opentext™

OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド

2025 年 1 月

保証と著作権

Copyright 2009 - 2024 Open Text All rights reserved.

Open Text、関連会社、およびライセンサ (「Open Text」) の製品およびサービスに対する保証は、当該製品およびサービスに付属する保証書に明示的に規定されたものに限られます。本書のいかなる内容も、当該保証に新たに保証を追加するものではありません。Open Text は、本書に技術的または編集上の誤りまたは不備があっても責任を負わないものとします。本書の内容は、将来予告なしに変更されることがあります。

目次

本書およびライブラリについて 9

1 OpenText eDirectory の 24.4 (v9.3) への移行 11

OpenText eDirectory を移行するための前提条件 . . . 11
OpenSSL 3.0 を採用後の OpenText eDirectory の重要な変更点 . . . 12
OpenText eDirectory の 9.3 への移行 . . . 14

2 インストールとアップグレードの機能 17

OpenText eDirectory 9.3 のインストール用に複数のパッケージ形式を用意 . . . 18
任意の場所に OpenText eDirectory 9.3 をインストールする . . . 18
アプリケーションファイルに任意の場所を指定する . . . 19
データファイルに任意の場所を指定する . . . 19
環境設定ファイルに任意の場所を指定する . . . 20
非ルートユーザによるインストール . . . 20
標準の準拠 . . . 21
FHS の準拠 . . . 21
LSB の準拠 . . . 22
サーバのヘルスチェック . . . 22
ヘルスチェックの必要性 . . . 23
サーバが正常であることの確認基準 . . . 23
ヘルスチェックを実行する . . . 23
ヘルスチェックのタイプ . . . 24
状態のカテゴリ . . . 25
ログファイル . . . 26
EBA 対応サーバへのブリッジパッチのインストール . . . 27
Linux で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール . . . 27
Windows で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール . . . 28
コンテナ内で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール . . . 29
その他の情報 . . . 30

3 OpenText eDirectory CE 24.4 (v9.3) の Linux へのインストールまたはアップグレード 31

システム要件 . . . 31
前提条件 . . . 33
ハードウェア要件 . . . 38
バックリンク処理の強制実行 . . . 38
edirAdmin ユーザグループの追加 . . . 39
OpenText eDirectory をアップグレードする . . . 40
サーバのヘルスチェック . . . 41
OES 以外の Linux サーバでのアップグレード . . . 41
Linux 上での OpenText eDirectory の無人アップグレード . . . 42
tarball デプロイメントの eDirectory 9.3 をアップグレードする . . . 43
非ルートユーザによる OpenText eDirectory 9.3 へのアップグレード . . . 44

複数インスタンスをアップグレードする . . . 45
OpenText eDirectory のインストール . . . 46
OpenText eDirectory での SLP の使用 . . . 47
nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする . . . 48
非ルートユーザによる OpenText eDirectory 9.3 のインストール . . . 51
ルートユーザがインストールした OpenText eDirectory の非ルートユーザによる設定 . . . 55
dsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する . . . 56
ndsconfig の使用による OpenText eDirectory 9.3 の複数インスタンスの設定 . . . 63
ndsconfig を使用してコンテナ名にドットを使用したツリーに Linux サーバをインストールする . . . 71
nmasinst ユーティリティを使用して NMAS を設定する . . . 72
非ルートユーザの SNMP 設定 . . . 73
ログファイルの場所 . . . 73

4 OpenText eDirectory CE 24.4 (v9.3) の Windows へのインストールまたはアップグレード 75

システム要件 . . . 75
前提条件 . . . 76
ハードウェア要件 . . . 79
バックリンク処理の強制実行 . . . 80
OpenText eDirectory の Windows へのインストール . . . 80
Windows サーバで OpenText eDirectory 9.3 をインストールまたは更新する . . . 81
サーバのヘルスチェック . . . 83
LDAP を介した OpenText eDirectory との通信 . . . 83
NMAS サーバソフトウェアのインストール . . . 85
コンテナ名にドットを使用したツリーへのインストール . . . 85
Windows での OpenText eDirectory 9.3 の無人インストールと設定 . . . 86
ログファイルの場所 . . . 93
Windows での OpenText eDirectory のアップグレード . . . 94
OpenText eDirectory をアップグレードする前の NMAS スキーマ拡張 . . . 94
Windows インストーラを使用して OpenText eDirectory をアップグレードする . . . 95
Windows での OpenText eDirectory の無人アップグレード . . . 96

5 Microsoft Azure での OpenText eDirectory のデプロイ 97

前提条件 . . . 97
デプロイメントの手順 . . . 97
Linux プラットフォームへのデプロイ . . . 97
Windows プラットフォームへのデプロイ . . . 113
Microsoft Azure コンテナインスタンスへの OpenText eDirectory コンテナのデプロイ . . . 122
コンテナのデプロイのチェックリスト . . . 122

6 Amazon Web Services EC2 での OpenText eDirectory のデプロイ 131

前提条件 . . . 131
デプロイメントの手順 . . . 131
AWS 仮想プライベートクラウドの準備 . . . 133
ネットワーク ACL の設定 . . . 134
セキュリティグループの設定 . . . 137
SSH キーペアを作成する . . . 138

インスタンスの作成とデプロイ . . . 138
OpenText eDirectory データの保存用に EBS ボリュームを設定する . . . 139
OpenText eDirectory および OpenText Identity Console のインストール . . . 140
監査サービスのデプロイ . . . 144
障害復旧 . . . 144
AWS での OpenText eDirectory デプロイメントのサイジングガイダンス . . . 146
AWS EC2 インスタンスの選択を開始する . . . 146
パフォーマンステスト環境の詳細 . . . 147
サイジング要件の決定 . . . 149

7 Docker コンテナを使用した OpenText eDirectory のデプロイ 157

Docker の利点 . . . 157
Docker コンテナを使用して OpenText eDirectory をデプロイするための計画 . . . 157
システム要件 . . . 158
前提条件 . . . 158
Docker CLI . . . 158
OpenText eDirectory コンテナのデプロイ . . . 158
ホストネットワークでの OpenText eDirectory コンテナのデプロイ . . . 161
ユーザ定義のオーバーレイネットワークでの OpenText eDirectory コンテナのデプロイ . . . 163
デプロイメント後のタスク . . . 165
実行中の OpenText eDirectory コンテナでのコマンドの実行 . . . 165
OpenSLP for OpenText eDirectory Docker コンテナの設定 . . . 166
OpenText eDirectory Docker コンテナへの NMAS メソッドのインストール . . . 166
OpenText eDirectory Docker コンテナへの新しいパッケージのインストール . . . 167
OpenText eDirectory Docker コンテナでの CEF プロパティファイルの設定 . . . 167
OpenText eDirectory データストレージの管理 . . . 168
Docker コンテナを使用した OpenText eDirectory のアップグレード . . . 169
OpenText eDirectory Docker コンテナの復旧 . . . 169

8 IPv6 アドレスを使用する Linux と Windows での OpenText eDirectory のインストール 171

IPv6 を使用する Linux での OpenText eDirectory の設定 . . . 172
新しい OpenText eDirectory ツリーを作成する . . . 172
既存の OpenText eDirectory ツリーへのサーバの追加 . . . 172
既存のもしくはアップグレードした OpenText eDirectory サーバでの IPv6 アドレスの有効化 . . . 173
IPV6 の LDAP URL の LDAP サーバオブジェクトへの追加 . . . 173
IPv6 を使用する Windows での OpenText eDirectory のインストールもしくはアップグレード 174
OpenText eDirectory のインストール中またはアップグレード中に行う IPv6 の有効化 . 174
既存のサーバ上での IPv6 の有効化 . . . 174
OpenText iMonitor へのアクセス . . . 174

9 FIPS モードでの OpenText eDirectory の運用 175

FIPS モードでの OpenText eDirectory の設定 . . . 175

10 DIB の移動 177

Linux . . . 177
Windows . . . 178

11 OpenText eDirectory 9.3 のアップグレード要件 179

8.8 SP1 以降のバージョンのリファレンスに関する変更点 . . . 179
9.3 のアップグレードプロセス . . . 180

12 Linux 上での OpenText eDirectory の設定 181

環境設定ユーティリティ . . . 181
ndsconfig ユーティリティ . . . 181
LDAP ツールを使用して LDAP サーバと LDAP グループオブジェクトを背呈する . . . 182
nmasinst ユーティリティを使用して NetIQ Modular Authentication Service を設定する . 182
OpenText eDirectory のカスタマイズ . . . 182
環境設定パラメータ . . . 184
セキュリティ上の考慮事項 . . . 190

13 高可用性クラスタで OpenText eDirectory を展開する 191

Linux での OpenText eDirectory サービスのクラスタリング . . . 192
前提条件 . . . 192
OpenText eDirectory をインストールして設定する . . . 192
クラスタ化した Linux 環境で SNMP サーバを設定する . . . 195
Windows での OpenText eDirectory サービスのクラスタリング . . . 195
前提条件 . . . 196
OpenText eDirectory をインストールして設定する . . . 196
クラスタ化した Windows 環境で SNMP サーバを設定する . . . 198
クラスタ化環境のトラブルシューティング . . . 198
クラスタ化ノードの OpenText eDirectory を修復またはアップグレードする . . . 198
Windows レジストリキーの作成 . . . 199
環境設定ユーティリティのオプション . . . 199

14 OpenText eDirectory のアンインストール 201

Windows での OpenText eDirectory のアンインストール . . . 201
OpenText eDirectory、ConsoleOne、および SLP DA のアンインストール . . . 201
OpenText eDirectory の無人アンインストール . . . 202
NICI のアンインストール . . . 206
Microsoft Visual C++ 2005 と Visual C++
2012 のランタイムライブラリのアンインストール . . . 206
Linux 上での OpenText eDirectory のアンインストール . . . 206
Linux 上での OpenText eDirectory の無人アンインストール . . . 207
OpenText eDirectory のアンインストールに関する注意 . . . 208

A OpenText eDirectory 用の Linux パッケージ 209

B OpenText eDirectory ヘルスチェック 213

ヘルスチェックの必要性 . . . 213
ヘルスチェックの実行 . . . 213
アップグレードと同時に実行 . . . 213
スタンドアロンユーティリティとして実行 . . . 214
ヘルスチェックのタイプ . . . 214
基本的なサーバの状態 . . . 215
パーティションとレプリカの状態 . . . 215

状態のカテゴリ . . . 215
正常 . . . 215
警告 . . . 216
重大 . . . 216
ログファイル . . . 216

C OpenSLP for OpenText eDirectory の設定 219

Service Location Protocol . . . 219
SLP の基本 . . . 219
OpenText Service Location Providers . . . 220
ユーザエージェント . . . 221
サービスエージェント . . . 222
環境設定パラメータ . . . 222

D 問題のトラブルシューティング 225

インストール問題のトラブルシューティング . . . 225
設定問題のトラブルシューティング . . . 227
OpenText eDirectory の複数インスタンス問題のトラブルシューティング . . . 230
ndsconfig ユーティリティ . . . 231
NMAS インストールのトラブルシューティング . . . 232
証明書サーバのインストールのトラブルシューティング . . . 232
インストールのトラブルシューティング . . . 234

本書およびライブラリについて

このインストールガイドには、OpenText™ eDirectory 9.3 のインストール方法が記載されています。本書の対象読者はネットワーク管理者です。

『*OpenText eDirectory CE 24.4 (v9.3) インストールガイド*』の最新版については、OpenText™ eDirectory オンラインヘルプ (https://www.netiq.com/documentation/edirectory-93/) の Web サイトを参照してください。

本書の読者

このガイドはネットワーク管理者を対象としています。

ライブラリに含まれているその他の情報

ライブラリには次の情報リソースが含まれています。

管理ガイド

OpenText eDirectory の管理および設定方法について説明します。

Linux プラットフォーム用チューニングガイド

Linux プラットフォーム上の OpenText eDirectory を分析し、すべての展開で優れたパフォーマンスが実現されるように調整する方法について説明します。

これらのガイドは、OpenText™ eDirectory CE 24.4 (v9.3) (https://www.netiq.com/documentation/edirectory-93/) で入手できます。

OpenText eDirectory 管理ユーティリティの詳細については、『OpenText™ Identity Console 管理ガイド』を参照してください。

1 OpenText eDirectory の 24.4 (v9.3) への移行

OpenText eDirectory 9.3 は OpenSSL 3.0 をサポートしています。既存の OpenText eDirectory セットアップをバージョン 9.3 に移行するには、移行手順を実行する必要があります。このセクションでは、OpenText eDirectory を 9.2 から 9.3 に移行する手順について概説します。

- 11 ページの 「OpenText eDirectory を移行するための前提条件」
- 12 ページの 「OpenSSL 3.0 を採用後の OpenText eDirectory の重要な変更点」
- 14 ページの 「OpenText eDirectory の 9.3 への移行」

OpenText eDirectory を移行するための前提条件

マイグレーションを開始する前に、次の作業を行います。

- ❒ **バックアップ :** 環境設定ファイルやデータを含む、OpenText eDirectory インスタンスの完全なバックアップを実行します。
- ❒ **システム互換性 :** OpenSSL 3.0 は、SLES 15 SP3、SP4、SP5、SP6、および RHEL 8.8、8.9、および 8.10、および RHEL 9.2、9.3、および 9.4 でのみサポートされています。ご使用のプラットフォームが、この要件を満たしていることを確認します。
- ❒ **パッチ :** OpenText eDirectory の既存のインストールに最新のパッチを適用して、バージョン 9.2 以降にします。
- ❒ **Bridge Patch:** マルチサーバ環境を使用していて、サーバのどれかまたはすべてのサーバで EBA が有効になっている場合、OpenText eDirectory 9.3.0 にアップグレードする前に、**すべての OpenText eDirectory サーバを最新の OpenText eDirectory 9.2.8 または 9.2.9 バージョンにアップグレードする必要があります**。アップグレードが成功した後、ユーザは EBA が有効になっているすべてのレプリカサーバに EBA ブリッジパッチを適用する必要があります。EBA ブリッジパッチは、次のパスにある OpenText eDirectory ビルドで利用できます。
  - **Linux の場合 :**

    `<Untarred_edirectory_location>/eDirectory/eba_bridge_patch`
  - **Windows の場合 :**

    `<extracted_patch>eDirectory\eba_bridge_patch`
  - **コンテナの場合 :**

    `<Untarred_edirectory_location>/eDirectory_930/eba_bridge_patch`

次のコマンドを実行して、EBA がシステムで有効になっているか確認します。

```
ndscheck -a  <admin_credentials> -w <tree_login_password>
```

詳細については、27 ページの 「Linux で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」、28 ページの 「Windows で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」、および 29 ページの 「コンテナ内で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」を参照してください。

- ❒ **同じボックスの互換性:** 古いバージョンのIdentity ConsoleがOpenText eDirectoryと同じシステムにインストールされている場合は、Identity Console をバージョン 1.9 以降にアップグレードします。アップグレード中に、同じマシンに古いバージョンの Identity Console が検出されると、OpenText eDirectory のインストールは終了します。

注

- OpenText eDirectory インストーラは、インストール時に、EBA が有効になっているすべてのレプリカサーバにブリッジパッチが適用されているか確認し、いずれかのレプリカサーバに適用されていない場合は終了します。非ルートユーザおよびコンテナでは、EBA チェックは機能しません。
- OpenText eDirectory 9.3.0 以降では、マスタサーバが 9.3.0 の場合、EBA の有無にかかわらず、下位バージョンの OpenText eDirectory をレプリカとして追加することはサポートされません。
- Linux で、OpenText eDirectory 9.2.9 または 9.2.8 が、OpenText Identity Console 1.8 と共存している場合は、OpenText eDirectory を 9.3 にアップグレードする前に、OpenText Identity Console を 1.9 にアップグレードすることをお勧めします。

OpenSSL 3.0 を採用後の OpenText eDirectory の重要な変更点

マイグレーションを開始する前に理解する必要がある重要な変更点を次に示します。

TLS プロトコルと暗号スイート

- この製品は TLS 1.2 と共に TLS 1.3 をサポートします。
- 暗号スイートは OpenSSL 3.0 に合わせて更新され、次の TLS 1.3 暗号スイートが含まれます。
  - TLS_AES_256_GCM_SHA384。
  - TLS_CHACHA20_POLY1305_SHA256 (非 FIPS のみ)。
  - TLS_AES_128_GCM_SHA256。
- TLS 1.2 暗号は、AES 暗号化オプションと RSA 暗号化オプションの組み合わせで続行されます。

重要 : TLS 1.3 を必要とする環境を適切に設定してください。非推奨または古くなった暗号スイートは、必ず削除または更新してください。

FIPS および NICI の設定

- OpenText eDirectory 9.3 では、デフォルトで FIPS モードは無効になっています。ただし、NICI (Novell International Cryptographic Infrastructure) の設定に従って有効にすることができます。これは、TLS FIPS および NICI FIPS に適用されます。以前のバージョンの製品では、TLS FIPS はデフォルトで有効になっています。
- NICI 環境設定ファイル nici64.cfg は、TLS FIPS および NICI FIPS を有効にするリファレンスポイントになります。詳細については、「FIPS モードでの eDirectory の運用 (https://www.netiq.com/documentation/edirectory-93/edir_install/data/b1gnsxb4.html)」を参照してください。

重要 : FIPS 準拠が必要なシステムの場合は、NICI 設定を確認してください。9.2 からアップグレードする場合は、TLS-FIPS 設定でのデフォルト動作が異なることが予想されます。

RSA 証明書と暗号化アルゴリズム

OpenText eDirectory 9.3 では、次の機能がサポートされなくなりました。

- FIPS モードで 512 ビット、768 ビット、または 1024 ビットのキーサイズを持つ RSA 証明書。
- OpenText eDirectory 9.3 での FIPS モードの SHA-1 証明書。ただし、これらは非 FIPS 設定ではサポートされています。
- FIPS 環境または非 FIPS 環境での DES と RC2-40。ただし、3DES は引き続き FIPS モードでサポートされています。

重要 : 特に、小さい鍵サイズまたは SHA-1 証明書を使用する場合は、証明書と暗号化の設定を確認し、更新してください。

NICI ツリーキーの動作

OpenText eDirectory 9.3 を新規インストールすると、AES 256 ビットツリーキー (CN=W1.CN=KAP.CN=Security) が自動的に作成されます。ただし、古いバージョンの OpenText eDirectory からアップグレードする場合、既存のツリーキーは変更されず、引き

続き想定どおりに機能します。NICI キーの詳細については、「NICI キーの概要 (Understanding the NICI Keys) (https://www.netiq.com/documentation/edirectory-9/nici_admin_guide/data/agtbw4b.html)」を参照してください。

後方互換性と統合

OpenText eDirectory 9.3 は、これらの製品の次のバージョンのみをサポートしています。古いバージョンを使用している場合は、アップグレードしてください。

- OpenText Identity Console 24.4 (v1.9)
- OpenText Identity Manager 24.4 (v4.10)

重要: アップグレードする前に、これらの製品との統合でバージョンの互換性を確認してください。古いバージョンでは、インストールの失敗や通信の問題が発生する可能性があります。

OpenText eDirectory の 9.3 への移行

製品を 9.3 に移行するために、関連コンポーネントをアップグレードします。

1 OpenText Identity Console の互換性 :

OpenText eDirectory 9.3 は、次のバージョンの OpenText Identity Console および OpenText Identity Manager のみをサポートしています。

- OpenText Identity Console 24.4 (v1.9)
- OpenText Identity Manager 24.4 (v4.10)

これらの製品のサポートされていないバージョンがインストールされている場合は、それらをサポートされているバージョンにアップグレードします。

2 アップグレード後の NMAS メソッドの処理

OpenText eDirectory 9.3.0 は、NMAS メソッドを OpenSSL 3.0.15 上に構築された最新の NICI と統合します。

- **単一サーバ環境** :

  アクションは必要ありません : NMAS メソッドを備えた OpenText eDirectory 9.3.0 のインスタンスが 1 つである場合、Linux では追加の手順は必要ありません。Windows の場合は、NMAS スキーマを拡張する必要があります。詳細については、94 ページの 「OpenText eDirectory をアップグレードする前の NMAS スキーマ拡張」を参照してください。

- **マルチサーバ環境** : バージョンが混在するマルチサーバセットアップのサーバがある場合、OpenText eDirectory 9.3.0 に新しい NMAS メソッドを追加しても影響を受けません。ただし、レプリカとして 9.2.9 以下のバージョンの OpenText eDirectory を使用している場合は、OpenText eDirectory 9.2.9 に NMAS メソッドを追加してから、9.3.0 にも追加します。

注 :

- **手動による手順は不要 :** すべてのサーバを OpenText eDirectory 9.3.0 にアップグレードした後、NMAS メソッドを手動で追加する作業は必要なくなりました。
- NMAS メソッドの詳細については、『OpenText™ eDirectory CE 24.4 (v9.3) eDirectory 管理ガイド (https://www.netiq.com/documentation/edirectory-9/edir_admin/data/a53vifw.html)』を参照してください。

2 インストールとアップグレードの機能

この章では、OpenText eDirectory CE 24.4 (v9.3) のインストールとアップグレードの機能について説明します。

次の表に、新機能とその新機能がサポートされるプラットフォームについて示します。

機能	Linux	Windows
OpenText eDirectory 9.3 のインストール用に複数のパッケージ形式を用意	✓	✗
任意の場所へのアプリケーションファイルのインストール	✓	✓
任意の場所へのデータファイルのインストール	✓	✓
任意の場所への環境設定ファイルのインストール	✓	✗
非ルートユーザによるインストール	✓	✗
高可用性クラスタへのインストールに対する強化されたサポート	✓	✓
FHS の準拠	✓	✗
LSB の準拠	✓	✗
サーバのヘルスチェック	✓	✓
SecretStore の統合	✓	✓
OpenText eDirectory Instrumentation インストール	✓	✓

注

- OpenText eDirectory CE 24.4 (v9.3) 9.2.3 以降では、AES 256 ビットツリーキーがデフォルトで有効になっています。ただし、9.2.x からアップグレードする場合は、OpenText eDirecotry をアップグレードした後、W1.KAP.Security オブジェクトを手動で追加する必要があります。
- EBA 対応セットアップで、サーバのいずれかを EBA を備えた OpenText eDirectory CE 24.4 (v9.3) にアップグレードする場合は、すべてのレプリカサーバが OpenText eDirectory 9.2.8 または 9.2.9 上にインストールされている必要があります。レプリカを 9.3.0 にアップグレードする前に、ブリッジパッチを適用してください。ブリッジパッチは、OpenText eDirectory 9.2.7 以前のバージョンには適用できません。

このセクションでは、次の情報について説明します。

- 18 ページの 「OpenText eDirectory 9.3 のインストール用に複数のパッケージ形式を用意」
- 18 ページの 「任意の場所に OpenText eDirectory 9.3 をインストールする」
- 20 ページの 「非ルートユーザによるインストール」
- 21 ページの 「標準の準拠」
- 22 ページの 「サーバのヘルスチェック」
- 27 ページの 「EBA 対応サーバへのブリッジパッチのインストール」
- 30 ページの 「その他の情報」

OpenText eDirectory 9.3 のインストール用に複数のパッケージ形式を用意

Linux では、OpenText eDirectory 9.3 をホストにインストールするときにさまざまなファイル形式から選択するオプションが用意されています。選択できるファイル形式を次の表に示します。

ユーザのタイプとインストール場所	Linux
ルートユーザ	
デフォルトの場所	RPM
任意の場所	tarball
非ルートユーザ	
任意の場所	tarball

tarball を使用したインストールの詳細については、43 ページの 「tarball デプロイメントの eDirectory 9.3 をアップグレードする」を参照してください。

任意の場所に OpenText eDirectory 9.3 をインストールする

OpenText eDirectory 9.3 では、アプリケーション、データ、および環境設定ファイルをインストールする場所を自由に選択できます。

OpenText eDirectory 9.3 を任意の場所にインストールするシナリオの 1 つは、ホストに以前のバージョンの OpenText eDirectory がインストールされており、それをアップグレードする前に OpenText eDirectory 9.3 をテストする場合です。このようにすると、既存の OpenText eDirectory 設定を変更しないで済み、この新しいバージョンをテストすることもできます。その後で、既存のバージョンを保持するか、OpenText eDirectory 9.3 にアップグレードするかを決定できます。

注 : SLP と SNMP サブエージェントはデフォルトの場所にインストールされます。

このセクションでは、任意の場所にさまざまなファイルをインストールする方法について説明します。

- 19 ページの 「アプリケーションファイルに任意の場所を指定する」
- 19 ページの 「データファイルに任意の場所を指定する」
- 20 ページの 「環境設定ファイルに任意の場所を指定する」

アプリケーションファイルに任意の場所を指定する

OpenText eDirectory のインストール中に、選択した場所にアプリケーションファイルをインストールできます。

Linux

OpenText eDirectory 9.3 を任意の場所にインストールする場合は、tarball インストールファイルを使用して、選択した場所に OpenText eDirectory 9.3 を untar することができます。

Windows

OpenText eDirectory 9.3 より前でも、インストールウィザードの実行中にアプリケーションファイルに任意の場所を指定することができました。

データファイルに任意の場所を指定する

OpenText eDirectory の設定中に、選択した場所にデータファイルを保存できます。データファイルには、data、dib、および log ディレクトリが含まれます。

Linux

任意の場所でデータファイルを設定する場合、ndsconfig ユーティリティの -d または -D オプションのいずれかを使用できます。

オプション	説明
-d *任意の場所*	指定したパスに DIB (OpenText eDirectory データベース) ディレクトリを作成します。
-D *任意の場所*	data (pid やソケット ID などのデータが入る)、dib、および log ディレクトリを、指定したパスに作成します。

Windows

Windows では、インストール中に DIB パスを入力するように指示されます。選択するパスを入力してください。

環境設定ファイルに任意の場所を指定する

OpenText eDirectory の設定中には、環境設定ファイルの保存先にするパスを選択できます。

Linux

nds.conf 環境設定ファイルを異なる場所に設定するには、ndsconfig ユーティリティの --config-file オプションを使用します。

その他の環境設定ファイル (modules.conf、ndsimon.conf、および ice.conf など) を異なる場所にインストールするには、次の操作を実行します。

1. すべての環境設定ファイルを新しい場所にコピーします。
2. 次のように入力して新しい場所を設定します。

   ndsconfig set n4u.nds.configdir *任意の場所*

Windows

Windows では、環境設定ファイルに任意の場所を指定することはできません。

非ルートユーザによるインストール

OpenText eDirectory 9.1 以上では、非ルートユーザによる OpenText eDirectory サーバのインストールと設定がサポートされています。OpenText eDirectory の以前のバージョンでは、ホストで実行される OpenText eDirectory のシングルインスタンスによってのみ、ルートユーザだけがインストールと設定を行うことができました。

eDirectory 9.3 以上では、任意の非ルートユーザが tarball ビルドを使って OpenText eDirectory をインストールできます。同一または異なるユーザによる OpenText eDirectory のバイナリインストールの複数インスタンスが存在できます。ただし、ルート以外のユーザのインストールに対しても、Novell International Cryptographic Infrastructure (NICI)、SNMP、および SLP などのシステムレベルのサービスはルート権限によってのみインストールが可能です。OpenText eDirectory の機能のために、NICI は必須のコンポーネントで、SNMP と SLP はオプションのコンポーネントです。また、パッケージのインストールについては、シングルインスタンスのみがルートユーザによってインストール可能です。

インストール後に、非ルートユーザは個々の tarball インストールやバイナリインストールを用いて、OpenText eDirectory サーバインスタンスの設定をできます。つまり、1 つのホストで OpenText eDirectory サーバの複数のインスタンスを実行できます。なぜなら、ルートユーザも非ルートユーザも、パッケージや tarball インストールを用いることで、異なる OpenText eDirectory サーバインスタンスを 1 つのホスト上で設定できるからです。複数インスタンスの機能の詳細については、45 ページの 「複数インスタンスをアップグレードする」を参照してください。

非ルートユーザによるインストールと設定は、Linux プラットフォームでのみ適用可能です。非ルートユーザによるインストールや設定に関する詳細については、51 ページの 「非ルートユーザによる OpenText eDirectory 9.3 のインストール」を参照してください。

標準の準拠

OpenText eDirectory 9.3 は次の標準に準拠しています。

- 21 ページの 「FHS の準拠」
- 22 ページの 「LSB の準拠」

FHS の準拠

他製品のアプリケーションファイルとの間でファイルの衝突を回避するため、OpenText eDirectory 9.3 は FHS (File system Hierarchy Standard) に従っています。この機能は、Linux のみで使用できます。

OpenText eDirectory がこのディレクトリ構造に従うのは、デフォルトの場所にインストールすることを選択した場合のみです。任意の場所を選択した場合、ディレクトリ構造は、*任意の場所 / デフォルトの場所*になります。

たとえば、eDir93 ディレクトリにインストールすることを選択した場合、eDir93 ディレクトリ内は同じディレクトリ構造になり、マニュアルページは、/eDir93/opt/novell/man ディレクトリにインストールされます。

次の表に、ディレクトリ構造の変更を示します。

ディレクトリに保存されるファイルのタイプ	ディレクトリの名前とパス
実行ファイルのバイナリとスタティックシェルスクリプト	/opt/novell/eDirectory/bin
ルートが使用する実行ファイルのバイナリ	/opt/novell/eDirectory/sbin
スタティックライブラリまたはダイナミックライブラリのバイナリ	/opt/novell/eDirectory/lib
環境設定ファイル	/etc/opt/novell/eDirectory/conf
読み書きを行う実行時のダイナミックデータ (DIB など)	/var/opt/novell/eDirectory/data
ログファイル	/var/opt/novell/eDirectory /log
Linux マニュアルページ	/opt/novell/man

環境変数のエクスポート

OpenText eDirectory 9.3 で FHS 実装を使用する場合は、パスの環境変数を更新してエクスポートする必要があります。これによって次の問題が生じます。

- エクスポートするすべてのパスを覚えておく必要があります。シェルを開くときには常に、これらのパスをエクスポートしてからユーティリティの使用を開始する必要があります。
- バイナリのセットを複数使用する場合は、複数のシェルを開くか、または設定を解除して異なるバイナリのセットへのパスを頻繁に設定する必要があります。

この問題を解決するため、/opt/novell/eDirectory/bin/ndspath スクリプトを次のように使用することができます。

- 次のとおり、ndspath スクリプトをユーティリティの前に指定して、ユーティリティを実行します。

  ```
  custom_location/opt/novell/eDirectory/bin/ndspath
  utility_name_with_parameters
  ```

- 次のとおり、現在のシェル内のパスをエクスポートします。

  ```
  . custom_location/opt/novell/eDirectory/bin/ndspath
  ```

- このコマンドの入力後、通常どおりにユーティリティを実行します。プロファイル内のスクリプト (bashrc、または同様のスクリプト) を呼び出します。こうすることで、ログインするか新しいシェルを開けば、直接ユーティリティを使い始めることができます。

LSB の準拠

OpenText eDirectory 9.3 は LSB (Linux Standard Base) に準拠するようになりました。LSB では、FHS に準拠することも推奨されています。Linux の OpenText eDirectory パッケージにはすべて、*novell* というプレフィックスが付けられています。たとえば、NDSserv の名前は novell-NDSserv になっています。

サーバのヘルスチェック

OpenText eDirectory CE 24.4 (v9.3) には、アップグレード前にサーバヘルスが安全な状態であるかどうかを判断するのに役立つ、サーバヘルスチェック機能が組み込まれています。

サーバのヘルスチェックは、どのアップグレードでもデフォルトで実行され、パッケージが実際にアップグレードされる前に行われます。ただし、診断ツールの ndscheck を実行してヘルスチェックを行うこともできます。

定期的なヘルスチェック手順の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*OpenText eDirectory のメンテナンス*」を参照してください。

ヘルスチェックの必要性

OpenText eDirectory の以前のリリースでは、アップグレードを進める前にサーバの状態はチェックされませんでした。状態が不安定な場合は、アップグレード処理に失敗し、OpenText eDirectory は不整合な状態になっていました。場合によっては、アップグレード前の設定に戻すことができない場合もあります。

新しいヘルスチェックツールによってこの問題が解決され、サーバをアップグレードする準備を確実に整えることができます。

サーバが正常であることの確認基準

サーバヘルスチェックのユーティリティは、ツリーが正常に機能していることを確認するため、所定のヘルスチェックを実行します。これらのヘルスチェックがすべて正しく完了すると、ツリーは正常に機能していると見なされます。

ヘルスチェックを実行する

サーバのヘルスチェックは次の 2 種類の方法で実行できます。

- 23 ページの 「アップグレードと同時に実行」
- 24 ページの 「スタンドアロンユーティリティとして実行」

注 : ヘルスチェックユーティリティを実行するには、管理者の権利を持っている必要があります。ユーティリティを実行するために設定できる最小限の権利はパブリックの権利です。ただし、パブリックの権利では、NetWare コアプロトコル (NCP) オブジェクトの一部とパーティション情報が利用できません。

アップグレードと同時に実行

OpenText eDirectory をアップグレードするときは常に、デフォルトでヘルスチェックが実行されます。

Linux

アップグレード時には常にデフォルトで、実際のアップグレード処理が開始される前にヘルスチェックが実行されます。

デフォルトのヘルスチェックを省略するため、nds-install ユーティリティで「-j」オプションを使用することができます。

Windows

サーバのヘルスチェックは、インストールウィザードの一部として行われます。ヘルスチェックは、プロンプトが表示されたときに有効または無効にすることができます。

スタンドアロンユーティリティとして実行

サーバのヘルスチェックは、いつでもスタンドアロンユーティリティとして実行できます。次の表では、ヘルスチェックユーティリティについて説明します。

***表 2-1** ヘルスチェックユーティリティ*

プラットフォーム	ユーティリティ名
Linux	ndscheck 構文 : `ndscheck -h hostname:port -a admin_FDN -F logfile_path --config-file configuration_file_name_and_path` **注 :** -h または --config-file を指定できますが、両方のオプションを同時に指定することはできません。
Windows	ndscheck

ヘルスチェックのタイプ

アップグレード時や ndscheck ユーティリティを実行する場合、次のタイプのヘルスチェックが行われます。

- 基本的なサーバの状態
- パーティションとレプリカの状態

ndscheck ユーティリティを実行すると、ヘルスチェックの結果は画面に表示され、ndscheck.log に記録されます。ログファイルの詳細については、「26 ページの 「ログファイル」」を参照してください。

アップグレードの一部としてヘルプチェックを実行した場合、ヘルスチェックの後にエラーの深刻度に基づいて、アップグレードを続行するかどうかの確認が求められるか、または処理が中断されます。エラーの詳細については、「25 ページの 「状態のカテゴリ」」に記載されています。

基本的なサーバの状態

これは、ヘルスチェックの最初の段階です。ヘルスチェックユーティリティは次の内容をチェックします。

1. OpenText eDirectory サービスが動作している。DIB が開いていて、ツリー名などの基本的なツリー情報を読むことができる。
2. サーバがそれぞれのポート番号を監視している。

   LDAP に関しては、TCP ポート番号と SSL ポート番号を取得して、サーバがこれらのポートを監視しているかどうかをチェックします。

   同様に、HTTP セキュアポート番号と HTTPS セキュアポート番号を取得して、サーバがこれらのポートを監視しているかどうかをチェックします。

パーティションとレプリカの状態

基本的なサーバの状態のチェック後は、次のとおり、パーティションとレプリカの状態をチェックします。

1. ローカルに保持されているパーティションのレプリカの状態をチェックします。
2. サーバによって保持されているすべてのパーティションのレプリカリングを読み込み、レプリカリング内のすべてのサーバが動作していて、すべてのレプリカが使用可能な状態であることをチェックします。
3. レプリカリング内のすべてのサーバについて、時刻同期を確認します。これによって、サーバ間の時刻の差が表示されます。

状態のカテゴリ

サーバの状態は、チェック中に検出されるエラーに基づいて、次の 3 つカテゴリに分類されます。ヘルスチェックのステータスは、ログファイルに記録されます。詳細については、「26 ページの 「ログファイル」」を参照してください。

ヘルスチェックのステータスは、正常、警告、および重大の 3 つに分類されます。

正常

ヘルスチェックが成功した場合、サーバの状態は正常です。

アップグレードは中断されずに続行されます。

警告

ヘルスチェック中に小さなエラーが見つかった場合、サーバの状態は警告に分類されます。

アップグレードの一部としてヘルスチェックが実行されている場合、中止するか続行するかの確認を求められます。

警告は通常、次の状況で発生します。

1. サーバが LDAP ポートと HTTP ポート (通常、セキュリティ保護、または両方) を監視していない。
2. レプリカリング内のいずれの非マスタサーバにも接続できない。
3. レプリカリング内のサーバが同期していない。

重大

ヘルスチェック中に致命的なエラーが見つかった場合、サーバの状態は重大に分類されます。

ヘルスチェックがアップグレードの一部として実行されている場合、アップグレード操作は破棄されます。

重大な状態は通常、次の状況で発生します。

1. DIB を開くことができないか読み込むことができない。DIB はロックされているか破損している可能性があります。
2. レプリカリング内のすべてのサーバに接続できない。
3. ローカルに保持されているパーティションが使用中である。
4. レプリカが使用可能な状態ではない。

ログファイル

サーバヘルスチェック操作は、アップグレードで実行される場合も、スタンドアロンユーティリティとして実行される場合も、状態をログファイルに保存します。

ログファイルの内容は、チェック実行時に画面に表示されるメッセージと同様です。

ヘルスチェックのログファイルには、次のものが含まれています。

- ヘルスチェックのステータス (正常、警告、または重大)。
- OpenText のサポートサイトの URL。

次の表に、さまざまなプラットフォームでのログファイルの場所を示します。

***表 2-2** ヘルスチェックのログファイルの場所*

プラットフォーム	[ログファイル名]	[ログファイルの場所]
Linux	ndscheck.log	ndscheck -F ユーティリティで指定した場所に依存します。 -F オプションを使用しない場合は、次に示すように、コマンドラインで指定した別のオプションによって、ndscheck.log ファイルの場所が決定されます。 1. -h オプションを使用した場合、ndscheck.log ファイルはユーザのホームディレクトリに保存されます。 2. --config-file オプションを使用した場合、ndscheck.log ファイルはサーバインスタンスのログディレクトリに保存されます。または、インスタンスの一覧からインスタンスを選択することもできます。
Windows	ndscheck.log	*インストールディレクトリ*

EBA 対応サーバへのブリッジパッチのインストール

- 27 ページの「Linux で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」
- 28 ページの 「Windows で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」
- 29 ページの 「コンテナ内で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」

Linux で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール

重要 : OpenText eDirectory 9.3.0 にアップグレードする前に、すべてのレプリカサーバまたはマスタサーバが 9.2.8 または 9.2.9 バージョンの OpenText eDirectory 上にあることを確認してください。これらのサーバをアップグレードするには、OpenText eDirectory 9.3.0 ビルドにバンドルされている EBA ブリッジパッチをインストールします。ブリッジパッチを適用する前にスキーマ拡張が適用されていない場合、OpenText eDirectory 9.3.0 は起動しません。

OpenText eDirectory 9.3.0 を新しいサーバにインストールし、マスタサーバが OpenText eDirectory 9.2.8 または 9.2.9 で EBA が有効になっているレプリカリングに追加する場合、OpenText eDirectory 9.3.0 をインストールする前に、EBA ブリッジパッチを OpenText eDirectory 9.2.8 または 9.2.9 サーバに適用します。

- **ルートユーザ用に Linux にブリッジパッチをインストールするには、次の手順に従います。**

1 OpenText eDirectory ビルドをダウンロードおよび untar し、untar した場所 <Untarred_edirectory_location>/eDirectory/eba_bridge_patch に移動して、次の手順を実行します。untar した後で、この場所にある eba_bridge.sch を見つけます。次のコマンドを使用してブリッジパッチスキーマを拡張します。

```
ndssch [-h <hostname>[:<port>]][-t <treename>] <admin-FDN> <schemafile>
```

次に例を示します。

```
ndssch  -h 10.10.10.10:524 -t tree1 admin.novell eba_bridge.sch
```

2 ブリッジパッチを適用するには、ルートユーザとしてログインし、次のコマンドを実行します。

```
rpm -ivh --force novell-eba-9.2.8.0000.x86_64.rpm
```

3 ndsd サービスを再開します。

- **非ルートユーザ用に Linux にブリッジパッチをインストールするには、次の手順に従います。**

1 OpenText eDirectory ビルドをダウンロードおよび untar し、untar した場所 <Untarred_edirectory_location>/eDirectory/eba_bridge_patch に移動して、次の手順を実行します。untar した後で、この場所にある eba_bridge.sch を見つけます。次のコマンドを使用してブリッジパッチスキーマを拡張します。

```
ndssch [-h <hostname>[:<port>]][-t <treename>] <admin-FDN> <schemafile>
```

次に例を示します。

```
ndssch  -h 10.10.10.10:524 -t tree1 admin.novell eba_bridge.sch
```

2 /home/nuser/eDirectory/opt/novell/eDirectory/lib64 の次のライブラリファイルをコピーします。ここで、/home/nuser はユーザ固有のパスを表しています。

```
libebacli.so, libebasrv.so, libebassl_cli.so, libebassl_srv.so and
libgssapi.so
```

3 ndsd サービスを再開します。

注 : ブリッジパッチを適用する前にスキーマ拡張が適用されていない場合、OpenText eDirectory は起動しません。

Windows で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール

Windows にブリッジパッチをインストールするには、次の手順に従います。

1 <extracted_patch>eDirectory\eba_bridge_patch にある OpenText eDirectory ビルドを圧縮解除します。

2 preUpgrade_930 フォルダにある upgrade.ni ファイルを更新します。既存の eDirectory サーバのツリー名、サーバ名、および管理者資格情報を、upgrade.ni レスポンスファイルに指定します。次に示すのは、アップグレードの設定を指定した upgrade.ni レスポンスファイルのサンプルです。

```
[NWI:NDS]
mode=configure
Tree Name=enewtree
Server Name=enewserver
Server Container=org
Admin Context=org
Admin Login Name=Admin
Admin Password=env:PASSWORD_VAR
Require IPV6=NO
Require EBA=NO
Require FIPS for TLS=YES
LDAP TCP Port=389
LDAP SSL Port=636
Require TLS=Yes
Require SS=Yes
Existing Server=10.10.10.10
Existing Server Port=524
[NWI:SNMP]
Stop service=No
[NWI:NMAS]
Methods=CertMutual,Challenge Response,DIGEST-MD5,NDS,Simple
Password,SAML
```

3 powershell から次のコマンドを実行します。

```
preUpgrade.ps1 -rfile <path_for_Configuration>/upgrade.ni
```

4 OpenText eDirectory サービスを停止してから、これらの DLL (ebacli.dll、ebai18n.dll、ebaldap.dll、ebassl_cli.dll、ebasym.dll) を <extracted_patch>eDirectory\eba_bridge_patch から C:\NetIQ\eDirectory (OpenText eDirectory バイナリの場所) にコピーします。

5 ndsd サービスを開始します。

コンテナ内で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール

コンテナにブリッジパッチをインストールするために、次の手順に従います。

1 OpenText eDirectory ビルドをダウンロードおよび untar し、untar した場所 <Untarred_edirectory_location>/eDirectory_930/eba_bridge_patch に移動して、次の手順を実行します。

2 untar した後で、

2a コンテナに eba_brdge.sch をコピーして、次のコマンドを実行します。

```
docker cp eba_brdge.sch <eDirectory Container name>:/config
```

2b コンテナに novell-eba-9.2.8.0000.x86_64.rpm をコピーして、次のコマンドを実行します。

```
docker cp novell-eba-9.2.8.0000.x86_64.rpm <eDirectory Container
name>:/config
```

3 コンテナにログインし、29 ページのステップ 2a でスキーマファイルをコピーしたディレクトリに移動して、次のコマンドを実行します。

スキーマの拡張

```
ndssch [-h <hostname>[:<port>]][-t <treename>] <admin-FDN> <schemafile>
```

例 :

```
ndssch  -h 10.10.10.10:524 -t tree1 admin.novell eba_bridge.sch
```

4 EBA ブリッジパッチ RPM をインストールするには、次の手順を実行します。

4a ルートユーザとしてコンテナにログインします。

```
docker exec -it -u root <eDirectory Container name> /bin/bash
```

4b ページ 146 のステップ 2b で rpm をコピーしたディレクトリに移動し、次のコマンドを実行します。

```
rpm -ivh --force novell-eba-9.2.8.0000.x86_64.rpm
```

5 ndsd サービスを再開するために、次のコマンドを使用して OpenText eDirectory コンテナを再起動します。

```
docker restart <eDirectory Container name>
```

その他の情報

この章で説明する機能の詳細については、次のマニュアルを参照してください。

- *OpenText™ eDirectory CE 24.4 (v9.3) 管理ガイド*
- Linux の場合 : nds-install ndsconfig、および ndscheck のマニュアルページ。

  例 : man nds-install

3 OpenText eDirectory CE 24.4 (v9.3) の Linux へのインストールまたはアップグレード

Linux サーバで OpenText eDirectory CE 24.4 (v9.3) をインストールまたはアップグレードする場合は、次の情報を参照してください。

- 31 ページの 「システム要件」
- 33 ページの 「前提条件」
- 38 ページの 「ハードウェア要件」
- 38 ページの 「バックリンク処理の強制実行」
- 39 ページの 「edirAdmin ユーザグループの追加」
- 40 ページの 「OpenText eDirectory をアップグレードする」
- 46 ページの 「OpenText eDirectory のインストール」

システム要件

OpenText eDirectory は、最小要件として以下のいずれかの 64 ビットのプラットフォームにインストールする必要があります。

- メモリ
  - OpenText eDirectory サーバのための 2GB の空きディスク容量
  - 50,000 ユーザごとに 2GB のディスク容量
- 仮想化システム
  - VMware ESXi
- 次のいずれかのオペレーティングシステムが必要です。

  次の表は、アイデンティティボールトが動作可能な、認定済みサーバオペレーティングシステムおよびサポートされているサーバオペレーティングシステムのリストを示しています。

重要 : 認定済みとは、完全にテストされてサポートされているオペレーティングシステムを意味します。ただし、サポートされているオペレーティングシステムとして一覧表示されている場合は、まだテストされていないが、機能することが想定されていることを意味します。

認定済みサーバオペレーティングシステムのバージョン	サポートされているオペレーティングシステム	備考
SUSE Linux Enterprise Server 15 SP3、SP4、SP5、および SP6	新しいバージョンのサポートパックでサポート	システム要件に関する最新情報については、『リリースノート』を参照してください。
RHEL 8.8 以上	新しいバージョンのサポートパックでサポート	システム要件に関する最新情報については、『リリースノート』を参照してください。 **注** ◆ RHEL 8.8 以上では、SELinux を permissive モードに設定する必要があります。
RHEL 9.2 以上	新しいバージョンのサポートパックでサポート	システム要件に関する最新情報については、『リリースノート』を参照してください。 **注 :** ◆ RHEL 9.2 以上では、SELinux を permissive モードに設定する必要があります。 ◆ SELNIUXがENFORCINGに設定されている場合は、33 ページの 「前提条件」に示されている手順に従ってください。

実行している SUSE Linux のバージョンを調べるには、/etc/os-release ファイルを確認します。

Red Hat システムに、Red Hat Errata から配布されている最新の glibc (http://rhn.redhat.com/errata) パッチ (32 ビットと 64 ビットの両方) が適用されていることを確認してください。glibc ライブラリのバージョンは 2.4 以上である必要があります。

Linux マシン用の適切なエントリが /etc/hosts ファイルで利用可能になっていることを確認してください。

注 :

- B ツリーファイルシステム (BTRFS) は、OpenText eDirectory ではサポートされていません。

OpenText eDirectory のバージョンの確認

OpenText eDirectory のバージョンを確認するには、次のいずれかの手順を実行します。

- ndsstat を実行する。

  ndsstat ユーティリティでは、OpenText eDirectory ツリー名、完全識別サーバ名、および OpenText eDirectory バージョンなど、OpenText eDirectory サーバに関連する情報が表示されます。次の例では、OpenText eDirectory 9.3 が製品バージョン (マーケティング文字列) で、40201.12 がバイナリバージョン (内部ビルド番号) です。

  ```
  osg-dt-srv17:/>ndsstat
  Tree Name: EDR93E
  Server Name: .CN=Master.O=server.T=EDR93E.
  Binary Version: 40301.00
  Root Most Entry Depth: 0
  Product Version: eDirectory for Linux x86_64 v9.3 [DS]
  ```

  ndsstat の実行方法の詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「OpenText eDirectory 用の Linux コマンドとそれらの使用法」、または ndsstat のマニュアルページ (ndsstat.1m) を参照してください。

- ndsd --version を実行する。

  ndsd の実行方法の詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「OpenText eDirectory 用の Linux コマンドとそれらの使用法」、または ndsd のマニュアルページ (ndsd.1m) を参照してください。

- iMonitor を実行する。

  エージェントの概要ページで［認識サーバ］をクリックします。次に、［データベースで認識されているサーバ］の下にある［認識サーバ］をクリックします。［エージェントリビジョン］カラムに各サーバの内部ビルド番号が表示されます。たとえば、OpenText™ eDirectory CE 24.4 (v9.3) のエージェントリビジョン番号は 40301.00 などとなります。

  iMonitor の実行方法の詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*OpenText iMonitor へのアクセス*」を参照してください。

- rpm -qi NDSserv を実行する。

  このコマンドを入力すると、ndsd --version に似た情報が表示されます。

前提条件

重要 : 既存の OpenText eDirectory 環境をアップグレードする前に、現在インストールされている OpenText アプリケーションとサードパーティ製アプリケーションが OpenText eDirectory 9.3 に対応しているかどうかを確認してください。他の OpenText 製品の前提条件

は、OpenText Documentation のサイト (http://www.netiq.com/documentation/) で確認できます。また、OpenText eDirectory インスタンスをアップグレードする前に、そのインスタンスをバックアップすることをお勧めします。

- ❒ オペレーティングシステムに基づいて、次の RPM をインストールするようにしてください。
  - **RHEL 8.x および RHEL 9:** dnf-utils および createrepo

    RHEL 8.x RPM をインストールするには、次の手順を実行します。

    1. yum-utils パッケージをインストールします。

       yum install createrepo yum-utils

    2. OpenText eDirectory に必要な次のライブラリをインストールします。

       yum install libgcc*.i686 libncurses*

    3. /etc/selinux/config にある SELinux 環境設定ファイルに移動し、次のようにして SELinux を permissive モードに設定します。

       SELINUX=permissive

  - **RHEL の最小バージョン :** ncurses、libxcrypt-compat、iproute、initscripts、procps、および net-tools。
  - **SLES:** Zypper
- ❒ OpenText eDirectory のインストール時に RHEL OS のアップグレードが失敗する場合は、回避策として次の手順に従ってください。
  - RHEL OS をアップグレードする前に、次のコマンドを使用してパッケージを削除します。

    rpm -e --nodeps novell-NLDAPsdk
  - RHEL OS を 8.x から 9.x 以降にアップグレードします。
  - OS をアップグレードした後、次のコマンドを使用してパッケージをインストールします。

    rpm -ivh ./novell-NLDAPsdk*
  - OpenText eDirectory サービスを再開します。
- ❒ SELINUX が Enforcing に設定されている場合は、次の手順を実行して、OpenText eDirectory をインストールします。
  1. OpenText eDirectory ビルドをダウンロードして、untar します。
  2. OpenText eDirectory パッケージをインストールし、./nds-install(untared_location/eDirectory/setup にある) を実行します。
  3. ツリーを設定する前に、OpenText eDirectory のデフォルト dib の場所 /var/opt/novell/eDirectory にデータディレクトリを作成します。

     dib の場所が、/var/opt/novell/eDirectory/data に変更されます。

     (任意の場所を設定するには、カスタムパス ex/home/edirectory/data を指定します)
  4. 次のコマンドを実行します。
     - touch /var/opt/novell/eDirectory/data/ndsd.pid

- semanage fcontext -a -t var_run_t '/var/opt/novell/eDirectory/data/ndsd.pid'
- restorecon -v '/var/opt/novell/eDirectory/data/ndsd.pid'

5. 次に、ndsconfig new コマンドを使用してツリーを設定します。

❒ (状況によって実行) Novell International Cryptographic Infrastructure (NICI) 3.3 および OpenText eDirectory 9.3 でサポートされているキーサイズは、最大 8192 ビットです。8KB のキーサイズを使用するには、すべてのサーバが OpenText eDirectory 9.3 にアップグレードされている必要があります。また、Identity Console などの管理ユーティリティを使用しているすべてのワークステーションに NICI 3.3 がインストールされている必要があります。

認証局 (CA) サーバを OpenText eDirectory 9.3 にアップグレードするとき、キーサイズは変更されず、2K のままになります。8K のキーサイズを作成するには、OpenText eDirectory 9.3 サーバで CA を再作成する必要があります。また、CA を作成する際に、デフォルトのキーサイズを 2K から 8K に変更する必要があります。

OpenText eDirectory をインストールすると、nds-install ユーティリティによって自動的に NICI がインストールされます。OpenText eDirectory のインストールの詳細については、48 ページの 「nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする」を参照してください。ただし、管理ユーティリティがインストールされているワークステーションに、OpenText eDirectory そのものではなく NICI だけをインストールする必要がある場合は、NICI を手動でインストールする必要があります。NICI の手動インストールの詳細については、「52 ページの 「NICI のインストール」」を参照してください。

❒ OpenText eDirectory サーバ用の 168 ビット 3DES ツリーキーを取得してください。

❒ (状況によって実行) サービスロケーションプロトコル (SLP) は、DNS が使用できないときに SLP を使用してツリー名を解決する計画の場合に限り、インストールおよび設定する必要があります。

OpenText eDirectory 9.3 では、SLP は OpenText eDirectory インストールの一部としてインストールされません。

ルートユーザだけが SLP をインストールできます。

SLP のインストールの詳細については、「47 ページの 「OpenText eDirectory での SLP の使用」」を参照してください。

❒ マルチキャストルーティングを使用するための Linux ホストの有効化

マルチキャストルーティングを使用するためにホストが有効になっていることを確認するには、次のコマンドを入力します。

```
/bin/netstat -nr
```

ルーティングテーブルに、次のエントリがあればマルチキャストルーティングが有効になっています。

```
224.0.0.0 0.0.0.0
```

このエントリがない場合は、ルートユーザとしてログインし、次のコマンドを入力してマルチキャストルーティングを有効にします。

```
route add -net 224.0.0.0 netmask 240.0.0.0 dev interface
```

interface は、取り付けられ、使用されている NIC に応じて、eth0、hme0、hme1、または hme2 などの値になります。

マルチキャストおよびブロードキャストルータの詳細については、OpenSLP Web サイト (http://www.openslp.org/doc/html/UsersGuide/Installation.html) を参照してください。

- ネットワークサーバ時刻が同期されている

  すべてのネットワークサーバの時刻を同期するには、NTP (ネットワーク時刻プロトコル) の ntp を使用します。

- (状況によって実行) セカンダリサーバをインストールする場合は、製品をインストールするパーティション内のレプリカがすべて、オンの状態になっている必要があります。
- (状況によって実行) 管理者以外のユーザとしてセカンダリサーバを既存のツリーにインストールしている場合、コンテナを作成し、そのコンテナをパーティションで分割します。次の権限を持っていることを確認します。
  - 対象のパーティションに対するスーパバイザ権。
  - すべての属性権 : W1.KAP.Security オブジェクトに対する読み込み権、比較権、および書き込み権
  - エントリ権 : Security コンテナオブジェクトに対するブラウズ権。
  - すべての属性権 : Security コンテナオブジェクトに対する読み込み権、比較権。
  - (状況によって実行)W1.KAP.Security オブジェクトが存在する場合は、すべての属性権 : このオブジェクトに対する読み込み権、比較権、および書き込み権。
    W1.KAP.Security オブジェクトの詳細については、『NICI Administration Guide (NICI 管理ガイド)』の「*Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)*」を参照してください。
- (状況によって実行) 管理者以外のユーザとしてセカンダリサーバを既存のツリーにインストールしている場合、ツリー内の少なくとも 1 台のサーバの OpenText eDirectory バージョンが、コンテナ管理者として追加しているセカンダリの OpenText eDirectory バージョンと同じかそれ以上である必要があります。追加しているセカンダリの方が新しいバージョンである場合、ツリーの管理者がスキーマを拡張してから、コンテナ管理者でセカンダリを追加する必要があります。
- セカンダリサーバを追加できるように、OpenText eDirectory の設定中に、SLP サービスとファイアウォールの NetWare Core Protocol (NCP) ポート (デフォルトは 524) を必ず有効にしてください。必要に応じて、さらに以下のサービスポートも有効にできます。
  - LDAP (クリアテキスト) - 389
  - LDAP (セキュリティ保護) - 636
  - HTTP (クリアテキスト) - 8028
  - HTTP (セキュリティ保護) - 8030

