NetIQ® Identity Manager インストールおよびアップグレードガイド (Linux 用)

2024 年 5 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。ここに記載されている情報は予告なく変更される場合があります。認定関連の通知や商標などの追加情報については、以下を参照してください。http://www.microfocus.com/about/legal/ (http://www.microfocus.com/about/legal/)

目次

本書およびライブラリについて **13**

ページのパート I Identity Manager 環境の概要 **15**

1 Identity Manager コンポーネントの簡単な紹介 **17**

Identity Manager サーバのコンポーネント . . . 18
Identity Manager サーバ . . . 19
リモートローダ . . . 20
ファンアウトエージェント . . . 20
Identity Console . . . 21
Identity アプリケーションのコンポーネント . . . 21
ユーザアプリケーション . . . 21
認証サービス . . . 22
Self-Service Password Reset . . . 22
Web アプリケーションサーバ . . . 22
Identity Applications データベース . . . 23
Identity Applications 用のドライバ . . . 23
Identity Reporting のコンポーネント . . . 24
Identity Reporting . . . 24
認証サービス . . . 24
Self-Service Password Reset . . . 25
Identity Reporting データベース . . . 25
Web アプリケーションサーバ . . . 25
Identity Reporting 用のドライバ . . . 26
Sentinel Log Management for Identity Governance and Administration . . . 26
Identity Manager ツール . . . 27
Designer for Identity Manager . . . 27
Identity Manager 用の Analyzer . . . 27
機能アーキテクチャ . . . 28
Identity Manager の展開オプション . . . 29
Identity Manager 展開のサンプル . . . 30
Advanced Edition 展開のサンプル . . . 32
Standard Edition 展開のサンプル . . . 34

ページのパート II Identity Manager のインストールの計画 **37**

2 インストールの計画 **39**

実装チェックリスト . . . 39
推奨されるインストールシナリオとサーバセットアップ . . . 40
SLM for IGA をインストールするタイミングの決定 . . . 40
分散セットアップでのインストールに関する考慮事項 . . . 41
ハードウェア要件の決定 . . . 43
システム要件ワークシート . . . 44
SLES サーバへの Identity Manager のインストール . . . 44

RHEL サーバへの Identity Manager のインストール 45
前提条件 46
サーバに依存ライブラリがあることの確認 47
RHEL 9.x でのインストールメディア用のリポジトリの作成 47
RHEL 8.x でのインストールメディア用のリポジトリの作成 49
前提条件チェックの実行 50

3 Identity Manager コンポーネントのインストールに関する考慮事項 51

インストールの順序 51
インストールおよび設定プロセスの理解 52
Identity
Manager エンジンコンポーネントおよびリモートローダのインストールに関する考慮事項 . 53
識別情報アプリケーションコンポーネントのインストールに関する考慮事項 55
インストールの考慮事項 55
データベースに関する検討事項 56
識別情報アプリケーションのデータベースの設定 57
Identity Reporting コンポーネントのインストールに関する考慮事項 59
Identity Reporting の前提条件 60
Identity Reporting の監査イベントの識別 60
Designer のインストールに関する考慮事項 61
Analyzer のインストールに関する考慮事項 62
SLM for IGA のインストールに関する考慮事項 62
Identity Manager コンポーネントが使用するポートの確認 63

ページのパート III Identity Manager コンポーネントのインストールと設定 65

4 Identity Manager のインストール 67

対話型インストールの実行 67
サイレントインストールの実行 68
root 以外のユーザによる Identity Manager エンジンのインストール 69
NICI のインストール 69
アイデンティティボールトの root 以外のインストールの実行 69
エンジンの root 以外のインストールを実行 71
SSPR のインストール 72
SSPR の対話型インストールの実行 73
SSPR のサイレントインストールの実行 73
リモートローダのインストール 73
インタラクティブインストール 74
サイレントインストール 74
Designer のインストール 74
Analyzer のインストール 75
ウィザードを使用した Analyzer のインストール 75
Analyzer のサイレントインストール 76
Analyzer.ini への XULRunner の追加 76
Sentinel Log Management for Identity Governance and Administration のインストール 77
root ユーザによる Sentinel Log Management のインストール 77
非 root ユーザによる Sentinel Log Management のインストール 78
Java リモートローダのインストール 79
ディレクトリ構造の理解 80

5 Identity Manager コンポーネントの設定 83

設定時における直感的でないパスワードの使用 . . . 83
環境設定パラメータの理解 . . . 83
ドライバセットの作成と設定 . . . 96
Identity Manager コンポーネントの設定 . . . 98
対話型設定の実行 . . . 98
サイレント設定の実行 . . . 98
SSPR の設定 . . . 98
対話型設定の実行 . . . 99
サイレント設定の実行 . . . 99
OSP サーバでのシングルサインオンアクセス設定の変更 . . . 99

6 インストールを完了するための最終ステップ 101

Identity Vault の設定 . . . 101
Identity Vault の値インデックスの作成 . . . 102
アイデンティティボールトへの Identity Applications および Identity Reporting 証明書の手動インポート . . . 102
アイデンティティボールト管理者としての非管理者ユーザの設定 . . . 103
リモートローダとドライバの設定 . . . 103
接続システムの設定 . . . 103
ドライバセットの作成と設定 . . . 104
ドライバの作成 . . . 107
ポリシーの定義 . . . 107
パスワードを忘れた場合の管理の設定 . . . 108
Self Service Password Reset によるパスワードを忘れた場合の管理 . . . 108
外部システムによるパスワードを忘れた場合の管理 . . . 110
分散環境またはクラスタ化環境におけるダッシュボードの SSPR リンクの更新 . . . 112
識別情報アプリケーションの設定 . . . 113
Configupdate ユーティリティのパスワードの保存 . . . 113
識別情報アプリケーションの設定の管理 . . . 114
パーミッションインデックスの場所の指定 . . . 142
識別情報アプリケーション用の REST API の展開 . . . 143
Identity Applications を使用した Oracle データベースの設定 . . . 143
Oracle サービス名を使用した Oracle データベースへのアクセス . . . 144
手動によるデータベーススキーマの作成 . . . 144
識別情報アプリケーションのシングルサインオン設定の管理 . . . 146
識別情報アプリケーションの起動 . . . 147
識別情報アプリケーションの設定と使用方法の検討事項 . . . 147
データ収集用のランタイム環境の設定 . . . 148
識別情報アプリケーションからのデータ収集に関するデータ収集サービスドライバの設定 148
データ収集サービスドライバの移行 . . . 149
カスタム属性とカスタムオブジェクトのサポートの追加 . . . 151
複数のドライバセットのサポートの追加 . . . 154
SSL を使用したリモートモードでのドライバ実行設定 . . . 156
Identity Reporting の設定 . . . 158
管理対象システムのゲートウェイドライバの設定 . . . 158
［Identity Data Collection Services (アイデンティティデータ収集サービス)］ページへのデータソースの手動追加 . . . 159
Oracle データベースでのレポートの実行 . . . 159
データベーススキーマの手動生成 . . . 159
Identity Reporting 用の REST API の展開 . . . 165

リモート PostgreSQL データベースへの接続 . . . 165
分散型セットアップでの Identity Reporting 用 OSP へのクライアントの登録 . . . 166
Identity Reporting のインストール後の手順 . . . 166
root 以外のインストールの完了 . . . 167
パスワードポリシーのコンテナの作成 . . . 167
電子メール通知のグラフィックサポートの追加 . . . 167
Identity Manager のアクティベート . . . 168
分散展開用の SSL 証明書の作成 . . . 168

ページのパート IV FIPS モードでの Identity Manager の設定 171

7 FIPS 用の JRE の設定 173

8 FIPS モードでの Identity Manager エンジンの設定 175

手順 . . . 175

9 FIPS モードでのファンアウトエージェントの設定 177

手順 . . . 177

10 FIPS モードでの Identity Applications と Identity Reporting の設定 179

手順 . . . 179

11 FIPS モードでのリモートローダの設定 183

ページのパート V Identity Manager のデータの新しいインストールへのマイグレート 185

12 Identity Manager をマイグレートする準備 187

マイグレーションを実行するためのチェックリスト . . . 187

13 Identity Manager の新しいサーバへのマイグレート 189

前提条件 . . . 189
Designer プロジェクトのマイグレーションの準備 . . . 189
Identity Manager エンジンの新しいサーバへのマイグレート . . . 190
ドライバセットのサーバ固有情報のコピー . . . 191
Designer でサーバ固有の情報をコピーする . . . 191
Identity Console でのサーバ固有情報の変更 . . . 192
ユーザアプリケーションのサーバ固有の情報を変更する . . . 193
Identity Applications の移行 . . . 193
データベースの新しいサーバへの移行 . . . 193
新しいサーバへの Identity Applications のインストール . . . 195
Identity Reporting の移行 . . . 196
Identity Reporting 用ドライバの更新 . . . 196
Identity Reporting 用ドライバの展開 . . . 196
新しいデータベースへの既存のデータの移行 . . . 197
新しい Reporting サーバの設定 . . . 201

データ同期ポリシーの作成 201

ページのパート VI Identity Manager のアップグレード 203

14 Identity Manager のアップグレードの準備 205

Identity Manager のアップグレードのチェックリスト 205
アップグレードプロセスの理解 207
サポートされているアップグレードパス 208
Identity Manager 4.7.5 バージョンからのアップグレード 208
Identity Manager 4.8.x バージョンからのアップグレード 210
現在の設定のバックアップ 212
Designer のプロジェクトのエクスポート 212
ドライバ環境設定のエクスポート 213

15 Identity Manager コンポーネントのアップグレード 215

アップグレード順序 215
アップグレードに関する考慮事項 216
Designer のアップグレード 217
Identity Manager エンジンのアップグレード 218
インストール前の Linux RPM の検証 218
アイデンティティボールトのアップグレード 219
Identity Manager エンジンのアップグレード 220
root 以外のユーザによる Identity Manager エンジンのアップグレード 222
リモートローダのアップグレード 222
Java リモートローダのアップグレード 223
Identity Manager ドライバの停止と起動 224
ドライバの停止 224
ドライバの起動 225
Identity Manager ドライバのアップグレード 226
新しいドライバの作成 226
既存のコンテンツをパッケージのコンテンツと交換 226
現在のコンテンツを維持しつつパッケージを使用する新しいコンテンツを追加 . . . 227
Identity Applications のアップグレード 228
アップグレードに関する考慮事項 229
システム要件 231
アップグレードプログラムについて 231
PostgreSQL のアップグレード 231
Identity Applications のコンポーネントのアップグレード 231
Identity Applications コンポーネントのアップグレード後のタスク 234
アップグレード後のバージョン番号の確認 238
Identity Reporting のアップグレード 238
アップグレードに関する考慮事項 239
Sentinel Log Management for IGA のアップグレード 239
Identity Reporting のアップグレード 239
Reporting のアップグレード後の手順 240
Identity Reporting のアップグレードの検証 242
Analyzer のアップグレード 242
アップグレード後のタスク 242
サービスドライバのアップグレード 242
未使用の Log4j JAR の削除 243
新しいサーバをドライバセットに追加する 244

ドライバセットから古いサーバを削除する 244
ドライバへのカスタムポリシーとルールの復元 245
Designer を使用したドライバへのカスタムポリシーとルールの復元 245
Identity Console を使用したドライバへのカスタムポリシーとルールの復元 246

16 Advanced Edition から Standard Edition への切り替え 247

ページのパート VII Identity Manager コンテナの展開 249

17 コンテナ展開の計画 251

システム要件 251
Docker イメージの取得 251

18 Identity Manager コンテナの展開 253

ベストプラクティス 253
コンテナボリュームデータの管理 254
サイレントプロパティファイルの作成 254
リモートローダコンテナの展開を使用したリモートローダインスタンスの自動起動 . . . 257
コンテナ展開のモード 258
単一サーバ上へのコンテナの展開 258
前提条件 258
Identity Manager エンジンコンテナの展開 259
リモートローダコンテナの展開 260
ファンアウトエージェントコンテナの展開 260
アイデンティティボールト認証局での証明書の生成 261
OSP コンテナの展開 262
PostgreSQL コンテナの展開 263
Identity Applications コンテナの展開 264
フォームレンダラコンテナの展開 265
ActiveMQ コンテナの展開 266
Identity Reporting コンテナの展開 266
SSPR コンテナの展開 267
Identity Console コンテナの展開 268
分散サーバ上へのコンテナの展開 269
オーバーレイネットワークの設定 269
前提条件 270
コンテナがアクセスするポートの公開 271
Identity Manager エンジンコンテナの展開 273
リモートローダコンテナの展開 273
ファンアウトエージェントコンテナの展開 274
アイデンティティボールト認証局での証明書の生成 275
OSP コンテナの展開 279
PostgreSQL コンテナの展開 280
Identity Applications コンテナの展開 281
フォームレンダラコンテナの展開 282
ActiveMQ コンテナの展開 283
Identity Reporting コンテナの展開 283
SSPR コンテナの展開 284
Identity Console コンテナの展開 285

19 Identity Manager コンテナのアップグレード **287**

コンテナの更新の前提条件 . . . 287
分散サーバ上でのコンテナの更新 . . . 288
Identity Manager エンジンコンテナの更新 . . . 288
リモートローダコンテナの更新 . . . 289
ファンアウトエージェントコンテナの更新 . . . 289
OSP コンテナの更新 . . . 290
PostgreSQL コンテナの更新 . . . 290
Identity Applications コンテナの更新 . . . 291
フォームレンダラコンテナの更新 . . . 292
ActiveMQ コンテナの更新 . . . 292
Identity Reporting コンテナの更新 . . . 292
SSPR コンテナの更新 . . . 293
Identity Console コンテナの更新 . . . 293
単一サーバ上でのコンテナの更新 . . . 293
Identity Manager エンジンコンテナの更新 . . . 294
リモートローダコンテナの更新 . . . 294
ファンアウトエージェントコンテナの更新 . . . 295
OSP コンテナの更新 . . . 295
PostgreSQL コンテナの更新 . . . 295
Identity Applications コンテナの更新 . . . 297
フォームレンダラコンテナの更新 . . . 297
ActiveMQ コンテナの更新 . . . 297
Identity Reporting コンテナの更新 . . . 297
SSPR コンテナの更新 . . . 298
Identity Console コンテナの更新 . . . 298

ページのパート VIII Ansible を使用した Identity Manager コンテナの展開 **299**

20 展開の計画 **301**

Ansible ノードの準備 . . . 301
コントロールノードの準備 . . . 302
管理対象ノードの準備 . . . 304
setup.csv ファイルの作成 . . . 304

21 コンテナの展開 **307**

22 展開後のタスク **309**

ページのパート IX Microsoft Azure Kubernetes Service への Identity Manager コンテナの展開 **311**

概要 . . . 312

23 展開の計画 **313**

24 展開の準備 **317**

25 展開手順 **319**

Azure クラウド上へのインフラストラクチャの展開 319
Configuration Generator を使用した展開 319
設定テンプレートを使用した展開 323
Azure Kubernetes Service への Identity Manager コンテナの展開 334
Azure Kubernetes Service でのリモートローダサービスの設定 336
Azure Kubernetes Service でのファンアウトエージェントサービスの設定 338

26 展開後のタスク **341**

Identity Manager アプリケーションへのアクセス 341
Azure 仮想マシンへの Designer の展開 341
Azure 仮想マシンへの SLM for IGA の展開 341
AKS の RPM 更新とサードパーティファイルの処理 342

27 Identity Manager コンテナの展開の保守 **343**

データ永続化レイヤの管理 343
ポッドの再起動 344
ポッドへのアクセス 345
Azure Kubernetes クラスタの再起動 345
Identity Manager エンジンのレプリカのスケーリング 347
Identity Manager エンジンのレプリカをダウンスケールするユースケース 347
Identity Manager エンジンのレプリカのアップスケール 348
OSP と Identity Applications のレプリカのスケーリング 348
Terraform で作成されたリソース全体の破棄 348

ページのパート X AWS EC2 での Identity Manager の展開 **351**

28 AWS EC2 での Identity Manager の計画と実装 **353**

前提条件 353
展開手順 353
AWS 仮想プライベートクラウドの準備 355
インスタンスの作成と展開 358
EC2 インスタンスの準備 359
Identity Manager コンポーネントの設定 361
Identity Applications および Identity Reporting 用のデータベースの設定 362
Designer の設定 363
AWS EC2 ロードバランサの作成 364
(オプション) 登録済みのホストゾーンを使用したエイリアス DNS の作成 369
Identity Manager のコンポーネントへのアクセス 370
セキュリティ上の考慮事項 371

29 ハイブリッド Identity Manager のシナリオの例 373

リモートローダによる接続の使用 . . . 373
マルチサーバドライバセット接続の使用 . . . 374
eDirectory ドライバ接続の使用 . . . 376

ページのパート XI 高可用性のための Identity Manager の展開 379

30 クラスタ環境における Identity Manager のインストールの準備 381

前提条件 . . . 381
識別ボールト . . . 381
識別情報アプリケーション . . . 382
Identity Applications のデータベース . . . 382
識別情報アプリケーションで使用するクラスタの準備 . . . 383
Tomcat 環境のクラスタグループの理解 . . . 383
ワークフローエンジン ID のシステムプロパティの設定 . . . 383

31 Identity Manager のクラスタ展開ソリューションのサンプル 385

前提条件 . . . 385
インストール手順 . . . 386
iSCSI サーバの設定 . . . 386
すべてのノード上の iSCSI Initiator の設定 . . . 387
共有ストレージのパーティション化 . . . 388
HA Extension のインストール . . . 388
softdog ウォッチドッグのセットアップ . . . 389
HA クラスタの設定 . . . 389
クラスタノードでのアイデンティティボールトおよび Identity Manager エンジンのインストールと設定 . . . 390
アイデンティティボールトのリソースの設定 . . . 392
eDirectory および共有ストレージの子リソースのプリミティブ . . . 392
場所制約スコアの変更 . . . 393

32 Tomcat アプリケーションサーバ上へのアイデンティティアプリケーションクラスタ展開ソリューションのサンプル 395

前提条件 . . . 396
クラスタの準備 . . . 397
Tomcat 環境のクラスタグループの理解 . . . 397
ワークフローエンジン ID のシステムプロパティの設定 . . . 397
インストール手順 . . . 398
クラスタリングのパーミッションインデックスの有効化 . . . 403
クラスタリング用のユーザアプリケーションドライバの環境設定 . . . 404
クラスタリング用の OSP と SSPR の設定 . . . 404
クラスタリングをサポートするための SSPR の設定 . . . 404
クラスタノード上でのタスクの設定 . . . 404

33 Identity Manager のコンポーネントのアンインストール 407

識別ボールトからのオブジェクトの削除 . . . 407
Identity Manager エンジンのアンインストール . . . 408
Identity Applications のアンインストール . . . 408

Identity Reporting のコンポーネントのアンインストール 409
レポーティングドライバの削除 409
Identity Reporting のアンインストール 410
Sentinel Log Management for IGA のアンインストール 410
Designer のアンインストール 410
Analyzer のアンインストール 411

34 トラブルシューティング 413

ログファイルの場所 413
ログインに関連する問題のトラブルシューティング 413
インストールとアンインストールに関連する問題のトラブルシューティング 416
アップグレードに関連する問題のトラブルシューティング 420
Ansible の展開に関連する問題のトラブルシューティング 425
AKS の展開に関連する問題のトラブルシューティング 428
Identity Manager エンジンのトラブルシューティング 434
Identity Applications と Identity Reporting のトラブルシューティング 436
その他の問題のトラブルシューティング 441

本書およびライブラリについて

この「インストールおよびアップグレードガイド」には、NetIQ Identity Manager (Identity Manager) 製品のインストールまたは更新の手順が記載されています。このガイドでは、分散環境に個々のアプリケーションをインストールおよび設定するプロセスについて説明します。

本書の読者

本書は、組織の識別情報管理ソリューションの構築に必要なコンポーネントのインストールを行う識別情報アーキテクトおよび識別情報管理者向けの情報を提供します。

ライブラリに含まれているその他の情報

Identity Manager のライブラリの詳細については、Identity Manager マニュアルの Web サイト (https://www.netiq.com/documentation/identity-manager-49/) を参照してください。

Identity Manager 環境の概要

このガイドでは、Identity Manager をインストールおよび設定するために完了する必要のあるタスクに焦点を当てています。

NetIQ Identity Manager を初めて使用する場合は、以下のセクションの情報を使用して、ソリューションとそれを構成するコンポーネントを理解してください。ダウンロードしてインストールできるコンポーネントは、Identity Manager Edition によって決まります。

- Identity Manager コンポーネントの簡単な紹介
- 機能アーキテクチャ

1 Identity Manager コンポーネントの簡単な紹介

お客様のさまざまなニーズに対応するために、Identity Manager は Advanced Edition と Standard Edition で利用可能です。各エディションは特定の機能セットで構成され、各機能は複数のコンポーネントによって処理されます。したがって、Identity Manager の実装には、要件に応じて次のコンポーネントの 1 つまたはすべてを含めることができます。

- Identity Manager サーバ
- 識別情報アプリケーション
- Identity Reporting
- Identity Manager ツール

図 1-1 には、Identity Manager Advanced Edition 環境に展開されるコンポーネントが示されています。

図 1-1 Identity Manager Advanced Edition のコンポーネント

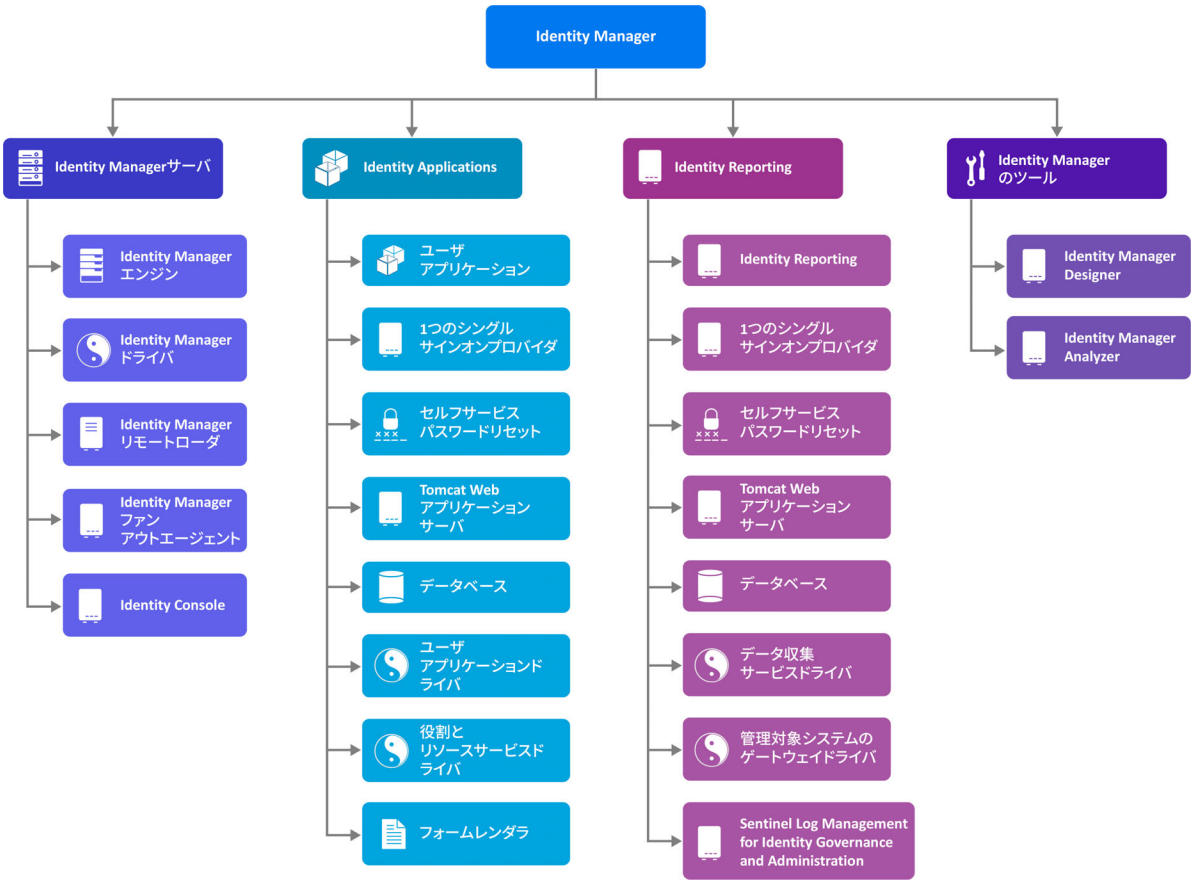
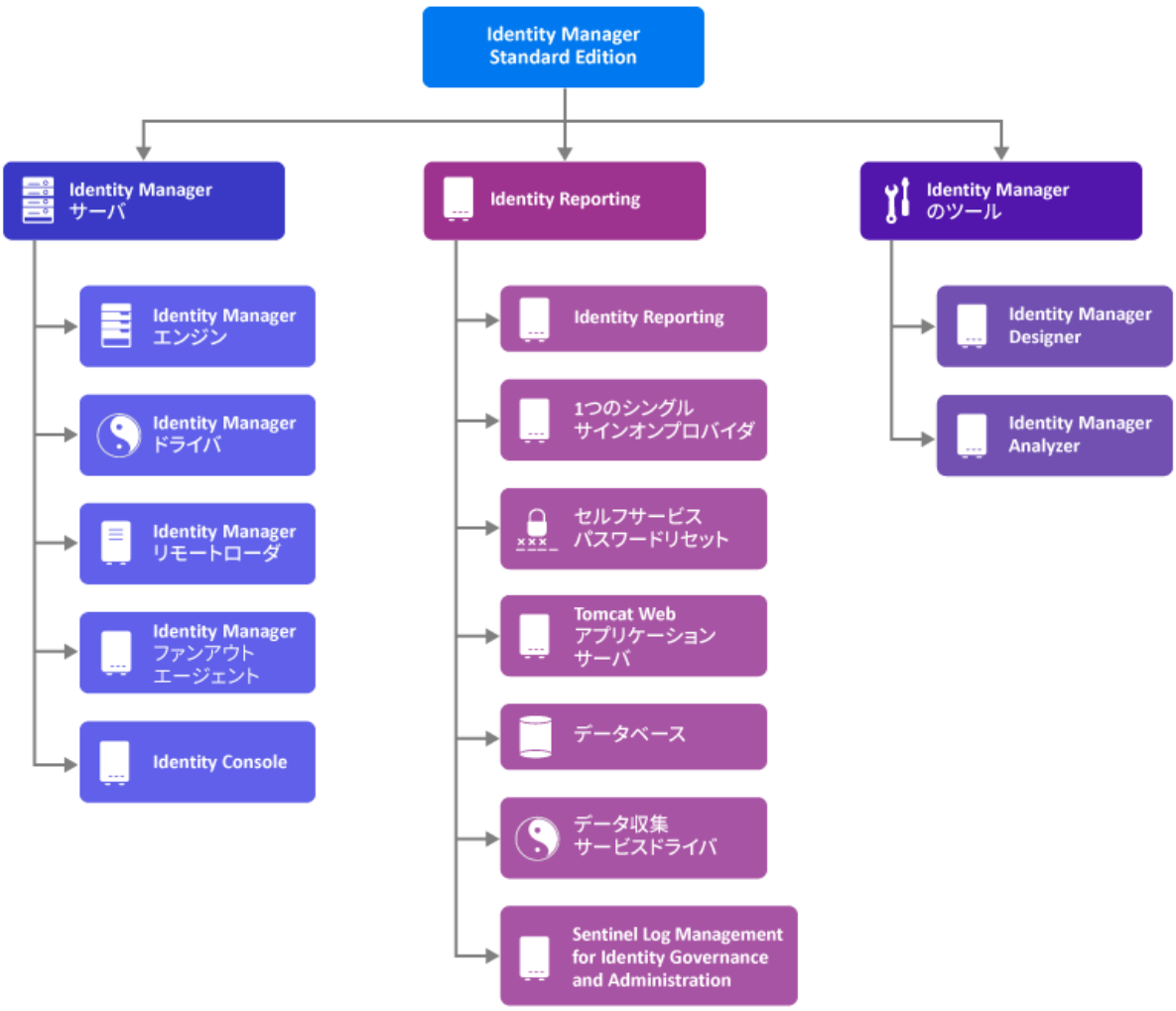


図 1-2 には、Identity Manager Standard Edition 環境に展開されるコンポーネントが示されています。

図 1-2 *Identity Manager Standard Edition のコンポーネント*



コンポーネント同士の相互作用に基づいて、一部のコンポーネントはコンポーネントのグループとして論理的にインストールされます。一部のコンポーネントは、インストール操作を簡単にするためにスタンドアロンコンポーネントとしてインストールされます。コンポーネントが互いにどのように相互作用するかについては、『NetIQ Identity Manager Overview and Planning Guide』を参照してください。

後続のセクションの情報を確認し、コンポーネントをグループ化する方法および各コンポーネントまたはコンポーネントのグループをインストールする方法を理解してください。

Identity Manager サーバのコンポーネント

すべてのインストールに必要

Identity Manager Server インストールは、以下のコンポーネントで構成されています。

Identity Manager サーバ

Identity Manager Server は Identity Manager 内でタスクを実行します。Identity Vault、Identity Manager エンジン、および Identity Manager ドライバで構成されます。

Identity Manager Server の操作をサポートするために、インストールプログラムによって、サポートされているバージョンの Azul Java Runtime Environment (JRE) がインストールされます。Identity Manager Server コンポーネントをインストールするには、インストールプログラムの **[Identity Manager エンジン]** インストールオプションを使用します。

識別ボールト

Identity Manager エンジンをインストールすると、インストールプロセスによって Identity Vault への接続が作成および設定されます。Identity Manager は、すべての識別情報データのデフォルトリポジトリとして Identity Vault を使用します。識別情報データには、ユーザアカウントと組織データを含む、管理された識別情報の現在の状態が含まれます。

Identity Manager エンジン

Identity Manager エンジンは、識別ボールトまたは接続アプリケーションで発生するすべてのデータ変更を処理します。Identity Manager エンジンが実行されているサーバを Identity Manager サーバと呼びます。

Identity Manager ドライバ

Identity Manager Server はユーザのプロビジョニングを処理し、ドライバを介して接続されたシステムアカウントとグループを管理します。ドライバは、接続されたシステムへのソフトウェアインタフェースです。

Identity Manager ドライバは、Identity Manager Server アーキテクチャの一部として実行されます。ドライバは、ネイティブエンドポイントタイプシステムテクノロジへのゲートウェイとして機能します。たとえば、Active Directory Services を実行しているコンピュータを管理できるのは、Active Directory ドライバが、Identity Manager サーバまたは Identity Manager サーバが通信できるターゲットアプリケーションサーバにインストールされている場合のみです。ドライバは、接続されたシステムに存在するオブジェクトを管理します。管理対象オブジェクトには、アカウント、グループ、およびオプションでエンドポイントタイプ固有のオブジェクトが含まれます。

ドライバは、Identity Manager エンジンのアクションを、「Microsoft Exchange 接続システムでの新しい電子メールアカウントの作成」などの接続システムでの変更に変換します。Identity Manager で設定されているすべてのドライバにはイベントキャッシュファイル (TAO ファイル) が関連付けられています。イベントは、ドライバによって処理される前に、キャッシュファイルにキャッシュされます。デフォルトでは、キャッシュファイルは Identity Vault の DIB (Data Information Base) ディレクトリに配置されます。

Identity Manager には、さまざまなタイプの接続システムとの接続を管理するためにいくつかのドライバ (Java、ネイティブ、.NET) が組み込まれています。また、Identity Manager には、カスタムドライバを開発する機能も用意されています。これにより、自家製のアプリケーションや、テクノロジインタフェースがなくすぐに使用できるドライバを利用できないリポジトリなど、さまざまな他のシステムとのデータ同期が可能になります。

リモートローダ

ドライバは、Identity Manager Server にローカルでインストールするか、リモートローダを使用してインストールできます。リモートローダはドライバをロードし、リモートサーバにインストールされているドライバの代わりに Identity Manager エンジンと通信します。アプリケーションを Identity Manager エンジンと同じサーバで実行する場合、そのサーバにドライバをインストールできます。一方、アプリケーションを Identity Manager エンジンと同じサーバで実行しない場合は、ドライバをアプリケーションサーバにインストールする必要があります。ご使用の環境のワークロードを軽減したり、設定を容易にしたりする場合、リモートローダを Tomcat および Identity Manager サーバとは別のサーバにインストールできます。リモートローダの詳細については、『NetIQ Identity Manager Driver Administration Guide』の「リモートローダを使用するタイミングの決定」を参照してください。

[Identity Manager リモートローダサーバ] インストールオプションを使用して、リモートローダサービスとドライバインスタンスをリモートローダにインストールします。

ファンアウトエージェント

Identity Manager ファンアウトエージェントは、Java Database Connectivity (JDBC) ファンアウトドライバが複数の JDBC ファンアウトドライバインスタンスを作成するために使用するインストールコンポーネントです。ファンアウトドライバは、最小限の手間で複数のデータベースにユーザ、グループ、およびパスワードをプロビジョニングします。これにより、Identity Manager 管理者が同じポリシーを使用して複数の JDBC ドライバを設定し、同じタイプの複数のデータベースをプロビジョニングする必要がなくなります。ユーザアカウントを一元管理し、必要に応じて自動的に作成、設定、保守、および削除することができます。詳細については、『NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/netiq-identity-manager-for-jdbc-fan-out-driver-implementation-guide.html)』を参照してください。

ファンアウトエージェントをインストールするには、インストールプログラムの **[Identity Manager ファンアウトエージェント]** インストールオプションを使用します。

Identity Console

NetIQ Identity Console は、インターネットと Web ブラウザを介してどこからでもネットワーク管理用のユーティリティにアクセスできる、最先端の Web ベースの管理コンソールです。アクセスは仮想化され、セキュリティ保護され、カスタマイズされます。Identity Console を使用すると、管理タスクを簡単に分散化できます。Identity Console は iManager に替わる製品です。

Identity Console では Designer と同様のタスクを実行できるほか、システムのヘルスも監視できます。NetIQ では、管理タスクに Identity Console を使用することをお勧めします。Designer はパッケージへの変更、モデリング、および展開前のテストを必要とする設定タスクに使用してください。

Identity Console の詳細については、『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。

Identity アプリケーションのコンポーネント

Advanced Edition のインストールに必要

Identity アプリケーションは、ブラウザベースの複数の Web アプリケーションが相互接続されたセットです。これにより、組織では、ユーザが利用できるさまざまな役割とリソースに関連付けられたユーザアカウントおよび許可を管理することができます。役割の要求やパスワードの変更など、ユーザに対してセルフサービスサポートを提供するように Identity アプリケーションを設定することができます。役割とリソースの管理および割り当ての効率が向上するようにワークフローを設定することも可能です。Identity アプリケーションは、管理コンソール (管理タスク用)、ユーザコンソール (ダッシュボード)、およびこれらのタスクの実行に役立つ REST サービスで構成されています。

注 : Identity アプリケーションをインストールする前に、Identity Manager エンジンをインストールしておく必要があります。

Identity アプリケーションコンポーネントをインストールするには、インストールプログラムの **[Identity アプリケーション]** インストールオプションを使用します。

Identity アプリケーションインストールは、以下のコンポーネントで構成されています。

ユーザアプリケーション

ユーザアプリケーションはブラウザベースの Web アプリケーションで、さまざまな識別情報セルフサービスタスクと役割プロビジョニングタスクを実行できます。製品の以前のバージョンでユーザアプリケーションインタフェースを使用して実行されたタスクの一部は、管理コンソールとユーザコンソールを含む新しいユーザインタフェースに移動されました。ユーザアプリケーションは、新しいユーザインタフェースにはまだ存在しない機能の一部を引き続き提供します。詳細については、『NetIQ Identity Manager - Identity アプリケーションに対する管理者ガイド』を参照してください。

認証サービス

認証サービスは、Identity アプリケーション機能へのアクセスを提供します。Identity Manager でのシングルサインオンアクセスの使用の詳細については、『NetIQ Identity Manager - Administrator's Guide to the Identity Applications』を参照してください。

認証サービスは、NetIQ One Single Sign-On Provider (OSP) コンポーネントによって提供されます。Identity アプリケーションでは、OSP のローカルインストールが必要です。OSP は、Identity アプリケーションとともに自動的にインストールされます。

Self-Service Password Reset

セルフサービスパスワード管理サービスは、セルフサービスパスワード管理へのアクセスを提供します。Identity アプリケーションの NetIQ SSPR (Self Service Password Reset) は、識別情報アプリケーションにアクセスできるユーザが、管理者の助けを借りずに自分でパスワードをリセットできる機能です。

Identity アプリケーションインストールプロセスによって SSPR がデフォルトで有効になります。ただし、展開で SSPR が必要な場合、または Standard Edition をインストールする場合は、SSPR を別のコンピュータにインストールすることを選択できます。SSPR を Advanced Edition で別のコンピュータにインストールする場合、両方のコンポーネントのインストールを手動または ConfigUpdate ユーティリティを使用して完了した後、Identity アプリケーション環境設定ファイル (`ism-configuration.properties`) でパスワード管理設定を定義する必要があります。

Web アプリケーションサーバ

アプリケーションサーバは、Identity アプリケーションコンポーネントが実行されるランタイムフレームワークを提供します。Identity アプリケーションは、WAR (Web Application Resource または Web application ARchive) ファイルとしてパッケージ化されています。インストールプロセスにより、WAR ファイルをアプリケーションサーバに展開することができます。アプリケーションサーバは Java™ 仮想マシンを実行し、アプリケーションコードのランタイム環境を提供します。次の WAR ファイルが、Identity アプリケーションのコンポーネントの URL に適用されます。

- 「IDMProv」 (ユーザアプリケーションへのアプリケーションプログラミングインタフェース (API) 用)
- 「idmdash」 (ダッシュボードおよび Identity Applications 管理インタフェース用)

ユーザが `idmdash` アプリケーションを操作するとき、このアプリケーションは基礎となる `IDMProv.war` ファイルに対してクエリを実行し、ユーザの情報を取得します。`IDMProv.war` は REST API および SOAP API を公開します。これらの API では、`idmdash` にユーザインタフェースを提供する情報が含まれています。

Identity アプリケーションは、インストールキットに含まれている Apache Tomcat アプリケーションサーバで実行されます。Tomcat アプリケーションサーバをサポートするために、インストールプログラムはサポートされているバージョンの JRE と Apache ActiveMQ をインストールします。

Identity Applications データベース

Identity アプリケーションデータベースは、ローカライズされたラベル、エンタイトルメント値、電子メールサーバ設定など、Identity アプリケーションの環境設定データを維持します。また、ワークフローエンジンに必要なワークフロー状態データも格納します。Identity アプリケーションでサポートされているデータベースは、PostgreSQL、Oracle、および Microsoft SQL Server です。

Identity アプリケーションインストールプログラムは、Identity アプリケーションのデフォルトデータベースとして機能する、サポートされているバージョンの PostgreSQL データベースを自動的にインストールします。PostgreSQL をデータベースとして使用しない場合は、Identity アプリケーションを使用してサポートされているバージョンの Oracle または MS SQL データベースを設定できます。Identity アプリケーションには、データベースと通信するために Java Database Connectivity ドライバ (JDBC タイプ 4 ドライバ) が必要です。インストールプログラムは、データベースの JDBC ドライバの場所と名前の入力を求めます。したがって、Identity アプリケーションのインストールを開始する前に、データベースインストールディレクトリからこの JDBC ドライバを取得する必要があります。Identity Reporting でサポートされているデータベースは、PostgreSQL、Oracle、および MS SQL です。

- PostgreSQL データベースの場合、ドライバは Identity Manager インストールプログラムにバンドルされています。
- Oracle データベースの場合、Oracle Web サイトからドライバをダウンロードできます。
- Microsoft SQL Server データベースの場合、Microsoft Web サイトからドライバをダウンロードします。

データベースは、Identity アプリケーションサーバまたはリモートコンピュータにローカルに配置できます。リモートデータベースを使用する場合は、データベースへの接続を設定する必要があります。

Identity Applications 用のドライバ

Identity アプリケーションコンポーネントには、次のドライバが必要です。

ユーザアプリケーションドライバ

設定情報を格納し、識別ボールトで変更が行われた場合に Identity アプリケーションに通知します。識別ボールト内のイベントでワークフローをトリガできるようドライバを設定できます。このドライバから、ワークフローのプロビジョニングアクティビティの成否をユーザアプリケーションに通知することもできます。これにより、ユーザは要求の最終的なステータスを参照することができます。

役割とリソースのサービスドライバ

すべての役割の割り当てを管理し、承認を必要とする役割の割り当て要求のワークフローを開始し、グループまたはコンテナメンバーシップに従って間接的な役割の割り当てを維持します。このドライバは、役割のメンバーシップに基づいてユーザのエンタイトルメントを付与および取り消しして、完了した要求のクリーンアップ手順を実行します。このドライバは、役割要求に加えてリソース要求も維持します。

インストールプログラムの [Identity アプリケーション] インストールオプションは、ユーザアプリケーションドライバおよび役割とリソースのサービスドライバを Identity Vault に展開します。

Identity Reporting のコンポーネント

(オプション) このコンポーネントはレポーティング機能を実装する予定がある場合にのみ、インストールしてください

Identity Reporting では、お客様のユーザのエンタイトルメントに関する完全なビューが提供され、組織内で識別情報に対して付与された権限や許可の過去および現在の状況を確認するのに必要な知識が得られます。Identity Manager は、Identity Vault および接続システムから収集された情報を含む、Identity Manager 環境のステータスを監視するために使用できる事前定義されたレポートを提供します。Identity Manager に用意されているレポートを使用するには、Identity Manager に含まれている Identity Reporting をインストールします。Identity Reporting には、カスタムレポートの作成プロセスを容易にするレポートパッケージツールも含まれています。Identity Reporting のユーザインタフェースを使用すると、パフォーマンスを最適化するために、レポートを混雑していない時間帯に実行するようスケジュールするのが楽になります。Identity Reporting の詳細については、『Administrator Guide to NetIQ Identity Reporting』を参照してください。

注 : Identity Reporting を Advanced Edition にインストールする前に、Identity アプリケーションをインストールする必要があります。

Identity Reporting インストールは、以下のコンポーネントで構成されています。

Identity Reporting

レポーティングサービスを呼び出してレポートを生成するブラウザベースのアプリケーション。レポーティングサービスは、すべてのレポート管理情報 (レポート定義やスケジュールなど)、データベースビュー、およびレポーティングに必要な設定情報を格納する、Identity Reporting リポジトリ (識別情報ウェアハウス) からレポートを生成するために必要なデータを取得します。

認証サービス

認証サービスは OSP コンポーネントによって提供されます。詳細については、22 ページの「認証サービス」を参照してください。

注 : OSP は、Identity Reporting とともに自動的にインストールされます。ただし、Advanced Edition のインストールでは、Identity Reporting は Identity Applications でインストールされるのと同じ認証サービスを使用できます。同じ認証サービスを使用する場合は、Identity Reporting 設定時に認証設定を指定する必要があります。

Self-Service Password Reset

セルフサービスパスワード管理サービスは、セルフサービスパスワード管理へのアクセスを提供します。詳細については、22 ページの 「Self-Service Password Reset」 を参照してください。

Identity Reporting データベース

Identity Reporting データベース (識別情報ウェアハウス) は、Identity Vault および組織内の接続システムの実際の状態と望ましい状態に関する情報を保存します。この情報からレポートを生成して、ユーザや役割などのオブジェクト間の関係を表示することができます。データベースは、Identity Reporting サーバまたはリモートコンピュータにローカルに配置できます。Identity Manager は、データソースを使用してデータベースに接続します。Identity Reporting には、データベースと通信するために Java Database Connectivity ドライバ (JDBC タイプ 4 ドライバ) が必要です。JDBC ドライバにより、Identity Reporting サーバはデータソースと通信できます。Identity Reporting でサポートされているデータベースは、PostgreSQL、Oracle、および Microsoft SQL です。

- PostgreSQL データベースの場合、このドライバは Identity Manager インストールプログラムにバンドルされています。
- Oracle データベースの場合、Oracle Web サイトからドライバをダウンロードできます。
- Microsoft SQL Server データベースの場合、Microsoft Web サイトからドライバをダウンロードします。

注 : Identity Reporting コンポーネントをインストールする前に、Identity Manager Server をインストールする必要があります。

Web アプリケーションサーバ

アプリケーションサーバは、Identity Reporting コンポーネントが実行されるランタイムフレームワークを提供します。次の WAR ファイルが、Identity Reporting のコンポーネントの URL に適用されます。

- **IDMRPT** (Identity Reporting アプリケーション / インタフェース用)
- **idmdcs** (Identity Manager データ収集サービス用)

ユーザが `IDMRPT` または `idmdcs` アプリケーションを操作するとき、これらのアプリケーションはレポーティングサービスに対してクエリを実行し、ユーザの情報を取得します。レポーティングサービスは、`IDMRPT` および `idmdcs` にユーザインタフェースを提供する情報が含まれる REST API を公開します。

Web アプリケーションサーバの詳細については、22 ページの 「Web アプリケーションサーバ」 を参照してください。

Identity Reporting 用のドライバ

Identity Reporting コンポーネントには、次のドライバが必要です。

Managed System Gateway Driver

識別ボールトに問い合わせて管理対象システムから次のタイプの情報を収集します。

- すべての管理対象システムのリスト
- 管理対象システムのすべてのアカウントのリスト
- エンタイトルメントの種類、値、割り当て、および管理対象システムのユーザアカウントプロファイル

データ収集サービスドライバ

データ収集サービスは、データ収集サービスドライバを使用して、識別ボールトに保存されているオブジェクト (アカウント、役割、リソース、グループ、チームメンバーシップなど) の変更をキャプチャします。このドライバは、自身をサービスに登録し、変更イベント (データの同期、追加、変更、および削除イベント) をサービスにプッシュします。

このサービスには、次の 3 つのサブサービスが含まれます。

- **レポートデータコレクタ** : プルデザインモデルを使用して、1 つ以上の識別ボールトデータソースからデータを取得します。収集は、一連の環境設定パラメータによって決定され、定期的に実行されます。データを収集するために、コレクタが Managed System Gateway Driver を呼び出します。
- **イベントドリブンデータコレクタ** : プッシュデザインモデルを使用して、データ収集サービスドライバが取得したイベントデータを収集します。
- **非管理対象アプリケーションデータコレクタ** : それぞれのアプリケーション専用に記述された REST エンドポイントを呼び出すことによって、1 つ以上の非管理対象アプリケーションからデータを取得します。非管理対象アプリケーションとは、識別ボールトに接続されていない企業内のアプリケーションのことです。

インストールプロセスの **[Identity Reporting]** インストールオプションは、Managed System Gateway Driver とデータ収集サービスドライバを Identity Vault に展開します。

Sentinel Log Management for Identity Governance and Administration

Sentinel Log Management for Identity Governance and Administration (IGA) は、脅威、リスク、およびポリシー関連の決定を行うための情報をエンタープライズ内の多くのソースから受信し、標準化し、優先順位を決定して表示する、セキュリティ情報およびイベント管理 (SIEM) ソリューションです。Sentinel Log Management for (IGA) は、Identity Reporting、Identity アプリケーション、および Identity Vault を含むいくつかの NetIQ 製品で実行されたアクションに関連するログイベントをキャプチャします。これらのイベントは、Identity Reporting リポジトリ (識別情報ウェアハウス) 内のパブリックスキーマに格納されます。

Identity Manager は、Sentinel Log Management for IGA 用に別個のインストールプログラム (`SentinelLogManagementForIGA8.6.1.0.tar.gz`) を提供します。

Identity Manager ツール

すべてのインストールに必要

Identity Manager には、ソリューションの実装、カスタマイズ、および保守を容易にする管理ツールのセットが含まれています。これらのツールの一部は Identity Manager とともにインストールされ、一部は個別にインストールする必要があります。

Designer for Identity Manager

Designer for Identity Manager (Designer) は、ネットワーク環境またはテスト環境における Identity Manager ソリューションの設計、テスト、ドキュメント化、および展開を支援します。Identity Manager プロジェクトをオフライン環境で設定してからライブシステムに展開できます。設計上の観点から、Designer は次のことに役立ちます。

- Identity Manager ソリューションを構成するすべてのコンポーネントをグラフィカルに表示し、それらがどのように相互作用しているかを確認する。
- テストソリューションの一部または全体を運用環境に展開する前に、Identity Manager 環境を変更およびテストして、想定どおりに動作しているかを確認します。

Designer は、設計情報とレイアウト情報を維持します。ボタンをクリックするだけで、好みのフォーマットで情報を印刷できます。さらに、エンタープライズレベルのプロジェクト作業を複数のチームで共有することもできます。

Identity Manager は、Designer 用に別個のインストールプログラムを提供します。

Identity Manager 用の Analyzer

Analyzer for Identity Manager (Analyzer) は、内部データ品質ポリシーへの準拠を支援するデータ分析、クレンジング、調整、およびレポーティングの各機能を提供します。Analyzer を使用すると、企業内に保存されているすべてのデータを分析、拡張、および制御できます。Analyzer には、次の機能があります。

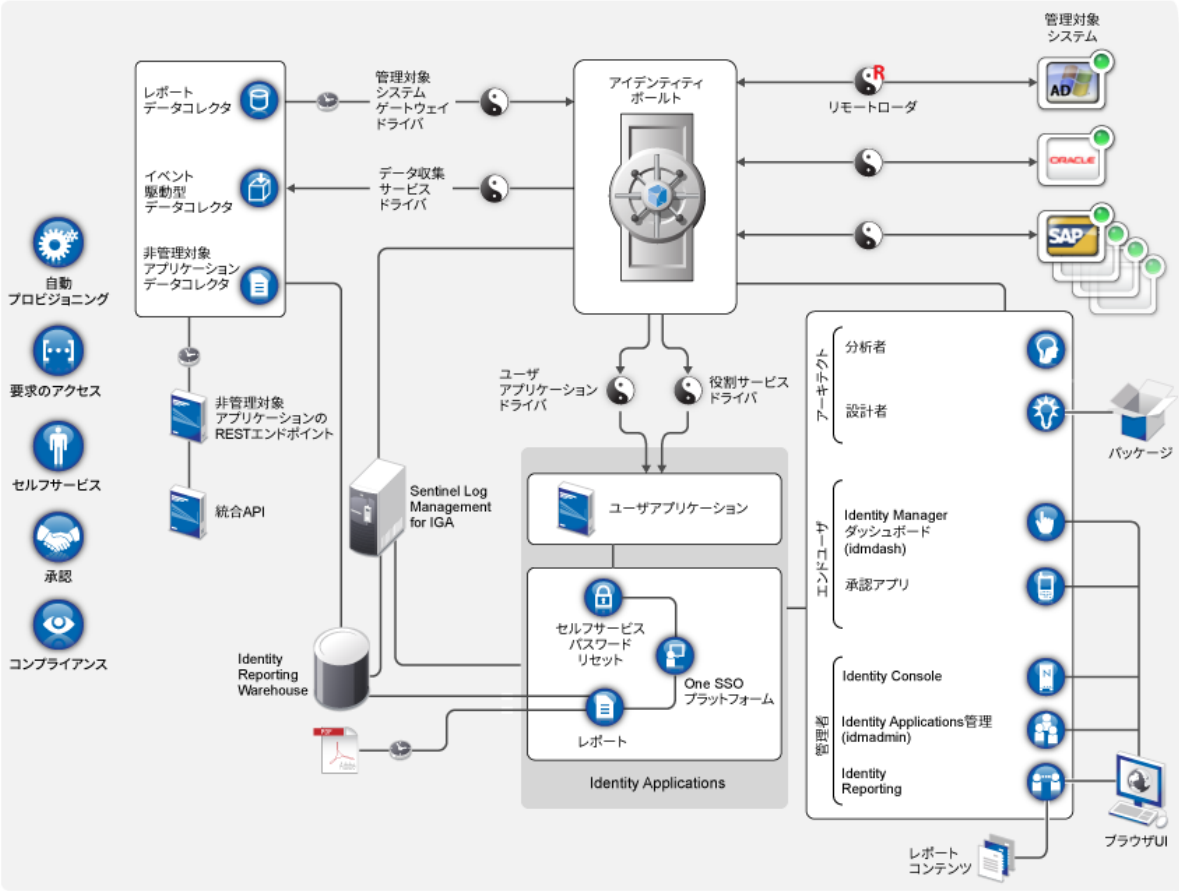
- Analyzer のスキーママップにより、アプリケーションのスキーマ属性を、Analyzer の基本スキーマの対応するスキーマ属性に関連付けます。これにより、データ分析およびクリーニング操作によって異種システム間の類似する値を適切に関連付けることができます。このために、Analyzer は Designer のスキーママッピング機能を利用します。
- Analysis Profile エディタを使用すると、1 つ以上のデータセットインスタンスを分析するためのプロファイルを設定できます。各分析プロファイルには 1 つ以上のメトリクスが含まれており、これらを基準にして属性値を評価することで、データが定義済みのデータフォーマット標準にどの程度準拠しているかを確認できます。
- Matching Profile エディタを使用して、1 つ以上のデータセットの値を比較できます。指定したデータセット内に重複する値がないかどうか、または 2 つのデータセット間で一致する値がないかどうかを確認できます。

Identity Manager は、Analyzer 用に別個のインストールプログラムを提供します。

重要 : Analyzer ツールは Identity Manager 4.9 の ISO に含まれません。Analyzer がすでにインストールされている場合は、必ずバージョン 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 とともに使用してください。

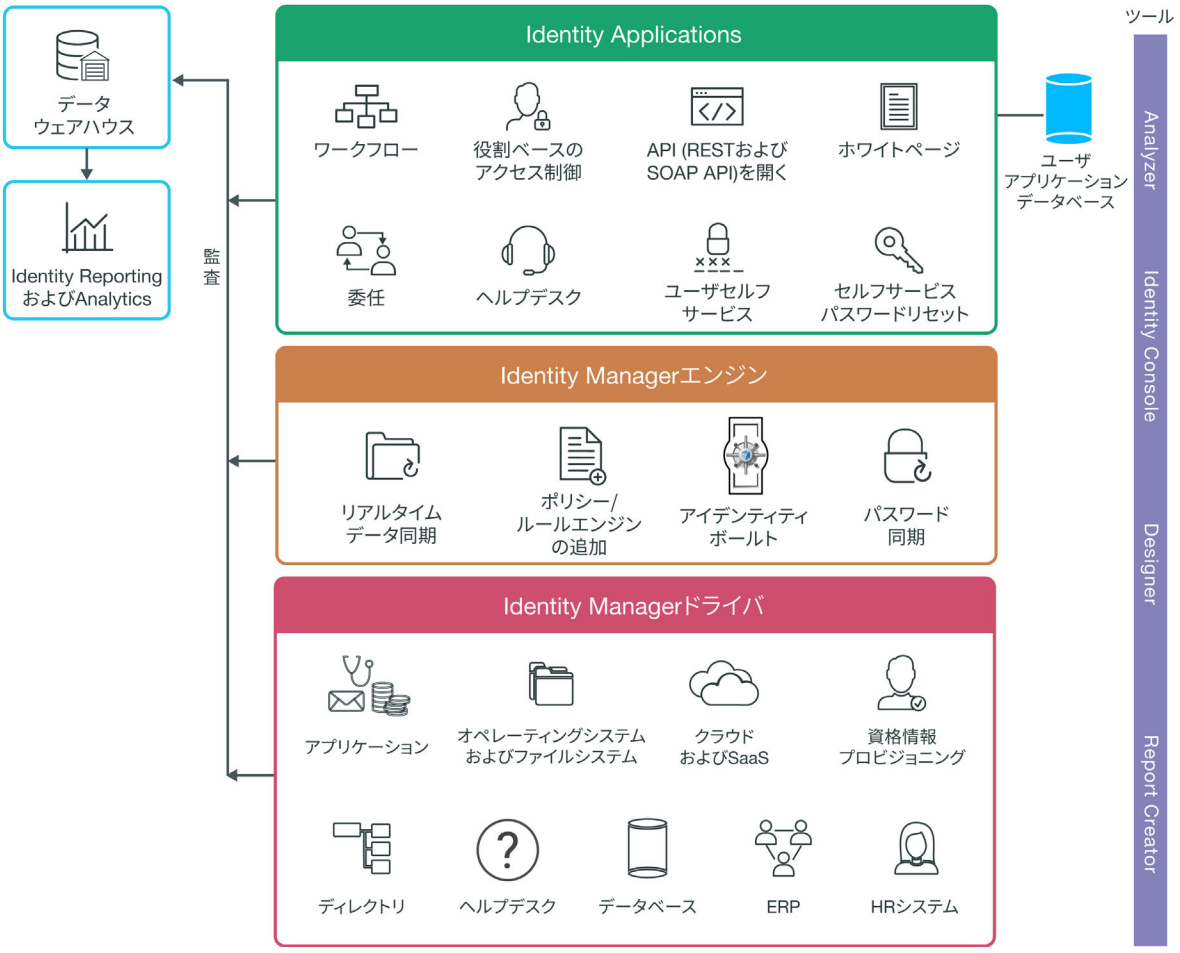
さまざまな Identity Manager コンポーネントの目的とインストール方法を理解したら、図 1-3 を参照して、コンポーネントが互いにどのように相互作用するかを理解してください。

図 1-3 *Identity Manager のコンポーネントの相互作用*



機能アーキテクチャ

次の図は、Identity Manager コンポーネントの基本的な機能アーキテクチャを示しています。この図は、考えられるすべての統合を網羅しているわけではありません。



可能な展開シナリオの詳細については、Identity Manager の展開オプションを参照してください。

Identity Manager の展開オプション

Identity Management ソリューションの物理環境を計画する場合は、次の表を参照してください。これらの展開ユースケースでは、Identity Management の物理アーキテクチャと、コンポーネント製品がどのように接続され、相互および他の製品と通信するかについての概要を提供します。Identity Management の機能アーキテクチャとコンポーネントの概要については、28 ページの 「機能アーキテクチャ」を参照してください。

展開オプション	概要
1 台のコンピュータ上の単一サーバ構成	最も基本的な展開構成では、1 台のコンピュータ上に Identity Manager サーバとその他の必要なアプリケーションが含まれます。ワークロードを満たすために、コンピュータに必要なメモリ、速度、および使用可能なディスク容量があることを確認する必要があります。これは、基本的な展開のユースケースであり、Proof-of-Concept (POC) およびデモ目的にのみ適しています。運用環境には適さない場合があります。
分散サーバ構成	この展開では、1 台のコンピュータ上に Identity Manager サーバがあり、1 台以上の追加のコンピュータに他のすべての必要なアプリケーションがあります。たとえば、Identity Applications、OSP、SSPR などのコンポーネントを、別のコンピュータで実行することができます。レポーティングサービスのコンポーネントをホストする追加のコンピュータを組み込んで、Sentinel Log Management for IGA コンポーネントを実行するためのシステム要件を満たすことができます。
高可用性の展開	高可用性とは、プライマリサーバで障害が発生した場合、または保守のために一時的にシャットダウンされた場合に、スタンバイサーバに自動的に切り替わる冗長動作です。Identity Manager は、以下のコンポーネントの高可用性環境へのインストールをサポートしています。 ◆ 識別ボールト ◆ Identity Manager エンジン ◆ リモートローダ ◆ Identity Reporting を除く、識別情報アプリケーション 一般的なクラスタ構成には、負荷分散と耐障害性のために Identity アプリケーションをホストする Tomcat Application Server ノードが含まれます。すべての通信はロードバランサを介してルーティングされています。すべてのノードは、Identity Vault および Identity アプリケーションデータベースの同じインスタンスと通信します。この構成はスケーラブルです。負荷を処理するノードの数を簡単に増やすことができます。

Identity Manager 展開のサンプル

Identity Manager を使用すると、ユーザの識別情報と、接続システム上のアプリケーションおよびアカウントへのアクセスを制御できます。必要な機能に基づいて、インストールする Identity Manager Edition を選択し、次にインストールするコンポーネントを決定します。次の表に、Identity Manager Advanced Edition および Identity Manager Standard Edition が提供する機能を示します。

機能	Advanced Edition	Standard Edition	インストールするコンポーネント
ルールベースの自動化されたユーザプロビジョニング	✓	✓	Identity Manager エンジンと Designer
リアルタイム Identity 同期	✓	✓	Identity Manager エンジンと Designer
パスワード管理とパスワードセルフサービス	✓	✓	Identity Manager エンジンと SSPR
ユニフォーム Identity 情報ツール (Analyzer)	✓	✓	Analyzer
REST API およびシングルサインオンのサポート	✓	✓ (限定サポートのみ)	Identity Manager エンジン、OSP、および Identity Reporting
現在の状態のレポーティング	✓	✓	Identity Manager エンジンおよび Identity Reporting
役割ベースのエンタープライズレベルのプロビジョニング	✓	✗	Identity Manager エンジンおよび Identity アプリケーション
ビジネスポリシー適用に対する自動化された承認ワークフロー	✓	✗	Identity Manager エンジン、Designer、および Identity アプリケーション
Identity アプリケーションの高度なセルフサービス	✓	✗	Identity Manager エンジンおよび Identity アプリケーション
簡単なリソースプロビジョニングのためのリソースモデルとカタログ	✓	✗	Identity Manager エンジンおよび Identity アプリケーション
履歴状態のレポーティング	✓	✗	Identity Manager エンジンおよび Identity Reporting
接続されたシステムのレポーティング	✓	✗	Identity Manager エンジンおよび Identity Reporting

機能	Advanced Edition	Standard Edition	インストールするコンポーネント
役割およびリソースの管理	✓	✗	Identity Manager エンジンおよび Identity アプリケーション

注 : すべての Identity Manager インストールにおいて、Identity Manager Server が中心的なコンポーネントとなります。Identity Manager のエディションに応じて、Identity Reporting のみ、または Identity Reporting と Identity アプリケーションの両方が Tomcat アプリケーションサーバにインストールされます。Identity Manager コンポーネント固有のインストーラを使用して、必要に応じて他のコンポーネントをインストールします。たとえば、Designer、Analyzer、または Sentinel Log Management for Identity Governance and Administration をインストールします。

さらに、Identity Manager をインストールする前に、実装の目標を確認し、高可用性やスケーラビリティなどの物理トポロジオプションに注意を払ってください。これにより、組織の要件に一致する構成を特定することができます。

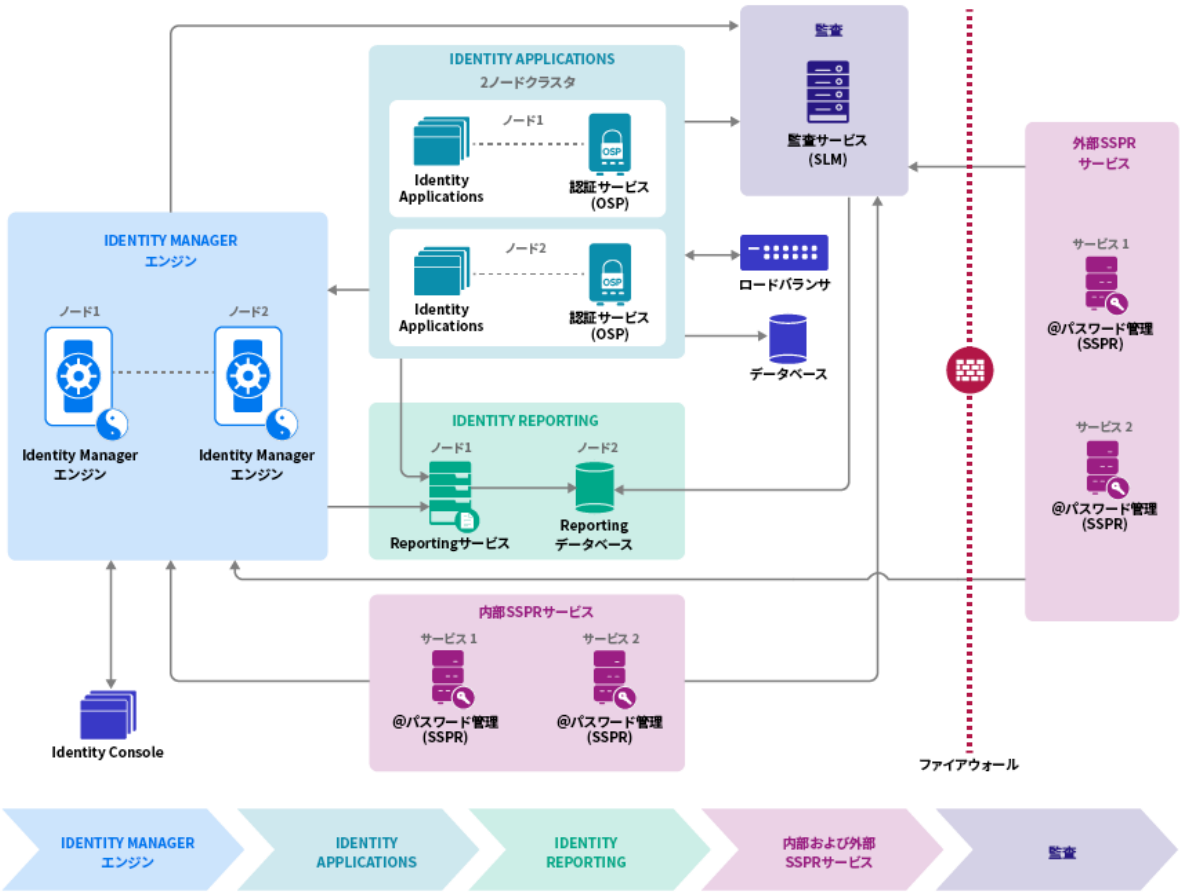
高可用性により、データ、アプリケーション、サービスなどの重要なネットワークリソースを効率的に管理できます。単一障害点を減らし、冗長コンポーネントを使用することにより、高可用性を実装することができます。同様に、識別情報管理コンポーネントの複数のインスタンスをロードバランサで接続すると、可用性の高い環境を実現できます。

このセクションでは、Advanced Edition および Standard Edition の実装を大まかに示す 2 つの例を説明します。これらを参考にして、実装の展開ダイアグラムを策定することができます。

Advanced Edition 展開のサンプル

図 1-4 は、Identity Manager Advanced Edition インストールの大まかな展開トポロジを示しています。

図 1-4 Advanced Edition 展開のサンプル



- Identity Manager Server コンポーネントとその基礎となるリポジトリ (Identity Vault) および Web 対応コンポーネント (Identity アプリケーションおよび Identity Reporting) は、イントラネットゾーンにインストールされます。この場合、ロードバランサにより、トラフィックが Identity アプリケーションコンポーネントにルーティングされます。この展開では、これらのコンポーネントがファイアウォールによってインターネットトラフィックから切り離されているので、セキュリティが強化されます。
- Identity Manager Server コンポーネントは、2 サーバ (プライマリ / セカンダリ) 構成を使用するように設定されています。仮想論理 IP アドレスはプライマリサーバでアクティブになります。プライマリサーバは、プライマリ (アクティブ) ノードとして機能し、もう一方のサーバがセカンダリノードとして機能します。プライマリサーバに障害が発生すると、論理 IP アドレスはセカンダリサーバに移動します。この場合、すべてのプロセスはセカンダリサーバで開始されます。論理 IP アドレスが移動し、他のすべてのプロセスが開始されると、セカンダリサーバにアクセスするアプリケーションプロセスで、一時的にサービスが失われる場合があります。すべてのコンポーネントは、常に同じ Identity Vault サーバを使用します。
- SSPR サービスはファイアウォールの内外で利用でき、組織のローカルユーザおよびモバイルユーザのパスワード管理のニーズに対処します。ファイアウォール内にインストールされたサービスは、ローカルパスワード管理のニーズに対処します。パスワードを忘れた場合、モバイルワーカーは VPN にアクセスできません。このため、内部に

配置された SSPR サービスにアクセスできなくなります。ファイアウォールの外側に配置されている SSPR サービスに直接アクセスして、パスワードを管理することができます。

- ユーザアプリケーションおよび認証サービス (OSP) は、負荷を処理し Identity アプリケーションのフェールオーバープロセスをサポートするために、クラスタに展開されます。クラスタノードは、別のコンピュータにインストールされている同じ Identity アプリケーションデータベースにアタッチされます。この展開では、クラスタにノードを追加できるので、スケーラビリティが向上します。クラスタ設定は、新たに追加されたノードにすぐに送信されます。ロードバランサは通常クラスタの一部で、クラスタ設定とともにフェールオーバーポリシーを理解しています。この設定では、すべてのクラスタノードはいつでもアクティブです。ロードバランサは複数のノードに負荷を分散し、各ノードのワークロードがほぼ同じになるようにします。ノードに障害が発生すると、そのノードに対して行われた要求がクラスタ内の存続しているノードに転送されます。このインストールはサイト内の高可用性ソリューションであるので、2 ノードのハードウェアベースのクラスタを使用して Identity アプリケーションコンポーネントの高可用性を実現し、ローカルのハードウェアおよびソフトウェアの障害からの保護を提供します。

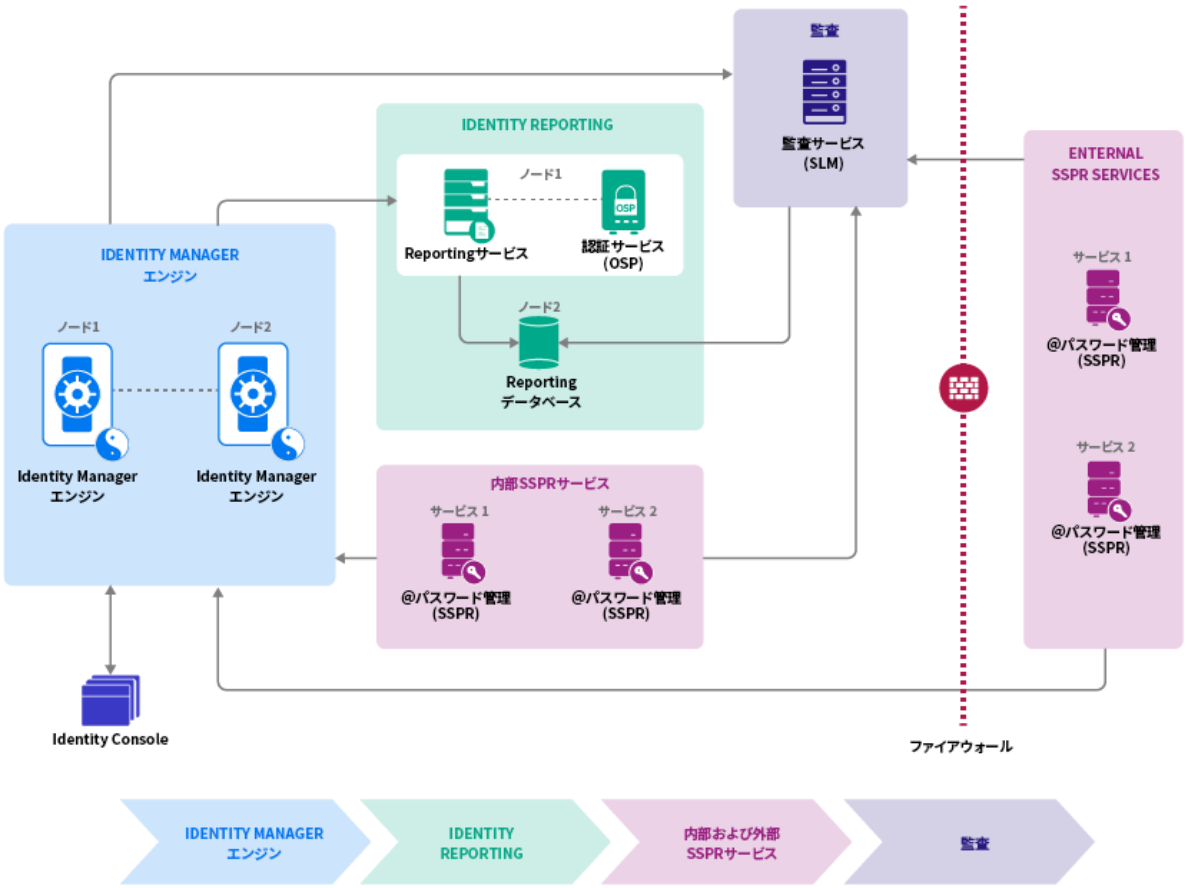
NetIQ ではこの設定をテスト済みであり、推奨しています。

注 : Identity Manager は、Identity Reporting コンポーネントのクラスタリングをサポートしていません。

Standard Edition 展開のサンプル

運用展開では、セキュリティポリシーにより、環境の高度な認証と保護を提供する認証サービスをパブリックネットワークに公開しないように指定することができます。図 1-5 は、Identity Manager Standard Edition インストールの大まかな展開トポロジを示しています。

図 1-5 Standard Edition 展開のサンプル



- Identity Manager Server コンポーネントとその基礎となるリポジトリ (Identity Vault) および Identity Reporting コンポーネントは、イントラネットゾーンにインストールされます。インターネット Web トラフィックは、保護を強化するためにファイアウォールの背後にインストールされている Web サーバを介して Identity Reporting コンポーネントにルーティングされます。この展開では、これらのコンポーネントがファイアウォールによってインターネットトラフィックから切り離されているので、セキュリティが強化されます。
- Identity Manager Server コンポーネントは、2 サーバ (プライマリ / セカンダリ) 構成を使用するように設定されています。仮想論理 IP アドレスはプライマリサーバでアクティブになります。プライマリサーバは、アクティブノードとして機能し、もう一方のサーバがセカンダリノードとして機能します。プライマリサーバに障害が発生すると、論理 IP アドレスはセカンダリサーバに移動します。この場合、すべてのプロセスはセカンダリサーバで開始されます。論理 IP アドレスが移動し、他のすべてのプロセスが開始されると、セカンダリサーバにアクセスするアプリケーションプロセスで、一時的にサービスが失われる場合があります。すべてのコンポーネントは、常に同じ Identity Vault サーバを使用します。
- SSPR サービスはファイアウォールの内外で利用でき、組織のローカルユーザおよびモバイルユーザのパスワード管理のニーズに対処します。ファイアウォール内にインストールされたサービスは、ローカルパスワード管理のニーズに対処します。パスワードを忘れた場合、モバイルワーカーは VPN にアクセスできません。このため、内部に

配置された SSPR サービスにアクセスできなくなります。ファイアウォールの外側に配置されている SSPR サービスに直接アクセスして、パスワードを管理することができます。

NetIQ ではこの設定をテスト済みであり、推奨しています。

注 : Identity Manager は、Identity Reporting コンポーネントのクラスタリングをサポートしていません。

II Identity Manager のインストールの計画

Identity Manager 実装の計画は、Identity Manager がユーザを管理する方法と、ビジネス目標を達成するために必要な機能に依存します。以下の点を考慮して、決定を行ってください。

- 識別情報をどのように管理するか。
- 自動プロビジョニングが必要か。
- ワークフローを使用して実装する必要があるのはどのビジネス要件か。

決定の結果により、要件に合わせて Identity Manager を実装する最適な方法が決まります。

大企業には Identity Manager を展開する前に計画を必要とする追加のタスクがあります。詳細については、『NetIQ Identity Manager Overview and Planning Guid』の「計画」セクションを参照してください。

2 インストールの計画

次の表に、実装する機能をサポートするためにインストールするコンポーネントを示します。これらのコンポーネントのインストール手順については、インストールセクションを参照してください。

機能	インストールするコンポーネント
社内ディレクトリのユーザの識別情報を管理する	Identity Manager サーバ
接続システムのアカウントをプロビジョニングする	Identity Manager サーバ 識別情報アプリケーション ユーザアプリケーションドライバ 役割とリソースのサービスドライバ Designer **注 :** Identity Manager ドライバのインストール手順については、Identity Manager ドライバのマニュアル Web サイトで、インストールするドライバのタイプに対応したドライバ実装ガイドを参照してください。
認証	Identity Manager サーバ 1 つの Single Sign-on プロバイダ
パスワード管理	Identity Manager サーバ Self Service Password Management
Identity Manager アクティビティに関するレポートを生成する	Identity Manager サーバ Identity Reporting 1 つの Single Sign-on プロバイダ

実装チェックリスト

Identity Manager の計画、インストール、および環境設定を実行する場合は、次に示すチェックリストを使用してください。

	チェックリストの項目
❑	1. 製品のアーキテクチャ情報を確認して Identity Manager コンポーネントについて学習します。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「How Identity Manager Works」を参照してください。

	チェックリストの項目
❑	2. Identity Manager のライセンス情報を確認して、Identity Manager の評価ライセンスとエンタープライズライセンスの、どちらのライセンスを使用する必要があるかを判断します。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Understanding Licensing and Activation」を参照してください。
❑	3. 実装する機能に基づいて、環境に適した展開のタイプを決定します。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Identity Manager Deployment Configurations」を参照してください。
❑	4. 優先言語でインストールプログラムを実行できるかどうかを判断します。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Understanding Identity Manager Localization」を参照してください。
❑	5. インストールするファイルを見つけます。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Identity Manager の入手場所」を参照してください。
❑	6. Identity Manager をインストールします。詳細については、65 ページのパート III「Identity Manager コンポーネントのインストールと設定」を参照してください。
❑	7. インストールされたコンポーネントを設定します。詳細については、83 ページの第 5 章「Identity Manager コンポーネントの設定」を参照してください。
❑	8. さまざまなコンポーネントが完全に機能するように、追加の設定手順を実行します。詳細については、101 ページの第 6 章「インストールを完了するための最終ステップ」を参照してください。

注 : クラスタ展開およびクラウド展開の場合、推奨される設定の詳細と要件を確認してください。

- 379 ページの 「高可用性のための Identity Manager の展開」
- 351 ページの 「AWS EC2 での Identity Manager の展開」

推奨されるインストールシナリオとサーバセットアップ

このセクションは、単一サーバまたは分散環境でのインストール順序とサーバのセットアップを決定するのに役立ちます。

- 40 ページの 「SLM for IGA をインストールするタイミングの決定」
- 41 ページの 「分散セットアップでのインストールに関する考慮事項」

SLM for IGA をインストールするタイミングの決定

Sentinel は、Identity Manager の優先する監査イベント転送先です。Identity Manager は、Sentinel Event Source Management (ESM) を使用して Sentinel リンクを設定することにより、Sentinel へのイベント転送機能を提供します。すでに Sentinel を監査に使用している場合、

または識別情報を追跡するための統合フレームワークとして使用している場合は、SLM for IGA をインストールする代わりに、既存の Sentinel を使用してイベントを監査することもできます。

既存の Sentinel サーバを再利用するか、Identity Manager に同梱されている SLM for IGA の新規インストールを実行するかに関係なく、Sentinel サーバを監査データのソースとして設定する必要があります。これを行うには、イベントを監査するために Identity Manager データ収集サービスページで Sentinel サーバに関するデータ同期ポリシーを作成します。詳細については、『*Administrator Guide to NetIQ Identity Reporting*』の「About the Data Sync Policies tab」を参照してください。

分散セットアップでのインストールに関する考慮事項

インストールを計画する場合は、次の考慮事項を確認してください。

コンポーネントの依存度

コンポーネント	独立したインストール	備考
Identity Manager エンジン	対応	
識別情報アプリケーション	対応	独自の OSP が必要です。Identity Applications と OSP は同じコンピュータにインストールされる必要があります。 **重要 :** Identity Manager は、リモートでインストールされた OSP をサポートしていません。このバージョンにアップグレードする場合は、Identity Applications のアップグレードでインストールされる OSP を使用してから、OSP 設定を既存の OSP サーバから OSP がインストールされている新しいサーバにコピーする必要があります。詳細については、234 ページの 「Identity Applications コンポーネントのアップグレード後のタスク」を参照してください。
Identity Reporting	対応	独自の OSP を持つことができます。インストーラは、Identity Reporting をインストールまたはアップグレードするためにローカルまたはリモートインストールされた OSP をサポートしています。

コンポーネント	独立したインストール	備考
OSP	非対応	インストーラは、Identity Applications のリモートインストールされた OSP をサポートしていません。同じコンピュータに OSP と Identity Applications をインストールする必要があります。 **重要 :** このバージョンにアップグレードする場合は、Identity Applications のアップグレードでインストールされる OSP を使用してから、OSP 設定を既存の OSP サーバから OSP がインストールされている新しいサーバにコピーする必要があります。詳細については、234 ページの 「Identity Applications コンポーネントのアップグレード後のタスク」を参照してください。
SSPR	対応	インストーラは、SSPR のスタンドアロンインストールおよびアップグレードをサポートしています。 Standard Edition で SSPR をインストールまたはアップグレードする場合、NetIQ では、スタンドアロンサーバに SSPR をインストールまたはアップグレードすることをお勧めします。つまり、SSPR を Identity Reporting と同じサーバにインストールしないでください。 **重要 :** Identity Applications と SSPR が異なるサーバに展開されている場合にこのバージョンにアップグレードし、SSPR がインストールされている新しいサーバに既存の SSPR 設定を復元する場合は、新しい SSPR サーバで ConfigUpdate ユーティリティを使用して SSPR 設定を変更してください。詳細については、234 ページの「Identity Applications コンポーネントのアップグレード後のタスク」を参照してください。
Identity Applications データベース	対応	
Reporting データベース	対応	
Sentinel Log Management for Identity Governance and Administration (Sentinel Log Management)	対応	

サーバのセットアップ

一般的な運用環境では、Identity Manager を 7 台以上のサーバと、クライアントワークステーションにインストールすることが考えられます。次に例を示します。

コンピュータのセットアップ	コンポーネントのセットアップ
All in One (デモ /POC セットアップにのみ推奨されます)	1 台のコンピュータにすべてのコンポーネント (Identity Manager エンジン、Identity Applications、Identity Reporting、OSP、SSPR、Identity Applications データベース、Reporting データベース) を、別のコンピュータに Sentinel Log management をインストールおよび設定します。
分散型セットアップ	
サーバ 1	◆ 識別ボールト ◆ Identity Manager エンジン
Server 2	Identity Applications および OSP (クラスタ化可能)
サーバ 3	Identity Reporting (OSP)
サーバ 4	SSPR
サーバ 5 および 6	次のコンポーネント用の Identity Manager データベース ◆ 識別情報アプリケーション ◆ Identity Reporting
Server 7	Sentinel Log Management

注 : 4.7 リリース以降、アイデンティティボールトの複数のインスタンスを持つサーバへの Identity Manager のインストールはサポートされなくなりました。

ハードウェア要件の決定

Identity Manager のインストールに必要なハードウェアは、次の 2 つの要因に左右されます。

- 実装する機能
- 展開のサイズ

以下の展開タイプは、展開のサイズを見積もるのに役立ちます。

展開のタイプ	ハードウェア要件
概念実証 (デモ)	開発環境でのデモまたは基本テストで使用する単一サーバ展開。
Basic	小規模から中規模の実装に適したマルチサーバ実装。 このタイプの実装では、Identity Manager Server およびそのコンポーネントを実行するために 1 台のサーバと、Identity Applications および Identity Reporting のコンポーネントを実行するために 2 台の追加サーバが必要です。

展開のタイプ	ハードウェア要件
中間	中規模の実装に適した高可用性実装。
大企業	フェールオーバー機能を提供する Identity Manager エンジンクラスタと、シングルサインオンアクセス (Windows 上の OSP) および負荷分散と耐障害性をサポートする別の Identity Applications クラスタと認証サービスを含む、高可用性実装。

システム要件ワークシート

推奨されるハードウェア、サポートされるオペレーティングシステム、およびサポートされる仮想環境については、Identity Manager 4.9 のシステム要件を参照してください。特定のリリースのシステム要件については、Identity Manager 4.9 のシステム要件で、リリースに付属のリリースノートを参照してください。

Identity Manager の実装は IT 環境のニーズに応じて異なるため、目的の環境に適った Identity Manager のアーキテクチャを最終決定する前に、NetIQ コンサルティングサービスまたは NetIQ Identity Manager パートナーにお問い合わせください。

SLES サーバへの Identity Manager のインストール

- Identity Manager をインストールする前に、必ず次の依存ライブラリまたは RPM をインストールしてください。
  - `libgcc*.x86_64`
  - `gettext.x86_64`
  - `unzip`
  - `bc`
  - `lsof`
  - `net-tools`
  - `openssl`
  - `pam-modules`
  - `coreutils`
  - `glibc-32bit-*.x86_64.rpm`
  - `libnsl-*.x86_64.rpm`
  - `libstdc++6-32bit`
  - `libz1-32bit`

  ここで * は、RPM の最新バージョンを示します。

  たとえば、`libnsl-<version>.x86_64.rpm` ファイルを追加するには、次のコマンドを入力します。

```
zypper in libsnl-<version>.x86_64.rpm
```

- (状況によって実行) SLES 15またはSLES 15 SPxにIdentity Managerをインストールする場合 (ここで、x は Identity Manager でサポートされるすべての SLES 15 バージョンを示します)、`libncurses5*`、`libncurses6*` および `insserv-compat*` RPM がインストールされていることを確認します。`*` 記号は、RPM の最新バージョンを示します。
- (オプション) SLES 上の Identity Manager に必要なパッケージをインストールするには、`install.sh` ファイルを作成し、そのファイルに次のコンテンツを追加します。

```
#!/bin/bash

zypper refresh

PKGS=" libstdc*.x86_64 libgcc*.x86_64 unzip bc bash coreutils
gettext.x86_64 glibc.x86_64 lsof net-tools openssl sed insserv ss* pam-
modules"

for PKG in $PKGS ;
do
    zypper -n install "$PKG"

done
```

`install.sh` ファイルを実行します。

注 : オペレーティングシステムベンダーからの継続的なサポートを確保するために、ご使用のオペレーティングシステムのサブスクリプションサービスから依存するパッケージを入手することをお勧めします。サブスクリプションサービスに加入していない場合は、http://rpmfind.net/linux などの Web サイトから最新のパッケージを見つけることができます。

RHEL サーバへの Identity Manager のインストール

Red Hat Enterprise Linux 以降のオペレーティングシステムを実行しているサーバに Identity Manager をインストールするには、サーバが一連の前提条件を満たしていることを確認します。

- 46 ページの 「前提条件」
- 47 ページの 「サーバに依存ライブラリがあることの確認」
- 47 ページの 「RHEL 9.x でのインストールメディア用のリポジトリの作成」
- 49 ページの 「RHEL 8.x でのインストールメディア用のリポジトリの作成」
- 50 ページの 「前提条件チェックの実行」

前提条件

NetIQ では、次の前提条件を確認することをお勧めします。

- `/etc/hosts` のエントリに、システムのホスト名に対するループバックアドレスの別名がある場合、この別名をホスト名または IP アドレスに変更する必要があります。つまり、`/etc/hosts` ファイルに次の 1 つ目の例のようなエントリがある場合は、2 つ目の例のように、正しいエントリに変更する必要があります。

  次の例では、いずれかのユーティリティが ndsd サーバに対して別名の解決を試みると、問題が発生します。

  ```
  <loopback IP address> test-system localhost.localdomain localhost
  ```

  次に、`/etc/hosts` 内の正しいエントリの例を示します。

  ```
  <loopback IP address> localhost.localdomain localhost
  <IP address> test-system
  ```

  サードパーティ製のツールやユーティリティが localhost を使用して別名を解決している場合は、localhost アドレスではなく、ホスト名または IP アドレスを使用して解決するように変更する必要があります。
- RHEL 8.x で SELinux (Security-Enhanced Linux) を設定している場合は、以下の操作を実行する必要があります。
  - Identity Manager エンジンをインストールするため、値を **[permissive]** に設定する。設定しない場合、エンジンのインストールは次のエラーで失敗します。`Identity Vault configuration failed with the exit code 10`

    たとえば、SELinux の値を permissive に設定するには、次の手順を実行します。

    1. `/etc/selinux/` ディレクトリにある `config` ファイルを変更します。
    2. **[SELINUX]** フィールドで、値を `permissive` に設定します。
    3. 変更内容を保存して、システムを再起動します。
  - Identity ReportingがIdentity Managerエンジンと Identity Applicationsとは異なるサーバ上にインストールされている場合は、SELinux を無効にする。無効にしない場合、Tomcat サービスが起動せず、Identity Reporting データベース設定が liquibase エラーを報告します。
- サーバに適切なライブラリをインストールします。詳細については、47 ページの「サーバに依存ライブラリがあることの確認」を参照してください。
- リモートローダをインストールするサーバ上の 2 つの環境変数、`$PATH` または `$JAVA_HOME` のいずれかに Java パスを設定していることを確認します。このアクションを実行してから `/RHEL-Prerequisite.sh script` を実行する必要があります。Java パスを設定するには、次のコマンドを実行します。

  ```
  export PATH=<java installed location>/bin:$PATH
  ```

  次に例を示します。`export PATH=/opt/netiq/common/jre/bin/:$PATH`

サーバに依存ライブラリがあることの確認

64 ビットプラットフォームでは、RHEL の必須ライブラリは、選択したインストール方法によって異なります。以下の順番で依存ライブラリまたは RPM をインストールします。

注 : `ksh` ファイルを追加するには、次のコマンドを入力できます。

```
yum -y install ksh
```

- `glibc-*.i686.rpm`
- `libgcc-*.i686.rpm`
- `libXtst-*.i686.rpm`
- `libXrender-*.i686.rpm`
- `libXi-*.i686.rpm`
- `unzip`
- `bc`
- `lsof`
- `net-tools`
- `libncurses`

注 : Identity Manager の場合、`/RHEL-Prerequisite.sh` スクリプトを編集し、発生するすべての `compat-libstdc++-33.x86_64.rpm` と `compat-libstdc++-33-*.i686.rpm` を削除できます。これらのパッケージは、Identity Manager のインストールには必要なくなりました。

RHEL 9.x でのインストールメディア用のリポジトリの作成

RHEL 9.x サーバでインストールメディア用のリポジトリが必要な場合、手動で作成できます。

注 : RHEL サーバに、適切なライブラリがインストールされていることが必要です。詳細については、47 ページの 「サーバに依存ライブラリがあることの確認」を参照してください。

インストールのためのリポジトリを設定するには :

1. ローカルサーバにマウントポイントを作成します。

   次に例を示します。

   ```
   mkdir -p /mnt/rhel9
   ```

2. RHEL 9.x インストール ISO をマウントします。

   ```
   mount -o loop <RHEL9.x.iso> /mnt/rhel9
   ```

   次に例を示します。

```
mount -o loop rhel-server-9.0-x86_64-dvd.iso mnt/rhel9
```

3 マウントされたディレクトリから /etc/yum.repos.d/ に media.repo ファイルをコピーし、必要な許可を設定します。

次に例を示します。

```
cp /mnt/rhel9/media.repo /etc/yum.repos.d/rhel9.repo


chmod 644 /etc/yum.repos.d/rhel9.repo
```

4 rhel9.repo ファイルを変更し、以下のコンテンツを追加します。

```
[InstallMedia]
name=Red Hat Enterprise Linux 9.0.0
mediaid=None
metadata_expire=-1
gpgcheck=0
cost=500
enabled=1
baseurl=file:///mnt/rhel9/BaseOS/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

5 32 ビットパッケージをインストールする場合は、/etc/yum.conf ファイルで exactarch パラメータの値を 1 から 0 に変更します。

6 次のコマンドを実行します。

```
yum clean all
```

7 (状況によって実行) DVD リポジトリからパッケージリストを取得する場合は、次のコマンドを実行します。

```
yum --noplugins list
```

8 yum-utils パッケージをインストールします。

```
yum install createrepo dnf-utils
```

9 RHEL 9.x 上の Identity Manager に必要なパッケージをインストールするには、install.sh ファイルを作成し、そのファイルに次のコンテンツを追加します。

注 : 重複する RPM に固有の警告が表示される場合は、適切な yum コマンドを使用して警告を手動で管理する必要があります。

```
#!/bin/bash
yum clean all
yum repolist
yum makecache

PKGS="ksh gettext.x86_64 libXrender.i686 libXau.i686 libxcb.i686
libX11.i686 libXext.i686 libXi.i686 libXtst.i686 glibc-*.i686.rpm
libstdc++.x86_64 libgcc-*.i686.rpm unzip bc lsof net-tools"

for PKG in $PKGS;
do
yum -y install "$PKG"
done
```

10 次のパッケージをインストールします。

```
yum install libgcc*.i686 libnsl* libnsl*.i686 libncurses*
```

11 install.sh ファイルを実行します。

12 前提条件が満たされていることを確認するには、50 ページの 「前提条件チェックの実行」に記載されているスクリプトを実行します。

13 Identity Manager 4.9 をインストールします。

RHEL 8.x でのインストールメディア用のリポジトリの作成

RHEL 8.x サーバでインストールメディア用のリポジトリが必要な場合、手動で作成できます。

注 : RHEL サーバに、適切なライブラリがインストールされていることが必要です。詳細については、47 ページの 「サーバに依存ライブラリがあることの確認」を参照してください。

インストールのためのリポジトリを設定するには :

1 ローカルサーバにマウントポイントを作成します。

次に例を示します。

```
mkdir -p /mnt/rhel8
```

2 RHEL 8 インストール ISO をマウントします。

```
mount -o loop rhel-server-8.0-x86_64-dvd.iso mnt/rhel8
```

3 マウントされたディレクトリから /etc/yum.repos.d/ に media.repo ファイルをコピーし、必要な許可を設定します。

次に例を示します。

```
cp /mnt/rhel8/media.repo /etc/yum.repos.d/rhel8.repo


chmod 644 /etc/yum.repos.d/rhel8.repo
```

4 rhel8.repo ファイルを変更し、以下のコンテンツを追加します。

```
[dvd-BaseOS]
name=DVD for RHEL8 - BaseOS
baseurl=file:///RHEL8/BaseOS
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[dvd-AppStream]
name=DVD for RHEL8 - AppStream
baseurl=file:///RHEL8/AppStream
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

5 32 ビットパッケージをインストールする場合は、/etc/yum.conf ファイルで exactarch パラメータの値を 1 から 0 に変更します。

6 次のコマンドを実行します。

```
yum clean all
```

7 (状況によって実行) DVD リポジトリからパッケージリストを取得する場合は、次のコマンドを実行します。

```
yum --noplugins list
```

8 yum-utils パッケージをインストールします。

```
yum install createrepo yum-utils
```

9 RHEL 8 上の Identity Manager に必要なパッケージをインストールするには、install.sh ファイルを作成し、そのファイルに次のコンテンツを追加します。

注 : 重複する RPM に固有の警告が表示される場合は、適切な yum コマンドを使用して警告を手動で管理する必要があります。

```
#!/bin/bash
yum clean all
yum repolist
yum makecache

PKGS="ksh gettext.x86_64 libXrender.i686 libXau.i686 libxcb.i686
libX11.i686 libXext.i686 libXi.i686 libXtst.i686 glibc-*.i686.rpm
libstdc++.x86_64 libgcc-*.i686.rpm unzip bc lsof net-tools"

for PKG in $PKGS;
do
yum -y install "$PKG"
done
```

