Identity Console インストールガイド

2022 年 9 月

保証と著作権

保証と著作権、商標、免責事項、保証、輸出およびその他の使用制限、米国政府の規制による権利、特許ポリシー、および FIPS コンプライアンスの詳細については、https://www.netiq.com/company/legal を参照してください。

目次

本書およびライブラリについて **5**
NetIQ 社について **7**

1 Identity Console のインストール計画 **11**

Docker インストールのシステム要件と前提条件 11
システム要件 11
前提条件 11
環境の設定 13
スタンドアロンインストール (非 Docker) のシステム要件と前提条件 16
システム要件 16
(オプション)OSP 設定の前提条件 17
ワークステーションのシステム要件と前提条件 18
システム要件 18
RPM 署名の検証 19

2 Identity Console の展開 **21**

セキュリティの推奨事項 21
Docker コンテナとしての Identity Console の展開 22
OSP コンテナの展開 22
Docker コンテナとしての Identity Console の展開 24
Identity Console を Docker として使用するマルチツリー 26
スタンドアロン Identity Console の展開 26
スタンドアロン Identity Console の展開（非 Docker） 27
スタンドアロンの Identity Console を使用するマルチツリー 28
ワークステーションとしての Windows 上の Identity Console 29
Identity Console をワークステーションとして使用するマルチツリー 30
Identity Console の停止と再起動 30
Identity Console を Docker コンテナとして停止と再起動する 30
スタンドアロンの Identity Console の停止と再起動 31
Identity Console ワークステーションの終了と再起動 31
データ永続性の管理 31
Azure Kubernetes サービスでの Identity Console の展開 32
AKS クラスタでの Identity Console の展開 32
サーバ証明書の変更 38
Docker コンテナでのサーバ証明書の変更 38
スタンドアロンの Identity Console でのサーバ証明書の変更 39

3 Identity Console のアップグレード **41**

Docker コンテナとしての Identity Console のアップグレード 41
スタンドアロン Identity Console(非 Docker) のアップグレード 43
OSP コンテナのアップグレード 44

4 Identity Console のアンインストール **45**

Docker 環境のアンインストール手順 45
スタンドアロンの Identity Console(非 Docker) のアンインストール手順 45

本書およびライブラリについて

『*Identity Console インストールガイド*』には、NetIQ Identity Console (Identity Console) 製品をインストールおよび管理する方法に関する情報が含まれています。本書には、用語の定義および実装シナリオを記載しています。

本書の読者

このガイドはネットワーク管理者を対象としています。

ライブラリに含まれているその他の情報

ライブラリには次の情報リソースが含まれています。

インストールガイド

Identity Console をインストールおよびアップグレードする方法が記載されています。本書はネットワーク管理者を対象としています。

NetIQ 社について

当社はグローバルなエンタープライズソフトウェア企業であり、お客様の環境において絶えず挑戦となる変化、複雑さ、リスクという 3 つの要素に焦点を当て、それらをお客様が制御するためにどのようにサポートできるかを常に検討しています。

当社の観点

変化に適応すること、複雑さとリスクを管理することは普遍の課題

実際、直面するあらゆる課題の中で、これらは、物理環境、仮想環境、およびクラウドコンピューティング環境の安全な評価、監視、および管理を行うために必要な制御を脅かす最大の要因かもしれません。

重要なビジネスサービスの改善と高速化を可能にする

当社は、IT 組織に可能な限りの制御能力を付与することが、よりタイムリーでコスト効率の高いサービス提供を実現する唯一の方法だと信じています。組織が継続的な変化を遂げ、組織を管理するために必要なテクノロジが実質的に複雑さを増していくにつれ、変化と複雑さという圧力はこれからも増え続けていくことでしょう。

当社の理念

単なるソフトウェアではなく、インテリジェントなソリューションを販売する

確かな制御手段を提供するために、まずお客様の IT 組織が日々従事している現実のシナリオを把握することに努めます。そのようにしてのみ、実証済みで測定可能な結果を成功裏に生み出す、現実的でインテリジェントな IT ソリューションを開発することができます。これは単にソフトウェアを販売するよりもはるかにやりがいのあることです。

当社の情熱はお客様の成功を推し進めること

お客様が成功するためにわたしたちには何ができるかということが、わたしたちのビジネスの核心にあります。製品の着想から展開まで、当社は次のことを念頭に置いています。お客様は既存資産とシームレスに連動して動作する IT ソリューションを必要としており、展開後も継続的なサポートとトレーニングを必要とし、変化を遂げるときにも共に働きやすいパートナーを必要としています。究極的に、お客様の成功こそがわたしたちの成功なのです。

当社のソリューション

- ID およびアクセスのガバナンス
- アクセス管理

- セキュリティ管理
- システムおよびアプリケーション管理
- ワークロード管理
- サービス管理

セールスサポートへのお問い合わせ

製品、価格、および機能についてのご質問は、各地域のパートナーへお問い合わせください。パートナーに連絡できない場合は、弊社のセールスサポートチームへお問い合わせください。

各国共通 :	www.netiq.com/about_netiq/officelocations.asp
米国およびカナダ :	1-888-323-6768
電子メール :	info@netiq.com
Web サイト :	www.netiq.com

テクニカルサポートへのお問い合わせ

特定の製品に関する問題については、弊社のテクニカルサポートチームへお問い合わせください。

各国共通 :	www.netiq.com/support/contactinfo.asp
北米および南米 :	1-713-418-5555
ヨーロッパ、中東、アフリカ :	+353 (0) 91-782 677
電子メール :	support@netiq.com
Web サイト :	www.netiq.com/support

マニュアルサポートへのお問い合わせ

弊社の目標は、お客様のニーズを満たすマニュアルの提供です。改善のためのご提案は、www.netiq.com/documentation に掲載されている本マニュアルの HTML 版で、各ページの下にある［**[コメントを追加]**］をクリックしてください。Documentation-Feedback@netiq.com 宛てに電子メールを送信することもできます。貴重なご意見をぜひお寄せください。

オンラインユーザコミュニティへのお問い合わせ

NetIQ のオンラインコミュニティである Qmunity は、他のユーザや NetIQ のエキスパートとやり取りできるコラボレーションネットワークです。より迅速な情報、有益なリソースへの役立つリンク、NetIQ エキスパートとのやり取りを提供する Qmunity は、頼みにしている IT 投資が持つ可能性を余すことなく実現するために必要な知識の習得に役立ちます。詳細については、http://community.netiq.com を参照してください。

1 Identity Console のインストール計画

この章では、Identity Console をインストールするためのシステム要件と前提条件について説明します。Identity Console は Docker コンテナまたはスタンドアロンアプリケーションの両方として実行できます。そのため、両方のタイプのインストールのシステム要件と前提条件については、それぞれのセクションを参照してください。

注 : Identity Console は、eDirectory 9.2.4 HF2、Identity Manager Engine 4.8.3 HF2、およびそれぞれのそれ以降のバージョンをサポートしています。Identity Console を使用する前に、eDirectory および Identity Manager エンジンインスタンスをアップグレードする必要があります。

- 11 ページの 「Docker インストールのシステム要件と前提条件」
- 16 ページの 「スタンドアロンインストール (非 Docker) のシステム要件と前提条件」
- 18 ページの 「ワークステーションのシステム要件と前提条件」
- 19 ページの 「RPM 署名の検証」

Docker インストールのシステム要件と前提条件

このセクションでは、Identity Console を Docker コンテナとしてインストールするためのシステム要件と前提条件について説明します。

- 11 ページの 「システム要件」
- 11 ページの 「前提条件」
- 13 ページの 「環境の設定」

システム要件

Identity Console は Docker コンテナとして実行されるため、Identity Console をインストールするためのシステム要件とサポートされているプラットフォームの詳細については、Docker マニュアルを参照してください。

前提条件

- ❒ Docker 20.10.9-ce 以降をインストールします。Docker のインストール方法の詳細については「Docker Installation」を参照してください。
- ❒ Identity Consoleサーバとバックエンドサーバ間のデータ交換を暗号化/復号化するには、秘密鍵を使用して pkcs12 サーバ証明書を取得する必要があります。このサーバ証明書は、http 接続のセキュリティを確保するために使用されます。外部認証局が生成したサーバ証明書を使用できます。詳細については、「Creating Server Certificate Objects」を

参照してください。サーバ証明書には、Identity Console サーバの IP アドレスと DNS を含む件名の代替名が含まれている必要があります。サーバ証明書オブジェクトを作成したら、それを .pfx 形式でエクスポートする必要があります。

