Identity Console
Guida all'installazione

Settembre 2022

Note legali

Sommario

Informazioni sulla guida e sulla libreria

La *Guida all'installazione di Identity Console* fornisce informazioni su come installare e gestire il prodotto NetIQ Identity Console (Identity Console). definisce la terminologia e include scenari di implementazione.

Destinatari

Questa guida è destinata agli amministratori di rete.

Altre informazioni incluse nella libreria

La libreria contiene le risorse seguenti:

Guida all'installazione

Descrive come installare ed eseguire l'upgrade di Identity Console. Questa guida è destinata agli amministratori di rete.

Informazioni su NetIQ Corporation

NetIQ Corporation è una società globale di software per le aziende, focalizzata su tre problematiche costanti dell'ambiente aziendale (cambiamento, complessità e rischio) e che offre soluzioni utili per gestirle.

Il nostro punto di vista

Adattamento al cambiamento e gestione della complessità: sfide ben note

Fra tutte le sfide da affrontare, queste sono forse le variabili principali che impediscono il controllo necessario a misurare, monitorare e gestire in modo sicuro gli ambienti informatici fisici, virtuali e cloud.

Erogazione migliore e più rapida dei servizi di business di importanza critica

Siamo convinti del fatto che le nostre soluzioni per assicurare il massimo controllo possibile alle organizzazioni IT siano il solo percorso possibile verso un'erogazione tempestiva ed economica dei servizi. Il costante processo di cambiamento delle organizzazioni e la maggiore complessità intrinseca delle tecnologie necessarie per gestirlo continueranno ad esercitare pressioni sempre più forti.

La nostra filosofia

Vendiamo soluzioni intelligenti e non semplice software

Al fine di poter garantire un controllo affidabile, prima di tutto ci dedichiamo a comprendere le situazioni reali in cui le aziende dell'IT operano quotidianamente. Questo approccio è il solo che consenta di sviluppare soluzioni IT pratiche e intelligenti, capaci di garantire risultati misurabili e comprovati, e che ci gratifica molto di più della semplice vendita di software.

Il successo degli utenti è il nostro stimolo

Il fulcro attorno al quale ruota la nostra attività aziendale è il successo degli utenti. Dall'ideazione del prodotto alla sua installazione, comprendiamo appieno l'esigenza dei nostri clienti di poter contare su soluzioni IT funzionanti che si integrino alla perfezione con i prodotti esistenti. Forniamo, inoltre, supporto costante e formazione successiva all'installazione, confermando di essere a tutti gli effetti un partner con cui è veramente facile collaborare. In sintesi: il successo degli utenti è il nostro successo.

Le nostre soluzioni

- Governance di identità e accessi
- Gestione degli accessi
- Gestione della sicurezza

- Gestione di sistemi e applicazioni
- Gestione del workload
- Gestione del servizio

Contattare l'assistenza alle vendite

Per informazioni su prodotti, prezzi e funzionalità, rivolgersi al partner locale. In caso d'impossibilità, contattare il team di assistenza alle vendite.

Sedi globali:	www.netiq.com/about_netiq/officelocations.asp
Stati Uniti e Canada:	1-888-323-6768
E-mail:	info@netiq.com
Sito Web:	www.netiq.com

Contattare il supporto tecnico

Per problemi specifici del prodotto, rivolgersi al team del supporto tecnico.

Sedi globali:	www.netiq.com/support/contactinfo.asp
Nord e Sud America:	1-713-418-5555
Europa, Medio Oriente e Africa:	+353 (0) 91-782 677
E-mail:	support@netiq.com
Sito Web:	www.netiq.com/support

Contattare il supporto per la documentazione

NetIQ desidera fornire tutta la documentazione necessaria per indicare le soluzioni più appropriate alle esigenze degli utenti. Per inviare suggerimenti di miglioramento, fare clic su **comments on this topic** (commenti su questo argomento) in fondo a qualsiasi pagina nelle versioni in HTML della documentazione pubblicata sul sito Web www.netiq.com/documentation. È inoltre possibile inviare un'e-mail a Documentation-Feedback@netiq.com. La collaborazione degli utenti è una fonte preziosa e saremo lieti di ricevere qualsiasi suggerimento.

Contattare la comunità online di utenti

La comunità online di NetIQ, Qmunity, è una rete collaborativa che unisce utenti ed esperti di NetIQ. È una risorsa che contribuisce ad acquisire la conoscenza approfondita necessaria a sfruttare a pieno il potenziale degli investimenti IT fondamentali, in quanto offre informazioni più immediate, collegamenti e risorse utili e accesso agli esperti di NetIQ. Per ulteriori informazioni, visitare il sito http://community.netiq.com.

1 Pianificazione dell'installazione di Identity Console

In questo capitolo vengono illustrati i requisiti di sistema e i prerequisiti per l'installazione di Identity Console. Poiché Identity Console può essere eseguito sia come container di Docker che come applicazione autonoma, fare riferimento alle rispettive sezioni relative ai requisiti di sistema e ai prerequisiti per entrambi i tipi di installazione.

Nota: Identity Console supporta eDirectory 9.2.4 HF2, Identity Manager Engine 4.8.3 HF2 e le rispettive versioni successive. È necessario eseguire l'upgrade delle istanze di eDirectory e di Identity Manager Engine prima di utilizzare Identity Console.

Requisiti di sistema e prerequisiti per l'installazione di Docker

In questa sezione vengono illustrati i requisiti di sistema e i prerequisiti per l'installazione di Identity Console come container di Docker.

Requisiti di sistema

Poiché Identity Console può essere eseguito come container di Docker, per ulteriori informazioni sui requisiti di sistema e sulle piattaforme supportate per l'installazione di Identity Console, vedere la Docker Documentation (Documentazione di Docker).

Prerequisiti

- ❒ Installare Docker 20.10.9-ce o versione successiva. Per ulteriori informazioni su come installare Docker, vedere Docker Installation (Installazione di Docker).
- ❒ È necessario ottenere un certificato server pkcs12 con la chiave privata per cifrare/decifrare lo scambio di dati tra il server di Identity Console e il server backend. Questo certificato server viene utilizzato per proteggere la connessione HTTP. È possibile utilizzare i certificati server generati da una CA esterna. Per ulteriori informazioni, vedere Creating Server Certificate

Objects (Creazione di oggetti Server Certificate). Il certificato server deve contenere il nome alternativo del soggetto con l'indirizzo IP e il DNS del server di Identity Console. Una volta creato l'oggetto Server Certificate, è necessario esportarlo in formato .pfx.