  ユーザ定義ポートを有効にしてある場合、OpenText eDirectory の設定時にそのポートを指定する必要があります。

注 : このステップはシステムにサービスロケーションプロトコル (SLP) が設定されている場合にのみ必要です。

- ❒ OpenText eDirectory 8.8 SP8 以降のバージョンを 9.2 にアップグレードする際に、ユーザ定義ポートを 8008 および 8010 に設定しないでください。ポートを 8008 または 8010 に設定すると、ndsconfig の実行時に、サーバが eDirectory 8.8.*x* 以前のサーバであると想定され、ポートが自動的に 8028 および 8030 にリセットされます。
- ❒ 以前のバージョンで SecretStore がまだ設定されていなかった場合、または SecretStore を設定しない場合は、OpenText eDirectory のアップグレードの際に、-m no_ss オプションを指定して nds-install ユーティリティを使用してください。
- ❒ OpenText eDirectory 管理ツールボックス (eMBox) を使用すると、サーバ上でもリモートでも eDirectory のバックエンドユーティリティすべてにアクセスできます。コマンドラインクライアントは、Java アプリケーションです。これを実行するには、Oracle Java の最新バージョン (1.8 以上) をインストールする必要があります。また、古いバージョンの Java すべてで、使用可能なパッチアップグレードをインストールしてアップグレードしてください。最新バージョンの Java をインストールしたら、次の環境変数をエクスポートします。
  - EDIR_JAVA_HOME
  - JAVA_HOME
  - JRE_HOME

  注 :
  - 前述の環境変数が何も検出されないと、コマンドラインクライアントはデフォルト PATH 環境変数で Java バイナリを検索します。
  - OpenText eDirectory 9.0 SP4 より前のバージョンを使用している場合、コマンドラインクライアントを実行するには、OpenText eDirectory と一緒にインストールされた Java ランタイム環境 (Oracle Java 1.8) にアクセスできなければなりません。

RPM 署名 : 署名を検証する公開鍵

OpenText eDirectory 9.2.7 以降から、Linux システムに OpenText eDirectory コンポーネントをインストールする前に、次の手順に従って RPM 署名の検証を実行します。

1. 公開鍵のある次の場所に移動します。

   <untarred location of eDirectory>/eDirectory/license/MicroFocusGPGPackageSign.pub
2. 次のコマンドを実行して、公開鍵をインポートします。

   rpm --import MicroFocusGPGPackageSign.pub
3. (オプション) 次のコマンドを実行して、RPM 署名を検証します。

   rpm --checksig -v <RPM Name>

   例 :

   rpm --checksig -v novell-NDSbase-9.2.x.0000.x86_64.rpm

注 : OpenText eDirectory 9.3.0 は、RHEL8 以降および RHEL 9 以降、SLES 12 SP5、および SLES 15 SP3、SP4、SP5、SP6 のサポートされているバージョンの FIPS 対応 OS にインストールされます。

スタティック IP アドレスを設定する

OpenText eDirectory を効率的に実行するには、サーバでスタティック IP アドレスが設定されている必要があります。DHCP アドレスのサーバで OpenText eDirectory を設定すると、予期しない結果が発生することがあります。

ハードウェア要件

ハードウェア要件は、OpenText eDirectory の特定の実装により異なります。キャッシュメモリの量とプロセッサの速さという、2 つの要因によってパフォーマンスが向上します。最適な結果を得るためには、ハードウェアで可能な限り多くの DIB (Directory Information Base) セットをキャッシュに入れるようにします。

OpenText eDirectory はシングルプロセッサ上で良好に動作します。ただし、OpenText eDirectory 9.3 では、マルチプロセッサの利点を活用できます。プロセッサを追加すると、ログインなど、一部の領域のパフォーマンスが向上します。また、複数のプロセッサ上で複数のスレッドをアクティブにすることでもパフォーマンスは上がります。OpenText eDirectory 自体は、プロセッサ集約型ではなく、入出力集約型です。

次の表に、OpenText eDirectory for Linux の一般的なシステム要件を示します。

オブジェクト	メモリ	ハードディスク
100,000	2GB 以上	300MB
100 万	4GB	1.5GB
1,000 万	4GB 以上	15GB

バックリンク処理の強制実行

OpenText eDirectory にアップグレードすると内部の OpenText eDirectory 識別情報が変わるため、オブジェクトの整合性を保つために、バックリンクされたオブジェクトを更新するバックリンク処理を行う必要があります。

バックリンクでは、他のサーバ上のオブジェクトへの外部参照が追跡されます。バックリンク処理は、サーバ上の各外部参照について、実オブジェクトが正しい位置に存在することを確認するほか、マスタレプリカのすべてのバックリンク属性を確認します。バックリンク処理はデータベースがオープンされた 2 時間後に実行され、その後 780 分 (13 時間) ごとに実行されます。実行間隔には、2 分から 10,080 分 (7 日) までの任意の値を設定できます。

OpenText eDirectory に移行した後、ndstrace -l>log& コマンドを発行して、DSTrace 処理を開始します。この処理はバックグラウンドで実行されます。これにより、バックリンクプロセス結果を適切に分析できます。分析には 4 ～ 10 分かかります。DSTrace OS のコマンドプロンプトから ndstrace -c 'set ndstrace=*B' コマンドを発行して、バックリンク処理を強制実行することができます。最初の手順で作成されたログファイルの結果を確認します。次に、ndstrace -u コマンドを発行して DSTrace 処理をアンロードします。バックリンク処理の実行は、レプリカが存在しないサーバ上では特に重要です。

edirAdmin ユーザグループの追加

OpenText eDirectory 9.3 リリースでは、eDirAdmin 固定グループ名が削除されました。これは、現在の動作を取り入れるために変更されました。柔軟で、設定可能で、変更可能で、オプションです (必須ではありません)。新規インストールでは、edirAdmin グループは作成されません。グループ名を入力するオプションが提供され、もし入力すると、その名前が edirAdmin の代わりに使用されます。グループ名は、ndsconfig new/add/upgrade オプションの -g オプションを使用して設定できます。ユーザは、有効な、既存のグループの名前を -g オプションに入力する必要があります。-g と groupname は間を空けずに入力してください (-g <groupname>)。指定されたグループ名は、環境設定ファイルに「n4u.server.osgroup-name」として保存されます。空の場合は、使用されません。それ以外の場合は、選択したディレクトリに制限を設定するために使用されます。インスタンスを作成するユーザは、制限を設定するためには、グループの一部である必要があります。これはオプションであり、指定しない場合は、グループ設定を行う必要がないと見なされます。9.2.5/9.2.6 からのアップグレード / マイグレーションの場合、edirAdmin はすでに所有者であり、設定しているユーザもそのグループの一部である必要があります。

注 : 次の edirAdmin ユーザグループ設定は、バージョン 9.2.5 および 9.2.6 の OpenText eDirectory に適用できます。

OpenText eDirectory 9.2.5 リリースでは、インストール中に新しい edirAdmin ユーザグループが OpenText eDirectory に追加されます。このユーザグループは、データファイル、ログファイル、および環境設定ファイルがインストールされているディレクトリへのアクセスパーミッションを持ちます。このグループに追加されたユーザには、これらのディレクトリに対するグループ許可が割り当てられます。

ルートユーザによるインストールと設定中には、インストーラによりデフォルトで edirAdmin ユーザグループが作成されます。ただし、非ルートユーザによるインストールと設定では、OpenText eDirectory の最新バージョンにアップグレードまたはインストールする前に、このグループを手動で作成する必要があります。グループを作成した後、ユーザが自分の tarball インストールを使用するかバイナリインストールを使用して OpenText eDirectory サーバインスタンスを設定できるようにするには、そのユーザをこのグループに追加する必要があります。一方、非ルートユーザがルートユーザによりインストールされた OpenText eDirectory の設定を実行すると、このグループはインストールの一部として追加されます。ユーザがこのグループに追加されていることを確認する必要があります。追加されていないと、ndsconfig ユーティリティを使用して OpenText eDirectory の設定を行うときにエラーが発生する可能性があります。

OpenText eDirectory がインストールされているサーバで、端末を開き、次のコマンドを実行します。

- edirAdmin グループを追加するには、groupadd edirAdmin を実行します。
- ユーザを edirAdmin グループに追加するには、usermod -a -G edirAdmin <username> を実行します。

  ここで、<username> はユーザの名前です。たとえば、John というユーザを edirAdmin グループに追加するには、次のコマンドを実行する必要があります。

  $ usermod -a -G edirAdmin john

注 : OpenText edirAdmin ユーザグループにユーザを追加する前に、管理者は、そのユーザが端末セッションを通じてサーバにログインしていないことを確認する必要があります。追加された後、ユーザは ndsconfig ユーティリティを使用して OpenText eDirectory にログインし、設定できます。

OpenText eDirectory をアップグレードする

OpenText eDirectory をバージョン 8.8 SP1 から 9.3 に直接アップグレードすることはしないようにお勧めします。9.2 にアップグレードしてから、9.3 にアップグレードする必要があります。OpenText eDirectory を 8.8 SP1 より前のバージョンから 9.2 にアップグレードすると、OpenText eDirectory サーバのパフォーマンスに影響が出ます。

注 : 32 ビットバージョンの OpenText eDirectory を 64 ビットバージョンの eDirectory にアップグレードするには、まず 32 ビットバージョンを eDirectory 8.8.x 64 ビットバージョンにアップグレードし、次にそれを OpenText eDirectory 9.2 にアップグレードします。64 ビットの OpenText eDirectory を OpenText eDirectory 9.3 にアップグレードする場合も、同じ手順に従います。

次のセクションでは、インストール済みの既存の OpenText eDirectory を現在のバージョンにアップグレードするのに役立つ情報を提供します。

- 41 ページの 「サーバのヘルスチェック」
- 41 ページの 「OES 以外の Linux サーバでのアップグレード」
- 42 ページの 「Linux 上での OpenText eDirectory の無人アップグレード」
- 43 ページの 「tarball デプロイメントの eDirectory 9.3 をアップグレードする」
- 44 ページの 「非ルートユーザによる OpenText eDirectory 9.3 へのアップグレード」
- 45 ページの 「複数インスタンスをアップグレードする」

注 : ndsconfig upgrade コマンドは、HTTP、LDAP、SNMP、SAS、および NMAS (NetIQ Modular Authentication Service) などの個別のコンポーネントで必要な環境設定をアップグレードするために使用します。

サーバのヘルスチェック

OpenText eDirectory 9.3 の場合、OpenText eDirectory のアップグレードの際に、デフォルトでサーバヘルスチェックが実行され、サーバをアップグレードしても安全かどうかが確認されます。

- 215 ページの 「パーティションとレプリカの状態」

ヘルスチェックの結果に基づいて、次のようにアップグレードが継続または中止されます。

- すべてのヘルスチェックに成功すると、アップグレードは継続されます。
- あまり重大でないエラーの場合、アップグレードを継続するか中止するかを問うメッセージが表示されます。
- 重大なエラーの場合、アップグレードは中止されます。

あまり重大でないエラーと重大なエラーの一覧については「213 ページの付録 B「OpenText eDirectory ヘルスチェック」」を参照してください。

サーバのヘルスチェックのスキップ

サーバのヘルスチェックをスキップするには、インストールフォルダから nds-install -j または ndsconfig upgrade -j を実行します。

詳細については、「213 ページの付録 B「OpenText eDirectory ヘルスチェック」」を参照してください。

OES 以外の Linux サーバでのアップグレード

OpenText eDirectory のアップグレードは、OpenText eDirectory 9.20 以降でサポートされます。

アップグレードするには、nds-install ユーティリティを使用します。このユーティリティは、Linux プラットフォーム用にダウンロードしたファイルの Setup ディレクトリにあります。Setup ディレクトリから次のコマンドを入力します。

```
./nds-install
```

OpenText eDirectory 9.2 にアップグレードすると、環境設定ファイル、データファイル、およびログファイルのデフォルトの保存先は、それぞれ /etc/opt/novell/eDirectory/conf、/var/opt/novell/eDirectory/data、および /var/opt/novell/eDirectory/log に変わります。

新しい /var/opt/novell/eDirectory/data ディレクトリは、/var/nds ディレクトリへのシンボリックリンクを使用します。

古い環境設定ファイル /etc/nds.conf は、/etc/opt/novell/eDirectory/conf ディレクトリに移行されます。古い環境設定ファイル /etc/nds.conf、および /var/nds にある古いログファイルは参照できるように保持されます。

注 : DIB のアップグレードが失敗し、nds-install から指示があった場合は、nds-install 実行後に ndsconfig upgrade を実行してください。RHEL 6.8 から 7.1 への OS のアップグレード後に OpenText eDirectory サービスが開始しない場合は、ndsconfig upgrade コマンドを実行してください。

注 : 時刻同期が原因でヘルスチェックに失敗することがあります。この問題を解決するには、インスタンス間で時刻同期を実行してください。アップグレード中のこの警告メッセージは無視してもかまいません。

Linux 上での OpenText eDirectory の無人アップグレード

Linux 用の OpenText eDirectory には、無人アップグレードを円滑に行うためのスイッチ、オプション、インストールスクリプト、および環境設定ユーティリティが用意されています。以降のセクションで、Linux で OpenText eDirectory を無人アップグレードするためのさまざまな手順について説明します。

1 OpenText eDirectory のヘルスチェックを実行します。

アップグレードが計画されているすべてのルートインスタンスのヘルスチェックは、ndscheck ユーティリティを使って、手動で実行します。

1a LD_LIBRARY_PATH を次の場所にエクスポートします :*<eDirectory を untar した場所 >*/eDirectory/setup/utils

1b 次のいずれかのコマンドを使って、ndscheck を実行します。

```
<untarred location of eDirectory>/eDirectory/setup/utils/ndscheck -a <user name> -w passwd --config-file <nds.conf with absolute path>
```

環境変数からパスワードを渡す場合 : *<88SP8 を untar した場所 >*/eDirectory/setup/utils/ndscheck -a < *ユーザ名 >* -w env:< *環境変数 >* --config-file < *絶対パスで指定した nds.conf>*

ファイルからパスワードを渡す場合 : *<88SP8 を untar した場所 >*/eDirectory/setup/utils/ndscheck -a < *ユーザ名 >* -w file:< *ファイル名 >* --config-file < *絶対パスで指定した nds.conf>*

いずれも、ヘルスチェックの自動化スクリプトで使用できます。次に例を示します。

```
/Builds/eDirectory/utils/ndscheck -a admin.novell -w n
/Builds/eDirectory/utils/ndscheck -a admin.novell -w env:ADM_PASWD
/Builds/eDirectory/utils/ndscheck -a admin.novell -w file:adm_paswd
```

2 OpenText eDirectory 9.3 パッケージをアップグレードします。

2a 以下のように nds-install スクリプトを実行して、パッケージをアップグレードします。

```
nds-install -u -i -j
```

3 次の環境変数を更新します。

```
PATH=/opt/novell/eDirectory/bin:/opt/novell/eDirectory/sbin:$PATH
LD_LIBRARY_PATH=/opt/novell/eDirectory/lib:/opt/novell/eDirectory/lib/
nds-modules:/opt/novell/lib:$LD_LIBRARY_PATH
MANPATH=/opt/novell/man:/opt/novell/eDirectory/man:$MANPATH
TEXTDOMAINDIR=/opt/novell/eDirectory/share/locale
```

4 ndsconfig ユーティリティで、すべてのルートインスタンスに対して以下のコマンドを実行して OpenText eDirectory をアップグレードします。

```
ndsconfig upgrade -a <user name> -w passwd -c --config-file <nds.conf with
absolute path> --configure-eba-now <yes/no>
```

注 : 拡張バックグラウンド認証を有効にするには、ndsconfig upgrade コマンドの --configure-eba-now スイッチで *yes* を指定します。有効にしない場合は、*no* を指定して後から設定します。

環境変数からパスワードを渡す場合 : ndsconfig upgrade -a < *ユーザ名* > -w env:< *環境変数* > -c --config-file < *絶対パスで指定した nds.conf*> --configure-eba-now <yes/no>

ファイルからパスワードを渡す場合 : ndsconfig upgrade -a < *ユーザ名* > -w file:< *絶対 / 相対パスで指定したファイル名* > -c --config-file < *絶対パスで指定した nds.conf*> --configure-eba-now <yes/no>

上記のいずれの方法も、eDirectory アップグレードの自動化スクリプトで使用できます。次に例を示します。

```
ndsconfig upgrade -a admin.novell -w n -c -–config-file /etc/opt/
novell/eDirectory/conf/nds.conf --configure-eba-now <yes/no>
```

```
ndsconfig upgrade -a admin.novell -w env:ADM_PASWD -c --config-file /
etc/opt/novell/eDirectory/conf/nds.conf --configure-eba-now <yes/no>
```

```
ndsconfig upgrade -a admin.novell -w <password file path>/adm_paswd -c
--config-file /etc/opt/novell/eDirectory/conf/nds.conf --configure-eba-
now <yes/no>
```

tarball デプロイメントの eDirectory 9.3 をアップグレードする

tarball デプロイメントを OpenText eDirectory 9.20 から OpenText eDirectory 9.3 にアップグレードする場合、次の手順を実行してください。

1 tarball ビルドをダウンロードします。

2 次の環境設定ファイルのバックアップを作成します。

- $NDSHOME/eDirectory/etc/opt/novell/eDirectory/conf/ndsimon.conf
- $NDSHOME/eDirectory/etc/opt/novell/eDirectory/conf/ice.conf
- $NDSHOME/eDirectory/etc/opt/novell/eDirectory/conf/ndsimonhealth.conf
- $NDSHOME/eDirectory/etc/opt/novell/eDirectory/conf/ndssnmp/ndssnmp.cfg
- $NDSHOME は、OpenText eDirectory がインストールされている場所です。

3 OpenText eDirectory 8.8 SP1 未満のバージョンをアップグレードする場合は、次の手順を実行します。

- ndscheck -D --config-file conf_file_path を使って、ディスクの空き容量をチェックします。
- 各サーバインスタンスの DIB ロケーションの下に、upgradeDIB という空のファイルを作成します。

  インスタンスのリストは、ndsmanage ユーティリティを使って取得できます。

4 ndscheck を使って、すべてのインスタンスに対してアップグレード前のヘルスチェックを実行し、アップグレードを進める前に、エラーがないか ndscheck.log ファイルを確認します。

5 ndsmanage を使って、すべてのインスタンスを停止します。

6 OpenText eDirectory のインストール先と同じ場所 ($NDSHOME) で tarball を untar します。tarball を同じ場所に untar することで、バイナリファイルとライブラリが上書きされます。

7 必要に応じて、以下のパッケージをアップグレードします。

プラットフォーム	コマンド	パッケージ
Linux		• novell-NOVLsubag-9.2.0-0.x86_64.rpm • netiq-openssl-3.0.15.x86_64.rpm **注 :** 64 ビットの NICI のインストールの詳細については、「52 ページの 「NICI のインストール」」を参照してください。

8 環境設定ファイルを復元します。

9 すべての環境変数を設定するには、$NDSHOME/eDirectory/opt/novell/eDirectory/bin/ndspath を実行します。

10 すべてのインスタンスに対して ndsconfig upgrade -j を実行します。ndsconfig upgrade を実行する際は、最初にマスタレプリカ、次に読み込み / 書き込みとその他という順番を守ってください。

非ルートユーザによる OpenText eDirectory 9.3 へのアップグレード

非ルートユーザは、新しいバージョンの tarball を使用して OpenText eDirectory をアップグレードできます。非ルートユーザとして OpenText eDirectory をアップグレードするには、次の手順を実行します。

1 ssh を使用し、非ルートユーザとして、OpenText eDirectory がインストールされているコンピュータにログインします。

2 OpenText eDirectory をインストールするディレクトリに移動します。

3 次のようにして、新しいバージョンの tar ファイルを untar します。

```
tar xvf /tar_file_name
```

etc、opt、var の各ディレクトリが作成されます。

4 次のようにして、すべての eDirectory サーバを停止します。

```
ndsmanage stopall
```

5 次のコマンドを使用して、ルートユーザとして OpenSSL および NICI をアップグレードします。

- NICI をアップグレードする前に、OpenSSL をアップグレードする必要があります。OpenSSL をインストールするために、次のコマンドを入力します。

  ```
  rpm -Uvh openssl_rpm_absolute_path/netiq-openssl-3.0.15.x86_64.rpm
  ```

- 64 ビット NICI をアップグレードするために、次のコマンドを入力します。

  ```
  rpm -Uvh NICI_rpm_absolute_path/nici64-3.3.0-0.00.x86_64.rpm
  ```

- NICI をサーバモードに設定するには、ルートユーザとして次のように入力します。

  ```
  /var/opt/novell/nici/set_server_mode64
  ```

6 次のコマンドを実行して、OpenText eDirectory サーバをアップグレードします。

```
ndsconfig upgrade
```

複数インスタンスをアップグレードする

このセクションでは、次のことを説明します。

- 45 ページの 「ルートユーザが複数のインスタンスを所有している」
- 46 ページの 「非ルートユーザのインスタンス」
- 46 ページの 「アップグレードの順番」

ルートユーザが複数のインスタンスを所有している

パッケージをアップグレードした後に nds-install を実行すると、すべての OpenText eDirectory サーバインスタンスの DIB ファイルをアップグレードするように求められます。この処理が完了するまで長時間かかることがあります。DIB アップグレードをパラレルで実行する場合は、手動で行うことができます。DIB の手動アップグレードの詳細については、『OpenText eDirectory リリースノート』を参照してください。すべてのアクティブなインスタンスの DIB を 1 つずつアップグレードする場合は、インスタンスごとに ndsconfig upgrade コマンドを実行します。DIB のサイズが大きい場合、[No] を選択し、個別のシェルで ndsconfig upgrade をパラレルで実行することができます。これにより、各インスタンスのアップグレード時間を短縮できます。

非ルートユーザのインスタンス

非ルートユーザのインスタンスがあり、それがルートユーザのバイナリを使用している場合、パッケージをアップグレードする前に、このようなインスタンスに対して ndscheck を実行し、ndscheck.log ファイルを参照して、インスタンスのヘルスが適切であることを確認する必要があります。nds-install を実行すると、非ルートユーザのインスタンスを含め、すべてのインスタンスが停止します。パッケージのアップグレード後、nds-install コマンドを実行しても、非ルートユーザのインスタンスに対して ndsconfig upgrade はコールされません。こうしたインスタンスを起動するには、すべての非ルートユーザのインスタンスに対して ndsconfig upgrade を手動で実行する必要があります。

アップグレードの順番

ndsconfig upgrade を実行する際は、最初にマスタレプリカに対して実行し、次に読み込み / 書き込みやその他のレプリカに対して実行することをお勧めします。

注 : Linux で eDirectory 9.2.8 および 9.2.9 での EBA 対応のレプリカサーバをアップグレードするには、27 ページの 「Linux で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」を参照してください。

OpenText eDirectory のインストール

次のセクションでは、OpenText eDirectory の Linux へのインストールについて説明します。

- 47 ページの 「OpenText eDirectory での SLP の使用」
- 48 ページの「nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする」
- 51 ページの 「非ルートユーザによる OpenText eDirectory 9.3 のインストール」
- 55 ページの 「ルートユーザがインストールした OpenText eDirectory の非ルートユーザによる設定」
- 56 ページの「dsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する」
- 63 ページの 「ndsconfig の使用による OpenText eDirectory 9.3 の複数インスタンスの設定」
- 71 ページの「ndsconfig を使用してコンテナ名にドットを使用したツリーに Linux サーバをインストールする」
- 72 ページの 「nmasinst ユーティリティを使用して NMAS を設定する」
- 73 ページの 「非ルートユーザの SNMP 設定」
- 73 ページの 「ログファイルの場所」

OpenText eDirectory での SLP の使用

以前のリリースの OpenText eDirectory では、SLP は OpenText eDirectory のインストール中にインストールされました。しかし、eDirectory 9.1 以上では、OpenText eDirectory をインストールする前に、個別に SLP をインストールする必要があります。

ツリー名の解決に SLP を使用する場合は、このプロトコルのインストールと設定を行う必要があります。SLP ディレクトリエージェント (DA) は安定している必要があります。

1. OpenSLP がインストールされていない場合はインストールします。
2. 画面の指示に従って、SLP のインストールを完了します。
3. 次を実行して SLP を手動で起動します。

   ```
   /etc/init.d/slpd start
   ```

詳細については、「219 ページの付録 C「OpenSLP for OpenText eDirectory の設定」」を参照してください。

同様に、SLP パッケージをアンインストールする場合は、次のようにして SLP を手動で終了する必要があります。

```
/etc/init.d/slpd stop
```

SLP を使用する予定がない (またはできない) 場合は、フラットファイル hosts.nds を使用して、サーバ参照に対するツリー名を解決できます。SLP DA がネットワークにない場合、hosts.nds ファイルを使用して SLP マルチキャストによる遅延を回避できます。

hosts.nds は、OpenText eDirectory アプリケーションによって OpenText eDirectory パーティションおよびサーバを検索するために使用されるスタティックなルックアップテーブルです。hosts.nds ファイルの各行には、各ツリーまたはサーバの以下の情報が記述されます。

- ツリー / サーバ名 : ツリー名は最後はドット (.) で終了します。
- インターネットアドレス :DNS 名の場合もあれば IP アドレスの場合もあります。
- サーバポート : オプションです。インターネットアドレスにコロンが付加されます。

デフォルト以外の NCP ポートで待ち受けしていない限り、ローカルサーバにこのファイルのエントリは必要ありません。

hosts.nds ファイルの構文は次のとおりです。

```
<[partition name.]tree name>.  <host-name/ip-addr>[:<port>]
<server name>  <dns-addr/ip-addr>[:<port>]
```

次に例を示します。

```
# This is an example of a hosts.nds file:
# Tree name             Internet address/DNS Resolvable Name
  CORPORATE.            myserver.mycompany.com
  novell.CORPORATE.     1.2.3.4:524

# Server name           Internet address
  CORPSERVER            myserver.mycompany.com
```

詳細については、hosts.nds のマニュアルページを参照してください。

ツリー名を解決し、eDirectory ツリーが通知されたことを確認するために SLP を使用する場合は、eDirectory および SLP のインストールが完了した後で次のように入力してください。

```
/usr/bin/slptool findattrs services:ndap.novell///(svcname-ws==[treename or *])"
```

たとえば、svcname-ws 属性が値 SAMPLE_TREE と一致するサービスを検索するには、次のコマンドを入力します。

/usr/bin/slptool findattrs services:ndap.novell///(svcname-ws==SAMPLE_TREE)/"

svcname-ws 属性が SAMPLE_TREE として登録されたサービスがある場合、出力は次のようになります。

```
service:ndap.novell:///SAMPLE_TREE
```

svcname-ws 属性が SAMPLE_TREE として登録されたサービスがない場合、何も出力されません。

詳細については、「219 ページの付録 C「OpenSLP for OpenText eDirectory の設定」」を参照してください。

nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする

nds-install ユーティリティを使用して、OpenText™ eDirectory コンポーネントを Linux システムにインストールします。このユーティリティは、Linux プラットフォーム用にダウンロードしたファイルの Setup ディレクトリにあります。このユーティリティでは、インストール対象として選択したコンポーネントに基づいて、必要なパッケージが追加されます。

1 setup ディレクトリで、次のコマンドを入力します。

```
./nds-install
```

コマンドラインに必要なパラメータを入力していない場合、パラメータを要求するプロンプトが nds-install ユーティリティに表示されます。

次の表では、nds-install ユーティリティのパラメータを説明します。

nds-install パラメータ	説明
-h または --help	nds-install のヘルプを表示します。
-i	アップグレード時に DIB が検出された場合、nds-install スクリプトが ndsconfig upgrade コマンドを呼び出さないようにします。
-j	OpenText eDirectory をインストールする前のヘルスチェックオプションをジャンプまたは無効化します。ヘルスチェックの詳細については、「213 ページの付録 B「OpenText eDirectory ヘルスチェック」」を参照してください。

nds-install パラメータ	説明
-m	設定するモジュール名を指定します。新しいツリーを設定するときは、DS モジュールのみを設定できます。DS モジュールを設定してから、add コマンドを使用して、NMAS、LDAP、SAS、SNMP、HTTP の各サービス、および OpenText SecretStore (ss) を追加することができます。モジュール名が指定されていない場合は、すべてのモジュールがインストールされます。
-u	無人インストールモードオプションを指定します。
-f	このオプションは、OpenText eDirectory のいずれかのバージョンに対して、強制的にアップグレード / ダウングレードを行う場合に使用します。

インストールプログラムによって、次の RPM がインストールされます。

eDirectory コンポーネント	インストールされるパッケージ	説明
eDirectory サーバ	◆ novell-NDSbase ◆ novell-NDScommon ◆ novell-NDSmasv ◆ novell-NDSserv ◆ novell-NDSimon ◆ novell-NDSrepair ◆ novell-NDSdexvnt ◆ novell-NOVLsubag ◆ novell-NOVLsnmp ◆ novell-NOVLpkit ◆ novell-NOVLpkis ◆ novell-NOVLpkia ◆ novell-NOVLembox ◆ novell-NOVLlmgnt ◆ novell-NOVLxis ◆ novell-NLDAPsdk ◆ novell-NLDAPbase ◆ novell-NOVLsas ◆ novell-NOVLntls ◆ novell-NOVLnmas ◆ novell-NOVLldif2dib ◆ novell-NOVLncp ◆ novell-eba	指定したサーバに、OpenText eDirectory レプリカサーバがインストールされます。

eDirectory コンポーネント	インストールされるパッケージ	説明
管理ユーティリティ	◆ novell-NOVLice ◆ novell-NDSbase ◆ novell-NLDAPbase ◆ novell-NLDAPsdk ◆ novell-NOVLpkia ◆ novell-NOVLxis ◆ novell-NOVLlmgnt	指定したワークステーションに、OpenText インポート / エクスポート変換ユーティリティおよび LDAP ツール管理ユーティリティがインストールされます。

2 画面の指示に従って、ライセンスファイルの完全パスを入力します。

インストールプラグラムがデフォルトの位置でライセンスファイルを見つけることができなかった場合のみ、ライセンスファイルの完全パスを入力するためのプロンプトが表示されます。デフォルトの位置は、/var、マウントされたライセンスディスク、またはカレントディレクトです。

入力したパスが有効でない場合、正しいパスを入力するようプロンプトが出されます。

3 インストールが完了したら、OpenText eDirectory ユーティリティを現在のセッションで使用するために次の環境変数を更新してエクスポートします。

```
export PATH=$PATH:/opt/novell/eDirectory/bin:/opt/novell/eDirectory/
sbin
```

```
export MANPATH=$MANPATH:/opt/novell/man:/opt/novell/eDirectory/man
```

```
export TEXTDOMAINDIR=/opt/novell/eDirectory/share/locale
```

インストール後に OpenText eDirectory サーバを設定するには、ndsconfig ユーティリティを使用します。

サーバコンポーネントの一部として、NMAS (NetIQ Modular Authentication Service) がインストールされます。デフォルトでは、ndsconfig を使用して NMAS を設定します。また、インストール後に NMAS サーバを設定するには、nmasinst ユーティリティを使用します。NMAS サーバの設定は、ndsconfig を使用して OpenText eDirectory の設定を行ってから実行する必要があります。

デフォルトでは、OpenText eDirectory サーバは FIPS モードで実行されます。FIPS モードを無効にするには、ndsconfig set コマンドで n4u.server.fips_tls=0 を渡し、サーバを再起動します。たとえば、ndsconfig set n4u.server.fips_tls=0 と指定します。

eDirectory 環境で FIPS モードを有効にすると、OpenSSL を使用するすべての eDirectory のアプリケーションまたはモジュールが、常に FIPS モードで OpenSSL を使用します。FIPS モードで eDirectory を動作させると、SSLv3 経由の通信が許可されず、サイファの使用が強度の高いサイファのみに制限されます。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「LDAP オブジェクトを環境設定する」および「*HTTP サーバオブジェクトの設定*」を参照してください。

ndsconfig ユーティリティの詳細については、「181 ページの 「ndsconfig ユーティリティ」」を参照してください。

nmasinst ユーティリティの詳細については、「72 ページの 「nmasinst ユーティリティを使用して NMAS を設定する」」を参照してください。

注 : eDirectory をインストールした後、OpenText eDirectory サーバの DIB ディレクトリは、ウィルス対策ソフトウェアやバックアップソフトウェアのプロセスから除外することをお勧めします。DIB ディレクトリのバックアップには、OpenText eDirectory バックアップツールを使用します。

OpenText eDirectory のバックアップの詳細については、『「OpenText™ eDirectory CE 24.4 (v9.3) eDirectory 管理ガイド』の「」OpenText eDirectory のバックアップと復元」を参照してください。

非ルートユーザによる OpenText eDirectory 9.3 のインストール

非ルートユーザは、tarball を使用して eDirectory 9.3 をインストールできます。

前提条件

- nds-installユーティリティではなくtarballを使用してOpenText eDirectoryをインストールする場合は、NICI をインストールする前に OpenSSL がインストールされていることを確認します。OpenSSL のインストールについては、51 ページの 「OpenSSL のインストール」を参照してください。
- nds-installユーティリティではなくtarballを使用してOpenText eDirectoryをインストールする場合は、NICI がインストールされていることを確認します。NICI のインストールについては、「52 ページの 「NICI のインストール」」を参照してください。
- rpm --nodeps *<SNMP サブエージェント RPM のパス>*コマンドを使って、SNMPサブエージェントがインストールされていることを確認します。
- SLP および SNMP を使用する場合は、それらがルートユーザによってインストールされていることを確認します。
- OpenText eDirectory のインストール先ディレクトリに対する書き込み権。

  管理者以外のユーザについては、「33 ページの 「前提条件」」セクションに示されている権限を持っていることを確認します。

OpenSSL のインストール

OpenSSL のインストールは、NICI をインストールした後、OpenText eDirectory のインストールに進む前に実行する必要があります。

ルートユーザによる OpenSSL のインストール

OpenSSL をインストールするには、次のコマンドを入力します。

```
rpm -ivh openssl_rpm_absolute_path/netiq-openssl-3.0.15.x86_64.rpm
```

NICI のインストール

NICI のインストールは、OpenText eDirectory のインストールに進む前に実行する必要があります。必須 NICI パッケージはシステム全体で使われるため、ルートユーザを使って、必要なパッケージをインストールすることをお勧めします。

OpenText eDirectory 9.3 では、1 つのシステムに 32 ビットと 64 ビットのアプリケーションを共存させることができます。NICI のインストールについては、以降のセクションを参照してください。

ルートユーザによる NICI のインストール

64 ビットの NICI をインストールするには、次のコマンドを入力します。

rpm -ivh NICI_rpm_absolute_path/nici64-3.3.0.0000.x86_64.rpm

NICI をサーバモードに設定するには、ルートユーザとして次のように入力します。

```
/var/opt/novell/nici/set_server_mode64
```

SLES 15 以上でのユーザサービスの設定

これらのプラットフォームで非ルートユーザ向けのサービスをサポートするには、systemd をそのユーザに対しワンタイムアクティビティとして開始します。

非ルートユーザとしてサービスを開始する利点を次に示します。

- システム管理者がサービスを監視できます。
- コンピュータが再起動時にサービスを開始します。

ユーザ固有の systemd を起動するには、OpenText eDirectory がインストールされているマシンにルートユーザとしてログインし、次のコマンドを実行します。

```
systemctl start user@<uid>.service
```

ここで uid は、そのユーザのユーザ ID です。

たとえば、systemctl start user@1001.service と指定します。

持続的な systemd ユーザインスタンスを有効にするには、次のコマンドを実行します。

```
loginctl enable-linger user
```

注 : SLES 15 以上で OpenText eDirectory を設定した後に datadir を新しい場所に移動する場合は、次の手順を実行してください。

- 場所 /usr/lib/systemd/system/ にあるサービスファイル内で、ndsd.pid ファイルを新しい場所に更新します。

  たとえば、nds.conf ファイルがもともと /etc/opt/novell/eDirectory に置かれている場合は、次のようにサンプルのサービスファイルが作成されます。

  /usr/lib/systemd/system/ndsdtmpl-etc-opt-novell-eDirectory-conf-ds.conf@.service。

- systemctl daemon-reload コマンドを使用して、デーモンを再ロードします。
- OpenText eDirectory サーバを再起動します。

OpenText eDirectory のインストール

1. ssh を使用し、非ルートユーザとして、OpenText eDirectory がインストールされているコンピュータにログインします。
2. OpenText eDirectory をインストールするディレクトリに移動します。
3. 次のコマンドを実行して、tar ファイルを展開します。

   ```
   tar xvf /tar_file_name
   ```

   etc ディレクトリと opt ディレクトリが作成されます。
4. 次を実行して、パスをエクスポートします。
   - **環境変数を手動でエクスポートするには、次のコマンドを入力します。**

     ```
     export LD_LIBRARY_PATH=custom_location/eDirectory/opt/novell/
     eDirectory/lib64:custom_location/eDirectory/opt/novell/eDirectory/
     lib64/nds-modules:custom_location/eDirectory/opt/novell/
     lib64:$LD_LIBRARY_PATH

     export PATH=custom_location/eDirectory/opt/novell/eDirectory/
     bin:custom_location/eDirectory/opt/novell/eDirectory/sbin:/opt/
     novell/eDirectory/bin:$PATH

     export MANPATH=custom_location/eDirectory/opt/novell/
     man:custom_location/eDirectory/opt/novell/eDirectory/man:$MANPATH

     export TEXTDOMAINDIR=custom_location/eDirectory/opt/novell/
     eDirectory/share/locale:$TEXTDOMAINDIR
     ```

     ndspath スクリプトを使って環境変数をエクスポートするには、次の手順を実行します。

     手動でパスをエクスポートしない場合は、ndspath スクリプトをユーティリティの前に指定します。
     - 次のように、必要なユーティリティを実行します。

       ```
       custom_location/eDirectory/opt/novell/eDirectory/bin/ndspath
       utility_name_with_parameters
       ```

     - または、OpenText eDirectory tarball の場所を明示的に ndspath ユーティリティに指定して、現在のシェル内にパスをエクスポートすることもできます。

       ```
       . custom_location/eDirectory/opt/novell/eDirectory/bin/ndspath
       custom_location/eDirectory
       ```

       注 : 上記のコマンドは、必ず *custom_location*/eDirectory/opt ディレクトリから入力してください。

上記のコマンドを入力したら、通常どおりにユーティリティを実行します。

- プロファイル内のスクリプト (bashrc、または同様のスクリプト) を呼び出します。こうすることで、ログインするか新しいシェルを開けば、直接ユーティリティを使い始めることができます。

5 通常の方法で OpenText eDirectory を設定します。

OpenText eDirectory は次のいずれかの方法で設定できます。

- 次のとおり、ndsconfig ユーティリティを使用します。

```
ndsconfig new [-t <treename>] [-n <server_context>] [-a <admin_FDN>]
[-w <admin password>] [-i] [-S <server_name>] [-d <path_for_dib>] [-
m <module>] [e] [-L <ldap_port>] [-l <SSL_port>] [-o <http_port>] -
O <https_port>] [-p <IP address:[port]>] [-c] [-b <port_to_bind>] [-
B <interface1@port1>, <interface2@port2>,..] [-D <custom_location>]
[--config-file <configuration_file>] [--configure-eba-now <yes/no>]
```

次に例を示します。

```
ndsconfig new -t mary-tree -n novell -a admin.novell -S linux1 -d /
home/mary/inst1/data -b 1025 -L 1026 -l 1027 -o 1028 -O 1029 -D /
home/mary/inst1/var --config-file /home/mary/inst1/nds.conf --
configure-eba-now yes
```

入力するポート番号は、1024 ～ 65535 の範囲内にする必要があります。1024 より小さいポート番号は通常、スーパユーザと標準アプリケーション用に予約されています。そのため、eDirectory アプリケーションには、デフォルトのポート 524 は使用できません。

これが原因で、次のアプリケーションで問題が発生する可能性があります。

- ターゲットサーバポートを指定するオプションがないアプリケーション。
- NCP を使用し、ポート 524 でルートとして動作する古いアプリケーション。

- ndsmanage ユーティリティを使用して、新しいインスタンスを設定します。詳細については、「67 ページの 「ndsmanage によるインスタンスの作成」」を参照してください。

拡張バックグラウンド認証を有効にするには、ndsconfig upgrade コマンドの --configure-eba-now スイッチで *yes* を指定します。有効にしない場合は、*no* を指定して後から設定します。--configure-eba-now スイッチを ndsconfig コマンドに渡さなかった場合は、選択を求めるプロンプトが OpenText eDirectory から出されます。デフォルトでは、この設定は no です。

画面の指示に従って、設定を完了します。

詳細については、「56 ページの 「dsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する」」を参照してください。

注 : OpenText eDirectory をインストールした後、OpenText eDirectory サーバの DIB ディレクトリは、ウィルス対策ソフトウェアやバックアップソフトウェアのプロセスから除外することをお勧めします。DIB ディレクトリのバックアップには、OpenText eDirectory バックアップツールを使用します。

OpenText eDirectory のバックアップの詳細については、『「OpenText™ eDirectory CE 24.4 (v9.3) eDirectory 管理ガイド』の「」OpenText eDirectory のバックアップと復元」を参照してください。

ルートユーザがインストールした OpenText eDirectory の非ルートユーザによる設定

非ルートユーザがインストール tarball を使用して OpenText eDirectory をインストールおよび設定する場合、いくつかのセキュリティ上の脅威が関連しています。OpenText eDirectory を設定する非ルートユーザがバイナリを所有しているため、外部からの攻撃に対して脆弱になります。セキュリティ対策としては、ルートユーザとしてインストーラスクリプトを使用して OpenText eDirectory をインストールした後、非ルートユーザとして ndsd サービスを実行することをお勧めします。非ルートユーザは非特権ユーザである必要があり、ルートによるインストール中に配置される OpenText eDirectory バイナリに対する書き込み許可を所有または持っていてはなりません。セキュリティの脅威が発生する可能性を減らすためです。これにより、nds-install ユーティリティを使用して OpenText eDirectory サーバを保守またはアップグレードできるようにもなります。

1 ルートユーザとして、nds-install ユーティリティを使用して OpenText eDirectory コンポーネントを Linux システムにインストールします。

```
./nds-install
```

ライセンス契約を受諾します。

詳細については、48 ページの 「nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする」を参照してください。

2 次を実行して、パスをエクスポートします。

- **環境変数を手動でエクスポートするには、次のコマンドを入力します。**

```
export LD_LIBRARY_PATH=/opt/novell/eDirectory/lib64:/opt/novell/
eDirectory/lib64/nds-modules:/opt/novell/lib64:$LD_LIBRARY_PATH

export PATH=/opt/novell/eDirectory/bin:/opt/novell/eDirectory/
sbin:/opt/novell/eDirectory/bin:$PATH

export MANPATH=/opt/novell/man:/opt/novell/eDirectory/man:$MANPATH

export TEXTDOMAINDIR=/opt/novell/eDirectory/share/
locale:$TEXTDOMAINDIR
```

ndspath スクリプトを使って環境変数をエクスポートするには、次の手順を実行します。

手動でパスをエクスポートしない場合は、ndspath スクリプトをユーティリティの前に指定します。

- 次のように、必要なユーティリティを実行します。

```
/opt/novell/eDirectory/bin/ndspath utility_name_with_parameters
```

- 次のとおり、現在のシェル内のパスをエクスポートします。

```
. /opt/novell/eDirectory/bin/ndspath
```

上記のコマンドを入力したら、通常どおりにユーティリティを実行します。

- プロファイル内のスクリプト (bashrc、または同様のスクリプト) を呼び出します。こうすることで、ログインするか新しいシェルを開けば、直接ユーティリティを使い始めることができます。

3 非ルートユーザとして、OpenText eDirectory のルートインストールを実行した後、ndsconfig または ndsmanage ユーティリティを使用して eDirectory サーバを設定します。次にサンプルコマンドを示します。

```
ndsconfig new [-t <treename>] [-n <server_context>] [-a <admin_FDN>] [-w <admin password>] [-i] [-S <server_name>] [-d <path_for_dib>] [-m <module>] [e] [-L <ldap_port>] [-l <SSL_port>] [-o <http_port>] -O <https_port>] [-p <IP address:[port]>] [-c] [-b <port_to_bind>] [-B <interface1@port1>, <interface2@port2>,..] [-D <custom_location>] [--config-file <configuration_file>] [--configure-eba-now <yes/no>]
```

次に例を示します。

```
ndsconfig new -t mary-tree -n novell -a admin.novell -S linux1 -d /home/mary/inst1/data -b 1025 -L 1026 -l 1027 -o 1028 -O 1029 -D /home/mary/inst1/var --config-file /home/mary/inst1/nds.conf --configure-eba-now yes
```

注

- OpenText eDirectory は、標準ポートを使用して設定することはできません。ポートの範囲は 1024 ～ 65535 である必要があります。
- 設定が失敗しない限り、非ルートユーザは OpenText eDirectory の環境設定ファイルが保存されている場所に対して必要な権限を持っているはずです。

ユーザ固有の systemd を起動するには、OpenText eDirectory がインストールされているマシンにルートユーザとしてログインし、次のコマンドを実行します。

```
systemctl start user@<uid>.service
```

ここで uid は、そのユーザのユーザ ID です。

たとえば、systemctl start user@1001.service と指定します。

持続的な systemd ユーザインスタンスを有効にするには、次のコマンドを実行します。

```
loginctl enable-linger user
```

dsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する

OpenText eDirectory をインストールした後、ndsconfig ユーティリティを使って OpenText eDirectory レプリカサーバを設定します。ndsconfig ユーティリティを使用するには、管理者の権利を持っている必要があります。引数付きでこのユーティリティを使用した場合は、すべての引数が確認され、管理者の権利を持つユーザのパスワード入力を要求するプロンプトが表示されます。引数なしで ndsconfig ユーティリティを使用した場合は、この

ユーティリティに関する説明と利用可能なオプションが表示されます。このユーティリティでは、OpenText eDirectory レプリカサーバを削除したり、OpenText eDirectory サーバの現在の設定を変更したりすることもできます。詳細については、181 ページの「ndsconfig ユーティリティ」を参照してください。

注 : OpenText eDirectory 9.3.0 以降では、マスタサーバが 9.3.0 の場合、EBA の有無にかかわらず、下位バージョンの OpenText eDirectory をレプリカとして追加することはサポートされません。

特定のロケールで OpenText eDirectory を設定するための前提条件

特定のロケールに OpenText eDirectory を設定する場合は、OpenText eDirectory の設定を行う前に、そのロケールに LC_ALL および LANG をエクスポートする必要があります。たとえば、OpenText eDirectory を日本ロケールに設定する場合は、次のコマンドを入力します。

export LC_ALL=ja

export LANG=ja

新しいツリーの作成

使用する構文は次のとおりです。

```
ndsconfig new [-m <modulename>] [-i] [-S <server name>] [-t <tree_name>] [-
n <server context>] [-d <path_for_dib>][-P <LDAP URL(s)>] [-L <ldap_port>]
[-l <ssl_port>] [-o http port] [-O https port] [-e] -a <admin FDN> [-R] [-
c] [-w <admin password>] [-b <port to bind>] [-B <interface1@port1,
interface2@port2,..>] [-D <path_for_data>] [--config-file <configuration
file>] [--configure-eba-now <yes/no>] [--pki-default-rsa-keysize <2048/
4096/8192>] [--pki-default-ec-curve <P256/P384/P521>] [--pki-default-cert-
life <in years>]
```

指定したツリー名とコンテキストの新しいツリーがインストールされます。

tree_name、*admin FDN*、および *server FDN* の変数には、文字数制限があります。これらの変数に使用できる最大文字数は次のとおりです。

- *tree_name*: 32 文字
- *admin-FDN*: 255 文字
- *server FDN*: 255 文字

重要 : OpenText eDirectory では、NCP サーバオブジェクトの FDN を最大 256 文字までで設定できますが、このオブジェクトの長さに基づいて他のより長いオブジェクトが作成されるため、この変数をずっと小さい値に制限することをお勧めします。

コマンドラインにパラメータが指定されていない場合、指定されていない各パラメータに値を入力するよう求めるプロンプトが ndsconfig によって表示されます。

また、次の構文も使用できます。

```
ndsconfig def [-t <treename>] [-n <server context>] [-a <admin FDN>] [-w
<admin password>] [-c] [-i] [-S <server name>] [-d <path for dib>] [-m
<module>] [-e] [-L <ldap port>] [-l <SSL port>] [-o <http port>] [-O <https
port>] [-D <custom_location>] [--config-file <configuration_file>] [--
configure-eba-now <yes/no>]
```

指定したツリー名とコンテキストの新しいツリーがインストールされます。コマンドラインにパラメータが指定されていない場合、ndsconfig によって、指定されていない各パラメータにデフォルト値が適用されます。

たとえば、新しいツリーを作成するには、次のようにコマンドを入力します。

```
ndsconfig new -t corp-tree -n o=company -a cn=admin.o=company
```

注 : 新しいツリーの作成時に使用する OpenText eDirectory 9.2 の ndsconfig コマンドに --enable-pbkdf2 という名前の新しいオプションが追加されました。このオプションが設定されている場合は、パスワードポリシーが作成され、ツリー全体に自動的に割り当てられます。このパスワードポリシーにより、ツリー内のすべてのユーザに対して、NDS パスワードの PBKDF2 パスワードとの同期が可能になります。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「非可逆パスワードストレージを理解する」を参照してください。

デフォルトサーバ証明書のデフォルトのパラメータを指定する

OpenText eDirectory には、新しい OpenText eDirectory ツリーを設定する際に、CA 証明書とデフォルトサーバ証明書のデフォルト RSA キーサイズ、楕円曲線、および有効期限を指定するオプションが用意されています。次のコマンドを使用して、ndsconfig new を使用して新しい OpenText eDirectory ツリーを設定する際に、CA およびデフォルトサーバの証明書にデフォルトパラメータを指定できます。

- **pki-default-rsa-keysize:** RSA 証明書のキーサイズを指定します。使用可能な値は 2048、4096、および 8192 ビットです。
- **pki-default-ec-curve:** EC 証明書の曲線の制限を指定します。使用可能な値は P256、P384、および P521 です。
- **pki-default-cert-life:** 証明書の有効期限を年数で指定します。

これらの属性は、新しい OpenText eDirectory サーバのインストール中に ndsconfig new で設定できます。

ここで指定される値は、新しいツリーが構成されるときに組織の CA オブジェクトの対応する属性に設定されます。

詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*組織の認証局オブジェクトを作成する*」を参照してください。

既存のツリーにサーバを追加する

使用する構文は次のとおりです。

```
ndsconfig add [-t <treename>] [-n <server context>] [-a <admin FDN>] [-w
<admin password>] [-e] [-P <LDAP URL(s)>][-L <ldap port>] [-l <SSL port>]
[-o <http port>] [-O <https port>] [-S <server name>] [-d <path for dib>]
[-m <module>] [-p <IP address:[port]>] [-R] [-c] [-b <port to bind>] [-B
<interface1@port1>, <interface2@port2>,..] [-D <custom_location>] [--
config-file <configuration_file>] [-E] [--configure-eba-now <yes/no>]
```

OpenText eDirectory により、指定したコンテキストで既存のツリーにサーバが追加されます。サーバオブジェクトの追加先として指定したコンテキストが存在しない場合は、ndsconfig によって該当するコンテキストが作成され、サーバが追加されます。

拡張バックグラウンド認証 (EBA) を有効にするには、ndsconfig upgrade コマンドの --configure-eba-now スイッチで *yes* を指定します。有効にしない場合は、*no* を指定して後から設定します。--configure-eba-now スイッチを ndsconfig コマンドに渡さなかった場合は、選択を求めるプロンプトが OpenText eDirectory から出されます。デフォルトでは、この設定は *no* です。

EBA を有効にしたセカンダリサーバをツリーに追加するには、ツリーで EBA CA を設定しておく必要があります。EBA CA が存在しない場合は、EBA を有効にせずにサーバを追加してから、EBA CA をホストするようにサーバをアップグレードします。この操作を行わないと、セカンダリサーバの設定が失敗します。

既存のツリーへ eDirectory をインストールした後で、LDAP およびセキュリティサービスを追加することもできます。

たとえば、新しいツリーをサーバに追加するには、次のようにコマンドを入力します。

```
ndsconfig add -t corp-tree -n o=company -a cn=admin.o=company -S srv1
```

-E オプションを使用して、追加するサーバの暗号化レプリケーションを有効にできます。暗号化レプリケーションの詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*暗号化レプリケーション*」を参照してください。

注 : OpenText eDirectory 9.2.7 リリースで、暗号化レプリケーションのサポートは非推奨になりました。

ツリーからサーバオブジェクトとディレクトリサービスを削除する

使用する構文は次のとおりです。

```
ndsconfig rm [-a <admin FDN>] [-w <admin password>] [-p <IP
address:[port]>] [-c]
```

サーバから eDirectory およびデータベースが削除されます。

注 : iMonitor を使用して作成した HTML ファイルは削除されません。これらのファイルは、OpenText eDirectory を削除する前に、/var/opt/novell/eDirectory/data/dsreports から手動で削除する必要があります。

