Course Guide

IBM Operations Analytics Predictive Insights 1.3.3: Configuration and Implementation

Course code TN612 ERC 1.0

IBM Training

March 2016 edition

NOTICES

This information was developed for products and services offered in the USA.

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
United States of America

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

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

Any references in this information to non-IBM 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 supply in any way it believes appropriate without incurring any obligation to you.

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

TRADEMARKS

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

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

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

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

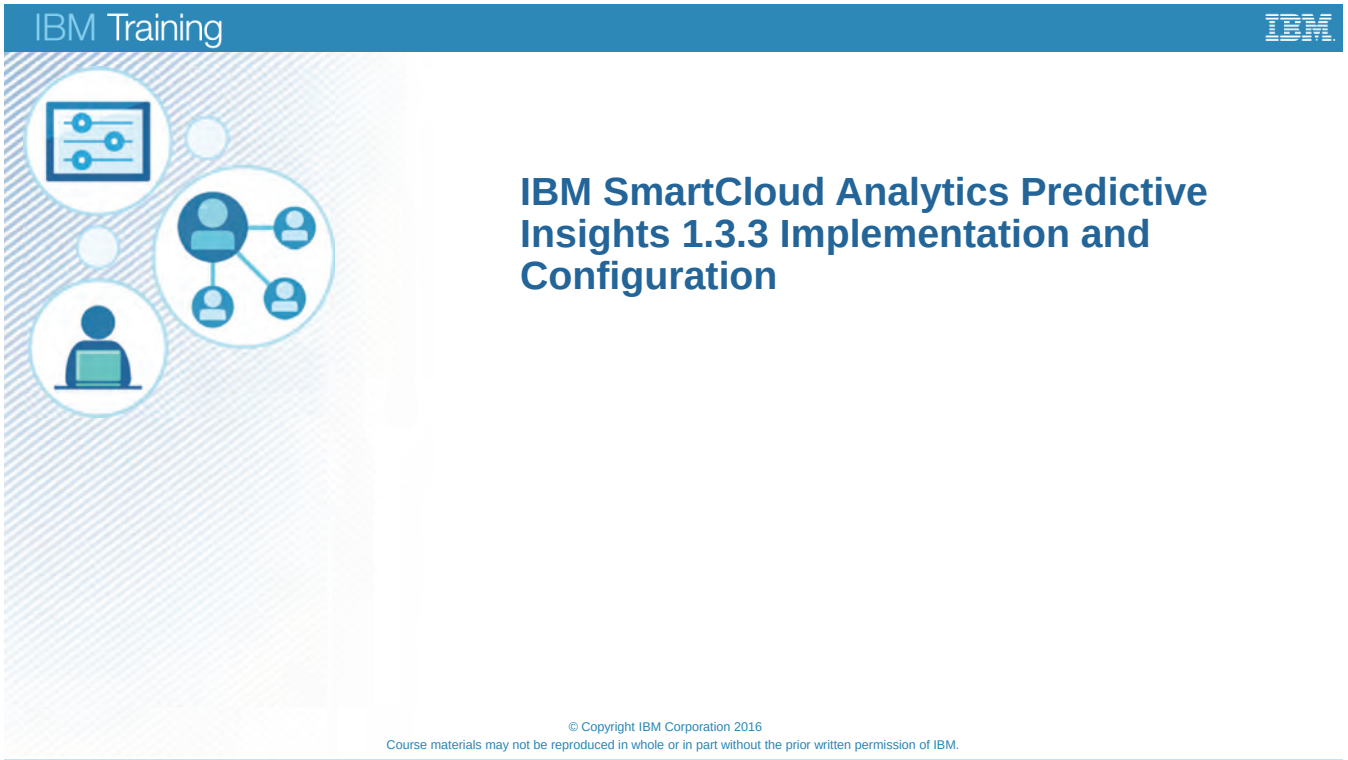
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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.

Contents

About this course



This two and one-half day class provides a complete overview of the Predictive Insights product. It introduces analytics and how it is used to create automated thresholds, and draws relationships between key performance indicators (KPIs). You learn to select useful KPIs from their monitored data and develop data models that are fed into the analysis engine. You also learn how to install and configure the analysis engine to account for different monitoring environments, and how to use the interface and interpret various alarms.

This training provides and overview of how analytics is used to monitor and alarm on services. It then describe the architecture and installation process of the Predictive Insights solutions. You then learn about how the solution collects monitoring data from various data sources and then use a mediation tool to select and filter the data inside these data sources. You learn how to configure the server and then analyze data to search for anomalies that can lead to service disruptions. Students complete exercises to install the software and use the mediation tool to connect to and manipulate data sources. They then configure the server and analyze these data sources and review the anomalies generated. You learn how to use advanced mediation techniques using the third-party tool logstash.

The lab environment for this course uses the RHEL 6.5 platform.

For information about other related courses, visit the Cloud & Smarter Infrastructure education training paths website:

ibm.com/software/software/tivoli/education/

	Details
Delivery method	Classroom or instructor-led online (ILO)
Course level	ERC 1.0
	This course is an update to the TN611 course. It includes new materials for advanced mediation
Product and version	IBM Operations Analytics Predictive Insights 1.3.3
Duration	2.5 days
Skill level	Intermediate

About the student

This course is designed for an intermediate user who has experience with both monitoring tools (for example, IBM® Tivoli® Monitoring) and the Linux operating system.

Before taking this course, make sure that you can use basic Linux operating system commands such as **cd**, **more**, **tail**, **ls**, as well as how to include JAR files to a class path for Java applications to find new or modified code.

Learning objectives

IBM Training

Learning objectives

In this course, you learn to perform the following tasks:

- Describe how analytics improves problem detection
- Install the Predictive Insights software
- Describe and mediate data sources that provide the basis of the analytics model
- Configure the server to perform an analysis
- Analyze and anomalies and review their details
- Learn techniques to help mediate complex data

The objectives of each unit are as follows:

- Unit 1 Overview
  - Describe how analytics improves problem detection
  - Describe how Predictive Insights learns about system performance
  - Describe when Predictive Insights alarms on anomalies
- Unit 2 Architecture
  - Describe the server layout options
  - Describe the roles of each server in the implementation
  - Calculate the required resources for implementation
- Unit 3 Installation
  - Describe the installation process
  - Describe potential problems with installation
  - Validate that installation is successful
- Unit 4 Data Modeling
  - Select useful key performance indicators
  - Connect to external data sources

  - Model data sources with client mediation utility
  - Deploy data model
- Unit 5 Configuring and Operating the Server
  - Configure the server for important properties
  - Start and monitor the analysis of your data
  - Enrich alarms
  - Suppress various alarms
  - Interact with alarms inside the GUI
  - Describe the data that is associated with an alarm
  - Search for various anomalies and metrics
- Unit 6 Administration
  - Address problems with installation
  - Fix common problems with mediation tool
  - Remediate startup issues with analytics server
  - Troubleshoot issues with extraction
  - Debug problems with alarms
- Unit 7 Advanced Mediation
  - Describe the third-party tool logstash
  - Build logstash conf files for mediating data
  - Describe mediation considerations

Course agenda

The course contains the following units:

1. Predictive Insights overview

   This unit provides an overview of the Predictive Insights solution and how you use it to detect problems in your infrastructure. You are introduced to the statistical models that you use for the monitoring data that you collect about your applications and servers. Using this model, you determine normal operations and then use alarms to alert you when your infrastructure deviates from normal operations.

2. Architectural information

   This unit is about the physical and logical server layout that is needed to implement Predictive Insights. You discover the role of each piece of the architecture. You also learn how to segment data to support large implementation and the resources necessary to support them.

3. Installation

   This unit provides an overview of the prerequisites and installation process for Predictive Insights. It also presents an optional process to validate that the installation was successful.

   In the following exercises, you install InfoSphere® Streams and IBM Operations Analytics Predictive Insights on the DB2® OMNIbus, and Web GUI software that is installed and running on your virtual machine. This installation is stand-alone with most software installed under one user.

4. Data sources and modeling

   This unit provides information about how data is collected and modeled with Predictive Insights. You learn about key performance indicators and how they are selected from the monitoring data that is available. You learn about the supported data sources and how to connect to them, manipulate the data sources, and select only the items that should be analyzed.

   In the following exercises, you learn about the three data sources that are available in this training. One source is a set of comma-separated value (CSV) files, and the other two are data that is in the DB2 and PostgreSQL databases. You connect to these data sources and model the data within them by using the data mediation tool that you installed earlier. You use the mediation tool to select appropriate KPIs and filter unnecessary resources from the data. You then deploy the model to the Predictive Insights server. As an optional exercise, you configure the mediation client and server to use the data that is in the PostgreSQL database.

5. Configuring and operating the server

   This unit describes how to configure the most important aspects of an analytics server. You learn how to start the server and analysis, and ensure that both are operating correctly. You also learn how to suppress alarms and work with the data that is generated when an alarm occurs.

   In this set of exercises, you configure the server for important attributes that determine the aggregation interval that it uses for data extraction and how many weeks it is used for training.

You also start the analysis of the historical data that was modeled in Unit 4, Exercise 11 on page 62. You check various logs and file systems to see the progress of the data extraction. Finally, you review the alarms that are generated by Predictive Insights.

6. Administration and troubleshooting

   This unit provides information about common administrative tasks and troubleshooting techniques.

7. Advanced mediation techniques

   This unit describes the tools and techniques to be used to mediate data that is not in the correct format for consumption by Predictive Insights.

   In this set of exercises, your introduction to logstash includes building a simple configuration file to work with a log file that is a combination of useful metric data and log messages that must be removed. For testing purposes, you stream messages to standard input, parse and modify each message, and then sending the results to standard output. Because you are using standard input, you use the head command and pipe the data stream into logstash with a command similar to this one:

Unit 1 Predictive Insights overview

This unit provides an overview of the Predictive Insights solution and how you use it to detect problems in your infrastructure. You are introduced to the statistical models that you use for the monitoring data that you collect about your applications and servers. Using this model, you determine normal operations and then use alarms to alert you when your infrastructure deviates from normal operations.

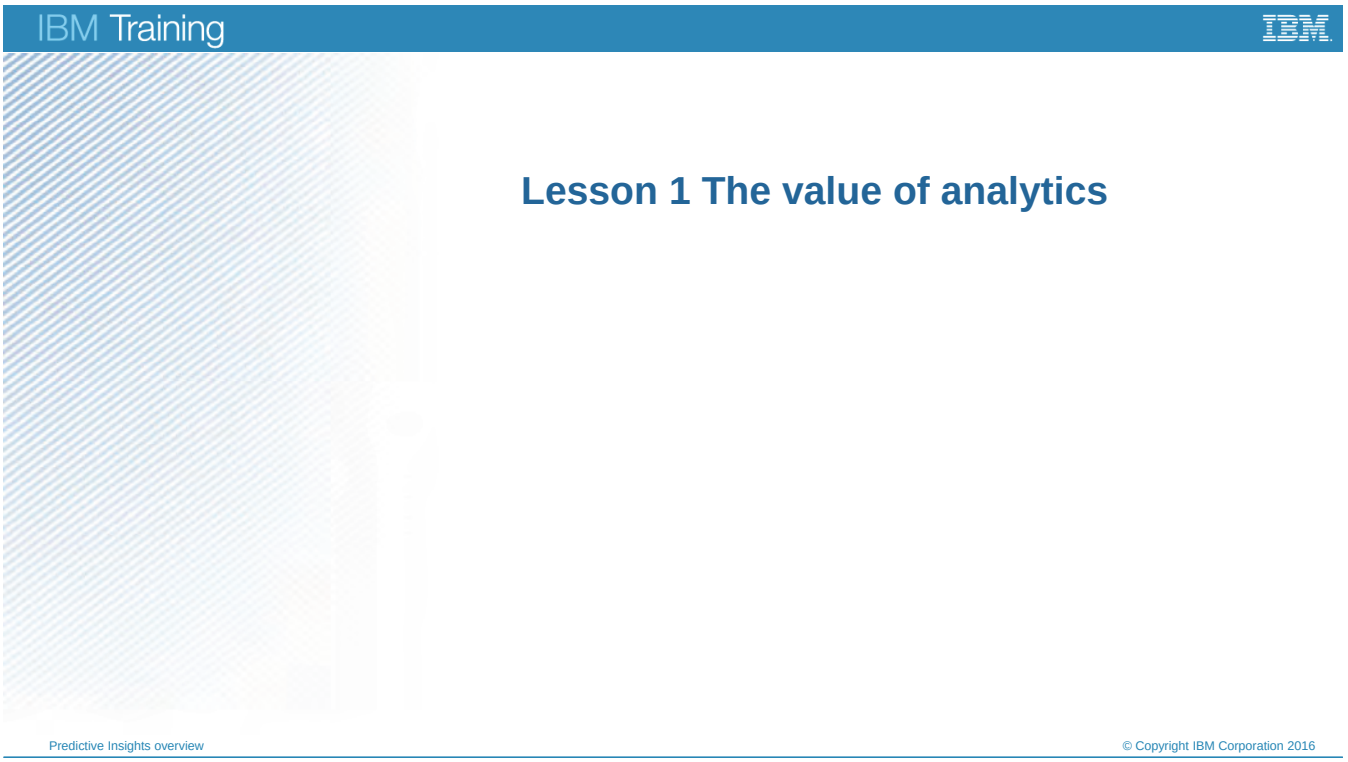
IBM Training

Unit objectives

- Explain the value of the analytics
- Describe Predictive Insights and its features
- Explain alarms on anomalies
- Describe relationships between performance indicators
- Describe detectors
- Controlling the number of alarms

Unit objectives

Lesson 1 The value of analytics



In this lesson, you learn about automated threshold analysis and model development.

IBM Training

Need earlier warnings of impending problems

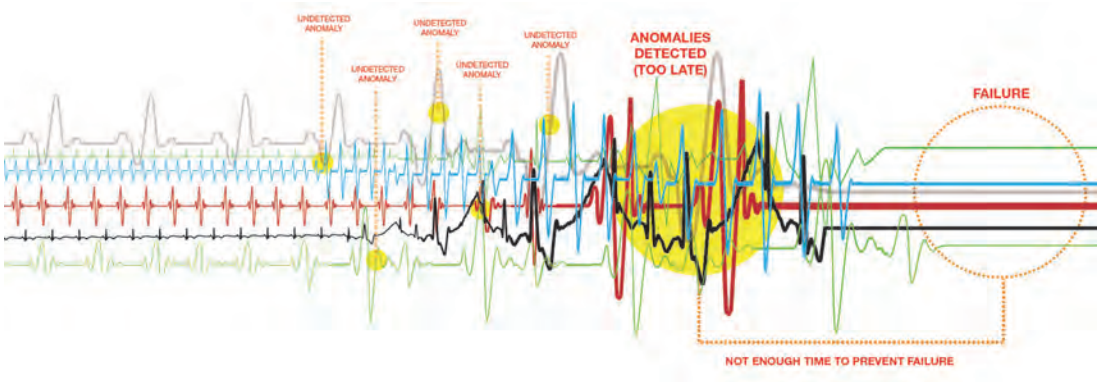
- Consider an approaching thunderstorm
  - Clouds get big
  - Sky gets dark
- Services have same early indicators
  - Memory usage not acting normal
  - Response time is dropping

Need earlier warnings of impending problems

As in any endeavor, it is important to know when problems are coming. Often times, the warnings are obvious. Consider an approaching thunderstorm. If you are outside, you are rarely surprised by a thunderstorm because of the early warnings you can see. Clouds rolling in and the darkening of the sky are just a couple of the obvious signs that a storm is coming. Services that are important to an organization also have early indicators that problems are on the horizon. Memory usage is not normal and response times are dropping can be those indicators. The biggest problem is seeing these indicators in the sea of information that inundates operations.

Current challenges for operations

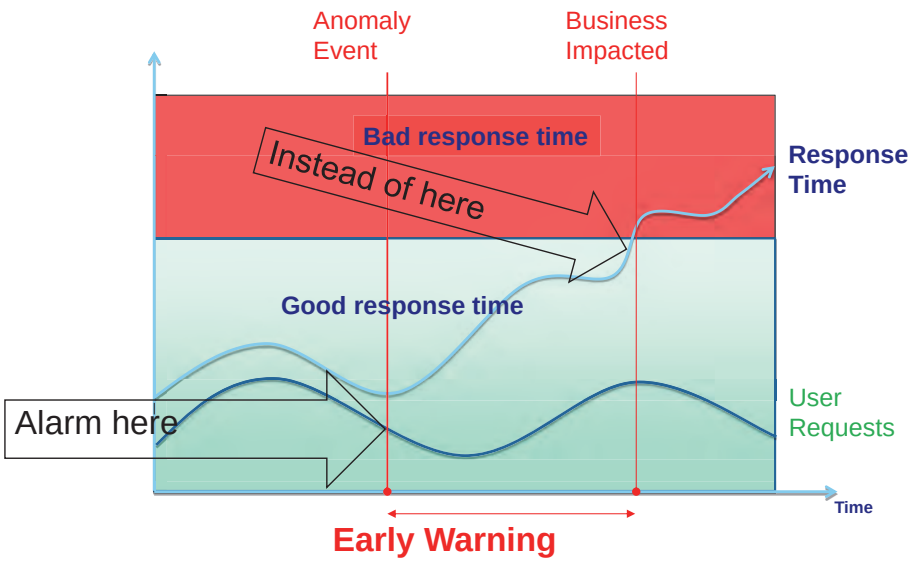
- Too much data to evaluate manually
- Standard thresholds for monitoring processes are not sufficient
  - Too high, insufficient warning
  - Too low, too much noise
- Problems are difficult to detect while they emerge



Current challenges for operations

The challenge for operators is the quantity of data and the complexity of the solutions. Typical monitoring systems rely on static thresholds. With manual thresholds set high, you do not get sufficient warning to address a problem before it impacts service. If you set thresholds too low and receive errant alarms, the warnings might be ignored. You can use the data that is generated by your infrastructure; however, quantity of data is large. With large quantities of data, the notifications might come too late and service disruptions can occur.

Operations needs early warning too!



Operations needs early warning too!

If you want earlier warnings before service problems impact the business, you can no longer set static thresholds on important performance indicators. To get earlier warnings, you need to detect when things are straying away from normal operations. For example, response time should increase and decrease in relation to the number of user requests. However, when response time no longer abides by that trend, operators can get an alarm that can give them time to respond before business is affected.

IBM Training

Analytics to the rescue

- **Gather a statistically significant amount of time-series data of key performance indicators**
  - Time series data comes in a lot of forms
- **Then use analytics to build a model that performs these tasks:**
  - Automates the threshold analysis of each key performance indicator
  - Determines if relationships exist between key performance indicators
  - Uses heuristics (rules of thumb) to alarm operators when something isn't right

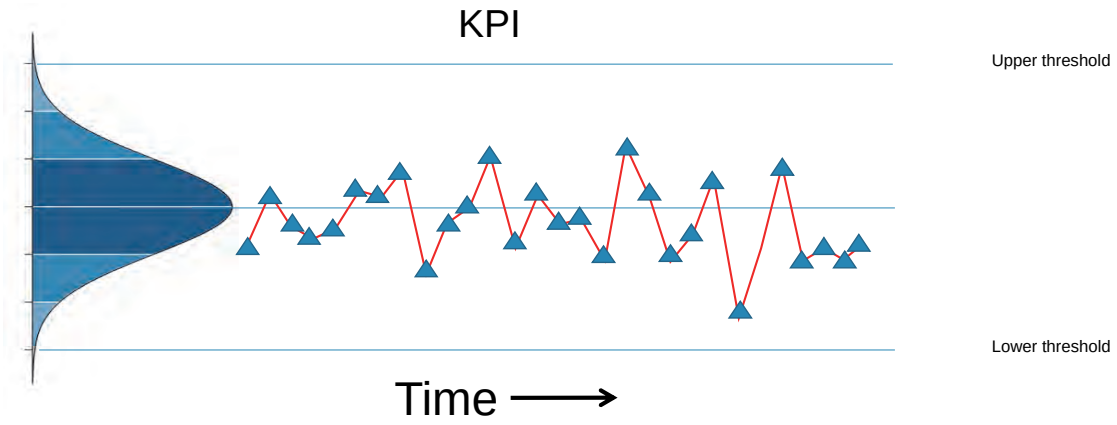
Analytics to the rescue

This earlier warning is possible by using analytics. If you gather enough historical data on your performance indicators, a valid statistical analysis can be completed on all these indicators to build a model of how your service performs. It is important to remember that time series data comes in many forms: IT, smarter buildings, manufacturing, and so on. You can then use this model to generate thresholds that better reflect the reality of your system performance. These thresholds provide a tighter tolerance on performance indicators and give better warning than using industry standards. These statistical models can also be used to determine relationships between performance indicators. When you have these thresholds and relationships, heuristics can be developed to warn operators when performance stays outside of these expectations.

IBM Training

Automated threshold analysis

- Each key performance indicator (KPI) is analyzed for its normal operating envelope
- Thresholds are developed that represent the volatility of the data



Automated threshold analysis

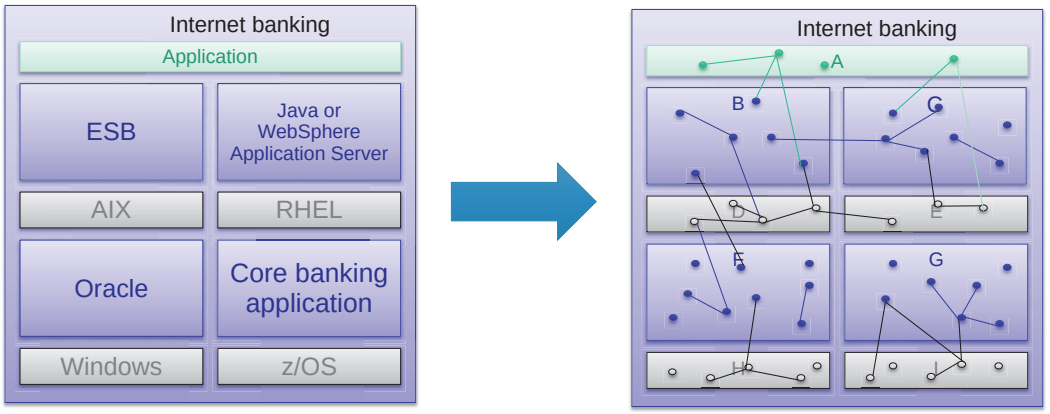
With a database of monitoring data, you can build models that describe what typical thresholds should be for parameters that are monitored. By seeking these normal operating thresholds for key processing indicators, you create thresholds that are neither too large or too small and are meaningful when performance creeps outside these thresholds. This process is called automated threshold analysis.

IBM Training

Developing a relationship model

You can use statistical models to discover mathematical relationships between KPIs

- Multivariate analysis of one KPI can predict the behavior of others
- KPIs are grouped together by the way they work together



Developing a relationship model

Even though statistics is a science that has existed for hundreds of years, it continues to evolve and expand its usefulness in finding relationships in data. Newer statistical models can find mathematical relationships between key performance indicators that exist on separate servers. By doing multivariate analysis across the entire system of servers and their monitoring data, you can derive a relationship model. You get information about a KPI that is anomalous, and also learn about key performance indicators (KPIs) that might have an impact on each other. The server also determines whether two or more KPIs can be grouped because of their historical tendency to have synchronized movement. If this relationship breaks down, then the server creates an alarm to inform you of this change. Because the analysis of interrelationships between KPIs is comprehensive, you can glean information that might not be noticed.

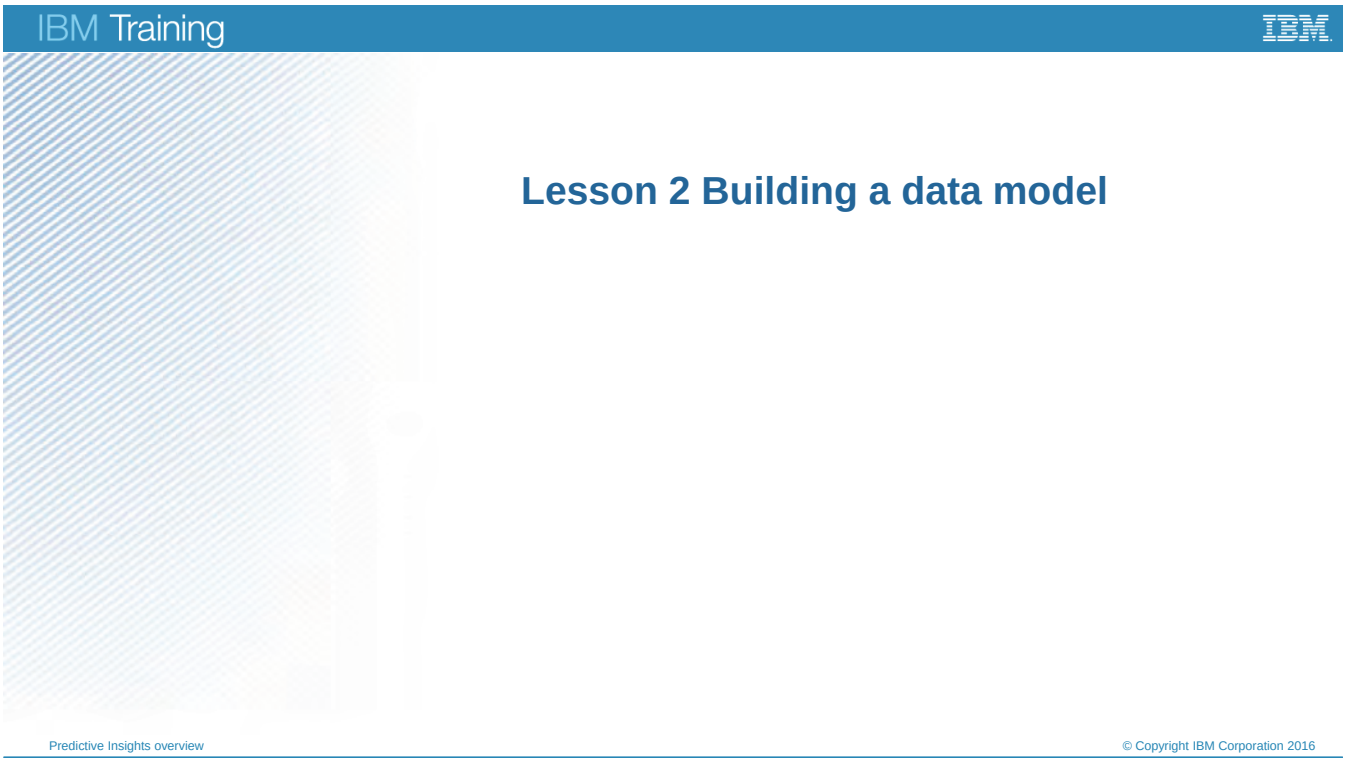
Predictive Insights

- Learns about normal operational behavior across the infrastructure, including how metrics work together
- Adjusts thresholds during regular, day-to-day variations
- Identifies anomalies before they affect service
- Develops relationship models, based on the data
- Assists with root cause analysis by indicating the most offending metrics
- Provides integrations with OMNIbus

Predictive Insights

You can use Predictive Insights to find the normal operating behavior of your infrastructure. Using this behavior, you can set thresholds for all the KPIs that are monitored and identify their volatility. With this statistical model of your infrastructure, you can develop service models and create relationships without knowing all the possible connections and interrelationships. Through models, you can perform root cause analysis of the problems that can be challenging to understand.

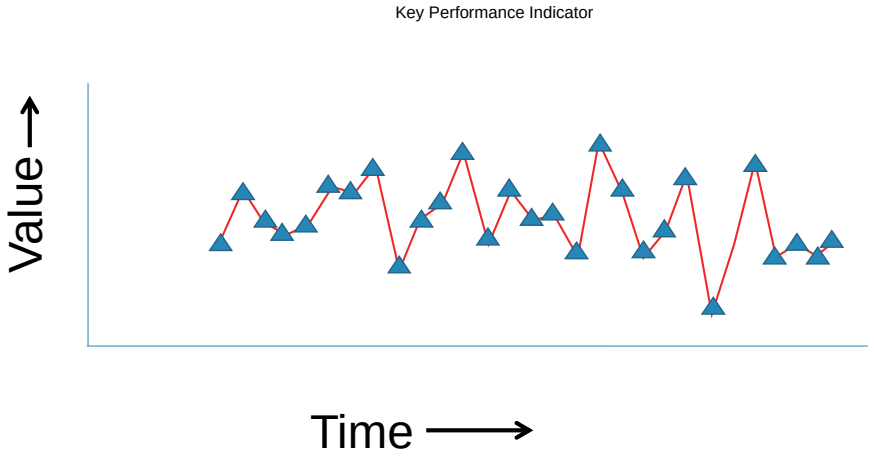
Lesson 2 Building a data model



This lesson is about the techniques used by Predictive Insights to build a data model and find anomalies.

IBM Training

Using time-based variable data



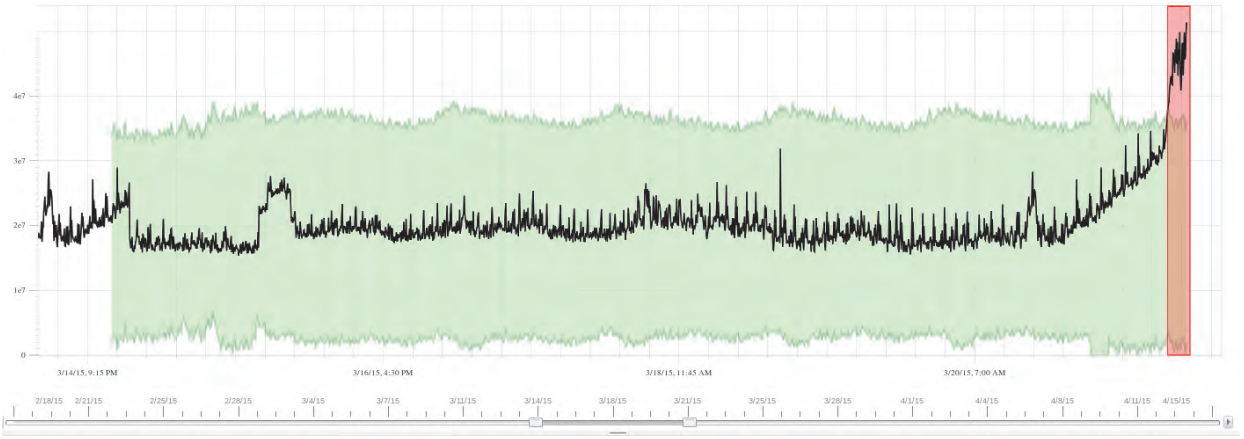
Using time-based variable data

With Predictive Insights, you look at data that varies over time. A Key Performance Indicator, or KPI, is a server-based parameter that varies every minute, or at the least, every hour. This variation is captured over time by the monitoring tools a company uses.

IBM Training

Thresholds

- Thresholds define upper and lower control limits on acceptable behavior
- Predictive Insights calculates thresholds for each KPI



Thresholds

For every KPI in the data model, the system generates thresholds. These control limits are statistically calculated. Data scientists have invested a large effort in setting appropriate thresholds, much of which is not revealed to users. Manipulation of these behind-the-scene efforts is beyond the scope of this training and is in fact beyond undergraduate level statistics classes.

Seeing patterns in KPI data

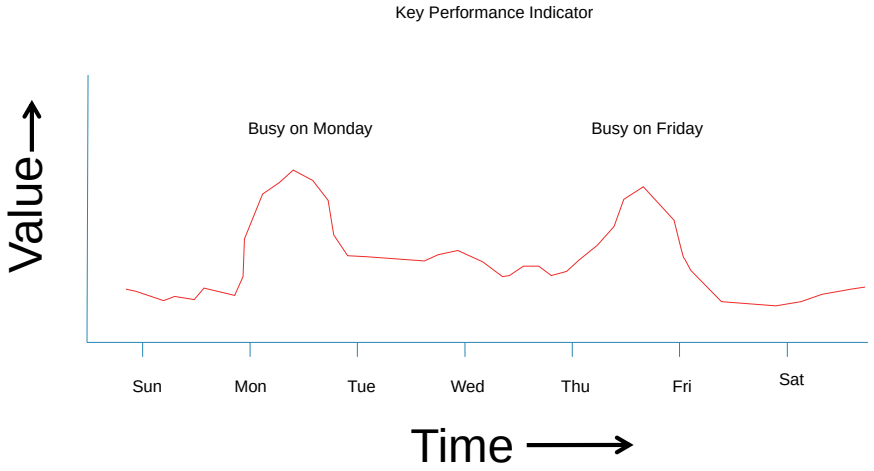
- Predictive Insights can identify patterns in time series data
  Uses algorithms to determine if data is seasonal
- Predictive Insights looks at workdays and weekends
  Web servers are busy during the week but quiet on weekends
- Predictive Insights has automated analysis of each KP

Seeing patterns in KPI data

Oftentimes, thresholds are seen as a static set of boundaries that do not vary with time. By using Predictive Insights, you can look for patterns in the data that has a regular variation with time. With these algorithms, you can discover whether the data has seasonal variations. In Predictive Insights, you can see patterns that occur every seven days. This analysis happens automatically and requires no user intervention. Every time the server analyzes a new set of data, it checks to see whether data includes enough information to be seasonal. Data can lose its seasonal characteristics and become nonseasonal or static.

IBM Training

Seasonality

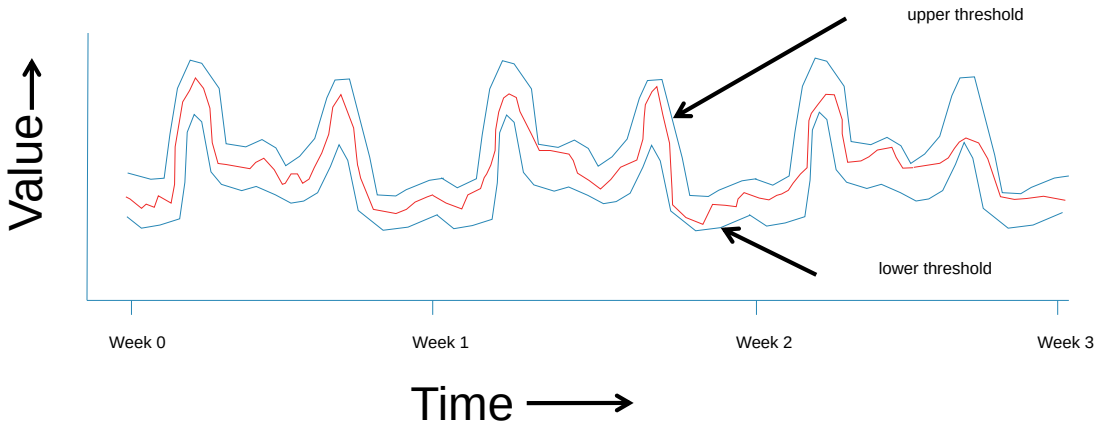


Seasonality

As you look through the performance data for one week, you can see that the server is busy on Monday and Friday. This variation in 2 - 4 weeks of training data is noted, and the KPI becomes seasonal.

IBM Training

Seasonal thresholds



Seasonal thresholds

With seasonality comes thresholds that vary over time. These seasonal thresholds provide a tighter tolerance for data and allows anomalies to be seen faster than traditional static boundaries. Because Predictive Insights has a seven-day seasonality, you can see this repeated pattern of variation when you look at multiple weeks of data.

Anomaly detectors

- Predictive Insights uses a series of detectors to search for anomalous data
  - Does data stray outside of thresholds?
  - Does the data no longer vary as much as it did previously?
  - Has data become flat-lined?
  - Are historical relationships breaking down?
- Predictive Insights notes when any of these things happen

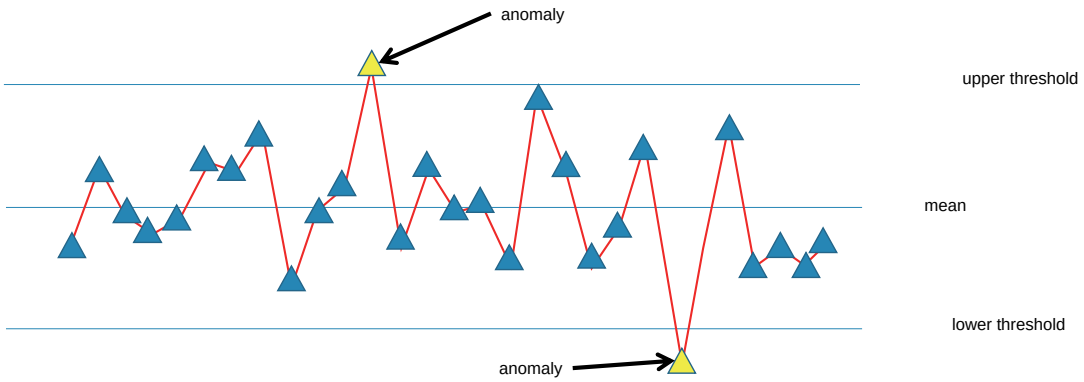
Anomaly detectors

Predictive Insights uses the concept of detectors to find anomalous data point. Each detector has a specific role in looking for anomalous behavior. One detector looks for data that strays outside of the thresholds that have been defined for each KPI. Other detectors look at how much the data is varying. For example, when the server was learning about the data model, it saw a lot variation in the data. But now that it is actively comparing data to the model, it no longer sees the same amount of variation. Along those same lines, a similar detector that notes when data has flat lined and is no longer varying at all. There are other detectors that compare KPIs to each other and determines if there are relationships in the data. One of these detectors notes when a relationship is breaking down. Whenever one of these detectors sees an anomaly, it notes it in the database.

IBM Training

Anomaly example

- Anomalies are generated when a data point is outside its thresholds
- A note is generated in the Predictive Insights database about anomaly



Predictive Insights overview 18 © Copyright IBM Corporation 2016

Anomaly example

After you define upper and lower thresholds (regardless of seasonality), you use these thresholds to determine when you get anomalous data points. An anomaly is any data point that falls outside your upper and lower thresholds. When a data point falls outside these boundaries, it is noted by the server. A single point of data outside a threshold does not yield an alarm.

IBM Training

Anomalies and alarms

- Alarms cannot be generated for every anomaly
- Predictive Insight sets thresholds for each KPI
  - 99.7% of data is usually between thresholds (0.3% of data is anomalous)
- A system with 80,000 KPIs sampled every 5 minutes leads to ~23,000,000 daily observations
  - Statistically, 69,000 anomalies occur per day
  - Alarms on every anomaly is unreasonable
- Ignore individual spikes and look for trends
  - N number of anomalies within M aggregation intervals

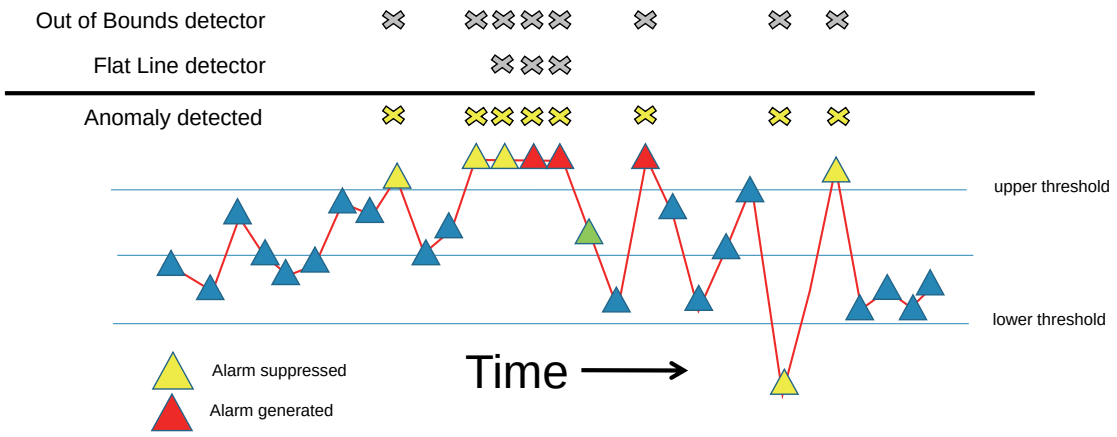
Anomalies and alarms

With the thresholds that are created by Predictive Insights, only about 0.3% of the data should ever fall outside its boundaries. However, with the total number of measurements that are made daily, even 0.3% can lead to many alarms if one was generated for every anomaly. With only 80,000 KPIs being sampled on a 5-minute basis (large companies can have upwards of a million KPIs), it might easily create 69,000 anomalies per day. The analysis of Predictive Insights ignores individual spikes in data and instead looks for trends in data toward being anomalous. An example of this type of trend is creating an alarm when only multiple anomalies are seen over time. This heuristic is called the N of M rule. The server must see N number of anomalies on a KPI inside of M aggregation intervals to generate an alarm.

IBM Training

Alarm

An alarm is generated when one or more detectors sees an anomaly in 3 of the last 6 aggregation periods



Alarm

In Predictive Insights, this N of M rule is expressed, by default, as three anomalies within six aggregation intervals. An anomaly is counted as one or more detectors sees anomalous data. As you can see in the graphic, both the Out of Bounds detector and the Flat Line detector are being tripped at the same time. However, this situation expresses itself only as one anomaly for the sake of the N of M rule. The first alarm is generated only after the third anomaly. Subsequent alarms are generated if the data stays outside the thresholds and the 3-of-6 six rule is met. This heuristic is only one of many that determine when an alarm should be generated and is only presented here as an example.

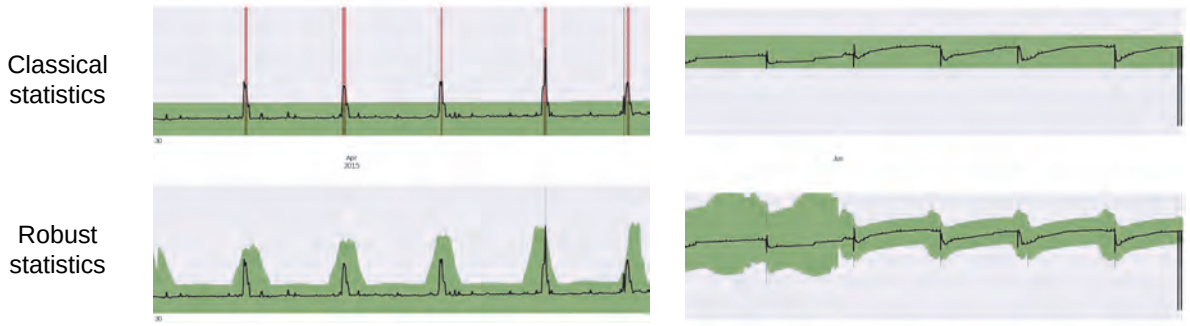
Lesson 3 Detectors

Predictive Insights uses detectors to locate anomalous behavior. This lesson reviews the detectors currently in use by the software.

IBM Training

Detector #1: Robust Bounds

- Defines the green thresholds for each KPI using robust statistics
- A data-driven approach that does not rely on idealized statistical distribution
  - Seeks to outperform classical statistical methods in the presence of outliers (anomalies)
- Thresholds adjust much more quickly to behaviors
  - Better seasonality (static thresholds mostly gone)
  - Thresholds tighten as behaviors become more apparent



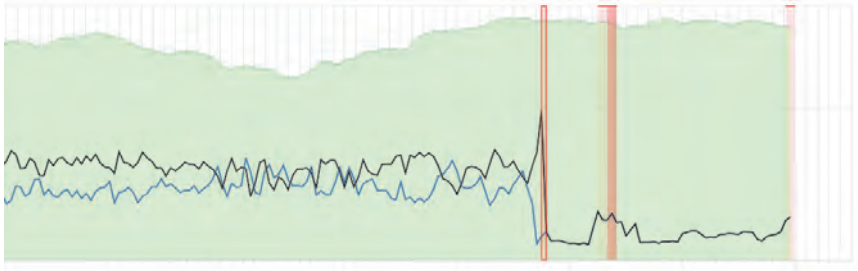
Detector #1: Robust Bounds

The Robust Bounds detector is what defines the upper and lower thresholds for a KPI. This green fairway is where you should expect you data to reside most of the time. If data falls outside this fairway, its considered anomalous. Robust statistics is different from the classical normal distributions that you might have learned in high school or college. It is designed to handle outlying data more effectively that classical statistics. As you can see in the slide, the robust statistics makes better use of seasonality and can quickly tighten thresholds as the behavior of the data becomes apparent.

IBM Training

Detector #2: Granger

- Granger looks for correlation between metrics
- Correlation needs to show a delay between an input and an output
- Detects when KPI relationships breakdown
- Requires only 3 days of data to find relationships



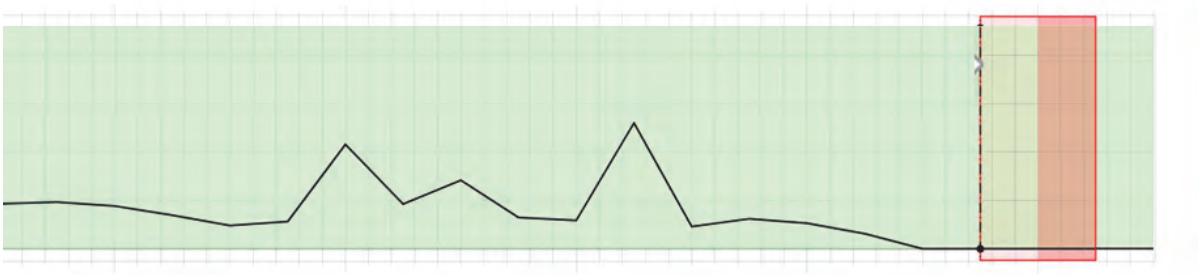
Detector #2: Granger

The Granger detector is designed to look at the relationships between KPIs. Using this analysis, it can determine if a KPI tends to predict the outcome of another KPI. For Granger to work properly, there needs to be a delay between the action of one KPI and the reaction of another KPI. Once the Granger analysis finds these relationships, it alarms when it sees the relationship breaks down. Unlike other detectors, the Granger analysis needs only a few days of data to build its model.

IBM Training

Detector #3: Flat line

Detects KPIs that have not gone flat during training, but do so during scoring



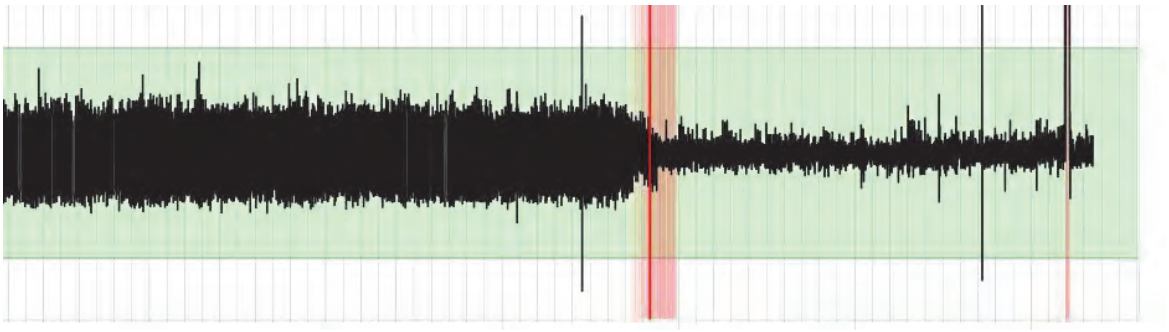
Detector #3: Flat line

The flat line detector looks for data that has stopped varying. If the detector trained on data that had no flat lines, then it will send an alarm once its sees the data no longer varying. The detector looks only for data that does not vary, so whether it's flat-lined at zero or a different value, an alarm is still generated.

