# Identity Console
## Installation Guide

**January 2024**

## Legal Notice

# Contents

# About this Book and the Library

The *Identity Console Install Guide* provides information on how to install and manage the NetIQ Identity Console (Identity Console) product. This book defines terminology and includes implementation scenarios.

## Intended Audience

This guide is intended for network administrators.

## Other Information in the Library

The library provides the following information resources:

**Installation Guide**

Describes how to install and upgrade Identity Console. The book is intended for network administrators.

# 1 Planning to Install Identity Console

This chapter explains the system requirements and prerequisites for installing Identity Console. As Identity Console can be run both as a Docker Container or as standalone application, refer to the respective sections for system requirements and prerequisites for both types of installation.

---

**NOTE:** Identity Console supports eDirectory 9.2.4 HF2, Identity Manager Engine 4.8.3 HF2, and their respective later versions. You must upgrade your eDirectory and Identity Manager Engine instances before using the Identity Console.

---



## System Requirements and Prerequisites for Docker Installation

This section explains the system requirements and prerequisites for installing Identity Console as Docker container.



### System Requirements

As Identity Console can be run as a Docker container, for more information about system requirements and supported platforms for installing Identity Console, see Docker Documentation.

### Prerequisites

- ❒ Install Docker 20.10.9-ce or later. For more information on how to install Docker, see Docker Installation.
- ❒ You must obtain a pkcs12 server certificate with the private key to encrypt/decrypt data exchange between the Identity Console server and the back-end server. This server certificate is used to secure the http connection. You can use server certificates generated by any external CA. For more information, see Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/netiq-edir_admin/data/b1j4tpo3.html#b1j4u0cm). The server certificate should contain the Subject Alternative Name with IP address and DNS of the Identity Console server. Once the server certificate object is created, you must export it in .pfx format.

- ❒ You must obtain a CA certificate for all the trees in `.pem` format to validate the CA signature of the server certificates obtained in the previous step. This rootCA certificate also ensures establishing a secured ldap communication between the client and the Identity Console server. For example, you can obtain the eDirectory CA certificate (SSCert.pem) from `/var/opt/novell/eDirectory/data/SSCert.pem`.
- ❒ (Optional) Using the One SSO Provider (OSP), you can enable the single sign-on authentication for your users to the Identity Console portal. You must install OSP before installing Identity Console. To configure OSP for Identity Console, follow the on-screen prompts and provide the required values for configuration parameters. To register Identity Console to an existing OSP server, you must manually add the following to the `ism-configuration.properties` file in `/opt/netiq/idm/apps/tomcat/conf/` folder:

  ```
  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
  ```

  **NOTE:** ◆With OSP, you can connect to only a single eDirectory tree as OSP does not support multiple eDirectory trees.

  - ◆ Third party OSP is not supported in Identity Console.
  - ◆ In a NAM integrated environment, Identity Console with OSP is currently not supported.

- ❒ Ensure that you have a proper DNS entry available for your host machine in `/etc/hosts` with a fully qualified host name.
- ❒ If you want to use Identity Console in Edge browser, you must download the latest version of Microsoft Edge for full functionality.

**NOTE:** While using Identity Console in Mozilla Firefox, the operation might fail with `Origin Mismatch` error message. To troubleshoot, perform the following steps:

1. Update Firefox to the latest version.
2. Specify `about:config` in the Firefox URL field and press Enter.
3. Search for Origin.
4. Double-click on `network.http.SendOriginHeader` and change its value to 1.

## Setting Up Your Environment

You might need to create a configuration file containing certain parameters. If you want to configure Identity Console with OSP, you must specify the OSP specific parameters in the configuration file. For example, create the below `edirapi.conf` file with OSP parameters:

**NOTE:** You must provide your eDirectory tree name in the `osp-redirect-url` field.

```
listen = ":9000"
ldapserver = "192.168.1.1:636"
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"
```

In case, you want to configure Identity Console without OSP, create a configuration file as shown below, without the OSP parameters:

```
listen = ":9000"
pfxpassword = "novell"
ospmode = "false"
bcert = "/etc/opt/novell/eDirAPI/cert/"
edir-hosts = "<ip_address-1>:636,<ip_address-2>:636"
```

**NOTE:** When you want to configure Identity Console with multiple eDirectory trees, you can skip "*ldapserver*", "*ldapuser*", and "*ldappassword*" parameters and create the configuration file.

***Table 1-1*** *Description of the configuration parameters in the configuration file*

| Configuration Parameters | Description |
|---|---|
| listen | Specify 9000 as the Identity Console server's listener port inside the container. |
| ldapserver | Specify the eDirectory host server IP and port number. |
| ldapuser | Specify the username of the eDirectory user. This parameter is used as a credential for initiating ldap calls to eDirectory using proxied authorization control in the case of OSP login. Ldap user must have supervisor rights on the eDirectory tree. |
| ldappassword | Specify the password of the LDAP user. |

| Configuration Parameters | Description |
|---|---|
| pfxpassword | Specify the password of the pkcs12 server certificate file. |
| ospmode | Specify `true` to integrate OSP with Identity Console. If you set this to `false`, Identity Console will use ldap login. |
| osp-token-endpoint | This URL is used to fetch certain attributes from the OSP server to verify the validity of the authentication token. |
| osp-authorize-url | This URL is used by the user to provide credentials to obtain an authentication token. |
| osp-logout-url | Use this URL to terminate the session between the user and the OSP server. |
| osp-redirect-url | The OSP server re-directs the user to this URL after granting the authentication token.<br><br>**NOTE:** Ensure to specify the eDirectory tree name in lowercase while configuring Identity Console. In case, the tree name is not specified in lowercase, the login to the Identity Console server might fail. |
| osp-client-id | Specify the OSP client ID which was provided at the time of the Identity Console registration with OSP. |
| ospclientpass | Specify the OSP client password which was provided at the time of the Identity Console registration with OSP. |
| ospcert | Specify the location of OSP server's CA certificate. |
| bcert | Specify location of Identity Console's CA certificate. |
| loglevel | Specify the log levels that you want to include in the log file. This parameter can be set to "fatal", "error", "warn" or "info". |
| check-origin | If this is set to `true`, the Identity Console server compares the origin value of requests. Available options are either `true` or `false`. The *origin* parameter is mandatory even if *check-origin* parameter value is set to `false` when DNS configuration is used. |
| origin | Identity Console compares the origin value of requests with the values specified in this field.<br><br>**NOTE:** From Identity Console 1.4 onward, this parameter is independent of *check-origin* parameter and is mandatory if DNS configuration is used. |
| maxclients | Maximum number of concurrent clients who can access `IDConsole`. Any additional clients beyond this limit have to wait in queue. |

| Configuration Parameters | Description |
|---|---|
| edir-hosts | `edir-host` is a parameter that contains the IP address(s) of eDirectory which you want to connect through Identity Console. |

**NOTE:**

- The `ospmode` configuration parameter should be used only if you plan to integrate OSP along with Identity Console.
- If Identity Applications (Identity Apps) is configured in cluster mode in your Identity Manager setup, you must provide the DNS name of the load balancer server in `osp-token-endpoint`, `osp-authorize-url` and `osp-logout-url` fields in the configuration file. In case, you provide the OSP server details in these fields, the Identity Console login will fail.
- If Identity Console is configured with the same OSP instance as Identity Apps and Identity Reporting, the Single Sign-On (authentication service) will take effect when you are logging into the Identity Console portal.
- OSP HTTPS URL should be validated with certificates containing 2048 bit key or higher with Identity Console 1.4 onwards.
- If you want to restrict the access to the Identity Console portal from different domains, set `samesitecookie` parameter to `strict`. If you want to allow access to the Identity Console portal from different domains, set `samesitecookie` parameter to `lax`. If the parameter is not specified during the configuration, the browser settings will be honored by default.