- ❒ 前の手順で取得したサーバ証明書の CA 署名を検証するには、.pem 形式のすべてのツリーで CA 証明書を取得する必要があります。また、このルート CA 証明書を使用すると、クライアントと Identity Console サーバの間でセキュリティ保護された LDAP 通信を確立できます。たとえば、/var/opt/novell/eDirectory/data/SSCert.pem から eDirectory CA 証明書 (SSCert.pem) を取得できます。
- ❒ (省略可能)One SSO Provider (OSP) を使用して、Identity Console ポータルに対してユーザのシングルサインオン認証を有効にできます。Identity Console をインストールする前に OSP をインストールする必要があります。Identity Console 用の OSP を構成するには、画面のプロンプトに従って、必要な環境設定パラメータの値を指定します。詳細については、22 ページの 「OSP コンテナの展開」を参照してください。Identity Console を既存の OSP サーバに登録するには、次の内容を、/opt/netiq/idm/apps/tomcat/conf/ フォルダの ism-configuration.properties ファイルに手動で追加する必要があります。

```
com.netiq.edirapi.clientID = identityconsole
com.netiq.edirapi.redirect.url = https://<Identity Console Server
IP>:<Identity Console Listener Port>/eDirAPI/v1/<eDirectory Tree Name>/
authcoderedirect
com.netiq.edirapi.logout.url = https://<Identity Console Server
IP>:<Identity Console Listener Port>/eDirAPI/v1/<eDirectory Tree Name>/
logoutredirect
com.netiq.edirapi.logout.return-param-name = logoutURL
com.netiq.edirapi.response-types = code,token
com.netiq.edirapi.clientPass._attr_obscurity = NONE
com.netiq.edirapi.clientPass = novell
```

注 : OSP では、OSP が複数の eDirectory ツリーをサポートしていないため、1 つの eDirectory ツリーにのみ接続できます。

- ❒ ホストマシンの適切なDNSエントリが/etc/hostsに完全修飾ホスト名で表示されていることを確認します。
- ❒ Identity Console を Edge ブラウザで使用する場合、すべての機能を利用するには Microsoft Edge の最新版をダウンロードする必要があります。

注 : Mozilla Firefox で Identity Console を使用している間に、オリジン不一致のエラーメッセージで操作が失敗する場合があります。トラブルシュートを行うには、次の手順を実行します。

1. Firefox を最新バージョンに更新します。
2. Firefox の URL フィールドに about:config と指定して、<Enter> を押します。
3. Origin を検索します。
4. ［network.http.SendOriginHeader］ をダブルクリックして、その値を 1 に変更します。

環境の設定

特定のパラメータを含む環境設定ファイルを作成する必要がある場合があります。OSP で Identity Console を構成する場合は、環境設定ファイルで OSP 固有のパラメータを指定する必要があります。たとえば、OSP パラメータを使用して以下の edirapi.conf ファイルを作成します。

注 : osp-redirect-url フィールドに eDirectory ツリー名を入力する必要があります。

```
listen = ":9000"
ldapserver = "2.168.1.1:636"
ldapuser = "cn=admin,ou=sa,o=system"
ldappassword = "novell"
pfxpassword = "novell"
ospmode = "true"
osp-token-endpoint = "https://10.10.10.10:8543/osp/a/idm/auth/oauth2/
getattributes"
osp-authorize-url = "https://10.10.10.10:8543/osp/a/idm/auth/oauth2/grant"
osp-logout-url = "https://10.10.10.10:8543/osp/a/idm/auth/app/logout"
osp-redirect-url = "https://10.10.10.10:9000/eDirAPI/v1/edirtree/
authcoderedirect"
osp-client-id = "identityconsole"
ospclientpass = "novell"
ospcert = "/etc/opt/novell/eDirAPI/cert/SSCert.pem"
bcert = "/etc/opt/novell/eDirAPI/cert/"
loglevel = "error"
check-origin = "true"
origin = "https://10.10.10.10:9000,https://192.168.1.1:8543"
```

OSP なしで Identity Console を構成する場合は、OSP パラメータを指定せずに、次に示すように環境設定ファイルを作成します。

```
listen = ":9000"
pfxpassword = "novell"
ospmode = "false"
bcert = "/etc/opt/novell/eDirAPI/cert/"
```

注 : Identity Console を複数の eDirectory ツリーで設定する場合は、「*ldapserver*」、「*ldapuser*」、および「*ldappassword*」パラメータをスキップして、環境設定ファイルを作成できます。

***表 1-1** 環境設定ファイル内の環境設定パラメータの説明*

環境設定パラメータ	説明
listen	Identity Console サーバのコンテナ内の待ち受けポートとして 9000 を指定します。

環境設定パラメータ	説明
ldapserver	eDirectory ホストサーバの IP およびポート番号を指定します。
ldapuser	eDirectory ユーザのユーザ名を指定します。このパラメータは、OSP ログインの場合にプロキシ認証コントロールを使用して eDirectory への ldap 呼び出しを開始するための資格情報として使用されます。Ldap ユーザは、eDirectory ツリーに対するスーパーバイザ権を持っている必要があります。
ldappassword	LDAP ユーザのパスワードを指定します。
pfxpassword	pkcs12 サーバ証明書ファイルのパスワードを指定します。
ospmode	OSP を Identity Console と統合するには true を指定します。false に設定すると Identity Console で LDAP ログインが使用されます。
osp-token-endpoint	この URL は、OSP サーバから特定の属性をフェッチして、認証トークンの有効性を検証するために使用します。
osp-authorize-url	この URL は、ユーザが認証トークンを取得するために資格情報を提供する際に使用します。
osp-logout-url	この URL を使用して、ユーザと OSP サーバ間のセッションを終了します。
osp-redirect-url	OSP サーバは、認証トークンを付与した後、ユーザをこの URL にリダイレクトします。 **注 :** Identity Console の構成時には、eDirectory ツリー名を小文字で指定してください。ツリー名が小文字で指定されていない場合、Identity Console サーバへのログインに失敗する場合があります。
osp-client-id	Identity Console を OSP に登録したときに提供された OSP クライアント ID を指定します。
ospclientpass	Identity Console を OSP に登録したときに提供された OSP クライアントパスワードを指定します。
ospcert	OSP サーバの認証局証明書の場所を指定します。
bcert	Identity Console の CA 証明書の場所を指定します。
loglevel	ログファイルに含めるログレベルを指定します。このパラメータは、「fatal」、「error」、「warn」、または「info」に設定できます。

環境設定パラメータ	説明
check-origin	これを true に設定すると、Identity Console サーバで要求の元の値が比較されます。使用可能なオプションは、true または false のいずれかです。DNS 設定を使用する場合、*check-origin* パラメータ値が false に設定されている場合でも、*origin* パラメータは必須です。
origin	Identity Console は、要求の起点値をこのフィールドで指定された値と比較します。 **注 :** Identity Console 1.4 以降では、このパラメータは *check-origin* パラメータとは独立しています。また、DNS 設定を使用する場合は必須です。
maxclients	IDConsole にアクセスできる同時クライアントの最大数。この制限を超えるクライアントはキュー内で待機する必要があります。

注

- ospmode 環境設定パラメータは、OSP を Identity Console と統合する計画の場合にのみ使用します。
- Identity Managerセットアップで Identity Applications (Identity Apps)がクラスタモードで設定されている場合は、環境設定ファイルの osp-token-endpoint、osp-authorize-url、および osp-logout-url フィールドにロードバランササーバの DNS 名を指定する必要があります。これらのフィールドに OSP サーバの詳細を入力した場合、Identity Console のログインは失敗します。
- Identity Console が、Identity Apps および Identity Reporting と同じ OSP インスタンスで構成されている場合、Identity Console ポータルにログインするときにシングルサインオン (認証サービス) が有効になります。
- OSP HTTPS URL は、Identity Console 1.4 以降の 2048 ビットキー以上を含む証明書で検証する必要があります。
- さまざまなドメインから Identity Console ポータルへのアクセスを制限する場合は samesitecookie パラメータを strict に設定します。さまざまなドメインから Identity Console ポータルへのアクセスを許可する場合は、samesitecookie パラメータを lax に設定します。設定時にパラメータが指定されていない場合、ブラウザの設定がデフォルトで適用されます。

環境設定ファイルの準備ができたら、コンテナの展開を続行します。詳細については、22 ページの 「Docker コンテナとしての Identity Console の展開」を参照してください。

スタンドアロンインストール (非 Docker) のシステム要件と前提条件

- 16 ページの 「システム要件」
- 17 ページの 「(オプション)OSP 設定の前提条件」

システム要件

このセクションでは、スタンドアロンの Identity Console をインストールするためのシステム要件と前提条件について説明します。

カテゴリ	最小要件
プロセッサ	1.4 GHz 64 ビット
メモリ	2GB
ディスク容量	Linux 上で 200 MB
サポートされているブラウザ	• **Microsoft Edge** の最新バージョン • **Google Chrome** の最新バージョン • **Mozilla Firefox** の最新バージョン **注 :** Mozilla Firefox で Identity Console を使用している間に、オリジン不一致のエラーメッセージで操作が失敗する場合があります。トラブルシュートを行うには、次の手順を実行します。 **1** Firefox を最新バージョンに更新します。 **2** Firefox の URL フィールドに about:config と指定して、<Enter> を押します。 **3** Origin を検索します。 **4** ［network.http.SendOriginHeader］をダブルクリックして、その値を 1 に変更します。