IBM Training

Detector #4: Variant - Invariant

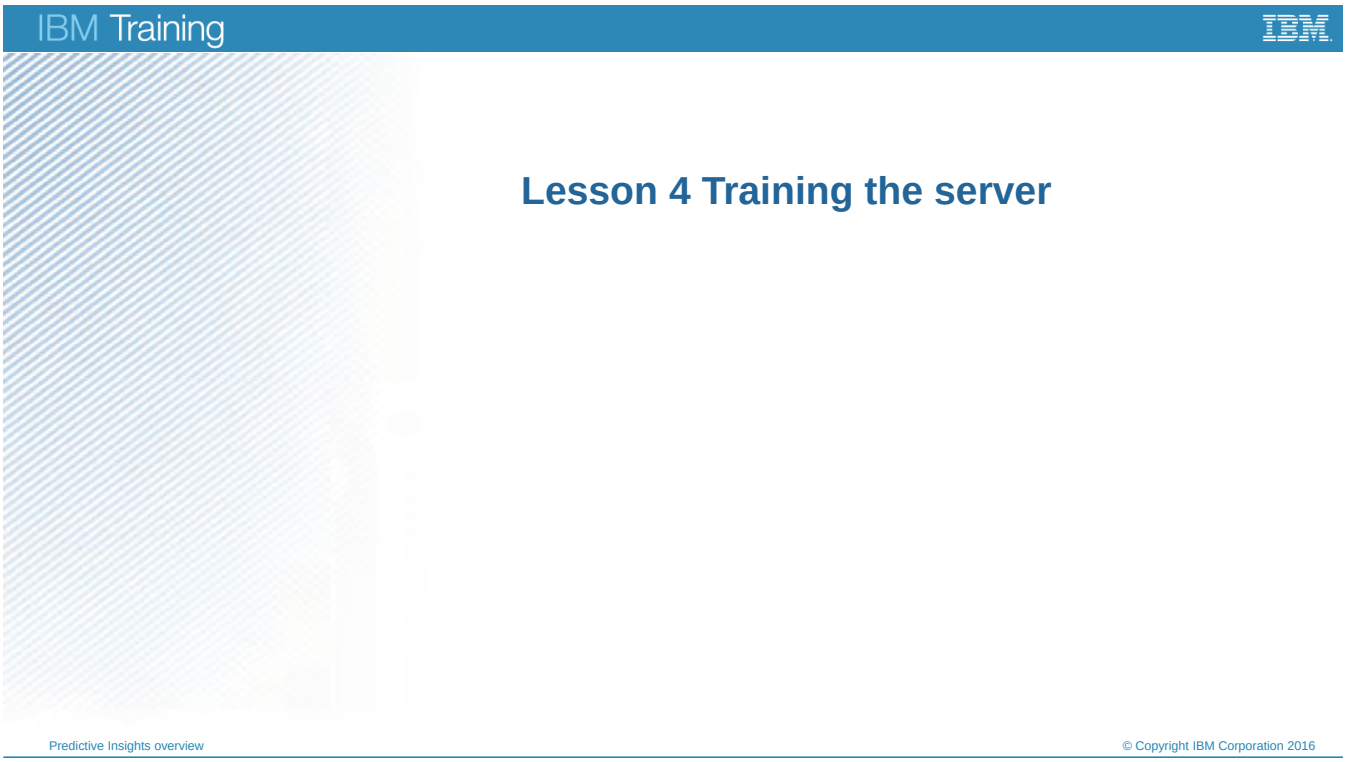
Detect KPIs which have been varying during training, but are significantly less so during scoring



Detector #4: Variant - Invariant

The variant/invariant detector alarms when the data it was training on had much variation but once it starts scoring data, it does not see the same wide variation. This alarm provides users with a heads up that something changed significantly but would not be picked up by the robust bounds detector.

Lesson 4 Training the server



To use detectors, you must load the system with a minimum amount of data that can be used to teach them what normal behavior is. This process of data collection is referred to as training the server. This lesson provides some background about how you train the Predictive Insights server.

IBM Training

Building the statistical model

- You must train a server to detect normal operations
  Assumes system is operating within expected tolerances most of the time
- Each detector is designed to have a minimum amount of data before its operational
  - Ranges from three days to two weeks based on detector
  - Cannot be changed
- You can use historical data, if available

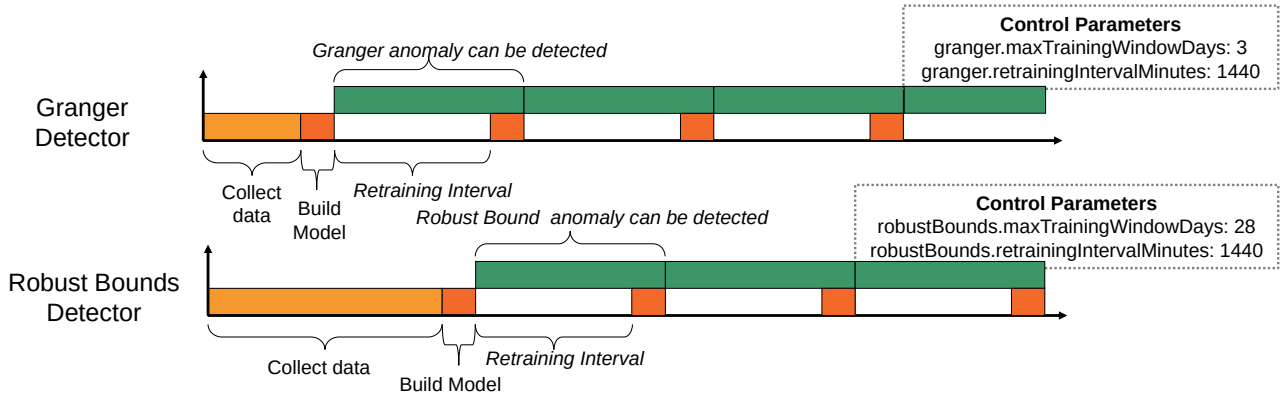
Building the statistical model

To create these statistical models for each KPI, you must train the server to identify a normal operation. Typically, a system of servers is running often enough that operations personnel consider it to be normal. To train the server, each detector needs a minimum amount of performance data to get a statistically valid analysis. If you have enough historical data in the data sources, you can quickly get the server trained.

IBM Training

Detectors rebuild their model often

- System collects minimum historical or current data for each detector
- Detectors come online once minimum data is collected
  - Detector scoring (searching for anomalies) begins
  - Detector rebuild models often using newest data
  - Detectors add data to their model to reach a maximum data amount



Detectors rebuild their model often

The detectors retrain themselves often to make use of the latest data collected. Each detector has different requirements for their statistical model to be valid. Once the detector collects the minimum amount of data necessary, it builds its model and comes online. Once online, all new data is scored against that model and anomalies can be detected. By default all detectors retrain themselves after 24 hours. After the detector is online, the new data received can be added to the model to provide a larger statistical sample to improve the accuracy of the model. New data is added to the model until a maximum training window is reached. After that, the oldest data is trimmed from the model.

IBM Training

Managing anomalies during training

- Four weeks of data generates over 8000 data points (5-minute data) per KPI
- Occasional anomalies have little effect on mean and standard deviation
- An example is adding 10 gallons of water into an average pool
  The level of the pool rises, but you do not see it

Managing anomalies during training

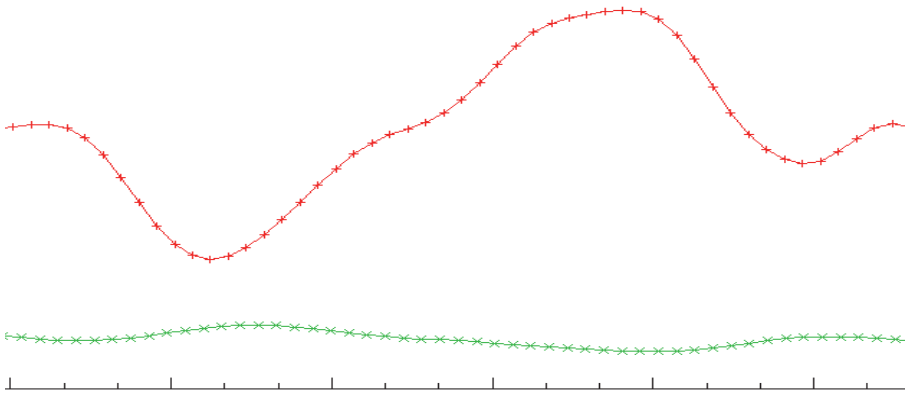
Normal operation does not occur 100% of the time. What you seek to analyze is a system of servers that are normal most of the time. If you use a four-week training period with an aggregation interval of 5 minutes, you generate over 8000 data points for each monitored KPI. If most of that data is normal, any spikes in it get averaged out. For example, if a pool contains over 10,000 gallons of water and you add 10 gallons of water, the level of the pool will not visibly rise. You know that the pool level rose; however, you cannot see it.

Lesson 5 Discovering relationships

With Predictive Insights, you can see the relationships between key process indicators. In this lesson, you learn how this mathematical analysis yields relationship models of your infrastructure where you can see connections between various KPIs.

Relationship #1: Analytics allows for causal modeling

- The Granger causality test is used for determining whether one time series is useful in forecasting another.
- For example, are these KPIs related?



Granger, Clive W. J. "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods." Econometrica 37, (July 1969), 424-438.

Relationship #1: Analytics allows for causal modeling

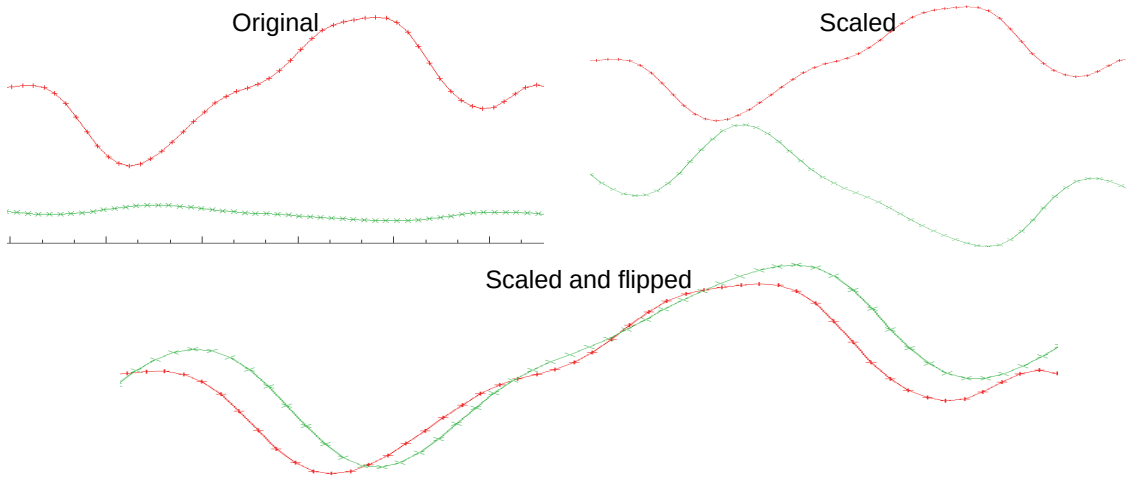
There are currently two types of relationships that are being made with the mathematical model of the data. The first relationship comes from determining whether one set of data tends to predict the outcome of another set of data. This analysis is the Granger causality test. This test comes from the statistical modeling that came from the Nobel award winning economist, Clive Granger. The analysis that he developed in the 1969 is a statistical hypothesis for determining whether one time series of data is useful in forecasting another.

The slide displays two sets of data. One must determine whether these metrics are related. Initially it is hard to determine whether these two data sets have any relationship.

Note: Just because one set of data is useful in predicting another set of data does not mean that trends in the former data set are causing the trends in the latter. **Correlation does not imply causality.** True causality can be a deeply philosophical argument. The Granger test finds only predictive causality.

IBM Training

Granger normalizes and compares multiple data sets

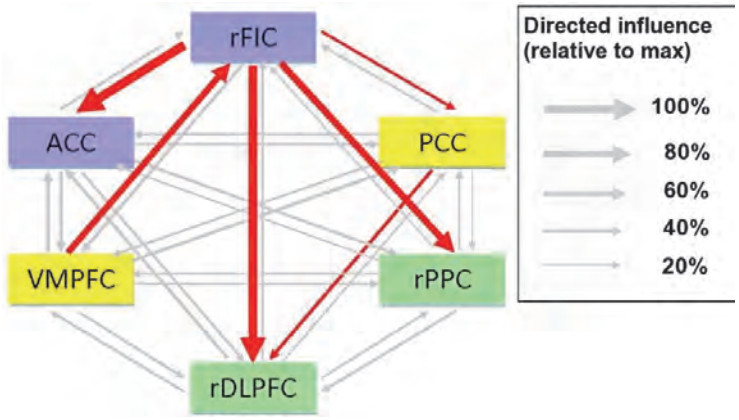


Granger normalizes and compares multiple data sets

The Granger analysis does a number of things to determine causality. First, it takes the data and normalizes it on a scale of 0 to 1. This step allows the differences in the data values to be removed from the calculations. After scaled, it compares these two data sets to see whether one of them tends to predict the other with a relevant lag in the response. As you can see, after scaled and flipped, the red data tends to predict the green data when you consider the lag in the timing. This correlation makes sense when you consider what this data represents. The red data reflects the number of unemployed works and the green consumer confidence. As the number of unemployed decreases, there is a correlated effect of rising consumer confidence (look at the scaled data to see this inverse relationship).

Compare all the KPIs to each other

Compare the relative strength of each relationship and select the strongest ones



Sridharan, D., Levitin, D.J. & Menon, V. A critical role for the right fronto-insular cortex in switching between central-executive and default-mode networks. Proceedings of the National Academy of Sciences of the United States of America 105, 12569-12574 (2008).

Compare all the KPIs to each other

Another way to view these causal relationships is to view them with all the data sets you are comparing. The Granger causality test comes with it a measure of how well or how influential one set of data has on another. The test compares all the data sets with each other and then judges which ones are the strongest. Here in this example, you see a series of proteins that are involved in brain activity. After measuring each proteins response to stimulation and analyzing the data, it is possible to determine that certain proteins directly affect other proteins. For example, the activity in rFIC protein tends to direct the actions of the ACC, rDLPFC, and rPPC proteins and to a much lesser extent, the PCC protein. The comparison of all the data sets often leads to results were there is no statistical relationship whatsoever, hence the many gray lines on the slide.

Mathematical relationships

- Have no relationships with known topology
- Can conflict with customer perceptions
  - Mental model of infrastructure
  - Beliefs about relationships
- Can reveal unrealized interactions
  - Customer wants to track number of logins to servers
  - System alarms display a relationship between servers that are not physically connected
  - Employees log in to both applications simultaneously
  - System alarm occurs when an authentication server is unavailable
- Can be coincidental
  - Relationships come and go with new historical data

Mathematical relationships

By using the mathematical analysis of the data, and the relationships that are generated, you can use Predictive Insights to generate new alarms and improve others. A new alarm might indicate a change in a pattern over the last 28 days. The alarm might also enrich another alarm by proposing other KPIs that might influence the situation that is being investigated. The relationships are mathematical, not topological. While it is possible that topology information might be gleaned from the analysis, it can find other relationships that do not make sense. Predictive Insights can reveal interactions that were not realized by even those users who know their infrastructure well.

For example, a customer was tracking the number of logins over 24 hours. Based on the mathematical model, the login patterns of two of the applications were similar. A relationship between them was identified. When an alarm was generated that was declaring this relationship was broken, the customer declared there was no relationship because the two systems were separate. Upon further review, it was seen that people were logging in to the two applications when they arrived at work to begin their day. The alarm was generated because the authentication server for one of the applications was down. The customer learned something new about the systems and patterns of use because of the alarm.

However, sometimes relationships are purely coincidental. Relationships are reevaluated every time that the system retrains itself and can change when the data changes.

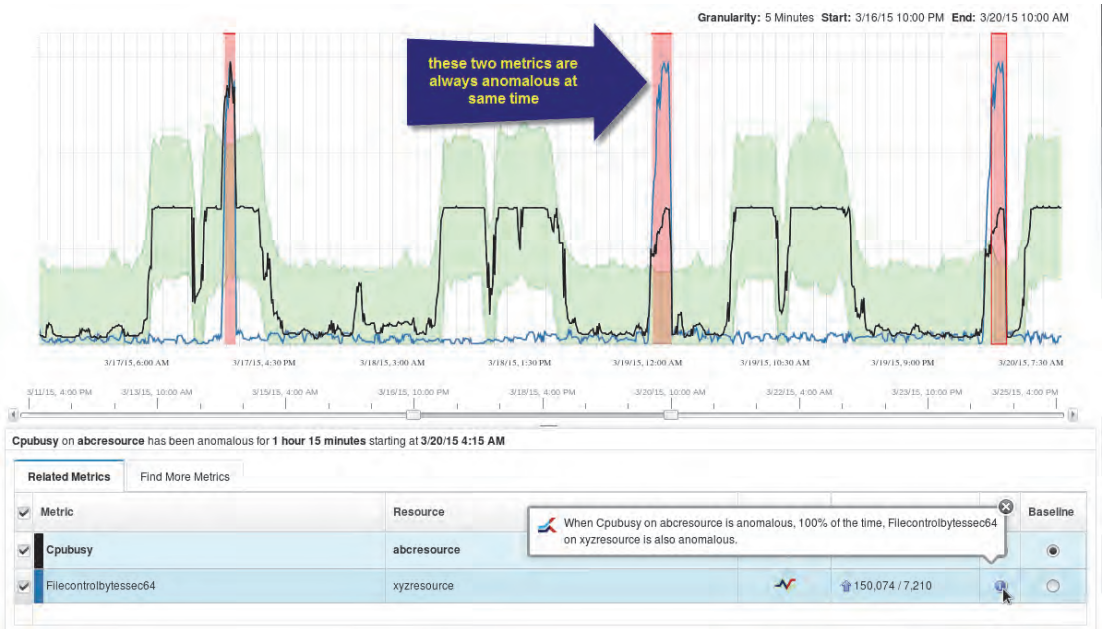
Relationship #2: Relating events when under stress

- Predictive Insights notes when KPIs are under stress
  - Is there any correlation in the timing between various extreme behavior?
  - Spikes in activity at the same time suggests that KPIs might be related
- Analogy: Observing the lunch room
  - On Monday you see Amy and Anthony walk in at the same time, which might be coincidental
  - On Wednesday you again observe both Amy and Anthony come in again together, perhaps still by chance
  - On Friday you make a third observation of Amy and Anthony arriving together in lunch room
  - Conclusion: Amy and Anthony work closely together

Relationship #2: Relating events when under stress

A second relationship that is developed between key performance indicators is noting when they react together when one or more items are under stress. Stress here is meant by unexpected high or low values for a KPI. By looking at the statistical probabilities, it is suggests that there might be a connection between them. An analogy might be helpful in understanding this idea. You go to lunch at work every day. On Monday, you see two people walk in; Amy and Anthony. Unless you see them holding hands, you cannot judge much from this observation. On Wednesday, you again see Amy and Anthony walk into the lunch room together. This observation still might be by chance. On Friday, you see for a third time that Amy and Anthony entering the lunch room together. At this point, you might conclude that Amy and Anthony work closely together. By seeing KPIs spike at the same time for numerous observations, you can make this same type of assumption.

Viewing related events



Viewing related events

When KPIs spike together when under stress, they are called related. Here is an example. If you look at the graph in the slide, you see Cpubusy data trends are not reflected in the Filecontrolbytessec64 data. The only time you see something synchronous happening is when both are under pressure. These two items show no causal relationship, but do show a relationship when under stress.

Relationship #3: Highly correlated data

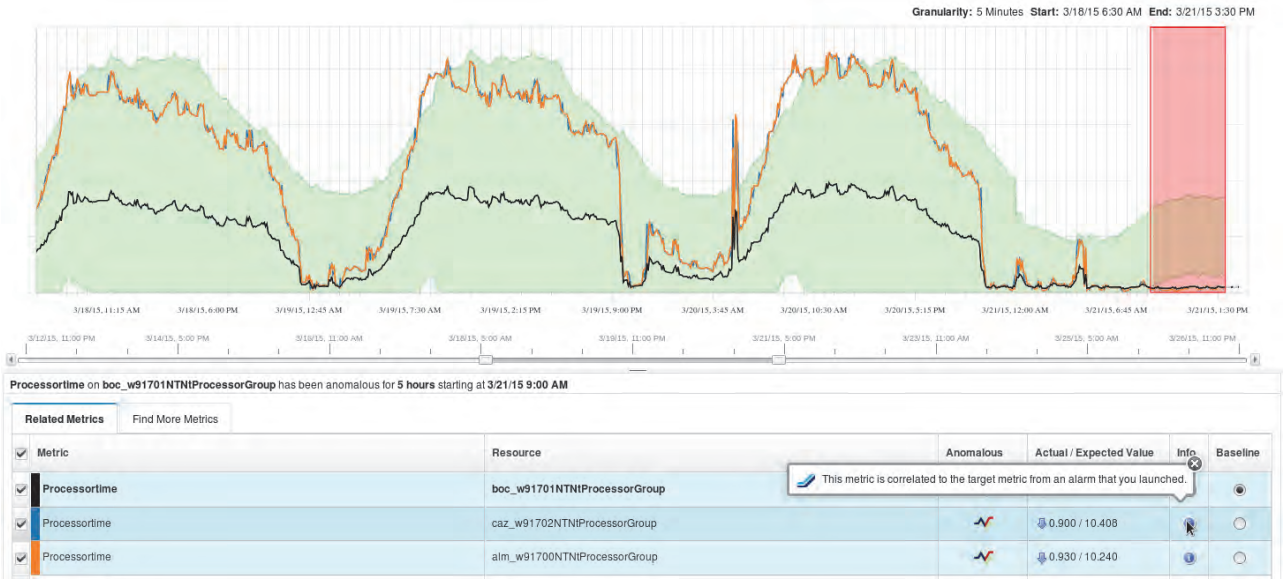
- A special form of data relationship
  - Data is much more tightly related
- Does not create alarms
- Shows user other metrics that are tightly related to the one being investigated
- Correlation is currently limited
  - Metrics that have the same name in the same metric group on different resources
  - Metrics on the same resource with different names
  - A maximum of three correlations will be displayed

Relationship #3: Highly correlated data

A third relationship is associated with data that is tightly correlated with each other. Unlike a Granger analysis, these data sets trend together without the concept of delay. Their values are closely linked together. Correlation is not designed to produce alarms or detect anomalies. It is used as a tool to show you other KPIs that are tightly related to the one you are viewing. Also, unlike Granger which compare each KPI with every other KPI, correlations are limited in their scope. Correlations are only drawn between metrics with the same name but different resources within the same metric group. Also, a correlation can be found between different metrics on the same resource. Currently, only three correlations can be found for any individual metric.

IBM Training

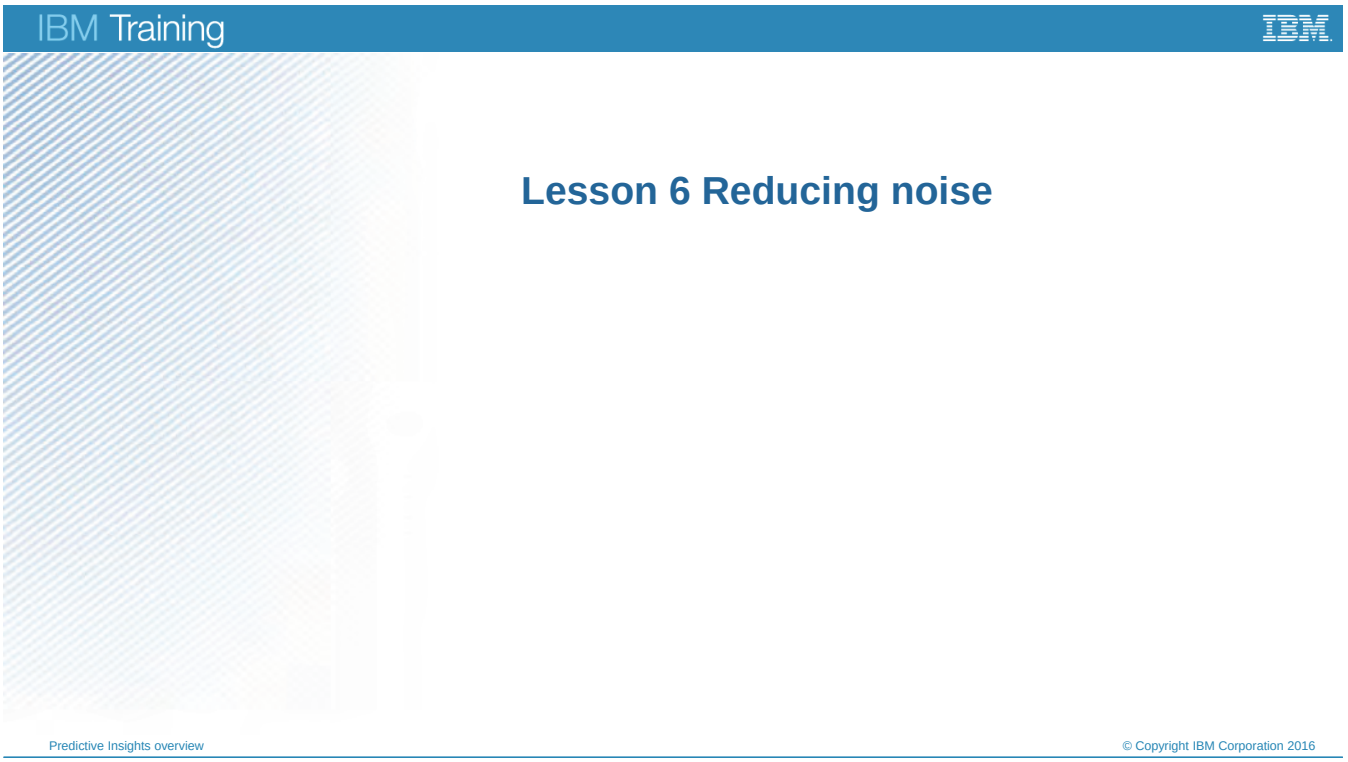
Viewing correlated data



Viewing correlated data

You can determine if a KPI is correlated to another by pointing to the information icon on the KPIs that appear in the related metrics tab of the Predictive Insights user interface.

Lesson 6 Reducing noise



Predictive Insights has a number of methods to reduce the number alarms that are generated by the software. These methods are called noise reduction.

IBM Training

Modifying N of M

- N of the last M intervals have to be anomalous to send event to OMNIbus (AEL)
- Default is 3 of 6
  - nofm.error.count.min.size: 3
  - nofm.error.count.window.size: 6
- Why tune N of M?
  - Anomalies that last longer tend to be of more interest
  - Reduces number of events in AEL

Bank1

N\M	5	6	7
2	162%	165%	168%
3	98%	100%	104%
4	69%	71%	74%
5	50%	54%	55%
6	0%	41%	45%

Bank2

N\M	5	6	7
2	144%	146%	148%
3	98%	100%	102%
4	81%	83%	84%
5	71%	73%	74%
6	0%	66%	68%

Telco1

N\M	5	6	7
2	165%	167%	169%
3	98%	100%	102%
4	68%	71%	73%
5	46%	51%	53%
6	0%	36%	40%

Modifying N of M

The one technique that is exposed to customers is the modification of the N of M rule. By modifying either the N (nofm.error.count.min.size) and/or the M (nofm.error.count.window.size), you can increase or decrease the number of events to your monitoring solution (OMNIbus). Note that the three examples where the N of M values were changed. Using 3 of 6 as the starting point, you can see how many more or how many fewer alarms were generated when looking at the exact same data sets. Depending on the nature of the systems being monitored, you might want to change these values to suit your specific needs.

IBM Training

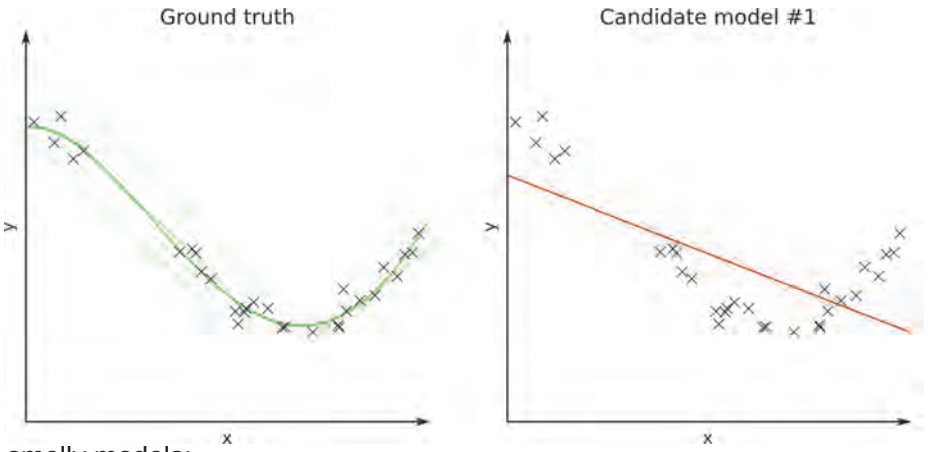
Model pruning

- **What is it?**
  - A way to ensure that Predictive Insights only deploys effective models
- **Why do it?**
  - It's an effective way to **reduce** the false alarm rate while maintaining a true detection rate
- **How does it work?**
  - Run the data that created the model, through it
  - Just like opening a carton of milk

Model pruning

Another noise reduction technique happens in the background without user intervention. Model pruning is a method to ensure only useful models are deployed to the system. Pruning is intended to reduce the number of false alarms while still maintaining a true detection rate. As models are built they also are tested using the data that was used in their construction. When this data is run through the model, the system uses heuristics to determine if the model is acceptable. Its like opening a carton of milk. If you smell something funny, you throw the milk out, otherwise, you use it for whatever you intended.

Not all models are created equal



Causes of smelly models:

- Using an algorithm that is not appropriate for the kind of data (square peg, round hole).
- Training on data that contains lots of anomalies.

Not all models are created equal!

Sometimes, the models that are used to predict data are not up to the task. An inappropriate algorithm might not provide a useful trend. For example, a data set that shows tendency to be modeled with sine or cosine function uses a linear regression as noted in the slide. Here the linear regression has much inherent error if it is to be used to predict data. Another means to create a smelly model is to use data that is already anomalous. In this case, you are trying to model behavior that you do not want included.

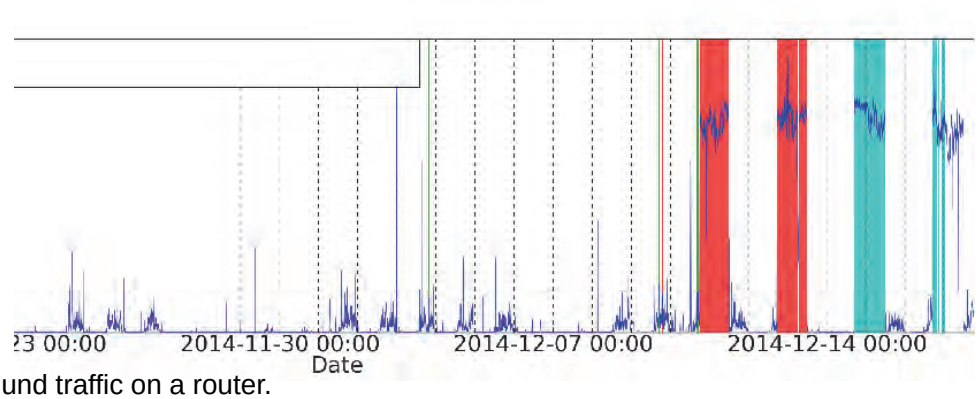
How pruning works

- By definition, anomalies should be infrequent
- Therefore, if an anomaly detector finds lots of anomalies, it's probably not a good anomaly detector
- Test the model quality using the data the model was trained on
  - If it finds very few anomalies, the model fits ➡ deploy it
  - If it finds a lot of anomalies, the model is bad ➡ throw it out
- Results
  - Fewer bad models ➡ fewer spurious alarms
  - Still deploy good models ➡ still catch relevant problems
  - Overall, the false alarm rate is lower, the number of alarms per algorithm is lower and perceived quality is higher
- Model pruning is enabled and configured for all algorithms by default

How pruning works

Pruning works in the idea that anomalies should be infrequent. That goes back to the idea that you have created solutions based on strong design principles and on top of quality infrastructure. Using this as the assumption, if a model produces a lot of anomalies when it reviews the data that created the model, it's probably not a good detector of problems. Pruning deploys fewer bad models into Predictive Insights and helps to ensure that the alarms generated are relevant.

Example: Change in normal behavior



- **Data:** Inbound traffic on a router.
- **Behavior:** Consistent seasonal activity over time, sudden change in magnitude from a specific point onwards.
- **Result:** Pruning ensures that Predictive Insights alarms on the initial change in behavior (red) but does not continue to alarm when this new behavior persists (cyan).

Example: Change in normal behavior

Here is an example. A router sees consistent low-level, seasonal usage. However, there is a major change in activity brought about by a change in the network infrastructure. Maybe there was a failover to a remote site and this router is carrying the entire load of the solution. Regardless, the system notes an anomaly for a time as noted by the red bars. However, by using pruning, the system quickly learns that this behavior has become the new normal and learns that alarming on this anomaly is no longer needed. The old model is no longer a good tool to use with this KPI and is either replaced or temporarily disabled.

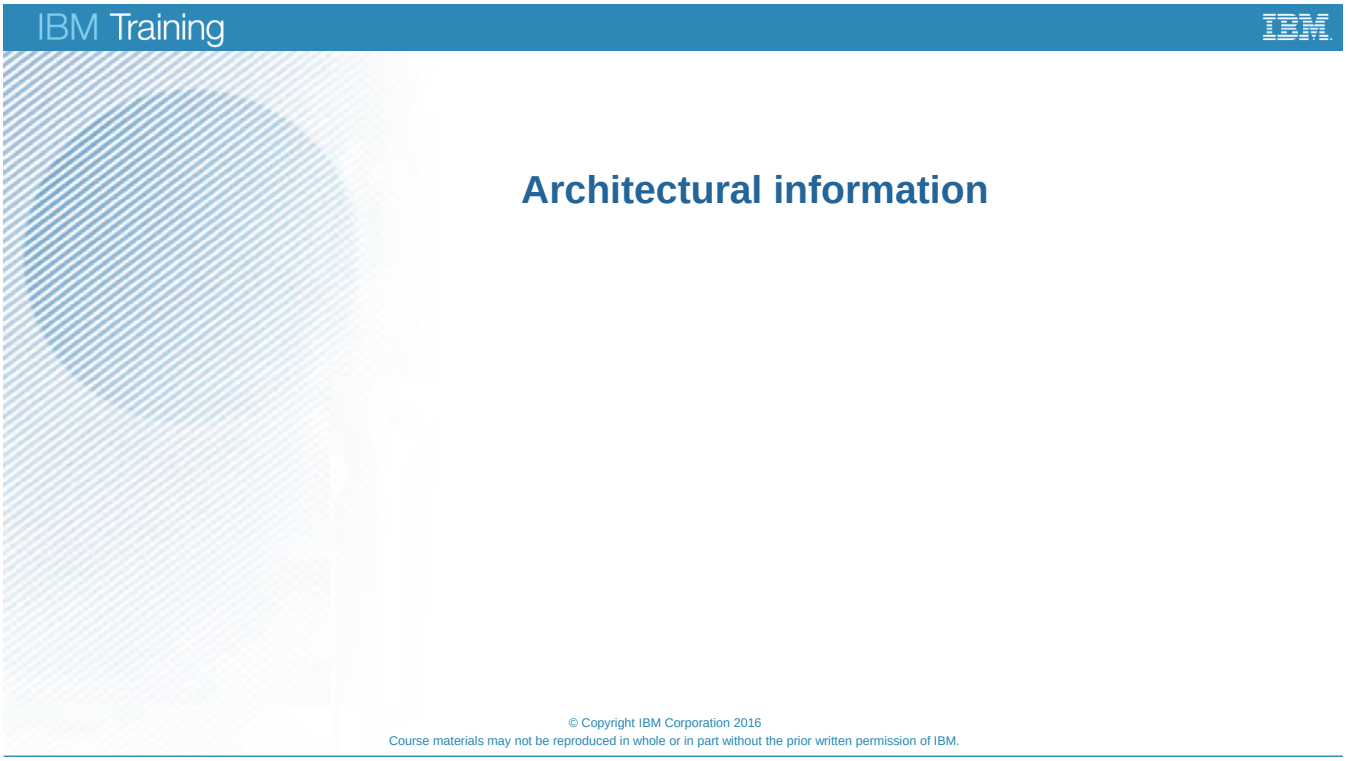
IBM Training

Unit summary

- Explain the value of the analytics
- Describe Predictive Insights and its features
- Explain alarms on anomalies
- Describe relationships between performance indicators
- Describe detectors
- Controlling the number of alarms

Unit summary

Unit 2 Architectural information



This unit is about the physical and logical server layout that is needed to implement Predictive Insights. You discover the role of each piece of the architecture. You also learn how to segment data to support large implementation and the resources necessary to support them.

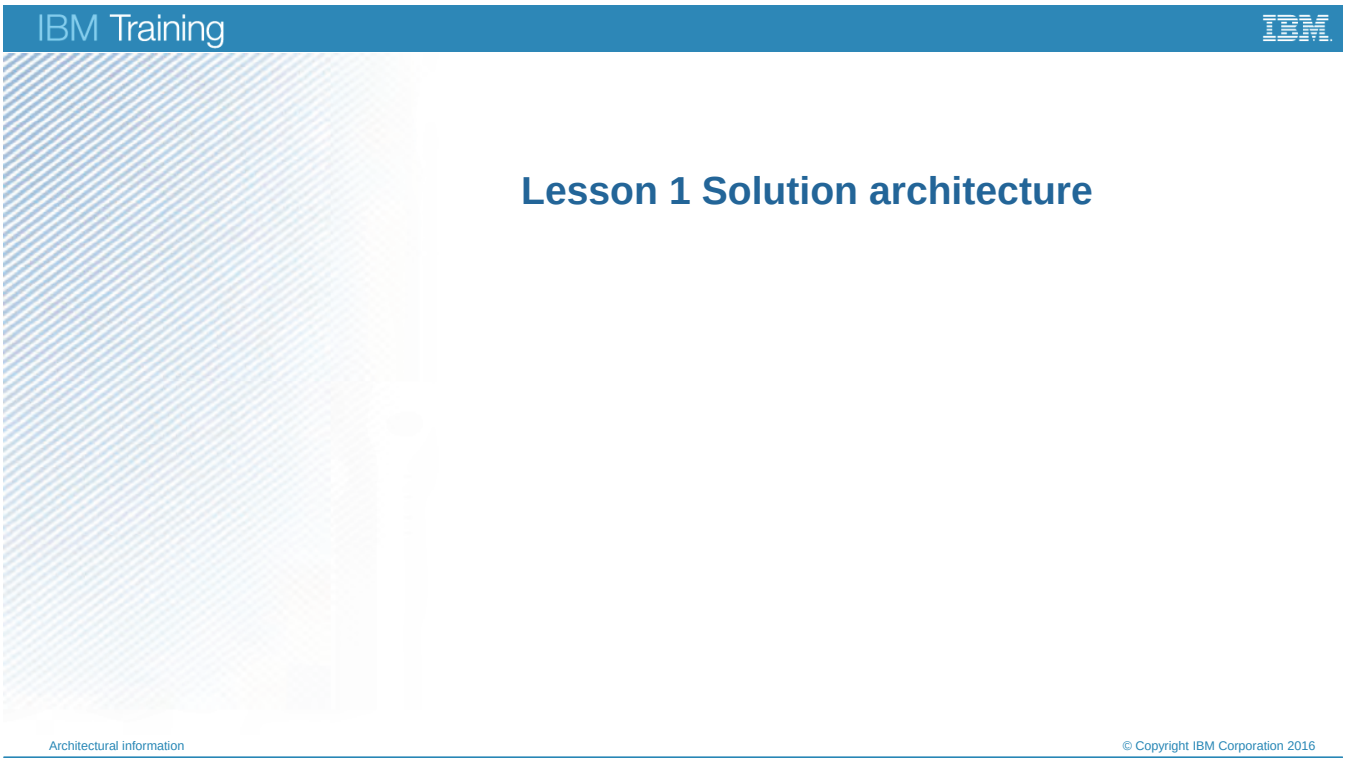
IBM Training

Unit objectives

- Describe the server layout options
- Describe the roles of each server in the implementation
- Describe topics and how they impact the solution
- Calculate the required resources for implementation

Unit objectives

Lesson 1 Solution architecture



This lesson is about the solution architecture that is used by Predictive Insights and how you can use it to segregate data.

IBM Training

IBM Operations Analytics Predictive Insights

Included software

- IBM DB2 Workgroup Server Edition 10.5 Fix Pack 3
- Jazz for Service Management 1.1.1 (after patching from 1.1.0.3)
- Netcool/OMNIbus 8.1
- Netcool/OMNIbus WebGUI 8.1
- InfoSphere Streams 3.2 (InfoSphere Streams 4.0*)
- IBM Operations Analytics Predictive Insights 1.3.3

* Streams 4.0 can be used for customers who do not allow SSH in network

IBM Operations Analytics Predictive Insights

The Predictive Insights 1.3.3 release includes the software that is listed on the slide.

IBM Training

Supported operating systems

- Data mediation client (Eclipse)
  - RHEL 6
  - Windows
- Analytics server (Streams, Predictive Insights)
  - RHEL 6 x86-64
- UI and database server (Jazz for Service Management, OMNIbus, WebGUI, DB2)
  - AIX
  - Linux
  - Windows

Supported operating systems

You can host the servers only on certain operating systems. The data mediation client is supported on RHEL 6.x and Windows, You might be able to run them on other Linux operating systems, but they might not be supported. The Analytics server is only supported on RHEL 6.x running on a 64-bit architecture. The UI and database components can run on any server that has the correct versions of WebGUI, DASH, Tivoli Integrated Portal, and DB2® are running on. The same is true for OMNIbus.

IBM Training

Supported configurations

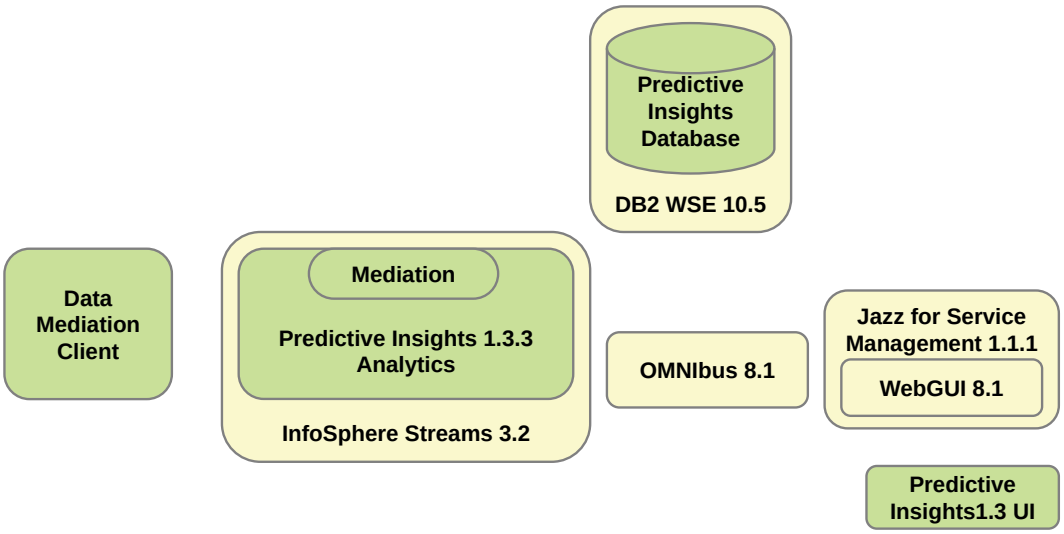
- Support stand-alone and distributed installations
- Currently support TIP and Dash (preferred) installations
  - UI installed directly into TIP
  - UI a separate WebSphere Liberty server along side Dash
- Distributed installation requires Predictive Insights analytics collocated with InfoSphere Streams
- Predictive Insights data mediation client is supported only on Linux and Windows

Supported configurations

You can install the Predictive Insights solution in a stand-alone or distributed configuration. In distributed installations, you must install Predictive Insights software on the same server as the InfoSphere® Streams application. Also, the Predictive Insights User Interface component is installed on the same server that hosts the Tivoli Integrated Portal software. This is not the case if you are installing into DASH. If you install the data mediation tool on a client server, the Predictive Insights solution is supported only on Linux and Windows operating systems. In most proof of concept and early implementations, you can have all software on the same server.

IBM Training

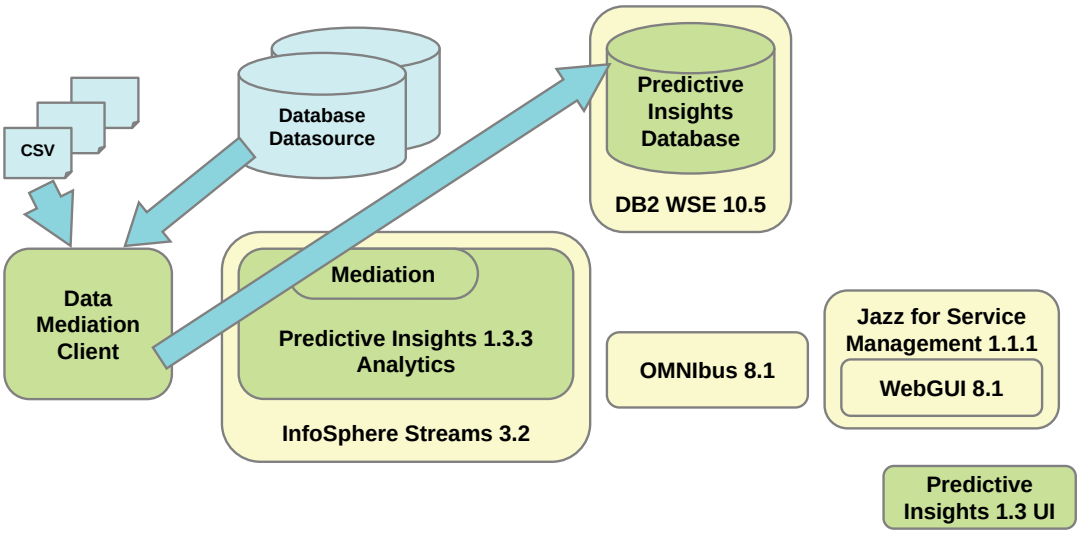
Predictive Insights component structure



Predictive Insights component structure

On the slide is an overview of the components of Predictive Insights. You have a data mediation client to define the data model and location of data sources from which the analytics engine draws data. The Predictive Insights analytics engine is built upon the InfoSphere Streams software. The Predictive Insights database is hosted inside a DB2 instance. An OMNIbus server captures the Predictive Insights alarms. Jazz™ for Service Management hosts the WebGUI portal. Through DASH, users can access the alarms and from there, the anomaly details. The Predictive Insights User Interface is a federated web server that you access through the DASH interface.

Predictive Insights data modeling

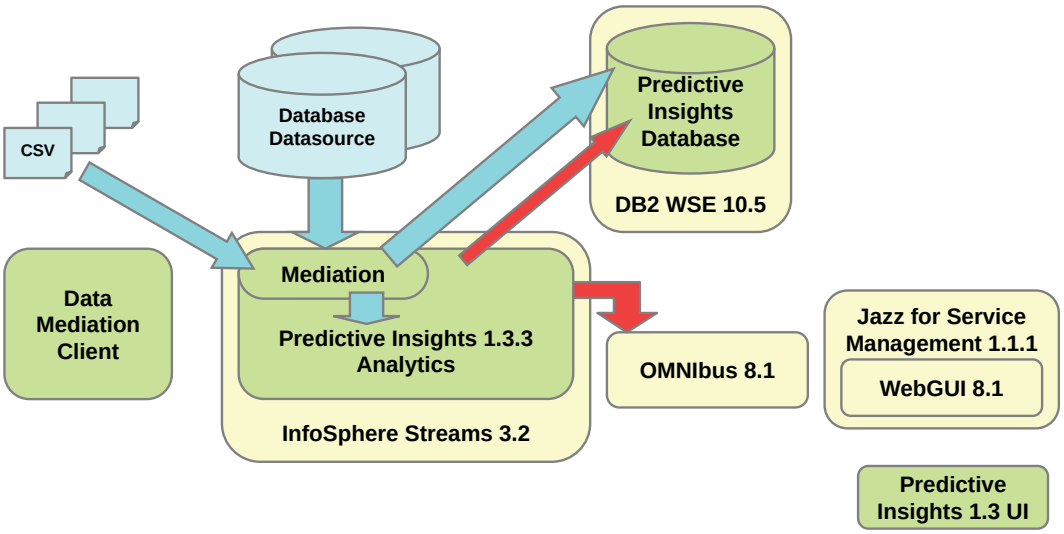


Predictive Insights data modeling

You use the data mediation client to define the location of the various data sources that are to be included in the analytics model. You also use it to define what portions of those data sources (that is, what metrics) are to be included. After you connect to these data sources with the client, you select your metrics, test your extractions, and save the resulting data model to the Predictive Insights database. If you model CSV files, the mediation client must be on a server where you can access the CSV files. If the CSV files are on separate servers, an NFS mount or other file transfer method might be used. Once you define your data model, you deploy it up into the Predictive Insights database.

IBM Training

Predictive Insights data analysis

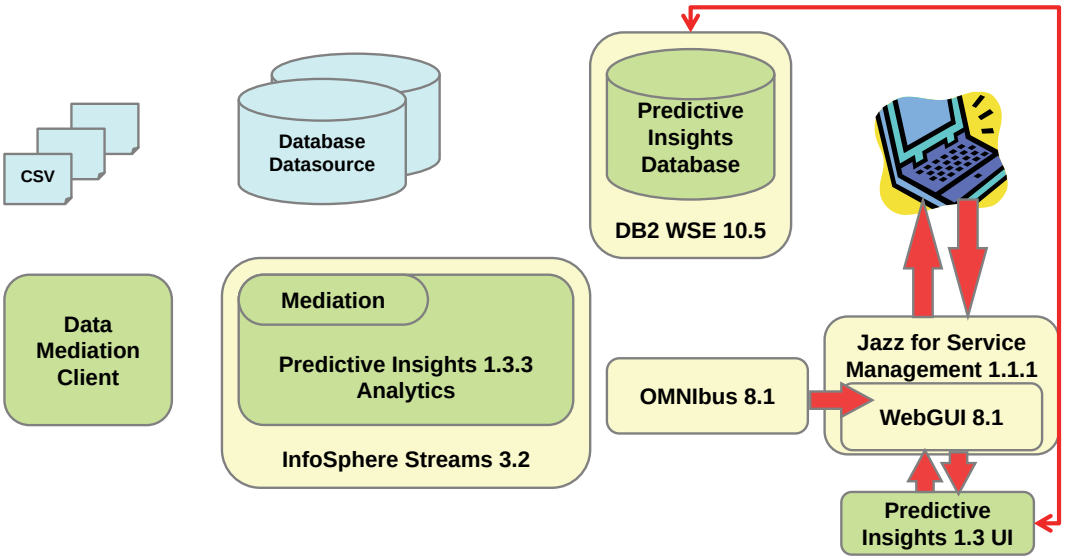


Predictive Insights data analysis

After you model data with the mediation client, you can start the analytics product. The analytics server has a mediation process that uses the stored data model to connect and extract data from the defined data sources. The server normalizes it and stores it in the Predictive Insights database. The server extracts the data, based on the aggregation interval. This data is used in the analytics engine to build the statistical model. After the model is built, it can search for anomalies. When anomalies are found, they are saved to the database. If enough anomalies are found for a KPI, then an alarm is sent to OMNIbus.

