

IBM IoT for Energy and Utilities
Version 2 Release 8

Application Guide



Note

Before using this information and the product it supports, read the information in [“Notices” on page 183](#).

This edition applies to IoT for Energy and Utilities on Cloud.

Executive summary

This document contains the IBM® IoT for Energy and Utilities product overview application overview, installation instruction, administrative guide, and application and user instructions for Asset Performance Management and Asset Investment, Connectivity Model, Asset 360 for Wind and Situational Awareness.

The installation instructions describe both a Docker install and the upgrade instructions for a non Docker installation.

The administering of the product describes the global settings for the product, the instructions for services, the system status monitoring instructions, and managing the standard operating procedures.

The Asset Performance Management application describes the custom data model that is used by the application, the customizing of the application, and the viewing and running of an analysis.

The Asset Investment application section describes the creating investment projects and viewing the results with comparisons.

The Using Connectivity Model describes the data flow and configuration of the application, the automation of the data flow and the user instructions for Connectivity Model.

The Optimizing Wind Farm operations describes the overview of the application, the custom data model information, the administration of the application and the monitoring of wind farm operations.

Contents

Executive summary.....	vii
Chapter 1. Product overview.....	1
What's new.....	1
Product offerings and features.....	1
Provided hardware configuration.....	3
IoT for Energy and Utilities and IoT Predictive Maintenance and Optimization.....	4
Architecture.....	5
Security.....	7
Accessibility.....	8
Chapter 2. Installing the solution.....	9
Planning the architecture.....	9
Planning the hardware and software needed.....	11
Software included in IBM IoT for Energy and Utilities.....	13
Installing the solution for a new deployment.....	14
Downloading and extracting the Docker installer packages.....	14
Preparing the target servers.....	15
Building the HDP registry data for the Connectivity Model application.....	18
Preparing the install node.....	19
Loading of the Docker image and preparing the installation environment.....	20
Installing the IBM IoT for Energy and Utilities environment.....	24
Validating the install process.....	27
Uninstall if the install fails.....	28
Customizing the standard configuration of nodes, slave nodes and HDP client nodes.....	28
Post installation for the Connectivity Model application.....	30
Configuring the Connectivity Model.....	30
Configuring the automatic restart time for the liberty server.....	31
Configuring Zeppelin Notebook, optional configuration.....	31
Testing the Connectivity Model End to End for cm_sample (Optional)	33
Upgrading from IBM IoT for Energy and Utilities version 2.5.0.x to version 2.8.0. for a Docker environment	33
Preparing the installation node for upgrading a Docker environment.....	34
Finalizing the upgrade for a Docker environment.....	35
Finalizing the upgrade for the Connectivity Model application.....	36
Upgrading from IBM IoT for Energy and Utilities version 2.5.0.x to version 2.8.0. for a non-docker environment	37
Upgrading from IBM IoT for Energy and Utilities version 2.5.1 to version 2.8.0. for a non-Docker environment.....	37
Upgrading from IBM IoT for Energy and Utilities version 2.5 to version 2.5.0.1 for a non Docker environment.....	41
Upgrade IoT for Energy and Utilities with Connectivity Model from version 2.5.0.1 to 2.8.....	44
Adding a new HDP slave node after installation.....	45
Preparing for the new HDP slave node.....	46
Updating the current installer configuration.....	46
Adding the new node to the existing etcd and calico network.....	48
Starting the ambari-agent container on the new node.....	50
Installing the HDP slave services on the container.....	50
Copying the existing keytab files to the new slave container.....	51
Installing the client component from Predictive Maintenance and Optimization.....	51
Troubleshooting the installation environment.....	51

Cannot access the Cognos BI link.....	52
Error message, Error response from daemon: service endpoint with name 'container name' already exists.....	52
The cron job to restart liberty server framework_server did not work due to file permissions.....	53
Cannot access the Application link through IBM HTTP Server.....	53

Chapter 3. Administering the product.....55

Stop and start solution software services.....	55
Start solution services.....	55
Stop solution services.....	57
Starting and stopping services.....	58
Stop the solution services.....	59
Start solution services.....	61
Applying calico network settings after a system reboot or a container restart.....	63
Managing system access.....	64
Adding users and user groups to access the user interface.....	64
Modifying users, user groups, and passwords for the user interface.....	66
Adding a Connectivity Model tenant user.....	68
Removing a Connectivity Model tenant user.....	70
Giving the UI user the permission to access Connectivity Model tenant user data.....	72
Modifying the tenant user and changing tenant user passwords for the Connectivity Model application.....	73
Mapping user groups to license types.....	74
Connecting IBM IoT for Energy and Utilities to a GIS system.....	74
Monitoring system status and backing up the system.....	76
Monitoring the Asset Performance Management and Asset 360 for Wind applications.....	76
Backing up the system for the Asset Performance Management and Asset 360 for Wind applications.....	77
Monitoring the Connectivity Model application.....	78
Backing up the system for the Connectivity Model application.....	78
License usage metrics.....	79
Changing the details of the login page.....	81
Optimizing performance of the IBM Db2 database.....	81
Performing IBM Watson IoT Platform integration administration.....	82
Creating integrations.....	82
Adding a subscription to an integration.....	83
Editing integrations.....	83
Managing integrations.....	83
Archiving the event files.....	84
Reuse existing LDAP Server for IBM IoT for Energy and Utilities.....	84
Preparing for configuring the LDAP registry for IBM IoT for Energy and Utilities.....	84
Configuring the LDAP user registry in Liberty.....	87
Managing the Standard Operating Procedures.....	89

Chapter 4. Using Asset Performance Management application..... 93

Managing the custom analysis model.....	94
Uploading a custom analysis model and stream.....	95
Setting the global parameters and configuring the analysis model.....	96
Running an analysis	97
Integrating with Visual Insights.....	97
Defining a measurement reading table name for Visual Insights integration	97
Defining the measurement type and associating it to the reading table.....	98
Importing the asset measurements.....	98
Configuring the Visual Insights rules.....	99
Uploading image files for Visual Insights reports.....	99
Showing and Hiding the map street view.....	99
Viewing and analyzing energy data.....	100

Logging on to the IBM IoT for Energy and Utilities Asset Performance Management application.....	101
Navigating the user interface of IBM IoT for Energy and Utilities.....	101
Preview cards.....	102
Filter selector.....	103
Viewing the health status of assets in the map view.....	106
Viewing the physical location of a single asset in street view.....	110
Viewing the health status of assets classes in the list view.....	110
Viewing the health status of multiple asset classes in the report view.....	111
Viewing the health status of a single asset class in the report view.....	113
Viewing the health status of a single asset in the report view.....	117
Viewing the health status of assets classes in the matrix view.....	120
Exporting data for a single asset class	122
Viewing analytics dashboards.....	122
Asset Investment application.....	122
Creating an investment project.....	123
Viewing the results of the investment project in the map view.....	123
Viewing the results of the investment project in the list view.....	125
Creating a scenario.....	126
Comparing the results of a scenario in the report view.....	127
Chapter 5. Using Connectivity Model.....	129
Overview of the data flow.....	129
Preparing the data for the ETL module.....	129
The ETL process.....	136
Validation of the ETL process (optional process).....	146
Preparing the data for the operational store.....	148
The analysis process of the data.....	148
Configuring the Connectivity Model application.....	149
Encrypting the SMTP password.....	153
Loading data to the Connectivity Model.....	154
Loading the master data for the Connectivity Model application.....	154
Loading the reading data for the Connectivity Model application.....	155
Populating the master to the operational store.....	156
Administration of the Connectivity Model application.....	156
Running the supplied Connectivity Model analyzes.....	157
Populating the analysis results to the operational store.....	158
Automating the data flow.....	159
Automating the data flow - Quality Reports.....	161
Disabling and enabling the load and voltage algorithms.....	169
Using the connectivity model application.....	169
Logging onto the Connectivity Model application.....	170
Viewing the legend of the connectivity model application.....	170
Viewing the basic information of a substation and its feeder.....	171
Viewing the detailed information for transformers and the meters.....	172
Showing the confidence level of the connectivity results.....	173
Exporting the asset information of a utility.....	174
Chapter 6. Optimizing wind farm operations.....	175
Overview of the Asset 360 for Wind application.....	175
The Asset 360 for Wind application dashboard.....	175
Subscribing to the IBM Insights for Weather service.....	177
Configuring for maintenance planning optimization analysis.....	177
Configuring and creating a maintenance plan.....	177
Administering the Asset 360 for Wind application.....	178
Performing simulator administration.....	178
Looking at a holistic view of the operations.....	178
Comparing one wind farm to the others in the same company.....	179

View the trend of the power generated over time.....	179
Viewing repair and restoration statistics.....	179
Viewing the maintenance costs.....	179
Viewing the utilization hours.....	180
Seeing implications of weather on the operation.....	180
Monitoring detailed operations.....	180
Monitoring status of a turbine.....	181
Viewing the detailed condition of a wind turbine.....	181
Comparing the wind speed and power generation.....	181
Viewing wind speed and direction trends.....	182
Viewing utilization hours.....	182
Trademarks.....	184
Terms and conditions for product documentation.....	184
IBM Online Privacy Statement.....	185

Chapter 1. Product overview

IBM IoT for Energy and Utilities is an open analytics solution that is designed to meet a wide range of current and future provider needs. Via data integration, analytics, and visualization it provides a detailed, accurate understanding of historical and current asset and network performance. It integrates with existing data sources and operational processes to analyze and predict asset performance and risk to help deliver safe, reliable, affordable, and sustainable energy. Key applications include asset performance management, situational awareness, health and risk, investment planning, connectivity model verification, and wind farm optimization.

What's new

This release of IBM IoT for Energy and Utilities includes new methodologies based on industry standards, new asset classes changes to the interface usability and performance, and includes new applications for the performance management of assets and investment planning. The release also includes the ability to integrate with IBM Cognos Analytics.

New methodologies

Enhancement of the data filters for the export of asset class data in the Asset Performance Management application. You can select which asset class and data type you want to be exported. Refer to [Exporting data for a single asset class](#).

Asset Manager can use Google Street Map to view the surrounding environment of the asset. Refer to [“Viewing the physical location of a single asset in street view”](#) on page 110.

IBM IoT for Energy and Utilities introduces the ability to integrate with IBM Visual Insights to upload images from the field and to use those images in reports for an asset class. Refer to [Integrating with IBM Visual Insights](#).

The integration with IBM Cognos Analytics lets you create your own IBM Cognos Analytics report and integrate that report as a page in IBM IoT for Energy and Utilities. Refer to [Integration with IBM Cognos Analytics](#).

The analytics model for Asset Performance Management is available via the user interface. You can manage, configure the analytic parameters, and run an analysis from the administration pages. Refer to [“Managing the custom analysis model”](#) on page 94

Product offerings and features

IoT for Energy and Utilities is an analytics platform for the energy and utilities industry. You can use the platform to develop new applications to support analytics use cases for your assets and networks, and to integrate existing applications with the solution.

Product introduction

IBM IoT for Energy and Utilities is an analytics platform for the energy and utilities industry. You can use the platform to develop new applications to support analytics use cases for your assets and networks, and to integrate existing applications with the solution.

You can use the extension capabilities that are provided by the user interface framework to build application user interfaces that meet your operational requirements.

IoT for Energy and Utilities is installed on IBM IoT Predictive Maintenance and Optimization, so that you can also use predictive maintenance capabilities to help you to anticipate asset failures and to predict the need for maintenance.

Asset Performance Management

Asset Performance Management provides a historical and real-time assessment of critical assets. By analyzing the relevant data sources and applying the built-in asset models, Asset Performance Management provides a real-time assessment of asset performance. The models can be customized as needed.

IBM IoT for Energy and Utilities is delivered with a custom data model that contains predefined asset classes for Asset Performance Management. These predefined asset classes are:

- Transformers,
 - Distribution transformer,
 - Substation transformer, (also know as power transformer),
 - Instrument transformer,
- Conductors
 - Overhead conductors,
 - Underground conductors,
 - Liquid-filled (field developed),
- Circuit breaker,
 - Oil filled,
 - Air blast,
 - Air magnetic,
 - Vacuum,
 - SF6
- Battery,
- Utility poles,
- Distribution towers and steel structures,
- Switchgear.

Asset Investment

Operations personnel can determine the best investment plan possible according to the provider's objectives and constraints. Investment plans can be exported to asset investment plans for analysis. The analysis can be guided by corporate level investment constraints such as risk-level or budget-level as the criterion for developing the investment scenario.

The risk-level constraint generates an asset replacement plan with fixed risk threshold. The plan keeps total risk within a threshold limit.

The budget-level constraint generates an asset replacement plan with a fixed budget. The plan keeps the total risk at lowest level.

Connectivity Model

The Connectivity Model application identifies which meters are connected to each of the electrical phases and provides recommendations to help fix connectivity records without having to send crews into the field. The patented linear analytics algorithm has accuracy greater than 90%, and uses AMI/smart meter data. Automated verification can improve fault location, isolation and service restoration, facilitate accurate analysis after system trouble, and provide greater detail of the outage extent.

Asset 360 for Wind

The Asset 360 for Wind application offers role-based access to historical, current and predictive insights for wind turbines and wind farms. Wind 360 provides situational awareness for assets that use historical and real-time data that is obtained from the relevant operational systems. Integration with the IBM

Weather Company further optimizes wind farm performance and assesses turbine health and risk. Reports provide an overall and detailed analysis of the wind farm performance, operation, and maintenance based on KPIs.

Situational Awareness

The Situational Awareness application helps users monitor the changes to the status of assets in the real time with visualizations, so that action can be taken to solve issues or plan and optimize the settings for the assets.

REST services and UI framework

The product provides REST services that can be used to extend the product and provide integration interfaces to other systems.

This foundation can be used to:

- Unify systems and business processes by integrating multiple data sources such as sensors, SCADA, weather, EAM (Enterprise Asset Management).
- Deliver contextual awareness by correlating, analyzing, and visualizing data within and across systems and processes.

The UI framework can be used to develop and extend your own applications.

IBM IoT Predictive Maintenance and Optimization

The IBM® IOT Predictive Maintenance and Optimization solution uses data from multiple sources to give you the information to make informed operational, maintenance, or repair decisions.

The IOT Predictive Maintenance and Optimization solution provides two options: A standard configuration and a big data configuration.

IBM Open Platform integration

IBM Open Platform with Apache Spark and Apache Hadoop is a platform for analyzing and visualizing Internet-scale data volumes that is powered by Apache Hadoop. Open Platform is an open source distributed computing platform.

IBM Watson™ IoT Platform Integration application

IBM Watson IoT Platform provides a user interface where you can add and manage your devices, control access to your IoT service, and monitor your usage.

You can integrate IoT for Energy and Utilities with IBM Watson® IoT Platform to collect asset data from devices that are connected to Watson™ IoT Platform.

Provided hardware configuration

IBM IoT for Energy and Utilities has a default hardware configuration that changes depending on the application.

Node	CPU	Memory	Disk
IIB node	Four cores	8 GB	250 GB
DB node	Sixteen cores	32 GB	200 GB
Ana node	Four cores	8 GB	400 GB
BI node (if used)	Four cores	8 GB	100 GB

Connectivity Model on big data architecture, the default disk space for each tenant is 1 TB.

IoT for Energy and Utilities and IoT Predictive Maintenance and Optimization

As described, IoT for Energy and Utilities is supported by IBM IoT Predictive Maintenance and Optimization.

IBM IoT Predictive Maintenance and Optimization includes supported products for both server and client use.

Software included in IBM IoT for Energy and Utilities

Software is included in the delivery of IoT for Energy and Utilities for use on the servers and that also includes optional software.

The server and client tools are included in the delivery and are listed here:

Server software

The server software that is installed as part of IoT for Energy and Utilities

- IBM DB2 Enterprise Server Edition
- IBM WebSphere Network Deployment Supplements Pack
- IBM WebSphere Application Server Liberty
- IBM Cognos Analytics Server
- IBM Integration Bus
- IBM Integration Bus Manufacturing Pack for Enterprise
- IBM WebSphere Message Queue
- IBM SPSS Modeler Server Premium
- IBM SPSS Modeler Batch

Optional server software

These optional server software offerings must be installed in a separate installation process by the user. If you wish to install any of these products, please refer the [Knowledge Center](#) for the installation instructions.

- IBM Installation Manager
- IBM WebSphere Application Server Network Deployment
- IBM SPSS Collaboration and Deployment Services Server
- IBM SPSS Statistics Server
- IBM SPSS Analytical Decision Management
- IBM SPSS Analytic Server
- IBM Cognos SDK
- IBM Data Server Runtime Client
- IBM Java SDK

Client tools

These tools are part of the client software package:

- IBM Cognos Framework Manager
- IBM Cognos SDK
- IBM Integration Bus Toolkit
- IIB WebSphere MQ Client
- IBM SPSS Collaboration and Deployment Services Deployment Manager
- IBM SPSS Modeler Client

- IBM SPSS Statistics Client
- IBM SPSS Data Access Pack
- IBM Data Server Client

Optional software under separate License agreements

These optional software offerings require separate license agreements:

Server software

- IBM Visual Insights
- IBM Decision Optimization Center
- IBM ILOG CPLEX Enterprise Server Software
- IBM Data Model for Energy and Utilities

Client software

- IBM ILOG CPLEX Optimization Studio

Architecture

Architectural overview

IBM IoT for Energy and Utilities is built upon the IBM Predictive Maintenance and Organization solution. The diagram shows the relationship between the components that are the solution artifacts that are used by IBM IoT for Energy and Utilities and IBM Predictive Maintenance and Organization and the base foundation components.

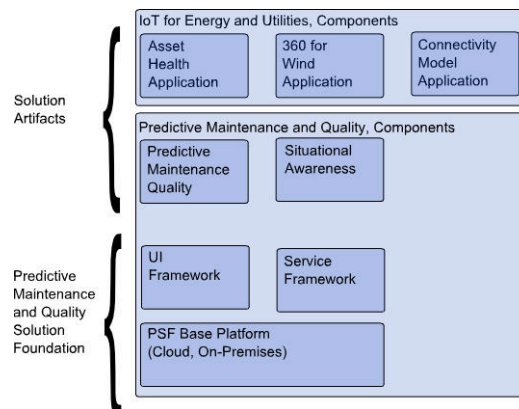


Figure 1. Architecture overview

Asset Health application

The figure shows the deployment architecture for IBM IoT for Energy and Utilities with the Asset Performance Management solution.

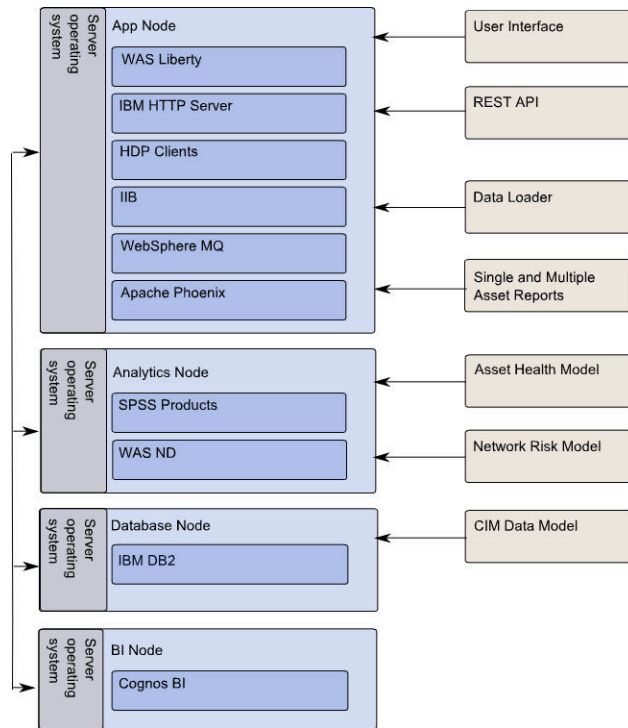


Figure 2. Asset Health Deployment Architecture

360 For Wind application

The figure shows the deployment architecture for IBM IoT for Energy and Utilities with the Asset 360 for Wind solution.

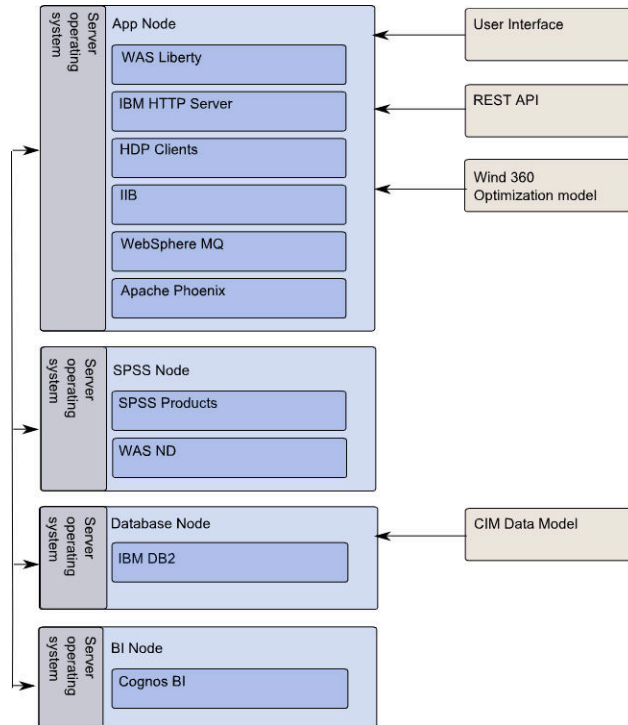


Figure 3. Asset 360 For Wind deployment architecture

Connectivity Model application

The figure shows the deployment architecture for IBM IoT for Energy and Utilities with the Connectivity Model solution.

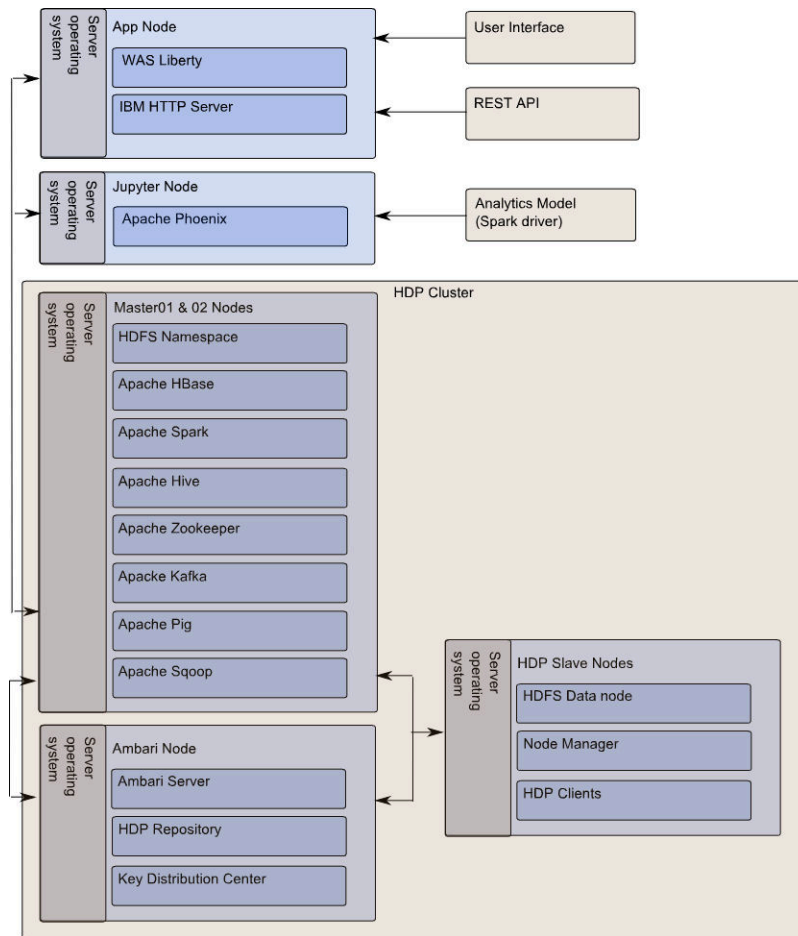


Figure 4. Connectivity Model Deployment Architecture

Security

The SaaS delivery for IBM IoT for Energy and Utilities has four levels of security.

The security levels are:

- Encryption
- LDAP
- Logging
- Single Sign on

Encryption

IBM IoT for Energy and Utilities uses SSL with RSA encryption for the internal server communication.

- Between the IHS (IIB Node) and the WAS Liberty (IIB Node)
- Between the IHS (IIB Node) and the Cognos (BI Node)
- Between the ASK (BI Node) and the Cognos (BI node)
- Between the IBM DB2 (DB Node) and the SPSS Modeler Server (SPSS Node), the WAS Liberty (IIB Node) the grid engine (SPSS Node) and Cognos (BI Node)
- Between the LDAP (DB Node) and the WAS Liberty (IIB Node)

- Between the LDAP (DB Node) and the Cognos and ASK (BI Node)

IoT for Energy and Utilities uses the native encryption in IBM DB2 for the encryption of data and the files that contain customer and user data are stored on an encrypted folder on the App Node.

The external communication is encrypted using TLS v1.2 between the browser interface and the Insights Foundation for Energy application, and between the browser interface and the Web Node on the Cognos portal using SSL (HTTPS).

LDAP internet protocol permissions

IoT for Energy and Utilities uses the LDAP protocol to store the login credentials of the users.

User logging

The login and logout of users and the user activity whilst active is recorded and monitored to detect abnormal activities at the following levels:

- Application level - The login and logout activities to an application are recorded.
- Middleware level - Activities are monitored and recorded for the middleware: WAS Liberty, HIS, ASK, Cognos, SPSS Modeler Server, IBM DB2, and Jena.
- Operating system level - Logrotate is enabled to manage the log files for: /var/log/cron, /var/log/maillog, /var/log/spooler, /var/log/dmesg, /var/log/boot.log, /var/log/messages, and /var/log/secure.
- Logrotate - Logrotate is enabled for activity logs on the middleware, the Insights Foundation for Energy application log files are included in the WAS Liberty server on the application node.

Single Sign On

IoT for Energy and Utilities uses single sign-on to secure the transfer of user ID and password credentials that is used to authenticate with the system. Users can switch between application without having to authenticate again.

Accessibility

IBM is committed to accessibility for tools and guidance.

For more information about the commitment, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: <http://www-03.ibm.com/able/>

Chapter 2. Installing the solution

The Docker install provides a flexible process for installing IBM IoT for Energy and Utilities.

The nodes that are required depend on the applications and processes you need.

Planning the architecture

The architecture of IBM IoT for Energy and Utilities depends on the applications to be used.

Each node must be installed on a separate computer or on a separate virtual machine image from other nodes.

Important: These node suggestions are for example only. You must check your own requirements for data base size and node requirements.

Asset Performance Management and Asset 360 for Wind

For Asset Performance Management and Asset 360 for Wind you required a three node configuration: App node, SPSS node and DB node. If Cognos BI is also required, then a four node configuration is required that includes a BI node for Cognos and the Cognos BI DB2 database.

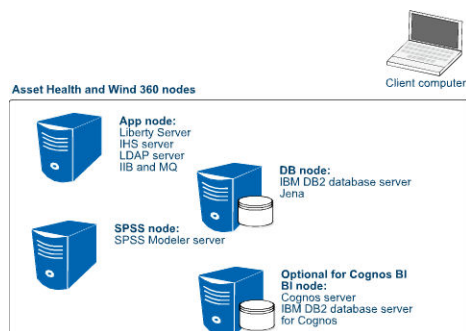


Figure 5. Nodes used for Asset Performance Management and Asset 360 for Wind

Connectivity Model with Hortonworks Data Platform (HDP)

For the Connectivity Model application the App node contains the IHS server, LDAP server and the Liberty server. The seven node cluster contains the Hortonworks Data Platform.

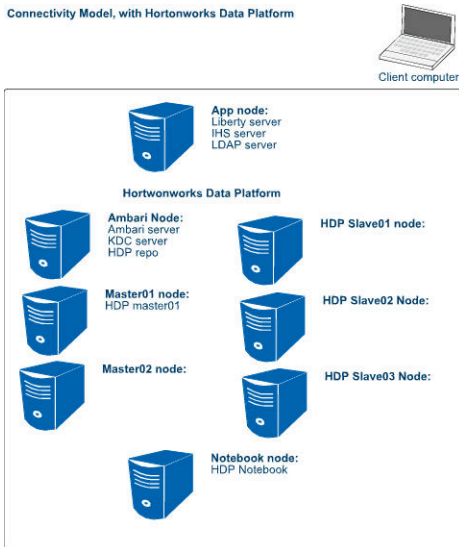


Figure 6. Nodes used for Connectivity Model

The services that are installed on each node are:

Ambari node

KDC server for Kerberos authentication, Ambari server,
HTTP server for the HDP repository for installation

Master01 node

HDFS, HBase, MapReduce2, App Timeline Server, Resource Manager, Hive
Spark History Server, WebHCat Server, Zookeeper, Spark Thrift Server,
Kafa, Knox, Ambari Metrics, HST Agent, Metrics Monitor,
HDP clients, ambari-agent, kdc-client, ldap-client

Master02 Node

HDFS, HBase, HIVE, Zookeeper, Resource Manager, Spark Thrift Server
Kafa, Knox, HST Agent, Metrics Monitor,
HDP Clients, ambari-agent, kdc-client, ldap-client

Slave01 Node

HDFS DataNode, HDFS JournalNode, HBase Region Server, Zookeeper, Node Manager,
ambari-agent, kdc-client, ldap-client, SmartSense, Metrics Monitor, HST Server,

Slave02/Slave03 Node

HDFS DataNode, HBase RegionServer, NodeManager,
HDP clients, ambari-agent, kdc-client, ldap-client, Metrics Monitor, HST agent

Notebook Node

Zeppelin Notebook, HDP clients, ambari-agent, kdc-client, ldap-client, Metrics Monitor, HST agent

Note: Kafka, Knox, Metrics, HST, SmartSense services are not used by Connectivity Model

Asset Performance Management or Asset 360 for Wind and Connectivity Model with Hortonworks Data Platform

When the Connectivity Model with HDP and either Asset Performance Management or Asset 360 for Wind are required, then you require the configuration 4 + 7. The App node, SPSS node and DB node and the optional BI node are required for the Asset Performance Management application and the App node and

the seven node cluster required for HDP is required for the Connectivity Model. The App node is shared between the applications.

Connectivity Model with Hortonworks Data Platform (HDP) in a one plus two topology

The one + two topology is used for developmental purposes. For the Connectivity Model application the roles that are installed on each node are:

App node

Liberty server, LDAP server, IHS server, installer

bigdataNodeOne

cm_backend_contaier, notebook_container, master01_container, slave01_container, cm_blueprint_container, registry

bigdataNodeTwo

repo_container, kdc_container, ambari_container, master02_container, slave02_container

For more information about Ambari, refer to <https://ambari.apache.org>

For more information about Hortonworks Data Platform (HDP), refer to <https://hortonworks.com/products/data-platforms/hdp/>

Planning the hardware and software needed

Review the minimum hardware and operating system requirements before you install IBM IoT for Energy and Utilities. The requirements apply for all computers or virtual machines that you use for the installation.

For an up-to-date list of environments that are supported, see the [IBM Software Product Compatibility Reports](#). Click **Create a report** under **In-depth reports**, search for **Energy and Utilities** in the **Full or partial product name** field, select **IBM IoT for Energy and Utilities**, and click **Submit**.

If you are expecting to store large volumes of data, you might need to increase your disk size.

Each node must be installed on a separate computer or on a separate virtual machine image.

Server computers

The IoT for Energy and Utilities server components must be installed on computers that are running one of the following operating systems:

- 64-bit Red Hat Enterprise Linux Server Edition version 7.2 (x86_64)
- CentOS Linux 7.3.1611
- Ubuntu 16.04 LTS (Xerus)

The prerequisite software requirements that are installed as part of these installation procedures are:

- Docker Community Edition 17.03.1

Note: The Docker installer assists you to install the IoT for Energy and Utilities solution. You need to confirm the acceptance of the IBM license terms before you install the solution.

Docker directory requirements

The following directory requirements for are for the installation and the retention of data for Docker. These requirements are extra to the requirements for the applications:

- The node that hosts the Docker private registry must have a 30 GB allocation to the /registry-data directory and 100 GB allocated to the /tmp directory. By default the host for the Docker private registry is the APP node.
- The host to build the HDP registry data file needs 30 GB space directly under the root or /data directory.
- The node that you use to the host the Docker private registry must be different than the node that you run the installation from. For example: If the APP node hosts the /registry-data directory, use the DB node to run the installation.

- All Docker data is located in the `/var/lib/docker` directory. If this directory is located in another disk other than the the same disk with `/root` directory, the directory must satisfy the different applications hardware disk requirements described in this section.
- The installer node for Docker CE must have the latest versions of `gzip` and `gunzip`.

Hardware Requirements for Asset Performance Management and Asset 360 for Wind

At a minimum, each computer or virtual machine that hosts a IoT for Energy and Utilities server component, or node, must have the following minimum hardware requirements. The disk size is for `/root` directory or `/var/lib/docker` directory:

- APP node: 4 processors, 8 GB of RAM, 200 GB hard disc space
- SPSS node: 4 processors, 8 GB of RAM, 200 GB hard disc space
- BI node: 4 processors, 8 GB of RAM, 200 GB hard disc space
- DB node: 4 processors, 16 GB of RAM, 200 GB hard disc space or 16 processors, 32 GB RAM, or 500 GB in the root directory if the database requires more space.

Hardware requirements for a Connectivity Module with Hortonworks Data Platform (HDP) environment

The suggested HDP topology is a one plus seven node environment, one Ambari node, two management nodes and three database or subordinate nodes and a notebook node. An HDP cluster hardware size is related to the data. For a large database, the topology can change for the number of nodes and require more hardware resources. This configuration is for normal data size. The disk size is for the `/root` directory or `/var/lib/docker` directory.

- APP node: 4 processors, 8 GB of RAM, 200 GB hard disc space
- Ambari node: 16 GB RAM, 8 processors, 100 GB hard disk space
- 2 x master nodes: 48 GB RAM, 8 processors, 1 TB hard disk space
- 3 x HDP slave nodes: 64 GB RAM, 16 processors, 3 TB hard disk space
- Notebook node: 16 GB of RAM, 4 processors, 1 TB hard disk space
- A Redhat or Ubuntu Linux node that has Internet access for you to build the Hortonworks registry. This node can be an extra node that is not required after the completion of the installation process or one of the target nodes. If you use one of the target nodes, after you build the Hortonworks registry, you must clear the Hortonworks artifacts from that node.

If you have a one plus two node topology for developmental purposes, then the requirements are for an application node and two big data nodes as follows:

- APP node: 54 GB of RAM, 8 processors, 1350 GB hard disk space
- `big_data_node_one`, 32 GB RAM, 8 processors, 700 GB hard disk space
- `big_data_node_two`, 32 GB RAM, 8 processors, 700 GB hard disk space

Client computers

The IoT for Energy and Utilities client components must be installed on computers that run Microsoft Windows 7 or Microsoft Windows 8 operating systems.

Clients can be installed on 32-bit or 64-bit computers. 64-bit is recommended.

At a minimum, the computer where you run the IoT for Energy and Utilities for the client components must have the following hardware requirements:

- 4 processors
- 32 GB of RAM
- 300 GB of hard disk space

Browsers

The IBM IoT for Energy and Utilities user interface is supported in several browsers.

- Google Chrome 61
- Microsoft Internet Explorer 11
- Mozilla Firefox 52 ESR
- Safari 11 for Mac OS

Software included in IBM IoT for Energy and Utilities

Software is included in the delivery of IoT for Energy and Utilities for use on the servers and that also includes optional software.

The server and client tools are included in the delivery and are listed here:

Server software

The server software that is installed as part of IoT for Energy and Utilities

- IBM DB2 Enterprise Server Edition
- IBM WebSphere Network Deployment Supplements Pack
- IBM WebSphere Application Server Liberty
- IBM Cognos Analytics Server
- IBM Integration Bus
- IBM Integration Bus Manufacturing Pack for Enterprise
- IBM WebSphere Message Queue
- IBM SPSS Modeler Server Premium
- IBM SPSS Modeler Batch

Optional server software

These optional server software offerings must be installed in a separate installation process by the user. If you wish to install any of these products, please refer the [Knowledge Center](#) for the installation instructions.

- IBM Installation Manager
- IBM WebSphere Application Server Network Deployment
- IBM SPSS Collaboration and Deployment Services Server
- IBM SPSS Statistics Server
- IBM SPSS Analytical Decision Management
- IBM SPSS Analytic Server
- IBM Cognos SDK
- IBM Data Server Runtime Client
- IBM Java SDK

Client tools

These tools are part of the client software package:

- IBM Cognos Framework Manager
- IBM Cognos SDK
- IBM Integration Bus Toolkit
- IIB WebSphere MQ Client

- IBM SPSS Collaboration and Deployment Services Deployment Manager
- IBM SPSS Modeler Client
- IBM SPSS Statistics Client
- IBM SPSS Data Access Pack
- IBM Data Server Client

Optional software under separate License agreements

These optional software offerings require separate license agreements:

Server software

- IBM Visual Insights
- IBM Decision Optimization Center
- IBM ILOG CPLEX Enterprise Server Software
- IBM Data Model for Energy and Utilities

Client software

- IBM ILOG CPLEX Optimization Studio

Installing the solution for a new deployment

The procedures lead you through the necessary steps to install IBM IoT for Energy and Utilities as a new deployment.

Downloading and extracting the Docker installer packages

Download and extract the images from Passport Advantage for IBM IoT for Energy and Utilities 2.8.0.

You need to download and unpack the following image from Passport Advantage:

- IBM IoT for Energy and Utilities 2.8.0.

The components, from the IoT for Energy and Utilities 2.8.0 image file `IBM_IOT4EU_Server_V2.8_LINUX_ML` in Passport Advantage, are:

- `registry-data-full28.tar.gz` Contains the middleware registry images and application artifacts registry images.
- `installer28.tar.gz`
- `hdp-deploy28.tar.gz`
- `ife_custom_ana_spss_service.war`

Note: The file `docker-upgrade28.tar.gz` is for an upgrade of an existing deployment only.

Extracting the files

Important: The contents in each of the compressed files might be more than are listed here. The list contains the contents for this release.

Extract the contents from the `IBM_IOT4EU_Server_V2.8_LINUX_ML.zip` file from the IoT for Energy and Utilities 2.8.0 image from Passport Advantage.

Extract the contents of the `hdp-deploy28.tar.gz` file.

The file `hdp-deploy28.tar.gz` contains:

- `ibmiot.base_os.tar.gz` - the base operating system image for building the HDP images.
- `hdp-repo`, `ambari-base`, `ambari-agent`, `ambari-notebook`, `ambari-server` - the Docker folders for building purposes.

- hdp-deploy The scripts to run build shell. The output is a registry image package for HDP: registry-data-hdp.tar.gz.

Preparing the target servers

Changes to the system setting and the installation of Docker CE are requirements for each node for IBM IoT for Energy and Utilities.

These sections tell you what need to do to change the system settings, where you can download the Docker CE v17.03.1 image and the find the install instructions for the following.

- Installation of the iptables and systemd package.
- Disabling Selinux.
- How to enable root login. If root login is not allowed to be used you need to create a sudo user. The instructions are here [“Create a SUDO user on all nodes if root user is not allowed” on page 15.](#)
- Disabling the firewall during the install.
- Updating the /etc/hosts file.
- Installation of Docker CE v17.03.1.

Create a SUDO user on all nodes if root user is not allowed

If root is not allowed to be used, you need to create a sudo user on all nodes that does not require to enter a password when running sudo commands.

About this task

The sudo user is used when log in to the hosts.

Procedure

1. Assume the sudo user is *installerUser*, and you must update *<password>* with the actual password of the user:

```
adduser -m installerUser
echo <password> | passwd --stdin installerUser
```

2. Open with a text editor of your choice:

```
/etc/sudoers
```

and add the lines:

```
installerUser ALL=(ALL) NOPASSWD: ALL
```

Note: The SUDO user uses the Bash shell.

Configuring all nodes for Docker

Do these steps on each node required for your settings to install Docker CE v17.03.1.

Procedure

1. Make sure the services iptables and systemd are installed.

If you have either Redhat or CentOS operating system and you have yum repositories configured for your Linux operating system use the following commands to install all the prerequisite RPM files:

- a. Log in the node as a root user.
- b. Run the following command:

```
sudo yum -y install iptables systemd
```

If you do not have yum repositories configured:

- a. Log in the node as a root user.
- b. Download the missing RPM package.
- c. Install the package by the command:

```
sudo rpm -ihv <full_package_name>.rpm
```

If you have Ubuntu operating system and you have apt repositories configured for your Linux operating system, use the following commands to install all the prerequisite Debian packages:

- a. Log in the node as a root user.
- b. Run the following command:

```
sudo apt-get -y install iptables systemd
```

If you do not have apt repositories configured:

- a. Log in the node as a root user.
- b. Download the missing .deb package.
- c. Install the package by the command:

```
sudo dpkg -i <full_package_name>.deb
```

2. If you have Selinux installed, open the file `/etc/selinux/config` and set:

```
SELINUX=disabled
```

For example:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. Reboot the host, and run command to check.

```
sestatus
```

the output will look like this:

```
SELinux status:                disabled
```

4. Enable root login if not permitted on all nodes.

The enable root login can be set temporarily and can be disabled after installation is complete.

- a) To enable a root user, open the file: `/etc/ssh/sshd_config`.
- b) Add or edit the line in the Authentication: section of the file that says `PermitRootLogin` yes.

For example:

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

- c) Restart the SSH server.

For example:

```
sudo systemctl restart sshd
```

5. Stop the services iptables and firewalld on all nodes.

For example:

```
sudo service iptables stop; sudo service firewalld stop
```

Note: If you receive an error, Failed to stop iptables.service: Unit iptables.service not loaded or Failed to stop firewalld.service: firewalld.service not loaded, when the system does not have iptables.service or firewalld.service installed you can ignore these errors and go to the next step.

6. Open the file /etc/hosts to configure the full host name and short host name of that node.

```
<ip address of the node> <full hostname of the node > <short hostname of the node>
```

where the <full host name of the node> is the fully qualified name, and the <short host name of the node> is the short hostname for the Linux host. For example:

```
127.0.0.1 localhost.localdomain localhost  
::1 localhost6.localdomain6 localhost6  
172.254.254.1 libertyServer.cn.ibm.com libertyServer
```

7. Use Bash shell for the root user on all nodes.

- a) Login to the node as root user, ensure all nodes must use bash shell.

Redhat and CentOS use bash as the default shell.

- b) Ubuntu, by default uses dash, so you need reconfigure to use bash. Run the command and select No to reconfigure Ubuntu not to use dash by default.

```
dpkg-reconfigure dash
```

- c) Run the command to make sure.

```
echo $0
```

the result is:

```
-bash
```

Installing Docker CE on all nodes and preparing the install node

Docker CE 17.03.1 must be installed on all nodes.

About this task

Go to the official site to install Docker CE 17.03.1, click the appropriate link for either Redhat and CentOS or for Ubuntu.

Procedure

1. Download and install Docker CE 17.03.1 on each node.

Important: You need to install the specific Docker CE version 17.03.1 from the official site. The steps to download is different depending on the operating system that you use.

- For Redhat and CentOS open the link [Docker Community Edition for CentOS](#) and do the steps to install Docker CE 17.03.1.
- For Ubuntu open the link [Docker Community Edition for Ubuntu](#) and do the steps to install Docker CE 17.03.1.

2. Choose one node as the installer node.

Note: By default, the App node hosts the Docker private registry. Use another node as the installer node, for example the DB node in a 4+7 topology, and bigdataone in a 1+2 topology. The Installation node must not be on the same node as the Docker registry.

3. Make sure that gunzip and gzip are installed on the installer node.

a) Install gzip if not installed.

b) Install gunzip if not installed.

c) If you use either RedHat or CentOS operating system, on the installer node open the file `/usr/lib/systemd/system/docker.service` and find the line `ExecStart=/usr/bin/dockerd` and append the storage setting `--storage-driver=devicemapper` and `--storage-opt dm.basesize=500G` in the line.

For example:

```
ExecStart=/usr/bin/dockerd --storage-driver=devicemapper --storage-opt dm.basesize=500G
```

a) Start docker with the commands:

```
sudo systemctl daemon-reload
```

```
sudo service docker restart
```

Building the HDP registry data for the Connectivity Model application

If you are planning to use the Connectivity Model application, you need to build the registry for the Hortonworks Data Platform (HDP).

Get the `hdp-deploy28.tar.gz` package from the build package and place it on a Redhat, Centos, or Ubuntu Linux node that has Internet access for you to build the Hortonworks registry. This node can be an extra node that is not required after the completion of the installation process or one of the target nodes. If you use one of the target nodes, after you build the Hortonworks registry, you must clear the Hortonworks artifacts from that node.

The node needs internet access and have Docker v17.03.01.

Building the HDP registry image packages

You need to select one node to build the HDP registry image package `registry-data-hdp.tar.gz`.

Before you begin

The system must have Docker v17.03.1.

The docker registry needs 30 GB space directly under the root or `/data` directory.