サポートされているオペレーティングシステム	• **認定済み** :
	  • SUSE Linux Enterprise Server (SLES) 15 SP1、SP2、および SP3
	  • SUSE Linux Enterprise Server (SLES) 12 SP1、SP2、SP3、SP4 および SP5
	  • Red Hat Enterprise Linux (RHEL) 7.8、7.9、8.0、8.1、8.2、8.3、8.4、および 8.5
	  • OpenSUSE 15.1 および 15.2
	• **サポート:** 上記の認定オペレーティングシステムのサポートパックの新しいバージョンでサポートされます。

カテゴリ	最小要件
証明書	• クライアントと Identity Console サーバ間のデータ交換を暗号化 / 復号化するには、秘密鍵を使用して pkcs12 サーバ証明書を取得する必要があります。このサーバ証明書は、http 接続のセキュリティを確保するために使用されます。外部認証局が生成したサーバ証明書を使用できます。詳細については、「Creating Server Certificate Objects」を参照してください。サーバ証明書には、Identity Console サーバの IP アドレスと DNS を含む件名の代替名が含まれている必要があります。サーバ証明書オブジェクトを作成したら、それを .pfx 形式でエクスポートする必要があります。 • 前の手順で取得したサーバ証明書の CA 署名を検証するには、.pem 形式のすべてのツリーで CA 証明書を取得する必要があります。また、このルート CA 証明書を使用すると、クライアントと Identity Console サーバの間でセキュリティ保護された LDAP 通信を確立できます。たとえば、/var/opt/novell/eDirectory/data/SSCert.pem から eDirectory CA 証明書 (SSCert.pem) を取得できます。

準備ができたら、Identity Console のインストールに進みます。詳細については、26 ページの 「スタンドアロン Identity Console の展開」を参照してください。

(オプション)OSP 設定の前提条件

One SSO Provider (OSP) を使用して、Identity Console ポータルに対してユーザのシングルサインオン認証を有効にできます。Identity Console をインストールする前に OSP をインストールする必要があります。Identity Console 用の OSP を構成するには、画面のプロンプトに従って、必要な環境設定パラメータの値を指定します。詳細については、22 ページの「OSP コンテナの展開」を参照してください。Identity Console を既存の OSP サーバに登録するには、次の内容を、/opt/netiq/idm/apps/tomcat/conf/ フォルダの ism-configuration.properties ファイルに手動で追加する必要があります。

```
com.netiq.edirapi.clientID = identityconsole
com.netiq.edirapi.redirect.url = https://<Identity Console Server
IP>:<Identity Console Listener Port>/eDirAPI/v1/<eDirectory Tree Name>/
authcoderedirect
com.netiq.edirapi.logout.url = https://<Identity Console Server
IP>:<Identity Console Listener Port>/eDirAPI/v1/<eDirectory Tree Name>/
logoutredirect
com.netiq.edirapi.logout.return-param-name = logoutURL
com.netiq.edirapi.response-types = code,token
com.netiq.edirapi.clientPass._attr_obscurity = NONE
com.netiq.edirapi.clientPass = novell
```

注

- OSP を初めてインストールする場合は、[eDir API で OSP を構成] にオプション [「y」] を指定し、画面上の指示に従って Identity Console を OSP に登録してください。
- Identity Console の構成時には、eDirectory ツリー名を小文字で指定してください。ツリー名が小文字で指定されていない場合、Identity Console サーバへのログインに失敗する場合があります。
- OSP では、OSP が複数の eDirectory ツリーをサポートしていないため、1 つの eDirectory ツリーにのみ接続できます。

ワークステーションのシステム要件と前提条件

- 18 ページの 「システム要件」

システム要件

このセクションでは、Identity Console を実行するためのシステム要件と前提条件について説明します。

カテゴリ	最小要件
プロセッサ	1.5 GHz 64 ビット
メモリ	2GB
ディスク容量	Windows 上で 1GB

サポートされているオペレーティングシステム	• **認定済み** :
	  • Windows Server 2016
	  • Windows server 2019
	  • Windows Server 2022
	  • Windows 10
	  • Windows 11

カテゴリ	最小要件
証明書	• Identity ConsoleクライアントとRESTサーバ間でデータを交換するには、pfx 形式でサーバ証明書を取得する必要があります。このサーバ証明書は、常に keys.pfx という名前にしてください。詳細については、「Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/b1j4tpo3.html#b1j4u0cm)」を参照してください。 • 前の手順で取得したサーバ証明書の CA 署名を検証するには、.pem 形式のすべてのツリーで CA 証明書を取得する必要があります。また、このルート CA 証明書を使用すると、クライアントと Identity Console サーバの間でセキュリティ保護された LDAP 通信を確立できます。 たとえば、/var/opt/novell/eDirectory/data/SSCert.pem から SSCert.pem の Linux 用の eDirectory CA 証明書を取得できます。 <eDirectory インストール場所>\NetIQ\eDirectory\DIBFiles\CertServ\SSCert.pem から Windows 用の eDirectory CA 証明書 SSCert.pem を取得します。

準備ができたら、Identity Console の展開に進みます。詳細については、29 ページの 「ワークステーションとしての Windows 上の Identity Console」 を参照してください。

RPM 署名の検証

RPM 署名の検証を実行するには、次の手順に従います。

1 ビルドを抽出するフォルダに移動します。

たとえば、<Identity Console の未展開の場所 >/IdentityConsole_150_Linux/license/MicroFocusGPGPackageSign.pub。

2 次のコマンドを実行して、公開鍵をインポートします。

```
rpm --import MicroFocusGPGPackageSign.pub
```

3 (オプション)RPM 署名 rpm --checksig -v <RPM 名 > を検証するには、次のコマンドを実行します

例 :

rpm --checksig -v identityconsole-1.5.0000.x86_64.rpm

identityconsole-1.5.0000.x86_64.rpm:

Header V4 RSA/SHA256 Signature, OK, key ID 786ec7c0: OK

Header SHA1 digest: OK

Header SHA256 digest: OK

Payload SHA256 digest: OK

V4 RSA/SHA256 Signature, key ID 786ec7c0: OK

MD5 digest: OK

2 Identity Console の展開

この章では、Identity Console をデプロイするプロセスと、セキュリティに関する推奨事項について説明します。デプロイメントの準備をするには、11 ページの第 1 章「Identity Console のインストール計画」に記載されている前提条件とシステム要件を確認してください。

- 21 ページの 「セキュリティの推奨事項」
- 22 ページの 「Docker コンテナとしての Identity Console の展開」
- 26 ページの 「スタンドアロン Identity Console の展開」
- 29 ページの 「ワークステーションとしての Windows 上の Identity Console」
- 30 ページの 「Identity Console の停止と再起動」
- 31 ページの 「データ永続性の管理」
- 32 ページの 「Azure Kubernetes サービスでの Identity Console の展開」
- 38 ページの 「サーバ証明書の変更」

セキュリティの推奨事項

- Docker コンテナには、デフォルトではリソースの制約がありません。そのため、すべてのコンテナには、ホストのカーネルによって提供されるすべての CPU リソースおよびメモリリソースへのアクセスが提供されます。コンテナで使用できるリソースの量に制限を設定することにより、実行中の 1 つのコンテナによってより多くのリソースが消費され、実行中の他のコンテナがリソース不足になることがないようにする必要があります。
  - Docker run コマンドの --memory フラグを使用して、Docker コンテナで、コンテナによって使用されるメモリに対してハード制限が適用されるようにする必要があります。
  - Docker run コマンドの --cpuset-cpus フラグを使用して、Docker コンテナで、実行中のコンテナによって使用される CPU の容量に制限が適用されるようにする必要があります。
- --pids-limit には 300 を設定して、任意の時点でコンテナ内で生成されるカーネルスレッドの数を制限する必要があります。これは、DoS 攻撃を防ぐためです。
- Docker run コマンドの --restart フラグを使用して、障害発生時のコンテナ再起動ポリシーを 5 に設定する必要があります。
- コンテナは、必ず、コンテナの起動後にヘルスステータスが［[正常]］と表示されてから使用する必要があります。コンテナのヘルスステータスを確認するには、次のコマンドを実行します。

```
docker ps <container_name/ID>
```

- Docker コンテナは、常に root 以外のユーザ (nds) として起動されます。追加のセキュリティ対策として、デーモンでのユーザ名前空間の再マッピングを有効にして、コンテナ内からの特権昇格攻撃を防止します。ユーザ名前空間の再マッピングの詳細については、「ユーザ名前空間でコンテナを分離する」を参照してください。

