On-prem ADO → Apache/.NET Kestrel deploy runbook

Reusable pattern from **SlingBuiltAuth** (General project). Use this for the next site pipelines.

SlingBuiltAuth OIDC issuer

- Public OIDC issuer is now `https://auth.slingbuilt.com`.
- Discovery: `https://auth.slingbuilt.com/.well-known/openid-configuration`.
- Admin console remains `https://slingbuilt.com` (apex).
- TLS: Let’s Encrypt cert for `www.slingbuilt.com` also covers `auth.slingbuilt.com`.
- Apache: vhosts on `sfl-web-001` and `sfl-web-004` with `ServerName auth.slingbuilt.com`; proxy to Kestrel on 004 `:5080`; set `X-Forwarded-Host` to `auth.slingbuilt.com`.
- Clients (e.g. `OrganizedInventory`) must use Authority `https://auth.slingbuilt.com`.

Architecture (SlingBuiltAuth reference)

- TLS edge: `sfl-web-001` — Apache + Let's Encrypt; ProxyPass to app host
- App host: `sfl-web-004` — Apache HTTP vhost → Kestrel `127.0.0.1:PORT`
- App: systemd + optional env wrapper; User= service account (often apache)
- CI: Default pool Windows agent (SFL-ADO-001); build + SSH deploy in a single job
- Secrets: ADO Library variable group; mark secrets as secret

Public DNS → edge. LAN DNS must also hit the TLS edge (001), not the app host alone, or HTTPS 404s / wrong certs.

ADO setup checklist

1. Repo + `azure-pipelines.yml` on master
2. Environment (e.g. `slingbuilt-prod`) + approval if needed
3. Variable group linked to the pipeline (e.g. `slingbuilt-prod`)
4. SSH service connection (e.g. `slingbuilt-linux → grok@app-host`)
5. Agent pool: Default (no Microsoft-hosted `ubuntu-latest` on this ADO)
6. Secure files optional — prefer host-side cert copy if agent Node TLS fails against internal CA

Pipeline pattern that works here

- One job: restore → build → publish → SSH deploy (avoid Publish/Download Build Artifacts when agent Node cannot verify the internal CA)
- Write `/etc/<app>.env` on the agent (PowerShell), then upload — avoids bash `set -u` expanding `$` inside passwords
- Install env as `root:<service-group>` mode 640 (e.g. `root:apache`) — 600 root:root crash-loops if the service user cannot read it
- Post-restart: wait/retry health check
- Health check against Kestrel with forwarded headers, example:

```
curl -H="X-Forwarded-Proto: https" -H="Host: public.example.com" http://127.0.0.1:PORT/.well-known/...
```

systemd

- Prefer dedicated `User=`/`Group=`
- `WorkingDirectory` + `ExecStart=/usr/bin/dotnet .../App.dll` OR a small loader script
- Hyphenated env keys (e.g. `Oidc__Clients__organized-inventory__Secret`) are rejected by systemd `EnvironmentFile=` — load the file in a wrapper (Python/shell) then exec dotnet
- `ProtectSystem=strict` needs `ReadWritePaths` for keys/certs/app data; `PrivateTmp=true` is fine if you do not require /tmp state

Apache

Edge (TLS)

- ACME only: `ProxyPass "/.well-known/acme-challenge/" "!"`
- Do NOT blanket-exclude `/.well-known/` — that breaks OpenID discovery and JWKS
- Set `X-Forwarded-Proto`, `X-Forwarded-Port`, `X-Forwarded-Host`

- `ProxyPass / http://app-host/` (or `app-host:port`)

App host (HTTP)

- `ProxyPreserveHost On`
- `ProxyPass / http://127.0.0.1:PORT/`
- Backup forwarded headers if edge already sets them

Secrets / SQL

- Keep DB + client secrets as secret variables; rotate after any screenshot/log exposure
- If several apps share one SQL login (e.g. `web_internet`), rotating the password requires updating every appsettings/env that uses it in the same change window
- Connection strings for .NET on Linux often need `Encrypt=False;TrustServerCertificate=True` (or proper CA trust) against internal SQL

Agent Node TLS (Secure Files / artifacts)

- Symptom: `UNABLE_TO_VERIFY_LEAF_SIGNATURE` / unable to verify the first certificate
- `NODE_EXTRA_CA_CERTS` alone may not be enough — prefer a full chain PEM
- Workaround used for SlingBuiltAuth: skip Secure File download; copy PFX from deploy user home during SSH

First-run verification

1. `systemctl is-active <service>`
2. Loopback discovery/health with forwarded Host/Proto
3. Public `https://<host>/...` through the edge
4. LAN DNS points at TLS edge

SlingBuiltAuth concrete paths

- App root: `/var/www/html/slingbuilt/{app,keys,certs,migrations}`
- Env: `/etc/slingbuilt-auth.env`
- Unit: `slingbuilt-auth → /usr/local/bin/slingbuilt-auth-start`
- Port: `5080`
- Public: `https://slingbuilt.com`
- Canonical repo detail: `DEPLOY.md`

Trouble we hit

- env 640/apache
- hyphenated EnvironmentFile
- ACME well-known exclude
- forwarded headers for OpenIddict
- LAN DNS to 001
- shared SQL login rotation blast radius
- single-job pipeline for Node TLS

Revision #3
Created 10 September 2026 22:10:49 by Steve Ling
Updated 11 September 2026 15:57:30 by Steve Ling