たとえば、OpenText eDirectory サーバオブジェクトとディレクトリサービスをツリーから削除するには、次のコマンドを入力します。

```
ndsconfig rm -a cn=admin.o=company
```

ndsconfig ユーティリティパラメータ

ndsconfig のパラメータ	説明
new	新しい OpenText eDirectory ツリーを作成します。コマンドラインにパラメータが指定されていない場合、指定されていない各パラメータに値を入力するよう求めるプロンプトが ndsconfig によって表示されます。
def	新しい OpenText eDirectory ツリーを作成します。コマンドラインにパラメータが指定されていない場合、ndsconfig によって、指定されていない各パラメータにデフォルト値が適用されます。
add	既存のツリーにサーバを追加します。また、既存のツリーで OpenText eDirectory の設定が完了した後、LDAP および SAS サービスを追加します。
rm	サーバオブジェクトとディレクトリサービスをツリーから削除します。 **注 :** このオプションを指定しても、キーマテリアルオブジェクトは削除されません。これらのオブジェクトは、手動で削除する必要があります。
アップグレード	OpenText eDirectory を、使用中のバージョンよりも新しいバージョンにアップグレードします。
-i	新しいツリーを設定するときに、同じ名前のツリーが存在するかのチェック結果を無視します。複数の同じ名前を持つツリーが存在できます。
-S *server name*	サーバ名を指定します。サーバ名にはドットも使用できます (例 :netiq.com)。ndsconfig はコマンドラインユーティリティのため、名前にドットを含むコンテナを使用する場合はそれらのドットをエスケープする必要があります。また、ドットを含むコンテキストが含まれたパラメータは二重引用符で囲む必要があります。 たとえば、O の名前として netiq.com を使用している Linux サーバに新しい OpenText eDirectory ツリーをインストールするには、次のコマンドを使用します。 ndsconfig new -a "admin.novell\\.com" -t netiq_tree -n "OU=servers.O=netiq\\.com" 管理者名のパラメータ、コンテキストとサーバコンテキストのパラメータは、二重引用符で囲みます。netiq.com の「.」だけを「\\」(円記号) 文字を使ってエスケープします。このフォーマットは、既存のツリーにサーバをインストールする場合にも使用できます。 **注 :** 名前の最初にドットを使用することはできません。たとえば、ドット (.) で始まる「.novell」という名前のサーバをインストールすることはできません。
-t *treename*	サーバの追加先のツリー名です。最大で 32 文字の名前を付けることができます。このパラメータを指定しない場合、ndsconfig は /etc/opt/novell/eDirectory/conf/nds.conf ファイル内の n4u.nds.tree-name パラメータに指定されているツリー名を採用します。デフォルトのツリー名は $LOGNAME-$HOSTNAME-NDStree です。

ndsconfig のパラメータ	説明
-n *server context*	サーバオブジェクトを追加するサーバのコンテキストを指定します。最大で 64 文字の名前を付けることができます。このコンテキストが指定されていない場合、ndsconfig は /etc/opt/novell/eDirectory/conf/nds.conf ファイルの n4u.nds.server-context 環境設定パラメータに指定されているコンテキストを採用します。サーバコンテキストはタイプつきの形式で指定する必要があります。デフォルトのコンテキストは org です。
-d *path for dib*	データベースファイルの格納先になる場所のディレクトリパスです。
-r	このオプションを使用すると、サーバにすでに追加されているサーバの数にかかわらず、サーバのレプリカが強制的に追加されます。
-L *ldap_port*	LDAP サーバの TCP ポート番号を指定します。デフォルトのポートである 389 が使用中の場合は、新しいポート番号を要求するプロンプトが表示されます。
-l *ssl_port*	LDAP サーバの SSL ポート番号を指定します。デフォルトのポートである 636 が使用中の場合は、新しいポート番号を要求するプロンプトが表示されます。
-a *admin FDN*	サーバオブジェクトとディレクトリサービスの作成先のコンテキストに対するスーパバイザ権を持つ、ユーザオブジェクトの完全識別名を指定します。admin 名はタイプつきの形式で指定する必要があります。最大で 64 文字の名前を付けることができます。デフォルトの admin 名は admin.org です。
-e	LDAP オブジェクトのクリアテキストパスワードを有効にします。
-m *modulename*	設定するモジュール名を指定します。新しいツリーを設定するときは、DS モジュールのみを設定できます。DS モジュールを設定してから、add コマンドを使用して、NMAS、LDAP、SAS、SNMP、HTTP の各サービス、および OpenText SecretStore (ss) を追加することができます。モジュール名が指定されていない場合は、すべてのモジュールがインストールされます。 **注 :** nds-install を使った OpenText eDirectory のアップグレード中に SecretStore を設定しない場合は、このオプションに no_ss という値を渡します。たとえば、nds-install '-m no_ss' などです。
-o	HTTP クリアポート番号を指定します。
-O	HTTP セキュアポート番号を指定します。
-p *<IP アドレス :[ポート]>*	このオプションは、セカンダリサーバをツリーに追加する場合 (add コマンド) に使われます。このサーバを追加するパーティションのレプリカを保持するリモートホストの IP アドレスを指定します。デフォルトのポート番号は 524 です。これにより、SLP ルックアップが回避されるため、ツリーのルックアップが高速化されます。
-R	デフォルトでは、サーバの追加先のパーティションのレプリカは、ローカルサーバに複製されます。このオプションを使用すると、レプリカをローカルサーバに追加することは許可されません。

ndsconfig のパラメータ	説明
-c	このオプションを使用すると、ndsconfig 操作中 (操作を続行するための yes/no の選択など) のプロンプトや、競合が発生した場合にポート番号の再入力を求めるプロンプトなどが表示されなくなります。コマンドラインで必須パラメータを渡さなかった場合のみ、必須パラメータの入力を求めるプロンプトが表示されます。
-w < *管理パスワード*>	このオプションを使用すると、管理ユーザパスワードをクリアテキストで渡すことができます。 **注 :** パスワードをクリアテキストで渡すと、安全が確保できないため、このオプションは推奨されません。
-E	追加するサーバに対して暗号化レプリケーションを有効にします。
-j	OpenText eDirectory をインストールする前のヘルスチェックオプションをジャンプまたは無効化します。
-b *port to bind*	特定のインスタンスが監視に使用するデフォルトのポート番号を設定します。これにより、n4u.server.tcp-port と n4u.server.udp-port にデフォルトのポート番号が設定されます。-b オプションを使用して NCP ポートが渡されると、デフォルトのポートと見なされ、それに応じて TCP と UDP のパラメータが更新されます。 **注 :** -b と -B は排他的に使われます。

-B *interface1@port1, interface2@port2,...*	ポート番号を IP アドレスまたはインタフェースとともに指定します。次に例を示します。
	
	`-B eth0@524`
	
	または
	
	`-B 100.1.1.2@524`
	
	注 : -b と -B は互いに排他的です。管理者がこのオプションに @<port-number> のみを指定した場合、NCP、http、および https サービスは使用可能なすべてのインターフェイスにバインドされます。

--config-file *configuration file*	nds.conf 環境設定ファイルを保存するための絶対パスとファイル名を指定します。たとえば、環境設定ファイルを /etc/opt/novell/eDirectory/ ディレクトリに保存する場合には、--config-file /etc/opt/novell/eDirectory/nds.conf と入力します。
-P <*LDAP URL*>	LDAP URL で、LDAP サーバオブジェクト上の LDAP インタフェースを設定できるようにします。 例 : -P ldap://1.2.3.4:1389,ldaps://1.2.3.4:1636
-D *path_for_data*	指定したパスに data、dib、および log のディレクトリを作成します。

ndsconfig のパラメータ	説明
set valuelist	指定した OpenText eDirectory 環境設定パラメータに対して値を設定します。ツリーを設定する前に、ブートストラップパラメータの設定に使用します。環境設定パラメータを変更した場合、新しい値を有効にするには ndsd を再起動する必要があります。ただし、環境設定パラメータによっては ndsd を再起動する必要がない場合があります。 再起動の必要のないパラメータは次のとおりです。 ◆ n4u.nds.inactivity-synchronization-interval ◆ n4u.nds.synchronization-restrictions ◆ n4u.nds.janitor-interval ◆ n4u.nds.backlink-interval ◆ n4u.nds.drl-interval ◆ n4u.nds.flatcleaning-interval ◆ n4u.nds.server-state-up-threshold ◆ n4u.nds.heartbeat-schema ◆ n4u.nds.heartbeat-data ◆ n4u.server.fips_tls ◆ n4u.server.eba_enabled
get help paramlist	指定した OpenText eDirectory 環境設定パラメータに関するヘルプを表示します。パラメータリストが指定されていない場合は、ndsconfig はすべての OpenText eDirectory 環境設定パラメータに関するヘルプ文字列を表示します。
set valuelist	指定した OpenText eDirectory 環境設定パラメータに対して値を設定します。ツリーを設定する前に、ブートストラップパラメータの設定に使用します。 環境設定パラメータを変更した場合、新しい値を有効にするには ndsd を再起動する必要があります。
get paramlist	指定した OpenText eDirectory 環境設定パラメータの現在の値を表示します。パラメータリストが指定されていない場合は、ndsconfig はすべての OpenText eDirectory 環境設定パラメータを表示します。
configure-eba-now	拡張バックグラウンド認証用に OpenText eDirectory サーバを設定するには、このスイッチを使用します。

ndsconfig の使用による OpenText eDirectory 9.3 の複数インスタンスの設定

単一のホスト上で OpenText eDirectory 9.3 の複数インスタンスを設定できます。OpenText eDirectory 9.3 では複数インスタンスの機能がサポートされるため、次の設定が可能です。

- 1 台のホスト上に複数インスタンスの OpenText eDirectory を設定する

- 1 台のホスト上に複数のユーザ用に複数のツリーを設定する
- 1 台のホスト上に同じツリーまたはパーティションの複数のレプリカを設定する

警告 : 同じユーザに対して複数のツリーを設定することはできません。1 ユーザに対して複数のツリーにサーバのインスタンスを設定することはできません。複数のツリーにサーバを設定する場合は、別々のユーザアカウントを使用してください。

次の表に、複数インスタンスをサポートするプラットフォームを示します。

機能	Linux	Windows
複数インスタンスのサポート	✓	✗

複数インスタンスの設定方法は、1 つのインスタンスを複数回設定する場合と同様です。各インスタンスは、次のように固有のインスタンス識別子を持つ必要があります。

- 異なるデータとログファイルの場所

  ndsconfig --config-file、-d、および -D オプションを使用して、これを実行できます。
- リスン対象インスタンスの固有のポート番号

  ndsconfig の -b と -B オプションを使って、これを実行できます。
- インスタンスの一意なサーバ名

  ndsconfig -S server name オプションを使用して、これを行えます。

  重要 : OpenText eDirectory の設定中に、デフォルトの NCP サーバ名がホストサーバ名として設定されます。複数のインスタンスを設定する場合、NCP サーバ名を変更する必要があります。ndsconfig のコマンドラインオプションである -S *<server_name>* を使って、別のサーバ名を指定します。

  同じツリーまたは複数の異なるツリーのどちらであっても、複数インスタンスを設定する場合は、NCP サーバ名が固有でなければなりません。

複数インスタンスの必要性

複数インスタンスは、次のことを行う必要性から提供されるようになりました。

- OpenText eDirectory のインスタンスを複数設定することによって、ハイエンドのハードウェアを活用する。
- 必要なハードウェアに投資する前に、1 台のホスト上でセットアップをテスト運用する。

複数インスタンスを展開する場合のシナリオ

同じツリーまたは複数のツリーに属する複数インスタンスは、次のようなシナリオで効果的に使用できます。

大企業における eDirectory の使用

- 大企業では、OpenText eDirectory サービスの負荷分散と高可用性を提供することができます。

  たとえば、ポート 1524、2524、および 3524 で LDAP サービスを実行するレプリカサーバ 3 台がある場合、OpenText eDirectory の新しいインスタンスを設定し、新しいポート 636 で高い可用性の LDAP サービスを提供できます。
- 1 台のホストに複数インスタンスを設定すると、組織内の複数の部門にまたがってハイエンドのハードウェアを活用できます。

評価用セットアップにおける OpenText eDirectory の使用

- **大学 :** 大勢の熱心なユーザ (学生) が、複数インスタンスを使用して 1 台のホストから OpenText eDirectory を評価できます。
- **eOpenText Directory 管理のトレーニング :**
  - 参加者は、複数インスタンスを使用して、実際に管理を行ってみることができます。
  - 講師は、1 台のホストを使用してクラスの受講者に教えることができます。各受講者に専用のツリーを用意できます。

複数インスタンスの使用

OpenText eDirectory 9.3 によって、複数インスタンスの設定が容易になります。複数インスタンスを効果的に使用するためには、セットアップを慎重に計画してから、複数インスタンスを設定する必要があります。

- 70 ページの 「セットアップの計画」
- 66 ページの 「複数インスタンスを設定する」

セットアップの計画

この機能を有効に使用するためには、OpenText eDirectory のインスタンスを複数計画し、各インスタンスが、ホスト名、ポート番号、サーバ名、または環境設定ファイルのように、確定的なインスタンス識別子を持つように設定することをお勧めします。

複数インスタンスの設定時には、次のことについて計画したかどうかを確認する必要があります。

- 環境設定ファイルの場所
- 変数データの場所 (ログファイルなど)
- DIB の場所
- NCP インタフェース、各インスタンスを識別する一意のポート、および他のサービスのポート (LDAP、LDAPS、HTTP、HTTPS ポートなど)
- 各インスタンスの一意なサーバ名

複数インスタンスを設定する

複数インスタンスの OpenText eDirectory は、ndsconfig ユーティリティを使用して設定できます。次の表に、複数インスタンスの設定時に指定する必要がある ndsconfig オプションを示します。

注 : すべてのインスタンスは同じサーバキー (NICI) を共有します。

オプション	説明
--config-file	nds.conf 環境設定ファイルを保存するための絶対パスとファイル名を指定します。 たとえば、環境設定ファイルを /etc/opt/novell/eDirectory/ ディレクトリに保存する場合には、--config-file /etc/opt/novell/eDirectory/nds.conf を使用します。
-b	新しいインスタンスが監視するときのポート番号を指定します。 **注 :** -b と -B は排他的に使われます。
-B	ポート番号を IP アドレスまたはインタフェースとともに指定します。次に例を示します。 -B eth0@524 または -B 100.1.1.2@524 **注 :** -b と -B は排他的に使われます。
-D	data、dib、および log のディレクトリを、新しいインスタンス用に指定したパスに作成します。
S	サーバ名を指定します。

上記のオプションを使用して、OpenText eDirectory の新しいインスタンスを設定できます。

ndsmanage ユーティリティを使用して、新しいインスタンスを設定することもできます。詳細については、「67 ページの 「ndsmanage によるインスタンスの作成」」を参照してください。

複数インスタンスを管理する

このセクションでは、次の情報を紹介します。

- 67 ページの 「ndsmanage ユーティリティ」
- 69 ページの 「特定のインスタンスの識別」
- 69 ページの 「特定のインスタンスに対するユーティリティの呼び出し」

ndsmanage ユーティリティ

ndsmanage ユーティリティを使用すると、次の操作を実行できます。

- 設定したインスタンスの表示
- 新しいインスタンスの作成
- 選択したインスタンスに対する次の操作の実行 :
  - サーバ上にあるレプリカの表示
  - インスタンスの開始
  - インスタンスの停止
  - インスタンスに対する DSTrace (ndstrace) の実行
  - インスタンスの設定解除
- すべてのインスタンスの開始と停止

インスタンスの表示

次の表で、eDirectory インスタンスを表示する方法について説明します。

表 3-1 *インスタンスを表示するための ndsmanage の使用*

構文	説明
ndsmanage	設定したすべてのインスタンスを表示します。
ndsmanage -a|--all	eDirectory の特定のインストールを使用しているすべてのユーザのインスタンスを表示します。
ndsmanage *ユーザ名*	特定のユーザによって設定されたインスタンスを表示します。

各インスタンスについて、次のフィールドが表示されます。

- 環境設定ファイルのパス
- サーバの FDN およびポート
- ステータス (インスタンスがアクティブか非アクティブか)

注 : このユーティリティは、単一のバイナリに対して設定されたすべてのインスタンスを表示します。

ndsmanage によるインスタンスの作成

ndsmanage を使用して新しいインスタンスを作成するには、次の手順を実行します。

1 次のコマンドを入力します。

```
ndsmanage
```

2 新しいインスタンスを作成するには、「c」と入力します。

新しいツリーを作成するか、既存のツリーにサーバを追加できます。画面の指示に従って、新しいインスタンスを作成します。

特定のインスタンスに対する操作の実行

各インスタンスについて、次の操作を実行できます。

下記の操作以外に、選択したインスタンスに対して DSTrace を実行することもできます。

特定のインスタンスの開始

自分が設定したインスタンスを開始するには、次の操作を実行します。

1 次のように入力します。

```
ndsmanage
```

2 開始するインスタンスを選択します。

メニューが拡張し、特定のインスタンスに対して実行可能なオプションが表示されます。

図 3-1 *ndsmanage ユーティリティのインスタンスオプションの出力画面*



3 インスタンスを開始するには、「s」と入力します。

または、コマンドプロンプトに次のコマンドを入力することもできます。

ndsmanage start --config-file *configuration_file_of_the_instance_configured_by_you*

特定のインスタンスの停止

自分が設定したインスタンスを停止するには、次の操作を実行します。

1 次のように入力します。

```
ndsmanage
```

2 停止するインスタンスを選択します。

メニューが拡張し、特定のインスタンスに対して実行可能なオプションが表示されます。詳細については、「(68 ページ) ndsmanage ユーティリティのインスタンスオプションの出力画面」を参照してください。

3 インスタンスを停止するには、「k」と入力します。

または、コマンドプロンプトに次のコマンドを入力することもできます。

ndsmanage stop --config-file *configuration_file_of_the_instance_configured_by_you*

インスタンスの設定解除

インスタンスの設定を解除するには、次の手順を実行します。

1. 次のように入力します。

   ```
   ndsmanage
   ```

2. 設定解除するインスタンスを選択します。

   メニューが拡張し、特定のインスタンスに対して実行可能なオプションが表示されます。詳細については、「(68 ページ) ndsmanage ユーティリティのインスタンスオプションの出力画面」を参照してください。

3. インスタンスを設定解除するには、「d」と入力します。

すべてのインスタンスの開始と停止

自分が設定したすべてのインスタンスを開始および停止できます。

すべてのインスタンスの開始

自分が設定したすべてのインスタンスを開始するには、コマンドプロンプトで次のコマンドを入力します。

ndsmanage startall

特定のインスタンスを開始するには、「68 ページの 「特定のインスタンスの開始」」を参照してください。

特定のインスタンスの識別

複数インスタンスの設定中に、ホスト名、ポート番号、および一意な環境設定ファイルのパスを、各インスタンスに割り当てます。このホスト名とポート番号が、インスタンスの識別子になります。

ほとんどのユーティリティには、特定のインスタンスを指定することができる「-h *ホスト名*:*ポート*」オプションまたは「--config-file *環境設定ファイルの場所*」オプションが用意されています。詳細については、ユーティリティのマニュアルページを参照してください。

特定のインスタンスに対するユーティリティの呼び出し

特定のインスタンスに対してユーティリティを実行する場合は、ユーティリティのコマンドにインスタンスの識別子を含める必要があります。インスタンスの識別子になるのは、環境設定ファイルのパス、ホスト名、およびポート番号です。「--config-file *環境設定ファイルの場所*」または「-h *ホスト名*:*ポート*」を使用すると、特定のインスタンスに対してユーティリティを実行できます。

コマンドにインスタンス識別子を指定しないと、ユーザが所有するさまざまなインスタンスが表示され、ユーティリティの実行対象にするインスタンスを選択するように求められます。

たとえば、--config-file オプションを指定して特定のユーティリティに対して DSTrace を実行する場合は、次のように入力します。

```
ndstrace --config-file configuration_filename_with_location
```

複数インスタンスのシナリオ

非ルートユーザである Mary が、1 台のホストマシン上で、1 つのバイナリに対し 2 つのツリーを設定しようとしています。

セットアップの計画

Mary は次のインスタンス識別子を指定します。

- **インスタンス 1:**

インスタンスが監視するポート番号	1524
環境設定ファイルのパス	/home/maryinst1/nds.conf
DIB ディレクトリ	/home/mary/inst1/var

- **インスタンス 2:**

インスタンスが監視するポート番号	2524
環境設定ファイルのパス	/home/mary/inst2/nds.conf
DIB ディレクトリ	/home/mary/inst2/var

インスタンスの設定

前述のインスタンス識別子に基づいてインスタンスを設定するために、Mary は次のコマンドを入力する必要があります。

- **インスタンス 1:**

```
ndsconfig new -t mytree -n o=novell -a cn=admin.o=company -b 1524 -D
/home/mary/inst1/var --config-file /home/mary/inst1/nds.conf
```

- **インスタンス 2:**

```
ndsconfig new -t corptree -n o=novell -a cn=admin.o=company -b 2524 -D
/home/mary/inst2/var --config-file /home/mary/inst2/nds.conf
```

インスタンスに対するユーティリティの呼び出し

Mary は、ポート 1524 でリスンしているインスタンス 1 に対して DSTrace ユーティリティを実行しようと思っています。環境設定ファイルは /home/mary/inst1/nds.conf にあり、DIB ファイルは /home/mary/inst1/var にあります。この場合、以下のようにユーティリティを実行することができます。

```
ndstrace --config-file /home/mary/inst1/nds.conf
```

または

```
ndstrace -h 164.99.146.109:1524
```

インスタンス識別子を指定しないと、Mary が所有するすべてのインスタンスが表示され、インスタンスを選択するように求められます。

インスタンスの表示

Mary がホストのインスタンスの詳細を知りたい場合は、ndsmanage ユーティリティを実行できます。

- Mary が所有するすべてのインスタンスを表示するには、次のコマンドを実行します。

  ```
  ndsmanage
  ```

- John(ユーザ名 john) が所有するすべてのインスタンスを表示するには､次のコマンドを実行します。

  ```
  ndsmanage john
  ```

- OpenText eDirectory の特定のインストールを使用しているすべてのユーザのインスタンスをすべて表示するには、次のコマンドを実行します。

  ```
  ndsmanage -a
  ```

ndsconfig を使用してコンテナ名にドットを使用したツリーに Linux サーバをインストールする

ndsconfig を使用して、名前にドットを使用したコンテナ (novell.com など) を含む OpenText eDirectory ツリーに Linux サーバをインストールできます。

ndsconfig はコマンドラインユーティリティのため、名前にドットを含むコンテナを使用する場合はそれらのドットをエスケープする必要があります。また、ドットを含むコンテキストが含まれたパラメータは二重引用符で囲む必要があります。たとえば、名前を O として "O=netiq.com" を使用している Linux サーバに新しい eDirectory のツリーをインストールするには、次のコマンドを使用します。

```
ndsconfig new -a 'admin.netiq.com' -t netiq_tree -n
'OU=servers.O=netiq.com'
```

管理者名のパラメータ、サーバとサーバコンテキストのパラメータを二重引用符で囲み、novell.com のドット (「.」) だけを「\」(円記号) 文字を使用してエスケープします。

このフォーマットは、既存のツリーにサーバをインストールする場合にも使用できます。

注 : DSRepair、Backup、DSMerge、DSLogin、および ldapconfig などのユーティリティを使用する際に、ドットを含む管理者名やコンテキストを入力する場合も、この形式を使用してください。

---

nmasinst ユーティリティを使用して NMAS を設定する

デフォルトでは、ndsconfig を使用して NMAS を設定します。nmasinst を使用して NMAS を設定することもできます。

ndsconfig が行うのは NMAS 設定のみです。ログインメソッドのインストールは行いません。ログインメソッドのインストールには、nmasinst を使用できます。

重要 : NMAS ログインメソッドをインストールする前に、ndsconfig を使用して OpenText eDirectory を設定する必要があります。ツリーに対する管理権限も必要です。

- 72 ページの 「NMAS の設定」
- 72 ページの 「ログインメソッドのインストール」

NMAS の設定

デフォルトでは、ndsconfig を使用して NMAS を設定します。nmasinst を同じ目的で使用することもできます。

NMAS を設定し、eDirectory に NMAS オブジェクトを作成するには、サーバコンソールのコマンドラインで次のコマンドを入力します。

```
nmasinst -i admin.context tree_name
```

nmasinst からパスワードの入力が要求されます。

このコマンドでは、NMAS に必要なセキュリティコンテナ内にオブジェクトが作成され、OpenText eDirectory 内の LDAP サーバオブジェクトの NMAS に対する LDAP 拡張がインストールされます。

ツリー内で最初の NMAS をインストールする場合、セキュリティコンテナ内にオブジェクトを作成できる十分な権利を持ったユーザがインストールする必要があります。ただし、それ以降のインストールはセキュリティコンテナに対して読み込み専用の権利のみを持つコンテナ管理者も実行できます。nmasinst は、NMAS オブジェクトを作成しようとする前に、セキュリティコンテナ内に NMAS オブジェクトが存在していることを確認します。

nmasinst ではスキーマを拡張できません。NMAS スキーマは OpenText eDirectory のベーススキーマの一部としてインストールされます。

ログインメソッドのインストール

nmasinst を使用してログインメソッドをインストールするには、サーバコンソールのコマンドラインで次のコマンドを入力してください。

```
nmasinst -addmethod admin.context tree_name config.txt_path
```

最後のパラメータで、インストールするログインメソッドの config.txt ファイルを指定します。各ログインメソッドに対して、config.txt ファイルが 1 つ提供されています。

-addmethod コマンドの一例を次に示します。

```
nmasinst -addmethod admin.netiq MY_TREE ./nmas-methods/novell/Simple
Password/config.txt
```

ログインメソッドがすでに存在する場合は、nmasinst によって更新されます。

詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「ログインとポストログインのメソッドとシーケンスの管理」を参照してください。

非ルートユーザの SNMP 設定

NICI と NOVLsubag は、ルートユーザとしてインストールする必要があります。

1. ルートユーザによる NICI のインストール。「52 ページの 「ルートユーザによる NICI のインストール」」を参照してください。
2. ルートユーザによる NOVLsubag のインストール。

   NOVLsubag をインストールするには、次の手順を実行します。

   次のコマンドを入力します。

   ```
   rpm -ivh --nodeps NOVLsubag_rpm_file_name_with_path
   ```

   次に例を示します。

   ```
   rpm -ivh --nodeps novell-novell-NOVLsubag-9.2.0-0.x86_64.rpm
   ```

3. 次を実行して、パスをエクスポートします。

   手動による環境変数のエクスポート .

   ```
   export LD_LIBRARY_PATH=custom_location/opt/novell/eDirectory/lib64:/
   opt/novell/eDirectory/lib64/nds-modules:/opt/novell/
   lib64:$LD_LIBRARY_PATH

   export   PATH=/opt/novell/eDirectory/bin:$PATH

   export MANPATH=/opt/novell/man:$MANPATH
   ```

ログファイルの場所

ndsd.log

ndsd.log ファイルには、サーバシャットダウンおよび開始メッセージ、PKI や LDAP サービスの開始およびシャットダウンメッセージといった OpenText eDirectory サーバ関連のメッセージに関する情報が含まれます。デフォルトでは、ファイルは /var/opt/novell/eDirectory/log ディレクトリ内に格納されています。

/etc/opt/novell/eDirectory/conf/nds.conf ファイルで、nds.conf ファイル内の次の変数を変更することにより、ndsd.log ファイルのデバッグレベルを大きくすることができます。

n4u.server.log-levels=Logxxxx

ndsd のログレベルの詳細については、「OpenText eDirectory のエラーログを管理する」を参照してください。

Linux でのログファイルサイズの指定

ログファイルのサイズを指定するには、nds.conf ファイルで n4u.server.log-file-size パラメータを使用します。最大ファイルサイズは 2GB で、デフォルトのファイルサイズは 1MB です。ただし、1MB より小さいサイズをファイルサイズに設定することもできます。

この設定は ndsd.log ファイルには適用できません。

ログファイルのサイズが指定した制限値に到達した場合は、ログファイルの先頭から上書きされます。

4 OpenText eDirectory CE 24.4 (v9.3) の Windows へのインストールまたはアップグレード

Windows プラットフォームで OpenText eDirectory 9.3 をインストールまたはアップグレードするには、次の情報を参照してください。

- 75 ページの 「システム要件」
- 76 ページの 「前提条件」
- 79 ページの 「ハードウェア要件」
- 80 ページの 「バックリンク処理の強制実行」
- 80 ページの 「OpenText eDirectory の Windows へのインストール」
- 94 ページの 「Windows での OpenText eDirectory のアップグレード」

重要 : OpenText eDirectory 9.3 では、Novell Client がなくても OpenText eDirectory for Windows をインストールできます。詳細については、81 ページの 「Windows サーバで OpenText eDirectory 9.3 をインストールまたは更新する」を参照してください。

システム要件

OpenText eDirectory は以下のいずれかのプラットフォームにインストールする必要があります。

- Windows Server 2016 (最小要件) および Windows Server 2022

重要 : Windows デスクトップバージョンはサポートされていません。

OpenText eDirectory には、次のような要件もあります。

システム要件に関する最新情報については、『リリースノート』を参照してください。

- 割り当て済みの IP アドレス
- Windows サーバに対する管理権、および OpenText eDirectory ツリー内でドメイン対応ユーザオブジェクトを格納する全コンテナに対する管理権。既存のツリー内にインストールする場合は、スキーマを拡張しオブジェクトを作成するために、その Tree オブジェクトに対する管理権が必要です。

ご使用の Windows サーバの OS 推奨ハードウェア要件を参照してください。

前提条件

重要 : 既存の eDirectory 環境をアップグレードする前に、現在インストールされている OpenText およびサードパーティ製のアプリケーションで OpenText eDirectory 9.3 がサポートされているかどうかを確認してください。OpenText 製品の現在のステータスは、「TID 7003446 (http://www.novell.com/support/kb/doc.php?id=7003446)」で確認できます。何らかのアップグレードを実行する前に、OpenText eDirectory をバックアップすることを強くお勧めします。

- ❒ FAT ファイルシステムは NTFS に比べてトランザクション処理の安全性が低いため、OpenText eDirectory は NTFS パーティションにのみインストールできます。FAT ファイルシステムしかない場合は、次のいずれかを実行します。
  - 新しいパーティションを作成し、NTFS としてフォーマットする。

    この作業には、Windows の「ディスクの管理」を使用します。詳細については、Windows サーバのマニュアルを参照してください。
  - CONVERT コマンドを使って、既存の FAT ファイルシステムを NTFS に変換する。

    詳細については、Windows サーバのマニュアルを参照してください。

  サーバに FAT ファイルシステムしか存在しないときに上記の措置をとらなかった場合は、インストールプログラムによって NTFS パーティションを作成するよう指示されます。
- ❒ (状況によって実行) NICI 3.3 および OpenText eDirectory 9.3 でサポートされている最大のキーサイズは、RSA 暗号化用の 8192 ビットです。8KB のキーサイズを使用するには、すべてのサーバが OpenText eDirectory 9.3 にアップグレードされている必要があります。また、Identity Console などの管理ユーティリティを使用しているすべてのワークステーションに NICI 3.3 がインストールされている必要があります。

  認証局 (CA) サーバを OpenText eDirectory 9.3 にアップグレードする場合、キーサイズは変わらず、2K のままです。8K のキーサイズを作成するには、OpenText eDirectory 9.3 サーバで CA を再作成する必要があります。また、CA を作成する際に、デフォルトのキーサイズを 2K から 8K に変更する必要があります。

  注 : Windows Silent Installer を使用するには、システムに NICI 3.3 がインストールされている必要があります。
- ❒ OpenText eDirectory サーバ用の 168 ビット 3DES ツリーキーを取得してください。
- ❒ OpenText eDirectory 9.3 にアップグレードする場合は、ツリー内にある OpenText eDirectory 9.3 以外のすべてのサーバに最新の OpenText eDirectory パッチがインストールされていることを確認してください。OpenText eDirectory パッチは、OpenText Support (http://support.novell.com) Web サイトから入手できます。
- ❒ .NET Management Framework 4.0 以降が必要です。
- ❒ Windows Server に最新の Service Pack がインストールされていることを確認します。更新された最新の Windows Server Service Pack は、Windows SNMP サービスのインストール後にインストールする必要があります。

- ❒ 以前のバージョンの OpenText eDirectory からアップグレードする場合は、eDirectory 9.3.0 以降である必要があります。eDirectory バージョンの確認方法の詳細については、78 ページの 「OpenText eDirectory のバージョンの確認」を参照してください。
- ❒ (状況によって実行) 管理者以外の OpenText eDirectory ユーザとしてセカンダリサーバを既存のツリーにインストールしている場合は、次の権限を持っていることを確認します。
  - サーバのインストール先となるコンテナに対するスーパバイザ権。
  - サーバを追加するパーティションに対するスーパバイザ権。

    注 : この権限は、レプリカ数が 3 未満の場合にレプリカを追加するために必要です。
  - すべての属性権 : W0.KAP.Security オブジェクトと W1.KAP.Security オブジェクトに対する読み込み権、比較権、および書き込み権。
  - エントリ権 : Security コンテナオブジェクトに対するブラウズ権。
  - すべての属性権 : Security コンテナオブジェクトに対する読み込み権、比較権。
  - (状況によって実行)W1.KAP.Security オブジェクトが存在する場合は、すべての属性権 : このオブジェクトに対する読み込み権、比較権、および書き込み権。
    W1.KAP.Security オブジェクトの詳細については、『NICI Administration Guide (NICI 管理ガイド)』の「*Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)*」を参照してください。
- ❒ (状況によって実行) 管理者以外のユーザとしてセカンダリサーバを既存のツリーにインストールしている場合、ツリー内の少なくとも 1 台のサーバの OpenText eDirectory バージョンが、コンテナ管理者として追加しているセカンダリの OpenText eDirectory バージョンと同じかそれ以上である必要があります。追加しているセカンダリの方が新しいバージョンである場合、ツリーの管理者がスキーマを拡張してから、コンテナ管理者でセカンダリを追加する必要があります。
- ❒ セカンダリサーバを追加できるように、eDirectory の設定中に、SLP サービスとファイアウォールの NCP ポート (デフォルトは 524) を必ず有効にしてください。NCP ポートは、インバウンドトラフィックとアウトバウンドトラフィックの両方を許可するように設定する必要があります。

  必要に応じて、さらに以下のサービスポートも有効にできます。
  - LDAP (クリアテキスト) - 389
  - LDAP (セキュリティ保護) - 636
  - HTTP (クリアテキスト) - 8028
  - HTTP (セキュリティ保護) - 8030

  ユーザ定義ポートを有効にした場合、OpenText eDirectory の設定時にそのポートを指定する必要があります。
- ❒ DHCP アドレスが設定された仮想マシン、または SLP がブロードキャストでない物理マシンまたは仮想マシンに OpenText eDirectory をインストールしている場合、ディレクトリエージェントがネットワークで設定されていることを確認します。

- ❒ OpenText eDirectory 9.3 へのアップグレード中に最新のプラットフォームエージェント (PA) がインストールされていない場合、場所 <C:\NetIQ\eDirectory\auditds/ から Novell_Audit_PlatformAgent_Win64.exe ファイルを実行してインストールしてください。
- ❒ OpenText eDirectory 管理ツールボックス (eMBox) を使用すると、サーバ上でもリモートでも eDirectory のバックエンドユーティリティすべてにアクセスできます。コマンドラインクライアントは、Java アプリケーションです。これを実行するには、Oracle または Azul Java の最新バージョン (1.8 以上) をインストールする必要があります。また、古いバージョンの Java すべてで、使用可能なパッチアップグレードをインストールしてアップグレードしてください。最新バージョンの Java をインストールしたら、次の環境変数をエクスポートします。
  - EDIR_JAVA_HOME
  - JAVA_HOME
  - JRE_HOME

  注
  - OpenText eDirectory 9.0 SP4 より前のバージョンを使用している場合、コマンドラインクライアントを実行するには、Java ランタイム環境にアクセスできなければなりません。
  - Identity Manager 4.8.7 をインストールすると、OpenText eDirectory 内の Java 依存パッケージが JRE 8 から JRE 11 にアップグレードされます。

OpenText eDirectory のバージョンの確認

OpenText eDirectory のバージョンを確認するには、次のいずれかの手順を実行します。

- iMonitor を実行する。

  エージェントの概要ページで［認識サーバ］をクリックします。次に、［データベースで認識されているサーバ］の下にある［認識サーバ］をクリックします。［エージェントリビジョン］カラムに各サーバの内部ビルド番号が表示されます。たとえば、eDirectory 9.3 のエージェントリビジョン番号は 40301.00 などです。

  OpenText iMonitor の実行方法の詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*OpenText iMonitor へのアクセス*」を参照してください。
- NDSCons.exe を実行する。

  Windows コントロールパネルで、［OpenText eDirectory Services］をダブルクリックします。［サービス］カラムで、ds.dlm を選択し、［設定］をクリックします。［エージェント］タブに、マーケティング文字列 (OpenText eDirectory 9.3 など) および内部ビルド番号 (40301.00 など) が表示されます。
- ds.dlm ファイルのプロパティを表示する。

  Windows エクスプローラーの .dlm ファイルを右クリックし、［プロパティ］ダイアログボックスの［バージョン］タブをクリックします。これにより、ユーティリティのバージョン番号が表示されます。ds.dlm ファイルのデフォルトの場所は C:\NetIQ\eDirectory です。

スタティック IP アドレスを設定する

OpenText eDirectory を効率的に実行するには、サーバでスタティック IP アドレスが設定されている必要があります。DHCP アドレスのサーバに eDirectory を設定すると、予期しない結果が発生することがあります。

ハードウェア要件

ハードウェア要件は、OpenText eDirectory の特定の実装により異なります。

たとえば、標準スキーマを使用する基本的な OpenText eDirectory のインストールでは、50,000 ユーザごとに約 74MB の空きディスク容量が必要です。ただし、新しい属性のセットを追加したり、既存の属性をすべて使用すると、オブジェクトのサイズは拡大します。それに対応して、必要な空きディスク容量、プロセッサ、およびメモリが変わります。

キャッシュメモリの量とプロセッサの速さという、2 つの要因によってパフォーマンスが向上します。

最適な結果を得るためには、ハードウェアで可能な限り多くの DIB セットをキャッシュに入れるようにします。

OpenText eDirectory はシングルプロセッサ上で良好に動作します。ただし、OpenText eDirectory 9.3 では、マルチプロセッサの利点を活用できます。プロセッサを追加すると、ログインなど、一部の領域のパフォーマンスが向上します。また、複数のプロセッサ上で複数のスレッドをアクティブにすることでもパフォーマンスは上がります。OpenText eDirectory 自体は、プロセッサ集約型ではなく、入出力集約型です。

次の表に、OpenText eDirectory for Windows の一般的なシステム要件を示します。

オブジェクト	メモリ	ハードディスク
10,000	384MB	144MB
100 万	2GB	1.5GB
1,000 万	2GB 以上	15GB

プロセッサの要件は、コンピュータで利用できる追加サービス、およびコンピュータが処理している認証と読み書きの数によって決まります。暗号化や索引付けなどの処理では、プロセッサが集中して使用されることがあります。

バックリンク処理の強制実行

OpenText eDirectory にアップグレードすると内部の OpenText eDirectory 識別情報が変わるため、オブジェクトの整合性を保つために、バックリンクされたオブジェクトを更新するバックリンク処理を行う必要があります。

バックリンクでは、他のサーバ上のオブジェクトへの外部参照が追跡されます。バックリンク処理は、サーバ上の各外部参照について、実オブジェクトが正しい位置に存在することを確認するほか、マスタレプリカのすべてのバックリンク属性を確認します。バックリンク処理はデータベースがオープンされた 2 時間後に実行され、その後 780 分 (13 時間) ごとに実行されます。実行間隔には、2 分から 10,080 分 (7 日) までの任意の値を設定できます。

OpenText eDirectory に移行後、次の手順を完了して、強制的にバックリンクを実行することをお勧めします。バックリンク処理の実行は、レプリカが存在しないサーバ上では特に重要です。

1 [スタート] > [設定] > [コントロールパネル] > [OpenText eDirectory サービス] の順にクリックします。
2 [サービス] タブで [ds.dlm] を選択します。
3 [設定] をクリックします。
4 [トリガ] タブの [バックリンカ] をクリックします。

OpenText eDirectory の Windows へのインストール

このセクションでは、次のことを説明します。

- 81 ページの「Windows サーバで OpenText eDirectory 9.3 をインストールまたは更新する」
- 83 ページの 「サーバのヘルスチェック」
- 83 ページの 「LDAP を介した OpenText eDirectory との通信」
- 85 ページの 「NMAS サーバソフトウェアのインストール」
- 85 ページの 「コンテナ名にドットを使用したツリーへのインストール」
- 86 ページの 「Windows での OpenText eDirectory 9.3 の無人インストールと設定」
- 93 ページの 「ログファイルの場所」

Windows サーバで OpenText eDirectory 9.3 をインストールまたは更新する

eDirectory 9.3 for Windows は、Novell Client を含めずにインストールできます。すでに Novell Client のあるコンピュータに OpenText eDirectory 9.3 をインストールすると、OpenText eDirectory で既存のクライアントが使用されるか、最新バージョンでない場合は更新されます。

1 Windows サーバで、管理者または管理特権を持つユーザとしてログインします。

2 Autorun がオフになっている場合は、eDirectory 9.3 CD 内の windows フォルダまたはダウンロードしたファイルから、eDirectory_930_Windows_x86_64.exe を実行します。

3 (新規インストールのみ) OpenText eDirectory のインストールの種類を［基本］タブの下から選択します。

- **新しい eDirectory ツリーの作成** 新しいツリーを作成します。ツリーに最初のサーバをインストールする場合、またはこのサーバに個別のツリーが必要となる場合は、このオプションを使用します。新しいツリー上で使用可能となるリソースは、別のツリーにログインしているユーザからは使用できません。
- **既存のツリーへの eDirectory のインストール** このサーバを eDirectory ネットワークに組み入れます。サーバはツリーのどのレベルにでもインストールできます。

4 OpenText eDirectory インストール画面で情報を入力します。

- 新しい eDirectory サーバをインストールする場合は、新しいツリーのツリー名、サーバオブジェクトのコンテキスト、および管理者のログイン名とパスワードを指定します。

  重要 : OpenText eDirectory では、NCP サーバオブジェクトの FDN を最大 256 文字までで設定できますが、このオブジェクトの長さに基づいて他のより長いオブジェクトが作成されるため、この変数をずっと小さい値に制限することをお勧めします。

- インストール先が既存のツリーの場合は、既存のツリーの IP アドレス、ツリー名、サーバオブジェクトのコンテキスト、および管理者のログイン名とパスワードを指定します。
- eDirectory サーバをアップグレードする場合は、管理者のパスワードを指定します。

  注 : OpenText eDirectory 9.3 では、すべてのユーティリティに大文字と小文字を区別するパスワードを使用できます。

  コンテナ名にドットを使用する場合の詳細については、85 ページの 「コンテナ名にドットを使用したツリーへのインストール」を参照してください。

5 インストールパスを指定するか、確認します。デフォルトの場所は、C:\NetIQ\eDirectory です。

6 DIB のパスを指定するか確認します。デフォルトの場所は、C:\NetIQ\eDirectory\DIBFiles です。

7 [詳細] タブで、次の情報を指定します。

- IPv6 アドレスを使用する場合は、[**IPV6 を有効にする**] を選択します。

  注 : インストールプロセス中に IPv6 アドレスを有効にせず、後から使用することにした場合は、セットアッププログラムを再度実行する必要があります。

- 拡張バックグラウンド認証 (EBA) を有効にする場合は、[**EBA を有効にする**] を選択します。

  注 : インストールプロセス中に EBA を有効にせず、後から有効にすることにした場合は、セットアッププログラムを再度実行する必要があります。

  EBA を有効にしたセカンダリサーバをツリーに追加するには、ツリーで EBA CA を設定しておく必要があります。EBA CA が存在しない場合は、EBA を有効にせずにサーバを追加してから、EBA CA をホストするようにサーバをアップグレードします。この操作を行わないと、セカンダリサーバの設定が失敗します。

- OpenText eDirectory 管理 HTTP サーバで使用する [**HTTP スタックポート**] を指定します。

  重要 : OpenText eDirectory インストールの実行中に設定する HTTP スタックポートには、OpenText Identity Console で使用している、または使用を予定している HTTP スタックポートとは別のポートを指定してください。詳細については、『*OpenText Identity Console 管理ガイド* (https://www.netiq.com/documentation/identity-console/identity_console-admin/data/bookinfo.html)』を参照してください。

- 使用する [**LDAP ポート**] を指定します。

  詳細については、「83 ページの 「LDAP を介した OpenText eDirectory との通信」」を参照してください。

8 インストールをクリックします。

OpenText eDirectory をインストールする前に、インストールプログラムによって次のコンポーネントがチェックされます。コンポーネントが検出されなかったり、バージョンが正しくない場合は、該当するコンポーネントのインストールが自動的に開始されます。コマンドプロンプトに次のようなメッセージが表示されます。**INSTALL PASSED。** このメッセージは、NICI のインストールが成功すると自動的に消え、eDirectory のインストールが続行されます。

- NICI 3.3

  Novell International Cryptographic Infrastructure (NICI) の詳細については、『*NICI Administration Guide (NICI 管理ガイド)*』を参照してください。

9 すべての必須コンポーネントが OpenText eDirectory によって自動的にインストールおよび設定されます。

10 インストーラがインストールが完了したら、[完了] をクリックしてウィザードを終了します。

重要 : OpenText eDirectory がインストールされているサーバにログインできるのは OpenText eDirectory 管理者だけでなければなりません。

注 : OpenText eDirectory をインストールした後、eDirectory サーバの DIB ディレクトリは、ウィルス対策ソフトウェアやバックアップソフトウェアのプロセスから除外することをお勧めします。DIB ディレクトリのバックアップには、OpenText eDirectory バックアップツールを使用します。

OpenText eDirectory のバックアップの詳細については、『「OpenText™ eDirectory CE 24.4 (v9.3) eDirectory 管理ガイド』の「」OpenText eDirectory のバックアップと復元」を参照してください。

サーバのヘルスチェック

OpenText eDirectory 9.3 の場合、eDirectory のアップグレードの際に、デフォルトでサーバヘルスチェックが実行され、サーバをアップグレードしても安全かどうかが確認されます。

- 215 ページの 「パーティションとレプリカの状態」

ヘルスチェックの結果に基づいて、次のようにアップグレードが継続または中止されます。

- すべてのヘルスチェックに成功すると、アップグレードは継続されます。
- あまり重大でないエラーの場合、アップグレードを継続するか中止するかを問うメッセージが表示されます。
- 重大なエラーの場合、アップグレードは中止されます。

あまり重大でないエラーと重大なエラーの一覧については「213 ページの付録 B「OpenText eDirectory ヘルスチェック」」を参照してください。

LDAP を介した OpenText eDirectory との通信

OpenText eDirectory をインストールする場合、LDAP サーバが監視するポートを選択して、LDAP 要求を処理できるようにする必要があります。次の表では、さまざまなインストールオプションを示します。

インストール	オプション	結果
eDirectory 9.3	クリアテキスト (ポート 389)	ポート 389 を選択します。
eDirectory 9.3	暗号化 (ポート 636)	ポート 636 を選択します。

ポート 389(業界標準の LDAP クリアテキストポート)

ポート 389 を通じた接続は暗号化されません。このポートへの接続を通して送信されるすべてのデータはクリアテキストです。このため、セキュリティの問題が伴います。たとえば、単純バインド要求で LDAP パスワードが見られる可能性があります。

LDAP 単純バインドでは、DN およびパスワードのみが要求されます。パスワードは平文形式です。ポート 389 を使用する場合、すべてのパケットは平文形式です。デフォルトでは、OpenText eDirectory インストールの実行中にこのオプションは使用できません。

ポート 389 ではクリアテキストが使用できるため、LDAP サーバサービスではこのポートを通じて eDirectory への読み込みおよび書き込みを処理します。このポートの使用は開放性が高く、通信に妨害を受けることがなく、パケットが不正受信されない信頼性の高い環境に適しています。

ポート 636 に対してセキュリティ保護された接続を行い、単純バインドを実行する場合は、接続はその時点ですでに暗号化されています。このため、パスワード、データパケット、またはバインド要求を閲覧することはできません。

ポート 636(業界標準のセキュリティ保護されたポート)

ポート 636 を通じた接続は暗号化されます。TLS(以前の SSL) によって暗号化が管理されます。デフォルトでは、eDirectory のインストールではこのポートが選択されます。

ポート 636 への接続では、自動的にハンドシェークをインスタンス生成します。ハンドシェークが失敗した場合、接続は拒否されます。

重要 : この設定をデフォルトで選択することで、ローカル LDAP サーバに問題が発生する場合があります。eDirectory がインストールされる前にホストサーバにロードされているサービスがポート 636 を使用している場合は、別のポートを指定する必要があります。

OpenText eDirectory のインストールでは、nldap.nlm ファイルがロードされ、dstrace.log ファイルにエラーメッセージが記録され、セキュアポートを使用せずに実行されます。

シナリオ : ポート 636 がすでに使用されている場合 : ローカルサーバで Active Directory を実行しています。Active Directory では、ポート 636 を使用して LDAP プログラムを実行しています。eDirectory をインストールします。インストールプログラムによってポート 636 がすでに使用されていることが検出されるため、OpenText LDAP サーバにポート番号は割り当てられません。LDAP サーバはロードを開始し、実行されているように見えますが、。LDAP サーバではすでに開いているポートを複製または使用できないため、複製されたポートでの要求は LDAP サーバで処理されません。

ポート 389 またはポート 636 が OpenText LDAP サーバに割り当てられているかどうか不明な場合は、ICE ユーティリティを実行してください。[**ベンダバージョン**] フィールドに OpenText が指定されていない場合は、OpenText eDirectory の LDAP Server を再設定し、別のポートを選択する必要があります。詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*LDAP サーバが実行されているか確認する*」を参照してください。

シナリオ : Active Directory が実行中の場合 : Active Directory が実行中です。クリアテキストポート 389 が開かれています。ポート 389 に ICE コマンドを実行して、ベンダバージョンを確認してください。レポートに Microsoft* が表示されます。次に、別のポートを選択して OpenText LDAP サーバを再設定します。OpenText eDirectory LDAP サーバが LDAP の要求を処理できるようになります。

また、OpenText iMonitor では、ポート 389 または 636 がすでに開かれていることも表示できるようになります。LDAP サーバが動作していない場合、OpenText iMonitor を使って、詳細を特定します。詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*LDAP サーバが実行されているか確認する*」を参照してください。

NMAS サーバソフトウェアのインストール

NMAS (Novell Modular Authentication Service) サーバコンポーネントは、eDirectory インストールプログラムを実行すると自動的にインストールされます。NDS ログインメソッドはデフォルトで設定されます。

ログインメソッドの詳細については、『 CE 24.4 (v9.3) OpenText eDirectory 管理ガイド』の「ログインとポストログインのメソッドとシーケンスの管理」を参照してください。

コンテナ名にドットを使用したツリーへのインストール

Windows サーバは、名前にドット (.) が含まれるコンテナ (O=netiq.com または C=u.s.a など) を保持している OpenText eDirectory ツリーにインストールできます。名前にドットが含まれているコンテナを使用するには、ドットを円記号 (\) でエスケープする必要があります。ドットをエスケープするには、コンテナ名に含まれるすべてのドットの前に円記号を挿入します。

名前の最初にドットを使用することはできません。たとえば、「.netiq」という名前のコンテナは作成できません。名前がドット (「.」) で始まっているためです。

重要 : 名前にドットが含まれるコンテナがツリー内に存在する場合、OpenText iMonitor、OpenText Identity Console、DHost iConsole などのユーティリティにログインする際に、その名前をエスケープする必要があります。たとえば、ツリーに O の名前として「netiq.com」がある場合、iMonitor にログインするときは［*ユーザ名*］フィールドに、ユーザ名 **.netiq\.com** と入力します。

Windows での OpenText eDirectory 9.3 の無人インストールと設定

OpenText eDirectory 9.3 では、OpenText eDirectory のインストールとアップグレードは自動化されているため、ユーザが操作しなくても、eDirectory が Windows サーバにサイレントでインストールまたはアップグレードされます。

Windows の eDirectory の無人インストールでは、無人インストールまたは無人アップグレードを円滑に実行できるように、事前定義されたテキストファイルが使用されます。eDirectory の無人インストールを使用して、次のいずれかのセットアップを実行できます。

- OpenText eDirectory のスタンドアロンインストールまたはアップグレード (OpenText eDirectory の完全インストールかどうかによる)。スタンドアロンアップグレード処理は、インストールされているファイルだけをアップグレードします。
- インストールされている OpenText eDirectory の設定。OpenText eDirectory をインストールすると、eDirectory の設定がすべて実行されます。インストールではなく、OpenText eDirectory をアップグレードする場合、インストーラはアップグレードされたファイルだけを設定します。

無人インストールのセットアップ方法の詳細については、「87 ページの 「自動インストールに機能を追加する」」を参照してください。

前提条件

- .NET Management Framework 4.0 以降が必要です
- Windows サーバが最新の Windows パッチで更新されるようにします

次のセクションでは、インストール先、スプラッシュ画面の非表示、ポート設定、その他の NMAP メソッド、SNMP サービスの停止と開始など、無人インストールの設定に使用できるさまざまな機能について説明します。

- 86 ページの 「レスポンスファイル」
- 87 ページの 「自動インストールに機能を追加する」
- 91 ページの 「自動インストールを制御する」
- 93 ページの 「レスポンスファイルを使った OpenText eDirectory の無人インストール」

レスポンスファイル

Windows オペレーティングシステムでの OpenText eDirectory 9.3 のインストールまたはアップグレードでは、次のようなレスポンスファイルを使うと、メッセージが表示されなくなるため、より柔軟に作業することができます。

- 必要なすべてのユーザ入力が用意された完全無人インストール
- コンポーネントのデフォルト設定
- インストール中のすべてのプロンプトのバイパス

レスポンスファイルとは、Windows.ini ファイルのようなセクションとキーが記述されているテキストファイルです。任意の ASCII テキストエディタを使って、レスポンスファイルの作成と編集ができます。eDirectory アップグレードでは、レスポンスファイルからインストールパラメータが直接読み込まれ、デフォルトのインストール値がレスポンスファイルの値に置き換えられます。インストールプログラムはレスポンスファイルからの値を使って、プロンプトなしでインストールを続行します。

レスポンスファイルのセクションとキー

eDirectory 9.3 のインストールで、インストールする OpenText eDirectory インスタンスに関する情報を追加するには、ツリー名、管理者コンテキスト、管理者資格情報 (ユーザ名とパスワードを含む)、インストール先など、レスポンスファイル内のセクションを変更する必要があります。キーとそのデフォルト値の全リストは、eDirectory のインストール時に提供されるサンプルのレスポンスファイルから入手できます。OpenText eDirectory のインストール時に <eDirectoryInstallPath>\NetIQ\eDirectory\Sample_Response_File から 4 つのレスポンスファイルを入手できます。

- newtree.ni: このファイルは、新しい OpenText eDirectory ツリーの設定に使用されます。
- existingtree.ni: このファイルは、既存の OpenText eDirectory ツリーにサーバを追加するために使用されます。
- upgrade.ni: このファイルは、OpenText eDirectory サーバをアップグレードするために使用されます。
- deconfigure.ni: このファイルは、OpenText eDirectory ツリーの設定を解除するために使用されます。

注 : OpenText eDirectory のインストール時に、提供されるレスポンスファイルのいずれかを使用してください。必須のパラメータがあり、これらのファイルでデフォルトで設定されます。これらのファイルを編集する場合は、キーと値のペアを結ぶ等号記号 (「=」) の前後にスペースが入らないようにしてください。

自動インストールに機能を追加する

eDirectory インストーラの細かな設定はほとんど、手動インストールのにデフォルト設定になっています。ただし無人インストール中は、各環境設定パラメータは明示的に設定されていなければなりません。このセクションでは、インストールの順番や追加機能に関係のない基本設定について説明します。

OpenText eDirectory サーバの詳細情報

アップグレードか、プライマリ / セカンダリサーバのインストールかにかかわらず、インストールまたはアップグレード対象のサーバの詳細情報が、インストーラに提供される必要があります。この情報のほとんどは、タグ [NWI:NDS] で設定されます。

```
[NWI:NDS]
```

- **mode:** デフォルトでは、mode キーは configure に設定されています。これによって OpenText eDirectory が設定されます。

- **Tree Name:** プライマリサーバのインストールでは、これはインストールする必要があるツリーの名前です。セカンダリサーバのインストールでは、サーバの追加先となるツリーになります。
- **Server Name:** インストールするサーバの名前です。
- **Server Container:** ツリーに追加されたサーバにはサーバオブジェクトがあり、そこにサーバ固有の詳細な設定情報がすべて入っています。このパラメータは、サーバオブジェクトの追加先となるツリーのコンテナオブジェクトです。プライマリサーバのインストールでは、このコンテナはサーバオブジェクトと共に作成されます。
- **Admin Login Name:** 少なくともサーバの追加先のコンテキストに対してフル権限を持つ、ツリー内の管理者オブジェクトの名前 (RDN)。ツリー内のすべての操作は、このユーザとして実行されます。
- **Admin Context:** ツリーに追加されたユーザにはユーザオブジェクトがあり、そこにユーザ固有の詳細情報がすべて入っています。このパラメータは、管理者オブジェクトの追加先となるツリーのコンテナオブジェクトです。プライマリサーバのインストールでは、このコンテナはサーバオブジェクトと共に作成されます。
- **Admin password:** 前述のパラメータで作成された管理者オブジェクトのパスワード。このパスワードは、プライマリサーバのインストール時に管理者オブジェクトに対して設定されます。セカンダリサーバのインストールでは、これは新しいサーバの追加先となるコンテキストに対して権限を持っているプライマリサーバの管理者オブジェクトのパスワードである必要があります。

  環境変数で管理者のパスワードを設定し、レスポンスファイルで環境変数の名前を指定することをお勧めします。サイレント設定が完了したら、環境変数からパスワードを削除します。

  重要 : 無人インストールで使用するレスポンスファイルに管理者ユーザ資格情報を入力します。このため、管理者資格情報が漏洩しないように、インストール後にこのファイルを完全に削除する必要があります。

- **DataDir:** デフォルトでは、DIB は NDS ロケーション内の Files サブフォルダにインストールされますが、管理者はこのパラメータを変更して別の場所を指定することができます。このパラメータに値が指定されない場合、値はデフォルトで <Install location>/DIBFiles に設定されます。
- **EBA:** 拡張バックグラウンド認証 (EBA) は、ツリー内の NCP サーバを認証するため、改良され安全性の高まったバックグラウンド認証プロトコルを提供します。OpenText eDirectory には、OpenText eDirectory ツリーの設定中または設定後に EBA を有効にするためのオプションが用意されています。レスポンスファイルを変更しない限り、EBA はデフォルトでは OpenText eDirectory 上に設定されません。EBA を有効にするには、［Require EBA］を［Yes］に設定します。
- **FIPS:** OpenText は、連邦情報処理標準 (FIPS) モードで実行する eDirectory をサポートしています。FIPS モードで実行する OpenText eDirectory を有効にするには、［Require FIPS for TLS］を［Yes］に設定します。
- **Enable PBKDF2:** 新しい環境設定パラメータ Enable PBKDF2 が、OpenText eDirectory 9.3 の newtree.ni レスポンスファイルに追加されました。このオプションを「yes」に設定すると、パスワードポリシーが作成され、ツリー全体に自動的に割り当てられます。このパスワードポリシーにより、ツリー内のすべてのユーザに対して、NDS パスワード

