



Identity Console

安装指南

2022 年 9 月

法律声明

有关法律声明、商标、免责声明、担保、出口和其他使用限制、美国政府权限、专利政策以及 FIPS 合规性的信息，请参见 <https://www.netiq.com/company/legal>。

版权所有 © 2022 NetIQ Corporation。保留所有权利。

目录

关于本书和库	5
关于 NetIQ Corporation	7
1 计划安装 Identity Console	9
Docker 安装的系统要求和先决条件	9
系统要求	9
先决条件	9
设置环境	10
独立安装（非 Docker）的系统要求和先决条件	13
系统要求	13
（可选）OSP 配置的先决条件	14
工作站的系统要求和先决条件	15
系统要求	15
RPM 签名校验	16
2 部署 Identity Console	19
安全建议	19
将 Identity Console 部署为 Docker 容器	20
部署 OSP 容器	20
将 Identity Console 部署为 Docker 容器	22
将 Identity Console 作为 Docker 的多树	24
部署独立 Identity Console	24
部署独立 Identity Console（非 Docker）	24
具有独立 Identity Console 的多树	26
Identity Console 在 Windows 上作为工作站	26
将 Identity Console 作为工作站的多树	27
停止和重新启动 Identity Console	27
停止并将 Identity Console 作为 Docker 容器重新启动	27
停止和重新启动独立 Identity Console	28
关闭并重启 Identity Console 工作站	28
管理数据持久性	28
在 Azure Kubernetes Services 中部署 Identity Console	29
在 AKS 群集中部署 Identity Console	29
修改服务器证书	35
修改 Docker 容器中的服务器证书	35
修改独立 Identity Console 中的服务器证书	36
3 升级 Identity Console	37
将 Identity Console 作为 Docker 容器进行升级	37
升级独立 Identity Console（非 Docker）	39
升级 OSP 容器	39

4 卸载 Identity Console	41
Docker 环境卸载过程	41
独立 Identity Console 的卸载过程（非 Docker）	41

关于本书和库

*Identity Console 安装指南*提供有关如何安装和管理 NetIQ Identity Console (Identity Console) 产品的相关信息。此指南定义术语，并提供各种实施案例。

目标受众

本指南面向网络管理员。

库中的其他信息

此库提供了以下信息资源：

安装指南

介绍如何安装和升级 Identity Console。该书适用于网络管理员。

关于 NetIQ Corporation

我们是一家全球性的企业软件公司，专注于您的环境中三大永恒挑战：变化、复杂性和风险，设法帮助您应对这些挑战。

我们的观点

适应变化及管理复杂性和风险实乃老生常谈

实际上在您面临的所有挑战中，这些也许是容易让您失控的最突出变数，从而无法安全地衡量、监视和管理您的物理环境、虚拟环境和云计算环境所需。

提供更好、更快的关键业务服务

我们相信，尽可能多地为 IT 组织提供控制，是更及时、经济有效地交付服务的唯一方法。只有在组织不断做出改变，并且管理这些变化所需的技术本身日益复杂时，持续存在的压力（如变化和复杂性）才会继续增大。

我们的理念

销售智能解决方案，而不只是软件

为了提供可靠的控制，我们首先务必了解 IT 组织（如贵组织）的实际日常运作情况。这才是我们可以开发出实用的智能型 IT 解决方案以成功取得公认的重大成果的唯一途径。并且，这比单纯销售软件要有价值得多。

推动您走向成功是我们的追求

我们将您的成功视为我们业务活动的核心。从产品启动到部署，我们深知：您需要与您当前购买的解决方案配合使用和完美集成的解决方案；您需要在部署后获得持续的支持并接受后续的培训；您还需要真正易于合作的伙伴一起应对变化。总之，只有您成功，才是我们都成功。

我们的解决方案

- ◆ 身份和访问管理
- ◆ 访问管理
- ◆ 安全管理
- ◆ 系统和应用程序管理
- ◆ 工作负载管理
- ◆ 服务管理

与销售支持联系

若对产品、定价和功能有疑问，请与本地合作伙伴联系。如果无法与合作伙伴联系，请与我们的销售支持团队联系。

全球：	www.netiq.com/about_netiq/officelocations.asp
美国和加拿大：	1-888-323-6768
电子邮件：	info@netiq.com
网站：	www.netiq.com

联系技术支持

有关具体的产品问题，请与我们的技术支持团队联系。

全球：	www.netiq.com/support/contactinfo.asp
北美和南美：	1-713-418-5555
欧洲、中东和非洲：	+353 (0) 91-782 677
电子邮件：	support@netiq.com
网站：	www.netiq.com/support

联系文档支持

我们的目标是提供满足您的需要的文档。如果您有改进建议，请单击 www.netiq.com/documentation 上发布的 HTML 版文档任何页面底部的添加注释。您还可以发送电子邮件至 Documentation-Feedback@netiq.com。我们会重视您的意见，欢迎您提供建议。

联系在线用户社区

Qmunity 是 NetIQ 在线社区的简称，它是让您可与同行和 NetIQ 专家沟通的协作网络。通过提供更多即时信息、指向实用资源的有用链接，以及 NetIQ 专家的支持，Qmunity 有助于确保您可以掌握必要的知识，以充分发挥所依赖的 IT 投资的潜力。有关详细信息，请访问 <http://community.netiq.com>。

1 计划安装 Identity Console

本章介绍安装 Identity Console 的系统要求和先决条件。由于 Identity Console 可以作为 Docker 容器运行，也可以作为独立应用程序运行，因此请参见两种安装类型的系统要求和先决条件的相关部分。

注释： Identity Console 支持 eDirectory 9.2.4 HF2、Identity Manager Engine 4.8.3 HF2 及其各自的更高版本。在使用 Identity Console 之前，必须升级您的 eDirectory 和 Identity Manager Engine 实例。

- ◆ [Docker 安装的系统要求和先决条件（第 9 页）](#)
- ◆ [独立安装（非 Docker）的系统要求和先决条件（第 13 页）](#)
- ◆ [工作站的系统要求和先决条件（第 15 页）](#)
- ◆ [RPM 签名校验（第 16 页）](#)

Docker 安装的系统要求和先决条件

本部分介绍将 Identity Console 安装为 Docker 容器的系统要求和先决条件。

- ◆ [系统要求（第 9 页）](#)
- ◆ [先决条件（第 9 页）](#)
- ◆ [设置环境（第 10 页）](#)

