Identity Console

安裝指南

2022 年 9 月

法律聲明

目錄

關於本書和文件庫

*《Identity Console 安裝指南》*會提供如何安裝和管理 NetIQ Identity Console (Identity Console) 產品的資訊。此指南定義了術語，並介紹了多個實作案例。

適用對象

本指南適用於網路管理員。

文件庫其他資訊

文件庫提供下列資訊資源：

安裝指南

描述如何安裝並升級 Identity Console。本書適用於網路管理員。

關於 NetIQ Corporation

我們是一家全球性企業軟體公司，著重於處理您環境中三個不斷出現的挑戰：變動、複雜性和風險，以及我們可以如何協助您進行控制。

我們的觀點

因應變動及管理複雜性和風險已不是新資訊

事實上，在您所面對的挑戰中，這些或許是最明顯的變數，可控制您是否可以安全地測量、監控及管理您的實體、虛擬和雲端運算環境。

更有效、更快速地啟用重要的業務服務

我們認為對 IT 組織提供最大控制權限，是提供及時服務交付並符合成本效益的唯一方式。隨著組織繼續推動革新，用來進行管理的技術也日益複雜，由變動及複雜性所帶來的壓力只會繼續提高。

經營理念

不只銷售軟體，而是銷售智慧型解決方案

為提供可靠的控制，我們會先確保已瞭解真實世界中與您類似的 IT 組織日常的操作方式。這是我們能夠開發出實際的智慧型 IT 解決方案的唯一方式，這些解決方案也已順利產生經過證明且可測量的成效。這比單純銷售軟體更有價值。

協助您成功是我們的目標

我們將您的成就視為我們的業務核心。從產品發想到部署，我們瞭解您需要能夠運作良好的 IT 解決方案，並與現有投資緊密結合；您需要持續的支援以及部署後訓練，並需要改與容易合作的對象往來。到了最後，您的成功就是我們的成就。

我們的解決方案

- 身分與存取治理
- 存取管理
- 安全性管理
- 系統與應用程式管理
- 工作量管理
- 服務管理

聯絡銷售支援

若有關於產品、價格及功能等方面的問題，請聯絡當地合作夥伴。如果您無法聯絡合作夥伴，請聯絡我們的銷售支援團隊。

全球：	www.netiq.com/about_netiq/officelocations.asp
美國和加拿大：	1-888-323-6768
電子郵件：	info@netiq.com
網站：	www.netiq.com

聯絡技術支援

若有關於特定產品的問題，請聯絡我們的技術支援團隊。

全球：	www.netiq.com/support/contactinfo.asp
北美與南美：	1-713-418-5555
歐洲、中東和非洲：	+353 (0) 91-782 677
電子郵件：	support@netiq.com
網站：	www.netiq.com/support

聯絡文件支援

我們的目標是提供符合您需求的文件。若您有任何改善建議，請按一下 HTML 文件版本任何頁面底部的「新增備註」，HTML 文件版本的張貼網址是：www.netiq.com/documentation。您也可以將電子郵件寄至 Documentation-Feedback@netiq.com。我們重視您的意見並期待您提出建議。

聯絡線上使用者社群

Qmunity (NetIQ 線上社群) 是一個協同網路，將您與使用者和 NetIQ 專家連接起來。透過提供更多立即的資訊、有用資源的實用連結以及諮詢 NetIQ 專家的管道，Qmunity 協助確保您精通必要知識，以充分發揮您所仰賴之 IT 投資的全部潛力。如需詳細資訊，請造訪 http://community.netiq.com。

1 規劃安裝 Identity Console

本章解釋安裝 Identity Console 的系統要求和先決條件。由於 Identity Console 可以同時作為 Docker 容器或獨立應用程式執行，因此請參閱兩種安裝類型個別的系統要求和先決條件部分。

附註：Identity Console 支援 eDirectory 9.2.4 HF2、Identity Manager Engine 4.8.3 HF2 和其個別更新版本。在使用 Identity Console 之前，您必須升級您的 eDirectory 和 Identity Manager Engine 例項。

Docker 安裝的系統要求和先決條件

本節說明將 Identity Console 安裝為 Docker 容器的系統要求和先決條件。

系統要求

因為 Identity Console 可以 Docker 容器形式執行，如需安裝 Identity Console 的系統要求和支援平台的詳細資訊，請參閱 Docker 文件。

先決條件

- ❒ 安裝 Docker 20.10.9-ce 或更新版本。如需如何安裝 Docker 的詳細資訊，請參閱 Docker 文件。
- ❒ 您必須取得 pkcs12 伺服器證書，並且有私密金鑰以加密 / 解密 Identity Console 伺服器和後端伺服器之間交換的資料。此伺服器證書是用來保護 HTTP 連線。您可以使用由任何外部 CA 產生的伺服器證書。如需詳細資訊，請參閱建立伺服器證書物件。伺服器證書應該包含帶有伺服器 IP 位址的「標題備用名稱」和 Identity Console 的 DNS。一旦建立伺服器證書物件，您必須以 .pfx 格式將其輸出。

- ❒ 您必須取得所有樹狀結構 .pem 格式的 CA 證書，才能驗證在先前步驟中取得的伺服器證書的 CA 簽名。這個 rootCA 證書也能確保在用戶端與 Identity Console 伺服器之間建立安全的 LDAP 通訊。例如，您可以從 /var/opt/novell/eDirectory/data/SSCert.pem 取得 eDirectory CA 證書 (SSCert.pem)。
- ❒ (選用) 使用單一 SSO 提供者 (OSP)，您可以為您的使用者啟用 Identity Console 入口網站的單一登入驗證。在安裝 Identity Console 之前，您必須安裝 OSP。若要設定 OSP for Identity Console，請遵循畫面提示，並提供組態參數的必要值。如需詳細資訊，請參閱「正在部署 OSP 容器」(第 20 頁)。若要向現有的 OSP 伺服器註冊 Identity Console，您必須將以下內容新增至 /opt/netiq/idm/apps/tomcat/conf/ 資料夾的 ism-configuration.properties 檔案中：

  ```
  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
  ```