Docker コンテナとしての Identity Console の展開

このセクションでは、次の手順について説明します。

- 22 ページの 「OSP コンテナの展開」
- 24 ページの 「Docker コンテナとしての Identity Console の展開」
- 26 ページの 「Identity Console を Docker として使用するマルチツリー」

OSP コンテナの展開

以下の手順を実行して、OSP コンテナを展開します。

1. Software License and Download (https://sld.microfocus.com/) にログインして、[ソフトウェアダウンロード] ページに移動します。
2. 次の項目を選択します。
   - 製品 : eDirectory
   - 製品名 : eDirectory per User Sub SW E-LTU
   - バージョン : 9.2
3. ファイル IdentityConsole_< バージョン >_Containers_tar.zip をダウンロードします。
4. ダウンロードしたファイルをフォルダに抽出します。
5. 要件に従ってサイレントプロパティファイルを変更します。サンプルのサイレントプロパティファイルを次に示します。

```
# Silent file for osp with edirapi
## Static contents Do not edit - starts
INSTALL_OSP=true
DOCKER_CONTAINER=y
EDIRAPI_PROMPT_NEEDED=y
UA_PROMPT_NEEDED=n
SSPR_PROMPT_NEEDED=n
RPT_PROMPT_NEEDED=n
CUSTOM_OSP_CERTIFICATE=y
## Static contents Do not edit - ends

# OSP Details
SSO_SERVER_HOST=osp.example.com
SSO_SERVER_SSL_PORT=8543
OSP_COMM_TOMCAT_KEYSTORE_FILE=/config/tomcat.ks
OSP_COMM_TOMCAT_KEYSTORE_PWD=novell
SSO_SERVICE_PWD=novell
```

```
OSP_KEYSTORE_PWD=novell
IDM_KEYSTORE_PWD=novell
OSP_CUSTOM_NAME="Identity Console"
USER_CONTAINER="o=novell"
ADMIN_CONTAINER="o=novell"

# IDConsole Details
IDCONSOLE_HOST=192.168.1.1
IDCONSOLE_PORT=9000
EDIRAPI_TREENAME=ed913

#If ENABLE_CUSTOM_CONTAINER_CREATION is set to y
#ie., when you have user and admin container different from o=data
#    and they need to be created in eDir
#then CUSTOM_CONTAINER_LDIF_PATH should be entered as well
ENABLE_CUSTOM_CONTAINER_CREATION=n
#ENABLE_CUSTOM_CONTAINER_CREATION=y
#CUSTOM_CONTAINER_LDIF_PATH=/config/custom-osp.ldif

# eDir Details
ID_VAULT_HOST=192.168.1.1
ID_VAULT_LDAPS_PORT=636
ID_VAULT_ADMIN_LDAP="cn=admin,o=novell"
ID_VAULT_PASSWORD=novell
```

注 : サイレントプロパティ (DOS テキスト) ファイルの使用時にスペースの制約を回避するには、dos2unix ツールを使用して、DOS テキストファイルを UNIX 形式に変換する必要があります。次のコマンドを実行して、テキストファイルを、DOS 行末から Unix 行末に変換します。

dos2unix filename

次に例を示します。

dos2unix samplefile

---

6 iManager を使用してサーバ証明書（cert.der）を生成し、それをキーストア（tomcat.ks）にインポートします。サイレントプロパティファイルとキーストア（tomcat.ks）を任意のディレクトリにコピーします。たとえば、/data。次の手順を実行して、サーバ証明書を作成し、キーストアにインポートします。

6a 次のコマンドを実行して、キーストア（tomcat.ks）を作成します。キーを生成し、マシンの CN 名または完全修飾ホスト名が IP アドレスであることを確認します。

```
keytool -genkey -alias osp -keyalg RSA -storetype pkcs12 -keystore /
opt/certs/tomcat.ks -validity 3650 -keysize 2048 -dname "CN=blr-
osp48-demo.labs.blr.novell.com" -keypass novell -storepass novell
```

6b 次のコマンドを実行して、証明書署名要求を作成します。たとえば、cert.csr。

```
keytool -certreq -v -alias osp -file /opt/certs/cert.csr -keypass
novell -keystore /opt/certs/tomcat.ks -storepass novell
```

6c この cert.csr を iManager に渡し、osp.der サーバ証明書を取得します。キータイプを [カスタム] として選択してください。キー使用オプションは、データ暗号化、キー暗号化、デジタル署名として選択し、証明書の件名の代替名フィールドに

OSP サーバの IP アドレスまたはホスト名を含むように選択します。詳細については、「Creating a Server Certificate Object(サーバ証明書オブジェクトの作成)」を参照してください。

6d 次のコマンドを実行して、CA 証明書（SSCert.der）とサーバ証明書（cert.der）を tomcat.ks キーストアにインポートします。

```
keytool -import -trustcacerts -alias root -keystore /opt/certs/
tomcat.ks -file /opt/certs/SSCert.der -storepass novell -noprompt
```

```
keytool -import -alias osp -keystore /opt/certs/tomcat.ks -file /
opt/certs/cert.der -storepass novell -noprompt
```

7 次のコマンドを実行して OSP イメージをロードします。

```
docker load --input osp.tar.gz
```

8 次のコマンドを使用してコンテナを展開します。

```
docker run -d --name OSP_Container --network=host -e
SILENT_INSTALL_FILE=/config/silent.properties -v /data:/config
osp:<version>
```

次に例を示します。

```
docker run -d --name OSP_Container --network=host -e
SILENT_INSTALL_FILE=/config/silent.properties -v /data:/config
osp:6.3.9
```

Docker コンテナとしての Identity Console の展開

このセクションでは、Identity Console を Docker コンテナとして展開する手順を説明します。

注 : この手順で説明する環境設定パラメータ、サンプル値、および例は参照専用です。これらを、ご使用の運用環境で直接使用しないようにしてください。

---

1 SLD: Software License and Download (https://sld.microfocus.com/) にログインし、[ソフトウェアダウンロード] ページに移動します。

2 次の項目を選択します。

- 製品 : eDirectory
- 製品名 : eDirectory per User Sub SW E-LTU
- バージョン : 9.2

3 ファイル IdentityConsole_< バージョン >_Container.tar.zip をダウンロードします。

4 イメージをローカルの Docker レジストリにロードする必要があります。次のコマンドを使用して、IdentityConsole_<version>_Containers.tar.gz ファイルを抽出してロードします。

```
tar -xvf IdentityConsole_<version>_Containers.tar.gz
```

```
docker load --input identityconsole.tar.gz
```

5 次のコマンドで、Identity Console Docker コンテナを作成します。

```
docker create --name <identityconsole-container-name> --env
ACCEPT_EULA=Y --network=<network-type> --volume <volume-name>:/config/
identityconsole:<version>
```

次に例を示します。

```
docker create --name identityconsole-container --env ACCEPT_EULA=Y --
network=host --volume IDConsole-volume:/config/
identityconsole:1.5.0.0000.
```

注

- ACCEPT_EULA 環境変数を「Y」に設定することによって、EULA に同意することができます。対話型モードの Docker create コマンドで -it オプションを使用することにより、コンテナの起動中に画面のプロンプトから EULA に同意することもできます。
- 上のコマンドの --volume パラメータによって、環境設定データとログデータを保存するボリュームが作成されます。この場合、IDConsole-volume というサンプルボリュームが作成されます。

6 次のコマンドを使用して、ローカルファイルシステムからコンテナに、サーバ証明書ファイルを /etc/opt/novell/eDirAPI/cert/keys.pfx としてコピーします。サーバ証明書の作成方法については、「11 ページの 「前提条件」」を参照してください。

```
docker cp <absolute path of server certificate file> <identityconsole-
container-name>:/etc/opt/novell/eDirAPI/cert/keys.pfx
```

次に例を示します。

```
docker cp /home/user/keys.pfx identityconsole-container:/etc/opt/
novell/eDirAPI/cert/keys.pfx
```

複数の eDirectory ツリーに接続する場合、すべての接続ツリーに対して少なくとも 1 つの keys.pfx サーバ証明書を取得する必要があります。

7 次のコマンドを使用して、ローカルファイルシステムからコンテナに、CA 証明書ファイル (.pem) を /etc/opt/novell/eDirAPI/cert/SSCert.pem としてコピーします。CA 証明書の取得の詳細については、「11 ページの 「前提条件」」を参照してください。

```
docker cp <absolute path of CA certificate file> <identityconsole-
container-name>:/etc/opt/novell/eDirAPI/cert/SSCert.pem
```

次に例を示します。

```
docker cp /home/user/SSCert.pem identityconsole-container:/etc/opt/
novell/eDirAPI/cert/SSCert.pem
```

ユーザが複数の eDirectory ツリーに接続する必要がある場合は、次のセクションを参照してください。26 ページの 「Identity Console を Docker として使用するマルチツリー」