Once you are ready with the configuration file, proceed with deploying the container. For more information, see “Deploying Identity Console as Docker Container” on page 26.

# System Requirements and Prerequisites for Standalone Installation (Non-Docker)



## System Requirements

This section explains the system requirements and prerequisites to install standalone Identity Console.

| Category | Minimum Requirement |
|---|---|
| Processor | 1.4 GHz 64-bit |
| Memory | 2GB |
| Disk Space | 200 MB on Linux |

| Category | Minimum Requirement |
| --- | --- |
| Supported Browser | ◆ Latest version of **Microsoft Edge**<br>◆ Latest version of **Google Chrome**<br>◆ Latest version of **Mozilla Firefox**<br><br>**NOTE:** While using Identity Console in Mozilla Firefox, the operation might fail with `Origin Mismatch` error message. To troubleshoot, perform the following steps:<br><br>**1** Update Firefox to the latest version.<br>**2** Specify `about:config` in the Firefox URL field and press Enter.<br>**3** Search for Origin.<br>**4** Double-click on `network.http.SendOriginHeader` and change its value to 1. |
| Supported Operating System | ◆ **Certified**:<br>◆ SUSE Linux Enterprise Server (SLES) 15 SP5<br>◆ Red Hat Enterprise Linux (RHEL) 8.7, 8.8, 9.2, and 9.3<br>◆ In Docker: Red Hat Universal Base Image 9.4<br><br>◆ **Supported:** Supported on later versions of support packs of the above certified Operating Systems. |

| Category | Minimum Requirement |
| --- | --- |
| Certificates | • Obtain a `pkcs12` server certificate with the private key to encrypt/decrypt data exchange between the client and the Identity Console server. From Identity Console 1.7.2 onwards users can generate server certificate during the installation process. This server certificate is used to secure the http connection. You can use server certificates generated by any external CA. For more information, see Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/netiq-edir_admin/data/b1j4tpo3.html#b1j4u0cm). The server certificate should contain the Subject Alternative Name with IP address and DNS of the Identity Console server. Once the server certificate object is created, you must export it in .pfx format.<br>• Obtain a CA certificate for all trees in `.pem` format to validate the CA signature of the server certificates obtained in the previous step. From Identity Console 1.7.2 onwards users can import CA Certificate (`SSCert.pem`) by providing IP address and port information during installation process. This rootCA certificate also ensures establishing a secured ldap communication between the client and the Identity Console server. For example, you can obtain the eDirectory CA certificate (`SSCert.pem`) from `/var/opt/novell/eDirectory/data/SSCert.pem`. |

Once you are ready, proceed with installing Identity Console. For more information, see “Deploying Standalone Identity Console (Non-Docker)” on page 17.

## (Optional) Prerequisite for OSP Configuration

Using the One SSO Provider (OSP), you can enable the single sign-on authentication for your users to the Identity Console portal. You must install OSP before installing Identity Console. To configure OSP for Identity Console, follow the on-screen prompts and provide the required values for configuration parameters. To register Identity Console to an existing OSP server, you must manually add the following to the `ism-configuration.properties` file in `/opt/netiq/idm/apps/tomcat/conf/` folder:

```
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
```

**NOTE:**

- If you are installing OSP for the first time, specify the option `'y'` for **Configure OSP with eDir API** and follow the on-screen prompts to register Identity Console with OSP.
- Ensure to specify the eDirectory tree name in lowercase while configuring Identity Console. In case, the tree name is not specified in lowercase, the login to the Identity Console server might fail.
- With OSP, you can connect to only a single eDirectory tree as OSP does not support multiple eDirectory trees.
- Third party OSP is not supported in Identity Console.
- In a NAM integrated environment, Identity Console with OSP is currently not supported.

# System Requirements and Prerequisites for Workstation


- "System Requirements" on page 14


## System Requirements

This section explains the system requirements and prerequisites to run workstation Identity Console (version 1.5 and later).

| Category | Minimum Requirement |
|---|---|
| Processor | 1.5 GHz 64-bit |
| Memory | 2GB |
| Disk Space | 1 GB on Windows |
| Supported Operating System | • **Certified**:<br>  • Windows Server 2019<br>  • Windows Server 2022<br>  • Windows 10<br>  • Windows 11 |

| Category | Minimum Requirement |
| --- | --- |
| Certificates | • You must obtain a server certificate in pfx format to exchange data between the Identity Console client and the REST server. This server certificate must always be named keys.pfx. For more information, see Creating Server Certificate Objects (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/b1j4tpo3.html#b1j4u0cm).<br>• From Identity Console 1.7.2 onwards users can generate Server Certificate during the installation process.<br>• You must obtain a CA certificate for all trees in `.pem` format to validate the CA signature of the server certificates obtained in the previous step. This root CA certificate also ensures establishing a secured `ldap` communication between the client and the Identity Console server.<br>By default, the eDirectory CA certificate for Linux is present at `/var/opt/novell/eDirectory/data/SSCert.pem`.<br>By default, the eDirectory CA certificate `SSCert.pem` for Windows is present at `<eDirectory installed Location>\NetIQ\eDirectory\DIBFiles\CertServ\SSCert.pem`.<br>• From Identity Console 1.7.2 onwards users can import CA Certificate (`SSCert.pem`) by providing IP address and port information during installation process. |

Once you are ready, proceed with deploying Identity Console. For more information, see "Deploying Identity Console Workstation on Windows" on page 23.

# RPM Signature Verification

Use the following steps to perform the RPM Signature Verification:

1. Navigate to the folder where the build is extracted.

   For example: `<untarred location of Identity Console>/IdentityConsole_<version>_Linux/license/MicroFocusGPGPackageSign.pub`.

