



Identity Console Installationshandbuch

September 2022

Rechtliche Hinweise

Informationen zu rechtlichen Hinweisen, Marken, Haftungsausschlüssen, Gewährleistungen, Ausführbeschränkungen und sonstigen Nutzungseinschränkungen, Rechten der US-Regierung, Patentrichtlinien und Erfüllung von FIPS finden Sie unter <https://www.netiq.com/company/legal>.

Copyright © 2022 NetIQ Corporation. Alle Rechte vorbehalten.

Inhalt

Info zu diesem Handbuch und zur Bibliothek	5
Info zu NetIQ Corporation	7
1 Planen der Identity Console-Installation	9
Systemanforderungen und Voraussetzungen für die Docker-Installation	9
Systemanforderungen	9
Voraussetzungen	9
Umgebung einrichten	11
Systemanforderungen und Voraussetzungen für die eigenständige Installation (ohne Docker)	14
Systemanforderungen	14
(Optional) Voraussetzung für die OSP-Konfiguration	16
Systemanforderungen und Voraussetzungen zur Ausführung als Arbeitsstation	16
Systemanforderungen	16
Überprüfung der RPM-Signatur	17
2 Bereitstellen von Identity Console	19
Sicherheitsempfehlungen	19
Identity Console als Docker-Container bereitstellen	20
OSP-Container bereitstellen	20
Identity Console als Docker-Container bereitstellen	22
Verwendung mehrerer Bäume mit Identity Console als Docker	24
Identity Console als eigenständige Installation bereitstellen	24
Identity Console als eigenständige Installation bereitstellen (ohne Docker)	25
Mehrere Bäume mit eigenständiger Bereitstellung von Identity Console	26
Identity Console unter Windows als Arbeitsstation	27
Verwendung mehrerer Bäume mit Identity Console als Arbeitsstation	28
Identity Console stoppen und neu starten	28
Identity Console als Docker-Container stoppen und neu starten	28
Eigenständige Identity Console-Installation stoppen und neu starten	29
Identity Console-Bereitstellung auf einer Arbeitsstation schließen und neu starten	29
Datenpersistenz verwalten	30
Identity Console in Azure Kubernetes Services bereitstellen	30
Identity Console im AKS-Cluster bereitstellen	30
Serverzertifikat ändern	37
Serverzertifikat im Docker-Container ändern	37
Serverzertifikat in der eigenständigen Bereitstellung von Identity Console ändern	37
3 Aufrüsten von Identity Console	39
Identity Console als Docker-Container aufrüsten	39
Eigenständige Bereitstellung von Identity Console (ohne Docker) aufrüsten	41
OSP-Container aufrüsten	42

4 Identity Console deinstallieren	43
Deinstallationsverfahren für Docker-Umgebung	43
Deinstallationsverfahren für eigenständige Identity Console-Installation (ohne Docker)	43

Info zu diesem Handbuch und zur Bibliothek

Das *Identity Console-Installationshandbuch* beschreibt die Installation und Verwaltung von NetIQ Identity Console (Identity Console). In diesem Buch wird die Terminologie definiert, und es werden Implementierungsszenarien vorgestellt.

Zielgruppe

Dieses Handbuch richtet sich an Netzwerkadministratoren.

Weitere Informationen in der Bibliothek

Die Bibliothek enthält folgende Informationsressourcen:

Installationsanleitung

Beschreibt die Installation und Aufrüstung von Identity Console. Dieses Handbuch richtet sich an Netzwerkadministratoren.

Info zu NetIQ Corporation

NetIQ ist ein globaler Hersteller von Unternehmenssoftware. Unser Fokus liegt auf drei besonderen Herausforderungen, die Sie in Ihrer Umgebung meistern müssen: Änderungen, Komplexität und Risiken. Unser Ziel ist es, Sie dabei zu unterstützen.

Unser Standpunkt

Sich an Änderungen anzupassen und Komplexität und Risiken zu beherrschen ist nichts Neues

Unter den verschiedenen Herausforderungen, denen Sie gegenüberstehen, beeinflussen diese drei Punkte sicherlich am meisten Ihre Möglichkeiten, Ihre physischen, virtuellen und Cloud-Umgebungen sicher zu messen, zu überwachen und zu verwalten.

Kritische Geschäftsservices schneller und besser bereitstellen

Wir sind davon überzeugt, dass IT-Organisationen über eine möglichst umfassende Kontrolle verfügen müssen, um eine zeitgerechte und kostenwirksame Servicebereitstellung zu ermöglichen. Der von Änderungen und Komplexität ausgehende, kontinuierliche Druck steigt ständig, weil sich die Unternehmen ständig ändern und die erforderlichen Technologien zur Verwaltung der Änderungen immer komplexer werden.

Unsere Philosophie

Intelligente Lösungen entwickeln, nicht einfach Software

Um zuverlässige Lösungen für die Kontrolle anbieten zu können, stellen wir erst einmal sicher, dass wir die Szenarien, in dem Unternehmen wie das Ihre täglich arbeiten, gründlich verstehen. Nur so können wir praxistaugliche, intelligente IT-Lösungen entwickeln, die nachweisbar messbare Ergebnisse liefern. Und das ist für uns wesentlich bereichernder, als einfach eine Software zu verkaufen.

Ihr Erfolg ist unsere Leidenschaft

Ihr Erfolg ist der Wegweiser für unser Geschäft. Wir wissen, dass Sie von der Produktkonzeption bis hin zur Bereitstellung IT-Lösungen benötigen, die richtig funktionieren und nahtlos mit Ihren vorhandenen Investitionen integriert werden können. Sie benötigen fortlaufenden Support, Schulungen nach der Bereitstellung und jemanden, mit dem Sie unkompliziert arbeiten können. Ihr Erfolg ist auch unser Erfolg.

Unsere Lösungen

- ♦ Identitäts- und Zugriffsregelung
- ♦ Zugriffsverwaltung
- ♦ Sicherheitsverwaltung

- ♦ System- und Anwendungsverwaltung
- ♦ Workload-Management
- ♦ Serviceverwaltung

Anfragen an die Verkaufsunterstützung

Bei Fragen zu Produkten, Preisen und Funktionen wenden Sie sich an Ihren Händler vor Ort. Wenn dies nicht möglich ist, wenden Sie sich an unser Verkaufsunterstützungsteam.

Weltweit:	www.netiq.com/about_netiq/officelocations.asp
Vereinigte Staaten und Kanada:	1-888-323-6768
Email:	info@netiq.com
Website:	www.netiq.com

Kontakt zum technischen Support

Bei spezifischen Produktproblemen, wenden Sie sich an unseren technischen Support.

Weltweit:	www.netiq.com/support/contactinfo.asp
Nord- und Südamerika:	1-713-418-5555
Europa, Naher Osten und Afrika:	+353 (0) 91-782 677
Email:	support@netiq.com
Website:	www.netiq.com/support

Kontakt zum Dokumentationssupport

Wir möchten Ihnen stets eine nützliche, aussagekräftige Dokumentation an die Hand geben. Wenn Sie uns einen Verbesserungsvorschlag mitteilen möchten, nutzen Sie die Schaltfläche **Comment on this topic** (Ihr Kommentar zu diesem Thema), die unten auf jeder Seite der unter www.netiq.com/documentation veröffentlichten HTML-Versionen unserer Dokumentation verfügbar ist. Sie können Verbesserungsvorschläge auch per Email an Documentation-Feedback@netiq.com senden. Wir freuen uns auf Ihre Rückmeldung.

Kontakt zur Online-Benutzer-Community

Qmunity, die NetIQ-Online-Community, ist ein Netzwerk zur Zusammenarbeit mit anderen NetIQ-Benutzern und -Experten. Qmunity bietet Ihnen aktuellste Informationen, nützliche Links zu hilfreichen Ressourcen und Kontakt zu NetIQ-Experten, damit Sie über alle Voraussetzungen verfügen, um das meiste aus den IT-Investitionen zu holen, auf die Sie sich verlassen. Weitere Informationen hierzu finden Sie im Internet unter <http://community.netiq.com>.

1 Planen der Identity Console-Installation

Dieses Kapitel beschreibt die Systemanforderungen und Voraussetzungen zur Installation von Identity Console. Identity Console kann sowohl als Docker-Container als auch als eigenständige Anwendung ausgeführt werden. Beachten Sie die jeweiligen Abschnitte zu den Systemanforderungen und Voraussetzungen für den betreffenden Installationstyp.

HINWEIS: Identity Console unterstützt eDirectory 9.2.4 HF2, Identity Manager Engine 4.8.3 HF2 und deren jeweils höhere Versionen. Sie müssen Ihre eDirectory- und Identity Manager Engine-Instanzen aufrüsten, bevor Sie Identity Console verwenden können.

- ♦ „Systemanforderungen und Voraussetzungen für die Docker-Installation“, auf Seite 9
- ♦ „Systemanforderungen und Voraussetzungen für die eigenständige Installation (ohne Docker)“, auf Seite 14
- ♦ „Systemanforderungen und Voraussetzungen zur Ausführung als Arbeitsstation“, auf Seite 16
- ♦ „Überprüfung der RPM-Signatur“, auf Seite 17

Systemanforderungen und Voraussetzungen für die Docker-Installation

Dieser Abschnitt erläutert die Systemanforderungen und Voraussetzungen für die Installation von Identity Console als Docker-Container.