IBM Training

Reviewing anomalies

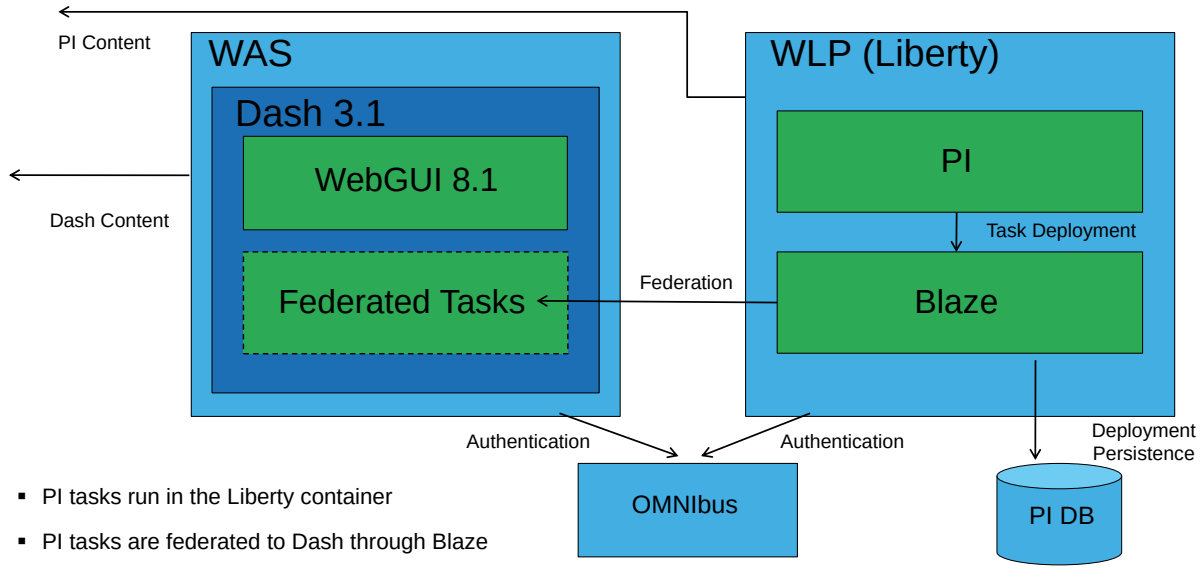


Reviewing anomalies

After an alarm is sent to OMNIbus, operators use the alarm in the WebGUI interface to monitor the solution. If the operator chooses to, they drill down into the alarm and look at the Predictive Insights UI. In the UI, you can find the details of the performance data that led to the alarm. The UI component goes back to the database to gather all the necessary details to generate the charts that are displayed in the GUI. You do not access the PI user interface directly but as a federated application that is access through DASH

IBM Training

Dash support: Federated Architecture



Dash Support: Federated Architecture

Predictive Insights is using a federated architecture to deploy its interface. Instead of deploying the interface inside of the WebSphere® container like WebGUI, there is a separate WebSphere container. The Predictive Insights user interface is running in a separate lightweight web application container that is federated into the DASH interface. Using Blaze as the means to control security and user access, you can access Predictive Insights through DASH with single sign on.

IBM Training

Data segregation

- You do not have to segregate data
- Customers might choose to segregate data
  - They have more than 250,000 KPIs to monitor
  - Prevent alarms from being seen by other groups
  - Isolate applications from multivariate analysis
  - Address issues with data collection intervals
- Topics are used to segregate data

Data segregation

Predictive Insights allows data (or KPIs) to be segregated into separate logical instances. Segregating data is not required, and is discouraged because it prevents multivariate analysis to occur between logical instances. However, there are reasons to segregate data into these containers. For example, customers might choose to segregate data to limit the delivery of alarms to only necessary groups. Segregation also prevents multivariate analysis between data models, which are known to be independent. Because each segregated set of data is analyzed separately, they can be configured separately to address certain data streams that might not be collected at the same rate as others. Predictive Insights segregates data by using topics.

IBM Training

Topics segregate data

- Topics are logical containers for individual data models
- Topics are independent of each other
  - No analysis occurs between them
  - Alarms can be addressed with separate rules
- Initial topic that is defined at installation time
- More topics can be added
  - Created with separate server installation
  - Created with command-line utility on same physical server
- One topic can support up to 250,000 KPIs

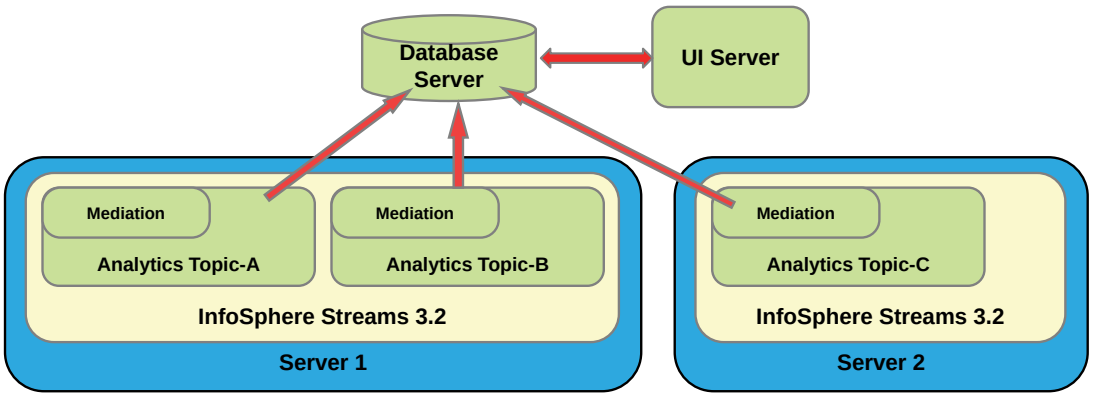
Topics segregate data

Topics are logical containers that create separate, stand-alone data instances. When more than one topic is created, it is independent of any other topic and there is no analysis that occurs between them. Separate systems are distinct entities within Predictive Insights. During installation, you define an initial topic as the default one for the system. If needed, you can generate more topics with an extra server installation if you install on multiple servers. You can also use a command-line utility on an existing Predictive Insights server. One topic can support up to 250,000 KPIs.

IBM Training

Multiple topic solutions

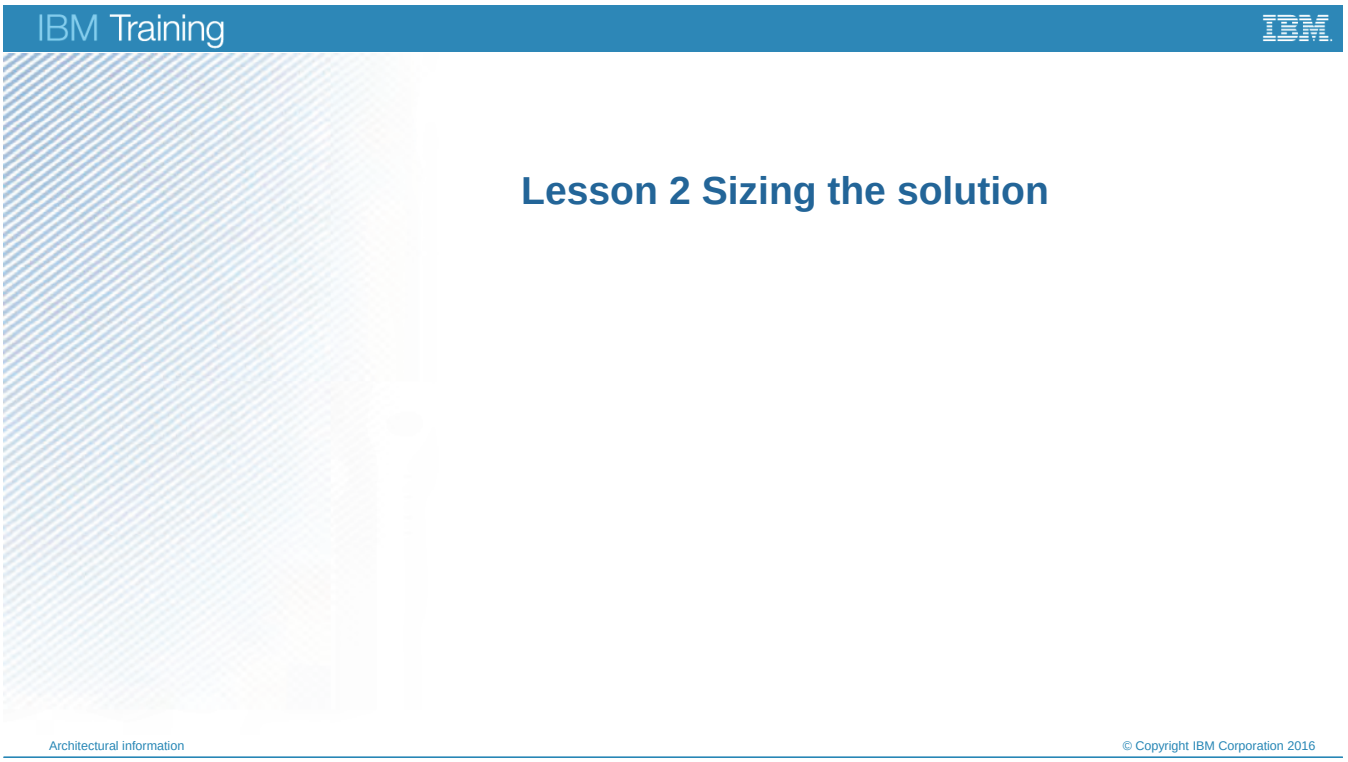
- Multiple topics can run on the same server or a separate one
  Dependent on number of KPIs and server resources
- Detached instances
  No multivariate analysis between instances
- Total KPIs for all topics cannot exceed 1,000,000



Multiple topic solutions

You can have more than one analytics servers in an implementation. Each analytics server defines a specific topic and is associated to a single data model. An installation of InfoSphere Streams can support one or more analytics servers depending on server resources. Each analytics server must have a unique topic name. Each instance is independent, and no multivariate analysis occurs between topics. The person who models the data knows what metrics should be in what topic. In this release, the total number of KPIs across all topics (regardless whether the same metric is used in multiple topics) cannot exceed 1,000,000.

Lesson 2 Sizing the solution



This lesson covers the major influencer on the sizing of your solutions, which includes the number of KPIs and how often you collect their data. You have data on a proof of concept and production implementation.

IBM Training

What effects of sizing

- Number of key performance indicators
  - A KPI is one server-specific metric
  - Example:
    - Total number of servers = 100
    - Metrics measured = 10 = memory_paging, memory_physical_used, memory_swap_used, memory_proc_queue_length, cpu_idle, cpu_user, cpu_system, cpu_wait, bytes_in, bytes_out
    - KPI total = 100 * 10 = 1000
- Aggregation interval
  - How often are you collecting data
  - 5 minute data collects 3 times more data than 15 minute data
  - Interval effects how quickly you are alerted to an anomaly
- Detectors that are enabled
  - Granger detector requires the largest amount of resources

What effects of sizing

The sizing of the solution is based on how many key performance indicators (KPIs) are analyzed by the server. The maximum number of KPIs that the solution can manage is 1,000,000. A KPI is one server-specific metric. For example, one KPI is the measure of processor utilization on Server A. If you have 100 servers in your implementation and you have 10 metrics to monitor, you use 1,000 KPIs of that total 250,000 KPI budget a topic can effectively analyze.

The server is also impacted by the aggregation interval. The more aggregation intervals per hour increases the total amount of memory and disk space that is needed to manage your KPIs. Choosing the aggregation interval affects when you see your first alarm in a situation that is generating anomalous data. An alarm can be generated when three anomalous events are seen on a KPI in six aggregation intervals. For a 5-minute aggregation interval, the earliest you see an alarm is after 15 minutes of the first anomalous event. If your data is flexible and you have a range of aggregation intervals to choose from, you need to select an appropriate interval. Too short of an interval might produce more false alarms because the system does not have time to settle. If you have too long of an interval, you might miss important anomalies. An aggregation interval should be 5 - 15 minutes.

Finally, you can control what detectors are enabled for each topic. The Granger detector, by far, uses the most computing resources. Disabling the Granger detector could allow for more KPIs to be analyzed in a single topic.

IBM Training

Proof of concept sizing of servers

- Up to 60k KPIs, 5-minute aggregation interval
- Analytics server
  - Resources to support DB2 and Streams/Predictive Insights
  - CPU: 8 cores (2.4 GHz)
  - Memory: 25 GB
  - Disk space: 200 GB
- UI server
  - Resource to support OMNIbus, DASH, and UI
  - CPU: 4 cores (2.4 GHz)
  - Memory: 8 GB
  - Disk space: 250 GB

Proof of concept sizing of servers

For smaller solutions or proof of concept servers, you can expect an analytics server with 25 GB of memory and an 8-core processor to support up to 60,000 KPIs at a 5-minute aggregation interval. Today's high performance laptop can host both UI and analytics server if it were analyzing historical data.

IBM Training

Medium production sizing of servers

- 100,000 KPIs, 5-minute aggregation interval
- Analytics server
  - CPU: 10 cores (2.4 GHz)
  - Memory: 35 GB
  - Disk space: 400 GB
- Database server
  - CPU: 4 cores (2.4 GHz)
  - Memory: 18 GB
  - Disk space: 250 GB
- UI server
  - 4 cores (2.4 GHz)
  - Memory: 8 GB
  - Disk space: 250 GB

Medium production sizing of servers

To support a system with 100,000 KPIs at 5-minute intervals, you can expect an analytics server to require a processor with 10 cores and a total of 35 GB of memory. The DB2 server needs 18GB of RAM, 250 GB of hard drive space, and 4 cores. The UI server which hosts Jazz for Service Management and the Predictive Insights UI, needs 8 GB of memory and a 4-core processor. A customer might be already using DASH, which could reduce this requirement.

IBM Training

Large production sizing of servers

- 1,000,000 KPIs, 5-minute aggregation interval
- Analytics server x 4 (250,000 KPIs per server)
  - CPU: 20 cores (2.4 GHz)
  - Memory: 69 GB
  - Disk space: 700 GB
- Database server x 1
  - CPU: 12 cores (2.4 GHz)
  - Memory: 102 GB
  - Disk space: 600 GB
- UI server (may be able to collocate on other servers)
  - CPU: 6 cores (2.4 GHz)
  - Memory: 12 GB
  - Disk space: 250 GB

Large production sizing of servers

To understand the scale of a large production system, each analytics server supports only 250,000 KPIs per topic at 5 minute intervals. A dedicated database server is also suggested to support the data I/O needs.

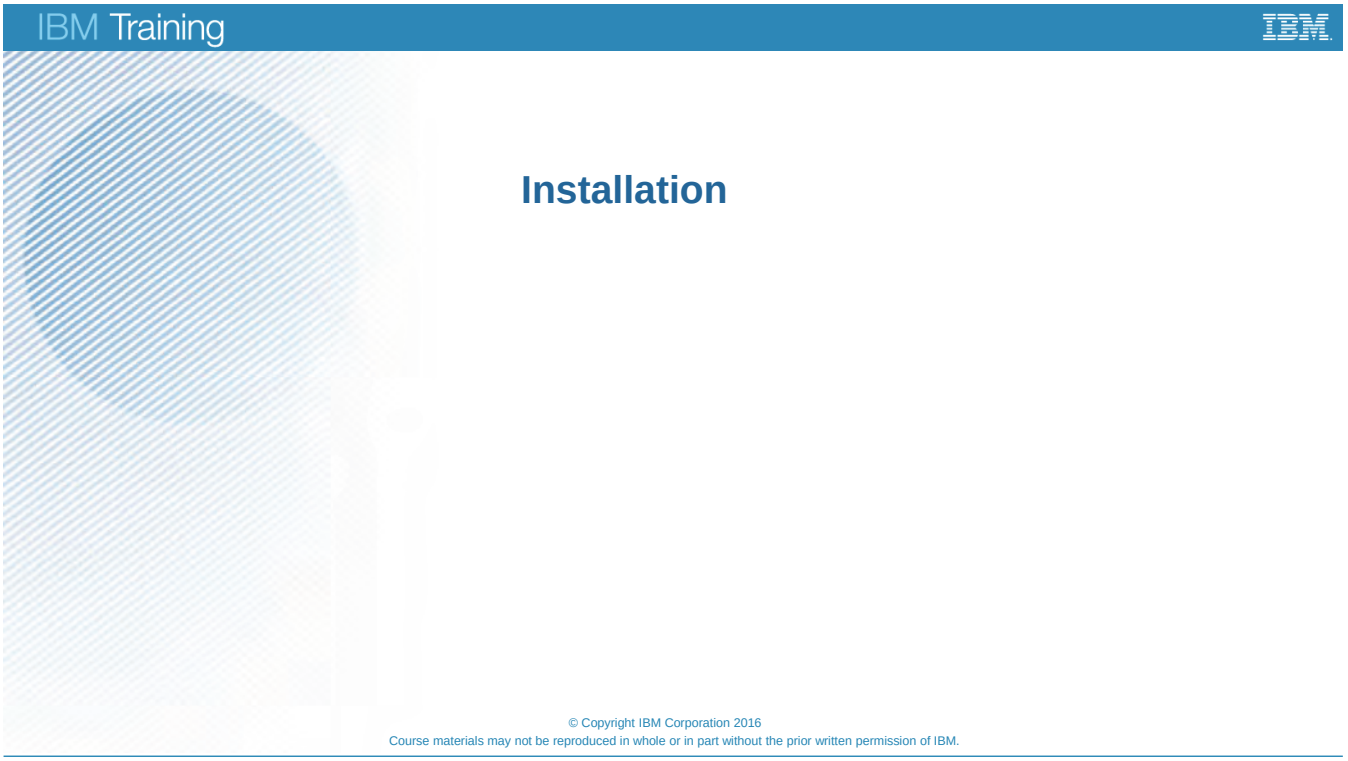
IBM Training

Unit summary

- Describe the server layout options
- Describe the roles of each server in the implementation
- Describe topics and how they impact the solution
- Calculate the required resources for implementation

Unit summary

Unit 3 Installation



This unit provides an overview of the prerequisites and installation process for Predictive Insights. It also presents an optional process to validate that the installation was successful.

IBM Training

Unit objectives

- Describe the installation options
- Describe prerequisites checker
- Validate that installation is successful

Unit objectives

Lesson 1 Installation overview

This lesson provides an overview of the installation process. In the exercises, you complete a partial installation of the Predictive Insights before you begin the next unit.

IBM Training

Necessary software

The following software must be installed before running Predictive Insights installer

- DB2 10.5.0.3
- OMNIbus 8.1 Fix Pack 2
- Jazz for Service Management 1.1.1 (patched from Jazz for Service Management 1.1.0.3)
- WebGUI 8.1 Fix Pack 2
- InfoSphere Streams 3.2*

* InfoSphere Streams 4.0 is supported for customers who do not allow SSH use in their network

Necessary software

IBM Training

Installation types

- Stand-alone (less than 50,000 KPI's)
  - DB2
  - OMNIbus
  - Jazz for Service Management/DASH/WebGUI
  - InfoSphere Streams/Predictive Insights analytics server
  - Predictive Insights UI server
- Distributed
  - Database server
    - DB2 with Predictive Insights schema applied
  - Alarm server
    - OMNIbus 8.1
  - Dashboard server
    - Jazz for Service Management 1.1.1
    - WebGUI 8.1
  - Analytics Server
    - Streams 3.2
    - Predictive Insights 1.3.3 analysis engine
  - User Interface Server
    - Predictive Insights 1.3.3 UI

Installation types

Predictive Insights can be installed on a stand-alone server or in a distributed manner. The only requirements are that the InfoSphere Streams and Predictive Insights be installed on the same logical server that is running RHEL 6.x 64-bit operating system. Also, WebGUI must be installed on the same server that Jazz for Service Management is running on.

IBM Training

Analytics server prerequisites

- Stand-alone installation
- Needed users
  - scadmin
    Owns SCAPI/Streams installations (optionally owns DASH/OMNIbus/WebGUI)
  - db2inst1
    Owns DB2
- Required Linux packages
  - Refer to installation guide
- Setting ulimits for file handles
  - Hard and soft limits set to 100,000
- $TASP_HOME
  - Typically is **/opt/IBM/scanalytics/analytics**
  - Tivoli Analytics for Service Performance (TASP)

Analytics server prerequisites

The requirements that are listed on the slide are for a stand-alone installation of Predictive Insights. To install the product, you need a Predictive Insights administration user, typically **scadmin**, and a DB2 instance owner user, typically db2inst1. You must install numerous packages for both DB2, OMNIbus, WebGUI, and InfoSphere Streams. Both applications have a prerequisite checker tool that checks to see whether these packages are installed. InfoSphere Streams opens many files when working with data sources. To work with that many sources, you increase the ulimits in the RHEL operating system as shown in the installation guide. The name, Tivoli Analytics for Service Performance (TASP), is in several variables and scripts. $TASP_HOME is typically in **/opt/IBM/scanalytics/analytics**.

IBM Training

Prerequisites checker

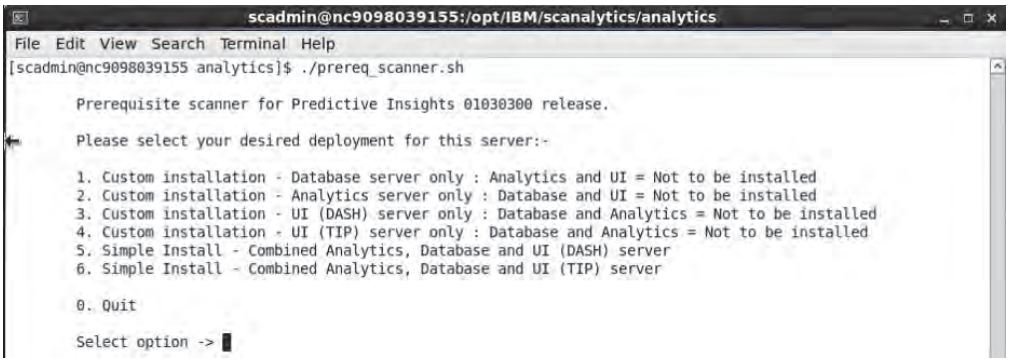
- Stand-alone prerequisite checking tool that analyzes system environments before installation of Predictive Insights and prerequisite software
  - Only checks if prerequisites have been installed
- Checks for the following items:
  - **System** checks including checking for minimum hardware specification PI requires
  - **Libraries and Packages** identifies missing or incorrect versions of necessary libraries
  - **Software** checks, currently DB2 version only
- Supports both stand-alone and distributed installations
- Prerequisite checker included with Predictive Insights installer
  `<INSTALLER-HOME>/predictiveInsightsInstaller1.3.3/prereq_scanner.sh`

Prerequisites checker

The installer has a convenient prerequisite checker to make sure that the necessary packages are installed and the default ports are open and available.

IBM Training

Script prompts for the type of installation

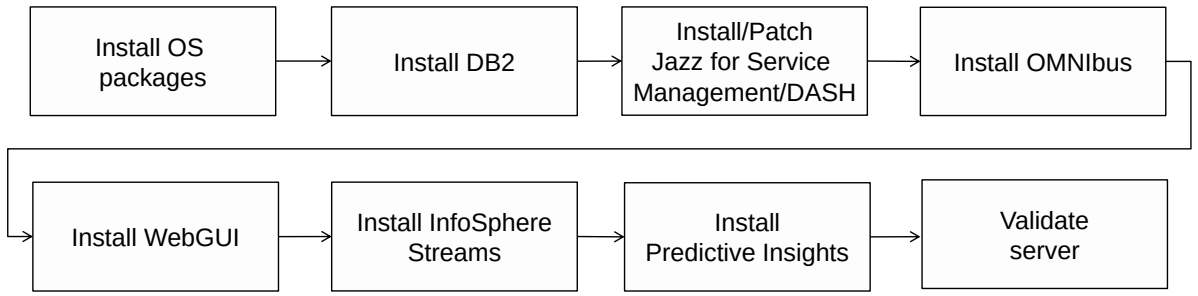


Script prompts for the type of installation

When you run the prerequisite checker, you are prompted for the type of installation you are performing.

IBM Training

Installation on a stand-alone server



Installation on a stand-alone server

After installing the prerequisite packages for the solution on a stand-alone server, you install the DB2 database, Jazz for Service Management, OMNIbus, and the WebGUI, in that order. You then install InfoSphere Streams and configure it to use SSH and validate that it can deploy instances. Next, you install Predictive Insights, which configure the DB2 database, installs the Analytics engine, adds the data mediation client, and configures Jazz for Service Management to use the user interface server. After the installation, you can then validate the server by running test data.

IBM Training

Start up process

- Start DB2 and OMNIbus
- Start Jazz for Service Management/DASH
  - /home/scadmin/IBM/JazzSM/profile/bin/startServer.sh server1 (note home directory)
- Start Predictive Insights UI
  - /opt/IBM/scanalytics/UI/bin/pi.sh
    - Used to start and stop the UI server
    - Usage: ./pi.sh { -status | -start | -stop | -restart }
    - Only restarts the UI server, does not affect Dash in any way
- Log into DASH as ncoadmin
  - Need to use Firefox 24.x
  - Without valid certificates, need to override security warnings between DASH and PI UI

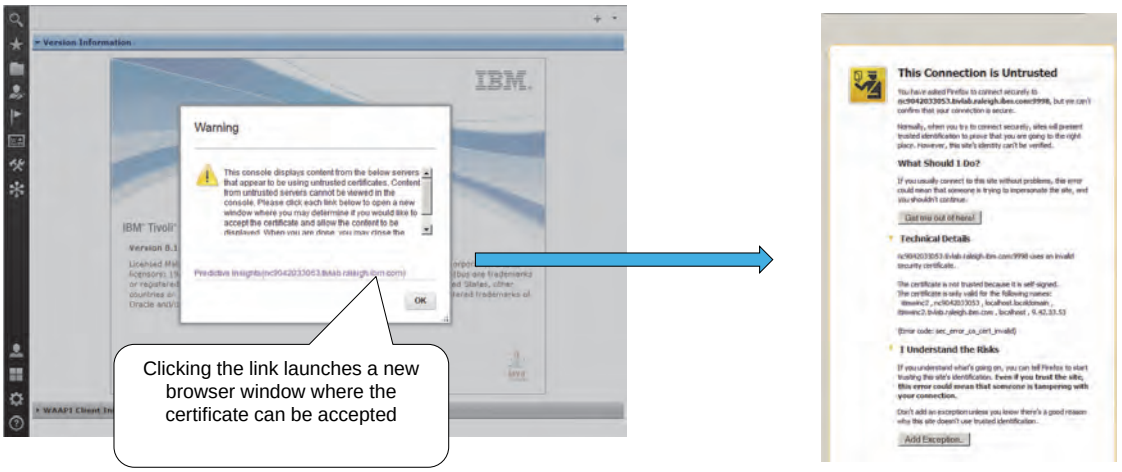
Start up process

After an installation, all the necessary components are up and running. However, if you had to start all the processes from scratch, you would follow these steps. You start DB2 and OMNIbus first. Next, you start Jazz for Service Management,. With it running, you can start the Predictive Insights user interface. This action is a new step from the previous 1.2version's line of code and can be easily missed. Finally, you can launch the DASH interface from a browser. It is important that you initially use a Firefox 24.x browser to address the fact the Predictive Insights user interface does not have a signed certificate.

IBM Training

Certificate acceptance

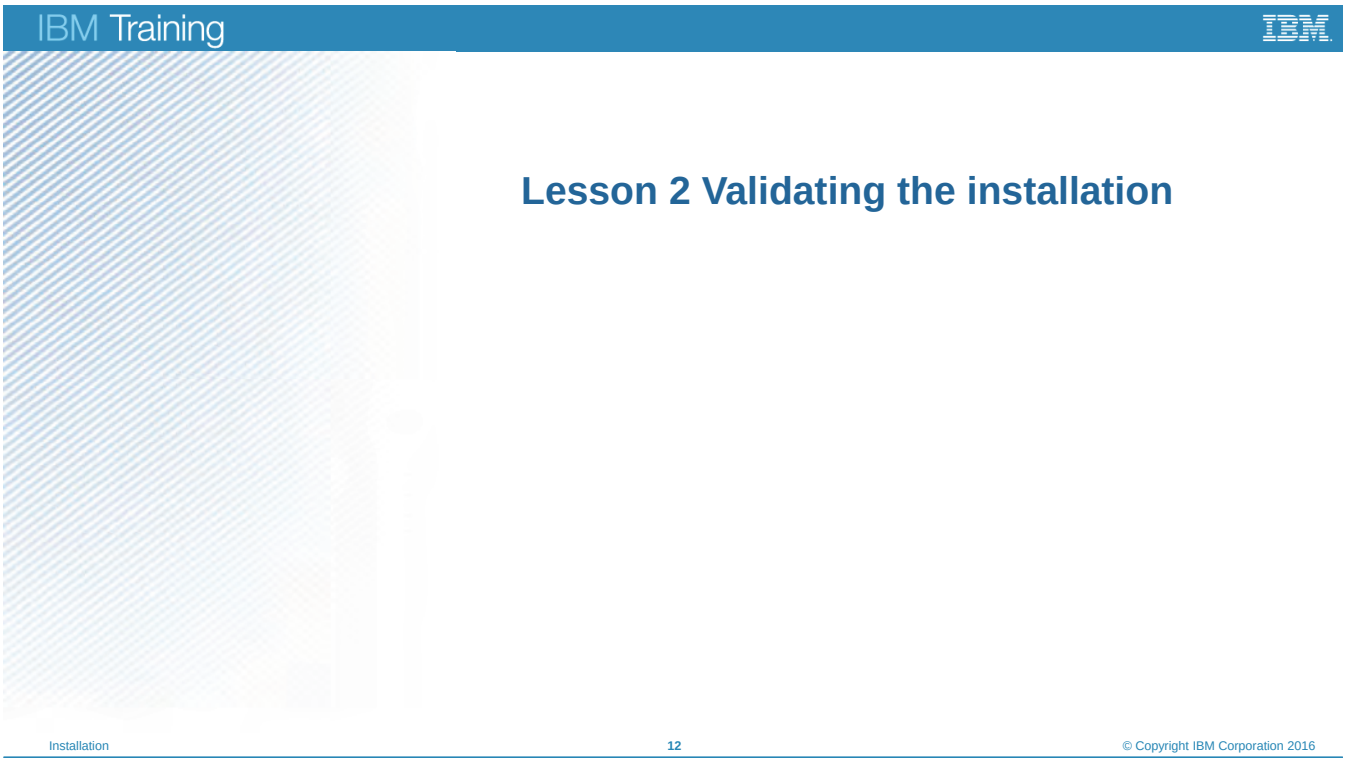
- Product-provided installation contains self-signed certificate in the Liberty container
- Need Firefox 24.x to get the ability to accept untrusted connection



Certificate Acceptance

Without having a valid signed site certificate that is installed on the WebSphere server that hosts Predictive Insights, you need to accept the untrusted certificate. This step requires Firefox 24.x so you have an opportunity to manually accept the certificate. Newer versions of Firefox do not provide easy means to add the exception.

Lesson 2 Validating the installation



This lesson describes a method to ensure that you successfully installed the Predictive Insights application.

IBM Training

Tutorial data

- Useful step to ensure that all pieces are working
- Small data set and predefined data model
- People can see end-to-end operation of tool
- Dataset available on IBM DeveloperWorks wiki
  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20SmartCloud%20Analytics%20-%20Predictive%20Insights/page/Predictive%20Insights%20tutorial%201.3.0

Tutorial data

An optional step to ensure that an installation is working correctly is to run a set of data through the analytics engine. This step ensures that alarms are generated and the user interface generates the correct graphs. You extract a small set of data that has with it a predefined data model. You can complete an end-to-end test of your solution. This data set is on the developerWorks website.

IBM Training

Validating the server process

- Download tutorial data
- Extract files to file system that Predictive Insights (scadmin) has rights to
- Import data model into mediation client
- Check or update access to CSV files
- Ensure that data is extracted from CSV files
- Deploy model
- Run extraction
- Validate alarms and charts
  Historical data is from May 2013

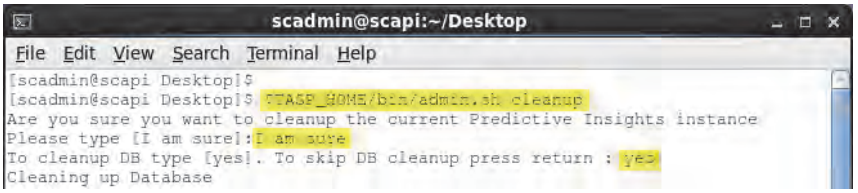
Validating the server process

If you choose to validate your server with this data, you download the tutorial package and place the downloaded archive file on the Analytics server that you created. As the Predictive Insights administrative user (scadmin), you extract the archive to a local directory. You import a data mediation file to the mediation client and use the mediation tool to ensure that you can access and parse the CSV files that were extracted from the archive. You can deploy this model and start the analytics process. After the data is extracted from the CSV files, it finds anomalies that are included in the data set. These anomalies generate alarms, which can be validated in the WebGUI. From the alarms, you can generate the performance charts that describe the details of the anomaly. Remember, the data is from May 2013. You must change the dates on the charts to reach the historical record.

IBM Training

Cleaning out the tutorial data

- Must clean out tutorial data before using customer data
- Run following command as the Predictive Insights user (scadmin):
  `$TASP_HOME/bin/admin.sh cleanup`
- Creates backups of various logs
- Empties database of all analytics data



Cleaning out the tutorial data

If you choose to validate your server with tutorial data, you must remove all this tutorial data before you work with customer data. A cleanup command backs up various files and optionally truncates the data in DB2.

IBM Training

Student exercises

Student exercises

In these exercises, you install Predictive Insights. This exercise is a partial installation. The DB2 database, Jazz for Service Management or DASH, OMNIbus, and WebGUI are installed under the **db2inst1** and **scadmin** users. You install InfoSphere Streams and Predictive Insights. You must successfully complete this exercise before moving to the next unit. The steps that were taken to preinstall the software is included in the student exercise appendix.

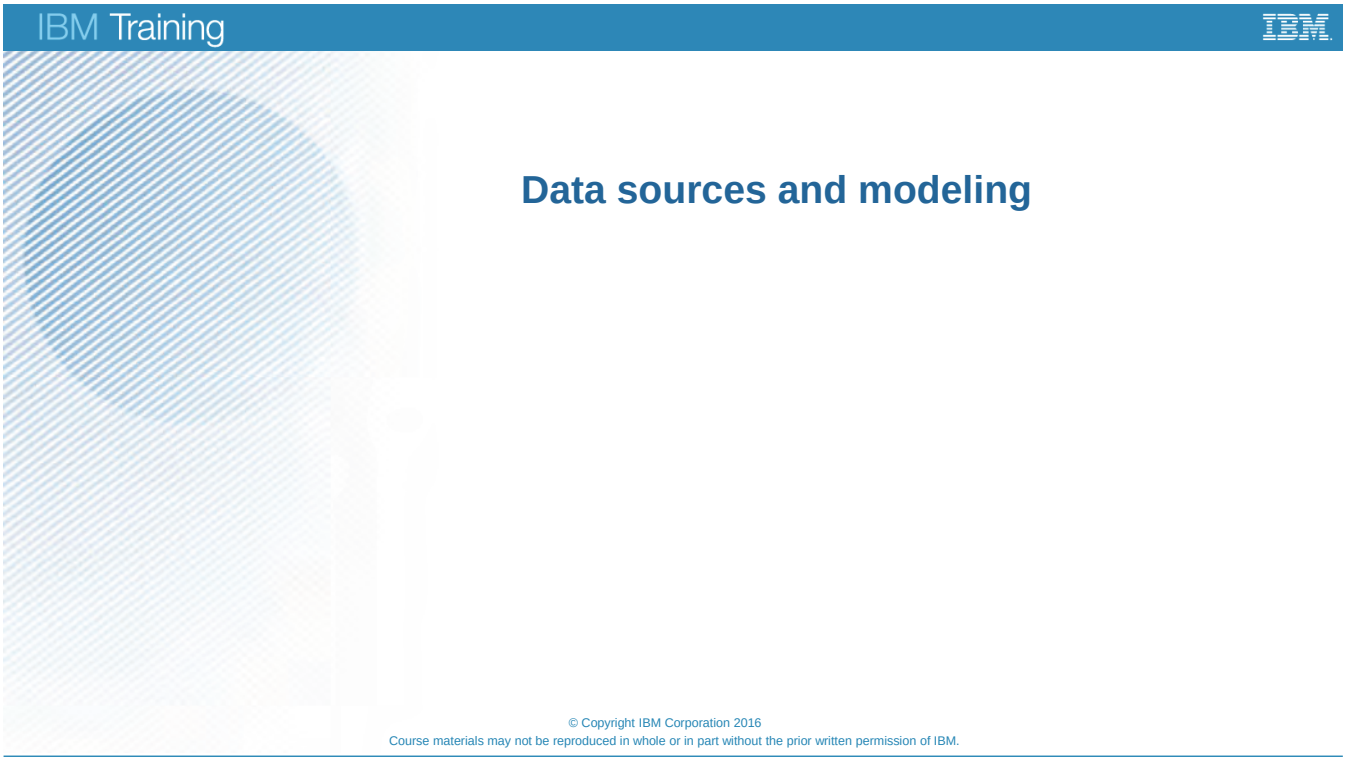
IBM Training

Unit summary

- Describe the installation options
- Describe prerequisites checker
- Validate that installation is successful

Unit summary

Unit 4 Data sources and modeling



This unit provides information about how data is collected and modeled with Predictive Insights. You learn about key performance indicators and how they are selected from the monitoring data that is available. You learn about the supported data sources and how to connect to them, manipulate the data sources, and select only the items that should be analyzed.

IBM Training

Unit objectives

After completing this unit, you should be able to complete the following tasks:

- Select useful key performance indicators
- Connect to external data sources
- Model data sources with the client mediation utility
- Deploy the data model

Unit objectives

IBM Training

Data modeling process

- Select data sources and key performance indicators
  - Databases and flat file CSV data
  - KPIs that change with time and address customer experience
- Use data mediation client to connect to data sources
  - Connect to data sources through JDBC or file system
  - Select resource name and time stamp data columns
  - Select useful KPIs
  - Select how to consolidate multiple data points (average, min, max)
  - Modify or filter resource names as needed
  - Test data extraction
- Deploy model into Predictive Insights server

Data modeling process

First, you must determine what databases and CSV files you need to access to gather the required performance data. This selection process is important for useful alarms to be generated. Then, you use the data mediation client to connect to these data sources and begin the process of modeling the data to ensure that you have useful resource names and time stamp information. You also select the most useful metrics and filter resources as necessary. After testing the data model and its ability to extract data, you deploy it to the analytics server.

Lesson 1 Selecting useful KPIs

This lesson provides a series of best practices for selecting key performance indicators (KPIs) that you want Predictive Insights to include in its analysis.

IBM Training

What is a key performance indicator

- Performance indicators vary depending on the organization
  - Marketing: Customer attrition
  - Manufacturing: Cycle time
  - Project management: Earned value
- A KPI or metric is a single measurement of one node that makes a key service that your company provides
  - Free memory in MB on Server_B
  - Bytes in per second from Router_1
  - Temperature in Celsius of inlet to chiller unit #1
- Must be based on time-series data instead of event data
- Data should vary with time

What is a key performance indicator

Organizationally, a key performance indicator is a number that has significant meaning to a specific group. For example, cycle time in manufacturing is a number that affects, for example, inventory, revenue, or profitability. Other indicators might typically address business performance. In Predictive Insights, a KPI or metric is a measurement of one property on a specific node in a key service that your company provides. These values can include the free memory that is available on a server or a response time from an application. These performance indicators are based on time-series data and not on events. A performance indicator must also vary over time to have meaningful analytical data.

IBM Training

Identifying good metrics to monitor

- There is no wrong answer to this question
- Analytics server identifies the best metrics it gets
  - Data availability
  - Data that has features that can be learned and anticipated
- Metrics can be filtered out
- You have metric budget
  - Can monitor only 250,000 metrics in a single analytics server topic
  - Can have up to 1,000,000 metrics across four analytics server topics
- Can provide only guidelines to give best possible results

Identifying good metrics to monitor

Predictive Insights monitors an alarm on any time-based data. The system takes whatever data you provide and selects the best key performance indicators. The concept of best is determined by having reliable data streams to analyze plus having features in the data that can be learned and anticipated. Data, which is either flat all the time or has only occasional spikes, is filtered out of the analysis because it provides little predictive information. A production system is limited by the number of metrics it can do a full analysis on. You can have only 250,000 metrics on a single analytics server. The biggest challenge in the analysis is to determine what, if any, relationships exist between all these metrics.

Collection rate and availability of data

- Is the data collected regularly
  - Is the data available 24/7
  - Missing data shows up in Predictive Insights GUI
- How often is the data collected
  - Data collection rate should align with a reasonable aggregation interval

Collection rate and availability of data

When selecting metrics, you must understand how and when the data is being collected. Is the data collected around the clock and can you expect it to be collected consistently? Is the data that is collected at regular intervals accessed by the Predictive Insights extraction process? These two questions are important to ask so that you can develop a useful statistical model of the data's variability and warn operators about anomalous trends.

IBM Training

Other data collection issues

- Late data
  - Data that takes a long time to arrive at data source
  - Late data is dropped if not retrieved by extraction call
    - Select data between ***<now>*** and ***<now – aggr. interval>***
  - Predictive Insights has latency property that gives data time to arrive
    - Select data between ***<now – latency>*** and ***<now – latency – aggr. interval>***
    - Latency delays the generation of alarm
- Collected data
  - 15-minute data that is added only to data source hourly
    - Most of the collected data is lost with aggregation interval of 15 minutes
  - Latency needs to be set to 60 minutes
    - Prevents timely alarms

Other data collection issues

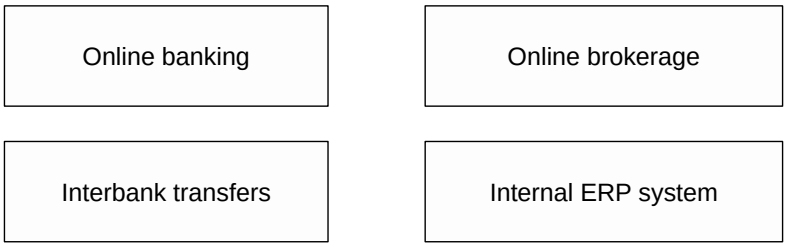
Late data is when data arrives late to the data source, which might be a database or CSV flat file. The concern is a delay from when it is collected to when it is written to the data source that Predictive Insights extracts data from. By default, the system calls the data source and asks for the data that was collected between now and one previous aggregation interval (for example, 5 minutes). If the data is late in arriving by that 5 minutes, then it is dropped from the system. It is not scored for anomalies, and it is not used in the statistical model. The solution provides a property that allows for latency to be included in the search of the data source. If you know that data arrives at the data source 15 minutes late, you can configure a 15-minute delay into the search that Predictive Insights uses. Instead of searching between now and one previous aggregation interval, Predictive Insights searches the past to ensure that you collect all the data. However, by including latency, you delay the appearance of an alarm by a time equal to that latency.

Another problem with some monitoring systems is that they collect several data points (for example, data points taken every 15 minutes) into one write statement to the data source that occurs every hour. If the aggregation interval is set to 15 minutes, then 75% of the data is lost. To prevent this data loss, you must set the latency to equal the amount of time between each write to the data source. In this case, the time is 60 minutes. This amount of latency can prevent the predictive nature of Predictive Insights from being a value.

IBM Training

Focus on a single service

- Try to limit scope during the early part of implementation
  Helps in deciphering an anomalous alarm
- Work with organization to select an important service whose degradation impacts customer satisfaction, revenue, and so on
  Find a service with recent outage and replay historical data

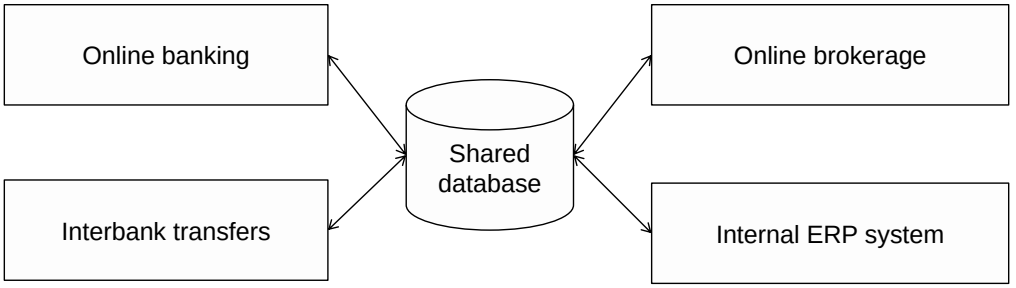


Focus on a single service

In the early implementation of Predictive Insights, you focus on a single service that is provided to either internal or external customers. By focusing on a single service, you can start the system quicker and analyze the anomalous alarms that are generated. Because the system looks for relationships between metrics, you might get alarms that detail these interactions. You can decipher these relationships when you consider only one service instead of many. Work with the customer to select a service that has high visibility or is important in generating revenue or maintaining business operations. If a service has a recent outage and historical data is available, you can use Predictive Insights to replay that data and determine other possible alarms in that situation.

Using shared resources

- Without application context, shared resource KPIs are confusing
- Alarms that are generated might not be applicable to monitored service



Using shared resources

When you monitor a resource shared by many applications, you must know that the alarms generated by the resource's KPIs might not relate to your application. In the previous slide, you could monitor only one service in an initial implementation. If you include a shared resource in your data model, you might need to take extra steps in deciphering an alarm. Predictive Insights does not have the complete picture of the other services that vie for that shared resource. When the metrics from all the services touch the resource are included in your model, you can see that your performance degradation is coming from another application.

Select meaningful metrics

- Engage system or application experts
- Every KPI in model is evaluated and can raise an alarm
  - Is an alarm on that KPI interesting
  - Is application expert interested if KPI was anomalous
- Rule of thumb
  - If you do not want to see an alarm on a KPI, do not include in the model

Select meaningful metrics

Monitoring systems might collect metrics that are not of interest. An application expert might be interested only in a subset of metrics that are collected for the server. By engaging these experts and getting this list of metrics, you limit the size of the data model and improve performance. You might also limit the alarms generated by noninteresting metrics. If you do not want to see an alarm on a particular metric, do not include it in the model.

Avoid metrics that mirror each other

- Consider MEMORY_PCT_USED and MEMORY_PCT_FREE
  MEMORY_PCT_USED + MEMORY_PCT_FREE = 100:
- If one is anomalous, the other is also
  With Node Consolidation, this is one alarm but it is extraneous information
- Does it make sense to include both KPIs

Avoid metrics that mirror each other

Servers, and hence monitoring systems, often have metrics that are mirrors of each other. For example, memory percent that is used and memory percent free are two numbers that when added up always equal 100. If you have one value, you can calculate the other. The problem with mirrored metrics is that if one is anomalous, the other is too, which creates two alarms. While it is consolidated into a single alarm in OMNIbus, it creates extraneous data that can be distracting. Select one or the other, but not both. This step saves on your KPI budget and improves performance.

Avoid metrics that have same underlying value

- Consider TOTAL_MEMORY_KB and TOTAL_MEMORY_MB
  Metrics measure the same thing except in different units
- Metrics are anomalous together
  Select one to reduce the number of alarms

Avoid metrics that have same underlying value

Some metrics have the same underlying value but are represented by different units of measure, for example, total memory in kilobytes and total memory in megabytes. These metrics represent the same value. As with mirrored KPIs, these metrics create anomalous alarms at the same time. Select one to reduce the alarming and the resources that are needed to calculate the extraneous value.

Avoid static values

- Typically, the same value for several months
- Not useful in the 28-day window that is used by Predictive Insights
- Examples
  - STATUS
  - SYSTEM_VERSION
  - NUMBER_OF_PROCESSORS
  - TOTAL_MEMORY

Avoid static values

Monitoring systems also collect metrics that do not change much with time. These metrics can have the same value for months and are useless with Predictive Insights 28-day training window. Some of these metrics include the total installed memory on the server, the number of processors on the server, and version information and system status. If these static values are included in a model, Predictive Insights excludes them from analysis because they do not vary.

Avoid minimum and maximum values

- Consider MEMORY_MIN and MEMORY_MAX
- They often reflect a value because the server or process was started
- Not likely to change often
- When they do change, decide whether you want an alarm

Avoid minimum and maximum values

Minimum and maximum memory values often reflect a value that was measured since the process or server was started. If so, then the value is not likely to change often. A maximum value climbs slowly and eventually stops at a value that might not change again until the process it represents is restarted. You might also consider whether you want an alarm when they change, which reflects that restart.