の PBKDF2 パスワードとの同期が可能になります。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「非可逆パスワードストレージを理解する」を参照してください。

前述の基本パラメータをすべて記述したレスポンスファイルのテキスト例を次に示します。

```
[NWI:NDS]
mode=configure
New Tree=Yes
Tree Name=ENEWTREE
Server Name=ENEWSERVER
Server Container=myorg
Admin Context=myorg
Admin Login Name=Admin
Admin Password=env: PASSWORD_VAR
Require IPV6=NO
Require EBA=NO
Require FIPS for TLS=NO
DataDir=C:\NetIQ\eDirectory\DIBFiles
LDAP TCP Port=389
LDAP SSL Port=636
Require TLS=No
Require SS=YES
Enable PBKDF2=No
```

NMAS メソッドを追加する

OpenText eDirectory のインストールでもアップグレードでも、複数の NMAS メソッドのインストールがサポートされています。手動インストール中に、インストールして設定する NMAS メソッドを選択できます。自動インストールでも、NMAS メソッドの選択ができます。

NMAS 関連の環境設定は、[NWI:NMAS] タグ内で指定します。タグには、設定するキーが 2 つあり、どちらも必須です。

- **Choices:** このキーは、インストールする必要がある NMAS メソッドの数を、OpenText eDirectory インストールコンポーネントに通知します。
- **Methods:** このキーは、インストールする必要がある NMAS メソッドオプションを一覧表示します。現在、サポートされている NMAS メソッドが 6 つあります。メソッド名とそのタイプは以下の通りです。

表 4-1 NMAS メソッド

メソッド名	メソッドタイプ
CertMutual	証明書相互ログインメソッド
Challenge Response	OpenText チャレンジ / レスポンス方式 NMAS メソッド
DIGEST-MD5	ダイジェスト MD5 ログインメソッド
SAML	Security Assertion Markup Language の認証メソッド
NDS	NDS ログインメソッド (デフォルト)
Simple Password	シンプルパスワード NMAS ログインメソッド
SCRAM	Salt Challenge Response Authentication Mechanism(SCRAM) では、PBKDF2 ハッシュベースのパスワードを使用します。

注 : メソッド名は、Method キーに対するオプションとして、上の表に表示された名前と正確に一致する必要があります。インストーラは、インストールする NMAS メソッドを選択するため、文字列の正確な比較 (大文字 / 小文字を含む) を行います。

MSD NMAS メソッドは必須で、NMAS メソッドのリストが無い場合に自動的にインストールされます。明示的リストが作成する場合も、リストからこのメソッドを削除しないでください。

この手法を使ってレスポンスファイルに NMAS メソッドが設定されている場合、OpenText eDirectory はユーザ入力のプロンプトを出さずに、インストール中にステータスメッセージを表示します。

NMAS メソッドを選択するレスポンスファイルのサンプルテキストを次に示します。

```
[NWI:NMAS]
Methods=CertMutual,Challenge Response,DIGEST-MD5,NDS,Simple Password,SAML
```

HTTP ポート

OpenText eDirectory は、Web を介したアクセスのために事前設定された HTTP ポートをリスンします。たとえば、OpenText iMonitor は Web インターフェイスを介して eDirectory にアクセスします。適切なアプリケーションにアクセスするには、具体的な名前を指定する必要があります。eDirectory を特定のポートに設定するには、インストールの前に設定できるキーが 2 種類あります。

- **クリアテキスト HTTP ポート :** クリアテキストの HTTP 操作のためのポート番号。
- **SSL HTTP ポート :** セキュアソケットレイヤ操作のための HTTP ポート番号。

HTTP ポート番号を設定するレスポンスファイルのテキスト例を次に示します。

```
[eDir:HTTP]

Clear Text HTTP Port=8028

SSL HTTP Port=8030
```

LDAP 環境設定

OpenText eDirectory は、LDAP 操作をサポートしています。eDirectory は、2 つの異なるポート上で、クリアテキストと SSL の LDAP リクエストをリスンします。インストール前にこれらのポートをレスポンスファイルで設定すれば、OpenText eDirectory の起動時に設定されたポートをリスンすることができます。

［NWI:NDS］タグには、LDAP ポートを設定する以下の 3 つのキーがあります。

- **LDAP TCP Port:** OpenText eDirectory がクリアテキストの LDAP リクエストをリスンするポート。ポートが指定されない場合、デフォルトで 389 が使用されます。
- **LDAP SSL Port:** OpenText eDirectory が SSL の LDAP リクエストをリスンするポート。バインドリクエストが平文でパスワードを送信する際にセキュリティ保護された接続を OpenText eDirectory が必須とするかどうか、キーを使って設定することもできます。ポートが指定されない場合、デフォルトで 636 が使用されます。
- **Require TLS:** OpenText eDirectory が LDAP リクエストをクリアテキストで受信する場合、TLS を必須とするかどうか指定します。このパラメータに値が指定されない場合、デフォルトで［はい］に設定されます。

LDAP 環境設定のレスポンスファイルのテキスト例を次に示します。

```
[NWI:NDS]

Require TLS=Yes

LDAP TLS Port=389

LDAP SSL Port=636
```

自動インストールを制御する

レスポンスファイルを編集して、自動インストールのフローを制御することもできます。

SNMP サービスを停止する

これは、Windows での OpenText eDirectory インストールに固有の機能です。ほとんどの Windows サーバでは、SNMP が設定され、動作しています。OpenText eDirectory をインストールするときは、SNMP サービスをダウン状態にして、インストール後に再開する必要があります。手動インストールの場合、インストールを続行する前に SNMP サービスを停止するよう求めるプロンプトがインストーラの画面に表示されます。[NWI:SNMP] タブのキーを次のように設定することで、自動インストール中にこのプロンプトを表示しないようにすることができます。

- **Stop service:** この値を「Yes」に設定すると、プロンプトを表示せずに SNMP サービスを停止します。状態が画面上に表示されます。

SNMP サービスを停止するレスポンスファイルのテキスト例を次に示します。

```
[NWI:SNMP]

Stop service=yes
```

デフォルトサーバ証明書のデフォルトのパラメータを指定する

OpenText eDirectory には、新しい OpenText eDirectory ツリーを設定する際に、CA 証明書とデフォルトサーバ証明書のデフォルト RSA キーサイズ、楕円曲線、および有効期限を指定するオプションが用意されています。レスポンスファイルの新しい eDirectory ツリーのサイレントインストール時に CA 証明書とデフォルトサーバ証明書のために次のデフォルトパラメータを指定できます。

- **RSA Key Size:** RSA 証明書のキーサイズを指定します。使用可能な値は 2048、4096、および 8192 ビットです。
- **EC Curve:** EC 証明書の曲線の制限を指定します。使用可能な値は P256、P384、および P521 です。
- **Certificate Life:** 証明書の有効期限を年数で指定します。

ここで指定される値は、新しいツリーが構成されるときに組織の CA オブジェクトの対応する属性に設定されます。

これらの属性は、以下の例に示すように、新しい eDirectory サーバのインストール中に newtree.in ファイルの [NWI:PKI] タグで設定できます。

```
[NWI:PKI]
RSA KeySize=4096
EC Curve=P521
Certificate Life=4
```

詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*組織の認証局オブジェクトを作成する*」を参照してください。

プライマリ / セカンダリサーバのインストール

OpenText eDirectory インストーラには、プライマリサーバまたはセカンダリサーバをネットワークに無人インストールする際に使用できるオプションがあります。プライマリサーバのインストールか、セカンダリサーバのインストールかを決定するキーが 1 つあります。

- **プライマリサーバ :** [NWI:NDS] タグで［新しいツリー］キーを使用し、newtree.ni ファイル、または新しいサーバのセットアップに必要な類似のレスポンスファイルで新規 / プライマリツリーのインストールを Yes に設定します。
- **セカンダリサーバ :** [NWI:NDS] タグで［新しいツリー］キーを使用し、existingtree.ni ファイル、またはセカンダリサーバのセットアップに必要な類似のレスポンスファイルでセカンダリツリーのインストールを No に設定します。

たとえば、新しいツリーにプライマリサーバをインストールする場合は、次のようになります。

```
[NWI:NDS]

New Tree=Yes
```

既存ツリーにセカンダリサーバをインストールする場合は、次のようになります。

```
[NWI:NDS]
```

```
New Tree=No
```

レスポンスファイルを使った OpenText eDirectory の無人インストール

Windows で OpenText eDirectory インストーラを起動することは簡単です。eDirectory リリースで配布されている eDirectory_930_Windows_x86_64.exe は、追加パラメータを指定してコマンドラインから起動します。

指定したセットアップモードに応じて、次のコマンドのどちらかを使用します。

インストール

Windows のコマンドプロンプトで次のコマンドを実行します。

<*Download Location Path*>\eDirectory_930_Windows_x86_64.exe /qn

例 : D:\builds\eDirectory_930_Windows_x86_64.exe /qn

注 : 任意の場所に OpenText eDirectory をインストールするには、次のコマンドを実行します。

eDirectory_930_Windows_x86_64.exe /qn INSTALLDIR="C:\<*Install Location*>

設定

Windows PowerShell で次のコマンドを実行します。

<*eDirectory のインストール場所*>。/EConfig.ps1 -rfile < サンプルレスポンスファイルの場所 >\newtree.ni

たとえば、C:\NetIQ\eDirectory>。/EConfig.ps1 -rfile C:\Sample_Response_Files\newtree.ni

---

注 : ログファイルには、以下の場所からアクセスできます。

- C:\Program Files\NetIQ\eDirectory\installlogs
- C:\Program Files\NetIQ\eDirectory\logs

ログファイルの場所

dsinstall.log

<Windows Drive>\NetIQ\eDirectory から入手可能な dsinstall.log ファイルの最初の部分には、設定される環境変数の一覧が示されます。2 番目の部分には、OpenText eDirectory のインストール処理を記録するステータスメッセージが含まれています。

Windows での OpenText eDirectory のアップグレード

eDirectory をバージョン 8.8 SP1 から 9.3 に直接アップグレードしないことをお勧めします。9.2 にアップグレードしてから、9.3 にアップグレードする必要があります。EBA 対応サーバを使用している場合は、OpenText eDirectory 9.3.0 にアップグレードする前に、OpenText eDirectory 9.2.8 または 9.2.9 を使用している必要があります。

NMAS メソッドがすでにインストールされている場合、OpenText eDirectory 9.30 にアップグレードするには、アップグレード前に NMAS スキーマを拡張する必要があります。詳細については、94 ページの 「OpenText eDirectory をアップグレードする前の NMAS スキーマ拡張」を参照してください。

注 : 32 ビットバージョンの OpenText eDirectory を 64 ビットバージョンの OpenText eDirectory にアップグレードするには、まず 32 ビットバージョンを eDirectory 8.8.x 64 ビットバージョンにアップグレードし、次にそれを OpenText eDirectory 9.2 にアップグレードします。64 ビットの OpenText eDirectory を eDirectory 9.3 にアップグレードする場合も、同じ手順に従います。

次のセクションでは、インストール済みの既存の OpenText eDirectory を現在のバージョンにアップグレードするのに役立つ情報を提供します。

- 94 ページの 「OpenText eDirectory をアップグレードする前の NMAS スキーマ拡張」
- 95 ページの「Windows インストーラを使用して OpenText eDirectory をアップグレードする」
- 96 ページの 「Windows での OpenText eDirectory の無人アップグレード」

OpenText eDirectory をアップグレードする前の NMAS スキーマ拡張

eDirectory_930_Windows_x86_64.exe をダウンロードして抽出すると、インストーラと一緒に preUpgrade_930 ファイルが存在します。preUpgrade_930 に移動すると、次のファイルが表示されます。

- preUpgrade_930/preUpgrade.ps
- preUpgrade_930/upgrade.ni
- preUpgrade_930/nmas.sch

Windows でのスキーマ拡張 :

OpenText eDirectory を 9.3.0 にアップグレードする前に、次の手順に従ってスキーマを拡張します。

1. preUpgrade_930 フォルダにある upgrade.ni ファイルを更新します。既存の eDirectory サーバのツリー名、サーバ名、および管理者資格情報を、upgrade.ni レスポンスファイルに指定します。次に示すのは、アップグレードの設定を指定した upgrade.ni レスポンスファイルのサンプルです。

```
[NWI:NDS]
mode=configure
Tree Name=enewtree
Server Name=enewserver
Server Container=org
Admin Context=org
Admin Login Name=Admin
Admin Password=env:PASSWORD_VAR
Require IPV6=NO
Require EBA=NO
Require FIPS for TLS=YES
LDAP TCP Port=389
LDAP SSL Port=636
Require TLS=Yes
Require SS=Yes
Existing Server=10.10.10.10
Existing Server Port=524
[NWI:SNMP]
Stop service=No
[NWI:NMAS]
Methods=CertMutual,Challenge Response,DIGEST-MD5,NDS,Simple
Password,SAML
```

2 powershell から次のコマンドを実行します。

```
preUpgrade.ps1 -rfile <path_for_Configuration>/upgrade.ni
```

3 スキーマ拡張が成功したら、eDirectory の 9.3.0 へのアップグレードに進みます。

Windows インストーラを使用して OpenText eDirectory をアップグレードする

Windows インストーラを使用して、OpenText eDirectory サーバをアップグレードすることができます。eDirectory サーバをアップグレードするには、次の手順を実行します。

1 Windows サーバで、管理者または管理特権を持つユーザとしてログインします

2 eDirectory 9.3 CD の windows フォルダから、またはダウンロードしたファイルから、eDirectory_930_Windows_x86_64.exe を実行します。

3 インストーラの画面に既存の OpenText eDirectory ツリー名が表示され、[**基本**] タブにサーバの FDN が表示されます。ツリー管理者の資格情報を入力し、[**アップグレード**] ボタンをクリックしてアップグレードプロセスを続行します。

4 [**詳細**] タブでは、OpenText eDirectory のインストール中に設定された既存の設定を変更できます。詳細については、81 ページの 「Windows サーバで OpenText eDirectory 9.3 をインストールまたは更新する」を参照してください。

注 : Windows で eDirectory 9.2.8 および 9.2.9 での EBA 対応のレプリカサーバをアップグレードするには、28 ページの 「Windows で EBA を使用する複数のレプリカサーバへのブリッジパッチのインストール」を参照してください。

Windows での OpenText eDirectory の無人アップグレード

Windows 上で OpenText eDirectory をアップグレードする場合は、サイレントモードで実行することができます。

Windows では、アップグレードする前に、OpenText eDirectory の最新バージョンをインストールする必要があります。

```
<Download Location Path>\eDirectory_930_Windows_x86_64.exe /qn
```

例 :

```
D:\builds\eDirectory_930_Windows_x86_64.exe /qn
```

詳細については、86 ページの 「Windows での OpenText eDirectory 9.3 の無人インストールと設定」を参照してください。

すべてのライブラリを正常にアップグレードした後、既存の eDirectory サーバのツリー名、サーバ名、および管理者資格情報を、upgrade.ni レスポンスファイルに指定します。

次に示すのは、アップグレードの設定を指定した upgrade.ni レスポンスファイルのサンプルです。

```
[NWI:NDS]
mode=configure
Tree Name=enewtree
Server Name=enewserver
Server Container=org
Admin Context=org
Admin Login Name=Admin
Admin Password=env:PASSWORD_VAR
Require IPV6=NO
Require EBA=NO
Require FIPS for TLS=YES
LDAP TCP Port=389
LDAP SSL Port=636
Require TLS=Yes
Require SS=Yes
Existing Server=10.10.10.10
Existing Server Port=524

[NWI:SNMP]
Stop service=No

[NWI:NMAS]
Methods=CertMutual,Challenge Response,DIGEST-MD5,NDS,Simple Password,SAML
```

必要な OpenText eDirectory サーバの詳細を指定して、upgrade.ni レスポンスファイルの更新が済んだら、次のコマンドを実行して OpenText eDirectory サーバをアップグレードします。

<*eDirectory installed location*> ./EConfig.ps1 -rfile <Sample_Response_Files location>\upgrade.ni

例 : C:\NetIQ\eDirectory> ./EConfig.ps1 -rfile C:\Sample_Response_Files\upgrade.ni。

5 Microsoft Azure での OpenText eDirectory のデプロイ

OpenText eDirectory は、Microsoft Azure 仮想マシンにデプロイできます。サポートされているオペレーティングシステムの詳細については、Linux の場合は 31 ページの 「システム要件」、Windows の場合は 75 ページの 「システム要件」を参照してください。

前提条件

OpenText eDirectory のシステム要件に加えて、次の要件を満たしていることを確認してください。

- Azure の管理アカウント。
- 仮想マシンにコピーできるように、OpenText eDirectory インストーラ (tarball ファイル) をダウンロードし、抽出してあること。Windows の場合は、Windows 実行可能ファイルをダウンロードします。
- クライアントマシンから Azure 仮想マシンに接続する SSH クライアント。

デプロイメントの手順

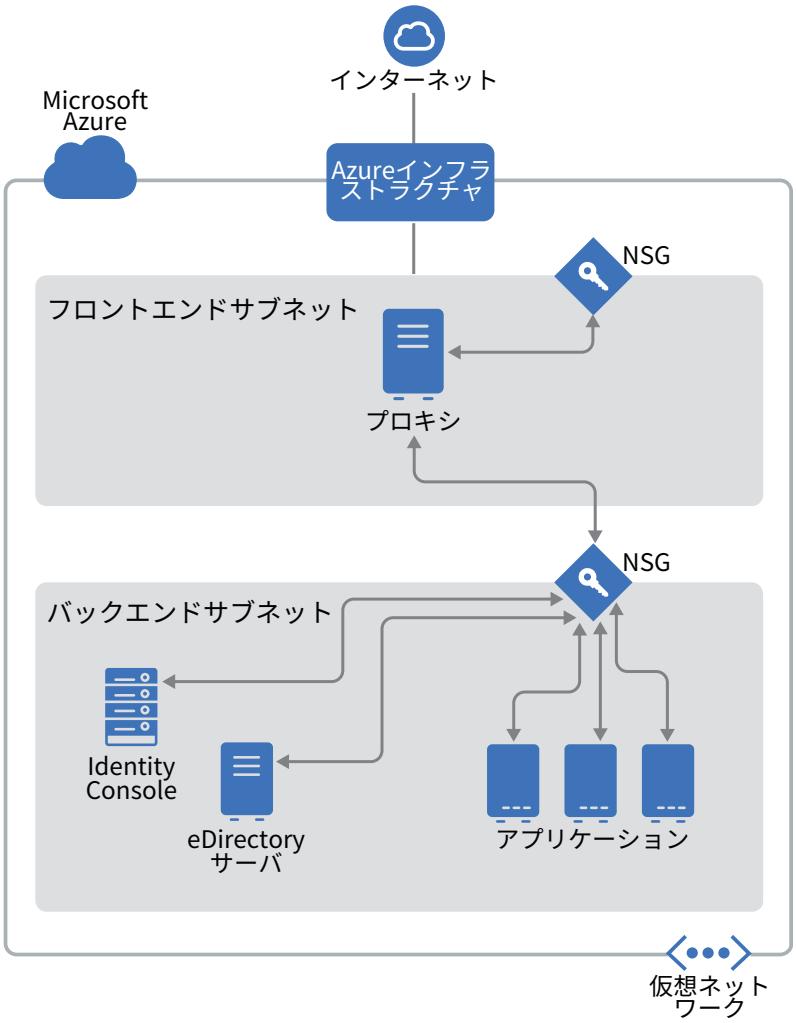
次のセクションでは、仮想マシンで実行されているオペレーティングシステムに応じて、Microsoft Azure に OpenText eDirectory をデプロイする手順について説明します。

- 97 ページの 「Linux プラットフォームへのデプロイ」
- 113 ページの 「Windows プラットフォームへのデプロイ」

Linux プラットフォームへのデプロイ

OpenText eDirectory は、必ず Azure Virtual Network のバックエンドサブネット内にデプロイする必要があります。図 5-1 は、以降のセクションで使用されるサンプル展開を示しています。

図 5-1 Azure 上での OpenText eDirectory のデプロイメント



注

- プロキシは、管理者が SSH を使用して接続し、SSH エージェント転送を使用してバックエンドサブネット内の他のインスタンスに接続する、フロントエンドサブネット内の要塞ホストです。
- OpenText eDirectory にアクセスする必要があるアプリケーションは、バックエンドサブネットにデプロイする必要があります。これらのアプリケーションにインターネットからアクセスする必要がある場合は、フロントエンドサブネット内に Azure ロードバランサーを設定して、アクセスを可能にします。詳細については、「Azure portal を使用して Basic Load Balancer を作成する」を参照してください。
- OpenText eDirectory マスタをオンプレミスにインストールし、レプリカを Azure 仮想マシン上にインストールする (またはその逆) というハイブリッドセットアップはサポートされていません。

デプロイメント手順は、次のステップで構成されます。

- 99 ページの 「Azure サービスの準備」
- 100 ページの 「アプリケーションセキュリティグループ (ASG) の設定」
- 100 ページの 「サブネット用のネットワークセキュリティグループ (NSG) の設定」
- 103 ページの 「仮想マシン用のネットワークセキュリティグループの設定」
- 106 ページの 「SSH キーペアを作成する」
- 106 ページの 「仮想マシンの作成とデプロイ」
- 107 ページの 「OpenText eDirectory データを格納するためのデータディスクの設定」
- 108 ページの 「OpenText eDirectory および OpenText Identity Console のインストール」
- 112 ページの 「監査サービスのデプロイ」
- 112 ページの 「障害復旧」

Azure サービスの準備

このセクションでは、OpenText eDirectory で使用する Azure サービスを作成するための一般的な手順について概説します。これには、リソースグループ、仮想ネットワーク (VNet)、およびサブネットの作成が含まれます。

重要 : 仮想ネットワーク、セキュリティグループ、仮想マシンなどのサービスを作成する際には、［場所］に同じ値を指定していることを確認してください。

リソースグループの作成

リソースグループは、Azure ソリューションに関連するリソースを保持するコンテナです。リソースグループには、ソリューションのすべてのリソースを含めるか、またはグループとして管理するリソースのみを含めることができます。たとえば、Azure に OpenText eDirectory をデプロイしている間は、リソースグループには仮想マシン、仮想ネットワーク、アプリケーションセキュリティグループ、ネットワークセキュリティグループ、パブリック IP アドレス、ネットワークインターフェイス、およびディスクが含まれている必要があります。リソースグループの作成方法の詳細については、「Azure portal を使用した Azure リソースの管理」を参照してください。

注 : すべての管理者が、新しいリソースグループを作成する権限を持っているわけではありません。

仮想ネットワークの作成

Azure Virtual Network を使用すると、Azure 仮想マシン (VM) などの多くの種類の Azure リソースが、Azure インターネット内で相互に、またオンプレミスのネットワークと安全に通信することができます。詳細については、「Azure Virtual Network とは」を参照してください。

仮想ネットワークを作成する際には、デフォルトで 1 つのサブネットが作成されます。複数のサブネットを作成する場合は、新しく作成された**仮想ネットワーク**から、［**サブネット**］>［**サブネットの追加**］に移動します。

アプリケーションセキュリティグループ (ASG) の設定

アプリケーションセキュリティグループを使用すると、アプリケーション構造の自然な拡張としてネットワークセキュリティを設定できます。さらに、仮想マシンをグループ化し、それらのグループに基づいてネットワークセキュリティポリシーを定義することもできます。詳細については、「アプリケーションセキュリティグループ」を参照してください。

ネットワークセキュリティグループを設定する前に、次のアプリケーションセキュリティグループを作成する必要があります。

表 5-1 *アプリケーションセキュリティグループ*

名前	説明
SSH_Proxy	SSH_Proxy を設定する仮想マシンのネットワークインターフェイスが含まれます
eDirectory	OpenText eDirectory を設定するすべての仮想マシンのネットワークインターフェイスが含まれます
eDirectory_CA	ツリー認証局 (CA) をホストする OpenText eDirectory サーバを設定する仮想マシンのネットワークインターフェイスが含まれます
OpenText Identity Console	OpenText Identity Console を設定する仮想マシンのネットワークインターフェイスが含まれます

サブネット用のネットワークセキュリティグループ (NSG) の設定

NSG を使用して、サブネットのインバウンドおよびアウトバウンドのネットワークトラフィックをフィルタすることができます。NSG には、ネットワークトラフィックを、IP アドレス、ポート、およびプロトコルによってフィルタするセキュリティルールが含まれています。

このセクションでは、フロントエンドサブネット用に NSG を作成するためのルールについて概説します。デフォルトのセキュリティルールの上に、次のルールを設定します。

- インバウンドルール :

表 5-2 フロントエンドサブネットのインバウンドルール

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	SSH	TCP 22	すべて (Any)	SSH_Proxy (ASG)	許可	インターネットからプロキシサーバへの SSH 接続を許可します
110	Allow Subnet Traffic	すべて (Any)	すべて	フロントエンドサブネット	許可	(オプション) すべてのサブネット内トラフィックを許可します。 **注 :** このルールは、相互に通信する必要のある他の仮想マシンがフロントエンドサブネット内にある場合にのみ設定します。
120	All Traffic	すべて (All)	すべて	すべて	拒否	前のルールによってまだ処理されていないすべてのインバウンドトラフィックを拒否します

このセクションでは、バックエンドサブネットで NSG を作成するためのルールについて概説します。ネットワークセキュリティグループに対して、次のルールを設定します。

- インバウンドルール :

表 5-3 バックエンドサブネットのインバウンドルール

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	SSH	TCP 22	プロキシ (ASG)	バックエンドサブネット	許可	SSH プロキシからのインバウンド SSH トラフィックを許可します
110	Identity Console	TCP 9000	プロキシ (ASG)	OpenText Identity Console (ASG)	許可	SSH プロキシから OpenText Identity Console にアクセスするための HTTPS トラフィックを許可します
120	HTTP CRL	TCP 8028	仮想ネットワーク	eDirectory_CA (ASG)	許可	ツリー CA によって発行された証明書を使用して設定されている VNet にサービスが存在する場合に、VNet から OpenText eDirectory ツリーの CRL にアクセスするために必要です。
130	Allow Subnet Traffic	すべて (Any)	バックエンドサブネット	バックエンドサブネット	許可	すべてのサブネット内トラフィックを許可します
140	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

仮想マシン用のネットワークセキュリティグループの設定

セキュリティグループは、仮想ネットワーク内の 1 つまたは複数の仮想マシンに割り当てることができる仮想ファイアウォールルールのセットです。

デフォルトでは、新しいセキュリティグループではポート 22 の着信トラフィックのみが許可されるので、インスタンスには SSH を使用してのみ接続できます。

詳細については、「セキュリティグループ」を参照してください。

Azure に OpenText eDirectory をデプロイするには、次のネットワークセキュリティグループを作成します : eDirectory_NSG_1、eDirectory_NSG_2、および IdentityConsole_NSG。これらのセキュリティグループを作成して次のポートルールを設定し、デフォルトのセキュリティルールを上書きします。

1. **eDirectory_NSG_1:** この NSG は、eDirectory ツリーの CA をホストする仮想マシンに関連付ける必要があります。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	SSH	TCP 22	SSH プロキシ (ASG)	ASG (OpenText eDirectory)	許可	SSH プロキシからの SSH トラフィックを許可します
110	NCP	TCP 524	バックエンドサブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内で OpenText eDirectory に対する NCP トラフィックを許可します
120	HTTP CRL	TCP 8028	仮想ネットワーク	eDirectory_CA (ASG)	許可	ツリー CA によって発行された証明書を使用して設定されている VNet にサービスが存在する場合に、VNet から eDirectory OpenText ツリーの CRL にアクセスするために必要です。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
130	LDAPS	TCP 636	バックエンドサブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内でセキュリティ保護された LDAP トラフィックを許可します
140	SLP	Any 427	バックエンドサブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内の SLP トラフィックを許可します
150	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

注 : OpenText eDirectory サーバは、LDAP ポート 389 でリスンするように設定してはならず、OpenText eDirectory に割り当てられているセキュリティグループに対してポート 389 へのアクセスを許可してはなりません。また、HTTP ポートへのアクセスを許可するのは、ツリーの CA をホストしている OpenText eDirectory サーバに割り当てられたセキュリティグループに対してのみにする必要があります。

2. **eDirectory_NSG_2:** この NSG は、OpenText eDirectory ツリー CA 以外の OpenText eDirectory サーバをホストしているすべての仮想マシンに関連付ける必要があります。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	SSH	TCP 22	SSH プロキシ (ASG)	ASG (OpenText eDirectory)	許可	SSH プロキシからの SSH トラフィックを許可します
110	NCP	TCP 524	バックエンドサブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内で OpenText eDirectory に対する NCP トラフィックを許可します

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
120	LDAPS	TCP 636	バックエンドサブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内でセキュリティ保護された LDAP トラフィックを許可します
130	SLP	Any 427	バックエンドサブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内の SLP トラフィックを許可します
140	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

3. **IdentityConsole_NSG:** この NSG は、Identity Console をホストする仮想マシンに関連付ける必要があります。次の NSG ルールを使用すると、プロキシサーバのみから Identity Console サーバにアクセスできるようになります。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	SSH	TCP 22	SSH プロキシ (ASG)	OpenText Identity Console (ASG)	許可	プロキシからの SSH トラフィックを許可します
110	HTTPS	TCP 9000	SSH プロキシ (ASG)	OpenText Identity Console (ASG)	許可	プロキシから Identity Console にアクセスする、セキュリティ保護された HTTP トラフィックを許可します
120	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

SSH キーペアを作成する

Azure VM を設定する前に、SSH キーペアを作成する必要があります。キーペアを作成するには、次の手順を実行します。

1 次のコマンドを使用して、クライアント上で 4096 ビットの RSA SSH キーペアを作成します。

```
ssh-keygen -t rsa -b 4096
```

ssh-keygen により、新しく作成された公開鍵が ~/.ssh/id_rsa.pub に置かれます。

2 この SSH 公開鍵を Azure アカウントに提供します。詳細については、「SSH 公開鍵を提供する」を参照してください。

重要 : SSH 秘密鍵を使用してのみ、仮想マシンに接続して管理できます。そのため、SSH 秘密鍵を紛失しないようにしてください。

仮想マシンの作成とデプロイ

サポートされているプラットフォームのいずれかで仮想マシン (VM) を作成して起動します。VM を作成して起動する方法の詳細については、「Azure portal で Linux 仮想マシンを作成する」を参照してください。インスタンスを作成および起動する際には、次の手順も実行する必要があります。

1 最初の OpenText eDirectory サーバを設定する VM に eDirectory_NSG_1 を関連付け、その他のすべての OpenText eDirectory サーバを設定する VM に eDirectory_NSG_2 を関連付け、Identity Console を設定する VM に IdentityConsole_NSG を関連付けます。セキュリティグループの詳細については、103 ページの 「仮想マシン用のネットワークセキュリティグループの設定」を参照してください。

2 106 ページの 「SSH キーペアを作成する」セクションで作成した公開鍵をインスタンスに関連付けます。

注 : 選択した Azure の場所に対して複数の可用性ゾーンが使用可能な場合は、レプリカサーバをマスタ OpenText eDirectory サーバと同じ可用性ゾーンにデプロイしないでください。

OpenText eDirectory データを格納するためのデータディスクの設定

Azure VM のクラッシュが発生した場合に OpenText eDirectory のデータと設定が失われるのを防ぐには、データディスクを設定する必要があります。OpenText eDirectory のデータと設定を復元する方法の詳細については、112 ページの 「障害復旧」を参照してください。VM を作成した後、次の手順を実行して、OpenText eDirectory をデプロイできるように VM を準備します。

1 データディスクを作成して接続するために、「ポータルを利用し、データディスクを Linux VM に接続する」の手順を実行します。

2 VM にログインし、ext4 ファイルシステムを使用してデータディスクをフォーマットし、データディスクをマウントします。データディスクをフォーマットしてマウントする方法の詳細については、「Linux VM を接続して新しいディスクをマウントする」を参照してください。

3 ディレクトリを、データディスクから OpenText eDirectory データディレクトリと NICI データディレクトリにバインドマウントします。ルートユーザとして次の手順を実行して、バインドマウントします。

3a 次のコマンドを使用して、OpenText eDirectory データディレクトリを作成します。

```
mkdir <mount_point>/eDirectory_data
```

3b 次のコマンドを使用して、NICI データディレクトリを作成します。

```
mkdir <mount_point>/nici_data
```

3c 次のコマンドを使用して、NICI および OpenText eDirectory の環境設定ディレクトリを作成します。

```
mkdir <mount_point>/eDirectory_nici_conf
```

3d 次のコマンドを使用して、OpenText eDirectory に必要なディレクトリを作成します。

```
mkdir --parents /var/opt/novell/eDirectory
mkdir –-parents /var/opt/novell/nici
mkdir –-parents /etc/opt/novell/eDirectory
```

3e ディレクトリをバインドマウントするために、次の内容を /etc/fstab に追加します。

```
<mount_point>/eDirectory_data /var/opt/novell/eDirectory none
defaults,bind 0 0

<mount_point>/nici_data /var/opt/novell/nici none defaults,bind 0 0

<mount_point>/eDirectory_nici_conf /etc/opt/novell/eDirectory none
defaults,bind 0 0
```

注 : VM 内でのすべての操作は、ルートユーザとして実行する必要があります。

OpenText eDirectory および OpenText Identity Console のインストール

前提条件

- ❒ 31 ページの 「システム要件」に記載されている要件を満たしていることを確認します。
- ❒ 103 ページの 「仮想マシン用のネットワークセキュリティグループの設定」で説明されているように、セキュリティグループを作成します。
- ❒ プロキシ VM は、強化され、セキュリティ保護されたサーバである必要があります。バックエンドサブネット内の VM およびプロキシ VM へのアクセスに必要な SSH 秘密鍵は、VNet 内には保存しないでください。クライアント上にのみ保存してください。このインスタンスに適したパフォーマンスとメモリを提供する VM サイズを選択します。
- ❒ プロキシ VM に対する追加のネットワークインターフェイスを作成し、そのインターフェイスにスタティックなパブリック IP アドレスを割り当てます。
- ❒ プロキシ VM 内に VNC サーバを設定します。VNC サーバは、強力なパスワードを使用して強化する必要があります。セキュリティ保護された通信を可能にするために、SSH トンネルを使用して VNC サーバに接続します。VNC サーバは、localhost からの接続についてのみリスンするように設定する必要があります。セッションのロックアウトを防止するために、画面ロックを無効にします。VNC サーバを使用した後、セッションを終了する必要があります。
- ❒ VM の /etc/hosts ファイルを手動で更新して、IP-Address Full-Qualified-Hostname Short-Hostname エントリを設定します。これは、逆引き DNS 検索を実行する際の Azure の制限を回避するためです。
- ❒ OpenText eDirectory/OpenText Identity Console を設定するバックエンドサブネット内の VM に SSH プロキシを使用して接続します。

  ```
  ssh -i edir_key.pem -A -J azureuser@<ssh_proxy_ip>
  azureuser@<instance_private_ip>
  ```

  注

  - 上記のサンプルコマンドで、edir_key.pem は、サーバキーを含むサンプルファイル名です。
  - SSH-Add コマンドを使用してエージェント内に識別情報ファイルを追加すれば、ログインするたびに識別情報ファイルを使用しないで済みます。

  VM のプライベート IP アドレスを表示するには、[**インスタンス**] > *[インスタンス]* > [**説明**] をクリックします。
- ❒ バックエンドサブネット内の VM に SLP ディレクトリエージェント (DA) サーバを設定します。SLP DA がデプロイされている VM に対して、NSG のインバウンドルールでポート 427 を開きます。slp.conf ファイルを編集して、DA 操作を有効にします。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*OpenSLP for OpenText eDirectory の設定*」を参照してください。

インストールと環境設定の手順

このセクションでは、Azure 環境に OpenText eDirectory と OpenText Identity Console をインストールして設定する手順について説明します。OpenText eDirectory をインストールした後、次の条件を満たしていることを確認する必要があります。

- EBA が有効になっている
- SNMP が無効になっている
- OpenText eDirectory がポート 389 でリスンしていない
- LDAP および HTTP サービスは、ECDSA 証明書のみを使用するように設定されている
- バックエンドサブネット内の Azure VM の SSH ポートへのアクセスは、不使用時は無効になっている
- OpenText iMonitor、eMBox、および DHost モジュールを無効にして、セキュリティを強化する。これらのモジュールを無効にした後、これらのモジュールに関係するすべてのアクティビティは、NDS ユーティリティのみを使用して実行する必要があります。

OpenText eDirectory のインストールと設定

1. OpenText eDirectory を設定するバックエンドサブネット内の VM に、セキュアコピー (scp) を使用して eDirectory_*<version>*_Linux_x86_64.tar.gz ファイルをコピーします (SSH プロキシを使用します)。

   ```
   scp -i <keyname> -o ProxyJump=vm-user@<ssh_proxy_ip>
   eDirectory_<version>_Linux_x86_64.tar.gz vm-user@<instance_ip>:/
   <directory>
   ```

2. OpenText eDirectory をインストールします。詳細については、「nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする」を参照してください。
3. OpenText eDirectory を設定します。詳細については、「ndsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する」を参照してください。たとえば、OpenText eDirectory をインストールして設定するためのコマンドのサンプルを次に示します。

   ```
   ndsconfig  new [-t <tree_name>] [-n <server context>] -a <admin FDN> [-
   w <admin password>] -P ldaps://<instance_ip> --configure-eba-now yes
   ```

4. openslp-server をインストールして、SLPD サービスを始動します。

Identity Console のインストールと設定

Identity Console を使用すると、Azure 環境での OpenText eDirectory の操作を管理できます。OpenText eDirectory をインストールした後、Azure VM に OpenText Identity Console をインストールする必要があります。

1. OpenText eDirectory を設定するバックエンドサブネット内のインスタンスに、セキュアコピー (scp) を使用して IdentityConsole_170_Containers.tar.gz ファイルをコピーします (SSH プロキシを使用します)。

```
scp -i <keyname> -o ProxyJump=vm-user@<ssh_proxy_ip>
IdentityConsole_170_Containers.tar.gz vm-user@<instance_ip>:/<directory>
```

2 OpenText Identity Console をインストールして設定します。詳細については、「Windows での OpenText Identity Console Workstation のインストール」を参照してください。OpenText Identity Console をインストールする前に、『OpenText™ Identity Console インストールガイド』の「System Requirements and Prerequisites for Workstation」セクションにあるシステム要件を参照してください。

3 Identity Console を実行している VM の自己署名証明書を、セキュア CA 署名証明書で置き換えます。詳細については、「OpenText iManager 用の一時的な自己署名証明書の置き換え (Replacing the Temporary Self-Signed Certificates for OpenText iManager)」を参照してください。

OpenText Identity Console の起動

OpenText Identity Console を起動するには、次の手順を実行します。

1 SSH トンネルを使用して、プロキシ VM の localhost 上で実行されている VNC サーバに接続します。

2 同じインスタンスにブラウザをインストールして起動します。

3 IP アドレスまたはツリー名を使用して、OpenText eDirectory ツリーを起動して接続します。

設定後のタスク

1 EBA が有効になっているかどうかを確認するため、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*EBA に関する情報の表示*」を参照してください。

2 証明書サーバで Suite B を有効にします。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*証明書サーバでの Suite B の有効化*」を参照してください。

3 1 番目の OpenText eDirectory サーバに対して AES 256 ビットツリーキーを設定します。詳細については、『*NICI Administration Guide (NICI 管理ガイド)*』の「Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)」を参照してください。

4 1 番目の OpenText eDirectory サーバにある CRL 配布ポイントを削除します。ポート 389 でのセキュリティ保護されていない LDAP アクセスはすべての OpenText eDirectory サーバで無効化されているため、ツリー CA の CRL は HTTP のみでダウンロードできるようにする必要があります。次の手順を実行して、CRL 配布ポイントを削除します。

   4a OpenText Identity Console に管理者としてログインします。

   4b ［**証明書管理**］>［**CA 管理**］>［**CRL**］>［CRL の作成］に移動します。

   4c ［**One**］をクリックします。HTTP CRL 配布ポイント (http://<instance_ip>:8028/crl/one.crl) 以外のすべての［**CRL 配布ポイント**］を選択して削除します。

   4d ［**適用**］をクリックし、［**閉じる**］をクリックします。

   4e ［**OneEC**］をクリックします。HTTP CRL 配布ポイント (http://<instance_ip>:8028/crl/oneec.crl) 以外のすべての［**CRL 配布ポイント**］を選択して削除します。

   4f ［**適用**］をクリックし、［**OK**］をクリックします。

5 OpenText Identity Console 証明書管理を使用して、サーバのデフォルト証明書を修復します。デフォルト証明書を修復するには、次の手順を実行します。

5a OpenText Identity Console に管理者としてログインします。

5b ［**証明書管理**］>［**デフォルト証明書**］に移動します。

5c 証明書を所有するサーバを選択して、［**次へ**］をクリックします。

5d ［**デフォルト証明書はすべて上書きされます**］を選択し、［**次へ**］をクリックします。

5e 実行するタスクを確認し、［**完了**］を選択し、［**OK**］をクリックします。

6 ECDSA 証明書と Suite B 暗号化を使用するように LDAP サービスと HTTP サービスを設定します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*ECDSA 証明書と Suite·B·Cipher を使用するための LDAP サービスと HTTP サービスの設定*」を参照してください。完了したら、OpenText eDirectory を再起動します。

7 SNMP サブエージェントがアンロードされているかどうかを確認する方法について、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*SNMP サーバモジュールのロードとアンロード*」を参照します。

8 OpenText eDirectory がポート 389 でリスンしていないことを確認します。

9 OpenText iMonitor、eMBox、DHost、および HTTP スタックを無効にします。

9a 次の手順を実行して、ツリー CA をホストしている OpenText eDirectory サーバの iMonitor、eMBox、および DHost を無効にします。

9a1 ndsmodules.conf ファイルを編集して、hconserv、imon、および embox をコメント化します。

9a2 OpenText eDirectory を再起動します。

9b 次の手順を実行して、OpenText eDirectory レプリカサーバの HTTP スタックを無効にします。

9b1 ndsmodules.conf ファイルを編集して、httpstk、hconserv、imon、および embox をコメント化します。

9b2 eDirectory を再起動します。

注 : httpstk は、コメント化の前に、ndsmodules.conf ファイル内で nds より上に配置されている必要があります。これにより、nds モジュールは HTTP スタックを有効にしなくなります。

---

10 アドバタイジング方式としてユニキャストを使用することを OpenText eDirectory に強制するように、SLP を設定します。バックエンドサブネット内の DA サーバの IP アドレスを提供するように、slp.conf ファイルを編集します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*環境設定パラメータ*」を参照してください。

注 : すべての OpenText eDirectory VM と OpenText Identity Console の設定が完了したら、SSH ポートへのアクセスを拒否し、必要な場合にのみ許可するように、Azure バックエンドサブネットのセキュリティルールを設定してください。

監査サービスのデプロイ

Azure に Common Event Format (CEF) 監査サービスをデプロイして、さまざまな OpenText eDirectory イベントを監査することができます。CEF 監査サービスをデプロイするには、次の手順を実行します。

1 VNet に監査サーバをインストールします。

2 ポート上でリスンするように監査サーバを設定します

注 : 監査サーバとしては Sentinel を使用することをお勧めします。

3 フロントエンドサブネットに新しいネットワークセキュリティグループルールを作成して次のように設定し、監査サーバが実行されている VM に関連付けます。

名前	ポート	ソース	宛先	説明
Auditing Server Port	TCP (監査サーバポート)	バックエンドサブネット	監査サーバの IP	OpenText eDirectory サーバからのイベントの受信を許可します

4 すべての OpenText eDirectory インスタンスで、次のように /etc/opt/novell/eDirectory/conf/auditlogconfig.properties ファイルを更新します。

```
log4j.appender.S.Host=<Auditing server ip>
log4j.appender.S.Port=<auditing server port>
```

5 OpenText Identity Console から、対応する CEF イベントを有効にします。詳細については、「監査する CEF イベントの設定 (https://www.netiq.com/documentation/edirectory-93/edir_admin/data/t44e7j6b8ufi.html#t44e7j6cx14i)」を参照してください。有効にされたイベントは、監査サーバに転送されます。

障害復旧

障害復旧は、OpenText eDirectory が実行されている VM でクラッシュが発生した場合に実行します。障害復旧を行うには、次の手順を実行します。

1 クラッシュした VM を停止し、データディスクとの関連付けを解除します。詳細については、「データディスクを Linux 仮想マシンから切断する方法」を参照してください。

2 クラッシュした VM と同じオペレーティングシステムを使用して、新しい VM を設定します。

3 新しい VM に同じバージョンの OpenText eDirectory をインストールします。

4 データディスクを新しい VM に接続し、ファイルシステムをマウントします。詳細については、「ポータルを利用し、データディスクを Linux VM に接続する」を参照してください。

5 ディレクトリをバインドマウントします。

ディレクトリをバインドマウントするには、/etc/fstab を次のように更新します。

```
<mount_point>/eDirectory_data /var/opt/novell/eDirectory none
defaults,bind 0 0

<mount_point>/nici_data /var/opt/novell/nici none defaults,bind 0 0

<mount_point>/eDirectory_nici_conf /etc/opt/novell/eDirectory none
defaults,bind 0 0
```

6 /etc/opt/novell/eDirectory/conf/nds.conf 内の IP アドレスを、現在の VM の IP アドレスに変更します。

7 OpenText eDirectory をアップグレードし、ヘルスチェックをスキップします。詳細については、「*OpenText eDirectory をアップグレードする*」を参照してください。

8 ndsrepair ユーティリティを使用して、ネットワークアドレスを修復します。詳細については、『*OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド*』の「DSRepair オプション」を参照してください。

9 ツリー CA の IP アドレスが変更された場合は、CRL 配布ポイントの IP アドレスを変更します。IP アドレスの変更方法の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド (https://www.netiq.com/documentation/edirectory-93/edir_admin/data/b1j4tpo3.html#b1j581ua)』の「CRL 設定オブジェクトのプロパティの表示と変更」を参照してください。

10 OpenText Identity Console 証明書管理を使用して、サーバのデフォルト証明書を修復します。デフォルト証明書を修復するには、次の手順を実行します。

10a OpenText Identity Console に管理者としてログインします。

10b [**証明書管理**] > [**デフォルト証明書**] に移動します。

10c 証明書を所有するサーバを選択して、[**次へ**] をクリックします。

10d [**デフォルト証明書はすべて上書きされます**] を選択し、[**次へ**] をクリックします。

10e 実行するタスクを確認し、[**完了**] を選択し、[**OK**] をクリックします。

11 新しい ECDSA 証明書を使用するように LDAP および HTTP サービスを設定します。

Windows プラットフォームへのデプロイ

デプロイメントの最初のステップでは、OpenText eDirectory で使用するために Azure サービスを準備する必要があります。詳細については、99 ページの 「Azure サービスの準備」を参照してください。Azure サービスを作成した後、次の手順を実行します。

- 114 ページの 「アプリケーションセキュリティグループ (ASG) の設定」
- 114 ページの 「仮想マシン用のネットワークセキュリティグループの設定」
- 117 ページの 「仮想マシンの作成とデプロイ」
- 118 ページの 「OpenText eDirectory データを格納するためのデータディスクの設定」
- 118 ページの 「OpenText eDirectory および OpenText Identity Console のインストール」
- 121 ページの 「障害復旧」

アプリケーションセキュリティグループ (ASG) の設定

アプリケーションセキュリティグループを使用すると、アプリケーション構造の自然な拡張としてネットワークセキュリティを設定できます。さらに、仮想マシンをグループ化し、それらのグループに基づいてネットワークセキュリティポリシーを定義することもできます。詳細については、「アプリケーションセキュリティグループ」を参照してください。

ネットワークセキュリティグループを設定する前に、次のアプリケーションセキュリティグループを作成する必要があります。

表 5-4 *アプリケーションセキュリティグループ*

名前	説明
OpenText eDirectory	OpenText eDirectory を設定するすべての仮想マシンのネットワークインターフェイスが含まれます
eDirectory_CA	ツリー認証局 (CA) をホストする OpenText eDirectory サーバを設定する仮想マシンのネットワークインターフェイスが含まれます
OpenText Identity Console	OpenText Identity Console を設定する仮想マシンのネットワークインターフェイスが含まれます

仮想マシン用のネットワークセキュリティグループの設定

セキュリティグループは、仮想ネットワーク内の 1 つまたは複数の仮想マシンに割り当てることができる仮想ファイアウォールルールのセットです。

詳細については、「セキュリティグループ」を参照してください。

Azure に OpenText eDirectory をデプロイするには、次のネットワークセキュリティグループを作成します : eDirectory_NSG_1、eDirectory_NSG_2、および IdentityConsole_NSG。これらのセキュリティグループを作成して次のポートルールを設定し、デフォルトのセキュリティルールを上書きします。

1. **eDirectory_NSG_1:** この NSG は、OpenText eDirectory ツリーの CA をホストする仮想マシンに関連付ける必要があります。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	NCP	TCP 524	OpenText eDirectory サブネット	ASG (OpenText eDirectory)	許可	OpenText eDirectory サブネット内の OpenText eDirectory の NCP トラフィックを許可します

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
110	HTTP CRL	TCP 8028	仮想ネットワーク	eDirectory_CA (ASG)	許可	ツリー CA によって発行された証明書を使用して設定されている VNet にサービスが存在する場合に、VNet から OpenText eDirectory ツリーの CRL にアクセスするために必要です。
120	LDAPS	TCP 636	OpenText eDirectory サブネット	ASG (OpenText eDirectory)	許可	OpenText eDirectory サブネット内でセキュリティ保護された LDAP トラフィックを許可します
130	SLP	Any 427	OpenText eDirectory サブネット	ASG (OpenText eDirectory)	許可	OpenText eDirectory サブネット内の SLP トラフィックを許可します
140	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

注

- OpenText eDirectory サブネットは、OpenText eDirectory がインストールされている仮想マシンを参照します。
- OpenText eDirectory サーバは、LDAP ポート 389 でリスンするように設定してはならず、OpenText eDirectory に割り当てられているセキュリティグループに対してポート 389 へのアクセスを許可してはなりません。また、HTTP ポートへのアクセスを許可するのは、ツリーの CA をホストしている OpenText eDirectory サーバに割り当てられたセキュリティグループに対してのみにする必要があります。

2. **eDirectory_NSG_2:** この NSG は、OpenText eDirectory ツリー CA 以外の OpenText eDirectory サーバをホストしているすべての仮想マシンに関連付ける必要があります。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	NCP	TCP 524	OpenText eDirectory サブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内で OpenText eDirectory に対する NCP トラフィックを許可します
110	LDAPS	TCP 636	OpenText eDirectory サブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内でセキュリティ保護された LDAP トラフィックを許可します
120	SLP	Any 427	OpenText eDirectory サブネット	ASG (OpenText eDirectory)	許可	バックエンドサブネット内の SLP トラフィックを許可します
130	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

3. **IdentityConsole_NSG:** この NSG は、OpenText Identity Console をホストする仮想マシンに関連付ける必要があります。次の NSG ルールを使用すると、プロキシサーバのみから OpenText Identity Console サーバにアクセスできるようになります。

優先度	名前	ポート範囲	ソース	宛先	アクション	説明
100	HTTPS	TCP 9000	仮想ネットワーク - サブネット	OpenText Identity Console (ASG)	許可	サブネット内のプロキシにアクセスするセキュリティで保護された HTTP トラフィックを許可します
110	All Traffic	すべて (All)	すべて	すべて	拒否	すべてのインバウンドトラフィックを拒否します

仮想マシンの作成とデプロイ

サポートされているプラットフォームのいずれかで仮想マシン (VM) を作成して起動します。VM を作成して起動する方法の詳細については、「Azure portal で Windows 仮想マシンを作成する」を参照してください。インスタンスを作成および起動する際には、次の手順も実行する必要があります。

1 最初の OpenText eDirectory サーバを設定する VM に eDirectory_NSG_1 を関連付け、その他のすべての OpenText eDirectory サーバを設定する VM に eDirectory_NSG_2 を関連付け、Identity Console を設定する VM に IdentityConsole_NSG を関連付けます。セキュリティグループの詳細については、114 ページの 「仮想マシン用のネットワークセキュリティグループの設定」を参照してください。

2 Windows VM に接続するための Bastion ホストを作成します。

Azure 上の Windows VM にアクセスするには、Bastion を作成します。Bastion を設定し、Windows VM に接続する方法の詳細については、Bastion の設定に関する記事を参照してください。

OpenText eDirectory データを格納するためのデータディスクの設定

Azure VM のクラッシュが発生した場合に OpenText eDirectory のデータと設定が失われるのを防ぐには、データディスクを設定する必要があります。OpenText eDirectory のデータと設定を復元する方法の詳細については、121 ページの 「障害復旧」を参照してください。VM を作成した後、次の手順を実行して、OpenText eDirectory をデプロイできるように VM を準備します。

1. データディスクを作成して接続するために、「Azure portal を使用して Windows VM にマネージドデータディスクを接続する」の手順を実行します。
2. VM にログインし、NTFS ファイルシステムを使用してデータディスクをフォーマットし、データディスクをマウントします。データディスクをフォーマットおよびマウントする方法の詳細については、新しいデータディスクを初期化する方法に関する記事を参照してください。

OpenText eDirectory および OpenText Identity Console のインストール

前提条件

- ❒ 75 ページの 「システム要件」に記載されている要件を満たしていることを確認します。
- ❒ 114 ページの 「仮想マシン用のネットワークセキュリティグループの設定」で説明されているように、セキュリティグループを作成します。
- ❒ このインスタンスに適したパフォーマンスとメモリを提供する VM サイズを選択します。
- ❒ OpenText eDirectory のデータを保存するためのデータディスクを設定します。
- ❒ Windows VM に接続するための Bastion を設定します。

インストールと環境設定の手順

このセクションでは、Azure 環境に OpenText eDirectory と OpenText Identity Console をインストールして設定する手順について説明します。OpenText eDirectory をインストールした後、次の条件を満たしていることを確認する必要があります。

- EBA が有効になっている
- SNMP が無効になっている
- OpenText eDirectory がポート 389 でリスンしていない
- LDAP および HTTP サービスは、ECDSA 証明書のみを使用するように設定されている
- Azure VM サブネットの RDP ポートへのアクセスは、不使用時は無効になっている
- OpenText iMonitor、eMBox、および DHost モジュールを無効にして、セキュリティを強化する。これらのモジュールを無効にした後、これらのモジュールに関係するすべてのアクティビティは、NDS ユーティリティのみを使用して実行する必要があります。

OpenText eDirectory のインストールと設定

1 eDirectory_*<version>*_Windows_x86_64.exe ファイルをダウンロードします。

2 eDirectory をインストールします。詳細については、「OpenText eDirectory CE 24.4 (v9.3) の Windows へのインストールまたはアップグレード」を参照してください。

OpenText Identity Console のインストールと設定

OpenText Identity Console の管理コンソールを使用すると、Azure 環境での OpenText eDirectory の操作を管理できます。OpenText eDirectory をインストールした後、Azure VM に Identity Console をインストールする必要があります。

1 ファイル IdentityConsole_*<version>*_win_x86_64.zip をダウンロードします。

2 OpenText Identity Console をインストールして設定します。詳細については、「Windows での OpenText Identity Console のインストール (https://www.netiq.com/documentation/identity-console/identity_console-install/data/t4j68b4oo7x1.html)」を参照してください。OpenText Identity Console をインストールする前に、『OpenText Identity Console インストールガイド (https://www.netiq.com/documentation/identity-console/identity_console-install/data/bookinfo.html)』の「システム要件 (https://www.netiq.com/documentation/identity-console/identity_console-install/data/t4ja6mu8vquv.html)」セクションにあるシステム要件を参照してください。

OpenText Identity Console の起動

OpenText Identity Console を起動するには、次の手順を実行します。

1 Bastion ホストのユーザインターフェイスを開きます。

2 新しいブラウザタブで、OpenText Identity Console を起動します。

3 OpenText Identity Console のポートと URL を入力します。

設定後のタスク

1 EBA が有効になっているかどうかを確認するため、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*EBA に関する情報の表示*」を参照してください。

2 証明書サーバで Suite B を有効にします。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*Certificate Server 上での Suite B の有効化*」を参照してください。

3 1 番目の OpenText eDirectory サーバに対して AES 256 ビットツリーキーを設定します。詳細については、『*NICI Administration Guide (NICI 管理ガイド)*』の「Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)」を参照してください。