- ♦ „Systemanforderungen“, auf Seite 9
- ♦ „Voraussetzungen“, auf Seite 9
- ♦ „Umgebung einrichten“, auf Seite 11

Systemanforderungen

Identity Console kann als Docker-Container ausgeführt werden. Weitere Informationen zu den Systemanforderungen und unterstützten Plattformen für diesen Installationstyp von Identity Console finden Sie in der [Docker-Dokumentation](#).

Voraussetzungen

- Installieren Sie Docker 20.10.9-ce oder höher. Weitere Informationen zum Installieren von Docker finden Sie im Abschnitt [Docker Installation](#) (Installation von Docker) in der Docker-Dokumentation.
- Sie benötigen ein pkcs12-Serverzertifikat mit dem privaten Schlüssel, um den Datenaustausch zwischen dem Identity Console-Server und dem Backend-Server zu verschlüsseln/entschlüsseln. Dieses Serverzertifikat wird zur Absicherung der HTTP-Verbindung verwendet. Sie können von einer externen ZS generierte Serverzertifikate verwenden. Weitere Informationen finden Sie

unter [Creating Server Certificate Objects](#) (Serverzertifikatsobjekte erstellen). Das Serverzertifikat muss den alternativen Antragstellernamen mit IP-Adresse und DNS des Identity Console-Servers enthalten. Nachdem das Serverzertifikatsobjekt erstellt wurde, müssen Sie es im PFX-Format exportieren.

- ❑ Sie benötigen für alle Bäume ein Zertifizierungsstellenzertifikat im PEM-Format, um die Zertifizierungsstellensignatur der im vorherigen Schritt erhaltenen Serverzertifikate zu überprüfen. Dieses Zertifikat der Stammzertifizierungsstelle gewährleistet außerdem das Einrichten einer gesicherten LDAP-Kommunikation zwischen dem Client und dem Identity Console-Server. Sie können beispielsweise das eDirectory-Zertifizierungsstellenzertifikat (`SSCert.pem`) aus `/var/opt/novell/eDirectory/data/SSCert.pem` abrufen.
- ❑ (Optional) Mit One SSO Provider (OSP) können Sie die Single Sign-On-Authentifizierung für Ihre Benutzer am Identity Console-Portal aktivieren. Sie müssen OSP installieren, bevor Sie Identity Console installieren. Um OSP für Identity Console zu konfigurieren, befolgen Sie die Aufforderungen auf dem Bildschirm und geben Sie die erforderlichen Werte für die Konfigurationsparameter an. Weitere Informationen finden Sie in [„OSP-Container bereitstellen“](#), auf Seite 20. Um Identity Console bei einem vorhandenen OSP-Server zu registrieren, fügen Sie Folgendes manuell zur Datei `ism-configuration.properties` im Ordner `/opt/netiq/idm/apps/tomcat/conf/` hinzu:

```
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
```

HINWEIS: Mit OSP können Sie nur eine Verbindung zu einem einzigen eDirectory-Baum herstellen, da OSP nicht mehrere eDirectory-Bäume unterstützt.

- ❑ Stellen Sie sicher, dass ein gültiger DNS-Eintrag mit vollständigem Hostnamen für Ihren Hostcomputer in `/etc/hosts` vorhanden ist.
- ❑ Wenn Sie Identity Console im Edge-Browser verwenden möchten, müssen Sie die neueste Version von Microsoft Edge herunterladen, um die volle Funktionalität zu erhalten.

HINWEIS: Bei Verwendung von Identity Console in Mozilla Firefox kann die Fehlermeldung `Origin Mismatch` (Ursprungskonflikt) angezeigt werden. Führen Sie die folgenden Schritte aus, um das Problem zu beheben:

- 1 Aktualisieren Sie Firefox auf die neueste Version.
 - 2 Geben Sie in Firefox im URL-Feld `about:config` ein und drücken Sie die Eingabetaste.
 - 3 Suchen Sie nach „Origin“.
 - 4 Doppelklicken Sie auf `network.http.sendOriginHeader` und ändern Sie den Wert in 1.
-

Umgebung einrichten

Unter Umständen müssen Sie eine Konfigurationsdatei mit bestimmten Parametern erstellen. Wenn Sie Identity Console mit OSP konfigurieren möchten, müssen Sie die OSP-spezifischen Parameter in der Konfigurationsdatei angeben. Erstellen Sie beispielsweise die folgende Datei `edirapi.conf` mit OSP-Parametern:

HINWEIS: Sie müssen den eDirectory-Baumnamen im Feld `osp-redirect-url` angeben.

```
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"
```

Falls Sie Identity Console ohne OSP konfigurieren möchten, erstellen Sie eine Konfigurationsdatei ohne OSP-Parameter:

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

HINWEIS: Wenn Sie Identity Console mit mehreren eDirectory-Bäumen konfigurieren möchten, können Sie die Parameter „`ldapserver`“, „`ldapuser`“ und „`ldappassword`“ überspringen und die Konfigurationsdatei erstellen.

Tabelle 1-1 Beschreibung der Konfigurationsparameter in der Konfigurationsdatei

Konfigurationsparameter	Beschreibung
<code>listen</code>	Geben Sie den Port 9000 als Listener-Port im Container für den Identity Console-Server an.
<code>ldapserver</code>	Geben Sie die IP-Adresse des eDirectory-Hostservers und die Portnummer an.

Konfigurationsparameter	Beschreibung
ldapuser	Geben Sie den Benutzernamen des eDirectory-Benutzers an. Dieser Parameter wird als Berechtigungsnachweis zum Initiieren von LDAP-Aufrufen an eDirectory mithilfe der Proxyautorisierungssteuerung im Falle einer OSP-Anmeldung verwendet. Der LDAP-Benutzer muss über Supervisor-Rechte für den eDirectory-Baum verfügen.
ldappassword	Geben Sie das Passwort für den LDAP-Benutzer an.
pfxpassword	Geben Sie das Passwort für die pkcs12-Serverzertifikatdatei an.
ospmode	Legen Sie <code>true</code> (wahr) fest, um OSP mit Identity Console zu integrieren. Wenn Sie diesen Parameter auf <code>false</code> (falsch) festlegen, verwendet Identity Console die LDAP-Anmeldung.
osp-token-endpoint	Diese URL wird zum Abrufen bestimmter Attribute vom OSP-Server verwendet, um die Gültigkeit des Authentifizierungs-Tokens zu überprüfen..
osp-authorize-url	Diese URL wird vom Benutzer verwendet, um den Berechtigungsnachweis zum Abrufen eines Authentifizierungs-Tokens anzugeben..
osp-logout-url	Mit dieser URL wird die Sitzung zwischen dem Benutzer und dem OSP-Server beendet..
osp-redirect-url	Der OSP-Server leitet den Benutzer nach dem Gewähren des Authentifizierungs-Tokens zu dieser URL um.. HINWEIS: Stellen Sie sicher, dass Sie den eDirectory-Baumnamen beim Konfigurieren von Identity Console in Kleinbuchstaben angeben. Falls der Baumname nicht in Kleinbuchstaben angegeben wird, kann bei der Anmeldung beim Identity Console-Server ein Fehler auftreten.
osp-client-id	Geben Sie die OSP-Client-ID an, die zur Registrierung von Identity Console bei OSP verwendet wurde..
ospclientpass	Geben Sie das OSP-Client-Passwort an, das zur Registrierung von Identity Console bei OSP verwendet wurde..
ospcert	Geben Sie den Speicherort des ZS-Zertifikats des OSP-Servers an..
bcert	Geben Sie den Speicherort des ZS-Zertifikats von Identity Console an.

Konfigurationsparameter	Beschreibung
loglevel	Geben Sie an, welcher Protokollumfang in der Protokolldatei enthalten sein soll. Dieser Parameter kann auf „fatal“ (schwerwiegende Fehler), „error“ (Fehler), „warn“ (Warnungen) oder „info“ (Infos) gesetzt werden.
check-origin	Wenn dieser Parameter auf <code>true</code> (wahr) festgelegt wird, vergleicht der Identity Console-Server den Ursprungswert der Anforderungen. Die verfügbaren Optionen sind <code>true</code> (wahr) und <code>false</code> (falsch). Der Parameter <i>origin</i> (Ursprung) ist bei Verwendung der DNS-Konfiguration auch dann obligatorisch, wenn der Wert des Parameters <i>check-origin</i> auf <code>false</code> (falsch) gesetzt ist.
origin	Identity Console vergleicht den Ursprungswert von Anforderungen mit den in diesem Feld angegebenen Werten. HINWEIS: Ab Identity Console 1.4 ist dieser Parameter unabhängig vom Parameter <i>check-origin</i> und ist obligatorisch, wenn die DNS-Konfiguration verwendet wird.
maxclients	Maximale Anzahl an Clients, die gleichzeitig auf <code>IDConsole</code> zugreifen können. Alle zusätzlichen Clients über diese Anzahl hinaus müssen in der Warteschlange warten.