  附註：透過 OSP，您僅可連線至單一 eDirectory 樹狀結構，因為 OSP 不支援多個 eDirectory 樹狀結構。
- ❒ 確定 /etc/hosts 中，有適用於主機機器且具有完全合格的主機名稱的適當 DNS 項目。
- ❒ 如果您想要在 Edge 瀏覽器中使用 Identity Console，您必須下載最新版本的 Microsoft Edge 才能取得完整功能。

附註：在 Mozilla Firefox 中使用 Identity Console 時，作業可能會失敗，並且有「來源不相符」錯誤訊息。若要進行疑難排解，請執行下列步驟：

1. 將 Firefox 更新為最新版本。
2. 在 [Firefox URL] 欄中指定 about:config，然後按下 Enter。
3. 搜尋 Origin。
4. 連按兩下 network.http.SendOriginHeader，然後將其值變更為 1。

設定環境

您可能需要建立包含特定參數的組態檔案。如果您想使用 OSP 設定 Identity Console，則必須在組態檔案中指定 OSP 特定參數。例如，使用 OSP 參數建立下列 edirapi.conf 檔案：

附註：您必須在 osp-redirect-url 欄位中提供您的 eDirectory 網路樹名稱。

```
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"
```

如果您想要在未使用 OSP 的情況下設定 Identity Console，請如下所示建立組態檔案，不使用 OSP 參數：

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

附註：當您要使用多個 eDirectory 樹狀結構來設定 Identity Console 時，您可以跳過「*ldapserver*」、「*ldapuser*」和「*ldappassword*」參數並建立組態檔案。

表格 1-1 *組態檔案中組態參數的描述*

組態參數	描述
listen	指定 9000 作為 Identity Console 伺服器在容器內的監聽埠。
ldapserver	指定 eDirectory 主機伺服器 IP 和連接埠號碼。
ldapuser	指定 eDirectory 使用者的使用者名稱。此參數是作為身分證明使用，以在 OSP 登入時，使用已代理授權控制來初始化對 eDirectory 的 LDAP 呼叫。LDAP 使用者必須有 eDirectory 樹狀結構的監督者權限。
ldappassword	指定 LDAP 使用者的密碼。
pfxpassword	指定 pkcs12 伺服器證書檔案的密碼。

組態參數	描述
ospmode	指定 true 以整合 OSP 與 Identity Console。如果您將此設定為 false，則 Identity Console 會使用 LDAP 登入。
osp-token-endpoint	此 URL 用來從 OSP 伺服器提取特定屬性，以驗證驗證記號的有效性 .
osp-authorize-url	使用者使用此 URL 來提供身分證明以取得驗證記號 .
osp-logout-url	使用此 URL 來終止使用者與 OSP 伺服器之間的工作階段 .
osp-redirect-url	在授與驗證記號之後，OSP 伺服器會將使用者重新導向至此 URL. **附註：**在設定 Identity Console 時，請確保指定小寫的 eDirectory 樹狀結構名稱。如果未指定小寫的樹狀結構名稱，則登入 Identity Console 會失敗。
osp-client-id	指定向 OSP 登錄 Identity Console 時所提供的 OSP 用戶端 ID.
ospclientpass	指定向 OSP 登錄 Identity Console 時所提供的 OSP 用戶端密碼 .
ospcert	指定 OSP 伺服器其 CA 證書的位置 .
bcert	指定 Identity Console 的 CA 證書位置。
loglevel	指定您要包含在記錄檔案中的層級。此參數可以設為 「嚴重」、「錯誤」、「警告」、「資訊」。
check-origin	如果這設為 true，則 Identity Console 伺服器會比較要求的原始值。可用選項為 true 或 false。即使在使用 DNS 組態時將 *check-origin* 參數值設為 false，*origin* 參數仍為必要。
origin	Identity Console 會比較要求的原始值與此欄位中指定的值。 **附註：**自 Identity Console 1.4 開始，此參數與 *check-origin* 參數無關，但如果使用 DNS 組態，則此參數為必要。
maxclients	同時可存取 IDConsole 用戶端的上限。超出此限制的任何其他客戶必須在佇列中等候。

附註：

- 只有在您計畫將 OSP 和 Identity Console 整合時，才能使用 ospmode 設定參數。

- 如果 Identity Applications (Identity Apps) 已在 Identity Manager 設定的叢集模式中設定，您必須在組態檔案的 osp-token-endpoint、osp-authorize-url 和 osp-logout-url 欄位中提供負載平衡器伺服器的 DNS 名稱。如果您在這些欄位中提供 OSP 伺服器詳細資訊，則 Identity Console 登入將會失敗。
- 如果 Identity Console 使用與 Identity Apps 和 Identity Reporting 相同的 OSP 例項進行設定，則單一登入 (驗證服務) 會在您登入 Identity Console 入口網站時生效。
- OSP HTTPS URL 應使用包含 Identity Console 1.4 或更新版本的 2048 位元 (或更高位元) 金鑰的證書進行驗證。
- 如果您想要限制從不同網域對於 Identity Console 入口網站的存取權，請將 samesitecookie 參數設定為嚴格。如果您想允許從不同網域存取 Identity Console 入口網站，請將 samesitecookie 參數設定為鬆懈。如果在設定過程中未指定參數，則預設情況下將遵循瀏覽器設定。

