NetIQ® Identity Manager 4.8.6

Installation and Upgrade Guide

September 2022

Legal Notice

For information about NetIQ legal notices, disclaimers, warranties, export and other use restrictions, U.S. Government restricted rights, patent policy, and FIPS compliance, see https://www.netiq.com/company/legal.

Contents

About this Book and the Library

This guide provides instructions for installing or updating Identity Manager to the 4.8.6 version.

Intended Audience

This book is intended for identity architects and identity administrators responsible for installing or updating Identity Manager to this service pack.

Other Information in the Library

For more information about the library for Identity Manager, see the Identity Manager documentation website.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate—day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with—for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Website:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Website:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ website in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit https://www.netiq.com/communities/.

Updating Identity Manager on Standalone Servers

This section guides you through the process of installing or updating to the Identity Manager 4.8.6 version on standalone servers.

1 Planning Your Identity Manager Update

This service pack contains the following deliverables:

Filename	Description
`Identity_Manager_4.8.6_Linux.iso`	Contains files for Identity Manager Server (Identity Manager Engine, Remote Loader, Fanout Agent, and iManager), Identity Applications, and Identity Reporting for Linux platforms.
`Identity_Manager_4.8.6_Windows.iso`	Contains files for Identity Manager Server (Identity Manager Engine, Remote Loader, Fanout Agent, and iManager), Identity Applications, and Identity Reporting for Windows platforms.
`Identity_Manager_4.8.6_Containers.tar.gz`	Contains individual container images for Identity Manager Engine, Remote Loader, Fanout Agent, ActiveMQ, Form Renderer, OSP, Identity Applications, Identity Reporting, iManager, PostgreSQL, and SSPR.
`Identity_Manager_4.8.6_Designer.zip`	Contains files for Designer for all platforms.

Supported Update Paths

The update process requires you to update Identity Manager components in a specific order.

NOTE: If you are currently on Identity Manager 4.7.5 or a prior version, first upgrade your components to 4.8 and apply 4.8.6 update according to the following update paths.

Base Version	Updated Version
Identity Manager Engine 4.8.x where x is 0 to 5 with eDirectory 9.2.x, where x is 0 to 6	Identity Manager Engine 4.8.6 with eDirectory 9.2.7 **NOTE:** Identity Manager 4.8.6 is only compatible with eDirectory 9.2.7. eDirectory 9.2.7 no longer supports Identity Manager versions prior to 4.8.6 and vice versa. Hence, make sure that you update the eDirectory to version 9.2.7.
Identity Manager 4.8.x with Remote Loader 4.8.x, where x is 0 to 5	Identity Manager 4.8.x with Remote Loader 4.8.6, where x is 0 to 6 Identity Manager 4.8.6 with Remote Loader 4.8.x, where x is 0 to 6
Identity Manager Designer 4.8.x, where x is 0 to 5	Identity Manager Designer 4.8.6
Identity Applications 4.8.x, where x is 0 to 5	Identity Applications 4.8.6
Identity Reporting 6.6.x, where x is 0 to 8	Identity Reporting 6.7.2

Base Version	Updated Version
Identity Analyzer 4.8	Identity Analyzer 4.8.5
Fanout Agent 1.2.x, where x is 2 to 6	Fanout Agent 1.2.8
Sentinel Log Management for IGA 8.4	Sentinel Log Management for IGA 8.5.0.1

Update Order

You must update the components in the following order:

1. Identity Vault (eDirectory)
2. Identity Manager Engine
3. Remote Loader
4. Fanout Agent
5. iManager Web Administration
6. (Conditional) PostgreSQL

   NOTE: NetIQ recommends you to update PostgreSQL to the latest version when you are using PostgreSQL shipped with Identity Manager and when PostgreSQL (shipped with Idenity Manager) is installed on the same server as Identity Applications or Identity Reporting. For information on the supported versions of PostgreSQL, see the Identity Manager 4.8.x System Requirements Guide.

7. Identity Applications (for Advanced Edition)
8. Identity Reporting
9. Designer
10. Analyzer
11. Sentinel Log Management for IGA
12. Self-Service Password Reset (SSPR)

    NOTE: Standalone update of SSPR is required if SSPR is installed on a remote server.

Considerations for Updating SSPR on Linux and Windows

The following considerations apply to Self Service Password Reset (SSPR) before you update Identity Manager to 4.8.6 version on Linux and Windows platforms:

- If auditing is enabled on SSPR server with Syslog output format type as CEF, then you must uninstall the NetIQ Self Service Password Reset Collector from Sentinel Syslog server, else the Syslog server will not be able to parse the SSPR audit events.
- SSPR supports both CEF and JSON output format type for auditing events. SSPR 4.5.0.4 will continue to support NetIQ Self Service Password Reset Collector for JSON output format type. If there are more than one SSPR servers connected to a single Sentinel Syslog server, then you must select only one format type for auditing events across all servers.

After you update Identity Manager to 4.8.6 version, SSPR is upgraded to 4.6.0.0 version which requires Universal CEF Collector for collecting auditing events in CEF format type.

NOTE: If you are enabling the SSPR auditing in CEF output format type for the first time, ensure that the NetIQ Self Service Password Reset Collector is not configured on the Sentinel Syslog server.

Consideration for Updating Identity Applications on OES

When you are installing Identity Applications on Open Enterprise Server 2018 SP3 version, make sure that the JRE used by the Identity Manager Engine has the required certificate to connect to the User Application. The Identity Manager drivers use Identity Manager Engine's keystore to access the User Application. If the Roles and Resource Service Driver is unable to locate the required certificate at the target location, you may observe an error in the driver's trace. For more information on how to troubleshoot the error, see NetIQ Identity Manager - Administrator's Guide to the Identity Applications.

Consideration for Updating Identity Manager Engine on OES

When installing Identity Manager on an operating system, you typically configure the components immediately after installation. However, installing Identity Manager Engine on Open Enterprise Server 2023 requires a slightly different approach. After installation, make sure that you update the Identity Manager Engine to version 4.8.6. Do not configure the engine parameters until it has been updated to version 4.8.6. Failure to do so may result in an LDAP exception during configuration.

2 Updating the Identity Manager Components on Linux

The following considerations apply before you update Identity Manager components on Linux platforms:

- Ensure that you install the `zip` and `unzip` RPM packages.

  NOTE: NetIQ recommends that you obtain the dependent packages from your operating system subscription service to ensure continued support from your operating system vendor. If you do not have a subscription service, you can find the recent packages from a website such as http://rpmfind.net/linux.

- Perform the following steps to verify the RPM signature:
  1. Navigate to the following location for public key:

     ```
     <ISO-MOUNT-location>/common/license/
     ```

     For example:

     cd /mnt/common/license/
  2. Run the following command to import the key:

     ```
     rpm --import MicroFocusGPGPackageSign.pub
     ```
  3. (optional) Run the following command to verify the signature:

     ```
     rpm --checksig -v <RPM Name>
     ```

     For example:

     ```
     rpm --checksig -v novell-NDSbase-9.2.7.0000.x86_64.rpm
     ```
- (Conditional) If you are updating the Identity Manager from 4.8 to 4.8.6 directly, then you must apply the Identity Applications 4.8.0.1 patch before 4.8.6 version in the following scenarios:
  - eDirectory 9.2 and Identity Applications 4.8 are installed on the same server.
  - iManager 3.2 and Identity Applications 4.8 are installed on the same server.
  - Identity Applications 4.8 and PostgreSQL are installed on the same server.

  The Identity Applications 4.8.0.1 patch resolves the dependencies between the NGINX module and the OpenSSL libraries. For instructions on applying the patch, see the NetIQ Identity Applications 4.8.0 Hotfix 1 Release Notes.

  If you do not apply the Identity Applications 4.8.0.1 patch, the Identity Vault update fails and the installer reports the following error message:

  ```
  Problem: patterns-edirectory-9.2.2-6.x86_64 requires netiq-openssl =
  1.0.2u, but this requirement cannot be provided not installable
  providers: netiq-openssl-1.0.2u-32.x86_64[edirectory-9.2.2]
  Solution 1: deinstallation of netiq-nginx-1.14.2-1.x86_64
  ```

```
Solution 2: do not install patterns-edirectory-9.2.2-6.x86_64
Solution 3: break patterns-edirectory-9.2.2-6.x86_64 by ignoring some
of its dependencies
```

- (Conditional) Before installing or upgrading Identity Manager to 4.8.6 version on RHEL 8.5 or SLES 15 SPx operating system, make sure that you download and execute the `idm_updated_os_linux.sh` script available at the following location: TID KM000007635 (https://portal.microfocus.com/s/article/KM000007635?language=en_US). If you do not replace the scripts, the PostgreSQL database configuration may fail with the following error:

  ```
  symbol lookup error: /opt/netiq/idm/postgres/bin/../lib/
  libgssapi_krb5.so.2: undefined symbol: krb5_ser_context_init, version
  krb5_3_MIT
  ```

  For more information, see TID KM000007635 (https://portal.microfocus.com/s/article/KM000007635?language=en_US).

Updating the Identity Vault

1. Download and mount the `Identity_Manager_4.8.6_Linux.iso` file from the download site.
2. Navigate to the `<ISO mounted location>/IDVault/setup` directory.
3. Run the following command:

   ```
   ./nds-install
   ```

4. Accept the license agreement.
5. Specify the Administrator DN and the password for the Identity Vault instance.

NOTE: This steps are not applicable for OES platform.

Updating the Identity Manager Components

The update of the Identity Manager components on Linux is supported through a single script. You must run the `install.sh` script to update these components. The components include Identity Manager Engine, Remote Loader, Fanout Agent, iManager Web Administration, Identity Applications, and Identity Reporting.

NOTE:

- Before updating the Remote Loader, ensure that the following components are stopped:
  - Remote Loader instances
  - Driver instances running with the Remote Loader
  - Identity Vault
- The installation program creates a new master key during Identity Applications configuration. This master key is stored under the property `com.novell.idm.masterkey` in the `ism-configuration.properties` file located at `/opt/netiq/idm/apps/tomcat/conf/`

directory. The User Application clustering requires every instance of the User Application to use the same master key. Before starting Tomcat, make sure to copy the master key value from the master node and replace the corresponding master key value in all nodes in cluster.

NetIQ provides two options for updating the components to the current version: interactive and silent.

Interactive Update

1. Download and mount the `Identity_Manager_4.8.6_Linux.iso` file from the download site.
2. Navigate to the `<ISO mounted location>` and run the following command:

   ```
   ./install.sh
   ```

3. Specify the component that you want to update.

   NOTE: You can update only one component at a time.

4. To start the Identity Manager components, run the following commands:
   - **Remote Loader:** `rdxml -config <filename>`
   - **Fanout Agent:** Perform the following steps:
     1. Navigate to `/opt/novell/dirxml/fanoutagent/bin` directory.
     2. Run the following command:

        ```
        ./startAgent -config <FanoutAgent Installation Location>/config/
        fanoutagentconfig.properties
        ```

   - **Identity Applications:** `systemctl start netiq-tomcat.service`
   - **Identity Reporting:** `systemctl start netiq-tomcat.service`
5. (Conditional) If you have applied any customizations on Identity Applications and Identity Reporting components, restore the customizations and restart the Tomcat service.
6. (Conditional) Clear your browser cache before accessing the updated Identity Applications Dashboard.

Silent Update

Locate the `silent.properties` file from the extracted directory and modify the file to update the required components.

- To update the Identity Vault, set `IDVAULT_SKIP_UPDATE=false`
- To update Identity Manager Engine, set `INSTALL_ENGINE=true`
- To update Remote Loader, set `INSTALL_RL=true`
- To update Fanout Agent, set `INSTALL_FOA=true`
- To update iManager, set `INSTALL_IMAN=true`
- To update Identity Reporting, set `INSTALL_REPORTING=true`
- To update Identity Applications, set `INSTALL_UA=true`

NOTE:

- You must set the value to `true` for only one component at a time.
- While updating any component other than Identity Vault, you must always set the value of `IDVAULT_SKIP_UPDATE` to `true` to skip the Identity Vault update.
- When you update iManager, the iManager plug-ins, if any, are also upgraded.

Perform the following actions to update the components silently:

1. Download and mount the `Identity_Manager_4.8.6_Linux.iso` file from the download site.
2. Navigate to the `<ISO mounted location>` directory.
3. Run the following command:

   ```
   ./install.sh -s -f silent.properties
   ```

4. To start the Identity Manager components, run the following commands:
   - **Remote Loader:** `rdxml -config <filename>`
   - **Fanout Agent:** Perform the following steps:
     1. Navigate to `/opt/novell/dirxml/fanoutagent/bin` directory.
     2. Run the following command:

        ```
        ./startAgent -config <FanoutAgent Installation Location>/config/
        fanoutagentconfig.properties
        ```

   - **Identity Applications:** `systemctl start netiq-tomcat.service`
   - **Identity Reporting:** `systemctl start netiq-tomcat.service`
5. (Conditional) If you have applied any customizations on Identity Applications and Identity Reporting components, restore the customizations and restart the Tomcat service.
6. (Conditional) Clear your browser cache before accessing the updated Identity Applications Dashboard.

Updating PostgreSQL

The following considerations apply before updating PostgreSQL:

- NetIQ recommends you to update PostgreSQL to the latest version when you are using PostgreSQL shipped with Identity Manager and when PostgreSQL (shipped with Idenity Manager) is installed on the same server as Identity Applications or Identity Reporting. For information on the supported versions of PostgreSQL, see the Identity Manager 4.8.x System Requirements Guide.
- If Identity Vault and PostgreSQL are installed on the same server, update Identity Vault before you update PostgreSQL.

NOTE: In addition to the default capabilities offered by PostgreSQL 12.12, this service pack allows you to configure the PostgreSQL database with SSL (OpenSSL 1.0.2zf built with FIPS). This service pack also bundles the PostgreSQL Contrib packages.

1. Download and mount the `Identity_Manager_4.8.6_Linux.iso` file from the download site.
2. Navigate to the `<ISO mounted location>/common/scripts` directory and run the `pg-upgrade.sh` script.

   NOTE: To specify a different directory than the existing directory, run the `SPECIFY_NEW_PG_DATA_DIR=true ./pg-upgrade.sh` command.

   The upgrade script performs the following actions:

   - Takes a backup of the existing postgres to a different folder. For example, from `/opt/netiq/idm/postgres` to `/opt/netiq/idm/postgres-<timestamp>-backup.`
   - Updates the existing Postgres directory. For example, `/opt/netiq/idm/postgres`.
3. Specify the following details to complete the installation:

   Existing Postgres install location: Specify the location where PostgreSQL is installed. For example, `/opt/netiq/idm/postgres`.

   Existing Postgres Data Directory: Specify the location of the existing PostgreSQL data directory. For example, `/opt/netiq/idm/postgres/data`.

   Existing Postgres Database Password: Specify the PostgreSQL password.

   Enter New Postgres Data Directory: Specify the location of the new PostgresSQL data directory. This prompt is displayed if you selected to specify a different directory other than the existing directory.

Performing a Standalone Update of SSPR

NOTE:

- If SSPR auditing output format type is CEF, make sure to uninstall the NetIQ Self Service Password Reset Collector on Sentinel Syslog server before updating SSPR. For more information, see "Considerations for Updating SSPR on Linux and Windows" on page 16.
- Use this method if SSPR is:
  - Installed on a different server than the Identity Applications server.
  - Installed in a Standard Edition.

Perform the following steps to update SSPR:

1. Download and mount the `Identity_Manager_4.8.6_Linux.iso` file.
2. Navigate to the `<ISO mounted location>/sspr` directory.
3. Run the following command:

   ```
   ./install.sh
   ```
4. Specify inputs in the prompt.

Performing a Non-Root Update

You can install Identity Manager Engine as a non-root user to enhance the security of your Linux server. You cannot install Identity Manager Engine as a non-root user if you installed the Identity Vault as root. You need to perform the following steps to install the Identity Manager Engine as a non-root user:

- Update NICI. For more information, see Updating NICI.
- Update eDirectory as a non-root user. For more information, see Updating eDirectory as a Non-root User.
- Update Identity Manager Engine as a non-root user. For more information, see Updating Identity Manager Engine as a Non-root User.

Updating NICI

Ensure that you are logged-in as a root user before updating NICI.

1 Run the following command to stop the eDirectory instance:

```
ndsmanage stopall
```

2 Navigate to the `/<location where you have mounted the ISO>/IDVault/setup` directory.

3 Run the following command:

```
rpm -Uvh nici64-3.2.0-00.x86_64.rpm
```

Updating eDirectory as a Non-root User

A non-root user can upgrade eDirectory using the new version of the tarball. Perform the following steps to upgrade eDirectory as a non-root user:

1 Log in as a non-root user.

2 Navigate to the `/<location where you mounted the ISO>/IDVault/` directory.

3 Copy the `eDir_NonRoot.tar.gz` file to a non-root home directory.

4 Run the following command to extract the `.tar.gz` file.

```
tar -zxvf eDir_NonRoot.tar.gz
```

5 (Conditional) Ensure the below paths are set in the `<non-root home directory>/.bash_profile` so that below path's are not required to be set for each time user logs in a session:

```
export LD_LIBRARY_PATH=<non-root home directory>/eDirectory/opt/novell/
eDirectory/lib64:<non-root home directory>/eDirectory/opt/novell/
eDirectory/lib64/nds-modules:<non-root home directory>/eDirectory/opt/
novell/lib64:$LD_LIBRARY_PATH
```

```
export PATH=<non-root home directory>/eDirectory/opt/novell/eDirectory/
bin:<non-root home directory>/eDirectory/opt/novell/eDirectory/sbin:/
opt/novell/eDirectory/bin:$PATH
```

```
export MANPATH=<non-root home directory>/eDirectory/opt/novell/
man:<non-root home directory>/eDirectory/opt/novell/eDirectory/
man:$MANPATH
```

```
export TEXTDOMAINDIR=<non-root home directory>/eDirectory/opt/novell/
eDirectory/share/locale:$TEXTDOMAINDIR. <non-root home directory>/
eDirectory/opt/novell/eDirectory/bin/ndspath
```

6 Restart eDirectory.

```
ndsmanage stopall
```

```
ndsmanage startall
```

Updating Identity Manager Engine as a Non-root User

Perform this action only if you have installed Identity Manager Engine as a non-root user. You can perform the update through an interactive or silent mode.

Interactive Update

Perform the follow steps to perform a non-root interactive update of Identity Manager Engine:

1. Download and mount the `Identity_Manager_4.8.6_Linux.iso` for non-root user to access.
2. Log in as a non-root user.
3. Run the following command from the location where you have mounted the `Identity_Manager_4.8.6_Linux.iso`:

   ```
   ./install.sh
   ```

4. Select **Identity Manager Engine** and press **Enter**.
5. Specify the non-root install location for Identity Vault.

   For example, `/home/user/eDirectory/`.
6. Specify **Y** to complete the update.

Silent Update

Perform the follow steps to perform a non-root silent update of Identity Manager Engine:

1. Copy the `silent.properties` file from the `/<ISO mounted location>/` to a folder accessible by the non-root user.
2. In the `silent.properties` file, edit the following:
   - Set the value for the below properties to **true**:
     - `INSTALL_ENGINE`
     - `IDVAULT_SKIP_UPDATE`
   - Specify the value of the **NONROOT_IDVAULT_LOCATION** parameter as `/home/`*`<non-root username>`*`/eDirectory`, where *`<non-root username>`* indicates the name of the non-root user.

3 Navigate to the location where you mounted the ISO.

4 Run the following command:

```
./install.sh -s -f /<location where you copied the silent.properties
file to in step 1>/silent.properties
```

Upgrading Standalone Identity Console

(Conditional) This section applies if you already use Identity Console with Identity Manager 4.8.x.

To upgrade standalone Identity Console on Identity Manager 4.8.6, see Upgrading Standalone Identity Console (Non Docker) (https://www.netiq.com/documentation/identity-console/identity_console-install/data/t4h7itft4ue7.html).

If you are installing Identity Console for the first time, see Identity Console Installation Guide (https://www.netiq.com/documentation/identity-console/identity_console-install/data/bookinfo.html)

Post-Update Tasks

Perform the following actions after updating Identity Manager to the 4.8.6 version:

Extending the Identity Vault Schema

(Conditional) This section does not apply if you have already upgraded to 4.8.1 and above and extended the Identity Vault Schema.

However, this section applies:

- if you have installed Identity Manager as a root or a non-root user, and
- if you want to extend the Identity Vault schema for the Resource Weightage feature

To extend the Identity Vault schema, perform the following steps:

1 Log in to the server where you want to extend the Identity Vault schema.

2 Navigate to `/opt/novell/eDirectory/bin` directory.

3 Run the following command to extend the schema:

```
./idm-install-schema
```

4 Update the Role and Resource Service Driver to 4.8.6. For more information, refer to the section “Update Driver Packages” on page 27.

5 Restart the Identity Vault.

Post-Update Tasks for Identity Manager Drivers

(Conditional) This section applies if you want to update the following versions for these drivers:

- DCS 4.2.1.0300
- RRSD 4.8.6

For more information, see the NetIQ Identity Manager Data Collection Service Driver 4.2 Service Pack 1 Patch 3 Readme (https://www.netiq.com/documentation/identity-manager-48-drivers/DataCollectionServices4210300/data/DataCollectionServices4210300.html) and NetIQ Identity Manager Role and Resource Service Driver 4.8.6 Release Notes (https://www.netiq.com/documentation/identity-manager-48-drivers/RRSDriver486readme/data/RRSDriver486readme.html).

Update Driver Packages

NOTE: Before updating the driver packages to 4.8.6, ensure that you have updated to the latest version of Identity Applications.

Once the Identity Applications is updated to the latest version, you can update the User Application driver package to 4.8.6.20220912161613.

Perform the following steps to update the driver package:

1. Open the project containing the User Application driver.
2. Right-click the User Application driver, then click **Driver** > **Properties**.
3. Click **Packages**.

   A check mark indicates a newer version of a package in the Upgrades column.
4. Click **Select** operation against the `NOVLUABASE_4.8.6.20220912161613.jar` package.
5. From the drop-down list, click **Upgrade**.
6. Select the version that you want to upgrade to, then click **OK**.

   NOTE: Designer lists all versions available for upgrade.
7. Click Apply.
8. (Conditional) Fill in the fields with appropriate information to upgrade the package, then click **Next**.

   Depending on which package you selected for upgrade, you must fill in the required information appropriately to upgrade the package.
9. Read the summary of the packages that will be installed, then click **Finish**.
10. Review the upgraded package, then click **OK** to close the Package Management page.
11. Deploy the changes and restart the driver.
12. Restart the eDirectory service.

Removing Unused Log4j JAR

The `log4j-1.2.13.jar` is installed in the `/opt/netiq/idm/lightWeightDesigner/plugins/com.novell.soa.eai.integrationActivity_4.0.0.201910221801/lib` directory as part of Identity Manager 4.8 installation. Identity Manager no longer uses this jar. Locate and delete the `log4j-1.2.13.jar` from the directory.

3 Updating the Identity Manager Components on Windows

This service pack includes a `Identity_Manager_4.8.6_Windows.iso` file for updating the Identity Manager components on Windows platforms.

NOTE: If Identity Manager Engine is installed on the same server as Identity Applications or Identity Reporting, then the Identity Applications or the Identity Reporting update process will restart the Identity Vault (eDirectory) service.

Updating the Identity Vault

1. Download and mount the `Identity_Manager_4.8.6_Windows.iso` file.
2. Navigate to the `<ISO mounted location>\IdentityManagerServer\eDirectory` directory and run the `eDirectory_927_Windows_x86_64.exe` file.

   NOTE: The Identity Vault update process restarts the Identity Vault (eDirectory) server.

   Tree Name

   Verify the tree name for Identity Vault.

   Server FDN

   Verify the server FDN.

   Tree Admin

   Specify an administrator name for Identity Vault in NCP or dot format.

   Admin Password

   Specify the administrator password.
3. In the **Install Location** field, verify the location where Identity Vault is installed.
4. In the **DIB Location** field, verify the location where the DIB files are located.
5. Select the **NICI** check box.
6. Click **Upgrade**.

Updating the Identity Manager Server Components

This section describes how to update Identity Manager Server Components:

1. Download and mount the `Identity_Manager_4.8.6_Windows.iso` file from the download site.
2. Stop the Identity Vault and Remote Loader instances.

(Conditional) This step is applicable only if you are upgrading Remote Loader.

2a Stop all Remote Loader instances.

2b Close Remote Loader console.

2c Stop all drivers.

2d Stop the Identity Vault.

3 (Conditional) If you are performing an interactive update, perform the following steps:

3a Navigate to the `<ISO mounted location>\IdentityManagerServer` directory.

3b Run `install.exe` file.

3c Select the component that you want to update from the list and click **Next**.

To update the Identity Manager Engine, select **Identity Manager Engine**.

To update the 32-bit Remote Loader, select **32-Bit Remote Loader Service**.

To update the 64-bit Remote Loader, select **64-Bit Remote Loader Service**.

To update the .NET Remote Loader, select **.NET Remote Loader Service**.

To update the Fanout Agent, select **Fanout Agent**.

To update the iManager, select **iManager**.

3d In the **Pre-Installation Summary** page click **Install**.

4 (Conditional) If you are performing a silent update, perform the following steps:

4a Navigate to the `<ISO mounted location>\IdentityManagerServer\response-file` directory.

4b Copy the `install.properties` file to a different location.

4c Edit the `install.properties` file and set the value of the components as appropriate.

To update Identity Manager Engine, set the value of **NETIQ_UPGRADE_ENGINE** to **true**.

To update the Remote Loader (root and non-root), set the value of **NETIQ_UPGRADE_REMOTE_LOADER** to **true**.

To update the 32-bit Remote Loader, set the value of **NETIQ_UPGRADE_REMOTE_LOADER_32** to **true**.