Avoid disk utilization

- Disk metrics are typically not good indicators of service quality
  - Moving large files between servers
  - Swap space utilization
  - Disk optimization processes
- Best monitored with a static threshold
  - Disk 95% full

Avoid disk utilization

Another set of metrics to avoid are measures of disk utilization. A hard disk does not typically give good indications of service quality. The work a disk drive does includes both supporting an application, the operating system as well as its own optimization processes. These processes occur at random times and can lead to alarms that are hard to decipher. Disks are best monitored by static thresholds that measure values, like how full the disk is.

IBM Training

Measure service quality

- Identify KPIs that are good indicators of service quality and availability
- Engage experts
- Good indicators of service quality
  - Response times
  - Error rates
  - Dropped packets

Measure service quality

Predictive Insights works best when it gets metrics that measure the quality and availability of a service. These types of indicators are holistic, and are eventually what customers experience. When Predictive Insights works with these types of metrics, it provides a warning about potential system problems before you reach the threshold where customer satisfaction is affected. A good idea is to engage solution experts in this effort to find what customers are also looking at. Response times are always good indicators of service quality, as are error rates and dropped packets.

IBM Training

Problems with monitoring processors individually

- Graph for an individual CPU is often random
  - OS decides which CPU is used, often not predictable
  - An alarm on a single processor might not mean anything
- Graph for the overall CPU usage is better
  - Often seasonal and predictable
  - If the overall usage is not fitting the expected norm, something changed

Problems with monitoring processors individually

With servers with multiple cores and multiple processors, do not monitor each individual processor on a server. The operating system decides which processor is used for processing, and it is not predictable which processor is tasked. If you receive an alarm on a single processor, you might not have any problems. The focus is on the overall processor usage of a server to determine anomalies. The processor utilization is often seasonal and can be included in the predictions that Predictive Insights creates. If the overall processor utilization changes from this expected norm, then something changed about that server.

IBM Training

Avoid resources that are used for testing

Resources such as Test servers do not make for ideal analysis

- Today, they are a test system for X
- Next week, they are a test system for Y
- There is no consistent pattern that emerges from the servers

Avoid resources that are used for testing

Ensure that you filter out resources that are used for testing. These test devices are not aligned with production. This inconsistency from week to week can lead to unnecessary alarms.

IBM Training

Suggested metrics to measure

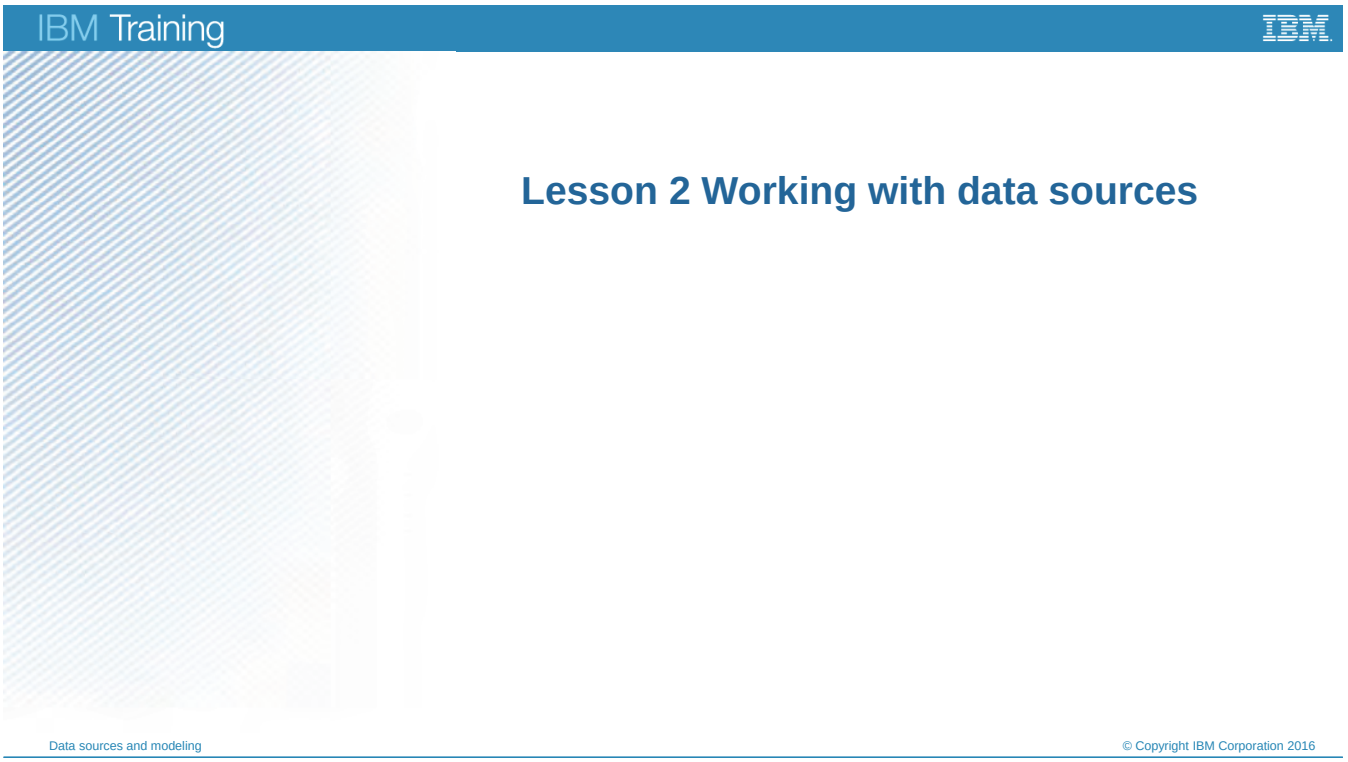
- Refer to the information on developerWorks
  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20SmartCloud%20Analytics%20-%20Predictive%20Insights/page/IBM%20SmartCloud%20Analytics%20-%20Predictive%20Insights
- Document on suggested metrics to include in analysis
- Off-The-Shelf data models for various IBM tools

  - ITCAM.pamodel: A model containing a subset of ITCAM KPIs.
  - ITM.pamodel: A model containing a subset of ITM KPIs.
  - WAREHOUS.pamodel: A model containing a subset of IBM ITM and ITCAM tables.
  - omegamon.pamodel: A model containing a subset of zOS OMEGAMON KPIs.
  - robotics.pamodel: A model containing a subset of ITCAM for Transactions KPIs.

Suggested metrics to measure

A comprehensive database of best metrics to use on an application-by-application basis is not available. However, some suggested metrics are available for the IBM Application Performance Manager suite of tools. On the IBM developerWorks website, you can find metrics that are used by Tivoli Monitoring for Linux and Windows servers, and metrics that are used in Tivoli Composite Application Manager. You can measure web response time, WebSphere® performance, and robotic response time.

Lesson 2 Working with data sources



This lesson is about data sources that you can use for extracting monitoring data. You learn which data source types are supported and how to arrange the data within them. You also learn about problems you might encounter when working with certain types of data sources.

What is a data source

- A connection to a database
  - Uses JDBC and a unique set of credentials
  - Can view only the tables available to that user
  - If other tables exist with useful data, you might need to create second data source
- A directory of flat files
  - Flat files must meet specific file naming convention
  - Directory must be accessible to analytics server through a path name
- A data model can have many data sources

What is a data source

In the preceding slides, the term data source refers to where Predictive Insights extracts monitoring data from the service that is analyzed. Currently, for Predictive Insights, a data source is a database connection or series of flat files in a directory. A database data source is where you make a connection to a single database through JDBC by using a user name and password. You can use this connection to extract data from any of the tables that this specific user can view. If other important tables are in that database or data that is on a different one, you must configure another data source. A file system data source is where a series of specially named files are in a directory that the analytics server has local access to and can extract data from. A data model can have as many data sources as necessary to supply Predictive Insights with the needed metrics.

IBM Training

Data source examples (wide)

- Databases with performance data

EndTime	node	dataSourceType	AvgPercMemUsed	AvgLoad
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	21.932495	6
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	45.12558	0
2013-04-03T08:10:00-0000	ethcsc-bonutti_research	CPULoad	20.754591	12
2013-04-03T08:10:00-0000	ethcsc-mtonilxc23w-bbcab1-36	CPULoad	23.803034	5
2013-04-03T08:10:00-0000	rtrcsc-chtnilxchrc-fo09-30	CPULoad	42.52896	1
2013-04-03T08:10:00-0000	svrvmw-chtnilxcvs0-minerva-app-sec	CPULoad	27.963974	7
2013-04-03T08:10:00-0000	ethcsc-arthilxc01w-cxr_01-32	CPULoad	17.637367	4
2013-04-03T08:10:00-0000	rtrcsc-ltfdilxchrd-lwv03-12	CPULoad	38.332874	0
2013-04-03T08:10:00-0000	svrsun-mtonilxc-voipm6-15	CPULoad	30.28202	1

- Fla

```
#EndTime,node,dataSourceType,AvgPercMemUsed,AvgLoad
2013-04-03T08:10:00-0000,edbgilxc01w,CPULoad,21.932495,6
2013-04-03T08:10:00-0000,ethbcd-chtnilxc11w-102-05-41,CPULoad,45.12558,0
2013-04-03T08:10:00-0000,ethcsc-bonutti_research,CPULoad,20.754591,12
2013-04-03T08:10:00-0000,ethcsc-mtonilxc23w-bbcab1-36,CPULoad,23.803034,5
2013-04-03T08:10:00-0000,rtrcsc-chtnilxchrc-fo09-30,CPULoad,42.52896,1
2013-04-03T08:10:00-0000,svrvmw-chtnilxcvs0-minerva-app-sec,CPULoad,27.963974,7
2013-04-03T08:10:00-0000,ethcsc-arthilxc01w-cxr_01-32,CPULoad,17.637367,4
2013-04-03T08:10:00-0000,rtrcsc-ltfdilxchrd-lwv03-12,CPULoad,38.332874,0
2013-04-03T08:10:00-0000,svrsun-mtonilxc-voipm6-15,CPULoad,30.28202,1
```


Data source examples (wide)

Regardless whether the data is in a database or in flat files, Predictive Insights expects the data to be in a rigorous format. This format must include a time stamp, a resource name, and a series of values. These values are positioned such that the column represents a specific measurement. This data organization is referred to as a wide format since the values in each column represent a specific metric.

IBM Training

Data source examples (skinny)

- Database

EndTime	node	dataSourceType	Metric	Value
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	AvgPercMemUsed	21.9325
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	AvgLoad	6
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	CPULoad	AvgPercMemUsed	23.80303
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	CPULoad	AvgLoad	5
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	AvgPercMemUsed	45.12558
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	AvgLoad	0
2013-04-03T08:10:00-0000	ethcsc-arthilxc01w-cxr_01-32	CPULoad	AvgPercMemUsed	17.63737
2013-04-03T08:10:00-0000	ethcsc-arthilxc01w-cxr_01-32	CPULoad	AvgLoad	4
2013-04-03T08:10:00-0000	ethcsc-bonutti_research	CPULoad	AvgPercMemUsed	20.75459
2013-04-03T08:10:00-0000	ethcsc-bonutti_research	CPULoad	AvgLoad	12
2013-04-03T08:10:00-0000	rtrcsc0chtnilxchrc-fo09-30	CPULoad	AvgPercMemUsed	42.52896
2013-04-03T08:10:00-0000	rtrcsc0chtnilxchrc-fo09-30	CPULoad	AvgLoad	1
2013-04-03T08:10:00-0000	rtrcsc-ltfdilxchrd-lwv03-12	CPULoad	AvgPercMemUsed	38.33287
2013-04-03T08:10:00-0000	rtrcsc-ltfdilxchrd-lwv03-12	CPULoad	AvgLoad	0
2013-04-03T08:10:00-0000	svrsun-mtonilxc-voipm6-15	CPULoad	AvgPercMemUsed	30.28202
2013-04-03T08:10:00-0000	svrsun-mtonilxc-voipm6-15	CPULoad	AvgLoad	1
2013-04-03T08:10:00-0000	svrvmw-chtnilxcvs0-minerva-app-sec	CPULoad	AvgPercMemUsed	27.96397
2013-04-03T08:10:00-0000	svrvmw-chtnilxcvs0-minerva-app-sec	CPULoad	AvgLoad	7

- Flat File

```
#EndTime,node,dataSourceType,Metric,Value
2013-04-03T08:10:00-0000,edbgilxc01w,CPULoad,AvgPercMemUsed,21.932495
2013-04-03T08:10:00-0000,edbgilxc01w,CPULoad,AvgLoad,6
2013-04-03T08:10:00-0000,ehtcsc-mtonilxc23w-bbcab1036,CPULoad,AvgPercMemUsed,23.803034
2013-04-03T08:10:00-0000,ehtcsc-mtonilxc23w-bbcab1036,CPULoad,AvgLoad,5
2013-04-03T08:10:00-0000,ethbcd-chtnilxc11w-102-05-41,CPULoad,AvgPercMemUsed,45.12558
2013-04-03T08:10:00-0000,ethbcd-chtnilxc11w-102-05-41,CPULoad,AvgLoad,0
2013-04-03T08:10:00-0000,ethcsc-arthilxc01w-cxr_01-32,CPULoad,AvgPercMemUsed,17.637367
2013-04-03T08:10:00-0000,ethcsc-arthilxc01w-cxr_01-32,CPULoad,AvgLoad,4
2013-04-03T08:10:00-0000,ethcsc-bonutti_research,CPULoad,AvgPercMemUsed,20.754591
2013-04-03T08:10:00-0000,ethcsc-bonutti_research,CPULoad,AvgLoad,12
2013-04-03T08:10:00-0000,rtrcsc0chtnilxchrc-fo09-30,CPULoad,AvgPercMemUsed,42.52896
2013-04-03T08:10:00-0000,rtrcsc0chtnilxchrc-fo09-30,CPULoad,AvgLoad,1
2013-04-03T08:10:00-0000,rtrcsc-ltfdilxchrd-lwv03-12,CPULoad,AvgPercMemUsed,38.332874
2013-04-03T08:10:00-0000,rtrcsc-ltfdilxchrd-lwv03-12,CPULoad,AvgLoad,0
2013-04-03T08:10:00-0000,svrsun-mtonilxc-voipm6-15,CPULoad,AvgPercMemUsed,30.28202
2013-04-03T08:10:00-0000,svrsun-mtonilxc-voipm6-15,CPULoad,AvgLoad,1
2013-04-03T08:10:00-0000,svrvmw-chtnilxcvs0-minerva-app-sec,CPULoad,AvgPercMemUsed,27.963974
2013-04-03T08:10:00-0000,svrvmw-chtnilxcvs0-minerva-app-sec,CPULoad,AvgLoad,7
```

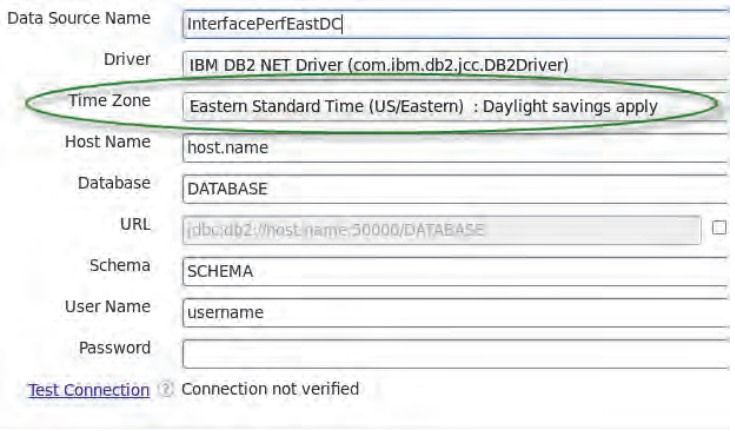

Data source examples (skinny)

An alternative to a wide format is referred to as skinny. Here, a pair of columns is used to represent a series of different metrics. One column declares the metric name and the other the metric value. Predictive Insights can mediate some skinny formats but others can require preprocessing.

IBM Training

Time zones

- Data from multiple data sources *must* be coordinated to a common time frame
- Mediation tool allows the selection of time zone for each data source
  Assumes that time stamps are coordinated to single time zone within a table or flat file



Time zones

For Predictive Insights to coordinate its data extraction and present data correctly in the GUI, the time zone of each data source's time stamp must be available. When you select this time zone, all the tables (or files in the directory) are coordinated with this same time zone. If a variation exists between time zones, you must create a data source for each one.

IBM Training

Comma-separated value (CSV) files

- The format of the file name is important
  - <*DataGroupName*>-<*StartTimestamp*>-<*EndTimestamp*>.csv
  - Mediation client requires the file name to be in this order
  - The <*DataGroupName*> is like a table name
  - Time stamps must be regex-readable format and consistent in format
- Each file must have column name header on first line
  - Column header must be the same between all files with the same <*DataGroupName*>

```
EndTime,Resource,AvgResponseTime,MaxResponseTime
1370088600000,"router-sw49.tut.com",6,6
1370088600000,"router-ne10.tut.com",7,7
1370088600000,"router-ne19.tut.com",32,32
1370088600000,"router-nw57.tut.com",6,6
1370088600000,"ethernet-csx546.tut.com",4,4
```

- Files *mu*

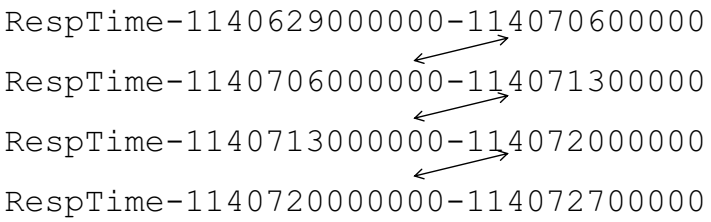
Comma-separated value (CSV) files

In Predictive Insights, flat file comma-separated value files are like tables in a database. The file system directory is like a database, and the files in the directory are tables. The files must have a specific naming convention. This naming convention starts with a DataGroupName, which might be a table name. Also, part of the naming convention is a start and end time stamp that declares the capture time for the metrics inside the file. This StartTimestamp and EndTimestamp must use a consistent time stamp convention, for example, 20140401120000 for April 1, 2014 at 12 noon. Inside each file, you must define a header line that has the names of the columns for the CSV data to follow. This header must be consistent for all the files that use the same DataGroupName. Finally, the analytics server accesses these files with simple Linux OS system commands to retrieve the data. The files must be locally accessible, and the analytics server must have the rights to access them.

IBM Training

Working with historical data files

- Often, you receive historical data as several large files
  - Represents several data table extractions
  - Represents many weeks of data
- Data files should align start and end time stamps
  - Overlapping time stamps is not supported



Working with historical data files

You can analyze historical data and produce anomalies after enough data is used for training. This feature is useful in proof-of-concept situations where you analyze historical data and determine what alarms PI might have generated. These historical files are typically an extraction from a database and might represent many weeks of data.

If you receive a set of files representing a specific time frame; for example, four files, each with one week of data, you must set the start and end time stamps in the file name. These dates must align that the end time of one file is the start time of the next. You must have continuity between files, which means that the end time of one file name must be used as the start time of the next file. Overlapping of time stamps is not supported. If one file is extracted beyond the start time of the second file, the extraction of the second file begins from the last extracted time of the first file.

IBM Training

Databases

- Preferred method for data extraction to Predictive Insights
- Uses Java Database Connectivity (JDBC)
  Must provide JDBC drivers for non-DB2 databases
- Predictive Insights expects structured data across one or more tables

EndTime	node	dataSourceType	AvgPercMemUsed	AvgLoad
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	21.932495	6
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	45.12558	0
2013-04-03T08:10:00-0000	ethcsc-bonutti_research	CPULoad	20.754591	12
2013-04-03T08:10:00-0000	ethcsc-mtonilxc23w-bbcab1-36	CPULoad	23.803034	5
2013-04-03T08:10:00-0000	rtrcsc-chtnilxchrc-fo09-30	CPULoad	42.52896	1
2013-04-03T08:10:00-0000	svrvmw-chtnilxcvs0-minerva-app-sec	CPULoad	27.963974	7
2013-04-03T08:10:00-0000	ethcsc-arthilxc01w-cxr_01-32	CPULoad	17.637367	4
2013-04-03T08:10:00-0000	rtrcsc-ltfdilxchrd-lwv03-12	CPULoad	38.332874	0
2013-04-03T08:10:00-0000	svrsun-mtonilxc-voipm6-15	CPULoad	30.28202	1

- Da

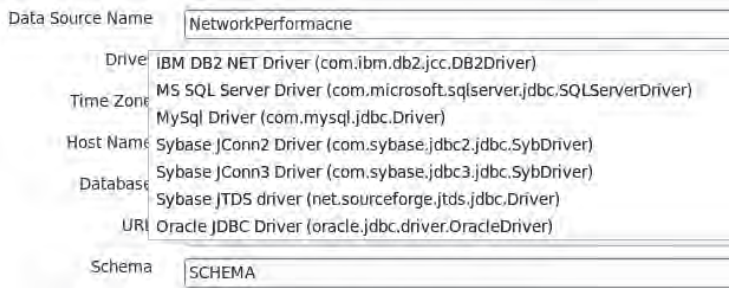
Databases

Databases are the preferred method for extracting data from data sources. Predictive Insights uses JDBC to connect to these data sources. If you need to connect to other databases to extract data, you must download the JDBC drivers from the appropriate location. The data in the database needs to be structured where you have a time stamp and host name columns and then a series of columns that represent metrics. If you have multiple tables, the metrics are coordinated by time stamps and host names.

IBM Training

Supported databases

- Current list of supported databases



- Non-DB2 database support process
  - Add JDBC driver to mediation tool
  - Add JDBC driver to Predictive Insights server
- List can be expanded but no guarantee of support
  - Potential issues with formatting of JDBC call

Supported databases

Predictive Insights has support for a series of popular databases through the data client mediation tool. If you need to connect to databases other than DB2, you need to download the required JDBC driver for that database. You must add the driver to both the data mediation client and analytics servers Java libraries. The list can be expanded to include other databases but there is no guarantee of support. Expanding the list is fairly easy. You might encounter potential issues with the formatting of the JDBC call and the way that the data is returned from the database.

Challenging data sources

- Database uses foreign keys for items
- Database does not support JDBC
- Additional levels of skinniness

Challenging data sources

Several data sources in the current release lack a specific metric schema, which is referred to as skinny data. Skinny data can be an issue with both database and flat file data sources. Another issue is monitoring tools that use foreign keys to reference data items, most notably host names. In some instances, the data source does not support JDBC calls.

IBM Training

Database uses foreign keys

- Mediation client cannot join tables
- Some monitoring tools use foreign keys in schema
  SolarWinds Orion uses foreign key for host name
- Option 1: Create view in the monitoring tool database
  Preferred option
- Option 2: Use external tool to combine data and output to a intermediate datasource
  Logstash has a plugin that can join tables and output to CSV

Database uses foreign keys

In some instances, the monitoring system uses a foreign key to represent various items in their schema. This foreign key is typically the host name. Unless you want the foreign keys in alarms, you need to resolve that key to its host name. Predictive Insights cannot join tables. You get permission to create a view in the database where the foreign key is. This method is the preferred option. The other option is to use a third-party tool, like InfoSphere DataStage or logstash, to combine the data and have the mediation tool search that application.

Database does not support JDBC calls

- Primitive or proprietary databases
- Export data
  - Flat files
  - Database that does support JDBC
- Requires the development of custom export code
  - Make sure that code has some self-monitoring
  - Make sure that housekeeping is included with database use

Database does not support JDBC calls

Some data sources do not support JDBC. In these instances, you might have an old database, or more likely, a proprietary system that can be accessed only with that vendor's language. In these instances, you need some tool to export data into either flat files or into a database that supports JDBC. The result is a need for middleware code development. In the past experiences with this type of data source, you need something to monitor this middleware daemon to keep it running. Consider the housekeeping details of this solution to ensure that old data is removed from the necessary locations.

Additional levels of skinniness

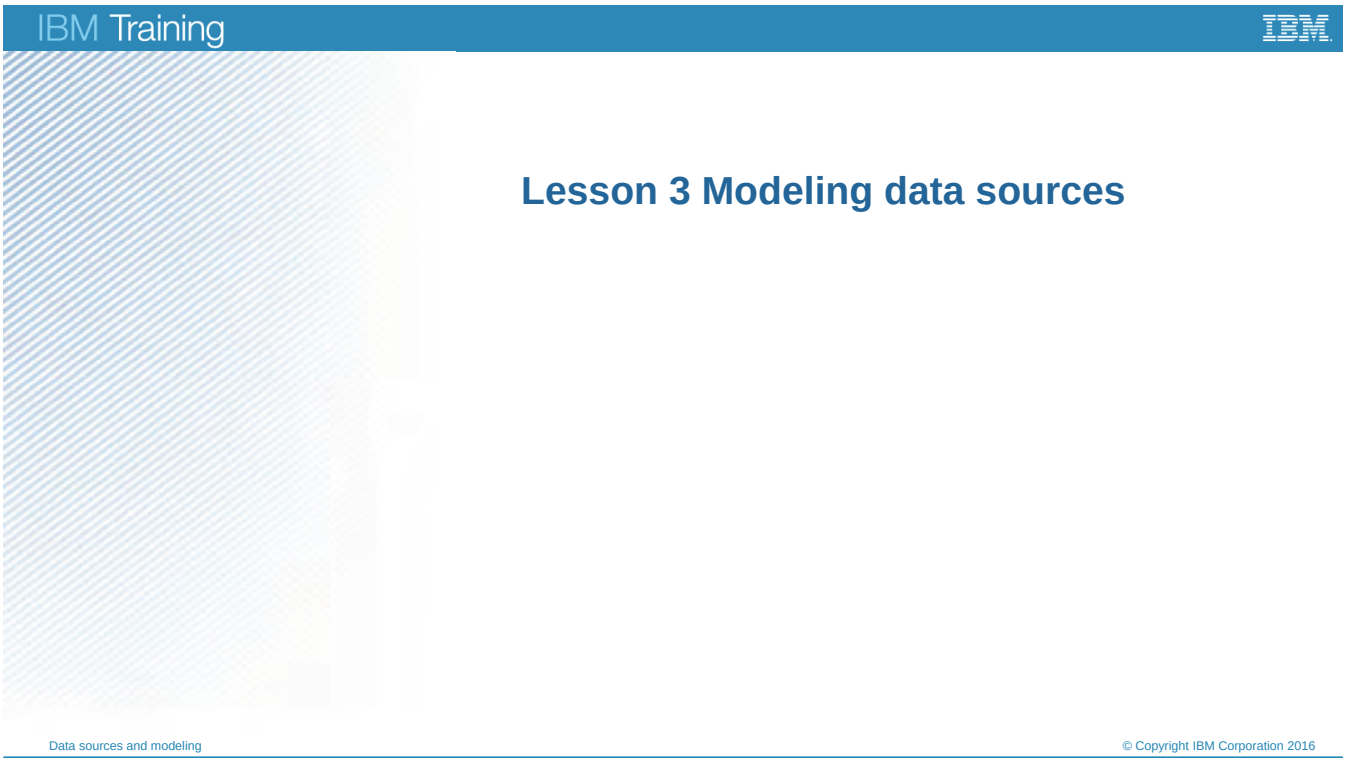
- Data source has more than one column defining the metric
  - dataSourceType contains both CPU and network metrics
  - Metric name is used both in CPU and network monitoring
- Mediation client can only handle one level of skinniness
- Option 1: Use external tool to break file into intermediate datasources
  - Logstash can filter each line and output them to separate CSV files

EndTime	node	dataSourceType	Metric	Value
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	AvgPercMemUsed	21.9325
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	AvgLoad	6
2013-04-03T08:10:00-0000	edbgilxc01w	NetUse	MaxLoad	124
2013-04-03T08:10:00-0000	edbgilxc01w	NetUse	AvgLoad	72
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	CPULoad	AvgPercMemUsed	23.80303
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	CPULoad	AvgLoad	5
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	NetUse	MaxLoad	210
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	NetUse	AvgLoad	100
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	AvgPercMemUsed	45.12558
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	AvgLoad	0
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	NetUse	MaxLoad	79
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	NetUse	AvgLoad	25

Additional levels of skinniness

Currently, the mediation client can support only one level of skinniness regarding a data source. Extra levels of skinniness can be included in a data source if there are metrics that have the same name but have different meanings. For example, the AvgLoad metric is used both for NetUse and CPULoad monitors. In their respective context, they mean different things and should not be combined. In this special case, you either need to rename the metric or parse the data source into separate files, one for CPULoad and the other for NetUse before attempting to mediate with Predictive Insights.

Lesson 3 Modeling data sources



In this lesson, you learn about the data mediation client, which you use to define the following items:

- Location of data sources
- The metrics that must be pulled from data sources
- Any manipulation or filtering of the data that must occur before it is analyzed

IBM Training

Data mediation tool

- Used to create a data model for the Predictive Insight server
  Only one data model per topic (most servers have only one topic)
- A tool for testing and modeling one or more data sources
  Data source is a single database connection or one file system directory
- Uses Java and is built on top of Eclipse-integrated development environment
- Can be installed on server or other client machine
  Linux or Windows
- Preferred installation is on Predictive Insights server
  Ensures that data sources are available to Predictive Insights
- Main purpose is to synchronize time and select metrics from each data source

Data mediation tool

When you installed Predictive Insights, one of the options was to install the data mediation tool. This tool is client software for creating a data model. A data model is an XML file that has all the details about your data sources and what you want to extract from them. A data model is defined for each topic that is defined for your installation. Most Predictive Insights implementations have only one topic. The data mediation tool is how you connect to data sources, test data extractions, and make specific decisions about what data to include in the model. The data model can have multiple data sources.

The tool is built on Java and runs inside the Eclipse-integrated development environment. You can install and run it on both Red Hat Linux and Windows servers. You might want to install the client on the same server as the Predictive Insights. This collocation ensures that connectivity to data sources is tested when modeling the data. If the mediation tool can reach the data, then the Predictive Insights server can also. The main purpose of this client is to determine the location of the data sources, synchronize time stamps, and select the metrics that you want to use. It creates an XML file of the information that is uploaded to the server. After the upload is complete, the tool is not needed unless changes to the model are necessary.

IBM Training

Metric groups

- A metric group is a logical grouping of metrics from tables that are defined in the data source
- A subset of columns from tables that have monitoring data
  - Time stamp
  - Resource name
  - More resource identifiers (for example, interface number)
  - KPIs that are included in analysis

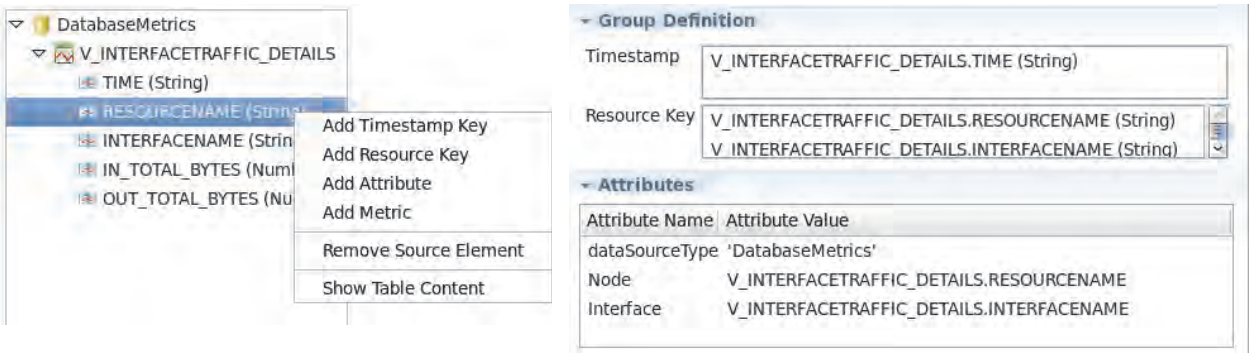
Metric groups

To help in the modeling process, Predictive Insights creates a concept called metric groups. Metric groups are created when you manipulate data sources to select only the data that you want to analyze. A metric group is a subset of data that the data source has and typically aligns with an existing table. This subset of data includes a time stamp and a resource name. It also includes a series of KPIs that are included in the data source. It can also include additional information to further identify details about the resource, for example, interface numbers.

IBM Training

Metric group requirements

- A metric group must have time stamp key
- A metric group must have resource key
  - Resource key can be a combination of one or more attributes
  - Adding a resource key requires addition of an attribute
- A metric group must have one or more metrics
- A metric group must define a **node** as an attribute

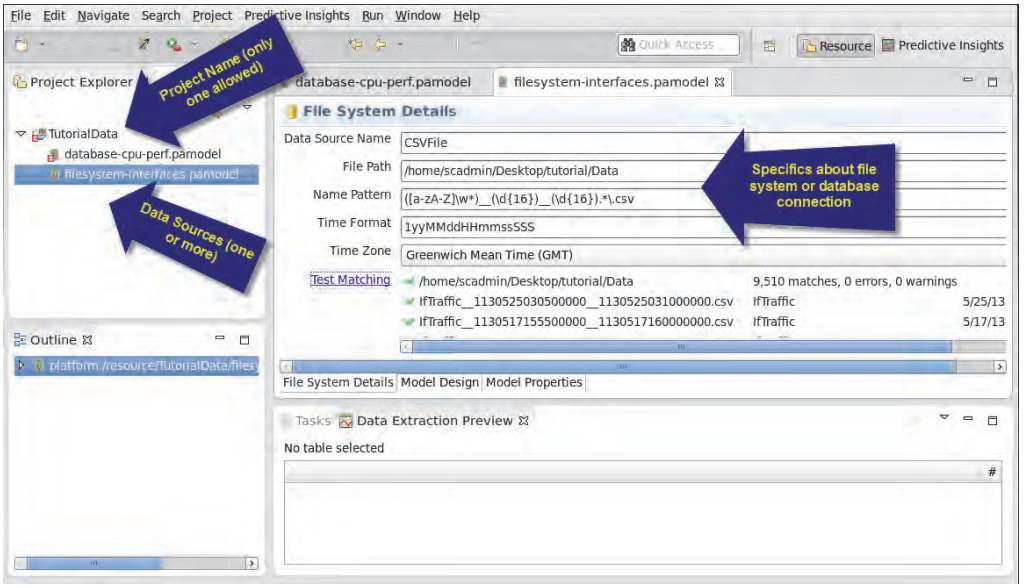


Metric group requirements

When you create a metric group, you must define a time stamp key. This key is typically referenced to a column in the table that the metric group was created from. However, the time stamp key can be built from several columns of data, if necessary. The time stamp key must create a time stamp that, at a minimum, is to the nearest minute. A metric group must also define a resource key. Like with the time stamp key, it is typically referenced to a column in the data table that the metric group is defined from. The key can also be built from using multiple columns of data. For example, the key can be built from the host name column and an interface column. If you build the time stamp or resource key from multiple columns of data, each column must be defined as an attribute in the metric group. Finally, a metric group must have one or more metrics that are defined in it. An important aspect of Predictive Insights is its ability to consolidate alarms around a node. If multiple alarms are generated by various metrics that are associated to a specific node, they are collected into one alarm. This consolidation is predicated on having an attribute that is named node, which is typically the host name.

IBM Training

Data mediation tool data sources

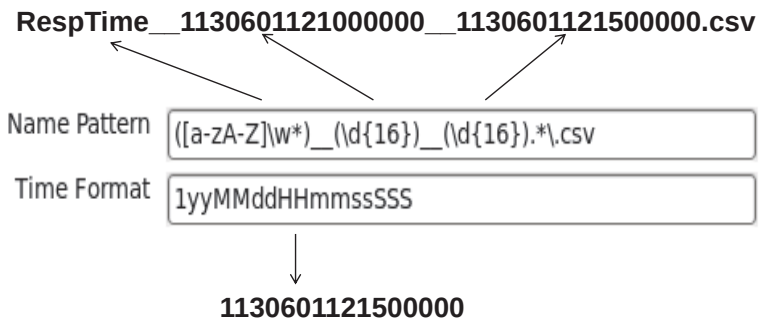


Data mediation tool data sources

Before you define a metric group, you must create a data source. When you initialize the data mediation tool, you define a project name, which is unique. You have one project per topic in Predictive Insights. Each project can have one or more data sources. For each data source you create, you define it as either a database or file system resource. By choosing one, for example a file system resource, you must complete the details of how to locate and connect to data inside it. In the example, you see a file system data source. You must provide a file path to the directory where the CSV files exist and information about how to interpret the file name that each file has.

Name pattern and time format of CSV files

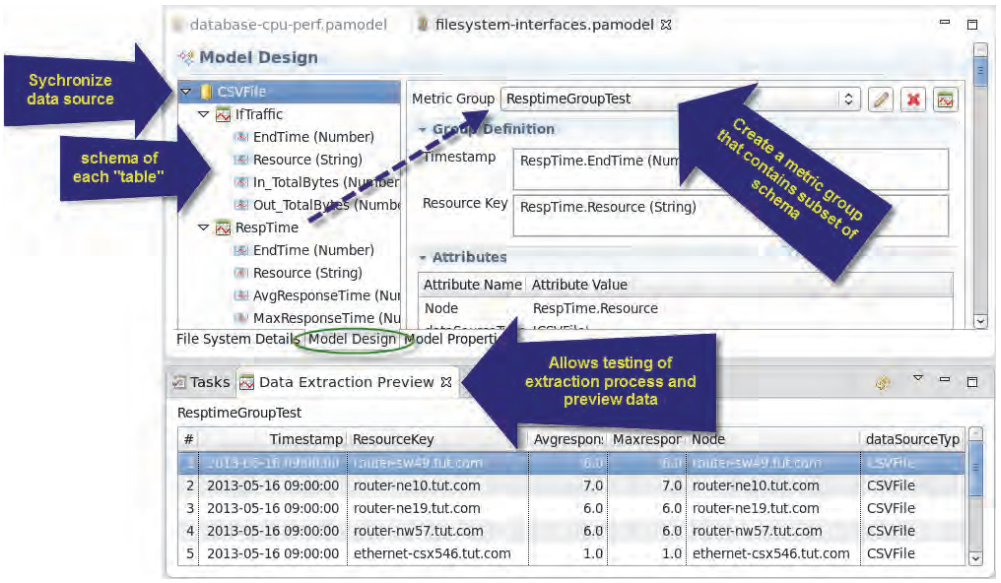
- Regular expression that is used to extract the three parts of CSV file name
  <DataGroupName>-<StartTimestamp>-<EndTimestamp>.csv
- Time format that is used to understand the time stamps that are extracted from file name
  Uses SimpleDateFormat Java class



Name pattern and time format of CSV files

CSV files must use a specific naming convention. The file system data source details page is where you interpret the naming convention. The Name Pattern is an argument field that must be configured with a regular expression to align with the naming convention that you selected, and from it, extract three pieces of information. The data includes the DataGroupName, the StartTimeStamp, and EndTimeStamp. The Time Format field allows the software to further dissect the time stamps that were retrieved by the Name Pattern regular expression. It defines the format of the retrieved data to determine the date and time of the StartTimeStamp and EndTimeStamp.

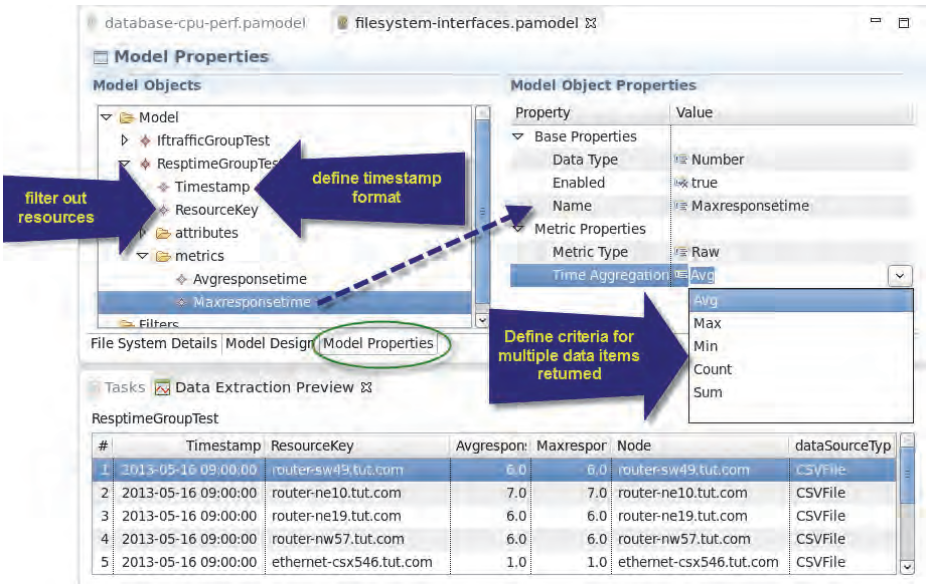
Data mediation tool modeling data



Data mediation tool modeling data

After providing the details of how you connect to a data source, regardless if it is a file system or database, you design the framework of your data model. First, you synchronize the data. During synchronization, the mediation tool looks into the data source and extracts its schema. This synchronization includes the name of each table that is accessible and the column names (and data types) that are defined. If you synchronize a file system data source, the mediation tool treats the DataGroupName as a table name. You then select data from the displayed tables, which ones you want to create metric groups from. In most cases, the metric group is a subset of columns from a table you select. After you select the columns of data and adjust it, you can preview the data that is extracted from those data sources.

Data mediation tool model properties



Data mediation tool model properties

When you create your metric groups, you might have to use the Model Properties tab to make specific adjustments to the data. These specific alterations are as follows:

- Ensuring that you can interpret the time stamp in the data source
- Filtering out resources that you do not want in your analysis
- Selecting the aggregation method when an extraction returns multiple lines of data for each resource, for example, a 5-minute aggregation of 1-minute data.

IBM Training

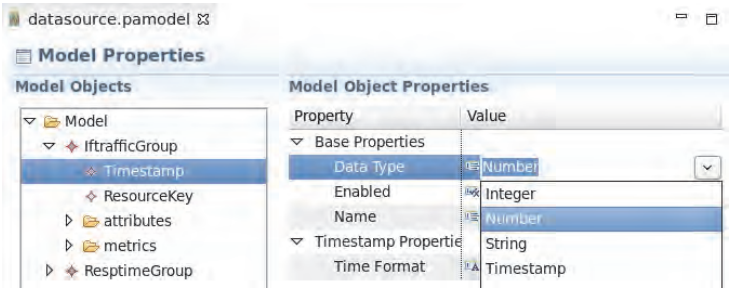
Addressing time stamps inside a file

- Check time stamp format for each metric group

Epoch time in milliseconds

```
EndTime,Resource,AvgResponseTime,MaxResponseTime
1370088600000,"router-sw49.tut.com",6,6
1370088600000,"router-ne10.tut.com",7,7
1370088600000,"router-ne19.tut.com",32,32
1370088600000,"router-nw57.tut.com",6,6
1370088600000,"ethernet-csx546.tut.com",4,4
```

- Set Data Type and pc
  Time Format uses SimpleDateFormat Java class



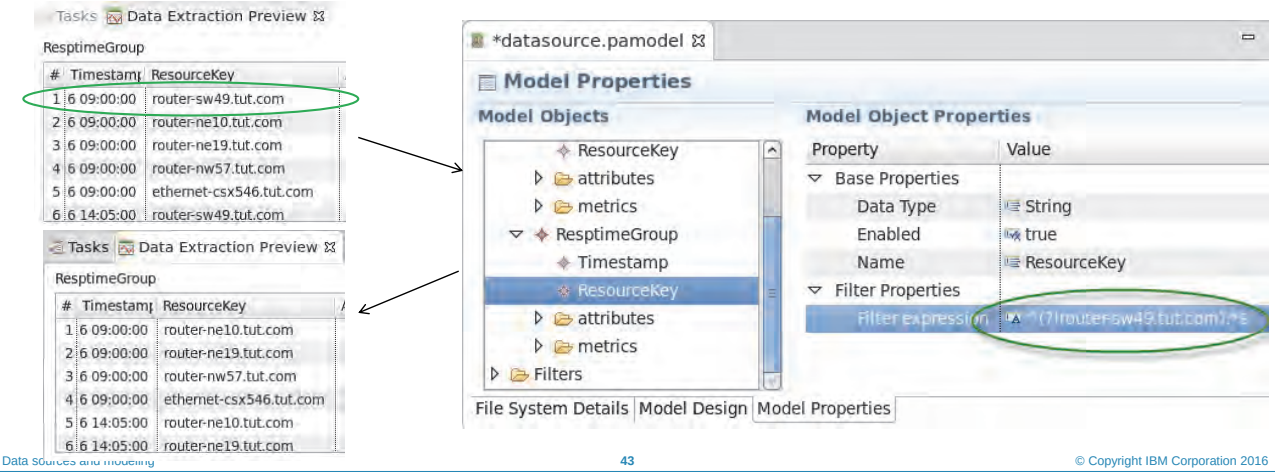
Addressing time stamps inside a file

When modeling the data source and the metric groups that you create from them, you must ensure that the mediation tool can interpret the time format for each metric group. You need to set the data type for the time stamp key that you created. If the time stamp is not a UNIX epoch time stamp, you need to use the Time Format field and the naming rules in the SimpleDataFormat Java class to decipher the time. Use the Integer data type for 10-digit epoch times (time stamps that reflect seconds) and the Number data type for 13-digit epoch times (time stamps that reflect milliseconds).

IBM Training

Filtering resources

- Optional filter to include or exclude resources that are based on the resource key
  - **Defines what to include**
  - Must be done for each metric group
- Use NOT statement to exclude resources

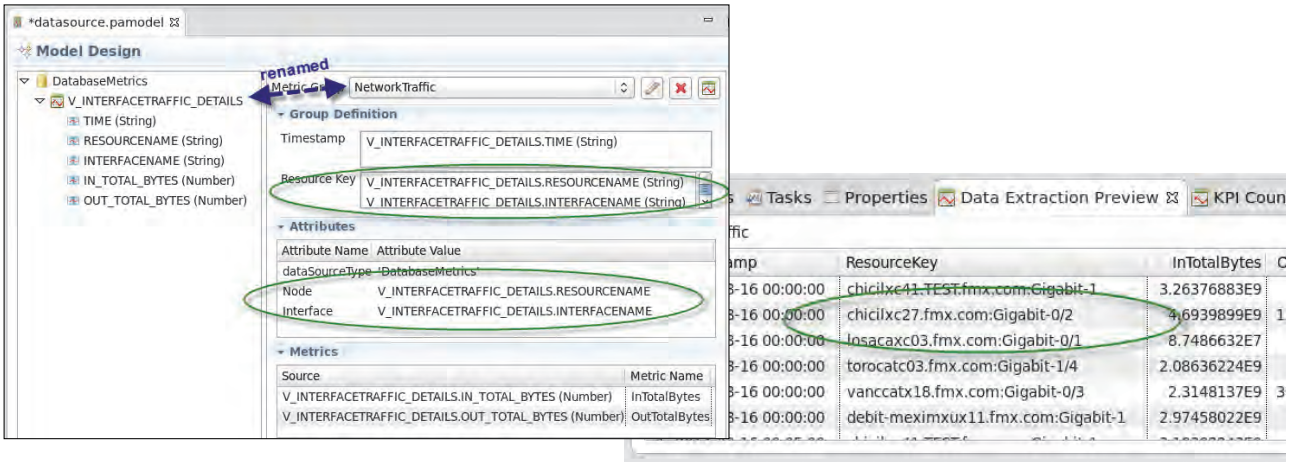


Filtering resources

Sometimes, you must remove resources from the data to prevent them from being analyzed and triggering alarms. For example, test devices might generate alarms because their use is unpredictable. You use the Model Properties tab to define a filter expression on the resource key for a metric group. The regular expression defines what resources to include in the analysis. Because you want to exclude resources from the analysis, you want to create a statement that defines what not to include. Whenever you create a filter expression, you must test its effectiveness by using the preview function that the mediation tool has. This filter expression is only effective for the metric group that you created. It might need to be replicated to all the other metric groups that have these unwanted resources. You are filtering the resource key. If the resource key is a combination of multiple columns, then you might have to adjust the regular expression.

Combining data to create unique resource key

- Resource key is what is analyzed and displayed in GUI
- Any item that is defined in resource key must have associated attribute
- Example: Combine RESOURCENAME with INTERFACENAME

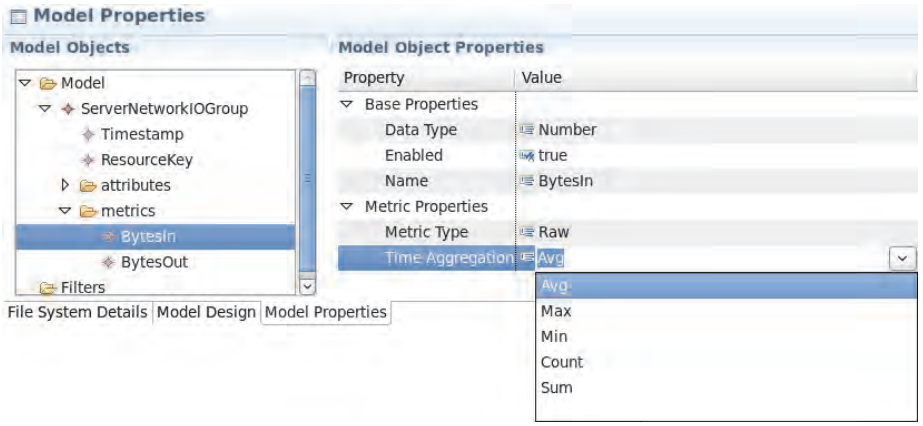


Combining data to create unique resource key

Sometimes, data refers to a subcomponent of a node name. For example, the data might be subdivided by a specific attribute that is associated to that device such as interfaces. If you do not create a more specific resource key, the data on all devices interfaces might be averaged together. You can use the mediation tool to define a resource key that is more specific than only a host name. You can add data columns to the resource key. For every column that you define as a resource key, you must have an associated attribute in the Attributes table.

