; manuel kontrol için yukarıdaki komut.

Adım 4 – Tek-komut kurulum

```
# En basit (admin email/parola interaktif sorulur):
curl -fsSL https://sentineldb360.com/install.sh | sudo bash
```

Veya **non-interactive** (CI/automation için):

```
curl -fsSL https://sentineldb360.com/install.sh | sudo \
  SENTINEL_MODE=poc \
  SENTINEL_ADMIN_EMAIL=admin@sirket.com.tr \
  SENTINEL_ADMIN_PASS='Güçlü123!Parola' \
  SENTINEL_DOMAIN=sentinel.sirket.com.tr \
  bash
```

Script otomatik: - OS tespit + Docker + Azure CLI kurulum - /opt/sentineldb360/ dizinine compose + nginx + env indirme - 5 servis (backend, frontend, collector, ingestor, advisor) + mongo + redis + nginx pull (~1.5 GB) - `compose up -d` - Admin user oluştur (`.env` 'deki credentials ile) - 14 günlük trial license aktif (`SENTINEL_MODE=poc` default)

Production mode (lisans gerekir):

```
curl -fsSL https://sentineldb360.com/install.sh | sudo \
  SENTINEL_MODE=production \
  SENTINEL_LICENSE_KEY="eyJhbGc..." \
  SENTINEL_ADMIN_EMAIL=admin@sirket.com.tr \
  SENTINEL_ADMIN_PASS='...' \
  bash
```

Adım 5 – İlk login + source ekleme

Browser: `http://<vm-ip>/` veya `https://sentinel.sirket.com.tr/` - Login: `.env.prod` 'taki `SENTINEL_ADMIN_EMAIL` + `SENTINEL_ADMIN_PASS` - İlk login'de **hemen UI'dan parolayı değiştir** (Admin → Users → Edit Self → Change Password)

İlk source ekleme: 1. Admin → Sources → + **Yeni Source** 2. Bilgileri doldur (engine + host + port + db + user + password) 3. **Test bağlantı** → yeşil olursa Kaydet 4. ~60 saniye sonra collector metric topluyor

Adım 6 – Target DB'de monitoring user yetki (zorunlu)

SQL Server:

```
CREATE LOGIN sentinel_monitor WITH PASSWORD = '...';
USE master; CREATE USER sentinel_monitor FOR LOGIN sentinel_monitor;
GRANT VIEW SERVER STATE TO sentinel_monitor;
GRANT VIEW ANY DEFINITION TO sentinel_monitor;
GRANT VIEW ANY DATABASE TO sentinel_monitor;
-- Her izlenecek DB için:
USE <db>; CREATE USER sentinel_monitor FOR LOGIN sentinel_monitor;
GRANT VIEW DATABASE STATE TO sentinel_monitor;
```

PostgreSQL:

```
CREATE USER sentinel_monitor WITH PASSWORD '...';
GRANT pg_monitor TO sentinel_monitor;
-- pg_stat_statements aktif olmalı:
-- postgresql.conf: shared_preload_libraries = 'pg_stat_statements'
-- restart sonrası: CREATE EXTENSION pg_stat_statements;
```

MySQL:

```
CREATE USER 'sentinel_monitor'@'%' IDENTIFIED BY '...';
GRANT SELECT, PROCESS, REPLICATION CLIENT ON *.* TO 'sentinel_monitor'@'%';
```

```
GRANT SELECT ON performance_schema.* TO 'sentinel_monitor'@'%';
FLUSH PRIVILEGES;
```

MongoDB:

```
db.getSiblingDB("admin").createUser({
  user: "sentinel_monitor",
  pwd: "...",
  roles: [
    { role: "clusterMonitor", db: "admin" },
    { role: "read", db: "local" }
  ]
});
```

Adım 7 – TLS (A senaryosu ile aynı)

7. Linux dağıtımına göre özel notlar

Script bu OS'leri otomatik tespit eder ve uygun paket yöneticisini kullanır. Aşağıda **özel notlar** (sadece sıkıntı çıkarsa):

OS	Paket yöneticisi	Özel not
Ubuntu 22.04 / 24.04	apt	Tek-komut çalışır. Docker auto-install OK.
Debian 12	apt	Aynı, Ubuntu ile birebir.
RHEL 9 / Rocky 9 / Alma 9	dnf	SELinux Permissive veya Disabled olmalı (<code>getenforce</code>).
CentOS Stream 9	dnf	RHEL ile aynı.
Fedora 40 / 41	dnf	OK. cgroup v2 default.
openSUSE Leap 15	zypper	Azure CLI repo ekleme manuel olabilir.
openSUSE Tumbleweed	zypper	Aynı.
Arch Linux	pacman	Azure CLI AUR'da : önce <code>yay -S azure-cli</code> , sonra <code>SKIP_AZ_INSTALL=yes</code> ile script.
Alpine Linux 3.18+	apk	Azure CLI pip venv ile kurulur (script otomatik halleder).