To update the 64-bit Remote Loader, set the value of **NETIQ_UPGRADE_REMOTE_LOADER_64** to **true**.

To update the Fanout Agent, set the value of **NETIQ_UPGRADE_FANOUT_AGENT** to **true**.

To update the iManager, set the value of **NETIQ_UPGRADE_iManager** to **true**.

4d In the command prompt, run the following command:

```
install.exe -i silent -f <absolute path of install.properties>
```

5 Start the Remote Loader and Fanout Agent instances.

Updating the PostgreSQL Database

The following considerations apply before updating PostgreSQL:

- NetIQ recommends you to update PostgreSQL to the latest version when you are using PostgreSQL shipped with Identity Manager and when PostgreSQL (shipped with Idenity Manager) is installed on the same server as Identity Applications or Identity Reporting. For information on the supported versions of PostgreSQL, see the Identity Manager 4.8.x System Requirements Guide.
- If Identity Vault and PostgreSQL are installed on the same server, update Identity Vault before you update PostgreSQL.

NOTE: In addition to the default capabilities offered by PostgreSQL 12.12, this service pack allows you to configure the PostgreSQL database with SSL (OpenSSL 1.0.2zf built with FIPS). This service pack also bundles the PostgreSQL Contrib packages.

1. Stop and disable the PostgreSQL service running on your server.
2. Navigate to the directory where PostgreSQL is installed. For example, `C:\Netiq\IDM`.
3. Rename the `postgres` directory.

   For example, rename `postgres` to `postgres_old`.
4. Remove the old PostgreSQL service by running the following command:

   ```
   sc delete <"postgres service name">
   ```

   For example, `sc delete "NetIQ PostgreSQL"`
5. Download and mount the `Identity_Manager_4.8.6_Windows.iso` file.
6. Navigate to the `<ISO mounted location>\common\postgres` directory and run the `NetIQ_PostgreSQL.exe` file.

   NOTE: Ensure that you have the Administrator privileges for the old and new PostgreSQL installation directories.
7. Specify the path where you want to install PostgreSQL. For example, `C:\Netiq\IDM`.
8. Click **Next**.
9. Specify the password for the `postgres` user.
10. Specify the PostgreSQL port. The default port is 5432.
11. Do not select the **Create database login account** and **Create empty database** check boxes.
12. Click **Next**.
13. Review the details on the **Pre-Installation summary** page and click **Next**.
14. Stop the newly installed PostgreSQL service.

    Go to **Services**, search for `NetIQ PostgreSQL` service, and stop the service.

    NOTE: Appropriate users can perform stop operations after providing valid authentication.

15 Change the permissions for the newly installed PostgreSQL directory by performing the following actions:

15a (Optional) If postgres user is not created, then perform the following steps to create a `postgres` user:

15a1 Go to **Control Panel** > **User Accounts** > **User Accounts** > **Manage Accounts**.

15a2 Click **Add a user account**.

15a3 In the **Add a User** page, specify postgres as the user name and provide a password for the user.

15b Assign permissions for the postgres user to the existing and newly installed PostgreSQL directories. Right-click the corresponding directories and go to **Properties** > **Security** > **Edit**.

15c Select **Full Control for the user** to provide complete permissions.

15d Click **Apply**.

16 Access the PostgreSQL directory as `postgres` user.

16a Log in to the server as postgres user.

Before logging in, make sure that postgres can connect to the Windows server by verifying if a remote connection is allowed for this user.

16b Delete the data directory from the new PostgreSQL installed location.

For example, `C:\NetIQ\IDM\postgres\data`.

16c Open a command prompt and set `PGPASSWORD` by using the following command:

```
set PGPASSWORD=<your pg password>
```

16d Change to the newly installed PostgreSQL directory.

For example, `C:\NetIQ\IDM\postgres\bin`.

16e Based on the encoding type that is set for the database, execute the following `initdb` commands as a `postgres` user from the `bin` directory. By default, the encoding type is set to WIN1252.

If the encoding type is set to WIN1252, run the following command:

```
initdb.exe -D <new_data_directory> -E <Encoding> WIN1252 -U postgres
```

For example, `initdb.exe -D C:\NetIQ\IDM\postgres\data -E WIN1252 -U postgres`

If the encoding type is set to UTF8, run the following command:

```
initdb.exe -D <new_data_directory> -E <Encoding> UTF8 -U postgres
```

For example, `initdb.exe -D C:\NetIQ\IDM\postgres\data -E UTF8 -U postgres`

16f Navigate to the `C:\NetIQ\idm\postgres\data\` directory, edit the `pg_hba.conf` file, and set the **Method** type from `md5` to `trust`.

IMPORTANT: You must also set the **Method** type from `md5` to `trust` in the `pg_hba.conf` file located in the `C:\NetIQ\idm\postgres_old\data\` directory.

17 Navigate to the `C:\NetIQ\idm\postgres\bin` directory and run the following command:

```
pg_upgrade.exe --old-datadir "C:\NetIQ\IDM\postgres_old\data" --new-
datadir "C:\NetIQ\IDM\postgres\data" --old-bindir
"C:\NetIQ\IDM\postgres_old\bin" --new-bindir
"C:\NetIQ\IDM\postgres\bin"
```

18 Once PostgreSQL is upgraded successfully, perform the following steps:

18a Navigate to the `C:\NetIQ\IDM\postgres_old\data` directory.

18b Copy the `pg_hba.conf` and `postgresql.conf` files.

18c Navigate to `C:\NetIQ\IDM\postgres\data` directory.

18d Replace the files you copied in Step 18b.

19 Start the PostgreSQL service.

Go to **Services**, search for `NetIQ PostgreSQL` service, and start the service.

NOTE: Appropriate users can perform start operations after providing valid authentication.

20 (Optional) To ensure that the old cluster's data files are deleted and the service does not start automatically, perform the following steps:

20a Log in as `postgres` user.

20b Navigate to the `C:\NetIQ\IDM\postgres\bin` directory.

20c Run the `analyze_new_cluster.bat` and `delete_old_cluster.bat` files.

Installing the Identity Console on Windows as Workstation

From Identity Manager 4.8.6 release onwards user can Install the Identity Console on Windows as Workstation, for more information see Deploying Identity Console Workstation on Windows .

Updating the Identity Applications

(Conditional) Delete or take a back-up of the existing logs from the `<install_directory>\IDM\apps\tomcat\logs` directory.

1 On your Identity Applications server, press Windows + R on your keyboard, type `services.msc` and select **OK** to open the Windows Services interface. From the Windows services, stop the **IDM Apps Tomcat Service** and the **NDS Service**.

2 Download and mount the `Identity_Manager_4.8.6_Windows.iso` file from the download site.

3 Navigate to the `<ISO mounted location>\IdentityApplications` directory.

4 Perform one of the following actions:

GUI: `install.exe`

Silent: In the command prompt, go to the `<ISO mounted location>\IdentityApplications` location and run `install.exe -i silent`

The Identity Applications update program will update User Application, OSP, SSPR, Tomcat, and JRE.

5 For GUI, on the **Introduction** page, click **Next**.

6 Review the **Deployed Applications** page, then click **Next**.

This page lists the currently installed components with their versions.

7 On the **Available Patches** page, click **Next**.

This page lists the available updates for the installed components.

8 Review the required disk space and available disk space for installation in the **Pre-Install Summary** page, then click **Install**.

The installation process might take some time to complete.

Before applying the service pack, the installation process automatically stops the Tomcat service.

The process also creates a back-up of the current configuration for the installed components.

In case, the installation reports any warnings or errors, see the logs from the Service Pack Installation/Logs directory.

For example, `C:\NetIQ\IDM\apps\Identity_Apps_4.8.6.0_Install\Logs`. You must fix the issues and manually restart the Tomcat service.

9 (Optional) The installation program creates a new master key during Identity Applications configuration. This master key is stored under the property `com.novell.idm.masterkey` in the `ism-configuration.properties` file located at `C:\NetIQ\IDM\apps\tomcat\conf` folder. The User Application clustering requires every instance of the User Application to use the same master key. Before starting Tomcat, make sure to copy the master key value from the master node and replace the corresponding master key value in all nodes in cluster.

10 Start the Tomcat and NDS services.

11 (Optional) To verify that the service pack has been successfully applied, launch the upgraded components and check the component versions.

12 Clear your browser cache before accessing Identity Applications.

NOTE: To modify any settings in the configuration update utility, launch `configupdate.bat` from the `<install_directory>\IDM\apps\configupdate` directory.

Updating Identity Reporting

(Conditional) Delete or take a back-up of the existing logs from the `<install_directory>\IDM\apps\tomcat\logs` directory.

1 On your Identity Reporting server, press Windows + R on your keyboard, type `services.msc` and select **OK** to open the Windows Services interface. From the Windows services, stop the **IDM Apps Tomcat Service** and the **NDS Service**.

2 Download and mount the `Identity_Manager_4.8.6_Windows.iso` file.

3 Navigate to the `<ISO mounted location>\IdentityReporting` directory.

4 Perform following steps:

Silent: In the command prompt, go to the `<ISO mounted location>\IdentityReporting` location and run `install.exe -i silent`

GUI: In the `IdentityReporting` directory, double-click on `install.exe`

5 For GUI, on the **Introduction** page, click Next.

6 Review the **Deployed Applications** page, then click **Next**.

This page lists the currently installed components with their versions.

7 On the **Available Updates** page, click **Next**.

This page lists the available updates for the installed components.

8 On the **Pre-Installation Summary** page, click **Install**.

9 Start the Tomcat and NDS services.

10 Clear your browser cache before accessing Identity Reporting.

NOTE: To modify any settings in the configuration update utility, launch `configupdate.bat` from the `<install_directory>\IDM\apps\configupdate` directory.

Post-Update Tasks

Perform the following actions after applying this service pack.

Extending the Identity Vault Schema

(Conditional) This section does not apply if you have already upgraded to 4.8.1 and above and extended the Identity Vault Schema.

This section applies if you want to extend the Identity Vault schema for the Resource Weightage feature.

To extend the Identity Vault schema, perform the following steps:

1 Log in to the server where you want to extend the Identity Vault schema.

2 Create a new file in your preferred directory.

For example, create `nrf-extensions.sch` file in the `C:\Temp` directory.

3 Open the `nrf-extensions.sch` file and add the following content:

```
--
-- The nrfResourceWeightage attribute contained by nrfResource object
class specifies the weightage of
-- resource object which is used for assignment/revocation based on
priority
--
NDSSchemaExtensions DEFINITIONS ::=
BEGIN
"nrfResourceWeightage" ATTRIBUTE ::=
{
                Operation                           ADD,
                Flags
{DS_SYNC_IMMEDIATE, DS_SINGLE_VALUED_ATTR},
             SyntaxID                        SYN_INTEGER,
                ASN1ObjID                           {2 16 840 1 113719 1
33 4 174}
}


"nrfResource" OBJECT-CLASS ::=
{
    Operation   MODIFY,
    MayContain  {"nrfResourceWeightage"}
}
END
```

4 Navigate to the `C:\NetIQ\eDirectory\` directory.

5 Run the following command to extend the schema:

```
ice -l <schema_update_log> -C -a -S SCH -f <file that you created in
step 2> -D LDAP -s <eDirectory DNS name/IP> -p <LDAP port> -d
<eDirectory_admin_dn> -w <eDirectory_admin_password>
```

where,

`-C -a` updates the destination schema.

`-f` indicates the schema file (sch).

`-p` indicates the port number of the LDAP server. The default port is 389. For secure communication, use port 636. Secure communication needs an SSL Certificate.

`-L` indicates a file in DER format containing a server key used for SSL authentication.

`-s` indicates the DNS name or IP address of the LDAP server.

For example,

```
ice -l schemaupdate.log -C -a -S SCH -f C:\Temp\nrf-extensions.sch -D
LDAP -s idmorg.com -p 636 -d cn=admin,ou=idm,o=microfocus -w password -
L cert.der
```

6 Update the Role and Resource Service Driver to 4.8.6. For more information, refer to the section “Update Driver Packages” on page 37.

7 Restart the Identity Vault.

Post-Update Tasks for Identity Manager Drivers

(Conditional) This section applies if you want to update to the following versions for these drivers:

- DCS 4.2.1.0300
- RRSD 4.8.6

For more information, see the NetIQ Identity Manager Data Collection Service Driver 4.2 Service Pack 1 Patch 3 Readme (https://www.netiq.com/documentation/identity-manager-48-drivers/DataCollectionServices4210300/data/DataCollectionServices4210300.html) and NetIQ Identity Manager Role and Resource Service Driver 4.8.6 Release Notes (https://www.netiq.com/documentation/identity-manager-48-drivers/RRSDriver486readme/data/RRSDriver486readme.html).

Update Driver Packages

NOTE: Before updating the driver packages to 4.8.6, ensure that you have the Identity Applications latest version.

Once the Identity Applications is updated to the latest version, you can update the User Application driver package to 4.8.6.20220912161613.

Perform the following steps to update the driver package:

1. Open the project containing the User Application driver.
2. Right-click the User Application driver, then click **Driver** > **Properties**.
3. Click **Packages**.

   A check mark indicates a newer version of a package in the Upgrades column.
4. Click **Select** operation against the `NOVLUABASE_4.8.6.20220912161613.jar` package.
5. From the drop-down list, click **Upgrade**.
6. Select the version that you want to upgrade to, then click **OK**.

   NOTE: Designer lists all versions available for upgrade.
7. Click Apply.
8. (Conditional) Fill in the fields with appropriate information to upgrade the package, then click **Next**.

   Depending on which package you selected for upgrade, you must fill in the required information appropriately to upgrade the package.
9. Read the summary of the packages that will be installed, then click **Finish**.
10. Review the upgraded package, then click **OK** to close the Package Management page.
11. Deploy the changes and restart the driver.
12. Restart the eDirectory service.

Removing Unused Log4j JAR

The `log4j-1.2.13.jar` is installed in the `<installed_path>\NetIQ\IDM\lightWeightDesigner\plugins\com.novell.soa.eai.integrationActivity_4.0.0.201910221801\lib` folder as part of Identity Manager 4.8 installation. Identity Manager no longer uses this jar. Locate and delete the `log4j-1.2.13.jar` from the folder.

4 Updating Designer

You must be on Designer 4.8 at a minimum to apply this update. The update process includes the following tasks:

Performing a Designer Update

You can apply the update in one of the following ways:

Online Update (using the Auto Update feature)

You can apply this update using the built-in auto-update feature of Designer. The auto-update feature notifies you of new features available at the Designer Download Site. This feature allows you to download Designer package and software updates when the computer that has Designer installed is connected to the Internet.

1. Launch Designer.
2. From Designer's main menu, click **Help > Check for Designer Updates**.
3. Click **Yes** to accept the Designer updates.
4. Restart Designer for the changes to take effect.

Offline Update (Using the download page to apply the update)

This service pack includes a `Identity_Manager_4.8.6_Designer.zip` file for updating Designer. You also can perform an offline update of Designer when the computer that has Designer installed is not connected to the Internet. To perform an offline update, first download this service pack on a local or remote computer and then point Designer to the directory containing the downloaded files.

To update Designer in an offline mode, create an offline copy of the Designer update files and then configure Designer to read the patch updates from the files copied to the local directory.

To create an offline copy of the Designer update files:

1. Go to Software Licence and Downloads Page.
2. Click **Downloads**.
3. In the **Product** dropdown list, select `Identity Manager (IDM)`.
4. In the **Product** Name dropdown list, select `Identity Manager Advanced Edition per User Sub SW E-LTU`.
5. In the **Version** dropdown list, select `4.8`.
6. Search for `Identity Manager Designer 4.8.6.0` and click **Download**.

7 Log in to the computer that has Designer installed and create a local directory.

8 Unzip the downloaded files into the local directory.

To configure Designer to read the patch updates from the local directory:

1 Launch Designer.

2 From Designer's main menu, click **Windows > Preferences**.

3 Click **NetIQ > Identity Manager** and select **Updates**.

4 For URL, specify `file:///media/`*`<path_to_update_file>`*`/updatesite1_0_0/`

For a Linux mounted ISO, use the following URL format:

`file:///media/designer486offline/updatesite1_0_0/`

5 Click **Apply**, then click **OK**.

6 From Designer's main menu, click **Help > Check for Designer Updates**.

7 Select the required updates and click **Yes** to accept and update the Designer.

8 Restart Designer for the changes to take effect.

Updating Azul Zulu OpenJRE 1.8.0_342

This service pack updates Designer to support Azul Zulu OpenJRE 1.8.0_342 (64-bit).

1 On the server where you installed Designer, download and install the Azul Zulu OpenJRE 1.8.0_342 (zulu8.64.0.15-ca-jre) files in a local directory.

2 Open the `Designer.ini` file located in the Designer installation directory.

3 Update the JRE path in the `Designer.ini` file.

5 Updating Analyzer

NOTE: This service pack does not support a new version of Analyzer. If you are using Analyzer version 4.8.5, skip this procedure.

You must be on Analyzer 4.8 at a minimum to apply this update. The update process includes the following tasks:

Performing an Analyzer Update

The update process includes the following tasks:

Updating Identity Analyzer on Linux

This section describes how to update Identity Analyzer on Linux:

1. Download the `Identity_Manager_4.8.5_Analyzer_Linux.tar.gz` file from the download site.
2. Untar the downloaded file as follows:

   ```
   tar -zxvf Identity_Manager_4.8.5_Analyzer_Linux.tar.gz
   ```
3. Navigate to the `analyzer_install` folder and run the following command:

   ```
   ./install
   ```
4. Select the language that you want to update from the list and click **OK**.
5. Click **Next**.
6. Select the License Agreement option and click **Next**.
7. Click **OK** in **Analyzer for Identity Manager is Running** window.
8. Check the install location path in the **Install Folder** page and click **Next**. In case, you want to modify the location, click the **Choose** button and select the desired path.
9. Click **Yes** in the **Analyzer Found** window.
10. Click **Next** in the **Create Shortcuts** page.
11. Click **Install** in the **Pre-Installation Summary** page.
12. Click **Done** in the **Install Complete** page.

Updating Identity Analyzer on Windows

This section describes how to update Identity Analyzer on windows:

1. Download and extract the `Identity_Manager_4.8.5_Analyzer_Windows.zip` file from the download site.
2. Navigate to the `<extracted zip file location>\analyzer_install` directory.

3. Run `install.exe` file.
4. Select the language that you want to update from the list and click **OK**.
5. Click **Next**.
6. Select the License Agreement option and click **Next**.
7. Click **OK** in **Analyzer for Identity Manager is Running** window.
8. Check the install location path in the **Install Folder** page and click **Next**. In case, you want to modify the location, click the **Choose** button and select the desired path.
9. Click **Yes** in the **Analyzer Found** window.
10. Click **Next** in the **Create Shortcuts** page.
11. Click **Install** in the **Pre-Installation Summary** page.
12. Click **Done** in the **Install Complete** page.

6 Updating Sentinel Log Management for IGA

NOTE: This service pack does not support a new version of Sentinel. If you are using the Sentinel Log Management for IGA 8.5.0.1 version, skip this procedure.

This service pack includes the `SentinelLogManagementForIGA8.5.0.1.tar.gz` file for updating the Sentinel Log Management for Identity Governance and Administration (IGA) component. Ensure that the required port is available before you update Sentinel.

1. Download the `SentinelLogManagementForIGA8.5.0.1.tar.gz` file to the server where you want to install this version.
2. Run the following command to extract the file:

   ```
   tar -zxvf SentinelLogManagementForIGA8.5.0.1.tar.gz
   ```

   NOTE: Ensure that you extract the `SentinelLogManagementForIGA8.5.0.1.tar.gz` file to a directory that has `novell` user permissions. NetIQ recommends that you extract the file under the `tmp` or `opt` directories.
3. Navigate to the `SentinelLogManagementforIGA` directory.
4. To install Sentinel Log Management for IGA, run the following command:

   ```
   ./install.sh
   ```

   NOTE: Identity Manager 4.8.6 supports Universal CEF Collector 2011.1r5 for CEF auditing.

7 Troubleshooting

This section provides useful information for troubleshooting problems with standalone deployment of the Identity Manager.

Identity Manager Engine Reports an Error After Upgrading iManager to Version 3.2.6.0300

Issue: After upgrading iManager to version 3.2.6.0300, you might encounter the following error in Identity Manager Engine component: `java.lang.NoClassDefFoundError: Could not initialize class novel.jclient.JCContext`

This issue happens because Identity Manager Engine 4.8.6 and iManager 3.2.6.0300 do not support the same version of Java.

Workaround: After upgrading iManager to version 3.2.6.0300, perform the following steps:

1. Log in to the server where iManager is installed.
2. Copy the `jclient.jar` file from the iManager `lib` folder and to eDIrectory `lib` folder using the following command:

   ```
   cp /var/opt/novell/iManager/nps/WEB-INF/lib/jclient.jar /opt/novell/eDirectory/lib64/jclient.jar
   ```

3. Restart nds using the following commands:

   ```
   ndsmanage stopall
   ndsmanage startall
   ```

II Deploying Identity Manager Containers

This section guides you through the process of deploying Identity Manager components using containers.

Identity Manager provides the flexibility of deploying Identity Manager components through a containerized mechanism. Identity Manager uses Docker for managing containers. The Identity Manager components, that support containerization, are delivered as Docker images. The Docker images are self-sufficient to run on their own.

All the functionalities and operations that can be achieved through the enterprise mode of installation are also available through the containerized mechanism.

However, the advantage of using containers is the ability to perform a fresh installation with every new version of containers along with the option of updating from previous versions. NetIQ recommends you to directly use the 4.8.6 version of containers if you are using containers for the first time.

8 Overview and Planning

The following sections describe the high-level planning required for a container-based deployment in Docker environment:

System Requirements

You must ensure that the following requirements are met for deploying the containers:

Software	Certified Versions
Docker	20.10.14

Obtaining the Docker Images

Perform the following steps to obtain the Docker images:

1. Download the `Identity_Manager_4.8.6_Containers.tar.gz` from the Software License and Download page.
2. Run the following command to extract the `.tar.gz` file:

   ```
   tar -zxvf Identity_Manager_4.8.6_Containers.tar.gz
   ```

Managing Container Volume Data

Docker supports several mechanisms for data storage and persistence. One such mechanism of persisting container data is by using shared directory in containers.

The examples used in this guide assumes that you create and use shared directory. For example, create a shared directory called `/data` on your Docker host.

```
mkdir /data
```

However, you can use other data storage and persistence mechanisms that Docker supports. For more information, see Docker documentation.

NOTE:
- The /data directory of the Docker host will be mapped to the /config directory of the containers. Ensure that you have read-write permissions for the shared directory. However, if you want to map the shared directory with a different directory inside the container, you must map them while deploying the container itself. For example, you can map the /data directory with the /etc/opt/novell/dirxml/rdxml/ directory inside the Remote Loader container.
- The shared directory must only be used by Identity Manager containers. It is recommended that you do not use the same shared directory for any third party containers.

Handling RPM Updates and Third Party Files

This service pack provides an efficient way to handle RPM updates and third-party files in a container. This can be achieved by placing the required RPM files, library (.so) files, and third-party .jar files in the mountfiles directory. The RPM files present in the mountfiles directory will be updated forcefully. The .so and .jar files are automatically soft linked to the /opt/novell/eDirectory/lib64/nds-modules/ and /opt/novell/eDirectory/lib/dirxml/classes/ directories respectively when the containers are deployed. However, each time you want to handle any supported files after the containers are deployed, place those files in the mountfiles directory and restart the container.

NOTE: Currently, this enhancement is applicable for Identity Manager and Remote Loader containers.

1 On your Docker host, navigate to the shared directory. For example, /data.

2 (Optional) For Identity Manager Engine container, perform the following steps:

2a Create the idm directory.

```
mkdir idm
```

NOTE: This applies only when you are deploying containers for the first time. In other words, if you are updating the Identity Manager Engine container and have already created the idm directory before, skip this step.

2b Navigate to the idm directory.

2c Create the mountfiles directory.

```
mkdir mountfiles
```

3 (Optional) For Remote Loader container, perform the following steps:

3a Create the rdxml directory.

```
mkdir rdxml
```

NOTE: This applies only when you are deploying containers for the first time. In other words, if you are updating the Remote Loader container and have already created the rdxml directory before, skip this step.

3b Navigate to the rdxml directory.

3c Create the mountfiles directory.

```
mkdir mountfiles
```

4 Copy the required files to the respective container-specific `mountfiles` directory.

For example, if you want to patch a driver to the latest version, place the driver RPM file in the `/data/idm/mountfiles` directory.

NOTE: The supported file formats are `.so`, `.jar`, and `.rpm`.

5 Deploy the container. For example, see Deploying Identity Manager Engine Container.

6 (Conditional) If you want to handle additional files after the container is deployed, perform the following steps:

6a Place the files in the `mountfiles` directory. For example, `/data/idm/mountfiles`.

6b Restart the container.

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

IMPORTANT: Whatever the rpm present inside the mount files is prioritized, in case of future updates you need to carefully remove the existing rpm and update the latest one.

Starting Remote Loader Instances Automatically With Remote Loader Container Deployment

If you want to start the Remote Loader instances automatically once the Remote Loader container is deployed, perform the following steps:

1 On your Docker host, navigate to the shared directory. For example, `data`.

2 Create the `rdxml` directory.

NOTE: This applies only when you are deploying containers for the first time. In other words, if you are updating the containers and have already created the `rdxml` directory before, skip this step.

3 Navigate to the `rdxml` directory.

4 In the `rdxml` directory, create the `driverconf` directory.

NOTE: If you have multiple configuration files running with different Remote Loader instances, copy all the files.

