IBM Journey to Cloud and AI

Analytics Modernization Workshop

Workbook

IBM

PoT.0238.01

Author:
Vikram S Khatri
Executive I/T Specialist
vikram.khatri@us.ibm.com

Dated:10-20-2018: Last Updated: 11-17-2018

Acknowledgements

- **Duane Almeter** and **Craig Maddux** for their exemplary leadership to bring best of the technical experts together to help create this multidiscipline asset and bring multiple technical sellers for them to deliver this workshop in their respective markets. Without your leadership, this would not have been possible – Thank you.
- **Anjali Shah**, **Rui Fan**, **Daniel Kikuchi** and **Ben Chard** for bringing their data science expertise to this asset and creating the AI-based predictive modeling for this asset.
- **John Van Buren** for his expertise in Catalog, Governance and Integration to enrich this asset.
- **Kent Rubin** for his expertise in business dashboards.
- **Jeff Tuck** for writing Predictive Modeling microservice to enrich the stock trader application written by **John Alcorn** and **Greg Hintermeister** under the guidance of **Eric Herness**, IBM Fellow and **Bratt Coffmann** for his UI assistance.
- **Sriram Srinivasan**, Chief Architect of the IBM Cloud Private for Data and his team – **Ashwin Dev**, **Prashant Patel**, **Sahil Shah**, **Parth Komperla** from the IBM development labs to provide technical details and help.
- **Dean Compher**, **Dominic Farrar**, **Gary Brunell**, **Karen Groski**, **Kyle Talish**, **Neal Finkelstein**, **Patrick Pitre**, **Paul Betts**, **Ryan Kather**, **S Suh**, **Dale Mumper** and **David Solomon** for active participation in testing the asset and for taking the role of proctor to facilitate delivery of this workshop.
- **Claus Huempel** from IBM Germany to learn and deliver this asset in Europe.
- **Eric Watson** for helping to refine the content.
- **Kristen Mucci** for planning and organizing the events throughout North America.

Thank you - **Beth Friday**, VP, North America Technical sales for being our Technical Executive Sponsor and **Hemanth Manda**, Director, Platform Offerings and **Sampada Basarkar**, Director of Development for providing help from the development teams.

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Lab 01 Setup and Introduction

1.1 IBM Journey to Cloud and AI: Analytics Modernization Workshop

This session provides hands-on experience with modernizing your microservices applications by enriching them with machine learning and artificial intelligence. The Journey to Artificial Intelligence requires a strong information architecture that supports self-service capabilities and balances the needs of both the agility required by lines of business and the 'Enterprise class' delivery required by IT. This journey can move significantly faster if you use an integrated platform to: **Collect**, **Organize** and **Analyze** your data. Let's start this journey by defining some key concepts.

1.2 What is IBM Cloud Private?

IBM Cloud Private is a cloud-computing model run solely for one organization. It can be managed internally or by a third party. It can be hosted on premises behind your company's firewall or externally on a third-party provider's cloud. Private cloud offers the benefits of a public cloud including: rapid deployment and scalability, ease of use, elasticity. It also provides: greater control, increased performance, predictable costs, tighter security and flexible management options. It can be customized to meet your unique needs and security requirements.

1.3 What is IBM Cloud Private for Data?

IBM Cloud Private for Data (ICP-D), built on the foundation of IBM Cloud Private, is an integrated end-to-end platform designed to help make data more accessible and trusted. It provides access to many analytical tools to help you gain insights from your data.

ICP-D provides the data platform which enables you to climb the ladder to AI faster. With it, you can quickly build, train and deploy machine learning (ML) and artificial intelligence (AI) models. You can also: 'Inventory and catalog' your data sources, provide 'self-service' shopping for data, and provide data integration and refinement capabilities. High quality, trusted data can be more easily prepared and assembled though this integrated platform.

1.4 Audience for this IBM Workshop

This IBM workshop is mainly designed for line-of-business professionals who are tasked to gain new insights from all available data – regardless of its type and origin. The following personas will greatly benefit from this workshop.

Role	Capabilities
Business analyst	Business analyst delivers value by taking data, using it to answer questions, and communicating the results to help make better business decisions.
Data scientist	Data scientist brings expertise in statistics and the process of building ML/AI models to make predictions and answer key business questions.

Data steward	Data steward brings integration and governance to the data.
Data engineer	Data engineer builds and optimizes the systems to allow data scientists and analysts to perform their work. The data engineer ensures that any data is properly received, transformed, stored, and made accessible to other users.

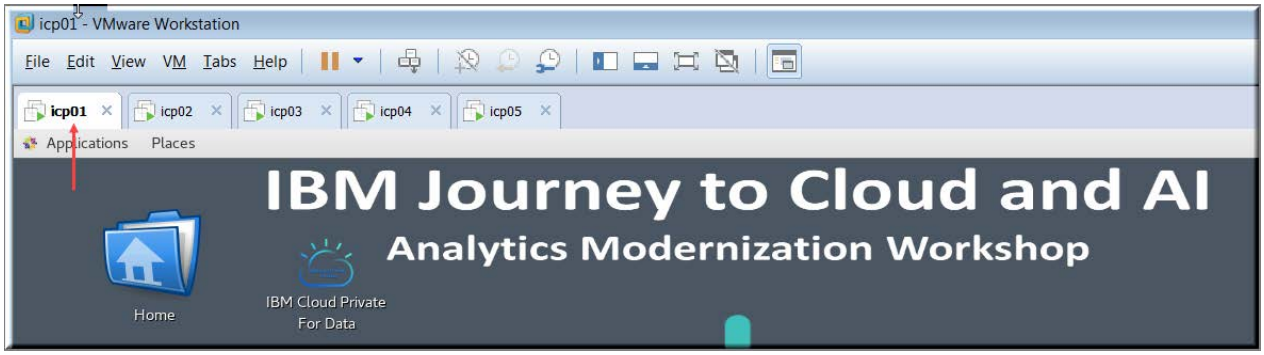
1.5 Lab environment

We are using eleven Virtual Machines (VMs) for this workshop. This software environment was built to the following specifications:

- IBM Cloud Private 3.1 as foundational technology
- IBM Cloud Private for Data 1.1.0.2 as an Integrated Data Platform

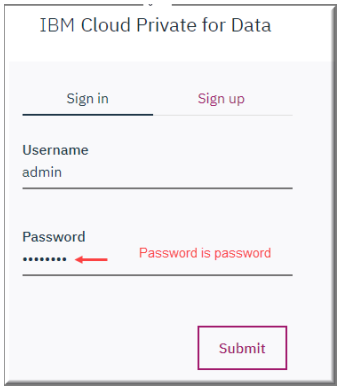
1.6 Let's get started!

__1. On your laptop/workstation/Cloud VM, multiple virtual machines are running. If you are not already in `icp01` VM, click the `icp01` tab to switch to the first virtual machine.

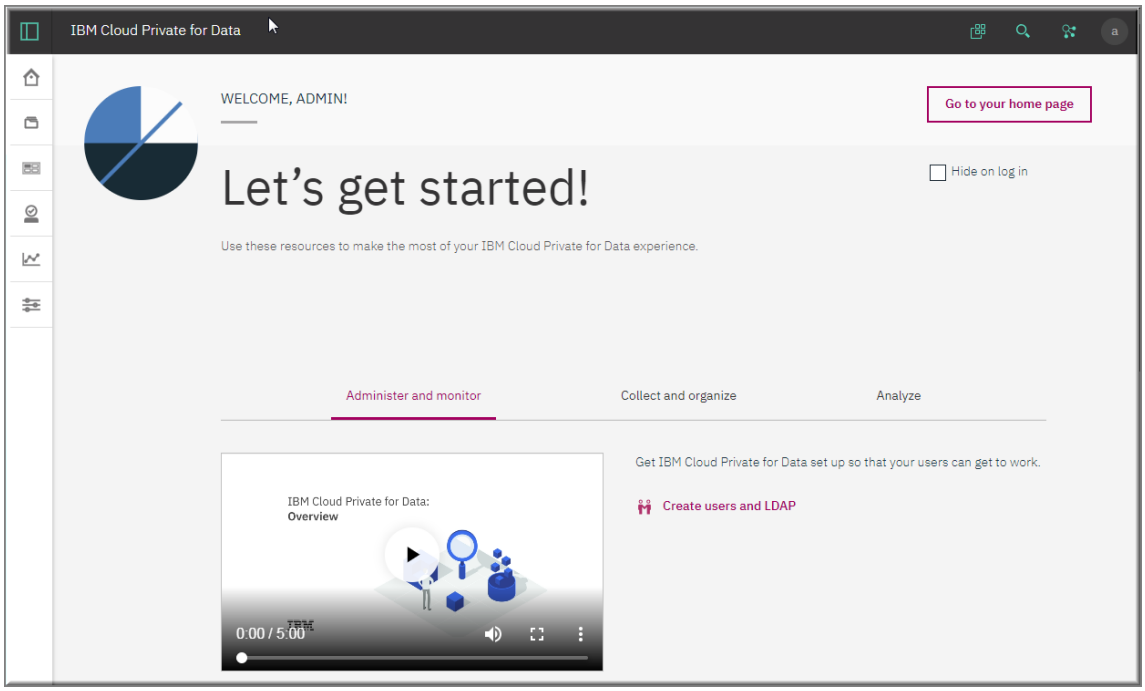


__2. Double-click `IBM Cloud Private for Data` from the desktop to open the web browser to start the IBM Cloud Private for Data web console. Maximize your browser window.

__3. The web UI console displays as shown. Please type `admin` and `password` for the *Username* and *Password* and click `Submit`.



__4. Review the `Let's get started!`



__5. Even if you only spend a few minutes on the `Let's get started!` page, be sure look at the following:

__a. One menu that provides capabilities to **Collect**, **Organize** and **Analyze** data using a single end-to-end platform.

__b. **Collect** – You can collect data from a myriad of sources, no matter where it resides. For example, you can collect data from IBM Db2®, Oracle, Teradata, Hadoop, flat files and so on. It does not matter if data is stored in a cloud or on-premises.

__c. **Organize** – Use embedded machine learning to automate the process of assigning business terms to newly discovered data sources. Enforce governance policies as you search, find and catalog your data. Create an enterprise-level data catalog through which you can search for the data you need. Collect and prepare the data for analysis that you need to perform. Use embedded security to provide visibility to all data and still control who accesses it.