The system must have Internet access.

Preparing the Target servers must be complete, refer to [“Preparing the target servers” on page 15](#).

About this task

The system chosen can be one of the nodes, or a separate node that Docker has been installed. These steps require between 3 to 4 hours to complete.

Procedure

1. Open the `/usr/lib/systemd/system/docker.service` file and add the `insecure-registry` configuration for the node that you are using for the build.

Add

```
--insecure-registry <ip>:5001
```

to the line:

```
ExecStart=/usr/bin/dockerd
```

For example:

```
ExecStart=/usr/bin/dockerd --insecure-registry 9.112.229.185:5001
```

2. Run the commands:

```
sudo systemctl daemon-reload
sudo service docker restart
```

3. Run the command to extract the `hdp-deploy28.tar.gz` file

```
sudo tar -zxvf hdp-deploy28.tar.gz
```

4. Go to `hdp-deploy/scripts` folder, and run the command:

```
sudo chmod a+x *.sh
```

5. Start to build the HDP registry data.

```
sudo ./hdp_build.sh
```

The registry data file will be in the `/data/registry-hdp-data/registry-data-hdp.tar.gz` file.

Note:

The scripts clean all the existing docker images or containers on the host at the beginning of the build HDP process.

6. Save the package `registry-data-hdp.tar.gz` to the installation server.

Note: Make sure you save the build package `registry-data-hdp.tar.gz` to the installation server before you run the commands to remove the Hortonworks artifacts.

7. Remove the Hortonworks artifacts. If you have reused one of the target nodes to build the Hortonworks registry, you need remove the Hortonworks artifacts from that node.

Note: This step is not required if you have used a separate node for the HDP registry build file.

```
sudo docker rm -f $(sudo docker ps -a -q)
```

```
sudo docker rmi -f $(sudo docker images -q)
```

```
sudo rm -rf /data/registry-hdp-data
```

Preparing the install node

Use the node you have chosen for the installer node in the section [“Installing Docker CE on all nodes and preparing the install node”](#) on page 17 and share the SSH keys among other nodes.

Before you begin

You need the IP addresses of each of the target nodes to be able to share the SSH keys.

Procedure

1. Log into the installer node with root access.
2. Run the following commands to generate the sshkey pair, and add the key to the list of authorized keys.

```
mkdir -p ~/.ssh
```

```
/bin/echo -e 'y' | /usr/bin/ssh-keygen -q -N '' -t rsa -b 2048  
-C "key for internal access" -f ~/.ssh/docker.id_rsa
```

```
cat ~/.ssh/docker.id_rsa.pub >> ~/.ssh/authorized_keys  
chmod 700 ~/.ssh  
chmod 600 ~/.ssh/authorized_keys
```

3. From the installer node, copy the key to all other nodes.

```
scp ~/.ssh/docker.id_rsa.pub `whoami`@<node_ip_address>:/tmp
```

Where **<node_ip_address>** is the IP address of each target node.

Repeat this step for each node in turn.

4. Go to each node and add the key into `authorized_keys` of the node.

For example:

```
mkdir -p ~/.ssh
```

```
cat ~/.ssh/docker.id_rsa.pub >> ~/.ssh/authorized_keys
```

```
chmod 700 ~/.ssh
```

```
chmod 600 ~/.ssh/authorized_keys
```

Repeat this step for each node in turn.

Loading of the Docker image and preparing the installation environment

Load the installer docker image to the install node and do the steps to start the installation process.

Before you begin

The install files that you need are:

- `installer28.tar.gz`,.
- `registry-data-full28.tar.gz`.
- `registry-data-hdp.tar.gz`.

About this task

The setting up of the environment has steps that must be carried out sequentially.

Procedure

1. Log into the install node as root user.
2. Copy the files `installer28.tar.gz`, `registry_data.full28.tar.gz`, `registry-data-hdp.tar.gz` images to the `/tmp` directory on the install node.

Important: The following commands use the `/tmp` directory as the default location for these files. If you use a different location, you need to modify the path in each of the commands.

3. Load the install image on the node that is used as the installer server.

The files are located in the `/tmp` directory:

```
sudo gunzip -c installer28.tar.gz | sudo docker load
```

4. Open the install node and create the install folder and mount the configuration.

Run the commands

```
sudo mkdir -p /install  
sudo docker run --rm -v /install:/data  
--name=installer ibmiot/ife.installer /bin/bash -c
```



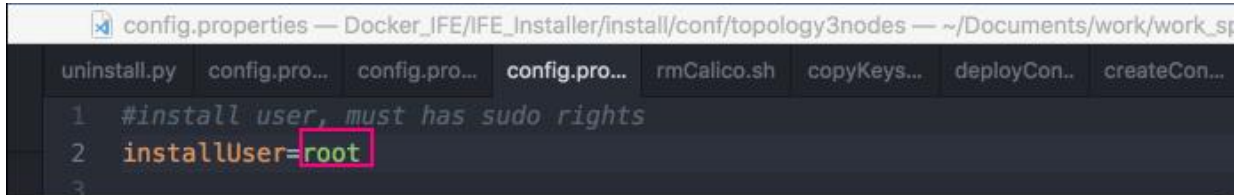
```
"cp -r /install/conf /data;mkdir -p /data/images;mkdir -p /data/ssh_key"
```

5. Open the /install/conf configuration folder.

By default the configuration is a 4+7 topology. If you want to install a 1+2 topology, make a backup of the default containers.ini, nodes.ini, config.properties files. Copy the 1+2 topology files from topology3nodes folder under /install/conf and overwrite the default files.

If you need to have a configuration other than the standard ones delivered in the solution, please refer to the topic, [Customizing the standard configuration of nodes, slave nodes and HDP client nodes](#).

Note: If you are using SUDO user, you must change the user account in config.properties. The default is root.



```
config.properties — Docker_IFE/IFE_Installer/install/conf/topology3nodes — ~/Documents/work/work_sp
uninstall.py  config.pro...  config.pro...  config.pro...  rmCalico.sh  copyKeys...  deployCon...  createCon...
1 #install user, must has sudo rights
2 installUser=root
3
```

Figure 7. Config.properties file installUser setting

- a) Optional: For the Connectivity Model application: When you want to expose more ports for Jupyter notebook, open the /install/conf/containers.ini file and go to the [notebook_container] section, add more ports that are mapping in the exposePorts item. The default exposePorts is:

```
exposePorts=[9995:9995;8888:8888;8889:8889;8887:8887;8886:8886]
```

For example, if you want to expose the external port 8884 for the container internal port 8884 that Jupyter notebook uses in the container, then the item will be like this:

```
exposePorts=[9995:9995;8888:8888;8889:8889;8887:8887;8886:8886;8884:8884]
```

- b) For all applications: Open the nodes.ini file.
c) For all applications: Type the hostname, IP address and the network interface (NIC) for each node. For example:

```
[app_node]
name=appNode
hostname=libertyServer.cn.ibm.com
ip=172.254.254.01
networkInterface=ens192
```

- d) For all applications: When using default the appNode for the registry node, choose another node for the installer node, for example dbNode (4+7) or bigdataNodeOne(1+2). Otherwise place the registry on another node other than the appNode. If you need to change the location of the registry for the Docker registry information, you do this in the registry.ini configuration file. For example if the default node is the appNode, deploy the registry on the dbNode, open the registry.ini file, and change the target value from appNode to dbNode:

```
[registry]
name=ibmiot.ife.registry
target=dbNode
exposePort=5000
```

Note: If you do change the host node of the registry, make sure that the host has at least 100 GB of space under the /tmp directory as described in the [“Planning the hardware and software needed” on page 11](#).

- e) For Asset Performance Management or Asset 360 for Wind applications: Open the /install/conf/nodes.ini file and delete the lines from [ambari_node] to the end of this file.

For 1+2 topology. Open the `/install/conf/nodes.ini` file and delete the lines from `[bigdata_node_one]` to the end of this file.

- f) For Asset Performance Management or Asset 360 for Wind applications: Open the `/install/conf/containers.ini` file and delete the lines from `[cm_backend_container]` to the end of this file.
- g) For the Connectivity Model application: Open the `/install/conf/nodes.ini` file and delete the sections that include the text `[spss_node]`, `[db_node]`, `[cognos_node]`.
- h) For the Connectivity Model application: Open the `/install/conf/containers.ini` file and delete the sections that includes the texts `[db_artifacts_container]`, `[db_base_container]`, `[spss_artifact_container]`, `[spss_base_container]`, `[cognos_db_base_container]`, `[cognos_base_container]`, `[jena_artifacts_container]`, `[jena_base_container]`.
- i) For all applications: If you need to change the password information, you do this in the `config.properties` configuration file.

For example:

```
#ldap server password  
password.ldapServerPassword=password123
```

The values with the default values and descriptions are listed:

encryptKeyString

ibmioteuibmioteu

The `encryptKeyString` must be 16 lengths long for the encrypt function to work.

password.ldapServerPassword

pw4ibmioteu

The password for LDAP server.

password.spssUserPassword

pw4ibmioteu

The SPSS user password to start the modeler server and the `run.sh` analysis program.

password.libertyUserPassword

pw4ibmioteu

The wlp user password to start the liberty server.

password.db2password

pw4ibmioteu

The `db2inst1` user password for the DB2 Server.

password.DBSSLConnKeyPassword

pw4ibmioteuStr0ngpassw0rd

For DB SSL communication. The password be strong . If the password is not strong, additional steps will be necessary.

password.DBEncryptKeyPassword

pw4ibmioteu

The password used for the DB2 server to encrypt the database.

password.cogDB2NodeDBInstanceUserPassword

pw4ibmioteu

The `db2inst1` user password for the DB2 Server.

password.cognosUserPassword

pw4ibmioteu

The Cognos user password to start the cognos server

password.cogDBInstanceUserPassword

pw4ibmioteu

The db2inst1 user password for the IFECogNode container.

password.BobuserPassword

pw4ibmioteu

The administrative user password for admin user for IBM IoT for Energy and Utilities.

password.user1userPassword

pw4ibmioteu

The sample user user1 password for IBM IoT for Energy and Utilities.

password.user2userPassword

pw4ibmioteu

The sample user user2 password for IBM IoT for Energy and Utilities.

password.GaryuserPassword

pw4ibmioteu

The password for the user Gary for the IBM IoT for Energy and UtilitiesAsset 360 for Wind application.

password.MelanieuserPassword

pw4ibmioteu

The password for wind 360 user Melanie for the IBM IoT for Energy and UtilitiesAsset 360 for Wind application.

password.RickuserPassword

pw4ibmioteu

The password for wind user Rick for the IBM IoT for Energy and UtilitiesAsset 360 for Wind application.

openId.enabled

true/false, default false

When openId.enabled is set to true, enables openId authorization.

The user must also provide a valid openId.IBMid and openId.redirectToRPHostAndPort

openId.IBMid

Provided by user

The IBMid that is used to log in application

openId.redirectToRPHostAndPort

Provided by user

This value must be added to Redirect URIs in IBM IoT for Energy and Utilities.

Contact IBM to add the redirect URIs.

For example:

```
https://<ip or hostname of where IFEIHSNode deployed>/oidcclient/redirect/blueid
```

password.kdcAdmin

pw4ibmioteu

The password for KDC admin user.

6. Copy the Docker image registry files registry-data-hdp.tar.gz and registry-data-full128.tar.gz to the /install/images folder.
7. Copy the ssh private key to the /install/ssh_key directory.

Run the command:

```
sudo cp ~/.ssh/docker.id_rsa /install/ssh_key
```

8. Prepare the environment.

Run the command:

```
sudo docker run -t --rm -v /install:/data --net=host ibmiot/ife.installer prepareEnv
```

Note: The prepare environment does:

- the configuration of the Docker engine.
- the update of the /etc/hosts environment.
- stops the iptables and firewalld services.
- restarts the docker service.

Important: If there are any errors in the configuration information the nodes.ini file, for example the incorrect the IP address, host name, or network interface, that has been entered, then you must manually log into each node, clear the host name mapping that has been added in the /etc/hosts file and correct the nodes.ini file and run the prepareEnv command again.

For Redhat and Centos operating systems, run following command to ensure the "--storage-opt dm.basesize=500G" setting in the /usr/lib/systemd/system/docker.service file is active, should able to see there are 500G for / directory.

```
sudo docker run --rm ibmiot/ife.installer df -h
```

If not, you must delete and reload the installer image to make sure that the docker engine setting works for all containers.

```
sudo docker rmi ibmiot/ife.installer;  
sudo docker rm installer  
sudo docker rmi ibmiot/ife.installer;  
sudo docker rmi ibmiot/ife.installer:v1.0  
sudo gunzip -c /tmp/installer.tar.gz | sudo docker load
```

Then run bellow command to check again.

```
sudo docker run --rm ibmiot/ife.installer df -h
```

Note: If you do not have iptables or firewalld installed, you can ignore the error iptables or firewalld are missing.

Note: If you receive the error "error getting events from daemon: unexpected error", you have executed the prepareEnv command more than once and you can ignore this error.

Installing the IBM IoT for Energy and Utilities environment

This task involves following the menu driven installation tasks.

Before you begin

The procedure requires the ife_custom_ana_spss_service.war file that is delivered in the installation package.

About this task

The menu items must be completed in sequential order:

- 0-Precheck the env...
- 1-Encrypt passwords in property file and generate runtime conf files...
- 2-Generate key files...
- 3-Deploy a private registry...
- 4-Set up calico network...
- 5-Deploy containers...

• 6-Updating calico network profile

Procedure

1. Run the command:

```
sudo docker run -it -v /install:/data  
--net=host --name=installer ibmiot/ife.installer setup
```

to return the menu with the steps 0 - 6.

2. Type 0 to check the environment.

Precheck the env checks the following conditions:

- The connections between the installation node and all other nodes.
- The Iptables service is not running on any node.
- Selinux is disabled.
- Docker is running on all nodes.

Note: If you receive an error, Failed to stop iptables.service: Unit iptables.service not loaded or, Selinux.etc is not loaded you can ignore these errors and go to the next step.

Important: If the precheck fails due to the incorrect information in the nodes.ini file, for example the incorrect IP address, or host name has been entered, then you must manually log into each node, clear the host name mapping that has been added in the /etc/hosts file and correct the nodes.ini file. You must run the prepareEnv command again in [Step 8, in Loading of the Docker image and preparing the installation environment.](#)

3. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

4. Type 1 to encrypt the passwords for all nodes and to generate the runtime configuration files.

If this step1 of the menu items fails the cause can be that the encryptKeyString is either too long or too short in length.

Change the encryptKeyString value to meet the length requirement, which is 16 characters in length. And rerun step1 of the menu item.

5. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

6. Type 2 to generate the keystore files for the Liberty server and the SPSS Modeler server to communicate with DB2 Server after SSL connection is enabled.

If this step2 of the menu items fails, the cause can be due to the password.DBSSLConnKeyPassword is not strong enough to generate an SSL key for DB2 communications. Find the latest backup file for the /install/conf/config.properties<timestamp> before encryption, and open the file to make sure this back up file has the modification you made and that the password strings are not encrypted. Overwrite the original config.properties in the /install/conf directory. For example:

```
sudo cp -rf /install/conf/config.properties<timestamp> /install/conf/config.properties
```

Change the password.DBSSLConnKeyPassword value in config.properties, then rerun from step 4, menu item 1.

7. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

8. Type 3 to deploy the private registry to all nodes.

The deploying of the private registry takes approximately 40 minutes to complete.

9. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

10. Type 4 to pull calico related image, and to deploy on each node, also create a virtual network for the containers to communicate.

This step pulls the calico image from the registry and takes approximately 15 minutes.

If you see an error in this step, go to the other nodes and run the following command:

```
sudo calicoctl node status
```

The command checks that the STATE of all peers is up and that the INFO is Established to make sure that there is no error.

11. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

12. Type 5 to deploy the Docker containers.

The return message happens before the work on the middleware images is complete. You can continue to the menu item 6 while the menu item 5 is being completed.

13. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

14. Type 6 to update the network calico profile.

15. Do these steps if openId is enabled.

By default openId is disabled.

- a) These substeps are required to temporary disable the openId authentication to be able to check the application links. As openId authorization requires IBM to register the new application link in the IBM application as described in config.properties file openId might not work in the current status. Go to the AppNode and enter the IFEAppNode container:

```
sudo docker exec -it IFEAppNode bash
su - wlp
```

- b) Go to /opt/IBM/WebSphere/Liberty/usr/servers/framework_server and open server.xml.

- c) Comment out the line `<include location="{server.config.dir}/openId.xml" />`
For example:

```
<!-- <include location="{server.config.dir}/openId.xml" /> -->
```

- d) Restart the liberty server with the commands:

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server
/opt/IBM/WebSphere/Liberty/bin/server start framework_server
```

16. After about 10 minutes, make sure that the backend installation tasks are complete, check the application links.

Note: When installing a 1+2 configuration, more waiting time may be necessary.

`https://<node of the ip where IFEAppNode container is deployed on>:<exposed port for liberty server in IFEAppNode, default 9443>/ibm`. If you have Cognos BI, check the link `http://<node of the ip where IFECogNode container is deployed on>:<exposed port for cognos server in IFECogNode, default 9300>/bi`

When both links work well, go to the next step to run postConfig.

If the Cognos BI link does not work, refer to [“Cannot access the Cognos BI link” on page 52](#)

17. From the installation node, run the command to do the post configuration step:

```
sudo docker run -t --rm -v /install:/data --net=host ibmiot/ife.installer postConfig
```

This step may take 20 minutes to complete.

18. Login to App node and run the commands:

```
sudo docker exec -it IFEAppNode bash
```

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server  
chown -R wlp:wlp /opt/IBM/WebSphere/Liberty/usr/servers/framework_server  
su - wlp -c "/opt/IBM/WebSphere/Liberty/bin/server start framework_server --clean"
```

19. Go to the SPSS node, or App node for 3nodes topology, and run the commands.

```
sudo docker exec -it IFESpssNode bash
```

```
rm -f /opt/IBM/energy/AHI/SPSS_stream/stream/Common/*  
cp /opt/IBM/energy/AHI.bak/SPSS_stream/stream/Common/*  
/opt/IBM/energy/AHI/SPSS_stream/stream/Common/  
chown -R spss:spss /opt/IBM/energy/AHI /opt/IBM/energy/AIP
```

20. Save the ife_custom_ana_spss_service.war file on to the node with IFESpssNode container in the /tmp folder.

21. Login to the node with IFESpssNode container, and run the commands:

```
sudo docker cp /tmp/ife_custom_ana_spss_service.war IFESpssNode:/tmp
```

```
sudo docker exec -it IFESpssNode bash
```

```
cp /tmp/ife_custom_ana_spss_service.war /opt/IBM/WebSphere/Liberty/usr/servers/  
framework_server/apps/  
chown -R wlp:wlp /opt/IBM/WebSphere/Liberty/usr/servers/framework_server  
su - wlp -c "/opt/IBM/WebSphere/Liberty/bin/server stop framework_server"  
su - wlp -c "/opt/IBM/WebSphere/Liberty/bin/server start framework_server --clean"
```

22. Do these steps if you have temporarily disabled openId authentication in [step 15](#).

a) Open /opt/IBM/WebSphere/Liberty/usr/servers/framework_server and open server.xml.

b) Restore the line `<include location="{server.config.dir}/openId.xml" />`:

For example:

```
<include location="{server.config.dir}/openId.xml" />
```

c) Restart the liberty server with the commands:

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server
```

```
/opt/IBM/WebSphere/Liberty/bin/server start framework_server
```

Validating the install process

After the Docker installation of IBM IoT for Energy and Utilities is complete, you must validate the install process.

About this task

To do the validation, check the functionality of the links to the application via the node where the IHSNode depolyed and to the HDP cluster via the Ambari interface. You also check that the HDP deploy process is complete.

Procedure

1. From the IHSserver, type the link to the application `https://<node of the ip where IFEIHSNode is deployed on>/ibm`.
The application should open.
2. From the IHSserver, type the link to the application `https://<node of the ip where IFEIHSNode is deployed on>/bi`.
The application should open.
3. Checks for the Connectivity Model application.
 - a) Check that the HDP cluster had deployed correctly by following command from the installer node:

```
sudo docker run -t --rm -v /install:/data --net=host ibmiot/ife.installer verifyInstall
```
 - b) Use a browser to access the link to the HDP cluster: `http://<node of the ip where ambari master is deployed on>:<exposed port, default 8080>`.
All the services except the SmartSense should be started.

Uninstall if the install fails

The uninstall procedure is described if the install process does not complete.

About this task

Do these steps to re-install IBM IoT for Energy and Utilities.

Procedure

1. Make a back up of the `/conf` folder in another location, for example `/tmp`.
Use the command:

```
sudo cp -r /install/conf /tmp
```
2. Run the following three commands to uninstall, delete installer container and to remove the `/install` folder:

```
sudo docker run -t --rm -v /install:/data --net=host ibmiot/ife.installer uninstall
```

```
sudo docker rm -f installer
```

```
sudo rm -rf /install
```
3. Re-start the install steps from [Loading the Docker Image and Preparing the Installation Environment](#).
Important: In place of [Step 5](#) copy the backup `/conf` folder from `/tmp` and overwrite the new one. You do not have to modify the files again.
Note: [Step 8](#) is not required to be done again.

Customizing the standard configuration of nodes, slave nodes and HDP client nodes

You can install IoT for Energy and Utilities with a system configuration that is custom built to meet your needs.

These sections give the necessary example setting to change the `*.ini` files.

Settings to install IoT for Energy and Utilities on more host nodes

Open the `/install/conf` configuration folder.

Open the `nodes.ini` file in a text editor.

In the file add the details of the additional host node as in the following example:


```
[<newnode>_name]
name=<newnode>
hostname=<hostname of the new node>
ip=<IP address of the new node>
networkInterface=<network interface of the new node>
```

When complete go back to [Step 5](#) of Loading of the Docker image and preparing the installation environment.

Settings to install with more HDP slave nodes

You need to follow "Settings to install IoT for Energy and Utilities on more host nodes" to add a new node, then use this section to configure the new HDP slave node on the new host node.

Open the `/install/conf` configuration folder.

Open the `containers.ini` file.

Paste the example text and change the section name, container name, target, and the external newPort. Make sure the newPort value is NOT used on the target host.

```
[<slavenew>_container]
name=<slavenew>
target=<newnode>
image=ibmiot.ife.registry:5000/ibmiot/ambari-agent
network=ife
volume=[<slavenew>_conf:/usr/hdp;<slavenew>_log:/var/log;<slavenew>_data:/hadoop]
volumesFrom=[]
privileged=true
env=[
    LDAP_SERVER_HOST:@{containers.ldap_container.name}.@{network.ife_network_info.name};
    AMBARI_SERVER_ADDR:@{containers.ambari_container.name}.@{network.ife_network_info.name};
    HDP_REPO_HOST:@{containers.repo_container.name}.@{network.ife_network_info.name};
    KDC_SERVER_HOST:@{containers.kdc_container.name}.@{network.ife_network_info.name};
    encryptKeyString:@{encryptKeyString};
    ldapServerPassword:@{password.ldapServerPassword}
]
extraParameters=[]
exposePorts=[<newPort>:1022]
```

In the `config.properties` file, add the new slave container hostnames for the `hdp.slave02` property. Old:

```
hdp.slave02=(slave02.ife,slave03.ife)
```

New:

```
hdp.slave02=(slave02.ife,slave03.ife,<slavenew>.ife)
```

When complete go back to [Step 5](#) of Loading of the Docker image and preparing the installation environment.

Settings to install more HDP client nodes

You need to follow "Settings to install IoT for Energy and Utilities on more host nodes" to add a new node, then use this section to configure the new HDP client node on the new host node.

Open the `/install/conf` configuration folder.

Open the `containers.ini` file.

Paste the example text and rename the section name, container name, and target.

```
[<client>_container]
name=<client>
target=<newnode>
image=ibmiot.ife.registry:5000/ibmiot/ambari-agent
network=ife
volume=[<client>_conf:/usr/hdp;<client>_log:/var/log;<client>_data:/hadoop]
volumesFrom=[]
```

```

privileged=true
env=[
    LDAP_SERVER_HOST:@{containers.ldap_container.name}.@{network.ife_network_info.name};
    AMBARI_SERVER_ADDR:@{containers.ambari_container.name}.@{network.ife_network_info.name};
    HDP_REPO_HOST:@{containers.repo_container.name}.@{network.ife_network_info.name};
    KDC_SERVER_HOST:@{containers.kdc_container.name}.@{network.ife_network_info.name};
    encryptKeyString:@{encryptKeyString};
    ldapServerPassword:@{password.ldapServerPassword};
]
extraParameters=[]
exposePorts=[]

```

In the config.properties file, add the client container hostnames for hdp.client01 property.

Old:

```
hdp.client01=()
```

New:

```
hdp.client01=(<client>.ife)
```

When complete go back to [Step 5](#) of Loading of the Docker image and preparing the installation environment.

Post installation for the Connectivity Model application

Configuring the Connectivity Model

The memory for the YARN container must be increased from the default setting.

About this task

Procedure

1. Access the link with a browser: `http://<Ambari node>:8080/#/main/services/YARN/configs`.
2. In the **Memory > Node** window, increase the memory on the nodes for the YARN containers. This value that depends on the memory configuration of the slave nodes.
For example: If the slave nodes have 64 GB memory one option is to configure this value to 48000 MB.
3. Click **Save** and type the reason for example Increase memory and click **Save**.
4. Click **Restart > Restart All Affected > Confirm Restart All** to restart the Ambari service.
5. These substeps are for if you want to make use of the HDP Smart Sense service
 - a) In the Ambari user interface click **Smart Sense > Activity Analysis**
 - b) Type the password twice for the user admin
By default the password is not set.
 - c) Click **Save** and **OK**.
 - d) Select **Service > Actions Start > Confirm restart > OK**.
 - e) Close the Ambari interface.

Configuring the automatic restart time for the liberty server

The liberty server `framework_server` must restart every 24 hours to refresh the kerberos authentication. The refresh maintains the visibility of data in the user interface of the Connectivity Model.

About this task

The restart time can be setup at mid-night when you are not using the liberty server.

Procedure

1. Login to the AppNode and open the IFEAppNode Docker container.

```
sudo docker exec -it IFEAppNode bash
```

2. Run the command with user `root`:

```
crond
```

3. Change the user to `wlp`

```
su - wlp
```

4. Create a file `/home/wlp/cron_restartLiberty.cron`.

5. Add the contents to the `cron_restartLiberty.cron` as in the example:

The example assumes you are going to start the `framework_server` at 1:00 AM each day. You can change the time to one of your choice.

```
00 01 * * * /opt/IBM/WebSphere/Liberty/bin/server stop framework_server;  
/opt/IBM/WebSphere/Liberty/bin/server start framework_server --clean
```

The example shows that at 00 (minutes) of 01(hour), *(everyday), *(every month), *(every day of week), the `framework_server` stops and starts with the command.

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server;  
/opt/IBM/WebSphere/Liberty/bin/server start framework_server --clean
```

.

Note: For the time format of the crontab file, please refer to the Linux crontab guide.

6. Apply the cron job with the commands:

```
cd /home/wlp
```

```
crontab cron_restartLiberty.cron
```

If the crontab does not work due to file permissions, then see the troubleshooting section, [“The cron job to restart liberty server framework_server did not work due to file permissions”](#) on page 53

Configuring Zeppelin Notebook, optional configuration

By default, Jupyter notebook is installed and configured for the Connectivity Model application. If you want to use the Zeppelin notebook, you can follow these steps to do the manual configuration.

About this task

Apache Python is the default language for the data ETL process and data mining in Connectivity Model therefore you must install Apache Python interpreter to the Apache Zeppelin notebook.

The logic for the use of Zeppelin notebook is:

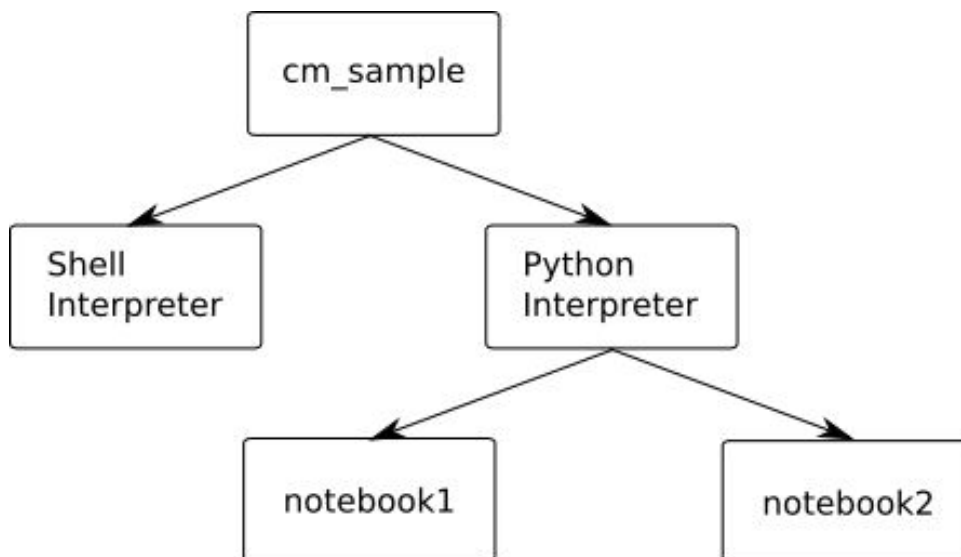


Figure 8. The logic of Zeppelin notebook with Spark

Start Spark for the Apache Python interpreter.

- SparkContext as sc.
- SQLContext as sqlContext. Spark can start one SparkContext.

All notebooks are created with the Apache Python interpreter and share Spark.

Procedure

1. From the Notebook node, type the link: `http://<Notebook node IP>:9995` with administrative credentials to open Zeppelin Notebook.
2. Click **Admin > Interpreter**
3. Scroll down the page to the shell entry `sh%sh` and click **Edit**.
4. Set the Option to: The interpreter is instantiated Per User in Isolated Per User process.
5. Check **User Impersonate**.
6. In the `shell.command.timeout.millisecs` field, increase the value to `600000000`.
7. Click **Save**.
8. Download the Apache Python interpreter installation package from `http://archive.apache.org/dist/zeppelin/zeppelin-0.7.0/zeppelin-0.7.0-bin-all.tgz`.
9. In the Ambari user interface, click **Zeppelin Notebook > Stop**.
10. Log in to the Notebook node and run the command:

```
sudo docker exec -it notebook bash
```

11. Copy the download files for Apache Python interpreter from the downloaded installation package to the notebook node directory `/usr/hdp/current/zeppelin-server/interpreter/python` and run the command:.

```
sudo docker cp /tmp/<package_folder_name>/interpreter/python notebook:
/usr/hdp/current/zeppelin-server/interpreter
```

Note: The group and user of the files must be zeppelin user.

12. In the Ambari user interface, click **Zeppelin Notebook > Start**.
13. In the Zeppelin web interface, click **cm_sample > Interpreter > Create**.
The **Create new interpreter screen** opens.
14. Type Python in the **Interpreter name** field.

15. Select python in the **Interpreter group** field.
16. In the **Options** field select *Per User isolated* and *User impersonate*
The text reads: The interpreter will be instantiated *Per User* in *isolated* process + *User Impersonate*.
17. Open Apache Zeppelin and create a new notebook with the Apache Python interpreter with the name **Spark initialization**.
18. Add the following code to the cell and run the cell. It initializes Spark with SparkContext and SQLContext.

```

from pyspark import SparkContext
from pyspark import SparkConf
from pyspark.sql import SQLContext
import sys
import os

sys.path.append("/usr/lib/python2.7/site-packages")
sys.path.append("/usr/hdp/current/spark-client/python")
sys.path.append("/usr/hdp/current/spark-client/python/lib/py4j-0.9-src.zip")

os.environ['SPARK_HOME'] = "/usr/hdp/current/spark-client"
os.environ["PYSPARK_SUBMIT_ARGS"] = "--keytab /etc/security/keytabs/cm_sample.keytab --principal cm_sample pyspark-shell --packages com.databricks:spark-csv_2.10:1.3.0"

conf = SparkConf().setMaster("yarn-client").setAppName("Spark - cm_sample")
sc = SparkContext(conf=conf)
sc.setLogLevel("ERROR")
sqlContext = SQLContext(sc)

```

19. Create more notebooks, for example spark notebook1, spark notebook2.
You have SparkContext as **sc**, SQLContext as **sqlContext**. You can do analyzes with Spark, reading the parquet files on HDFS.
20. To close Spark, open the **Interpreter** menu in Zeppelin web interface, and click **restart**.

Testing the Connectivity Model End to End for cm_sample (Optional)

If you want to do an end to end test for connectivity model you can do these steps.

About this task

This task does an end to end test for the Connectivity Model application. The test creates a sample tenant **cm_sample**, for Jupyter notebook with port 8888 for the **cm_sample**.

The test loads the sample data, does ETL and run analysis for **cm_sample**, and gives the UI users Bob, **user1**, and **user2** the permission to access **cm_sample**.

Procedure

1. Login to notebook container.
2. Run the command with root:

```
/home/cmopsadmin/cm/sample/bin/jupyter_test.sh
```

Upgrading from IBM IoT for Energy and Utilities version 2.5.0.x to version 2.8.0. for a Docker environment

Preparing the installation node for upgrading a Docker environment

The procedure contains the information to prepare the installation node for the upgrade of IBM IoT for Energy and Utilities that uses the Docker installation environment from version 2.5.0 to version 2.8.0.

Before you begin

You need the following files from the download package.

- `install28.tar.gz`
- `docker-upgrade28.tar.gz`

You must run the `postConfig` step from the 2.5.0 installation before you can start the upgrade to version 2.8.0.

From the installation node, run the command:

```
sudo docker run -t --rm -v /install:/data --net=host ibmiot/ife.installer postConfig
```

Procedure

1. Log in to the installation node as root user or the user with sudo permission as for the previous installation.
2. Copy the files `install28.tar.gz` and `registry-data-full28.tar.gz` to the `/tmp` directory on the installation node.
3. Unpack the installation image and load the image ready for the installation. Run the command:

```
sudo gunzip -c /tmp/installer28.tar.gz | sudo docker load
```

Note: The installation image is contained in the files that are located in the `/tmp` directory.

4. Create the installation folder `install2.8` and mount the configuration.

Note: If you have the installation folder for the previous release, make a backup of that directory.

Run the commands:

```
sudo mkdir -p /install2.8
```

```
sudo docker run --rm -v /install2.8:/data --name=installer2.8  
ibmiot/ife.installer /bin/bash -c "cp -r /install/conf  
/data;mkdir -p /data/images;mkdir -p /data/ssh_key"
```

5. Modify the configuration files in the folder `/install2.8/conf` for the correct topology. The configuration files are `config.properties`, `containers.ini`, and `nodes.ini`.
 - a) For 1 or 3 node topology only, copy the `containers.ini` file from `/install2.8/conf/topology3nodes` to `/install2.8/conf/`.
 - b) For all topologies, copy the `config.properties`, `nodes.ini`, `registry.ini`, and `network.ini` files from previous installation location `/install/conf` to `/install2.8/conf`.
 - c) For all topologies, if the previous installation is either 2.5 or 2.5.0.1 add this property to the `/install2.8/conf/config.properties` file so that the installation user has sudo permission.

```
installUser=root
```

- d) For topologies other than 1 or 3 node, make sure the target properties for **jena_artifacts_container** and **jena_base_container** in the `/install2.8/conf/container.ini` and `delta_25_to_28/conf/containers_all.ini` files are the same as the properties in the `/install/conf/containers.ini` file.
- e) For a customized `containers.ini` file, if you customized the `containers.ini` file in `/install/conf` for the current installation, you must customize the `containers.ini` file in the `/install2.8/conf` folder and the `*.ini` files in the `delta_25_to_28/conf` folder in the same

way. If you use the 1 or 3 node topology, you must customize the `delta_25_to_28/conf/3nodes_*.ini` files. For all other configurations, you need to customize the `delta_25_to_28/conf/containers_all.ini` and `delta_25_to_28/conf/delta_containers.ini` files.

6. Copy the ssh private key to the `/install2.8/ssh_key` folder.

Run the command:

```
sudo cp ~/.ssh/docker.id_rsa /install2.8/ssh_key
```

If the `docker.id_rsa` folder does not exist in `~/.ssh`, run the following command:

```
sudo cp ~/.ssh/docker-machine_rsa /install2.8/ssh_key
```

7. Run the command:

```
sudo docker run -it -v /install2.8:/data --net=host --name=installer2.8 ibmiot/ife.installer  
setup
```

Type 0 to check the environment.

8. For sudo users only. Open the `installer2.8` and run the following commands to do the changes:

```
sudo docker start installer2.8  
sudo docker exec -it installer2.8 bash  
cd /install/scripts/containerSetup
```

Edit the `Containers.py` file in the `containerSetup` folder to add "sudo" before "docker commit" for the `commit` function in the example:

```
def commit(self, name, reMout,...)  
    ...  
    cmd ="sudo docker commit " + name + " " + reMout...
```

Finalizing the upgrade for a Docker environment

The finalizing process for completing the upgrade of IBM IoT for Energy and Utilities for a Docker environment.

Before you begin

Make sure that enough disk space left for upgrading so that the generated temporary files do not use up all the disk space. You must have at least 50 GB of free disk space for the upgrade.

Procedure

1. Verify the environment.

- a) Make sure all of the Docker containers can start and stop successfully.

The Docker environment must have a stable status. Otherwise, the upgrade can fail caused by an inconsistent status.

- b) Make sure **IPv4 forwarding** is enabled by running the following command.

```
sysctl net.ipv4.ip_forward
```

Note: If IPv4 forwarding is disabled, the network does not work.

If IPv4 forwarding is still disabled, then issues with the Calico network setup exist. Refer to the next step.

If issues with calico network exist, follow the steps:

- c) Make sure that calico network **ife** works as expected:

Run the command:

```
docker network ls
```

The correct result is **Ife**. If issues with the Calico network still exist, do the steps:

- 1) Run the following two commands on each host.

```
docker rm -f etcd calico-node
docker volume rm etcd-data
```

- 2) Go to installation node to create a Calico node:

```
docker start -i installer
```

Do the steps 1, 2, and 4 of the installer.

- 3) Restart the Docker service:

```
systemctl restart docker
```

2. Save the `registry-data-full128.tar.gz` from the download package to the `Deployment/delta_25_to_28` folder.
3. Prepare the upgrade topology.

If you want to use the default 3 node topology that includes Connectivity Model or one node topology only for Asset Performance Management, change the setting **topology_1_or_3_nodes=false** to **topology_1_or_3_nodes=true** in the `docker_delta_25_to_28.sh` file.

By default, **topology_1_or_3_nodes** is set as false. In this case the following two `.ini` files are used in the course of upgrade:

- `containers_all.ini` For all containers.
- `delta_containers.ini` Includes the containers that are to be upgraded.

Note: These two `.ini` files are provided as the templates. When you create your own topology for your own needs, you can overwrite the two files.

4. Optional: If you did not adopt the suggested installer name and you put the installer into another location.

You must export the two **env** variables before you do the upgrade:

- **export INSTALLER=<installer name>**
- **export INSTALL_PATH=<your path>**

By default the installer uses the following settings:

- **export INSTALLER=installer2.8**
- **export INSTALL_PATH=/install2.8**

5. Run the command with `sudo` or root permissions to do the upgrade.

```
sudo ./docker_delta_25_to_28.sh
```

Finalizing the upgrade for the Connectivity Model application

These steps are only applicable for the Connectivity Model application.

About this task

If you plan to upgrade for Connectivity Model, make sure `delta_containers.ini` contains **cm_backend_container** and **notebook_container**.

Procedure

1. Back up the Connectivity Model etl and other update codes.
We update all Connectivity Model codes, and backup them to `/tmp/bk_cm_delta_28`.
2. Run the command on the notebook host to update Connectivity Model.


```
sudo docker exec -it notebook bash -c "/build/install_artifact.sh"
```

3. You can rerun the Connectivity Model analysis for each tenant to update the analysis results.
4. If you still have the `cm_sample` tenant, and you want to rerun the analysis with the new sample data. Do the steps:
 - a. Go to the node with notebook container, and run the command:

```
sudo docker exec -it notebook bash
```

- b. Edit the `/home/cm_sample/conf/tenant.cfg` and set:

- **hbase_clean=true**, the property cleans old data.
- **analysis_voltage_duration=90**
- **analysis_voltage_until_time=2017-12-31T00:00:00**

The two properties are to use the new sample data date range.

- c. Run the following commands:

```
su - cm_sample
kinit -k -t /etc/security/keytabs/cm_sample.keytab cm_sample
hdfs dfs -rm -r -f cm/raw/*
hdfs dfs -rm -r -f cm/ana/*
hdfs dfs -rm -r -f cm/tmp/*
hdfs dfs -rm -r -f cm/job/*
/home/cm_sample/sample/bin/jupyter_test_1_sample_data.zip.sh
/home/cm_sample/sample/bin/jupyter_test_2_etl.sh 2017-10-01 2017-10-01 2017-12-31 90
/home/cm_sample/sample/bin/jupyter_test_3_analysis.sh 2017-12-31 90
d) Run this command with root user
/home/cmopsadmin/cm/bin/APP_updateMapCenter.sh cm_sample
```

Upgrading from IBM IoT for Energy and Utilities version 2.5.0.x to version 2.8.0. for a non-docker environment

The steps for upgrading from version 2.5 to version 2.8 are different according to the application that you use.

For Asset Performance Management and Asset 360 for Wind, you can upgrade from version 2.5.1 to version 2.8.0.

To be able to use the latest Connectivity Model, you must first upgrade from version 2.5.0 to version 2.5.0.1. If you are at version 2.5.0.1, you can upgrade straight to 2.5.0.

Upgrading from IBM IoT for Energy and Utilities version 2.5.1 to version 2.8.0. for a non-Docker environment

Use the following steps to upgrade IBM IoT for Energy and Utilities with the Asset Performance Management and Asset 360 for Wind applications.

Before you begin

The actions that must be completed before you can start the upgrade:

- The command line calculator must be available on each node. If the tool is not installed, use the command to install it.

```
yum install bc
```

- Make a note of the IP address and host name of each node. The IP address and host name are in the `/etc/hosts` directory.

- On the App of SPSS nodes, check the presence of `wlp` user. If `wlp` user is present, remove it with the command.

```
userdel -rf wlp
```

The installation file that you need is `IBM_IOT4EU_Upgrade_V2.8_LINUX_ML`, part number `CNV1CML` from the download package. The upgrade files are located in the `upgrade251To28` directory.

Important: The upgrade scripts must run under the root user account.

Make a backup of all corresponding directories before you start to upgrade the DB node, App node, and SPSS node.

Procedure

1. Upgrade the DB node.

- a) Run the following commands in turn to back up the database:

```
su - db2inst1
```

```
db2stop force
```

```
db2start
```

```
db2 backup db IFEDB
```

Note: If you get an error during upgrade or the upgrade did not complete, run the following commands to restore the database:

```
su - db2inst1
```

```
db2 restore db IFEDB
```

Run the following command when the database has the status `rollforward pending` after the restore.

```
db2 rollforward db IFEDB
```

- b) Run the following steps under root user account:

- 1) Create the directory `/build/IFE/DB` if it does not exist.
- 2) Extract the DB upgrade package `s1_db_upgrade.tar.gz` to the `/build/IFE/DB` directory.
- 3) Go to the `/build/IFE/DB` directory.
- 4) Run the command:

```
./installArtifacts2.8.sh 2>&1 | tee /tmp/installArtifacts2.8.log
```

- c) Verify the completion of the upgrade:

- 1) Open the log file `installArtifacts2.8.log`, and make sure errors are not generated during the running of the command `installArtifacts2.8.sh`.
- 2) Check the tables are created successfully by running the following commands:

```
su - db2inst1
```

```
db2 connect to IFEDB
```

```
db2 list tables for schema VI
```

```
db2 list tables for schema CUSTOMANA
```

Then, the tables VISUAL_INSPECTION, VI_CONFIG, and HISTORY are listed in the example figure.

```
[db2inst1@pmo-db root]$ db2 list tables for schema VI
Table/View              Schema      Type      Creation time
-----
VISUAL_INSPECTION      VI          T         2018-08-02-15.12.46.188932
VI_CONFIG              VI          T         2018-08-02-15.12.47.261357

  2 record(s) selected.

[db2inst1@pmo-db root]$ db2 list tables for schema CUSTOMANA
Table/View              Schema      Type      Creation time
-----
HISTORY                CUSTOMANA  T         2018-08-02-15.12.39.812509

  1 record(s) selected.
```

Figure 9. Checking the making of tables

2. Upgrade the SPSS node.

a) Make a backup of the /opt/IBM/energy/AHI directory.

b) If SSL is enabled, prepare the keystore file for the Db2 connection.