5 Copy all the required configuration files to the `driverconf` directory. For example, `config8000.txt`.

6 In the `rdxml` directory, create the `keystore` directory.

7 Copy all the required keystore files and certificates to the `keystore` directory.

8 In the `rdxml` directory, create a new `.txt` file. For example, `StartupRL.txt`.

9 In the `StartupRL.txt` file, specify the required content in the following format:

```
<Remote Loader configuration file> -sp <driver password> <Remote Loader password>
```

For example:

```
config8000.txt -sp dirxml dirxml
```

Alternatively, you can also specify the entries in the following format:

```
<Remote Loader configuration file> -ksp <keystore password> -kp <key password>
```

For example:

```
config8000.txt -ksp dirxml -kp dirxml
```

10 Deploy the Remote Loader container by passing the `RL_DRIVER_STARTUP` environment variable in the docker run command. For example, `-e RL_DRIVER_STARTUP="StartupRL.txt"`.

For more information, see "Deploying Remote Loader Container" on page 60.

9 Fresh Deployment of Identity Manager Containers

This section guides you through the process of installing Identity Manager containers. After Identity Manager containers are deployed, you must perform some additional configuration steps for the components to be fully functional. For more information, see Final Steps for Completing the Installation section in the NetIQ Identity Manager Setup Guide for Linux.

The Docker images are available for the following Identity Manager components:

- Identity Manager Engine
- Remote Loader
- iManager
- One SSO Provider (OSP)
- Fanout Agent
- ActiveMQ
- PostgreSQL (Redistribution)
- Identity Applications
- Self Service Password Reset (SSPR)
- Form Renderer
- Identity Reporting

NOTE: The Identity Configuration Generator image is used for generating the silent properties file. For information about creating the silent properties file, see “Creating the Silent Properties File” on page 56.

The procedures for deploying containers are described in subsequent sections.

- “Preparing Your Container Deployment” on page 53
- “Deploying Containers on Distributed Servers” on page 57
- “Deploying Containers on a Single Server” on page 73

Preparing Your Container Deployment

The Identity Manager containers deployment process requires pre-installation, installation, and post-installation work. Use the information in this section as you prepare to deploy the Identity Manager containers.

Some containers are dependent on others. The following table provides details on those containers that are dependent on other containers.

Table 9-1 Dependent Containers

Container	Dependent containers
OSP	- Identity Manager Engine - iManager
Identity Applications	- OSP - Databases for Identity Applications
Form Renderer	Identity Applications
Identity Reporting	- Identity Applications - Databases for Identity Reporting
SSPR	OSP

Prerequisites for Deploying Containers

Based on your container deployment, NetIQ recommends that you review the following prerequisites before deploying containers.

- The `/etc/hosts` file of all the Docker hosts in your Docker deployment must be updated with the details of all the containers running on that host. Ensure that the hostname for all containers are in Fully Qualified Domain Name (FQDN) format only.
  - If you are deploying containers on distributed servers, ensure that the host file entries follows the below format for all the components:

    ```
    <IP of the container> <FQDN> <short_name>
    ```

    In the sample deployment used in this guide, add the following entries in the `/etc/hosts` file:

    ```
    192.168.0.12    identityengine.example.com       identityengine
    192.168.0.2     remoteloader.example.com         remoteloader
    192.168.0.3     fanoutagent.example.com          fanoutagent
    192.168.0.4     imanager.example.com             imanager
    192.168.0.5     osp.example.com                  osp
    192.168.0.6     postgresql.example.com           postgresql
    192.168.0.7     identityapps.example.com         identityapps
    192.168.0.8     formrenderer.example.com         formrenderer
    192.168.0.9     activemq.example.com             activemq
    192.168.0.10   identityreporting.example.com   identityreporting
    192.168.0.11    sspr.example.com                 sspr
    ```

    You must also add the following entries on the `hosts` file of the machine where you will access the containers from:

```
<IP Address of Docker host A>   <FQDN of all containers deployed on
Docker Host A>   <short name of all containers deployed on Docker
host A>

<IP Address of Docker host B>   <FQDN of all containers deployed on
Docker Host B>   <short name of all containers deployed on Docker
host B>
```

- If you are deploying containers on a single server, ensure that the host file entry follows the below format:

  ```
  <IP of the host> <FQDN> <short_name>
  ```

  For example:

  ```
  172.120.0.1     identitymanager.example.com      identitymanager
  ```

NOTE: The examples in the guide assume virtual IP addresses for all the containers. Based on your requirement, you can assign IP addresses that are accessible across your network.

- You must know the ports that you want to use for each containers in your deployment. You must expose the required ports and map the container ports with the ports on the Docker host. The following table provides information on ports that you must expose on the Docker hosts based on the examples provided in the guide.

Table 9-2 *Default Ports Exposed As per the Sample Deployment*

Container	Default ports assumed as per the sample deployment
Remote Loader	8090
Fanout Agent	Not applicable
iManager	8743
iMonitor	8030
OSP	8543
Identity Applications	18543
Identity Reporting	28543
Form Renderer	8600
ActiveMQ	61616
PostgreSQL	5432
SSPR	8443 **NOTE:** SSPR container runs only on 8443 port.

However, you can customize the ports based on your requirement. The following considerations apply while you expose the ports:

- Ensure that you expose those ports that are not in use.
- The container port must be mapped to the same port on the Docker host. For example, the 8543 port on the container must be mapped to the 8543 port on the Docker host.

Creating the Silent Properties File

Identity Manager supports silent mode only for deployment of containers. You must generate the silent properties file if you are deploying containers for the first time. If you are updating containers from previous versions, the silent properties file is not required.

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_idm_conf_generator.tar.gz
   ```

4. Deploy the container using the following command:

   ```
   docker run --rm -it --name=idm_conf_generator --
   hostname=identitymanager.example.com -v /data:/config
   idm_conf_generator:idm-4.8.6
   ```

   NOTE: - Ensure that you specify the machine FQDN as a value for the hostname.
   - The `--rm` flag deletes the container after the silent properties file is created.

5. Specify `n` for the **Do you want to deploy Identity Manager Containers on Azure** parameter.
6. Specify the silent property file name with the absolute path:

   NOTE: Ensure that you create the `silent.properties` file in the `/config` shared directory location. In other words, the silent properties file will be available in the `/data` directory of the Docker host.

7. Specify `n` for the **Do you want to generate inputs for Kubernetes Orchestration** parameter.
8. Decide the Identity Manager server edition you want to install. Enter `y` for Advanced Edition and `n` for Standard Edition.
9. From the list of components available for installation, select the required components:
   - To install Identity Manager Engine, select **Identity Manager Engine**.
   - To install Identity Reporting, select **Identity Reporting**.
   - To install Identity Applications, select **Identity Applications**.

   NOTE: - You must generate a single `silent.properties` file for deploying all the Identity Manager components.
   - Ensure that you specify the following values for the ports used by different containers:

Prompt	Port to be specified
One SSO Server SSL port	8543
Identity Reporting Tomcat HTTPS port	28543
Identity Applications Tomcat HTTPS port	18543

- Use FQDN for all IP related configuration prompts. In other words, the hostname that you provide in the `/etc/hosts` entry for all components must be specified while generating the `silent.properties` file.
- The `SSO_SERVER_SSL_PORT, TOMCAT_HTTPS_PORT, UA_SERVER_SSL_PORT`, and `RPT_TOMCAT_HTTPS_PORT` must be unique ports.

10 (Conditional) If you are deploying containers on a single server using the host network mode, you must perform the following tasks after the silent properties file is generated:

10a Modify the `TOMCAT_HTTPS_PORT` and `UA_SERVER_SSL_PORT` to *18543*, and `RPT_TOMCAT_HTTPS_PORT` to *28543* respectively.

10b Remove the `SSO_SERVER_SSL_PORT` parameter from the `silent.properties` file.

```
sed -i.bak '/SSO_SERVER_SSL_PORT/d' silent.properties
```

10c Add the following parameters:

```
SSO_SERVER_SSL_PORT=8543
SKIP_PORT_CHECK=1
```

NOTE: When the `silent.properties` file is generated, it will be available in the shared directory of your Docker host. For example, `/data`.

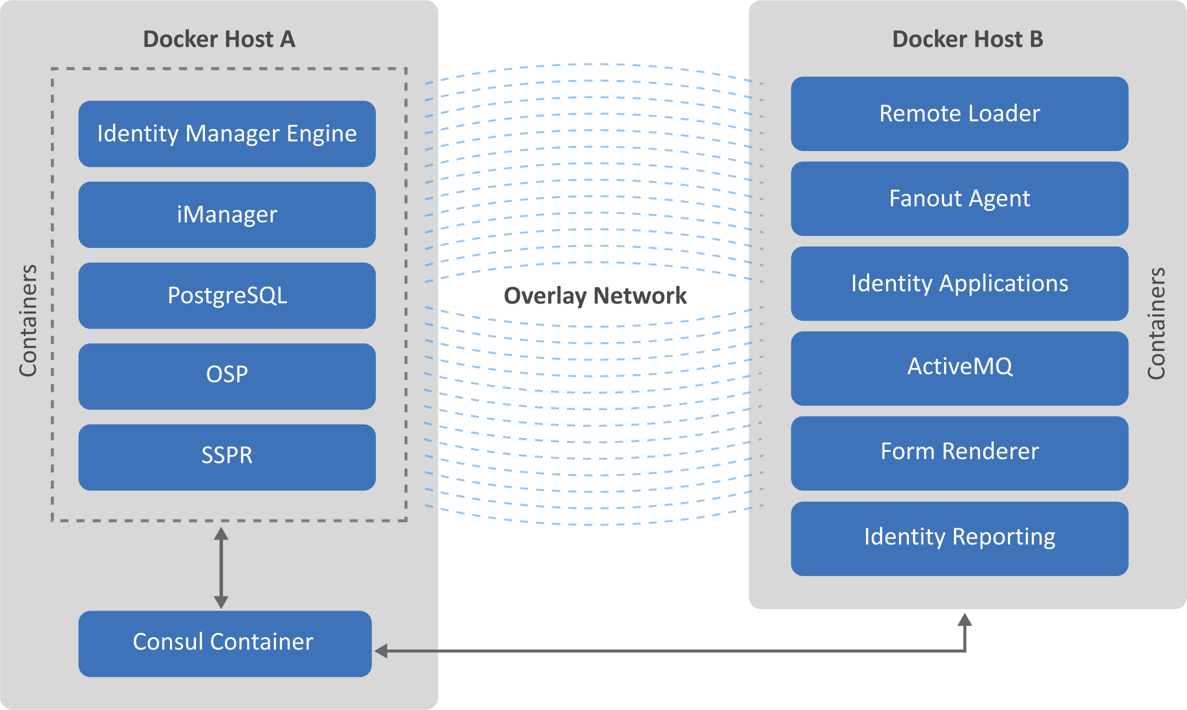
Deploying Containers on Distributed Servers

NetIQ recommends you to use overlay or bridge network mode for deploying all Identity Manager containers in a distributed setup. The scenarios documented in the guide provide instructions and commands to deploy containers in a overlay network. However, you can also use bridge network for deploying containers.

In the following distributed servers scenario, the Identity Manager Engine, iManager, PostgreSQL, OSP, and SSPR containers will be deployed on Docker Host A. On Docker Host B, the Remote Loader, Fanout Agent, Identity Applications, ActiveMQ, Form Renderer, and Identity Reporting containers will be deployed. The Consul container will be deployed on Docker host A. However, you can deploy the Consul container on any of the Docker hosts in your deployment.

The following figure illustrates the deployment of Identity Manager containers on two Docker hosts in a overlay network.

Figure 9-1 Containers Deployment Architecture in an Overlay Network



The containers must be deployed in the following order:

- "Setting Up an Overlay Network" on page 58
- "Deploying Identity Manager Engine Container" on page 59
- "Deploying Remote Loader Container" on page 60
- "Deploying Fanout Agent Container" on page 61
- "Deploying iManager Container" on page 61
- "Generating Certificates With Identity Vault Certificate Authority" on page 63
- "Deploying OSP Container" on page 67
- "Deploying PostgreSQL Container" on page 67
- "Deploying Identity Applications Container" on page 69
- "Deploying Form Renderer Container" on page 70
- "Deploying ActiveMQ Container" on page 70
- "Deploying Identity Reporting Container" on page 71
- "Deploying SSPR Container" on page 72
- "Deploying Identity Console Container" on page 73

Setting Up an Overlay Network

Perform the following steps to set up an overlay network:

1 Run the following command on Docker Host A:

```
docker run -d -p <host port>:8500 -h consul --name <container name> --
restart unless-stopped progrium/consul -server -bootstrap
```

For example:

```
docker run -d -p 8500:8500 -h consul --name consul --restart unless-
stopped progrium/consul -server -bootstrap
```

2 On both the Docker Hosts, edit the **docker** file located at `/etc/sysconfig/` directory and add the following line:

```
DOCKER_OPTS="-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock --
cluster-advertise <Master Server Network Interface>:2375 --cluster-
store consul://<Docker Host A IP Address>:<Docker Host A Port>"
```

For example:

```
DOCKER_OPTS="-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sock --
cluster-advertise eth0:2375 --cluster-store consul://172.120.0.1:8500"
```

3 Restart the Docker service on both the Docker hosts:

```
systemctl restart docker
```

NOTE: Try rebooting the machine if the docker restart fails.

4 On Docker Host B, run the following command to check whether Docker Host B is added to the cluster:

```
docker info
```

The sample output will be as follows:

```
Cluster store: consul://<Docker HOST A IP Address>:8500

Cluster advertise: <Docker HOST B IP Address>:2375
```

5 Create an overlay network on any of the Docker hosts:

```
docker network create -d overlay --subnet=<subnet in CID format that
represents a network segment> --gateway=<ipv4 gateway> <name of the
overlay network>
```

For example:

```
docker network create -d overlay --subnet=192.168.0.0/24 --
gateway=192.168.0.1 idmoverlaynetwork
```

6 Run the following command to verify whether the overlay network is created:

```
docker network ls
```

Deploying Identity Manager Engine Container

1 Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.

2 (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.

3 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

4 Navigate to the `docker-images` directory.

5. Run the following command to load the image:

   ```
   docker load --input IDM_486_identityengine.tar.gz
   ```

6. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.12 --
   network=idmoverlaynetwork --hostname=identityengine.example.com --
   name=engine-container -v /etc/hosts:/etc/hosts -v /data:/config -p
   8028:8028 -p 524:524 -p 389:389 -p 8030:8030 -p 636:636 -e
   SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100
   identityengine:idm-4.8.6
   ```

7. To verify whether the container was successfully deployed, check the log files by running the following command:

   ```
   tail -f /data/idm/log/idmconfigure.log
   ```

8. To log in to the container, run the following command:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it engine-container bash
   ```

NOTE: To run the Identity Vault utilities such as `ndstrace` or `ndsrepair`, log in to the container as a non-root user called as `nds`. These utilities cannot be run if you are logged in as a root user. To log in to the container as a `nds` user, run the `docker exec -it engine-container su nds` command.

Deploying Remote Loader Container

1. (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.
2. (Conditional) To start Remote Loader instances automatically with the container, perform the steps mentioned in Starting Remote Loader Instances Automatically With Remote Loader Container Deployment.
3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

   ```
   docker load --input IDM_486_remoteloader.tar.gz
   ```

6. (Conditional) If you do not want to use configuration files while deploying the container, deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.2 --
   network=idmoverlaynetwork --hostname=remoteloader.example.com -p
   8090:8090 --name=rl-container -v /etc/hosts:/etc/hosts -v /data:/config
   --stop-timeout 100 remoteloader:idm-4.8.6
   ```

The driver files can be found at the `/opt/novell/eDirectory/lib/dirxml/classes/` directory of the container.

NOTE: The 32-bit Remote Loader is not supported with containers.

7 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it rl-container bash
```

8 Configure Remote Loader. For more information, see Configuring the Remote Loader and Drivers in the *NetIQ Identity Manager Driver Administration Guide*.

Deploying Fanout Agent Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input IDM_486_fanoutagent.tar.gz
```

4 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.3 --
network=idmoverlaynetwork --hostname=fanoutagent.example.com --
name=foa-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-
timeout 100 fanoutagent:idm-4.8.6
```

5 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it foa-container bash
```

6 Configure the Fanout Agent. For more information, see Configuring the Fanout Agent in the *NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide*.

Deploying iManager Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input iManager_326_P2.tar.gz
```

4 Create a `.env` file with the required configuration to suit your environment. For example, the `iManager.env` is created in the `/data` directory.

```
# Certificate Public Key Algorithm
# Allowed Values: RSA, ECDSA256, ECDSA384
CERTIFICATE_ALGORITHM=RSA
# Cipher Suite
# Allowed Values:
# For RSA - NONE, LOW, MEDIUM HIGH
# For ECDSA256 - SUITEB128ONLY
# For ECDSA384 - SUITEB128, SUITEB192
CIPHER_SUITE=NONE
# Tomcat Server HTTP Port
TOMCAT_HTTP_PORT=8080
# Tomcat Server SSL Port
TOMCAT_SSL_PORT=8743
# iManager Authorized User (admin_name.container_name.tree_name)
AUTHORIZED_USER=
```

5. Create a sub-directory called as `iManager` under the shared directory `/data`.
6. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.4 --name=iman-
   container --network=idmoverlaynetwork --hostname=imanager.example.com -
   v /etc/hosts:/etc/hosts -v /data:/config -v /data/iManager.env:/etc/
   opt/novell/iManager/conf/iManager.env -p 8743:8743 --stop-timeout 100
   imanager:3.2.6-p2
   ```

7. To install the Identity Manager plug-ins, perform the following steps:
   1. Download the `Identity_Manager_4.8.6_Linux.iso` from the NetIQ Downloads website.
   2. Mount the downloaded `.iso`.
   3. From the mounted location, navigate to the `/iManager/plugins` directory and obtain the required plug-ins.

   Alternatively, you can install the plug-ins from the iManager plug-ins website.
8. Restart the iManager container.

   ```
   docker restart iman-container
   ```

9. To log in to the container, run the following command:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it iman-container bash
   ```

For more information about deploying the iManager container, see the Deploying iManager Using Docker Container in the *NetIQ iManager Installation Guide*.

Generating Certificates With Identity Vault Certificate Authority

(Conditional) This section applies only if you are using Identity Vault as the Certificate Authority.

The following components require you to generate certificates before they are deployed. Before you generate the certificates for the following components, ensure that you deploy the Identity Manager Engine and iManager containers.

- OSP
- Identity Applications
- Identity Reporting

Generating Certificates for OSP

Perform the following steps to generate the certificates:

1 Log in to the iManager container.

```
docker exec -it -u root <container> <command>
```

For example,

```
docker exec -it -u root iman-container bash
```

2 Ensure that you set the Java path. For example, run the following command:

```
export PATH=<java installed location>/bin:$PATH
```

For example,

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

NOTE: Ensure that the Java version installed is Azul Zulu 1.80_292 or later.

3 Generate the PKCS keystore:

```
keytool -genkey -alias osp -keyalg RSA -storetype pkcs12 -keystore /
config/tomcat-osp.ks -validity 3650 -keysize 2048 -dname
"CN=osp.example.com" -keypass <password> -storepass <password>
```

4 Generate a certificate signing request:

```
keytool -certreq -v -alias osp -file /config/osp.csr -keypass
<password> -keystore /config/tomcat-osp.ks -storepass <password>
```

5 Generate a self-signed certificate:

5a Launch iManager from Docker host and log in as an administrator.

5b Navigate to **Roles and Tasks** > **NetIQ Certificate Server** > **Issue Certificate**.

5c Browse to the `.csr` file created in step 3. For example, `osp.csr`.

5d Click **Next**.

5e Specify the key usage and click **Next**.

5f For the certificate type, select **Unspecified**.

5g Click **Next**.

5h Specify the validity of the certificate and click **Next**.

5i Select the **File in binary DER format radio** button.

5j Click **Next**.

5k Click **Finish**.

5l Download the certificate and copy the downloaded certificate to the `/data` directory.

6 Export the root certificate in `.der` format:

6a Launch iManager from Docker host and log in as an administrator.

6b Navigate to **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**.

6c Select the **SSL CertificateDNS** check box and click **Export**.

6d In the **Certificates** drop-down list, select the Organizational CA.

6e In the **Export Format** drop-down list, select `DER`.

6f Click **Next**.

6g Download the certificate and copy the downloaded certificate to the `/data` directory.

7 Import the certificates into the PKCS keystore you created in step 2:

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat-
osp.ks -file /config/cert.der -storepass <password> -noprompt
```

```
keytool -import -alias osp -keystore /config/tomcat-osp.ks -file /
config/osp.der -storepass <password> -noprompt
```

NOTE: Ensure that the keystore is available in the path that was specified as an input for deployment.

Generating Certificates for Identity Applications

Perform the following steps to generate the certificates:

1 Log in to the iManager container.

```
docker exec -it -u root <container> <command>
```

For example,

```
docker exec -it -u root iman-container bash
```

2 Ensure that you set the Java path. For example, run the following command:

```
export PATH=<java installed location>/bin:$PATH
```

For example,

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

NOTE: Ensure that the Java version installed is Azul Zulu 1.80_292 or later.

3 Generate the PKCS keystore:

```
keytool -genkey -alias ua -keyalg RSA -storetype pkcs12 -keystore /
config/tomcat-ua.ks -validity 3650 -keysize 2048 -dname
"CN=identityapps.example.com" -keypass <password> -storepass <password>
```

4 Generate a certificate signing request:

```
keytool -certreq -v -alias ua -file /config/ua.csr -keypass <password>
-keystore /config/tomcat-ua.ks -storepass <password>
```

5 Generate a self-signed certificate:

5a Log in to iManager as an administrator.

5b Navigate to **Roles and Tasks** > **NetIQ Certificate Server** > **Issue Certificate**.

5c Browse to the `.csr` file created in step 3. For example, `ua.csr`.

5d Click **Next**.

5e Specify the key usage and click **Next**.

5f For the certificate type, select **Unspecified**.

5g Click **Next**.

5h Specify the validity of the certificate and click **Next**.

5i Select the **File in binary DER format radio** button.

5j Click **Next**.

5k Click **Finish**.

5l Download the certificate and copy the downloaded certificate to the `/data` directory.

6 Export the root certificate in `.der` format:

6a Log in to iManager as an administrator.

6b Navigate to **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**.

6c Select the **SSL CertificateDNS** check box and click **Export**.

6d In the **Certificates** drop-down list, select the Organizational CA.

6e In the Export Format drop-down list, select `DER`.

6f Click **Next**.

6g Download the certificate and copy the downloaded certificate to the `/data` directory.

7 Import the certificates into the PKCS keystore in step 2:

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat-ua.ks -file /config/cert.der -storepass <password> -noprompt
```

```
keytool -import -alias ua -keystore /config/tomcat-ua.ks -file /config/ua.der -storepass <password> -noprompt
```

NOTE: Ensure that the certificates are available in the path that was specified as an input for deployment.

Generating Certificates for Identity Reporting

Perform the following steps to generate the certificates:

1 Log in to the iManager container.

```
docker exec -it -u root <container> <command>
```

For example,

```
docker exec -it -u root iman-container bash
```

2 Ensure that you set the Java path. For example, run the following command:

```
export PATH=<java installed location>/bin:$PATH
```

For example,

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

NOTE: Ensure that the Java version installed is Azul Zulu 1.80_292 or later.

3 Generate the PKCS keystore:

```
keytool -genkey -alias rpt -keyalg RSA -storetype pkcs12 -keystore /
config/tomcat-rpt.ks -validity 3650 -keysize 2048 -dname
"CN=identityreporting.example.com" -keypass <password> -storepass
<password>
```

4 Generate a certificate signing request:

```
keytool -certreq -v -alias rpt -file /config/rpt.csr -keypass
<password> -keystore /config/tomcat-rpt.ks -storepass <password>
```

5 Generate a self-signed certificate:

- 5a Log in to iManager as an administrator.
- 5b Navigate to **Roles and Tasks** > **NetIQ Certificate Server** > **Issue Certificate**.
- 5c Browse to the `.csr` file created in step 3. For example, `rpt.csr`.
- 5d Click **Next**.
- 5e Specify the key usage and click **Next**.
- 5f For the certificate type, select **Unspecified**.
- 5g Click **Next**.
- 5h Specify the validity of the certificate and click **Next**.
- 5i Select the **File in binary DER format radio** button.
- 5j Click **Next**.
- 5k Click **Finish**.
- 5l Download the certificate and copy the downloaded certificate to the `/data` directory.

6 Export the root certificate in `.der` format:

- 6a Log in to iManager as an administrator.
- 6b Navigate to **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**.
- 6c Select the **SSL CertificateDNS** check box and click **Export**.
- 6d In the **Certificates** drop-down list, select the Organizational CA.
- 6e In the Export Format drop-down list, select `DER`.
- 6f Click **Next**.
- 6g Download the certificate and copy the downloaded certificate to the `/data` directory.

7 Import the certificates into the PKCS keystore you created in step 2:

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat-
rpt.ks -file /config/cert.der -storepass <password> -noprompt
```

```
keytool -import -alias rpt -keystore /config/tomcat-rpt.ks -file /
config/rpt.der -storepass <password> -noprompt
```

NOTE: Ensure that the certificates are available in the path that was specified as an input for deployment.

Deploying OSP Container

NOTE: Before you deploy the OSP container, ensure that you generate the required certificates. For more information, see Generating Certificates for OSP.

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
3. Navigate to the `docker-images` directory.
4. Run the following command to load the image:

   ```
   docker load --input IDM_486_osp.tar.gz
   ```

5. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.5 --
   network=idmoverlaynetwork --hostname=osp.example.com -p 8543:8543 --
   name=osp-container -v /etc/hosts:/etc/hosts -v /data:/config -e
   SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100
   osp:idm-4.8.6
   ```

6. To verify whether the container was successfully deployed, check the log files by running the following command:

   ```
   tail -f /data/osp/log/idmconfigure.log
   ```

7. Run the following command to log in to the container:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it osp-container bash
   ```

8. Navigate to the `/opt/netiq/idm/apps/configupdate/` directory.
9. Modify the `configupdate.sh.properties` file.
10. Set the value of the **no_nam_oauth** parameter to *false*.
11. Save the `configupdate.sh.properties` file.
12. Run the following command to exit the container.