Aggregating results of data queries

- Database and flat file queries might return multiple metric values for each resource
- Need to decide how to resolve to a single value
  AVG (default), MIN, MAX, COUNT, SUM



Aggregating results of data queries

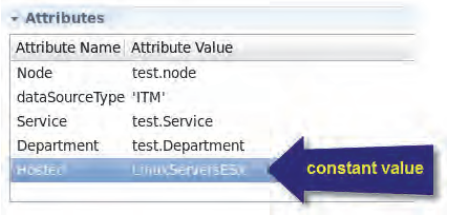
If more than one result is returned for your metric, you use the Model Properties tab to determine the time aggregation that you want to consolidate that date into one value. This need arises when the data in a data source is aggregated more often than the aggregation interval that is used by the analytics server. For example, the monitoring tool has a 1-minute data and the analytics server aggregates every 5 minutes. The default setting is to average all the items that are retrieved from the query; for example, add all 1-minute data items and then divide by 5. However, you can select from minimum and maximum values, the summation of all the values, or a count. In this example, the count is 5.

IBM Training

Adding attributes to metrics

- Attributes allow additional context to be added to a metric and the anomalies generated
  - Data source data
  - Constants
- Attributes included in alarm and viewable in Predictive Insights GUI.

EndTime	node	dataSourceType	Metric	Value	Service	Department
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	AvgPercMemUsed	21.9325	monitoring	31C
2013-04-03T08:10:00-0000	edbgilxc01w	CPULoad	AvgLoad	6	monitoring	31C
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	CPULoad	AvgPercMemUsed	23.80303	online-bank	B22
2013-04-03T08:10:00-0000	ehtcsc-mtonilxc23w-bbcab1036	CPULoad	AvgLoad	5	online-bank	B22
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	AvgPercMemUsed	45.12558	online-bank	B22
2013-04-03T08:10:00-0000	ethbcd-chtnilxc11w-102-05-41	CPULoad	AvgLoad	0	online-bank	B22
2013-04-03T08:10:00-0000	ethcsc-arthilxc01w-cxr_01-32	CPULoad	AvgPercMemUsed	17.63737	brokerage	4T3
2013-04-03T08:10:00-0000	ethcsc-arthilxc01w-cxr_01-32	CPULoad	AvgLoad	4	brokerage	4T3
2013-04-03T08:10:00-0000	ethcsc-bonutti_research	CPULoad	AvgPercMemUsed	20.75459	lab-service	31C
2013-04-03T08:10:00-0000	ethcsc-bonutti_research	CPULoad	AvgLoad	12	lab-service	31C
2013-04-03T08:10:00-0000	rtrcsc0chtnilxchrc-fo09-30	CPULoad	AvgPercMemUsed	42.52896	exchange	4T3
2013-04-03T08:10:00-0000	rtrcsc0chtnilxchrc-fo09-30	CPULoad	AvgLoad	1	exchange	4T3
2013-04-03T08:10:00-0000	rtrcsc-ltfdilxchrd-lwv03-12	CPULoad	AvgPercMemUsed	38.33287	exchange	4T3
2013-04-03T08:10:00-0000	rtrcsc-ltfdilxchrd-lwv03-12	CPULoad	AvgLoad	0	exchange	4T3
2013-04-03T08:10:00-0000	svrsun-mtonilxc-voipm6-15	CPULoad	AvgPercMemUsed	30.28202	internal-voip	31C
2013-04-03T08:10:00-0000	svrsun-mtonilxc-voipm6-15	CPULoad	AvgLoad	1	internal-voip	31C
2013-04-03T08:10:00-0000	svrvmw-chtnilxcvs0-minerva-app-sec	CPULoad	AvgPercMemUsed	27.96397	exchange	4T3
2013-04-03T08:10:00-0000	svrvmw-chtnilxcvs0-minerva-app-sec	CPULoad	AvgLoad	7	exchange	4T3



Data sources and modeling 46 © Copyright IBM Corporation 2016

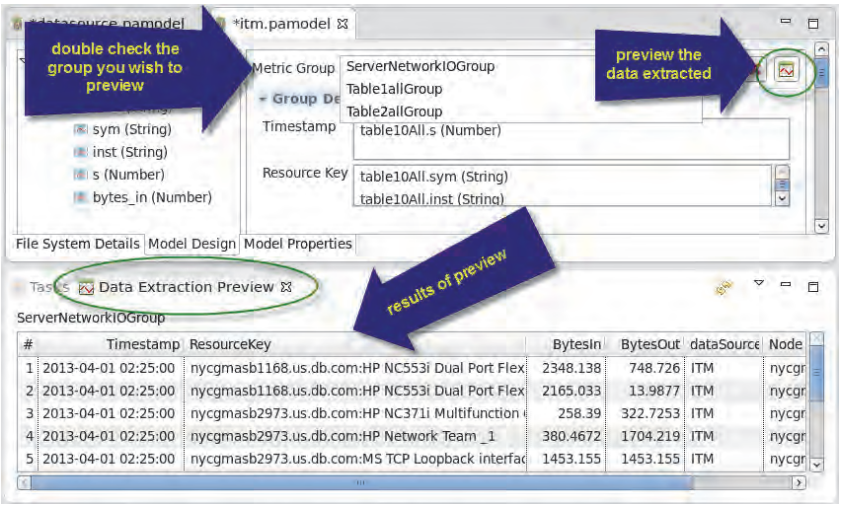
Adding attributes to metrics

Predictive Insights allows extra attributes to be added to the model to allow a KPI to have more context. However, these additional attributes must either be a constant value, or its value is included in the data source. These attributes appear both in the Predictive Insights GUI as well as a hash list in the alarm that is sent to OMNIbus.

IBM Training

Testing data sources

- After defining a model, you should always preview data
- Generates any errors that prevent previewing
  - Data type mismatches
  - Resource keys missing aligning attribute

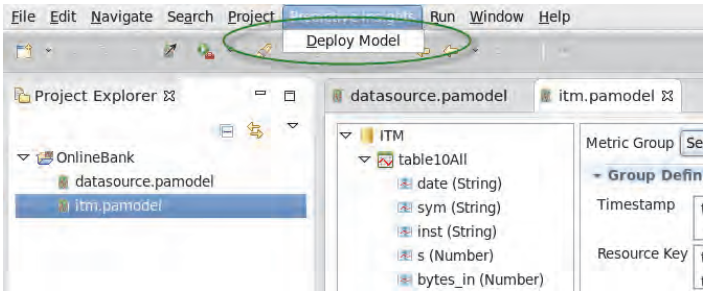


Testing data sources

When you modify your data model, you can always test the extraction process with the mediation tool. This preview of data generates errors if data type mismatches or resource keys are missing their associated attributes. After you are confident that the extraction process works, you can save the model and deploy the model into the server. It is always important to check the time stamp for correctness. There are a number of ways that a time stamp can be misinterpreted by the mediation client.

Deploying models

- Models that are generated in data mediation client must be deployed to server
- Deploying model saves it to Predictive Insights database
  - Need Predictive Insights database location and credentials to complete deployment
  - Need passwords for all database data sources
  - If multiple topics are installed, you are prompted to select one
- Can deploy a data source or entire project
- Check **Task** tab for any errors in deployment



Deploying models

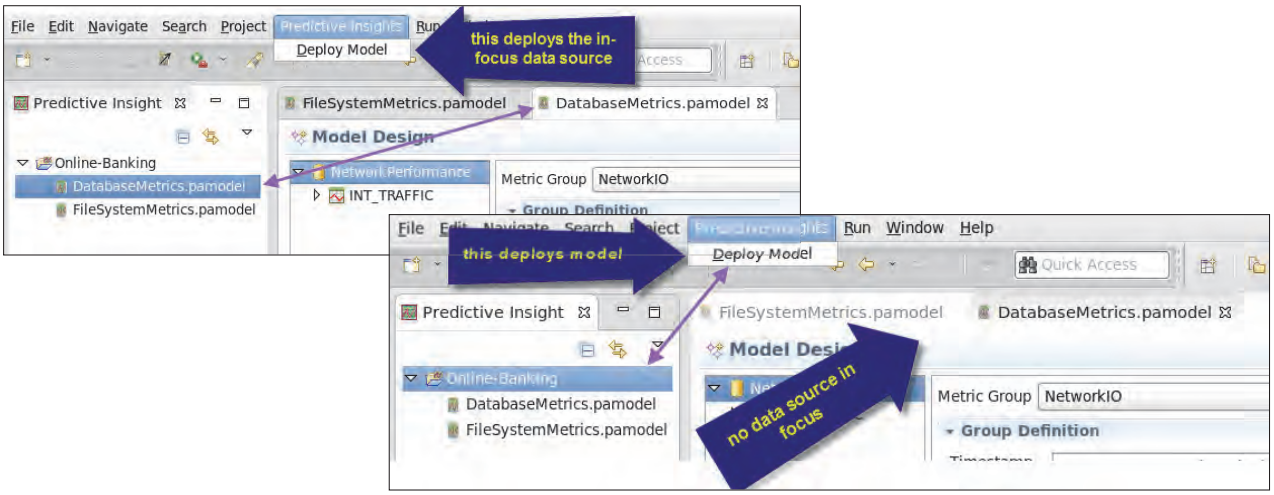
The output of the mediation tool is an XML file that must be deployed to the server to begin its analysis. The deploying process saves the XML file to a Predictive Insights database. The deployment process requests the location and credentials of the Predictive Insights database. If any of your data sources connect to a database, you are prompted for the user name and password for those databases also. If you have more than one topic that is deployed in your server, you are prompted to select which topic the data model is assigned to.

IBM Training

Deploying a data source versus deploying a model

Interface distinguishes between model and data source

Depends on what is selected



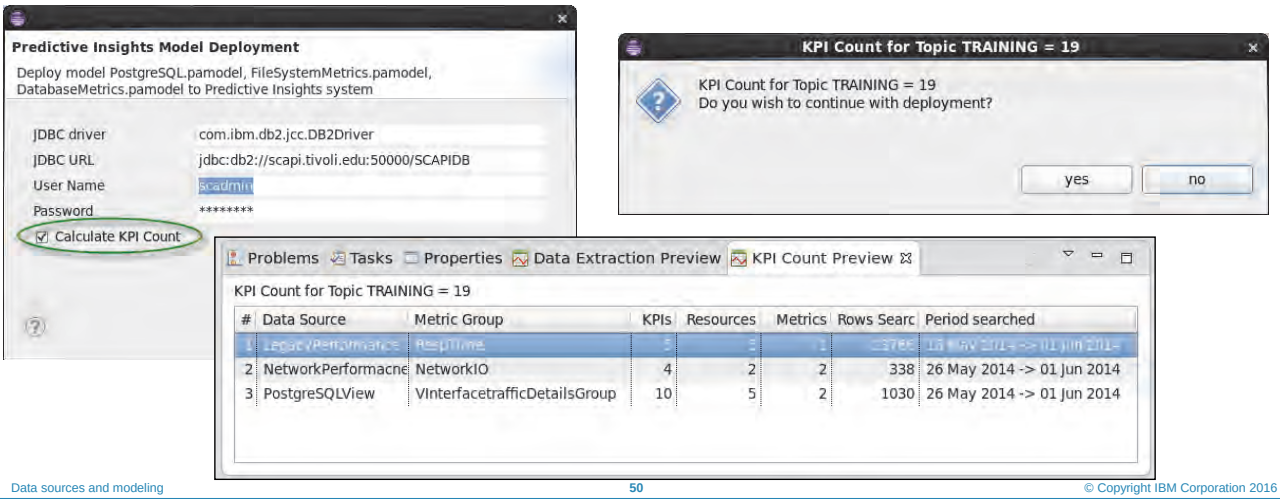
Deploying a data source versus deploying a model

The items that you select in the Project Explorer (the upper left window in the mediation tool) determine what is deployed. You can deploy a data source only, and not the entire project. This distinction is a subtle difference in the mediation tool GUI and can cause confusion when you believe that you deployed the entire model and not only a data source. Use the Project Explorer and select the item that you want to deploy. Right-click either the model or a data source to open a menu and deploy what you selected.

IBM Training

Total estimated KPIs

- Mediation tool can estimate the total number of KPIs
- Looks at days of data from each data source
  Calculates the number of metrics and resources from each source



Total estimated KPIs

When you deploy your model, you can calculate the number of KPIs that you selected. The total number of KPIs is based on how many metrics you have for each data source times the number of different resources you have in each data source. To determine the number of resources, the tool polls each data source for days of data and calculates the total number of unique resources it finds. Because resources are associated with each data source some but not all of the time, this calculation is an estimate. The details of each data source and their KPI count is displayed in the mediation client. This calculation is run only at deployment.

IBM Training

Updating a deployed model

- Updating the model requires the model be redeployed
  - Adding or removing data sources
  - Adding or removing KPIs
  - Changing resource names
  - Adding or removing attributes
- Redeployment to **existing** instance only allowed in either of these circumstances
  - Only changes to model are the addition or deletion (no modification) of metric groups
  - No data was loaded into the system
- Cleaning out the database is done with a cleanup command
- Adding metric groups require time to begin creating anomalies
  - No way to back-fill a metric group with historical data

Updating a deployed model

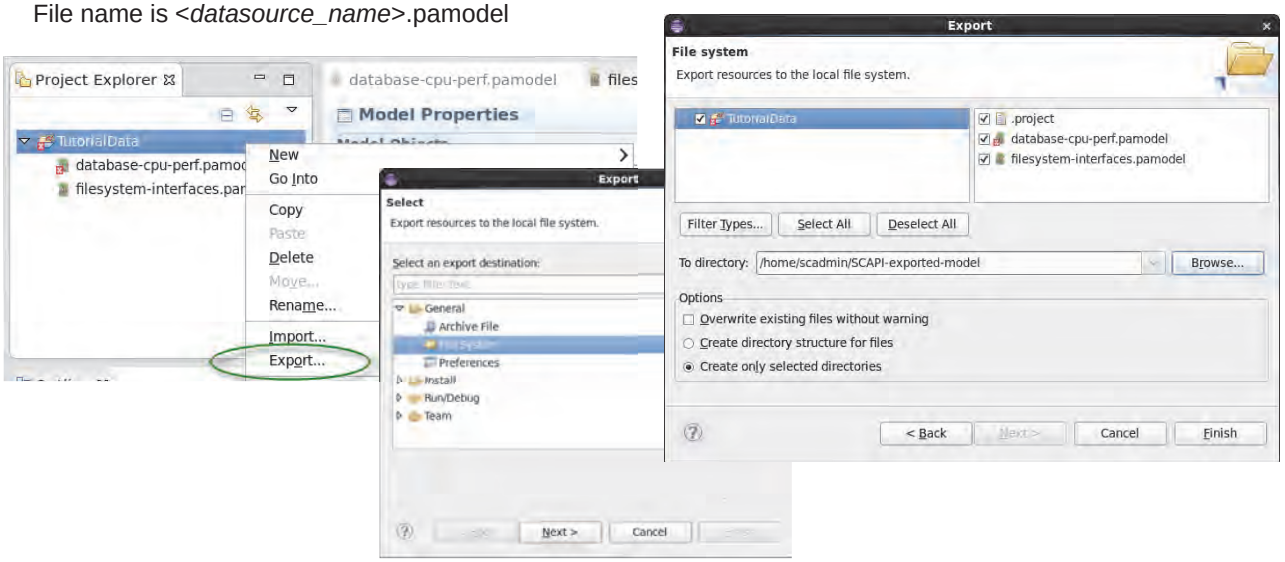
After you deploy a model and it is actively used, you might have to provide updates. Updates might include adding or removing data sources, KPIs, or changing the naming conventions inside the data model. Typically, the server does not allow updates, and you must delete the model, clean up the database, and restart the analysis. You can redeploy a model without these drastic steps only if you add or delete metric groups to the model or no data is loaded into the system. In all other instances, you must clean out the database with a command-line utility. Even if you add metric groups and you can redeploy, it takes 14 days for the new metrics to be trained and alarms to be generated. If you have historical data on all the metrics in the system, you might clean out the server with the command-line utility and retrain it with the historical data.

IBM Training

Exporting data models

Mediation tool creates an XML file for each data source

File name is <*datasource_name*>.pamodel



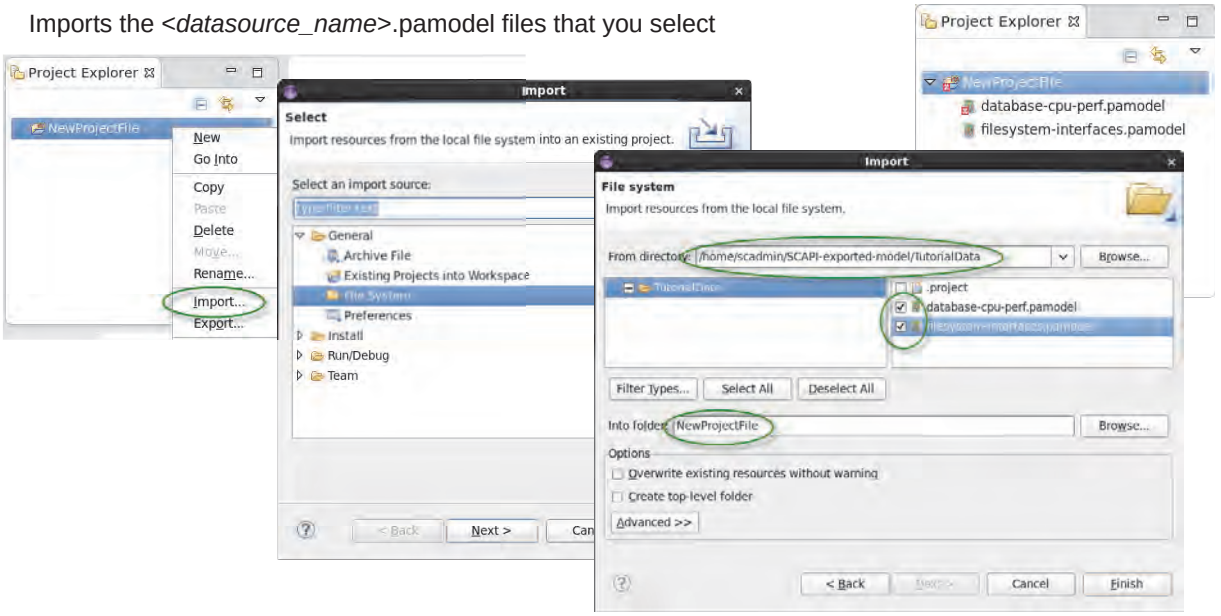
Exporting data models

After creating a model that meets your needs, you can export it from the system and reimport it elsewhere by using an XML file. To export a model in the Project Explorer, you right-click the project name. You are prompted for the types of data sources and the location where you want to export the data to. The export process then places the *.pamodel files in the selected directory.

IBM Training

Importing data models

Imports the <*datasource_name*>.pamodel files that you select



Importing data models

When you import a model into the mediation tool, you import the data sources and not a model. You create a project in the mediation tool, right-click the project, and select Import. You browse to the location of the *.pamodel files. Then, you are prompted to select which data sources you want to import.

IBM Training

Student exercises

Student exercises

In these exercises, you model a series of data sources, including a file system and databases. You use the mediation tool to rename objects, filter resources, and test extractions. After you define the model, you deploy into the analytics server.

IBM Training

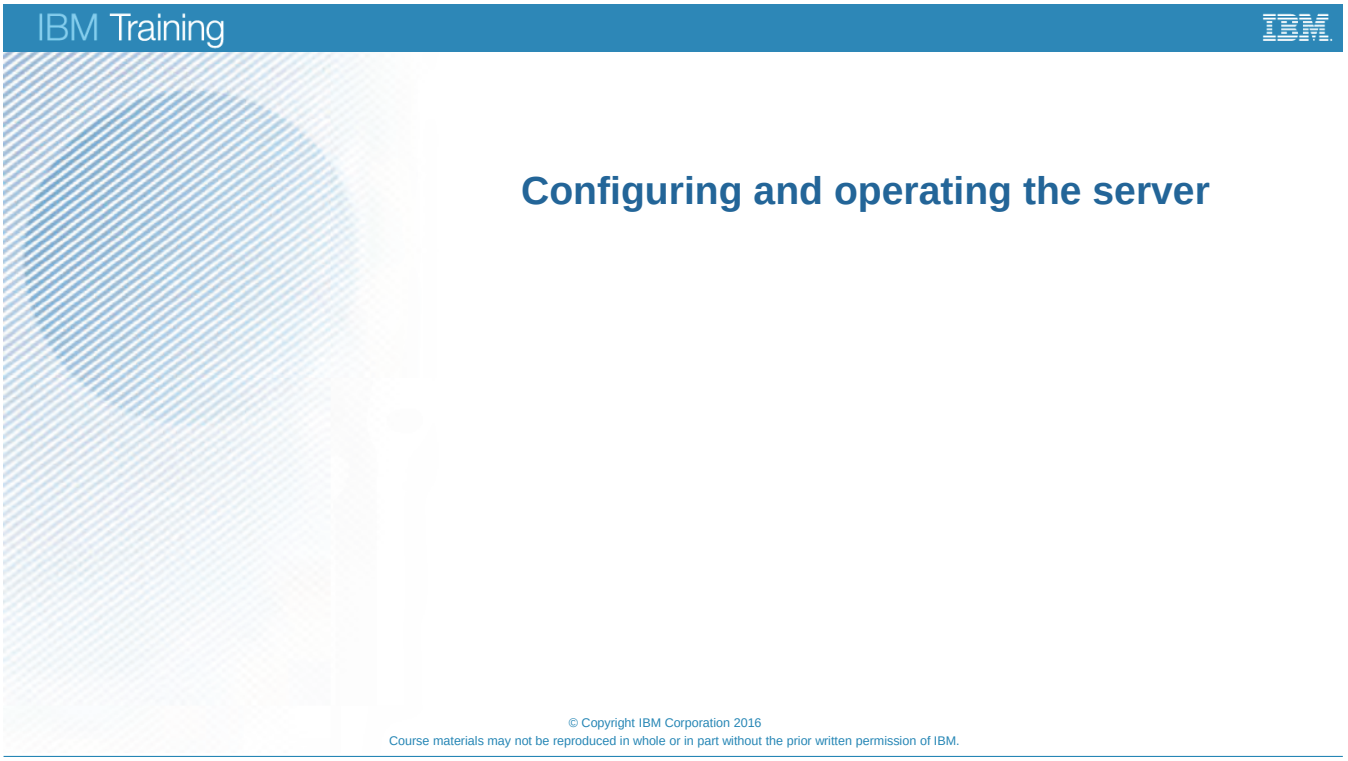
Unit summary

Having completed this unit, you should be able to complete the following tasks:

- Select useful key performance indicators
- Connect to external data sources
- Model data sources with the client mediation utility
- Deploy the data model

Unit summary

Unit 5 Configuring and operating the server



This unit describes how to configure the most important aspects of an analytics server. You learn how to start the server and analysis, and ensure that both are operating correctly. You also learn how to suppress alarms and work with the data that is generated when an alarm occurs.

IBM Training

Unit objectives

After completing this unit, you should be able to complete the following tasks:

- Configure the server for important properties
- Start and monitor the analysis of your data
- Working with attributes
- Enriching and filtering alarms
- Interact with alarms inside the GUI
- Describe the data that is associated with an alarm
- Search for various anomalies and metrics

Unit objectives

Lesson 1 Configuring server properties

In this lesson, you learn about the server configuration properties that are most important for operating the analytics server.

IBM Training

Configuration settings are by topic

- Topics are analyzed independently of each other
- Topics can be configured separately
  - **What is the aggregation interval?**
  - Should alarms be cleared when anomaly disappears?
- Need to know the topic names to configure settings

```
./admin.sh show_topics
```


Configuration settings are by topic

With topics, you segregate data and create separate analytics servers that analyze their data independently of each other. With this separation, you can also configure the servers separately. For each server, you can define aggregation intervals and how often the server should retrain itself. The key to configuring each of these topics separately is to know the topic name that you defined either at installation time or when you created it with the command-line utility.

IBM Training

What can be changed about the analytics server

flatLine.enabled: true
flatLine.maxTrainingWindowDays: 14
flatLine.retrainingIntervalMinutes: 1440
granger.enabled: true
granger.maxSeasonalTrainingWindowDays: 14
granger.maxTrainingWindowDays: 3
granger.retrainingIntervalMinutes: 1440
granger.threads: 4
variantInvariant.enabled: true
variantInvariant.maxTrainingWindowDays: 28
variantInvariant.retrainingIntervalMinutes: 1440
robustBounds.enabled: true
robustBounds.maxTrainingWindowDays: 28
robustBounds.retrainingIntervalMinutes: 1440
relatedEvents.enabled: true
relatedEvents.retrainingIntervalMinutes: 1440
nofm.error.count.min.size: 3
nofm.error.count.window.size: 6
system.aggregation.interval: 5
system.alarm.autoclear: false
system.alarm.history.retention.days: 180
system.alarm.session.keepalive.hours: 6
system.da.days.before.alarm.cleared: 7
system.da.missing.intervals.before.alarm: 3
system.instance.name: PI
system.max.metricgroups: 50
system.metric.retention.days: 15
system.omnibus.enabled: true
system.timeZone: Default
system.topic.allowNegativeNumbers: false
ui.granularity.default.min: 0
ui.summary.dlg.attr.names: Node

What can be changed about the analytics server

A number of items that can be configured about the Predictive Insights server. Most of them never need to be modified. On the slide, ones that you might consider changing are highlighted in bold text.

IBM Training

Determine how often to extract data

- Use system.aggregation.interval
- An important value that affects entire application
  - Aggregation interval is 5 - 15 minutes
  - You set to the least common multiple of collection times
  - Latency also affects what data is extracted from data source
- Default setting is 15 minutes
- Example

```
./admin.sh set_property –t=<TopicName> system.aggregation.interval 5
```


Determine how often to extract data

You must configure the server for how often to extract data. You typically decide this interval early in the implementation. The value should be 5 - 15 minutes to produce timely alarms. If you monitor data sources that have different collection intervals, you should set your aggregation interval to the least common multiple. For example, collection times of 2 minutes and 5-minutes result in a least common multiple of 10 minutes for the aggregation interval. If data arrives late to the data source, you can set a latency factor to ensure that the late data is collected and analyzed. Latency is a value that you set when you start the server.

IBM Training

Determine when to let the server clear alarms

- Use system.alarm.autoclear
- Predictive Insights creates alarms for OMNIbus
- By default, alarms are cleared when anomaly goes away
  - After cleared, you can search for anomaly from GUI
  - Historical data analysis often clears the alarms
- Setting to false requires alarms to be manually cleared
- Example

```
./admin.sh set_property -t=<TopicName> system.alarm.autoclear false
```


Determine when to let the server clear alarms

As anomalies are detected, they can eventually lead to an alarm sent to OMNIbus where they can be investigated by operators. However, the default behavior of Predictive Insights is to clear the alarm when the anomaly is not seen in the next aggregation interval. This step clears only the alarm from OMNIbus. The anomaly is retained in the database and can be searched for from the Predictive Insights GUI. If you are reviewing historical data, you should consider setting this property to false so that you can see all the alarms that the historical data would have produced.

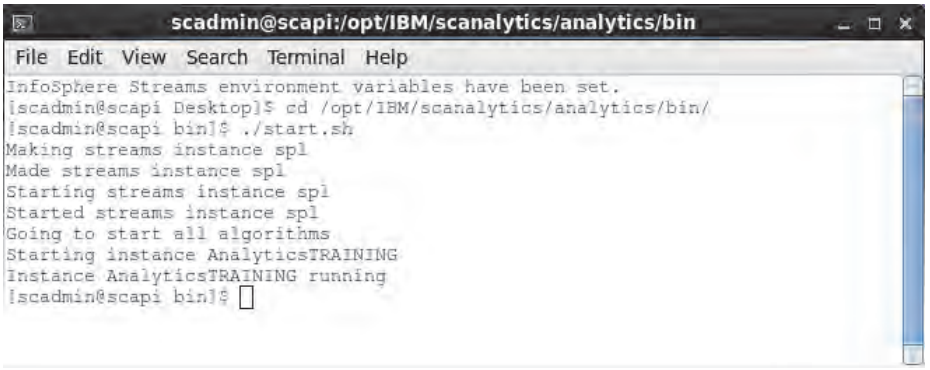
Lesson 2 Starting and monitoring analysis

In this lesson, you learn how to start the analytics server and start the analysis of your data sources. After you start the analysis of your data source, you learn how to confirm that the server is extracting data.

IBM Training

Starting the server

- Starting the server
  - Use **/opt/IBM/scanalytics/analytics/bin/start.sh**
  - Starts all topics on that physical server
- You must deploy a model to start the server



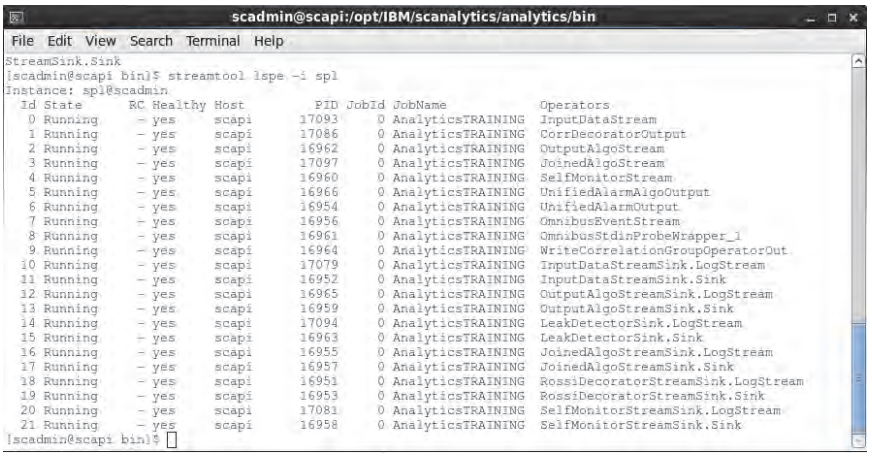
Starting the server

The start.sh command in the analytics bin directory starts the server and deploys all the instances. If you do not define a topic name, then all the topics that are associated to that server instance are started. If you attempt to start a server that does not have data models that are deployed to its topics, then you get a message and the server does not start. The starting of the server can take a few minutes as it starts all the algorithms that are needed for the server.

IBM Training

Validating that the server is running

- Confirming all processes started
  - streamtool lspe –i spl
  - Make sure all processes are running and healthy
- If some are not running, wait a few minutes and list them again
- If still not functioning, call Support



Validating that the server is running

The server initialization process starts many processes. A good practice is to ensure that all the processes are started and are healthy. An InfoSphere Streams utility displays all the processes and their current states. If some of the processes are not running, you should wait a few minutes and try the command again. If some processes are not started, or worse, marked as not healthy, call Support.

IBM Training

Starting the analysis

- After the server is started, you can extract the data
- Use the **run_extractor_instance** command
  - `/opt/IBM/scanalytics/analytics/bin/admin.sh run_extractor_instance -s <start-time> [-e <end-time>]`
  - Starts extraction on all topics
  - For other options, see the administration guide
- For historical-only data, define explicit <end-time>
  - Otherwise, system searches for data until current time
  - Loose alarms that were generated
- Timestamp format
  - yyyyMMdd-hhmm
  - Example: 10 April, 2016 = 20160410-1200

Starting the analysis

After the server is started and is healthy, you can start the analysis of your data. The run_extractor_instance command begins, and potentially stops the extraction of data and its analysis. The default command is EXTRACT to begin the extraction of your data source beginning at a specific time. If you do not declare a topic, extraction is started for all of the topics. If you are analyzing historical data, it is important to set an end time for the analysis. If an end time is not specified, the server continues to extract data from the data source until the current server time is reached. If the historical data is old enough, the server deletes anomalies that are older than 90 days. Setting the end time prevents this problem.

Data is late reaching the data source

- Data can be slow reaching the data source
  - Late data is dropped if not retrieved by extraction call
- Set latency to slowest data source update
- run_extractor_instance –s <start-time> -l <latency>
  - Latency defined in minutes
- Example
  - Current time: 12:00 pm
  - Latency: 15 minutes
  - Aggregation interval: 5 minutes
  - Extract data between 11:40 am and 11:45 am
- Latency delays alarm generation

Data is late reaching the data source

Data can be slow in reaching the data source. If there are delays, they must be accounted for in the calls that the extractor makes. If adjustments are not made, then when PI makes its call to the data source and asks for data between time stamps that has not arrived, that data is never recovered. When implementing the solution, cataloging any delays in the data sources is important. With this list of delays, you can set the latency of the Predictive Insights server to the slowest source. The latency is set when you start the server and can be set for each individual topic. For example, if the current time is 12 noon and the aggregation interval is 5 minutes, by default the extractor selects data between the current time and 5 minutes into the past. However, if a data source has a 15-minute delay in getting its data, you can set the latency to that amount. This step changes the call to select data between 11:40 AM and 11:45 AM. It important to understand that by setting latency, you are looking that many minutes into the past and not at current data. This difference means that anomalous conditions that are currently happening have alarms that are delayed by the latency time.

Confirming whether analysis is running

- System extracts data from data sources based on aggregation interval
  - Historical data retrieved quickly
- Review alarms sent to OMNIbus
- Log of data sources that are read by extractor
  - `$TASP_HOME/log/<TopicName>/Analytics<TopicName>_log_SelfMonitorOperator.log`
- Any data extracted from data sources is copied to file system
  - `$TASP_HOME/var/spool/topics/<TopicName>/extracted`
  - File name <datasource>_<metricgroup>__<start-time>__<end-time>.csv

Confirming whether analysis is running

After starting the analyzer, it is helpful to confirm that its extraction process is working without any errors. In normal operations, it extracts data at every aggregation interval. If you are loading historical data (time stamps older than the current system time), the data is retrieved quickly. To check to see whether data is being retrieved from the data source, tail the log file **Analytics<*TopicName*>_log_SelfMonitorOperator.log**. Note the path requires the topic name to be included. Tailing this file notes the calls that are made to each data source and how much data was extracted from each. If there are problems connecting to the data source, it is included in this log. Another method for checking Predictive Insights extractions is to look at the directory extracted. Any data that is extracted from the data sources is placed in this directory as a CSV file. This data is not only useful to check Predictive Insights progress but also the data is used to reload the server in case you need to retrain it. Note the format of the file name as Predictive Insights breaks up the data by metric group and time stamps.

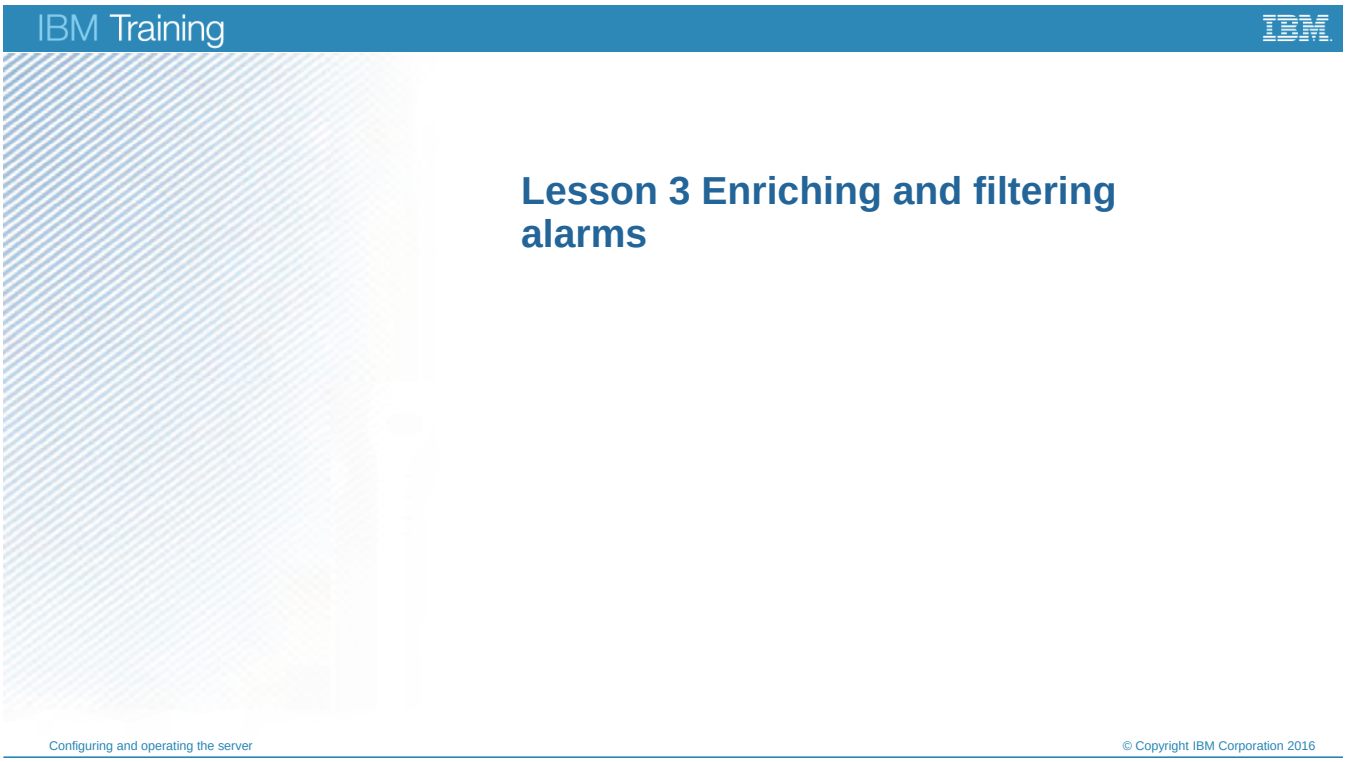
Flat file extraction process

- As files are read, they are moved from defined directory
  - <defined-directory>/good, if file is successfully read
  - <defined-directory>/bad, if there is a problem reading file
- These subdirectories must be cleaned using manual housekeeping

Flat file extraction process

If you have a data source that uses flat files, the files are moved to subdirectories in the file location named good and bad. If the flat file is successfully read, it is moved to the good directory. If there was a problem reading the file, it is moved to the bad directory. If a file is moved to the bad directory, any data in that file was not loaded into the analytics server. Eventually, these two subdirectories need to be cleaned out by using a manual housekeeping function.

Lesson 3 Enriching and filtering alarms



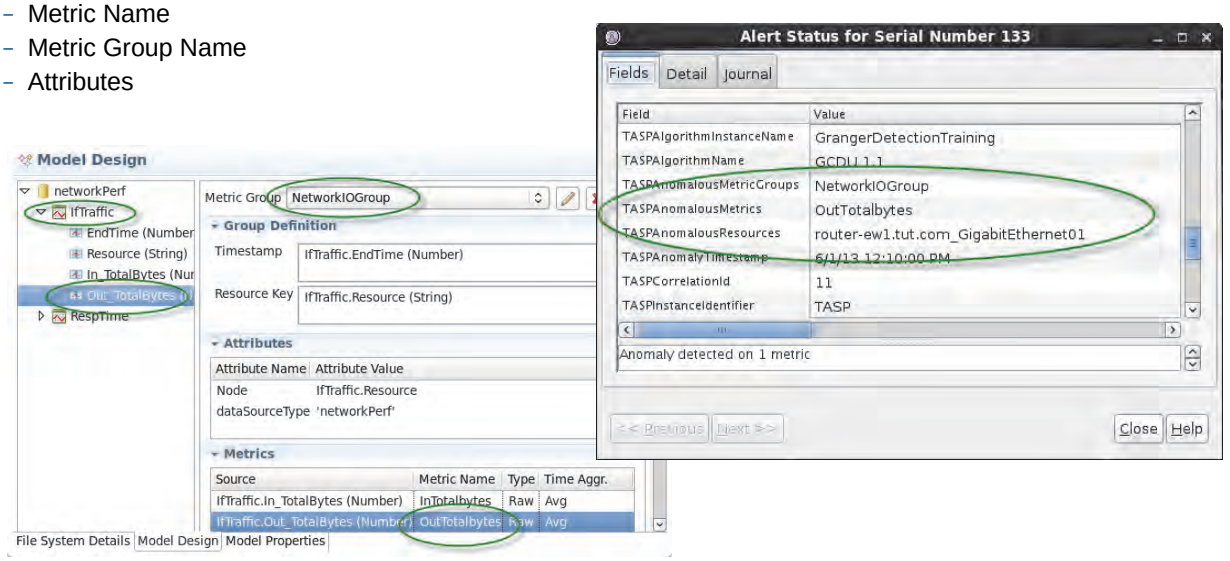
In this lesson, you learn how to suppress alarms and understand the fields that are available for various filtering operations by using OMNIbus rules files.

IBM Training

How a data model affects alarm fields

Naming conventions can be implemented in model

- Metric Name
- Metric Group Name
- Attributes

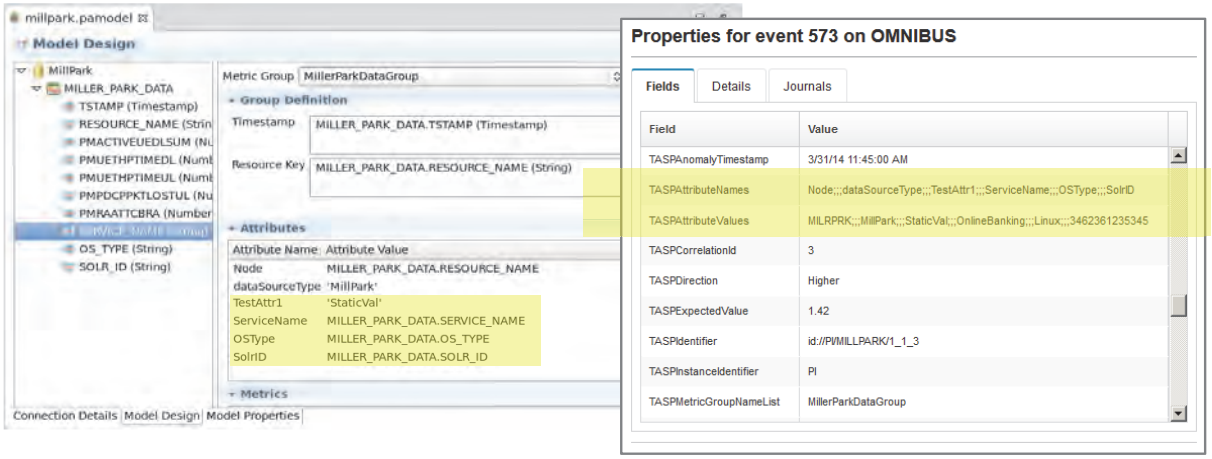


How a data model affects alarm fields

A useful feature of the system is the naming conventions that can be implemented in the data model and then show in the alarms. Both the metric name and the metric group name are included in the alarm that is sent to OMNIbus. You can configure both of these attributes when you create the model. You can use both of these values either suppress the creation of the alarm or the manipulation of the alarm by the probe rules file.

Attributes are incorporated into alarm

- Fields: TASPAttributeNames and TASPAttributeValues
- Delimiter: ;;;



Attributes are incorporated into alarm

When you define attributes in your model, you are able to see those values in the alarms that are sent to OMNIbus. You can use this list of attributes and their values with OMNIbus rules file to enrich the alarm as necessary. The TASPAttributeNames and TASPAttributeValues have a list of either attributes or values. It uses four semicolons (**;;;**) as a delimiter. You must break this list apart in the rules file or other postprocessing tasks.

IBM Training

Filtering alarms

- Predictive Insights filter file can suppress alarms before they leave server
  - $TASP_HOME/spl/instances/Analytics<TopicName>/config/filtered_alarms.txt
- Allows for a list of filters to be defined
  - Either forward or discard alarm
  - Can optionally escalate or de-escalate the alarm when forwarding
- Alarm filters are tested against each alarm
  - Filters are tested from top to bottom of list
  - First match is used to either forward or discard alarm
  - No match, alarm is forwarded
- Any changes to file are immediately applied
- Anomalies always recorded in database even when suppressed

Filtering alarms

Predictive Insights is supplied with a filtering mechanism that suppresses an alarm so that it is not released from the server. This methodology is controlled through a text file that is like an access control list used in networking. The file **filtered_alarms.txt** is associated to each topic in the noted directory. This file has an ordered list of rules that either forwards or discards the alarm to the OMNIbus probe. When the Predictive Insights alarm is generated, it is sent to this file. Each filter is applied to the alarm, in order, from the top of the list to its bottom. The first rule that matches the fields in the alarm is used to process it and either forwards it or discards it. If no rule matches the alarm, it is automatically forwarded to the probe. This file is reread every time that an alarm is created. Any changes that are made to the file are used on the next alarm, and nothing needs to be restarted.

IBM Training

Predictive Insights simple filter format

regex|wild,<*resource name*>,<*metric group name*>,<*metric name*>, forward|discard|critical|major|minor|warning

- regex|wild
  - regex allows regular expressions: [Dd]isk_space_.*
  - wild allows the basic wildcards of * and ?: *_CPU*
- Can filter by the resource name, metric group name, and/or metric name
- Examples
  - wild,*,*,*,forward
    Forwards all alarms, no subsequent filter is used
  - wild,*,IOStats,*,discard
    Discards all alarms from metrics in the IOStats metric group
  - regex,*,IOStats,%.*,critical
    Forwards all alarms with metric names starting with “%” and are in IOStats metric group as critical events
  - regex,.*[Tt][Ee][Ss][Tt].*,*,*,warning
    Forwards all alarms from resources with **test** in their name as warnings

Predictive Insights simple filter format

To use this filter file, you must understand the format of the forward and discard statements that it uses. You can filter on any combination of the resource name, the metric name, or the metric group that the metric name belongs to. The format uses a matching method flag of either **regex** or **wild**.

This flag is followed by three fields that can be configured with either an asterisk or a matching expression for the resource name, metric group name, and metric name. The fifth field uses a flag that notes whether to forward or discard the alarm. Declaring the regex match method flag, you can use a full regular expression (Java Regex) to create a matching criteria. Otherwise, the wild flag allows the simple use of an asterisk or question mark. Regardless whether an alarm is discarded, the anomaly that generated the alarm is still captured in the database and can be reviewed later.

In the first example, the wild flag is declared followed by three asterisks. Because all three fields can match anything, all alarms match this statement. If you place this statement at the top of the file, then all other statements are ignored and the alarm is forwarded. The second filter discards any alarms that are generated by any metrics that are part of the metric group called IOStats. The third statement forwards all alarms with metric names starting with a percent sign and is in the IOStats group. It also sets their severity to critical. The final statement uses a regular expression to forward any alarms from resources that have any variation of TEST in their name and sets their severity to warning.

IBM Training

Adding rules to filter

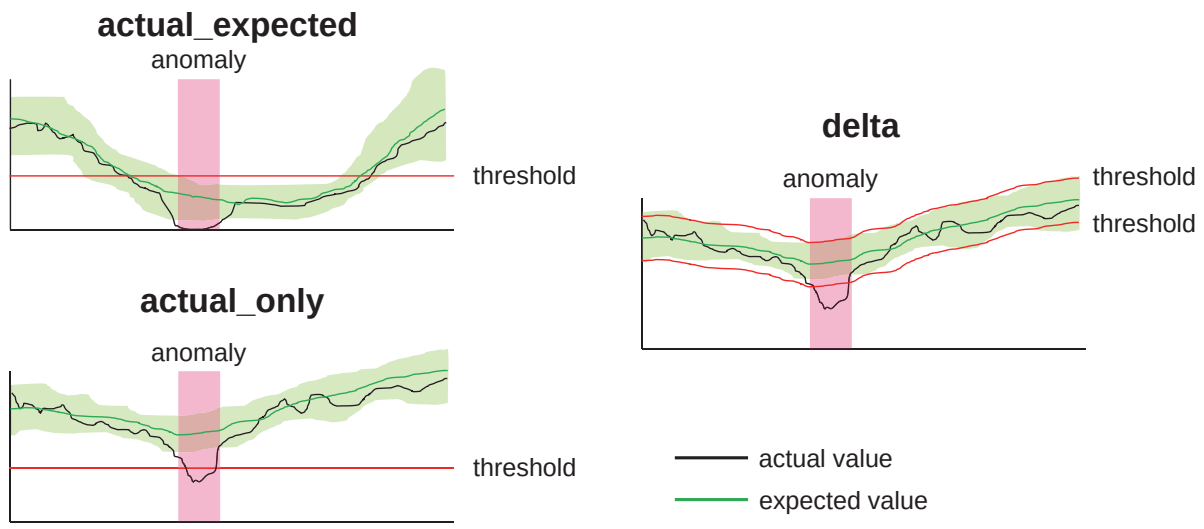
regex|wild,<resource name>,<metric group name>,<metric name>, <rule type>, <threshold>,forward|discard|major|minor|warning|critical

- At the time of the alarm, the **actual value** and **expected value** of metric is noted
- Rule types that use thresholds
  - **actual_expected:** checks if both the actual and expected values are ***less than or equal to*** the threshold
  - **actual_only:** checks only if the actual value is ***less than or equal to*** the threshold
  - **delta:** checks if the difference between the actual and expected value is ***less than or equal to*** threshold
- Rule types that ignore thresholds
  - **higher**: checks if the actual value is ***higher*** than the expected value
  - **lower:** checks if the actual value is ***lower*** than the expected value
  - Remember, the actual value will already be significantly higher or lower than the expected value

Adding rules to filter

An extra level of filtering available from the **filtered_alarms.txt** file is the use of rules to check the various metric values. When the alarm is created, it has an actual and expected value associated with it. You can use these values to compare against a threshold that is above and beyond the thresholds used by the robust bounds. You can also use rules that check whether the anomaly is above or below the expected value.