10 次のパッケージをインストールします。

```
yum install libgcc*.i686 libnsl* libnsl*.i686 libncurses*
```

11 install.sh ファイルを実行します。

12 前提条件が満たされていることを確認するには、50 ページの 「前提条件チェックの実行」に記載されているスクリプトを実行します。

13 Identity Manager 4.9 をインストールします。

前提条件チェックの実行

各 Identity Manager コンポーネントで満たされていない前提条件のレポートを生成できます。インストールキットのマウントディレクトリにある ./RHEL-Prerequisite.sh スクリプトを実行します。

3 Identity Manager コンポーネントのインストールに関する考慮事項

このセクションでは、Identity Manager コンポーネントのインストールに必要な前提条件、考慮事項、およびシステムセットアップについて説明します。

- 51 ページの 「インストールの順序」
- 52 ページの 「インストールおよび設定プロセスの理解」
- 53 ページの「Identity Manager エンジンコンポーネントおよびリモートローダのインストールに関する考慮事項」
- 55 ページの 「識別情報アプリケーションコンポーネントのインストールに関する考慮事項」
- 59 ページの 「Identity Reporting コンポーネントのインストールに関する考慮事項」
- 61 ページの 「Designer のインストールに関する考慮事項」
- 62 ページの 「Analyzer のインストールに関する考慮事項」
- 62 ページの 「SLM for IGA のインストールに関する考慮事項」
- 63 ページの 「Identity Manager コンポーネントが使用するポートの確認」

インストールの順序

一部のコンポーネントのインストールプログラムでは、以前にインストールしたコンポーネントに関する情報が必要になるため、コンポーネントは以下の順序でインストールする必要があります。

- Sentinel Log Management for Identity Governance and Administration
- Identity Manager エンジンのコンポーネント
- Identity Applications のコンポーネント (Advanced Edition の場合のみ)
- Identity Reporting のコンポーネント
- Identity Console
- Designer for Identity Manager
- Identity Manager 用の Analyzer

重要 : Analyzer ツールは Identity Manager 4.9 の ISO に含まれません。Analyzer がすでにインストールされている場合は、必ずバージョン 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 とともに使用してください。

コンポーネントをインストールする前に、各コンポーネントのインストールの前提条件と考慮事項を確認する必要があります。

インストールおよび設定プロセスの理解

インタラクティブインストール : インタラクティブインストールでは、インタラクティブモードで入力を指定する必要があります。Identity Manager は、個別のコンポーネントまたはコンポーネントのグループを 2 つの別個のフェーズでインストールおよび設定するためのスクリプト化されたインストールプログラムを提供します。インストールフェーズでは、コンポーネントがインストールされます。インストールスクリプト `install.sh` は、Identity Manager インストールパッケージの `.iso` イメージファイルのルートにあります。設定については、環境設定を参照してください。

***表 3-1** インストールオプション*

インストールオプション	インストールされるコンポーネント
Identity Manager エンジン	アイデンティティボールト、Identity Manager エンジン、および Identity Manager ドライバをインストールします。インストールプロセスでは、Azul Zulu JDK もインストールします。
Identity Manager リモートローダサーバ	リモートローダのリモートローダサービスとドライバインスタンスをインストールします。
Identity Manager 展開エージェント	JDBC 展開ドライバ用の展開エージェントをインストールします。詳細については、「Installing the Fanout Agent with Identity Manager 4.9 (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/how-to-install-fan-out-agent-with-identity-manager-4-7.html)」を参照してください。
Identity Applications	Identity Applications の基本フレームワークを提供するいくつかのコンポーネントをインストールします。 ◆ ユーザアプリケーション ◆ OSP ◆ SSPR ◆ Tomcat ◆ PostgreSQL データベース Tomcat アプリケーションサーバをサポートするために、インストールプログラムはサポートされているバージョンの JRE と Apache ActiveMQ をインストールします。 このインストールプロセスでは、ユーザアプリケーションドライバと役割およびリソースサービスドライバも Identity Vault に展開されます。

インストールオプション	インストールされるコンポーネント
Identity Reporting	Identity Reporting の基本フレームワークを提供するいくつかのコンポーネントをインストールします。 ◆ Identity Reporting ◆ Managed System Gateway driver (MSGW) ◆ データ収集サービスドライバ (DCS) ◆ OSP (Identity Applications とは異なるサーバにインストールした場合) ◆ Tomcat (Identity Applications とは異なるサーバにインストールした場合) ◆ PostgreSQL データベース (Identity Applications とは異なるサーバにインストールした場合) Tomcat アプリケーションサーバをサポートするために、インストールプログラムはサポートされているバージョンの JRE をインストールします。

注 : Identity Manager は、Designer、Analyzer、および Sentinel Log Management の個別のインストールプログラムを提供します。

サイレントインストール : インストーラは、インタラクティブモードでサイレントプロパティファイルを作成するオプションを提供します。プロパティファイルにインストールオプションを記録してから、このファイルを使用して、環境内の別のサーバ上でサイレントインストールを実行することができます。サイレントインストールプログラムは、このファイルから値を読み取ってインストールを実行します。コンポーネントに関する設定の詳細については、83 ページの 「環境設定パラメータの理解」を参照してください。

構成 : Identity Manager には次の 2 つの環境設定モードがあります。

- 一般的な環境設定
- カスタム環境設定

標準設定は、ほとんどの設定オプションでデフォルト設定が選択されます。カスタム設定では、要件に応じてカスタム値を指定できます。このオプションを使用して、ほとんどの設定を設定できます。

Identity Manager エンジンコンポーネントおよびリモートローダのインストールに関する考慮事項

- Identity Manager エンジンのインストールプロセスでは、インストールに次の最小スペースが必要です。

パス	コンポーネント	必要とされる安全な最小スペース
/opt	Identity Manager エンジン	3GB
/var	Identity Manager エンジン	100,000 オブジェクトの dib に対して 5 GB
/etc	Identity Manager エンジン	5MB

- リモートローダをインストールする前に、Identity Manager エンジンがインストールされていることを確認します。
- Identity Manager エンジンをインストールしたコンピュータと同じコンピュータにリモートローダをインストールできます。オペレーティングシステムが両方のコンポーネントをサポートしていることを確認してください。
- 管理対象システムと通信可能なサーバにリモートローダをインストールします。各管理対象システム用のドライバが関連 API によって利用できる必要があります。
- (状況によって実行) Identity Manager エンジンを非ルートユーザとしてインストールした場合、Linux アカウントドライバまたはリモートローダはインストールされません。これらのコンポーネントは別途インストールする必要があります。
- NetIQ では、Identity Manager エンジンまたはリモートローダ以外のサーバ上に `changelog` モジュールをインストールすることをお勧めします。
- デフォルト以外の展開コンテキスト (`o=system` 以外) を使用する予定の場合、Identity Manager エンジンをインストールする前にカスタム LDIF ファイルを作成することをお勧めします。カスタム LDIF ファイルの場所は設定プロセス中にプロンプト表示されます。

注 : インストールの前に OU 構造を作成していない場合は、ドライバセットが作成されるすべてのコンテナの階層を使用して、必要な OU 構造を定義できます。この構造と階層は、LDIF ファイルを使用して新しいドライバセットを設定するときに使用されます。

- インストールプロセスを開始する前に、すべての言語用の次のコマンドを実行することをお勧めします。

  ```
  export LC_ALL=<language>
  ```

  次に例を示します。

  ```
  export LC_ALL=zh_TW
  ```

識別情報アプリケーションコンポーネントのインストールに関する考慮事項

識別情報アプリケーションのインストールプロセスを開始する前に、その前提条件とコンピュータ要件をレビューすることをお勧めします。アプリケーションコンポーネントのインストール後の識別情報アプリケーション環境の設定の詳細については、『*NetIQ Identity Manager - Administrator's Guide to the Identity Applications*』を参照してください。

- 55 ページの 「インストールの考慮事項」
- 56 ページの 「データベースに関する検討事項」
- 57 ページの 「識別情報アプリケーションのデータベースの設定」

インストールの考慮事項

- 識別情報アプリケーションのインストールプロセスでは、コンポーネントをインストールするために次の最小スペースが必要です。
  - /opt - 5GB
  - /var - 100MB
- 識別情報アプリケーションでは、次の Identity Manager コンポーネントのサポートされるバージョンが必要です。
  - Identity Manager エンジン
  - リモートローダ
- (オプション) NetIQ は、インストール時に Secure Sockets Layer (SSL) プロトコルを有効にします。環境内の Identity Applications コンポーネント間の通信設定を変更するには、『*NetIQ Identity Manager - Administrator's Guide to the Identity Applications*』の「Configuring Security in the Identity Applications」を参照してください。
- 役割とリソースサービスドライバは jClient を使用するので、リモートローダと組み合わせて使用することはできません。
- 各ユーザアプリケーションインスタンスは 1 つのユーザコンテナのみ処理できます。たとえば、インスタンスに関連付けられているコンテナに対してのみユーザの追加、検索、およびクエリを実行できます。また、アプリケーションとコンテナの関係は永続的なものと見なされます。
- 分散環境では、Identity Applications サーバのキーストア (`idm.jks`) に Identity Applications として CN を使用した証明書が必要です。拡張 Java セキュリティの一環として、現在では Identity Applications では OSP と通信するためのトラステッド証明書が必要です。

データベースに関する検討事項

このデータベースには、識別情報アプリケーションのデータと環境設定情報が格納されます。

データベースインスタンスをインストールする前に、次の前提条件をレビューします。

- Tomcat で使用するデータベースを設定するには、必要な JDBC jar ファイルが含まれていることを確認する必要があります。識別情報アプリケーションは標準の JDBC コールを使用してデータベースのアクセスや更新を行います。識別情報アプリケーションは、JNDI ツリーにバインドされた JDBC データソースファイルを使用して、データベースへの接続を開きます。
- データベースを参照するデータソースファイルが既存である必要があります。ユーザアプリケーションのインストール プログラムは、データベースを参照する `server.xml` と `context.xml` に Tomcat のデータソースエントリを作成します。
- サポートされているバージョンの Oracle または Microsoft SQL Server データベースを使用している場合は、Identity Applications が正しく機能するように、Identity Applications (例 : `idmuserappdb`) データベースおよびワークフロー (例 : `igaworkflowdb`) データベースの 2 つのデータベースインスタンスを設定する必要があります。
- 次の情報を手元に用意します。
  - データベースサーバのホストとポート。
  - 作成するデータベースの名前。識別情報アプリケーションのデフォルトのデータベースは `idmuserappdb` です。
  - データベースのユーザ名およびパスワード。データベースのユーザ名は、管理者アカウントを表すか、データベースサーバでテーブルを作成できる十分な許可を持っている必要があります。ユーザアプリケーションのデフォルトの管理者は `idmadmin` です。
  - 使用しているデータベース用にデータベースベンダーから提供されるドライバ `.jar` ファイル。NetIQ では、サードパーティベンダーが提供するドライバ JAR ファイルを推奨します。
- データベースインスタンスは、ローカルコンピュータまたは接続されたサーバのどちらにも配置できます。
- データベース文字セットは Unicode エンコーディングを使用する必要があります。たとえば、UTF-8 は Unicode エンコード方式を使用する文字セットですが、Latin1 は Unicode エンコード方式を使用しません。文字セットの指定の詳細については、58 ページの 「文字セットの設定」または 57 ページの 「Oracle データベースの設定」を参照してください。
- リモートデータベースに接続する場合は、Identity Applications をインストールする前に、データベースを作成してください。リモート PostgreSQL データベースへの接続については、165 ページの 「リモート PostgreSQL データベースへの接続」を参照してください。
- データベースの大文字と小文字を区別した照合により、移行中に重複キーエラーが発生する場合があります。照合を確認して修正し、識別情報アプリケーションを再インストールします。

- (状況によって実行) 1つのデータベースインスタンスを監査目的とアイデンティティアプリケーションの両方で使用する場合、アイデンティティアプリケーションを実行する Tomcat をホストするサーバとは別の専用サーバにデータベースをインストールすることをお勧めします。
- (状況によって実行) 新しいバージョンの識別情報アプリケーションに移行する場合、移行前のインストールで使用していたデータベースと同じデータベースを使用する必要があります。
- MS SQL でサポートされる照合は SQL_Latin1_General_CP1_CI_AS のみです。

識別情報アプリケーションのデータベースの設定

識別情報アプリケーションのデータベースは、設定データの保存やワークフローアクティビティのデータの保存などのタスクをサポートします。アプリケーションをインストールする前に、データベースがインストールされ、設定されている必要があります。

デフォルトでは、インストールプロセスにより、識別情報アプリケーション用の PostgreSQL データベースがインストールされ、データベースを所有する `idmadmin` という名前の管理ユーザが作成されます。ただし、インストール時には識別情報アプリケーション用のデータベース内にスキーマは作成されません。スキーマ情報は識別情報アプリケーションのインストール時に追加されます。

Oracle データベースの設定

このセクションでは、ユーザアプリケーションで Oracle データベースを使用する場合の設定オプションについて説明します。

- 57 ページの 「データベースの互換性レベルの確認」
- 58 ページの 「文字セットの設定」
- 58 ページの 「管理者ユーザアカウントの設定」

データベースの互換性レベルの確認

Oracle の異なるリリースからのデータベースは、これらが同じ機能をサポートし、これらの機能が同様に実行される場合には互換性があります。互換性がない場合は、特定の機能または操作が想定されるように動作しない場合があります。たとえば、識別情報アプリケーションを展開できないスキーマの作成は失敗します。

データベースの互換性レベルを確認するには、次の手順を実行します。

1. データベースエンジンに接続します。
2. SQL Server データベースエンジンの適切なインスタンスに接続した後で、**[オブジェクトエクスプローラ]** で、サーバ名をクリックします。
3. **[データベース]** を展開します。さらに、データベースに応じて、ユーザデータベースを選択するか、**[システムデータベース]** を展開してシステムデータベースを選択します。
4. データベースを右クリックし、**[プロパティ]** をクリックします。

   [データベースのプロパティ] ダイアログボックスが開きます。

5. [ページの選択] ペインで、[オプション] をクリックします。

   現在の互換性レベルが [互換性レベル] リストボックスに表示されます。

6. [互換性レベル] を確認するには、クエリウィンドウで以下を入力し、[実行] をクリックします。

   ```
   SQL> SELECT name, value FROM v$parameter
   WHERE name = 'compatible';
   ```

   想定される出力は 12.2.x.x、18.x.x、または 19.x.x です。

注 : Oracle 19c は Identity Manager 4.8.1 以降でサポートされています。

文字セットの設定

ユーザアプリケーションデータベースは、Unicode エンコーディング文字セットを使用する必要があります。データベースを作成する際に AL32UTF8 を使用してこの文字セットを指定します。

サポートされている Oracle データベースに UTF-8 が設定されていることを確認するには、次のコマンドを実行します。

```
select * from nls_database_parameters;
```

データベースに UTF-8 が設定されていない場合、次の情報が表示されます。

```
NLS_CHARACTERSET
WE8MSWIN1252
```

設定されている場合、データベースに UTF-8 が設定されていることを示す次の情報が表示されます。

```
NLS_CHARACTERSET
AL32UTF8
```

文字セットの設定の詳細については、「Choosing an Oracle Database Character Set」を参照してください。

管理者ユーザアカウントの設定

ユーザアプリケーションを使用するには、Oracle データベースユーザアカウントが特定の特権を持っている必要があります。SQL Plus ユーティリティで、次のコマンドを入力します。

```
CREATE USER idmuser IDENTIFIED BY password;
GRANT CREATE SESSION TO idmuser;
GRANT CREATE CLUSTER TO idmuser;
GRANT CREATE PROCEDURE TO idmuser;
GRANT CREATE SEQUENCE TO idmuser;
GRANT CREATE TABLE TO idmuser;
GRANT CREATE TRIGGER TO idmuser;
ALTER USER idmuser quota 100M on USERS;
```

ここで、*idmuser* はユーザアカウントを表します。

注 : サポートされているサードパーティの JDBC jar を使用することをお勧めします。例 : `ojdbc8.jar`

SQL Server データベースの設定

このセクションでは、ユーザアプリケーションで SQL Server データベースを使用する場合の設定オプションについて説明します。

- 59 ページの 「文字セットの設定」
- 59 ページの 「管理者ユーザアカウントの設定」

文字セットの設定

SQL Server では、ユーザはデータベースの文字セットを指定できません。ユーザアプリケーションは SQL Server の文字データを NCHAR カラムタイプ (UTF-8 をサポート) で保存します。

管理者ユーザアカウントの設定

Microsoft SQL Server をインストールした後、SQL Server Management Studio などのアプリケーションを使用してデータベースとデータベースユーザを作成します。データベースユーザアカウントは、次の特権を持つ必要があります。

- `CREATE TABLE`
- `DELETE`
- `INSERT`
- `SELECT`
- `UPDATE`
- `REFERENCES`

注 : サポートされているサードパーティの JDBC jar を使用することをお勧めします。たとえば、`mssql-jdbc-7.2.2.jre11.jar` です。

Identity Reporting コンポーネントのインストールに関する考慮事項

このセクションでは、Identity Reporting のコンポーネントのインストールを準備するためのガイドが記載されています。Sentinel を監査イベントに使用できます。

インストールプロセスを開始する前に、次の情報を確認することをお勧めします。

- 60 ページの 「Identity Reporting の前提条件」
- 60 ページの 「Identity Reporting の監査イベントの識別」

Identity Reporting の前提条件

- このインストールプロセスには、次の最小スペース要件が必要です。
  - /opt - 5GB
  - /var - 2GB
  - /etc - 2GB
- インストールプロセスでは、サポートされていて設定済みのバージョンの次の Identity Manager コンポーネントが必要です。
  - ユーザアプリケーションドライバを含む識別情報アプリケーション (Advanced Edition のみに適用可能)
  - 別の Linux コンピュータにインストールされた Sentinel Log Management。
- インストールプロセスにより、Tomcat 用 `setenv.sh` ファイルの JRE マッピングのための `JAVA_OPTs` エントリまたは `CATALINA_OPTS` エントリが変更されます。
- Identity Reporting をクラスタ環境のサーバにインストールしないでください。
- リモートデータベースに接続する場合は、Identity Reporting をインストールする前に、データベースを作成してください。リモート PostgreSQL データベースへの接続については、165 ページの 「リモート PostgreSQL データベースへの接続」を参照してください。
- Oracle データベースに対してレポートを実行するには、サポートされているサードパーティの jar をコピーしておく必要があります。詳細については、159 ページの「Oracle データベースでのレポートの実行」を参照してください。
- レポーティング機能にアクセスできるようにするすべてのユーザにレポート管理者の役割を割り当てます。
- Identity Manager 環境のすべてのサーバが同時に設定されていることを確認します。サーバの時刻を同期していない場合、レポートによっては実行時に空になることがあります。たとえば、Identity Manager エンジンをホストしているサーバとウェアハウスをホストしているサーバでタイムスタンプが異なる場合、この問題により、新しいユーザに関連するデータが影響を受ける可能性があります。ユーザを作成してから変更すると、レポートにデータが取り込まれます。
- Reporting を設定するには、ホスト名を小文字で指定する必要があります。Identity Reporting 6.6.0 以降のバージョンでは、IP アドレスを使用して Reporting を設定できなくなりました。

Identity Reporting の監査イベントの識別

このセクションでは、Identity Manager レポートおよびカスタムレポートに必要な異なる監査イベントを識別する方法について説明します。すべてのレポートソースを解凍し、次のスクリプトを実行して、監査イベントを識別することができます。

```
find . -name *.jrxml -print0 |xargs -0 grep -H "'000[B3]" | perl -ne
'($file) = /^\.\/(.*?)\//;@a = /000[3B]..../g; foreach $a (@a) { print
"$file;$a\n"}' |sort -u
```

次のセクションでは、Identity Manager レポートおよびカスタムレポートのさまざまな監査イベントを識別して選択する方法について説明します。

イベント名	監査フラグ
認証およびパスワードの変更	**SSPR を使用した監査フラグの選択 :** [SSPR Configuration Editor] を起動し、[Audit Configuration (監査環境設定)] を選択して、次の監査フラグから選択します。 ◆ Authenticate ◆ パスワードの変更 ◆ パスワードのロック解除 ◆ パスワードの回復 ◆ 侵入者の試み ◆ 侵入者ロック ◆ 侵入者ロックユーザ **Identity Console を使用した監査フラグの選択 :** Identity Console にログインし、[Auditing (監査)] > [NCP Server (NCP サーバ)] > [Security Events (セキュリティイベント)] に移動し、次の監査フラグから選択します。 ◆ パスワードの変更 ◆ パスワードの確認 ◆ ログイン ◆ ログアウト
他のすべてのレポーティングイベント	[NetIQ Identity Manager UserApp] > [管理] > [ログ記録] > [監査サービスを有効にする] の順に移動します。

Designer のインストールに関する考慮事項

- RHEL プラットフォームに Designer をインストールするには、RHEL リポジトリを作成する必要があります。詳細については、45 ページの 「RHEL サーバへの Identity Manager のインストール」を参照してください。
- このリリースでは、以前のバージョンからの Designer のアップグレードをサポートしていません。したがって、以前のバージョンがインストールされている場合は、以前のバージョンをアンインストールしてから Designer 4.9 をインストールする必要があります。また、組み込みの自動更新機能とオフライン更新機能を使用して Designer 4.9 にアップグレードすることもできません。
- SLES または RHEL を実行しているコンピュータで、Designer をインストールする前に、GNU gettext ユーティリティ（`gettext`）をインストールします。これらのユーティリティにより、国際化および多言語のメッセージに対応するためのフレームワークが提供されます。

注 : オペレーティングシステムベンダーの Web サイトからの継続的なサポートを確保するために、ご使用のオペレーティングシステムのサブスクリプションサービスから依存するパッケージを入手することをお勧めします。サブスクリプションサービスに加入していない場合は、http://rpmfind.net/linux (http://rpmfind.net/linux) などの Web サイトから最新のパッケージを見つけることができます。

Analyzer のインストールに関する考慮事項

重要 : Analyzer ツールは Identity Manager 4.9 の ISO に含まれません。Analyzer がすでにインストールされている場合は、必ずバージョン 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 とともに使用してください。

- SLES 12 SP3 プラットフォームを実行しているコンピュータに Analyzer をインストールする前に、次のライブラリがインストールされていることを確認します。
  - `libswt3-gtk2`
  - `libxcomposite`
  - `libgdk_pixbuf`
  - `libgtk+-x11`
  - `gettext` (GNU gettext ユーティリティ)
- RHEL 8 以降のプラットフォームを実行しているコンピュータに Analyzer をインストールする前に、次のライブラリがインストールされていることを確認します。
  - `gtk2.i686.rpm`。たとえば、オペレーティングシステムベンダーの Web サイトからパッケージをダウンロードできます。
  - `gettext` (GNU gettext ユーティリティ)

  注 : オペレーティングシステムベンダーからの継続的なサポートを確保するために、ご使用のオペレーティングシステムのサブスクリプションサービスから依存するパッケージを入手することをお勧めします。サブスクリプションサービスに加入していない場合は、http://rpmfind.net/linux/ などの Web サイトから最新のパッケージを見つけることができます。
- Analyzer を実行しているコンピュータのビデオ解像度が 1024x768 (1280x1025 を推奨) であることを確認してください。

SLM for IGA のインストールに関する考慮事項

SLM for IGA サーバのオペレーティングシステムに、少なくとも SLES サーバか RHEL サーバのベースサーバコンポーネントが含まれている必要があります。Sentinel では、次の RPM の 64 ビットバージョンが必要です。

- `bash`
- `bc`

- coreutils
- gettext
- glibc
- grep
- libgcc
- libstdc
- lsof
- net-tools
- openssl
- python-libs
- sed
- zlib

詳細については、NetIQ Sentinel の技術情報 Web サイト (https://www.microfocus.com/documentation/sentinel/8.5/) を参照してください。

Identity Manager コンポーネントが使用するポートの確認

Identity Manager コンポーネントは、Identity Manager コンポーネント間の適切な通信のために異なるポートを使用します。

注 : デフォルトポートがすでに使用されている場合は、Identity Manager コンポーネント用に別のポートを指定してください。

ポート番号	コンポーネント	ポートの使用
389	識別ボールト	Identity Manager コンポーネントとの平文での LDAP 通信に使用されます。
465	Identity Reporting	SMTP メールサーバとの通信に使用されます。
524	識別ボールト	NetWare Core Protocol (NCP) 通信に使用されます。
636	識別ボールト	Identity Manager コンポーネントとの TLS/SSL による LDAP 通信に使用されます。
5432	識別情報アプリケーション	識別情報アプリケーションデータベースとの通信に使用されます。
7707	Identity Reporting	Managed System Gateway ドライバによって、識別ボールトとの通信に使用されます。

ポート番号	コンポーネント	ポートの使用
8000	リモートローダ	TCP/IP 通信のためにドライバインスタンスによって使用されます。 **注 :** リモートローダの各インスタンスには一意のポートが割り当てられる必要があります。
8005	識別情報アプリケーション	Tomcat によって、シャットダウンコマンドのリスンに使用されます。
8009	識別情報アプリケーション	Tomcat によって、HTTP ではなく AJP プロトコルを使用した Web コネクタとの通信に使用されます。
8028	識別ボールト	NCP 通信との平文での HTTP 通信に使用されます。
8030	識別ボールト	NCP 通信との HTTPS 通信に使用されます。
3000	Identity Applications	NGINX サービスによって使用されます。
8090	リモートローダ	リモートローダによって、リモートインタフェースシムからの TCP/IP 接続のリスンに使用されます。 **注 :** リモートローダの各インスタンスには一意のポートが割り当てられる必要があります。
8109	識別情報アプリケーション	統合インストールプロセスの使用時にのみ適用されます。 Tomcat によって、HTTP ではなく AJP プロトコルを使用した Web コネクタとの通信に使用されます。
8443	識別情報アプリケーション	Tomcat によって、HTTPS (SSL) 通信、または SSL 通信に対する要求のリダイレクトに使用されます。
8543	識別情報アプリケーション	*デフォルトではリスンしません。* TLS/SSL プロトコルを使用していないときに、Tomcat によって、SSL 転送を求める要求のリダイレクトに使用されます。
8600	フォームレンダラ	NGINX サービスによって使用されます。
9000	Identity Console	Identity Console によって使用されます。
5432	Identity Reporting	PostgreSQL データベース Sentinel で使用されます。
45654	ユーザアプリケーション	Tomcat がクラスタグループとともに実行されている場合、識別情報アプリケーションのデータベースがインストールされているサーバによって、通信のリスンに使用されます。

III Identity Manager コンポーネントのインストールと設定

このセクションでは、Identity Manager コンポーネントのインストールと設定のプロセスを順を追って説明します。詳細については、67 ページの第 4 章「Identity Manager のインストール」を参照してください。Identity Manager コンポーネントの設定手順については、83 ページの第 5 章「Identity Manager コンポーネントの設定」を参照してください。

Identity Manager コンポーネントがインストールされ、基本設定が完了したら、コンポーネントを完全に機能させるために追加の設定手順をいくつか実行する必要があります。詳細については、101 ページの第 6 章「インストールを完了するための最終ステップ」を参照してください。

4 Identity Manager のインストール

このセクションでは、Identity Manager コンポーネントをインストールするさまざまな方法について説明します。次の方法で Identity Manager コンポーネントをインストールすることができます。

- インタラクティブインストール
- サイレントインストール

対話型インストールの実行

1 ［Software and License (ソフトウェアおよびライセンス)］ページ (https://sld.microfocus.com/) から `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` をマウントします。

3 `.iso` ファイルのルートディレクトリから、次のコマンドを実行します。

```
./install.sh
```

4 使用許諾契約を読みます。

5 使用許諾契約の条項を確認し、「`y`」と入力します。

6 インストールする Identity Manager サーバのエディションを決定します。Advanced Edition の場合は「`y`」、Standard Edition の場合は「`n`」と入力します。

7 インストール可能なコンポーネントのリストから、必要なコンポーネントを選択します。

- エンジンをインストールするには、**[Identity Manager エンジン]** を選択します。
- リモートローダをインストールするには、**[Identity Manager リモートローダ]** を選択します。
- ファンアウトエージェントをインストールするには、**[Identity Manager ファンアウトエージェント]** を選択します。
- Identity Applications をインストールするには、**[Identity Applications]** を選択します。
- Identity Reporting をインストールするには、**[Identity Reporting]** を選択します。
- Identity Console をインストールするには、**[Identity Console]** を選択します。

8 (状況によって実行) インストールされたコンポーネントを設定します。詳細については、83 ページの第 5 章「Identity Manager コンポーネントの設定」を参照してください。

サイレントインストールの実行

1 ［Software and License (ソフトウェアおよびライセンス)］ページ (https://sld.microfocus.com/) から `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` をマウントします。

3 `.iso` のルートディレクトリから、`create_silent_props.sh` スクリプトを実行します。

4 「`y`」と入力して、ファイルの作成を確定します。

5 サイレントプロパティファイルのパスを指定します。

6 インストールする Identity Manager サーバのエディションを決定します。Advanced Edition の場合は「`y`」、Standard Edition の場合は「`n`」と入力します。

7 コンポーネントを標準モードとカスタムモードのどちらで設定するかを決定します。

8 インストール可能なコンポーネントのリストから、必要なコンポーネントを選択します。

- エンジンをインストールするには、**[Identity Manager エンジン]** を選択します。
- リモートローダをインストールするには、**[Identity Manager リモートローダ]** を選択します。
- ファンアウトエージェントをインストールするには、**[Identity Manager ファンアウトエージェント]** を選択します。
- Identity Applications をインストールするには、**[Identity Applications]** を選択します。
- Identity Reporting をインストールするには、**[Identity Reporting]** を選択します。
- Identity Console をインストールするには、**[Identity Console]** を選択します。

インストールされたコンポーネントを設定します。詳細については、83 ページの第 5 章「Identity Manager コンポーネントの設定」を参照してください。

9 次のコマンドを実行して、サイレントインストールを実行します。

`./install.sh -s -f <` *サイレントプロパティファイルの場所* `>`

次に例を示します。

`./install.sh -s -f /home/silent.properties`。ここで、`/home/silent.properties` は、サイレントプロパティファイルを格納した場所です。

root 以外のユーザによる Identity Manager エンジンのインストール

root 以外のユーザとして Identity Manager エンジンをインストールすると、Linux サーバのセキュリティを強化できます。アイデンティティボールトを root としてインストールした場合、Identity Manager エンジンを root 以外のユーザとしてインストールすることはできません。root 以外のユーザとしてエンジンをインストールする場合は、次の手順を実行する必要があります。

1. NICI がインストールされていることを確認します。詳細については、69 ページの「NICI のインストール」を参照してください。
2. アイデンティティボールトのルート以外のインストールを実行します。詳細については、69 ページの 「アイデンティティボールトの root 以外のインストールの実行」を参照してください。
3. Identity Manager エンジンの root 以外のインストールを実行します。詳細については、71 ページの 「エンジンの root 以外のインストールを実行」を参照してください。

NICI のインストール

アイデンティティボールトのインストールを続行する前に、NICI をインストール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、root 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。ただし、必要であれば、sudo を使用して別のアカウントにアクセス権限を委任し、そのアカウントを使用して NICI パッケージをインストールすることができます。

1 マウントした iso から、/IDVault/setup/ ディレクトリに移動します。

2 次のコマンドを実行します。

```
rpm -qa|grep nici64-3.2.0.0500-0.x86_64
```

3 NICI がサーバモードに設定されていることを確認します。次のコマンドを入力します。

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

これはアイデンティティボールトの設定プロセスが失敗しないようにするための必須の手順です。

アイデンティティボールトの root 以外のインストールの実行

このセクションでは、アイデンティティボールトをインストールするために tarball を使用する方法について説明します。ファイルを抽出すると、システムにより etc、opt、および var ディレクトリが作成されます。

1 識別ボールトをインストールするコンピュータに対する適切な権利を持つ sudo ユーザとしてログインします。

注 : カスタムインストールパスを指定する場合は、root ユーザとしてログインすることもできます。

2 マウントした iso から、/IDVault/ ディレクトリに移動します。

3 新しいディレクトリを作成して、そのディレクトリに eDir_NonRoot.tar.gz ファイルをコピーします。たとえば、/home/user/install/eDirectory です。

4 次のコマンドを使用してファイルを展開します。

```
tar -zxvf eDir_NonRoot.tar.gz
```

5 環境変数のパスを手動でエクスポートするには、以下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/
ndsmodules:
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
```

6 ndspath スクリプトを使用して、環境変数のパスをエクスポートするには、ユーティリティの前に ndspath スクリプトを指定する必要があります。次の手順に従います。

6a custom_location/eDirectory/opt ディレクトリから、次のコマンドを使用して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
```

6b 次のコマンドを使用して、現在のシェルのパスをエクスポートします。

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

6c ユーティリティを通常どおり実行します。

6d パスのエクスポート指示は、/etc/profile や ~/bashrc などのスクリプトの最後に追加します。

この手順により、ログイン時または新しいシェルのオープン時には常に、ユーティリティを直接起動できます。

7 以下のいずれかの方法を使用して、アイデンティティボールトを設定します。

- 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>]
```

ここで、-t はサーバの追加先のツリー名を示します。

-n は、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示します。

-a は、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示しします。

-s は、サーバ名を示します

-d は、データベースファイルが保存されているディレクトリパスを示します。

-m は、モジュール名を示します。

設定プロセス中に指定した値と同じ値を指定する必要があります。

次に例を示します。

```
ndsconfig new -t novell-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/
inst1/var --config-file /home/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、『NetIQ eDirectory インストールガイド』の「ndsmanage ユーティリティ」を参照してください。

エンジンの root 以外のインストールを実行

この方法を使用する場合、次のコンポーネントはインストールできません。

- **リモートローダ** : root 以外のユーザとしてリモートローダをインストールするには、Java リモートローダを使用します。詳細については、79 ページの 「Java リモートローダのインストール」を参照してください。
- **Linux アカウントドライバ** : 機能させるには root 権限が必要です。

注 : root 以外のユーザとして Identity Manager エンジンをインストールすると、インストールファイルは root 以外のユーザディレクトリに置かれます。たとえば、/home/user です。このユーザは root 以外です。インストールファイルは、Identity Manager を実行するために必要ではありません。インストール後にファイルを削除できます。

Identity Manager エンジンを非 root ユーザとしてインストールする

1 アイデンティティボールトのインストールに使用した root 以外のユーザとしてログインします。

このユーザアカウントには、root 以外でインストールしたアイデンティティボールトのディレクトリおよびファイルに対する書き込みアクセス権が必要です。

2 Identity_Manager_4.9.0_Linux.iso をマウントした場所に移動します。

3 マウント場所から、/IDM ディレクトリに移動します。

4 次のコマンドを実行します :

```
./idm-nonroot-install.sh
```

5 次の情報を使用して、インストールを完了します。

root 以外でインストールした eDirectory の基本ディレクトリ

root 以外でインストールした eDirectory があるディレクトリを指定します。たとえば、/home/user/install/eDirectory です。

eDirectory スキーマの拡張

これが eDirectory のこのインスタンスにインストールされている最初の Identity Manager サーバである場合、「Y」と入力してスキーマを拡張します。スキーマを拡張しないと、Identity Manager は機能しません。

root 以外でインストールした eDirectory によってホストされていて、root 以外のユーザが所有している eDirectory の各インスタンスのスキーマを拡張するようメッセージが表示されます。

スキーマを拡張するよう選択した場合、スキーマを拡張する権限を持つ eDirectory ユーザの完全識別名 (DN) を指定します。スキーマを拡張するには、ユーザにツリー全体に対するスーパバイザ権限が必要です。root 以外のユーザとしてスキーマを拡張する場合の詳細については、schema.log ファイルを参照してください。このファイルは、eDirectory の各インスタンスの data ディレクトリにあります。

インストール完了後、/opt/novell/eDirectory/bin/idm-install-schema プログラムを実行して、追加の eDirectory インスタンスのスキーマを拡張します。

6 インストールプロセスを完了するには、167 ページの 「root 以外のインストールの完了」に移動してください。

SSPR のインストール

インストーラには、SSPR を個別にインストールするオプションが用意されています。これは、Identity Applications と SSPR を別々のコンピュータにインストールする場合に役立ちます。

注 : Standard Edition をインストールする場合は、以下の手順を使用して SSPR をインストールする必要があります。デフォルトでは、Standard Edition を使用した場合 SSPR はインストールされません。

SSPR の対話型インストールの実行

1. NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。
2. ダウンロード済みの `.iso` をマウントします。
3. `.iso` ファイルのルートディレクトリから、`sspr` ディレクトリに移動します。
4. SSPR をインストールするには、次のコマンドを実行します。

   ```
   ./install.sh
   ```

5. 使用許諾契約を読みます。
6. 使用許諾契約の条項を確認し、「`y`」と入力します。
7. (状況によって実行) SSPR を設定します。詳細については、98 ページの 「SSPR の設定」を参照してください。

SSPR のサイレントインストールの実行

1. NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。
2. ダウンロード済みの `.iso` をマウントします。
3. `.iso` ファイルのルートディレクトリから、`sspr` ディレクトリに移動します。

   注 : 要件に応じて `sspr_silentinstall.properties` を編集します。

4. サイレントインストールを実行するには、次のコマンドを実行します。

   ```
   ./install.sh -s -f sspr_silentinstall.properties
   ```

5. (状況によって実行) SSPR を設定します。詳細については、98 ページの 「SSPR の設定」を参照してください。

リモートローダのインストール

Identity Manager には、スタンドアロンサーバにリモートローダをインストールするオプションが用意されています。Identity Manager エンジンとリモートローダを別々のコンピュータにインストールする場合、このオプションを使用します。

インタラクティブインストール

1 NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_RL_Linux.iso` をダウンロードします。
2 ダウンロード済みの `.iso` をマウントします。
3 マウントされた場所から、次のコマンドを実行します。

   `./install.sh`

4 使用許諾契約を読みます。
5 使用許諾契約の条項を確認し、「`y`」と入力します。

サイレントインストール

1 NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_RL_Linux.iso` をダウンロードします。
2 ダウンロード済みの `.iso` をマウントします。
3 `.iso` のルートディレクトリから、次のコマンドを実行します。

   `./create_silent_props.sh`

4「`y`」と入力して、ファイルの作成を確定します。
5 ファイルの場所を指定します。
6 **[Do you want to configure the silent properties file for Docker containers (Docker コンテナのサイレントプロパティファイルを設定しますか)]** パラメータに対して *n* を指定します。
7 次のコマンドを実行して、サイレントインストールを実行します。

   `./install.sh -s -f <` *サイレントプロパティファイルの場所* `>`

   次に例を示します。

   `./install.sh -s -f /home/silent.properties`。ここで、`/home/silent.properties` は、サイレントプロパティファイルを格納した場所です。

Designer のインストール

Designer は、GUI またはコンソールモードでインストールできます。

1 Software License and Download (ソフトウェアライセンスおよびダウンロード) (https://sld.microfocus.com/index) ポータルから、`Identity_Manager_4.9.0_Designer_Linux.tar.gz` をダウンロードします。
2 ファイルの抽出先のディレクトリに移動します。
3 次のコマンドを実行します。

   `tar -zxvf Identity_Manager_4.9.0_Designer_Linux.tar.gz`

4 次のいずれかのコマンドを実行して、Designer をインストールします。

   コンソール : `./install -i console`

GUI: `./install`

5 プロンプトに従ってインストールを続行します。

Analyzer のインストール

重要 : Analyzer ツールは Identity Manager 4.9 の ISO に含まれません。Analyzer がすでにインストールされている場合は、必ずバージョン 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 とともに使用してください。

このセクションでは、Analyzer をインストールし、Analyzer 用に環境を設定するさまざまな方法について説明します。

- ウィザードを使用した Analyzer のインストール
- Analyzer のサイレントインストール
- Analyzer.ini への XULRunner の追加

ウィザードを使用した Analyzer のインストール

次の手順では、GUI 形式またはコンソールからインストールウィザードを使用して、Linux または Windows プラットフォームに Analyzer をインストールする方法について説明します。無人のサイレントインストールを実行するには、76 ページの 「Analyzer のサイレントインストール」を参照してください。

1 Analyzer をインストールするコンピュータに `root` または管理者としてログインします。

2 (状況によって実行) Analyzer のインストールファイルをダウンロードした場合は、次の手順を実行します。

2a ダウンロードしたイメージの `.tgz` ファイルまたは `win.zip` ファイルへ移動します。

2b ファイルの内容をローカルコンピュータ上のフォルダに抽出します。

3 インストールプログラムを実行します。

```
./install
```

4 ウィザードの指示に従って、Analyzer のインストールを完了します。

5 インストールプロセスが完了したら、インストール後の概要を参照し、Analyzer のインストールステータスおよびログファイルの場所を確認します。

6 **[完了]** をクリックします。

7 (状況によって実行)76 ページの 「Analyzer.ini への XULRunner の追加」の手順を完了します。

8 Analyzer をアクティベートします。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Activating Analyzer」を参照してください。

Analyzer のサイレントインストール

サイレント (非対話型) インストールでは、ユーザインタフェースは表示されず、ユーザに対する質問も行われません。代わりに、InstallAnywhere はデフォルトの `analyzerInstaller.properties` ファイルの情報を使用します。デフォルトのファイルを使用してサイレントインストールを実行することも、ファイルを編集してインストールプロセスをカスタマイズすることもできます。

デフォルトでは、インストールプログラムは Analyzer を `root/Analyzer` ディレクトリにインストールします。

1. Analyzer をインストールするコンピュータに `root` または管理者としてログインします。
2. Analyzer のインストールファイルを［Software and License (ソフトウェアおよびライセンス)］ページ (https://sld.microfocus.com/) からダウンロードした場合は、次の手順を実行します。
   - **2a** ダウンロードしたイメージの `.tgz` ファイルまたは `win.zip` ファイルへ移動します。
   - **2b** ファイルの内容をローカルコンピュータ上のフォルダに抽出します。
3. (オプション) デフォルト以外のインストールパスを指定するには、次の手順を実行します。
   - **3a** `analyzerInstaller.properties` ファイル (デフォルトの場所は `<location where you have extracted the file>/analyzer_install/` ディレクトリです) を開きます。
   - **3b** このプロパティファイルに次のテキストを追加します。

     `USER_INSTALL_DIR=`*`installation_path`*
4. サイレントインストールを実行するには、プロパティファイルを含むディレクトリから次のコマンドを実行します。

   `./install -i silent -f analyzerInstaller.properties`
5. (状況によって実行) Linux コンピュータでは、76 ページの 「Analyzer.ini への XULRunner の追加」の手順を実行します。
6. Analyzer をアクティベートします。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Activating Analyzer」を参照してください。

Analyzer.ini への XULRunner の追加

Linux プラットフォームで Analyzer を実行する前に、XULRunner のマッピングを変更する必要があります。

1. XULRunner をインストールします。
2. `Analyzer` のインストールディレクトリに移動します。デフォルトの場所は次のとおりです。

   `root/analyzer`

3 gedit エディタで Analyzer.ini ファイルを開きます。

4 パラメータのリストの末尾に次の行を追加します。

```
-Djava.util.Arrays.useLegacyMergeSort=true
-Dorg.eclipse.swt.internal.gtk.disablePrinting
-Dorg.eclipse.swt.browser.XULRunnerPath=/opt/xulrunner
```

5 Analyzer.ini ファイルを保存します。

6 Analyzer を起動します。

Sentinel Log Management for Identity Governance and Administration のインストール

次の方法を使用して Sentinel Log Management をインストールできます。

- root ユーザによる Sentinel Log Management のインストール
- 非 root ユーザによる Sentinel Log Management のインストール

注 : インストールする前に、ホスト名が正しく指定されていること、IPV6 の設定が有効になっていないことを確認してください。

---

root ユーザによる Sentinel Log Management のインストール

1 NetIQ ダウンロード Web サイトから、SentinelLogManagementForIGA8.6.1.0.tar.gz をダウンロードします。

2 ファイルの抽出先のディレクトリに移動します。

3 次のコマンドを実行して、ファイルを抽出します。

```
tar -zxvf SentinelLogManagementForIGA8.6.1.0.tar.gz
```

4 SentinelLogManagementforIGA ディレクトリに移動します。

5 次のコマンドを実行します。

```
./install.sh
```

6 「y」と入力して使用許諾契約に同意し、インストールを続行します。

インストールパッケージをロードするのに数秒かかることがあります。

7 SLM for IGA のカスタム設定を実行するには、「2」を指定します。

8 デフォルトの評価版ライセンスキーを使用するには、「1」と入力します。

または

SLM for IGA の購入済みライセンスキーを入力するには、「2」と入力します。

9 管理者ユーザ admin のパスワードを指定し、パスワードを再度確認します。

10 データベースユーザ dbauser のパスワードを指定し、パスワードを再度確認します。

dbauser アカウントは、SLM for IGA がデータベースとのやり取りに使用する ID です。ここで入力するパスワードは、管理者パスワードを忘れた場合や紛失した場合の管理者パスワードのリセット操作を含む、データベース保守タスクの実行に使用します。

11 アプリケーションユーザ appuser のパスワードを指定し、パスワードを再度確認します。

12 必要な番号を入力して、ポート割り当てを変更します。

たとえば、Web サーバのデフォルトのポートは 8443 です。Web サーバのポート番号を変更するには、「4」を指定します。Web サーバの新しいポートの値を入力します。たとえば、8643 です。

13 ポートを変更してから「8」を指定し、完了します。

14 内部データベースのみを使用してユーザを認証するには、「1」を入力します。

または

ドメインで LDAP ディレクトリを設定している場合に、LDAP ディレクトリ認証を使用してユーザを認証するには、「2」を入力します。

デフォルト値は 1 です。

15 FIPS 140-2 モードを有効にするように促されたら「n」を入力します。

16 スケーラブルストレージを有効にするように促されたら「n」を入力します。

のインストールが終了し、サーバが起動します。システムが一度初期化を実行するため、インストール後にすべてのサービスを起動するのに数分かかることがあります。インストールが完了してから、Sentinel サーバにログインしてください。

SLM for IGA メインインタフェースにアクセスするには、Web ブラウザで次の URL を指定します。

```
https://<IP_Address/DNS_SLM for IGA_server>:<port>/sentinel/views/
main.html
```

<IP_Address/DNS_SLM for IGA_server> は SLM for IGA サーバの IP アドレスまたは DNS 名で、*<port>* は SLM for IGA サーバのポートです。

非 root ユーザによる Sentinel Log Management のインストール

Sentinel Log Management は novell ユーザとしてインストールする必要があります。NetIQ では、novell ユーザ以外の非 root ユーザのインストール環境はサポートしていませんが、その場合でもインストールは正常に完了します。

1 novell という非ルートユーザを作成します。

```
useradd novell
```

2 novell という非ルートグループを作成します。

```
groupadd novell
```

3 Sentinel をインストールするためのディレクトリを作成します。

```
mkdir /home/slmnonroot
```

4 非 root インストールディレクトリに novell ユーザ許可を割り当てます。

```
chown novell:novell /home/slmnonroot
```

5 適切なパスワードを設定します。

```
sudo passwd <password>
```

6 novell ユーザとしてログインします。

7 NetIQ ダウンロードの Web サイトから、SentinelLogManagementForIGA8.6.1.0.tar.gz をダウンロードします。

8 ファイルの抽出先のディレクトリに移動します。

9 次のコマンドを実行して、ファイルを抽出します。

```
tar -zxvf SentinelLogManagementForIGA8.6.1.0.tar.gz
```

10 SentinelLogManagementforIGA ディレクトリに移動します。

11 次のコマンドを実行します。

```
./install.sh --location=/home/slmnonroot
```

12 77 ページの 「root ユーザによる Sentinel Log Management のインストール」の手順 6 ～ 16 に従います。

Java リモートローダのインストール

Java リモートローダ、dirxml_jremote は、オペレーティングシステムがネイティブリモートローダと互換性がないコンピュータにインストールします。ただし、Java リモートローダは、ネイティブリモートローダをインストールする可能性がある同一サーバ上でも実行できます。Identity Manager は、Java リモートローダを使用して、1 つのサーバで実行されている Identity Manager エンジンと、rdxml が動作しない別の場所で実行されている Identity Manager ドライバとの間でデータを交換します。公式にサポートされているバージョンの Java とともに、サポート対象の任意の Linux コンピュータに dirxml_jremote をインストールできます。

1 Identity Manager エンジンをホストするサーバで、アプリケーションシムの .iso ファイルまたは .jar ファイルをコピーします。これらのファイルは、デフォルトでは /opt/novell/eDirectory/lib/dirxml/classes ディレクトリにあります。

2 Java リモートローダをインストールするコンピュータ (ターゲットコンピュータ) にログインします。

3 ターゲットコンピュータに、サポートされているバージョンの JRE がインストールされていることを確認します。

4 インストールプログラムにアクセスするため、次のいずれかの手順を実行します。

4a (状況によって実行) Identity Manager インストールパッケージの .iso イメージファイルがある場合は、Java リモートローダのインストールファイルが保存されているディレクトリへ移動します。デフォルトの場所は /IDM/packages/java_remoteloader です。

4b (状況によって実行) Java リモートローダのインストールファイルを [Software and License (ソフトウェアおよびライセンス)] ページ (https://sld.microfocus.com/) からダウンロードした場合は、次の手順を実行します。

4b1 ダウンロードしたイメージの .tgz ファイルへナビゲートします。

4b2 ファイルの内容をローカルコンピュータ上のフォルダに抽出します。

5 dirxml_jremote_dev.tar.gz ファイルをターゲットコンピュータ上の目的の場所にコピーします。たとえば、/usr/idm にファイルをコピーします。

6 次のいずれかのファイルをターゲットコンピュータ上の目的の場所にコピーします。

- dirxml_jremote.tar.gz
- dirxml_jremote_mvs.tar

  mvs の詳細については、dirxml_jremote_mvs.tar ファイルを untar し、ドキュメント usage.html を参照してください。

7 ターゲットコンピュータで、.tar.gz ファイルを圧縮解除して展開します。

たとえば、tar -zxvf dirxml_jremote.tar.gz です。

8 ステップ 1 でコピーしたアプリケーションシムの .iso ファイルまたは .jar ファイルを、lib ディレクトリの下層にある dirxml/classes ディレクトリに保存します。

9 RDXML_PATH 環境変数によって Java 実行可能ファイルにアクセスできるように dirxml_jremote スクリプトをカスタマイズするため、次の手順を実行します。

9a 次のいずれかのコマンドを入力して、環境変数 RDXML_PATH を設定します。

- set RDXML_PATH=path
- export RDXML_PATH

9b dirxml_jremote スクリプトを編集して、Java 実行可能ファイルへのパスを Java を実行するスクリプトラインに追加します。

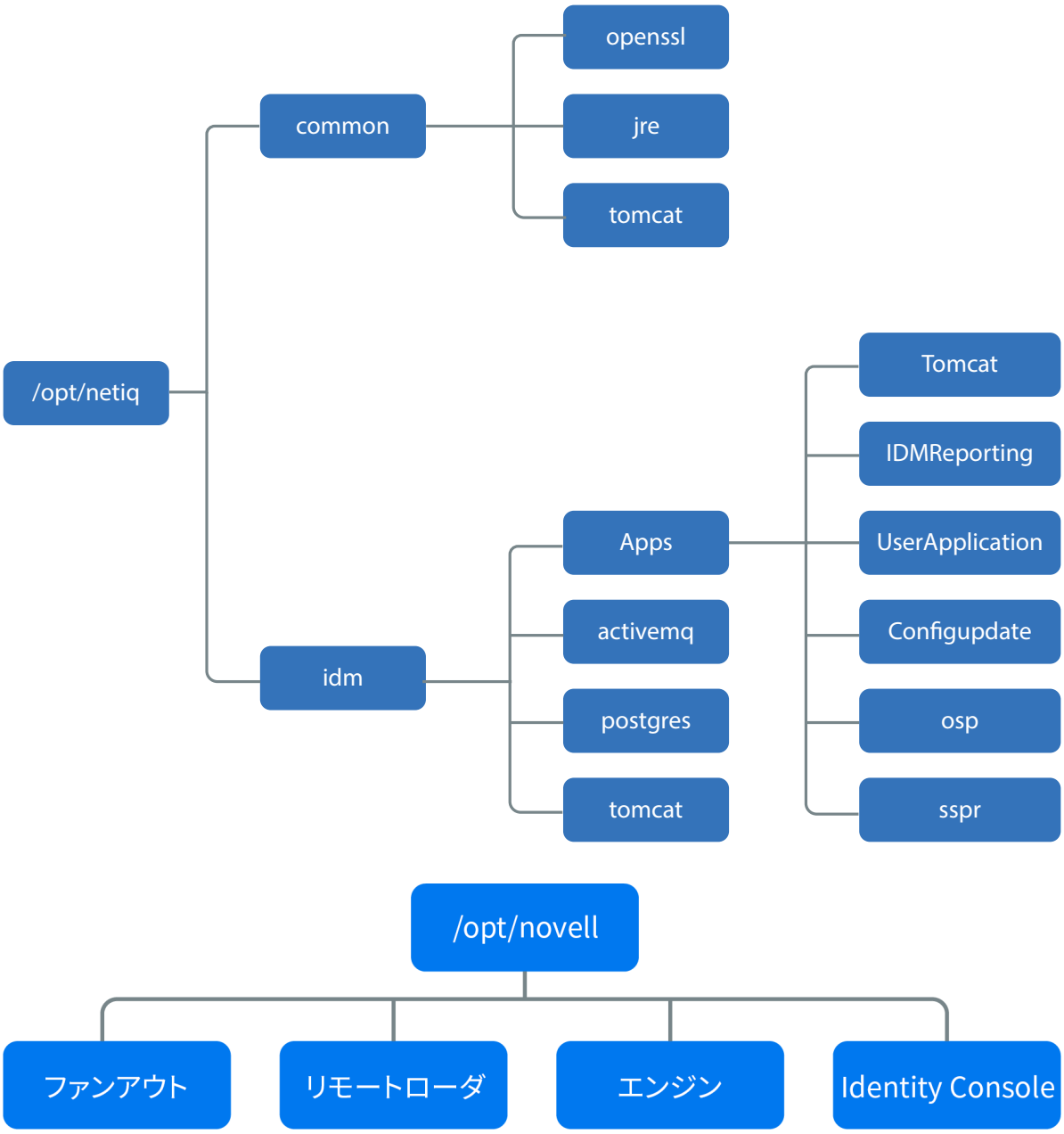
10 展開されていない dirxml_jremote.tar.gz ディレクトリの lib サブディレクトリから dirxml_jremote スクリプトに jar ファイルの場所を指定する必要があります。たとえば、/lib/*.jar です。

11 サンプル環境設定ファイル config8000.txt をアプリケーションシム用に設定します。

このサンプルファイルは、デフォルトでは /opt/novell/dirxml/doc ディレクトリにあります。詳細については、103 ページの 「リモートローダとドライバの設定」を参照してください。

ディレクトリ構造の理解

インストールプロセスにより、次のディレクトリ構造が作成されます。



- `/opt/netiq` ディレクトリは、ディレクトリ構造の開始点です。他のすべてのファイルおよびディレクトリはこのディレクトリの下にあります。
- `common` ディレクトリには、サポートするソフトウェアが含まれています。このソフトウェアは、必要とするコンポーネント間で共有されます。
- `idm` ディレクトリには、コンポーネントをインストールおよび設定するためのバイナリファイルを含むコンポーネント固有のサブディレクトリが含まれています。
- エンジン、ファンアウトエージェント、リモートローダ、および Identity Console は、`/opt/novell` ディレクトリにインストールされます。

5 Identity Manager コンポーネントの設定

このセクションでは、Identity Manager コンポーネントの設定のプロセスを順を追って説明します。設定プロセスを開始する前に、各コンポーネントの設定オプションを確認する必要があります。詳細については、83 ページの 「環境設定パラメータの理解」を参照してください。

Designer、Analyzer、リモートローダ、ファンアウトエージェントなどの一部のコンポーネントは、設定が不要な場合があります。

設定時における直感的でないパスワードの使用

ほとんどの Identity Manager コンポーネントでは、設定段階でパスワードを指定する必要があります。設定時間を短縮するために、すべての環境設定パラメータに同じパスワードを適用するようプロセスに指示できます。

パスワードは 6 文字以上にする必要があります。辞書にある単語を使用しないでください。辞書の単語は、辞書リストによく付属している自由に利用できるパスワード解読ツールに対して脆弱です。辞書の単語を使用する必要がある場合は、数字と句読点を組み合わせてみてください。

環境設定パラメータの理解

このセクションでは、インストールした Identity Manager を設定するために指定する必要があるパラメータを定義します。インストールプログラムを使用してインストール直後にコンポーネントを設定したり、`configure.sh` スクリプトを実行してコンポーネントを後で設定することができます。

注

- 標準設定モードで設定された Identity Applications および Identity Reporting は、別のコンピュータにインストールされたデータベースサーバに接続できません。
- インストールプロセスでは、Identity Manager コンポーネントの監査を有効にすることはできません。インストールの完了後、各コンポーネントの監査を設定する必要があります。詳細については、『*NetIQ Identity Manager - Configuring Auditing in Identity Manager*』を参照してください。
- アイデンティティボールトは OES とともに自動的にインストールされます。OES プラットフォームで Identity Manager エンジンを設定するには、**[カスタム環境設定]** を選択してから **[Add to an Existing Vault (既存のボールトに追加)]** を選択する必要があります。

- Identity Applications、Identity Reporting、およびデータベースが、FQDN または IP アドレスで均一に設定されていることを確認します。つまり、これらのコンポーネントを FQDN と IP アドレスの組み合わせで設定しないでください。
- コンテナの場合、IP アドレスの代わりに FQDN 値を指定することをお勧めします。

表 5-1 は、標準モードで Identity Manager コンポーネントを設定するために必要なパラメータを示しています。

表 5-1 典型的な構成

パラメータ	サイレントプロパティファイルのパラメータ	典型的な構成
Identity Manager エンジン		
共通パスワード	IS_COMMON_PASSWORD	共通パスワードを設定するかどうかを指定します。
Identity Vault Hostname/IP Address (アイデンティティボールトホスト名 /IP アドレス)	ID_VAULT_HOST	アイデンティティボールトがインストールされるサーバの IP アドレスを指定します。
Identity Vault Administrator name (アイデンティティボールト管理者名)	ID_VAULT_ADMIN_LDAP	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相対識別名 (RDN) を指定します。
識別情報アプリケーション		
共通パスワード	IS_COMMON_PASSWORD	共通パスワードを設定するかどうかを指定します。パスワードが 83 ページの 「設定時における直感的でないパスワードの使用」セクションに記載された考慮事項を満たしていることを確認してください。
Common One SSO Service password (One SSO Service 共通パスワード)	SSO_SERVICE_PWD	One SSO サーバの認証サービスのパスワードを指定します。
SSPR 設定パスワード		SSPR の認証サービスのパスワードを指定します。
Identity Applications administrator name (Identity Applications 管理者名)	UA_ADMIN	識別情報アプリケーションの管理者アカウントの名前を指定します。

パラメータ	サイレントプロパティファイルのパラメータ	典型的な構成
Identity Applications Administrator Password (Identity Applications 管理者パスワード)		Identity Applications の管理者パスワードを指定します。
Identity Applications and Workflow Engine database user password (Identity Applications およびワークフローエンジンデータベースユーザパスワード)		Identity Applications およびワークフローエンジンデータベースユーザパスワードを指定します。
Identity Applications and Workflow Engine database Administrator password (Identity Applications およびワークフローエンジンデータベース管理者パスワード)		Identity Applications およびワークフローエンジンデータベース管理者パスワードを指定します。
NGINX port (NGINX ポート)	NGINX_HTTPS_PORT	NGINX ポートを指定します。
Identity Reporting		
共通パスワード	IS_COMMON_PASSWORD	共通パスワードを設定するかどうかを指定します。パスワードが 83 ページの「設定時における直感的でないパスワードの使用」セクションに記載された考慮事項を満たしていることを確認してください。
Identity Reporting データベースアカウントパスワード	RPT_DATABASE_SHARE_PASSWORD	Identity Reporting のデータベースアカウントパスワードを指定します。
Identity Reporting 管理者名	RPT_ADMIN	Identity Reporting の管理者名を指定します。デフォルト値は、`cn=uaadmin,ou=sa,o=data` です。
Identity Reporting 管理者パスワード		管理者オブジェクトのパスワードを指定します。たとえば、*password* です。
Identity Console		
Identity Console Server Name (Identity Console サーバ名)	ID_CONSOLE_SERVER_HOST	Identity Console をホストするサーバの名前を指定します。
Identity Console Server Port (Identity Console サーバポート)	ID_CONSOLE_SERVER_SSL_PORT	Identity Console をホストするポートを指定します。

パラメータ	サイレントプロパティファイルのパラメータ	典型的な構成
OSP for authorization with Identity Console (Identity Console による許可のための OSP)	ID_CONSOLE_USE_OSP	Identity Console による許可に OSP を使用するかどうかを指定します (y/n)。
One SSO Service password (Identity Reporting の One SSO Service パスワード)	SSO_SERVICE_PWD	*認可に対して OSP を [はい] として選択した場合にのみ適用されます。* One SSO Service のパスワードを指定します。
Trusted root certificate folder path (トラステッドルート証明書フォルダのパス)	ID_CONSOLE_CA_CERT_LOCATION	トラステッドルート証明書 (SSCert.pem) フォルダのパスを指定します。
Server certificate path including the filename (ファイル名を含むサーバ証明書パス)	ID_CONSOLE_SERVER_CERT_LOCATION	ファイル名を含むサーバ証明書パスを指定します。
Server certificate password (サーバ証明書のパスワード)	ID_CONSOLE_SERVER_CERT_PASSWD	サーバ証明書のパスワードを指定します。

表 5-2 は、カスタムモードで Identity Manager コンポーネントを設定するために必要なパラメータを示しています。

***表 5-2** カスタム環境設定*

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Identity Manager エンジン		

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Create a new Identity Vault (アイデンティティボールトの新規作成)	TREE_CONFIG	インストールするアイデンティティボールトを指定します。
Add to an Identity Vault existing on local machine (ローカルマシンに存在するアイデンティティボールトに追加)		Identity Manager エンジンをインストールする同じサーバ上の既存のアイデンティティボールトに接続するかどうかを指定します。
Add to an Identity Vault existing on remote machine (リモートマシンに存在するアイデンティティボールトに追加)		Identity Manager エンジンとは異なるサーバにインストールされたアイデンティティボールトに接続するかどうかを指定します。
Identity Vault Tree Name (アイデンティティボールトツリー名)	ID_VAULT_TREENAME	アイデンティティボールトの新しいツリーを指定します。ツリー名は次の要件を満たしている必要があります。 ◆ ネットワーク内では、重複するツリー名を使用することはできません。 ◆ ツリー名の長さは2〜32文字にする必要があります。 ◆ ツリー名で使用できる文字は、文字 (A 〜 Z)、数字 (0 〜 9)、ハイフン (-)、およびアンダースコア (_) のみです。 **注 :** OES に Identity Manager をインストールする場合、既存のツリー名を指定します。
Identity Vault Server Name (アイデンティティボールトサーバ名)	ID_VAULT_SERVERNAME	アイデンティティボールトサーバ名を指定します。
アイデンティティボールトコンテキスト DN	ID_VAULT_SERVER_CONTEXT	既存のアイデンティティボールトサーバのコンテキスト DN を指定します。デフォルト値は servers.system です。
Identity Vault Administrator name (アイデンティティボールト管理者名)	ID_VAULT_ADMIN_LDAP	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相対識別名 (RDN) を指定します。
識別ボールト管理者パスワード	ID_VAULT_PASSWORD	管理者オブジェクトのパスワードを指定します。たとえば、*password* です。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
RSA Key Size	ID_VAULT_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。デフォルトの設定は 4096 です。
EC Curve	ID_VAULT_EC_CURVE	EC 証明書の楕円曲線 (EC) 制限を指定します。使用可能な値は P256、P384、および P521 です。デフォルト値は P384 です。
証明書の有効期間	ID_VAULT_CA_LIFE	**[新しいツリーの作成]** オプションを選択した場合にのみ適用されます。 証明書の有効期間を年単位で指定します。
NDS var フォルダの場所	ID_VAULT_VARDIR	このサーバ上のこのアイデンティティボールトインスタンスのパスを指定します。デフォルトのパスは /var/opt/novell/eDirectory です。
NDS データの場所	ID_VAULT_DIB	ディレクトリ情報データベース (DIB) ファイルのインストール先であるローカルシステムのパスを指定します。DIB ファイルはアイデンティティボールトデータベースファイルです。デフォルトの場所は /var/opt/novell/eDirectory/data/dib です。
NCP ポート	ID_VAULT_NCP_PORT	識別ボールトが Identity Manager コンポーネントとの通信に使用する NCP (NetWare Core Protocol) ポートを指定します。デフォルトの設定は 524 です。
LDAP 非 SSL ポート	ID_VAULT_LDAP_PORT	識別ボールトが平文の LDAP 要求をリスンするポートを指定します。デフォルトの設定は 389 です。
LDAP SSL ポート	ID_VAULT_LDAPS_PORT	識別ボールトが SSL (Secure Sockets Layer) プロトコルを使用した LDAP 要求をリスンするポートを指定します。デフォルトの設定は 636 です。
アイデンティティボールト HTTP ポート	ID_VAULT_HTTP_PORT	HTTP スタックが平文で動作するポートを指定します。デフォルトの設定は 8028 です。
アイデンティティボールト HTTPS ポート	ID_VAULT_HTTPS_PORT	HTTP スタックが TLS/SSL プロトコルを使用して動作するポートを指定します。デフォルトの設定は 8030 です。
パスを含む NDS 設定ファイル	ID_VAULT_CONF	アイデンティティボールトの設定ファイルの場所を指定します。デフォルト値は /etc/opt/novell/eDirectory/conf/nds.conf です。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
アイデンティティボールトドライバセット名	ID_VAULT_DRIVER_SET	新しい Identity Manager ドライバセットオブジェクトの名前を指定します。
アイデンティティボールトドライバセット展開コンテキスト	ID_VAULT_DEPLOY_CTX	ドライバセットオブジェクトの作成先コンテナの LDAP DN を指定します。
識別情報アプリケーション		
共通パスワード	IS_COMMON_PASSWORD	共通パスワードを設定するかどうかを指定します。パスワードが 83 ページの「設定時における直感的でないパスワードの使用」セクションに記載された考慮事項を満たしていることを確認してください。
ホスト名 (小文字の FQDN)	UA_SERVER_HOST	サーバの完全修飾識別名またはデフォルト IP アドレスを指定します。 **注 :** FQDN が小文字で指定されていることを確認します。コンピュータをホストしているサーバが小文字の FQDN を使用するように設定されている必要もあります。
Identity Vault Hostname/IP Address (アイデンティティボールトホスト名 /IP アドレス)	ID_VAULT_HOST	アイデンティティボールトがインストールされるサーバの IP アドレスまたはホスト名を指定します。
Identity Vault Administrator name (アイデンティティボールト管理者名)	ID_VAULT_ADMIN_LDAP	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相対識別名 (RDN) を指定します。
識別ボールト管理者パスワード	ID_VAULT_PASSWORD	管理者オブジェクトのパスワードを指定します。たとえば、*password* です。
OSP カスタムログイン画面名	OSP_CUSTOM_NAME	OSP ログイン画面に表示される名前を指定します。
ユーザ検索コンテナ DN	USER_CONTAINER	アイデンティティボールト内にあるすべてのユーザオブジェクトのデフォルトコンテナを指定します。
管理者検索コンテナ DN	ADMIN_CONTAINER	認証サービス (OSP) が認証する必要がある管理者ユーザオブジェクトが含まれる識別ボールト内のコンテナの識別名を指定します。たとえば、「o=data」と指定します。
アプリケーションサーバ DNS/IP アドレス	TOMCAT_SERVLET_HOSTNAME	Tomcat サーバの IP アドレスまたはホスト名を指定します。
One SSO Server SSL ポート	SSO_SERVER_SSL_PORT	シングルサインオンサービスで使用するポートを指定します。デフォルトの設定は 8543 です。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Identity Application One SSO Service password (Identity Applications の One SSO Service パスワード)	SSO_SERVICE_PWD	*共通パスワードが* **[いいえ]** に設定されている場合にのみ適用されます。 Identity Applications で使用するシングルサインオンクライアントのパスワードを指定します。
Identity Applications administrator name (Identity Applications 管理者名)	UA_ADMIN	識別情報アプリケーションの管理者アカウントの名前を指定します。
SSPR 設定パスワード	CONFIGURATION_PWD	*共通パスワードが* **[いいえ]** に設定されている場合にのみ適用されます。 Identity Applications によって使用されるパスワード管理用パスワードを指定します。
OAuth keystore password (OAuth キーストアのパスワード)	OSP_KEYSTORE_PWD	*共通パスワードが* **[いいえ]** に設定されている場合にのみ適用されます。 OAuth サーバで新しいキーストアをロードするために作成するパスワードを指定します。
データベースプラットフォーム	UA_DB_PLATFORM_OPTION	Identity Applications に必要なデータベースを指定します。
Configure PostgreSQL on current server (現在のサーバ上に PostgreSQL を設定)	INSTALL_PG_DB	同じサーバ上に PostgreSQL データベースを設定するかどうかを指定します。
Identity アプリケーションデータベースポート	UA_DB_PORT	Identity Applications のデータベースポートを指定します。
Identity アプリケーションデータベース名	UA_DATABASE_NAME	データベースの名前を指定します。デフォルト値は idmuserappdb です。
Identity Applications database and Workflow Engine user name (Identity Applications データベースおよびワークフローエンジンユーザ名)	UA_WFE_DATABASE_USER	Identity Applications データベースの管理者のユーザ名を指定します。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Identity Applications and Workflow Engine database administrator password for user postgres (postgres ユーザ用の Identity Applications およびワークフローエンジンデータベース管理者パスワード)	UA_WFE_DATABASE_ADMIN_PWD	postgres ユーザ用の Identity Applications およびワークフローエンジンデータベース管理者パスワードを指定します。
Use custom container as root container (カスタムコンテナをルートコンテナとして使用する)	ENABLE_CUSTOM_CONTAINER_CREATION	カスタムコンテナをルートコンテナとして使用するかどうかを指定します。デフォルトでは、インストーラは o=data を作成し、それをユーザコンテナとして選択して、パスワードポリシーと必要なトラスティ権を割り当てます。 カスタムコンテナを作成するには、**[はい]** を選択します。
Create the User Application and Roles and Resources Services drivers for Identity Applications (Identity Applications 用のユーザアプリケーションドライバと役割およびリソースサービスドライバを作成する)	UA_CREATE_DRIVERS	UA ドライバおよび RRSD ドライバをインストールするかどうかを指定します。**[N]** を選択した場合、既存のユーザアプリケーションドライバの名前を指定する必要があります。
既存のユーザアプリケーションドライバの名前	UA_DRIVER_NAME	UA ドライバおよび RRSD ドライバの作成の値を **[いいえ]** に設定した場合にのみ適用されます。 既存のユーザアプリケーションドライバ DN の詳細を指定します。
User Application Workflow Engine ID (ユーザアプリケーションワークフローエンジン ID)	UA_WORKFLOW_ENGINE_ID	Identity Applications ワークフローエンジン ID を示します。
Identity Applications and Workflow Engine database platform (Identity Applications およびワークフローエンジンデータベースプラットフォーム)	UA_WFE_DB_PLATFORM_OPTION	Identity Applications およびワークフローエンジンデータベースプラットフォームを示します。サポートされている値は、［Postgres］、［Oracle］、および［MsSQL］です。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Identity Applications and Workflow Engine database port (Identity Applications およびワークフローエンジンデータベースポート)	UA_WFE_DB_PORT	Identity Applications およびワークフローエンジンデータベースポートを示します。
ワークフローエンジンデータベース名	WFE_DATABASE_NAME	ワークフローエンジンデータベース名を示します。. デフォルトのデータベース名は igaworkflowdb です。
Identity Applications and Workflow Engine database password (Identity Applications およびワークフローエンジンデータベースパスワード)	UA_WFE_DATABASE_PWD	Identity Applications およびワークフローエンジンデータベースパスワードを示します。
Identity Applications およびワークフローエンジンデータベース JDBC ドライバ jar ファイル	UA_WFE_DB_JDBC_DRIVER_JAR	データベースプラットフォームの JAR ファイルを指定します。
Create schema (スキーマの作成)	UA_WFE_DB_CREATE_OPTION	インストールプロセスの一環として Identity Applications およびワークフローエンジンのデータベーススキーマをいつ作成するかを示します。使用可能なオプションは、**[Now]**、**[Startup]**、および **[File]** です。
Create a new database or upgrade/migrate from an existing database for Workflow Engine (新しいデータベースを作成するか、ワークフローエンジンの既存のデータベースからアップグレード / 移行)	WFE_DB_NEW_OR_EXIST	新しいデータベースのテーブルを作成するか、ワークフローエンジンの既存のデータベースから更新 / 移行するかを指定します。サポートされている値は［new (新規)］と［exist (既存)］です。
Create a new database or upgrade/migrate from an existing database for Identity Applications (新しいデータベースを作成するか、Identity Applications の既存のデータベースからアップグレード / 移行)	UA_DB_NEW_OR_EXIST	新しいデータベースを作成するか、または既存のデータベースからアップグレードするかを指定します。
ルートコンテナ	ROOT_CONTAINER	ルートコンテナを指定します。デフォルト値は o=data です。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
グループ検索のルートコンテナ DN	GROUP_ROOT_CONTAINER	グループ検索のルートコンテナ DN を指定します。
Identity Applications Custom Certificate (Identity Applications カスタム証明書)	CUSTOM_UA_CERTIFICATE	Identity Applications のカスタム証明書を使用するかどうかを示します。
カスタムコンテナの LDIF ファイルパス		*カスタムコンテナを* [はい] として設定する場合にのみ適用されます。 カスタムコンテナの LDIF ファイルのパスを指定します。
NGINX port (NGINX ポート)	NGINX_HTTPS_PORT	NGINX ポートを指定します。
Identity Reporting		
共通パスワード	IS_COMMON_PASSWORD	共通パスワードを設定するかどうかを指定します。パスワードが 83 ページの 「設定時における直感的でないパスワードの使用」 セクションに記載された考慮事項を満たしていることを確認してください。
ホスト名 (小文字の FQDN)		サーバの完全修飾識別名またはデフォルト IP アドレスを指定します。 **注 :** FQDN が小文字で指定されていることを確認します。コンピュータをホストしているサーバが小文字の FQDN を使用するように設定されている必要もあります。
Identity Vault Hostname/IP Address (アイデンティティボールトホスト名 /IP アドレス)	ID_VAULT_HOST	アイデンティティボールトがインストールされるサーバの IP アドレスまたはホスト名を指定します。
Identity Vault Administrator name (アイデンティティボールト管理者名)	ID_VAULT_ADMIN_LDAP	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相対識別名 (RDN) を指定します。
アイデンティティボールト管理者パスワード	ID_VAULT_PASSWORD	管理者オブジェクトのパスワードを指定します。たとえば、*password* です。
アプリケーションサーバ HTTPS ポート	TOMCAT_HTTPS_PORT	Tomcat サーバがクライアントコンピュータとの通信に使用する HTTPS ポートを指定します。デフォルトの設定は 8543 です。
アプリケーションサーバ DNS/IP アドレス	TOMCAT_SERVLET_HOSTNAME	Tomcat サーバの IP アドレスまたはホスト名を指定します。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Select the database platform for Identity Reporting (Identity Reporting のデータベースプラットフォームを選択)	RPT_DATABASE_PLATFORM_OPTION	Identity Reporting で使用するデータベースを指定します。
Configure PostgreSQL on current server (現在のサーバ上に PostgreSQL を設定)	INSTALL_PG_DB_FOR_REPORTING	同じサーバ上に PostgreSQL データベースを設定するかどうかを指定します。
Identity Reporting データベースアカウントパスワード	RPT_DATABASE_SHARE_PASSWORD	Identity Reporting のデータベースアカウントパスワードを指定します。
Create a new database or upgrade/migrate from an existing database (新しいデータベースを作成するか、既存のデータベースからアップグレード / 移行する)	RPT_DATABASE_NEW_OR_EXIST	新しいデータベースを作成するか、または既存のデータベースからアップグレードするかを指定します。
Identity Reporting 管理者名	RPT_ADMIN	Identity Reporting の管理者名を指定します。デフォルト値は、cn=uaadmin,ou=sa,o=data です。
Identity Reporting 管理者パスワード	RPT_ADMIN_PWD	Identity Reporting の管理者パスワードを指定します。
Identity Reporting データベース名	RPT_DATABASE_NAME	Identity Reporting のデータベース名を指定します。デフォルト値は idmrptdb です。
Identity Reporting データベースユーザ	RPT_DATABASE_USER	Identity Reporting がデータベースのデータにアクセスし、変更することを許可する管理アカウントを指定します。デフォルト値は rptadmin です。
Identity Reporting データベースホスト		データベースを作成するサーバの DNS 名または IP アドレスを指定します。
Identity Reporting データベースポート	RPT_DATABASE_PORT	データベースに接続するポートを指定します。デフォルトポートは 5432 です。
Identity Application database JDBC jar file (Identity Applications データベースの JDBC jar ファイル)	RPT_DATABASE_JDBC_DRIVER_JAR	データベースプラットフォームの JAR ファイルを指定します。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Identity Reporting Tomcat keystore file with subject alternate name (サブジェクトの代替名を持つ Identity Reporting Tomcat キーストアファイル)	RPT_COMM_TOMCAT_KEYSTORE_FILE	Tomcat キーストアファイルの場所を指定します。 **注 :** カスタム証明書は PKCS タイプでのみサポートされます。
Create schema (スキーマの作成)	RPT_DATABASE_CREATE_OPTION	インストールプロセスの一環としていつデータベーススキーマを作成するかを指定します。使用可能なオプションは、**[Now]**、**[Startup]**、および **[File]** です。
Create the MSGW and DCS drivers for Identity Reporting (Identity Reporting の MSGW および DCS ドライバの作成)	RPT_CREATE_DRIVERS	MSGW および DCS ドライバを作成するかどうかを指定します。
NGINX HTTPS ポート	NGINX_HTTPS_PORT	NGINX のポートを指定します。
Identity Console		
Identity Console Server Name (Identity Console サーバ名)	ID_CONSOLE_SERVER_HOST	Identity Console をホストするサーバの名前を指定します。
Identity Console Server Port (Identity Console サーバポート)	ID_CONSOLE_SERVER_SSL_PORT	Identity Console をホストするポートを指定します。
OSP for authorization with Identity Console (Identity Console による許可のための OSP)	ID_CONSOLE_USE_OSP	Identity Console による許可に OSP を使用するかどうかを指定します (y/n)。
One SSO Service password (Identity Reporting の One SSO Service パスワード)	SSO_SERVICE_PWD	*認可に対して OSP を* **[はい]** *として選択した場合にのみ適用されます。* One SSO Service のパスワードを指定します。
Trusted root certificate folder path (トラステッドルート証明書フォルダのパス)	ID_CONSOLE_CA_CERT_LOCATION	トラステッドルート証明書 (SSCert.pem) フォルダのパスを指定します。
Server certificate path including the filename (ファイル名を含むサーバ証明書パス)	ID_CONSOLE_SERVER_CERT_LOCATION	ファイル名を含むサーバ証明書パスを指定します。

パラメータ	サイレントプロパティファイルのパラメータ	カスタム環境設定
Server certificate password (サーバ証明書のパスワード)	ID_CONSOLE_SERVER_CERT_PASSWD	サーバ証明書のパスワードを指定します。
IDVault ツリー名	ID_CONSOLE_TREENAME	Identity Console のホスト先のアイデンティティボールトツリー名を入力します。

ドライバセットの作成と設定

Identity Manager インストールキットの IDM/LDIF/ ディレクトリの sample-driverset.ldif ファイルを使用して、ドライバセットを作成することができます。ファイルの内容は次のとおりです。

```
dn: cn=driverset1,o=system
changetype: add
DirXML-LogLimit: 0
DirXML-ConfigValues::
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48Y29u

ZmlndXJhdGlvbi12YWx1ZXM+Cgk8ZGVmaW5pdGlvbnMvPgo8L2NvbmZpZ3VyYXRpb24tdmFsdW
VzPg==
objectClass: DirXML-DriverSet
objectClass: Top
objectClass: Partition
objectClass: nsimPasswordPolicyAux

dn: cn=DirXML-PasswordPolicy,cn=Password Policies,cn=Security
changetype: add
nsimPwdRuleEnforcement: FALSE
nspmSpecialAsLastCharacter: TRUE
nspmSpecialAsFirstCharacter: TRUE
nspmSpecialCharactersAllowed: TRUE
nspmNumericAsLastCharacter: TRUE
nspmNumericAsFirstCharacter: TRUE
nspmNumericCharactersAllowed: TRUE
description: This Password Policy is used by IDM Engine
nspmMaximumLength: 64
nspmConfigurationOptions: 596
passwordUniqueRequired: FALSE
passwordMinimumLength: 1
passwordAllowChange: TRUE
objectClass: nspmPasswordPolicy
objectClass: Top
cn: DirXML-PasswordPolicy
nsimAssignments: cn=driverset1,o=system
```

新規インストールでのドライバセットの作成

テキストエディタで、sample-driverset.ldif ファイルを開き、以下の変更を行います。カスタム LDIF ファイルパスを提供している場合は、カスタムパスを指定します。

1. ドライバセット DN が新しいドライバセットを指すようにします。たとえば、dn: cn=driverset1,o=system を dn:cn=Driverset47,ou=drivers,o=acme に変更します。
2. nsimAssignments 属性値を新しいドライバセットの DN に変更します。たとえば、nsimAssignments: cn=driverset1,o=system を nsimAssignments: cn=Driverset47,ou=drivers,o=acme に変更します。

注 : コンテンツをそのままコピーすると、ファイルにいくつかの非表示の特殊文字が挿入される場合があります。これらの属性をアイデンティティボールトに追加する際に ldif_record() = 17 のエラーメッセージが表示される場合は、2 つの DN 間にスペースを挿入してください。

既存のサーバ上のドライバセットの設定

Identity Manager が eDirectory ツリーのサーバにすでにインストールされている場合、DirXML-PasswordPolicy オブジェクトがツリーに存在します。次のオプションから選択できます。

- 既存のパスワードポリシーを使用する

  変更前 :

  ```
  dn: cn=DirXML-PasswordPolicy,cn=Password Policies,cn=Security
  changetype: modify
  add: nsimAssignments
  nsimAssignments: cn=driverset1,o=system
  ```

- 別のパスワードポリシーを使用する

  使用対象 :

  ```
  dn: cn=DirXML-PasswordPolicy,cn=Password Policies,cn=Security
  changetype: add
  ```

テキストエディタで、sample-driverset.ldif ファイルを開き、以下の変更を行います。

1. ドライバセット DN が新しいドライバセットを指すようにします。
2. nsimAssignments 属性値を新しいドライバセットの DN に変更します。
3. 既存の DirXML-PasswordPolicy オブジェクトまたは別のパスワードポリシーを指すように DirXML-PasswordPolicy 属性を変更します。

Identity Manager コンポーネントの設定

次の方法で設定を実行できます。

- 対話型設定
- サイレント環境設定

対話型設定の実行

1. Identity_Manager_4.9.0_Linux.iso ファイルをマウントした場所に移動します。
2. コマンドラインで次のコマンドを指定して、configure.sh スクリプトを実行します。

   ./configure.sh
3. 標準設定またはカスタム設定のどちらを実行するかを決定します。設定オプションは、設定用に選択したコンポーネントに応じて変わります。
4. コンポーネントを設定するには、83 ページの 「環境設定パラメータの理解」の情報を参考にします。

注 : Identity Manager エンジンの設定プロセス中に、単純認証の TLS が [はい] に設定されます。単純バインドの TLS の値を [No] に変更する場合は、Identity Manager エンジンがインストールされているサーバで次のコマンドを実行します。

```
ldapconfig set "Require TLS for Simple Binds with Password=no"
```

サイレント設定の実行

1. Identity_Manager_4.9.0_Linux.iso ファイルをマウントした場所に移動します。
2. サイレントインストールを実行するには、次のコマンドを実行します。

   ./configure.sh -s -f < *サイレントプロパティファイルの場所* >

   次に例を示します。

   ./configure.sh -s -f /home/silent.properties。ここで、/home/silent.properties は、サイレントプロパティファイルを格納した場所です。
3. コンポーネントを設定するには、83 ページの 「環境設定パラメータの理解」の情報を参考にします。

SSPR の設定

次のセクションでは、SSPR の設定について説明します。コンポーネントを設定する前に、83 ページの 「環境設定パラメータの理解」の情報を確認してください。

注 : SSPR を設定する前に、アイデンティティボールトに次のコンテナとユーザオブジェクトが存在することを確認してください。

- ユーザ検索コンテナ

- 管理者検索コンテナ
- 識別情報アプリケーション管理者ユーザ

対話型設定の実行

1. `Identity_Manager_4.9.0_Linux.iso` をマウントした場所に移動します。
2. `sspr` ディレクトリに移動します。
3. 次のコマンドを実行します :

   `./configure.sh`
4. 設定を行います。

サイレント設定の実行

設定を開始する前に、`sspr_silentinstall.properties` (`<iso mounted path>/sspr/`) ファイルが書き込み可能なディレクトリにコピーされていることを確認してください。たとえば、ファイルを `/tmp` ディレクトリにコピーします。

1. `Identity_Manager_4.9.0_Linux.iso` をマウントした場所に移動します。
2. `sspr` ディレクトリに移動します。
3. 次のコマンドを実行します :

   `./configure.sh -s -f <`*サイレントプロパティファイルの場所*`>`

   次に例を示します。

   `./configure.sh -s -f /tmp/sspr_silentinstall.properties`。ここで、`/tmp/sspr_silentinstall.properties` はサイレントプロパティファイルを保存した場所です。
4. 設定を行います。

OSP サーバでのシングルサインオンアクセス設定の変更

SSPR と OSP が別々のサーバにインストールされている場合 (SSPR が Identity Applications または Identity Reporting とは異なるサーバにインストールされている場合)、OSP サーバでシングルサインオンアクセス設定を行う必要があります。このセクションでは、現在の環境でその設定が正常に機能することを保証する方法について説明します。

1. OSP がインストールされているサーバで RBPM 設定更新ユーティリティを起動します。
2. **[SSO クライアント]** > **[セルフサービスパスワードリセット]** の順に移動します。
3. **[OSP OAuth リダイレクト URL]** フィールドで SSPR サーバの詳細を指定します。たとえば、`https://<SSPR Hostname IP>:port/sspr/public/oauth` です。
4. **[OK]** をクリックして変更を保存し、設定ユーティリティを閉じます。
5. 変更を有効にするために Tomcat を再起動します。

6 インストールを完了するための最終ステップ

Identity Manager コンポーネントのインストールと設定が完了したら、ご使用の環境でソリューションが適切に機能するように特定のタスクを実行する必要があります。たとえば、ビジネスプロセスで定義されたポリシーと要件を満たすようにインストールしたドライバを設定し、監査イベントを収集するように Sentinel Log Management for IGA を設定します。

インストール後のタスクには、通常次の項目が含まれます。

- 101 ページの 「Identity Vault の設定」
- 103 ページの 「アイデンティティボールト管理者としての非管理者ユーザの設定」
- 103 ページの 「リモートローダとドライバの設定」
- 103 ページの 「接続システムの設定」
- 108 ページの 「パスワードを忘れた場合の管理の設定」
- 113 ページの 「識別情報アプリケーションの設定」
- 148 ページの 「データ収集用のランタイム環境の設定」
- 158 ページの 「Identity Reporting の設定」
- 167 ページの 「root 以外のインストールの完了」
- 168 ページの 「Identity Manager のアクティベート」
- 168 ページの 「分散展開用の SSL 証明書の作成」

Identity Vault の設定

- Identity Vault の値インデックスの作成
- アイデンティティボールトへの Identity Applications および Identity Reporting 証明書の手動インポート

Identity Vault の値インデックスの作成

識別情報アプリケーションは、識別ボールト内のオブジェクトとデータをやり取りできる必要があります。Identity Applications のパフォーマンスを高めるために、アイデンティティボールト管理者は manager、および srvprvUUID の各属性に値インデックスを作成する必要があります。これらの属性に値インデックスがない場合、識別情報アプリケーションユーザは、特にクラスタ化環境で、パフォーマンスの低下を感じる可能性があります。

以下のいずれかの方法を使用して、Identity Applications のインストールが完了した後にこれらの値インデックスを作成できます。

- Identity Console。Index Manager を使用します。詳細については、『NetIQ eDirectory 管理ガイド』の「インデックスの作成」を参照してください。
- 環境設定ユーティリティ。その他 > [識別ボールトインデックス] に移動し、[サーバ DN] から [作成] を選択して値を指定します。[OK] をクリックし、アイデンティティボールトを再起動して変更を保存します。

アイデンティティボールトへの Identity Applications および Identity Reporting 証明書の手動インポート

- Identity Applications および Identity Reporting コンポーネントのカスタム証明書がある場合は、これらの証明書をアイデンティティボールトの cacerts (`/opt/netiq/common/jre/lib/security/cacerts`) にインポートします。

  たとえば、次の keytool コマンドを使用して、証明書をアイデンティティボールトにインポートすることができます。

  ```
  keytool -import -trustcacerts -alias <User Application certificate alias name> -keystore <cacerts file> -file <User Application certificate file>
  ```

- SSPR をユーザアプリケーションサーバとは異なるサーバにインストールする場合、SSPR アプリケーション証明書をユーザアプリケーションの idm.jks (`/opt/netiq/idm/apps/tomcat/conf/idm.jks`) にインポートします。

  たとえば、次の keytool コマンドを使用して、証明書をユーザアプリケーションにインポートすることができます。

  ```
  keytool -import -trustcacerts -alias <SSPR certificate alias name> -keystore <idm.jks> -file <SSPR certificate file>
  ```

アイデンティティボールト管理者としての非管理者ユーザの設定

非管理者ユーザをアイデンティティボールト管理者として使用するように Identity Applications が設定されている場合、非管理者ユーザは、ユーザが存在するサブツリーの oidpInstanceData 属性に対する [書き込み] 権限を持っている必要があります。それ以外の場合、OSP のログインが失敗する可能性があります。

非管理者ユーザの oidpInstanceData 属性に対する書き込み権限を設定する

1. Identity Console にログインします。
2. **[eDirectory]** ビューで、**[Rights Management (権限の管理)]** > **［Modify Trustees (トラスティの変更)]** をクリックします。
3. 非管理者ユーザオブジェクトを選択し、**[OK]** をクリックします。
4. トラスティが表示されるので、**[Assigned Rights (割り当てられた権限)]** をクリックします。
5. oidpInstanceData 属性に対して、**[比較]**、**[読み取り]**、および **[書き込み]** の権限を設定します。
6. **[適用]** をクリックして、変更を保存、および適用します。

リモートローダとドライバの設定

リモートローダにより、Identity Manager ドライバは、接続されたアプリケーションと同じサーバにアイデンティティボールトおよび Identity Manager エンジンをインストールすることなく、接続されたアプリケーションにアクセスできます。リモートローダを使用するには、Identity Manager エンジンと安全に接続できるようにアプリケーションシムを設定する必要があります。さらに、リモートローダと Identity Manager ドライバ両方の設定も必要です。この情報は、『*NetIQ Identity Manager Driver Administration Guide*』の「Configuring the Remote Loader and Drivers」に詳細に説明されています。

接続システムの設定

Identity Manager により、アプリケーション、ディレクトリ、およびデータベースで情報を共有できます。ドライバ固有の設定手順については、Identity Manager ドライバのマニュアル (https://www.netiq.com/documentation/identity-manager-49-drivers/) を参照してください。

ドライバセットの作成と設定

ドライバセットは、複数の Identity Manager ドライバを格納するコンテナです。1 つのサーバで一度にアクティブにできるドライバセットは 1 つだけです。ドライバセットを作成するには Designer ツールを使用できます。

アイデンティティボールトとのパスワード同期をサポートするためには、Identity Manager でドライバセットにパスワードポリシーが設定されている必要があります。Identity Manager でデフォルトのユニバーサルパスワードポリシーパッケージを使用するか、既存の組織の要件に基づいてパスワードポリシーを作成できます。ただし、パスワードポリシーには `DirMXL-PasswordPolicy` オブジェクトが含まれている必要があります。ポリシーオブジェクトにアイデンティティボールトが存在しない場合は、オブジェクトを作成できます。

- 104 ページの 「ドライバセットの作成」
- 104 ページの 「デフォルトのパスワードポリシーのドライバセットへの割り当て」
- 105 ページの 「アイデンティティボールトでのパスワードポリシーオブジェクトの作成」
- 106 ページの 「カスタムパスワードポリシーの作成」
- 106 ページの 「アイデンティティボールトでのデフォルト通知コレクションオブジェクトの作成」

ドライバセットの作成

Designer for Identity Manager では、ドライバセットを作成および設定するための多数の設定を用意しています。これらの設定により、グローバル環境設定値、ドライバセットパッケージ、ドライバセット名前付きパスワード、ログレベル、トレースレベル、および Java 環境パラメータを指定できます。詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「Configuring Driver Sets」を参照してください。

デフォルトのパスワードポリシーのドライバセットへの割り当て

アイデンティティボールトの各ドライバセットに DirMXL-PasswordPolicy オブジェクトを割り当てる必要があります。Identity Manager のデフォルトのユニバーサルパスワードポリシーパッケージには、このポリシーオブジェクトが含まれています。デフォルトのポリシーによりユニバーサルパスワードポリシーがインストールされて割り当てられ、Identity Manager エンジンがドライバ用にランダムパスワードを自動的に生成する方法を制御します。

または、カスタムパスワードポリシーを使用する場合は、パスワードポリシーオブジェクトとポリシーを作成する必要があります。詳細については、105 ページの 「アイデンティティボールトでのパスワードポリシーオブジェクトの作成」および 106 ページの 「カスタムパスワードポリシーの作成」を参照してください。