系统要求

由于 Identity Console 可以作为 Docker 容器运行，有关安装 Identity Console 的系统要求和平台支持平台的信息，请参见 [Docker Documentation](#)（《Docker 文档》）。

先决条件

- 安装 Docker 20.10.9-ce 或更高版本。关于如何安装 Docker 的更多信息，请参见 [Docker Installation](#)（《Docker 安装》）。
- 您必须获取包含私用密钥的 pkcs12 服务器证书，以加密 / 解密 Identity Console 服务器与后端服务器之间的数据交换。此服务器证书用于保护 http 连接。您可以使用任何外部证书颁发机构生成的服务器证书。更多信息，请参见 [Creating Server Certificate Objects](#)（创建服务器证书对象）。服务器证书应包含主题备用名称以及 Identity Console 服务器的 IP 地址和 DNS。创建服务器证书对象后，必须以 .pfx 格式导出。

- ❑ 您必须为所有树获取 .pem 格式的证书颁发机构证书，以验证前一步骤中获得的服务器证书的证书颁发机构签名。此 rootCA 证书还确保客户端和 Identity Console 服务器之间建立安全的 LDAP 通信。例如，您可以从 /var/opt/novell/eDirectory/data/SSCert.pem 获取 eDirectory CA 证书 (SSCert.pem)。
- ❑ (可选) 使用 One SSO Provider (OSP)，您可以为用户启用到 Identity Console 门户的单点登录 (SSO) 鉴定。在安装 Identity Console 之前，您必须安装 OSP。要为 Identity Console 配置 OSP，按照屏幕提示并提供配置参数所需值。有关详细信息，请参见部署 OSP 容器 (第 20 页)。要将 Identity Console 注册到现有的 OSP 服务器上，您必须将以下内容手动添加到 /opt/netiq/idm/apps/tomcat/conf/ 文件夹中的 ism-configuration.properties 文件中：

```
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
```

注释：对于 OSP，您只能连接到单个 eDirectory 树，因为 OSP 不支持多个 eDirectory 树。

- ❑ 确保 /etc/hosts 中存在针对主机的具有完全限定的主机名的正确 DNS 条目。
- ❑ 如果您想在 Edge 浏览器中使用 Identity Console，您必须下载最新版本的 Microsoft Edge 以获得完整的功能。

注释：在 Mozilla Firefox (火狐) 中使用 Identity Console 时，操作可能会失败，显示 Origin Mismatch (源不匹配) 错误讯息。要查错，执行以下步骤：

- 1 将 Firefox 更新为最新版本。
 - 2 在 Firefox URL 字段中指定 about:config 然后按 Enter 键。
 - 3 搜索来源。
 - 4 双击 network.http.SendOriginHeader 并将其值更改为 1。
-

设置环境

您可能需要创建包含某些参数的配置文件。如果您想用 OSP 配置 Identity Console，则必须在配置文件中指定 OSP 特定参数。例如，使用 OSP 参数创建以下 edirapi.conf 文件：

注释：您必须在 osp-redirect-url 字段中提供您的 eDirectory 树名。

```
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"
```

如果不想用 OSP 配置 Identity Console，创建下文所示的配置文件，无 OSP 参数：

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

注释：当您要使用多个 eDirectory 树配置 Identity Console 时，您可以跳过 *ldapserver*、*ldapuser* 和 *ldappassword* 参数并创建配置文件。

表 1-1 配置文件中配置参数的说明

配置参数	说明
listen	指定 9000 作为容器内 Identity Console 服务器的侦听端口。
ldapserver	指定 eDirectory 主机服务器 IP 和端口号。
ldapuser	指定 eDirectory 用户的用户名。此参数用作身份凭证，用于在 OSP 登录的情况下使用代理授权控制启动对 eDirectory 的 ldap 调用。Ldap 用户必须对 eDirectory 树具有主管权限。
ldappassword	指定 LDAP 用户的口令。
pfxpassword	指定 pkcs12 服务器证书文件的口令。
ospmode	指定 true 以将 OSP 和 Identity Console 集成。如果您将此设置为 false，Identity Console 将使用 LDAP 登录。

配置参数	说明
osp-token-endpoint	此 URL 用于从 OSP 服务器提取特定属性，以校验身份验证令牌的有效性。
osp-authorize-url	此 URL 供用户使用，提供身份凭证以获取身份验证令牌。
osp-logout-url	使用此 URL 终止用户和 OSP 服务器之间的会话。
osp-redirect-url	OSP 服务器在授予身份验证令牌后将用户重新定向到此 URL。 注释： 在配置 Identity Console 时，确保以小写形式指定 eDirectory 树名称。如果未以小写形式指定树名称，那么登录 Identity Console 服务器可能会失败。
osp-client-id	指定 Identity Console 注册 OSP 时提供的 OSP 客户端 ID。
ospclientpass	指定 Identity Console 注册 OSP 时提供的 OSP 客户端口令。
ospcert	指定 OSP 服务器 CA 证书的位置。
bcert	指定 Identity Console 的证书颁发机构证书的位置。
loglevel	指定您想要包括在日志文件中的日志等级。此参数可以设置为 "fatal"、"error"、"warn" 或 "info"。
check-origin	如果将其设置为 true，Identity Console 服务器会比较请求的原始值。可用选项为 true 或 false。使用 DNS 配置时，即使 <i>check-origin</i> 参数值设置为 false， <i>origin</i> 参数也是必需的。
origin	Identity Console 将请求的原始值与此字段中指定的值进行比较。 注释： 从 Identity Console 1.4 开始，此参数独立于 <i>check-origin</i> 参数，如果使用 DNS 配置，则此参数是必需的。
maxclients	可访问 IDConsole 的并发客户端的最大数量。超出此限制的任何其他客户必须排队等候。

注释：

- ◆ 仅当计划将 OSP 与 Identity Console 集成时才应使用 *ospmode* 配置参数。
- ◆ 如果 Identity Applications (Identity Apps) 在 Identity Manager 设置中以群集模式配置，您必须在配置文件中的 *osp-token-endpoint*、*osp-authorize-url* 和 *osp-logout-url* 字段中提供负载均衡器服务器的 DNS 名称。如果您在这些字段中提供 OSP 服务器细节，则 Identity Console 登录将失败。