4 1 番目の OpenText eDirectory サーバにある CRL 配布ポイントを削除します。ポート 389 でのセキュリティ保護されていない LDAP アクセスはすべての OpenText eDirectory サーバで無効化されているため、ツリー CA の CRL は HTTP のみでダウンロードできるようにする必要があります。次の手順を実行して、CRL 配布ポイントを削除します。

   4a OpenText Identity Console に管理者としてログインします。

   4b ［**証明書管理**］> ［**CA 管理**］> ［**CRL**］> ［CRL の作成］に移動します。

4c ［One］をクリックします。HTTP CRL 配布ポイント (http://<instance_ip>:8028/crl/one.crl) 以外のすべての［CRL 配布ポイント］を選択して削除します。

4d ［保存］をクリックし、［OK］をクリックします。

4e ［OneEC］をクリックします。HTTP CRL 配布ポイント (http://<instance_ip>:8028/crl/oneec.crl) 以外のすべての［CRL 配布ポイント］を選択して削除します。

4f ［保存］をクリックし、［OK］をクリックします。

5 OpenText Identity Console 証明書管理を使用して、サーバのデフォルト証明書を修復します。デフォルト証明書を修復するには、次の手順を実行します。

5a OpenText Identity Console に管理者としてログインします。

5b ［証明書管理］>［デフォルト証明書］に移動します。

5c 証明書を所有するサーバを選択して、［次へ］をクリックします。

5d ［デフォルト証明書はすべて上書きされます］を選択し、［次へ］をクリックします。

5e 実行するタスクを確認し、［完了］を選択し、［OK］をクリックします。

6 ECDSA 証明書と Suite B 暗号化を使用するように LDAP サービスと HTTP サービスを設定します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*ECDSA 証明書と Suite·B·Cipher を使用するための LDAP サービスと HTTP サービスの設定*」を参照してください。完了したら、OpenText eDirectory を再起動します。

7 SNMP サブエージェントがアンロードされているかどうかを確認する方法について、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*SNMP サーバモジュールのロードとアンロード*」を参照します。

8 OpenText eDirectory がポート 389 でリスンしていないことを確認します。

9 OpenText iMonitor、eMBox、DHost、および HTTP スタックを無効にします。

9a 次の手順を実行して、ツリー CA をホストしている OpenText eDirectory サーバの OpenText iMonitor、eMBox、および DHost を無効にします。

9a1 ndsmodules.conf ファイルを編集して、hconserv、imon、および embox をコメント化します。

9a2 OpenText eDirectory を再起動します。

9b 次の手順を実行して、OpenText eDirectory レプリカサーバの HTTP スタックを無効にします。

9b1 ndsmodules.conf ファイルを編集して、httpstk、hconserv、imon、および embox をコメント化します。

9b2 OpenText eDirectory を再起動します。

注 : httpstk は、コメント化の前に、ndsmodules.conf ファイル内で nds より上に配置されている必要があります。これにより、nds モジュールは HTTP スタックを有効にしなくなります。

---

10 アドバタイジング方式としてユニキャストを使用することを OpenText eDirectory に強制するように、SLP を設定します。バックエンドサブネット内の DA サーバの IP アドレスを提供するように、slp.conf ファイルを編集します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*環境設定パラメータ*」を参照してください。

注 : すべての OpenText eDirectory VM と Identity Console の設定が完了したら、SSH ポートへのアクセスを拒否し、必要な場合にのみ許可するように、Azure バックエンドサブネットのセキュリティルールを設定してください。

障害復旧

Windows 仮想マシンで障害復旧を行う場合は、次の手順を実行します。

注 : ツリーの設定後は、dsbk バックアップを使用して nici ファイルのバックアップを取ることをお勧めします。詳細については、「Windows 上での障害復旧計画」を参照してください。

1. クラッシュした VM を停止し、データディスクとの関連付けを解除します。詳細については、「Windows 仮想マシンからディスクを切断する方法」を参照してください。
2. クラッシュした VM と同じオペレーティングシステムを使用して、新しい VM を設定します。
3. 新しい VM に同じバージョンの OpenText eDirectory をインストールします。
4. nici バックアップを使用して dsbk バックアップファイルを新しい VM にコピーします。ndsd サービスを停止し、nici で dsbk 復元を実行します。
5. データディスクを新しい VM に接続し、ファイルシステムをマウントします。詳細については、「ポータルを使用してデータディスクを Windows VM に接続する」を参照してください。
6. 新しい VM で ndsd サービスを始動します。
7. OpenText eDirectory をアップグレードします。詳細については、『OpenText™ eDirectory CE 24.4 (v9.3) インストールガイド』の「*OpenText eDirectory をアップグレードする*」を参照してください。
8. ndsrepair ユーティリティを使用して、ネットワークアドレスを修復します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「DSRepair オプション」を参照してください。
9. ツリー CA の IP アドレスが変更された場合は、CRL 配布ポイントの IP アドレスを変更します。IP アドレスの変更方法の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド (https://www.netiq.com/documentation/edirectory-93/edir_admin/data/b1j4tpo3.html#b1j581ua)』の「CRL 設定オブジェクトのプロパティの表示と変更」を参照してください。
10. OpenText Identity Console 証明書管理を使用して、サーバのデフォルト証明書を修復します。デフォルト証明書を修復するには、次の手順を実行します。

    10a OpenText Identity Console に管理者としてログインします。

    10b [**証明書管理**] > [**デフォルト証明書**] に移動します。

    10c 証明書を所有するサーバを選択して、[**次へ**] をクリックします。

10d［デフォルト証明書はすべて上書きされます］を選択し、［次へ］をクリックします。

10e 実行するタスクを確認し、［完了］を選択し、［OK］をクリックします。

11 新しい ECDSA 証明書を使用するように LDAP および HTTP サービスを設定します。

Microsoft Azure コンテナインスタンスへの OpenText eDirectory コンテナのデプロイ

OpenText eDirectory は、Microsoft Azure コンテナインスタンスへの OpenText eDirectory コンテナのデプロイをサポートしています。デプロイメントプロセスはシンプルで、時間効率が向上します。OpenText eDirectory コンテナは、自己完結型であり、独自に実行できる Docker イメージとしてレジストリにプッシュされます。このセクションでは、Azure に OpenText eDirectory Docker イメージをデプロイするプロセスについて説明します。

コンテナのデプロイのチェックリスト

Azure コンテナをデプロイ用にセットアップするには、次のチェックリストの手順を実行することをお勧めします。

	チェックリストの項目
❑	1. OpenText eDirectory Docker コンテナのイメージを保存および管理するために、Azure コンテナレジストリを作成する必要があります。https://portal.azure.com/ の Azure ポータルにサインインし、「Azure portal を使用して Azure コンテナレジストリを作成する」の手順に従ってコンテナレジストリを作成します。
❑	2. 次に、Azure ストレージアカウントを設定する必要があります。このアカウントには、HTTP または HTTPS 経由で世界中のどこからでもアクセスできるすべてのデータが保存されます。Azure portal にサインインし、「Azure Storage アカウントを作成する」の手順に従って Azure ストレージアカウントを作成します。
❑	3. ストレージアカウントを作成したら、Azure ファイル共有を作成します。共有ファイルシステムは、コンテナが実行するインスタンスに関係なく、コンテナがファイルシステムにアクセスするために必要です。Azure ファイル共有を作成する際の考慮事項とプロセスの詳細については、「Azure ファイル共有を作成する方法」を参照してください。
❑	4. 仮想ネットワークを作成する必要があります。Azure Virtual Network を使用すると、Azure 仮想マシン (VM) などの多くの種類の Azure リソースが、Azure インターネット内で相互に、またオンプレミスのネットワークと安全に通信することができます。仮想ネットワークの作成に関するページの手順に従います。
❑	5. 次に、OpenText eDirectory Docker イメージをコンテナレジストリにプッシュします。詳細については、123 ページの 「Docker イメージの Azure コンテナレジストリへのプッシュ」を参照してください。
❑	6. OpenText eDirectory Docker イメージの tarball にバンドルされている YAML ファイルを編集して保存します。詳細については、124 ページの 「YAML 設定ファイルの編集」を参照してください。

	チェックリストの項目
❑	7. 最後に、この YAML ファイルを使用して Azure コンテナインスタンスを作成します。詳細については、128 ページの 「Azure コンテナインスタンスの作成」を参照してください。

Docker イメージの Azure コンテナレジストリへのプッシュ

Docker イメージを Azure コンテナレジストリにプッシュするには :

1 OpenText eDirectory Docker イメージの tarball をソフトウェアライセンスとダウンロードポータルからダウンロードします。

2 ダウンロードしたビルドを untar して Docker イメージを抽出し、次のコマンドを使用してローカル Docker レジストリにロードします。

```
tar -xvf eDirectory_93x_Container.tar.gz
docker load --input eDirectory_93x/eDirectory_93x.tar.gz
```

この docker load コマンドは、tar アーカイブから edirectory:9.3.x という名前の Docker イメージをロードします。

3 次のコマンドを使用して、Docker イメージにレジストリ名をタグ付けします。

```
Docker tag edirectory:<image-id> <registry-name>.azurecr.io/
<repository-name-tag>
```

各要素は次のように指定します。

edirectory:<image-id> は、ローカルの Docker イメージ ID です。

<registry-name> は、プライベートレジストリの名前です。

<repository-name-tag> は、Docker イメージに割り当てるタグです。

例 : docker tag edirectory:93x myazureregistry.azurecr.io/edir93

4 次のコマンドを使用して、Docker イメージをレジストリにプッシュします。

```
docker push <registry-name>.azurecr.io/<repository-name-tag>
```

各要素は次のように指定します。

<registry-name> は、プライベートレジストリの名前です。

<repository-name-tag> は、Docker イメージに割り当てたタグです。

例 : docker push myazureregistry.azurecr.io/edir93x

YAML 設定ファイルの編集

YAML ファイルは、コンテナグループをすばやく簡単に作成または更新する手段です。Azure コマンドラインで az container create コマンドを実行すると、YAML ファイルからグループの設定が読み込まれます。Azure コンテナインスタンスを作成する前に、OpenText eDirectory Docker イメージの tarball にバンドルされている YAML ファイルを編集してアップロードする必要があります。

次のセクションでは、主要なプロパティを強調するコメント (# 記号付き) を含む、YAML ファイルのスキーマを示します。このスキーマのプロパティの説明については、「プロパティ値」セクションを参照してください。

```
name:  # Provide the name of the container group
apiVersion: '2019-12-01'
properties: # Properties of container group
  containers: # The containers within the container group
  - name: test-edir93 # Provide a name of container instance
    properties: # Properties of a container instance
      image: myazureregistry.azurecr.io/edir93 # Provide the name of the
container image used to create the container instance
      command:
      - /start.sh #Not optional
      - 'new -t 93edir1 -n novell -a admin.novell -w xx -S server1 -i -B
@1524 -o 1028 -O 1030 -L 1389 -l 1636 --configure-eba-now no' #Sample
command
     ports: # External-facing ports exposed on the instance, must also be
set in group ipAddress property
      - protocol: tcp
       port: 1030 #HTTPS port. Depends on the parameter provided in command
for -O
      - protocol: tcp
       port: 1389 #LDAP port. Depends on the parameter provided in command
for -L
      environmentVariables:
      - name: YAML_MODE #Mandatory variable. Required if yaml file is used
for configuring edirectory
        value: '1'
      resources: # Minimum resource requirements of the instance
        requests:
          memoryInGB: 2
          cpu: 1.5
      volumeMounts: # Array of volume mounts for the instance
      - name: configvolume
        mountPath: /config #Mandatory. Do not change this path.
  imageRegistryCredentials: # Credentials to pull a private image
  - server: myazureregistry.azurecr.io #Provide the image registry server
name
    username: myazureregistry #Provide the image registry username
password: xxxx #Provide the image registry password
  #restartPolicy: Never #Optional
  ipAddress: # IP address configuration of container group
    ports:
    - protocol: tcp
      port: 1030 #HTTPS port. Depends on the parameter provided in command
```

```
for -O
    - protocol: tcp
      port: 1389 #LDAP port. Depends on the parameter provided in command
for -L
    type: Private
  networkProfile: # Virtual network profile for container group
    id: subscriptions/xxx-xxx-xxx/resourceGroups/myresourcegroup/
providers/Microsoft.Network/networkProfiles/aci-network-profile-
VirtualNetmy-containersubnet
#Can be found by running the command, az network profile list --resource-
group myResourceGroup --query [0].id --output tsv, if no output then run
the first time using azure cloud shell
  osType: Linux
  volumes: # Array of volumes available to the instances
  - name: configvolume
    azureFile:
      shareName: testshare  #Provide the name of the Azure File share to be
mounted as a volume
      storageAccountName: myfileshare  #Provide name of the storage account
       storageAccountKey: xxx_xxx #Provide storage account access key
```

プロパティ値

次の表は、スキーマで設定する必要がある値を示しています。

名前	型	必須	値
containerGroups オブジェクト			
Name	文字列	対応	コンテナグループの名前。コンテナ名も指定できます。
apiVersion	enum	対応	リソース API のバージョン。たとえば、「2019-12-01」のように指定します。
プロパティ	object	対応	コンテナグループのプロパティ。properties でサポートされる値を探すには、この表の ContainerGroupProperties オブジェクトを参照してください。
ContainerGroupProperties オブジェクト			
containers	array	対応	コンテナグループ内のコンテナ。containers でサポートされる値を探すには、この表の Container オブジェクトを参照してください。
imageRegistryCredentials	array	非対応	コンテナグループの作成に使用されるイメージレジストリ資格情報。imageRegistryCredentials でサポートされる値を探すには、この表の ImageRegistryCredential オブジェクトを参照してください。

名前	型	必須	値
restartPolicy	enum	非対応	コンテナグループ内のすべてのコンテナに対するポリシーを再起動します。 ◆ - Always: 常に再起動 ◆ - OnFailure: 失敗時に再起動 ◆ - Never: 再起動しない
ipAddress	object	非対応	コンテナグループの IP アドレスのタイプ。ipAddress でサポートされる値を探すには、この表の IPAddress オブジェクトを参照してください。
osType	enum	対応	コンテナグループ内のコンテナに必要なオペレーティングシステムのタイプ。- Windows または Linux
volumes	array	非対応	このコンテナグループ内のコンテナによってマウントできるボリュームのリスト。volumes でサポートされる値を探すには、この表の Volume オブジェクトを参照してください。
コンテナオブジェクト			
name	文字列	対応	コンテナインスタンスの、ユーザが指定した名前。たとえば、「test-edir93」とします。
プロパティ	object	対応	コンテナインスタンスのプロパティ。properties でサポートされる値を探すには、この表の ContainerProperties オブジェクトを参照してください。
ContainerProperties オブジェクト			
image	文字列	対応	コンテナインスタンスの作成に使用するイメージの名前。たとえば、myazureregistry.azurecr.io/edir93 (ステップ 3)。
command	array	非対応	コンテナインスタンス内で実行するコマンドを exec 形式で指定します。OpenText eDirectory Docker コンテナは、Docker コマンドで ndsconfig ユーティリティのすべてのパラメータを受け入れます。ndsconfig ユーティリティの詳細については、181 ページの 「ndsconfig ユーティリティ」を参照してください。
ports	array	非対応	コンテナインスタンスの公開ポート。ports でサポートされる値を探すには、この表の ContainerPort オブジェクトを参照してください。
environmentVariables	array	非対応	コンテナインスタンスで設定する環境変数。environmentVariables でサポートされる値を探すには、この表の EnvironmentVariable オブジェクトを参照してください。

名前	型	必須	値
リソース	object	対応	コンテナインスタンスのリソース要件。resources でサポートされる値を探すには、この表の ResourceRequirements オブジェクトを参照してください。
volumeMounts	array	非対応	コンテナインスタンスで使用可能なボリュームマウント。volumeMounts でサポートされる値を探すには、この表の VolumeMount オブジェクトを参照してください。
ContainerPort オブジェクト			
プロトコル	enum	非対応	ポートに関連付けられているプロトコル。- TCP または UDP
port	integer	対応	ポート番号。
EnvironmentVariable オブジェクト			
name	文字列	対応	環境変数の名前。
value	文字列	非対応	環境変数の値。
ResourceRequirements オブジェクト			
要求	object	対応	このコンテナインスタンスのリソース要求。requests でサポートされている値を探すには、この表の ResourceRequests オブジェクトを参照してください。
ResourceRequests オブジェクト			
memoryInGB	number	対応	このコンテナインスタンスのメモリ要求 (GB)。
cpu	number	対応	このコンテナインスタンスの CPU 要求。
VolumeMount オブジェクト			
name	文字列	対応	ボリュームマウントの名前。
mountPath	文字列	対応	ボリュームをマウントするコンテナ内のパス。コロン (:) は含めないでください。
ImageRegistryCredential オブジェクト			
サーバ	文字列	対応	Docker イメージのレジストリサーバ。「http」や「https」などのプロトコルは付けません。チェックリストのステップ 1 で作成したレジストリ名を指定します (「コンテナのデプロイのチェックリスト」を参照)。
ユーザ名	文字列	非対応	プライベートレジストリのユーザ名。詳細については、ユーザ名情報の検索方法に関するページを参照してください。

名前	型	必須	値
password	文字列	非対応	プライベートレジストリのパスワード。詳細については、Azure レジストリパスワードの検索方法に関するページを参照してください。
IPAddress オブジェクト			
ports	array	対応	コンテナグループに公開されているポートのリスト。ports でサポートされる値を探すには、この表の Port オブジェクトを参照してください。
type	enum	対応	IP がパブリックインターネットに公開されているか、プライベート VNET に公開されているかを示します。- Public または Private
Port オブジェクト			
プロトコル	enum	非対応	ポートに関連付けられているプロトコル。- TCP または UDP
port	integer	対応	ポート番号。
Volume オブジェクト			
name	文字列	対応	ボリュームの名前。
azureFile	object	非対応	Azure ファイルボリューム。azureFile でサポートされる値を探すには、この表の AzureFileVolume オブジェクトを参照してください。
AzureFileVolume オブジェクト			
shareName	文字列	対応	ボリュームとしてマウントする Azure ファイル共有の名前。チェックリストのステップ 3 で作成したストレージファイル共有名を指定します (「コンテナのデプロイのチェックリスト」を参照)。
storageAccountName	文字列	対応	Azure ファイル共有を含むストレージアカウントの名前。
storageAccountKey	文字列	非対応	Azure ファイル共有へのアクセスに使用されるストレージアカウントのアクセスキー。ストレージアカウントキーを見つけるには、「アカウントのアクセスキーを表示する」を参照してください。

Azure コンテナインスタンスの作成

次のコマンドを実行して、前に編集した YAML ファイルを使用してコンテナグループ内にコンテナを作成します。

```
az container create --resource-group <MyResourceGroup> --name <container-instance-name>  --file <containerGroup.yaml>
```

各要素は次のように指定します。

<MyResourceGroup> は、リソースグループの名前を表します。必須フィールド。

<container-instance-name> は、コンテナグループ内のコンテナインスタンスの名前を表します。

<containerGroup.yaml> は、入力ファイルへのパスを表します。

例：az container create --resource-group MyResourcegroup --name edir93 --file edir-azure-conf.yaml

6 Amazon Web Services EC2 での OpenText eDirectory のデプロイ

OpenText eDirectory は、Amazon Web Services (AWS) EC2 インスタンスにデプロイできます。サポートされているオペレーティングシステムの詳細については、31 ページの 「システム要件」を参照してください。

次のセクションでは、AWS に OpenText eDirectory をデプロイするためのさまざまな前提条件と手順について説明します。

- 131 ページの 「前提条件」
- 131 ページの 「デプロイメントの手順」
- 146 ページの 「AWS での OpenText eDirectory デプロイメントのサイジングガイダンス」

前提条件

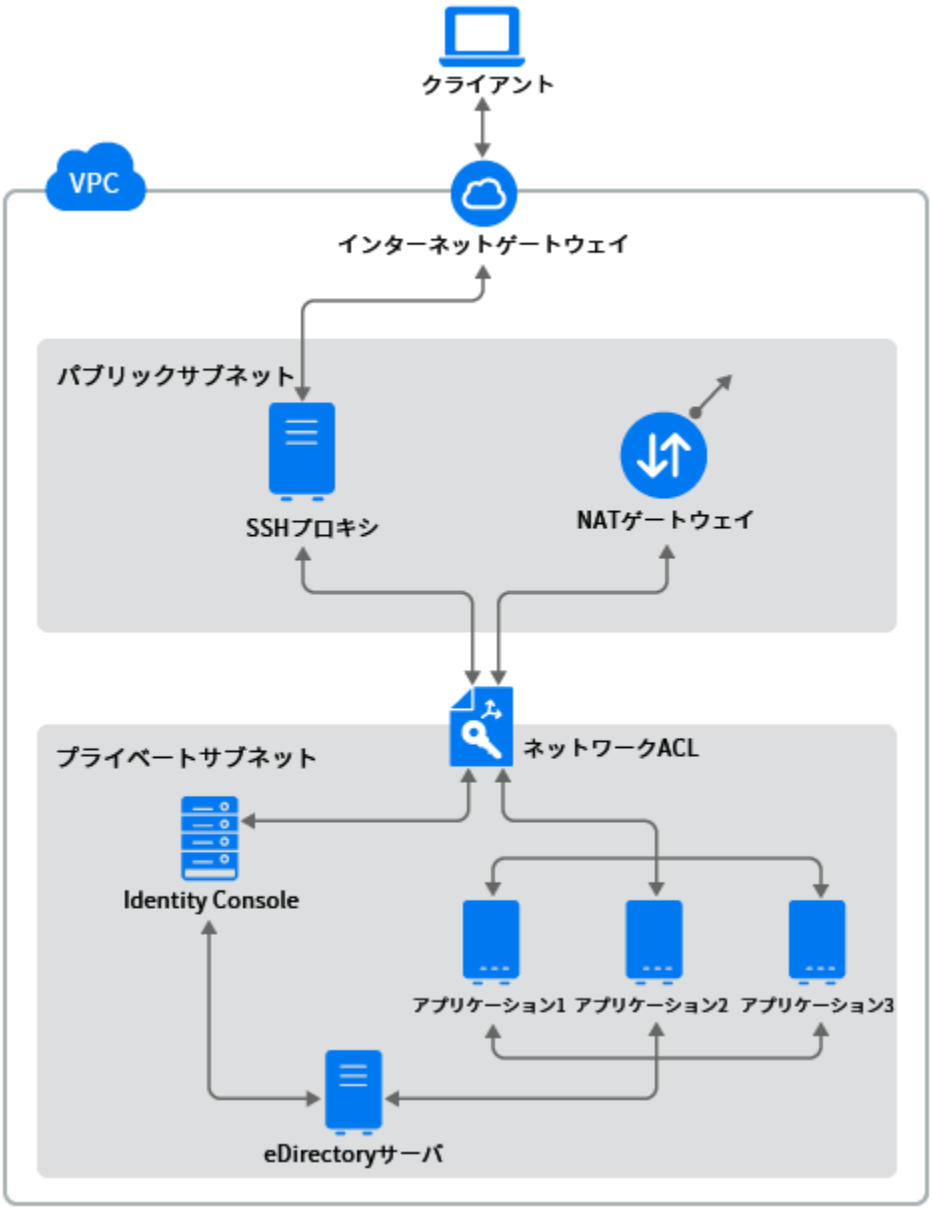
OpenText eDirectory のシステム要件に加えて、次の要件を満たしていることを確認してください。

- AWS EC2 の管理アカウント。
- インスタンスにコピーできるように、OpenText eDirectory インストーラ (tarball ファイル) をダウンロードし、展開してあること。
- クライアントマシンから AWS EC2 インスタンスに接続する SSH クライアント。

デプロイメントの手順

OpenText eDirectory は、Amazon VPC 内のプライベートサブネットにのみデプロイされる必要があります。図 5-1 は、以降のセクションで使用されるサンプル展開を示しています。

図 6-1 AWS EC2 上での OpenText eDirectory のデプロイメント



注

- SSH プロキシは、管理者が SSH を使用して接続し、SSH エージェント転送を使用してプライベートサブネット内の他のインスタンスに接続する、パブリックサブネット内の要塞ホストです。
- OpenText eDirectory にアクセスする必要があるアプリケーションは、プライベートサブネットにデプロイする必要があります。これらのアプリケーションにインターネットからアクセスする必要がある場合は、パブリックサブネット内に AWS EC2 ロードバランサーを設定して、アクセスを可能にします。詳細については、「Application Load Balancer の作成」を参照してください。

デプロイメント手順は、次のステップで構成されます。

- 133 ページの 「AWS 仮想プライベートクラウドの準備」
- 134 ページの 「ネットワーク ACL の設定」
- 137 ページの 「セキュリティグループの設定」
- 138 ページの 「SSH キーペアを作成する」
- 138 ページの 「インスタンスの作成とデプロイ」
- 139 ページの 「OpenText eDirectory データの保存用に EBS ボリュームを設定する」
- 140 ページの 「OpenText eDirectory および OpenText Identity Console のインストール」
- 144 ページの 「監査サービスのデプロイ」
- 144 ページの 「障害復旧」

AWS 仮想プライベートクラウドの準備

このセクションでは、OpenText eDirectory で使用するために AWS VPC を設定する一般的な手順について概説します。詳細については、『Amazon Elastic Compute Cloud Documentation』を参照してください。

以下の手順を実行して、AWS VPC サービスを作成します。

1 AWS 管理コンソールにログインします。

2 次のサービスを作成します。

サービス	説明
VPC	Amazon VPC コンソールを使用して、VPC を作成することができます。VPC の作成方法の詳細については、「VPC の作成」を参照してください。 VPC の概要に関する詳細については、「Amazon Virtual Private Cloud ドキュメント」を参照してください。
重要 : [VPC ウィザードの起動] を使用して VPC を作成すると、VPC 用に 2 つのサブネット、インターネットゲートウェイ、およびルートテーブルと NAT ゲートウェイが作成されます。これらの項目を表示または編集するには、次の手順に従います。	
サブネット	VPC 作成の一環として、2 つのサブネットが作成されます。パブリックおよびプライベートサブネットです。OpenText eDirectory は、プライベートサブネットにデプロイする必要があります。図 5-1 に示したとおり、OpenText eDirectory にアクセスするすべてのアプリケーションは、同じプライベートサブネットにデプロイする必要があります。プライベートサブネット内のインスタンスへの SSH アクセスは、パブリックサブネット内の SSH プロキシを経由して行う必要があります。詳細については、「VPC とサブネット (https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html)」を参照してください。

サービス	説明
インターネットゲートウェイ	インターネットゲートウェイは、図 5-1 に示したとおり、SSH プロキシへの SSH 接続を可能にするために必要です。VPC を使用してインターネットゲートウェイを作成および接続する方法の詳細については、「インターネットゲートウェイ」を参照してください。
ルートテーブル	ルートテーブルを作成する方法の詳細については、「ルートテーブル」を参照してください。
NAT ゲートウェイ	オペレーティングシステムの更新をダウンロードするためのプライベートサブネット内のインスタンスには、NAT ゲートウェイが必要です。NAT ゲートウェイの作成に関する詳細については、「NAT ゲートウェイ」を参照してください。
Elastic IP アドレス	Elastic IP アドレスは、SSH プロキシおよび NAT ゲートウェイを実行するインスタンスに割り当てる必要のあるパブリックなスタティック IP アドレスです。Elastic IP アドレスを作成する方法の詳細については、「Elastic IP アドレスを VPC 内のリソースに関連付ける (https://docs.aws.amazon.com/vpc/latest/userguide/vpc-eips.html#WorkWithEIPs)」を参照してください。

ネットワーク ACL の設定

ネットワークアクセス制御リスト (ACL) は、1 つまたは複数のサブネットに対するトラフィックを制御するファイアウォールとして機能する、VPC 用のセキュリティのオプション層です。詳細については、「ネットワーク ACL」を参照してください。

このセクションでは、プライベートサブネットにネットワーク ACL を作成するためのルールについて概説します。プライベートサブネットのネットワーク ACL に対して、次のルールを設定してください。

- インバウンドルール :

表 6-1

ルール	タイプ	ポート範囲	ソース	アクション	説明
10	SSH	TCP 22	*<SSH Proxy IP Address>*/32	許可	パブリックサブネット内の SSH プロキシの IP アドレスからのインバウンド SSH トラフィックを許可します

ルール	タイプ	ポート範囲	ソース	アクション	説明
15	HTTPS	TCP 9000	*<SSH Proxy IP Address>*/32	許可	パブリックサブネット内の SSH プロキシの IP アドレスからのインバウンド HTTPS トラフィックを許可します
20	カスタム TCP ルール	TCP 32768-65535	0.0.0.0/0	許可	サブネットからの要求に対するインターネット上のホストからのインバウンド返信トラフィックを許可します
*	All Traffic	すべて (All)	0.0.0.0/0	拒否	前のルールによってまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可能)

- アウトバウンドルール :

表 6-2

ルール	タイプ	ポート範囲	宛先	アクション	説明
10	カスタム TCP ルール	TCP 32768-65535	*<SSH Proxy IP Address>*/32	許可	プライベートサブネットからパブリックサブネット内の SSH プロキシに対するアウトバウンド SSH トラフィックを許可します

ルール	タイプ	ポート範囲	宛先	アクション	説明
12	カスタム TCP ルール	TCP 32768-65535	*<SSH Proxy IP Address>*/32	許可	パブリックサブネットから Identity Console にアクセスするためのアウトバウンドトラフィックを許可します
15	HTTPS	TCP 443	0.0.0.0/0	許可	プライベートサブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します
20	HTTP	TCP 80	0.0.0.0/0	許可	プライベートサブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します
25	HTTPS	監査サーバがリスンしているポート番号	*<IP Address of the audit server>*	許可	eDirectory イベントの監査を許可します。 **注 :** このルールは、監査サーバがプライベートサブネットの外部にある場合にのみ適用されます。
*	All Traffic	すべて (All)	0.0.0.0/0	拒否	前のルールによってまだ処理されていないすべてのアウトバウンド IPv4 トラフィックを拒否します (変更不可能)

セキュリティグループの設定

セキュリティグループは、VPC 内の 1 つまたは複数のインスタンスに割り当てることができる仮想ファイアウォールルールのセットです。

デフォルトでは、新しいセキュリティグループではポート 22 の着信トラフィックのみが許可されるので、インスタンスには SSH を使用してのみ接続できます。

詳細については、「Amazon EC2 Security Groups for Linux Instances」を参照してください。

AWS に OpenText eDirectory をデプロイするには、3 つのセキュリティグループを作成します。たとえば、Security Group 1、Security Group 2、Security Group 3 のようにします。次のポートルールを使用して、これらのセキュリティグループを作成します。

1. **Security Group 1 (eDirectory 用):**

ポート	ソース	説明
TCP 22	パブリックサブネット	パブリックサブネットからの SSH トラフィックを許可します
TCP 636	プライベートサブネット	プライベートサブネット内で LDAPS トラフィックを許可します
TCP 524	プライベートサブネット	プライベートサブネット内で eDirectory の NCP トラフィックを許可します
UDP 427	プライベートサブネット	プライベートサブネット内の SLP トラフィックを許可します

注 : OpenText eDirectory に割り当てられているセキュリティグループでは、LDAP ポート 389 を有効にしないでください。HTTP ポートは、ツリー CA をホストする OpenText eDirectory サーバに割り当てられているセキュリティグループでのみ有効にする必要があります。

2. **Security Group 2 (eDirectory 用):**

ポート	ソース	説明
TCP 8028	VPC	ツリー CA によって発行された証明書を使用して設定されている VPC にサービスが存在する場合に、VPC から OpenText eDirectory ツリーの CRL にアクセスするために必要です。このセキュリティグループは、ツリー CA をホストしている OpenText eDirectory サーバに割り当てられます。

3. **Security Group 3 (Identity Console 用):**

ポート	ソース	説明
TCP 22	パブリックサブネット	パブリックサブネットからの SSH トラフィックを許可します
TCP 9000	パブリックサブネット	パブリックサブネットから Identity Console にアクセスするための HTTPS トラフィックを許可します

SSH キーペアを作成する

Amazon EC2 インスタンスを設定する前に、SSH キーペアを作成する必要があります。キーペアを作成するには、次の手順を実行します。

1 次のコマンドを使用して、クライアント上で 4096 ビットの RSA SSH キーペアを作成します。

```
ssh-keygen -t rsa -b 4096
```

ssh-keygen により、新しく作成された公開鍵が ~/.ssh/id_rsa.pub に置かれます。

2 ご使用の Amazon EC2 アカウントに SSH 公開鍵をインポートします。詳細については、「独自の公開鍵を Amazon EC2 にインポートする (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/create-key-pairs.html)」を参照してください。

重要 : SSH 秘密鍵を使用してのみ、インスタンスに接続して管理できます。そのため、SSH 秘密鍵を紛失しないようにしてください。

インスタンスの作成とデプロイ

サポートされているプラットフォームのいずれかに EC2 インスタンスを作成して起動します。インスタンスを作成して起動する方法の詳細については、「EC2 リソースを作成し EC2 インスタンスを起動する」を参照してください。インスタンスを作成および起動する際には、次の手順も実行する必要があります。

1 最初の OpenText eDirectory サーバを設定するインスタンスに Security Group 1 を関連付け、その他のすべての OpenText eDirectory サーバを設定するインスタンスに Security Group 2 を関連付け、Identity Console を設定するインスタンスに Security Group 3 を関連付けます。セキュリティグループの詳細については、103 ページの 「仮想マシン用のネットワークセキュリティグループの設定」を参照してください。

2 106 ページの 「SSH キーペアを作成する」セクションで作成した公開鍵をインスタンスに関連付けます。

OpenText eDirectory データの保存用に EBS ボリュームを設定する

EC2 インスタンスのクラッシュが発生した場合に OpenText eDirectory のデータと設定が失われるのを防ぐには、EBS ボリュームを設定する必要があります。OpenText eDirectory のデータと設定を復元する方法の詳細については、144 ページの 「障害復旧」を参照してください。EC2 インスタンスを作成した後、次の手順を実行して、OpenText eDirectory デプロイできるようにインスタンスを準備します。

1. EBS ボリュームを作成するために、「Amazon EBS ボリュームの作成」にある手順を実行します。
2. EC2 インスタンスに EBS ボリュームを関連付けます。詳細については、「インスタンスへの Amazon EBS ボリュームのアタッチ」を参照してください
3. インスタンスにログインし、EBS ボリュームを ext4 ファイルシステムでフォーマットし、EBS ボリュームをマウントします。EBS ボリュームをフォーマットしてマウントする方法の詳細については、「Linux で Amazon EBS ボリュームを使用できるようにする」を参照してください。
4. ディレクトリを EBS ボリュームから OpenText eDirectory データおよび NICI データディレクトリにバインドマウントします。ルートユーザとして次の手順を実行して、バインドマウントします。

   4a 次のコマンドを使用して、OpenText eDirectory データディレクトリを作成します。

   ```
   mkdir <mount_point>/eDirectory_data
   ```

   4b 次のコマンドを使用して、NICI データディレクトリを作成します。

   ```
   mkdir <mount_point>/nici_data
   ```

   4c 次のコマンドを使用して、NICI および OpenText eDirectory の環境設定ディレクトリを作成します。

   ```
   mkdir <mount_point>/eDirectory_nici_conf
   ```

   4d 次のコマンドを使用して、OpenText eDirectory に必要なディレクトリを作成します。

   ```
   mkdir --parents /var/opt/novell/eDirectory
   mkdir --parents /var/opt/novell/nici
   mkdir --parents /etc/opt/novell/eDirectory
   ```

   4e ディレクトリをバインドマウントするために、次の内容を /etc/fstab に追加します。

   ```
   <mount_point>/eDirectory_data /var/opt/novell/eDirectory none
   defaults,bind 0 0

   <mount_point>/nici_data /var/opt/novell/nici none defaults,bind 0 0

   <mount_point>/eDirectory_nici_conf /etc/opt/novell/eDirectory none
   defaults,bind 0 0
   ```

注 : インスタンス内でのすべての操作は、ルートユーザとして実行する必要があります。

OpenText eDirectory および OpenText Identity Console のインストール

前提条件

- ❒「システム要件」に記載されている要件を満たしていることを確認します。
- ❒ 103 ページの 「仮想マシン用のネットワークセキュリティグループの設定」で説明されているように、セキュリティグループを作成します。
- ❒ SSH プロキシインスタンスは、強化され、セキュリティ保護されたサーバである必要があります。このインスタンスについては SSH ポート 22 のみを開き、適切なパフォーマンスとメモリを持つ AWS インスタンスタイプを選択します。プライベートサブネット内のインスタンスと SSH プロキシが実行されているインスタンスへのアクセスに必要な SSH 秘密鍵は、VPC 内には保存しないようにしてください。クライアント上にのみ保存してください。SSH プロキシサーバには Elastic IP アドレスを関連付けて、スタティックな IP アドレスを設定します。
- ❒ SSH プロキシインスタンス内に VNC サーバを設定します。VNC サーバは、強力なパスワードを使用して強化する必要があります。セキュリティ保護された通信のみを可能にするために、SSH トンネルを使用して VNC サーバに接続します。VNC サーバは、localhost からの接続についてのみリスンするように設定する必要があります。セッションのロックアウトを防止するために、画面ロックを無効にします。VNC サーバを使用した後、セッションを終了する必要があります。
- ❒ OpenText eDirectory/OpenText Identity Console を設定するプライベートサブネット内のインスタンスには、SSH プロキシを使用して接続します。

  ```
  ssh -i edir_key.pem -A -J ec2-user@<ssh_proxy_ip> ec2-
  user@<instance_private_ip>
  ```

  注

  - 上記のサンプルコマンドで、edir_key.pem は、サーバキーを含むサンプルファイル名です。
  - SSH-Add コマンドを使用してエージェント内に識別情報ファイルを追加すれば、ログインするたびに識別情報ファイルを使用しないで済みます。

  インスタンスのプライベート IP アドレスを表示するには、[**インスタンス**] > [*インスタンス*] > [**説明**] をクリックします。
- ❒ バックエンドサブネット内の VM に SLP ディレクトリエージェント (DA) サーバを設定します。SLP DA がデプロイされている VM に対して、NSG のインバウンドルールでポート 427 を開きます。slp.conf ファイルを編集して、DA 操作を有効にします。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*OpenSLP for OpenText eDirectory の設定*」を参照してください。

インストールと環境設定の手順

このセクションでは、AWS EC2 環境に OpenText eDirectory と OpenText Identity Console をインストールして設定する手順について説明します。OpenText eDirectory をインストールした後、次の条件を満たしていることを確認する必要があります。

- EBA が有効になっている
- SNMP が無効になっている
- OpenText eDirectory がポート 389 でリスンしていない
- LDAP および HTTP サービスは、ECDSA 証明書のみを使用するように設定されている
- 使用されていない場合は、AWS EC2 プライベートインスタンスの SSH ポートへのアクセスを無効にする必要があります。
- OpenText iMonitor、eMBox、および DHost モジュールを無効にして、セキュリティを強化する。これらのモジュールを無効にした後、これらのモジュールに関係するすべてのアクティビティは、NDS ユーティリティのみを使用して実行する必要があります。

OpenText eDirectory のインストールと設定

1. eDirectory を設定するプライベートサブネット内のインスタンスに、セキュアコピー (scp) を使用して OpenText eDirectory_*<version>*_Linux_x86_64.tar.gz ファイルをコピーします (SSH プロキシを使用します)。

   ```
   scp -i <keyname> -o ProxyJump=ec2-user@<ssh_proxy_ip>
   eDirectory_<version>_Linux_x86_64.tar.gz ec2-user@<instance_ip>:/
   <directory>
   ```

2. OpenText eDirectory をインストールします。詳細については、「nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする」を参照してください。
3. OpenText eDirectory を設定します。詳細については、「ndsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する」を参照してください。たとえば、OpenText eDirectory をインストールして設定するためのコマンドのサンプルを次に示します。

   ```
   ndsconfig  new [-t <tree_name>] [-n <server context>] -a <admin FDN> [-
   w <admin password>] -P ldaps://<instance_ip> --configure-eba-now yes
   ```

4. openslp-server をインストールして、SLPD サービスを始動します。

OpenText Identity Console のインストールと設定

OpenText Identity Console の管理コンソールを使用すると、AWS 環境での OpenText eDirectory の操作を管理できます。OpenText eDirectory をインストールした後、AWS インスタンスに OpenText Identity Console をインストールする必要があります。

1. OpenText Identity Console を設定するプライベートサブネット内のインスタンスに、セキュアコピー (scp) を使用して IdentityConsole_*<version>*_linux_x86_64.tgz ファイルをコピーします (SSH プロキシを使用します)。

```
scp -i <keyname> -o ProxyJump=ec2-user@<ssh_proxy_ip>
IdentityConsole_<version>_linux_x86_64.tgz ec2-user@<instance_ip>:/
<directory>
```

2 OpenText Identity Console をインストールして設定します。詳細については、「Installing Standalone OpenText Identity Console (Non-Docker)」を参照してください。OpenText Identity Console をインストールする前に、『OpenText™ Identity Console インストールガイド』の「System Requirements and Prerequisites for Standalone Installation (Non-Docker)」セクションにあるシステム要件を参照してください。

3 Identity Console を実行している VM の自己署名証明書を、セキュア CA 署名証明書で置き換えます。詳細については、「iManager 用の一時的な自己署名証明書の置き換え」を参照してください。

注 : OpenText Identity Console サーバは、ECDSA 証明書のみを使用するように設定してください。OpenText Identity Console をインストールした後、権限を持ったユーザと、このユーザが管理する適切な OpenText eDirectory ツリー名を指定します。

OpenText Identity Console の起動

OpenText Identity Console を起動するには、次の手順を実行します。

1 SSH トンネルを使用して、SSH プロキシの localhost 上で実行されている VNC サーバに接続します。

2 同じインスタンスにブラウザをインストールして起動します。

3 OpenText Identity Console を起動し、IP アドレスまたはツリー名を使用して OpenText eDirectory ツリーを接続します。

設定後のタスク

1 EBA が有効になっているかどうかを確認するため、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*EBA に関する情報の表示*」を参照してください。

2 証明書サーバで Suite B を有効にします。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*Certificate Server 上での Suite B の有効化*」を参照してください。

3 1 番目の OpenText eDirectory サーバに対して AES 256 ビットツリーキーを設定します。詳細については、『*NICI Administration Guide (NICI 管理ガイド)*』の「Creating an AES 256-Bit Tree Key (AES 256 ビットツリーキーを作成する)」を参照してください。

4 1 番目の OpenText eDirectory サーバにある CRL 配布ポイントを削除します。ポート 389 でのセキュリティ保護されていない LDAP アクセスはすべての OpenText eDirectory サーバで無効化されているため、ツリー CA の CRL は HTTP のみでダウンロードできるようにする必要があります。次の手順を実行して、CRL 配布ポイントを削除します。

   4a OpenText Identity Console に管理者としてログインします。

   4b [**証明書管理**] > [**CA 管理**] > [**CRL**] > [CRL の作成] に移動します。

4c ［One］をクリックします。HTTP CRL 配布ポイント (http://<instance_ip>:8028/crl/one.crl) 以外のすべての［CRL 配布ポイント］を選択して削除します。

4d ［保存］をクリックし、［OK］をクリックします。

4e ［OneEC］をクリックします。HTTP CRL 配布ポイント (http://<instance_ip>:8028/crl/oneec.crl) 以外のすべての［CRL 配布ポイント］を選択して削除します。

4f ［保存］をクリックし、［OK］をクリックします。

5 OpenText Identity Console 証明書マネージャを使用して、サーバのデフォルト証明書を修復します。デフォルト証明書を修復するには、次の手順を実行します。

5a OpenText Identity Console に管理者としてログインします。

5b ［証明書管理］>［デフォルト証明書］に移動します。

5c 証明書を所有するサーバを選択して、［次へ］をクリックします。

5d ［デフォルト証明書はすべて上書きされます］を選択し、［次へ］をクリックします。

5e 実行するタスクを確認し、［完了］を選択し、［OK］をクリックします。

6 ECDSA 証明書と Suite B 暗号化を使用するように LDAP サービスと HTTP サービスを設定します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*ECDSA 証明書と Suite·B·Cipher を使用するための LDAP サービスと HTTP サービスの設定*」を参照してください。完了したら、OpenText eDirectory を再起動します。

7 SNMP サブエージェントがアンロードされているかどうかを確認する方法について、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*SNMP サーバモジュールのロードとアンロード*」を参照します。

8 OpenText eDirectory がポート 389 でリスンしていないことを確認します。

9 OpenText iMonitor、eMBox、DHost、および HTTP スタックを無効にします。

9a 次の手順を実行して、ツリー CA をホストしている OpenText eDirectory サーバの OpenText iMonitor、eMBox、および DHost を無効にします。

9a1 ndsmodules.conf ファイルを編集して、hconserv、imon、および embox をコメント化します。

9a2 OpenText eDirectory を再起動します。

9b 次の手順を実行して、eDirectory レプリカサーバの HTTP スタックを無効にします。

9b1 ndsmodules.conf ファイルを編集して、httpstk、hconserv、imon、および embox をコメント化します。

9b2 OpenText eDirectory を再起動します。

注 : httpstk は、コメント化の前に、ndsmodules.conf ファイル内で nds より上に配置されている必要があります。これにより、nds モジュールは HTTP スタックを有効にしなくなります。

---

10 アドバタイジング方式としてユニキャストを使用することを OpenText eDirectory に強制するように、SLP を設定します。バックエンドサブネット内の DA サーバの IP アドレスを提供するように、slp.conf ファイルを編集します。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*環境設定パラメータ*」を参照してください。

注 : すべての OpenText eDirectory インスタンスと OpenText Identity Console の設定が完了したら、SSH ポートへのアクセスを拒否し、必要な場合にのみ許可するように、AWS プライベートサブネットのネットワーク ACL を設定してください。

監査サービスのデプロイ

AWS EC2 に Common Event Format (CEF) 監査サービスをデプロイして、さまざまな OpenText eDirectory イベントを監査することができます。CEF 監査サービスをデプロイするには、次の手順を実行します。

1 VPC に監査サーバをインストールします。

2 ポート上でリスンするように監査サーバを設定します

注 : 監査サーバとしては Sentinel を使用することをお勧めします。

3 次の設定を使用して新しいセキュリティグループを作成し、監査サーバが実行されているインスタンスに関連付けます。

ポート	ソース	説明
TCP (監査サーバポート)	プライベートサブネット	OpenText eDirectory サーバからのイベントの受信を許可します

4 すべての OpenText eDirectory インスタンスで、次のように /etc/opt/novell/eDirectory/conf/auditlogconfig.properties ファイルを更新します。

```
log4j.appender.S.Host=<Auditing server ip>
log4j.appender.S.Port=<auditing server port>
```

5 OpenText Identity Console から、対応する CEF イベントを有効にします。詳細については、「監査する CEF イベントの設定 (https://www.netiq.com/documentation/edirectory-93/edir_admin/data/t44e7j6b8ufi.html#t44e7j6cx14i)」を参照してください。有効にされたイベントは、監査サーバに転送されます。

障害復旧

障害復旧は、OpenText eDirectory が実行されているインスタンスでクラッシュが発生した場合に実行します。障害復旧を行うには、次の手順を実行します。

1 クラッシュしたインスタンスを停止し、EBS ボリュームとの関連付けを解除します。詳細については、「インスタンスからの Amazon EBS ボリュームのデタッチ」を参照してください。

2 クラッシュしたインスタンスと同じオペレーティングシステムを使用して、新しい EC2 インスタンスを設定します。

3 新しい EC2 インスタンスに同じバージョンの OpenText eDirectory をインストールします。

4 新しいインスタンスに EBS ボリュームをアタッチし、ファイルシステムをマウントします。詳細については、「インスタンスへの Amazon EBS ボリュームのアタッチ」を参照してください。

5 ディレクトリをバインドマウントします。

ディレクトリをバインドマウントするには、/etc/fstab を次のように更新します。

```
<mount_point>/eDirectory_data /var/opt/novell/eDirectory none
defaults,bind 0 0

<mount_point>/nici_data /var/opt/novell/nici none defaults,bind 0 0

<mount_point>/eDirectory_nici_conf /etc/opt/novell/eDirectory none
defaults,bind 0 0
```

6 /etc/opt/novell/eDirectory/conf/nds.conf 内の IP アドレスを、現在のインスタンスの IP アドレスに変更します。

7 OpenText eDirectory をアップグレードし、ヘルスチェックをスキップします。詳細については、「*OpenText eDirectory をアップグレードする*」を参照してください。

8 ndsrepair ユーティリティを使用して、ネットワークアドレスを修復します。詳細については、『*OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド*』の「DSRepair オプション」を参照してください。

9 ツリー CA の IP アドレスが変更された場合は、CRL 配布ポイントの IP アドレスを変更します。IP アドレスの変更方法の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド (https://www.netiq.com/documentation/edirectory-93/edir_admin/data/b1j4tpo3.html#b1j581ua)』の「CRL 設定オブジェクトのプロパティの表示と変更」を参照してください。

10 OpenText Identity Console 証明書サーバプラグインを使用して、サーバのデフォルト証明書を修復します。デフォルト証明書を修復するには、次の手順を実行します。

10a OpenText Identity Console に管理者としてログインします。

10b [**証明書管理**] > [**デフォルト証明書**] に移動します。

10c 証明書を所有するサーバを選択して、[**次へ**] をクリックします。

10d [**デフォルト証明書はすべて上書きされます**] を選択し、[**次へ**] をクリックします。

10e 実行するタスクを確認し、[**完了**] を選択し、[**OK**] をクリックします。

11 新しい ECDSA 証明書を使用するように LDAP および HTTP サービスを設定します。

AWS での OpenText eDirectory デプロイメントのサイジングガイダンス

このガイドでは、AWS VPC での OpenText eDirectory デプロイメントに最適な AWS EC2 インスタンスと EBS ボリュームを推奨します。サイズ要件の見積もりを支援するために、最も有用な見積もり要因とさまざまなパフォーマンスシナリオを提供しました。これらのシナリオは、特定の環境設定パラメータを設定した制御されたラボ環境でテストされています。

- 146 ページの 「AWS EC2 インスタンスの選択を開始する」
- 147 ページの 「パフォーマンステスト環境の詳細」
- 149 ページの 「サイジング要件の決定」

AWS EC2 インスタンスの選択を開始する

AWS に OpenText eDirectory をデプロイする前に、OpenText eDirectory サーバから最適なパフォーマンスを得る EC2 インスタンスおよび EBS ボリュームのタイプを決定する必要があります。OpenText eDirectory のパフォーマンスは、企業が eDirectory をどのように使用することを計画しているかに関する複数のリソースに応じて異なります。見積もりの際に考慮する必要があるリソースは次のとおりです。

- **AWS EC2 インスタンスカテゴリ**: 環境に必要な適切なインスタンスカテゴリを選択する必要があります。当社では、次のインスタンスカテゴリをさまざまなパフォーマンスシナリオについてラボ環境で検討しました。
  - 一般用途
  - メモリ最適化
  - コンピューティング最適化

  また、EC2 インスタンスと接続された EBS ボリュームの間で専用の帯域幅を活用するには、OpenText eDirectory のデプロイに EBS 最適化 EC2 インスタンスを使用することをお勧めします。その場合、EBS ボリュームにプロビジョニングされる IOPS を効果的に使用することもできます。一部のインスタンスでは、EBS 最適化はデフォルトでは有効になっていません。そのようなシナリオでは、手動で有効にする必要があります。
- **AWS EC2 インスタンスタイプ**: 環境に適したインスタンスカテゴリを選択した後、ビジネスニーズまたは予想される負荷に応じてインスタンスタイプを選択する必要があります。AWS では、各インスタンスカテゴリの下でさまざまな EC2 インスタンスタイプを選択できます。各カテゴリには、プロセッサのタイプやネットワークのスループットなど、さまざまなパラメータに応じてサブカテゴリがあります。CPU コアの数、RAM のボリューム、ネットワーク容量、およびコストは、選択したインスタンスタイプによって異なります。インスタンスタイプの詳細については、「Amazon EC2 インスタンス」を参照してください。
- **AWS EBS ボリュームタイプ**: EC2 インスタンスのルートボリュームまたはローカルストレージは永続的ではないため、永続化する必要がある OpenText eDirectory DIB やその他のサーバファイルは、EC2 インスタンスに AWS Elastic Block Storage (EBS) ボリュームを

アタッチして保存する必要があります。EBS ボリュームを使用すると、保存されたデータは AWS EC2 インスタンスが終了した後でも永続化されるため、データ回復のシナリオで役立ちます。EBS ボリュームに保存されているデータは、デフォルトで同じ可用性ゾーン内で複製されます。詳細については、「EBS ボリュームタイプ」を参照してください。

OpenText eDirectory は読み込み集中型のディレクトリサービスなので、EBS ボリュームタイプの選択はディスク I/O の高さに依存します。このため、HDD ボリュームタイプよりも SSD ボリュームタイプを使用することをお勧めします。SSD ボリュームタイプは小さいディスク I/O またはランダムなディスク I/O を効率的に処理するからです。これは基本的に、ディスク I/O のストリーミングに使用されます。

プロビジョニングされた IOPS SSD ボリュームタイプ (io1 および io2) では、パフォーマンスに一貫性がない汎用 SSD ボリュームタイプ (gp2) と比較して、ボリュームタイプの作成時に一貫した IOPS レートを指定できます。また、汎用 SSD は、IOPS の設定を、プロビジョニングされた IOPS SSD のボリュームタイプには存在しないディスクサイズと関連付けます。

パフォーマンステスト環境の詳細

このセクションでは、制御されたラボ環境で使用されているさまざまな環境設定パラメータ、インスタンスカテゴリ、およびタイプについて説明します。

EC2 インスタンスタイプ :

パフォーマンステストでは、次のインスタンスタイプが使用されています。

***表 6-3** インスタンスタイプ 4xLarge*

カテゴリ	型	CPU	メモリ	EBS 最適化	EBS 帯域幅
コンピューティング最適化	c5a.4xlarge	16 CPU、3.3GHz	32GB	対応	3,170Mbps

***表 6-4** インスタンスタイプ 2xLarge*

カテゴリ	型	CPU	メモリ	EBS 最適化	EBS 帯域幅
一般用途	m5a.2xlarge	8 CPU、2.5GHz	32GB	対応	2,880Mbps
メモリ最適化	r5a.2xlarge	8 CPU、2.5GHz	64GB	対応	2,880Mbps
コンピューティング最適化	c5a.2xlarge	8 CPU、3.3GHz	16GB	対応	3,170Mbps

表 6-5 インスタンスタイプ xLarge

カテゴリ	型	CPU	メモリ	EBS 最適化	EBS 帯域幅
一般用途	m5a.xlarge	4 CPU、2.5GHz	16GB	対応	2,880Mbps
メモリ最適化	r5a.xlarge	4 CPU、2.5GHz	64GB	対応	2,880Mbps
コンピューティング最適化	c5a.xlarge	4 CPU、3.3GHz	8GB	対応	3,170Mbps

共通パラメータ

表 6-6 共通パラメータの説明

パラメータ	説明
AWS VPC の詳細	1 つのパブリックサブネットと 1 つのプライベートサブネット。クライアントと OpenText eDirectory の両方が VPC プライベートサブネット内にあります。
OpenText eDirectory EC2 インスタンスの詳細	◆ **オペレーティングシステム** : SUSE Linux Enterprise Server 15 SP2 (HVM)、SSD ボリュームタイプ ◆ **OpenText eDirectory ビルド** : OpenText eDirectory 9.2 ◆ **Elastic Block Storage**: プロビジョニングされた IOPS SSD io2、100GB、OPS – 1000 ◆ **Dib**: 3GB、200 万人以上のユーザ
操作	◆ **操作** : LDAPS (636) 検索および変更操作 ◆ **パーセンテージ** : 80% の読み込み操作と 20% の書き込み操作。 ◆ **検索スコープ** : サブツリー検索 ◆ **検索ベース** : ou=users,o=novell ◆ **検索フィルタ** : 検索ベースのランダムユーザ ◆ **属性** : sn;title
OpenText eDirectory のチューニング	**OpenText eDirectory Flaim DB 設定 :** preallocatecache=true cache=4000000000cpinterval=360

パラメータ	説明
LDAP クライアント EC2 インスタンス	◆ **オペレーティングシステム** : RHEL8.0 ◆ **LDAP クライアントツール** : JMeter 5.1 ◆ **EC2 インスタンスタイプ** : m5d.4xlarge、16 CPU、64GB RAM ◆ **TCP ポートの範囲** : 1024-65534