__d. **Analyze** – Build predictive and prescriptive models using open source programming languages and/or simpler graphical user interfaces. With the click of a button, deploy these models into production and publish an Open REST API that can be consumed by any application which can be running either on IBM Cloud Private or any other platform. Build dashboards and interact with your data to gain insight and business understanding.

__e. The key personas' roles are divided into different categories. If you are a data engineer / data steward / data analyst, you will focus areas on the **Collect** and **Organize** category.

__f. The data scientists and business analysts will use **Analyze** category. All personas will collaborate using an integrated single platform.

__g. The **Administer** category provides tools to organize teams, define roles and provide access and control.

__6. IBM Cloud Private for Data is most useful in these four scenarios.

__a. **Manage All Your Data**: Use Discovery to automate the process of assigning business terms to technical assets. Use enterprise catalog to secure, govern and control access to your data regardless where it resides. Use a combination of virtualization and transformation to prepare your data for analysis.

__b. **Build Your Ladder to AI**: Build a strong information architecture to help you realize the value of leveraging ML/AI.

__c. **Modernize Your Data and Analytics Workloads**: Modernize your data platform and provide data scientists and application developers the ability to quickly add AI to your applications. ICP is the foundation of ICP-D. Use ICP to build cloud-native, microservice-based applications and use ICP-D to enrich them with machine learning and artificial intelligence.

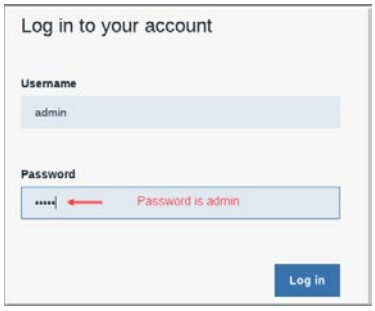
__d. **Compliance Readiness** – Our strong governance capabilities allow you to provide regulatory compliance.

```
We will be demonstrating all of the above capabilities in this workshop
from Collect, Organize and Analyze to demonstrate AI-driven model
development and consumption of those APIs into a microservice
application.
```

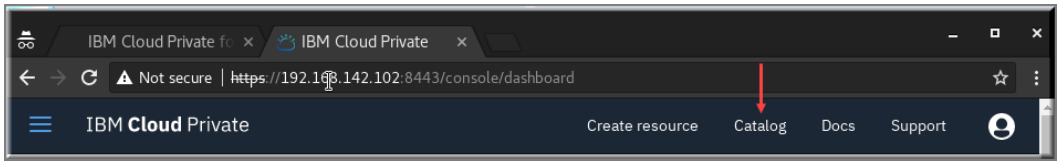
1.7 Explore IBM Cloud Private

__7. ICP-D foundation is IBM Cloud Private. On your `icp01` VM's desktop, double-click **IBM Cloud Private Web UI.**

__8. Log in using `admin` and `admin`.



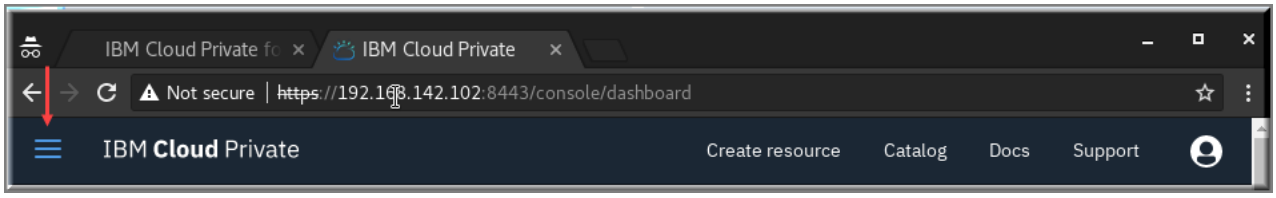
__9. Click `Catalog`.



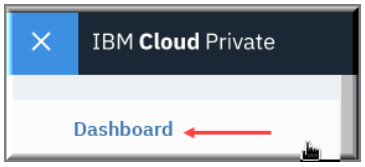
__10. This is where you can find IBM and Open Source software that you can install in the IBM Cloud Private platform. These are IBM-vetted software products that can be installed in the IBM Cloud Private platform.

```
IBM is bringing power of cloud platform within the bounds of your firewall.
```

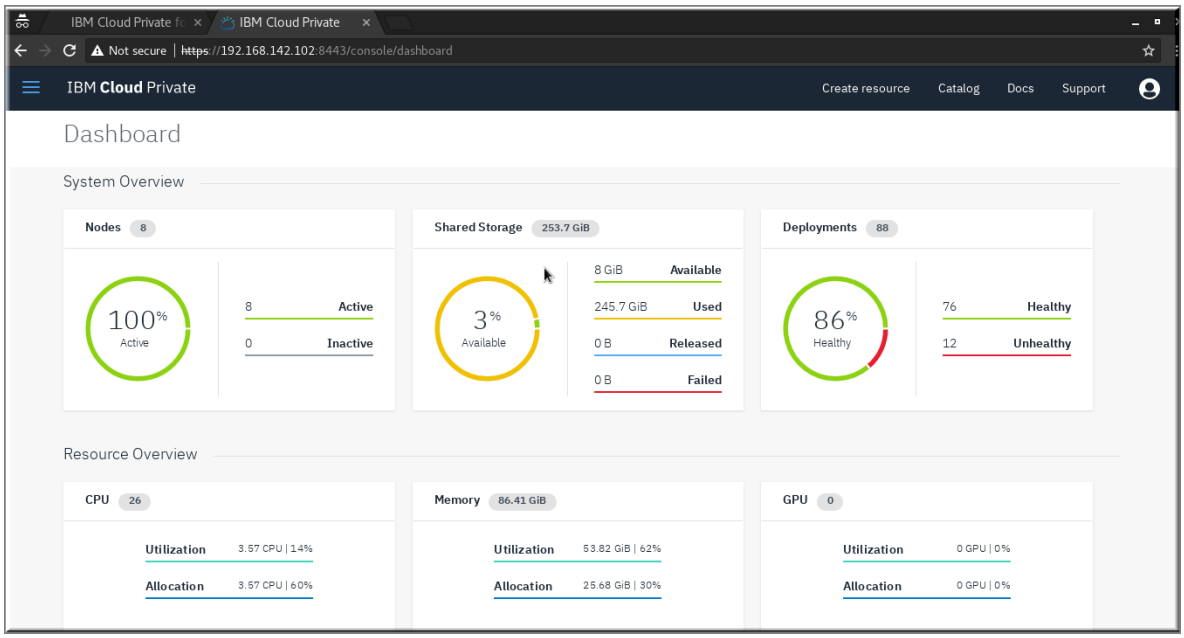
__11. Click the three horizontal bar symbol (or hamburger).



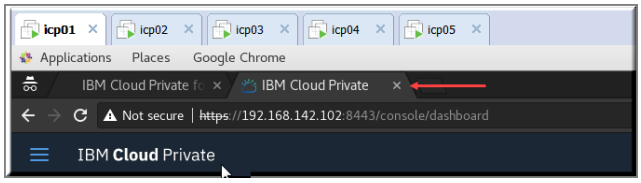
__12. Click `Dashboard`.



__13. The `System Overview` dashboard gives you a picture of the overall health of the IBM Cloud Private cluster through single pane of glass.



__14. Close the `IBM Cloud Private` tab.

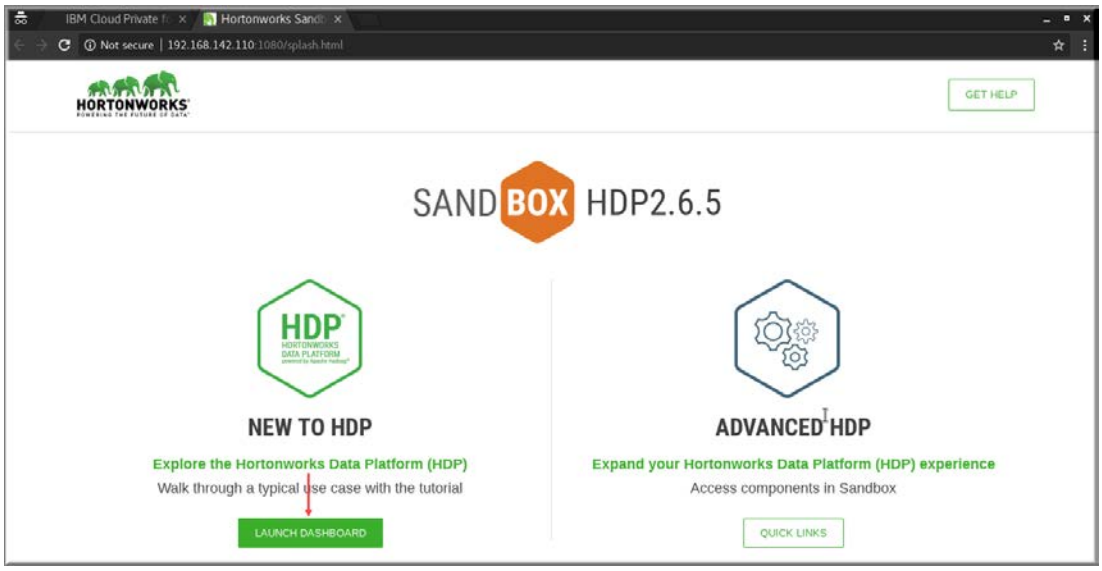


1.8 Explore Hadoop Platform

__15. In our journey of the integrated data platform, we will show `Collect` – which is about collection of data regardless where it resides.

__16. On your `icp01` VM's desktop, double-click `Hadoop Web Console` to launch Hortonworks Hadoop Data Platform.

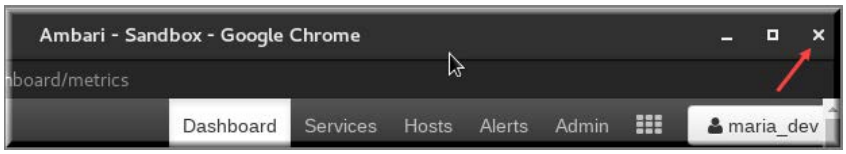
__17. Click `Launch Dashboard`



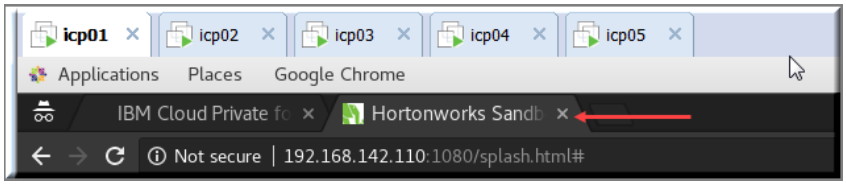
__18. Type in Username: `admin` and Password: `password`. Click `Sign in`.

