Version 2 Release 3

IBM i2 Enterprise Insight Analysis Deployment guide

SC27-5091-00

Note

Before using this information and the product it supports, read the information in "Notices" on page 67.

This edition applies to version 2, release 3, modification 2 of IBM® i2® Enterprise Insight Analysis (product number 5725-G23) and to all subsequent releases and modifications until otherwise indicated in new editions. Ensure that you are reading the appropriate document for the version of the product that you are using. To find a specific version of this document, access the Installing and Deploying section of the IBM Knowledge Center, and ensure that you select the correct version.

Contents

Contacting IBM Support

IBM Support provides assistance with product defects, answers FAQs, and helps users to resolve problems with the product.

About this task

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Support. Before contacting IBM Support, your company or organization must have an active IBM software subscription and support contract, and you must be authorized to submit problems to IBM. For information about the types of available support, see the Support portfolio topic in the *Software Support Handbook*.

Procedure

To contact IBM Support about a problem:

1. Define the problem, gather background information, and determine the severity of the problem. For more information, see the Getting IBM Support topic in the *Software Support Handbook*.
2. Gather diagnostic information.
3. Submit the problem to IBM Support in one of the following ways:
   - Online through the IBM Support Portal at Support Portal. You can open, update, and view all of your service requests from the Service Request portlet on the Service Request page.
   - By phone. For the phone number to call in your region, see the Directory of worldwide contacts web page at https://www.ibm.com/planetwide/

Results

If the problem that you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website daily, so that other users who experience the same problem can benefit from the same resolution.

Deployment types

Before you start a deployment of i2 Analyze, choose the type of deployment that you want to create. Use the following information to ensure that you choose the correct type of deployment for your requirements.

Example deployment

You can use an example deployment to learn about i2 Analyze, demonstrate the features of the system, and ensure that any software prerequisites are installed correctly on a single server.

When you create an example deployment, the deployment toolkit populates all of the mandatory configuration settings with default values and deploys on a single server. The deployment uses an example i2 Analyze schema, security schema, and data. Some configuration settings that are not mandatory for deployment are also populated to demonstrate extra features of the system.

For more information about example deployments, see "Creating an example deployment" on page 5.

Production deployment

A production deployment is available to analysts to complete mission critical analysis on real-world data. When you decide to create a production deployment of i2 Analyze, you must start from a clean installation of i2 Analyze.

The process for creating a production deployment involves a number of different deployment and configuration activities. As part of the process, you must develop an i2 Analyze schema and security schema for your data.

For more information about production deployments, see "Creating a production deployment" on page 13.

Creating an example deployment

If you want to understand what i2 Analyze is and demonstrate the features of the system, you can create an example deployment.

An example deployment uses default values that are provided by the deployment toolkit, and contains a configuration that demonstrates the features provided by i2 Analyze. You can also use an example deployment to verify that any prerequisites are installed correctly.

Creating an example with the Information Store

An installation of i2 Analyze includes example settings for deploying the server with an Information Store. After you deploy the Information Store, you can access the data that it contains by using Analyst's Notebook Premium.

Before you begin

Install IBM i2 Analyze and any software prerequisites. For more information, see “Installing i2 Analyze” on page 32. To deploy the preconfigured examples for the Information Store, you need either IBM Db2® or Microsoft SQL Server 2019. You do not need IBM HTTP Server.

If you are using SQL Server, download the Microsoft JDBC Driver 7.4 for SQL Server archive from https://www.microsoft.com/en-us/download/details.aspx?id=58505. Extract the contents of the download, and locate the `sqljdbc_7.4\enu\mssql-jdbc-7.4.1.jre8.jar` file.

About this task

The following procedure describes how to create an example deployment of i2 Analyze with the Information Store. The i2 Analyze toolkit contains an example configuration for the deployment. The `deployExample` task generates the default values for the mandatory settings and deploys the platform.

The example deployment demonstrates a working i2 Analyze system with an example user so that you can log in.

In the example deployment, i2 Analyze runs with the example security schema and matching Liberty security groups and users. The example user has the following credentials:

- The user name is `Jenny`
- The password is `Jenny`

The example deployment uses the `law-enforcement-schema.xml` schema as the i2 Analyze schema with the associated `law-enforcement-schema-charting-schemes.xml` as the charting scheme.

Procedure

1. Create the configuration directory:
   a) Navigate to the `\toolkit\examples\configurations\information-store-opal` directory.
      This directory contains the preconfigured files that you require to deploy a system that uses the Information Store to store data. The data can be accessed by using Analyst's Notebook Premium as a rich desktop client.
   b) Copy the `configuration` directory to the root of the toolkit.
      For example, `C:\IBM\i2analyze\toolkit\configuration`.

If you are using SQL Server as your database management system, you must complete extra configuration actions.

2. Copy the example `topology.xml` file for SQL Server from the `toolkit\configuration\examples\topology\sqlserver` to the `toolkit\configuration\environment` directory. Overwrite the existing `topology.xml` file in the destination directory.
3. Copy the `mssql-jdbc-7.4.1.jre8.jar` file that you downloaded to the `toolkit\configuration\environment\common\jdbc-drivers` directory.

Regardless of your database management system, you must complete the following steps after you create the configuration directory.

4. Specify the credentials to use for the deployment.

   For more information about credentials, see Modifying the credentials.

   a) Using a text editor, open the `toolkit\configuration\environment\credentials.properties` file.

   b) Enter the user name and password to use with the database.

   c) Enter the user name and password to use with the Solr index.

   d) Enter the password to use to encrypt LTPA tokens.

   e) Save and close the `credentials.properties` file.
5. Run the setup script to create the example deployment.

   a) Open a command prompt and navigate to the `toolkit\scripts` directory.

   b) To deploy the example, run the following command:

   ```
   setup -t deployExample
   ```

   c) To start the WebSphere® Application Server Liberty profile server, run the following command:

   ```
   setup -t start
   ```
6. Optional: To populate your Information Store with the provided example data for the `law-enforcement-schema.xml` schema, run the following command:

   ```
   setup -t ingestExampleData
   ```

What to do next

When you start i2 Analyze, the URI that users must specify is displayed in the console. For example, `This application is configured for access on http://`*`host_name`*`:9082/opal`.

Install Analyst's Notebook Premium and connect to your deployment. For more information, see Installing IBM i2 Analyst's Notebook Premium and Connecting IBM i2 Analyst's Notebook Premium to IBM i2 Analyze.

Creating an example with the Information Store and the i2 Connect gateway

An installation of i2 Analyze includes example settings for deploying the server with an Information Store and support for the i2 Connect gateway. Analyst's can access data store in the Information

Store in the usual manner, and the i2 Connect gateway enables analysts to search for and retrieve data from an example external data source.

Before you begin

Install IBM i2 Analyze and any software prerequisites. For more information, see “Installing i2 Analyze” on page 32. To deploy the preconfigured examples for the Information Store, you need either IBM Db2 or Microsoft SQL Server 2019. You do not need IBM HTTP Server.

If you are using SQL Server, download the Microsoft JDBC Driver 7.4 for SQL Server archive from https://www.microsoft.com/en-us/download/details.aspx?id=58505. Extract the contents of the download, and locate the `sqljdbc_7.4\enu\mssql-jdbc-7.4.1.jre8.jar` file.

Before you create the example deployment, you must download and install Node.js to host the example connector. Download Node.js for your operating system from: https://nodejs.org/en/download/. You can install Node.js with the default settings.

About this task

The following procedure describes how to create an example deployment of i2 Analyze with the Information Store and the i2 Connect gateway. To use a deployment with the i2 Connect gateway, you must obtain or create a custom connector to the external data source that you want to search. The i2 Analyze toolkit contains an example configuration for the deployment, and an example connector. The `deployExample` task generates the default values for the mandatory settings and deploys the platform.

The example deployment demonstrates a working i2 Analyze system with an example user so that you can log in. You can ingest example data into the Information Store and then perform searches and analysis, and use the example connector to query and retrieve data from an external data source.

In the example deployment, i2 Analyze runs with the example security schema and matching Liberty security groups and users. The example user has the following credentials:

- The user name is `Jenny`
- The password is `Jenny`

The example deployment uses the `law-enforcement-schema.xml` schema as the i2 Analyze schema with the associated `law-enforcement-schema-charting-schemes.xml` as the charting scheme.

Procedure

1. Create the configuration directory:
   a) Navigate to the `\toolkit\examples\configurations\information-store-daod-opal` directory.
      This directory contains the preconfigured files that you require to deploy a system that uses the i2 Connect gateway to connect to an external data source and uses the Information Store to store data.
   b) Copy the configuration directory to the root of the toolkit.
      For example, `C:\IBM\i2analyze\toolkit\configuration`.

If you are using SQL Server as your database management system, you must complete extra configuration actions.

2. Copy the example `topology.xml` file for SQL Server from the `toolkit\configuration\examples\topology\sqlserver` to the `toolkit\configuration\environment` directory. Overwrite the existing `topology.xml` file in the destination directory.

3. Copy the `mssql-jdbc-7.4.1.jre8.jar` file that you downloaded to the `toolkit\configuration\environment\common\jdbc-drivers` directory.

Regardless of your database management system, you must complete the following steps after you create the configuration directory.

4. Specify the credentials to use for the deployment.

   For more information about credentials, see Modifying the credentials.

   a) Using a text editor, open the `toolkit\configuration\environment\credentials.properties` file.

   b) Enter the user name and password to use with the database.

   c) Enter the user name and password to use with the Solr index.

   d) Enter the password to use to encrypt LTPA tokens.

   e) Save and close the `credentials.properties` file.

5. Run the setup script to create the example deployment.

   a) Open a command prompt and navigate to the `toolkit\scripts` directory.

   b) To deploy the example, run the following command:

   ```
   setup -t deployExample
   ```

6. Download and start the server that hosts the example connector.

   Note: The example connector uses port number 3700. Ensure that no other processes are using this port number before you start the connector.

   a) In a command prompt, navigate to the `toolkit\examples\connectors\example-connector` directory.

   b) To install the dependencies that are required for the example connector, run the following command:

   ```
   npm install
   ```

   Note: You must be connected to the internet to install the dependencies.

   c) To start the Node.js server, run the following command:

   ```
   npm start
   ```

7. Start i2 Analyze.

   a) Open a command prompt and navigate to the `toolkit\scripts` directory.

   b) To start i2 Analyze, run the following command:

   ```
   setup -t start
   ```

What to do next

When you start i2 Analyze, the URI that users must specify in Analyst's Notebook Premium is displayed in the console. For example, `This application is configured for access on http://`*`host_name`*`:9082/opal`.

Install Analyst's Notebook Premium and connect to your deployment. For more information, see Installing IBM i2 Analyst's Notebook Premium and Connecting IBM i2 Analyst's Notebook Premium to IBM i2 Analyze.

Production deployments of i2 Analyze use client-authenticated SSL communication between i2 Analyze and any connectors. This example does not use it, Analyst's Notebook Premium displays a warning to that effect when you open the external searches window. For more information about configuring client authenticated SSL, see Client authenticated Secure Sockets Layer with IBM i2 Connect.

You can create your own connectors to use with the deployment of i2 Analyze. For more information about creating your own connectors, see IBM i2 Analyze and i2 Connect.

Creating an example with the i2 Connect gateway

An installation of i2 Analyze includes example settings for deploying the server with support for i2 Connect. i2 Connect enables analysts to search for and retrieve data from external data sources by using the Opal quick search functions, and then analyze the results on a chart in Analyst's Notebook Premium.

Before you begin

Install IBM i2 Analyze and any software prerequisites. For more information, see “Installing i2 Analyze” on page 32. You do not need IBM HTTP Server.

Before you create the example deployment of i2 Analyze with the i2 Connect gateway, you must download and install Node.js to host the example connector. Download Node.js for your operating system from: https://nodejs.org/en/download/. You can install Node.js with the default settings.

About this task

To use a deployment with the i2 Connect gateway, you must obtain or create a custom connector to the external data source that you want to search. The i2 Analyze toolkit contains an example configuration for the deployment, and an example connector.

The example deployment demonstrates a working i2 Analyze system that can query and retrieve data from an external data source by using the i2 Connect gateway and an example connector. You can log in by using an example user. In the example deployment, i2 Analyze runs with the example security schema and matching Liberty security groups and users. The example user has the following credentials:

- The user name is Jenny
- The password is Jenny

Procedure

1. Create the configuration directory:
   a) Navigate to the `\toolkit\examples\configurations\daod-opal` directory.
      This directory contains the preconfigured files that you require to deploy a system that uses the i2 Connect gateway to connect to an external data source.
   b) Copy the `configuration` directory to the root of the toolkit.
      For example, `C:\IBM\i2analyze\toolkit\configuration`.
2. Specify the credentials to use for the deployment.
   For more information about credentials, see Modifying the credentials.
   a) Using a text editor, open the `toolkit\configuration\environment\credentials.properties` file.
   b) Enter the user name and password to use with the Solr index.
   c) Enter the password to use to encrypt LTPA tokens.

d) Save and close the `credentials.properties` file.

3. Run the setup script to create the example deployment.
   a) Open a command prompt and navigate to the `toolkit\scripts` directory.
   b) To deploy the example, run the following command:

```
setup -t deployExample
```

4. Download and start the server that hosts the example connector.

   Note: The example connector uses port number 3700. Ensure that no other processes are using this port number before you start the connector.

   a) In a command prompt, navigate to the `toolkit\examples\connectors\example-connector` directory.
   b) To install the dependencies that are required for the example connector, run the following command:

```
npm install
```

   Note: You must be connected to the internet to install the dependencies.

   c) To start the Node.js server, run the following command:

```
npm start
```

5. Start i2 Analyze.
   a) Open a command prompt and navigate to the `toolkit\scripts` directory.
   b) To start i2 Analyze, run the following command:

```
setup -t start
```

What to do next

When you start i2 Analyze, the URI that users must specify to connect to it from Analyst's Notebook Premium is displayed in the console. For example, `This application is configured for access on http://`*`host_name`*`:9082/opaldaod`.

Install Analyst's Notebook Premium and connect to your deployment. For more information, see Installing IBM i2 Analyst's Notebook Premium and Connecting IBM i2 Analyst's Notebook Premium to IBM i2 Analyze.

Production deployments of i2 Analyze use client-authenticated SSL communication between i2 Analyze and any connectors. This example does not use it, Analyst's Notebook Premium displays a warning to that effect when you open the external searches window. For more information about configuring client authenticated SSL, see Client authenticated Secure Sockets Layer with IBM i2 Connect.

You can create your own connectors to use with the deployment of i2 Analyze. For more information about creating your own connectors, see IBM i2 Analyze and i2 Connect.

Creating an example with the Analysis Repository

An installation of i2 Analyze includes example settings for deploying the server with an Analysis Repository data store. After you deploy the Analysis Repository, you can connect to it by using the Intelligence Portal and Analyst's Notebook Premium.