- 1) Check whether the Liberty Server connects to Db2 with SSL enabled by referring to the file /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server_ife_frwk.xml on the App node. If you find the following values ssl***** in the server_ife_frwk.xml file, then SSL is enabled for Db2. You must prepare the keystore file.

```
<dataSource id="ifedb" ...
<properties.db2.jcc databaseName="{db2databaseName}" ...
  sslConnection="true"
  sslTrustStoreLocation="/home/db2inst1/ssl/testKeystore"
  sslTrustStorePassword="testonly"/>
</dataSource>
```

Otherwise, SSL is not enabled for Db2 connection, and you do not need to prepare the keystore file.

- 2) If SSL is enabled, copy the keystore file /home/db2inst1/ssl/testKeystore from the App node to the SPSS node.

c) Run these steps as root user:

1) If the /build/IFE/Spss directory does not exist, you must create it.

2) Extract the SPSS upgrade package sl_spss_upgrade.tar.gz to the /build/IFE/Spss directory.

3) Go to the /build/IFE/Spss directory.

4) Edit the env.inc file to have all of the necessary parameters set correctly.

a) Set **run_as_wlp_user** to 0

b) Set **spssUser** to **root**, and set **spssUser_Password** as the password of root.

c) If SSL is enabled, set **DBSSLKeystoreLocation** to be the location of the keystore file copied from the App node. The **DBSSLKeystorePassword** must be consistent.

d) The value of **httpsPort_server** must be consistent with the value that is specified in server.xml in the App node. In the App node, if the value of **httpsPort** is set to **9449** in

the `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server.xml` file as follows, then the value of `httpsPort_server` must also be set to **9449**.

```
<httpEndpoint id="defaultHttpEndpoint"
              host="*"
              httpPort="9086"
              httpsPort="9449" />
```

e) Source `env.inc` by running the following command:

```
. env.inc
```

d) Run the following command to upgrade:

```
./installArtifacts2.8.sh 2>&1 | tee /tmp/installArtifacts2.8.log
```

e) After the upgrade is complete, copy the directory `SPSS_stream/stream/Common` from the backup directory for `/opt/IBM/energy/AHI` from step (a) back to the `/opt/IBM/energy/AHI` directory.

3. Upgrade the App node.

a) Make a backup of the `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server` directory.

b) Run the following steps under root user account:

- 1) If the `/build/IFE/Liberty` directory does not exist, you need to create it.
- 2) Extract the App upgrade package `s1_app_upgrade.tar.gz` to the `/build/IFE/Liberty` directory.
- 3) Go to `/build/IFE/Liberty` directory.
- 4) Edit the `env.inc` file to have the necessary parameters set correctly.
- 5) Run the command:

```
./installArtifacts2.8.sh 2>&1 | tee /tmp/installArtifacts2.8.log
```

4. Upgrade the Cognos node.

a) Make sure that environment variable `LD_LIBRARY_PATH` is configured for the root user. If not, run the following commands to configure and then relogin:

```
export DB2_HOME=/opt/ibm/db2/V10.5
export CognosInstallLocation=/opt/ibm/cognos/analytics
export JAVA_HOME=$CognosInstallLocation/jre
echo "export LD_LIBRARY_PATH=$DB2_HOME/lib32:$DB2_HOME/lib64:
$CognosInstallLocation/bin64:$CognosInstallLocation/bin:$JAVA_HOME/bin" >> ~/.bash_profile
echo "export LD_LIBRARY_PATH=$DB2_HOME/lib32:$DB2_HOME/lib64:
$CognosInstallLocation/bin64:$CognosInstallLocation/bin:$JAVA_HOME/bin" >> ~/.bashrc
```

b) Copy the Cognos package `AH_Cognos_Artifact.zip` into Cognos node and extract it to the `/build/AH_Cognos_Artifact` directory.

c) Edit the script `deployReport.sh` and have the necessary parameters set correctly as follows:

```
CognosInstallLocation=/opt/ibm/cognos/analytics           # install location of cognos
COGNOS_DB2_REMOTE_HOSTNAME=DB.ibm.com                   # host name of DB node
# DB2 port, default value is 50000, if enabled SSL default value is 50005
COGNOS_DB2_REMOTE_PORT=50000
DB_NAME=IFEDB
# DB2 user name and password
DB_INSTANCE_USER=db2inst1
DB_PASSWORD=db2inst1
# need to specify the two below if DB SSL enabled
SSL_TRUST_STORE_LOCATION=/enableSSL/DBSSLKeystore
SSL_TRUST_STORE_PASSWORD=pw4ibmiotou

#auth info for cognos if needed
anonymousLoginFlag=true                                # need to be set to
true
#connection string for CLI and JDBC
CLI_CONN_URL=jdbc:db2://$COGNOS_DB2_REMOTE_HOSTNAME:$COGNOS_DB2_REMOTE_PORT/$DB_NAME
```

```
# JDBC connection URL for DB with SSL enabled. If DB SSL is not enabled, need to comment
this line by adding # at the beginning of the line.
JDBC_CONN_URL="jdbc:db2://$COGNOS_DB2_REMOTE_HOSTNAME:$COGNOS_DB2_REMOTE_PORT/
$DB_NAME;sslConnection=true;sslTrustStoreLocation=$
{SSL_TRUST_STORE_LOCATION};sslTrustStorepassword=${SSL_TRUST_STORE_PASSWORD};"
# JDBC connection URL for DB without SSL enabled. If DB SSL is enabled, need to comment
this line by adding # at the beginning of the line.
JDBC_CONN_URL=jdbc:db2://$COGNOS_DB2_REMOTE_HOSTNAME:$COGNOS_DB2_REMOTE_PORT/$DB_NAME
...
```

d) Edit the script `install_cognos_report.sh` and change **Cognos_user** to **root**.

e) Run the following command:

```
chmod a+x *.sh
```

f) Run script `install_cognos_report.sh` as root user to deploy and wait for the deployment to complete.

Note: The running of the script can take a few minutes.

g) When complete, verify the package is deployed with the following link.

`http://<ip address of cognos node>:9300/bi`

5. Optional: Import the Liberty server certificates on SPSS node into the Liberty server on the App node.

Note: If the certificates are different on the App node and SPSS node, do the following steps.

a) Run the following command to export the certificate from the Liberty server file `key.jks` on SPSS node keystore:

```
<jre_dir>/bin/keytool -export -alias default -storepass
passw0rd -keystore /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/resources/
security/key.jks
-file /home/certLiberty01
```

a) Copy the certificate file `/home/certLiberty01` from the SPSS node to the App node.

b) Run the following command to import certificate into keystore file `key.jks` into the Liberty server on App node:

```
echo "yes" | <jre_dir>/bin/keytool -import -trustcacerts -alias
libertyForANAFramework -file /home/certLiberty01 -keystore /opt/IBM/WebSphere/Liberty/usr/
servers/framework_server/resources/security/key.jks
-storepass passw0rd
```

If the truststore file `trust.jks` exists in App Node, you must also run the following command to import certificate into `trust.jks`:

```
echo "yes" | <jre_dir>/bin/keytool -import -trustcacerts -alias
libertyForANAFramework -file /home/certLiberty01 -keystore /opt/IBM/WebSphere/Liberty/usr/
servers/framework_server/resources/security/trust.jks
-storepass passw0rd
```

c) After you import the certificates, restart the Liberty server on the App node with the following command:

```
/opt/IBM/WebSphere/Liberty/bin/server start framework_server
```

Upgrading from IBM IoT for Energy and Utilities version 2.5 to version 2.5.0.1 for a non Docker environment

IoT for Energy and Utilities version 2.5 offers two methods to install the solution. By the use of Docker containers for a new installation, and by an upgrade from version 2.1 to version 2.5.

About this task

If you upgraded from IoT for Energy and Utilities version 2.1 to version 2.5 using the non Docker upgrade, to take advantage for the 2.5.0.1 you must use this procedure to do the upgrade.

Procedure

1. Copy and unpack the `IOT4EU_2501_non_docker.zip` to the APP node.

You see the folders:

- `cim`
- `frameworkArtifact`
- `cm`

2. Update the `frameworkArtifact` files:

a) Open the `frameworkArtifact` folder.

b) Open the `delta_install_framework_2.5.0.1.sh` file in a text editor and apply the correct liberty server name for the item `server`:

The default name is `framework_server`.

```
server=framework_server
```

c) Open the `monitorUser.properties` file in a text editor and apply the correct name for the items `logFolder` and `liberty user groups`.

The default names are:

```
groups_StandardUser=admins
```

```
groups_LimitedUser=users,director,operator
```

```
logFolder=/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/logs
```

d) Run the command as either root or sudo user:

```
chmod a+x delta_install_framework_2.5.0.1.sh
```

e) Run the command with the user who currently started the liberty server.

```
./delta_install_framework_2.5.0.1.sh
```

3. Update the CIM if you have the CIM installed for either Asset Performance Management or Asset 360 for Wind applications.

a) Open the `cim` folder.

b) Open the `delta_install_cim_2.5.0.1.sh` file with a text editor and apply the correct liberty server name for the item `Liberty_Server`.

The default value is:

```
Liberty_Server=framework_server
```

c) Run the command with either root or sudo user:

```
chmod a+x delta_install_cim_2.5.0.1.sh
```

d) Run the command with the user who currently started the liberty server.

```
./delta_install_framework_2.5.0.1.sh
```

4. Update the Connectivity Model if installed.

The prerequisites are:

- Python 2.7,
- Internet access.

Note: If you do not have internet access, you need to download and install the required packages manually.

a) Login in to the host you installed Connectivity Model application. In most case this is the APP node. If you have a separate notebook node, Connectivity Model should be installed there.

b) Run these commands from the root directory:

```
pip install --upgrade pip
```

```
pip install toree
```

```
jupyter toree install --spark_home=/usr/iop/4.2.0.0/spark/  
--spark_opts='--master=yarn-client'  
--interpreters=Scala,PySpark,SparkR,SQL
```

```
pip install numpy
```

```
pip install scipy
```

```
pip install matplotlib==2.1.2
```

```
pip install pandas
```

c) Backup the bin, etl, and sample folders in the Connectivity Model tenant user and cm home installation folders for example: /home/ibmife/cm if you want to preserve them, otherwise these folders will upgrade to IBM IoT for Energy and Utilities.

d) Copy the IOT4EU_2501_non_docker/cm folder on the host.

e) Open the IOT4EU_2501_non_docker/cm folder.

f) Run the commands with root:

```
chmod a+x /etc/rc.d/rc.local
```

```
cp -r ./jupyter /opt
```

```
chmod -R a+rx /opt/jupyter
```

g) Unpack IFE_CM_Artifact.zip.

h) Open the bin/Delta_update.sh file with a text editor.

- Update the element cm_home_directory with the actual value that Connectivity Model installed.
- Update the current_tenant element with the existing tenant users on the host, each tenant user being separated by a comma.
- Apply the correct liberty server name for the liberty_server element.

The default values are:

```
liberty_server=framework_server
```

```
cm_home_directory=/home/cmopsadmin/cm
```

```
current_tenant=(cm_sample)
```

For example, you can change the values to:

```
cm_home_directory=/home/ibmife/cm
```

```
current_tenant=(cm_sample,cm_tenant1)
```

i) Copy the conf/tenant.cfg file to \$cm_home_directory/conf.

- j) Open the `$cm_home_directory/conf/global.cfg` file, add the correct `python_lib` path in the file. For example, if the python 2.7 library is in `/usr/lib/python2.7`, then add this line:

```
python_lib=/usr/lib/python2.7
```

- k) Go to `IOT4EU_2501_non_docker/cm` directory, run the command with root or sudo user to update the Connectivity Model backend.

```
chmod a+x bin/Delta_update.sh
```

```
bin/Delta_update.sh notebook
```

- l) If the liberty server is on the host, you can run these commands to update Connectivity Model user interface. Otherwise, you need to copy `IFE_CM_Artifact.zip` to the APP node, and repeat steps “4.g” on page 43 and “4.h” on page 43 first, then run the below commands:

```
chmod a+x bin/Delta_update.sh
```

```
bin/Delta_update.sh app
```

Upgrade IoT for Energy and Utilities with Connectivity Model from version 2.5.0.1 to 2.8

About this task

If you upgraded from IoT for Energy and Utilities version 2.1 to version 2.5, and subsequently to 2.5.0.1 using the non Docker upgrade, to take advantage for the 2.8.0 you must use this procedure to do the upgrade.

Procedure

1. Login in to the host you installed for the Connectivity Model application. In most case this is the APP node. If you have a separate notebook node, Connectivity Model should be installed there.
2. Make a backup of the `/bin`, `/etl`, and `/sample` folders in the Connectivity Model tenant `/user` and `/cm` home installation folders. For example: `/home/cmopsadmin/cm` if you need to preserve the folder, otherwise they will be upgraded.
3. Get `IFE_CM_Artifact.zip` from `upgrade251To28` and put it on the node.
4. Unpack `IFE_CM_Artifact.zip` using the command:

```
unzip IFE_CM_Artifact.zip -d /tmp/cm
```

5. Open the `/tmp/cm/bin/Delta_update_2.8.sh` file with a text editor.
 - a) Update the element `cm_home_directory` with the actual value that Connectivity Model installed, by default Connectivity Model is installed in `/home/cmopsadmin/cm`.
 - b) Update the `current_tenant` element with the existing tenant users on the host, each tenant user being separated by a comma.
 - c) Apply the correct liberty server name to the `liberty_server` element.
 - d) Set the value of `docker_env` from `true` to `false`.

The default values are:

```
liberty_server=framework_server
cm_home_directory=/home/cmopsadmin/cm
current_tenant=(cm_sample)
docker_env=true
```

For example, you can change the values to:

```
liberty_server=framework_server
cm_home_directory=/home/cmopsadmin/cm
```



```
current_tenant=(cm_sample,cm_tenant1)
docker_env=false
```

6. Go to /tmp/cm directory, run the command with root or sudo user to update the Connectivity Model application.

```
dos2unix bin/Delta_update_2.8.sh
chmod a+x bin/Delta_update_2.8.sh
./bin/Delta_update_2.8.sh notebook
```

7. If the liberty server is on the host, you can run these commands to update Connectivity Model application user interface. Otherwise, you need to copy IFE_CM_Artifact.zip to the APP node, and repeat steps 4 and 5 first, then run the below commands:

```
dos2unix bin/Delta_update_2.8.sh
```

```
chmod a+x bin/Delta_update_2.8.sh
```

```
./bin/Delta_update_2.8.sh app
```

Note: You may need to rerun the Connectivity Model analysis for each tenant to update the analysis results.

8. Optional: If you still have cm_sample tenant, and want to rerun analysis with the new sample data. Do the following the steps:

- a) Edit the /home/cm_sample/conf/tenant.cfg file.

- 1) Set hbase_clean=true, the property cleans the old data.

- 2) Set analysis_voltage_duration=90 and analysis_voltage_until_time=0:00:00.

Note: The two properties are to use the new sample data date range.

- b) Run the following commands with root user.

```
su - cm_sample
```

```
hdfs dfs -rm -r -f cm/raw/*
```

```
hdfs dfs -rm -r -f cm/ana/*
```

```
hdfs dfs -rm -r -f cm/tmp/*
```

```
hdfs dfs -rm -r -f cm/job/*
```

```
/home/cm_sample/sample/bin/jupyter_test_1_sample_data_zip.sh
/home/cm_sample/sample/bin/jupyter_test_2_etl.sh 2017-10-01 2017-10-01 2017-12-31 90
/home/cm_sample/sample/bin/jupyter_test_3_analysis.sh 2017-12-31 90
```

- c) Run the command with root user to reset mapcenter for the new sample data.

```
/home/cmopsadmin/cm/bin/APP_updateMapCenter.sh cm_sample
```

Adding a new HDP slave node after installation

After you have completed the install of IBM IoT for Energy and Utilities 2.5.0.1 with Hortonworks Data Platform (HDP) services, you can add a new slave node to the HDP cluster.

Follow these steps to add a new HDP node.

Preparing for the new HDP slave node

After installing IBM IoT for Energy and Utilities with HDP services, you can add a new slave node to HDP cluster by following these steps.

About this task

You must do the same prepare procedures as the other nodes in the existing cluster. You do not need to create the install node again, the existing install node is used.

Procedure

1. Configure the node and install Docker CE on the new HDP node. Do this section: [“Preparing the target servers” on page 15](#)
2. Open the file `/usr/lib/systemd/system/docker.service` and find the line

```
ExecStart=/usr/bin/dockerd
```

and append the `--insecure-registry` setting, for example:

```
ExecStart=/usr/bin/dockerd --insecure-registry ibmiot.ife.registry:5000
```

3. Restart the docker with the commands:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

4. From the installer node, copy the key to the new node.

```
scp /root/.ssh/docker.id_rsa.pub root@<new_node_ip_address>:/tmp
```

Where `<new_node_ip_address>` is the IP address of new node.

If root is used, the command remains the same. If sudo user is used, the command is .

```
cd /home/<sudo user>; scp .ssh/docker.id_rsa.pub <sudo user>@<new_node_ip_address>:/tmp
```

5. On the new node, login with root or sudo user and run the command to add the key into the `authorized_keys` for that node. For example:

```
mkdir -p ~/.ssh;sudo chmod a+r /tmp/docker_id_rsa; cat /tmp/docker.id_rsa.pub >> ~/.ssh/authorized_keys;chmod 700 ~/.ssh; chmod 600 ~/.ssh/authorized_keys
```

6. On each existing node, open the `/etc/hosts` file and add the new node hostname and IP.
7. On the new node, open the `/etc/hosts` file and add the existing nodes hostname and IP.
8. Also add the `<IP> ibmiot.ife.registry` to the `/etc/hosts` file.
You can get the IP for the `ibmiot.ife.registry` from the other existing nodes.

Updating the current installer configuration

You need to update configuration and properties files on the install host.

About this task

All these files are on the install host.

Procedure

1. Open the `/install/runtime/conf/nodes.ini` and `/install/conf/nodes.ini` files and replace these elements with the actual values:
 - `<newnode>_name`,
 - `<new_node>`,

- <hostname of the new node>.
- <network interface of the new node>.

In these files add the details of the additional host node as in the following example:

```
[<newnode>_name]
name=<newnode>
hostname=<hostname of the new node>
ip=<IP address of the new node>
networkInterface=<network interface of the new node>
```

2. Open the `/install/conf/containers.ini` file and copy paste the `[slave02_container]` section and change:

- section name `<slavenew>_container`
- container name `<slavenew>`
- target `<newnode>`
- external port `<newPort>`

Important: Make sure the `<newPort>` value is not used on the target host. The `<newnode>` value is the same as step 1.

For example:

```
[<slavenew>_container]
name=<slavenew>
target=<newnode>
image=ibmiot.ife.registry:5000/ibmiot/ambari-agent
network=ife
volume=[<slavenew>_conf:/usr/hdp; <slavenew>_log:/var/log; <slavenew>_data:/hadoop]
volumesFrom=[]
privileged=true
env=[
    LDAP_SERVER_HOST:@{containers.ldap_container.name} @{network.ife_network_info.name};
    AMBARI_SERVER_ADDR:@{containers.ambari_container.name} @{network.ife_network_info.name};
    HDP_REPO_HOST:@{containers.repo_container.name} @{network.ife_network_info.name};
    KDC_SERVER_HOST:@{containers.kdc_container.name} @{network.ife_network_info.name};
    encryptKeyString:@{encryptKeyString};
    ldapServerPassword:@{password.ldapServerPassword}
]
extraParameters=[]
exposePorts=[<newPort>:1022]
```

3. Open the `/install/runtime/conf/container.ini` file and copy paste the `[slave02_container]` section and change:

- section name `<slavenew>_container`
- container name `<slavenew>`
- target `<newnode>`
- external port `<newPort>`

Important: Make sure the `<slavenew>`, `<newnode>`, `<newPort>` values are the same as step 2.

Note: Make sure the `ldapServerPassword` and `encryptKeyString` values are the same as the `[slave02_container]` section.

For example:

```
[<slavenew>_container]
name=<slavenew>
target=<newnode>
image=ibmiot.ife.registry:5000/ibmiot/ambari-agent
network=ife
volume=[ <slavenew>_conf:/usr/hdp; <slavenew>_log:/var/log; <slavenew>_data:/hadoop]
volumesFrom=[]
privileged=true
env=[
    LDAP_SERVER_HOST:IFEldapNode.ife;
    AMBARI_SERVER_ADDR:ambari.ife;
```

```

HDP_REPO_HOST:repo.ife;
KDC_SERVER_HOST:kdc.ife;
encryptKeyString:ibmioteuibmioteu;
ldapServerPassword:c759fd13f7c3cea9c29ef756bf39a7a0
]
extraParameters=[]
exposePorts=[<newPort>:1022]

```

Important: Make a record of the properties and values for later use.

4. Open the `/install/runtime/conf/config.properties` and `/install/conf/config.properties` files and add the new slave container name `<slavenew>` for `hdp.slave02` property.

Assume this is the previous property value:

```
hdp.slave02=(slave02.ife,slave03.ife)
```

then this is the new property value:

```
hdp.slave02=(slave02.ife,slave03.ife,<slavenew>.ife)
```

Adding the new node to the existing etcd and calico network

You must add the new HDP node to the existing etcd and calico network.

Procedure

1. From the installer node, run the command:

```
sudo docker exec -it etcd sh
```

2. From the etcd container, run the command, replace `<hostname of the new node>` with the actual value. For example if there are 12 nodes in the cluster, including the new nodes, the index should be 12.

```
etcdctl member add etcd<index> http://<hostname of the new node>:2380
```

`<index>` is the total amount of the nodes.

For example if there are 12 nodes in the cluster, including the new nodes, the index must be 12.

3. In this step, the example uses a three node cluster plus the one new node and assumes the existing cluster hostnames are **host1.ibm.com**, **host2.ibm.com**, and **host3.ibm.com**. The new node hostname is **host4.ibm.com**.

- a) Run the command:

```
etcdctl member add etcd4 http://host4.ibm.com:2380
```

- b) Make sure the output is similar to:

```

ETCD_NAME="etcd4"
ETCD_INITIAL_CLUSTER="etcd4=http://host4.ibm.com:2380,
etcd3=http://host3.ibm.com:2380,etcd2=http://host2.ibm.com:2380,etcd1=http://
host1.ibm.com:2380"

```

- c) Make a record of the value `ETCD_INITIAL_CLUSTER` that will be used in next step.
4. On the new node, run this commands to create the new etcd container.

Make sure initial-cluster value is the same as the output of step 3. Here is the example:

```
index=4
```

```
newhost=host4.ibm.com
```

```
sudo docker run -d -p 2380:2380 -p 2379:2379 --name etcd --restart
always --net=host --volume=/data/etcd:/etcd
```

```
ibmiot.ife.registry:5000/quay.io/coreos/etcd:v3.2.4
/usr/local/bin/etcd --data-dir=/etcd-data
--name etcd${index}
--listen-peer-urls http://${newhost}:2380
--initial-advertise-peer-urls http://${newhost}:2380
--listen-client-urls http://${newhost}:2379
--advertise-client-urls http://${newhost}:2379
--initial-cluster "etcd4=http://host4.ibm.com:2380,
etcd3=http://host3.ibm.com:2380,etcd2=http://host2.ibm.com:2380,etcd1=http://
host1.ibm.com:2380"
--initial-cluster-state existing
--initial-cluster-token my-etcd-token
```

5. Verify the etcd cluster,

a) From the new node run the command:

```
sudo curl -L http://127.0.0.1:2379/v2/members
```

You will see all the members in the cluster.

b) From the installer node run this command in the etcd container:

```
etcdctl cluster-health
```

The new etcd member shows healthy as do the other members.

6. Start the calico1 container. Run the commands on the new node and replace `<newnode_ip>` with the actual value.

```
sudo docker pull ibmiot.ife.registry:5000/quay.io/calico/node:v2.4.0
```

```
sudo calicoctl node run --ip=<newnode_ip> --node-image=ibmiot.ife.registry:5000/quay.io/
calico/node:v2.4.0
```

7. From an existing node, copy the `/usr/local/bin/calicoctl` to the new node and put it the `/usr/local/bin` location.

8. Run the command on the new node.

```
sudo chmod a+x /usr/bin/calicoctl
```

9. Verify the status of the calico nodes.

a) Run this command on the new node

```
sudo calicoctl node status
```

b) Run the same command on the install node.

```
sudo calicoctl node status
```

All status of all nodes should be up.

10. Replace the `docker.service` file on the new node.

a) On the new node, make a back-up of the `/usr/lib/systemd/system/docker.service` file.

b) From an existing node, copy the `/usr/lib/systemd/system/docker.service` file and replace the file on the new node.

c) Edit the new `/usr/lib/systemd/system/docker.service` file to make sure the hostname in the `ExecStart` line is the new node hostname.

For example:

```
ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2376
-H unix:///var/run/docker.sock --storage-driver devicemapper
--insecure-registry ibmiot.ife.registry:5000
--cluster-store etcd://host4.ibm.com:2379
--cluster-advertise ens192:2376
--storage-opt dm.basesize=500G
```

d) Restart docker services on the new node with the commands:

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

11. Do steps “5” on page 49 and “9” on page 49 to verify the etcd cluster and the status of the nodes.

Starting the ambari-agent container on the new node

The ambari-agent container must be started on the new node.

Procedure

1. From the new node, run the command:

Replace the `<slavenew>`, `<encryptKeyString>`, `<ldapServerPassword>` values that you recorded from step “3” on page 47 of the Updating the current installer configuration procedure. Make sure all other variables are the same value as step 3 too.

```
image=ibmiot.ife.registry:5000/ibmiot/ambari-agent
container_name=<slavenew>
network=ife
LDAP_SERVER_HOST=IFELdapNode.ife
AMBARI_SERVER_ADDR=ambari.ife
HDP_REPO_HOST=repo.ife
KDC_SERVER_HOST=kdc.ife
encryptKeyString=<encryptKeyString>
ldapServerPassword=<ldapServerPassword>

sudo docker pull ${image}
sudo docker volume create ${container_name}_conf
sudo docker volume create ${container_name}_log
sudo docker volume create ${container_name}_data

sudo docker container create --privileged=true -v
${container_name}_conf:/usr/hdp -v
${container_name}_log:/var/log -v ${container_name}_data:/hadoop -e
LDAP_SERVER_HOST=${LDAP_SERVER_HOST} -e
AMBARI_SERVER_ADDR=${AMBARI_SERVER_ADDR} -e
HDP_REPO_HOST=${HDP_REPO_HOST} -e
KDC_SERVER_HOST=${KDC_SERVER_HOST} -e
encryptKeyString=${encryptKeyString} -e
ldapServerPassword=${ldapServerPassword}
--network ${network} -h ${container_name}.$network
--name=${container_name} $image
```

2. To start the new container, run the command:

Replace `<slavenew>` with the actual value.

```
sudo docker start <slavenew>
```

3. Go to the installer node, run the command:

```
sudo docker start -i installer
```

Type 6 as the input to re-update the calico network profile.

Installing the HDP slave services on the container

After you have started the ambari-agent you can install the HDP slave services on the container using the ambari user interface.

About this task

To install the HDP slave services you must use the ambari service from a browser window.

Procedure

1. From a browser window, open the ambari service:

Replace `<ambari-host>` with the host name or IP address of the ambari node.

```
http:<ambari-host>:8080
```

2. Click **Hosts > Actions > Add New hosts**
3. In the **Install Option** page, apply the container name, for example: `slave03.ife`.
4. Check the option **Perform manual registration on hosts and do not use SSH** and click **Register and confirm**.
5. In the **Confirm Hosts** page, when the Status shows **Success**, click **Next**.
6. In the **Assign Slaves and Clients** page, check the options **DataNode**, **NodeManager**, **RegionServer**, and **Clients** services. Click **Next**.
7. Accept the default configuration for all the remaining pages.
8. Type the **Admin principle** and **Admin password** in the **Admin session expiration error** page.
By default the **Admin principle** is `admin/admin`, the password is set in the `config.properties` file when you did the installation.
9. Wait and verify all services start successfully on the new slave container.

Copying the existing keytab files to the new slave container

Procedure

1. Go to the kdc container on the ambari host.

```
sudo docker exec -it kdc bash
```

2. Run the command to copy the keytab files to new slave container.
Replace `<newslave>` with the actual container name.

```
scp /etc/security/keytabs/* <newslave>:/etc/security/keytabs
```

3. Go to the `<newslave>` container and change the keytab files in `/etc/security/keytabs` to the correct owner.

For example, change `cmopsadmin.keytab` to `cmopsadmin user`, `cm_sample.keytab` to `cm_sample owner`.

```
chown cmopsadmin:cm /etc/security/keytabs/cmopsadmin.keytab  
chown cm_sample:cm /etc/security/keytabs/cm_sample.keytab
```

Installing the client component from Predictive Maintenance and Optimization

There are several client components that are included with IBM® Predictive Maintenance and Optimization. You can install the components as you require.

Important: The client components must be installed on computers running the Microsoft Windows 7 or Microsoft Windows 8 32 or 64-bit operating system.

Important: Install the client components only after you successfully install the server components.

Troubleshooting the installation environment

Cannot access the Cognos BI link

If you cannot access the Cognos BI link.

Symptoms

You cannot open the link to the Cognos BI node. The link `http://<node of the ip where IFECogNode container is deployed>:< exposed port for cognos server in IFECogNode, default 9300>/bi` takes too long to open.

Causes

The Cognos content store database is not created and can occur when the host resource is limited, for example, two DBNodes are in one node.

Diagnosing the problem

Go to the BI Node, run the following commands:

```
sudo docker exec -it IFECogDBNode bash
su - db2inst1
db2 connect to COGNOSCS
exit;
exit;
```

Resolving the problem

If the content store database is not created, enter the IFECogNode container and re-run the scripts that create the COGNOSCS database.

```
sudo docker exec -it IFECogNode bash
cogDBServer_db2instUserPassword=`python /encrypt/lib/encrypt.py $encryptKeyString decrypt
$cogDBServer_db2instUserPassword`
su - db2inst1 -c "/images/CognosDBCcreation.sh $cogDBServer 50000 db2inst1
$cogDBServer_db2instUserPassword"
```

Error message, Error response from daemon: service endpoint with name 'container name' already exists

Symptoms

The Docker containers fail to start with the error “Error response from daemon: service endpoint with name <container_name> already exists.”

Causes

The container is not started on the host, but on other nodes there is a cache status for it.

Diagnosing the problem

1. On the host for the container, run the command to ensure that the container is not started:

```
sudo docker ps -a | grep <container_name>
```

The output shows that the container status is Exited.

2. On the installer host, run the command:

```
sudo docker network inspect ife | grep <container_name>
```

The expected output is nothing, but if you get an output similar to: "Name": " <container_name>" Then the output means that there is the issue on that host. In most case, the issue occurs on installer node, but if you can't find it on installer node, try the other nodes.

3. Repeat step 2 for all other hosts except for the host for the container.

Resolving the problem

1. On the host that you find the issue, run the command:

```
sudo docker network disconnect -f ife <container_name>
```

The cron job to restart liberty server framework_server did not work due to file permissions

If the liberty server framework_server has been started with root user, the cron job for the wlp user no longer works.

Symptoms

The cron job to restart liberty server framework_server did not work due to file permission.

Causes

As liberty server framework_server has been started with root user, some of the files in the /opt/IBM/WebSphere/Liberty/usr/servers/framework_server folder belong to root user.

The wlp user neither has the permission to write to these files, nor the permission to start the liberty server.

Diagnosing the problem

1. Login to the AppNode host, and then enter the IFEAppNode container with the command:

```
sudo docker exec -it IFEAppNode bash
```

2. Run the command to see the files owner:

```
ls -l /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/
```

Resolving the problem

Run the commands in the IFEAppNode container with root user:

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server  
chown -R wlp:wlp /opt/IBM/WebSphere/Liberty/usr/servers/framework_server
```

Cannot access the Application link through IBM HTTP Server

In the Ubuntu environment, sometimes you can only access the application through liberty port number. This checks to see if the IHSServer configuration needs to be added.

Symptoms

You cannot access the application through IBM HTTP Server.

Causes

This may caused by the configuration for the IBM HTTP Server is not added in the httpd.conf file.

Diagnosing the problem

Login to the AppNode where IFEIHSNode is deployed. Type the command:

```
sudo docker exec -it IFEIHSNode bash  
cat /opt/IBM/HTTPServer/conf/httpd.conf
```

The httpd.conf file should have at least the setting:

```
#Add proxy settings to redirect IBMIOTE&U services  
ProxyPass /ibm https://IFEAppNode.ife:9443/ibm nocanon
```

```
ProxyPreserveHost ON  
SSLProxyEngine ON
```

Resolving the problem

If the IHSServer configuration is missing, type the command to configure IHSServer:

```
chmod +x /config/httpConfProxy.sh;  
/config/httpConfProxy.sh
```

Chapter 3. Administering the product

The applications in IBM IoT for Energy and Utilities can be administered with users with the correct administrative rights.

Stop and start solution software services

IBM IoT for Energy and Utilities is an integrated solution that includes many applications. If you must stop the services, you must do so in the correct order. The product services must also be started in the correct order.

Start solution services

You must start the IoT for Energy and Utilities node services in a specific order.

Start the node services in the following order:

1. [“Starting services on the data node” on page 55](#)
2. [“Starting services on the analytics node” on page 55](#)
3. [“Starting services on the integration bus node” on page 56](#)
4. [“Starting services on the BI node” on page 56](#)

Starting services on the data node

You must start the IBM DB2 instance on the IBM IoT for Energy and Utilities data node.

Procedure

1. Log in to the IoT for Energy and Utilities data node as **root**.
2. Log in to the data node computer as the IBM DB2 administrator user.
3. In a terminal window, type the following command to change the DB2 instance owner:

```
su - db2inst1
```
4. Click **Start > IBM DB2 > DB2COPY1 (Default) > DB2 Command Window - Administrator**.
5. Enter the following command to start the DB2 administration server:

```
db2start
```
6. Start Jena on the data node computer:

```
/opt/Jena/startJena_Linux.sh
```

Starting services on the analytics node

You must start the IBM SPSS services on the IBM IoT for Energy and Utilities SPSS node.

Procedure

1. Log in to the analytics node as **root**.
2. Go to the WebSphere® Application Server CNDSprofile/bin directory.
For example, go to /opt/IBM/WebSphere/AppServer/profiles/CNDSprofile/bin.
3. Enter the following command:

```
./startServer.sh server1
```



```
startserver.bat server1
```
4. Go to the IBM SPSS Modeler Server directory.
For example, go to /opt/IBM/SPSS/ModelerServer/18.0.
5. Enter the following command:

```
./modelersrv.sh start
```

6. Click **Start > Control Panel > Administrative Tools > Services**.
7. Select **IBM SPSS Modeler Server 16.0**, and click **Start Service**.

Starting services on the integration bus node

You must start the IBM Integration Bus services on the IBM IoT for Energy and Utilities integration bus node.

Procedure

1. Log in to the integration bus node as **root**.
2. Change to the **mqmuser**.
For example, enter the following command:

```
su - mqmuser
```

3. Go to the **mq bin** directory.
For example, go to **/opt/mqm/bin**.
4. To load the profile for the mqm user, enter the following command and press enter:

```
~/ .bash_profile
```

5. To start the queue manager, enter the following command:

```
strmqm queue_manager_name
```

For example, to start the default broker that is named **pmqmanager** enter the following command:

```
strmqm pmqmanager
```

6. Go to the IBM Integration Bus bin directory. For example, go to **/opt/ibm/iib-10.0.0.7/server/bin**.
7. Enter the following command:

```
./mqsistart pmqbroker
```
8. To verify that the services started, enter one of the following commands:

```
./mqsilist pmqbroker
```

 or

```
./mqsilist pmqbroker -e pmqgroup1
```

. The

```
./mqsilist pmqbroker -e pmqgroup1
```

 command lists all entries that are running under **pmqgroup1**.
9. Enter the following commands:

```
runmqsc pmqmanager  
START LISTENER  
END
```

10. Start the Liberty servers. Go to the Liberty installation location, for example, **/opt/IBM/WebSphere/Liberty/**, and run the following commands:

```
bin/server start controller_server  
bin/server start framework_server
```

Starting services on the BI node

You must start the IBM Cognos Business Intelligence services and IBM HTTP Server on the IBM IoT for Energy and Utilities BI node.

You start the IBM Cognos BI services by starting the WebSphere Application Server profile where IBM Cognos BI is running.

About this task

Use the following steps to start IBM Cognos in silent mode.

Procedure

1. Go to the `cognos_install_location/bin64` directory. For example, go to the `/opt/ibm/cognos/analytics/bin64` directory.
2. Enter the following command:

```
export JAVA_HOME=/opt/ibm/cognos/analytics/jre/  
./cogconfig.sh -s
```

Stop solution services

You must stop the IBM IoT for Energy and Utilities node services in a specific order.

Stop the node services in the following order:

1. [“Stopping services on the BI node” on page 57](#)
2. [“Starting services on the integration bus node” on page 56](#)
3. [“Stopping services on the analytics node ” on page 58](#)
4. [“Stopping services on the data node” on page 58](#)

Stopping services on the BI node

You must stop the IBM Cognos Business Intelligence services and IBM HTTP Server on the IBM IoT for Energy and Utilities BI node.

You stop the IBM Cognos BI services by stopping the WebSphere Application Server profile where IBM Cognos BI is running.

About this task

Use the following steps to stop IBM Cognos in silent mode.

Procedure

1. Go to the `cognos_install_location/bin64` directory. For example, go to the `/opt/ibm/cognos/analytics/bin64` directory.
2. Enter the following command:

```
export JAVA_HOME=/opt/ibm/cognos/analytics/jre/  
./cogconfig.sh -stop
```

Stopping services on the integration bus node

You must stop the IBM Integration Bus services on the integration bus node.

Procedure

1. Log in to the integration bus node computer as **root**.
2. Change to the **mqmuser**.
For example, enter the following command:

```
su - mqmuser
```
3. Go to the IBM Integration Bus `bin` directory folder.
For example, go to `/opt/ibm/iib-10.0.0.7/server/bin`.
For example, go to `/opt/ibm/iib-10.0.0.7/server/bin`.
For example, go to `C:\Program Files\IBM\MQSI\9.0.0.1\bin`.
4. Enter the following command:

```
./mqsistop pmobroker -i
```
5. Enter the following command to verify that the services are stopped:

```
./mqsilist pmobroker
```

6. Go to the Liberty installation location, for example, `/opt/IBM/WebSphere/Liberty/`, and stop the Liberty servers using the following commands:
`bin/server stop framework_server`
`bin/server stop controller_server`

Stopping services on the analytics node

You must stop the IBM SPSS services on the IBM IoT for Energy and Utilities analytics node.

Procedure

1. Log in to the analytics node as **root**.
2. Go to the WebSphere Application Server CNDSprofile/bin directory.
For example, go to `/opt/IBM/WebSphere/AppServer/profiles/CNDSprofile/bin`.
3. Enter the following command:
`./stopServer.sh server1`
`stopserver.bat server1`
4. Go to the IBM SPSS Modeler Server directory.
For example, go to `/opt/IBM/SPSS/ModelerServer/18.0`.
5. Enter the following command:
`./modelersrv.sh stop`
6. To verify whether any services are still running enter the following command:
`ps -ef | grep statisticsd`
7. To stop any services that are still running enter the following command:
`kill -9 'cat statisticsd.pid'`
8. Click **Start > Control Panel > Administrative Tools > Services**.
9. Select **IBM SPSS Modeler Server 16.0**, and click **Stop Service**.

Stopping services on the data node

You must stop the IBM DB2 instance on the IBM IoT for Energy and Utilities data node.

Procedure

1. Log in to the data node as **root**.
2. Log in to the data node computer as the IBM DB2 administrator user.
3. In a terminal window, type the following command to change the DB2 instance owner:
`su - db2inst1`
4. Click **Start > IBM DB2 > DB2COPY1 (Default) > DB2 Command Window - Administrator**.
5. Enter the following command to stop the DB2 administration server:
`db2stop`
6. Stop Jena on the data node:
`/opt/Jena/stopJena_Linux.sh`

Starting and stopping services

All services are inside docker containers so you need these commands to stop and start the Docker services.

Procedure

1. Run the command:

```
sudo docker exec -t <container name> bash -c '<put actual start/stop commands here>
```

2. Run this command to check the progress:

```
sudo docker exec -t <container name> bash -c 'ps -ef'
```

Stop the solution services

You must stop the IBM IoT for Energy and Utilities node services in a specific order.

1. Business Intelligence (BI) node
2. App node
3. SPSS node
4. Database node
5. HDP services
6. HDP nodes (HDP master, slave, notebook, and client nodes)
7. Ambari server node

Stopping services on the BI node

You must stop the IBM® Cognos® Business Intelligence services and IBM DB2 for Cognos® Business Intelligence on the BI node.

About this task

You stop the IBM Cognos BI services by stopping the Cognos server and the IBM DB2 database for IBM Cognos Business Intelligence.

Procedure

1. Log into the BI Node as root or sudo user.
2. Run this command to stop the Cognos Server.

```
sudo docker exec -t IFECogNode bash -c 'su - cognos -c "/opt/ibm/cognos/analytics/bin64/cogconfig.sh -stop"'
```

3. Run this command to stop IBM DB2 Server for Cognos BI.

```
sudo docker exec -t IFECogDBNode bash -c 'su - db2inst1 -c "db2 force application all;db2stop"'
```

Stopping services on the App node

You must stop the Jena, IBM HTTP, Liberty server, and the LDAP server services on the App node.

Procedure

1. Log in to the App node as root or sudo user.
2. Run the command to stop IBM HTTP Server.

```
sudo docker exec -t IFEIHSNode bash -c '/opt/IBM/HTTPServer/bin/apachectl stop'
```

3. Run the command to stop the liberty server used by the IBM IoT for Energy and Utilities framework_server.

```
sudo docker exec -t IFEAppNode bash -c 'su - wlp -c "/opt/IBM/WebSphere/Liberty/bin/server stop framework_server"'
```

4. Run bellow command to stop Ldap Server.

```
sudo docker exec -it IFELdapNode bash -c 'systemctl stop slapd'
```

Stopping services on the SPSS node

You must stop the SPSS Modeler server for IBM IoT for Energy and Utilities on the SPSS node.

Procedure

1. Log in SPSS Node as root or sudo user.
2. Run the command to stop SPSS Modeler Server.

```
sudo docker exec -t IFESpssNode bash -c 'su - spss -c "/usr/IBM/SPSS/ModelerServer/18.1/modelersrv.sh stop"'
```

Stopping services on the DB node computer

You must stop the IBM® DB2® instance on the IBM IoT for Energy and Utilities data node.

Procedure

1. Log in to the data node computer as root or sudo user.
2. Run bellow command to stop IBM DB2.

```
sudo docker exec -t IFEDBNode bash -c 'su - db2inst1 -c "db2 force application all;db2stop"'
```

3. Run bellow command to stop Jena.

```
sudo docker exec -t IFEJenaNode bash -c 'su - root -c "/opt/Jena/stopJena_Linux.sh"'
```

Stopping the HDP services

You must use the Ambari user interface to stop the Hortonworks Data Platform (HDP) service for IBM IoT for Energy and Utilities.

Procedure

1. Login to `http://<ambari host>:8080`.
2. Go to the services link.
3. Select the services to stop, or use **Actions > Stop All** to stop all services.

Stopping the Ambari agent on the HDP nodes

To stop the Ambari agent on all the HDP master, slave, notebook, and client nodes for IBM IoT for Energy and Utilities.

Procedure

1. Log in each HDP Node as root or sudo user.

HDP nodes includes the HDP master nodes, HDP slave nodes, HDP notebook nodes, and HDP client nodes.

Note: By default there is no client node, however you may install it by customization.

2. Run the command replacing the `<container_name>` with the actual value of the node.

```
sudo docker exec -it <container_name> bash -c "systemctl stop ambari-agent"
```

For example: For the master01 node, login to the master01 node as root or sudo user, and run the command:

```
sudo docker exec -it master01 bash -c "systemctl stop ambari-agent"
```

Repeat this step for all other HDP nodes.

Stopping services on the Ambari node computer

You must stop the Ambari server, the KDC server and the HTTP server on the IBM IoT for the IBM IoT for Energy and Utilities Ambari node.

Procedure

1. Log in to the Ambari node computer as root or sudo user.
2. Run the command to stop the Ambari server.

```
sudo docker exec -it ambari bash -c "systemctl stop ambari-server"
```

3. Run the command to stop the KDC server.

```
sudo docker exec -it kdc bash -c " systemctl stop krb5kdc;systemctl stop kadmind"
```

4. The repo container is used to install HDP services. Run the command to stop the HTTP server.

```
sudo docker exec -it repo bash -c "systemctl stop httpd"
```

Start solution services

You must start the IBM IoT for Energy and Utilities node services in a specific order.

1. Database node
2. Business Intelligence (BI) node
3. App node
4. SPSS node
5. Ambari server node
6. HDP nodes (HDP master, slave, notebook and client nodes)
7. HDP services

Starting services on the DB node computer

You must start the IBM® DB2® instance on the IBM IoT for Energy and Utilities data node.

Procedure

1. Log in to the data node computer as root or sudo user.
2. Run bellow command to start Jena.

```
sudo docker exec -t IFEJenaNode bash -c 'su - root -c "/opt/Jena/startJena_Linux.sh"'
```

3. Run bellow command to start IBM DB2.