準備好組態檔案之後，請繼續部署容器。如需詳細資訊，請參閱 「將 Identity Console 部署為 Docker 容器」(第 20 頁)。

獨立安裝 (非 Docker) 的系統要求和先決條件

系統要求

本節說明安裝獨立 Identity Console 的系統要求和先決條件。

類別	最低要求
處理器	1.4 GHz 64 位元
記憶體	2 GB
磁碟空間	Linux 上 200 MB

類別	最低要求
支援的瀏覽器	◆ 最新版本的 **Microsoft Edge** ◆ 最新版本的 **Google Chrome** ◆ 最新版本的 **Mozilla Firefox** **附註：**在 Mozilla Firefox 中使用 Identity Console 時，作業可能會失敗，並且有「來源不相符」錯誤訊息。若要進行疑難排解，請執行下列步驟： **1** 將 Firefox 更新為最新版本。 **2** 在 [Firefox URL] 欄中指定 about:config，然後按下 Enter。 **3** 搜尋 Origin。 **4** 連按兩下 network.http.SendOriginHeader，然後將其值變更為 1。

支援的作業系統	◆ **已認證**：
	  ◆ SUSE Linux Enterprise Server (SLES) 15 SP1、SP2 和 SP3
	  ◆ SUSE Linux Enterprise Server (SLES) 12 SP1、SP2、SP3、SP4 和 SP5
	  ◆ Red Hat Enterprise Linux (RHEL) 7.8、7.9、8.0、8.1、8.2、8.3、8.4 和 8.5
	  ◆ openSUSE 15.1 與 15.2
	◆ **支援：**支援上述認證作業系統的支援套件更新版本。

證書	◆ 您必須取得有可用於加密 / 解密用戶端與 Identity Console 伺服器之間交換資料時所需的 pkcs12 伺服器證書的私密金鑰。此伺服器證書是用來保護 HTTP 連線。您可以使用由任何外部 CA 產生的伺服器證書。如需詳細資訊，請參閱建立伺服器證書物件。伺服器證書應該包含帶有伺服器 IP 位址的「標題備用名稱」和 Identity Console 的 DNS。一旦建立伺服器證書物件，您必須以 .pfx 格式將其輸出。 ◆ 您必須針對所有樹狀結構取得 .pem 格式的 CA 證書，才能驗證在先前步驟中取得的伺服器證書的 CA 簽名。這個 rootCA 證書也能確保在用戶端與 Identity Console 伺服器之間建立安全的 LDAP 通訊。例如，您可以從 /var/opt/novell/eDirectory/data/SSCert.pem 取得 eDirectory CA 證書 (SSCert.pem)。

準備好後，請繼續安裝 Identity Console。如需詳細資訊，請參閱 「正在部署獨立 Identity Console」(第 24 頁)。

(選用) OSP 設定的先決條件

使用單一 SSO 提供者 (OSP)，您可以為您的使用者啟用 Identity Console 入口網站的單一登入驗證。在安裝 Identity Console 之前，您必須安裝 OSP。若要設定 OSP for Identity Console，請遵循畫面提示，並提供組態參數的必要值。如需詳細資訊，請參閱 「正在部署 OSP 容器」(第 20 頁)。若要向現有的 OSP 伺服器註冊 Identity Console，您必須將以下內容新增至 /opt/netiq/idm/apps/tomcat/conf/ 資料夾的 ism-configuration.properties 檔案中：

```
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
```

附註：

- 如果您是第一次安裝 OSP，請為**以 eDir API 設定 OSP** 指定選項 'y'，然後遵循畫面上的提示向 OSP 註冊 Identity Console。
- 在設定 Identity Console 時，請確保指定小寫的 eDirectory 樹狀結構名稱。如果未指定小寫的樹狀結構名稱，則登入 Identity Console 會失敗。
- 透過 OSP，您僅可連線至單一 eDirectory 樹狀結構，因為 OSP 不支援多個 eDirectory 樹狀結構。

工作站的系統要求和先決條件

系統要求

本節說明執行工作站 Identity Console 的系統要求和先決條件。

類別	最低要求
處理器	1.5 GHz 64 位元
記憶體	2 GB
磁碟空間	Windows 上有 1 GB

類別	最低要求
支援的作業系統	◆ 已認證： ◆ Windows Server 2016 ◆ Windows Server 2019 ◆ Windows Server 2022 ◆ Windows 10 ◆ Windows 11
證書	◆ 您必須取得 pfx 格式的伺服器證書以在 Identity Console 用戶端與 REST 伺服器之間交換資料。此伺服器證書必須一律命名為 keys.pfx。如需詳細資訊，請參閱建立伺服器證書物件 (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/b1j4tpo3.html#b1j4u0cm)。 ◆ 您必須針對所有樹狀結構取得 .pem 格式的 CA 證書，才能驗證在先前步驟中取得的伺服器證書的 CA 簽名。這個根 CA 證書也能確保在用戶端與 Identity Console 伺服器之間建立安全的 LDAP 通訊。 例如，您可以從 /var/opt/novell/eDirectory/data/SSCert.pem 取得適用於 Linux 的 eDirectory CA 證書 (SSCert.pem)。 從 <eDirectory install Location>\NetIQ\eDirectory\DIBFiles\CertServ\SSCert.pem 取得適用於 Windows 的 eDirectory CA 證書 SSCert.pem 。

準備好後，請繼續部署 Identity Console。如需詳細資訊，請參閱 「將 Windows 上的 Identity Console 作為工作站」(第 26 頁)。

RPM 簽名驗證

使用下列步驟來執行 RPM 簽名驗證：

1. 瀏覽至解壓縮組建的資料夾。

   例如：<untarred location of Identity Console>/IdentityConsole_150_Linux/license/MicroFocusGPGPackageSign.pub。

2. 執行下列指令以匯入公用金鑰：

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

3. (選擇性) 執行下列指令以驗證 RPM 簽名：rpm --checksig -v <RPM Name>

   例如：

   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 部署 Identity Console

本章說明部署 Identity Console 的程序與安全性建議。若要進行部署準備工作，請檢閱 第 1 章「規劃安裝 Identity Console」(第 9 頁) 中提供的先決條件和系統要求。