1. Designer でプロジェクトを開きます。
2. ［アウトライン］ペインで、プロジェクトを展開します。

3 [パッケージカタログ] > [共通] を展開し、デフォルトのユニバーサルパスワードポリシーパッケージが存在するかどうかを確認します。

4 (状況によって実行) パスワードポリシーパッケージが Designer に一覧表示されていない場合は、次の手順を実行します。

4a [パッケージカタログ] を右クリックします。

4b [パッケージのインポート] を選択します。

4c [Identity Manager Default Universal Password Policy (Identity Manager のデフォルトのユニバーサルパスワードポリシー)] を選択し、[OK] をクリックします。

表にすべての使用可能なパッケージが表示されるようにするには、[Show Base Packages Only (基本パッケージのみ表示)] の選択を解除する必要があります。

5 各ドライバセットを選択し、パスワードポリシーを割り当てます。

アイデンティティボールトでのパスワードポリシーオブジェクトの作成

DirMXL-PasswordPolicy オブジェクトがアイデンティティボールトに存在しない場合は、Designer または ldapmodify ユーティリティを使用してこのオブジェクトを作成できます。Designer でこれを実行する方法の詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「Configuring Driver Sets」を参照してください。ldapmodify ユーティリティを使用するには、次の手順を使用します。

1 テキストエディタで、次の属性を持つ LDAP Data Interchange Format (LDIF) ファイルを作成します。

```
dn: cn=DirXML-PasswordPolicy,cn=Password Policies,cn=Security
changetype: add
nsimPwdRuleEnforcement: FALSE
nspmSpecialAsLastCharacter: TRUE
nspmSpecialAsFirstCharacter: TRUE
nspmSpecialCharactersAllowed: TRUE
nspmNumericAsLastCharacter: TRUE
nspmNumericAsFirstCharacter: TRUE
nspmNumericCharactersAllowed: TRUE
nspmMaximumLength: 64
nspmConfigurationOptions: 596
passwordUniqueRequired: FALSE
passwordMinimumLength: 1
passwordAllowChange: TRUE
objectClass: nspmPasswordPolicy

dn: cn=DirXML-PasswordPolicy,cn=Password Policies,cn=Security
changetype: modify
add: nsimAssignments
nsimAssignments: <driverset LDAP dn>
```

注 : コンテンツをそのままコピーすると、ファイルにいくつかの非表示の特殊文字が挿入される場合があります。これらの属性をアイデンティティボールトに追加する際に `ldif_record() = 17` のエラーメッセージが表示される場合は、2 つの DN 間にスペースを挿入してください。

2 アイデンティティボールトに DirMXL-PasswordPolicy オブジェクトを追加するには、以下のアクションを実行して、ファイルから属性をインポートします。

ldapmodify ユーティリティを含むディレクトリから、次のコマンドを入力します。

```
ldapmodify -x -c -h hostname_or_IP_address -p 389 -D
"cn=admin,ou=sa,o=system" -w password -f path_to_ldif_file
```

次に例を示します。

```
ldapmodify -x -ZZ -c -h server1.test.com -p 389 -D
"cn=admin,ou=sa,o=system" -w test123 -f /root/dirxmlpasswordpolicy.ldif
```

ldapmodify ユーティリティは、デフォルトで /opt/novell/eDirectory/bin ディレクトリにあります。

カスタムパスワードポリシーの作成

Identity Manager でデフォルトのパスワードポリシーを使用するのではなく、組織の要件に基づいて新しいポリシーを作成できます。パスワードポリシーは、ツリー構造全体、パーティションルートコンテナ、コンテナ、または特定のユーザに割り当てることができます。管理を簡易化するには、ツリー内のできるだけ高い位置にパスワードポリシーを割り当てることをお勧めします。詳細については、『*Password Management 3.3.2 Administration Guide*』の「Creating Password Policies」を参照してください。

注 : ドライバセットに DirXML-PasswordPolicy オブジェクトを割り当てる必要もあります。詳細については、105 ページの 「アイデンティティボールトでのパスワードポリシーオブジェクトの作成」を参照してください。

アイデンティティボールトでのデフォルト通知コレクションオブジェクトの作成

デフォルトの通知コレクションは、電子メール通知テンプレートのセットとテンプレートから生成された電子メールを送信する際に使用される SMTP サーバを含むアイデンティティボールトオブジェクトです。デフォルトの通知コレクションオブジェクトがアイデンティティボールトに存在しない場合は、Designer を使用して作成できます。

1 Designer でプロジェクトを開きます。

2 ［アウトライン］ペインで、プロジェクトを展開します。

3 アイデンティティボールトを右クリックし、アイデンティティボールトの **[プロパティ]** をクリックします。

4 **[パッケージ]** をクリックし、**[Add Packages (パッケージの追加)]** アイコンをクリックします。

5 すべての通知テンプレートパッケージを選択し、**[OK]** をクリックします。

6 **[インストール]** 操作でパッケージをインストールするには、**[適用]** をクリックします。

7 通知テンプレートをアイデンティティボールトに展開します。

ドライバの作成

ドライバを作成するには、Designer で提供されているパッケージ管理機能を使用します。使用する各 Identity Manager ドライバに対して、ドライバオブジェクトを作成し、ドライバ設定をインポートします。ドライバオブジェクトには、設定パラメータとそのドライバのポリシーが含まれます。ドライバオブジェクトを作成する一貫として、ドライバパッケージをインストールしてから、環境に合わせてドライバ設定を変更します。

ドライバパッケージには、デフォルトのポリシーセットが含まれています。これらのポリシーは、データ共有モデルを簡単に実装できるようにすることを目的としています。ほとんどの場合は、出荷時のデフォルト設定を使用してドライバを設定してから、環境の要件に応じてドライバの設定を変更します。ドライバを作成して設定した後で、それをアイデンティティボールトに展開して起動します。一般的に、ドライバの作成プロセスには次のアクションが含まれます。

1. ドライバパッケージのインポート
2. ドライバパッケージのインストール
3. ドライバオブジェクトの設定
4. ドライバオブジェクトの展開
5. ドライバオブジェクトの起動

追加情報およびドライバ固有の情報については、Identity Manager ドライバの Web サイト (https://www.netiq.com/documentation/identity-manager-49-drivers/) から関連するドライバ実装ガイドを参照してください。

ポリシーの定義

ポリシーにより、識別ボールトに対する情報フローを特定の環境に合わせてカスタマイズできます。たとえば、ある会社ではメインのユーザクラスとして inetorgperson を使用していて、別の会社では User を使用しているとします。これを処理するために、各システムで呼び出すユーザを Identity Manager エンジンに指示するポリシーが作成されています。接続システム間でユーザに影響する操作をやり取りする場合、Identity Manager は、この変更を行うポリシーを適用します。

また、ポリシーは、新しいオブジェクトの作成、属性値の更新、スキーマ変換の実行、一致条件の定義、Identity Manager の関連付けの維持など、多くのタスクを実行します。

NetIQ では、Designer を使用して、ビジネスニーズを満たすようにドライバのポリシーを定義することをお勧めします。ポリシーの詳細ガイドについては、『*NetIQ Identity Manager - Using Designer to Create Policies* guide』および『*NetIQ Identity Manager Understanding Policies Guide*』を参照してください。Identity Manager が使用する文書型定義 (DTD) については、Identity Manager DTD Reference を参照してください。リソースには次のものが含まれます。

- 使用可 \'94\'5c な各ポリシーの詳細な説明
- 各条件、アクション、名詞、および動詞のサンプルと \'8d\'5c 文を含む、Policy Builder の詳細なユーザガイドとリファレンス
- XSLT スタイルシートを使用したポリシー作成の説明

パスワードを忘れた場合の管理の設定

Identity Manager インストールには、忘れたパスワードをリセットするプロセスの管理に役立つセルフサービスパスワードリセットが含まれています。それとは別に、外部パスワード管理システムを使用する方法があります。

- 108 ページの 「Self Service Password Reset によるパスワードを忘れた場合の管理」
- 110 ページの 「外部システムによるパスワードを忘れた場合の管理」
- 112 ページの 「分散環境またはクラスタ化環境におけるダッシュボードの SSPR リンクの更新」

Self Service Password Reset によるパスワードを忘れた場合の管理

SSPR と識別情報アプリケーションをインストールすると、通常はパスワードを忘れた場合の管理を有効にできます。ただし、パスワードを変更した後に SSPR がユーザを転送する識別情報アプリケーションのランディングページの URL を指定していない場合があります。また、パスワードを忘れた場合の管理を有効にする必要がある場合もあります。この節では、次のトピックについて説明します。

- 108 ページの 「セルフサービスパスワードリセットを使用するための Identity Manager の設定」
- 109 ページの 「Identity Manager で使用するためのセルフサービスパスワードリセットの設定」
- 109 ページの 「SSPR 設定のロック」

セルフサービスパスワードリセットを使用するための Identity Manager の設定

このセクションでは、SSPR を使用するために Identity Manager を設定する方法について説明します。

1. 識別情報アプリケーションをインストールしたサーバにログインします。
2. RBPM 設定ユーティリティを実行します。詳細については、114 ページの 「識別情報アプリケーション設定ユーティリティの実行」を参照してください。
3. ユーティリティで [認証] > [パスワードの管理] の順に移動します。
4. [パスワード管理プロバイダ] では [SSPR] を指定します。
5. [パスワードを忘れた場合] を選択します。
6. [SSO クライアント] > [セルフサービスパスワードリセット] の順に移動します。
7. [OSP client ID (OSP クライアント ID)] では認証サーバに認識させる SSPR のシングルサインオンクライアントの名前を指定します。デフォルト値は `sspr` です。
8. [OSP client secret (OSP クライアントシークレット)] では SSPR のシングルサインオンクライアントのパスワードを指定します。

9 [OSP redirect URL (OSP リダイレクト URL)] では認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

次の形式を使用を使用します : `protocol://server:port/path`。たとえば、`http://10.10.10.48:8180/sspr/public/oauth` です。

10 変更を保存して、ユーティリティを閉じます。

Identity Manager で使用するためのセルフサービスパスワードリセットの設定

このセクションでは、Identity Manager と一緒に使用するために SSPR を設定する方法について説明します。たとえば、パスワードポリシーや秘密の質問の答えの質問を変更できます。

Identity Manager と一緒に SSPR をインストールしたときに、管理者がこのアプリケーションを設定するために使用できるパスワードを指定しました。SSPR 設定を変更してから、管理者アカウントまたは管理者グループが SSPR を設定できるように指定することをお勧めします。

注 : ユーザアプリケーションサーバとは異なるサーバに SSPR をインストールする場合は、SSPR アプリケーション証明書がユーザアプリケーション `cacerts` に追加されていることを確認します。

---

1 インストール時に指定した設定パスワードを使用して、SSPR にログインします。

2 [設定] ページで、パスワードポリシーと秘密の質問の答えの質問の設定を変更します。SSPR 設定のデフォルト値を設定する方法の詳細については、『*NetIQ Self Service Password Reset Administration Guide* (https://www.netiq.com/documentation/self-service-password-reset-47/sspr-adminguide/data/bookinfo.html)』の「Configuring Self Service Password Reset (https://wwwtest.netiq.com/documentation/self-service-password-reset-47/sspr-adminguide/data/b14gnrxl.html)」を参照してください。

3 SSPR 設定ファイル (`SSPRConfiguartion.xml`) をロックします。設定ファイルのロックの詳細については、109 ページの 「SSPR 設定のロック」を参照してください。

4 (オプション) 設定をロックした後で SSPR 設定を変更するには、`SSPRConfiguartion.xml` ファイルで `configIsEditable` 設定を `true` に設定する必要があります。

5 SSPR からログアウトします。

6 変更を有効にするには、Tomcat を再起動します。

SSPR 設定のロック

1 [http://<IP/DNS name>:<port>/sspr] にアクセスします。SSPR ポータルに移動します。

2 Identity Manager に管理者アカウントでログインするか、または既存のログイン資格情報でログインします。

3 ページ上部の [Configuration Manager (環境設定マネージャ)] をクリックし、インストール時に指定した設定パスワードを指定します。

4 [Configuration Editor (環境設定エディタ)] をクリックし、[設定] > [LDAP Settings (LDAP 設定)] の順に移動します。

5 SSPR 設定ファイル (SSPRConfiguartion.xml) をロックします。

5a [Administrator Permission (管理者許可)] セクションで、識別ボールト内の SSPR に対する管理者権限を持つユーザまたはグループを表すフィルタを LDAP フォーマットで定義します。デフォルトでは、このフィルタは groupMembership=cn=Admins,ou=Groups,o=example と設定されています。

たとえば、ユーザアプリケーション管理者の場合は「uaadmin (cn=uaadmin)」と設定します。

これにより、設定を変更するためのすべての権限を持つ SSPR 管理者ユーザを除いて、ユーザは SSPR で設定を変更できなくなります。

5b LDAP クエリが結果を返していることを確認するために、[View Matches (一致の表示)] をクリックします。

設定にエラーがある場合は、次の設定オプションに進めません。SSPR は問題のトラブルシューティングに役立つエラー詳細を表示します。

5c [保存] をクリックします。

5d ポップアップされる確認ウィンドウで [OK] をクリックします。

SSPR がロックされている間は、管理者ユーザが表示する管理インタフェースには、[Dashboard (ダッシュボード)] 、[User Activity (ユーザアクティビティ)] 、[Data Analysis (データ分析)] など、SSPR がロックされる前は表示されていなかった追加オプションが表示されます。

6 (オプション) 設定をロックした後で SSPR 設定を変更するには、SSPRConfiguartion.xml ファイルで configIsEditable 設定を true に設定する必要があります。

7 SSPR からログアウトします。

8 ステップ 3 で定義されている管理者ユーザとして SSPR に再ログインします。

9 [Close Configuration (設定を閉じる)] をクリックし、[OK] をクリックして変更を確認します。

10 変更を有効にするには、Tomcat を再起動します。

外部システムによるパスワードを忘れた場合の管理

外部システムを使用するには、パスワードを忘れた場合機能を含む WAR ファイルの場所を指定する必要があります。このプロセスには次の作業が含まれます。

- 111 ページの 「外部からパスワードを忘れた場合を管理する WAR ファイルの指定」
- 112 ページの 「外部パスワードを忘れた場合の環境設定のテスト」
- 112 ページの 「アプリケーションサーバ間の SSL 通信の設定」

外部からパスワードを忘れた場合を管理する WAR ファイルの指定

インストール時にこれらの値を指定せずに、後で設定を変更する場合、RBPM 設定ユーティリティを使用するか、ユーザアプリケーションで管理者として変更します。

1 (状況によって実行) RBPM 設定ユーティリティで設定を変更するには、次の手順を実行します。

   1a 識別情報アプリケーションをインストールしたサーバにログインします。

   1b RBPM 設定ユーティリティを実行します。詳細については、114 ページの 「識別情報アプリケーション設定ユーティリティの実行」を参照してください。

   1c ユーティリティで **[認証］ > ［パスワードの管理]** の順に移動します。

   1d **[パスワード管理プロバイダ]** では **[ユーザアプリケーション (レガシ)]** を指定します。

2 (状況によって実行) ユーザアプリケーションで設定を変更するには、次の手順を実行します。

   2a ユーザアプリケーションの管理者としてログインします。

   2b **[管理］ > ［アプリケーション環境設定］ > ［パスワードモジュールのセットアップ］ > ［ログイン]** の順に移動します。

3 ［**パスワードを忘れた場合**］では、**[外部]** を指定します。

4 **[パスワードを忘れた場合リンク]** では、ログインページでユーザが［**パスワードを忘れた場合**］をクリックしたときに表示するリンクを指定します。ユーザがこのリンクをクリックすると、アプリケーションはユーザを外部パスワード管理システムに転送します。次に例を示します。

   ```
   http://localhost:8180/ExternalPwd/jsps/pwdmgt/ForgotPassword.jsp
   ```

5 **[パスワードを忘れた場合の返信リンク]** では、ユーザがパスワードを忘れた場合の手順の実行完了後に表示するリンクを指定します。このリンクをクリックすると、指定したリンクにリダイレクトされます。次に例を示します。

   ```
   http://localhost/IDMProv
   ```

6 **[パスワードを忘れた場合の Web サービス URL]** では、パスワードを忘れた場合の外部 WAR が識別情報アプリケーションを呼び戻すために使用する Web サービスの URL を指定します。次の形式を使用してください。

   ```
   https://idmhost:sslport/idm/pwdmgt/service
   ```

   識別情報アプリケーションに対してセキュアな Web サービス通信を保証するために、返信リンクでは SSL を使用する必要があります。詳細については、112 ページの 「アプリケーションサーバ間の SSL 通信の設定」を参照してください。

7 `ExternalPwd.war` を、外部パスワード WAR 機能を実行するリモートアプリケーションサーバ展開ディレクトリに手動でコピーします。

外部パスワードを忘れた場合の環境設定のテスト

外部パスワード WAR ファイルがあり、これにアクセスして［パスワードを忘れた場合］機能をテストする場合は、次の場所でアクセスできます。

- ブラウザ内で直接アクセスします。外部パスワード WAR ファイルで［パスワードを忘れた場合］ページに移動します。たとえば、`http://localhost:8180/ExternalPwd/jsps/pwdmgt/ForgotPassword.jsp` に移動します。
- ユーザアプリケーションログインページで、**[パスワードを忘れた場合]** のリンクをクリックします。

アプリケーションサーバ間の SSL 通信の設定

外部パスワード管理システムを使用する場合、アイデンティティアプリケーションおよびパスワードを忘れた場合の外部管理 WAR ファイルを展開している Tomcat インスタンス間に SSL 通信を設定する必要があります。詳細については、Tomcat マニュアルを参照してください。

分散環境またはクラスタ化環境におけるダッシュボードの SSPR リンクの更新

インストールプロセスは、識別情報アプリケーションおよび Identity Reporting と同じアプリケーションサーバ上に SSPR が展開されていると想定しています。デフォルトでは、ダッシュボードの **[アプリケーション]** ページにある組み込みリンクは、ローカルシステム上の SSPR を参照する相対 URL フォーマットを使用します。たとえば、`\sspr\private\changepassword` です。分散環境またはクラスタ化環境にアプリケーションをインストールする場合、SSPR リンクの URL を更新する必要があります。

詳細については、*アイデンティティアプリケーションのヘルプ*を参照してください。

1. ダッシュボードに管理者としてログインします。たとえば、`uaadmin` としてログインします。
2. **[編集]** をクリックします。
3. ［Edit Home Items (ホームアイテムの編集)］ページで、更新するアイテムの上にマウスを移動し、編集アイコンをクリックします。たとえば、**[マイパスワードの変更]** を選択します。
4. **[リンク]** では絶対 URL を指定します。たとえば、「`http://10.10.10.48:8180/sspr/changepassword`」と指定します。
5. **[保存]** をクリックします。
6. 更新する SSPR リンクごとにこの手順を繰り返します。
7. 終了したら、**[I'm done (完了)]** をクリックします。
8. ログアウトしてから一般ユーザとしてログインし、変更されているかどうかをテストします。

識別情報アプリケーションの設定

- 113 ページの 「Configupdate ユーティリティのパスワードの保存」
- 114 ページの 「識別情報アプリケーションの設定の管理」
- 142 ページの 「パーミッションインデックスの場所の指定」
- 143 ページの 「識別情報アプリケーション用の REST API の展開」
- 143 ページの 「Identity Applications を使用した Oracle データベースの設定」
- 144 ページの 「Oracle サービス名を使用した Oracle データベースへのアクセス」
- 144 ページの 「手動によるデータベーススキーマの作成」
- 146 ページの 「識別情報アプリケーションのシングルサインオン設定の管理」
- 147 ページの 「識別情報アプリケーションの起動」
- 147 ページの 「識別情報アプリケーションの設定と使用方法の検討事項」

Configupdate ユーティリティのパスワードの保存

Configupdate の起動中に、マスタキーストア (encrypt-keys.pkcs12) のパスワードを入力するよう求められます。デフォルトでは、このパスワードは `changeit` です。このキーストアは、`/opt/netiq/idm/apps/tomcat/conf` にあります。

パスワードを変更して再暗号化するには、次の手順を実行します。

1. 次のコマンドを実行して、Configupdate ユーティリティのパスワードを変更します。

   ```
   keytool -storepasswd -new <new master keystore password> -keystore /
   opt/netiq/idm/apps/tomcat/conf/encrypt-keys.pkcs12 -storepass changeit
   -storetype PKCS12
   ```

2. 新しいパスワードを暗号化するために、次のコマンドを実行します。

   ```
   /opt/netiq/common/jre/bin/java -cp /opt/netiq/idm/apps/tomcat/lib/
   obscurity-<version>-uber.jar
   com.netiq.ism.obfuscate.server.util.PasswordObfuscator <new master
   keystore password>
   ```

3. 必ず、`/opt/netiq/idm/apps/tomcat/conf` にある `ism-sensitive.properties` ファイルの `com.microfocus.ism.master-storepass` の値を変更して、暗号化されたパスワードを保存してください。

識別情報アプリケーションの設定の管理

識別情報アプリケーション設定ユーティリティを使用して、ユーザアプリケーションドライバと識別情報アプリケーションの設定を管理できます。識別情報アプリケーションのインストールプログラムは、アプリケーションの設定にかかる時間を短縮するために、このユーティリティを呼び出します。これらの設定のほとんどは、インストール後にも変更できます。

設定ユーティリティ (configupdate.sh) を実行するファイルは、デフォルトで /opt/netiq/idm/apps/configupdate ディレクトリにあります。

注

- configudate.sh は、configupdate ディレクトリからのみ実行してください。カスタムの場所から configupdate.sh を実行すると失敗します。
- クラスタの場合、クラスタのすべてのメンバで環境設定を同一にする必要があるため、ism-sensitive.properties、ism-configuration.properties、および encrypt-keys.pkcs12 をプライマリノードからすべてのユーザアプリケーションノードにコピーします。

このセクションでは、設定ユーティリティの設定について説明します。これらの設定は複数のタブとして編成されています。Identity Reporting をインストールする場合、インストールプロセスはユーティリティに Identity Reporting 用のパラメータを追加します。

識別情報アプリケーション設定ユーティリティの実行

1 configupdate.sh で次のオプションが正しく設定されていることを確認します。

```
edit_admin="true"
use_console="false"
```

注 : ユーティリティをコンソールモードで実行する必要がある場合のみ、use_console の値を true に設定する必要があります。

2 configupdate.sh.properties を保存して閉じます。

3 コマンドプロンプトで次のコマンドを実行して、設定ユーティリティを実行します。

```
./configupdate.sh
```

注 : ユーティリティが起動するまで数分待つ必要がある場合があります。

注 : Identity Manager は NAM OAuth をサポートしていないため、設定ユーティリティの［Authentication Server (認証サーバ)］タブの **[Access Manager is the OAuth provider (Access Manager は OAuth プロバイダ)]** オプションは使用しないでください。

User Application Parameters (ユーザアプリケーションのパラメータ)

識別情報アプリケーションを設定する際、このタブでは、識別情報アプリケーションが識別ボールトと通信する場合に使用する値を定義します。いくつかの設定は、インストールプロセスを完了するために必須です。

デフォルトでは、このタブには基本オプションが表示されます。すべての設定を表示するには、**[詳細オプションの表示]** をクリックします。このタブには、次の設定グループがあります。

- 115 ページの 「識別ボールト設定」
- 116 ページの 「識別ボールト DN」
- 119 ページの 「識別ボールトユーザ ID」
- 120 ページの 「識別ボールトユーザグループ」
- 121 ページの 「識別ボールト証明書」
- 121 ページの 「電子メールサーバ設定」
- 124 ページの 「トラステッドキーストア」
- 124 ページの 「NetIQ Sentinel デジタル署名証明書 & キー」
- 124 ページの 「その他」
- 126 ページの 「コンテナオブジェクト」

識別ボールト設定

このセクションでは、識別情報アプリケーションが識別ボールト内のユーザの識別情報と役割にアクセスできるようにする設定を定義します。いくつかの設定は、インストールプロセスを完了するために必須です。

識別ボールトサーバ

必須

LDAP サーバのホスト名または IP アドレスを指定します。たとえば、「`myLDAPhost`」と指定します。

LDAP ポート

識別ボールトが平文の LDAP 要求をリスンするポートを指定します。デフォルトは 389 です。

LDAP セキュアポート

識別ボールトが SSL (Secure Sockets Layer) プロトコルを使用した LDAP 要求をリスンするポートを指定します。デフォルトは 636 です。

eDirectory がインストールされる前にサーバにロードされているサービスがデフォルトのポート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。

識別ボールト管理者

必須

LDAP 管理者の資格情報を指定します。たとえば、「cn=admin」というようになります。このユーザは識別ボールトにすでに存在している必要があります。

識別情報アプリケーションはこのアカウントを使用して、識別ボールトへの管理接続を行います。この値は、マスタキーに基づいて暗号化されます。

識別ボールト管理者パスワード

必須

LDAP 管理者のパスワードを指定します。このパスワードは、マスタキーに基づいて暗号化されます。

パブリック匿名アカウントの使用

ログインしていないユーザが LDAP パブリック匿名アカウントにアクセスできるかどうかを指定します。

セキュア管理者接続

RBPM が管理者アカウント関連のすべての通信で SSL プロトコルを使用するかどうかを指定します。この設定を行っても、SSL を必要としない他の処理は SSL を使用せずに処理を実行できます。

注 : このオプションを選択すると、パフォーマンスが低下する可能性があります。

セキュアなユーザ接続

RBPM がログインユーザアカウント関連のすべての通信で TLS/SSL プロトコルを使用するかどうかを指定します。この設定を行っても、TLS/SSL を必要としない他の処理は TLS/SSL を使用せずに動作できます。

注 : このオプションを選択すると、パフォーマンスが低下する可能性があります。

識別ボールト DN

このセクションでは、識別情報アプリケーションと Identity Manager の他のコンポーネントの間で通信できるようにするコンテナとユーザアカウントの識別名を定義します。いくつかの設定は、インストールプロセスを完了するために必須です。

ルートコンテナ DN

必須

ルートコンテナの LDAP 識別名を指定します。これは、ディレクトリ抽象化層で検索ルートが指定されない場合に、デフォルトのエンティティ定義検索ルートとして使用されます。たとえば、「o=mycompany」と指定します。

ユーザコンテナ DN

必須

詳細オプションを表示すると、このパラメータは［識別ボールトユーザ識別情報］に表示されます。

ユーザコンテナの LDAP 識別名 (DN) または完全修飾 LDAP 名を指定します。この設定には次の考慮事項が適用されます。

- このコンテナ (とその下位) のユーザは、識別情報アプリケーションへのログインを許可されます。
- Identity Applications をホストする Tomcat を起動したことがある場合、`configupdate.sh` ファイルを使用してこの設定を変更することはできません。
- このコンテナには、ユーザがユーザアプリケーションドライバを設定したときに指定したユーザアプリケーション管理者が含まれている必要があります。含まれていない場合、指定したアカウントはワークフローを実行できません。

グループコンテナ DN

必須

詳細オプションを表示すると、このパラメータは［識別ボールトユーザグループ］に表示されます。

グループコンテナの LDAP 識別名 (DN) または完全修飾 LDAP 名を指定します。この設定には次の考慮事項が適用されます。

- ディレクトリ抽象化レイヤ内のエンティティ定義では、この DN を使用します。
- Identity Applications をホストする Tomcat を起動したことがある場合、`configupdate.sh` ファイルを使用してこの設定を変更することはできません。

ユーザアプリケーションドライバ

必須

ユーザアプリケーションドライバの識別名を指定します。

たとえば、ドライバが UserApplicationDriver、ドライバセットの名前が myDriverSet、ドライバセットのコンテキストが o=myCompany である場合、「`cn=UserApplicationDriver,cn=myDriverSet,o=myCompany`」と指定します。

ユーザアプリケーション管理者

必須

ユーザアプリケーションの指定したユーザコンテナの管理タスクを実行する権限を持つ識別ボールト内の既存のユーザアカウントを指定します。この設定には次の考慮事項が適用されます。

- ユーザアプリケーションをホストする Tomcat を起動したことがある場合、`configupdate.sh` ファイルを使用してこの設定を変更することはできません。
- ユーザアプリケーションの展開後にこの割り当てを変更するには、ユーザアプリケーションの **[管理］ > ［セキュリティ]** ページを使用します。

- このユーザアカウントは、ユーザアプリケーションの [管理] タブを使用してポータルを管理する権利を持ちます。
- ユーザアプリケーション管理者が、Designer、またはユーザアプリケーション ([要求と承認] タブ) に公開されているワークフロー管理タスクに参加する場合は、この管理者に、ユーザアプリケーションドライバに含まれるオブジェクトインスタンスに対する適切なトラスティ権限を与える必要があります。詳細については、『*User Application Administration Guide*』を参照してください。

プロビジョニング管理者

ユーザアプリケーション全体で使用可能なプロビジョニングワークフロー機能を管理する識別ボールト内の既存のユーザアカウントを指定します。

ユーザアプリケーションを展開した後でこの割り当てを変更するには、ユーザアプリケーションの [管理］ >［管理者の割り当て] ページを使用します。

整合性管理者

[コンプライアンス] タブのすべての機能を実行することをメンバーに許可するシステム役割を実行する識別ボールト内の既存のアカウントを指定します。この設定には次の考慮事項が適用されます。

- 識別情報アプリケーションを展開した後にこの割り当てを変更するには、ユーザアプリケーションの [管理］ >［管理者の割り当て] ページを使用します。
- 設定を更新する際、この値に対する変更が有効になるのは、有効なコンプライアンス管理者が割り当てられていない場合のみです。有効なコンプライアンス管理者が存在する場合は、変更は保存されません。

役割管理者

任意の役割の作成、削除、または変更および任意のユーザ、グループ、またはコンテナへの役割割り当ての付与または取消をメンバーに許可する役割を指定します。さらに役割のメンバーは、任意のユーザに対してレポートを実行できます。この設定には次の考慮事項が適用されます。

- デフォルトでは、この役割にはユーザアプリケーション管理者が割り当てられています。
- 識別情報アプリケーションを展開した後にこの割り当てを変更するには、ユーザアプリケーションの [管理］ >［管理者の割り当て] ページを使用します。
- 設定を更新する際、この値に対する変更が有効になるのは、有効な役割管理者が割り当てられていない場合のみです。有効な役割管理者が存在する場合は、変更は保存されません。

セキュリティ管理者

セキュリティドメイン内のすべての機能をメンバーに付与する役割を指定します。この設定には次の考慮事項が適用されます。

- セキュリティ管理者は、セキュリティドメイン内のすべてのオブジェクトで可能なアクションをすべて実行できます。セキュリティドメインを使用すると、セキュリティ管理者は RBPM 内のすべてのドメインのすべてのオブジェクトのアク

セス許可を設定できます。セキュリティ管理者はチームを構成でき、またドメイン管理者、委任管理者、およびその他のセキュリティ管理者も割り当てることができます。

- 識別情報アプリケーションを展開した後にこの割り当てを変更するには、ユーザアプリケーションの **[管理] > [管理者の割り当て]** ページを使用します。

リソース管理者

リソースドメイン内のすべての機能をメンバーに付与する役割を指定します。この設定には次の考慮事項が適用されます。

- リソース管理者はリソースドメイン内のすべてのオブジェクトで可能なアクションをすべて実行できます。
- 識別情報アプリケーションを展開した後にこの割り当てを変更するには、ユーザアプリケーションの **[管理] > [管理者の割り当て]** ページを使用します。

RBPM 設定管理者

構成ドメイン内のすべての機能をメンバーに付与する役割を指定します。この設定には次の考慮事項が適用されます。

- RBPM 設定管理者は、構成ドメイン内のすべてのオブジェクトで可能なアクションをすべて実行できます。RBPM 設定管理者は、RBPM 内のナビゲーション項目へのアクセスを制御します。また、RBPM 設定管理者は委任と代理サービス、ユーザインタフェースのプロビジョニング、およびワークフローエンジンを設定します。
- 識別情報アプリケーションを展開した後にこの割り当てを変更するには、ユーザアプリケーションの **[管理] > [管理者の割り当て]** ページを使用します。

RBPM レポーティング管理者

レポーティング管理者を指定します。デフォルトでは、インストールプログラムが他のセキュリティフィールドと同じユーザをこの値に表示します。

識別ボールトユーザ ID

このセクションでは、識別情報アプリケーションが識別ボールト内のユーザコンテナと通信できるようにする値を定義します。いくつかの設定は、インストールプロセスを完了するために必須です。

このセクションの設定は、**[詳細オプションの表示]** を選択した場合のみ表示されます。

ユーザコンテナ DN

必須

詳細オプションを表示していない場合、このパラメータは［識別ボールト DN］に表示されます。

ユーザコンテナの LDAP 識別名 (DN) または完全修飾 LDAP 名を指定します。この設定には次の考慮事項が適用されます。

- このコンテナ (とその下位) のユーザは、識別情報アプリケーションへのログインを許可されます。

- Identity Applications をホストする Tomcat を起動したことがある場合、configupdate.sh ファイルを使用してこの設定を変更することはできません。
- このコンテナには、ユーザがユーザアプリケーションドライバを設定したときに指定したユーザアプリケーション管理者が含まれている必要があります。含まれていない場合、指定したアカウントはワークフローを実行できません。

ユーザ検索スコープ

識別ボールトユーザがコンテナを検索できるスコープの深さを指定します。

ユーザオブジェクトクラス

LDAP ユーザのオブジェクトクラスを指定します。通常は inetOrgPerson です。

ログイン属性

ユーザのログイン名を表す LDAP 属性を指定します。たとえば、「cn」と指定します。

名前付け属性

ユーザまたはグループをルックアップする際に識別子として使用する LDAP 属性を指定します。これはログイン属性とは異なります。ログイン属性はログイン時にのみ使用されます。たとえば、「cn」と指定します。

ユーザメンバーシップ属性

(オプション) ユーザのグループメンバーシップを表す LDAP 属性を指定します。この名前を指定する際、スペースを使用しないでください。

識別ボールトユーザグループ

このセクションでは、識別情報アプリケーションが識別ボールト内のグループコンテナと通信できるようにする値を定義します。いくつかの設定は、インストールプロセスを完了するために必須です。

このセクションの設定は、**[詳細オプションの表示]** を選択した場合のみ表示されます。

グループコンテナ DN

必須

詳細オプションを表示していない場合、このパラメータは［識別ボールト DN］に表示されます。

グループコンテナの LDAP 識別名 (DN) または完全修飾 LDAP 名を指定します。この設定には次の考慮事項が適用されます。

- ディレクトリ抽象化レイヤ内のエンティティ定義では、この DN を使用します。
- Identity Applications をホストする Tomcat を起動したことがある場合、configupdate.sh ファイルを使用してこの設定を変更することはできません。

グループコンテナのスコープ

識別ボールトユーザがグループコンテナを検索できるスコープの深さを指定します。

グループオブジェクトクラス

LDAP グループのオブジェクトクラスを指定します。通常は groupofNames です。

グループメンバーシップ属性

(オプション) ユーザのグループメンバーシップを指定します。この名前にはスペースを使用しないでください。

ダイナミックグループの使用

ダイナミックグループを使用するかどうかを指定します。

[ダイナミックグループオブジェクトクラス] の値も指定する必要があります。

ダイナミックグループオブジェクトクラス

[ダイナミックグループの使用] を選択している場合のみ適用されます。

LDAP ダイナミックグループのオブジェクトクラスを指定します。通常は `dynamicGroup` です。

識別ボールト証明書

このセクションでは、JRE キーストアのパスとパスワードを定義します。いくつかの設定は、インストールプロセスを完了するために必須です。

キーストアパス

必須

Tomcat が動作するために使用している JRE のキーストア (`cacerts`) ファイルへのフルパスを指定します。手動でパスを入力するか、または `cacerts` ファイルを参照して指定できます。この設定には次の考慮事項が適用されます。

- 現在の環境における RBPM のインストールディレクトリを指定する必要があります。デフォルト値は正しい場所に設定されます。
- 識別情報アプリケーションのインストールプログラムは、キーストアファイルを変更します。Linux では、ユーザにこのファイルへの書き込み許可が必要です。

キーストアパスワード

必須

キーストアファイルのパスワードを指定します。デフォルトは、「`changeit`」です。

電子メールサーバ設定

このセクションでは、電子メールベースの承認に使用できる、電子メール通知を有効にする値を定義します。

通知テンプレートホスト

アイデンティティアプリケーションをホストする Tomcat の名前または IP アドレスを指定します。たとえば、「`myapplication serverServer`」と指定します。

この値は、電子メールテンプレートの $HOST$ トークンと置き換えられます。インストールプログラムはこの情報を使用して、プロビジョニング要求タスクと承認通知を参照する URL を作成します。

通知テンプレート PORT

アイデンティティアプリケーションをホストする Tomcat のポート番号を指定します。

プロビジョニング要求タスクと承認通知で使用する電子メールテンプレートの $PORT$ トークンがこの値で置き換えられます。

通知テンプレートセキュアポート

アイデンティティアプリケーションをホストする Tomcat のセキュアポート番号を指定します。

プロビジョニング要求タスクと承認通知で使用する電子メールテンプレートの $SECURE_PORT$ トークンがこの値で置き換えられます。

通知テンプレートプロトコル

ユーザの電子メールを送信する際に URL に使用する非セキュアプロトコルを指定します。たとえば、「`http`」と指定します。

プロビジョニング要求タスクと承認通知で使用する電子メールテンプレートの $PROTOCOL$ トークンがこの値で置き換えられます。

通知テンプレートセキュアプロトコル

ユーザの電子メールを送信する際に URL に使用するセキュアプロトコルを指定します。たとえば、「`https`」と指定します。

プロビジョニング要求タスクと承認通知で使用する電子メールテンプレートの $SECURE_PROTOCOL$ トークンがこの値で置き換えられます。

通知 SMTP 電子メール送信者

識別情報アプリケーションが電子メール通知を送信するために使用する電子メールアカウントを指定します。

SMTP サーバ名

識別情報アプリケーションがプロビジョニング電子メールに使用する SMTP 電子メールホストの IP アドレスまたは DNS 名を指定します。`localhost` は使用しないでください。

サーバには認証が必要です

サーバに認証が必要かどうかを指定します。

電子メールサーバに対する資格情報も指定する必要があります。

ユーザ名

[サーバには認証が必要です] を有効にした場合にのみ適用されます。

電子メールサーバのログインアカウントの名前を指定します。

[Password (パスワード)]

[サーバには認証が必要です] を有効にした場合にのみ適用されます。

メールサーバのログインアカウントのパスワードを指定します。

SMTP TLS の使用

メールサーバ間の転送中に電子メールメッセージのコンテンツをセキュリティ保護するかどうかを指定します。

電子メール通知イメージの場所

電子メール通知に添付するイメージへのパスを指定します。

Identity Applications サーバと電子メールサーバの両方がセキュアな接続を使用するように設定されている場合は、次の条件が満たされていることを確認します。

- Identity Applications サーバと電子メールサーバ間のセキュアな接続を確立するために使用される証明書は、信頼できる CA 証明書です。
- イメージパスに `https` を使用します。たとえば、`https://localhost:8543/IDMProv/images` です。

Identity Applications がプレーンテキスト通信に http を使用するサーバ上で動作している場合は、イメージパスの `https` を `http` に置き換えます。イメージパスの例 : `http://localhost:8080/IDMProv/images`

Sign email (電子メールの署名)

送信メッセージにデジタル署名を追加するかどうかを指定します。

このオプションを有効にする場合は、キーストアと署名キーの設定も指定する必要があります。

キーストアパス

[Sign email (電子メールの署名)] を有効にした場合にのみ適用されます。

電子メールをデジタルで署名するために使用するキーストア (`cacerts`) ファイルへのフルパスを指定します。手動でパスを入力するか、または `cacerts` ファイルを参照して指定できます。

たとえば、`/opt/netiq/idm/apps/jre/lib/security/cacerts` です。

キーストアパスワード

[Sign email (電子メールの署名)] を有効にした場合にのみ適用されます。

キーストアファイルのパスワードを指定します。たとえば、`changeit` です。

Alias of signature key (署名キーのエイリアス)

[Sign email (電子メールの署名)] を有効にした場合にのみ適用されます。

キーストアの署名キーの別名を指定します。たとえば、`idmapptest` です。

Signature key password (署名キーのパスワード)

[Sign email (電子メールの署名)] を有効にした場合にのみ適用されます。

署名キーを含むファイルを保護するパスワードを指定します。たとえば、`changeit` です。

トラステッドキーストア

このセクションでは、識別情報アプリケーションのトラステッドキーストアの値を定義します。このセクションの設定は、**[詳細オプションの表示]** を選択した場合のみ表示されます。

トラステッドストアパス

信頼される署名者のすべての証明書が含まれるトラステッドキーストアへのパスを指定します。入力しない場合は、識別情報アプリケーションはシステムプロパティ `javax.net.ssl.trustStore` からパスを取得します。このシステムプロパティからパスを取得できない場合、インストールプログラムはデフォルトで `jre/lib/security/cacerts` に設定します。

トラステッドストアパスワード

トラステッドキーストアのパスワードを指定します。入力しない場合は、識別情報アプリケーションはシステムプロパティ `javax.net.ssl.trustStorePassword` からパスワードを取得します。このシステムプロパティからパスを取得できない場合、インストールプログラムはデフォルトで `changeit` に設定します。

このパスワードは、マスタキーに基づいて暗号化されます。

トラステッドストアタイプ

トラステッドストアパスがデジタル署名に Java キーストア (JKS) または PKCS12 のどちらを使用するかを指定します。

NetIQ Sentinel デジタル署名証明書 & キー

このセクションでは、Identity Manager がイベント監査のために Sentinel と通信できるようにする値を定義します。このセクションの設定は、**[詳細オプションの表示]** を選択した場合のみ表示されます。

Sentinel デジタル署名証明書

Sentinel に送信される監査メッセージを OAuth サーバが認証するために使用するカスタム公開鍵証明書が表示されます。

Sentinel デジタル署名秘密鍵

Sentinel に送信される監査メッセージを OAuth サーバが認証するために使用するカスタム秘密鍵ファイルへのパスを指定します。

その他

このセクションの設定は、**[詳細オプションの表示]** を選択した場合のみ表示されます。

OCSP URI

クライアントインストールがオンライン証明書状態プロトコル (OCSP) を使用する際に使用する Uniform Resource Identifier(URI) を指定します。たとえば、「`http://host:port/ocspLocal`」と指定します。

OCSP URI によって、トラステッド証明書オンラインの状態は更新されます。

許可設定パス

許可環境設定ファイルの完全修飾名を指定します。

識別ボールトインデックス

識別情報アプリケーションのパフォーマンスを高めるために、manager、ismanager、および srvprvUUID の各属性に値インデックスを作成することができます。

Identity Applications のインストールが完了した後、設定ユーティリティを使用して値インデックスを作成できます。この設定には次の考慮事項が適用されます。

- これらの属性にインデックスがない場合、Identity Applications ユーザは、Identity Applications のパフォーマンスの低下を感じる可能性があります。
- Identity Applications をインストールした後、Identity Console を使用して、手動でこれらのインデックスを作成できます。
- パフォーマンスを最大化するには、インストール時にインデックスを作成する必要があります。
- 識別情報アプリケーションをユーザが使用できるようにする前に、これらのインデックスをオンラインモードにする必要があります。
- インデックスを作成するには、**[サーバ DN]** 設定で **[作成]** を選択し、**[サーバ DN]** の値を指定します。**[OK]** をクリックし、識別ボールトを再起動して変更を有効にします。
- インデックスを削除するには、**[サーバ DN]** 設定で **[削除]** を選択し、**[サーバ DN]** の値を指定します。**[OK]** をクリックし、識別ボールトを再起動して変更を有効にします。

サーバ DN

識別ボールトインデックスを作成または削除する必要がある場合にのみ適用されます。

インデックスを作成または削除する eDirectory サーバを指定します。

指定できるサーバは一度に 1 つだけです。複数の eDirectory サーバでインデックスを設定するには、RBPM 設定ユーティリティを複数回実行する必要があります。

Reinitialize RBPM Security (RBPM セキュリティの再初期化)

インストールプロセスの終了時に RBPM セキュリティをリセットするかどうかを指定します。識別情報アプリケーションを再展開する必要もあります。

IDMReport URL

Identity Manager Reporting モジュールの URL を指定します。たとえば、「`http://hostname:port/IDMRPT`」と指定します。

カスタムテーマのコンテキスト名

ブラウザで Identity Applications を表示する際に使用するカスタマイズしたテーマの名前を指定します。

ログメッセージの識別子プレフィックス

`idmuserapp_logging.xml` ファイルの CONSOLE アペンダと FILE アペンダのレイアウトパターンで使用する値を指定します。デフォルト値は `RBPM` です。

RBPM コンテキスト名の変更

RBPM のコンテキスト名を変更するかどうかを指定します。

役割とリソースドライバの新しい名前と DN も指定する必要があります。

RBPM コンテキスト名

[RBPM コンテキスト名の変更] *を選択している場合にのみ適用されます。*

RBPM の新しいコンテキスト名を指定します。

役割ドライバの DN

[RBPM コンテキスト名の変更] *を選択している場合にのみ適用されます。*

役割とリソースドライバの DN を指定します。

コンテナオブジェクト

このセクションのパラメータはインストール時にのみ適用されます。

このセクションでは、コンテナオブジェクトの値を定義したり、新しいコンテナオブジェクトを作成したりできます。

Selected(選択済み)

使用するコンテナオブジェクトタイプを指定します。

コンテナオブジェクトタイプ

コンテナの地域、国、部門、組織、またはドメインを指定します。

Identity Console 内で自分のコンテナを定義して、**[新規コンテナオブジェクトの追加]** の下に追加することもできます。

コンテナ属性名

指定したコンテナオブジェクトタイプに関連付けられている属性タイプの名前を指定します。

新規コンテナオブジェクトの追加 : コンテナオブジェクトタイプ

新しいコンテナとして使用可能な識別ボールトのオブジェクトクラスの LDAP 名を指定します。

新規コンテナオブジェクトの追加 : コンテナ属性名

新しいコンテナオブジェクトタイプに関連付けられている属性タイプの名前を指定します。

Reporting Parameters (Reporting パラメータ)

識別情報アプリケーションを設定する際、このタブでは、Identity Reporting を管理するための値を定義します。Identity Reporting をインストールすると、このタブが追加されます。

デフォルトでは、このタブには基本オプションが表示されます。すべての設定を表示するには、**[詳細オプションの表示]** をクリックします。このタブには、次の設定グループがあります。

- 127 ページの 「電子メール配信設定」
- 128 ページの 「レポート保持の値」
- 128 ページの 「ロケールの変更」
- 128 ページの 「役割の設定」
- 128 ページの 「送信プロキシ」

電子メール配信設定

このセクションでは、通知を送信するための値を定義します。

SMTP サーバホスト名

Identity Reporting が通知を送信する際に使用する電子メールサーバの DNS 名または IP アドレスを指定します。`localhost` は使用しないでください。

SMTP サーバポート

SMTP サーバのポート番号を指定します。

SMTP は SSL を使用

電子メールサーバとの通信に TLS/SSL プロトコルを使用するかどうかを指定します。

サーバは認証が必要

電子メールサーバとの通信に認証を使用するかどうかを指定します。

SMTP ユーザ名

認証に使用する電子メールアドレスを指定します。

値を指定する必要があります。サーバで認証が不要の場合は、無効なアドレスを指定できます。

SMTP ユーザパスワード

サーバは認証が必要と指定している場合にのみ適用されます。

SMTP ユーザアカウントのパスワードを指定します。

デフォルト電子メールアドレス

電子メールサーバとの通信に認証を使用するかどうかを指定します。

レポート保持の値

このセクションでは、完了したレポートを保持するための値を定義します。

レポートの単位 , レポート有効期間

Identity Reporting が完了したレポートを保持する期間を指定します。この期間が経過すると、完了したレポートは削除されます。たとえば、6 カ月を指定するには、[レポート有効期間] フィールドに「6」と入力し、[レポートの単位] フィールドで [月] を選択します。

レポートの場所

レポート定義を保存する場所のパスを指定します。たとえば、「/opt/netiq/IdentityReporting」と指定します。

ロケールの変更

このセクションでは、Identity Reporting の使用言語に関する値を定義します。Identity Reporting は特定のロケールを使用して検索します。詳細については、『Administrator Guide to NetIQ Identity Reporting』を参照してください。

役割の設定

このセクションでは、Identity Reporting がレポートを生成する際に使用する認証ソースに関する値を定義します。

認証ソースの追加

レポーティングのために追加する認証ソースのタイプを指定します。次の認証ソースを指定できます。

- **デフォルト**
- **LDAP ディレクトリ**
- **File (ファイル)**

送信プロキシ

「Identity Manager 4.8.1 以降のバージョンを使用する場合にのみ適用されます。」

このセクションでは、Identity Reporting がレポートのダウンロードに使用するリバースプロキシサーバを使用するための値を定義します。

Use Proxy

レポーティングでリバースプロキシサーバを使用するオプションを指定します。

- **ホスト名または IP アドレス**
- **ポート**
- **Use TLS (TLS の使用)**

  ネットワークプロトコルとして TCP を使用する場合にのみ適用されます。

認証パラメータ

アイデンティティアプリケーションを設定する際、このタブでは、Tomcat がユーザをアイデンティティアプリケーションおよびパスワード管理のページに転送するために使用する値を定義します。

デフォルトでは、このタブには基本オプションが表示されます。すべての設定を表示するには、**[詳細オプションの表示]** をクリックします。このタブには、次の設定グループがあります。

- 129 ページの 「[Authentication Server(認証サーバ)]」
- 130 ページの 「認証の設定」
- 130 ページの 「認証方式」
- 133 ページの 「パスワード管理」
- 134 ページの 「Sentinel デジタル署名証明書とキー」

[Authentication Server(認証サーバ)]

このセクションでは、識別情報アプリケーションが認証サーバに接続するための設定を定義します。

OAuth サーバのホスト識別子

必須

トークンを OSP に発行する認証サーバの相対 URL を指定します。たとえば、「192.168.0.1」と指定します。

OAuth サーバの TCP ポート

認証サーバのポートを指定します。

Access Manager is the OAuth provider (Access Manager は OAuth プロバイダです)

OAuth の OSP から NAM への変換は、設定更新ユーティリティの認証タブからはサポートされていません。このオプションを非表示にするには、`configupdate.sh.properties` ファイルで、`no_nam_oauth` の値を “true” に設定します。

OAuth サーバは TLS/SSL を使用しています

認証サーバが通信に TLS/SSL を使用するかどうかを指定します。

オプションの TLS/SSL トラストストアファイル

[OAuth サーバは TLS/SSL を使用しています] *を選択し、詳細オプションを表示している場合にのみ適用されます。*

オプションの TLS/SSL トラストストアパスワード

[OAuth サーバは TLS/SSL を使用しています] *を選択し、詳細オプションを表示している場合にのみ適用されます。*

TLS/SSL 認証サーバのキーストアファイルをロードする際に使用するパスワードを指定します。

注 : キーストアパスおよびパスワードを指定せず、認証サーバの信頼証明書が JRE トラストストア (cacerts) に存在しない場合、Identity Applications は TLS/SSL プロトコルを使用する認証サービスに接続できません。

認証の設定

このセクションでは、認証サーバの設定を定義します。

管理コンテナの LDAP DN

必須

OSP が認証する必要がある管理者ユーザオブジェクトが含まれる識別ボールト内のコンテナの識別名を指定します。たとえば、「`ou=sa,o=data`」と指定します。

重複解決名前付け属性

同じ `cn` 値を持つ複数の eDirectory ユーザオブジェクトを区別するために使用する LDAP 属性の名前を指定します。デフォルト値は `mail` です。

コンテキストへの認証ソースの制限

識別ボールト内のユーザコンテナおよび管理者コンテナにおける検索を、そのコンテナ内のユーザオブジェクトのみに限定するのか、それともサブコンテナも検索対象にするのかを指定します。

セッションタイムアウト (分)

ユーザが何も操作せずに一定時間が経過するとサーバはそのユーザセッションをタイムアウトさせますが、その時間を分単位で指定します。デフォルト値は 20 分です。

アクセストークンのライフタイム (秒)

OSP アクセストークンの有効期間の秒数を指定します。デフォルト値は 60 秒です。

リフレッシュトークンのライフタイム (時間)

OSP リフレッシュトークンの有効期間の秒数を指定します。リフレッシュトークンは OSP が内部的に使用します。既定値は 48 時間です。

認証方式

このセクションでは、Identity Manager のブラウザベースのコンポーネントにログインするユーザを OSP が認証できるようにする値を定義します。

メソッド

ユーザがログオンする際に Identity Manager が使用する認証タイプを指定します。

- ［名前とパスワード］ : OSP は識別ボールトを使用して認証を検証します。
- ［**Kerberos**］: OSP は Kerberos チケットサーバと識別ボールトの両方から認証を受け入れます。
- ［**SAML 2.0**］: OSP は SAML アイデンティティプロバイダとアイデンティティボールトの両方から認証を受け入れます。

reCAPTCHA を有効化

[名前とパスワード] を指定した場合にのみ適用されます。

ログインページで reCAPTCHA を有効にするかどうかを指定します。

reCAPTCHA は、ユーザがロボットではないことを確認するように要求することにより、セキュリティの追加レイヤを提供します。一致条件に基づいてユーザが選択する必要がある画像が表示されます。応答が成功すると、Access Manager はユーザの認証資格情報を認証します。応答が失敗した場合、Access Manager はユーザ資格情報を認証せず、ログインページにリダイレクトします。

2 要素認証を有効化

[名前とパスワード] を指定した場合にのみ適用されます。

2 要素認証を有効にするかどうかを指定します。

これには、[第 2 要素] タブでいくつかの設定を行う必要があります。詳細については、139 ページの 「第 2 要素パラメータ」 を参照してください。

マッピング属性名

[Kerberos] または [SAML] を指定している場合にのみ適用されます。

Kerberos チケットサーバまたは識別情報プロバイダの SAML 表現にマッピングする属性の名前を指定します。

フォールバックの reCAPTCHA を有効化

[Kerberos] を指定している場合にのみ適用されます。

Kerberos を使用できない場合に、フォールバックのユーザ名とパスワードを使用して reCAPTCHA を有効にするかどうかを指定します。

Number of attempts before required (必要になるまでの試行回数)

[フォールバックの reCAPTCHA を有効化] チェックボックスを選択した場合にのみ適用されます。

reCAPTCHA が有効になるまでのログイン試行の失敗回数を指定します。この値をゼロに設定すると、reCAPTCHA が常に必要であることを示します。

Site Key (サイト鍵)

[フォールバックの reCAPTCHA を有効化] チェックボックスを選択した場合にのみ適用されます。

Google reCAPTCHA Web サイトから取得した reCAPTCHA サイト鍵値を指定します。

Private Key(秘密鍵)

[フォールバックの reCAPTCHA を有効化] チェックボックスを選択した場合にのみ適用されます。

Google reCAPTCHA Web サイトから取得した reCAPTCHA 秘密鍵値を指定します。

フォールバックの 2 要素認証を有効化

[Kerberos] を指定している場合にのみ適用されます。

Kerberos を使用できない場合に、フォールバックのユーザ名とパスワードを使用して 2 要素認証を有効にするかどうかを指定します。

これには、[第 2 要素] タブでいくつかの設定を行う必要があります。詳細については、139 ページの 「第 2 要素パラメータ」 を参照してください。

ログアウト時にランディングページを使用

[Kerberos] を指定している場合にのみ適用されます。

ログアウトの成功後にログインページにリダイレクトするのではなく、ランディングページを有効にするかどうかを指定します。

ランディングページ

[SAML] *を指定している場合にのみ適用されます。*

- **なし** : ランディングページを使用しないことを指定します。IDP URL が示されている場合、このオプションを選択します。
- **内部** : 内部の OSP ランディングページを使用することを指定します。
- **外部** : 外部の OSP ランディングページにリダイレクトされることを指定します。

URL

[ランディングページ] フィールドで [外部] を選択した場合にのみ適用されます。

外部のランディングページの URL を指定します。

メタデータソース

[SAML] *を指定している場合にのみ適用されます。*

IDP メタデータのソースを指定します。URL からメタデータをロードするか、以前に取得したメタデータをコピーできます。

メタデータ URL

[メタデータ URL] フィールドに [URL] を指定した場合にのみ適用されます。

URL からメタデータをロードして、アプリケーションを終了する前に設定に保存するかどうかを指定します。

Load on save (保存時にロード)

[メタデータ URL] フィールドに [URL] を指定した場合にのみ適用されます。

OSP が認証要求を SAML にリダイレクトする際に使用する URL を指定します。

IDP メタデータ

[メタデータ URL] *フィールドに* [コピー / 貼り付け] を指定した場合にのみ適用されます。

SAML IDP から取得した、貼り付けるデータを指定します。

Configure Access Manager on exit (終了時に Access Manager を設定)

[メタデータ URL] *フィールドに* [コピー / 貼り付け] を指定した場合にのみ適用されます。

Access Manager で SAML サービスプロバイダ定義を自動的に設定するかどうかを指定します。

パスワード管理

このセクションでは、ユーザがパスワードの変更をセルフサービス操作として実行できるようにする値を定義します。

パスワード管理プロバイダ

使用するパスワード管理システムのタイプを指定します。

［**ユーザアプリケーション (レガシ)**］ : Identity Manager が従来使用していたパスワード管理プログラムを使用します。このオプションを使用すると、外部パスワード管理プログラムを使用することもできます。

パスワードを忘れた場合

このチェックボックスパラメータは、SSPR を使用する場合にのみ適用されます。

ユーザがパスワードを忘れた場合にヘルプデスクに連絡せずに回復するように設定するかどうかを指定します。

パスワードを忘れた場合の機能で秘密の質問の答えに関するポリシーも設定する必要があります。詳細については、『*NetIQ Self Service Password Reset Administration Guide* (https://wwwtest.netiq.com/documentation/self-service-password-reset-47/sspr-adminguide/data/bookinfo.html)』を参照してください。

パスワードを忘れた場合

このメニューリストは、[ユーザアプリケーション (レガシ)] *を選択している場合にのみ適用されます。*

ユーザアプリケーションまたは外部システムのどちらに統合されているパスワード管理システムを使用するかを指定します。

- ［**内部**］ : デフォルトの内部パスワード管理機能を使用します。`/jsps/pwdmgt/ForgotPassword.jsp`(最初は http(s) プロトコルなし)。これは、ユーザを、外部 WAR ではなく、ユーザアプリケーションに組み込まれた［パスワードを忘れた場合］機能にリダイレクトします。
- ［外部］ : パスワードを忘れた場合の外部 WAR を使用して、Web サービス経由でユーザアプリケーションを呼び戻します。外部システムの設定も指定する必要があります。

［パスワードを忘れた場合］リンク

外部パスワード管理システムを使用する場合にのみ適用されます。

パスワードを忘れた場合の機能ページを参照する URL を指定します。外部または内部のパスワード管理 WAR にある `ForgotPassword.jsp` ファイルを指定します。

パスワードを忘れた場合の返信リンク

外部パスワード管理システムを使用する場合にのみ適用されます。

ユーザがパスワードを忘れた場合の操作を実行した後でクリックできるように、[**パスワードを忘れた場合の返信リンク**] の URL を指定します。

パスワードを忘れた場合の Web サービス URL

外部パスワード管理システムを使用する場合にのみ適用されます。

パスワードを忘れた場合の外部 WAR がユーザアプリケーションを呼び戻してパスワードを忘れた場合のコア機能を実行するために使用する URL を指定します。次の形式を使用してください。

```
https://<idmhost>:<sslport>/<idm>/
pwdmgt/service
```

Sentinel デジタル署名証明書とキー

このセクションでは、Identity Manager がイベント監査のために Sentinel と通信できるようにする値を定義します。

Sentinel デジタル署名証明書

監査システムに送信される監査メッセージを OSP サーバが認証するために使用するカスタム公開鍵証明書を指定します。

Novell Audit で使用するために証明書を設定する方法の詳細については、『*Novell Audit Administration Guide*』の「Managing Certificates」を参照してください。

Sentinel デジタル署名秘密鍵

監査システムに送信される監査メッセージを OSP サーバが認証するために使用するカスタム秘密鍵ファイルへのパスを指定します。

SSO Clients Parameters (SSO クライアントパラメータ)

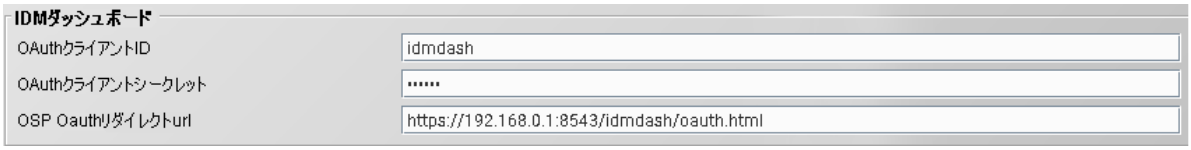
識別情報アプリケーションを設定する際、このタブでは、識別情報アプリケーションへのシングルサインオンアクセスを管理する値を定義します。

デフォルトでは、このタブには基本オプションが表示されます。すべての設定を表示するには、[**詳細オプションの表示**] をクリックします。このタブには、次の設定グループがあります。

- 135 ページの 「IDM ダッシュボード」
- 135 ページの 「IDM 管理者」
- 136 ページの 「RBPM」
- 137 ページの 「レポーティング」
- 138 ページの 「IDM データ収集サービス」
- 138 ページの 「DCS Driver (DCS ドライバ)」
- 138 ページの 「セルフサービスパスワードリセット」

IDM ダッシュボード

このセクションでは、ユーザが識別情報アプリケーションのプライマリログインの場所である、Identity Manager ダッシュボードにアクセスするために使用する必要がある URL の値を定義します。



OAuth クライアント ID

必須

ダッシュボードのシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。デフォルト値は `idmdash` です。

OAuth クライアントシークレット

必須

ダッシュボードのシングルサインオンクライアントのパスワードを指定します。

OSP OAuth リダイレクト URL

必須

認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/idmdash/oauth.html` です。

IDM 管理者

このセクションでは、ユーザが［Identity Manager 管理者］ページにアクセスするために必要な URL の値を定義します。

OAuth クライアント ID

必須

Identity Manager 管理者のシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。デフォルト値は `idmadmin` です。

OAuth クライアントシークレット

必須

［Identity Manager 管理者］のシングルサインオンクライアントのパスワードを指定します。

OSP OAuth リダイレクト URL

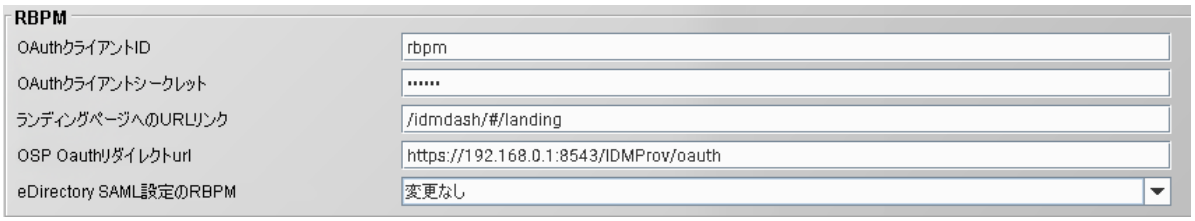
必須

認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/idmadmin/oauth.html` です。

RBPM

このセクションでは、ユーザがユーザアプリケーションにアクセスするために必要な URL の値を定義します。



OAuth クライアント ID

必須

ユーザアプリケーションのシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。デフォルト値は `rbpm` です。

OAuth クライアントシークレット

必須

ユーザアプリケーションのシングルサインオンクライアントのパスワードを指定します。

ランディングページへの URL リンク

必須

ユーザアプリケーションから［ダッシュボード］にアクセスするために使用する相対 URL を指定します。デフォルト値は `/landing` です。

OSP OAuth リダイレクト URL

必須

認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/IDMProv/oauth` です。

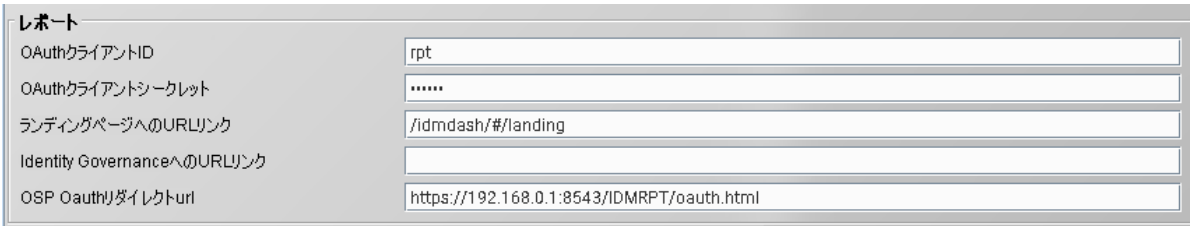
RBPM から eDirectory SAML への設定

必須

SSO 認証に必要な、RBPM から識別ボールト SAML への設定を指定します。

レポーティング

このセクションでは、ユーザが Identity Reporting にアクセスするために必要な URL の値を定義します。このセクションの値は、Identity Manager ソリューションに Identity Reporting を追加している場合にのみ表示されます。



OAuth クライアント ID

必須

Identity Reporting のシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。デフォルト値は `rpt` です。

OAuth クライアントシークレット

必須

Identity Reporting のシングルサインオンクライアントのパスワードを指定します。

ランディングページへの URL リンク

必須

Identity Reporting から［ダッシュボード］にアクセスするために使用する相対 URL を指定します。デフォルト値は `/idmdash/#/landing` です。

Identity Reporting と識別情報アプリケーションを異なるサーバにインストールしている場合は、絶対 URL を指定します。使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/IDMRPT/oauth` です。

OSP OAuth リダイレクト URL

必須

認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/IDMRPT/oauth` です。

IDM データ収集サービス

このセクションでは、ユーザが Identity Manager データ収集サービスにアクセスするために必要な URL の値を定義します。

OAuth クライアント ID

必須

Identity Manager データ収集サービスのシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。デフォルト値は `idmdcs` です。

OAuth クライアントシークレット

必須

Identity Manager データ収集サービスのシングルサインオンクライアントのパスワードを指定します。

OSP OAuth リダイレクト URL

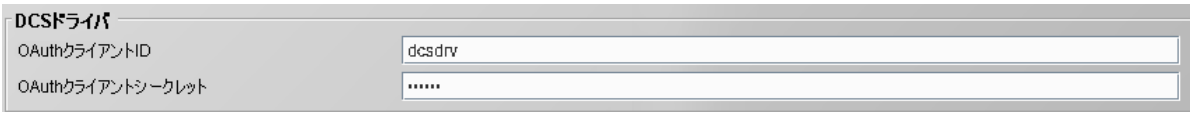
必須

認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/idmdcs/oauth.html` です。

DCS Driver (DCS ドライバ)

このセクションでは、データ収集サービスドライバを管理するための値を定義します。



OAuth クライアント ID

データ収集サービスドライバのシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。このパラメータのデフォルト値は `dcsdrv` です。

OAuth クライアントシークレット

データ収集サービスドライバのシングルサインオンクライアントのパスワードを指定します。

セルフサービスパスワードリセット

このセクションでは、ユーザが SSPR にアクセスするために必要な URL の値を定義します。

OAuth クライアント ID

必須

SSPR のシングルサインオンクライアントを認証サーバに認識させるために使用する名前を指定します。デフォルト値は `sspr` です。

OAuth クライアントシークレット

必須

SSPR のシングルサインオンクライアントのパスワードを指定します。

OSP OAuth リダイレクト URL

必須

認証完了時に認証サーバがブラウザクライアントをリダイレクトする絶対 URL を指定します。

使用するフォーマットは、`protocol://server:port/path` です。たとえば、`https://192.168.0.1:8543/sspr/public/oauth.html` です。

第 2 要素パラメータ

次に進む前に、Advanced Authentication で方法、チェーン、およびイベントを作成したことを確認してください。Advanced Authentication からの認証を受け入れるように OSP を設定する必要があります。

デフォルトでは、このタブには基本オプションが表示されます。すべての設定を表示するには、［詳細オプションの表示］をクリックします。このタブには、次の設定グループがあります。

AAF 管理者

このセクションでは、Advanced Authentication 管理者の設定を定義します。

管理者名 (リポジトリ \ 名前)

必須

OSP が Advanced Authentication とのインタフェースに使用する Advanced Authentication 管理者アカウントのリポジトリ修飾名を指定します。通常、このアカウントは LOCAL リポジトリにあります。

デフォルトの Advanced Authentication 管理者アカウントの名前は admin です。このアカウントを使用した場合、**[管理者名]** の値は次のとおりです。

LOCAL\admin (リポジトリ名 + \ + ユーザ名)

管理者パスワード

必須

上記で指定した Advanced Authentication 管理者ユーザのパスワードを指定します。

AAF ユーザリポジトリ

このセクションでは、Advanced Authentication ユーザリポジトリの設定を定義します。

ユーザリポジトリ名

必須

作成した Advanced Authentication のリポジトリの名前を指定します。このリポジトリは、Identity Manager 用のアイデンティティボールトに対応します。

AAF サーバ

このセクションでは、Advanced Authentication サーバの設定を定義します。

テスト TLS 証明書を許可

必須

AAF サーバからの無効なテスト証明書サブジェクトを無視するかどうかを指定します。これは、初期設定とテストにのみ適用されます。

[追加] をクリックし、Advanced Authentication サーバの DNS 名または IP アドレスを指定します。443 とは異なるポートを使用する場合は、そのポートも指定します。

(状況によって実行) Advanced Authentication サーバをクラスタ化した場合は、もう一度 **[追加]** をクリックして、クラスタ内の各サーバの各 DNS 名または IP アドレスを指定します。

チューニングパラメータを表示

必須

チューニングパラメータを有効にするかどうかを指定します。

ログアウトセッションのクリーンアップ (分): **[チューニングパラメータを表示]** チェックボックスを選択した場合にのみ適用されます。

アクティブな AAF ログインセッションがタイムアウトでクリーンアップの対象になっていないかどうかを検証する間隔を指定します。

ハートビート間隔 (ミリ秒): **[チューニングパラメータを表示]** チェックボックスを選択した場合にのみ適用されます。

AAF サーバにハートビート要求を送信して利用可否を確認する間隔を指定します。

AAF エンドポイント

このセクションでは、Advanced Authentication エンドポイントの設定を定義します。

新規エンドポイントの作成

必須

2 要素認証用の新しいエンドポイントを作成するかどうかを指定します。

識別子 : **[新規エンドポイントの作成]** チェックボックスを選択していない場合にのみ適用されます。

AAF 管理で設定されたエンドポイント識別子を指定します。

部外者秘 : **[新規エンドポイントの作成]** チェックボックスを選択していない場合にのみ適用されます。

AAF 管理で設定されたエンドポイントシークレットを指定します。

名前 : **[新規エンドポイントの作成]** チェックボックスを選択した場合にのみ適用されます。

AAF 管理ページでエンドポイントを識別するために使用される新しいエンドポイントの名前を指定します。

説明 : **[新規エンドポイントの作成]** チェックボックスを選択した場合にのみ適用されます。

上記で指定した新しいエンドポイントの説明を指定します。

第 2 要素条件

このセクションでは、第 2 要素条件の設定を定義します。

すべてのユーザ、常時

必須

すべてのユーザが常に第 2 要素認証を提供できるようにするかどうかを指定します。

ユーザログイン条件 **[すべてのユーザ、常時]** チェックボックスを選択していない場合にのみ適用されます。

Identity Manager で第 2 要素認証を使用するために特定の式と条件を定義できることを指定します。

第 2 要素認証方法

このセクションでは、Advanced Authentication 方法の設定を定義します。

さまざまな方法で第 2 要素認証を有効にするかどうかを指定します。

特定の方法に対して第 2 要素認証を無効にするには、方法名の横のチェックボックスを選択解除します。

ユーザが複数の方法で登録している場合、Identity Manager は第 2 要素方法の相対的な優先順位を使用します。

CEF 監査パラメータ

このセクションでは、シングルサインオンクライアントの CEF 監査パラメータを管理するための値を定義します。

Send audit events (監査イベントの送信)

監査イベントに CEF を使用するかどうかを指定します。

接続先のホスト

監査サーバの DNS 名または IP アドレスを指定します。

送信先ポート

監査サーバのポートを指定します。

ネットワークプロトコル

CEF イベントを受信するために監査サーバによって使用されるネットワークプロトコルを指定します。

Use TLS (TLS の使用)

ネットワークプロトコルとして TCP を使用する場合にのみ適用されます。

監査サーバが TCP と TLS を併用するように設定されているかどうかを指定します。**[TLS の使用]** > ［詳細オプションの表示］を選択し、**[Identity Manager Keystore file name (Identity Manager キーストアファイル名)]** と **[Identity Manager Keystore password (Identity Manager キーストアパスワード)]** を入力します。

Intermediate event store directory (中間イベントストアディレクトリ)

CEF イベントが監査サーバに送信される前のキャッシュディレクトリの場所を指定します。選択した中間イベントストアディレクトリを提供している場合はまず、許可と所有権がそのディレクトリに対して `novlua` に変更されていることを確認する必要があります。ディレクトリの許可を変更するには、次のコマンドを実行します。

```
chown novlua:novlua <directory_path>
```

```
chmod 755 <directory_path>
```

ここで、`<directory_path>` は中間イベントストアディレクトリへのパスです。

CEF 監査の設定に関する完全なドキュメントについては、『*NetIQ Identity Manager - Configuring Auditing in Identity Manager*』の「Configuring Identity Applications」を参照してください。

パーミッションインデックスの場所の指定

アイデンティティアプリケーションをインストールすると、Tomcat のパーミッションインデックスが作成されます。インデックスの場所を指定しない場合、一時ディレクトリにフォルダが作成されます。たとえば、Tomcat 上の `/opt/netiq/idm/apps/tomcat/temp/permindex` です。

通常は、テスト環境では場所は問題になりません。ただし、運用環境またはステージング環境では、一時ディレクトリにパーミッションインデックスを配置することが好ましくない場合があります。

インデックスの場所を指定するには

1. Tomcat を停止します。
2. テキストエディタで `ism-configuration.properties` ファイルを開きます。
3. ファイルの末尾に、次のテキストを追加します。

   ```
   com.netiq.idm.cis.indexdir = path/permindex
   ```

   例 :

   ```
   com.netiq.idm.cis.indexdir = /opt/netiq/idm/apps/tomcat/temp/permindex
   ```

4. ファイルを保存して閉じます。
5. 一時ディレクトリの既存の `permindex` フォルダを削除します。
6. Tomcat を起動します。

クラスタリングの許可インデックスを有効にするには、395 ページの第 32 章「Tomcat アプリケーションサーバ上へのアイデンティティアプリケーションクラスタ展開ソリューションのサンプル」を参照してください。

識別情報アプリケーション用の REST API の展開

識別情報アプリケーションコンポーネントには、識別情報アプリケーション内のさまざまな機能を有効にする複数の REST API が組み込まれています。REST サービスは、OAUTH2 プロトコルを使用して認証を提供します。ブラウザまたはスクリプトで curl コマンドを使用してこれらの API を呼び出して、管理タスクを自動化することができます。REST API および対応するドキュメントは、`idmappsdoc.war` ファイルで入手できます。war は、Identity Applications がインストールされるときに自動的に展開されます。詳細については、REST API のマニュアルを参照してください。

Identity Applications がインストールされているサーバで REST API ドキュメントにアクセスするには、ブラウザのアドレスバーで `https://<identity applications servername>:<Port>/idmappsdoc` を指定します。たとえば、`https://192.168.0.1:8543/idmappsdoc` です。

注 : ワークフローデータベーステーブルをエクスポートするには、`ism-configuration.properties` ファイルに以下のエントリを追加してください。

```
com.microfocus.workflow.migration.tables = <list-of-tables-to-exported-comma-separated>
```

次に例を示します。

```
com.microfocus.workflow.migration.tables = afmodel,afform,afprocess,afdocument,afactivity,afactivitytimertasks,afbranch,afcomment,afprovisioningstatus,afquorum,afworktask,configuration,email_approval_token,localization,processed_eba_mails
```

Identity Applications を使用した Oracle データベースの設定

Oracle データベースを設定するには、次の手順を実行します。

1. Oracle データベースの最新バージョンをダウンロードしてインストールします。

   データベースのデフォルトの SID は `orcl` です。

2. データベースインスタンスを作成します。

   2a コマンドラインから、次のコマンドを実行して、データベース設定アシスタントを起動します。

   ```
   dbca
   ```

   2b **[Global Database Name (グローバルデータベース名)]** フィールドで、データベース名を指定します。たとえば、`idappsdb` です。

3 データベースユーザを作成します。

3a Oracle SQL 開発者ツールにログインします。

3b 次のコマンドを実行します。

```
CREATE USER idmadmin IDENTIFIED BY <password>;
GRANT CONNECT, RESOURCE to idmadmin;
ALTER USER idmadmin quota 100M on USERS;
```

ここで、

idmadmin は、ユーザアカウントを表し、

<password> は、ユーザパスワードを表します。

4 手順 2 と 3 を繰り返して、ワークフローエンジンデータベースを設定します。

注 : Oracle データベースを使用して Identity Applications を設定する場合は、先に述べたセクションで指定した詳細情報と同じ情報を使用してください。

Oracle サービス名を使用した Oracle データベースへのアクセス

Oracle System ID (SID) および Oracle サービス名を使用して、Oracle データベースに接続することができます。サービス名を使用してデータベースにアクセスする場合は、SID を介して接続して、1 つのデータベースインスタンスへの Identity Applications インストールを完了します。インストールが完了したら、以下のアクションを実行します。

1 次のコマンドを実行して、Oracle データベースにサービス名を作成します。

```
alter system set service_names='SERVICE1' scope=both sid='*';
```

ここで、SERVICE 1 は Oracle サービスの名前です。

注 : サービス名は大文字でも小文字でも指定できます。大文字と小文字は区別されません。

---

2 Tomcat の server.xml ファイルで、ファイル内の Oracle データソースの詳細を変更して、サービス名を定義します。

```
url="jdbc:oracle:thin:@IP:PORT/service1"
```

3 Tomcat を再起動します。

4 サービス名が catalina.out ログファイルに含まれていることを確認します。

5 Identity Applications がデータベースに適切に接続されていることを確認します。

手動によるデータベーススキーマの作成

識別情報アプリケーションをインストールする際、データベースへの接続またはデータベーステーブルの作成を後回しにできます。データベースに対する許可を持たないユーザは、このオプションを選択する必要がある場合があります。インストールプログラムは、

ユーザがデータベーススキーマを作成するために使用できる SQL ファイルを作成します。インストール後に、再インストールを行わずに、データベーステーブルを再作成することもできます。それには、識別情報アプリケーションのデータベースを削除して、それと同じ名前で新しいデータベースを作成します。

SQL ファイルによるデータベーススキーマの生成

このセクションでは、ユーザがデータベーススキーマを生成するために使用できる SQL ファイルをインストールプログラムが作成していると想定しています。この SQL ファイルが存在しない場合は、145 ページの 「データベーススキーマを生成する SQL ファイルの手動による作成」を参照してください。

注 : この SQL ファイルを SQL*Plus で実行しないでください。このファイルの行長は 4000 文字を超えています。

---

1. アプリケーションサーバを停止します。
2. データベースサーバにログインします。
3. 識別情報アプリケーションが使用するデータベースを削除します。
4. ステップ 3 で削除したデータベースと同じ名前で新しいデータベースを作成します。
5. インストールプロセスで作成された SQL スクリプトのある場所に移動します。デフォルトでは、*/installation_path*/userapp/sql ディレクトリにあります。
6. (状況によって実行) Oracle データベースの場合、関数 CONCAT_BLOB の定義の後にバックスラッシュ (/) を挿入します。次に例を示します。

```
  -- Changeset icfg-data-load.xml::700::IDMRBPM
CREATE OR REPLACE FUNCTION CONCAT_BLOB(A IN BLOB, B IN BLOB) RETURN BLOB
AS
                C BLOB;
            BEGIN
                DBMS_LOB.CREATETEMPORARY(C, TRUE);
                DBMS_LOB.APPEND(C, A);
                DBMS_LOB.APPEND(C, B);
                RETURN c;
            END;
/
```

7. データベース管理者に、ユーザアプリケーションデータベースを作成および設定する SQL スクリプトを実行させます。
8. Tomcat を再起動します。

データベーススキーマを生成する SQL ファイルの手動による作成

インストール後に、SQL ファイルがなくても、再インストールを行わずに、データベーステーブルを再作成できます。このセクションでは、SQL ファイルを持たない場合にデータベーススキーマを作成する方法について説明します。

1. Tomcat を停止します。

2 識別情報アプリケーションデータベースをホストするサーバにログインします。

3 既存のデータベースを削除します。

4 ステップ 3 で削除したデータベースと同じ名前で新しいデータベースを作成します。

5 テキストエディタで NetIQ-Custom-Install.log ファイルを開きます。このファイルは、デフォルトでは、識別情報アプリケーションのインストールディレクトリのルートにあります。次に例を示します。

```
/opt/netiq/idm/apps/UserApplication
```

6 NetIQ-Custom-Install.log ファイルで次のコマンドを検索し、コピーします。

```
/opt/netiq/common/jre/bin/java -Xms256m -Xmx256m  -
Dwar.context.name=IDMProv -Ddriver.dn="cn=User Application
Driver,cn=driverset1,o=system" -Duser.container="o=data" -jar /opt/
netiq/idm/apps/UserApplication/liquibase.jar --
databaseClass=liquibase.database.core.PostgresDatabase --
driver=org.postgresql.Driver --classpath=/opt/netiq/idm/apps/
postgresql/postgresql-9.4.1212jdbc42.jar opt/netiq/idm/apps/
UserApplication/IDMProv.war --changeLogFile=DatabaseChangeLog.xml  --
url="jdbc:postgresql://localhost:5432/idmuserappdb" --
contexts="prov,newdb" --logLevel=info --logFile=/opt/netiq/idm/apps/
UserApplication/db.out --username******** --password******** update
```

7 識別情報アプリケーションで使用するデータベースをインストールしたサーバにログインします。

8 端末でコピーしたコマンド文字列を貼り付けます。

注 : 正しいコマンドは updateSQL です。update だった場合は updateSQL に変更してください。

---

9 コマンドでデータベースユーザ名とパスワードを表すアスタリスク (*) を、認証に必要な実際の値で置き換えます。また、SQL ファイルの名前が一意であることを確認します。

10 コマンドを実行します。

11 (状況によって実行) インストールプロセスでデータベースにデータを設定せずに SQL ファイルを生成した場合、データベース管理者にそのファイルを渡してデータベースサーバにインポートします。詳細については、145 ページの 「SQL ファイルによるデータベーススキーマの生成」を参照してください。

12 データベース管理者が SQL ファイルをインポートした後、Tomcat を起動します。

識別情報アプリケーションのシングルサインオン設定の管理

インストールプロセスにより、Identity Manager のシングルサインオンアクセス用の認証サービス (OSP) がインストールされます。ただし、OSP 認証サーバを設定して、Kerberos チケットサーバまたは SAML から認証を受け入れることもできます。インストー

ル後に識別情報アプリケーションのシングルサインオン設定を行うには、『*NetIQ Identity Manager - Administrator's Guide to the Identity Applications*』の「Configuring Single Sign-on Access in Identity Manager」を参照してください。

識別情報アプリケーションの起動

Identity Applications を設定した後は、Tomcat サービスと ActiveMQ サービスを再開してください。

```
systemctl restart netiq-tomcat.service

systemctl restart netiq-activemq.service
```

識別情報アプリケーションの設定と使用方法の検討事項

識別情報アプリケーションを設定および初めて使用する際、次の検討事項が適用されます。

- インストールプロセス中、インストールプログラムによりログファイルがインストールディレクトリに書き込まれます。これらのファイルには、設定に関する情報が含まれています。現在の識別情報アプリケーション環境を設定した後で、これらのログファイルを削除するか、安全な場所に保存することを検討する必要があります。インストールプロセス中、データベーススキーマをファイルに書き込むことも選択できます。このファイルにはデータベースについての説明的な情報が含まれているので、インストールプロセスが完了した後で、安全な場所に移動する必要があります。
- (状況によって実行) アイデンティティアプリケーションを監査するには、現在の環境に Identity Reporting と監査サービスがインストールされ、イベントをキャプチャするように設定されている必要があります。また、識別情報アプリケーションを監査用に設定する必要があります。
- ユーザが識別情報アプリケーションにアクセスできるようにするには、次のアクティビティを完了する必要があります。
  - 必要なすべての Identity Manager ドライバがインストールされていることを確認します。
  - 識別ボールトのインデックスがオンラインモードであることを確認します。インストール時またはインストール後にインデックスを設定する方法の詳細については、102 ページの 「Identity Vault の値インデックスの作成」を参照してください。
  - すべてのブラウザで cookie を有効にします。cookie が無効な場合、アプリケーションは機能しません。
- 異なるサーバ上に Identity Applications および SSPR がインストールされている場合は、Identity Applications サーバの `idm.jks` に SSPR トラステッド証明書をインポートする必要があります。

データ収集用のランタイム環境の設定

このセクションでは、ランタイム環境が正常に動作していることを確認するために実行する必要がある追加の設定手順について説明します。さらに、トラブルシューティングの手法に加え、注意を要するデータベーステーブルに関する情報についても説明します。

このプロセスには次の作業が含まれます。

- 148 ページの 「識別情報アプリケーションからのデータ収集に関するデータ収集サービスドライバの設定」
- 149 ページの 「データ収集サービスドライバの移行」
- 151 ページの 「カスタム属性とカスタムオブジェクトのサポートの追加」
- 154 ページの 「複数のドライバセットのサポートの追加」
- 156 ページの 「SSL を使用したリモートモードでのドライバ実行設定」

理解が困難な問題が 1 つ以上のドライバで発生する場合は、『*NetIQ Identity Reporting Module Guide*』の「Troubleshooting the Drivers」を参照してください。

識別情報アプリケーションからのデータ収集に関するデータ収集サービスドライバの設定

識別情報アプリケーションを Identity Reporting と適切に連携させるには、OAuth プロトコルをサポートするように DCS ドライバを設定する必要があります。

注

- 現在の環境で Identity Reporting を使用している場合は、DCS ドライバをインストールして設定するだけで済みます。
- 現在の環境で DCS ドライバが複数設定されている場合は、各ドライバに対して次の手順を実行する必要があります。

---

1. Designer にログインします。
2. Designer でプロジェクトを開きます。
3. (状況によって実行) DCS ドライバをサポートされているパッチバージョンにまだアップグレードしていない場合は、次の手順を実行します。
   - **3a** 最新の DCS ドライバパッチファイルをダウンロードします。
   - **3b** パッチファイルをサーバ上の場所に展開します。
   - **3c** ターミナルで、ご使用の環境用のパッチ RPM を展開した場所へ移動し、次のコマンドを実行します。

     ```
     rpm -Uvh novell-DXMLdcs.rpm
     ```

   - **3d** アイデンティティボールトを再起動します。

3e Designer で、サポートされているバージョンのデータ収集サービスベースパッケージがインストールされていることを確認します。必要に応じて、続行する前に最新バージョンをインストールします。ソフトウェア要件の詳細については、59 ページの 「Identity Reporting コンポーネントのインストールに関する考慮事項」を参照してください。

3f Designer で DCS ドライバを再展開して再起動します。

4 ［**アウトライン**］ビューで、DCS ドライバを右クリックし、[プロパティ] を選択します。

5 [ドライバ環境設定] をクリックします。

6 [ドライバパラメータ] タブをクリックします。

7 [Show connection parameters (接続パラメータの表示)] をクリックし、[show (表示)] を選択します。

8 [SSO Service Support (SSO サービスのサポート)] をクリックし、[はい] を選択します。

9 Identity Reporting の IP アドレスとポートを指定します。

10 SSO サービスクライアントのパスワードを指定します。デフォルトのパスワードは `driver` です。

11 [適用] をクリックし、[OK] をクリックします。

12 [アウトライン] ビューで、DCS ドライバを右クリックし、[プロパティ] > ［展開] の順に選択します。

13 [展開] をクリックします。

14 DCS ドライバの再起動を求めるプロンプトが表示される場合は、[はい] をクリックします。

15 [OK] をクリックします。

データ収集サービスドライバの移行

オブジェクトを識別情報ウェアハウスに同期するには、データ収集サービスドライバを移行する必要があります。

1 Identity Console にログインします。

2 [IDM Administration (IDM 管理)] タイルで、データ収集サービスドライバを選択し、[Data Transformation and Synchronization (データ変換と同期)] に移動します。

3 データ同期ビューで、[Migrate From Identity Vault (アイデンティティボールトからの移行)] を選択します。

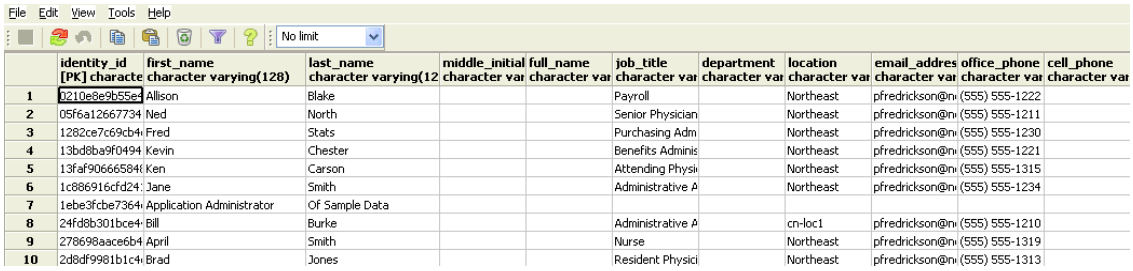
4 関連データが含まれる組織を選択して、[開始] をクリックします。

注 : 保存されているデータの量によっては、マイグレーションプロセスに数分かかる場合があります。次に進む前にマイグレーションプロセスが完了するまで待ってください。

---

5 マイグレーションプロセスが終わるまでしばらく待ちます。

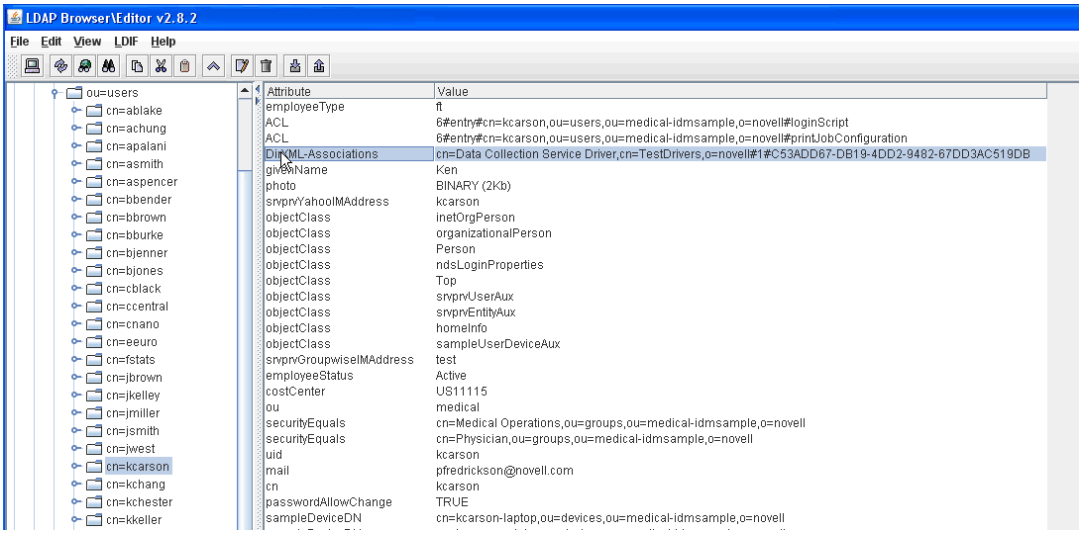
6 識別ボールト内にある識別情報とアカウントの情報を提供する [idmrpt_identity] テーブルおよび [idmrpt_acct] テーブルに、次のタイプの情報が含まれていることを確認します。



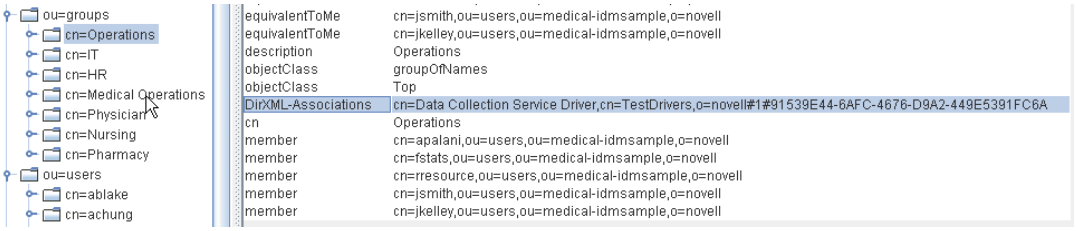
	identity_id [PK] characte	first_name character varying(128)	last_name character varying(12	middle_initial character var	full_name character var	job_title character var	department character var	location character var	email_addres character var	office_phone character var	cell_phone character var
1	0210e8e9b55e4	Allison	Blake			Payroll		Northeast	pfredrickson@n	(555) 555-1222	
2	05f6a12667734	Ned	North			Senior Physician		Northeast	pfredrickson@n	(555) 555-1211	
3	1282ce7c69cb4	Fred	Stats			Purchasing Adm		Northeast	pfredrickson@n	(555) 555-1230	
4	13bd8ba9f0494	Kevin	Chester			Benefits Adminis		Northeast	pfredrickson@n	(555) 555-1221	
5	13faf906665846	Ken	Carson			Attending Physi		Northeast	pfredrickson@n	(555) 555-1315	
6	1c886916cfd24	Jane	Smith			Administrative A		Northeast	pfredrickson@n	(555) 555-1234	
7	1ebe3fcbe7364	Application Administrator	Of Sample Data								
8	24fd8b301bce4	Bill	Burke			Administrative A		cn-loc1	pfredrickson@n	(555) 555-1210	
9	278698aace6b4	April	Smith			Nurse		Northeast	pfredrickson@n	(555) 555-1319	
10	2d8df9981b1c4	Brad	Jones			Resident Physici		Northeast	pfredrickson@n	(555) 555-1313	

7 LDAP ブラウザで、マイグレーションプロセスによって［DirXML-Associations (DirXML 関連付け)］に次の参照が追加されていることを確認します。

- 各ユーザについて次のタイプの情報を確認します。



- 各グループについて次のタイプの情報を確認します。



8 [idmrpt_group] テーブルのデータが次の情報のように表示されることを確認します。

group_name character var	group_desc character var	dynamic_gro boolean	dynamic_rule character var	nested_group boolean	idmrpt_valid_from timestamp without tin	idmrpt_deleted boolean	idmrpt_syn_state smallint
Pharmacy	Pharmacy	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1
IT	Information Tec	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1
HR	Human Resourc	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1
Physician	Physician	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1
Operations	Operations	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1
Medical Operatic	Medical Operatic	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1
Nursing	Nursing	FALSE		FALSE	2010-07-07 21:28:11	FALSE	1

