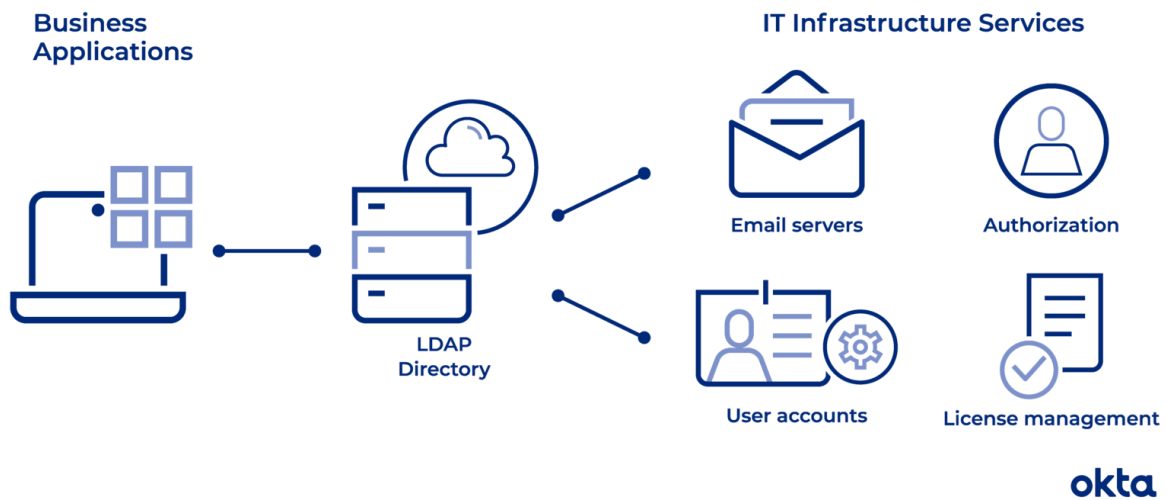# Integrating NiFi with LDAP Hands On

What Is LDAP?

Companies store usernames, passwords, email addresses, printer connections, and other static data within directories. LDAP is an open, vendor-neutral application protocol for accessing and maintaining that data. LDAP can also tackle authentication, so users can sign on just once and access many different files on the server.

LDAP is a protocol, so it doesn't specify how directory programs work. Instead, it's a form of language that allows users to find the information they need very quickly.

How LDAP Works

okta

Integrating NiFi with LDAP Hands On

Step 1: Create LDAP server via docker compose

```
services:
    openldap:
        container_name: nifi-example-openldap
        environment:
            - LDAP_ORGANISATION=nifi-example
            - LDAP_DOMAIN=nifi-example.com
            - LDAP_ADMIN_PASSWORD=SuperSecret
        ports:
            - 389:389
            - 636:636
        image: osixia/openldap
```

Step 2: Create users.ldif

```
dn: cn=admin,dc=nifi-example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword: SuperSecret

dn: cn=john,dc=nifi-example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: john
description: Regular user
userPassword: password123
```

Step 3: Add users to LDAP

```
ldapadd -x -H ldap://localhost:389 -D cn=admin,dc=nifi-example,d
```

Step 4: Generate TLS for nifi

```
./tls-toolkit.sh standalone -n 'nifi-example.com' -o target-nif
```

Step 5: Use generated properties file and modify parameters

```
nifi.security.user.authorizer=managed-authorizer
nifi.security.allow.anonymous.authentication=false
nifi.security.user.login.identity.provider=ldap-provider
```

Step 6: modify ldap-provider in `login-identity-providers.xml`

```xml
<provider>
    <identifier>ldap-provider</identifier>
    <class>org.apache.nifi.ldap.LdapProvider</class>
    <property name="Authentication Strategy">SIMPLE</propert

    <property name="Manager DN">cn=admin,dc=nifi-example,dc=
    <property name="Manager Password">SuperSecret</property>

    <property name="TLS - Keystore"></property>
    <property name="TLS - Keystore Password"></property>
    <property name="TLS - Keystore Type"></property>
    <property name="TLS - Truststore"></property>
    <property name="TLS - Truststore Password"></property>
    <property name="TLS - Truststore Type"></property>
    <property name="TLS - Client Auth"></property>
    <property name="TLS - Protocol"></property>
    <property name="TLS - Shutdown Gracefully"></property>

    <property name="Referral Strategy">FOLLOW</property>
    <property name="Connect Timeout">10 secs</property>
    <property name="Read Timeout">10 secs</property>

    <property name="Url">ldap://nifi-example.com:389</proper
    <property name="User Search Base">dc=nifi-example,dc=com
    <property name="User Search Filter">cn={0}</property>
```

```
        <property name="Identity Strategy">USE_DN</property>
        <property name="Authentication Expiration">12 hours</pr
    </provider>
```

Step 6: Modify `authorizers.xml` file-provider

```
<authorizer>
    <identifier>file-provider</identifier>
    <class>org.apache.nifi.authorization.FileAuthorizer</class>
    <property name="Authorizations File">./conf/authorizations.x
    <property name="Users File">./conf/users.xml</property>
    <property name="Initial Admin Identity">cn=admin,dc=vanducng
    <property name="Legacy Authorized Users File"></property>

    <property name="Node Identity 1">CN=localhost, OU=NIFI</prop
</authorizer>
```

Sources

Setup LDAP for Apache Nifi

Default Apache Nifi installation comes without security layer
which exposes the development UI. As a result, users can
freely access the Nifi project development with knowledge

 https://vanducng.dev/2019/11/20/Setup-LDAP-for-Apache-
Nifi/