安全性建議

- Docker 容器預設沒有任何資源限制。這會將主機核心所提供所有 CPU 與記憶體資源的存取權提供給每一個容器。您也必須透過設定容器可用資源數量的限制，來確保一個執行中的容器不會耗用較多的資源，而使其他執行中的容器缺乏資源。
  - 透過在 Docker run 命令上使用 --memory 旗標，Docker 容器可確定固定限制適用於容器所使用的記憶體。
  - 透過在 Docker run 命令上使用 --cpuset-cpus 旗標，Docker 容器可確定限制會套用到執行中容器使用的 CPU 數量。
- --pids-limit 應設定為 300，以限制任可指定時間在容器內繁衍的核心執行緒數目。這可防止 DoS 攻擊。
- 您必須在 Docker run 命令上使用 --restart 旗標，將失敗時容器重新啟動規則設定為 5。
- 容器啟動後，一旦狀態顯示成良好，您就只能使用容器。若要檢查容器的健康情況，請執行以下指令：

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

- Docker 容器一律會以非 root 使用者身分 (nds) 啟動。作為額外的安全措施，讓使用者命名空間能夠重新對應到精靈，以防止來自容器內的權限提升攻擊。如需使用者命名空間對應的詳細資訊，請參閱「以使用者命名空間隔離容器」。

將 Identity Console 部署為 Docker 容器

本節包含以下程序：

正在部署 OSP 容器

部署 OSP 容器需要執行的步驟如下：

1. 登入軟體授權和下載 (https://sld.microfocus.com/)，然後瀏覽至 「軟體下載」頁面。
2. 選取下列項目：
   - 產品：eDirectory
   - 產品名稱：eDirectory per User Sub SW E-LTU
   - 版本：9.2
3. 下載檔案：IdentityConsole_<version>_Containers_tar.zip。
4. 將下載的檔案解壓縮至資料夾。
5. 根據您的要求修改靜音屬性檔案。下面顯示了一個範例靜音屬性檔案：

```
# 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
```

附註：若要在使用靜默屬性 (DOS 文字) 檔案時避免空間限制，您必須使用 dos2unix 工具將 DOS 文字檔案轉換至 UNIX 格式。執行下列指令，將文字檔從 DOS 行尾轉換至 Unix 行尾：

dos2unix filename

例如，

dos2unix samplefile

---

6 使用 iManager 產生伺服器證書 (cert.der) 並匯入至 KeyStore (tomcat.ks)。將靜音屬性檔案和 KeyStore ((tomcat.ks)) 複製到任何目錄中。例如，/data。執行下列步驟來建立伺服器證書，並將其匯入至 KeyStore：

6a 執行下列指令來建立 KeyStore (tomcat.ks)。產生金鑰，確保 CN 名稱或機器的完全合格主機名稱為 IP 位址。

```
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 執行下列指令來建立證書登記申請。例如，cert.csr。

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

6c 將此 cert.csr 傳送給 iManager 並取得 osp.der 伺服器證書。確保您將金鑰類型選取為「自訂」、將金鑰使用選項選取為資料加密、鍵密碼，以及數位簽名與證書的標題備用名稱欄位，以包含 OSP 伺服器的 IP 位址或主機名稱。如需詳細資訊，請參閱建立伺服器證書物件。

6d 執行下列指令來匯入 CA 證書 (SSCert.der 和伺服器證書 (cert.der) 至 tomcat.ks KeyStore。

```
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 執行以下指令以載入 OSP 影像：

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

8 使用以下指令部署容器：

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

例如，

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

正在將 Identity Console 部署為 Docker 容器

本節解釋部署 Identity Console 作為 Docker 容器的程序：

附註：此程序中提及的組態參數、範例值和範例僅供參考。您必須確保不要直接在線上環境中使用。

---

1 登入 SLD：軟體授權和下載 (https://sld.microfocus.com/)，然後瀏覽至 「軟體下載」頁面。

2 選取下列項目：

- 產品：eDirectory
- 產品名稱：eDirectory per User Sub SW E-LTU
- 版本：9.2

3 下載檔案：IdentityConsole_<version>_Container.tar.zip。

4 影像必須載入到本機 Docker 登錄中。使用以下命令擷取並載入 IdentityConsole_<version>_Containers.tar.gz 檔案：

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

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

5 使用以下指令建立 Identity Console Docker 容器：

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

例如，

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

附註：

- 您可以接受 EULA，作法是將 ACCEPT_EULA 環境變數設為 'Y'。您可以在啟動容器時，透過使用 Docker 中的 -it 選項為互動模式建立命令，以從提示畫面上接受 EULA。
- 上述指令中的 --volume 參數，將會建立磁碟區以儲存組態和記錄資料。在這個案例中，我們建立了稱為 IDConsole-volume 的範例磁碟區。

6 使用以下指令，將伺服器證書檔案從本地檔案系統複製到容器作為 /etc/opt/novell/eDirAPI/cert/keys.pfx。如需有關建立伺服器證書的詳細資訊，請參閱「先決條件」(第 9 頁)：

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

例如，

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

連線至多個 eDirectory 樹狀結構時，您必須確保針對所有連線的樹狀結構取得至少一個 keys.pfx 伺服器證書。

7 使用以下指令，將 CA 證書檔案 (.pem) 從本地檔案系統複製到容器作為 /etc/opt/novell/eDirAPI/cert/SSCert.pem。如需有關取得 CA 證書的詳細資訊，請參閱「先決條件」(第 9 頁)：

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

例如，

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

如果使用者必須連線至多個 eDirectory 樹狀結構，請參閱章節：「具有 Identity Console 作為 Docker 的多個樹狀結構」(第 24 頁)

8 根據您的要求修改組態檔案，並使用下列指令將組態檔案 (edirapi.conf) 從本地檔案系統複製到容器中，作為 /etc/opt/novell/eDirAPI/conf/edirapi.conf：

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

例如，

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

9 使用以下指令啟動 Docker 容器：

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

例如，

```
docker start identityconsole-container
```

附註：您可以在 var/lib/docker/volumes/<volume_name>/_data/eDirAPI/var/log 目錄中找到下列記錄檔案：

- edirapi.log- 這是用於記錄 edirapi 中的不同事件和除錯問題。
- edirapi_audit.log - 這是用於記錄 edirapi 的稽核事件。記錄會遵循 CEF 稽核格式。
- container-startup.log - 這是用於擷取 Identity Console Docker 容器的安裝記錄。