このテーブルには、各グループの名前のほかに、そのグループが動的であるか、それともネストされているかを示すフラグが表示されます。さらに、グループが移行済みかどうかも表示されます。オブジェクトがユーザアプリケーションで変更されているものの、まだ移行されていない場合、同期ステータス (idmrpt_syn_state) が 0 に設定されている可能性があります。たとえば、ユーザをグループに追加し、ドライバがまだ移行されていない場合、この値が 0 に設定されている可能性があります。

9 (オプション) 次のテーブルのデータを確認します。

- idmrpt_approver
- idmrpt_association
- idmrpt_category
- idmrpt_container
- idmrpt_idv_drivers
- idmrpt_idv_prd
- idmrpt_role
- idmrpt_resource
- idmrpt_sod

10 (オプション) 管理対象システムのゲートウェイドライバのデータ収集状態に関する情報が表示される **[idmrpt_ms_collect_state]** テーブルに新しい行が含まれていることを確認します。

このテーブルには、管理対象システムのどの REST エンドポイントが実行されたかに関するデータが記録されます。この時点では、まだこのドライバのデータ収集プロセスを開始していないため、テーブルに行はありません。

カスタム属性とカスタムオブジェクトのサポートの追加

デフォルトのデータ収集スキームに含まれないカスタム属性とカスタムオブジェクトのデータを収集して永続化するよう、データ収集サービスドライバを設定できます。このためには、データ収集サービスドライバフィルタを変更する必要があります。フィルタを変更しても、オブジェクトの同期はすぐにはトリガされません。その代わりに、新しく追加した属性とオブジェクトは、識別ボールトで追加、変更、または削除イベントが発生したときにデータ収集サービスに送信されます。

カスタム属性とカスタムオブジェクトのサポートを追加する場合、レポートを変更して拡張属性と拡張オブジェクトの情報を組み込む必要があります。拡張オブジェクトと拡張属性に関する最新データと履歴データは、次のビューで提供されます。

- idm_rpt_cfg.idmrpt_ext_idv_item_v
- idm_rpt_cfg.idmrpt_ext_item_attr_v

このプロセスには次の作業が含まれます。

- 152 ページの 「拡張オブジェクトを使用するためのドライバの設定」
- 152 ページの 「データベースへの名前と説明の組み込み」
- 153 ページの 「既知のオブジェクトタイプへの拡張属性の追加」

拡張オブジェクトを使用するためのドライバの設定

任意のオブジェクトまたは属性をデータ収集サービスフィルタポリシーに追加できます。新しいオブジェクトまたは属性を追加する場合、GUID (購読者同期を使用) およびオブジェクトクラス (購読者通知を使用) をマップする必要があります。次に例を示します。

```
<filter-class class-name="Device" publisher="ignore" publisher-create-
homedir="true" publisher-track-template-member="false" subscriber="sync">
<filter-attr attr-name="CN" merge-authority="default" publisher="ignore"
publisher-optimize-modify="true" subscriber="sync"/>
<filter-attr attr-name="Description" merge-authority="default"
publisher="ignore" publisher-optimize-modify="true" subscriber="sync"/>
<filter-attr attr-name="GUID" merge-authority="default" publisher="ignore"
publisher-optimize-modify="true" subscriber="sync"/>
<filter-attr attr-name="Object Class" merge-authority="default"
publisher="ignore" publisher-optimize-modify="true" subscriber="notify"/>
<filter-attr attr-name="Owner" merge-authority="default"
publisher="ignore" publisher-optimize-modify="true" subscriber="sync"/>
<filter-attr attr-name="Serial Number" merge-authority="default"
publisher="ignore" publisher-optimize-modify="true" subscriber="sync"/>
<filter-attr attr-name="sampleDeviceModel" from-all-classes="true" merge-
authority="default" publisher="ignore" publisher-optimize-modify="true"
subscriber="sync"/>
<filter-attr attr-name="sampleDeviceType" from-all-classes="true" merge-
authority="default" publisher="ignore" publisher-optimize-modify="true"
subscriber="sync"/>
</filter-class>
```

データベースへの名前と説明の組み込み

データベースにオブジェクトの名前と設定を指定する場合、_dcsName および _dcsDescription に対するスキーママッピングポリシーを追加する必要があります。このスキーママッピングポリシーは、オブジェクトインスタンスの属性値をそれぞれ列 idmrpt_ext_idv_item.item_name および idmrpt_ext_idv_item.item_desc にマップします。スキーママッピングポリシーを追加しない場合、属性は子テーブル idmrpt_ext_item_attr で設定されます。

次に例を示します。

```
<attr-name class-name="Device">
<nds-name>CN</nds-name>
<app-name>_dcsName</app-name>
</attr-name>
<attr-name class-name="Device">
<nds-name>Description</nds-name>
<app-name>_dcsDescription</app-name>
</attr-name>
```

次の SQL の例では、データベース内にあるこれらのオブジェクトと属性の値を表示できます。

```
SELECT
    item.item_dn,
    item.item_name,
    item.item_desc,
    attr.attribute_name,
    itemAttr.attribute_value,
    item.idmrpt_deleted as item_deleted,
    itemAttr.idmrpt_deleted as attr_deleted,
    item.item_desc,
    obj.object_class
FROM
    idm_rpt_data.idmrpt_ext_idv_item as item,
idm_rpt_data.idmrpt_ext_item_attr itemAttr, idm_rpt_data.idmrpt_ext_attr
as attr, idm_rpt_data.idmrpt_ext_obj as obj
WHERE
    item.object_id = obj.object_id and itemAttr.attribute_id =
attr.attribute_id and itemAttr.cat_item_id = item.item_id
ORDER BY
    item.item_dn, item.item_name
```

既知のオブジェクトタイプへの拡張属性の追加

属性をデータ収集サービスドライバのフィルタポリシーに追加し、XML 参照ファイル (`IdmrptIdentity.xml`) でレポーティングデータベースに明示的にマップしていない場合、値には idmrpt_ext_attr テーブルの属性参照が取り込まれ、idmrpt_ext_item_attr テーブルで管理されます。

次の SQL の例は、これらの拡張属性を示しています。

```
SELECT
    acct.idv_acct_dn,
    attrDef.attribute_name,
    attribute_value,
    attrVal.idmrpt_valid_from,
    cat_item_attr_id,
    attrVal.idmrpt_deleted,
    attrVal.idmrpt_syn_state
FROM
    idm_rpt_data.idmrpt_ext_item_attr as attrVal,
idm_rpt_data.idmrpt_ext_attr as attrDef, idm_rpt_data.idmrpt_identity as
idd, idm_rpt_data.idmrpt_idv_acct as acct
WHERE attrVal.attribute_id = attrDef.attribute_id and idd.identity_id =
acct.identity_id and attrVal.cat_item_id = acct.identity_id and
cat_item_type_id = 'IDENTITY'
```

ユーザオブジェクトのほかに、次のオブジェクトのフィルタポリシーにも拡張属性を追加し、データベースにそれらの属性を入力できます。

- nrfRole
- nrfResource
- コンテナ

  注 : インストールした製品は organizationUnit、Organization、および Domain をサポートしています。コンテナタイプは idmrpt_container_types テーブルで管理されます。

- グループ
- nrfSod

拡張属性と親テーブルまたはオブジェクトの関連付けは、idmrpt_cat_item_types.idmrpt_table_name 列を参照することで確認できます。この列には、idm_rpt_data.idmrpt_ext_item_attr.cat_item_id 列を親テーブルのプライマリキーに結合する方法が記述されています。

複数のドライバセットのサポートの追加

Data Collection Service Scoping パッケージ (NOVLDCSSCPNG) は、複数のドライバセットと、データ収集サービスドライバおよび管理対象システムのゲートウェイドライバのペアを複数使用するエンタープライズ環境において、静的および動的なスコープ設定機能を実現します。

インストール中またはインストール後に、このパッケージをインストールするデータ収集サービスドライバの役割を決定する必要があります。次のいずれかの役割を選択する必要があります。

- **プライマリ** このドライバは、他のドライバセットのサブツリーを除くすべてを同期します。プライマリデータ収集サービスドライバは、識別ボールト全体にサービスを提供したり、1 つ以上のセカンダリドライバと連携して動作したりする可能性があります。

- **セカンダリ** このドライバは専用のドライバセットのみを同期し、それ以外は何も同期しません。通常、セカンダリデータ収集サービスドライバには、別のドライバセットで実行されているプライマリドライバが必要です。そうでない場合、ローカルドライバセットの外部にあるデータはデータ収集サービスに送信されません。
- **Custom** 管理者はカスタムスコープ設定ルールを定義できます。暗黙のスコープはローカルドライバセットのみで、それ以外は、カスタムスコープのリストに明示的に追加されていない限り、すべてスコープの範囲外とみなされます。カスタムスコープは、同期するサブオーディネートまたはサブツリーが含まれる識別ボールト内にあるコンテナのスラッシュ形式の識別名です。

このスコープ設定パッケージが必要になるのは、次に挙げるような特定の設定シナリオのみです。

- **1 つのサーバと、1 つのドライバセットの識別ボールト。**: このシナリオでは、スコープを設定する必要がないため、スコープ設定パッケージをインストールする必要はありません。
- **複数のサーバと、1 つのドライバセットの識別ボールト。**: このシナリオでは、次のガイドラインに従う必要があります。
  - Identity Manager サーバがデータ収集元の全パーティションのレプリカを保持する必要があります。
  - このシナリオでは、スコープ設定は必要ないため、スコープ設定パッケージはインストールしません。
- **複数のサーバと、複数のドライバセットの識別ボールト。**: このシナリオでは、次の 2 つの基本設定があります。
  - すべてのサーバがデータ収集元の全パーティションのレプリカを保持する設定。

    この設定では、次のガイドラインに従う必要があります。
    - 複数の DCS ドライバによって同じ変更が処理されるのを避けるため、スコープ設定が必要です。
    - すべてのDCSドライバにスコープ設定パッケージをインストールする必要があります。
    - 1 つの DCS ドライバをプライマリドライバとして選択する必要があります。
    - 他のDCSドライバはすべてセカンダリドライバになるように設定する必要があります。
  - すべてのサーバがデータ収集元の全パーティションのレプリカを「保持しない」設定。

    この設定では、次の 2 つの状況が考えられます。
    - データ収集元の全パーティションは「1 つの」 Identity Manager サーバによってのみ保持されます。

      この場合、次のガイドラインに従う必要があります。
      - 複数の DCS ドライバによって同じ変更が処理されるのを避けるため、スコープ設定が必要です。

- すべてのDCSドライバにスコープ設定パッケージをインストールする必要があります。
- すべてのDCSドライバをプライマリドライバになるように設定する必要があります。

- データの収集元の全パーティションは「1つのIdentity Managerサーバのみによって保持されません」(一部のパーティションは複数のIdentity Managerサーバで保持されます)。

  この場合、次のガイドラインに従う必要があります。

  - 複数のDCSドライバによって同じ変更が処理されるのを避けるため、スコープ設定が必要です。
  - すべてのDCSドライバにスコープ設定パッケージをインストールする必要があります。
  - すべてのDCSドライバをプライマリドライバになるように設定する必要があります。

    各ドライバに対してカスタムスコープ設定ルールを定義し、作成したスコープが重複していないことを確認する必要があります。

SSLを使用したリモートモードでのドライバ実行設定

リモートモードで実行する場合、データ収集サービスドライバおよび管理対象のシステムゲートウェイドライバを、SSLを使用するように設定できます。このセクションでは、SSLを使用してリモートモードで実行されるようにドライバを設定する手順について説明します。

管理対象システムのゲートウェイドライバに対してキーストアを使用してSSLを設定する

1 Identity Consoleでサーバ証明書を作成します。

  1a [eDirectory]で、[Certificate Management(証明書管理)]>[Server Certificate Management(サーバ証明書管理)]をクリックします。

  1b [+]アイコンをクリックすると、[Create Server Certificate(サーバ証明書の作成)]タブが表示されます。

  1c 管理対象システムのゲートウェイドライバがインストールされているサーバオブジェクトを参照して選択します。

  1d 証明書のニックネームを指定します。

  1e 作成方法として[Standard(標準)]を選択し、[次へ]をクリックします。

  1f [終了]をクリックし、[閉じる]をクリックします。

2 Identity Consoleを使用してサーバ証明書をエクスポートします。

  2a [eDirectory]で、[Certificate Management(証明書管理)]>[Server Certificate Management(サーバ証明書管理)]をクリックします。

  2b 156ページのステップ1で作成した証明書を選択して、[エクスポート]をクリックします。

  2c [証明書]メニューで、証明書の名前を選択します。

2d [Export private key (秘密鍵のエクスポート)] がオンになっていることを確認します。

2e パスワードを入力し、[次へ] をクリックします。

2f [Save the exported certificate (エクスポートされた証明書の保存)] をクリックし、エクスポートされた pfx 証明書を保存します。

3 156 ページのステップ 2 でエクスポートした pfx 証明書を Java キーストアにインポートします。

3a Java に付属するキーツールを使用します。JDK 6 以上を使用する必要があります。

3b コマンドプロンプトで次のコマンドを入力します。

```
keytool -importkeystore -srckeystore pfx certificate -srcstoretype
PKCS12 -destkeystore Keystore Name
```

次に例を示します。

```
keytool -importkeystore -srckeystore cert.pfx -srcstoretype PKCS12
-destkeystore msgw.jks
```

3c 要求されたときにはパスワードを入力してください。

4 Identity Console を使用して、このキーストアを使用するように Managed System Gateway Driver の設定を変更します。

4a [IDM Adminstration (IDM 管理)] から、Managed System Gateway Driver が含まれるドライバセットをクリックします。

4b [Managed System Gateway Driver] アイコンをクリックして、[環境設定] に移動します。

4c [ドライバパラメータ] ドロップダウンをクリックします。

4d [Driver Settings (ドライバ設定)] タブで、[Connection Parameters (接続パラメータ)] を［show (表示)］に設定し、[Driver configuration mode (ドライバ環境設定モード)] を［remote (リモート)］に設定します。

4e キーストアファイルの完全なパスとパスワードを入力します。

4f 保存してドライバを再起動します。

5 Identity Console を使用して、このキーストアを使用するようにデータ収集サービスドライバの設定を変更します。

5a [IDM Adminstration (IDM 管理)] から、データ収集サービスドライバが含まれるドライバセットをクリックします。

5b [Data Collection Service Driver (データ収集サービスドライバ)] アイコンをクリックして、[ドライバ環境設定] に移動します。

5c [ドライバパラメータ] ドロップダウンをクリックして、[Registration Parameters (登録パラメータ)] を［show (表示)］に設定します。

5d [Managed System Gateway Registration (管理対象システムのゲートウェイの登録)] という見出しの下にある [Managed System Gateway Driver Configuration Mode (管理対象システムのゲートウェイドライバの環境設定モード)] を［remote (リモート)］に設定します。

5e キーストアの完全なパス、パスワード、および 156 ページのステップ 1d で入力した別名を入力します。

5f 保存してドライバを再起動します。

Identity Reporting の設定

Identity Reporting のインストール後でも、さまざまなインストールプロパティを変更できます。変更するには、設定更新ユーティリティ (`configupdate.sh`) ファイルを実行します。

環境設定ツールで Identity Reporting の設定を変更した場合、変更を反映するには Tomcat を再起動する必要があります。ただし、Identity Reporting の Web ユーザインタフェースで変更を加えた場合は、サーバの再起動は必要ありません。

- 158 ページの 「管理対象システムのゲートウェイドライバの設定」
- 159 ページの 「[Identity Data Collection Services (アイデンティティデータ収集サービス)] ページへのデータソースの手動追加」
- 159 ページの 「Oracle データベースでのレポートの実行」
- 159 ページの 「データベーススキーマの手動生成」
- 165 ページの 「Identity Reporting 用の REST API の展開」
- 165 ページの 「リモート PostgreSQL データベースへの接続」
- 166 ページの「分散型セットアップでの Identity Reporting 用 OSP へのクライアントの登録」
- 166 ページの 「Identity Reporting のインストール後の手順」

管理対象システムのゲートウェイドライバの設定

インストーラは、Identity Reporting 用のデータ収集サービス用ドライバおよび Managed System Gateway Driver を作成して設定します。

Managed System Gateway Driver が設定されたら、KMO を **[SSL EC Certificate DNS (SSL EC 証明書 DNS)]** に変更することをお勧めします。KMO を変更するには、次の手順を実行します。

1 Identity Console にログインします。

2 **[IDM Administration (IDM 管理)]** から、Managed System Gateway Driver が含まれるドライバセットをクリックします。

3 **[Managed System Gateway Driver]** アイコンをクリックして、**[ドライバ環境設定]** に移動します。

4 [環境設定] タブで、[ドライバパラメータ] ドロップダウンをクリックして、次の手順を実行します。

4a [Show connection parameters (接続パラメータの表示)] フィールドから show （表示）を選択します。

4b [KMO Name (KMO 名)] フィールドで、値を [SSL CertificateDNS (SSL 証明書 DNS)] から [SSL EC Certificate DNS (SSL EC 証明書 DNS)] に変更します。

4c [適用] をクリックし、[OK] をクリックします。

[Identity Data Collection Services (アイデンティティデータ収集サービス)] ページへのデータソースの手動追加

1. Identity Reporting アプリケーションにログインします。
2. [データソース] をクリックします。
3. [追加] をクリックします。
4. [データソースの追加] ダイアログボックスで、[事前定義リストから選択します] ラジオボタンをクリックします。
5. [IDMDCSDataSource] を選択します。
6. [保存] をクリックします。

Oracle データベースでのレポートの実行

Identity Reporting は、リモートの Oracle データベースに対してレポートを実行できます。Oracle データベースを実行しているサーバに ojbc8.jar ファイルがあることを確認します。

データベーススキーマの手動生成

インストール後にデータベーススキーマを手動で生成するには、データベースについて次のいずれかの手順を実行します。

- 160 ページの 「PostgreSQL データベースに対する Create_rpt_roles_and_schemas.sql スキーマの設定」
- 160 ページの 「Oracle データベースに対する Create_rpt_roles_and_schemas.sql スキーマの設定」
- 163 ページの「MS SQL データベースに対する Create_rpt_roles_and_schemas.sql スキーマの設定」
- 163 ページの 「データベースチェックサムのクリア」
- 164 ページの 「(オプション) カラムのデータサイズを増やす」

PostgreSQL データベースに対する Create_rpt_roles_and_schemas.sql スキーマの設定

1 /opt/netiq/idm/apps/IDMReporting/sql/ にある SQL、create_dcs_roles_and_schemas.sql および create_rpt_roles_and_schemas.sql を使用して、データベースに必要な役割を追加します。

2 postgres ユーザとして PGAdmin にログインします。

3 クエリツールを実行します。

4 Create_rpt_roles_and_schemas および Create_dcs_roles_and_schemas プロシージャを作成するには、これらの SQL からクエリツールにコンテンツをコピーして、接続されているデータベースに対して実行します。

5 IDM_RPT_DATA、IDM_RPT_CFG、および IDMRPTUSER 役割を作成するには、次のコマンドを指定された順序で実行します。

```
Select CREATE_DCS_ROLES_AND_SCHEMAS('<Set pwd for IDM_RPT_DATA>', '<Set pwd for IDMRPTUSER>');

Select CREATE_RPT_ROLES_AND_SCHEMAS('<Set pwd for IDM_RPT_CFG>');
```

たとえば、IDM_RPT_DATA、IDMRPTUSER、および IDM_RPT_CFG のパスワードが、それぞれ *password*、*password1*、および *password2* の場合、以下のコマンドを実行する必要があります。

```
Select CREATE_DCS_ROLES_AND_SCHEMAS('password', 'password1');

Select CREATE_RPT_ROLES_AND_SCHEMAS('password2');
```

6 get_formatted_user_dn.sql のコンテンツを /opt/netiq/idm/apps/IDMReporting/sql/ からクエリツールにコピーし、接続されたデータベースに対して実行します。

注 : データベーススキーマ作成オプションとして **[File]** を選択した場合、get_formatted_user_dn.sql 関数を手動で追加する必要があります。データベーススキーマ作成オプションとして **[Now]** または **[Startup]** を選択した場合、インストーラはこの関数をデータベースに追加します。

Oracle データベースに対する Create_rpt_roles_and_schemas.sql スキーマの設定

1 データベース管理者 (sysdba) ユーザとして SQL Developer にログインします。

2 以下の SQL の内容を SQL Developer にコピーし、接続されたデータベースに対して実行することで、Create_rpt_roles_and_schemas を作成します。

```
alter session set "_ORACLE_SCRIPT"=true;

CREATE OR REPLACE PROCEDURE create_rpt_roles_and_schemas(
   idm_rpt_cfg_password character varying)
AUTHID CURRENT_USER
AS
   cnt number;
BEGIN
   /* Create user IDM_RPT_CFG if it does not exist */
   select count(*) into cnt from ALL_USERS WHERE USERNAME =
'IDM_RPT_CFG';
   IF cnt = 0 THEN
       execute immediate 'CREATE USER idm_rpt_cfg IDENTIFIED BY ' ||
idm_rpt_cfg_password;
       DBMS_OUTPUT.put_line('Created user idm_rpt_cfg');
   END IF;

   /* Grant rights to the idm_rpt_cfg user */
   execute immediate 'GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW,
CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER, UNLIMITED TABLESPACE
to idm_rpt_cfg';
   DBMS_OUTPUT.put_line('Granted rights to user idm_rpt_cfg');
   execute immediate 'alter session set "_ORACLE_SCRIPT"=true';
   execute immediate 'alter session set "_ORACLE_SCRIPT"=false';
END;
/
exec CREATE_RPT_ROLES_AND_SCHEMAS('Novell123');
/
alter session set "_ORACLE_SCRIPT"=false;
/
```

3 System ユーザとして SQL Developer にログインし、以下の SQL の内容を SQL Developer にコピーし、接続されたデータベースに対して実行することで、`Create_dcs_roles` と `schemas` を作成します。

```
alter session set "_ORACLE_SCRIPT"=true;

CREATE OR REPLACE PROCEDURE create_dcs_roles_and_schemas(
   idm_rpt_data_password character varying,
   idmrptuser_password character varying)
AUTHID CURRENT_USER
AS
   cnt number;
BEGIN
/* Create user IDM_RPT_DATA if it does not exist already */
   select count(*) into cnt from ALL_USERS WHERE USERNAME =
'IDM_RPT_DATA';
   IF cnt = 0 THEN
       execute immediate 'CREATE USER idm_rpt_data IDENTIFIED BY ' ||
idm_rpt_data_password;
       DBMS_OUTPUT.put_line('Created user idm_rpt_data');
   END IF;

   /* Grant rights to the idm_rpt_data user */
   execute immediate 'GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW,
```

```
CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER, UNLIMITED TABLESPACE
to idm_rpt_data';
   execute immediate 'GRANT CREATE PUBLIC SYNONYM to IDM_RPT_DATA';
   execute immediate 'GRANT SELECT ON DBA_SEGMENTS TO IDM_RPT_DATA';
   DBMS_OUTPUT.put_line('Granted rights to user idm_rpt_data');
   /* Create user IDMRPTUSER if it does not exist */
   select count(*) into cnt from ALL_USERS WHERE USERNAME =
'IDMRPTUSER';
   IF cnt = 0 THEN
       execute immediate 'CREATE USER idmrptuser IDENTIFIED BY ' ||
idmrptuser_password;
       DBMS_OUTPUT.put_line('Created user idmrptuser');
   END IF;

   /* Grant rights to the idmrptuser user */
   execute immediate 'GRANT CREATE SESSION to idmrptuser';
   execute immediate 'GRANT SELECT ON sys.all_tables TO idmrptuser';
   DBMS_OUTPUT.put_line('Granted rights to user idmrptuser');
END;
/
```

4 IDM_RPT_CFG 役割を作成するには、次のコマンドを実行します。

```
begin
CREATE_RPT_ROLES_AND_SCHEMAS('<Set pwd for IDM_RPT_CFG>');
end;
```

たとえば、IDM_RPT_CFG のパスワードが password である場合は、次のコマンドを実行する必要があります。

```
begin
CREATE_RPT_ROLES_AND_SCHEMAS('password');
end;
```

5 次の許可を割り当てます。

```
GRANT CREATE PUBLIC SYNONYM to IDM_RPT_CFG;
```

6 get_formatted_user_dn-oracle.sql のコンテンツを /opt/netiq/idm/apps/IDMReporting/sql/ から SQL Developer にコピーし、接続されたデータベースに対して実行します。

注 : データベーススキーマ作成オプションとして File を選択した場合、[get_formatted_user_dn-oracle.sql] 関数を手動でデータベースに追加する必要があります。データベーススキーマ作成オプションとして [Now] または [Startup] を選択した場合、インストーラはこの関数をデータベースに追加します。

MS SQL データベースに対する Create_rpt_roles_and_schemas.sql スキーマの設定

1 delete_create_dcs_roles_and_schemas-mssql.sql および delete_get_formatted_user_dn-mssql.sql を実行します。

2 /opt/netiq/idm/apps/IDMReporting/sql/ から create_dcs_roles_and_schemas.mssql および create_rpt_roles_and_schemas.mssql を使用して、データベースに必要な役割を追加します。

3 データベース管理者ユーザとして SQL Developer にログインします。

4 Create_rpt_roles_and_schemas および Create_dcs_roles_and_schemas プロシージャを作成するには、コンテンツを create_dcs_roles_and_schemas.mssql および create_rpt_roles_and_schemas.mssql から SQL Developer にコピーし、接続されたデータベースに対して実行します。

5 IDM_RPT_DATA、IDM_RPT_CFG、および IDMRPTUSER 役割を作成するには、次のコマンドを指定された順序で実行します。

```
execute CREATE_DCS_ROLES_AND_SCHEMAS '<Set pwd for IDM_RPT_DATA>',
'<Set pwd for IDMRPTUSER>'

execute CREATE_RPT_ROLES_AND_SCHEMAS \'<Set pwd for IDM_RPT_CFG>\'
```

6 get_formatted_user_dn.sql のコンテンツを /opt/netiq/idm/apps/IDMReporting/sql/ から SQL Developer にコピーし、接続されたデータベースに対して実行します。

データベースチェックサムのクリア

1 /opt/netiq/idm/apps/IDMReporting/sql で次の .sql ファイルを見つけます。

- DbUpdate-001-run-as-idm_rpt_data.sql
- DbUpdate-01-run-as-idm_rpt_cfg.sql
- DbUpdate-02-run-as-idm_rpt_cfg.sql
- DbUpdate-03-run-as-idm_rpt_data.sql
- DbUpdate-04-run-as-idm_rpt_data.sql
- DbUpdate-05-run-as-idm_rpt_data.sql
- DbUpdate-06-run-as-idm_rpt_cfg.sql

2 データベースチェックサムをクリアします。

2a 各 .sql で clearchsum コマンドを実行するには、各ファイルの初めに次の行を追加します。

```
update DATABASECHANGELOG set MD5SUM = NULL
go
```

変更されたコンテンツは次のようになるはずです。

```
--
**********************************************************************
**
-- Update Database Script
--
**********************************************************************
**
-- Change Log: IdmDcsDataDropViews.xml
-- Ran at: 2/23/18 5:17 PM
-- Against: IDM_RPT_CFG@jdbc:oracle:thin:@192.168.0.1:1521/orcl
-- Liquibase version: 3.5.1
--
**********************************************************************
**
update databasechangelog set md5sum = null
go
```

2b 対応するユーザで各 `.sql` を実行します。

3 データベースに変更をコミットします。

(オプション) カラムのデータサイズを増やす

以前のバージョンの Identity Manager では、文字制限により、長いデータフィールドで Identity Reporting サーバとデータを同期できませんでした。Identity Manager 4.9 では、次のテーブルおよびその各カラムに対して、PostgreSQL、Oracle、および MS SQL データベースでの文字制限を増やすオプションが提供されています。

テーブルの名前	カラムの名前
idm_rpt_data.idmrpt_idv_ent_bindings	ent_param_str
idm_rpt_data.idmrpt_idv_ent_bindings	ent_param_val
idm_rpt_data.idmrpt_idv_identity_trust	idv_ent_ref
idm_rpt_data.idmrpt_idv_identity_trust	trust_params
idm_rpt_data.idmrpt_idv_ent_bindings_hist	ent_param_str
idm_rpt_data.idmrpt_idv_ent_bindings_hist	ent_param_val
idm_rpt_data.idmrpt_idv_identity_trust_hist	idv_ent_ref
idm_rpt_data.idmrpt_idv_identity_trust_hist	trust_params

- PostgreSQL の場合、Identity Manager 4.9 では、上記のテーブルで示すすべてのフィールドに対して文字制限が自動的に増やされています。

- Oracle の場合、/opt/netiq/idm/apps/IDMReporting/sql/ ディレクトリから alter_column_length-oracle.sql スクリプトを実行して、上記のテーブルで示すすべてのカラムに対して文字制限を増やす必要があります。
- MS SQL の場合、/opt/netiq/idm/apps/IDMReporting/sql/ ディレクトリから alter_column_length-mssql.sql スクリプトを実行して、インデックス付きカラムのみに文字制限を増やす必要があります。この場合、ENT_PARAM_STR は、テーブル idmrpt_idv_ent_bindings の下の唯一のインデックス付きカラムです。

Identity Reporting 用の REST API の展開

Identity Reporting には、レポーティング機能内でさまざまな機能を可能にする複数の REST API が組み込まれています。これらの REST API は認証に OAuth2 プロトコルを使用します。

Tomcat では、Identity Reporting がインストールされるときに、rptdoc war および dcsdoc war が自動的に展開されます。

リモート PostgreSQL データベースへの接続

PostgreSQL データベースが別のサーバにインストールされている場合は、リモートデータベースの postgresql.conf および pg_hba.conf ファイルのデフォルト設定を変更する必要があります。

1 postgresql.conf ファイルのリスニングアドレスを変更します。

デフォルトでは、PostgreSQL は localhost 接続をリスンできます。リモート TCP/IP 接続は許可されません。リモート TCP/IP 接続を許可するには、/opt/netiq/idm/postgres/data/postgresql.conf ファイルに次のエントリを追加します。

```
listen_addresses = '*'
```

サーバ上に複数のインタフェースがある場合は、リスンされる特定のインタフェースを指定できます。

2 pg_hba.conf ファイルにクライアント認証エントリを追加します。

デフォルトで、PostgreSQL は localhost からの接続のみを受諾します。リモート接続は拒否します。これは、有効なパスワード (md5 キーワード) を入力したユーザには IP アドレスからのログインを許可する、アクセス制御ルールを適用することによって制御されます。リモート接続を受け入れるには、/opt/netiq/idm/postgres/data/pg_hba.conf ファイルに次のエントリを追加します。

```
host all all 0.0.0.0/0 md5
```

たとえば、192.168.104.24/26 trust です。

これは IPv4 アドレスに対してのみ機能します。IPv6 アドレスの場合、次のエントリを追加します。

```
host all all ::0/0 md5
```

特定のネットワーク上にある複数のクライアントコンピュータからの接続を許可する場合は、このエントリに CIDR アドレス形式でネットワークアドレスを指定します。

pg_hba.conf ファイルは、次のクライアント認証形式をサポートしています。

- ローカルデータベースユーザ認証方法 [認証オプション]
- ホストデータベースユーザ CIDR アドレス認証方法 [認証オプション]
- hostssl データベースユーザ CIDR アドレス認証方法 [認証オプション]
- hostnossl データベースユーザ CIDR アドレス認証方法 [認証オプション]

CIDR アドレス形式の代わりに、次の形式を使用して別のフィールドに IP アドレスとネットワークマスクを指定できます。

- ホストデータベースユーザ IP アドレス IP マスク認証方法 [認証オプション]
- hostssl データベースユーザ IP アドレス IP マスク認証方法 [認証オプション]
- hostnossl データベースユーザ IP アドレス IP マスク認証方法 [認証オプション]

3 リモート接続をテストします。

3a リモート PostgreSQL サーバを再起動します。

3b ユーザ名とパスワードを使用してリモートでサーバにログインします。

分散型セットアップでの Identity Reporting 用 OSP へのクライアントの登録

1 OSP をインストールした場所に移動し、`configupdate.sh.properties` ファイルの次のエントリを編集します。

```
app_versions="ua#<version>,rpt#<version>"
sso_apps=ua,rpt
```

例 :

```
app_versions="ua#4.9.0,rpt#7.2.0"
sso_apps=ua,rpt
```

2 configupdate を起動し、次のフィールドで Identity Reporting のクライアントの詳細を更新します。

- Reporting client (Reporting クライアント)
- データ収集サービス
- DCS Driver (DCS ドライバ)

3 OSP の Tomcat を再起動します。

Identity Reporting のインストール後の手順

インストール時に、[Create Schema for Identity Reporting (Identity Reporting 用のスキーマの作成)] で **[Startup]** または **[File]** を選択した場合、次の手順を実行してください。

1. Identity Reporting にログインします。

2. Identity Reporting リポジトリから既存のデータソースとレポート定義を削除します。
3. 新しい Identity Manager データ収集サービスデータソースを追加します。

また、利用可能なデータベースが正しく実行されるように、Tomcat を一度起動した後に再起動してください。

root 以外のインストールの完了

root 以外のユーザとして Identity Manager エンジンおよびプラグインをインストールする場合、プロセスによりすべての意図されているインストールアクティビティが実行されます。このセクションでは、インストールを完了するために必要な手動のプロセスをガイドします。

パスワードポリシーのコンテナの作成

Identity Manager には、アイデンティティボールトのパスワードポリシーオブジェクトが必要です。ただし、root 以外のインストールプロセスでは、パスワードポリシーのコンテナは作成されません。

1 Identity Console にログインし、[ツリービュー] に移動します。
2 アイデンティティボールトのセキュリティコンテナに移動します。
3 パスワードポリシーのコンテナを作成します。

電子メール通知のグラフィックサポートの追加

アイデンティティボールトと Identity Manager エンジンを非 root ユーザとしてインストールする場合は、電子メール通知に電子メールテンプレートで提供されるグラフィックやイメージが含まれない可能性があります。たとえば、do-send-email-from-template アクションを実行する場合、Identity Manager は電子メールを送信しますが、含まれているイメージは空です。グラフィックを確実にサポートするにためには、driverset を更新する必要があります。

1 Designer でプロジェクトにログインします。
2 ［アウトライン］ペインで、[識別ボールト] を展開します。
3 [ドライバセット] を右クリックします。
4 [プロパティ ］ > ［Java] を選択します。
5 JVM オプションで、次のコンテンツを入力します。

```
-Dcom.novell.nds.dirxml.util.mail.templatepath=path_to_graphics_files
```

次に例を示します。

```
-Dcom.novell.nds.dirxml.util.mail.templatepath=/prod/eDirectory/opt/
novell/eDirectory/lib/dirxml/rules/manualtask/mt_files
```

6 [OK] をクリックします。

7 ドライバセットに変更を展開します。

7a [ドライバセット] を右クリックします。

7b [ライブ] > [展開] の順に選択します。

7c [展開] を選択します。

8 アイデンティティボールトを再起動します。

Identity Manager のアクティベート

Identity Manager をインストールする場合や、Identity Manager を初めて実行する場合、アクティベーションコードは必要ありません。ただし、アクティベーションコードがない場合、インストールから 90 日を経過すると Identity Manager は機能しなくなります。この 90 日の期間中またはその後いつでも Identity Manager をアクティベートできます。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Activating Identity Manager」を参照してください。

分散展開用の SSL 証明書の作成

次の手順では、分散展開用の SSL 証明書を作成する方法について説明します。

1 次のコマンドを実行して、Java パスを設定します。

```
export PATH=/opt/netiq/common/jre/bin:$PATH
```

サーバに Java パスが設定されていることを確認します。例 :

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

2 サンプルの作業ディレクトリを作成します。

```
mkdir -p /opt/certs
```

3 PKCS キーストアを生成します。

```
keytool -genkey -alias < alias_name > -keyalg RSA -keystore <absolute
path of keystore > -validity 3650 -keysize 1024 -dname "CN=<Subject
Alternate Name>" -keypass <keypass> -storepass <storepass>
```

4 CSR (証明書署名要求) を生成します。

```
keytool -certreq -v -alias < alias_name >  -file < name of the csr file
> -keypass < keypass >  -keystore < absolute path of the keystore >  -
storepass < storepass >
```

5 自己署名証明書を生成します。

5a Identity Console を起動してログインします。

5b [Certificate Management (証明書管理)] > [Issue Certificate (証明書の発行)] の順に移動します。

5c 手順 3 で作成した `.csr` ファイルを参照します。たとえば、`ua.csr` というファイルを参照します。

5d [開く] をクリックします。

5e [Key Usage Specifications (キーの使用法の指定)] ドロップダウンで、キータイプを指定します。

5f [Extended Key Usage Specifications (拡張キーの使用法の指定)] ドロップダウンで、拡張キータイプを指定します。

5g 証明書タイプには、[Basic Constrains (基本制約)] ドロップダウンで [Unspecified (未指定)] を選択します。

5h [Certificate Parameters (証明書パラメータ)] ドロップダウンで証明書の有効期間を指定します。

5i [Certificate format (証明書形式)] ドロップダウンで、[File in binary DER format (バイナリ DER 形式のファイル)] を選択します。

5j [次へ] をクリックします。

5k [OK] をクリックします。

5l 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

6 .der 形式で root 証明書をエクスポートします。

6a Identity Console を起動してログインします。

6b [Certificate Management (証明書管理)] > [Server Certificate Management (サーバ証明書管理)] に移動します。

6c [SSL CertificateDNS (SSL 証明書 DNS)] チェックボックスをオンにして、[エクスポート] をクリックします。

6d [証明書] ドロップダウンリストで、組織の CA を選択します。

6e [エクスポート形式] ドロップダウンリストで、DER を選択します。

6f [OK] をクリックします。

6g 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

7 先に説明したように証明書を PKCS キーストアにインポートします。

```
keytool -import -trustcacerts -alias root -keystore <name of the
keystore> -file <root certificate > -storepass <password> -noprompt
```

例 :

```
keytool -import -alias <alias name> -keystore <name of the keystore> -
file <self-signed certificate> -storepass <password> -noprompt
```

8 osp、userapp、および reporting が異なるマシンで使用されている場合は、それぞれについて先に述べた手順を繰り返します。

IV FIPS モードでの Identity Manager の設定

Identity Manager 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。FIPS 140-2 標準は、暗号ソフトウェアの実装と通信を規定します。Bouncy Castle は、FIPS 140-2 レベル 1 の認定を受けたオープンソースの JCA プロバイダを使用しています。このプロバイダは、定評ある Bouncy Castle オープンソース暗号プロバイダに基づいて構築されています。

注 :

- SSPR は FIPS モードをサポートしていません。非 FIPS の SSPR 設定は別のサーバに移行する必要があります。詳細については、『*Self Service Password Reset 4.7 Administration Guide*』の「Backing Up Configuration Information」と「Importing Configuration Information」を参照してください。
- マスタキーストアのパスワードが tomcat キーストアのパスワードと同じであることを確認してください。パスワードを変更するには、113 ページの 「Configupdate ユーティリティのパスワードの保存」を確認してください。

このセクションでは、FIPS モードの Identity Manager の環境設定と運用について説明します。

- 173 ページの第 7 章「FIPS 用の JRE の設定」
- 175 ページの第 8 章「FIPS モードでの Identity Manager エンジンの設定」
- 177 ページの第 9 章「FIPS モードでのファンアウトエージェントの設定」
- 179 ページの第 10 章「FIPS モードでの Identity Applications と Identity Reporting の設定」
- 183 ページの第 11 章「FIPS モードでのリモートローダの設定」

7 FIPS 用の JRE の設定

JRE を BCFIPS の approved-only モード用に設定するには、次の手順を実行します。

1 BCFIPS.jar ファイルを Java クラスパスに追加します。

必要な BCFIPS ファイルは次のとおりです。

- bc-fips-1.0.2.4.jar
- bcpkix-fips-1.0.7.jar
- bctls-fips-1.0.18.jar

2 <jre-root>/conf/security ディレクトリにある java.security ファイルを変更し、BCFIPS で定義された Java システムプロパティを設定します。

テキストエディタで、Web コンテナをホストしている JRE 用の上記のファイルを開きます。ファイルは <jre-root>/conf/security/java.security にあります。セキュリティプロバイダのリストを見つけて、既存のプロバイダをすべてコメント化し、そのエントリを以下に置き換えます。

- security.provider.1=BCFIPS
- security.provider.2=BCJSSE
- security.provider.3=SunJSSE
- security.provider.4=SUN
- security.provider.5=SunJCE

OSP はトラストストアを初期化する際に特定の TrustManager の実装を必要とするため、SunJSSE プロバイダ (Java 11) が必要です。セキュリティプロバイダを変更する以外に、BCFIPS JSSE KeyManager の実装を使用する必要もあります。これが必要な理由は主に、Sun の実装に不備があり、FIPS で必要な、キーに関する特定の決定を行えないためです。これは、java.security の次の行を置き換えることで実行します。

```
ssl.KeyManagerFactory.algorithm=SunX509
```

これを以下に置き換えます。

```
ssl.KeyManagerFactory.algorithm=PKIX
```

また、古い SSL および TLS のアルゴリズムを無効にすることもお勧めします。その理由は、TLS v2.0 よりも前の SSL/TLS バージョンに必要なアルゴリズムは、BCFIPS approved モードでは利用できないためです。古いアルゴリズムを無効にするには、jdk.tls.disabledAlgorithms エントリを見つけます。次に例を示します。

```
jdk.tls.disabledAlgorithms=SSLv3, RC4, MD5withRSA, DH keySize < 1024, \ EC
keySize < 224, DES40_CBC, RC4_40, 3DES_EDE_CBC
```

このリストに TLSv1 と TLSv1.1 を追加します。

```
jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, MD5withRSA, DH
keySize < 1024, \ EC keySize < 224, DES40_CBC, RC4_40, 3DES_EDE_CBC
```

TLSv1 および TLSv1.1 の無効化はオプションですが、無効にしておくと、OSP が、TLS v1.1 以前のみをサポートするサービス (LDAP など) との安全な接続を試みたときに、わかりにくいエラーメッセージを回避できます。以前の TLS バージョンを無効にした場合に表示されるエラーメッセージは、より理解しやすく、問題を直接指示します。

注 : eDirectory 8.8 (またはそれ以前) は、TLS v1.0 のみをサポートするため、セキュア LDAP とともに使用することはできません。セキュア LDAP を使用するには、eDirectory 9 以降を使用する必要があります。別の代替手段は、デフォルトのキーストアタイプを変更することです。`keystore.type=jks` または `keystore.type=pkcs12` の行を見つけて、それを `keystore.type=bcfks` に変更します。

---

8 FIPS モードでの Identity Manager エンジンの設定

Identity Manager エンジンに FIPS モードを設定する前に、次の前提条件を考慮する必要があります。

- セキュリティ強化拡張を有効にするため、次の jar ファイルを `/opt/netiq/common/jre/lib/` ディレクトリに配置します。
  - `bc-fips-1.0.2.4.jar`
  - `bcpkix-fips-1.0.7.jar`
  - `bctls-fips-1.0.18.jar`

  注 : これらの jar は、次の maven リポジトリからダウンロードできます。
  - bc-fips (https://mvnrepository.com/artifact/org.bouncycastle/bc-fips)
  - bcpkix-fips (https://mvnrepository.com/artifact/org.bouncycastle/bcpkix-fips)
  - bctls-fips (https://mvnrepository.com/artifact/org.bouncycastle/bctls-fips)
- これらの `.jar` ファイルの許可を有効にするため、次のコマンドを実行します。

  ```
  chown novlua:novlua <jar-file>
  chmod 755 <jar-file>
  ```
- 環境変数を設定するため、次のコマンドを実行します。

  ```
  export IS_FIPS_ENABLED=true
  export BC_FIPS_VERSION=1.0.2.4
  export JAVA_OPTS =" -Dorg.bouncycastle.fips.approved_only=true  -Dorg.bouncycastle.rsa.allow_multi_use=true"
  export CLASSPATH=/opt/netiq/common/jre/lib/bc-fips-1.0.2.4.jar
  ```

  設定後に、`echo` コマンドを使用してすべての環境変数を確認できます。

注 : ポート 636 で `dxcmd` にアクセスするには、LDAP 設定で SSL EC 証明書を有効にします。

手順

1. `/opt/netiq/common/jre/conf/security/` ディレクトリに移動します。
2. `java.security` ファイルのバックアップを取ります。

3 次の手順を実行して java.security ファイルを編集します。

3a List of providers and their preference orders という文字列を検索して、既存のプロバイダすべてをコメント化し、以下のエントリを追加します。

```
security.provider.1=BCFIPS
security.provider.2=BCJSSE
security.provider.3=SunJSSE
security.provider.4=SUN
security.provider.5=SunJCE
```

3b ssl.KeyManagerFactory.algorithm=SunX509 という文字列を検索してコメント化し、以下を追加します。

```
ssl.KeyManagerFactory.algorithm=PKIX
```

3c keystore.type という文字列を検索してコメント化し、以下を追加します。

```
keystore.type=bcfks
```

4 /opt/netiq/common/jre/lib/security/ ディレクトリに移動します。

5 cacerts ファイルのバックアップを取ります。

6 セキュリティ強化拡張をサポートするため、次のコマンドを使用して cacerts ファイルを bcfks に変換します。

```
keytool -importkeystore -srckeystore cacerts -srcstorepass changeit -
srcstoretype JKS -destkeystore cacerts -deststorepass changeit -
deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH
```

cacerts_bcfks ファイルの生成後、古い cacerts を削除します。cacerts_bcfks の名前を cacerts に変更します。

7 Identity Console にログインします。

8 **[IDM Administrator (IDM 管理者)]** に移動し、ドライバセットの アクションメニューをクリックします。

9 **[Driver Set Properties （ドライバセットのプロパティ)]** > **[Driver Set Configuration (ドライバセット環境設定)]** > **[Java Environment (Java 環境)]** に移動します。

10 **[JVM Options (JVM オプション)]** で以下を入力します。

```
Djavax.net.ssl.trustStore=/opt/netiq/common/jre/lib/security/cacerts -
Djavax.net.ssl.trustStorePassword=changeit -
Djavax.net.ssl.trustStoreType=BCFKS
```

11 **[保存]** ボタンをクリックします。

12 eDirectory を再起動します。

9 FIPSモードでのファンアウトエージェントの設定

Identity Manager ファンアウトエージェントに FIPS モードを設定する前に、次の前提条件を考慮する必要があります。

- セキュリティ強化拡張を有効にせずに、ファンアウトエージェントを起動して config フォルダ内にキーストアファイルを生成します。
- /opt/novell/dirxml/fanoutagent/lib/ ディレクトリに次の jar ファイルを配置します。
  - bc-fips-1.0.2.4.jar
  - bcpkix-fips-1.0.7.jar
  - bctls-fips-1.0.18.jar
- これらの .jar ファイルの許可を有効にするため、次のコマンドを実行します。

  ```
  chown root:bin <jar-file>
  ```

  ```
  chmod 755 <jar-file>
  ```
- 環境変数を設定するため、次のコマンドを実行します。

  ```
  export IS_FIPS_ENABLED=true
  ```

  ```
  export BC_FIPS_VERSION=1.0.2.4
  ```

  ```
  export JAVA_OPTS =" -Dorg.bouncycastle.fips.approved_only=true  -Dorg.bouncycastle.rsa.allow_multi_use=true"
  ```

  ```
  export CLASSPATH=/opt/novell/dirxml/fanoutagent/lib/bc-fips-1.0.2.4.jar
  ```

  設定後に、echo コマンドを使用してすべての環境変数を確認できます。

手順

1. /opt/netiq/common/jre/conf/security/ ディレクトリに移動します。
2. java.security ファイルのバックアップを取ります。
3. 次の手順を実行して java.security ファイルを編集します。

   3a List of providers and their preference orders という文字列を検索して、既存のプロバイダすべてをコメント化し、以下のエントリを追加します。

```
security.provider.1=BCFIPS
security.provider.2=BCJSSE
security.provider.3=SunJSSE
security.provider.4=SUN
security.provider.5=SunJCE
```

3b `ssl.KeyManagerFactory.algorithm=SunX509` という文字列を検索してコメント化し、以下を追加します。

```
ssl.KeyManagerFactory.algorithm=PKIX
```

3c `keystore.type` という文字列を検索してコメント化し、以下を追加します。

```
keystore.type=bcfks
```

4 `/opt/netiq/common/jre/lib/security/` の場所に移動します。

5 `cacerts` ファイルのバックアップを取ります。

6 セキュリティ強化拡張をサポートするため、次のコマンドを使用して `cacerts` ファイルを `bcfks` に変換します。

```
keytool -importkeystore -srckeystore cacerts -srcstorepass changeit -
srcstoretype JKS -destkeystore cacerts -deststorepass changeit -
deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH
```

7 `keystore` ファイルを `jks` から `bcfks` に変換するため、次のコマンドを実行します。

```
keytool -importkeystore -srckeystore /opt/novell/dirxml/fanoutagent/
config/keystore.jks -srcstorepass changeit -srcstoretype PKCS12 -
destkeystore /opt/novell/dirxml/fanoutagent/config/keystore.jks -
deststorepass changeit -deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH
```

8 `/opt/novell/dirxml/fanoutagent/bin/` ディレクトリに移動し、`startAgent.sh` ファイルに次の変更を加えます。

```
${JAVA} --add-opens=java.base/sun.security.internal.spec=ALL-UNNAMED -
Djavax.net.ssl.trustStore=/opt/netiq/common/jre/lib/security/cacerts -
Djavax.net.ssl.trustStorePassword=changeit -Dkeystore.type=BCFKS -
Dorg.bouncycastle.fips.approved_only=true -
Dorg.bouncycastle.rsa.allow_multi_use=true -cp ${CLASS_PATH}
com.netiq.idm.fanoutagent.FanoutAgent $filename $SERVERONLY &
```

9 次のコマンドで、ファンアウトエージェントを起動します。

```
./startAgent
```

10 FIPS モードでの Identity Applications と Identity Reporting の設定

Identity Manager Reporting および Identity Applications に FIPS モードを設定する前に、次の前提条件を考慮する必要があります。

- Identity ApplicationsおよびIdentity Manager Reportingのコンポーネントをホストしているサーバ上で Tomcat サービスを停止します。
- 管理者が Identity Manager エンジンのシステムの dxcmd ツールにログインできることを確認します。

手順

1. FIPS 用の新しいフォルダを作成します。たとえば、/home/fips です。
2. 作成したフォルダ内に次のファイルのバックアップを作成します。

ファイル	既存のファイルパス
idm.jks	/opt/netiq/idm/apps/tomcat/conf
tomcat.ks	
encrypt-keys.pkcs12	
server.xml	
ism-configuration.properties	
configupdate.sh	/opt/netiq/idm/apps/configupdate
configupdate.bat.properties	

3. /mnt/osp/packages/misc/ ディレクトリに移動し、osp-tomcat8-pwd-handler-<version>.jar を /opt/netiq/idm/apps/tomcat/lib ディレクトリにコピーします。
4. /opt/netiq/idm/apps/tomcat/bin/ ディレクトリに移動し、setenv.sh ファイルを次のように変更します。
   - JAVA_OPTS= -Dorg.bouncycastle.fips.approved_only=true -Dinternal.osp.framework.fips-password=novell -Dorg.bouncycastle.rsa.allow_multi_use=true -Djavax.net.ssl.keyStoreType=BCFKS -Djavax.net.ssl.trustStoreType=BCFKS -Djavax.net.ssl.trustStore=/opt/netiq/idm/apps/tomcat/conf/idm.bcfks -Djavax.net.ssl.trustStorePassword=novell

- BCFIPS_JARS=/opt/netiq/common/jre/lib/bc-fips-1.0.2.4.jar:/opt/netiq/common/jre/lib/bcpkix-fips-1.0.7.jar:/opt/netiq/common/jre/lib/bctls-fips-1.0.18.jar
- export CLASSPATH="${BCFIPS_JARS}"
- CATALINA_OPTS を検索し、encrypt-keys.pkcs12 を encrypt-keys.bcfks に変更し、storetype を bcfks に変更します。

5 /opt/netiq/idm/apps/tomcat/conf/ ディレクトリに移動し、以下のコマンドを実行します。

```
keytool -importkeystore -srckeystore idm.jks -srcstorepass novell -
srcstoretype PKCS12 -destkeystore idm.bcfks -deststorepass novell -
deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH

keytool -importkeystore -srckeystore tomcat.ks -srcstorepass novell -
srcstoretype PKCS12 -destkeystore tomcat.bcfks -deststorepass novell -
deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH

keytool -importkeystore -srckeystore idm.jks -srcstorepass novell -
srcstoretype PKCS12 -destkeystore /opt/netiq/idm/apps/osp/osp.bcfks -
deststorepass novell -deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH

keytool -importkeystore -srckeystore tomcat.ks -srcstorepass novell -
srcstoretype PKCS12 -destkeystore /opt/netiq/idm/apps/osp/osp.bcfks -
deststorepass novell -deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH --noprompt

java -jar <mount-location>/osp/packages/misc/osp-keymanager-6.7.1.jar -
genkeys -keystore /opt/netiq/idm/apps/osp/osp.bcfks -storepass novell -
storetype BCFKS -application idm -providerClass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerPath
$CLASSPATH -keypass novell -use Encrypting,Signing,TLS

keytool -importkeystore -srckeystore encrypt-keys.pkcs12 -srcstorepass
changeit -srcstoretype PKCS12 -destkeystore encrypt-keys.bcfks -
deststorepass changeit -deststoretype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH
```

6 server.xml ファイルを次のように編集します。

```
<!--Connector port="8543"
protocol="org.apache.coyote.http11.Http11NioProtocol" maxThreads="150"
SSLEnabled="true" scheme="https" secure="true" clientAuth="false"
sslProtocol="TLSv1.2" keystoreFile="conf/tomcat.ks"
keystorePass="novell" maxHttpHeaderSize="65536"
sslEnabledProtocols="TLSv1.2" /-->

<Connector
protocol="com.microfocus.osp.tomcat.http.PasswordHttp11NioProtocol"
port="8543" maxThreads="150" maxHttpHeaderSize="65536"
SSLEnabled="true" scheme="https" secure="true">
<SSLHostConfig protocol="TLS" protocols="TLSv1.2"
ciphers="TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,TLS_DHE_DSS_WITH_AES_256_
GCM_SHA384,TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AES_12
8_CBC_SHA,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256
_CBC_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_
GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WIT
H_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_EC
DSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECD
HE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_
ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA
_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_CCM,TLS_RSA_WITH_AES_256
_CCM,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_
RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA"
truststoreFile="/opt/netiq/idm/apps/osp/osp.bcfks"
truststorePassword="novell" truststoreType="BCFKS" > <Certificate
certificateKeystoreFile="/opt/netiq/idm/apps/tomcat/conf/tomcat.bcfks"
certificateKeystorePassword="novell" certificateKeyAlias="ssl
certificatedns" certificateKeystoreType="BCFKS" />
</Certificate>
</SSLHostConfig>
</Connector>
```

7 ism-configuration.properties ファイルを次のように編集します。

```
com.netiq.idm.osp.ssl-keystore.file = /opt/netiq/idm/apps/tomcat/conf/
tomcat.bcfks

com.netiq.idm.osp.ssl-keystore.type = BCFKS

com.netiq.idm.osp.oauth-keystore.file =/opt/netiq/idm/apps/tomcat/conf/
osp.bcfks

com.netiq.idm.osp.oauth-keystore.type = BCFKS

com.netiq.rpt.ssl-keystore.file = /opt/netiq/idm/apps/tomcat/conf/
idm.bcfks

com.netiq.rpt.ssl-keystore.type = BCFKS

DirectoryService/realms/jndi/params/KEYSTORE_PATH = /opt/netiq/idm/
apps/tomcat/conf/idm.bcfks

com.sssw.fw.security.sigcert.truststore.type = BCFKS

com.netiq.idm.osp.oauth-truststore.file = /opt/netiq/idm/apps/tomcat/
conf/idm.bcfks
```

```
com.netiq.idm.ua.ldap.keystore-type = BCFKS

com.netiq.idm.osp.oauth-key-alias = ssl certificatedns
```

8 /opt/netiq/idm/apps/tomcat/conf/ ディレクトリに移動し、以下のコマンドを実行します。

```
keytool -storepasswd -new <new master keystore password> -keystore /
opt/netiq/idm/apps/tomcat/conf/encrypt-keys.bcfks -storepass changeit -
storetype BCFKS -providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH

keytool -keypasswd -alias ism.config.encrypt.0 -keypass changeit -new
<new master keystore password> -keystore /opt/netiq/idm/apps/tomcat/
conf/encrypt-keys.bcfks -storepass novell  -storetype BCFKS -
providerclass
org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider -providerpath
$CLASSPATH
```

このコマンドを実行すると、keystorepass の値が生成されます。

```
java -cp ../lib/obscurity-0.13.2-uber.jar
com.netiq.ism.obfuscate.server.util.PasswordObfuscator novell
```

9 keystorepass の値を取得した後で、次の行を追加して ism-sensitive.properties ファイルを更新します。

```
com.microfocus.ism.master-keypass = <keystorepass-value>
com.microfocus.ism.master-storepass = <keystorepass-value>
```

10 次の jar を configupdate/lib ディレクトリにコピーします。

- bc-fips-1.0.2.4.jar
- bcpkix-fips-1.0.7.jar
- bctls-fips-1.0.18.jar

11 configupdate.bat.properties ファイルを次のように編集します。

```
master_keystore=/opt/netiq/idm/apps/tomcat/conf/encrypt-keys.bcfks
master_keystore_type=bcfks
```

12 configupdate.sh ファイルを次のように編集します。

```
set "JAVA_OPTS= -Dorg.bouncycastle.fips.approved_only=true -
Dinternal.osp.framework.fips-password=novell -
Dorg.bouncycastle.rsa.allow_multi_use=true -
Djavax.net.ssl.keyStoreType=BCFKS -Djavax.net.ssl.trustStoreType=BCFKS
-Djavax.net.ssl.trustStore=/opt/netiq/idm/apps/tomcat/conf/idm.bcfks -
Djavax.net.ssl.trustStorePassword=novell  "

set CLASSPATH=/opt/netiq/common/jre/lib/bc-fips-1.0.2.4.jar:/opt/netiq/
common/jre/lib/bcpkix-fips-1.0.7.jar:/opt/netiq/common/jre/lib/bctls-
fips-1.0.18.jar:%CLASSPATH%
```

13 configupdate.sh ユーティリティを起動します。

14 Tomcat を起動します。

11 FIPS モードでのリモートローダの設定

リモートローダに FIPS モードを設定するには、次の手順を実行します。

1. /opt/novell/dirxml/remoteloader/lib/ ディレクトリに次の jar ファイルを配置します。
   - bc-fips-1.0.2.4.jar
   - bcpkix-fips-1.0.7.jar
   - bctls-fips-1.0.18.jar
2. /opt/novell/dirxml/doc/ ディレクトリに移動します。
3. RL インスタンスを設定する際に、環境設定ファイルに is_fips_enabled=true フラグを追加します。たとえば、サンプル環境設定ファイル config8000.txt にこのフラグを追加できます。
4. ファイルを保存し、rdxml コマンドを使用してリモートローダインスタンスを起動します。

V Identity Manager のデータの新しいインストールへのマイグレート

このセクションでは、Identity Manager のコンポーネントの既存データを新しいインストールにマイグレートするための情報を提供します。ほとんどのマイグレーションタスクは、識別情報アプリケーションに適用されます。Identity Manager のコンポーネントをアップグレードするには、203 ページのパート VI「Identity Manager のアップグレード」を参照してください。アップグレードとマイグレーションの違いの詳細については、207 ページの「アップグレードプロセスの理解」を参照してください。

12 Identity Manager をマイグレートする準備

このセクションでは、Identity Manager ソリューションを新しいインストールにマイグレートする準備について説明します。

マイグレーションを実行するためのチェックリスト

マイグレーションを実行するために、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. Identity Manager のデータをマイグレートするために最新のインストールキットを用意していることを確認します。
❑	2. eDirectory を識別ボールトでサポートされている最新バージョンにアップグレードします。詳細については、219 ページの 「アイデンティティボールトのアップグレード」を参照してください。
❑	3. 現在の Identity Manager サーバにあるのと同じ eDirectory のレプリカを、新しいサーバに追加します。詳細については、190 ページの 「Identity Manager エンジンの新しいサーバへのマイグレート」を参照してください。
❑	4. 新しいサーバで Identity Manager をインストールします。詳細については、37 ページの 「Identity Manager のインストールの計画」を参照してください。
❑	5. (状況によって実行) ドライバセットのいずれかのドライバがリモートローダドライバである場合、各ドライバのリモートローダサーバをアップグレードします。詳細については、222 ページの 「リモートローダのアップグレード」を参照してください。
❑	6. (状況によって実行) 古いサーバでユーザアプリケーションを実行している場合、そのコンポーネントとドライバをアップデートします。詳細については、189 ページの 「前提条件」を参照してください。
❑	7. ドライバごとに、サーバ固有の情報を変更します。詳細については、191 ページの「Designer でサーバ固有の情報をコピーする」を参照してください。
❑	8. (状況によって実行) ユーザアプリケーションを実行している場合、ユーザアプリケーションのサーバ固有情報を古いサーバのものから新しいサーバのものに更新します。詳細については、191 ページの 「ドライバセットのサーバ固有情報のコピー」を参照してください。
❑	9. ドライバをアップデートしてパッケージフォーマットにします。詳細については、226 ページの 「Identity Manager ドライバのアップグレード」を参照してください。

	チェックリストの項目
❑	10. (状況によって実行) カスタムポリシーとルールがある場合、カスタマイズされている設定を復元します。詳細については、245 ページの 「ドライバへのカスタムポリシーとルールの復元」を参照してください。
❑	11. Identity Reporting と関連ドライバをインストールします。詳細については、196 ページの 「Identity Reporting の移行」を参照してください。
❑	12. ドライバセットから古いサーバを削除します。詳細については、244 ページの 「ドライバセットから古いサーバを削除する」を参照してください。
❑	13. アップグレードした Identity Manager ソリューションをアクティベートします。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Activating Identity Manager」を参照してください。

13 Identity Manager の新しいサーバへのマイグレート

このセクションでは、ユーザアプリケーションから新しいサーバ上の識別情報アプリケーションにマイグレートする方法について説明します。既存のインストールをアップグレードできない場合もマイグレーションが必要になる可能性があります。このセクションでは、次のアクティビティについて説明します。

- 189 ページの 「前提条件」
- 189 ページの 「Designer プロジェクトのマイグレーションの準備」
- 190 ページの 「Identity Manager エンジンの新しいサーバへのマイグレート」
- 191 ページの 「ドライバセットのサーバ固有情報のコピー」
- 193 ページの 「Identity Applications の移行」
- 196 ページの 「Identity Reporting の移行」

前提条件

- Identity Manager ソリューションのディレクトリとデータベースのバックアップ
- 識別情報アプリケーションを除いて、Identity Manager のコンポーネントの最新バージョンがインストールされていることを確認します。

  注 : 現在のユーザアプリケーションデータベースを引き続き使用するには、インストールプログラムで [**既存のデータベース**] を指定します。詳細については、67 ページの第 4 章「Identity Manager のインストール」を参照してください。
- 識別ボールトのヘルスチェックを実行し、スキーマが適切に拡張されていることを確認します。TID 3564075 を使用してヘルスチェックを完了します。
- 既存のユーザアプリケーションドライバを Designer にインポートします。

Designer プロジェクトのマイグレーションの準備

Designer プロジェクトをアーカイブする必要があります。これはドライバのマイグレーション前の状態を表します。

ドライバをマイグレートする前に、Designer プロジェクトのマイグレーションを準備するためにいくつかの手順を実行する必要があります。

注 : マイグレートする Designer プロジェクトが存在しない場合、[ファイル] > [インポート] > [プロジェクト (識別ボールトから)] を使用して新しいプロジェクトを作成します。

1 Designer を起動します。

2 (状況によって実行) マイグレートするユーザアプリケーションを含む Designer プロジェクトが既存の場合、そのプロジェクトをバックアップします。

2a プロジェクトビューでプロジェクト名を右クリックして、[プロジェクトのコピー] を選択します。

2b プロジェクトの名前を指定して、[OK] をクリックします。

3 既存のプロジェクトのスキーマをアップデートするには、次の手順を実行します。

3a [モデラー] ビューで [識別ボールト] を選択します。

3b [ライブ] > [スキーマ] > [インポート] の順に選択します。

4 (オプション) プロジェクトの Identity Manager のバージョン番号が適切であることを検証するには、次の手順を実行します。

4a [モデラー] ビューで、[識別ボールト] を選択し、[プロパティ] をクリックします。

4b 左のナビゲーションメニューで [サーバリスト] を選択します。

4c サーバを選択し、[編集] をクリックします。

[Identity Manager バージョン] に最新バージョンが表示されるはずです。

Identity Manager エンジンの新しいサーバへのマイグレート

1 [Software and License (ソフトウェアおよびライセンス)] ページ (https://sld.microfocus.com/) から `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` をマウントします。

3 `.iso` から Identity Manager エンジンをインストールします。

```
./install.sh
```

4 Identity Manager エンジンを設定します。

```
configure.sh
```

5 新しいサーバ上で、ドライバセットパーティションとデータパーティションの読み取り / 書き込みレプリカを作成します。

6 新しいサーバを Designer プロジェクトに追加します。

ドライバセットのサーバ固有情報のコピー

各ドライバおよびドライバセットに保存されているサーバ固有のすべての情報を、新しいサーバの情報にコピーする必要があります。その中には、新しいサーバに存在せず、コピーする必要がある、ドライバセットの GCV と他のデータも含まれます。サーバ固有の情報は、次のものに含まれています :

- グローバル構成値
- エンジン制御値
- 名前付きパスワード
- ドライバの認証情報
- ドライバの起動オプション
- ドライバパラメータ
- ドライバセットデータ

これは、Designer で実行できます。Designer を使用する場合には、自動的なプロセスです。Identity Console を使用する場合には、手動のプロセスです。バージョン 3.5 より古い Identity Manager サーバを 3.5 以降の Identity Manager サーバに移行する場合、Identity Console を使用する必要があります。サポートされている他のすべてのマイグレーションパスの場合は、Designer を使用できます。

- 191 ページの 「Designer でサーバ固有の情報をコピーする」
- 192 ページの 「Identity Console でのサーバ固有情報の変更」
- 193 ページの 「ユーザアプリケーションのサーバ固有の情報を変更する」

Designer でサーバ固有の情報をコピーする

この手順は、ドライバセットに保存されているすべてのドライバに影響します。

1. Designer で、プロジェクトを開きます。
2. [アウトライン] タブで、サーバを右クリックして、[移行] を選択します。
3. 概要を読んで新しいサーバにマイグレートされる項目を確認し、[次へ] をクリックします。
4. 選択可能なサーバのリストからターゲットサーバを選択して、[次へ] をクリックします。

   リストに表示されているサーバだけが、現在ドライバセットに関連付けられておらず、ソースサーバの Identity Manager のバージョンと等しいか新しいサーバです。
5. 次のいずれかのオプションを選択します。
   - **ターゲットサーバをアクティブにする** : ソースサーバの設定をターゲットサーバにコピーして、ソースサーバのドライバを無効にします。このオプションを使用することをお勧めします。
   - **ソースサーバをアクティブのままにする** : 設定をコピーせずに、ターゲットサーバのすべてのドライバを無効にします。

- **ターゲットソースサーバの両方をアクティブにする**：ソースサーバの設定をターゲットサーバにコピーし、ソースサーバまたはターゲットサーバのドライバは無効にしません。このオプションを使用する場合は、各ドライバが新しいサーバで正常に実行されるまで、移行元サーバまたは移行先サーバのいずれかでドライバを無効にする必要があります。NetIQ では、ドライバのイベントを一度に 2 台のサーバにキャッシュしないことをお勧めします。

  注：ドライバがサーバ上でアクティブな場合は、ドライバはイベントをキャッシュします。ドライバを無効にすると、サーバ上のイベントのキャッシュが停止し、キャッシュ (.TAO) ファイルが削除されます。

  ---

6 **[移行]** をクリックします。

7 変更されたドライバを識別ボールトに展開します。

詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「識別ボールトへのドライバセットの展開」を参照してください。

8 ドライバを起動します。

詳細については、225 ページの 「ドライバの起動」を参照してください。

Identity Console でのサーバ固有情報の変更

1 Identity Console で、**[Identity Manager Administration (Identity Manager の管理)]** に移動します。

2 必要なドライバを含むドライバセットオブジェクトをクリックします。

3 [Action menu (アクションメニュー)] アイコン をクリックし、**[Stop driver (ドライバの停止)]** をクリックします。

4 ドライバアイコンをクリックします。

5 **[環境設定]** タブに移動します。

6 古いサーバの情報を含むすべてのサーバ固有のドライバパラメータ、グローバル環境設定値、エンジン制御値、名前付きパスワード、ドライバ認証データ、およびドライバの起動オプションを、新しいサーバの情報にコピーまたはマイグレートします。最大ヒープサイズ、Java の設定などのグローバル環境設定値やドライバセットのその他のパラメータは、古いサーバの値と同一の値を持つ必要があります。

7 **[保存]** をクリックして、すべての変更を保存し、終了します。

8 [Action menu (アクションメニュー)] をクリックし、**[Start driver (ドライバの起動)]** をクリックします。

9 ドライバセットのドライバごとに、ステップ 3 ～ステップ 8 を繰り返します。

ユーザアプリケーションのサーバ固有の情報を変更する

新しいサーバを認識するようにユーザアプリケーションを再設定する必要があります。`configupdate.sh` を実行します。

1. デフォルトでユーザアプリケーションのインストールサブディレクトリにある設定更新ユーティリティに移動します。
2. コマンドプロンプトで、設定更新ユーティリティを起動します。

   ```
   configupdate.sh
   ```

3. 114 ページの 「識別情報アプリケーションの設定の管理」の説明に従って、値を指定します。

Identity Applications の移行

大文字と小文字を区別する照合をデータベースに対して使用しないでください。大文字と小文字を区別する照合はサポートされていません。大文字と小文字を区別する照合により、移行中に重複キーエラーが発生する場合があります。重複キーエラーが発生した場合は、照合を確認して修正してから、識別情報アプリケーションを再インストールします。サポートされる照合は SQL_Latin1_General_CP1_CI_AS のみです。

Identity Applications を移行する前に、`/opt/netiq/common/openssl/lib64/` ディレクトリから、`libssl.so.1.0.0` および `libcrypto.so.1.0.0` ライブラリをインストールする必要があります。

Identity Applications の移行には以下が含まれます。

- 193 ページの 「データベースの新しいサーバへの移行」
- 195 ページの 「新しいサーバへの Identity Applications のインストール」

データベースの新しいサーバへの移行

ユーザアプリケーションデータベースが PostgreSQL 上にある場合、以下の手順を実行します。

1. PostgreSQL がインストールされているサーバに `postgres` ユーザとしてログインします。

   ```
   #su - postgres
   ```

2. `.sql` ファイルにデータをエクスポートします。Postgres ユーザが、ファイルをエクスポートするディレクトリにフルアクセスできることを確認します。

   ```
   pg_dump -p <portnumber> -U <username> -d <dbname> -f <export location>
   ```

   次に例を示します。

   ```
   pg_dump -p 5432 -U postgres -d idmuserappdb -f /tmp/idmuserappdb.sql
   ```

3. PostgreSQL をインストールする新しいサーバにログインします。

4 PostgreSQL をインストールします。

4a Identity_Manager_4.9.0_Linux.iso をマウントした場所に移動します。

4b /common/packages/postgres/ ディレクトリに移動します。

4c PostgreSQL をインストールするには、次のコマンドを使用します。

```
rpm -ivh netiq-postgresql-9.6.6-0.noarch.rpm
```

4d 次のコマンドを使用して、グループを postgres ユーザに関連付けます。

```
/usr/sbin/usermod -a -G postgres postgres
```

4e /etc/passwd ファイルにある postgres ユーザのホームディレクトリパスを /opt/netiq/idm/postgres/ に変更します。

4e1 /etc/ ディレクトリに移動します。

4e2 passwd ファイルを編集します。

```
vi /etc/passwd
```

4e3 postgres ユーザのホームディレクトリを /opt/netiq/idm/postgres/ に変更します。

4f postgres ユーザとしてログインします。

次に例を示します。

```
su - postgres
```

4g PostgreSQL のインストール場所に data ディレクトリを作成します。

mkdir -p <POSTGRES_HOME>/data。ここで、<POSTGRES_HOME> は /opt/netiq/idm/postgres です

次に例を示します。

```
mkdir -p /opt/netiq/idm/postgres/data
```

4h PostgreSQL home ディレクトリをエクスポートします .

export PGHOME=<postgres ホームディレクトリパス >

次に例を示します。

```
export PGHOME=/opt/netiq/idm/postgres
```

4i PostgreSQ パスワードをエクスポートします。

```
export PGPASSWORD=<enter the database password>
```

4j データベースを初期化します。

```
LANG=en_US.UTF-8 <POSTGRES_HOME>/bin/initdb -D <POSTGRES_HOME>/data
```

次に例を示します。

```
LANG=en_US.UTF-8 /opt/netiq/idm/postgres/bin/initdb -D /opt/netiq/idm/postgres/data
```

4k /opt/netiq/idm/postgres/bin ディレクトリに移動します。

4l 以下のコンポーネントのデータベースを作成します。

```
$ createdb idmuserappdb
$ psql -s idmuserappdb
# create user idmadmin password 'somepassword';
# GRANT ALL PRIVILEGES ON DATABASE idmuserappdb TO idmadmin;
# ALTER DATABASE idmuserappdb OWNER TO idmadmin;
```

5 新しい PostgreSQL データベースにデータをインポートします。

5a 手順 2 でエクスポートされたファイルを、postgres ユーザがフルアクセスできる場所にコピーします。

5b 次のコマンドを実行して、PostgreSQL データベースにデータをインポートします。

```
psql -d <dbname> -U <username> -f <full path where the exported file
is located> -W
```

次に例を示します。

```
psql -d idmuserappdb -U idmadmin -f /tmp/idmuserappdb.sql -W
```

新しいサーバへの Identity Applications のインストール

1 NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 `.iso` をマウントします。

3 iso の内容を、書き込みアクセス権のある別のディレクトリにコピーします。

次に例を示します。

```
cp /mnt/* /home
```

4 ユーザアプリケーションと役割およびリソースサービスドライバの展開をスキップするように、環境設定ファイルの内容を編集します。

注 : デフォルトでは、Identity Applications インストールは役割およびリソースサービスとユーザアプリケーション用のドライバを作成して展開します。

4a `/home/user_application` ディレクトリに移動します。

4b `configure.sh` ファイルを編集します。

```
vi configure.sh
```

4c 次の行をコメントアウトします :

```
install_service_drivers "UA" "${ID_VAULT_ADMIN_LDAP}"
"${ID_VAULT_PASSWORD}" "${ID_VAULT_HOST}" ${ID_VAULT_LDAPS_PORT}
"cn=${ID_VAULT_DRIVER_SET},${ID_VAULT_DEPLOY_CTX}"
```

4d `configure.sh` ファイルを保存します。

5 `/mnt` ディレクトリから Identity Applications をインストールします。

```
./install.sh
```

6 `/mnt` ディレクトリから Identity Applications を設定します。

```
./configure.sh
```

7 [カスタム環境設定] を選択して、次のプロンプトに対して [No] を選択します。

```
Do you want to configure PostgreSQL database on current server?
```

8 /opt/netiq/idm/apps/configupdate ディレクトリにある設定更新ユーティリティに移動し、環境設定が正しいことを確認します。

```
./configupdate.sh
```

Identity Reporting の移行

Identity Reporting の移行には以下が含まれます。

- 196 ページの 「Identity Reporting 用ドライバの更新」
- 196 ページの 「Identity Reporting 用ドライバの展開」
- 197 ページの 「新しいデータベースへの既存のデータの移行」
- 201 ページの 「新しい Reporting サーバの設定」
- 201 ページの 「データ同期ポリシーの作成」

Identity Reporting 用ドライバの更新

1 データ収集サービスドライバと管理対象サービスゲートウェイドライバのパッケージをアップグレードします。詳細については、『NetIQ Designer for Identity Manager Administration Guide』の「Upgrading Installed Packages」を参照してください。

注 : パッケージのアップグレード中に、新しい Identity Reporting サーバの詳細を指定するようにしてください。

---

2 ドライバを展開します。詳細については、196 ページの 「Identity Reporting 用ドライバの展開」を参照してください。

3 (状況によって実行) 4.5.x から移行しているときに、EAS データを移行する場合は、197 ページの 「新しいデータベースへの既存のデータの移行」の手順を実行します。

Identity Reporting 用ドライバの展開

1 Designer でプロジェクトを開いて、マイグレートされたオブジェクトに対してプロジェクトチェッカを実行します。

詳細については、『*NetIQ Identity Manager - Administrator's Guide to Designing the Identity Applications*』の「Validating Provisioning Objects」を参照してください。設定に検証エラーがある場合はそのことが表示されます。これらのエラーを修正するまでは、ドライバを展開できません。

2 [アウトライン] ビューで、データ収集サービスドライバを右クリックします。

3 [展開] を選択します。

4 ドライバセットのデータ収集サービスドライバごとにこのプロセスを繰り返します。データ収集サービスドライバが展開されたら、管理対象サービスゲートウェイドライバに対してこのプロセスを繰り返します。

新しいデータベースへの既存のデータの移行

移行時に確実にエラーないようにするため、必要な役割およびテーブルスペースを作成する必要があります。

新しい PostgreSQL データベースの準備

1 EAS を停止して、どのイベントも EAS サーバに送信されていないことを確認します。

2 Identity Console を使用して、DCS ドライバを停止します。

2a Identity Console にログインします。

2b DCS ドライバを停止します。

2c ドライバプロパティを編集して、起動オプションを [手動] に変更します。

この手順により、ドライバが自動的に起動しないようにします。

3 次の SQL コマンドを実行して、PGAdmin を使用して必要な役割、テーブルスペース、およびデータベースを作成します。

この手順により、移行時にエラーがないことが保証されます。

3a 必要な役割を作成するには、次のコマンドを実行します。

```
CREATE ROLE esec_app
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;

CREATE ROLE esec_user
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;

CREATE ROLE admin LOGIN
  ENCRYPTED PASSWORD '<specify the password for admin>'
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;
GRANT esec_user TO admin;

CREATE ROLE appuser LOGIN
  ENCRYPTED PASSWORD '<specify the password for appuser>'
  NOSUPERUSER INHERIT NOCREATEDB CREATEROLE;
GRANT esec_app TO appuser;

CREATE ROLE dbauser LOGIN
  ENCRYPTED PASSWORD '<specify the password for dbauser>'
  SUPERUSER INHERIT CREATEDB CREATEROLE;
```

```
CREATE ROLE idmrptsrv LOGIN
  ENCRYPTED PASSWORD '<specify the password for idmrptsrv>'
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;
GRANT esec_user TO idmrptsrv;

CREATE ROLE idmrptuser LOGIN
  ENCRYPTED PASSWORD '<specify the password for idmrptuser>'
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;

CREATE ROLE rptuser LOGIN
  ENCRYPTED PASSWORD '<specify the password for rptuser>'
  NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE;
GRANT esec_user TO rptuser;
```

3b (状況によって実行) 次のコマンドを実行して、テーブルスペースを作成します。

```
 CREATE TABLESPACE sendata1
  OWNER dbauser
  LOCATION '<provide the location where table space has to be
created>';
```

次に例を示します。

```
CREATE TABLESPACE sendata1
  OWNER dbauser
  LOCATION '</opt/netiq/idm/apps/postgres/data>';
```

3c (状況によって実行) 既存の EAS データを移行する場合は、次のコマンドを実行して SIEM データベースを作成することをお勧めします。

```
CREATE DATABASE "SIEM"
  WITH OWNER = dbauser
       ENCODING = 'UTF8'
       TABLESPACE = sendata1
       CONNECTION LIMIT = -1;
```

3d 次のコマンドを実行して、Reporting データベースを作成します。

```
CREATE DATABASE "idmrptdb"
  WITH OWNER = dbauser
       ENCODING = 'UTF8'
       CONNECTION LIMIT = -1;
```

EAS データのエクスポート

現在 Identity Manager 4.5.x を実行していて、既存の EAS データを SIEM データベースに移行する場合にのみ、以下のアクションを実行します。

- 199 ページの 「EAS データのエクスポート」
- 199 ページの 「新しい PostgreSQL データベースへの EAS データのインポート」

EAS データのエクスポート

1 EAS を停止して、どのイベントも EAS サーバに送信されていないことを確認します。

2 Identity Console を使用して、DCS ドライバを停止します。

2a Identity Console にログインします。

2b DCS ドライバを停止します。

2c ドライバプロパティを編集して、起動オプションを **[手動]** に変更します。

この手順により、ドライバが自動的に起動しないようにします。

3 EAS データベースからファイルにデータをエクスポートします。

3a EAS ユーザアカウントにログインします。

```
# su - novleas
```

3b たとえば、`/home/novleas` のように、EAS ユーザがフルアクセスできる場所を指定します。

3c PostgreSQL インストールディレクトリに移動し、次のコマンドを実行します。

次に例を示します。

```
export PATH=/opt/novell/sentinel_eas/3rdparty/postgresql/bin/:$PATH
```

```
export LD_LIBRARY_PATH=/opt/novell/sentinel_eas/3rdparty/
postgresql/lib/:$LD_LIBRARY_PATH
```

3d 次のコマンドを使用して、データを `.sql` ファイルにエクスポートします。

```
.·/pg_dump -p <portnumber> -U <username> -d <dbname> -f <export
location>
```

次に例を示します。

```
./pg_dump -p 15432 -U dbauser SIEM -f /home/novleas/SIEM.sql
```

新しい PostgreSQL データベースへの EAS データのインポート

1 EAS を停止して、どのイベントも EAS サーバに送信されていないことを確認します。

2 Identity Console を使用して、DCS ドライバを停止します。

2a Identity Console にログインします。

2b DCS ドライバを停止します。

2c ドライバプロパティを編集して、起動オプションを **[手動]** に変更します。

この手順により、ドライバが自動的に起動しないようにします。

3 新しい PostgreSQL データベースに EAS データをインポートします。

3a エクスポートされた `.sql` ファイルを、postgres ユーザがフルアクセスできる場所にコピーします。たとえば、`/opt/netiq/idm/postgres` にコピーします。

3b 次のコマンドを実行して、PostgreSQL データベースに EAS データをインポートします。

```
psql -d <dbname> -U <username> -f <full path where the exported file
is located>
```

次に例を示します。

```
psql -d SIEM -U postgres -f /opt/netiq/idm/apps/postgres/SIEM.sql
```

4 移行ログエラーを確認して、解決します。

Reporting データのエクスポート

現在 Identity Manager 4.7.x を実行していて、既存の Reporting データを新しいサーバに移行する場合にのみ、以下のアクションを実行します。

- 200 ページの 「Reporting データのエクスポート」
- 200 ページの 「新しい Reporting サーバへのデータのインポート」

Reporting データのエクスポート

1 PostgreSQL がインストールされているサーバに `postgres` ユーザとしてログインします。

```
#su - postgres
```

2 `.sql` ファイルにデータをエクスポートします。Postgres ユーザが、ファイルをエクスポートするディレクトリにフルアクセスできることを確認します。

```
pg_dump -p <portnumber> -U <username> -d <dbname> -f <export location>
```

次に例を示します。

```
pg_dump -p 5432 -U dbauser -W idmrptdb -f /tmp/idmrptdb.sql
```

新しい Reporting サーバへのデータのインポート

1 PostgreSQL がインストールされているサーバに `postgres` ユーザとしてログインします。

```
#su - postgres
```

2 新しい PostgreSQL データベースにデータをインポートします。

2a エクスポートされた `.sql` ファイルを、postgres ユーザがフルアクセスできる場所にコピーします。

2b 次のコマンドを実行して、PostgreSQL データベースにデータをインポートします。

```
psql -d <dbname> -U <username> -f <full path where the exported file is located>
```

次に例を示します。

```
psql -d idmrptdb -U dbauser -f /tmp/idmrptdb.sql
```

3 移行ログエラーを確認して、解決します。

新しい Reporting サーバの設定

1 ［Software and License (ソフトウェアおよびライセンス)］ページ (https://sld.microfocus.com/) から `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 `.iso` をマウントします。

3 ISO がマウントされた場所から、次のコマンドを実行します。

```
./install.sh
```

4 Identity Reporting を設定します。

```
./configure.sh
```

5 **[カスタム環境設定]** を選択して、次のプロンプトに対して **[No]** を選択します。

```
Do you want to configure PostgreSQL database on current server?
Do you want to install a new driverset?
```

注 : デフォルトでは、Identity Reporting インストールは管理対象サービスゲートウェイおよびデータ収集サービス用のドライバを作成して展開します。

---

6 `/opt/netiq/idm/apps/configupdate` ディレクトリにある設定更新ユーティリティに移動し、環境設定が正しいことを確認します。

```
./configupdate.sh
```

データ同期ポリシーの作成

レポーティングサーバを設定した後は、SLM for IGA からレポーティングデータベースにイベントを転送するためのデータ同期ポリシーを作成する必要があります。

VI Identity Manager のアップグレード

このセクションでは、Identity Manager のコンポーネントのアップグレードについて説明します。

- 205 ページの第 14 章「Identity Manager のアップグレードの準備」
- 215 ページの第 15 章「Identity Manager コンポーネントのアップグレード」
- 247 ページの第 16 章「Advanced Edition から Standard Edition への切り替え」

14 Identity Manager のアップグレードの準備

このセクションでは、Identity Manager ソリューションを最新バージョンにアップグレードする準備について説明します。

警告 : Identity Manager 4.9 とともにインストールされるコンポーネントを更新するには、常に Identity Manager パッチのダウンロード Web サイトを利用する必要があります。そうしないと、通常の Identity Manager パッチの更新中に重大な競合が発生する可能性があります。

- 205 ページの 「Identity Manager のアップグレードのチェックリスト」
- 207 ページの 「アップグレードプロセスの理解」
- 208 ページの 「サポートされているアップグレードパス」
- 212 ページの 「現在の設定のバックアップ」

Identity Manager のアップグレードのチェックリスト

アップグレードを実行するには、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. アップグレードプロセスについて理解します。詳細については、207 ページの「アップグレードプロセスの理解」を参照してください。
❑	2. Identity Manager 4.9 にアップグレードするためのサポートされているアップグレードパスを確認します。サポートされているアップグレードパスについては、208 ページの 「サポートされているアップグレードパス」を参照してください。
❑	3. Identity Manager をアップグレードするインストールキットを用意していることを確認します。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Identity Manager の入手場所」を参照してください。
❑	4. 現在のプロジェクト、ドライバ環境設定、およびデータベースをバックアップします。詳細については、212 ページの 「現在の設定のバックアップ」を参照してください。
❑	5. Designer の最新バージョンをインストールします。
❑	6. Sentinel Log Management for IGA を最新バージョンにアップグレードします。詳細については、239 ページの 「Sentinel Log Management for IGA のアップグレード」を参照してください。

	チェックリストの項目
❑	7. アイデンティティボールト (eDirectory) を 9.2.9 にアップグレードします。詳細については、219 ページの 「アイデンティティボールトのアップグレード」を参照してください。
❑	8. Identity Manager エンジンのインストール先であるサーバに関連付けられているドライバを停止します。詳細については、224 ページの 「ドライバの停止」を参照してください。
❑	9. Identity Manager エンジンをアップグレードします。詳細については、220 ページの「Identity Manager エンジンのアップグレード」を参照してください。 **注 :** Identity Manager エンジンを新しいサーバにマイグレートする場合は、現在の Identity Manager サーバ上のものと同じ eDirectory レプリカを使用できます。詳細については、190 ページの 「Identity Manager エンジンの新しいサーバへのマイグレート」を参照してください。
❑	10. (状況によって実行) Identity Manager エンジンのドライバセットのいずれかのドライバがリモートローダドライバである場合、ドライバごとにリモートローダサーバをアップグレードします。詳細については、222 ページの 「リモートローダのアップグレード」を参照してください。
❑	11. (状況によって実行) 新しいバージョンのパッケージが利用可能な場合は、既存のドライバでパッケージをアップグレードします。詳細については、226 ページの「Identity Manager ドライバのアップグレード」を参照してください。 これは、既存のドライバの新しいパッケージに含まれる機能を使用する場合にのみ必要です。
❑	12. 識別情報アプリケーションをアップグレードします。詳細については、228 ページの 「Identity Applications のアップグレード」を参照してください。
❑	13. Identity Reporting をアップグレードします . 詳細については、238 ページの「Identity Reporting のアップグレード」を参照してください。
❑	14. 識別情報アプリケーションと Identity Manager エンジンの関連ドライバを起動します。詳細については、225 ページの 「ドライバの起動」を参照してください。
❑	15. (状況によって実行) Identity Manager エンジンまたは識別情報アプリケーションを新しいサーバにマイグレートした場合、この新しいサーバをドライバセットに追加します。詳細については、244 ページの 「新しいサーバをドライバセットに追加する」を参照してください。
❑	16. (状況によって実行) カスタムポリシーとルールがある場合は、カスタマイズされている設定を復元します。詳細については、245 ページの 「ドライバへのカスタムポリシーとルールの復元」を参照してください。
❑	17. Analyzer のアップグレード . 詳細については、242 ページの 「Analyzer のアップグレード」を参照してください。
❑	18. アップグレードした Identity Manager ソリューションをアクティベートします。詳細については、『*NetIQ Identity Manager Overview and Planning Guide*』の「Activating Identity Manager」を参照してください。

アップグレードプロセスの理解

既存の Identity Manager インストールの最新バージョンをインストールする場合、通常は**アップグレード**を実行します。ただし、新しいバージョンの Identity Manager が既存のバージョンからの直接アップグレードパスをサポートしていない場合は、まず 4.9 へのアップグレードが可能なバージョンにアップグレードする必要があります。または、新しいマシンへのマイグレーションを実行することもできます。「マイグレーション」は、新しいサーバに Identity Managers をインストールして、既存のデータをこの新しいサーバにマイグレートすることとして定義されています。

アップグレード

- **Identity Manager 4.8 Standard Edition:** Identity Manager 4.9 から Identity Manager 4.9 Standard Edition にアップグレードする場合は、『Quick Start Guide for Installing NetIQ Identity Manager 4.9 Standard Edition』の「Upgrading Identity Manager 4.8 Standard Edition to Identity Manager 4.8 Standard Edition」セクションに示されている手順を実行します。
- **Identity Manager 4.8 Advanced Edition:** 現在 Identity Manager 4.8 Advanced Edition をインストールしている場合は、それを Identity Manager 4.9 Advanced Edition に直接アップグレードできます。詳細については、205 ページの 「Identity Manager のアップグレードのチェックリスト」を参照してください。

マイグレーション

直接アップグレードを実行できない場合もあります。このようなシナリオでは、マイグレーションが推奨されます。たとえば、今後サポートされないオペレーティングシステムを実行しているサーバ上に Identity Manager をインストールしていた場合、アップグレードではなく、マイグレーションを実行する必要があります。

1 つのドライバセットに複数のサーバを関連付けている場合、一度に 1 台のサーバに対してアップグレードまたはマイグレーションを実行することができます。すべてのサーバを同時にアップグレードできない場合、各サーバのアップグレードが完了するまでは、ドライバは複数のバージョンの Identity Manager と連携して動作します。

重要 : Identity Manager 4.9 以降でのみサポートされているドライバの機能を有効にした場合、バージョンが混在しているサーバでは、ドライバは動作を停止します。古いエンジンは新しい機能を扱うことができません。そのため、すべてのサーバを Identity Manager 4.9 以降にアップグレードするまでは、ドライバは動作しません。

Advanced Edition から Standard Edition への切り替え

Identity Manager では、製品評価期間または Advanced Edition のアクティベーション後に、Advanced Edition から Standard Edition に切り替えることができます。

重要 : Advanced Edition のアクティベーションをすでに適用している場合は、すべての Standard Edition 機能が Advanced Edition で使用可能であるため、Standard Edition に移行する必要はありません。ご使用の環境で Advanced Edition 機能が必要なく、Identity Manager の展開をスケールダウンする場合にのみ Standard Edition に切り替える必要があります。詳細については、247 ページの第 16 章「Advanced Edition から Standard Edition への切り替え」を参照してください。

サポートされているアップグレードパス

Identity Manager 4.9 は、4.7.5 および 4.8.x バージョンからのアップグレードをサポートします。アップグレードを開始する前に、現在のバージョンに対応するリリースノートの情報を確認することをお勧めします。

- 208 ページの 「Identity Manager 4.7.5 バージョンからのアップグレード」
- 210 ページの 「Identity Manager 4.8.x バージョンからのアップグレード」

注 : Identity Manager 4.9 は eDirectory バージョン 9.2.8 および 9.2.9 とのみ互換性があります。必ず、先に eDirectory を 9.2.8 または 9.2.9 にアップグレードした後で、Identity Manager エンジンをアップグレードしてください。

Identity Manager 4.7.5 バージョンからのアップグレード

次の表に、Identity Manager 4.7.5 バージョンのコンポーネントごとのアップグレードパスを一覧表示します。

コンポーネント	ベースバージョン	アップグレード済みのバージョン
Identity Manager エンジン	4.7.5	1. オペレーティングシステムを、サポートされるバージョンにアップグレードします。 2. アイデンティティボールトを 9.2.9 にアップグレードします。 3. Identity Manager エンジンを 4.9 にアップグレードします。
リモートローダ / ファンアウトエージェント	4.7.5	4.7 リモートローダ / 展開エージェントをインストールします。
Designer		Designer 4.9 をインストールします。
Analyzer		Analyzer 4.8.5 をインストールします。

コンポーネント	ベースバージョン	アップグレード済みのバージョン
Identity Applications	4.7.5	Identity Applications をアップグレードする前に、アイデンティティボールトおよび Identity Manager エンジンがそれぞれ 9.2.9 および 4.9 にアップグレードされていることを確認します。 1. オペレーティングシステムを、サポートされるバージョンにアップグレードします。 2. データベースを、サポートされるバージョンにアップグレードします。サポートされているデータベースのバージョンについては、NetIQ Identity Manager 技術情報 Web サイトを参照してください。 3. (状況によって実行) SSPR が別のサーバにインストールされている場合は、コンポーネントを 4.9 バージョンにアップグレードします。 4. ユーザアプリケーションドライバ、役割およびリソースドライバ、データ収集サービスドライバのパッケージを更新します。パッケージの更新後、ドライバを再起動します。 5. Identity Applications を 4.9 にアップグレードします。 6. Tomcat および NGINX サービスを起動します。