2. Run the following command to import the Public Key:

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

3. (Optional) Run the following command to verify the RPM signature: `rpm --checksig -v <RPM Name>`

For Example:

```
rpm --checksig -v identityconsole-1.8.0.0000.x86_64.rpm

identityconsole-1.8.0.0000.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 Deploying Identity Console

This chapter covers the deployment of the Identity Console and provides security recommendations. Before the deployment, review the prerequisites and system requirements in Chapter 1, “Planning to Install Identity Console,” on page 7.



## Deploying Standalone Identity Console (Non-Docker)

You can deploy the Identity Console in one of the following ways:

- Interactive Installation
- Silent Installation

---

**NOTE:** NetIQ recommends that when installing Identity Console and eDirectory on the same machine, the machine have at least one instance of eDirectory available.

---

### Performing an Interactive Installation

This section covers how to deploy standalone Identity Console using the interactive installation method.

1. Log in to the Software License and Download (https://sld.microfocus.com/) portal and navigate to the Software Downloads page.
2. Select the following:
   - Product: eDirectory
   - Product Name: eDirectory per User Sub SW E-LTU
   - Version: 9.2
3. Download and extract the latest Identity Console build.
4. Navigate to the directory where you extracted the Identity Console build > `IdentityConsole_<version>_Linux`.
5. Run the following command while logged in as root or root-equivalent user:

   ```
   ./identityconsole_install
   ```

6. Read the Introduction, and then click **ENTER**.
7. Enter `'y'` to accept the License Agreement. This will install all the required RPMs on your system.

8 Enter the Identity Console server's hostname (FQDN)/IP address. For example, `10.10.33.100`

If you press **Enter** without specifying an IP address, your system's IP address/hostname will be used by default.

9 Enter the port number for Identity Console to listen. The default value is `9000`.

10 (Conditional) Do one of the following depending on your requirement:

- If you do not want to integrate OSP with Identity Console, choose "n" and continue with Step 11.
- If you want to integrate OSP with Identity Console, choose "y" and provide inputs for the following steps:
  1. Enter the eDirectory/Identity Vault server's Domain name/IP address with LDAPS port number.

     For example:

     `192.168.1.1:636`
  2. Enter the eDirectory/Identity Vault username.

     Example:

     `cn=admin,ou=org_unit,o=org`
  3. Enter the eDirectory/Identity Vault password.
  4. Enter the eDirectory/Identity Vault password again to confirm the password.
  5. Enter the OSP server domain name/IP address with SSO server SSL port number.
  6. Enter the OSP client ID and OSP client password.
  7. Enter the eDirectory/Identity Vault tree name.

11 Specify which `eDirectory-hosts` to connect. You can provide either IP address or domain name. For example: `localhost:636,xx.xx.xx.xx:636 or edir.domain.com:636`

If you want Identity Console to connect to multiple eDirectory trees, enter their IP addresses or domain names separated by commas.

---

**NOTE:** If you want Identity Console to connect to multiple eDirectory trees, enter the IP addresses or domain names separated by commas.

---

12 (Conditional) Do one of the following to import the CA certificate:

- If you want to import the CA certificate from the server, input "y" and press **Enter**. Then, enter the eDirectory server domain name/IP address with LDAPS port number.

  For example: `10.10.10.10:636`
- If you do not want to import the CA certificate from the server, input "n" and press **Enter**. Then, provide the location of your CA certificate directory path manually.
- If you enter "q", the installation will be terminated.

13 (Conditional) Do one of the following to generate a Server Certificate:

- If you want to generate the Server Certificate, input "y" and press **Enter**. Provide inputs for the following steps:
  1. Enter the eDirectory server domain name or IP address with LDAPS port number.
  2. Enter the eDirectory user name. Example: `cn=admin,o=novell`.
  3. Enter the eDirectory user password.

4. Re-enter the eDirectory user password.
5. Enter the server certificate name.

   Example: `servercert`
6. Enter the server certificate password.

   Example: `password@123`
7. Re-enter the server certificate password.

   Example: `password@123`

- If you already have a Server Certificate that you want to use, input “n” and press **Enter**. Provide inputs for the following steps:
  1. Specify the location of your Server Certificate directory path manually.

     For example, `/home/cert/keys.pfx`
  2. Enter the server certificate password.
  3. Re-enter the server certificate password.

---

**NOTE:**

- You can find the following log files in the `/var/opt/novell/eDirAPI/log` directory:
  - `edirapi.log` - This file logs edirapi events and debugging issues.
  - `edirapi_audit.log` - This file logs edirapi audit events. The logs follow a CEF auditing format.
  - `identityconsole_install.log` - This file logs Identity Console events.
- You can check the logs for Identity Console start and stop operations in the `/var/log/messages` file.
- When you are generating the CA certificate and Server Certificate, make sure to run the Identity Console installer from the `IdentityConsole_<version>_Linux` directory in the extracted location.
- If installation fails, uninstalling the existing Identity Console is not required, instead the user can run the following command:

  `/usr/bin/identityconsoleConfigure`

---

# Utilities to Generate Certificates

You have the option to obtain CA Certificates and Server Certificates for other trees using the following tools or utilities.

## Generate CA Certificate

1. Download and extract the latest Identity Console build.
2. Navigate to the directory where you extracted the Identity Console build > `IdentityConsole_<version>_Linux`.
3. Run the following command while logged in as root or root-equivalent user.

```
./get_cacert
```

4 Enter the eDirectory IP address with LDAPS port number. For example: `10.10.10.10:636`.

Trusted root certificate(s) copied successfully from server to `/tmp/SSCert.pem`.

### Generate Server Certificate

1 Download and extract the latest Identity Console build.

2 Navigate to the directory where you extracted the Identity Console build > `IdentityConsole_<version>_Linux`.

3 Run the following command while logged in as root or root-equivalent user.

```
./get_servercert <eDirectory IP address with LDAPS port number>
<eDirectory/Identity Vault username> <userpassword> <server certificate
name> <server certificate password> <path_of_CA _certificate with
filename> <path of server certificate with filename>
```

Example:

```
./get_servercert 10.10.10.10:636 cn=admin,ou=org_unit,o=org password
keys password /var/opt/novell/eDirectory/data/SSCert.pem /home/user/
keys.pfx
```

## Performing a Silent Installation

Silent installation enables you to install Identity Console without any interactive input. To use this method, you must define your options for installing Identity Console in the `silent_properties` file and then run the installation process from the command line. To create the `silent_properties` file, you can use the `create_silent_properties` utility that comes with Identity Console version 1.7.2 and later. After you generate the properties file, the system uses the information from it to complete the installation silently.

Before starting the silent installation, ensure that you meet all the prerequisites.

**To generate the silent properties file:**

1 Navigate to the `IdentityConsole_<version>_Linux` directory in the location where you have extracted the Identity Console build.

2 Run the following command to use the `create_silent_properties` utility:

```
./create_silent_properties
```

3 Enter the Identity Console server hostname or IP address. For example, `10.0.0.1`.

4 Enter the port number on which you want Identity Console to listen. For example, `9000`.

5 (Conditional) Do one of the following depending on your requirement:

- If you do not want to integrate OSP with Identity Console, choose “n” and continue with Step 6.

- If you want to integrate OSP with Identity Console, choose “y” and provide inputs for the following steps:
    1. Enter the eDirectory/Identity Vault server’s Domain name/IP address with LDAPS port number.

       For example:

       `192.168.1.1:636`
    2. Enter the eDirectory/Identity Vault username.

       Example:

       `cn=admin,ou=org_unit,o=org`
    3. Enter the eDirectory/Identity Vault password.
    4. Enter the eDirectory/Identity Vault password again to confirm the password.
    5. Enter the OSP server domain name/IP address with SSO server SSL port number.
    6. Enter the OSP client ID and OSP client password.
    7. Enter the eDirectory/Identity Vault tree name.

6. Enter the eDirectory server host names or IP address to which you want Identity Console to establish a connection. For example, `10.10.10.10:636`.
7. Enter the eDirectory server host names or IP address to which you want Identity Console to establish a connection. For example, `10.10.10.10:636`.
8. Enter the directory path of the server certificate, including the filename. For example, `/home/cert/keys.pfx`.
9. Enter the server certificate password.
10. Re-enter the server certificate password.

    The `silent_properties` file is generated successfully in `IdentityConsole_<version>_Linux` location.

**To perform a silent installation:**

1. After generating the `silent_properties` file, run the following command to run the installer in silent mode:

   ```
   ./identityconsole_install -s silent_properties
   ```
2. Enter the eDirectory server Domain name/IP address with LDAPS port number. For example: `10.10.10.10:636`

You can find installation-related logs in the `Identity Console installation` > `identityconsole_install.log` file.

# Multi-tree with Standalone Identity Console

To connect Identity Console with multiple eDirectory trees, you must provide eDirectory IP and LDAPS port separated commas in the `edirapi.conf` file located at `/etc/opt/novell/eDirAPI/conf/`. Also, you must copy the CA certificates from all the eDirectory trees to the `/etc/opt/novell/eDirAPI/cert/` directory.

For example, to connect Identity Console to three eDirectory trees, provide the IP address in the following format:

```
edir-hosts="10.0.0.1:636,10.0.0.2:636,10.0.0.3:636"
```

Then copy the CA certificates as follows:

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

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

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

Run one of the following command to restart Identity Console:

- `/usr/bin/identityconsole restart`
- `systemctl restart netiq-identityconsole.service`

## Modifying Server Certificate in Standalone Identity Console

Perform the following steps to modify server certificate in Standalone Identity Console:

1. Run NLPCERT to store the keys:

   ```
   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. Restart the Identity Console:

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

## Stopping and Restarting Standalone Identity Console

- To stop Identity Console, run one of the following command:

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

  or

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

- To restart Identity Console, run one of the following command:

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

  or

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

- To start Identity Console, run one of the following command:

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

  or

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

# Deploying Identity Console Workstation on Windows

Identity Console (version 1.5 and later) can be launched on Windows as workstation, and requires the REST services running. Therefore, when it is launched, an eDirAPI process runs in the `edirapi.exe` cmd prompt. If you close `edirapi.exe` terminal, Identity Console will no longer function.

The following procedure describes how to run Identity Console on Windows.

1. Log in to the Software License and Download (https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0) portal and navigate to the Software Downloads page.
2. Select the following:
   - Product: eDirectory
   - Product Name: eDirectory per User Sub SW E-LTU
   - Version: 9.2
3. Download and extract the `IdentityConsole_<version>_workstation_win_x86_64.zip`.
4. Navigate to the extracted folder, install `NICI_w32` or `NICI_wx64` depending on your system configuration.
5. (Conditional) If you already have the `keys.pfx` and `SSCert.pem` files that you want to use, then copy the files manually into `cert` folder located in `C:\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\cert`.

   Then proceed to step 9.
6. (Conditional) If you want to generate the CA certificate, navigate to the `eDirAPI` folder, run the `get_cacert.exe` binary in power shell, located in `C:\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI`, and provide the following details:
   - eDirectory/Identity Vault server Domain name/IP address with LDAPS port number.
   - Trusted-root CA certificate path with certificate name.

   For example,

   ```
   get_cacert.exe 10.10.10.125:636
   C:\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\cert\SSCert.pem
   ```

   A `SSCert.pem` file is generated.
7. (Conditional) If you want to generate the Server Certificate, run `get_servercert.exe` binary through power shell along with the following details:
   - eDirectory/Identity Vault server Domain name/IP address with LDAPS port number.
   - eDirectory/Identity Vault user name.
   - eDirectory/Identity Vault password.
   - Server certificate name.
   - Server certificate password.

- Trusted-root CA certificate path with certificate name.
- Trusted-root server certificate path with certificate name(`keys.pfx` is hard coded).

For example,

```
get_servercert.exe 10.10.10.125:636 cn=admin,o=novell novell keys
novell
C:\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\cert\SSCer
t.pem
C:\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\cert\keys.
pfx
```

A `keys.pfx` file is generated.

8 Copy the files `keys.pfx` and `SSCert.pem` (that are generated in step: 6 and step: 7) manually into `cert` folder.

9 Navigate to the extracted folder, double-click the `configure.bat` file and enter the server certificate (`keys.pfx`) password in the command prompt.

**NOTE:** If the server certificate is changed, then re-run the `configure.bat` file with the password of the new certificate.

10 Navigate to the extracted folder and double-click the `run.bat` file.

The eDirAPI process terminal (edirapi.exe) starts running, and the Identity Console login page appears.

**NOTE:**

- For subsequent logins to the Identity Console application, double click the `run.bat`. The login page will appear.

  If the `eDirAPI` process terminal (edirapi.exe) is already running, then run `identityconsole.exe` from the build extracted folder.
- Users can find the following logs in: `\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\log`

  `edirapi.log` - This is used for logging different events in `edirapi` and debugging issues.

  `edirapi_audit.log` - This is used for logging audit events of `edirapi`. The logs follow CEF auditing format.
- OSP based logins are not supported in workstation mode.
- Identity Console Workstation is listening on port 9000. Do not modify the `edirapi_win.conf` file.

## Utilities to Generate Certificates

You have the option to obtain CA Certificates and Server Certificates for other eDirectory Trees using the following utilities.

### Generate CA Certificate

1. Download and extract the latest Identity Console build.
2. Navigate to the directory where you extracted the Identity Console build. Example: `C:\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI`
3. Run the following binary through command prompt.

   `get_cacert.exe`
4. Provide the eDirectory IP address and LDAPS port number.

   Example: `get_cacert.exe 10.10.10.125:636`

   A `SSCert.pem` file is generated. Copy `SSCert.pem` manually into `cert` folder.

### Generate Server Certificate

1. Navigate to the folder where you have extracted the Identity Console build.

   Example: `IdentityConsole_<version>_win`
2. Run `get_servercert.exe help` for more help options through command prompt.
   - eDirectory/Identity Vault server Domain name/IP address with LDAPS port number.
   - eDirectory/Identity Vault user name.
   - eDirectory/Identity Vault password.
   - Server certificate name.
   - Server certificate password.
   - Trusted-root certificate path with certificate name.

     To generate the Server Certificate, run the following command through command prompt:

     Example:

     ```
     get_servercert.exe 10.10.10.125:636 cn=admin,o=novell novell keys
     novell SSCert.pem
     ```

     A `keys.pfx` file is generated. Copy `keys.pfx` manually into `cert` folder.

## Multi-tree with Identity Console as Workstation

Identity Console allows user to connect to multiple trees by obtaining individual CA certificate of the tree.

1. Close the Identity Console workstation and eDirAPI terminal.
2. Copy the CA certificates `SSCert.pem` into the location: `IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\cert.`

   For example, if you want to connect to three eDirectory trees, copy the CA certificates as `SSCert1.pem`, `SSCert2.pem` and `SSCert3.pem` respectively.
3. Navigate to the folder where the build is extracted and double click the `run.bat` file (Windows batch file).

## Closing and Re-launching Identity Console Workstation

**To close the application and the process:**

1. Close the Identity Console desktop windows application.
2. Stop the eDirAPI process by closing the `eDirAPI` process terminal.

To relaunch Identity Console Workstation, navigate to the folder where the build is extracted and double click the `run.bat` file (Windows batch file).

---

**NOTE:** If the `eDirAPI` process terminal is already running, then run `identityconsole.exe` from the build extracted folder to relaunch Identity Console Workstation.