具有 Identity Console 作為 Docker 的多個樹狀結構

Identity Console 允許使用者透過取得樹狀結構的個別 CA 證書來連線至多個樹狀結構。

例如，如果連接線至三個 eDirectory 樹狀結構，則您必須將所有三個 CA 證書複製到 Docker 容器中：

```
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
```

執行下列指令以重新啟動 Identity Console：

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

正在部署獨立 Identity Console

- 「正在部署獨立 Identity Console (非 Docker)」(第 24 頁)
- 「具有獨立 Identity Console 的多個樹狀結構」(第 26 頁)

正在部署獨立 Identity Console (非 Docker)

本節說明部署獨立 Identity Console 的程序：

1. 登入 SLD：軟體授權和下載 (https://sld.microfocus.com/)，然後瀏覽至 「軟體下載」頁面。
2. 選取下列項目：
   - 產品：eDirectory
   - 產品名稱：eDirectory per User Sub SW E-LTU
   - 版本：9.2
3. 下載最新的 Identity Console 版次。
4. 將下載的檔案解壓縮到資料夾中。
5. 打開外圍程序並瀏覽至擷取 Identity Console 組建所在的資料夾。

6 以根或根相等的使用者身分登入時，執行以下指令：

```
./identityconsole_install
```

7 閱讀「簡介」內容，然後按 **ENTER**。

8 按一下 'Y' 以接受授權合約。這將會在您的系統上安裝所有必要的 RPM。

9 輸入 Identity Console 伺服器的主機名稱 (FQDN)/IP 位址。

10 輸入 Identity Console 要監聽的連接埠號碼。預設值為 9000。

11 輸入整合 OSP 與 Identity Console 或要使用 LDAP 登入的選項。

12 如果要整合 OSP 與 Identity Console：

1. 使用 LDAPS 連接埠號碼輸入 eDirectory/Identity Vault 伺服器的網域名稱 /IP 位址。

   例如，

   192.168.1.1:636

2. 輸入 eDirectory/Identity Vault 使用者名稱。

   例如，

   cn=admin,ou=org_unit,o=org

3. 輸入 eDirectory/Identity Vault 密碼。
4. 再次輸入 eDirectory/Identity Vault 密碼以確認密碼。
5. 輸入 OSP 伺服器網域名稱 /IP 位址和 SSO 伺服器 SSL 連接埠號碼。
6. 輸入 OSP 用戶端 ID。
7. 輸入 OSP 用戶端密碼。
8. 輸入 eDirectory/Identity Vault 網路樹名稱。

13 輸入可信的根證書 (SSCert.pem) 路徑，包含資料夾。

例如，

/home/Identity_Console/certs

附註：使用者必須確保未在證書資料夾內建立子目錄。

---

14 輸入伺服器證書 (keys.pfx) 路徑，包含檔案名稱。

例如，

/home/Identity_Console/keys.pfx

15 輸入伺服器證書密碼。若要確認您輸入的密碼正確，請重新輸入伺服器證書密碼。已起始安裝。

附註：您可以在 /var/opt/novell/eDirAPI/log 目錄中找到下列記錄檔案：

- edirapi.log- 這是用於記錄 edirapi 中的不同事件和除錯問題。
- edirapi_audit.log - 這是用於記錄 edirapi 的稽核事件。記錄會遵循 CEF 稽核格式。
- identityconsole_install.log - 這是用於擷取 Identity Console 的安裝紀錄。

Identity Console 程序啟動 / 停止的記錄可以在 /var/log/messages 檔案中找到。

附註：NetIQ 建議在相同機器上安裝 Identity Console 和 eDirectory 時，且機器至少有一個可用的 eDirectory 例項。

具有獨立 Identity Console 的多個樹狀結構

當連線至多個 eDirectory 樹狀結構時，您必須確保取得樹狀結構的個別 CA 證書。

例如，如果您連線至三個 eDirectory 樹狀結構，則必須將所有三個 CA 證書複製到 etc/opt/novell/eDirAPI/cert/ 目錄中：

```
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
```

執行下列其中一個指令以重新啟動 Identity Console：

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

或者

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

將 Windows 上的 Identity Console 作為工作站

Identity Console 可以在 Windows 上啟動作為工作站，並需要執行中的 REST 服務。因此，當啟動時，eDirAPI 程序將在 edirapi.exe 命令提示中執行。如果關閉此 edirapi.exe 終端機，則 Identity Console 將無法正常運作。

下列程序說明如何在 Windows 上執行 Identity Console。

1. 登入 SLD 軟體授權和下載 (https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0)，瀏覽至「軟體下載」頁面。
2. 選取下列項目：
   - 產品：eDirectory
   - 產品名稱：eDirectory per User Sub SW E-LTU
   - 版本：9.2
3. 下載檔案 IdentityConsole_<version>_workstation_win_x86_64.zip。
4. 將下載的 IdentityConsole_<version>_workstation_win_x86_64.zip 檔案解壓縮至資料夾。
5. 瀏覽至解壓縮的資料夾：IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert，然後複製可信的根 CA SSCert.pem 和伺服器證書 keys.pfx。

   若要取得證書，請參閱章節：「工作站的系統要求和先決條件」(第 15 頁)

   如果使用者必須連線至多個 eDirectory 樹狀結構，請參閱章節：「具有 Identity Console 作為工作站的多個樹狀結構」(第 27 頁)