- ◆ 如果 Identity Console 配置了和 Identity Apps 以及 Identity Reporting 一样的 OSP 实例，那么单点登录 (SSO)（鉴定服务）在您登录 Identity Console 门户时生效。
- ◆ 从 Identity Console 1.4 开始，OSP HTTPS URL 应使用包含 2048 位或更高密钥的证书进行验证。
- ◆ 如果您想限制从不同域访问 Identity Console 门户，将 samesitecookie 参数设置为 strict。如果您想允许从不同域访问 Identity Console 门户，将 samesitecookie 参数设置为 lax。如果配置过程中未指定参数，则默认使用浏览器设置。

在您准备好配置文件之后，部署容器。有关更多信息，请参见 [将 Identity Console 部署为 Docker 容器（第 20 页）](#)。

独立安装（非 Docker）的系统要求和先决条件

- ◆ [系统要求（第 13 页）](#)
- ◆ [（可选）OSP 配置的先决条件（第 14 页）](#)

系统要求

本部分介绍安装独立 Identity Console 的系统要求和先决条件。

类别	最低要求
处理器	1.4 GHz 64 位
内存	2 GB
磁盘空间	Linux 200 MB
支持的浏览器	<ul style="list-style-type: none"> ◆ 最新版 Microsoft Edge ◆ 最新版 Google Chrome ◆ 最新版 Mozilla Firefox <p>注释：在 Mozilla Firefox（火狐）中使用 Identity Console 时，操作可能会失败，显示 Origin Mismatch（源不匹配）错误讯息。要查错，执行以下步骤：</p> <ol style="list-style-type: none"> 1 将 Firefox 更新为最新版本。 2 在 Firefox URL 字段中指定 about:config 然后按 Enter 键。 3 搜索来源。 4 双击 network.http.SendOriginHeader 并将其值更改为 1。

类别	最低要求
支持的操作系统	<ul style="list-style-type: none"> ◆ 已认可： <ul style="list-style-type: none"> ◆ 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 和 15.2 ◆ 支持：支持上述认可操作系统的更高支持包版本。
证书	<ul style="list-style-type: none"> ◆ 您必须获取包含私用密钥的 <code>pkcs12</code> 服务器证书，以加密 / 解密客户端与 Identity Console 服务器之间的数据交换。此服务器证书用于保护 <code>http</code> 连接。您可以使用任何外部证书颁发机构生成的服务器证书。更多信息，请参见 Creating Server Certificate Objects（创建服务器证书对象）。服务器证书应包含主题备用名称以及 Identity Console 服务器的 IP 地址和 DNS。创建服务器证书对象后，必须以 <code>.pfx</code> 格式导出。 ◆ 您必须为所有树获取 <code>.pem</code> 格式的证书颁发机构证书，以验证前一步骤中获得的服务器证书的证书颁发机构签名。此 <code>rootCA</code> 证书还确保客户端和 Identity Console 服务器之间建立安全的 LDAP 通信。例如，您可以从 <code>/var/opt/novell/eDirectory/data/SSCert.pem</code> 获取 eDirectory CA 证书 (<code>SSCert.pem</code>)。

准备好后，继续安装 Identity Console。有关更多信息，请参见[部署独立 Identity Console](#)（第 24 页）。

（可选）OSP 配置的先决条件

使用 One SSO Provider (OSP)，您可以为用户启用到 Identity Console 门户的单点登录 (SSO) 鉴定。在安装 Identity Console 之前，您必须安装 OSP。要为 Identity Console 配置 OSP，按照屏幕提示并提供配置参数所需值。有关详细信息，请参见[部署 OSP 容器](#)（第 20 页）。要将 Identity Console 注册到现有的 OSP 服务器上，您必须将以下内容手动添加到 `/opt/netiq/idm/apps/tomcat/conf/` 文件夹中的 `ism-configuration.properties` 文件中：

```
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
```

注释:

- ◆ 如果您是首次安装 OSP，为 **Configure OSP with eDir API**（通过 eDir API 配置 OSP）指定 "y" 选项，然后按照屏幕上的提示向 OSP 注册 Identity Console。
 - ◆ 在配置 Identity Console 时，确保以小写形式指定 eDirectory 树名称。如果未以小写形式指定树名称，那么登录 Identity Console 服务器可能会失败。
 - ◆ 对于 OSP，您只能连接到单个 eDirectory 树，因为 OSP 不支持多个 eDirectory 树。
-

工作站的系统要求和先决条件

- ◆ [系统要求（第 15 页）](#)

系统要求

本部分介绍运行工作站 Identity Console 的系统要求和先决条件。

类别	最低要求
处理器	1.5 GHz 64 位
内存	2 GB
磁盘空间	Windows 1 GB
支持的操作系统	<ul style="list-style-type: none">◆ 已认可：<ul style="list-style-type: none">◆ Windows Server 2016◆ Windows Server 2019◆ Windows Server 2022◆ Windows 10◆ Windows 11

类别	最低要求
证书	<ul style="list-style-type: none"> 您必须获取 pfx 格式的服务器证书，以便在 Identity Console 客户端和 REST 服务器之间交换数据。此服务器证书必须始终命名为 keys.pfx。更多信息，请参见 Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/b1j4tpo3.html#b1j4u0cm)（创建服务器证书对象）。 您必须为所有树获取 .pem 格式的证书颁发机构证书，以验证前一步骤中获得的服务器证书的证书颁发机构签名。此根证书颁发机构证书还确保客户端和 Identity Console 服务器之间建立安全的 ldap 通信。 <p>例如，您可以从 /var/opt/novell/eDirectory/data/SSCert.pem 获取适用于 Linux 的 eDirectory 证书颁发机构证书 (SSCert.pem)。</p> <p>从 <eDirectory 安装位置>\NetIQ\edirectory\DIBFiles\CertServ\SSCert.pem 获取适用于 Windows 的 eDirectory 证书颁发机构证书 SScert.pem。</p>

准备好后，继续部署 Identity Console。有关更多信息，请参见 [Identity Console 在 Windows 上作为工作站（第 26 页）](#)。

RPM 签名校验

使用以下步骤执行 RPM 签名校验：

- 1 导航到提取版本的文件夹。

例如：<Identity Console 的解压位置 >/IdentityConsole_150_Linux/license/MicroFocusGPGPackageSign.pub。