__19. The Apache Ambari dashboard opens in a new window. We will return to this in later lab exercises.

__20. Click the `Ambari` browser window.



__21. Close the `Hortonworks Sandbox` tab.



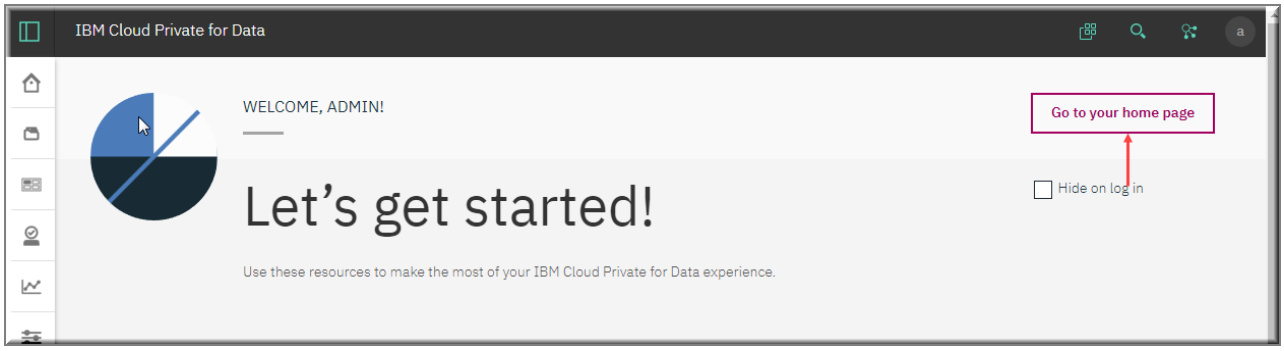
__22. ICP-D allows integration of Hadoop edge node (or gateway) to transfer data to and from the Hadoop cluster. Livy integration also provides the ability to execute jobs remotely on the Hadoop cluster. This can be done using `Administer` ⇨ `Hadoop Integration`. Note: We will use the Hadoop integration later lab exercises.

	**Note:** By design, Hadoop is not running (nor is it intended to run) on the ICP-D cluster. Many companies use Hadoop as an enterprise data repository. Our approach with ICP-D is to leave the data in Hadoop, and access it as needed for specific projects.

__23. Switch back to the ICP-D web console.

1.9 Explore Home Page

__24. Click `Go to your home page`.



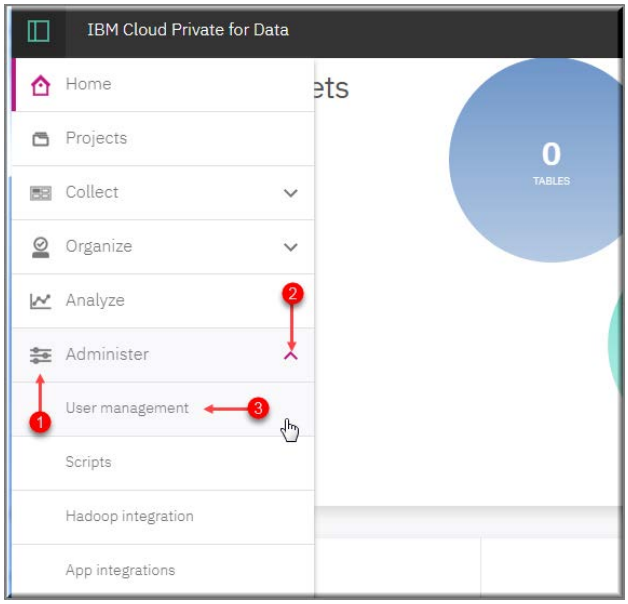
__25. The home page shows the inventory of the integrated data platform for Governance, Terms, Policies, Assets, Governed Assets, Asset Distribution and Transformation jobs. The intent of this page is to provide individual customizations on daily tasks for different personas.

__26. Scroll through the page to see inventory and asset types. Since we are starting fresh, we will visit the home page later to see the assets and various lists.

1.10 User Management: Persona-based Roles and Teams

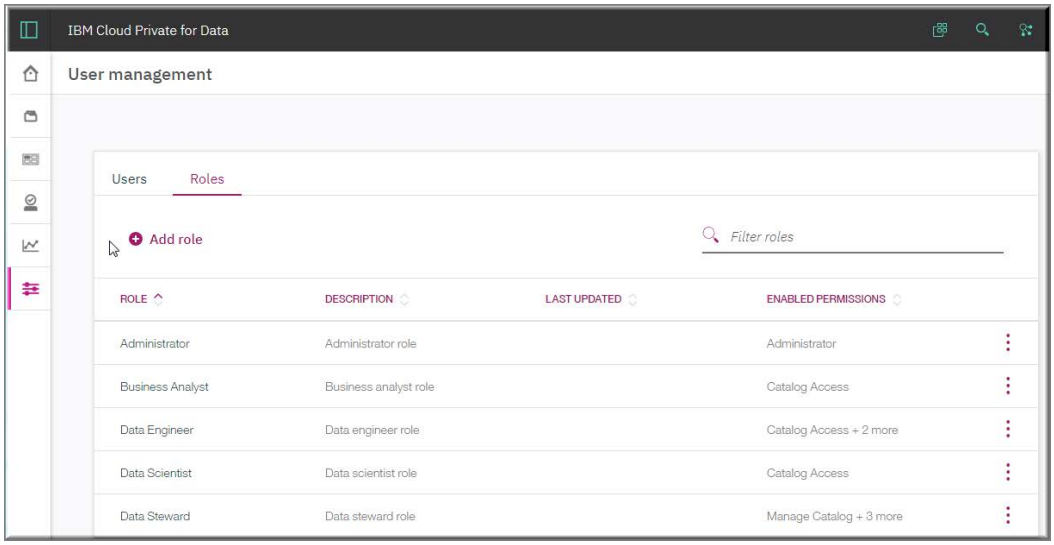
__27. This section explores data governance and user authorizations for the various stages of the data analytics pipeline.

__28. Hover your mouse on the navigation bar on the left side of the page to pull down the menu and click the down arrow on `Administer` to bring the drop-down menu and click `User management`.



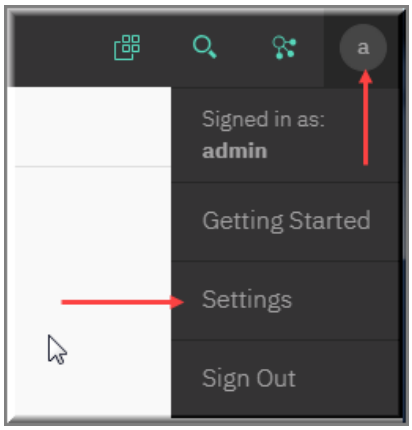
__29. ICP-D can interact with your LDAP for the user management.

__30. Click `Roles` and notice Business Analyst, Data Engineer, Data Scientist, and Data Steward roles. These roles are customizable for different types of permissions such as Catalog, Policy, Provision Database, Manage Catalog, Manage Governance Policies and Virtualize and Transform data.

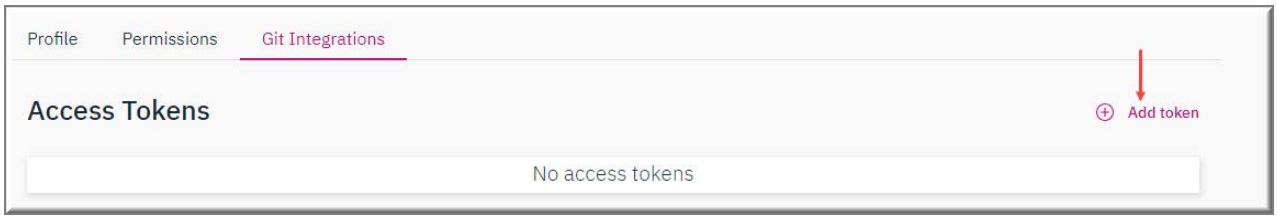


1.11 Integration with Source Control

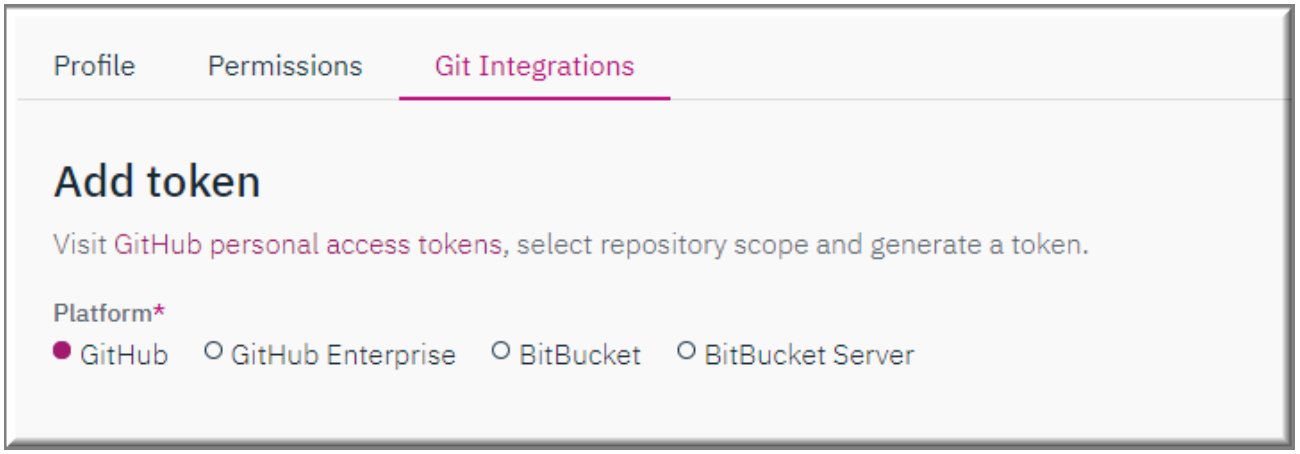
__31. Click a. This is located towards the right on the menu bar and click `Settings`.