- ❒ È necessario ottenere un certificato CA per tutti gli alberi in formato `.pem` per convalidare la firma CA dei certificati server ottenuti nel passaggio precedente. Questo certificato CA radice garantisce inoltre una comunicazione LDAP protetta tra il client e il server di Identity Console. Ad esempio, è possibile ottenere il certificato CA di eDirectory (`SSCert.pem`) da `/var/opt/novell/eDirectory/data/SSCert.pem`.
- ❒ (Facoltativo) Utilizzando One SSO Provider (OSP), è possibile abilitare l'autenticazione single sign-on degli utenti al portale di Identity Console. Prima di installare Identity Console è necessario installare OSP. Per configurare OSP per Identity Console, seguire i prompt visualizzati e specificare i valori richiesti per i parametri di configurazione. Per ulteriori informazioni, vedere "Distribuzione del container OSP" a pagina 20. Per registrare Identity Console in un server OSP esistente, è necessario aggiungere manualmente il seguente comando al file `ism-configuration.properties` nella cartella `/opt/NetIQ/IDM/Apps/tomcat/conf/`:

  ```
  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
  ```

  Nota: Con OSP è possibile effettuare la connessione a un solo albero eDirectory; OSP non supporta alberi eDirectory multipli.

- ❒ Accertarsi di disporre di una voce DNS corretta per il computer host in `/etc/hosts` con un nome host completo.
- ❒ Se si desidera utilizzare Identity Console con il browser Edge, è necessario effettuare il download della versione più recente di Microsoft Edge per usufruire di tutte le funzionalità.

Nota: Durante l'utilizzo di Identity Console in Mozilla Firefox, è possibile che l'operazione abbia esito negativo e che venga visualizzato il messaggio di errore `Origin Mismatch` (Discordanza dell'origine). Per risolvere il problema, eseguire i seguenti passaggi:

1. Aggiornare Firefox alla versione più recente.
2. Specificare `about:config` nel campo URL di Firefox e premere Invio.
3. Ricercare l'origine.
4. Fare doppio clic su `network.http.SendOriginHeader` e impostarne il valore su 1.

Configurazione dell'ambiente

Potrebbe essere necessario creare un file di configurazione contenente parametri specifici. Se si desidera configurare Identity Console con OSP, è necessario specificare i parametri specifici di OSP nel file di configurazione. Ad esempio, creare il file `edirapi.conf` seguente con i parametri OSP:

Nota: È necessario specificare il nome dell'albero eDirectory nel campo `osp-redirect-url`.

```
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"
```

Se si desidera configurare Identity Console senza OSP, creare un file di configurazione come illustrato di seguito, senza i parametri OSP:

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

Nota: Se si desidera configurare Identity Console con alberi eDirectory multipli, è possibile ignorare i parametri "*ldapserver*", "*ldapuser*" e "*ldappassword*" e creare il file di configurazione.

Tabella 1-1 *Descrizione dei parametri di configurazione nel file di configurazione*

Parametri di configurazione	Descrizione
listen	Specificare 9000 come porta di ascolto del server di Identity Console all'interno del container.
ldapserver	Specificare l'indirizzo IP del server host di eDirectory e il numero di porta.

Parametri di configurazione	Descrizione
ldapuser	Specificare il nome utente dell'utente eDirectory. Questo parametro viene utilizzato come credenziale per l'inizializzazione delle chiamate LDAP a eDirectory mediante il controllo delle autorizzazioni gestite via proxy nel caso di login OSP. L'utente LDAP deve disporre dei diritti di supervisore sull'albero eDirectory.
ldappassword	Specificare la password dell'utente LDAP.
pfxpassword	Specificare la password del file del certificato server pkcs12.
ospmode	Specificare `true` per integrare OSP con Identity Console. Se si imposta su `false`, Identity Console userà il login LDAP.
osp-token-endpoint	Questo URL viene utilizzato per recuperare determinati attributi dal server OSP per verificare la validità del token di autenticazione..
osp-authorize-url	Questo URL viene utilizzato dall'utente per fornire le credenziali per ottenere un token di autenticazione..
osp-logout-url	Utilizzare questo URL per terminare la sessione tra l'utente e il server OSP..
osp-redirect-url	Il server OSP reindirizza l'utente all'URL dopo aver concesso il token di autenticazione.. **Nota:** Durante la configurazione di Identity Console, assicurarsi di specificare in minuscolo il nome dell'albero eDirectory. Se il nome dell'albero non viene specificato in minuscolo, è possibile che il login al server di Identity Console abbia esito negativo.
osp-client-id	Specificare l'ID del client OSP fornito al momento della registrazione di Identity Console con OSP..
ospclientpass	Specificare la password del client OSP fornita al momento della registrazione di Identity Console con OSP..
ospcert	Specificare l'ubicazione del certificato CA del server OSP..
bcert	Specificare l'ubicazione del certificato CA di Identity Console.
loglevel	Specificare i livelli di log che si desidera includere nel file di log. Questo parametro può essere impostato su "fatal" (errore irreversibile), "error" (errore), "warn" (avviso) o "info".

Parametri di configurazione	Descrizione
check-origin	Se questa opzione è impostata su `true`, il server di Identity Console confronta il valore di origine delle richieste. Le opzioni disponibili sono `true` o `false`. Il parametro *origin* (origine) è obbligatorio anche se il valore del parametro *check-origin* (controllo origine) è impostato su `false` quando si utilizza la configurazione DNS.
origin	Identity Console confronta il valore origin (origine) delle richieste con i valori specificati in questo campo. **Nota:** A partire dalla versione 1.4 di Identity Console, questo parametro è indipendente dal parametro *check-origin* (controllo origine) ed è obbligatorio se viene utilizzata la configurazione DNS.
maxclients	Numero massimo di client simultanei che possono accedere a `IDConsole`. Tutti gli altri client oltre tale limite dovranno attendere in coda.

Nota

- Utilizzare il parametro di configurazione `ospmode` solo se si prevede di integrare OSP con Identity Console.
- Se Identity Applications (Identity Apps) è configurato in modalità cluster nella configurazione di Identity Manager, è necessario specificare il nome DNS del server del sistema di bilanciamento del carico nei campi `osp-token-endpoint`, `osp-authorize-url` e `osp-logout-url` nel file di configurazione. Se si specificano i dettagli del server OSP in questi campi, il login a Identity Console avrà esito negativo.
- Se Identity Console è configurato con la stessa istanza OSP di Identity Apps e Identity Reporting, il Single Sign-On (servizio di autenticazione) avrà effetto quando si esegue il login al portale di Identity Console.
- L'URL HTTPS OSP deve essere convalidato con certificati contenenti chiavi a 2048 bit o superiori a partire da Identity Console 1.4.
- Se si desidera limitare l'accesso al portale di Identity Console da domini diversi, impostare il parametro `samesitecookie` su `strict`. Se si desidera consentire l'accesso al portale di Identity Console da domini diversi, impostare il parametro `samesitecookie` su `lax`. Se il parametro non viene specificato durante la configurazione, di default verranno applicate le impostazioni del browser.

Una volta preparato il file di configurazione, procedere con la distribuzione del container. Per ulteriori informazioni, vedere “Distribuzione di Identity Console come container di Docker” a pagina 20.

Requisiti di sistema e prerequisiti per l'installazione autonoma (non Docker)

Requisiti di sistema

In questa sezione vengono illustrati i requisiti di sistema e i prerequisiti per l'installazione di Identity Console come applicazione autonoma.