8 要件に従って構成ファイルを変更し、次のコマンドを使用して、環境設定ファイル (edirapi.conf) をローカルのファイルシステムからコンテナに /etc/opt/novell/eDirAPI/conf/edirapi.conf としてコピーします。

```
docker cp <absolute path of configuration file> <identityconsole-container-name>:/etc/opt/novell/eDirAPI/conf/edirapi.conf
```

次に例を示します。

```
docker cp /home/user/edirapi.conf identityconsole-container:/etc/opt/novell/eDirAPI/conf/edirapi.conf
```

9 次のコマンドで、Docker コンテナを起動します。

```
docker start <identityconsole-container-name>
```

次に例を示します。

```
docker start identityconsole-container
```

注 : 次のログファイルは、/var/lib/docker/volumes/< ボリューム名 >/_data/eDirAPI/var/log ディレクトリにあります。

- edirapi.log - これは、edirapi でさまざまなイベントをログ記録し、問題をデバッグする場合に使用します。
- edirapi_audit.log - edirapi の監査イベントをログに記録するために使用されます。ログは、CEF 監査フォーマットに従います。
- container-startup.log - これは、Identity Console Docker コンテナのインストールログをキャプチャするために使用されます。

Identity Console を Docker として使用するマルチツリー

Identity Console を使用すると、ユーザはツリーの個々の CA 証明書を取得することにより、複数のツリーに接続できます。

たとえば、3 つの eDirectory ツリーに接続する場合、3 つすべての CA 証明書を Docker コンテナにコピーする必要があります。

```
docker cp /home/user/SSCert1.pem identityconsole-container:/etc/opt/novell/eDirAPI/cert/SSCert.pem
```

```
docker cp /home/user/SSCert2.pem identityconsole-container:/etc/opt/novell/eDirAPI/cert/SSCert1.pem
```

```
docker cp /home/user/SSCert3.pem identityconsole-container:/etc/opt/novell/eDirAPI/cert/SSCert2.pem
```

次のコマンドを実行して、Identity Console を再起動します。

```
docker restart <identityconsole-container-name>
```

スタンドアロン Identity Console の展開

- 27 ページの 「スタンドアロン Identity Console の展開（非 Docker）」
- 28 ページの 「スタンドアロンの Identity Console を使用するマルチツリー」

スタンドアロン Identity Console の展開（非 Docker）

このセクションでは、スタンドアロン Identity Console を展開する手順について説明します。

1. SLD: Software License and Download (https://sld.microfocus.com/) にログインし、[ソフトウェアダウンロード] ページに移動します。
2. 次の項目を選択します。
   - 製品 : eDirectory
   - 製品名 : eDirectory per User Sub SW E-LTU
   - バージョン : 9.2
3. 最新の Identity Console ビルドをダウンロードします。
4. ダウンロードしたファイルをフォルダに抽出します。
5. シェルを開き、Identity Console ビルドを抽出したフォルダに移動します。
6. root または root と同等のユーザとしてログインしている間に、次のコマンドを実行します。

   ```
   ./identityconsole_install
   ```

7. 概要を読み、[**[ENTER]**] をクリックします。
8. 「Y」をクリックして、ライセンス契約を受け入れます。これにより、システムに必要なすべての RPM がインストールされます。
9. Identity Console サーバのホスト名 (FQDN)/IP アドレスを入力します。
10. Identity Console がリッスンするポート番号を入力します。デフォルト値は 9000 です。
11. OSP と Identity Console を統合するオプション、または Identity Console で LDAP ログインを使用するオプションを入力します。
12. OSP を Identity Console と統合する場合 :
    1. eDirectory/ アイデンティティボールトサーバのドメイン名 /IP アドレスを LDAPS ポート番号で入力します。

       次に例を示します。

       192.168.1.1:636
    2. eDirectory/ アイデンティティボールトのユーザ名を入力します。

       次に例を示します。

       cn=admin,ou=org_unit,o=org
    3. eDirectory/ アイデンティティボールトのパスワードを入力します。
    4. eDirectory/ アイデンティティボールトのパスワードを再度入力して、パスワードを確認します。
    5. OSP サーバのドメイン名 /IP アドレスと SSO サーバの SSL ポート番号を入力します。
    6. OSP クライアント ID を入力します。

7. OSP クライアントパスワードを入力します。
8. eDirectory/ アイデンティティボールトのツリー名を入力します。

13 フォルダを含むルート認証局証明書 (SSCert.pem) パスを入力します。

次に例を示します。

/home/Identity_Console/certs

注 : ユーザは、cert フォルダ内にサブディレクトリを作成しないようにしてください。

---

14 ファイル名を含むサーバ証明書 (keys.pfx) パスを入力します。

次に例を示します。

/home/Identity_Console/keys.pfx

15 サーバ証明書のパスワードを入力します。パスワードが正しく入力されていることを確認するには、サーバ証明書のパスワードを再入力します。インストールが開始されます。

注 : 次のログファイルは、/var/opt/novell/eDirAPI/log ディレクトリにあります。

- edirapi.log - これは、edirapi でさまざまなイベントをログ記録し、問題をデバッグする場合に使用します。
- edirapi_audit.log - edirapi の監査イベントをログに記録するために使用されます。ログは、CEF 監査フォーマットに従います。
- identityconsole_install.log - これは、Identity Console のインストールログをキャプチャするために使用されます。

Identity Console プロセス開始 / 中止のログは、/var/log/messages ファイルにあります。

注 : NetIQ では、Identity Console と eDirectory を同じマシンにインストールする際に、少なくとも 1 つの eDirectory インスタンスが使用可能なマシンにインストールすることをお勧めします。

スタンドアロンの Identity Console を使用するマルチツリー

複数の eDirectory ツリーに接続する場合は、ツリーの個々の CA 証明書を取得する必要があります。

たとえば、3 つの eDirectory ツリーに接続する場合、3 つすべての CA 証明書を etc/opt/novell/eDirAPI/cert/ ディレクトリにコピーする必要があります。

```
cp /home/user/SSCert.pem /etc/opt/novell/eDirAPI/cert/SSCert1.pem
cp /home/user/SSCert.pem /etc/opt/novell/eDirAPI/cert/SSCert2.pem
cp /home/user/SSCert.pem /etc/opt/novell/eDirAPI/cert/SSCert3.pem
```

次のいずれかのコマンドを実行して、Identity Console を再起動します。

```
/usr/bin/identityconsole restart
```

または

```
systemctl restart netiq-identityconsole.service
```

ワークステーションとしての Windows 上の Identity Console

Identity Console は Windows 上でワークステーションとして起動できます。REST サービスが実行されている必要があります。したがって、起動した際に、eDirAPI プロセスが edirapi.exe cmd プロンプトで実行されます。この edirapi.exe 端末を閉じた場合、Identity Console は機能しなくなります。

次の手順では、Windows で Identity Console を実行する方法について説明します。

1. SLDSoftware License and Download (https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0) にログインし、[ソフトウェアダウンロード] ページに移動します。
2. 次の項目を選択します。
   - 製品 : eDirectory
   - 製品名 : eDirectory per User Sub SW E-LTU
   - バージョン : 9.2
3. ファイル IdentityConsole_< バージョン >_workstation_win_x86_64.zip をダウンロードします。
4. ダウンロードした IdentityConsole_< バージョン >_workstation_win_x86_64.zip ファイルをフォルダに抽出します。
5. 抽出されたフォルダ IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert に移動し、ルート認証局 CA SSCert.pem およびサーバ証明書 keys.pfx をコピーします。

   証明書を取得するには、次のセクションを参照してください。18 ページの 「ワークステーションのシステム要件と前提条件」

   ユーザが複数の eDirectory ツリーに接続する必要がある場合は、次のセクションを参照してください。30 ページの 「Identity Console をワークステーションとして使用するマルチツリー」

   注 : サーバ証明書名は、常に keys.pfx として指定する必要があります。
6. ビルドが抽出されたフォルダに移動し、run.bat ファイル (Windows バッチファイル) をダブルクリックします。
7. サーバ証明書 (keys.pfx) パスワードをコマンドプロンプトに入力します。

   eDirAPI プロセス端末 (edirapi.exe) が起動し、Identity Console のログインページが表示されます。

注 :

- eDirAPI プロセス端末 (edirapi.exe) がすでに実行されている場合は、ビルド抽出フォルダから identityconsole.exe を実行します。

- ユーザは次のログを \IdentityConsole_150_workstation_win_x86_64\eDirAPI\log で見つけることができます。

  edirapi.log - これは edirapi でさまざまなイベントを記録し、問題をデバッグする場合に使用します。

  edirapi_audit.log - これは edirapi の監査イベントをログに記録するために使用されます。ログは、CEF 監査フォーマットに従います。

- OSP ベースのログインは、ワークステーションモードではサポートされていません。
- Identity Console ワークステーションがリッスンしているポートは 9000 ポートのみです。edirapi_win.conf ファイルは変更しないでください。