Before you begin

Install IBM i2 Analyze, a database management system, and the IBM HTTP Server. For more information, see "Installing i2 Analyze" on page 32.

About this task

The following procedure describes how to create an example deployment of i2 Analyze with the Analysis Repository that uses DB2 as the database management system. The i2 Analyze toolkit contains an example configuration for the deployment. The `deployExample` task generates the default values for the mandatory settings, provides the JDBC driver, and deploys the platform.

The example deployment demonstrates a working i2 Analyze system with an example user so that you can log in.

In the example deployment, i2 Analyze runs with the example security schema and matching Liberty security groups and users. The example user has the following credentials:

- The user name is `Jenny`
- The password is `Jenny`

The example deployment uses the `law-enforcement-schema.xml` schema as the i2 Analyze schema with the associated `law-enforcement-schema-charting-schemes.xml` as the charting scheme.

Procedure

1. Create the configuration directory:
   a) Navigate to the `\toolkit\examples\configurations\analysis-repository` directory.

      This directory contains the preconfigured files that you require to deploy a system that uses the Analysis Repository to store data. The data can be accessed using both the Intelligence Portal as a thin web client and Analyst's Notebook Premium as a rich desktop client.

   b) Copy the `configuration` directory to the root of the toolkit.
   For example: `C:\IBM\i2analyze\toolkit\configuration`
2. Specify the credentials to use for the deployment.
   For more information about credentials, see Modifying the credentials.
   a) Using a text editor, open the `toolkit\configuration\environment\credentials.properties` file.
   b) Enter the user name and password to use with the database.
   c) Enter the password to use to encrypt LTPA tokens.
   d) Save and close the `credentials.properties` file.
3. Run the setup script to create the example deployment.
   a) Open a command prompt and navigate to the `toolkit\scripts` directory.
   b) To deploy the example, run the following command:

```
setup -t deployExample
```

c) To start i2 Analyze, run the following command:

```
setup -t start
```

d) Start, or restart, the HTTP server that hosts the reverse proxy.

What to do next

Deploy any other components that are required for the deployment pattern. After you deploy all the components that you require, test your deployment by connecting a client to your deployment. For more information, see “Connecting clients” on page 63.

Creating a production deployment

The process of creating a production deployment is separated into a number of different activities, which you complete in an iterative process. The suggested process involves the creation and retention of several environments, each one focused on different aspects of a production deployment of i2 Analyze.

Planning for production

Before you can start to install and deploy i2 Analyze, you must first understand how i2 Analyze fits into your organization:

- Understand what i2 Analyze is.
- Understand the requirements of the deployment, and align these to a deployment pattern.
- Understand the data and security models of the environment that i2 Analyze is deployed in.

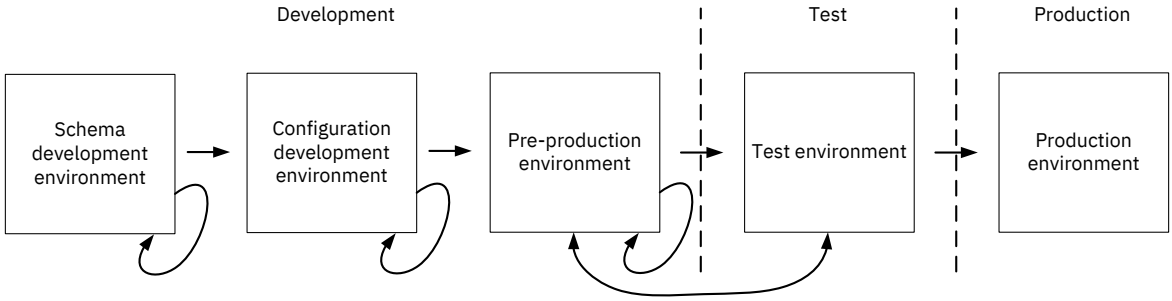
For more information, see the Understanding section.

Deploying

After you identify the requirements of the deployment, you can start to create the production deployment. The process of deploying i2 Analyze is completed in three phases that are explained in “Deployment phases and environments” on page 13.

Deployment phases and environments

The process of creating a production deployment involves distinct phases that focus on different aspects of an i2 Analyze deployment. You should complete the activities in each phase in distinct environments, which you should retain to refer to later in the process.



There are three phases in the process to deploy i2 Analyze into production.

Development

The *development phase* is where you configure i2 Analyze to meet the requirements of the final deployment. In this phase, you develop the configuration in an iterative process that involves a number of configuration changes and deployments of the system. During this phase, the lifetime of a deployment is short.

Test

The *test phase* is where you deploy i2 Analyze for testing. In the test phase, you deploy i2 Analyze with the configuration from the development phase and perform comprehensive testing of the system.

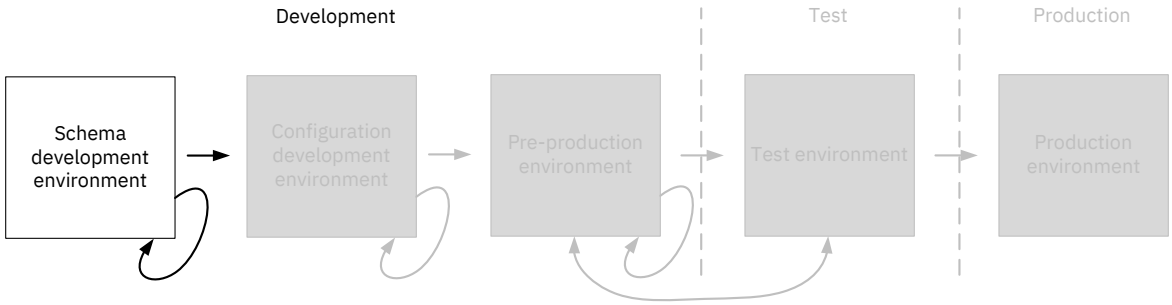
Production

The *production phase* is where you deploy i2 Analyze into production. i2 Analyze is deployed with the configuration that you tested in the test phase. In production, the deployment is fully operational and used by analysts to complete mission critical work.

To start creating your production deployment, create your "Schema development environment" on page 14.

Schema development environment

The first task in the development phase, is to develop the i2 Analyze schema and security schema for the deployment. You can use a *schema development environment*, which features a single-server deployment of i2 Analyze with the i2 Connect gateway to develop the schemas.



Creating the schema development environment

In the schema development environment, you first populate the values of some of the mandatory settings for deploying i2 Analyze. After you deploy i2 Analyze for the first time, you can then develop the schemas to meet your requirements.

About this task

The schema development environment features an instance of i2 Analyze with the i2 Connect gateway; there is no Information Store. By using a deployment without a database, you can quickly prototype changes to the schema and security schema, and then visualize the changes in Analyst's Notebook Premium.

Procedure

1. Install any prerequisite software to prepare your server for the schema development environment.
   - For the schema development environment, prepare your server for the i2 Connect deployment topology.

- Install Analyst's Notebook Premium to connect to your deployment and IBM i2 Analyze Schema Designer to edit your schema files. For more information, see “Installing IBM i2 Analyst's Notebook Premium” on page 37.

2. In the deployment toolkit that you installed, copy the `toolkit\examples\configurations\daod-opal\configuration` directory to the `toolkit` directory.
3. Enter a user name and password to use with the Solr index and a password to encrypt the LTPA tokens in the `configuration\environment\credentials.properties` file.

   For more information about credentials in i2 Analyze, see .
4. In a command prompt, navigate to the `toolkit\scripts` directory, and run the following command to populate some mandatory settings with default values:

   ```
   setup -t generateDefaults
   ```

   The `environment.properties` and `topology.xml` are modified by this toolkit task. For more information about the default values that are provided, see Configuration files reference.
5. In i2 Analyze Schema Designer, either create a new schema or open one of the examples to modify.

   For information about creating or modifying schema files, see Creating schemas and Charting schemes.

   Example schema files are located in sub directories of the `configuration\examples\` directory. For more information, see “Example schemes” on page 40.

   a) Save the initial version of your schema in the `configuration\fragments\common\WEB-INF\classes` directory.

      A charting scheme file is saved in the same location when you save the schema.

   b) Keep your schema file open in Schema Designer so that you can complete more modifications after you deploy i2 Analyze.
6. Copy the example security schema to the `configuration\fragments\common\WEB-INF\classes` directory.

   The example security schema file is located in the `configuration\examples\security-schema` directory. For more information, see “Example schemes” on page 40.
7. In the `configuration\fragments\common\WEB-INF\classes\ApolloServerSettingsMandatory.properties` file, set the values of the following settings to the file names of your schema, charting scheme, and security schema:

   - `SchemaResource`
   - `ChartingSchemesResource`
   - `DynamicSecuritySchemaResource`

   For example, `SchemaResource=custom-schema.xml`
8. Deploy i2 Analyze:

   ```
   setup -t deploy
   ```
9. Create an example user that you can use to log in:

   ```
   setup -t ensureExampleUserRegistry
   ```

   The user has the user name 'Jenny' and the password 'Jenny'.
10. Start i2 Analyze:

```
setup -t startLiberty
```

When you start i2 Analyze, the URI that users must specify to connect to it from Analyst's Notebook Premium is displayed in the console. For example, `This application is configured for access on http://host_name:9082/opaldaod`.

11. Connect to your deployment by using Analyst's Notebook Premium. Log in with the example 'Jenny' user.

    For more information, see Connecting IBM i2 Analyst's Notebook Premium to IBM i2 Analyze.

12. In Analyst's Notebook Premium, create items on the chart to visualize the i2 Analyze schema by using the **Opal DAOD** palette.

What to do next

After you deploy i2 Analyze with the initial schema files, you can develop them for your own data requirements:

- “Updating the i2 Analyze schema and charting scheme” on page 16
- “Updating the i2 Analyze security schema” on page 17

After you develop your schema files, you move to configuration development environment. In the configuration development environment, you deploy i2 Analyze with the schemas that you developed in your schema development environment.

You can choose whether to retain your schema development environment or not. If need to make small changes to your schemas later, you can do this in your configuration development environment. If you need to make more substantial changes, you might create another schema development environment.

When you have finalized your schemas, create the “Configuration development environment” on page 18.

Updating the i2 Analyze schema and charting scheme

In the schema development environment, you can quickly deploy your changes to the i2 Analyze schema and charting scheme. Use this environment to develop the schema and charting scheme for your production deployment.

About this task

Develop a schema and associated charting scheme to meet your data requirements. Then update your schema development deployment with the modified schema and charting scheme.

Procedure

1. In i2 Analyze Schema Designer, modify the schema that is deployed in your schema development environment.

   For information about creating or modifying your schema files, see Creating schemas and Charting schemes.

After you modify your schema, update the deployment with your changes.

2. Update and redeploy the system:

```
setup -t updateSchema
setup -t deployLiberty
```

3. Start i2 Analyze:

```
setup -t startLiberty
```

4. Test the changes to the schema and charting scheme by connecting to your deployment in Analyst's Notebook Premium and modeling representative data on the chart by using the **Opal DAOD** palette.

What to do next

Repeat this process until your schema and charting scheme meet your requirements. When your schema development is complete, store your schema and charting scheme files in a version control system.

This is not the final chance to modify your schema, however your schema should contain all the entity and link types that you require, and most of the property types.

When you are satisfied with your schema, develop the security schema for the deployment. For more information, see “Updating the i2 Analyze security schema” on page 17.

Updating the i2 Analyze security schema

In the schema development environment, you can quickly deploy your changes to the i2 Analyze security schema. Use this environment to develop the security schema for your production deployment.

About this task

Develop a security schema to meet your security requirements. Then update your schema development environment with the modified security schema.

Procedure

1. In an XML editor, either create a new security schema or open the one that is deployed in the schema development environment.

   For information about creating or modifying a security schema file for i2 Analyze, see The i2 Analyze Security schema.

   a) Save your security schema in the `toolkit\configuration\fragments\common\WEB-INF\classes` directory.

   b) Ensure that your security schema file is specified in `configuration\fragments\common\WEB-INF\classes\ApolloServerSettingsMandatory.properties`.

After you modify your security schema, update the deployment with your changes.

2. Update and redeploy the system:

```
setup -t updateSecuritySchema
setup -t deployLiberty
```

3. Start i2 Analyze:

```
setup -t startLiberty
```

4. If you changed the names of the user groups in the security schema, update the basic user registry to match the new names.

   For more information, see Configuring the Liberty user registry.

5. Test the changes to the security schema by connecting to the deployment in Analyst's Notebook Premium as different users and changing the security permissions on records that you create.

What to do next

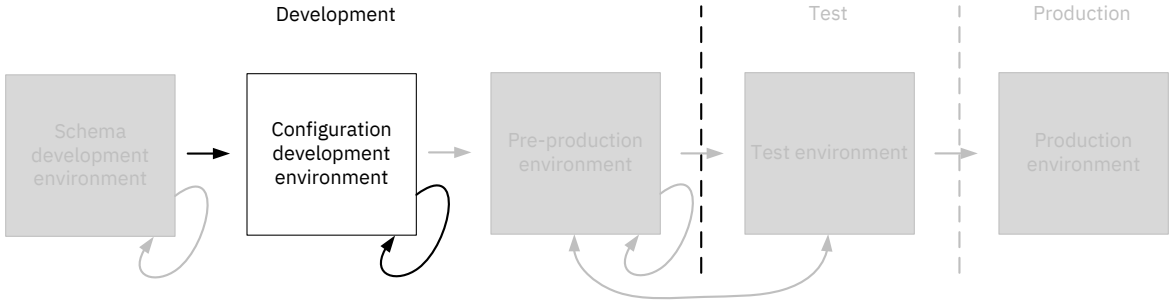
Repeat this process until your security schema meets your requirements. When your security schema development is complete, store your security schema and Liberty user registry files in a version control system.

This is not the final time that you can modify the security schema, but you should aim to have most of the security dimensions and dimension values defined.

After you finish developing your schema files, create the "Configuration development environment" on page 18.

Configuration development environment

In the configuration development environment, you configure i2 Analyze to meet your requirements. The aspects of i2 Analyze that you might configure include how your data is added to the system and how analysts interact with that data.



After you create your schemas, you can use a *configuration development environment* to configure i2 Analyze in a single server environment. In the configuration development environment, you deploy i2 Analyze with the same data stores as your intended production system.

In this deployment of i2 Analyze, you use the schema files that you previously created. If you have not created your schema files, complete the instructions in "Schema development environment" on page 14.

Creating the configuration development environment

When you create the configuration development deployment, you deploy i2 Analyze on a single server in the same deployment pattern as your intended production system.

Before you begin

If you intend your production deployment to contain the i2 Connect gateway only, use your schema development environment as your configuration development environment.