---

# Deploying Identity Console as Docker Container

This section includes the following procedures:



## Security Recommendations

- Docker containers do not have any resource constraints by default. This provides every container with the access to all the CPU and memory resources provided by the host's kernel. You must also ensure that one running container should not consume more resources and starve other running containers by setting limits to the amount of resources that can be used by a container.
  - Docker container should ensure that a Hard Limit is applied for the memory used by the container using the `--memory` flag on Docker run command.
  - Docker container should ensure that a limit is applied to the amount of CPU used by a running container using the --`cpuset-cpus` flag on the Docker run command.
- `--pids-limit` should be set to 300 to restrict the number of kernel threads spawned inside the container at any given time. This is to prevent DoS attacks.
- You must set the on-failure container restart policy to 5 using the `--restart` flag on Docker run command.
- You must only use the container once the health status shows as **Healthy** after the container comes up. To check the container's health status, run the following command:

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

- Docker container will always start as non-root user (nds). As an additional security measure, enable user namespace remapping on the daemon to prevent privilege-escalation attacks from within the container. For more information on user namespace remapping, see Isolate containers with a user namespace.

## Deploying Identity Console As a Docker Container

---

**NOTE:** Identity Console can be configured with or without OSP. If you choose to configure it with OSP, you must first deploy the OSP container, followed by the Identity Console container. Make sure to modify the `edirapi.conf` file to include your desired values for deployment.

---

**To deploy Identity Console as a Docker container:**

*The configuration parameters, sample values and examples mentioned in this procedure are for reference purposes only. You must ensure not to use them directly in your production environment.*

1. Log in to the Software License and Download (https://sld.microfocus.com/) portal and navigate to the Software Downloads page.
2. Select the following:
   - Product: eDirectory
   - Product Name: eDirectory per User Sub SW E-LTU
   - Version: 9.2
3. Download the `IdentityConsole_<version>_Container.tar.zip`.
4. The image has to be loaded into the local Docker registry. Extract and load the `IdentityConsole_<version>_Containers.tar.gz` file using the below commands:

   ```
   tar -xvf IdentityConsole_version_Containers.tar.gz
   ```

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

5. Create the Identity Console Docker container using the following command:

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

   For example,

   ```
   docker create --name identityconsole-container-1 --env ACCEPT_EULA=Y --
   network=host --volume IDConsole-volume:/config/ identityconsole:version
   ```

   ---

   **NOTE:**

   - You can accept the EULA by setting ACCEPT_EULA environment variable to `'Y'`. You can also accept the EULA from the on-screen prompt while starting the container by using `-it` option in the Docker create command for interactive mode.
   - `--volume` parameter in the above command will create a volume for storing configuration and log data. In this case, we have created a sample volume called `IDConsole-volume`.

   ---

6 Copy the server certificate file from your local file system to the container as `/etc/opt/novell/eDirAPI/cert/keys.pfx` using the following command. For more information on creating the server certificate, see "Prerequisites" on page 7:

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

For example,

```
docker cp /home/user/keys.pfx identityconsole-container-1:/etc/opt/novell/eDirAPI/cert/keys.pfx
```

When you connect to multiple eDirectory trees, you must ensure to obtain at least one keys.pfx server certificate for all the connected trees.

7 Copy the CA certificate file (`.pem`) from your local file system to the container as `/etc/opt/novell/eDirAPI/cert/SSCert.pem` using the following command. For more information on obtaining the CA certificate, see "Prerequisites" on page 7:

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

For example,

```
docker cp /home/user/SSCert.pem identityconsole-container-1:/etc/opt/novell/eDirAPI/cert/SSCert.pem
```

If the user need to connect to multiple eDirectory trees, refer section: "Multi-tree with Identity Console as Docker" on page 29.

8 Depending on whether you want to configure Identity Console with or without OSP, modify the `edirapi.conf` configuration file as needed. Then use the following command to copy it from your local file system to the container at `/etc/opt/novell/eDirAPI/conf/edirapi.conf`:

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

For example,

```
docker cp /home/user/edirapi.conf identityconsole-container-1:/etc/opt/novell/eDirAPI/conf/edirapi.conf
```

A sample configuration file is shown below:

```
listen = ":9000"
pfxpassword = "novell"
bcert = "/etc/opt/novell/eDirAPI/cert/"
ospmode=false
edir-hosts = "<ip_address>:636"
```

---

**NOTE:** To access the eDirectory through Identity Console, it is required to add the `edir-hosts="x.x.x.x:ldaps_port` in the `edirapi.conf` file.

Example: `edir-hosts="10.10.10.10:636"`

---

A sample configuration file when configuring Identity Console with OSP is shown below:

```
listen = ":9000"
ldapserver = "10.71.39.15:636"
ldapuser = "cn=admin,o=novell"
ldappassword = "novell"
pfxpassword = "novell"
osp-token-endpoint = "https://<osp_ipaddress>:8543/osp/a/idm/auth/
oauth2/getattributes"
osp-authorize-url = "https://<osp_ipaddress>:8543/osp/a/idm/auth/
oauth2/grant"
osp-logout-url = "https://<osp_ipaddress>:8543/osp/a/idm/auth/app/
logout"
osp-redirect-url = "https://<identity_console_ipaddress>:9000/eDirAPI/
v1/t/authcoderedirect"
osp-client-id = "identityconsole"
ospclientpass = "novell"
ospcert = "/etc/opt/novell/eDirAPI/cert/SSCert.pem"
bcert = "/etc/opt/novell/eDirAPI/cert/"
ospmode=true
check-origin =true
origin = "https://<identity_console_ipaddress>:9000"
```

9 Start the Docker container using the following command:

```
docker start identityconsole-container-name
```

For example,

```
docker start identityconsole-container-1
```

---

**NOTE:** You can find the following log files in `/var/lib/docker/volumes/<volume_name>/_data/eDirAPI/var/log` directory:

- `edirapi.log` - This is used for logging different events in edirapi and debugging issues.
- `edirapi_audit.log` - This is used for logging audit events of edirapi. The logs follow CEF auditing format.
- `container-startup.log` - This is used for capturing installation logs of Identity Console Docker container.

---

## Multi-tree with Identity Console as Docker

Identity console allows user to connect to Multiple trees by obtaining individual CA certificate of the tree.

---

**NOTE:** To access the eDirectory through Identity Console, it is required to add the `edir-hosts="x.x.x.x:ldaps_port, y.y.y.y:ldaps_port, z.z.z.z:ldaps_port"` at the eDirectory configuration file.

```
edir-hosts="10.10.10.10:636, 20.20.20.20:636, 30.30.30.30:636"
```

---

For example, if you connect to three eDirectory trees, then you must copy all the three CA certificates in to Docker Container:

```
docker cp /home/user/SSCert1.pem identityconsole-container-1:/etc/opt/
novell/eDirAPI/cert/SSCert1.pem

docker cp /home/user/SSCert2.pem identityconsole-container-1:/etc/opt/
novell/eDirAPI/cert/SSCert2.pem

docker cp /home/user/SSCert3.pem identityconsole-container-1:/etc/opt/
novell/eDirAPI/cert/SSCert3.pem
```

Run the following commands to restart Identity Console:

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

Example:

```
docker restart identityconsole-container-1
```

## Deploying the OSP Container

Perform the following steps to deploy the OSP container:

1. Log in to Software License and Download (https://sld.microfocus.com/) portal and navigate to the Software Downloads page.
2. Select the following:
   - Product: eDirectory
   - Product Name: eDirectory per User Sub SW E-LTU
   - Version: 9.2
   - Identity Console Standalone
3. Download and extract the `IdentityConsole_<version>_Containers_tar.zip` file.
4. To deploy Identity Console in the OSP container, a keystore (`tomcat.ks`) is required.