Script özel davranış (Arch için):

```
# Arch'ta önce Azure CLI'yı AUR'dan kur, sonra:
curl -fsSL https://sentineldb360.com/install/demo.sh | \
sudo SKIP_AZ_INSTALL=yes bash
```

Script tetikleme sırası (her dağıtımda aynı): 1. `/etc/os-release` parse → OS_ID tespit 2. Docker yoksa → `repo + install + systemctl enable --now docker` 3. Azure CLI yoksa → resmi paket / pip venv 4. Compose + nginx + env örnekleri `sentineldb360.com/install/`'tan indir 5. `.env` üret (otomatik secret'lar) 6. `docker compose pull + up -d` 7. Smoke test

8. Operasyonel komutlar

Hangi senaryoda olursan ol, izin ve compose dosya adı farklı:

```
# Senaryo A/B (showcase)
DIR=/opt/sentinel-demo
COMPOSE_FILE=docker-compose.demo.yml
ENV_FILE=.env.demo

# Senaryo C (boş prod)
DIR=/opt/sentineldb360
COMPOSE_FILE=docker-compose.prod.yml
ENV_FILE=.env.prod

cd $DIR
alias dc="sudo docker compose -f $COMPOSE_FILE --env-file $ENV_FILE"
```

Sık kullanılan komutlar:

```
dc ps # Container durumu
dc logs -f backend # Backend log canlı
dc logs --tail 100 collector # Son 100 satır
dc restart backend # Tek servis restart
dc down # Stop (data kalır)
dc up -d # Start
dc pull && dc up -d # Image upgrade (örn. v1.7.12 yayınlandığında)

# Mongo backup (Senaryo C için kritik)
sudo docker exec sentinel-mongo mongodump --out /backup/$(date +%F)
sudo docker cp sentinel-mongo:/backup/$(date +%F) ~/sentinel-backups/

# Disk temizlik (Docker dangling image)
sudo docker system prune -af
```

Yeni sürüm upgrade (örn. v1.7.12 yayınlandığında):

```
sed -i 's/IMAGE_TAG=v0\..6\..2/IMAGE_TAG=v1.7.12/' $ENV_FILE
dc pull
dc up -d
```

9. Sorun giderme

Sorun	Olası neden + çözüm
Failed to pull image: unauthorized	ACR login expire. <code>az acr login --name dmcacr22819</code> yeniden çalıştır.
Backend restart loop	MongoDB henüz hazır değildir. 2dk bekle veya <code>dc restart backend</code> . Log: <code>dc logs --tail 50 backend</code> .
MongoDB connection refused	<code>/opt/sentineldb360/mongo-data</code> izinleri uid 999 olmalı: <code>sudo chown -R 999:999 /opt/sentineldb360/mongo-data</code>
Frontend 502 Bad Gateway	Backend container <code>running</code> mı? <code>dc ps</code> . Nginx config'inde resolver/upstream sorunu olabilir. <code>dc exec nginx nginx -t</code> .
Live trend "Veri bekleniyor"	WebSocket: backend log'da <code>WS connected</code> görmüyor musun? Nginx Upgrade header'ları + <code>write_blocked</code> default kontrol et. v1.7.11'de fix'lendi.
Login: Invalid credentials	<code>.env</code> 'deki <code>SENTINEL_ADMIN_EMAIL</code> + <code>SENTINEL_ADMIN_PASS</code> birebir mi? Kopyala-yapıştır boşluk olabilir.
Source ekledim ama metric yok	Target DB yetki eksik. <code>dc logs collector \ grep <source-id></code> ile spesifik hata.
Compose pull: rate limit (Docker Hub)	<code>toomanyrequests</code> . 30-60dk bekle veya ACR Cache Rule ile Docker Hub PAT credential ekle.
Demo'da Toplam Instance görünmüyor	InstanceComparison sayfasında 1+ instance seçili olmalı. Sidebar dropdown'dan seç.
traffic-gen container crash loop	Demo DB'leri henüz healthy değil; <code>dc ps</code> ile bekle. ~90 saniye sonra stabilize olur.

Log lokasyonları: - Script log: `/var/log/sentinel-demo-install-*.log` (demo) veya `/var/log/sentinel-install-*.log` (prod) - Container log: `docker logs <container-name>` veya `dc logs <service>`

10. Air-gap / kapalı ağ kurulumu (image bundle)

İnternet erişimi olmayan / kısıtlı ağda kurulum.

