

---

# Identity Intelligence 1.1

## Administrator Guide

April 2020

## **Legal Notice**

© Copyright 2020 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors ("Micro Focus") are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

For additional information, such as certification-related notices and trademarks, see <https://www.microfocus.com/about/legal/>.

---

# Contents

|   |           |
|---|-----------|
| <b>About This Book</b>  | <b>9</b>  |
| <b>1 Introduction</b>   | <b>11</b> |
| Understanding the Identity Intelligence Architecture . . . . .      | 11        |
| Understanding the Identity Intelligence Components . . . . .        | 12        |
| <b>Part I Planning the Installation</b>                             | <b>13</b> |
| <b>2 Implementation Checklist</b>                                   | <b>15</b> |
| <b>3 Deployment Considerations</b>                                  | <b>17</b> |
| Components Needed for Deployment . . . . .                          | 17        |
| Deployment Options . . . . .  | 17        |
| Single-node Deployment . . . . .                                    | 18        |
| Multi-node Deployment . . . . .                                     | 18        |
| High Availability Deployment . . . . .                              | 18        |
| Securing Communication Between Components . . . . .                 | 18        |
| One-Way SSL Authentication . . . . .                                | 19        |
| Mutual SSL Authentication . . . . .                                 | 19        |
| Security Modes Between Components . . . . .                         | 20        |
| <b>4 Installation Options</b>                                       | <b>23</b> |
| Installation Using Scripts . . . . .                                | 23        |
| Manual Installation . . . . .                                       | 23        |
| Deciding to Use the Scripts or Manual Installation Method . . . . . | 23        |
| <b>Part II Installing Identity Intelligence</b>                     | <b>25</b> |
| <b>5 Installing Identity Intelligence by Using Scripts</b>          | <b>27</b> |
| Prerequisites . . . . .   | 27        |
| Understanding the Installation Scripts . . . . .                    | 27        |
| Using the Scripts in Single-node Deployments . . . . .              | 27        |
| Using the Scripts in Multi-node Deployments . . . . .               | 28        |
| <b>6 Installing Identity Intelligence Manually</b>                  | <b>29</b> |
| Installing Vertica . . . . .  | 29        |
| Prerequisites . . . . .   | 29        |
| Installing Vertica . . . . .  | 33        |
| Preparing Your Environment for CDF . . . . .                        | 34        |
| Configuring the Nodes . . . . .                                     | 34        |
| Set System Parameters (Network Bridging) . . . . .                  | 35        |
| Check MAC and Cipher Algorithms . . . . .                           | 36        |
| Check Password Authentication Settings . . . . .                    | 36        |

|   |           |
|---|-----------|
| Installing the Required Operating System Packages . . . . .                       | 36        |
| Remove Libraries . . . . .  | 37        |
| Configuring Time Synchronization . . . . .  | 37        |
| Configuring Firewall . . . . .  | 38        |
| Configuring Proxy . . . . .   | 39        |
| Configuring DNS . . . . .   | 39        |
| Configuring the NFS Server . . . . .  | 42        |
| Disabling Swap Space . . . . .  | 45        |
| (Optional) Create Docker Thinpools . . . . .                                      | 45        |
| Enabling Installation Permissions for a sudo User . . . . .                       | 47        |
| Installing CDF . . . . .  | 49        |
| Installing Identity Intelligence . . . . .  | 50        |
| Configuring the Cluster . . . . .   | 50        |
| Uploading Images to Local Registry . . . . .                                      | 52        |
| Deploying Transformation Hub and Identity Intelligence . . . . .                  | 53        |
| <br>  |           |
| <b>7 Deploying Identity Intelligence in an Existing Cluster</b>                   | <b>55</b> |
| Prerequisites . . . . .   | 55        |
| Deploying Identity Intelligence . . . . .   | 55        |
| <br>  |           |
| <b>8 Post-Installation Configurations</b>   | <b>57</b> |
| Labeling Nodes . . . . .  | 57        |
| Labeling Nodes in a New Cluster . . . . .   | 57        |
| Labeling Nodes in an Existing Cluster . . . . .                                   | 58        |
| Configuring Flannel Memory . . . . .  | 58        |
| Setting the Default Locale for the Database . . . . .                             | 59        |
| Configuring SSL for Vertica . . . . .   | 59        |
| Obtaining Vertica Client Certificates . . . . .                                   | 60        |
| Generating Vertica Server Certificate . . . . .                                   | 60        |
| Enabling SSL in Vertica . . . . .   | 61        |
| Establishing an SSL Communication with Identity Intelligence . . . . .            | 62        |
| Creating a Kafka Scheduler with SSL . . . . .                                     | 62        |
| Performance Tuning for Data Ingestion . . . . .                                   | 63        |
| <br>  |           |
| <b>9 Verifying the Installation</b>   | <b>65</b> |
| <br>  |           |
| <b>10 Installing and Configuring ArcMC</b>  | <b>67</b> |
| <br>  |           |
| <b>Part III Configuring Data Collection</b>                                       | <b>69</b> |
| <br>  |           |
| <b>11 Data Collection Configuration Checklist</b>                                 | <b>71</b> |
| <br>  |           |
| <b>12 Tuning Ingestion of Backdated Events</b>                                    | <b>73</b> |
| <br>  |           |
| <b>13 Installing and Configuring the SmartConnector</b>                           | <b>75</b> |
| Prerequisites . . . . .   | 75        |
| Installing the SmartConnector . . . . .   | 75        |
| Adding Categorization Files . . . . .   | 76        |
| Creating TrustStore for One-Way SSL with Transformation Hub . . . . .             | 77        |
| Creating TrustStore and KeyStore for Mutual SSL with Transformation Hub . . . . . | 77        |
| Configuring the SmartConnector . . . . .  | 81        |

|  |            |
|--|------------|
| Configuring Additional Destination for Identity Manager Audit Events . . . . .       | 83         |
| Verifying the SmartConnector Configuration . . . . .                                 | 84         |
| <b>14 Configuring Entity Change Events Collection</b>                                | <b>85</b>  |
| Understanding the Collection of Entity Change Events . . . . .                       | 85         |
| Configuring the Collection of Entity Change Events . . . . .                         | 85         |
| <b>15 Customizing Data Reconciliation</b>  | <b>87</b>  |
| Understanding Default Reconciliation Fields . . . . .                                | 87         |
| Reconciling Data for the Identity Entity . . . . .                                   | 87         |
| Customizing Data Reconciliation . . . . .  | 88         |
| <b>16 Configuring Data Collection from Identity Manager</b>                          | <b>89</b>  |
| Understanding the Data Collection Process . . . . .                                  | 89         |
| Configuring Entity Data Collection . . . . .   | 90         |
| Configuring Audit Events Collection . . . . .  | 90         |
| Prerequisites . . . . .  | 90         |
| Obtaining the SmartConnector Certificate . . . . .                                   | 90         |
| Configuring Identity Applications . . . . .  | 91         |
| Configuring the Identity Manager Engine . . . . .                                    | 93         |
| Audit Events Used by Identity Intelligence . . . . .                                 | 94         |
| Reverting Backdated Events Configuration . . . . .                                   | 95         |
| <b>17 Configuring Data Collection from Identity Governance</b>                       | <b>97</b>  |
| Data Collection Configuration Checklist . . . . .                                    | 97         |
| Understanding the Data Collection Process . . . . .                                  | 97         |
| Configuring Data Collection . . . . .  | 98         |
| Prerequisites . . . . .  | 98         |
| Configuring One-Way SSL Between Identity Governance and Transformation Hub . . . . . | 99         |
| Configuring Mutual SSL Between Identity Governance and Transformation Hub . . . . .  | 100        |
| Creating Fact Configuration Files . . . . .  | 104        |
| Mapping Attributes for Data Reconciliation . . . . .                                 | 104        |
| Collecting Data from Identity Governance . . . . .                                   | 106        |
| <b>18 Reverting Backdated Events Configuration</b>                                   | <b>107</b> |
| <b>19 Verifying Data Collection Configuration</b>                                    | <b>109</b> |
| Verifying SmartConnector Log Files . . . . .   | 109        |
| Viewing Data in the Identity Intelligence User Interface . . . . .                   | 109        |
| <b>Part IV Upgrading Identity Intelligence</b>                                       | <b>111</b> |
| <b>20 Upgrade Checklist</b>  | <b>113</b> |
| <b>21 Upgrading Identity Intelligence</b>  | <b>115</b> |
| Prerequisites . . . . .  | 115        |
| Upgrading Vertica . . . . .  | 115        |
| Upgrading CDF . . . . .  | 117        |

|   |            |
|---|------------|
| Manual Upgrade . . . . .  | 117        |
| Automated Upgrade . . . . .   | 120        |
| Upgrading SmartConnector . . . . .  | 122        |
| Upgrading Identity Intelligence . . . . .   | 122        |
| Post-Upgrade Configurations . . . . .   | 124        |
| <b>Part V Managing Identity Intelligence</b>                                      | <b>125</b> |
| <b>22 Installing Licenses</b>   | <b>127</b> |
| Installing the License for Identity Intelligence and Transformation Hub . . . . . | 127        |
| Installing the License for Vertica . . . . .                                      | 128        |
| Installing the License for ArcMC . . . . .  | 128        |
| <b>23 Assigning Permissions and Roles</b>   | <b>129</b> |
| Creating and Assigning Permissions to a Role . . . . .                            | 129        |
| Creating a User . . . . .   | 129        |
| <b>24 Adding Vertica Nodes</b>  | <b>131</b> |
| <b>25 Restarting Nodes in the Cluster</b>   | <b>133</b> |
| Restarting Nodes by Using Scripts . . . . .                                       | 133        |
| Restarting Nodes Manually . . . . .   | 133        |
| <b>26 Configuring the Log Level</b>   | <b>135</b> |
| <b>27 Collecting Diagnostic Logs</b>  | <b>137</b> |
| <b>28 Modifying Transformation Hub Configurations</b>                             | <b>139</b> |
| Disabling Plain Text Communication . . . . .                                      | 139        |
| Prerequisite . . . . .  | 139        |
| Disabling Plain Text Communication . . . . .                                      | 139        |
| Enabling Client Authentication . . . . .  | 140        |
| <b>29 Resetting the CDF Administrator Password</b>                                | <b>143</b> |
| <b>30 Renewing CDF Certificates</b>   | <b>145</b> |
| Renewing Certificate Before Expiration . . . . .                                  | 145        |
| Renewing Certificates After Expiration . . . . .                                  | 145        |
| <b>31 Changing the CDF Certificate Authority</b>                                  | <b>147</b> |
| Generating a New CA . . . . .   | 147        |
| Updating the CDF CA . . . . .   | 150        |
| <b>32 Retrieving CDF Root CA</b>  | <b>151</b> |
| Retrieving the CDF Root CA from Browser . . . . .                                 | 151        |
| Retrieving the CDF Root CA Using Command Line . . . . .                           | 151        |

|   |            |
|---|------------|
| <b>33 Managing Vertica</b>  | <b>153</b> |
| <b>Part VI Appendices</b>   | <b>155</b> |
| <b>A Troubleshooting</b>  | <b>157</b> |
| Recovering from Loss of Entity Data Being Collected from Identity Manager to Identity Intelligence. . . . . | 157        |
| Restarting the Node Fails with an Error . . . . .   | 157        |
| <b>B Uninstalling Identity Intelligence</b>   | <b>159</b> |
| Uninstalling Manually . . . . .   | 159        |
| Uninstalling Identity Intelligence . . . . .  | 159        |
| Uninstalling Vertica. . . . .   | 159        |
| Uninstalling CDF . . . . .  | 159        |
| Uninstalling by Using the Script . . . . .  | 160        |





# About This Book

The Administrator Guide provides information about deploying, configuring, and managing Identity Intelligence.

## Intended Audience

This book provides information for IT administrators who are responsible for managing the Identity Intelligence software and its environment. Usually, these individuals have experience configuring servers and identity management applications such as Micro Focus Identity Manager.

## Additional Documentation

The Identity Intelligence documentation library includes the following resources:

- ◆ *User Guide to Identity Intelligence*, which is embedded in the product to provide both contextual Help and conceptual information
- ◆ *Release Notes for Identity Intelligence*
- ◆ *System Requirements for Identity Intelligence*

For the most recent version of this guide and other Identity Intelligence documentation resources, visit the [documentation for Identity Intelligence](#).

## Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the [comment on this topic](#) link at the bottom of each page of the online documentation, or send an email to [Documentation-Feedback@microfocus.com](mailto:Documentation-Feedback@microfocus.com).

For specific product issues, contact Micro Focus Customer Care at <https://www.microfocus.com/support-and-services/>.



# 1 Introduction

Micro Focus Identity Intelligence provides interactive and reporting capabilities for identity governance data so you can perform the following types of activities:

- ◆ Provide data in the form of visuals and reports to support audits of identity governance processes
- ◆ Export data for analysis, as well as for reporting to management and other stakeholders such as compliance officers or resource administrator
- ◆ Look for possible issues or breaches in identity governance processes and protocols
- ◆ Evaluate request and approval processes to determine their efficiency and adherence to enterprise standards, such as service level agreements (SLAs)

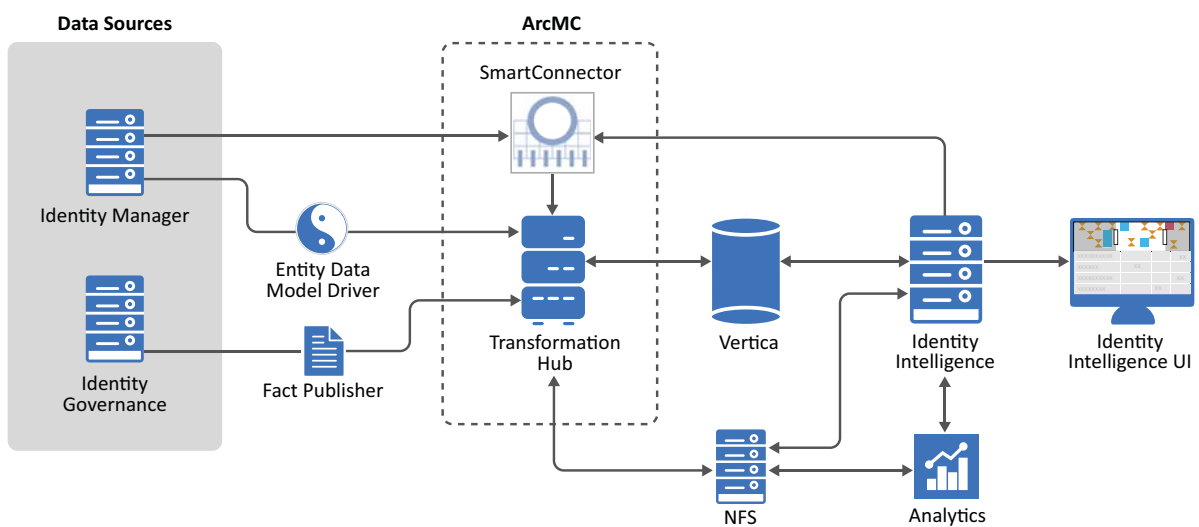
Identity Intelligence gathers data from sources such as Micro Focus Identity Manager and Micro Focus Identity Governance. It then sends the data to Micro Focus Transformation Hub for processing and to Micro Focus Vertica for storage. Depending on the data that you want to analyze, you can create a **View** and explore the **Profiles** of **users** and **access rights**.

- ◆ “Understanding the Identity Intelligence Architecture” on page 11
- ◆ “Understanding the Identity Intelligence Components” on page 12

## Understanding the Identity Intelligence Architecture

The Identity Intelligence environment incorporates several components that enable it to receive identity governance data, manage and store that data, and organize the data for user interaction. Identity Intelligence needs one or more data sources, such as Identity Governance, as well as mechanisms for collecting data from those sources. The following diagram and table help you understand the software and components that comprise your Identity Intelligence environment.

*Figure 1-1 Identity Intelligence Architecture*



# Understanding the Identity Intelligence Components

The following table describes the components incorporated into the Identity Intelligence environment.

| Software/Component                             | Description  |
|--|--|
| Data sources                                   | <p>Provide identity governance data to Identity Intelligence through drivers or special utilities</p> <p>For example, Identity Manager and Identity Governance</p>   |
| SmartConnector for Syslog NG Daemon            | <p>Collects the following types of data, and then sends the data to Transformation Hub for processing:</p> <ul style="list-style-type: none"> <li>◆ <a href="#">Entity change events</a> generated by Identity Intelligence, when there are changes to entity data in the data source</li> <li>◆ <a href="#">Audit events</a> generated in Identity Manager</li> </ul> |
| Entity Data Model Driver                       | Collects <a href="#">entity data</a> from Identity Manager, and then sends the data to Transformation Hub for processing   |
| Identity Governance Fact Configuration Utility | Collects data from Identity Governance   |
| Transformation Hub                             | Receives audit events and entity data from publishers such as SmartConnectors, driver, or IG Fact. It centralizes data collection and sends data to Vertica  |
| Vertica  | Stores Identity Intelligence data, which includes all collected events; user and entity profile content; users and permissions; and View content   |
| Analytics (framework)                          | Provides a high-capacity data management and search engine that enables you to search data stored in Vertica from the Identity Intelligence user interface   |
| Network File System (NFS)                      | Stores some of the persistent data generated by Transformation Hub, Identity Intelligence, and Analytics   |
| Identity Intelligence                          | Standardizes and processes data from Vertica and stores the processed data in Vertica.   |
| Identity Intelligence User Interface (UI)      | <p>Provides a browser-based console where you can create Views and review the Profiles of users and access rights</p> <p>For more information, see the User Guide (also available in the UI)</p>   |
| ArcSight Management Center (ArcMC)             | <p><i>(Optional)</i></p> <p>A centralized management interface that helps you to effectively administrate and monitor software and components such as Transformation Hub and SmartConnectors</p>   |

# Planning the Installation

This section provides the necessary information to plan your Identity Intelligence installation.

- ◆ [Chapter 2, “Implementation Checklist,” on page 15](#)
- ◆ [Chapter 3, “Deployment Considerations,” on page 17](#)
- ◆ [Chapter 4, “Installation Options,” on page 23](#)



# 2 Implementation Checklist

Use the following checklist to install and configure Identity Intelligence. You should perform the tasks in the listed order.

| Task   | See   |
|--|---|
| <input type="checkbox"/> 1. Review product architecture information to learn about the software and components that you need to install and configure                | <a href="#">Deployment Considerations</a>                     |
| <input type="checkbox"/> 2. Ensure that the computers on which you are installing the Identity Intelligence components meet the specified requirements               | <a href="#">Identity Intelligence 1.1 System Requirements</a> |
| <input type="checkbox"/> 3. Decide the deployment type, and how you want to configure your component installation  | <a href="#">Installation Options</a>                          |
| <input type="checkbox"/> 4. Review the considerations and prerequisites for installation   | <a href="#">Deployment Options</a>                            |
| <input type="checkbox"/> 5. Decide the security mode (one-way or mutual SSL) that must be used for communication between components                                  | <a href="#">Securing Communication Between Components</a>     |
| <input type="checkbox"/> 6. Install Identity Intelligence  | <a href="#">Installing Identity Intelligence</a>              |
| <input type="checkbox"/> 7. (Conditional) Perform the post-installation configuration tasks<br><br><i>Required only when you do not use the installation scripts</i> | <a href="#">Post-Installation Configurations</a>              |
| <input type="checkbox"/> 8. Configure data collection  | <a href="#">Configuring Data Collection</a>                   |
| <input type="checkbox"/> 9. Add users and assign their permissions and roles   | <a href="#">Assigning Permissions and Roles</a>               |
| <input type="checkbox"/> 10. Install licenses before the trial period expires  | <a href="#">Installing Licenses</a>                           |
| <input type="checkbox"/> 11. (Optional) Install ArcMC  | <a href="#">Installing and Configuring ArcMC</a>              |





# 3 Deployment Considerations

To deliver powerful and hardened capabilities, Identity Intelligence includes certain components from Micro Focus ArcSight suite of products, such as Transformation Hub and SmartConnectors. Identity Intelligence and Transformation Hub are containerized applications that are based on Container Deployment Foundation (CDF). CDF is a container-based delivery and management model built on Kubernetes and Docker Containers. Basically, you install CDF. Then you use CDF to deploy and manage the container-based products, specifically Identity Intelligence and Transformation Hub. You also need to install supporting components, such as a driver to collect data from Identity Manager.

- ◆ [“Components Needed for Deployment” on page 17](#)
- ◆ [“Deployment Options” on page 17](#)
- ◆ [“Securing Communication Between Components” on page 18](#)

## Components Needed for Deployment

The installation process deploys the following components:

- ◆ Analytics framework
- ◆ Identity Intelligence framework
- ◆ Transformation Hub
- ◆ Vertica

Depending on your Identity Intelligence environment, you will need to install and configure one or more of the following components:

- ◆ SmartConnector for Syslog NG Daemon
- ◆ Entity Data Model Driver - for collecting data from Identity Manager
- ◆ Identity Governance Fact Configuration Utility - for collecting data from Identity Governance
- ◆ (Optional) ArcSight Management Center (ArcMC) - for managing Transformation Hub and SmartConnectors

## Deployment Options

You can choose to deploy in a single-node or multi-node environment, depending on your anticipated workload and whether you need high availability. For more information about deployment sizing and tuning, see [Hardware Requirements and Tuning Guidelines](#).

If you already have an existing cluster for Transformation Hub or ArcSight Investigate, you can deploy Identity Intelligence to the same cluster. Reusing existing clusters would reduce costs and system management effort compared to deploying these software in a new cluster.

- ◆ [“Single-node Deployment” on page 18](#)
- ◆ [“Multi-node Deployment” on page 18](#)
- ◆ [“High Availability Deployment” on page 18](#)

## Single-node Deployment

In a [single-node deployment](#), you deploy all of the Identity Intelligence components on a single node. This method of deployment is suitable only for small workloads and where you do not need high availability.

## Multi-node Deployment

For larger workloads, you must deploy Identity Intelligence and the required software in a multi-node cluster setup. Multi-node deployment does load balancing across several worker nodes and is scalable to handle large workloads. You can add multiple master nodes and worker nodes to scale. While you can add worker nodes even after the installation, you can add master nodes only during the installation. Therefore, plan your deployment before you start the installation process.

## High Availability Deployment

To avoid single point failures and reduce downtime, you should ensure that your deployment is highly available.

For high availability deployment, you must set up three master nodes and at least two or more worker nodes depending on the workload. For information about the number of worker nodes for different workloads, see [System Requirements](#). Three master nodes are required to ensure that even in cases where two master nodes are unavailable, there is still another master node available. If only two master nodes are used and the primary master node is taken offline for maintenance or upgrade, there will only be a single master node available creating a single point of failure. If the available single master node fails, the cluster stops and cluster orchestration will not be possible until the master is back online.

For high availability of Vertica, you must set up three Vertica nodes.

While you can add worker nodes even after the installation, you can add master nodes only during the installation. Therefore, plan your deployment before you start the installation process.

## Securing Communication Between Components

You need to determine the security mode you want for communication between infrastructure components and ensure that the security mode is same across all components. The security mode described here is applicable only if the components are container-based applications. To identify if the application is container-based, refer to the application's documentation.

Identity Intelligence does not support plain text communication. You must use one of the following security modes for communication between Identity Intelligence components:

- ◆ [“One-Way SSL Authentication” on page 19](#)
- ◆ [“Mutual SSL Authentication” on page 19](#)

For information about the security modes supported between Identity Intelligence components, see [Security Modes Between Components](#).

# One-Way SSL Authentication

By default, Identity Intelligence installed using scripts support one-way (server) SSL authentication between Identity Intelligence and its related components.

In one-way SSL authentication, the client authenticates the server to ensure that it is communicating with a trusted server. For the client to trust the server, the client's trust store must have the server's Certificate Authority (CA).

In Identity Intelligence, Transformation Hub acts as an SSL server and all the components communicating with Transformation Hub act as SSL clients. Therefore, you must retrieve the CDF root CA certificate which is used by Transformation Hub and add the certificate to the truststore of all Transformation Hub clients, such as Identity Governance, SmartConnector, Identity Manager Driver for Entity Data Model, and Vertica. To identify Transformation Hub's clients, see [Identity Intelligence Architecture diagram](#).

---

**NOTE:** By default, Vertica must always be configured to use mutual SSL authentication in the Identity Intelligence environment.

---

If you plan to use one-way SSL authentication, you must:

- ◆ Decide the CA to be used as the CDF root CA. You can use one of the following:
  - ◆ Self-signed CA that is generated during the installation of CDF by default. For instructions to retrieve the default CA, see [Retrieving CDF Root CA](#).
  - ◆ Replace the default CA with a well-known CA, your organization's CA or newly generated root CA. For instructions to change the CDF CA, see [Changing the CDF Certificate Authority](#).
- ◆ Get the root CA certificate for configuring mutual SSL in Vertica.
  - ◆ If you are installing by using scripts, root CA generation and SSL configuration for Vertica will be automatically done by the script.
  - ◆ If you are installing manually, you can do one of the following:
    - ◆ Generate a new root CA certificate for Vertica  
For instructions to generate a root CA, see [Generating a New CA](#) section.
    - ◆ Use any well-known CA or your organization's root CA
    - ◆ Reuse the CDF root CA for Vertica if you are planning to generate a new root CA for CDF
- ◆ Add the CDF root CA certificate to the truststore of all Transformation Hub clients. The instructions to add certificate to truststore are available in the respective client configuration sections.

# Mutual SSL Authentication

To enhance security, you can configure mutual (two-way or client) SSL between Identity Intelligence and some of the components.

In mutual SSL authentication, the client and the server authenticate each other to ensure that both parties involved in the communication are trusted. For the client to trust the server, the client's trust store must have the server's CA certificate. For the server to trust the client, the client's certificate must be signed by the server's CA, which is already trusted by the server.

In Identity Intelligence, Transformation Hub acts as an SSL server and all the components communicating with Transformation Hub act as SSL clients. Therefore, you must:

- ◆ Add the CDF root CA which is used by Transformation Hub to the truststore of all its clients, such as Identity Governance, SmartConnector, Identity Manager Driver for Entity Data Model, and Vertica. This ensures that all the clients trust the Transformation Hub Server.
- ◆ Add the client's private key and public certificate, which is signed by the CDF root CA to the keystore of all Transformation Hub's clients. This ensures that the Transformation Hub Server trusts all its clients.

To identify Transformation Hub's clients, see [Identity Intelligence Architecture diagram](#).

If you plan to use mutual SSL authentication, you must:

- ◆ Enable client authentication in Transformation Hub either during [manual installation](#) or [after installation](#).

- ◆ Decide the CA to be used as the CDF root CA.

You cannot use the default self-signed CA that is generated during CDF installation.

You must replace the default CDF CA with a well-known CA, your organization's CA or newly generated root CA. For instructions to change CDF root CA, see [Changing the CDF Certificate Authority](#).

- ◆ Get the root CA certificate for configuring mutual SSL in Vertica. You can do one of the following:
  - ◆ Generate a new root CA certificate for Vertica  
For instructions to generate a root CA, see [Generating a New CA](#) section.
  - ◆ Use any well-known CA or organization's root CA
  - ◆ Reuse the CDF root CA for Vertica if you are planning to generate a new root CA for CDF
- ◆ Configure mutual SSL authentication in all the clients of Transformation Hub. For instructions, see the respective client configuration sections.

## Security Modes Between Components

The following tables list Identity Intelligence components, supported security modes, and where to find more information on configuring the security mode.