   Perform the following steps to generate the keystore:

   **4a** Create a folder `certs` in the `/opt/` directory.

   **4b** Run the following command to create a keystore (`tomcat.ks`):

   ```
   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
   ```

   **NOTE:** Ensure that the IP address of the machine is named as CN name or as fully qualified hostname. For example: `CN=xx.xx.xx.xx`

   **4c** Run the following command to create a certificate signing request. For example: `cert.csr`.

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

   **4d** Pass the created `cert.csr` to Identity Console and get the `cert.der` as explained:

   **4d1** Launch Identity Console as Administrator.

   **4d2** Click **Certificate Management** > **Issue Certificate** and select the file.

   **4d3** Go to **Key Usage Specifications** > **Key Type** and select the **Custom** radio button.

**4d4** Click **Key Usage** and select the following check boxes:

- **Data Encipherment**
- **Key Encipherment**
- **Digital Signature**

**4d5** Click **Certificate Parameters** > **Subject Alternative Names** > OSP Server IP address or OSP Server DNS Name > **Next** > **OK**.

The message appears as **Certificate has been generated successfully**.

**4d6** Click **OK**.

**4d7** Download the issued certificate `cert.der` and copy it to `/opt/certs/`.

**4e** Copy the SS`Cert.der` from eDirectory to `/opt/certs`

**4f** Run the following commands to import the CA certificate (SS`Cert.der`) and server certificate (`cert.der`) into the `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
```

**5** Create a new folder as `/data`.

**6** Copy the `tomcat.ks` from `/opt/certs` and paste it to the `data` folder.

**7** From the extracted Identity Console container build, copy the `osp-edirapi-silent.properties` files in to the `data` folder.

**8** Modify the `osp edirapi silent` properties file as per your requirement. A sample silent properties file has been shown below:

```
# 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 (osp confiugred serevr IP address)
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 (IdentityConsole confiugred serevr IP
address)
IDCONSOLE_PORT=9000
EDIRAPI_TREENAME=ed913 (Tree name should be in lowercase)

#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 (eDir/ID_Vault confiugred serevr IP address)
ID_VAULT_LDAPS_PORT=636
ID_VAULT_ADMIN_LDAP="cn=admin,o=novell"
ID_VAULT_PASSWORD=novell
```

**NOTE:** To avoid space constraints while using the silent properties (DOS text) file, you must convert the DOS text file to UNIX format using the dos2unix tool. Run the below command to convert text file from DOS line endings to Unix line endings:

```
dos2unix filename
```

For example:

```
dos2unix samplefile
```

9 Run the following command to load the OSP image:

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

10 Deploy the container using the following command:

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

For example:

```
docker run -d --name OSP_Container --network=host -e
SILENT_INSTALL_FILE=/config/osp-edirapi-silent.properties -v /data:/
config osp:6.6.6
```

**NOTE:** After deploying OSP container, install and configure Identity Console with OSP server details.

## Stopping and Restarting Identity Console As Docker Container

To stop Identity Console, run the following command:

```
docker stop identityconsole-container-name
```

To restart Identity Console, run the following command:

```
docker restart identityconsole-container-name
```

To start Identity Console, run the following command:

```
docker start identityconsole-container-name
```

## Managing Data Persistence

Along with the Identity Console containers, volumes for data persistence are also created. To use the configuration parameters of an old container using the volumes, perform the following steps:

1. Stop your current Docker Container using the following command:

   ```
   docker stop identityconsole-container-name
   ```

   Example:

   ```
   docker stop identityconsole-container-1
   ```

2. Create the second container using the application data of the old container stored in Docker volume (`IDConsole-volume-1`).

   ```
   docker create --name identityconsole-container-name --network=host --
   volume IDConsole-volume-1:/config/ identityconsole:< version >
   ```

   Example:

   ```
   docker create --name identityconsole-container-2 --network=host --
   volume IDConsole-volume-1:/config/ identityconsole:1.7.1.0000
   ```

3. Start the second container using the following command:

   ```
   docker start identityconsole-container-name
   ```

   Example:

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

4. (Optional) The first container can be removed using the following command:

   ```
   docker rm identityconsole-container-name
   ```

   Example:

   ```
   docker rm identityconsole-container-1
   ```

## Modifying Server Certificate in Docker Container

Perform the following steps to modify server certificate in Docker Container:

1. Run the following command to copy the new server certificate in any location of your container.

   Example:

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

2. Login to the container by using the following command:

   ```
   docker exec -it container_name bash
   ```

3. Run the NLPCERT to store the keys as a pseudo-user:

```
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 Exit the container console using the command:

```
exit
```

5 Restart the container by entering:

```
docker restart container name
```

# Deploying Identity Console In Azure Kubernetes Services

Azure Kubernetes Service (AKS) is a managed Kubernetes service that enables you to deploy and manage clusters. This section includes the following procedures:

## Deploying Identity Console in AKS Cluster

This section explains the following procedures to deploy Identity Console in AKS Cluster:



### Creating an Azure Container Registry (ACR)

Azure Container Registry (ACR) is an Azure-based, private registry, for Docker container images.

For more detail steps see Create an Azure container registry using the Azure portal section in the Create container registry - Portal or perform the following steps to create an Azure Container Registry (ACR):

1. Sign in to Azure Portal.
2. Go to **Create a resource** > **Containers** > **Container Registry**.
3. In the **Basics** tab, specify values for **Resource group** and **Registry name**. The registry name must be unique within Azure and contain minimum of 5 and maximum of 50 alphanumeric characters.

   Accept default values for the remaining settings.
4. Click **Review + create**.
5. Click **Create**.
6. Sign in to Azure CLI, run the following command to log in to Azure Container Registry.

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

   Example:

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

7. Retrieve the login server of the Azure Container Registry using the command:

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

   Example:

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

8. Tag the local image of Identity Console with the name of the ACR login server (registryname.azureacr.io) using the following command:

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

   Example:

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

9. Push the tagged image to the registry.

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

   Example:

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

10. Retrieve the list of images in the registry using the command:

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

## Setting a Kubernetes cluster

Create a kubernetes service resource using Azure portal or CLI.

For more detail steps to create a Kubernetes service resource in azure with a node, see Create an AKS Cluster in the Azure Quickstart.

---

**NOTE:**

- Ensure to select Azure CNI as network.
- Select the existing virtual network (where the eDirectory server is deployed in the subnet).
- Select the existing container registry where Identity Console image is available.

---

## Creating a standard SKU public IP address

A Public IP address resource under Kubernetes cluster resource group acts as load Balancer IP for the application.

For detail steps, see the Create a public IP address using the Azure portal in the Create public IP address – Portal.