Visualizing rule types that use thresholds



Visualizing rule types that use thresholds

Here are how these rules would look when you apply them to alarms. As you can see, the threshold value can either be a static value or it can be a boundary that surrounds the expected value much like a normal threshold.

IBM Training

Best practices for filter rules

- Put some thought into writing filters that use thresholds
  - Uses ***less than or equal to*** logic
  - When using rules to control alarm severity, put lowest alarms first.

```
wild,*,ResptimeGroup,Maxresponsetime,delta,1000,discard
wild,*,ResptimeGroup,Maxresponsetime,delta,2000,minor
wild,*,ResptimeGroup,Maxresponsetime,delta,3000,major
wild,*,ResptimeGroup,Maxresponsetime,critical
```

- Use **higher** and **lower** to suppress alarms

Best practices for filter rules

Here are some thoughts on using rules. The **filtered_alarms.txt** file uses a less than or equal to logic. If you want to control the severity of an alarm, then you want to define the lowest severity first and work up. You can use the higher and lower rules to suppress alarms. For example, if interface is experience lower than expected traffic, it might no be useful to generate an alarm.

Enriching alarms with OMNIbus rules files

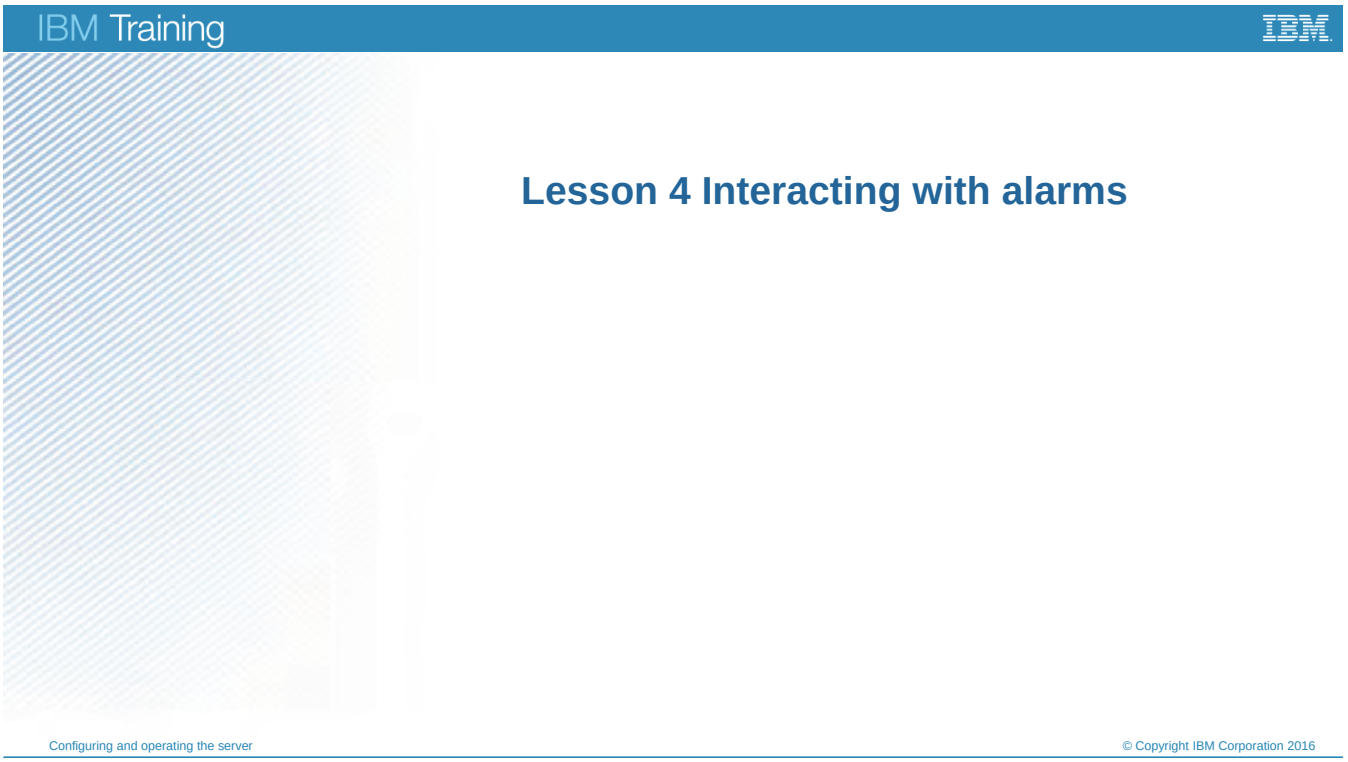
- Typical alarm fields used for enrichment
  - TASPAnomalousMetricGroups
  - TASPAnomalousMetric
  - TASPAnomalousResources
  - TASPAttributeNames and TASPAttributeValues
  - TASPActualValue and TASPExpectedValue

- Location of rules file
  - $TASP_HOME/probe/omnibus/probes/linux2x86
  - stdin-tasp.rules

Enriching alarms with OMNIbus rules files

The second place where alarms can be discarded or enriched is in the rules files that are used by the OMNIbus probe that is installed with Predictive Insights. The rules file can use the additional fields that are provided by Predictive Insights, which include TASPAnomalousMetricGroups, TASPAnomalousMetric, TASPAnomalousResources, and TASPTopic. Remember that TASP is the name of this software before Predictive Insights. The rules file installed with Predictive Insights is in the probe's subdirectory and is named **stdin-tasp.rules**.

Lesson 4 Interacting with alarms

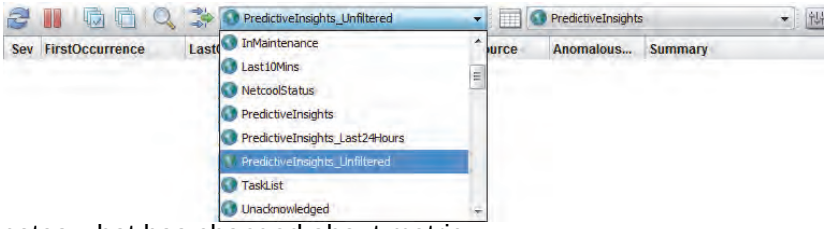


In this lesson, you learn how to operate the user interface when alarms from Predictive Insights are generated. You also learn how to access the historical anomalies that are kept in the database.

IBM Training

What an operator sees

- Use predefined filters and view to help isolate Predictive Insights alarms



- Summary notes what has changed about metric

Sev	Summary	Node	AnomalousResource	AnomalousMetric	Direction
	Incident on 5 metrics across 3 nodes	PIINCIDENT30	alm_w91700NTNtPro...	Processortime;;;Usertime	
	Node boc_w91701NTNtProcessorGroup has 2 simultaneous alarms	boc_w91701NTNtProcessorGroup	boc_w91701NTNtPro...	Processortime;;;Usertime	Lower
	Correlated metric Processortime has 3 correlated alarms	PICORRELATIONS1000000142	alm_w91700NTNtPro...	Processortime	Lower
	Related Events consolidation on 2 metrics	PIRELATEDEVENT0	abcresource;;;xyzreso...	Cpubusy;;;Filecontrolbyte...	Higher
	Usertime is Lower than expected. Actual: 0.556 Expected: 7.426	boc_w91701NTNtProcessorGroup	boc_w91701NTNtPro...	Usertime	Lower
	Packetsreceived is now a flat line where before it was varying.	Sapa:ARG9:SD_gon.com	Sapa:ARG9:SD_gon.com	Packetsreceived	
	Intotalbytes is Higher than expected. Actual: 5.135e7 Expected: 1.851e7	GigabitFlow-c0372	GigabitFlow-c0372	Intotalbytes	Higher
	Filecontrolbytessec64 is Higher than expected. Actual: 311008 Expected: ...	xyzresource	xyzresource	Filecontrolbytessec64	Higher
	Cpubusy is Higher than expected. Actual: 70 Expected: 12.98	abcresource	abcresource	Cpubusy	Higher

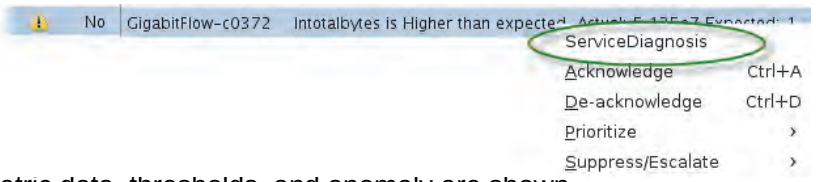
What an operator sees

In most cases, users do not go directly to the Predictive Insights GUI but starts with the Active Event List and possibly sees an alarm that was forwarded from the analytics server. A set of predefined filters were installed when it configured WebGUI/Tivoli Integrated Portal. Using these filters display just the alarms generated by Predictive Insights. Note that the alarm descriptions include information on the actual and expected values for the anomalous metric.

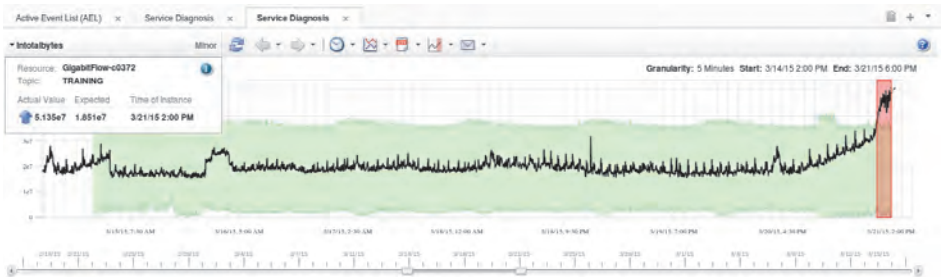
IBM Training

ServiceDiagnosis allows detail viewing

- Right-click alarm and select **ServiceDiagnosis**



- Details of the metric data, thresholds, and anomaly are shown

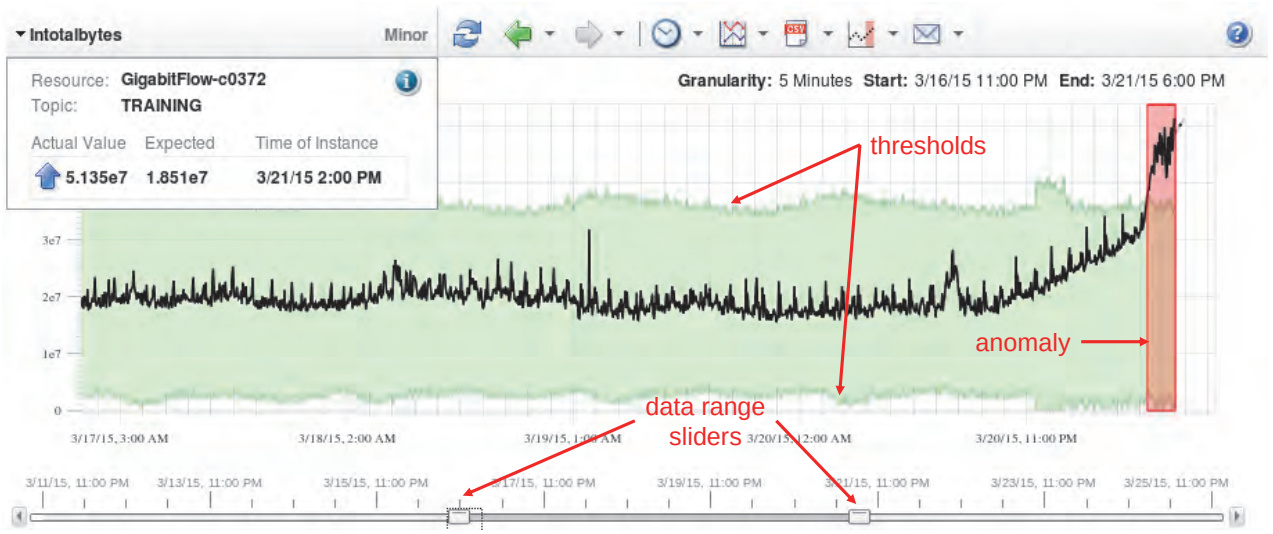


ServiceDiagnosis allows detailed viewing

After recognizing the alarm, the operator can examine its details by right-clicking and selecting **ServiceDiagnosis**. Selecting this option opens a tab and displays the most recent anomaly with the calculated thresholds. Along the bottom is the name of the metric that is displayed.

IBM Training

What is displayed

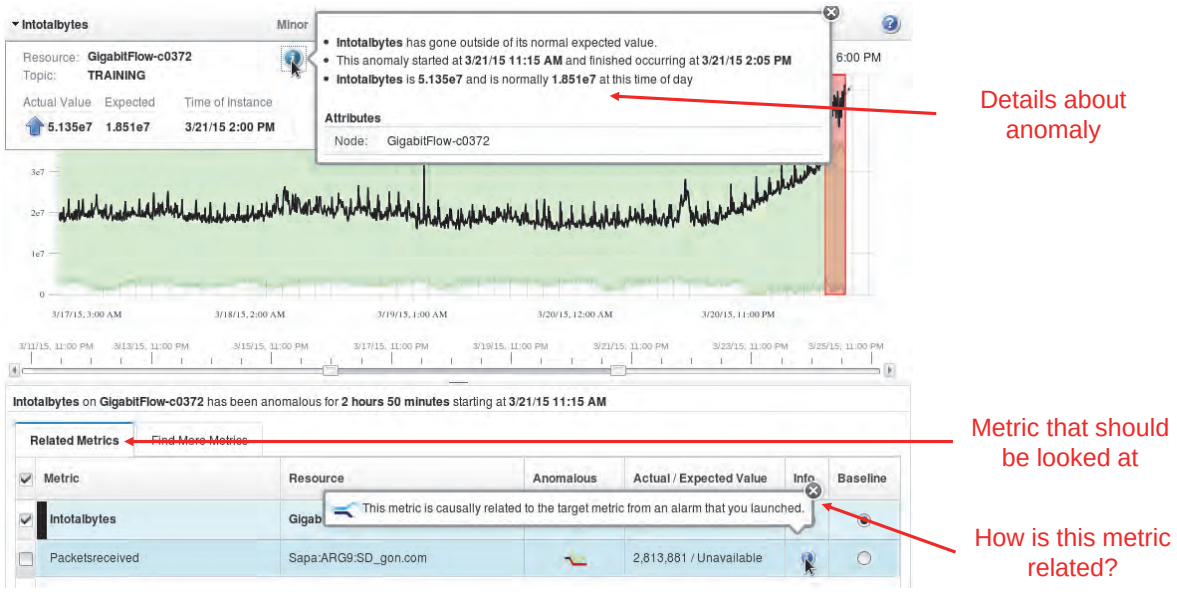


What is displayed

The Predictive Insights user interface shows the details of the alarm. If you are looking at a single metric, the green fairway that makes up the thresholds is displayed. You see a shaded red area that shows the anomaly that is associated to the alarm. You can expand the range of the data by using the data range sliders. Note the actual and expected values in the upper left corner of the screen. By default, the screen shows a lighter shade of red for anomalies that were suppressed because they did not reach the N of M rule and become an alarm.

IBM Training

Items of interest



Items of interest

By pointing to the information icon at the top of the screen, you can get more details about the anomaly and why it alarmed. On the lower portion of the screen, you can review the metrics that are related to the one that alarmed. These metrics have a relationship with the alarm being displayed. Sometimes that relationship comes from alarm consolidation that occurs automatically. Other times, there can be a deeper relationship with the alarming metric. These deeper relationships could be associated with the Granger analysis, or they are related metrics or correlated metrics. Pointing to the information icon in the lower screen describes what the relationship is.

IBM Training

Anomaly icons displayed in user interface

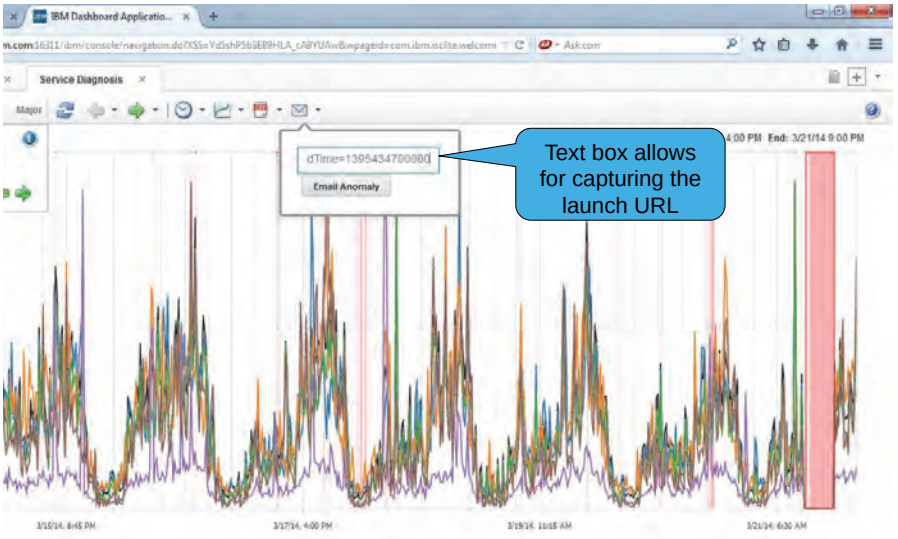
Alarm Type	Description
Robust Bounds	Metric x is higher/lower than expected.
Flat Line	Metric x is now a flat line where before it was varying.
Variant/Invariant	Metric x is now steady where before it was varying.

Anomaly icons displayed in user interface

To help in alarm identification, KPIs that become anomalous have icons to display the type of anomaly they are experiencing.

IBM Training

Email the anomaly



Email the anomaly

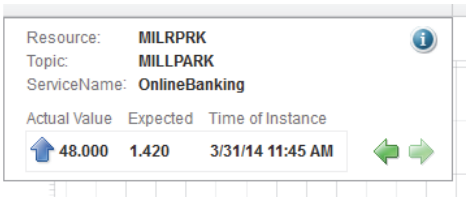
If needed, a URL to the anomaly can be captured and sent to interested parties.

IBM Training

Adding attributes to user interface

- Custom attributes can be added to interface to help build context
  `./admin.sh set -t=<topic>` **`ui.summary.dlg.attr.names`** `Location, Contact`

- A favorite attribute can be placed on the highly visible summary
  `./admin.sh set -t=<topic>` **`ui.summary.dlg.favorite.attr.name`** `ServiceName`



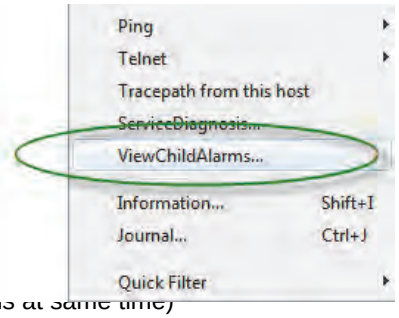
Adding attributes to user interface

To help with context, the attributes you defined in the data model can be included in the user interface. You can add as many attributes to the information page associated with the anomaly. By default, the node name is all that is displayed. Using the administration tool, you can change or add more attributes. It is important that the names are spelled correctly and have the correct capitalization. In addition, you can have a single favorite attribute that is displayed on the highly visible placard that is shown when the anomaly is brought up in the interface.

IBM Training

Alarm consolidation

- The goal is two fold:
  - Provide additional context and understanding
  - Reduce the number of anomalies that need to be investigated
- Node consolidation
  - A node can have one or more resources associated to it
    Example: node=hostname and resource=hostname:interface
  - Consolidates all the anomalous metrics happening on the node
- Related event consolidation
  - Consolidates anomalous metrics that are related to each other (anomalous at same time)
- Correlation consolidation
  - Consolidates anomalous metrics that are correlated to each other (data that is tightly related)



Alarm consolidation

Alarm consolidation is a feature to reduce the amount of clutter that is displayed in OMNIbus and a means to provide context when a series of anomalies are occurring simultaneously. The alarms can be consolidated in one of three ways. If multiple metrics are anomalous at the same time on a single node, they are rolled into a single consolidated alarm.

Attention: Note the difference between a node and a resource. A node is always a single host name. However, because of the methods that you can use to model data, a host name can have one or more resources associated to it. A resource can be just the host name but can also be an interface or some other subcomponent associated to the node.

Another method of consolidation is to group all the alarms that are related to each other. This consolidation means the alarms are anomalous at the same time. A third consolidation method is to group anomalous alarms that are highly correlated to each other. These metrics track each other closely. You can break a consolidated alarm down into its individual alarms by right-clicking the alarm and selecting ViewChildAlarms.

IBM Training

Viewing consolidations in Active Event List

Summary	Node	AnomalousResource	AnomalousMetric
Node boc_w91701NTNtProcessorGroup has 2 simultaneous alarms	boc_w91701NTNtProcessorGroup	boc_w91701NTNtProcessorGroup	Processortime;;;Usertime
Correlated metric Processortime has 3 correlated alarms	PICORRELATIONS1000000142	alm_w91700NTNtProcessorGroup;;;boc_w91701NTNtProcessor...	Processortime
Related Events consolidation on 2 metrics	PIRELATEDEVENT0	abcresource;;;xyzresource	Cpubusy;;;Filecontrolbyte...

- Node consolidation
  Displays node name, list of anomalous resources and list of anomalous metrics
- Correlated event consolidation
  Displays metric name and list of anomalous resources
- Related event consolidation
  Displays list of anomalous resources and list of anomalous metric

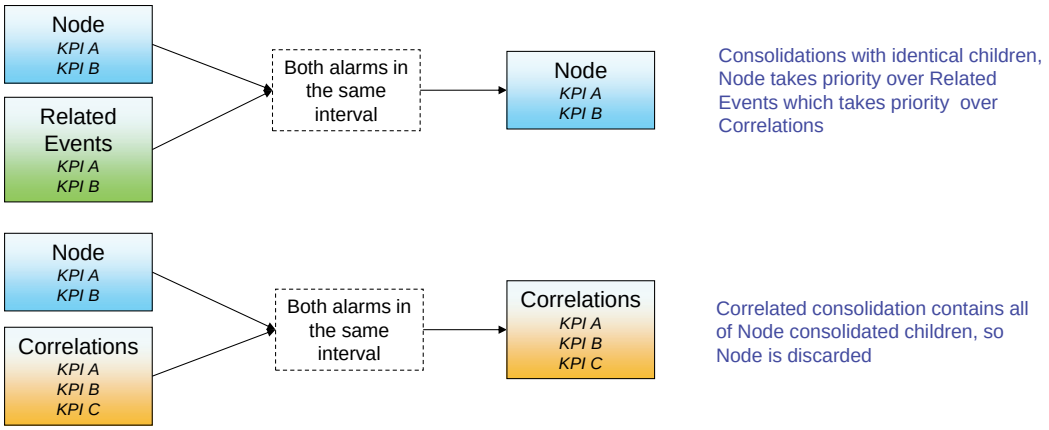
- Delimiter is ;;;

Viewing consolidations in Active Event List

Note the differences in the descriptions for each of these consolidated events. If multiple nodes are involved, they are delimited with three semicolons.

Consolidation collisions

- Multiple consolidations can occur in an interval that may share child metrics
- To reduce the number of alarms, PI attempts to only show one consolidation
  Note that metrics of one consolidation are, at least, a subset of the other



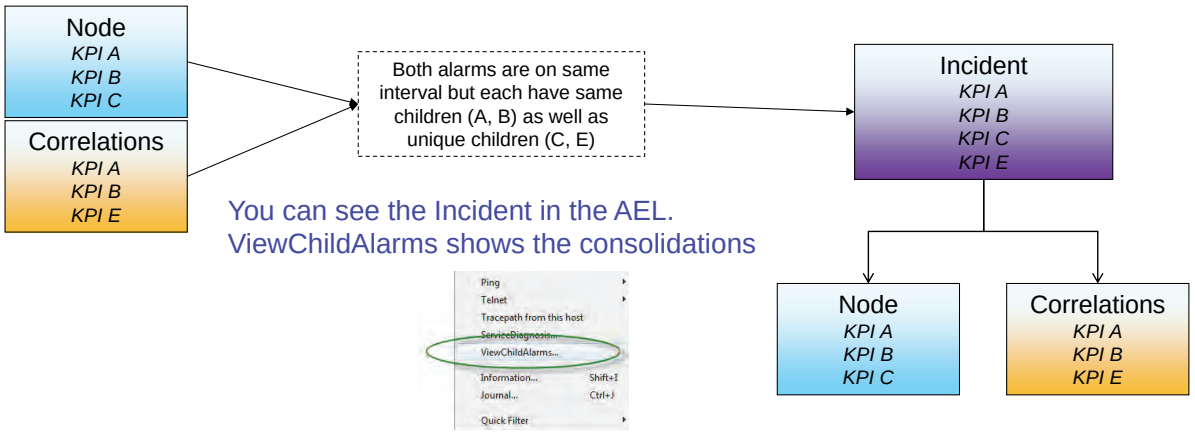
Consolidation collisions

On occasion, multiple consolidations are displayed at the same time. If those consolidation share metrics so that one consolidation can be shown that includes all of the metrics, then Predictive Insights attempts to reduce clutter more by showing only one consolidation.

IBM Training

Overlapping consolidations

- Referred to as Incidents (consolidation of consolidations)
- Incidents are created when only some of the children are shared between consolidations

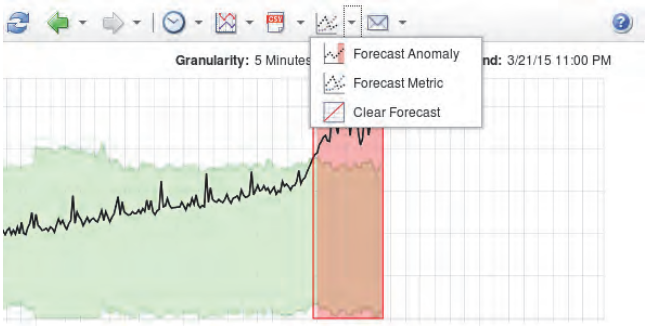


Overlapping consolidations

However, not all consolidations have metrics where one consolidation has all the metrics of the other. In this case, an ***incident*** is created. An incident is a consolidation of consolidations. Using the same method to break down a consolidation into its constituent alarms, the ViewChildAlarms menu pick breaks down the incident into its constituent consolidations.

Forecasting metrics

- Predictive Insights attempts to predict what your target KPI will do
- Predicted values obtained on-the-fly
  Many algorithms available, each suited to a different kind of metric pattern
- Best model selected automatically to generate predicted values
- Forecasts are based on either the anomaly or all available data



Forecasting metrics

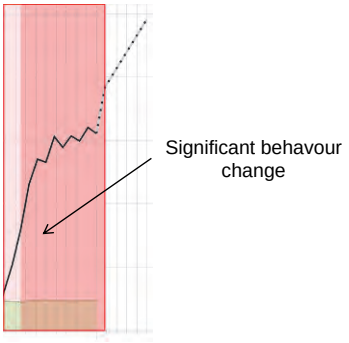
Predictive Insights has forecasting capabilities built into its math model. Using many different algorithms that are suited to certain data patterns, it attempts to predict what the metric might do if it is left unattended. These predictions are based on just the anomalous data or the larger historical trend.

IBM Training

Forecasting behavior

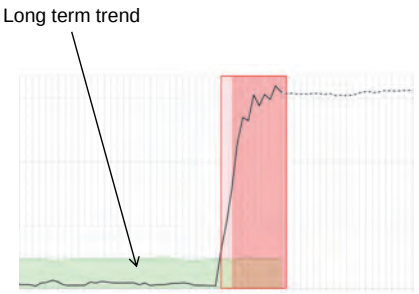
Forecast anomaly

- Uses only data inside anomalous window (up to 24 intervals)
- Only considers recent significant behavior changes



Forecast metric

- Builds a forecast model based on all available data (up to 48 intervals)
- Takes into account overall, long term trend & seasonality



Forecasting behavior

When using the forecasting methods, you can either forecast the anomaly or forecast the metric. The difference is the amount of data that is included in the analysis and any recent changes in behavior. The slide details the difference between the two analyses.

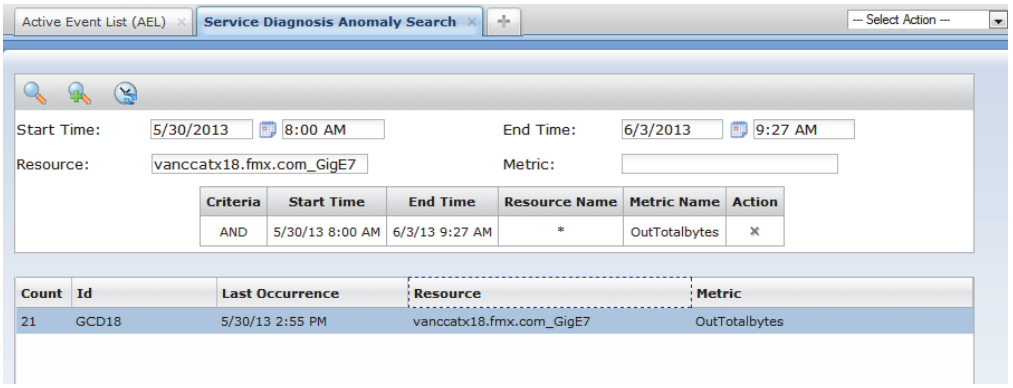
When forecasting the anomaly, it uses only the data that is captured in the anomaly itself. And even then, it is looking for major changes in behavior and including that in its predictions. As you can see, the prediction carries on with the major change in behavior.

The forecast that is associated with the metric uses a larger amount of data and considers the long-term trend of the data. On the slide, you can see that the prediction carries on the long-term trend of the data but only at the higher value.

IBM Training

Searching for anomalies

- Search the database for previous anomalies
- Search by dates, resource name, or metrics
  Can have multiple *and* search criteria
- Double-click anomaly to see details

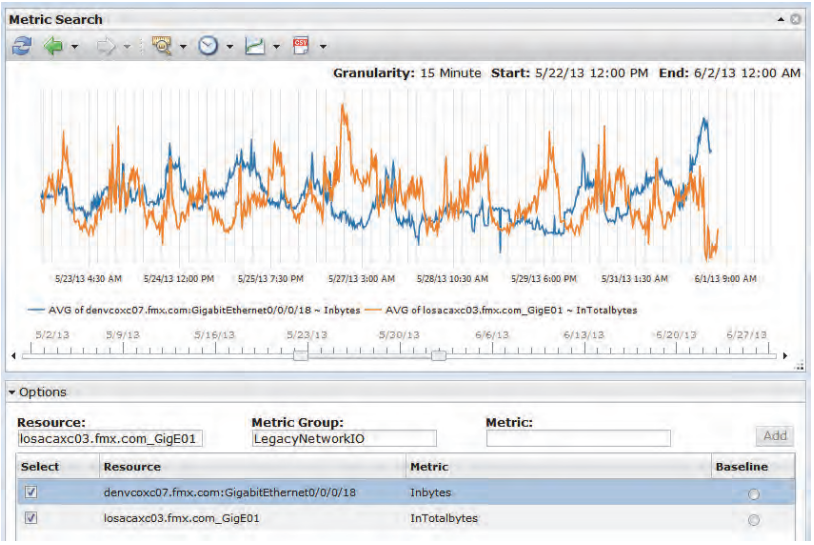


Searching for anomalies

In the Predictive Insights GUI, you can search the database for previous anomalies. You can search by date, resource name, or by metric. You can add extra criteria through AND statements. After you find the anomalies, you can double-click them to see the detailed chart of the data.

Searching for metric data

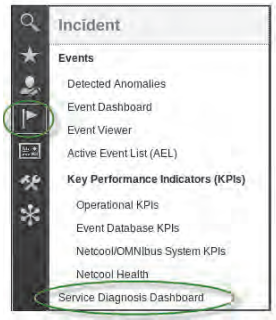
Displays one or more metrics
- Data is normalized by default

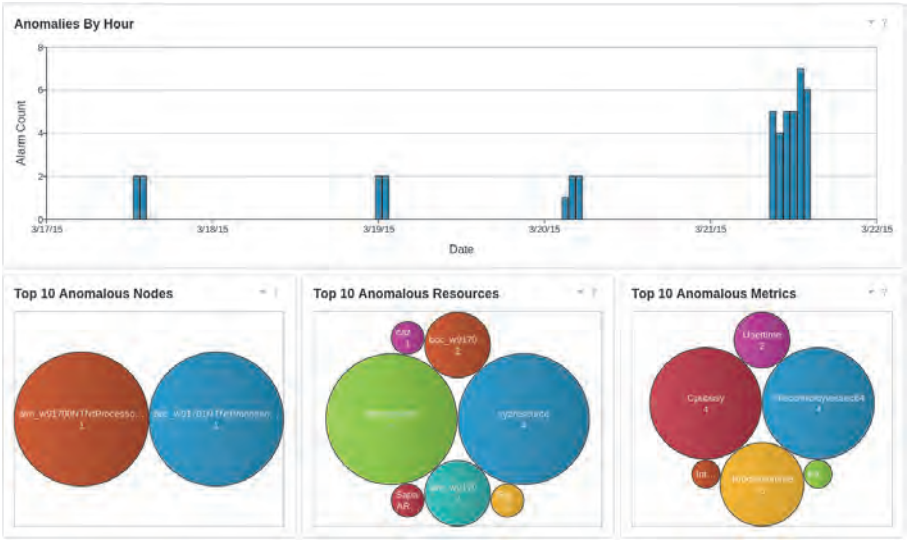


Searching for metric data

You can also search for metrics and display their data for a particular date range. You can add multiple metrics to the screen simultaneously to look for trends. When multiple metrics are displayed simultaneously, the screen normalizes the data to remove any skew that you see between data with significantly different units. This normalization can be removed, but the two data streams might be hard to compare.

Dashboards





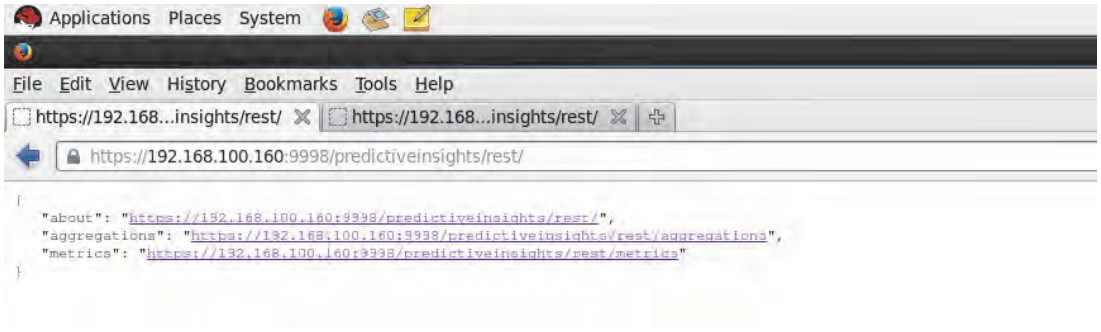
- Drill down to view details is enabled

Dashboards

Predictive Insights has dashboards where you can view the recent anomalies that have occurred on the system. It breaks the anomalies down into the top anomalous nodes, resources, and metrics. Remember that a node can have one or more resources associated with it. On these dashboards, you can drill down and view the anomaly details.

REST API

- Exposes anomaly counts and metric baselines through REST for use in other tooling
- Simple interface returning either HTML or JSON
- HTML interface shows URIs and data format
- Top level REST URI: https://<machine>:9998/predictiveinsights/rest



REST API

A REST (Representational State Transfer) API is now included with the server. With this API, you can extract anomaly counts and metrics baselines to be included in other applications. It currently a simple interface that either returns HTML or JSON data to the requesting application. You can access this interface by using a browser and pointing it to the top-level URI at this location:

https://<*Predict User Interface*>:9998:/predictiveinsights/rest.

IBM Training

Student exercises

Student exercises

These exercises include starting your server and analyzing the data sources that you defined earlier. You also look into the anomalies that the system creates and use the Predictive Insights GUI to investigate their detail as well as search and display other metrics.

IBM Training

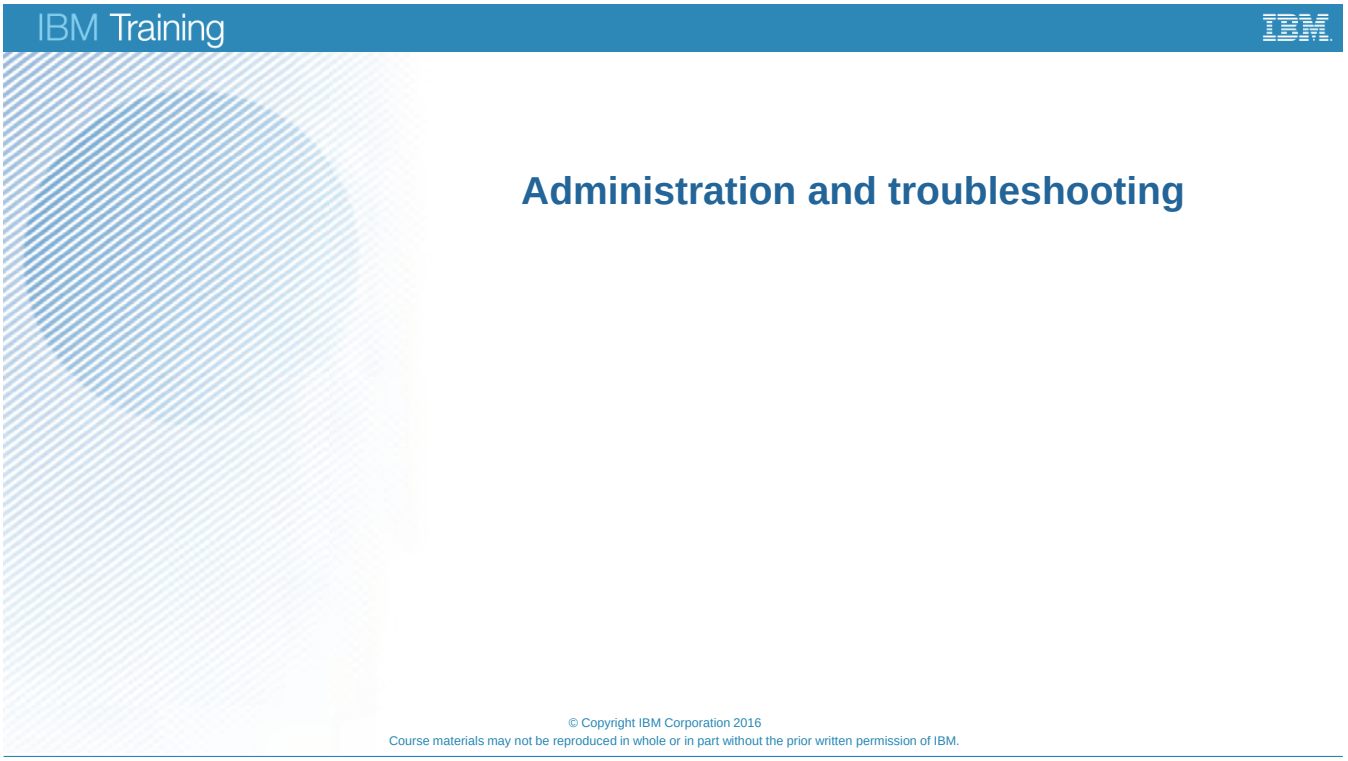
Unit summary

Having completed this unit, you should be able to complete the following tasks:

- Configure the server for important properties
- Start and monitor the analysis of your data
- Working with attributes
- Enriching and filtering alarms
- Interact with alarms inside the GUI
- Describe the data that is associated with an alarm
- Search for various anomalies and metrics

Unit summary

Unit 6 Administration and troubleshooting



This unit provides information about common administrative tasks and troubleshooting techniques.

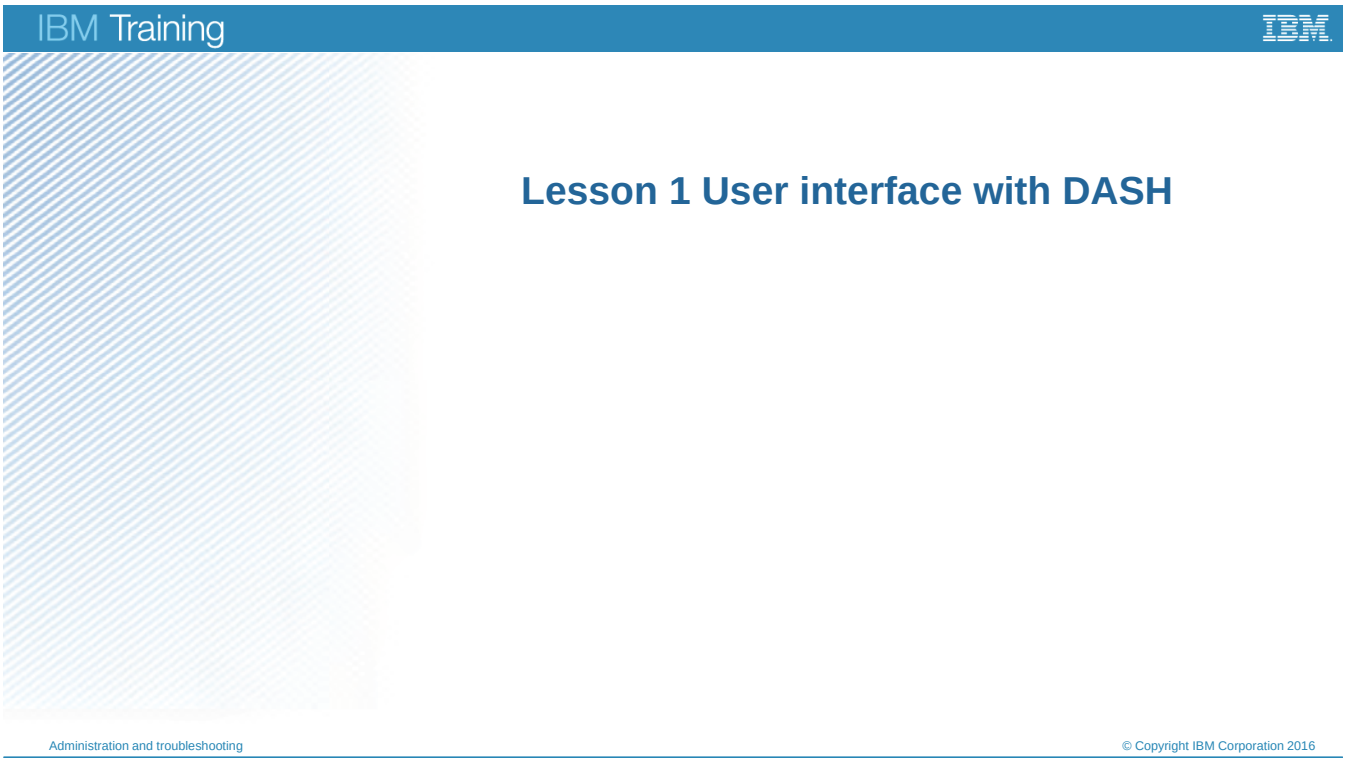
IBM Training

Unit objectives

- Start and update security settings with user interface
- Describe extraction options and processes
- Troubleshoot issues with mediation client and extraction
- Debug problems with alarms

Unit objectives

Lesson 1 User interface with DASH

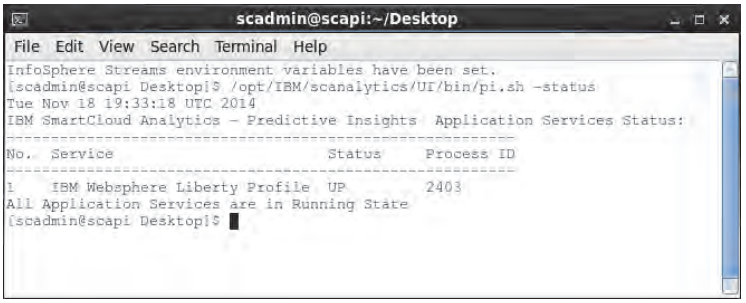


This unit covers the process of data extraction and the various methods and messages that are generated when you work with the applications data mediation processes.

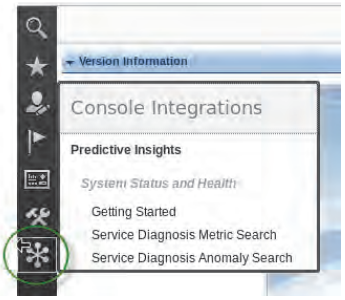
IBM Training

Starting and stopping Predictive Insights UI

```
/opt/IBM/scanalytics/UI/bin/pi.sh [ -start | -stop | -restart | -status]
```



If you are missing snowflake icon, PI is most likely not started



Starting and stopping Predictive Insights UI

The Predictive Insights user interface is a separate web server that is federated into the DASH interface. This web server must be started and stopped like the other parts of the application. If you log in to the DASH interface with a user that should have access to Predictive Insights and you do not see the snowflake icon, you know that the server is either not running or is having problems. You can also use the status command to determine the health of the server. If the server is running but you still cannot see the snowflake icon, then restart the server.

IBM Training

DASH support: Authentication

- DASH federation requires single sign-on between DASH container and Liberty container
- SSO requires shared authorization
  - OMNIbus (default)
  - LDAP
- Authentication configuration is included with product
  - WebGUI configures OMNIbus authentication in the DASH container
  - WebGUI creates default users and groups in OMNIbus store
    - **Netcool_OMNIbus_User** and **Netcool_OMNIbus_Admin** groups
      Users in these groups automatically have access to PI user interface
    - **ncouser** and **ncoadmin** users
  - Predictive Insights installs and configures Liberty based on WebGUI authentication

DASH support: Authentication

Because the Predictive Insights user interface is hosted in WebSphere Liberty and is federated into DASH, there must be an authentication mechanism between the two. You authenticate through single sign-on using either OMNIbus or LDAP. OMNIbus authentication is configured by default when you install Predictive Insights. Predictive authentication uses the WebGUI authentication mechanisms that were installed when WebGUI was installed. WebGUI creates default users and groups. The groups Netcool_OMNIbus_User and Netcool_OMNIbus_Admin are configured to automatically have access to Predictive Insights. WebGUI also installs the **ncouser** and **ncoadmin** users. When Predictive Insights installs, it bases its configuration on this WebGUI authentication mechanism.

IBM Training

DASH support: Authorization scenarios

- Introduction of a new group
  - Create group in DASH repository and assign users to group
  - Use the role mapping task in DASH to assign required WebGUI roles
  - Use **addAccess.sh** command to give group access on Predictive Insights UI server
- Introduction of a new user
  - Role mapping can be done for users, but not recommended
  - Best practice is to map user to groups
    For example: Netcool_OMNIbus_User or Netcool_OMNIbus_Admin

DASH support: Authorization scenarios

If you need to extend beyond the initial Netcool_OMNIbus_User or Netcool_OMNIbus_Admin groups, you can create new groups from the DASH repository. You access the WebSphere administrative console to create new groups. After the group is created, you must add the required WebGUI roles to gain access to the Active Event List. You then must let Predictive Insights know that this new group should be given access to its user interface. This authorization is provided by using a command line tool that is included with the Predictive Insights user interface.

While it is possible to give an individual user access to the Predictive Insights user interface, it is not recommended. The best practice it to create or use an existing group that has access to the Predictive Insights user interface.