コンポーネント	ベースバージョン	アップグレード済みのバージョン
Identity Reporting	4.7.5	1. オペレーティングシステムを、サポートされるバージョンにアップグレードします。 2. データベースを、サポートされるバージョンにアップグレードします。サポートされているデータベースのバージョンの詳細については、Identity Manager 4.9 のシステム要件を参照してください。 3. SLM for IGA を、サポートされるバージョンにアップグレードします。 4. Identity Reporting を 7.2.0 にアップグレードします。 5. (状況によって実行) [Identity Manager データ収集サービス] ページからデータ同期ポリシーを作成します。

アップグレードを開始する前に、NetIQ ドキュメント (https://www.netiq.com/documentation/identity-manager-49/) ページで現在のバージョンに対応するリリースノートの情報を確認することをお勧めします。

Identity Manager 4.8.x バージョンからのアップグレード

次の表に、Identity Manager 4.8.x バージョンのコンポーネントごとのアップグレードパスを一覧表示します。

コンポーネント	ベースバージョン	アップグレード済みのバージョン
Identity Manager エンジン	4.8.x	1. オペレーティングシステムを、サポートされるバージョンにアップグレードします。 2. アイデンティティボールトを 9.2.9 にアップグレードします。 3. Identity Manager エンジンを 4.9 にアップグレードします。
リモートローダ / ファンアウトエージェント	4.8.x	4.7 リモートローダ / 展開エージェントをインストールします。
Designer		Designer 4.9 をインストールします。
Analyzer		Analyzer 4.8.5 をインストールします。

コンポーネント	ベースバージョン	アップグレード済みのバージョン
Identity Applications	4.8.x	Identity Applications をアップグレードする前に、アイデンティティボールトおよび Identity Manager エンジンがそれぞれ 9.2.9 および 4.9 にアップグレードされていることを確認します。 1. オペレーティングシステムを、サポートされるバージョンにアップグレードします。 2. データベースを、サポートされるバージョンにアップグレードします。サポートされているデータベースのバージョンについては、NetIQ Identity Manager 技術情報 Web サイトを参照してください。 3. (状況によって実行) SSPR が別のサーバにインストールされている場合は、コンポーネントを 4.9 バージョンにアップグレードします。 4. ユーザアプリケーションドライバ、役割およびリソースドライバ、データ収集サービスドライバのパッケージを更新します。パッケージの更新後、ドライバを再起動します。 5. Identity Applications を 4.9 にアップグレードします。 6. Tomcat および NGINX サービスを起動します。
Identity Reporting	4.8.x	1. オペレーティングシステムを、サポートされるバージョンにアップグレードします。 2. データベースを、サポートされるバージョンにアップグレードします。サポートされているデータベースのバージョンの詳細については、Identity Manager 4.9 のシステム要件を参照してください。 3. SLM for IGA を、サポートされるバージョンにアップグレードします。 4. Identity Reporting を 7.2.0 にアップグレードします。

アップグレードを開始する前に、NetIQ ドキュメント (https://www.netiq.com/documentation/identity-manager-49/) ページで現在のバージョンに対応するリリースノートの情報を確認することをお勧めします。

現在の設定のバックアップ

アップグレードを実行する前に、Identity Manager ソリューションの現在の設定をバックアップすることをお勧めします。ユーザアプリケーションをバックアップする必要はありません。すべてのユーザアプリケーションの環境設定は、ユーザアプリケーションドライバに保存されます。バックアップは次の方法で作成できます。

- 212 ページの 「Designer のプロジェクトのエクスポート」
- 213 ページの 「ドライバ環境設定のエクスポート」

Designer のプロジェクトのエクスポート

Designer のプロジェクトには、スキーマおよびすべてのドライバ構成情報が含まれています。Identity Manager ソリューションのプロジェクトを作成すると、すべてのドライバを 1 ステップでエクスポートでき、ドライバごとに個別のエクスポートファイルを作成する必要はありません。

- 212 ページの 「現在のプロジェクトのエクスポート」
- 213 ページの 「識別ボールトからプロジェクトを新規作成する」

現在のプロジェクトのエクスポート

すでに Designer プロジェクトがある場合には、以下の方法で、プロジェクト内の情報が識別ボールトの内容と同期されているかどうか確認してください。

1. Designer で、プロジェクトを開きます。
2. モデラーで、[識別ボールト] アイコンを右クリックして、[ライブ] > [比較] の順に選択します。
3. プロジェクトを評価し、相違点があれば一致させて、[OK] をクリックします。

   詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「展開時の比較機能の使用」を参照してください。
4. ツールバーで、[プロジェクト] > [エクスポート] を選択します。
5. [すべて選択] をクリックして、すべてのリソースをエクスポートするように選択します。
6. プロジェクトを保存する場所と、そのフォーマットを選択し、[完了] をクリックします。

   プロジェクトは、現在のワークスペースの場所を除き、任意の場所に保存できます。Designer にアップグレードする場合には、ワークスペースロケーションを新規作成する必要があります。詳細については、『NetIQ Designer for Identity Manager Administration Guide』の「プロジェクトのエクスポート」を参照してください。

識別ボールトからプロジェクトを新規作成する

現在の Identity Manager ソリューションの Designer プロジェクトがない場合は、現在のソリューションをバックアップするためにプロジェクトを作成する必要があります。

1. Designer をインストールします。
2. Designer を起動して、ワークスペースの場所を指定します。
3. オンラインのアップデートをチェックするかどうかを指定して、[OK] をクリックします。
4. ［ようこそ］ページで、[Designer の実行] をクリックします。
5. ツールバーで、[プロジェクト] > ［プロジェクトのインポート］ > ［識別ボールト] を選択します。
6. プロジェクトの名前を指定します。それから、プロジェクトのデフォルトの場所を使用するか、または別の場所を選択します。
7. [次へ] をクリックします。
8. 識別ボールトに接続するために次の値を指定します。
   - ［**ホスト名**］：アイデンティティボールトサーバの IP アドレスまたは DNS 名
   - ［**ユーザ名**］：アイデンティティボールトへの認証に使用するユーザの DN
   - ［**パスワード**］：認証ユーザのパスワード
9. [次へ] をクリックします。
10. ［識別ボールトのスキーマ］と［デフォルトの通知コレクション］は選択したままにします。
11. ［デフォルト通知コレクション］を展開し、必要のない言語を選択解除します。

    デフォルトの通知コレクションは、多くの言語に翻訳されています。すべての言語をインポートすることもできますし、使用する言語だけを選択することもできます。
12. [参照] をクリックして、インポートするドライバセットを参照し、選択します。
13. この識別ボールトのドライバセットごとにステップ 12 を繰り返し、[完了] をクリックします。
14. プロジェクトがインポートされたら、[OK] をクリックします。
15. 識別ボールトが 1 つだけの場合には、これで完了です。複数の識別ボールトがある場合には、ステップ 16 に進みます。
16. ツールバーの [ライブ] > ［インポート] をクリックします。
17. 追加の識別ボールトごとに、ステップ 8 からステップ 14 を繰り返します。

ドライバ環境設定のエクスポート

ドライバのエクスポートを作成すると、現在の環境設定のバックアップが取られます。ただし、Designer は現在のところ、役割ベースのエンタイトルメントドライバとポリシーのバックアップは作成しません。

Designer によるドライバ環境設定のエクスポート

1. Designer のプロジェクトで最新バージョンのドライバが使用されていることを確認します。詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「Importing a Library, a Driver Set, or a Driver from the Identity Vault」を参照してください。
2. ［モデラー］で、アップグレードするドライバの行を右クリックします。
3. **[環境設定ファイルのエクスポート]** を選択します。
4. 環境設定ファイルを保存する場所を参照して、**[保存]** をクリックします。
5. ［結果］ページで **[OK]** をクリックします。
6. 各ドライバに対して、ステップ 1 ～ステップ 5 を繰り返します。

15 Identity Manager コンポーネントのアップグレード

このセクションでは、Identity Manager の個々のコンポーネントをアップグレードする方法の詳細について説明します。アップグレード後に実行する必要がある可能性がある手順についても説明します。

- 215 ページの 「アップグレード順序」
- 216 ページの 「アップグレードに関する考慮事項」
- 217 ページの 「Designer のアップグレード」
- 218 ページの 「Identity Manager エンジンのアップグレード」
- 224 ページの 「Identity Manager ドライバの停止と起動」
- 226 ページの 「Identity Manager ドライバのアップグレード」
- 228 ページの 「Identity Applications のアップグレード」
- 238 ページの 「Identity Reporting のアップグレード」
- 242 ページの 「Analyzer のアップグレード」
- 242 ページの 「アップグレード後のタスク」
- 244 ページの 「新しいサーバをドライバセットに追加する」
- 245 ページの 「ドライバへのカスタムポリシーとルールの復元」

アップグレード順序

一度に 1 つの Identity Manager コンポーネントのみをアップグレードする必要があります。以下の順序でコンポーネントをアップグレードします。

1. Designer
2. Sentinel Log Management for IGA
3. 識別ボールト
4. Identity Manager エンジン
5. リモートローダ
6. ファンアウトエージェント
7. Identity Applications (Advanced Edition 用)
8. Identity Reporting (Standard Edition の場合 OSP もインストールします)
9. Analyzer

10. (状況によって実行) SSPR
11. Identity Console

注 :

- 必ず、Analyzer を 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 で使用してください。
- SSPR がリモートサーバ上にインストールされている場合は、SSPR のスタンドアロンアップデートが必要です。
- すべての Identity Manager コンポーネントを同じサーバ上で実行する場合は、必ずすべてのコンポーネントを最新バージョンに更新してください。

アップグレードに関する考慮事項

Identity Manager コンポーネントのアップグレードを開始する前に、次の考慮事項を確認してください。

- このコンポーネントは、コンソールモードでのみアップグレードできます。サイレントアップグレードはサポートされていません。
- 一度に 1 つのコンポーネントをアップグレードする必要があります。
- Identity Vault は別個にアップグレードする必要があります。このバージョンは、eDirectory 9.2.8 以降をサポートしています。
- Identity Vault が BTRFS ファイルシステムで設定されている場合、アップグレードプロセスは成功しません。このバージョンは、Ext3、Ext4、および XFS ファイルシステムのみをサポートしています。
- Identity Manager エンジンをアップグレードする前に、キャッシュファイルにイベントがないことを確認します。ドライバが MapDB を使用している場合、アップグレードされたドライバがアップグレードされたエンジンで正しく機能することを確認してください。詳細については、220 ページの 「MapDB 3.0.9 の使用」を参照してください。
- コンポーネントをアップグレードする前に、推奨されたサーバセットアップを確認する必要があります。Identity Manager では、Identity Applications と OSP を同じコンピュータにインストールすることが要求されます。ただし、Identity Reporting はローカルまたはリモートでインストールされた OSP をサポートします。
- Identity Applications、Identity Reporting、およびデータベースが、一律に FQDN で設定されていることを確認します。
- サポートされているバージョンの Oracle または Microsoft SQL Server データベースを使用している場合は、Identity Applications が正しく機能するように、Identity Applications (`idmuserappdb`) データベースおよびワークフロー (`igaworkflowdb`) データベースの 2 つのデータベースインスタンスを設定する必要があります。同じサーバでデータベースインスタンスを設定してください。
- Identity Applications で NAM リバースプロキシ設定を使用してフォームをロードする場合は、NAM 5.x バージョンを使用する必要があります。

- Identity Manager 4.9 以降、NetIQ SecretStore のサポートは廃止されました。最新の UAD パッケージにアップグレードして、Tomcat を再起動すると、NetIQ SecretStore リポジトリにある電子メールアカウントの既存のパスワードはすべて `notfSMTPMailPassword` 属性に自動的に移行されます。このパスワード移行プロセスは、`ism-configuration.properties` ファイルで利用可能な、`com.netiq.idm.migrate.secretstore` プロパティで制御されます。デフォルトでは、このプロパティは `true` に設定されています。移行が完了したら、プロパティは自動的に false に戻され、この状態を維持します。
- Self Service Password Reset (SSPR) には次の考慮事項が適用されます。
  - Syslog 出力形式タイプが CEF の SSPR サーバで監査が有効な場合、Sentinel Syslog サーバから NetIQ Self Service Password Reset コレクタをアンインストールする必要があります。アンインストールしない場合、Syslog サーバが SSPR 監査イベントを解析できません。
  - SSPR はイベントの監査用に CEF と JSON 両方の出力形式タイプをサポートします。SSPR 4.7.0.1 では引き続き JSON 出力形式タイプの NetIQ Self Service Password Reset コレクタをサポートします。単一の Sentinel Syslog サーバに接続されている複数の SSPR サーバがある場合は、すべてのサーバ間でイベントを監査するために 1 つの形式タイプのみを選択する必要があります。
  - SSPR が別個のサーバにインストールされている場合は、そのコンポーネントを Identity Manager 4.9 にバンドルされている最新バージョンにアップグレードします。

  Identity Manager を 4.9 バージョンに更新すると、SSPR は CEF 形式タイプで監査イベントを収集するため、Universal CEF Collector を必要とする 4.7.0.1 バージョンにアップグレードされます。

  注 : 初めて CEF 出力形式タイプで SSPR 監査を有効にする場合は、NetIQ Self Service Password Reset コレクタが Sentinel Syslog サーバで設定されていないことを確認します。

Designer のアップグレード

注 : このリリースでは、以前のバージョンからの Designer のアップグレードをサポートしていません。したがって、以前のバージョンがインストールされている場合は、以前のバージョンをアンインストールしてから Designer 4.9 をインストールする必要があります。また、組み込みの自動更新機能とオフライン更新機能を使用して Designer 4.9 にアップグレードすることもできません。

1. Designer がインストールされているサーバに管理者としてログインします。
2. プロジェクトのバックアップコピーを作成するために、プロジェクトをエクスポートします。

   エクスポートの詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「プロジェクトのエクスポート」を参照してください。

3 次の手順で現在の Designer バージョンをアンインストールします。

3a Designer を閉じます。

3b アンインストールスクリプトを含むディレクトリに移動します。このスクリプトは、デフォルトでは `<installation_directory>/designer/UninstallDesigner/Uninstall Designer for Identity Manager` にあります。

3c スクリプトを実行するには、「`./uninstall`」と入力します。

4 Designer は、GUI またはコンソールモードでインストールできます。

4a Software License and Download (ソフトウェアライセンスおよびダウンロード) (https://sld.microfocus.com/index) ポータルから、`Identity_Manager_4.9.0_Designer_Linux.tar.gz` をダウンロードします。

4b ファイルの抽出先のディレクトリに移動します。

4c 次のコマンドを実行します。

```
tar -zxvf Identity_Manager_4.9.0_Designer_Linux.tar.gz
```

4d 次のいずれかのコマンドを実行して、Designer をインストールします。

コンソール : `./install`

GUI: `./install -i console`

4e プロンプトに従ってインストールを続行します。

5 Designer 4.9 のインストール後、古いバージョンから Designer プロジェクトをすべてインポートする必要があります。インポートプロセスを開始すると、Designer はプロジェクトコンバータウィザードを実行します。このウィザードは、古いバージョンで作成したプロジェクトを最新バージョンに変換します。ウィザードで **[プロジェクトをワークスペースにコピーする]** を選択します。プロジェクトコンバータの詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「Converting Earlier Projects」を参照してください。

Identity Manager エンジンのアップグレード

Identity Manager エンジンをアップグレードする前に、アイデンティティボールトをアップグレードしてください。Identity Manager エンジンアップグレードプロセスは、ホストコンピュータのファイルシステムに保存されているドライバシムファイルをアップデートします。

インストール前の Linux RPM の検証

RPM を検証するには、次の手順を実行します。

1 [Software and License (ソフトウェアおよびライセンス)] ページ (https://sld.microfocus.com/) から `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` をマウントします。

3 公開鍵に使用する次の場所に移動します。

```
<ISO-MOUNT-location>/common/license/
```

たとえば、「cd /mnt/common/license/」と入力します。

4 次のコマンドを実行して鍵をインポートします。

```
rpm --import MicroFocusGPGPackageSign.pub
```

```
rpm --import OT-package-sign.pub
```

5 (オプション) 次のコマンドを実行して署名を検証します。

```
rpm --checksig -v <mount-location>/IDVault/setup/<RPM Name>
```

次に例を示します。

```
rpm --checksig -v /mnt/IDVault/setup/novell-NDSbase-*.x86_64.rpm
```

アイデンティティボールトのアップグレード

1 『NetIQ Identity Manager Overview and Planning Guide』の「Identity Manager の入手場所」の指示に従って、*Identity_Manager_4.9.0_Linux.iso* をダウンロードします。

2 ダウンロード済みの .iso をマウントします。

3 .iso ファイルのルートディレクトリから、IDVault/setup ディレクトリに移動します。

4 次のコマンドを実行します。

```
./nds-install
```

5 iManager は非推奨化されるため、次の警告プロンプトで応答を入力します。

```
This version of eDirectory installation will upgrade various packages
required by eDirectory and iManager. The updated packages may impact
iManager running on the same server. Do you want to continue
installation/upgrade (y/n)
```

5a (オプション)「はい (y)」と応答した場合は、手順 6 に進みます。

5b (オプション)「いいえ (n)」と応答した場合、更新は終了します。

6 使用許諾契約に同意し、インストールを続行します。

7 **[adminDN]** を指定します。たとえば、cn=admin.ou=sa.o=system です。

8 eDirectory インスタンスを停止して NICI をアップグレードするかどうかを求められたら、「y」と指定します。

9 **[Enhanced Background Authentication (拡張バックグラウンド認証)]** を設定するかどうかを指定します。

注 : DIB アップグレードが失敗し、nds-install コマンドプロンプトで、nds-install の後に ndsconfig アップグレードを実行するように指示された場合は、その指示に従って実行します。アップグレード後に eDirectory サービスが開始されない場合は、ndsconfig アップグレードコマンドを実行します。詳細については、『NetIQ eDirectory Installation Guide (NetIQ eDirectory インストールガイド)』を参照してください。

Identity Manager エンジンのアップグレード

ドライバを停止していることを確認します。詳細については、224 ページの 「ドライバの停止」を参照してください。

Identity Manager エンジンをアップグレードするには、次の手順を実行します。

1. NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。
2. ダウンロード済みの `.iso` をマウントします。
3. 次のコマンドを実行します。

   ```
   ./install.sh
   ```

4. 使用許諾契約を読みます。
5. 使用許諾契約の条項を確認し、「`y`」と入力します。
6. Identity Manager コンポーネントをアップグレードするかどうかを指定します。使用可能なオプションは、**[y]** および **[n]** です。
7. Identity Manager エンジンを選択します。
8. 次の詳細を指定します。

   識別ボールト管理者 : アイデンティティボールト管理者名を指定します。

   識別ボールト管理者パスワード : アイデンティティボールト管理者パスワードを指定します。

エンジンのアップグレードプロセスでは、Identity Vault の DIB ディレクトリにある既存の MapDB キャッシュファイル (`dx*`) の一部が保持されます。ドライバのアップグレード後に、MapDB を使用するドライバのこれらのファイルを手動で削除する必要があります。詳細については、220 ページの 「MapDB 3.0.9 の使用」を参照してください。

MapDB 3.0.9 の使用

Identity Manager は MapDB 3.0.9 をサポートしています。MapDB は、Identity Manager エンジン以外に Office365 や Azure Active Directory などの複数のドライバで使用されます。

これらのドライバのいずれかを使用している場合は、ドライバをアップグレードする前に次のセクションを確認する必要があります。

- 221 ページの「ドライババージョンに関する Identity Manager 4.9 エンジンサポートの理解」
- 221 ページの 「(オプション) MapDB キャッシュファイルの手動削除」

ドライババージョンに関する Identity Manager 4.9 エンジンサポートの理解

MapDB を使用する Identity Manager ドライバをアップグレードする前に、以下の考慮事項を確認してください。

- Identity Manager 4.9 に付属のドライバは、Identity Manager 4.9 エンジンまたはリモートローダと互換性があります。特定のドライバ実装ガイドのドライバアップグレード手順に従う必要があります。

(オプション) MapDB キャッシュファイルの手動削除

Identity Manager エンジンのアップグレードプロセスでは、既存の MapDB キャッシュファイル (`dx*`) の一部が Identity Vault の DIB ディレクトリ (`/var/opt/novell/eDirectory/data/dib`) に残ります。ドライバのアップグレード後に、ドライバのこれらのファイルを手動で削除する必要があります。このアクションにより、ドライバが Identity Manager 4.9 エンジンで正しく動作することが保証されます。

次の表に、削除する必要がある MapDB キャッシュファイルを示します。

Identity Manager ドライバ	削除する MapDB 状態キャッシュファイル
データ収集サービス	`DCSDriver_<driver instance guid>-*` `<driver instance guid>-*`
JDBC	`jdbc_<driver instance guid>_*`
LDAP	`ldap_<driver instance guid>*`
Managed System Gateway	`MSGW-<driver-instance-guid>.*`
Office 365 および Azure Active Directory	`<Azure driver name>_obj.db.*`
Salesforce	`<Salesforce driver name>.*` `<Salesforce driver name>`

ここで、* は、MapDB 状態キャッシュファイルのバージョンを表します。バージョンに関係なく、先に示した名前を含む MapDB 状態キャッシュファイルはすべて削除する必要があります。Salesforce ドライバの場合、MapDB 状態キャッシュファイルもドライバ名で表されます。これらのファイルの例を以下に示します。

- `DCSDriver_<driver instance guid>-0.t`, `<driver instance guid>-1.p`
- `jdbc_<driver instance guid>_0.t`, `jdbc_<driver instance guid>_0`, `jdbc_<driver instance guid>_1`
- `ldap_<driver instance guid>b`, `ldap_<driver instance guid>b.p`
- `MSGW-<driver instance guid>.p`, `MSGW-<driver instance guid>.t`

- <Azure driver name>_obj.db.t, <Azure driver name>_obj.db.p
- <Salesforce driver name>.p, <Salesforce driver name>.t,Salesforce driver1

root 以外のユーザによる Identity Manager エンジンのアップグレード

このアクションは、Identity Manager エンジンを root 以外のユーザとしてインストールした場合にのみ実行してください。

1. [Software and License (ソフトウェアおよびライセンス)] ページ (https://sld.microfocus.com/) から Identity_Manager_4.9.0_Linux.iso をダウンロードします。
2. ダウンロード済みの .iso をマウントします。
3. 次のコマンドを実行します。

   ```
   ./install.sh
   ```

4. [Identity Manager エンジン] を選択し、Enter を押します。
5. Identity Manager エンジンの root 以外のインストール場所を指定します。たとえば、/home/user/eDirectory です。
6. 「y」を指定して、アップグレードを完了します。

リモートローダのアップグレード

リモートローダを実行している場合は、リモートローダファイルをアップグレードする必要があります。

1. リモートローダ環境設定ファイルのバックアップを作成します。
2. ドライバを停止していることを確認します。詳細については、224 ページの 「ドライバの停止」を参照してください。
3. 各ドライバのリモートローダサービスまたはデーモンを停止します。

   ```
   rdxml -config path_to_configfile -u
   ```

4. NetIQ ダウンロード Web サイトから Identity_Manager_4.9.0_Linux.iso をダウンロードします。
5. ダウンロード済みの .iso をマウントします。
6. 次のコマンドを実行します。

   ```
   ./install.sh
   ```

7. 使用許諾契約を読みます。
8. 使用許諾契約の条項を確認し、「y」と入力します。
9. Identity Manager コンポーネントをアップグレードするかどうかを指定します。使用可能なオプションは、[y] および [n] です。
10. リモートローダを選択します。

11 インストールが完了したら、環境設定ファイルに現在の環境の情報が含まれていることを確認します。

12 (状況によって実行) 環境設定ファイルに問題がある場合は、手順 1 で作成したバックアップファイルをコピーします。問題がない場合は、次の手順に進んでください。

13 各ドライバのリモートローダサービスまたはデーモンを起動します。

```
rdxml -config path_to_config_file
```

重要 : ドライバが MapDB を使用する場合、ドライバをアップグレードした後、ドライバの既存の MapDB 状態キャッシュファイルを手動で削除します。Identity Manager エンジンのアップグレードプロセスでは、これらのファイルのすべてが Identity Vault の DIB ディレクトリから削除されないため、この操作が必要になります。詳細については、220 ページの「MapDB 3.0.9 の使用」を参照してください。

Java リモートローダのアップグレード

1 Java リモートローダ環境設定ファイルのバックアップを作成します。

2 ドライバを停止していることを確認します。詳細については、224 ページの 「ドライバの停止」を参照してください。

3 各ドライバのリモートローダサービスまたはデーモンを停止します。

```
dirxml_jremote -config path_to_configfile -u
```

4 NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

5 ダウンロード済みの `.iso` をマウントします。

6 `/IDM/packages/java_remoteloader` ディレクトリに移動します。

7 既存の Java リモートローダインストール済みディレクトリにある `dirxml_jremote_dev.tar.gz` ファイルをコピーして置き換えます。

8 既存のセットアップに存在するファイルに基づいて、既存の Java リモートローダインストール済みディレクトリにある次のファイルのいずれかをコピーして置き換えます。

- `dirxml_jremote.tar.gz`
- `dirxml_jremote_mvs.tar`

9 手順 7 および手順 8 でコピーしたファイルを展開します。

たとえば、`tar -zxvf dirxml_jremote.tar.gz` です。

10 (状況によって実行) 環境設定ファイルに問題がある場合は、手順 1 で作成したバックアップファイルをコピーします。問題がない場合は、次の手順に進んでください。

注 : `version.txt` ファイルを使用して、最新バージョンの Java リモートローダがあることを確認してください。

---

11 各ドライバのリモートローダサービスまたはデーモンを起動します。

```
dirxml_jremote -config path_to_config_file
```

Identity Manager ドライバの停止と起動

アップグレード、マイグレーション、またはインストールプロセスで正しいファイルを変更または置換できるようにするには、Identity Manager ドライバを起動または停止しなければならない場合があります。このセクションでは、次の操作について説明します。

- 224 ページの 「ドライバの停止」
- 225 ページの 「ドライバの起動」

ドライバの停止

ドライバのファイルを変更する場合、あらかじめドライバを停止しておくことが重要です。

- 224 ページの 「Designer を使用したドライバの停止」
- 224 ページの 「Identity Console を使用したドライバの停止」

Designer を使用したドライバの停止

1 Designer で、識別ボールトのオブジェクトを [アウトライン] タブで選択します。

2 モデラーツールバーで、[すべてのドライバを停止] アイコンをクリックします。

これにより、プロジェクトの一部であるすべてのドライバが停止します。

3 ドライバを手動開始に設定すると、アップグレードプロセスを完了しない限りドライバは開始されません。

3a [アウトライン] タブの［ドライバ］アイコンをダブルクリックします。

3b [ドライバ環境設定］ >［起動時のオプション] の順に選択します。

3c [手動] を選択し、[OK] をクリックします。

3d 各ドライバに対して、ステップ 3a ～ステップ 3c を繰り返します。

Identity Console を使用したドライバの停止

1 Identity Console で、[Identity Manager Administration (Identity Manager の管理)] を選択します。

2 ドライバセットオブジェクトで [Action menu (アクションメニュー)] をクリックします。

3 [すべてのドライバを停止] をクリックします。

4 各ドライバセットオブジェクトに対して、手順 3 を繰り返します。

5 ドライバを手動開始に設定すると、アップグレードプロセスを完了しない限りドライバは開始されません。

5a Identity Console で、[Identity Manager Administration (Identity Manager の管理)] を選択します。

5b ドライバセットオブジェクトをクリックします。

5c ドライバアイコンの右上隅で、[Action menu (アクションメニュー)] をクリックし、[起動時のオプション] を選択します。

5d [手動] を選択し、[OK] をクリックします。

5e ツリーの各ドライバに対して、ステップ 5a ～ステップ 5d を繰り返します。

ドライバの起動

Identity Manager コンポーネントがすべてアップデートされたら、ドライバを再起動します。ドライバが実行状態になったら、ドライバをテストして、すべてのポリシーが依然と同様に機能していることを確認することをお勧めします。

- 225 ページの 「Designer を使用したドライバの起動」
- 225 ページの 「Identity Console を使用したドライバの起動」

Designer を使用したドライバの起動

1 Designer で、識別ボールトのオブジェクトを [アウトライン] タブで選択します。

2 モデラーツールバーの [すべてのドライバを起動] アイコン をクリックします。これにより、プロジェクト内のすべてのドライバが起動されます。

3 ドライバ起動オプションを設定します。

3a [アウトライン] タブの［ドライバ］アイコン をダブルクリックします。

3b [ドライバ環境設定］ >［起動時のオプション] を選択します。

3c [自動開始] を選択するか、ドライバの起動方法を選択し、[OK] をクリックします。

3d 各ドライバに対して、ステップ 3a ～ステップ 3c を繰り返します。

4 ドライバをテストして、ポリシーが設計どおりに機能していることを確認します。ポリシーのテスト方法の詳細については、『*NetIQ Identity Manager - Using Designer to Create Policies*』の「ポリシーシミュレータを使用したポリシーのテスト」を参照してください。

Identity Console を使用したドライバの起動

1 Identity Console で、[Identity Manager の概要] を選択します。

2 ドライバセットオブジェクトで [Action menu (アクションメニュー)] アイコン をクリックします。

3 [すべてのドライバを起動] をクリックします。

4 複数のドライバがある場合は、各ドライバセットオブジェクトに対して手順 3 を繰り返します。

5 ドライバ起動オプションを設定します。

5a Identity Console で、[Identity Manager Administration (Identity Manager の管理)] を選択します。

5b ドライバセットオブジェクトをクリックします。

5c ドライバアイコンの右上隅で、[Action menu (アクションメニュー)] をクリックし、[起動時のオプション] を選択します。

5d [手動] を選択し、[OK] をクリックします。

5e 各ドライバに対して、ステップ 5c ～ステップ 5d を繰り返します。

6 ドライバをテストして、ポリシーが設計どおりに機能していることを確認します。

Identity Manager ドライバのアップグレード

新しいドライバコンテンツは、「パッケージ」で配信されます。パッケージの管理、保守、および作成には、Designer を使用します。Identity Console で行ったドライバの内容への変更は、Designer では保持されません。パッケージ管理の詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「Understanding Packages」を参照してください。

ドライバをパッケージにアップグレードするには、次の手順を実行します。

- 226 ページの 「新しいドライバの作成」
- 226 ページの 「既存のコンテンツをパッケージのコンテンツと交換」
- 227 ページの 「現在のコンテンツを維持しつつパッケージを使用する新しいコンテンツを追加」

新しいドライバの作成

ドライバをパッケージにアップグレードする最も簡単な方法は、既存のドライバを削除し、パッケージを使用して新しいドライバを作成する方法です。新しいドライバに必要なすべての機能を追加します。手順はドライバごとに異なります。手順については、Identity Manager ドライバマニュアルの Web サイト (https://www.netiq.com/documentation/identity-manager-49-drivers/) で個別のドライバガイドを参照してください。ドライバは以前と同様に機能しますが、ドライバの環境設定ファイルのコンテンツの代わりにパッケージのコンテンツを使用するようになります。

既存のコンテンツをパッケージのコンテンツと交換

ドライバによって作成された関連付けを維持する必要がある場合、ドライバを削除して再作成する必要はありません。関連付けを維持したまま、ドライバのコンテンツをパッケージで置き換えることができます。

パッケージからのコンテンツで既存のコンテンツを置き換えるには

1 ドライバおよびドライバに含まれるカスタマイズされたすべてのコンテンツのバックアップを作成します。

方法については、213 ページの 「ドライバ環境設定のエクスポート」を参照してください。

2 Designer で、ドライバ内に保存されているすべてのオブジェクトを削除します。ドライバ内部に保存されているポリシー、フィルタ、エンタイトルメント、および他のすべての項目を削除します。

注 : Designer には、最新のパッケージをインポートする自動インポート機能があります。ドライバパッケージをパッケージカタログに手動でインポートする必要はありません。

詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「パッケージカタログへのカタログのインポート」を参照してください。

---

3 最新のパッケージをドライバにインストールします。

これらの手順は各ドライバに固有です。手順については、Identity Manager ドライバマニュアルの Web サイト (https://www.netiq.com/documentation/identity-manager-49-drivers/) で個別のドライバガイドを参照してください。

4 カスタムポリシーとルールがある場合はドライバに復元します。方法については、245 ページの 「ドライバへのカスタムポリシーとルールの復元」を参照してください。

現在のコンテンツを維持しつつパッケージを使用する新しいコンテンツを追加

パッケージをインストールする前に、ドライバ環境設定ファイルのバックアップを作成します。パッケージをインストールすると、パッケージが既存のポリシーを上書きし、ドライバが動作を停止する可能性があります。ポリシーが上書きされた場合、バックアップのドライバ環境設定ファイルをインポートして、ポリシーを再作成できます。

開始前に、カスタマイズされたポリシーに、デフォルトのポリシーと異なるポリシー名が付いていることを確認します。ドライバ環境設定が新しいドライバファイルでオーバーレイされると、既存のポリシーは常に上書きされます。一意の名前が付けられていないカスタムポリシーは失われます。

パッケージを使用してドライバに新しいコンテンツを追加するには

1 ドライバおよびドライバに含まれるカスタマイズされたすべてのコンテンツのバックアップを作成します。

方法については、213 ページの 「ドライバ環境設定のエクスポート」を参照してください。

注 : Designer には、最新のパッケージをインポートする自動インポート機能があります。ドライバパッケージをパッケージカタログに手動でインポートする必要はありません。

詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「パッケージカタログへのカタログのインポート」を参照してください。

---

2 ドライバにパッケージをインストールします。

手順については、Identity Manager ドライバマニュアルの Web サイト (https://www.netiq.com/documentation/identity-manager-49-drivers/) で個別のドライバガイドを参照してください。

3 ドライバに必要なパッケージを追加します。これらの手順は各ドライバに固有です。詳細については、Identity Manager ドライバマニュアルの Web サイト (https://www.netiq.com/documentation/identity-manager-49-drivers/) を参照してください。

ドライバには、パッケージによって追加された新しい機能が含まれます。

注 : LDAP ドライバをアップグレードした後に、必ず一時ディレクトリの `ldap_*.zoomdb` フォルダを削除してください。`ldap_*.zoomdb` フォルダを削除する際にデータが失われるのを避けるには、次の手順を実行します。

1. Designer に移動します。
2. **[プロパティ] > [ドライバ環境設定] > [ドライバパラメータ] > [Publisher option (発行者オプション)]** に移動します。
3. **[Search results to synchronize on first startup (初回起動時に同期する検索結果)]** 属性を **[synchronize everything (すべて同期)]** に変更します。

Identity Applications のアップグレード

このセクションでは、次のコンポーネントの更新を含む、Identity Applications およびサポートするソフトウェアのアップグレードについて説明します。

- Identity Manager ユーザアプリケーション
- Self-Service Password Reset (SSPR)
- Tomcat、JDK、および ActiveMQ
- PostgreSQL データベース
- OSP (One SSO Provider)

重要 : Identity Manager 4.9 では、Identity Applications と OSP を同じコンピュータにインストールすることが要求されます。このバージョンにアップグレードする場合は、Identity Applications のアップグレード時にインストールされる OSP を使用し、既存の OSP サーバから新しい OSP サーバに OSP 設定をコピーします。詳細については、234 ページの 「Identity Applications コンポーネントのアップグレード後のタスク」を参照してください。

このセクションでは、次のトピックについて説明します。

- 229 ページの 「アップグレードに関する考慮事項」
- 231 ページの 「システム要件」
- 231 ページの 「アップグレードプログラムについて」

- 231 ページの 「PostgreSQL のアップグレード」
- 231 ページの 「Identity Applications のコンポーネントのアップグレード」
- 234 ページの 「Identity Applications コンポーネントのアップグレード後のタスク」
- 238 ページの 「アップグレード後のバージョン番号の確認」

アップグレードに関する考慮事項

Identity Applications アップグレードプロセスは、Identity Applications コンポーネントのアップグレード方法によって異なる場合があります。Identity Applications をアップグレードする前に、次の考慮事項が適用されます。

- Identity Applications のアップグレードプロセスを開始する前に、Oracle または MS SQL データベースを使用している場合は `igaworkflowdb` を作成し、すべての権限を `idmadmin` ユーザに割り当ててデータベースを所有する必要があります。
- Identity Applications コンポーネントを Identity Manager 4.7.5 から 4.9 にアップグレードする前に、ワークフロー移行ツールを使用して Identity Applications データベース (`idmuserappdb`) からワークフローエンジンデータベース (`igaworkflowdb`) にデータをエクスポートします。詳細については、『NetIQ Identity Manager - Administrator's Guide to the Identity Applications』の「*Migrating Data from Identity Applications Database into Workflow Engine Database*」を参照してください。
- Identity Applications と SSPR が異なるサーバにインストールされている場合は、SSPR を個別にアップグレードすることを選択できます。
- アップグレード中は、データベース名と他のデフォルト値を確認し、必要に応じて変更してください。
- Identity Manager は、Identity Applications サーバでの OSP のローカルインストールをサポートしています。アップグレードプログラムは、このバージョンへの OSP のスタンドアロンアップグレードをサポートせず、Identity Applications のアップグレード中に OSP の新しいコピーをインストールします。既存の OSP 設定を新しくインストールされた OSP に復元するには、234 ページの 「Identity Applications コンポーネントのアップグレード後のタスク」の One SSO Provider を参照してください。

表 15-1 Identity Applications のアップグレードプロセス

Identity Applications の展開	アップグレードプロセス
Identity Applications、SSPR、および OSP は同じサーバにインストールされています。	すべてのコンポーネントをアップグレードするには、232 ページの 「Identity Applications のアップグレード」の手順に従ってください。
Identity Applications と OSP は同じサーバにインストールされています。SSPR は別のサーバにインストールされています。	1. Identity Applications と OSP をアップグレードするには、228 ページの 「Identity Applications のアップグレード」の手順に従ってください。 2. 別のサーバの SSPR をアップグレードするには、234 ページの 「SSPR のアップグレード」の手順に従ってください。
Identity Applications は、SSPR および OSP とは異なるサーバにインストールされています。この場合、SSPR は Identity Applications サーバまたは別のサーバにインストールできます。ただし、OSP は Identity Applications サーバにインストールされている必要があります。	1. Identity Applications と OSP をアップグレードするには、228 ページの 「Identity Applications のアップグレード」の手順に従ってください。 2. 別のサーバの SSPR をアップグレードするには、234 ページの 「SSPR のアップグレード」の手順に従ってください。 3. 設定更新ユーティリティを起動し、OSP がインストールされている新しいサーバの詳細を指定します。この場合、新しいサーバは Identity Applications がインストールされているサーバです。詳細については、134 ページの 「SSO Clients Parameters (SSO クライアントパラメータ)」を参照してください。

- Open Enterprise Server (OES) 23.4 バージョン上の Identity Applications をアップグレードする場合は、必ず、Identity Manager エンジンが使用する JRE に、ユーザアプリケーションへの接続に必要な証明書があることを確認してください。Identity Manager ドライバは、Identity Manager エンジンのキーストアを使用してユーザアプリケーションにアクセスします。役割およびリソースサービスドライバがターゲットの場所で必要な署名を見つけることができない場合、ドライバのトレースにエラーが記録されることがあります。このエラーのトラブルシューティング方法の詳細については、『NetIQ Identity Manager - Administrator's Guide to the Identity Applications』を参照してください。

システム要件

アップグレードプロセスにより、インストールされたコンポーネントの現在の設定のバックアップが作成されます。ご使用のサーバにバックアップを格納するための十分な容量と、アップグレードに使用可能な追加の空き容量があることを確認します。詳細については、NetIQ Identity Manager の技術情報 Web サイト (https://www.microfocus.com/en-us/products/netiq-identity-manager/specs) を参照してください。

アップグレードプログラムについて

アップグレードプロセスは、既存のコンポーネントから設定値を読み込みます。この情報には、`ism-configuration.properties`、`server.xml`、`SSPRConfiguration.xml`、および他の設定ファイルが含まれます。これらの設定ファイルを使用すると、アップグレードプロセスにより、指定されたコンポーネントのアップグレードプログラムが内部的に起動します。アップグレードプログラムは現在のインストールのバックアップも作成します。

PostgreSQL のアップグレード

PostgreSQL をアップグレードするには次の手順を実行します。

1. NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードして展開します。
2. `/common/scripts` ディレクトリに移動します。
3. 次のコマンドを実行します。

   ```
   ./pg-upgrade.sh
   ```

4. 以下の詳細を指定して、インストールを完了します。

   既存の Postgres インストール場所 : PostgreSQL がインストールされる場所を指定します。デフォルトの場所は、`/opt/netiq/idm/postgres` です。

   注 : Postgres ユーザに `/opt/netiq/idm/postgres` ディレクトリに対する適切な許可があることを確認してください。

   既存の Postgres データディレクトリ : PostgreSQL データディレクトリの場所を指定します。デフォルトの場所は、`/opt/netiq/idm/postgres/data` です。

   既存の Postgres データベースパスワード : PostgreSQL パスワードを指定します。

   新しい Postgres データディレクトリ : 新しい PostgreSQL データディレクトリを指定します。たとえば、`/opt/netiq/idm/postgres_new/data` を指定します。

Identity Applications のコンポーネントのアップグレード

- Identity Applications のドライバパッケージのアップグレード
- Identity Applications のアップグレード
- SSPR のアップグレード

Identity Applications のドライバパッケージのアップグレード

Tomcat を停止し、ユーザアプリケーションドライバ、役割およびリソースサービスドライバ、データ収集サービスドライバのパッケージを最新バージョンに更新する必要があります。パッケージを最新バージョンにアップグレードする方法については、『*NetIQ Designer for Identity Manager Administration Guide*』の「Upgrading Installed Packages」を参照してください。

ユーザアプリケーションドライバパッケージをアップグレードした後、ワークフローテンプレートパッケージを手動で追加する必要があります。

1 Designer で、［ユーザアプリケーションドライバ］ > ［プロパティ］に移動します。

2 ［パッケージ］をクリックして、をクリックします。

3 **[Show only applicable package versions (適用可能なパッケージバージョンのみを表示する)]** チェックボックスを選択します。

4 **[ワークフローテンプレートの作成]** を選択します。

5 **[OK]** をクリックしてから、**[完了]** をクリックしてインストールを完了します。

6 ユーザアプリケーションドライバを展開します。

重要 : ユーザアプリケーションドライバのアップグレードの一環として、電子メール通知テンプレートがインストールまたはアップグレードされる場合は、**[デフォルトの通知コレクション]** オブジェクトを展開する必要があります。

Identity Applications のアップグレード

次の手順では、Identity Applications をアップグレードする方法について説明します。

1 NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` をマウントします。

3 次のコマンドを実行します。

```
./install.sh
```

4 使用許諾契約を読みます。

5 使用許諾契約の条項を確認し、「`y`」と入力します。

6 Identity Applications をアップグレードするかどうかを指定します。使用可能なオプションは、**[y]** および **[n]** です。

7 アップグレードを続行する場合、次の詳細を指定します。

バックアップ用の OSP インストールフォルダ

これは、OSP と Identity Applications が同じサーバ上にある場合にのみ適用されます。

OSP バックアップデータを格納する OSP バックアップフォルダを指定します。

SSPR Installation Folder (SSPR インストールフォルダ)

これは、SSPR と Identity Applications が同じサーバ上にある場合にのみ適用されます。

SSPR インストールフォルダを指定します。

SSPR not found on system.Do you want to install & configure it? (SSPR がシステムに見つかりません。インストールして設定しますか ?)

これは、Identity Applications と SSPR が異なるサーバ上にある場合にのみ適用されます。

[y] を選択すると、SSPR は Identity Applications と同じサーバにインストールされます。既存のカスタマイズ設定を、新しい SSPR インストール済みのサーバにコピーする必要があります。

- ［SSPR 設定パスワード］ : SSPR 設定パスワードを指定します。
- ［One SSO Server DNS/IP アドレス］: OSP がインストールされるサーバの IP アドレスを指定します。
- ［One SSO Server SSL ポート］ : OSP SSL ポートを指定します。

[n] を選択すると、SSPR はインストールされず、Identity Applications はアップグレードされます。

User Application Installation Folder (ユーザアプリケーションインストールフォルダ)

ユーザアプリケーションインストールフォルダを指定します。

Identity Applications One SSO Service Password (Identity Applications の One SSO Service パスワード)

One SSO パスワードを指定します。指定されたパスワードは、設定更新ユーティリティで設定したすべてのクライアントのクライアントシークレットを更新します。必要に応じて、設定更新ユーティリティから各クライアントのパスワードをリセットできます。詳細については、134 ページの 「SSO Clients Parameters (SSO クライアントパラメータ)」を参照してください。

Identity Applications Database JDBC jar file (Identity Applications データベースの JDBC jar ファイル)

データベース JAR ファイルを指定します。たとえば、PostgreSQL データベースを使用していて、このデータベースが同じサーバにインストールされている場合、既存のデータベース jar ファイルのデフォルトの場所は `/opt/netiq/idm/postgres/postgresql-42.6.1.jar` です。

Create Schema for Identity Applications (Identity Applications のスキーマの作成)

データベーススキーマをいつ作成するかを指定します。使用可能なオプションは、[Now]、[Startup]、および [File] です。デフォルトのオプションは [Now] です。

注 : MsSQL または Oracle を使用して idmadmin ユーザにこのデータベースのすべての権限を割り当てる場合は、`igaworkflowdb` を作成する必要があります。

Identity Applications Database User Password (Identity Applications データベースユーザパスワード)

データベースのユーザのパスワードを指定します。

Identity Applications Database Administrator Password (Identity Applications データベース管理者パスワード)

データベースの管理者のパスワードを指定します。

8 Tomcat を起動します。データベーススキーマを直ちに作成することを選ぶ場合は、[Now] を選択します。

```
systemctl start netiq-tomcat.service
```

(オプション) アップグレード時に、データベーススキーマを作成するために [Startup (起動)] または [Write to file (ファイルに書き込み)] のオプションを選択した場合、データをワークフローデータベースにインポートする必要があります。詳細については、『NetIQ Identity Manager - Administrator's Guide to the Identity Applications』の「*Migrating Data from Identity Applications Database into Workflow Engine Database*」を参照してください。

9 NGINX サービスを再起動します。

```
systemctl restart netiq-nginx.service
```

SSPR のアップグレード

SSPR が Advanced Edition の Identity Applications サーバとは異なるサーバにインストールされている場合、この方法を使用します。

これは、Standard Edition で SSPR をアップグレードする唯一の方法です。

SSPR をアップグレードするには :

1 NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` をマウントします。

3 `.iso` ファイルのルートディレクトリから、`sspr` ディレクトリに移動します。

4 次のコマンドを実行します。

```
./install.sh
```

5 使用許諾契約を読みます。

6 使用許諾契約の条項を確認し、「`y`」と入力します。

7 [y] を指定して、SSPR をアップグレードします。

8 [識別ボールト管理者パスワード] を指定して、アップグレードを完了します。

Identity Applications コンポーネントのアップグレード後のタスク

Identity Applications の使用を開始する前に、以下のタスクを実行します。

- Tomcat および ActiveMQ サービスの以前のバージョンを手動で削除します。たとえば、次のコマンドを実行します。

```
/etc/init.d/idmapps_tomcat_init
```

```
/etc/init.d/idmapps_activemq_init
```

- Tomcat、SSPR、OSP、および Kerberos のカスタマイズされた設定を手動で復元する必要があります。
- Identity Applications としての CN での証明書は、Identity Applications サーバのキーストア (idm.jks) に存在している必要があります。拡張 Java セキュリティの一環として、現在では Identity Applications では OSP と通信するためのトラステッド証明書が必要です。
- 既存の Identity Applications キーストアファイルを使用して、署名済みの証明書を idm.jks にインポートします。次に例を示します。

```
./keytool -import -alias mycerts -keystore /opt/netiq/idm/apps/tomcat/
conf/idm.jks -file /opt/certs/chap8.der
```

注 : この手順は 4.7.5 から 4.9 へのアップグレードに必要です。

- Identity Applications をクラスタ環境でアップグレードしている場合は、Identity Applications のアップグレード後に次の手順を実行する必要があります。
  - /opt/netiq/idm/apps/tomcat/conf ディレクトリに移動して、context.xml ファイルの Context タグに次の行を追加します。

    ```
    <Manager notifyListenersOnReplication="true"
    expireSessionsOnShutdown="false"
    className="org.apache.catalina.ha.session.DeltaManager"/>
    ```

  - /opt/netiq/idm/apps/tomcat/conf ディレクトリに移動して、server.xml ファイルの Cluster タグに次の行を追加します。

    ```
    <Channel className="org.apache.catalina.tribes.group.GroupChannel">
                <Membership
    className="org.apache.catalina.tribes.membership.McastService"
                            address="228.0.0.4"
                            port="45564"
                            frequency="500"
                            dropTime="3000"/>
                <Receiver
    className="org.apache.catalina.tribes.transport.nio.NioReceiver"
                          address="auto"
                          port="5000"
                          selectorTimeout="100"
                          maxThreads="6"/>

                <Sender
    className="org.apache.catalina.tribes.transport.ReplicationTransmit
    ter">
                  <Transport
    className="org.apache.catalina.tribes.transport.nio.PooledParallelS
    ender"/>
                </Sender>
                <Interceptor
    className="org.apache.catalina.tribes.group.interceptors.TcpFailure
    Detector"/>
                <Interceptor
    className="org.apache.catalina.tribes.group.interceptors.MessageDis
    ```

```
patchInterceptor"/>
            <Interceptor
className="org.apache.catalina.tribes.group.interceptors.Throughput
Interceptor"/>
          </Channel>

          <Valve
className="org.apache.catalina.ha.tcp.ReplicationValve"

filter=".*\.gif|.*\.js|.*\.jpeg|.*\.jpg|.*\.png|.*\.htm|.*\.html|.*
\.css|.*\.txt"/>

          <Deployer
className="org.apache.catalina.ha.deploy.FarmWarDeployer"
                    tempDir="/tmp/war-temp/"
                    deployDir="/tmp/war-deploy/"
                    watchDir="/tmp/war-listen/"
                    watchEnabled="false"/>

          <ClusterListener
className="org.apache.catalina.ha.session.ClusterSessionListener"/>
```

  - 以下のファイルをプライマリノードからすべてのユーザアプリケーションノードにコピーします。
    - ism-sensitive.properties
    - ism-configuration.properties
    - encrypt-keys.pkcs12
- データベースが SSL 経由で設定されている場合は、/opt/netiq/idm/apps/tomcat/conf/ にあるパスから取得される server.xml ファイルで ssl=true を sslmode=require に置き換えます。

  たとえば、次のように変更します。

  ```
  jdbc:postgresql://<postgres db>:5432/idmuserappdb?ssl=true
  ```

  変更先 :

  ```
  jdbc:postgresql://<postgres db>:5432/idmuserappdb?sslmode=require
  ```

Tomcat

- クラスタ環境で、server.xml の Cluster タグを手動でコメント解除して、/opt/netiq/idm/apps/osp_backup_<date> にある最初のノードからすべてのノードに osp.jks をコピーします。
- カスタマイズしたキーストアファイルがある場合は、新しい server.xml ファイルに正しいパスを含めてください。

SSPR

Identity Applications と SSPR が異なるサーバに展開されており、SSPR がインストールされている新しいサーバに既存の SSPR のカスタマイズされた設定を復元することを選択した場合、新しい SSPR サーバで ConfigUpdate ユーティリティを使用して SSPR 設定を変更してください。詳細については、134 ページの 「SSO Clients Parameters (SSO クライアントパラメータ)」を参照してください。

One SSO Provider

アップグレード前のセットアップで Identity Applications と OSP が異なるサーバに展開されている場合、既存の OSP 設定を、OSP がインストールされている新しいサーバ (Identity Applications サーバ) にコピーし、このサーバ上でインストールキットから `merge_jars` メソッドを実行して設定を復元します。

1. Identity Applications をアップグレードしたサーバで Tomcat を停止します。(OSP は Identity Applications のアップグレードでインストールされます)
2. カスタマイズを復元します。
   - **2a** 既存の OSP サーバの OSP インストールディレクトリに移動し、`osp-custom-resource.jar` ファイルを見つけます。

     たとえば、`/opt/netiq/backup_idm/osp/osp-extras/l10n-resources/osp-custom-resource.jar` というファイルを見つけます。
   - **2b** `osp-custom-resource.jar` ファイルを、Identity Applications をアップグレードしたサーバ上の場所にコピーします。
   - **2c** `<location where you have mounted the Identity_Manager_4.9.0_Linux.iso>/osp/scripts/merge_cust_loc.sh` に移動します。

     このスクリプトには、既存のカスタマイズと新しくインストールされた OSP のマージを処理する `merge_jars` メソッドが含まれています。
   - **2d** コマンドプロンプトを開き、次のコマンドを実行します。

     ```
     merge_jars ${IDM_BACKUP_FOLDER in the remote OSP server}/tomcat/lib/
     osp-custom-resource.jar ${IDM 4.9_OSP_INSTALLED_HOME}/osp-extras/
     l10n-resources/osp-custom-resource.jar)
     ```

     次に例を示します。

     ```
     merge_jars /opt/netiq/backup_idm/osp/osp-extras/l10n-resources/osp-
     custom-resource.jar /opt/netiq/idm/apps/osp/osp-extras/l10n-
     resources/osp-custom-resource.jar
     ```

     ここで `backup_idm` ディレクトリには、既存の OSP サーバの OSP 設定が含まれます。
3. OSP がインストールされている新しいサーバで Tomcat を起動します。

その他の設定の更新については、134 ページの 「SSO Clients Parameters (SSO クライアントパラメータ)」を参照してください。

Kerberos

アップグレードユーティリティにより、コンピュータ上に新しい Tomcat フォルダが作成されます。`keytab` や `Kerberos_login.config` などの Kerberos ファイルが古い Tomcat フォルダ上にある場合は、バックアップしたフォルダから新しい Tomcat フォルダにこれらのファイルをコピーします。

アップグレード後のバージョン番号の確認

Identity Manager 4.9 にアップグレードした後、コンポーネントが以下のバージョンにアップグレードされていることを確認します。

- Tomcat – `9.0.85`
- ActiveMQ – `5.18.3`
- Java – `1.8.0_222`
- One SSO Provider – `6.7.0`
- セルフサービスパスワードリセット – `4.7.0.1`

Identity Reporting のアップグレード

Identity Reporting には 2 つのドライバが含まれます。アップグレードは次の順序で実行します。

注 : データベースは、サポートされるバージョンにアップグレードしてください。

---

1. データベースを、サポートされるバージョンにアップグレードします。PostgreSQL データベースのアップグレードについては、231 ページの 「PostgreSQL のアップグレード」を参照してください。
2. Sentinel Log Management for IGA をアップグレードします。詳細については、239 ページの 「Sentinel Log Management for IGA のアップグレード」を参照してください。
3. Identity Reporting をアップグレードします . 詳細については、239 ページの 「Identity Reporting のアップグレード」を参照してください。
4. Managed System Gateway Driver を設定します。詳細については、158 ページの 「管理対象システムのゲートウェイドライバの設定」を参照してください。
5. データ収集を設定します。詳細については、『Administrator Guide to NetIQ Identity Reporting』の「Configuring Settings and Data Collection」を参照してください。

アップグレードに関する考慮事項

アップグレード時に、postgresql-9.4.1212.jar ファイルの正しい場所を指定していることを確認します。デフォルトの場所は、/opt/netiq/idm/postgres/ です。次のシナリオでは、データベース接続が失敗します。

- 間違ったパスを指定した場合
- 間違った jar ファイルを提供した場合
- ファイアウォールが有効になっている場合
- データベースがリモートマシンからの接続を許可しない場合

Sentinel Log Management for IGA のアップグレード

1 NetIQ Downloads の Web サイトから、SentinelLogManagementForIGA8.6.1.0.tar.gz をダウンロードします。

2 ファイルの抽出先のディレクトリに移動します。

3 次のコマンドを実行して、ファイルを抽出します。

```
tar -zxvf SentinelLogManagementForIGA8.6.1.0.tar.gz
```

4 SentinelLogManagementforIGA ディレクトリに移動します。

5 SLM for IGA をインストールするには、次のコマンドを実行します。

```
./install.sh
```

6 インストールに使用する言語を指定し、<Enter> を押します。

7 「y」と入力して使用許諾契約に同意し、アップグレードを完了します。

注 : SLM for IGA をアップグレードした後で、最新のコレクタを手動でインポートします。

1. SentinelLogManagementForIGA8.6.1.0.tar.gz ファイルを抽出したディレクトリに移動します。
2. /content/ ディレクトリに移動します。
3. コレクタをインポートして設定します。詳細については、『*NetIQ Identity Manager - Configuring Auditing in Identity Manager*』の「Installing and Configuring the Sentinel Collectors」を参照してください。

Identity Reporting のアップグレード

1 NetIQ ダウンロード Web サイトから Identity_Manager_4.9.0_Linux.iso をダウンロードします。

2 ダウンロード済みの .iso をマウントします。

3 次のコマンドを実行します。

```
./install.sh
```

4 使用許諾契約を読みます。

5 使用許諾契約の条項を確認し、「y」と入力します。

6 Identity Manager コンポーネントをアップグレードするかどうかを指定します。使用可能なオプションは、[y] および [n] です。

7 Identity Reporting を選択して、アップグレードを続行します。

8 次の詳細を指定します。

OSP Installed (OSP のインストール): OSP がインストールされているかどうかを指定します。

OSP Install Folder (OSP インストールフォルダ): OSP のバックアップインストールフォルダを指定します。

Reporting Installation Folder for backup (バックアップ用の Reporting インストールフォルダ): Reporting インストールフォルダを指定します。

When do you want to create schema for Identity Reporting (Identity Reporting 用のスキーマを作成するタイミング): データベースのスキーマを今すぐ作成するか、後で作成するかを指定します。使用可能なオプションは、[Now]、[Startup]、および [File] です。

Identity Reporting データベース JDBC jar ファイル : Identity Reporting のデータベース JAR ファイルを指定します。既存データベース jar ファイルのデフォルトの場所は、/opt/netiq/idm/apps/postgres/postgresql-9.4.1212.jar です。

Identity Reporting データベースユーザ : Reporting データベースユーザの名前を指定します。

Identity Reporting データベースアカウントパスワード : Reporting データベースのパスワードを指定します。

注 : Identity Manager を 4.9 にアップグレードした後は、次のような状態になります。

- データ同期ポリシーは IDMDCS UI で表示できなくなります。新しいポリシーの作成を計画している場合は、Sentinel サーバの既存のデータ同期ポリシーを削除して、Identity Reporting の設定後に IDMDCS UI を使用して、新しいデータ同期ポリシーを作成する必要があります。
- ism-configuration.properties ファイルの com.netiq.rpt.ssl-keystore.type プロパティはアップグレード前に設定された値 (JKS/PKCS12) を保持します。

注 : Identity Reporting を 4.8.7 から 4.9 にアップグレードする場合、Java 8 ベースの tomcat は次の場所にバックアップされます : pt/netiq/idm/apps/deprecate-tomcat-jre8。

Reporting のアップグレード後の手順

- 分散セットアップでは、Identity Applications および Identity Reporting を 4.8.x から 4.9 にアップグレードした後で、次の手順を実行してください。
  1. 次のコマンドを実行して、Identity Applications の idm.jks ファイルから OSP 証明書をインポートし、新しい Java キーストアファイルに配置します。

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /
opt/netiq/idm/apps/tomcat/conf/idm.jks -srcstorepass novell-
destkeystore ./idm.jks -deststorepass novell -srcalias "cn=<user-
name>, o=<organization-name>" -destalias "cn=<user-name>" -noprompt
```

次に例を示します。

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /
opt/netiq/idm/apps/tomcat/conf/idm.jks -srcstorepass novell-
destkeystore ./idm.jks -deststorepass novell -srcalias "cn=sean,
o=novell" -destalias "cn=sean" -noprompt
```

2. Identity Reporting サーバの既存の Java キーストアファイルをこの新しく作成されたキーストアファイルと置き換えて、Identity Reporting サーバを再起動します。

- アップグレード時に、[Create Schema for Identity Reporting (Identity Reporting 用のスキーマの作成)] で **[Startup]** または **[File]** を選択した場合、次の手順を実行してください。

1. Identity Reporting にログインします。
2. Identity Reporting リポジトリから既存のデータソースとレポート定義を削除します。
3. 新しい Identity Manager データ収集サービスデータソースを追加します。

また、利用可能なデータベースが正しく実行されるように、Tomcat を一度起動した後に再起動してください。

- Identity Reporting を 4.9 にアップグレードした後で、`/opt/netiq/idm/apps/tomcat/conf/` ディレクトリにある `ism-configuration.properties` ファイルに移動し、次のアクションを実行します。
  - `com.netiq.rpt.landing.url` プロパティの値を次のように変更します。

    ```
    com.netiq.rpt.landing.url = ${com.netiq.idm.osp.url.host}/idmdash/
    #/landing
    ```

  - `com.netiq.idmdcs.landing.url` プロパティの値を次のように変更します。

    ```
    com.netiq.idmdcs.landing.url = ${com.netiq.idm.osp.url.host}/
    idmdash/#/landing
    ```

  - 次の形式で `com.netiq.rpt.redirect.url` プロパティの値を指定します。

    `https:<hostname>:<port>/path`

    例 : `com.netiq.rpt.redirect.url = https://192.168.0.1:8543/IDMRPT/oauth.html`

必要な変更を行った後で、ファイルを保存して、Tomcat を再起動します。

- データベースが SSL 経由で設定されている場合は、`/opt/netiq/idm/apps/tomcat/conf/` にあるパスから取得される `server.xml` ファイルで `ssl=true` を `sslmode=require` に置き換えます。

  たとえば、次のように変更します。

  ```
  jdbc:postgresql://<postgres db>:5432/idmrptdb?ssl=true
  ```

  変更先 :

  ```
  jdbc:postgresql://<postgres db>:5432/idmrptdb?sslmode=require
  ```

Identity Reporting のアップグレードの検証

1 Identity Reporting を起動します。
2 ツールで古いレポートと新しいレポートが表示されることを検証します。
3 [カレンダ] を参照し、スケジュールされたレポートが表示されるかどうかを確認します。
4 [設定] ページに管理対象アプリケーションと管理対象外アプリケーションの設定が表示されることを確認します。
5 他の設定がすべて正しく見えることを検証します。
6 完了したレポートがリストに表示されるかどうかを検証します。

Analyzer のアップグレード

重要 : Analyzer ツールは Identity Manager 4.9 の ISO に含まれません。Analyzer がすでにインストールされている場合は、必ずバージョン 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 とともに使用してください。

1 [Software and License (ソフトウェアおよびライセンス)] ページ (https://sld.microfocus.com/) から `Identity_Manager_4.8.5_Linux_Analyzer.tar.gz` をダウンロードします。
2 この `.zip` ファイルを Analyzer インストールファイル (プラグイン、アンインストールスクリプト、および他の Analyzer ファイルなど) が置かれているディレクトリに展開します。
3 Analyzer を再起動します。
4 新しいパッチの適用に成功したことを検証するために、次の手順を実行します。
   4a Analyzer を起動します。
   4b [ヘルプ] > [About Analyzer (バージョン情報)] の順にクリックします。
   4c プログラムに新しいバージョンが表示されるかどうかを確認します。

アップグレード後のタスク

アップグレード後に次のアクションを実行します。

サービスドライバのアップグレード

注 : ドライバパッケージを更新する前に、Identity Applications の最新バージョンに更新済みであることを確認します。

Identity Applications を最新バージョンに更新したら、ユーザアプリケーションドライバパッケージを 4.8.8.20240401125038 に、役割およびリソースサービスドライバパッケージを 4.8.8.20231216023752 に、データ収集サービスドライバパッケージを 2.4.2.20230627174512 にそれぞれ更新できます。

ドライバパッケージを更新するには、次の手順を実行します。

1. 必要なサービスドライバが含まれるプロジェクトを開きます。
2. サービスドライバを右クリックし、[ドライバ] > [プロパティ] をクリックします。
3. [Packages (パッケージ)] をクリックします。

   [Available Upgrades (利用可能なアップグレード)] 列のチェックマークは、新しいバージョンのパッケージを示します。
4. (状況によって実行) UAD パッケージの場合は、`NOVLUABASE_4.8.8.20240401125038.jar` パッケージに対して [選択] 操作をクリックします。
5. (状況によって実行) RRSD パッケージの場合は、`NOVLRSERVB_4.8.8.20231216023752.jar` パッケージに対して [選択] 操作をクリックします。
6. (状況によって実行) DCS パッケージの場合は、`NOVLRSERVB_2.4.2.20230627174512.jar` パッケージに対して [選択] 操作をクリックします。
7. ドロップダウンリストで [アップグレード] をクリックします。
8. アップグレード先のバージョンを選択して、[OK] をクリックします。

   注 : Designer に、アップグレードに利用できるすべてのバージョンが一覧にされます。
9. [適用] をクリックします。
10. (状況によって実行) パッケージをアップグレードするための適切な情報をフィールドに入力して、[[次へ] をクリックします。

    アップグレード対象として選択したパッケージに応じて適宜、パッケージをアップグレードするために必要な情報を入力する必要があります。
11. インストールされるパッケージの概要を読んで、[終了] をクリックします。
12. アップグレードされたパッケージを確認し、[OK] をクリックして［パッケージ管理］ページを閉じます。
13. 変更内容を展開して、ドライバを再起動します。
14. eDirectory サービスを再開します。

未使用の Log4j JAR の削除

Identity Manager 4.9 のインストールの一部として、`/opt/netiq/idm/lightWeightDesigner/plugins/com.novell.soa.eai.integrationActivity_4.0.0.201910221801/lib` ディレクトリに `log4j-1.2.13.jar` がインストールされます。Identity Manager は現在はこの jar を使用しません。`log4j-1.2.13.jar` を見つけてディレクトリから削除します。

新しいサーバをドライバセットに追加する

Identity Manager を新しいサーバにアップグレードまたはマイグレートする場合、ドライバセット情報をアップデートする必要があります。このセクションでは、このプロセスについて説明します。Designer または Identity Console を使用してドライバセットをアップデートできます。

たとえば、Identity Console を使用している場合は、次の手順を実行します。

1 Identity Console で、[IDM Administration (IDM 管理)] をクリックします。
2 ドライバセットを選択します。
3 をクリックします。
4 [Add Server (サーバの追加)] をクリックします。
5 新しい Identity Manager サーバを参照して選択し、[OK] をクリックします。

ドライバセットから古いサーバを削除する

新しいサーバがすべてのドライバを実行した後、ドライバセットから古いサーバを削除できます。

- 244 ページの 「Designer を使用してドライバセットから古いサーバを削除する」
- 244 ページの「Identity Console を使用してドライバセットから古いサーバを削除する」
- 245 ページの 「古いサーバの退役」

Designer を使用してドライバセットから古いサーバを削除する

1 Designer で、プロジェクトを開きます。
2 Modeler で、ドライバセットを右クリックし、[プロパティ] を選択します。
3 [サーバリスト] を選択します。
4 [Selected Servers (選択したサーバ)] リストで古い Identity Manager サーバを選択し、[<] をクリックして、[Selected Servers (選択したサーバ)] リストからサーバを削除します。
5 [OK] をクリックし、変更を保存します。
6 識別ボールトに変更を展開します。
詳細については、『*NetIQ Designer for Identity Manager Administration Guide*』の「識別ボールトへのドライバセットの展開」を参照してください。

Identity Console を使用してドライバセットから古いサーバを削除する

1 Identity Console で、[IDM Administration (IDM 管理)] をクリックします。
2 ドライバセットを選択します。
3 をクリックします。

4 [サーバの削除] をクリックします。

5 新しい Identity Manager サーバを参照して選択し、[OK] をクリックします。

古いサーバの退役

この時点で、古いサーバがホストしているドライバはありません。このサーバが必要でなくなった場合は、追加の手順を実行し、サーバを廃止する必要があります。

1 このサーバから eDirectory のレプリカを削除します。

詳細については、『*NetIQ eDirectory 管理ガイド*』を参照してください。

2 このサーバから eDirectory を削除します。

ドライバへのカスタムポリシーとルールの復元

ドライバの新しいパッケージをインストールまたはアップグレードした後、新しいドライバ環境設定ファイルをオーバーレイした後にカスタムポリシーとルールを復元する必要があります。これらのポリシーに別の名前が付いている場合、ポリシーはドライバ内にそのまま保存されていますが、リンクが壊れているので、再設定する必要があります。

- 245 ページの 「Designer を使用したドライバへのカスタムポリシーとルールの復元」
- 246 ページの「Identity Console を使用したドライバへのカスタムポリシーとルールの復元」

Designer を使用したドライバへのカスタムポリシーとルールの復元

ポリシーセットにポリシーを追加できます。この手順は、アップグレードしたドライバを運用環境に移動する前に、テスト環境で実行する必要があります。

1 [アウトライン] ビューで、アップグレードしたドライバを選択してから、[ポリシーフローの表示] アイコン をクリックします。

2 カスタマイズしたポリシーをドライバに復元する必要があるポリシーセットを右クリックして、[ポリシーの追加] > [既存の項目をコピー] の順に選択します。

3 カスタマイズしたポリシーを参照して選択し、[OK] をクリックします。

4 カスタマイズしたポリシーの名前を指定し、[OK] をクリックします。

5 ファイルの競合を示すメッセージが表示されたら、[はい] をクリックしてプロジェクトを保存します。

6 ポリシービルダでポリシーが開いたら、コピーしたポリシーの情報が正しいことを確認します。

7 ドライバに復元する必要があるカスタマイズした各ポリシーに対して、ステップ 2 ～ステップ 6 を繰り返します。

8 ドライバを起動してテストします。

ドライバの起動の詳細については、<TBD - 「ドライバの起動」セクションが記載される文書へのリンクを提供する > を参照してください。ドライバのテストの詳細については、『*NetIQ Identity Manager - Using Designer to Create Policies*』の「ポリシーシミュレータを使用したポリシーのテスト」を参照してください。

9 ポリシーが動作することを確認したら、ドライバを運用環境に移します。

Identity Console を使用したドライバへのカスタムポリシーとルールの復元

アップグレードしたドライバを運用環境に移す前に、テスト環境でこれらの手順を実行します。

1 Identity Console で、**[Identity Manager Administration (Identity Manager の管理)]** に移動します。

2 アップグレードしたドライバを含むドライバセットオブジェクトをクリックします。

3 ドライバアイコンをクリックします。

4 **[Data Transformation and Synchronization (データ変換と同期)]** > **[Policies (ポリシー)]** に移動し、カスタマイズされたポリシーを復元する必要があるポリシーセットを選択します。

5 **[Insert Policy (ポリシーの挿入)]** をクリックします。

6 **[Use an existing policy (既存のポリシーを使用)]** ポップアップで、カスタムポリシーを参照して選択します。

7 **[保存]** をクリックします。

8 ドライバに復元する必要がある各カスタムポリシーに対して、ステップ 2 ～ステップ 7 を繰り返します。

9 ドライバを起動してテストします。

ドライバの起動については、225 ページの 「ドライバの起動」を参照してください。Identity Console にはポリシーシミュレータはありません。ポリシーをテストするには、ポリシーを実行するイベントを発生させます。たとえば、ユーザの作成、ユーザの変更、またはユーザの削除などです。

10 ポリシーが動作することを確認したら、ドライバを運用環境に移します。

16 Advanced Edition から Standard Edition への切り替え

ご使用の環境で Advanced Edition 機能が必要なく、Identity Manager の展開をスケールダウンする場合にのみ Standard Edition に切り替える必要があります。

1 (状況によって実行) すでに Advanced Edition のアクティベーションを適用している場合は、それを削除します。

2 (状況によって実行) Standard Edition 評価モードに切り替える場合は、次のアクションを実行します。

2a アイデンティティボールト `dib` ディレクトリに移動します。

```
/var/opt/novell/eDirectory/data/dib
```

2b 新しいファイルを作成し、それに `.idme` という名前を付け、そのファイルに 2 (数字) を追加します。

2c アイデンティティボールトを再起動します。

2d 手順 4 に進みます

3 (状況によって実行) すでに Standard Edition のアクティベーションを購入している場合は、アクティベーションを適用します。

4 Tomcat を停止します。

5 `/opt/netiq/idm/apps/tomcat/webapps` ディレクトリから次の WAR ファイルと Webapps フォルダを削除します。

- `IDMProv*`
- `IDMRPT*`
- `dash*`
- `idmdash*`
- `landing*`
- `rra*`
- `rptdoc*`

6 次の既存のフォルダをバックアップディレクトリに移動します。

- `IDMReporting`
- `UserApplication`

7 `ism-configuration.properties` ファイルを `<install folder>/tomcat/conf` ディレクトリからバックアップディレクトリにコピーします。

8 Identity Reporting をインストールします。詳細については、67 ページの第 4 章「Identity Manager のインストール」を参照してください。

9 *<reporting install folder>*/bin ディレクトリから configupdate.sh を起動し、次のパラメータの値を指定します。

［**レポーティング**］**タブ**：次のセクションの設定を指定します。

- アイデンティティボールト
- 識別ボールトユーザ ID
- レポート管理者
  - **[レポーティング管理者の役割コンテナの DN]**. たとえば、ou=sa,o=data
  - **[レポート管理者]**。たとえば、cn=uaadmin,ou=sa,o=data

［**認証**］**タブ**：次のセクションの設定を指定します。

- Authentication Server(認証サーバ)
  - **[OAuth サーバのホスト識別子]**. たとえば、192.168.0.1 のような認証サーバの IP アドレスまたは DNS 名
  - **[OAuth サーバの TCP ポート]**
  - **[OAuth サーバは TLS/SSL を使用しています]**
- 認証の設定
  - **[OAuth キーストアファイル]**. たとえば、/opt/netiq/idm/apps/osp/osp.jks
  - **[OAuth で使用するためのキー別名]**
  - **[Key password of key for use by OAuth (OAuth で使用するキーのキーパスワード)]**
  - **[セッションタイムアウト (分)]**. たとえば、60 分です。

［**SSO クライアント**］**タブ**：次のセクションの設定を指定します。

- レポーティング
  - **[ランディングページへの URL リンク]**. たとえば、http://192.168.0.1:8180/IDMRPT です。
- セルフサービスパスワードリセット
  - **[OAuth クライアント ID]**. たとえば、*sspr* です。
  - OAuth クライアントシークレット。たとえば、*<sspr client secret>* です。
  - **[OSP OAuth リダイレクト URL]**. たとえば、http://192.168.0.1:8180/sspr/public/oauth です。

設定ユーティリティの詳細については、114 ページの 「識別情報アプリケーション設定ユーティリティの実行」を参照してください。

10 設定ユーティリティで変更を保存して終了します。

11 Tomcat を起動します。

VII Identity Manager コンテナの展開

Identity Manager では、コンテナ化されたメカニズムにより Identity Manager コンポーネントを柔軟に展開することができます。Identity Manager は、コンテナの管理に Docker を使用します。コンテナ化をサポートする Identity Manager コンポーネントは、Docker イメージとして配信されます。Docker イメージは自己完結型です。

Docker イメージは次の Identity Manager コンポーネントで使用できます。

- Identity Manager エンジン
- リモートローダ
- OSP (One SSO Provider)
- ファンアウトエージェント
- ActiveMQ
- PostgreSQL (Redistribution)
- 識別情報アプリケーション
- SSPR (Self Service Password Reset)
- フォームレンダラ
- Identity Reporting
- Identity Console

注 : Identity Configuration Generator イメージは、サイレントプロパティファイルの生成に使用されます。

エンタープライズモードのインストールで実現可能なすべての機能と操作は、コンテナ化されたメカニズムでも利用可能です。

17 コンテナ展開の計画

以下のセクションでは、Docker 環境でのコンテナベースの展開に必要な大まかな計画について説明します。

- 251 ページの 「システム要件」
- 251 ページの 「Docker イメージの取得」

システム要件

コンテナを展開するための次の要件が満たされていることを確認する必要があります。

ソフトウェア	認定バージョン
Docker	24.0.7-ce 以降

Docker イメージの取得

Docker イメージを取得するには、次の手順を実行します。

1. Software License and Download (ソフトウェアライセンスおよびダウンロード) (https://sld.microfocus.com) ポータルから、`Identity_Manager_4.9.0_Containers.tar.gz` をダウンロードします。
2. 次のコマンドを実行して、`.tar` ファイルを抽出します。

   ```
   tar -zxvf Identity_Manager_4.9.0_Containers.tar.gz
   ```

18 Identity Manager コンテナの展開

Identity Manager コンテナの展開プロセスには、インストール前、インストール、およびインストール後の作業が必要です。このセクションで使用される例は、Advanced Edition のコンテナの展開に関する情報を提供します。ただし、コンテナは Standard Edition でも展開できます。

一部のコンテナは他のコンテナに依存しています。次のテーブルに、他のコンテナに依存しているコンテナの詳細を示します。

表 18-1 *依存しているコンテナ*

コンテナ	依存しているコンテナ
OSP	◆ Identity Engine
識別情報アプリケーション	◆ OSP ◆ Identity Applications 用のデータベース
フォームレンダラ	識別情報アプリケーション
Identity Reporting	◆ 識別情報アプリケーション ◆ Identity Reporting 用のデータベース
SSPR	識別情報アプリケーション

ベストプラクティス

このセクションには、Docker コンテナの展開のいくつかのヒントおよびベストプラクティスが含まれます。

- NetIQ では、コンテナに使用する CPU 量に制限を設定することをお勧めします。これは、docker run コマンドで `--cpuset-cpus` フラグを使用して実現できます。
- コンテナの再起動ポリシーを設定するには、docker run コマンドで `--restart` フラグを使用します。障害時の再起動ポリシーを選択し、再起動の試みを 5 に制限することをお勧めします。
- コンテナによって使用されるメモリに制限を設定するには、docker run コマンドで `--memory` フラグを使用します。
- 展開されたドライバのトレースファイルをバックアップする場合は、`/config/idm/` の下にトレースファイルを置くか、ボリューマイズされたフォルダにトレースファイルを手動でコピーします。
- 任意の時点で実行可能なプロセス数に制限を設定するには、docker run コマンドで --pids-limit フラグを使用します。PID 値を 300 に制限することをお勧めします。

- Identity Manager エンジンコンテナでは、/proc ファイルシステムの /process ディレクトリにある environ ファイルを表示する場合は、docker run コマンドで --cap-add=SYS_PTRACE フラグを使用します。デフォルトでは、権限のほとんどが制限され、必要な権限のみが有効になります。詳細については、Docker のマニュアルを参照してください。
- コンテナが起動されるたびに使用できるように、サードパーティの jar ファイルがボリュームマウントされていることを確認します。たとえば、ojdbc.jar がコンテナの /opt/netiq/idm/apps/tomcat/lib ディレクトリに存在する場合は、次のコマンドを使用して jar ファイルをボリュームマウントする必要があります。

  ```
  -v /host/ojdbc.jar:/opt/netiq/idm/apps/tomcat/lib/ojdbc.jar
  ```

たとえば、コンテナを展開するために上記すべての引数を含む次のサンプルコマンドを実行します。

```
docker run -itd --cap-add=SYS_PTRACE --pids-limit<tune container pids
limit> --memory=<maximum amout of memory container can use> --restart=on-
failure:5 --cpuset-cpus=<CPUs in which to allow execution> --
network=<connect a container to network> -v <bind mount a volume> --
name=<assign a name to the container> <image name>
```

コンテナボリュームデータの管理

Docker は、データの保存と永続化のためにいくつかのメカニズムをサポートしています。コンテナデータを永続化するこのようなメカニズムの 1 つは、コンテナ内の共有ボリュームを使用するものです。

このガイドで使用される例は、共有ボリュームを作成して使用することを前提としています。たとえば、Docker ホストに /data という共有ボリュームを作成します。

```
mkdir /data
```

作成した共有ボリュームには許可が必要です。次のコマンドを使用して許可を指定します。

```
chmod -R 777 /data
```

ただし、Docker がサポートする他のボリュームを使用できます。詳細については、Docker のマニュアルを参照してください。

注 : 共有ボリュームの読み書き許可があることを確認します。

サイレントプロパティファイルの作成

Identity Manager コンテナのほとんどがインタラクティブモードのインストールを提供しています。ただし、NetIQ では、さまざまなコンテナの展開にサイレントプロパティファイルを使用することを推奨しています。

注 : silent.properties ファイルが生成されたら、Docker ホストの /data で使用できるようになります。

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers/docker-images ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_idm_conf_generator.tar.gz
```

3 次のサンプルコマンドを使用してコンテナを展開します。

注 : マシン FQDN を必ずホスト名の値として指定してください。

```
docker run -it --name=idm_conf_generator --
hostname=identitymanager.example.com -v /data:/config
idm_conf_generator:idm-4.9.0-580
```

4 idm ディレクトリに移動します。

5 create_silent_props.sh ファイルを実行します。

```
./create_silent_props.sh
```

6 コンポーネントのインストールと設定を続行するには、「y」と入力します。

7 JRE をインストールするには、「y」と入力します。

8 プロパティファイルのパスを指定します。

注 : 共有ボリュームの場所に silent.properties ファイル、たとえば、/config を作成してください。

9 以下の設定を指定して、サイレントプロパティファイルを作成します。

パラメータ	説明
Silent Property file name with absolute path (絶対パスを使用したサイレントプロパティファイル名)	サイレントプロパティファイルのパスを指定します。
Configure the Silent properties for Docker Containers (Docker コンテナのサイレントプロパティの設定)	Docker コンテナのプロパティファイルを設定するかどうかを指定します。
Generate inputs for Kubernetes Orchestration (Kubernetes オーケストレーションの入力の生成)	**[Configure the Silent properties for Docker Containers (Docker コンテナのサイレントプロパティの設定)]** オプションで「y」を選択した場合にのみ適用されます。 Kubernetes の YAML ファイルを生成するかどうかを指定します。

パラメータ	説明
Directory name with absolute path for creating kube yaml file (kube yaml ファイルを作成するための絶対パスを使用したディレクトリ名)	[Generate inputs for Kubernetes Orchestration (Kubernetes オーケストレーションの入力の生成)] オプションで [y] を選択した場合にのみ適用されます。 Kubernetes の YAML ファイルを作成するためのパスを指定します。 **注 :** Identity Applications と Identity Reporting の YAML ファイルには異なるパスを指定することをお勧めします。
Kubernetes volume mount path (Kubernetes ボリュームマウントパス)	［Generate inputs for Kubernetes Orchestration (Kubernetes オーケストレーションの入力の生成)］オプションで［y］を選択した場合にのみ適用されます。 Kubernetes ボリュームのパスを指定します。
Identity Manager Engine hostname for Kubernetes deployment (Kubernetes 展開の Identity Manager エンジンホスト名)	[Generate inputs for Kubernetes Orchestration (Kubernetes オーケストレーションの入力の生成)] オプションで [y] を選択した場合にのみ適用されます。 Identity Manager エンジンのホスト名を指定します。

10 インストールする Identity Manager サーバのエディションを決定します。Advanced Edition の場合は「y」、Standard Edition の場合は「n」と入力します。

11 コンポーネントを標準モードとカスタムモードのどちらで設定するかを決定します。

12 インストール可能なコンポーネントのリストから、必要なコンポーネントを選択します。

- エンジンをインストールするには、[Identity Manager エンジン] を選択します。
- Identity Reporting をインストールするには、[Identity Reporting] を選択します。
- Identity Applications をインストールするには、[Identity Applications] を選択します。

環境設定パラメータの詳細については、83 ページの 「環境設定パラメータの理解」を参照してください。

注 :

- すべてのコンポーネントの `silent.properties` ファイルを一度に生成する必要があります。
- すべての IP 関連の設定プロンプトに FQDN を使用します。

- SSO_SERVER_SSL_PORT、TOMCAT_HTTPS_PORT、UA_SERVER_SSL_PORT、および RPT_TOMCAT_HTTPS_PORT は固有のポートである必要があります。たとえば、SSO_SERVER_SSL_PORT は 8543 に、TOMCAT_HTTPS_PORT と UA_SERVER_SSL_PORT は 18543 に、RPT_TOMCAT_HTTPS_PORT は 28543 にそれぞれ変更します。
- (状況によって実行)ホストネットワークモードを使用して単一サーバにコンテナを展開する場合は、OSP、Identity Applications、および Identity Reporting に固有の証明書関連のプロンプトには、/opt/netiq/idm/apps/tomcat/conf/tomcat.ks として tomcat.ks パスを指定する必要があります。

13 (状況によって実行) ホストネットワークモードを使用して単一サーバにコンテナを展開する場合は、サイレントプロパティファイルが生成された後で、次のステップを実行する必要があります。

- CUSTOM_OSP_CERTIFICATE の値が n に設定されていることを確認します。
- 以下のエントリを silent.properties ファイルの最後に追加します。

```
SKIP_PORT_CHECK=1
CUSTOM_UA_CERTIFICATE="n"
TOMCAT_SSL_KEYSTORE_PASS="<password>"
CUSTOM_RPT_CERTIFICATE="n"
```

リモートローダコンテナの展開を使用したリモートローダインスタンスの自動起動

リモートローダコンテナを展開した後にリモートローダインスタンスを自動的に起動する場合は、次の手順を実行します。

1 Docker ホストで、共有ディレクトリに移動します。たとえば、data です。

2 rdxml ディレクトリを作成します。

注 : これは、コンテナを初めて展開する場合にのみ適用されます。つまり、コンテナを更新する場合で、すでに rdxml ディレクトリを作成済みのときは、この手順をスキップします。

3 rdxml ディレクトリに移動します。

4 rdxml ディレクトリ内に driverconf ディレクトリを作成します。

注 : 複数の環境設定ファイルを異なるリモートローダインスタンスで実行している場合は、すべてのファイルをコピーします。

5 必要な環境設定ファイルをすべて driverconf ディレクトリにコピーします。たとえば、config8000.txt です。

6 rdxml ディレクトリ内に keystore ディレクトリを作成します。

7 必要なキーストアファイルと証明書をすべて keystore ディレクトリにコピーします。

8 `rdxml` ディレクトリ内に新しい `.txt` ファイルを作成します。たとえば、`StartupRL.txt` です。

9 `StartupRL.txt` ファイルに、必要なコンテンツを次の形式で指定します。

```
<Remote Loader configuration file> -sp <driver password> <Remote Loader password>
```

例 :

```
config8000.txt -sp dirxml dirxml
```

または、次の形式でもエントリを指定できます。

```
<Remote Loader configuration file> -ksp <keystore password> -kp <key password>
```

例 :

```
config8000.txt -ksp dirxml -kp dirxml
```

10 docker run コマンドで `RL_DRIVER_STARTUP` 環境変数を渡して、リモートローダコンテナを展開します。たとえば、`-e RL_DRIVER_STARTUP="StartupRL.txt"` です。

コンテナ展開のモード

NetIQ では、コンテナを展開するために以下のモードを使用することをお勧めします。

- 単一サーバ上へのコンテナの展開
- 分散サーバ上へのコンテナの展開

単一サーバ上へのコンテナの展開

この例では、ホストネットワークモードを使用して、すべての Identity Manager コンテナが単一の Docker ホスト上に展開されます。

前提条件

- ホスト名が以下に示されるように、FQDN 形式であることを確認します。

  ```
  <IP of the host> <FQDN> <short_name>
  ```

  例 :

  ```
  172.120.0.1     identitymanager.example.com      identitymanager
  ```

- コンテナを展開する前に、サイレントプロパティファイルを生成する必要があります。サイレントプロパティファイルの生成に関する詳細については、254 ページの 「サイレントプロパティファイルの作成」を参照してください。

コンテナは次の順序で展開される必要があります。

- 259 ページの 「Identity Manager エンジンコンテナの展開」

- 260 ページの 「リモートローダコンテナの展開」
- 260 ページの 「ファンアウトエージェントコンテナの展開」
- 261 ページの 「アイデンティティボールト認証局での証明書の生成」
- 262 ページの 「OSP コンテナの展開」
- 263 ページの 「PostgreSQL コンテナの展開」
- 264 ページの 「Identity Applications コンテナの展開」
- 265 ページの 「フォームレンダラコンテナの展開」
- 266 ページの 「ActiveMQ コンテナの展開」
- 266 ページの 「Identity Reporting コンテナの展開」
- 267 ページの 「SSPR コンテナの展開」

Identity Manager エンジンコンテナの展開

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 `Identity_Manager_4.9.0_Containers.tar.gz file` を抽出した場所から、`Identity_Manager_4.9.0_Containers` ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityengine.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=engine-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 identityengine:idm-4.9.0-580
```

5 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/idm/log/idmconfigure.log
```

6 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it engine-container bash
```

注 : `ndstrace` や `ndsrepair` などのアイデンティティボールトユーティリティを実行するには、`nds` と呼ばれるルート以外のユーザとしてコンテナにログインします。ルートユーザとしてログインしている場合にはこれらのユーティリティは実行できません。`nds` ユーザとしてコンテナにログインするには、`docker exec -it engine-container sudo nds` コマンドを実行します。

リモートローダコンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_remoteloader.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run -d --network=host --name=rl-container -v /data:/config
remoteloader:idm-4.9.0-580
```

これにより、64 ビットバージョンおよび 32 ビットバージョンのリモートローダが展開されます。ドライバファイルはコンテナの /opt/novell/eDirectory/lib/dirxml/classes/ ディレクトリにあります。

4 (状況によって実行) コンテナの展開時に設定ファイルを使用しない場合は、次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=rl-
container -v /data:/config --stop-timeout 100 remoteloader:idm-4.9.0-
580
```

5 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it rl-container bash
```

6 リモートローダを設定します。詳細については、『*NetIQ Identity Manager Driver Administration Guide*』の「Configuring the Remote Loader and Drivers」を参照してください。

7 設定ファイルが共有ボリュームで利用できることを確認します。

ファンアウトエージェントコンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_fanoutagent.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=foa-
container -v /data:/config --stop-timeout 100 fanoutagent:idm-4.9.0-580
```

4 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it foa-container bash
```

5 ファンアウトエージェントを設定します。詳細については、『*NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/netiq-identity-manager-for-jdbc-fan-out-driver-implementation-guide.html)*』の「Configuring the Fanout Agent (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/how-to-configure-fan-out-agent.html)」を参照してください。

アイデンティティボールト認証局での証明書の生成

(状況によって実行) このセクションは、認証局としてアイデンティティボールトを使用している場合にのみ適用されます。

次のコンポーネントは、証明書を生成してから、展開する必要があります。次のコンポーネントの証明書を生成する前に、Identity Manager エンジンおよび Identity Console コンテナを展開していることを確認します。

- OSP
- 識別情報アプリケーション
- Identity Reporting

証明書を生成するには、次の手順を実行します。

1 インストールされている Java バージョンが Azul Zulu 1.8.0_392 以降であることを確認します。

2 Java パスを設定していることを確認します。たとえば、次のコマンドを実行します。

```
export PATH=<java installed location>/bin:$PATH
```

次に例を示します。

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

3 PKCS キーストアを生成します。

```
keytool -genkey -alias idm -keyalg RSA -storetype pkcs12 -keystore /
data/tomcat.ks -validity 3650 -keysize 2048 -dname
"CN=identitymanager.example.com" -keypass <password> -storepass
<password>
```

4 証明書署名要求を生成します。

```
keytool -certreq -v -alias idm -file /config/idm.csr -keypass
<password> -keystore /config/tomcat.ks -storepass <password>
```

5 自己署名証明書を生成します。

5a Identity Console を起動してログインします。

5b **[Certificate Management (証明書管理)]** > **[Issue Certificate (証明書の発行)]** の順に移動します。

5c 手順 3 で作成した `.csr` ファイルを参照します。たとえば、`idm.csr` です。

5d **[開く]** をクリックします。

5e [Key Usage Specifications (キーの使用法の指定)] ドロップダウンで、キータイプを指定します。

5f [Extended Key Usage Specifications (拡張キーの使用法の指定)] ドロップダウンで、拡張キータイプを指定します。

5g 証明書タイプには、[Basic Constrains (基本制約)] ドロップダウンで [Unspecified (未指定)] を選択します。

5h [Certificate Parameters (証明書パラメータ)] ドロップダウンで証明書の有効期間を指定します。

5i [Certificate format (証明書形式)] ドロップダウンで、[File in binary DER format (バイナリ DER 形式のファイル)] を選択します。

5j [次へ] をクリックします。

5k [OK] をクリックします。

5l 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

6 .der 形式で root 証明書をエクスポートします。

6a Identity Console を起動してログインします。

6b [Certificate Management (証明書管理)] > [Server Certificate Management (サーバ証明書管理)] に移動します。

6c [SSL CertificateDNS (SSL 証明書 DNS)] チェックボックスをオンにして、[エクスポート] をクリックします。

6d [証明書] ドロップダウンリストで、組織の CA を選択します。

6e [エクスポート形式] ドロップダウンリストで、DER を選択します。

6f [OK] をクリックします。

6g 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

7 証明書を手順 2 で作成した PKCS キーストアにインポートします。

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat.ks -
file /config/cert.der -storepass <password> -noprompt

keytool -import -alias idm -keystore /config/tomcat.ks -file /config/
idm.der -storepass <password> -noprompt
```

注 : キーストアが展開用の入力として指定されたパスで使用可能であることを確認します。

OSP コンテナの展開

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 SSO_SERVER_SSL_PORT プロパティが固有のポートに設定されていることを確認します。

3 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

4 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_osp.tar.gz
```

5 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=osp-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 osp:idm-4.9.0-580
```

6 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/osp/log/idmconfigure.log
```

7 次のコマンドを使用してコンテナを停止します。

```
docker stop osp-container
```

8 次のコマンドを実行して、server.xml ファイルの Tomcat シャットダウンポートを変更します。次の例では、ポート 8005 が 18005 に変更されます。

```
sed -i "s~8005~18005~g" /data/osp/tomcat/conf/server.xml
```

9 次のコマンドを使用してコンテナを起動します。

```
docker start osp-container
```

10 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it osp-container bash
```

PostgreSQL コンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_postgres.tar.gz
```

3 共有ボリューム /data の下にサブディレクトリ、たとえば、postgres を作成します。

```
mkdir postgres
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=<password> -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.18
```

次に例を示します。

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=novell -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.18
```

5 Identity Applications 用の idmdamin ユーザを作成します。

```
docker exec -it postgresql-container psql -U postgres -c "CREATE USER
idmadmin WITH ENCRYPTED PASSWORD '<password>'"
```