Identity Console をワークステーションとして使用するマルチツリー

Identity Console を使用すると、ユーザはツリーの個々の CA 証明書を取得することにより、複数のツリーに接続できます。

1. Identity Console ワークステーションと eDirAPI 端末を閉じます。
2. CA 証明書 SSCert.pem を場所 IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert にコピーします。

   たとえば、3 つの eDirectory ツリーに接続する場合は、CA 証明書をそれぞれ SSCert1.pem、SSCert2.pem、および SSCert3.pem としてコピーします。
3. ビルドが抽出されたフォルダに移動し、run.bat ファイル (Windows バッチファイル) をダブルクリックします。
4. 端末プロンプトに keys.pfx パスワードを入力し、目的の eDirectory ツリーにログインします。

Identity Console の停止と再起動

- 30 ページの 「Identity Console を Docker コンテナとして停止と再起動する」
- 31 ページの 「スタンドアロンの Identity Console の停止と再起動」
- 31 ページの 「Identity Console ワークステーションの終了と再起動」

Identity Console を Docker コンテナとして停止と再起動する

Identity Console を停止するには、次のコマンドを実行します。

```
docker stop <identityconsole-container-name>
```

Identity Console を再起動するには、次のコマンドを実行します。

```
docker restart <identityconsole-container-name>
```

Identity Console を起動するには、次のコマンドを実行します。

```
docker start <identityconsole-container-name>
```

スタンドアロンの Identity Console の停止と再起動

Identity Console を停止するには、次のいずれかのコマンドを実行します。

```
/usr/bin/identityconsole stop
```

または

```
systemctl stop netiq-identityconsole.service
```

Identity Console を再起動するには、次のいずれかのコマンドを実行します。

```
/usr/bin/identityconsole restart
```

または

```
systemctl restart netiq-identityconsole.service
```

Identity Console を起動するには、次のいずれかのコマンドを実行します。

```
/usr/bin/identityconsole start
```

または

```
systemctl start netiq-identityconsole.service
```

Identity Console ワークステーションの終了と再起動

アプリケーションとプロセスを閉じるには、次の手順に従います。

1. Identity Console デスクトップウィンドウアプリケーションを閉じます。
2. eDirAPI プロセス端末を閉じて、eDirAPI プロセスを停止します。

Identity Console ワークステーションを再起動するには、ビルドを抽出したフォルダに移動し、run.bat ファイル (Windows バッチファイル) をダブルクリックします。

注 : eDirAPI プロセス端末がすでに実行されている場合は、ビルド抽出フォルダから identityconsole.exe を実行して、Identity Console ワークステーションを再起動します。

データ永続性の管理

Identity Console コンテナとともに、データ永続性のためのボリュームも作成されます。ボリュームを使用して古いコンテナの環境設定パラメータを使用するには、次の手順を実行します。

1. 次のコマンドを使用して、現在の Docker コンテナを停止します。

   ```
   docker stop identityconsole-container
   ```

2 Docker ボリューム (edirapi-volume-1) に保存されている古いコンテナのアプリケーションデータを使用して、2 番目のコンテナを作成します。

```
docker create --name identityconsole-container-2 --network=host --
volume edirapi-volume-1:/config/ identityconsole:1.0.0
```

3 次のコマンドを使用して、2 番目のコンテナを開始します。

```
docker start identityconsole-container-2
```

4 (省略可能) これで、次のコマンドを使用して最初のコンテナを削除できるようになりました。

```
docker rm identityconsole-container
```

Azure Kubernetes サービスでの Identity Console の展開

Azure Kubernetes Service (AKS) は、クラスタの展開と管理を可能にするマネージド Kubernetes サービスです。このセクションでは、次の手順について説明します。

AKS クラスタでの Identity Console の展開

このセクションでは、AKS クラスタに Identity Console を展開するための次の手順について説明します。

- 32 ページの 「Azure Container Registry (ACR) の作成」
- 33 ページの 「Kubernetes クラスタの設定」
- 34 ページの 「標準 SKU パブリック IP アドレスの作成」
- 34 ページの 「Cloud Shell のセットアップと Kubernetes クラスタへの接続」
- 34 ページの 「アプリケーションの展開」

Azure Container Registry (ACR) の作成

Azure Container Registry (ACR) は、Docker コンテナイメージ用の、Azure ベースのプライベートレジストリです。

詳細については、「Create container registry - Portal(コンテナレジストリの作成 - ポータル)」の「Create an Azure container registry using the Azure portal (Azure ポータルを使用して Azure Container Registry を作成する)」セクションを参照するか、次の手順を実行して、Azure Container Registry (ACR) を作成します。

1. Azure ポータルにサインインします。
2. [**リソースの作成**] > [**コンテナ**] > [**コンテナレジストリ**] に移動します。
3. [**基本**] タブで、[リソースグループ] と [レジストリ名] の値を指定します。レジストリ名は、Azure 内で固有で、最小 5 文字および最大 50 文字の英数字を含む必要があります。

   残りの設定のデフォルト値を受け入れます。

4. [[レビュー + 作成]] をクリックします。
5. [[作成]] をクリックします。
6. Azure CLI にサインインし、次のコマンドを実行して、Azure Container Registry にログインします。

   ```
   az acr login --name registryname
   ```

   例 :

   ```
   az acr login --name < idconsole >
   ```

7. 次のコマンドを使用して、Azure Container Registry のログインサーバを取得します。

   ```
   az acr show --name registryname --query loginServer --output table
   ```

   例 :

   ```
   az acr show --name < idconsole > --query loginServer --output table
   ```

8. 次のコマンドを使用して、Identity Console のローカルイメージに、ACR ログインサーバ (registryname.azureacr.io) の名前をタグ付けします。

   ```
   docker tag idconsole-image <login server>/idconsole-image
   ```

   次に例を示します。

   ```
   docker tag identityconsole:<version> registryname.azurecr.io/
   identityconsole:<version>
   ```

9. タグ付きイメージをレジストリにプッシュします。

   ```
   docker push <login server>/idconsole: <version>
   ```

   次に例を示します。

   ```
   docker push registryname.azurecr.io/identityconsole:<version>
   ```

10. 次のコマンドを使用して、レジストリ内のイメージのリストを取得します。

    ```
    az acr show --name registryname --query loginServer --output table
    ```

Kubernetes クラスタの設定

Azure ポータルまたは CLI を使用して、kubernetes サービスリソースを作成します。

ノードを使用して、Azure で Kubernetes サービスリソースを作成する手順の詳細については、Azure クイックスタートの「AKS クラスタの作成」を参照してください。

注 :

- 必ず Azure CNI をネットワークとして選択してください。
- 既存の仮想ネットワーク (eDirectory サーバがサブネット内に展開されているネットワーク) を選択します。
- Identity Console イメージが使用可能な既存のコンテナレジストリを選択します。

標準 SKU パブリック IP アドレスの作成

Kubernetes クラスタリソースグループの下のパブリック IP アドレスリソースは、アプリケーションのロードバランサ IP として機能します。

詳細な手順については、「Create public IP address – Portal(パブリック IP アドレスの作成 - ポータル)」の「Azure ポータルを使用してパブリック IP アドレスを作成する」を参照してください。

Cloud Shell のセットアップと Kubernetes クラスタへの接続

すべての操作に対して Azure ポータルで使用可能なクラウドシェルを使用します。

Azure ポータルでクラウドシェルをセットアップするには、「バッシュ - クイックスタート」の「Cloud Shell の起動」セクションを参照するか、次の手順を実行して Cloud Shell をセットアップし、Kubernetes クラスタに接続します。

1. Azure ポータルで、  ボタンをクリックして Cloud Shell を開きます。

   注 : Kubernetes クラスタを管理するには、Kubernetes コマンドラインクライアントの kubectl を使用します。Azure Cloud Shell を使用している場合は、kubectl はすでにインストールされています。

2. 次のコマンドを使用して Kubernetes クラスタに接続するために kubectl を設定します。

   ```
   az aks get-credentials --resource-group "resource group name" --name
   "Kubernetes cluster name"
   ```

   次に例を示します。

   ```
   az aks get-credentials --resource-group myResourceGroup --name
   myAKSCluster
   ```

3. 次のコマンドを使用して、クラスタノードのリストを検証します。

   ```
   kubectl get nodes
   ```

アプリケーションの展開

Identity Console を展開するには、idc-services.yaml、idc-statefulset.yaml、idc-storageclass.yaml、および idc-pvc.yaml のサンプルファイルを使用できます。