IBM Training

DASH support: Authorization commands

- New groups must be given access to Predictive Insights GUI
- Use commands in $TASP_HOME/UI/bin
  - addAccess.sh – add role access for a user or a group

```
Usage:  ./addAccess.sh group|user <user or group name> [admin]

    <user or group name>
        the name of a group or user to give Predictive Insights role access

    admin
        optional flag to grant administrative privilege
```

  - removeAccess.sh – remove role access for a user or a group

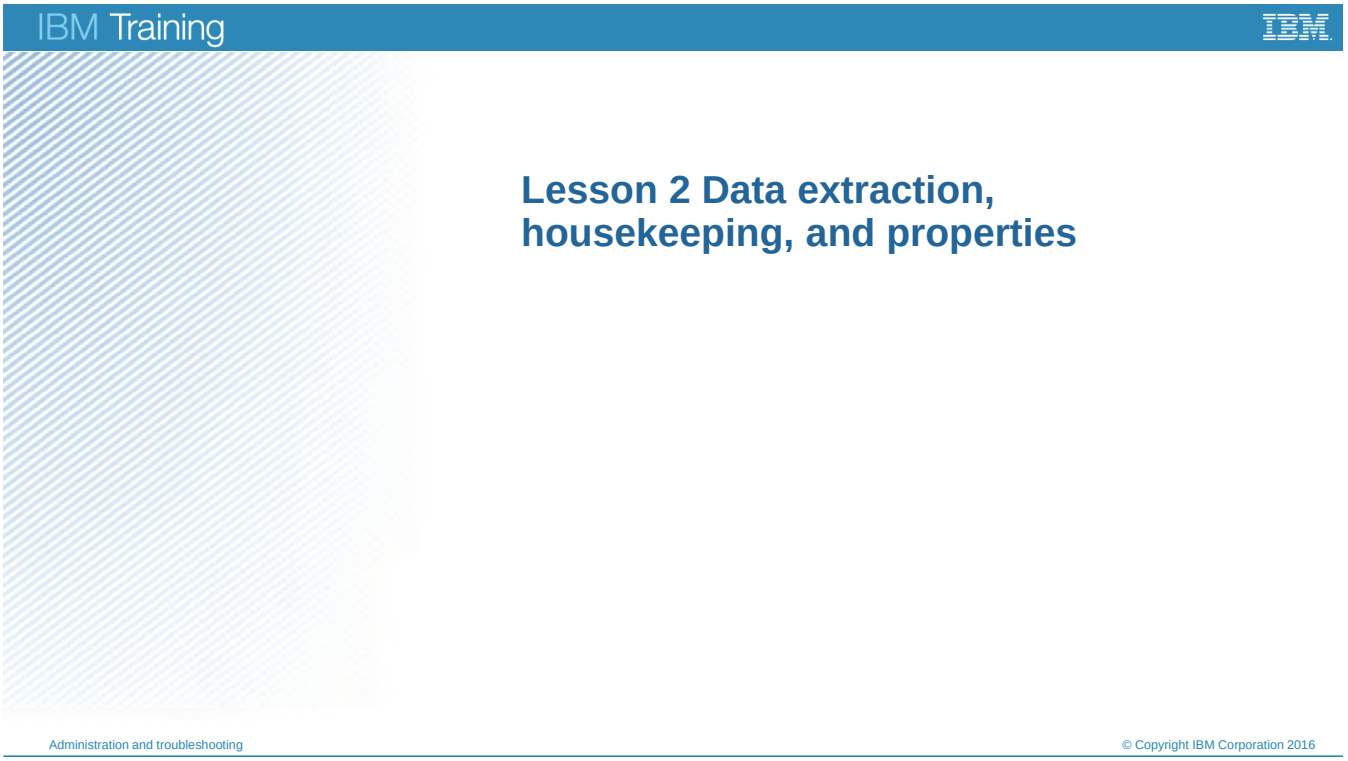
```
Usage:  ./removeAccess.sh group|user <user or group name>

    <user or group name>
        the name of a group or user to revoke Predictive Insights role access
```


DASH support: Authorization commands

After you have created a new group via the WebSphere administrative console and want to use Predictive Insights, that group must be added to the Predictive Insights authorization database. Adding this group to Predictive Insights must be done with a command line tool that is included in the user interface installation. You use the addAccess.sh and removeAccess.sh tools to add and remove these DASH groups.

Lesson 2 Data extraction, housekeeping, and properties



This lesson covers the process of data extraction and the various methods and messages that are generated when you work with the data mediation processes. It also describes the housekeeping methods that are used on an ongoing basis as well as completely erasing the server data. You are presented the properties that can be modified on a per topic basis, as well.

IBM Training

Negative values for metrics

- By default, negative numbers treated as missing data
  Many systems still use negative numbers to indicate missing data or invalid data
- Negative numbers can processed like any other number
  For example: temperature and sound
- Must be enabled before data is extracted
  `./admin.sh set -t=<topic> system.topic.allowNegativeNumbers true`

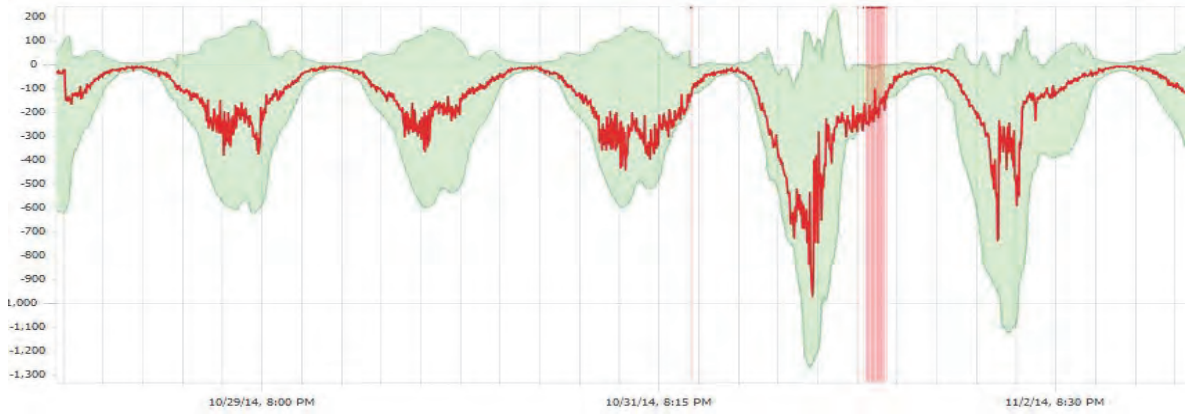
Negative values for metrics

It is not often that IT infrastructures have monitoring data that uses negative numbers. If it does, negative numbers can mean invalid data. However, Predictive Insights can monitor other types of time-series data, including temperatures and sound levels. These metrics can be negative. By default, the system treats negative numbers as missing data. If negative numbers have real value in your solution, you must enable the system to consider it as a true value. You enable it with the administration tool and turn it on for each individual topic.

Displaying negative numbers

Charts display negative values only when necessary

Baselines will stop at zero unless the metric has negative values



Displaying negative numbers

With negative numbers enabled, charts display negative values but only when necessary. Thresholds stop at zero unless the metric has negative values.

Normal extraction methods

- Uses EXTRACT mode by default with run_extractor_instance
- Steady state mode
  - Server is reading current data from data sources
  - run_extractor_instance has no end date
- Backlog mode
  - Server is reading historical data from data sources
  - run_extractor_instance has both start and end dates in past
- Switch mode
  - Start in backlog and becomes steady state
  - run_extractor_instance has both start date in past and no end date
- Performance issues when in backlog mode and analyzing
  - Want server in steady state before reaching trainingWindowWeeks

Normal extraction methods

When working with data extraction, you typically use the EXTRACT mode of the **run_extractor_instance** command that is part of the admin.sh utility, for example:

```
$TASP_HOME/bin/admin.sh run_extractor_instance –mode EXTRACT –starttime
20140623-0000
```

When you use the EXTRACT command, the data extraction happens in one of three ways, depending on the start and end times. In steady state, the system selects data from the data sources, based on the system clock of the server. It selects that data, based on the aggregation interval. For example, it selects data from the data sources every 5 minutes. Typically, in steady state, no end time is specified and the server continues to gather data until explicitly stopped by the administrator or by a reboot. The server might also be running in backlog where it collects historical data from the data sources as fast as possible. This behavior occurs when the start time is set to a time in the past. To stay in backlog, the end time for the analysis must also be in the past. There is also a switch mode where the server goes from backlog into steady state based on the server clock.

When the extractor reaches the current time for the system clock, it switches from the quick extraction process of the backlog mode to the more measured and regular process of steady state. When working with large numbers of KPIs, it is important that the server does not try to extract data in backlog while trying to analyze. Both these processes are processor-intensive and conflict with each other. If possible, set the start date for something less than the trainingWindowWeeks. For example, start time is 27 days in the past)

IBM Training

Special extraction methods

- EXTRACT_ONLY mode
  - Extracts only data from data sources, and does not analyze any of it
  - Places data in **$TASP_HOME/var/spool/topics/<topic>/extracted**
- REPLAY_RAW mode
  - Reads data in **$TASP_HOME/var/spool/topics/<topic>/extracted**
  - Processes only the data files
- Use case
  - A backlog data source is slow
    - Server falls behind, and RAM is filled waiting for data
  - Solution:
    - Use EXTRACT_ONLY to get backlog data
    - Use REPLAY_RAW to consume backlog
    - Use EXTRACT to move to steady state

Special extraction methods

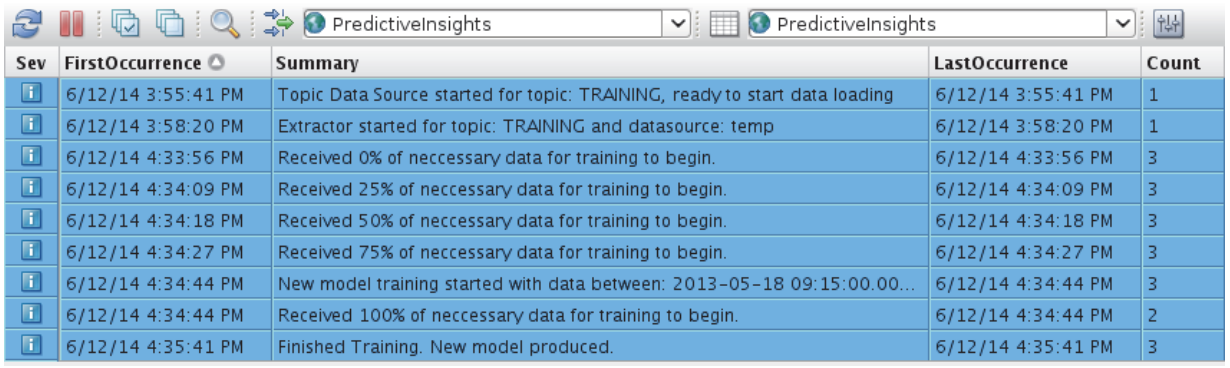
Two other special modes that you can use with the **run_extractor_instance** command are EXTRACT _ONLY and REPLAY_RAW. They are rarely used but can be helpful in certain situations. The EXTRACT _ONLY process extracts only the data from data sources; it does not attempt to analyze it. It extracts the data from the data sources and places it in the extracted directory. The REPLAY_RAW process then takes the extracted data in this directory and analyzes it. The REPLAY_RAW process uses all the data in the extracted directory and then stops. It does not move onto a steady state mode. You would need to execute the run_extractor_instance command one more time so it could enter the EXTRACT mode and begin extracting live data from the data sources.

These two methods have proved useful in the field when a customer has historical data in their data sources. Using the EXTRACT method with a start time in the past is an option. However, extracting that data quickly and analyzing it simultaneously has proven too much for the server to do simultaneously. In normal steady state operations, there is an aggregation interval that gives the server time to do both things without issue. To solve this problem, the administrator can use EXTRACT_ONLY to grab data from data sources. After that process completes, you can use REPLAY_RAW to place the data in the database and build the statistical model. After the REPLAY_RAW is complete, the administrator can use the EXTRACT method to return the server to steady state extraction.

IBM Training

Notifications about extraction progress

- Server sends status messages on its training



Sev	FirstOccurrence	Summary	LastOccurrence	Count
	6/12/14 3:55:41 PM	Topic Data Source started for topic: TRAINING, ready to start data loading	6/12/14 3:55:41 PM	1
	6/12/14 3:58:20 PM	Extractor started for topic: TRAINING and datasource: temp	6/12/14 3:58:20 PM	1
	6/12/14 4:33:56 PM	Received 0% of neccessary data for training to begin.	6/12/14 4:33:56 PM	3
	6/12/14 4:34:09 PM	Received 25% of neccessary data for training to begin.	6/12/14 4:34:09 PM	3
	6/12/14 4:34:18 PM	Received 50% of neccessary data for training to begin.	6/12/14 4:34:18 PM	3
	6/12/14 4:34:27 PM	Received 75% of neccessary data for training to begin.	6/12/14 4:34:27 PM	3
	6/12/14 4:34:44 PM	New model training started with data between: 2013-05-18 09:15:00.00...	6/12/14 4:34:44 PM	3
	6/12/14 4:34:44 PM	Received 100% of neccessary data for training to begin.	6/12/14 4:34:44 PM	2
	6/12/14 4:35:41 PM	Finished Training. New model produced.	6/12/14 4:35:41 PM	3

- After initial training complete, a once-a-day retraining message is received

Notifications about extraction progress

Whatever method you choose to extract data, the system posts messages to OMNIbus about its progress and how much data was retrieved. After the server has 100% of the necessary data, it creates a new model that can be used to score data and create anomalies. You get a daily message that a new model was produced.

Pause and resume mode

- Analyzing historical data requires special treatment
  - Prevent new data from being compared to old models
- Paused data loading while model is being built
  - Less data gets buffered in memory
  - Data is evaluated to the nearest model
- Default is to rebuild model every day
  - 40 days of data
  - 28-day training window
  - 13 models get built
  - Can take a long time
- In steady state mode, only pause for the first model

Pause and resume mode

Analyzing historical data presents special problems. It is much easier to extract data than it is to analyze it. If you leave these two processes running in parallel, data gets extracted much faster than models get updated. This problem causes extracted data to be evaluated against models that are out of date. To prevent false anomalies in historical data, data extraction is paused when models are being built. This action better reflects what would happen if data is being extracted in real time. It makes better use of server memory and ensures that data gets evaluated against the most recent model. However, this process slows down the analysis of historical data, especially when you have more data than is needed to train the server.

For example, if you have 40 days of data and need only 28 days of training, the system creates 13 models before all the data is used. If it takes hours of computing for each model, then the complete analysis can take some time. The most effective way to reduce this total time is to increase the trainingDelayDays server property. When looking at real-time data, this pause occurs only for the first model that is built by the server. Because the consumption of data is slowed considerably by the aggregation interval, the system can keep up with model building and data extraction simultaneously.

IBM Training

Housekeeping

- Server keeps raw extract files for largest <*detector*>.maxTrainingWindowDays
  - Allows the restoration of server state if necessary
- Database housekeeping
  - Metric data kept for 15 days
    ```
    ./admin.sh set -t=<topic> system.metric.retention.days  <# of days>
    ```
  - Alarm history data is kept for 180 days
    ```
    ./admin.sh set -t=<topic> system.alarm.history.retention.days <# of days>
    ```
- Removing CSV files
  - Delete files that are read by analytics server and placed in good or bad subdirectories
  - Must be done manually or through CRON job

Housekeeping

Because the topic is data extraction, you might wonder about housekeeping and what happens to all the extracted data. The server keeps only a trainingWindowWeeks worth of data in the database and in the file system. As it collects data every day, it deletes the extracted files for data older than trainingWindowWeeks. It keeps this much data around so that you can run REPLAY_RAW in case of an emergency. By default, the database keeps only 15 days of historical data that shows in the charts that Predictive Insights creates. The history of the alarms that were forwarded from Predictive Insights is, by default, kept for only 180 days. You can change both of these defaults. Increasing the number of days does affect the amount of space needed in the database. If one of your data sources uses CSV files, you must create a process to delete the extracted files that are moved to the good or bad subfolders. A suggestion is to use CRON to run this process regularly.

Cleaning out extracted data

- Can completely clean out data from server
  - Remove test data
  - Change data model
- $TASP_HOME/bin/admin.sh cleanup
  - Prompts you if you want to clean out database
- Backs up important directories to **$TASP_HOME as a BAK*<timestamp>*** directory
  - Backs up extracted subdirectory to allow restoration of server state
- Does not delete alarms in OMNIbus

Cleaning out extracted data

The last topic about data extraction is the tool that you can use to delete all the extracted data from the server and the database. First, the cleanup process backs up important files if you want to use REPLAY_RAW to restore the statistical model. It then proceeds to delete the extracted data and data model in the database and on the server. After you run this command, you need to redeploy your data model. This cleanup process does not communicate with OMNIbus, and any Predictive Insights alarms need to be manually removed.

IBM Training

Topic properties that can be modified

./admin.sh set -t=<topic> <property> <value>
./admin set –t=TRAINING system.aggregation.interval 5

- flatLine.enabled: true
- flatLine.maxTrainingWindowDays: 14
- flatLine.retrainingIntervalMinutes: 1440
- granger.enabled: true
- granger.maxTrainingWindowDays: 3
- granger.retrainingIntervalMinutes: 1440
- variantInvariant.enabled: true
- variantInvariant.maxTrainingWindowDays: 28
- variantInvariant.retrainingIntervalMinutes: 1440
- relatedEvents.enabled: true
- relatedEvents.retrainingIntervalMinutes: 1440
- robustBounds.enabled: true
- robustBounds.maxTrainingWindowDays: 28
- robustBounds.retrainingIntervalMinutes: 1440
- nofm.error.count.min.size: 3
- nofm.error.count.window.size: 6

Anomaly Detectors

- **system.aggregation.interval: 15**
- **system.alarm.autoclear: true**
- **system.alarm.history.retention.days: 180**
- system.alarm.session.keepalive.hours: 6
- system.da.days.before.alarm.cleared: 7
- system.da.missing.intervals.before.alarm: 3
- system.max.metricgroups: 50
- **system.metric.retention.days: 15**
- system.omnibus.enabled: true
- system.timeZone: Default
- system.topic.allowNegativeNumbers: false
- ui.granularity.default.min: 0
- **ui.summary.dlg.attr.names: ???**
- **ui.summary.dlg.favorite.attr.name: ???**

Topic properties that can be modified

Each topic that Predictive Insights has been configured for has a series of properties that can be modified. Each anomaly detector can be altered from their default if there is a compelling argument to do so. Modifying these defaults should be considered with care. There are a number of system and UI parameters that can be changed also. Some are shown in bold type, indicating that they are good candidates for changing from their default values.

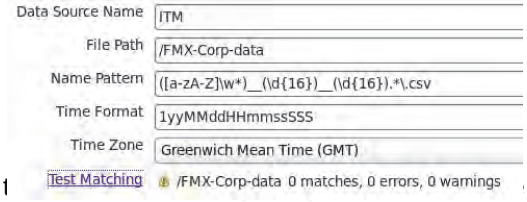
Lesson 3 Troubleshooting

In this lesson, you learn troubleshooting steps to ensure that the data mediation client is functional and the server is started and is extracting data.

IBM Training

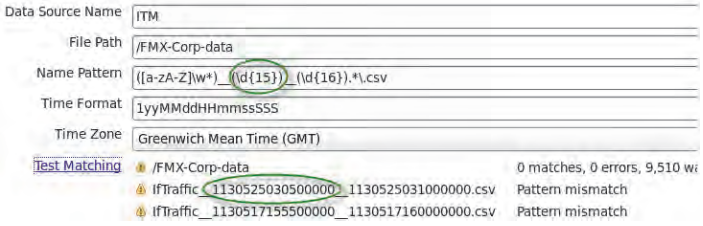
Troubleshoot mediation client issues

- Symptom: Files cannot be found by test matching



Solution: Ensure t adequate permissions

- Symptom: CSV source file not matching or preview problems



Solution: Check r

Troubleshoot mediation client issues

The data mediation tool can sometimes have problems with CSV data sources. The first issue is not having the right permissions on the CSV files for them to be read, and later, to be moved to the good or bad subdirectories. Making sure that the CSV files are owned by the analytics Linux user is the simplest method to solve this issue. The second issue with CSV files is the naming convention that they must follow and ensuring that the regular expression that is used to parse those file names is correctly defined. When selecting the Test Matching option, you learn whether the regular expression is correct by seeing whether a match was successful.

Common mediation tool problems

Symptom: Deploy option is not available

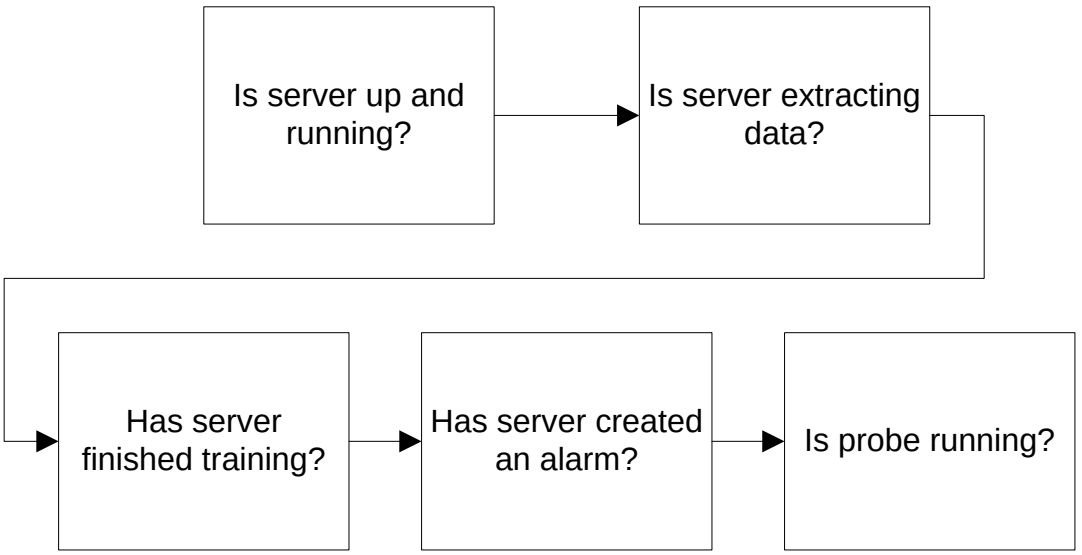
- Solution: Check to make sure that there are no outstanding errors
- Solution: Models to be deployed must have focus in the tree before selecting Deploy
- Solution: Click somewhere else inside the screen that has focus

Common mediation tool problems

One last issue that is commonly seen is that the deploy option in the client might be disabled when you select it. One reason is that the client detected an error about your data model or the GUI focus is not on a model that can be deployed. Often times, checking for possible errors and switching back and forth between screens can clear this issue.

IBM Training

Troubleshooting server issues



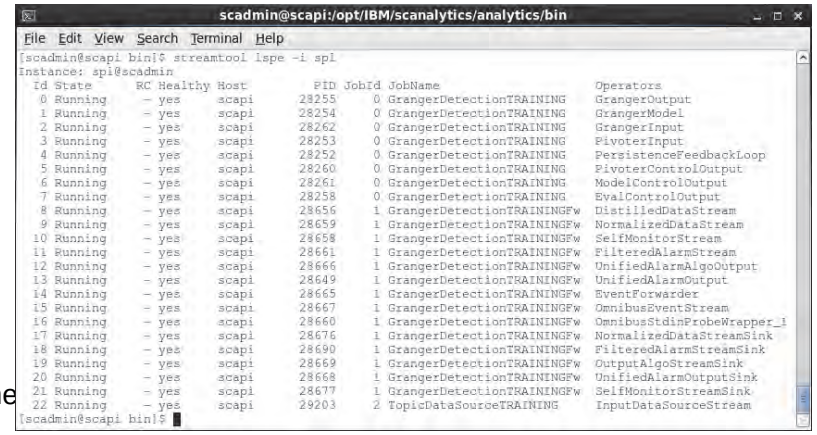
Troubleshooting server issues

Troubleshooting the server can be approached with a series of steps to see whether the process of creating an alarm is broken somewhere. To troubleshoot the server, start with ensuring that it is running and all its processes are healthy. Next, check to see whether it is extracting data regularly. After you know that it is gathering data, you must ensure that it has enough data to complete its training process. If it has completed its training, you can then see whether any alarms were generated. Finally, you check to see whether the probe starts successfully and is configured correctly.

IBM Training

Checking server status

- Check server status
  `streamtool lspe –i spl`
  Are all the processes running and healthy?



- If there are problems, ensure that the
  Must use InfoSphere Streams Java

Checking server status

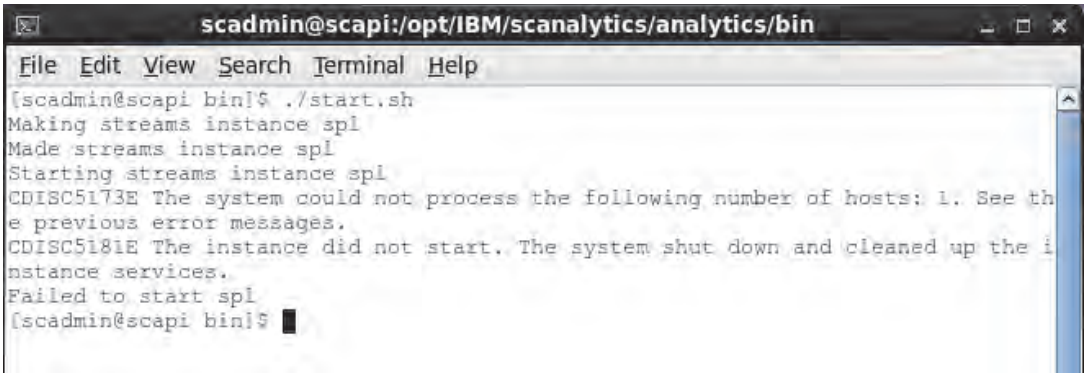
Before you try other troubleshooting steps, check to see whether the server is running. You should also verify that all the processes that are started are running and healthy. The command `streamtool lspe –i spl` (that is, an Lspe and spL) displays this information. If you have problems, and processes are not working correctly, check to see whether the Java for the Predictive Insights administrator (usually the Linux user **scadmin**) is pointing at the one that is used by InfoSphere Streams.

Problems with starting the server

Symptom: Message **Failed to start spl**

Solution: Run the following commands and try restart:

```
streamtool stopinstance -i spl –force
streamtool rminstance -i spl –noprompt
```



Problems with starting the server

If the server is not started, then attempt to start it with the `$TASP_HOME/bin/start.sh` command. Sometimes, especially after shutting down InfoSphere Steams, the server does not completely clean up all of its initiation files. When that happens, the server does not start and gives a failed to start message. When this error happens, use the two commands that are shown on the slide to ensure that the instance is stopped and that any instance files are removed. Attempt to start the server again after completing these commands.

IBM Training

Check whether server is extracting data

- Check for extracted files
  - Look in **$TASP_HOME/var/spool/topics/<TopicName>/extracted**
  - Verify whether files are created for each metric group or data source
- For CSV file extraction
  - Files are moved to **good** or **bad** subdirectory
  - Are there files in the **bad** directory?
- Extraction log file to review
  $TASP_HOME/log/*<TopicName>*/*Analytics<TopicName>*_log_SelfMonitorOperator.log

```
2016-11-16 22:32:26,209 INFO  [STATUS]   [Read] Group :  CputimeGroup; Interval :  (2016-03-01 00:20:00 - 2016-03-01 00:25:00); No Of Rows:  2
2016-11-16 22:32:26,210 INFO  [STATUS]   [Read] Group :  TotalbytesGroup; Interval :  (2016-03-01 00:15:00 - 2016-03-01 00:20:00); No Of Rows:  1
2016-11-16 22:32:26,211 INFO  [STATUS]   [Read] Group :  TotalbytesGroup; Interval :  (2016-03-01 00:20:00 - 2016-03-01 00:25:00); No Of Rows:  1
2016-11-16 22:32:26,251 INFO  [STATUS]   [Read] Group :  CputimeGroup; Interval :  (2016-03-01 00:25:00 - 2016-03-01 00:30:00); No Of Rows:  2
2016-11-16 22:32:26,252 INFO  [STATUS]   [Read] Group :  CputimeGroup; Interval :  (2016-03-01 00:30:00 - 2016-03-01 00:35:00); No Of Rows:  2
```

Metric group name

What time stamps were searched

Number of rows found

Check whether server is extracting data

With the server started and healthy, the next step is to check that data is being extracted from all your data sources. Regardless of data source (file-based or RDMS), the data that is retrieved from those sources is placed in a file in the $TASP_HOME/var/spool/topics/<topic>/extracted directory. Go into this directory and look for files with the following naming convention:

- *<datasource-name>_<metric-group-name>__<start-timestamp>__<end-timestamp>*.csv
- LegacyPerformance_LegacyNetworkIO__20130516-00-00UTC__20130516-00-05UTC.csv
- LegacyPerformance_LegacyNetworkIO__20130516-00-05UTC__20130516-00-10UTC.csv
- NetworkPerformance_NetworkIO__20130516-00-00UTC__20130516-00-05UTC.csv
- NetworkPerformance_NetworkIO__20130516-00-05UTC__20130516-00-10UTC.csv

If you use CSV files as a data source, you can also look for files in that directory location under the good and bad subdirectories. If there are many files in the bad directory, then a formatting issue is keeping those files from being extracted.

You can also check the **Analytics<*TopicName*>_log_SelfMonitorOperator.log** file and look for any errors that are being generated to determine why data is not being extracted.

IBM Training

Check for enough data to complete the training

Review alarms in Active Event List

	6/12/14 4:33:56 PM	Received 0% of neccessary data for training to begin.	6/12/14 4:33:56 PM	3
	6/12/14 4:34:09 PM	Received 25% of neccessary data for training to begin.	6/12/14 4:34:09 PM	3
	6/12/14 4:34:18 PM	Received 50% of neccessary data for training to begin.	6/12/14 4:34:18 PM	3
	6/12/14 4:34:27 PM	Received 75% of neccessary data for training to begin.	6/12/14 4:34:27 PM	3

Check for enough data to complete the training

You need a specific amount of data before training the server is completed. You can check the progress of this training easily by review the status alarms that are viewable in the Active Event List.

Check whether alarms were generated

- Anomalies are generated after the server is trained
  - Anomalies lead to alarms
- Check to see whether alarms were generated
  - $TASP_HOME/spl/instance/log/GrangerDetectionInstance1Fw/GrangerDetection<Topic>Fw_Sink_FilteredAlarmStreamSink.log
  - Look for entries with GCD AND do not have SUPPRESS
    ```
    grep ^GCD GrangerDetection<Topic>Fw_Sink_FilteredAlarmStreamSink.log | grep -v SUPPRESS | wc -l
    ```
- No entries means no alarms were generated
- Entries means that you had alarms that should be in OMNIbus

Check whether alarms were generated

After confirming you have a server with enough data to be trained, you should check to see whether anomalies were found and alarms were generated. The **GrangerDetection<*Topic*>Fw_Sink_FilteredAlarmStreamSink.log** notes whenever an alarm was created and is sent to the probe. Run a search on this log file and look for the letters *GCD*. *GCD* is how an alarmed anomaly is tracked in the database. You also want to know that the alarm was not suppressed because of customer-specific suppression rules. If you do not find messages that meet these conditions, then no alarms were generated for OMNIbus to report on.

Check whether the probe is running

- Probe does not start until an alarm is generated
- Check that the Predictive Insights OMNIbus probe is running
  `ps -ef | grep nco_p_stdin`
  - This probe automatically starts when you get the first event to be sent to OMNIbus
- Reasons for <*defunct*> process
  - Missing requirements, specifically /usr/lib/libstdc++.so.5 and /usr/lib/libstdc++.so.6
  - NCHOME was set in analytics environment
- Check that you configured event forwarding with the correct ObjectServer details

Check whether the probe is running

If you found alarms in the **GrangerDetection<*Topic*>Fw_Sink_FilteredAlarmStreamSink.log** file but not in OMNIbus, then you should check the Predictive Insights probe. This probe is started by the Predictive Insights software upon the creation of the first alarm. Check to see whether the process is started and running. If the process comes back as <defunct>, one of two things might be wrong. You are either missing libstdc++ libraries or the Predictive Insights administrative environment is misconfigured. Most likely, you have declared NCHOME, and the probe is confused. If the probe is running and still no alarms, ensure that it is forwarding those alarms to the correct OMNIbus server.

IBM Training

Working with support

If you have unresolvable problems, collect server data
`$TASP_HOME/bin/collect.sh`

- Provides good information to get to the root cause of a problem
- Generates a subdirectory of useful data and logs
  - collect_output
  - Create an archive and attach to support ticket

```
drwxrwxr-x 3 scadmin scadmin   4096 Nov 17 18:08 ..
-rw-rw-r-- 1 scadmin scadmin   6586 Nov 17 18:09 alarms.txt
-rw-rw-r-- 1 scadmin scadmin   8349 Nov 17 18:09 AnalyticsTRAINING.spl
-rw-rw-r-- 1 scadmin scadmin 197948 Nov 17 18:09 collect.log
-rw-rw-r-- 1 scadmin scadmin  30293 Nov 17 18:09 mediationtool.log.tgz
-rw-rw-r-- 1 scadmin scadmin   9306 Nov 17 18:09 start_stop.log
-rw-rw-r-- 1 scadmin scadmin 346592 Nov 17 18:09 TRAINING_datasourcelogs.tgz
-rw-rw-r-- 1 scadmin scadmin  15874 Nov 17 18:09 TRAINING.pamodel
```


Working with support

If you cannot troubleshoot the problem, start working with IBM support. Run the **collect.sh** script, which collects useful information about your server that can help resolve your issues. This script creates a subdirectory that is called **collect_output**. Archive this directory and attach it to a support ticket.

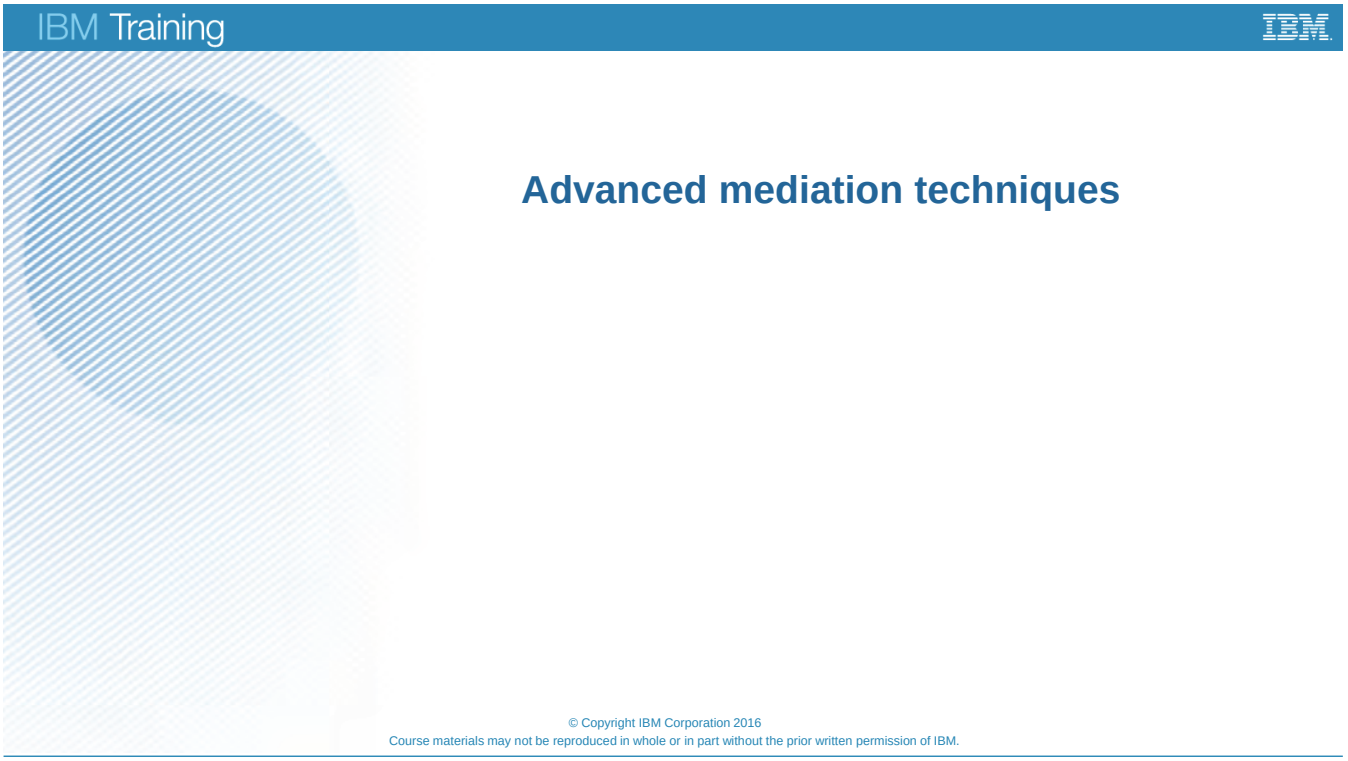
IBM Training

Unit summary

- Fix common problems with mediation tool
- Remediate startup issues with analytics server
- Troubleshoot issues with extraction
- Debug problems with alarms

Unit summary

Unit 7 Advanced mediation techniques



This unit describes the tools and techniques to be used to mediate data that is not in the correct format for consumption by Predictive Insights.

IBM Training

Unit objectives

- Describe the third-party tool logstash
- Build logstash conf files for mediating data
- Describe mediation considerations

Unit objectives

Lesson 1 Logstash

You are introduced to the open source application logstash.

IBM Training IBM

Logstash vitals

- Logstash is an open source log management solution
  - Apache 2.0 license
- Consists of two applications
  - Logstash is used to consume and annotate logging messages
  - Elasticsearch is a web server that collects, displays, and reports on annotated log messages
  - Similar features to SCALA
- Logstash details
  - Built on Ruby
  - Needs a recent Java Runtime Environment (JRE) to run
  - Expand archive and you are ready to begin

Logstash vitals

The open source software logstash (it is usually spelled in all lowercase) is a log management tool that is designed in concert with another application to centralize an organization's application, operating system, and network logging. It is licensed via Apache 2.0, making it easy to use and distribute.

When people discuss logstash, they might be referring to one of two applications. Logstash itself is a tool that is used to read various logging streams and to annotate the data in those streams. A second web server application is used to index and centralize these annotated streams. System administrators then view these messages in a GUI.

Logstash itself is built on Ruby (more specifically, JRuby) which requires a recent Java Runtime Environment be installed on the server. To install the software, you can download the latest release from http://logstash.net. You expand the archive file on the server and you should be able to use the software immediately.

Apache 2.0 Details

The Apache license is a free software license written by the Apache Software Foundation (ASF). The Apache license requires preservation of the copyright notice and disclaimer. Like other free software licenses, the license allows the user of the software the freedom to use the software for any purpose. It can be distributed and modified under the terms of the license, without concern for royalties.

The ASF and its projects release the software that they produce under the Apache license. Some non-ASF software is also licensed under this agreement. In October 2012, 8708 projects at

SourceForge.net were available under the terms of the Apache license. In a blog post from May 2008, Google mentioned that over 25% of the nearly 100,000 projects that are then hosted on Google Code were using the Apache license, including the Android operating system.

The ASF adopted the Apache license 2.0 in January 2004.

Why logstash

- SmartCloud Analytics Log Analysis
  - Using it to build Insight packs for various logs
  - Easier to work with than IBM's Annotated Query Language (AQL) for adding structure to messages
- SmartCloud Analytics Predictive Insights
  - Flat file data (possibly JDBC data) must be in correct CSV format for consumption
  - Various languages being used by field personnel (Awk, Perl, Python, and others)
  - Want to standardize on a single language to enable reuse of development efforts
  - Lightweight, easy to learn and deploy
  - Many features in plug-ins developed by community

Why logstash

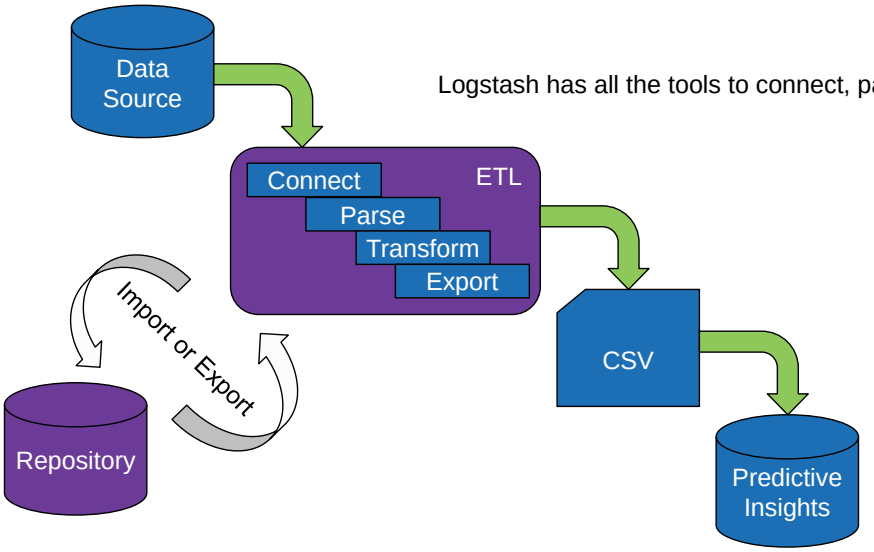
At this point, you might be asking; "Why are we using logstash?" Both Log Analysis and Predictive Insights have found useful reasons to direct customizations toward this tool. For Log Analysis, logstash has provided an easier means to build Insight Packs. An Insight Pack is a set of rules to annotate a specific log from an application, operating system, or network device. Using logstash, field engineers have been able to create custom Insight Packs much easier than the Annotated Query Language (AQL) that is part of the Big Data tool set.

Predictive Insights has found that the logstash tool set is useful in helping in the mediation of data that is not in the correct format for consumption. Predictive Insights expects data to be in a specific format and many data sources have been found that need a certain amount of manipulation before its data can be analyzed by the tool. These data sources could be rearranged in any number of different languages. However, if field engineers choose their own favorite scripting language, then the ability to reuse these mediations is reduced. By settling on a preferred language for Predictive Insights data mediation, it makes it easier to build a library of mediations that can be either reused directly or customized easily. Logstash has a large, predefined set of functions that you can use to read, manipulate, and publish data streams. Because of its open-sourced nature, an active community of users constantly adds to it.

Because both of these applications are part of the overall solution that is called Netcool Operations Insight, logstash is a useful tool that a field engineer should have for proof of concept and production deployments.

IBM Training

Extract, Transform, and Load tool needed



Extract, Transform, and Load tool needed

Predictive Insights needed a middleware tool to extract, transform, and load. The middleware must connect to a data source of various sorts and extract data from it. Next, it must transform that data into a correct format and then convert it into another data format, typically CSV files. When the data is in the correct CSV format, Predictive Insights can mediate the data and add it to its analysis. It is important that the ETL tool also be able to export and import data to a separate repository for local caching or permanent storage of useful information. Logstash currently provides a function to support most of these operations. Being built on JRuby and able to incorporate Java, it can be extended to support functions that are not currently available in its existing tool set.

IBM Training

Supporting information

- Main logstash website
  - https://www.elastic.co/products/logstash
  - Download application and documentation on all plug-ins
- The Logstash Book
  - http://www.logstashbook.com/
  - $10 for ePub version
- Logstash Forum for good Q&A
  - https://groups.google.com/forum/#!forum/logstash-users

Supporting information

You can find details on logstash at its main website. This site includes both the software and all the documentation on the plug-ins in its current release. There is also a useful publication call *The Logstash Book* that provides useful details on the tool and how it can be deployed as a central logging solution. While the book is focused more on a Log Analysis scenario, it provides useful information on how the tool works and how it can be extended with custom plug-ins. There is also a logstash forum where you can ask questions and find answers.

IBM Training

Key logstash functions

- Logstash is an event pipeline
  - Inputs → filters/codecs → outputs
  - Inputs capture events
  - codecs and filters modify them
  - outputs ship them somewhere
  - *Each event is treated individually (no product-included method for referencing previous events)*
- Inputs can declare a type for each stream, for example, `type=JazzSMSystemOut.log`
  - Used mainly for filter activation
  - They persist with the event
- You can add tags throughout event processing
  - Use for debugging purposes
  - Can be used to specify an order for event processing
- With conditionals, events can be processed in many ways

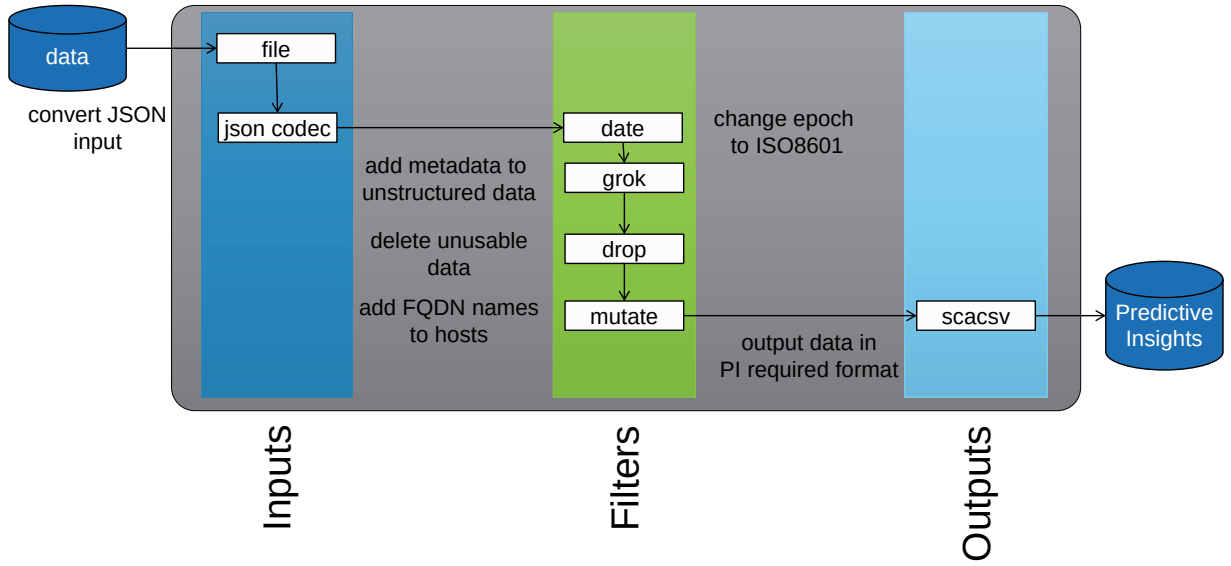
Key logstash functions

To understand how logstash works, you need to understand that it is an event pipeline. This idea means that events are sent through logstash either by it actively parsing a file or passively listening for information to be streamed to it. Events are processed via a three-step process of capturing input, processing the message, and then outputting the result. Inputs are where events are captured via the inspection or listening to a data source. Once an event is captured, it is processed via a series of filter or codecs that modify and annotate those messages to give them structure. Finally, the messages are sent on to some type of output. A key thing to understand about this processing is that each message is treated individually. There is no easy means to reference previous events or cache data for use by later events.

When reading inputs from various data sources, each of these streams can be typed to make them easier to work with later. For example, messages in the WebSphere log used by Jazz can be named to a specific type so they can be processed individually via the use of filters. This metadata persists with the message through out its time in the logstash processing. Additionally, you can add your own custom tags to a message as it is read and processed by logstash. These tags can be used for both debugging purposes and as triggers for IF/THEN logic. Conditionals and boolean logic can be used to allow events to be processed in many different ways.

IBM Training

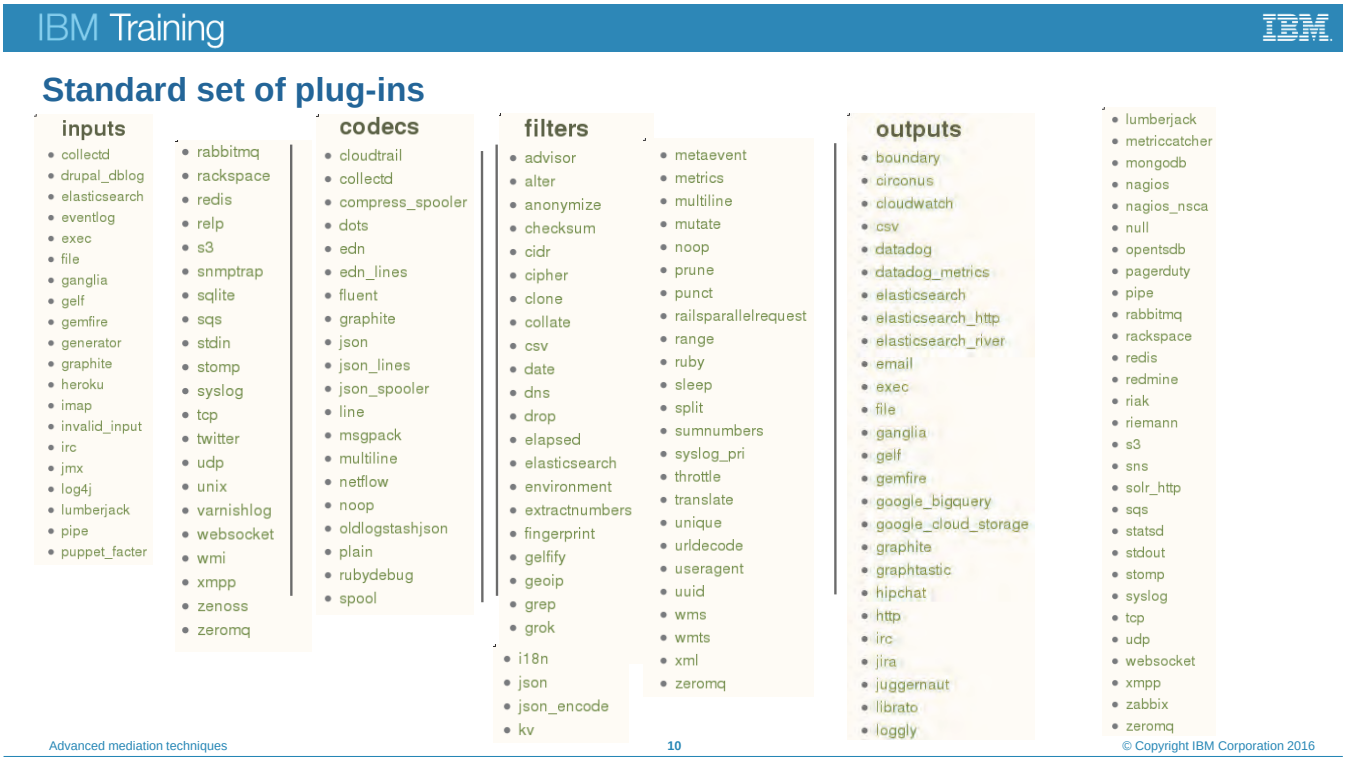
Sample logstash process



Example logstash process

Here is an example of how logstash can be used to process metric data from a flat file. In the Inputs section of the code, you use a file codec that is designed to read or monitor a file. Because the file uses JSON to store its data, a JSON codec is used to translate the JSON fields into logstash fields. With the data translated into fields, the message is handed to the Filters section of the code. Here, the data is cleaned up, standardized, and annotated.