Ensure that you have access to the following files so that you can copy them to the configuration development environment:

- i2 Analyze schema
- Charting scheme
- Security schema
- `user-registry.xml`

Procedure

1. Install any prerequisite software to prepare your server for the configuration development environment.
   - For the configuration development environment, prepare your server for the Single server deployment topology with the same data store and database management system as in your intended production system.
2. Create the configuration directory in the i2 Analyze deployment toolkit.
   a) Navigate to the `toolkit\examples\configurations` directory where you installed i2 Analyze.
   b) Copy the `configuration` directory from your chosen base configuration to the root of the toolkit.
   For example, copy the `toolkit\examples\configurations\information-store-opal\configuration` directory to the `toolkit` directory.
   For more information about the different base configurations, see "The base configurations" on page 39.
3. If you are using SQL Server as your database management system, copy the example `topology.xml` file from `configuration\examples\topology\sqlserver` to the `configuration\environment` directory. Overwrite the existing `topology.xml` file in the destination directory.
4. If you are using IBM HTTP Server, in the `topology.xml` file set the value of the `http-server-host` attribute to `true`.

To allow the deployment toolkit to create and modify components of i2 Analyze, you provide user names and password for each of the components in your configuration.

5. Specify the user names and passwords to use for the deployment in the `toolkit\configuration\environment\credentials.properties` file.
   For more information about credentials in i2 Analyze, see .
6. Copy the JDBC driver to use with the deployment to the `configuration\environment\common\jdbc-drivers` directory.
   For more information, see .
   You do not need to provide a JDBC driver for an i2 Connect gateway only deployment.
7. Use the deployment toolkit to populate some of the mandatory settings with default values:

   ```
   setup -t generateDefaults
   ```

   The `environment.properties` and `topology.xml` are modified by this toolkit task. For more information about the default values that are provided, see Configuration files reference.
8. Specify the i2 Analyze schema, charting scheme, and security schema that you previously developed.
   a) Copy your i2 Analyze schema, charting scheme, and security schema files to the `configuration\fragments\common\WEB-INF\classes` directory.
   b) In the `configuration\fragments\common\WEB-INF\classes\ApolloServerSettingsMandatory.properties` file, set the values of the following settings to the file names of the schema, charting scheme, and security schema:
      - `SchemaResource`
      - `ChartingSchemesResource`
      - `DynamicSecuritySchemaResource`

For example, `SchemaResource=custom-schema.xml`.

9. Deploy i2 Analyze:

```
setup -t deploy
```

10. Copy the user registry file from your schema development environment to the `deploy\wlp\usr\shared\config` directory in the new deployment.
11. Start i2 Analyze:

```
setup -t start
```

The URI that users must specify is displayed in the console. For example, `This application is configured for access on http://host_name:9082/opal`.

What to do next

After you deploy i2 Analyze, you can begin “Developing the i2 Analyze configuration ” on page 20.

Developing the i2 Analyze configuration

At this stage, use the configuration development environment to define how the i2 Analyze application behaves specifically for your organization.

About this task

Developing the configuration is separated into two parts, enabling i2 Analyze to work with your data and defining how analysts interact with your data.

The i2 Analyze deployment in the configuration development environment is not your final deployment. When you are configuring the deployment, use small amounts of data with a few users to ensure that the processes are functional.

Procedure

There are three methods that you can configure to get your data into i2 Analyze for analysis.

1. Develop the process for ingesting data into the Information Store, refer to Ingesting data into the Information Store.
2. Develop connections to external data sources, refer to Connecting to external data sources.
3. Ensure that analysts can import and create representative records in Analyst's Notebook Premium. Then, if required, upload records to the Information Store. For more information, see Import data and Create i2 Analyze chart items.

During this process, you might realize that your i2 Analyze schema or security schema are not correct for your data. You can update the deployed schemas to better represent you data and security model. Some changes require you to remove and recreate the underlying database.

4. To update your deployed i2 Analyze schema, refer to Configuring the schema.
5. To update your deployed security schema, refer to Configuring the security schema.

After you develop the mechanisms for making data available to analysts, you can configure how analysts interact with the data when they use the system.

6. To configure which features or types of commands analysts can access, refer to Controlling access to features.
7. To configure how analysts search for information, and the options that are available to them, refer to Configuring search.
8. To configure how analysts can identify matching records, refer to Configuring matching.

9. To configure user security, refer to Configure user authentication and authorization.

   For more information about the configuration changes that you can make, see Configuring i2 Analyze.

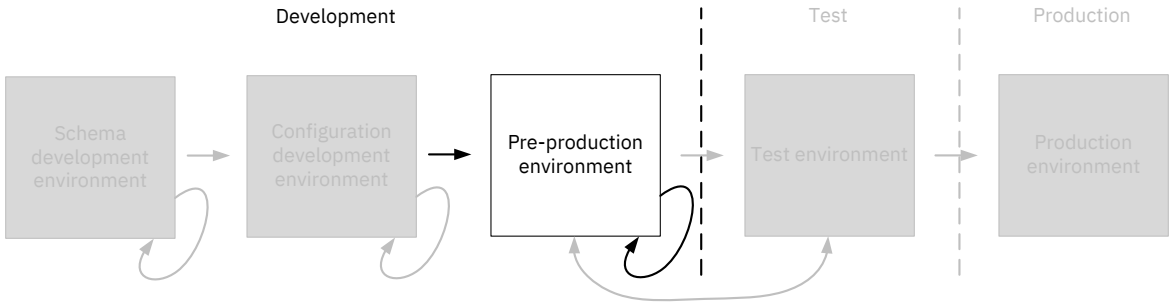
What to do next

After you configure your deployment sufficiently in the single-server environment, you can move another environment that is more representative of the production deployment. Keep your configuration development environment in place so that you can access the `configuration` directory in later phases of the production process, and return to it to make further configuration changes.

Next, create the "Pre-production environment" on page 21.

Pre-production environment

Your target deployment topology is probably different from the single-server configuration development environment. In the pre-production environment, deploy i2 Analyze in the deployment topology that matches your intended production deployment.



After you develop the i2 Analyze configuration, you can use a more representative *pre-production environment* for aspects of the configuration that rely on environment-specific variables. In this environment, deploy i2 Analyze in the same physical deployment topology as the target production deployment. For example, you might modify the configuration to deploy the components of i2 Analyze on multiple servers.

Creating the pre-production environment

In the pre-production environment, you can deploy i2 Analyze in the same physical deployment topology as your target production deployment. The process starts with your configuration from the configuration development environment, which you modify to match your chosen deployment topology.

Before you begin

Ensure that you have access to the following files and directories so that you can copy them to the pre-production development environment:

- The `toolkit\configuration` directory from the configuration development environment
- Liberty user configuration, for example:
  - `user-registry.xml`

  - `server.xml`
- Any configuration that you completed in the Information Store database. For example, your merged property values definition views.
- Any certificates and certificate stores that are required

About this task

Update your configuration to match the physical deployment topology of the pre-production environment.

Procedure

1. Install any prerequisite software to prepare your servers for the pre-production environment.

   For the pre-production environment, use the same deployment topology as your intended production environment. For more information, see "Deployment topologies" on page 29.
2. Copy the `toolkit\configuration` directory from the configuration development environment, to the `toolkit` directory at the root of the deployment toolkit on the Liberty server in the pre-production environment.
3. Update the values for any configuration settings that are specific to the environment.
   a) If you are creating a deployment with a database that is remote from the Liberty server, follow the instructions in "Specifying remote database storage" on page 51 to update the `environment.properties` and `topology.xml` files for this deployment topology.
   b) If you are creating a deployment with multiple Solr and ZooKeeper servers, follow the instructions in "Specifying remote Solr and ZooKeeper servers" on page 54 to update the `topology.xml` file for these deployment topologies.

   The `environment.properties`, `http-server.properties`, and `topology.xml` contain host name and file path settings that you might need to update for the servers in your pre-production environment. For more information, see Configuration files reference.
4. Specify the URI that clients use to connect to i2 Analyze.
   a) In the `configuration\fragments\opal-services\WEB-INF\classes\DiscoClientSettings.properties` file, set the value of the `FrontEndURI` setting to the URI that can be used to connect to your deployment of i2 Analyze.

   For more information, see Specifying the connection URI.
5. Deploy and start i2 Analyze:
   - In a single-server topology, see "Deploying i2 Analyze" on page 57.
   - In a multiple-server topology, see "Deploying i2 Analyze on multiple servers" on page 57.

After you deploy i2 Analyze, you can replicate any configuration changes that are not stored in the configuration of i2 Analyze.

6. Configure Liberty security for your environment. To do this, repeat any changes that you made to the Liberty configuration in the previous environment.

   This might involve copying the user registry file, or updating the `server.xml` file.
7. Complete any configuration changes in the Information Store database.
   a) If you created any rules or schedules to delete records by rule, replicate the rules and schedules that you created in the previous environment.
   b) If you created any merged property values definition views for your ingestion process, replicate the view definition that you created in the previous environment.

What to do next

After you deploy i2 Analyze in the pre-production development environment, you might want to configure aspects of the deployment that are deployment topology specific. For example, in a multiple-server environment you can secure the connections between servers by using SSL.

To configure i2 Analyze in pre-production, see "Configuring i2 Analyze in pre-production" on page 23.

Configuring i2 Analyze in pre-production

In the pre-production environment, configure the aspects of i2 Analyze that depend on the deployment topology.

Procedure

1. To configure SSL connections between the servers in your deployment, refer to Secure Sockets Layer connections with i2 Analyze.
2. To develop the process for backing up and restoring your deployment of i2 Analyze, refer to Backing up a deployment.

What to do next

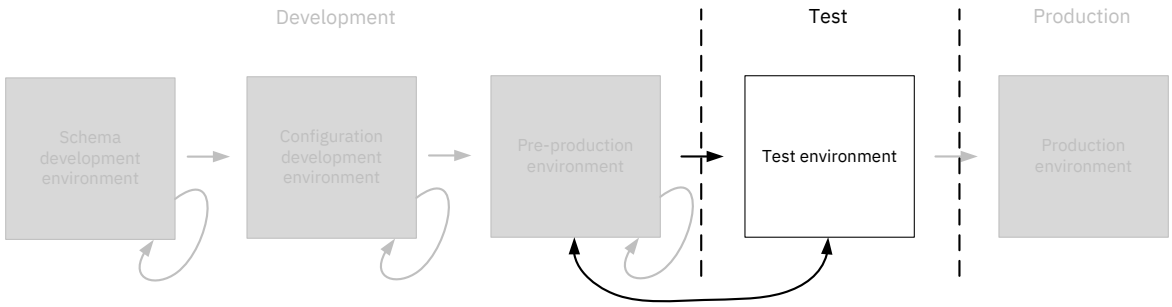
After you configure i2 Analyze sufficiently, you can move to the test environment. Keep your pre-production environment in place so that you can access the `configuration` directory in later phases of the production process, and return to it to make further configuration changes.

Next, create the "Test environment" on page 23.

Test environment

The second phase of creating a production deployment focuses on testing. This phase is important, because it enables you to identify any changes to the environment or configuration that must be completed before you deploy in to production.

About this task



You use a *test environment* to test the deployment to ensure that it meets the requirements of the production deployment. The test environment should match the production environment as closely as possible. In the test environment, perform comprehensive testing of the deployment against the final requirements with a selection of the users and a sample data set.

Creating the test environment

In the test environment, you deploy i2 Analyze with the configuration that you developed previously. With i2 Analyze deployed in the test environment, you can perform comprehensive testing of the deployment to confirm that it meets your production requirements.

Before you begin

Ensure that you have access to the following files and directories so that you can copy them to the configuration development environment:

- The `toolkit\configuration` directory from the pre-production deployment
- Liberty user configuration, for example the `server.xml` file
- Any configuration that you completed in the Information Store database. For example, your merged property values definition views.
- Any certificates and certificate stores that are required

Procedure

1. Install any prerequisite software to prepare your servers for the test environment.

   For the test environment, use the same deployment topology as in your pre-production environment. For more information, see "Deployment topologies" on page 29.
2. Copy the `toolkit\configuration` directory from the pre-production environment to the `toolkit` directory at the root of the deployment toolkit on the Liberty server in the test environment.
3. Update the values for any configuration settings that are specific to the environment.

   The `environment.properties`, `http-server.properties`, and `topology.xml` contains settings that you might need to update. For more information, see Configuration files reference.
4. Specify the URI that clients use to connect to i2 Analyze.
   a) In the `configuration\fragments\opal-services\WEB-INF\classes\DiscoClientSettings.properties` file, set the value of the `FrontEndURI` setting to the URI that can be used to connect to your deployment of i2 Analyze.

   For more information, see Specifying the connection URI.
5. Deploy and start i2 Analyze:
   - In a single-server topology, see "Deploying i2 Analyze" on page 57.
   - In a multiple-server topology, see "Deploying i2 Analyze on multiple servers" on page 57.

After you deploy i2 Analyze, you can replicate any configuration changes that are not stored in the configuration of i2 Analyze.

6. Configure Liberty security for your environment. To do this, repeat any changes that you made to the Liberty configuration in the previous environment.

   This might involve copying the user registry file, or updating the `server.xml` file.
7. Complete any configuration changes in the Information Store database.
   a) If you created any rules or schedules to delete records by rule, replicate the rules and schedules in the current environment.
   b) If you created any merged property values definition views for your ingestion process, replicate the view definitions in the current environment.

What to do next

After you deploy i2 Analyze in the test environment, you might want to test your deployment to ensure that it meets the requirements of your organization.

For information about testing i2 Analyze, see “Testing your deployment” on page 25.

Testing your deployment

In your test environment, complete the comprehensive testing of the system that is required to prove that the configuration is ready for production.

About this task

Create a test charter that contains a list of tests that you can use to verify that your deployment is ready for production.

In your test environment, you should use real data at a representative volume.

Procedure

1. Create the test charter.

   Focus your test charter in the following areas that are applicable to your deployment:

   - Data ingestion into the Information Store
   - Data deletion from the Information Store
   - Data acquisition from external sources
   - Searching and analyzing data
   - Performance of the system
   - Backup and recover the system

After you create a test charter for your deployment, you can start begin your testing.

2. Complete each test in your charter, and record if the deployment passed or failed each test.
3. If any of the tests fail, return to your development or pre-production environment and change the configuration. Then, deploy the same change to the test environment and repeat the testing.

What to do next

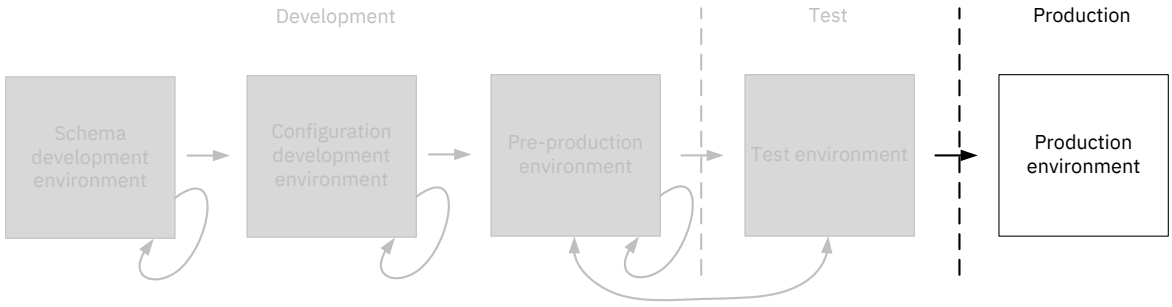
After you test i2 Analyze sufficiently, you can move to the production phase. Keep your test environment in place, so that you can access the `configuration` directory in later phases of the production process and continue to test any changes to your configuration.