Categoria	Requisiti minimi
Processore	1,4 GHz a 64 bit
Memoria	2 GB
Spazio su disco	200 MB su Linux
Browser supportato	◆ Versione più recente di **Microsoft Edge** ◆ Versione più recente di **Google Chrome** ◆ Versione più recente di **Mozilla Firefox** **Nota:** Durante l'utilizzo di Identity Console in Mozilla Firefox, è possibile che l'operazione abbia esito negativo e che venga visualizzato il messaggio di errore `Origin Mismatch` (Discordanza dell'origine). Per risolvere il problema, eseguire i seguenti passaggi: **1** Aggiornare Firefox alla versione più recente. **2** Specificare `about:config` nel campo URL di Firefox e premere Invio. **3** Ricercare l'origine. **4** Fare doppio clic su `network.http.SendOriginHeader` e impostarne il valore su 1.
Sistema operativo supportato	◆ **Certificato**: ◆ SUSE Linux Enterprise Server (SLES) 15 SP1, SP2 e SP3 ◆ SUSE Linux Enterprise Server (SLES) 12 SP1, SP2, SP3, SP4 e SP5 ◆ Red Hat Enterprise Linux (RHEL) 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, 8.4 e 8.5 ◆ OpenSUSE 15.1 e 15.2 ◆ **Supportato:** supportato nelle versioni successive dei Support Pack dei sistemi operativi certificati sopra indicati.

Categoria	Requisiti minimi
Certificati	◆ È necessario ottenere un certificato server pkcs12 con la chiave privata per cifrare/decifrare lo scambio di dati tra il client e il server di Identity Console. Questo certificato server viene utilizzato per proteggere la connessione HTTP. È possibile utilizzare i certificati server generati da una CA esterna. Per ulteriori informazioni, vedere Creating Server Certificate Objects (Creazione di oggetti Server Certificate). Il certificato server deve contenere il nome alternativo del soggetto con l'indirizzo IP e il DNS del server di Identity Console. Una volta creato l'oggetto Server Certificate, è necessario esportarlo in formato .pfx. ◆ È necessario ottenere un certificato CA per tutti gli alberi in formato `.pem` per convalidare la firma CA dei certificati server ottenuti nel passaggio precedente. Questo certificato CA radice garantisce inoltre una comunicazione LDAP protetta tra il client e il server di Identity Console. Ad esempio, è possibile ottenere il certificato CA di eDirectory (`SSCert.pem`) da `/var/opt/novell/eDirectory/data/SSCert.pem`.

Quando si è pronti, procedere con l'installazione di Identity Console. Per ulteriori informazioni, vedere “Distribuzione di Identity Console come applicazione autonoma” a pagina 24.

(Facoltativo) Prerequisito per la configurazione OSP

Utilizzando One SSO Provider (OSP), è possibile abilitare l'autenticazione Single Sign-On degli utenti al portale di Identity Console. Prima di installare Identity Console è necessario installare OSP. Per configurare OSP per Identity Console, seguire i prompt visualizzati e specificare i valori richiesti per i parametri di configurazione. Per ulteriori informazioni, vedere “Distribuzione del container OSP” a pagina 20. Per registrare Identity Console in un server OSP esistente, è necessario aggiungere manualmente il seguente comando al file `ism-configuration.properties` nella cartella `/opt/NetIQ/IDM/Apps/tomcat/conf/`:

```
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
```

Nota

- Se si installa OSP per la prima volta, è necessario specificare `'y'` per **Configure OSP with eDir API** (Configurare OSP con eDir API) e seguire le istruzioni sullo schermo per registrare Identity Console con OSP.
- Durante la configurazione di Identity Console, assicurarsi di specificare in minuscolo il nome dell'albero eDirectory. Se il nome dell'albero non viene specificato in minuscolo, è possibile che il login al server di Identity Console abbia esito negativo.
- Con OSP è possibile effettuare la connessione a un solo albero eDirectory; OSP non supporta alberi eDirectory multipli.

Requisiti di sistema e prerequisiti per la workstation

- “Requisiti di sistema” a pagina 16

Requisiti di sistema

In questa sezione vengono illustrati i requisiti di sistema e i prerequisiti per l'esecuzione della workstation di Identity Console.

Categoria	Requisiti minimi
Processore	1.5 GHz a 64 bit
Memoria	2 GB
Spazio su disco	1 GB su Windows

Sistema operativo supportato	• **Certificato**:
	  • Windows Server 2016
	  • Windows Server 2019
	  • Windows Server 2022
	  • Windows 10
	  • Windows 11

Categoria	Requisiti minimi
Certificati	◆ Per scambiare dati tra il client di Identity Console e il server REST, è necessario ottenere un certificato server in formato pfx. Il certificato server deve sempre essere denominato keys.pfx. Per ulteriori informazioni, vedere Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/b1j4tpo3.html#b1j4u0cm) (Creazione di oggetti Server Certificate). ◆ È necessario ottenere un certificato CA per tutti gli alberi in formato `.pem` per convalidare la firma CA dei certificati server ottenuti nel passaggio precedente. Questo certificato CA radice garantisce inoltre una comunicazione LDAP protetta tra il client e il server di Identity Console. Ad esempio, è possibile ottenere il certificato CA di eDirectory per Linux `SSCert.pem` da `/var/opt/novell/eDirectory/data/SSCert.pem`. Ottenere il certificato CA di eDirectory `SSCert.pem` per Windows da `<percorso di installazione di eDirectory>\NetIQ\eDirectory\DIBFiles\CertServ\SSCert.pem`.

Quando si è pronti, procedere con la distribuzione di Identity Console. Per ulteriori informazioni, vedere "Identity Console su Windows come workstation" a pagina 27.

Verifica della firma RPM

Per eseguire la verifica della firma RPM, utilizzare la procedura seguente:

1. Accedere alla cartella in cui viene estratta la build.

   Ad esempio: `<percorso di decompressione di Identity Console>/IdentityConsole_150_Linux/license/MicroFocusGPGPackageSign.pub`.