6 Identity Applications、ワークフロー、および Identity Reporting のデータベースを作成します。

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmuserappdb"

docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE igaworkflowdb"

docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmrptdb"
```

注 : これらのデータベースは、Identity Applications および Identity Reporting コンテナの設定中に使用されます。

7 idmadmin ユーザにデータベースに関するすべての権限を付与します。

```
docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE idmuserappdb TO idmadmin"

docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE igaworkflowdb TO idmadmin"
```

8 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it postgresql-container bash
```

Identity Applications コンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityapplication.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=idapps-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 identityapplication:idm-4.9.0-580
```

4 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/userapp/log/idmconfigure.log
```

5 次のコマンドを実行してキーストアをインポートします。

注 : この手順を実行する前に、コンテナが正常に展開されていることを確認します。

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/apps/
tomcat/conf/idm.jks -deststorepass <password>
```

6 エイリアス idm の既存のエントリを上書きするかどうかを尋ねるプロンプトが表示されます。上書きするには、「yes」と入力します。

7 エイリアス root の既存のエントリを上書きするかどうかを尋ねるプロンプトが表示されます。上書きするには、「yes」と入力します。

8 次のコマンドを使用してコンテナを停止します。

```
docker stop idapps-container
```

9 次のコマンドを実行して、server.xml ファイルの Tomcat シャットダウンポートを変更します。次の例では、ポート 8005 が 28005 に変更されます。

```
sed -i "s~8005~28005~g" /data/userapp/tomcat/conf/server.xml
```

10 次のコマンドで、Docker コンテナを起動します。

```
docker start idapps-container
```

11 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it idapps-container bash
```

注 : 設定更新ユーティリティの設定を変更するには、Identity Applications コンテナの /opt/netiq/idm/apps/configupdate/ ディレクトリから configupdate.sh を起動します。設定更新ユーティリティはコンソールモードでのみ起動できます。

フォームレンダラコンテナの展開

1 サイレントプロパティファイルを生成します。サイレントプロパティファイルの生成中に [Identity Applications] を選択します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_formrenderer.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=fr-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 formrenderer:idm-4.9.0-580
```

5 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it fr-container bash
```

ActiveMQ コンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_activemq.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=amq-
container -v /data:/config --env-file /data/silent.properties --stop-
timeout 100 activemq:idm-4.9.0-580
```

4 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it amq-container bash
```

5 ActiveMQ を設定します。詳細については、『NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/how-to-setup-activemq-startup-service.html)』の「*Setting Up ActiveMQ Startup Service (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/netiq-identity-manager-for-jdbc-fan-out-driver-implementation-guide.html)*」を参照してください。

Identity Reporting コンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityreporting.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=rpt-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 identityreporting:idm-4.9.0-580
```

4 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/reporting/log/idmconfigure.log
```

5 次のコマンドを実行してキーストアをインポートします。

注 : この手順を実行する前に、コンテナが正常に展開されていることを確認します。

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/apps/
tomcat/conf/idm.jks -deststorepass <password>
```

6 エイリアス idm の既存のエントリを上書きするかどうかを尋ねるプロンプトが表示されます。上書きするには、「yes」と入力します。

7 エイリアス root の既存のエントリを上書きするかどうかを尋ねるプロンプトが表示されます。上書きするには、「yes」と入力します。

8 次のコマンドを使用してコンテナを停止します。

```
docker stop rpt-container
```

9 次のコマンドを実行して、server.xml ファイルの Tomcat シャットダウンポートを変更します。次の例では、ポート 8005 が 38005 に変更されます。

```
sed -i "s~8005~38005~g" /data/reporting/tomcat/conf/server.xml
```

10 (状況によって実行) 認証局としてアイデンティティボールトを使用している場合にのみ適用されます。

-Dcom.sun.net.ssl.checkRevocation=false パラメータを setenv.sh ファイルの export CATALINA_OPTS エントリに追加します。この例では、setenv.sh ファイルは /data/reporting/tomcat/bin/ ディレクトリにあります。

11 次のコマンドで、Docker コンテナを起動します。

```
docker start rpt-container
```

12 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it rpt-container bash
```

SSPR コンテナの展開

以下のタスクを実行して、SSPR コンテナを展開します。

1 SSPR のサイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 共有ボリューム /data の下にサブディレクトリ、たとえば、sspr を作成します。

```
mkdir sspr
```

3 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

4 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_sspr.tar.gz
```

5 次のサンプルコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --network=host --name=sspr-container -v /data/sspr:/config --stop-timeout 100 sspr/sspr-webapp:4.7.0.1-580
```

6 Docker ホストから SSPR コンテナに `silent.properties` ファイルをコピーするには、Docker ホストから次のコマンドを実行します。

```
docker cp /data/silent.properties sspr-container:/tmp
```

7 サイレントプロパティファイルを SSPR コンテナにロードします。

```
docker exec -it sspr-container /app/command.sh ImportPropertyConfig /tmp/silent.properties
```

注 : `SSPRConfiguration.xml` が、SSPR コンテナの `/config` ディレクトリに作成されているかどうかを確認し、ファイルの内容を確認します。

---

8 OAuth 証明書を SSPR にインポートします。

8a Docker ホストから、`/data/sspr/` ディレクトリにある `SSPRConfiguration.xml` ファイルを編集し、`configIsEditable` フラグの値を [true] に設定して、変更内容を保存します。

8b ブラウザを起動し、`https://identitymanager.example.com:8443/sspr` URL を入力します。

8c 管理者資格情報、たとえば、`uaadmin` を使用してログインします。

8d 右上隅のユーザ、たとえば、`uaadmin` をクリックし、[Configuration Editor (環境設定エディタ)] をクリックします。

8e 設定パスワードを指定し、[Sign In (サインイン)] をクリックします。

8f [設定] > [Single Sign On (SSO) Client (シングルサインオン (SSO) クライアント)] > [OAuth] の順にクリックして、すべての URL が HTTPS プロトコルと正しいポートを使用していることを確認します。

8g [OAuth Server Certificate (OAuth サーバ証明書)] の下で、[Import From Server (サーバからインポート)] をクリックして、新しい証明書をインポートし、[OK] をクリックします。

8h 右上隅の  をクリックして、証明書を保存します。

8i 変更を確認し、[OK] をクリックします。

8j SSPR アプリケーションが再起動した後で、`SSPRConfiguration.xml` ファイルを編集し、`configIsEditable` フラグの値を [false] に設定して、変更内容を保存します。

Identity Console コンテナの展開

Identity Console コンテナの展開の詳細については、「Docker コンテナとしての Identity Console の展開」を参照してください。

分散サーバ上へのコンテナの展開

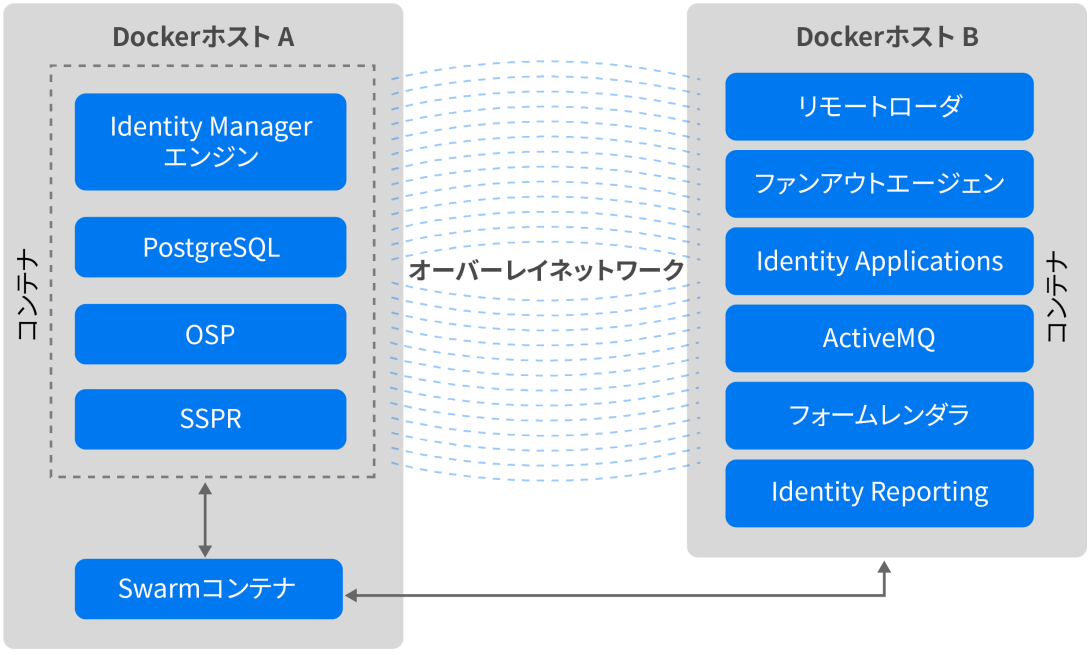
オーバーレイネットワークの設定

NetIQ では、オーバーレイネットワークモードまたはブリッジネットワークモードを使用して、すべての Identity Manager コンテナを分散セットアップで展開することをお勧めします。このガイドに記載されているシナリオは、オーバーレイネットワークでのコンテナの展開に固有のものです。ただし、ブリッジネットワークを使用してコンテナを展開することもできます。以下の分散サーバシナリオでは、Identity Manager エンジン、PostgreSQL、OSP、および SSPR コンテナを Docker ホスト A に展開します。Docker ホスト B には、リモートローダ、ファンアウトエージェント、Identity Applications、ActiveMQ、フォームレンダラ、および Identity Reporting コンテナを展開します。

以下の図は、Identity Manager コンテナをオーバーレイネットワーク内の 2 つの Docker ホストに展開した場合を示しています。

注 : クラスタストアはサポートされなくなったため、Docker swarm を使用して、オーバーレイネットワークを設定できます。非推奨化の詳細については、クラスタストア (https://docs.docker.com/engine/deprecated/#classic-swarm-and-overlay-networks-using-cluster-store) についての説明を確認してください。

***図 18-1** オーバーレイネットワークでのコンテナ展開アーキテクチャ*



オーバーレイネットワークを設定するには、次の手順を実行します。

1 Docker ホスト A で、次のコマンドを実行します。

```
docker swarm init
```

2 手順 1 の出力から生成される join コマンドをメモし、この join コマンドを Docker ホスト B で実行して、これらの 2 つのホスト間に swarm ネットワークを作成します。

例 :

```
docker swarm join --token SWMTKN-1-
24vozonapa4anwszd3ykthgomo84e9y3p3vbcrtibz6jzo94k2-
148g7yrnepmz963x3z5udnae4 10.71.142.191:2377
```

注 : ワーカノードに参加するためのコマンドを見逃した場合は、次のコマンドを実行して join コマンドを取得します。

```
docker swarm join-token worker
```

3 次のコマンドを実行して、すべてのホストのステータスを確認します。

```
docker node ls
```

4 次のコマンドを使用して、オーバーレイネットワークを Docker ホスト B 上に作成します。

```
docker network create --driver overlay --subnet=<subnet-cidr> --
gateway=<gateway-ip> --attachable <network-name>
```

例 :

```
docker network create --driver overlay --subnet=192.168.0.0/24 --
gateway=192.168.0.1 --attachable idmoverlaynetwork
```

5 次のコマンドを実行して、オーバーレイネットワークが作成されているかどうかを確認します。

```
docker network ls
```

注 : swarm は動的に IP を割り当てるため、コンテナ展開時に、swarm ロードバランサが、どの Identity Manager コンポーネントでも使用されていない新しい IP アドレスを使用していることを確認します。そうしないと、次のエラーが発生する可能性があります。

```
Error response from daemon: attaching to network failed, make sure your
network options are correct and check manager logs: context deadline
exceeded
```

次のコマンドを実行して、swarm ロードバランサの IP を確認します。

```
docker inspect idmoverlaynetwork | grep lb-idmoverlaynetwork -A 7
```

前提条件

- Docker 展開におけるすべての Docker ホストの `/etc/hosts` ファイルは、そのホスト上で実行されているすべてのコンテナの詳細で更新される必要があります。すべてのコンテナのホスト名が完全修飾ドメイン名 (FQDN) 形式のみであることを確認してください。

  ホストファイルのエントリは、すべてのコンポーネントについて次の形式に従うことができます。

```
<IP of the container> <FQDN> <short_name>
```

このガイドで使用されるサンプル展開では、/etc/hosts ファイルに次のエントリを追加します。

```
192.168.0.12     identityengine.example.com       identityengine
192.168.0.2      remoteloader.example.com         remoteloader
192.168.0.3      fanoutagent.example.com          fanoutagent
192.168.0.5      osp.example.com                  osp
192.168.0.6      postgresql.example.com           postgresql
192.168.0.7      identityapps.example.com         identityapps
192.168.0.8      formrenderer.example.com         formrenderer
192.168.0.9      activemq.example.com             activemq
192.168.0.10     identityreporting.example.com    identityreporting
192.168.0.11     sspr.example.com                 sspr
```

- コンテナが起動されるたびに使用できるように、サードパーティの jar ファイルがボリュームマウントされていることを確認します。たとえば、ojdbc.jar がコンテナの /opt/netiq/idm/apps/tomcat/lib ディレクトリに存在する場合は、次のようなサンプルコマンドを使用して jar ファイルをボリュームマウントする必要があります。

  ```
  -v /host/ojdbc.jar:/opt/netiq/idm/apps/tomcat/lib/ojdbc.jar
  ```

- コンテナを展開する前に、サイレントプロパティファイルを生成する必要があります。サイレントプロパティファイルの生成に関する詳細については、254 ページの 「サイレントプロパティファイルの作成」を参照してください。

コンテナがアクセスするポートの公開

前提条件として、コンテナに使用するポートを知っておく必要があります。必要なポートを公開し、コンテナポートを Docker ホスト上のポートにマップする必要があります。次のテーブルでは、このガイドで提供される例に基づいて Docker ホスト上で公開する必要があるポートに関する情報を提供します。

表 18-2 展開例で公開されるデフォルトのポート

コンテナ	展開例に従って想定されるデフォルトのポート
リモートローダ	8090
ファンアウトエージェント	対象外。
OSP	8543
識別情報アプリケーション	18543
Identity Reporting	28543
フォームレンダラ	8600
ActiveMQ	• 8161 • 61616
PostgreSQL	5432
SSPR	8443 **注 :** SSPR コンテナは 8443 ポートでのみ実行します。

ただし、環境に基づいてポートをカスタマイズできます。ポートの公開時には、次の考慮事項が適用されます。

- 必ず使用していないポートを公開してください。
- コンテナポートは Docker ホスト上の同じポートにマップされる必要があります。たとえば、コンテナ上の 8543 ポートは Docker ホスト上の 8543 ポートにマップされる必要があります。

コンテナは次の順序で展開される必要があります。

- 273 ページの 「Identity Manager エンジンコンテナの展開」
- 273 ページの 「リモートローダコンテナの展開」
- 274 ページの 「ファンアウトエージェントコンテナの展開」
- 279 ページの 「OSP コンテナの展開」
- 280 ページの 「PostgreSQL コンテナの展開」
- 281 ページの 「Identity Applications コンテナの展開」
- 282 ページの 「フォームレンダラコンテナの展開」
- 283 ページの 「ActiveMQ コンテナの展開」
- 283 ページの 「Identity Reporting コンテナの展開」
- 284 ページの 「SSPR コンテナの展開」

Identity Manager エンジンコンテナの展開

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityengine.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.12 --
network=idmoverlaynetwork --hostname=identityengine.example.com --
name=engine-container -v /etc/hosts:/etc/hosts -v /data:/config -p
8028:8028 -p 524:524 -p 389:389 -p 8030:8030 -p 636:636 -e
SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100
identityengine:idm-4.9.0-580
```

5 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/idm/log/idmconfigure.log
```

6 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it engine-container bash
```

注 : ndstrace や ndsrepair などのアイデンティティボールトユーティリティを実行するには、nds と呼ばれるルート以外のユーザとしてコンテナにログインします。ルートユーザとしてログインしている場合にはこれらのユーティリティは実行できません。nds ユーザとしてコンテナにログインするには、docker exec -it engine-container sudo nds コマンドを実行します。

リモートローダコンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_remoteloader.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run -d --ip=192.168.0.2 --network=idmoverlaynetwork --
hostname=remoteloader.example.com -p 8090:8090 --name=rl-container -v /
etc/hosts:/etc/hosts -v /data:/config remoteloader:idm-4.9.0-580
```

これにより、64 ビットバージョンおよび 32 ビットバージョンのリモートローダが展開されます。ドライバファイルはコンテナの /opt/novell/eDirectory/lib/dirxml/classes/ ディレクトリにあります。

4 (状況によって実行) コンテナの展開時に設定ファイルを使用しない場合は、次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.2 --
network=idmoverlaynetwork --hostname=remoteloader.example.com -p
8090:8090 --name=rl-container -v /etc/hosts:/etc/hosts -v /data:/config
--stop-timeout 100 remoteloader:idm-4.9.0-580
```

5 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it rl-container bash
```

6 リモートローダを設定します。詳細については、『*NetIQ Identity Manager Driver Administration Guide*』の「Configuring the Remote Loader and Drivers」を参照してください。

ファンアウトエージェントコンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_fanoutagent.tar.gz
```

3 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.3 --
network=idmoverlaynetwork --hostname=fanoutagent.example.com --
name=foa-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-
timeout 100 fanoutagent:idm-4.9.0-580
```

4 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it foa-container bash
```

5 ファンアウトエージェントを設定します。詳細については、『NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/netiq-identity-manager-for-jdbc-fan-out-driver-implementation-guide.html)』の「Configuring the Fanout Agent (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/how-to-configure-fan-out-agent.html)」を参照してください。

アイデンティティボールト認証局での証明書の生成

(状況によって実行) このセクションは、認証局としてアイデンティティボールトを使用している場合にのみ適用されます。

次のコンポーネントは、証明書を生成してから、展開する必要があります。次のコンポーネントの証明書を生成する前に、Identity Manager エンジンコンテナを展開していることを確認します。

- OSP
- 識別情報アプリケーション
- Identity Reporting

OSP の証明書の生成

証明書を生成するには、次の手順を実行します。

1 インストールされている Java バージョンが Azul Zulu 1.8.0_392 以降であることを確認します。

2 Docker ホストに Java パスを設定していることを確認します。たとえば、次のコマンドを実行します。

```
export PATH=<java installed location>/bin:$PATH
```

次に例を示します。

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

3 PKCS キーストアを生成します。

```
keytool -genkey -alias osp -keyalg RSA -storetype pkcs12 -keystore /
data/tomcat-osp.ks -validity 3650 -keysize 2048 -dname
"CN=osp.example.com" -keypass <password> -storepass <password>
```

4 証明書署名要求を生成します。

```
keytool -certreq -v -alias osp -file /config/osp.csr -keypass
<password> -keystore /config/tomcat-osp.ks -storepass <password>
```

5 自己署名証明書を生成します。

5a Identity Console を起動してログインします。

5b **[Certificate Management (証明書管理)]** > **[Issue Certificate (証明書の発行)]** の順に移動します。

5c 手順 3 で作成した `.csr` ファイルを参照します。たとえば、`osp.csr` というファイルを参照します。

5d **[開く]** をクリックします。

5e **[Key Usage Specifications (キーの使用法の指定)]** ドロップダウンで、キータイプを指定します。

5f **[Extended Key Usage Specifications (拡張キーの使用法の指定)]** ドロップダウンで、拡張キータイプを指定します。

5g 証明書タイプには、[Basic Constrains (基本制約)] ドロップダウンで [Unspecified (未指定)] を選択します。

5h [Certificate Parameters (証明書パラメータ)] ドロップダウンで証明書の有効期間を指定します。

5i [Certificate format (証明書形式)] ドロップダウンで、[File in binary DER format (バイナリ DER 形式のファイル)] を選択します。

5j [次へ] をクリックします。

5k [OK] をクリックします。

5l 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

6 .der 形式で root 証明書をエクスポートします。

6a Identity Console を起動してログインします。

6b [Certificate Management (証明書管理)] > [Server Certificate Management (サーバ証明書管理)] に移動します。

6c [SSL CertificateDNS (SSL 証明書 DNS)] チェックボックスをオンにして、[エクスポート] をクリックします。

6d [証明書] ドロップダウンリストで、組織の CA を選択します。

6e [エクスポート形式] ドロップダウンリストで、DER を選択します。

6f [OK] をクリックします。

6g 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

7 証明書を手順 2 で作成した PKCS キーストアにインポートします。

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat-osp.ks -file /config/cert.der -storepass <password> -noprompt
```

```
keytool -import -alias osp -keystore /config/tomcat-osp.ks -file /config/osp.der -storepass <password> -noprompt
```

注 : キーストアが展開用の入力として指定されたパスで使用可能であることを確認します。

Identity Applications の証明書の生成

証明書を生成するには、次の手順を実行します。

1 インストールされている Java バージョンが Azul Zulu 1.8.0_392 以降であることを確認します。

2 Java パスを設定していることを確認します。たとえば、次のコマンドを実行します。

```
export PATH=<java installed location>/bin:$PATH
```

次に例を示します。

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

3 PKCS キーストアを生成します。

```
keytool -genkey -alias ua -keyalg RSA -storetype pkcs12 -keystore /
config/tomcat-ua.ks -validity 3650 -keysize 2048 -dname
"CN=identityapps.example.com" -keypass <password> -storepass <password>
```

4 証明書署名要求を生成します。

```
keytool -certreq -v -alias ua -file /config/ua.csr -keypass <password>
-keystore /config/tomcat-ua.ks -storepass <password>
```

5 自己署名証明書を生成します。

5a Identity Console を起動してログインします。

5b **[Certificate Management (証明書管理)]** > **[Issue Certificate (証明書の発行)]** の順に移動します。

5c 手順 3 で作成した .csr ファイルを参照します。たとえば、ua.csr というファイルを参照します。

5d **[開く]** をクリックします。

5e **[Key Usage Specifications (キーの使用法の指定)]** ドロップダウンで、キータイプを指定します。

5f **[Extended Key Usage Specifications (拡張キーの使用法の指定)]** ドロップダウンで、拡張キータイプを指定します。

5g 証明書タイプには、**[Basic Constrains (基本制約)]** ドロップダウンで **[Unspecified (未指定)]** を選択します。

5h **[Certificate Parameters (証明書パラメータ)]** ドロップダウンで証明書の有効期間を指定します。

5i **[Certificate format (証明書形式)]** ドロップダウンで、**[File in binary DER format (バイナリ DER 形式のファイル)]** を選択します。

5j **[次へ]** をクリックします。

5k **[OK]** をクリックします。

5l 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

6 .der 形式で root 証明書をエクスポートします。

6a Identity Console を起動してログインします。

6b **[Certificate Management (証明書管理)]** > **[Server Certificate Management (サーバ証明書管理)]** に移動します。

6c **[SSL CertificateDNS (SSL 証明書 DNS)]** チェックボックスをオンにして、**[エクスポート]** をクリックします。

6d **[証明書]** ドロップダウンリストで、組織の CA を選択します。

6e **[エクスポート形式]** ドロップダウンリストで、DER を選択します。

6f **[OK]** をクリックします。

6g 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

7 証明書を手順 2 の PKCS キーストアにインポートします。

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat-
ua.ks -file /config/cert.der -storepass <password> -noprompt

keytool -import -alias ua -keystore /config/tomcat-ua.ks -file /config/
ua.der -storepass <password> -noprompt
```

注 : 証明書が展開用の入力として指定されたパスで利用可能であることを確認します。

Identity Reporting の証明書の生成

証明書を生成するには、次の手順を実行します。

1. インストールされている Java バージョンが Azul Zulu 1.8.0_392 以降であることを確認します。
2. Java パスを設定していることを確認します。たとえば、次のコマンドを実行します。

   ```
   export PATH=<java installed location>/bin:$PATH
   ```

   次に例を示します。

   ```
   export PATH=/opt/netiq/common/jre/bin/:$PATH
   ```

3. PKCS キーストアを生成します。

   ```
   keytool -genkey -alias rpt -keyalg RSA -storetype pkcs12 -keystore /
   config/tomcat-rpt.ks -validity 3650 -keysize 2048 -dname
   "CN=identityreporting.example.com" -keypass <password> -storepass
   <password>
   ```

4. 証明書署名要求を生成します。

   ```
   keytool -certreq -v -alias rpt -file /config/rpt.csr -keypass
   <password> -keystore /config/tomcat-rpt.ks -storepass <password>
   ```

5. 自己署名証明書を生成します。
   - **5a** Identity Console を起動してログインします。
   - **5b** **[Certificate Management (証明書管理)]** > **[Issue Certificate (証明書の発行)]** の順に移動します。
   - **5c** 手順 3 で作成した `.csr` ファイルを参照します。たとえば、`rpt.csr` というファイルを参照します。
   - **5d** **[開く]** をクリックします。
   - **5e** **[Key Usage Specifications (キーの使用法の指定)]** ドロップダウンで、キータイプを指定します。
   - **5f** **[Extended Key Usage Specifications (拡張キーの使用法の指定)]** ドロップダウンで、拡張キータイプを指定します。
   - **5g** 証明書タイプには、**[Basic Constrains (基本制約)]** ドロップダウンで **[Unspecified (未指定)]** を選択します。
   - **5h** **[Certificate Parameters (証明書パラメータ)]** ドロップダウンで証明書の有効期間を指定します。

5i [Certificate format (証明書形式)] ドロップダウンで、[File in binary DER format (バイナリ DER 形式のファイル)] を選択します。

5j [次へ] をクリックします。

5k [OK] をクリックします。

5l 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

6 .der 形式で root 証明書をエクスポートします。

6a Identity Console を起動してログインします。

6b [Certificate Management (証明書管理)] > [Server Certificate Management (サーバ証明書管理)] に移動します。

6c [SSL CertificateDNS (SSL 証明書 DNS)] チェックボックスをオンにして、[エクスポート] をクリックします。

6d [証明書] ドロップダウンリストで、組織の CA を選択します。

6e [エクスポート形式] ドロップダウンリストで、DER を選択します。

6f [OK] をクリックします。

6g 証明書をダウンロードし、ダウンロードされた証明書を /data ディレクトリにコピーします。

7 証明書を手順 2 で作成した PKCS キーストアにインポートします。

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat-rpt.ks -file /config/cert.der -storepass <password> -noprompt
```

```
keytool -import -alias rpt -keystore /config/tomcat-rpt.ks -file /config/rpt.der -storepass <password> -noprompt
```

注 : 証明書が展開用の入力として指定されたパスで利用可能であることを確認します。

OSP コンテナの展開

注 : OSP コンテナを展開する前に、必要な証明書を生成してください。詳細については、OSP の証明書の生成を参照してください。

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_osp.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.5 --
network=idmoverlaynetwork --hostname=osp.example.com -p 8543:8543 --
name=osp-container -v /etc/hosts:/etc/hosts -v /data:/config -e
SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100
osp:idm-4.9.0-580
```

5 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/osp/log/idmconfigure.log
```

6 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it osp-container bash
```

PostgreSQL コンテナの展開

1 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

2 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_postgres.tar.gz
```

3 共有ボリューム /data の下にサブディレクトリ、たとえば、postgres を作成します。

```
mkdir postgres
```

4 次のサンプルコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=<password>
-v /data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v
/data:/config --stop-timeout 100 postgres:12.18
```

次に例を示します。

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=novell -v /
data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v /
data:/config --stop-timeout 100 postgres:12.18
```

5 Identity Applications 用の idmdamin ユーザを作成します。

```
docker exec -it postgresql-container psql -U postgres -c "CREATE USER
idmadmin WITH ENCRYPTED PASSWORD '<password>'"
```

6 Identity Applications、ワークフロー、および Identity Reporting のデータベースを作成します。

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmuserappdb"
```

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE igaworkflowdb"

docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmrptdb"
```

注 : これらのデータベースは、Identity Applications および Identity Reporting コンテナの設定中に使用されます。

7 `idmadmin` ユーザにデータベースに関するすべての権限を付与します。

```
docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE idmuserappdb TO idmadmin"

docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE igaworkflowdb TO idmadmin"
```

8 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it postgresql-container bash
```

Identity Applications コンテナの展開

注 : Identity Applications コンテナを展開する前に、必要な証明書を生成してください。詳細については、Identity Applications の証明書の生成を参照してください。

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

注 : アプリケーションサーバポートの値として、公開されているポート 18543 を指定します。

2 `Identity_Manager_4.9.0_Containers.tar.gz file` を抽出した場所から、`Identity_Manager_4.9.0_Containers` ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityapplication.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.7 --
network=idmoverlaynetwork --hostname=identityapps.example.com -p
18543:18543 --name=idapps-container -v /etc/hosts:/etc/hosts -v /data:/
config -e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout
100 identityapplication:idm-4.9.0-580
```

5 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/userapp/log/idmconfigure.log
```

6 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it idapps-container bash
```

7 次のコマンドを実行します。

注 : この手順を実行する前に、コンテナが正常に展開されていることを確認します。

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat-osp.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/
apps/tomcat/conf/idm.jks -deststorepass <password>
```

8 root エイリアスのエントリを上書きするには、yes と入力します。

9 Identity Applications コンテナを再起動します。

```
docker restart idapps-container
```

注 : 設定更新ユーティリティの設定を変更するには、Identity Applications コンテナの /opt/netiq/idm/apps/configupdate/ ディレクトリから configupdate.sh を起動します。設定更新ユーティリティはコンソールモードでのみ起動できます。

フォームレンダラコンテナの展開

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 Identity_Manager_4.9.0_Containers.tar.gz file を抽出した場所から、Identity_Manager_4.9.0_Containers ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_formrenderer.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.8 --
network=idmoverlaynetwork --hostname=formrenderer.example.com -p
8600:8600 --name=fr-container -v /etc/hosts:/etc/hosts -v /data:/config
-e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100
formrenderer:idm-4.9.0-580
```

5 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it fr-container bash
```

ActiveMQ コンテナの展開

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 `Identity_Manager_4.9.0_Containers.tar.gz file` を抽出した場所から、`Identity_Manager_4.9.0_Containers` ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_activemq.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.9 --
network=idmoverlaynetwork --hostname=activemq.example.com -p
61616:61616 --name=amq-container -v /etc/hosts:/etc/hosts -v /data:/
config --env-file /data/silent.properties --stop-timeout 100
activemq:idm-4.9.0-580
```

5 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it amq-container bash
```

6 ActiveMQ を設定します。詳細については、『NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/how-to-setup-activemq-startup-service.html)』の「*Setting Up ActiveMQ Startup Service (https://www.netiq.com/documentation/identity-manager-49-drivers/jdbc_fanout/data/netiq-identity-manager-for-jdbc-fan-out-driver-implementation-guide.html)*」を参照してください。

Identity Reporting コンテナの展開

注 : Identity Reporting コンテナを展開する前に、必要な証明書を生成してください。詳細については、Identity Reporting の証明書の生成を参照してください。

---

1 サイレントプロパティファイルを生成します。詳細については、サイレントプロパティファイルの作成を参照してください。

注 : アプリケーションサーバポートの値として、公開されているポート 28543 を指定します。

---

2 `Identity_Manager_4.9.0_Containers.tar.gz file` を抽出した場所から、`Identity_Manager_4.9_Containers` ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityreporting.tar.gz
```

4 次のコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.10 --
network=idmoverlaynetwork --hostname=identityreporting.example.com -p
28543:28543 --name=rpt-container -v /etc/hosts:/etc/hosts -v /data:/
config -e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout
100 identityreporting:idm-4.9.0-580
```

5 コンテナが正常に展開されたかどうかを確認するには、次のコマンドを実行してログファイルを確認します。

```
tail -f /data/reporting/log/idmconfigure.log
```

6 コンテナにログインするには、次のコマンドを実行します。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it rpt-container bash
```

7 次のコマンドを実行します。

注 : この手順を実行する前に、コンテナが正常に展開されていることを確認します。

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat-osp.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/
apps/tomcat/conf/idm.jks -deststorepass <password>
```

8 `root` エイリアスのエントリを上書きするには、`yes` と入力します。

9 Identity Reporting コンテナを再起動します。

```
docker restart rpt-container
```

SSPR コンテナの展開

以下のタスクを実行して、SSPR コンテナを展開します。

1 SSPR のサイレントプロパティファイルを生成します。サイレントプロパティファイルの生成中に **[Identity Applications]** を選択します。詳細については、サイレントプロパティファイルの作成を参照してください。

2 共有ボリューム `/data` の下にサブディレクトリ、たとえば、`sspr` を作成します。

```
mkdir sspr
```

3 `Identity_Manager_4.9.0_Containers.tar.gz file` を抽出した場所から、`Identity_Manager_4.9.0_Containers` ディレクトリに移動します。

4 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_sspr.tar.gz
```

5 次のサンプルコマンドを使用してコンテナを展開します。

```
docker run --restart unless-stopped -d --ip=192.168.0.11 --
network=idmoverlaynetwork --hostname=sspr.example.com --name=sspr-
container -v /etc/hosts:/etc/hosts -v /data/sspr:/config -p 8443:8443 -
-stop-timeout 100 sspr/sspr-webapp:4.7.0.1-580
```

6 Docker ホストから SSPR コンテナに `silent.properties` ファイルをコピーするには、Docker ホストから次のコマンドを実行します。

```
docker cp /data/silent.properties sspr-container:/tmp
```

7 サイレントプロパティファイルを SSPR コンテナにロードします。

```
docker exec -it sspr-container /app/command.sh ImportPropertyConfig /
tmp/silent.properties
```

注 : `SSPRConfiguration.xml` が、SSPR コンテナの `/config` ディレクトリに作成されているかどうかを確認し、ファイルの内容を確認します。

---

8 OAuth 証明書を SSPR にインポートします。

8a Docker ホストから、`/data/sspr/` ディレクトリにある `SSPRConfiguration.xml` ファイルを編集し、`configIsEditable` フラグの値を **[true]** に設定して、変更内容を保存します。

8b ブラウザを起動し、`https://sspr.example.com:8443/sspr` URL を入力します。

8c 管理者資格情報、たとえば、`uaadmin` を使用してログインします。

8d 右上隅のユーザ、たとえば、`uaadmin` をクリックし、**[Configuration Editor (環境設定エディタ)]** をクリックします。

8e 設定パスワードを指定し、**[Sign In (サインイン)]** をクリックします。

8f **[設定]** > **[Single Sign On (SSO) Client (シングルサインオン (SSO) クライアント)]** > **[OAuth]** の順にクリックして、すべての URL が HTTPS プロトコルと正しいポートを使用していることを確認します。

8g **[OAuth Server Certificate (OAuth サーバ証明書)]** の下で、**[Import From Server (サーバからインポート)]** をクリックして、新しい証明書をインポートし、**[OK]** をクリックします。

8h 右上隅の  をクリックして、証明書を保存します。

8i 変更を確認し、**[OK]** をクリックします。

8j SSPR アプリケーションが再起動した後で、`SSPRConfiguration.xml` ファイルを編集し、`configIsEditable` フラグの値を **[false]** に設定して、変更内容を保存します。

Identity Console コンテナの展開

Identity Console コンテナの展開の詳細については、「Docker コンテナとしての Identity Console の展開」を参照してください。

19 Identity Manager コンテナのアップグレード

このセクションでは、Identity Manager の個々のコンテナを更新する方法について説明します。

コンテナを更新する手順については、以降のセクションで説明します。

- 287 ページの 「コンテナの更新の前提条件」
- 288 ページの 「分散サーバ上でのコンテナの更新」
- 293 ページの 「単一サーバ上でのコンテナの更新」

コンテナの更新の前提条件

Identity Manager の各コンテナを更新する前に、以下の手順を実行します。

重要 : このセクションは、PostgreSQL コンテナには適用されません。PostgreSQL コンテナの更新については、*288 ページの 「分散サーバ上でのコンテナの更新」*セクションの「PostgreSQL コンテナの更新」または *293 ページの 「単一サーバ上でのコンテナの更新」*セクションの「PostgreSQL コンテナの更新」を参照してください。

---

1 すべての Identity Manager コンテナを停止します。

```
docker stop <container name>
```

次に例を示します。

```
docker stop engine-container
```

2 共有ディレクトリのバックアップを取ります。このガイドの例では、`/data` を共有ディレクトリとして想定しています。

3 すべての Identity Manager コンテナを削除します。

```
docker rm <container name>
```

次に例を示します。

```
docker rm engine-container
```

4 (状況によって実行) 古い Docker イメージをすべて削除します。

```
docker rmi <image ID>
```

分散サーバ上でのコンテナの更新

コンテナは次の順序で更新する必要があります。

- 288 ページの 「Identity Manager エンジンコンテナの更新」
- 289 ページの 「リモートローダコンテナの更新」
- 289 ページの 「ファンアウトエージェントコンテナの更新」
- 290 ページの 「OSP コンテナの更新」
- 290 ページの 「PostgreSQL コンテナの更新」
- 291 ページの 「Identity Applications コンテナの更新」
- 292 ページの 「フォームレンダラコンテナの更新」
- 292 ページの 「ActiveMQ コンテナの更新」
- 292 ページの 「Identity Reporting コンテナの更新」
- 293 ページの 「SSPR コンテナの更新」
- 293 ページの 「Identity Console コンテナの更新」

Identity Manager エンジンコンテナの更新

1 共有ディレクトリ `/data` に次の内容で `credentials.properties` ファイルを作成します。

```
ID_VAULT_ADMIN_LDAP="<ID_VAULT_ADMIN_LDAP>"
ID_VAULT_PASSWORD="<ID_VAULT_PASSWORD>"
```

ここで、`ID_VAULT_ADMIN_LDAP` は、コンマ区切り形式である必要があります。

次に例を示します。

```
ID_VAULT_ADMIN_LDAP="cn=admin,ou=sa,o=system"
```

```
ID_VAULT_PASSWORD="novell"
```

2 `Identity_Manager_4.9.0_Containers.tar.gz` ファイルを抽出した場所に移動します。

3 `docker-images` ディレクトリに移動します。

4 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityengine.tar.gz
```

5 オーバーレイネットワークを使用して Identity Manager エンジンを展開する場合は、次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.12 --
network=idmoverlaynetwork --hostname=identityengine.example.com --
name=engine-container -v /etc/hosts:/etc/hosts -v /data:/config -p
8028:8028 -p 524:524 -p 389:389 -p 8030:8030 -p 636:636 -e
SILENT_INSTALL_FILE=/config/credentials.properties --stop-timeout 100
identityengine:idm-4.9.0-580
```

ホストネットワークを使用して Identity Manager エンジンを展開する場合は、次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=engine-container -v /etc/hosts:/etc/hosts -v /data:/config -e SILENT_INSTALL_FILE=/config/credentials.properties --stop-timeout 100 identityengine:idm-4.9.0-580
```

リモートローダコンテナの更新

1. Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。
2. docker-images ディレクトリに移動します。
3. 次のコマンドを実行して、イメージをロードします。

   ```
   docker load --input IDM_490_remoteloader.tar.gz
   ```

4. 次のコマンドを実行してコンテナを展開します。

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.2 --network=idmoverlaynetwork --hostname=remoteloader.example.com -p 8090:8090 --name=rl-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-timeout 100 remoteloader:idm-4.9.0-580
   ```

   ドライバファイルはコンテナの /opt/novell/eDirectory/lib/dirxml/classes/ ディレクトリにあります。
5. (状況によって実行) リモートローダインスタンスが実行されていない場合は、リモートローダインスタンスを起動します。

ファンアウトエージェントコンテナの更新

1. Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。
2. docker-images ディレクトリに移動します。
3. 次のコマンドを実行して、イメージをロードします。

   ```
   docker load --input IDM_490_fanoutagent.tar.gz
   ```

4. 次のコマンドを使用してコンテナを更新します。

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.3 --network=idmoverlaynetwork --hostname=fanoutagent.example.com --name=foa-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-timeout 100 fanoutagent:idm-4.9.0-580
   ```

5. ファンアウトエージェントを起動します。

OSP コンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_osp.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.5 --
network=idmoverlaynetwork --hostname=osp.example.com -p 8543:8543 --
name=osp-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-
timeout 100 osp:idm-4.9.0-580
```

5 次のコマンドを実行してコンテナにログインします。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it osp-container bash
```

PostgreSQL コンテナの更新

注 : PostgreSQL コンテナを更新する前に、Identity Applications や Identity Reporting などの依存コンテナを必ず停止してください。

1 Docker ホストで、任意の場所に移動します。例 :

```
cd /tmp
```

2 次のコマンドを実行して、既存の PostgreSQL コンテナのデータのバックアップを取ります。

```
docker exec postgresql-container pg_dumpall -U postgres > dump.sql
```

3 PostgreSQL コンテナを停止します。

```
docker stop <container name>
```

次に例を示します。

```
docker stop postgresql-container
```

4 PostgreSQL コンテナを削除します。

```
docker rm <container name>
```

5 既存の PostgreSQL データディレクトリを削除します。

```
rm -rf /data/postgres
```

6 (状況によって実行) PostgreSQL Docker イメージを削除します。

```
docker rmi <image ID>
```

7 共有ディレクトリ /data の下にサブディレクトリ、たとえば postgres を作成します。

```
mkdir postgres
```

8 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

9 docker-images ディレクトリに移動します。

10 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_postgres.tar.gz
```

11 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=<password>
-v /data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v
/data:/config --stop-timeout 100 postgres:12.18
```

次に例を示します。

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=novell -v /
data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v /
data:/config --stop-timeout 100 postgres:12.18
```

12 Docker ホスト上にバックアップしたデータファイル (ステップ 2) を新しい PostgreSQL data ディレクトリにコピーします。

```
cp /tmp/dump.sql /data/postgres
```

13 次のコマンドを実行してコンテナにログインします。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it postgresql-container bash
```

14 /var/lib/postgresql/data/ ディレクトリに移動します。

15 ステップ 2 でバックアップしたデータを新しい PostgreSQL コンテナに復元します。

```
psql -U postgres < dump.sql >
```

16 次のコマンドを実行して、コンテナを終了します。

```
exit
```

Identity Applications コンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityapplication.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.7 --
network=idmoverlaynetwork --hostname=identityapps.example.com -p
18543:18543 --name=idapps-container -v /etc/hosts:/etc/hosts -v /data:/
config --stop-timeout 100 identityapplication:idm-4.9.0-580
```

フォームレンダラコンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_formrenderer.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.8 --
network=idmoverlaynetwork --hostname=formrenderer.example.com -p
8600:8600 --name=fr-container -v /etc/hosts:/etc/hosts -v /data:/config
--stop-timeout 100 formrenderer:idm-4.9.0-580
```

ActiveMQ コンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_activemq.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.9 --
network=idmoverlaynetwork --hostname=activemq.example.com -p
61616:61616 --name=amq-container -v /etc/hosts:/etc/hosts -v /data:/
config --stop-timeout 100 activemq:idm-4.9.0-580
```

Identity Reporting コンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityreporting.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.10 --
network=idmoverlaynetwork --hostname=identityreporting.example.com -p
28543:28543 --name=rpt-container -v /etc/hosts:/etc/hosts -v /data:/
config --stop-timeout 100 identityreporting:idm-4.9.0-580
```

SSPR コンテナの更新

1 `Identity_Manager_4.9.0_Containers.tar.gz` ファイルを抽出した場所に移動します。

2 `docker-images` ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_sspr.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --ip=192.168.0.11 --
network=idmoverlaynetwork --hostname=sspr.example.com --name=sspr-
container -v /etc/hosts:/etc/hosts -v /data/sspr:/config -p 8443:8443 -
-stop-timeout 100 sspr/sspr-webapp:4.7.0.1-580
```

Identity Console コンテナの更新

Identity Console コンテナの更新の詳細については、「Docker コンテナとしての Identity Console のアップグレード」を参照してください。

単一サーバ上でのコンテナの更新

コンテナは次の順序で更新する必要があります。

- 294 ページの 「Identity Manager エンジンコンテナの更新」
- 294 ページの 「リモートローダコンテナの更新」
- 295 ページの 「ファンアウトエージェントコンテナの更新」
- 295 ページの 「OSP コンテナの更新」
- 295 ページの 「PostgreSQL コンテナの更新」
- 297 ページの 「Identity Applications コンテナの更新」
- 297 ページの 「フォームレンダラコンテナの更新」
- 297 ページの 「ActiveMQ コンテナの更新」
- 297 ページの 「Identity Reporting コンテナの更新」
- 298 ページの 「SSPR コンテナの更新」
- 298 ページの 「Identity Console コンテナの更新」

Identity Manager エンジンコンテナの更新

1 共有ディレクトリ /data に次の内容で credentials.properties ファイルを作成します。

```
ID_VAULT_ADMIN_LDAP="<ID_VAULT_ADMIN_LDAP>"
ID_VAULT_PASSWORD="<ID_VAULT_PASSWORD>"
```

ここで、ID_VAULT_ADMIN_LDAP は、コンマ区切り形式である必要があります。

次に例を示します。

```
ID_VAULT_ADMIN_LDAP="cn=admin,ou=sa,o=system"
```

```
ID_VAULT_PASSWORD="novell"
```

2 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

3 docker-images ディレクトリに移動します。

4 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityengine.tar.gz
```

5 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=engine-
container -v /etc/hosts:/etc/hosts -v /data:/config -e
SILENT_INSTALL_FILE=/config/credentials.properties --stop-timeout 100
identityengine:idm-4.9.0-580
```

リモートローダコンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_remoteloader.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=<container
name> -v /data:/config --stop-timeout 100 remoteloader:idm-4.9.0-580
```

例 :

```
docker run --restart unless-stopped -d --network=host --name=rl-
container -v /data:/config --stop-timeout 100 remoteloader:idm-4.9.0-
580
```

ドライバファイルはコンテナの /opt/novell/eDirectory/lib/dirxml/classes/ ディレクトリにあります。

5 (状況によって実行) リモートローダインスタンスが実行されていない場合は、リモートローダインスタンスを起動します。

ファンアウトエージェントコンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_fanoutagent.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=foa-
container -v /data:/config --stop-timeout 100 fanoutagent:idm-4.9.0-580
```

5 ファンアウトエージェントを起動します。

OSP コンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_osp.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=osp-
container -v /data:/config --stop-timeout 100 osp:idm-4.9.0-580
```

5 次のコマンドを実行してコンテナにログインします。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it osp-container bash
```

PostgreSQL コンテナの更新

注 : PostgreSQL コンテナを更新する前に、Identity Applications や Identity Reporting などの依存コンテナを必ず停止してください。

---

1 Docker ホストで、任意の場所に移動します。例 :

```
cd /tmp
```

2 次のコマンドを実行して、既存の PostgreSQL コンテナのデータのバックアップを取ります。

```
docker exec postgresql-container pg_dumpall -U postgres > dump.sql
```

3 PostgreSQL コンテナを停止します。

```
docker stop <container name>
```

次に例を示します。

```
docker stop postgresql-container
```

4 PostgreSQL コンテナを削除します。

```
docker rm <container name>
```

5 既存の PostgreSQL データディレクトリを削除します。

```
rm -rf /data/postgres
```

6 (状況によって実行) PostgreSQL Docker イメージを削除します。

```
docker rmi <image ID>
```

7 共有ディレクトリ `/data` の下にサブディレクトリ、たとえば `postgres` を作成します。

```
mkdir postgres
```

8 `Identity_Manager_4.9.0_Containers.tar.gz` ファイルを抽出した場所に移動します。

9 `docker-images` ディレクトリに移動します。

10 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_postgres.tar.gz
```

11 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=<password> -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.18
```

次に例を示します。

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=novell -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.18
```

12 Docker ホスト上にバックアップしたデータファイル (ステップ 2) を新しい PostgreSQL `data` ディレクトリにコピーします。

```
cp /tmp/dump.sql /data/postgres
```

13 次のコマンドを実行してコンテナにログインします。

```
docker exec -it <container> <command>
```

次に例を示します。

```
docker exec -it postgresql-container bash
```

14 `/var/lib/postgresql/data/` ディレクトリに移動します。

15 ステップ 2 でバックアップしたデータを新しい PostgreSQL コンテナに復元します。

```
psql -U postgres < dump.sql
```

16 次のコマンドを実行して、コンテナを終了します。

```
exit
```

Identity Applications コンテナの更新

1. Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。
2. docker-images ディレクトリに移動します。
3. 次のコマンドを実行して、イメージをロードします。

   ```
   docker load --input IDM_490_identityapplication.tar.gz
   ```

4. 次のコマンドを使用してコンテナを更新します。

   ```
   docker run --restart unless-stopped -d --network=host --name=idapps-container -v /data:/config --stop-timeout 100 identityapplication:idm-4.9.0-580
   ```

フォームレンダラコンテナの更新

1. Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。
2. docker-images ディレクトリに移動します。
3. 次のコマンドを実行して、イメージをロードします。

   ```
   docker load --input IDM_490_formrenderer.tar.gz
   ```

4. 次のコマンドを使用してコンテナを更新します。

   ```
   docker run --restart unless-stopped -d --network=host --name=fr-container -v /data:/config --stop-timeout 100 formrenderer:idm-4.9.0-580
   ```

ActiveMQ コンテナの更新

1. Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。
2. docker-images ディレクトリに移動します。
3. 次のコマンドを実行して、イメージをロードします。

   ```
   docker load --input IDM_490_activemq.tar.gz
   ```

4. 次のコマンドを使用してコンテナを更新します。

   ```
   docker run --restart unless-stopped -d --network=host --name=amq-container -v /data:/config --stop-timeout 100 activemq:idm-4.9.0-580
   ```

Identity Reporting コンテナの更新

1. Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。
2. docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_identityreporting.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=rpt-
container -v /data:/config --stop-timeout 100 identityreporting:idm-
4.9.0-580
```

SSPR コンテナの更新

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出した場所に移動します。

2 docker-images ディレクトリに移動します。

3 次のコマンドを実行して、イメージをロードします。

```
docker load --input IDM_490_sspr.tar.gz
```

4 次のコマンドを使用してコンテナを更新します。

```
docker run --restart unless-stopped -d --network=host --name=sspr-
container -v /data/sspr:/config --stop-timeout 100 sspr/sspr-
webapp:4.7.0.1-580
```

Identity Console コンテナの更新

Identity Console コンテナの更新の詳細については、「Docker コンテナとしての Identity Console のアップグレード」を参照してください。

VIII Ansible を使用した Identity Manager コンテナの展開

Identity Manager のこのリリースでは、Ansible を使用した Identity Manager コンテナの展開のサポートが導入されています。Ansible によるアプローチを使用すると、自動化されたプロセスでコンテナを簡単に展開できます。展開プロセスがよりシンプルになり、時間効率が向上します。Identity Manager には、コンテナ展開を自動化するための Ansible プレイブックが付属しています。

このセクションでは、Ansible を使用してコンテナを展開する手順について説明します。

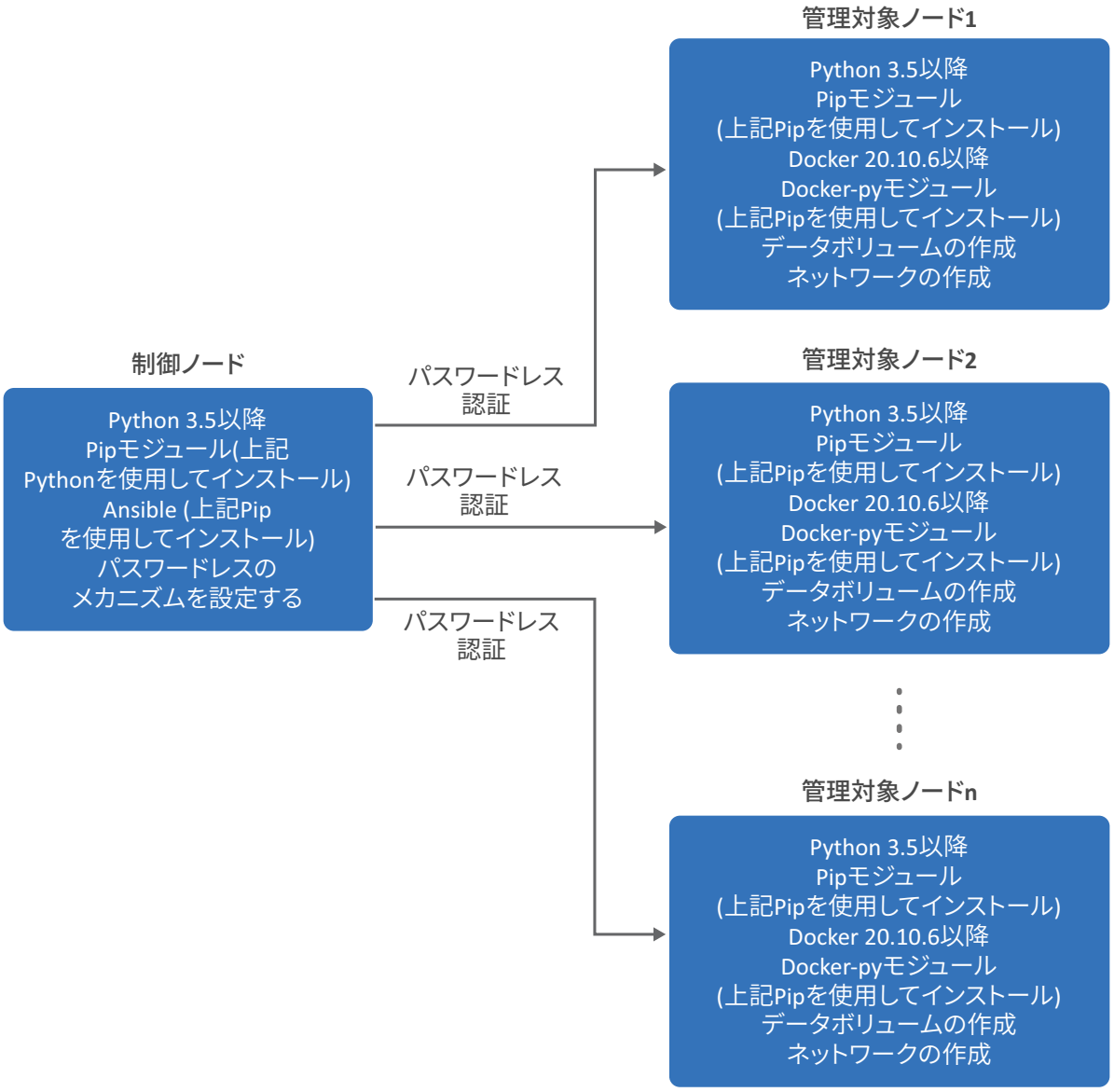
20 展開の計画

コンテナを展開する際には、いくつかの計画と前提条件に従う必要があります。このセクションでは、展開計画の詳細について説明します。

Ansible ベースでコンテナを展開するためのサーバを 2 つ以上特定します。サーバの 1 つは Ansible Control Node (コントロールノード) と呼ばれ、残りのサーバは Managed Nodes (管理対象ノード) と呼ばれます。コントロールノードと管理対象ノードの詳細については、Ansible (https://docs.ansible.com/ansible/latest/network/getting_started/basic_concepts.html#basic-concepts) のマニュアルを参照してください。

Ansible ノードの準備

展開プロセスを開始する前に、Ansible ノードが適切に設定されていることを確認する必要があります。コントロールノードと管理対象ノードの前提条件はそれぞれ異なります。次の図は、コントロールノードと管理対象ノードをどのように準備する必要があるかについての概要を示しています。



コントロールノードの準備

コントロールノードで以下のタスクを実行していることを確認します。

- Python3 以降がインストールされていることを確認します。Python のバージョンを確認するには、`/usr/bin/` ディレクトリに移動し、次のコマンドを実行します。

  例：

  ```
  python3 --version
  ```

  詳細については、Python のマニュアルを参照してください。
- pip がインストールされていることを確認します。pip のバージョンを確認するには、次のコマンドを実行します。

  例：

```
pip --version
```

以前にインストールした Python3 以降のバージョンを使用して `pip` がインストールされていることを確認します。

詳細については、Python のマニュアルを参照してください。

- pip を使用して Docker モジュールをインストールします。次のコマンドを実行します。

  例 :

  ```
  pip install docker
  ```

  Docker のバージョンが 20.10.6 以降であることを確認します。詳細については、Docker のマニュアルを参照してください。

- 以前にインストールした pip を使用して Ansible をインストールします。Ansible バージョン 2.10.5 以降をインストールしていることを確認します。

  例 :

  ```
  pip install ansible
  ```

  詳細については、Ansible (https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-with-pip) のマニュアルを参照してください。

- コントロールノードから管理対象ノードにアクセス可能であることを確認します。たとえば、`ping` や関連するメカニズムを使用して、ノードにアクセス可能であることを確認できます。
- コントロールノードと展開内のすべての管理対象ノードとの間にパスワードレス認証が設定されていることを確認します。次の手順を実行します。
  1. SSH キーを生成します。

     例 :

     ```
     ssh-keygen
     ```

  2. パスワードは入力せずに、キーの生成を続行します。
  3. 次のコマンドを実行して、管理対象ノードに対するパスワードレス認証を有効にします。

     ```
     ssh-copy-id root@<FQDN or IP Address of the managed node>
     ```

     例 :

     ```
     ssh-copy-id root@192.168.0.25
     ```

  4. 管理対象ノードのパスワードを指定します。

     たとえば、`password` です。

  5. 管理対象ノードへの接続をテストします。

     ```
     ssh 'root@<FQDN or IP Address of the managed node>'
     ```

     例 :

     ```
     ssh 'root@<192.168.0.25>'
     ```

管理対象ノードの準備

すべての管理対象ノードで以下のタスクを実行していることを確認します。

- Python3 以降がインストールされていることを確認します。Python のバージョンを確認するには、/usr/bin/ ディレクトリに移動し、次のコマンドを実行します。

  例 :

  ```
  python3 --version
  ```

  詳細については、Python のマニュアルを参照してください。
- pip がインストールされていることを確認します。pip のバージョンを確認するには、次のコマンドを実行します。

  例 :

  ```
  pip --version
  ```

  以前にインストールした Python3 以降のバージョンを使用して pip がインストールされていることを確認します。

  詳細については、Python のマニュアルを参照してください。
- Docker をインストールします。Docker のバージョンが 20.10.6 以降であることを確認します。詳細については、Docker のマニュアルを参照してください。
- pip を使用して Docker python モジュールをインストールします。

  例 :

  ```
  pip install docker-py
  ```
- 共有ディレクトリを作成します。詳細については、254 ページの 「コンテナボリュームデータの管理」を参照してください。

setup.csv ファイルの作成

setup.csv ファイルは、Ansible でコンテナを展開する際に使用する入力ファイルです。Identity Manager では、Identity Manager コンテナの tar ファイルに setup.csv ファイルのデフォルトテンプレートがバンドルされています。

setup.csv ファイルのデフォルトテンプレートは、/<location where you extracted the container tar file>/ansible/input/ ディレクトリにあります。必要に応じて、setup.csv ファイルを編集できます。

setup.csv ファイルに含まれるパラメータと各パラメータの目的については、次のセクションで説明します。

- **Component:** 展開するコンテナを指定します。たとえば、「engine」と指定します。
- **Deploy:** コンテナを展開するかどうかを指定します。サポートされている値は [yes] および [no] です。

- **DockerHost:** コンテナを展開する Docker ホストを指定します。つまり、展開対象として特定した管理対象ノードのいずれかを指定できます。たとえば、「DockerHostA」と指定します。
- **IP Address:** コンテナを展開する Docker ホストの IP アドレスを指定します。たとえば、「192.168.0.15」と指定します。
- **ContainerName:** コンテナの名前を指定します。たとえば、「engine-container」と指定します。
- **ContainerHostname:** コンテナを展開する Docker ホストまたはサーバのホスト名を指定します。NetIQ では、ホスト名を FQDN 形式で指定することをお勧めします。たとえば、「identityengine.example.com」と指定します。
- **ExposedPorts:** コンテナがリスンできるように公開するポートを指定します。たとえば、「636」と指定します。

注 : 各コンテナに対して固有のポートを公開し、silent.properties ファイルの作成時に指定したポートと同じポートを指定してください。たとえば、272 ページの 表 18-2 に記載されているサンプルポートを参照して、公開するポートを計画できます。

- **FileMounting:** ojdbc.jar など、カスタムファイルのパスを指定します。たとえば、「/opt/novell/eDirectory/lib/dirxml/classes/ojdbc.jar」と指定します。

注

- 複数の値がある場合は、スペース区切りの変数リストとして指定します。たとえば、「/opt/novell/eDirectory/lib/dirxml/classes/ojdbc.jar /opt/novell/eDirectory/lib/dirxml/classes/mssql.jar」と指定します。
- (状況によって実行) これは、Deploy 列でコア DNS コンテナの値を [no] に設定している場合にのみ適用されます。

  hosts ファイルが FileMounting フィールドにマッピングされていることを確認します。たとえば、「/etc/hosts」と指定します。

- **SharedVolume:** コンテナがデータ永続化のために使用する共有ディレクトリを指定します。たとえば、「/data」と指定します。このディレクトリをすべての Docker ホストに作成する必要があります。

21 コンテナの展開

コンテナを展開するには、次の手順を実行します。

1 コントロールノードで、次の手順を実行します。

1a Identity Manager コンテナ tar ファイルをダウンロードして、抽出します。詳細については、251 ページの 「Docker イメージの取得」を参照してください。

1b `/<location where you extracted the tar file>/docker-images/` ディレクトリに移動します。

1c `IDM_490_idm_conf_generator.tar.gz` ファイルをコピーして、任意の管理対象ノードに配置します。

2 任意の管理対象ノードで、次の手順を実行します。

2a ステップ 1c でコピーした `IDM_490_idm_conf_generator.tar.gz` ファイルを任意の場所に配置します。たとえば、`/home` です。

2b `silent.properties` ファイルを作成します。詳細については、254 ページの 「サイレントプロパティファイルの作成」を参照してください。

3 コントロールノードで、次の手順を実行します。

3a `/<location where you extracted the tar file>/ansible/input/` ディレクトリに移動し、以下のファイルを配置します。

- ステップ 2b で作成した `silent.properties` ファイル
- 304 ページの 「setup.csv ファイルの作成」セクションで作成した `setup.csv` ファイル
- 外部認証局から取得したカスタム証明書

注 : 証明書を生成する認証局としてアイデンティティボールトを使用している場合は、275 ページの 「アイデンティティボールト認証局での証明書の生成」で説明した手順を実行します。

- `ojdbc.jar` やカスタム LDIF ファイルなどのカスタムファイル

注 : これらのファイルの宛先パスが `setup.csv` ファイルの [FileMounting] 列で指定されていることを確認します。詳細については、304 ページの 「setup.csv ファイルの作成」を参照してください。

3b `/<location where you extracted the tar file>/ansible/` ディレクトリに移動します。

3c (オプション) この手順は上級ユーザに適用されます。展開用の `ansible.cfg` ファイルを確認します。

3d 次のコマンドを実行して、`setup.yml` プレイブックを展開します。

```
ansible-playbook setup.yml
```

3e (オプション) この手順は上級ユーザに適用されます。展開用の idminventory.ini ファイルを確認します。

3f 次のコマンドを実行して、deploy.yml プレイブックを展開します。

```
ansible-playbook deploy.yml -e 'network_set=<Docker network name>'
```

例 :

```
ansible-playbook deploy.yml -e 'network_set=idmoverlaynetwork'
```

22 展開後のタスク

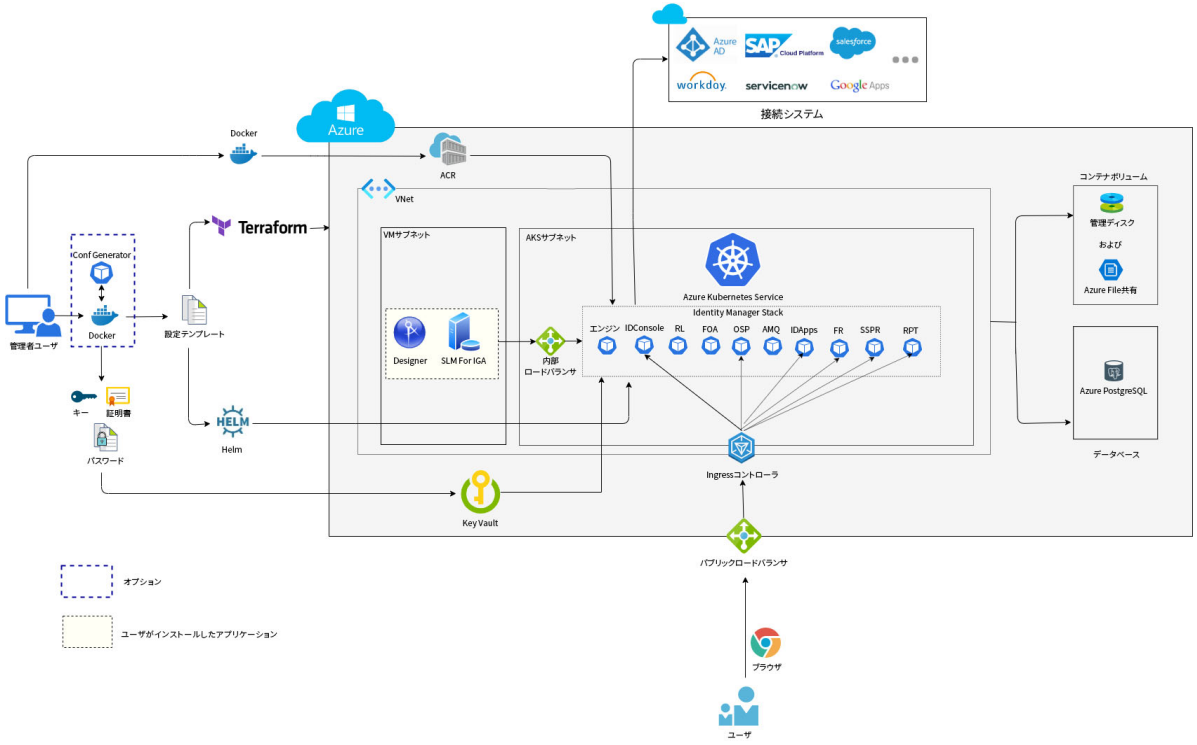
Identity Manager コンテナの展開が完了したら、特定のタスクを実行して、Identity Manager ソリューションがお使いの環境で適切に機能するようにする必要があります。

OAuth 証明書を SSPR にインポートする必要があります。詳細については、284 ページの「SSPR コンテナの展開」セクションの手順 8 を参照してください。

IX Microsoft Azure Kubernetes Service への Identity Manager コンテナの展開

このリリースでは、Identity Manager コンテナを Microsoft Azure Kubernetes Service に展開できます。Identity Manager コンテナの展開は、Terraform と Helm チャートを使用して自動化されています。

次の図は、コンテナを Azure に展開する場合のアーキテクチャの概要を示しています。



- `ACRcreate.sh` ファイルは、新しい Azure コンテナレジストリを作成し、ユーザが Docker イメージを Azure portal にアップロード / 保存できるようにします。
- Identity Manager Configuration Generator イメージは、次に示すタスクを実行するために使用します。
  - 環境設定ファイル (Terraform ファイルと Helm チャート) を生成する。
  - Azure リソースグループを作成する。
  - Key Vault を Azure リソースグループ下に作成する。
  - すべての機密情報を Key Vault にプッシュする。

- Terraform スクリプトを使用して、ユーザは次のようなインフラストラクチャを設定できます。
  - ネットワークの作成。
  - Azure PostgreSQL (オプション)。
  - Azure Kubernetes Service (AKS)。
- Helm チャートを使用して、ユーザは Identity Manager コンテナを Azure Kubernetes クラスタ内に展開できます。
- NGINX Ingress Controller は、AKS クラスタ内で実行されている Identity Manager の Web アプリケーション (Identity Applications、Identity Reporting、Identity Console などを含む) にアクセスするためのリバースプロキシとして機能します。
  - 単一のドメイン (たとえば、`identitymanager.eastus.cloudapp.azure.com`) を購入する必要があります。
  - 同じドメインが NGINX Ingress Controller に割り当てられます。
- 内部ロードバランサは、エンジンの Docker ホストと Kubernetes クラスタ間の内部通信に使用されます。
- 永続ストレージは動的に作成され、インフラストラクチャ全体にマウントされます。
- Identity Applications と Reporting は、Azure が提供する Azure PostgreSQL データベースを使用するように設定できます。
- Designer と Sentinel Log Management (SLM) for Identity Governance Administration は、Terraform の実行後にユーザが手動で展開する必要があります。

概要

Azure Kubernetes クラスタへの Identity Manager コンテナの展開を計画するには、次のセクションを参照してください。

- 313 ページの第 23 章「展開の計画」
- 317 ページの第 24 章「展開の準備」
- 319 ページの第 25 章「展開手順」
- 341 ページの第 26 章「展開後のタスク」
- 343 ページの第 27 章「Identity Manager コンテナの展開の保守」

23 展開の計画

このセクションでは、展開の計画プロセスについて説明し、Azure 上にコンテナを展開するために必要な前提条件のリストを提供します。

Identity Manager コンテナ展開プロセスには、以下の Azure リソースが必要です。

1 Azure アカウントに、インフラストラクチャを作成するための読み書き許可が付与されていることを確認します。

1a Azure Kubernetes Service (AKS)。

1b ネットワークの作成。

1c (オプション) Azure PostgreSQL。

> **注 :** これらの要素は、Terraform 展開スクリプトで作成されます。

> **注 :** 展開する前に、リソースが個別であること、および Identity Manager の環境設定ファイルが Azure アカウントに対してまだ存在していないことを確認します。

2 Identity Manager でさまざまなアプリケーションにアクセスするためにドメインを 1 つ購入します。

各リソースグループ内に独自のドメインを設定できます。たとえば、「`identitymanager.eastus.cloudapp.azure.com`」です。

> **注 :** この例では、**eastus** は場所で、**cloudapp.azure.com** は一般的な Azure サフィックスドメインです。

3 任意の CA 機関からドメインの SSL 証明書 (`.crt` ファイルと `.key` ファイル) を取得します。

4 (オプション) コマンドラインインタフェースで次のコマンドを実行して、専用の自己署名証明書を生成します。

```
openssl req -new -newkey rsa:2048 -nodes -keyout domain.key -out
domain.csr
```

4a 次のプロンプトの入力を指定します。

表 23-1

プロンプト	説明
国名 (2 文字のコード)	国名を 2 文字のコードで指定します。
都道府県名	都道府県の正式名称を指定します。
地域名	市町村名を指定します。

プロンプト	説明
組織名	組織名を指定します。たとえば、「Internet Widgits Pty Ltd」です。
部門名	部門名を指定します。たとえば、「部署」です。
共通名	サーバの完全修飾ドメイン名 (FQDN) を指定します。たとえば、「identitymanager.eastus.cloudapp.azure.com」です。
電子メールアドレス	電子メールアドレスを指定します。
チャレンジパスワード	固有のパスワードを指定します。
オプションの会社名	オプションの会社名を指定します。

4b 次のコマンドを実行して、秘密鍵を RSA 形式に変換します。

```
openssl rsa -in domain.key -out tls.key
```

4c 次の手順を実行して、ルート証明書を取得します。

4c1 `https://<IP address or Host name>:9000/identityconsole/#/` にログインします。

4c2 **[Certificate Management (証明書管理)]** > **[Server Certificate Management (サーバ証明書管理)]** に移動します。

4c3 **[エクスポート]** をクリックします。

4c4 ［証明書］ドロップダウンで、**[Organizational CA (組織の CA)]** を選択します。

4c5 ［エクスポート形式］ドロップダウンで、**[BASE64]** を選択します。

4c6 **[OK]** をクリックします。

4c7 **[Save the exported certificate (エクスポートされた証明書の保存)]** をクリックします。

4d 次の手順を実行して、CSR を CA 機関に送信し、署名済みサーバ証明書を取得します。

4d1 `https://<IP address or Host name>:9000/identityconsole/#/` にログインします。

4d2 **[Certificate Management (証明書管理)]** > **[Issue Certificate (証明書の発行)]** の順に移動します。

4d3 **[Choose File (ファイルの選択)]** をクリックして、314 ページのステップ 4b で生成された `<domain>.csr` ファイルをアップロードします。

4d4 **[Certificate Parameters (証明書パラメータ)]** ドロップダウンで有効期間を選択します。たとえば、［maximum (最大)］を選択します。

4d5 **[Certificate format (証明書形式)]** ドロップダウンで、**[File in Base64 format (Base64 形式のファイル)]** を選択します。

4d6 **[次へ]** をクリックします。

4d7 指定した入力を確認したら **[OK]** をクリックします。

4d8 **[Download the issued certificate (発行された証明書をダウンロードする)]** をクリックします。

4e 次のコマンドを実行して、サーバ証明書とルート証明書を連結します。

```
cat domain.b64 <(echo) cert.b64 <(echo) > tls.crt
```

5 サービスプリンシパル資格情報を生成するには、次の手順を実行します。

重要 : 環境設定を生成し、すべての機密情報を Azure Key Vault にプッシュするには、サービスプリンシパル資格情報が必要です。

5a Azure portal にログインします。

5b  をクリックします。

5c ターミナルウィンドウで、次のコマンドを実行します。

```
az ad sp create-for-rbac --skip-assignment
```

注 : コマンド出力から appid、パスワード、およびテナント ID をメモします。

---

24 展開の準備

このセクションでは、Identity Manager コンテナを展開するための展開前タスクの手順について説明します。

1 ［Software and License (ソフトウェアおよびライセンス)］ページ (https://sld.microfocus.com/) から `Identity_Manager_4.9.0_Containers.tar.gz` をダウンロードします。

2 次のコマンドを実行して、`Identity_Manager_4.9.0_Containers.tar.gz` ファイルを展開します。

```
tar -zxvf Identity_Manager_4.9.0_Containers.tar.gz
```

3 `Identity_Manager_4.9.0_Containers` ディレクトリに移動して、次のコマンドを実行します。

```
./ACRCreate.sh
```

注 : スクリプトを実行するには、現在のディレクトリに最低 6GB の空き容量が必要です。

3a 次の手順を実行して、Microsoft Azure CLI にサインインします。

3a1 https://microsoft.com/devicelogin にログインします。

3a2 出力されたコードを入力します。

3a3 **[次へ]** をクリックします。

3a4 **[続行]** をクリックします。

3b 次のプロンプトの入力を指定します。

プロンプト	説明
Enter the appropriate Azure Account id printed above as-is without double quotes. (出力された適切な Azure アカウント ID をダブルクォーテーションなしでそのまま入力してください。)	上記の手順で生成された Azure アカウント ID を指定します。

プロンプト	説明
Do you want to create azure container registry? (Azure コンテナレジストリを作成しますか ?)	Azure コンテナレジストリを作成するかどうかを指定します。 ◆ (状況によって実行)「はい」の場合、次の操作を行います。 1. 新しいリソースグループを作成するかどうかを指定します。 ◆ (状況によって実行)「はい」の場合、Azure リソースグループ名を指定します。 ◆ (状況によって実行)「いいえ」の場合、既存の Azure リソースグループ名を指定します。 **注 :** リソースグループ名には英数字のみを使用できます。 2. Azure リソースグループの場所を入力します。たとえば、「`eastus`」と入力します。 ◆ (状況によって実行)「いいえ」の場合、既存の Azure コンテナレジストリ URL を入力します。たとえば、「`azureregname.azurecr.io`」です。

3c (状況によって実行) Azure Container Registry の詳細を取得するには、次の手順を実行します。

3c1 Azure portal にログインします。

3c2 リソースグループに移動します。

3c3 ［リソース］の下で、**[コンテナレジストリ]** をクリックします。

3c4 **[設定]** > **[アクセス キー]** に移動し、**[管理者ユーザ]** トグルを有効にします。

3c5 今後使用するためにレジストリの詳細をメモします。

25 展開手順

このセクションでは、Identity Manager コンテナを Azure 上に展開する手順について説明します。展開には 2 つの段階があります。

- 319 ページの 「Azure クラウド上へのインフラストラクチャの展開」
- 334 ページの 「Azure Kubernetes Service への Identity Manager コンテナの展開」
- 336 ページの 「Azure Kubernetes Service でのリモートローダサービスの設定」
- 338 ページの 「Azure Kubernetes Service でのファンアウトエージェントサービスの設定」

Azure クラウド上へのインフラストラクチャの展開

Azure クラウド上へのインフラストラクチャの展開は、Terraform の設定を使用して定義します。Azure 上にインフラストラクチャを展開するには、2 つの異なる方法があります。要件に基づいて、インフラストラクチャの展開に Configuration Generator を使用するか、設定テンプレートを使用するかを選択できます。

- 319 ページの 「Configuration Generator を使用した展開」
- 323 ページの 「設定テンプレートを使用した展開」

Configuration Generator を使用した展開

以下の手順を実行し、Identity Manager Configuration Generator を使用して環境設定ファイルを生成します。

重要 : 次の手順を実行するには、Docker がインストールおよび実行されているマシンが必要です。

---

1. 共有ボリュームを作成します。詳細については、254 ページの 「コンテナボリュームデータの管理」を参照してください。
2. ドメインの証明書を共有ボリュームにコピーします。
3. 次のコマンドを実行して、Docker イメージをロードします。

   ```
   docker load --input docker-images/IDM_490_idm_conf_generator.tar.gz
   ```

4. 解凍した場所に移動し、次のコマンドを実行して Configuration Generator コンテナを展開します。

   ```
   docker run --rm -it --name=idm_conf_generator -v /data:/config
   idm_conf_generator:idm-4.9.0-580
   ```

5. 次の表に、新しいプロンプトに関する情報を示します。

注 : 既存のすべてのプロンプトについては、83 ページの 「環境設定パラメータの理解」のセクションを参照してください。

プロンプト	説明
Do you want to deploy Identity Manager Containers on Azure? (Identity Manager コンテナを Azure に展開しますか ?)For secondary server deployment, please select n and proceed with silent property generation (y/n)? (セカンダリサーバの展開の場合は、n を選択し、サイレントプロパティの生成を続行してください (y/n))	Identity Manager コンテナを展開するかどうかを指定します。 ◆ (状況によって実行)「はい」の場合は、次のプロンプトに進みます。 ◆ (状況によって実行)「いいえ」の場合は、サイレントプロパティファイルの名前を絶対パスで入力します。たとえば、「`/config/silent.properties`」です。 **注 :** セカンダリサーバの展開の場合は、「**n**」と入力して、「サイレントプロパティファイルの作成」に進みます。
Specify the namespace for Kubernetes Deployment. (Kubernetes Deployment のネームスペースを指定してください。)	Kubernetes Deployment のネームスペースを入力します。たとえば、「idm」です。
Do you want to create a new Azure PostgreSQL Server instance? (新しい Azure PostgreSQL サーバインスタンスを作成しますか ?)	新しい Azure PostgreSQL サーバインスタンスを作成するかどうかを指定します。 ◆ (状況によって実行)「はい」の場合は、Azure PostgreSQL サーバ名のプレフィックスを指定します。たとえば、「idmpgserver」です。 **注 :** 入力したプレフィックスには、ハイフンとランダムに生成された 14 桁の数字が追加されます。 ◆ (状況によって実行)「いいえ」の場合は、次のプロンプトに進みます。
Enter the fully qualified domain name (FQDN) for accessing the Identity Manager web applications. (Identity Manager Web アプリケーションにアクセスするための完全修飾ドメイン名 (FQDN) を入力してください。)	Identity Manager Web アプリケーションにアクセスするための FQDN を指定します。たとえば、「`identitymanager.eastus.cloudapp.azure.com`」です。 **注 :** Identity Manager Web アプリケーションには、Identity Applications、Identity Reporting、SSPR、OSP、および Identity Console が含まれます。
Enter the TLS certificate file. (TLS 証明書を入力してください。)	上記で指定したドメインのサブジェクトの代替名と共通名を含む PEM 形式の TLS 証明書ファイルを指定します。たとえば、「`/config/tls.crt`」です。
Enter the private key file for the TLS certificate. (TLS 証明書の秘密鍵ファイルを入力してください。)	TLS 証明書の秘密鍵ファイルを指定します。たとえば、「`/config/tls.key`」です。

プロンプト	説明
Enter the number of instances you want to deploy on Azure for Identity Manager Engine (pod replicas)? (Identity Manager エンジン用に Azure に展開するインスタンス (ポッドレプリカ) の数を入力してください。)	Identity Manager エンジン用に Azure に展開するインスタンス (ポッドレプリカ) の数を指定します。たとえば、「5」と指定します。
OSP 用に Azure 上に展開するインスタンス (ポッドレプリカ) の数を入力してください。	OSP 用に Azure 上に展開するインスタンス (ポッドレプリカ) の数を指定します。たとえば、「5」と指定します。
Enter the number of instances you want to deploy on Azure for Identity Application (pod replicas)? (Identity Applications 用に Azure に展開するインスタンス (ポッドレプリカ) の数を入力してください。)	Identity Applications 用に Azure に展開するインスタンス (ポッドレプリカ) の数を指定します。たとえば、「5」と指定します。
Enter the Azure Service Principal ID. (Azure サービスプリンシパル ID を入力してください。)	「展開の計画」セクションの手順 5 で生成された Azure サービスプリンシパル ID を指定します。
Enter the Azure Service Password. (Azure サービスパスワードを入力してください。)	「展開の計画」セクションの手順 5 で生成された Azure サービスパスワードを入力します。
Enter the Tenant ID of your Service Principal. (サービスプリンシパルのテナント ID を入力してください。)	「展開の計画」セクションの手順 5 で生成されたテナント ID を指定します。
Enter the existing Azure Container Registry Server Name. (既存の Azure Container Registry のサーバ名を入力してください。)	Azure Container Registry のサーバ名を指定します。ステップ 3c を参照してください。
Enter the Azure Container Registry user name. (Azure Container Registry のユーザ名を入力してください。)	Azure Container Registry のユーザ名を指定します。ステップ 3c を参照してください。
Enter the Azure Container Registry user password. (Azure Container Registry のユーザパスワードを入力してください。)	Azure Container Registry のパスワードを指定します。ステップ 3c を参照してください。
Enter the appropriate Azure Account ID printed above as-is without double quotes. (出力された適切な Azure アカウント ID をダブルクォーテーションなしでそのまま入力してください。)	上記の手順で生成された Azure アカウント ID を指定します。
Enter the Azure Resource Group Name. (Azure リソースグループ名を入力してください。)	Azure リソースグループ名を指定します。たとえば、「`idvault-rg`」です。
Enter the Azure Resource Group Location. (Azure リソースグループの場所を入力してください。)	リソースグループの場所を指定します。たとえば、「`eastus`」と入力します。

プロンプトにすべて答えると、Identity Manager Configuration Generator によって次の処理が実行されます。

- 利用可能なリソースグループが Azure にない場合は、リソースグループが作成されます。

- Key Vault とストレージアカウントがリソースグループの下に作成されます。
- すべての機密情報が Key Vault にプッシュされます。

注 : Azure Key Vault の機密情報にアクセスするには、「クイック スタート : Azure portal を使用して Azure Key Vault との間でシークレットの設定と取得を行う」を参照してください。

- Terraform ファイルと Helm チャートを含む `IDM_4.9.0_Cloud_Deployment_files.zip` ファイルが共有ボリュームの下に作成されます。

6 Azure portal にログインします。

注 : Azure Cloud Shell は、最初にサインインしたアカウントを使用して自動的に認証されます。別のアカウントを使用する必要がある場合は、`az login` コマンドを実行して、Azure-CLI にサインインします。

7 次の手順を実行して、Azure Cloud Shell に `IDM_4.9.0_Cloud_Deployment_files.zip` ファイルをアップロードします。

7a をクリックします。

7b ターミナルウィンドウで、 をクリックします。

7c zip ファイルを選択して Azure にアップロードします。

8 次のコマンドを実行して、zip ファイルの内容を展開します。

```
unzip IDM_4.9.0_Cloud_Deployment_files.zip
```

9 `IDM_4.9.0_Cloud_Deployment_files` ディレクトリに移動します。

10 `values.yaml` ファイルに次の変更を加えます。

セクション	フィールド	説明
TLS configuration for the Identity Manager web applications (Identity Manager Web アプリケーションの TLS 設定)	kubernetesSecret	値を `false` として指定します。
	azurekeyVault	値を `true` として指定します。 Ingres TLS 証明書は kubernetes シークレットとして管理されます。
Secret Manager for sensitive data (機密データ用シークレットマネージャ)	kubernetesSecret	値を `false` として指定します。
	azurekeyVault	値を `true` として指定します。 Azure Key Vault をシークレットマネージャとして使用します。
AWS Secret Manager as Secret Manager (AWS Secret Manager をシークレットマネージャとして使用)	awsSecretManager	値を `false` として指定します。

11 (オプション) terraform.tfvars と values.yaml を確認します。

12 (オプション) 次のコマンドを実行して、ストレージアカウントキーを作成します。

```
ACCOUNT_KEY=$(az storage account keys list --resource-group
$AZURE_RESOURCE_GROUP_NAME --account-name
$AZURE_STORAGE_ACCOUNT_FOR_TFSTATE --query '[0].value' -o tsv)
```

次に例を示します。

```
ACCOUNT_KEY=$(az storage account keys list --resource-group idvault-rg
--account-name stract10226600913781 --query '[0].value' -o tsv)
```

13 (オプション) 次のコマンドを実行します。

```
export ARM_ACCESS_KEY=$ACCOUNT_KEY
```

14 次のコマンドを実行して、インフラストラクチャ展開に必要なプラグインをすべてダウンロードします。

```
terraform init
```

15 次のコマンドを実行し、入力に基づいて展開を計画し、理解します。

```
terraform plan
```

16 次のコマンドを実行し、入力で定義されたインフラストラクチャを作成します。

```
terraform apply --auto-approve
```

17 (オプション) 次のコマンドを実行して、ストレージアカウント名、Key Vault 名、データベース管理者の詳細など、Azure アカウント固有の情報を確認します。

```
terraform output
```

Terraform コマンドが正常に実行されたら、334 ページの 「Azure Kubernetes Service への Identity Manager コンテナの展開」で説明されている手順を実行して、Azure 上への Identity Manager コンテナの展開を完了します。

設定テンプレートを使用した展開

次の手順を実行して、仮想マシンの設定テンプレートを更新します。

1 Identity_Manager_4.9.0_Containers.tar.gz ファイルを抽出したディレクトリに移動します。

2 Identity_Manager_4.9.0_Containers/terraform/ ディレクトリに移動し、次のコマンドを実行して、zip ファイルの内容を抽出します。

```
unzip IDM_4.9.0_Azure_Terraform_Configuration.zip
```

3 IDM_Azure_Terraform_Configuration フォルダに移動し、terraform.tfvars ファイルの以下のフィールドを更新します。

表 25-1

フィールド	説明
resource_group_name	Azure リソースグループ名を指定します。たとえば、「idvault-rg」です。
resource_group_location	リソースグループの場所を指定します。たとえば、「eastus」です。
resource_group_exists	既存のリソースグループを確認するかどうかを指定します。たとえば、「true」です。
keyvault_name	固有の Key Vault 名を指定します。
keyvault_exists	既存の Key Vault を確認するかどうかを指定します。たとえば、「true」です。
image_registry_server	Azure Container Registry のサーバ名を指定します。
image_registry_server_username	Azure Container Registry のユーザ名を指定します。
image_registry_server_password	Azure Container Registry のパスワードを指定します。
aks_kubernetes_namespace	Kubernetes Deployment のネームスペースを入力します。たとえば、「idm」です。
azure_postgres_server_name	Azure PostgreSQL サーバ名を指定します。たとえば、「idmpgserver」です。

注 : 残りのフィールドはすべて、`terraform.tfvars` ファイルの生成中に事前入力されています。

4 次のコマンドを実行し、Azure CLI にサインインします。

```
az login
```

5 次の手順を実行して、Azure ストレージアカウントに terraform 状態ファイルを保存します。

5a 次のコマンドを実行して、Azure にストレージアカウントを作成します。

```
az storage account create --name
"${AZURE_STORAGE_ACCOUNT_FOR_TFSTATE}" --resource-group
"${AZURE_RESOURCE_GROUP_NAME}" --location
"${AZURE_RESOURCE_GROUP_LOCATION}" --sku Standard_LRS --encryption-
services blob
```

次に例を示します。

```
az storage account create --name stract10226600913781 --resource-
group idvault-rg --location eastus --sku Standard_LRS --encryption-
services blob
```

5b 次のコマンドを実行して、Azure Storage コンテナを作成します。

```
az storage container create -n terraform-state --account-name
"${AZURE_STORAGE_ACCOUNT_FOR_TFSTATE}"
```

次に例を示します。

```
az storage container create -n terraform-state --account-name
stract10226600913781
```

5c `Identity_Manager_4.8.7_Containers/terraform/IDM_Azure_Terraform_Configuration/` ディレクトリに移動し、`main.tf` ファイルに移動して、`backend "azurerm"` フィールドで以下の詳細を更新します。

フィールド	説明
resource_group_name	Azure リソースグループ名を指定します。たとえば、「idvault-rg」です。
storage_account_name	Azure ストレージアカウント名を指定します。たとえば、「stract89671501132193」です。
container_name	Azure コンテナ名を示します。このフィールドは自動生成されます。
key	Azure キー名を示します。このフィールドは自動生成されます。

6 `terraform.tfvars` と `main.tf` ファイルの変更された詳細を確認し、更新された `IDM_4.9.0_Azure_Terraform_Configuration.zip` ファイルを Azure Cloud Shell にアップロードします。

7 次のコマンドを実行して、インフラストラクチャ展開に必要なプラグインをすべてダウンロードします。

```
terraform init
```

8 次のコマンドを実行し、入力に基づいて展開を計画し、理解します。

```
terraform plan
```

9 次のコマンドを実行し、入力で定義されたインフラストラクチャを作成します。

```
terraform apply --auto-approve
```

10 (オプション) 次のコマンドを実行して、ストレージアカウント名、Key Vault 名、データベース管理者の詳細など、Azure アカウント固有の情報を確認します。

```
terraform output
```

11 次のコマンドを実行して、Key Vault にシークレット値を作成します。

```
az keyvault secret set --name idm-common-password --vault-name <key
vault name> --value "novell@123"
```

注 : 要件に応じて、values.yaml ファイルを使用して複数のシークレット値を指定できます。

12 TLS 証明書を生成したディレクトリに移動し、次のコマンドを実行して、証明書とキーを .pfx 形式にエクスポートできます。

```
openssl pkcs12 -export -out tls.pfx -inkey tls.key -in tls.crt -passout pass:''
```

注 : `tls.pfx` ファイルは、Key Vault でシークレット値とキーを検証するために使用されます。

13 `tls.pfx` ファイルを Azure Cloud Shell にアップロードし、次のコマンドを実行してこのファイルを Key Vault にインポートします。

```
az keyvault certificate import --vault-name <key vault name> -n "ingress-tls-cert" -f tls.pfx
```

次に例を示します。

```
az keyvault certificate import --vault-name idmkv20220712 -n "ingress-tls-cert" -f tls.pfx
```

14 `Identity_Manager_4.9.0_Containers/helm_charts/` ディレクトリに移動し、`values.yaml` ファイルの次のフィールドを更新します。

表 25-2

セクション	フィールド	説明
Advanced Edition of Identity Manager (Identity Manager の Advaned Edition)	IS_ADVANCED_EDITION	Identity Manager の Advanced Edition を展開するかどうかを指定します。(true または false)
Azure PostgreSQL Server instance (Azure PostgreSQL サーバインスタンス)	AZURE_POSTGRESQL_REQUIRED	Identity Applications と Reporting のデータベースサーバとして Azure PostgreSQL サーバインスタンスを使用するかどうかを指定します。(y/n)
Registry credentials for Identity Manager docker images (Identity Manager Docker イメージのレジストリ資格情報)	registry	Azure Container Registry のサーバ名を指定します。
	name	レジストリのログイン資格情報を含む Kubernetes シークレットの名前を指定します。

セクション	フィールド	説明
Data Persistence (データ永続化)	Persistent Storage for Identity Engine (Identity Engine の永続ストレージ)	Identity Engine が使用するボリューム要求テンプレートのストレージクラス名とストレージサイズを指定します。
	volumeClaimTemplate: ◆ storageClassName ◆ storageSize	
	Shared Persistent Storage (共有永続ストレージ) ◆ existingClaim	既存の永続ボリューム要求 (PVC) を使用する場合は、既存の要求の名前を入力します。
	dynamicClaim: ◆ storageClassName ◆ storageSize	また、PVC の動的プロビジョニングでは、ストレージクラス名とストレージサイズを指定します。
Secret Manager for sensitive data such as passwords, keys and certificates (パスワード、キー、証明書などの機密データのためのシークレットマネージャ)	azureKeyVaultName:	Azure Key Vault の名前を指定します。
	azureKeyVaultTenantId	Azure Key Vault のテナント ID を指定します。ステップ 10 の terraform output を参照してください。
	azureUserAssignedIdentityID	Azure Key Vault のシークレットプロバイダが使用するユーザ定義マネージド ID のクライアント ID を指定します。次の az cli コマンドを実行して、その ID のクライアント ID を取得できます。 `az aks show -g <Resource Group> -n <AKS Cluster Name> --query addonProfiles.azureKeyvaultSecretsProvider.identity.clientId -o tsv`
Ingress Configuration (Ingress の設定)	ホスト	Identity Manager Web アプリケーションにアクセスするためのドメイン名を指定します。
	azureKeyVaultCertificateName	TLS 証明書と秘密鍵を含む Azure Key Vault 証明書の名前を指定します。

セクション	フィールド	説明
Identity Engine Configuration (Identity Engine の設定)	展開	Identity Engine を展開するかどうかを指定します。(true または false)
	replicaCount	Identity Engine レプリカポッドの数を指定します。たとえば、「2」です。
	ID_VAULT_TREENAME	アイデンティティボールトツリー名を指定します。
	ID_VAULT_SERVER_CONTEXT	アイデンティティボールトのサーバコンテキストを指定します。
	ID_VAULT_DRIVER_SET	アイデンティティボールトのデフォルトドライバセット名を指定します。
	ID_VAULT_DEPLOY_CTX	アイデンティティボールトのデフォルトドライバセット展開コンテキストを指定します。
	ID_VAULT_ADMIN_LDAP	アイデンティティボールト管理者 DN を指定します。
	ID_VAULT_PASSWORD: secret	アイデンティティボールト管理者パスワードを含む Azure Key Vault シークレットを指定します。
	ID_VAULT_RSA_KEYSIZE	RSA 認証局キーとサーバキーを作成するためのキーサイズを指定します。
	ID_VAULT_EC_CURVE	EC 認証局キーとサーバキーを作成するための Curve を指定します。
	ID_VAULT_CA_LIFE	デフォルトのサーバ証明書を作成するための証明書の有効期間を指定します。
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	Identity Engine のリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