    ```
    exit
    ```

Deploying PostgreSQL Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
2. Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input IDM_486_postgres.tar.gz
```

4 Create a sub-directory under the shared directory `/data`, for example, `postgres`.

```
mkdir postgres
```

5 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=<password>
-v /data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v
/data:/config --stop-timeout 100 postgres:12.12
```

For example,

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=novell -v /
data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v /
data:/config --stop-timeout 100 postgres:12.12
```

6 Create the `idmdamin` user for Identity Applications.

```
docker exec -it postgresql-container psql -U postgres -c "CREATE USER
idmadmin WITH ENCRYPTED PASSWORD '<password>'"
```

7 Create the Identity Applications, Workflow, and Identity Reporting databases.

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmuserappdb"
```

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE igaworkflowdb"
```

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmrptdb"
```

NOTE: These databases are used while you configure the Identity Applications and Identity Reporting containers.

8 Grant all the privileges on the databases for the `idmadmin` user:

```
docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE idmuserappdb TO idmadmin"
```

```
docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE igaworkflowdb TO idmadmin"
```

9 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it postgresql-container bash
```

Deploying Identity Applications Container

NOTE: Before you deploy the Identity Applications container, ensure that you generate the required certificates. For more information, see Generating Certificates for Identity Applications.

1 Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.

   NOTE: Specify the exposed port, 18543, as the value for the application server port.

2 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

3 Navigate to the `docker-images` directory.

4 Run the following command to load the image:

```
docker load --input IDM_486_identityapplication.tar.gz
```

5 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.7 --
network=idmoverlaynetwork --hostname=identityapps.example.com -p
18543:18543 --name=idapps-container -v /etc/hosts:/etc/hosts -v /data:/
config -e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout
100 identityapplication:idm-4.8.6
```

6 To verify whether the container was successfully deployed, check the log files by running the following command:

```
tail -f /data/userapp/log/idmconfigure.log
```

7 Run the following command to log in to the container:

```
docker exec -it <container> <command>
```

   For example,

```
docker exec -it idapps-container bash
```

8 Run the following command:

   NOTE: Before performing this step, ensure that the container is deployed successfully.

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat-osp.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/
apps/tomcat/conf/idm.jks -deststorepass <password>
```

9 Type `yes` to overwrite the entry for the `root` alias.

10 Run the following command to exit the container.

```
exit
```

11 Restart the Identity Applications container.

```
docker restart idapps-container
```

NOTE: To modify any settings in the configuration update utility, launch configupdate.sh from the /opt/netiq/idm/apps/configupdate/ directory of the Identity Applications container. The configuration update utility can be launched in console mode only.

Deploying Form Renderer Container

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Navigate to the location where you have extracted the Identity_Manager_4.8.6_Containers.tar.gz file.
3. Navigate to the docker-images directory.
4. Run the following command to load the image:

   ```
   docker load --input IDM_486_formrenderer.tar.gz
   ```

5. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.8 --
   network=idmoverlaynetwork --hostname=formrenderer.example.com -p
   8600:8600 --name=fr-container -v /etc/hosts:/etc/hosts -v /data:/config
   -e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100
   formrenderer:idm-4.8.6
   ```

6. To log in to the container, run the following command:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it fr-container bash
   ```

Deploying ActiveMQ Container

NOTE: This procedure assumes that you will use the ActiveMQ container with the Identity Applications container. To use the ActiveMQ container with the Fanout Agent container, you must deploy a new instance of the ActiveMQ container with different IP address and ports.

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Navigate to the location where you have extracted the Identity_Manager_4.8.6_Containers.tar.gz file.
3. Navigate to the docker-images directory.
4. Run the following command to load the image:

   ```
   docker load --input IDM_486_activemq.tar.gz
   ```

5. Deploy the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.9 --
network=idmoverlaynetwork --hostname=activemq.example.com -p
61616:61616 --name=amq-container -v /etc/hosts:/etc/hosts -v /data:/
config --env-file /data/silent.properties --stop-timeout 100
activemq:idm-4.8.6
```

6 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it amq-container bash
```

Deploying Identity Reporting Container

NOTE: Before you deploy the Identity Reporting container, ensure that you generate the required certificates. For more information, see Generating Certificates for Identity Reporting.

1 Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.

NOTE: Specify the exposed port, 28543, as the value for the application server port.

2 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

3 Navigate to the `docker-images` directory.

4 Run the following command to load the image:

```
docker load --input IDM_486_identityreporting.tar.gz
```

5 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.10 --
network=idmoverlaynetwork --hostname=identityreporting.example.com -p
28543:28543 --name=rpt-container -v /etc/hosts:/etc/hosts -v /data:/
config -e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout
100 identityreporting:idm-4.8.6
```

6 To verify whether the container was successfully deployed, check the log files by running the following command:

```
tail -f /data/reporting/log/idmconfigure.log
```

7 Run the following command to log in to the container:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it rpt-container bash
```

8 Run the following command:

NOTE: Before performing this step, ensure that the container is deployed successfully.

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat-osp.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/
apps/tomcat/conf/idm.jks -deststorepass <password>
```

9. Type `yes` to overwrite the entry for the `root` alias.
10. Run the following command to exit the container.

    ```
    exit
    ```

11. Restart the Identity Reporting container.

    ```
    docker restart rpt-container
    ```

Deploying SSPR Container

Perform the following tasks to deploy the SSPR container:

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Create a sub-directory under the shared directory `/data`, for example, `sspr`.

    ```
    mkdir sspr
    ```

3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

    ```
    docker load --input IDM_486_sspr.tar.gz
    ```

6. Deploy the container using the following command:

    ```
    docker run --restart unless-stopped -d --ip=192.168.0.11 --
    network=idmoverlaynetwork --hostname=sspr.example.com --name=sspr-
    container -v /etc/hosts:/etc/hosts -v /data/sspr:/config -p 8443:8443 -
    -stop-timeout 100 sspr/sspr-webapp:4.6.0.0
    ```

7. Run the following command from the Docker host to copy the `silent.properties` file from the Docker host to SSPR container:

    ```
    docker cp /data/silent.properties sspr-container:/tmp
    ```

8. Load the silent properties file to the SSPR container.

    ```
    docker exec -it sspr-container /app/command.sh ImportPropertyConfig /
    tmp/silent.properties
    ```

    NOTE: Check if the `SSPRConfiguration.xml` is created under the `/config` directory of SSPR container and verify the content of the file.

9. Import the OAuth certificate to SSPR:

    9a From the Docker host, edit the `SSPRConfiguration.xml` file located at `/data/sspr` directory and set the value of the `configIsEditable` flag to **true** and save the changes.

    9b Launch a browser and enter the `https://sspr.example.com:8443/sspr` URL.

    9c Click **OK**.

9d Log in using administrator credentials, for example, `uaadmin`.

9e Click on the user, for example, `uaadmin`, on the top-right corner and then click **Configuration Editor**.

9f Specify the configuration password and click **Sign In**.

9g Click **Settings** > **Single Sign On (SSO) Client** > **OAuth** and ensure that all URLs use the HTTPS protocol and correct ports.

9h Under **OAuth Server Certificate**, click **Import from Server** to import a new certificate and then click **OK**.

9i Click 💾 at the top-right corner to save the certificate.

9j Review the changes and click **OK**.

9k After the SSPR application is restarted, edit the `SSPRConfiguration.xml` file and set the value of the `configIsEditable` flag to **false** and save the changes.

Deploying Identity Console Container

For more information on deploying Identity Console Container, see Deploying Identity Console as Docker Container.

Deploying Containers on a Single Server

In this example, all the Identity Manager containers are deployed on a single Docker host using the host network mode.

The containers must be deployed in the following order:

- "Deploying Identity Manager Engine Container" on page 74
- "Deploying Remote Loader Container" on page 74
- "Deploying Fanout Agent Container" on page 75
- "Deploying iManager Container" on page 75
- "Generating Certificate With Identity Vault Certificate Authority" on page 77
- "Deploying OSP Container" on page 78
- "Deploying PostgreSQL Container" on page 79
- "Deploying Identity Applications Container" on page 80
- "Deploying Form Renderer Container" on page 81
- "Deploying ActiveMQ Container" on page 82
- "Deploying Identity Reporting Container" on page 82
- "Deploying SSPR Container" on page 83
- "Deploying Identity Console Container" on page 84

Deploying Identity Manager Engine Container

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.
3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

   ```
   docker load --input IDM_486_identityengine.tar.gz
   ```

6. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=engine-
   container -v /data:/config -e SILENT_INSTALL_FILE=/config/
   silent.properties --stop-timeout 100 identityengine:idm-4.8.6
   ```

7. To verify whether the container was successfully deployed, check the log files by running the following command:

   ```
   tail -f /data/idm/log/idmconfigure.log
   ```

8. To log in to the container, run the following command:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it engine-container bash
   ```

NOTE: To run the Identity Vault utilities such as `ndstrace` or `ndsrepair`, log in to the container as a non-root user called as `nds`. These utilities cannot be run if you are logged in as a root user. To log in to the container as a `nds` user, run the `docker exec -it engine-container su nds` command.

Deploying Remote Loader Container

1. (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.
2. (Conditional) To start Remote Loader instances automatically with the container, perform the steps mentioned in Starting Remote Loader Instances Automatically With Remote Loader Container Deployment.
3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

   ```
   docker load --input IDM_486_remoteloader.tar.gz
   ```

6 (Conditional) If you do not want to use configuration files while deploying the container, deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=rl-
container -v /data:/config --stop-timeout 100 remoteloader:idm-4.8.6
```

The driver files can be found at the `/opt/novell/eDirectory/lib/dirxml/classes/` directory of the container.

NOTE: The 32-bit Remote Loader is not supported with containers.

7 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it rl-container bash
```

8 Configure Remote Loader. For more information, see Configuring the Remote Loader and Drivers in the *NetIQ Identity Manager Driver Administration Guide*.

Deploying Fanout Agent Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input IDM_486_fanoutagent.tar.gz
```

4 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=foa-
container -v /data:/config --stop-timeout 100 fanoutagent:idm-4.8.6
```

5 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it foa-container bash
```

6 Configure the Fanout Agent. For more information, see Configuring the Fanout Agent in the *NetIQ Identity Manager Driver for JDBC Fanout Implementation Guide*.

Deploying iManager Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input iManager_326_P2.tar.gz
```

4 Create a `.env` file with the required configuration to suit your environment. For example, the `iManager.env` is created in the `/data` directory.

```
# Certificate Public Key Algorithm
# Allowed Values: RSA, ECDSA256, ECDSA384
CERTIFICATE_ALGORITHM=RSA
# Cipher Suite
# Allowed Values:
# For RSA - NONE, LOW, MEDIUM HIGH
# For ECDSA256 - SUITEB128ONLY
# For ECDSA384 - SUITEB128, SUITEB192
CIPHER_SUITE=NONE
# Tomcat Server HTTP Port
TOMCAT_HTTP_PORT=8080
# Tomcat Server SSL Port
TOMCAT_SSL_PORT=8743
# iManager Authorized User (admin_name.container_name.tree_name)
AUTHORIZED_USER=
```

5 Create a sub-directory called as `iManager` under the shared directory `/data`.

6 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=iman-
container -v /data:/config -v /data/iManager.env:/etc/opt/novell/
iManager/conf/iManager.env --stop-timeout 100 imanager:3.2.6-p2
```

7 To install the Identity Manager plug-ins, perform the following steps:

1. Download the `Identity_Manager_4.8.6_Linux.iso` from the NetIQ Downloads website.
2. Mount the downloaded `.iso`.
3. From the mounted location, navigate to the `/iManager/plugins` directory and obtain the required plug-ins.

Alternatively, you can install the plug-ins from the iManager plug-ins website.

8 Restart the iManager container.

```
docker restart iman-container
```

9 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it iman-container bash
```

For more information about deploying the iManager container, see the Deploying iManager Using Docker Container in the *NetIQ iManager Installation Guide*.

Generating Certificate With Identity Vault Certificate Authority

(Conditional) This section applies only if you are using Identity Vault as the Certificate Authority.

The following components require you to generate certificate before they are deployed. Before you generate the certificates for the following components, ensure that you deploy the Identity Manager Engine and iManager containers.

- OSP
- Identity Applications
- Identity Reporting

Perform the following steps to generate the certificate:

1 Log in to the iManager container.

```
docker exec -it -u root <container> <command>
```

For example,

```
docker exec -it -u root iman-container bash
```

2 Ensure that you set the Java path. For example, run the following command:

```
export PATH=<java installed location>/bin:$PATH
```

For example,

```
export PATH=/opt/netiq/common/jre/bin/:$PATH
```

NOTE: Ensure that the Java version installed is Azul Zulu 1.80_292 or later.

3 Generate the PKCS keystore:

```
keytool -genkey -alias idm -keyalg RSA -storetype pkcs12 -keystore /
config/tomcat.ks -validity 3650 -keysize 2048 -dname
"CN=identitymanager.example.com" -keypass <password> -storepass
<password>
```

4 Generate a certificate signing request:

```
keytool -certreq -v -alias idm -file /config/idm.csr -keypass
<password> -keystore /config/tomcat.ks -storepass <password>
```

5 Generate a self-signed certificate:

5a Launch iManager from Docker host and log in as an administrator.

5b Navigate to **Roles and Tasks** > **NetIQ Certificate Server** > **Issue Certificate**.

5c Browse to the `.csr` file created in step 3. For example, `idm.csr`.

5d Click **Next**.

5e Specify the key usage and click **Next**.

5f For the certificate type, select **Unspecified**.

5g Click **Next**.

5h Specify the validity of the certificate and click **Next**.

5i Select the **File in binary DER format radio** button.

5j Click **Next**.

5k Click **Finish**.

5l Download the certificate and copy the downloaded certificate to the `/data` directory.

6 Export the root certificate in `.der` format:

6a Launch iManager from Docker host and log in as an administrator.

6b Navigate to **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**.

6c Select the **SSL CertificateDNS** check box and click **Export**.

6d In the **Certificates** drop-down list, select the Organizational CA.

6e In the **Export Format** drop-down list, select `DER`.

6f Click **Next**.

6g Download the certificate and copy the downloaded certificate to the `/data` directory.

7 Import the certificates into the PKCS keystore you created in step 2:

```
keytool -import -trustcacerts -alias root -keystore /config/tomcat.ks -
file /config/cert.der -storepass <password> -noprompt
```

```
keytool -import -alias idm -keystore /config/tomcat.ks -file /config/
idm.der -storepass <password> -noprompt
```

NOTE: Ensure that the keystore is available in the path that was specified as an input for deployment.

Deploying OSP Container

NOTE: Before you deploy the OSP container, ensure that you generate the required certificate. For more information, see Generating Certificate With Identity Vault Certificate Authority.

1 Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.

2 Ensure that the `SSO_SERVER_SSL_PORT` property is set to a unique port.

3 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

4 Navigate to the `docker-images` directory.

5 Run the following command to load the image:

```
docker load --input IDM_486_osp.tar.gz
```

6 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=osp-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 osp:idm-4.8.6
```

7 To verify whether the container was successfully deployed, check the log files by running the following command:

```
tail -f /data/osp/log/idmconfigure.log
```

8 Stop the container using the following command:

```
docker stop osp-container
```

9 Run the following command to modify the Tomcat shutdown port in the `server.xml` file. In the following example, the port 8005 will be changed to 18005:

```
sed -i "s~8005~18005~g" /data/osp/tomcat/conf/server.xml
```

10 Start the container using the following command:

```
docker start osp-container
```

11 Run the following command to log in to the container:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it osp-container bash
```

12 Navigate to the `/opt/netiq/idm/apps/configupdate/` directory.

13 Modify the `configupdate.sh.properties` file.

14 Set the value of the **no_nam_oauth** parameter to *false*.

15 Save the `configupdate.sh.properties` file.

16 Run the following command to exit the container.

```
exit
```

Deploying PostgreSQL Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input IDM_486_postgres.tar.gz
```

4 Create a sub-directory under the shared directory `/data`, for example, `postgres`.

```
mkdir postgres
```

5 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=<password> -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.12
```

For example,

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=novell -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.12
```

6 Create the `idmdamin` user for Identity Applications.

```
docker exec -it postgresql-container psql -U postgres -c "CREATE USER
idmadmin WITH ENCRYPTED PASSWORD '<password>'"
```

7 Create the Identity Applications, Workflow, and Identity Reporting databases.

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmuserappdb"
```

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE igaworkflowdb"
```

```
docker exec -it postgresql-container psql -U postgres -c "CREATE
DATABASE idmrptdb"
```

NOTE: These databases are used while you configure the Identity Applications and Identity Reporting containers.

8. Grant all the privileges on the databases for the `idmadmin` user:

```
docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE idmuserappdb TO idmadmin"
```

```
docker exec -it postgresql-container psql -U postgres -c "GRANT ALL
PRIVILEGES ON DATABASE igaworkflowdb TO idmadmin"
```

9. To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it postgresql-container bash
```

Deploying Identity Applications Container

NOTE: Before you deploy the Identity Applications container, ensure that you generate the required certificate. For more information, see Generating Certificate With Identity Vault Certificate Authority.

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Ensure that the `UA_SERVER_SSL_PORT` property is set to a unique port.
3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

```
docker load --input IDM_486_identityapplication.tar.gz
```

6. Deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=idapps-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 identityapplication:idm-4.8.6
```

7. To verify whether the container was successfully deployed, check the log files by running the following command:

```
tail -f /data/userapp/log/idmconfigure.log
```

8. Run the following command to log in to the container.

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it idapps-container bash
```

9 Run the following command:

NOTE: Before performing this step, ensure that the container is deployed successfully.

```
/opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
tomcat.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/apps/
tomcat/conf/idm.jks -deststorepass <password>
```

10 Run the following command to exit the container.

```
exit
```

11 Run the following command to modify the Tomcat shutdown port in the `server.xml` file. In the following example, the port 8005 will be changed to 28005:

```
sed -i "s~8005~28005~g" /data/userapp/tomcat/conf/server.xml
```

12 Restart the container using the following command:

```
docker restart idapps-container
```

NOTE: To modify any settings in the configuration update utility, launch `configupdate.sh` from the `/opt/netiq/idm/apps/configupdate/` directory of the Identity Applications container. The configuration update utility can be launched in console mode only.

Deploying Form Renderer Container

1 Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.

2 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

3 Navigate to the `docker-images` directory.

4 Run the following command to load the image:

```
docker load --input IDM_486_formrenderer.tar.gz
```

5 Deploy the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=fr-
container -v /data:/config -e SILENT_INSTALL_FILE=/config/
silent.properties --stop-timeout 100 formrenderer:idm-4.8.6
```

6 To log in to the container, run the following command:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it fr-container bash
```

Deploying ActiveMQ Container

NOTE: This procedure assumes that you will use the ActiveMQ container with the Identity Applications container. To use the ActiveMQ container with the Fanout Agent container, you must deploy a new instance of the ActiveMQ container with different IP address and ports.

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_activemq.tar.gz
   ```

4. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=amq-container -v /data:/config --env-file /data/silent.properties --stop-timeout 100 activemq:idm-4.8.6
   ```

5. To log in to the container, run the following command:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it amq-container bash
   ```

Deploying Identity Reporting Container

NOTE: Before you deploy the Identity Reporting container, ensure that you generate the required certificate. For more information, see Generating Certificate With Identity Vault Certificate Authority.

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Ensure that the `TOMCAT_HTTPS_PORT` property is set to a unique port.
3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

   ```
   docker load --input IDM_486_identityreporting.tar.gz
   ```

6. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=rpt-container -v /data:/config -e SILENT_INSTALL_FILE=/config/silent.properties --stop-timeout 100 identityreporting:idm-4.8.6
   ```

7. To verify whether the container was successfully deployed, check the log files by running the following command:

```
tail -f /data/reporting/log/idmconfigure.log
```

8. Run the following command to log in to the container:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it rpt-container bash
   ```

9. Run the following command:

   NOTE: Before performing this step, ensure that the container is deployed successfully.

   ```
   /opt/netiq/common/jre/bin/keytool -importkeystore -srckeystore /config/
   tomcat.ks -srcstorepass <password> -destkeystore /opt/netiq/idm/apps/
   tomcat/conf/idm.jks -deststorepass <password>
   ```

10. Run the following command to exit the container.

    ```
    exit
    ```

11. Run the following command to modify the Tomcat shutdown port in the `server.xml` file. In the following example, the port 8005 will be changed to 38005:

    ```
    sed -i "s~8005~38005~g" /data/reporting/tomcat/conf/server.xml
    ```

12. *(Conditional) Applies only if you are using Identity Vault as the Certificate Authority.*

    Add the `-Dcom.sun.net.ssl.checkRevocation=false` parameter in the `export CATALINA_OPTS` entry of the `setenv.sh` file. In this example, the `setenv.sh` file is located under the `/data/reporting/tomcat/bin/` directory.

13. Restart the container using the following command:

    ```
    docker restart rpt-container
    ```

Deploying SSPR Container

Perform the following tasks to deploy the SSPR container:

1. Use the silent properties file generated in the Creating the Silent Properties File section for deploying the container.
2. Create a sub-directory under the shared directory `/data`, for example, `sspr`.

   ```
   mkdir sspr
   ```

3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

   ```
   docker load --input IDM_486_sspr.tar.gz
   ```

6. Deploy the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=sspr-
   container -v /data/sspr:/config --stop-timeout 100 sspr/sspr-
   webapp:4.6.0.0
   ```

7 Run the following command from the Docker host to copy the `silent.properties` file from the Docker host to SSPR container:

```
docker cp /data/silent.properties sspr-container:/tmp
```

8 Load the silent properties file to the SSPR container.

```
docker exec -it sspr-container /app/command.sh ImportPropertyConfig /
tmp/silent.properties
```

NOTE: Check if the `SSPRConfiguration.xml` is created under the `/config` directory of SSPR container and verify the content of the file.

9 Import the OAuth certificate to SSPR:

9a From the Docker host, edit the `SSPRConfiguration.xml` file located at `/data/sspr/` directory and set the value of the `configIsEditable` flag to **true** and save the changes.

9b Launch a browser and enter the `https://identitymanager.example.com:8443/sspr` URL.

9c Click **OK**.

9d Log in using administrator credentials, for example, `uaadmin`.

9e Click on the user, for example, `uaadmin`, on the top-right corner and then click **Configuration Editor**.

9f Specify the configuration password and click **Sign In**.

9g Click **Settings** > **Single Sign On (SSO) Client** > **OAuth** and ensure that all URLs use the HTTPS protocol and correct ports.

9h Under **OAuth Server Certificate**, click **Import from Server** to import a new certificate and then click **OK**.

9i Click 💾 at the top-right corner to save the certificate.

9j Review the changes and click **OK**.

9k After the SSPR application is restarted, edit the `SSPRConfiguration.xml` file and set the value of the `configIsEditable` flag to **false** and save the changes.

Deploying Identity Console Container

For more information on deploying Identity Console Container, see Deploying Identity Console as Docker Container.

10 Updating Identity Manager Containers

This section provides information on updating individual containers of Identity Manager.

The procedures for updating containers are described in subsequent sections.

- "Prerequisites for Updating Containers" on page 85
- "Updating Containers on Distributed Servers" on page 85
- "Updating Containers on a Single Server" on page 92

Prerequisites for Updating Containers

Perform the following steps before you update each of the Identity Manager containers.

IMPORTANT: This section does not apply for the PostgreSQL container. For information about updating the PostgreSQL container, see Updating PostgreSQL Container in the *"Updating Containers on Distributed Servers" on page 85* section or Updating PostgreSQL Container in the *"Updating Containers on a Single Server" on page 92* section.

1 (Conditional) Copy the required dependent files to the mount directory. For more information, see "Handling RPM Updates and Third Party Files" on page 50.

2 Stop all the Identity Manager containers.

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

For example,

```
docker stop engine-container
```

3 Take a back up of the shared directory. The examples in the guide assumes `/data` as the shared directory.

4 Delete all the Identity Manager containers.

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

For example,

```
docker rm engine-container
```

5 (Conditional) Delete all obsolete Docker images.

```
docker rmi <image ID>
```

Updating Containers on Distributed Servers

The containers must be updated in the following order:

- "Updating Identity Manager Engine Container" on page 86
- "Updating Remote Loader Container" on page 87

Updating Identity Manager Engine Container

1 Create a `credentials.properties` file under the shared directory `/data` with the following content.

```
ID_VAULT_ADMIN_LDAP="<ID_VAULT_ADMIN_LDAP>"
ID_VAULT_PASSWORD="<ID_VAULT_PASSWORD>"
```

where, `ID_VAULT_ADMIN_LDAP` must be in comma format.

For example,

```
ID_VAULT_ADMIN_LDAP="cn=admin,ou=sa,o=system"

ID_VAULT_PASSWORD="novell"
```

2 (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.

3 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz` file.