The first step is to use the logstash date plug-in to convert the epoch time stamp into an ISO8601 format that is humans can read. Next, the raw data in the message is parsed and metadata structure added through a plug-in called *grok*. If data is unusable, Predictive Insights uses the *drop* plug-in to prevent future processing of the message. Finally, the host name in the message is updated to include its fully qualified host name through a plug-in that is called *mutate*. After the Filters section has completed its processing, the newly modified and annotated message is handed to the Outputs section. Here the message is sent to a custom plug-in called *scacsv* that sends the data into the correct format needed by Predictive Insights.



Standard set of plug-ins

An Important aspect of logstash is its ever growing list of plug-ins that are used to read, manipulate, and publish data. This list of functions includes both core functions that are released with logstash as well as community supplied methods that can be easily added to the logstash source.

IBM Training

Important plug-ins that are included with Predictive Insights

- Inputs
  - file: Typically reading an output file
  - stdin: Used for testing logstash conf files
- Filters
  - csv: Add field names to CSV input data
  - drop: Delete events that are not useful
  - date: Convert time stamp
  - grok: Add fields to data in unstructured text
  - mutate: Add, modify, or delete fields and the data within those fields
  - translate: Search and replace tool that uses an internal or external hash table
- Outputs
  - csv: Convert data in fields into CSV format
  - stdout: Used for testing

Important plug-ins that are included with Predictive Insights

Here is a subset of those functions that prove useful for Predictive Insights.

From an input perspective, you are often reading data files that must have data that is extracted and reformatted for use by Predictive Insights. With the file plug-in, you can read one or more files to extract messages from. With the stdin function, you can enter data into the logstash process from the command line. You can test your logstash configuration file by piping data to it with the following command:

```
head -100 test-data-file.dat | logstash -f test-configuration-file.conf
```

After messages are received and passed to the filters section, you have a choice of plug-ins to manipulate the data. For example, the csv plug-in adds field names to data that is already in a CSV-like format. The drop plug-in quickly deletes data that is not useful. The grok plug-in might be, by far, the most important of the plug-ins. It finds the important pieces of data from unstructured text and surrounds it with metadata. The mutate plug-in is useful because you can create, modify, and delete fields and the data within those fields.

The output section is where the completed manipulations to your message are sent to some location. For Predictive Insights, that destination is usually a CSV file but it could also be a database if you want. You use the csv plug in, in this situation, to export data to a flat file by using comma-delimited text. Like with stdin, the stdout plug-in is useful for testing.

IBM Training

Custom plug-ins

- Custom plug-ins can work with logstash
- Current custom plug-ins built on Ruby
  - scacsv
    Produces files complying with PI requirements related to header, and file naming.
  - genjdbc
    Logstash can read from database from JDBC

Custom plug-ins

An important feature for logstash is its openness for creating custom plug-ins. All the source code for the standard plug-ins is included with the release, making it easy to extend a particular plug-in to do something different. A number of plug-ins already have already been designed for Predictive Insights. The scacsv plug-in creates files with the correct format and header information as well as creates the correct file-naming convention. The genjdbc tools is used to access JDBC databases on a regular basis.

IBM Training

Installation

- Download logstash from https://www.elastic.co/downloads/logstash
  - Download the version with all the plug-ins
- Unpack in a directory of your choice (`$logstash_home`)
- Add logstash to your $PATH for convenience, if you want
- Obtain the SCAPI plug-in packages
  - Currently available at https://github.com/IBM-ITOAdev
  - Need to include a `--pluginpath` to `logstash` instantiation command

Installation

Installing logstash is trivial and can be completed in less than 5 minutes. The current release can be downloaded from the logstash website. A version of logstash is available with all the current plug-ins installed. Unpack this archive into an appropriate location and you are ready to use the tool. You might want to add the location of the logstash binary to your PATH variable to make it easier to start.

The two previously mentioned plug-ins built by Predictive Insights development are available on the Internet at GitHub. Search for IBM-ITOA and you should find them. Download the .rb files that define these plugins and place them in a directory on the server where logstash is running. When starting logstash, be sure to provide the path to this directory so the application can find these files.

IBM Training

Running logstash

- `$logstash_home/logstash -f sample.conf`
  sample.conf is a file with input, filter, and output plug-ins and logic for events entering this
- Using custom Predictive Insight plug-ins requires path location
  `$logstash_home/logstash -f sample.conf --pluginpath <custom-plug-in-directory>/scaLogstash`
- Use of $PATH can make this a bit shorter

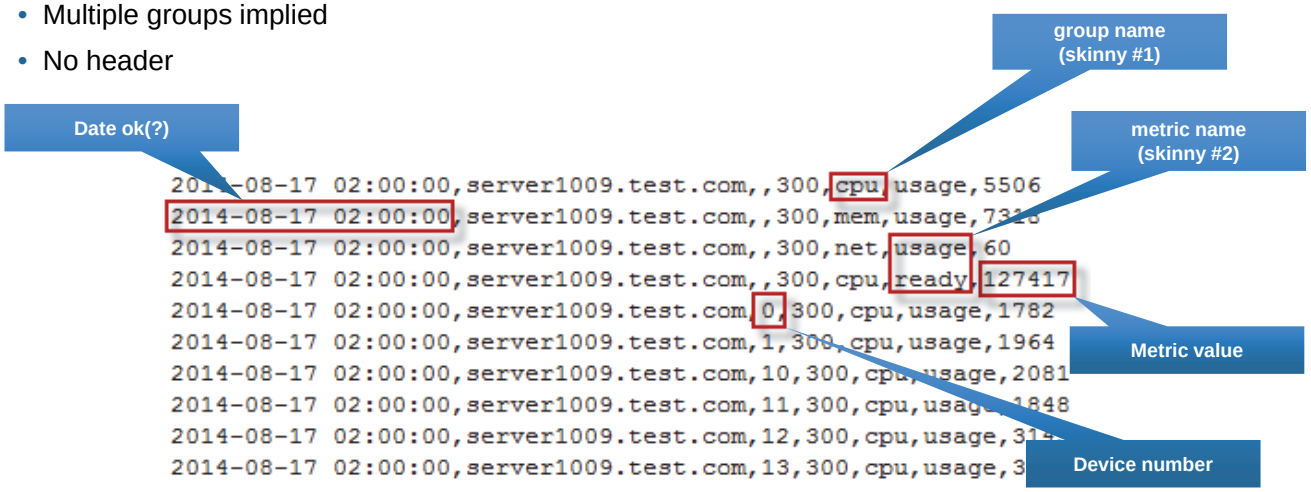
Running logstash

To run logstash, you invoke the logstash script in the binary directory where you installed the archive file. You point logstash at a configuration file that has the input, filter, and output actions you want to take against the messages that logstash sees. If you use the custom plug-ins that are created by development, you must provide there location with the `--pluginpath` option.

IBM Training

Example 1: Reviewing data file

- Skinny format
- Multiple groups implied
- No header



Advanced mediation techniques 15 © Copyright IBM Corporation 2016

Example 1: Reviewing data file

It is time now to look at an example of how you can use logstash to reorganize data for Predictive Insights. Here a data file has a number of issues that must be addressed before it can be mediated. The data looks acceptable from a comma-separated value perspective. However, upon closer inspection, there are problems:

- There is no header in this file to denote the meaning of the columns. Predictive Insights requires that a header line be defined as the first row of every CSV file that it mediates
- The file includes data on both CPU, mem, and net metrics. Note the group name column in the file. This type of data layout makes this file skinny.
- Not only is the file skinny based on a group name, it is also additionally skinny based on the metric name where there is the concept of usage and ready values.

The intent in the following slides is to break down this file and show how logstash works with these messages.

IBM Training

Example 1: Testing logstash

```
head -1000 test-data.txt | logstash –f test.conf | more
```

```
input {
  stdin {
  }
}

output {
  stdout {
  }
}
```

http://logstash.net/docs/1.4.2/inputs/stdin

http://logstash.net/docs/1.4.2/outputs/stdout

```
head -50 /opt/scapi-training-data/logstash-examples/example-skinny-data.csv | logstash -f skinny-example.conf
2015-02-04T17:02:22.505+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,,300,cpu,usage,5506
2015-02-04T17:02:22.508+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,,300,mem,usage,7318
2015-02-04T17:02:22.510+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,,300,net,usage,60
2015-02-04T17:02:22.511+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,,300,cpu,ready,127417
2015-02-04T17:02:22.512+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,0,300,cpu,usage,1782
2015-02-04T17:02:22.513+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,1,300,cpu,usage,1964
2015-02-04T17:02:22.514+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,10,300,cpu,usage,2081
2015-02-04T17:02:22.515+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,11,300,cpu,usage,1848
2015-02-04T17:02:22.516+0000 scapi.tivoli.edu 2014-08-17 02:00:00,server1009.test.com,12,300,cpu,usage,3143
```

Host which processed record

Time stamp when message or record was processed

Actual record

Example 1: Testing logstash

This example is a simple test of how logstash takes in messages and then sends them out. This example just takes the first 100 lines of the previous data file and sends them through logstash. Logstash is making use of a simple conf file. It takes data from standard inputs and immediately sends it to standard output. It takes this input message:

```
2014-08-17 02:00:00,plysvmw1009.rxcorp.com,,300,cpu,usage,5506
```

and immediately turns it into this message:

```
2014-09-08T15:04:28 0c312215850 2014-08-17
02:00:00,plysvmw1009.rxcorp.com,,300,cpu,usage,5506
```

See that it added to the message a system clock time stamp of when it processed the message and the host name of the server that processed it.

IBM Training

Example 1: Publishing data in structured format

```
output {
  stdout {
    codec => rubydebug
  }
}
```

Publishes data using ruby awesome_print

```
head -50 /opt/scapi-training-data/logstash-examples/example-skinny-data.csv | logstash -f skinny-example.conf
{
       "message"    "2014-08-17 02:00:00,server1009.test.com,,300,cpu,usage,5506\r",
      "@version"    "1",
    "@timestamp"    "2015-02-04T17:09:35.766Z",
          "host"    "scapi.tivoli.edu"
}
```

```
output {
  stdout {
    codec => json
  }
}
```

Publishes data as json

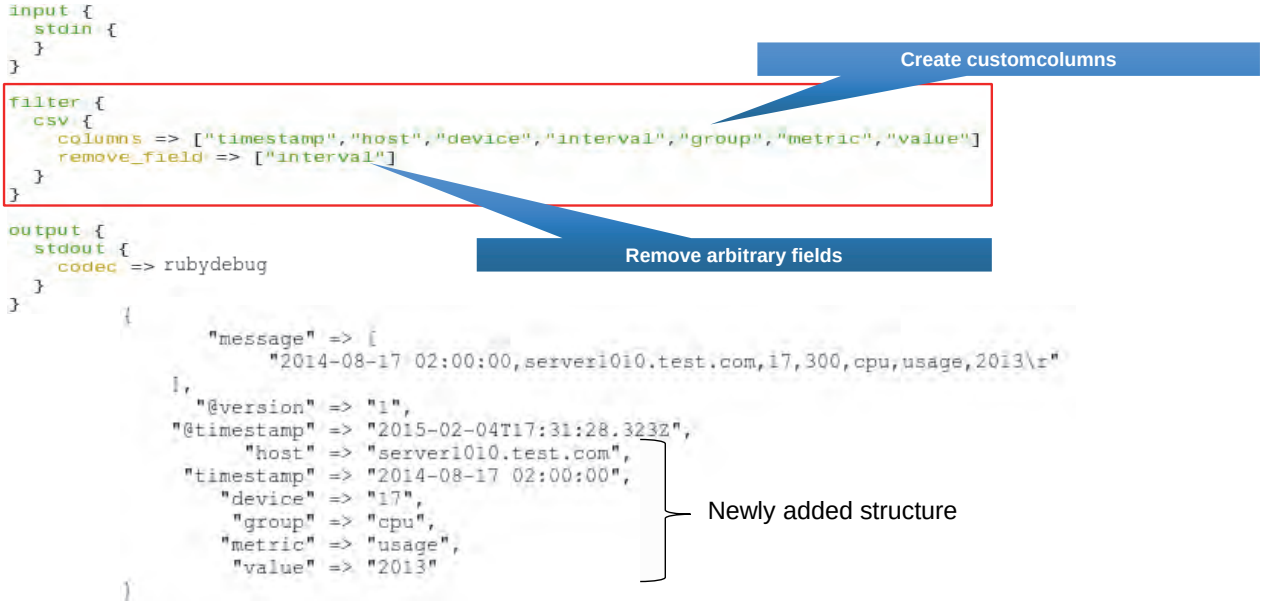
```
head -50 /opt/scapi-training-data/logstash-examples/example-skinny-data.csv | logstash -f skinny-example.conf
{"message":"2014-08-17 02:00:00,server1009.test.com,,300,cpu,usage,5506\r","@version":"1","@timestamp":"2015-02-0
4T17:15:29.378Z","host":"scapi.tivoli.edu"}{"message":"2014-08-17 02:00:00,server1009.test.com,,300,mem,usage,731
8\r","@version":"1","@timestamp":"2015-02-04T17:15:29.457Z","host":"scapi.tivoli.edu"}{"message":"2014-08-17 02:0
0:00,server1009.test.com,,300,net,usage,60\r","@version":"1","@timestamp":"2015-02-04T17:15:29.457Z","host":"scap
i.tivoli.edu"}{"message":"2014-08-17 02:00:00,server1009.test.com,,300,cpu,ready,127417\r","@version":"1","@times
tamp":"2015-02-04T17:15:29.458Z","host":"scapi.tivoli.edu"}{"message":"2014-08-17 02:00:00,server1009.test.com,0,
```


Example 1: Publishing data in structured format

To help with the interpretation of the output, you can use codecs to send the data in various formats. For example, you could choose JSON to send the data to another application that readily accepts JSON input. The rubydebug format is useful for testing because it has good print capabilities when sent to standard output.

IBM Training

Example 1: Adding structure through a filter and plug-in



Example 1: Adding structure through a filter and plug-in

In the previous slides, all you did was accept input and immediately sent this information with some formatting. In this step, you are going to begin adding structure to the data by using the filter section of the conf file. This example uses the csv plug-in. Because the data in the file is in a CSV format, you can easily add a field name to each of the items in the message so that you can later manipulate them by referencing this name.

With the csv plug in, you assign a column name to each item in the message based on its position. With the csv plug-in, you can remove useless data to make things simpler, if necessary. Note the change in the standard output. Not only do you have the original message, you now have each individual value listed and associated to a new field name. Think of these field names as variables that you can now use in your script.

IBM Training

Example 1: Sending data to a file

Produces most of the necessary information

- No header
- Data not separated by group

```
output {
  csv {
    fields => ["timestamp","host","device","group","metric","value"]
    path => "./output.csv"
  }
}
```

```
[scadmin@scapi Desktop]$ more ./output.csv
2014-08-17 02:00:00,server1009.test.com,,cpu,usage,5506
2014-08-17 02:00:00,server1009.test.com,,mem,usage,7318
2014-08-17 02:00:00,server1009.test.com,,net,usage,60
2014-08-17 02:00:00,server1009.test.com,,cpu,ready,127417
2014-08-17 02:00:00,server1009.test.com,0,cpu,usage,1782
2014-08-17 02:00:00,server1009.test.com,1,cpu,usage,1964
2014-08-17 02:00:00,server1009.test.com,10,cpu,usage,2081
2014-08-17 02:00:00,server1009.test.com,11,cpu,usage,1848
2014-08-17 02:00:00,server1009.test.com,12,cpu,usage,3143
2014-08-17 02:00:00,server1009.test.com,13,cpu,usage,3869
2014-08-17 02:00:00,server1009.test.com,14,cpu,usage,3498
2014-08-17 02:00:00,server1009.test.com,15,cpu,usage,3240
```

Data still skinny

Example 1: Sending data to a file

It is important that you have other output options than standard out. Here you use the output plug-in called csv. This plugin is different from the csv plugin used in the filter section. Unlike the csv plugin in the filter section that annotated field names, this csv function outputs comma delimited data to a file. As you can see, you have not changed the file very much from the original data.

IBM Training

Example 1: Using conditionals

Advanced mediation techniques 20 © Copyright IBM Corporation 2016

Example 1: Using conditionals

Now you have data within the message that is broken up into individual fields by the csv plug-in. You can now use those variables to create IF-THEN logic. You can divide the files by their group names. With these three IF statements, you can create three separate files for each data group: mem, net, and cpu. This step helps make the data less skinny for Predictive Insights mediation. Although you broke up the data by group with this process, you still have not added the required header, and the file name does not have the required start and end time stamps that Predictive Insights needs.

Although you have reduced the skinniness of the data, there is still a problem. The cpu group still has a skinny component associated with it. Within the cpu group are metrics that define *usage* and *ready*. Refer to the output in the slide to see this issue. This situation must be addressed by further processing with the logstash configuration file.

IBM Training

Custom plug-in: scacsv

Produces files complying with PI requirements related to header and file naming

```
output {
    scacsv {
        fields => … # array (required)
        header => … # array (optional), default: {}
        path => … # string (required)
        group => ... # string (required)
        time_field => … # string (required)
        time_field_format => … # string (required)
        timestamp_output_format => … # string (required)
        flush_interval => … # number (optional)
        }
}
```

Details

fields (required)	Fields to be produced
header (optional)	Header strings to produce, to override the field names that would be used by default
path (required)	Path of output file This is a temporary file and is renamed to PI standard
group (required)	Name of group to be used as a prefix for the final file name (<group>__<starttime>__<endtime>.csv)
time_field	The field that denotes the time stamp of the message
time_field_format	The format of the time information in the time_field using Java time stamp format
timestamp_output_format	The format of the time stamp to be used in the CSV file naming. Use Java time stamp format
flush_interval	Amount of time in seconds to wait before flushing, closing, and renaming a file, if there has been no data received. Default is 60 seconds

Custom plug-in: scacsv

To help with getting data files in the necessary format for Predictive Insights, you can use the previously mentioned scacsv plug-in that was created specifically for this application. This plug-in is similar to the csv plug-in used in the output section of the configuration file. However, it adds the header line to the file and creates the start and end time stamps for the file name.

This plug-in needed a flush_interval for these reasons:

- to define when it should stop looking for data to add to the file
- To then close the temporary file
- To rename the file with its corresponding time stamps that reflect the data within the file

IBM Training

Example 1: Replacing csv with scascv

```
output {
        if "mem" in [group] {
                scacsv {
                        fields => ["timestamp","host","subcomponent","metric","value"]
                        path => "./mem.csv"
                        group => "memory"
                        time_field => "timestamp"
                        time_field_format => "yyyy-MM-dd HH:mm:ss"
                        timestamp_output_format => "yyyy_MM_dd-HH_mm_ss"
                        }
        }
        if "cpu" in [group] {
                scacsv {
                        fields => ["timestamp","host","subcomponent","metric","value"]
                        path => "./cpu.csv"
                        group => "processor"
                        time_field => "timestamp"
                        time_field_format => "yyyy-MM-dd HH:mm:ss"
                        timestamp_output_format => "yyyy_MM_dd-HH_mm_ss"
                }
        }
        if "net" in [group] {
                scacsv {
                        fields => ["timestamp","host","subcomponent","metric","value"]
                        path => "./net.csv"
                        group => "network"
                        time_field => "timestamp"
                        time_field_format => "yyyy-MM-dd HH:mm:ss"
                        timestamp_output_format => "yyyy_MM_dd-HH_mm_ss"
                }
        }
}
```

Additional fields

Output files

```
memory__2014_08_17-02_00_00__2014_08_17-02_15_00.csv
network__2014_08_17-02_00_00__2014_08_17-02_15_00.csv
processor__2014_08_17-02_00_00__2014_08_17-02_15_00.csv
```

```
[scadmin@scapi logstash]$ more processor__2014_08_17-02_00_00__2014_08_17-02_20_00.csv
timestamp,host,subcomponent,metric,value
2014-08-17 02:00:00,server1009.test.com,,usage,5506
2014-08-17 02:00:00,server1009.test.com,,ready,127417
2014-08-17 02:00:00,server1009.test.com,0,usage,1782
2014-08-17 02:00:00,server1009.test.com,1,usage,1964
2014-08-17 02:00:00,server1009.test.com,10,usage,2081
```

cpu file still skinny! Need to pivot

Example 1: Replacing csv with scascv

By using the scacsv plug-in instead of csv, you can see the results of the file name and the addition of a header on the file's first line.

IBM Training

Custom plug-ins for Predictive Insights

- IBM Developerworks
  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20SmartCloud%20Analytics%20-%20Predictive%20Insights/page/Logstash%20based%20Mediation
- Custom plug-ins on web
  https://github.com/IBM-ITOAdev/Logstashplug-ins

Custom plug-ins for Predictive Insights

These custom plug-ins and more are available through IBM developerWorks. You can access the code on the web.

IBM Training

Example 2: Reviewing data

- A series of report files
  Each file is unique to a specific host
- Metadata in header
- Selection of header and data lines
- Simple format clean up of individual fields, for example, G, %, -

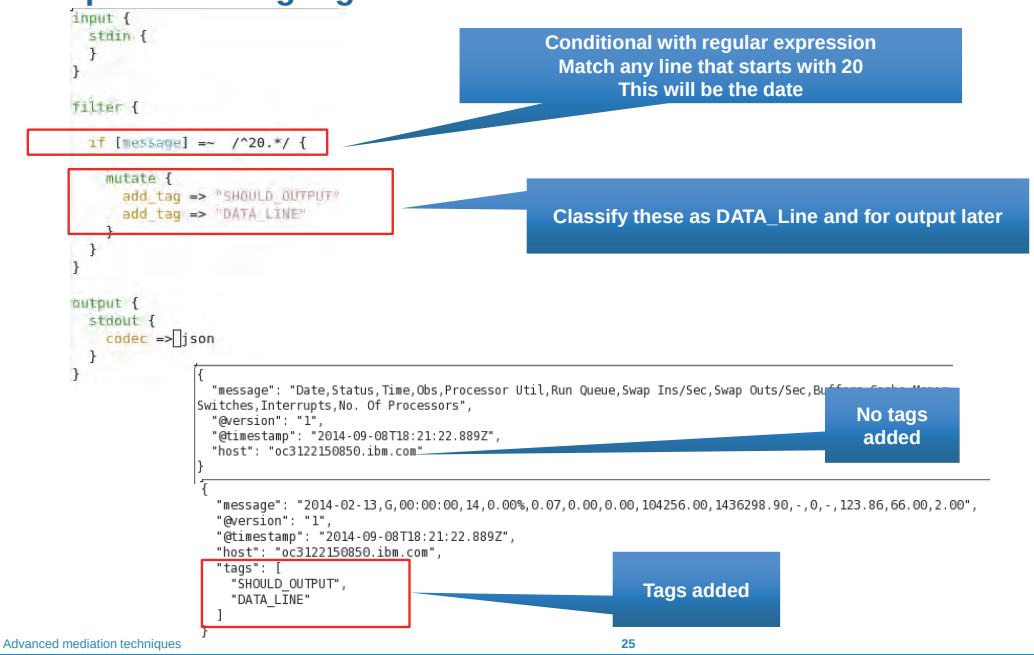
```
Report Type:  Processor/Memory
Rules in Effect: Performance
Frequency:  15 Minute,Server: ausSomeBankserver1,Frequency: Fifteen Minute Report,
First Period: Feb 1 (Sat), 2014,Last Period: Latest day with data,
Enterprise,Customer,Sub Client ID1,Sub Client ID2,OS,Server
Austria,SomeBank,MLC509,Production,Linux,ausSomeBankserver1
Date,Status,Time,Obs,Processor Util,Run Queue,Swap Ins/Sec,Swap Outs/Sec,Buffers,Cache,Memory Average % Used,Memory Size
2014-02-13,G,00:00:00,14,0.00%,0.07,0.00,0.00,104256.00,1436298.90,-,0,-,123.86,66.00,2.00
2014-02-13,G,00:15:00,15,0.07%,0.33,0.00,0.00,113909.87,1430425.90,-,0,-,132.67,75.13,2.00
2014-02-13,G,00:30:00,15,0.20%,0.33,0.00,0.00,123157.07,1412973.60,-,0,-,139.67,82.53,2.00
2014-05-02,G,22:45:00,15,0.00%,0.13,0.00,0.00,272929.06,1265073.60,-,0,-,125.00,71.33,2.00
2014-05-02,G,23:00:00,15,0.00%,0.20,0.00,0.00,271376.00,1263068.00,-,0,-,121.60,72.73,2.00
2014-05-02,G,23:15:00,15,0.13%,0.07,0.00,0.00,272999.20,1253161.10,-,0,-,120.33,71.33,2.00
2014-05-02,G,23:30:00,15,0.00%,0.00,0.00,0.00,274203.72,1243775.20,-,0,-,121.80,74.33,2.00
2014-05-02,G,23:45:00,15,0.07%,0.07,0.00,0.00,280827.72,1241951.20,-,0,-,118.47,72.33,2.00
```


Example 2: Reviewing data

Now you look at a different example. Here you see a data file that is a mix of free form data as well as CSV entries. The example in the slide resembles a report that was generated by some application. Each report is specific to single host. There is metadata in the header that defines the host name. Underneath the header is a series of data lines that are in a CSV format. As an extra step, you are going to clean up the data by removing some values like percentage signs and dashes.

IBM Training

Example 2: Adding tags to data for various reasons



Example 2: Adding tags to data for various reasons

Before diving too deep into this example, you can use optional tags to help process these messages. In this example, you tag the messages in the previous file that have useful data that you want to publish.

In the filter section, a piece of code looks at each message. When it finds one that begins with a 20 (as in 2014), it assumes that line has useful information. Here, you use the mutate plug-in to add some data to the tags field that each message has by default. The example adds two tags to any interesting message that say SHOULD_OUTPUT and DATA_LINE. In the two sample outputs, the first has a message that begins with Date, Status, Time. This message does not have tags because it did not pass the IF-THEN logic. The second message, which starts with 2014-04-13, does have the tags. These tags can now be used later in the script for other actions.

IBM Training

Example 2: Structuring unstructured data with grok

- Grok is one of the most important plug-ins for use with PI
- Parses arbitrary text and structure it
- Uses high-level regex objects called patterns
  - Some patterns are more complex than others
    %DATA versus %DATESTAMP
  - High-level patterns can call lower level patterns
- The syntax for a grok pattern is %{SYNTAX:SEMANTIC}
  - SYNTAX is the name of the pattern that will match your text
  - SEMANTIC is the identifier you give to the piece of text being matched

- Refer to https://grokdebug.herokuapp.com/patterns# to see all the current grok patterns
  - Time stamps
  - Syslog messages
  - Paths and URIs
  - Networking

Example 2: Structuring unstructured data with grok

In the next slides, you are introduced the grok plug-in. It is probably the most important of the logstash plug-ins. Grok parses arbitrary text and adds structure to it. It can take your messages and find the time stamp, host name, important error codes, and other text and place that data into separate fields. Grok takes regular expression and creates higher-level objects that make your work easier. These objects are referred to as ***patterns***. Instead of creating a regular expression for parsing the year, month, day, and time from a time stamp, you can call the grok pattern DATESTAMP that contains all the regular expressions to do this task for you. Some patterns are collections of lower-level patterns. By stepping down through these patterns, you eventually come to the regular expressions that these patterns use.

Grok uses a syntax that combines a pattern with a field name. This pattern is usually shown as %{SYNTAX:SEMANTIC}. The SYNTAX is the pattern that you are trying parse in a location in the message. The SEMANTIC is the field name that you give to whatever is retrieved from search. Examples in the upcoming slides help clarify this idea.

Many patterns are included with the current logstash release. You can investigate these patterns at https://grokdebug.herokuapp.com/patterns#. You can use these patterns to select and parse time stamps, syslog messages, URLs, and networking data from various messages.

IBM Training

Example 2: Extracting year, month, day, and time from message

Example 2: Extracting year, month, day, and time from message

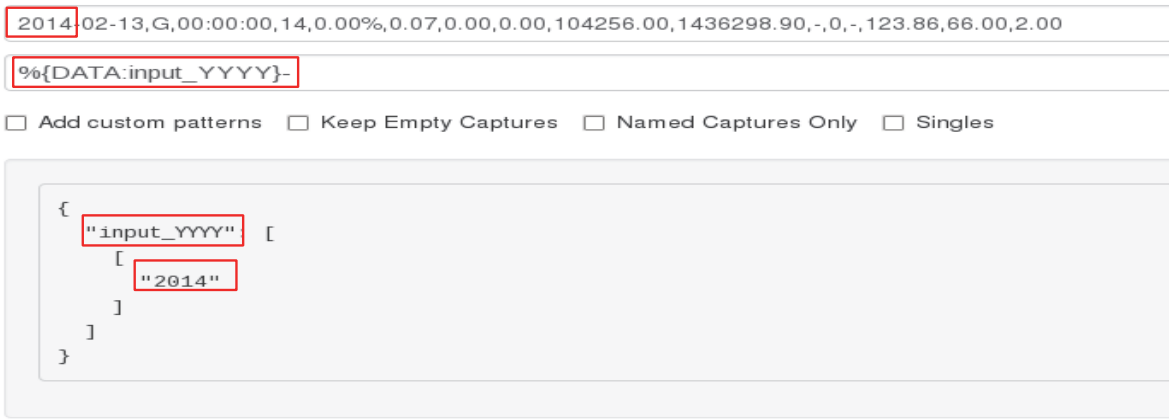
Here is an example of the grok plug-in taking a message and parsing its details to create a time stamp. Here the date and time are separate entities in the message and need to be extracted. This example uses the generic %DATA pattern, which captures the characters between known entities (like the beginning of the line or a dash). Notice the dashes in the grok pattern between each selection. Those dashes represent the actual dashes in the message.

Important: The existing grok patterns work best when the underlying message has a discernible structure in which to parse. Most log messages have a format that an administrator can understand. If the messages were truly free form and did not have an underlying format, you can either use your own regular expressions with grok or even create your own custom patterns.

IBM Training

Example 2: Testing grok patterns

- Not many tools to help you build grok patterns
  http://grokdebug/herokuapp.com is useful
- From website you can build grok patterns to test on messages
  Displays what was matched and what semantic (field) name it will be given



Advanced mediation techniques 28 © Copyright IBM Corporation 2016

Example 2: Testing grok patterns

Logstash is only 4 years old, and it has few tools to work with it. This situation is evident with working with grok. Currently, there seems to be only one tool available to you to build grok patterns. A website is available where you can enter a free-form message and then define a series of grok patterns to test what, if anything, they can extract. The site provides an output of what it extracted and what fields those data are associated with.

IBM Training

Example 2: Capturing time stamp with grok

```
filter {

  if [message] =~  /^20.*/ {

    mutate {
      add_tag => "SHOULD_OUTPUT"
      add_tag => "DATA_LINE"
    }

    grok {
      match => ["message", "%{DATA:input_YYYY}-%{DATA:input_MM}-%{DATA:input_DD},G,%{DATA:input_hhmmss},"]
    }
  }
}
```

Both plug-in actions are within the IF clause

```
{
  "message": "2014-02-13,G,00:00:00,14,0.00%,0.07,0.00,0.00,104256.00,1436298.90,-,0,-,123.86,66.00,2.00",
  "@version": "1",
  "@timestamp": "2014-09-08T18:59:26.792Z",
  "host": "oc3122150850.ibm.com",
  "tags": [
    "SHOULD_OUTPUT",
    "DATA_LINE"
  ],
  "input_YYYY": "2014",
  "input_MM": "02",
  "input_DD": "13",
  "input_hhmmss": "00:00:00"
}
```


Example 2: Capturing time stamp with grok

Now you can add the grok plug-in to the filter. For any message that begins with 20, you add the two tags SHOULD_OUTPUT and DATA_LINE and then parse the message. The code uses grok patterns to search the message for the year, month, day, and time. See how each field is created for the message and a value associated with it. If there was a problem and the pattern returned nothing, then the value for the field would be null.

IBM Training

Example 2: Cleaning up data

```
filter {

  if [message] =~  /^20.*/ {

    mutate {
      add_tag => "SHOULD_OUTPUT"
      add_tag => "DATA_LINE"
    }

    grok {
      match => ["message", "%{DATA:input_YYYY}-%{DATA:input_MM}-%{DATA:input_DD},G,%{DATA:input_hhmmss},"]
    }

    mutate {
      gsub => [ "message", "\,-\,", ",,",
                "message", "\%\,", ",," ]
    }
  }
}
```

Convert a string field by applying a regular expression and replacement values. Commas and percent signs are special characters and must be preceded with backslash

```
2014-02-13,G,00:00:00,14,0.00%,0.07,0.00,0.00,104256.00,1436298.90,-,0,-,123.86,66.00,2.00
{
  "message": "2014-02-13,G,00:00:00,14,0.00,0.07,0.00,0.00,104256.00,1436298.90,,0,,123.86,66.00,2.00",
  "@version": "1",
  "@timestamp": "2014-09-08T19:45:54.652Z",
  "host": "oc3122150850.ibm.com",
  "tags": [
    "SHOULD_OUTPUT",
    "DATA_LINE"
  ],
  "input_YYYY": "2014",
  "input_MM": "02",
  "input_DD": "13",
  "input_hhmmss": "00:00:00"
}
```

Advanced mediation techniques 30 © Copyright IBM Corporation 2016

Example 2: Cleaning up data

The *mutate* plug-in has options that you can use to add and remove fields as well as alter the values in those fields. Here the mutate function uses gsub, which can use a regular expression search and replace. In the slide example, you are removing the dash and the percent signs from the message. Because regular expressions are searching the message, the escape character (in this case, a backslash [****]) ensures that the comma and percent signs are treated as normal characters. The comma and percent signs have special meaning in regular expressions.

IBM Training

Example 2: Adding field names to the items in the message

```
    csv {
        columns => [
             "Date","Status","Time","Obs","Processor Util",
             "Run Queue","Swap Ins/Sec","Swap Outs/Sec","Buffers",
             "Cache","Memory Average % Used","Memory Size","Effective Memory % Used",
             "Context Switches","Interrupts","No. Of Processors" ]

        add_field => { "StandardDateTime" =>
          "%{input_YYYY}-%{input_MM}-%{input_DD}T%{input_hhmmss}" }
        add_field => { "Server" => "serverName" }
      }
    }
```

Aligns with input file

Reformatting time stamp

Watch this spot !

```
{
  "message": [
    "2014-02-13,G,00:00:00,14,0.00,0.07,0.00,0.00,104256.00,1436298.90,,0,,123.86,66.00,2.00"
  ],
  "@version": "1",
  "@timestamp": "2014-09-08T20:27:12.880Z",
  "host": "oc3122150850.ibm.com",
  "tags": [
    "SHOULD_OUTPUT",
    "DATA_LINE"
  ],
  "input_YYYY": "2014",
  "input_MM": "02",
  "input_DD": "13",
  "input_hhmmss": "00:00:00",
  "Date": "2014-02-13",
  "Status": "G",
  "Time": "00:00:00",
  "Obs": "14",
  "Processor Util": "0.00",
  "Run Queue": "0.07",
  "Swap Ins/Sec": "0.00",
  "Swap Outs/Sec": "0.00",
  "Buffers": "104256.00",
  "Cache": "1436298.90",
  "Memory Average % Used": null,
  "Memory Size": "0",
  "Effective Memory % Used": null,
  "Context Switches": "123.86",
  "Interrupts": "66.00",
  "No. Of Processors": "2.00",
  "StandardDateTime": "2014-02-13T00:00:00",
  "Server": "serverName"
```


Example 2: Adding field names to the items in the message

Within the IF-THEN logic of the example, you can now use the csv plug-in to associate the data in the message to the various fields that can be used for output. In this example, you even create a new field, StandardDateTime, which merges the fields that you created in the previous grok command.

Important: Note the addition of the **Server** field. It is associated to a value of serverName. In the following slides, you learn how to associate a host name to this message. Because the host name is not in the message itself, you must create a separate method to reference it.

IBM Training

Example 2: Output

```
output {

  if "SHOULD_OUTPUT" in [tags] {
    scacsv {
      fields => ["timestamp", "Server", "Processor Util",
                 "Run Queue","Swap Ins/Sec","Swap Outs/Sec","Buffers",
                 "Cache","Memory Average % Used","Memory Size","Effective Memory % Used",
                 "Context Switches","Interrupts" ]
      path   => "processorMemory.csv"
      group  => "processorMemory"
    }
  }
}
```

Subset of fields

```
processorMemory__2014-02-13T00:00:00__2014-05-02T23:45:00.csv

timestamp,Server,Processor Util,Run Queue,Swap Ins/Sec,Swap Outs/Sec,Buffers,Cache,Memory A
verage % Used,Memory Size,Effective Memory % Used,Context Switches,Interrupts
2014-02-13T00:00:00,serverName,0.00,0.07,0.00,0.00,104256.00,1436298.90,,0,,123.86,66.00
2014-02-13T00:15:00,serverName,0.07,0.33,0.00,0.00,113909.87,1430425.90,,0,,132.67,75.13
2014-02-13T00:30:00,serverName,0.20,0.33,0.00,0.00,123157.07,1412973.60,,0,,139.67,82.53
```

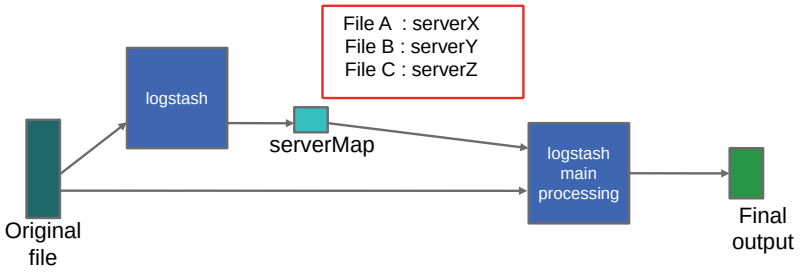
Still need to determine this

Example 2: Output

Finally, you can send only the data from the important messages. Notice the IF-THEN statement that relies on the tags that you added earlier. Using the key SHOULD_OUTPUT, you are going to send only data from the messages that earlier deemed important. Notice that each message in the output file has a host name of serverName. This generic name is a problem. You still need to resolve this name to the host name that was in the header file of the report.

Example 2: Determining server name from header information

- Events are usually ***independent***
  - Multiline events are an exception
  - Cannot easily carry information across events
- In this example, the server identity is in a separate event in the header
  - Other events do not know about the server identity



Example 2: Determining server name from header information

Determining the server name from the header was described earlier, but it is an important reminder of how logstash works. When logstash picks up a message, the message is often called an *event*. In almost all cases, these events are independent of all the other events that logstash sees. With a plug-in called *multiline*, you can append a message to the previous event based on some criteria, but does not store specific data. The report file that you have been working with had a host name in the header lines of the file. The line in the report file was treated as a separate event, and you cannot store it locally inside of logstash. To process this information and use it across events, you must create your own process.

Here is an example of how it could be done. The original report file has a file name that is associated to each of the messages that are extracted from it. With this data, you can create a separate logstash instance whose sole purpose is to read all the report files and extract just the host name from the one event where it is defined. Whenever it finds this single event, it sends the file name and host name to a hash file to save for later use. Then a second logstash instance normally processes the performance data. When it is time to associate the event's performance data to a host name, you take the file name associated to that event and lookup the host name from the hash file.

IBM Training

Example 2: Determining server name

```
Report Type:  Processor/Memory
Rules in Effect: Performance
Frequency:  15 Minute,Server: server1.SomeBank.com,Frequency: Fifteen Minute Report,
First Period: Feb 1 (Sat), 2014,Last Period: Latest day with data,
Enterprise,Customer,Sub Client ID1,Sub Client ID2,OS,Server
Austria,SomeBank,MLC509,Production,Linux,ausSomeBankserver1
Date,Status,Time,Obs,Processor Util,Run Queue,Swap Ins/Sec,Swap Outs/Sec,Buffers,Cache,
```

```
{
    "message" => "Frequency:  15 Minute,Server: server1.SomeBank.com,Frequency: Fifteen Minute Report,\r",
   "@version" => "1",
 "@timstamp" => 2014-12-02T21:23:03.640Z",
       "host" => "scapi.tivoli.edu"
       "path" => "/opt/scapi-training-data/logstash-examples/report-server1.com
}
```

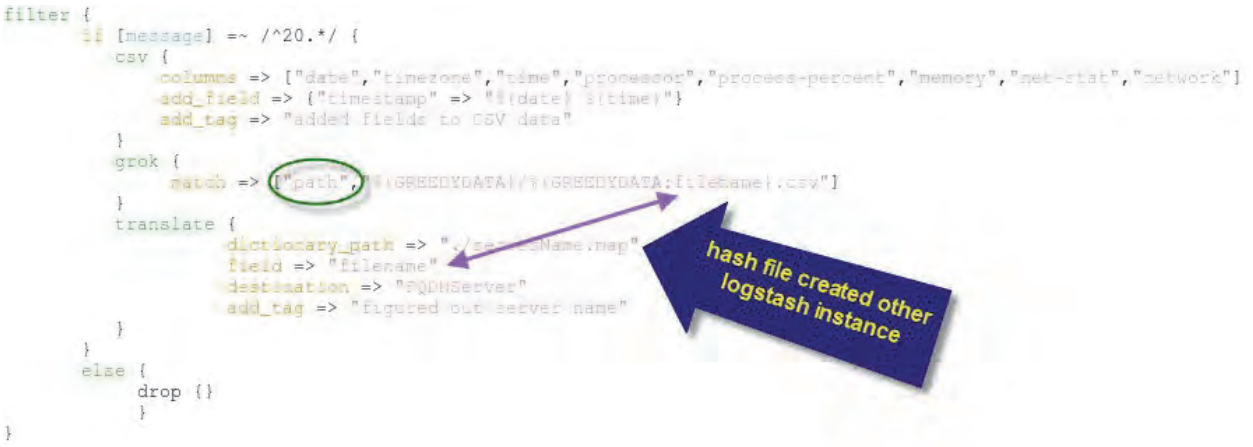
Advanced mediation techniques 34

Example 2: Determining server name

Here is how you could determine the server name with the sample data. Note the line (or event) in the header that denotes the server name. When logstash processes this line, the server automatically associates the path of the file that it was reading to the event. You now have a way of creating a file name-to-host name hash. When you see this specific event, you send these two items to a file in the output section of the conf file. You work with this situation in an upcoming exercise.

IBM Training

Example 2: Replacing server name with the translate plug-in



Example 2: Replacing server name with the translate plug-in

The second half of the challenge is to use the external hash file as a means to find the host name of a specific event. Because the path information is part of the event, you extract the file name where the event came from. Here you can use the translate plug-in to access the hash file or what is called a ***dictionary***. Give the translate plug-in the file name, and it returns the matching pair and associates it to a field name FQDNServer. You can now use that field when you publish data.

Extending logstash by building custom plug-ins

- Plug-ins are written primarily in Ruby
  Can call out to Java easily because logstash runs on jRuby
- Have the source for all the current plug-ins
  Is there something similar in the current list of inputs, filters, outputs
- Chapter 8 of *The Logstash Book, Extending Logstash*, has the details

Extending logstash by building custom plug-ins

It is possible to build custom plug-ins for logstash. Plug-ins are written in Ruby and can call to Java easily because logstash runs on jRuby. With all the source code of the current plug-ins available, it is possible to use them as starting points for your customizations. The *Logstash Book* has a chapter that explains creating a custom plug-in.

Lesson 2 Mediating customer databases

In this lesson, you are presented some thoughts on how to connect and mediate customer databases.

IBM Training

You now know what data you need

You need data with the following attributes:

- Time-based data that can be retrieved on a regular basis of every 5-15 minutes
- Data that can be organized by host and have columns, each one associated to a specific metric name
- Data that is numeric or can be converted to a numeric

You now know what data you need

At this point in your training, you should be aware of the data that you need and the required format to load Predictive Insights. It must be time-based data that you can retrieve on regular basis. The data must be organized with a time stamp, a host name, and a series of columns where each column is a metric name. The data must either be numeric or able to be converted to a number.

IBM Training

Thoughts on customer databases

- Customer has one or more tools that are collecting monitoring data
  - Have we mediated this data before?
  - How is the data being stored?
- Are the tools in-house custom or using a software package, open source or purchased?
  - Custom in-house tool
    - There is an expert somewhere
    - Probably has accessible back-end systems, for instance, open source database or flat files
    - May not be reliable
  - Software package
    - Might not know how things are stored
    - Proprietary interfaces may prevent direct connections
    - Might not support JDBC
- Who, What, When, Where, Why and How much…

Thoughts on customer databases

If you are working with a customer and want to use Predictive Insights, it is assumed that the customer has monitoring data to work with. You must know the nature of their monitoring systems to understand whether IBM has mediated that system before. If not, you must know where the data is being stored. Next, you must know if the data sources you need to connect to are in-house custom tools or a commercial or open source software package. Each of these options has pros and cons.

A custom in-house tool that gathers monitoring data normally has one or more experts who developed and manage the lifecycle of the tool. Access to this expert can make your implementation much easier. If the tool was developed in-house, it might have open source back-end systems where data is processed and stored. If you are lucky, you can access open-source expertise and tooling to help connect and extract necessary data. However, it has been shown that even the largest customers have unreliable custom in-house tools. A customer requested that Predictive Insights use a data source that ran on a user's notebook computer. When the notebook went home at night, the data flow from that tool stopped.

On the other end of the spectrum is a monitoring data source that is part of a software package. Here, the customer might know nothing about how the application works or how the data is stored. The customer might interact with the data by using proprietary interfaces that might be inaccessible from a JDBC perspective. Custom mediation development might be required in this situation.

When working with customer databases, you must rely on the five Ws and one H: who, what, where, when, why and how much.

Customer in-house tool

- Understand how the tool is used by customer
- Engage the developer or system administrator
  - How does it gather data
  - How does it store the data
  - How do you get the data out
- Can Predictive Insights access database for read-only purposes?
  - If it is needed, can you create a view in the database?

Customer in-house tool

If the customer has an in-house tool, you should be able to get access to the developer who is maintaining it. Meeting with users who drove the requirements for the tool is also useful. Important in this interaction is to engage these people and understand how the tool gathers and stores its data and what is their approved means of retrieving data. If it does have JDBC connection points, can Predictive Insights have read-only access to the data? If the data is in a database, can you create a view, if necessary, to produce the correct data format?

Determining access to performance data in software package

- Understand how the tool is used by customer
- Look for online support documentation
  For example, Zabbix is open-source monitoring system with really good documentation
- Engage the system administrators to learn what they understand
  How would you get data out of the system?
- Can Predictive Insights access database for read-only purposes?
  If it is needed, can you create a view in the database?

Determining access to performance data in software package

If the customer is using a commercial or open source software package, you must understand how the customer is using the tool. Putting a context around the data that is in the solution might help later in designing the mediation solution. Often times, there is useful support documentation online. This situation would be the case if the tool is an open source solution. Zabbix is one such solution and has good, available, online documentation. If online documentation is not useful, your next best option is to talk to the people who administer the solution. They have hopefully been trained on how the solution works and how it is designed. Possibly they understand how to get data out of the system. As with in-house solutions, determine whether there is a back-end database and if it is possible to create a view within it to support possible data formatting needs.

Example: Proof of concept with Zabbix

- Zabbix is an open source monitoring tool a financial customer was using
  - Pretty good online documentation
  - Uses MySQL as the back-end data base
  - Very little customer help
- Engineer investigated the approximately 100 tables in the database
  - Received a copy of the database
  - Reviewed the tables with JDBC client (SQuirreL) to look at data
  - Used MySQL workbench to understand the schema
- Discovered how the data was stored and which tables needed to be referenced to get all the necessary information
- Created a view and join to get data into an acceptable format
  - How would this view affect database performance?

Example: Proof of concept with Zabbix

Here is an actual field example that occurred for a customer. This customer was using Zabbix, which is a free open-source monitoring solution. Because it is open source, the documentation is fairly comprehensive to support ongoing development by others. The important aspect to this solution was the use of a MySQL database as the back-end storage location. However, although the customer did install the solution and was using it to collect performance data, they had little knowledge of how data was being captured within the database.

With the customer's permission, the engineer easily copied the database as a set of SQL files. With a copy of the database at hand, he began investigating the data by using a simple SQL client to understand what data was captured within what table. Using MySQL workbench would also have been useful to graphically see the schema of all the 100 tables that were in the database.

After reviewing the tables, the engineer found the three tables that were necessary to extract the necessary performance data with the time stamps and host names associated with it. Using this information, he created an appropriate view in the database. Because of the way the data was stored in database schema, this view might have serious impacts on a production system. It would be important when moving from the proof-of-concept stage to production to understand if this view was indeed a usable solution.

IBM Training

Student exercises

Student exercises

IBM Training

Unit summary

- Describe the third-party tool logstash
- Build logstash conf files for mediating data
- Describe mediation considerations

Unit summary

IBM Training