- 2 运行以下命令以导入公共密钥：

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

- 3（可选）运行以下命令以校验 RPM 签名：rpm --checksig -v <RPM 名称 >

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

本章介绍部署 Identity Console 的过程以及安全建议。要准备部署，查看 [第 1 章“计划安装 Identity Console”](#)（第 9 页）中规定的先决条件和系统要求。

- [安全建议](#)（第 19 页）
- [将 Identity Console 部署为 Docker 容器](#)（第 20 页）
- [部署独立 Identity Console](#)（第 24 页）
- [Identity Console 在 Windows 上作为工作站](#)（第 26 页）
- [停止和重新启动 Identity Console](#)（第 27 页）
- [管理数据持久性](#)（第 28 页）
- [在 Azure Kubernetes Services 中部署 Identity Console](#)（第 29 页）
- [修改服务器证书](#)（第 35 页）

安全建议

- Docker 容器默认没有任何资源限制。这为每个容器提供了主机内核所提供的所有 CPU 和内存资源的访问权限。您还必须通过设置容器可使用的资源量限制，确保一个正在运行的容器不会使用更多资源，致使其他正在运行的容器停止。
 - Docker 容器应在 Docker run 命令中使用 `--memory` 标志确保容器使用的内存应用了硬限制。
 - Docker 容器应在 Docker run 命令中使用 `--cpuset-cpus` 标志确保运行的容器所使用的 CPU 量应用了限制。
- `--pids-limit` 应设置为 300 以限制任何指定时间容器内生成的内核线程的数量。这样是为了防止 DoS 攻击。
- 您必须在 Docker run 命令中使用 `--restart` 标志将故障容器重新启动策略设置为 5。
- 容器出现后，您只能在运行状态显示为 **Healthy**（健康）后使用容器。要检查容器的运行状态，运行以下命令：

```
docker ps <container_name/ID>
```
- Docker 容器将始终以非 root user (nds) 启动。作为额外的安全措施，启用守护程序用户名称空间重新映射，以防止容器内部发生特权降级攻击。有关用户名称空间重新映射的更多信息，请参见 [Isolate containers with a user namespace](#)（用用户名称空间隔离容器）。

将 Identity Console 部署为 Docker 容器

本部分包含下列操作过程的信息：

- ◆ 部署 OSP 容器（第 20 页）
- ◆ 将 Identity Console 部署为 Docker 容器（第 22 页）
- ◆ 将 Identity Console 作为 Docker 的多树（第 24 页）

部署 OSP 容器

要部署 OSP 容器，执行下列操作：

- 1 登录到 [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/)，然后导航到“软件下载”页面。
- 2 选择下列项目：
 - ◆ 产品：eDirectory
 - ◆ 产品名称：eDirectory per User Sub SW E-LTU
 - ◆ 版本：9.2
- 3 下载文件：IdentityConsole_<版本>_Containers_tar.zip。
- 4 将下载的文件提取到文件夹中。
- 5 根据您的要求修改静默属性文件。下面显示了一个示例静默属性文件：

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

# OSP Details
SSO_SERVER_HOST=osp.example.com
SSO_SERVER_SSL_PORT=8543
OSP_COMM_TOMCAT_KEYSTORE_FILE=/config/tomcat.ks
OSP_COMM_TOMCAT_KEYSTORE_PWD=novell
SSO_SERVICE_PWD=novell
OSP_KEYSTORE_PWD=novell
IDM_KEYSTORE_PWD=novell
OSP_CUSTOM_NAME="Identity Console"
USER_CONTAINER="o=novell"
ADMIN_CONTAINER="o=novell"

# IDConsole Details
IDCONSOLE_HOST=192.168.1.1
IDCONSOLE_PORT=9000
```

```

EDIRAPI_TREENAME=ed913

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

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

```

注释： 要在使用静默属性（DOS 文本）文件时避免空间限制，必须使用 `dos2unix` 工具将 DOS 文本文件转换为 UNIX 格式。运行以下命令将文本文件从 DOS 行尾转换为 Unix 行尾：

```
dos2unix filename
```

例如，

```
dos2unix samplefile
```

-
- 6 使用 iManager 生成一个服务器证书 (`cert.der`) 并将其导入密钥存储区 (`tomcat.ks`)。将静默属性文件和密钥存储区 (`tomcat.ks`) 复制到任何目录中。例如，`/data`。执行下列操作创建服务器证书并将其导入密钥存储区：

- 6a 运行下列命令创建密钥存储区 (`tomcat.ks`)。生成密钥，确保计算机的 CN 名或完全限定的主机名是 IP 地址。

```
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 运行下列命令创建证书签名请求。例如 `cert.csr`。

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

- 6c 将此 `cert.csr` 传递给 iManager 并获取 `osp.der` 服务器证书。确保选择密钥类型为“自定义”；密钥用法选项为数据加密、密钥加密和数字签名，并将证书的主题备用名称字段设置为包含 OSP 服务器的 IP 地址或主机名。有关更多信息，请参见[创建服务器证书对象](#)。

- 6d 运行下列命令将 CA 证书 (`SSCert.der`) 和服务器证书 (`cert.der`) 导入 `tomcat.ks` 密钥存储区。

```
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 运行以下命令以装载 OSP 映像：

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

8 使用以下命令部署容器:

```
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
```

将 Identity Console 部署为 Docker 容器

本部分介绍将 Identity Console 部署为 Docker 容器的过程:

注释: 此过程中提及的配置参数、示例值和示例仅供参考。您必须确保不要直接在生产环境中使用它们。

- 1 登录到 SLD: [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/), 然后导航到 “软件下载” 页面。
- 2 选择下列项目:
 - ◆ 产品: eDirectory
 - ◆ 产品名称: eDirectory per User Sub SW E-LTU
 - ◆ 版本: 9.2
- 3 下载文件: IdentityConsole_<版本>_Container.tar.zip。
- 4 映像必须加载到本地 Docker 注册表中。使用以下命令解压缩并装载 IdentityConsole_<版本>_Containers.tar.gz 文件:

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

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

5 使用以下命令创建 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.
```

注释:

- ◆ 您可以通过将 ACCEPT_EULA 环境变量设置为 "Y" 以接受 EULA。您还可以在启动容器时通过使用互动模式 Docker 创建命令中的 -it 选项从屏幕提示中接受 EULA。
- ◆ 上述命令中的 --volume 参数将创建存储配置和日志数据的卷。在这种情况下，我们创建了一个名为 IDConsole-volume 的卷。