Next, complete the instructions in “Production environment” on page 26.

Production environment

The third phase of the deployment process focuses on deploying into production. In this phase you make your deployment of i2 Analyze available to analysts.

About this task



You use a *production environment* to host the deployment in production. When you have a configuration of i2 Analyze that passed your test phase, you can deploy i2 Analyze with that configuration into your production environment.

In the production environment, run the tests again to confirm that the deployment is working successfully. If these tests are successful, you can make the deployment available to analysts. If the tests are not successful, you must return to your test or development environment to make any necessary changes. Then, complete the test phase once more.

Creating the production environment

To deploy i2 Analyze in production, you use the i2 Analyze configuration that you developed previously. Before you deploy i2 Analyze, update the configuration to reflect any environment changes.

Before you begin

Ensure that you have access to the following files and directories so that you can copy them to the configuration development environment:

- The `toolkit\configuration` directory from the test deployment
- Liberty user configuration, for example the `server.xml` file
- Any configuration that you completed in the Information Store database. For example, your merged property values definition views.
- Any certificates and certificate stores that are required

Procedure

1. Install any prerequisite software to prepare your servers for the production environment.

   For the production environment, use the same deployment topology as in your test environment. For more information, see “Deployment topologies” on page 29.
2. Copy the `toolkit\configuration` directory from the test environment to the `toolkit` directory at the root of the deployment toolkit on the Liberty server in the production environment.

3. Update the values for any configuration settings that are specific to the environment.

   The `environment.properties`, `http-server.properties`, and `topology.xml` contains settings that you might need to update. For more information, see Configuration files reference.
4. Specify the URI that clients use to connect to i2 Analyze.
   a) In the `configuration\fragments\opal-services\WEB-INF\classes\DiscoClientSettings.properties` file, set the value of the `FrontEndURI` setting to the URI that can be used to connect to your deployment of i2 Analyze.

   For more information, see Specifying the connection URI.
5. Deploy and start i2 Analyze:
   - In a single-server topology, see "Deploying i2 Analyze" on page 57.
   - In a multiple-server topology, see "Deploying i2 Analyze on multiple servers" on page 57.

After you deploy i2 Analyze, you can replicate any configuration changes that are not stored in the configuration of i2 Analyze.

6. Configure Liberty security for your environment. To do this, repeat any changes that you made to the Liberty configuration in the previous environment.

   This might involve copying the user registry file, or updating the `server.xml` file.
7. Complete any configuration changes in the Information Store database.
   a) If you created any rules or schedules to delete records by rule, replicate the rules and schedules in the current environment.
   b) If you created any merged property values definition views for your ingestion process, replicate the view definitions in the current environment.

What to do next

Complete testing of the deployment to ensure that it is working in the production environment before you make i2 Analyze available to analysts.

Deployment resources

The deployment resources section contains information that is referenced elsewhere in the deployment section. The information is used in a many places throughout the deployment process.

Deployment topologies

You can use the deployment toolkit to deploy i2 Analyze in a number of topologies. Throughout the process of creating a production deployment, you might deploy i2 Analyze in a number of the topologies depending on the purpose of the deployment.

The following diagrams show the servers that are used in each deployment topology, the prerequisites that are required on each server, and the components of i2 Analyze that are deployed.

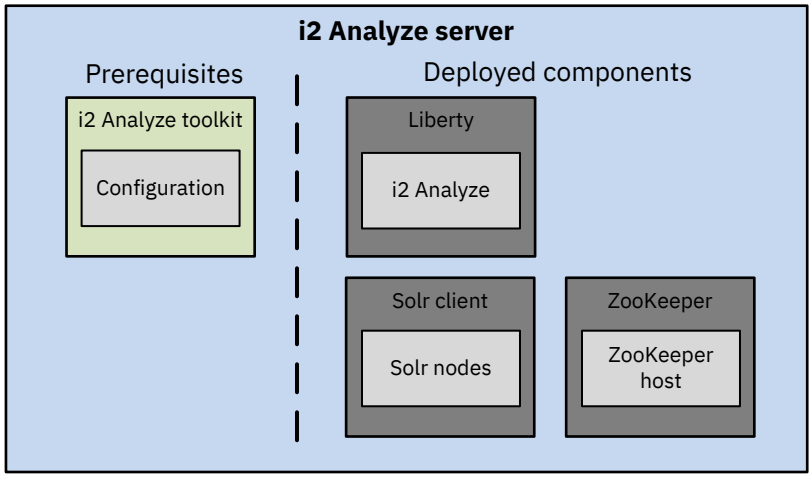
You can deploy i2 Analyze in the following physical deployment topologies:

- “i2 Connect (single server)” on page 29
- “Single server” on page 29
- “Multiple servers” on page 30

i2 Connect (single server)

In the i2 Connect deployment, all of the components of i2 Analyze are on the same server. No database management system is required.

i2 Connect only topology



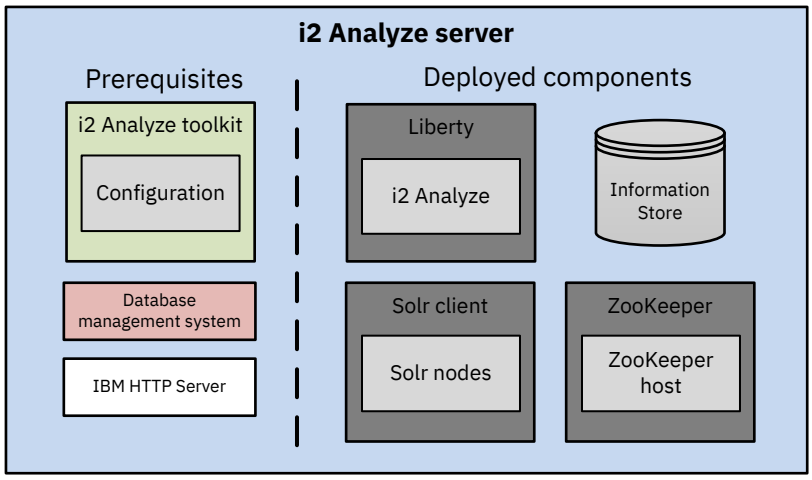
On the i2 Analyze server, install the prerequisites as follows:

- “Installing i2 Analyze” on page 32

Single server

In the single-server topology, all of the components of i2 Analyze are on the same server.

Single-server topology



On the i2 Analyze server, install the prerequisites:

- “Installing i2 Analyze” on page 32
- “Database management system” on page 33

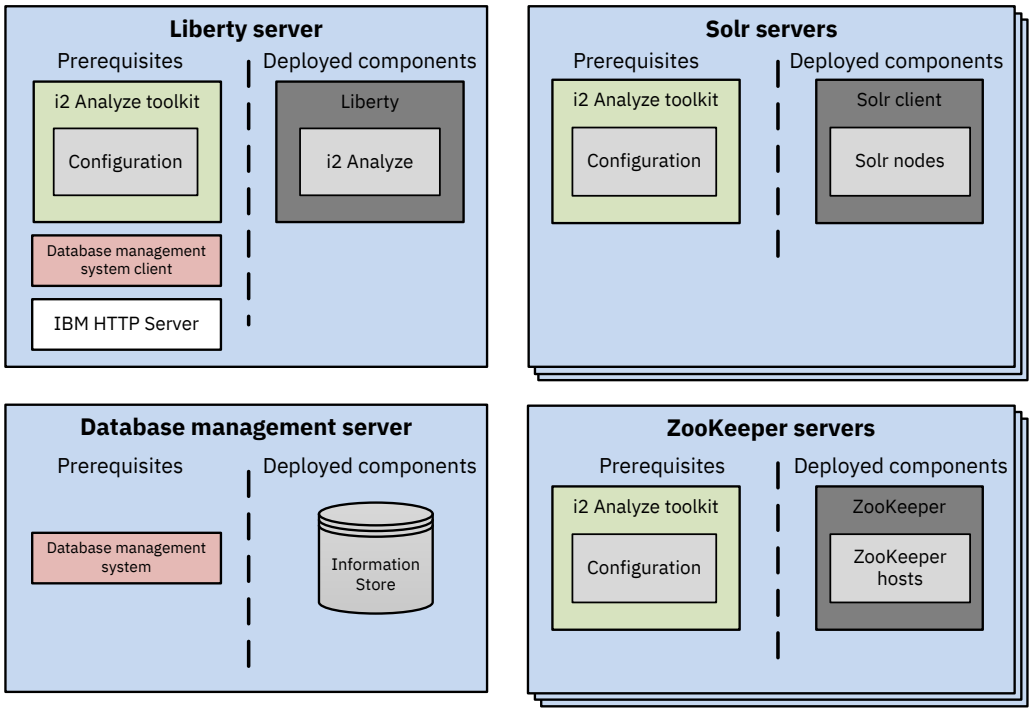
Optionally, you can also install the HTTP Server:

- “IBM HTTP Server and Web Server Plug-ins for WebSphere Application Server” on page 36

Multiple servers

In the multiple servers topology, each component of i2 Analyze is on its own server.

Multiple-server topology



In the diagram, Liberty, Solr, ZooKeeper, and the Information Store are each deployed on their own server.

You can also have a multiple-server deployment topology any of number the components are located on the same server.

On the Liberty server, install i2 Analyze, the database management system client, and optionally the IBM HTTP Server:

- “Installing i2 Analyze” on page 32
- Db2 client or SQL Server client
- “IBM HTTP Server and Web Server Plug-ins for WebSphere Application Server” on page 36

On your database management server, install your database management system:

- “Database management system” on page 33

On each Solr and ZooKeeper server, install i2 Analyze:

- “Installing i2 Analyze” on page 32

You cannot deploy i2 Analyze with the Analysis Repository in the multiple-server topology.

Installing components of IBM i2 Enterprise Insight Analysis

Installing i2 Analyze

You can install i2 Analyze by extracting an archive file. By using the archive file to install i2 Analyze, you do not need to install IBM Installation Manager on the server that you install i2 Analyze on.

Before you begin

For details of the system requirements, see Software Product Compatibility Reports.

To install i2 Analyze, you must have the i2 Analyze version 4.3.2 distribution. Choose one of the following distributions to install i2 Analyze from:

- IBM i2 Analyze V4.3.2 (Archive install) for Windows
- IBM i2 Analyze V4.3.2 (Archive install) for Linux

About this task

Installing i2 Analyze by extracting an archive file is useful when you are installing and deploying i2 Analyze on multiple servers, or on a server without a graphical user interface.

i2 Analyze is provided in a `.zip` archive file for Windows, and a `.tar.gz` archive file for Linux. To install i2 Analyze, extract the archive file and then accept the license agreement.

Procedure

1. Download the i2 Analyze distribution file for your operating system, and extract the contents into one of the following directories:
   - On Windows, `C:\IBM\i2analyze`
   - On Linux, `/opt/IBM/i2analyze`

   The following files and directories are present in the `IBM\i2analyze` directory:
   - `license`
   - `swidtag`
   - `toolkit`
   - `license_acknowledgment.txt`

Before you can use i2 Analyze, you must read and accept the license agreement and copyright notices.

2. In a text editor, open the `notices` file and the license file for your language from the `i2analyze\license` directory.
   For example, the English license is in the `LA_en` file.
3. Accept the license and copyright notices.
   a) Open the `IBM\i2analyze\license_acknowledgment.txt` file.
   b) To accept the license and copyright notices, change the value of `LIC_AGREEMENT` to `ACCEPT`.
   For example:

   ```
   LIC_AGREEMENT = ACCEPT
   ```

   c) Save and close the file.

Software prerequisites

The software prerequisites that you require depend on the deployment pattern and deployment topology of i2 Analyze that you want to deploy.

The software prerequisites that you might require are:

- A supported database management system.
  - IBM Db2 Enterprise Server, Advanced Enterprise Server, Workgroup Server, or Advanced Workgroup Server editions at version 10.5 Fix Pack 10 or later and version 11.1 Fix Pack 3 or later, or Advanced and Standard editions at version 11.5.

    Note:

    - IBM Db2 Advanced Workgroup Server Edition version 11.1 is included with i2 Analyze.
    - If you are deploying the Analysis Repository, you must use version 10.5 or 11.1 of IBM Db2.
  - Microsoft SQL Server Standard or Enterprise at version 14.0 (2017) or 15.0 (2019) for the Information Store.
  - Microsoft SQL Server 2014 Standard or Enterprise for the Analysis Repository.
  - Oracle Database 12c Standard Edition Release 1 and Enterprise Edition or Oracle Database 12c Standard Edition Release 2 and Enterprise Edition for the Analysis Repository.
- An HTTP server that supports a reverse proxy.

  An HTTP server is required for the Analysis Repository.

  If you decide to use an HTTP server in an Opal deployment, the HTTP server that you use must be configured to handle WebSocket requests.

  The deployment toolkit can automatically configure an IBM HTTP Server instance on the server where the i2 Analyze application is deployed to act as a reverse proxy. To support this approach, you must install IBM HTTP Server 9.0.0.7.

For more information about the system requirements and prerequisites, see Release Material.

Installing prerequisite software

When you install the prerequisite software, you must do so according to the following specifications.

Database management system

IBM Db2

Location

If you are creating a production deployment, you can install Db2 in any location. When you install Db2, record the location of the installation directory because you must specify this location in the deployment toolkit before you can deploy i2 Analyze.

If you are creating an example deployment, install Db2 in the following location:

- For Windows: `C:\Program Files\IBM\SQLLIB`
- For Linux®: `/opt/ibm/db2/`*`Db2_version`*

Features and language

In all deployments, you must ensure that the following features are installed:

- Spatial Extender server support
- Spatial Extender client

Note: Do not install Tivoli® System Automation for Multiplatforms.

When you deploy i2 Analyze with the Information Store, you must install Db2 with the product interface language set to English only. Additionally, if you install Db2 on Red Hat Enterprise Linux, you must use an English version of Red Hat Enterprise Linux. For more information, see Changing the Db2 interface language (Linux and UNIX) and Changing the Db2 product interface language (Windows).

For Linux deployments, if you are deploying with a schema that contains non-English characters, ensure that the operating system's LANG environment variable is set to the locale of the non-English characters.