サイジング要件の決定

このセクションでは、一般的に使用されるシナリオのパフォーマンスデータに基づいた推奨事項について説明します。このデータは、ご使用の環境に最適な AWS EC2 インスタンスを決定するのに役立ちます。OpenText eDirectory デプロイに適したサイズ見積もりを開始する前に、デプロイに最適なインスタンスカテゴリを特定する必要があります。

シナリオ 1 - AWS EC2 インスタンスカテゴリの識別

このパフォーマンスシナリオでは、OpenText eDirectory のデプロイと操作に適した AWS EC2 インスタンスカテゴリを決定します。

次のグラフは、スレッド数に対する 1 秒あたりの平均応答時間 (ミリ秒) と操作数の測定値を示しています。測定値は、2xlarge インスタンスタイプに属する、汎用、メモリ最適化、およびコンピューティング最適化の 3 つのインスタンスカテゴリから取得したものです。

図 6-2 *m5a と r5a と c5a の場合の同時に実行されるスレッド数に基づく平均応答時間の変化*

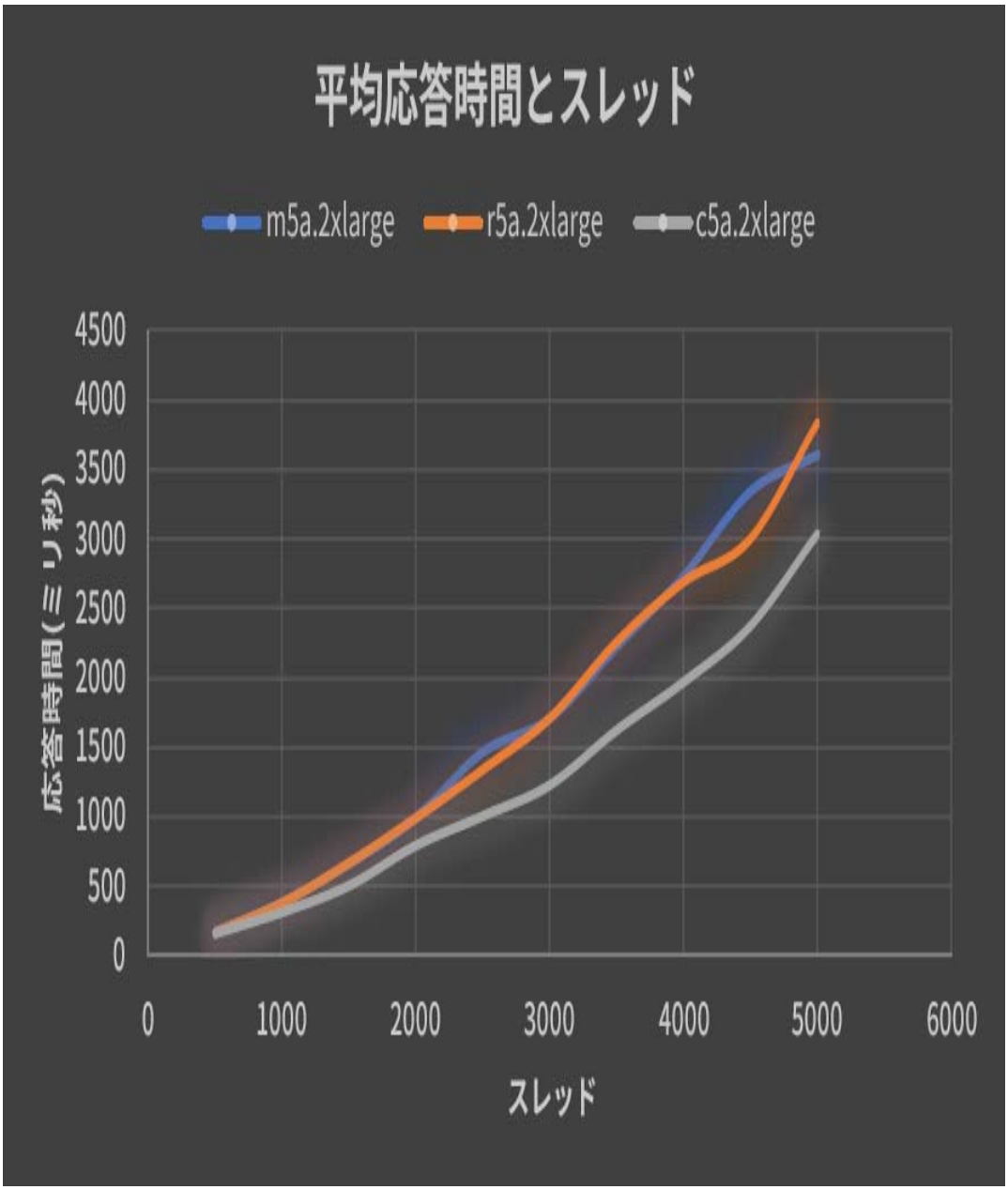
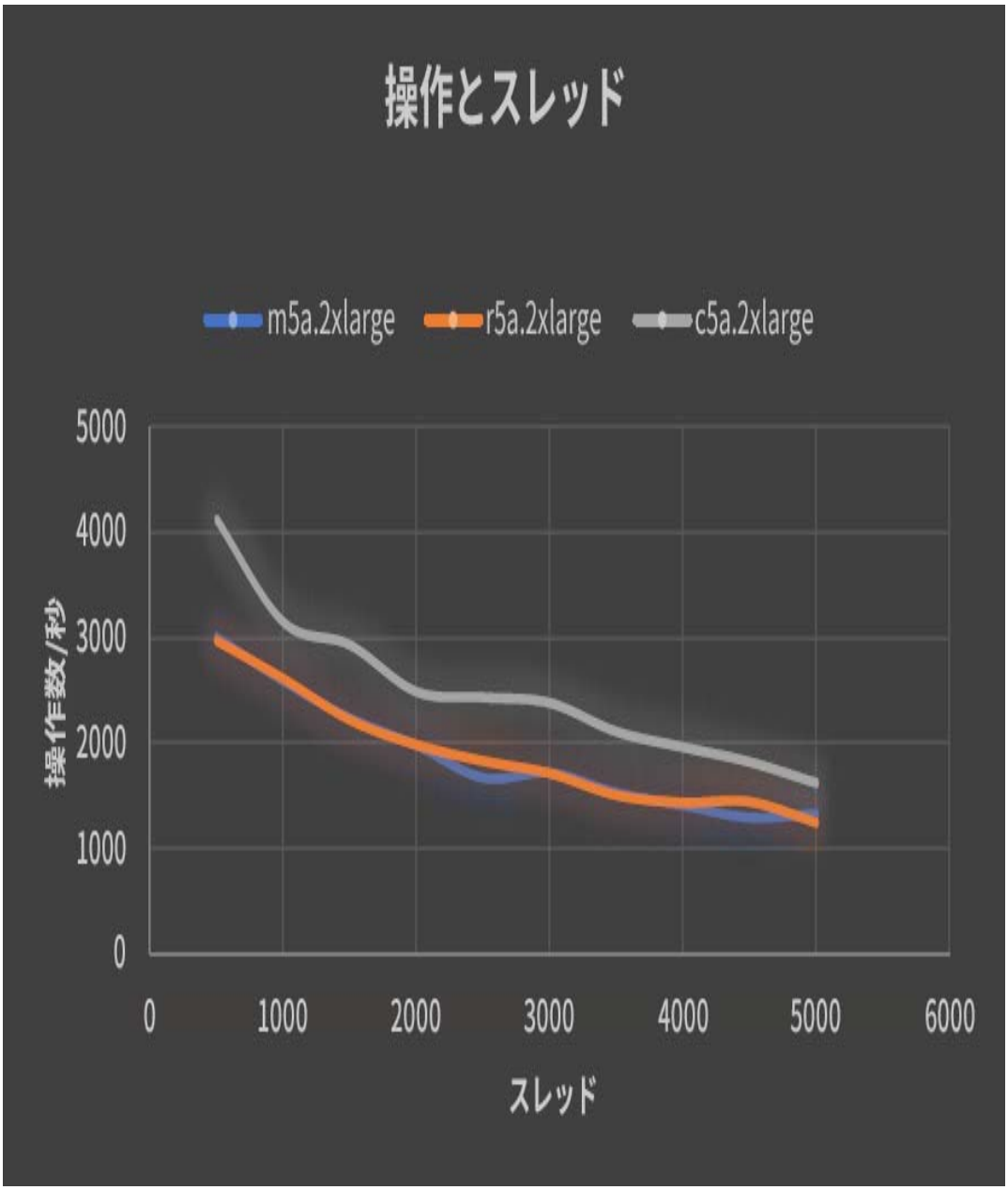


図 6-3 スレッド数に基づく同時に実行される操作数の変化



次のグラフは、スレッド数に対する 1 秒あたりの平均応答時間 (ミリ秒) と操作数の測定値を示しています。測定値は、xlarge タイプに属する 3 つのインスタンスカテゴリ (一般用途、メモリ最適化、およびコンピューティング最適化) から取得したものです。

図 6-4 *m5a と r5a と c5a の場合の同時に実行されるスレッド数に基づく平均応答時間の変化*

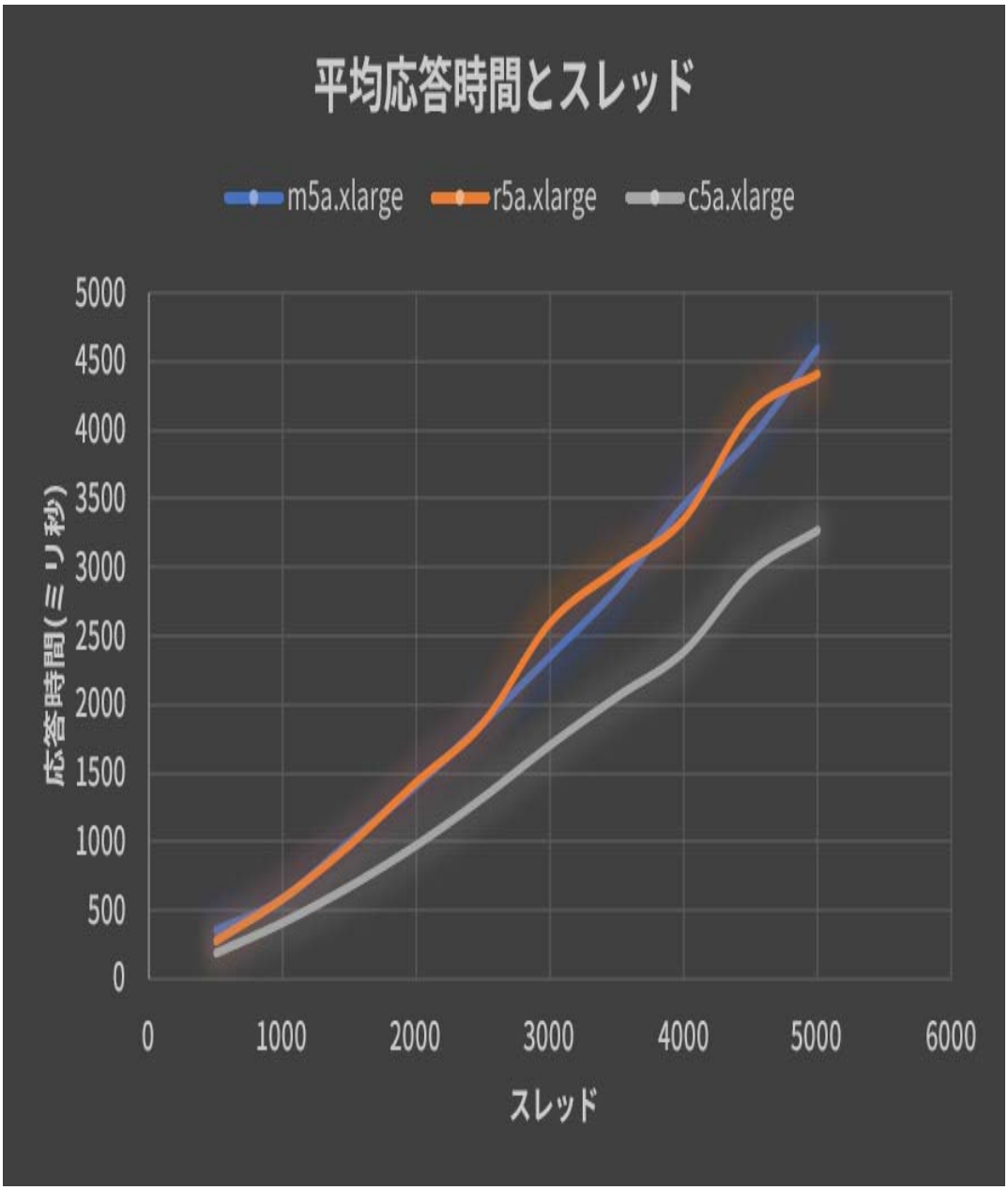
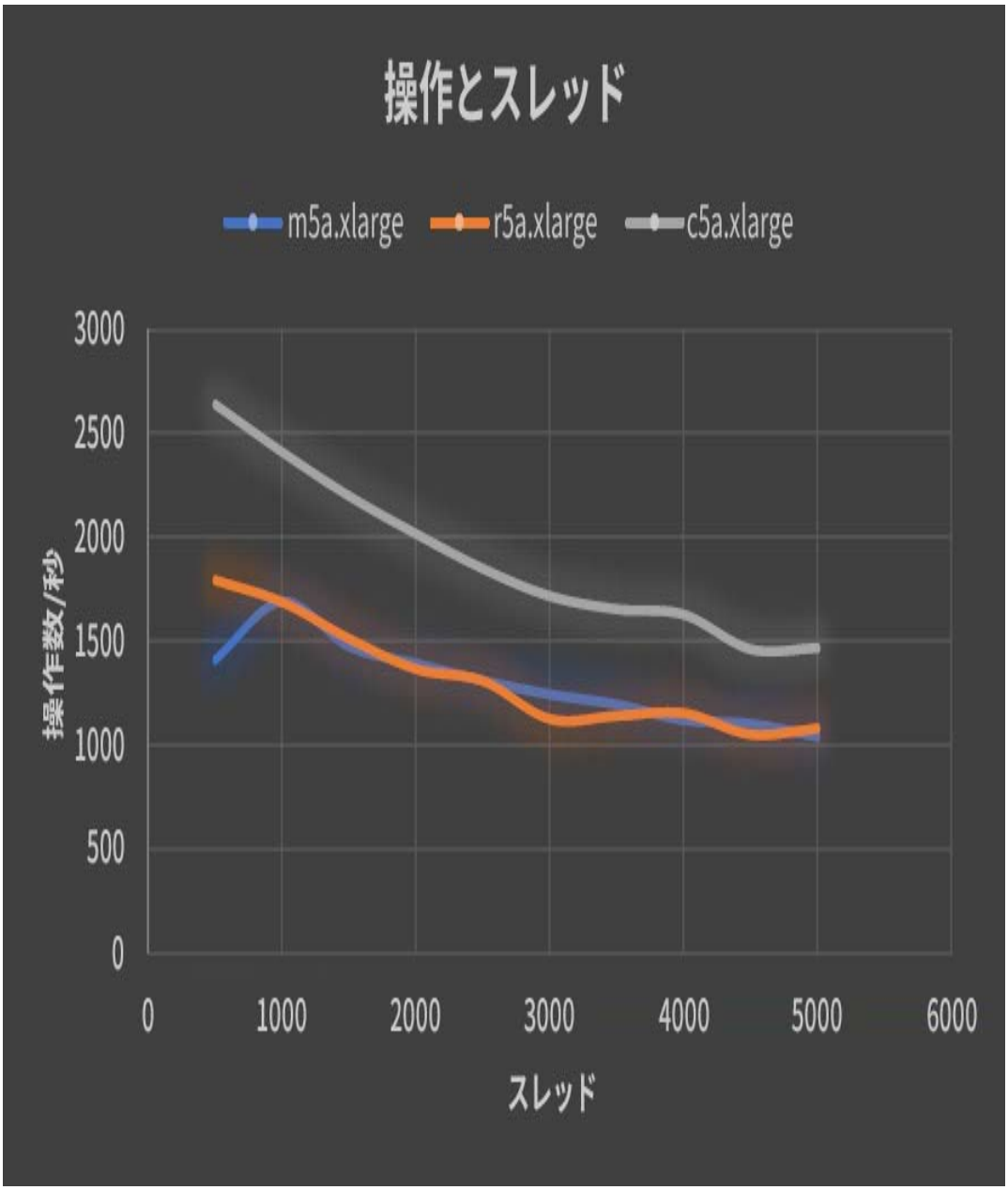


図 6-5 スレッド数に基づく同時に実行される操作数の変化



推奨

上記のテストシナリオによると、AWS 環境にデプロイする場合に OpenText eDirectory のパフォーマンスを最大限に高めるには、コンピューティング最適化インスタンスを EBS ボリュームと共に使用する必要があります。インスタンスタイプは、最初は、想定される負荷に基づいて選択する必要があります。それがリソース要件を満たしていない場合は、いつでもインスタンスタイプのサイズを変更できます。また、IO とスループットのパフォーマンスを最適化するには、プロビジョニングされた IOPS SSD ボリュームを EC2 インスタンスにアタッチする必要があります。

シナリオ 2 - AWS EC2 サイジング要件の識別

前のパフォーマンスシナリオでは、コンピューティング最適化が、OpenText eDirectory のデプロイと操作に適した AWS EC2 インスタンスカテゴリであると判定しました。

このパフォーマンステストシナリオでは、最適なサイジングガイドを提供するために、3 種類のコンピューティング最適化インスタンスタイプ (xlarge、2xlarge、および 4xlarge) からデータを収集しました。

次のグラフは、スレッド数に対する 1 秒あたりの平均応答時間 (ミリ秒) と操作数の測定値を示しています。このテストでは、500 ミリ秒を、許容される最大平均応答時間と見なしました。

図 6-6 *xlarge と 2xlarge と 4xlarge の場合の同時に実行されるスレッド数に基づく平均応答時間の変化*

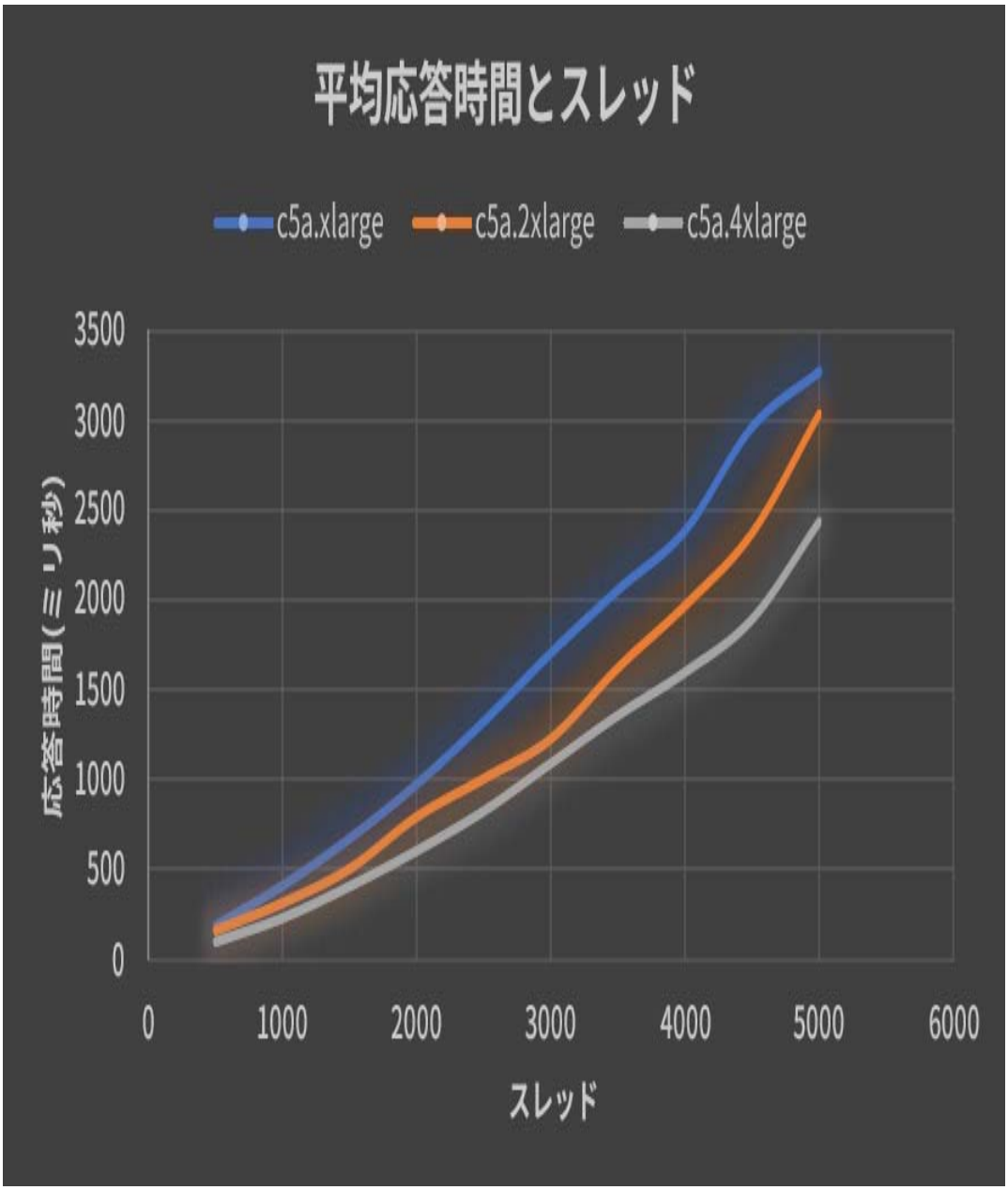
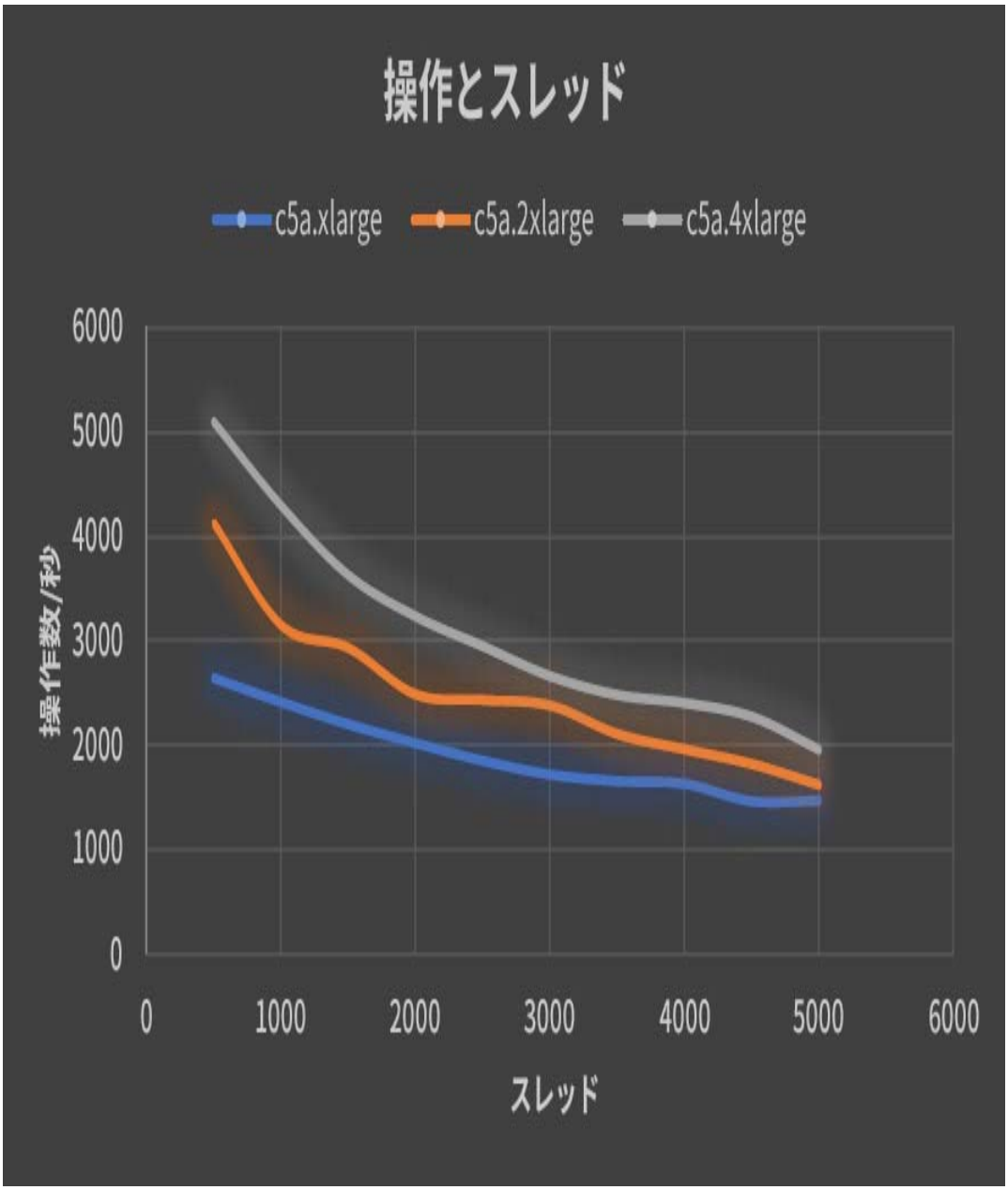


図 6-7 スレッド数に基づく同時に実行される操作数の変化



推奨

次の表は、示されている EC2 インスタンスごとに平均応答時間を 500 ミリ秒以内に維持するためのスレッドの最大数とそれに対応する 1 秒あたりの操作数の測定値です。

インスタンスタイプ	スレッド	1 秒あたりの操作数
c5a.xlarge	1200	2300
c5a.2xlarge	1500	2900
c5a.4xlarge	1750	3500

上で説明した、想定されるスレッドとスループットの値に基づいて、最適なコンピューティングインスタンスサイズを選択することができます。

7 Docker コンテナを使用した OpenText eDirectory のデプロイ

この章では、Docker コンテナを使用して OpenText eDirectory をデプロイする方法について説明します。

- 157 ページの 「Docker の利点」
- 157 ページの「Docker コンテナを使用して OpenText eDirectory をデプロイするための計画」
- 158 ページの 「OpenText eDirectory コンテナのデプロイ」
- 165 ページの 「デプロイメント後のタスク」
- 168 ページの 「OpenText eDirectory データストレージの管理」
- 169 ページの 「Docker コンテナを使用した OpenText eDirectory のアップグレード」

Docker の利点

Docker は、最も一般的なアプリケーションのコンテナ化技術です。コンテナを使用してアプリケーションを作成、デプロイ、および実行するために設計されたプラットフォームです。コンテナは、独自のオペレーティングシステムを搭載し、ライブラリやパッケージなどのその他すべての依存関係があるアプリケーションをカプセル化します。Docker コンテナを使用した OpenText eDirectory のデプロイには、次のような利点があります。

- **移植性の高さ**: コンテナで実行されているアプリケーションは、Docker でサポートされている任意のオペレーティングシステムおよびハードウェアプラットフォームに容易にデプロイできます。
- **デプロイは簡単。**コンテナを使用すると、オーケストレーションツールにより、アプリケーションをより迅速にデプロイ、アップグレード、さらにはスケールすることができます。
- **一貫性**: コンテナをデプロイする場所に関係なく、OpenText eDirectory の機能に影響はありません。

Docker とそのコンポーネントの詳細については、「Docker Overview」を参照してください。

Docker コンテナを使用して OpenText eDirectory をデプロイするための計画

このセクションでは、OpenText eDirectory Docker コンテナをデプロイするためのシステム要件と前提条件について説明します。

システム要件

プラットフォームの要件

- OpenText eDirectory Docker コンテナのデプロイには、Docker Community Edition バージョン 18.06 以上が必要です。
- 推奨される Docker ストレージドライバは overlay2 です。BTRFS は、Docker をインストールできるホストでサポートされているファイルシステムではありません。
- Linux カーネルバージョン 3.10 以降。
- openSUSE Leap 15.4 は、Docker Community Edition バージョン 20.10.9-ce またはそれに近いバージョンで効率的に動作します。

ハードウェア要件

- Docker ホストマシンには、最低でも 4GB の RAM と 30GB のハードディスク容量をプロビジョニングする必要があります。

注 : メモリ、CPU、およびハードディスクの要件は、デプロイメントのタイプとデプロイするコンテナの数によって異なります。将来のスケールアップの可能性に備えるため、常に、現在の要件よりも多いリソースをプロビジョニングするようにしてください。

前提条件

- Docker ホストマシンにはスタティック IP アドレスが設定されている必要があります。
- Docker がインストールされている必要があります。サポートされているプラットフォームの詳細については『Docker マニュアル』を参照してください。
- Docker デーモンが稼動している必要があります .
- OpenText eDirectory Docker イメージ tarball は、ソフトウェアライセンスとダウンロードポータルからダウンロードする必要があります。
- Docker でコンテナ管理を実行する必要があるユーザは、docker グループに追加する必要があります。

Docker CLI

Docker CLI で使用されるさまざまなコマンドの説明は、こちらを参照してください。

OpenText eDirectory コンテナのデプロイ

OpenText eDirectory Docker イメージの OS ベースイメージは openSUSE Leap 15.4 です。OpenText eDirectory イメージのアーカイブファイルを Docker ホストマシンにダウンロードする必要があります。アーカイブファイルをダウンロードした後、それを抽出し、次のコマンドを使用してイメージをローカル Docker レジストリにロードする必要があります。

```
tar -xvf eDirectory_93x_Container.tar.gz
docker load --input eDirectory_93x/eDirectory_93x.tar.gz
```

上記のコマンドでは、edirectory:9.3.x という名前の Docker イメージがロードされます。

OpenText eDirectory Docker コンテナは、Docker Run コマンドで ndsconfig ユーティリティのすべてのパラメータを受け入れます。ndsconfig ユーティリティの詳細については、56 ページの 「dsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する」を参照してください。

注 : Docker Run コマンドで ndsconfig ユーティリティの -w オプションを使用してパスワードを設定することはお勧めできません。このオプションを使用して設定したパスワードは、Docker inspect コマンドを使用してプレーンテキストで表示できます。管理者の資格情報を設定するには、管理者の FDN と管理者パスワードをプロンプトで指定する方法がより安全です。

OpenText eDirectory Docker コンテナでは、次の ndsconfig パラメータに対してデフォルト値が使用されます。したがって、Docker run コマンドではこれらのパラメータを設定しないでください。

- **設定ファイル :** /config/eDirectory/inst/conf/nds.conf
- **インスタンスの場所 :** /config/eDirectory/inst/data/data
- **DIB 保存先 :** /config/eDirectory/inst/data/data/dib

注 : インスタンスデータと環境設定をコンテナの /config フォルダに含めることが大切です。これは、OpenText eDirectory コンテナの永続的な保存とアップグレードの機能を有効にするためです。詳細については、168 ページの 「OpenText eDirectory データストレージの管理」を参照してください。

OpenText eDirectory コンテナのデフォルトのログファイルの場所は、/config/eDirectory/inst/data/log です。

OpenText eDirectory をデプロイする前に、次の推奨事項を考慮する必要があります。

- Docker コンテナには、デフォルトではリソースの制約がありません。そのため、すべてのコンテナには、ホストのカーネルによって提供されるすべての CPU リソースおよびメモリリソースへのアクセスが提供されます。コンテナで使用できるリソースの量に制限を設定することにより、実行中の 1 つのコンテナによってより多くのリソースが消費され、実行中の他のコンテナがリソース不足になることがないようにする必要があります。
  - Docker run コマンドの --memory フラグを使用して、Docker コンテナで、コンテナによって使用されるメモリに対してハード制限が適用されるようにする必要があります。
  - Docker run コマンドの --cpuset-cpus フラグを使用して、Docker コンテナで、実行中のコンテナによって使用される CPU の容量に制限が適用されるようにする必要があります。

- --pids-limit には 300 を設定して、任意の時点でコンテナ内で生成されるカーネルスレッドの数を制限する必要があります。これは、DoS 攻撃を防ぐためです。
- Docker run コマンドで --restart フラグを使用して、コンテナ再起動ポリシーを on-failure に設定し、再試行回数を 5 に設定する必要があります。ホストマシン上の Docker デーモンが再起動された場合は、コンテナを手動で再起動する必要があります。
- OpenText eDirectory コンテナは、必ず、コンテナの起動後にヘルスステータスが［良好］と表示されてから使用する必要があります。コンテナのヘルスステータスを確認するには、次のコマンドを実行します。

  ```
  docker ps --filter status="running"
  ```

- 通常、Docker コンテナでは Linux 機能のデフォルトリストが有効になっています。OpenText eDirectory コンテナで次の機能のみが有効になっていることを確認し、その他の機能をドロップする必要があります。
  - AUDIT_WRIT
  - CHOWN
  - DAC_OVERRIDE
  - SETGID
  - SETUID
  - NET_BIND_SERVICE
  - SYS_CHROOT (SLP サービスを有効にしている場合のみ)
  - SYS_PTRACE (Linux ptrace を活用するユーティリティを使用する場合のみ。gdb など)

  機能の追加とドロップの詳細については、「Runtime privilege and Linux capabilities (https://docs.docker.com/engine/containers/run/#runtime-privilege-and-linux-capabilities)」を参照してください。
- eDirectory コンテナは、常に root 以外のユーザ (nds) として起動されます。追加のセキュリティ対策として、デーモンでのユーザ名前空間の再マッピングを有効にして、コンテナ内からの特権昇格攻撃を防止します。ユーザ名前空間の再マッピングの詳細については、「ユーザ名前空間でコンテナを分離する」を参照してください。
- Docker run コマンドで、--stop-timeout オプションの値を 180 に設定する必要があります。これは、nds サービスが正常にシャットダウンするのに十分な時間を確保するために必要です。

注 : 以前のバージョンのスタンドアロン OpenText eDirectory を使用している場合は、OpenText eDirectory 9.3 Docker コンテナを使用して、セットアップを Docker 環境に移行することはできません。

OpenText eDirectory Docker コンテナでは、マルチホストの Docker 環境にデプロイするためのホストおよびオーバーレイネットワークドライバがサポートされています。

- 161 ページの 「ホストネットワークでの OpenText eDirectory コンテナのデプロイ」
- 163 ページの「ユーザ定義のオーバーレイネットワークでの OpenText eDirectory コンテナのデプロイ」

ホストネットワークでの OpenText eDirectory コンテナのデプロイ

OpenText eDirectory コンテナは、Linux 上でのみホストネットワークドライバを使用してハイブリッド環境にデプロイできます。Docker ネットワークの詳細については、「Configure Networking」を参照してください。

注 : ホストネットワークは Windows ではサポートされていません。

ハイブリッド環境は、同じツリーに OpenText eDirectory サーバのレガシデプロイメントおよびコンテナベースのデプロイメントの両方を組み合わせたものです。ハイブリッドネットワークを使用すると、レガシ OpenText eDirectory デプロイメントをすでにホストしている既存の運用環境に、OpenText eDirectory Docker コンテナをシームレスに導入できます。Docker ホストネットワーキングでは、ホストのネットワークスタックがレガシ eDirectory デプロイメントとコンテナ化された eDirectory デプロイメントの両方によって共有されているため、サービスポートを再使用することはできません。また、コンテナ化された OpenText eDirectory サーバは、クライアントとツリー内の他のサーバに対して、レガシ OpenText eDirectory サーバとして表示されます。

次の例は、OpenText eDirectory コンテナを使用して新しいツリーを作成する方法を示しています。

```
docker run -it --name eDir-container-1 --stop-timeout 180 --restart on-
failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="300" --volume
eDir-volume1:/config --network=host edirectory:9.3.0 new -t docker-tree1 -
n novell -S m1 -B 164.99.1.1@524 -o 1028 -O 1030 -L 389 -l 636 --configure-
eba-now yes
```

次の例は、OpenText eDirectory コンテナレプリカサーバを既存のツリーに追加する方法を示しています。

```
docker run -it --name eDir-container-2 --stop-timeout 180 --restart on-
failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="300" --volume
eDir-volume2:/config --network=host edirectory:9.3.0 add -t docker-tree1 -
n novell -S m2 -B 164.99.10.10@2524 -o 2028 -O 2030 -L 2389 -l 2636 --
configure-eba-now yes -p 164.99.1.1@1524
```

EBA の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「EBA に関する情報の表示」を参照してください。

次のリストでは、OpenText eDirectory コンテナレプリカサーバを既存のツリーに追加するためのパラメータについて説明します。

***表 7-1** 環境設定パラメータの説明*

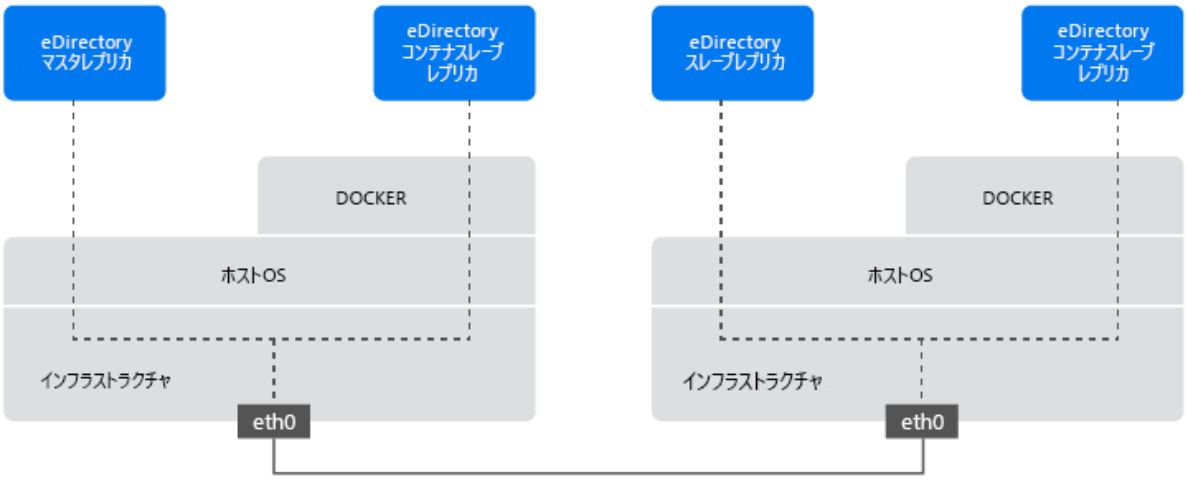
環境設定パラメータ	説明
--name eDir-container-1	docker コンテナの名前
--volume eDir-volume1	OpenText eDirectory データの保存に使用される docker ボリュームの名前

環境設定パラメータ	説明
edirectory:9.3.0	ダウンロードしてロードした Docker イメージ。バージョン番号に注意してください。ダウンロードした実際のバージョンではない可能性があっても、警告は表示されません。
-t docker-tree1	OpenText eDirectory ツリー名
-n novel	サーバオブジェクトが配置される OpenText eDirectory コンテナ
-S m1	OpenText eDirectory に作成されるサーバオブジェクトの名前
-B164.99.1.1	docker ホストの IP アドレス
-o 1028	OpenText eDirectory の HTTP ポート
-O1030	OpenText eDirectory の HTTPS ポート
-L 389OpenText	eDirectory の LDAP ポート
-l 636	OpenText eDirectory の LDAPS ポート
--configure-eba-now Yes	Yes - EBA を今すぐインストールします

注

- --network フラグは、ホストネットワーキングドライバを使用してコンテナをデプロイするために使用します。
- 同じ Docker ホスト上で実行されている OpenText eDirectory コンテナ間で、サービスのポート番号を繰り返すことはできません。
- 上記のコマンドで使用する IP アドレスは、コンテナが実行されることを想定している Docker ホストマシンのものです。

***図 7-1** ホストネットワークでの OpenText eDirectory コンテナのデプロイ*



ユーザ定義のオーバーレイネットワークでの OpenText eDirectory コンテナのデプロイ

ユーザ定義のオーバーレイネットワークを使用して、複数の Docker デーモンホスト上で実行される、OpenText eDirectory コンテナの分散ネットワークを作成できます。ユーザ定義のオーバーレイネットワーク内の OpenText eDirectory コンテナは、Linux と Windows の両方にデプロイできます。Docker ホストを Swarm に参加させ、それらのホスト上で実行されている OpenText eDirectory コンテナがシームレスに通信できるようにするには、Docker Swarm サービスを使用する必要があります。Docker オーバレイネットワークドライバの詳細については「Use Overlay Networks」を参照してください。

注 : Docker Swarm のスケーリングとスケジューリングの機能は、OpenText eDirectory コンテナでは認定されていません。Swarm サービスによるホスト間での OpenText eDirectory コンテナのマイグレーションもサポートされていません。

前提条件

- ❒ 少なくとも 1 つの Docker ホストを manager として設定し、他のホストを worker として設定して Docker Swarm を作成する必要があります。
- ❒ myOverlay という名前のアタッチ可能なオーバーレイネットワークを作成します。
- ❒ Docker Swarm 内でクラスタ管理と通信を行うため、Docker ホスト間のファイアウォールで次のポートを開きます。
  - TCP ポート 2377
  - TCP および UDP ポート 7946
  - UDP ポート 4789
- ❒ オーバーレイネットワークにデプロイされたコンテナには、myOverlay サブネットに属するスタティック内部 IP アドレスを割り当てる必要があります。

Swarm をデプロイし、ユーザ定義のオーバーレイネットワークを作成する方法については、「Networking with overlay networks」を参照してください。

ユーザ定義のオーバーレイネットワークに OpenText eDirectory コンテナをデプロイする前に、次の推奨事項を考慮する必要があります。

- OpenText eDirectory コンテナのマスタレプリカサーバとその R/W レプリカを同じオーバレイネットワーク内にデプロイする必要があります。オーバーレイネットワークの外部で実行されている他のスタンドアロンの OpenText eDirectory サーバまたはコンテナとの通信はサポートされません。
- Identity Console Docker コンテナは、OpenText eDirectory 管理用の同じユーザ定義のオーバーレイネットワークにデプロイすることをお勧めします。Identity Console Docker コンテナのデプロイ方法の詳細については、「Installing OpenText Identity Console as Docker Container」を参照してください。

- 次のコマンドを実行すると、ユーザ定義のオーバーレイネットワークのネットワークに関する詳細を確認できます。

  ```
  docker inspect myOverlay
  ```

次のコマンドは、OpenText eDirectory コンテナを使用して新しいツリーを作成する方法を示しています。

```
docker run -it --name eDir-container-1 --stop-timeout 180 --restart on-
failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="300" --volume
eDir-volume1:/config --network=myOverlay --ip=10.0.0.5 edirectory:9.3.0
new -t docker-tree1 -n novell -S m1 –B @524 -o 8028 -O 8030 -L 389 -l 636 -
-configure-eba-now yes
```

次のコマンドは、上で作成した OpenText eDirectory コンテナの IP アドレスを取得する方法を示しています。

```
docker inspect eDir-container-1 --format
{{.NetworkSettings.Networks.myOverlay.IPAddress}}
```

表示された IP アドレスは、OpenText eDirectory コンテナのレプリカサーバをツリーに追加するときに、remote_IP_Address として使用できます。

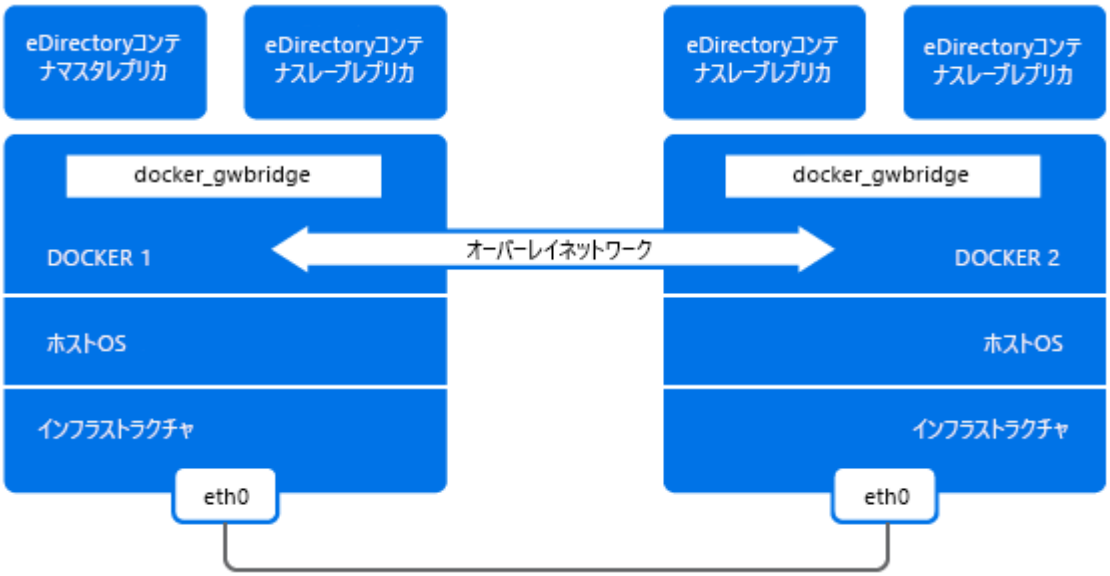
次のコマンドは、eDirectory コンテナレプリカサーバを既存のツリーに追加する方法を示しています。

```
docker run -it --name eDir-container-2 --stop-timeout 180 --restart on-
failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="300" --volume
eDir-volume2:/config --network=myOverlay --ip=10.0.0.6 edirectory:9.3.0
add -t docker-tree1 -n novell -S m2 –B @524 -o 8028 -O 8030 -L 389 -l 636 -
-configure-eba-now yes -p <remote_IP_Address>
```

注

- --network フラグは、オーバーレイネットワークドライバを使用して myOverlay というユーザ定義のオーバーレイネットワーク内にコンテナをデプロイするために使用されます。
- 上の例では、--ip フラグを使用して、myOverlay サブネットに属するコンテナにスタティック内部 IP アドレスを割り当てます。
- -B @524 オプションを使用すると、NCP、http、および https サービスを、使用可能なすべてのインターフェイスにバインドできるようになります。

図 7-2 ユーザ定義のオーバーレイネットワークでの OpenText eDirectory コンテナのデプロイ



デプロイメント後のタスク

次のタスクを、OpenText eDirectory コンテナのデプロイ後に実行する必要があります。

- 165 ページの 「実行中の OpenText eDirectory コンテナでのコマンドの実行」
- 166 ページの 「OpenSLP for OpenText eDirectory Docker コンテナの設定」
- 166 ページの「OpenText eDirectory Docker コンテナへの NMAS メソッドのインストール」
- 167 ページの 「OpenText eDirectory Docker コンテナへの新しいパッケージのインストール」
- 167 ページの「OpenText eDirectory Docker コンテナでの CEF プロパティファイルの設定」

注 : OpenText eDirectory Docker イメージの OS ベースイメージは、sysVinit を init システムとして使用した openSUSE Leap です。OpenText eDirectory サービスの実行に必要なすべての環境変数を、pre_ndsd_start_ ファイルに追加する必要があります。

実行中の OpenText eDirectory コンテナでのコマンドの実行

Docker ホストマシンで次のコマンドを実行すると、OpenText eDirectory Docker コンテナで bash シェルを使用できます。

```
docker exec -it eDir-container-1 /bin/bash
```

上記のコマンドでは、OpenText eDirectory バイナリパスを /opt/novell/eDirectory/bin に設定します。

NDS ユーティリティコマンドは、コンテナプロンプトで実行できます。以下に例を示します。

```
nds@abbae7c93b1c:~> ndsstat

Tree Name: EDR93E
Server Name: .CN=Master.O=server.T=EDR93E.
Binary Version: 40301.00
Root Most Entry Depth: 0
Product Version: OpenText eDirectory for Linux x86_64 v9.3 [DS]
```

上記のコマンドはホストマシンから直接実行できます。以下に例を示します。

```
docker exec -it eDir-container-1 /opt/novell/eDirectory/bin/ndsstat

Tree Name: EDR93E
Server Name: .CN=Master.O=server.T=EDR93E.
Binary Version: 40301.00
Root Most Entry Depth: 0
Product Version: OpenText eDirectory for Linux x86_64 v9.3 [DS]
```

OpenSLP for OpenText eDirectory Docker コンテナの設定

実行中の OpenText eDirectory コンテナで SLP サーバを起動するには、次の手順を実行します。

1. 次のコマンドを実行して、slpd を起動します。

   ```
   docker exec --user root eDir-container-1 /usr/sbin/slpd
   ```

2. 次のコマンドを実行して、OpenText eDirectory を再起動します。

   ```
   docker exec eDir-container-1 /opt/novell/eDirectory/bin/ndsmanage
   stopall

   docker exec eDir-container-1 /opt/novell/eDirectory/bin/ndsmanage
   startall
   ```

注

- コンテナを停止して再起動すると、SLP デーモンが停止します。手動で再起動する必要があります。/var/run/slpd.pid に古い PID ファイルが存在する場合は、その PID ファイルを削除してからデーモンを起動する必要があります。
- 古い PID ファイルは、停止したプロセスまたは強制終了されたプロセス (この例では、SLP デーモンプロセス) の PID を持つファイルです。
- オーバーレイ環境では、SLP DA を同じオーバレイネットワーク内で実行する必要があります。
- OpenSLP を設定しない場合は、エラーメッセージ -626 が表示されることがあります。

OpenText eDirectory Docker コンテナへの NMAS メソッドのインストール

OpenText eDirectory コンテナに NMAS メソッドをインストールするには、次のタスクを実行します。

注 : NMAS メソッドは、デフォルトでは /home/nds/eDirectory/nmas で使用できます。

1 次のコマンドを実行して、OpenText eDirectory コンテナにログインします。

```
docker exec -it eDir-container-1 bash
```

2 NMAS メソッドを追加します。

```
cd /home/nds/eDirectory/nmas/NmasMethods/Novell/<method-name>
```

```
nmasinst -addmethod admin.novell docker-tree1 ./config.txt
```

注 : NMAS メソッドを追加する方法の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*nmasinst ユーティリティを使用したログインメソッドのインストール*」を参照してください。

3 コンテナコンソールを終了します。

```
exit
```

4 OpenText eDirectory コンテナを再起動します。

```
docker restart eDir-container-1
```

OpenText eDirectory Docker コンテナへの新しいパッケージのインストール

OpenText eDirectory イメージに含まれる基本 OS には、コンテナ環境での OpenText eDirectory アプリケーションの通常の機能に対する依存関係が含まれています。デフォルトではイメージに含まれていない新しいユーティリティまたはデバッグツールを導入する必要がある場合は、実行中のコンテナに個別に持ち込む必要があります。

次のコマンドを実行して、zypper を使用して、実行中の OpenText eDirectory コンテナに新しいパッケージをインストールします。

```
docker exec --user root -it eDir-container-1 zypper in <package>
```

または、docker cp コマンドを使用して Docker ホストマシンからコンテナに新しいパッケージを直接コピーした後、新しいパッケージをインストールすることもできます。詳細については、「docker cp」を参照してください。

OpenText eDirectory Docker コンテナでの CEF プロパティファイルの設定

キャッシュおよびローリングファイルのログを有効にするには、CEF プロパティファイルで次の変更を実行する必要があります。CEF プロパティファイル (auditlogconfig.properties) は、/config/eDirectory/inst/conf フォルダにあります。

```
# Cache location directory
# Directory should be available for creating cache files
log4j.appender.S.CacheDir=/config/eDirectory/inst/data

# Log file for appender R.
log4j.appender.R.File=/config/eDirectory/inst/data/log/cef-events.log
```

詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*CEF プロパティファイルの設定*」を参照してください。

注 : イベントシステムキャッシングを設定するには、OpenText eDirectory コンテナの /opt/novell/eDirectory/sbin フォルダにある pre_ndsd_start ファイルで、次のように環境変数を設定します。

```
export NDSD_EVENT_DISK_CACHE=1
export NDSD_EVENT_DISK_CACHE_DIR=/config/eDirectory/inst/data/data
```

OpenText eDirectory データストレージの管理

OpenText eDirectory のデータと環境設定を永続的に保存するには、Docker Volume を使用することをお勧めします。永続的に保存することの詳細については「Manage data in Docker」参照してください。

永続的に保存する必要のある OpenText eDirectory アプリケーションデータは、起動時にコンテナ内の /config ディレクトリの下に配置されます。Docker ホストファイルシステム上のデータをコンテナの外部に永続的に保存するには、Docker ボリュームを OpenText eDirectory コンテナ内の /config パスにマウントする必要があります。管理目的でコンテナが停止または削除されても、ボリューム内のアプリケーションデータは保持されます。

この方法は、OpenText eDirectory コンテナのアップグレード時に古い設定とデータを保持する場合に便利です。eDirectory コンテナのアップグレードの詳細については、169 ページの 「Docker コンテナを使用した OpenText eDirectory のアップグレード」を参照してください。

次の例は、eDir-volume-1 という名前の Docker ボリュームを作成する方法を示しています。

```
docker volume create eDir-volume-1
```

次の例は、保存する目的でボリュームをマウントして OpenText eDirectory コンテナを起動する方法を示しています。

```
docker run -it --name eDir1-Host --restart on-failure:5 --memory="700M" --
cpuset-cpus="1" --pids-limit="300" --volume eDir-volume1:/config --
network=host edir930:latest new -t docker-tree1 -n novell -S m1 -B
164.99.179.213@1524 -o 1028 -O 1030 -L 1389 -l 1636 --configure-eba-now yes
```

上記のコマンドでは、eDir-volume1 という Docker ボリュームが作成され、OpenText eDirectory コンテナ内の場所 /config にマウントされます。

Docker コンテナを使用した OpenText eDirectory のアップグレード

OpenText eDirectory イメージの新しいバージョンが使用可能になった場合、管理者は、最新バージョンの OpenText eDirectory をコンテナにデプロイするためのアップグレード手順を実行できます。アップグレードを実行する前に、すべての必要なアプリケーション関連データを Docker ボリュームに永続的に保存するようにしてください。Docker コンテナを使用して OpenText eDirectory をアップグレードするには、次の手順を実行します。

1. 実行中の eDirectory コンテナを停止して削除します。実行中のコンテナは、新しいイメージを使用できないため、アップグレードを実行する前に停止して削除する必要があります。
2. 新しい OpenText eDirectory Docker イメージと、Docker ボリュームに保存されている古いコンテナのアプリケーションデータを使用して、新しいコンテナを起動します。

   次の例は、保存する目的でボリュームをマウントして OpenText eDirectory コンテナを起動する方法を示しています。

   ```
   docker run -it --name eDir1-Host --stop-timeout 180 --restart on-failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="300" --volume eDir-volume1:/config --network=host <eDirectory_image> new -t docker-tree1 -n novell -S m1 -B <Host_IP_Address>@1524 -o 1028 -O 1030 -L 1389 -l 1636 --configure-eba-now yes
   ```

   次の例は、ステップ 2 で作成した OpenText eDirectory コンテナをアップグレードする方法を示しています。

   ```
   docker run -it --name eDir1-Host --stop-timeout 180 --restart on-failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="300" --volume eDir-volume1:/config --network=host <Latest_eDirectory_image> upgrade
   ```

   eDir-volume1 は、古い OpenText eDirectory コンテナのアプリケーションデータを保持している同じボリュームです。

注

- OpenText eDirectory コンテナをアップグレードするには、その前に、古いバージョンのイメージが実行されているコンテナを削除する必要があります。
- ndsconfig コマンドの -a オプションと -w オプションを使用しないでください。コンテナのアップグレード時には、画面上のプロンプトを使用して管理者の資格情報を入力してください。

OpenText eDirectory Docker コンテナの復旧

不明な理由が原因で、実行中の OpenText eDirectory コンテナにアクセスできなくなったり、コンテナが削除されたり、使用不能になったりした場合は、コンテナの復旧を実行する必要があります。このような場合には、影響を受けたコンテナを停止して削除する必要があ

ります。新しいコンテナは、影響を受けたコンテナと同じ OpenText eDirectory イメージと Docker ボリュームを使用して開始する必要があります。OpenText eDirectory コンテナを復旧するには、次の手順を実行します。

1. 影響を受けたコンテナを停止して削除します。
2. 影響を受けたコンテナと同じ OpenText eDirectory イメージとボリュームを使用して、新しいコンテナを起動します。次の例は、影響を受けたコンテナを復旧する方法を示しています。