要件に従って独自の yaml ファイルを作成することもできます。

1. 下記コマンドを使用してストレージクラスリソースを作成します。

   ```
   kubectl apply -f <location of the YAML file>
   ```

   次に例を示します。

   ```
   kubectl apply -f idc-storageclass.yaml
   ```

(オプション)Azure ファイル共有で永続ボリュームを動的に作成して使用する方法の詳細については、「Azure Kubernetes Service (AKS) の Azure ファイルで永続ボリュームを動的に作成して使用する」を参照してください。

ストレージクラスリソースファイルのサンプルを次に示します。

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: azurefilesc
provisioner: kubernetes.io/azure-file
mountOptions:
  - dir_mode=0777
  - file_mode=0777
  - uid=0
  - gid=0
  - mfsymlinks
  - cache=strict
  - actimeo=30
parameters:
  skuName: Standard_LRS
  shareName: fileshare
~
```

ストレージクラスリソースにより、ダイナミックストレージプロビジョニングが可能です。Azure ファイル共有の作成方法を定義するために使用されます。

2. 下記コマンドを使用して、storageclass の詳細を表示します。

```
kubectl get sc
```

3. idc-pvc.yaml ファイルを使用して pvc リソースを作成します。

```
kubectl apply -f <location of the YAML file>
```

次に例を示します。

```
kubectl apply -f idc.pvc.yaml
```

サンプルの pvc リソースファイルを次に示します。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvcforsc
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: azurefilesc
  resources:
    requests:
      storage: 5Gi
```

永続ボリューム要求リソースは、ファイル共有を作成します。永続ボリューム要求 (PVC) は、ストレージクラスオブジェクトを使用して、Azure ファイル共有を動的にプロビジョニングします。

4. edirapi.conf、CA 証明書、およびサーバ証明書を Cloud Shell にアップロードします。

   Cloud Shell の [ファイルのアップロード / ダウンロード] ボタンアイコン  をクリックし、edirapi.conf、SSCert.pem、keys.pfx ファイルをアップロードします。

   注 : edirapi.conf にはパラメータ「origin」があります。ここでは、Identity Console アプリケーションにアクセスする IP アドレスを指定する必要があります。(34 ページの「標準 SKU パブリック IP アドレスの作成」セクションで作成された IP アドレスを使用します。)

   Identity Console の展開には、サーバ証明書 (keys.pfx) が必要です。

   サーバ証明書を作成する際には、件名の代替名に有効な DNS 名を指定してください。

   有効な DNS 名を作成する手順は次のとおりです。

   StatefulSet を使用して展開される一般的なポッドには、次のような DNS 名があります。{statefulsetname}-{ordinal}.{servicename}.{namespace}.svc.cluster.local

   - idconsole-statefulset.yaml ファイル内の StatefulSet 名が idconsole-app の場合、statefulsetname = idconsole-app
   - 最初のポッドの場合、ordinal = 0
   - idconsole -statefulset.yaml ファイルで serviceName を idconsole として定義した場合、serviceName = idconsole
   - デフォルトが名前空間の場合、namespace=default

   出力 : idconsole-app-0.idcosole.default.svc.cluster.local

   ---

5. 証明書と一緒に設定ファイルを保存する、Kubernetes クラスタ内に configmap リソースを作成します。

   コマンドを実行する前に、ディレクトリ内にファイル (edirapi.conf、SSCert.pem、および keys.pfx) が存在する必要があります。

   ```
   kubectl create configmap <confgimapName> --from-file= "path where the files are present"
   ```

   次に例を示します。

   ```
   kubectl create configmap config-data --from-file=/data
   ```

6. kubectl describe コマンドを使用して、configmap オブジェクトの詳細を表示します。

   ```
   kubectl describe configmap <configmapName>
   ```

   次に例を示します。

   ```
   kubectl describe configmap confg-data
   ```

7. コンテナを展開するための StatefulSet リソースを作成します。

   下記コマンドを実行して、コンテナを展開します。

   ```
   kubectl apply -f <location of the YAML file>
   ```

   次に例を示します。

```
kubectl apply -f idc-statefulset.yaml
```

サンプルの StatefulSet リソースファイルを次に示します。

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: idconsole-app
spec:
  serviceName: idconsole
  selector:
    matchLabels:
      app: idconsole
  replicas: 1
  template:
    metadata:
      labels:
        app: idconsole
    spec:
      containers:
      - name: idconsole-container
        image: registryname.azurecr.io/identityconsole:<version>
        env:
        - name: ACCEPT_EULA
          value: "Y"
        ports:
        - containerPort: 9000
        volumeMounts:
          - name: configfiles
            mountPath: /config/data
          - name: datapersistenceandlog
            mountPath: /config
            subPath: log
      volumes:
        - name: configfiles
          configMap:
            name: config-data
        - name: datapersistenceandlog
          persistentVolumeClaim:
            claimName: pvcforsc
```

8. 次のコマンドを実行して、展開されたポッドのステータスを検証します。

```
kubectl get pods -o wide
```

9. loadBalancer タイプのサービスリソースを作成します。

yaml ファイルで指定されているサービスのタイプは、loadBalancer です。

下記コマンドを使用してサービスリソースを作成します。

```
kubectl apply -f <location of the YAML file>
```

次に例を示します。

```
kubectl apply -f ids-service.yaml
```

サンプルのサービスリソースファイルを次に示します。

```
apiVersion: v1
kind: Service
metadata:
  name: idconsole-service
  labels:
    run: idconsole-service
spec:
  type: LoadBalancer
  loadBalancerIP: xx.xx.xx.xx
  selector:
    app: idconsole
  ports:
   - port: 9000
     targetPort: 9000
     protocol: TCP
```

下記コマンドを使用して、EXTERNAL-IP アドレス (または loadBalancerIP) を確認します。

```
kubectl get svc -o wide
```

10. EXTERNAL-IP (または loadBalancerIP アドレス) を使用して URL を起動します。

    次に例を示します。

    https://<EXTERNAL-IP>:9000/identityconsole

サーバ証明書の変更

このセクションでは、Docker コンテナおよびスタンドアロン Identity Console でサーバ証明書を変更する方法について説明します。

- 38 ページの 「Docker コンテナでのサーバ証明書の変更」
- 39 ページの 「スタンドアロンの Identity Console でのサーバ証明書の変更」

Docker コンテナでのサーバ証明書の変更

Docker コンテナ内のサーバ証明書を変更するには、次の手順を実行します。

1 次のコマンドを実行して、コンテナの任意の場所に新しいサーバ証明書をコピーします。

次に例を示します。

```
docker cp /path/to/new-keys.pfx <container_id/name>:/tmp/new-keys.pfx
```

2 次のコマンドを使用してコンテナにログインします。

```
docker exec -it <container_name> bash
```

3 NLPCERT を実行して、擬似ユーザとしてキーを保存します。

```
LD_LIBRARY_PATH=/opt/novell/lib64/:/opt/novell/eDirectory/lib64/:/opt/
netiq/common/openssl/lib64/ /opt/novell/eDirAPI/sbin/nlpcert -i /tmp/
new-keys.pfx -o /etc/opt/novell/eDirAPI/conf/ssl/private/cert.pem
```

4 次のコマンドを使用して、コンテナコンソールを終了します。

```
exit
```

5 次のコマンドを入力して、コンテナを再起動します。

```
docker restart <container name>
```

スタンドアロンの Identity Console でのサーバ証明書の変更

スタンドアロンコンテナでサーバ証明書を変更するには、次の手順を実行します。

1 NLPCERT を実行して、キーを保存します。

```
su - nds -c "LD_LIBRARY_PATH=/opt/novell/lib64/:/opt/novell/eDirectory/
lib64/:/opt/netiq/common/openssl/lib64/ /opt/novell/eDirAPI/sbin/
nlpcert -i /Expiredcert/noexpire/new-keys.pfx -o /etc/opt/novell/
eDirAPI/conf/ssl/private/cert.pem"
```

2 Identity Console を再起動します。

```
systemctl restart netiq-identityconsole.service
```

3 Identity Console のアップグレード

この章では、Identity Console を最新バージョンにアップグレードするプロセスについて説明します。アップグレードの準備をするには、11 ページの第 1 章「Identity Console のインストール計画」に記載されている前提条件とシステム要件を確認してください。

このセクションでは、次の手順について説明します。

- 41 ページの 「Docker コンテナとしての Identity Console のアップグレード」
- 43 ページの 「スタンドアロン Identity Console(非 Docker) のアップグレード」
- 44 ページの 「OSP コンテナのアップグレード」

Docker コンテナとしての Identity Console のアップグレード

Identity Console イメージの新しいバージョンが使用可能になった場合、管理者は、最新バージョンの Identity Console をコンテナに展開するためのアップグレード手順を実行できます。アップグレードを実行する前に、すべての必要なアプリケーション関連データを Docker ボリュームに永続的に保存するようにしてください。Docker コンテナを使用して Identity Console をアップグレードするには、次の手順を実行します。

1. Software License and Download (https://sld.microfocus.com/) から最新バージョンの Docker イメージをダウンロードしてロードし、21 ページの 「Identity Console の展開」に記載されている最新バージョンの Identity Console をインストールする手順を実行します。
2. 最新の Docker イメージがロードされた後、次のコマンドを使用して現在の Docker コンテナを停止します。

   ```
   docker stop identityconsole-container
   ```

3. (オプション) 共有ボリュームのバックアップを取ります。
4. 次のコマンドを実行して、既存の Identity Console コンテナを削除します。

   ```
   docker rm <container name>
   ```

   次に例を示します。

   ```
   docker rm identityconsole-container
   ```

5. (省略可能) 次のコマンドを実行して、廃止された Identity Console Docker イメージを削除します。

   ```
   docker rmi identityconsole
   ```

6. 次のコマンドで、Identity Console Docker コンテナを作成します。