__32. Click `Git Integrations` and click `Add Token`.



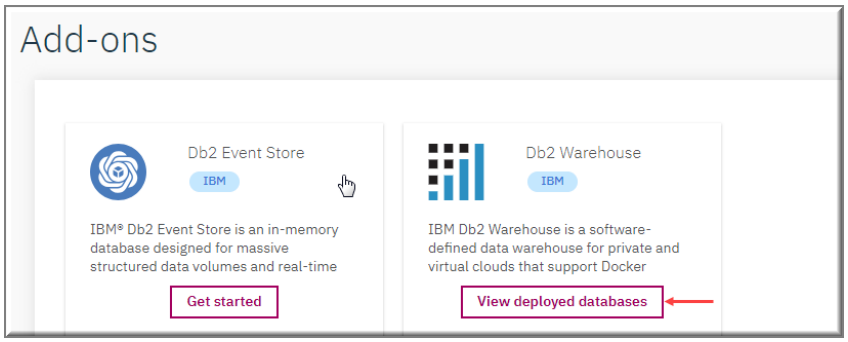
__33. Notice that you can manage repository with `GitHub`, `GitHub Enterprise`, `BitBuket` and `BitBucket Server`.



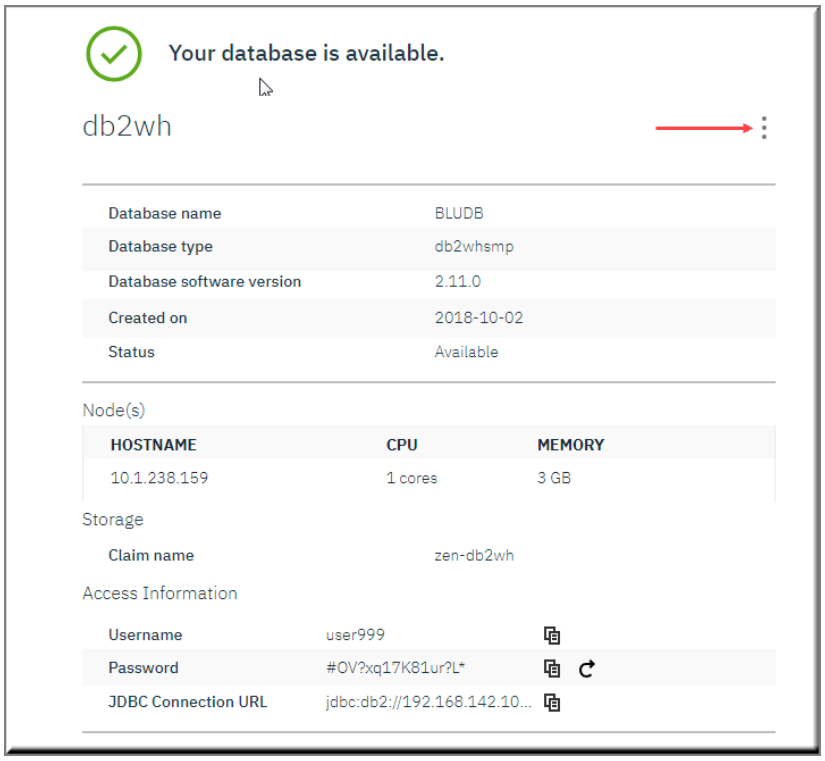
__34. This will allow you to integrate ICP-D projects with your current CICD (Continuous Integration and Continuous Delivery) pipe line to automate delivery of the artifacts. You can use capabilities from IBM Cloud Platform to build cloud native microservices applications which are tied to the ML/AI model development and delivery pipeline.

__35. Click `Add-on` icon.

__36. Notice that `Db2 Event Store` and `Db2 Warehouse` applications available and ready to be installed in the IBM Cloud Private for Data platform with click of a button. You will see more IBM and IBM Business Partners applications in the Add-on as time progresses. Click `View Deployed Databases`.



__37. In this local ICP-D cluster – we have already installed Db2 Warehouse. Click the three vertical dots (AKA `More Vert`) and click `Details`.



	Note: By design DB2 Warehouse is running on the ICP-D cluster. The intent of this is to provide 'sandbox' capability for projects on the platform. Clients can also choose to use this capability to run production data marts and data warehouses.

__38. Note that applications and databases available through `Add-on` are native to the ICP-D. IBM collaborates and partners with `MongoDB`, `portworx`, `Datameer`, `Lightbend`, `Senzing`, `Prolifics` and others to build an ecosystem around ICP-D. The details can be found at `https://www.ibm.com/products/cloud-private-for-data/partners`

1.12 Why make IBM Cloud Private for Data your platform for data

__39. IBM understands data and provides an integrated, end-to-end data platform that enables enterprises to:

- Collect relevant data and make it simpler and more accessible
- Use federation, virtualization and/or transformation to combine and refine data sets
- Organize data so it can be trusted
- Analyze insights on demand
- Embed machine learning everywhere

__40. Let's begin this journey with the following lab exercises to demonstrate all of the above.

**** End of Lab 01 - Setup and Introduction.**

Lab 02 Executive Demo

Before we proceed with the lab exercises, let's demonstrate some of the key functions of ICP-D and the microservices-based application before and after implementation of the machine learning models.

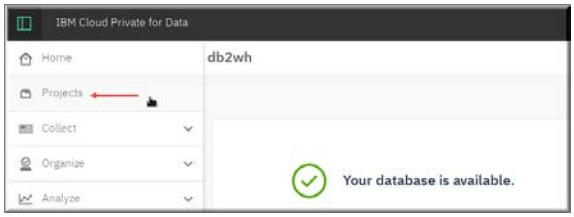
2.1 Import Project

__1. ICP-D is a platform that allows a team to work collaboratively using tools and a collection of related assets in a project with access and view control.

__2. We will work with a project that we have created for the purpose of this workshop. You will create your own projects and explore features in the following lab exercises.

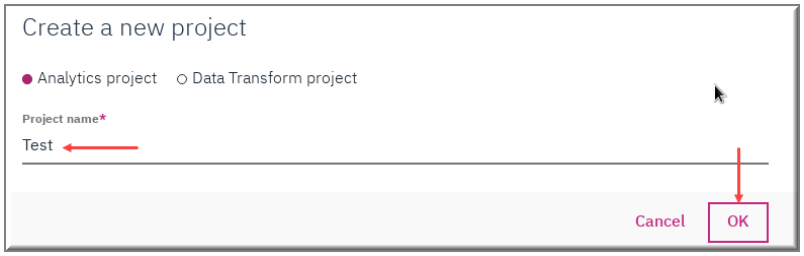
__3. Switch to the ICP-D web console.

__4. Hover your mouse on the left navigation pane and it will pull the menu towards right. Click `Projects`.



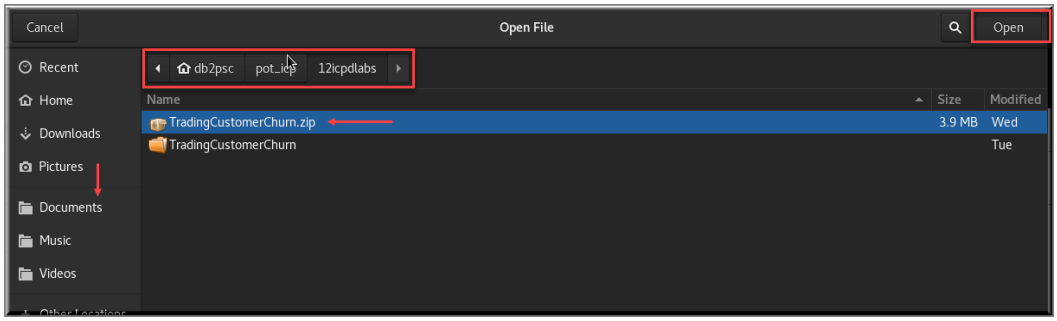
__5. Click `New Project`.

__6. Give the project the name `Test` and click `OK`.



__7. Click the middle tab `From File` and then click `browse`.

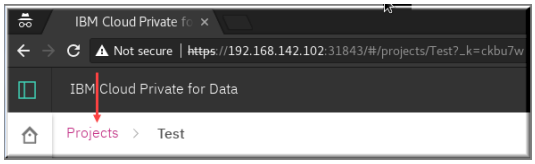
__8. On the left pane, double-click `Documents`. On the right pane, double-click `pot_icp`, double-click `12icpdlabs` and click `TradingCustomerChurn.zip`.



__9. Click `Create` (towards the bottom right).

__10. The project creation for the first time is an involved process that launches many resources in the ICP-D cluster. It may take a minute or two.

__11. Click **Projects** (top left-hand corner).



__12. Notice that we imported Project `TradingCustomerChurn` from a zip file.

__13. Click `TradingCustomerChurn.`

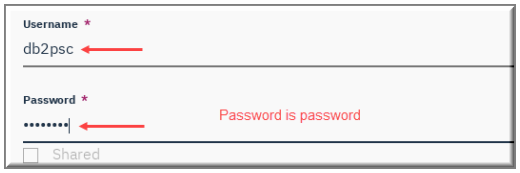
__14. It displays the number of assets, environments, data sources, jobs and collaborator.

2.2 Connect to a Data Source

__15. Click `Data Sources` and then click `Db2Warehouse`.

	Note: In our environment, we added Db2 Warehouse as an add-on feature – indicating that it has tight integration with ICP-D. This will be the case for any add-on that you choose to deploy. [Remember: the add-on feature icon is available on the top menu bar.]

__16. Scroll down and specify *Username:* `db2psc` and *Password*: `password` and click `Save` [Located in bottom right corner.]



__17. Note: There was no test connection facility. Let's see how this can be done easily.

__18. Click `TradingCustomerChurn` [middle breadcrumb in the top menu bar.]

__19. Click `Assets`. [Note: Your number may be different.]

__20. Click ⊕ `add notebook`.

__21. Enter Name: `Test` and select the *Environment:* `Jupyter with Python 3.5, Scala 2.11, R 3.4.3, Spark 2.2.1.`

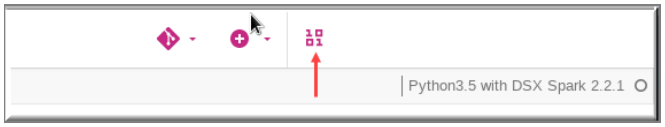
__22. Click `Create` [bottom right corner.]

	**Note:** What just happened?

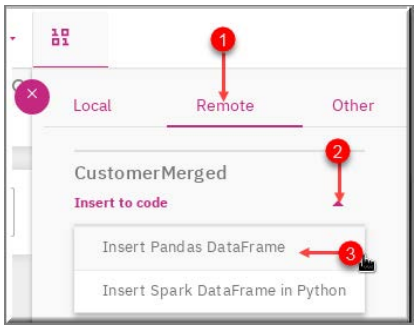
__23. Wait for the Jupyter notebook to launch and notice what happens.