   ```
   docker run -it --name eDir1-Host --stop-timeout 180 --restart on-
   failure:5 --memory="700M" --cpuset-cpus="1" --pids-limit="150" --volume
   eDir-volume1:/config --network=host <same_eDirectory_image> upgrade
   ```

   eDir-volume1 は、影響を受けた OpenText eDirectory コンテナのアプリケーションデータを保持している同じボリュームです。

8 IPv6 アドレスを使用する Linux と Windows での OpenText eDirectory のインストール

OpenText eDirectory 932 では IPv4 アドレスと IPv6 アドレスの両方がサポートされています。OpenText eDirectory のインストール処理中に IPv6 アドレスを有効にできます。以前のバージョンからアップグレードしている場合、IPv6 アドレスを手動で有効にする必要があります。

OpenText eDirectory 9.3 では、デュアル IP スタック方式、トンネリング方式、およびピュア IPv6 移行方式をサポートしています。グローバルの IP アドレスのみがサポートされます。次に例を示します。

- [2015::12]
- [2015::12]:524

IPv6 アドレスは角括弧 [ ] 内に指定しなければならないということを除けば、OpenText eDirectory の機能は IPv6 でも IPv4 と同じです。IP アドレスの代わりにホスト名を使用することもできます。ホスト名を用いる場合、etc\hosts ファイルでホスト名を指定し、それを IPv6 アドレスと関連付ける必要があります。

IPv6 アドレスを使用している OpenText eDirectory ユーティリティの例を以下に示します。

```
ndsstat -h [6015:abc:def:123:456:12:0:123]

ndsstat -h [6015:abc:def:123:456:12:0:123]:524

ndslogin -h [2015::4] admin.organization

ndscheck -h [6015:abc:def:123:456:12:0:123] -a admin.organization -w
password

ldapadd -h [2015::4] -p 389 -D cn=admin,o=organization -w password -f
adduser.ldif

ldapdelete -h [6015:abc:def:123:456:12:0:123] -p 389 -D
cn=admin,o=organization -w password cn=user21,o=organization

ldapmodify -h [2015::4] -p 389 -D cn=admin,o=organization -w password -f
modify.ldif

ldapsearch -h [6015:abc:def:123:456:12:0:123] -p 389 -D
cn=admin,o=organization -w password -b o=organization
objectclass=inetorgperson

http://[2015::3]:8028/nds
```

OpenText eDirectory 9.3 ではリンクローカルアドレスタイプ、IPv4 マップド IPv6 アドレスタイプ、および IPv4 互換 IPv6 アドレスタイプはサポートしていません。

次のセクションでは、IPv6 がすでに設定されている Linux および Windows に OpenText eDirectory 9.3 をインストールして設定する方法を示します。

- 172 ページの 「IPv6 を使用する Linux での OpenText eDirectory の設定」
- 174 ページの「IPv6 を使用する Windows での OpenText eDirectory のインストールもしくはアップグレード」

IPv6 の Linux プラットフォームと Windows プラットフォームの違いについては、「「IPV6 に関する問題のトラブルシューティング」」を参照してください。

IPv6 を使用する Linux での OpenText eDirectory の設定

このセクションでは IPv6 アドレスをすでにサポートしている Linux コンピュータ上で OpenText eDirectory を設定する方法について説明します。

新しい OpenText eDirectory ツリーを作成する

ndsconfig コマンドで -B オプションを用いて IPv6 アドレスを指定することで、IPv6 アドレスを用いた OpenText eDirectory ツリーを新しく設定することができます。次に例を示します。

```
ndsconfig new -t CORP-TREE -B [2015::3]@524 -P ldap://[2015::3]:389,ldaps://[2015::3]:636
```

LDAP リスナーに自動的に IPv6 アドレスのリスンを開始させるには、OpenText eDirectory の設定中に -P オプションで LDAP URL を指定する必要があります。初期設定の段階で指定しない場合は、後から ldapconfig コマンドまたは Identity Console を用いて ldapInterfaces 属性に追加できます。詳細については、173 ページの 「IPV6 の LDAP URL の LDAP サーバオブジェクトへの追加」を参照してください。

既存の OpenText eDirectory ツリーへのサーバの追加

ndsconfig コマンドの -B オプションを用いて IPv6 アドレスを指定することで、既存の IPv6 のツリーにサーバを追加できます。次に例を示します。

```
ndsconfig add -t CORP-TREE -B [2015::4]@524 -P ldap://[2015::4]:389,ldaps://[2015::4]:636
```

LDAP リスナーに自動的に IPv6 アドレスのリスンを開始させるには、OpenText eDirectory の設定中に -P オプションで LDAP URL を指定する必要があります。初期設定の段階で指定しない場合は、後から ldapconfig コマンドまたは Identity Console を用いて ldapInterfaces 属性に追加できます。詳細については、173 ページの 「IPV6 の LDAP URL の LDAP サーバオブジェクトへの追加」を参照してください。

既存のもしくはアップグレードした OpenText eDirectory サーバでの IPv6 アドレスの有効化

1 /etc/opt/novell/eDirectory/conf/nds.conf ファイルに、ポート番号を含む IPv6 インタフェースアドレスを追加します。コンピュータに複数のインスタンスが設定されている場合は、それぞれの環境設定ファイルにこのアドレスを追加する必要があります。

次に例を示します。

```
n4u.server.interfaces=164.99.90.148@524,[2015::4]@524,[2015:1234:2345:3456:abcd:bcde:cdef:aaaa]@524
```

```
http.server.interfaces=164.99.90.148@8028,[2015::4]@8028,[2015:1234:2345:3456:abcd:bcde:cdef:aaaa]@8028
```

```
https.server.interfaces=164.99.90.148@8030,[2015::4]@8030,[2015:1234:2345:3456:abcd:bcde:cdef:aaaa]@8030
```

2 以下のコマンドを使用して、ndsd を再起動します。

```
ndsmanage stopall
ndsmanage startall
```

IPV6 の LDAP URL の LDAP サーバオブジェクトへの追加

OpenText eDirectory の初期設定の段階で LDAP URL を指定しない場合は、後から ldapconfig コマンドまたは Identity Console を用いて、ldapInterfaces 属性に LDAP URL を追加できます。

以下は、ldapconfig set および ldapconfig-s コマンドの使用例です。

```
ldapconfig set "ldapInterfaces=ldap://[2015::3]:389,ldaps://[2015::3]:636"
```

```
ldapconfig -s
"ldapInterfaces=ldap://[2015::3]:389,ldapInterfaces=ldaps://[2015::3]:636"
```

Identity Console に LDAP URL を追加するには :

1 OpenText Identity Console にログインし、［**LDAP 環境設定**］をクリックします。

2 ［**検索結果**］メニューで、［LDAP サーバ - <server name/CN>］を選択します。

3 ［**LDAP サーバの変更**］>［**情報**］>［LDAP インターフェイス］の順にクリックします 。

4 ［**項目の追加**］ウィンドウで、［LDAP インターフェイス］フィールドに LDAP URLS を追加します。

5 ［**追加**］をクリックしてから、［**保存**］をクリックします。

「LDAP サーバが正常に変更されました !」というメッセージが表示されたら、［**OK**］をクリックします。

IPv6 を使用する Windows での OpenText eDirectory のインストールもしくはアップグレード

このセクションでは IPv6 アドレスをすでにサポートしている Windows コンピュータ上で OpenText eDirectory を設定する方法について説明します。

OpenText eDirectory のインストール中またはアップグレード中に行う IPv6 の有効化

IPv6 アドレスを使用する場合は、OpenText eDirectory のインストール中に［**IPV6 の初期設定**］の下の［**IPV6 を有効にする**］チェックボックスを選択します。選択すると DHost が IPv6 アドレスの認識を開始します。インストールプロセス中に IPv6 アドレスを有効にせず、後から使用することにした場合は、セットアッププログラムを再度実行する必要があります。

既存のサーバ上での IPv6 の有効化

設定済みの OpenText eDirectory サーバで IPv6 アドレスを使用する場合は、インストールを再実行し、［**IPV6 の初期設定**］の下の［**IPV6 を有効にする**］チェックボックスを選択する必要があります。それにより IPv6 アドレスの、NCP、HTTP、および HTTPS プロトコルが有効になります。

OpenText iMonitor へのアクセス

IPv6 アドレスを介して OpenText iMontior にアクセスするためのリンクは次のとおりです。

```
http://[2015::3]:8028/nds
```

9 FIPS モードでの OpenText eDirectory の運用

OpenText eDirectory 9.3 は、非常にセキュアな環境を使用している米国連邦政府機関や顧客のセキュリティ要件を満たすため、FIPS (連邦情報処理規格) に準拠した機能を利用しています。この章では、FIPS モードでの OpenText eDirectory の環境設定と運用について説明します。

NICI および OpenSSL モジュールでサポートされる FIPS 140-2 モードで OpenText eDirectory を実行できます。

- 175 ページの 「FIPS モードでの OpenText eDirectory の設定」

FIPS モードでの OpenText eDirectory の設定

OpenText eDirectory 9.3 以降では、デフォルトですべてのプラットフォームで NICI および OpenSSL の両方で FIPS モードが無効になっています。OpenText eDirectory には、要件に合わせて FIPS モードを設定するためのスイッチが用意されています。OpenText eDirectory サーバで FIPS モードが有効になっている場合、OpenText eDirectory 内で実行されているアプリケーションとモジュール (OpenSSL を含む) はすべて FIPS モードになります。たとえば、LDAP、HTTP、EBA 内のすべての暗号化処理などがこれに該当します。FIPS モードで OpenText eDirectory を動作させると、SSLv3 経由の通信が許可されず、サイファの使用が強度の高いサイファのみに制限されます。詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「LDAP オブジェクトを環境設定する」および「*HTTP サーバオブジェクトの設定*」を参照してください。

OpenText eDirectory で FIPS モードを有効にするには、フラグを 1 つのみ設定する必要があります。これにより、NICI と OpenSSL の両方が有効にされます。

NICI FIPS を有効にし、NDS ログインを機能させる前に、次の前提条件を実行します。

1. PBKDF2 (Password-Based Key Derivation Function 2) パスワードポリシーを作成します。詳細については、「非可逆パスワードストレージを理解する」を参照してください。次に、すべてのユーザに PBPDF2 パスワードポリシーを割り当てます。
2. SCRAM をデフォルトのログインシーケンスとして設定します。詳細については、「パスワード認証」を参照してください。
3. PBKDF2 ポリシーが割り当てられたら、NICI で FIPS を有効にする前に、すべてのユーザが自分のパスワードを変更する必要があります。
4. すべてのユーザのパスワードが変更された後、NICI FIPS を有効にします。FIPS モードで NICI を有効にするには、「システムレベルの FIPS モードの設定に NICI を使用する (Using NICI for Configuring System-Level FIPS Mode)」を参照してください。
5. NICI の TLS と FIPS は、nici.cfg に基づいて有効化されます。

NICI と OpenSSL の両方に対して OpenText eDirectory で FIPS モードを有効にするには :

- **Windows:** Windows では FIPS を有効にできます。Windows レジストリで NICI 制限レベルフラグを 1 に設定します。詳細については、「システムレベルの FIPS モードの設定に NICI を使用する (Using NICI for Configuring System-Level FIPS Mode) (https://www.netiq.com/documentation/edirectory-93/nici_admin_guide/data/agbdek3.html#b1habb0n)」を参照してください。
- **Linux:** NICI 内の FIPS は、nici.cfg の設定に基づいて有効化されます。詳細については、「システムレベルの FIPS モードの設定に NICI を使用する (Using NICI for Configuring System-Level FIPS Mode) (https://www.netiq.com/documentation/edirectory-93/nici_admin_guide/data/agbdek3.html#b1habb0n)」を参照してください。

注 : FIPS をいったん有効した場合、後から無効にすることはお勧めできません。

---

10 DIB の移動

OpenText eDirectory をインストールして設定した後、DIB の移動が必要な場合は、DIB を移動できます。DIB の移動が必要となる理由はさまざまです。たとえば、ツリー内のオブジェクト数の増加が予想されるが、DIB がある現在のファイルシステムに十分な容量がない場合などが挙げられます。

Linux

DIB を移動するには、次の手順を実行します。

1. コマンドラインで次のコマンドを入力して、サーバの状態を確認します。

   ndscheck

2. 次のようにして、ndsmanag を使用して OpenText eDirectory サービスを停止します。

   2a コマンドプロンプトで「ndsmanage」と入力します。

   2b 停止するインスタンスを選択します。

   メニューが拡張し、特定のインスタンスに対して実行可能なオプションが表示されます。

   2c インスタンスを停止するには、「k」と入力します。

3. 次のコマンドを使用して、DIB の現在位置を取得します。

   ```
   ndsconfig get n4u.nds.dir
   ```

4. 次のとおり、DIB を新しい場所にコピーします。

   ```
   cp -rp current_location new_location
   ```

   たとえば、DIB を /home/nds/ にコピーするには、次のコマンドを入力します。

   ```
   cp -rp /var/opt/novell/eDirectory/data//* /home/nds//
   ```

5. インスタンス固有の nds.conf 環境設定ファイルを編集して、次のように n4u.nds.dir のパラメータ値を変更します。

   ```
   n4u.nds.dir=new_location
   ```

   たとえば、DIB のパスを /var/nds/ から /home/nds/ に変更する場合は、次のように入力します。

   ```
   n4u.nds.dir=/home/nds/
   ```

6. 次のようにして、OpenText eDirectory サービスを始動します。

   6a コマンドプロンプトで「ndsmanage」と入力します。

   6b 開始するインスタンスを選択します。

メニューが拡張し、特定のインスタンスに対して実行可能なオプションが表示されます。

6c インスタンスを開始するには、「s」と入力します。

7 サーバステータスを次のようにチェックします。

```
ndscheck
```

Windows

DIB の移動は現在サポートされていません。ただし、OpenText eDirectory のインストール中に、DIB を任意の場所に配置することができます。

11 OpenText eDirectory 9.3 のアップグレード要件

OpenText eDirectory の独特の機能の 1 つとして、厳格なリファレンス整合性を維持することができるようになっています。最上位から派生しているオブジェクトクラスのクラス定義はどれも、リファレンス属性を持つようになります。これは、OpenText eDirectory が内部的に維持している、すべてのリファレンス先オブジェクトに追加される隠し属性です。リファレンス先オブジェクトとリファレンス元オブジェクト間のリンクをチェックするバックグラウンドプロセスは、継続的に実行されます。

リファレンス先オブジェクトのパーティションがサーバ内でローカルに保持されていたオブジェクトのパーティションと異なる場合、そのオブジェクトへの外部参照が外部参照パーティション内のローカルで作成されます。外部参照とは、OpenText eDirectory ツリー内に存在するオブジェクトを表現したものです。ただし、外部参照はオブジェクトとそのオブジェクトの割り当てられた属性のコピーではありません。

OpenText eDirectory からリファレンス属性を削除することはできますが、ツリーの後方互換性を維持するため、クラス定義を操作することは今のところできません。

この章では、OpenText eDirectory 9.3 の変更点、および考えられるアップグレードシナリオについて説明します。

- 179 ページの 「8.8 SP1 以降のバージョンのリファレンスに関する変更点」
- 180 ページの 「9.3 のアップグレードプロセス」

8.8 SP1 以降のバージョンのリファレンスに関する変更点

リファレンス属性は隠し属性で、リファレンス先オブジェクトごとに保持されます。これは、DS によって作成および保持されます。DS の新しいリファレンスコードは、Flexible Adaptable Information Manager (FLAIM) インデックスをベースにしています。これは DS が作成する LocalEntryIDIndex と呼ばれています。FLAIM はインデックスを維持しますが、使用方法は DS が決定します。DN 値が追加または削除されると、FLAIM はインデックスを自動的に更新します。インデックス内の各キーは合成キー (リファレンス先のオブジェクトの DN+ リファレンス元オブジェクトのエントリ ID) です。たとえば、エントリ ID343 のオブジェクトがあり、オブジェクト #899 をポイントする「メンバー」値が存在する場合、FLAIM は 899+343 というインデックスでキーを自動生成します。これで DS はインデックス内のルックアップを実行し、オブジェクト #899 をポイントしているすべてのオブジェクトを検索できるようになります。参照元のすべてのオブジェクトを記憶するために、オブジェクト #899 自体がリファレンス属性を保持する必要はありません。実際、FLAIM はインデックスがどのように使われるかを認識せずにインデックスを保持しますが、DS にはインデックスの使用方法を認識しているコードが存在します。

注 : OpenText eDirectory をバージョン 8.8 SP1 から 9.3 に直接アップグレードすることはしないようにお勧めします。9.2 にアップグレードしてから、9.3 にアップグレードする必要があります。OpenText eDirectory を 8.8 SP1 より前のバージョンから 9.2 にアップグレードすると、OpenText eDirectory サーバのパフォーマンスに影響が出ます。これは、すべてのインデックスが自動的に再作成されるために発生します。8.8 SP1 以降のバージョンからアップグレードする場合、この問題は発生しません。

ただし、リファレンスを維持する新しい方法では、既存の OpenText eDirectory インスタンスを 8.8 以降のバージョンにアップグレードするときにデータベースをアップグレードする必要があります。アップグレードでは、新しいインデックスの作成が必要で、そこでデータベースの各エントリを詳細に検証することが必要になります。また、データベース内の各エントリから「リファレンス」属性をすべて削除することも必要です。さらに、DN を埋め込んでいた DS が使用する内部オクテット文字列属性が、オクテット文字列値と一緒に保存する新しい DN 値を生成する必要も出てきます。これらの処理はすべて、大規模なデータベースでは非常に時間のかかるプロセスです。DS は新しい FLAIM 機能を使ってリファレンスの整合性を保持するように変更されており、それが新しいインデックスに依存しているため、この変換が完了するまで、DS を実際に稼働できません。したがって、初めて既存データベースを開いたときに、すべてのリファレンス属性を新しいインデックスに変更する必要があります。大型データベースの場合、実際に開いてアプリケーションが使用できる状態になるまで数時間かかることがあります。

9.3 のアップグレードプロセス

ndsconfig upgrade コマンドは、HTTP、LDAP、SNMP、SAS、および NMAS などのコンポーネントで必要な環境設定をアップグレードするために使用します。

強制オプションを使って Linux の OpenText eDirectory を旧バージョンからアップグレードする

Linux では、OpenText eDirectory 9.3 は、9.2 以降のバージョンからのアップグレードのみに対応しています。

OpenText eDirectory のバージョンを 8.8.8 からアップグレードするには、OpenText eDirectory 9.2 にアップグレードしてから、eDirectory 9.3 にアップグレードします。eDirectory 9.2 にアップグレードした後、識別情報コンソールで修復証明書を実行してから、OpenText eDirectory 9.3 にアップグレードします。デフォルト証明書の修復の詳細については、次を参照してください。

重要 : Identity Manager 変更ログモジュールがすでにインストールされている場合は、OpenText eDirectory を 9.2 にアップグレードするときに、環境変数 NDSD_IGNORE_IDM_CHECK を 1 に設定する必要があります。次に例を示します。

- Linux の場合 : NDSD_IGNORE_IDM_CHECK=1 ./nds-install
- Windows の場合 : setup.exe を実行する前に NDSD_IGNORE_IDM_CHECK を true に設定します。

12 Linux 上での OpenText eDirectory の設定

OpenText eDirectory には、Linux コンピュータ上でのさまざまな OpenText eDirectory コンポーネントの設定を容易にする設定ユーティリティが含まれています。次のセクションでは、OpenText eDirectory 環境設定コンポーネントの機能および使用法について説明します。

- 181 ページの 「環境設定ユーティリティ」
- 184 ページの 「環境設定パラメータ」
- 190 ページの 「セキュリティ上の考慮事項」

環境設定ユーティリティ

このセクションでは、次の OpenText eDirectory 環境設定ユーティリティの使用法について説明します。

- 181 ページの 「ndsconfig ユーティリティ」
- 182 ページの「LDAP ツールを使用して LDAP サーバと LDAP グループオブジェクトを背呈する」
- 182 ページの 「nmasinst ユーティリティを使用して NetIQ Modular Authentication Service を設定する」
- 182 ページの 「OpenText eDirectory のカスタマイズ」

ndsconfig ユーティリティ

ndsconfig ユーティリティを使用して、OpenText eDirectory を設定できます。このユーティリティは、既存のツリーに OpenText eDirectory レプリカサーバを追加するときや新しいツリーを作成するときにも使用できます。詳細については、56 ページの 「dsconfig ユーティリティを使用して OpenText eDirectory レプリカサーバを追加または削除する」を参照してください。

注

- NCP サーバ名がネットワーク内で一意であることを確認してください。

インストールされているコンポーネントの現在の設定を変更するには、次の構文を使用します。

```
ndsconfig {set value_list | get [parameter_list] | get help
[parameter_list]}
```

ndsconfig パラメータの説明については、「184 ページの 「環境設定パラメータ」」を参照してください。

重要 : インストール後、ndsconfig ユーティリティをサーバのインストール場所から実行します。この場所はデフォルトでは /opt/novell/eDirectory/bin です。インストールパッケージから ndsconfig を実行しないでください。

LDAP ツールを使用して LDAP サーバと LDAP グループオブジェクトを背呈する

Linux コンピュータ上で LDAP サーバおよびグループオブジェクトの属性を変更、表示、およびリフレッシュするには、OpenText eDirectory に同梱されている LDAP ツールを使用できます。

詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*Linux での LDAP ツールの使用*」を参照してください。

nmasinst ユーティリティを使用して NetIQ Modular Authentication Service を設定する

OpenText eDirectory 9.3 では、デフォルトで、ndsconfig を使用して NMAS を設定します。nmasinst を使用して NMAS を設定することもできます。

ndsconfig が行うのは NMAS 設定のみです。ログインメソッドのインストールは行いません。ログインメソッドのインストールには、nmasinst を使用できます。詳細については、「72 ページの 「nmasinst ユーティリティを使用して NMAS を設定する」」を参照してください。

OpenText eDirectory のカスタマイズ

- 182 ページの 「ndsd init スクリプトを使用する」
- 183 ページの「SLES 15 および RHEL 8 プラットフォームで OpenText eDirectory を使用する」

ndsd init スクリプトを使用する

システムが起動すると、デフォルトの環境設定ファイル /etc/opt/novell/eDirectory/conf/nds.conf の環境設定パラメータを使用して、ndsd init スクリプトがデーモンを開始します。

注 : systemd 環境では /etc/init.d/ndsd スクリプトを使用しないでください。Systemd は現在、SLES 15 および RHEL 8 および RHEL 9 以上のプラットフォームでのみサポートされています。詳細については、183 ページの 「SLES 15 および RHEL 8 プラットフォームで OpenText eDirectory を使用する」を参照してください。

ndsd を開始する前に、すべての SLP (Service Location Protocol) エージェントがホスト上で実行されていることを確認してください。OpenSLP、ご利用のオペレーティングシステムで使用可能な SLP、または NetIQ SLP がインストール可能です。

注 : OpenText eDirectory を開始するには、ndsmanage ユーティリティを使用します。

注 : 以下のコマンドを実行して、SLES 15 および RHEL8 で OpenText eDirectory を開始および停止します。

- ndsd を開始するには、systemctl start ndsd* を実行します
- ndsd を停止するには、systemctl stop ndsd* を実行します

OpenText eDirectory の設定によって、/opt/novell/eDirectory/sbin 内に次のシェルスクリプトが作成されます。

- pre_ndsd_start
- post_ndsd_start
- pre_ndsd_stop
- post_ndsd_stop

名前が示すように、ndsd バイナリが /etc/init.d/ndsd スクリプトによって起動される前に、pre_ndsd_start スクリプトが実行されます。post_ndsd_start スクリプトは、/etc/init.d/ndsd スクリプトによって ndsd バイナリが起動された後に実行されます。同様に、pre_ndsd_stop と post_ndsd_stop スクリプトも、ndsd プロセスを停止する前と後にそれぞれ実行されます。

選択したコマンドをこれらのスクリプトに追加して、実行することができます。デフォルトでは、LDAP サービスが起動した後 /etc/init.d/ndsd を実行するコマンドが post_ndsd_start スクリプトに記述されています。

注 : /etc/opt/novell/eDirectory/conf ディレクトリにある env_custom スクリプトで、OpenText eDirectory サービスに必要なすべての環境変数を追加する必要があります。端末または /etc/init.d/ndsd スクリプトでの環境変数のエクスポートは、OpenText eDirectory では使用されません。環境変数の詳細については、「TID 7018431」を参照してください。

SLES 15 および RHEL 8 プラットフォームで OpenText eDirectory を使用する

システムが起動すると、デフォルトの環境設定ファイル /etc/opt/novell/eDirectory/conf/nds.conf の環境設定パラメータを使用して、OpenText eDirectory がデーモンを開始します。

ndsd を開始する前に、すべての SLP (Service Location Protocol) エージェントがホスト上で実行されていることを確認してください。OpenSLP、ご利用のオペレーティングシステムで使用可能なネイティブ SLP、または OpenText SLP がインストール可能です。

OpenText eDirectory を開始または停止するには、ndsmanage ユーティリティを使用します。

OpenText eDirectory の設定によって、/opt/novell/eDirectory/sbin 内に次のシェルスクリプトが作成されます。

- pre_ndsd_start_custom: eDirectory を実行する前にこのスクリプトを使用して、コマンドをカスタムで追加します。

- post_ndsd_start_custom: eDirectory を実行した後でこのスクリプトを使用して、コマンドをカスタムで追加します。
- post_ndsd_stop_custom: eDirectory を停止した後でこのスクリプトを使用して、コマンドをカスタムで追加します。

注

- /opt/novell/eDirectory/sbin にある工場出荷時のスクリプトは使用しないでください。OpenText eDirectory の設定では、工場出荷時のスクリプトが使用されます。選択したコマンドを含めるには、カスタムスクリプトを使用してください。
- オペレーティングシステムをアップグレードした後、ndsconfig upgrade ユーティリティを実行します。

環境設定パラメータ

OpenText eDirectory の環境設定パラメータは nds.conf ファイルに格納されています。

環境設定パラメータを変更した場合、新しい値を有効にするには ndsd を再起動する必要があります。ndsd を再起動するには、ndsmanage を使用してください。

ただし、環境設定パラメータによっては ndsd を再起動する必要がない場合があります。再起動の必要のないパラメータは次のとおりです。

- n4u.nds.inactivity-synchronization-interval
- n4u.nds.synchronization-restrictions
- n4u.nds.janitor-interval
- n4u.nds.backlink-interval
- n4u.nds.drl-interval
- n4u.nds.flatcleaning-interval
- n4u.nds.server-state-up-thresholdn4u.nds.heartbeat-scheman4u.nds.heartbeat-data

次の表では、すべての環境設定パラメータの説明を示します。

パラメータ	説明
n4u.nds.preferred-server	OpenText eDirectory サービスをホストするマシンのホスト名です。 デフォルト値は NULL です。
n4u.base.tree-name	Account Management が使用するツリー名です。この必須パラメータは、Account Management のインストーラによって設定されます。このパラメータを設定することはできません。

パラメータ	説明
n4u.base.dclient.use-udp	DClient では、OpenText eDirectory サーバとの通信に、TCP のほかに UDP も使用できます。このパラメータにより、UDP 転送機能が使用できるようになります。 デフォルト値は 0 です。 範囲は 0 または 1 です。
n4u.base.slp.max-wait	SLP (Service Location Protocol) API 呼び出しのタイムアウトです。 デフォルト値は 30 です。 範囲は 3 ～ 100 です。 この値は秒単位で表します。 このオプションは、OpenText SLP によってのみサポートされ、OpenSLP ではサポートされません。
n4u.nds.advertise-life-time	指定の時間が過ぎると、OpenText eDirectory はディレクトリエージェントに再び自己登録します。 デフォルト値は 3600 です。 範囲は 1 ～ 65535 です。 この値は秒単位で表します。
n4u.server.signature-level	これにより、拡張セキュリティサポートのレベルが決まります。この値を大きくするとセキュリティは向上しますが、パフォーマンスは低下します。 デフォルト値は 1 です。 範囲は 0 ～ 3 です。
n4u.nds.dir	OpenText eDirectory ディレクトリ情報データベースです。 デフォルト : `/var/opt/novell/eDirectory/data/` ndsconfig set コマンドを使用してこのパラメータを設定することはできません。DIB を再配置する場合は、このパラメータを手動で変更できます。ただし、この操作はお勧めできません。
n4u.nds.server-guid	OpenText eDirectory サーバ全体で固有の識別子です。 デフォルト値は NULL です。

パラメータ	説明
n4u.nds.server-name	OpenText eDirectory サーバの名前です。 デフォルト値は NULL です。
n4u.nds.bindery-context	バインダリコンテキストの文字列です。 デフォルト値は NULL です。
n4u.nds.server-context	OpenText eDirectory サーバの追加先コンテキストです。このパラメータを設定または変更することはできません。
n4u.nds.external-reference-life-span	使用されていない外部参照を削除するまでの時間数です。 デフォルト値は 192 です。 範囲は 1 ～ 384 です。
n4u.nds.inactivity-synchronization-interval	レプリカの完全同期の実行後、サーバ上の OpenText eDirectory に格納されている情報が最初に変更されてから次にレプリカの完全同期を実行するまでの時間間隔 (分) です。 デフォルト値は 60 です。 範囲は 2 ～ 1440 です。
n4u.nds.synchronization-restrictions	値を Off に設定すると、OpenText eDirectory の任意のバージョンと同期できます。値を On に設定すると、同期するバージョン番号がパラメータで指定した値 (ON,420,421 など) に制限されます。 デフォルト = オフ
n4u.nds.janitor-interval	OpenText eDirectory ジャニタ処理の実行間隔 (分) です。 デフォルト値は 2 です。 範囲は 1 ～ 10080 です。
n4u.nds.backlink-interval	OpenText eDirectory バックリンクの整合性チェックの実行間隔 (分) です。 デフォルト値は 780 です。 範囲は 2 ～ 10080 です。
n4u.nds.drl-interval	OpenText eDirectory 分散参照リンクの整合性チェックの実行間隔 (分) です。 デフォルト値は 780 です。 範囲は 2 ～ 10080 です。

パラメータ	説明
n4u.nds.flatcleaning-interval	flatcleaner 処理によるデータベースからのエントリの自動パージおよび削除の実行間隔 (分) です。 デフォルト値は 720 です。 範囲は 1 ～ 720 です。
n4u.nds.server-state-up-threshold	サーバの状態の UP しきい値 (分) です。このしきい値に達すると、OpenText eDirectory はサーバの状態をチェックし、その後 -625 エラーを戻します。 デフォルト値は 30 です。 範囲は 1 ～ 720 です。
n4u.nds.heartbeat-schema	Heartbeat ベーススキーマの同期間隔 (分) です。 デフォルト値は 240 です。 範囲は 2 ～ 1440 です。
n4u.nds.heartbeat-data	Heartbeat 同期間隔 (分) です。 デフォルト値は 60 です。 範囲は 2 ～ 1440 です。
n4u.nds.dofsync	このパラメータを 0 に設定すると、大規模なデータベースで更新のパフォーマンスが大幅に上がります。しかし、システムがクラッシュした場合にはデータベースが破損する危険があります。
n4u.server.configdir	OpenText eDirectory の環境設定ファイルがここに配置されます。 デフォルト値は /etc です。
n4u.server.vardir	OpenText eDirectory およびユーティリティのログファイルがここに配置されます。 デフォルト値は /var/opt/novell/eDirectory/log です。
n4u.server.libdir	eDirectory 固有のライブラリは、nds-modules ディレクトリのこの場所に配置されます。 デフォルト値は /opt/novell/eDirectory/lib です。
n4u.server.sid-caching	SSL セッション ID の変更を有効にします。SSL のセッション ID のキャッシュの詳細については、『SSL v3.0 RFC』を参照してください。

パラメータ	説明
n4u.server.tcp-port	n4u.server.interfaces パラメータでポート番号が指定されない場合に使われるデフォルトポートです。
n4u.server.interfaces	OpenText eDirectory サーバが、クライアント接続のためにリスンする IP アドレスおよびポート番号です。設定の組み合わせを複数指定する場合は、値をコンマ区切りのリスト形式で指定できます。例 : n4u.server.interfaces=101.1.2.3@524,100.1.2.3@1524
n4u.server.max-interfaces	このパラメータは、OpenText eDirectory が使用するインターフェイスの最大数を指定します。 デフォルト値は 128 です。 範囲は 1 ～ 2048 です。
n4u.server.max-openfiles	このパラメータは、OpenText eDirectory が使用できるファイルデスクリプタの最大数を指定します。 デフォルト値は、管理者によって設定される最大値です
n4u.server.max-threads	OpenText eDirectory サーバが開始するスレッドの最大数です。この数は、OpenText eDirectory サーバ内で同時に実行できる同時並行操作の最大数です。 デフォルト値は 64 です。 範囲は 32 ～ 512 です。 最適値を設定するには、「*トラブルシューティング*」を参照してください。
n4u.server.idle-threads	OpenText eDirectory サーバで許可されるアイドル状態のスレッドの最大数です。 デフォルト値は 8 です。 範囲は 1 ～ 128 です。
n4u.server.start-threads	最初に開始されるスレッドの数です。 デフォルト値は 8 です。
n4u.server.log-levels	このパラメータは、サーバ側のメッセージのエラーログを設定するときに役立ちます。メッセージログのレベルは LogFatal、LogWarn、LogErr、LogInfo、LogDbg のいずれかに設定されます。

パラメータ	説明
n4u.server.log-file	このパラメータは、メッセージを書き込むログファイルの場所を指定します。デフォルトでは、ndsd.log ファイルにメッセージが書き込まれます。
n4u.ldap.lburp.transize	1 つの LBURP パケットで OpenText インポート / エクスポートクライアントから LDAP サーバに送られるレコード数です。トランザクションのサイズを増やし、1 つの要求で確実に複数の追加操作を実行することができます。 デフォルト値は 25 です。 範囲は 1 ～ 250 です。
n4u.server.listen-on-loopback	ブール値のパラメータで、デフォルトで有効になっています。最近のいくつかの Linux 配布パッケージでは、/etc/hosts ファイル内のホスト名がループバックアドレスに関連付けられています。SLES システムで提供される共通アドレスは 127.0.0.2 ですが、127.0.0.0 から 127.255.255.255 までの任意の値 (有効なループバックアドレス) にすることができます。
http.server.interfaces	HTTP サーバが使用するインタフェースのカンマ区切りのリストです。
http.server.request-io-buffer-size	デフォルトの IO バッファサイズです。
http.server.request_timeout-seconds	サーバ要求のタイムアウトです。
http.server.keep-timeout-seconds	同じ接続上にある同じクライアントからの次の要求を待つ秒数です。
http.server.threads-per-processor	プロセッサごとの HTTP スレッドプールのサイズです。
http.server.session-exp-seconds	セッションの有効期間 (秒) です。
http.server.sadmin-passwd	セッション管理者のパスワードです。
http.server.module-base	HTTP サーバの Web ルートです。
https.server.cached-cert-dn	HTTPS サーバがキャッシュした証明書の DN です。
https.server.cached-server-dn	HTTPS サーバがキャッシュした DN です。
http.server.trace-level	HTTP サーバの診断追跡レベルです。
http.server.auth-req-tls	HTTP サーバ認証に TLS が必要です。
http.server.clear-port	HTTP プロトコルのサーバポートです。
http.server.tls-port	HTTPS プロトコルのサーバポートです。

パラメータ	説明
n4u.server.fips	OpenText eDirectory サーバを FIPS モードで実行するかどうかを指定します。 デフォルトは 1 です。この場合、OpenText eDirectory は FIPS モードで実行されます。 FIPS モードを無効にするには、ndsconfig set コマンドで n4u.server.fips=0 を渡し、サーバを再起動します。

注 : OpenText eDirectory の環境設定パラメータの詳細については、nds.conf のマニュアルページを参照してください。

セキュリティ上の考慮事項

次のセキュリティ上の検討事項を推奨します。

- ツリーのブラウズ権限を認証されたユーザだけに割り当ててください。これを制限するには、次の操作を実行します。
  - ツールルートの［パブリック］のブラウズ権限を削除します。
  - ツリールートの［ルート］のブラウズ権限を割り当てます。
- LDAP サーバオブジェクトの ldapBindRestrictions 属性を、[匿名単純バインドを不許可にする] に設定します。これにより、クライアントが匿名バインドをしなくなります。

13 高可用性クラスタで OpenText eDirectory を展開する

OpenText eDirectory が高可用性をサポートする主な方法は、同期により複数サーバを設定する方法です。ただし一部の環境で高可用性を実現する場合、クラスタリングの方が実現性が高い選択肢となることがあります。

このセクションでは、共有ストレージを使って、高可用性クラスタ上で OpenText eDirectory を環境設定するためのガイドラインについて説明します。このセクションでは、サポート対象の Windows または Linux プラットフォームの、一般的な高可用性クラスタの共有ストレージについて説明しています。特定のクラスタマネージャに特化したものではありません。

OpenText eDirectory の状態データは、サービスを現在実行しているクラスタノードで使用できるように、共有ストレージに配置する必要があります。つまり、OpenText eDirectory の DIB は、クラスタの共有ストレージに配置する必要があります。各クラスタノード上の OpenText eDirectory ルートインスタンスは、共有ストレージの DIB を使用するよう設定する必要があります。

DIB の他に、サーバ固有のキーをクラスタノード間で複製するために、NICI のデータも共有する必要があります。すべてのクラスタノードが使用する NICI のデータは、クラスタ共有ストレージに配置する必要があります。

その他の OpenText eDirectory 設定データやログデータも、共有ストレージに常駐する必要があります。

OpenText·eDirectory·9.3 には、Linux サーバと Windows サーバの両方で、指定した共有ストレージ場所へのデータコピー、適切な環境設定パラメータの更新、プライマリノード以外のクラスタノード上での OpenText eDirectory サービスの設定など、クラスタ環境内の OpenText eDirectory を自動的に環境設定するユーティリティが用意されています。

以下のセクションの手順は、以下の前提条件に基づいています。

- ユーザが OpenText eDirectory のインストール手順に精通している。
- 2 ノードクラスタを使用している。

注 : 2 ノードクラスタは、高可用を実現するために使われる最小環境設定です。ただし、このセクションのコンセプトは、ノードを追加することで、簡単にクラスタに拡張することができます。OpenText eDirectory は複数のクラスタノードを使用する負荷分散をサポートしていないことに注意してください。

この章の構成は次のとおりです。

- 192 ページの 「Linux での OpenText eDirectory サービスのクラスタリング」
- 195 ページの 「Windows での OpenText eDirectory サービスのクラスタリング」

- 198 ページの 「クラスタ化環境のトラブルシューティング」
- 199 ページの 「環境設定ユーティリティのオプション」

Linux での OpenText eDirectory サービスのクラスタリング

このセクションでは、Linux の高可用性クラスタリングを使用して、OpenText eDirectory 9.3 を環境設定する方法について説明します。

- 192 ページの 「前提条件」
- 192 ページの 「OpenText eDirectory をインストールして設定する」
- 195 ページの 「クラスタ化した Linux 環境で SNMP サーバを設定する」

前提条件

- クラスタリングソフトウェアがインストールされた 2 台以上の Linux サーバ
- すべての OpenText eDirectory および NICI データを保存するための十分なディスク容量を持つ、クラスタソフトウェアがサポートしている外部共有ストレージ
- 仮想 IP アドレス
- OpenText eDirectory 9.1 以降

注 : nds-cluster-config ユーティリティは、OpenText eDirectory のルートインスタンスの環境設定だけをサポートしています。クラスタ環境内での OpenText eDirectory の複数インスタンスの環境設定と、OpenText eDirectory の非ルートインストールはサポートされていません。

OpenText eDirectory をインストールして設定する

1 プライマリクラスタノードとして使用するサーバに OpenText eDirectory をインストールして、設定を行います。インストールと環境設定の手順の詳細については、「48 ページの 「nds-install ユーティリティを使用して OpenText eDirectory コンポーネントをインストールする」」を参照してください。

注

- OpenText eDirectory を環境設定する場合、デフォルトの NCP サーバ名は、OpenText eDirectory をインストールしたコンピュータのホストサーバ名になります。ただし、クラスタ環境内では OpenText eDirectory は複数のホスト上でホストされてい

るため、デフォルト名を使用する代わりに、クラスタに対して固有の NCP サーバ名を指定してください。たとえば、プライマリクラスタノード上に OpenText eDirectory を設定した場合、NCP サーバに clusterserver という名前を指定できます。

- 環境設定処理中、OpenText eDirectory のインストールで必ず仮想 IP アドレスを設定してください。クラスタ環境では、OpenText eDirectory はシステムの IP アドレスではなく、仮想 IP アドレスだけをリスンします。

---

2 eDirectory のインストールと環境設定を実行した後、/etc/opt/novell/eDirectory/conf に格納されている nds.conf ファイルに移動します。

3 nds.conf ファイルを編集して、n4u.nds.preferred-server 設定の値をクラスタインストールの仮想 IP アドレスに設定し、ファイルを保存して閉じます。

4 ndsstat コマンドを使って、OpenText eDirectory のインストールを確認します。

OpenText eDirectory はプライマリクラスタノードで稼働している必要があります。

5 クラスタマネージャを使って、共有ファイルシステムをマウントします。

6 環境設定ユーティリティを実行する前に、次のディレクトリ内のすべてのデータをバックアップします。

- /var/opt/novell/nici
- /var/opt/novell/eDirectory/data (n4u.server.vardir)
- /var/opt/novell/eDirectory/data/dib (n4u.nds.dibdir)
- /etc/opt/novell/eDirectory/conf (n4u.server.configdir)
- /var/opt/novell/eDirectory /log

注 : デフォルト以外の場所に OpenText eDirectory をインストールする場合、ndsconfig get コマンドを使って、インストールで使われた vardir や dir パスを検索できます。nds.conf はデフォルトの場所 (/etc/opt/novell/eDirectory/conf/nds.conf) に配置する必要があります。

---

7 プライマリクラスタノードサーバで端末を開き、次のコマンドを実行して、OpenText eDirectory サービスを停止します。

```
ndsmanage stopall
```

8 端末で環境設定ユーティリティ nds-cluster-config の場所に移動します。このユーティリティは、/opt/novell/eDirectory/bin ディレクトリにあります。

9 次のコマンドを実行します。

```
nds-cluster-config -s /<sharedfilesystem>
```

ここで、*<sharedfilesystem>* は、eDirectory 共有クラスタデータに使用する場所を指しています。

注 : -u オプションを使って、ユーティリティを無人モードで実行することもできます。このオプションを使用すると、クラスタで OpenText eDirectory を設定するときに、ユーティリティによる確認は行われなくなります。

無人オプションを使用する場合、-s オプションを併せて使用し、共有クラスタファイルシステムを指定する必要があります。

10 クラスタ共有ストレージが有効であることをユーティリティが確認した後、[y] をクリックして、クラスタの環境設定を続行します。

環境設定ユーティリティは、先に示したディレクトリ内のデータを、共有ファイルシステムの次の場所に移動します。

- *<sharedfilesystem>*/nici
- *<sharedfilesystem>*/data
- *<sharedfilesystem>*/data/
- *<sharedfilesystem>*/conf
- *<sharedfilesystem>*/log

11 次のコマンドを実行して、OpenText eDirectory サービスを始動します。

```
ndsmanage startall
```

12 ndsstat を使って、OpenText eDirectory のステータスを確認します。OpenText eDirectory サービスは稼働している必要があります。

13 次のコマンドを実行して、OpenText eDirectory サービスを停止します。

```
ndsmanage stopall
```

14 クラスタのセカンダリノードとして、使用するサーバにログインします。

15 クラスタマネージャを使って、共有ストレージをセカンダリノードに移動します。

16 プライマリクラスタノードにインストールしたバージョンと同じ OpenText eDirectory のバージョンを、セカンダリクラスタノードにインストールします。ただし、OpenText eDirectory は設定しないでください。

17 端末で、セカンダリノードの環境設定ユーティリティの場所に移動します。このユーティリティは、/opt/novell/eDirectory/bin ディレクトリにあります。

18 端末を開き、次のコマンドを実行します。

```
nds-cluster-config -s /<sharedfilesystem>
```

ここで、*<sharedfilesystem>* は、クラスタ共有ストレージを指しています。*<sharedfilesystem>* のパスは、プライマリノードの設定時に指定したパスの場所と同じにしてください。

nds-cluster-config ユーティリティは、セカンダリクラスタノードを、共有クラスタファイルシステム上に配置されている共有 OpenText eDirectory データにリンクします。

19 次のコマンドを実行して、OpenText eDirectory サービスを始動します。

```
ndsmanage startall
```

ndsstat コマンドを使って、eDirectory の状態を確認します。

20 ndsmanage stopall コマンドを実行して、セカンダリノード上の OpenText eDirectory サービスを停止します。

21 クラスタの両方のノード上で OpenText eDirectory を正常に環境設定した後、次のコマンドを使って、各ノード上の NDSD サービスのスタートアップモードを変更する必要があります。

```
chkconfig -d ndsd
```

22 環境設定ユーティリティで 2 番目のノードの環境設定が終了した後、クラスタマネージャを使って、OpenText eDirectory サービスをクラスタに追加できます。

Linux でのクラスタサービスの詳細については、次のマニュアルを参照してください。

- SUSE Linux Enterprise Server (SLES 12 以降)
- SLES 11 SP4

重要 : 2 台以上のノードが同じ DIB を同時にアクセスしていないことをクラスタマネージャがチェックするのが、理想的です。ただし、ユーザが 2 台以上のクラスタノードから同時に NDSD が実行されないことを確認する必要があります。これは、2 台以上のノードを介して同じ DIB にアクセスすると、DIB の破損につながるからです。

クラスタ化した Linux 環境で SNMP サーバを設定する

1 すべてのノード上で snmpd.conf ファイルを変更します。詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*OpenText eDirectory の SNMP サービスのインストールと設定*」を参照してください。

2 ndssnmpsa を起動します。

3 ［パスワードを保存しますか］オプションに「Yes」を選択します。

4 SNMP サービスを開始するには、次のいずれかの操作を実行します。

- /etc/init.d/ndssnmpsa start を post_ndsd_start スクリプトに追加し、/etc/init.d/ndssnmpsa stop を pre_ndsd_stop スクリプトに追加します。
- eDirectory リソースに依存するクラスタリソースとして ndssnmpsa を追加します。

注 : OpenText eDirectory は仮想 IP アドレスをリスンしているため、トラップにはホストの IP アドレスが設定されます。これはエージェントの IP アドレスになります。

Windows での OpenText eDirectory サービスのクラスタリング

このセクションでは、Windows の高可用性クラスタリングを使用して、OpenText eDirectory 9.3 を環境設定する方法について説明します。

- 196 ページの 「前提条件」
- 196 ページの 「OpenText eDirectory をインストールして設定する」
- 198 ページの 「クラスタ化した Windows 環境で SNMP サーバを設定する」

前提条件

- クラスタリングソフトウェアをインストールした 2 台以上の Windows サーバ
- クラスタソフトウェアがサポートしている外部共有ストレージ
- 仮想 IP アドレス
- OpenText eDirectory 9.3

OpenText eDirectory をインストールして設定する

1 プライマリクラスタノードとして使用するサーバに OpenText eDirectory をインストールして、設定を行います。インストールと環境設定の手順の詳細については、「81 ページの 「Windows サーバで OpenText eDirectory 9.3 をインストールまたは更新する」」を参照してください。

2 クラスタマネージャを使って、共有ボリュームをマウントします。

3 環境設定ユーティリティを実行する前に、すべての DIB ファイルと NICI データをバックアップします。

4 プライマリクラスタノードで端末を開き、NDSCons.exe ユーティリティに移動します。このユーティリティは、デフォルトでは *<eDirectory のインストールフォルダ>* フォルダにあります。

5 端末で次のコマンドを実行します。

```
NDSCons.exe
```

6 NDSCons ユーティリティで［**シャットダウン**］をクリックして、すべての eDirectory サービスを停止します。

7 「**Yes**」をクリックして確認します。

8 端末で環境設定ユーティリティ dsclusterconfig.exe の場所に移動します。このユーティリティは、デフォルトでは <OpenText eDirectory installation folder> フォルダにあります。

9 次のコマンドを実行します。

```
dsclusterconfig.exe -s /<sharedfilesystem>
```

ここで、*<sharedfilesystem>* は、OpenText eDirectory 共有クラスタデータに使用する場所を指しています。

注

- -s と -u オプションを使って、無人モードでユーティリティを実行することもできます。
- プライマリクラスタノードにマウントされている共有ドライブ内のフォルダを指定する必要があります。ドライブ名だけを指定することはできません。たとえば、E: を指定するのではなく、E:\Novell を指定する必要があります。

---

10 クラスタ共有ストレージが有効であることをユーティリティが確認した後、［**y**］をクリックして、クラスタの環境設定を続行します。

環境設定ユーティリティは、先に示したディレクトリ内のデータを、共有ファイルシステムの次の場所に移動します。

- *<sharedfilesystem>*/nici
- *<sharedfilesystem>*/Files

共有ファイルシステムへの eDirectory データの移動に加え、ユーティリティは OpenText eDirectory サービスのレジストリキーを共有ボリュームにコピーし、キーを ndsConfigKey ファイルとして保存します。

また、プライマリノードコンピュータの NDS Server サービスの［起動のタイプ］を、［自動］から［手動］に変更します。

11 NDSCons ユーティリティで［**起動**］をクリックすると、すべての eDirectory サービスが起動します。

12 すべての OpenText eDirectory サービスが実行されていることを確認し、NDSCons ユーティリティを使って、サービスをもう一度停止します。

13 NDSCons ユーティリティを終了します。

14 クラスタのセカンダリノードとして、使用するサーバにログインします。

15 クラスタマネージャを使って、共有ストレージをセカンダリノードに移動します。

16 OpenText eDirectory インストーラを使って、セカンダリノードで OpenText eDirectory の無人インストールを実行します。インストールのモードが「Install」であることを確認します。

17 端末で、セカンダリノードの環境設定ユーティリティの場所に移動します。このユーティリティは、デフォルトでは OpenText eDirectory のインストールフォルダにあります。

18 次のコマンドを実行します。

```
dsclusterconfig.exe -s /<sharedfilesystem>
```

ここで、*<sharedfilesystem>* は、クラスタ共有ストレージを指しています。*<sharedfilesystem>* のパスは、プライマリノードの設定時に指定したパスの場所と同じにしてください。

19 dsclusterconfig ユーティリティはセカンダリクラスタノードのレジストリを、共有クラスタファイルシステム上に格納されている共有 OpenText eDirectory データに更新します。

20 環境設定ユーティリティによるセカンダリノードの環境設定が終了したら、NDSCons ユーティリティを実行します。

21 NDSCons ユーティリティで［**起動**］をクリックします。

22 「**Yes**」をクリックして確認します。

23 NDSCons がすべての OpenText eDirectory サービスを開始したら、OpenText eDirectory を確認し、［**シャットダウン**］をクリックします。

24 **[はい]** をクリックして確認します。

25 クラスタリソースグループで OpenText eDirectory を設定するには、OpenText eDirectory に使用するリソースグループで新しいリソースを作成します。

次の詳細情報を提供する必要があります。

- リソースタイプ - 汎用サービス
- 依存先 - リソースグループの IP アドレスと共有ディスク
- サービス名 - NDS Server0
- 起動パラメータなし
- レジストリキー -SYSTEM\CurrentControlSet\Services\NDS Server0

注 : 2 台以上のノードが同じ DIB を同時にアクセスしていないことをクラスタマネージャがチェックするのが、理想的です。ただし、ユーザが 2 台以上のクラスタノードから同時に NDSD が実行されないことを確認する必要があります。これは、2 台以上のノードを介して同じ DIB にアクセスすると、DIB の破損につながるからです。

クラスタ化した Windows 環境で SNMP サーバを設定する

1. プライマリクラスタノードで、マスタエージェントを設定し、自動化する起動タイプを設定します。詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*OpenText eDirectory の SNMP サービスのインストールと設定*」を参照してください。
2. パスワードの入力を求めるプロンプトが表示されたら、OpenText eDirectory パスワードを保存します。
3. サブエージェントを起動します。
4. 他のノードでステップ 1 ～ステップ 3 を実行します。

クラスタ化環境のトラブルシューティング

クラスタ化ノードの OpenText eDirectory を修復またはアップグレードする

任意のクラスタノードで修復またはアップグレードを実行した場合、自動フェールオーバーが発生しないように、他のクラスタノードを一時停止またはスタンバイ状態にする必要があります。

Windows レジストリキーの作成

Windows のクラスタ環境での環境設定処理の一部として、環境設定ユーティリティは、クラスタ共有ファイルシステムでレジストリキー HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDS Server0\ImagePath を自動作成します。eDirectory はクラスタノード上で x86 NDS Server サービスを開始するため、このレジストリキーを必要とします。

ユーティリティでこのレジストリキーが作成できず、環境設定中にエラーメッセージが返された場合は、完了設定ユーティリティが環境設定を正常に完了したように見える場合でも、レジストリエディタを使って、すべてのクラスタノードでレジストリキーを手動作成する必要があります。

すべてのノードで次のレジストリキーを作成します。

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDS Server0\ImagePath
```

ImagePath キーに次の値を割り当てます。

```
"<primarynodeinstallfolder>\NDS\ndsserv.exe" /
DataDir="<sharedstorage>\Files" ds
```

ここで、*<primarynodeinstallfolder>* は、プライマリノードに eDirectory をインストールしたフォルダで、*<sharedstorage>* は、共有ファイルシステムの場所へのパスです。

環境設定ユーティリティのオプション

環境設定ユーティリティで使用可能なオプションを次に示します。