セクション	フィールド	説明
One SSO Provider (OSP) Configuration (One SSO Provider (OSP) の設定)	展開	OSP を展開するかどうかを指定します。(true または false)
	replicaCount	Identity Engine レプリカポッドの数を指定します。たとえば、「2」です。
	OSP_CUSTOM_NAME	OSP カスタムログイン画面名を指定します。
	SSO_SERVICE_PWD: secret	OSP クライアントパスワードを含む Azure Key Vault シークレットを指定します。
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	OSP のリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

セクション	フィールド	説明
Identity Applications Configuration (Identity Applications の設定)	展開	Identity Applications を展開するかどうかを指定します。(true または false)
	replicaCount	Identity Engine レプリカポッドの数を指定します。たとえば、「2」です。
	UA_ADMIN	Identity Applications 管理者 DN を指定します。
	UA_ADMIN_PWD: secret	Identity Applications 管理者パスワードを含む Azure Key Vault シークレットを指定します。
	UA_WFE_DB_PLATFORM_OPTION	Identity Applications およびワークフローエンジンデータベースプラットフォームを指定します。サポートされている値は、postgres、oracle、および mssql です。
	UA_ORACLE_DATABASE_TYPE	データベースプラットフォームが Oracle の場合、 データベースの設定を指定します。(sid/service)
	UA_WFE_DB_HOST	Identity Applications およびワークフローエンジンデータベースサーバのホストを指定します。
	UA_WFE_DB_PORT	Identity Applications およびワークフローエンジンデータベースサーバのポート番号を指定します。
	UA_DATABASE_NAME	Identity Applications データベース名を指定します。
	WFE_DATABASE_NAME	ワークフローエンジンデータベース名を指定します。
	UA_WFE_DATABASE_USER	Identity Applications データベースユーザ名を指定します。
	UA_WFE_DATABASE_PWD: secret	Identity Applications データベースユーザパスワードを含む Azure Key Vault シークレットを指定します。
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	Identity Applications のリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

セクション	フィールド	説明
Form Renderer Configuration (フォームレンダラの設定)	展開	フォームレンダラを展開するかどうかを指定します。(true または false)
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	フォームレンダラのリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
ActiveMQ Configuraion (ActiveMQ の設定)	展開	ActiveMQ を展開するかどうかを指定します。(true または false)
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	フォームレンダラのリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

セクション	フィールド	説明
Identity Reporting Configuration (Identity Reporting の設定)	展開	Identity Reporting を展開するかどうかを指定します。(true または false)
	RPT_ADMIN	Identity Reporting 管理者 DN を指定します。
	RPT_ADMIN_PWD: secret	Identity Reporting 管理者パスワードを含む Azure Key Vault シークレットを指定します。
	RPT_DATABASE_PLATFORM_OPTION	Identity Repoting データベースプラットフォームを指定します。サポートされている値は、postgres、oracle、および mssql です。
	RPT_ORACLE_DATABASE_TYPE	データベースプラットフォームが Oracle の場合、 データベースの設定を指定します。(sid/service)
	RPT_DATABASE_HOST	Identity Reporting データベースホストを指定します。
	RPT_DATABASE_PORT	Identity Reporting データベースポート番号を指定します。
	RPT_DATABASE_NAME	Identity Reporting データベース名を指定します。
	RPT_DATABASE_USER	Identity Reporting データベースユーザを指定します。
	RPT_DATABASE_SHARE_PASSWORD: secret	Identity Reporting データベースアカウントパスワードを含む Azure Key Vault シークレットを指定します。
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	フォームレンダラのリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

セクション	フィールド	説明
Self Service Password Reset (SSPR) Configuration (Self Service Password Reset (SSPR) の設定)	展開	SSPR を展開するかどうかを指定します。(true または false)
	CONFIGURATION_PWD:	管理者が SSPR を設定するために作成するパスワードを指定します。
	secret:	
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	SSPR のリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
Identity Console Configuration (Identity Console の設定)	展開	Identity Console を展開するかどうかを指定します。(true または false)
	ID_CONSOLE_USE_OSP	One SSO Provider (OSP) は Identity Console のログイン方法としてサポートされていないため、デフォルト値は「n」にする必要があります。
	Resource requests and limits (リソース要求と制限) ◆ cpu ◆ memory	Identity Console のリソース要求と制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
詳細環境設定	**DATA_CONTAINERS**:	
	DATA_CONTAINERS_LDIF	データコンテナを作成するための LDIF 設定を指定します。
	ROOT_CONTAINER GROUP_ROOT_CONTAINER USER_CONTAINER ADMIN_CONTAINER	ルートコンテナ、グループルートコンテナ、ユーザコンテナ、および管理者コンテナの DN を指定します。
	Kubernetes Cluster Domain (Kubernetes クラスタドメイン):	Kubernetes クラスタドメインを指定します
	KUBE_SUB_DOMAIN	
TLS configuration for the Identity Manager web applications (Identity Manager Web アプリケーションの TLS 設定)	kubernetesSecret	値を `false` として指定します。
	azurekeyVault	値を `true` として指定します。 Ingres TLS 証明書は kubernetes シークレットとして管理されます。

セクション	フィールド	説明
Secret Manager for sensitive data (機密データ用シークレットマネージャ)	kubernetesSecret	値を `false` として指定します。
	azurekeyVault	値を `true` として指定します。 Azure Key Vault をシークレットマネージャとして使用します。
AWS Secret Manager as Secret Manager (AWS Secret Manager をシークレットマネージャとして使用)	awsSecretManager	値を `false` として指定します。

`terraform.tfvars` および `values.yaml` ファイルを更新した後、334 ページの 「Azure Kubernetes Service への Identity Manager コンテナの展開」で説明されている手順を実行して、Azure への Identity Manager コンテナの展開を完了します。

Azure Kubernetes Service への Identity Manager コンテナの展開

このセクションでは、Azure Kubernetes Service への Identity Manager コンテナの展開について説明します。

1 次のコマンドを実行して Kubernetes クラスタに接続し、必要な設定を保存します。

```
az aks get-credentials --resource-group <resource group> --name
<cluster name> --overwrite-existing
```

次に例を示します。

```
az aks get-credentials --resource-group idvault-rg --name cluster-name
--overwrite-existing
```

2 次のコマンドを実行して、ロードバランサの実行に使用する Nginx インスタンスを作成します。

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
```

3 次の手順を実行して、Azure portal から Kubernetes Service のパブリック IP アドレスを取得します。

3a Azure のホームページに移動し、[すべてのリソース] をクリックします。

3b 検索ボックスに、「ロードバランサ」と入力して、<Enter> を押します。

3c Azure Cloud Shell で次のコマンドを実行し、新しいパブリック IP を作成します。

```
az network public-ip create     --resource-group <aks-cluster-
resource-group-name> --name <name-public-ip> --sku Standard  --
version IPv4 --zone 1 2 3 --ip-tags 'RoutingPreference=Internet' --
dns-name <Domain Name>   --query publicIp.ipAddress -o tsv
```

3d リソースグループと AKS クラスタに関連付けられているロードバランサをクリックします。

3e [設定] > [フロントエンド IP 構成] に移動して、334 ページのステップ 3c で作成したパブリック IP を指定します。

3f [保存] をクリックします。

4 IDM_4.9.0_Cloud_Deployment_files ディレクトリに移動し、次のコマンドを実行して Nginx-ingress コントローラをインストールします。

```
helm install ingress-nginx ingress-nginx/ingress-nginx --namespace
<your namespace> --set controller.replicaCount=1 --set
controller.nodeSelector."kubernetes\.io/os"=linux --set
defaultBackend.nodeSelector."kubernetes\.io/os"=linux --set
controller.service.loadBalancerIP=<Kubernetes public IP address
obtained from the Azure portal> --set
controller.service.externalTrafficPolicy=Local --set
controller.service.annotations."service\.beta\.kubernetes\.io/azure-
dns-label-name"=<Domain Name>
```

334 ページのステップ 3c で作成したパブリック IP アドレスを使用します。

次に例を示します。

```
helm install ingress-nginx ingress-nginx/ingress-nginx --namespace idm
--set controller.replicaCount=1 --set
controller.nodeSelector."kubernetes\.io/os"=linux --set
defaultBackend.nodeSelector."kubernetes\.io/os"=linux --set
controller.service.loadBalancerIP="20.33.59.34" --set
controller.service.externalTrafficPolicy=Local --set
controller.service.annotations."service\.beta\.kubernetes\.io/azure-
dns-label-name"=identitymanager
```

5 Helm チャートを実行するには、次のコマンドを実行します。

```
helm install identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

次に例を示します。

```
helm install identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace idm -f values.yaml
```

6 次のコマンドを実行して、Azure Kubernetes Service で実行されているポッドのリストを表示します。

```
kubectl get pods --watch -n <namespace>
```

次に例を示します。

```
kubectl get pods --watch -n idm
```

Azure Kubernetes Service でのリモートローダサービスの設定

次の手順を実行して、Azure Kubernetes Service にリモートローダサービスをセットアップします。

1. リモートローダのデータを永続化するために、永続ボリューム (PV) と永続ボリューム要求 (PVC) を作成します。
2. PV 内にディレクトリ `rdxml` を作成します。
3. `rdxml` ディレクトリ内に `driverconf` ディレクトリを作成します。
4. `rdxml` ディレクトリ内に `keystore` ディレクトリを作成します。
5. 必要なキーストアファイルと証明書をすべて keystore ディレクトリにコピーします。
6. `rdxml` ディレクトリ内に新しい `.txt` ファイルを作成します。たとえば、`StartupRL.txt` です。
7. `StartupRL.txt` ファイルに、必要なコンテンツを次の形式で指定します。

   ```
   <Remote Loader configuration file> -sp <driver password> <Remote Loader password>
   ```

   例 :

   ```
   config8000.txt -sp dirxml dirxml
   ```

   または、次の形式でもエントリを指定できます。

   ```
   <Remote Loader configuration file> -ksp <keystore password> -kp <key password>
   ```

   例 :

   ```
   config8000.txt -ksp dirxml -kp dirxml
   ```

8. コンテナ tarball から `helm-charts/rl-values.yaml` をコピーし、ファイル内の以下のフィールドを更新します。

表 25-3

セクション	フィールド	説明
Registry credentials for Remote Loader docker images (リモートローダ Docker イメージのレジストリ資格情報)	レジストリ	Azure Container Registry のサーバ名を指定します。
	name	レジストリのログイン資格情報を含む Kubernetes シークレットの名前を指定します。

セクション	フィールド	説明
Data Persistence (データ永続化)	Shared Persistent Storage (共有永続ストレージ) ◆ existingClaim dynamicClaim: ◆ storageClassName ◆ storageSize	既存の永続ボリューム要求 (PVC) を使用する場合は、既存の要求の名前を入力します。 また、PVC の動的プロビジョニングでは、ストレージクラス名とストレージサイズを指定します。
Driver Startup File (ドライバ起動ファイル)	driverStartupFileName	ドライバ起動ファイルの名前を指定します。たとえば、「`StartupRL.txt`」です。
Remote Loader Service (リモートローダサービス)	ports	リモートローダサービスがリスンするポートのリストを指定します。
Resource Requests and Limits (リソース要求と制限)	Resource requests (リソース要求) ◆ cpu ◆ memory	リソース要求の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
	Limits (制限) ◆ cpu ◆ memory	制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

9 コンテナ tarball から `helm-charts/ remote-loader-1.4.0.tgz` をコピーし、次の helm コマンドを実行してリモートローダをインストールします。

```
helm install remote-loader helm_charts/remote-loader-1.4.0.tgz --
namespace idm -f rl-values.yaml
```

10 (オプション) 複数のリモートローダをインストールする場合は、コンテナ tarball から `helm-charts/ remote-loader-1.4.0.tgz` をコピーし、次の helm コマンドを実行します。

```
helm install <remote-loader-n> helm_charts/remote-loader-1.4.0.tgz --
namespace idm -f rl-values.yaml
```

`remote-loader-n` の n は、インストールする n 番目のリモートローダです。

例 :

リモートローダを 2 つインストールする場合、2 番目のリモートローダをインストールするコマンドは次のようになります。

```
helm install remote-loader-2 helm_charts/remote-loader-1.4.0.tgz --
namespace idm -f rl-values.yaml
```

Azure Kubernetes Service でのファンアウトエージェントサービスの設定

次の手順を実行して、Azure Kubernetes Service にファンアウトエージェントサービスをセットアップします。

1 コンテナ tarball から `helm-charts/fa-values.yaml` をコピーし、ファイル内の以下のフィールドを更新します。

表 25-4

セクション	フィールド	説明
Registry credentials for Remote Loader docker images (リモートローダ Docker イメージのレジストリ資格情報)	レジストリ	Azure Container Registry のサーバ名を指定します。
	name	レジストリのログイン資格情報を含む Kubernetes シークレットの名前を指定します。
Data Persistence (データ永続化)	Shared Persistent Storage (共有永続ストレージ) ◆ existingClaim	既存の永続ボリューム要求 (PVC) を使用する場合は、既存の要求の名前を入力します。
	dynamicClaim: ◆ storageClassName ◆ storageSize	また、PVC の動的プロビジョニングでは、ストレージクラス名とストレージサイズを指定します。
Fanout Agent Service (ファンアウトエージェントサービス)	ports	リスンするリモートローダサービスのポートの値を指定します。

セクション	フィールド	説明
Resource Requests and Limits (リソース要求と制限)	Resource requests (リソース要求) ◆ cpu ◆ memory	リソース要求の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
	Limits (制限) ◆ cpu ◆ memory	制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
ActiveMQ Service (ActiveMQ サービス)	install	ファンアウトエージェントに ActiveMQ サービスを展開するかどうかを指定します (true/false)。
	Resource requests (リソース要求) ◆ cpu ◆ memory	ActiveMQ サービスのリソース要求の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。
	Limits (制限) ◆ cpu ◆ memory	ActiveMQ サービスの制限の CPU 値とメモリ値を指定します。詳細については、「Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)」を参照してください。

2 コンテナ tarball から、`helm-charts/fanout-agent-1.4.0.tgz` をコピーし、次の helm コマンドを実行してファンアウトエージェントをインストールします。

```
helm install fanout-agent helm_charts/fanout-agent-1.4.0.tgz --
namespace idm -f rl-values.yaml
```

3 (オプション) 複数のファンアウトエージェントをインストールする場合は、コンテナ tarball から `helm-charts/ fanout-agent-1.4.0.tgz` をコピーし、次の helm コマンドを実行します。

```
helm install <fanout-agent-n> helm_charts/fanout-agent-1.4.0.tgz --
namespace idm -f rl-values.yaml
```

`fanout-agent-n` の n は、インストールする n 番目のファンアウトエージェントです。

例：

ファンアウトエージェントを 2 つインストールする場合、2 番目のファンアウトエージェントをインストールするコマンドは次のようになります。

```
helm install fanout-agent-2 helm_charts/fanout-agent-1.4.0.tgz --
namespace idm -f rl-values.yaml
```

26 展開後のタスク

このセクションでは、Identity Manager コンテナを展開するための展開後タスクの手順について説明します。

Identity Manager アプリケーションへのアクセス

Identity Manager アプリケーションにアクセスするには、ブラウザに次の URL を入力します。

```
https://<domainname>/<APP NAME>
```

次に例を示します。

```
https://identitymanager.eastus.cloudapp.azure.com/idmdash
```

Azure 仮想マシンへの Designer の展開

次の手順を実行して、VM サブネット下の Azure 仮想マシンに Designer を展開します。

1. Linux マシンを作成するには「クイック スタート : Azure portal で Linux 仮想マシンを作成する」を、Windows VM を作成するには「クイック スタート : Azure Portal で Windows 仮想マシンを作成する」を参照してください。
2. Designer を接続するためのパブリック IP アドレスを割り当てます。
3. Designer をダウンロードして、作成した仮想マシンにインストールします。詳細については、74 ページの 「Designer のインストール」または Windows 用の「Designer のインストール」を参照してください。

Azure 仮想マシンへの SLM for IGA の展開

VM サブネット下の Azure 仮想マシンに Sentinel Log Management (SLM) for Identity Governance and Administration (IGA) を展開するには、次の手順を実行します。

1. Linux マシンを作成します。「クイック スタート : Azure portal で Linux 仮想マシンを作成する」を参照してください。
2. SLM for IGA のインストーラを接続するためのパブリック IP アドレスを割り当てます。
3. SLM for IGA をダウンロードして、作成した仮想マシンにインストールします。詳細については、77 ページの 「Sentinel Log Management for Identity Governance and Administration のインストール」を参照してください。

AKS の RPM 更新とサードパーティファイルの処理

このサービスパックは、RPM 更新とサードパーティファイルをコンテナ内で効率的に処理する方法を提供します。これは、mountfiles ディレクトリに必要な RPM ファイル、ライブラリ (.so) ファイル、サードパーティ .jar ファイルを配置することで実現できます。mountfiles ディレクトリにある RPM ファイルは強制的に更新されます。コンテナが再起動すると、.so ファイルと .jar ファイルはそれぞれ、/opt/novell/eDirectory/lib64/nds-modules/ ディレクトリと /opt/novell/eDirectory/lib/dirxml/classes/ ディレクトリに自動的にソフトリンクされます。このオプションの目的は、更新されたドライバを Docker コンテナ内に配置することです。

注 : 現在、この機能拡張は、Identity Manager エンジンとリモートローダのコンテナに適用されます。

---

1 Azure portal にログインし、  をクリックします。

2 次のコマンドを実行して永続ボリュームにログインします。

```
kubectl exec –it -n idm <container name> bash
```

次に例を示します。

```
kubectl exec –it -n idm identityengine-0 bash
```

3 必要なファイルを各コンテナに固有の mountfiles ディレクトリにコピーします。

3a (状況によって実行) Identity Manager エンジンコンテナの場合は、次のコマンドを実行して、ファイルを /config/idm/mountfiles にコピーします。

```
kubectl cp -n <namespace> <file to be copied> <container name>:/
config/idm/mountfiles
```

3b (状況によって実行) Identity リモートローダコンテナの場合は、次のコマンドを実行して、ファイルを /config/rdxml/mountfiles にコピーします。

```
kubectl cp -n <namespace> <file to be copied> <container name>:/
config/rdxml/mountfiles
```

注 : サポートされるファイル形式は、.so、.jar、および .rpm です。

---

4 次のコマンドを実行して、コンテナを再起動します。

```
helm upgrade identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

重要 : マウントファイル内に存在する rpm が優先されるため、今後更新を行う場合は、既存の rpm を慎重に削除し、最新の rpm に更新する必要があります。

---

27 Identity Manager コンテナの展開の保守

このセクションでは、Azure Kubernetes クラスタ上のポッド (1 つ以上のコンテナのグループ) の展開を保守する方法について説明します。

重要 : 展開にさらにチャートを追加する必要がある場合、展開を再作成する必要がある場合、または展開内のポッド数をスケールアップ (またはスケールダウン) する必要がある場合は、次の手順を実行します。

1 Azure portal にログインします。

2  をクリックして、次のコマンドを実行します。

```
helm upgrade identity-manager helmcharts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

データ永続化レイヤの管理

次の手順を実行して、Visual Studio Code アプリケーションのポッドのデータレイヤを変更します。

1 Visual Studio Code をダウンロードしてインストールします。code.visualstudio.com/download を参照してください。

2 仮想コードアプリケーションを開いて、  をクリックします。

3 検索ボックスに「**Azure**」と入力して、<Enter> を押します。

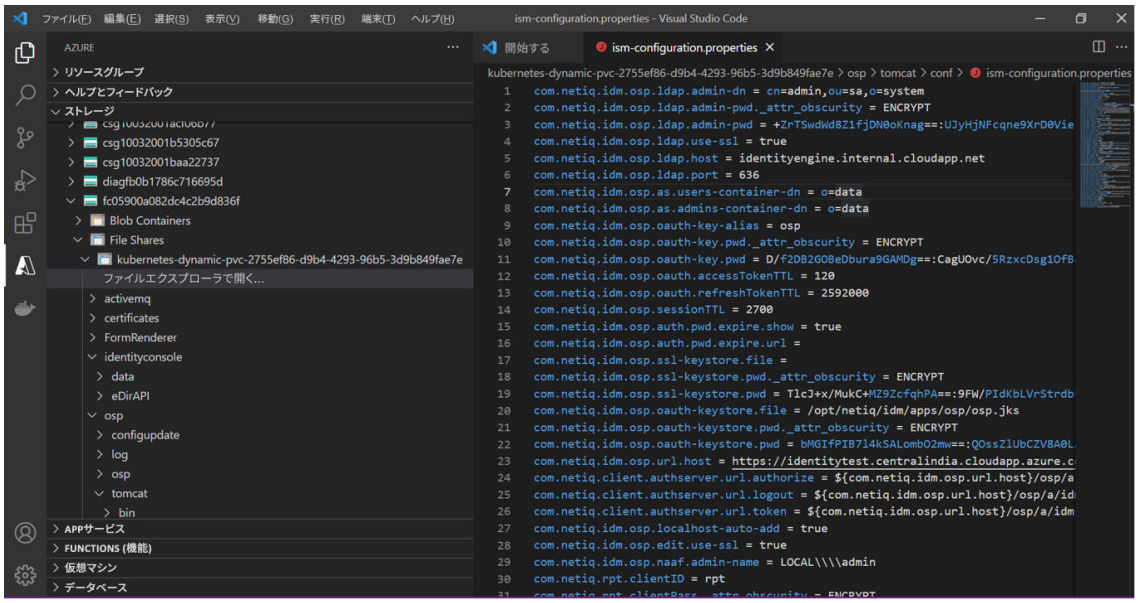
4 Azure Tools で、**[Install (インストール)]** をクリックします。

5  をクリックします。

6 **[STORAGE (ストレージ)]** > **[Security-IdentityMgmtAZ-NonProd]** > **[fc05900a082dc4c2b9d836f]**> **[File shares (ファイル共有)]** > **[Kubernetes dynamic pvc (Kuberne 動的 pvc)]** に移動します (詳細については、344 ページの 図 27-1 を参照)。

注 : 上記のナビゲーションパスで、**Security-IdentityMgmtAZ-NonProd** と **fc05900a082dc4c2b9d836f** キーは参照用に記載されており、お使いのログイン資格情報に応じて変わります。

図 27-1 Visual Studio のストレージ



7 ファイルを選択し、必要なプロパティを変更します。

8 [ファイル] メニューに移動し、[保存] をクリックします。

9 すべての変更を保存したら、ポッドを再起動します (344 ページの 「ポッドの再起動」を参照)。

ポッドの再起動

次の手順を実行して、Azure でポッドを再起動します。

たとえば、データレイヤに変更を加えた場合は、コンテナを再起動する必要があります。

1 Azure portal にログインします。

2 検索ボックスに「すべてのリソース」と入力して、<Enter> を押します。

3 [サービス] の下で、[すべてのリソース] を選択します。

4 Kubernetes クラスタに移動します。

5 [Kubernetes リソース] の下で、[Workload] をクリックします。

6 [Pod] タブに移動し、[ネームスペースでフィルタ] ドロップダウンリストをクリックして、[idm] を選択します。

7 再起動するポッドを選択します。

8 [削除] をクリックします。詳細については、345 ページの 図 27-2 を参照してください。

注 : 削除されたポッドが識別され、新しいポッドへの展開が自動的にトリガされます。

図 27-2 Kubernetes クラスタ



ポッドへのアクセス

次の手順を実行して、Azure のポッドにアクセスします。

1 Azure portal にログインします。

2 をクリックします。

3 次のコマンドを実行して、ポッドの名前を特定します。

```
kubectl get pods --watch -n <namespace>
```

次に例を示します。

```
kubectl get pods --watch -n idm
```

4 次のコマンドを実行します。

```
kubectl exec -it -n idm <pod-name> bash
```

次に例を示します。

```
kubectl exec -it -n idm userapp-d54f98b58-qcpmz bash
```

Azure Kubernetes クラスタの再起動

次の手順を実行して、すべての IDM コンポーネントが正常にシャットダウンするようにします。

1 Azure portal にログインします。

2 Kubernetes クラスタに移動します。

3 ［Kubernetes リソース］の下で、[Workload] をクリックします。

4 [Deployment] タブに移動し、[ネームスペースでフィルタ] ドロップダウンリストをクリックして、ネームスペースを選択します。

5 展開を選択し、[削除] をクリックします。

6 [Pod] タブに移動し、[ネームスペースでフィルタ] ドロップダウンリストをクリックして、ネームスペースを選択します。

7 [リフレッシュ] をクリックし、コンポーネントポッドが完全に削除されていることを確認します。

8 [ステートフルセット] タブに移動し、[ネームスペースでフィルタ] ドロップダウンリストをクリックして、ネームスペースを選択します。

9 ステートフルセットを選択し、[削除] をクリックします。

10 もう一度、[Pod] タブに移動し、[ネームスペースでフィルタ] ドロップダウンリストをクリックして、ネームスペースを選択します。

11 [リフレッシュ] をクリックし、コンポーネントが完全に削除されていることを確認します。

12 [Kubernetes リソース] の下の [概要] に移動し、[停止] をクリックします。

13 クラスタが完全に停止したら、[開始] をクリックします。

14 Azure CLI で次のコマンドを実行し、Nginx-ingress コントローラを再起動します。

```
helm upgrade nginx-ingress ingress-nginx/ingress-nginx --namespace
<your namespace> --set controller.replicaCount=1 --set
controller.service.loadBalancerIP=<Public IP address assigned to load
balancer> --set
controller.service.annotations."service\.beta\.kubernetes\.io/azure-
dns-label-name"=<LoadBalancerName>
```

次に例を示します。

```
helm upgrade nginx-ingress ingress-nginx/ingress-nginx --namespace idm
--set controller.replicaCount=1 --set
controller.service.loadBalancerIP=192.168.0.1 --set
controller.service.annotations."service\.beta\.kubernetes\.io/azure-
dns-label-name"=identitymanager
```

15 次のコマンドを実行して、残りのコンポーネントを再起動します。

```
helm upgrade identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

次に例を示します。

```
helm upgrade identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace idm -f values.yaml
```

16 すべてのポッドが準備完了の状態になるまで待ちます。

ポッドの現在のステータスを確認するには、次のコマンドを実行します。

```
kubectl get pods --watch -n <namespace>
```

17 次のコマンドを実行して、エンジンサービスを再起動します。

```
kubectl rollout restart -n <namespace> statefulset/identityengine
```

次に例を示します。

```
kubectl rollout restart -n idm statefulset/identityengine
```

18 ステップ 16 を繰り返します。

Identity Manager エンジンのレプリカのスケーリング

Identity Manager エンジンのレプリカのスケーリング手順は、次の 2 つの部分に分かれています。

Identity Manager エンジンのレプリカをダウンスケールするユースケース

このトピックでは、Identity Manager エンジンのレプリカをダウンスケールするために参照できる基本的なユースケースについて説明します。

始める前に

- Identity Manager エンジンのレプリカをダウンスケールする前に、このプロセスをよく理解してください。
- ダウンスケールできるポッドは一度に 1 つだけです。

ユースケース

Identity Manager エンジンのインスタンスが 3 つあり、それらのポッド名が次のとおりであると仮定します。

- identityengine-0
- identityengine-1
- identityengine-2

2 つの Identity Manager エンジンインスタンスにダウンスケールすると、identityengine-2 がダウンします。

ダウンスケールする前に、identityengine-2 で次の操作が必要です。

1 インスタンス identityengine-2 で実行されているドライバを別のインスタンスに移行 / 再作成します。

2 サーバをツリーから削除します。

3 次のコマンドを実行して、永続ボリューム要求 (PVC) を一覧表示します。

```
kubectl get pvc -n <namespace>
```

例 :

```
kubectl get pvc -n idm
```

4 次のコマンドを実行して、identityengine-2 の PVC を削除します。

```
kubectl delete pvc -n <namespace> <identityengine replica name>
```

例 :

```
kubectl delete pvc -n idm identityenginedata-identityengine-2
```

5 values.yaml ファイルでエンジンのレプリカ数を 2 に設定します。

6 次のコマンドを使用してスケールダウンします。

```
helm upgrade identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

Identity Manager エンジンのレプリカのアップスケール

次の手順を実行して、Identity Manager エンジンのレプリカをアップスケールします。

1 Azure portal にログインします。

2  をクリックします。

3 values.yaml ファイルで Identity Manager エンジンのレプリカ数を増やします。

4 次のコマンドを使用してスケールアップします。

```
helm upgrade identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

OSP と Identity Applications のレプリカのスケーリング

次の手順を実行して、OSP と Identity Applications のレプリカをアップスケールまたはダウンスケールします。

1 Azure portal にログインします。

2  をクリックします。

3 values.yaml ファイルで、OSP/Identity Applications のレプリカ数を増減します。

4 次のコマンドを使用してスケールアップまたはスケールダウンします。

```
helm upgrade identity-manager helm_charts/identity-manager-1.4.0.tgz --
namespace <namespace> -f values.yaml
```

Terraform で作成されたリソース全体の破棄

次の手順を実行して、Terraform で作成されたリソースを破棄します。

注 : 十分注意を払ったうえでこのプロセスを続行してください。

1 Terraform を使用してリソースを作成したディレクトリに移動します。

2 次のコマンドを実行して、リソースを削除します。

```
terraform destroy --auto-approve
```

3 (オプション) 上記のコマンドの実行中にエラーが発生した場合は、同じコマンドを再実行してください。

X AWS EC2 での Identity Manager の展開

このセクションでは、AWS クラウドでの Identity Manager の計画と実装について説明します。

- 353 ページの第 28 章「AWS EC2 での Identity Manager の計画と実装」
- 373 ページの第 29 章「ハイブリッド Identity Manager のシナリオの例」

28 AWS EC2 での Identity Manager の計画と実装

Identity Manager では、Amazon Web Services (AWS) EC2 上のサービスとして、以下の Identity Manager コンポーネントを展開するためのサポートが追加されています。

- 識別ボールト
- Identity Manager エンジン
- Identity Manager ドライバとリモートローダ
- Designer
- 識別情報アプリケーション
- Identity Reporting

AWS EC2 でサポートされるオペレーティングシステムの詳細については、Identity Manager 4.9 のシステム要件を参照してください。

前提条件

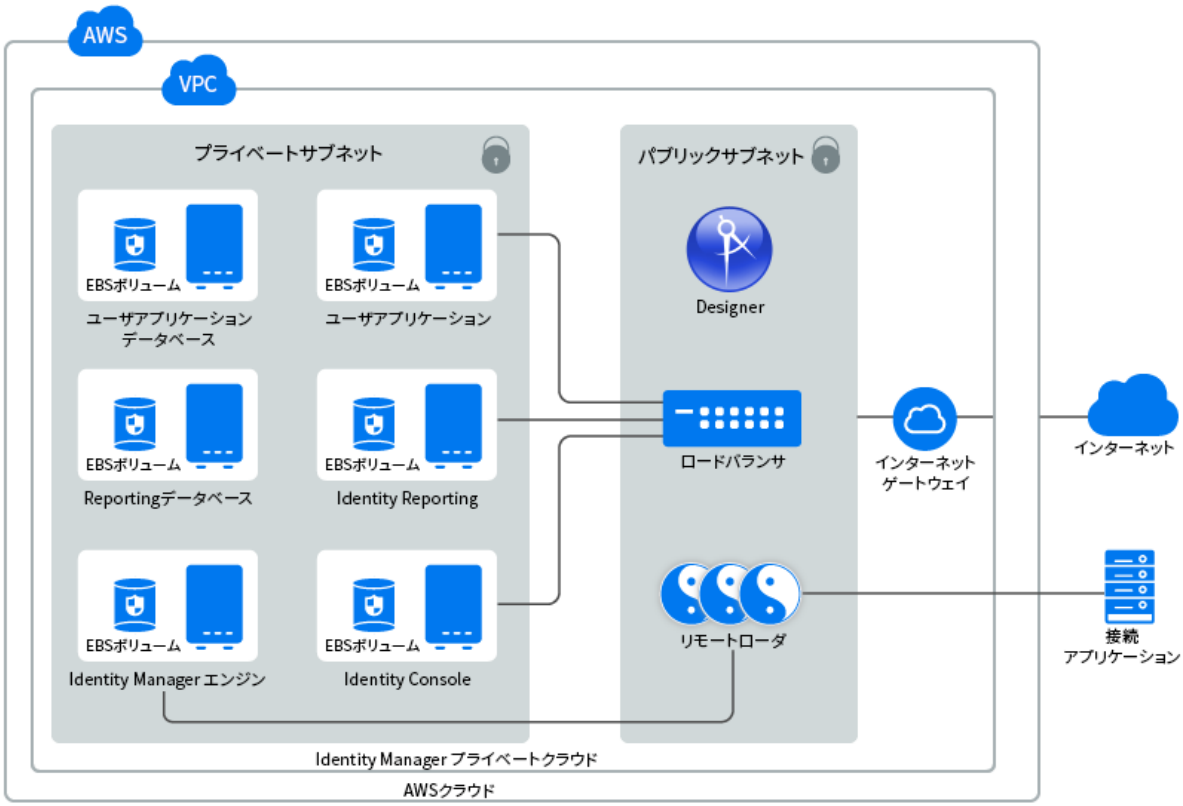
Identity Manager コンポーネントのシステム要件に加えて、以下の前提条件を満たしていることを確認してください。

- AWS EC2 の管理アカウント。
- `Identity_Manager_4.9.0_Linux.iso` と Designer がダウンロードされ、抽出され、Identity Manager コンポーネントのインスタンスで利用可能になっている。
- ローカルクライアントマシンから AWS EC2 インスタンスに接続するための SSH クライアント。

展開手順

Identity Manager コンポーネントは、要件に基づいてプライベートネットワークまたはパブリックネットワークに展開できます。354 ページの 図 28-1 「AWS EC2 での Identity Manager の展開」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 28-1 AWS EC2 での Identity Manager の展開



Identity Manager コンポーネントは、複数のコンポーネントが異なるサーバにどのように分散されるかに応じて、さまざまな組み合わせで展開することができます。ただし、展開手順はすべてのシナリオで同じです。

展開手順は、次の手順で構成されます。

- 355 ページの 「AWS 仮想プライベートクラウドの準備」
- 358 ページの 「インスタンスの作成と展開」
- 359 ページの 「EC2 インスタンスの準備」
- 361 ページの 「Identity Manager コンポーネントの設定」
- 362 ページの 「Identity Applications および Identity Reporting 用のデータベースの設定」
- 363 ページの 「Designer の設定」
- 364 ページの 「AWS EC2 ロードバランサの作成」
- 369 ページの 「(オプション) 登録済みのホストゾーンを使用したエイリアス DNS の作成」
- 370 ページの 「Identity Manager のコンポーネントへのアクセス」

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、Identity Manager で使用する AWS VPC をセットアップする一般的な手順の概要を説明します。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 [サービス] をクリックして、以下のサービスを作成します。

サービス	手順
VPC	1. [ネットワーキング & コンテンツ配信] で [サービス] > [VPC] をクリックします。 2. [VPC ウィザードの開始] をクリックします。 3. VPC 設定の種類を選択して、[選択] をクリックします。 4. フォームで詳細を指定し、[VPC の作成] をクリックします。 これにより、指定されたサイズのプライベートネットワークが作成されます。VPC およびサブネットの作成では、アドレス範囲に CIDR 表記を使用します。最大の VPC サイズは、/16 ネットワークです。 詳細については、『Amazon Virtual Private Cloud Documentation (https://docs.aws.amazon.com/vpc/index.html)』を参照してください。
重要 : [VPC ウィザードの開始] を使用して VPC を作成すると、VPC 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が作成されます。これらの項目を次のように表示または編集できます。	

サービス	手順
サブネット	図 28-1 に示すように Identity Manager コンポーネントを展開するには、VPC に 3 つのサブネットを作成します。たとえば、privateSN、publicSN1、および publicSN2 を作成します。 サブネットを作成するには、次の手順に従います。 1. 左側のメニューで、**[サブネット]** をクリックします。 2. **[サブネットの作成]** をクリックします。 3. **[名前タグ]** を指定してサブネットを識別します。 4. VPC 内で **[IPv4 CIDR ブロック]** を指定します。 たとえば、`10.0.0.0/24` と指定します。 さまざまなアベイラビリティゾーンでパブリックサブネットを作成する必要があります。 5. **[はい]** をクリックし、［作成］をクリックします。 6. (状況によって実行) パブリックサブネットの場合、自動割り当てパブリック IP アドレスを有効にします。 a. **[サブネットのアクション] > [自動割り当て IP 設定の変更]** を選択します。 b. **[パブリック IPv4 アドレスの自動割り当てを有効にする]** を選択します。 c. **[保存]** をクリックします。 これらの手順を繰り返して、追加のサブネットを作成します。
インターネットゲートウェイ	1. 左側のメニューで、**[インターネットゲートウェイ]** をクリックします。 2. **[インターネットゲートウェイの作成]** をクリックします。 3. **[名前タグ]** を指定して、**[作成]** をクリックします。 4. 新しく作成されたインターネットゲートウェイを選択し、VPC にアタッチします。 a. **[アクション] > [VPC にアタッチ]** を選択します。 b. リストから VPC を選択して、**[アタッチ]** をクリックします。

サービス	手順
ルートテーブル	1. 左側のメニューで、[ルートテーブル] をクリックします。 2. この VPC 用に自動的に作成されたルートテーブルを選択します。 3. [ルート] タブで : a. [編集] をクリックします。 b. [別のルートの追加] をクリックします。 c. [送信先] で、0.0.0.0/0 を指定します。 d. [ターゲット] で、この VPC に関連付けられているインターネットゲートウェイテーブルを選択します。インターネットゲートウェイを参照してください。 e. [保存] をクリックします。 4. [サブネットの関連付け] タブで : a. [編集] をクリックします。 b. この VPC に関連付けるサブネットを見つけて、[保存] をクリックします。
(オプション) ホストゾーン	登録済みのドメインがある場合は、そのドメインを使用して、以下のアクションを実行して、Identity Manager コンポーネントをホストすることができます。 1. [サービス] > [Route 53] > [ホストゾーン] をクリックします。 2. [Create Hosted Zone (ホストゾーンの作成)] をクリックして、以下のように詳細を指定します。 ◆ **［ドメイン名］** : ドメイン名を指定します。 ◆ **［コメント］** : コメントを追加します。 ◆ **［タイプ］** : ホストゾーンのタイプを指定します。 3. [作成] をクリックします。
Elastic IP アドレス	1. [サービス] > [EC2] をクリックします。 2. 左側のメニューで、[Elastic IPs] を選択します。 3. [新しいアドレスの割り当て] をクリックします。 4. [割り当て] をクリックします。 他のリソースによって使用されていない静的 IPv4 アドレスが割り当てられます。 5. [閉じる] をクリックします。

インスタンスの作成と展開

このセクションでは、Identity Manager の基本セットアップのインスタンスを作成および展開する手順の概要を説明します。これには、Identity Manager エンジン、Identity Applications、Reporting、ユーザアプリケーションデータベース、および Reporting データベースが含まれます。

以下の手順を実行して、Identity Manager コンポーネントのインスタンスを作成します。

1 **[サービス]** > **[EC2]** をクリックします。

2 **[Launch Instance (インスタンスの起動)]** をクリックします。

3 `SLES 15 SPx` またはイメージを選択します。

4 基本オペレーティングシステムおよび Identity Manager コンポーネントの展開の要件を満たすインスタンスタイプを選択します。Identity Manager 4.9 のシステム要件を参照してください。

5 **[Next: Configure Instance Details (次へ : インスタンスの詳細を設定する)]** をクリックします。

インスタンスが正しい VPC とサブネットを使用していることを確認してください。このページには、サブネット設定が自動入力されます。

フィールド	アクション
[Auto-assign Public IP]	パブリックの場合、`Enable` に設定します。この設定により、サブネット設定が自動的に入力されます。プライベートサブネットの場合、この値を `Disable` に設定します。

6 **[Next: Add Storage (次の手順 : ストレージの追加)]** をクリックします。

デフォルトのストレージサイズは 10GB です。要件に応じてストレージサイズを変更します。Identity Manager 4.9 のシステム要件を参照してください。

7 **[Next: Add Tags (次の手順 : タグの追加)]** をクリックします。

必要に応じてタグを追加します。タグを使用すると、インスタンスを整理できます。たとえば、各インスタンスに次の 2 つのタグを追加できます。

- インスタンスが何に使用されているかを示すタグ
- このマシンの所有者を示すタグ

8 **[Next: Configure Security Group (次へ : セキュリティグループの設定)]** をクリックします。

セキュリティグループは、インスタンスのグループの仮想ファイアウォールルールです。同じファイアウォール要件を持つインスタンスのグループごとに個別のセキュリティグループを作成することをお勧めします。

たとえば、Identity Manager エンジンのすべてのノードに 1 つのセキュリティグループ、Identity Applications のすべてのノードに 1 つのセキュリティグループ、Identity Reporting のすべてのノードに 1 つのセキュリティグループを設定することができます。デフォルトでは、新しいセキュリティグループはポート 22 の着信トラフィックのみを許可するため、SSH を使用してのみインスタンスに接続できます。

詳細については、「Amazon EC2 Security Groups for Linux Instances (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html)」を参照してください。

9 新しいセキュリティグループを作成します。そのセキュリティグループに名前と説明を指定します。

以下の Identity Manager コンポーネントをインストールする前に、追加のポートルールを追加します。

コンポーネント	ポート	説明
識別ボールト用の LDAP	TCP 636	セキュアな LDAP の通信に必要。
識別情報アプリケーション	TCP 8543	HTTPS 通信で Identity Applications にアクセスするために必要。
Identity Reporting	TCP 8643	HTTPS 通信で Identity Reporting にアクセスするために必要。
PostgreSQL データベース	TCP 5432	セキュアなデータベース通信で PostgreSQL にアクセスするために必要。

10 **[Review and Launch (レビューと起動)]** をクリックします。

11 詳細を確認したら、**[起動]** をクリックします。

12 既存のキーペアを選択するか、新しいキーペアを作成します。

このキーペアは、インスタンスへの SSH アクセスに使用されます。複数のマシンで同じキーペアを使用できます。

13 **[Download Key Pair (キーペアのダウンロード)]** をクリックします。

重要 : 秘密キーを使用してのみ、インスタンスに接続して管理できます。したがって、秘密キーをダウンロードした後に失くさないでください。

14 インスタンスの初期化時に作成される Elastic IP アドレスをアタッチします。

15 ステップ 1 ～ステップ 13 を繰り返して、他のインスタンスを作成します。

EC2 インスタンスの準備

インスタンスを起動し、ソフトウェアリポジトリを確認します。設定されたソフトウェアリポジトリを確認するには、以下の手順を実行します。

1 キーペアを使用してインスタンスにログインします。

2 `root` ユーザに切り替えます。

3 ご使用のオペレーティングシステムで以下の更新が利用可能であることを確認します。

SLES 上の SLES12-SP3-Pool and SLES12-SP3-Updates: 確認するには、`zypper lr -n` コマンドを実行します。

RHEL 上の rhui-REGION-rhel-server-releases/7Server/x86_64: 確認するには、`yum repolist` コマンドを実行します。

注 : オペレーティングシステムにリポジトリが存在しない場合、設定された Elastic IP アドレスがインスタンスにアタッチされていることを確認してから、インスタンスを再起動します。

4 オペレーティングシステムに以下の前提条件をインストールします。

SLES

zypper コマンドを使用して glibc-32bit ライブラリをインストールします。

Red Hat

yum install コマンドを使用して以下の前提条件をインストールします。

- unzip
- ksh
- bc
- glibc-*.i686
- libXau-1.0.8-2.1.el7.i686
- libxcb.i686
- libX11.i686
- libXtst.i686
- libXrender.i686
- libgcc.i686
- lsof

Identity Manager エンジンの場合、prerequisite.sh スクリプトを編集して、compat-libstdc++-33.x86_64 の出現を削除できます。このパッケージは、Identity Manager のインストールには必要なくなりました。

5 /etc/hosts および hostname を設定します。

5a インスタンスのプライベート IP アドレスを使用して、ファイアウォール内の Identity Manager サーバを保護します。

5b インスタンスに DNS 名を割り当て、hosts ファイルを更新します。

例 :

```
# 10.0.0.1 identityEngine.example.com identityEngine
```

5c hostname および domain name を設定します。

SLES

```
yast lan
```

RHEL

```
hostnamectl set-hostname idmengine.example.com
```

6 (状況によって実行) 暗号化された Elastic Block Store (EBS) ボリュームを作成して、クラウド内のデータを暗号化します。

6a [サービス] > [EC2] をクリックします。

6b [Elastic Block Store] で、[Volumes (ボリューム)] を選択し、[Create Volume (ボリュームの作成)] をクリックします。

6c ボリュームに必要なサイズを指定します。

6d [Encrypt this volume (このボリュームを暗号化する)] を選択し、[Create Volume (ボリュームの作成)] をクリックします。

6e リストで新しく作成されたボリュームを選択します。

6f [アクション] で、[ボリュームのアタッチ] をクリックして、EC2 インスタンスにボリュームをアタッチします。

6g これらの手順をインスタンスごとに繰り返します。

EBS の詳細については、「Amazon EBS」を参照してください。

7 オペレーティングシステムのツールを使用して、ボリュームをフォーマットし、パーティションをマウントします。

SLES

`yast disk` コマンドを実行して、ボリュームをフォーマットします。

RHEL

`mkfs` を実行してフォーマットし、`/etc/fstab` に追加します。詳細については、『Red Hat Enterprise Linux Deployment Guide』を参照してください。

注

- Identity Manager エンジンデータパーティションをマウントします。デフォルトでは、データパーティションは `/var/opt/novell/` です。
- 他の Identity Manager コンポーネントを `/opt/netiq/` にマウントします。

---

8 すべてのマシンの IP アドレスへの DNS ですべてのインスタンスの `/etc/hosts` ファイルを更新します。

Identity Manager コンポーネントの設定

Identity Manager コンポーネントをインストールする前に、以下の手順を実行します。

1 Identity Manager コンポーネントをインストールするインスタンスに `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

2 ダウンロード済みの `.iso` ファイルをマウントします。

3 (状況によって実行) Identity Applications と Identity Reporting のデータベースを作成します。詳細については、362 ページの 「Identity Applications および Identity Reporting 用のデータベースの設定」を参照してください。

4 `.iso` ファイルのルートディレクトリから、`./install.sh` コマンドを実行します。

5 使用許諾契約書を読み、「`y`」と入力して使用許諾契約に同意します。

6 カスタムインストールオプションを選択し、インスタンスにインストールするコンポーネントを選択します。コンポーネントを設定します。詳細については、86 ページの 表 5-2 「カスタム環境設定」を参照してください。

7 Identity Reporting および Identity Applications で `configupdate.sh` を実行して、すべてのクライアントを設定します。

Identity Applications および Identity Reporting 用のデータベースの設定

Identity Applications および Identity Reporting 用の PostgreSQLX データベースを外部サーバにインストールする場合は、インストールする前に次の手順を実行する必要があります。

1 `Identity_Manager_4.9.0_Linux.iso` をマウントした場所に移動します。

2 `/common/packages/postgres/` ディレクトリを見つけて、PostgreSQL をインストールします。

```
rpm -ivh netiq-postgresql-9.6.6-0.noarch.rpm
```

3 次のコマンドを実行して、グループを `postgres` ユーザに関連付けます。

```
/usr/sbin/usermod -a -G postgres postgres
```

4 `/etc/passwd` ファイルにある `postgres` ユーザのホームディレクトリパスを `/opt/netiq/idm/postgres/` に変更します。

4a `/etc/` ディレクトリに移動します。

4b `passwd` ファイルを編集します。

```
vi /etc/passwd
```

4c `postgres` ユーザのホームディレクトリを `/opt/netiq/idm/postgres/` に変更します。

5 `postgres` ユーザとしてログインします。

例 :

```
su - postgres
```

6 PostgreSQL のインストール場所に `data` ディレクトリを作成します。

`mkdir -p <POSTGRES_HOME>/data`。ここで、`<POSTGRES_HOME>` は `/opt/netiq/idm/postgres` です

例 :

```
mkdir -p /opt/netiq/idm/postgres/data
```

7 PostgreSQL `home` ディレクトリをエクスポートします

`export PGHOME=<postgres ホームディレクトリパス >`

例 :

```
export PG_HOME=/opt/netiq/idm/postgres
```

8 PostgreSQ パスワードをエクスポートします。

```
export PGPASSWORD=<enter the database password>
```

9 データベースを初期化します。

```
"LANG=en_US.UTF-8 <POSTGRES_HOME>/bin/initdb -D <POSTGRES_HOME>/data"
```

例 :

```
"LANG=en_US.UTF-8 /opt/netiq/idm/postgres/bin/initdb -D /opt/netiq/idm/
postgres/data"
```

10 /opt/netiq/idm/postgres/ ディレクトリに以下のコンポーネントのデータベースを作成します。

識別情報アプリケーション

```
$ createdb idmuserappdb
$ psql -s idmuserappdb
# create user idmadmin password 'somepassword';
# GRANT ALL PRIVILEGES ON DATABASE idmuserappdb TO idmadmin;
# ALTER DATABASE idmuserappdb OWNER TO idmadmin;
```

Identity Reporting

```
$ createdb idmrptdb
```

11 postgres ユーザとしてログアウトします。

12 PostgreSQL インスタンスが localhost 以外のネットワークインスタンスでリスンできるように、postgresql.conf ファイルを変更します。

12a /opt/netiq/idm/postgres/data/ ディレクトリに移動します。

12b postgresql.conf ファイルを編集します。

```
vi postgresql.conf
```

12c ファイルに次の行を追加します。

```
listen_addresses = '*'
```

13 <postgres ホームディレクトリパス >/data の下に pg_log ディレクトリを作成します。

例 :

```
mkdir -p /opt/netiq/idm/postgres/data/pg_log
```

14 pg_log ディレクトリの許可を変更します。

```
chown -R postgres:postgres <postgres ディレクトリパス >/data/pg_log
```

例 :

```
chown -R postgres:postgres /opt/netiq/idm/postgres/data/pg_log
```

15 PostgreSQL サービスを開始します。

```
systemctl start netiq-postgresql
```

これにより、新しい PostgreSQL サービスが開始されます。

Designer の設定

Designer を使用するには、Windows マシンに Designer をインストールする必要があります。

1 パブリックサブネットで、サポートされている Windows インスタンスを起動します。

Windows セキュリティグループの場合、`rdesktop` ポートのみを使用します。たとえば、`3389` と指定します。

2 Windows インスタンスに Designer をインストールします。詳細については、『NetIQ Identity Manager インストールおよびアップグレードガイド (Windows 用)』の「*Designer のインストール*」を参照してください。

AWS EC2 ロードバランサの作成

ロードバランサを作成して、Identity Manager コンポーネント間で着信要求の負荷のバランスを取ることができます。ロードバランサを使用して、Identity Manager サーバをパブリックアクセスから保護することができます。

以下の手順では、サンプル展開シナリオ用にロードバランサを設定するために必要な設定の詳細について説明します。

セキュア通信を使用するためにロードバランサの証明書を作成する

ロードバランサではこの証明書を使用して、Identity Manager コンポーネント間のセキュア通信を確立します。以下の 3 つの方法でロードバランサの証明書を作成できます。

- 364 ページの 「AWS Certificate Manager (ACM) を使用する」
- 364 ページの 「外部証明書を ACM にアップロードする」
- 364 ページの 「外部証明書を IAM にアップロードする」

AWS Certificate Manager (ACM) を使用する

1 **[サービス] > [Certificates Manager (証明書マネージャ)]** をクリックします。

2 **[Request Certificate (証明書の要求)]** をクリックします。

3 証明書を作成する DNS 名を指定します。

4 DNS 権限を確認します。

外部証明書を ACM にアップロードする

1 **[サービス] > [Certificates Manager (証明書マネージャ)]** をクリックします。

2 **[証明書のインポート]** をクリックします。

3 証明書の詳細を指定します。

外部証明書を IAM にアップロードする

1 **[サービス]** をクリックします。

2 **[Security, Identity & Compliance (セキュリティ、アイデンティティ & コンプライアンス)]** で、**[IAM]** をクリックします。

3 IAM API を使用して証明書の詳細を指定します。詳細については、「Uploading Server Certificate Using IAM API」を参照してください。

ターゲットグループの作成

ターゲットグループは、ロードバランサを、負荷が分散されるインスタンス (ターゲット) の IP アドレスに関連付ける方法を提供します。

ターゲットグループを作成するには、次の手順に従います。

1 EC2 ダッシュボードで、[ロードバランシング] の [ターゲットグループ] をクリックします。

2 [ターゲットグループの作成] をクリックします。

3 次の詳細を指定します。

フィールド	説明
[ターゲットグループ名]	ターゲットグループの名前を指定します。 このターゲットグループを設定するコンポーネントの名前を指定できます。たとえば、Identity Applications、Identity Reporting です。
[プロトコル]	[HTTPS] を選択します。
[ポート]	サーバがリスンするように設定されているポートを指定します。 以下は、さまざまな Identity Manager コンポーネントに使用されるポート値の例です。 ◆ **識別情報アプリケーション :** 8543 ◆ **Identity Reporting:** 8643
[ターゲットタイプ]	[インスタンス] を選択します。
[VPC]	Identity Manager コンポーネントのインスタンスに選択したものと同じ VPC を選択します。
ヘルスチェック設定	
[プロトコル]	[HTTPS] を選択します。 ロードバランサは、ヘルスチェックの実行中にこのプロトコルを使用します。
[パス]	ヘルスチェックの送信先を指定します。 ヘルスチェックを実行するための Identity Manager コンポーネントのデフォルトパスは次のとおりです。 ◆ **識別情報アプリケーション :** `/idmdash/index.html` ◆ **Identity Reporting:** `/IDMRPT/index.html`
[ヘルスチェックの詳細設定]	デフォルト値のままにします。

4 [作成] をクリックします。

5 セッションの持続性を有効にします。

5a 作成したターゲットグループを選択します。

5b [説明] タブで、[Edit attributes (属性の編集)] をクリックします。

5c [Stickiness (持続性)] に対して [Enable (有効化)] を選択します。

6 これらの手順を繰り返して、各アプリケーションのターゲットグループを作成します。

注 : SSPR が別のサーバにインストールされている場合は、このコンポーネント用に別個のターゲットグループを作成してください。

ロードバランサの作成

ロードバランサを作成するには、次の手順に従います。

1 左側のメニューで、[ロードバランサ] をクリックします。

2 [ロードバランサの作成] をクリックします。

3 [Application Load Balancer] で [作成] をクリックします。

4 次の詳細を指定します。

フィールド	説明
[名前]	ロードバランサの名前を指定します。
[スキーム]	[internet-facing] を選択します。
[リスナー]	ロードバランサにリスナーを追加するには、[リスナーの追加] をクリックします。 以下のようにリスナーポートを指定します。 **Identity Applications の場合 :** ◆ [ロードバランサプロトコル]: HTTPS ◆ [ロードバランサポート]: 8543 **Identity Reporting の場合 :** ◆ [ロードバランサプロトコル]: HTTPS ◆ [ロードバランサポート]: 8643
[アベイラビリティゾーン]	1. Identity Manager コンポーネント用に以前に作成したものと同じ [VPC] を選択します。 2. パブリックサブネットが利用可能な [アベイラビリティゾーン] を選択します。 **注 :** 少なくとも 2 つのサブネットを選択する必要があります。
[タグ]	(オプション) タグを追加して、ロードバランサを識別することができます。

5 [Next: Configure Security Settings (次へ : セキュリティの設定)] をクリックします。

6 HTTPS プロトコルを使用するように証明書の詳細を指定します。次のいずれかを実行できます。

- 364 ページの 「セキュア通信を使用するためにロードバランサの証明書を作成する」で作成した証明書タイプを選択します。
- 証明書を IAM または ACM にアップロードします - 証明書の詳細を指定します。

7 [Next: Configure Security Groups (次へ : セキュリティグループの設定)] をクリックします。

8 [セキュリティグループの割り当て] で、[新しいセキュリティグループを作成する] を選択します。

9 (オプション) ロードバランサの名前と説明を指定します。

10 設定されたリスナーにトラフィックをルーティングするセキュリティグループにルールを追加します。

フィールド	説明
[タイプ]	[カスタム TCP ルール] を選択します。
[プロトコル]	ルールに使用されるプロトコルタイプが表示されます。
[ポート範囲]	Identity Manager コンポーネントのポート範囲を選択します。 • **識別情報アプリケーション :** 8543 • **Identity Reporting:** 8643
[ソース]	Identity Manager コンポーネントが展開されているインスタンスに接続するには、[任意の場所] を選択します。

11 [Next: Configure Routing (次へ : ルーティングの設定)] をクリックします。

12 [ターゲットグループ] で、以下の詳細を指定します。

フィールド	説明
[ターゲットグループ]	[既存のターゲットグループ] を選択します。このリストには、365 ページの 「ターゲットグループの作成」で Identity Manager コンポーネント用に作成されたターゲットグループが表示されます。
[名前]	リストからターゲットグループを選択します。 ここで選択できるターゲットグループは 1 つだけです。たとえば、Identity Applications 用に作成したターゲットグループを選択します。 ロードバランサを作成したら、HTTPS プロトコル用に設定されたターゲットグループを使用するように、リスナーポート 8443 を変更する必要があります。このセクションの 368 ページのステップ 18 を参照してください。
[プロトコル]	指定されたターゲットグループで設定した値が入力されます。値が正しくリストされていることを確認してください。

フィールド	説明
[ポート]	指定されたターゲットグループで設定した値が入力されます。値が正しくリストされていることを確認してください。
[ターゲットタイプ]	指定されたターゲットグループで設定した値が入力されます。正しい値がリストされていることを確認してください。

13 [ヘルスチェック] で、以下の詳細を確認します。

フィールド	説明
[プロトコル]	ステップ 12 で選択したターゲットグループの設定に基づいて、[HTTPS] または [HTTP] が入力されます。 365 ページの 「ターゲットグループの作成」を参照してください。
[パス]	ステップ 12 で選択されたターゲットグループで設定したヘルス URL が入力されます。 365 ページの 「ターゲットグループの作成」を参照してください。
[ヘルスチェックの詳細設定]	デフォルト値のままにします。

14 [Next: Register Targets (次へ : ターゲットの登録)] をクリックします。

選択されたターゲットグループに登録されているすべてのターゲットのリスト。このリストは、ロードバランサを作成した後にのみ変更できます。

15 [Next: Review (次へ : 確認)] をクリックします。

16 ロードバランサの詳細が正しいことを確認します。

17 [作成] をクリックし、[閉じる] をクリックします。

18 (状況によって実行) Identity Manager コンポーネントのリスナーポートの作成をスキップした場合、または新しいリスナーポートを追加する場合は、適切なターゲットグループを使用するようにリスナーポートを更新します。

18a 作成したロードバランサを選択します。

18b [リスナー] タブを選択します。

18c [リスナーの追加] をクリックして、各リスナーに必要な詳細を指定します。ステップ 4 を参照してください。

18d ロードバランサに使用される証明書を選択します。364 ページの 「セキュア通信を使用するためにロードバランサの証明書を作成する」を参照してください。

18e [作成] をクリックします。

SSPR が別のマシンで設定されている場合、リスナーを追加する必要がある場合があります。

重要 : 分散セットアップで単一のロードバランサを使用するには、別の DNS エイリアスレコードを作成して、セットアップ内のサーバを区別します。それ以外の場合は、Web アプリケーションごとに個別のロードバランサを作成します。

(オプション) 登録済みのホストゾーンを使用したエイリアス DNS の作成

登録済みのサイトがある場合は、そのサイトを使用して、Identity Manager コンポーネントごとに個別のレコードセットを作成できます。

1 [サービス] > [Route 53] をクリックします。

2 左側のメニューで [ホストゾーン] をクリックし、AWS EC2 サービスの設定中に作成されるホストゾーンを選択します。355 ページの 「AWS 仮想プライベートクラウドの準備」 を参照してください。

3 [レコードセットに移動] をクリックします。

4 [レコードセットの作成] をクリックします。

フィールド	説明
[名前]	レコードセットに意味のある名前を指定します。 たとえば、Identity Applications レコードセットの名前を `rbpm` と指定します。
[タイプ]	[A – IPv4 アドレス] を選択します。
[別名]	[はい] を選択します。
[エイリアスターゲット]	Identity Manager コンポーネントを接続するように設定されたロードバランサを選択します。
[ルーティングポリシー]	[Simple (シンプル)] を選択します。

5 [作成] をクリックします。

6 ステップ 4 とステップ 5 を繰り返して、各 Identity Manager インスタンスのレコードセットを作成します。

7 Identity Applications、Identity Reporting、および OSP インスタンスで `configupdate.sh` を実行し、パブリック DNS 名を使用して SSO 設定を更新します。

8 Tomcat を再起動します。

9 パブリック DNS を使用してアプリケーションにアクセスして、設定を確認します。

```
https://<public-DNS-name>:<port>/<application-context-name>
```

Identity Manager のコンポーネントへのアクセス

ロードバランサのパブリック DNS 名またはエイリアス DNS レコードセットを使用して、Identity Manager インスタンスにアクセスするとができます。Identity Manager インスタンスが相互に通信できるようにするには、各インスタンスの /etc/hosts ファイルを編集し、そのホスト名をプライベート IP アドレスに解決するエントリを追加します。

以下のインスタンスを更新して、他のインスタンスに内部的にアクセスします。

インスタンス	説明
OSP	OSP インスタンスには、パスワードをリセットするために SSPR インスタンスへのアクセスが必要です。 **ホストファイルの場所 :** /etc/hosts 次のエントリで hosts ファイルを変更します。 <IP_address> <Private_DNS_Name> <Public_DNS_Name> 例 : 10.0.1.5 sspr.privatedns.local sspr.publicdns.com
識別情報アプリケーション	Identity Applications インスタンスでは、ログインのために OSP インスタンスへのアクセスが必要です。 **ホストファイルの場所 :** /etc/hosts 次のエントリで hosts ファイルを変更します。 <IP_address> <Private_DNS_Name> <Public_DNS_Name> 例 : 10.0.1.6 osp.privatedns.local osp.publicdns.com
Identity Reporting	Identity Reporting インスタンスでは、ログインのために OSP インスタンスへのアクセスが必要です。 **ホストファイルの場所 :** /etc/hosts 次のエントリで hosts ファイルを変更します。 <IP_address> <Private_DNS_Name> <Public_DNS_Name> 例 : 10.0.1.6 osp.privatedns.local osp.publicdns.comm

セキュリティ上の考慮事項

AWS クラウドに Identity Manager コンポーネントを展開するために、以下の考慮事項を確認することをお勧めします。

- Identity Manager コンポーネントは、パブリックアクセスのないプライベートネットワークで設定されているか、Elastic IP アドレスにアタッチされています。
- Identity Applications、Identity Reporting などの Web アプリケーションには、ロードバランサを介してアクセスされます。
- Identity Manager コンポーネントは、セキュアな通信チャネルを使用するように設定されています。
- データは、コンポーネントごとに個別の暗号化された EBS ボリュームで設定されます。
- Identity Manager サーバでは以下のポートが利用可能であり、サブネット内で使用できます。

ポート	アプリケーション
636	LDAP
8543	識別情報アプリケーション
8643	Identity Reporting
5432	PostgreSQL

29 ハイブリッド Identity Manager のシナリオの例

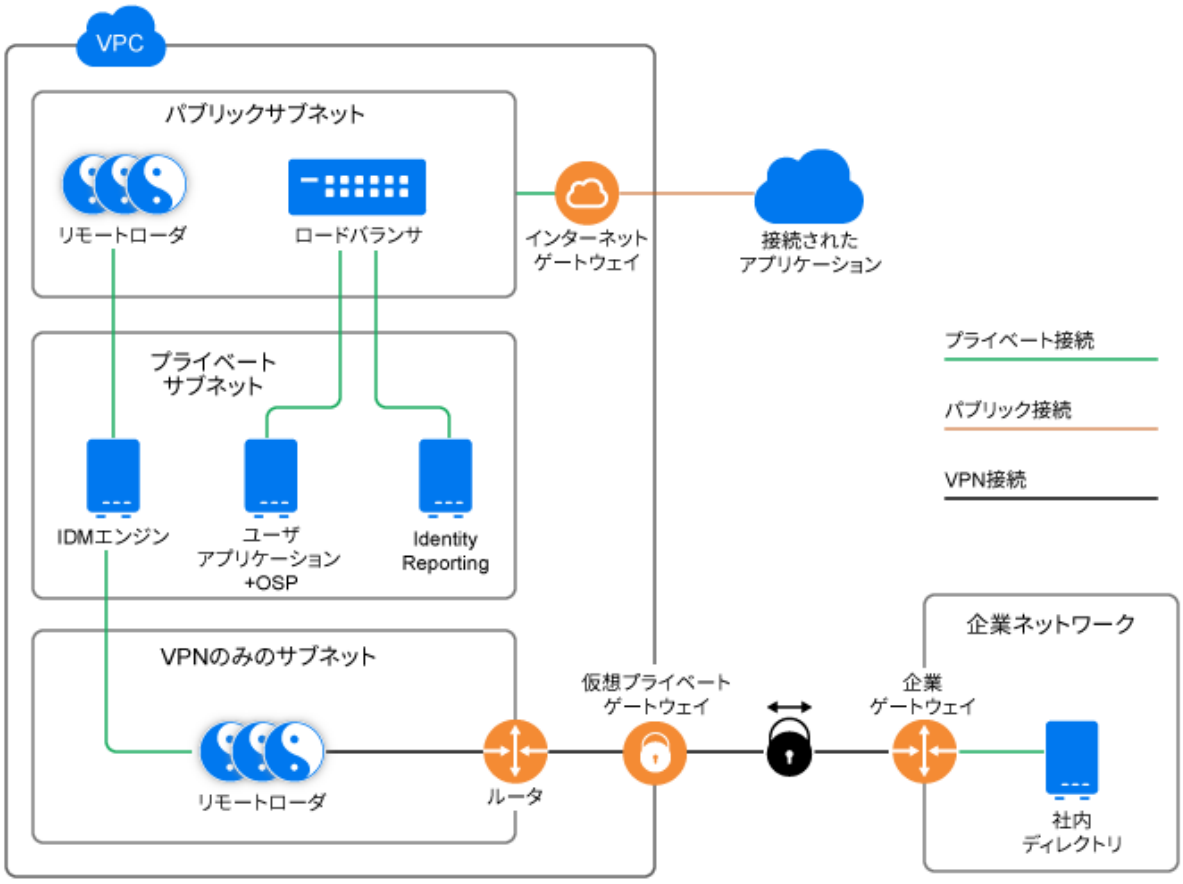
エンタープライズプレミスと AWS クラウド間で識別情報がシームレスに同期される Identity Manager コンポーネントを設定できます。このタイプのハイブリッドシナリオを実装するには、AWS サブネットとエンタープライズネットワークの間に VPN 接続を設定する必要があります。このセクションでは、以下のハイブリッド Identity Manager シナリオについて説明します。

- 373 ページの 「リモートローダによる接続の使用」
- 374 ページの 「マルチサーバドライバセット接続の使用」
- 376 ページの 「eDirectory ドライバ接続の使用」

リモートローダによる接続の使用

リモートローダは、VPN が設定されているサブネットにインストールされます。同期を有効にすると、リモートローダドライバシムはエンタープライズネットワークで実行されているアプリケーションに接続し、AWS クラウド上の Identity Manager とアプリケーションの間で識別情報を同期します。

図 29-1 リモートローダ接続を使用したハイブリッドシナリオ



このシナリオは、接続されているアプリケーションが少ないシステムに適しており、リモートローダのリスナーポートを開く必要があります。この接続では、同期中に設定済みの属性のみが通過できます。

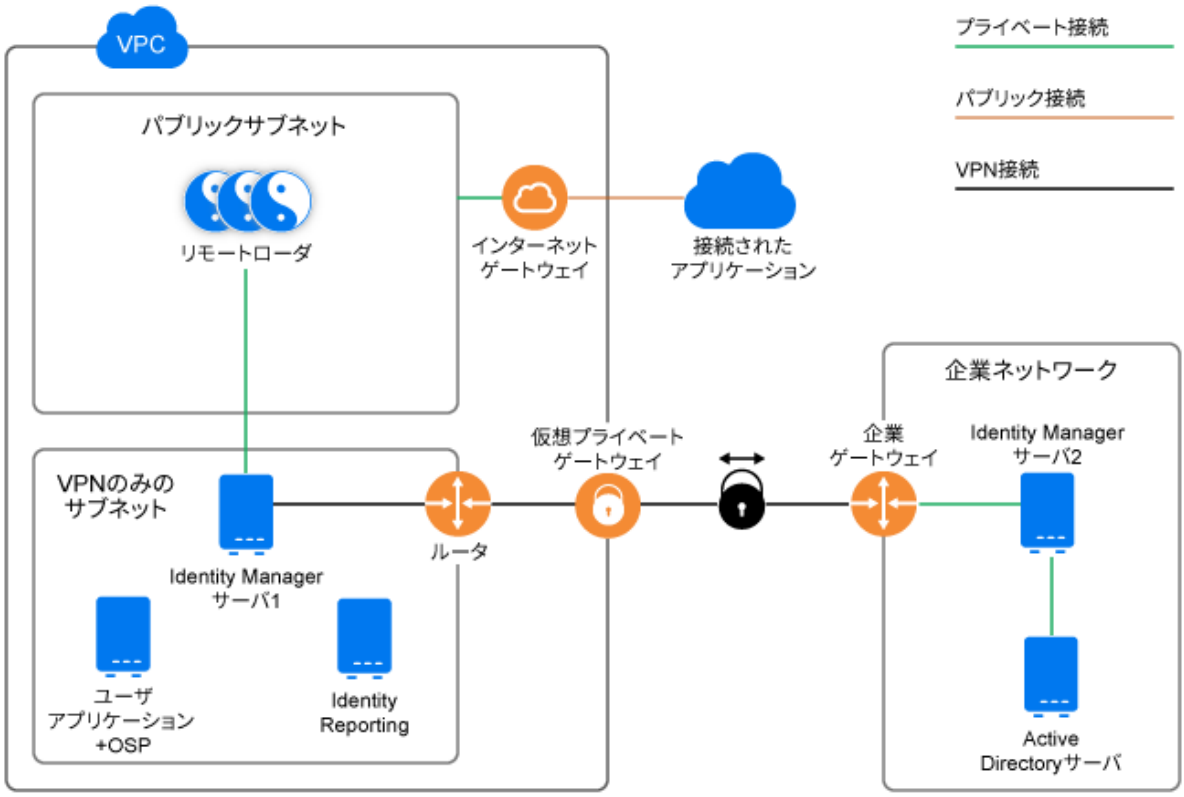
制限：

- このシナリオは、リモートローダの使用をサポートするドライバに適用されます。
- 多数のアプリケーションが接続されていると、リモートローダへのトラフィックが増えます。

マルチサーバドライバセット接続の使用

このシナリオでは、1 つのサーバが AWS クラウドにインストールされ、もう 1 つのサーバがエンタープライズプレミスにインストールされている、少なくとも 2 つの Identity Manager サーバが、同じドライバセットを使用します。これには、VPN 接続を介して識別情報を同期するために Identity Vault レプリケーションチャネルを使用する完全なレプリカサーバが含まれます。エンタープライズネットワークまたは AWS クラウド上で実行されている Identity Manager サーバは、それぞれの接続アプリケーション間で識別情報を同期します。

図 29-2 マルチサーバドライバセット接続を使用したハイブリッドシナリオ

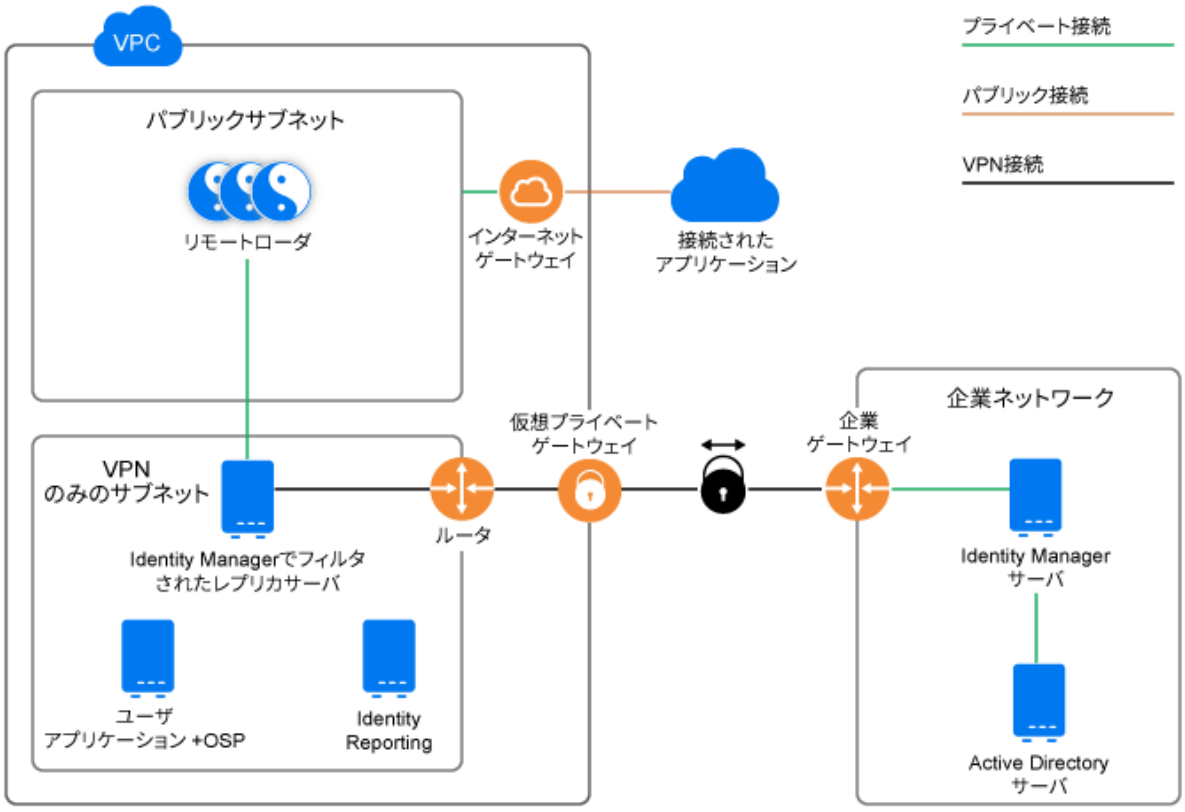


この設定では、両側で Identity Manager サーバ間のデルタ変更を同期するためにのみ VPN 接続を使用します。

フィルタされたレプリケーションの使用

これは、マルチサーバドライバセットシナリオの変形版であり、AWS クラウド内のサーバ上のデータパーティションのフィルタされた読み取り / 書き込みレプリカが含まれます。ドライバセットパーティションの場合、どちらの側でも常に完全なレプリカパーティションを使用する必要があります。

図 29-3 制御されたレプリケーションを使用したハイブリッドシナリオ

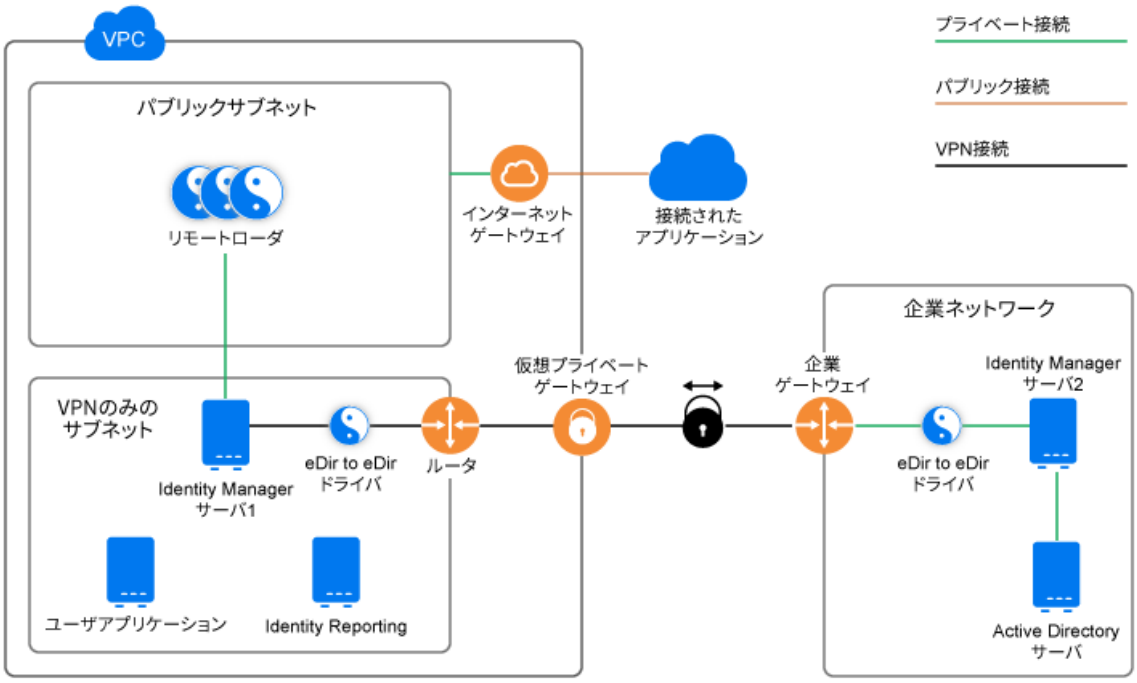


このシナリオでは、同期する属性をより詳細に制御できます。たとえば、機密属性が AWS クラウド上の Identity Manager サーバと同期するのを防ぐことができます。

eDirectory ドライバ接続の使用

このシナリオは、1 つのツリーが AWS クラウドに属し、他のツリーがエンタープライズネットワークに属する 2 つの別個の eDirectory ツリーに Identity Manager サーバがインストールされている場合に適しています。この設定では、eDirectory ドライバを使用して、VPN 接続を介して AWS クラウドとエンタープライズネットワーク間の識別情報を同期します。エンタープライズネットワークまたは AWS クラウドで実行されている Identity Manager サーバは、それぞれの接続アプリケーション間で識別情報を同期します。

図 29-4 eDirectory ドライバ接続を使用したハイブリッドシナリオ



AWS クラウドとエンタープライズネットワーク間の通信は制限されています。デルタ変更のみを同期します。ドライバフィルタを設定することにより、同期する属性を制御できます。ポリシーエンジンを利用して、属性を同期するための追加のコントロールを定義することもできます。たとえば、パスワード属性の同期を制限し、複数のユーザが異なるパスワードを使用して AWS クラウドとエンタープライズネットワークから Identity Manager サーバにアクセスできるようにします。

XI 高可用性のための Identity Manager の展開

高可用性により、データ、アプリケーション、サービスなどの重要なネットワークリソースを効率的に管理できます。NetIQ では、VMWare Vmotion など、クラスタリングまたは Hypervisor クラスタリングを介した Identity Manager ソリューションの高可用性をサポートしています。高可用性環境を計画する際には、次の考慮事項が適用されます。

- 高可用性環境には次のコンポーネントをインストールできます。
  - 識別ボールト
  - Identity Manager エンジン
  - リモートローダ
  - Identity Reporting を除く、識別情報アプリケーション
- Identity Manager 環境のネットワークリソースの可用性を管理するには、最新のパッチがインストールされた SUSE Linux Enterprise Server (SLES) 12 SP3 とともに SUSE Linux Enterprise High Availability Extension を使用します。
- アイデンティティボールトをクラスタ環境で実行する場合は、Identity Manager エンジンもクラスタ化されます。

注 : Identity Manager は、Identity Vault と Identity Applications の間の LDAP または LDAPS 通信の負荷分散をサポートしていません。

参照する情報 ...	代わりの手段 ...
Identity Manager コンポーネントのサーバ設定の決定	『NetIQ Identity Manager Overview and Planning Guide』の「*High Availability Configuration*」を参照してください。
クラスタ内のアイデンティティボールトの実行	Identity Manager のクラスタ展開ソリューションのサンプル 『*NetIQ eDirectory Installation Guide*』の「Deploying eDirectory on High Availability Clusters」
クラスタ内の識別情報アプリケーションの実行	Tomcat アプリケーションサーバ上へのアイデンティティアプリケーションクラスタ展開ソリューションのサンプル

お客様の Identity Manager 環境における高可用性と障害復旧の実装に関する詳しい情報は、NetIQ テクニカルサポート (https://www.netiq.com/support/) にお問い合わせください。

この次の章では、高可用性環境で Identity Manager コンポーネントをインストールおよび設定する手順を説明します。

- 381 ページの第 30 章「クラスタ環境における Identity Manager のインストールの準備」
- 385 ページの第 31 章「Identity Manager のクラスタ展開ソリューションのサンプル」
- 395 ページの第 32 章「Tomcat アプリケーションサーバ上へのアイデンティティアプリケーションクラスタ展開ソリューションのサンプル」

30 クラスタ環境における Identity Manager のインストールの準備

- 前提条件
- 識別情報アプリケーションで使用するクラスタの準備

前提条件

- 381 ページの 「識別ボールト」
- 382 ページの 「識別情報アプリケーション」
- 382 ページの 「Identity Applications のデータベース」

識別ボールト

NetIQ では、クラスタ環境に識別ボールトをインストールする前に、次の考慮事項を確認することをお勧めします。

- すべての識別ボールトおよび NICI データを格納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が必要です。
  - 識別ボールト DIB は、クラスタ共有ストレージに存在している必要があります。識別ボールトの状態データは、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。
  - 各クラスタノード上のルート識別ボールトインスタンスは、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。
  - さらに、共有 NICI (NetIQ International Cryptographic Infrastructure) データを共有して、サーバ固有のキーがクラスタノード間で複製されるようにすることも必要です。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。
  - NetIQ では、他のすべての eDirectory 設定およびログデータを共有ストレージに格納することをお勧めします。
- 仮想 IP アドレスが必要です。
- (状況によって実行)識別ボールトのサポート構造としてeDirectoryを使用している場合、`nds-cluster-config` ユーティリティでは、ルート eDirectory インスタンスの設定のみがサポートされます。クラスタ環境内での eDirectory の複数インスタンスの環境設定と、eDirectory の非ルートインストールはサポートされていません。

クラスタ環境内におけるアイデンティティボールドのインストールの詳細については、『*NetIQ eDirectory Installation Guide*』の「Deploying eDirectory on High Availability Clusters」を参照してください。

識別情報アプリケーション

Tomcat がサポートする環境に識別情報アプリケーション用のデータベースをインストールできますが、次の検討事項があります。

- クラスタには、固有のクラスタパーティション名、マルチキャストアドレス、およびマルチキャストポートが必要です。固有の識別子を使用して複数のクラスタを分離することによって、パフォーマンスの問題や異常な動作の発生を防ぎます。
  - クラスタの各メンバーで、識別情報アプリケーションのデータベースの待ち受けポートのポート番号として同一の値を指定する必要があります。
  - クラスタの各メンバーで、識別情報アプリケーションのデータベースをホストするサーバのホスト名または IP アドレスとして同一の値を指定する必要があります。
- クラスタ内のサーバの時刻を同期化する必要があります。サーバの時刻が同期していない場合、セッションタイムアウトが早期に発生し、HTTP セッションのフェールオーバーが正しく機能しない可能性があります。
- 同一ホストでは、複数のブラウザタブまたは複数のブラウザセッションで複数のログインを使用しないことをお勧めします。一部のブラウザはすべてのタブとプロセス間で cookie を共有します。そのため、複数のログインを行うと、(1 台のコンピュータを複数ユーザが共有することで認証機能に予期しない動作が発生するおそれがあるだけでなく) HTTP セッションのフェールオーバーの際に問題が発生する可能性があります。
- クラスタノードは同じサブネットに存在しています。
- フェールオーバープロキシまたは負荷分散ソリューションは、別のコンピュータにインストールされています。

Identity Applications のデータベース

データベースクラスタリングは、個々のデータベースサーバの機能です。クラスタリングは本製品の機能とは無関係なので、NetIQ はどのクラスタ化データベース設定についても公式なテストを実行していません。したがって、次の警告付きで、クラスタ化データベースサーバをサポートします。

- デフォルトで、最大接続数は 100 に設定されます。この値は、クラスタ内のワークフロー要求を処理するには小さすぎる場合があります。次の例外が表示される場合があります。

  ```
  (java.sql.SQLException: Data source rejected establishment of
  connection, message from server: "Too many connections."
  ```

  最大接続数を増やすには、`my.cnf` ファイルにある `max_connections` 変数をより高い値に設定してください。
- クラスタ化データベースサーバの一部の機能または側面を無効にする必要がある場合があります。たとえば、トランザクションレプリケーションは、重複キーを挿入しようとしたときに制約違反になるので、特定のテーブルでは無効にする必要があります。

- クラスタ化データベースサーバのインストール、設定、または最適化について、クラスタ化データベースサーバへの弊社製品のインストールも含めて、支援を提供することはありません。
- クラスタ化データベース環境で弊社製品を使用する際に問題が発生した場合は、その解決に向けて最善の努力を尽くします。複雑な環境におけるトラブルシューティング方法の多くは、問題を解決するために連携作業を必要とします。NetIQ は、自社製品の分析、プラニング、およびトラブルシューティングを実行するための専門知識を提供します。他のサードパーティ製品の分析、プラニング、およびトラブルシューティングを実行するための専門知識は、お客様から提供していただく必要があります。NetIQ 製品の問題とクラスタ設定の潜在的な問題を切り分けるために、お客様には問題の再現または非クラスタ化環境におけるコンポーネントの動作の分析をお願いします。

識別情報アプリケーションで使用するクラスタの準備

識別情報アプリケーションは HTTP のセッションレプリケーションとセッションフェールオーバーをサポートします。つまり、ノードでセッション処理中にノードにエラーが発生した場合、そのセッションは中断されることなく、同じクラスタ内の別のサーバ上で再開されます。クラスタに識別情報アプリケーションをインストールする前に、環境を準備する必要があります。

- 383 ページの 「Tomcat 環境のクラスタグループの理解」
- 383 ページの 「ワークフローエンジン ID のシステムプロパティの設定」

Tomcat 環境のクラスタグループの理解

ユーザアプリケーションクラスタグループは UUID 名を使用して、ユーザがサーバに追加する他のクラスタグループと競合するリスクを最小限に抑えます。ユーザアプリケーション管理機能を使用して、ユーザアプリケーションクラスタグループの環境設定を変更できます。クラスタ設定の変更がサーバノードで有効になるのは、そのノードを再起動した場合のみです。

ワークフローエンジン ID のシステムプロパティの設定

クラスタ内で識別情報アプリケーションをホストする各サーバでは、ワークフローエンジンを実行できますクラスタとワークフローエンジンのパフォーマンスを保証するために、クラスタ内のすべてのサーバが同じパーティション名とパーティション UDP グループを使用する必要があります。また、クラスタ内の各サーバを起動する際に一意のワークフロー

エンジン ID を指定する必要があります。なぜなら、ワークフローエンジンのクラスタリングは、識別情報アプリケーションのキャッシュフレームワークとは独立して動作するからです。

ワークフローエンジンが適切に動作することを保証するには、Tomcat のシステムプロパティを設定する必要があります。

1. クラスタ内の識別情報アプリケーションサーバごとに、新しい JVM システムプロパティを作成します。
2. このシステムプロパティに `com.novell.afw.wf.`*`engine-id`* という名前を付けます。ここで、engine-id は一意な値です。
3. Identity Applications と Identity Reporting は、Identity Vault と Identity Manager エンジンにあります。

31 Identity Manager のクラスタ展開ソリューションのサンプル

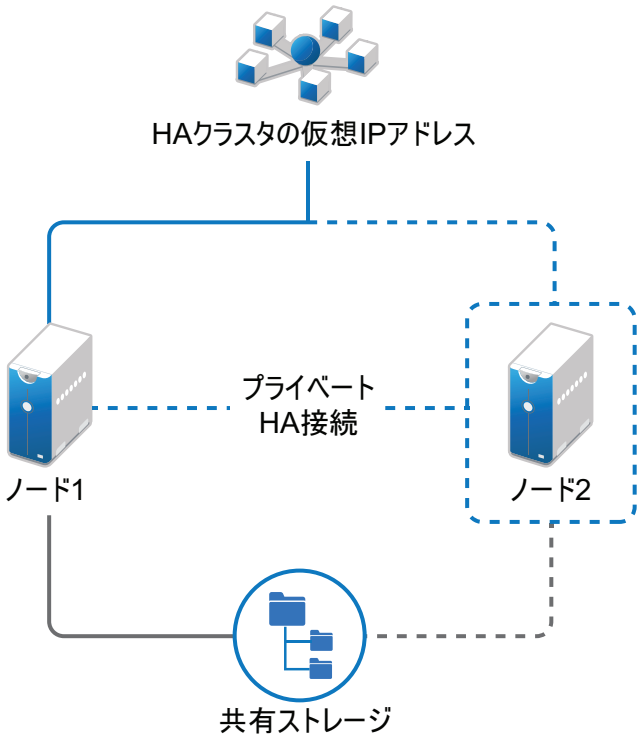
この章では、共有ストレージを使用する、サポートされている SUSE Linux Enterprise Server (SLES) クラスタ環境に eDirectory と Identity Manager を設定する段階的な手順およびクラスタ化された Identity Manager の展開例について説明します。

- 385 ページの 「前提条件」
- 386 ページの 「インストール手順」

共有ストレージを使用する運用レベルの Linux 高可用性 (HA) ソリューションの場合、クラスタにフェンシングメカニズムを実装することをお勧めします。クラスタにフェンシングメカニズムを実装する方法は複数ありますが、この例では、スプリットブレインディテクタ (SBD) を使用する STONITH リソースを使用します。

385 ページの 図 31-1 に、クラスタ展開ソリューションのサンプルを示します。

図 31-1 *クラスタ展開ソリューションのサンプル*



前提条件

- SLES 12 SP3 64 ビット版を実行する 2 つのサーバをノードとして使用

- SLES 12 SP3 64 ビット版を実行する 1 つのサーバを iSCSI サーバとして使用
- SLES12 SP3 64 ビット版 HA Extension の ISO イメージファイル
- 6 個の静的 IP:
  - ノードごとに 2 個の静的 IP アドレス。
  - クラスタに1個の静的IPアドレス。このIPアドレスは現在eDirectoryを実行しているノードに動的に割り当てられます。
  - iSCSI サーバに 1 個の IP アドレス。

インストール手順

このセクションでは、クラスタ環境に Identity Manager をインストールする手順について説明します。SLES High Availability Extension の設定の詳細については、『SUSE Linux Enterprise High Availability Extention』ガイドを参照してください。

- iSCSI サーバの設定
- すべてのノード上の iSCSI Initiator の設定
- 共有ストレージのパーティション化
- HA Extension のインストール
- softdog ウォッチドッグのセットアップ
- HA クラスタの設定
- クラスタノードでのアイデンティティボールトおよび Identity Manager エンジンのインストールと設定
- アイデンティティボールトのリソースの設定
- eDirectory および共有ストレージの子リソースのプリミティブ
- 場所制約スコアの変更

iSCSI サーバの設定

iSCSI ターゲットは、クラスタ内のすべてのノードの共通ストレージとして設定するデバイスです。これは、iSCSI Initiator がイーサネット接続経由でリモートアクセスできるように Linux サーバ上に作成される仮想ディスクです。iSCSI Initiator は、ターゲット (iSCSI) に接続してサービスを要求するように設定されているクラスタ内のノードです。iSCSI ターゲットは、イニシエータとして動作するホストがターゲットに接続できるように、常時稼動している必要があります。iSCSI サーバに iSCSI ターゲットをインストールする前に、iSCSI ターゲットに共通ストレージに使用できる十分な容量があることを確認します。SLES 12 SP3 をインストールした後、他の 2 つのノードに iSCSI Initiator パッケージをインストールします。

SLES 12 SP3 をインストールする際に、次の操作を実行します。

1. 独立したパーティションを作成し、そのパーティションパスを iSCSI 共有ストレージパーティションとして指定します。
2. iSCSI ターゲットパッケージをインストールします。

iSCSI サーバを設定するには :

1 ターゲットサーバ上にブロックデバイスを作成します。
2 端末で `yast2 disk` コマンドを実行します。
3 新しい Linux パーティションを作成し、[フォーマットしない] を選択します。
4 [パーティションをマウントしない] を選択します。
5 パーティションサイズを指定します。
6 端末で `yast2 iscsi-server` または `yast2 iscsi-lio-server` コマンドを入力します。
7 [サービス] タブをクリックし [サービスの開始] オプションで [ブート時] を選択します。
8 [ターゲット] タブで [追加] をクリックして、パーティションパス (SLES インストール時に作成) を入力します。
9 [Modify iSCSI Target Initiator Setup (iSCSI ターゲットイニシエータセットアップの変更)] ページで、ターゲットサーバの iSCSI クライアントイニシエータホスト名を指定し、[次へ] をクリックします。

   たとえば、「*iqn.sles12sp3node3.com* および *iqn.sles12sp2node3.com*」です。
10 [完了] をクリックします。
11 端末で `cat /proc/net/iet/volume` コマンドを実行し、iSCSI ターゲットがインストールされているかどうかを検証します。

すべてのノード上の iSCSI Initiator の設定

クラスタのすべてのノードで、iSCSI ターゲットに接続するように、iSCSI Initiator を設定する必要があります。

iSCSI Initiator を設定するには :

1 iSCSI Initiator パッケージをインストールします。
2 端末で `yast2 iscsi-client` を実行します。
3 [サービス] タブをクリックし、[サービスの開始] オプションで [ブート時] を選択します。
4 [接続したターゲット] タブをクリックし、[追加] をクリックして iSCSI ターゲットサーバの IP アドレスを入力します。
5 [認証なし] を選択します。
6 [次へ] をクリックし、[接続] をクリックします。
7 [起動の切り替え] をクリックして起動オプションを手動から自動に切り替えて、[次へ] をクリックします。
8 [次] をクリックし、[OK] をクリックします。
9 ターゲットに接続しているイニシエータのステータスをチェックするには、ターゲットサーバ上で `cat /proc/net/iet/session` コマンドを実行します。iSCSI サーバに接続しているイニシエータのリストが表示されます。

共有ストレージのパーティション化

SBD 用とクラスタファイルシステム用の 2 つの共有ストレージパーティションを作成します。

共有ストレージをパーティション化するには :

1 端末で `yast2 disk` コマンドを実行します。

2 [エキスパートパーティショナ] ダイアログボックスで共有ボリュームを選択します。この例では、[エキスパートパーティショナ] ダイアログボックスで [sdb] を選択します。

3 [追加] をクリックし、[プライマリパーティション] オプションを選択して [次へ] をクリックします。

4 [カスタムサイズ] を選択し、[次へ] をクリックします。この例では、カスタムサイズは 100MB です。

5 [フォーマットのオプション] で [パーティションをフォーマットしない] を選択します。この例では、［ファイルシステム ID］は［0x83 Linux］です。

6 [マウントのオプション] で [パーティションをマウントしない] を選択し、[完了] をクリックします。

7 [追加] をクリックし、[プライマリパーティション] を選択します。

8 [次へ] をクリックし、[最大サイズ] を選択して [次へ] をクリックします。

9 [フォーマットのオプション] で [パーティションをフォーマットしない] を選択します。この例では、［ファイルシステム ID］は［0x83 Linux］です。

10 [マウントのオプション] で [パーティションをマウントしない] を選択し、[完了] をクリックします。

HA Extension のインストール

HA Extension をインストールするには :

1 SUSE Downloads の Web サイトに移動します。

使用可能な各プラットフォームでは、SUSE Linux Enterprise High Availability Extension (SLE HA) を 2 つの ISO イメージとしてダウンロードできます。メディア 1 にはバイナリパッケージが、メディア 2 にはソースコードが含まれます。

注 : システムアーキテクチャに基づいて適切な HA Extension ISO ファイルを選択し、インストールします。

---

2 各サーバにメディア 1 ISO ファイルをダウンロードします。

3 [YaST コントロールセンター] ダイアログボックスを開いて、[アドオン製品］ >［追加] の順にクリックします。

4 [参照] をクリックし、DVD またはローカル ISO イメージを選択して、[次へ] をクリックします。

5 [パターン] タブで、[Primary Functions (プライマリ機能)] の下にある [高可用性] を選択します。

［高可用性］の下にあるすべてのコンポーネントがインストールされていることを確認します。

6 [承諾] をクリックします。

softdog ウォッチドッグのセットアップ

SLES HA Extension では、カーネルのウォッチドッグサポートはデフォルトで有効です。ウォッチドッグサポートは、ハードウェア固有のウォッチドッグドライバを提供するさまざまなカーネルモジュールに付属しています。ハードウェアに適したウォッチドッグドライバが、システム再起動時に自動的にロードされます。

1 softdog ウォッチドッグを有効にします。