-
- 6** 使用以下命令将本地文件系统中的服务器证书文件复制到容器中作为 /etc/opt/novell/eDirAPI/cert/keys.pfx。有关创建服务器证书的更多信息，请参见 [先决条件（第 9 页）](#)：

```
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
```

连接到多个 eDirectory 树时，必须确保为所有连接的树获取至少一个 keys.pfx 服务器证书。

- 7** 使用以下命令将本地文件系统中的 CA 证书文件 (.pem) 复制到容器中作为 /etc/opt/novell/eDirAPI/cert/SSCert.pem。有关获得 CA 证书的更多信息，请参见 [先决条件（第 9 页）](#)：

```
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
```

如果用户需要连接到多个 eDirectory 树，请参见此部分：[将 Identity Console 作为 Docker 的多树（第 24 页）](#)

- 8** 根据您的要求修改配置文件，并使用以下命令将配置文件 (edirapi.conf) 从本地文件系统复制到容器中，作为 /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** 使用以下命令启动 Docker 容器：

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

例如，

```
docker start identityconsole-container
```

注释: /var/lib/docker/volumes/<volume_name>/_data/eDirAPI/var/log 目录下将有下列日志文件：

- ◆ edirapi.log - 用于记录 edirapi 中的不同事件和调试问题。

- ◆ edirapi_audit.log - 用于记录 edirapi 的审计事件。日志遵循 CEF 审计格式。
 - ◆ container-startup.log - 用于捕获 Identity Console docker 容器的安装日志。
-

将 Identity Console 作为 Docker 的多树

Identity Console 允许用户通过获取树的单个证书颁发机构证书来连接到多个树。

例如，如果连接到三个 eDirectory 树，则必须将所有三个证书颁发机构证书复制到 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
```

运行以下命令以重新启动 Identity Console：

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

部署独立 Identity Console

- ◆ 部署独立 Identity Console（非 Docker）（第 24 页）
- ◆ 具有独立 Identity Console 的多树（第 26 页）

部署独立 Identity Console（非 Docker）

本部分介绍部署独立 Identity Console 的过程：

- 1 登录到 SLD: [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/)，然后导航到“软件下载”页面。
- 2 选择下列项目：
 - ◆ 产品：eDirectory
 - ◆ 产品名称：eDirectory per User Sub SW E-LTU
 - ◆ 版本：9.2
- 3 下载最新的 Identity Console 版本。
- 4 将下载的文件提取到一个文件夹中。
- 5 打开壳层并导航到提取 Identity Console 版本的文件夹。
- 6 以 root user 或 root 等效用户的身份登录时运行以下命令：

```
./identityconsole_install
```
- 7 阅读“简介”内容，然后单击 **Enter**。

- 8 单击 "Y" 接受许可协议。这将在您的系统上安装所有所需的 RPM。
- 9 输入 Identity Console 服务器的主机名（完全限定的域名 (FQDN)）/IP 地址。
- 10 输入要侦听的 Identity Console 端口号。默认值是 9000。
- 11 输入将 OSP 与 Identity Console 集成或使 Identity Console 使用 LDAP 登录的选项。
- 12 如果要将在 OSP 与 Identity Console 集成：

1. 输入 eDirectory/ 身份库服务器的域名 /IP 地址以及 LDAPS 端口号。

例如，

192.168.1.1:636

2. 输入 eDirectory/ 身份库用户名。

例如，

cn=admin,ou=org_unit,o=org

3. 输入 eDirectory/ 身份库口令。

4. 再次输入 eDirectory/ 身份库口令以确认口令。

5. 输入 OSP 服务器域名 /IP 地址以及 SSO 服务器 SSL 端口号。

6. 输入 OSP 客户端 ID。

7. 输入 OSP 客户端口令。

8. 输入 eDirectory/ 身份库目录树名称。

- 13 输入可信根证书 (SSCert.pem) 路径，包括文件夹。

例如，

/home/Identity_Console/certs

注释： 用户必须确保不在 cert 文件夹中创建子目录。

- 14 输入服务器证书 (keys.pfx) 路径，包括文件名。

例如，

/home/Identity_Console/keys.pfx

- 15 输入服务器证书口令。要确认您输入的口令正确无误，请重新输入服务器证书口令。安装启动。

注释： /var/opt/novell/eDirAPI/log 目录中将有下列日志文件：

- ◆ edirapi.log - 用于记录 edirapi 中的不同事件和调试问题。
- ◆ edirapi_audit.log - 用于记录 edirapi 的审计事件。日志遵循 CEF 审计格式。
- ◆ identityconsole_install.log - 用于捕获 Identity Console 的安装日志。

Identity Console 进程启动 / 停止日志位于 /var/log/messages 文件中。

注释： NetIQ 建议，如果在同一台机器上安装 Identity Console 和 eDirectory，则该机器至少有一个可用的 eDirectory 实例。

具有独立 Identity Console 的多树

当您连接到多个 eDirectory 树时，必须确保获取树的单个证书颁发机构证书。

例如，如果连接到三个 eDirectory 树，则必须将所有三个证书颁发机构证书复制到 `etc/opt/novell/eDirAPI/cert/` 目录中：