附註：伺服器證書名稱必須一律為 keys.pfx。

6 瀏覽至解壓縮組建的資料夾並按兩下檔案 run.bat (Windows 批次檔案)。

7 在命令提示中輸入伺服器證書 (keys.pfx) 密碼。

eDirAPI 程序終端機 (edirapi.exe) 開始執行，並顯示 Identity Console 登入頁面。

附註：

- 如果 eDirAPI 程序終端機 (edirapi.exe) 已執行，請從已解壓縮組建的資料夾執行 identityconsole.exe。
- 使用者可以在：\IdentityConsole_150_workstation_win_x86_64\eDirAPI\log 中找到下列記錄

  edirapi.log - 這是用於記錄 edirapi 中的不同事件和除錯問題。

  edirapi_audit.log - 這是用於記錄 edirapi 的稽核事件。記錄會遵循 CEF 稽核格式。
- 工作站模式不支援 OSP 登入。
- Identity Console 工作站僅在 9000 連接埠 上監聽。請勿修改 edirapi_win.conf 檔案。

具有 Identity Console 作為工作站的多個樹狀結構

Identity Console 允許使用者透過取得樹狀結構的個別 CA 證書來連線至多個樹狀結構。

1 關閉 Identity Console 工作站和 eDirAPI 終端機。

2 將 CA 證書 SSCert.pem 複製到下列位置：
IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert。

例如，如果您要連線至三個 eDirectory 樹狀結構，請將 CA 證書分別複製為 SSCert1.pem、SSCert2.pem 和 SSCert3.pem。

3 瀏覽至解壓縮組建的資料夾並按兩下檔案 run.bat (Windows 批次檔案)。

4 在終端機提示中輸入 keys.pfx 密碼，並登入所需的 eDirectory 樹狀結構。

停止和重新啟動 Identity Console

- 「正在停止並將 Identity Console 重新啟動為 Docker 容器」(第 27 頁)
- 「正在停止並重新啟動獨立 Identity Console」(第 28 頁)
- 「關閉並重新啟動 Identity Console 工作站」(第 28 頁)

正在停止並將 Identity Console 重新啟動為 Docker 容器

若要停止 Identity Console，請執行以下指令：

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

若要重新啟動 Identity Console，請執行以下指令：

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

若要啟動 Identity Console，請執行以下指令：

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

正在停止並重新啟動獨立 Identity Console

若要停止 Identity Console，請執行下列其中一個指令：

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

或者

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

若要重新啟動 Identity Console，請執行下列其中一個指令：

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

或者

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

若要啟動 Identity Console，請執行下列其中一個指令：

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

或者

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

關閉並重新啟動 Identity Console 工作站

若要關閉應用程式和程序，請遵循下列程序：

1. 關閉 Identity Console 桌面視窗應用程式。
2. 透過關閉 eDirAPI 程序終端機來停止 eDirAPI 程序。

若要重新啟動 Identity Console 工作站，請瀏覽至解壓縮組建的資料夾並按兩下檔案 run.bat (Windows 批次檔案)。

附註：如果 eDirAPI 程序終端機已執行，請從已解壓縮組建的資料夾執行 identityconsole.exe 以重新啟動 Identity Console 工作站。

管理資料持續性

除了 Identity Console 容器以外，也會建立適用於資料持續性的磁碟區。若要使用舊容器 (使用磁碟區) 的組態參數，請執行下列步驟：

1. 使用以下指令來停止目前的 Docker 容器：

   ```
   docker stop identityconsole-container
   ```

2 使用儲存在 Docker 磁碟區 (edirapi-volume-1) 中舊容器的應用程式資料，來建立第二個容器：

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

3 使用以下指令啟動第二個容器：

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

4 (選用) 現在可以使用以下指令移除第一個容器：

```
docker rm identityconsole-container
```

在 Azure Kubernetes Services 中部署 Identity Console

Azure Kubernetes Service (AKS) 是受管理的 Kubernetes 服務，可讓您部署和管理叢集。本節包含以下程序：

在 AKS 叢集中部署 Identity Console

本節說明下列程序以在 AKS 叢集中部署 Identity Console：

建立 Azure Container Registry (ACR)

Azure Container Registry (ACR) 是一個 Azure 的私人登錄，適用於 Docker 容器映像。

如需詳細步驟，請參閱「建立容器登錄 - 入口網站」中的使用 Azure 入口網站來建立 Azure 容器登錄，或執行下列步驟以建立 Azure Container Registry (ACR)：

1. 登入 Azure 入口網站。
2. 前往 **「建立資源」** > **「容器」** > **「容器登錄」**。
3. 在 **「基本」**索引標籤中，請指定 「資源群組」和 「登錄名稱」的值。在 Azure 中，登錄名稱必須為唯一並包含最少 5 個和最大 50 個英數字元。

   接受其餘設定的預設值。
4. 按一下 「檢閱 + 建立」。
5. 按一下 「建立」。
6. 登入 Azure CLI，執行以下指令以登入 Azure Container Registry

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

例如：

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

7. 使用指令來擷取 Azure Container Registry 的登入伺服器：

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

   例如：

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

8. 使用下列指令來使用 ACR 登入伺服器 (registryname.azureacr.io) 標記 Identity Console 的本地映像：

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

   例如，

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

9. 將標記的映像推送至登錄。

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

   例如，

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

10. 使用指令來擷取登錄中映像清單：

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

設定 Kubernetes 叢集

使用 Azure 入口網站或 CLI 來建立 Kubernetes 服務資源。

如需使用節點在 Azure 中建立 Kubernetes 服務資源的詳細步驟，請參閱 Azure 快速入門中的建立 AKS 叢集。

附註：

- 確保選取 「Azure CNI」作為網路。
- 選取現有的虛擬網路 (其中已在子網路中部署 eDirectory 伺服器)。
- 當 Identity Console 映像可用時，請選取現有的容器登錄。

建立標準 SKU 公用 IP 位址

Kubernetes 叢集資源群組下的公用 IP 位址資源會作為應用程式的負載平衡器 IP。

如需詳細步驟，請參閱建立公用 IP 位址 - 入口網站中的使用 Azure 入口網站建立公用 IP 位址。

設定 Cloud Shell 並連線至 Kubernetes 叢集

使用在 Azure 入口網站中適用於所有作業的 Cloud Shell。

若要在 Azure 入口網站中設定 Cloud Shell，請參閱 Bash – 快速入門中的啟動 Cloud Shell 一節，或執行下列步驟來設定 Cloud Shell 並連線至 Kubernetes 叢集：