**Table 3-1** *Securing Communication Between Identity Intelligence Core Components*

| Communication                           | Supported security modes  | More information   |
|---|---|--|
| Identity Intelligence to Vertica        | One-way SSL   | <a href="#">Configuring SSL for Vertica</a>  |
| Transformation Hub to Vertica           | Mutual SSL  | <a href="#">Creating a Kafka Scheduler with SSL</a>  |
| Vertica to Transformation Hub           | <ul style="list-style-type: none"> <li>◆ One-way SSL</li> <li>◆ Mutual SSL</li> </ul> | Enabled by default   |
| Identity Intelligence to SmartConnector | NA<br>(Internal communication)  | Install SmartConnector on the worker node with label <code>analytics:yes</code> , where the Identity Intelligence service is running, so that the communication is within the same node. |

| Communication  | Supported security modes  | More information   |
|--|---|--|
| Identity Intelligence to Analytics                                     | NA<br>(Internal communication)  | Identity Intelligence and Analytics run on the worker node with label <code>analytics:yes</code> , so that the communication is within the same node.      |
| ArcMC to Transformation Hub  | <ul style="list-style-type: none"> <li>◆ One-way SSL</li> <li>◆ Mutual SSL</li> </ul> | Install ArcMC before installing Transformation Hub.<br><br>ArcMC Administrator's Guide   |
| Transformation Hub, Analytics, and Identity Intelligence to NFS Server |   | For optimal security, secure all NFS settings to allow only required hosts to connect to the NFS server.<br><br><a href="#">Configuring the NFS Server</a> |
| Web browser to NGINX (proxy)   | One-way SSL   | Enabled by default   |

**Table 3-2** *Securing Communication Between Data Sources and Identity Intelligence*

| Communication   | Supported security modes  | More information  |
|---|---|---|
| SmartConnector to Transformation Hub                                | <ul style="list-style-type: none"> <li>◆ One-way SSL</li> <li>◆ Mutual SSL</li> </ul> | <a href="#">Configuring One-Way SSL between the SmartConnector and Transformation Hub</a><br><br><a href="#">Configuring Mutual SSL between the SmartConnector and Transformation Hub</a>   |
| Identity Manager to Identity Manager Driver for Entity Data Model   | One-way SSL   | <a href="#">Creating a Secure Connection to the Identity Manager Engine</a>   |
| Identity Manager Driver for Entity Data Model to Transformation Hub | <ul style="list-style-type: none"> <li>◆ One-way SSL</li> <li>◆ Mutual SSL</li> </ul> | <a href="#">Configuring One-Way SSL between the driver and Transformation Hub</a><br><br><a href="#">Configuring Mutual SSL between the driver and Transformation Hub</a>                   |
| Identity Manager to SmartConnector over Syslog                      | One-way SSL   | <a href="#">Configuring Audit Events Collection</a>   |
| Identity Governance to Transformation Hub                           | <ul style="list-style-type: none"> <li>◆ One-way SSL</li> <li>◆ Mutual SSL</li> </ul> | <a href="#">Configuring One-Way SSL Between Identity Governance and Transformation Hub</a><br><br><a href="#">Configuring Mutual SSL Between Identity Governance and Transformation Hub</a> |



# 4 Installation Options

You can install Identity Intelligence either by using the provided installation scripts or manually.

- ♦ [“Installation Using Scripts” on page 23](#)
- ♦ [“Manual Installation” on page 23](#)
- ♦ [“Deciding to Use the Scripts or Manual Installation Method” on page 23](#)

## Installation Using Scripts

To enable an easier installation, Identity Intelligence provides scripts that automatically take care of all the prerequisites, software installations, and post-installation configurations. The scripts are applicable for [single-node deployments](#) where high availability is not needed. However, if you prefer to manually set the configurations and the installations because of your organization’s security policies, you can install Identity Intelligence manually in single-node deployments as well. The scripts configure the system to match the settings described for performing a [manual installation](#).

The installation scripts expect your environment to be in a specific state. Before deciding to use the installation scripts, review the [considerations for installation](#).

For information about installing Identity Intelligence by using scripts for a single-node deployment, see [Chapter 5, “Installing Identity Intelligence by Using Scripts,” on page 27](#).

## Manual Installation

In deployments with a larger workload where high availability is mandatory, you must manually perform all the necessary system configurations and software installations. However, you can use some of the installation scripts to make your tasks easier, then complete the rest of the configurations and installations manually.

For information about installing Identity Intelligence manually, see [Chapter 6, “Installing Identity Intelligence Manually,” on page 29](#).

## Deciding to Use the Scripts or Manual Installation Method

To determine whether to use the installation scripts or perform a manual installation, review the following considerations:

- ♦ The scripts install Identity Intelligence on the operating system with a [default minimum installation](#). If you have any customizations on the operating system, we recommend you to perform the prerequisites manually and perform installation and post installation configuration using scripts.
- ♦ The scripts install Identity Intelligence only on a singled-homed network (a single-homed stub system is one that is connected with a single network link). If you have a dual-homed network (dual or redundant connections to a single Internet Service Provider), we recommend that you use the [manual installation](#) process.

- ◆ The scripts automatically tune the system for a [single-node deployment](#) with a [small workload](#).
- ◆ The script configures Vertica agent to use the port 5445 instead of the default port 5444, as the script installs both CDF and Vertica on the same node.
- ◆ The scripts register a service with the operating system to automatically start the Vertica Kafka scheduler to collect event data.
- ◆ The scripts install the cluster with a single master node and single worker node running on the same system. You can add worker nodes after the installation to scale and enable worker high availability.
- ◆ If you use the scripts, you cannot configure high availability for the master node. If you want high availability for the master node, we recommend that you use the [manual installation](#) process.
- ◆ The scripts disable the option to authorize Micro Focus to collect suite usage data.
- ◆ The scripts create NFS shares on the system used by the containers in the cluster. They configure the firewall to disable remote access to this NFS server. If you plan to add additional nodes to the cluster, you must enable remote access to the NFS server in the firewall.
- ◆ The scripts use the following paths by default:
  - ◆ To install Kubernetes: `/opt/kubernetes`
  - ◆ To create NFS shared directories: `/opt/arcsight/nfs/volumes`
  - ◆ To unzip Vertica installer file: `/opt/arcsight-vertica`
  - ◆ To install Vertica: `/opt/vertica`
- ◆ If you must use proxy in your environment, you must use the [manual installation](#) process.
- ◆ If your network is already using the subnets defined for the [default CDF subnets](#), we recommend using the [manual installation](#) process. In this way, you can configure CDF to use a different subnet.





# Installing Identity Intelligence

This section provides information about installing and configuring Identity Intelligence.

- ◆ [Chapter 5, “Installing Identity Intelligence by Using Scripts,” on page 27](#)
- ◆ [Chapter 6, “Installing Identity Intelligence Manually,” on page 29](#)
- ◆ [Chapter 7, “Deploying Identity Intelligence in an Existing Cluster,” on page 55](#)
- ◆ [Chapter 8, “Post-Installation Configurations,” on page 57](#)
- ◆ [Chapter 9, “Verifying the Installation,” on page 65](#)
- ◆ [Chapter 10, “Installing and Configuring ArcMC,” on page 67](#)



# 5 Installing Identity Intelligence by Using Scripts

You can use the [installation scripts](#) in [single-node deployments](#) for end-to-end installation starting from configuring prerequisites to completing post-installation configurations. In [multi-node deployments](#), you can use the scripts only for some specific prerequisites and post-installation configurations.

- ◆ [“Prerequisites” on page 27](#)
- ◆ [“Understanding the Installation Scripts” on page 27](#)
- ◆ [“Using the Scripts in Single-node Deployments” on page 27](#)
- ◆ [“Using the Scripts in Multi-node Deployments” on page 28](#)

## Prerequisites

Ensure the system requirements mentioned in [Identity Intelligence 1.1 System Requirements](#) are met.

## Understanding the Installation Scripts

The installation scripts automatically take care of all the prerequisites, software installations, and post-installation configurations:

| Script   | Purpose   |
|--|---|
| <code>./prepare-install-single-node-host.sh</code> | Installs all the necessary packages and configures the prerequisites.                       |
| <code>./install-single-node.sh</code>              | Installs Vertica, CDF, Transformation Hub, and Identity Intelligence.                       |
| <code>./install-single-node-post.sh</code>         | Performs post-installation configurations, such as labeling the nodes and scheduling Kafka. |

## Using the Scripts in Single-node Deployments

*Applies only when your deployment does not need high availability*

The [installation scripts](#) automatically take care of all the prerequisites, software installations, and post-installation configurations. For deployments with a small workload, the script sets the appropriate configuration settings for Vertica. For medium and large workloads, you must manually adjust the configuration settings after the installation is complete.

**IMPORTANT:** For Identity Intelligence 1.1.1 and later, the scripts disable plain text communication between Transformation Hub (Kafka) and all the components outside the Kubernetes cluster, such as Identity Governance, Identity Manager Driver for Entity Data Model, Vertica, and so on.

Therefore, ensure to configure SSL between Transformation Hub (Kafka) and the components that are outside the Kubernetes cluster. The scripts automatically configures SSL for Vertica as Vertica is installed as part of the script.

---

### To install Identity Intelligence by using scripts:

1 Log in to the master node as `root`.

2 Change to the directory where you downloaded the Identity Intelligence installer files.

```
cd /opt
```

For information about downloading the Identity Intelligence installer files, see [Downloading Identity Intelligence](#).

3 Execute the scripts in the following order:

**3a** `./prepare-install-single-node-host.sh`

**3b** `./install-single-node.sh`

**3c** `./install-single-node-post.sh`

4 (Conditional) For deployments with medium and large workloads, complete the following steps to modify the Vertica resource pool settings:

**4a** Log in to the Vertica node.

**4b** Change to the following directory:

```
cd /opt/arcsight-vertica/scripts
```

**4c** Execute the following command with the appropriate values:

```
tuning_util.sh <parameter_1> <value> <parameter_2> <value>
```

For example:

```
tuning_util.sh -m 2048 -c 3 -x 4
```

To see a list of tuning parameters, use the following command:

```
tuning_util.sh -h
```

For more information about the tuning values for your deployment, see [Hardware Requirements and Tuning Guidelines](#).

5 (Conditional) By default, Vertica is installed with community edition. If you want to upgrade to enterprise edition, perform steps in the [Installing the License for Vertica](#).

6 (Conditional) If you want to use [mutual SSL authentication](#) between Transformation Hub and its clients, perform steps in the [Enabling Client Authentication](#) section.

7 Continue with [Configuring Data Collection](#).

## Using the Scripts in Multi-node Deployments

For multi-node deployments, you can use some of the scripts to make your tasks easier, and complete the rest of the configurations and installations manually. For more information, see [Chapter 6, “Installing Identity Intelligence Manually,” on page 29](#).

# 6 Installing Identity Intelligence Manually

This chapter provides information about installing Identity Intelligence and the required software.

- ♦ [“Installing Vertica” on page 29](#)
- ♦ [“Preparing Your Environment for CDF” on page 34](#)
- ♦ [“Installing CDF” on page 49](#)
- ♦ [“Installing Identity Intelligence” on page 50](#)

## Installing Vertica

This section provides information about installing and configuring Vertica.

- ♦ [“Prerequisites” on page 29](#)
- ♦ [“Installing Vertica” on page 33](#)

## Prerequisites

To complete the prerequisites, see the following sections:

- ♦ [“Configuring the Vertica Node” on page 29](#)
- ♦ [“Enabling Password-less SSH Access” on page 31](#)
- ♦ [“Setting Vertica Properties” on page 32](#)

## Configuring the Vertica Node

- 1 Provision the Vertica server with at least 3 GB of swap space.

---

**NOTE:** Identity Intelligence supports using Vertica on a host with a Linux Logical Volume Manager (LVM) formatted disk.

---

- 2 Register a service with the operating system to start the Vertica Kafka scheduler automatically when the operating system starts. You can register a service manually or by using the `/opt/<Identity_Intelligence_Installer>/scripts/postinstall_create_kafka_scheduler_svc.sh` script.
- 3 Add the following parameters to `/etc/sysctl.conf` and reboot the server for the changes to take effect:

---

| Parameter                                 | Description  |
|---|--|
| <code>net.core.somaxconn = 1024</code>    | Increases the number of incoming connections         |
| <code>net.core.wmem_max = 16777216</code> | Sets the send socket buffer maximum size in bytes    |
| <code>net.core.rmem_max = 16777216</code> | Sets the receive socket buffer maximum size in bytes |

---

| Parameter  | Description  |
|--|--|
| <code>net.core.wmem_default = 262144</code>                | Sets the receive socket buffer default size in bytes   |
| <code>net.core.rmem_default = 262144</code>                | Controls the default size of receive buffers used by sockets   |
| <code>net.core.netdev_max_backlog = 100000</code>          | Increases the length of the processor input queue  |
| <code>net.ipv4.tcp_mem = 16777216 16777216 16777216</code> |  |
| <code>net.ipv4.tcp_wmem = 8192 262144 8388608</code>       |  |
| <code>net.ipv4.tcp_rmem = 8192 262144 8388608</code>       |  |
| <code>net.ipv4.udp_mem = 16777216 16777216 16777216</code> |  |
| <code>net.ipv4.udp_rmem_min = 16384</code>                 |  |
| <code>net.ipv4.udp_wmem_min = 16384</code>                 |  |
| <code>net.ipv4.tcp_max_syn_backlog = 4096</code>           | Increases the allowed number of outstanding syn requests   |
| <code>dirty_ratio = 8</code>                               | Sets the number of pages at which a process that generates disk writes starts writing out dirty data<br><br>For more information, see <a href="#">Tuning Linux Dirty Data Parameters for Vertica</a> . |
| <code>vm.swappiness = 1</code>                             | Defines the amount and frequency at which the kernel copies RAM contents to a swap space<br><br>For more information, see <a href="#">Check for Swappiness</a> .                                       |

- 4 Add the following parameters to `/etc/rc.local`. You must reboot the server for the changes to take effect.

| Parameter   | Description   |
|---|---|
| <code>echo 'echo &lt;scheduler_value&gt; &gt; /sys/block/sda/queue/scheduler' &gt;&gt; /etc/rc.local</code> | Changes I/O scheduling to a supported scheduler<br><br>For more information, see <a href="#">I/O Scheduling</a> . |
| <code>chmod +x /etc/rc.local</code>   |   |

- 5 To increase the process limit, add the following parameters including `*` in the `/etc/security/limits.d/20-nproc.conf`:

```
* soft nproc 10240
* hard nproc 10240
* soft nofile 65536
* hard nofile 65536
* soft core unlimited
* hard core unlimited
```

- 6 In `/etc/default/grub`, append line `GRUB_CMDLINE_LINUX` with `intel_idle.max_cstate=0 processor.max_cstate=1`.

Example:

```
GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto
vconsole.font=latarcyrheb-sun16 rhgb quiet intel_idle.max_cstate=0
processor.max_cstate=1"
```

- 7 Execute the command `grub2-mkconfig -o /boot/grub2/grub.cfg` to update the configuration changes.
- 8 If you have firewall configured in the Vertica node, ensure to open the [Vertica ports](#) in the firewall.

- 9 Disable the firewall:

```
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X
systemctl mask firewalld
systemctl disable firewalld
systemctl stop firewalld
```

You can enable the firewall after the installation. For more information, see [Firewall Considerations](#).

- 10 Set SELinux to permissive mode:

```
vi /etc/selinux/config
SELINUX=permissive
```

For more information, see [SELinux Configuration](#).

- 11 Configure the BIOS for maximum performance:

**System Configuration > BIOS/Platform Configuration (RBSU) > Power Management > HPE Power Profile > Maximum Performance**

- 12 Reboot the server and use the `ulimit -a` command to verify that the limits have increased.

## Enabling Password-less SSH Access

Before you install the Vertica database, generate a key pair on node 1 and then copy the public key to all nodes in the cluster, including node 1. This enables password-less SSH access from the node 1 server to all of the other node servers in the cluster.

---

**NOTE:** You must repeat the authentication process for each node in the cluster.

---

- 1 Log in to Vertica cluster node 1 server.

- 2 Run the command:

```
ssh-keygen -q -t rsa
```

- 3 Copy the key from node 1 to all of the nodes, including node 1, using the node IP address:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@11.111.111.111
```

The system displays the key fingerprint and requests to authenticate the node server.

- 4 Specify the credentials for the node.

The operation is successful when the system displays the following message:

Number of key(s) added: 1

- 5 To verify successful key installation, run the following command from node 1 to the target node to verify that node 1 can successfully log in:

```
ssh root@11.111.111.111
```

## Setting Vertica Properties

Before installing Vertica, you must set the values for various Vertica properties based on your deployment size.

**To set Vertica properties, perform the following:**

- 1 Log in to the Vertica cluster node 1 server.
- 2 Change to the directory where you want to install Vertica:

```
cd <vertica_installer_directory>
```

For example:

```
cd /opt
```

- 3 Create a folder for Vertica database installer script:

```
mkdir arcsight-vertica
```

- 4 Copy the downloaded Vertica installer file to arcsight-vertica:

```
cp <download_directory>/identityintelligence-x.x.x.x/installers/arcsight-vertica-installer_x.x.x.tar.gz /opt/arcsight-vertica
```

For information about downloading the Vertica installer file, see [Downloading Identity Intelligence](#).

- 5 Unzip the copied file using the command:

```
tar xvfz arcsight-vertica-installer_x.x.x.tar.gz
```

- 6 (Conditional) If you plan to deploy Vertica and Identity Intelligence on the same node, set the values for the following properties. For information about the values that must be set, see [Hardware Requirements and Tuning Guidelines](#)

| File                             | Properties  |
|----------------------------------|---|
| vertica.properties               | <ul style="list-style-type: none"><li>◆ tm_concurrency</li><li>◆ tm_memory</li><li>◆ active_partition</li></ul>   |
| config/resource_pools.properties | <ul style="list-style-type: none"><li>◆ ingest_pool_memory_size</li><li>◆ mf_entity_ingest_pool_memory_size</li><li>◆ mf_entity_ingest_pool_planned_concurrency</li></ul> |
| config/sched.properties          | <ul style="list-style-type: none"><li>◆ plannedconcurrency</li><li>◆ maxconcurrency</li><li>◆ tm_memory_usage</li></ul>   |

- 7 (Conditional) If you plan to deploy Vertica on an independent node, retain the default Vertica resource tuning values.



- 8 In the `config/vertica_user.properties` file, set the values as explained in the following table:

| Property                      | Description  |
|-------------------------------|--|
| <code>hosts</code>            | <p>Specify the IP address of the Vertica node.</p> <p>If you want to install Vertica on multiple nodes, provide a comma-separated list of node IP addresses in IPv4 format (for example, 1.1.1.1, 1.1.1.2, 1.1.1.3).</p> <p>For high availability, you must set up minimum 3 Vertica nodes. However, if you have modified the value of K-safe, ensure that you set up the required number of nodes. For more information, see <a href="#">K-Safety</a>.</p> <p>If you plan to expand the cluster by adding nodes after installation, avoid using loopback address (localhost, 127.0.0.1, etc.) and specify the IP address or hostname of the node.</p> |
| <code>license</code>          | <p>Vertica is installed with community edition when there is no license file.</p> <p>To upgrade to enterprise edition, download the license file from the <a href="#">Micro Focus Customer Center</a> portal and edit this parameter to point to the license file. This installs the Vertica license during installation.</p> <p>You can also install the license anytime during the trial period. For more information, see <a href="#">"Installing the License for Vertica" on page 128</a>.</p>   |
| <code>db_retention_day</code> | Used for the data retention policy.  |

## Installing Vertica

To install Vertica, complete the following steps.

- 1 Log in to Vertica cluster node 1 server.

- 2 Change to the directory where you have the Vertica installation package.

```
cd <vertica_installer_directory>/arcsight-vertica
```

- 3 Install Vertica using the command:

```
./vertica_installer install
```

- 4 Execute the following firewall commands to complete the installation successfully:

```
systemctl unmask firewalld
systemctl start firewalld
systemctl enable firewalld
```

- 5 When prompted, create the database administrator user and the search user.

You will need the database administrator credentials to access the Vertica database host. You will need the search user credentials when you configure Vertica in CDF.

- 6 Create the schema:

```
./vertica_installer create-schema
```

- 7 Set the database lock time out value as 600 by using the command:

```
runuser -l <db_admin_username> -c "vsq1 -w \"<db_admin_password>\" -c \"ALTER DATABASE investigate set LockTimeout = 600\""
```

- 8 (Conditional) If you plan to have both Vertica and CDF on the same node, you must configure the Vertica agent to use a different port.

To change the Vertica agent port:

**8a** Change the Vertica agent port using the command:

```
sed -i 's/^agent_port = 5444/agent_port = <new_port>/g' "/opt/vertica/agent/config.py"
```

Example:

```
sed -i 's/^agent_port = 5444/agent_port = 5445/g' "/opt/vertica/agent/config.py"
```

**8b** Restart Vertica agent by using the command:

```
systemctl restart vertica_agent
```

## Preparing Your Environment for CDF

The procedures in this section enable you to configure your environment for a successful installation of CDF.

- ◆ [“Configuring the Nodes” on page 34](#)
- ◆ [“Set System Parameters \(Network Bridging\)” on page 35](#)
- ◆ [“Check MAC and Cipher Algorithms” on page 36](#)
- ◆ [“Check Password Authentication Settings” on page 36](#)
- ◆ [“Installing the Required Operating System Packages” on page 36](#)
- ◆ [“Remove Libraries” on page 37](#)
- ◆ [“Configuring Time Synchronization” on page 37](#)
- ◆ [“Configuring Firewall” on page 38](#)
- ◆ [“Configuring Proxy” on page 39](#)
- ◆ [“Configuring DNS” on page 39](#)
- ◆ [“Configuring the NFS Server” on page 42](#)
- ◆ [“Disabling Swap Space” on page 45](#)
- ◆ [“\(Optional\) Create Docker Thinpools” on page 45](#)
- ◆ [“Enabling Installation Permissions for a sudo User” on page 47](#)

## Configuring the Nodes

For multi-node deployment, consider the following when configuring master and worker nodes:

- ◆ Deploy master and worker nodes on virtual machines. Since most of the processing occurs on worker nodes, we recommend you to deploy worker nodes on physical servers.
- ◆ You must keep the host system configuration identical across master and worker nodes.
- ◆ When using virtual machines, ensure:
  - ◆ Resources are reserved and not shared.
  - ◆ UUID and MAC addresses are static because dynamic addresses cause the Kubernetes cluster to fail.
- ◆ Install all master and worker nodes in the same subnet.

- ♦ Adding more worker nodes is typically more effective than installing bigger and faster hardware. Using more worker nodes enables you to perform maintenance on your cluster nodes with minimal impact to uptime. Adding more nodes also helps with predicting costs due to new hardware.

For high availability, consider the following when configuring [master](#) and [worker](#) nodes:

- ♦ Create a virtual IP that is shared by all master nodes and ensure that virtual IP is under the same subnet. The VIP must not respond when pinged before you install Identity Intelligence.
- ♦ All master and worker nodes must be installed in the same subnet.

## Set System Parameters (Network Bridging)

Ensure that the `br_netfilter` module is installed on all master and worker nodes before changing system settings.

You can either run the following scripts that set system parameters automatically or you can set the system parameters manually:

- ♦ `/opt/<Identity_Intelligence_Installer>/scripts/prereq_sysctl_conf.sh`
- ♦ `/opt/<Identity_Intelligence_Installer>/scripts/prereq_rc_local.sh`

Perform the following steps on all the master and worker nodes to set the system parameters manually:

- 1 Log in to the node.
- 2 Check whether the `br_netfilter` module is enabled:
 

```
lsmod |grep br_netfilter
```
- 3 If there is no return value and the `br_netfilter` module is not installed, then install it:
 

```
modprobe br_netfilter
echo "br_netfilter" > /etc/modules-load.d/br_netfilter.conf
```
- 4 Open the `/etc/sysctl.conf` file.
- 5 Ensure the following system parameters are set:
 

```
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
net.ipv4.ip_forward = 1
net.ipv4.tcp_tw_recycle = 0
kernel.sem=50100 128256000 50100 2560
```
- 6 Save the `/etc/sysctl.conf` file.
- 7 Apply the updates to the node:
 

```
/sbin/sysctl -p
```

## Check MAC and Cipher Algorithms

To configure MAC and Cipher algorithms manually, ensure the `/etc/ssh/sshd_config` files on each and every master and worker nodes are configured with at least one of the following values, which lists all supported algorithms. Add only the algorithms that meet the security policy of your organization.

- ♦ For MAC algorithms: `hmac-sha1,hmac-sha2-256,hmac-sha2-512,hmac-sha1-96`
- ♦ For Cipher algorithms: `3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr,arcfour128,arcfour256,blowfish-cbc`

For example, you could add the following lines to the `/etc/ssh/sshd_config` file on all master and worker nodes:

```
MACs hmac-sha2-256,hmac-sha2-512
```

```
Ciphers aes128-cbc,aes192-cbc,aes256-cbc,aes128-ctr,aes192-ctr,aes256-ctr
```

## Check Password Authentication Settings

If you will use a user name and password authentication for adding cluster nodes during the installation, make sure the `PasswordAuthentication` parameter in the `/etc/ssh/sshd_config` file is set to `yes`. There is no need to check the password authentication setting when you add cluster nodes using a user name and key authentication.

**To ensure the password authentication is enabled, perform the following steps on every master and worker nodes:**

- 1 Log in to the cluster node.
- 2 Open the `/etc/ssh/sshd_config` file.
- 3 Check if the parameter `PasswordAuthentication` is set to `yes`. If not, set the parameter to `yes` as below:

```
PasswordAuthentication yes
```

- 4 Restart the `sshd` service:

```
systemctl restart sshd.service
```

## Installing the Required Operating System Packages

Ensure that the packages listed in the following table are installed on appropriate nodes. These packages are available in the standard `yum` repository.

| Package                         | Nodes             |
|---------------------------------|-------------------|
| <code>bind-utils</code>         | Master and worker |
| <code>device-mapper-libs</code> | Master and worker |
| <code>java-1.8.0-openjdk</code> | Master            |
| <code>libcrypt</code>           | Master and worker |
| <code>libseccomp</code>         | Master and worker |
| <code>libtool-ltdl</code>       | Master and worker |

| Package                       | Nodes                                    |
|-------------------------------|--|
| net-tools                     | Master and worker                        |
| nfs-utils                     | Master and worker                        |
| rpcbind                       | Master node, worker node, and NFS server |
| systemd-libs (version >= 219) | Master and worker                        |
| unzip                         | Master and worker                        |
| httpd-tools                   | Master and worker                        |
| conntrack-tools               | Master and worker                        |
| lvm2                          | Master and worker                        |
| curl                          | Master and worker                        |
| libtool-libs                  | Master and worker                        |
| openssl                       | Master and worker                        |
| socat                         | Master and worker                        |
| container-selinux             | Master and worker                        |

You can either run the `/opt/<Identity_Intelligence_Installer>/scripts/prereq_1_required_packages.sh` script that installs the required OS packages automatically or install the required OS packages manually.

#### To install the packages manually:

- 1 Log in to the master or worker nodes.
- 2 Verify whether the package exists:
 

```
yum list installed <package name>
```
- 3 (Conditional) If the package is not installed, install the required package:
 

```
yum -y install <package name>
```

## Remove Libraries

Remove libraries that prevents Ingress from starting and confirm the removal when prompted:

```
yum remove rsh rsh-server vsftpd
```

## Configuring Time Synchronization

You must implement a Network Time Protocol (NTP) to synchronize time of all nodes in the cluster. To implement this protocol, use chrony. Ensure that chrony is running on all nodes of the cluster. By default chrony is installed on some versions of RHEL.

You can either run the `/opt/<Identity_Intelligence_Installer>/scripts/prereq_synchronize_time.sh` script that synchronizes time automatically or configure the time synchronization manually.

### To configure the time synchronization manually:

- 1 Verify chrony configuration:  
`chronyc tracking`
- 2 (Conditional) If chrony is not installed, install chrony:  
`yum install chrony`
- 3 Start and enable chrony:  
`systemctl start chronyd`  
`systemctl enable chronyd`
- 4 Synchronize the operating system time with the NTP server:  
`chronyc makestep`
- 5 Restart the chrony daemon:  
`systemctl restart chronyd`
- 6 Check the server time synchronization:  
`timedatectl`
- 7 Synchronize the hardware time:  
`hwclock -w`

## Configuring Firewall

Ensure that the `firewalld.service` is enabled and running on all nodes. Execute the `systemctl status firewalld` command to check the firewall status.

To enable the firewall:

```
systemctl unmask firewalld
systemctl start firewalld
systemctl enable firewalld
```

You can either run the `/opt/<Identity_Intelligence_Installer>/scripts/prereq_firewall.sh` script that configures the firewall automatically or configure the firewall manually.

When the firewall is enabled, you must also enable the masquerade settings. To enable masquerade settings:

- 1 Verify whether the masquerade setting is enabled:  
`firewall-cmd --query-masquerade`  
If the command returns `yes`, then masquerade is enabled.  
If the command returns `no`, then masquerade is disabled.
- 2 (Conditional) If masquerade setting is not enabled, enable masquerade:  
`firewall-cmd --add-masquerade --permanent`  
`firewall-cmd --reload`

## Configuring Proxy

The cluster should have no access to the Internet and proxy settings (`http_proxy`, `https_proxy`, and `no_proxy`) are not set. However, if a connection with the Internet is needed and you already specified a proxy server for http and https connection, you must correctly configure `no_proxy`.

If you have the `http_proxy` or `https_proxy` set, then `no_proxy` definitions must contain at least the following values:

```
no_proxy=localhost, 127.0.0.1, <all Master and Worker cluster node IP
addresses>, <all cluster node FQDNs>, <HA virtual IP Address>, <FQDN for the HA
Virtual IP address>
```

For example:

- ◆ 

```
export http_proxy="http://web-proxy.example.net:8080"
export https_proxy="http://web-proxy.example.net:8080"

export
no_proxy="localhost,127.0.0.1,node1.swinfra.net,10.94.235.231,node2.swinfra.net,10.94.235.232,node3.swinfra.net,10.94.235.233,node3.swinfra.net,10.94.235.233,node4.swinfra.net,10.94.235.234,node5.swinfra.net,10.94.235.235,node6.swinfra.net,10.94.235.236,ha.swinfra.net 10.94.235.200"
```
- ◆ 

```
export http_proxy="http://web-proxy.eu.example.net:8080"
export
https_proxy="localhost,127.0.0.1,swinfra.net,10.94.235.231,10.94.235.232,10.94.235.233,10.94.235.233,10.94.235.233,10.94.235.234,10.94.235.235,10.94.235.236,10.94.235.200"
```

---

**NOTE:** Incorrect configuration of proxy settings has proven to be a frequent installation troubleshooting problem. To verify that proxy settings are configured properly on all master and worker nodes, run the following command and ensure the output corresponds to the recommendations:

```
echo $http_proxy, $https_proxy, $no_proxy
```

---

If the firewall is turned off, the install process will generate a warning. To prevent the warning, the CDF install parameter `--auto-configure-firewall` should be set to true.

## Configuring DNS

Ensure that the host name resolution through Domain Name Services (DNS) is working across all nodes in the cluster, including correct forward and reverse DNS lookups. Host name resolution must not be performed through `/etc/hosts` file settings.

You can either run the `<download_directory>/scripts/prereq_disable_ipv6.sh` script that configures DNS automatically or configure DNS manually.

Ensure that all nodes are configured with a Fully Qualified Domain Name (FQDN) and are in the same subnet. Transformation Hub uses the host system FQDN as its Kafka `advertised.host.name`. If the FQDN resolves successfully in the Network Address Translation (NAT) environment, then Producers and consumers will function correctly. If there are network-specific issues resolving FQDN through NAT, then DNS will need to be updated to resolve these issues.

- ◆ Transformation Hub supports ingestion of event data that contains both IPv4 and IPv6 addresses. However, its infrastructure cannot be installed in an IPv6-only network.
- ◆ `localhost` must not resolve to an IPv6 address.

For example, Open the `/etc/host` file. Reference: `::1` – this is the default state. The install process expects only IPv4 resolution to IP address `127.0.0.1`.

Comment out any one of the following:

- ◆ `127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4`
- ◆ `::1 localhost localhost.localdomain localhost6 localhost6.localdomain6`
- ◆ The initial master node host name must not resolve to multiple IPv4 addresses and this includes lookup in `/etc/hosts`.
- ◆ [“Test Forward and Reverse DNS Lookup” on page 40](#)
- ◆ [“Kubernetes Network Subnet Settings” on page 41](#)

## Test Forward and Reverse DNS Lookup

Test that the forward and reverse lookup records for all servers were properly configured.

To test the forward lookup, run the following commands on every master and worker nodes in the cluster and on every producer and consumer host system, including:

- ◆ All master nodes: `master1.yourcompany.com, ..., mastern.yourcompany.com`
- ◆ All worker nodes: `worker1.yourcompany.com, ..., workern.yourcompany.com`
- ◆ Your ArcMC nodes: `arcmc1.yourcompany.com, ..., arcmcn.yourcompany.com`

Use the `nslookup` or `host` commands to verify your DNS configuration.

---

**NOTE:** Do not use the `ping` command.

---

You must run the `nslookup` commands on every server specified in your `/etc/resolv.conf` file. Every server must be able to forward and reverse lookup properly and return the exact same results.

If you have a public DNS server specified in your `/etc/resolv.conf` file, such as the Google public DNS servers `8.8.8.8` or `8.8.4.4`, you must remove this from your DNS configuration.

Run the commands as follows. Expected sample output is shown below each command.

- ◆ `hostname`  
`master1`
- ◆ `hostname -s`  
`master1`
- ◆ `hostname -f`  
`master1.yourcompany.com`



- ◆ `hostname -d`  

```
yourcompany.com
```
- ◆ `nslookup master1.yourcompany.com`  

```
Server: 192.168.0.53
Address: 192.168.0.53#53
Address: 192.168.0.1
Name: master1.example.com
```
- ◆ `nslookup master1`  

```
Server: 192.168.0.53
Address: 192.168.0.53#53
Name: master1.example.com
Address: 192.168.0.1
```
- ◆ `nslookup 192.168.0.1`  

```
Server: 192.168.0.53
Address: 192.168.0.53#53
1.0.168.192.in-addr.arpa name = master1.example.com.
```

## Kubernetes Network Subnet Settings

The Kubernetes network subnet is controlled by the `--POD_CIDR` and `-SERVICE_CIDR` parameters to the CDF Installer.

The `--POD_CIDR` parameter specifies the network address range for Kubernetes pods. The address range specified in the `--POD_CIDR` parameter must not overlap with the IP range assigned for Kubernetes services, which is specified in the `-SERVICE_CIDR` parameter. The expected value is a Classless Inter-Domain Routing (CIDR) format IP address. CIDR notation comprises an IP address, a slash (‘/’) character, and a network prefix (a decimal number). The minimum useful network prefix is /24 and the maximum useful network prefix is /8. The default value is 172.16.0.0/16.

For example:

```
POD_CIDR=172.16.0.0/16
```

The `CIDR_SUBNETLEN` parameter specifies the size of the subnet allocated to each host for Kubernetes pod network addresses. The default value is dependent on the value of the `POD_CIDR` parameter, as described in the following table.

| <b>POD_CIDR Prefix</b> | <b>POD_CIDR_SUBNETLEN defaults</b> | <b>POD_CIDR_SUBNETLEN allowed values</b> |
|------------------------|------------------------------------|--|
| /8 to /21              | /24                                | /(POD_CIDR prefix + 3) to /27            |
| /22 to /24             | /(POD_CIDR prefix + 3)             | /(POD_CIDR prefix + 3) to /27            |

Smaller prefix values indicate a larger number of available addresses. The minimum useful network prefix is /27 and the maximum useful network prefix is /12. The default value is 172.17.17.0/24.

Change the default `POD_CIDR` or `CIDR_SUBNETLEN` values only when your network configuration requires you to do so. You must also ensure that you have sufficient understanding of the flannel network fabric configuration requirements before you make any changes.

# Configuring the NFS Server

The CDF Installer platform requires a NFS server to maintain state information about the infrastructure and to store other pertinent data.

For high availability, NFS must run on a highly available external server in the case of a dedicated master deployment having a minimum of three master nodes. For optimal security, secure all NFS settings to allow only required hosts to connect to the NFS server.

For more information on external server, see [“External NFS Server”](#).

- ◆ [“Prerequisites:” on page 42](#)
- ◆ [“Creating NFS Shared Directories” on page 43](#)
- ◆ [“Exporting the NFS Configuration” on page 43](#)
- ◆ [“Verifying NFS Configuration” on page 44](#)
- ◆ [“Setting Up NFS By Using the Script” on page 44](#)

## Prerequisites:

- ◆ Ensure that the ports 111, 2049, and 20048 are open on the NFS server for communication.
- ◆ Enable the `rpcbind` and `nfs-server` package by executing the following commands on your NFS server:

```
systemctl enable rpcbind
systemctl start rpcbind
systemctl enable nfs-server
systemctl start nfs-server
```

- ◆ The following are the shared directories that you must create and configure. For information about the minimum memory requirement for each directory, see [Identity Intelligence 1.1 System Requirements](#).

| Directory  | Description  |
|--|--|
| <code>&lt;NFS_ROOT_DIRECTORY&gt;/itom-vol</code>         | This is the CDF NFS root folder, which contains the CDF database and files. The disk usage will grow gradually.  |
| <code>&lt;NFS_ROOT_DIRECTORY&gt;/db-single-vol</code>    | This volume is only available when you did not choose PostgreSQL High Availability (HA) for CDF database setting. It is for CDF database.<br><br>During the install you will not choose the Postgres database HA option. |
| <code>&lt;NFS_ROOT_DIRECTORY&gt;/db-backup-vol</code>    | This volume is used for backup and restore of the CDF PostgreSQL database. Its sizing is dependent on the implementation's processing requirements and data volumes.   |
| <code>&lt;NFS_ROOT_DIRECTORY&gt;/itom-logging-vol</code> | This volume stores the log output files of CDF components. The required size depends on how long the log will be kept.   |
| <code>&lt;NFS_ROOT_DIRECTORY&gt;/arcsight-vol</code>     | This volume stores the component installation packages.  |

## Creating NFS Shared Directories

- 1 Log in to the NFS server as `root`.
- 2 Create the following:
  - ♦ **Group:** `arcsight` with a GID 1999  
Example: `groupadd -g 1999 arcsight`
  - ♦ **User:** `arcsight` with a UID 1999  
Example: `useradd -g 1999 arcsight`
  - ♦ **NFS root directory:** Root director under which you can create all NFS shared directories.  
Example (NFS\_Root\_Directory): `/opt/arcsight/nfs/volumes`

- 3 (Conditional) If you have previously installed any version of CDF, you must remove all NFS directories by using the following command for each directory:

```
rm -rf <path to NFS directory>
```

Example:

```
rm -rf /opt/arcsight/nfs/volumes/itom-vol
```

- 4 Create each NFS shared directory using the command:

```
mkdir -p <path to NFS directory>
```

Example:

```
mkdir -p /opt/arcsight/nfs/volumes/itom-vol
```

- 5 For each NFS directory, set the permission to 755 by using the command:

```
chmod -R 755 <path to NFS directory>
```

Example:

```
chmod -R 755 /opt/arcsight/nfs/volumes/itom-vol
```

- 6 For each NFS directory, set the ownership to UID 1999 and GID 1999 using the command:

```
chown -R 1999:1999 <path to NFS directory>
```

Example:

```
chown -R 1999:1999 /opt/arcsight/nfs/volumes/itom-vol
```

If you use a UID/GID different than 1999/1999, then provide it during the CDF installation in the install script arguments `--system-group-id` and `--system-user-id`.

## Exporting the NFS Configuration

For every NFS volume, run the following set of commands on the External NFS server based on the IP address. You will need to export the NFS configuration with the appropriate IP address in order for the NFS mount to work properly.

- 1 Navigate to `/etc/` and open the `exports` file.
- 2 For every node in the cluster, you must update the configuration to grant the node access to the NFS volume shares.

For example:

```
/opt/arcsight/nfs/volumes/arcsight-vol
```

```
192.168.1.0/24(rw, sync, anonuid=1999, anongid=1999, all_squash)
```

- 3 Save the `/etc/exports` file and run the following command:

```
exportfs -ra
```

If you add more NFS shared directories later, you must restart the NFS service.

## Verifying NFS Configuration

- 1 Create the NFS directory under /mnt.

For example,

```
cd /mnt
mkdir nfs
```

- 2 Mount the NFS directory on your local system by using the command:

- ♦ **NFS v3:** `mount -t nfs 192.168.1.25:/opt/arcsight/nfs/volumes/arcsight-vol/mnt/nfs`
- ♦ **NFS v4:** `mount -t nfs4 192.168.1.25:/opt/arcsight/nfs/volumes/arcsight-vol/mnt/nfs`

- 3 After creating all the directories, run the following commands on the NFS server:

```
exportfs -ra
systemctl restart rpcbind
systemctl enable rpcbind
systemctl restart nfs-server
systemctl enable nfs-server
```

## Setting Up NFS By Using the Script

*Applicable only for non-high-availability and single-node deployments.*

You can either run the `/opt/<Identity_Intelligence_Installer>/scripts/preinstall_create_nfs_share.sh` script that sets up the NFS automatically or set up the NFS manually.

### To set up NFS manually:

- 1 Copy `setupNFS.sh` to the NFS server.

The `setupNFS.sh` is located on the master node in the `<download_directory>/identityintelligence-x.x.x/installers/cdf-x.x.x/cdf/scripts` folder.

- 2 (Conditional) If you are using the default UID/GID, then use the command:

```
sh setupNFS.sh <path_to_nfs_directory>/volumes/volume_name
```

Example, `sh setupNFS.sh /opt/arcsight/nfs/volumes/itom-vol`

- 3 (Conditional) If you are using a non-default UID/GID, then use the command:

```
sh setupNFS.sh <path_to_nfs_directory>/volumes/volume_name true <uid> <gid>
```

- 4 Restart the NFS service:

```
systemctl restart nfs
```

## Disabling Swap Space

You must disable swap space on all master and worker nodes excluding the node which has Vertica.

- 1 Log in to the node where you want to disable swap space.
- 2 Run the following command:

```
swapoff -a
```

- 3 In the `/etc/fstab` file, comment out the lines that contain `swap` as the disk type and save the file.

For example:

```
#/dev/mapper/centos_shcentos72x64-swap swap
```

## (Optional) Create Docker Thinpools

Optionally, to improve performance of Docker processing, set up a thinpool on each master and worker node. Before setting up a thinpool on each node, create a single disk partition on the node, as explained below.

For the thinpool device for Docker (for example, **sdb1**): the minimum physical volume size is 30GB.

- ♦ [“Creating a New Partition” on page 45](#)
- ♦ [“Setting Up a Thinpool for Docker” on page 45](#)

## Creating a New Partition

- 1 Log in to the node.
- 2 Run the command:  

```
fdisk <name of the new disk device that was added>
```

Example:

```
# fdisk /dev/sdb1
```
- 3 Enter `n` to create a new partition.
- 4 When prompted, enter partition number, sector, type (Linux LVM), and size for the first partition. To select Linux LVM partition type:
  - ♦ Enter `t` to change the default partition type to Linux LVM
  - ♦ Type `L` to list the supported partition types
  - ♦ Type `8e` to select Linux LVM type
- 5 When prompted, enter partition number, sector, type (Linux LVM), and size for the second partition.
- 6 Type `p` to view the partition table.
- 7 Type `w` to save the partition table to disk.
- 8 Type `partprobe`.

## Setting Up a Thinpool for Docker

- 1 Create a physical volume with the following command:  

```
# pvcreate [physical device name]
```

Example:

```
# pvcreate /dev/sdb1
```

- 2 Create a volume group with the following command:

```
# vgcreate [volume group name] [logical volume name]
```

Example:

```
# vgcreate docker /dev/sdb1
```

- 3 Create a logical volume (LV) for the thinpool and bootstrap with the following command:

```
# lvcreate [logical volume name] [volume group name]
```

For example: the data LV is 95% of the 'Docker' volume group size (leaving free space allows for automatic expanding of either the data or metadata if space is running low, as a temporary stopgap):

```
# lvcreate --wipesignatures y -n thinpool docker -l 95%VG
```

```
# lvcreate --wipesignatures y -n thinpoolmeta docker -l 1%VG
```

- 4 Convert the pool to a thinpool with the following command:

```
# lvconvert -y --zero n -c 512K --thinpool docker/thinpool --poolmetadata docker/thinpoolmeta
```

Optionally, you can configure the auto-extension of thinpools using an lvm profile.

- 4a Open the lvm profile.

- 4b Specify a value for the parameters `thin_pool_autoextend_threshold` and `thin_pool_autoextend_percent`, each of which represents a percentage of the space used.

For example:

```
activation {
  thin_pool_autoextend_threshold=80
  thin_pool_autoextend_percent=20
}
```

- 4c Apply the lvm profile with the following command:

```
# lvchange --metadataprofile docker-thinpool docker/thinpool
```

- 4d Verify that the lvm profile is monitored with the following command:

```
# lvs -o+seg_monitor
```

- 4e Clear the graph driver directory with the following command, if Docker was previously started:

```
# rm -rf /var/lib/docker/*
```

- 4f Monitor the thinpool and volume group free space with the following commands:

```
# lvs
```

```
# lvs -a
```

```
# vgs
```

- 4g Check logs to see the auto-extension of the thinpool when it hits the threshold:

```
# journalctl -fu dm-event.service
```

# Enabling Installation Permissions for a sudo User

If you choose to install the Installer as a `sudo` user, the root user must grant non-root (`sudo`) users installation permission before they can perform the installation. Please make sure the provided user has permission to execute scripts under temporary directory `/tmp` on all master and worker nodes.

There are two distinct file edits that need to be performed: first on the Initial master node only, and then on all remaining master and worker nodes.

- ♦ [“Edit the sudoers File on the Initial Master Node” on page 47](#)
- ♦ [“Edit the sudoers File on the Remaining Master and Worker Nodes” on page 48](#)

## Edit the sudoers File on the Initial Master Node

Make the following modifications on the initial master node only.

---

**WARNING:** In the following commands you must ensure there is, at most, a single space character after each comma that delimits parameters. Otherwise, you may get an error similar to this when you attempt to save the file.

```
>>> /etc/sudoers: syntax error near line nn<<<
```

---

- 1 Log in to the Initial master node as the `root`.
- 2 Open the `/etc/sudoers` file using Visudo.
- 3 Add the following `Cmd_Alias` line to the command aliases group in the `sudoers` file.  

```
Cmd_Alias CDFINSTALL = <CDF_installation_package_directory>/scripts/  
precheck.sh, <CDF_installation_package_directory>/install, <K8S_HOME>/  
uninstall.sh, /usr/bin/kubect1, /usr/bin/docker, /usr/bin/mkdir, /bin/rm, /bin/  
su, /bin/chmod, /bin/tar, <K8S_HOME>/scripts/uploadimages.sh, /bin/chown
```

  - 3a Replace `<CDF_installation_package_directory>` with the directory where you unzipped the installation package.  
For example: `/tmp/cdf-2019.05.0xxx`.
  - 3b Replace `<K8S_HOME>` with the value defined from a command line. By default, `<K8S_HOME>` is `/opt/arcsight/kubernetes`.
- 4 Add the following lines to the wheel users group, replacing `<username>` with your `sudo` user password:  

```
%wheel ALL=(ALL) ALL  
cdfuser ALL=NOPASSWD: CDFINSTALL  
Defaults: <username>!requiretty  
Defaults: root !requiretty
```
- 5 Locate the `secure_path` line in the `sudoers` file and ensure the following paths are present:  

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

By doing this, the `sudo` user can execute the `showmount`, `curl`, `ifconfig`, and `unzip` commands when installing the CDF Installer.
- 6 Save the file.

## Installing Components Using the `sudo` User

After completing the modifications to the `sudoers` files as described above, perform the following steps.

- 1 Log in to the initial master node as the non-root `sudo` user to perform the installation
- 2 Download the installer files to a directory where the non-root `sudo` user has write permissions
- 3 Run the CDF Installer using the `sudo` command (for more details, refer to the your product's Deployment Guide)

## Edit the `sudoers` File on the Remaining Master and Worker Nodes

Make the following modifications only on the remaining master and worker nodes.

---

**WARNING:** In the following commands you must ensure there is, at most, a single space character after each comma that delimits parameters. Otherwise, you may get an error similar to this when you attempt to save the file.

```
>>> /etc/sudoers: syntax error near line nn<<<
```

---

- 1 Log in to each master and worker node.
- 2 Open the `/etc/sudoers` file.
- 3 Add the following `Cmdnd_Alias` line to the command aliases group in the `sudoers` file.

```
Cmdnd_Alias CDFINSTALL = /tmp/scripts/pre-check.sh,  
<ITOM_Suite_Foundation_Node>/install, <K8S_HOME>/uninstall.sh, /usr/bin/  
kubectl, /usr/bin/docker, /usr/bin/mkdir, /bin/rm, /bin/su, /bin/chmod, /bin/  
tar, <K8S_HOME>/scripts/uploadimages.sh, /bin/chown
```

  - 3a Replace `<ITOM_Suite_Foundation_Node>` with the directory where you unzipped the installation package.  
For example: `/tmp/ITOM_Suite_Foundation_2019.05.0xxx`
  - 3b Replace `<K8S_HOME>` with the value defined from a command line. By default, `<K8S_HOME>` is `/opt/arcsight/kubernetes`.
- 4 Add the following lines to the wheel users group, replacing `<username>` with your `sudo` user password:

```
%wheel ALL=(ALL) ALL  
cdfuser ALL=NOPASSWD: CDFINSTALL  
Defaults: <username> !requiretty  
Defaults: root !requiretty
```
- 5 Locate the `secure_path` line in the `sudoers` file and ensure the following paths are present:

```
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
```

By doing this, the `sudo` user can execute the `showmount`, `curl`, `ifconfig`, and `unzip` commands when installing the CDF Installer.
- 6 Save the file.

Repeat the process for each remaining master and worker node.



# Installing CDF

This section provides guidance for installing the CDF Installer.

---

**NOTE:** You can install the CDF Installer as a root user or `sudo` user. However, if you choose to install as a `sudo` user, you must first configure installation permissions from the root user. For more information on providing permissions for the `sudo` user, see [Enabling Installation Permissions for a `sudo` User](#).

---

- 1 Log in to the master node as `root` or `sudo` user.
- 2 Change to the directory where you downloaded the installer files. For information about downloading the installer files, see [Downloading Identity Intelligence](#).

```
cd <download_directory>/identityintelligence-x.x.x/installers/cdf-x.x.x
```

For example:

```
cd opt/identityintelligence-x.x.x/installers/cdf-x.x.x
```

- 3 (Conditional) If you want to install CDF on the node where Vertica is installed, use the command:

```
./install -m <metadata_file_path>  
--fail-swap-on false  
--k8s-home <cdf_installer_directory>  
--nfs-server <NFS_server_IP_address>  
--nfs-folder <NFS_ITOM_volume_file_path>
```

**Example:**

```
./install -m /opt/identityintelligence-x.x.x.x/suite_images/arcsight-installer-  
metadata-x.x.x.x-master.tar.gz  
--fail-swap-on false  
--k8s-home /opt/arcsight/kubernetes  
--nfs-server 10.204.100.224  
--nfs-folder /opt/arcsight/nfs/volumes/itom-vol
```

- 4 (Conditional) If you want to install CDF on a node where Vertica is not installed, use the command:

```
./install -m <metadata_file_path>  
--k8s-home <cdf_installer_directory>  
--nfs-server <NFS_server_IP_address>  
--nfs-folder <NFS_ITOM_volume_file_path>
```

**Example:**

```
./install -m /opt/identityintelligence-x.x.x.x/suite_images/arcsight-installer-  
metadata-x.x.x.x-master.tar.gz  
--k8s-home /opt/arcsight/kubernetes  
--nfs-server 10.204.100.224  
--nfs-folder /opt/arcsight/nfs/volumes/itom-vol
```

- 5 (Conditional) For high availability, use the following command:

```
./install -m <metadata_file_path>
--k8s-home <NFS_server_IP_address>
--nfs-folder <NFS_ITOM_volume_file_path>
--registry-orgname srg
--fail-swap-on false
--ha-virtual-ip <HA_virtual_IP_address>
```

**Example:**

```
/install -m /opt/arcsight-installer-metadata-2.1.0.16-pre-release-2.1.0.tar
--k8s-home /opt/arcsight/kubernetes
--nfs-server 164.99.175.220
--nfs-folder /opt/arcsight/nfs/volumes/itom-vol
--registry-orgname srg
--fail-swap-on false
--ha-virtual-ip 164.99.175.225
```

- 6 When prompted, specify the administrator password. This password is required to log into CDF Management Portal.
- 7 Copy the CDF Management Portal URL displayed in the installation success message.  
Using the CDF Management Portal, you can deploy Identity Intelligence and all required software in a cluster.
- 8 To deploy Identity Intelligence and all required software, continue with the section [Installing Identity Intelligence](#).

## Installing Identity Intelligence

This section provides information about installing Identity Intelligence.

- ♦ [“Configuring the Cluster” on page 50](#)
- ♦ [“Uploading Images to Local Registry” on page 52](#)
- ♦ [“Deploying Transformation Hub and Identity Intelligence” on page 53](#)

## Configuring the Cluster

- 1 Launch the CDF Management Portal using the link ([https://master\\_FQDN:3000](https://master_FQDN:3000)) displayed after CDF installation in [Step 7](#).

Ensure that the browser is not using proxy to access CDF because this might result in inaccessible web pages.

---

**NOTE:** Use port 3000 when you are setting up the CDF for the first time. After the initial setup, use port 5443 to access the CDF Management Portal.

---

- 2 Log in with the following credentials:  
**User name:** admin  
**Password:** Use the password that you provided during CDF installation.
- 3 Select the metadata file version in **version** and click **Next**.
- 4 Read the license agreement and select **I agree. and I authorize**.
- 5 Click **Next**.
- 6 In the Capabilities page, select the following and click **Next**:
  - ♦ Transformation Hub

- ◆ Identity Intelligence
  - ◆ Analytics. Prerequisite for ArcSight Investigate and Identity Intelligence.
- 7 In the Database page, retain the default values, select **Out-of-the-box PostgreSQL**, and click **Next**.
  - 8 In the Deployment Size page, select the required cluster and click **Next**.
    - 8a (Conditional) For worker node configuration, select **Medium Cluster**.

The installation will not proceed if the minimal hardware requirements are not met. For information about the hardware requirements, see [Hardware Requirements and Tuning Guidelines](#).
  - 9 In the Connection page, an external host name is automatically populated. This is resolved from the virtual IP (VIP) specified during the CDF installation (--ha-virtual-ip parameter). Confirm that the VIP is correct and then click **Next**.
  - 10 (Conditional) If you want to set up high availability, select **Make master highly available** and add at least two additional master nodes in the Master High Availability page.

---

**NOTE:** If you do not configure high availability in this step, you cannot add master nodes and configure high availability after installation.

---

In the Add Master Node page, specify the following details:

- ◆ **Host:** Fully qualified domain name (FQDN) of the node you are adding.
- ◆ **Ignore Warnings:** If selected, the installer will ignore any warnings that occur during the pre-checks on the server. If deselected, the add node process will stop and a window will display any warning messages. We recommend that you start with **Ignore Warnings** deselected in order to view any warnings displayed. You may then evaluate whether to ignore or rectify any warnings, clear the warning dialog, and then click **Save** again with the box selected to avoid stopping.
- ◆ **User Name:** User credential for login to the node.
- ◆ **Verify Mode:** Choose the verification mode as *Password* or *Key-based*, and then either enter your password or upload a private key file. If you choose *Key-based*, you must first enter a username and then upload a private key file when connecting the node with a private key file.
- ◆ **Thinpool Device:** (optional) Enter the Thinpool Device path, that you configured for the master node (if any). For example: `/dev/mapper/docker-thinpool`. You must have already set up the Docker thin pool for all cluster nodes that need to use thinpools, as described in the CDF Planning Guide.
- ◆ **flannel IFace:** (optional) Enter the flannel IFace value if the master node has more than one network adapter. This must be a single IPv4 address or name of the existing interface and will be used for Docker inter-host communication.

Click **Save**. Repeat the same for other master nodes.

- 11 Click **Next**.
- 12 (Conditional) For multi-node deployment, add additional worker nodes in the Add Worker Node page and click **Save**. To add a worker node click **+** (Add) and enter the required configuration information. Repeat this process for each of the worker nodes.
- 13 Click **Next**.
- 14 (Conditional) If you want to run the worker node in the master node, then select **Allow suite workload to be deployed on the master node** and then click **Next**.

---

**NOTE:** Before selecting this option, ensure that the master node meets the [system requirements](#) specified for the worker node.

---

**15** To configure each NFS volume, complete the following steps:

**15a** Navigate to the **File Storage** page.

**15b** For **File System Type**, select **Self-Hosted NFS**.

Self-hosted NFS refers to the [external NFS](#) that you created while preparing the environment for CDF installation.

**15c** For **File Server**, specify the IP address or FQDN of the NFS server.

**15d** For **Exported Path**, specify the following paths for the NFS volumes:

| NFS Volume       | File Path                          |
|------------------|------------------------------------|
| arcsight-vol     | <NFS_ROOT_FOLDER>/arcsight-vol     |
| db-single-vol    | <NFS_ROOT_FOLDER>/db-single-vol    |
| itom-logging-vol | <NFS_ROOT_FOLDER>/itom-logging-vol |
| db-backup-vol    | <NFS_ROOT_FOLDER>/db-backup-vol    |

**15e** Click **Validate**.

Ensure that you have validated all NFS volumes successfully before continuing with the next step.

**16** Click **Next**.

**17** To start deploying master and worker nodes, click **Yes** in the Confirmation dialog box.

**18** Continue with [Uploading Images to Local Registry](#).

## Uploading Images to Local Registry

For the docker registry to deploy Identity Intelligence, it needs the following images associated with the deployment:

- ♦ transformationhub-x.x.x.x
- ♦ idi-x.x.x.x
- ♦ analytics-x.x.x.x

You must upload those images to the local registry.

**1** Launch a terminal session, then log in to the master node as `root` or a `sudo` user.

**2** Change to the following directory:

```
cd /<cdf_installer_directory>/kubernetes/scripts/
```

For example:

```
cd /opt/arcsight/kubernetes/scripts
```

**3** Upload each of the software images to the local registry:

```
./uploadimages.sh -d <downloaded_image_file_path> -u registry-admin -p <password>
```

Example:

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x./  
suite_images/analytics-x.x.x.x -u registry-admin -p password
```

- 4 Continue with [Deploying Transformation Hub and Identity Intelligence](#).