4 Navigate to the `docker-images` directory.

5 Run the following command to load the image:

```
docker load --input IDM_486_identityengine.tar.gz
```

6 Update the container using the following command if you are deploying the Identity Manager Engine using the overlay network:

```
docker run --restart unless-stopped -d --ip=192.168.0.12 --
network=idmoverlaynetwork --hostname=identityengine.example.com --
name=engine-container -v /etc/hosts:/etc/hosts -v /data:/config -p
8028:8028 -p 524:524 -p 389:389 -p 8030:8030 -p 636:636 -e
SILENT_INSTALL_FILE=/config/credentials.properties --stop-timeout 100
identityengine:idm-4.8.6
```

Update the container using the following command if you are deploying the Identity Manager Engine using the host network:

```
docker run --restart unless-stopped -d --network=host --name=engine-
container -v /etc/hosts:/etc/hosts -v /data:/config -e
SILENT_INSTALL_FILE=/config/credentials.properties --stop-timeout 100
identityengine:idm-4.8.6
```

Updating Remote Loader Container

1. (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.
2. (Conditional) To start Remote Loader instances automatically with the container, perform the steps mentioned in Starting Remote Loader Instances Automatically With Remote Loader Container Deployment.
3. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
4. Navigate to the `docker-images` directory.
5. Run the following command to load the image:

   ```
   docker load --input IDM_486_remoteloader.tar.gz
   ```

6. Deploy the container by running the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.2 --
   network=idmoverlaynetwork --hostname=remoteloader.example.com -p
   8090:8090 --name=rl-container -v /etc/hosts:/etc/hosts -v /data:/config
   --stop-timeout 100 remoteloader:idm-4.8.6
   ```

   The driver files can be found at the `/opt/novell/eDirectory/lib/dirxml/classes/` directory of the container.
7. (Conditional) If the Remote Loader instances are not running, start the Remote Loader instances.

Updating Fanout Agent Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_fanoutagent.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.3 --
   network=idmoverlaynetwork --hostname=fanoutagent.example.com --
   name=foa-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-
   timeout 100 fanoutagent:idm-4.8.6
   ```

5. Start Fanout Agent.

Updating iManager Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input iManager_326_P2.tar.gz
```

4 Ensure that the `iManager.env` file is created and present in the `/data` directory.

```
# Certificate Public Key Algorithm
# Allowed Values: RSA, ECDSA256, ECDSA384
CERTIFICATE_ALGORITHM=RSA
# Cipher Suite
# Allowed Values:
# For RSA - NONE, LOW, MEDIUM HIGH
# For ECDSA256 - SUITEB128ONLY
# For ECDSA384 - SUITEB128, SUITEB192
CIPHER_SUITE=NONE
# Tomcat Server HTTP Port
TOMCAT_HTTP_PORT=8080
# Tomcat Server SSL Port
TOMCAT_SSL_PORT=8743
# iManager Authorized User (admin_name.container_name.tree_name)
AUTHORIZED_USER=
```

5 Update the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.4 --name=iman-
container --network=idmoverlaynetwork --hostname=imanager.example.com -
v /etc/hosts:/etc/hosts -v /data:/config -v /data/iManager.env:/etc/
opt/novell/iManager/conf/iManager.env -p 8743:8743 --stop-timeout 100
imanager:3.2.6-p2
```

6 (Conditional) If you have already installed Identity Manager, run the following command to check whether the plug-ins are loaded.

```
docker log <container name>
```

For example,

```
docker log <iman-container>
```

7 To install the Identity Manager plug-ins, perform the following steps:

1. Download the `Identity_Manager_4.8.6_Linux.iso`.
2. Mount the downloaded `.iso`.
3. From the mounted location, navigate to the `/iManager/plugins` directory and obtain the required plug-ins.

Alternatively, you can install the plug-ins from the iManager plug-ins website.

8 Restart the iManager container.

```
docker restart iman-container
```

Updating OSP Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_osp.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.5 --
   network=idmoverlaynetwork --hostname=osp.example.com -p 8543:8543 --
   name=osp-container -v /etc/hosts:/etc/hosts -v /data:/config --stop-
   timeout 100 osp:idm-4.8.6
   ```

5. Run the following command to log in to the container:

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it osp-container bash
   ```

6. Navigate to the `/opt/netiq/idm/apps/configupdate/` directory.
7. Modify the `configupdate.sh.properties` file.
8. Set the value of the **no_nam_oauth** parameter to *false*.
9. Save the `configupdate.sh.properties` file.
10. Run the following command to exit the container.

    ```
    exit
    ```

Updating PostgreSQL Container

NOTE: Before you update the PostgreSQL container, ensure that you stop the dependent containers such as Identity Applications and/or Identity Reporting.

1. On the Docker host, navigate to any location. For example:

   ```
   cd /tmp
   ```

2. Run the following command to take a back up of the existing PostgreSQL container data.

   ```
   docker exec postgresql-container pg_dumpall -U postgres > dump.sql
   ```

3. Stop the PostgreSQL container.

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

   For example,

   ```
   docker stop postgresql-container
   ```

4. Delete the PostgreSQL container.

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

5 Delete the existing PostgreSQL data directory.

```
rm -rf /data/postgres
```

6 (Conditional) Delete the PostgreSQL Docker image.

```
docker rmi <image ID>
```

7 Create a sub-directory under the shared directory `/data`, for example, `postgres`.

```
mkdir postgres
```

8 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

9 Navigate to the `docker-images` directory.

10 Run the following command to load the image:

```
docker load --input IDM_486_postgres.tar.gz
```

11 Update the container using the following command:

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=<password>
-v /data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v
/data:/config --stop-timeout 100 postgres:12.12
```

For example,

```
docker run --restart unless-stopped -d --ip=192.168.0.6 --
network=idmoverlaynetwork --hostname=postgresql.example.com --
name=postgresql-container -p 5432:5432 -e POSTGRES_PASSWORD=novell -v /
data/postgres:/var/lib/postgresql/data -v /etc/hosts:/etc/hosts -v /
data:/config --stop-timeout 100 postgres:12.12
```

12 Copy the data file you backed up on the Docker host (Step 2) to the new PostgreSQL `data` directory.

```
cp /tmp/dump.sql /data/postgres
```

13 Run the following command to log in to the container:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it postgresql-container bash
```

14 Navigate to the `/var/lib/postgresql/data/` directory.

15 Restore the data backed up in Step 2 to the new PostgreSQL container.

```
psql -U postgres < dump.sql
```

16 Run the following command to exit the container.

```
exit
```

Updating Identity Applications Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_identityapplication.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.7 --
   network=idmoverlaynetwork --hostname=identityapps.example.com -p
   18543:18543 --name=idapps-container -v /etc/hosts:/etc/hosts -v /data:/
   config --stop-timeout 100 identityapplication:idm-4.8.6
   ```

Updating Form Renderer Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_formrenderer.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.8 --
   network=idmoverlaynetwork --hostname=formrenderer.example.com -p
   8600:8600 --name=fr-container -v /etc/hosts:/etc/hosts -v /data:/config
   --stop-timeout 100 formrenderer:idm-4.8.6
   ```

Updating ActiveMQ Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_activemq.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.9 --
   network=idmoverlaynetwork --hostname=activemq.example.com -p
   61616:61616 --name=amq-container -v /etc/hosts:/etc/hosts -v /data:/
   config --stop-timeout 100 activemq:idm-4.8.6
   ```

Updating Identity Reporting Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_identityreporting.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.10 --
   network=idmoverlaynetwork --hostname=identityreporting.example.com -p
   28543:28543 --name=rpt-container -v /etc/hosts:/etc/hosts -v /data:/
   config --stop-timeout 100 identityreporting:idm-4.8.6
   ```

Updating SSPR Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_sspr.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --ip=192.168.0.11 --
   network=idmoverlaynetwork --hostname=sspr.example.com --name=sspr-
   container -v /etc/hosts:/etc/hosts -v /data/sspr:/config -p 8443:8443 -
   -stop-timeout 100 sspr/sspr-webapp:4.6.0.0
   ```

Updating Identity Console Container

For more information on updating Identity Console Container, see Upgrading Identity Console As Docker Container.

Updating Containers on a Single Server

The containers must be updated in the following order:

- "Updating Identity Manager Engine Container" on page 93
- "Updating Remote Loader Container" on page 93
- "Updating Fanout Agent Container" on page 94
- "Updating iManager Container" on page 94
- "Updating OSP Container" on page 95
- "Updating PostgreSQL Container" on page 96
- "Updating Identity Applications Container" on page 97

Updating Identity Manager Engine Container

1 Create a `credentials.properties` file under the shared directory `/data` with the following content.

```
ID_VAULT_ADMIN_LDAP="<ID_VAULT_ADMIN_LDAP>"
ID_VAULT_PASSWORD="<ID_VAULT_PASSWORD>"
```

where, `ID_VAULT_ADMIN_LDAP` must be in comma format.

For example,

```
ID_VAULT_ADMIN_LDAP="cn=admin,ou=sa,o=system"


ID_VAULT_PASSWORD="novell"
```

2 (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.

3 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

4 Navigate to the `docker-images` directory.

5 Run the following command to load the image:

```
docker load --input IDM_486_identityengine.tar.gz
```

6 Update the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=engine-
container -v /etc/hosts:/etc/hosts -v /data:/config -e
SILENT_INSTALL_FILE=/config/credentials.properties --stop-timeout 100
identityengine:idm-4.8.6
```

Updating Remote Loader Container

1 (Conditional) To handle any driver RPM updates or third-party files, perform the steps mentioned in Handling RPM Updates and Third Party Files.

2 (Conditional) To start Remote Loader instances automatically with the container, perform the steps mentioned in Starting Remote Loader Instances Automatically With Remote Loader Container Deployment.

3 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file.`

4 Navigate to the `docker-images` directory.

5 Run the following command to load the image:

```
docker load --input IDM_486_remoteloader.tar.gz
```

6 Update the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=<container
name> -v /data:/config --stop-timeout 100 remoteloader:idm-4.8.6
```

For example:

```
docker run --restart unless-stopped -d --network=host --name=rl-
container -v /data:/config --stop-timeout 100 remoteloader:idm-4.8.6
```

The driver files can be found at the `/opt/novell/eDirectory/lib/dirxml/classes/` directory of the container.

7 (Conditional) If the Remote Loader instances are not running, start the Remote Loader instances.

Updating Fanout Agent Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input IDM_486_fanoutagent.tar.gz
```

4 Update the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=foa-
container -v /data:/config --stop-timeout 100 fanoutagent:idm-4.8.6
```

5 Start Fanout Agent.

Updating iManager Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input iManager_326_P2.tar.gz
```

4 Ensure that the `iManager.env` file is created and present in the `/data` directory.

```
# Certificate Public Key Algorithm
# Allowed Values: RSA, ECDSA256, ECDSA384
CERTIFICATE_ALGORITHM=RSA
# Cipher Suite
# Allowed Values:
# For RSA - NONE, LOW, MEDIUM HIGH
# For ECDSA256 - SUITEB128ONLY
# For ECDSA384 - SUITEB128, SUITEB192
CIPHER_SUITE=NONE
# Tomcat Server HTTP Port
TOMCAT_HTTP_PORT=8080
# Tomcat Server SSL Port
TOMCAT_SSL_PORT=8743
# iManager Authorized User (admin_name.container_name.tree_name)
AUTHORIZED_USER=
```

5 Update the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=iman-
container -v /data:/config -v /data/iManager.env:/etc/opt/novell/
iManager/conf/iManager.env --stop-timeout 100 imanager:3.2.6-p2
```

6 To install the Identity Manager plug-ins, perform the following steps:

   1. Download the `Identity_Manager_4.8.6_Linux.iso` from the Software and License (https://sld.microfocus.com/) portal.

   2. Mount the downloaded `.iso`.

   3. From the mounted location, navigate to the `/iManager/plugins` directory and obtain the required plug-ins.

   Alternatively, you can install the plug-ins from the iManager plug-ins website.

7 Restart the iManager container.

```
docker restart iman-container
```

Updating OSP Container

1 Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.

2 Navigate to the `docker-images` directory.

3 Run the following command to load the image:

```
docker load --input IDM_486_osp.tar.gz
```

4 Update the container using the following command:

```
docker run --restart unless-stopped -d --network=host --name=osp-
container -v /data:/config --stop-timeout 100 osp:idm-4.8.6
```

5 Run the following command to log in to the container:

```
docker exec -it <container> <command>
```

For example,

```
docker exec -it osp-container bash
```

6. Navigate to the `/opt/netiq/idm/apps/configupdate/` directory.
7. Modify the `configupdate.sh.properties` file.
8. Set the value of the **no_nam_oauth** parameter to *false*.
9. Save the `configupdate.sh.properties` file.
10. Run the following command to exit the container.

    ```
    exit
    ```

Updating PostgreSQL Container

NOTE: Before you update the PostgreSQL container, ensure that you stop the dependent containers such as Identity Applications and/or Identity Reporting.

1. On the Docker host, navigate to any location. For example:

    ```
    cd /tmp
    ```

2. Run the following command to take a back up of the existing PostgreSQL container data.

    ```
    docker exec postgresql-container pg_dumpall -U postgres > dump.sql
    ```

3. Stop the PostgreSQL container.

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

    For example,

    ```
    docker stop postgresql-container
    ```

4. Delete the PostgreSQL container.

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

5. Delete the existing PostgreSQL data directory.

    ```
    rm -rf /data/postgres
    ```

6. (Conditional) Delete the PostgreSQL Docker image.

    ```
    docker rmi <image ID>
    ```

7. Create a sub-directory under the shared directory `/data`, for example, `postgres`.

    ```
    mkdir postgres
    ```

8. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
9. Navigate to the `docker-images` directory.
10. Run the following command to load the image:

    ```
    docker load --input IDM_486_postgres.tar.gz
    ```

11. Update the container using the following command:

    ```
    docker run --restart unless-stopped -d --network=host --
    name=postgresql-container -e POSTGRES_PASSWORD=<password> -v /data/
    postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.12
    ```

    For example,

```
docker run --restart unless-stopped -d --network=host --
name=postgresql-container -e POSTGRES_PASSWORD=novell -v /data/
postgres:/var/lib/postgresql/data --stop-timeout 100 postgres:12.12
```

12. Copy the data file you backed up on the Docker host (Step 2) to the new PostgreSQL `data` directory.

    ```
    cp /tmp/dump.sql /data/postgres
    ```

13. Run the following command to log in to the container:

    ```
    docker exec -it <container> <command>
    ```

    For example,

    ```
    docker exec -it postgresql-container bash
    ```

14. Navigate to the `/var/lib/postgresql/data/` directory.
15. Restore the data backed up in Step 2 to the new PostgreSQL container.

    ```
    psql -U postgres < dump.sql
    ```

16. Run the following command to exit the container.

    ```
    exit
    ```

Updating Identity Applications Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

    ```
    docker load --input IDM_486_identityapplication.tar.gz
    ```

4. Update the container using the following command:

    ```
    docker run --restart unless-stopped -d --network=host --name=idapps-
    container -v /data:/config --stop-timeout 100 identityapplication:idm-
    4.8.6
    ```

Updating Form Renderer Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

    ```
    docker load --input IDM_486_formrenderer.tar.gz
    ```

4. Update the container using the following command:

    ```
    docker run --restart unless-stopped -d --network=host --name=fr-
    container -v /data:/config --stop-timeout 100 formrenderer:idm-4.8.6
    ```

Updating ActiveMQ Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_activemq.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=amq-
   container -v /data:/config --stop-timeout 100 activemq:idm-4.8.6
   ```

Updating Identity Reporting Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_identityreporting.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=rpt-
   container -v /data:/config --stop-timeout 100 identityreporting:idm-
   4.8.6
   ```

Updating SSPR Container

1. Navigate to the location where you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz file`.
2. Navigate to the `docker-images` directory.
3. Run the following command to load the image:

   ```
   docker load --input IDM_486_sspr.tar.gz
   ```

4. Update the container using the following command:

   ```
   docker run --restart unless-stopped -d --network=host --name=sspr-
   container -v /data/sspr:/config --stop-timeout 100 sspr/sspr-
   webapp:4.6.0.0
   ```

Updating Identity Console Container

For more information on updating Identity Console Container, see Upgrading Identity Console As Docker Container.

11 Best Practices

This section includes some tips and best practices for deploying Docker containers:

- NetIQ recommends you to set a limit on the amount of CPU used for a container. This can be achieved by using the `--cpuset-cpus` flag in the docker run command.
- To set a limit on the memory used by a container, use the `--memory` flag in the docker run command.
- To gracefully stop a container, use the `--stop-timeout` flag. NetIQ recommends you to set the value of this flag to 100. If there are any active processes running inside the container, the container waits for 100 seconds and then exits. If all the processes are killed before the time specified in the `--stop-timeout` flag, the container exits when the last process is killed.
- To redirect the default log output to customized docker logs, use the `LOGTOFOLLOW` flag with the docker run command. For example, if you want to follow the new logs for OSP, specify the `-e LOGTOFOLLOW="<list of files separated by space>"` in the docker run command. This prints the logs in the new docker logs. You can use the `docker logs -f <container-name>` command to monitor the log files. The default logs for each containers are listed in the following table.

Container	Default logs
Identity Manager Engine	`/var/opt/novell/eDirectory/log/ndsd.log`
OSP	`/opt/netiq/idm/apps/tomcat/logs/catalina.out`
Identity Applications	`/opt/netiq/idm/apps/tomcat/logs/catalina.out`
Form Renderer	`/opt/netiq/idm/apps/sites/logs/formslogger.log`
ActiveMQ	`/opt/netiq/idm/activemq/data/activemq.log`
Identity Reporting	`/opt/netiq/idm/apps/tomcat/logs/catalina.out`

- For all containers except Remote Loader and Fanout Agent, you can monitor the health of the containers. Based on your requirement, you can customize the health status using the Docker runtime health checks. For example, to check the health of the `rdxml` service, use the `--health-cmd "ps -eaf | grep -i rdxml" --health-interval 60` flag.
- If you want to back up the trace files for the deployed drivers, then you can place the trace file under `/config/idm/` or manually copy the trace file to the volumized folder.
- To set a limit on the number of processes allowed to run at any point in time, use the `--pids-limit` flag in the docker run command. It is recommended to limit the PID value to 300.

- For Identity Manager Engine container, if you want to view the environ file located at the /process directory of the /proc file system, use the --cap-add=SYS_PTRACE flag in the docker run command. By default, most of the privileges are restricted and only the required privileges are enabled. For more information, see Docker documentation.
- It is recommended to map individual data volume for each component.
- Ensure that the third party jar files are volume mounted so that they are available when the container is started every time. For example, if the ojdbc.jar is present in the /opt/netiq/idm/apps/tomcat/lib directory of the container, then you must volume mount the jar file using the following command:

  ```
  -v /host/ojdbc.jar:/opt/netiq/idm/apps/tomcat/lib/ojdbc.jar
  ```
- Once the containers are deployed, it is recommended that you remove all the input files that were used for bringing up containers. This includes files such as the silent.properties, credentials.properties, and StartupRL.txt.

For example, run the following sample command containing all the above arguments for deploying containers:

```
docker run -d --name=<assign a name to the container> --network=<> --cap-add=SYS_PTRACE --pids-limit <tune container pids limit> --memory=<maximum amout of memory container can use> --restart=on-failure:5 --cpuset-cpus=<CPUs in which to allow execution> --network=<connect a container to network> --stop-timeout 100 -e LOGTOFOLLOW "/opt/netiq/idm/apps/tomcat/logs/catalina.out /opt/netiq/idm/apps/tomcat/logs/idapps.out" --health-cmd "ps -eaf | grep -i tomcat" --health-interval 60 -v <bind mount a volume> <image name>
```

12 Troubleshooting

This section provides useful information for troubleshooting problems with the Identity Manager containers.

Identity Applications Container Displays Portlet Registration Exception

Issue: While deploying Identity Applications container, it displays the following exception:

```
ERROR
[com.novell.afw.portlet.consumer.core.EboPortletProducerChangeListener]
(main) [RBPM] Portlet registration with portletID: 'HeaderPortlet' does not
exist.
com.novell.afw.portlet.exception.EboPortletRegistrationException: Portlet
registration with portletID: 'HeaderPortlet' does not exist.
```

Workaround: Restart the Identity Applications container.

Forms Are Not Loaded When Requesting For a Permission

Issue: After deploying the Identity Applications container, when you try to request for a permission that is associated with new forms, the form does not load as expected. This issue has been randomly observed.

Workaround: Ensure that the Form Renderer server and port details are specified in the `nginx.conf` file. To update the `nginx.conf` file, perform the following steps:

1. Log in to the Form Renderer container.

   ```
   docker exec -it <container> <command>
   ```

   For example,

   ```
   docker exec -it fr-container bash
   ```

2. Navigate to the `/opt/netiq/common/nginx/` directory.
3. Edit the `nginx.conf` file.
4. Specify the Form Renderer server and port details. For example:

   ```
   server {
   listen 8600 ssl;
   server_name  formrenderer.example.com;
   ```

Unable to Log In to iManager After Updating iManager Container

Issue: After updating the iManager container, the iManager user interface cannot be accessed. This issue has been randomly observed.

Workaround: To workaround this issue, perform the following steps:

1. Log in to the iManager container as a root user.

   ```
   docker exec -it -u root <container> <command>
   ```

   For example,

   ```
   docker exec -it -u root iman-container bash
   ```

2. Navigate to the `/var/opt/novell/tomcat9/work/` directory.
3. Assign the `novlwww` permissions on the directory.

   ```
   chown -R novlwww:novlwww
   ```

4. Run the following command to exit the container:

   ```
   exit
   ```

5. Restart the iManager container.

   ```
   docker restart <container>
   ```

   For example,

   ```
   docker restart iman-container
   ```

Docker Installation of the Second Identity Manager Engine Node Fails with EBA Configuration Failed

Issue: While installing Identity Manager 4.8.5 containers on SLES15SP3. Consider there are two Docker hosts available. The installation on the first Docker host that creates the eDirectory tree is successful. The installation on the second Docker host fails with the error message "`EBA Configuration Failed`" in the `idmconfigure.log`.

Workaround: From Identity Manager 4.8.6 onwards; we can override the IPs@NCP for each instance. User can override the value passed for the -B option with variable IPs with NCP override like below within the silent properties:

```
IPswithNCPoverride=<IP1>@<NCP Port>, <IP2>@<NCP Port>
```

III

Deploying Identity Manager Containers Using Ansible

This release of Identity Manager introduces support for the deployment of Identity Manager containers using Ansible. Through the Ansible approach, the containers can be easily deployed through an automated process. The deployment process is simpler and time-efficient. Identity Manager ships Ansible playbook for automating the container deployment.

NOTE: This release only supports a fresh deployment of containers using Ansible.

This section provides instructions on deploying containers through Ansible.

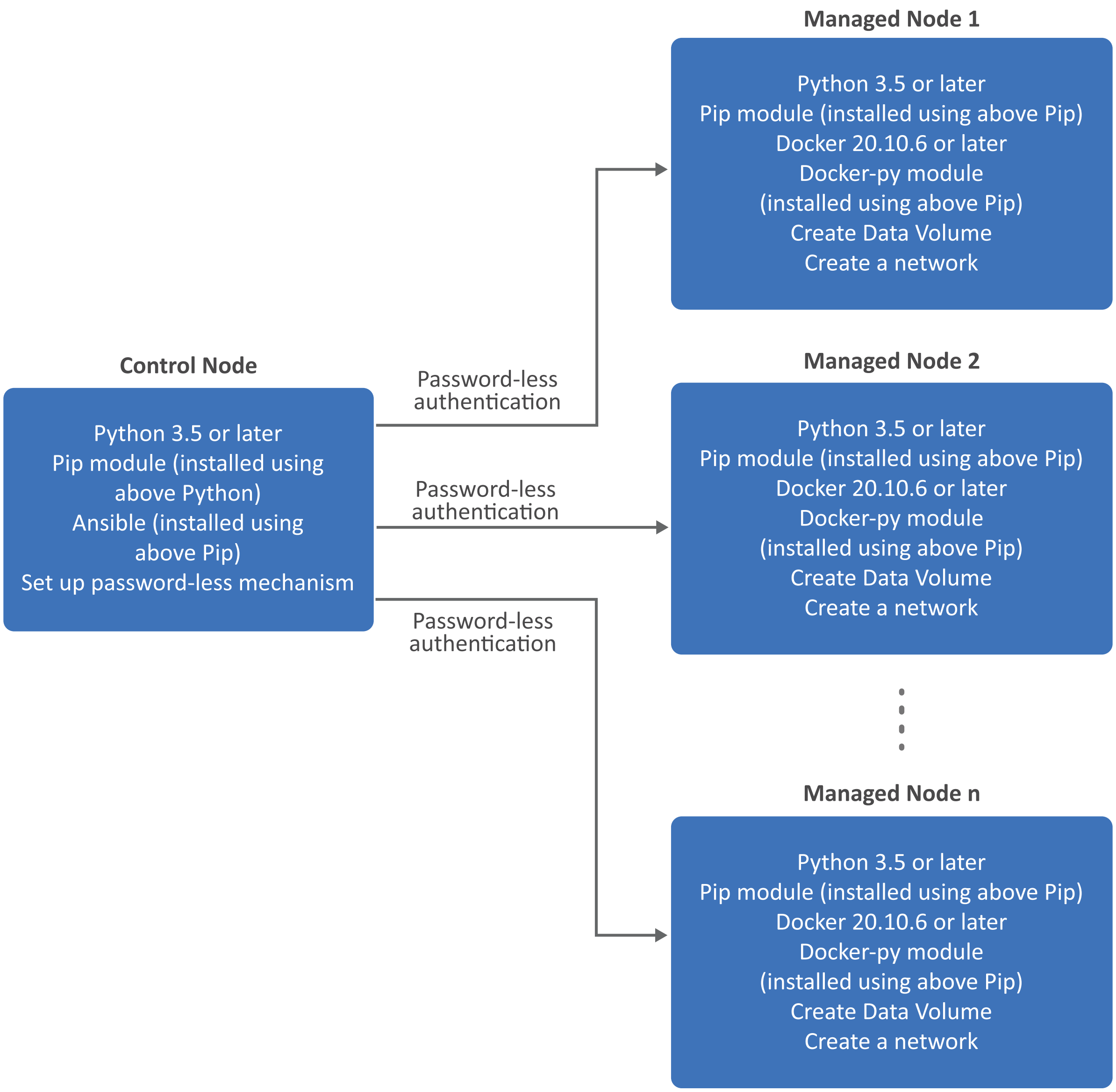
13 Planning Your Deployment

The containers deployment requires some planning and prerequisites to be followed. This section provides details on planning your deployment.

Identify two or more servers for Ansible-based container deployment. One of the servers is called Ansible Control Node (control node) and the remaining servers are called Managed Nodes (managed nodes). For more details on control node and managed nodes, see Ansible documentation.

Preparing your Ansible Nodes

You must ensure that the Ansible nodes are set up appropriately before you begin with the deployment process. The prerequisites on the control and managed nodes are different from each other. The following figure provides a high-level view on how you must prepare your control and managed nodes.


Managed Node 1
Python 3.5 or later
Pip module (installed using above Pip)
Docker 20.10.6 or later
Docker-py module
(installed using above Pip)
Create Data Volume
Create a network
Control Node
Python 3.5 or later
Pip module (installed using above Python)
Ansible (installed using above Pip)
Set up password-less mechanism
Password-less authentication
Password-less authentication
Password-less authentication
Managed Node 2
Python 3.5 or later
Pip module (installed using above Pip)
Docker 20.10.6 or later
Docker-py module
(installed using above Pip)
Create Data Volume
Create a network
Managed Node n
Python 3.5 or later
Pip module (installed using above Pip)
Docker 20.10.6 or later
Docker-py module
(installed using above Pip)
Create Data Volume
Create a network

Preparing Your Control Node

Ensure that you perform the following tasks on the control node:

- Ensure Python3 or later is installed. To check for the Python version, navigate to the `/usr/bin/` directory and run the following command:

  For example:

  ```
  python3 --version
  ```

  For more information, see Python documentation.

- Ensure pip is installed. To check for the pip version, run the following command:

  For example:

  ```
  pip --version
  ```

Ensure that `pip` has been installed through the Python3 or later version that you installed earlier.

For more information, see Python documentation.

- Install Docker module using pip, run the following command:

  For example:

  ```
  pip install docker
  ```

  Ensure that the Docker version is 20.10.6 or later. For more information, see Docker documentation.

- Install Ansible using the pip that you installed earlier. Ensure that you install Ansible version 2.10.5 or later.

  For example:

  ```
  pip install ansible
  ```

  For more information, see Ansible documentation.

- Ensure that the managed nodes are reachable from the control node. For example, you can use `ping` or any relevant mechanisms to ensure the nodes are reachable.

- Ensure that you establish a password-less authentication between the control node and all the managed nodes in your deployment. Perform the following steps:

  1. Generate a SSH key.

     For example:

     ```
     ssh-keygen
     ```

  2. Do not enter any password and proceed with the key generation.

  3. Run the following command to enable password-less authentication to the managed node:

     ```
     ssh-copy-id root@<FQDN or IP Address of the managed node>
     ```

     For example:

     ```
     ssh-copy-id root@192.168.0.25
     ```

  4. Specify the password of the managed node.

     For example, `password`.

  5. Test the connection to the managed node:

     ```
     ssh 'root@<FQDN or IP Address of the managed node>'
     ```

     For example:

     ```
     ssh 'root@<192.168.0.25>'
     ```

Preparing Your Managed Nodes

Ensure that you perform the following tasks on all the managed nodes:

- Ensure Python3 or later is installed. To check for the Python version, navigate to the `/usr/bin/` directory and run the following command:

  For example:

  ```
  python3 --version
  ```

For more information, see Python documentation.

- Ensure pip is installed. To check for the pip version, run the following command:

  For example:

  ```
  pip --version
  ```

  Ensure that pip has been installed through the Python3 or later version that you installed earlier.

  For more information, see Python documentation.

- Install Docker. Ensure that the Docker version is 20.10.6 or later. For more information, see Docker documentation.
- Install Docker python module using pip:

  For example:

  ```
  pip install docker-py
  ```

- Create a shared directory. For more information, see "Managing Container Volume Data" on page 49.
- Create a network for establishing communication between containers. For example, to create an overlay network, see "Setting Up an Overlay Network" on page 58.

Creating the setup.csv File

The setup.csv file is an input file that will be used by Ansible while deploying containers. Identity Manager bundles a default template of the setup.csv file in the Identity Manager container tar file.

The default template of the setup.csv file is located at the /<location where you extracted the container tar file>/ansible/input/ directory. You can edit the setup.csv file as per your requirement.

The parameters that the setup.csv file contains and the purpose of each parameters are described in the following section:

- **Component:** Indicates the container that you want to deploy. For example, engine.
- **Deploy:** Indicates whether you want to deploy the container. The supported values are **yes** and **no**.
- **DockerHost:** Indicates the Docker host where the container will be deployed. In other words, this can be any of the managed nodes you have identified for your deployment. For example, DockerHostA
- **IP Address:** Indicates the IP Address of the Docker host where the container will be deployed. For example, 192.168.0.15
- **ContainerName:** Indicates the name of the container. For example, engine-container.
- **ContainerHostname:** Indicates the host name of the Docker hosts or server where the container will be deployed. NetIQ recommends that you specify the hostname in the FQDN format. For example, identityengine.example.com.
- **ExposedPorts:** Indicates the ports that you want to expose for the container to listen on. For example, 636.

NOTE: Ensure that you expose unique ports for each containers and specify the same ports that you provided while creating the `silent.properties` file. For example, you can plan for the ports that you want to expose by referring to the sample ports provided in Table 7-2.

- **FileMounting:** Indicates the path for any custom files such as `ojdbc.jar`. For example, `/opt/novell/eDirectory/lib/dirxml/classes/ojdbc.jar`.

NOTE:
- If there are multiple values, specify them as a space-separated variable list. For example, `/opt/novell/eDirectory/lib/dirxml/classes/ojdbc.jar /opt/novell/eDirectory/lib/dirxml/classes/mssql.jar`
- (Conditional) This applies only when you have set the value for the Core DNS container as **no** in the `Deploy` column.

  Ensure that the `hosts` file is mapped in the FileMounting field. For example, `/etc/hosts`.

- **SharedVolume:** Indicates the shared directory that you want the containers to use for data persistence. For example, `/data`. It is necessary to create this directory across the docker hosts.

14 Deploying Containers

Perform the following steps to deploy containers:

1 On the control node, perform the following steps:

1a Download and extract the Identity Manager container tar file. For more information, see "Obtaining the Docker Images" on page 49.

1b Navigate to the `/<location where you extracted the tar file>/docker-images/` directory.

1c Copy the `IDM_486_idm_conf_generator.tar.gz` file and place the file on any of the managed nodes.

2 On any of the managed nodes, perform the following steps:

2a Place the `IDM_486_idm_conf_generator.tar.gz` file you copied in Step 1c in any location. For example, `/home`.

2b Create the `silent.properties` file. For more information, see "Creating the Silent Properties File" on page 56.

3 On the control node, perform the following steps:

3a Navigate to the `/<location where you extracted the tar file>/ansible/input/` directory and place the following files:

- `silent.properties` file that you created in Step 2b
- `iManager.env` file. For more information on creating the `iManager.env` file, see Step 4 in the "Deploying iManager Container" on page 61 section.
- `setup.csv` file that you created in the "Creating the setup.csv File" on page 108 section
- any custom certificates that you obtained from an external certificate authority

NOTE: If you are using Identity Vault as the certificate authority for generating certificates, perform the steps mentioned in "Generating Certificate With Identity Vault Certificate Authority" on page 77.

- any custom files such as `ojdbc.jar` or custom LDIF files

NOTE: Ensure that the destination path for these files are specified in the **FileMounting** column of the `setup.csv` file. For more information, see "Creating the setup.csv File" on page 108.

3b Navigate to the `/<location where you extracted the tar file>/ansible/` directory.

3c (Optional) This step applies for advanced users. Review the `ansible.cfg` file for your deployment.

3d Run the following command for deploying the `setup.yml` playbook:

```
ansible-playbook setup.yml
```

3e (Optional) This step applies for advanced users. Review the `idminventory.ini` file for your deployment.

3f Run the following command for deploying the `deploy.yml` playbook:

```
ansible-playbook deploy.yml -e 'network_set=<Docker network name>'
```

For example:

```
ansible-playbook deploy.yml -e 'network_set=idmoverlaynetwork'
```

15 Post-deployment Tasks

After completing the deployment of Identity Manager containers, you must perform certain tasks to ensure the Identity Manager solution works properly in your environment.

You must perform the following post-deployment tasks:

- (Conditional) This step applies only when you have set the value for the Core DNS container as **no** in the `setup.csv` file and want to log in to iManager user interface by specifying the hostname of the Identity Manager Engine container in the **Tree** field.

  1. Log in to the iManager container.

     ```
     docker exec -it -u root <container> <command>
     ```

     For example,

     ```
     docker exec -it -u root iman-container bash
     ```

  2. Navigate to the `/etc/` directory.
  3. Edit the `hosts` file.
  4. Add the entries of all the containers running on that Docker host.

     NOTE: Ensure that the hostname for all containers are in Fully Qualified Domain Name (FQDN) format only.

     The entries must follow the below format:

     ```
     <IP of the container> <FQDN> <short_name>
     ```

     For example,

     ```
     192.168.0.7   identityapps.example.com    identityapps
     ```

  5. Save the `hosts` file.

- Install the latest iManager plug-ins. For more information, see Step 7 of the Deploying iManager Container section.
- Set the value of the **no_nam_auth** parameter to `False`. For more information see, Step 7 to Step 11 of the Deploying OSP Container section.
- Import the OAuth certificate to SSPR. For more information, see Step 9 of the Deploying SSPR Container section.

16 Troubleshooting

This section provides useful information for troubleshooting problems with the Identity Manager containers that are deployed using Ansible.

Running the deploy.yml File for the First Time Displays an Exception

Issue: When you are running the `deploy.yml` for the first time in your deployment, you will see the following message indicating that the Docker images are not present on the target nodes. For example, if you are deploying the Core DNS container, you will see the following error:

```
fatal: [<ip address/DNS>]: FAILED! => {"changed": true, "cmd": "docker
images | grep coredns | grep 1.8.0", "delta": "0:00:00.914078", "end":
"msg": "non-zero return code", "rc": 1, "start": "stderr": "",
"stderr_lines": [], "stdout": "", "stdout_lines": []}
```

Workaround: There is no workaround at this time. However, you can ignore the message and proceed with the deployment. This does not cause any loss in functionality.

Exception Reported When the IP Address Is Already In Use in Your Network

Issue: The container deployment fails when the IP address is already in use by a different container across your network. The following exception is reported on the console.

```
fatal: [<ip address/DNS>]: FAILED! => {"changed": false, "msg": "Error
starting container
b1eb07f42cf6bd63787ae6167f5e3a0f7cbee0f8be80a5764bcc7c7f9d6b96b1: 403
Client Error for http+docker://localhost/v1.40/containers/
b1eb07f42cf6bd63787ae6167f5e3a0f7cbee0f8be80a5764bcc7c7f9d6b96b1/start:
Forbidden (\"Address already in use\")"}
```

Workaround: Assign a different IP address for the container.

Unable to Fetch Tasks After Deploying Identity Applications Container

Issue: After deploying the Identity Applications container, when you log in to the Identity Manager Dashboard and navigate to the **Tasks** page, the Dashboard does not fetch the list of tasks as expected. The following error is reported in the `catalina.out` file.

```
SEVERE [main]
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource
Unable to process web resource [/WEB-INF/classes/com/microfocus/idm/nrf/
resources/NRFRsrc_fr.class] for annotations
     java.io.EOFException
     at java.io.DataInputStream.readFully(DataInputStream.java:197)
     at java.io.DataInputStream.readUTF(DataInputStream.java:609)
     at java.io.DataInputStream.readUTF(DataInputStream.java:564)
     at
org.apache.tomcat.util.bcel.classfile.ConstantUtf8.getInstance(ConstantUtf
8.java:36)
     at
org.apache.tomcat.util.bcel.classfile.Constant.readConstant(Constant.java:
79)
     at
org.apache.tomcat.util.bcel.classfile.ConstantPool.<init>(ConstantPool.jav
a:53)
     at
org.apache.tomcat.util.bcel.classfile.ClassParser.readConstantPool(ClassPa
rser.java:174)
     at
org.apache.tomcat.util.bcel.classfile.ClassParser.parse(ClassParser.java:8
3)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsStream(Context
Config.java:2351)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(Co
ntextConfig.java:2250)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(Co
ntextConfig.java:2244)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(Co
ntextConfig.java:2244)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(Co
ntextConfig.java:2244)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(Co
ntextConfig.java:2244)
     at
org.apache.catalina.startup.ContextConfig.processAnnotationsWebResource(Co
ntextConfig.java:2244)
     at
org.apache.catalina.startup.ContextConfig.processClasses(ContextConfig.jav
a:1397)
     at
org.apache.catalina.startup.ContextConfig.webConfig(ContextConfig.java:130
2)
     at
org.apache.catalina.startup.ContextConfig.configureStart(ContextConfig.jav
a:985)
```

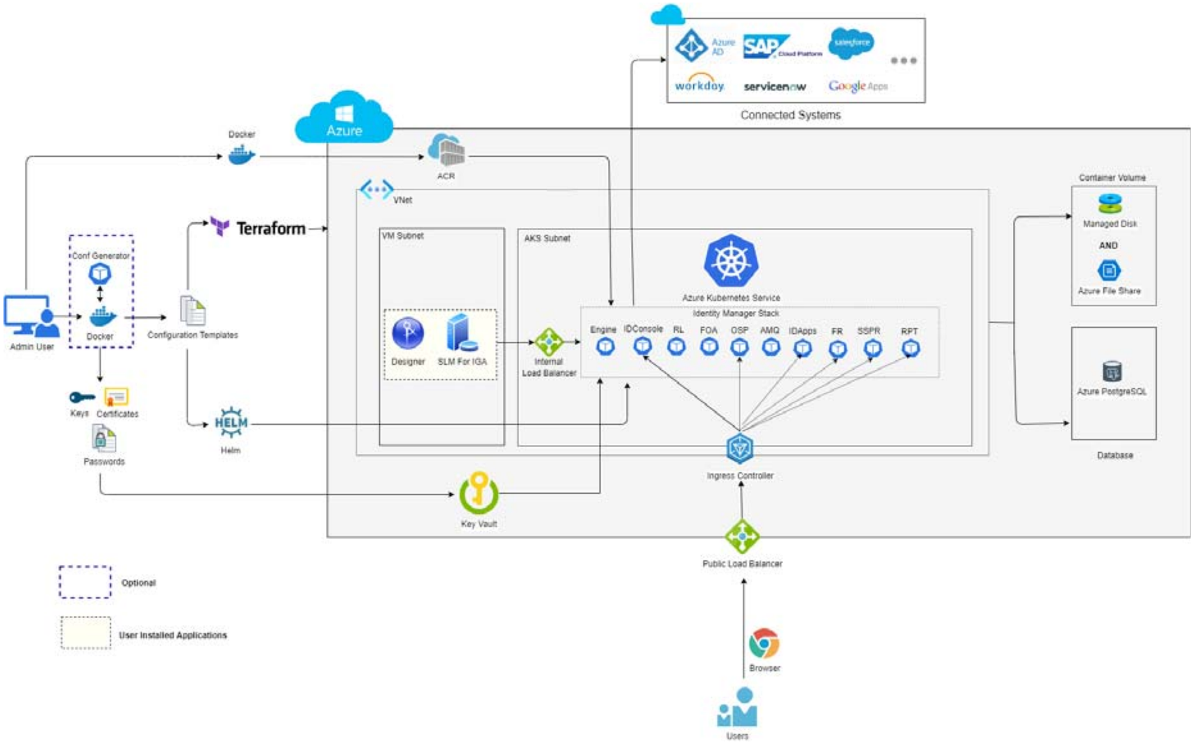
Workaround: To workaround this issue, perform the following steps:

1. Navigate to the `/opt/netiq/idm/apps/tomcat/webapps/` directory.
2. Delete the `workflow` folder.
3. (Optional) Restart Tomcat.

IV Deploying Identity Manager Containers on Microsoft Azure Kubernetes Service

This release allows you to deploy Identity Manager containers on Microsoft Azure Kubernetes Service. The deployment of Identity Manager containers is automated with the help of Terraform and Helm charts.

The following figure provides an architectural view of deploying containers on Azure.



- `ACRcreate.sh` file creates a new Azure container registry and allows users to upload/store the docker images on the Azure portal.
- The Identity Manager Configuration Generator image is used to accomplish the tasks listed below:
  - Generate Configuration files (Terraform files and Helm charts).
  - Create an Azure Resource Group.
  - Create a Key Vault under Azure Resource Group.
  - Push all the sensitive information to the Key Vault.
- Using Terraform scripts, users can set up infrastructure such as:
  - Identity manager engine docker VM.

    - Network creation.
    - Azure PostgreSQL (optional).
    - Azure Kubernetes Service (AKS).
- Using Helm charts, users can deploy the Identity Manager Containers inside the Azure Kubernetes cluster.
- NGINX Ingress Controller acts as a reverse proxy to access the identity manager web applications (Includes Identity applications, Identity Reporting, and Identity Console etc.) that are running inside the AKS cluster.
    - A single domain (For example, `identitymanager.eastus.cloudapp.azure.com`) needs to be purchased.
    - The same domain is assigned to the Nginx ingress controller.
- Internal load balancer is used for internal communications between engine docker host and Kubernetes cluster.
- Persistent storage is dynamically created and mounted in the overall infrastructure.
- Identity Applications and Reporting can be configured to use the Azure PostgreSQL database provided by Azure.
- Designer and Sentinel Log Management (SLM) for Identity Governance Administration should be manually deployed by the user after Terraform execution.

Overview

To plan the deployment of Identity Manager containers on the Azure Kubernetes cluster, see the following sections:

- Chapter 17, "Planning Your Deployment," on page 121
- Chapter 18, "Preparing for Deployment," on page 125
- Chapter 19, "Deployment Procedures," on page 127
- Chapter 20, "Post-deployment Tasks," on page 147
- Chapter 21, "Maintaining the Deployment of Identity Manager Containers," on page 149
- Chapter 22, "Troubleshooting," on page 155

17 Planning Your Deployment

This section guides you through the process of planning your deployment and provides you with the list of prerequisites required to deploy the containers on Azure.

The Identity Manager containers deployment process requires the following Azure resources:

1 Ensure your Azure account is provided with the read-write permissions to create an infrastructure.

1a Azure Kubernetes Service (AKS).

1b Network creation.

1c (Optional) Azure PostgreSQL.

NOTE: These elements are created by the Terraform deployment scripts.

2 Purchase a domain to access different applications in Identity Manager.

You can have your own domain within the respective resource group. For example, **identitymanager.eastus.cloudapp.azure.com**.

NOTE: For the above example, **eastus** is the location and **cloudapp.azure.com** is the general azure suffix domain.

3 Obtain SSL Certificates (`.crt` and `.key` files) for your domain from any CA authority.

4 (Optional) Generate your own self-signed certificate by running the following command in the command-line interface:

```
openssl req -new -newkey rsa:2048 -nodes -keyout domain.key -out
domain.csr
```

4a Specify input for the following prompts:

Table 17-1

Prompt	Description
Country Name (two letter code)	Specify the country name with a 2 letter code.
State or Province Name	Specify the full name for state or province.
Locality Name	Specify the city name.
Organization Name	Specify the organization name. For example, Internet Widgits Pty Ltd.
Organizational Unit Name	Specify the organizational unit name. For example, section.
Common Name	Specify the server Fully qualified domain name (FQDN) name. For example, `identitymanager.eastus.cloudapp.azure.com`.
Email Address	Specify your email address.
A challenge password	Specify a unique password.
An optional company name	Specify an optional company name.

4b Run the following command to convert the private key to RSA format:

```
openssl rsa -in domain.key -out tls.key
```

4c Perform the following steps to obtain the Root certificate.

4c1 Log in to `https://<IP address or Host name>:8443/nps/servlet/portal`.

4c2 Go to **Roles and Tasks** > **NetIQ Certificate Access** > **Server Certificates**, and then select the **SSL CertificateDNS** check box.

4c3 Click **Export**.

4c4 In the Certificates drop-down, select the **Organizational CA**.

4c5 In the Export format drop-down, select **BASE64**.

4c6 Click **Next**.

4c7 Click **Save the exported certificate**.

4d Perform the following steps to submit your CSR to the CA authority and get a signed server certificate.

4d1 Log in to `https://<IP address or Host name>:8443/nps/servlet/portal`.

4d2 Go to **Roles and Tasks** > **NetIQ Certificate Server** > **Issue Certificate**.

4d3 Click **Choose File**, upload the `<domain>.csr` file that was generated in Step 4b on page 122.

4d4 Click **Next** thrice.

4d5 In the validity period drop-down, select the validity period. For example, maximum.

4d6 Click **Next**.

4d7 Select the **File in Base64 format** radio button.

4d8 Click **Next**.

4d9 Review the provided inputs and then click **Finish**.

4d10 Click **Download the issued certificate**.

4e Run the following command to concatenate server certificate and root certificate.

```
cat domain.b64 <(echo) cert.b64 <(echo) > tls.crt
```

5 To generate the Service Principal Credentials, perform the following steps.

IMPORTANT: You must have service principal credentials for configuration generation and to push all the sensitive information to the Azure Key vault.

5a Log in to the Azure Portal.

5b Click >_.

5c In the terminal window, run the following command:

```
az ad sp create-for-rbac --skip-assignment
```

NOTE: Make a note of the appid, password, and the tenant ID from the command output.

18 Preparing for Deployment

This section provides instructions on the pre-deployment tasks for deploying the Identity Manager containers.

1 Download the `Identity_Manager_4.8.6_Containers.tar.gz` from the Software and License page (https://sld.microfocus.com/).

2 Run the following command to extract the `Identity_Manager_4.8.6_Containers.tar.gz` file:

```
tar -zxvf Identity_Manager_4.8.6_Containers.tar.gz
```

3 Navigate to the `Identity_Manager_4.8.6_Containers` directory, run the following command:

```
./ACRCreate.sh
```

NOTE: A minimum of 6 GB space is required on the current directory to execute the script.

3a Perform the following steps to sign in to Microsoft Azure CLI.

3a1 Log in to https://microsoft.com/devicelogin.

3a2 Enter the printed code.

3a3 Click **Next**.

3a4 Click **Continue**.

3b Specify input for the following prompts:

Prompt	Description
Enter the appropriate Azure Account id printed above as-is without double quotes.	Specify the Azure Account ID generated in the above step.

Prompt	Description
Do you want to create azure container registry?	Specify your choice to create azure container registry. ◆ (Conditional) If Yes, 1. Specify your choice to create new resource group. ◆ (Conditional) If yes, specify the Azure Resource group name. ◆ (Conditional) If no, specify the existing Azure Resource group name. **NOTE:** Only alpha numeric characters are allowed in the resource group name. 2. Enter the azure resource group location. For example, `eastus`. ◆ (Conditional) If No, enter the existing azure container registry URL. For example, `azureregname.azurecr.io`.

3c (Conditional) To obtain the Azure Container Registry details, perform the following steps:

3c1 Login to the Azure portal.

3c2 Go to your Resource Group.

3c3 Under Resources, click the **container registry**.

3c4 Go to **Settings** > **Access keys** and enable the **Admin user** toggle.

3c5 Make a note of Registry Details for further use.

19 Deployment Procedures

This section describes the deployment procedures involved in deploying the Identity Manager containers on Azure. There are two stages of deployment:

- "Infrastructure Deployment on Azure cloud" on page 127
- "Identity Manager Container Deployment on Azure Kubernetes Service" on page 141
- "Setting up of Remote Loader Service on Azure Kubernetes Service" on page 142
- "Setting up of Fanout Agent service on Azure Kubernetes Service" on page 143

Infrastructure Deployment on Azure cloud

Infrastructure deployment on Azure cloud is defined using terraform configurations. There are two different methods to deploy the infrastructure on Azure. Based on your requirement, you can choose to deploy the infrastructure either using a configuration generator or the configuration templates.

- "Deployment using the Configuration Generator" on page 127
- "Deployment using the Configuration Templates" on page 131

Deployment using the Configuration Generator

Perform the following steps to generate the configuration files using the Identity Manager Configuration Generator.

IMPORTANT: You must have a machine with docker installed and running to perform the following steps.

1. Create a shared volume. For more information, see Managing Container Volume Data.
2. Copy the certificates of your domain to the shared volume.
3. Run the following command to load the docker image:

   ```
   docker load --input docker-images/IDM_486_idm_conf_generator.tar.gz
   ```

4. Navigate to the unzipped location and run the following command to deploy the configuration generator container:

   ```
   docker run --rm -it --name=idm_conf_generator -v /data:/config
   idm_conf_generator:idm-4.8.6
   ```

5. The following table provides information on the new prompts:

   NOTE: For all the existing prompts, refer the section Understanding the Configuration Parameters.

Prompt	Description
Do you want to deploy Identity Manager Containers on Azure? For secondary server deployment, please select n and proceed with silent property generation (y/n)?	Specify your choice to deploy Identity Manager Containers. ◆ (Conditional) If Yes, proceed with the next prompt. ◆ (Conditional) If No, enter silent property file name with absolute path. For example, `/config/silent.properties`. **NOTE:** For secondary server deployment, enter **n** and proceed with Creating the Silent Properties File.
Specify the namespace for Kubernetes Deployment.	Enter the namespace for Kubernetes Deployment. For example, idm.
Do you want to create a new Azure PostgreSQL Server instance?	Specify your choice to create a new Azure PostgreSQL Server instance. ◆ (Conditional) If Yes, specify the prefix for the Azure PostgreSQL server name. For example, idmpgserver. **NOTE:** The entered prefix will be appended by a hyphen and a randomly generated 14 digit number. ◆ (Conditional) If No, proceed with next prompt.
Enter the fully qualified domain name (FQDN) for accessing the Identity Manager web applications.	Specify the FQDN to access the identity manager web applications. For example, `identitymanager.eastus.cloudapp.azure.com`. **NOTE:** Identity Manager web applications include Identity Applications, Identity Reporting, SSPR, OSP and also Identity Console.
Enter the TLS certificate file.	Specify the TLS certificate file in PEM format, which contains the subject alternate name and common name for the domain specified above. For example, `/config/tls.crt`.
Enter the private key file for the TLS certificate.	Specify the private key file for the TLS certificate. For example, `/config/tls.key`.
Enter the number of instances you want to deploy on Azure for Identity Manager Engine (pod replicas)?	Specify the number of instances (pod replicas) to deploy on Azure for Identity Manager Engine. For example, 5.
Enter the number of instances you want to deploy on Azure for OSP (pod replicas)?	Specify the number of instances (pod replicas) to deploy on Azure for OSP. For example, 5.
Enter the number of instances you want to deploy on Azure for Identity Applications (pod replicas)?	Specify the number of instances (pod replicas) to deploy on Azure for Identity Applications. For example, 5.

Prompt	Description
Enter the Identity Vault Server Name.	Specify the Identity Vault Server Name. For example, `IDVAULTSERVER`.
Enter the Azure Service Principal ID.	Specify the Azure Service Principal ID generated in Section: Planning your deployment, Step 5.
Enter the Azure Service Password.	Specify the Azure service password generated in Section: Planning your deployment, Step 5.
Enter the Tenant ID of your Service Principal.	Specify the Tenant ID generated in Section: Planning your deployment, Step 5.
Enter the existing Azure Container Registry Server Name.	Specify the Azure Container Registry Server Name. Refer to Step 3c.
Enter the Azure Container Registry user name.	Specify the Azure Container Registry user name. Refer to Step 3c.
Enter the Azure Container Registry user password.	Specify the Azure Container Registry password. Refer to Step 3c.
Enter the appropriate Azure Account ID printed above as-is without double quotes.	Specify the Azure Account ID generated in the above Step.
Enter the Azure Resource Group Name.	Specify the Azure Resource group name. For example, `idvault-rg`.
Enter the Azure Resource Group Location.	Specify the Resource Group Location. For example, `eastus`.

After answering all the prompts, Identity Manager configuration generator performs the following actions:

- If there is no resource group available in azure then a resource group is created.
- A Key vault and storage account is created under the resource group.
- All the sensitive information is pushed to the Key vault.

  NOTE: To access the sensitive information in Azure Key Vault, refer to Quickstart:Azure Key Vault.

- `IDM_4.8.6_Cloud_Deployment_files.zip` file is created including Terraform files and Helm Charts under the shared volume.

6 Log in to the Azure portal.

NOTE: Azure Cloud Shell is automatically authenticated using the initial signed-in account. If you need to use a different account, run the `az login` command and sign-in to Azure-CLI.

7 Perform the following steps to upload the `IDM_4.8.6_Cloud_Deployment_files.zip` file to the azure cloud shell.

7a Click >_.

7b In the terminal window, click ⇅.

7c Select the zip file to upload to Azure.

8 Run the following command to extract the content of the zip file:

```
unzip IDM_4.8.6_Cloud_Deployment_files.zip
```

9 Navigate to the `IDM_4.8.6_Cloud_Deployment_files` directory.

10 (Optional) Review the following files.

- terraform.tfvars
- values.yaml

11 (Optional) Run the following command to create a storage account key:

```
ACCOUNT_KEY=$(az storage account keys list --resource-group
$AZURE_RESOURCE_GROUP_NAME --account-name
$AZURE_STORAGE_ACCOUNT_FOR_TFSTATE --query '[0].value' -o tsv)
```

For example,

```
ACCOUNT_KEY=$(az storage account keys list --resource-group idvault-rg
--account-name stract10226600913781 --query '[0].value' -o tsv)
```

12 (Optional) Run the following command:

```
export ARM_ACCESS_KEY=$ACCOUNT_KEY
```

13 Run the following command to download all the required plug-ins needed for infrastructure deployment.

```
terraform init
```

14 Run the following command to plan and understand the deployment based on the input.

```
terraform plan
```

15 Run the following command to create the infrastructure as defined in the input.

```
terraform apply --auto-approve
```

NOTE: If you see any errors while running the Terraform commands, refer to troubleshooting Running the Terraform apply Command Displays an Exception.

16 (Optional) Run the following command to identify Azure account specific information such as storage account name, key vault name and database administrator details.

```
terraform output
```

Once the Terraform commands are executed successfully, perform the steps mentioned in the “Identity Manager Container Deployment on Azure Kubernetes Service” on page 141 to complete the Identity Manager container deployment on Azure.

Deployment using the Configuration Templates

Perform the following steps to update the configuration templates in your virtual machine.

1. Navigate to the directory you have extracted the `Identity_Manager_4.8.6_Containers.tar.gz` file.
2. Navigate to the `Identity_Manager_4.8.6_Containers/terraform/` directory and run the following command to extract the contents of the zip file.

   ```
   unzip IDM_4.8.6_Azure_Terraform_Configuration.zip
   ```

3. Go the `IDM_Azure_Terraform_Configuration` folder and update the following fields in `terraform.tfvars` file.

Table 19-1

Fields	Description
resource_group_name	Specify the Azure Resource group name. For example, idvault-rg.
resource_group_location	Specify the Resource Group Location. For example, eastus.
resource_group_exists	Specify your choice to know the existing resource group. For example, true.
keyvault_name	Specify a unique key vault name.
keyvault_exists	Specify your choice to know the existing key vault. For example, true.
image_registry_server	Specify the Azure Container Registry Server Name.
image_registry_server_username	Specify the Azure Container Registry username.
image_registry_server_password	Specify the Azure Container Registry password.
aks_kubernetes_namespace	Enter the namespace for Kubernetes Deployment. For example, idm.
azure_postgres_server_name	Specify the Azure PostgreSQL server name. For example, idmpgserver.

NOTE: All the remaining fields are pre-filled while generating the `terraform.tfvars` file.

4. Run the following command and sign-in to Azure-CLI.

   ```
   az login
   ```

5. Perform the following steps to store terraform state file in azure storage account:

   5a Run the following command to create a storage account in azure.

```
az storage account create --name
"${AZURE_STORAGE_ACCOUNT_FOR_TFSTATE}" --resource-group
"${AZURE_RESOURCE_GROUP_NAME}" --location
"${AZURE_RESOURCE_GROUP_LOCATION}" --sku Standard_LRS --encryption-
services blob
```

For example,

```
az storage account create --name stract10226600913781 --resource-
group idvault-rg --location eastus --sku Standard_LRS --encryption-
services blob
```

5b Run the following command to create a azure storage container:

```
az storage container create -n terraform-state --account-name
"${AZURE_STORAGE_ACCOUNT_FOR_TFSTATE}"
```

For example,

```
az storage container create -n terraform-state --account-name
stract10226600913781
```

5c Navigate to `Identity_Manager_4.8.6_Containers/terraform/IDM_Azure_Terraform_Configuration/`directory, go to `main.tf` file and update the following details under `backend "azurerm"` field.

Fields	Description
resource_group_name	Specify the azure Resource group name. For example, idvault-rg.
storage_account_name	Specify the azure storage account name. For example, stract89671501132193.
container_name	Indicates the azure container name. This field is auto generated.
key	Indicates the azure key name. This field is auto generated.

6 Review the modified details in `terraform.tfvars` and `main.tf` file and then upload the updated `IDM_4.8.6_Azure_Terraform_Configuration.zip` file into azure cloud shell.

7 Run the following command to download all the required plug-ins needed for infrastructure deployment.

```
terraform init
```

8 Run the following command to plan and understand the deployment based on the input.

```
terraform plan
```

9 Run the following command to create the infrastructure as defined in the input.

```
terraform apply --auto-approve
```

NOTE: If you see any errors while running the terraform commands, refer to troubleshooting Running the Terraform apply Command Displays an Exception.

10 (Optional) Run the following command to identify Azure account specific information such as storage account name, key vault name and database administrator details.

```
terraform output
```

11 Run the following command to create a secret value in the key vault.

```
az keyvault secret set --name idm-common-password --vault-name <key
vault name> --value "novell@123"
```

NOTE: As per your requirement, you can specify multiple secret values using Values.yaml file.

12 Navigate to the directory you have generated the TLS certificates, run the following command to export the certificate and key to .pfx format.

```
openssl pkcs12 -export -out tls.pfx -inkey tls.key -in tls.crt -passout
pass:''
```

NOTE: The `tls.pfx` file will be used in the key vault to validate the secret values and keys.

13 Upload the `tls.pfx` file in the azure cloud shell and run the following command to import the file to the key vault.

```
az keyvault certificate import --vault-name <key vault name> -n
"ingress-tls-cert" -f tls.pfx
```

For example,

```
az keyvault certificate import --vault-name idmkv20220712 -n "ingress-
tls-cert" -f tls.pfx
```

14 Navigate to the `Identity_Manager_4.8.6_Containers/helm_charts/` directory and update the following fields in `values.yaml` file.

Table 19-2

Sections	Fields	Description
Advanced Edition of Identity Manager	IS_ADVANCED_EDITION	Specify your choice to deploy Advanced Edition of Identity Manager.(true/false)
Azure PostgreSQL Server instance	AZURE_POSTGRESQL_REQUIRED	Specify your choice to use Azure PostgreSQL server instance as Database Server for Identity Applications and Reporting.(y/n)
Registry credentials for Identity Manager docker images	registry	Specify the Azure Container Registry Server Name.
	name	Specify the name of the Kubernetes secret which contains the login credentials of the registry.

Sections	Fields	Description
Data Persistence	**Persistent Storage for Identity Engine** volumeClaimTemplate: ◆ storageClassName ◆ storageSize	Specify the storage class name and the storage size for the Volume Claim Template to be used by Identity Engine.
	Shared Persistent Storage ◆ existingClaim dynamicClaim: ◆ storageClassName ◆ storageSize	If you want to use an existing Persistent Volume Claim (PVC), enter the name of the existing claim. Else, for dynamic provisioning of PVC, specify the storage class name and the storage size.
Secret Manager for sensitive data such as passwords, keys and certificates	azureKeyVaultName:	Specify name of the Azure Key Vault.
	azureKeyVaultTenantId	Specify the azure key vault TenantId. Refer to terraform output Step 10.
	azureUserAssignedIdentityID	Specify the Client ID of the user-defined Managed Identity used by the Azure Key Vault Secret Provider. You can run the following az cli command to retrieve the Client ID of the identity: `az aks show -g <Resource Group> -n <AKS Cluster Name> --query addonProfiles.azureKeyvaultSecretsProvider.identity.clientId -o tsv`
Ingress Configuration	host	Specify the domain name for accessing the Identity Manager web applications.
	azureKeyVaultCertificateName	Specify the name of the Azure Key Vault Certificate containing the TLS certificate and the private key.

Sections	Fields	Description
Identity Engine Configuration	deploy	Specify your choice to deploy Identity Engine.(true/false)
	replicaCount	Specify the number of Identity Engine replica pods. For example: ‘2’.
	ID_VAULT_TREENAME	Specify Identity Vault tree name.
	ID_VAULT_SERVER_CONTEXT	Specify Identity Vault Server Context.
	ID_VAULT_DRIVER_SET	Specify Identity Vault Default Driver Set Name.
	ID_VAULT_DEPLOY_CTX	Specify Identity Vault Default Driver Set Deploy Context.
	ID_VAULT_ADMIN_LDAP	Specify Identity Vault Admin DN.
	ID_VAULT_PASSWORD:	Specify Azure Key Vault secret containing Identity Vault Admin Password.
	secret	
	ID_VAULT_RSA_KEYSIZE	Specify Key size for creation of RSA certificate authority keys and server keys.
	ID_VAULT_EC_CURVE	Specify Curve for the creation of EC certificate authority keys and server keys.
	ID_VAULT_CA_LIFE	Specify Certificate life for the creation of default server certificates.
	Resource requests and limits ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests and limits for Identity Engine. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

Sections	Fields	Description
One SSO Provider (OSP) Configuration	deploy	Specify your choice to deploy OSP. (true/false)
	replicaCount	Specify the number of Identity Engine replica pods. For example: ‘2’.
	OSP_CUSTOM_NAME	Specify OSP custom login screen name.
	SSO_SERVICE_PWD: secret	Specify Azure Key Vault secret containing OSP Client Password.
	Resource requests and limits ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests and limits for OSP. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

Sections	Fields	Description
Identity Applications Configuration	deploy	Specify your choice to deploy Identity application.(true/false)
	replicaCount	Specify the number of Identity Engine replica pods. For example: ‘2’.
	UA_ADMIN	Specify Identity Applications Administrator DN
	UA_ADMIN_PWD: secret	Specify Azure Key Vault secret containing Identity Applications Administrator Password.
	UA_WFE_DB_PLATFORM_OPTION	Specify Identity Applications and Workflow Engine Database Platform. The supported values are: postgres, oracle and mssql
	UA_ORACLE_DATABASE_TYPE	If Database Platform is Oracle, Specify the configuration of database.(sid/service)
	UA_WFE_DB_HOST	Specify Identity Applications and Workflow Engine Database Server Host.
	UA_WFE_DB_PORT	Specify Identity Applications and Workflow Engine Database Server Port number.
	UA_DATABASE_NAME	Specify Identity Applications Database Name.
	WFE_DATABASE_NAME	Specify Workflow Engine Database Name.
	UA_WFE_DATABASE_USER	Specify Identity Applications Database Username.
	UA_WFE_DATABASE_PWD: secret	Specify Azure Key Vault secret containing Identity Applications Database User Password.
	Resource requests and limits ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests and limits for Identity Application. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

Sections	Fields	Description
Form Renderer Configuration	deploy	Specify your choice to deploy Form renderer. (true/false)
	Resource requests and limits - cpu - memory	Specify the cpu and memory values of resource requests and limits for Form renderer. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
ActiveMQ Configuraion	deploy	Specify your choice to deploy ActiveMQ. (true/false)
	Resource requests and limits - cpu - memory	Specify the cpu and memory values of resource requests and limits for Form renderer. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

Sections	Fields	Description
Identity Reporting Configuration	deploy	Specify your choice to deploy Identity reporting.(true/false)
	RPT_ADMIN	Specify Identity Reporting Administrator DN
	RPT_ADMIN_PWD: secret	Specify Azure Key Vault secret containing Identity Reporting Administrator Password.
	RPT_DATABASE_PLATFORM_OPTION	Specify Identity Repoting Database Platform. The supported values are: postgres, oracle and mssql
	RPT_ORACLE_DATABASE_TYPE	If Database Platform is Oracle, Specify the configuration of database.(sid/service)
	RPT_DATABASE_HOST	Specify Identity Reporting Database Host.
	RPT_DATABASE_PORT	Specify Identity Reporting Database Port number.
	RPT_DATABASE_NAME	Specify Identity Reporting Database Name.
	RPT_DATABASE_USER	Specify Identity Reporting Database User.
	RPT_DATABASE_SHARE_PASSWORD: secret	Specify Azure Key Vault secret containing Identity Reporting Database Account Password
	Resource requests and limits ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests and limits for Form renderer. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

Sections	Fields	Description
Self Service Password Reset (SSPR) Configuration	deploy	Specify your choice to deploy SSPR. (true/false)
	CONFIGURATION_PWD:	Specify the password that you want to create for an administrator to configure SSPR.
	secret:	
	Resource requests and limits ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests and limits for SSPR. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
Identity Console Configuration	deploy	Specify your choice to deploy Identity console.(true/false)
	ID_CONSOLE_USE_OSP	Do you want to use One SSO Provider (OSP) as the login method for Identity Console. For example 'n'
	Resource requests and limits ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests and limits for Identity Console. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
Advanced Configuration	**DATA_CONTAINERS**:	
	DATA_CONTAINERS_LDIF	Specify LDIF configuration for creating the data containers.
	ROOT_CONTAINER GROUP_ROOT_CONTAINER USER_CONTAINER ADMIN_CONTAINER	Specify DNs for Root container, Group root container, User Container and Admin container
	Kubernetes Cluster Domain:	Specify Kubernetes Cluster Domain
	KUBE_SUB_DOMAIN	

After updating the `terraform.tfvars` and `values.yaml` file, follow the steps mentioned in the "Identity Manager Container Deployment on Azure Kubernetes Service" on page 141 to complete the Identity Manager Container deployment on azure.

Identity Manager Container Deployment on Azure Kubernetes Service

This section provides information on deploying the Identity Manager containers on Azure kubernetes service.

1 Run the following command to connect to the Kubernetes cluster and store the required configuration:

```
az aks get-credentials --resource-group <resource group> --name
<cluster name> --overwrite-existing
```

For example,

```
az aks get-credentials --resource-group idvault-rg --name cluster-name
--overwrite-existing
```

2 Run the following command to create an Nginx instance used to run the Load balancer:

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
```

3 Perform the following steps to obtain the public IP address of the Kubernetes Service from the Azure portal.

3a Go to the Azure home page, and click **All resources**.

3b In the search box, type load balancer and click enter.

3c Click the load balancer associated with your resource group and AKS cluster.

3d Go to **Settings** > **Front end IP configuration** and copy the first IP address.

4 Navigate to the IDM_4.8.6_Cloud_Deployment_files directory, run the following command to install Nginx-ingress controller:

```
helm install nginx-ingress ingress-nginx/ingress-nginx --namespace
<your namespace> --set controller.replicaCount=1 --set
controller.service.loadBalancerIP=<Kubernetes public IP address
obtained from the Azure portal> --set
controller.service.annotations."service\.beta\.kubernetes\.io/azure-
dns-label-name"=<Domain Name>
```

For example,

```
helm install nginx-ingress ingress-nginx/ingress-nginx --namespace idm
--set controller.replicaCount=1 --set
controller.service.loadBalancerIP=192.168.0.1 --set
controller.service.annotations."service\.beta\.kubernetes\.io/azure-
dns-label-name"=identitymanager
```

5 To run the Helm Charts, run the following command:

```
helm install identity-manager helm_charts/identity-manager-1.1.0.tgz --
namespace <namespace> -f values.yaml
```

For example,

```
helm install identity-manager helm_charts/identity-manager-1.1.0.tgz --
namespace idm -f values.yaml
```

6 Run the following command to view the list of pods that are running in the Azure Kubernetes Service:

```
kubectl get pods --watch -n <namespace>
```

For example,

```
kubectl get pods --watch -n idm
```

Setting up of Remote Loader Service on Azure Kubernetes Service

Perform the following steps to setup Remote Loader service on Azure Kubernetes Service:

1 Create Persistent Volume (PV) and Persistent Volume Claim(PVC) for persisting the Remote Loader data.

2 Inside the PV, create a directory `rdxml`.

3 In the `rdxml` directory, create the `driverconf` directory.

4 In the `rdxml` directory, create the `keystore` directory.

5 Copy all the required keystore files and certificates to the keystore directory.

6 In the `rdxml` directory, create a new `.txt` file. For example, `StartupRL.txt`.

7 In the `StartupRL.txt` file, specify the required content in the following format:

```
<Remote Loader configuration file> -sp <driver password> <Remote Loader
password>
```

For example:

```
config8000.txt -sp dirxml dirxml
```

Alternatively, you can also specify the entries in the following format:

```
<Remote Loader configuration file> -ksp <keystore password> -kp <key
password>
```

For example:

```
config8000.txt -ksp dirxml -kp dirxml
```

8 From the container tarball, copy `helm-charts/rl-values.yaml` and update the following fields in the file:

Table 19-3

Section	Fields	Description
Registry credentials for Remote Loader docker images	registry	Specify the Azure Container Registry Server Name.
	name	Specify the name of the Kubernetes secret which contains the login credentials of the registry.

Section	Fields	Description
Data Persistence	**Shared Persistent Storage** ◆ existingClaim dynamicClaim: ◆ storageClassName ◆ storageSize	If you want to use an existing Persistent Volume Claim (PVC), enter the name of the existing claim. Else, for dynamic provisioning of PVC, specify the storage class name and the storage size.
Driver Startup File	driverStartupFileName	Specify Name of the driver startup file. For example, `StartupRL.txt`
Remote Loader Service	ports	Specify the list of ports Remote Loader service listening on.
Resource Requests and Limits	Resource requests ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
	Limits ◆ cpu ◆ memory	Specify the cpu and memory values of limits. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

9. From the container tarball, copy `helm-charts/ remote-loader-1.1.0.tgz` and run the following helm command to install remote loader:

   ```
   helm install remote-loader helm_charts/remote-loader-1.1.0.tgz --
   namespace idm -f rl-values.yaml
   ```

Setting up of Fanout Agent service on Azure Kubernetes Service

Perform the following steps to setup on Azure Kubernetes Service:

1. From the container tarball, copy `helm-charts/fa-values.yaml` and update the following fields in the file.:

Table 19-4

Section	Fields	Description
Registry credentials for Remote Loader docker images	registry	Specify the Azure Container Registry Server Name.
	name	Specify the name of the Kubernetes secret which contains the login credentials of the registry.
Data Persistence	**Shared Persistent Storage** - existingClaim dynamicClaim: - storageClassName - storageSize	If you want to use an existing Persistent Volume Claim (PVC), enter the name of the existing claim. Else, for dynamic provisioning of PVC, specify the storage class name and the storage size.
Fanout Agent Service	ports	Specify the value of Remote Loader service ports listening on.
Resource Requests and Limits	Resource requests - cpu - memory	Specify the cpu and memory values of resource requests.for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
	Limits - cpu - memory	Specify the cpu and memory values of limits. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

Section	Fields	Description
ActiveMQ Service	install	Specify If you want to deploy ActiveMQ Service for Fanout Agent(true/false)
	Resource requests ◆ cpu ◆ memory	Specify the cpu and memory values of resource requests for ActiveMQ services. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)
	Limits ◆ cpu ◆ memory	Specify the cpu and memory values of limits for ActiveMQ services. for more details refer Resource Management for Pods and Containers (https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/)

2. From the container tarball, copy `helm-charts/fanout-agent-1.1.0.tgz` and run the following helm command to install fanout agent:

```
helm install fanout-agent helm_charts/remote-loader-1.1.0.tgz --
namespace idm -f rl-values.yaml
```

20 Post-deployment Tasks

This section provides instructions on the post-deployment tasks for deploying the Identity Manager containers.

Access Identity Manager Applications

To access any of the Identity Manager Applications, enter the following URL in your browser.

```
https://<domainname>/<APP NAME>
```

For example,

```
https://identitymanager.eastus.cloudapp.azure.com/idmdash
```

Deploy Designer on Azure Virtual Machine

Perform the following steps to deploy Designer on Azure Virtual Machine under VM subnet.

1. Refer to Quickstart:Linux VMs to create a Linux machine or refer to Quickstart:Windows VMs to create a Windows VM.
2. Assign a public IP address for connecting the designer.
3. Download and install Designer on the virtual machine created. For more information, see Installing Designer Linux VM or Installing Designer Windows VM.

Deploy SLM for IGA on Azure Virtual Machine

Perform the following steps to deploy Sentinel Log Management (SLM) for Identity Governance and Administration (IGA) on Azure Virtual Machine under VM subnet.

1. Create a Linux machine, refer to Quickstart:Linux VMs.
2. Assign a public IP address for connecting the installer for SLM for IGA.
3. Download and install SLM for IGA on the virtual machine created. For more information, see Installing SLM for IGA.

Handling RPM Updates and Third Party Files for AKS

This service pack provides an efficient way to handle RPM updates and third-party files in a container. This can be achieved by placing the required RPM files, library (`.so`) files, and third-party `.jar` files in the `mountfiles` directory. The RPM files present in the `mountfiles` directory will be updated forcefully. The `.so` and `.jar` files are automatically soft linked to the `/opt/novell/`

`eDirectory/lib64/nds-modules/` and /`opt/novell/eDirectory/lib/dirxml/classes/` directories respectively when the containers are restarted. The purpose of this option is to have the updated drivers within the docker containers.

NOTE: Currently, this enhancement is applicable for Identity Manager Engine and Remote Loader containers.

---

1 Log in to Azure portal and click >_.

2 Run the following command to log in to your Persistent Volume:

```
kubectl exec -it -n idm <container name> bash
```

For example

```
kubectl exec -it -n idm identityengine-0 bash
```

3 Inside the container, create the respective container-specific mountfiles directory.

3a (Optional) For Identity Manager Engine container, run the following command:

```
mkdir -p /config/idm/mountfiles
```

3b (Conditional) For Remote Loader container, run the following command:

```
mkdir -p /config/rdxml/mountfiles
```

4 Copy the required files to the respective container-specific `mountfiles` directory.

4a (Conditional) For Identity Manager Engine container, copy the files to `/config/idm/mountfiles` by running the following command:

```
kubectl cp -n <namespace> <file to be copied> <container name>:/
config/idm/mountfiles
```

4b (Conditional) For Identity Remote Loader container, copy the files to `/config/rdxml/mountfiles` by running the following command:

```
kubectl cp -n <namespace> <file to be copied> <container name>:/
config/rdxml/mountfiles
```

NOTE: The supported file formats are `.so`, `.jar`, and `.rpm`.

---

5 Run the following comment to restart the container:

```
helm upgrade identity-manager helm_charts/identity-manager-1.1.0.tgz --
namespace <namespace> -f values.yaml
```

IMPORTANT: Whatever the rpm present inside the mount files is prioritized, in case of future updates you need to carefully remove the existing rpm and update the latest one.

---

21 Maintaining the Deployment of Identity Manager Containers

This section provides information on how to maintain your deployment of pods (group of one or more containers) on the Azure Kubernetes cluster.