- A pod was instantiated – which means loading a complete compute Jupyter notebook environment (7+ GB) with all the artifacts from the private ICP-D registry.
- IBM Cloud Private schedules this pod on any VM – wherever CPU and memory resources are available.
- IP addresses and connections are all configured automatically.
- The same working environment can be used by multiple users. If a single pod's resources are not sufficient, another environment is created automatically.
- When the number of users grow, you can add more machines to the ICP-D cluster and scheduling of resources is handled automatically.
- ICP-D's scale-out model is pretty effective.
- You no longer have to wait days or even weeks to get the compute resources.
- IBM Cloud Private makes 40 – 60 percent more efficient use of compute resources. This means more users can be accommodated with same compute capacity. As one task completes, its resources are freed up to work on next one.

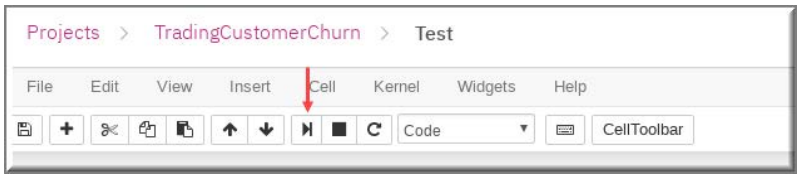
__24. Click the **Binary Number** icon on the menu bar (last icon).



__25. Click `Remote` [Middle]. Click `down arrow` for the first item and click `Insert Pandas Data Frame`.



__26. Let's run this Jupyter Notebook cell. [Remember `Shift-Enter` is the keyboard shortcut to run a selected cell.]



__27. During the execution of a cell in a Jupyter notebook, an asterisk (*) displays in the square bracket which changes to a number when execution for that cell completes.

__28. If the connection is successful, you get the top few rows of the DB2 warehouse table. Skip the next step – which is only required if your cell did not execute.

__29. If your cell did not execute successfully, don't worry. Minimize the browser window and double-click `Create Workshop User in Db2 Warehouse` to create a temporary Db2 Warehouse user.

__30. Switch back to the browser window and press `Shift-Enter` to execute the cell again. You should now see the first few rows from the Db2 Warehouse table as shown below.

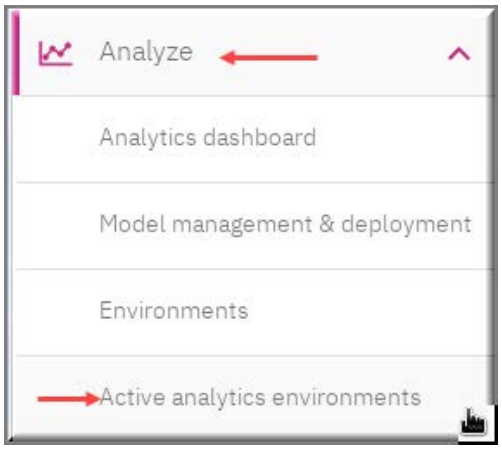
```
In [3]:
import dsx_core_utils, requests, jaydebeapi, os, io, sys
from pyspark.sql import SparkSession
import pandas as pd
df1 = None
dataSet = dsx_core_utils.get_remote_data_set_info('CustomerMerged')
dataSource = dsx_core_utils.get_data_source_info(dataSet['datasource'])
if (sys.version_info >= (3, 0)):
    conn = jaydebeapi.connect(dataSource['driver_class'], dataSource['URL'], [dataSource['user'], dataSource['password']])
else:
    conn = jaydebeapi.connect(dataSource['driver_class'], [dataSource['URL'], dataSource['user'], dataSource['password']])
query = 'select * from ' + (dataSet['schema'] + '.' if (len(dataSet['schema'].strip()) != 0) else '') + dataSet['table']
df1 = pd.read_sql(query, con=conn)
df1.head()
```

Out[3]:

	ID	CHURNRISK	GENDER	STATUS	CHILDREN	ESTINCOME	HOMEOWNER	AGE	TOTALDOLLARVALUETRADED	TOTALUNITSTRADED	LARGESTSINGLETRANSA
0	0	Low	F	S	1	38000.00	N	24	59755.98	206	29877.990
1	1	Low	M	M	2	29616.00	N	49	29782.98	45	14891.490
2	2	Low	M	M	0	19732.80	N	51	24812.48	22	12406.240
3	3	High	M	S	2	96.33	N	56	26132.61	32	13066.305
4	4	High	F	M	2	52004.80	N	25	5030.50	23	1257.625

2.3 Check Running Environment

__31. Hover the mouse on the left menu bar and click `Analyze ⇨ Active analytics environments`.



__32. Notice the Jupyter runtime environment and check the status. It should be green.

	Note: Make sure that you have only one environment active at any given time in this resource-constrained environment. If more than one environment is running, the system may become sluggish.

2.4 Stock Trader Opening Bell Analysis dashboard

__33. Boatswain Trading is aware that revenues are declining. Let's discover the actual trend and what the data presents. In our scenario, the business analyst has requested aggregated stock transaction data. It contains data such as the historical total number of individual customer visits to our website and number of trades per customer for the past year.

__34. Let's take a look at the dashboard first. We will build same dashboard in a later exercise to learn to use the ICP-D integrated data platform to gain an insight into our business.

__35. Hover your mouse on the left menu bar and click `Projects`. Click `TradingCustomerChurn` project.

__36. Click `Assets` and then click `Dashboards`.

__37. We have prebuilt three dashboards. [Note: You will get a chance to build these dashboards in later exercises.]

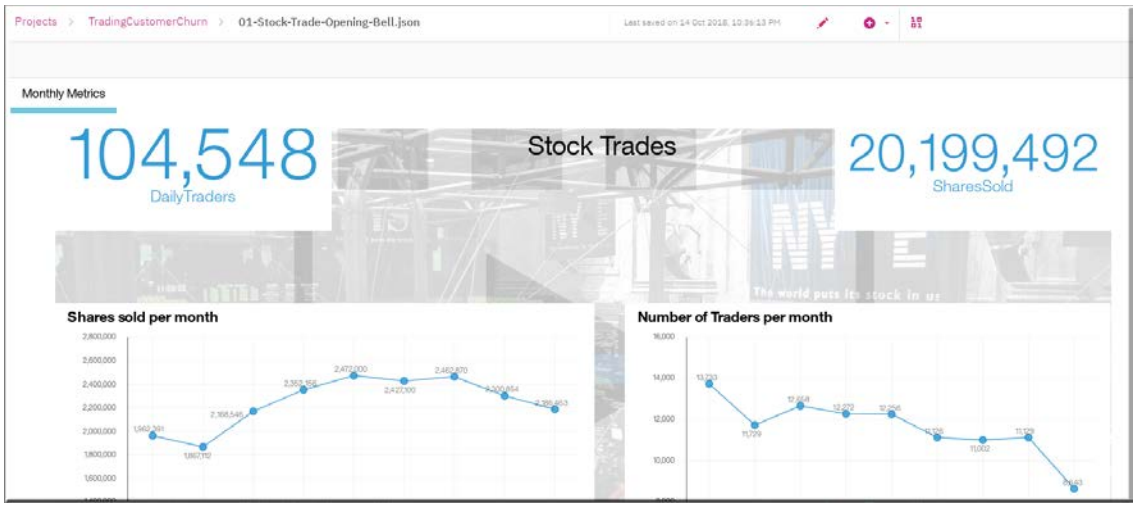
__38. The first dashboard shows a visualization of the 'Stock's Before' analysis.

__39. The second dashboard shows the results of the analysis of the customer demographics data that was collected from various sources, such as Data Warehouse and Hadoop environment, after curation of data. This information is used to evaluate which measures (AKA factors or variables) have the greatest impact on the customer attrition.

__40. The third dashboard shows the analysis after implementation of the AI models and the impact that it had on results of the business.

__41. Click `01-Stock-Trader-Opening-Bell.json`.

__42. The dashboard shows the consolidated report of our business. You can see that the number of shares sold per month is relatively flat and daily trades are declining.



__43. The data for this report is stored at a central place and is governed and managed through the team collaboration between business analysts, data engineer, data scientist and data steward.

2.5 Stock Trader Analysis Closing Bell Analysis

The purpose of our lab exercises is to show the journey of analytics modernization from our opening dashboard to closing bell dashboard.

__44. Let's look at the finished product, and then dive into the details.

__45. Go back to `Project` ⇨ `TradingCustomerChurn` ⇨ `Assets` ⇨`Dashboard` and click `03-Stock-Trader-Closing-Bell.json`

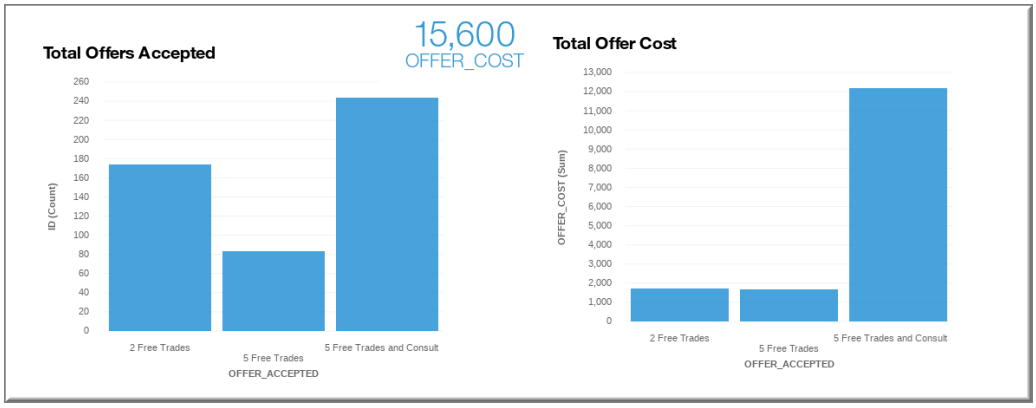
__46. The dashboard is built from the data sets present in the project.

__47. Notice the trends of the Shares Sold per Month and Number of Traders per Month and compare with the Opening Bell dashboard.

__48. The graph should display as the following figure:

__49. After implementation of the predictive models built and scored using different machine learning algorithms, changes to the modern microservice application were made to offer personalized promotions based on customer's history and risk of separation. The results after a few months show an increase in the number of shares sold per month and number of stock traders per month.