## Deploying Transformation Hub and Identity Intelligence

After you upload the images to the local directory, CDF uses these images to deploy the respective software in the cluster.

- 1 Switch back to the CDF Management Portal.
- 2 Click **Next** in the **Download Images** page because all the required packages are already downloaded and uncompressed.
- 3 After the **Check Image Availability** page displays All images are available in the registry, click **Next**.

If the page displays any missing image error, [upload the missing image](#).

- 4 After the **Deployment of Infrastructure Nodes** page displays the status of the node in green, click **Next**.

The deployment process can take up to 15 minutes to complete.

- 5 (Conditional) If any of the nodes show a red icon in the **Deployment of Infrastructure Nodes** page, click the **retry** icon.

CDF might display the red icon if the process times out for a node. Because the retry operation executes the script again on that node, ensure that you click **retry** only once.

- 6 After the **Deployment of Infrastructure Services** page indicates that all the services are deployed and the status indicates green, click **Next**.

The **Preparation Complete** message appears after the deployment process is complete and it can take up to 15 minutes to complete.

(Optional) To monitor the progress of service deployment, complete the following steps:

**6a** Launch a terminal session.

**6b** Log in to the master node as `root`.

**6c** Execute the command:

```
watch 'kubectl get pods --all-namespaces'
```

- 7 (Conditional) If you want to use [mutual SSL authentication](#) between Transformation Hub and its clients by enabling client authentication, you must change the default CA that is generated during the installation. For steps to change the CDF CA, see [Changing the Certificate Authority of CDF](#).

- 8 Click **Next**.

- 9 To configure pre-deploy settings for all the following software, complete the following steps:

**9a** In the **Transformation Hub** tab:

- ♦ Set the values based on the workload or high availability configuration. For information about this value for your deployment, see the “Transformation Hub Tuning” section in the [Hardware Requirements and Tuning Guidelines](#).
- ♦ Set **Allow plain text (non-TLS) connection to Kafka** to `False` to disable plain text communication between Transformation Hub (Kafka) and all the components outside the Kubernetes cluster.

When you set this option to `False`, ensure to configure SSL between Transformation Hub (Kafka) and all the components outside the Kubernetes cluster, such as Identity Governance, Identity Manager Driver for Entity Data Model, Vertica, and so on.

- ◆ **Enable Connection to Kafka uses TLS Client Authentication:** This option is used to enable client authentication between Transformation Hub and all the components outside the Kubernetes cluster. When you enable this option, ensure that you configure [mutual SSL authentication](#) between Transformation Hub (Kafka) and all the components outside the Kubernetes cluster, such as Identity Governance, Identity Manager Driver for Entity Data Model, Vertica, and so on.

**9b** In the **Analytics** tab:

- ◆ Specify Vertica connection details.
- ◆ (Optional) Specify SMTP server details to enable users of Identity Intelligence to receive email notification.
- ◆ Specify the values for **Client ID** and **Client Secret** for Single Sign-On.

**10** To finish the deployment, click **Next**.

**11** Copy the Management portal link displayed in the **Configuration Complete** page.

Some of the pods in the **Configuration Complete** page might remain in a pending status until the product labels are applied on worker nodes. To label the nodes, see [Labeling Nodes](#).

**12** (Conditional) For high availability and multi-master deployment, after the deployment has been completed, manually restart the keepalive process.

**12a** Log in to the master node.

**12b** Change to the directory:

```
cd /<k8S_HOME>/bin/
```

For example:

```
cd /opt/arcsight/kubernetes/bin/
```

**12c** Run the script:

```
./start_lb.sh
```

**13** Continue with the following activities:

- ◆ [Post-Installation Configurations](#)
- ◆ [Configuring Data Collection](#)

# 7 Deploying Identity Intelligence in an Existing Cluster

If you already have an existing cluster for Transformation Hub or CDF, you can deploy Identity Intelligence to the same cluster. Reusing existing clusters would reduce costs and system management effort compared to deploying these software in a new cluster.

- ◆ [“Prerequisites” on page 55](#)
- ◆ [“Deploying Identity Intelligence” on page 55](#)

## Prerequisites

Before installing Identity Intelligence, complete the following tasks:

- ◆ Review the [Implementation Checklist](#) to understand the tasks involved for installing and configuring Identity Intelligence.
- ◆ Ensure that you have the supported version of CDF. If your deployment does not have the supported version, you must [upgrade CDF](#).

For information about the supported version of Transformation Hub, see the [Identity Intelligence 1.1 System Requirements](#).


- ◆ Ensure that you have the supported version of Transformation Hub. If your deployment does not have the supported version of Transformation Hub, you can upgrade when deploying Identity Intelligence.

For information about the supported version of Transformation Hub, see the [Identity Intelligence 1.1 System Requirements](#).

- ◆ If your deployment does not have Vertica, install [Vertica](#). If your deployment already has Vertica ensure that it is the supported version, else [upgrade Vertica](#).
- ◆ Upgrade the existing cluster to the correct version of the ArcSight Suite in the CDF Management Portal to deploy the current version of Identity Intelligence.
- ◆ Download the [Identity Intelligence installer files](#) and verify the signatures. You need the following files to deploy Identity Intelligence in an existing cluster:
  - ◆ `idi-x.x.x`
  - ◆ `analytics-x.x.x` (if Analytics is not installed)
  - ◆ `transformationhub-x.x.x.x` (to upgrade Transformation Hub to the supported version)

## Deploying Identity Intelligence

To deploy Identity Intelligence in an existing cluster, perform the following:

- 1 Log in to the CDF management portal.
- 2 Click **Deployment > Deployments >**  of `arcsight-installer`, then click **Change**.

- 3 In the **Capabilities** page, select the following options:
  - ◆ Identity Intelligence
  - ◆ Analytics (if not selected already)
  - ◆ Transformation Hub (to upgrade Transformation Hub to the supported version)
- 4 Click **Next** until you reach the **Configuration Complete** page.
- 5 Launch a terminal session, then log in to the master node as `root` or as a `sudo` user.
- 6 Change to the following directory:
 


```
cd /<cdf_installer_directory>/scripts/
```

 For example:
 

```
cd /opt/cdf-x.x.x.x/scripts/
```
- 7 (Conditional) If you want to upgrade Transformation Hub, upload the Transformation Hub image:
 

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x./suite_images/transformationhub-x.x.x.x -u registry-admin -p <cdf_password>
```
- 8 (Conditional) If your deployment does not have Analytics, upload the Analytics image:
 

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x./suite_images/analytics-x.x.x.x -u registry-admin -p <cdf_password>
```
- 9 Upload the Identity Intelligence image to the local registry:
 

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x./suite_images/idi-x.x.x.x -u registry-admin -p <cdf_password>
```
- 10 Switch to the CDF Management portal, then click **CHECK AGAIN** to ensure that the images have been uploaded.
- 11 Click **Next** until you reach the **Configuration Complete** page.
- 12 After all the pods are displayed in green in the **Configuration Complete** page, click **Next**.
- 13 (Conditional) If your deployment did not have Vertica and you installed Vertica, then you must configure Vertica connection details in Identity Intelligence as follows:
  - 13a Click **Deployment > Deployments > ** of `arcsight-installer`, then click **Reconfigure**.
  - 13b Click **Analytics**.
  - 13c Specify Vertica connection details in the **Vertica Configuration** section.
  - 13d Click **Save**.
- 14 Identity Intelligence supports only SSL communication between Identity Intelligence and its related components. Therefore, if SSL is not configured in your environment, configure SSL by performing the steps in the [Disabling Plain Text Communication](#) section.
- 15 Continue with [Chapter 8, "Post-Installation Configurations,"](#) on page 57.



# 8

## Post-Installation Configurations

This chapter provides information about the post-installation configurations you must perform after installing Identity Intelligence manually.

- ◆ “Labeling Nodes” on page 57
- ◆ “Configuring Flannel Memory” on page 58
- ◆ “Setting the Default Locale for the Database” on page 59
- ◆ “Configuring SSL for Vertica” on page 59
- ◆ “Performance Tuning for Data Ingestion” on page 63

### Labeling Nodes

*Does not apply if you used the `./install-single-node-post.sh` installation script*

Labeling a node tells Kubernetes what type of application can run on a specific node. It is a means for identifying application processing and qualifying the application as a candidate to run on a specific host system. Labeling is required only for worker nodes and not for master nodes.

The method for labeling nodes depends on whether you installed Identity Intelligence in a new or an [existing](#) cluster.

- ◆ “Labeling Nodes in a New Cluster” on page 57
- ◆ “Labeling Nodes in an Existing Cluster” on page 58

### Labeling Nodes in a New Cluster

You can follow the instructions in this section to label the nodes manually for both single-node and multi-node deployment. However, for single node deployment, you can alternatively use the `/opt/<Identity_Intelligence_Installer>/postinstall_label_master_node.sh` script to label the node automatically.

To label the nodes in a new cluster:

- 1 Launch the CDF Management portal using the link (<https://ha-address:5443>) that you copied from the Configuration Complete page in [Step 11](#).
- 2 Log in with the following credentials:  
**User name:** `admin`  
**Password:** The password that you provided during CDF installation
- 3 Select **Cluster > Nodes**.
- 4 In **Predefined Labels**, click **+** to add labels.
- 5 Specify the labels as follows:

---

**NOTE:** Labels are case-sensitive. Ensure that you enter the values correctly.

---

- ◆ `analytics:yes`

- ◆ th-platform:yes
  - ◆ th-processing:yes
  - ◆ zk:yes
  - ◆ kafka:yes
- 6 (Conditional) For single-node deployment, drag all the newly added labels to the worker node.
  - 7 (Conditional) For multi-node deployment, drag-and-drop the new labels from the predefined set to each of the worker nodes based on your workload sharing configuration.  
You may need to click **Refresh** to see the attached labels.
  - 8 For Kafka and ZooKeeper, ensure that the number of the nodes you labeled corresponds to **# of Worker Nodes in the Kafka cluster** and **# of Worker Nodes running ZooKeeper in the Kafka cluster** properties from the pre-deployment configuration page.  
In a multi-worker deployment, the default number is 3.

## Labeling Nodes in an Existing Cluster

You do not need to label the nodes if you have Investigate because both Identity Intelligence and Investigate use the same label: `analytics:yes`. However, if your deployment does not have Investigate, you can either add Identity Intelligence to the Transformation Hub node or add a new node for Identity Intelligence.

- 1 Launch the CDF Management portal.
- 2 Log in with the following credentials:  
**User name:** `admin`  
**Password:** The password that you provided during CDF installation
- 3 Select **Cluster > Nodes**.
- 4 (Conditional) To deploy Identity Intelligence in a new node, click **ADD**.
- 5 Specify the details of the node, such as worker node type, host name or IP address, user name, and password.
- 6 Drag `analytics:yes` label to the newly added worker node or the Transformation Hub node.

## Configuring Flannel Memory

*Applies only if you installed Identity Intelligence in a new cluster. Does not apply if you used the `./install-single-node-post.sh` installation script, which automatically performed this configuration.*

In some cases, after flannel pods have been running continuously for some time, the Kafka Manager pod (and others) might terminate abruptly. To prevent this issue, you should modify the flannel file.

You can either use the `/opt/<Identity_Intelligence_Installer>/postinstall_adjust_flannel_mem.sh` script automatically or you can configure the flannel memory manually:

- 1 Back up the existing flannel `yaml` file by executing the following commands:
 

```
cp $
{K8S_HOME}/objectdefs/flannel.yaml ${K8S_HOME}
/objectdefs/flannel.yaml.orig
```

- 2 In the `vi ${K8S_HOME}/objectdefs/flannel.yaml` file, change both the request and limits memory to 250Mi under `spec > spec > resources`.
- 3 Delete the existing flannel file by executing the following command:
 

```
kubectl delete -f ${K8S_HOME}/objectdefs/flannel.yaml
```
- 4 Create a new flannel `yaml` file by executing the following command:
 

```
kubectl create -f ${K8S_HOME}/objectdefs/flannel.yaml
```
- 5 Verify the change by executing the following command on each flannel pod:
 

```
kubectl get pod $f_pod -n kube-system -o yaml | grep -A6 resources|grep memory
```
- 6 Ensure that the memory value is set to 250Mi.

## Setting the Default Locale for the Database

*Does not apply if you used the `./install-single-node-post.sh` installation script, which automatically performed this configuration.*

For Identity Intelligence to function properly, when Vertica is running in the same cluster as Investigate, the locale needs to be `en_US@colstrength=secondary` (case-insensitive). However, using this local might reduce the search performance of Investigate.

You can either use the `/opt/<Identity_Intelligence_Installer>/postinstall_vertica_set_default_session_locale.sh` script automatically or you can set the default locale manually:

- 1 Log in to Vertica as a database administrator.
- 2 To set the locale, execute the following command:

```
alter database Investigate set DefaultSessionLocale = <locale>;
```

For example:

```
alter database Investigate set DefaultSessionLocale =
'en_US@colstrength=secondary';
```

For more information, see the [Implement Locales for International Data Sets](#) section in the [Vertica Documentation](#).

## Configuring SSL for Vertica

*You must configure [mutual SSL](#) in Vertica to secure the communication with Vertica.*

To configure SSL in Vertica manually, perform the following:

- ♦ [“Obtaining Vertica Client Certificates” on page 60](#)
- ♦ [“Generating Vertica Server Certificate” on page 60](#)
- ♦ [“Enabling SSL in Vertica” on page 61](#)
- ♦ [“Establishing an SSL Communication with Identity Intelligence” on page 62](#)

For more information about configuring SSL in Vertica, see [SSL Authentication](#) section in Vertica Documentation.

## Obtaining Vertica Client Certificates

You must obtain the certificates from search engine (Vertica client) and use them for enabling SSL between Vertica and scheduler as well as Vertica and search engine.

- 1 To obtain the search engine pod, run the following command on the master node:

```
kubectl get pods --all-namespaces | grep hercules-search-engine
```

Example output:

```
arcsight-installer-9tm5c hercules-search-engine-c97657f9999xpx 2/2 Running 0  
17m
```

---

**NOTE:** The search engine pod is reflected as `hercules-search-engine-c97657f9999xpx` in the example above.

---

- 2 To obtain the search engine certificates, run the following command on the master node:

```
kubectl cp <namespace>/<pod>:/vault-crt/RE <path to copy> -c <container>
```

Example:

```
kubectl cp arcsight-installer-9tm5c/hercules-search-engine-c97657f99-99xpx:/  
vault-crt/RE /root -c hercules-search-engine
```

---

**NOTE:** Three files will be generated: `issue_ca.crt`, `vertica.crt`, and `vertica.key`.

---

- 3 Copy `issue_ca.crt`, `vertica.crt`, and `vertica.key` to `<vertica-cluster-node-1>/root`.

## Generating Vertica Server Certificate

- 1 Log in to Vertica cluster node 1 as a `root`.
- 2 Get the CA certificate for Vertica:
  - ♦ If you have a well-known root CA, organization's root CA, or generated a new CDF root CA, you can use the same CA.
  - ♦ If you do not have a CA, generate a new CA by executing the instructions in the [Generating a New CA](#) section.
- 3 To generate a Vertica server certificate, run the following command and specify the necessary information that will be incorporated into your certificate request:

```
openssl req -newkey rsa:2048 -new -nodes -keyout server.key -out server.csr
```

- 4 After entering the requested information, run the following command:

```
openssl x509 -req -in server.csr -days 3650 -sha1 -CAcreateserial -CA ca.crt -  
CAkey ca.key -out server.crt
```

- 5 Verify the generated certificate:

- ♦ `openssl x509 -noout -purpose -in server.crt | grep "SSL server"`

Example output:

```
SSL server : Yes
```

```
SSL server CA : No
```

```
Netscape SSL server : Yes
```

```
Netscape SSL server CA : No
```

- ♦ `openssl x509 -noout -purpose -in ca.crt | grep "SSL server CA : Yes"`

Example output:

```
SSL server CA : Yes (WARNING code=3)
Netscape SSL server CA : Yes (WARNING code=3)
```

- ♦ `openssl verify -CAfile ca.crt server.crt`

Example output:

```
server.crt: OK
```

- 6 Copy `server.key` and `server.crt` to `<vertica-installer-directory>/arcsight-vertica:`  
`cp server.key server.crt /opt/arcsight-vertica`

## Enabling SSL in Vertica

- 1 Log in to Vertica cluster node 1 as `root`.

- 2 Change to the root directory.

```
cd /root
```

- 3 Copy the [files obtained from search engine](#) and the [Vertica server certificate](#) to `/tmp` folder:

```
cp vertica.crt vertica.key ca.crt issue_ca.crt /tmp
```

- 4 Change to the directory where Vertica is installed:

```
cd /opt/arcsight-vertica
```

- 5 Enable SSL using the command:

```
./vertica_ssl_setup --enable-ssl --vertica-cert-path /opt/arcsight-vertica/
server.crt --vertica-key-path /opt/arcsight-vertica/server.key --client-ca-
path /tmp/issue_ca.crt
```

- 6 Switch to `dbadmin` user:

- 7 (Conditional) If the `~/vsq1` folder exists already, you must delete the contents of the folder.

```
rm -rf ~/vsq1/*
```

- 8 (Conditional) If the `~/vsq1` folder does not exist, create the folder:

```
mkdir ~/vsq1
```

- 9 Copy the following certificates to `~/vsq1` folder:

```
cp /tmp/vertica.crt ~/vsq1/client.crt
```

```
cp /tmp/vertica.key ~/vsq1/client.key
```

```
cp /tmp/ca.crt ~/vsq1/ca_root.crt
```

```
chmod 600 ~/vsq1/client.key
```

- 10 Verify the SSL configuration:

- ♦ Verify whether the SSL cipher is displayed:

```
vsq1 -m require
```

Example output:

```
SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256, protocol:
TLSv1.2)
```

- ♦ Verify whether `ssl_state` is `Mutual`:

```
dbadmin=> select user, authentication_method, ssl_state from sessions where
session_id = current_session();
```


Example output:

```
current_user | authentication_method | ssl_state
-----+-----+-----
dbadmin      | Password                          | Mutual
(1 row)
```

## Establishing an SSL Communication with Identity Intelligence

For the SSL communication between Vertica and Identity Intelligence, you need to upload the Vertica CA certificate in the Analytics component. Identity Intelligence will use the Vertica CA certificate uploaded in the Analytics component.

### To add Vertica certificate:

- 1 Log in to the CDF management portal as an administrator.
- 2 Click **Deployment > Deployments**.
- 3 Click  of `arcsight-installer`, then click **Reconfigure**.
- 4 Click **Analytics**.
- 5 Perform the following in the **Vertica Configuration** section:
  - 5a Enable **Vertica connections will use SSL**.
  - 5b Copy the content of **Vertica Root CA** certificate (`ca.crt`) and paste in **Vertica certificate(s)**. Ensure that there are no additional entries, such as space or line breaks at the end of the certificate content.
- 6 Click **Save**.
- 7 Continue with [Creating a Kafka Scheduler with SSL](#).

## Creating a Kafka Scheduler with SSL

*Applies only if you installed Identity Intelligence either in a new cluster or in an existing cluster that has Transformation Hub without Investigate. Does not apply if you used the [./install-single-node-post.sh](#) installation script, which automatically performs this configuration.*

You must create a Kafka scheduler for Vertica to receive data from Transformation Hub.

To create a kafka scheduler, perform the following:

- 1 Log in to the Vertica cluster node 1 as `root`.
- 2 Change to the directory where Vertica is installed:

```
cd /opt/arcsight-vertica
```
- 3 Set up Kafka scheduler:

```
./sched_ssl_setup --enable-ssl --sched-cert-path /tmp/vertica.crt --sched-key-path /tmp/vertica.key --vertica-ca-path /tmp/ca.crt --kafka-ca-path /tmp/issue_ca.crt
```
- 4 Create the SSL Kafka scheduler:

```
./kafka_scheduler create
<Transformation_Hub_Node_1_IP>:9093,<Transformation_Hub_Node_ 2_IP>:9093
<Transformation_Hub_Node_3_IP>:9093
```

- 5 Verify Kafka scheduler creation and validate whether the port number is the Kafka SSL port number (default 9093):

```
./kafka_scheduler status
```

Example output:

```
SSL/TLS mode is enabled
Scheduler Kafka Configuration:
  kafka cluster | topic | partitions | enabled
-----+-----+-----+-----
vlab052002.dom052000.lab:9093 | th-arcsight-avro | 1 | t
(1 row)
Active Scheduler Process:
  scheduler name
-----
investigation_scheduler_1_vlab052002.dom052000.lab
(1 row)
```

- 6 Check the event-copy progress and messages:

```
./kafka_scheduler events
./kafka_scheduler messages
```

## Performance Tuning for Data Ingestion

Depending on the data load (workload), Identity Intelligence takes a while to process and load the data in Identity Intelligence user interface. Therefore, for [medium](#) and [large](#) workloads, you must perform the following tuning to improve data processing throughput:

- 1 Log in to the CDF management portal.
- 2 Navigate to **Cluster > Dashboard link** and click **Launch Dashboard**.
- 3 Select **arcsight-installer-<xxxx>** in the Namespace.
- 4 Select **Config Maps > itom-di-dp-cm**.
- 5 Edit **process.vertica.max.rows.per.batch** to set the value to 5000 and click **Update**.





# 9 Verifying the Installation

To determine whether the installation is successful, launch Identity Intelligence.

- 1 Open a certified web browser.
- 2 Specify the URL of Identity Intelligence:

```
https://<IDI-Server>/idi
```

- 3 Log in with the administrator name and password that you specified during the installation.

The first time that you log in to Identity Intelligence, the application prompts you to create credentials for the Identity Intelligence administrator. After logging in, you can [create additional Identity Intelligence users](#).



# 10 Installing and Configuring ArcMC

ArcMC is a centralized management interface that helps you to effectively administrate and monitor software and components such as Transformation Hub and SmartConnectors.

---

**NOTE:** Installing ArcMC is optional.

---

To install and configure ArcMC, refer to the following table:

---

| <b>Task</b>                               | <b>See</b>  |
|---|---|
| Install and configure ArcMC               | “Software Installation” in the <a href="#">ArcSight Management Center Administrator’s Guide</a>             |
| Add SmartConnector to ArcMC               | “Managing SmartConnectors on ArcMC” in the <a href="#">ArcSight Management Center Administrator’s Guide</a> |
| Add Transformation Hub as a host in ArcMC | “Configuring ArcMC to Manage Transformation Hub” in the <a href="#">Transformation Hub Deployment Guide</a> |
| Configure ArcMC in Transformation Hub     | “Post-Deployment Configuration” in the <a href="#">Transformation Hub Deployment Guide</a>                  |

---





# Configuring Data Collection

Identity Intelligence gathers entity data and events from data sources such as Identity Manager and Identity Governance:

## **Entity Data**

Represents contextual information about users, such as title, manager, access rights, and accounts assigned.

## **Entity Change Events**

Represents changes to entity data, such as addition, deletion, modification, or change in relationships.

## **Audit/Activity Events**

Represents activities such as user requests, approvals, and provisioning of permissions for roles and resources.

In general, [Views](#) display information from audit and activity events, while [users](#) and [access rights](#) display information from entity data.

- ♦ [Chapter 11, “Data Collection Configuration Checklist,” on page 71](#)
- ♦ [Chapter 12, “Tuning Ingestion of Backdated Events,” on page 73](#)
- ♦ [Chapter 13, “Installing and Configuring the SmartConnector,” on page 75](#)
- ♦ [Chapter 14, “Configuring Entity Change Events Collection,” on page 85](#)
- ♦ [Chapter 15, “Customizing Data Reconciliation,” on page 87](#)
- ♦ [Chapter 16, “Configuring Data Collection from Identity Manager,” on page 89](#)
- ♦ [Chapter 17, “Configuring Data Collection from Identity Governance,” on page 97](#)
- ♦ [Chapter 18, “Reverting Backdated Events Configuration,” on page 107](#)
- ♦ [Chapter 19, “Verifying Data Collection Configuration,” on page 109](#)



# 11

## Data Collection Configuration Checklist

To successfully configure data collection, complete the following checklist in the listed order:

---

| Task                     |   |
|--------------------------|---|
| <input type="checkbox"/> | 1. <a href="#">Install and configure the SmartConnector for Syslog NG Daemon</a>    |
| <input type="checkbox"/> | 2. <a href="#">Tuning Ingestion of Backdated Events</a>                             |
| <input type="checkbox"/> | 3. <a href="#">Configure entity change events collection</a>                        |
| <input type="checkbox"/> | 4. <a href="#">Customize data reconciliation</a>                                    |
| <input type="checkbox"/> | 5. (Conditional) <a href="#">Configure data collection from Identity Manager</a>    |
| <input type="checkbox"/> | 6. (Conditional) <a href="#">Configure data collection from Identity Governance</a> |
| <input type="checkbox"/> | 7. <a href="#">Reverting Backdated Events Configuration</a>                         |
| <input type="checkbox"/> | 8. <a href="#">Verify whether the data collection configuration is successful</a>   |

---

If you have both Identity Manager and Identity Governance, to leverage data from both the data sources, you must configure data collection as follows:

- ◆ Configure Identity Governance to collect data from Identity Manager.  
For more information, see [Identity Governance documentation](#).
- ◆ Configure Identity Intelligence to collect data from both Identity Manager and Identity Governance and ensure that the data reconciliation fields for both the data sources are same.

Similarly, if you have multiple Identity Manager systems or multiple Identity Governance systems, you should configure all the systems to send data to Identity Intelligence. When multiple systems are sending data, it is important to correctly configure [data reconciliation](#) so that Identity Intelligence can reconcile any overlapping data.

---

**NOTE:** Identity Intelligence reports data as accurately as provided by the data sources. Therefore, if you observe any data accuracy issues, validate the data accuracy in the data source.

---





# 12 Tuning Ingestion of Backdated Events

Events Kafka scheduler reads events from Transformation Hub and stores the events in Vertica. Before initiating data migration, you must tune events Kafka scheduler to preserve the time stamp of the backdated events.

## To tune event Kafka scheduler for data migration:

1 Log in to Vertica node 1.

2 Change to the directory:

```
cd <vertica_installation_dir>/scripts
```

**Example:** `cd /opt/arcsight-vertica/scripts`

3 Before data migration, execute the following to accept backdated data:

```
./tuning_util.sh -d
```

---

**IMPORTANT:** After data migration from data sources, you must revert events Kafka scheduler to the default configuration to avoid any performance issue. For more information, see [“Reverting Backdated Events Configuration” on page 107](#).

---



# 13 Installing and Configuring the SmartConnector

Identity Intelligence uses the SmartConnector for Syslog NG Daemon for the following purposes:

- ♦ Collect [entity change events](#) of the data source, generated by Identity Intelligence and send the data to Transformation Hub for processing.
- ♦ Collect [audit events](#) generated in Identity Manager and send the data to Transformation Hub for processing.
- ♦ Collect [user activity events](#) from Identity Governance and send the data to Transformation Hub for processing.

This chapter provides information about installing and configuring the SmartConnector in deployments where ArcMC is not being used. For information about installing and configuring the SmartConnector where ArcMC is in use, see the SmartConnector documentation.

- ♦ [“Prerequisites” on page 75](#)
- ♦ [“Installing the SmartConnector” on page 75](#)
- ♦ [“Adding Categorization Files” on page 76](#)
- ♦ [“Creating TrustStore for One-Way SSL with Transformation Hub” on page 77](#)
- ♦ [“Creating TrustStore and KeyStore for Mutual SSL with Transformation Hub” on page 77](#)
- ♦ [“Configuring the SmartConnector” on page 81](#)
- ♦ [“Configuring Additional Destination for Identity Manager Audit Events” on page 83](#)
- ♦ [“Verifying the SmartConnector Configuration” on page 84](#)

## Prerequisites

Complete the following prerequisites before you install the SmartConnector:

- ♦ [Install Identity Intelligence](#) and complete the [post-installation](#) tasks as applicable.
- ♦ [Download the SmartConnector installation file](#) to `/opt` in the worker node with label `analytics:yes` (Identity Intelligence node) node.
- ♦ Install the following packages in the node where you plan to install the SmartConnector:

```
yum install libXext libXrender libXtst fontconfig
```

## Installing the SmartConnector

You must install two different instances of SmartConnector in the Identity Intelligence node but in different directories: one for audit events collection using SSL and one for entity change events collection using non-SSL. This is because Identity Intelligence can send entity change events to the SmartConnector only using non-SSL communication.

## To install the SmartConnector:

- 1 Log in to the Identity Intelligence node as the `root` user.
- 2 Change to the directory (for example, `/opt`) where you [downloaded and unzipped](#) the SmartConnector installation file.
- 3 Change to the following directory:  

```
cd smartconnectorforidentityintelligence x.x.x
```
- 4 Update permissions for the `<SmartConnector_installation>` file:  
Example:  

```
chmod 755 ArcSight-<version>-Connector-Linux64.bin
```
- 5 Install the SmartConnector:  
Example:  

```
./ArcSight-<version>-Connector-Linux64.bin
```
- 6 Enter the installation directory (for example, `/opt/SmartConnector`).

---

**NOTE:** For audit events collection, specify a different directory for the installation where the SmartConnector for entity change events is not installed.

---

- 7 When prompted for create links, select the default, which is to create the links in the desired location (for example, `/root`).  
**Create links** creates a link to the [uninstallation script](#).
- 8 Continue with [Adding Categorization Files](#).

## Adding Categorization Files

After installing the SmartConnector, you must add the categorization files to the appropriate location on the agent and configure the SmartConnector for Syslog NG Daemon. These files categorize the data coming in from the data source. They enrich events with normalized ArcSight category definitions.

You must add the categorization files for both the instances of the SmartConnector.

To copy the categorization files in the [SmartConnector installer](#) to the appropriate location on the agent, run the following commands:

- ◆ `cp -R <SmartConnector Installer Directory>/micro_focus/ <SmartConnector Installation Directory>/current/user/agent/acp/categorizer/current/`

For example, `cp -R /opt/smartconnectorforidentityintelligence1.1.0/micro_focus/ /opt/SmartConnector/current/user/agent/acp/categorizer/current/`

- ◆ `cp -R <SmartConnector Installer Directory>/netiq/ <SmartConnector Installation Directory>/current/user/agent/acp/categorizer/current/`

For example, `cp -R /opt/smartconnectorforidentityintelligence1.1.0/netiq/ /opt/SmartConnector/current/user/agent/acp/categorizer/current/`

# Creating TrustStore for One-Way SSL with Transformation Hub

If you want to establish [one-way SSL authentication](#) between SmartConnector and Transformation Hub, you must first obtain the certificate from Transformation Hub, and then upload this certificate to a TrustStore in the computer where you have installed SmartConnector for audit events collection:

- 1 [Retrieve the CDF CA certificate](#) and copy the certificate to the computer where you plan to install the SmartConnector.
- 2 Upload the certificate to the TrustStore file:

```
/usr/lib/jvm/jre/bin/keytool -import -alias <alias name> -file <cert_file> -storetype JKS -keystore <trust_store_file>
```

Example:

```
/usr/lib/jvm/jre/bin/keytool -import -alias vlab2004 -file /tmp/ca.cer -storetype JKS -keystore VLAB2004.JKS
```

- 3 (Conditional) If you have multiple CA certificates, repeat Step 2 for each CA certificate in the certificate chain.
- 4 Continue with [Configuring the SmartConnector](#).

# Creating TrustStore and KeyStore for Mutual SSL with Transformation Hub

If you have enabled client authentication in Transformation Hub, you must configure [mutual SSL authentication](#) between SmartConnector and Transformation Hub.

**To configure mutual SSL between Transformation Hub and SmartConnector, perform the following:**

- 1 On the SmartConnector server:

- 1a Change to the current directory,

**Linux:**

```
cd <install dir>/current
```

**Windows:**

```
cd <install dir>\current
```

- 1b Set the environment variables for the static values used by keytool:

**Linux:**

```
export CURRENT=<full path to this "current" folder>
```

```
export TH=<th hostname>_<th port>
```

```
export STORES=${CURRENT}/user/agent/stores
```

```
export STORE_PASSWD=<password>
```

```
export TH_HOST=<TH master host name>
```

```
export CA_CERT=ca.cert.pem
```

```
export CERT_CA_TMP=/opt/cert_ca_tmp
```

**Windows:**

```
set CURRENT=<full path to this "current" folder>
```

```

set TH=<th hostname>_<th port>
set STORES=%CURRENT%\user\agent\stores
set STORE_PASSWD=<password>
set TH_HOST=<TH master host name>
set CA_CERT=C:\Temp\ca.cert.pem
set CERT_CA_TMP=\opt\cert_ca_tmp

```

**1c** (Conditional) Create the `stores` directory if it does not exist:

**Linux:**

```
mkdir ${STORES}
```

**Windows:**

```
mkdir %STORES%
```

**1d** From a command prompt, change to the installation directory of the `keytool` utility. The default installation directory is:

**Linux:**

```
/usr/lib/jvm/jre/bin
```

**Windows:**

```
c:\usr\lib\jvm\jre\bin
```

**1e** Create the key pair:

**1e1** Execute the command:

**Linux:**

```
./keytool -genkeypair -alias ${TH} -keystore ${STORES}/
${TH}.keystore.jks -dname "cn=<Connector
FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -validity 375
```

**Windows:**

```
./keytool -genkeypair -alias %TH% -keystore %STORES%\%TH%.keystore.jks
-dname "cn=<Connector FQDN>,OU=Arcsight,O=MF,L=Sunnyvale,ST=CA,C=US" -
validity 375
```

---

**NOTE:** For `dname`, the `FQDN`, `OU`, `O`, `L`, `ST` and `C` values must be appropriate for your company and location. For example, `-dname "CN=ig.mf.com,OU=IG,O=MF,L=Sunnyvale,ST=CA,C=US"`

---

**1e2** When prompted, enter the password. Note the password as you will need it in a later step.

---

**NOTE:** Ensure that the password is same as the store password you specified in [Step 1b](#).

---

**1e3** When prompted for the key password, press `Enter` if you want the key password to be same as the keystore password. Save the password. You will need it again in a later step.

**1f** List the keystore entries and verify that you have minimum one private key:

**Linux:**

```
./keytool -list -keystore ${STORES}/${TH}.keystore.jks -storepass
${STORE_PASSWD}
```

**Windows:**

```
.\keytool -list -keystore %STORES%\%TH%.keystore.jks -storepass  
%STORE_PASSWD%
```

### 1g Create a Certificate Signing Request (CSR):

#### Linux:

```
./keytool -certreq -alias ${TH} -keystore ${STORES}/${TH}.keystore.jks -  
file ${STORES}/${TH}-cert-req -storepass ${STORE_PASSWD}
```

#### Windows:

```
.\keytool -certreq -alias %TH% -keystore  
%STORES%\%TH%.keystore.jks -file %STORES%\%TH%-cert-req -storepass  
%STORE_PASSWD%
```

## 2 On the Transformation Hub Server:

### 2a Ensure that the CDF root CA certificate and root CA key used by Transformation Hub are available in /tmp directory with the following names:

```
/tmp/ca.key.pem  
/tmp/ca.cert.pem
```

### 2b Set the environment variables for the static values used by keytool:

```
export CA_CERT_TH=/tmp/ca.cert.pem  
export CA_KEY_TH=/tmp/ca.key.pem  
export CERT_CA_TMP_TH=/opt/cert_ca_tmp  
export TH=<Transformation Hub hostname>_<Transformation Hub port>
```

### 2c Create a temporary directory on the Transformation Hub master server:

```
mkdir $CERT_CA_TMP_TH
```

## 3 Copy the \${STORES}/\${TH}-cert-req file from a Linux based SmartConnector server or %STORES%\%TH%-cert-req file from a Windows based SmartConnector Server to the CERT\_CA\_TMP\_TH directory in the Transformation Hub master server created in [Step 2c](#).

## 4 On the Transformation Hub server, create the signed certificate using the openssl utility:

```
/bin/openssl x509 -req -CA ${CA_CERT_TH} -CAkey ${CA_KEY_TH} -in  
${CERT_CA_TMP_TH}/${TH}-cert-req -out ${CERT_CA_TMP_TH}/${TH}-cert-signed -  
days 366 -CAcreateserial -sha256
```

## 5 On the SmartConnector server:

### 5a Copy the \${TH\_CERT\_CA\_TMP\_TH}/\${TH}-cert-signed and /tmp/ca.cert.pem certificates from the Transformation Hub server to the \${STORES} directory on the Linux based SmartConnector server or %STORES% directory on the Windows based SmartConnector server.

### 5b Import the CDF root CA certificate to the truststore:

#### 5b1 Execute the command:

##### Linux:

```
./keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot  
-keystore ${STORES}/${TH}.truststore.jks -storepass ${STORE_PASSWD}
```

##### Windows:

```
.\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot  
-keystore %STORES%\%TH%.truststore.jks -storepass %STORE_PASSWD%
```

**5b2** When prompted, specify a password for the truststore. Note the password as you will need it again in a later step.

**5b3** When you are asked to trust the certificate, enter *Yes*.

**5c** Import the CDF root CA certificate to the keystore:

**5c1** Execute the command:

**Linux:**

```
./keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -  
keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
```

**Windows:**

```
.\keytool -importcert -file %STORES%\${CA_CERT} -alias CARoot -  
keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
```

**5c2** When you are asked to trust the certificate, enter *Yes*.

**5d** Import the signed certificate to the keystore:

**Linux:**

```
./keytool -importcert -file ${STORES}/${TH}-cert-signed -alias ${TH}  
-keystore ${STORES}/${TH}.keystore.jks -storepass ${STORE_PASSWD}
```

**Windows:**

```
.\keytool -importcert -file %STORES%\%TH%-cert-signed -alias %TH%  
-keystore %STORES%\%TH%.keystore.jks -storepass %STORE_PASSWD%
```

**5e** Note the keystore and truststore paths:

**Linux:**

```
echo ${STORES}/${TH}.truststore.jks  
echo ${STORES}/${TH}.keystore.jks
```

**Windows:**

```
echo %STORES%\%TH%.truststore.jks  
echo %STORES%\%TH%.keystore.jks
```

**5f** Delete the following files:

---

**CAUTION:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

---

**Linux:**

```
rm ${STORES}/${CA_CERT}  
rm ${STORES}/ca.key.pem  
rm ${STORES}/${TH}-cert-signed  
rm ${STORES}/${TH}-cert-req
```

**Windows:**

```
del %STORES%\ca.cert.pem  
del %STORES%\ca.key.pem  
del %STORES%\%TH%-cert-signed  
del %STORES%\%TH%-cert-req
```



- 6 On the Transformation Hub server, delete the `/tmp` folder where the CDF root CA certificate, and root CA key of Transformation Hub are available.

---

**CAUTION:** The temporary certificate folder should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

---

- 7 Continue with [Configuring the SmartConnector](#).

## Configuring the SmartConnector

You must perform the following steps for both the instances of SmartConnector: SmartConnector for audit events collection and SmartConnector for entity change events collection.

- 1 Change to the following directory:  
`<SmartConnector Installation Directory>/current/bin`
- 2 Execute the following command:  
`./runagentsetup.sh`
- 3 Read through the warning, then enter `yes` to continue.
- 4 Specify the following details:
  - 4a Enter a *Unique Generator ID*. If you press Enter without specifying any value here, connector will not have a Generator ID and the *Global Event ID* will be zero.
  - 4b Enter the corresponding number to either enable or disable *FIPS Mode*.
  - 4c Enter the corresponding number to disable *Remote Management*.
  - 4d Press Enter to accept the default port number for *Remote Management Listener Port*.
  - 4e Press Enter to select IPV4 as the *Preferred IP Version*.
  - 4f Enter *Format Preserving Host URL[]*.
  - 4g Press Enter to disable the *Format Preserving Encryption*.
  - 4h Press Enter for *Proxy Host (https)*.
    - 4i Press Enter for *Proxy Port*.
    - 4j Press Enter for *Format Preserving Identity*.
    - 4k Press Enter for *Format Preserving Secret*.
    - 4l Enter `yes` to confirm the values are correct.
  - 4m Read through the message for Unique Generator ID, then enter `yes` to continue.
- 5 Read through the summary, then enter `yes` to continue.
- 6 Select the corresponding number for **Syslog NG Daemon** as the *SmartConnector Type* and enter `yes` to confirm the value.
- 7 Enter the following parameter details:
  - 7a For *Network Port*, enter the port number depending on the communication protocol you want to establish with the data source.

---

**NOTE:** To allow communication, enable this port in the firewall as well.

---

- 7b Specify the *IP address* of the device that the connector should listen to.  
Alternatively, to bind the connector to all available IP addresses, press Enter to accept the default.

7c Specify the protocol.

---

**NOTE:** If you are configuring the SmartConnector to receive [entity change events from Identity Intelligence](#), select **Raw TCP**. If you are configuring the SmartConnector to receive [audit events from Identity Manager](#), select **TLS**.

---

enter the number corresponding to **TLS**.

7d Press Enter to select the default value for *Forwarder*.

7e Press Enter to accept the default value for *IETF Standard (RFC 5424) Enabled*.

7f Enter *yes* to confirm the parameter values are correct.

8 Enter the corresponding number for **Transformation Hub** as the *destination type*.

9 Configure the **destination** parameters:

9a For *Initial Host:Port(s)*, enter the *FQDN* and *port* of Kafka.

- ♦ For *Raw TCP*:

*<kafka\_host\_name>*:9092

- ♦ For *SSL/TLS*:

*<kafka\_host\_name>*:9093

---

**NOTE:** Ensure that the FQDNs of Kafka nodes resolve successfully.

---

9b Press Enter to accept the default *content type*.

9c Press Enter to accept *th-cef* as the default *Topic*.

9d Press Enter to accept the default *ESM version*.

9e Enter the corresponding number to select the *Acknowledgment mode* as *none*.

9f (Conditional) If you want to configure [one-way SSL authentication](#) with Transformation Hub:

Enter the number corresponding to *true* for **Use SSL/TLS** and provide the following information:

9f1 **SSL/TLS Trust Store file:** Specify the full path to the [truststore file](#) that contains the CDF root CA certificate.

9f2 **SSL/TLS Trust Store password:** Specify the password used to access the truststore file that contains the CDF root CA certificate.

9g (Conditional) If you want to configure [mutual SSL authentication](#) with Transformation Hub:

Enter the number corresponding to *true* for **Use SSL/TLS** and provide the following information:

9g1 **SSL/TLS Trust Store file:** Specify the full path to the [truststore file](#) that contains the CDF root CA certificate.

9g2 **SSL/TLS Trust Store password:** Specify the password used to access the truststore file that contains the CDF root CA certificate.

9g3 **Use SSL/TLS Authentication:** Select *yes* if you want to Transformation Hub to authenticate SmartConnector.

9g4 **SSL/TLS Key Store file:** Specify the full path of the [keystore file](#) that contains SSL private key and certificate.

9g5 **SSL/TLS Key Store pass:** Specify the password to access the keystore file.

9g6 **SSL/TLS Key password:** Specify the password to access the private key.

9h Enter *yes* to confirm the destination parameter values are correct.

- 10 Enter the connector details:
  - 10a Enter the *name* of the SmartConnector.
  - 10b (Optional) Enter the *Location*, *DeviceLocation*, and *Comment* for the SmartConnector.
  - 10c Verify the connector details and enter *yes* to confirm the values are correct and then enter *yes* to continue.
- 11 To run the connector as a service, enter the corresponding number for *Install as a service*.
- 12 Specify the service parameters.
  - 12a Enter a unique *Service Internal Name*.
  - 12b Enter a unique *Service Display Name*.
  - 12c Press Enter to accept the default settings to start the service automatically.
  - 12d Verify the service parameters and enter *yes* to confirm the values are correct.
- 13 Read the *Install Service Summary*, press Enter to continue.
- 14 Start the SmartConnector using the following command:
 

```
/etc/init.d/arc_<name of the SmartConnector> start
```
- 15 (Conditional) If you want to configure Identity Manager to send audit events to multiple destinations such as Sentinel Log Manager for IGA and Identity Intelligence through the SmartConnector, complete the tasks mentioned in “[Configuring Additional Destination for Identity Manager Audit Events](#)” on page 83.

For more information, see the [SmartConnector for Syslog NG Daemon Configuration Guide](#) and the “Configuring Connectors” section in the [SmartConnector User Guide](#).

## Configuring Additional Destination for Identity Manager Audit Events

*Applies only if you are configuring [audit events collection from Identity Manager](#).*

To collect audit events from Identity Manager, you must configure Identity Applications and Identity Manager Engine to send audit events to the SmartConnector. If you had configured Identity Applications and Identity Manager Engine to send audit events to other destination (for example, Sentinel Log Management for IGA), you must add another destination for the SmartConnector in addition to Transformation Hub.

- 1 Change to the following directory:
 

```
<SmartConnector Installation Directory>/current/bin
```
- 2 Run the following command:
 

```
./runagentsetup.sh
```
- 3 Enter the corresponding number to *Modify Connector*.
- 4 Enter the corresponding number to *Add, modify, or remove destinations*.
- 5 Enter the corresponding number to *Add destination*.
- 6 Enter the corresponding number to select *CEF Syslog* as the *destination*.
- 7 Specify values for the destination parameters.
- 8 To complete adding the *destination*, follow the prompts.
- 9 Continue with [Verifying the SmartConnector Configuration](#).

# Verifying the SmartConnector Configuration

To verify whether the SmartConnector is configured correctly, you can check the `<SmartConnector Installation Directory>/current/logs/agentsetup.log` file.

# 14 Configuring Entity Change Events Collection

To ensure that [Views](#) provide information about changes to entity data, you must configure Identity Intelligence for entity change events collection.

- ♦ [“Understanding the Collection of Entity Change Events” on page 85](#)
- ♦ [“Configuring the Collection of Entity Change Events” on page 85](#)

## Understanding the Collection of Entity Change Events

The Entity Data Model component in Identity Intelligence generates and stores entity change events as follows:


- ♦ When there are changes to entity data, such as addition, deletion, modification, or change in relationships in the data source, the Entity Data Model component generates entity change events in common event format (CEF) to track any type of changes to entity data.
- ♦ The Entity Data Mode component sends these events to the SmartConnector for Syslog NG Daemon.
- ♦ SmartConnector sends the events to the `th-cef` Kafka topic in Transformation Hub.
- ♦ Transformation Hub stores events in Vertica.

## Configuring the Collection of Entity Change Events

Identity Intelligence uses the SmartConnector for Syslog NG Daemon to send entity change events to Transformation Hub. You must install and configure the SmartConnector before performing this task. For more information, see [Chapter 13, “Installing and Configuring the SmartConnector,” on page 75](#).

- 1 Log in to the CDF Management Portal.

The CDF Management Portal uses the same credentials that you specified when you [installed CDF](#). The first time that you log in to the portal, you might be prompted to change the administrator password.

- 2 Click  of `arcsight-installer`, then click **Reconfigure**.
- 3 Select **Identity Intelligence**.
- 4 Enable **Monitor data changes** under the **Data Change Monitoring Configuration** section.
- 5 Specify the details for the SmartConnector server.
- 6 Click **Save**.



# 15 Customizing Data Reconciliation

Data sources such as Identity Manager and Identity Governance perform data reconciliation when they ingest data from their data sources. Similarly, Identity Intelligence can reconcile data received from its data sources. For example, when Identity Intelligence collects data from both Identity Manager and Identity Governance, the two data sources might send records for the same entity, such as a user or a role. Rather than having duplicate entries, Identity Intelligence associates those multiple records to a single entity. Identity Intelligence bases data reconciliation on certain unique fields for each entity type: identity, identity group, entitlement, account, and application.

- ♦ [“Understanding Default Reconciliation Fields” on page 87](#)
- ♦ [“Reconciling Data for the Identity Entity” on page 87](#)
- ♦ [“Customizing Data Reconciliation” on page 88](#)

## Understanding Default Reconciliation Fields

The following table lists the default reconciliation fields for each entity type:

| Entity         | Default Data Reconciliation Field   |
|----------------|---|
| Identity       | See <a href="#">“Reconciling Data for the Identity Entity” on page 87</a> |
| Identity Group | identitygroup_id  |
| Entitlement    | entitlement_id  |
| Account        | account_domain@Account, external_id_value                                 |
| Application    | application_name  |

## Reconciling Data for the Identity Entity

By default, the `entity_reconciliation_id` field is not populated with any value. Therefore, by default, Identity Intelligence does not do reconciliation for identity records.

You can either map `entity_reconciliation_id` to the appropriate attribute in the data source or use any of the attributes available in the Identity Intelligence schema.

- ♦ **Use `entity_reconciliation_id`:** You can use `entity_reconciliation_id` in situations such as multi-affiliation where a person might have multiple User objects in the data source. Mapping `entity_reconciliation_id` with the correct attribute in the data source helps Identity Intelligence to correctly track those user objects separately.

To map `entity_reconciliation_id` in the original data source, you must specify the appropriate attributes for the `entity_reconciliation_id` field. If you have multiple identity data sources, ensure that the `entity_reconciliation_id` is mapped to the same attribute across all data sources. Complete the instructions referenced in the following table for your data sources:

| Data Source         | Configuration Instructions   |
|---------------------|--|
| Identity Manager    | <a href="#">“Creating the Driver Object” in the Identity Manager Driver for Entity Data Model Implementation Guide</a> |
| Identity Governance | <a href="#">“Mapping Attributes for Data Reconciliation” on page 104</a>   |

- ♦ **Use attributes available in the schema:** If you do not need to handle situations such as multi-affiliation in your environment, you can modify the `tg_reconciliation_fields.json` file with any of the attributes available in the `mf_shared` schema instead of populating the `entity_reconciliation_id` field.

## Customizing Data Reconciliation

To customize data reconciliation, update the `tg_reconciliation_fields.json` file with the desired reconciliation fields. The naming convention of the field names must be the same as what is used in the `mf_shared` schema in Vertica.

In the `mf_shared` schema, refer to the respective entity tables to get field names.

| Entity        | Table Name                                  |
|---------------|---|
| Identity      | <code>mf-shared-entity-identity</code>      |
| IdentityGroup | <code>mf-shared-entity-identitygroup</code> |
| Account       | <code>mf-shared-entity-externalid</code>    |
| Entitlement   | <code>mf-shared-entity-entitlement</code>   |
| Application   | <code>mf-shared-entity-application</code>   |

You can access the `mf_shared` schema and the tables by using the `\dn` and `\dt` commands respectively. For more information, see [Vertica documentation](#).

- 1 Log in to the Identity Intelligence server as the `root` user.
- 2 Change to the following directory:

### For single node deployments

```
cd /<install_directory>/<arcsight_nfs_volume>/eventbroker/di/data-processor/conf
```

### For multi-node deployments

```
cd /<install_directory>/<nfs_share>/<arcsight_nfs_volume>/eventbroker/di/data-processor/conf
```

- 3 Edit the data reconciliation fields in the `tg_reconciliation_fields.json` file.



# 16 Configuring Data Collection from Identity Manager

This chapter provides information about collecting entity data and audit events from Identity Manager:

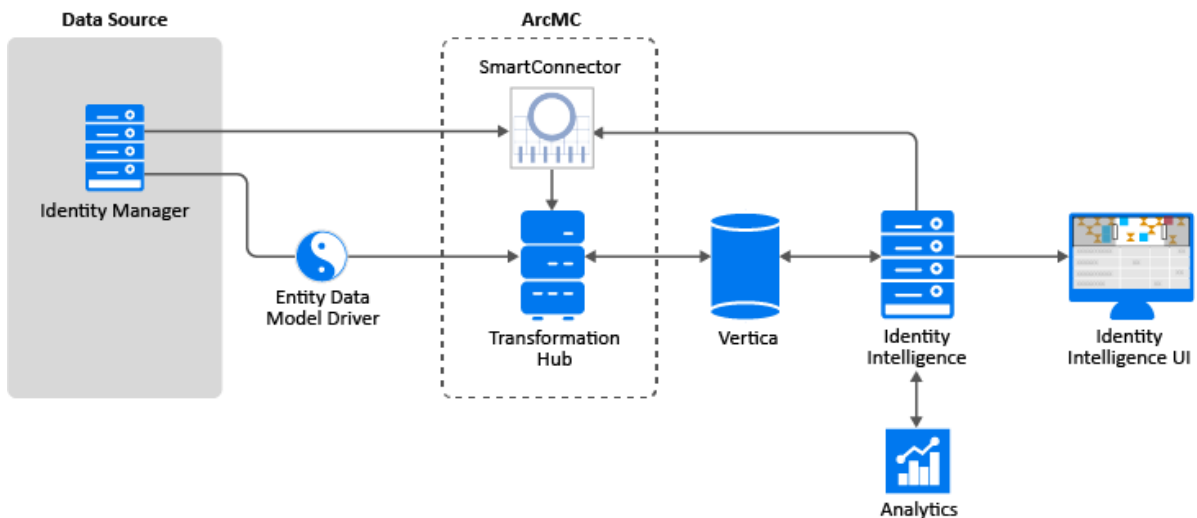
- ♦ “Understanding the Data Collection Process” on page 89
- ♦ “Configuring Entity Data Collection” on page 90
- ♦ “Configuring Audit Events Collection” on page 90
- ♦ “Reverting Backdated Events Configuration” on page 95

## Understanding the Data Collection Process

Identity Intelligence collects, processes, and stores entity data and audit events as follows:

- ♦ The Identity Manager Driver for Entity Data Model collects **entity data** from Identity Manager. The Entity Data Model driver collects various types of entity-specific data such as identities, accounts, and access rights from Identity Manager and sends the data to the `mf-shared-entity-ingest` Kafka topic in Transformation Hub.
- ♦ Identity Manager sends **audit events** to the SmartConnector for Syslog NG Daemon, which then sends these audit events to the `th-cef` Kafka topic in Transformation Hub.
- ♦ Identity Intelligence does the following:
  - ♦ Gathers data from Kafka topics.
  - ♦ Standardizes and processes data to build data structures for analytics use cases.
  - ♦ Sends the processed data to Vertica for storage.

The following diagram helps you understand the data collection process:



# Configuring Entity Data Collection

Configure the Entity Data Model driver to collect entity data from Identity Manager. For more information, see the [Identity Manager Driver for Entity Data Model Implementation Guide](#).

## Configuring Audit Events Collection

Identity Intelligence uses the SmartConnector for Syslog NG Daemon to collect audit events from Identity Manager. Identity Manager sends audit events to the SmartConnector, which then sends these audit events to the `th-cef` Kafka topic in Transformation Hub. The SmartConnector collects audit events that are in Common Event Format (CEF). Therefore, you must configure the Identity Manager Engine and Identity Applications to log audit events in CEF.

- ◆ [“Prerequisites” on page 90](#)
- ◆ [“Obtaining the SmartConnector Certificate” on page 90](#)
- ◆ [“Configuring Identity Applications” on page 91](#)
- ◆ [“Configuring the Identity Manager Engine” on page 93](#)
- ◆ [“Audit Events Used by Identity Intelligence” on page 94](#)

### Prerequisites

- ◆ To configure audit events collection, you must install and configure the SmartConnector for Syslog NG Daemon. For more information, see [Configuring the SmartConnector](#).

You must configure Identity Applications and Identity Manager Engine to send audit events to the SmartConnector. Therefore, if you had configured Identity Applications and Identity Manager Engine to send audit events to other destinations such as Sentinel Log Management for IGA, you must add two destinations for the SmartConnector:

- ◆ Transformation Hub: for SmartConnector to send audit events to the `th-cef` Kafka topic in Transformation Hub.
- ◆ CEF Syslog: for SmartConnector to send audit events to the existing destination configured in Identity Applications and Identity Manager Engine.

For more information, see [Installing and Configuring the SmartConnector](#).

- ◆ To configure audit events collection using SSL:
  - ◆ Ensure that you are using Identity Manager version 4.7.4 or later.
  - ◆ Complete the tasks mentioned in the [“Obtaining the SmartConnector Certificate” on page 90](#) section.
- ◆ Ensure to enable ingestion of backdated data to Vertica. For more information, see [“Tuning Ingestion of Backdated Events” on page 73](#).

### Obtaining the SmartConnector Certificate

The SmartConnector and some of the Identity Manager components utilize embedded certificates generated by an internal Certificate Authority (CA). These SSL certificates ensure that communication between the Identity Manager components and the SmartConnector is secure.

- ◆ [Obtaining the Certificate from Browser](#)
- ◆ [Obtaining the Certificate from Command-Line](#)

## Obtaining the Certificate from Browser

You can obtain the SmartConnector certificate from a Web browser. This section provides information about obtaining the SmartConnector certificate in Google Chrome.

- 1 Specify the following URL in the browser:  
`https://<smartconnector_node_hostname>:<port>`
- 2 Click the icon next to the left of the URL, then click **Certificate**.
- 3 Click **Certification Path** and select the CA certificate.
- 4 For the selected certificate, click **Details**, then click **Copy to File...**
- 5 Click **Next**.
- 6 Select **Base-64 encoded X.509 (.CER)** and click **Next**.
- 7 Specify a file name (for example, `smartconn.cer`) and click **Next**.
- 8 Click **Finish**.

## Obtaining the Certificate from Command-Line

- 1 Log in to the machine where you have installed SmartConnector.
- 2 Execute the following command to obtain the certificate:

```
echo | openssl s_client -connect <smartconnector ip>:<port> 2>&1 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > smartconn.cer
```

## Configuring Identity Applications

The configuration settings for the Identity Applications logging are stored in the Identity Manager server in the `idmuserapp_logging.xml` and `workflow_logging.xml` files located in the following path. You must update these files to include the SmartConnector details.

**Linux:** `/opt/netiq/idm/apps/tomcat/conf`

**Windows:** `C:\netiq\idm\apps\tomcat\conf`

- 1 (Conditional) To configure audit events collection using SSL, perform the following:
  - 1a Navigate to `/opt/netiq/common/jre/bin`
  - 1b Copy the **SmartConnector certificate** in the Identity Manager machine and add it to the KeyStore file using the following command:  

```
./keytool -import -file <smartconnector_certificate> -keystore <keystore_file> -storepass <keystore_password>
```
  - 1c Change the ownership of the keystore file to `novlua`:  

```
chown novlua:novlua <keystore_file>
```
- 2 To log audit events in CEF, edit the `idmuserapp_logging.xml` file as follows:

Remove the **<!-- remove this line to turn on CEF auditing and remove this line to turn on CEF auditing -->** lines from the following `<logger>` sections:

```
<logger name="com.novell" level="INFO" additivity="true">
  <!-- remove this line to turn on Novell Audit
  <appender-ref ref="NAUDIT"/>
  remove this line to turn on Novell Audit -->
  <!-- remove this line to turn on CEF auditing
  <appender-ref ref="CEF"/>
  remove this line to turn on CEF auditing -->
</logger>
<logger name="com.sssw" level="INFO" additivity="true">
  <!-- remove this line to turn on Novell Audit
  <appender-ref ref="NAUDIT"/>
  remove this line to turn on Novell Audit -->
  <!-- remove this line to turn on CEF auditing
  <appender-ref ref="CEF"/>
  remove this line to turn on CEF auditing -->
</logger>
<logger name="com.netiq" level="INFO" additivity="true">
  <!-- remove this line to turn on Novell Audit
  <appender-ref ref="NAUDIT"/>
  remove this line to turn on Novell Audit -->
  <!-- remove this line to turn on CEF auditing
  <appender-ref ref="CEF"/>
  remove this line to turn on CEF auditing -->
</logger>
```

3 Save and close the file.

4 To log workflow events in CEF, edit the `workflow_logging.xml` file as follows:

Remove the **<!-- remove this line to turn on CEF audit and remove this line to turn on CEF audit -->** lines from the following `<logger>` sections:

```
<logger name="workflow.log" level="INFO" additivity="true">
  <!-- remove this line to turn on CEF Audit
  <appender-ref ref="WFCEF"/>
  remove this line to turn on CEF Audit -->
</logger>
<logger name="com.novell" level="INFO" additivity="true">
  <!-- remove this line to turn on CEF Audit
  <appender-ref ref="WFCEF"/>
  remove this line to turn on CEF Audit -->
</logger>
<logger name="com.netiq" level="INFO" additivity="true">
  <!-- remove this line to turn on CEF Audit
  <appender-ref ref="WFCEF"/>
  remove this line to turn on CEF Audit -->
</logger>
<logger name="com.sssw" level="INFO" additivity="true">
  <!-- remove this line to turn on CEF Audit
  <appender-ref ref="WFCEF"/>
  remove this line to turn on CEF Audit -->
</logger>
<logger name="com.microfocus" level="INFO" additivity="true">
  <!-- remove this line to turn on CEF Audit
  <appender-ref ref="WFCEF"/>
  remove this line to turn on CEF Audit -->
</logger>
```

5 Save and close the file.

- 6 Specify the SmartConnector details:
  - 6a Log in to Identity Manager Dashboard.
  - 6b Navigate to **Configuration > Logging > Auditing Configuration**.
  - 6c Select **Enable CEF format** to log the events in CEF.
  - 6d In the **Destination host**, specify the host name of the SmartConnector server.
  - 6e In the **Destination port**, specify the port number of the SmartConnector server.
  - 6f Select the network protocol from the drop-down list.
  - 6g (Conditional) Enable **Use TLS** for SSL communication.  
If you enabled SSL, specify the file path and password of the [keystore file](#).
  - 6h In the **Intermediate event store directory**, specify the cache file path to store the events until the connection is established.  
  
Ensure the cache file path (`/opt/netiq/idm/apps/tomcat/cache`) you are specifying exists and the user `novlua` have owner permission to that folder.
  - 6i Click **Apply**.
- 7 Restart the Tomcat service for Identity Manager:
 

```
systemctl restart netiq-tomcat.service
```

For more information, see [Configuring Identity Applications](#) and [Understanding the idmuserapp\\_logging.xml File](#) sections in the *Identity Manager Administrator's Guide to Configure Auditing*.

## Configuring the Identity Manager Engine

Before you configure the Identity Manager Engine to log audit events in CEF, you must add the SmartConnector details in the [auditlogconfig.properties](#) file so that the Identity Manager Engine can send CEF events to the specified SmartConnector.

**To add the SmartConnector details in the auditlogconfig.properties file:**

- 1 Log in to the Identity Manager server as the `root` user.
- 2 Change to the following directory:
 

```
cd /etc/opt/novell/eDirectory/conf
```
- 3 Open the `auditlogconfig.properties` file.
- 4 Update the file as indicated in the following source snippet:
 

```
# Set the level of the root logger to DEBUG and attach appenders.
log4j.rootLogger=debug, S, R

# Defines appender S to be a SyslogAppender.
log4j.appender.S=org.apache.log4j.net.SyslogAppender

# Defines location of Syslog server.
log4j.appender.S.Host=<ip address or hostname of the SmartConnector>
log4j.appender.S.Port=<Port of the SmartConnector>

# Specify protocol to be used (UDP/TCP/SSL)
log4j.appender.S.Protocol=<Protocol of the SmartConnector>

# Specify SSL certificate file for SSL connection.
# File path should be given with double backslash.
```

```

log4j.appender.S.SSLCertFile=<Certificate of the SmartConnector in .pem format>

# Minimum log-level allowed in syslog.
log4j.appender.S.Threshold=INFO

# Defines the type of facility.
log4j.appender.S.Facility=USER

# Defines Caching for SyslogAppender.
# Inputs should be yes/no
log4j.appender.S.CacheEnabled=yes

# Cache location directory
# Directory should be available for creating cache files
log4j.appender.S.CacheDir=<cache files directory>

# Cache File Size
# Cache File size should be in the range of 50MB to 4000 MB
log4j.appender.S.CacheMaxFileSize=500MB

# Layout definition for appender Syslog S.
log4j.appender.S.layout=org.apache.log4j.PatternLayout
log4j.appender.S.layout.ConversionPattern=%c: %m%n

# Defines appender R to be a Rolling File Appender.
log4j.appender.R=org.apache.log4j.RollingFileAppender

# Log file for appender R.
log4j.appender.R.File=<directory of log file appender>

# Max size of log file for appender R.
log4j.appender.R.MaxFileSize=100MB

# Set the maximum number of backup files to keep for appender R.
# Max can be 13. If set to zero, then there will be no backup files.
log4j.appender.R.MaxBackupIndex=10

# Layout definition for appender Rolling log file R.
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{MMM dd HH:mm:ss} %c %m%n

```

## 5 Configure the Identity Manager Engine to log events in CEF.

---

**IMPORTANT:** While configuring the Identity Manager Engine, when you select **Log specific events**, by default all the necessary events are sent to the SmartConnector. However, ensure to select at least one event in [Step 8](#). For information about audit events that Identity Intelligence uses, see [“Audit Events Used by Identity Intelligence” on page 94](#).

---

### 6 Stop the Identity Manager Engine using the following command:

```
ndsmanage stopall
```

### 7 Start the Identity Manager Engine using the following command:

```
ndsmanage startall
```

### 8 Configure the collection of entity change events.

## Audit Events Used by Identity Intelligence

The following table lists the events that Identity Intelligence uses from Identity Manager:

| Event ID | Description               | Trigger   |
|----------|---------------------------|---|
| 31520    | Workflow Error            | Occurs when there is a workflow error                             |
| 31521    | Workflow Started          | Occurs when the workflow starts                                   |
| 31522    | Workflow Forwarded        | Occurs when the workflow is forwarded                             |
| 31523    | Workflow Reassigned       | Occurs when the workflow is reassigned                            |
| 31524    | Workflow Approved         | Occurs when the workflow is approved                              |
| 31525    | Workflow Refused          | Occurs when the workflow is refused                               |
| 31526    | Workflow Ended            | Occurs when the workflow ends                                     |
| 31527    | Workflow Claimed          | Occurs when the workflow is claimed                               |
| 31528    | Workflow Unclaimed        | Occurs when the workflow is not claimed                           |
| 31529    | Workflow Denied           | Occurs when the workflow is denied                                |
| 31533    | Workflow Retracted        | Occurs when the workflow is retracted                             |
| 31534    | Workflow Escalated        | Occurs when the workflow is escalated                             |
| 31535    | Workflow Reminder Sent    | Occurs when reminders are sent to addressees of a workflow task   |
| 31610    | Role Request              | Occurs when a role is requested                                   |
| 31611    | Role Request Failure      | Occurs when the request for a role fails                          |
| 31614    | Retract Role Request      | Occurs when the role request is retracted                         |
| 3152B    | Workflow Timedout         | Occurs when the workflow timed out                                |
| 31663    | Retract Resource Request  | Occurs when the resource request is canceled.                     |
| 31660    | Resource Request          | Occurs when a resource is requested                               |
| 3152C    | User Message              | This is a user adhoc log message                                  |
| 31662    | Resource Request Workflow | Occurs when a resource with approval process is initiated         |
| 31612    | Role Request Workflow     | Occurs when the approval workflow is initiated for a role request |

## Reverting Backdated Events Configuration

The [events Kafka scheduler](#) was tuned to read correct time stamp of the event data before migration. Hence after data migration, you must revert events Kafka scheduler to the default configuration to avoid any performance issue. For information about how to revert the Kafka scheduler configuration, see [Reverting Backdated Events Configuration](#).





# 17 Configuring Data Collection from Identity Governance

This chapter provides information about configuring data collection from Identity Governance.

- ♦ [“Data Collection Configuration Checklist” on page 97](#)
- ♦ [“Understanding the Data Collection Process” on page 97](#)
- ♦ [“Configuring Data Collection” on page 98](#)

## Data Collection Configuration Checklist

To successfully configure data collection, complete the following checklist in the listed order:

---

| Task  |
|---|
| <input type="checkbox"/> 1. <a href="#">Understand the data collection process</a>                            |
| <input type="checkbox"/> 2. <a href="#">Use the Identity Governance configuration utility to create facts</a> |
| <input type="checkbox"/> 3. <a href="#">Map the attributes for data reconciliation</a>                        |
| <input type="checkbox"/> 4. <a href="#">Collect metrics from Identity Governance</a>                          |

---

## Understanding the Data Collection Process

Identity Governance collects all user and access information, also called facts, from various data sources and stores this data in fact tables. Identity Intelligence collects the following data from these facts tables:

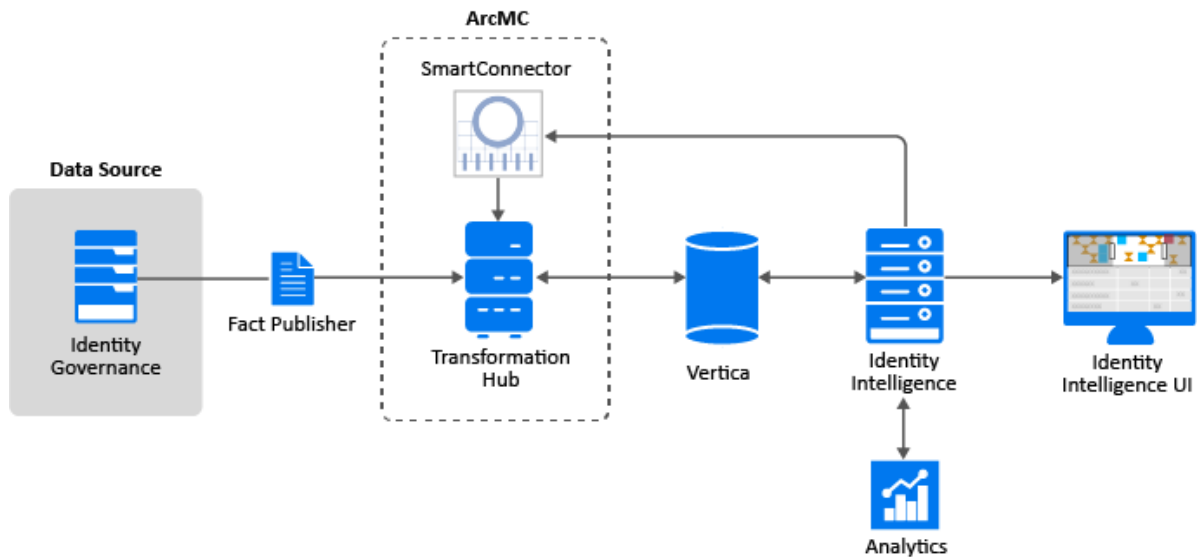
- ♦ Entities and their relations such as identities, groups, accounts, and applications.
- ♦ Entitlements (access rights) such as permissions and technical roles.
- ♦ User activities such as access requests and access reviews.

Identity Intelligence collects, processes, and stores the data as follows:

- ♦ Identity Intelligence provides a fact configuration utility that enables you to create facts in Identity Governance to collect the required data.
- ♦ Identity Governance publishes snapshots of these facts at the scheduled interval to Kafka topics in Transformation Hub.
- ♦ Identity Intelligence gathers data from Kafka topics.
- ♦ Identity Intelligence compares the existing and new data snapshots to check for any new data and for modification or deletion to existing data.
- ♦ Based on the comparison, it updates the *mf-shared-entity-ingest* Kafka topic in Transformation Hub.

- ◆ Standardizes and processes data in the *mf-shared-entity-ingest Kafka* topic to build data structures for analytics use cases.
- ◆ Sends the processed data to Vertica for storage.

The following diagram helps you understand the data collection process:



## Configuring Data Collection

- ◆ [“Prerequisites” on page 98](#)
- ◆ [“Configuring One-Way SSL Between Identity Governance and Transformation Hub” on page 99](#)
- ◆ [“Configuring Mutual SSL Between Identity Governance and Transformation Hub” on page 100](#)
- ◆ [“Creating Fact Configuration Files” on page 104](#)
- ◆ [“Mapping Attributes for Data Reconciliation” on page 104](#)
- ◆ [“Collecting Data from Identity Governance” on page 106](#)

## Prerequisites

Complete the following prerequisites before configuring data collection from Identity Governance:

- ◆ Download and install Python 2.7.61 or later. This prerequisite is applicable only if you want to run the fact configuration utility on Windows or older versions of Linux machines. RHEL 7.6 and later already have the appropriate version of Python.
- ◆ Add the Python executable to the PATH environment variable in the computer where you configure data collection.
- ◆ Enable entity change events collection. For more information, see [“Configuring the Collection of Entity Change Events” on page 85](#).
- ◆ Ensure to enable ingestion of backdated data to Vertica. For more information, see [“Tuning Ingestion of Backdated Events” on page 73](#).

# Configuring One-Way SSL Between Identity Governance and Transformation Hub

If you want to configure [one-way authenticated SSL](#) communication between Identity Governance and Transformation Hub, perform the following:

- 1 Retrieve the CDF root CA certificate as described in [Retrieving CDF Root CA](#).
- 2 Copy the CDF root CA certificate file to a temporary location in the Identity Governance server machine.
- 3 Log in to the Identity Governance server machine as a user with `root` access on a Linux server or administrative privileges on a Windows server. From a command prompt, change to the installation directory of the Java `keytool` utility.

The default installation directory is:

- ♦ **Linux:** `/opt/netiq/idm/apps/jre/bin`
- ♦ **Windows:** `c:\netiq\idm\apps\jre\bin`

- 4 Using the `keytool` utility, import the CDF root CA certificate to a truststore file:

4a Run the following command:

- ♦ **Linux:** `./keytool -import -file <cert_file> -alias <alias> -keystore <trust_store_file>`
- ♦ **Windows:** `.\keytool -import -file <cert_file> -alias <alias> -keystore <trust_store_file>`

4b When prompted, specify a password for the truststore. Note the password; you will need it again in a later step.

4c When you are asked to trust the certificate, enter `Yes`.

- 5 Using the `keytool` utility, create a KeyStore file:

5a Run the following command:

- ♦ **Linux:** `./keytool -keystore <key_store_file> -alias <alias> -dname "CN=<ig_server_fqdn>,OU=<organizational_unit_name>,O=<organization_name>,L=<city_name>,ST=<state_name>,C=<two_letter_country_code>" -validity <validity_in_days> -genkeypair -keyalg RSA`
- ♦ **Windows:** `.\keytool -keystore <key_store_file> -alias <alias> -dname "CN=<ig_server_fqdn>,OU=<organizational_unit_name>,O=<organization_name>,L=<city_name>,ST=<state_name>,C=<two_letter_country_code>" -validity <validity_in_days> -genkeypair -keyalg RSA`

---

**NOTE:** In the `dname` option, the FQDN, OU, O, L, ST, and C values must be appropriate for your company and location. For example, `-dname`

`"CN=ig.mf.com,OU=IG,O=MF,L=Sunnyvale,ST=CA,C=US"`

---

5b When prompted, specify a password for the KeyStore. Note the password; you will need it again in a later step.

5c When prompted for the key password, press `Enter` if you want the key password to be same as the KeyStore password. Note the password; you will need it again in a later step.

- 6 As mentioned in step 8 of the [Creating Custom Metrics](#) section of the *Identity Governance User and Administration Guide*, use the [Identity Governance Configuration Utility](#) to configure the following properties in the Identity Governance server:

| Property   | Value  |
|--|--|
| <code>com.netiq.iac.kafka.publisher.truststore.location</code> | Absolute path of the truststore file created in Step 4 |
| <code>com.netiq.iac.kafka.publisher.truststore.password</code> | The truststore password specified in Step 4b           |
| <code>com.netiq.iac.kafka.publisher.keystore.location</code>   | Absolute path of the KeyStore file created in Step 5   |
| <code>com.netiq.iac.kafka.publisher.keystore.password</code>   | The keystore password specified in Step 5b             |
| <code>com.netiq.iac.kafka.publisher.key.password</code>        | The key password specified in Step 5c                  |

## Configuring Mutual SSL Between Identity Governance and Transformation Hub

If client authentication is enabled in Transformation Hub, perform the following steps to configure [mutual authenticated SSL](#) between Identity Governance and Transformation Hub:

- 1 On the Identity Governance Server:
  - 1a Log in to the Identity Governance Server machine as a user with root access on a Linux server or administrative privileges on a Windows server.
  - 1b Set the environment variables for the static values used by `keytool`:

**Linux:**

```
export IG=<identity governance server hostname>
export STORES=<directory where keystore and truststore will be created>
export CA_CERT=ca.cert.pem
```

**Windows:**

```
set IG=<identity governance server hostname>
set STORES=<directory where keystore and truststore will be created>
set CA_CERT=C:\Temp\ca.cert.pem
```
  - 1c (Conditional) Create the `STORES` directory if it does not exist:

**Linux:**

```
mkdir ${STORES}
```

**Windows:**

```
mkdir %STORES%
```
  - 1d From a command prompt, change to the installation directory of the `keytool` utility. The default installation directory is:

**Linux:**

```
/opt/netiq/idm/apps/jre/bin
```

**Windows:**

c:\netiq\idm\apps\jre\bin

**1e** Using the `keytool` utility, create the Identity Governance key pair in a keystore file:

**1e1** Execute the command:

**Linux:**

```
./keytool -genkeypair -alias ${IG} -keystore ${STORES}/  
${IG}.keystore.jks -dname  
"CN=<ig_server_fqdn>,OU=<organizational_unit_name>,O=<organization_nam  
e>,L=<city_name>,ST=<state_name>,C=<two_letter_country_code>" -  
validity <validity_in_days>
```

**Windows:**

```
.\keytool -genkeypair -alias %IG% -keystore %STORES%\%IG%.keystore.jks  
-dname  
"CN=<ig_server_fqdn>,OU=<organizational_unit_name>,O=<organization_nam  
e>,L=<city_name>,ST=<state_name>,C=<two_letter_country_code>" -  
validity <validity_in_days>
```

---

**NOTE:** In the `dname`, the FQDN, OU, O, L, ST and C values must be appropriate for your company and location. For example, `-dname "CN=ig.mf.com,OU=IG,O=MF,L=Sunnyvale,ST=CA,C=US"`

---

**1e2** When prompted, specify a password for the keystore. Save the password. You will need it again in a later step.

**1e3** When prompted for the key password, press `Enter` if you want the key password to be same as the keystore password. Save the password. You will need it again in a later step.

**1f** List the keystore entries and verify that you have minimum one private key:

**Linux:**

```
./keytool -list -keystore ${STORES}/${IG}.keystore.jks -storepass <keystore  
password>
```

**Windows:**

```
.\keytool -list -keystore %STORES%\%IG%.keystore.jks -storepass <keystore  
password>
```

**1g** Create a Certificate Signing Request (CSR):

**Linux:**

```
./keytool -certreq -alias ${IG} -keystore ${STORES}/${IG}.keystore.jks -  
file ${STORES}/${IG}-cert-req -storepass <keystore password> -keypass <key  
password>
```

**Windows:**

```
.\keytool -certreq -alias %IG% -keystore %STORES%\%IG%.keystore.jks -file  
%STORES%\%IG%-cert-req -storepass <keystore password> -keypass <key  
password>
```

**2** On the Transformation Hub Server:

**2a** Ensure that the CDF root CA certificate and root CA key used by Transformation Hub are available in `/tmp` directory with the following names:

```
/tmp/ca.cert.pem
```

```
/tmp/ca.key.pem
```

**2b** Set the environment variables for the static values used by `keytool`:

```
export CA_CERT=/tmp/ca.cert.pem
export CA_KEY=/tmp/ca.key.pem
export IG_CERT_CA_TMP=/opt/ig_cert_ca_tmp
export IG=<identity governance server hostname>
```

**2c** Create a temporary directory on the Transformation Hub master server:

```
mkdir $IG_CERT_CA_TMP
```

**3** Copy the `${STORES}/${IG}-cert-req` file from a Linux based Identity Governance server or `%STORES%\%IG%-cert-req` file from a Windows based Identity Governance Server to the `IG_CERT_CA_TMP` directory in the Transformation Hub master server created in step 2c

**4** On the Transformation Hub Server, create the signed certificate using the `openssl` utility:

```
/bin/openssl x509 -req -CA ${CA_CERT} -CAkey ${CA_KEY} -in ${IG_CERT_CA_TMP}/${IG}-cert-req -out ${IG_CERT_CA_TMP}/${IG}-cert-signed -days <validity_in_days> -CAcreateserial -sha256
```

**5** On the Identity Governance server:

**5a** Copy the `${IG_CERT_CA_TMP}/${IG}-cert-signed` and `/tmp/ca.cert.pem` certificates from the Transformation Hub server to the `${STORES}` directory on the Linux based Identity Governance server or `%STORES%` directory on the Windows based Identity Governance server.

**5b** Import the CDF root CA certificate to the truststore:

**5b1** Execute the command:

**Linux:**

```
./keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -keystore ${STORES}/${IG}.truststore.jks
```

**Windows:**

```
.\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot -keystore %STORES%\%IG%.truststore.jks
```

**5b2** When prompted, specify a password for the truststore. Note the password as you will need it again in a later step

**5b3** When you are asked to trust the certificate, enter *Yes*.

**5c** Import the CDF root CA certificate to the keystore:

**5c1** Execute the command:

**Linux:**

```
./keytool -importcert -file ${STORES}/${CA_CERT} -alias CARoot -keystore ${STORES}/${IG}.keystore.jks -storepass <keystore password>
```

**Windows:**

```
.\keytool -importcert -file %STORES%\%CA_CERT% -alias CARoot -keystore %STORES%\%IG%.keystore.jks -storepass <keystore password>
```

**5c2** When you are asked to trust the certificate, enter *Yes*.

**5d** Import the signed certificate to the keystore:

**Linux:**

```
./keytool -importcert -file ${STORES}/${IG}-cert-signed -alias ${IG} -keystore ${STORES}/${IG}.keystore.jks -storepass <keystore password>
```

**Windows:**

```
.\keytool -importcert -file %STORES%\%IG%-cert-signed -alias %IG% -keystore %STORES%\%IG%.keystore.jks -storepass <keystore password>
```

**5e** Note the keystore and truststore file paths:

**Linux:**

```
echo ${STORES}/${IG}.truststore.jks  
echo ${STORES}/${IG}.keystore.jks
```

**Windows:**

```
echo %STORES%\%IG%.truststore.jks  
echo %STORES%\%IG%.keystore.jks
```

**5f** Set the value of the following properties in the Identity Governance server by using Identity Governance Configuration Utility. For more information about using the configuration utility, see the [Identity Governance Configuration Utility](#).

| Property  | Value                                |
|---|--------------------------------------|
| com.netiq.iac.kafka.publisher.truststore.location | Absolute path of the truststore file |
| com.netiq.iac.kafka.publisher.truststore.password | Truststore password                  |
| com.netiq.iac.kafka.publisher.keystore.location   | Absolute path of the KeyStore file   |
| com.netiq.iac.kafka.publisher.keystore.password   | Keystore password                    |
| com.netiq.iac.kafka.publisher.key.password        | Key password                         |

**5g** Delete the following files:

**CAUTION:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

**Linux:**

```
rm ${STORES}/${CA_CERT}  
rm ${STORES}/${IG}-cert-signed  
rm ${STORES}/${IG}-cert-req
```

**Windows:**

```
del %STORES%\%CA_CERT%  
del %STORES%\%IG%-cert-signed  
del %STORES%\%IG%-cert-req
```

**6** On the Transformation Hub server:

**CAUTION:** The following files should be deleted to prevent the distribution of security certificates that could be used to authenticate against the Transformation Hub. These files are very sensitive and should not be distributed to other machines.