```
docker create --name <identityconsole-container-name> --env
ACCEPT_EULA=Y --network=<network-type> --volume <volume-name>:/config/
identityconsole:<version>
```

例 :

```
docker create --name identityconsole-container --env ACCEPT_EULA=Y --
network=host --volume IDConsole-volume:/config/
identityconsole:1.5.0.0000
```

注

- ACCEPT_EULA 環境変数を「Y」に設定することによって、EULA に同意することができます。対話型モードの Docker create コマンドで -it オプションを使用することにより、コンテナの起動中に画面のプロンプトから EULA に同意することもできます。
- 上のコマンドの --volume パラメータによって、環境設定データとログデータを保存するボリュームが作成されます。この場合、IDConsole-volume というサンプルボリュームが作成されます。

---

7 次のコマンドを使用して、ローカルファイルシステムから新しく作成されたコンテナに、サーバ証明書ファイルを /etc/opt/novell/eDirAPI/cert/keys.pfx としてコピーします。

```
docker cp <absolute path of server certificate file> identityconsole-
container:/etc/opt/novell/eDirAPI/cert/keys.pfx
```

次に例を示します。

```
docker cp /home/user/keys.pfx identityconsole-container:/etc/opt/
novell/eDirAPI/cert/keys.pfx
```

複数の eDirectory ツリーに接続している場合、接続されているすべてのツリーに対して少なくとも 1 つの keys.pfx サーバ証明書をコピーする必要があります。

8 次のコマンドを使用して、ローカルファイルシステムから新しく作成されたコンテナに、CA 証明書ファイル (.pem) を /etc/opt/novell/eDirAPI/cert/SSCert.pem としてコピーします。

```
docker cp <absolute path of CA certificate file> identityconsole-
container:/etc/opt/novell/eDirAPI/cert/SScert.pem
```

次に例を示します。

```
docker cp /home/user/SSCert.pem identityconsole-container:/etc/opt/
novell/eDirAPI/cert/SSCert.pem
```

複数の eDirectory ツリーに接続する場合は、すべての接続ツリーに対して個々の CA 証明書を取得する必要があります。たとえば、3 つの eDirectory ツリーに接続する場合、3 つすべての CA 証明書を Docker コンテナにコピーする必要があります。

```
docker cp /home/user/SSCert.pem identityconsole-container:/etc/opt/
novell/eDirAPI/cert/SSCert.pem
docker cp /home/user/SSCert1.pem identityconsole-container:/etc/opt/
novell/eDirAPI/cert/SSCert1.pem
docker cp /home/user/SSCert2.pem identityconsole-container:/etc/opt/
novell/eDirAPI/cert/SSCert2.pem
```

注 : Identity Console 1.4 以降では、環境設定ファイル (edirapi.conf) に「*ldapuser*」パラメータ、「*ldappassword*」パラメータ、および「*ldapserver*」パラメータは明示的に含まれていません。「*bcert*」パラメータ値には、ルート認証局証明書のディレクトリパスが含まれる必要があります。たとえば、bcert = "/etc/opt/novell/eDirAPI/cert/" など。また、「*origin*」パラメータは「*check-origin*」パラメータとは独立しています。DNS 設定を使用する場合は必須です。

9 次のコマンドを使用して、ローカルファイルシステムから新しく作成されたコンテナに、環境設定ファイル (edirapi.conf) を /etc/opt/novell/eDirAPI/conf/edirapi.conf としてコピーします。

```
docker cp <absolute path of configuration file> identityconsole-container:/etc/opt/novell/eDirAPI/conf/edirapi.conf
```

次に例を示します。

```
docker cp /home/user/edirapi.conf identityconsole-container:/etc/opt/novell/eDirAPI/conf/edirapi.conf
```

10 次のコマンドを使用して、2 番目のコンテナを開始します。

```
docker start identityconsole-container
```

11 実行中のコンテナ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。

```
docker ps -a
```

スタンドアロン Identity Console(非 Docker) のアップグレード

このセクションでは、スタンドアロン Identity Console をアップグレードする手順について説明します。

1 Software License and Download (https://sld.microfocus.com/) から IdentityConsole_< バージョン >_Containers.tar.gz をダウンロードします。

2 SLD にログインし、[ソフトウェアダウンロード SLD] ページに移動し、[**[ダウンロード]**] をクリックします。

3 製品 : **[eDirectory]** > 製品名 : **[eDirectory per User Sub SW E-LTU]** > バージョン : **[9.2]** を選択し、移動します。

4 最新の Identity Console ビルドをダウンロードします。

5 次のコマンドを使って、ダウンロードしたファイルを抽出します。

```
tar -zxvf IdentityConsole_<version>_Linux.tar.gz
```

6 Identity Console のビルドを抽出したフォルダに移動します。

7 フォルダに接続する eDirectory ツリーのすべてのルート認証局証明書をコピーします。ルート認証局証明書をフォルダにコピーするには、次のコマンドを実行します。

```
cp /var/opt/novell/eDirectory/data/SSCert.pem <folder path>
```

次に例を示します。

```
cp /var/opt/novell/eDirectory/data/SSCert.pem /home/Identity_Console/
certs
```

8 次のコマンドを実行します。

```
./identityconsole_install
```

9 **手順 4** で使用するルート認証局証明書フォルダパスを指定します。

10 Identity Console は正常にアップグレードされます。

OSP コンテナのアップグレード

以下の手順を実行して、OSP コンテナをアップグレードします。

1 Software License and Download (https://sld.microfocus.com/) から OSP イメージの最新バージョンをダウンロードしてロードします。

次に例を示します。

```
docker load --input osp.tar.gz
```

2 最新の OSP イメージがロードされた後、次のコマンドを使用して現在の OSP コンテナを停止します。

```
docker stop <OSP container name>
```

3 (オプション) 共有ボリュームのバックアップを取ります。

4 次のコマンドを実行して、既存の OSP コンテナを削除します。

```
docker rm <OSP container name>
```

次に例を示します。

```
docker rm OSP_Container
```

5 キーストア (tomcat.ks) とサイレントプロパティファイルを含むディレクトリに移動し、既存のキーストア (tomcat.ks) を削除して、既存の OSP フォルダを保持します。キーサイズが 2048 の新しいキーストア (tomcat.ks) を生成します。詳細については、Identity Console インストールガイドの「OSP コンテナの展開」セクションの**手順 4** を参照してください。

6 次のコマンドを使用してコンテナを展開します。

```
docker run -d --name OSP_Container --network=host -e
SILENT_INSTALL_FILE=/config/silent.properties -v /data:/config
osp:<version>
```

次に例を示します。

```
docker run -d --name OSP_Container --network=host -e
SILENT_INSTALL_FILE=/config/silent.properties -v /data:/config
osp:6.5.3
```

4 Identity Console のアンインストール

この章では、Identity Console をアンインストールするプロセスについて説明します。

- 45 ページの 「Docker 環境のアンインストール手順」
- 45 ページの 「スタンドアロンの Identity Console(非 Docker) のアンインストール手順」

Docker 環境のアンインストール手順

Identity Console Docker コンテナをアンインストールするには、次の手順を実行します。

1. Identity Console コンテナを停止します。

   ```
   docker stop <container-name>
   ```

2. 次のコマンドを実行して、Identity Console Docker コンテナを削除します。

   ```
   docker rm -f <container_name>
   ```

3. 次のコマンドを実行して、Docker イメージを削除します。

   ```
   docker rmi -f <docker_image_id>
   ```

4. Docker ボリュームを削除します。

   ```
   docker volume rm <docker-volume>
   ```

注 : ボリュームを削除すると、データもサーバから削除されます。

スタンドアロンの Identity Console(非 Docker) のアンインストール手順

スタンドアロンの Identity Console をアンインストールするには、次の手順を実行します。

1. Identity Console がインストールされているマシンの /usr/bin 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。

   ```
   ./identityconsoleUninstall
   ```

3. Identity Console が正常にアンインストールされます。

注 : eDirectory または他の NetIQ 製品がマシンにインストールされている場合、ユーザは *nici* および *openssl* を手動でアンインストールする必要があります。