You can deploy i2 Analyze with a Db2 instance only where the DB2_WORKLOAD environment variable is not set. If you have an existing Db2 instance where DB2_WORKLOAD is set, you must create a Db2 instance where the value is not set and deploy i2 Analyze with it. For more information about the DB2_WORKLOAD environment variable, see System environment.

Users

On Windows, Db2 creates a Windows user account (db2admin), and two Windows groups (DB2ADMNS, DB2USERS). To work successfully with Db2, ensure that your Windows user account is a member of the DB2ADMNS Windows group.

On Linux, Db2 creates an Administration Server user (dasusr1) and group (dasadm1), an instance-owning user (db2inst1) and group (db2iadm1), and a fenced user (db2fenc1) and group (db2fadm1). To work successfully with Db2, ensure that the user that runs the deployment script is a member of the dasadm1 and db2iadm1 groups.

Make a note of any user names and passwords that are specified during the installation process.

Post-install

After you install Db2 for the Information Store, you must enable the administrative task scheduler and set the code page on the Db2 installation:

1. On the command line, navigate to the SQLLIB\bin directory of your Db2 installation. On Linux, navigate to the db2inst1/sqllib/bin directory.
2. To enable the administrative task scheduler, run the following command:

```
db2set DB2_ATS_ENABLE=YES
```

3. To set the code page for UTF-8 encoding, run the following command:

```
db2set DB2CODEPAGE=1208
```

For more information about installing Db2, see Installing Db2 database servers.

Remote Db2 database storage

If you plan to deploy i2 Analyze with remote database storage, you must install Db2 on your database server, and Db2 or IBM Data Server Client on the application server. Install Db2 according to the previous instructions; if you are using IBM Data Server Client, also ensure that Spatial Extender client support is installed. For more information about IBM Data Server Client, see Installing IBM Data Server drivers and clients.

The instance of Db2 or IBM Data Server Client on the application server must be the same version level as the instance of Db2 on the database server. For example, if the instance of Db2 on your database server is version 11.1, the instance of Db2 or IBM Data Server Client on the application server must also be version 11.1.

Microsoft SQL Server version 14.0 (2017) or 15.0 (2019)

Location

If you are creating a production deployment, you can install SQL Server in any location. When you install SQL Server, record the location of the installation directory because you must specify this location in the deployment toolkit before you can deploy i2 Analyze.

If you are creating an example deployment, install SQL Server in the default location:

- For Windows: `C:\Program Files\Microsoft SQL Server`
- For Linux: `/opt/mssql`. Install the SQL Server tools in the default path: `/opt/mssql-tools`

Features

In all deployments, you must ensure that the following features are installed or enabled:

- Database Engine Services
- SQL Server Authentication
- TCP/IP protocol

In all deployments, you must install the ODBC Driver for SQL Server and **`sqlcmd`** utility on your database server.

On Windows:

- Microsoft ODBC Driver 17 for SQL Server, Microsoft ODBC Driver 17 for SQL Server.
- The **`sqlcmd`** utility https://docs.microsoft.com/en-us/sql/tools/sqlcmd-utility?view=sql-server-2017#download-the-latest-version-of-sqlcmd-utility.

On Linux:

- Microsoft ODBC Driver 17 for SQL Server, Microsoft ODBC Driver 17 for SQL Server.
- The SQL Server command-line tools, https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-setup-tools?view=sql-server-2017

You can also install Microsoft SQL Server Management Studio to administer your SQL Server installation. If you are using SQL Server on Linux, you can install SQL Server Management Studio on a Windows machine and connect to your SQL Server installation.

To create an example deployment on Windows, the instance name that you use must be *MSSQLSERVER*. Regardless of your operating system, the port number must be 1433.

Users

You must have an SQL Server Authentication Login that has the following permissions:

- Server Roles:
  - *dbcreator*
  - *bulkadmin*, to ingest the example data. The *bulkadmin* role is not supported on Linux.
- User mappings for the *msdb* database:
  - *SQLAgentUserRole*
  - *db_datareader*

Post-install

- Ensure that the *SQL Server Agent* service is running.
- On Windows, if you want to use the instance name to connect to SQL Server ensure that the *SQL Server Browser* service is running.

For more information about installing SQL Server on Windows, see SQL Server installation and on Linux, see SQL Server on Linux.

Remote SQL Server database storage

If you plan to deploy i2 Analyze with remote database storage, you must install SQL Server on the database server, and SQL Server or Microsoft Command Line Utilities 17 for SQL Server on the application server. You can install SQL Server and the Command Line Utilities according to the previous instructions.

Microsoft SQL Server 2014

The Analysis Repository can be deployed into a Microsoft SQL Server database. If you choose to use the Microsoft SQL Server database management system, you can use all of the default settings, except that you must enable SQL Server Authentication and the TCP/IP protocol.

Oracle Database 12c Standard Edition Release 1 and Enterprise Edition or Oracle Database 12c Standard Edition Release 2 and Enterprise Edition

The Analysis Repository can be deployed into an Oracle database. If you choose to use the Oracle database management system, you can use all of the default settings, except that you must set the character set to Unicode (AL32UT8). You must also clear the **Create as a container database** check box.

Important: Oracle database creation is not currently handled by the deployment scripts, so you must create the database manually, and the database must be configured to use a unique user. To work with the example values given in the deployment guide, you must create the database on the server that hosts the i2 Analyze application, and it must be of type `WriteStore`.

Note: In all scenarios, the user that you use to run the deployment scripts must have permission to create and modify the database.

IBM HTTP Server and Web Server Plug-ins for WebSphere Application Server

An i2 Analyze deployment might require a reverse proxy server that routes client requests to the appropriate destination.

The deployment toolkit can automatically configure an IBM HTTP Server instance to act as a reverse proxy. To support this approach, install IBM HTTP Server as follows:

Location

Install IBM HTTP Server from an archive file in the following location:

- For Windows: `C:\IBM\HTTPServer`
- For Linux: `/opt/IBM/HTTPServer`

For more information about installing, see Installing IBM HTTP Server.

The Web Server Plug-ins for WebSphere Application Server are included in the archive installer.

Post-install

- The `bin/ikeyman` file and the `bin/gskcmd` file use the Java in your provided `$JAVA_HOME` environment variable. After you deploy i2 Analyze, you can set your `$JAVA_HOME` environment variable to the directory where the deployment toolkit installs Java.
- Ensure that Microsoft Internet Information Server is either inactive or not present on the i2 Analyze server.
- On Linux, the user that you use to run the deployment scripts must have write permissions on the `/opt/IBM/HTTPServer/conf/httpd.conf` file.

Installing IBM i2 Analyst's Notebook Premium

You can install IBM i2 Analyst's Notebook Premium by using an Installation Manager. Ensure that the system that you are installing Analyst's Notebook Premium on matches the system requirements.

Before you begin

For more information about the system requirements, see Software Product Compatibility Reports.

Procedure

1. Extract the product files from your downloaded distribution.
2. Using Windows Explorer, browse to the root of the distribution and run **`setup.exe`**.
3. Follow the prompts. You are asked to choose the setup type:

Opal Connector

Install Analyst's Notebook Premium with the Opal Connector to connect to a deployment that is configured to use the i2 Analyze Opal services.

Onyx Connector

Install Analyst's Notebook Premium with the Onyx Connector to connect to an Analysis repository and any external data sources that use Data Access on Demand.

Local Analysis Repository

Install Analyst's Notebook Premium with a preconfigured Local Analysis Repository.

Custom

You can select which parts of Analyst's Notebook Premium you require. In addition to selecting a combination of the connectors for the repositories, you can also choose to install extra components:

IBM i2 Analyze Schema Designer

To design and modify the schema that is used by i2 Analyze, install IBM i2 Analyze Schema Designer.

ANB Online Help and Example Materials

If you are planning to install Analyst's Notebook Premium into an environment with limited space, you can choose to exclude the embedded help files and example materials.

Esri maps

If you would like to use the embedded Esri maps features, you can select to install these.

Note: This option is not the same as the separate IBM i2 Analyst's Notebook Connector for Esri that is used by the Onyx connector to search for geographical information.

Installing IBM i2 Analyst's Notebook Connector for Esri

You can install IBM i2 Analyst's Notebook Connector for Esri using an Installation Manager. Ensure that the system that you are installing Analyst's Notebook Connector for Esri on matches the system requirements.

Before you begin

For details of the system requirements, see Software Product Compatibility Reports.

Procedure

1. Extract the product files from your downloaded distribution.

2. Using Windows Explorer, browse to the root of the distribution and run **setup.exe**. The Installation Manager opens.
3. Click **Install** in the left menu to start the installation steps.

Installing IBM i2 iBase

You can install IBM i2 iBase using an Installation Manager. Ensure that the system that you are installing iBase on matches the system requirements.

Before you begin

For details of the system requirements, see Software Product Compatibility Reports.

About this task

IBM i2 iBase Geographic Information Systems Interfaces are installed with iBase 8.9. If you intend to use a mapping application with iBase, then in most cases you must install the mapping application first.

- Except for blue 8 xd, you must install the mapping application before you install iBase. For example, if you install the Microsoft MapPoint interface before you install the MapPoint application, you will see the "`Component not correctly registered`" warning after you complete the interface installation.
- For blue 8 world, you must install the interface while you are logged on as the user who runs the interface. This user must be a local administrator.

Procedure

1. Extract the product files from your downloaded distribution.
2. Using Windows Explorer, browse to the root of the distribution and run **setup.exe**.
3. Follow the prompts. You will be asked for the setup type:

Typical	Installs iBase User along with its documentation
Complete	Installs: • iBase User and iBase Designer • All tools apart from the iBase Index Service Configuration tool and iBase Database Replication. • All of the documentation, including the Administration Center.
Custom	You can select which parts of iBase you require.

4. Follow the prompts to complete the installation.

   Note: If you installed the Coordinate Extensions option, as part of a custom installation, then the following message might display when you first start iBase:

   ```
   An application plug-in failed to load: i2 iBase Bulk Coordinate Converter.
   ```

   This message indicates that a required environment variable does not exist. To resolve this problem, either log off from Windows, or restart the computer.

The i2 Analyze deployment toolkit

The deployment toolkit contains scripts and components that you need to deploy and maintain i2 Analyze. Configuring components within the toolkit allows you to consistently deploy to the same locations with the same settings.

Directories

The deployment toolkit contains files in several directories. On most occasions, you need to interact with only three of the directories:

- The `examples` directory includes the base configurations that you can use to create example deployments, example data, and an example i2 Connect connector.
- The `configuration` directory contains files that you must update with information specific to your deployment.

  Note: When the deployment toolkit is first installed, this directory does not exist. Use one of the base configurations to create this directory.
- The `scripts` directory contains the `setup` script that you use to deploy and configure i2 Analyze.

The `setup` script

The `setup` script completes specific actions that are called tasks. For a list of available tasks and other information, refer to "Deployment toolkit tasks" on page 41.

Alternatively, on the i2 Analyze server open a command prompt and navigate to `toolkit\scripts` and run one of the following commands:

`setup -h`

The `-h` argument displays the usage, common tasks, and examples of use for the `setup` script.

`setup -a`

The `-a` argument displays the same content as when you use `-h`, and a list of additional tasks.

`setup -dh`

The `-dh` argument displays the help information for data access points.

Note: On Linux, whenever you run the `setup` script you must prefix the script with `./`. For example, `./setup -t deploy`.

The base configurations

In the IBM i2 Analyze deployment toolkit, several base configurations are available. To reduce the amount of configuration that you need to complete, you can start from the configuration that is closest to the requirements for your final deployment pattern.

The i2 Analyze deployment toolkit includes the following base configurations:

information-store-opal

The `information-store-opal` base configuration contains settings for a deployment that includes:

- An Information Store that is accessed by using the i2 Analyze Opal services

daod-opal

The `daod-opal` base configuration contains settings for a deployment that includes:

- i2 Analyze with the i2 Connect gateway only, which can provide access to other data stores through connectors that you create

information-store-daod-opal

The `information-store-daod-opal` base configuration contains settings for a deployment that includes:

- An Information Store that is accessed by using the i2 Analyze Opal services
- i2 Analyze with the i2 Connect gateway, which can provide access to other data stores through connectors that you create

analysis-repository

The `analysis-repository` base configuration contains settings for a deployment that includes:

- An Analysis Repository that is accessed by using the i2 Analyze Onyx services

ar-information-store-opal

The `ar-information-store-opal` base configuration contains settings for a deployment that includes:

- An Information Store that is accessed by using the i2 Analyze Opal services
- An Analysis Repository that is accessed by using the i2 Analyze Onyx services

It is possible to extend your deployment after you select and deploy by adding the appropriate data stores and configuration files.

Example schemes

The i2 Analyze deployment toolkit includes example schema files that you can use as a starting point for your own schema.

Example schemas

The deployment toolkit includes four pairs of example i2 Analyze schemas and associated charting schemes in the `toolkit\configuration\examples\schemas` directory:

Law Enforcement

The law enforcement schema deals with criminal activity. It contains entity and link types that are designed to track connections within criminal networks.

Commercial Insurance

The commercial insurance schema deals with fraud in a commercial setting. It contains entity and link types that are designed to track financial transactions such as credit card payments and insurance claims.

Military

The military schema helps with military intelligence tracking. It contains entity and link types that target military operations.

Signals Intelligence

The signals intelligence schema focuses particularly on the cellphones and cell towers that are involved in mobile telecommunications, and on the calls that take place between them.

Example security schema

The example security schema file specifies a number of security dimensions and dimension values and five security groups for users.

The deployment toolkit includes an example security schema file in the `toolkit\configuration \examples\security-schema` directory.

Deployment toolkit tasks

The following toolkit options and tasks are available to use with the `setup` script in the deployment toolkit.

Usage: setup [-h] [-dh] [-a] [-s SERVER] [-w WAR | -id IDENTIFIERS | -hn HOST | --all] [--force] -t TASK [--stacktrace]

Argument name	Argument Description
`-h, --help`	Shows this help message, and exits
`-a, --additional`	Shows additional arguments and tasks, and exits
`-dh, --dahelp`	Shows Data Access specific tasks, and exits
`-t, --task <task>`	Specifies the task to perform
`-s, --server <server>`	Specifies the server profile to manage
`-w, --war <war>`	Specifies the war name
`-co, --collection <collection>`	Specifies the solr collection ID
`-id, --id <ids>`	Specifies the ids of the components to manage
`-hn, --hostname <hostname>`	Restrict the task to operate only on components with the specified hostname attribute
`-l, --locale <locale>`	Specify the language code of the schema files to use in the example deployment
`--all`	Apply the operation to all applicable components
`-sn, --schemaName <Schema>`	Specify the name of the schema file to use in the example deployment
`-e, --exampleData <Example Data>`	Specify the name of the directory containing the example data to ingest
`-f, --force`	Suppress the warning prompt that is associated with tasks that might result in data loss
`--scripts`	Generates scripts
`-st, --stacktrace`	Print the stacktrace if an error occurs

The following installation and deployment tasks are available:

Task name	Task Description
`installLiberty`	Installs Liberty from the Liberty binaries in the toolkit, to the directory specified in environment.properties.
`installZookeeper`	Installs ZooKeeper from the ZooKeeper binaries in the toolkit, to the directory specified in environment.properties.