1. 在 Azure 入口網站，按一下打開  按鈕來開啟 Cloud Shell。

   附註：若要管理 Kubernetes 叢集，請使用 Kubernetes 指令列用戶端 kubectl。如果您使用 Azure Cloud Shell，則已安裝 kubectl。

2. 使用下列指令，設定 kubectl 以連線至 Kubernetes 叢集：

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

   例如，

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

3. 使用指令，驗證叢集節點的清單：

   ```
   kubectl get nodes
   ```

正在部署應用程式

若要部署 Identity Console，您可使用 idc-services.yaml、idc-statefulset.yaml、idc-storageclass.yaml 和 idc-pvc.yaml 範例檔案。

您也可以根據需要建立自己的 yaml 檔案。

1. 使用下列指令以建立儲存類別資源：

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

   例如，

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

   (選擇性) 如需如何使用 Azure 檔案共用來動態建立和使用持續性磁碟區，請參閱在 Azure Kubernetes Service (AKS) 中使用 Azure 檔案共用來動態建立和使用持續性磁碟區

   範例儲存類別資源檔案如下所示：

```
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
~
```

儲存類別資源可啟用動態儲存佈建。這是用來定義建立 Azure 檔案共用的方式。

2. 使用下列指令以檢視儲存類別的詳細資料：

   ```
   kubectl get sc
   ```

3. 使用 idc-pvc.yaml 檔案建立 pvc 資源：

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

   例如，

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

   範例 pvc 資源檔案如下所示：

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

   持續性磁碟區宣告資源會建立檔案共用。持續性磁碟區宣告 (PVC) 使用儲存類別物件來動態佈建 Azure 檔案共用。

4. 將 edirapi.conf、CA 證書和伺服器證書上傳到 Cloud Shell。

   按一下 Cloud Shell 上的「**上傳 / 下載檔案**」按鈕圖示  並上傳 edirapi.conf、SSCert.pem 和 keys.pfx 檔案。

附註：edirapi.conf 有參數「origin」。在這裡，我們必須佈建存取 Identity Console 應用程式的 IP 位址。(使用 「建立標準 SKU 公用 IP 位址」(第 31 頁) 一節中建立的 IP 位址。)

Identity Console 部署需要伺服器證書 (keys.pfx)。

建立伺服器證書時，請確保在標題備用名稱中提供有效 DNS 名稱。

建立有效 DNS 名稱的步驟：

使用 StatefulSet 部署的一般 Pod 有 DNS 名稱，如下所示 - {statefulsetname}-{ordinal}。{servicename}。{namespace}.svc.cluster.local

- 如果 idconsole-statefulset.yaml 檔案中的 StatefulSet 名稱為 idconsole-app，則 statefulsetname = idconsole-app
- 若是第一個 Pod，則 ordinal = 0
- 如果將 idconsole -statefulset.yaml 檔案中的 serviceName 定義為 idconsole，則 serviceName = idconsole
- 若是預設命名空間，則 namespace=default

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

---

5. 在 Kubernetes 叢集中建立 configmap 資源，以儲存組態檔案和證書。

   在執行指令之前，請確保目錄中有檔案 (edirapi.conf、SSCert.pem 和 keys.pfx)。

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

   例如，

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

6. 使用 kubectl 說明指令以檢視 configmap 物件的詳細資料：

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

   例如，

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

7. 建立 StatefulSet 資源以部署容器。

   執行下方指令以部署容器：

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

   例如，

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

   範例 StatefulSet 資源檔案如下所示：

```
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. 執行下列指令以驗證已部署 pod 的狀態：

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

9. 建立 loadBalancer 類型的服務資源。

   yaml 檔案中指定的服務類型為 loadBalancer。

   使用下方指令以建立服務資源：

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

   例如，

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

   範例服務資源檔案如下所示：

```
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
```

使用下方指令以檢查 EXTERNAL-IP 位址 (或 loadBalancerIP)：

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

10. 使用 EXTERNAL-IP (或 loadBalancerIP 位址) 啟動 URL。

    例如，

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

修改伺服器證書

本節提供如何在 Docker 容器和獨立 Identity Console 中修改伺服器證書的相關資訊。

- 「在 Docker 容器中修改伺服器證書」(第 35 頁)
- 「在獨立 Identity Console 中修改伺服器證書」(第 36 頁)

在 Docker 容器中修改伺服器證書

執行下列步驟以在 Docker 容器中修改伺服器證書：

1 執行下列指令以在容器的任一位置中複製新的伺服器證書。

例如，

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

2 使用下列指令以登入容器：

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

3 執行 NLPCERT 以將金鑰儲存為虛擬使用者：

```
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 使用指令以離開容器主控台：

```
exit
```