```
sudo docker exec -t IFEDBNode bash -c 'su - db2inst1 -c "db2start"'
```

Starting services on the BI node

You must start the IBM® Cognos® Business Intelligence services and IBM DB2 for Cognos® Business Intelligence on the BI node.

About this task

You start the IBM Cognos BI services by starting the Cognos server and the IBM DB2 database for IBM Cognos Business Intelligence.

Procedure

1. Log into the BI Node as root or sudo user.
2. Run this command to start IBM DB2 Server for Cognos BI.

```
sudo docker exec -t IFECogDBNode bash -c 'su - db2inst1 -c "db2start"'
```

3. Run this command to start the Cognos Server.

```
sudo docker exec -t IFECogNode bash -c 'su - cognos -c "/opt/ibm/cognos/analytcs/bin64/cogconfig.sh -s"'
```

Starting services on the App node

You must start the Jena, IBM HTTP, Liberty server, and the LDAP server services on the App node.

Procedure

1. Log in to the App node as root or sudo user.
2. Run the command to start IBM HTTP Server.

```
sudo docker exec -t IFEIHSNode bash -c '/opt/IBM/HTTPServer/bin/apachectl start'
```

3. Run the command to start the liberty server used by the IBM IoT for Energy and Utilities framework_server.

```
sudo docker exec -t IFEAppNode bash -c 'su - wlp -c "/opt/IBM/WebSphere/Liberty/bin/server start framework_server"'
```

4. Run the command to start the LDAP Server.

```
sudo docker exec -it IFELdapNode bash -c 'systemctl start slapd'
```

Starting services on the SPSS node

You must start the SPSS Modeler server for IBM IoT for Energy and Utilities on the SPSS node.

Procedure

1. Log in SPSS Node as root or sudo user.
2. Run the command to start SPSS Modeler Server.

```
sudo docker exec -t IFESpssNode bash -c 'su - spss -c "/usr/IBM/SPSS/ModelerServer/18.1/modelersrv.sh start"'
```

Starting services on the Ambari node computer

The Ambari server, the KDC server and the HTTP server need to start on the IBM IoT for Energy and Utilities Ambari node.

About this task

By default the Ambari server is on the ambari container, the KDC server is on the kdc container, and the HTTP server on repo container and are be started automatically when starting the container.

Here are the commands to start the service manually.

Procedure

1. Log in to the Ambari node computer as root or sudo user.
2. Run the command to start the Ambari server.

```
sudo docker exec -it ambari bash -c "systemctl start ambari-server"
```

3. Run the command to start the KDC server.

```
sudo docker exec -it kdc bash -c " systemctl start krb5kdc;systemctl start kadmind"
```

4. The repo container is used to install HDP services. If you have such requirement, run the command to start the HTTP server.

```
sudo docker exec -it repo bash -c "systemctl start httpd"
```

Starting the Ambari agent on the HDP nodes

To start the Ambari agent on all HDP nodes: the HDP master, slave, notebook, and client nodes for IBM IoT for Energy and Utilities.

About this task

By default the Ambari agent starts automatically when starting the container. Here are the commands to start it manually.

Procedure

1. Log in to each HDP Node as root or sudo user.

The HDP nodes includes the HDP master nodes, HDP slave nodes, HDP notebook nodes, and the HDP client nodes.

Note: By default there is no client node, however the you may install it by customization.

2. Run the command replacing the `<container_name>` with the actual value.

```
sudo docker exec -it <container_name> bash -c "systemctl start ambari-agent"
```

For example: Login to master01Node as root, and run the command:

```
sudo docker exec -it master01 bash -c "systemctl start ambari-agent"
```

Starting the HDP services

You must use the Ambari user interface to start the Hortonworks Data Platform (HDP) service for IBM IoT for Energy and Utilities.

Procedure

1. Login to `http://<ambari host>:8080`.
2. Go to the services link.
3. Select the services to start, or use **Actions** > **Start All** to start all services.

Applying calico network settings after a system reboot or a container restart

When you reboot the host Linux server or do a container restart, you must apply the calico network settings.

About this task

A server reboot or a container restart can change the IP address of the container.

Note: After a system reboot, you must manually log in and start each container that is deployed on this host.

Procedure

1. On each host, use the command to start all containers on the host.

```
sudo docker start $(docker ps -a -q)
```

Wait 10 minutes for the services to start.

If the container fails to start, see the troubleshooting section, [“Error message, Error response from daemon: service endpoint with name 'container name' already exists”](#) on page 52.

2. On the AppNode, run the command to re-start the IFEIHSNode, so the /etc/hosts file in IFEIHSNode container is updated based on the new virtual IP address.

```
sudo docker stop IFEIHSNode; docker start IFEIHSNode
```

You can check if the IHS server has started with the command:

```
sudo docker exec -it IFEIHSNode ps -ef
```

The output will return the following scripts that has the httpd process:

```
[root@testlibertyserver ~]# docker exec -it IFEIHSNode ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 09:02 ? 00:00:00 /bin/bash /entrypoint.sh
root 15 1 0 09:02 ? 00:00:00 /opt/IBM/HTTPServer/bin/httpd -d
nobody 17 15 0 09:02 ? 00:00:00 /opt/IBM/HTTPServer/bin/httpd -d
nobody 20 15 0 09:02 ? 00:00:00 /opt/IBM/HTTPServer/bin/httpd -d
nobody 21 15 0 09:02 ? 00:00:00 /opt/IBM/HTTPServer/bin/httpd -d
nobody 136 15 0 09:04 ? 00:00:00 /opt/IBM/HTTPServer/bin/httpd -d
root 1421 1 0 09:25 ? 00:00:00 sleep 1
root 1422 0 0 09:25 ? 00:00:00 ps -ef
```

Note: If the IHSNode has not started, you must repeat this step.

Note: If the IHSNode has still not started, use the command:

```
sudo docker exec -it IFEIHSNode /opt/IBM/HTTPServer/bin/apachectl start
```

to manually start the service.

3. On installer node, stop the installer container with this command:

```
sudo docker stop installer
```

4. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

5. Type 1 to encrypt the passwords for all nodes and to generate the runtime configuration files.
6. Run the command to restore the installation menu:

```
sudo docker start -i installer
```

7. Type 6 to update the network calico profile.

Managing system access

Securing your IBM IoT for Energy and Utilities solution is an important consideration. To ensure that the system is secure, you must manage who can access the system and assign the correct level of access within the solution.

Securing access to the solution

IoT for Energy and Utilities uses a IBM WebSphere Application Server Liberty Server basic user registry to authenticate and authorize users. For more information about Liberty profile user registries, see the related link.

Your administrator assigns access to features, data, and services in your solution based on the user role groups.

The following topics explain the security and how to manage user access to IoT for Energy and Utilities.

Adding users and user groups to access the user interface

To access specific features or services in the solution, a user must be a member of a user role group that provides the required level of access to that feature or service. IoT for Energy and Utilities on Cloud uses

an open LDAP registry to define users and user role groups. You can add users and user role groups to the solution by adding users and groups to the LDAP server on the App node.

Before you begin

Decide on the groups, users, and user passwords that you want to add to the IoT for Energy and Utilities on Cloud basic user registry.

Procedure

1. Log in to the App Node, and enter the IFELdapNode Docker container.

```
sudo docker exec -it IFELdapNode bash
```

2. Add user and group in LDAP Server.

- a) Create a LDIF file to describe the group and user information.

```
touch /tmp/temp.ldif
```

- b) Add the new group and users of IoT for Energy and Utilities on Cloud to the temp.ldif file and replace the userpassword attribute value with the password you want to use for the new user.

For example, if the user group `reliability_group`, the users are `rel_engineer_01`, `rel_engineer_02`, and are added to the content to the temp.ldif file.

```
dn: cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com
objectclass: top
objectclass: groupOfNames
member: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
member: cn=rel_engineer_02,cn=ife,ou=application,dc=ibmiot,dc=com
cn: reliability_group

dn: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
objectclass: person
objectclass: top
cn: rel_engineer_01
sn: rel_engineer_01
userpassword: password_for_rel_engineer_01

dn: cn=rel_engineer_02,cn=ife,ou=application,dc=ibmiot,dc=com
objectclass: person
objectclass: top
cn: rel_engineer_02
sn: rel_engineer_02
userpassword: password_for_rel_engineer_02
```

- c) Use the command to add to LDAP Server and replace ``${LDAP_PASSWORD}` with actual `ldapServerPassword` that was provided during install in the `config.properties` file.

```
ldapadd -x -D "cn=Manager,dc=ibmiot,dc=com" -w `${LDAP_PASSWORD}` -f /tmp/temp.ldif
```

3. Check the users and user groups in the LDAP server.

- a) Check new added user with the command.

```
ldapsearch -x | grep rel_engineer_01
```

The output is:

```
[root@IFELdapNode /]# ldapsearch -x | grep rel_engineer_01
member: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
# rel_engineer_01, ife, application, ibmiot.com
dn: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
cn: rel_engineer_01
sn: rel_engineer_01
```

- b) Check new added group with the command.

```
ldapsearch -x | grep reliability_group
```

The output is:

```
[root@IFELdapNode /]# ldapsearch -x | grep reliability_group
# reliability_group, ife, application, ibmiot.com
dn: cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com
cn: reliability_group
```

Results

The new groups and users are added to the basic user registry, and the users can now be authenticated when they log on to IoT for Energy and Utilities on Cloud.

What to do next

- To generate usage information for the IBM License Metric Tool, you must map each user role group to the relevant license type in the `s1mtag_groups.properties` file on the application server. For more information, see [“Mapping user groups to license types” on page 74](#).
- If you are an administrator, you can now assign access to pages and REST services in the solution to each new user role. For more information about configuring access control for pages and services, see the related links.

Modifying users, user groups, and passwords for the user interface

You can change passwords and group membership for users in IoT for Energy and Utilities on Cloud. Membership of a user role group gives users access to the parts of the solution that are appropriate to that user role. You can change the access level of a user by updating the basic user registry to remove the user from one group and add the user to another group. You can also update the basic user registry to remove users and groups that no longer require access to the solution.

Before you begin

- While you update the basic user registry, ensure that the affected users are not logged on to IBM IoT for Energy and Utilities.
- Before you remove groups from the basic registry, ensure that the groups are not assigned access to pages and services in the solution. For more information about configuring access to pages and services in IoT for Energy and Utilities, see the related links.

Procedure

1. Log in App Node and enter IFELdapNode docker container.

```
sudo docker exec -it IFELdapNode bash
```

2. Create a LDIF file to the modify the group or user.

```
touch /tmp/temp.ldif
```

3. To add the user `rel_engineer_02` to the group `reliability_group`.

Note: If the user `rel_engineer_02` does not exist, run the steps a and b to create the user.

- a) Create the file `/tmp/user2.ldif` with the content, replace `password_for_rel_engineer_02` with the actual password.

```
dn: cn=rel_engineer_02,cn=ife,ou=application,dc=ibmiot,dc=com
objectClass: person
objectClass: top
cn: rel_engineer_02
sn: rel_engineer_02
userpassword: password_for_rel_engineer_02
```

- b) Run the command to add the user, replace `#{LDAP_PASSWORD}` with the `ldapServerPassword` that was provided during install in the `config.properties` file.

```
ldapadd -x -D "cn=Manager,dc=ibmiot,dc=com" -w  
${LDAP_PASSWORD} -f /tmp/user2.ldif
```

- c) Edit the /tmp/temp.ldif file to add LDAP data and commands.
- d) Add the script to the file:

```
dn:cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com  
changetype: modify  
add: member  
member: cn=rel_engineer_02,cn=ife,ou=application,dc=ibmiot,dc=com
```

- e) Using the command to add the contents of the temp.ldif file into LDAP Server, replace \${LDAP_PASSWORD} with actual *ldapServerPassword* which was provided during install in the config.properties file.

```
ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD} -f /tmp/temp.ldif
```

4. To change the password for the user *rel_engineer_01*.

- a) Edit the /tmp/temp.ldif file to add LDAP data and commands.
- b) Add the script to the file:

Where the *passwdOrdChange* is the new password.

```
dn:cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com  
changetype: modify  
replace: userpassword  
userpassword: passwdOrdChange
```

- c) Using the command to add the contents of the temp.ldif file into LDAP Server, replace \${LDAP_PASSWORD} with actual *ldapServerPassword* which was provided during install in the config.properties file.

```
ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD} -f /tmp/temp.ldif
```

5. To remove the user *rel_engineer_01* from the group *reliability_group*.

- a) Edit the /tmp/temp.ldif file to add LDAP data and commands.
- b) Add the script to the file:

```
dn:cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com  
changetype: modify  
delete: member  
member: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
```

- c) Using the command to add the contents of the temp.ldif file into LDAP Server, replace \${LDAP_PASSWORD} with actual *ldapServerPassword* which was provided during install in the config.properties file.

```
ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD} -f /tmp/temp.ldif
```

6. To delete the user *rel_engineer_01* from the system you directly use the *ldapdelete* command and replace *LDAP_PASSWORD* with the *ldapServerPassword* that was provided during install in the config.properties file:

```
ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com" -w  
${LDAP_PASSWORD} "cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com"
```

7. To delete the group *reliability_group* from the system you directly use the *ldapdelete* command and replace *LDAP_PASSWORD* with the *ldapServerPassword* that was provided during install in the config.properties file.

```
ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com" -w  
${LDAP_PASSWORD} "cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com"
```

Results

The users and groups are modified or removed in the IoT for Energy and Utilities basic user registry.

Adding a Connectivity Model tenant user

The steps to add a new tenant user to the Connectivity Model application.

Procedure

1. Create a user in LDAP.

- a) Log in the App Node, and open the IFELdapNode Docker container.

```
sudo docker exec -it IFELdapNode bash
```

- b) Run the command with user `root`:

```
/config/cm_add_tenant_user.sh <username> <userpassword> <uidNumber> <LDAP password>
```

Replace the parameters `<username>` `<userpassword>` `<uidNumber>` `<LDAP password>` with the actual value.

Note:

The `uidNumber` must be a unique number, not used by other members of LDAP.

- c) Check the new user, replace the parameter `<username>` with the actual value.

```
ldapsearch -x | grep <username>
```

2. Create `keytab`.

Note: The `keytab` is distributed to all HDP nodes.

- a) Login in the Ambari Node, and open the `kdc` Docker container.

```
sudo docker exec -it kdc bash
```

- b) Run the command with user `root`:

```
/opt/kdc/createTenant.sh <username> <keytab_name>
```

Note: The `<username>` must be the same as the `<username>` added in step 2, and `<keytab_name>` suggested to be `username.keytab`

For example: If `username=cmuser`, then `keytab_name` suggested to be `cmuser.keytab`.

3. Create tenant.

- a) Login to the Notebook node, and go to the notebook container.

```
sudo docker exec -it notebook bash
```

- b) Run the command with user `root`:

```
su - hdfs -c "kinit -k -t /etc/security/keytabs/hdfs.headless.keytab hdfs"
```

- c) Run the commands with user `root`.

```
/home/cmopsadmin/cm/bin/APP_createUtility.sh  
<username> <path for keytab> <user principle>
```

Currently, the `<user principle>` is the same as `<username>`, the `<path for keytab>` is `/etc/security/keytabs`. The `<username>` must be the same as for previous steps.

For example, if `<username>` is `cmuser`, then the suggested `<keytab_name>` is `cmuser.keytab`, and the command is:

```
/home/cmopsadmin/cm/bin/APP_createUtility.sh cmuser
/etc/security/keytabs/cmuser.keytab cmuser
```

4. Configure a Jupyter notebook for the tenant.

- a) Login to the Notebook node, and go to the notebook container.

```
sudo docker exec -it notebook bash
```

- b) Run the command with user `root`:

```
/opt/jupyter/configureNotebook.sh <username>
<notebook-password> <notebook-port>
```

Note: The `username` must be the same as in the previous steps. The `notebook-password` is the password to login to the notebook: you need to create and identify the password here. The `notebook-port` is the internal port for the container that is defined in the `container.ini` file for the notebook container.

For example:

```
/opt/jupyter/configureNotebook.sh cmuser passwd 8889
```

- c) To verify the Jupyter notebook, open the link with your browser:

```
https://<notebook_node_hostname_or_IP>:<port>
```

Where the login password is `<notebook-password>` and `<port>` is the external port that maps to the internal port or `notebook-port`. The external port, internal port, and the mapping relationship are defined the `containers.ini` file.

Note: When you run the above commands to create a notebook for the tenant, the tenant notebook starts automatically after a restart of the Docker container.

If you do not want the automatic start you can comment out the line `/opt/jupyter/startNotebook.sh <username>` in the `/etc/rc.local` file.

5. Change the default `map_center` for the tenant user.

By default, the tenant `map_center` is `(-83.44,42.60)`, you can do the following steps to change the default setting.

- a) Login to the Notebook node, and go to the notebook container.

```
sudo docker exec -it notebook bash
```

- b) Change the current user to `cmopsadmin`.

```
su - cmopsadmin
```

- c) Run the following commands with user `cmopsadmin`.

```
cd /usr/hdp/current/phoenix-client/bin/
```

```
./sqlline.py master01:2181:/hbase-secure:
cmopsadmin:/etc/security/keytabs/cmopsadmin.keytab
```

Note: If you get an error Caused by:

```
org.apache.hadoop.hbase.ipc.RemoteWithExtrasException
(org.apache.hadoop.hbase.security.AccessDeniedException):
org.apache.hadoop.hbase.security.AccessDeniedException:
Insufficient permissions (user=cmopsadmin@IBMIOT.COM,
scope=SYSTEM:CATALOG,
```

family=0:_0, params=[table=SYSTEM:CATALOG,family=0:_0],action=WRITE), you can ignore it. The error is a known defect from HDP.

- d) Run the following in phoenix sqlline.py to update the default map center and related configuration:

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values ('<utilityname>','<map_center>','<map_min_zoom>',
'<map_max_zoom>','<map_default_zoom>')
```

Note:

The *<utilityname>* is the tenant user, for example cmuser or cm_sample.

The *<map_center>* is the center of map, for example: [-83.44, 42.61].

The *<map_min_zoom>* is the minimum zoom setting for the map, for example 1.

The *<map_max_zoom>* is the maximum zoom setting for the map, for example 22.

The *<map_default_zoom>* is the default zoom setting for the map, for example 12.

For example, the sql command is like this:

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values ('cmuser','[-83.44, 42.61]','1','22','12')
```

Removing a Connectivity Model tenant user

The steps to remove a tenant user from the Connectivity Model application.

About this task

Important: When you remove a Connectivity Model tenant user from the application, the operation cannot be undone. Make sure you are certain when you remove a tenant user.

The following steps use **cm_sample/CM_SAMPLE** as the example, you need to replace it with the actual tenant user name. The tenant name is case sensitive, you need to replace **cm_sample** with the same letter case as when you create the tenant.

Procedure

1. Clear hdfs files.

- a) Login to notebook node and notebook container:

```
sudo docker exec -it notebook bash
```

- b) Run the command with root user:

```
su - hdfs
```

- c) Run the commands with hdfs user. You must replace **cm_sample** with the actual tenant name.

```
kinit -k -t /etc/security/keytabs/hdfs.headless.keytab hdfs
hdfs dfs -rm -r -f /user/cm_sample
```

- d) Verify the folder is deleted with the command:

```
hdfs dfs -ls /user
```

- e) Exit hdfs user with the command:

```
exit
```

2. Clear the hbase data and tables.

- a) Login to notebook node and notebook container:

```
sudo docker exec -it notebook bash
```

- b) Run the command with root user:

```
su - hbase
```

- c) Create a file with hbase user with the content:

Note: Assume the file path is /home/hbase/drop_tables.sql

```
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_ANALYSIS_HISTORY;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_ANALYSIS_HISTORY;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE_BY_LOAD;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE_BY_AMI_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE_BY_AMI_AND_SCADA_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE_BY_LOAD;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE_BY_AMI_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE_BY_AMI_AND_SCADA_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.LATERAL_TRANSFORMER_PHASE_ERROR;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.KPI_HISTORY_BY_UTILITY;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.KPI_HISTORY_BY_SUBSTATION_REGION;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.KPI_HISTORY_BY_FEEDER;
DROP SCHEMA IF EXISTS CMODEL_CM_SAMPLE;
DELETE FROM CMODEL.UTILITY where UTILITY='cm_sample';
DELETE FROM CMODEL.USER where UTILITY='cm_sample';
```

Note: The last two sql commands are case sensitive. The schema name is always translated to upper case in hbase, so the schema name is as CMODEL_CM_SAMPLE. In the tables however, the tenant name remains the same as your created tenant, so in the last two sql commands, they are **cm_sample**.

- d) Drop tables with file created in last step.

```
cd /usr/hdp/current/phoenix-client/bin
./psql.py master01:2181:/hbase-secure:hbase:/etc/security/keytabs/hbase.headless.keytab /
home/hbase/drop_tables.sql
```

- e) Verify the hbase tables are cleaned:

```
cd /usr/hdp/current/phoenix-client/bin
./sqlline.py master01:2181:/hbase-secure:hbase:/etc/security/keytabs/hbase.headless.keytab
!tables
Select * from CMODEL.UTILITY;
Select * from CMODEL.USER;
!exit
```

3. Disable the Jupyter Notebook and clean the local files.

- a) Login to notebook node and notebook container.

```
sudo docker exec -it notebook bash
```

- b) Run the command to get the Jupyter Notebook process id:

```
ps -ef | grep cm_sample
```

Here is the sample output, **235** is the process id.

```
cm_samp+ 235      1  0 02:52 ?          00:00:00 /usr/bin/python2 /bin/jupyter-notebook --
ip=192.168.202.72 --port=8888 --notebook-dir=/home/cm_sample --certfile=mycert.pem --
keyfile mykey.key --no-browser
```

- c) Stop the Jupyter process with the command:

```
kill -9 <process id>
```

for example:

```
Kill -9 235
```

- d) Open `/etc/rc.local`, and delete the line with the tenant name.

For example: Delete this line in the file:

```
/opt/jupyter/startNotebook.sh cm_sample
```

- e) Run the command to delete local files:

```
rm -rf /home/cm_sample
```

4. Clean the keytab files.

Important: You must delete the `/etc/security/keytabs/cm_sample.keytab` file in these containers:

- kdc container in Ambari node.
- Notebook container.
- all hdp slave containers.
- all hdp master containers.

...

5. Delete the tenant user in LDAP.

- a) Log in the App node and enter the IFELdapNode docker container.

```
sudo docker exec -it IFELdapNode bash
```

- b) Create a file `/tmp/delete.ldif`. Here is the file content, replace `cm_sample` with tenant name.

```
dn: cn=cm,ou=application,dc=ibmiot,dc=com
changetype: modify
delete: memberUid
memberUid: cm_sample
```

- c) Run the command, replace `${LDAP_PASSWORD}` with actual `ldapServerPassword` which was provided during install in the `config.properties` file, replace `cm_sample` with the tenant name.

```
ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com"
-w ${LDAP_PASSWORD} -f /tmp/delete.ldif
ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com"
-w ${LDAP_PASSWORD} "cn=cm_sample,ou=tenant,dc=ibmiot,dc=com"
```

6. Verify the removal of the tenant user:

```
ldapsearch -x | grep cm_sample
```

Giving the UI user the permission to access Connectivity Model tenant user data

This procedure gives the rights to the UI user to access Connectivity Model data from the user interface.

Before you begin

Before you do the following steps, ensure there is a valid UI user. If a UI user does not exist, you must do either [Adding users and user groups to access the user interface](#) or [“Modifying users, user groups, and passwords for the user interface”](#) on page 66.

About this task

The UI user for Connectivity Model does not have the permission to read the tenant data by default. Only the tenant user can access the connectivity model data. Doing this procedure enables the UI user to access the Connectivity Model data.

Procedure

1. Log into the Notebook node and open the notebook container.
2. Open the file `/home/cmopsadmin/cm/conf/input.txt` and edit the contents with the format:

```
<utilityId>;<username>;<role1>,<role2>;
```

Where `<utilityId>` is the utility to access, `<username>` is the UI user to access the utility, and `<role>` is the role of the UI user. For example:

```
cm_sample;Bob;admin,user;  
cm_sample;user1;user;  
cm_utility;Bob;admin,user;
```

3. Run this command with root.

```
/home/cmopsadmin/cm/bin/APP_manageUtilityAccess.sh  
/home/cmopsadmin/cm/conf/input.txt
```

Modifying the tenant user and changing tenant user passwords for the Connectivity Model application

Procedure

1. Log in App Node and open the IFELdapNode Docker container.

```
sudo docker exec -it IFELdapNode bash
```

2. To change the password for the user `cmtestuse`.
 - a) Edit the `/tmp/temp.ldif` file to add LDAP data and commands.
 - b) Add the script to the file:

Where the `passwdOrdChange` is the new password.

```
dn:cn=cmtestuser,ou=tenant,dc=ibmiot,dc=com  
changetype: modify  
replace: userpassword  
userpassword: passwdOrdChange
```

- c) Using the command to add the contents of the `temp.ldif` file into LDAP Server, replace `{LDAP_PASSWORD}` with actual `ldapServerPassword` which was provided during install in the `config.properties` file.

```
ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w {LDAP_PASSWORD} -f /tmp/temp.ldif
```

3. Create a LDIF file to the modify the group or user.

```
touch /tmp/temp.ldif
```

4. To remove the user `cmtestuser` from the `cm` group.
 - a) Edit the `/tmp/temp.ldif` file to add LDAP data and commands.
 - b) Add the script to the file:

```
dn:cn=cm,ou=application,dc=ibmiot,dc=com  
changetype: modify  
delete: memberUid  
memberUid: cmtestuser
```

- c) Using the command to add the contents of the `temp.ldif` file into LDAP Server, replace `{LDAP_PASSWORD}` with actual `ldapServerPassword` which was provided during install in the `config.properties` file.

```
ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w {LDAP_PASSWORD} -f /tmp/temp.ldif
```

5. To delete the user `cmtestuse` from the system you directly use the `ldapdelete` command:

```
ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD}
"cn=cmtestuser,ou=tenant,dc=ibmiot,dc=com"
```

Mapping user groups to license types

IBM IoT for Energy and Utilities has standard user licenses and limited user licenses. To generate usage information for the IBM License Metric Tool, you must map each user role group to the relevant license type in the `slmtag_groups.properties` file on the application server.

Before you begin

For more information about license usage metrics in IoT for Energy and Utilities, see [“License usage metrics”](#) on page 79.

About this task

To map a user role group to a license type, edit the `slmtag_groups.properties` file that is packaged in the `ife_service_ui.war` folder. There are two properties in the file: `groups_StandardUser` is the property for the standard user license, and `groups_LimitedUser` is the property for the limited user license.

Procedure

1. Log on to the application server as a user who has edit access to the `slmtag_groups.properties` file, for example, the `wlp` user.
2. Edit the `slmtag_groups.properties`. By default the file is in `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_frwk_app.ear/ife_service_ui.war/WEB-INF/classes/services` file.

The default file content maps the `admins` sample group to the standard user license, and maps the `users` sample group to the limited user license:

```
groups_StandardUser=admins
groups_LimitedUser=users
```

3. To map a group to the standard user license, add the group name as a value for the `groups_StandardUser` property. Use a comma as the delimiter between group names. For example, `groups_StandardUser=standardUserGroup1,standardUserGroup2`.
4. To map a group to the limited user license, add the group name as a value for the `groups_LimitedUser` property. Use a comma as the delimiter between group names. For example, `groups_LimitedUser=limitedUserGroup1,limitedUserGroup2`.

Results

Usage information for the two types of licensed users in IoT for Energy and Utilities is generated for the IBM License Metric Tool.

Connecting IBM IoT for Energy and Utilities to a GIS system

You can change the map default configuration for IBM IoT for Energy and Utilities for the applications Asset Performance Management, Asset 360 for Wind, and Connectivity Model.

Asset Performance Management map, including Asset Investment.

For Asset Performance Management the location for the map configuration is in: `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_ah_app.ear/ife_ah_web.war/config/model.json`.

For Asset Investment the location for the map configuration is in: /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_aip_app.ear/ife_aip_web.war/config/model.json

The map features for Asset Performance Management use OpenLayers 3.5. IoT for Energy and Utilities can connect to all GIS systems that supports OpenLayers 3.5.

The controls for the map features are:

center

Specifies the initial center point. Center is an array representing xy coordinates. Example: [-83.27366, 42.62893].

zoom

Specifies the initial zoom level, the range of values are 1 to maxZoom.

maxZoom

Specifies the maximum zoom level and is dependent on the baseLayerUrl provider.

baseLayerUrl

Specifies the base layer URL. Asset Performance Management uses xyz layers by default. The layer source for tile data with URLs in a set XYZ format that are defined in a URL template. Must include {x}, {y} or {-y}, and {z} placeholders.

```
"map": {
  "center": [-83.44, 42.60],
  "zoom": 12,
  "maxZoom": 19,
  "baseLayerUrl": "///server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/tile/{z}/{y}/{x}"
```

Wind 360 map

For Wind 360 the map configuration is in /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/wind_page.war/config/monitor/center.json

The map features for Wind 360 use OpenLayers 3.5.

center

Specifies the initial center point. Center is an array representing xy coordinates. Example: [-83.27366, 42.62893].

zoom

Specifies the initial zoom level, the range of values are 1 to maxZoom.

maxZoom

Specifies the maximum zoom level and is dependent on the baseLayerUrl provider.

baseLayerUrl

Specifies the base layer URL. Wind 360 uses xyz layers by default. The layer source for tile data with URLs in a set XYZ format are defined in a URL template. Must include {x}, {y} or {-y}, and {z} placeholders.

```
"Layers": [
  {
    "type": "ol.source.XYZ",
    "visible": true,
    "title": "XYZ",
    "Parameters": {
      "maxzoom": 13,
      "url": "http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/tile/{z}/{y}/{x}"
    }
  }
],
"view": {
  "center": [-8.8090, 33.2234],
  "zoom": 2
}
```

Connectivity Model map

The map for the Connectivity Model configuration depends on the tenant. Each tenant has its own map configuration. You can update the default map configuration using step 5 of Adding a Connectivity Model tenant user, “5” on page 69.

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values
    ('${utilityname}','${map_center}','${map_min_zoom}','${map_max_zoom}','${map_default_zoom}')
where utility = '${utilityname}';
```

map_center

The center of map, for example: [-83.44, 42.60]

map_min_zoom

The minimum zoom setting for the map. For example 1.

map_max_zoom

The maximum zoom setting for the map. For example 22.

map_default_zoom

The default zoom setting for the map. For example 8.

For example: to change the utility cm_sample, you need to run the following sql on hbase:

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values
    ('cm_sample', '[-83.44, 42.60]', '1', '22', '8') where utility = 'cm_sample';
```

Monitoring system status and backing up the system

The monitoring of the system is fundamental to the maintenance of the solution and the applications.

We recommend to monitor the system on three level: infrastructure, middleware and web link URL. These items are recommended to be monitored to maintain the solution for each application and status of each node.

Monitoring the Asset Performance Management and Asset 360 for Wind applications

These items are monitored to maintain the solution and status of each node for the Asset Performance Management and Asset 360 for Wind applications.

Infrastructure

SPSS Node, BI Node, IIB Node, DB Node

SPSS Node, BI Node, App node, DB Node

CPU

Disk

Memory

Network connectivity

Process

cpu_iowait

hostname_changed

max_open_file

max_processes

system_free_memory

system_swap_free

system_swap_pfree

system_total_memory

system_uptime
uname_changed

Middleware

SPSS node

status_spss_modeler_server

BI node

status_cognos_server

IIB node

App node

status_ldap_server

status_http_server

status_liberty_server_framework_server

status_mq_broker

status_mq_manager

status_pmo_broker

status_pmo_queue_manager

DB Node

status_DB2_server

Web links URL

Asset Health & Wind 360 applications

URL links

Backing up the system for the Asset Performance Management and Asset 360 for Wind applications

You must backup important information and data on a regular basis.

A backup of data can protect you against system failure and accidental loss. These are the IoT for Energy and Utilities items that you should consider for backing up. For IoT for Energy and Utilities we use docker containers to do the installation. So the directories to backup are not on the host directly, you need to go to different containers to get them, and may need to backup the same directories in different containers.

Item backup plan

App node

log: /var/log/

IIB server: /opt/IBM/IIB/10.0.0.7/

IHS server: /opt/IBM/HTTPServer/conf/

Message queue manager: /opt/mqm/

LDAP server, need to backup the data in ldap server, and the folders: /etc/openldap/, /opt/ldap/

Liberty server: /opt/IBM/WebSphere/Liberty

IoT for Energy and Utilities: /opt/IBM/energy

SPSS Node

log: /var/log/

IoT for Energy and Utilities: /opt/IBM/energy/

SPSS modeler: /usr/IBM/SPSS/

DB Node

IFEDB: /home/db2inst1

BI Node

log: /var/log/

Cognos: /opt/ibm/cognos/analytics/deployment

Monitoring the Connectivity Model application

These items are monitored to maintain the solution and status of each node for the Connectivity Model application.

Infrastructure

Ambari node and all HDP nodes, including HDP slave, master, notebook, and client nodes

CPU

Disk

Memory

Network connectivity

Process

cpu_iowait

hostname_changed

max_open_file

max_processes

password_file_changed

running_processes

system_free_memory

system_swap_free

system_swap_pfree

system_total_memory

system_uptime

uname_changed

HDP Services

HDFS node

Hbase

Web links URL

Connectivity Model URL link

URL link

Zeppelin URL link

URL link

Ambari interface URL link

URL link

Jupyter notebook URL links

URL links

Backing up the system for the Connectivity Model application

You must backup important information and data on a regular basis.

A backup of data can protect you against system failure and accidental loss. These are the IoT for Energy and Utilities items that you should consider for backing up. For IoT for Energy and Utilities we use docker

containers to do the installation. So the directories to backup are not on the host directly, you need to go to different containers to get them, and may need to backup the same directories in different containers.

Item backup plan

App node

log: /var/log/
IIB server: /opt/IBM/IIB/10.0.0.7/
IHS server: /opt/IBM/HTTPServer/conf/
Message queue manager: /opt/mqm/
LDAP server, need to backup the data in ldap server, and the folders: /etc/openldap/, /opt/ldap/
Liberty server: /opt/IBM/WebSphere/Liberty
IoT for Energy and Utilities: /opt/IBM/energy

Ambari Node

log: /var/log/
ambari-server: /var/lib/ambari-server
KDC server keytabs: /etc/security/keytabs
kdc server configuration: /var/kerberos, /etc/krb5.conf

HDP master and slave nodes

HDP services configuration: /etc/*/conf
keytab files: /etc/security/keytabs
log: /var/log, /hadoop/yarn/log
ambari-agent: /var/lib/ambari-agent/
ldap-client: /etc/openldap/
sssd config: /etc/sss/sss.conf
kdc client: /etc/krb5.conf, /etc/krb5.conf.d/, /var/kerberos/
Hadoop files: /hadoop

Notebook node

notebook: /usr/hdp/current/zeppelin-server/notebook
CM: /home/cmopsadmin
tenant users: /home/<tenant users>
HDP services configuration: /etc/*/conf
keytab files: /etc/security/keytabs
log: /var/log, /hadoop/yarn/log
ambari-agent: /var/lib/ambari-agent/
ldap-client: /etc/openldap/
sssd config: /etc/sss/sss.conf
kdc client: /etc/krb5.conf, /etc/krb5.conf.d/, /var/kerberos/
Hadoop files: /hadoop

License usage metrics

IBM License Metric Tool helps Passport Advantage clients determine their full and sub-capacity PVU licensing requirements.

Learn more: [IBM License Metric Tool](#).

IBM IoT for Energy and Utilities

For more information about using IBM License Management Tool, see the [IBM License Management Tool 9.0 Knowledge Center](#).

When IBM IoT for Energy and Utilities is running, license management information is logged every day to the /opt/IBM/energy/slmtags directory on the application server in one tag file that is created for IFE framework and applications.

IoT for Energy and Utilities framework user and asset information

The file ae796422b666e95033a951734467639f.slmtag contains information for three types of usage:

Standard user

The usage information that is logged is the number of licensed standard users in the system.

Limited user

The usage information that is logged is the number of licensed limited users in the system.

Asset analytics

The usage information that is logged is the number of managed assets in the system. For the IBM IoT for Energy and Utilities this value is always 0.

Note: The numbers of licensed standard users and limited users in the system is retrieved from the basic user registry that is deployed with IoT for Energy and Utilities. To ensure the accuracy of these numbers, the configuration file that maps user groups to license types must be kept up to date.

The following content is an example of usage information from the ae796422b666e95033a951734467639f.slmtag file:

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
  <PersistentId>e137414b35d140dca5fd631df1098e0d</PersistentId>
  <Name>IBM IoT for Energy and Utilities</Name>
  <InstanceId>/opt/IBM/energy</InstanceId>
</SoftwareIdentity>
<Metric logTime="2017-07-20T17:20:55+08:00">
  <Type>AUTHORIZED_USER</Type>
  <SubType>Standard User</SubType>
  <Value>1</Value>
  <Period>
    <StartTime>2017-07-20T17:20:55+08:00</StartTime>
    <EndTime>2017-07-20T17:20:55+08:00</EndTime>
  </Period>
</Metric>
<Metric logTime="2017-07-20T17:20:55+08:00">
  <Type>AUTHORIZED_USER</Type>
  <SubType>Limited User</SubType>
  <Value>3</Value>
  <Period>
    <StartTime>2017-07-20T17:20:55+08:00</StartTime>
    <EndTime>2017-07-20T17:20:55+08:00</EndTime>
  </Period>
</Metric>
<Metric logTime="2017-07-20T17:20:55+08:00">
  <Type>ASSET</Type>
  <SubType>Assets in Asset Health</SubType>
  <Value>41278</Value>
  <Period>
    <StartTime>2017-07-20T17:20:55+08:00</StartTime>
    <EndTime>2017-07-20T17:20:55+08:00</EndTime>
  </Period>
</Metric>
```

Changing the details of the login page

You can customize the text and image of the login page of IBM IoT for Energy and Utilities.

About this task

Important: You must have administrative "wlp" user or root user rights for the IIB server.

Procedure

1. Log into the IIB node with administrative "wlp" or root user rights.
2. To change the product name:
 - a) Open the `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_frwk_app.ear/ife_frwk_web.war/js/nls/` directory.
 - b) Open the `AdminUI.js` file in a text editor.
 - c) Edit the line `login_project_title:"IoT for Energy and utilities"` by editing the "IoT for Energy and utilities".
For example: `login_project_title:"IoT for Energy and utilities of Company Name"`.

3. To change the background image:

- a) Open the `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_frwk_app.ear/ife_frwk_web.war/images/login` directory.
- b) Backup the current image with the command:

```
mv ioc_login_background_19201280.jpg ioc_login_background_19201280.jpg.bak
```

- c) Rename the background image you want to use with the command:

```
mv <the name of your image>.jpg ioc_login_background_19201280.jpg
```

4. Restart the Liberty server with the command:

Important: You must have administrative "wlp" user or root user rights for the IIB server. For IoT for Energy and Utilities 2.5 and later release, we only use the wlp to start, stop, or edit the liberty profile.

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server  
/opt/IBM/WebSphere/Liberty/bin/server start framework_server
```

Optimizing performance of the IBM Db2 database

You can use the **update configuration** and the **ALTER bufferpool** commands to optimize the IBM Db2 settings for the Asset Performance Management application in IBM IoT for Energy and Utilities.

Before you begin

You need access to the DB node for IoT for Energy and Utilities as the instance owner, for example, `db2inst1`.

The minimum hardware configuration for the database is 16 CPU cores and 32G of memory.

If hyper-thread functionality is enabled in the core and you are using only one core, the **DFT_DEGREE** parameter must specify the degree of intrapartition parallelism based on the number of processors and the **NUM_IOCLEANERS** parameter must specify the number of page cleaners. If these parameters are not specified, then the Db2 SQL optimization ignores the intrapartition parallelism.

Procedure

1. Log into the db node and change the user to db2inst1.
2. Run the following command to connect to Db2.

```
db2 connect to ifedb
```

3. Run the following commands:

```
db2 update db cfg for ifedb using DFT_DEGREE any
db2 update db cfg for ifedb using SELF_TUNING_MEM OFF
db2 update db cfg for ifedb using SHEAPTHRES_SHR 614400 automatic
db2 update db cfg for ifedb using SORTHEAP 204800 automatic
```

```
db2 ALTER bufferpool IBMDEFAULTBP size 50000 automatic
db2 ALTER bufferpool BP32K_01 IMMEDIATE size 150000 automatic
db2 ALTER bufferpool BP32K_02 IMMEDIATE size 50000 automatic
db2 ALTER bufferpool BP32K_03 IMMEDIATE size 250000 automatic
db2 ALTER bufferpool BP32K_04 IMMEDIATE size 300000 automatic
```

4. Reset the current connection:

```
db2 connect reset
```

Performing IBM Watson IoT Platform integration administration

You can integrate IBM IoT for Energy and Utilities with IBM Watson IoT Platform to collect data from connected devices to provide data management, visualizations, and analytic capabilities from existing devices within Watson IoT Platform.

After you create an integration, all connected devices are detected and data collection begins. The files are automatically parsed to identify the variables, attributes, and dimensions.

The integration with Watson IoT Platform enables IoT for Energy and Utilities to instantly save the data on the IFE server. Every event is saved as a single file. When the connection is disconnected, the subscription is also disconnected and IoT for Energy and Utilities no longer saves the events.

Creating integrations

You can integrate IoT for Energy and Utilities on Cloud with Watson™ IoT Platform to collect asset data from devices that are connected to Watson IoT Platform.

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as an administrator.
2. Click **Administration** > **IOT** > **Add Integration**.
3. Type the following information.
 - a) The name of the integration.
 - b) The organization ID.
 - c) The API key.
 - d) The authentication token.

What to do next

After you create an integration with a Watson™ IoT Platform organization, you need to add a subscription to configure the collection of data from the devices in that organization before you can connect to the integration.

Adding a subscription to an integration

After you integrate with Watson™ IoT Platform, you need to configure the integration by adding a subscription to the devices in Watson™ IoT Platform.

Procedure

1. Click **Add Subscription** to the integration you need.
2. Type the name of the subscription.
3. The name and location of the subscription file is automatically generated. You can edit the subfolder name, but you must use the location of the integration folder.
For example, if the current integration name is INTG1, the subscription name is SUB1 then the address shows as INTG1/SUB1.
4. Select the devices for this integration.
You can add all devices, or make a selection.
5. Click **Save** to save the subscription choices.

Results

You can view the details of the devices in the list by **Device ID**, **Device Type**, **Events**, and **Date Added**.

Editing integrations

After you create an integration with a Watson™ IoT Platform organization, you can manage and edit the integration, for example, to change the devices from which you collect data.

Procedure

1. Click **Administration > IOT**.
2. Click the integration card that you need to edit.
3. If the integration is connected, click **Disconnect**.
4. Click **Edit** and edit as necessary.
5. Click the **Back** icon to exit the current configuration.

What to do next

You can click **Connect** to reconnect an integration.

Managing integrations

You can manage your integrations by connecting to and disconnecting from an integration. You can also delete an integration.

Procedure

1. Click **Administration > IOT**.
2. Hover over an integration card to view the connection status.
You can disconnect or connect to an integration, depending on the status, and delete an integration.
3. If you select **Connect** or **Disconnect**, the status changes without further input.
4. If you select **Delete**, you are asked to confirm before deletion.
5. Click the **Back** icon to exit the current configuration.

Results

The event files are found on the IIB server at the location `/opt/IBM/energy/IOT/<the subscription location user input from UI>/<Device type name>/<Device id>/<event name>/yyyy_MM_dd_HH_mm_ss_SSS.txt`.

Archiving the event files

The archive facility in IBM IoT for Energy and Utilities sends all the event files that are created before a target date to an archive. If no target date given then IoT for Energy and Utilities sends all the event files that are created before today's date.

Procedure

1. Find the archive script at the location `/opt/IBM/energy/IOT/archive.sh`.
2. Use the following syntax to start the archive:
 - The script `archive.sh /opt/IBM/energy/IOT` creates the archive to include all event files before the present date.
 - The script `archive.sh /opt/IBM/energy/IOT <target date yyyyymmdd>` creates the archive to include all event files before the target date.

Results

You can find the archive at the location `/opt/IBM/energy/IOT/Archive/<date>.zip`.

Reuse existing LDAP Server for IBM IoT for Energy and Utilities

You can use an existing LDAP server for application authentication in IoT for Energy and Utilities.

Preparing for configuring the LDAP registry for IBM IoT for Energy and Utilities

You need to make sure that you can access the LDAP server from the Liberty server and then gather the information that will be used for configuration.

Procedure

1. Login the Liberty node and run the command:

```
telnet <ldap server> portnumber
```

where `<ldap server>` is the name of the LDAP server and the `portnumber` is the port you want to test the connection to.

Important: If you are using a Docker environment, this command must be run in the IFEAppNode container.

2. Add the Signer certificate for the LDAP server to the truststore for the liberty server.

The following example is with OpenLdap.

- a) Run the command to find the location of the `ldap.conf` file:

```
find / -name ldap.conf
```

- b) Open the `ldap.conf` file, and check the location of the `cacert.pem` file for `TLS_CACERT`.