```
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
```

运行以下命令之一以重启动 Identity Console：

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

或者

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

Identity Console 在 Windows 上作为工作站

Identity Console 可以在 Windows 上作为工作站启动，并且需要运行 REST 服务。因此，当它启动时，eDirAPI 进程将在 `edirapi.exe cmd` 提示符中运行。如果关闭此 `edirapi.exe` 终端，则 Identity Console 将无法正常工作。

以下过程介绍如何在 Windows 上运行 Identity Console。

- 1 登录到 SLD Software License and Download (<https://sldlogin.microfocus.com/nidp/idff/sso?pid=5&sid=0&option=credential&sid=0>)，导航到“软件下载”页面。
- 2 选择下列项目：
 - ◆ 产品：eDirectory
 - ◆ 产品名称：eDirectory per User Sub SW E-LTU
 - ◆ 版本：9.2
- 3 下载文件 `IdentityConsole_<版本>_workstation_win_x86_64.zip`。
- 4 将下载的 `IdentityConsole_<版本>_workstation_win_x86_64.zip` 文件提取到文件夹中。
- 5 导航到提取的文件夹：`IdentityConsole_150_workstation_win_x86_64\eDirAPI\cert`，然后复制可信根证书颁发机构 `SSCert.pem` 和服务器证书 `keys.pfx`。

要获取证书，请参阅此部分：[工作站的系统要求和先决条件（第 15 页）](#)

如果用户需要连接到多个 eDirectory 树，请参阅此部分：[将 Identity Console 作为工作站的多树（第 27 页）](#)

注释： 服务器证书名称必须始终为 `keys.pfx`。

- 6 导航到提取版本的文件夹，然后双击文件 `run.bat`（Windows 批文件）。

- 7 在命令提示符中输入服务器证书 (keys.pfx) 口令。
eDirAPI 进程终端 (edirapi.exe) 开始运行，并显示 Identity Console 登录页面。

注释:

- ◆ 如果 eDirAPI 进程终端 (edirapi.exe) 已在运行，则从版本提取文件夹中运行 identityconsole.exe。
 - ◆ 用户可以在 \IdentityConsole_150_workstation_win_x86_64\edirAPI\log 中找到以下日志
edirapi.log - 用于记录 edirapi 中的不同事件和调试问题。
edirapi_audit.log - 用于记录 edirapi 的审计事件。日志遵循 CEF 审计格式。
 - ◆ 在工作站模式下不支持基于 OSP 的登录。
 - ◆ Identity Console 工作站侦听仅在 9000 端口上进行。请勿修改 edirapi_win.conf 文件。
-

将 Identity Console 作为工作站的多树

Identity Console 允许用户通过获取树的单个证书颁发机构证书来连接到多个树。

- 1 关闭 Identity Console 工作站和 eDirAPI 终端。
- 2 将证书颁发机构证书 SSCert.pem 复制到以下位置：
IdentityConsole_150_workstation_win_x86_64\edirAPI\cert。
例如，如果要连接到三个 eDirectory 树，请将证书颁发机构证书分别复制为 SSCert1.pem、SSCert2.pem 和 SSCert3.pem。
- 3 导航到提取版本的文件夹，然后双击文件 run.bat（Windows 批文件）。
- 4 在终端提示符中输入 keys.pfx 口令，然后登录到所需的 eDirectory 树。

停止和重新启动 Identity Console

- ◆ [停止并将 Identity Console 作为 Docker 容器重新启动](#)（第 27 页）
- ◆ [停止和重新启动独立 Identity Console](#)（第 28 页）
- ◆ [关闭并重启 Identity Console 工作站](#)（第 28 页）

停止并将 Identity Console 作为 Docker 容器重新启动

要停止 Identity Console，运行以下命令：

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

要重新启动 Identity Console，运行以下命令：

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

要启动 Identity Console，运行以下命令：

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

停止和重启动独立 Identity Console

要停止 Identity Console，运行以下命令之一：

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

或者

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

要重启动 Identity Console，运行以下命令之一：

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

或者

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

要启动 Identity Console，运行以下命令之一：

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

或者

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

关闭并重启 Identity Console 工作站

要关闭应用程序和进程，请按以下步骤操作：

- 1 关闭 Identity Console 桌面窗口应用程序。
- 2 通过关闭 eDirAPI 进程终端停止 eDirAPI 进程。

要重启 Identity Console 工作站，请导航到提取版本的文件夹，然后双击文件 run.bat（Windows 批文件）。

注释：如果 eDirAPI 进程终端已在运行，则从版本提取文件夹中运行 identityconsole.exe 以重启 Identity Console 工作站。

管理数据持久性

除了 Identity Console 容器外，还创建了数据持久性卷。要借助使用卷的旧容器的配置参数，执行以下步骤：

- 1 使用以下命令停止当前的 Docker 容器：

```
docker stop identityconsole-container
```

- 2 使用存储在 Docker 卷 (edirapi-volume-1) 中的旧容器的应用程序数据创建第二个容器：

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

- 3 使用以下命令启动第二个容器:

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

- 4 (可选) 现在, 可以使用以下命令去除第一个容器:

```
docker rm identityconsole-container
```

在 Azure Kubernetes Services 中部署 Identity Console

Azure Kubernetes Services (AKS) 是一项托管式 Kubernetes 服务, 可用于部署和管理群集。本部分包含下列操作过程的信息:

在 AKS 群集中部署 Identity Console

本部分介绍下列在 AKS 群集中部署 Identity Console 的操作过程:

- ◆ [创建 Azure 容器注册表 \(ACR\)](#) (第 29 页)
- ◆ [设置 Kubernetes 群集](#) (第 30 页)
- ◆ [创建标准库存单位 \(SKU\) 公共 IP 地址](#) (第 31 页)
- ◆ [设置云壳层并连接到 Kubernetes 群集](#) (第 31 页)
- ◆ [部署应用程序](#) (第 31 页)

创建 Azure 容器注册表 (ACR)

Azure 容器注册表 (ACR) 是一个基于 Azure 的专用注册表, 适用于 Docker 容器映像。

有关更多详细步骤, 请参阅 [Create container registry - Portal](#) (创建容器注册表 - 门户) 中的 [Create an Azure container registry using the Azure portal](#) (使用 Azure 门户创建 Azure 容器注册表) 部分, 或执行以下步骤创建 Azure 容器注册表 (ACR):

1. 登录到 [Azure 门户](#)。
2. 转到 **Create a resource** (创建资源) > **Containers** (容器) > **Container Registry** (容器注册表)。
3. 在 **Basics** (基本信息) 选项卡中, 指定 **Resource group** (资源组) 和 **Registry name** (注册表名称) 的值。注册表名称在 Azure 中必须是唯一的, 并且至少包含 5 个字母数字字符, 最多包含 50 个字母数字字符。
接受其余设置的默认值。
4. 单击 **Review + create** (查看 + 创建)。
5. 单击 **创建**。
6. 登录到 Azure CLI, 运行以下命令以登录到 Azure 容器注册表

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

例如：

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

7. 使用以下命令检索 Azure 容器注册表的登录服务器：

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

例如：

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

8. 使用以下命令用 ACR 登录服务器的名称 (registryname.azurecr.io) 标记 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. 将标记的映像推送到注册表。

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

例如，

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

10. 使用以下命令检索注册表中的映像列表：

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

设置 Kubernetes 群集

使用 Azure 门户或 CLI 创建 kubernetes 服务资源。

有关在 Azure 中创建一个包含节点的 Kubernetes 服务资源的更多详细步骤，请参阅 [Azure Quickstart](#)（Azure 快速入门）中的 [Create an AKS Cluster](#)（创建一个 AKS 群集）。

注释：

- ◆ 确保选择 Azure CNI 作为网络。
 - ◆ 选择现有虚拟网络（其中 eDirectory 服务器部署在子网中）。
 - ◆ 选择 Identity Console 映像在其中可用的现有容器注册表。
-

创建标准库存单位 (SKU) 公共 IP 地址


Kubernetes 群集资源组下的公共 IP 地址资源充当应用程序的负载均衡器 IP。

有关详细步骤，请参阅 [Create public IP address – Portal](#)（创建公共 IP 地址 - 门户）中的 [Create a public IP address using the Azure portal](#)（使用 Azure 门户创建公共 IP 地址）。

设置云壳层并连接到 Kubernetes 群集

使用 Azure 门户中可用于所有操作的云壳层。

若要在 Azure 门户中设置云壳层，请参阅 [Bash – Quickstart](#)（Bash – 快速入门）中的 [Start Cloud Shell](#)（启动云壳层）部分，或执行以下步骤来设置云壳层并连接到 Kubernetes 群集：

1. 在 Azure 门户中，单击  按钮打开云壳层。

注释：要管理 Kubernetes 群集，请使用 Kubernetes 命令行客户端 kubectl。如果使用的是 Azure Cloud Shell，则已安装 kubectl。

2. 使用以下命令配置 kubectl 以连接到您的 Kubernetes 群集：