HINWEIS

- ♦ Sie sollten den Konfigurationsparameter `ospmode` nur verwenden, wenn Sie OSP mit Identity Console integrieren möchten.
 - ♦ Wenn Identity Applications (Identity Apps) in Ihrer Identity Manager-Einrichtung im Clustermodus konfiguriert ist, müssen Sie den DNS-Namen des Lastausgleichsers in den Feldern `osp-token-endpoint`, `osp-authorize-url` und `osp-logout-url` in der Konfigurationsdatei angeben. Falls Sie in diesen Feldern die OSP-Serverdetails angeben, kann die Anmeldung bei Identity Console nicht ausgeführt werden.
 - ♦ Wenn Identity Console mit derselben OSP-Instanz wie Identity Apps und Identity Reporting konfiguriert ist, wird der Single Sign-On-Authentifizierungsdienst wirksam, sobald Sie sich beim Identity Console-Portal anmelden.
 - ♦ Für Identity Console 1.4 und höhere Versionen sollte die HTTPS-URL von OSP mit Zertifikaten bestätigt werden, die einen 2048-Bit-Schlüssel enthalten.
 - ♦ Wenn Sie den Zugriff auf das Identity Console-Portal von anderen Domänen einschränken möchten, legen Sie den Parameter `samesitecookie` auf `strict` fest. Wenn Sie den Zugriff auf das Identity Console-Portal von anderen Domänen zulassen möchten, legen Sie den Parameter `samesitecookie` auf `lax` fest. Wenn der Parameter während der Konfiguration nicht angegeben wird, werden standardmäßig die Browsereinstellungen berücksichtigt.
-

Nachdem Sie die Konfigurationsdatei vorbereitet haben, fahren Sie mit der Bereitstellung des Containers fort. Weitere Informationen finden Sie im „[Identity Console als Docker-Container bereitstellen](#)“, auf Seite 20.

Systemanforderungen und Voraussetzungen für die eigenständige Installation (ohne Docker)

- ♦ „Systemanforderungen“, auf Seite 14
- ♦ „(Optional) Voraussetzung für die OSP-Konfiguration“, auf Seite 16

Systemanforderungen

Dieser Abschnitt erläutert die Systemanforderungen und Voraussetzungen für die eigenständige Installation von Identity Console.

Kategorie	Mindestanforderung
Prozessor	1,4 GHz, 64-Bit
Arbeitsspeicher	2 GB
Festplattenspeicher	200 MB unter Linux
Unterstützte Browser	<ul style="list-style-type: none">♦ Neueste Version von Microsoft Edge♦ Neueste Version von Google Chrome♦ Neueste Version von Mozilla Firefox <p>HINWEIS: Bei Verwendung von Identity Console in Mozilla Firefox kann die Fehlermeldung <code>Origin Mismatch</code> (Ursprungskonflikt) angezeigt werden. Führen Sie die folgenden Schritte aus, um das Problem zu beheben:</p> <ol style="list-style-type: none">1 Aktualisieren Sie Firefox auf die neueste Version.2 Geben Sie in Firefox im URL-Feld <code>about:config</code> ein und drücken Sie die Eingabetaste.3 Suchen Sie nach „Origin“.4 Doppelklicken Sie auf <code>network.http.SendOriginHeader</code> und ändern Sie den Wert in 1.

Kategorie	Mindestanforderung
Unterstützte Betriebssysteme	<ul style="list-style-type: none"> ♦ Zertifiziert: <ul style="list-style-type: none"> ♦ SUSE Linux Enterprise Server (SLES) 15 SP1, SP2 und SP3 ♦ SUSE Linux Enterprise Server (SLES) 12 SP1, SP2, SP3, SP4 und SP5 ♦ Red Hat Enterprise Linux (RHEL) 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, 8.4 und 8.5 ♦ OpenSUSE 15.1 und 15.2 ♦ Unterstützt: Unterstützt auf späteren Versionen von Support Packs der oben als zertifiziert aufgeführten Betriebssysteme.
Zertifikate	<ul style="list-style-type: none"> ♦ Sie benötigen ein pkcs12-Serverzertifikat mit dem privaten Schlüssel zum Verschlüsseln/ Entschlüsseln des Datenaustauschs zwischen dem Client und dem Identity Console-Server. Dieses Serverzertifikat wird zur Absicherung der HTTP-Verbindung verwendet. Sie können von einer externen ZS generierte Serverzertifikate verwenden. Weitere Informationen finden Sie unter Creating Server Certificate Objects (Serverzertifikatsobjekte erstellen). Das Serverzertifikat muss den alternativen Antragstellernamen mit IP-Adresse und DNS des Identity Console-Servers enthalten. Nachdem das Serverzertifikatsobjekt erstellt wurde, müssen Sie es im PFX-Format exportieren. ♦ Sie benötigen für alle Bäume ein Zertifizierungsstellenzertifikat im PEM-Format, um die Zertifizierungsstellensignatur der im vorherigen Schritt erhaltenen Serverzertifikate zu überprüfen. Dieses Zertifikat der Stammzertifizierungsstelle gewährleistet außerdem das Einrichten einer gesicherten LDAP-Kommunikation zwischen dem Client und dem Identity Console-Server. Sie können beispielsweise das eDirectory-Zertifizierungsstellenzertifikat (<code>SSCert.pem</code>) aus <code>/var/opt/novell/eDirectory/data/SSCert.pem</code> abrufen.

Wenn Sie bereit sind, fahren Sie mit der Installation von Identity Console fort. Weitere Informationen finden Sie unter „[Identity Console als eigenständige Installation bereitstellen](#)“, auf [Seite 24](#).

(Optional) Voraussetzung für die OSP-Konfiguration

Mit One SSO Provider (OSP) können Sie die Single Sign-on-Authentifizierung für Ihre Benutzer am Identity Console-Portal aktivieren. Sie müssen OSP installieren, bevor Sie Identity Console installieren. Um OSP für Identity Console zu konfigurieren, befolgen Sie die Aufforderungen auf dem Bildschirm und geben Sie die erforderlichen Werte für die Konfigurationsparameter an. Weitere Informationen finden Sie in „[OSP-Container bereitstellen](#)“, auf Seite 20. Um Identity Console bei einem vorhandenen OSP-Server zu registrieren, fügen Sie Folgendes manuell zur Datei `ism-configuration.properties` im Ordner `/opt/netiq/idm/apps/tomcat/conf/` hinzu:

```
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
```

HINWEIS

- ♦ Bei einer Erstinstallation von OSP geben Sie die Option `y` (ja) für **Configure OSP with eDir API** (OSP mit eDir-API konfigurieren) ein und befolgen Sie die Anweisungen auf dem Bildschirm, um Identity Console mit OSP zu registrieren.
 - ♦ Stellen Sie sicher, dass Sie den eDirectory-Baumnamen beim Konfigurieren von Identity Console in Kleinbuchstaben angeben. Falls der Baumname nicht in Kleinbuchstaben angegeben wird, kann bei der Anmeldung beim Identity Console-Server ein Fehler auftreten.
 - ♦ Mit OSP können Sie nur eine Verbindung zu einem einzigen eDirectory-Baum herstellen, da OSP nicht mehrere eDirectory-Bäume unterstützt.
-

Systemanforderungen und Voraussetzungen zur Ausführung als Arbeitsstation

- ♦ „[Systemanforderungen](#)“, auf Seite 16

Systemanforderungen

Dieser Abschnitt erläutert die Systemanforderungen und Voraussetzungen zum Ausführen von Identity Console als Arbeitsstation.

Kategorie	Mindestanforderung
Prozessor	1,5 GHz, 64-Bit
Arbeitsspeicher	2 GB

Kategorie	Mindestanforderung
Festplattenspeicher	1 GB unter Windows
Unterstützte Betriebssysteme	<ul style="list-style-type: none"> ◆ Zertifiziert: <ul style="list-style-type: none"> ◆ Windows Server 2016 ◆ Windows Server 2019 ◆ Windows Server 2022 ◆ Windows 10 ◆ Windows 11
Zertifikate	<ul style="list-style-type: none"> ◆ Sie müssen ein Serverzertifikat im PFX-Format abrufen, um Daten zwischen dem Identity Console-Client und dem REST-Server auszutauschen. Dieses Serverzertifikat muss immer „keys.pfx“ heißen. Weitere Informationen finden Sie unter Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/b1j4tpo3.html#b1j4u0cm) (Serverzertifikatsobjekte erstellen). ◆ Sie benötigen für alle Bäume ein Zertifizierungsstellenzertifikat im PEM-Format, um die Zertifizierungsstellensignatur der im vorherigen Schritt erhaltenen Serverzertifikate zu überprüfen. Dieses Zertifikat der Stammzertifizierungsstelle gewährleistet außerdem das Einrichten einer gesicherten LDAP-Kommunikation zwischen dem Client und dem Identity Console-Server. Sie können beispielsweise das eDirectory-Zertifizierungsstellenzertifikat für Linux (SSCert.pem) aus /var/opt/novell/eDirectory/data/SSCert.pem abrufen. Rufen Sie das eDirectory-Zertifizierungsstellenzertifikat SSSCert.pem für Windows aus <eDirectory-Installationsverzeichnis>\NetIQ\edirectory\DIBFiles\CertServ\SSCert.pem ab.

Wenn Sie bereit sind, fahren Sie mit der Bereitstellung von Identity Console fort. Weitere Informationen finden Sie unter „[Identity Console unter Windows als Arbeitsstation](#)“, auf Seite 27.

Überprüfung der RPM-Signatur

Führen Sie die folgenden Schritte aus, um die RPM-Signaturüberprüfung durchzuführen:

- 1 Navigieren Sie zum Ordner, in dem der Build extrahiert wurde.

Beispiel: <Speicherort der entpackten Identity Console-Dateien>/
IdentityConsole_150_Linux/license/MicroFocusGPGPackageSign.pub.