2. Eseguire il seguente comando per importare la chiave pubblica:

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

3. (Facoltativo) Eseguire il comando seguente per verificare la firma RPM: `rpm --checksig -v <Nome RPM>`

   Ad esempio:

   ```
   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 Distribuzione di Identity Console

In questo capitolo viene descritto il processo di installazione di Identity Console e i suggerimenti sulla sicurezza. Per prepararsi all'installazione, rivedere i prerequisiti e i requisiti di sistema che si mostrano nel Capitolo 1, “Pianificazione dell'installazione di Identity Console”, a pagina 9.

Suggerimenti sulla sicurezza

- Per default, i container di Docker non dispongono di vincoli di risorsa. In questo modo, tutti i container dispongono dell'accesso a tutte le risorse della CPU e della memoria fornite dal kernel dell'host. È inoltre necessario garantire che un container in esecuzione non utilizzi più risorse e che non le sottragga agli altri container in esecuzione, impostando limiti per la quantità di risorse che è possibile utilizzare in un container.
  - In questo caso, il container di Docker deve verificare che venga applicato un limite rigido per la memoria utilizzata dal container tramite il flag `--memory` sul comando di esecuzione di Docker.
  - In questo caso, il container di Docker deve verificare che il limite venga applicato alla quantità di CPU utilizzata da un container in esecuzione mediante il flag `--cpuset-cpus` sul comando di esecuzione di Docker.
- `--pids-limit` deve essere impostato su 300 per limitare il numero di thread del kernel generati all'interno del container in un determinato momento. In questo modo si evitano gli attacchi DoS.
- È necessario impostare la policy di riavvio del container in caso di errore su 5, mediante il flag `--restart` sul comando di esecuzione di Docker.
- È necessario utilizzare solo il container una volta che lo stato di integrità risulta essere **Integro** dopo l'avvio del container. Per verificare lo stato di integrità del container, eseguire il comando seguente:

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

- Il container di Docker viene sempre avviato come utente non root (nds). Come ulteriore misura di sicurezza, abilitare la rimappatura dello spazio dei nomi dell'utente sul daemon per impedire attacchi con escalation dei privilegi all'interno del container. Per ulteriori informazioni sulla rimappatura dello spazio dei nomi dell'utente, vedere Isolate containers with a user namespace (Isolare i container con uno spazio dei nomi dell'utente).

Distribuzione di Identity Console come container di Docker

In questa sezione sono incluse le seguenti procedure:

Distribuzione del container OSP

Per distribuire il container OSP, eseguire i seguenti passaggi:

1. Eseguire il login a Software License and Download (https://sld.microfocus.com/) (Licenze software e download) e accedere alla pagina Software Downloads (Download dei software).
2. Selezionare quanto segue:
   - Product (Prodotto): eDirectory
   - Product Name (Nome prodotto): eDirectory per User Sub SW E-LTU
   - Version (Versione): 9.2
3. Eseguire il download del file: `IdentityConsole_<versione>_Containers_tar.zip`.
4. Estrarre il file scaricato in una cartella.
5. Modificare il file delle proprietà distribuite in batch in base alle proprie esigenze. Di seguito viene riportato un esempio di file delle proprietà distribuite in batch:

```
# 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
```

Nota: Per evitare vincoli di spazio durante l'utilizzo del file delle proprietà distribuite in batch (testo DOS), è necessario convertire il file di testo DOS in formato UNIX utilizzando lo strumento dos2unix. Eseguire il comando seguente per convertire il file di testo da terminazioni riga DOS a terminazioni riga Unix:

```
dos2unix nomefile
```

Ad esempio:

```
dos2unix filediesempio
```

---

6 Generare un certificato server (`cert.der`) tramite iManager e importarlo nell'archivio chiavi (`tomcat.ks`). Copiare il file delle proprietà distribuite in batch e l'archivio chiavi (`tomcat.ks`) in qualsiasi directory. Ad esempio, `/data`. Per creare un certificato server e importarlo nell'archivio chiavi, eseguire i seguenti passaggi:

6a Eseguire il seguente comando per creare un archivio chiavi (`tomcat.ks`). Generare la chiave e accertarsi che il nome CN o il nome host completo del computer corrispondano all'indirizzo IP.

```
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 Eseguire il seguente comando per creare una richiesta di firma del certificato. Ad esempio, `cert.csr`.

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

6c Passare `cert.csr` a iManager per ottenere il certificato server `osp.der`. Assicurarsi di selezionare il tipo di chiave Personalizzato e le opzioni di Utilizzo chiavi quali Cifratura dei dati, Cifratura chiavi e Firma digitale nonché il campo Nome/i alternativo/i del soggetto del

certificato in modo da includere l'indirizzo IP o il nome host del server OSP. Per ulteriori informazioni, vedere Creating a Server Certificate Object (Creazione di un oggetto Server Certificate).

6d Eseguire il seguente comando per importare il certificato CA (SSCert.der) e il certificato server (cert.der) nell'archivio chiavi tomcat.ks.

```
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 Eseguire il comando seguente per caricare l'immagine OSP:

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

8 Distribuire il container utilizzando il seguente comando:

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

Ad esempio:

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

Distribuzione di Identity Console come container di Docker

In questa sezione viene illustrata la procedura di distribuzione di Identity Console come container di Docker:

Nota: I parametri di configurazione, i valori di esempio e gli esempi menzionati in questa procedura sono solo a scopo di riferimento. Accertarsi di non utilizzarli direttamente nell'ambiente di produzione.

---

1. Eseguire il login a SLD: Software License and Download (https://sld.microfocus.com/) (Licenze software e download) e accedere alla pagina Software Downloads (Download dei software).
2. Selezionare quanto segue:
   - Product (Prodotto): eDirectory
   - Product Name (Nome prodotto): eDirectory per User Sub SW E-LTU
   - Version (Versione): 9.2
3. Eseguire il download del file: IdentityConsole_<versione>_Container.tar.zip.
4. L'immagine deve essere caricata nel registro locale di Docker. Estrarre e caricare il file IdentityConsole_<versione>_Containers.tar.gz utilizzando i comandi seguenti:

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

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

5. Creare il container Docker di Identity Console utilizzando il seguente comando:

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

Ad esempio:

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

Nota

- È possibile accettare l'EULA impostando la variabile d'ambiente ACCEPT_EULA su `'Y'`. È anche possibile accettare l'EULA dal prompt sullo schermo mentre si avvia il container utilizzando l'opzione `-it` nel comando di creazione di Docker per la modalità interattiva.
- Il parametro `--volume` nel comando precedente creerà un volume per la memorizzazione dei dati di configurazione e di log. In questo caso, è stato creato un volume di esempio denominato `IDConsole-volume`.

6 Copiare il file del certificato server dal file system locale al container come `/etc/opt/novell/eDirAPI/cert/keys.pfx` utilizzando il seguente comando. Per ulteriori informazioni sulla creazione del certificato del server, vedere "Prerequisiti" a pagina 9:

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

Ad esempio:

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

Quando si esegue la connessione a più alberi eDirectory, è necessario assicurarsi di ottenere almeno un certificato server keys.pfx per tutti gli alberi connessi.

7 Copiare il file del certificato CA (`.pem`) dal file system locale al container come `/etc/opt/novell/eDirAPI/cert/sscert.pem` utilizzando il seguente comando. Per ulteriori informazioni su come ottenere il certificato CA, vedere "Prerequisiti" a pagina 9:

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

Ad esempio:

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

Se l'utente deve effettuare la connessione a più alberi eDirectory, fare riferimento alla sezione seguente: "Alberi multipli con Identity Console come Docker" a pagina 24

8 Modificare il file di configurazione in base alle proprie esigenze e copiare il file di configurazione (`edirapi.conf`) dal file system locale al container come `/etc/opt/novell/eDirAPI/conf/edirapi.conf` utilizzando il seguente comando:

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

Ad esempio:

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

9 Avviare il container Docker utilizzando il seguente comando:

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

Ad esempio:

```
docker start identityconsole-container
```

Nota: Nella directory `/var/lib/docker/volumes/<nome_volume>/_data/eDirAPI/var/log` sono disponibili i seguenti file di log:

- edirapi.log - Utilizzato per registrare diversi eventi in edirapi e per eseguire il debug dei problemi.
- edirapi_audit.log - Utilizzato per la registrazione degli eventi di revisione di edirapi. I log seguono il formato di revisione CEF.
- container-startup.log - Utilizzato per acquisire i log di installazione del container di Docker di Identity Console.

Alberi multipli con Identity Console come Docker

Identity Console consente all'utente di effettuare la connessione a più alberi ottenendo il certificato CA individuale dell'albero.

Ad esempio, se si effettua la connessione a tre alberi eDirectory, è necessario copiare tutti e tre i certificati CA nel container di Docker:

```
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
```

Per riavviare Identity Console, eseguire i comandi seguenti:

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

Distribuzione di Identity Console come applicazione autonoma

- “Distribuzione di Identity Console come applicazione autonoma (non Docker)” a pagina 25
- “Alberi multipli con Identity Console come applicazione autonoma” a pagina 26

Distribuzione di Identity Console come applicazione autonoma (non Docker)

In questa sezione viene illustrata la procedura di distribuzione di Identity Console come applicazione autonoma:

1. Eseguire il login a SLD: Software License and Download (https://sld.microfocus.com/) (Licenze software e download) e accedere alla pagina Software Downloads (Download dei software).
2. Selezionare quanto segue:
   - Product (Prodotto): eDirectory
   - Product Name (Nome prodotto): eDirectory per User Sub SW E-LTU
   - Version (Versione): 9.2
3. Effettuare il download della build più recente di Identity Console.
4. Estrarre il file scaricato in una cartella.
5. Aprire una shell e accedere alla cartella in cui è stata estratta la build di Identity Console.
6. Eseguire il comando seguente dopo aver eseguito il login come utente root o equivalente:

   ```
   ./identityconsole_install
   ```

7. Leggere l'Introduzione e fare clic su **ENTER**.
8. Fare clic su `'Y'` (S) per accettare il Contratto di licenza. Verranno installati tutti gli RPM necessari nel sistema.
9. Immettere il nome host del server di Identity Console (FQDN)/l'indirizzo IP.
10. Immettere il numero della porta di ascolto di Identity Console. Il valore di default è 9000.
11. Immettere l'opzione che consente di integrare OSP con Identity Console o che consente a Identity Console di utilizzare il login LDAP.
12. Se si desidera integrare OSP con Identity Console:
    1. Immettere il nome di dominio/l'indirizzo IP del server eDirectory/Identity Vault con il numero di porta LDAPS.

       Ad esempio:

       192.168.1.1:636
    2. Immettere il nome utente eDirectory/Identity Vault.

       Ad esempio:

       cn=admin,ou=org_unit,o=org
    3. Immettere la password di eDirectory/Identity Vault.
    4. Immettere nuovamente la password di eDirectory/Identity Vault per confermarla.
    5. Immettere il nome di dominio/indirizzo IP del server OSP con il numero di porta SSL del server SSO.
    6. Immettere l'ID del client OSP.
    7. Immettere la password del client OSP.
    8. Immettere il nome dell'albero eDirectory/Identity Vault.
13. Immettere il percorso dei certificati fonte attendibile (`SSCert.pem`), inclusa la cartella.

Ad esempio:

`/home/Identity_Console/certs`

Nota: L'utente deve assicurarsi di non creare sottodirectory all'interno della cartella cert.

14 Immettere il percorso del certificato server (`keys.pfx`), incluso il nome del file.

Ad esempio:

`/home/Identity_Console/keys.pfx`

15 Immettere la password del certificato server. Per confermare la corretta immissione della password del certificato server, immetterla nuovamente. L'installazione viene avviata.

Nota: Nella directory `/var/opt/novell/eDirAPI/log` sono disponibili i seguenti file di log:

- edirapi.log - Utilizzato per registrare diversi eventi in edirapi e per eseguire il debug dei problemi.
- edirapi_audit.log - Utilizzato per la registrazione degli eventi di revisione di edirapi. I log seguono il formato di revisione CEF.
- identityconsole_install.log - Utilizzato per acquisire i log di installazione di Identity Console.

I log di avvio/arresto del processo Identity Console sono disponibili nel file `/var/log/messages`.

Nota: Quando si installa Identity Console ed eDirectory sullo stesso computer, NetIQ consiglia di mettere a disposizione del computer almeno un'istanza di eDirectory.

Alberi multipli con Identity Console come applicazione autonoma

Quando si esegue la connessione a più alberi eDirectory, è necessario assicurarsi di ottenere il certificato CA individuale dell'albero.

Ad esempio, se si effettua la connessione a tre alberi eDirectory, è necessario copiare tutti e tre i certificati CA nella directory `etc/opt/novell/eDirAPI/cert/`:

```
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
```

Per riavviare Identity Console, eseguire uno dei comandi seguenti:

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

oppure

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

Identity Console su Windows come workstation

Identity Console può essere avviato su Windows come workstation e richiede l'esecuzione dei servizi REST. Pertanto, quando viene avviato, un processo eDirAPI viene eseguito nel prompt dei comandi edirapi.exe. Se il terminale edirapi.exe viene chiuso, Identity Console smette di funzionare.

Nella procedura seguente viene descritto come eseguire Identity Console su Windows.

1. Eseguire il login a SLD: Software License and Download (https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0) (Licenze software e download) e accedere alla pagina Software Downloads (Download dei software).
2. Selezionare quanto segue:
   - Product (Prodotto): eDirectory
   - Product Name (Nome prodotto): eDirectory per User Sub SW E-LTU
   - Version (Versione): 9.2
3. Effettuare il download del file `IdentityConsole_<versione>_workstation_win_x86_64.zip`.
4. Estrarre il file `IdentityConsole_<version>_workstation_win_x86_64.zip` scaricato in una cartella.
5. Accedere alla cartella estratta: `IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert` e copiare la CA della fonte attendibile SS`Cert.pem` e il certificato server `keys.pfx`.

   Per ottenere i certificati, fare riferimento alla sezione: “Requisiti di sistema e prerequisiti per la workstation” a pagina 16

   Se l'utente deve effettuare la connessione a più alberi eDirectory, fare riferimento alla sezione: “Alberi multipli con Identity Console come workstation” a pagina 28

   Nota: Il nome del certificato server deve sempre essere `keys.pfx`.
6. Accedere alla cartella in cui viene estratta la build e fare doppio clic sul file `run.bat` (file batch di Windows).
7. Immettere la password del certificato server (`keys.pfx`) nel prompt dei comandi.

   Il terminale del processo eDirAPI (edirapi.exe) viene avviato e viene visualizzata la pagina di login di Identity Console.

Nota:

- Se il terminale del processo `eDirAPI` (edirapi.exe) è già in esecuzione, eseguire `identityconsole.exe` dalla cartella estratta della build.
- Gli utenti possono trovare i seguenti log in `:\IdentityConsole_150_workstation_win_x86_64\eDirAPI\log`

`edirapi.log` - Utilizzato per registrare diversi eventi in `edirapi` e per eseguire il debug dei problemi.

`edirapi_audit.log` - Utilizzato per registrare gli eventi di revisione di `edirapi`. I log seguono il formato di revisione CEF.

- Il login basato su OSP non è supportato in modalità workstation.
- L'ascolto sulla workstation di Identity Console è solo sulla `porta 9000`. Non modificare il file `edirapi_win.conf`.

Alberi multipli con Identity Console come workstation

Identity Console consente all'utente di effettuare la connessione a più alberi ottenendo il certificato CA individuale dell'albero.

1. Chiudere la workstation di Identity Console e il terminale eDirAPI.
2. Copiare i certificati CA `SSCert.pem` nell'ubicazione: `IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert.`

   Ad esempio, se si desidera eseguire la connessione a tre alberi eDirectory, copiare i certificati CA rispettivamente come `SSCert1.pem`, `SSCert2.pem` e `SSCert3.pem`.
3. Accedere alla cartella in cui viene estratta la build e fare doppio clic sul file `run.bat` (file batch di Windows).
4. Immettere la password di `keys.pfx` nel prompt del terminale ed eseguire il login all'albero eDirectory desiderato.

Arresto e riavvio di Identity Console

Arresto e riavvio di Identity Console come container di Docker

Per arrestare Identity Console, eseguire il seguente comando:

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

Per riavviare Identity Console, eseguire il seguente comando:

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

Per avviare Identity Console, eseguire il seguente comando:

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

Arresto e riavvio di Identity Console come applicazione autonoma

Per arrestare Identity Console, eseguire uno dei seguenti comandi:

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

oppure

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

Per riavviare Identity Console, eseguire uno dei seguenti comandi:

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

oppure

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

Per avviare Identity Console, eseguire uno dei seguenti comandi:

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

oppure

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

Chiusura e riavvio della workstation di Identity Console

Per chiudere l'applicazione e il processo, seguire la procedura riportata di seguito:

1. Chiudere l'applicazione desktop Windows Identity Console.
2. Arrestare il processo eDirAPI chiudendo il terminale del processo `eDirAPI`.

Per riavviare la workstation di Identity Console, accedere alla cartella in cui viene estratta la build e fare doppio clic sul file `run.bat` (file batch di Windows).

Nota: Se il terminale del processo `eDirAPI` è già in esecuzione, eseguire `identityconsole.exe` dalla cartella estratta della build per riavviare la workstation di Identity Console.

Gestione della persistenza dei dati

Oltre ai container di Identity Console, vengono creati anche i volumi per la persistenza dei dati. Per utilizzare i parametri di configurazione di un container precedente utilizzando i volumi, eseguire i passaggi seguenti:

1. Arrestare il container di Docker corrente utilizzando il comando seguente:

   `docker stop` *`identityconsole-container`*

2. Creare il secondo container utilizzando i dati dell'applicazione del container precedente memorizzato nel volume di Docker (`edirapi-volume-1`):

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

3 Avviare il secondo container utilizzando il seguente comando:

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

4 (Facoltativo) A questo punto è possibile rimuovere il primo container utilizzando il comando seguente:

```
docker rm identityconsole-container
```

Distribuzione di Identity Console in Azure Kubernetes Service

Azure Kubernetes Service (AKS) è un servizio Kubernetes gestito che consente di distribuire e gestire cluster. In questa sezione sono incluse le seguenti procedure:

Distribuzione di Identity Console nel cluster AKS

In questa sezione vengono descritte le procedure per la distribuzione di Identity Console nel cluster AKS:

Creazione di un Azure Container Registry (ACR)

Azure Container Registry (ACR) è un registro privato basato su Azure per le immagini del container di Docker.

Per ulteriori dettagli sui passaggi, vedere la sezione Create an Azure container registry using the Azure portal (Creare un'istanza di Registro Azure Container usando il portale di Azure) in Create container registry - Portal (Creare un registro contenitori - Portale) o attenersi ai seguenti passaggi per creare un Azure Container Registry (ACR):

1. Accedere ad Azure Portal (Portale di Azure).
2. Accedere a **Create a resource** (Crea una risorsa) > **Containers** (Contenitori) > **Container Registry**.
3. Nella scheda **Basics** (Informazioni di base), specificare i valori per **Resource group** (Gruppo di risorse) e **Registry name** (Nome registro). Il nome del registro deve essere univoco all'interno di Azure e contenere un minimo di 5 e un massimo di 50 caratteri alfanumerici.

   Accettare i valori di default per le impostazioni rimanenti.
4. Fare clic su **Review + create** (Rivedi e crea).
5. Fare clic su **Create** (Crea).

6. Accedere all'interfaccia della riga di comando di Azure, eseguire il comando seguente per eseguire il login ad Azure Container Registry

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

   Ad esempio:

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

7. Recuperare il server di login dell'Azure Container Registry utilizzando il comando:

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

   Ad esempio:

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

8. Contrassegnare l'immagine locale di Identity Console con il nome del server di login ACR (registryname.azureacr.io) utilizzando il seguente comando:

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

   Ad esempio:

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

9. Eseguire il push dell'immagine contrassegnata nel registro.

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

   Ad esempio:

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

10. Recuperare l'elenco di immagini nel registro utilizzando il comando:

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

Impostazione di un cluster Kubernetes

Creare una risorsa del servizio Kubernetes utilizzando il portale di Azure o l'interfaccia della riga di comando.

Per ulteriori dettagli sui passaggi relativi alla creazione di una risorsa del servizio Kubernetes in Azure con un nodo, vedere Create an AKS Cluster (Creare un cluster del servizio Azure Kubernetes) in Azure Quickstart (Avvio rapido Azure).

Nota:

- Assicurarsi di selezionare Azure CNI come rete.
- Selezionare la rete virtuale esistente (in cui il server eDirectory è distribuito nella sottorete).
- Selezionare il Container Registry esistente in cui è disponibile l'immagine di Identity Console.

Creazione di un indirizzo IP pubblico SKU standard

Una risorsa indirizzo IP pubblico nel gruppo di risorse cluster di Kubernetes funge da IP del sistema di bilanciamento del carico per l'applicazione.

Per i dettagli sui passaggi, vedere Create a public IP address using the Azure portal (Creare un indirizzo IP pubblico usando il portale di Azure) in Create public IP address – Portal (Creare indirizzo IP pubblico – Portale).

Configurazione di Cloud Shell e connessione al cluster Kubernetes

Utilizzare Cloud Shell, disponibile nel portale di Azure per tutte le operazioni.

Per configurare Cloud Shell nel portale di Azure, vedere la sezione Start Cloud Shell (Avviare Cloud Shell) in Bash – Quickstart (Bash - Guida introduttiva) oppure eseguire la procedura seguente per configurare Cloud Shell ed effettuare la connessione al cluster Kubernetes:

1. Nel portale di Azure, fare clic sul pulsante  per aprire Cloud Shell.

   Nota: Per gestire un cluster Kubernetes, utilizzare il client della riga di comando Kubernetes `kubectl`. kubectl è già installato se si utilizza Azure Cloud Shell.

2. Configurare kubectl per effettuare la connessione al cluster Kubernetes utilizzando il seguente comando:

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

   Ad esempio:

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

3. Verificare l'elenco dei nodi del cluster utilizzando il comando:

   ```
   kubectl get nodes
   ```

Distribuzione dell'applicazione

Per distribuire Identity Console, è possibile utilizzare file di esempio `idc-services.yaml`, `idc-statefulset.yaml`, `idc-storageclass.yaml` e `idc-pvc.yaml`.

È inoltre possibile creare file yaml personalizzati in base alle necessità.

1. Creare una risorsa di classe di memorizzazione utilizzando il comando seguente:

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

   Ad esempio:

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

(Facoltativo) Per ulteriori informazioni su come creare e utilizzare dinamicamente il volume di persistenza con la condivisione file di Azure, vedere Dynamically create and use a persistent volume with Azure Files in Azure Kubernetes Service (AKS) (Creare dinamicamente e usare un volume persistente con File di Azure nel servizio Azure Kubernetes)

Di seguito è riportato un esempio di file di risorsa della classe di memorizzazione:

```
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
~
```

Una risorsa di classe di memorizzazione consente di eseguire il provisioning dinamico della memorizzazione. Viene utilizzata per definire la modalità di creazione di una condivisione file di Azure.

2. Visualizzare i dettagli della classe di memorizzazione utilizzando il comando seguente:

   ```
   kubectl get sc
   ```

3. Creare una risorsa PVC utilizzando il file `idc-pvc.yaml`:

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

   Ad esempio:

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

   Di seguito è riportato un esempio di file di risorsa PVC:

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

   Una risorsa di richiesta del volume di persistenza crea la condivisione file. Un Persistent Volume Claim (PVC) utilizza l'oggetto della classe di memorizzazione per eseguire dinamicamente il provisioning di una condivisione file di Azure.

4. Effettuare l'upload del file edirapi.conf, del certificato CA e del certificato server in Shell Cloud.

   Fare clic sull'icona del pulsante **Upload/Download files** (Carica/Scarica file)  su Cloud Shell ed effettuare l'upload dei file `edirapi.conf`, `SSCert.pem` e `keys.pfx`.

   Nota: edirapi.conf ha un parametro "origin". Qui è necessario specificare l'indirizzo IP con cui accedere all'applicazione Identity Console. Utilizzare l'indirizzo IP creato nella sezione "Creazione di un indirizzo IP pubblico SKU standard" a pagina 32.

   La distribuzione di Identity Console richiede il certificato server (`keys.pfx`).

   Durante la creazione del certificato server, assicurarsi di fornire un nome DNS valido nel campo del nome alternativo del soggetto.

   Passaggi per creare un nome DNS valido:

   Un pod tipico distribuito mediante StatefulSet ha un nome DNS simile al seguente: {statefulsetname}-{ordinal}.{servicename}.{namespace}.svc.cluster.local

   - Se il nome StatefulSet nel file idconsole-statefulset.yaml è idconsole-app, statefulsetname = idconsole-app
   - Se è il primo pod, ordinal = 0
   - Se si definisce serviceName come idconsole nel file idconsole-statefulset.yaml, serviceName = idconsole
   - Se è lo spazio dei nomi di default, namespace=default

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

   ---

5. Creare una risorsa configmap nel cluster Kubernetes in cui memorizzare i file di configurazione insieme ai certificati.

   Prima di eseguire il comando, verificare che i file (`edirapi.conf`, `SSCert.pem` e `keys.pfx`) siano presenti nella directory.

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

   Ad esempio:

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

6. Visualizzare i dettagli dell'oggetto configmap utilizzando il comando kubectl describe:

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

   Ad esempio:

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

7. Creare una risorsa StatefulSet per distribuire il container.

   Eseguire il comando seguente per distribuire il container:

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

   Ad esempio:

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

   Di seguito è riportato un esempio di file di risorsa StatefulSet:

```
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. Eseguire il seguente comando per verificare lo stato del pod distribuito:

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

9. Creare una risorsa del servizio di tipo loadBalancer.

   Il tipo di servizio specificato nel file yaml è loadBalancer.

   Creare una risorsa del servizio utilizzando il comando seguente:

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

   Ad esempio:

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

   Di seguito è riportato un esempio di file di risorsa di servizio:

```
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
```

Controllare l'indirizzo EXTERNAL-IP (o loadBalancerIP) utilizzando il comando seguente:

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

10. Avviare l'URL utilizzando EXTERNAL-IP (o l'indirizzo loadBalancerIP).

    Ad esempio:

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

Modifica del certificato server

Le sezioni seguenti forniscono informazioni sulla modifica del certificato server nel container di Docker e in Identity Console come applicazione autonoma.

- "Modifica del certificato server nel container di Docker" a pagina 36
- "Modifica del certificato server in Identity Console come applicazione autonoma" a pagina 37

Modifica del certificato server nel container di Docker

Per modificare il certificato server nel container di Docker, eseguire la procedura seguente:

1. Eseguire il comando seguente per copiare il nuovo certificato server in qualsiasi ubicazione del container.

   Ad esempio:

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

2. Eseguire il login al container utilizzando il seguente comando:

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

3. Eseguire NLPCERT per memorizzare le chiavi come pseudo utente:

   ```
   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. Uscire dalla console del container utilizzando il comando:

```
exit
```

5 Riavviare il container immettendo:

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

Modifica del certificato server in Identity Console come applicazione autonoma

Per modificare il certificato server nel container autonomo, eseguire la procedura seguente:

1 Eseguire NLPCERT per memorizzare le chiavi:

```
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 Riavviare Identity Console:

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

3 Upgrade di Identity Console

In questo capitolo viene descritto il processo di upgrade di Identity Console alle versioni più recenti. Per prepararsi all'upgrade, rivedere i prerequisiti e i requisiti di sistema riportati nel Capitolo 1, “Pianificazione dell'installazione di Identity Console”, a pagina 9.

