Public Hostname & SSO

Downloadable reference generated from the NetOS Markdown documentation.

Xiber NetOS — Public Hostname & SSO

> Cloudflare Tunnel deployment with Microsoft Entra ID single sign-on.

Overview

NetOS can be exposed publicly through a **Cloudflare Tunnel** and protected with **Microsoft Entra ID** SSO. The auth layer sits in front of the app — unauthenticated users never reach the API or web UI.

```
Internet
  → Cloudflare Tunnel (netos.xiberian.net)
    → OAuth2-Proxy (Entra OIDC)
      → Nginx (route /api → FastAPI, / → Next.js)
        → API + Web containers
```

Target Hostname

```
https://netos.xiberian.net
```

The hostname routes to the `sso-proxy` container, not directly to web or API.

Runtime Components

CONTAINER	ROLE	PORT
`cloudflared`	Outbound tunnel to Cloudflare edge	—
`sso-proxy`	OAuth2-Proxy with Entra OIDC	4180
`public-proxy`	Nginx reverse proxy	8080
`web`	Next.js UI	3000
`api`	FastAPI backend	8000

CONTAINER	ROLE	PORT
`postgres`	PostgreSQL 16	5432
`redis`	Redis 7	6379

Routing Rules (Nginx)

PATH	DESTINATION
`/`	Web app (Next.js)
`/api/`	FastAPI backend
`/docs-api/`	FastAPI Swagger UI
`/openapi.json`	OpenAPI schema

Prerequisites

- Cloudflare account with Zero Trust enabled
- Microsoft Entra ID tenant (Xiber: `4944843d-e347-4fe7-a279-4006ce5efc33`)
- Docker Engine 24+ with Docker Compose v2

Setup Steps

1. Create Entra App Registration

1. Go to **Azure Portal → App registrations → New registration**
2. Name: `Xiber NetOS`
3. Redirect URI (Web): `https://netos.xiberian.net/oauth2/callback`
4. Note the **Application (client) ID** and create a **client secret**

Recommended access policy:

- Assign only Xiber users or a dedicated `NetOS-Users` security group
- Require MFA and compliant device if available
- Map Entra groups to NetOS roles (exec, finance, network_eng, etc.)

2. Create Environment File

Copy the example and fill in real values:

```
cp infra/docker/.env.public.example infra/docker/.env.public
```

Required variables:

VARIABLE	DESCRIPTION	EXAMPLE
`ENTRA_OIDC_ISSUER_URL`	Entra OIDC issuer	`https://login.microsoftonline.com/{tenant_id}/v2.0`
`ENTRA_CLIENT_ID`	App registration client ID	`29833a06-d27e-...`
`ENTRA_CLIENT_SECRET`	App registration client secret	`b-V8Q~bO_YRL...`
`OAUTH2_PROXY_COOKIE_SECRET`	Random 32-byte base64 string	(generate below)
`CLOUDFLARE_TUNNEL_TOKEN`	Tunnel token from Cloudflare dashboard	`eyJhIjoi...`

Generate cookie secret:

```
python3 -c "import base64, secrets;
print(base64.urlsafe_b64encode(secrets.token_bytes(32)).decode())"
```

3. Create Cloudflare Tunnel

1. Go to **Cloudflare Zero Trust → Tunnels → Create a tunnel**
2. Name: `netos`
3. Copy the tunnel token into `.env.public` as `CLOUDFLARE_TUNNEL_TOKEN`
4. Add a public hostname:

   - **Hostname:** `netos.xiberian.net`
   - **Service:** `http://sso-proxy:4180`

1. Optionally add a Cloudflare Access policy (Microsoft Entra ID, Xiber users only) as a second enforcement layer

4. Start the Public Stack

```
cd infra/docker
docker compose --env-file .env.public \
  -f docker-compose.yml \
  -f docker-compose.public.yml \
  --profile public \
  up -d
```

5. Verify

Check containers:

```
docker compose -f docker-compose.yml \
  -f docker-compose.public.yml \
  --profile public \
  ps
```

Open in browser:

```
https://netos.xiberian.net
```

You should be redirected to Microsoft login. After authenticating, you'll see the NetOS UI.

Architecture Notes

Why OAuth2-Proxy + Cloudflare Access?

Both layers serve complementary purposes:

LAYER	PURPOSE
Cloudflare Access	Edge enforcement — blocks unauthenticated traffic before it reaches your server
OAuth2-Proxy	Origin enforcement — protects the app even if tunnel routing is misconfigured

Keep both. The overhead is negligible and the defense-in-depth is worth it.

JWT Validation (Future)

The current API uses a header-based dev auth shim. Production should:

1. Extract the JWT from the OAuth2-Proxy `Authorization` header or cookie
2. Validate signature against Entra JWKS endpoint
3. Extract user email and group claims
4. Map Entra groups to NetOS roles
5. Reject requests with invalid/expired tokens

This is tracked in Roadmap → Authentication.

Security Checklist

ITEM	STATUS
HTTPS via Cloudflare	Automatic with tunnel
Entra OIDC authentication	Via OAuth2-Proxy
Cloudflare Access policy	Recommended additional layer
Cookie secret rotation	Manual — regenerate and restart periodically
Client secret rotation	Via Azure Portal — update `.env.public` after rotation
MFA enforcement	Configure in Entra Conditional Access
Role-based access	Dev shim now, JWT group mapping planned

Troubleshooting

ISSUE	SOLUTION
Redirect loop after login	Check `OAUTH2_PROXY_COOKIE_SECRET` is exactly 32 bytes base64-encoded
502 Bad Gateway	Verify `sso-proxy` and `public-proxy` containers are running
Tunnel not connecting	Check `CLOUDFLARE_TUNNEL_TOKEN` is correct; verify tunnel is active in CF dashboard
CORS errors in browser	Verify Nginx config allows the public hostname as an origin
"Access Denied" after login	User may not be in the assigned Entra group; check app assignment
API returns 401	JWT validation not yet implemented — ensure dev auth headers are passed through proxy