__50. Scroll down the dashboard and to see the total number of retention offers accepted and the total cost of those offers. Given the increased number of 'shares sold' and 'traders per month' from the prior graph, the USD 15,600 cost (that is, lost revenue from free trades and wealth consulting) seems like a very good investment.

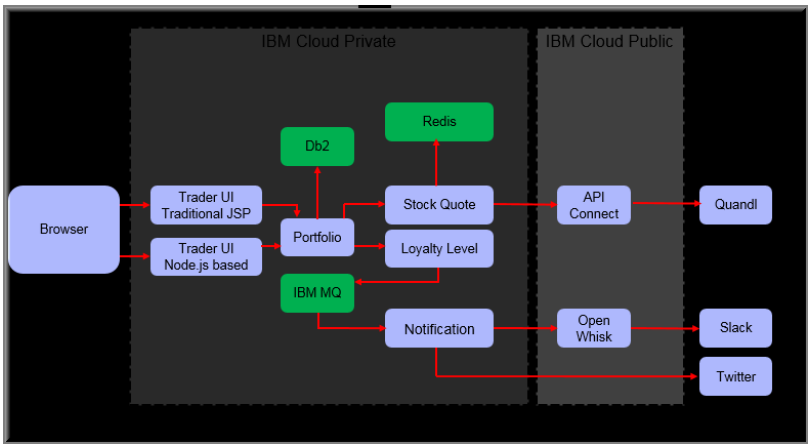


__51. This dashboard uses data for a full year from a .csv file and the second data set uses the customer offers and costs to business from a Db2 Warehouse table.

2.6 Run Stock-Trader Before Application

__52. The software development and distribution methods and deployment have changed rapidly since 2015. The application that we show is built on a microservices-based architecture that allows loosely coupled services to work together. This allows components to upgraded incrementally, as opposed to traditional, monolithic application development.

__53. The sample stock trader application architecture is displayed below showing interaction between various microservices.



__54. The infrastructure services Db2, MQ and Redis are shown in green boxes whereas microservices are shown in purple.

__55. The main advantage of microservice-based application is the ability to modify, deploy, update and scale services dynamically and individually.

__56. In a modern microservice mesh architecture, it is possible to run multiple versions of the same service in production and control the traffic dynamically based on rules that can be defined outside of the service itself, eliminating the need to need to modify any of the microservice code.

	Note: Start talking with your developers about the concept of a Service Mesh and implement it on IBM Cloud Private platform. Consult us if you would like more information.

__57. The application shown here is the demo application that does not address all features and functions of a commercial application but gives you insight into the concept of application modernization. The IBM Cloud Private platform provides tools to help you migrate your monolithic applications to a microservices-based architecture, free of cost[ML1].

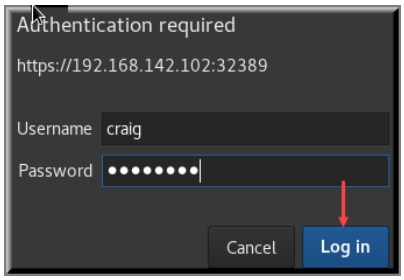
__58. If your journey on this path has not yet begun, share this application with your development teams. It will provide you with the framework you need to start building your own microservice-based applications. You can find it at: `https://github.com/IBMStockTrader`

__59. Let's launch this application.

__60. Minimize your browser window.

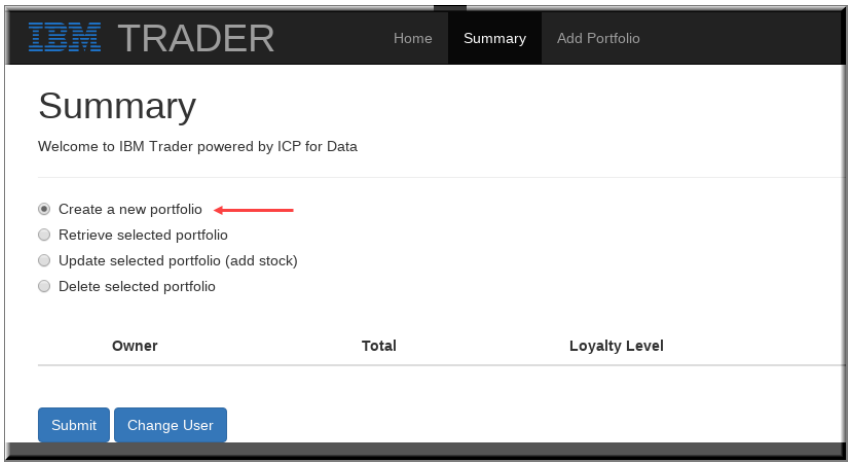
__61. Double-click `Stock Before.`

__62. A login popup text box displays – Enter Username: `craig` and Password: `password`.



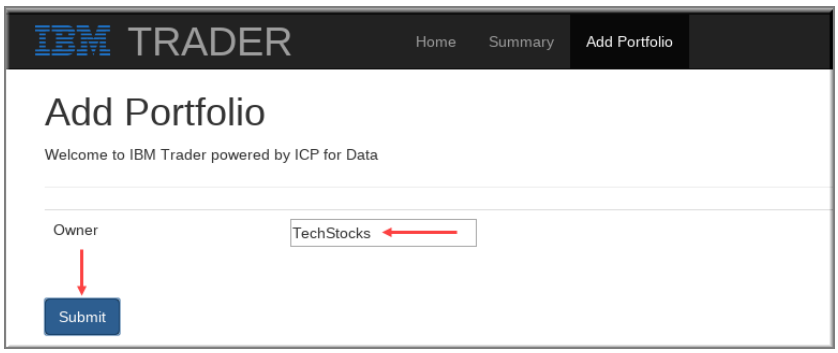
__63. NOTE: The first time the application is launched it may take a little longer to start due to limited system resources of the workshop environment.

__64. Select `Create a new portfolio` and click `Submit`.

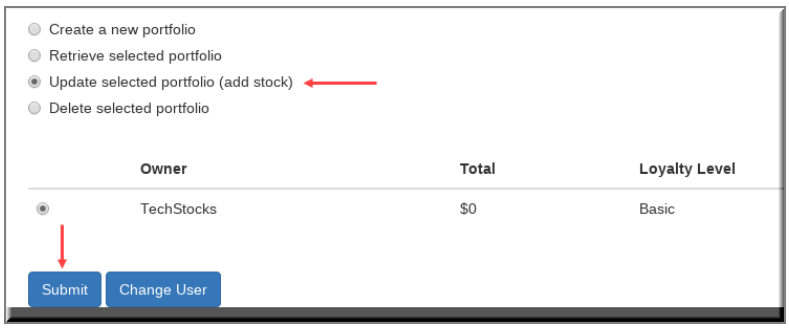


	Note: One advantage of a microservices-based application is that it allows polyglot services to be written in the language of choice by the developers. This allows services and applications to work together easily. Developers are no longer forced to change their preferred language or learn a new language due to monolithic legacy applications.

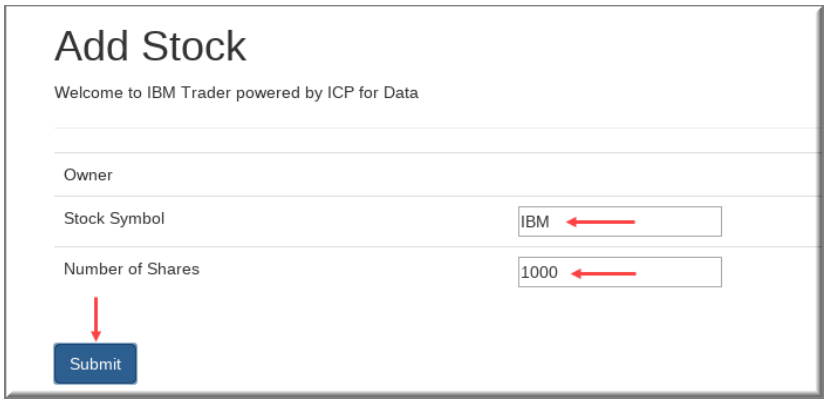
__65. Type `TechStocks` as name of the portfolio and click `Submit`.



__66. Select `Update selected portfolio` and click `Submit`.

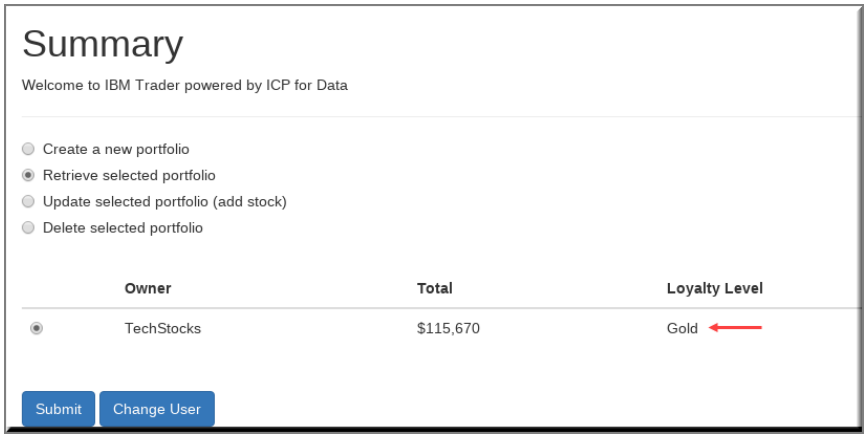


__67. Type `IBM` and `1000` and click **Submit** to buy 1000 stocks of IBM. [Note: You can choose any other, if you know the stock symbol.]



	Note: If Internet access is available – the stock quotes are taken from `quandl.com` using IBM API Connect through IBM Cloud. The cached values from Quandl are stored in redis.

__68. Notice that the loyalty level changed to Gold.



__69. The Ingress rules are defined such that the `notification-service` can trigger a notification to either `Slack` or `Twitter`.

__70. If `notification-service` sent it to Twitter, you can check the message at `https://twitter.com/ibmstocktrader`.

__71. The goal is to enhance this application with analytics modernization to increase revenue and profits. We will name this application as "`Stock - After`".

__72. The modern Stock Trader application built on modern IBM Cloud Private platform uses agile technologies with a modern CICD pipeline to implement changes rapidly.

	**Note:** Gone are the days of "Code Freeze" – Welcome to the modern CICD world.

2.7 Run Stock-Trader After Application

__73. Companies that are able to keep pace with the technological breakthroughs will have a competitive advantage over companies that do not.