- 2 Führen Sie den folgenden Befehl aus, um den öffentlichen Schlüssel zu importieren:

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

- 3 (Optional) Führen Sie den folgenden Befehl aus, um die RPM-Signatur zu überprüfen: `rpm --checksig -v <RPM-Name>`

Beispiel:

```
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 Bereitstellen von Identity Console

Dieses Kapitel beschreibt den Vorgang zur Bereitstellung von Identity Console und verwandte Sicherheitsempfehlungen. Überprüfen Sie in Vorbereitung auf die Bereitstellung die Checkliste der Voraussetzungen und Systemanforderungen unter [Kapitel 1, „Planen der Identity Console-Installation“](#), auf Seite 9.

- ♦ „Sicherheitsempfehlungen“, auf Seite 19
- ♦ „Identity Console als Docker-Container bereitstellen“, auf Seite 20
- ♦ „Identity Console als eigenständige Installation bereitstellen“, auf Seite 24
- ♦ „Identity Console unter Windows als Arbeitsstation“, auf Seite 27
- ♦ „Identity Console stoppen und neu starten“, auf Seite 28
- ♦ „Datenpersistenz verwalten“, auf Seite 30
- ♦ „Identity Console in Azure Kubernetes Services bereitstellen“, auf Seite 30
- ♦ „Serverzertifikat ändern“, auf Seite 37

Sicherheitsempfehlungen

- ♦ Docker-Container haben standardmäßig keine Ressourcenbeschränkungen. Jeder Container verfügt daher über Zugriff auf alle CPU- und Arbeitsspeicherressourcen, die vom Kernel des Hosts bereitgestellt werden. Beschränken Sie die Menge der Ressourcen, die von einem Container verwendet werden kann, um zu verhindern, dass ein ausgeführter Container zu viele Ressourcen verbraucht und für andere ausgeführte Container dann nicht mehr genügend Ressourcen verfügbar sind.
 - ♦ Stellen Sie sicher, dass Sie ein Hardlimit für die vom Docker-Container verwendeten Arbeitsspeicherressourcen festlegen. Verwenden Sie hierzu die Flagge `--memory` im Befehl „docker run“.
 - ♦ Stellen Sie sicher, dass Sie die von einem ausgeführten Docker-Container verwendeten CPU-Ressourcen beschränken. Verwenden Sie hierzu die Flagge `--cpuset-cpus` im Befehl „docker run“.
- ♦ `--pids-limit` sollte auf 300 festgelegt werden, um die Anzahl der zu einem bestimmten Zeitpunkt im Container entstehenden Kernel-Threads zu begrenzen. Dies dient dem Verhindern von DoS-Angriffen.
- ♦ Legen Sie die Richtlinie zum Neustart von Containern nach einem Fehler im Befehl „docker run“ mit der Flagge `--restart` auf 5 fest.
- ♦ Verwenden Sie den Container nur, wenn nach dem Starten des Containers der Zustand **Healthy** (Fehlerfrei) angezeigt wird. Führen Sie den folgenden Befehl aus, um den Zustand des Containers zu überprüfen:

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

- ♦ Der Docker-Container wird immer als Nicht-Root-Benutzer (`nds`) gestartet. Aktivieren Sie als zusätzliche Sicherheitsmaßnahme die Neuordnung der Benutzer-Namespaces auf dem Daemon, um Rechtheausweitungsangriffe aus dem Container heraus zu verhindern. Weitere Informationen zur Neuordnung von Benutzer-Namespaces finden Sie unter [Isolate containers with a user namespace](#) (Container mit einem Benutzer-Namespace isolieren).

Identity Console als Docker-Container bereitstellen

Dieser Abschnitt enthält folgende Verfahren:

- ♦ „OSP-Container bereitstellen“, auf Seite 20
- ♦ „Identity Console als Docker-Container bereitstellen“, auf Seite 22
- ♦ „Verwendung mehrerer Bäume mit Identity Console als Docker“, auf Seite 24

OSP-Container bereitstellen

Führen Sie zur Bereitstellung des OSP-Containers die folgenden Schritte aus:

- 1 Melden Sie sich auf der Seite [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) an und navigieren Sie zur Seite für die Softwaredownloads.
- 2 Wählen Sie Folgendes aus:
 - ♦ Produkt: eDirectory
 - ♦ Produktname: eDirectory per User Sub SW E-LTU
 - ♦ Version: 9.2
- 3 Laden Sie die Datei `IdentityConsole_<Version>_Containers_tar.zip` herunter.
- 4 Extrahieren Sie die heruntergeladene Datei in einen Ordner.
- 5 Ändern Sie die Datei „`silent.properties`“ gemäß Ihrer Anforderung. Ein Beispiel einer `silent.properties`-Datei wird unten gezeigt:

```
# 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

```

HINWEIS: Um Platzprobleme bei der Verwendung der .properties-Datei für die Installation im Automatikmodus (DOS-Textdatei) zu vermeiden, müssen Sie die DOS-Textdatei mit dem Tool dos2unix in das UNIX-Format konvertieren. Führen Sie den folgenden Befehl aus, um die Textdatei von DOS-Zeileneenden in Unix-Zeileneenden zu konvertieren:

```
dos2unix <Dateiname>
```

Beispiel:

```
dos2unix beispieldatei
```

-
- 6 Generieren Sie mit iManager ein Serverzertifikat (`cert.der`) und importieren Sie es in den Keystore (`tomcat.ks`). Kopieren Sie die `silent.properties`-Datei und den Keystore (`tomcat.ks`) in ein beliebiges Verzeichnis, zum Beispiel `/data`. Führen Sie die folgenden Schritte aus, um ein Serverzertifikat zu erstellen und in den Keystore zu importieren:

- 6a Führen Sie den folgenden Befehl aus, um einen Keystore (`tomcat.ks`) zu erstellen. Generieren Sie den Schlüssel und stellen Sie sicher, dass der Eigenname oder vollständige Hostname des Computers die IP-Adresse ist.

```
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 Führen Sie den folgenden Befehl aus, um einen Zertifizierungsantrag zu erstellen. Beispiel: `cert.csr`.

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

- 6c Übergeben Sie diesen Zertifizierungsantrag `cert.csr` an iManager und rufen Sie das Serverzertifikat `osp.der` ab. Stellen Sie sicher, dass Sie den Schlüsseltyp als „Benutzerdefiniert“ auswählen und die Schlüsselverwendungsoptionen auf

Datenverschlüsselung, Schlüsselverschlüsselung und digitale Signatur festlegen und dass im Feld für den/die alternativen Antragstellernamen des Zertifikats die IP-Adresse oder der Hostname des OSP-Servers enthalten sind. Weitere Informationen finden Sie unter [Creating a Server Certificate Object](#) (Erstellen eines Serverzertifikatsobjekts).

- 6d** Führen Sie die folgenden Befehle aus, um das ZS-Zertifikat (`SSCert.der`) und das Serverzertifikat (`cert.der`) in den Keystore `tomcat.ks` zu importieren.

```
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** Führen Sie den folgenden Befehl aus, um das OSP-Image zu laden:

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

- 8** Stellen Sie den Container mit dem folgenden Befehl bereit:

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

Beispiel:

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

Identity Console als Docker-Container bereitstellen

Dieses Kapitel beschreibt die Prozedur zum Bereitstellen von Identity Console als Docker-Container:

HINWEIS: Die in diesem Verfahren genannten Konfigurationsparameter, Beispielwerte und Beispiele sind nur als Referenz gedacht. Verwenden Sie diese Werte nicht direkt in Ihrer Produktionsumgebung.

- 1** Melden Sie sich auf der Seite [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) an und navigieren Sie zur Seite für die Softwaredownloads.
- 2** Wählen Sie Folgendes aus:
 - ♦ Produkt: eDirectory
 - ♦ Produktname: eDirectory per User Sub SW E-LTU
 - ♦ Version: 9.2
- 3** Laden Sie die Datei `IdentityConsole_<Version>_Container.tar.zip` herunter.
- 4** Das Image muss in die lokale Docker-Registrierung geladen werden. Extrahieren und laden Sie die Datei `IdentityConsole_<Version>_Containers.tar.gz` mit den folgenden Befehlen:

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

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

5 Erstellen Sie mit folgendem Befehl den Identity Console-Docker-Container:

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

Beispiel:

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

HINWEIS

- ♦ Setzen Sie die Umgebungsvariable `ACCEPT_EULA` auf „Y“, um die Endbenutzer-Lizenzvereinbarung zu akzeptieren. Sie können die Endbenutzer-Lizenzvereinbarung auch über die Aufforderung auf dem Bildschirm beim Starten des Containers akzeptieren, indem Sie im Docker-Befehl „create“ die Option `-it` für den interaktiven Modus verwenden.
- ♦ Der Parameter `--volume` im obigen Befehl erstellt ein Volume zum Speichern von Konfigurations- und Protokoll Daten. In diesem Beispiel wird ein Volume mit dem Namen `IDConsole-volume` erstellt.

6 Kopieren Sie die Serverzertifikatdatei mit folgendem Befehl vom lokalen Dateisystem als `/etc/opt/novell/eDirAPI/cert/keys.pfx` zum Container. Weitere Informationen zum Erstellen des Serverzertifikats finden Sie unter „[Voraussetzungen](#)“, auf Seite 9:

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

Beispiel:

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

Wenn Sie eine Verbindung zu mehreren eDirectory-Bäumen herstellen, müssen Sie sicherstellen, dass mindestens ein `keys.pfx`-Serverzertifikat für alle verbundenen Bäume abgerufen wird.

7 Kopieren Sie die ZS-Zertifikatdatei (`.pem`) mit folgendem Befehl vom lokalen Dateisystem als `/etc/opt/novell/eDirAPI/cert/SSCert.pem` zum Container. Weitere Informationen zum Erhalt des Zertifizierungsstellenzertifikats finden Sie unter „[Voraussetzungen](#)“, auf Seite 9:

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

Beispiel:

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

Wenn der Benutzer eine Verbindung zu mehreren eDirectory-Bäumen herstellen muss, lesen Sie den Abschnitt: „[Verwendung mehrerer Bäume mit Identity Console als Docker](#)“, auf Seite 24

8 Ändern Sie die Konfigurationsdatei gemäß Ihren Anforderungen und kopieren Sie die Konfigurationsdatei (`edirapi.conf`) mit dem folgenden Befehl aus Ihrem lokalen Dateisystem als `/etc/opt/novell/eDirAPI/conf/edirapi.conf` in den Container:

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

Beispiel:

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

9 Starten Sie den Docker-Container mit folgendem Befehl:

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

Beispiel:

```
docker start identityconsole-container
```

HINWEIS: Im Verzeichnis `/var/lib/docker/volumes/<Volumename>/_data/eDirAPI/var/log` finden Sie die folgenden Protokolldateien:

- ♦ `edirapi.log` – Diese Datei wird zur Protokollierung verschiedener Ereignisse in edirapi und von Problemen bei der Fehlersuche verwendet.
 - ♦ `edirapi_audit.log` – Diese Datei wird für die Protokollierung von Revisionsereignissen von edirapi verwendet. Die Protokolle folgen dem CEF-Revisionsformat.
 - ♦ `container-startup.log` – Diese Datei wird zum Erfassen von Installationsprotokollen des Docker-Containers von Identity Console verwendet.
-

Verwendung mehrerer Bäume mit Identity Console als Docker

Identity Console bietet dem Benutzer die Möglichkeit, eine Verbindung zu mehreren Bäumen herzustellen, indem für jeden Baum ein eigenes ZS-Zertifikat abgerufen wird.

Wenn Sie beispielsweise eine Verbindung zu drei eDirectory-Bäumen herstellen, müssen Sie alle drei ZS-Zertifikate in den Docker-Container kopieren:

```
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/SSCert1.pem
```

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

Führen Sie die folgenden Befehle aus, um Identity Console neu zu starten:

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

Identity Console als eigenständige Installation bereitstellen

- ♦ [„Identity Console als eigenständige Installation bereitstellen \(ohne Docker\)“](#), auf Seite 25
- ♦ [„Mehrere Bäume mit eigenständiger Bereitstellung von Identity Console“](#), auf Seite 26

Identity Console als eigenständige Installation bereitstellen (ohne Docker)

Dieser Abschnitt erläutert das Verfahren zur Bereitstellung von Identity Console als eigenständige Installation:

- 1 Melden Sie sich auf der Seite [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) an und navigieren Sie zur Seite für die Softwaredownloads.
- 2 Wählen Sie Folgendes aus:
 - ♦ Produkt: eDirectory
 - ♦ Produktname: eDirectory per User Sub SW E-LTU
 - ♦ Version: 9.2
- 3 Laden Sie den neuesten Identity Console-Build herunter.
- 4 Extrahieren Sie die heruntergeladene Datei in einem Ordner.
- 5 Öffnen Sie eine Shell und wechseln Sie zum Ordner, in den Sie den Identity Console-Build extrahiert haben.
- 6 Führen Sie den folgenden Befehl aus, während Sie als root-Benutzer oder gleichwertiger Benutzer angemeldet sind:

```
./identityconsole_install
```
- 7 Lesen Sie die Einführung und klicken Sie dann auf **ENTER**.
- 8 Klicken Sie auf „Y“, um die Lizenzvereinbarung zu akzeptieren. Dadurch werden alle erforderlichen RPMs auf Ihrem System installiert.
- 9 Geben Sie den Hostnamen (FQDN) bzw. die IP-Adresse des Identity Console-Servers ein.
- 10 Geben Sie die Portnummer ein, die Identity Console überwachen soll. Der Standardwert ist 9000.
- 11 Geben Sie die Option zur Integration von OSP in Identity Console oder zur Verwendung der LDAP-Anmeldung in Identity Console ein.
- 12 Wenn Sie OSP in Identity Console integrieren möchten:
 1. Geben Sie den Domännennamen/die IP-Adresse des eDirectory-Servers/ Identitätsdepotservers zusammen mit der LDAPS-Portnummer ein.
Beispiel:
192.168.1.1:636
 2. Geben Sie den Benutzernamen für eDirectory bzw. das Identitätsdepot ein.
Beispiel:
cn=admin,ou=org_unit,o=org
 3. Geben Sie das Passwort für eDirectory bzw. das Identitätsdepot ein.
 4. Geben Sie das Passwort für eDirectory bzw. das Identitätsdepot erneut ein, um es zu bestätigen.
 5. Geben Sie den Domännennamen bzw. die IP-Adresse des OSP-Servers mit der SSL-Portnummer für den SSO-Server ein.
 6. Geben Sie die OSP-Client-ID ein.

7. Geben Sie das OSP-Client-Passwort ein.
8. Geben Sie den Namen des eDirectory-Servers/Identitätsdepotbaums ein.
- 13 Geben Sie den Pfad für das vertrauenswürdige Stammzertifikat (`SSCert.pem`) einschließlich Ordnername ein.

Beispiel:

```
/home/Identity_Console/certs
```

HINWEIS: Der Benutzer muss sicherstellen, dass im Zertifikatordner kein Unterverzeichnis erstellt wird.

- 14 Geben Sie den Pfad des Serverzertifikats (`keys.pfx`) einschließlich Dateiname ein.

Beispiel:

```
/home/Identity_Console/keys.pfx
```

- 15 Geben Sie das Passwort für das Serverzertifikat ein. Um zu bestätigen, dass Sie das Passwort korrekt eingegeben haben, geben Sie das Passwort für das Serverzertifikat erneut ein. Die Installation wird initiiert.

HINWEIS: Im Verzeichnis `/var/opt/novell/eDirAPI/log` finden Sie die folgenden Protokolldateien:

- ♦ `edirapi.log` – Diese Datei wird zur Protokollierung verschiedener Ereignisse in `edirapi` und von Problemen bei der Fehlersuche verwendet.
- ♦ `edirapi_audit.log` – Diese Datei wird für die Protokollierung von Revisionsereignissen von `edirapi` verwendet. Die Protokolle folgen dem CEF-Revisionsformat.
- ♦ `identityconsole_install.log` – Diese Datei wird zum Erfassen von Installationsprotokollen von Identity Console verwendet.

Die Protokolle für das Starten/Stoppen des Identity Console-Prozesses sind in der Datei `/var/log/messages` enthalten.

HINWEIS: NetIQ empfiehlt, dass bei der Installation von Identity Console und eDirectory auf demselben Computer mindestens eine Instanz von eDirectory verfügbar ist.

Mehrere Bäume mit eigenständiger Bereitstellung von Identity Console

Wenn Sie eine Verbindung zu mehreren eDirectory-Bäumen herstellen, müssen Sie sicherstellen, dass Sie für jeden Baum ein eigenes ZS-Zertifikat erhalten.

Wenn Sie beispielsweise eine Verbindung zu drei eDirectory-Bäumen herstellen, müssen Sie alle drei ZS-Zertifikate in das Verzeichnis `etc/opt/novell/eDirAPI/cert/` kopieren:

```
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
```

Führen Sie einen der folgenden Befehle aus, um Identity Console neu zu starten:

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

Alternativ:

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

Identity Console unter Windows als Arbeitsstation

Identity Console kann unter Windows als Arbeitsstation gestartet werden und erfordert die Ausführung der REST-Dienste. Daher wird beim Start ein eDirAPI-Prozess in der edirapi.exe-Eingabeaufforderung ausgeführt. Wenn dieses edirapi.exe-Terminal geschlossen wird, ist Identity Console nicht mehr funktionsfähig.

Die folgende Prozedur beschreibt, wie Identity Console unter Windows ausgeführt wird.

- 1 Melden Sie sich auf der Seite [Software License and Download \(https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0\)](https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0) an und navigieren Sie zur Seite für die Softwaredownloads.
 - 2 Wählen Sie Folgendes aus:
 - ♦ Produkt: eDirectory
 - ♦ Produktname: eDirectory per User Sub SW E-LTU
 - ♦ Version: 9.2
 - 3 Laden Sie die Datei IdentityConsole_<Version>_workstation_win_x86_64.zip herunter.
 - 4 Extrahieren Sie die heruntergeladene Datei IdentityConsole_<Version>_workstation_win_x86_64.zip in einen Ordner.
 - 5 Navigieren Sie zum extrahierten Ordner IdentityConsole_150_workstation_win_x86_64\edirAPI\cert und kopieren Sie das vertrauenswürdige Stammzertifikat `SSCert.pem` und das Serverzertifikat `keys.pfx`.
Informationen zum Abrufen der Zertifikate finden Sie im folgenden Abschnitt: [„Systemanforderungen und Voraussetzungen zur Ausführung als Arbeitsstation“](#), auf Seite 16
Wenn der Benutzer eine Verbindung zu mehreren eDirectory-Bäumen herstellen muss, lesen Sie den Abschnitt: [„Verwendung mehrerer Bäume mit Identity Console als Arbeitsstation“](#), auf Seite 28
-
- HINWEIS:** Der Name des Serverzertifikats muss immer `keys.pfx` lauten.
-
- 6 Navigieren Sie zum Ordner, in dem der Build extrahiert wurde, und doppelklicken Sie auf die Datei `run.bat` (Windows-Batchdatei).
 - 7 Geben Sie das Passwort für das Serverzertifikat (`keys.pfx`) in die Eingabeaufforderung ein. Das eDirAPI-Prozessterminal (`edirapi.exe`) wird gestartet und die Identity Console-Anmeldeseite wird angezeigt.

HINWEIS:

- ♦ Wenn das eDirAPI-Prozessterminal (edirapi.exe) bereits ausgeführt wird, führen Sie `identityconsole.exe` aus dem Ordner mit dem extrahierten Build aus.
 - ♦ Benutzer finden die folgenden Protokolle unter:
`\IdentityConsole_150_workstation_win_x86_64\edirapi\log`
`edirapi.log` – Diese Datei wird zur Protokollierung verschiedener Ereignisse in `edirapi` und von Problemen bei der Fehlersuche verwendet.
`edirapi_audit.log` – Diese Datei wird für die Protokollierung von Revisionsereignissen von `edirapi` verwendet. Die Protokolle folgen dem CEF-Revisionsformat.
 - ♦ Die OSP-basierte Anmeldung wird im Arbeitsstationsmodus nicht unterstützt.
 - ♦ Identity Console als Bereitstellung auf einer Arbeitsstation überwacht nur den Port 9000. Ändern Sie die Datei `edirapi_win.conf` nicht.
-

Verwendung mehrerer Bäume mit Identity Console als Arbeitsstation

Identity Console bietet dem Benutzer die Möglichkeit, eine Verbindung zu mehreren Bäumen herzustellen, indem für jeden Baum ein eigenes ZS-Zertifikat abgerufen wird.

- 1 Schließen Sie die Identity Console-Arbeitsstation und das eDirAPI-Terminal.
- 2 Kopieren Sie die ZS-Zertifikate `SSCert.pem` an den Speicherort `IdentityConsole_150_workstation_win_x86_64\edirapi\cert`.
Wenn Sie beispielsweise eine Verbindung zu drei eDirectory-Bäumen herstellen möchten, kopieren Sie die ZS-Zertifikate als `SSCert1.pem`, `SSCert2.pem` und `SSCert3.pem`.
- 3 Navigieren Sie zum Ordner, in dem der Build extrahiert wurde, und doppelklicken Sie auf die Datei `run.bat` (Windows-Batchdatei).
- 4 Geben Sie das Passwort für `keys.pfx` in die Terminal-Eingabeaufforderung ein und melden Sie sich am gewünschten eDirectory-Baum an.

Identity Console stoppen und neu starten

- ♦ [„Identity Console als Docker-Container stoppen und neu starten“](#), auf Seite 28
- ♦ [„Eigenständige Identity Console-Installation stoppen und neu starten“](#), auf Seite 29
- ♦ [„Identity Console-Bereitstellung auf einer Arbeitsstation schließen und neu starten“](#), auf Seite 29

Identity Console als Docker-Container stoppen und neu starten

Führen Sie den folgenden Befehl aus, um Identity Console zu stoppen:

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

Führen Sie den folgenden Befehl aus, um Identity Console neu zu starten:

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

Führen Sie den folgenden Befehl aus, um Identity Console zu starten:

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

Eigenständige Identity Console-Installation stoppen und neu starten

Führen Sie einen der folgenden Befehle aus, um Identity Console zu stoppen:

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

Alternativ:

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

Führen Sie einen der folgenden Befehle aus, um Identity Console neu zu starten:

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

Alternativ:

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

Führen Sie einen der folgenden Befehle aus, um Identity Console zu starten:

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

Alternativ:

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

Identity Console-Bereitstellung auf einer Arbeitsstation schließen und neu starten

Gehen Sie wie folgt vor, um die Anwendung und den Prozess zu schließen:

- 1 Schließen Sie die Identity Console-Windows-Desktopanwendung.
- 2 Stoppen Sie den eDirAPI-Prozess, indem Sie das eDirAPI-Prozessterminal schließen.

Um die Identity Console-Bereitstellung auf einer Arbeitsstation neu zu starten, navigieren Sie zum Ordner, in den der Build extrahiert wurde, und doppelklicken Sie auf die Datei `run.bat` (Windows-Batchdatei).

HINWEIS: Wenn das eDirAPI-Prozessterminal bereits ausgeführt wird, führen Sie `identityconsole.exe` aus dem Ordner mit dem extrahierten Build aus, um die Identity Console-Bereitstellung auf der Arbeitsstation neu zu starten.

Datenpersistenz verwalten

Zusammen mit den Identity Console-Containern werden auch Volumes für die Datenpersistenz erstellt. Führen Sie die folgenden Schritte aus, um die Konfigurationsparameter eines alten Containers zu übernehmen, der diese Volumes verwendet:

- 1 Stoppen Sie den aktuellen Docker-Container mit dem folgenden Befehl:

```
docker stop identityconsole-container
```

- 2 Erstellen Sie den zweiten Container mit den Anwendungsdaten des alten Containers, die im Docker-Volume (`edirapi-volume-1`) gespeichert sind:

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

- 3 Starten Sie den zweiten Container mit folgendem Befehl:

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

- 4 (Optional) Nun kann der erste Container mit dem folgenden Befehl entfernt werden:

```
docker rm identityconsole-container
```

Identity Console in Azure Kubernetes Services bereitstellen

Azure Kubernetes Service (AKS) ist ein verwalteter Kubernetes-Dienst, mit dem Sie Cluster bereitstellen und verwalten können. Dieser Abschnitt beschreibt die folgenden Prozeduren:

Identity Console im AKS-Cluster bereitstellen

In diesem Abschnitt werden die folgenden Prozeduren zum Bereitstellen von Identity Console in einem AKS-Cluster erläutert:

- ♦ „[Azure Container Registry \(ACR\) erstellen](#)“, auf Seite 30
- ♦ „[Kubernetes-Cluster festlegen](#)“, auf Seite 32
- ♦ „[Öffentliche IP-Adresse mit Standard-SKU erstellen](#)“, auf Seite 32
- ♦ „[Cloud Shell einrichten und Verbindung mit dem Kubernetes-Cluster herstellen](#)“, auf Seite 32
- ♦ „[Anwendung bereitstellen](#)“, auf Seite 33

Azure Container Registry (ACR) erstellen

Azure Container Registry (ACR) ist eine Azure-basierte, private Registrierung für Docker-Container-Images.

Ausführlichere Schritte finden Sie im Abschnitt [Create an Azure container registry using the Azure portal](#) (Azure-Containerregistrierung mit dem Azure-Portal erstellen) unter „Create container registry – Portal“ (Containerregistrierung erstellen – Portal). Führen Sie ansonsten die folgenden Schritte aus, um eine Azure-Containerregistrierung (ACR) zu erstellen:

1. Melden Sie sich beim [Azure-Portal](#) an.

2. Wechseln Sie zu **Create a resource > Containers > Container Registry** (Ressource erstellen > Container > Containerregistrierung).
3. Geben Sie auf der Registerkarte **Basics** (Basiseinstellungen) Werte für **Resource group** (Ressourcengruppe) und **Registry name** (Registrierungsname) an. Der Registrierungsname muss in Azure eindeutig sein und mindestens 5 und maximal 50 alphanumerische Zeichen enthalten. Übernehmen Sie die Standardwerte für die verbleibenden Einstellungen.
4. Klicken Sie auf **Review + create** (Überprüfen und erstellen).
5. Klicken Sie auf **Create** (Erstellen).
6. Melden Sie sich bei der Azure-Befehlszeilenschnittstelle an und führen Sie den folgenden Befehl aus, um sich bei der Azure-Containerregistrierung anzumelden:


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

Beispiel:

```
az acr login --name < idconsole >
```
7. Rufen Sie den Anmeldeserver der Azure-Containerregistrierung mit dem folgenden Befehl ab:


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

Beispiel:

```
az acr show --name < idconsole > --query loginServer --output table
```
8. Versehen Sie das lokale Image von Identity Console mit dem folgenden Befehl mit einem Tag mit dem Namen des ACR-Anmeldeservers (registryname.azureacr.io):


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

Beispiel:

```
docker tag identityconsole:<version> registryname.azurecr.io/identityconsole:<version>
```
9. Verschieben Sie das mit Tag versehene Bild in die Registrierung.


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

Beispiel:

```
docker push registryname.azurecr.io/identityconsole:<version>
```
10. Rufen Sie die Liste der Images in der Registrierung mit dem folgenden Befehl ab:


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

Kubernetes-Cluster festlegen

Erstellen Sie eine Kubernetes-Service-Ressource mithilfe des Azure-Portals oder der Befehlszeilenschnittstelle.

Ausführlichere Schritte zum Erstellen einer Kubernetes-Service-Ressource in Azure mit einem Knoten finden Sie unter [Create an AKS Cluster](#) (AKS-Cluster erstellen) in der Schnellanleitung [Azure Quick Start](#).

HINWEIS:

- ♦ Stellen Sie sicher, dass „Azure CNI“ als Netzwerk ausgewählt ist.
 - ♦ Wählen Sie das vorhandene virtuelle Netzwerk aus (in dem der eDirectory-Server im Teilnetz bereitgestellt ist).
 - ♦ Wählen Sie die vorhandene Containerregistrierung aus, in der das Identity Console-Image verfügbar ist.
-

Öffentliche IP-Adresse mit Standard-SKU erstellen

Eine öffentliche IP-Adressressource unter der Kubernetes-Clusterressourcengruppe fungiert als Lastausgleichs-IP für die Anwendung.

Ausführliche Anweisungen finden Sie unter [Create a public IP address using the Azure portal](#) (Erstellen einer öffentlichen IP-Adresse mithilfe des Azure-Portals) unter „Create public IP address – Portal“ (Erstellen einer öffentlichen IP-Adresse – Portal).

Cloud Shell einrichten und Verbindung mit dem Kubernetes-Cluster herstellen

Cloud Shell steht im Azure-Portal zur Verfügung und kann für alle Vorgänge verwendet werden.

Informationen zum Einrichten von Cloud Shell im Azure-Portal finden Sie im Abschnitt [Start Cloud Shell](#) (Cloud Shell starten) unter [Bash – Quick Start](#) (Bash – Kurzanleitung). Befolgen Sie alternativ die folgenden Schritte, um Cloud Shell einzurichten und eine Verbindung mit dem Kubernetes-Cluster herzustellen:

1. Klicken Sie im Azure-Portal auf die Schaltfläche , um Cloud Shell zu öffnen.

HINWEIS: Um einen Kubernetes-Cluster zu verwalten, verwenden Sie den Kubernetes-Befehlszeilenclient `kubectl`. `kubectl` ist bereits installiert, wenn Sie Azure Cloud Shell verwenden.

2. Konfigurieren Sie `kubectl` mit dem folgenden Befehl zum Herstellen einer Verbindung mit Ihrem Kubernetes-Cluster:

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

Beispiel:

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

3. Überprüfen Sie die Liste der Clusterknoten mit dem folgenden Befehl:

```
kubectl get nodes
```

Anwendung bereitstellen

Zum Bereitstellen von Identity Console können Sie die Beispieldateien `idc-services.yaml`, `idc-statefulset.yaml`, `idc-storageclass.yaml` und `idc-pvc.yaml` verwenden.

Sie können auch je nach Bedarf Ihre eigenen yaml-Dateien erstellen.

1. Erstellen Sie mit dem folgenden Befehl eine Speicherklassenressource:

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

Beispiel:

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

(Optional) Weitere Informationen zum dynamischen Erstellen und Verwenden eines persistenten Volumes mit einer Azure Files-Freigabe finden Sie unter [Dynamically create and use a persistent volume with Azure Files in Azure Kubernetes Service \(AKS\)](#) (Dynamisches Erstellen und Verwenden eines persistenten Volumes mit Azure Files in Azure Kubernetes Service (AKS)).

Im Folgenden wird ein Beispiel für eine Speicherklassenressourcendatei gezeigt:

```
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
~
```

Eine Speicherklassenressource ermöglicht die dynamische Speicherbereitstellung. Sie definiert, wie eine Azure Files-Freigabe erstellt werden soll.

2. Mit dem folgenden Befehl können Sie die Details der Speicherklasse anzeigen:

```
kubectl get sc
```

3. Erstellen Sie eine PVC-Ressource mit der Datei `idc-pvc.yaml`:

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


Beispiel:


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

Im Folgenden wird ein Beispiel für eine PVC-Ressourcendatei gezeigt:

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

Eine PVC-Ressource erstellt die Dateifreigabe. Eine Anforderung zur Bereitstellung eines persistenten Volumes (Persistent Volume Claim, PVC) verwendet das Speicherklassenobjekt, um dynamisch eine Azure Files-Freigabe bereitzustellen.

4. Laden Sie die Datei `edirapi.conf`, das ZS-Zertifikat und das Serverzertifikat zu Cloud Shell hoch.

Klicken Sie in Cloud Shell auf das Schaltflächensymbol **Dateien hoch-/herunterladen**  und laden Sie die Dateien `edirapi.conf`, `SSCert.pem` und `keys.pfx` hoch.

HINWEIS: Die Datei „edirapi.conf“ hat einen Parameter mit dem Namen „origin“. Hier muss die IP-Adresse angegeben werden, mit der auf die Identity Console-Anwendung zugegriffen wird. (Verwenden Sie die im Abschnitt „[Öffentliche IP-Adresse mit Standard-SKU erstellen](#)“, auf [Seite 32](#) erstellte IP-Adresse.)

Für die Bereitstellung von Identity Console ist ein Serverzertifikat (`keys.pfx`) erforderlich.

Stellen Sie beim Erstellen des Serverzertifikats sicher, dass Sie im alternativen Antragstellernamen einen gültigen DNS-Namen angeben.

Schritte zum Erstellen eines gültigen DNS-Namens:

Ein typischer Pod, der mit StatefulSet bereitgestellt wird, hat einen DNS-Namen, der sich wie folgt zusammensetzt: `{StatefulSet-Name}-{Ordinalzahl}.{Servicename}.{Namespace}.svc.cluster.local`

- ♦ Wenn der StatefulSet-Name in der Datei „idconsole-statefulset.yaml“ „idconsole-app“ lautet, dann ist StatefulSet-Name = idconsole-app.
- ♦ Wenn es sich um den ersten Pod handelt, ist „Ordinalzahl“ = 0.
- ♦ Wenn Sie „serviceName“ in der Datei „idconsole-statefulset.yaml“ als „idconsole“ definieren, ist „Servicename“= idconsole.
- ♦ Wenn der standardmäßige Namespace verwendet wird, ist „Namespace“=default.

Für dieses Beispiel lautet die Ausgabe für den Podnamen: `idconsole-app-0.idconsole.default.svc.cluster.local`

5. Erstellen Sie eine configmap-Ressource im Kubernetes-Cluster, in der die Konfigurationsdateien zusammen mit den Zertifikaten gespeichert werden.

Stellen Sie vor dem Ausführen des Befehls sicher, dass die Dateien (`edirapi.conf`, `SSCert.pem` und `keys.pfx`) im Verzeichnis vorhanden sind.

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

Beispiel:

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

6. Sie können die Details des configmap-Objekts mit dem Befehl „`kubectl describe`“ anzeigen:

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

Beispiel:

```
kubectl describe configmap config-data
```

7. Erstellen Sie eine StatefulSet-Ressource zum Bereitstellen des Containers.

Führen Sie den folgenden Befehl aus, um den Container bereitzustellen:

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

Beispiel:

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

Im Folgenden wird ein Beispiel für eine StatefulSet-Ressourcendatei gezeigt:

```
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. Führen Sie den folgenden Befehl aus, um den Status des bereitgestellten Pods zu überprüfen:

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

9. Erstellen Sie eine Service-Ressource vom Typ „loadBalancer“.

Der in der yml-Datei angegebene Servicetyp ist „loadBalancer“.

Erstellen Sie eine Service-Ressource mit dem folgenden Befehl:

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

Beispiel:

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

Im Folgenden wird ein Beispiel für eine Service-Ressourcendatei gezeigt:

```

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

```

Überprüfen Sie die mit dem folgenden Befehl die externe IP-Adresse (oder die loadBalancer-IP-Adresse):

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

10. Rufen Sie die URL mit der externen IP-Adresse (oder der loadBalancer-IP-Adresse) auf.

Beispiel:

```
https://<EXTERNE IP-ADRESSE>:9000/identityconsole
```

Serverzertifikat ändern

Dieser Abschnitt enthält Informationen zum Ändern des Serverzertifikats in einem Docker-Container und in einer eigenständigen Bereitstellung von Identity Console.

- ♦ „[Serverzertifikat im Docker-Container ändern](#)“, auf Seite 37
- ♦ „[Serverzertifikat in der eigenständigen Bereitstellung von Identity Console ändern](#)“, auf Seite 37

Serverzertifikat im Docker-Container ändern

Führen Sie die folgenden Schritte aus, um das Serverzertifikat im Docker-Container zu ändern:

- 1 Führen Sie den folgenden Befehl aus, um das neue Serverzertifikat an einen beliebigen Speicherort im Container zu kopieren.

Beispiel:

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

- 2 Melden Sie sich mit dem folgenden Befehl beim Container an:

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

- 3 Führen Sie NLP CERT aus, um die Schlüssel als Pseudobeneutzer zu speichern:

```
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 Beenden Sie die Containerkonsole mit dem folgenden Befehl:

```
exit
```

- 5 Starten Sie den Container neu, indem Sie Folgendes eingeben:

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

Serverzertifikat in der eigenständigen Bereitstellung von Identity Console ändern

Führen Sie die folgenden Schritte aus, um das Serverzertifikat im eigenständigen Container zu ändern:

- 1 Führen Sie NLP CERT aus, um die Schlüssel zu speichern:

```
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 Starten Sie Identity Console neu:

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

3 Aufrüsten von Identity Console

Dieses Kapitel beschreibt die Vorgehensweise zum Aufrüsten von Identity Console auf die neueste Version. Überprüfen Sie in Vorbereitung auf die Aufrüstung die Checkliste der Voraussetzungen und Systemanforderungen unter [Kapitel 1, „Planen der Identity Console-Installation“](#), auf Seite 9.