```
<configuration utility> [-h] [-u] [-s /<sharedfilesystem>]
```

ここで、*<configuration utility>* は、プラットフォームによって nds-cluster-config か dsclusterconfig.exe のいずれかになります。*<sharedfilesystem>* は、OpenText eDirectory 共有クラスタデータに使用する場所を指しています。

パラメータ	説明
-h	環境設定ユーティリティのヘルプを表示します。
-s	クラスタの共有ディレクトリパスを指定します。
-u	ユーティリティがクラスタ上の OpenText eDirectory の環境設定を無人モードで実行できるようにします。 -u オプションを使ってユーティリティを実行した場合、-s オプションも併せて使用し、共有ディレクトリパスを指定してください。次に例を示します。 `nds-cluster-config -u -s <sharedfilesystem>`

14 OpenText eDirectory のアンインストール

この章では、次の情報について説明します。

- 201 ページの 「Windows での OpenText eDirectory のアンインストール」
- 206 ページの 「Linux 上での OpenText eDirectory のアンインストール」
- 207 ページの 「Linux 上での OpenText eDirectory の無人アンインストール」
- 208 ページの 「OpenText eDirectory のアンインストールに関する注意」

Windows での OpenText eDirectory のアンインストール

Windows の［コントロールパネル］を使用して、Windows サーバから OpenText eDirectory、ConsoleOne、SLP DA、および NICI を削除します。

重要 : OpenText eDirectory を削除すると、ロールフォワードログディレクトリおよびそれに含まれるすべてのログも削除されます。このサーバの OpenText eDirectory の復元にログを使用する予定がある場合は、OpenText eDirectory を削除する前にロールフォワードログを別の場所にコピーする必要があります。ロールフォワードログの詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*ロールフォワードログを使用する*」を参照してください。

- 201 ページの 「OpenText eDirectory、ConsoleOne、および SLP DA のアンインストール」
- 202 ページの 「OpenText eDirectory の無人アンインストール」
- 206 ページの 「NICI のアンインストール」
- 206 ページの「Microsoft Visual C++ 2005 と Visual C++ 2012 のランタイムライブラリのアンインストール」

注 : iMonitor を使用して作成した HTML ファイルは削除されません。iMonitor を使って作成した HTML ファイルは、eDirectory を削除する前に、*< インストールディレクトリ* >\novell\NDS\ndsimon\dsreports から手動で削除してください。

OpenText eDirectory、ConsoleOne、および SLP DA のアンインストール

1 OpenText eDirectory がインストールされている Windows サーバで、［**スタート**］>［**設定**］>［**コントロールパネル**］>［**プログラムの追加と削除**］の順に選択します。

2 リストから［**OpenText eDirectory**］、［**ConsoleOne**］、または［**SLP ディレクトリエージェント**］を選択し、［**追加と削除**］をクリックします。

3 選択したアプリケーションの削除を確認するメッセージが表示されたら、「Yes」をクリックします。

インストールウィザードによって、該当するプログラムがサーバから削除されます。

OpenText eDirectory の無人アンインストール

Windows での OpenText eDirectory の無人アンインストールでは、無人アンインストールを円滑に実行できるように、事前定義されたテキストファイルが使用されます。OpenText eDirectory の無人アンインストールモードを使うことで、以下の操作を実行できます。

- インストールした eDirectory の環境設定の解除。
- OpenText eDirectory のスタンドアロンアンインストール。
- eDirectory のアンインストールと設定解除の両方。

次のセクションでは、OpenText eDirectory の無人アンインストールのさまざまな機能について説明します。

- 202 ページの 「レスポンスファイル」
- 203 ページの 「remove.rsp ファイルのセクションとキー」
- 203 ページの 「自動アンインストールに機能を追加する」
- 204 ページの 「環境設定ファイルの変更の削除」
- 205 ページの「レスポンスファイルを使った OpenText eDirectory の無人アンインストール」

レスポンスファイル

Windows オペレーティングシステムでの OpenText eDirectory のアンインストールは、レスポンスファイル (remove.rsp) を使って以下のタスクを実行することで、サイレントでかつ柔軟に実行することができます。

- 必要なすべてのユーザ入力が用意された完全無人アンインストール
- コンポーネントのデフォルト設定
- インストール中のすべてのプロンプトのバイパス

レスポンスファイルとは、Windows.ini ファイルのようなセクションとキーが記述されているテキストファイルです。任意の ASCII テキストエディタを使って、レスポンスファイルの作成と編集ができます。OpenText eDirectory はレスポンスファイルから直接アンインストールパラメータを読み取り、デフォルトのアンインストール値をレスポンスファイルの値に置き換えます。アンインストールプログラムは、レスポンスファイルからの値を受け入れ、プロンプトを表示せずにアンインストールを続行します。

remove.rsp ファイルのセクションとキー

OpenText eDirectory のアンインストールでは、レスポンスファイル内のセクションを変更し、ツリー名、管理コンテキスト、管理者資格情報 (ユーザ名やパスワード) などの情報を追加する必要があります。キーとそのデフォルト値の全リストは、OpenText eDirectory のインストール時に提供されるサンプルの remove.rsp ファイルから入手できます。

注 : OpenText eDirectory インストール内の eDirectory\windows\x64\NDSonNT\remove.rsp にある remove.rsp ファイルを使用する必要があります。必須パラメータはデフォルトで、このファイルで設定されます。remove.rsp ファイルを編集するとき、キーと値のペアを結ぶ等号記号 (「=」) の前後にスペースが入らないようにしてください。

無人アンインストールで使用する remove.rsp ファイルに管理者ユーザ資格情報を入力します。このため、管理者資格情報が漏洩しないように、アンインストール後はファイルを完全に削除してください。

自動アンインストールに機能を追加する

OpenText eDirectory アンインストーラの細かな設定はほとんど、手動アンインストールのデフォルト設定になっています。ただし、無人アンインストール中、各環境設定パラメータを明示的に設定する必要があります。このセクションでは、設定を解除する基本設定について説明します。

OpenText eDirectory サーバの詳細情報

アンインストールするサーバの詳細を、アンインストーラに提供してください。この情報のほとんどは、3 つのタグ [Novell:NDSforNT:1.0.0]、[Initialization]、および [Selected Nodes] で設定されます。

remove.rsp の [Initialization] および [Selected Nodes] で指定されているすべての値をそのまま使用してください。

[Novell:NDSforNT:1.0.0]

Tree Name: サーバがアンインストールされるツリーの名前。

Admin Login Name: 少なくともサーバの追加先のコンテキストに対してフル権限を持つ、ツリー内の管理者オブジェクトの名前 (RDN)。ツリー内のすべての操作は、このユーザとして実行されます。

Admin Context: ツリーに追加されたユーザにはユーザオブジェクトがあり、そこにユーザ固有の詳細情報がすべて入っています。このパラメータは、管理者オブジェクトの追加先となるツリーのコンテナオブジェクトです。プライマリサーバのインストールでは、このコンテナはサーバオブジェクトと共に作成されます。

Admin Password: 前述のパラメータで作成された管理者オブジェクトのパスワード。このパスワードは、プライマリサーバのインストール時に管理者オブジェクトに対して設定されます。セカンダリサーバのインストールでは、これは新しいサーバの追加先となるコンテキストに対して権限を持っているプライマリサーバの管理者オブジェクトのパスワードである必要があります。

NDS Location: ライブラリとバイナリがコピーされる、ローカルシステムの OpenText eDirectory のインストール場所。レスポンスファイルで変更されていない限り、OpenText eDirectory はデフォルトで C:\Novell\NDS にインストールされます。

DataDir: OpenText eDirectory 9.2 までは、DIB は NDS ロケーション内にサブフォルダとしてインストールされていました。後に、管理者が別の DIB ロケーションを指定するオプションが追加されました。これは、DIB に保存されるデータが多くなりすぎて NDS ロケーションに収まらなくなる可能性があるためです。現在デフォルトでは、DIB は NDS ロケーション内の Files サブフォルダにインストールされますが、管理者はこのパラメータを変更して別の場所を指定することができます。

mode: OpenText eDirectory のセットアップのタイプ。セットアップには、次の 3 つのタイプがあります。

- deconfigure: OpenText eDirectory の環境設定解除を実行します。
- uninstall: OpenText eDirectory のアンインストールを実行します。
- full: OpenText eDirectory の設定解除とアンインストールの両方を実行します。

注 : 無人インストール時にフルセットアップモードを選択すると、OpenText eDirectory のアンインストール時に、個別の設定解除とアンインストールオプションを選択することはできません。

ConfigurationMode: mode キーで指定したセットアップを設定解除した場合、ConfigurationMode キーの RestrictNodeRemove 値を変更していないことを確認してください。

Prompt: アンインストールモードのタイプは、この変数で指定する必要があります。無人アンインストールの場合、デフォルトでは「silent」に設定されます。「silent」以外の値を設定すると、通常のアンインストールが実行されます。

前述の基本パラメータをすべて記述したレスポンスファイルのテキスト例を次に示します。

```
[Novell:NDSforNT:1.0.0]

  Tree Name=SILENTCORP-TREE

  Admin Context=Novell

  Admin Login Name=Admin

  Admin Password=novell

  prompt=silent
```

環境設定ファイルの変更の削除

<Windows インストールドライブ >\Program Files\Common Files\novell\ni\bin *にある* remove.cfg ファイルを次のように変更します。

```
[PARAMETERS]0/OUTPUT_TO_FILE
```

変更後 :

```
[PARAMETERS]0/OUTPUT_TO_FILE /SILENT
```

レスポンスファイルを使った OpenText eDirectory の無人アンインストール

編集したファイル remove.rsp を *<Windows インストールドライブ>*\Program Files\Common Files\novell\ni\data にコピーします。

eDirectory にインストールされた install.exe は、追加パラメータを指定して、コマンドラインから起動します。必要なセットアップに応じて、次のいずれかのコマンドを実行する必要があります。

設定解除

```
<Windows Installed Drive>\Program Files\Common
Files\novell\ni\bin>install.exe -remove /restrictnoderemove /nopleasewait
..\data\ip.db ..\data\remove.rsp Novell:NDSForNT:1.0.0 0 NDSonNT
```

アンインストール

1. <Windows ドライブ >\Program Files\Common Files\novell\ni\data *ディレクトリ内にある* ip.db ファイルの名前を別の名前に変更します。
2. <Windows ドライブ >\Program Files\Common Files\novell\ni\data *フォルダの* ip_conf.db ファイルを、ip.db にコピーします。
3. 次のコマンドを実行します。

   <Windows がインストールされているドライブ>\Program Files\Common Files\novell\ni\bin>install.exe -remove /nopleasewait ..\data\ip.db ..\data\remove.rsp Novell:NDSForNT:1.0.0 0 NDSonNT

OpenText eDirectory の設定解除とアンインストール

```
<Windows Installed Drive>\Program Files\Common
Files\novell\ni\bin>install.exe -remove /nopleasewait ..\data\ip.db
..\data\remove.rsp Novell:NDSForNT:1.0.0 0 NDSonNT
```

OpenText eDirectory またはそれに伴うセットアップのアンインストールを実行したら、次のフォルダを削除します。

- C:\Novell\NDS(デフォルトの場所、または OpenText eDirectory のインストールディレクトリ)
- C:\Novell\NDS\Files (デフォルトの場所、または OpenText eDirectory DIB の格納先)
- *<Windows がインストールされているドライブ>*:\Program Files\Common Files\Novell\ni
- *<Windows がインストールされているドライブ>*:\Windows\system32\NDScpa.cpl

NICI のアンインストール

1 OpenText eDirectory がインストールされている Windows サーバで、［**スタート**］>［**設定**］>［**コントロールパネル**］>［**プログラムの追加と削除**］の順に選択します。

2 リストから［**NICI**］を選択し、［**追加と削除**］をクリックします。

3 NICI の削除を確認するメッセージが表示されたら、［**Yes**］をクリックします。

インストールウィザードによって、サーバから NICI が削除されます。

NICI をアンインストールした後、システムから NICI を完全に削除するには、C:\Windows\system32\novell\nici (32 ビット) と C:\Windows\SysWOW64\novell\nici (64 ビット) サブディレクトリを削除します。これらを削除するには、いくつかのファイルおよびディレクトリの所有権を持っている必要がある場合があります。

警告 : NICI サブディレクトリの削除後は、以前に NICI で暗号かされたすべてのデータまたは情報が失われます。

Microsoft Visual C++ 2005 と Visual C++ 2012 のランタイムライブラリのアンインストール

Microsoft Visual C++ 2005 と Microsoft Visual C++ 2012 のランタイムライブラリが OpenText eDirectory 以外の製品で使用されていない場合は、次の手順を使用してそれらのランライムライブラリをアンインストールしてください。

1 eDirectory がインストールされている Windows サーバの［*プログラムの追加と削除*］または［**プログラムと機能**］に移動します。

2 次の再配布パッケージを削除します。

Microsoft Visual C++ 2012 再配布可能モジュールと Microsoft Visual C++ 2005 再配布可能パッケージ (x64)

Linux 上での OpenText eDirectory のアンインストール

Linux コンピュータから OpenText eDirectory コンポーネントをアンインストールするには、nds-uninstall ユーティリティを使用します。このユーティリティはローカルホストから eDirectory をアンインストールします。nds-uninstall を実行する前に、OpenText eDirectory サーバの設定を解除してください。ndsconfig rm -a <*admin FDN*> を実行して、eDirectory サーバを削除します。このユーティリティは、/opt/novell/eDirectory/sbin/nds-uninstall で入手できます。

ndsconfig rm を OES サーバ上で実行しないでください。

重要 : OpenText eDirectory を削除すると、ロールフォワードログディレクトリおよびそれに含まれるすべてのログも削除されます。このサーバの eDirectory の復元にログを使用する予定がある場合は、OpenText eDirectory を削除する前にロールフォワードログを別の場所

にコピーする必要があります。ロールフォワードログの詳細については、『「OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド」』の「*ロールフォワードログを使用する*」を参照してください。

1 nds-uninstall コマンドを実行します。

2 使用する構文は次のとおりです。

```
nds-uninstall [-s][-h]
```

コマンドラインに必要なパラメータを入力していない場合、パラメータを要求するプロンプトが nds-install ユーティリティに表示されます。

パラメータ	説明
-h	ヘルプを表示します。
-s	インスタンスが設定されている場合でも、OpenText eDirectory パッケージとバイナリを削除します。ただし、このオプションでは DIB ディレクトリおよび NDS 環境設定ファイルは削除されません。 **重要**: このオプションを使用した場合、他のサービスに長時間影響を与えないことを確認してください。

nds-uninstall では、次のパッケージはアンインストールされません。

パッケージ	削除されない理由
NICI パッケージ	NICI は次のいずれかで使用されている可能性があります。 ◆ その他の製品 ◆ 任意の場所にインストールされた OpenText eDirectory ◆ 非ルートユーザによってインストールされた OpenText eDirectory
NOVLsubag	NOVLsubag は次のいずれかで使用されている可能性があります。 ◆ 任意の場所にインストールされた OpenText eDirectory ◆ 非ルートユーザによってインストールされた OpenText eDirectory

Linux 上での OpenText eDirectory の無人アンインストール

1 OpenText eDirectory のインスタンスを削除します。

```
ndsconfig rm -a <user name> -w passwd -c
```

2 OpenText eDirectory の設定を解除するために、自動化スクリプトで次のどちらかを使用します。

環境変数からパスワードを渡す場合: ndsconfig rm -a < *ユーザ名* > -w env:< *環境変数* > -c

ファイルからパスワードを渡す場合 : ndsconfig rm -a < *ユーザ名* > -w file:< *絶対パス / 相対パスで指定したファイル名* > -c

3 (オプション) 複数インスタンスの場合、インスタンスごとに次のコマンドを実行します。

```
ndsconfig rm -a <user name> -w passwd  --config-file <absolute path for configuration file>
```

次に例を示します。

```
ndsconfig rm -a admin.novell -w n -c
```

```
ndsconfig rm -a admin.novell -w env:ADM_PASWD -c
```

```
ndsconfig rm -a admin.novell -w file:/Builds/88SP8/adm_paswd -c
```

4 OpenText eDirectory パッケージをアンインストールするには、nds-uninstall スクリプトを実行して、OpenText eDirectory パッケージを削除します。

```
nds-uninstall -u
```

OpenText eDirectory のアンインストールに関する注意

OpenText eDirectory をアンインストールし、再インストールした場合、OpenText eDirectory サーバからネットワーク内の他のサーバにアクセスすることはできません。レプリカが OpenText eDirectory サーバ内に存在するパーティションに対して、同期化や破損通知など、分散型操作を実行することはできません。この状態がしばらく続く場合、すべてのサーバやサーバ上で実行されている処理が影響を受けることがあります。

OpenText eDirectory の新しいバージョンをアンインストールして、以前のバージョンをインストールすることはしないでください。これには次の理由があります。

- スキーマ関連のアップグレードは元に戻りません。
- DIB が新しいバージョンにアップグレードされている場合、OpenText eDirectory が機能しないことがあります。
- nds.conf を除き、既存の環境設定ファイルがすべて削除されます。

ただし、OpenText eDirectory の新しいバージョンをアンインストールし、以前のバージョンをインストールする場合、以下の点を考慮してください。

- DIB を新しいバージョンにアップグレードします。DIB をアップグレードしないと、eDirectory は機能しないことがあります。
- nds.conf を除く既存の環境設定ファイルをバックアップし、OpenText eDirectory を再インストールしたときに復元します。
- スキーマ関連のアップグレードは元に戻りません。

A OpenText eDirectory 用の Linux パッケージ

OpenText eDirectory には、Linux パッケージシステムが含まれています。このパッケージは、さまざまな OpenText eDirectory コンポーネントのインストールとアンインストールを容易にするツールをまとめたものです。パッケージには、特定の OpenText eDirectory コンポーネントの構築に必要な条件を示した Makefile が含まれています。パッケージには、OS と一緒にインストールされた標準の Linux ツールで使用される環境設定ファイル、ユーティリティ、ライブラリ、デーモン、およびマニュアルページも含まれています。

下の表に、OpenText eDirectory に含まれている Linux パッケージに関する情報を示します。

注 : Linux では、すべてのパッケージの先頭に *novell-* というプレフィックスが追加されます (**eba** を除く)。たとえば、NDSserv は novell-NDSserv となります。

パッケージ	説明
NOVLice	NetIQ インポート / エクスポート変換ユーティリティが含まれています。NOVLlmgnt、NOVLxis、および NLDAPbase の各パッケージに依存します。
NDSbase	ディレクトリユーザエージェントを表します。このパッケージは NICI パッケージに依存しています。 NDSbase には、以下のものが含まれています。 ◆ eDirectory に必要な RSA 認証を格納する認証ツールボックス ◆ プラットフォーム独立システム抽象ライブラリ、すべての定義済みディレクトリユーザエージェント機能を格納するライブラリ、およびスキーマ拡張ライブラリ ◆ 統合設定ユーティリティおよびディレクトリユーザエージェントテストユーティリティ ◆ OpenText eDirectory 環境設定ファイルおよびマニュアルページ
NDScommon	OpenText eDirectory 環境設定ファイルのマニュアルページと、インストールおよびアンインストールユーティリティが含まれています。このパッケージは DNSbase パッケージに依存しています。
NDSmasv	必須アクセスコントロールサービス (MASV) に必要なライブラリが含まれています。

パッケージ	説明
NDSserv	OpenText eDirectory サーバが必要とするすべてのバイナリとライブラリが含まれています。また、システム上の OpenText eDirectory サーバを管理するユーティリティも含まれています。このパッケージは NDSbase、NDScommon、NDSmasv、NLDAPsdk、NOVLpkia、および NOVLpkit の各パッケージに依存しています。 NDSserv パッケージには次のものが含まれています。 ◆ NDS インストールライブラリ、FLAIM ライブラリ、トレースライブラリ、NDS ライブラリ、LDAP サーバライブラリ、LDAP インストールライブラリ、インデックスエディタライブラリ、DNS ライブラリ、マージライブラリ、および LDAP SDK 用 LDAP 拡張ライブラリ ◆ OpenText eDirectory サーバのデーモン ◆ DNS 用バイナリ、および LDAP のロード / アンロード用バイナリ ◆ MAC アドレスの作成に必要なユーティリティ、サーバの追跡およびサーバの一部のグローバル変数の変更用ユーティリティ、OpenText eDirectory のバックアップと復元用ユーティリティ、および OpenText eDirectory ツリーのマージユーティリティ ◆ DNS、NDSD、および NLDAP の起動スクリプト ◆ マニュアルページ
NDSimon	OpenText eDirectory サービスからのデータの検索および取得に使用される、ランタイムライブラリおよびユーティリティが含まれています。このパッケージは DNSbase パッケージに依存しています。
NDSrepair	OpenText eDirectory データベースの問題を修正するランタイムライブラリおよびユーティリティが含まれています。このパッケージは DNSbase パッケージに依存しています。
NLDAPbase	LDAP ライブラリ、LDAP ライブラリの拡張、および次の LDAP ツールが含まれています。 ◆ ldapdelete ◆ ldapmodify ◆ ldapmodrdn ◆ ldapsearch このパッケージは NLDAPsdk パッケージに依存しています。
NOVLnmas	NMAS サーバが必要とする、すべての NMAS ライブラリと nmasinst バイナリが含まれています。このパッケージは、NICI および NDSmasv のパッケージに依存しています。
NLDAPsdk	LDAP ランタイムとセキュリティライブラリ (Client NICI) に対する OpenText 拡張が含まれています。
NOVLsubag	OpenText eDirectory SNMP サブエージェント用のランタイムライブラリおよびユーティリティが含まれています。このパッケージは、NICI、NDSbase、および NLDAPbase の各パッケージに依存しています。

パッケージ	説明
NOVLpkit	OpenText eDirectory を必要としない PKI サービスを提供します。このパッケージは、NICI と NLDAPsdk のパッケージに依存しています。
NOVLpkis	PKI サーバサービスを提供します。このパッケージは、NICI、NDSbase、および NLDAPsdk の各パッケージに依存しています。
NOVLsnmp	SNMP 用のランタイムライブラリおよびユーティリティです。このパッケージは NICI パッケージに依存しています。
NDSdexvnt	OpenText eDirectory から他のデータベースに生成されたイベントを管理するライブラリが含まれています。
NOVLpkia	PKI サービスを提供します。このパッケージは、NICI、NDSbase、および NLDAPsdk の各パッケージに依存しています。
NOVLembox	eMBox インフラストラクチャおよび eMTool を提供します。
NOVLlmgnt	OpenText Language Management のランタイムライブラリが含まれています。
NOVLxis	NetIQ XIS 用ランタイムライブラリが含まれています。
NOVLsas	OpenText SAS ライブラリが含まれています。
NOVLntls	OpenText TLS ライブラリが含まれています。 Linux 上では、このパッケージは ntls です。
NOVLldif2	OpenText Offline Bulkload ユーティリティが含まれています。NDSbase、NDSserv、NOVLntls、NOVLlmgnt、および NICI の各パッケージに依存しています。
NOVLncp	OpenText Encrypted NCP Services for Linux が含まれています。このパッケージは NDScommon パッケージに依存します。
novell-eba	拡張バックグラウンド認証をサポートするためのライブラリが含まれています。このパッケージは、NICI、NDSbase、および NDSServ の各パッケージに依存しています。

B OpenText eDirectory ヘルスチェック

OpenText eDirectory 9.3 は、使用している eDirectory のヘルスが安全かどうかを特定できるようにする診断ツールを提供します。このツールの主な用途は、アップグレード前にサーバの状態をチェックすることです。

OpenText ヘルスチェックは、デフォルトでアップグレードの実行時に必ず実行されます。実際のパッケージアップグレードが開始される前に実行されます。診断ツール ndscheck は、ヘルスチェックを確認するために、いつでも実行できます。

ヘルスチェックの必要性

OpenText eDirectory の以前のリリースでは、アップグレードを進める前にサーバの状態はチェックされませんでした。状態が不安定な場合は、アップグレード処理に失敗し、OpenText eDirectory は不整合な状態になっていました。場合によっては、アップグレード前の設定に戻すことができない場合もあります。

新しいヘルスチェックツールによってこの問題が解決され、サーバをアップグレードする準備を確実に整えることができます。

ヘルスチェックの実行

OpenText eDirectory ヘルスチェックは次の 2 とおりの方法で実行できます。

注 : ヘルスチェックユーティリティを実行するには、管理者の権利を持っている必要があります。

- 213 ページの 「アップグレードと同時に実行」
- 214 ページの 「スタンドアロンユーティリティとして実行」

アップグレードと同時に実行

OpenText eDirectory をアップグレードするときは常に、デフォルトでヘルスチェックが実行されます。

Linux

アップグレード時には常にデフォルトで、実際のアップグレード処理が開始される前にヘルスチェックが実行されます。

デフォルトのヘルスチェックをスキップするには、-j オプションを指定して nds-install を実行します。

Windows

OpenText eDirectory ヘルスチェックは、インストールウィザードの一部として実行されます。ヘルスチェックは、プロンプトが表示されたときに有効または無効にすることができます。

スタンドアロンユーティリティとして実行

OpenText eDirectory ヘルスチェックはいつでも好きなときにスタンドアロンユーティリティとして実行できます。次の表に、プラットフォームごとのヘルスチェックユーティリティ名を示します。

***表 B-1** ヘルスチェックユーティリティ*

プラットフォーム	ユーティリティ名

Linux	ndscheck
	
	構文 :
	
	`ndscheck [--help | -?] Display command usage`
	`ndscheck [--version | -v] Display version information ndscheck [-h <hostname port]>] [-a <admin FDN>] [-F <log file>] [-D] [-q] [--config-file <file name>]`

Windows	ndscheck
	
	構文 :
	
	`ndscheck [--help | -?] Display command usage`
	`ndscheck [--version | -v] Display version information ndscheck [-h <hostname port]>] [-a <admin FDN>] [-F <log file>] [-D] [-q] [--config-file <file name>]`

ヘルスチェックのタイプ

ndscheck ユーティリティを実行またはアップグレードすると、次のタイプのヘルスチェックが実行されます。

- 基本的なサーバの状態
- パーティションとレプリカの状態

ndscheck ユーティリティを実行すると、その結果が画面に表示され、ndscheck.log に記録されます。ログファイルの詳細については、「216 ページの 「ログファイル」」を参照してください。

アップグレードの一部としてヘルスチェックを実行した場合、エラーが検出されると、検出されたエラーのタイプに応じて、アップグレードを続行するかどうかの確認が求められるか、または処理が中断されます。エラーの種類については、「215 ページの 「状態のカテゴリ」」で説明します。

基本的なサーバの状態

これは、ヘルスチェックの最初の段階です。ヘルスチェックユーティリティは次の内容をチェックします。

1. OpenText eDirectory サービスが動作している。DIB が開いていて、ツリー名などの基本的なツリー情報を読むことができる。
2. サーバがそれぞれのポート番号を監視している。

   LDAP に関しては、TCP ポート番号と SSL ポート番号を取得して、サーバがこれらのポートを監視しているかどうかをチェックします。

   同様に、HTTP セキュアポート番号と HTTPS セキュアポート番号を取得して、サーバがこれらのポートを監視しているかどうかをチェックします。

パーティションとレプリカの状態

基本的なサーバの状態のチェック後は、次のとおり、パーティションとレプリカの状態をチェックします。

1. ローカルに保持されているパーティションのレプリカの状態をチェックします。
2. サーバによって保持されているすべてのパーティションのレプリカリングを読み込み、レプリカリング内のすべてのサーバが動作していて、すべてのレプリカが使用可能な状態であることをチェックします。
3. レプリカリング内のすべてのサーバの時刻同期をチェックし、サーバ間の時刻の違いを表示します。

状態のカテゴリ

OpenText eDirectory サーバのヘルスチェック中に検出されたエラーに応じて、ヘルスには 3 つの潜在的なカテゴリがあります。

- (215 ページ) 正常
- (216 ページ) 警告
- (216 ページ) 重大

ヘルスチェックの状態は、ログファイルに記録されます。詳細については、「216 ページの「ログファイル」」を参照してください。

正常

すべてのヘルスチェックに成功し、サーバの状態は正常です。

アップグレードは中断されずに続行されます。

警告

サーバの状態のチェック中に、あまり重大でないエラーが検出されました。

アップグレードの一部としてヘルスチェックが実行されている場合、中止するか続行するかの確認を求められます。

警告は通常、次の状況で発生します。

- サーバが LDAP ポートと HTTP ポート (通常、セキュリティ保護、または両方) を監視していない。
- レプリカリング内のいずれの非マスタサーバにも接続できない。
- レプリカリング内のサーバが同期していない。

重大

OpenText eDirectory のヘルスチェック中に重大なエラーが検出されました。

ヘルスチェックが OpenText eDirectory アップグレードの一部として実行されている場合、アップグレード操作は中止されます。

重大な状態は通常、次の状況で発生します。

- DIB を開くことができないか読み込むことができない (ロックされているか破損している可能性がある)。
- レプリカリング内のすべてのサーバに接続できない。
- ローカルに保持されているパーティションが使用中である。
- レプリカが使用可能な状態ではない。

ログファイル

アップグレード時に実行するか、スタンドアロンユーティリティとして実行するかにかかわらず、OpenText eDirectory ヘルスチェック操作を実行するたびに、ヘルスのステータスがログファイルに記録されます。

ログファイルの内容は、チェック実行時に画面に表示されるメッセージと同様です。

ヘルスチェックのログファイルには、次のものが含まれています。

- ヘルスチェックのステータス (正常、警告、または重大)。
- 考えられる解決方法を示す URL。
  - サポートフォーラム (http://forums.novell.com/netiq/netiq-product-discussion-forums/edirectory/)
  - ドキュメントのトラブルシューティング (https://www.netiq.com/documentation/edir88/edir88tshoot/data/bookinfo.html)
  - エラー番号 (http://www.novell.com/documentation/nwec/)

- パッチ (http://support.novell.com/patches.html)
- Cool Solutions (http://www.novell.com/communities/coolsolutions/edirectory)

次の表は、さまざまなプラットフォームにおけるデフォルトのログファイルの位置を示しています。

***表 B-2** ヘルスチェックのログファイルの場所*

プラットフォーム	ログファイル名	場所
Linux	ndscheck.log	1. -h オプションを指定すると、ndscheck.log ファイルはユーザのホームディレクトリに保存されます。 2. --config-file オプションを指定すると、ndscheck.log ファイルはサーバインスタンスのログディレクトリに保存されます。または、インスタンスの一覧からインスタンスを選択することもできます。
Windows	nsdcheck.log	ログファイルは *install_directory*\novell nds\ に保存されます。 **注 :** *install_directory* は、ユーザが指定するディレクトリです。

C OpenSLP for OpenText eDirectoryの設定

この付録では、ネットワーク管理者向けに、Novell Client なしで OpenSLP for OpenText eDirectory をインストールための適切な環境設定について説明します。

- 219 ページの 「Service Location Protocol」
- 219 ページの 「SLP の基本」
- 222 ページの 「環境設定パラメータ」

Service Location Protocol

OpenSLP は、IETF Service Location Protocol Version 2.0 規格のオープンソースによる実装です。この規格は、IETF Request-For-Comments (RFC) 2608 (http://www.ietf.org/rfc/rfc2608.txt?number=2608) で文書化されました。

SLP v2 プロトコルの実装に加え、OpenSLP ソースコードが提供するインタフェースは、SLP 機能にプログラム的にアクセスするためのもう 1 つの IETF 規格を実装したもので、RFC 2614 (http://www.ietf.org/rfc/rfc2614.txt?number=2614) で文書化されています。

SLP の動作を完全に理解するため、前述の文書を読んで修得することをお勧めします。読みやすい文書ではありませんが、インターネットでの SLP の正しい設定を行うためには重要なドキュメントです。

OpenSLP プロジェクトの詳細については、OpenSLP (http://www.OpenSLP.org) の Web サイトと SourceForge (http://sourceforge.net/projects/openslp) の Web サイトを参照してください。OpenSLP の Web サイトには、環境設定に関する貴重なヒントを含んださまざまな文書があります。ただし、このガイドの作成時点では、これらのドキュメントの多くは未完成です。

SLP の基本

Service Location Protocol では、次の 3 種類のコンポーネントが定義されています。

- ユーザエージェント (UA)
- サービスエージェント (SA)
- ディレクトリエージェント (DA)

ユーザエージェントは、クライアントがサービスを問い合わせたり、サービスがそれ自体を通知するためのプログラムインタフェースを提供します。ユーザエージェントはディレクトリエージェントに接続し、指定したスコープ内の指定したサービスクラスに登録されたサービスを問い合わせます。

サービスエージェントは、SLP で登録されたローカルサービスを持続的に格納し、維持する場所を提供します。サービスエージェントは主として、登録済みのローカルサービスをメモリ内データベースとして維持します。この場合、サービスはローカル SA がない限り SLP で登録できません。クライアントがサービスを検出するのは UA ライブラリ内のみですが、登録するには SA が必要です。これは主に、ディレクトリエージェントを受信して登録を維持するためには、登録済みサービスの存在を SA が定期的に表明する必要があるためです。

ディレクトリエージェントは、通知されたサービスに対して長期間持続的にキャッシュを提供し、ユーザエージェントがサービスを検索するためのアクセスポイントとなります。キャッシュ機能を提供する DA は、SA が新しいサービスを通知するのを受信し、これらの通知をキャッシュします。DA のキャッシュは短時間で完了します。ディレクトリエージェントは、期限切れのアルゴリズムを使用してエントリキャッシュを有効期限切れにします。ディレクトリエージェントが起動すると、持続的な格納領域 (通常はハードドライブ) からキャッシュを読み込み、アルゴリズムに従ってエントリを有効期限切れにします。新しい DA が起動したり、キャッシュが削除されると、DA はこの条件を検出して受信中のすべての SA に特別な通知を送信します。SA は、DA が直ちにキャッシュを作成できるようにローカルデータベースをダンプします。

ディレクトリエージェントが存在しない場合、UA は SA が応答できる一般的なマルチキャスト方式のクエリを使用し、DA がキャッシュを作成するのとほぼ同じ方法で、要求されたサービスのリストを作成します。このクエリによって返されるサービスのリストは、DA が提供するリストと比較すると不完全かつ局所的です。特に、多くのネットワーク管理者が使用するマルチキャスト方式でのフィルタ処理では、ブロードキャストおよびマルチキャストの対象がローカルサブネットのみに制限されるためです。

つまり、指定されたスコープに対してユーザエージェントが検索するものは、すべてディレクトリエージェントに依存します。

OpenText Service Location Providers

OpenText バージョンの SLP では、強力なサービスアドバタイジング環境を提供するため、SLP 標準が一部変更されます。しかし、このために一部のスケーラビリティを犠牲にしています。

たとえば、サービスアドバータイズのフレームワークの拡張性を改善するために、サブネット上でのブロードキャストまたはマルチキャストのパケット数が制限されます。SLP の仕様では、これを管理するために、ディレクトリエージェントのクエリに関してサービスエージェントおよびユーザエージェントに制限を加えています。必要なスコープに対応するための最初に検出されたディレクトリエージェントは、サービスエージェント (つまり結果的にローカルユーザエージェント) がそのスコープ上の将来の要求すべてに使用するエージェントとなります。

OpenText SLP を実装すると、クエリ情報の検索について既知のディレクトリエージェントがすべてスキャンされます。スキャンの所要時間は 300 ミリ秒とかなり長く、したがって、約 3 ～ 5 秒以内で 10 台のサーバしかスキャンできません。SLP がネットワーク上で正しく設定されている場合にはこのような検索の必要はありません。OpenSLP では、ネットワークが実際に SLP トラフィック用に設定されていると見なされます。OpenSLP の応答タ

イムアウト値は OpenText SLP サービスプロバイダの応答タイムアウト値よりも大きい値です。ディレクトリエージェント数は、エージェントの情報が正確で完全であるかどうかに関係なく、最初に応答するディレクトリエージェントに制限されます。

ユーザエージェント

ユーザエージェントの物理形式は、アプリケーションにリンクされたスタティックライブラリまたはダイナミックライブラリです。ユーザエージェントにより、アプリケーションは SLP サービスに対して問い合わせることができます。

ユーザエージェントは、アルゴリズムに従って、クエリの送信先になるディレクトリエージェントのアドレスを取得します。指定したスコープの DA アドレスを取得すると、ユーザエージェントはそのスコープから応答がなくなるまで同じアドレスを使用し続けます。応答がなくなると、ユーザエージェントはそのスコープに対する別の DA アドレスを取得します。ユーザエージェントは、指定されたスコープのディレクトリエージェントのアドレスを次の方法で検索します。

1. 現在の要求のソケットハンドルが、指定したスコープの DA に接続されているかどうかを確認する。複数の要求の場合は、すでにキャッシュ化された接続がある可能性がある。
2. 指定したスコープと一致している DA の、既知のローカル DA キャッシュをチェックする。
3. 指定したスコープでローカル SA に対して DA を確認する (その後キャッシュに新しいアドレスを追加します)。
4. 指定したスコープに一致する DA のネットワーク設定済みのアドレスを DHCP に問い合わせる (その後キャッシュに新しいアドレスを追加します)。
5. 既知のポートで DA の検出要求をマルチキャストする (その後キャッシュに新しいアドレスを追加します)。

スコープを指定しない場合、指定スコープは「デフォルト」になります。つまり、SLP 設定ファイルで静的に定義されたスコープがなく、クエリでスコープを指定していない場合は、使用されるスコープは「デフォルト」という単語になります。また、OpenText eDirectory では、登録の際にスコープが指定されないことに注意してください。ただし、OpenText eDirectory で使用されるスコープが常に「デフォルト」というわけではありません。スコープが静的に設定されている場合、そのスコープがすべてのローカル UA 要求および SA 登録に対して、指定したスコープがない場合のデフォルトのスコープになります。

サービスエージェント

サービスエージェントの物理形式は、ホストマシン上での個別のプロセスです。Windows の場合は、slpd.exe がローカルマシン上のサービスとして実行されます。ユーザエージェントは、既知のポート上のループバックアドレスにメッセージを送信することによって、ローカルサービスエージェントを問い合わせます。

サービスエージェントは、潜在 DA アドレスに DA 検出要求を直接送信することにより、ディレクトリエージェントおよびそれがサポートするスコープリストを検出してキャッシュします。DA 検出要求は、次の方法で送信されます。

1. 静的に設定された DA アドレスをすべてチェックする (その後 SA の既知の DA キャッシュに新しい DA アドレスを追加します)。
2. DHCP から DA とスコープのリストを要求する (その後 SA の既知の DA キャッシュに新しいリストを追加します)。
3. 既知のポートで DA の検出要求をマルチキャストする (その後 SA の既知の DA キャッシュに新しいポートを追加します)。
4. DA によって定期的にブロードキャストされた DA のアドバータイズパケットを受信する (その後 SA の既知の DA キャッシュに新しいアドバータイズパケットを追加します)。

ユーザエージェントは常に最初にローカルサービスエージェントに対して問い合わせます。ローカルサービスエージェントの応答によってユーザエージェントが次の検出段階を続行するかどうかが決定されるため、このことは重要な点です (DHCP のこのケースについては、「221 ページの 「ユーザエージェント」」の手順 3 および 4 を参照してください)。

環境設定パラメータ

%systemroot%/slp.conf ファイル内の各環境設定パラメータも、次のようにして DA の検出を制御します。

```
net.slp.useScopes = <comma delimited scope list>
net.slp.DAAddresses = <comma delimited address list>
net.slp.passiveDADetection = <"true" or "false">
net.slp.activeDADetection = <"true" or "false">
net.slp.DAActiveDiscoveryInterval = <0, 1, or a number of seconds>
```

useScopes オプションは、SA の通知先のスコープ、および、サービスまたはクライアントアプリケーションで作成された登録またはクエリに指定したスコープが存在しない場合に、クエリが作成されるスコープを示します。OpenText eDirectory は常にデフォルトのスコープに通知し、問い合わせを行うため、このリストが OpenText eDirectory の登録およびクエリのデフォルトのスコープのリストになります。

DAAddresses オプションはコンマで区切られた IP アドレスのリストで、アドレスは 10 進数とドットで表記されます。このアドレスが他のすべてに対して優先されます。設定された DA のこのリストが登録またはクエリのスコープをサポートしない場合、検出を無効にしていない限りは、SA および UA はマルチキャスト方式で DA を検出します。

passiveDADetection オプションのデフォルトは「TRUE」です。ディレクトリエージェントは、設定に応じて定期的にそれ自体の存在をサブネットの既知のポート上にブロードキャストします。これらのパケットは DAAdvert パケットと名付けられます。このオプションに「FALSE」を設定した場合、ブロードキャスト方式のすべての DAAdvert パケットは SA に無視されます。

activeDADetection オプションのデフォルトも「TRUE」です。この設定により、SA はすべての DA に対して、指示された DAAdvert パケットで応答するように、定期的にブロードキャスト方式で要求できます。指示されたパケットはブロードキャストではありませんが、この要求に対する応答では SA に直接送信されます。このオプションに「FALSE」を設定した場合、SA は定期的な DA の検出要求をブロードキャストしません。

DAActiveDiscoveryInterval オプションは try-state パラメータです。デフォルト値は 1 です。これは、初期化の際に、SA が DA の検出要求を 1 回送る設定であることを意味する特別な値です。このオプションに 0 を設定すると、activeDADetection オプションに「FALSE」を設定した場合と結果は同じです。その他の値は、検出をブロードキャストする間隔を秒数で表します。

このオプションを正しく使用すると、サービスアドバータイズに使用するネットワーク帯域幅を適切に設定できます。ただし、デフォルト設定は平均的なネットワークで拡張性を最適化するように設計されています。

D 問題のトラブルシューティング

このセクションでは、OpenText eDirectory のインストールおよび設定に関する問題のトラブルシューティングに役立つ情報について説明します。

インストール問題のトラブルシューティング

次の表に、発生する可能性のある問題とそれを解決するための推奨されるアクションを示します。問題が続く場合は、OpenText の担当者にお問い合わせください。

問題	推奨されるアクション
インストールに長い時間がかかる . OpenText eDirectory を既存のツリーにインストールする場合に、インストールの完了までに長時間かかるときは、サーバの DSTrace 画面を確認してください。「-625 トランスポートできません」というメッセージが表示された場合は、アドレスキャッシュをリセットする必要があります。	アドレスキャッシュをリセットするには、システムコンソールで次のコマンドを入力します。 `set dstrace = *A`
コンテナ管理者の OpenText eDirectory インストールの失敗 OpenText eDirectory 9.0 のインストールプログラムでは、サーバが格納されているコンテナのスーパバイザ権を持つ管理者によるインストールがサポートされています。これを実行するには、OpenText eDirectory 9.0 をインストールする最初のサーバに、スキーマを拡張する［Root］のスーパバイザ権がある必要があります。この点から、後続のサーバには［Root］のスーパバイザ権は必要ありません。ただし、OpenText eDirectory 9.0 では、最初に OpenText eDirectory 9.0 がインストールされたプラットフォームによっては一部のスキーマが拡張されない場合があり、以降異なるプラットフォームでサーバをインストールするときに、［Root］に対するスーパバイザ権が要求される場合があります。	OpenText eDirectory 9.0 を複数のプラットフォームにインストールする場合は、各プラットフォームで OpenText eDirectory がインストールされる最初のサーバの［Root］に対するスーパバイザ権があることを確認してください。たとえば、OpenText eDirectory 9.0 をインストールする最初のサーバが Linux で実行されていて、OpenText eDirectory 9.0 を Solaris にもインストールする場合、各プラットフォームの最初のサーバは［Root］に対するスーパバイザ権を持っている必要があります。それ以降、各プラットフォームでインストールする場合は、サーバがインストールされているコンテナに対するコンテナ管理者の権利のみが必要になります。 追加情報については、『*OpenText eDirectory 8.7.x Readme Addendum*』のソリューション「NOVL83874 (http://support.novell.com/docs/Tids/Solutions/10073723.html)」を参照してください。

問題	推奨されるアクション
新しいネットワークインタフェースのデフォルトリスナ	Windows では、OpenText eDirectory はデフォルトで、NCP、HTTP、HTTPS、LDAP、および LDAPS のコンピュータ上に設定されたすべてのインターフェイスをリスンします。コンピュータに新しいネットワークインターフェイスアドレスを追加し、OpenText eDirectory を再起動すると、自動的にそのアドレスのリスンが開始され、それに応じて参照も追加されます。 **注 :** Linux では、インタフェースを n4u.server.interfaces パラメータに手動で追加する必要があります。
アップグレード後の複製に関する問題 OpenText eDirectory 9.0 にアップグレードし、暗号化複製を有効にしている場合は、ごくまれに複製に失敗します。	この問題を解決するには : 1. OpenText Identity Console で、[**オブジェクト管理**] を選択します。 2. [**オブジェクト**] ウィンドウで、[**タイプ**] ドロップダウンタブをクリックし、NCP サーバを選択し、[**検索**] をクリックします。 3. [**検索結果**] の下で NCP サーバを選択します。 4. [一般] タブで、[**その他**] を選択します。 5. [**オブジェクトの変更**] タブで、[**値がある属性**] をクリックします。 6. [属性の選択] 画面で、[**NCPKeyMaterialName**] を選択し、[**OK**] をクリックします。 たとえば、[**SSLCertificateDNS**]。 7. ステップ 6 で属性を変更したサーバでリンバを実行します。リンバの使用の詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』を参照してください。

問題	推奨されるアクション
OpenText eDirectory のパッチ適用中にエラー 1603 が発生しました Windows で sysprep.exe を実行してから、認証ユーザグループ特権を必要とするディレクトリに eDirectory をインストールまたは更新すると、1603 エラーが発生する可能性があります。 ［**管理者**］ユーザは、［**認証ユーザグループ**］(すべてのログインユーザ) と［**管理者グループ**］の両方の一部になっていました。Microsoft のトラブルシューティングガイドラインによると、このような二重のグループメンバーシップが原因で、ユーザアカウント制御 (UAC) システムとの競合が発生することがあり、管理者のインストール権限がブロックされる可能性があります。［**認証済みユーザ (Authenticated Users)**］グループの権限を調整することで問題が解決され、アップグレードを正常に完了できるようになりました。	Windows システムで sysprep.exe を実行した後に OpenText eDirectory をインストールまたは更新すると発生するエラーを解決するには、管理者が OpenText eDirectory のインストール済みディレクトリパーミッションを変更する必要があります。具体的には、ディレクトリのパーミッションから［**認証済みユーザ (Authenticated Users)**］グループを削除します。これを行うには、ディレクトリの［**プロパティ**］に移動し、［**セキュリティ**］タブを選択して、［**認証済みユーザ (Authenticated Users)**］グループを付与された権限のリストから削除します。

設定問題のトラブルシューティング

次の表に、発生する可能性のある問題とそれを解決するための推奨されるアクションを示します。問題が続く場合は、OpenText の担当者にお問い合わせください。

問題	推奨されるアクション
ループバック参照がディレクトリサーバから返される	OpenText eDirectory がループバックアドレスを監視するように設定されていると、ループバックアドレスは格納され、クライアントが検索または他の操作を実行したときにクライアントにループバックアドレスが返されます。サーバ以外のマシンから接続を試行したクライアントには、参照は適用されません。そのため、そのような方法でループバック参照を使用したクライアントは接続に失敗します。しかし、サーバから返されるそれ以外の参照については、クライアントは引き続き正常に使用できます。 各ループバック参照に接続して正しい参照を選択しようとするなら、クライアントのパフォーマンスに影響が出る可能性があります。

問題	推奨されるアクション

Linux 上で OpenText eDirectory 9.0 を設定中に発生する「ツリー名の検索に失敗しました : -632」エラー	Linux 上で eDirectory 9.0 を設定している間に、「ツリー名の検索に失敗しました : -632」エラーが発生する可能性があります。この問題を解決するには、次の手順を実行します。
	1. SLP パッケージをインストールした後、次のコマンドを入力し、手動で SLP を起動します。
	`/etc/init.d/slpuasa start`
	2. SLP パッケージをアンインストールした後、次のコマンドを入力し、手動で SLP を終了します。
	`/etc/init.d/slpuasa stop`

EBA を有効にしたセカンダリサーバを、EBA を有効にしていないサーバに追加すると、設定が失敗する	この問題を回避するにはまず、EBA を設定せずにセカンダリサーバを設定し、EBA 設定を使用する EBA にアップグレードします。
バックアップまたはアンチウィルス処理からの DIB ディレクトリの除外 OpenText eDirectory をインストールした後、OpenText eDirectory サーバの DIB ディレクトリをウィルス対策またはバックアップソフトウェアの処理の対象外となるように環境設定する必要があります。DIB ディレクトリをこれらの処理の対象外にしないと、DIB ディレクトリの破損や「-618 FFFFFD96 INCONSISTENT DATABASE」エラーが発生する可能性があります。	DIB ディレクトリのバックアップには、OpenText eDirectory バックアップツールを使用します。OpenText eDirectory のバックアップの詳細については、『OpenText™ CE 24.4 (v9.3) eDirectory 管理ガイド』の「*ロールフォワードログのバックアップと削除*」を参照してください。
IP AG 証明書が SLES 11 64 ビットプラットフォームで作成されない	OpenText eDirectory 9.0 が IPv4 と IPv6 の両方で設定されていて、その片方 (たとえば IPv4) のエントリのみが /etc/hosts ファイルにあり、もう片方のインターフェイスはリモートマシンからアクセス可能であるとします。両方の IP を監視するように OpenText eDirectory を設定すると、IP AG 証明書は /etc/hosts ファイルにリストされている IP 用のみが生成されます。この例で生成されるのは、IPv4 用となります。
複数のインスタンスのデフォルトのインスタンスパス . ホストで OpenText eDirectory の 2 つ目のインスタンスを設定する際に、デフォルトのパスに設定するよう求めるメッセージが表示されます。	別のパスを選択し、続行します。

問題	推奨されるアクション
eDirectory を非ルートユーザとして設定すると ndsconfig ユーティリティでエラーが表示される ◆ Group edirAdmin set failed ◆ WARNING: Unable to set permission on directory ◆ Checking if server is ready to service requests 次の条件が当てはまる場合、OpenText eDirectory 9.2.5 以降のバージョンのインストールまたはアップグレード中に、ユーザにこれらのエラーが表示されることがあります。 ◆ edirAdmin ユーザグループが OpenText eDirectory に見つからない場合。 ◆ ユーザが edirAdmin ユーザグループのメンバーではない場合。	この問題をトラブルシューティングするには、管理者は最初に edirAdmin ユーザグループを手動で作成してから、そのユーザをこのグループに追加する必要があります。 OpenText eDirectory がインストールされているサーバで、端末を開き、次のコマンドを実行します。 ◆ edirAdmin グループを追加するには、groupadd edirAdmin を実行します。 ◆ ユーザを edirAdmin グループに追加するには、usermod -a -G edirAdmin <username> を実行します。 ここで、<username> はユーザの名前です。たとえば、John というユーザを edirAdmin グループに追加するには、次のコマンドを実行する必要があります。 $ usermod -a -G edirAdmin john **注 :** ユーザを edirAdmin ユーザグループに追加する前に、管理者は、ユーザが端末セッションを通じてログインしていないことを確認する必要があります。追加された後、ユーザは ndsconfig ユーティリティを使用して OpenText eDirectory にログインし、設定できます。

OpenText eDirectory の複数インスタンス問題のトラブルシューティング

次の表に、発生する可能性のある問題とそれを解決するための推奨されるアクションを示します。問題が続く場合は、OpenText の担当者にお問い合わせください。

問題	推奨されるアクション
最初のインスタンスが停止している場合、HTTP が動作しない Linux プラットフォーム上で、OpenText eDirectory が複数の NIC カードを持つコンピュータ上に設定されており、HTTP が 1 つ以上のインターフェイスにバインドされている場合、最初のインターフェイスが停止すると、残りのインターフェイスから HTTP にアクセスできなくなります。 これは、残りのインタフェースが要求を最初のインタフェースへリダイレクトしているのに対して、最初のインタフェースが停止しているためです。	この問題を解決するには、最初のインターフェイスが停止している場合、OpenText eDirectory を再起動します。
指定されたインタフェースが正しくない場合に、ndsd がデフォルトポートに戻る	eDirectory の 2 番目のインスタンスを ndsconfig new または ndsmanage を使用して作成する場合、指定したインタフェースが正しくないと、nds はデフォルトインタフェースを使用しようとします。デフォルト以外のポート (1524 など) を指定し、指定したインタフェースが正しくないと、デフォルトインタフェースおよびデフォルトポート 524 が使用されます。 n4u.server.interfaces の場合、指定したインタフェースが正しくないと、ndsd は 1 番目のインタフェースの監視を試行し、ポート番号は n4u.server.tcp-port に指定されているものが使用されます。

問題	推奨されるアクション

.edir ディレクトリを再構築する方法	OpenText eDirectory の複数インスタンスのトラッキングには、.edir ディレクトリが使用されます。失われたまたは破損したインスタンスファイル (instances.$uid ファイル。$uid はシステム内でのユーザのユーザ ID を表す) を再作成するには、インスタンスファイルを個別に再作成する必要があります。
	
	これらのファイルには、ユーザによって設定されたすべてのインスタンスの nds.conf ファイルの絶対位置が含まれています。たとえば、uid が 1000 であるユーザは、次のエントリを含む /etc/opt/novell/eDirectory/conf/.edir/instances.1000 インスタンスファイルを作成する必要があります。
	
	`/home/user1/instance1/nds.conf`
	
	`/home/user1/instance2/nds.conf`

ndsconfig ユーティリティ

次の表に、発生する可能性のある問題とそれを解決するための推奨されるアクションを示します。問題が続く場合は、OpenText の担当者にお問い合わせください。

問題	推奨されるアクション
デフォルト以外の場所から実行するように ndsconfig を設定する	デフォルトの /opt/novell/eDirectory/bin ディレクトリ以外の場所から ndsconfig ユーティリティを実行するとエラーを受け取る場合は、ndsconfig を実行する前に必ず ndspath をエクスポートしてください。次のコマンドを実行します。 `source /opt/novell/eDirectory/bin/ndspath` コマンドをエクスポートした後、ndsconfig と入力して ndsconfig ユーティリティを実行します。./ndsconfig とは入力しないでください。
無効な環境設定ファイルパスが ndsconfig で検証されない	必要な環境設定ファイルを作成するには、ndsconfig で完全なパスと環境設定ファイル名を使用する必要があります。環境設定ファイルとインスタンスディレクトリの両方に同じパス名が指定されると、ndsconfig は環境設定ファイルを作成できないため、処理を中止します。

問題	推奨されるアクション
ndsconfig で英語以外の文字がジャンク文字で出力される	Linux 上で ndsconfig get コマンドを実行すると、英字以外の文字を含むパラメータに対しては無意味な文字が出力されます。 この問題に対処するには、表示したい特定のパラメータ名を次のように入力します。 ndsconfig get < *表示する対象のパラメータ* > パラメータの一覧については、nds.conf マニュアルページを参照してください。
OES 2018SP2 上で OpenText eDirectory をアップグレードするときに、/etc/opt/novell/eDirectory/conf/env ファイルに NDSD_NLDAP_IGNORE_CRITICALITY = True 環境変数がない。	この問題をトラブルシューティングするには、ndsconfig upgrade コマンドを手動で実行して、/etc/opt/novell/eDirectory/conf/env ファイルに NDSD_NLDAP_IGNORE_CRITICALITY = True 環境変数を追加する必要があります。

NMAS インストールのトラブルシューティング

- Novell Client をアンインストールする場合は、別のアプリケーションで使用されている NMAS クライアントをアンインストールし、再インストールする必要があります。
- ユーザが NMAS を使用するためには、ユーザのオブジェクトの書き込み可能なレプリカを保持するサーバに NMAS をインストールしておく必要があります。
- NMAS ソフトウェアを実行する各クライアントワークステーションに Novell International Cryptographic Infrastructure (NICI) クライアントをインストールしておく必要があります。
- NMASをインストールした後でサーバを再起動せずにパスワードをリセットしようとすると、エラーメッセージが表示されます。
- ログインメソッドは最新の状態にしておく必要があります。OpenText eDirectory の OES/Linux のインストールでは、メソッドをアップグレードする手段が提供されない場合があります。

証明書サーバのインストールのトラブルシューティング

インストール中にファイルデータの競合が発生

以前インストールしたファイルより新しいバージョンのファイルが存在することを示すメッセージを受け取った場合は、常に新しいファイルを上書きするように選択してください。

サーバのリストが不完全

インストール時に表示されるサーバのリストには、IP のみを使用するように設定されたサーバが含まれない可能性があります。テキストボックスにサーバの名前を入力すると、名前が表示されていないサーバに OpenText 証明書サーバをインストールできます。

インストール中に障害が発生

組織の認証局またはサーバ証明書の作成中、またはルート認証局証明書のエクスポート中にインストールに失敗する場合は、インストールを繰り返す必要はありません。この時点では、ソフトウェアは正常にインストールされています。Identity Console を使用すれば、組織の認証局およびサーバ証明書を作成でき、ルート認証局をエクスポートできます。

SLES 11 64 ビットプラットフォームで IP 自動生成証明書が作成されない

OpenText eDirectory 9.0 が IPv4 と IPv6 の両方で設定されていて、その片方 (たとえば IPv4) のエントリのみが /etc/hosts ファイルにあり、もう片方のインターフェイスはリモートマシンからアクセス可能であるとします。両方の IP を監視するように OpenText eDirectory を設定すると、IP AG 証明書は /etc/hosts ファイルにリストされている IP 用のみが生成されます。この例で生成されるのは、IPv4 用となります。

証明書オブジェクトの RDN が最大長を超えていると IP 自動生成 IPv6 証明書が作成されない

OpenText eDirectory 9.0 のインストール中、IPv4 と IPv6 の両方のアドレスでリスンしているときに、IP AG <IPv6> 証明書 (KMO) が作成されません。

これは、証明書オブジェクトの RDN が最大長の 64 文字を超えたときに発生します。これを処理するために、圧縮形式の IPv6 アドレスが使用されます。これにより、最大長を超えている場合でも、要求に合わせてアドレスが分割されます。アドレスは、(逆順で)3 番目のコロンから分割されます。

たとえば、2508:f0g0:1003:0061:0000:0000:0000:0002 という IPv6 アドレスが切り捨てられると、0000:0000:0002 となります。これにより、アドレスが切り捨てられても、ホストは正しく識別されます。

CA がホストされていないサーバ用にデフォルトのサーバ証明書が再作成されると、HTTP サーバが IP AG 証明書に関連付けられる

Identity Console を使用して、デフォルトの関連付けを手動で変更します。次の手順では、変更方法について説明します。

1. Identity Console にログインし、[**オブジェクト管理**] の [**タイプ**] ドロップダウンメニューから http サーバオブジェクトを選択し、[**検索**] をクリックします。
2. [**検索結果**] メニューで [http サーバ - <server name/CN>] を選択します。
3. [**オブジェクトの変更**] タブで、[**値がある属性** ] をクリックします。

4 ［**属性の選択**］画面で、httpKeyMaterialObjects 属性を検索し、［**編集** ］をクリックします。

5 SSL 証明書 DNS に対する HTTP サーバオブジェクトの関連付けを編集し、［**OK**］をクリックします。

インストールのトラブルシューティング

ブリッジパッチが EBA 対応の OpenText eDirectory 9.2.8 環境に適用されていない場合、OpenText eDirectory 9.3 レプリカを追加すると、次のエラーメッセージがスローされます。

- **Linux:** リング内の 1 つ以上 EBA 対応サーバにブリッジパッチが適用されていません。
- **Windows:** 資格情報が正しくありません。確認して再試行してください。

回避策 : ブリッジパッチをすべてのサーバに適用します。