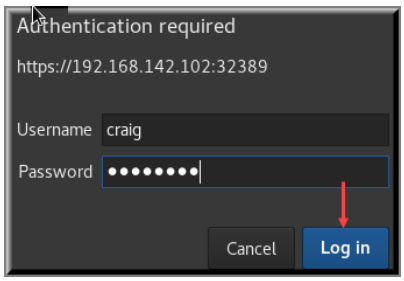
__74. Modernization of your application is the essence, but ICP-D goes one step further – it allows you to use your data to modernize your analytics capabilities.

__75. The key here is to speed the organization of the data and make it ready for machine learning and artificial intelligence to provide analytics modernization.

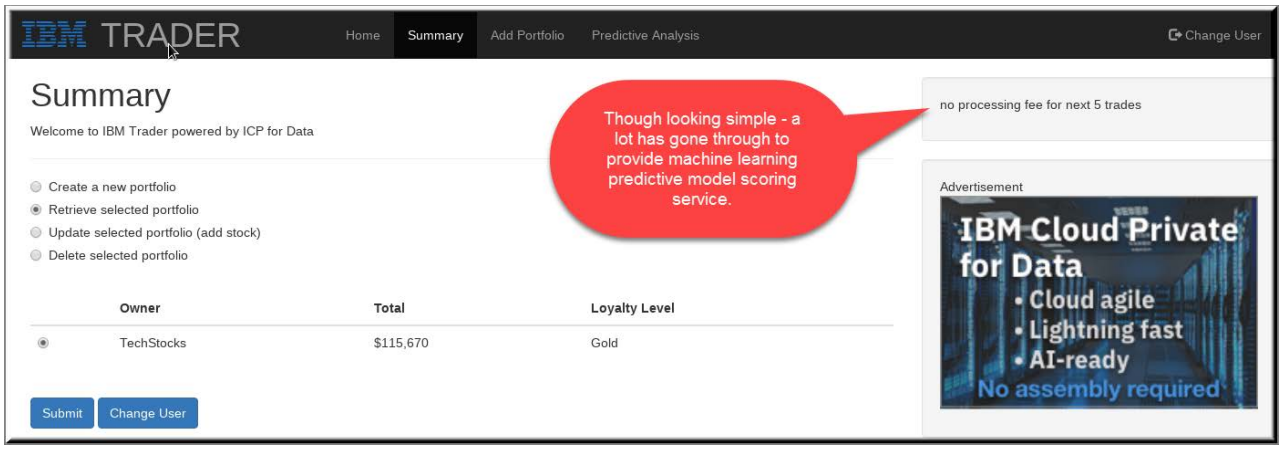
__76. Let's "begin with the end in mind" and take a look at the finished product. Then we will dive into the steps that lead us to `Analytics Modernization`.

__77. Double-click `Stocks After.`

__78. Log in using Username: `craig` and Password: `password.`



__79. Notice the offer "`no processing fee for next 5 trades`".



__80. A predictive model that was built (a journey through which we will go in subsequent lab exercises) looks for the score of the user "`craig`" and returns a number of parameters. Our new predictive analysis microservice uses a single parameter and transforms it into an offer to the customer based on the "separation risk" that the model predicts.

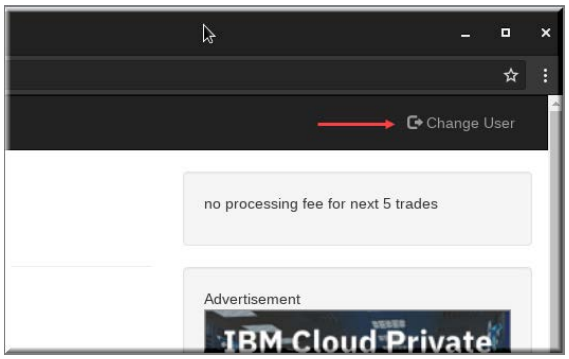
__81. This will become clear as you work through details in the `Analyze` lab exercise.

__82. Of course, models are only as good as the data they use. We will use `Organize` lab to classify, catalog, curate and establish the lineage. This will provide the best relevant and necessary data to the `Analyze` phase – where models are built, tested, scored and continuously improved.

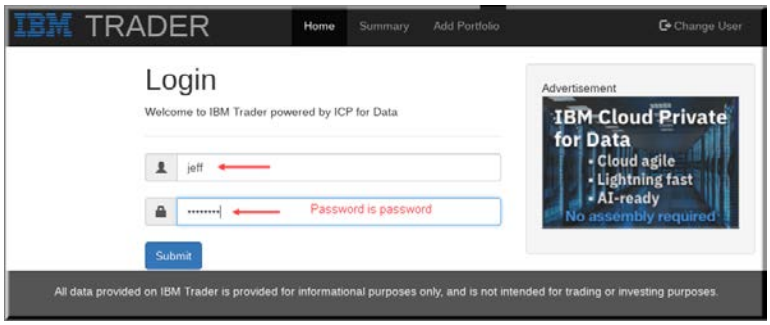
__83. Before we `Organize` the data, we first must collect it – regardless where it resides. The `Collect` exercise shows you the process of getting data.

	**Note:** Remember the term "Borderless Data".

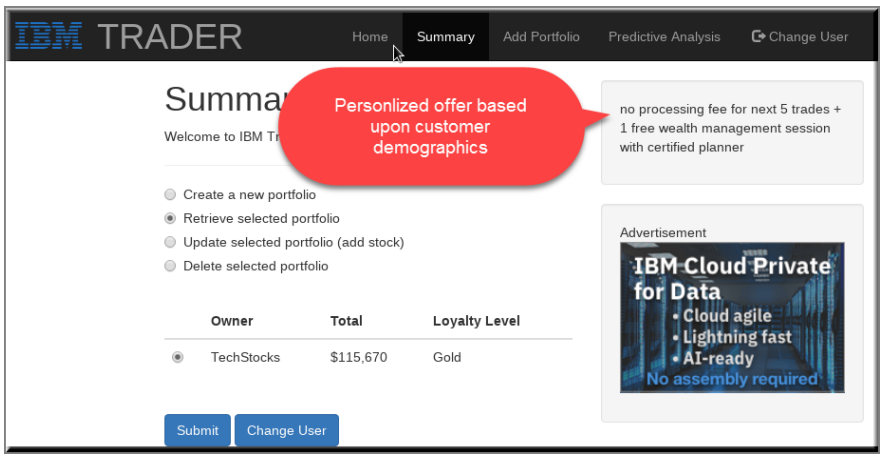
__84. Click `Change User`.



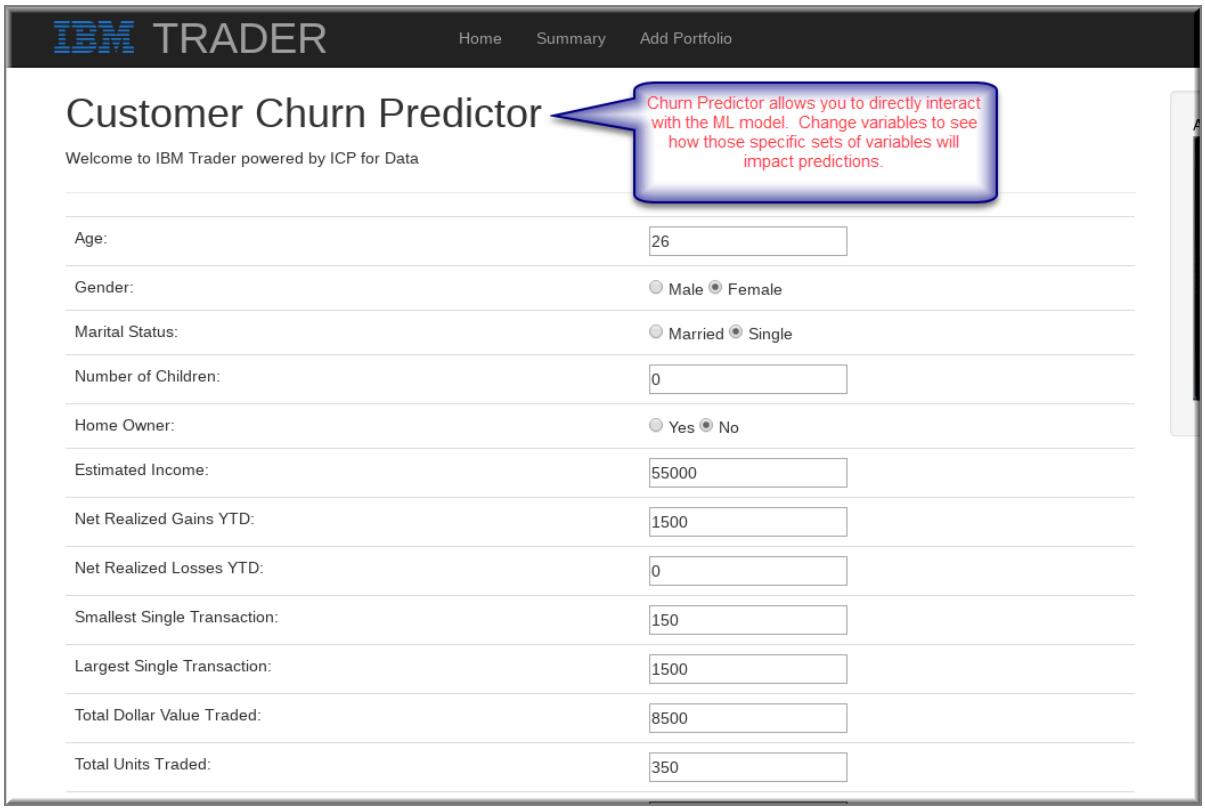
__85. Specify credentials: `jeff` and `password`.



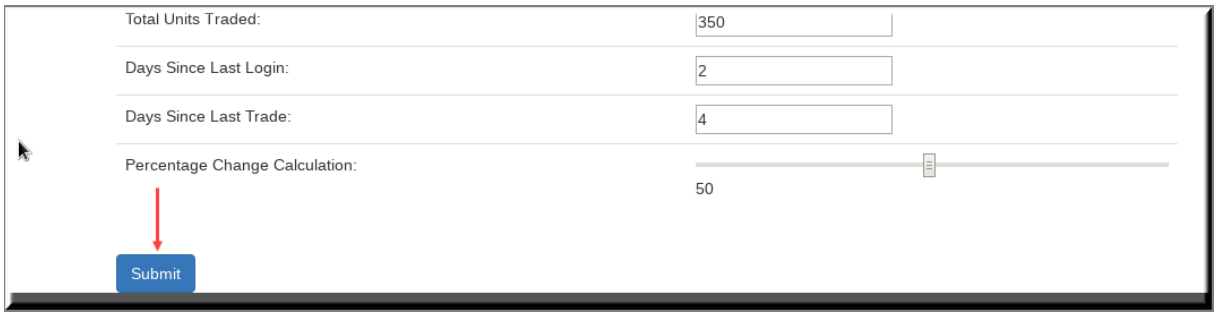
__86. Notice the personalized award offer.