```
rm ${CA_CERT}
rm ${CA_KEY}
rm ${IG_CERT_CA_TMP}/${IG}-cert-signed
rm ${IG_CERT_CA_TMP}/${IG}-cert-req
```

## Creating Fact Configuration Files

Identity Intelligence provides a script that creates the necessary files to enable data collection from Identity Governance. When you run the script for the first time, you must specify all the necessary details of Transformation Hub, such as host, port, protocol, and database schema (only for MSSQL and Oracle).

You must also specify the interval (in hours) at which Identity Governance collects and publishes data from its data source. The interval represents the elapsed time between when changes to the identity or resource occurs in the data source and when the data is collected and published in Identity Governance. If Identity Governance is configured to collect data from its data source at scheduled intervals, there will be a time lag before you see the changes in Identity Intelligence. As data must be first collected by Identity Governance and then it must be sent to Identity Intelligence through facts.

These details are stored in the `ig-facts-configuration-tool/fact.conf` file and will be used for subsequent execution of the script.

To modify any of the configuration details, you must edit the information in the `fact.conf` file and run the script again.

- 1 Log in to the Identity Intelligence server either as the `root` or a non-root user.
- 2 Change to the directory where you downloaded the `ig-facts-configuration-tool.tar` utility file.

For information about downloading the utility file, see [Downloading Identity Intelligence](#).

- 3 Execute the `tar -xvf ig-facts-configuration-tool.tar` command to unzip the file.
- 4 Change to `ig-facts-configuration-tool` directory:

```
cd ig-facts-configuration-tool
```

- 5 Run the script:

```
python ig_facts.py
```

The script creates the following files in the `ig-facts-configuration-tool/output` directory:

- ♦ `entity_reconciliation_id_attribute_payload.json`
- ♦ `account_domain_attribute_payload.json`
- ♦ `facts_creation.json`

- 6 Copy these files to the Identity Governance server.
- 7 Continue with [Mapping Attributes for Data Reconciliation](#).

## Mapping Attributes for Data Reconciliation

Identity Intelligence does data reconciliation based on certain [unique fields](#) for each entity type. When you run the fact configuration utility, it creates the `entity_reconciliation_id_attribute_payload.json` file to enable data reconciliation and the



`account_domain_attribute_payload.json` files to get the account domain of the data source. Therefore, you must import these files into Identity Governance. These files create the following attributes and you must map the appropriate values for these attributes:

- ♦ **entity\_reconciliation\_id:** You can either use the `entity_reconciliation_id` field or use any of the attributes available in the Identity Intelligence schema for Identity entity reconciliation. For more information, see [“Reconciling Data for the Identity Entity” on page 87](#).

If you want to use `entity_reconciliation_id`, map this field with the appropriate attribute in Identity Governance.

- ♦ **Account Domain:** Map the domain name of the data source by extracting it from the distinguished name.

To extract domain name from the distinguished name, Identity Intelligence provides `domainExtraction.js` transformation script, which is available in the configuration utility.

#### To map the newly added attributes:

- 1 Log in to Identity Governance as a Global Administrator.
- 2 Import the identity attribute:
  - 2a Click **Data Administration > Identity Attributes > Import Attributes**.
  - 2b Browse to select the `entity_reconciliation_id_attribute_payload.json` file that you copied from Identity Intelligence.
  - 2c Click **Import**.
- 3 (Conditional) If you want to use `entity_reconciliation_id` for identity reconciliation, map the `entity_reconciliation_id` attribute on every Identity source as follows:
  - 3a Click **Data Sources > Identities**.
  - 3b Select the appropriate Identity source.
  - 3c Click **Identity source name > Collect Identity**.
  - 3d Specify the appropriate attribute that must be used for identity data reconciliation in `entity_reconciliation_id`.
- 4 Import the account attribute:
  - 4a Click **Data Administration > Account Attributes > Import Attributes**.
  - 4b Browse to select the `account_domain_attribute_payload.json` file that you copied from Identity Intelligence.
  - 4c Click **Import**.
- 5 Map the `Account domain` attribute on every Account source as follows:
  - 5a Click **Data Sources > Applications**.
  - 5b Select the appropriate Account source.
  - 5c Click **Account source name > Collect Connected Account**.
  - 5d Click the script icon of **Account Domain**.
  - 5e Click **Or upload a script file** and uploaded the `domainExtraction.js` transformation script.  
For more information, see the “Creating Identity and Application Sources” section in the [Identity Governance Administrator Guide](#).
- 6 Continue with [Collecting Metrics from Identity Governance](#).

# Collecting Data from Identity Governance

To initiate data collection, you must collect metrics from Identity Governance:

- 1 Log in to Identity Governance as a Global Administrator.
- 2 Import the custom metrics:
  - 2a Click **Configuration** > **Analytics and Role Mining Settings** > **Metrics Collection** > **Import Custom Metrics**.
  - 2b Browse to select the `facts_creation.json` file that you copied from Identity Intelligence.
  - 2c Click **Import**.

You can see the list of newly created facts prefixed with an asterisk (\*).

- 3 Select the required facts and click **Actions** > **Collect metrics** > **Collect Now**.
- 4 Continue with [Reverting Backdated Events Configuration](#).

# 18 Reverting Backdated Events Configuration

Before data migration, you had tuned [events Kafka scheduler](#) to preserve the time stamp of the backdated events. Hence after data migration, you must revert events Kafka scheduler to the default configuration to avoid any performance issue.

Data migration may take a while depending on the number of entities and events. Hence, ensure that you perform the following after migration is complete.

## To revert event Kafka scheduler configuration:

- 1 Log in to Vertica node 1.

- 2 Change to the directory:

```
cd <vertica_installation_dir>/scripts
```

**Example:** `cd /opt/arcsight-vertica/scripts`

- 3 Execute the following command to revert to default configuration:

```
./tuning_util.sh -n
```



# 19 Verifying Data Collection Configuration

Once you have configured data collection from data sources, you can verify whether data collection configuration is successful by performing any of the following:

- “[Verifying SmartConnector Log Files](#)” on page 109
- “[Viewing Data in the Identity Intelligence User Interface](#)” on page 109

## Verifying SmartConnector Log Files

Once you configure the data source for data collection, the configured data sources start sending data to the SmartConnector and you can verify whether the SmartConnector is receiving data and picked the [categorization files](#) correctly by checking the `opt/smartconnectorforidentityintelligence1.1.1/current/logs/agent.log` file.

Following is a snippet of the `agent.log` file:

```
...[processSingleAlert] First event from [NetIQ|Identity Manager|<fqdn>] received.
...[getInputStream] Resource [netiq/identity_manager.link.csv] found in [/root/
ArcSightSmartConnectors/current/user/agent/acp/categorizer/current/netiq/
identity_manager.link.csv]
...[getInputStream] Resource [netiq/identity_manager.csv] found in [/root/
ArcSightSmartConnectors/current/user/agent/acp/categorizer/current/netiq/
identity_manager.csv]
...[processSingleAlert] Successfully loaded categorization file [netiq/
identity_manager.csv]
...[getInputStream] Resource [netiq/identity_manager.0.csv] found in [/root/
ArcSightSmartConnectors/current/user/agent/acp/categorizer/current/netiq/
identity_manager.0.csv]
...[processSingleAlert] Successfully loaded categorization file [netiq/
identity_manager.0.csv]
...[categorize] Successfully loaded categorization link file [netiq/
identity_manager.link.csv]
```

## Viewing Data in the Identity Intelligence User Interface

Depending on the data that you want to analyze, you can create a [View](#) and explore the **Profiles** of [users](#) and [access rights](#). For more information about viewing and analyzing data, see the [User's Guide to Identity Intelligence](#), which is also the context-sensitive help in the user interface.

**To launch Identity Intelligence:**

- 1 Specify the following URL:  
`https://<IDI-Server>/idi`
- 2 Log in with your credentials.



# IV Upgrading Identity Intelligence

This section provides information about upgrading a single-node deployment of Identity Intelligence.

- ◆ [Chapter 20, “Upgrade Checklist,” on page 113](#)
- ◆ [Chapter 21, “Upgrading Identity Intelligence,” on page 115](#)





# 20 Upgrade Checklist

Before you upgrade Identity Intelligence, review the following checklist to ensure a successful upgrade.

| Task  | See   |
|---|---|
| <input type="checkbox"/> 1. Review any updates to system requirements     | <a href="#">Identity Intelligence 1.1 System Requirements</a> |
| <input type="checkbox"/> 2. Complete the tasks mentioned in prerequisites | <a href="#">Prerequisites</a>                                 |
| <input type="checkbox"/> 3. Upgrade Vertica                               | <a href="#">Upgrading Vertica</a>                             |
| <input type="checkbox"/> 4. Upgrade CDF                                   | <a href="#">Upgrading CDF</a>                                 |
| <input type="checkbox"/> 5. Upgrade Identity Intelligence                 | <a href="#">Upgrading Identity Intelligence</a>               |
| <input type="checkbox"/> 6. Complete the post-upgrade configuration tasks | <a href="#">Post-Upgrade Configurations</a>                   |



# 21 Upgrading Identity Intelligence

This section provides information about upgrading Identity Intelligence.

- ◆ “Prerequisites” on page 115
- ◆ “Upgrading Vertica” on page 115
- ◆ “Upgrading CDF” on page 117
- ◆ “Upgrading SmartConnector” on page 122
- ◆ “Upgrading Identity Intelligence” on page 122
- ◆ “Post-Upgrade Configurations” on page 124

## Prerequisites

Before upgrading Identity Intelligence, complete the following tasks:

- ◆ Download the Identity Intelligence installer files, untar the files and verify the signatures. For more information, see [Downloading Identity Intelligence](#).

You need the following images and files in Identity Intelligence installer for upgrading:

- ◆ `arcsight-vertica-installer_x.x.x.tar.gz`
- ◆ `arcsight-installer-metadata-x.x.x.tar.gz`
- ◆ `idi-x.x.x`
- ◆ `analytics-x.x.x`
- ◆ `transformationhub-x.x.x.x`
- ◆ `cdf-2020.02.xxxx`
- ◆ `cdf-upgrade-2019.08.xxxx`
- ◆ `SmartConnector for Identity Intelligence x.x.x.x.zip`
- ◆ Download the following file in a computer from which you access the CDF Management Portal:  
`arcsight-installer-metadata-x.x.x.tar.gz`
- ◆ **Save any custom configuration information:**

If you have set any custom configurations in the `tg_reconciliation_fields.json` file, save the file in a separate location so that the upgrade does not override the customizations.

This file is located at:

```
cd /<install_directory>/<arcsight_nfs_volume>/eventbroker/di/data-processor/  
conf
```

## Upgrading Vertica

This section provides information about upgrading Vertica.

## To upgrade Vertica:

- 1 The upgrade process is irreversible. Therefore, ensure that you back up the database. For more information, see [Creating Full Backups](#).
- 2 Launch a terminal session, then log in to the master node as `root` or as a `sudo` user.
- 3 Navigate to `/opt/` and stop the kubernetes services:

```
kube-stop.sh
```

- 4 Change to the directory where you have untarred Vertica Installer:

```
cd <download_directory>/identityintelligence-x.x.x.x/installers/arcsight-vertica-installer_x.x.x.x
```

- 5 Run the upgrade commands in the following order:

---

**NOTE:** You cannot rerun the commands.

---

1. `./investigate_upgrade -c upgrade-investigate`
2. `./investigate_upgrade -c update-configuration`
3. `./investigate_upgrade -c upgrade-vertica-rpm`

- 6 Start the Vertica database:

```
/opt/arcsight-vertica/vertica_installer start-db
```

- 7 Update the Kafka scheduler:

```
/opt/arcsight-vertica/kafka_scheduler update
```

- 8 Start the Kafka scheduler:

```
/opt/arcsight-vertica/kafka_scheduler start
```

- 9 Stop the Vertica agent:

```
systemctl stop vertica_agent
```

- 10 Disable the Vertica agent:

```
systemctl disable vertica_agent
```

- 11 Navigate to `/opt/` and start the kubernetes services:

```
kube-start.sh
```

- 12 Check the version of Vertica to validate successful upgrade:

- 12a Navigate to `<vertica_installer_directory>/bin`.

- 12b Switch to `dbadmin` user:

```
su dbadmin
```

- 12c Log in to `vsq1` and specify the password when prompted:

```
./vsq1
```

- 12d To view the version:

```
SELECT version();
```

# Upgrading CDF

Upgrading CDF has 2 phases: upgrading from CDF 2019.05 to CDF 2019.08 and upgrading from CDF 2019.08 to 2020.02.

You can upgrade CDF in the following ways:

- ♦ [“Manual Upgrade” on page 117](#)
- ♦ [“Automated Upgrade” on page 120](#)

## Manual Upgrade

This section provides information about upgrading CDF.

- ♦ [“Prerequisites” on page 117](#)
- ♦ [“Upgrading CDF from CDF 2019.05 to CDF 2019.08” on page 117](#)
- ♦ [“Upgrading CDF from CDF 2019.08 to CDF 2020.02” on page 118](#)
- ♦ [“Troubleshooting” on page 119](#)

## Prerequisites

- ♦ Ensure that you have downloaded the [Identity Intelligence package](#) on all the CDF nodes. You need the following files in Identity Intelligence package for upgrading CDF:
  - ♦ `cdf-2020.02.xxxxx`
  - ♦ `cdf-upgrade-2019.08.x.x.x.x`
- ♦ Ensure that you have minimum 50 GB free space in master node and 30 GB free space in worker node.
- ♦ Create a backup directory with minimum 30 GB of space on every node of your cluster:

```
mkdir /tmp/upgrade-backup
```

If you do not create a backup directory, the backup files will be stores in the default location (`\tmp`).
- ♦ Install `socat` and `container-selinux` packages on all nodes in the cluster by using the command:

```
yum install <package_name>
```
- ♦ Ensure that you have appropriate permission to restart nodes. You may need to restart nodes if there is an issue during upgrade.
- ♦ Ensure that all nodes are currently running:

```
kubectl get nodes
```
- ♦ Ensure that all pods are currently running:

```
<K8S_HOME>/bin/kube-status.sh
```

## Upgrading CDF from CDF 2019.05 to CDF 2019.08

1 Run the following commands on each node:

```
cd <download_directory>/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-2019.08.x.x.x.x
./upgrade.sh -t <path_to_backup_directory> -i
```

Example:

```
cd /opt/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-2019.08.x.x.x.x
./upgrade.sh -t /tmp/upgrade-backup -i
```

---

**NOTE:** If you do not specify `-t <path>`, the backup files will be stored in the default location (`\tmp`).

---

**2** Run the following commands on one of the master nodes:

```
cd <download_directory>/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-
2019.08.x.x.x.x
./upgrade.sh -u
```

Example:

```
cd /opt/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-2019.08.x.x.x.x
./upgrade.sh -u
```

**3** (Optional) Clean unused docker daemon images by executing the following command on all the worker and master nodes:

```
cd <download_directory>/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-
2019.08.x.x.x.x
./upgrade.sh -c
```

Example:

```
cd /opt/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-2019.08.x.x.x.x
./upgrade.sh -c
```

**4** Ensure that upgrade is successful by verifying the following on all the nodes:

- ◆ Check the CDF version by executing the command:  
`cat <K8S_HOME>/version.txt`
- ◆ Check the current status of CDF pods by executing the command:  
`<K8S_HOME>/bin/kube-status.sh`

---

**NOTE:** If the pods are not in running state, execute the following command to recreate main cluster services:

```
<K8s_HOME>/bin/kube-restart.sh
```

---

## Upgrading CDF from CDF 2019.08 to CDF 2020.02

**1** Run the following commands on each node:

```
cd <download_directory>/identityintelligence-x.x.x.x/installers/cdf-
2020.02.x.x.x.x
./upgrade.sh -t <path_to_backup_directory> -i
```

Example:

```
cd /opt/identityintelligence-x.x.x.x/installers/cdf-2020.02.x.x.x.x
./upgrade.sh -t /tmp/upgrade-backup -i
```

---

**NOTE:** If you do not specify `-t <path>`, the backup files will be stored in the default location (`\tmp`).

---

**2** Run the following commands on one of the master nodes:

```
cd <download_directory>/identityintelligence-x.x.x.x/installers/cdf-2020.02.x.x.x.x
./upgrade.sh -u
```

Example:

```
cd /opt/identityintelligence-x.x.x.x/installers/cdf-2020.02.x.x.x.x
./upgrade.sh -u
```

**3** (Optional) Clean unused docker daemon images by executing the following command on all the worker and master nodes:

```
cd <download_directory>/identityintelligence-x.x.x.x/installers/cdf-2020.02.x.x.x.x
./upgrade.sh -c
```

Example:

```
cd /opt/identityintelligence-x.x.x.x/installers/cdf-2020.02.x.x.x.x
./upgrade.sh -c
```

**4** Ensure that upgrade is successful by verifying the following on all the nodes:

- ◆ Check the CDF version by executing the command:

```
cat <K8S_HOME>/version.txt
```

- ◆ Check the current status of CDF pods by executing the command:

```
<K8S_HOME>/bin/kube-status.sh
```

---

**NOTE:** If the pods are not in running state, execute the following command to recreate main cluster services:

```
<K8s_HOME>/bin/kube-restart.sh
```

---

## Troubleshooting

This section provides workaround for the following problems during CDF upgrade:

- ◆ If any of the upgrade process fails:

1. Ensure that `kubelet` is running by executing the command:

```
kubectl get pod -all-namespaces
```

2. Rerun the upgrade command.

- ◆ If upgrade process timeouts, perform the following:

1. Restart the node. For steps to restart the node, see [Restarting Nodes in the Cluster](#).

2. Ensure that all the pods are in running state:

```
<K8S_HOME>/bin/kube-status.sh
```

3. Rerun the upgrade command.

## Automated Upgrade

The CDF automated upgrade is run with a single command and requires no interaction until completion of each phase. Typically, the upgrade process takes around 1 hour for a 3x3 cluster.

Automatic upgrade allows you to upgrade CDF from any host (known as the upgrade manager). The upgrade manager can be one of the following:

- ♦ One of the cluster nodes
- ♦ A host outside the cluster in a secure network location

There are 4 directories involved in the auto-upgrade process:

- ♦ An auto-upgrade directory `/tmp/autoUpgrade` will be auto generated on the upgrade manager. It will store the upgrade process steps and logs.
- ♦ A backup directory `/tmp/CDF_201905_upgrade` will be auto generated on every node. (approximate size 1.5 GB)
- ♦ A backup directory `/tmp/CDF_201908_upgrade` will be auto generated on every node. (approximate size 1.7 GB)
- ♦ A working directory will be auto generated on the upgrade manager and every node at the location provided by the `-d` parameter. The upgrade package will be copied to this directory. (approximate size 9 GB). The directory will be automatically deleted after the upgrade.

---

**NOTE:** The working directory can be created manually on upgrade manager and every node and then passed as `-d` parameter to the auto-upgrade script. If you are a non-root user on the nodes inside the cluster, make sure you have permission to this directory.

---

## Prerequisite

- ♦ Ensure that you have downloaded the [Identity Intelligence installer package](#) to the download directory (`<download_directory>`) on the upgrade manager. Verify that you have CDF 2019.08 and CDF 2020.02 upgrade packages in the following location:
  - ♦ `{download-directory}/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-2019.08.00134-2.2.0.2/autoUpgrade.sh`
  - ♦ `{download-directory}/identityintelligence-x.x.x.x/installers/cdf-2020.02.00120-2.2.0.2/autoUpgrade.sh`
- ♦ Install `socat` and `container-selinux` packages on all nodes in the cluster by using the command:

```
yum install <package_name>
```
- ♦ Configure passwordless SSH communication between the upgrade manager and all the nodes in the cluster as follows:
  1. Run the following command on the upgrade manager to generate key pair:

```
ssh-keygen -t rsa
```
  2. Run the following command on the upgrade manager to copy the generated public key to every node of your cluster:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@<node_fqdn_or_ip>
```

## Upgrading from CDF 2019.05 to CDF 2019.08

- 1 Change to the directory where you have downloaded the CDF 2019.08 package:



```
cd /{download-directory}/identityintelligence-x.x.x.x/upgrade/cdf-upgrade-2019.08.00134-2.2.0.2/
```

**2** Run the following command for automatic upgrade:

```
./autoUpgrade.sh -d /path/to/workinig_directory -n  
{any_cluster_node_adress_or_ip}
```

**Example:**

```
./autoUpgrade.sh -d /tmp/upgrade -n pueas-ansi-nodel.swinfra.net
```

## Upgrading from CDF 2019.08 to CDF 2020.02

**1** Delete 2019.08 cdf upgrade directory (cdf-upgrade-2019.08.xxxx):

```
rm -rf {download-directory}/cdf-upgrade-2019.08.xxxx
```

**2** Run a kubectl patch command to configure IDM pod affinity and wait until IDM pods are up and running:

```
kubectl patch deployment idm -n core --patch '{ "spec": { "template": { "spec": {  
  "affinity": { "podAffinity": {  
    "preferredDuringSchedulingIgnoredDuringExecution": [ { "labelSelector": {  
      "matchExpressions": [ { "key": "app", "operator": "In", "values": [ "idm-app" ]  
    } ] }, "topologyKey": "kubernetes.io/hostname" } ] } } } } } } } }'
```

**3** Change to the directory where you have downloaded the CDF 2020.02 package:

```
cd /{download-directory}/identityintelligence-x.x.x.x/installers/cdf-2020.02.00120-2.2.0.2/
```

**4** Run the following command for automatic upgrade:

```
./autoUpgrade.sh -d /path/to/workinig_directory -n  
{any_cluster_node_adress_or_ip}
```

**Example:**

```
./autoUpgrade.sh -d /tmp/upgrade -n pueas-ansi-nodel.swinfra.net
```

**5** Delete the auto-upgrade temporary directory from the upgrade manager.

The auto-upgrade temporary directory contains the upgrade steps and logs. If you want to upgrade another cluster from the same upgrade manager, remove that directory.

```
rm -rf /tmp/autoUpgrade
```

## Troubleshooting

- ◆ If the automatic upgrade fails, run `autoUpgrade.sh` again as outlined above. The process may take several attempts to succeed.
- ◆ In some cases, the automatic upgrade may return an error message about the upgrade process still running and the existence of a `*.lock` file which prevents `autoUpgrade.sh` to continue. This file is automatically deleted in a few minutes. Alternatively, you can manually delete this file. Once the file is deleted either automatically or manually, run `autoUpgrade.sh` again.
- ◆ If the automated upgrade process is still unsuccessful, continue the process on the failed node using the procedure outlined in [Manual Upgrade](#).

# Upgrading SmartConnector

Identity Intelligence contains two different instances of SmartConnector in different directories: one for audit events collection using SSL and one for entity change events collection using non-SSL. Therefore, you must upgrade both the instances of SmartConnector.

## To upgrade SmartConnector:

- 1 Log in to the Identity Intelligence node as the `root` user.
- 2 Stop the SmartConnector service:  

```
/etc/init.d/arc_<name of the SmartConnector> stop
```
- 3 Change to the directory (for example, `/opt`) where you [downloaded and unzipped](#) the SmartConnector installation file.
- 4 Change to the following directory:  

```
cd smartconnectorforidentityintelligence<version>
```
- 5 Update permissions for the `<SmartConnector_installation>` file:  
Example:  

```
chmod 755 ArcSight-<version>-Connector-Linux64.bin
```
- 6 Run the SmartConnector installer to upgrade:  
Example:  

```
./ArcSight-<version>-Connector-Linux64.bin
```
- 7 Enter the directory where the SmartConnector is installed (for example, `/opt/SmartConnector`).  

---

**NOTE:** Based on the SmartConnector instance you are upgrading, provide the appropriate location.


---
- 8 Execute the following command to complete the upgrade:  

```
./runagentsetup.sh
```
- 9 Verify SmartConnector upgrade by validating the SmartConnector Version:  

```
<SmartConnector Installation Directory>/current/bin/arcsight agents -v
```
- 10 Start the SmartConnector service:  

```
/etc/init.d/arc_<name of the SmartConnector> start
```

# Upgrading Identity Intelligence

- 1 Log in to the CDF Management Portal.
- 2 To accept the Configuration Page certificate:
  - 2a On the installed cluster, ensure that you access configuration properties at least once to accept the certificate. This step is important to avoid any certificate error during the upgrade.
  - 2b Go to **Deployment > Deployments >  > Reconfigure**.
  - 2c Accept the certificate.
- 3 Click **Deployment > Metadata**.
- 4 Click **+ADD** on the top-right and add the installer metadata file.

**5** Click **Deployment > Deployments**.

Under **Update**, you will see a notification that indicates there are updates available to components in your cluster.

**6** Click the notification icon and then click the installer metadata link.

**7** Since you have already [downloaded the required files for upgrade](#), continue to click **Next** until you are in the **Import suite images** screen.

**8** Launch a terminal session, then log in to the master node as `root` or as a `sudo` user.

**9** Change to the following directory:

```
cd /<cdf_installer_directory>/kubernetes/scripts/
```

For example:

```
cd /opt/arcsight/kubernetes/scripts
```

**10** Upload the Analytics image:

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x.x/suite_images/analytics-x.x.x.x -u registry-admin -p <cdf_password>
```

**11** Upload the Identity Intelligence image to the local registry:

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x.x/suite_images/idi-x.x.x.x -u registry-admin -p <cdf_password>
```

**12** Upload the Transformation Hub image to the local registry:

```
./uploadimages.sh -d <download_directory>/identityintelligence-x.x.x.x/suite_images/transformationhub-x.x.x.x -u registry-admin -p <cdf_password>
```

**13** Switch to the CDF Management portal, then click **CHECK AGAIN** to ensure that the images have been uploaded.

**14** Click **Next** until you get the **Deployment/Restart** screen.

**15** Click **Deployment > Metadata** and delete the metadata file that is not in use.

**16** Check the upgrade status by monitoring the pods status:

**16a** Go to CDF Management Portal.

**16b** Click **Cluster > Dashboard**.

**16c** In the left pane, switch to the `arcsight-installer-<XXXX>` **Namespace**.

**16d** Go to the **Pods** section and continue reloading the page until you see the status for all the pods as *Running*. On the **Status** column, sort the pod status to see if any pod is not running. Once the status for all the pods is changed to *Running*, Identity Intelligence upgrade is complete.

**17** To determine whether the upgrade is successful:

**17a** Go to **Deployment > Deployments**.

**17b** Under the **Version** column, you will see the new version of the suite.

Also, under the **Update** column, you will see zero, which indicates there are no updates available for components in your cluster.

**18** Identity Intelligence supports only SSL communication between Identity Intelligence and its related components. Therefore, if SSL is not configured in your environment, configure SSL by performing the steps in the [Disabling Plain Text Communication](#) section.

**19** Continue with [“Post-Upgrade Configurations”](#) on page 124.

# Post-Upgrade Configurations

If you had set any custom configurations in the `tg_reconciliation_fields.json` file, you must restore the custom configurations after the upgrade.

## To restore the custom configurations:

- 1 Change to the following directory:

```
cd /<install_directory>/<arcsight_nfs_volume>/eventbroker/di/data-processor/  
conf
```

- 2 Copy any custom configuration parameter values from the `tg_reconciliation_fields.json` file to the `tg_reconciliation_fields.json_<timestamp>` file and delete `tg_reconciliation_fields.json` file.
- 3 Rename the `tg_reconciliation_fields.json_<timestamp>` file as `tg_reconciliation_fields.json`.

# V Managing Identity Intelligence

This section provides information about managing Identity Intelligence.

- ♦ [Chapter 22, “Installing Licenses,” on page 127](#)
- ♦ [Chapter 23, “Assigning Permissions and Roles,” on page 129](#)
- ♦ [Chapter 24, “Adding Vertica Nodes,” on page 131](#)
- ♦ [Chapter 25, “Restarting Nodes in the Cluster,” on page 133](#)
- ♦ [Chapter 26, “Configuring the Log Level,” on page 135](#)
- ♦ [Chapter 27, “Collecting Diagnostic Logs,” on page 137](#)
- ♦ [Chapter 28, “Modifying Transformation Hub Configurations,” on page 139](#)
- ♦ [Chapter 29, “Resetting the CDF Administrator Password,” on page 143](#)
- ♦ [Chapter 30, “Renewing CDF Certificates,” on page 145](#)
- ♦ [Chapter 31, “Changing the CDF Certificate Authority,” on page 147](#)
- ♦ [Chapter 32, “Retrieving CDF Root CA,” on page 151](#)
- ♦ [Chapter 33, “Managing Vertica,” on page 153](#)




# 22 Installing Licenses

To access Identity Intelligence and its features post the trial period, you must install valid license files. Download licenses for the following from [Micro Focus Customer Center](#):

- ♦ Identity Intelligence
- ♦ Transformation Hub
- ♦ Vertica
- ♦ ArcMC (only if you plan to use ArcMC)

## Installing the License for Identity Intelligence and Transformation Hub

- 1 Log in to the CDF Management Portal.
- 2 Navigate **Administration** > **License** .
- 3 From the drop-down menu, select **License**.
- 4 To install the Transformation Hub license files, complete the following steps:
  - 4a Click **Install Licenses** > **Choose File**, browse to the location of your valid license file, and click **Next**.
  - 4b Click **Install Licenses** and follow the prompts to apply the license.
- 5 To install the license for Identity Intelligence repeat [Step 4](#).
- 6 Log in to the master node as `root`.
- 7 After applying your license file, restart each Kafka broker in the cluster, *one at a time*, as follows:
  - 7a Run the following command to restart the selected broker node:

```
kubectl delete pod th-kafka-(x) -n arcsight-installer-XXX
```
  - 7b Watch the logs and ensure that the Kafka broker node is up and running:

```
kubectl logs th-kafka-(x) -n arcsight-installer-XXX
```

Once the selected broker node is up and running, only then proceed to restart the next node.
- 8 (Conditional) The following step is applicable only for Identity Intelligence 1.1 and not for Identity Intelligence 1.1.1 or later:
  - 8a Run the following command to restart the selected Users and Entities pod:

```
kubectl delete pod ueba-web-app-(x)-(x) -n arcsight-installer-XXX
```
  - 8b Watch the logs and ensure that the Users and Entities pod is up and running:

```
kubectl logs ueba-web-app-(x)-(x) -n arcsight-installer-XXX
```
- 9 (Conditional) The following step is applicable only for Identity Intelligence 1.1 and not for Identity Intelligence 1.1.1 or later:
  - 9a Run the following command to restart the selected Identity Intelligence pod:

```
kubectl delete pod idi-web-app-(x)-(x) -n arcsight-installer-XXX
```

**9b** Watch the logs and ensure that the Identity Intelligence pod is up and running:

```
kubectl logs idi-web-app-(x)-(x) -n arcsight-installer-XXX
```

**10** Under **View Licenses**, verify whether the licenses have been applied.

Identity Intelligence licenses come up with a default validity period. To prevent any interruption in the functionality, ensure that you [renew the license](#) before its validity expires.