5 透過輸入下列內容以重新啟動容器：

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

在獨立 Identity Console 中修改伺服器證書

執行下列步驟以在獨立容器中修改伺服器證書：

1 執行 NLPCERT 以儲存金鑰：

```
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 重新啟動 Identity Console：

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

3 正在升級 Identity Console

本章說明將 Identity Console 升級至其最新版的程序。若要進行升級準備工作，請檢閱 第 1 章「規劃安裝 Identity Console」(第 9 頁) 中提供的先決條件和系統要求。

本節包含以下程序：

- 「將 Identity Console 升級為 Docker 容器」(第 37 頁)
- 「升級獨立 Identity Console (非 Docker)」(第 39 頁)
- 「正在升級 OSP 容器」(第 39 頁)

將 Identity Console 升級為 Docker 容器

當有新版的 Identity Console 影像時，管理員可以執行升級程序以部署含最新版 Identity Console 的容器。請確保將所有必要的應用程式相關資料，永久儲存在 Docker 磁碟區中，再執行升級。執行下列步驟以使用 Docker 容器升級 Identity Console：

1. 如同 中所述，從 (https://sld.microfocus.com/) 軟體授權和下載 「部署 Identity Console」(第 19 頁) 下載並載入最新版本的 Docker 映像，然後執行步驟以安裝最新版本的 Identity Console。
2. 載入最新的 Docker 影像之後，請使用以下指令來停止您目前的 Docker 容器：

   ```
   docker stop identityconsole-container
   ```

3. (選擇性) 備份共用磁碟區。
4. 透過執行下列指令，刪除現有 Identity Console 容器：

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

   例如，

   ```
   docker rm identityconsole-container
   ```

5. (選用) 透過執行下列指令，刪除已過時的 Identity Console Docker 影像：

   ```
   docker rmi identityconsole
   ```

6. 使用以下指令建立 Identity Console Docker 容器：

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

   例如：

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

附註：

- 您可以接受 EULA，作法是將 ACCEPT_EULA 環境變數設為 'Y'。您可以在啟動容器時，透過使用 Docker 中的 -it 選項為互動模式建立命令，以從提示畫面上接受 EULA。
- 上述指令中的 --volume 參數，將會建立磁碟區以儲存組態和記錄資料。在這個案例中，我們建立了稱為 IDConsole-volume 的範例磁碟區。

7 使用以下指令，將伺服器證書檔案從本地檔案系統複製到新建立的容器作為 /etc/opt/novell/eDirAPI/cert/keys.pfx：

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

例如，

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

連線至多個 eDirectory 樹狀結構時，您必須確保針對所有連線的樹狀結構至少複製一個 keys.pfx 伺服器證書。

8 使用以下指令，將 CA 證書檔案 (.pem) 從本地檔案系統複製到新建立的容器作為 /etc/opt/novell/eDirAPI/cert/SSCert.pem：

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

例如，

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

連線至多個 eDirectory 樹狀結構時，您必須確保針對所有連線的樹狀結構取得個別 CA 證書。例如，如果連線至三個 eDirectory 樹狀結構，則您必須將所有三個 CA 證書複製到 Docker 容器中：

```
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
```

附註：自 Identity Console 1.4 及更新版本，組態檔案 (edirapi.conf) 不會明確包含「*ldapuser*」、「*ldappassword*」和「*ldapserver*」參數。「*bcert*」參數值必須包括可信的根證書的目錄路徑。例如，bcert = "/etc/opt/novell/eDirAPI/cert/"。此外，「*origin*」參數與「*check-origin*」參數無關，並在使用 DNS 組態時為必要參數。

9 使用以下指令，將組態檔案 (edirapi.conf) 從本地檔案系統複製到新建立的容器作為 /etc/opt/novell/eDirAPI/conf/edirapi.conf：

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

例如，

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

10 使用以下指令啟動第二個容器：

```
docker start identityconsole-container
```

11 若要檢查執行容器的狀態，請執行下列指令：

```
docker ps -a
```

升級獨立 Identity Console (非 Docker)

本節說明升級獨立 Identity Console 的程序：

1 從軟體授權和下載 (https://sld.microfocus.com/) 中下載 IdentityConsole_<version>_Containers.tar.gz

2 登入 SLD，瀏覽至「軟體下載 SLD」頁面並按一下「下載」

3 瀏覽作法是選取產品：**eDirectory** > 產品名稱：**eDirectory per User Sub SW E-LTU** > 版本：**9.2**

4 下載最新的 Identity Console 版次。

5 使用下列指令來解壓縮下載的檔案：

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

6 瀏覽至解壓縮 Identity Console 版次的資料夾。

7 將所有要連線至 eDirectory 樹狀結構的可信的根證書複製到資料夾。若要將可信的根證書複製到資料夾，請執行下列指令：

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

例如，

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

8 執行以下指令：

```
./identityconsole_install
```

9 指定在**步驟 4** 中使用的可信的根證書資料夾路徑。

10 Identity Console 升級成功。

正在升級 OSP 容器

升級 OSP 容器需要執行的步驟如下：

1 從下載授權和下載 (https://sld.microfocus.com/) 下載並載入最新版本的 OSP 映像。

例如，

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

2 載入最新的 OSP 映像之後，請使用以下指令來停止您目前的 OSP 容器：

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

3 (選擇性) 備份共用磁碟區。

4 透過執行下列指令，刪除現有 OSP 容器：

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

例如，

```
docker rm OSP_Container
```

5 前往包含金鑰儲存區 (tomcat.ks) 和靜默屬性檔案的目錄、刪除現有的金鑰儲存區 (tomcat.ks) 並保留現有 OSP 資料夾。產生金鑰大小為 2048 的新金鑰儲存區 (tomcat.ks)。如需詳細資訊，請參閱 《Identity Console 安裝指南》的部署 OSP 容器一節中**步驟 4**。

6 使用以下指令部署容器：

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

例如，

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

4 解除安裝 Identity Console

本章描述解除安裝 Identity Console 的程序：

Docker 環境解除安裝程序

若要解除安裝 Identity Console Docker 容器，請執行以下步驟：

1. 停止 Identity Console 容器：

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

2. 執行以下指令以移除 Identity Console Docker 容器：

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

3. 執行以下指令以移除 Docker 影像：

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

4. 移除 Docker 容量：

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

附註：如果您移除容量，則資料也將從您的伺服器中移除。

獨立 Identity Console 的解除安裝程序 (非 Docker)

要解除安裝獨立 Identity Console，請執行以下步驟：

1. 導覽至安裝了 Identity Console 的電腦上的 /usr/bin 目錄。
2. 執行以下指令：

   ```
   ./identityconsoleUninstall
   ```

3. Identity Console 成功解除安裝。

附註：當在機器中安裝 eDirectory 或 NetIQ 產品時，使用者必須手動解除安裝 *nici* 和 *openssl*。