IMPORTANT: If you need to add more charts to your deployment, to re-create a deployment or to scale up (or down) the number of pods in deployment, perform the following steps:

1. Log in to the Azure Portal.
2. Click >_, and run the following command.

```
helm upgrade identity-manager helmcharts/identity-manager-1.1.0.tgz --
namespace <namespace> -f values.yaml
```

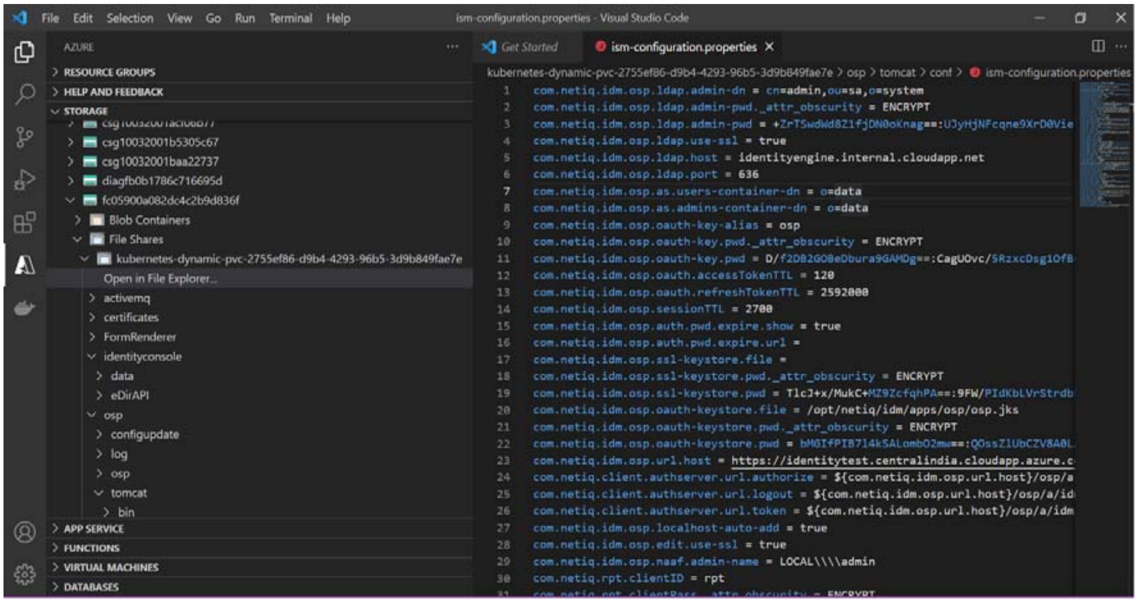
Managing the Data Persistence Layer

Perform the following steps to modify the data layer of a pod in Visual Studio Code Application:

1. Download and install the visual studio code. Refer to code.visualstudio.com/download.
2. Open the visual code application, click ⊞.
3. Type **Azure** in the search and press Enter.
4. In Azure Tools, click **Install.**
5. Click A.
6. Go to **STORAGE** > **Security-IdentityMgmtAZ-NonProd** > **fc05900a082dc4c2b9d836f** > **File shares** > **Kubernetes dynamic pvc** (For more information, see Figure 21-1 on page 150).

NOTE: In the above navigation path, the **Security-IdentityMgmtAZ-NonProd** and **fc05900a082dc4c2b9d836f** key is mentioned for your reference and would change as per your login credentials.

Figure 21-1 Visual studio storage



```
com.netiq.idm.osp.ldap.admin-dn = cn=admin,ou=sa,o=system
com.netiq.idm.osp.ldap.admin-pwd._attr_obscurity = ENCRYPT
com.netiq.idm.osp.ldap.admin-pwd = +ZrTSwdWd8Z1fjDN0oKnag==:UJyHjNFcqne9XrD0Vie
com.netiq.idm.osp.ldap.use-ssl = true
com.netiq.idm.osp.ldap.host = identityengine.internal.cloudapp.net
com.netiq.idm.osp.ldap.port = 636
com.netiq.idm.osp.as.users-container-dn = o=data
com.netiq.idm.osp.as.admins-container-dn = o=data
com.netiq.idm.osp.oauth-key-alias = osp
com.netiq.idm.osp.oauth-key.pwd._attr_obscurity = ENCRYPT
com.netiq.idm.osp.oauth-key.pwd = D/f2DB2GOBeDbura9GAMDg==:CagUOvc/5RzxcDsg1OfB
com.netiq.idm.osp.oauth.accessTokenTTL = 120
com.netiq.idm.osp.oauth.refreshTokenTTL = 2592000
com.netiq.idm.osp.sessionTTL = 2700
com.netiq.idm.osp.auth.pwd.expire.show = true
com.netiq.idm.osp.auth.pwd.expire.url =
com.netiq.idm.osp.ssl-keystore.file =
com.netiq.idm.osp.ssl-keystore.pwd._attr_obscurity = ENCRYPT
com.netiq.idm.osp.ssl-keystore.pwd = TlcJ+x/MukC+MZ9ZcfqhPA==:9FW/PIdKbLVrStrdb
com.netiq.idm.osp.oauth-keystore.file = /opt/netiq/idm/apps/osp/osp.jks
com.netiq.idm.osp.oauth-keystore.pwd._attr_obscurity = ENCRYPT
com.netiq.idm.osp.oauth-keystore.pwd = bMGIfPIB714kSALombO2mw==:QOssZ1UbCZV8A0L
com.netiq.idm.osp.url.host = https://identitytest.centralindia.cloudapp.azure.c
com.netiq.client.authserver.url.authorize = ${com.netiq.idm.osp.url.host}/osp/a
com.netiq.client.authserver.url.logout = ${com.netiq.idm.osp.url.host}/osp/a/id
com.netiq.client.authserver.url.token = ${com.netiq.idm.osp.url.host}/osp/a/idm
com.netiq.idm.osp.localhost-auto-add = true
com.netiq.idm.osp.edit.use-ssl = true
com.netiq.idm.osp.naaf.admin-name = LOCAL\\\\admin
com.netiq.rpt.clientID = rpt
com.netiq.rpt.clientPass._attr_obscurity = ENCRYPT
```

7 Select the file, change the required properties.

8 Go to **File** menu, click **save**.

9 After saving all the changes, restart the pods (see "Restarting the Pods" on page 150).

Restarting the Pods

Perform the following steps to restart the pods in Azure.

For example, if you have made any changes to the data layer, containers need to be restarted.

1 Log in to the Azure portal.

2 Type All resources in the search and press enter.

3 Under Services, select **All resources**.

4 Go to your Kubernetes cluster.

5 Under Kubernetes resources, click **Workloads**.

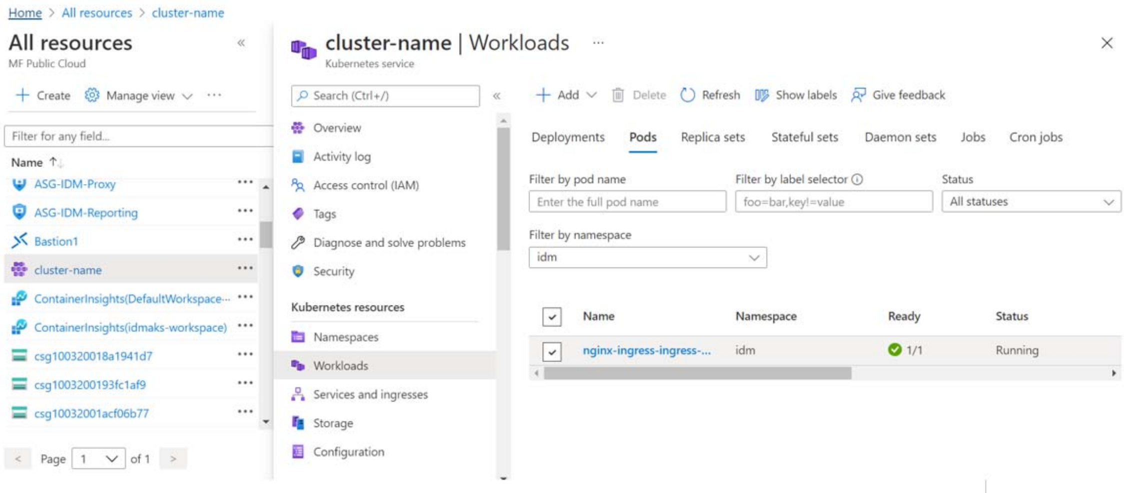
6 Go to **Pods** tab, click **Filter by namespace** drop-down list, select **idm**.

7 Select the pod (s) you want to restart.

8 Click **Delete**. For more information, see Figure 21-2 on page 151.

NOTE: The system identifies the deleted pod (s) and automatically triggers the deployment to the new pod(s).

Figure 21-2 *Kubernetes cluster*



☑	Name	Namespace	Ready	Status
☑	nginx-ingress-ingress-...	idm	✓ 1/1	Running

Accessing the Pods

Perform the following steps to access the pods in Azure:

1. Log in to the Azure portal.
2. Click >_.
3. Run the following command to identify the name of the pod:

   ```
   kubectl get pods --watch -n <namespace>
   ```

   For example,

   ```
   kubectl get pods --watch -n idm
   ```

4. Run the following command:

   ```
   kubectl exec -it -n idm <pod-name> bash
   ```

   For example,

   ```
   kubectl exec -it -n idm userapp-d54f98b58-qcpmz bash
   ```

Restarting the Azure Kubernetes cluster

Perform the following steps to ensure that all the IDM components shutdown gracefully:

1. Log in to the Azure portal.
2. Go to your Kubernetes cluster.
3. Under Kubernetes resources, Click **Workloads**.
4. Go to **Deployments** tab, Click **Filter by namespace** drop-down list, select your namespace.
5. Select the deployment (s), Click **Delete**.
6. Go to **Pods** tab, Click **Filter by namespace** drop-down list, select your namespace.

7. Click **Refresh**, to check the component-pod(s) are completely removed.
8. Go to **Stateful sets** tab, click **Filter by namespace** drop-down list, select your namespace.
9. Select the Stateful set (s), Click **Delete**.
10. Again, Go to **Pods** tab, Click **Filter by namespace** drop-down list, select your namespace.
11. Click **Refresh**, to check the components are completely removed.
12. Go to Overview Under Kubernetes resources, Click **Stop** and wait for some time.
13. Once the cluster is stopped completely, then click **Start**.
14. In Azure CLI, run the following command to restart Nginx-ingress controller:

    ```
    helm upgrade nginx-ingress ingress-nginx/ingress-nginx --namespace 
    <your namespace> --set controller.replicaCount=1 --set 
    controller.service.loadBalancerIP=<Public IP address assigned to load 
    balancer> --set 
    controller.service.annotations."service\.beta\.kubernetes\.io/azure-
    dns-label-name"=<LoadBalancerName>
    ```

    For example

    ```
    helm upgrade nginx-ingress ingress-nginx/ingress-nginx --namespace idm 
    --set controller.replicaCount=1 --set 
    controller.service.loadBalancerIP=192.168.0.1 --set 
    controller.service.annotations."service\.beta\.kubernetes\.io/azure-
    dns-label-name"=identitymanager
    ```

15. Run the following command to restart the remaining components:

    ```
    helm upgrade identity-manager helm_charts/identity-manager-1.1.0.tgz --
    namespace <namespace> -f values.yaml
    ```

    For example

    ```
    helm upgrade identity-manager helm_charts/identity-manager-1.1.0.tgz --
    namespace idm -f values.yaml
    ```

16. Wait till all the pods are in ready state.

    To check the current status of the pods run the following command:

    ```
    kubectl get pods --watch -n <namespace>
    ```

17. Run the following command to restart the engine service:

    ```
    kubectl rollout restart -n <namespace> statefulset/identityengine
    ```

    For example

    ```
    kubectl rollout restart -n idm statefulset/identityengine
    ```

18. Repeat Step 16.

Scaling Identity Manager Engine Replicas

The Procedure for Scaling Identity Manager Engine replicas is divided into two parts:

Use case for Downscaling Identity Manager Engine Replicas

This topic describes a basic use case that you can refer for downscaling Identity Manager Engine replicas.

Before You Start

- Understand the process carefully before downscaling the Identity Manager engine replicas.
- You can scale down one pod at a time.

Use case

Assume that we have 3 instances of Identity Manager engine, and their pod names are:

- identityengine-0
- identityengine-1
- identityengine-2

When you downscale to two Identity Manager engine instances, identityengine-2 will go down.

The following actions are required on identityengine-2 before downscaling:

1. Migrate/recreate drivers running on instance identityengine-2 to another instance.
2. Delete the server from the tree.
3. Run the following command to List Persistent Volume Claims (PVC):

   ```
   kubectl get pvc -n <namespace>
   ```

   For example:

   ```
   kubectl get pvc -n idm
   ```
4. Run the following command to delete PVC of identityengine-2:

   ```
   kubectl delete pvc -n <namespace> <identityengine replica name>
   ```

   For example:

   ```
   kubectl delete pvc -n idm identityenginedata-identityengine-2
   ```
5. Set the engine replica count to 2 in the `values.yaml` file.
6. Scale down using the following command:

   ```
   helm upgrade identity-manager helm_charts/identity-manager-1.1.0.tgz --namespace <namespace> -f values.yaml
   ```

Upscaling of Identity Manager Engine Replicas

Perform the following steps to upscale Identity Manager Engine replicas:

1. Log in to the Azure portal.
2. Click >_.
3. Increase Identity Manager Engine replica count in the `values.yaml` file.

4 Scale up using the following command:

```
helm upgrade identity-manager helm_charts/identity-manager-1.1.0.tgz --
namespace <namespace> -f values.yaml
```

Scaling OSP and Identity Applications Replicas

Perform the following steps to upscale or downscale OSP and Identity Applications replicas:

1 Log in to the Azure portal.

2 Click >_.

3 Increase or decrease OSP/Identity Applications replica count in the `values.yaml` file.

4 Scale up or scale down using the following command:

```
helm upgrade identity-manager helm_charts/identity-manager-1.1.0.tgz --
namespace <namespace> -f values.yaml
```

Destroying entire resources created with Terraform

Perform the following steps to destroy the resources created with Terraform:

NOTE: Be cautious before proceeding with this process.

1 Navigate to the directory where you have used terraform to create resources.

2 Run the following command to delete the resources:

```
terraform destroy --auto-approve
```

3 (optional) In case there are any errors seen during the above command, please re-run the same command.

22 Troubleshooting

This section provides useful information for troubleshooting problems with the Identity Manager containers that are deployed on Azure Kubernetes Service.

Running the Terraform Commands Displays an Error

Issue: When you are running the Terraform commands in the Azure portal, you might come across errors that can block the infrastructure deployment. Some of the possible errors are given below.

```
Error: making Read request on Azure KeyVault Secret slesvmpwd:
azure.BearerAuthorizer#WithAuthorization: Failed to refresh the Token for
request to https://idmkv88211894240361.vault.azure.net/secrets/slesvmpwd/
?api-version=7.1: StatusCode=401 -- Original Error: adal: Refresh request
failed. Status Code = '401'. Response body:
{"error":{"code":"invalid_request","message":"Required audience parameter
not specified"}} Endpoint http://localhost:50342/oauth2/token
│
│ with module.server.data.azurerm_key_vault_secret.slesvmpwd,
│ on sles15sp2-server/main.tf line 42, in data "azurerm_key_vault_secret"
"slesvmpwd":
│ 42: data "azurerm_key_vault_secret" "slesvmpwd" {

kubernetes_secret.example3: Creation complete after 1s [id=idm/ingress-
tls]
?│ Error: error detecting capabilities: error PostgreSQL version: dial tcp
52.186.162.130:5432: connect: connection timed out
│
│ with module.dbserver.postgresql_database.igaworkflowdb,
│ on Azure-PG-Server/main.tf line 217, in resource "postgresql_database"
"igaworkflowdb":
│ 217: resource "postgresql_database" "igaworkflowdb" {
│ az aks get-credentials --resource-group rajj_rg --name cluster-name --
overwrite-existing│ Error: making Read request on Azure KeyVault Secret
uawfedbuser: azure.BearerAuthorizer#WithAuthorization: Failed to refresh
the Token for request to https://idmkv80357639695488.vault.azure.net/
secrets/uawfedbuser/?api-version=7.1: StatusCode=401 -- Original Error:
adal: Refresh request failed. Status Code = '401'. Response body:
{"error":{"code":"invalid_request","message":"Required audience parameter
not specified"}} Endpoint http://localhost:50342/oauth2/token
│
│ with module.dbserver.data.azurerm_key_vault_secret.uawfedbuser,
│ on Azure-PG-Server/main.tf line 28, in data "azurerm_key_vault_secret"
"uawfedbuser":
│ 28: data "azurerm_key_vault_secret" "uawfedbuser" {
```

```
module.dbserver.null_resource.delay: Creation complete after 5m0s
[id=5961346882319865424]
?

│ Error: waiting for creation of Managed Kubernetes Cluster "test-
akscluster" (Resource Group "idmtest_rg"):
Code="ReconcileStandardLoadBalancerError" Message="Reconcile standard load
balancer failed. Details: outboundReconciler retry failed: Category:
ClientError; SubCode: PublicIPCountLimitReached; Dependency:
Microsoft.Network/PublicIPAddresses; OrginalError:
Code=\"PublicIPCountLimitReached\" Message=\"Cannot create more than 10
public IP addresses for this subscription in this region.\" Details=[];
AKSTeam: Networking, Retriable: false."
│
│ with module.aks.azurerm_kubernetes_cluster.main,
│ on .terraform/modules/aks/main.tf line 10, in resource
"azurerm_kubernetes_cluster" "main":
│ 10: resource "azurerm_kubernetes_cluster" "main" {
```

Workaround: To resolve this issue, perform the following steps:

1. Login to the Azure portal.
2. (Optional) Locate the Azure Kubernetes cluster configuration file path.
3. Run the following commands to overwrite the default config file `~/.kube/config`.

   ```
   aks get-credentials --resource-group <resource-group-name> --name
   <kube-cluster-name> --overwrite-existing
   ```

   ```
   export KUBE_CONFIG_PATH=~/.kube/config
   ```

4. Re-run the Terraform commands. For more information, see Step 13, Step 14, Step 15 in the Deployment using the Configuration Generator.

Creating Public IP Address Displays an Exception

Issue: When you are trying to create a public IP address on Azure, you might see the following error.

```
│ Error: waiting for creation of Managed Kubernetes Cluster "cluster-name"
(Resource Group "raj_rg"): Code="ReconcileStandardLoadBalancerError"
Message="Reconcile standardload balancer failed. Details:
outboundReconciler retry failed: Category: ClientError; SubCode:
PublicIPCountLimitReached; Dependency: Microsoft.Network/
PublicIPAddresses;OrginalError: Code=\"PublicIPCountLimitReached\"
Message=\"Cannot create more than 10 public IP addresses for this
subscription in this region.\" Details=[]; AKSTeam: Networking, Retriable:
false."
│
│ with module.aks.azurerm_kubernetes_cluster.main,
│ on .terraform/modules/aks/main.tf line 10, in resource
"azurerm_kubernetes_cluster" "main":
│ 10: resource "azurerm_kubernetes_cluster" "main" {
```

Workaround: Contact the Azure team to increase the quota for subscription or try to remove unused public IP addresses that are already assigned.

Running the Terraform apply Command Displays an Exception

Issue: When you are running the `terraform apply` command in the Azure portal, you might see the following exception:

```
(AgentPoolK8sVersionNotSupported) Version <version> is not supported in
this region.
Please use [az aks get-versions] command to get the supported version list
in this region.
For more information, please check https://aka.ms/supported-version-list
```

Workaround: To resolve this issue, perform the following steps:

1. Log in to the Azure portal.
2. Click >_.
3. Run the following command to identify the supported versions available in your location:

   ```
   az aks get-versions -l <location>
   ```

   For example,

   ```
   az aks get-versions -l eastus
   ```

4. Navigate to the `IDM_4.8.6_Cloud_Deployment_files` directory, go to `main.tf` file and edit the following values:

   ```
   kubernetes_version   = "<version>"
   orchestrator_version = "<version>"
   ```

   For example,

   ```
   kubernetes_version = "1.20.15"
   orchestrator_version = "1.20.15"
   ```

   NOTE: Version `1.20.x` is certified and supported on AKS Kubernetes.

5. Re-run the following command:

   ```
   terraform apply --auto-approve
   ```

Multi-Instance Container Deployment Displays Pending in Azure Kubernetes Service

Issue: When you are deploying multi-instance containers on Azure Kubernetes Service, you might see a few pods in pending state.

Workaround: To resolve the issue, perform the following steps:

1. Log in to the Azure portal.
2. Type kubernetes services in the search and press enter.
3. Select your kubernetes cluster.

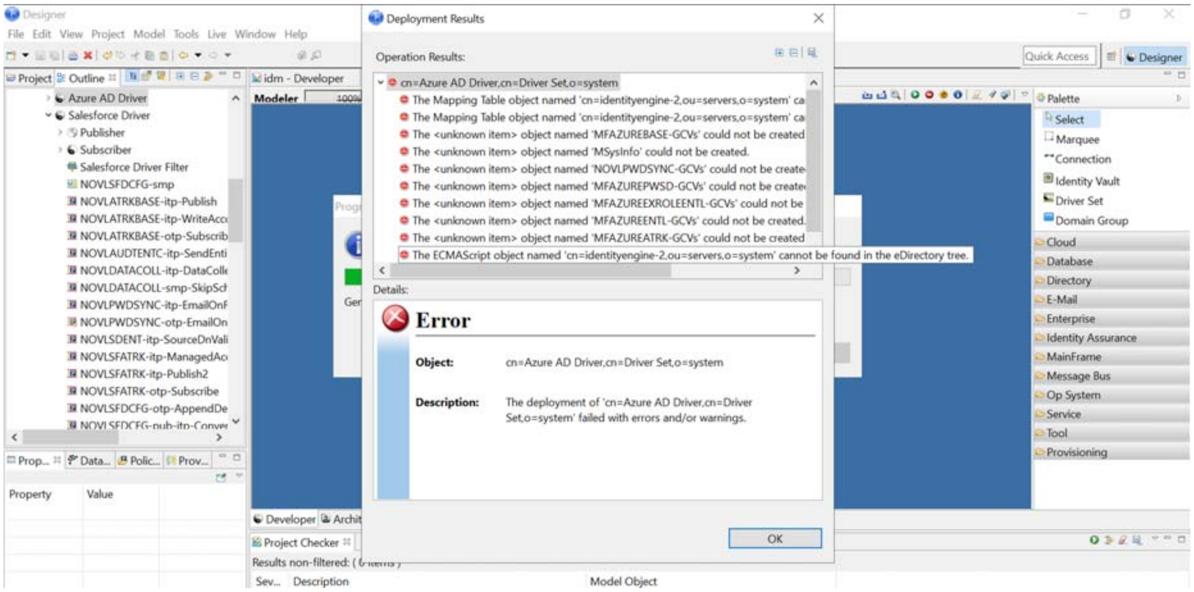
4 Click **Diagnose and solve problems** > **Troubleshoot**.

5 (Conditional) If the issue is not resolved using troubleshooting method, check the usage quota of your azure account.

- 5a Go to Home page, type quotas in the search and press enter.
- 5b Click **Compute**.
- 5c (Conditional) If the current usage column displays 100% with a red line, click **Request quota increase**.
- 5d Enter a new limit in **New limit** text box.
- 5e Click **Submit**.

Driver Deployment is Failing in Designer After Restarting the Engine Instance Pod

Issue: When you restart the engine instance pod, the driver's deployment will fail for the existing Identity Vault. You might see the following error in the Designer application.



```
cn=Azure AD Driver,cn=Driver Set,o=system
  The Mapping Table object named 'cn=identityengine-2,ou=servers,o=system' ca
  The Mapping Table object named 'cn=identityengine-2,ou=servers,o=system' ca
  The <unknown item> object named 'MFAZUREBASE-GCVs' could not be created
  The <unknown item> object named 'MSysInfo' could not be created.
  The <unknown item> object named 'NOVLPWDSYNC-GCVs' could not be create
  The <unknown item> object named 'MFAZUREPWSD-GCVs' could not be create
  The <unknown item> object named 'MFAZUREEXROLEENTL-GCVs' could not be
  The <unknown item> object named 'MFAZUREENTL-GCVs' could not be created.
  The <unknown item> object named 'MFAZUREATRK-GCVs' could not be created
  The ECMAScript object named 'cn=identityengine-2,ou=servers,o=system' cannot be found in the eDirectory tree.
```

Error

Object: cn=Azure AD Driver,cn=Driver Set,o=system

Description: The deployment of 'cn=Azure AD Driver,cn=Driver Set,o=system' failed with errors and/or warnings.

Workaround: Go to Designer application and re-import the Identity Vault into Designer.

The Certificate Revocation List URL of the Identity Manager Engine's Secondary Server SSL Certificate does not Contain Correct IP Address of the Primary Server

Issue: When you restart the primary server a new IP address is assigned to it, this new IP address does not match the one in the SSL certificate of the secondary server.

Workaround: Restarting the secondary server would make it reissue the SSL certificate with the new IP address.

Error Code 413 Displayed When Importing Multiple Reports in Identity Manager Reporting

Issue: In Identity Manager Reporting, when you try to import multiple reports with a total size exceeding 1MB, it will show error 413. This happens because the default request body size allowed by the nginx-ingress controller is 1MB.

```
There was an error calling the URL (/IDMRPT-CORE/rpt/definitions/
import?uploadid=<id>&content=text/html&format=application/
json&overwrite=true). The server returned the status code 413 with the
following data which was not parsable JSON data:
```

Workaround: This issue can be resolved by allowing a larger request body size, by adding the following annotation in the `values.yaml` file under the `Ingress Configuration` section:

```
nginx.ingress.kubernetes.io/proxy-body-size: 3m
```