Hazırlık (internetli makinede)

```
# 1. ACR'dan tüm image'lar pull
az acr login --name dmcacr22819
for ns in sentineldb360-demo sentineldb360; do
  for svc in backend frontend collector ingestor advisor; do
    docker pull dmcacr22819.azurecr.io/$ns/$svc:v1.7.11
  done
done

# Demo DB'leri (sadece showcase için):
```

```

for db in db-mssql db-postgres db-mysql db-mongo traffic-gen; do
  docker pull dmcacr22819.azurecr.io/sentineldb360-demo/$db:v1.7.11
done

# Mongo + Redis + Nginx (resmi)
docker pull mongo:7.0
docker pull redis:7-alpine
docker pull nginx:1.27-alpine

# 2. Hepsini tarball'a kaydet
docker save \
  dmcacr22819.azurecr.io/sentineldb360-demo/
{backend,frontend,collector,ingestor,advisor,db-mssql,db-postgres,db-mysql,db-
mongo,traffic-gen}:v1.7.11 \
  dmcacr22819.azurecr.io/sentineldb360/
{backend,frontend,collector,ingestor,advisor}:v1.7.11 \
  mongo:7.0 redis:7-alpine nginx:1.27-alpine \
  | gzip -9 > sentineldb360-v1.7.11-bundle.tar.gz

# 3. Compose + nginx + env örnekleri + scriptleri de bundle'a ekle
curl -O https://sentineldb360.com/install/demo.sh
curl -O https://sentineldb360.com/install.sh
curl -O https://sentineldb360.com/install/docker-compose.demo.yml
curl -O https://sentineldb360.com/install/docker-compose.prod.yml
curl -O https://sentineldb360.com/install/nginx-demo.conf
curl -O https://sentineldb360.com/install/nginx-prod.conf
curl -O https://sentineldb360.com/install/.env.demo.example
curl -O https://sentineldb360.com/install/.env.prod.example
curl -O https://sentineldb360.com/install/bootstrap-demo.sh
curl -O https://sentineldb360.com/install/seed-demo-data.py

tar czf sentineldb360-v1.7.11-config.tar.gz \
  demo.sh install.sh docker-compose.*.yml nginx-*.conf .env.*.example bootstrap-demo.sh
seed-demo-data.py

```

Müşteri tarafında (air-gap VM)

```

# Image bundle'ı yükle
zcat sentineldb360-v1.7.11-bundle.tar.gz | sudo docker load

# Config'i unpack et
mkdir -p /opt/sentinel-demo && cd /opt/sentinel-demo # veya sentinel-prod
tar xzf sentineldb360-v1.7.11-config.tar.gz

# Script'i air-gap için flag ile çalıştır (skip-docker, skip-az, skip-curl)
sudo SKIP_DOCKER_INSTALL=yes \
  SKIP_AZ_INSTALL=yes \
  SENTINEL_BASE_URL=file:///opt/sentinel-demo \
  bash demo.sh

```

Not: `SENTINEL_BASE_URL` env'i ile script'in dosya indirme adımları `file://` üzerinden lokal okur. Script `file://` desteklemiyorsa, manuel olarak: ```bash`

Skip download step, direkt başlat:

```
docker compose -f docker-compose.demo.yml --env-file .env.demo up -d ``
```

Destek

- **E-posta:** iletisim@dmcteknoloji.com
- **Doküman:** <https://sentineldb360.com>
- **GitHub:** <https://github.com/dmcteknoloji/dmc-dba-toolkit> (yardımcı DBA scriptleri)

Ek – Script env değişkenleri özet tablosu

Env	Default	Açıklama
<code>SENTINEL_VERSION</code>	<code>v1.7.11</code>	Image tag
<code>SENTINEL_DIR</code>	<code>/opt/sentinel-demo</code> veya <code>/opt/sentineldb360</code>	Kurulum dizini
<code>SENTINEL_BASE_URL</code>	<code>https://sentineldb360.com/install</code>	Compose/nginx/env örneklerinin host'u
<code>SENTINEL_DOMAIN</code>	(boş)	TLS domain (örn. <code>demo.sirket.com.tr</code>)
<code>SENTINEL_MODE</code> (sadece <code>install.sh</code>)	<code>poc</code>	<code>poc</code> (14gün trial) veya <code>production</code> (lisans zorunlu)
<code>SENTINEL_LICENSE_KEY</code> (production'da)	—	DMC'den temin edilen Ed25519 imzalı JWT
<code>SENTINEL_ADMIN_EMAIL</code>	(interaktif sorgu)	Boş Sentinel için ilk admin
<code>SENTINEL_ADMIN_PASS</code>	(interaktif sorgu)	Min 12 karakter
<code>SENTINEL_ACR_USERNAME</code>	—	Service principal client-id (manuel <code>az login</code> yerine)
<code>SENTINEL_ACR_PASSWORD</code>	—	Service principal secret
<code>SKIP_DOCKER_INSTALL</code>	<code>no</code>	Docker zaten kuruluysa atla
<code>SKIP_AZ_INSTALL</code>	<code>no</code>	Azure CLI zaten varsa atla
<code>AZURE_OPENAI_ENDPOINT</code>	—	AI insights için (demo.sh)
<code>AZURE_OPENAI_API_KEY</code>	—	AI insights için (demo.sh)

Env	Default	Açıklama
AZURE_OPENAI_DEPLOYMENT	gpt-4o-mini	AI deployment adı