```
echo softdog > /etc/modules-load.d/watchdog.conf
systemctl restart systemd-modules-load
```

2 softdog モジュールが正しくロードされているかどうかをテストします。

```
lsmod | grep dog
```

HA クラスタの設定

この例では、クラスタで 2 つのノードを設定していることを前提としています。

最初のノードの設定 :

1 クラスタノードとして使用したい物理または仮想マシンに root としてログインします。

2 次のコマンドを実行します。

```
ha-cluster-init
```

このコマンドは、NTP 設定とハードウェアウォッチドッグサービスを確認します。また、SSH アクセスと Csync2 の同期に使用される公開および秘密 SSH キーが生成され、それぞれのサービスが起動されます。

3 クラスタ通信層を設定します。

3a バインドするネットワークアドレスを入力します。

3b マルチキャストアドレスを入力します。スクリプトはデフォルトとして使用できるランダムなアドレスを提案します。

3c マルチキャストポートを入力します。デフォルトでは、ポートは 5405 です。

4 SBD をノードフェンシングメカニズムとして設定します。

4a SBD を使用するには、「*y*」を押します。

4b SBD に使用するブロックデバイスのパーティションの永続的なパスを入力します。パスはクラスタ内の両方のノードで一致している必要があります。

5 クラスタを管理するための仮想 IP アドレスを設定します。

5a 仮想 IP アドレスを設定するには、「*y*」を押します。

5b SUSE Hawk GUI の管理 IP として使用する、未使用の IP アドレスを入力します。たとえば、*192.168.1.3* と入力します。

個々のクラスタノードにログインする代わりに、その仮想 IP アドレスに接続できるようになります。

最初のノードが稼働したら、`ha-cluster-join` コマンドを使用して 2 番目のクラスタノードを追加します。

2 番目のノードの設定 :

1 クラスタに接続する物理マシンまたは仮想マシンに root としてログインします。

2 次のコマンドを実行します。

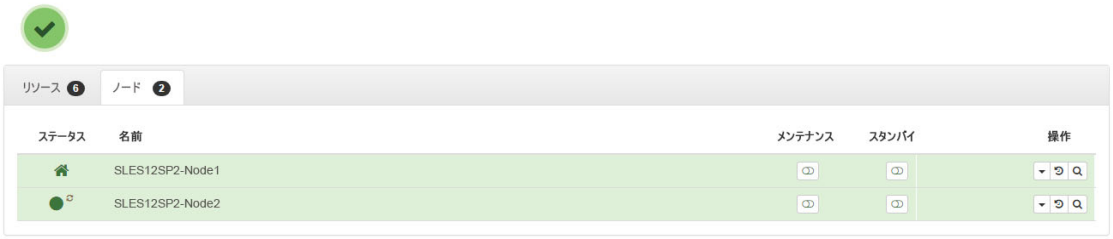
```
ha-cluster-join
```

NTP が設定されていない場合は、メッセージが表示されます。コマンドはハードウェアウォッチドッグデバイスを確認し、存在していない場合はそのことを知らせます。

3 最初のノードの IP アドレスを入力します。

4 最初のノードの root パスワードを入力します。

5 SUSE Hawk GUI にログインして、**[ステータス]**> **[ノード]** の順にクリックします。たとえば、`https://192.168.1.3:7630/cib/live` です。



クラスタノードでのアイデンティティボールトおよび Identity Manager エンジンのインストールと設定

1 クラスタノードに Identity Manager エンジンをインストールします。

1a NetIQ ダウンロード Web サイトから `Identity_Manager_4.9.0_Linux.iso` をダウンロードします。

1b ダウンロード済みの `.iso` をマウントします。

1c ISO がマウントされた場所から、次のコマンドを実行します。

```
./install.sh
```

1d 使用許諾契約を読みます。

1e 使用許諾契約の条項を確認し、「`y`」と入力します。

1f インストールする Identity Manager サーバのエディションを決定します。Advanced Edition の場合は「y」、Standard Edition の場合は「n」と入力します。

1g リストから **[Identity Manager エンジン]** を選択し、インストールを続行します。

この手順では、サポートされているアイデンティティボールトのバージョンをインストールします。

2 すべてのノードで Identity Manager エンジンを設定します。

2a `Identity_Manager_4.9.0_Linux.iso` をマウントした場所に移動します。

2b ISO がマウントされた場所から、次のコマンドを実行します。

```
./configure.sh
```

2c 標準設定またはカスタム設定のどちらを実行するかを決定します。設定オプションは、設定用に選択したコンポーネントに応じて変わります。

2d リストから **[Identity Manager エンジン]** コンポーネントを選択します。

2e 初めてアイデンティティボールトを設定する場合は、**[Create a new Identity Vault (アイデンティティボールトの新規作成)]** オプションを選択します。アイデンティティボールトを以前にインストールしていて、そのアイデンティティボールトインスタンスに接続する場合は、**[Add to an Identity Vault existing on local machine (ローカルマシンに存在するアイデンティティボールトに追加)]** または **[Add to an Identity Vault existing on local machine (ローカルマシンに存在するアイデンティティボールトに追加)]** オプションを選択します。

3 `/etc/opt/novell/eDirectory/conf` ディレクトリに移動します。

4 `nds.conf` ファイルを編集し、`n4u.nds.preferred-server` フィールドにクラスタの仮想 IP アドレスを指定します。

5 アイデンティティボールトサービスを停止します。

```
ndsmanage stopall
```

6 `/var/opt/novell/nici`、`/etc/opt/novell/eDirectory/conf`、および `/var/opt/novell/eDirectory/` ディレクトリから、すべてのフォルダとファイルをバックアップします。

7 `/opt/novell/eDirectory/bin` ディレクトリに移動します。

8 次のコマンドを実行します。

```
nds-cluster-config -s /<shared cluster path>
```

ここで、`<shared cluster path>` は、アイデンティティボールト共有クラスタデータに使用する場所を示します。

9 アイデンティティボールトサービスを開始します。

```
ndsmanage startall
```

クラスタ化されたセットアップでのアイデンティティボールドの設定の詳細については、『eDirectory Installation Guide』の「Deploying eDirectory on High Availability Clusters」を参照してください。

アイデンティティボールトのリソースの設定

1 SUSE Hawk GUI にログインします。

2 [リソースの追加] をクリックし、新しいグループを作成します。

2a [グループ] の横の⊕をクリックします。

2b グループ ID を指定します。たとえば、*Group-1* です。

グループの作成時に次の子リソースが選択されていることを確認します。

- *stonith-sbd*
- *admin_addr (クラスタ IP アドレス)*

3 [Meta Attributes (メタ属性)] タブで、[target-role] フィールドを［開始日］に、[is-managed] フィールドを［はい］に設定します。

4 [作成] をクリックします。

5 [Edit Configuration (設定の編集)] をクリックし、手順 2 で作成したグループの横にある✎をクリックします。

6 [子] フィールドで、次の子リソースを追加します。

- *shared-storage*
- *eDirectory-resource*

たとえば、グループ内では次の順序でリソースを追加する必要があります。

- *stonith-sbd*
- *admin_addr (クラスタ IP アドレス)*
- *shared-storage*
- *eDirectory-resource*

必要に応じてリソース名を変更できます。すべてのリソースに、定義する必要があるパラメータのセットがあります。shared-storage および eDirectory リソースの例については、eDirectory および共有ストレージの子リソースのプリミティブを参照してください。

eDirectory および共有ストレージの子リソースのプリミティブ

「stonith-sbd」リソースと admin_addr リソースは、クラスタノードの初期化時に HA クラスタコマンドによってデフォルト設定されます。

表 31-1 shared-storage の例

リソース ID	共有ストレージリソースの名前
クラス	ocf
プロバイダ	ハートビート
タイプ	ファイルシステム
デバイス	/dev/sdc1
ディレクトリ	/shared
fstype	xfs
操作	◆ start (60, 0) ◆ stop (60, 0) ◆ monitor (40, 20)
is-managed	対応
resource-stickiness	100
target-role	開始日

表 31-2 eDirectory-resource の例

リソース ID	eDirectory リソースの名前
クラス	systemd
タイプ	ndsdtmpl-shared-conf-nds.conf@-shared-conf-env
操作	◆ start (100, 0) ◆ stop (100, 0) ◆ monitor (100, 60)
target-role	開始日
is-managed	対応
resource-stickiness	100
failure-timeout	125
migration-threshold	0

場所制約スコアの変更

場所制約スコアを 0 に変更します。

1 SUSE Hawk GUI にログインします。

2 **[Edit Configuration (設定の編集)]** をクリックします。

3 **[制約]** タブで、クラスタのノード 1 の横にある をクリックします。

4 **[Simple (シンプル)]** タブで、スコアを *0* に設定します。

5 **[適用]** をクリックします。

クラスタ内のすべてのノードに対してスコアを *0* に設定してください。

注 : SUSE Hawk GUI の **[ステータス]**> **[リソース]**> **[移行]** オプションを使用して、あるノードから別のノードにリソースを移行する場合は、場所制約スコアが *Infinity* または *-Infintity* に変わります。これにより、クラスタ内で 1 つのノードのみが優先され、eDirectory 操作の遅延を招きます。

---

32 Tomcat アプリケーションサーバ上へのアイデンティティアプリケーションクラスタ展開ソリューションのサンプル

この章では、サンプル展開で Tomcat 上のクラスタ環境に Identity Applications を設定する方法に関する手順を記載します。

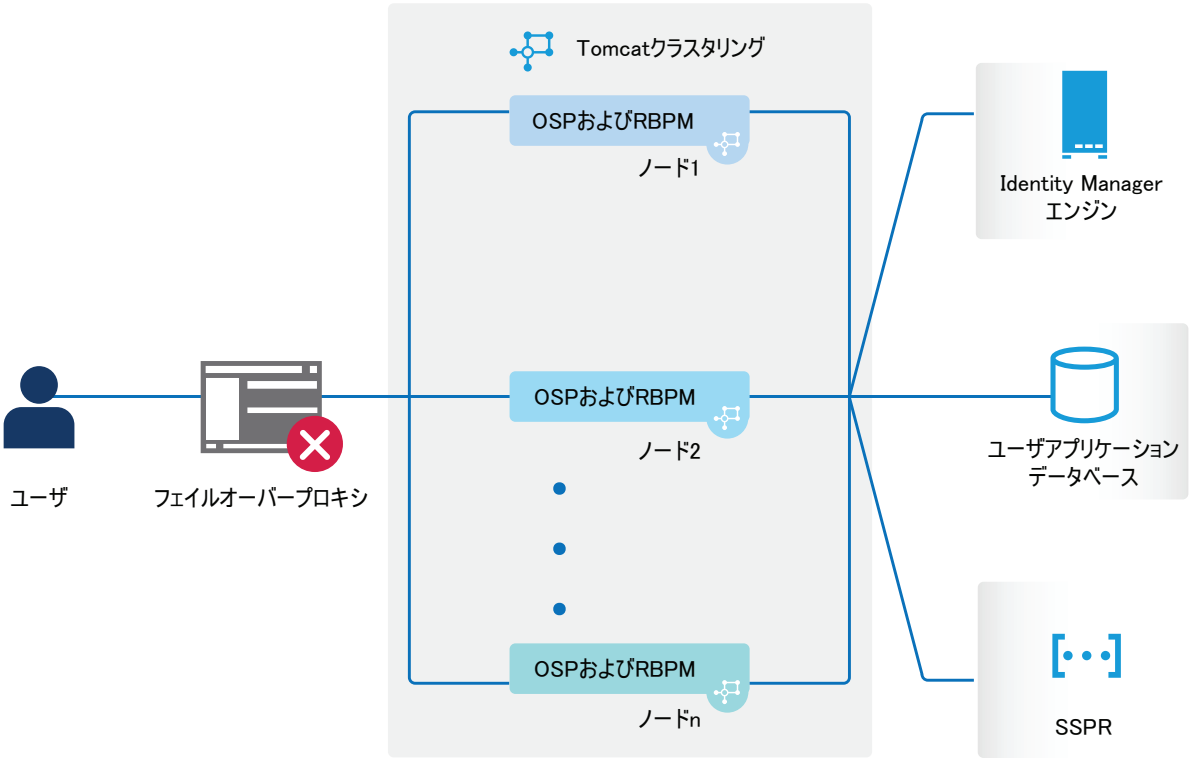
クラスタリングにより、いくつかのパラレルサーバ (クラスタノード) 上にアイデンティティアプリケーションを実行することができ、高可用性を実現できます。クラスタを構築するには、いくつかの Tomcat インスタンス (ノード) をグループ分けする必要があります。異なるサーバ間で負荷が分散され、サーバのいずれかで障害が発生した場合にも、識別情報アプリケーションに他のクラスタノードからアクセスできます。フェールオーバー用に、アイデンティティアプリケーションのクラスタを作成し、単一サーバとして機能するように設定できます。ただし、この設定には Identity Reporting は含まれません。

すべてのユーザ要求を処理し、それらをクラスタ内のサーバノードにディスパッチするロードバランサソフトウェアを使用することをお勧めします。ロードバランサは通常クラスタの一部で、クラスタ設定とともにフェールオーバーポリシーを理解しています。このため、ユーザは最適なソリューションを選択できます。

図 32-1 には、以下の前提条件を持つ 2 ノードクラスタのサンプル展開を示しています。

- すべての通信はロードバランサを介してルーティングされています。
- Identity Manager エンジンやユーザアプリケーションなどのコンポーネントは別のサーバにインストールされています。これは運用レベルの展開に推奨されるアプローチです。
- eDirectory、Identity Manager エンジン、アイデンティティアプリケーション、Tomcat アプリケーションサーバ、およびユーザアプリケーション用のデータベースのインストール手順に精通しています。
- SSPR (Single Sign-On Password Reset) が別のコンピュータにインストールされています。運用レベルの展開については、これが推奨されるアプローチです。
- PostgreSQL はユーザアプリケーション用のデータベースとして使用されています。ただし、Oracle または MsSQL などの、サポートされるデータベースのいずれかを使用できます。
- ユーザアプリケーションノードのすべてが eDirectory およびユーザデータベースの同じインスタンスと通信できます。要件に基づいて、ユーザアプリケーションインスタンスの数を増やすことができます。

図 32-1 クラスタ展開ソリューションのサンプル



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。

段階的な設定を理解するのを支援するために、このサンプル展開がドキュメントの後続のセクションで参照されます。

前提条件

Tomcat がサポートする環境に識別情報アプリケーション用のデータベースをインストールできますが、次の検討事項があります。

- クラスタには、固有のクラスタパーティション名、マルチキャストアドレス、およびマルチキャストポートが必要です。固有の識別子を使用して複数のクラスタを分離することによって、パフォーマンスの問題や異常な動作の発生を防ぎます。
  - クラスタの各メンバーで、識別情報アプリケーションのデータベースの待ち受けポートのポート番号として同一の値を指定する必要があります。
  - クラスタの各メンバーで、識別情報アプリケーションのデータベースをホストするサーバのホスト名または IP アドレスとして同一の値を指定する必要があります。
- クラスタ内のサーバ間でクロック時間が同期されます。そうしないと、セッションが早期にタイムアウトし、HTTP セッションのフェールオーバーが適切に機能しなくなる可能性があります。

- 同一ホストでは、複数のブラウザタブまたは複数のブラウザセッションで複数のログインを使用しないことをお勧めします。一部のブラウザはすべてのタブとプロセス間で cookie を共有します。そのため、複数のログインを行うと、(1 台のコンピュータを複数ユーザが共有することで認証機能に予期しない動作が発生するおそれがあるだけでなく) HTTP セッションのフェールオーバーの際に問題が発生する可能性があります。
- クラスタノードは同じサブネットに存在しています。
- フェールオーバープロキシまたは負荷分散ソリューションは、別のコンピュータにインストールされています。
- フォームのクラスタリングを実現するには、サーバ上のロードバランサの 2 つのインスタンス (一方は Identity Applications 用、他方はフォームレンダラ用) を起動します。

クラスタの準備

識別情報アプリケーションは HTTP のセッションレプリケーションとセッションフェールオーバーをサポートします。つまり、ノードでセッション処理中にノードにエラーが発生した場合、そのセッションは中断されることなく、同じクラスタ内の別のサーバ上で再開されます。クラスタに識別情報アプリケーションをインストールする前に、環境を準備する必要があります。

- 397 ページの 「Tomcat 環境のクラスタグループの理解」
- 397 ページの 「ワークフローエンジン ID のシステムプロパティの設定」

Tomcat 環境のクラスタグループの理解

ユーザアプリケーションクラスタグループは UUID 名を使用して、ユーザがサーバに追加する他のクラスタグループと競合するリスクを最小限に抑えます。ユーザアプリケーション管理機能を使用して、ユーザアプリケーションクラスタグループの環境設定を変更できます。クラスタ設定の変更がサーバノードで有効になるのは、そのノードを再起動した場合のみです。

ワークフローエンジン ID のシステムプロパティの設定

クラスタ内で識別情報アプリケーションをホストする各サーバでは、ワークフローエンジンを実行できますクラスタとワークフローエンジンのパフォーマンスを保証するために、クラスタ内のすべてのサーバが同じパーティション名とパーティション UDP グループを使用する必要があります。また、クラスタ内の各サーバを起動する際に一意のワークフロー

エンジン ID を指定する必要があります。なぜなら、ワークフローエンジンのクラスタリングは、識別情報アプリケーションのキャッシュフレームワークとは独立して動作するからです。

ワークフローエンジンが適切に動作することを保証するには、Tomcat のシステムプロパティを設定する必要があります。

1 クラスタ内の識別情報アプリケーションサーバごとに、新しい JVM システムプロパティを作成します。
2 このシステムプロパティに `com.novell.afw.wf.`*`engine-id`* という名前を付けます。ここで、engine-id は一意な値です。

インストール手順

このセクションでは、Tomcat に識別情報アプリケーションの新しいインスタンスをインストールし、それをクラスタリング用に設定するための段階的な手順を説明します。

注 : クラスタの場合、クラスタのすべてのメンバで環境設定を同一にする必要があるため、ism-sensitive.properties、ism-configuration.properties、および encrypt-keys.pkcs12 をプライマリノードからすべてのユーザアプリケーションノードにコピーします。

---

1. Identity Manager エンジンをインストールします。運用レベルの展開については、別のサーバ上に Identity Manager エンジンをインストールすることをお勧めします。
2. Identity Applications のデータベースをインストールします。Identity Applications とともにインストールされる PostgreSQL データベースを使用できます。ただし、別のサーバにデータベースをインストールすることをお勧めします。
3. Node1 には、Identity Applications をインストールして設定します。

   設定時には、次のことを確認してください。

   - 新しいデータベースオプションを選択する
   - 固有のワークフローエンジン ID を提供する。たとえば、`Node1` です。
   - クラスタ内のすべてのユーザアプリケーションノードで使用可能なデータベース jar ファイルがある。PostgreSQL の場合、`postgresql-42.6.1.jar` は `/opt/netiq/idm/postgres` にあります。

   Identity Applications は、マスタキーを使用して機密データを暗号化します。インストールプログラムは、Identity Applications の設定中に新しいマスタキーを作成します。クラスタ内で、ユーザアプリケーションのクラスタリングでは、ユーザアプリケーションのすべてのインスタンスが同じマスタキーを使用する必要があります。マスタキーは、`/opt/netiq/idm/apps/tomcat/conf/` ディレクトリにある `ism-configuration.properties` ファイルの `com.novell.idm.masterkey` プロパティの下に格納されます。
4. Node2 に、Identity Applications をインストールして設定します。

   設定時には、次のことを確認してください。

   - 既存のデータベースオプションを選択する。

- 固有のワークフローエンジン ID を提供する。たとえば、Node2 です。
- クラスタ内のすべてのユーザアプリケーションノードで使用可能なデータベース jar ファイルがある。PostgreSQL の場合、postgresql-9.4.1212.jar は /opt/netiq/idm/postgres にあります。

Node2 のユーザアプリケーション設定を完了した後で、Node1 の ism-configuration.properties からのマスタキー値をコピーし、Node 2 の ism-configuration.properties に格納されている対応するマスタキー値を置き換えます。マスタキーは ism-configuration.properties (/opt/netiq/idm/apps/tomcat/conf/) の com.novell.idm.masterkey プロパティの下に格納されます。

5. ロードバランササーバで、Identity Applications ポート番号を使用したロードバランサの一方のインスタンスと、すべてのクラスタノード用のフォームレンダラポート番号を使用したロードバランサの他方のインスタンスを起動します。次に例を示します。
   - ./balance 8543 apps1-au.edu.in:8543 ! apps2-au.edu.in:8543
   - .·/balance 8600 apps1-au.edu.in:8600 ! apps2-au.edu.in:8600
6. 別のコンピュータに SSPR をインストールします。

   SSPR インストールが完了したら、SSPR (https://<IP>:<port>/sspr/private/config/editor) を起動し、ログインします。**[Configuration Editor (環境設定エディタ)]** > **[設定]** > **[セキュリティ]** > **[Web Security (Web セキュリティ)]** > **[Redirect Whitelist (ホワイトリストをリダイレクト)]** をクリックします。

   a. **[Add value (値の追加)]** をクリックし、次の URL を指定します。

      https://<dns of the failover>:<port>/osp

   b. 変更内容を保存します。

   c. SSPR 設定ページで、**[設定]** > **[Single Sign On (SSO) Client (シングルサインオン (SSO) クライアント)]** > **[OAuth]** の順にクリックし、IP アドレスを、ロードバランソフトウェアがインストールされているサーバの DNS 名に置き換えることによって、**[OAuth ログイン URL]**、**[OAuth コード解決サービス URL]**、および **[OAuth プロファイルサービス URL]** リンクを変更します。

   d. **[設定]** > **[アプリケーション]** > **[アプリケーション]** をクリックし、ロードバランサソフトウェアがインストールされているサーバの DNS 名で IP アドレスを置き換えて、**[フォワード URL]**、および **[ログアウト URL]** を更新します。SSPR がインストールされているサーバ / システムの IP アドレスまたはホスト名を指定して、**[サイト URL]** を更新します。

   e. Node1 上の SSPR 情報を更新するには、/opt/netiq/idm/apps/configupdate/ にある設定ユーティリティを起動します。

      ./configupdate.sh

注 : configudate.sh ファイルは、configupdate ディレクトリからのみ実行してください。カスタムの場所から configupdate.sh を実行すると失敗します。

f. **[SSO クライアント]> [セルフサービスパスワードリセット]** の順にクリックし、**[クライアント ID]**、**[パスワード]**、および **[OSP Auth redirect URL (OSP 認証リダイレクト URL)]** パラメータの値を入力します。詳細については、138 ページの 「セルフサービスパスワードリセット」を参照してください。

注 : これらのパラメータの値が Node2 で更新されていることを確認します。

7. Node1 で、Tomcat を終了し、次のコマンドを使用して、ロードバランササーバの DNS 名を指定して、新しい osp.jks ファイルを生成します。

```
/opt/netiq/common/jre/bin/keytool -genkey -storetype PKCS12 -keyalg RSA
-keysize 2048 -keystore osp.jks -storepass <password> -keypass
<password> -alias osp -validity 1800 -dname "cn=<loadbalancer IP/DNS>"
```

例 : /opt/netiq/common/jre/bin/keytool -genkey -storetype PKCS12 -keyalg RSA -keysize 2048 -keystore osp.jks -storepass changeit -keypass changeit -alias osp -validity 1800 -dname "cn=mydnsname"

注 : キーパスワードが OSP インストール中に入力したパスワードと同じであることを確認します。または、これをキーストアパスワードを含む設定更新ユーティリティを使用して変更することもできます。

8. (状況に応じて実行) osp.jks ファイルが変更で更新されているかどうかを確認するには、次のコマンドを実行します。

```
/opt/netiq/common/jre/bin/keytool -list -v -keystore osp.jks -storepass
changeit
```

9. /opt/netiq/idm/apps/osp/ にある元の osp.jks ファイルのバックアップをとり、新しい osp.jks ファイルをこの場所にコピーします。
10. Node1 からクラスタの他のユーザアプリケーションノードに /opt/netiq/idm/apps/osp/ にある新しい osp.jks ファイルをコピーします。
11. 各クラスタノード上で、
    a. /opt/netiq/idm/apps/sites ディレクトリに移動して、ServiceRegistry.json ファイルを編集し、ロードバランサ詳細を追加します。

```
{"serviceRegisteries":[{"serviceID":"IDM","restUrl":"https://<DNS
of the load balancer>:8543/IDMProv"}]}
```

    b. /opt/netiq/idm/apps/sites/ ディレクトリに移動して、config.ini ファイルを編集し、ロードバランサ DNS とポート番号を追加します。

```
OSPIssuerUrl=https://<DNS of the load balancer>:8543/osp/a/idm/
auth/oauth2
OSPRedirectUrl=https://<DNS of the load balancer>:8600/forms/
oauth.html
ClientID=forms
OSPLogoutUrl=https://<DNS of the load balancer>:8543/osp/a/idm/
auth/app/logout
```

12. Node1 で設定ユーティリティを起動し、［ランディングページへの URL リンク］や［OAuth リダイレクト URL］などの URL 設定のすべてを［SSO クライアント］タブのロードバランサ DNS 名に変更します。

    a. 設定ユーティリティで変更を保存します。変更について ism-configuration properties ファイルを確認し、URL が Node 1 DNS およびポートを依然としてポイントしている場合は変更します。

    b. クラスタの他のすべてのノードでこの変更を反映させるには、Node1 からクラスタ内の他のユーザアプリケーションノードに、/TOMCAT_INSTALLED_HOME/conf にある ism-configuration properties ファイルをコピーします。

    注

    - Node1 から、クラスタ内の他のノードに ism-configuration.properties ファイルをコピーしました。ユーザアプリケーションインストール中にカスタムインストールパスを指定した場合は、参照パスがクラスタノードの設定更新ユーティリティを使用して修正されていることを確認します。
    - あるノードから別のノードに ism-configuration.properties ファイルをコピーした後、そのファイルに novlua:novlua 許可があることを確認してください。
    - このシナリオでは、OSP とユーザアプリケーションのどちらも同じサーバにインストールされます。したがって、同じ DNS 名が URL をリダイレクトするために使用されます。
    - OSP およびユーザアプリケーションが別のサーバにインストールされている場合は、OSP URL をロードバランサを参照する異なる DNS 名に変更します。OSP がインストールされるすべてのサーバに対してこれを実行します。これを行うと、すべての OSP 要求がロードバランサを介して OSP クラスタ DNS 名にディスパッチされます。これには、OSP ノードに別のクラスタがある必要があります。

13. novlua 許可を osp.jks ファイルに割り当てます。

    ```
    chown novlua:novlua osp.jks
    ```

14. /TOMCAT_INSTALLED_HOME/bin/ ディレクトリにある setenv.sh ファイルで次のアクションを実行します。

    a. mcast_addr バインディングが成功するためには、JGroup で preferIPv4Stack プロパティが **[true]** に設定されている必要があります。これを実行するには、すべてのノードの setenv.sh ファイルに JVM プロパティ「-Djava.net.preferIPv4Stack=true」を追加します。

    b. Node1 の setenv.sh ファイルに -Dcom.novell.afw.wf.Engine-id=*Engine1* を追加します。同様に、クラスタ内の各ノードに固有のエンジン名を追加します。たとえば、Node2 の場合、*Engine2* として既存の名前を追加できます。

15. ユーザアプリケーションでクラスタリングを有効にします。

    a. Node1 で Tomcat を起動します。

    他のサーバを起動しないでください。

b. ユーザアプリケーション管理者としてユーザアプリケーションにログインします。

c. [環境設定] > [キャッシングとクラスタ] オプションをクリックします。

ユーザアプリケーションに［キャッシュマネージャー］ページが表示されます。

d. [クラスタキャッシュ環境設定] をクリックし、[有効なクラスタ] プロパティに対して [True] を選択します。

e. [保存] をクリックします。

f. Tomcat を再起動します。

注 : ［Enable Local settings (ローカル設定を有効化)］を選択している場合は、クラスタ内の各サーバについてこの手順を繰り返します。

ユーザアプリケーションクラスタは、デフォルト UDP を使用してノード間のキャッシュ同期に JGroups を使用します。TCP を使用するようにこのプロトコルを変更する場合は、『NetIQ Identity Manager - Administrator's Guide to the Identity Applications』の「Configuring User Application Caching to use TCP (https://www.netiq.com/documentation/identity-manager-47/identity_apps_admin/data/b6iiw3v.html)」を参照してください。

---

16. クラスタリングのパーミッションインデックスを有効にします。

a. IDVault で Identity Console にログインし、[ツリービュー] に移動します。

b. ツリービューの左ペインで、[システム] から、ユーザアプリケーションドライバを含む [ドライバセット] を選択します。

c. 次に、[AppConfig] > [AppDefs] を選択します。

d. 右側のペインで、[環境設定] を選択します。

e. [Valued Attributes (値の属性)] で、XMLData 属性を選択し、`com.netiq.idm.cis.clustered` プロパティを [true] に設定します。

例 :

```
<property>
<key>com.netiq.idm.cis.clustered</key>
<value>true</value>
</property>
```

f. [保存] をクリックします。

17. Tomcat クラスタを有効にします。

Tomcat `server.xml` ファイルを `/TOMCAT_INSTALLED_HOME/conf/` から開き、すべてのクラスタノード上でこのファイルのこの行をコメント解除します。

```
<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
```

高度な Tomcat クラスタリング設定の場合、https://tomcat.apache.org/tomcat-8.5-doc/cluster-howto.html の手順を実行します。

18. すべてのノードで Tomcat を再起動します。

19. クラスタリング用のユーザアプリケーションドラバを設定します。

クラスタで、ユーザアプリケーションドライバは、クラスタのロードバランサの DNS 名を使用するように設定する必要があります。ユーザアプリケーションドライバは、Identity Console を使用して環境設定します。

a. Identity Manager エンジンを管理する Identity Console にログインします。

b. **[Identity Manager Administration (Identity Manager の管理)]** をクリックします。

c. ユーザアプリケーションドライバ、役割およびリソースサービスドライバ、データ収集サービスドライバを含むドライバセットを選択します。

d. **[ユーザアプリケーションドライバ]** アイコンをクリックします。

e. **[環境設定]** タブに移動します。

f. **[ドライバパラメータ]** ドロップダウンで、**[ホスト]** をロードバランサのホスト名または IP アドレスに変更します。

g. **[保存]** をクリックします。

h. ドライバを再起動します。

20. 役割とリソースのサービスドライバの URL を変更するには、19a から 19e までの手順を繰り返し、**[ドライバパラメータ]** ドロップダウンをクリックして、**[ユーザアプリケーションの URL]** をロードバランサ DNS 名で更新します。

21. セッションの粘着度がユーザアプリケーションノード用にロードバランサソフトウェアで作成したクラスタに対して有効になっていることを確認します。

22. Identity Manager ダッシュボード上でのクライアント環境設定詳細については、『NetIQ Identity Manager - Administrator's Guide to the Identity Applications』の「Configuring Client Settings Mode」を参照してください。

クラスタリングのパーミッションインデックスの有効化

1 Node1 で Identity Console にログインし、**[ツリービュー]** に移動します。

2 ツリービューの左ペインで、**[システム]** から、ユーザアプリケーションドライバを含む **[ドライバセット]** を選択します。

3 次に、**[AppConfig]** > **[AppDefs]** を選択します。

4 右側のペインで、**[環境設定]** を選択します。

5 **[Valued Attributes (値の属性)]** で、XMLData 属性を選択し、`com.netiq.idm.cis.clustered` プロパティを **[true]** に設定します。

例 :

```
<property>
<key>com.netiq.idm.cis.clustered</key>
<value>true</value>
</property>
```

6 **[保存]** をクリックします。

クラスタリング用のユーザアプリケーションドライバの環境設定

クラスタ化された環境で、ユーザアプリケーションの複数のインスタンスとともに単一のユーザアプリケーションドライバを使用できます。ドライバには、アプリケーション固有のさまざまな情報 (例 : ワークフロー環境設定情報、クラスタ情報) が保持されています。ドライバはクラスタのディスパッチャまたはロードバランサのホスト名または IP アドレスを使用するように設定する必要があります。

1 アイデンティティボールトを管理する Identity Console のインスタンスにログインします。

2 **[Identity Manager Administration (Identity Manager の管理)]** をクリックします。

3 ユーザアプリケーションドライバを含むドライバセットを選択します。

4 **[ユーザアプリケーションドライバ]** アイコンをクリックします。

5 **[環境設定]** タブに移動します。

6 **[ドライバパラメータ]** ドロップダウンで、**[ホスト]** をディスパッチャのホスト名または IP アドレスに変更します。

7 **[保存]** をクリックします。

8 ドライバを再起動します。

クラスタリング用の OSP と SSPR の設定

Identity Manager は、Tomcat クラスタ環境の SSPR 設定をサポートします。

クラスタリングをサポートするための SSPR の設定

クラスタの 1 番目のノードで SSPR 情報を更新するには、`/opt/netiq/idm/apps/configupdate/configupdate.sh` から設定ユーティリティを起動します。

表示されるウィンドウで、**[SSO クライアント]** > **[Self Service Password Reset]** をクリックし、**[クライアント ID]**、**[パスワード]**、および **[OSP Auth redirect URL (OSP 認証リダイレクト URL)]** パラメータの値を入力します。

クラスタノード上でのタスクの設定

クラスタノード上で次の設定タスクを実行します。

1 SSPR IP アドレスで［パスワードを忘れた場合］リンクを更新するには、1 番目のノードでユーザアプリケーションにログインし、**[管理]** > **[パスワードを忘れた場合]** をクリックします。

SSPR 設定の詳細については、108 ページの 「パスワードを忘れた場合の管理の設定」を参照してください。

2 [パスワードの変更] リンクを変更するには、112 ページの 「分散環境またはクラスタ化環境におけるダッシュボードの SSPR リンクの更新」を参照してください。

3 [パスワードを忘れた場合] および [パスワードの変更] リンクがクラスタの他のノードの SSPR IP アドレスで更新されていることを確認します。

注 : [パスワードの変更] および [パスワードを忘れた場合] リンクがすでに SSPR IP アドレスで更新されている場合、変更は必要ありません。

---

4 1 番目のノードで、Tomcat を終了し、次のコマンドを使用してロードバランササーバの DNS 名を指定して、新しい `osp.jks` ファイルを生成します。

```
/opt/netiq/common/jre/bin/keytool -genkey -keyalg RSA -keysize 2048 -keystore osp.jks -storepass <password> -keypass <password> -alias osp -validity 1800 -dname "cn=<loadbalancer IP/DNS>"
```

例 : `/opt/netiq/common/jre/bin/keytool -genkey -keyalg RSA -keysize 2048 -keystore osp.jks -storepass changeit -keypass changeit -alias osp -validity 1800 -dname "cn=mydnsname"`

注 : キーパスワードが OSP インストール中に入力したパスワードと同じであることを確認します。または、これをキーストアパスワードを含む設定更新ユーティリティを使用して変更することもできます。

---

5 (状況に応じて実行) `osp.jks` ファイルが変更で更新されているかどうかを確認するには、次のコマンドを実行します。

```
/opt/netiq/common/jre/bin/keytool -list -v -keystore osp.jks -storepass changeit
```

6 `/opt/netiq/idm/apps/osp_/` にある元の `osp.jks` ファイルのバックアップをとり、新しい `osp.jks` ファイルをこの場所にコピーします。新しい `osp.jks` ファイルがステップ 4 に作成されました。

7 1 番目のノードからクラスタ内のユーザアプリケーションノードのすべてに `/opt/netiq/idm/apps/osp` にある新しい `osp.jks` ファイルをコピーします。

8 1 番目のノードで設定ユーティリティを起動し、[ランディングページへの URL リンク] や [OAuth リダイレクト URL] などの URL 設定のすべてを [SSO クライアント] タブのロードバランサ DNS 名に変更します。

8a 設定ユーティリティで変更を保存します。

8b クラスタの他のすべてのノードにこの変更を反映させるには、1 番目のノードから、他のユーザアプリケーションノードのすべてに `/TOMCAT_INSTALLED_HOME/conf` にある `ism-configuration properties` ファイルをコピーします。

注 : 1 番目のノードから、クラスタ内の他のノードに `ism.properties` ファイルをコピーしています。ユーザアプリケーションのインストール中にカスタムインストールパスを指定した場合は、参照パスがクラスタノードで設定更新ユーティリティを使用して修正されていることを確認します。

このシナリオでは、OSP とユーザアプリケーションの両方が同じサーバにインストールされます。したがって、同じ DNS 名がリダイレクト URL に使用されます。

OSP およびユーザアプリケーションが別のサーバにインストールされている場合は、OSP URL をロードバランサを参照する異なる DNS 名に変更します。OSP がインストールされるすべてのサーバに対してこれを実行します。これにより、すべての OSP 要求が OSP クラスタ DNS 名にロードバランサを介してディスパッチされます。これには、OSP ノードに別のクラスタがある必要があります。

---

9 /TOMCAT_INSTALLED_HOME/bin/ ディレクトリの setenv.sh ファイルで次のアクションを実行します。

9a mcast_addr バインディングが成功するためには、JGroup で preferIPv4Stack プロパティが **[true]** に設定されている必要があります。これを実行するには、すべてのノードの setenv.sh ファイルに JVM プロパティ「-Djava.net.preferIPv4Stack=true」を追加します。

9b 1 番目のノードの setenv.sh ファイルに「-Dcom.novell.afw.wf.Engine-id=*Engine*」を追加します。

エンジン名は固有である必要があります。1 番目のノードのインストール中に指定された名前を提供します。名前が指定されてない場合、デフォルト名は「エンジン」です。

同様に、クラスタ内の他のノードに固有のエンジン名を追加します。たとえば、2 番目のノードの場合、エンジン名は Engine2 にできます。

33 Identity Manager のコンポーネントのアンインストール

このセクションでは、Identity Manager のコンポーネントをアンインストールするプロセスについて説明します。コンポーネントによっては、アンインストールするための前提条件があります。アンインストールプロセスを始める前に、必ずコンポーネントごとのセクション全体をレビューしてください。

注 : Identity Manager コンポーネントのアンイストールプロセスを開始する前に、必ず以下のアクションを実行してください。

- Tomcat、PostgreSQL、および ActiveMQ サービスを停止します。
- `/var/opt/netiq/idm/log/` ディレクトリからインストールログファイルのバックアップを取ります。

識別ボールトからのオブジェクトの削除

Identity Manager をアンインストールする最初のステップでは、すべての Identity Manager オブジェクトを識別ボールトから削除します。ドライバセットの作成時に、ウィザードにより、ドライバセットを 1 つのパーティションにするようメッセージが表示されます。いずれかのドライバセットオブジェクトが eDirectory のパーティションルートオブジェクトでもある場合、ドライバセットオブジェクトを削除するには、パーティションを親パーティションにマージする必要があります。

識別ボールトからオブジェクトを削除するには :

1. eDirectory データベースでヘルスチェックを実行し、発生したエラーを修正してから次に進みます。

   詳細については、『*NetIQ eDirectory 管理ガイド*』の「eDirectory の正常な動作の維持」を参照してください。
2. eDirectory ツリーに対するすべての権限を持つ管理者として Identity Console にログインします。
3. **[Partitions Management (パーティション管理)]** を選択します。
4. パーティションのルートオブジェクトであるドライバセットオブジェクトを参照して選択し、 をクリックします。
5. マージプロセスが完了するまで待ってから、**[OK]** をクリックします。
6. ドライバセットオブジェクトを削除します。

   ドライバセットオブジェクトを削除する際、そのドライバセットに関連付けられているすべてのドライバオブジェクトが削除されます。

7 eDirectory データベースにある各ドライバセットオブジェクトに対して、すべて削除されるまで、ステップ 3 ～ステップ 6 を繰り返します。

8 ステップ 1 を繰り返して、すべてのマージが完了し、オブジェクトがすべて削除されたことを確認します。

Identity Manager エンジンのアンインストール

インストーラでは、Identity Manager 用のアンインストールスクリプトが用意されています。このスクリプトを使用して、インストール時に作成されたすべてのサービス、パッケージ、およびディレクトリを削除できます。

注 : Identity Manager エンジンをアンインストールする前に、識別ボールトを準備します。詳細については、407 ページの 「識別ボールトからのオブジェクトの削除」を参照してください。

Identity Manager エンジンをアンインストールするには :

1 インストール用の iso をマウントしている場所に移動します。

2 `.iso` ファイルのルートディレクトリから、次のコマンドを実行します。

```
./uninstall.sh
```

3 **[Identity Manager エンジン]** の値を指定します。

4 Identity Manager エンジンとともに Identity Vault をアンインストールする場合は、**[Do you want to deconfigure and uninstall IDVault (IDVault を設定解除してアンインストールしますか)]** パラメータに **[y]** を指定し、以下の詳細を指定します。

- **Identity Vault Tree Name (アイデンティティボールトツリー名):** アイデンティティボールトツリー名を指定します。
- **Identity Vault Administrator name (アイデンティティボールト管理者名):** アイデンティティボールト管理者名を指定します。
- **アイデンティティボールト管理者パスワード :** アイデンティティボールト管理者パスワードを指定します。

Identity Applications のアンインストール

1 インストール用に `.iso` をマウントした場所に移動します。

2 `.iso` ファイルのルートディレクトリから、次のコマンドを実行します。

```
./uninstall.sh
```

3 **[Identity Applications]** の値を指定します。

Identity Reporting のコンポーネントのアンインストール

Identity Reporting のコンポーネントをアンインストールする場合、次の順序で実行する必要があります。

1. ドライバを削除します。詳細については、409 ページの 「レポーティングドライバの削除」を参照してください。
2. Identity Reporting を削除します。詳細については、410 ページの 「Identity Reporting のアンインストール」を参照してください。
3. Sentinel を削除します。詳細については、410 ページの 「Sentinel Log Management for IGA のアンインストール」を参照してください。

注 : ディスク容量を節約するために、Identity Reporting のインストールプログラムは Java 仮想マシン (JVM) をインストールしません。したがって、1 つまたは複数のコンポーネントをアンインストールするには、使用可能な JVM が存在し、その JVM が PATH で指定されていることを確認します。アンインストールの際にエラーが発生した場合、JVM の場所をローカルの PATH 環境変数に追加してからアンインストールプログラムを再実行します。

レポーティングドライバの削除

Designer または Identity Console を使用して、データ収集および Managed System Gateway Driver を削除できます。

1 ドライバを停止します。使用するコンポーネントに応じて、次のどちらかのアクションを実行します。

- **Designer:** ドライバごとに、ドライバ行を右クリックして、**[ライブ] > [ドライバの停止]** の順にクリックします。
- **Identity Console:** [Driver Set Overview (ドライバセットの概要)] ページで、各ドライバ画像の右上隅をクリックして、**[Stop Driver (ドライバの停止)]** をクリックします。

2 ドライバを削除します。使用するコンポーネントに応じて、次のどちらかのアクションを実行します。

- **Designer:** ドライバごとに、ドライバ行を右クリックして、**[削除]** をクリックします。
- **Identity Console:** [Driver Set Overview (ドライバセットの概要)] ページで、**[Drivers (ドライバ)] > [Delete drivers (ドライバの削除)]** の順にクリックし、削除するドライバをクリックします。

Identity Reporting のアンインストール

Identity Reporting を削除する前に、データ収集ドライバと Managed System Gateway Driver を削除済みであることを確認します。詳細については、409 ページの 「レポーティングドライバの削除」を参照してください。

1 インストール用に .iso をマウントした場所に移動します。

2 .iso ファイルのルートディレクトリから、次のコマンドを実行します。

```
./uninstall.sh
```

3 [Identity Reporting] の値を指定します。

Sentinel Log Management for IGA のアンインストール

1 Sentinel サーバにログインします。

2 アンインストールスクリプトを含むディレクトリに移動します。

```
/opt/novell/sentinel/setup/
```

3 次のコマンドを実行します :

```
./uninstall.sh
```

4 アンインストールを続行するかどうか再確認を求められたら、「y」を押します。

スクリプトはまずサービスを停止し、その後に削除を実行します。

Designer のアンインストール

1 Designer を閉じます。

2 Designer をアンインストールします

アンインストールスクリプトのあるディレクトリに移動します。このスクリプトは、デフォルトでは、*<installation_directory>*/designer/UninstallDesigner/Uninstall Designer for Identity Manager です。

スクリプトを実行するには、「./uninstall

Analyzer のアンインストール

重要 : Analyzer ツールは Identity Manager 4.9 の ISO に含まれません。Analyzer がすでにインストールされている場合は、必ずバージョン 4.8.5 に更新し、そのバージョンを Identity Manager 4.9 とともに使用してください。

1 Analyzer を閉じます。

2 オペレーティングシステムに応じた手順で、Analyzer をアンインストールします。

`Uninstall Analyzer for Identity Manager` スクリプトのある場所に移動します。デフォルトでは、`<installation_directory>/analyzer/UninstallAnalyzer` ディレクトリにあります。

スクリプトを実行するには、「`/Uninstall`」と入力します。

34 トラブルシューティング

このセクションでは、Identity Manager のインストールとアップグレード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。トラブルシューティングに関する他のヒントについては、特定のコンポーネントのガイドを参照してください。

ログファイルの場所

Identity Manager は、問題のデバッグに役立つログファイルを保持しています。ログファイルは、次の場所にあります。

- すべての Identity Manager コンポーネントのインストールログファイル : `/var/opt/netiq/idm/log/idminstall.log`
- すべての Identity Manager コンポーネントの環境設定ログファイル : `/var/opt/netiq/idm/log/idmconfigure.log`
- ユーザアプリケーション: `/opt/netiq/idm/apps/tomcat/logs/catalina.out` および `/opt/netiq/idm/apps/tomcat/logs/idapps.out`
- Identity Reporting: `/opt/netiq/idm/apps/tomcat/logs/localhost.<date>.log`
- DCS: `/opt/netiq/idm/apps/tomcat/logs/catalina.out`
- eDirectory: `/var/opt/novell/eDirectory/log/ndsd.log`
- Tomcat: `/opt/netiq/idm/apps/tomcat/logs/catalina.out`
- OSP: `/opt/netiq/idm/apps/tomcat/logs/osp-idm.log`
- SSPR: `/opt/netiq/idm/apps/tomcat/logs/catalina.out`
- Sentinel Log Management for IGA: `/var/opt/novell/sentinel/log/server0.0log`
- フォームレンダラ : `opt/netiq/common/nginix/logs`

ログインに関連する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、NetIQ 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
Designer を起動すると、次のエラーが表示され、Designer の readme は表示されません。 `Exception... "Update manifest is missing a required addons property."`	エラーを無視して Designer を起動します。機能はすべて正常に実行されています。

問題	推奨されるアクション
大規模環境に (200 万オブジェクトを超える) ユーザがログインできない	eDirectory マスタとレプリカサーバの両方で ルールセットを `Value` として指定して `mail(Internet Mail Address)` 属性のインデックスを追加します。
アイデンティティアプリケーションページからサインアウトする場合、SSPR に `5053 ERROR_APP_UNAVALIABLE` エラーが表示される。	このエラーは無視します。機能が損なわれることはないからです。
Identity Applications への最初のログイン時に本人確認の回答が表示されない	1. SSPR サーバに、FQDN を使用して作成された証明書があることを確認します。 2. ユーザアプリケーションサーバにログインして、ConfigUpdate (`/opt/netiq/idm/apps/configupdate/`) ユーティリティを起動します。 3. **[SSO クライアント] > [セルフサービスパスワードリセット]** の順に移動して、設定が正しいことを確認します。 SSPR を別のサーバにインストールした場合は、SSPR 証明書が `/opt/netiq/idm/apps/tomcat/conf` のユーザアプリケーションサーバにある `idm.jks` にインポートされていることを確認します。

問題	推奨されるアクション
SSPR URL にアクセスする際にブラウザに空のページが表示される	これは、SSPR が OSP で適切に設定されていない場合に発生します。SSPR ログに次の情報が表示されます。 `2018-01-24T22:24:02Z, ERROR, oauth.OAuthConsumerServlet, 5071 ERROR_OAUTH_ERROR (unexpected error communicating with oauth server: password.pwm.error.PwmUnrecoverableException: 5071 ERROR_OAUTH_ERROR (io error during oauth code resolver http request to oauth server: Certificate for <IP> doesn't match any of the subject alternative names: [IP]))`  1. OSP が実行されている Tomcat サーバに、FQDN を使用して作成された有効な証明書があることを確認します。ユーザアプリケーションサーバにログインし、ConfigUpdate ユーティリティを起動します。**[SSO クライアント] > [セルフサービスパスワードリセット ]** の順に移動して、設定が正しいことを確認します。 2. OSP ログイン方法を上書きして SSPR にログインします。( たとえば、`https://<sspr sserver ip>:<port>/sspr/private/Login?sso=false`) 3. ページの右上隅にある **[Configuration Editor (環境設定エディタ)]** に移動します。 4. **[Configure Password (パスワードの設定)]** を指定し、**[Sign In (サインイン)]** をクリックします。 5. **[LDAP] > [LDAP ディレクトリ] > [デフォルト] > [接続]** の順に移動します。 6. LDAP 証明書が正しくない場合は、**[クリア]** をクリックします。 7. 証明書を再インポートするには、**[Import From Server (サーバからインポート)]** をクリックします。 8. **[設定] > [Single Sign On (SSO)Client (シングルサインオン (SSO) クライアント)] > [OAuth]** の順に移動して、**[OAUTH Web サービスサーバ証明書]** の下にある証明書が正しいことを確認します。 9. 証明書が正しくない場合は、**[クリア]** をクリックします。 10. 証明書を再インポートするには、**[Import From Server (サーバからインポート)]** をクリックします。

問題	推奨されるアクション
ConfigUpdate ユーティリティを異なるディレクトリから起動する場合のエラー	ConfigUpdate ユーティリティがエラーを報告します。変更は保存されません。たとえば、/opt/netiq/idm/apps/configupdate/configupdate.sh コマンドを使用して configupdate ユーティリティを起動する場合、起動しません。 代わりに、/opt/netiq/idm/apps/configupdate/ ディレクトリに移動して、/configupdate.sh コマンドを実行します。

分散セットアップで展開された SSPR コンテナに管理者としてログインする際に尋ねられるセキュリティの質問をスキップできません。	ssprconfiguration.xml で、次の行の objectclass の近くに余分なスペースがある場合は削除してください。
	
	`<setting key="pwmAdmin.queryMatch" syntax="USER_PERMISSION" syntaxVersion="2">`
	
	`<label>Administrator Permission</label>`
	
	`<value>{"ldapProfileID":"default","ldapQuery":"(objectClass=*)","ldapBase":"cn=admin,ou=sa,o=system"}</value>`
	
	`<value>{"ldapProfileID":"default","ldapQuery":"(objectClass=*)","ldapBase":"cn=uaadmin,ou=sa,o=data"}</value>`
	
	`</setting>`

インストールとアンインストールに関連する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、NetIQ 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
Identity Manager 4.9 を Open Enterprise Server (OES) 24.2 にインストールすると、glibc-32bit*.rpm で依存関係の失敗のエラーが発生します。このエラーは、Identity Manager 4.9 にアップグレードする際にも表示されることがあります。	Identity Manager をインストールする前に、必ず次の依存ライブラリまたは RPM をインストールしてください。 ◆ glibc-32bit-*.x86_64.rpm ◆ libz1-32bit glibc-32bit-*.x86_64.rpm がすでにシステムで利用できる場合、このエラーは無視して構いません。

問題	推奨されるアクション
Identity Manager 4.9 を SLES 15 SP5 とともにオールインワン設定でインストールする際に、SSPR のページのロードに失敗し、SSPR 5203 CONFIG_FORMAT_ERROR が表示されます。	Tomcat サービスを再開します。
Identity Manager 4.9 が RHEL 8 オペレーティングシステムにインストールされているシステムを再起動すると、Identity Manager ダッシュボードにアクセスできなくなります。`catalina.log` ファイルに次の警告メッセージが含まれます。 `WARN [com.novell.soa.af.impl.core.WorkflowEngineImpl] (main) [WORKFLOW] Duplicate engine id detected.This engine may not have been shutdown cleanly or another engine is running with engine-id: ENGINE .Waiting 60000 ms for heartbeat to timeout.` この問題は、システムを再起動する前に Tomcat サービスが停止されていない場合に発生します。ワークフローエンジンが起動すると、すでに使用されているエンジン ID を持つ別のノードがクラスタ内で検出され、警告がログに記録されます。	システムを再起動する前に Tomcat サービスを停止することを常に推奨します。警告が表示される場合に解決するには、`igaworkflowdb` データベースの afenginestate テーブルのエンジンステータスを変更します。pgAdmin などのデータベース管理ツールにログインして、エンジンのステータスを手動でシャットダウン状態に更新します。
RHEL 8.3 に Identity Manager 4.9 をインストールする場合、PostgreSQL サービスが正常に開始されません。	次の手順を実行します。 1. RHEL 8 に Identity Manager 4.9 をインストールします。 2. RHEL OS バージョンを 8.3 にアップグレードします。

問題	推奨されるアクション

SLES 15 または SLES 15 SPx に Identity Manager をインストールすると (ここで、x は Identity Manager でサポートされるすべての SLES 15 バージョンを示します)、Tomcat および ActivemQ サービスが無効状態になります。	次の手順を実行します。
	1. Tomcat および ActiveMQ サービスが無効になっているサーバにログインします。
	2. `insserv-compat*` RPM をインストールします。* 記号は、RPM の最新バージョンを示します。
	注 : オペレーティングシステムベンダーからの継続的なサポートを確保するために、ご使用のオペレーティングシステムのサブスクリプションサービスから依存するパッケージを入手することをお勧めします。サブスクリプションサービスに加入していない場合は、http://rpmfind.net/linux などの Web サイトから最新のパッケージを見つけることができます。
	3. 次のコマンドを実行して、Tomcat および ActiveMQ サービスを有効にします。
	`systemctl enable netiq-tomcat.service`
	`systemctl enable netiq-activemq.service`

RHEL 8.1 に Identity Manager 4.9 をインストールする場合は、以下の場合に、次のエラーメッセージが表示されます。
◆ `RHEL-Prerequisite.sh` スクリプトを実行する :
`dnf-utils rpm required but not found`
`RHEL 8.1 iso does not contain dnf-utils packages, its replaced by yum-utils`
◆ Identity Manager エンジンコンポーネントをインストールする :
`Some of the dependencies required for installation of Identity Vault are not found`
	エラーメッセージを無視して、インストールプロセスを続行します。

問題	推奨されるアクション
Identity Applications を 4.9 バージョンにアップグレードした後で、Identity Applications でワークフローフォームをレンダリングできません。一部のシナリオでは /opt/netiq/idm/apps/sites/config.ini に存在するクライアントパスワード (CientPass) が空のままです。	この問題を解決するには、次の手順を実行してください。 1. /opt/netiq/idm/apps/sites/ ディレクトリに移動します。 2. config.ini ファイルを編集して、ClientPass パラメータに base64 エンコードされたパスワードを入力します。 **注 :** 設定更新ユーティリティで **[forms client (フォームクライアント)]** に指定したのと同じパスワードを指定します。 3. NGINX サービスを再起動します。 `systemctl restart netiq-nginx.service` 詳細については、TID 7024492 を参照してください。
コンテナ環境で、Identity Manager ダッシュボードに対して設定されたアイドル時間およびセッションタイムアウトを超えた場合、**[Extend (延長)]** ボタンが予期しているように動作しません。	OSP コンテナの ism-configuration.properties ファイルに com.netiq.idm.session-timeout プロパティを追加します。
マルチサーバ環境で、ドライバをセカンダリサーバに展開しようとするときに、LDAP 例外が表示されます。	セカンダリサーバがインストールされ、パーティションがセカンダリサーバのアイデンティティボールトに追加された後で、両方のサーバ上の ndsd を再起動する必要があります。これは、コンテナの展開で Identity Manager エンジンをインストールしている場合にも当てはまります。
クラスタ化された環境で、ノードをネットワークから切断し (たとえば、ノード 2)、アクティブなノードに役割を作成した場合 (たとえば、ノード 1)、再度ネットワークに接続したときに、新しく追加された役割はノード 2 で同期されません。	ノード 2 で以下の手順を実行します。 1. .../temp/permindex files を削除します。 2. Tomcat を再起動します。
Identity Applications の設定時に、リモートサーバにインストールされた SSPR に接続する場合、SSPR 設定はスキップされます。ただし、Tomcat を再起動すると、インストールプロセス中にインストールされた sspr.war が展開されます。	リモートサーバにインストールされた SSPR に接続する場合は、Identity Applications を設定する前に /opt/netiq/idm/apps/tomcat/webapps ディレクトリから sspr.war を削除します。

問題	推奨されるアクション
Identity Applications の設定中に、Identity Applications 管理者用のカスタムサブコンテナ、たとえば `cn=uaadmin,ou=univ,o=data` を使用すると、`uaadmin` がデフォルトコンテナ (`ou=sa,o=data`) の下に作成されます。	Identity Applications 管理者用のカスタムサブコンテナを使用する場合は、以下のいずれかの手順を実行します。 ◆ Identity Applications を設定する前に、`ou=univ,o=data` カスタムサブコンテナを作成します。 ◆ 設定プロセス中に、**[Do you want to use custom container as root container (ルートコンテナとしてカスタムコンテナを使用しますか)]** プロンプトに対して **[No]** を指定します。インポートするカスタム LDIF ファイルには、カスタムルートコンテナとサブコンテナの詳細が含まれている必要があります。
サイレントインストールプロセスでは、サイレントプロパティファイルが作成されるときにシステム要件がチェックされません。サイレントインストール中、ログファイルには、システム要件が満たされていないことを示すエラーメッセージが表示されます。	この問題が発生した場合、`silent.properties` ファイルに `IS_SYSTEM_CHECK_DONE` パラメータを手動で追加してください。システム要件チェックをスキップするには、`IS_SYSTEM_CHECK_DONE` パラメータの値を 1 に設定します。
Identity Applications または Identity Reporting をアンインストールして再インストールすると、データベースユーザとスキーマのセットアップ時に設定プロセスが失敗します。この問題は、コンポーネントの再インストール時に標準設定を実行するときに見られます。	Identity Applications または Identity Reporting コンポーネントを再インストールする場合、カスタム設定を実行する必要があります。
アンインストールプロセスが未完了と報告されるが、ログファイルには何もエラーが表示されない。	インストールファイルがデフォルトで保存される `netiq` ディレクトリがプロセス中に削除されていません。このディレクトリは、コンピュータから NetIQ ソフトウェアをすべて削除している場合は削除できます。

アップグレードに関連する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、NetIQ 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
Open Enterprise Server (OES) 24.2 上の Identity Manager エンジンをアップグレードする際に、idmupgrade.log に次のエラーが報告されます。 [SEVERE]: Could not connect to the server Exception: Connect ErrorLDAPException: I/O Exception on host <IP_address>, port <port_number> Connect Error java.net.SocketException: Connection or outbound has closed	このエラーは無視します。機能が損なわれることはないからです。
Identity Manager は、デジタル証明書を使用してコンポーネントを許可し、コンポーネントと安全に通信します。アイデンティティボールト証明書は、idm.jks および tomcat.ks のキーストアファイルにインポートする必要があります。ただし、証明書をインポートした後に Identity Applications にアクセスしようとすると、次のエラーが発生する可能性があります。 javax.net.ssl.SSLHandshakeException: PKIX path validation failed: java.security.cert.CertPathValidatorException: Could not determine revocation status. 証明書は、［CRL Distribution Point (CDP) (CRL 配布ポイント (CDP))］フィールドで指定されている証明書取り消しリスト (CRL) をチェックして検証され、その証明書が取り消されているかどうかが判断されます。CRLDP は、ルート証明書と、キーストアファイル tomcat.ks および idm.jks に存在する中間証明書の両方で使用できます。ただし、証明書取り消しチェックは、デフォルトで無効になっています。その結果、PKIX 信頼マネージャは証明書の取り消しステータスを判断できません。	この問題を修正するには、-Dcom.sun.security.enableCRLDP プロパティを true に設定して、CRL 配布ポイントチェックを有効にします。 このプロパティを設定するには、次のアクションを実行してください。 1. Tomcat を停止します。 2. Tomcat の bin ディレクトリにある setenv.sh ファイルに移動します。たとえば、/opt/netiq/idm/apps/tomcat/bin/setenv.sh です。 3. CATALINA_OPTS に次のようにプロパティ -Dcom.sun.security.enableCRLDP=true を追加します。 export CATALINA_OPTS="-Dcom.sun.security.enableCRLDP=true" 4. Tomcat を起動します。

問題	推奨されるアクション
Identity Manager をアップグレードした後、管理者以外のユーザは Identity Manager ダッシュボードへのログインに非常に時間がかかります。アプリケーションページとダッシュボードページのロードには、かなりの遅延があります。 この問題は、デフォルトで有効になっているネストされたグループ検索が原因で発生します。アプリケーションは、環境内にネストされたグループがあるかどうかに関係なく、ネストされたグループメンバーシップを介してログインしたユーザによって継承された許可を検索します。	(状況によって実行) 次の手順は、Identity Manager 4.9 に適用されます。 1. Identity Applications が 4.9 バージョンにアップグレードされたサーバにログインします。 2. `/opt/netiq/idm/apps/tomcat/conf/` の場所に移動します。 3. テキストエディタで `ism-configuration.properties` ファイルを開きます。 4. ファイルの末尾に、次のプロパティを追加します。 `DirectoryService/realms/jndi/params/USE_NESTED_GROUPS=false` 5. ファイルを保存し、Tomcat を再起動します。
Identity Applications を以前のバージョンから 4.9 バージョンにアップグレードした後、フォームレンダラは期待どおりに動作しません。この問題は、デフォルトの IDMProv 展開コンテキストがカスタムコンテキストに変更された場合に発生します。	この問題を解決するには、次の手順を実行してください。 1. Identity Applications が 4.9 バージョンにアップグレードされたサーバにログインします。 2. `/opt/netiq/idm/apps/sites` ディレクトリに移動します。 3. `ServiceRegistry.json` ファイルを編集します。 4. 展開コンテキストを `IDMProv` から、アップグレード前に指定されたカスタムコンテキストに変更します。 5. `ServiceRegistry.json` ファイルを保存します。 6. `/opt/netiq/idm/apps/sites/forms/` ディレクトリに移動します。 7. `main.<version>.js` ファイルを編集します。ここで、*<version>* はランダムに生成された英数字の値です。 8. 展開コンテキストを `IDMProv` から、アップグレード前に指定されたカスタムコンテキストに変更します。 9. `main.<version>.js` ファイルを保存します。 10. Tomcat を再起動します。

問題	推奨されるアクション

分散環境の Identity Manager を 4.9 バージョンにアップグレードした後で、Identity Applications へのログインに失敗します。以下のようなエラーメッセージが表示されます。
 	
`Your login process did not complete successfully.`
 	
Identity Applications にログインするには、Identity Applications と OSP 間のセキュアな接続を確立するためのトラストアンカー証明書が必要です。トラストアンカー証明書には、サブジェクトタイプが CA に設定された基本制約拡張が含まれている必要があります。Identity Manager は、プロパティ `jdk.security.allowNonCaAnchor` を使用して、証明書のトラストアンカーを検証します。デフォルトでは、このプロパティは `false` に設定されています。したがって、トラストアンカーが証明書で見つからない場合は、Identity Applications と OSP 間の接続を確立することができず、ログインに失敗します。`idm-osp.log` ファイルで次の例外に気づくでしょう。
 	
`sun.security.validator.ValidatorException: TrustAnchor with subject "CN=***, L=***, O=***" is not a CA certificate`
	この問題を解決するには、次の条件のいずれかを満たす必要があります。
	
	◆ Identity ApplicationsとOSP間のセキュアな接続を確立するために使用される証明書は、適切な基本制約拡張を含む信頼できる CA 証明書であることを確認します。
	
	◆ クライアントによって信頼される自己署名証明書およびカスタム証明書の場合、基本制約拡張を含まない CA 以外の証明書を許可するように、プロパティ `jdk.security.allowNonCaAnchor` を変更できます。次のアクションを実行して、Java セキュリティ設定を変更します。
	
	1. `/opt/netiq/common/jre/lib/security/java.security` ディレクトリに移動します。
	2. プロパティ `jdk.security.allowNonCaAnchor=true` の値を設定します。
	3. ファイルを保存します。

Identity Applications 4.9 バージョンにアップグレードした後で、Identity Applications ダッシュボードで許可の要求中に、フォームを開くことができません。	この問題を解決するには、次のコマンドを使用して、NGNIX および Golang サービスを手動で再起動します。 ◆ **NGNIX:** `/opt/netiq/common/ngnix/ngnix` ◆ **Golang:** `/etc/init.d/netiq-golang.sh` Identity Applications は Identity Applications ダッシュボードでフォームをレンダリングするために NGNIX サービスを使用します。
Standard Edition の Identity Reporting をアップグレードすると、`configupdate.sh.properties` の **[is_prov]** パラメータは `true` に設定されます。Identity Applications は Standard Edition では使用できないため、このパラメータの値を `false` に設定する必要があります。	`configupdate.sh.properties` ファイルで **[is_prov]** パラメータを `false` に手動で設定します。

問題	推奨されるアクション
Identity Manager エンジンの以前のアップグレードに失敗した場合、Identity Manager エンジンインストーラを再実行できません。たとえば、最初の試行で Identity Manager エンジンの 4.9 アップグレードが失敗し、Identity Manager エンジンのアップグレードを再試行した場合、アップグレードプロセスはトリガされません。	次の手順を実行します。 1. novell-DXMLengnx RPM を使用して以前のバージョンに Identity Manager エンジンをダウングレードします。 2. Identity Manager エンジンをアップグレードします。
Identity Manager をアップグレードした後で、次のプロパティが ism-configuration.properties ファイルに追加されます。 com.netiq.idm.osp.ldap.admin-dn = cn=admin,ou=sa,o=system	ism-configuration.properties ファイルのプロパティをコメントアウトして、Tomcat を再起動します。機能が損なわれることはないからです。
Identity Manager をアップグレードすると、展開に SSPR がない場合でも、次の SSPR プロパティが ism-configuration.properties ファイルに追加されます。 com.netiq.sspr.redirect.url = https://___SSPR_IP___:___SSPR_TOMCAT_HTTPS_PORT___/sspr/public/oauth	ism-configuration.properties ファイルのプロパティをコメントアウトして、Tomcat を再起動します。機能が損なわれることはないからです。
Identity Manager のアップグレード後に、ism-configuration.properties ファイルは、java.protocol.handler.pkgs プロパティの重複値を入力します。	機能が失われることはありません。この問題を解決するには、次のアクションを実行してください。 1. /opt/netiq/idm/apps/tomcat/conf/ ディレクトリにある ism-configuration.properties ファイルに移動します。 2. ism-configuration.properties ファイルを変更し、java.protocol.handler.pkgs プロパティの重複値を削除します。 3. ファイルを保存し、Tomcat を再起動します。

問題	推奨されるアクション
Identity Manager アップグレードの後で、Tomcat を起動できません。Tomcat ログにはいくつかの例外があり、ワークフローエンジンとアイデンティティボールト間で通信障害が発生しています。	1. Identity Console にログインします。 2. **[Certificate Management (証明書管理)]** > **[Server Certificate Management (サーバ証明書管理)]** に移動します。 3. **[SSL CertificateDNS (SSL 証明書 DNS)]** チェックボックスをオンにして、**[エクスポート]** をクリックします。 4. **[証明書]** ドロップダウンリストで、［SSL CertificateDNS］を選択します。 5. **[Export private key (秘密鍵のエクスポート)]** チェックボックスをクリアします。**[Export format (エクスポート形式)]** が *DER* に設定されていることを確認します。 6. **[OK]** > **[Save the exported certificate (エクスポートされた証明書の保存)]** をクリックして、システムに証明書をダウンロードします。 7. Identity Applications サーバにログインします。 8. Tomcat を停止します。 9. `opt/netiq/common/jre/bin` ディレクトリに移動して、次のコマンドを使用して、証明書を `idm.jks` ファイルにインポートします。 `opt/netiq/common/jre/bin/keytool -import -trustcacerts -alias <certificate_alias_name> -keystore /opt/netiq/idm/apps/tomcat/conf/idm.jks -file <certificate_file_downloaded>` 10. Tomcat を再起動します。

Ansible の展開に関連する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、NetIQ 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
展開で初めて `deploy.yml` を実行すると、Docker イメージがターゲットノードに存在しないことを示す次のメッセージが表示されます。たとえば、Core DNS コンテナを展開する場合、次のエラーが表示されます。 `fatal: [<ip address/DNS>]: FAILED! => {"changed": true, "cmd": "docker images | grep coredns | grep 1.8.0", "delta": "0:00:00.914078", "end": "msg": "non-zero return code", "rc": 1, "start": "stderr": "", "stderr_lines": [], "stdout": "", "stdout_lines": []}`	現時点で解決策はありません。ただし、このメッセージを無視して、展開を続行できます。これによって機能が失われることはありません。
IP アドレスがネットワーク上の別のコンテナですでに使用されている場合、コンテナ展開は失敗します。次の例外がコンソールに報告されます。 `fatal: [<ip address/DNS>]: FAILED! => {"changed": false, "msg": "Error starting container b1eb07f42cf6bd63787ae6167f5e3a0f7cbee0f8be80a5764bcc7c7f9d6b96b1: 403 Client Error for http+docker://localhost/v1.40/containers/b1eb07f42cf6bd63787ae6167f5e3a0f7cbee0f8be80a5764bcc7c7f9d6b96b1/start: Forbidden (\"Address already in use\")"}`	コンテナに別の IP アドレスを割り当てます。

問題	推奨されるアクション

Identity Applications コンテナを展開した後で、Identity Manager ダッシュボードにログインし、**[タスク]** ページに移動すると、ダッシュボードにタスクのリストが期待どおりにはフェッチされません。次のエラーが `catalina.out` ファイルで報告されます。
 	
`SEVERE [main] org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource Unable to process web resource [/WEB-INF/classes/com/microfocus/idm/nrf/resources/NRFRsrc_fr.class] for annotations`
`java.io.EOFException`
`at java.io.DataInputStream.readFully(DataInputStream.java:197)`
`at java.io.DataInputStream.readUTF(DataInputStream.java:609)`
`at java.io.DataInputStream.readUTF(DataInputStream.java:564)`
`at org.apache.tomcat.util.bcel.classfile.ConstantUtf8.getInstance(ConstantUtf8.java:36)`
`at org.apache.tomcat.util.bcel.classfile.Constant.readConstant(Constant.java:79)`
`at org.apache.tomcat.util.bcel.classfile.ConstantPool.<init>(ConstantPool.java:53)`
`at org.apache.tomcat.util.bcel.classfile.ClassParser.readConstantPool(ClassParser.java:174)`
`at org.apache.tomcat.util.bcel.classfile.ClassParser.parse(ClassParser.java:83)`
`at org.apache.catalina.startup.ContextConfig.processAnnotationsStream(ContextConfig.java:2351)`
`at org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(ContextConfig.java:2250)`
`at org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(ContextConfig.java:2244)`
`at org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(ContextConfig.java:2244)`
`at org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(ContextConfig.java:2244)`
`at org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(ContextC`
	この問題を回避するには、次の手順を実行してください。
	
	1. `/opt/netiq/idm/apps/tomcat/webapps/` ディレクトリに移動します。
	2. `workflow` フォルダを削除します。
	3. (オプション) Tomcat を再起動します。

AKS の展開に関連する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、NetIQ 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

Azure portal で Terraform コマンドを実行する際に、インフラストラクチャの展開をブロックする可能性があるエラーが発生する場合があります。想定されるエラーのいくつかを以下に示します。
 	
`Error: making Read request on Azure KeyVault Secret slesvmpwd: azure.BearerAuthorizer#WithAuthorization: Failed to refresh the Token for request to https://idmkv88211894240361.vault.azure.net/secrets/slesvmpwd/?api-version=7.1: StatusCode=401 -- Original Error: adal: Refresh request failed.Status Code = '401'.Response body: {"error":{"code":"invalid_request","message":"Required audience parameter not specified"}} Endpoint http://localhost:50342/oauth2/token │ │ with module.server.data.azurerm_key_vault_secret.slesvmpwd, │ on sles15sp2-server/main.tf line 42, in data "azurerm_key_vault_secret" "slesvmpwd": │ 42: data "azurerm_key_vault_secret" "slesvmpwd" { kubernetes_secret.example3: Creation complete after 1s [id=idm/ingress-tls] ? │ Error: error detecting capabilities: error PostgreSQL version: dial tcp 52.186.162.130:5432: connect: connection timed out │ │ with module.dbserver.postgresql_database.igaworkflowdb, │ on Azure-PG-Server/main.tf line 217, in resource "postgresql_database" "igaworkflowdb": │ 217: resource "postgresql_database" "igaworkflowdb" { │ az aks get-credentials --resource-group rajj_rg --name cluster-name --overwrite-existing │ Error: making Read request on Azure KeyVault Secret uawfedbuser: azure.BearerAuthorizer#WithAuthorization: Failed to refresh the Token for request to https://idmkv80357639695488.vault.azure.net/secrets/uawfedbuser/?api-version=7.1: StatusCode=401 -- Original Error: adal: Refresh request failed.Status Code = '401'.Response body: {"error":{"code":"invalid_request","message":"Required audience parameter not specified"}} Endpoint http://localhost:50342/oauth2/token │ │ with module.dbserver.data.azurerm_key_vault`
	この問題を解決するには、次の手順を実行してください。
	
	1. Azure portal にログインします。
	2. (オプション) Azure Kubernetes クラスタ環境設定ファイルのパスを見つけます。
	3. 次のコマンドを実行して、デフォルトの環境設定ファイル `~/.kube/config` を上書きします。
	
	`aks get-credentials --resource-group <resource-group-name> --name <kube-cluster-name> --overwrite-existing`
	
	`export KUBE_CONFIG_PATH=~/.kube/config`
	
	4. Terraform コマンドを再実行します。詳細については、Configuration Generator を使用した展開のステップ 14、ステップ 15、ステップ 16 を参照してください。

問題	推奨されるアクション
Azure でパブリック IP アドレスを作成しようとすると、次のようなエラーが表示される場合があります。 `| Error: waiting for creation of Managed Kubernetes Cluster "cluster-name" (Resource Group "raj_rg"): Code="ReconcileStandardLoadBalancerError" Message="Reconcile standardload balancer failed.Details: outboundReconciler retry failed: Category: ClientError; SubCode: PublicIPCountLimitReached; Dependency: Microsoft.Network/PublicIPAddresses;OrginalError: Code=\"PublicIPCountLimitReached\" Message=\"Cannot create more than 10 public IP addresses for this subscription in this region.\" Details=[]; AKSTeam: Networking, Retriable: false." | | with module.aks.azurerm_kubernetes_cluster.main, | on .terraform/modules/aks/main.tf line 10, in resource "azurerm_kubernetes_cluster" "main": | 10: resource "azurerm_kubernetes_cluster" "main" {`	Azure チームに連絡して、サブスクリプションのクォータを増やすか、割り当て済みの未使用のパブリック IP アドレスを削除してみてください。

問題	推奨されるアクション

Azure ポータルで `terraform apply` コマンドを実行している場合は、次の例外が表示される可能性があります。
 	
`(AgentPoolK8sVersionNotSupported) Version <version> is not supported in this region.Please use [az aks get-versions] command to get the supported version list in this region.For more information, please check https://aka.ms/supported-version-list`
	この問題を解決するには、次の手順を実行してください。
	
	1. Azure portal にログインします。
	2.  をクリックします。
	3. 次のコマンドを実行して、現在の場所で利用可能なサポートされているバージョンを特定します。
	
	`az aks get-versions -l <location>`
	
	次に例を示します。
	
	`az aks get-versions -l eastus`
	
	4. `IDM_4.9_Cloud_Deployment_files` ディレクトリに移動して、`main.tf` ファイルに移動し、次の値を編集します。
	
	`kubernetes_version = "<version>"`
	
	`orchestrator_version = "<version>"`
	
	次に例を示します。
	
	`kubernetes_version = "1.20.15"`
	
	`orchestrator_version = "1.20.15"`
	
	注 : バージョン `1.20.x` は、AKS Kubernetes で認定およびサポートされています。
	
	5. 次のコマンドを再実行します。
	
	`terraform apply --auto-approve`

問題	推奨されるアクション
Azure Kubernetes Service に複数インスタンスのコンテナを展開する場合は、いくつかのポッドが保留中の状態で表示されることがあります。	この問題を解決するには、次の手順を実行します。 1. Azure portal にログインします。 2. 検索ボックスに「kubernetes サービス」と入力し、<Enter> を押します。 3. kubernetes クラスタを選択します。 4. **[問題の診断と解決]** > **[トラブルシューティング]** をクリックします。 5. (状況によって実行) トラブルシューティング方法を使用しても問題が解決しない場合は、Azure アカウントの使用クォータを確認します。 a. ホームページに移動し、検索ボックスに「クォータ」と入力して、<Enter> を押します。 b. **[Compute (計算)]** をクリックします。 c. (状況によって実行) 現在の使用量の列に赤い線とともに 100% と表示されている場合は、**[Request quota increase (クォータの増加を要求)]** をクリックします。 d. **[新しい制限]** テキストボックスに新しい制限を入力します。 e. **[送信]** をクリックします。
エンジンインスタンスのポッドを再起動すると、既存のアイデンティティボールトに対してドライバの展開が失敗します。Designer アプリケーションに次のエラーが表示される場合があります。 `The deployment of 'cn=Azure AD Driver,cn=Driver Set,o=system' failed with errors and/or warnings.`	Designer アプリケーションに移動し、アイデンティティボールトを Designer に再インポートします。
プライマリサーバを再起動すると、新しい IP アドレスが割り当てられますが、この新しい IP アドレスがセカンダリサーバの SSL 証明書の IP アドレスと一致しません。	セカンダリサーバを再起動すると、SSL 証明書が新しい IP アドレスで再発行されます。

問題	推奨されるアクション
Identity Manager Reporting で、合計サイズが 1MB を超えるレポートを複数インポートしようとすると、エラー 413 が表示されます。これが発生する理由は、nginx-ingress コントローラで許可される要求本文のデフォルトのサイズが 1MB であるためです。 `There was an error calling the URL (/IDMRPT-CORE/rpt/definitions/import?uploadid=<id>&content=text/html&format=application/json&overwrite=true).The server returned the status code 413 with the following data which was not parsable JSON data:`	この問題は、`Ingress Configuration` セクションの `values.yaml` ファイルに次の注釈を追加し、より大きい要求本文のサイズを許可することで解決できます。 `nginx.ingress.kubernetes.io/proxy-body-size: 3m`

Identity Manager エンジンのトラブルシューティング

次の表のリストは、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で構成されています。問題が続く場合は、NetIQ 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
OES への Identity Manager エンジンのインストール中に、次のメッセージがコンソールおよび `/var/opt/netiq/idm/log/` ディレクトリにある `idminstall.log` ファイルに報告されます。	`/etc/OES-brand` ファイルが OES サーバに存在していることを確認します。ファイルが存在していない場合は、新しいファイルを作成して、Identity Manager エンジンをもう一度インストールしてみてください。
eDirectory 初期化プロセスが進行中の場合、Identity Manager エンジンは起動できません。この問題は、ほとんどの場合、eDirectory DIB が非常に大きい場合に発生します。	この問題を回避するには、次の手順を実行します。 1. `/etc/opt/novell/eDirectory/conf/` ディレクトリに移動します。 2. `env_idm` ファイルで、**[SLEEP_BEFORE_ENGINE_STARTUP]** 環境変数を追加して、変数の値を 0 から 600 に設定します。値は秒単位で示されます。 **注 :** 無効な値、または 600 より大きい値を指定すると、値はデフォルトの 600 になります。 3. eDirectory を再起動します。 4. (状況によって実行) `ndsd.log` をチェックして、メッセージとログを確認します。

問題	推奨されるアクション
Linux システムで Identity Manager エンジンを実行すると、/tmp ディレクトリで、使用可能な容量があるにもかかわらず、ディスク容量が不足します。このステータスは、df(disk free) および du (disk used) コマンドを使用して確認できます。df コマンドでは使用可能な容量がないと表示されますが、du コマンドでは /tmp に割り当てられたすべての容量が使用されているわけではないと表示されます。この問題は、インスタンス化されるすべての Identity Manager ドライバがメモリ内の複数のライブラリをロードするために発生します。JVM は、これらのドライバを一時的に /tmp ディレクトリにコピーしてから削除します。削除されたファイルは、これらのファイルを作成した JVM プロセスが終了するまでメモリを使用し続けます。lsof コマンドを使用して、この動作を判定できます。この状態のファイルは削除済みとしてマークされます。消費される合計ディスク容量は、サーバで実行されているドライバの数によって異なります。	消費される容量は比較的静的です。したがって、/tmp ディレクトリに十分な空き容量があることを確認します。問題が解決しない場合は、eDirectory を再起動してください。
マルチサーバ環境では、認識されない拡張例外が表示されます。	プライマリサーバに、セカンダリサーバ用の読み取り / 書き込みパーティションを確保します。 1. iManager にログインします。 2. **[役割およびタスク]** > **[パーティションとレプリカ]** > **[レプリカビュー]** の順にクリックします。 3. セカンダリサーバを選択します。 4. 選択したサーバに読み取り / 書き込み許可を割り当てます。 **注 :** ドライバセットにセカンダリサーバが追加されていることを確認してください。
dxcmd を介して大きなクエリを実行する場合、dxcmd は 625 ERR_TRANSPORT_FAILURE エラーで終了し、結果ファイルは生成されません。 Identity Manager エンジンでは、dxcmd クエリを実行するためのデフォルト時間として環境変数 NCPCLIENT_REQ_TIMEOUT 値を使用します。デフォルトでは、NCP 接続で 115 秒でタイムアウトします。クエリを実行し、結果を返すために費やされた時間がこの値を超える場合は、エラーが表示されます。	環境変数 NCPCLIENT_REQ_TIMEOUT をクエリで費やす合計時間より大きい秒数に設定することで、タイムアウト値を増やします。dxcmd に対して環境変数を永続的に設定することは、export NCPCLIENT_REQ_TIMEOUT=value を dxcmd スクリプト /opt/novell/eDirectory/bin/dxcmd に追加することで実現できます。dxcmd を実行する前に、export NCPCLIENT_REQ_TIMEOUT=value を実行することで実施されるスクリプトからターミナルで変数を手動で設定することもできます。

問題	推奨されるアクション
Identity Manager エンジンを 4.9 にアップグレードすると、OES がサーバから切断されたことに関するエラーがログに記録される場合があります。	この問題は、パラメータ ${IDM_KEYSTORE_PWD} が定義されていないために発生します。 この問題は無視して構いませんが、回避したい場合は、サイレントプロパティまたは環境で ${IDM_KEYSTORE_PWD} パラメータを定義します。

Identity Applications と Identity Reporting 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、NetIQ 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
証明書サブジェクトで指定された LDAP サーバ名とアプリケーション設定が異なる場合、Identity Manager のアップグレード後に Identity Applications がアイデンティティボールトに接続できません。	Java では LDAPS 接続でエンドポイントを識別できるため、Identity Manager サーバへの接続時に指定したサーバ名と証明書で返されたサーバ名が同じであることが要求されます。サーバ名が異なる場合は、次の手順を実行します。 1. /opt/netiq/idm/apps/configupdate ディレクトリに移動します。 2. 次のコマンドを実行して、設定更新ユーティリティを起動します。 ./configupdate.sh 3. **[ユーザアプリケーション]** タブに移動して、**[Identity Vault server (アイデンティティボールトサーバ)]** をクリックし、サーバ名を LDAP サーバ証明書サブジェクトで指定された名前に変更します。 このアクションは、ism-configuration.properties ファイルの DirectoryService/realms/jndi/params/AUTHORITY property を更新します。 4. **[OK]** をクリックします。

問題	推奨されるアクション
Identity Applications と Identity Reporting が同じサーバにインストールされ、<reporting install folder>/bin ディレクトリにある設定更新ユーティリティを使用して設定変更を実行する場合は、Identity Manager ダッシュボードが起動に失敗します。次のエラーが catalina.out ログファイルで報告されます。 EboPortalBootServlet [RBPM] +++++WARNING!!!! : This portal application context, IDMProv, does not match the portal.context property set in the PortalService-conf/config.xml file.Only one portal per database is allowed.Data has been loaded using the previous portal context.To correct this you must revert back to the previous portal name of, NoCacheFilter, please consult the documentation.	設定変更については、/opt/netiq/idm/apps/configupdate/ ディレクトリにある設定更新ユーティリティを使用します。
設定更新ユーティリティを使用してプロパティのパスワードを変更することはできません。	次の手順を実行して、コマンドラインから、com.netiq.rpt.ssl-keystore のようなプロパティのパスワードを変更することができます。 1. パスワードを暗号化するには、次のユーティリティを使用します。 /opt/netiq/common/jre/bin/java -jar tomcat/lib/obscurity-0.13.2-uber.jar com.netiq.ism.obfuscate.server.util.PasswordObfuscator <<Password>> 2. /opt/netiq/idm/apps/tomcat/conf/ ディレクトリにある ism-configuration.properties ファイルに移動します。 3. ism-configuration.properties ファイルを変更して、com.netiq.rpt.ssl-keystore.pwd パラメータの手順 2 で指定された暗号化パスワードを追加します。 4. ファイルを保存し、Tomcat を再起動します。

問題	推奨されるアクション
Identity Reporting がスタンドアロンサーバ上にインストールされていて、ダッシュボードから Identity Reporting または IDM DCS URL を起動する場合、URL は起動に失敗します。	Identity Reporting または IDM DCS URL を起動した後で、次の手順を実行します。 1. アドレスバーに移動します。 2. URL を変更し、Identity Reporting がインストールされているサーバのホスト名とポートの詳細を手動で提供します。 それに加えて、Identity Applications がインストールされているサーバ上の `configupdate.sh.properties` ファイルに移動して、`sso_apps` パラメータの `rpt` と呼ばれるエントリを追加し、変更内容を保存します。例：**[sso_apps=ua,rpt]**
Identity Applications と Identity Reporting が同じサーバにインストールされ、OSP と Identity Applications の CEF 監査が有効になっている場合、Reporting コンポーネントは起動に失敗します。	この問題を回避するには、次の手順を実行します。 1. `/opt/netiq/idm/apps/tomcat/conf` ディレクトリにある `idmrptcore_logging.xml` ファイルに移動します。 2. `<keystore file>` パラメータを追加し、`idmrptcore_logging.xml` ファイルでキーストアファイルパスを指定します。たとえば、次の行を追加します。 `<keystore-file>/opt/netiq/idm/app/tomcat/conf/idm.jks</keystore-file>` 3. Tomcat を再起動します。
Identity Applications および Identity Reporting が同じサーバ上にインストールされている場合、データベース作成オプションとして **[Startup]** を選択すると、いくつかの例外がログに記載されます。	これらの例外をクリアするには、Tomcat を手動で再起動します。

問題	推奨されるアクション
既存の Identity Applications または Identity Reporting 環境設定がポートなしで設定されている場合、Identity Manager にアップグレードしようとすると、設定更新ユーティリティの **[認証]** および **[SSO クライアント]** タブに記載されている IP アドレスとポートに誤った値が表示されます。	Identity Applications と Identity Reporting をアップグレードしたら、以下の手順を実行します。 1. `/opt/netiq/idm/apps/configupdate` ディレクトリに移動します。 2. 次のコマンドを実行します。 `./configupdate.sh` 3. **[認証]** タブで、**[OAuth サーバのホスト識別子]** および **[OAuth サーバの TCP ポート]** フィールドにそれぞれ正しい IP アドレスとポートを指定します。 4. **[SSO クライアント]** タブで、IDM Administrator、Reporting、および IDM データ収集サービスの URL が正しい形式であることを確認します。 5. Tomcat を再起動します。
インストール時に作成された次のユーザアプリケーション環境設定を 1 つまたは複数変更する必要がある。 ◆ 識別ボールトの接続および証明書 ◆ 電子メール設定 ◆ Identity Managerエンジンのユーザ識別情報とユーザグループ ◆ Access Manager または iChain の設定	インストーラとは別に、環境設定ユーティリティを実行します。 **Linux**: インストールディレクトリ (デフォルトでは `/opt/netiq/idm/apps/configupdate/`) から次のコマンドを実行します。 `./configupdate.sh`
Tomcat を起動すると次の例外が発生する。 `port 8180 already in use`	すでに実行されている Tomcat (または他のサーバソフトウェア) のすべてのインスタンスをシャットダウンします。8180 以外のポートを使用するように Tomcat を再設定する場合は、ユーザアプリケーションドライバの `config` 設定を編集します。
Tomcat を起動すると、トラステッド証明書が見つからないと報告される。	ユーザアプリケーションのインストール時に指定した JDK を使用して Tomcat を起動していることを確認します。
ポータル管理ページにログインできない。	ユーザアプリケーション管理者アカウントが存在することを確認します。このアカウントは、Identity Console 管理者アカウントと同じではありません。

問題	推奨されるアクション
管理者アカウントを使用しても、新しいユーザを作成できない。	ユーザアプリケーション管理者は、最上位コンテナのトラスティである必要があり、スーパバイザ権が必要です。Identity Applications 管理者の権利を LDAP 管理者と同等の権利に設定することを試すことができます (Identity Console を使用)。
アプリケーションサーバを起動すると、キーストアエラーが発生する。	アプリケーションサーバが、ユーザアプリケーションのインストール時に指定した JDK を使用しない。 次のように `keytool` コマンドを使用して、証明書ファイルをインポートします。 `keytool -import -trustcacerts -alias aliasName -file certFile -keystore ..\lib\security\cacerts -storepass changeit` • *aliasName* は、この証明書に選択した一意の名前に置き換えます。 • *certFile* は、証明書ファイルのフルパスおよび名前に置き換えます。 • デフォルトのキーストアパスワードは、`changeit` です (別のパスワードがある場合は、それを指定します)。
電子メール通知が送信されない。	`configupdate` ユーティリティを実行して、ユーザアプリケーション環境設定パラメータの電子メールの送信者と電子メールホストに値を指定したかどうかを確認します。 **Linux**: インストールディレクトリ (デフォルトでは `/opt/netiq/idm/apps/configupdate/`) から次のコマンドを実行します。 `./configupdate.sh`
[IG SSO クライアント] タブは、設定更新ユーティリティに表示されません	`configupdate.sh.properties` ファイルで、`sso_apps` パラメータに `ig` のエントリを追加し、変更内容を保存します。`sso_apps` パラメータにすでに Identity Applications および Identity Reporting のエントリが含まれている場合、Identity Governance のエントリをリストに追加します。たとえば、**[sso_apps=ua,rpt,ig]** を追加します。

問題	推奨されるアクション
Identity Applications コンテナの展開中に、次の例外が表示されます。 `ERROR [com.novell.afw.portlet.consumer.core.EboPortletProducerChangeListener] (main) [RBPM] Portlet registration with portletID: 'HeaderPortlet' does not exist.com.novell.afw.portlet.exception.EboPortletRegistrationException: Portlet registration with portletID: 'HeaderPortlet' does not exist.`	Identity Applications コンテナを再起動します。

Identity Applications コンテナを展開した後に、新しいフォームに関連付けられている許可を要求しようとすると、フォームが期待どおりにロードされません。この問題はランダムに発生しています。	フォームレンダラサーバとポート詳細が `nginx.conf` ファイルで指定されていることを確認します。`nginx.conf` ファイルを更新するには、次の手順を実行します。
	
	1. フォームレンダラコンテナにログインします。
	
	`docker exec -it <container> <command>`
	
	次に例を示します。
	
	`docker exec -it fr-container bash`
	
	2. `/opt/netiq/common/nginx/` ディレクトリに移動します。
	3. `nginx.conf` ファイルを編集します。
	4. フォームレンダラサーバとポート詳細を指定します。例 :
	
	`server { listen 8600 ssl; server_name formrenderer.example.com;`

その他の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、NetIQ 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
Identity Manager エンジンで FIPS が有効になっているときに、Identity Console に接続しようとすると、FIPS が有効なアイデンティティボールト / eDirectory でパスワードの検証に失敗します。	Identity Console ディレクトリに移動し、次のファイルを実行します。 `./identityconsole_install` さらに、『*Identity Console Installation Guide*』の「Performing an Interactive Installation」の手順 5 から続行します。