## Setting Up Cloud Shell and Connecting to Kubernetes Cluster

Use cloud Shell which is available in azure portal for all operations.

To setup cloud shell in Azure portal see Start Cloud Shell section in Bash – Quickstart or perform the following steps to set Up Cloud Shell and connect to Kubernetes Cluster:

1. In the Azure portal, click the  button to Open Cloud Shell.

   **NOTE:** To manage a Kubernetes cluster, use the Kubernetes command-line client, `kubectl`. kubectl is already installed if you use Azure Cloud Shell.

2. Configure kubectl to connect to your Kubernetes cluster using the following command:

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

   Example:

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

3. Verify the list of the cluster nodes using the command:

   ```
   kubectl get nodes
   ```

## Deploying the Application

To deploy Identity Console, you can use `idc-services.yaml` , `idc-statefulset.yaml`, `idc-storageclass.yaml` and `idc-pvc.yaml` sample files.

You can also create your own yaml files as per the requirement.

1. Create a storage class resource using below command:

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

   Example:

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

   (Optional) For more information on how to dynamically create and use persistence volume with azure files share, see Dynamically create and use a persistent volume with Azure Files in Azure Kubernetes Service (AKS)

   A sample storage class resource file has been shown below:

```
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
~
```

A storage class resource enables dynamic storage provisioning. It is used to define how an Azure file share is created.

2. View the details of storageclass using below command:

   ```
   kubectl get sc
   ```

3. Create a pvc resource using `idc-pvc.yaml` file:

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

   Example:

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

   A sample pvc resource file has been shown below:

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

   A persistence volume claim resource creates the file share. A persistent volume claim (PVC) uses the storage class object to dynamically provision an Azure file share.

4. Upload the edirapi.conf, CA cert, and the server certificate to the cloud shell.

   Click the **Upload/Download files** button icon  on cloud shell and upload `edirapi.conf`, `SSCert.pem` and `keys.pfx` files.

**NOTE:** edirapi.conf has a parameter “origin”. Here we need to provide IP address with which we will access Identity Console application. (use the IP address which is created in the “Creating a standard SKU public IP address” on page 36 section.)

Identity Console deploy requires server certificate(`keys.pfx`).

While creating server certificate make sure to provide valid DNS name in subject Alternative Name.

Steps to build a valid DNS name:

A typical pod deployed using StatefulSet has DNS name like below - {statefulsetname}-{ordinal}.{servicename}.{namespace}.svc.cluster.local

- If StatefulSet name in idconsole-statefulset.yaml file is idconsole-app then statefulsetname = idconsole-app
- If it is 1st pod, then ordinal = 0
- If you define serviceName in idconsole -statefulset.yaml file as idconsole then serviceName = idconsole
- If it is by default namespace, then namespace=default

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

5. Create a configmap resource in Kubernetes cluster which stores the configuration files along with the certificates.

   Before running the command make sure that files (`edirapi.conf`, `SSCert.pem` and `keys.pfx`) are present in the directory.

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

   Example:

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

6. View the details of the configmap object, using kubectl describe command:

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

   Example:

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

7. Create StatefulSet resource to deploy container.

   Run the below command to deploy the container:

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

   Example:

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

   A sample StatefulSet resource file has been shown below:

```
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. Run the following command to verify the status of the deployed pod:

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

9. Create Service resource of type loadBalancer.

   The type of service specified in yaml file is of loadBalancer.

   Create a service resource using the below command:

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

   Example:

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

   A sample service resource file has been shown below:

```
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
```

Check the EXTERNAL-IP address (or the loadBalancerIP) using the below command:

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

10. Launch url using EXTERNAL-IP (or the loadBalancerIP address).

    Example:

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

# 3 Upgrading Identity Console

This chapter describes the process for upgrading Identity Console to its latest versions. To prepare for the upgrade, review the prerequisites and system requirements provided in Chapter 1, “Planning to Install Identity Console,” on page 7.

This section includes the following procedures:



## Upgrading Identity Console on Standalone Server

This section explains the procedure to upgrade standalone Identity Console:

1. Log in to the Software License and Download (https://sld.microfocus.com/) portal and navigate to Software Download SLD page.
2. Navigate by selecting the Product: **eDirectory** > Product Name: **eDirectory per User Sub SW E-LTU** > Version: **9.2**
3. Download the latest `IdentityConsole_<version>_Linux.tar.gz` build.
4. Extract the downloaded build by using the following command:

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

5. Navigate to the folder where you extracted the Identity Console build.
6. Run the following command:

   ```
   ./identityconsole_install
   ```

7. Specify the previously configured `eDirectory-hosts` to connect. You can provide either an IP address or a domain name. Example: `localhost:636,xx.xx.xx.xx:636` or `edir.domain.com:636`

   Identity Console gets upgraded successfully.

# Upgrading Identity Console Workstation on Windows

The following procedure describes how to upgrade Identity Console (version 1.5 and later) workstation on Windows.

1. Before upgrading to the new version, ensure to uninstall the previous version as explained: "Uninstallation of Identity Console Workstation on Windows" on page 48.
2. Log in to the Software License and Download (https://sldlogin.microfocus.com/nidp/idff/sso?id=5&sid=0&option=credential&sid=0) portal and navigate to the Software Downloads page.
3. Select the following:
   - Product: eDirectory
   - Product Name: eDirectory per User Sub SW E-LTU
   - Version: 9.2
4. Download the latest version: `IdentityConsole_<version>_workstation_win_x86_64.zip`.
5. Extract the downloaded `IdentityConsole_<version>_workstation_win_x86_64.zip`.
6. Navigate to the extracted folder: `IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\cert`, and copy the `cert` folder that was taken as backup during "Uninstallation of Identity Console Workstation on Windows" on page 48.

   To get the certificates, refer to the section: "System Requirements and Prerequisites for Workstation" on page 14.

   If the user need to connect to multiple eDirectory trees, refer to the section: "Multi-tree with Identity Console as Workstation" on page 25.

   **NOTE:** The server certificate name must always be as `keys.pfx`.
7. Navigate to the folder where the build is extracted and install NICI_w32 or NICI_wx64 depending on the system configuration
8. Double-click on the file `configure.bat` > enter the server certificate (`keys.pfx`) password in the command prompt.
9. Double-click on the file `run.bat` (Windows batch file).

   The eDirAPI process terminal (edirapi.exe) starts running, and the Identity Console login page appears.

**NOTE:**

- If the `eDirAPI` process terminal (edirapi.exe) is already running, then run `identityconsole.exe` from the build extracted folder.
- Users can find the following logs in: `\IdentityConsole_<version>_workstation_win_x86_64\eDirAPI\log`

`edirapi.log` - This is used for logging different events in `edirapi` and debugging issues.

`edirapi_audit.log` - This is used for logging audit events of `edirapi`. The logs follow CEF auditing format.

- OSP based login is not supported in workstation mode.
- Identity Console Workstation is listening on `9000 port`. Do not modify the `edirapi_win.conf` file.

---

# Upgrading Identity Console As Docker Container

When a new version of Identity Console Image is available, the administrator can perform an upgrade procedure to deploy container with the latest version of Identity Console. Ensure to store all necessary application related data persistently in Docker volumes before performing an upgrade. Perform the following steps to upgrade Identity Console using Docker Container:

1. Download and load the latest version of the Docker image from the Software License and Download (https://sld.microfocus.com/) and perform the steps to install the latest version Identity Console as mentioned in "Deploying Identity Console" on page 17.
2. Once the latest Docker image is loaded, stop your current Docker Container using the following command:

   ```
   docker stop identityconsole-container-name
   ```

   Example:

   ```
   docker stop identityconsole-container-1
   ```

3. Delete the existing Identity Console container by running the following command:

   ```
   docker rm identityconsole-container-name
   ```

   For example,

   ```
   docker rm identityconsole-container-1
   ```

4. (Optional) Delete the obsolete Identity Console Docker image by running the following command:

   ```
   docker rmi identityconsole-image
   ```

   Example:

   ```
   docker rmi identityconsole:1.7.0.0000
   ```

5. The image has to be loaded into the local Docker registry. Extract and load the `IdentityConsole_<version>_Containers.tar.gz` file using the below commands:

   ```
   tar -xvf IdentityConsole_version_Containers.tar.gz
   ```

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

6. Create the Identity Console Docker Container using the following command:

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

For example:

```
docker create --name identityconsole-container-2 --env ACCEPT_EULA=Y --
network=host --volume IDConsole-volume:/config/
identityconsole:1.7.1.0000
```

---

**NOTE:**

- You can accept the EULA by setting ACCEPT_EULA environment variable to `'Y'`. You can also accept the EULA from the on-screen prompt while starting the container by using `-it` option in the Docker create command for interactive mode.
- `--volume` parameter in the above command will create a volume for storing configuration and log data. In this case, we have created a sample volume called `IDConsole-volume`.

---

**7** Copy the configuration file (`edirapi.conf`) from your local file system to the newly created container as `/etc/opt/novell/eDirAPI/conf/edirapi.conf` using the following command:

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

For example,

```
docker cp /home/user/edirapi.conf identityconsole-container-2:/etc/opt/
novell/eDirAPI/conf/edirapi.conf
```

A sample configuration file is shown below:

```
listen = ":9000"
pfxpassword = "novell"
bcert = "/etc/opt/novell/eDirAPI/cert/"
ospmode=false
edir-hosts = "<ip_address-1>:636,<ip_address-2>:636"
```

---

**NOTE:** ◆While upgrading to Identity Console 1.7.2 and above, it is required to add eDirectory server IP to `edirapi.conf` file before copying it to the container.

- If you want Identity Console to connect to multiple eDirectory trees, enter their IP addresses or domain names separated by commas

---

**8** Start the second container using the following command:

```
docker start identityconsole-container-name
```

Example:

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

**9** To check status of the running container, run the following command:

```
docker ps -a
```

# Upgrading the OSP Container

Perform the following steps to upgrade the OSP container:

1. Download and load the latest version of the OSP image from the Software License and Download (https://sld.microfocus.com/).

   For example,

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

2. Once the latest OSP image is loaded, stop your current OSP container using the following command:

   ```
   docker stop OSP container name
   ```

3. (Optional) Take the backup of the shared volume.
4. Delete the existing OSP container by running the following command:

   ```
   docker rm OSP container name
   ```

   For example,

   ```
   docker rm OSP_Container
   ```

5. Go to the directory that contains keystore (`tomcat.ks`) and silent properties file, delete the existing keystore (`tomcat.ks`) and retain the existing OSP folder. Generate a new keystore (`tomcat.ks`) with key size as 2048. For more information, see Step 4 in the "Deploying the OSP Container" on page 30.
6. Deploy the container using the following command:

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

   For example,

   ```
   docker run -d --name OSP_Container --network=host -e
   SILENT_INSTALL_FILE=/config/osp-edirapi-silent.properties -v /data:/
   config osp:6.6.6
   ```

# 4 Uninstalling Identity Console

This chapter describes the process for uninstalling Identity Console:



## Uninstallation of Identity Console Docker Container

To uninstall Identity Console Docker container, perform the following steps:

1. Stop the Identity Console container:

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

2. Run the following command to remove the Identity Console Docker Container:

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

3. Run the following command to remove the Docker image:

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

4. Remove the Docker volume:

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

**NOTE:** If you remove the volume, the data will also be removed from your server.

## Uninstallation of Standalone Identity Console (Non-Docker)

To uninstall standalone Identity Console, perform the following steps:

1. Navigate to the `/usr/bin` directory on the machine where Identity Console is installed.
2. Run the following command:

   ```
   ./identityconsoleUninstall
   ```

   Identity Console is successfully uninstalled.

**NOTE:** When the eDirectory or other NetIQ product is installed in the machine, the user must manually uninstall *nici* and *openssl*.

# Uninstallation of Identity Console Workstation on Windows

1. Close all the Identity Console applications.
2. Close the eDirAPI terminal.
3. Navigate to the folder `IdentityConsole_<version>_workstation_win_x86_64` where the build is extracted, navigate to `eDirAPI` folder, and take a backup of `cert` folder.
4. Delete the Workstation folder `IdentityConsole_<version>_workstation_win_x86_64`.

   The Identity Console Workstation on Windows is uninstalled.

# 5 Troubleshooting

This section includes some tips and best practices while using Identity Console. If you find something that works well for you, please share it at Cool Solutions (http://www.novell.com/coolsolutions).



For the Identity Console to perform at its best, use latest supported browser, and clear the browser cache and cookies at regular intervals.

## ERROR: Login Failure. Invalid Credentials

If you see a `"Login Error. Invalid Credentials"` (applies only for Container and Linux installed IDC) error message when you log into the Identity Console, you should check the Identity Console network call to find if `"Entered Hostname invalid"` error appears. If you receive an error message, it is because the eDirectory IP address associated with the Identity Console may not be present in the `edirapi.conf` file. In this situation, edit the `edirapi.conf` file and add the `edirectory ip` address to the file under the `edir-hosts` parameter.

A sample configuration file is shown below:

```
listen = ":9000"
pfxpassword = "novell"
bcert = "/etc/opt/novell/eDirAPI/cert/"
ospmode=false
edir-hosts = "<ip_address-1>:636,<ip_address-2>:636"
```

## INFO: Latest Identity Console version is already installed. Exiting

If Identity Console installation fails or gets interrupted, and the installer exits, on re-install of Identity Console, the error message is displayed as "Latest identity console version is already installed. Exiting".

In this situation uninstallation of the Identity Console is not required, user can run the following command to configure:

```
/usr/bin/identityconsoleConfigure
```

# ERROR: Installing Identity Console with Older NICI Instance

If an unsupported version of NICI is installed or if the NICI is not updated, Identity Console (version 1.8 and later) displays the following error message.

```
Server not configured with configure.bat file.
```

You have to uninstall/ update the NICI instance to support the current version of Identity Console.

# ERROR: Failed Dependencies

If you see the failed dependency error message for libldap_r-2.4.so.2 while installing the Identity console (version 1.7.2 and later) on later version of RHEL 9.X, install the `openldap-compact rpm` running the below command.

```
yum install openldap-compat.x86_64
```

# ERROR: Server Certificate Failed Message

While installing the Identity console (version 1.7.2 and later) on FIPS-enabled RHEL (version 9.X), you may see an auto-fetching server certificate failed error message.

This issue can be overcome by manually providing the certificates.