```

@IFELdapNode:/
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASE    dc=example,dc=com
#URI     ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT    12
#TIMELIMIT    15
#DEREF        never

TLS_CACERTDIR    /etc/openldap/certs

# Turning this off breaks GSSAPI used with krb5 when rdns = false
SASL_NOCANON    on
BASE    dc=ibmiot,dc=com
URI     ldaps://IFELdapNode.ife:636 ldap://IFELdapNode.ife
TLS_CACERT /etc/openldap/cacerts/cacert.pem

```

Figure 10. The location of cacert for TLS_CACERT

- c) Copy the cacert . pem file to the server that has Liberty on it for importing into truststore later. The cacert . pem for OpenLdap looks as follows:

```

-----BEGIN CERTIFICATE-----
MIIDLTCcAhWgAwIBAgIJAL9PxE/vLoR0MA0GCSqGSIb3DQEBCwUAMC0xCzAJBgNV
BAYTAKNOMRAdgYDVQQIDAdCZWlqaW5nMQwwCgYDVQQKDANJQk0wHhcNMTcwOTA4
MDCwODIwHhcNMjcwOTA2MDCwODIwWjAtMQswCQYDVQQGEwJDTjEOMAA4GA1UECAWH
QmVpamluZzEMMAoGA1UECgwDSUJNMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAr+UmpGi3rMLBNp0ygp1N70iNKmE55qZu56IxGX+byRBRN9Qi90uNuJWI
rPGsd/2IBBzELCGNT31rbfySrdHdo9+YsJicuaZH6RAH4L7K3cyQfVMP6h9qqixc
WHu2WgVAvPbMkWKmGY/HfIcGo1hYESo7J9a9LcCQzNM90d6Ho7Qo3IUnuiZ0wC3
pRrAXyVc4ywh6xfU05ZSxDdjGu2vqG5Z4xr1gmtCHOqYqUIEw1fT4891v+C5RNS3
sN7Lp5//OPvMR6CveVJiH88ENA7ukOR3A5FQrDWeTUyywEq+jTeYMD1D0DhCmiC8
JSzeJWYpDFRYmDPMQYvFnSeToUswIDAQABO1AwTjAdBgNVHQ4EFgQUsnj0aXnF
qnfxEOREJ0i+rRwDviYwHwYDVR0jBBgwFoAUSnj0aXnFqnfxEOREJ0i+rRwDviYw
DAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAgEhVSEYyqv/QvYrYmb6+L
aWoBKh9jLVQuX1n1E6MiejoSio0RW53GPG2Ejqkoa84qWC72eg/pQXBVPb0Tyefj
B2U+s7ymnCKbQU15bpgH27/PhtwV4410pew/++fjDf1/ts9/yutKaVBgiXdxSsgv
rIet7MB2FBtgNFpb6sFKhSh5zX67anfEPSxp6pJx/FBH1W65tXC90HtfnF1/7U7L
MZNMRhMJwfiVgyWQPWdsecqpb9FnE017XLCiDSIb0vLVhc01TfIux3y60WVLocI+
eMHj9rf01700/VqC3iIa+kVqMioCGz2SdBATzfG3wCfjWD5Y8dunPhHUsAnGAocU
8g==
-----END CERTIFICATE-----

```

- d) Copy the cacert . pem file onto the /tmp directory of the Liberty node.
If you use the Docker environment for IoT for Energy and Utilities 2.5, do these steps to copy the cacert . pem file:
 - 1) Copy the cacert . pem to the AppNode.
 - 2) Copy the cacert . pem from the AppNode to IFEAppNode container in the /tmp directory on the App node.

```

sudo docker cp /tmp/cacert.pem IFEAppNode:/tmp

```

3. Gather information from the LDAP Server that will be used for configuring ldapRegistry.xml.

- a) Get the user group information.

Consult with your LDAP administrator for the user group name or search on the LDAP server:

- 1) You need to know how users are grouped in the LDAP server, for example in OpenLDAP the alternatives are `objectClass posixGroup` and `objectClass groupOfNames`.
- 2) Run the command to search, and find the corresponding entries that matches `objectClass` value.

```
ldapsearch -x
```

An example output looks as follows:

```
.....  
# admins, ife, application, ibmiot.com  
dn: cn=admins,cn=ife,ou=application,dc=ibmiot,dc=com  
objectClass: top  
objectClass: groupOfNames  
member: cn=Bob,cn=ife,ou=application,dc=ibmiot,dc=com  
cn: admins  
.....
```

- 3) Consult your LDAP administrator to understand which user groups or users you need to access the IoT for Energy and Utilities offering.
- b) Get the LDAP server connect information.

- 1) Check that the LDAP server is enabled for SSL communication and get LDAP port number. For example: OpenLDAP normally uses port 389 for non SSL by default and port 636 for SSL communication. In IoT for Energy and Utilities version 2.5 the configuration is in the `/etc/openldap/ldap.conf` file. For example:

```
# Turning this off breaks GSSAPI used with krb5 when rdns = false  
SASL_NOCANON    on  
BASE            dc=ibmiot,dc=com  
URI             ldaps://IFELdapNode.ife:636 ldap://IFELdapNode.ife  
TLS_CACERT      /etc/openldap/cacerts/cacert.pem
```

Figure 11. An example configuration for SSL communication

- 2) Obtain the Base DN or Bind attributes for the LDAP server. Consult your LDAP administrator for this information.
- 3) Obtain the administrative password for LDAP to authenticate with the LDAP server. Consult your LDAP administrator for this information. For example, in OpenLDAP, the configured is in `/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif` file as the `olcRootPw` attribute.

```
AUTO-GENERATED FILE - DO NOT EDIT!! Use ldapmodify.
# CRC32 873b478d
dn: olcDatabase={2}hdb
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=ibmiot,dc=com
olcRootDN: cn=Manager,dc=ibmiot,dc=com
olcDbIndex: objectClass eq,pres
olcDbIndex: ou,cn,mail,surname,givenname eq,pres,sub
structuralObjectClass: olcHdbConfig
entryUUID: 307b1d96-5bfe-1037-8064-93f78f184384
creatorsName: cn=config
createTimestamp: 20171112140448Z
entryCSN: 20171112140448.7281712#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20171112140448Z
olcRootPW: passw0rd
~
~
~
"etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif" 19L, 623C
```

Figure 12. The attribute `olcRootPw`

- 4) Set the baseDN for the point from where a Ldap server searches for users. For example: , if you have a bindDN cn=admin, dc=example, dc=com, then you can use dc=example, dc=com as baseDN.

4. Retrieve the password details.

The password for access to the liberty keystore key . jks is provided during installation. It is used when you import the Signer certificate into the liberty keystore. In IoT for Energy and Utilities V2.1, this password is set during installation. In IoT for Energy and Utilities V2.5 has a default value that is passw0rd.

Configuring the LDAP user registry in Liberty

Before you can add access for a new user group, you need to configure the LDAP user registry.

Procedure

- 1. Import the LDAP certificate.

- a) Locate the path to keytool.
- b) Run the commands to import the LDAP certificate and specify a password for the keystore ldapKeyStore . jks. Replace the path value to where cacert . pem is copied and the password for access to key . jks.

For example, for OpenLdap use:

```
IM_Java=`find / -name jre_* | grep InstallationManager/eclipse`
$IM_Java/jre/bin/keytool -import -alias ldap_cert -noprompt -storepass passw0rd -
file /tmp/cacert.pem -keystore /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/
resources/security/ldapKeyStore.jks

$IM_Java/jre/bin/keytool -import -alias ldap_cert -noprompt -storepass
<setYourLdapKeystorePassword> -file /tmp/cacert.pem -keystore /opt/IBM/WebSphere/
Liberty/usr/servers/framework_server/resources/security/key.jks
```

2. On the Liberty Node, run the commands to comment out the basicRegistry in server_ife_frwk.xml and server_wind_web.xml files.

Run this command for all applications as well as Asset 360 for Wind:

```
sed -i 's/<basicRegistry/<!-- basicRegistry/g' /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server_ife_frwk.xml
sed -i 's/basicRegistry>/basicRegistry -->/g' /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server_ife_frwk.xml
```

Run this command for Asset 360 for Wind only:

```
sed -i 's/<basicRegistry/<!-- basicRegistry/g' /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server_wind_web.xml
sed -i 's/basicRegistry>/basicRegistry -->/g' /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server_wind_web.xml
```

Note: The user groups in the server_ife_frwk.xml file apply to all applications as well as Asset 360 for Wind. The user groups in server_wind_web.xml file apply for Asset 360 for Wind only.

3. Create a file: ldapRegistry.xml.

Replace the **Bold** sections with actual values.

```
<server>
  <featureManager>
    <feature>ldapRegistry-3.0</feature>
    <feature>appSecurity-2.0</feature>
    <feature>ssl-1.0</feature>
  </featureManager>
  <ldapRegistry ldapType="Custom" port="ldap_port" realm="WebRealm"
host="ldap_hostname" bindDN="bindDN" bindPassword="ldapAdminPassword"
id="ldapid" baseDN="baseDN" sslEnabled="sslEnabled" sslRef="ldapssl">
    <customFilters userFilter="(&cn=%v)(objectclass=person)" groupFilter="(&cn=%v)
(|(objectclass=posixGroup)(objectclass=groupOfNames))"
groupMemberIdMap="posixGroup:memberUid;groupOfNames:member">
    </customFilters>
  </ldapRegistry>

  <ssl keyStoreRef='ldapKeyStore' id="ldapssl"></ssl>
  <keyStore id="ldapKeyStore" location="{server.config.dir}/resources/security/
ldapKeyStore.jks" type="JKS" password="ldapKeystorePassword" />

  <ltpa keysFileName="{server.config.dir}/resources/security/ltpa.keys"
keysPassword="SetYourLtpaPassword"></ltpa>
</server>
```

Note: 1. If you do not want to use plain text passwords, you can use securityUtility to get an encrypted string to use in the xml file, for example:

```
/opt/IBM/WebSphere/Liberty/bin/securityUtility encode <password string>
```

Note: 2. Update customFilters segment to filter out the user groups or users in LDAP server who need to access IoT for Energy and Utilities. Check the Liberty knowledge center for details: https://www.ibm.com/support/knowledgecenter/en/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/twlp_sec_ldap.html.

Note: 3. IoT for Energy and Utilities version 2.5 has a built-in OpenLDAP server. Liberty uses this OpenLDAP server for authentication. The ldapRegistry.xml in IoT for Energy and Utilities version 2.5 is configured as follows:

```

<server>
  <featureManager>
    <feature>ldapRegistry-3.0</feature>
    <feature>appSecurity-2.0</feature>
    <feature>ssl-1.0</feature>
  </featureManager>

  <ldapRegistry ldapType="Custom" port="636" realm="WebRealm"
    host="IFELdapNode.ife" bindDN="cn=Manager,dc=ibmiot,dc=com" bindPassword="{xor}Lz4sLChvLTs="
    id="ldapid" baseDN="dc=ibmiot,dc=com" sslEnabled="true" sslRef="ldapssl">
    <customFilters userFilter="( &amp; (cn=%v) (objectclass=person))"
      groupFilter="( &amp; (cn=%v) (| (objectclass=posixGroup) (objectclass=groupOfNames)))"
      groupMemberIdMap="posixGroup:memberUid;groupOfNames:member">
    </customFilters>
  </ldapRegistry>

  <ssl keyStoreRef='ldapKeyStore' id="ldapssl"></ssl>
  <keyStore id="ldapKeyStore" location="{server.config.dir}/resources/security/ldapKeyStore.jks" type="JKS" password="{xor}Lz4sLChvLTs" />

  <ltpa keysFileName="{server.config.dir}/resources/security/ltpa.keys" keysPassword="{xor}Lz4sLChvLTs"></ltpa>
</server>

```

Figure 13. The details for the LDAP Registry

4. Regenerate ltpa.keys.

- a) Delete the previous ltpa.keys file in the folder

/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/resources/security/

```
rm -rf /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/resources/security/ltpa.keys
```

The ltpa.keys file is generated again after you restart framework_server.

- b) Restart the framework_server to update server.xml to include ldapRegistry.xml, and restart framework_server for the LDAP user registry to take effect.

```
sed -i '$d' /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server.xml
addConfigure='\t<include location="{server.config.dir}/ldapRegistry.xml" />\n</server>'
echo -e $addConfigure >> /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server.xml
```

```
/opt/IBM/WebSphere/Liberty/bin/server stop framework_server
/opt/IBM/WebSphere/Liberty/bin/server start framework_server
```

Managing the Standard Operating Procedures

A Standard Operating Procedure (SOP) is a set of instructions that describes all the relevant steps and activities of a process or procedure.

When you define an SOP, you define activities that are included in the SOP. SOP enables an administrator to organize personnel, information, and tasks in response to events and incidents in order to achieve a comprehensive control of the operation. A SOP comprises of these components:

Standard Operating Procedure Definition

An SOP definition is the template that is used when a SOP is instantiated in response to a particular occurrence. A SOP Definition is made up of activities that are described by Activity Definitions.

Activity Definition

A SOP Definition contains one or more Activity Definitions. An activity definition sets the individual instructions that need to be performed as part of the SOP.

SOP Instance

A single Instance of an SOP in response to a particular event or occurrence. One SOP Definition can be used for many SOP Instances. An SOP Instance can be in one of these states.

- Active
- Started
- Stopped
- Completed
- Canceled

Activity Instance

An Activity Instance is the instantiation of a single Activity Definition. A single Activity Definition can be used to create multiple Activity Instances. An Activity Instance can be in a number of states:

- Active
- Waiting
- Started
- Skipped
- Completed

References

Supplemental information which is relevant to a Standard Operating Procedure or Activity. References can also be used to define e-mail templates.

Roles

There are two abilities, Owners and Readers. These can be set against administrative and user roles.

- A Reader can monitor the activities that are associated with a standard operating procedure.
- An Owner can monitor and complete the activities that are associated with the standard operating procedure.

Activity Type

The Activity Type describes the response to the activity. The activities can be of different types and execution models. Any combination of different activities in an SOP is allowed.

- Manual: This type of activity must be manually carried out by the owner of the SOP.
- If-Then-Else Activity: A conditional activity that allows branching based on specific criteria. The user can choose which of the SOP definitions to instantiate when starting the activity. Either enter or select values for Then and Else.
- Alert Activity: This activity displays an e-mail template for the SOP owner to complete and send an email notification to predefined personnel.
- REST Activity: An activity that creates a REST service call. The user can specify the service URL and any required authentication information to be invoked when the activity is started.
- SOP Activity: An activity that starts another standard operating procedure.

Roles for Standard Operating Procedures

The abilities for each of the roles for SOPs are as follows:

SOP Administrator roles

- View and delete an SOP definition
- Launch, view, and edit an SOP instance
- Start and complete activities in an SOP instance

SOP author roles

- Create, edit, view, and delete an SOP definition
- Create an SOP draft
- View, edit, and delete an SOP activity
- Submit an SOP draft for approval
- Approve an SOP draft

Reference Librarian Role

Create shared references

Owner Roles (SOP definition)

- Create an SOP draft

- View, edit, and delete an SOP definition
- Edit and delete an SOP activity
- Submit an SOP draft for approval
- Approve an SOP draft
- Launch, view, and edit an SOP instance

Reader roles (SOP Definition)

- View an SOP definition
- View an SOP instance from My Activities
- View an SOP activity, provided the user has Reader role in the Activity definition

Owners roles (SOP activity)

- View an SOP instance from My Activities
- Start and complete activities in an SOP instance for their own activities from My Activities

Reader roles (SOP activity)

View SOP instance from My Activities

Approval life cycle for an Standard Operating Procedure

An SOP definition can assume different status during its life cycle.

- **Draft:** When the SOP is first created, a draft version is saved initially. From an approved version of an SOP, it is also possible to create another draft version, when it is necessary to change the SOP definition using the approved version as a base. A draft can be edited, submitted for approval, or discarded.
- **Pending approval:** This is a draft SOP definition submitted for approval, ready to be approved or disapproved. The name of the version is defined in this status and it will name the SOP definition version if approved. If this version is not approved, the SOP definition goes back to the draft version status.
- **Approved:** When an SOP definition is approved it is ready to be launched.



Figure 14. SOP Life Cycle

Chapter 4. Using Asset Performance Management application

The Asset Performance Management application shows how well a specific asset will provide its service in the future.

You have a direct visualization of the status of any asset from the reports and charts in the application. IBM IoT for Energy and Utilities IoT for Energy and Utilities application provides indices and numerical values that indicate the health and risk of asset and network failure. These reported values are:

- Health index
- Failure
- Criticality
- Risk

Health

Health is an index that is an aggregate score for the health of an asset and is calculated from the historical performance of the asset and the measured physical condition. The higher the value for the Health index, the less likely the asset will fail. Example factors that are used to calculate the health index are age, manufacturer, and overload time. The Health index returns values in the range 0 to 100 where 100 is as new condition and 0 is very poor condition.

Failure

Failure is the probability that the network will fail. The higher the value, the more likely the network fails. Failure is calculated from the probability that an particular asset will fail and the impact that failure has on other assets in the network. Failure is calculated from four probabilities: The probability of failure of an individual asset. This is calculated as $\{(100 - \text{Health index}) * \text{constant}\}$ The probability of failure of an asset downstream from the individual failing asset. The probability of failure of an asset upstream from the individual failing asset. The probability of the physical failure of a supporting asset. A supporting asset is one that gives physical support to the individual asset, for example, an overhead cable is physically supported by 2 poles. Failure returns values in the range 0 to 100 where 0 is no probability of failure and 100 is an imminent network failure.

Criticality

Criticality is a measure of the number of customers that are supported by an asset. Assets that support a greater number customers have a higher Criticality rating. As the number of downstream network nodes propagates, the Criticality rating of one asset is the summation of all Criticality rating of all downstream nodes, plus its own rating. Criticality is rated from 0 to 100 where 0 is no customers and 100 is all customers.

Risk

Risk is a measure of the risk to the business if a failure in the network occurs. The higher the value for the Risk the more risk there is to the business. Risk is a percentage value given by the product of the values for Failure and Criticality / 100. If Failure is 30 and Criticality is 65, then Risk is 19.5%.

If Criticality is much less in terms of customers, then if Failure is 30 and Criticality is 10, then Risk is 3%. The Risk returns values in the range 0 to 100 where 0 is no risk and 100 is a potential catastrophic risk.

Managing the custom analysis model

The **Custom Analysis Model** has the ability to download, upload, and delete a customized analysis model, manage the configuration of a model, and run an analysis for a configuration of a model.

Under the **Manage Model** tab you can upload your own customized SPSS streams and manage the folder hierarchy for a model.

Under the **Manage configurations** tab you can create parameters configurations for those SPSS models. One model can be used in more than one configuration using different parameters..

Under the **Run new analysis** tab you can run an analysis for a particular configuration.

Prerequisites for an analysis

1. The data loading for Asset Performance Management must be complete.
2. The input parameters for the asset health models of each asset class are:
 - `Curve_Params_<asset>.csv` Defines the parameters for the degradation curve generation for the asset class.
Note: Different subtypes can have different parameters. You can modify the parameter as necessary.
 - `AHI_Factor_<asset>.csv` Defines the parameters of factors that contributes to the asset health score.
Note: Different subtypes can have different parameters. You can modify the parameter as necessary.
 - The files `AHI_Factor_<asset>.csv` and `Curve_Params_<asset>.csv` are in the location: `/opt/IBM/energy/AHI/SPSS_stream/data/ah_input`.
3. The required parameter configuration for analysis:
 - Is the configuration for the spss modeler server.
The file includes configuration of:
 - The host name of server where the spss modeler runs
 - user name and password runs the modeler server
 - The location of the modeler batch.
 - `/opt/IBM/energy/AHI/SPSS_stream/conf/streamParams.cfg` Defines all the execution parameters that are needed when doing the analysis.

Example content for the `streamParams.cfg` file.

```
dsname=IFEDB
dsuser=db2inst1
dspwd=db2inst1
-log /opt/IBM/energy/AHI/SPSS_stream/log/AHBacth.log
-appendlog
ana_year=20

# streams
[/opt/IBM/energy/AHI/SPSS_stream/stream/CircuitBreaker/CircuitBreaker_AHI.stri]
csvFolder=/opt/IBM/energy/AHI/SPSS_stream/data/model_output
AHI_Factor_Weight=AHI_Factor_CircuitBreaker.csv
assetTable=CIM.CIRCUITBREAKER
asset_AHI_Factor_csv=AHI_factors_CB.csv
asset_AHI_csv=AHI_CB.csv
asset_Detail_csv=CircuitBreaker.csv
```

Important: The first 6 lines are for common parameters:

dsname

is the datasource name

dsuser

is for db2 user

-log

is modeler batch option, defines the log file

-appendlog

is modeler batch option, append log to above log file

dspwd

is datasource password for db2 user

ana_year

is the analysis scope, default is 20 years from current year

Note: You can use plain text to modify dspwd, or you can encrypt the password with the file `encrypt.sh`.

- `/opt/IBM/energy/AHI/SPSS_stream/conf/stream_model.cfg` Defines the execution order of the models. You can edit this file to add and remove the models to be run.
- The utility models are not required to be changed.

`/opt/IBM/energy/AHI/SPSS_stream/conf/stream_assetHealth_clearup.cfg` Defines the stream to clean up the asset health database.

`/opt/IBM/energy/AHI/SPSS_stream/conf/stream_assetHealth.cfg` Defines the stream to move the analysis results into the database.

Uploading a custom analysis model and stream

The custom analysis model up-loader has a folder management feature that lets you manage the different models and streams that you wish to upload.

About this task

Before you upload an analysis, create the folder hierarchy for the analysis: model and data stream.

Procedure

1. Click **Administration > Custom Analysis Model**
2. Click the **Manage Model** tab.
3. Click **Create folder** and type a name for the analysis model, for example APM.
4. Create the folder for the data, in the column **Create folder** click the and type a name for the data for example data.



Figure 15. Create folder icon in a row

5. Type the folder name for model data.
6. Create the folder for the data stream, in the column **Create folder** click the and type a name for the data for example stream.



Figure 16. Create folder icon in a row

7. Upload the data stream, click the **Upload** icon for the stream folder and select the str file from your source.

You can add one file per upload. If you want more than one file, you must add each one separately.

8. Repeat for the data folder.

Data folders can be in the form csv.

Setting the global parameters and configuring the analysis model

The Manage Configurations page is where you set the parameters for the stream and stream order.

About this task

You can organize the configurations in folders that you create.

Procedure

1. Click **Administration > Custom Analysis Model > Manage Configurations**
2. Type the name and description for the configuration.
This is the folder to manage the configuration parameters.
3. Under **Set Global Parameters**, click **Add new**.
Global parameters are used for all data streams.
4. Complete the global parameters for the configuration for Name and Command:

Name	Command
dsname	IoT4EUDB
dsuser	db2inst1
dspwd	pw4ibmioteusw
ana_year	20

Click **Add new** for each new parameter.

5. Under **Configure Model**, Click **Import**.
6. Make a selection the streams for the configuration that you need to import. The streams are from the **Manage Models** page.
7. When you have completed the selection, click **Import**.
8. Add the stream that you want to add the parameters for. Click on each of the imported streams as required.
The **Edit models** window opens.
9. Click **Add new**.
10. Type the parameters and commands for each of the required parameters.

Name	Command
csvFolder	/opt/IBM/energy/AHI/SPSS_stream/data/ model_output
AHI_Factor_Weight	AHI_Factor_CircuitBreaker.csv
assetTable	CIM.CIRCUITBREAKER
asset_AHI_Factor_csv	AHI_factors_CB.csv
asset_AHI_csv	AHI_CB.csv
asset_Detail_csv	circuitBreaker.csv

11. Click **Save** and **Save and back**

Running an analysis

You can either run an analysis from the **Manage Configuration** page or create a name for the analysis and select the configuration for the analysis you need to run.

Procedure

1. Click **Administration > Custom Analysis Model**.
2. In the **Manage Analysis** page, click **Run New Analysis**.
3. Type a name for the analysis and **Select configuration** from the drop down menu.
4. Click **Save and Run**.
5. When the analysis is complete, the time duration and log files are available.

Integrating with Visual Insights

IBM IoT for Energy and Utilities has the ability to integrate with IBM Visual Insights.

A field technician is able to upload images to Visual Insights and for those images to be used for analysis in IoT for Energy and Utilities.

The Workflow

The data scientist trains the model in Visual Insights. You need the information from Visual Insights for login credentials and score image service link. Refer to <https://www.ibm.com/support/knowledgecenter/SS5U3Qhttps://www.ibm.com/support/knowledgecenter/SSC5ZE/com.ibm.vi.doc/welcome.html> for more information.

The engineer creates the measurement type and reading table in IoT for Energy and Utilities and configures the relationship between them. He then loads the measurement and adds inspection rule in the custom database.

The data scientist updates the SPSS Asset Health Index stream for the asset class and adds the factor from Visual Insights.

The reliability manager uploads the images that the field technician created, and verifies and confirms the results.

The analysis program is run and the SPSS asset health index is updated with a new score.

In report, you see the new health score with the Visual Insights factors added in the new SPSS AH stream.

Defining a measurement reading table name for Visual Insights integration

The measurement reading table is used to store the data from measurements made on an asset class.

About this task

The tab **Measurement Reading** is where you define the table name and description of the measurement. The measurement reading is used when defining the measurement type for an asset class.

Procedure

1. In IBM IoT for Energy and Utilities, select **Administration > Custom Data Model**.
2. Select the **Measurement Reading** tab, and click the **Add New** icon.
3. Add the **Table Name**.

The table name must contain an existing schema name and a table name, for example `cim.vi_measurement_01` where `cim` is an existing schema name.

4. Add the **Description** of the measurement table.
5. Click **Save**.

6. Click the edit button to add the table column to the measurement reading table.
7. Type a name for the **Column Name**.
8. Select the **Column Type** and **Column Length**.
9. Click **Save**.
10. Click **save & back** to exit.

Defining the measurement type and associating it to the reading table

You can add a measurement type to an asset class and associate it to a reading table.

About this task

This procedure adds a measurement type to the .csv reading file located here: /opt/IBM/energy/data/<your_directory_name>/reading. The columns 1 and 2 are generated with default names, **measurement** and **timestamp**, you are creating additional measurement types from the names already created in the reading .csv file.

Procedure

1. Select the **Asset Classes** tab and select the row for the asset class you want to edit and click the **Edit** icon.
2. Click the **Measurement** tab and click the **Add New** icon.
3. In the **Measurement Type** field type the measurement description. The letter a to Z and the number 0 to 9 are supported.
4. Select the target table.
The target table is the one you defined for the **Table Name** in add Measurement Reading table.
5. Click the **Add New** icon.
6. In the **Source Column** type the inspection name from the tags in the Visual Insights . The code is a column name in the .csv reading file. The code is a unique alpha-numeric code for the measurement type property. The letters a to Z and numbers 0 to 9 are supported.
7. Select **Number** in the **Type**. The type is the column type in the .csv file.
8. Select the **Target Column**. This is the column in the database to where you want to import the .csv file.
9. Click **Save > save & back** to exit.

Importing the asset measurements

You use the data loader in IBM IoT for Energy and Utilities to import measurement data in csv file format.

Before you begin

You need to have the measurement data file prepared with your data before you can import it. You can find a template for the file in IoT for Energy and Utilities **Administration > Custom Data Model > Upload Data Source > Download Template**. The template files for measurements are located in the measurement folder of the compressed file.

About this task

The format of the csv file is:

```
isActive,mRID,name,measurementType,resource,phaseCode,unitMultiplier,unitSymbol,terminal
```

You can find the description for the columns here: [Reference Asset ID](#).

Procedure

1. From the menu bar click **Administration > Custom Data Model**
2. From **Custom Data Model** click **Upload Data Source > Upload Data Source**

3. You can either drag and drop the zip file to the **Upload Data Source** window, or browse to the location where you have saved the zip file. The **Custom Data Model** window shows the status of the upload, and a log file is available in the **View Log** column.
4. Click the icon in the View Log column to view the log file.

Configuring the Visual Insights rules

After you have completed the measurement reading table and measurement type, you configure the IBM Visual Insights inspection rules.

About this task

You configure the inspection rules in IBM IoT for Energy and Utilities. This completes the information you require for the measurement type and the relationship between the inspection type and the model that is used in IBM Visual Insights and authentication to Visual Insights.

Procedure

1. Click **Administration > Configure Visual Inspection Rules > Add New**.
2. Select the asset class that you need the inspection rule for.
3. Type the inspection type.
4. Select the **Measurement Type** that you defined.
5. Paste the URL of the Visual Insights score image service link and APIkey.
6. Click **Save**.

Uploading image files for Visual Insights reports

As the field technician you can upload images

About this task

Procedure

1. Click **Asset Performance Management > Visual Inspection Image Uploader > Upload images**.
2. Select the images you need to upload.
3. Select the asset class and type the asset ID.
4. Select the inspection type and click **Upload**.

Showing and Hiding the map street view

In IBM IoT for Energy and Utilities you can view the location of an asset view the street view. You can configure the application if you do not require this function.

About this task

Here you can change the configuration file to show or hide the map street view of IoT for Energy and Utilities.

Procedure

1. Find the Google map configuration file.
 - If you want to change distribution page open the file `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_ah_app.ear/ife_ah_web.war/config/modelDistribution.json`.

- If you want to change transmission page open the file /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_ah_app.ear/ife_ah_web.war/config/modelTransmission.json
2. To show the Google map street view, change the parameters as follows:


```

"showStreetMap":true
"googleStreetMapKey":yourMapsJavaScriptAPIKeyNumber
      
```
 3. To hide the Google map street view, change the parameter as follows:


```

"showStreetMap":false
      
```
 4. For information on how to apply your Maps JavaScript API key, use the link: <https://developers.google.com/maps/documentation/javascript/get-api-key>

Viewing and analyzing energy data

Use IBM IoT for Energy and Utilities to provide data analysis, the calculation of failure and risk as well as providing an estimate of failure and risk.

Assets can be viewed on a geospatial map or as a list. The displayed assets can be filtered using criteria specified by the user.

Detailed reports can be displayed for individual assets, or groups of assets.

The user interface

The application user interface is composed of four parts:

- Filter selector - to filter assets for different criteria, asset class, type, score range, geography, and advanced.
- View selector - to select the content viewer between map, list, report, and matrix view.
- Content area - to visualize differing approaches of assets, including map, list, report and matrix views.
- Legend panel - to show the different asset classes and the score ranges.

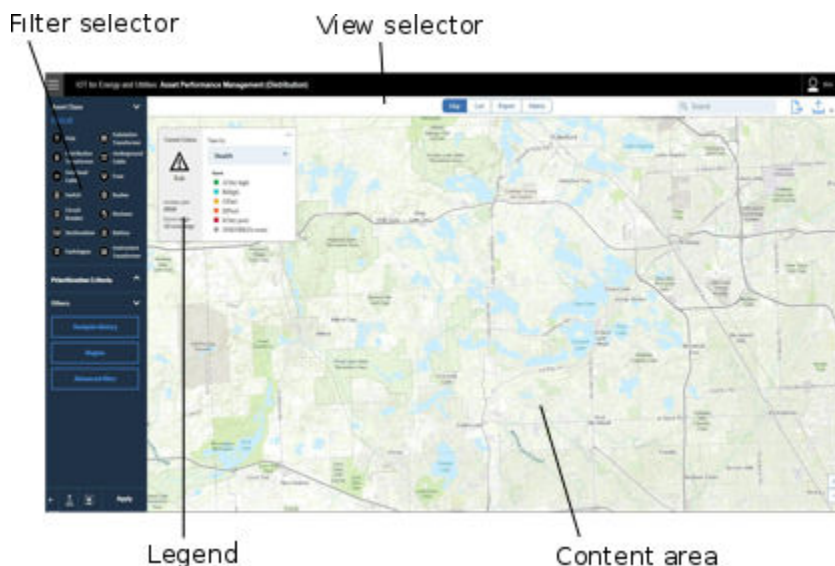


Figure 17. User interface for Asset Health

Logging on to the IBM IoT for Energy and Utilities Asset Performance Management application

Log on to access the IBM IoT for Energy and Utilities user interface.

Before you begin

Contact your local administrator to obtain your user ID and password. Your administrator is responsible for ensuring that you have the security access level that is appropriate to your role in your organization. Your administrator will also supply you with the web address URL for accessing the solution portal.

About this task

Use the following procedure to start a new browser session and access IoT for Energy and Utilities.

Procedure

1. Enter the URL into the address field of the browser.

Note: The fully qualified domain name is required in the URL, for example, `https://<App Node>/ibm`. If you use the IP address instead of the registered fully qualified domain name, some windows do not open correctly. Also, if you do not use the https protocol, the link is redirected to use the https protocol.

2. On the login page, enter your user ID and password.
3. Click **Log In**.
4. Click the menu icon and click **Asset Performance Management**.

Results

Only the pages, features, and data that you have permission to access are displayed. Contact your administrator if you require more access.

Navigating the user interface of IBM IoT for Energy and Utilities

In IBM IoT for Energy and Utilities, you can navigate to a specific page using the navigation bar.

The navigation bar contains four parts:

- Navigation segment
- Search box
- Transfer icon
- Download icon

Navigation segment

The navigation segment is part of the navigation bar. IoT for Energy and Utilities has four choices:

- **Map** - map view
- **List** - list view
- **Report** - report view
- **Matrix** - matrix view

Select the navigation segment for the view you need.

Search box

The search box is available in the list, report, and matrix views and is in the navigation bar. Here you can search for assets classes. The search box completes the search automatically.

Transfer icon

The transfer icon is part of the navigation bar. When you select an asset you can transfer the data for that asset to Asset Investment.

Down-load icon

The download icon is part of the navigation bar. When you click the icon, the report is downloaded.

Preview cards

When a user clicks on an asset or region, a preview card is displayed with additional information on that asset or region.

If a region containing multiple assets is selected, the preview card will display the average scores for the region, the total number of assets by class within the region, and available actions.

If a single asset is selected, the preview card will display the scores for that asset as well as available actions.

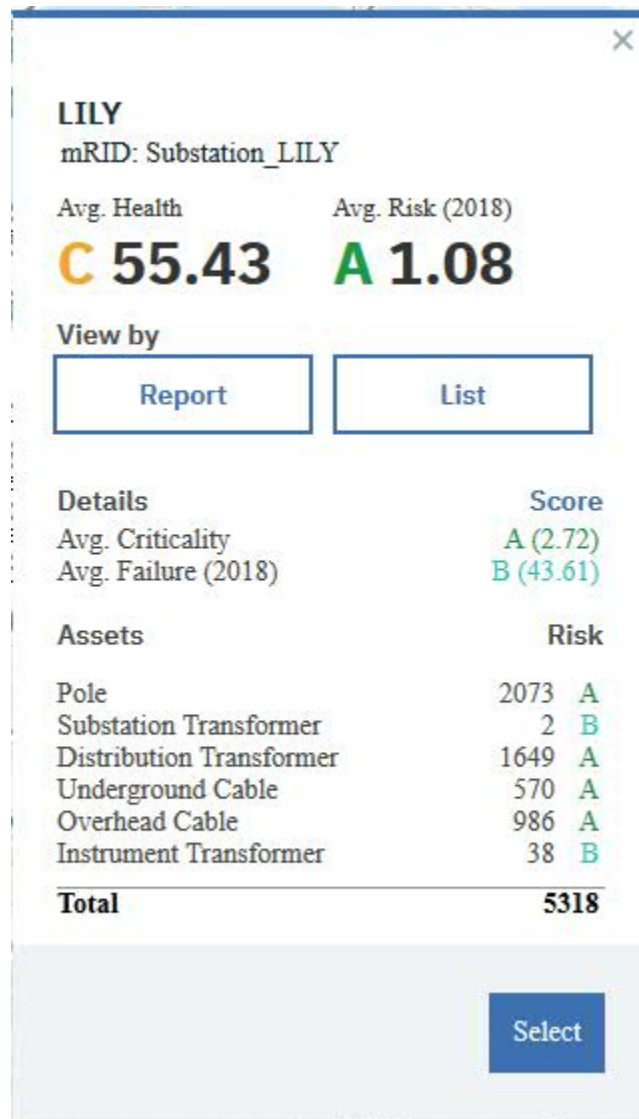


Figure 18. Preview card for a substation

Filter selector

The filters selector contains the controls to filter the visualization of assets on the main view.

You can select the filter control options that are available. The predefined filters are:

- **Asset class** - Shows the asset class that you want to view.



Figure 19. Asset class filter

- **Prioritization Criteria** - the filter criteria are:

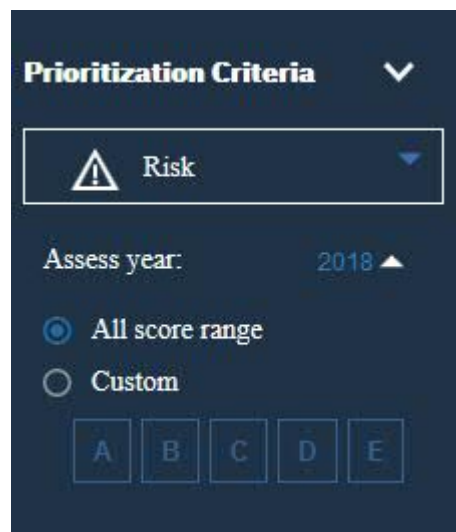


Figure 20. Prioritization criteria filter

- **Score type** - The four score types are: Health, Risk, Failure and Criticality, where **Health** is the asset health index, **Failure** is probability of failure, **Criticality** is the seriousness of that failure and **Risk** is the measure of the risk to the business if a failure in the network occurs. When you select **Risk** you can also select the year.
- **Asses year** - You can choose the present year, or a year in the future assess the risk of network or asset failure for that year.
- **All score range** - The status filters for all score types.

- **Custom** - The status filters for the score type you select. There are six states, A - Very low, B - Low, C - Poor, D - High, E Very high, and No score.
- **Others** - You can refine your selection to include:

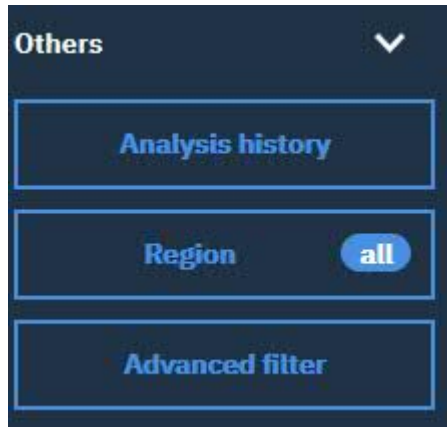


Figure 21. The other filter criteria

- **Analysis history** - Every time you run an SPSS analysis the generated results are on stored on the database. Analysis history lets you select the results from a previous analysis or select the system default which is the most recent analysis.
- **Region** - Filters the area map that you require.
- **Advanced** - You can add your own filters. The **Property** options are:
 - serialNumber
 - isActive
 - installationDate
 - removalDate
 - manufacturer
 - model
 - operatingVoltage
 - interruptingMedium
 - facilityId

The Relationship gives you the ability to set a value to a property. depending on the property selected you can set a relationship type:

- For string input:
 - is
 - is not
 - starts with
 - contains
- For a boolean input:
 - is
- For a date input:
 - after
 - no sooner than
 - before
 - no later than
- For a number input:

- =
- >
- <
- >=
- <=

The value relates directly to the property and relationship selected.

- For a string, type the string value.
- For a boolean value:
 - True
 - False
- For a date value, select the date from the calendar.
- For a number, type the number as a value.
- Time line- The period of time for the assets that you want to view.

Filtering assets

The assets displayed on the map or list can be filtered based on selected criteria.

About this task

IoT for Energy and Utilities on Cloud has the following predefined filter options:

- Asset Class
- Score Range
- Type, where **Health** is the asset health, **Failure** is probability of failure, **Consequence** is the consequence of failure.
- Region
- Advanced

Additional filter criteria can be specified using the **Advanced** option.

To reduce the number of displayed assets to those meeting the desired criteria, do the following.

Procedure

1. Select the asset-classes that you need to assess. The icon for the selected asset class becomes light blue.
2. From the **Prioritization Criteria** select the score type.
3. Click **All score range** or select **Custom** to select the A,B,C,D, or E in the score range field.
4. Click the **Analysis history** button to select a previously saved analysis or to use the system default.
5. Click the **Region** button to open the region dialog window. Make the required selection of utility, substation and feeders.
6. Click the **Advanced filter** button to open the advanced dialog window.
You can make a selection to your own criteria. When complete click **OK**.
7. Click **Apply**.

Results

The map or list view will display the assets meeting the selected filter criteria.

Creating a filter preset

You can create filter presets to be able to analyze similar filter selections.

About this task

You can use the filter criteria to create and save a preset. The preset save icon:



Procedure

1. Click the menu icon and click **Asset Performance Management > Asset Performance Management (Transmission)** or **Asset Performance Management > Asset Performance Management (Distribution)**.
2. Select the asset-classes that you need to create the preset for. The icon for the selected asset class becomes light blue.
3. From the **Prioritization Criteria**, select the score type.
4. Click **All score range** or select **Custom** to select the A,B,C,D, or E in the score range field.
5. Click the **Analysis history** button to select a previously saved analysis or to use the system default.
6. Click the **Region** button to open the region dialog window. Make the required selection of utility, substation, and feeders.
7. Click the **Advanced filter** button to open the advanced dialog window.
You can make a selection to your own criteria. When complete click **OK**.
8. Click the preset save icon and type the name of the preset.
9. Click **Save**.

Results

You can use this saved preset to load to the application.

Loading a filter preset

A preset can be used to load filter criteria Asset Performance Management application.

About this task

After you have created a preset you can load the preset to the Asset Performance Management application. The preset load icon:



Procedure

1. Click the menu icon and click **Asset Performance Management > Asset Performance Management (Transmission)** or **Asset Performance Management > Asset Performance Management (Distribution)**.
2. Click the load preset icon and select the preset you want to use.
3. Click **Load**.
You can now use this preset or edit the filter criteria to create a new analysis.
4. Click **Apply**.

Viewing the health status of assets in the map view

You can view the health status of assets classes in the map view. The map provides a visualized distribution of risk and failure.

About this task

The map contains the following parts:

- Map area
- Asset class legend
- Score Type legend

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as a user.
2. In the Navigation segment click **Map**.
The map is displayed.
3. Use the **Filter** selector to make a selection of an **Asset Class, Prioritization Criteria, Others**, and click **Apply**.
4. Make a change to the Score Type legend



and view the change on the map.

Note: If you choose the score type Risk or Failure then a time line shows. The Asset Health application estimates Risk and Failure for the future. If you choose a different year in the time line, the map shows the estimated values for that year in the map view.

5. Click on an area to view the preview card for the substation for that area.

COMLK
mRID: Substation_COMLK

Avg. Health **C 56** Avg. Risk (2018) **A 0.54**

View by

[Report](#) [List](#)

Details	Score
Avg. Criticality	A (2.14)
Avg. Failure (2018)	B (25.59)

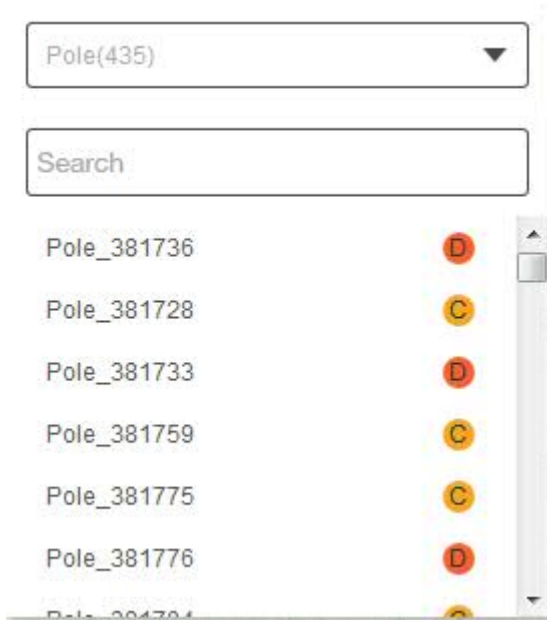
Assets	Risk
Pole	4294 A
Substation Transformer	2 B
Distribution Transformer	1989 A
Underground Cable	633 A
Overhead Cable	1515 A
Circuit Breaker	25 A
Battery	23 B
Switchgear	30 B
Instrument Transformer	32 B
Total	8543

[Select](#)

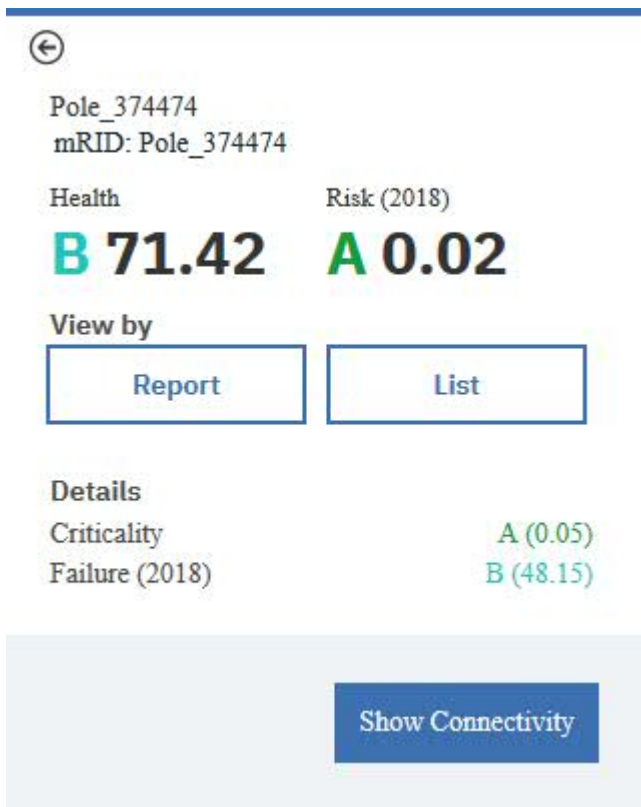
6. You have choices:
 - a) To see the assets in the report view, click **Report**.
 - b) To see the assets in the list view, click **List**.
 - c) To see the assets in the map view, click **Select**.
7. Zoom in the map to see the details of an asset class.
8. Click the circle to open the details of the substation.
9. Click a grey circle that denotes a cluster which contains a number.
A card opens that shows the details of the cluster.

Note: The grey circle with the number indicates a cluster of assets. The number indicates the quantity of assets in that cluster.

Note: You can search for a particular asset using the search box.



10. Click an asset from the list to open the details of that asset.



Results

In the map view you have a visualization of risk, failure, criticality and health for the different assets. By selecting different items in the legend and year in the time line, you can see the information for the different conditions.

Viewing the physical location of a single asset in street view

You can view the physical environment of a single asset that has a point location in IBM IoT for Energy and Utilities.

About this task

With the **Street View** you can see the environment and conditions for the location of an asset. For example you can see the proximity of trees and branches to a pole with the overhead conductors. An assessment team can also view the location before an inspection.

The **Street View** of an asset is available in the **Map** view.

You can also get to a single asset from the List view, and selecting **View on map**.

Procedure

1. In the **Map** view, click the icon for the single asset.
2. Click **Street View**.
3. The **Street View** opens at the closest location to the asset from the street or road.
With your mouse you can rotate the field of vision through 360 degrees.

Viewing the health status of assets classes in the list view

Assets and their network health and risk values can be displayed as a list.

About this task

The List view contains the following parts:

- Main table
- The individual asset class tabs

A list view has these columns:

- Asset Name
- Feeder
- Container
- Health, Risk, Criticality, Failure, Effective age, and Age.

All Pole Overhead Cable Fuse Pole_374224 ***									
Asset Name	Feeder	Container	Health	Risk	Criticality	Failure	Effective Age	Age	
OHC_3128929	COMLK9538	COMLK9538	B 84	A 0	A 0.05	A 4.67	14	10.08	
OHC_3128930	COMLK9538	COMLK9538	B 71	A 0	A 0.05	A 4.67	14	10.08	

Figure 22. List view

When you view a single asset list, the year column also shows.

In this task you will select a different asset class tab and select a different year and review the results in the table.

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as a user.
2. Use the **Filter** selector to make a selection of the **Asset Class, Prioritization Criteria, Others**.
3. In the navigation segment bar click **List**.
The list is displayed.
4. Click an asset class tab and view the result in the table.
The filter selection you have made determines the content of each asset class tab, when you change the filter selector you change the items on display in the list view.

5. Select a different year in the time column to see how changes to the year changes the health of the asset.
6. Click an asset in the main table to open a hover menu with the items, **Open in new Tab**, **View by Map**, **View by Report**.

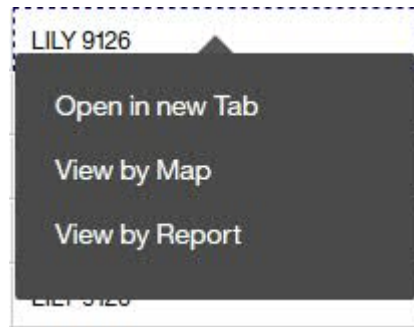


Figure 23. Hover menu items

7. Click **Open in new Tab** to view a single asset list with its history.
8. Click on the **View by Map** or **View by Report** to open the map or report views.

Results

In the list view, you can see the details of every asset. You can open a specific asset in new tab, in the map view or in the report view to see more details.

Viewing the health status of multiple asset classes in the report view

The report view provides many visual charts. You can get a visualization of the health status of multiple asset classes.

About this task

In this task you select different tabs and in the report view see the results as visualizations. Both single and multiple asset class reports are available. The charts that are available for multiple asset classes are:

- Multiple asset class report showing the basic information for more than one asset class.

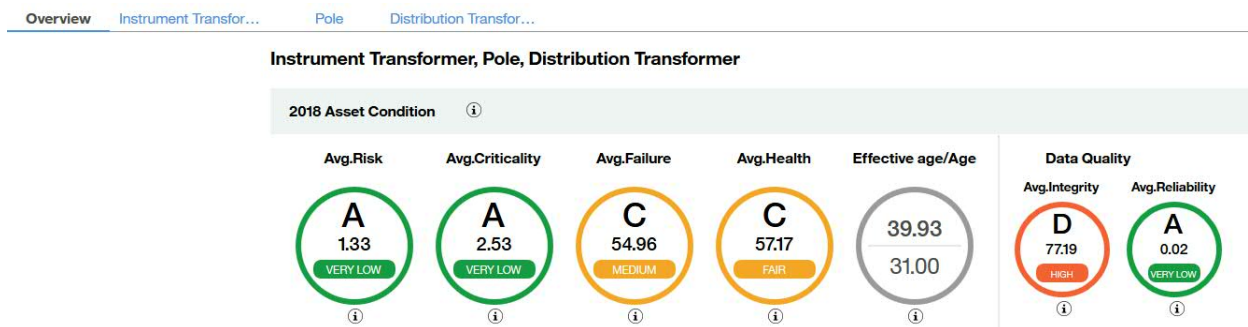


Figure 24. Multiple asset classes basic information

- The Average failure and risk over time shows the changes to average failure and risk over time.



Figure 25. The changes to average failure and risk over time report

- Additional information report

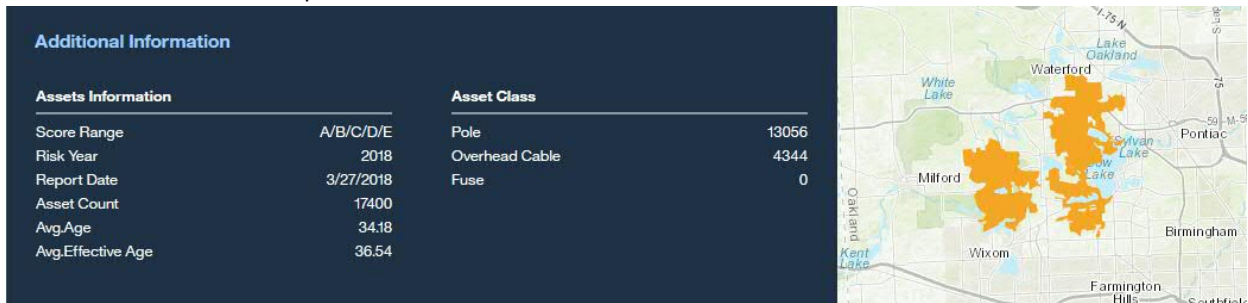


Figure 26. Additional information report

- Asset Risk Distribution



Figure 27. Asset risk distribution report

- Top 10 Highest Ranking - by region



Figure 28. Top 10 ranking by region

- Top 10 Highest Ranking by feeder



Figure 29. Top 10 ranking by feeder

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.
When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **Report**.
The report view opens.
4. See the reports available in the **Overview** report view.

Viewing the health status of a single asset class in the report view

The report view provides many visual charts. You can get a visualization of the health status of a selection of asset classes, a single asset class, and a single asset.

About this task

In this task you select different tabs and in the report view see the results as visualizations. Both single and multiple asset class reports are available. The charts that are available for a single asset class are:

- The overview report showing the basic information for a single asset class.

Pole

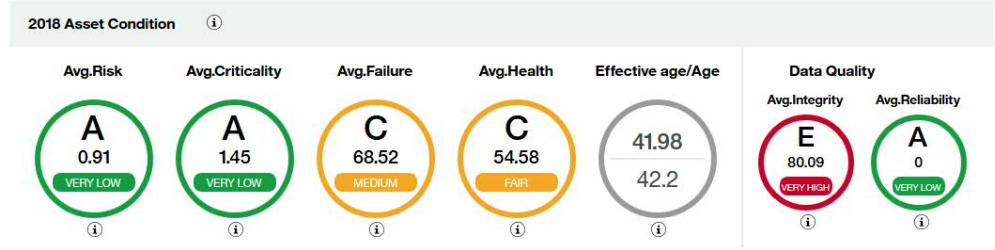


Figure 30. Overview showing a single asset class

- The basic information report shows the changes to average failure and risk over time.



Figure 31. The changes to average failure and risk

- Additional and location information.



Figure 32. The additional information and map locator

- The distribution of average risk for all assets over time.



Figure 33. The score for average risk for an asset over time

- The level of average risk for a region changes over time.



Figure 34. Changes to ranking of the highest risk to multiple regions over time

- The level of average risk for a feeder changes over time.

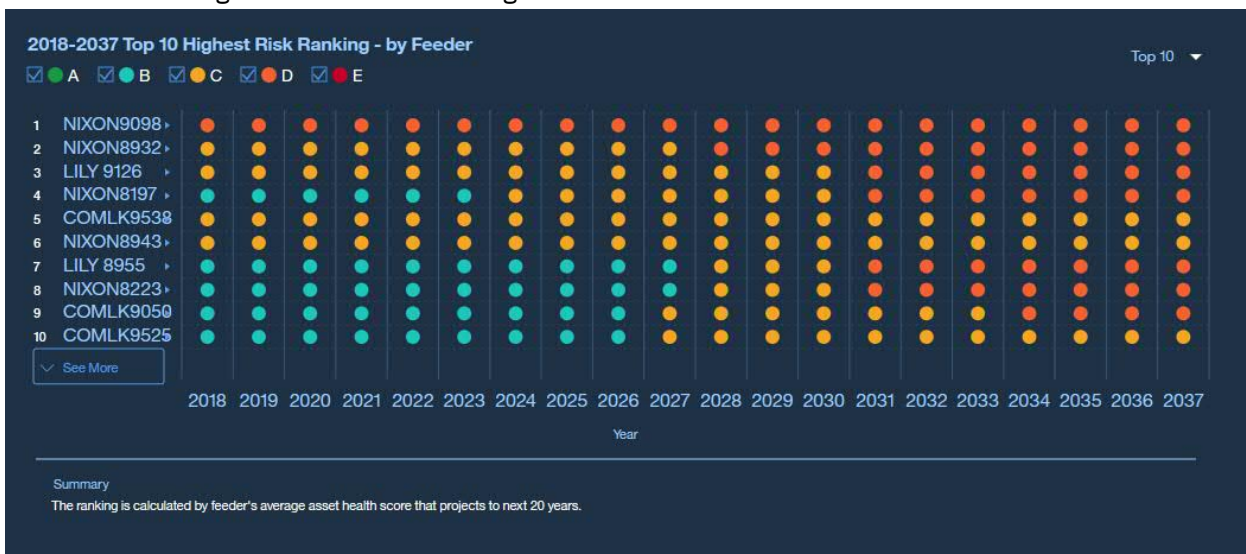


Figure 35. Changes to the average risk for a feeder over time

- The degradation curve for an asset over time.



Figure 36. Degradation curve for an asset over time

- Asset Health Index by age distribution.



Figure 37. AHI age distribution

- Asset Health Index by effective age distribution.

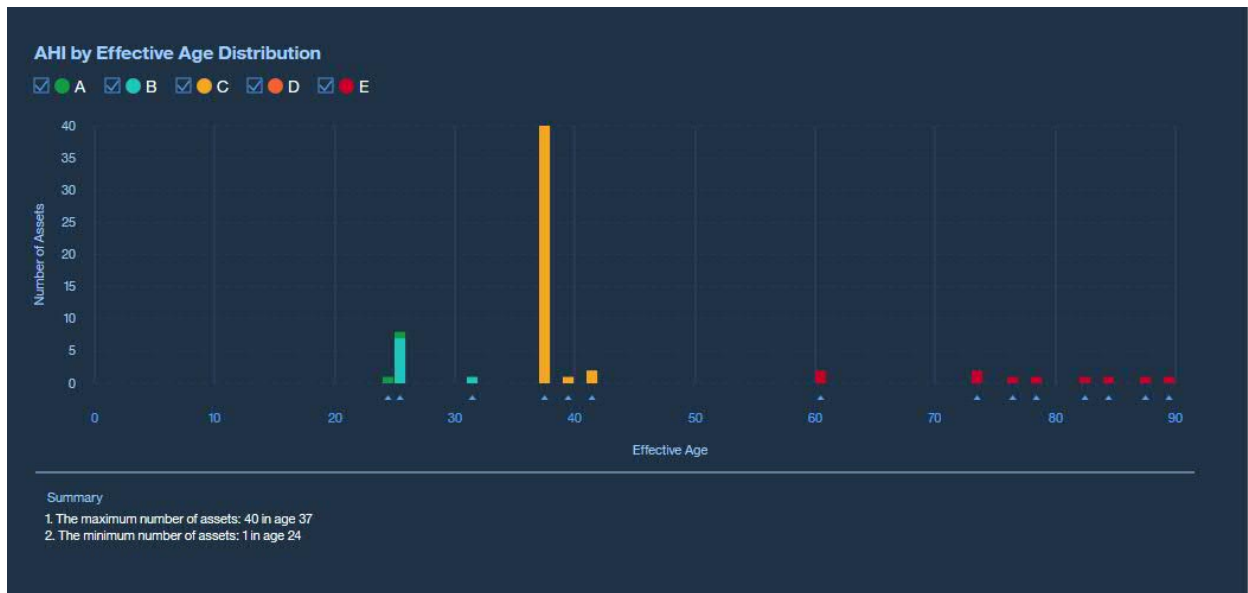


Figure 38. AHI effective age distribution

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.
When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **Report**.
The report view opens.
4. Click an asset class tab and view the result in the report.
The filter select you have made determines the content of each asset class tab, when you change the filter selector you change the items on display in the report view.
5. See the reports available in the report view.

Results

In the report page, you can see the different charts available for the health status of an asset class.

Viewing the health status of a single asset in the report view

The report view provides many visual charts. You can get a visualization of the health status of a single asset.

About this task

In this task you select different tabs and in the report view see the results as visualizations. Both single and multiple asset class reports are available. The charts that are available for a single asset class are:

- Single asset class report showing the basic information for one asset.

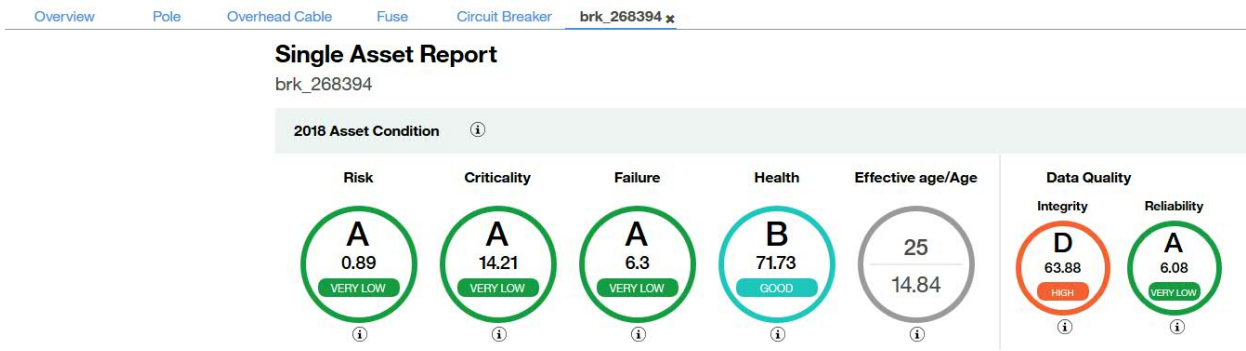


Figure 39. Single asset report overview information

- Basic Information



Figure 40. Basic Information report

- Multimedia

You can add images and video clips to the single asset report.



Figure 41. Multimedia images

- Health Breakdown



Figure 42. Health breakdown report

- Criticality Breakdown



Figure 43. Criticality breakdown report

- Degradation Curve

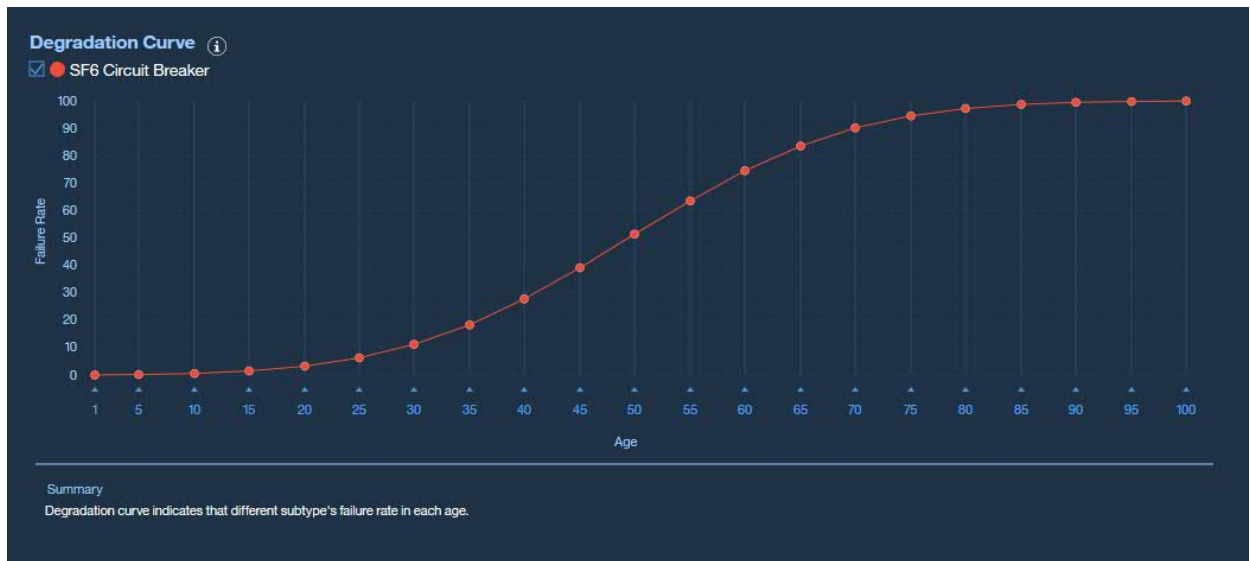


Figure 44. Degredation curve report

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.
When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **List**.
The list view opens.
4. Select the single asset from the **List** view and select **View by Report**.

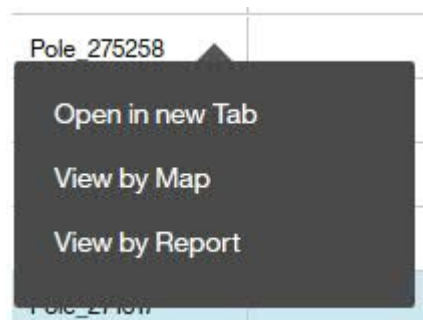


Figure 45. Drop down menu in the list view

5. See the reports available in the report view for a single asset.

Viewing the health status of assets classes in the matrix view

Asset classes and their health status can be displayed as a matrix.

About this task

In this task, you select different tabs in the matrix and select different score type to see the detail information of an asset class.

Risk		A	B			C	D		E		Risk Summary				
Criticality	100	1	0	0	0	0	0	0	0	0	E	0	Very High Risk		
	90	4	1	0	0	0	0	0	0	0					
	80	1	1	0	0	0	0	0	0	0					
	70	1	0	0	0	0	0	0	0	0	D	0	High Risk		
	60	1	1	0	0	0	0	0	0	0					
	50	0	5	0	0	0	0	0	0	0	C	0	Medium Risk		
	40	4	5	0	0	0	0	0	0	0					
	30	3	8	0	0	0	0	0	1	0	B	5	Low Risk		
	20	4	5	0	0	0	0	0	1	0					
	10	9	5	0	0	0	0	0	2	0	2	A	60	Very Low Risk	
			10	20	30	40	50	60	70	80	90	100	65		
			Probability												

Figure 46. The health status of assets shown as a matrix

Procedure

1. Sign on IoT for Energy and Utilities on Cloud as a user.
2. Use the **Filter** selector to make a selection of an **Asset Class**, **Score Range**, **Type**, and **Region** and click **Apply**.
3. In the navigation segment bar click **Matrix**.
The matrix view opens.
4. Click an asset class tab and view the result in the matrix.
The filter selection you have made determines the content of each asset class tab, when you change the filter selector you change the items on display in the matrix view.
5. Click different risk levels to highlight the results that correspond to that level of risk.
The number in the results matrix indicates the number of qualified asset in that asset class.
6. View the preview panel to view the information details for the qualified asset class. You can also view the information by **Feeder** and **Region**.
7. Select a different year in the time line and observe the change to the information in the matrix and preview panel.

Exporting data for a single asset class

In IBM IoT for Energy and Utilities you make an export of data in csv format for an asset class without having first to create a report.

Before you begin

You can export data directly from Asset Performance Management Transmission or Distribution, you do not need to create a filter before you make the export.

About this task

You can make a single selection for an asset class, and define the type of data you need to export directly from the database. The types of data you can export are:

- Asset master data.
- Asset health data.
- Asset measurement data.

Procedure

1. Click Asset Performance Management and select either **Transmission** or **Distribution**.
2. Click the **Export all data** icon.



Figure 47. Exporting all data

3. Click **Export all data**.
4. Select the asset class that you want to export and the data type. Click **Next**.
5. Select the columns to export. The default selects all columns.

Results

You can export the csv file to your system and open it in the software of your choice.

Viewing analytics dashboards

When viewing a report, additional analytic data is available from IBM Predictive Maintenance and Organization.

Procedure

- When viewing a report, click **Advanced Analytics**.

Results

When viewing a single asset report, the Equipment Dashboard for that single asset is displayed. When viewing a multiple asset report, the site overview dashboard is displayed.

Asset Investment application

The Asset Investment application in IBM IoT for Energy and Utilities helps asset managers to determine the best investment plan possible according to the utility objectives and constraints.

With IoT for Energy and Utilities, you can create an investment project for a particular asset class. The replacement costs for that asset are included when the investment project is set up. Various scenarios or models can be set to include the level of risk and failure, budget constraints, and planning duration.

Based on the asset health indexes, failure probability, criticality, risk and planning interval, you can review the future years for a single asset plan by map, list, or report view mode. You can then export the reports to whom it concerns. The default number of years to review is 20. The number of years can be changed by the user.

Creating an investment project

To create the various scenarios, you first must create the investment project in IBM IoT for Energy and Utilities.

About this task

The investment project includes the initial asset class, the subtype, and the replacement cost of those assets.

The results show an overview of the years of the report, the risk, the costs, and the number of asset replacements on an average, maximum, and minimum basis.

The yearly results can also be shown.

Procedure

1. Sign on IoT for Energy and Utilities as an administrator.
2. Click **Asset Investment**.
 - If this entry is the first project, click **Create Project**.
 - If there are multiple projects, you can create a project by clicking **Duplicate** on an existing project.
3. Click **Open**.
4. Type:
 - a) The project name.
 - b) Select the asset class to include.
 - c) Select the asset subtype.
 - d) Type the replacement cost.
5. Click **OK**.

IoT for Energy and Utilities calculates the result and shows the results as a map view.

Viewing the results of the investment project in the map view

After the investment calculation finished in IBM IoT for Energy and Utilities, you can view the results in a map view showing the assets as color circles on the map.

Before you begin

An investment project must be created and available for opening.

About this task

You can anticipate the replacement needs by cost, year, and location on the map view.

Procedure

1. Hover over the investment project you need and click **Open**.
2. The default map shows an overview of the results.

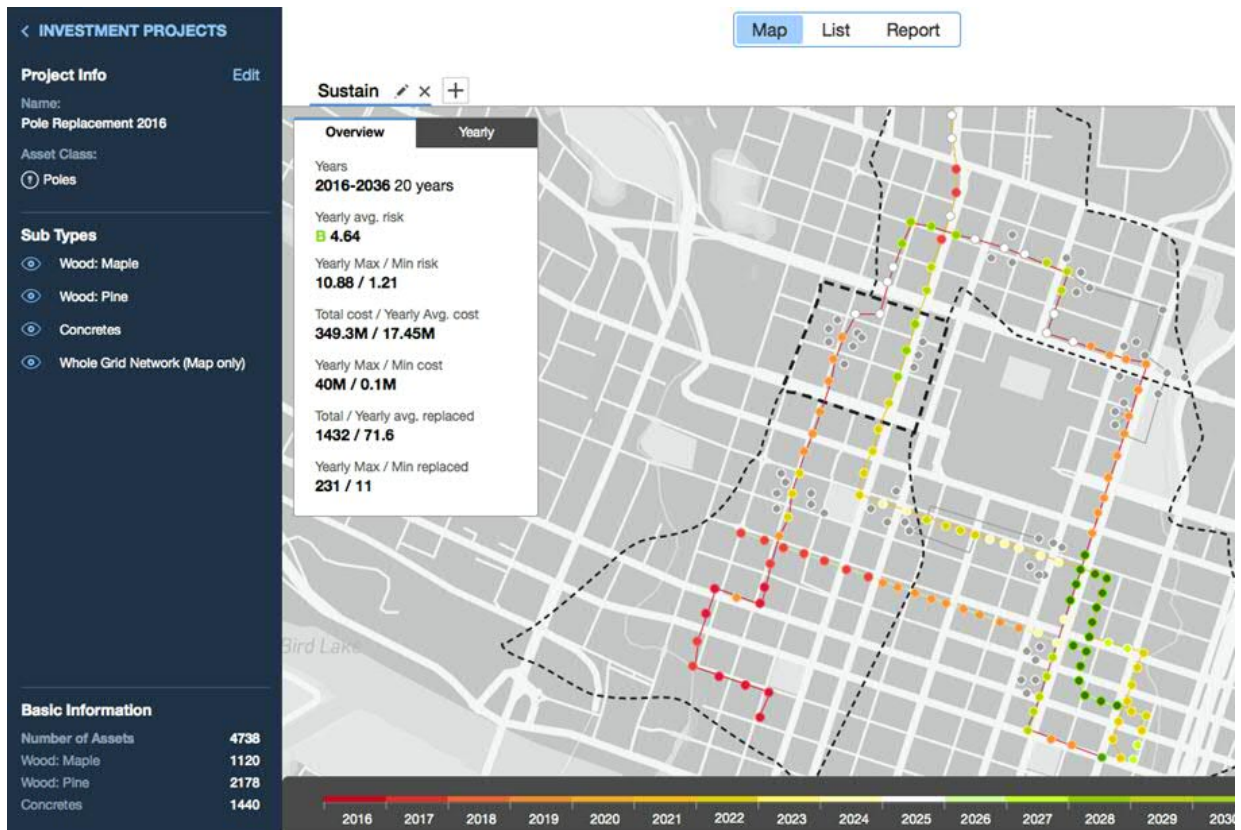


Figure 48. The map view of an investment project

- To view the details of an individual asset, select a substation or feeder then zoom in until you see the specific asset and click.

A chart opens that shows the replacement year, the risk before and after replacement, and the replacement cost.



Figure 49. Asset details showing replacement year and risk

4. To see the yearly results, click the **Yearly** tab and move the slider on the timeline to view the year.
5. To select a range of years, in the **Yearly** tab, slide the **Select range** button and use the two slides in the timeline to select the range. The default is 20 years.

Viewing the results of the investment project in the list view

After the investment calculation finished in IBM IoT for Energy and Utilities, you can view the results in a list view showing the assets as rows in the list.

Before you begin

An investment project must be created and available for opening.

About this task

You can anticipate the replacement needs by cost, year, risk, and failure in the list view.

Procedure

1. Hover over the investment project you need and click **Open**.
2. In the Navigation segment click **List**. The list is displayed.
3. Click the row containing the asset to view the details of that asset.

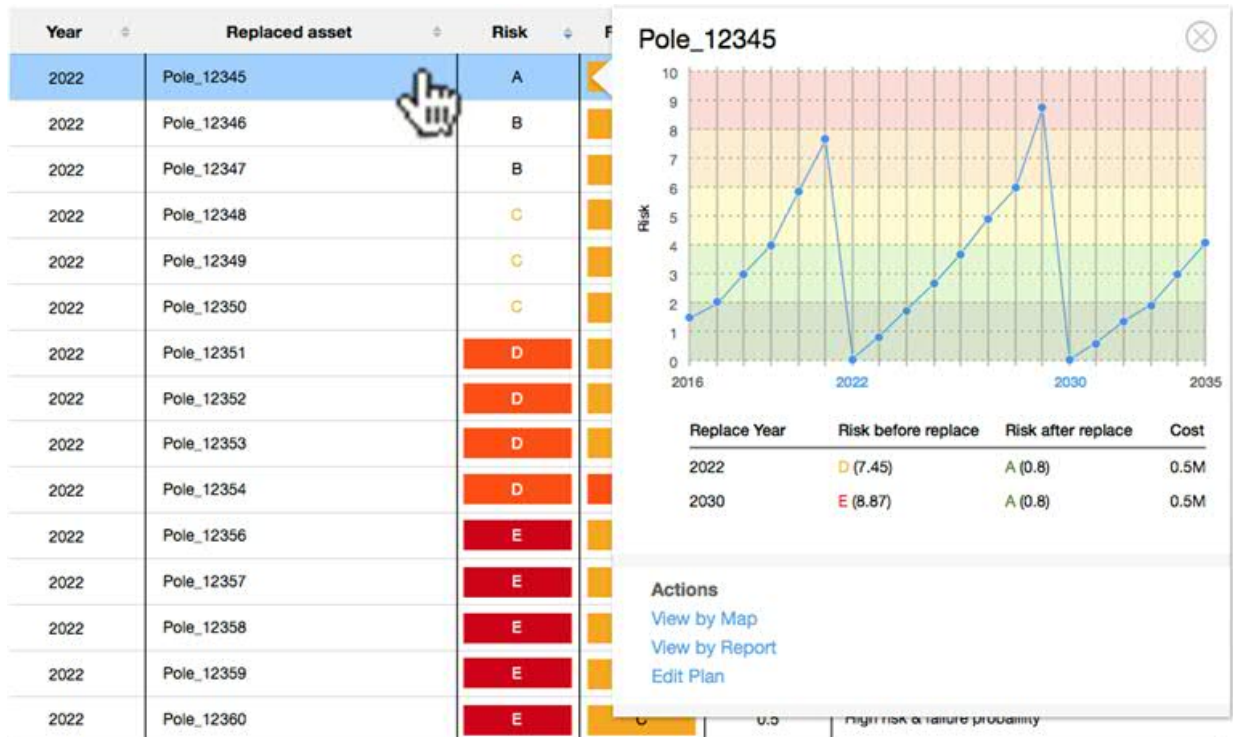


Figure 50. List view in Investment Planning

Creating a scenario

Using the Asset Investment Planning application in IBM IoT for Energy and Utilities, you can create models for what if scenarios.

Before you begin

You must have an investment plan set-up in IoT for Energy and Utilities.

About this task

You can set up the plan duration and the failure threshold against either a budget or an acceptable level of risk.

Procedure

1. Click **Asset Investment**.
2. Click the investment project to work with.
3. Click the **Add** icon next to the **Sustain** tab.
4. Type the name of the scenario.
5. Select the orientation of the scenario.
 - **Budget orientated**
 - **Risk orientated**
6. Set the start year of the plan.
The default is first year of asset health result.
7. Set the number for years of the plan.
The default setting is 20 years.
8. Set the failure probability threshold in which assets must be replaced, for example 99%.
9. Click **OK**.

Results

You can now create a report that compares the difference between the project to sustain the assets and the scenario.

Comparing the results of a scenario in the report view

In IBM IoT for Energy and Utilities you can compare the results of a scenario against the existing scenarios and against the sustain project.

Procedure

1. Open the investment project with the scenario you want to compare.
2. Click **Report**.
3. You can compare the scenario against the sustain for risk and budget.
4. You can compare **Risk, Failure probability, Replacement** and **Cost** for the different scenarios, click the each row in the visualization.

The results are directly displayed in the comparison visualization.

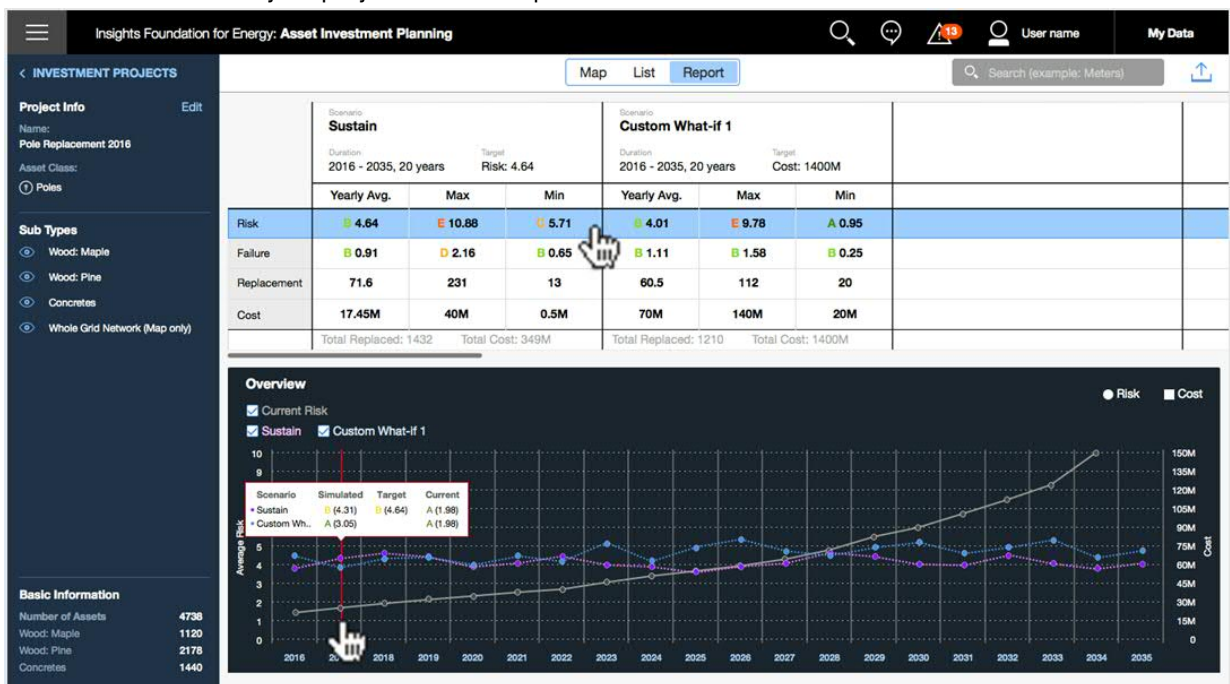


Figure 51. The comparison of results for Risk

Chapter 5. Using Connectivity Model

The Connectivity Model gives feedback to managers, data analysts and grid operators as to the accuracy of the phase and connectivity details of the network.

The Connectivity Model application ensures:

- The model is always current: The user is sure that the connectivity model represents the latest analytic results.
- The connectivity and reliability of information is quickly ascertained: The user can fully understand the state of assets in the network within 5 minutes or less.
- The network details are simplified: The user can focus on the assets and connections that are important to them.
- The analysis details can be reported and shared: The users can collect the necessary charts or maps, and download and share them with the key stakeholders.
- The reports for customer to phase and customer to transformer mapping are accurate.
- The ability to fix connectivity records without the time and expense of sending crews into the field.

Overview of the data flow

The data flow contains four parts: the data preparation for the Connectivity Model application, the Extract, Transform, and Load (ETL) process, the data validation, and the analysis of the data.

The details of the ETL module are described in this section.

You prepare you own data in .csv format and then load the .csv files into the HDFS.

After preparing the raw data, you run the ETL and validation module to generate the data for the analysis.

Preparing the data for the ETL module

Explains how you prepare the raw data for the ETL module on the Jupyter node as a .csv file.

The raw data is created as a .csv files, the raw data folder structure is shown as the following:

- connectivity
- electric_station
- exclude_time
- feeder_root
- meter
- meter_load
- meter_voltage
- overhead_cable
- scada
- scada_load
- substation_region
- transformer
- underground_cable

The prepared data includes master data and reading data. The master data contains the assets in the power grid, and the reading data contains the measurement readings, for example: voltage and load.

As the ETL module supports data in .csv format, by using the csv format, you can edit and update the master data and reading data. You can also incrementally add reading data to the ana store. You can define multiple csv files in each folder. The ETL module does not support sub-folders.

The format of the .csv files is given as follows:

Master data

The master data contains the data of the assets.

Column name	Type	Description	Constraints
assetId	String	asset ID	Primary Key, Unique, Not NULL.
substation	String	ID of substation region to where electric station belongs.	Foreign Key, must be a valid substation in the substation region table.
type	String	The asset type.	
phase	String	The phase of the asset.	
numberOfPhases	String	The amount of phases.	
normalStatus	String	The normal status.	
node1	String	The node 1.	
node2	String	The node 2.	

Column name	Type	Description	Constraints
id	String	ID of the electric station	Primary Key, Unique, Not NULL.
substation	String	ID of substation region to where electric station belongs.	Foreign Key, must be a valid substation in the substation region table.
geometry	String	The geometry of the electric station.	Must be in a valid WKT POLYGON, in WSG84 projection.

Column name	Type	Description	Constraints
feeder	String	ID of the feeder.	Primary Key. Must be a valid feeder in the feeder table or feeder_group in the feeder_group table
type	String	Type of analysis.	Primary Key. The candidate value include <i>Load, Voltage, Voltage_with_scada.</i>

Table 6. Exclude time master data (continued)

Column name	Type	Description	Constraints
startTime	String	Timestamp, in the format of yyyy-MM-ddThh:mm:ss.sss.	Primary Key. The time range to be excluded from the analysis with the startTime being the start time of the range (inclusive).
endTime	String	Timestamp, in the format of yyyy-MM-ddThh:mm:ss.sss.	Primary Key. The time range to be excluded from the analysis with the endTime being the end time of the range (exclusive).

Table 7. Feeder root master data

Column name	Type	Description	Constraints
feeder	String	ID of the feeder where the meter is connected	Foreign Key, must be a valid feeder in the feeder table.
rootAsset	String	The root asset.	

Table 8. Meter master data

Column name	Type	Description	Constraints
id	String	ID of the meter	Primary Key, Unique, Not NULL.
substation	String	ID of substation region where the meter belongs to.	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the meter is connected	Foreign Key, must be a valid feeder in the feeder table.
transformer	String	ID of the transformer	Foreign Key, must be valid transformer in transformer table.
phase	String	The phase of meter	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be a valid WKT POINT, in WSG84 projection.

Table 9. Overhead cable master data

Column name	Type	Description	Constraints
id	String	ID of the overhead cable	Primary Key, Unique, Not NULL.
substation	String	ID of the substation region where the overhead cable belongs to.	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the overhead cable is connected.	Foreign Key, must be a valid feeder in the feeder table.
phase	String	The phase of the overhead cable.	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	The geometry of the overhead cable.	Must be a valid WKT MULTILINESTRING, in WSG84 projection.

Table 10. SCADA master data

Column name	Type	Description	Constraints
assetId	String	ID of the feeder.	Primary Key, Unique, Not NULL.
measurement	String	Type of analysis	Primary Key Candidate value must include <i>Load, Voltage, Voltage_with_scada</i>
scadaId	String	ID of SCADA.	

Table 11. Substation region master data

Column name	Type	Description	Constraints
id	String	ID of the substation region.	Primary Key, Unique, Not Null.
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection

Table 12. Transformer master data

Column name	Type	Description	Constraints
id	String	ID of the transformer.	Primary Key, Unique, Not NULL.

Table 12. Transformer master data (continued)

Column name	Type	Description	Constraints
substation	String	ID of the substation region where the transformer belongs to.	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the transformer is connected.	Foreign Key, must be a valid feeder in the feeder table.
phase	String	The phase of the transformer	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
kva	Int	The kilovolt-amps.	
voltage	Int	The voltage.	
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be a valid WKT POINT, in WSG84 projection.

Table 13. Underground cable master data

Column name	Type	Description	Constraints
id	String	ID of the underground cable.	Primary Key, Unique, Not NULL.
substation	String	ID of the substation region where the underground cable belongs to,	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the underground cable is connected.	Foreign Key, must be a valid feeder in the feeder table.
phase	String	The phase number.	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	The geometry of the underground cable.	Must be a valid WKT MULTILINESTRING, in WSG84 projection.

Reading data

The reading data contains the current record values, for example, the voltage values and the load values. The reading data is used for the analysis.

Note: These notes are for all reading data tables:

1. The timestamp must be in same time zone as the corresponding load table. For example both load tables and reading tables must either use UTC or use the local time zone.
2. The time interval between two timestamps must be fixed and be the same as time interval in corresponding load table. For example, if the time interval is 1 hour, then all readings in the meter load table and feeder load table should use 1 hour interval.
3. The timestamp should align with timestamp in the corresponding load table. For example, if the feeder load has a timestamp 9:00 then the meter load timestamp should be the same.
4. The timestamp represents the end edge. For example, if the timestamp is 10:00 and interval used is 1 hour, the kWh is the load between 9:00-10:00.

Column name	Type	Description	Constraint
scadaId	String	ID of the feeder	Derived
timestamp	String	Timestamp in the format yyyy-MM-ddThh:mm:ss.sss .	Expected in UTC
kwh1	Double	Load of phase A.	The value represents the aggregated active load in the past time interval. All values must be in units of kWh not MWh. The gap between feeder load and aggregated meter load should less than 10% otherwise the accuracy is affected. The kwh1 is load of phase A that maps to phase code 0b100=4.
kwh2	Double	Load of phase B.	
kwh3	Double	Load of phase C.	

Column name	Type	Description	Constraint
meterId	String	ID of the meter	Primary Key, Foreign Key Must be a valid meter in the meter table
timestamp	String	Timestamp of reading data, in the format of yyyy-MM-ddThh:mm:ss.sss	

Table 15. Meter load reading data (continued)

Column name	Type	Description	Constraint
kwh	Double	Load	Value represent aggregated active load of all 3 phases in past interval. All values in unit to KWH.

Table 16. Feeder voltage reading data

Column name	Type	Description	Constraint
scadaId	String	ID of the SCADA tag.	Derived
timestamp	String	Timestamp in the format yyyy-MM-ddThh:mm:ss.sss.	Expected in UTC
volt1	Double	Phase A voltage	The value represents the average voltage in the past time interval. The value must be normalized to the same voltage level. For example, if some meters are 120v, and others are 240v, when the feeder voltage is 1kV, the values is normalized to the minimal voltage level, that is: Value * 120 / 240. When ch1volt, ch2volt, ch3volt represent the voltage of 3 phases, ch1volt maps to phase A, ch2volt maps to phase B, and ch3volt maps to phase C.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	

Table 17. Meter voltage reading data

Column name	Type	Description	Constraint
meterId	String	ID of the meter	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table, or feeder group in the feeder_group table.
timestamp	String	Timestamp yyyy-MM-ddThh:mm:ss.sss.	
volt1	Double	Phase A voltage	The value represents the average voltage in the past interval. The value must be normalized to the same voltage level. For example, if some meters are 120v, and others are 240v, when the feeder voltage is 1kV, the values is normalized to the minimal voltage level, that is: Value * 120 / 240. Channels ch1, ch2 and ch3 need not be the same value. For a single phase meter, only one channel should have a value. For two phases meters, only two channels should have values. For 3 phases meters, all channels must have values.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	

The ETL process

The ETL is the major process of the ETL module and prepares the required data for the next procedure of the analysis.

The ETL module has three stores:

- The raw store stores the raw data from the user.
- The tmp store stores the results of ETL.
- The ana store stores the data required by analysis. After the optional validation, the data under tmp store should be moved to the ana store so that the analysis can be executed.

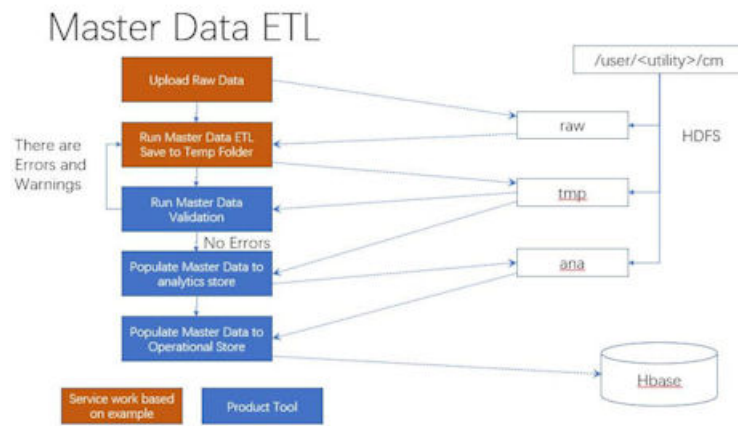


Figure 52. Master data in the ETL module

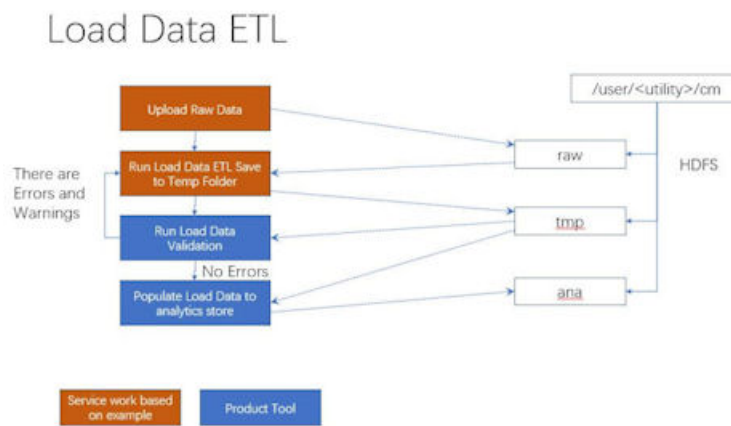


Figure 53. Load data to the ETL module

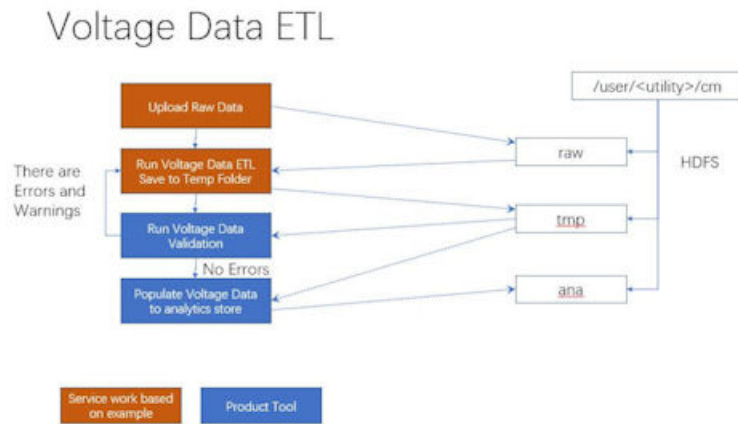


Figure 54. Voltage data in the ETL module

The ETL process includes the group feeder ETL, the exclude time range ETL, the master data ETL, and the reading data ETL. You can find the files on the Jupyter node:

The configure file: /home/<utility_id>/conf/

The python source code: /home/<utility_id>/etl

The shell files: /home/<utility_id>/etl

The ETL results are written to a tmp store, and not into ana store immediately. After the validation procedure, the data is moved from the tmp store to the ana store.

The output files of ETL are in parquet format, use the pyspark to read and show the data of the parquet format. For example, create a readTable.py file in the /home/<utility_id>/etl and run this python file.