Task name	Task Description
`installSolr`	Installs Solr from the Solr binaries in the toolkit, to the directory specified in environment.properties.
`deployExample`	Deploys i2 Analyze with default files and settings.
`ingestExampleData`	Ingests entity and link record examples into the Information Store.
`deploy`	Creates the databases, creates the application profile, and deploys i2 Analyze.
`deployLiberty`	Deploys the i2 Analyze application into Liberty.
`start`	Starts all i2 Analyze services on the current server.
`stop`	Stops all i2 Analyze services on the current server.
`restart`	Restarts all i2 Analyze services on the current server.
`configSummary`	Summarizes the configuration of the toolkit.

Examples of use:

- `setup -t deployExample`
- `setup -t ingestExampleData`
- `setup -t deploy`
- `setup -t start`
- `setup -t configSummary`

The following upgrade tasks are available:

Task name	Task Description
`upgrade`	Upgrades the configuration files, and upgrades i2 Analyze.
`upgradeConfiguration`	Upgrades the configuration files for an Information Store deployment.
`upgradeZookeeper`	Upgrades ZooKeeper and ZooKeeper configuration files to the version required by the toolkit.
`upgradeSolr`	Upgrades Solr and Solr configuration files to the version required by the toolkit.
`upgradeDatabases`	Upgrades the Information Store database and clears the search index if required.
`upgradeSolrCollections`	Upgrades ZooKeeper and Solr, and creates a new collection if required.
`upgradeLiberty`	Upgrades Liberty to the version required by the toolkit.

Examples of use:

- `setup -t upgrade`
- `setup -t upgradeConfiguration`
- `setup -t upgradeSolr -hn "example.solr.hostname`

The "upgradeZookeeper", "upgradeSolr", "upgradeDatabases", and "upgradeSolrCollections" tasks support an optional -hn argument that restricts their effect to a single host.

The following administration tasks are available:

Task name	Task Description
`replayFromTimestamp`	Starts Liberty in a mode that replays all events since the time specified by the 'datetime.to.replay.from' property in environment-advanced.properties.
`indexSyncCheck`	Checks whether all Solr indexes are in sync with the Information Store database.
`configureHttpServer`	Sets up the reverse proxy configuration for IBM HTTP Server.
`migrateFileRegistry`	Migrate a file based user registry from a full WebSphere Application Server profile. The was.home.dir and wlp.home.dir properties must be set in environment.properties.
`enableLibertyAdminCenter`	Enable the Liberty Admin Center.
`disableLibertyAdminCenter`	Disable the Liberty Admin Center.
`generateDefaults`	Configures the environment with default property values.
`ensureBasicUserRegistry`	Configures the application for basic user registry authentication.
`ensureExampleUserRegistry`	Populates the user registry with an example user and user groups that map to the example security schema.
`ensureDefaultUserProfileProvider`	Configures the application with the default user profile provider.
`addInformationStore`	Generates a fragment for the Information Store, and updates topology.xml.
`addI2Connect`	Updates topology.xml for i2 Connect.
`updateConnectorsConfiguration`	Updates the i2 Analyze server with the connection details of the connectors defined in topology.xml. In a deployment without the Information Store, the i2 Analyze schema and charting schemes are also updated.
`updateSchema`	Updates i2 Analyze to conform to the schema file referenced in the configuration.
`updateSecuritySchema`	Updates i2 Analyze to use the security schema file referenced in the configuration.
`updateLiveConfiguration`	Updates the i2 Analyze server with the latest version of the files in the configuration/live directory from the deployment toolkit.

Task name	Task Description
createDatabaseStorage	Creates the database storage*
createDatabases	Creates the database storage and tables*
modifyInformationStoreDatabase	Runs the informationStoreModifications.sql script on the Information Store database.
dropTables	Drops all of the tables from the database but leaves the database intact*
dropDatabases	Drops the entire database and de-allocates storage*
emptyInformationStore	Empties the Information Store of data, apart from metadata.
addInformationStoreIngestionSource	Adds or replaces information about an ingestion source to the Information Store.
createInformationStoreStagingTable	Creates an Information Store staging table for a specific entity or link type.
ingestInformationStoreRecords	Ingests records into the Information Store.
deleteProvenance	Deletes (entity/link) provenance from the Information Store.
previewDeleteProvenance	Previews deleting (entity/link) provenance from the Information Store.
syncInformationStoreCorrelation	Synchronizes data in the Information Store after a correlation operation failed during ingestion.
enableMergedPropertyValues	Creates the database views used to define the property values of merged i2 Analyze records.
disableMergedPropertyValues	Removes the database views used to define the property values of merged i2 Analyze records.
duplicateProvenanceCheck	Checks the Information Store for duplicated origin identifiers. Any provenance that has a duplicated origin identifier is added to a staging table.
duplicateProvenanceDelete	Deletes (entity/link) provenance from the Information Store that has duplicated origin identifiers. The provenance to delete is identified in the staging tables created by the duplicateProvenanceCheck task.
deleteOrphanedDatabaseObjects	Deletes (entity/link) database objects that are not associated with an i2 Analyze record from the Information Store.
createEtlToolkit	Creates a DataStage ETL toolkit that contains the files DataStage requires to run pipeline jobs.
generateInformationStore IndexCreationScripts	Generates Information Store 'create indexes' DDL scripts for the specified item type

Task name	Task Description
`generateInformationStore IndexDropScripts`	Generates Information Store 'drop indexes' DDL scripts for the specified item type
`clearData`	Clears the search index and all the data in the database.
`clearSearchIndex`	Clears the search index.
`clearLTPAkeys`	Clears the LTPA keys.

The "clearData" and "clearSearchIndex" tasks support an optional -co argument that restricts their effect to a single Solr collection.

Example of use:

- `setup -t clearData -co "solr.collection.id`

* The exact behavior of these tasks may change depending on the chosen database engine.

Tasks for DB2 only:

Task name	Task Description
`catalogRemoteDB2Nodes`	Adds a remote database server entry to the DB2 node directory for each remote DB2 database that is defined in topology.xml
`uncatalogRemoteDB2Nodes`	Removes the remote database server entry in the DB2 node directory for each remote DB2 database that is defined in topology.xml.
`recatalogRemoteDB2Nodes`	Removes, then re-adds the remote database entry in the DB2 node directory for each remote DB2 database that is defined in topology.xml.
`listDB2NodeDirectory`	Lists the contents of the DB2 node directory.
`catalogDB2Databases`	Adds an entry to the system database directory for each DB2 database that is defined in topology.xml. If the database is remote from the i2 Analyze server, the database is cataloged at the node specified for that database in topology.xml.
`uncatalogDB2Databases`	Removes the entry in the system database directory for each DB2 database that is defined in topology.xml.
`recatalogDB2Databases`	Removes, then re-adds the entry in the system database directory for each DB2 database that is defined in topology.xml.
`listDB2SystemDatabaseDirectory`	Lists the contents of the local DB2 system database directory.

Examples of use:

- `setup -t configureHttpServer`
- `setup -t replayFromTimestamp`

In addition to the start, stop and restart tasks, the following tasks are available:

Task name	Task Description
`startLiberty`	Starts Liberty.
`stopLiberty`	Stops Liberty.
`restartLiberty`	Restarts Liberty.
`startSolrAndZk`	Starts the Solr nodes and ZooKeeper hosts.
`stopSolrAndZk`	Stops the Solr nodes and ZooKeeper hosts.
`restartSolrAndZk`	Restarts the Solr nodes and ZooKeeper hosts.
`startSolrNodes`	Starts the Solr nodes.
`stopSolrNodes`	Stops the Solr nodes.
`restartSolrNodes`	Restarts the Solr nodes.
`startZkHosts`	Starts ZooKeeper hosts.
`stopZkHosts`	Stops ZooKeeper hosts.
`restartZkHosts`	Restarts ZooKeeper hosts.

Examples of use:

- `setup -t startSolrAndZk`
- `setup -t stopZkHosts -id "1,3"`
- `setup -t restartSolrNodes -id node1`
- `setup -t startSolrNodes -hn "example.solr.hostname"`

The "SolrNodes" and "ZkHosts" tasks support an optional -id argument.

The comma-separated list of identifiers that you specify restricts the task to the nodes and hosts with matching identifiers in the topology.

The "SolrNodes" and "ZkHosts" tasks support an optional -hn argument that restricts their effect to a single host.

The following Solr and ZooKeeper tasks are available:

Task name	Task Description
`createSolrNodes`	Creates the Solr nodes that are defined in topology.xml. If the nodes already exist, their configuration is updated.
`createZkHosts`	Creates the ZooKeeper hosts that are defined in topology.xml. If the hosts already exist, their configuration is updated.
`getZkStatus`	Reports the status of the ZooKeeper hosts that are defined in topology.xml
`createAndUploadSolrConfig`	Creates and uploads the Solr configuration to the ZooKeeper hosts.
`createSolrCollections`	Creates the Solr collections that are defined in topology.xml.

Task name	Task Description
deleteSolrCollections	Deletes the Solr collections that are defined in topology.xml.

Examples of use:

- `setup -t createSolrNodes -hn "example.solr.hostname"`
- `setup -t createSolrCollections -co "solr.collection.id"`

These tasks support an optional -hn argument that restricts their effect to a single host.

The "createAndUploadSolrConfig", "createSolrCollections", and "deleteSolrCollections" tasks support an optional -co argument that restricts their effect to a single Solr collection.

The following tasks manage main indexes and match indexes. ZooKeeper and the application server must be running for these commands to succeed:

Task name	Task Description
switchStandbyMatchIndexToLive	Sets the standby match index to live if it is READY, and sets the previous live match index to standby.
clearStandbyMatchIndex	Deletes the contents of the standby match index and sets its state to DISABLED.
updateMatchRules	Uploads the system match rules and applies them to the standby match index, which starts BUILDING again.
switchStandbyMainIndexToLive	Sets the standby main index to live if it is READY, and sets the previous live main index to standby.
clearStandbyMainIndex	Deletes the contents of the standby main index and sets its state to DISABLED.
rebuildStandbyMainIndex	Instructs the standby main index to start BUILDING again, if it exists.
switchStandbyChartIndexToLive	Sets the standby chart index to live if it is READY, and sets the previous live chart index to standby.
clearStandbyChartIndex	Deletes the contents of the standby chart index and sets its state to DISABLED.
rebuildStandbyChartIndex	Instructs the standby chart index to start BUILDING again, if it exists.
pauseIndexPopulation	Pauses index population for a specified index or all indexes. Population is paused until the Liberty server restarts.
resumeIndexPopulation	Resumes index population for a specified index or all indexes where population is currently paused.

Data access options:

Argument name	Argument Description
-x, --schema <schema>	Specifies the full path including the file name of the schema file used to generate the mapping

Argument name	Argument Description
`-o, --mappingJAR <mappingJAR>`	Specifies the full path including the file name to the mapping JAR file to be created
`-z, --xsdPath <xsdPath>`	Specifies the full path to a directory where the generated XSD files will be stored
`-dn, --datasourceName <datasourceName>`	Specifies the name of the data source to be created or modified
`-sc, --singleCardFormat`	Generates files that validate using a single card format instead of the standard multiple card format
`-s, --server <server>`	Specifies the server profile to manage
`-c, --configFile <configFile>`	Specifies the full path including the file name of the Connector Creator configuration file
`-lp, --logProperties <logProperties>`	Specifies the full path including the file name of the logging properties file

The following tasks are available:

Task name	Task Description
`generateMappingJar`	Creates mapping classes and XSD files for the specified schema
`addDaodDataSource`	Generates a fragment for data access on-demand, and updates topology.xml
`addDelpsDataSource`	Generates a fragment for data load ELP stage, and updates topology.xml
`addConnectorCreator`	Generates a fragment for Connector Creator, and updates topology.xml
`createI2AnalyzeSchemaFromIBase`	Creates an i2 Analyze schema that reflects the entity, link, and property types in an existing iBase database
`replaceIBaseSchemaFromI2Analyze`	Replaces the schema of a new iBase database with one that is consistent with an existing i2 Analyze deployment
`addIBaseConnector`	Generates a fragment for an iBase connector, and updates topology.xml
`addIBaseDatasource`	Adds or updates an iBase data source in the deployment configuration
`uploadCcConfig`	Uploads a configuration file to Connector Creator
`downloadCcConfig`	Downloads the configuration file from Connector Creator

Examples of use:

- `setup -t generateMappingJar -x MySchema -o MyMappingJAR [-sc]`
- `setup -t addDaodDataSource -dn MyDataSource`
- `setup -t addDelpsDataSource -dn MyDataSource`

- `setup -t addConnectorCreator`
- `setup -t uploadCcConfig -c "C:/ccConfigIn.xml"`
- `setup -t downloadCcConfig -c "C:/ccConfigOut.xml"`

i2 Analyze deployment settings

Some settings in the deployment toolkit require values that are specific to the environment that you are deploying in. Although these settings can sometimes use the default values, at different times during the production process you might need to specify values that meet your environment and deployment requirements.

Specifying the deployment credentials

To allow the deployment toolkit to update the database, Lightweight Third-Party Authentication (LTPA) keys, and Solr search platform components, you must provide user names and passwords. The user names and passwords that you provide allow the system to set up and administer components of i2 Analyze, and are not used to access i2 Analyze.

About this task

Database

For each database that is identified in `topology.xml`, you must specify a user name and a password in the `credentials.properties` file. The `setup` script uses this information to authenticate with the database.

Note: The user that you specify must have privileges to create and populate databases in the database management system.

The database credentials are stored in the following format:

```
db.identifier.user-name=user name
db.identifier.password=password
```

The name of each database credential has three parts, two of which are variable:

Part	Description
identifier	The database identifier, for example `infostore`.
user-name or **password**	Indicates whether the value is a user name or a password.

For example:

```
db.infostore.user-name=admin
db.infostore.password=password
```

The db.**identifier**.truststore.**password** credential is used only when you configure the connection between the database and Liberty to use SSL. If you are not using SSL to secure this connection, you do not need to specify a value for the db.**identifier**.truststore.**password** credential. For more information about configuring SSL, see Configure Secure Sockets Layer with i2 Analyze.

LTPA keys

You must provide a value for the `ltpakeys.password` property. This value is used by the system to encrypt LTPA tokens.

- For a stand-alone deployment of i2 Analyze, you can specify any value as the password.
- For a deployment of i2 Analyze that uses LTPA tokens to authenticate with other systems, you must specify the same password that those systems use.