```
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. 使用以下命令校验群集节点的列表：

```
kubectl get nodes
```

部署应用程序

要部署 Identity Console，您可以使用 `idc-services.yaml`、`idc-statefulset.yaml`、`idc-storageclass.yaml` 和 `idc-pvc.yaml` 示例文件。

您还可以根据要求创建自己的 `yaml` 文件。

1. 使用以下命令创建储存类资源：

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

例如，

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

（可选）有关如何通过 Azure 文件共享动态创建和使用持久性卷的更多信息，请参阅 [Dynamically create and use a persistent volume with Azure Files in Azure Kubernetes Service \(AKS\)](#)（在 Azure Kubernetes Service (AKS) 中通过 Azure 文件动态创建和使用持久性卷）

示例储存类资源文件如下所示：

```

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
~

```

储存类资源支持动态储存供应。它用于定义如何创建 Azure 文件共享。

2. 使用以下命令查看储存类的细节：

```
kubectl get sc
```

3. 使用 `idc-pvc.yaml` 文件创建一个 `pvc` 资源：

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

例如，

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

示例 `pvc` 资源文件如下所示：


```

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

```

持久性卷声明资源创建文件共享。持久性卷声明 (PVC) 使用储存类对象动态供应 Azure 文件共享。

4. 将 `edirapi.conf`、证书颁发机构证书和服务器证书上传到云壳层。

单击云壳层上的 **Upload/Download files**（上载 / 下载文件）按钮图标 ，然后上载 `edirapi.conf`、`SSCert.pem` 和 `keys.pfx` 文件。

注释： `edirapi.conf` 有一个参数“原点”。在这里，我们需要提供用于访问 Identity Console 应用程序的 IP 地址。（使用在 [创建标准库存单位 \(SKU\) 公共 IP 地址](#)（第 31 页）部分中创建的 IP 地址。）

Identity Console 部署需要服务器证书 (`keys.pfx`)。

创建服务器证书时，请确保在主题备用名称中提供有效的 DNS 名称。

构建有效 DNS 名称的步骤：

使用 StatefulSet 部署的典型 pod 具有如下所示的 DNS 名称 - `{statefulsetname}-{ordinal}.{servicename}.{namespace}.svc.cluster.local`

- ◆ 如果 `idconsole-statefulset.yaml` 文件中的 StatefulSet 名称是 `idconsole-app`，则 `statefulsetname = idconsole-app`
- ◆ 如果是第一个 Pod，则序号 = 0
- ◆ 如果将 `idconsole -statefulset.yaml` 文件中的 `serviceName` 定义为 `idconsole`，则 `serviceName = idconsole`
- ◆ 如果是默认命名空间，则 `namespace=default`

输出：`idconsole-app-0.idconsole.default.svc.cluster.local`

5. 在 Kubernetes 群集中创建一个 `configmap` 资源，该资源将储存配置文件以及证书。
在运行该命令之前，请确保目录中存在文件 `edirapi.conf`、`SSCert.pem` 和 `keys.pfx`。

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

例如，

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

6. 查看 `configmap` 对象的细节，使用 `kubectl` 描述命令：

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

例如，

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

7. 创建 StatefulSet 资源以部署容器。

运行以下命令以部署容器：

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

例如，

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

示例 StatefulSet 资源文件如下所示：

```

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: pvcforsec

```

8. 运行以下命令校验已部署 Pod 的状态:

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

9. 创建 `loadBalancer` 类型的服务资源。

yaml 文件中指定的服务类型是 `loadBalancer`。

使用以下命令创建服务资源:

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

例如,

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

示例服务资源文件如下所示:


```
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
```

使用以下命令检查外部 IP 地址（或 loadBalancerIP）：

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

10. 使用外部 IP（或 loadBalancerIP 地址）启动 URL。

例如，

```
https://< 外部 IP>:9000/identityconsole
```

修改服务器证书

本部分提供有关修改 Docker 容器和独立 Identity Console 中的服务器证书的信息。

- [修改 Docker 容器中的服务器证书（第 35 页）](#)
- [修改独立 Identity Console 中的服务器证书（第 36 页）](#)

修改 Docker 容器中的服务器证书

执行以下步骤以修改 docker 容器中的服务器证书：

- 1 运行以下命令以在容器的任何位置复制新的服务器证书。

例如，

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

- 2 使用以下命令登录到容器：

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

- 3 运行 NLPCERT 以伪用户的身份储存密钥：

```
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 使用以下命令退出容器控制台：

```
exit
```

5 通过输入以下内容重新启动容器:

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

修改独立 Identity Console 中的服务器证书

执行以下步骤以修改独立容器中的服务器证书:

1 运行 NLPCERT 来储存密钥:

```
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 重新启动 Identity Console:

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

3 升级 Identity Console

本章介绍将 Identity Console 升级到最新版本的过程。要准备升级，请查看 第 1 章 “计划安装 Identity Console”（第 9 页）中提供的先决条件和系统要求。

本部分包含下列操作过程的信息：

- ◆ 将 Identity Console 作为 Docker 容器进行升级（第 37 页）
- ◆ 升级独立 Identity Console（非 Docker）（第 39 页）
- ◆ 升级 OSP 容器（第 39 页）

将 Identity Console 作为 Docker 容器进行升级

当新版本的 Identity Console 映像可用时，管理员可执行升级过程，以使用最新版 Identity Console 部署容器。执行升级之前，确保将所有必需的应用程序相关数据一直存储在 Docker 卷中。执行以下步骤以使用 Docker 容器升级 Identity Console：

- 1 从 [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) 中下载并装载最新版 Docker 映像，然后执行以下步骤安装最新版的 Identity Console，如 [部署 Identity Console](#)（第 19 页）所述。
- 2 装载最新的 Docker 映像后，使用以下命令停止当前的 Docker 容器：

```
docker stop identityconsole-container
```

- 3（可选）对共享卷进行备份。
- 4 运行下列命令删除现有 Identity Console 容器：

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

例如，

```
docker rm identityconsole-container
```

- 5（可选）运行下列命令删除过时的 Identity Console Docker 映像：

```
docker rmi identityconsole
```

- 6 使用以下命令创建 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
```

注释:

- ◆ 您可以通过将 `ACCEPT_EULA` 环境变量设置为 "Y" 以接受 EULA。您还可以在启动容器时通过使用互动模式 Docker 创建命令中的 `-it` 选项从屏幕提示中接受 EULA。
- ◆ 上述命令中的 `--volume` 参数将创建存储配置和日志数据的卷。在这种情况下，我们创建了一个名为 `IDConsole-volume` 的卷。

-
- 7 使用以下命令将本地文件系统中的服务器证书文件复制到新建的容器中，作为 `/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
```

连接到多个 eDirectory 树时，必须确保为所有连接的树复制至少一个 `keys.pfx` 服务器证书。

- 8 使用以下命令将本地文件系统中的 CA 证书文件 (`.pem`) 复制到新建的容器中作为 `/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
```

连接到多个 eDirectory 树时，必须确保为所有连接的树获取单独的证书颁发机构证书。例如，如果连接到三个 eDirectory 树，则必须将所有三个证书颁发机构证书复制到 `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
```

注释: 从 Identity Console 1.4 开始，配置文件 (`edirapi.conf`) 不显式包含 `"ldapuser"`、`"ldappassword"` 和 `"ldapservers"` 参数。 `"bcert"` 参数值必须包括可信根证书的目录路径。例如，`bcert = "/etc/opt/novell/eDirAPI/cert/"`。 `"origin"` 参数独立于 `"check-origin"` 参数，并且在使用 DNS 配置时是必需的。

-
- 9 使用以下命令将本地文件系统中的配置文件 (`edirapi.conf`) 复制到新建的容器中，作为 `/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 使用以下命令启动第二个容器:

```
docker start identityconsole-container
```

- 11 要检查运行容器的状态, 运行下列命令:

```
docker ps -a
```

升级独立 Identity Console (非 Docker)

本部分介绍升级独立 Identity Console 的过程:

- 1 从 [Software License and Download](https://sld.microfocus.com/) 中下载 IdentityConsole_<版本>_Containers.tar.gz (<https://sld.microfocus.com/>)
- 2 登录到 SLD, 导航到软件下载 SLD 页面, 然后单击下载
- 3 浏览并选择产品: **eDirectory** > 产品名称: **eDirectory per User Sub SW E-LTU** > 版本: **9.2**
- 4 下载最新的 Identity Console 版本。
- 5 使用以下命令提取下载的文件:

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

- 6 导航到在其中提取 Identity Console 版本的文件夹。
- 7 将要连接的 eDirectory 树的所有可信根证书复制到一个文件夹中。若要将可信根证书复制到文件夹中, 请运行以下命令:

```
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 运行以下命令:

```
./identityconsole_install
```
- 9 指定 **步骤 4** 中所用的可信根证书的文件夹路径。
- 10 Identity Console 已成功升级。

升级 OSP 容器

要升级 OSP 容器, 请执行以下步骤:

- 1 从 [Software License and Download \(https://sld.microfocus.com/\)](https://sld.microfocus.com/) 中下载并加载最新版本 OSP 映像。

例如,

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

- 2 装载最新的 Docker 映像后，使用以下命令停止当前的 OSP 容器：

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

- 3 （可选）对共享卷进行备份。

- 4 运行下列命令删除现有 OSP 容器：

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

例如，

```
docker rm OSP_Container
```

- 5 转到包含密钥存储区 (tomcat.ks) 和静默属性文件的目录，删除现有密钥存储区 (tomcat.ks) 并保留现有的 OSP 文件夹。生成密钥大小为 2048 的新密钥存储区 (tomcat.ks)。有关更多信息，请参阅 [《Identity Console 安装指南》](#) 中的部署 OSP 容器的步骤 4。

- 6 使用以下命令部署容器：

```
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 卸装 Identity Console

本章介绍卸装 Identity Console 的过程：

- [Docker 环境卸装过程](#)（第 41 页）
- [独立 Identity Console 的卸装过程（非 Docker）](#)（第 41 页）

Docker 环境卸装过程

要卸装 Identity Console Docker 容器，执行下列操作：

- 1 停止 Identity Console 容器：

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

- 2 运行以下命令去除 Identity Console Docker 容器：

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

- 3 运行以下命令去除 Docker 映像：

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

- 4 去除 Docker 卷：

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

注释： 如果去除卷，也将一并从服务器中去除数据。

独立 Identity Console 的卸装过程（非 Docker）

要卸装独立 Identity Console，执行下列操作：

- 1 导航到安装 Identity Console 的计算机上的 `/usr/bin` 目录。

- 2 运行以下命令：

```
./identityconsoleUninstall
```

- 3 Identity Console 已成功卸载。

注释： 当 eDirectory 或其他 NetIQ 产品安装在机器中时，用户必须手动卸载 `nici` 和 `opensl`。