```
etl_utility = ETLUtil()
(sc,sqlContext) = etl_utility.init("readTable")
sqlContext.read.parquet(hdfs:///user/cm_sample/cm/tmp/feeder_load).show()
```

The results are:

feeder	timestamp	kwh1	kwh2	kwh3	year	week
NIX08943	2012-01-05 04:00:...	31329.39453125	29201.60546875	38259.37109375	2012	11
NIX08943	2012-01-07 22:00:...	27076.27734375	29999.46484375	41325.609375	2012	11
NIX08943	2012-01-10 02:00:...	35938.890625	27460.6328125	42180.26953125	2012	21
NIX08943	2012-01-06 09:00:...	33440.50390625	31753.73046875	38686.05859375	2012	11
NIX08943	2012-01-08 03:00:...	34617.5	33180.6171875	38329.6484375	2012	11

Figure 55. The results of the ETL process

Group feeder ETL

The group feeder ETL is used to support multiple feeders that belong to one group. The group feeder ETL defines the feeders for the group and how the results are mapped from the SCADA raw data.

Table 18. The output data format of the group feeder ETL			
Column name	Type	Description	Constraints
feeder_group	String	ID of feeder group	Primary Key. Used to group feeders when they share the same SCADA reading
type	String	The type of analysis.	Primary Key. The candidate value includes <i>Load</i> , <i>Voltage</i> , <i>Voltage_with_scada</i> . Note: all feeders in same group are to be analyzed in a single batch.
feeder	String	ID of the feeder.	Primary Key

Exclude time range ETL

The exclude time range is used to support the analysis when the you want to exclude data in a specific time range.

You specify the time range in the raw data as a csv file. And the exclude time range ETL generates the exclude time range table for the analysis.

The input data:

- /user/<utility>/cm/raw/exclude_time
- /user/<utility>/cm/raw/scada
- /user/<utility>/cm/tmp/feeder_group

The output data: /user/<utility>/cm/tmp/exclude_time

<i>Table 19. The output data format of the exclude time range ETL</i>			
Column name	Type	Description	Constraints
feeder	String	ID of feeder	Primary Key Must be a valid feeder in the feeder table or feeder_group in the feeder_group table.
type	String	Type of analysis.	Primary Key Candidate value includes <i>Load, Voltage, Voltage_with_scada</i> .
startTime	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss.	Primary Key The time range to be excluded from the analysis with the startTime being the start time of the range (inclusive).
endTime	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss.	Primary Key. The time range to be excluded from the analysis with the endTime being the end time of the range (exclusive).

Master data ETL

The master data ETL process converts raw data into the master data for the user interface. The master data ETL contains:

- substation region
- electric station
- feeder
- lateral
- transformer
- meter

The details are:

for a substation region:

The input data: /user/<utility_id>/cm/raw/substation_region

The output data: /user/<utility_id>/cm/tmp/substation_region

<i>Table 20. The output data format of the substation region</i>			
Column name	Type	Description	Constraints
substation	String	ID of substation region	Primary Key, Unique, Not Null

Table 20. The output data format of the substation region (continued)

Column name	Type	Description	Constraints
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection.

electric station:

The input data: /user/<utility>/cm/raw/electric_station

The output data: /user/<utility>/cm/tmp/electric_station

Table 21. The output data format of the electric station

Column name	Type	Description	Constraints
substation	String	The ID of the substation region that the station belongs to.	Foreign Key, must be a valid substation in the substation region table
electricStation	String	ID of the electric station.	Primary Key, Unique, Not NULL
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection.

feeder:

The input data:

- /user/<utility>/cm/raw/transformer
- /user/<utility>/cm/raw/overhead_cable
- /user/<utility>/cm/raw/underground_cable

The output data: /user/<utility>/cm/tmp/feeder

Table 22. The output data format of the feeder

Column name	Type	Description	Constraints
feeder	String	The ID of the feeder.	Primary Key, Unique, Not NULL
substation	String	The substation ID.	Foreign Key, must be a valid substation in the substation region table
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection.

lateral:

The input data: /user/<utility>/cm/raw/lateral

The output data: /user/<utility>/cm/tmp/lateral

Table 23. The output data format of the lateral

Column name	Type	Description	Constraints
substation	String	The substation ID.	Foreign Key, must be a valid substation in the substation region table
feeder	String	The ID of the feeder.	Foreign Key, must be a valid feeder in the feeder table
lateral	String	ID of the lateral	Primary Key, Unique, Not NULL
phase	Int	The phase of the lateral	Must be a valid phase code, from 1 to 7 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a line.	Must be valid WKT LINESTRING. In WSG84 projection.
out	String	The cable type.	

transformer:

The input raw data: /user/<utility>/cm/raw/transformer

The output data: /user/<utility>/cm/tmp/transformer

Table 24. The output data format of the transformer

Column name	Type	Description	Constraints
substation	String	The ID of the substation region where the feeder belongs to.	Foreign Key, must be a valid substation in the substation region table
feeder	String	The ID of the feeder where the transformer connected belongs to.	Foreign Key, must be a valid feeder in the feeder table
lateral	String	The ID of the lateral.	
transformer	String	The ID of the transformer.	Primary Key, Unique, Not NULL
phase	Int	The phase of the transformer.	Must be a valid phase code, from 1 to 7 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be valid WKT POINT. In WSG84 projection.
out	String	The transformer type.	

<i>Table 24. The output data format of the transformer (continued)</i>			
Column name	Type	Description	Constraints
kva	String	The kVA level.	
voltage	String	the voltage level.	

meter:

The input data: /user/<utility>/cm/raw/meter

The output data: /user/<utility>/cm/tmp/meter

<i>Table 25. The output data format of the meter</i>			
Column name	Type	Description	Constraints
substation	String	The ID of the substation region where the feeder belongs to.	Foreign Key, must be a valid substation in the substation region table
feeder	String	The ID of the feeder where the transformer connected belongs to.	Foreign Key, must be a valid feeder in the feeder table
lateral	String	The ID of the lateral.	
transformer	String	The ID of the transformer where the meter is connected.	Primary Key, Unique, Not NULL
phase	Int	The phase of the meter.	Must be a valid phase code, from 1 to 7 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be valid WKT POINT. In WSG84 projection.

Reading data ETL

The reading data ETL process uses the reading data, including voltage data and load data and appends the year and week of the reading data timestamp. The year and week data is used as the partition for the incremental saving of the output of the reading data ETL.

In reading data ETL, you can use the command parameters to control the time range of the reading data ETL. The reading data is incrementally saved as the figure shows:

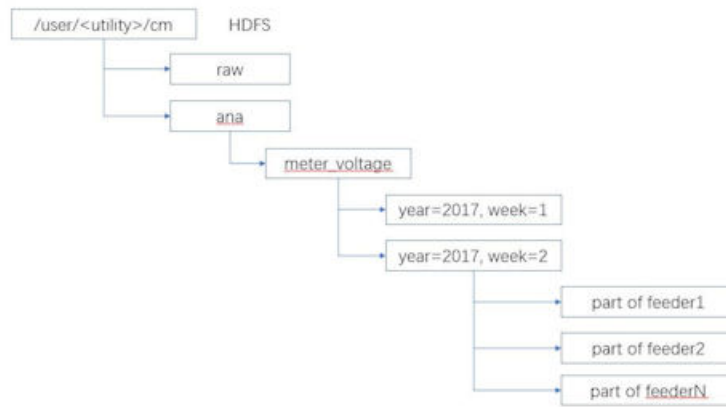


Figure 56. Incremental saves

Note: These notes are for all ETL reading data tables.

1. The timestamp must be in same time zone as the corresponding load table. For example both load tables and reading tables must either use UTC or use the local time zone.
2. The time interval between two timestamps must be fixed and be the same as time interval in corresponding load table. For example, if the time interval is 1 hour, then all readings in meter load table and feeder load table should use 1 hour interval.
3. The timestamp should align with timestamp in the corresponding load table. For example, if the feeder load has a timestamp 9:00 then the meter load timestamp should be the same.
4. The timestamp represents the end edge. For example, if the timestamp is 10:00 and interval used is 1 hour, the kWh is the load between 9:00-10:00.

The ETL reading data files are in the structure:

- feeder_load
- meter_load
- feeder_voltage
- meter_voltage

The input raw data: /user/<utility>/cm/raw/<etl reading data>

The output data: /user/<utility>/cm/tmp/<etl reading data>

To run both the voltage and load reading data ETL, use the command

```
/home/<utility_id>/etl/py_ReadingDataETL.sh
```

To run the voltage reading data ETL, use the command

```
/home/<utility_id>/etl/py_VoltageDataETL.sh
```

To run the load the reading data ETL, use the command

```
/home/<utility_id>/etl/py_LoadDataETL.sh
```

Table 26. The output data format of the feeder load ETL reading data

Column name	Type	Description	Constraint
feeder	String	ID of the feeder	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table or feeder_group in the feeder_group table.
timestamp	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss .	
kwh1	Double	Load of phase A.	The value represents the aggregated active load in past interval. All values are in units of kWh (not MW h) Gap between feeder load and aggregated meter load should less than 10%, otherwise will impact accuracy. The kwh1 is load of phase A that maps to phase code 0b100=4.
kwh2	Double	Load of phase B.	
kwh3	Double	Load of phase C.	

Table 27. Meter load ETL reading data

Column name	Type	Description	Constraint
feeder	String	ID of the feeder.	Primary Key, Foreign Key. The feeder should be a valid feeder in the feeder table, or feeder_group in the feeder_group table.
meter	String	ID of the meter	Primary Key, Foreign Key Must be a valid meter in the meter table
timestamp	Timestamp	Timestamp, yyyy-MM-ddThh:mm:ss.sss	

Table 27. Meter load ETL reading data (continued)

Column name	Type	Description	Constraint
kwh	Double	kWh load	Value represent aggregated active load of all 3 phases in past interval. All values are in units of kWh.
year	Int	Year of the timestamp.	
week	Int	Week of the timestamp.	

Table 28. Feeder voltage ETL reading data

Column name	Type	Description	Constraint
feeder	String	ID of the feeder	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table, or feeder group in the feeder_group table.
timestamp	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss.	
volt1	Double	Phase A voltage	The value represents the average voltage in past time interval. The value must be normalized to the same voltage level. For example, if some meters are 120v, and others are 240v, when the feeder voltage is 1kV, the values is normalized to the minimal voltage level, that is: Value * 120 / 240 When ch1volt, ch2volt, ch3volt represent the voltage of 3 phases, ch1volt maps to phase A, ch2volt maps to phase B, and ch3volt maps to phase C.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	
year	Int	Year of the timestamp.	
week	Int	Week of the timestamp	

Table 29. Meter voltage ETL reading data

Column name	Type	Description	Constraint
feeder	String	ID of the feeder	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table, or feeder group in the feeder_group table.
meter	String	ID of the meter	Primary Key, Foreign Key must be valid meter in meter table
timestamp	String	Timestamp yyyy-MM-ddThh:mm:ss.sss.	
volt1	Double	Phase A voltage	The value represents the average voltage in the past interval. Channels ch1, ch2 and ch3 need not be the same value. For a single phase meter, only one channel should have a value. For two phases meters, only two channels should have values. For 3 phases meters, all channels must have values.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	
year	Int	Year of the timestamp.	
week	Int	Week of the timestamp.	

Validation of the ETL process (optional process)

The validation module gives information about the data from the ETL process and is an optional process. You can make any necessary corresponding changes to the input data before validation.

The data after ETL procedure is saved in the tmp store on HDFS. You can choose to run this validation procedure.

Any errors must be corrected otherwise it might affect the analysis results. You can also define customized validation rules in this module.

The files are located on the Jupyter node:

- The configure file: /home/<utility>/conf/
- The python source code: /home/<utility>/bin
- The shell files: /home/<utility>/bin

Master data validation

Validation of the master data changes according to the asset.

For different assumptions of the data, the validation module gives different validation levels, warning and error.

The input data of validation is from: /user/<utility>/cm/tmp/<asset entity>.

Voltage data validation

Validation of the voltage data focuses on the timestamps and the voltage values levels.

The timestamps describe the process variation of the voltage data that is required for the phase analysis in the analysis module.

The level of the voltage values describes the voltage level of different meters in the feeders. In practice, the voltage drop between the meters and their feeder should not be large.

The input data of validation is from: /user/<utility>/cm/tmp/feeder_voltage or /user/<utility>/cm/tmp/meter_voltage.

- Basic validation
- Timestamp validation - the validation of the fixed time interval and the validation of timestamp alignment between meters and their feeder. The fixed time interval is a basic assumption that the SCADA device provides a fixed time interval. The timestamp alignment of the feeder and the meters are not aligned. This is the basis of the analysis as it compares the variation trend of the feeder and the variation trend of meters in the feeder.
- Voltage data level validation - the validation of the voltage data three parts: the feeder voltage level, the meter voltage level, and the voltage level between the meter and feeder. In practice, the voltage level of the feeder and the meter must be the same. The population standard deviation value is used as the index to verify the voltage level. The user can design different indices to check the voltage level.

Load data validation

Validation of the load data focuses on the timestamps and the load values level.

The timestamps describe the variation process of the load data that is used in the phase analysis in the analysis module. The load values level describes the load level of different meters in feeders. In practice, the load gaps between meters and their feeders should not be large.

The input data of validation is from: /user/<utility>/cm/tmp/feeder_load or /user/<utility>/cm/tmp/meter_load

- Basic validation
- Timestamp validation - the validation of the fixed time interval and the validation of timestamp alignment between meters and their feeder. The fixed time interval is a basic assumption that the SCADA device provides a fixed time interval. The timestamp alignment of the feeder and the meters are not aligned. This is the basis of the analysis as it compares the variation trend of the feeder and the variation trend of meters in the feeder.
- Load data level validation - the validation of the load data level is in three parts: the feeder load level, the meter load level, and the load level between meter and feeder. In practice, the load level of the feeder and the meter must be the same. The population standard deviation value is used as the index to verify the load level. The user can design different indices to check the load level.

Preparing the data for the operational store

After ETL or ETL with validation, you move the data from the tmp store to the ana store, and then populate the operational store.

Move data from the tmp store to the ana store

After ETL or ETL with Validation, the you move the data from the tmp store to ana store on the data on HDFS.

The input raw data: /user/<utility>/cm/tmp/*

The output data: /user/<utility>/cm/ana/*

The master data will overwrite from .tmp to ana.

Populate the operational store

After the ETL procedure, the python script py_PopulateOperationalStore writes the master data into HBase.

The input raw data: /user/<utility>/cm/raw/feeder_voltage.

The analysis process of the data

Four analyzes are provided by the Connectivity Model application that ensures that the required data is made available.

The four analyzes are:

- Load based meter phase analysis
- Voltage based meter phase analysis
- Voltage with SCADA based meter phase analysis
- Lateral transformer phase analysis

Log files and the Analysis result

The log files are created on the Jupyter node under the directory of /home/<utility>/cm/logs/ once the scripts are run. Each type of analysis and persistence has its own sub-directory to keep its log file. The image shows the corresponding sub-directories.

```
[cm_sample@pmo-11b logs]$ ls
2017-06-16T13-49-08-load
2017-06-16T13-53-00-voltage
2017-06-16T13-56-41-voltage_with_scada
2017-06-16T13-59-36-lateral_transformer_phase
2017-06-16T14-00-40-persist
```

Figure 57. Sample log file directory structure

Each sub-directory name includes two parts: the timestamp and type.

You can check all the analysis results on HDFS: `hdfs://user/<utility_id>/cm/<JOB_ID>/data/<type>`

The following image shows the directory structure for the utility_id with JOB_ID of utility_id_sample_job.

```
[cm_sample@pmo-11b ~]$ hdfs dfs -ls cm/job/cm_sample_sample_job/data
Found 4 items
drwxr-xr-x - cm_sample ibmife 0 2017-06-16 13:50 cm/job/cm_sample_sample_job/data/load
drwxr-xr-x - cm_sample ibmife 0 2017-06-16 14:00 cm/job/cm_sample_sample_job/data/ltp
drwxr-xr-x - cm_sample ibmife 0 2017-06-16 13:56 cm/job/cm_sample_sample_job/data/voltage
drwxr-xr-x - cm_sample ibmife 0 2017-06-16 13:59 cm/job/cm_sample_sample_job/data/voltage_scada
```

Figure 58. The directory structure on HDFS

The format of the analysis result for load, voltage and voltage with SCADA is described in the table:

<i>Table 30. The result format for load, voltage, and voltage with SCADA</i>		
Column name	Type	Description
feeder	String	ID of the feeder
meter	String	The ID of the meter
analysis_time	Timestamp	The timestamp when the analysis was run.
phase	Integer	The phase of the meter
confidence	Double	The confidence value

The format of the analysis result for a lateral transformer phase is described in the table:

<i>Table 31. The result format for lateral transformer phase</i>		
Column name	Type	Description
feeder	String	ID of the feeder.
lateral	String	ID of the lateral.
transformer	String	ID of the transformer
analysis_time	Timestamp	The timestamp when the analysis was run.
lateral_phase	Integer	The phase of the lateral.
transformer_phase	Integer	The phase of the transformer.
phase_match	Boolean	If the phase is matched or not.

Configuring the Connectivity Model application

Before you load the data for analysis, the SMTP server, the ETL locations and the time duration for the ETL must be configured on the Jupyter node.

The items that can be configured are described as follows:

<i>Table 32. The configurable elements, descriptions and sample</i>			
Item	Description	Relevant section	Sample
smtp_type	The communication protocol to smtp server; the valid values could be: tls, ssl.	“Automating the data flow” on page 159	tls
smtp_server	The SMTP server address. If a different smtp_type is specified, then the smtp_server should be changed accordingly.	“Automating the data flow” on page 159	smtp.gmail.com:587
smtp_login	The login account used to verify the SMTP server.	“Automating the data flow” on page 159	

Table 32. The configurable elements, descriptions and sample (continued)

Item	Description	Relevant section	Sample
smtp_password	The corresponding password of the login account.	“Automating the data flow” on page 159	
cm_mail_to	Email addresses to receive the automation flow mail. If there are multiple addresses, then separate each one with a comma.	“Automating the data flow” on page 159	
etl_input_path	The ETL input path where the raw csv data is stored.	“Loading the master data for the Connectivity Model application” on page 154, “Loading the reading data for the Connectivity Model application” on page 155	hdfs:///user/cm_sample/cm/raw
etl_staging_path	The ETL intermediate path where the parquet files are stored.	“Loading the master data for the Connectivity Model application” on page 154, “Loading the reading data for the Connectivity Model application” on page 155	hdfs:///user/cm_sample/cm/tmp
etl_output_path	The ETL final path where the parquet files are moved to after verification.	“Loading the master data for the Connectivity Model application” on page 154, “Loading the reading data for the Connectivity Model application” on page 155	hdfs:///user/cm_sample/cm/ana
etl_connectivity_used	The indicator to judge whether the connectivity data is available. By default it is FALSE. If you have connectivity.csv table available, you must change the default value to "TRUE".	“Loading the master data for the Connectivity Model application” on page 154	

Table 32. The configurable elements, descriptions and sample (continued)

Item	Description	Relevant section	Sample
master_etl_spark_defaults_conf	The SPARK default configuration file for the master data ETL. The file name must be specified if the resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file /usr/hdp/current/spark-client/conf/spark-defaults.conf	“Loading the master data for the Connectivity Model application” on page 154	
reading_etl_spark_defaults_conf	The SPARK default configuration file for the reading data ETL. The file name must be specified if the resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file /usr/hdp/current/spark-client/conf/spark-defaults.conf	“Loading the reading data for the Connectivity Model application” on page 155	
analysis_load_spark_defaults_conf	The SPARK default configuration file for load analysis. The file name must be specified if the resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file /usr/hdp/current/spark-client/conf/spark-defaults.conf	“The analysis process of the data” on page 148	

Table 32. The configurable elements, descriptions and sample (continued)

Item	Description	Relevant section	Sample
analysis_load_duration	The duration of the load analysis in days. The value is changed if the load reading has a different duration.	“The analysis process of the data” on page 148	30
analysis_load_until_time	The end of duration time for the load reading data used for the load analysis. The value is changed if the new load reading data has a different time value.	“The analysis process of the data” on page 148	2017-10-09T00:00:00
analysis_load_resource_guard	The resource guard used to control the resource consumption when executing the load analysis. The default value of 0 that indicates there is no limits to the resource consumption. If there are many items in the reading data, then the value should be set based on the number of the meters belonging to the feeder. For example, if there are 5000 meters on one feeder, then a number 20% greater than 5000 should be set.	“The analysis process of the data” on page 148	6000
analysis_voltage_feeders	The feeder list file that contains all of the feeders to be analyzed by the voltage analysis.	“The analysis process of the data” on page 148	../conf/ feeder_voltage.lst

Table 32. The configurable elements, descriptions and sample (continued)

Item	Description	Relevant section	Sample
analysis_voltage_with_scada_spark_defaults_conf	The SPARK default configuration file for the voltage with SCADA analysis. The file name must be specified if resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file /usr/hdp/current/spark-client/conf/spark-defaults.conf	“The analysis process of the data” on page 148	
analysis_voltage_with_scada_duration	The duration of the voltage with SCADA analysis in days. The value must be changed if the voltage reading has a different duration.	“The analysis process of the data” on page 148	30
analysis_voltage_with_scada_until_time	The end of duration time for voltage reading data used to do the voltage with SCADA analysis. It must be changed if the new voltage reading data has a different time value.	“The analysis process of the data” on page 148	2017-10-09T00:00:00
analysis_voltage_with_scada_feeders	The feeder list file that contains the details of all the feeders to be analyzed by voltage analysis.	“The analysis process of the data” on page 148	../conf/feeder_voltage_with_scada.lst
analysis_lateral_transformer_phase_feeders	The feeder list file that contains all details of the feeders to be analyzed by lateral transformer phase analysis.	“The analysis process of the data” on page 148	../conf/feeder_lateral_transformer_phase.lst

Encrypting the SMTP password

You should provide and encrypt the SMTP password in the tenant.cfg file.

Procedure

1. Log into the Jupyter node.

2. Edit the `/home/<utility_id>/conf/tenant.cfg` file to provide the plain text value for the `smtp_password` item.
3. Go to the `/home/<utility_id>/automation` directory.
4. Run the command:

```
./encrypt.sh ../conf/tenant.cfg smtp_password aes
```

The plain text password in the `tenant.cfg` is now encrypted.

Loading data to the Connectivity Model

When you have completed the configuring of the Connectivity Model, you must load your data into HDFS (Hadoop Distribution File System). It should be executed on Jupyter node.

Loading the master data for the Connectivity Model application

The following steps are give as an example of how to do the extracting and loading of the master data.

About this task

In this example set of steps, the **utility_id** is **utility1**, the work folder is `/home/utility1`, and the master data is in the `/home/utility1/raw` folder on the Jupyter node.

Procedure

1. Log into the Jupyter node.
2. To upload the source data to the HDFS, type the command:

```
hdfs dfs -put /home/<utility_id>/raw /user/utility_id/cm
```

3. To support multiple feeders that belong to one group, use the command:

```
/user/<utility_id>/cm/tmp/feeder_group  
/user/<utility_id>/cm/raw/scada
```

Note: The group feeder ETL defines the feeders for the group and how the results are mapped from the SCADA raw data.

4. To generate `feeder_group` in the `tmp` folder, use the command:

```
/home/<utility_id>/etl/py_GroupFeederETL.sh
```

5. To exclude a specific time range from the data, use the command:

```
/user/<utility_id>/cm/raw/exclude_time
```

6. To generate `exclude_time` in the `tmp` folder, use the command:

```
/home/<utility_id>/etl/py_ExcludeTimeRangeETL.sh
```

7. To translate the master data from the `.csv` format to the Parquet file format type the command:

```
py_MasterDataETL.sh
```

For example: You replace the `<utility_id>` by `utility1` in this example.

```
/home/<utility_id>/etl/py_MasterDataETL.sh
```

8. As an optional step for validation of the master data, type the command:

```
/home/<utility_id>/etl/py_MasterDataValidation.sh
```

Loading the reading data for the Connectivity Model application

The following steps are given as an example of how to do the extracting and loading of the reading data.

Before you begin

You must complete the “Loading the master data for the Connectivity Model application” on page 154 before you can load the reading data to the Connectivity Model application.

About this task

In this example set of steps, the **utility_id** is **utility1**, the work folder is `/home/utility1`, and the master data is in the `/home/utility/raw` folder.

Procedure

1. Log into the Jupyter node.
2. Load the reading data from the .csv format to the Parquet format and generate mapping parquet files, type the command:

```
py_LoadDataETL.sh
```

Example 1, to load all data and overwrite the load data in the Parquet folder: you replace the `<utility_id>` with actual `utility1` in this example.

```
/home/<utility_id>/etl/py_LoadDataETL.sh
```

Example 2, to load specific data and to append the existing load data in Parquet folder: you replace the `<utility_id>` with `utility1` and the `<filename>` with the actual file name in this example.

```
/home/<utility_id>/etl/py_LoadDataETL.sh
```

The system writes to `LoadDataETL.out` and system errors write to `LoadDataETL.err`.

3. To extract, transfer, and load the voltage reading data from the input .csv format into standard parquet format and generate mapping parquet files, type the command:

```
py_VoltageDataETL.sh <csvPath> <overwrite>
```

Example 1, extract, transfer and load all voltage data, and overwrite the existing voltage data in the Parquet folder, you replace the `<utility_id>` with actual value.

```
/home/<utility_id>/etl/py_VoltageDataETL.sh
```

Example 2, extract, transfer, and load specific voltage data and to append the existing voltage data in parquet folder, you replace the `<utility_id>` with `utility1` and the `<filename>` with actual file name in this example.

```
/home/<utility_id>/etl/VoltageDataETL.sh
```

The system writes to `VoltageDataETL.out` and system error write to `VoltageDataETL.err`.

4. After you load the reading data, you can validate it.

- To validate load reading data, type the command:

```
/home/<utility_id>/etl/py_LoadDataValidation.sh
```

- To validate the voltage reading data, type the command:

```
/home/<utility_id>/etl/py_VoltageDataValidation.sh
```

- To specify a time duration for the reading data, the format is:

```
/home/<utility_id>/etl/py_<data_type>DataValidation.sh -s 2012-01-01 -e 2012-02-01
```

Populating the master to the operational store

After you have loaded the master and reading data, you need to populate the data to the operational store.

Procedure

1. Log into the Jupyter node.
2. Move the data in the tmp store to the ana store with the command:

```
/home/<utility_id>/etl/py_MoveTmpDataToAnaStore.sh
```

3. To populate the master data to the operational store run the command:

```
/home<utility_id>/etl/py_PopulateOperationalStore.sh
```

Administration of the Connectivity Model application

Before you can start an analysis in the Connectivity Model application, the tenant environment must be available and the variable JOB_ID must be set that is used to group all the following analysis tasks:

- Run an analysis on the Connectivity Model application.
- Populate the results in the operational store so that user interface can show the analysis results.

Tenant environment must be setup for the execution of the analysis.

From the Jupyter node, type the commands:

```
su - cm_sample
```

and for Kerberos authentication

```
kinit -kt /etc/security/keytab/cm_sample.keytab cm_sample@PMQ.IBM.COM
```

The JOB_ID is usually set as a date:

```
JOB_ID=2017-06-10
```

The corresponding configuration items for the analysis in the tenant configuration file should be adjusted according to the reading data which can be different each time when the analysis is run.

```
###  
### load analysis items  
###  
analysis_load_duration=30 #days  
analysis_load_until_time=`date +%Y-%m-%dT00:00:00`  
#the end time of the analysis, change this value if  
necessary to synchrnize with the current reading data  
analysis_load_feeders=./conf/feeder_load.lst  
#feeders to be analyzed  
  
###  
### voltage analysis items  
###  
analysis_voltage_duration=30 #days  
analysis_voltage_until_time=`date +%Y-%m-%dT00:00:00`  
#the end time of the analysis, change this value if necessary to synchrnize  
with the current reading data  
analysis_voltage_feeders=./conf/feeder_voltage.lst #feeders to be analyzed  
analysis_voltage_sample_minutes=60 #analysis algorithm sample minutes  
analysis_voltage_iterations=2  
#analysis algorithm iterations  
  
###  
### voltage with scada analysis items
```



```

####
analysis_voltage_with_scada_duration=30 #days
analysis_voltage_with_scada_until_time=`date +%Y-%m-%dT00:00:00`
#the end time of the analysis, change this value if necessary to
synchronize with the current reading data
analysis_voltage_with_scada_feeders=./conf/feeder_voltage_with_scada.lst #feeders to be
analyzed
analysis_voltage_with_scada_sample_minutes=60
#analysis algorithm sample minutes
analysis_voltage_with_scada_iterations=2
#analysis algorithm iterations

####
#### lateral transformer phase analysis items
####
analysis_lateral_transformer_phase_feeders=./conf/feeder_lateral_transformer_phase.lst
#feeders to be analyzed

```

Optimize the hardware configuration items to maximize performance.

By default, the following three items are not specified as the hardware capability is unknown in the deployment environment.

Specify these values of you environment so that your system resources can be fully utilized.

Note: The values can not exceed the largest configuration value of your system; otherwise the analysis will fail as the resource cannot be allocated.

```

cpu_cores=
driver_memory=
executor_memory=

```

Running the supplied Connectivity Model analyzes

Some environment variables are provided so that the configuration items in the tenant configuration file are not required to be modified each time the data is changed.

When the scale of the reading data is increased, a tuning process adapts the parameters, `cpu_core`, `driver_memory` and `executor_memory` to maximize the running of the analysis.

The environment variables for export are:

```

CPU_CORES
DRIVER_MEMORY
EXECUTOR_MEMORY

```

Export the environment variables for each analysis type:

Load analysis

If you do not want to use the default settings in the tenant `.cfg` file for load analysis, you can replace the settings by exporting the following environment variables:

```

ANALYSIS_LOAD_DURATION
ANALYSIS_LOAD_UNTIL_TIME
ANALYSIS_LOAD_FEEDERS

```

For example:

```

export ANALYSIS_LOAD_DURATION=90
export ANALYSIS_LOAD_UNTIL_TIME=2017-06-18T00:00:00

```

From the Jupyter node, type the command:

```

./bin/run_load_analysis.sh

```

Voltage analysis

If you do not want to use the default settings in the `tenant.cfg` file for voltage analysis, you can replace the settings by exporting the following environment variables:

```
ANALYSIS_VOLTAGE_DURATION  
ANALYSIS_VOLTAGE_UNTIL_TIME  
ANALYSIS_VOLTAGE_FEEDERS
```

For example:

```
export ANALYSIS_VOLTAGE_DURATION=30  
ANALYSIS_VOLTAGE_UNTIL_TIME=2017-10-09T00:00:00  
ANALYSIS_VOLTAGE_FEEDERS=./conf/feeder_voltage.lst
```

From the Jupyter node, type the command

```
./bin/run_voltage_analysis.sh
```

Voltage with SCADA analysis

If you do not want to use the default settings in the `tenant.cfg` for voltage with SCADA analysis, you can replace the settings by exporting the following environment variables:

```
ANALYSIS_VOLTAGE_WITH_SCADA_DURATION  
ANALYSIS_VOLTAGE_WITH_SCADA_UNTIL_TIME  
ANALYSIS_VOLTAGE_WITH_SCADA_FEEDERS
```

for example:

```
export ANALYSIS_VOLTAGE_WITH_SCADA_DURATION=30  
ANALYSIS_VOLTAGE_WITH_SCADA_UNTIL_TIME=2017-10-09T00:00:00  
ANALYSIS_VOLTAGE_WITH_SCADA_FEEDERS=./conf/feeder_voltage_with_scada.lst
```

From the Jupyter node, type the command:

```
./bin/run_voltage_with_scada_analysis.sh
```

Lateral transformer phase analysis

If you do not want to use the default settings in the `tenant.cfg` for lateral transformer phase analysis, you can replace the settings by exporting the following environment variable.

```
ANALYSIS_LATERAL_TRANSFORMER_PHASE_FEEDERS
```

For example:

```
export ANALYSIS_LATERAL_TRANSFORMER_PHASE_FEEDERS=./conf/feeder_lateral_transformer_phase.lst
```

From the Jupyter node, type the command:

```
./bin/run_lateral_transformer_phase_analysis.sh
```

Populating the analysis results to the operational store

After you have run the analysis is run, the results must be populated into the operational store so that user interface can show the results of the analysis.

Procedure

1. Log into the Jupyter node and go to the tenant home, for example:
 `/home/cm_sample`
2. Type the command:

```
./bin/persist_result.sh <type>
```

The <type> can be:

- *raw*: Populates the analysis result for the review purpose and user interface does not get the result until `persist` is called with the aggregate specified.
 - *aggregate*: Submits the raw result so that the user interface shows the results.
 - *all*: Both raw and aggregate results show in the user interface.
3. Verify the analysis results in the user interface. See the section [“Using the connectivity model application”](#) on page 169

Automating the data flow

To simplify the use of the Connectivity Model application, three flow scripts are delivered with IBM IoT for Energy and Utilities

Before you begin

Before executing the flow scripts, the configuration items listed in the [“Configuring the Connectivity Model application”](#) on page 149 must be complete.

The incoming master and reading data must be prepared and be in the `/home/<utility>/staging` directory. The files must be in a .zip format and the names must start with `master_data_*.zip` and `reading_data_*.zip` respectively. For example:

```
master_data_2017-12-19.zip
```

and

```
reading_data_2017-12-19.zip
```

Procedure

1. Log into the notebook node as a tenant user with access rights to HDFS and Hbase and start the master data flow.
 - a) Open the `/home/<utility>/automation` directory.
 - b) Run the script: `./master_data_flow.sh`

The output example for the zip file `master_data_2017-12-19.zip`:

```
master_data_2017-12-19.log
master_data_2017-12-19.report
master_data_2017-12-19.success
```

Figure 59. The example output

Where `master_data_2017-12-19.log` is the log directory, `master_data_2017-12-19.report` contains the quality report, and `master_data_2017-12-19.success` indicates that the flow was completed without errors.

2. From the notebook node, start the reading data flow.
 - a) Open the `/home/<utility>/automation` directory.
 - b) Run the script: `./cm_automation/reading_data_flow.sh`.

Note: Export the `ANALYSIS_LOAD_UNTIL_TIME` and `ANALYSIS_VOLTAGE_UNTIL_TIME` environment variables with a suitable time for the reading data before running the script if the latest time in the reading data is not yesterday.

The output example for the zip file `reading_data_2017-12-19.zip`:

```
reading_data_2017-12-19.log
reading_data_2017-12-19.report
reading_data_2017-12-19.success
```

Figure 60. The reading data output

Where `reading_data_2017-12-19.log` is the log directory, `reading_data_2017-12-19.report` contains the quality report, and `reading_data_2017-12-19.success` indicates that the flow was completed without errors.

3. From the notebook node, start the analysis flow.
 - a) Open the `/home/<utility>/automation` directory.
 - b) Run the script:

```
./cm_automation/analysis_flow.sh
```

Note: Export the `ANALYSIS_LOAD_UNTIL_TIME` and `ANALYSIS_VOLTAGE_UNTIL_TIME` environment variables with a suitable time if necessary.

The output example for the zip file `analysis_flow_2018-01-08.zip`:

```
analysis_flow_2018-01-08.log
analysis_flow_2018-01-08.report
analysis_flow_2018-01-08.success
```

Figure 61. The analysis flow output

Where `analysis_flow_2018-01-08.log` is the log directory, `analysis_flow_2018-01-08.report` contains the quality report, and `analysis_flow_2018-01-08.success` indicates that the flow was completed without errors.

- c) To enable or disable voltage or load analysis open `analysis_flow.sh` in a text editor.
 - 1) Open the file `analysis_flow.sh` in a text editor.
 - 2) Edit the **tasks** part of this file.
 - To enable the load or voltage analysis remove the comment symbol `#`.
 - To disable the load or voltage analysis add the comment symbol `#`.

Examples:

To enable voltage analysis:

```
"voltage_analysis" "run_voltage_analysis"
#"load_analysis" "run_load_analysis"
```

To enable load analysis:

```
#"voltage_analysis" "run_voltage_analysis"
"load_analysis" "run_load_analysis"
```

4. Schedule the flows with crontab script.

The three flows can be run separately, or scheduled with crontab for a specified time.

- a) Log in as a tenant user.
- b) Run the command:

```
crontab -e
```

- c) Put the contents into a text editor.

Note: For the format of the crontab file, please refer to the Linux crontab guide.

An example crontab file:

```
#specify time zone to be used. Right now it is UTC timezone
CRON_TZ=UTC
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
# specify the necessary environment variables
#PYTHON_LIB=
#SPARK_HOME=

#uncomment the following env variables to adjust
the time used in the flow if needed
#ANALYSIS_LOAD_UNTIL_TIME=2016-09-01
#ANALYSIS_VOLTAGE_UNTIL_TIME=2016-08-31
#LOAD_UNTIL_TIME=2016-09-01
#VOLTAGE_UNTIL_TIME=2016-08-31

#master_data_flow & reading_data_flow
#master data flow scheduled at 14:00 every day, UTC timezone
00 14 * * * $HOME/automation/master_data_flow.sh &
#reading data flow scheduled at 15:00 every day, UTC timezone
00 15 * * * $HOME/automation/reading_data_flow.sh &