Solr search platform

The Solr search platform is used to search data in the Information Store. You must provide values for the `solr.user-name` and `solr.password` properties. Any Solr indexes are created when i2 Analyze is first deployed, and the values that you provide here become the Solr user name and password.

When you deploy i2 Analyze, the passwords in the `credentials.properties` file are encoded.

Procedure

1. Using a text editor, open the `toolkit\configuration\environment\credentials.properties` file.
2. Specify the database, LTPA, and Solr credentials for your deployment:
   - the database user names and passwords
   - the LTPA keys password
   - the Solr user name and password
3. Save and close the `credentials.properties` file.

Specifying the JDBC driver

The application server requires a JDBC driver to enable communication with the database. The JDBC driver that you provide to the deployment toolkit depends on the database management system to use with the deployment.

Procedure

1. Locate the JDBC driver for your database management system.
   - If you are using Db2, locate the `IBM\SQLLIB\java\db2jcc4.jar` file.
   - If you are using SQL Server with the Information Store, download the Microsoft JDBC Driver 7.4 for SQL Server archive from https://www.microsoft.com/en-us/download/details.aspx?id=58505. Extract the contents of the download, and locate the `sqljdbc_7.4\enu\mssql-jdbc-7.4.1.jre8.jar` file.
   - If you are using SQL Server with the Analysis Repository, download the Microsoft JDBC Driver 6.0 for SQL Server from https://www.microsoft.com/en-us/download/details.aspx?id=11774. Extract the contents of the download, and locate the `sqljdbc_6.0\enu\jre8\sqljdbc42.jar` file.
   - If you are using Oracle 12c Release 1, locate the `jdbc\lib\ojdbc7.jar` file.
   - If you are using Oracle 12c Release 2, locate the `jdbc\lib\ojdbc8.jar` file.
2. Copy the relevant driver file for your database management system to the `toolkit\configuration\environment\common\jdbc-drivers` directory.
3. If you are using Oracle, or you want to use a JDBC driver `.jar` file that has a custom name, you must add the `jdbc-driver` attribute to the `<database>` element in your `topology.xml` file.

For example:

```
<database dialect="oracle" xa="false" instance-name=""
    database-type="WriteStore" id="write1" host-name=""
    port-number="1521" jdbc-driver="jdbc_file_name.jar" />
```

jdbc_file_name is the name of the `.jar` file to locate in the `toolkit\configuration\environment\common\jdbc-drivers` directory to use for the deployment.

If you are using Oracle 12c Release 2, set the value as follows: `jdbc-driver="ojdbc8.jar"`.

Specifying remote database storage

Configuring remote IBM Db2 database storage

You can deploy i2 Analyze with Db2 database storage that is remote from the i2 Analyze server. When i2 Analyze is configured to deploy to a remote instance of Db2, the databases are created and updated remotely without the i2 Analyze deployment toolkit present on the server that hosts the Db2 instance.

Before you begin

To deploy i2 Analyze that uses remote Db2 database storage, you must install Db2 on your database server, and Db2 or IBM Data Server Client on the application server. Both instances of Db2 must be installed according to the specifications defined in the i2 Analyze software prerequisites. For more information about installing the prerequisites, see Software prerequisites.

About this task

To deploy with remote storage, the i2 Analyze configuration must contain certain information about your Db2 instances. After you update the configuration, the specified databases can be created and updated by the deployment toolkit on the remote server.

Note: You can complete the steps that are performed by the **catalogRemoteDB2Nodes** task manually. For example, if you are deploying a system that uses Transport Layer Security (TLS).

To catalog the remote nodes manually, you can run the **CATALOG TCPIP NODE** instead of using the **setup -t catalogRemoteDB2Nodes** command. For more information about the command, see **CATALOG TCPIP/TCPIP4/TCPIP6 NODE** command.

The following table shows how the **CATALOG** command parameters map to the values in the `topology.xml` file:

CATALOG TCPIP NODE command parameters	**<database> element attributes**
TCPIP NODE *nodename*	`node-name`
REMOTE *hostname*	`host-name`
SERVER *port number*	`port-number`
REMOTE_INSTANCE *instance-name*	`instance-name`

Procedure

1. Edit the `configuration\environment\topology.xml` file to specify your remote Db2 databases:

a) Update the `host-name` and `port-number` attribute values of the `<database>` element to match the values of your remote Db2 instance.

   Note: The value of the `instance-name` attribute must match the instance name of the local Db2 instance.

   You can use the **db2level** command to get the name of your local instance. For more information, see db2level - Show Db2 service level command.

b) Add the `node-name` attribute to the `<database>` element of the databases to be hosted remotely.
   For example:

```
<database dialect="db2" xa="false" instance-name="DB2"
    database-name="WriteSto" database-type="WriteStore" id="write1"
    host-name="hostname" port-number="50000" node-name="node1" />
```

   Where the value for `node-name` is the name of the node to create in the Db2 node directory. The value of the `node-name` attribute must start with a letter, and have fewer than 8 characters. For more information about naming in Db2, see Naming conventions.

   Note: If the Analysis Repository and Information Store are using the same Db2 instance, they can use the same node.

c) If you are deploying the Information Store database, add the `os-type` attribute to the `<database>` element for the Information Store database. The value of the `os-type` is used to support the search functions for the Information Store.
   For example:

```
<database database-type="InfoStore" dialect="db2" instance-name="DB2"
    database-name="ISTORE" xa="false" id="infostore"
    host-name="hostname" port-number="50000"
    node-name="node1" os-type="WIN" />
```

   Where the value for `os-type` is the operating system of the remote Db2 server.

   Note: The value of the `os-type` attribute must be one of the following values: `AIX`, `UNIX`, or `WIN`.

2. Edit the `configuration\environment\`*`server-name`*`\environment.properties` file, to specify the details of your remote and local instance of Db2.

   Where *server-name* is the name of your application server.

   a) Ensure that the value of the `db.installation.dir` property is set for the local instance of Db2 or Data Server Client on the Liberty server.

      If you are using a non-root installation, set the value for this property to the `sqllib` directory in the installation user's home directory. For example, `/home/db2admin/sqllib`.

   b) Set the value of the `db.database.location.dir` property for the remote instance of Db2 on the database server.

3. Ensure that the users that are specified for your databases in the `configuration\environment\credentials.properties` file are valid for your remote instance of Db2 on the database server.

Results

When i2 Analyze is deployed, the Information Store database is created or updated on the remote database management system. A remote node is created with the name that is specified for the `node-name` attribute, and the database is cataloged against that node.

To check that the remote nodes and databases are cataloged, you can use the `listDB2NodeDirectory` and `listDB2SystemDatabaseDirectory` tasks after you deploy i2 Analyze:

- The `listDB2NodeDirectory` task lists the contents of the Db2 node directory.
- The `listDB2SystemDatabaseDirectory` task lists the contents of the local Db2 system database directory.

Configuring remote SQL Server database storage

You can deploy i2 Analyze with SQL Server database storage that is remote from the i2 Analyze server. When i2 Analyze is configured to deploy to a remote instance SQL Server, the Information Store database can be created and updated remotely without the i2 Analyze deployment toolkit present on the server that hosts the SQL Server instance.

Before you begin

To deploy i2 Analyze that uses remote SQL Server database storage for the Information Store, you must install SQL Server on your database server, and the SQL Server client tools on your application server. SQL Server and the client tools must be installed according to the specifications defined in the i2 Analyze software prerequisites. For more information about installing the prerequisites, see Software prerequisites.

About this task

To deploy with remote storage, the deployment toolkit must contain certain information about your SQL Server installation. After you update the configuration, the specified database can be created and updated on the remote server by the deployment toolkit.

Procedure

1. Edit the `configuration\environment\topology.xml` file to specify the details of your remote database:
   a) Update the `host-name`, and `port-number` or `instance-name`, attribute values of the `<database>` element for the Information Store to match the values for SQL Server on your database server.
   b) Add the `os-type` attribute to the `<database>` element for the Information Store database. The value of the `os-type` is used to support the search functions for the Information Store. For example:

      ```
      <database database-type="InfoStore" dialect="sqlserver" instance-
      name="SQLSERVER"
          database-name="ISTORE" xa="false" id="infostore"
          host-name="remote.hostname" os-type="WIN" />
      ```

      Where the value for `os-type` is the operating system of the remote database server.

      Note: The value of the `os-type` attribute must be one of the following values: `UNIX` or `WIN`.
2. Edit the `configuration\environment\opal-server\environment.properties` file to specify the details of your remote and local installations of SQL Server:

a) Ensure that the value of the `db.installation.dir` property is set for the local installation of SQL Server or the Microsoft Command Line Utilities for SQL Server on the Liberty server.

b) Set the value of the `db.database.location.dir` property for the remote installation of SQL Server on the database server.

3. Ensure that the user that is specified for your database in the `configuration\environment\credentials.properties` file is valid for your remote installation of SQL Server on the database server.

Results

When i2 Analyze is deployed, the Information Store database is created or updated on the remote database management system.

What to do next

If the connection details for the remote database management system changes, you can update the `topology.xml` file and redeploy the system.

Specifying remote Solr and ZooKeeper servers

In a deployment of i2 Analyze, the location of each Solr server and ZooKeeper server is specified in the `topology.xml` file. When you run the commands to deploy i2 Analyze, the Solr nodes and ZooKeeper servers are created on servers that you specify.

About this task

Add the information about your Solr and ZooKeeper servers to the `configuration\environment\topology.xml` file.

Procedure

1. In the `topology.xml` file, add a `<solr-node>` element for each Solr server in your deployment topology.

   For example:

```
<solr-nodes>
  ...
  <solr-node
    memory="2g"
    data-dir="C:/IBM/i2analyze/data/solr"
    host-name="solr_server_host_name"
    id="node2"
    port-number="8984"
  />
</solr-nodes>
```

   Where *solr_server_host_name* is the hostname of a Solr server.

   For more information about the possible values for each attribute, see Solr.

2. In the `topology.xml` file, add a `<zkhost>` element for each ZooKeeper server in your deployment topology.

   For example:

```
<zkhosts>
  ...
  <zkhost
    quorum-port-number="10484"
    leader-port-number="10984"
    data-dir="C:/IBM/i2analyze/data/zookeeper"
    host-name="zookeeper_server_host_name"
    id="2"
    port-number="9984"
  />
</zkhosts>
```

Where *zookeeper_server_host_name* is the hostname of ZooKeeper server.

For more information about the possible values for each attribute, see ZooKeeper.

Specifying the connection URI

If your deployment contains the Opal services and uses an extra proxy server between the clients and i2 Analyze, you must specify the URI that clients use to connect to i2 Analyze. Clients that do not use the specified URI are unable to connect to i2 Analyze.

Before you begin

Your proxy server must be configured so that any security configurations, for example Secure Sockets Layer or client certificate authentication, are passed through to the i2 Analyze server. You must complete this according to the documentation for your proxy server.

About this task

The example deployments use a single HTTP server proxy on the i2 Analyze server that clients connect to. Your system might include an extra proxy server between the clients and the i2 Analyze server. Usually this extra proxy server has a different URI to i2 Analyze. If users try to connect to i2 Analyze using the URI of the proxy server, they are unable to connect to i2 Analyze. To allow users to connect by using the URI of the proxy server, you must specify that URI in the i2 Analyze configuration. After you configure the connection URI for your deployment, users that connect to i2 Analyze must use the URI that you specify.

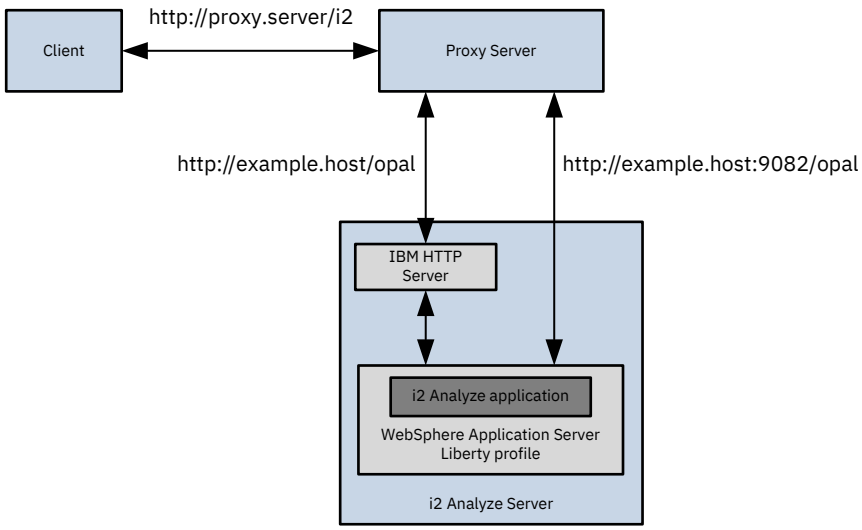
Note: You can specify only one URI that clients can use to connect to your deployment of i2 Analyze.

When clients connect to your proxy server using the connection URI that you specify, you must configure your proxy server to route requests from the client to the URI of the i2 Analyze server. The URI of the i2 Analyze server is defined in the topology file. The URI is determined from the `host-name` attribute of the `<application>` element, and the `context-root` attribute of the `<war>` element. For example, the URI for a deployment with the following topology file is `http://example.host/opal`.

```
<application http-server-host="true" name="opal-server" host-name="example.host">
  <wars>
    <war context-root="opal" name="opal-services-is" i2-data-source-id="infostore"
     target="opal-services-is">
  ...
</application>
```

Note: If your Opal application has the `http-server-host` attribute set to `false`, your proxy server must route requests with the port number of the application. You can find the port number of the application in the `toolkit\configuration\environment\opal-server\port-def.props` file.

The following image shows the URIs that you might use in this example:



Procedure

1. Using a text editor, open the `toolkit\configuration\fragments\opal-services\WEB-INF\classes\DiscoClientSettings.properties` file.
2. Set the value of the `FrontEndURI` property to the URI that can be used to connect to your deployment of i2 Analyze through the proxy server.
   For example, `FrontEndURI=http://proxy.server/i2`.
3. Save and close the file.

What to do next

After you update the connection URI, you can either modify other aspects of the deployment toolkit or redeploy the system to update the deployment with any changes. After you deploy i2 Analyze and start the server, the URI that can be used to access the system is displayed in the console. Ensure that you can access i2 Analyze by using this URI from a client workstation that uses the proxy server.

i2 Analyze deployment commands

The scripts that deploy i2 Analyze depend on the values in the i2 Analyze configuration. The scripts and commands that you run depend on the topology of the deployment that you want to create.

Deploying i2 Analyze

To deploy i2 Analyze in a single-server deployment topology, you can run a script to deploy i2 Analyze. After i2 Analyze is successfully deployed, you can start the system.

Before you begin

You must have an i2 Analyze configuration that is set up for a single-server or i2 Connect gateway only deployment topology. For more information about creating a valid configuration, see “Creating the pre-production environment” on page 21.

Run any toolkit commands from the `toolkit\scripts` directory in the deployment toolkit.