In questa sezione sono incluse le seguenti procedure:

Aggiornamento di Identity Console come container di Docker

Quando è disponibile una nuova versione dell'immagine di Identity Console, l'amministratore può eseguire una procedura di upgrade per la distribuzione del container con la versione più recente di Identity Console. Accertarsi di memorizzare costantemente tutti i dati necessari relativi alle applicazioni nei volumi di Docker prima di eseguire un upgrade. Eseguire i seguenti passaggi per eseguire l'upgrade di Identity Console mediante il container di Docker:

1. Effettuare il download e caricare l'ultima versione dell'immagine Docker da Software License and Download (https://sld.microfocus.com/) (Licenze software e download) ed eseguire i passaggi per installare la versione più recente di Identity Console come indicato in “Distribuzione di Identity Console” a pagina 19.
2. Una volta caricata l'immagine Docker più recente, arrestare il container di Docker corrente utilizzando il comando seguente:

   ```
   docker stop identityconsole-container
   ```

3. (Facoltativo) Eseguire il backup del volume condiviso.
4. Eliminare il container di Identity Console esistente eseguendo il seguente comando:

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

   Ad esempio:

   ```
   docker rm identityconsole-container
   ```

5. (Facoltativo) Eliminare l'immagine Docker di Identity Console obsoleta eseguendo il seguente comando:

   ```
   docker rmi identityconsole
   ```

6. Creare il container Docker di Identity Console utilizzando il seguente comando:

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

Ad esempio:

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

Nota

- È possibile accettare l'EULA impostando la variabile d'ambiente ACCEPT_EULA su `'Y'`. È anche possibile accettare l'EULA dal prompt sullo schermo mentre si avvia il container utilizzando l'opzione `-it` nel comando di creazione di Docker per la modalità interattiva.
- Il parametro `--volume` nel comando precedente creerà un volume per la memorizzazione dei dati di configurazione e di log. In questo caso, è stato creato un volume di esempio denominato `IDConsole-volume`.

---

7 Copiare il file del certificato server dal file system locale al container appena creato come `/etc/opt/novell/eDirAPI/cert/keys.pfx` utilizzando il seguente comando:

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

Ad esempio:

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

Quando si esegue la connessione a più alberi eDirectory, è necessario assicurarsi di copiare almeno un certificato server keys.pfx per tutti gli alberi connessi.

8 Copiare il file del certificato CA (`.pem`) dal file system locale al container appena creato come `/etc/opt/novell/eDirAPI/cert/sscert.pem` utilizzando il seguente comando:

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

Ad esempio:

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

Quando si esegue la connessione a più alberi eDirectory, è necessario assicurarsi di ottenere il certificato CA individuale per tutti gli alberi connessi. Ad esempio, se si effettua la connessione a tre alberi eDirectory, è necessario copiare tutti e tre i certificati CA nel container di Docker:

```
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
```

Nota: A partire dalla versione 1.4 di Identity Console, il file di configurazione (edirapi.conf) non include esplicitamente i parametri "*ldapuser*", "*ldappassword*" e "*ldapserver*". Il valore del parametro "*bcert*" deve includere il percorso della directory per i certificati fonte attendibile. Ad esempio, bcert = "/etc/opt/novell/eDirAPI/cert/". Il parametro "*origin*" è indipendente dal parametro "*check-origin*" ed è obbligatorio quando si utilizza la configurazione DNS.

9 Copiare il file di configurazione (`edirapi.conf`) dal file system locale al container appena creato come `/etc/opt/novell/eDirAPI/conf/edirapi.conf` utilizzando il seguente comando:

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

Ad esempio:

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

10 Avviare il secondo container utilizzando il seguente comando:

```
docker start identityconsole-container
```

11 Eseguire il seguente comando per verificare lo stato del container in esecuzione:

```
docker ps -a
```

Upgrade di Identity Console come applicazione autonoma (non Docker)

In questa sezione viene illustrata la procedura di upgrade di Identity Console come applicazione autonoma:

1 Effettuare il download di `IdentityConsole_<versione>_Containers.tar.gz` da Software License and Download (https://sld.microfocus.com/) (Licenze software e download)

2 Eseguire il login a SLD, accedere alla pagina SLD di download del software e fare clic su **Download**

3 Spostarsi selezionando Product (Prodotto): **eDirectory** > Product Name (Nome prodotto): **eDirectory per User Sub SW E-LTU** > Version (Versione): **9.2**

4 Effettuare il download della build più recente di Identity Console.

5 Estrarre il file scaricato utilizzando il seguente comando:

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

6 Accedere alla cartella in cui è stata estratta la build di Identity Console.

7 Copiare in una cartella tutti i certificati fonte attendibile degli alberi eDirectory a cui si desidera eseguire la connessione. Per copiare il certificato fonte attendibile nella cartella, eseguire il seguente comando:

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

Ad esempio:

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

8 Eseguire il comando seguente:

```
./identityconsole_install
```

9 Specificare il percorso della cartella dei certificati fonte attendibile utilizzato al **passaggio 4**.

10 L'upgrade di Identity Console viene completato correttamente.

Upgrade del container OSP

Per eseguire l'upgrade del container OSP, eseguire i seguenti passaggi:

1 Effettuare il download e caricare l'ultima versione dell'immagine OSP da Software License and Download (https://sld.microfocus.com/) (Licenze software e download).

Ad esempio:

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

2 Una volta caricata l'immagine OSP più recente, arrestare il container di OSP corrente utilizzando il comando seguente:

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

3 (Facoltativo) Eseguire il backup del volume condiviso.

4 Eliminare il container OSP esistente eseguendo il seguente comando:

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

Ad esempio:

```
docker rm OSP_Container
```

5 Accedere alla directory contenente l'archivio chiavi (`tomcat.ks`) e il file delle proprietà distribuite in batch, eliminare l'archivio chiavi esistente (`tomcat.ks`) e mantenere la cartella OSP esistente. Generare un nuovo archivio chiavi (`tomcat.ks`) con dimensione della chiave pari a 2048. Per ulteriori informazioni, vedere il **passaggio 4** nella sezione Deploying the OSP Container (Distribuzione del container OSP) della Identity Console Installation Guide (Guida all'installazione di Identity Console).

6 Distribuire il container utilizzando il seguente comando:

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

Ad esempio:

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

4 Disinstallazione di Identity Console

In questo capitolo viene descritto il processo di disinstallazione di Identity Console:

Procedura di disinstallazione per l'ambiente Docker

Per disinstallare il container di Docker di Identity Console, eseguire i seguenti passaggi:

1. Arrestare il container di Identity Console:

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

2. Eseguire il seguente comando per rimuovere il container di Docker di Identity Console:

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

3. Eseguire il seguente comando per rimuovere l'immagine di Docker:

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

4. Rimuovere il volume di Docker:

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

Nota: Se si rimuove il volume, i dati verranno rimossi anche dal server.

Procedura di disinstallazione per Identity Console come applicazione autonoma (non Docker)

Per disinstallare Identity Console come applicazione autonoma, eseguire i seguenti passaggi:

1. Passare alla directory `/usr/bin` sul computer in cui è installato Identity Console.
2. Eseguire il comando seguente:

   ```
   ./identityconsoleUninstall
   ```

3. La disinstallazione di Identity Console viene completata correttamente.

Nota: Se sul computer è installato eDirectory o un altro prodotto NetIQ, l'utente deve disinstallare manualmente *nici* e *openssl*.