#analysis_flow scheduled at 18:00 every Saturday, UTC timezone
00 18 * * sat $HOME/automation/analysis_flow.sh &
```

Automating the data flow - Quality Reports

Several quality reports are generated after the data flow automation has been executed:

These reports are useful tools that verifies the quality of the data and analysis result.

- Master data quality report
- Reading data quality report
- Analysis result report

The email address that are that are configured in the item `cm_mail_to` to receive the reports as an attachment.

The following sections introduce each report and describes how to interpret them.

Master data quality report

The Master data quality report shows the different types of master data in the form of radar charts and tables. The values in the tables are indexed to measure the data quality: The larger of the value, the better the quality of the data.

Feeder

A feeder has three quality attributes defined:

- `distinct_feeder`: The percentage of the feeders with unique names. Each feeder should have a unique name.
- `substation`: The percentage of feeder that have defined data for substations.
- `geometry`: The percentage of feeders that have the geometry defined.

The example shows a data quality chart and table for a feeder:

	distinct_feeder	substation	geometry
0	100.0	100.0	0.0

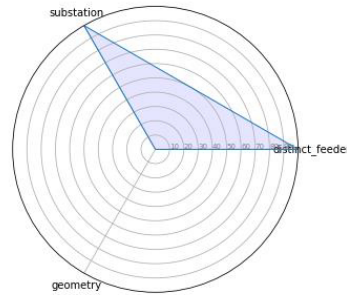


Figure 62. Feeder quality report

Lateral

A lateral has seven quality attributes defined:

- distinct_lateral: The percentage of the laterals with unique names. Each lateral should have a unique name.

Note: If two feeders have the same name, then the laterals that belong to those two feeders can also have duplicated names. The value of distinct_lateral will not be 100.

- substation: The percentage of laterals that have defined data for substations.
- feeder: The percentage of laterals that have defined data for feeders.
- phase: The percentage of laterals that have defined data for phase.
- nPhases: The percentage of laterals with single phase.
- geometry: The percentage of laterals that have geometry defined.
- ou: The percentage of laterals that have ou defined.

The example shows a data quality chart and table for a lateral:

	distinct_lateral	substation	feeder	phase	nPhases	geometry	ou
0	100.0	100.0	100.0	100.0	100.0	0.0	100.0

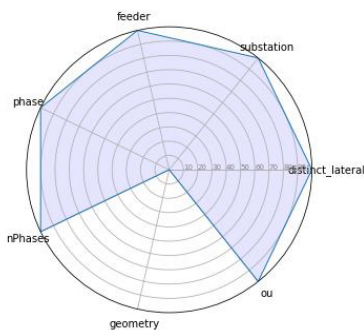


Figure 63. The lateral data quality

Transformer

A transformer has nine quality attributes defined:

- Distinct_transformer: The percentage of the transformers with unique names.

Note: If two laterals have the same name, then the transformers that belong to those two laterals can also have duplicated names. The value of distinct_transformer will not be 100. This also applies for duplicated feeder names.

- Substation: The percentage of transformers that have defined data for a substation.

- Feeder: The percentage of transformers that have defined data for a feeder.
- Lateral: The percentage of transformers that have defined data for a lateral.
- phase: The percentage of transformers that have defined phase data.
- nPhase: The percentage of transformers with single phase.
- geometry: The percentage of transformers that have defined geometry data.
- ou: The percentage of transformers that have defined ou data.
- kva: The percentage of transformers that have defined kVA data.
- has_meters: The percentage of transformers that have meters defined.

The example shows a data quality chart and table for a transformer:

	distinct_transformer	substation	feeder	lateral	phase	nPhases	geometry	ou	kva	has_meters
0	99.89833	100.0	100.0	95.599129	100.0	100.0	100.0	95.599129	100.0	78.329702

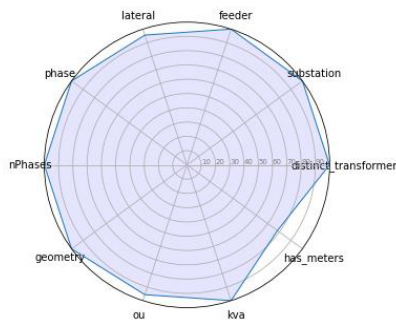


Figure 64. The transformer data quality

Meter

A meter has nine quality data attributes defined:

- distinct_meter: The percentage of the meters with unique names.

Note: If two transformers have the same name, then the meters that belong to those two transformers can also have duplicated names. The value of distinct_meter will not be 100. This also applies for duplicated lateral and feeder names.
- substation: The percentage of meters that have defined data for a substation.
- feeder: The percentage of meters that have defined data for a feeder.
- lateral: The percentage of meters that have defined data for a lateral.
- transformer: The percentage of meters that have defined data for a transformer.
- phase: The percentage of meters that have defined phase data.
- nPhase: The percentage of meters with single phase.
- geometry: The percentage of meters that have defined geometry data.
- meter_transformer_distance: The percentage of meters with a suitable distance to the corresponding transformers without being excluded by the spatial filtering.

The example shows a data quality chart and table for a meter:

	distinct_meter	substation	feeder	lateral	transformer	phase	nPhases	geometry	meter_transformer_distance
0	99.24569	100.0	100.0	94.12069	100.0	100.0	100.0	100.0	99.353448

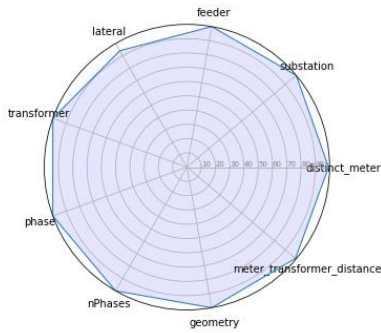


Figure 65. The meter data quality

Reading data quality report

Reading data quality report shows the different types of reading data in the form of radar charts and tables. The values in the tables are indexed to measure the data quality: The larger of the value, the better the quality of the data.

Feeder load

Feeder load has six quality data attributes defined:

- feeder_coverage: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- timestamp_coverage: The percentage of the timestamp coverage.
- timestamp_duplication: The percentage of timestamps that have duplicated values.
- kwh1: The percentage of valid kWh values on phase 1.
- kwh2: The percentage of valid kWh values on phase 2.
- kwh3: The percentage of valid kWh values on phase 3

The example shows a data quality chart and table for feeder load:

	feeder_coverage	timestamp_coverage	timestamp_duplication	kwh1	kwh2	kwh3
0	45.833333	97.777778	21.221064	99.98295	99.983562	99.424462

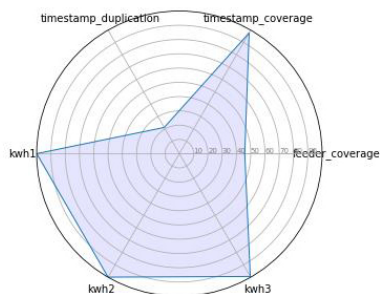


Figure 66. The feeder load data quality

Meter load

The average aggregated meter load gap against the feeder load is used to measure the meter load data quality. An example is as below:

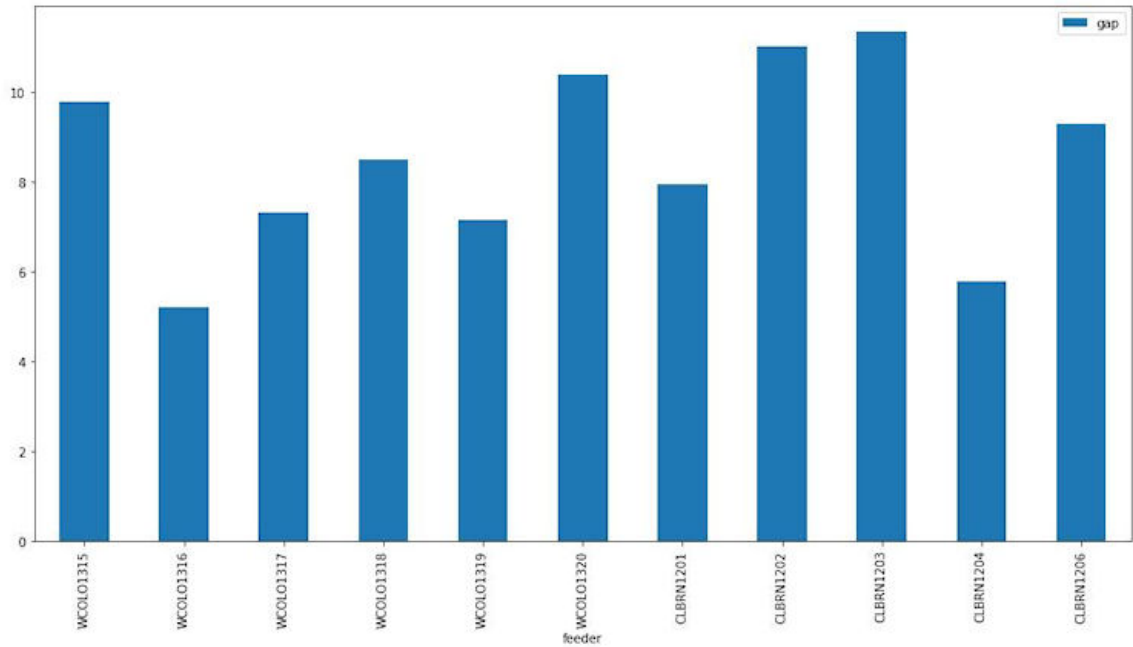


Figure 67. The meter load gap

Meter load has four data attributes defined:

- **feeder_coverage**: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- **meter_coverage**: The percentage of meters covered by the reading data flow. Some meters may not be recorded in the reading data for the specified time range.
- **distinct_timestamp**: The percentage of the timestamps with unique values.
- **duplicate_timestamp**: The percentage of timestamps with non-duplicated values.
- **kwh**: The percentage of valid kWh values.

The example shows a data quality chart and table for meter load:

	feeder_coverage	meter_coverage	distinct_timestamp	duplicate_timestamp	kwh
0	95.833333	98.434783	96.864307	99.547199	95.426399

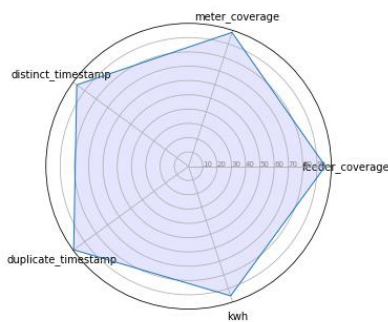


Figure 68. The meter load data quality

Feeder voltage

Feeder voltage has six data attributes defined:

- **feeder_coverage**: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- **timestamp_coverage**: The percentage of the timestamp coverage.

- timestamp_duplication: The percentage of timestamps with non-duplicated values.
- volt1: The percentage of valid volt values on phase 1.
- volt2: The percentage of valid volt values on phase 2.
- volt3: The percentage of valid volt values on phase 3.

The example shows a data quality chart and table for feeder voltage:

	feeder_coverage	timestamp_coverage	timestamp_duplication	volt1	volt2	volt3
0	95.833333	97.777778	48.676537	98.351038	80.959733	80.687085

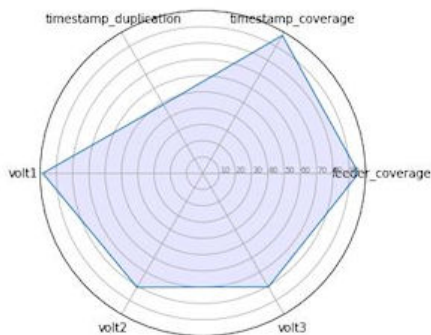


Figure 69. The feeder voltage data quality

Meter voltage

Meter voltage has seven data attributes defined:

- feeder_coverage: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- meter_coverage: The percentage of meters covered by the reading data flow. Some meters may not be recorded in the reading data for the specified time range.
- distinct_timestamp: The percentage of the timestamps with unique values.
- duplicate_timestamp: The percentage of timestamps with non-duplicated values.
- volt1: The percentage of valid volt values on phase 1.
- volt2: The percentage of valid volt values on phase 2.
- volt3: The percentage of valid volt values on phase 3.

The example shows a data quality chart and table for meter voltage:

	feeder_coverage	meter_coverage	distinct_timestamp	duplicate_timestamp	volt1	volt2	volt3
0	95.833333	91.956522	92.624585	99.566525	95.788308	12.475503	12.474821

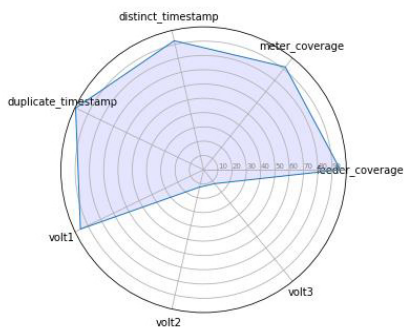


Figure 70. The meter voltage data quality report

Analysis result report

Analysis result report shows the quality of the different types of analysis reports in the form of radar charts and tables. The values in the tables are actual values of the measured data.

Meter coverage analysis for load

The meter coverage analysis for load has six data attributes defined:

- Total: The total number of meters.
- Analyzed: the total number of meters in the analysis.
- Not_analyzed_poly_phase: The number of meters not analyzed that have multiple phases.
- Not_analyzed_missing_load_data: The number of meters not analyzed due to missing load data.
- Not_analyzed_spatial_filtered: The number of meters not analyzed due to being filtered by spatial condition.
- Not_analyzed_other: The number of meters not analyzed due to other factors.

The example shows a chart and table for analysis result report for load:

	total	analyzed	not_analyzed_poly_phase	not_analyzed_missing_load_data	not_analyzed_spatial_filtered	not_analyzed_other
0	26021	24229	1711	81	0	0

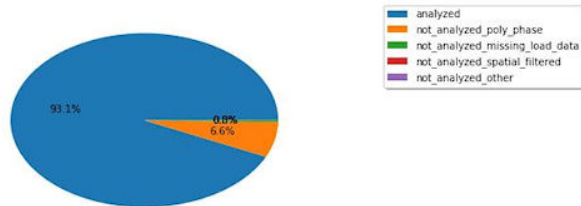


Figure 71. The meter coverage analysis report for load

Accuracy analysis for load

Load accuracy analysis is a statistics chart based on the feeder. An example is as below:

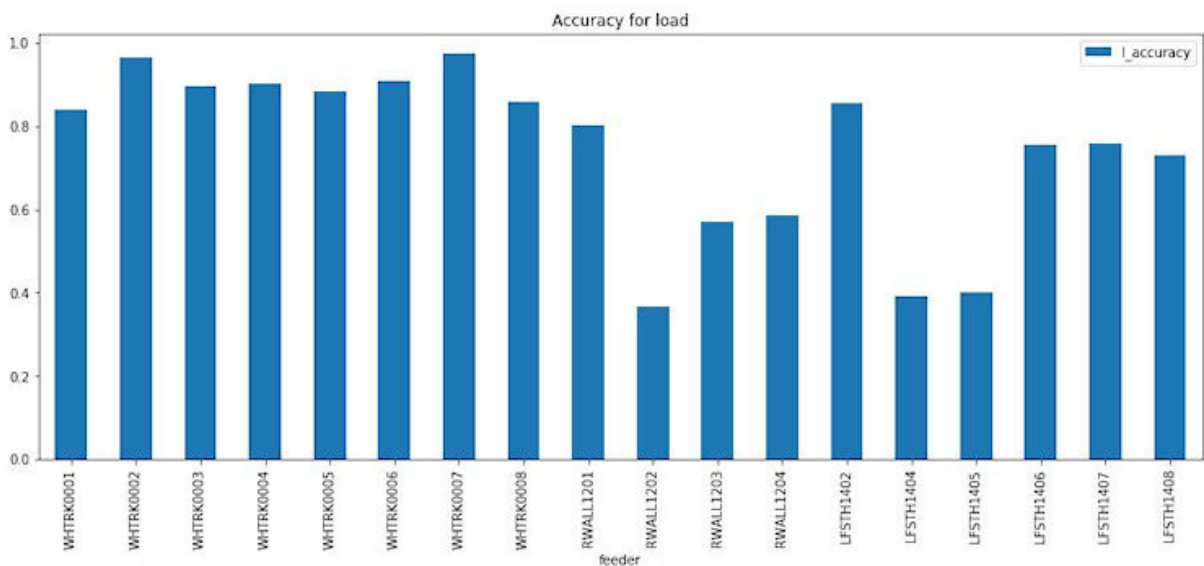


Figure 72. The accuracy analysis for load chart

Meter coverage analysis for voltage

The meter coverage analysis for voltage has six data attributes defined:

- Total: The total number of meters.
- Analyzed: the total number of meters in the analysis.
- Not_analyzed_poly_phase: The number of meters not analyzed that have multiple phases.
- Not_analyzed_missing_voltage_data: The number of meters not analyzed due to missing load data.
- Not_analyzed_spatial_filtered: The number of meters not analyzed due to being filtered by spatial condition.
- Not_analyzed_other: The number of meters not analyzed due to other factors.

The example shows a chart and table for analysis result report for voltage:

	total	analyzed	not_analyzed_poly_phase	not_analyzed_missing_voltage_data	not_analyzed_spatial_filtered	not_a
0	23288	20854	2150	176	108	0

: <matplotlib.legend.Legend at 0x7fd15cd87b10>

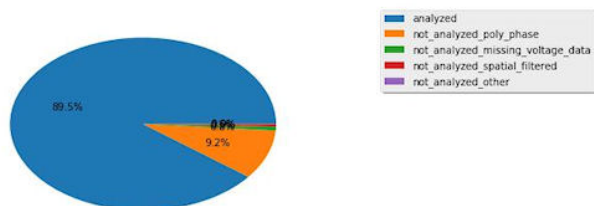


Figure 73. The meter coverage analysis report for voltage

Accuracy analysis for voltage

The voltage accuracy analysis chart shows the statistics based on the feeder. An example is as below:

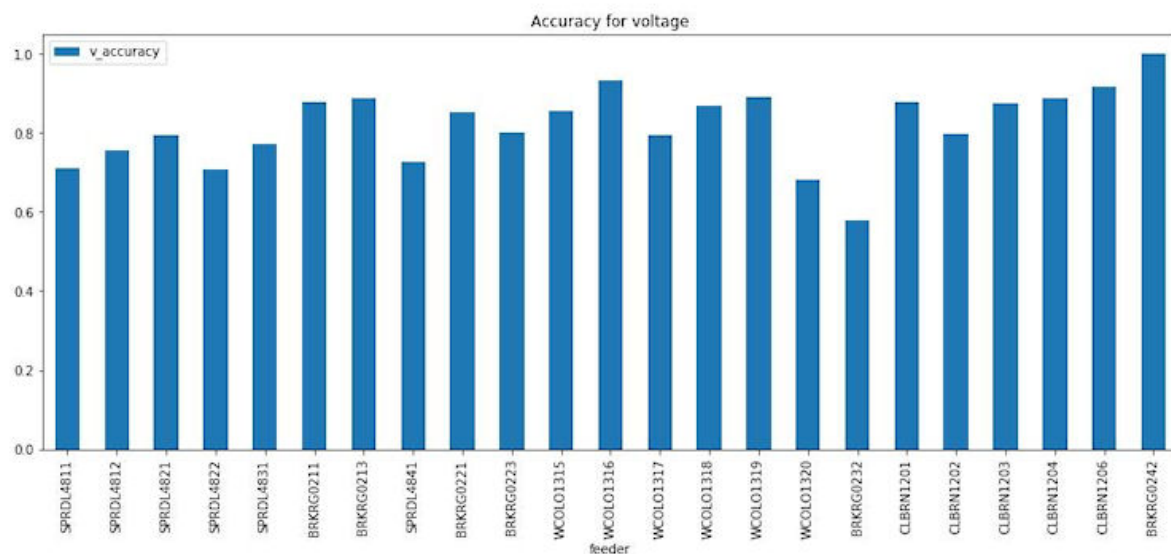


Figure 74. The accuracy analysis for voltage chart

Accuracy distribution for voltage

The accuracy distribution for voltage analysis is a range based chart, from which you can view the number of feeders in each accuracy range (0-100). An example is as below:

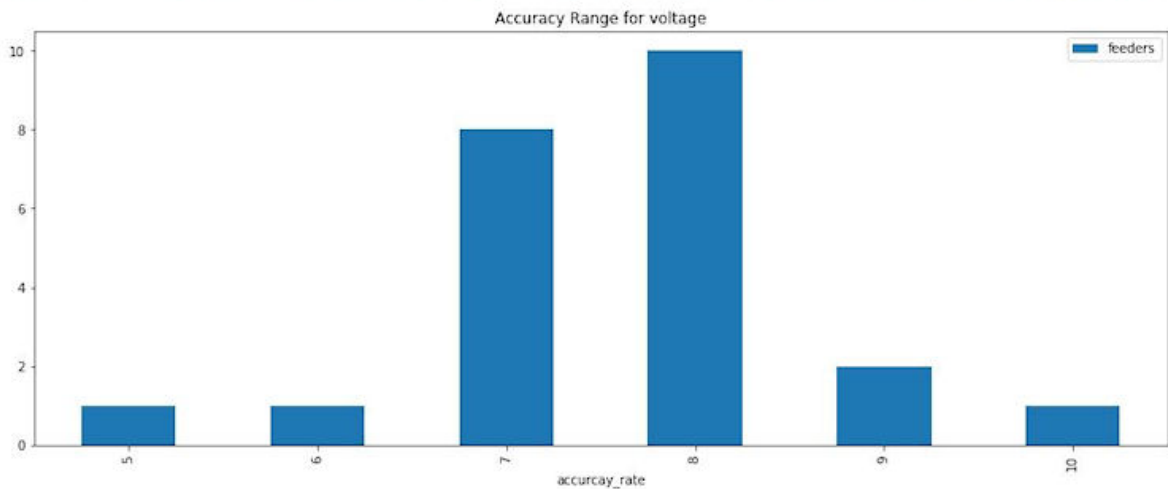


Figure 75. The accuracy distribution for voltage chart

Disabling and enabling the load and voltage algorithms

Configure the load and voltage algorithms in IBM IoT for Energy and Utilities.

About this task

You need to log into the Jupyter Notebook, and in the automation folder find the file `analysis_flow.sh` is in the notebook container at `/home/<utility>/automation/analysis_flow.sh`. The container runs on the notebook node.

Procedure

1. Log in to the Jupyter node as a tenant user.
2. Open the `/home/<utility>/automation` directory.
3. Open the file `analysis_flow.sh` in a text editor.
4. Edit the **tasks** part of this file.
 - To enable the load or voltage analysis remove the comment symbol `#`.
 - To disable the load or voltage analysis add the comment symbol `#`.

For example:

To enable voltage analysis:

```
"voltage_analysis" "run_voltage_analysis"
#"load_analysis" "run_load_analysis"
```

To enable load analysis:

```
#"voltage_analysis" "run_voltage_analysis"
"load_analysis" "run_load_analysis"
```

Using the connectivity model application

To use the connectivity model application in IBM IoT for Energy and Utilities, you must have completed the loading of your data to the application.

When you first open the Connectivity Model application you are presented with a map of the area of the utility with breadcrumbs for the utility name, substations and feeders.

Logging onto the Connectivity Model application

Log on to access the IBM IoT for Energy and Utilities user interface for the Connectivity Model application.

Before you begin

Contact your local administrator to obtain your user ID and password. Your administrator is responsible for ensuring that you have the security access level that is appropriate to your role in your organization. Your administrator will also supply you with the web address URL for accessing the solution portal.

About this task

Use the following procedure to start a new browser session and access IoT for Energy and Utilities.

Utility name, substation, and feeder

Area of the utility

User ID

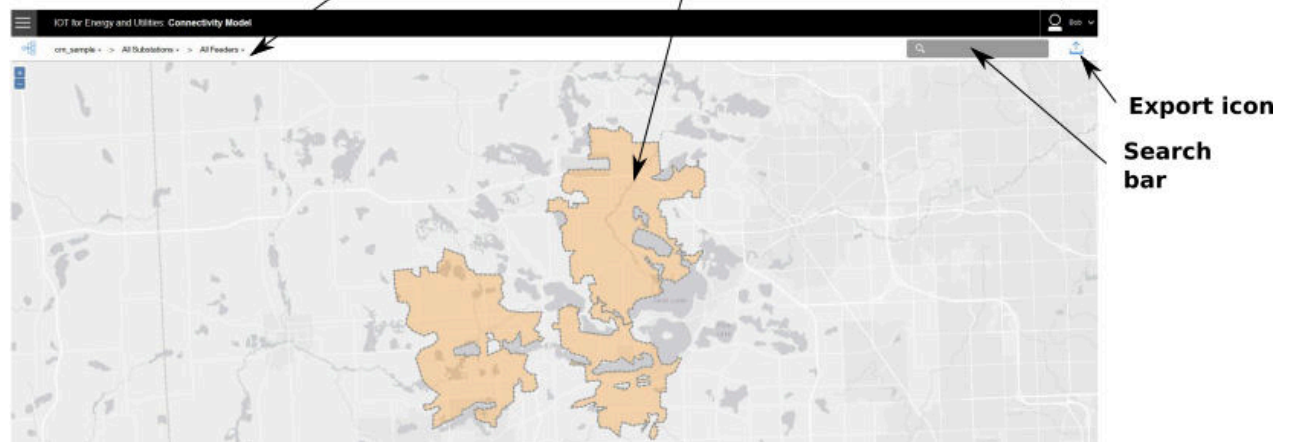


Figure 76. The login screen of the Connectivity Model application

Procedure

1. Enter the URL into the address field of the browser.

Note: The fully qualified domain name is required in the URL, for example, `https://web_hostname/ibm/#page_cmodel` where `web_hostname` is the host name of the web server. If you use the IP address instead of the registered fully qualified domain name, some windows do not open correctly. Also, if you do not use the `https` protocol, the link is redirected to use the `https` protocol.

2. On the login page, enter your user ID and password.
3. Click **Log In**.
4. Click the down arrow, and click **Energy > Connectivity Model**.

Results

Only the pages, features, and data that you have permission to access are displayed. Contact your administrator if you require more access.

Viewing the legend of the connectivity model application

The legend shows the icons used for the assets in the connectivity model and the colors indicating their status.

About this task

Via the legend you can filter the assets that show on the map for the connectivity model.

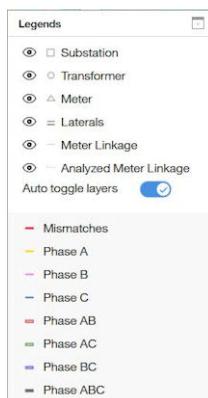


Figure 77. The legend for the Connectivity Model map

The assets that show in the legend are:

- Substation
- Transformer
- Meter
- Laterals
- Meter Linkage
- Analyzed Meter Linkage

The eye icon hides or displays the asset type on the map.

Different asset types are represented by different shapes on the map.

The **Auto toggle layers** switch enables the feature to hide or show assets when zoomed in.

The red color indicates the assets that have a mismatch status. The other colors represent the different phases of the assets.

Procedure

1. Select **Connectivity Model** from the menu bar.
2. Click the first drop-down button in the search bar and select the utility you need.
3. Click the second drop-down button to choose the substation.

The Legends window opens.

Viewing the basic information of a substation and its feeder

You can view a substation and its feeders.

About this task

After you have selected utility, you can see all the substations in the selected utility. The substation information contains two parts:

- The name of the substation.
- The total number of asset phase and connection errors if errors are present.

The drop-down menu of a feeder is almost the same with the one of substation. You can open it by clicking the third drop-down button.

Procedure

1. In IBM IoT for Energy and Utilities, click the first drop-down button. Click the utility you want to view.
2. Click the second drop-down button to select the substation you want to view.

3. To search for a certain sub-station, in the search field type a letter contained in the name of the sub-station.
4. Click on the name of the sub-station you want to view.
5. Click the third drop-down button to select the feeder you want to view.
6. Zoom in on the map to see the clusters of transformers connected to the sub station, and again to see the clusters of meters connected to the transformers.

Viewing the detailed information for transformers and the meters

In IBM IoT for Energy and Utilities you can view more details of a transformer and meters in the hover card and preview panel.

About this task

When you click a cluster of transformers (circle that contains a number) a card shows all the transformers in the cluster and their names.

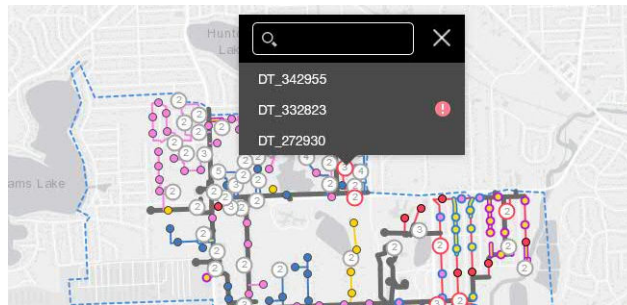


Figure 78. Cluster of transformers

When you click a cluster of meters (diamond that contains a number) a similar card shows all the meters in the cluster and their names.

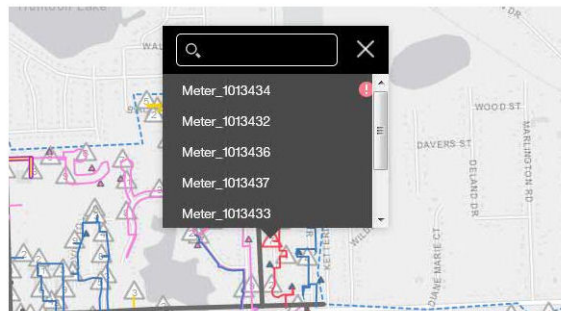


Figure 79. Cluster of meters

When you click a lateral (a thick line) a hover card shows the basic details of the lateral: the name of the lateral, the feeder details, the phase and the connectivity state, the number of transformers on the lateral and the number of meters.

When you click a transformer (small circle) a hover card shows the basic details of the transformer: the name of the transformer, the feeder of the transformer, the lateral, the number of meters, the phase and connectivity state.

When you click a meter (small diamond) a hover card shows the basic details of the meter: the name of the meter, the feeder of the meter, the lateral, the phase and connectivity state.

When you click on the asset in the hover card, the summary panel shows that contains more details the hierarchy tree of the assets and the detailed information about the selected asset. Click on the details tab to see the basic information and property details.

Procedure

1. In IoT for Energy and Utilities, click the first drop-down menu button and click the utility you want to view.
2. Click the second drop-down menu button and click the name of the sub-station you want to view.
3. Zoom in on the map to see the clusters of transformers connected to the sub station, and again to see the clusters of meters connected to the transformers.
4. Hover over a transformer on the map to see the general details of a transformer.
5. Click on the transformer to see the preview panel for the transformer.
6. Click on the meter to see the preview panel for the meter.
7. Click on the asset in the hierarchy tree to see the details of different assets.

Showing the confidence level of the connectivity results

A confidence score shows for meters and transformers for the level of confidence of the connectivity result.

About this task

The confidence score given for a meter is the level of confidence for the result as a percentage value. The confidence score given for transformer is the weighted average of the meters that are supplied by the transformer.

Procedure

1. Select **Connectivity Model** from the menu bar.
2. Click the first drop-down button in the search bar and select the utility you need.
3. Click the second drop-down button to choose the substation.
4. Click the third drop-down button to choose the feeder.
5. From the map, click a transformer, shown as a circle, that you want to view.
6. The **Connectivity Summary** screen shows:

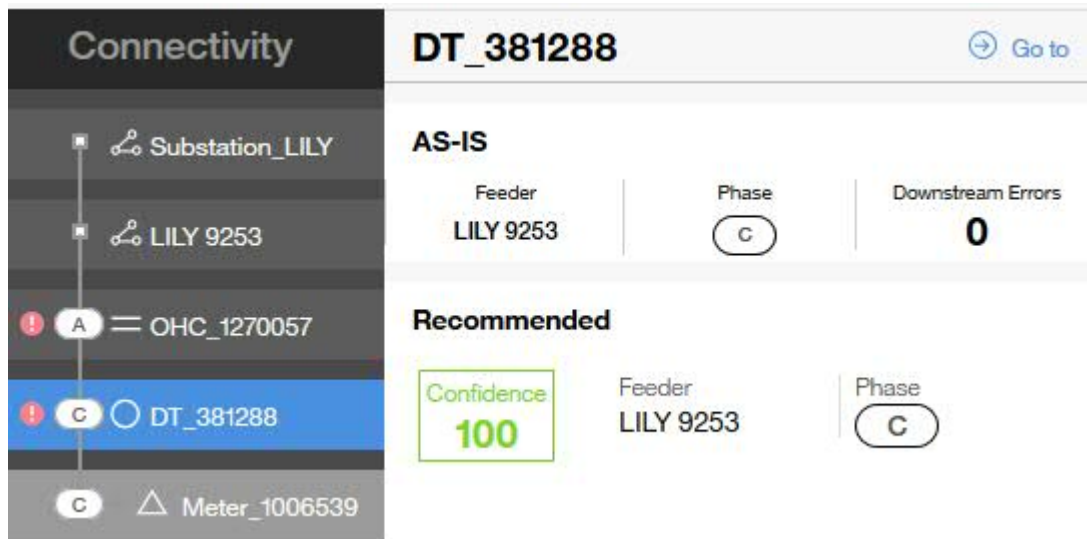


Figure 80. The connectivity results with the level of confidence for a transformer

7. From the **Connectivity Summary** screen click a meter, shown as a triangle, downstream from the transformer.

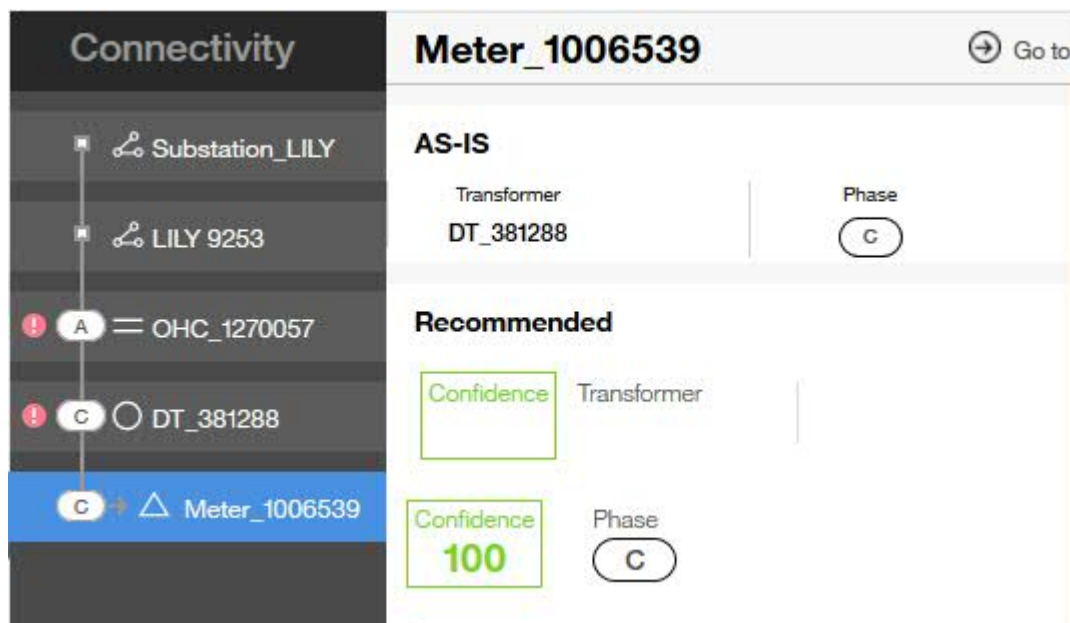


Figure 81. The connectivity results with the level of confidence for a meter

Exporting the asset information of a utility

You can export key performance indicators or asset information from the selected utility using the export facility.

About this task

In the **Export Asset** tab you can select to export the details on transformers or meters to your own system. In the **Export KPI** tab you can export the key performance indicators by utility or by sub-station. You can also select the date of the KPI.

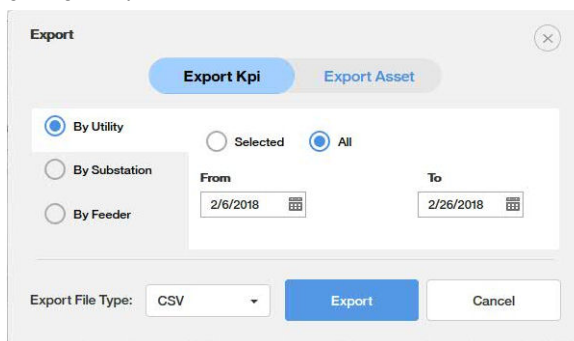


Figure 82. The export dialog box

Procedure

1. Click the Export icon in the search bar. [↑](#)
2. Select either **Export KPI** or the **Export Asset** tab.
3. Select the conditions.
 - **By Utility**
 - **By Substation**
 - **By Feeder**
 - **From** date **To** date
 - **Export File Type**
4. Click **Export**. You can either view or save the exported data.

Chapter 6. Optimizing wind farm operations

IBM IoT for Energy and Utilities provides situational awareness for wind farm assets to help optimize wind farm performance and assess turbine health and risk.

Overview of the Asset 360 for Wind application

The IBM IoT for Energy and Utilities application Asset 360 for Wind provides an advisor for wind farm operations and maintenance.

The Wind 360 application integrates the operational systems of the wind farm to deliver predictive, prescriptive, and cognitive solutions that improves the operational and maintenance efficiency of the wind farm.

The Asset 360 for Wind application dashboard

The Asset 360 for Wind application for IBM IoT for Energy and Utilities has dashboards that have differing access rights dependent on the user profile.

Deputy manager - KPI Dashboard

The key performance indicator dashboard consists of reports that keep the user aware of the current performance of the wind farm without requesting further information. After logging in, the user with KPI Dashboard privileges can view the KPI dashboard and see an overall view of the performance of the wind farm.

Farm KPI report

The KPI report dashboard shows the overall operation index of the wind farm. The index is developed from four dimensions: power production, turbine availability, wind power conversion rate and turbine health. The deputy manager can also see the overall ranking among similar wind farms. The color of the scores represents different ranges: red: <60, yellow: 60-80, green: >=80.

Power Generation report

The Power Generation report shows the power production in mega watts, loss due to down time and the total lost time compared over a three week period which is the previous complete two weeks and current week.

MTBF and MTBR report

MTBF - Mean Time Between Failure

MTTR - Mean Time To Restore

The MTBF and MTBR report shows the comparison between MTBF and MTTR over the previous eight months. The time period can be altered.

Maintenance Cost report

The Maintenance Cost report compares the cost of maintenance of the turbines over the same time period of this year and last year on a monthly basis. You can use the calendar to choose the time period you want to view.

Utilization Hours report

The Utilization Hours report shows the power generated, loss time and utilization hours during the period of time selected from the calendar.

Utilization hour = power generation / turbine full capacity. The Utilization Hours report is the most important KPI for a wind farm.

Weather Forecast report

The Weather Forecast report, shows the weather information such as temperature, wind speed, wind direction, and humidity. You are also provided with maintenance suggestions according to the weather condition. The wind speed shows also for the next 12 hours.

All the weather data is from The Weather Company API.

Operational Engineer - Monitoring dashboard

The operational engineer is responsible for individual turbine and overall farm monitoring. The engineer needs to know the current state of each turbine within the wind farm.

The Map and list views

The Map view shows a visualization of data items associated with their relevant positions on the map. Using the information displayed on the map together with the list view, you can identify location patterns. A score for each turbine indicates the working effectiveness of each turbine.

You can view the basic information for each turbine by hovering over the icon representing the turbine. The hover card shows the turbine serial number, working status, index score, and wind speed. The icon color is consistent with the legend color.

Wind Speed and Direction Trend report

The Wind Speed and Direction Trend report shows the wind condition for the farm. You can hover on it to see the condition for a specific day. The deep blue part of the report represents the past, and the light blue part represents the forecast for the future days.

Farm report

The Farm report shows the three reports:

- Power generation report
- Utilization Hours report
- Turbine Availability report

Turbine report

The Turbine report shows three reports for an individual turbine:

- Turbine KPI report shows the overall operation index for a single turbine. The index is developed from four dimensions: power production, turbine availability, wind power convention rate and turbine health. The color of the scores represents different ranges: red: <60; yellow: 60-80, green: >=80.
- Power curve report shows the actual power curve of the wind turbine compared to the conceptual power curve. If the curves differ greatly, then the turbine is not operating in its optimum state.
- Utilization Hours report shows the utilization hours for a single wind turbine.

Turbine detail report

The Turbine detail report shows three reports:

- Turbine basic information shows the real-time data including active production, wind speed, communication state. The Turbine detail report also shows the turbine type, identification number, installation date, region of installation, wind farm information, and working wind speed.
- Health Degradation report shows the information for health / failure risk and asset details for the major components of the turbine, blade, generator, and transformer.
- Real-Time Monitoring shows 11 real time readings for: Wind Speed, Angle-Blade1, Angle-Blade2, Angle-Blade3, Power At, Turbine Speed, Vibration-X, Vibration-Y, Wind Direction, Yaw Speed, Yaw Wind Direction.

Deputy Manager - Maintenance Dashboard

The Maintenance Dashboard for the Wind360 application gives the user an automatic and optimized maintenance plan that takes into consideration differing constraints and goals. The user can review the plan and do user interface customization as necessary.

Subscribing to the IBM Insights for Weather service

IBM IoT for Energy and Utilities uses widgets from The Weather Company. You need to subscribe to the IBM Insights for Weather to receive data for use with these widgets.

About this task

You need to be able to subscribe to Weather Company Data for IBM Bluemix and have the credentials for either a user for IoT for Energy and Utilities to be able to use the widgets for the Asset 360 for Wind application.

Procedure

1. Open the Blue Mix portal and subscribe to Weather Company Data for IBM Bluemix under Data and Analysis.

You receive the credential information as example:

```
{ "credentials": { "username":  
  "71b23b9c-65de-4c19-9e30-88a344a0bde8", "password": "MtTRzSLW6B", "host":  
  "twcservice.mybluemix.net", "port": 443, "url":  
  "https://71b23b9c-65de-4c19-9e30-88a344a0bde8:MtTRzSLW6B@twcservice.mybluemix.net"  
} }
```

2. Open the file for editing on the Liberty server: `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/lib/weather.properties`
3. Add the credentials to the file.

For example:

```
user=71b23b9c-65de-4c19-9e30-88a344a0bde8  
password=MtTRzSLW6B
```

4. Save and close the file.

Configuring for maintenance planning optimization analysis

You can prioritize the analysis of the maintenance to be carried out based on the lowest production loss, the resources you have at disposal, and for maintenance of the stable production of energy.

Configuring and creating a maintenance plan

Use the maintenance planner in IBM IoT for Energy and Utilities to analyze the best schedule for maintenance based on the options: lowest production loss, resource work balance, and stable production.

Procedure

1. Go to **Wind 360 > Maintenance planner** to open the **Maintenance planner** in IoT for Energy and Utilities.
2. Select one of the analysis options: `Lowest production loss` `Resource work balance` `Stable production`
3. The Maintenance planner calculates the best maintenance plan according to your maintenance criteria.

Administering the Asset 360 for Wind application

The Asset 360 for Wind application needs to be set up before use.

The IoT for Energy and Utilities on Cloud needs to be subscribed to the IBM Insights for Weather service, and to be able to use real time data, you must first regenerate the static data and then generate the real time data.

Performing simulator administration

Generating real time data for Wind 360

You must clean the application of existing data and generate static data from the IBM IoT for Energy and Utilities Asset 360 for wind application before you can generate real time data.

Before you begin

To be able to regenerate static data you must have **User** credentials.

Procedure

1. In IoT for Energy and Utilities, click **Wind 360 > Data simulator > Select All > Clear Static Data**

When the static data is cleared a message is returned:

```
clear:Turbine Power Statistic,Turbine Status Statistic,Turbine Score,Turbine
Status,Generator Risk,Rotor Risk,Transformer Risk,
Turbine Power,Turbine Wind Speed,
Turbine Wind And Power Statistic,
Windfarm Failure Recovery Statistic,Windfarm Score,
Windfarm Wind History,Maintenance Basic Data,
Turbine Maintenance
cost-done
```

2. Click **Regenerate static Data**.

When the static data is regenerated a message is returned:

```
generate:Turbine Power Statistic,Turbine Status Statistic,Turbine Score,Turbine
Status,Generator Risk,Rotor Risk,Transformer Risk,Turbine Power,
Turbine Wind Speed,Turbine Wind And Power Statistic,
Windfarm Failure Recovery Statistic,Windfarm Score,Windfarm
Wind History,Maintenance Basic Data,Turbine Maintenance
cost-done
```

3. Click the **Start** button to start the process of generating real time data.

A message reads: **Now the realtime data generation is running**.

4. Click the **Stop** button to end the process.

A message reads **Now the realtime data generation is stopped**.

Looking at a holistic view of the operations

IBM IoT for Energy and Utilities provides an analysis of the key performance indicators of the wind farm operations.

The analyzes give:

- A comparison between wind farms belonging to the same company.
- A view of the trend of the power generated.
- A view the mean time between failure and mean time to repair.
- A comparison of the maintenance costs compared to the previous year.
- A view of the utilization hours.

- The implications of weather on the operation.

Comparing one wind farm to the others in the same company

As a manager of a wind farm you can compare the key performance indicators of wind farms within the same company.

About this task

You must have management rights of access for IBM IoT for Energy and Utilities

Procedure

1. Click **Wind 360 > KPI**.
2. Select the **Country > Region > Wind farm** from the menu.
3. You can compare the key performance indicators by making a selection from the different wind farms.

View the trend of the power generated over time

In the Power Generation report you can see a comparison of the power production, lost power production and loss time in hours over a period of three weeks, this week and the two previous weeks.

About this task

Note: The trend of power generated over time is not linear. You need to consider for when the wind does not blow. You can do this by removing the out of service due to no wind hours and focus on the actual available hours of the wind turbine generator.

Procedure

1. Click **Wind 360 > KPI**.
2. View the **Power Generation** report for the visualization.

Viewing repair and restoration statistics

In the MTBF and MTTR report you can see the comparison between the mean time between failure and mean time to repair over a defined period.

About this task

You define the time period that the report shows.

Procedure

1. Click **Wind 360 > KPI**.
2. View the **MTBF & MTTR** report.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Viewing the maintenance costs

The Maintenance costs report shows the maintenance cost of the turbines in the wind farm during the same time periods for the current year and previous year.

About this task

You can use the calendar to select the period you require.

Procedure

1. Click **Wind 360 > KPI**.
2. View the **Maintenance Cost** report for the visualization.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Viewing the utilization hours

The Utilization Hours report shows the power generation and power loss in MW and utilization hours during one time period.

About this task

One utilization hour = actual power generation in MW / turbine power generation at full capacity. It can be the most important KPI for a wind farm.

You can select the time period that the report shows.

Procedure

1. Click **Wind 360 > KPI**.
2. View the **Utilization Hours** report.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Seeing implications of weather on the operation

In this report you get the current weather information: temperature and sun condition, wind speed and direction, and humidity and also wind speed and direction for the next 12 hours.

About this task

You are also provided with maintenance suggestions according to the weather condition.

Procedure

1. Click **Wind 360 > KPI**.
2. View the **Weather Forecast** report.

Monitoring detailed operations

As an operations engineer you can monitor both the wind farm, and more detailed turbine operations.

The reports show:

- The status of a turbine
- The details condition of a turbine
- The wind speed comparison to power generation
- The wind speed and direction trend over time
- The utilization hours

Monitoring status of a turbine

As an operations engineer, you can monitor the status of each wind turbine in a wind farm.

About this task

You need operations engineer access rights for IBM IoT for Energy and Utilities.

Procedure

1. Click **Wind 360 > Monitoring**.
2. In the map view, hover over the wind farm icon and click **View Details**.
3. You can hover over each of the wind turbines to view the status of each one.

The information that shows is:

- Serial number of the turbine
- Working status
- Index score
- Wind speed

Viewing the detailed condition of a wind turbine

As the operations engineer you can view the detail condition of each wind turbine.

About this task

You can view the health degradation of the blade, generator and transmission components of the turbine. Each gives information about the asset and the health and failure risk of the asset over time.

The asset information is:

- Asset name
- Asset serial number
- Manufacturer
- Working years
- Installation date
- Wind farm

Procedure

1. Click **Wind 360 > Monitoring**.
2. In the map view, hover over the wind farm icon and click **View Details**.
3. Click the wind turbine that you need to see the details.
4. Click **Details** in the turbine overview report.
5. Click **BladeGeneratorTransmission** to view the **Asset information** and **Health/ failure risk** reports for that asset.

Comparing the wind speed and power generation

The details of wind speed and power generation are shown in the **Real-time** data report.

About this task

Procedure

1. Click **Wind 360 > Monitoring**.
2. In the map view, hover over the wind farm icon and click **View Details**.

3. View the **Real-time data** report for **Active Production** and **Wind Speed**.

Viewing wind speed and direction trends

About this task

In the **Wind Speed/Direction Trend** report you can view the wind condition of the farm. You can also hover on it to see the specific number of these days. The deep blue part of the report represents the past days and the light blue part represents the forecast in the near future days.

Procedure

1. Click **Wind 360 > Monitoring**.
2. View the **Wind Speed/Direction Trend** report for the current data, past data and future trend.
3. Hover over each of the days that show to view the wind speed and direction for that day.

Viewing utilization hours

The Utilization Hours report shows the power generation and power loss in MW and utilization hours during one time period.

About this task

One utilization hour = actual power generation in MW / turbine power generation at full capacity. It can be the most important KPI for a wind farm.

You can select the time period that the report shows.

Procedure

1. Click **Wind 360 > Monitoring**.
2. View the **Utilization Hours** report.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 actual people or business enterprises 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.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

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.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's [Privacy Policy](http://www.ibm.com/privacy) at <http://www.ibm.com/privacy> and IBM's [Online Privacy Statement](http://www.ibm.com/privacy/details) at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies" and the "[IBM Software Products and Software-as-a-Service Privacy Statement](http://www.ibm.com/software/info/product-privacy)" at <http://www.ibm.com/software/info/product-privacy>.