Procedure

1. Deploy i2 Analyze:

```
setup -t deploy
```

2. Start i2 Analyze:

```
setup -t start
```

What to do next

If an error message is displayed, refer to “Troubleshooting the deployment process” on page 60.

After you deploy and start i2 Analyze, return to perform the rest of the instructions for creating a deployment in your current environment:

- “Creating the pre-production environment” on page 21
- “Creating the test environment” on page 24
- “Creating the production environment” on page 26

Deploying i2 Analyze on multiple servers

To deploy i2 Analyze in a multiple-server deployment topology, you must run the commands to install, deploy, and start the components of i2 Analyze on each server.

Before you begin

You must have an i2 Analyze configuration that is set up for a multiple-server physical deployment topology. For more information about creating a valid configuration, see “Creating the pre-production environment” on page 21.

About this task

To deploy i2 Analyze in a multiple server deployment topology, you must provide the configuration to each deployment toolkit. Then, you can run the commands to deploy the components of i2 Analyze on each server. It is important to note which server you must run the command on, and whether you need to specify the hostname of that server.

Run any toolkit commands from the `toolkit\scripts` directory in the deployment toolkit on the specified server in your environment.

Copying the i2 Analyze configuration

The i2 Analyze configuration is required by all servers that host components of i2 Analyze. You do not have to copy the configuration to the database server, if it contains only the Information Store database and no other components of i2 Analyze.

Procedure

Copy the `toolkit\configuration` from the server where you created and populated your i2 Analyze configuration, to the `toolkit` directory of the deployment toolkit on each server in your environment.

Installing components

Install the components of i2 Analyze on the specified servers.

Procedure

1. On the Liberty server, run the following commands:

```
setup -t installLiberty
```

2. On each ZooKeeper server, run the following command:

```
setup -t installZookeeper
```

3. On each Solr server, run the following command:

```
setup -t installSolr
```

Deploying and starting components

Deploy and start the components of i2 Analyze on the specified servers.

Procedure

1. On each ZooKeeper server, create and start any ZooKeeper hosts:

```
setup -t createZkHosts --hostname zookeeper.hostname
setup -t startZkHosts --hostname zookeeper.hostname
```

   Where `zookeeper.hostname` is the hostname of the ZooKeeper server where you are running the command, and matches the value for the `host-name` attribute of a `<zkhost>` element in the `topology.xml` file.

2. On the Liberty server, run the command to upload the Solr configuration to ZooKeeper:

```
setup -t createAndUploadSolrConfig --hostname liberty.hostname
```

   Where `liberty.hostname` is the hostname of the Liberty server where you are running the command, and matches the value for the `host-name` attribute of the `<application>` element in the `topology.xml` file.

3. On each Solr server, create and start any Solr nodes:

```
setup -t createSolrNodes --hostname solr.hostname
setup -t startSolrNodes --hostname solr.hostname
```

Where `solr.hostname` is the hostname of the Solr server where you are running the command, and matches the value for the `host-name` attribute of a `<solr-node>` element in the `topology.xml` file.

On the Liberty server, run the commands to deploy and start a number of the components.

4. Create the Solr collections:

```
setup -t createSolrCollections --hostname liberty.hostname
```

To test that the Solr Collection is created correctly, click **Cloud** in the Solr Web UI, or you can go to http://*solr.hostname*:*port-number*/solr/#/~cloud. Log in with the user name and password for Solr in the `credentials.properties` file.

A horizontal tree with the collection as the root is displayed. Here you can see the breakdown of the shards, nodes, and replicas in any collections.

5. If you are using Db2, catalog the remote node:

```
setup -t catalogRemoteDB2Nodes
```

6. Create the Information Store database:

```
setup -t createDatabases
```

To check that the database is created correctly, connect to the database by using a database management tool.

7. Deploy the i2 Analyze application:

```
setup -t deployLiberty
```

8. If you are using IBM HTTP Server, configure the HTTP Server:

```
setup -t configureHttpServer
```

9. Start the Liberty server:

```
setup -t startLiberty
```

What to do next

After you deploy and start i2 Analyze, return to perform the rest of the instructions for creating a deployment in your current environment:

- “Creating the pre-production environment” on page 21
- “Creating the test environment” on page 24
- “Creating the production environment” on page 26

Troubleshooting the deployment process

i2 Analyze provides mechanisms that assist with deployment before, during, and after you do it.

Deployment toolkit validation messages

The i2 Analyze deployment toolkit validates some of the configuration files that are required for a deployment. By reading the validation messages, you can be certain that required properties have values, that property values have the correct format, and that required files are in the correct locations.

Validation takes place when you run the `setup -t deploy` command. Validation can complete in the following ways:

Success

On success, the validation process displays a brief configuration summary, and the main deployment process continues. For example:

```
============== Configuration Summary (brief) ==============
 + configuration/fragments/common/WEB-INF/classes/ApolloServerSettings
Mandatory.properties:
   - ChartingSchemesResource='law-enforcement-schema-charting-schemes.xml'
   - SchemaResource='law-enforcement-schema.xml'
   - DynamicSecuritySchemaResource='security-schema.xml'
============================================================
```

Warning

When there is a warning, the validation process displays a brief configuration summary and a `WARNINGS` section. The `WARNINGS` section identifies settings that might not be configured correctly, but the deployment process continues. For example:

```
====================== WARNINGS (1) ======================
 + configuration/fragments/common/WEB-INF/classes/ApolloServerSettings
Mandatory.properties:
   - The <SchemaResource> property has not been set.
============================================================
```

Here, the schema is not set, and so the default law enforcement schema is used.

Error

If an error occurs, the validation process displays a longer configuration summary, and an `ERRORS` section. The `ERRORS` section identifies missing values that must be present. The deployment process stops, and you must correct the errors before you attempt to deploy again. For example:

```
====================== ERRORS (1) ========================
 + configuration/environment/opal-server/environment.properties:
   - "db.database.location.dir" has not been set
============================================================
```

Here, the database location directory is not set, so the database cannot be configured.

Deployment progress messages

During the deployment process, i2 Analyze displays detailed messages that provide information about the state and configuration of the system.

The output from the deployment process describes each task that the `setup` command performs during deployment. If a task runs successfully, then only its name appears. For example:

```
:buildApplication
```

There are two reasons why a task might not run, but deployment can still proceed.

UP-TO-DATE

The task was performed earlier, or its output is already present. For example:

```
:installJDBCDrivers UP-TO-DATE
```

SKIPPED

The task is not required for this deployment. For example:

```
:importLTPAKey SKIPPED
```

If an error occurs, deployment stops in a controlled manner. i2 Analyze displays a stack trace that contains the name of the task that failed, and information about the location of the error. For example:

```
:createDatabasesIfNecessary FAILED

FAILURE: Build failed with an exception.

* Where:
Script 'C:\IBM\i2analyze\toolkit\scripts\gradle\database.gradle' line: 173

* What went wrong:
Execution failed for task ':createDatabasesIfNecessary'.
```

The messages are displayed on screen and sent to the log files that are stored in the `toolkit\configuration\logs` directory.

Deployment log files

i2 Analyze produces logging information about deployment tasks and the transactions that take place when the system is operational. After deployment, you can check these logs to help diagnose potential issues.

The following types of logged information are available for you to review:

Deployment logs

Each time that you run the `setup` command, a log file is created in the deployment toolkit. The messages in these logs describe which tasks were called, whether the tasks completed successfully, and the details of any issues that occurred.

These log files contain the same information as the console output. You can find them in the `toolkit\configuration\logs` directory.

Liberty logs

In addition to the information in the i2 Analyze logs, extra information that relates to the application server is also logged in `deploy\wlp\usr\servers\<server>\logs`.

The directory contains the following log files that you might use to help troubleshoot any issues:

`console.log`

The `console.log` file contains basic server status and operation messages, which are also displayed on the console.

`\messages\messages.log`

The `messages.log` file contains messages from Liberty.

`\messages\ffdc\ffdc.log`

The `ffdc.log` file contains Fist Failure Data Capture (FFDC) output from Liberty components, for example the database drivers, that typically include selective memory dumps of diagnostic data.

`\<war-name>\IBM_i2_Alert_Scheduler.log`

The `IBM_i2_Alert_Scheduler.log` contains messages that record when the scheduled operations for saved Visual Queries run.

`\<war-name>\IBM_i2_Analysis_Repository.log`

The `IBM_i2_Analysis_Repository.log` file contains messages that are intercepted by log4j from the i2 Analyze application.

`\<war-name>\IBM_i2_General.log`

The `IBM_i2_General.log` file contains any messages that are intercepted by log4j, but not from the i2 Analyze application.

`\<war-name>\IBM_i2_Solr_Cluster_Status.log`

The `IBM_i2_Solr_Cluster_Status.log` file contains entries for changes to the state of the Solr clusters in your deployment.

`\<war-name>\IBM_i2_Update_Live_Configuration.log`

The `IBM_i2_Update_Live_Configuration.log` file contains messages that are displayed in the console when you run the `updateLiveConfiguration` toolkit task and use the `admin/reload` endpoint.

Solr logs

By default, information that relates to Solr is logged in `deploy\solr\server\logs\<node port>`.

Troubleshooting specific errors

The following sections describe how to react to some more common failures that can occur during the deployment of i2 Analyze.

The results configuration file contains an invalid property type identifier
The following message is displayed if there are any identifiers in your results configuration file that are not present in your i2 Analyze schema when you start i2 Analyze.

```
# [opal-services-is] OneOffStartupServiceInitializer encountered a problem
communicating with the database that it cannot recover from.
# [opal-services-is] Exception during initialization. The application is in an
unusable state.
# [opal-services-is] Errors occurred accessing system resources from
ApplicationLifecycleManager:
# [opal-services-is] The results configuration file contains an invalid property
type identifier: ADD1.
```

To resolve this issue, ensure that all of the identifiers in your results configuration file are present in your i2 Analyze schema. For more information, see Setting up search results filtering.

Default security dimension values provider specifies dimension values that are not in the security schema or **Default security dimension values provider specified no values for the dimensions with these IDs: '[ ]'.**
The security dimension values that are specified in the security schema for the `<DefaultSecurityDimensionValues>` element are incorrect. For more information, see Setting default dimension values.

Unable to determine the DB2 version as you do not have execute permission to db2level
This message is displayed if the JDBC driver is not present or the installation path to Db2 is specified incorrectly when you run the `generateDefaults` toolkit task.

There are two possible solutions to resolve this issue:

- Ensure that you provide the JDBC driver for your deployment. For more information, see “Specifying the JDBC driver” on page 50.
- Ensure that the value for the `db.installation.dir` setting in the `environment.properties` file is correct. For more information, see

Connecting clients

After you deploy and start i2 Analyze, you can connect to your system using one of the supported clients.

Before you begin

- You must have at least one user set up within the application server that has permission to access items in i2 Analyze.
- Ensure that i2 Analyze is started.

Procedure

To use Analyst's Notebook to connect, follow the instructions in Connecting IBM i2 Analyst's Notebook Premium to IBM i2 Analyze.

Opening the web client

If you have deployed the Information Store, you can open the web client from any client computer with access to the i2 Analyze server. Ensure that you can access the web client, and search for data.

Before you begin

The web client is licensed as a part of the IBM i2 Enterprise Insight Analysis Investigate Add On. Ensure that you have the correct agreements in place before accessing the web client.

Procedure

1. Open a web browser, and navigate to `http://`*`host_name`*`/opal` (where *host_name* is the fully qualified domain name or IP address of the i2 Analyze server).

   The web client displays a login dialog.
2. Enter the name and password of a user who is registered in the application server.
3. Search and visualize data to ensure that the application is running.

Opening the Intelligence Portal

If you have deployed the Analysis Repository, you can open the Intelligence Portal from any client computer with access to the HTTP server that hosts the reverse proxy. Ensure that you can access the Intelligence Portal, and create, browse, and search for data.

Procedure

1. Open a web browser, and navigate to `http://`*`host_name`*`/apollo` (where *host_name* is the fully qualified domain name or IP address of the HTTP server).

   i2 Analyze displays a login dialog.
2. Enter the name and password of a user who is registered in the application server.
3. Create, browse, and search for data, to ensure that the application is running.

   Note: When you create test items, ensure that the permissions are set up so that you have access to view them.

Removing a deployment of i2 Analyze

You might need to reset a deployment of i2 Analyze to its just-installed state. For example, if you completed an example deployment and then want to start creating a production deployment on the same system.

Before you begin

Important: This completely removes all the data and deployed components of i2 Analyze. Do not complete these steps without backing up your data. For more information about backing up your data, see Backing up a deployment.

About this task

To remove a deployment of i2 Analyze, you must remove the database from the system and then the deployment and data files from disk.

Procedure

1. Stop the deployment of i2 Analyze:
   - In a single-server or remote database topology, run `setup -t stop`.
   - In a multiple-server topology, see Stopping and starting i2 Analyze.
2. If your deployment contains a data store, drop the database from the system:

   ```
   setup -t dropDatabases
   ```

   A message is displayed when you run the toolkit task to confirm that you want to complete the action. Enter Y to continue.

   To ensure that the database is removed correctly, use a database management tool to try and connect to the database.
3. Move or delete the deployment and data directories in every i2 Analyze deployment toolkit in your environment.

   By default, the directories are: `IBM\i2analyze\deploy` and `IBM\i2analyze\data`:
   - The deployment directories for the components of i2 Analyze, and the data directory for i2 Analyze is specified in the `environment.properties` file. For more information see `environment.properties`.
   - The data directories for the components of i2 Analyze are specified in the `topology.xml` file. For more information about this file, see `topology.xml`.
4. Move or delete the `toolkit\configuration` directory in every i2 Analyze deployment toolkit in your environment.

Results

The i2 Analyze deployment is returned to its just-installed state.

Best practices

When you create or manage a production deployment of i2 Analyze, there are a number of best practices that you might follow.

Server host names

When you have multiple environments with i2 Analyze deployed, it can be useful to use host names for the servers and application that identifies their purpose. For example, in your development environment you might set the host names of servers as follows: *i2analyze.development.server*.

Configuration management

When you create and maintain a deployment of i2 Analyze, it is an iterative process. Because of the iterative nature, it is important to keep a record of changes to the i2 Analyze configuration. By using a source control system, you can maintain a history of changes to the configuration.

After you populate the configuration directory in the pre-production environment, make a copy of the directory in a location that is managed by a source control system.

Permanent environments

You can maintain your final development and test environments so that you can return to previous deployment phases or use the environments to test any future upgrades of the system.

Production changes

Make any configuration changes in the lowest available deployment, then promote up to production.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Limited Hursley House Hursley Park Winchester, Hants, SO21 2JN UK

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other names may be trademarks of their respective owners. Other company, product, and service names may be trademarks or service marks of others.

Part Number: 99F9999
Product Number: 5725-G23

SC27-5091-00