## Installing the License for Vertica

By default, Vertica is installed with the community edition. If you want to upgrade to enterprise edition, perform the following:

- 1 Log in as `root` to the server where Vertica is installed.
- 2 Copy the Vertica license file to a temporary directory. For example, `/tmp`.
- 3 Switch to database administrator user created during Vertica installation. For more information, see [“Installing Vertica” on page 29](#).

**4** Run `adminTools`:

```
/opt/vertica/bin/adminTools
```

- 5** Select **Advanced Menu**.
- 6** Select **Upgrade License Key**.
- 7** From the **Select database to Upgrade License Key**, select **Investigate > OK**.
- 8** Specify the license file path name in the following format and select **OK**:

```
/tmp/<license file name>
```

- 9** Select **Accept > OK**.
- 10** Enter password for the database and select **OK**.
- 11** Read through the license summary and select **OK**.
- 12** Select **Exit > OK**.

## Installing the License for ArcMC

- 1 Download the license file to the computer from which you can connect to ArcMC web interface.
- 2 Log in to the ArcMC web interface.
- 3 Click **Administration > Setup > System Admin**.
- 4 Click **License & Update** in the **System** section.
- 5 Click **Browse** to locate the file.
- 6 Click **Upload Update**.

Once the update has completed, the Update Results page displays the update result (success/failure) and whether the update requires a reboot. If the update requires a reboot, ArcMC reboots automatically.



# 23

## Assigning Permissions and Roles

Permissions help you control user access to Identity Intelligence and its features. By default, only the **System Administrator** and **Identity Intelligence User** roles have the Identity Intelligence permission. You can create new roles or add the necessary permission to existing roles to allow users to use Identity Intelligence.

The following table lists the permissions required to use Identity Intelligence capabilities:

| Permission                   | Allows users to...  |
|------------------------------|---|
| Access Identity Intelligence | Manage Views and also explore all access rights.                              |
| Access Users and Entities    | Explore all user details about the user profiles listed in the Summary table. |

- ♦ [“Creating and Assigning Permissions to a Role” on page 129](#)
- ♦ [“Creating a User” on page 129](#)

### Creating and Assigning Permissions to a Role

You can group multiple permissions into a role and assign the relevant role to Identity Intelligence users. Assigning a role to users helps in consistent application of permissions such that all users in the role have the same set of permissions.

- 1 Log in to Identity Intelligence.
- 2 Click **ADMIN > Roles > Create Role**.
- 3 Specify a role name and press Enter.
- 4 Select the **Access Identity Intelligence** permission. Users will also get the Access Users and Entities permission by default.

### Creating a User

- 1 Log in to Identity Intelligence.
- 2 Click **ADMIN > Account Groups > Create User**.
- 3 Specify the email ID and name of the user.
- 4 Select the groups to which you want to add the user.
- 5 Assign the role that includes Identity Intelligence permissions.
- 6 Click **Save**.
- 7 Under **Account Groups**, click **All Users**.
- 8 Select the user you just created.
- 9 Click **RESET PASSWORD**.
- 10 Set the password and click **SAVE**.



# 24 Adding Vertica Nodes

For high availability and to manage the workload, you can add additional nodes after the installation.

If you had installed Vertica on a single node by specifying the loopback address, you cannot expand the cluster by adding nodes. To add nodes, you must uninstall and reinstall the Vertica node by specifying an IP address or hostname. For more information, see [Setting Vertica Properties](#).

For adding Vertica nodes, you must first add the hosts to the cluster and then add them as nodes to the Vertica database.

## To add nodes to the database:

1 Log in as `root` to the Vertica cluster node where the Vertica installer is available.

2 Stop the Kafka scheduler by using the command:

```
<vertica-installer-dir>/kafka_scheduler stop
```

3 Add hosts to the cluster by using the command:

```
/opt/vertica/sbin/update_vertica --add-hosts  
<new_host1>,<new_host2>,<new_host3> --rpm <vertica_rpm_pkg_location> --dba-  
user <user_name> --dba-group <group_name> --dba-user-password-disabled --data-  
dir=<data_dir_location>
```

For example:

```
/opt/vertica/sbin/update_vertica --add-hosts <new-host1>,<new-host2>,<new-  
host2> --rpm <vertica-installer-dir>/data/vertica-x.x.x.x.rpm --dba-user  
dbadmin --dba-group dbadmin --dba-user-password-disabled --data-dir=/opt/  
vertica/data
```

4 Change the logged in user to a database administrator:

```
su - dbadmin
```

5 Add nodes to the Vertica database by using the command:

```
admintools -t db_add_node -d investigate -p <db-admin_password> -s <new-  
host1>,<new-host2>,<new-host3>
```

6 Re-balance data in the database and set the K-safe value by using the command:

```
admintools -t rebalance_data -d investigate -k <k_safe_value> -p <db-  
admin_password>
```

Valid K-safe values for highly available databases are 1 and 2. Databases without high availability do not have to be K-safe and can be set to 0. A K-safe 1 database must have at least three nodes. For more information about the K-safe value, see [Designing for K-Safety](#).

7 Start Kafka scheduler by using the command:

```
<vertica-installer-dir>/kafka_scheduler start
```



# 25 Restarting Nodes in the Cluster

If you want to restart or shutdown any node in the Identity Intelligence cluster, you must stop the Kubernetes and Vertica services running on the node. If you do not stop the services running on the node, the Vertica database on the node may be corrupted and the Kubernetes pods will not start after the restart.

You can restart nodes in one of the following ways:

- ♦ [“Restarting Nodes by Using Scripts” on page 133](#)
- ♦ [“Restarting Nodes Manually” on page 133](#)

## Restarting Nodes by Using Scripts

*Applicable only if you have installed Identity Intelligence by using scripts.*

**To restart the node, perform the following:**

- 1 Log in to the node you need to restart.
- 2 To restart the node:
  - ♦ For Identity Intelligence 1.1, execute the command:  
`/opt/idi/bin/idi-single-node-util.sh reboot_node`
  - ♦ For Identity Intelligence 1.1.1 and later, execute the command:  
`/opt/identityintelligence/bin/single-node-util.sh reboot_node`
- 3 After the node restarts, you must start the Vertica services:
  - ♦ For Identity Intelligence 1.1, execute the command:  
`/opt/idi/bin/idi-single-node-util.sh start`
  - ♦ For Identity Intelligence 1.1.1 and later, execute the command:  
`/opt/identityintelligence/bin/single-node-util.sh start`

## Restarting Nodes Manually

*Applicable only if you have installed Identity Intelligence manually.*

**To restart the node manually, perform the following:**

- 1 (Conditional) If the node contains CDF, perform the following:
  - 1a Log in to the node you need to restart as `root`.
  - 1b Change to directory:  
`cd <K8S_HOME>/bin/`  
For example:  
`/opt/arcsight/kubernetes/bin`
  - 1c Stop the kubernetes services by using the command:

```
kube-stop.sh
```

- 1d** Unmount Kubernetes volumes by using the command:

```
kubelet-umount-action.sh
```

- 2** (Conditional) If the node contains Vertica, perform the following:

- 2a** Log in to the node as a database administrator.

- 2b** Stop Vertica services by using the command:

```
/opt/vertica/bin/admintools -t stop_db -p <Vertica_password> -d investigate  
--force
```

- 3** Restart the node using the command:

```
reboot
```

- 4** (Conditional) After the node restarts, perform the following if the node contains Vertica:

- 4a** Log in to the node as a database administrator.

- 4b** Start Vertica services by using the command:

```
/opt/vertica/bin/admintools -t start_db -p <Vertica_password> -d  
investigate --force
```

- 5** (Conditional) After the node restarts, perform the following if the node contains CDF:

- 5a** Log in to the node as `root`.

- 5b** Change to directory:

```
cd <K8S_HOME>/bin/
```

For example:

```
/opt/arcsight/kubernetes/bin
```

- 5c** Check whether all Kubernetes services are running:

```
kube-status.sh
```


- 5d** (Conditional) If any of the service is not running, start the service by using the command:

```
kube-start.sh
```

# 26 Configuring the Log Level

The log levels for Identity Intelligence and Users and Entities are set to **Info** by default. You can configure the log level as desired for troubleshooting purposes.

## To configure the log level:

- 1 Log in to the CDF Management Portal.
- 2 Click  of `arcsight-installer` and click **Reconfigure**.
- 3 Click **Identity Intelligence**.
- 4 Select the appropriate log level for Identity Intelligence and Users and Entities.
- 5 Click **Save**.





# 27 Collecting Diagnostic Logs

Diagnostic log files help in investigating and troubleshooting issues. This section provides information about collecting diagnostic logs for Identity Intelligence.

## To collect diagnostic logs:

1 Log in to the master node as `root`.

2 Obtain the docker ID of web service:

```
docker ps | grep atlas_web-service
```

3 Obtain the diagnostic file from the web service pod:

```
docker cp <docker_id of atlas_web-service>:/th/ws/th_diag/th_diag.tgz  
<directory where you want to copy the .tgz file>
```

4 Change to the directory where you copied the `.tgz` file:

```
cd <directory where you copied the .tgz file>
```

5 Untar the diagnostic file:

```
tar -xzf th_diag.tgz
```

6 Provide the execute permission for the extracted files:

```
chmod 0700 *.sh
```

7 Obtain the diagnostic logs:

```
./th-diag.sh
```

All the log files are available in the following location:

```
thdiag-<FQDN>-<yyyy-mm-dd-hh-mm-ss>/<directory where you copied the .tgz file>
```



# 28 Modifying Transformation Hub Configurations

This section describes how to modify some of the Transformation Hub configurations that are available only during installation. To modify these configurations after installation, you need to uninstall and reinstall Transformation Hub with the modifications. Therefore, plan for a downtime before performing the following:

- ♦ [“Disabling Plain Text Communication” on page 139](#)
- ♦ [“Enabling Client Authentication” on page 140](#)

## Disabling Plain Text Communication

By default, both SSL and plain text communications are enabled in Transformation Hub. If you want only SSL to be enabled in Transformation Hub, you can disable plain text communication.

- ♦ [“Prerequisite” on page 139](#)
- ♦ [“Disabling Plain Text Communication” on page 139](#)

### Prerequisite

When you are disabling plain text communication, ensure that you are configuring SSL between Transformation Hub (Kafka) and all the following components outside the Kubernetes cluster:



| Component                                     | See  |
|---|--|
| Vertica                                       | <a href="#">Configuring SSL for Vertica</a>  |
| Identity Governance                           | <a href="#">Configuring SSL between Identity Governance and Transformation Hub</a> |
| Identity Manager Driver for Entity Data Model | <a href="#">Creating and Configuring the Driver.</a>                               |
| SmartConnector                                | <a href="#">Installing SmartConnector</a>  |

## Disabling Plain Text Communication

Disabling plain text communication involves uninstalling and reinstalling Transformation Hub. Uninstalling Transformation Hub removes all Transformation Hub configurations, however data in Kafka topics and Kafka topic offsets are retained.

**To disable plain text communication:**

- 1 Log in to CDF Management Portal.
- 2 Click **Deployment** > **Deployments**.



- 3 Uninstall Transformation Hub:
  - 3a Click  of `arcsight-installer`, then click **Change**.
  - 3b In the **Capabilities** page, deselect **Transformation Hub**.
  - 3c Click **Next** until you reach the **Configuration Complete** page.
  - 3d Click **Next** after all the pods in the **Configuration Complete** page are displayed in green.
- 4 Reinstall Transformation Hub and update the configuration to allow only SSL:
  - 4a Click  of `arcsight-installer`, then click **Change**.
  - 4b In the **Capabilities** page, select **Transformation Hub**.
  - 4c Click **Next** until you reach the **Transformation Hub** configuration page.
  - 4d In the **Transformation Hub** configuration page:
    - ♦ All the values in this page are reset to default during reinstallation. Therefore, you must set the appropriate values for all the configuration fields.  
For information about the value, see the “Transformation Hub Tuning” section in the [Hardware Requirements and Tuning Guidelines](#).
    - ♦ Disable **Allow plain text (non-TLS) connections to Kafka**.
  - 4e Click **Next** until you reach the **Configuration Complete** page.
  - 4f Click **Next** after all the pods in the **Configuration Complete** page are displayed in green.
- 5 Restart the ITOM-DI pods manually:
  - 5a Get the name of all ITOM-DI pods:
 

```
kubectl get pods --all-namespaces | grep itom-d
```
  - 5b Restart all the pods individually by executing the command:
 

```
kubectl delete pod -n <namespace> <ITOM-DI pod name>
```
- 6 Restart all the components mentioned in the [prerequisite](#) for the changes to take effect.

## Enabling Client Authentication

By default, client authentication is disabled in Transformation Hub. To enable client authentication after installation, perform the following:

- 1 Replace the default CA with a new CDF root CA. For more information, see [Changing the CDF Certificate Authority](#).
- 2 Log in to CDF Management Portal.
- 3 Click **Deployment > Deployments**.
- 4 Click  of `arcsight-installer`, then click **Uninstall** to uninstall all the software.
- 5 Click  of `arcsight-installer`, then click **Install** to reinstall Identity Intelligence and all the software.
- 6 Select the metadata file version in **version** and click **Next**.
- 7 Read the license agreement and select **I agree**.
- 8 Click **Next**.
- 9 In the **Capabilities** page, select the following and click **Next**:
  - ♦ Transformation Hub

- ◆ Identity Intelligence
  - ◆ Analytics. Prerequisite for ArcSight Investigate and Identity Intelligence.
- 10** Specify the values you provided during installation:
- 10a** In the **Transformation Hub** configuration page, enable **Enable Connection to Kafka uses TLS Client Authentication**.
- Ensure that you provide the appropriate values for other configuration fields. For more information, see the “Transformation Hub Tuning” section in the [Hardware Requirements and Tuning Guidelines](#).
- 10b** In the **Analytics** configuration page:
- ◆ Specify Vertica connection details
  - ◆ Specify values for **Client ID** and **Client Secret** for Single Sign-On
- For more information about the values, see [Installing Identity Intelligence](#).
- 11** Click **Next** until you reach the **Configuration Complete** page.
- 12** Restart the ITOM-DI pods manually after all the pods are displayed in green in the **Configuration Complete** page:
- 12a** Get the name of all ITOM-DI pods:
- ```
kubectl get pods --all-namespaces | grep itom-d
```
- 12b** Restart all the pods individually by executing the command:
- ```
kubectl delete pod -n <namespace> <ITOM-DI pod name>
```
- 13** Ensure to configure [mutual authentication SSL](#) in all the following components:

| Component                                     | See  |
|---|--|
| Vertica                                       | <a href="#">Configuring SSL for Vertica</a>  |
| Identity Governance                           | <a href="#">Configuring SSL between Identity Governance and Transformation Hub</a> |
| Identity Manager Driver for Entity Data Model | <a href="#">Creating and Configuring the Driver.</a>                               |
| SmartConnector                                | <a href="#">Installing SmartConnector</a>  |



# 29 Resetting the CDF Administrator Password

To change the administrator password of CDF, perform the following:

- 1 Log in to the CDF Management Portal as an admin.
- 2 Click the **Application > IdM Administration > SRG**.
- 3 In the left navigation bar, click **Users**.
- 4 In the list of users on the right, select **Admin** and click the edit icon.
- 5 Click **Remove Password**.
- 6 Click **Add Password**.
- 7 Enter a new admin password, and then click **Save**.





# 30 Renewing CDF Certificates

The validity of CDF certificate is one year. If you do not upgrade CDF within one year, the certificate will expire and you must renew the certificate. The CDF certificates can be renewed before and after expiration.

CDF contains the following certificates:

- ♦ Internal certificates which are used within the cluster nodes, such as `client.crt`, `client.key`, `server.crt`, `server.key`, `kubernetes.crt`, and `kubernetes.key`.
- ♦ External certificate which are used for the ingress service of the CDF management portal.

## Renewing Certificate Before Expiration

You can renew both internal and external certificates before expiration.

**To renew certificates before expiration:**

- 1 Log in to the master node.
- 2 Change to the directory:

```
cd <k8s_HOME>
```

By default, `k8s_HOME` is `/opt/kubernetes`.

- 3 (Conditional) For internal certificate, run the following command to generate new certificate:

```
./scripts/renewCert --renew -t internal
```

In a multi-node deployment, executing the above command automatically distributes the new certificate to all nodes in the cluster.

- 4 (Conditional) For external certificate, run the following command to generate new certificate:

```
./scripts/renewCert --renew -t external
```

## Renewing Certificates After Expiration

You can renew both internal and external certificates after expiration.

**To renew certificates after expiration:**

- 1 Log in to the master node.
- 2 Change to the following directory:

```
cd <k8s_HOME>
```

By default, `k8s_HOME` is `/opt/kubernetes`.

- 3 (Conditional) For internal certificate:

- 3a Run the following command to generate new `client.crt`, `client.key` and `server.crt` certificates:

```
./scripts/renewCert --renew -V 375 -t internal
```

**3b** (Conditional) If you have multiple master nodes, run the following on all the master node:

```
./scripts/renewCert --renew -t internal
```

**4** (Conditional) For external certificate:

- ◆ To generate new external self-signed certificates:

```
./scripts/renewCert --renew -t external
```

- ◆ To generate the external custom self-signed certificates:

```
./scripts/renewCert --renew -t external --tls-cert /<cert file directory>/  
<cert file> --tls-key <private key directory>/<private key> [--tls-cacert  
<CA cert directory>/<CA cert file>]
```

# 31 Changing the CDF Certificate Authority

The cluster maintains its own certificate authority (CA) to issue certificates for external communication. A self-signed CA is generated during the installation of CDF by default. Pods of the deployed products use the certificates generated by the CA on pod startup. You can change the CA with your own CA (well-known or organization's root CA) or you can generate a new CA and include the CA to the CDF.

---

**NOTE:** Changing the CA after Identity Intelligence deployment will necessitate uninstall and reinstall of the CDF suite. Uninstalling the CDF suite will uninstall all the installed capabilities such as Identity Intelligence, Transformation Hub, Analytics, and Investigate. As a result, we recommend that you perform this procedure when Identity Intelligence is first installed to avoid downtime and data loss.

---

- ◆ [“Generating a New CA” on page 147](#)
- ◆ [“Updating the CDF CA” on page 150](#)

## Generating a New CA

If you do not have a CA certificate, you can generate a new CA certificate as follows:

---

**NOTE:** When you are generating a new certificate, ensure that the validity is more than 365 days by specifying appropriate value.

---

- 1 Create a directory and configure the directory permissions:

```
mkdir /root/ca
cd /root/ca
mkdir certs crl newcerts private
chmod 700 private
touch index.txt
echo 1000 > serial
```

- 2 Open the configuration file in a text editor (`vi /root/ca/openssl.cnf`), and add the following content (values shown are examples; change parameter values to match yours):

```

# OpenSSL root CA configuration file.
# Copy to `/root/ca/openssl.cnf`.
[ ca ]
default_ca = CA_default
[ CA_default ]
# Directory and file locations.
dir = /root/ca
certs = $dir/certs
crl_dir = $dir/crl
new_certs_dir = $dir/newcerts
database = $dir/index.txt
serial = $dir/serial
RANDFILE = $dir/private/.rand
# The root key and root certificate.
private_key = $dir/private/ca.key.pem
certificate = $dir/certs/ca.cert.pem
# For certificate revocation lists.
crlnumber = $dir/crlnumber
crl = $dir/crl/ca.crl.pem
crl_extensions = crl_ext
default_crl_days = 30
# SHA-1 is deprecated, so use SHA-2 instead.
default_md = sha256
name_opt = ca_default
cert_opt = ca_default
default_days = 375
preserve = no
policy = policy_strict
[ policy_strict ]
# The root CA should only sign intermediate certificates that match.
# See the POLICY FORMAT section of `man ca`.
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
[ policy_loose ]
# Allow the intermediate CA to sign a more diverse range of certificates.
# See the POLICY FORMAT section of the `ca` man page.
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
[ req ]
# Options for the `req` tool (`man req`).
default_bits = 2048
distinguished_name = req_distinguished_name
string_mask = utf8only
# SHA-1 is deprecated, so use SHA-2 instead.
default_md = sha256
# Extension to add when the -x509 option is used.
x509_extensions = v3_ca
[ req_distinguished_name ]
countryName = Country
stateOrProvinceName = State
localityName = Locality

```

```

0.organizationName = EntCorp
organizationalUnitName = OrgName
commonName = Common Name
emailAddress = Email Address
# Optionally, specify some defaults.
countryName_default = <your country code>
stateOrProvinceName_default = <your state or province>
localityName_default =
0.organizationName_default = <your company name>
organizationalUnitName_default =
emailAddress_default =
[ v3_ca ]
# Extensions for a typical CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ v3_intermediate_ca ]
# Extensions for a typical intermediate CA (`man x509v3_config`).
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = client, email
nsComment = "OpenSSL Generated Client Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection
[ server_cert ]
# Extensions for server certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
nsCertType = server
nsComment = "OpenSSL Generated Server Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
[ crl_ext ]
# Extension for CRLs (`man x509v3_config`).
authorityKeyIdentifier=keyid:always
[ ocsp ]
# Extension for OCSP signing certificates (`man ocsp`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning

```

### 3 Generate a CA root key:

```

openssl genrsa -out private/ca.key.pem 4096
chmod 400 private/ca.key.pem

```

### 4 Create a CA cert:

```

openssl req -config openssl.cnf -key private/ca.key.pem -new -x509 -days 375 -
sha256 -extensions v3_ca -out certs/ca.cert.pem

```

**5** Verify the root CA:

```
chmod 444 certs/ca.cert.pem
```

```
openssl x509 -noout -text -in certs/ca.cert.pem
```

## Updating the CDF CA

You can update your own CA or newly generated CA by performing the following:

**1** Log in to the master node.

**2** Update the CA by executing the following command:

```
<K8S_HOME>/scripts/cdf-updateRE.sh write --re-key=private/ca.key.pem --re-  
crt=certs/ca.cert.pem --re-ca=certs/ca.cert.pem
```

**3** To verify, read the CDF CA file by executing the following command and ensure that it is same as the `ca.cert.pem` file. You must execute the following command on the initial master node.

```
<K8S_HOME>/scripts/cdf-updateRE.sh read
```

# 32 Retrieving CDF Root CA

You can retrieve the CDF root CA from a web browser or by using the command line.

- ♦ “Retrieving the CDF Root CA from Browser” on page 151
- ♦ “Retrieving the CDF Root CA Using Command Line” on page 151

## Retrieving the CDF Root CA from Browser

This section provides information about obtaining the CDF root CA from Google Chrome.

- 1 Specify the following URL in the browser:

```
https://<master_node_FQDN>:5443
```

- 2 Click the icon next to the left of the URL, then click **Certificate**.

- 3 Click **Certification Path**.

- 4 Double-click the CA certificate.

A pop-up window appears.

- 4a In the pop-up window, click **Details**, then click **Copy to File...**

- 4b Click **Next**.

- 4c Select **Base-64 encoded X.509 (.CER)** and click **Next**.

- 4d Specify a file name (for example, `ca.cer`) and click **Next**.

- 4e Click **Finish** and close the pop-up window.

- 5 (Conditional) If you have multiple CA certificates, repeat Step 4 for each CA certificate in the certificate chain.

## Retrieving the CDF Root CA Using Command Line

- 1 Log in to the initial master node of the cluster.
- 2 Execute the following command to retrieve the CDF CA certificate:

```
${K8S_HOME}/scripts/cdf-updateRE.sh read > ca.cer
```





# 33 Managing Vertica

For information about managing Vertica, such as modifying Vertica connection details, updating the search string, enabling data retention policy and so on, see the following:

- ♦ [Vertica Documentation](#)
- ♦ [Investigate Deployment Guide](#)



# VI Appendices

This section provides additional information for managing Identity Intelligence environment.

- ◆ [Appendix A, “Troubleshooting,” on page 157](#)
- ◆ [Appendix B, “Uninstalling Identity Intelligence,” on page 159](#)



# A Troubleshooting

Refer to the following section if you are experiencing a problem with Identity Intelligence.

- ♦ [“Recovering from Loss of Entity Data Being Collected from Identity Manager to Identity Intelligence” on page 157](#)
- ♦ [“Restarting the Node Fails with an Error” on page 157](#)

## Recovering from Loss of Entity Data Being Collected from Identity Manager to Identity Intelligence

**Issue:** You detect that Identity Intelligence fails to properly collect and store some entity data from Identity Manager.

**Workaround:** To recover from the data loss, you can migrate the entity data from the Identity Vault to Identity Intelligence in one of the following ways:

- ♦ Manually select the User, Group, Role, Resource, and Entitlement objects. Then use the **Migrate from Identity Vault** option in iManager to migrate those objects.
- ♦ Use the **Synchronize** option in iManager to allow the Identity Vault to automatically submit all objects.

To avoid storing duplicate data when you migrate entity data from the Identity Vault, Identity Intelligence preserves the data that is already present and adds the data that is missing.

## Restarting the Node Fails with an Error

**Issue:** When you restart the Identity Intelligence node that you installed with the supplied scripts, you will see `Failed to start reboot.target` error.

**Workaround:** To work around the issue, you must follow the instructions in the [Restarting Nodes by Using Scripts](#) section.



# B Uninstalling Identity Intelligence

You can uninstall Identity Intelligence and all other software, such as Transformation Hub, Analytics, and Investigate in the following ways:


- ♦ [“Uninstalling Manually” on page 159](#)
- ♦ [“Uninstalling by Using the Script” on page 160](#)

## Uninstalling Manually

To uninstall gracefully, first stop all Collectors and Connectors from sending events to Transformation Hub. Next, stop all consumers from receiving events and then perform uninstallation.

### Uninstalling Identity Intelligence

To uninstall Identity Intelligence and other software except Vertica, perform the following:

- 1 Log in to the CDF Management Portal.
- 2 Click **Deployment > Deployments**.
- 3 Click  of `arcsight-installer` and click **Uninstall**.

### Uninstalling Vertica

- 1 Log in to Vertica cluster node 1.
- 2 Change to the directory where you have Vertica installation packages:

```
cd <vertica_installer_directory>/arcsight-vertica
```

- 3 Uninstall Vertica using the command:

```
./vertica_installer uninstall
```

### Uninstalling CDF

- 1 Log in to the CDF master node.
- 2 Change to the CDF installation directory:

```
cd <k8s_home>
```

- 3 Uninstall CDF using the command:

```
./uninstall.sh -r false
```

- 4 Reboot the server.

# Uninstalling by Using the Script

You can uninstall by using the script only if the installer files are available in the following locations:

- ♦ **Kubernetes:** /opt/kubernetes
- ♦ **NFS volumes:** /opt/arcsight
- ♦ **Vertica Installer files:** /opt/Vertica
- ♦ **Vertica Installed at:** /opt/arcsight-vertica

If the installer files are in different location, you must uninstall manually using the CDF Management portal. For more information, see [Uninstalling Manually](#).

## To uninstall Identity Intelligence by using the script:

- 1 Log in to the master node as `root`.
- 2 Change to the directory where you extracted the Identity Intelligence installation files:  

```
cd <install_directory>
```
- 3 Execute the script by using the following command:  

```
./uninstall-single-node.sh
```