Dieser Abschnitt beschreibt die folgenden Prozeduren:

- ♦ „Identity Console als Docker-Container aufrüsten“, auf Seite 39
- ♦ „Eigenständige Bereitstellung von Identity Console (ohne Docker) aufrüsten“, auf Seite 41
- ♦ „OSP-Container aufrüsten“, auf Seite 42

Identity Console als Docker-Container aufrüsten

Wenn eine neue Version des Identity Console-Image verfügbar ist, kann der Administrator eine Aufrüstungsprozedur ausführen, um den Container mit der neuesten Version von Identity Console bereitzustellen. Stellen Sie vor dem Aufrüsten sicher, dass alle erforderlichen anwendungsbezogenen Daten permanent in Docker-Volumes gespeichert sind. Führen Sie die folgenden Schritte aus, um Identity Console mit einem Docker-Container aufzurüsten:

- 1 Laden Sie die neueste Version des Docker-Images von der Seite [Software License and Download \(https://sld.microfocus.com/\)](#) herunter und laden Sie das Image. Führen Sie dann die in „Bereitstellen von Identity Console“, auf Seite 19 beschriebenen Schritte aus, um die neueste Version von Identity Console zu installieren.

- 2 Nachdem das neueste Docker-Image geladen wurde, stoppen Sie den aktuellen Docker-Container mit dem folgenden Befehl:

```
docker stop identityconsole-container
```

- 3 (Optional) Erstellen Sie eine Sicherung des freigegebenen Volumes.

- 4 Löschen Sie den vorhandenen Identity Console-Container, indem Sie den folgenden Befehl ausführen:

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

Beispiel:

```
docker rm identityconsole-container
```

- 5 (Optional) Löschen Sie das veraltete Identity Console-Docker-Image, indem Sie den folgenden Befehl ausführen:

```
docker rmi identityconsole
```

- 6 Erstellen Sie mit folgendem Befehl den Identity Console-Docker-Container:

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

Beispiel:

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

HINWEIS

- ♦ Setzen Sie die Umgebungsvariable `ACCEPT_EULA` auf „Y“, um die Endbenutzer-Lizenzvereinbarung zu akzeptieren. Sie können die Endbenutzer-Lizenzvereinbarung auch über die Aufforderung auf dem Bildschirm beim Starten des Containers akzeptieren, indem Sie im Docker-Befehl „create“ die Option `-it` für den interaktiven Modus verwenden.
- ♦ Der Parameter `--volume` im obigen Befehl erstellt ein Volume zum Speichern von Konfigurations- und Protokoll Daten. In diesem Beispiel wird ein Volume mit dem Namen `IDConsole-volume` erstellt.

-
- 7** Kopieren Sie die Serverzertifikatdatei mit folgendem Befehl vom lokalen Dateisystem als `/etc/opt/novell/eDirAPI/cert/keys.pfx` zum neu erstellten Container:

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

Beispiel:

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

Wenn Sie eine Verbindung zu mehreren eDirectory-Bäumen hergestellt haben, müssen Sie mindestens ein `keys.pfx`-Serverzertifikat für alle verbundenen Bäume kopieren.

- 8** Kopieren Sie die ZS-Zertifikatdatei (`.pem`) mit folgendem Befehl vom lokalen Dateisystem als `/etc/opt/novell/eDirAPI/cert/SSCert.pem` zum neu erstellten Container:

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

Beispiel:

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

Wenn Sie eine Verbindung zu mehreren eDirectory-Bäumen herstellen, müssen Sie sicherstellen, dass Sie für jeden Baum ein eigenes ZS-Zertifikat erhalten. Wenn Sie beispielsweise eine Verbindung zu drei eDirectory-Bäumen herstellen, müssen Sie alle drei ZS-Zertifikate in den Docker-Container kopieren:

```
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
```

HINWEIS: Ab Identity Console 1.4 enthält die Konfigurationsdatei (`edirapi.conf`) nicht explizit die Parameter „`ldapuser`“, „`ldappassword`“ und „`ldapserver`“. Der Parameterwert „`bcert`“ muss den Verzeichnispfad für vertrauenswürdige Stammzertifikate enthalten. Beispiel: `bcert = "/etc/opt/novell/eDirAPI/cert/"`. Der Parameter „`origin`“ ist unabhängig vom Parameter „`check-origin`“ und bei Verwendung der DNS-Konfiguration obligatorisch.

- 9 Kopieren Sie die Konfigurationsdatei (`edirapi.conf`) mit folgendem Befehl vom lokalen Dateisystem als `/etc/opt/novell/eDirAPI/conf/edirapi.conf` zum neu erstellten Container:

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

Beispiel:

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

- 10 Starten Sie den zweiten Container mit folgendem Befehl:

```
docker start identityconsole-container
```

- 11 Führen Sie den folgenden Befehl aus, um den Status des ausgeführten Containers zu überprüfen:

```
docker ps -a
```

Eigenständige Bereitstellung von Identity Console (ohne Docker) aufrüsten

Dieser Abschnitt erläutert das Verfahren zur Aufrüstung von Identity Console als eigenständige Installation:

- 1 Laden Sie `IdentityConsole_<Version>_Containers.tar.gz` von der Seite [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) (SLD) herunter.
- 2 Melden Sie sich auf der SLD-Seite an, navigieren Sie zur Seite für Softwaredownloads und klicken Sie auf **Download**.
- 3 Wählen Sie durch Navigation folgendes Produkt aus: **eDirectory** > Produktname: **eDirectory per User Sub SW E-LTU** > Version: **9.2**.
- 4 Laden Sie den neuesten Identity Console-Build herunter.
- 5 Extrahieren Sie die heruntergeladene Datei mit folgendem Befehl:

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

- 6 Navigieren Sie zum Ordner, in den Sie den Identity Console-Build extrahiert haben.
- 7 Kopieren Sie alle vertrauenswürdigen Stammzertifikate der eDirectory-Bäume, mit denen Sie eine Verbindung herstellen möchten, in einen Ordner. Führen Sie den folgenden Befehl aus, um ein vertrauenswürdigen Stammzertifikat in einen Ordner zu kopieren:

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

Beispiel:

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

8 Führen Sie den folgenden Befehl aus:

```
./identityconsole_install
```

9 Geben Sie den in **Schritt 4** verwendeten Ordnerpfad für vertrauenswürdige Stammzertifikate an.

10 Identity Console wird erfolgreich aufgerüstet.

OSP-Container aufrüsten

Führen Sie zur Aufrüstung des OSP-Containers die folgenden Schritte aus:

1 Laden Sie die neueste Version des OSP-Images von der Seite [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) herunter und laden Sie sie.

Beispiel:

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

2 Nachdem das neueste OSP-Image geladen wurde, stoppen Sie den aktuellen OSP-Container mit dem folgenden Befehl:

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

3 (Optional) Erstellen Sie eine Sicherung des freigegebenen Volumes.

4 Löschen Sie den vorhandenen OSP-Container, indem Sie den folgenden Befehl ausführen:

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

Beispiel:

```
docker rm OSP_Container
```

5 Wechseln Sie in das Verzeichnis, das den Keystore (`tomcat.ks`) und die properties-Datei für Installationen im Automatikmodus enthält, löschen Sie den vorhandenen Keystore (`tomcat.ks`) und behalten Sie den vorhandenen OSP-Ordner bei. Generieren Sie einen neuen Keystore (`tomcat.ks`) mit der Schlüsselgröße 2048. Weitere Informationen finden Sie in **Schritt 4** im Abschnitt [OSP-Container bereitstellen](#) im [Identity Console-Installationshandbuch](#).

6 Stellen Sie den Container mit dem folgenden Befehl bereit:

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

Beispiel:

```
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 deinstallieren

Dieses Kapitel beschreibt den Prozess zum Deinstallieren von Identity Console:

- „Deinstallationsverfahren für Docker-Umgebung“, auf Seite 43
- „Deinstallationsverfahren für eigenständige Identity Console-Installation (ohne Docker)“, auf Seite 43

Deinstallationsverfahren für Docker-Umgebung

Führen Sie die folgenden Schritte aus, um den Identity Console-Docker-Container zu deinstallieren:

- 1 Beenden Sie den Identity Console-Container:

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

- 2 Führen Sie den folgenden Befehl aus, um den Identity Console-Docker-Container zu entfernen:

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

- 3 Führen Sie den folgenden Befehl aus, um das Docker-Image zu entfernen:

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

- 4 Entfernen Sie das Docker-Volume:

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

HINWEIS: Wenn Sie das Volume entfernen, werden auch die Daten von Ihrem Server entfernt.

Deinstallationsverfahren für eigenständige Identity Console-Installation (ohne Docker)

Führen Sie die folgenden Schritte aus, um eine eigenständige Identity Console-Installation zu deinstallieren:

- 1 Navigieren Sie zum Verzeichnis `/usr/bin` auf dem Computer, auf dem Identity Console installiert ist.

- 2 Führen Sie den folgenden Befehl aus:

```
./identityconsoleUninstall
```

- 3 Identity Console wird erfolgreich deinstalliert.

HINWEIS: Wenn eDirectory oder ein anderes NetIQ-Produkt auf dem Computer installiert ist, muss der Benutzer `nici` und `openssl` manuell deinstallieren.