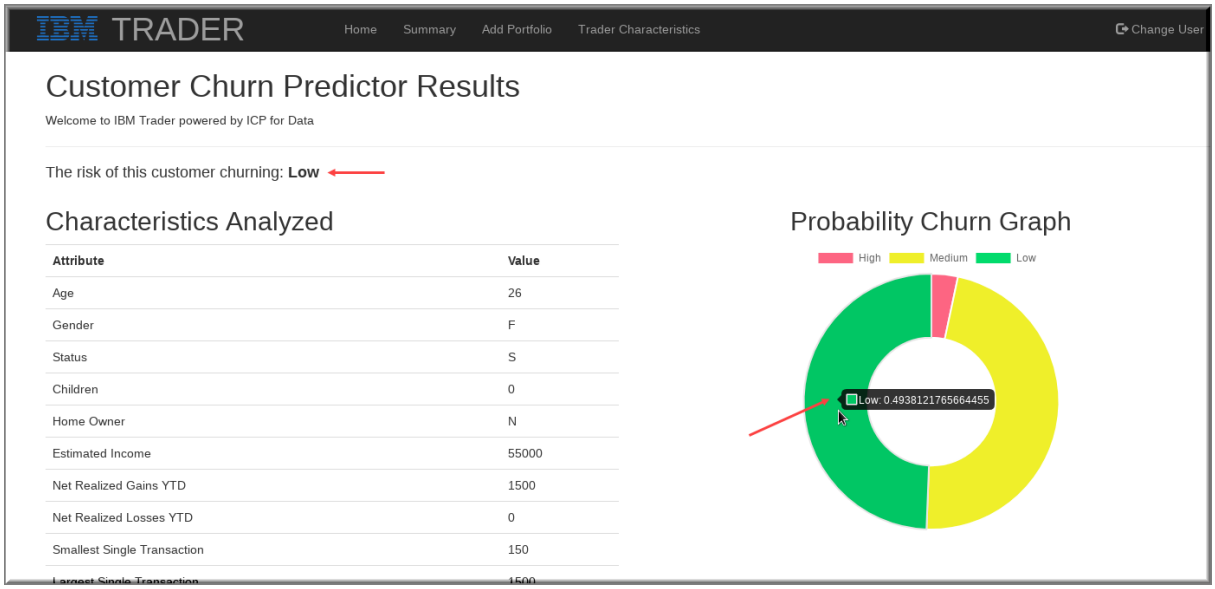
__87. Let's dive deep in behind the scene ML model prediction. Click `Predictive Analysis`. This screen is built to directly interact with the ML model. Change variables to see how those specific sets of variables impact predictions.



__88. Scroll down and click `Submit` to assess the separation risk based on default parameters.

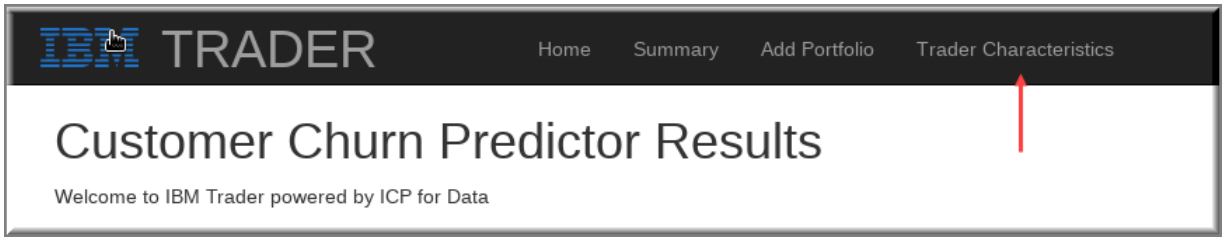


__89. This is the result using the default values of separation risk predictor. In this case, there is a *Low* to moderate risk of this customer leaving. This customer has a profitable portfolio.



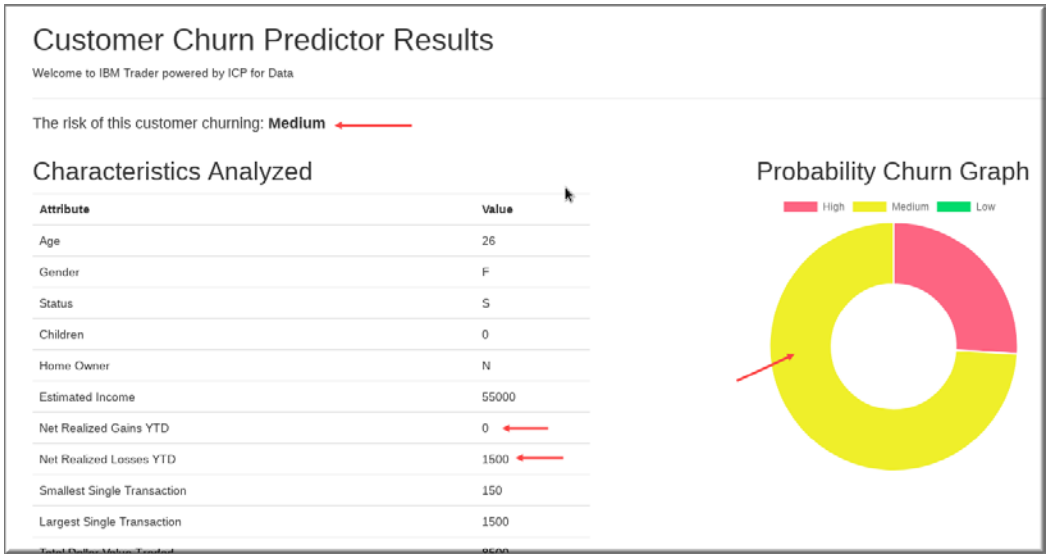
__90. Let's change some parameters.

__91. Click `Trader Characteristics`.

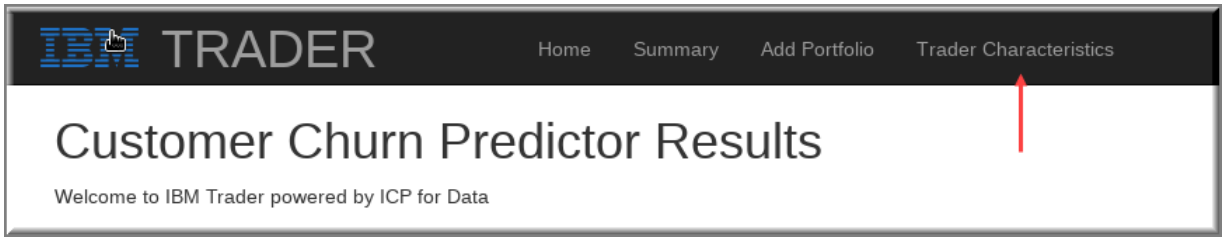


__92. Change `Net Realized Gains YTD` from `1500` to `0` and change `Net Realized Losses YTD` to `1500` from `0`. Basically, making this portfolio as not profitable. Scroll to the bottom and click `Submit`.

__93. Notice the probability of separation risk of this customer. The risk has changes from `Low` to `Medium` given the fact that portfolio of this customer is no longer profitable.

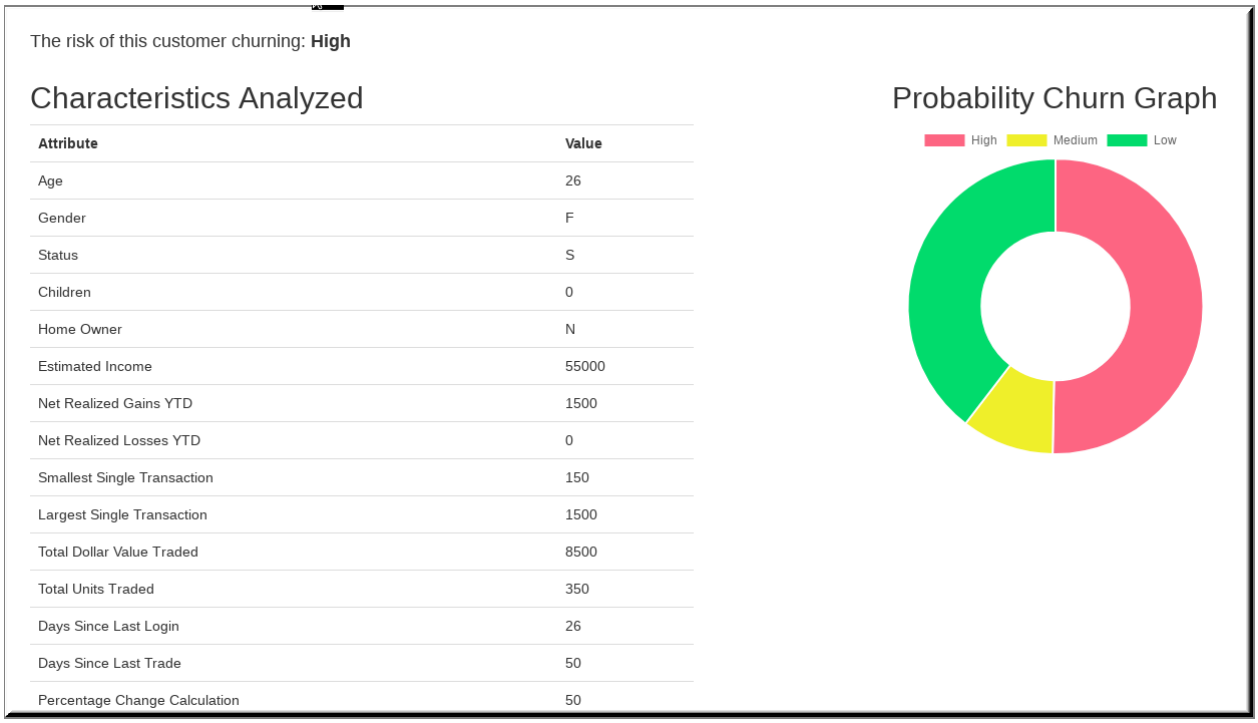


__94. Let's make some more modifications. Click `Trader Characteristics`.



__95. The default parameters are restored and the portfolio is profitable again. Scroll down and let's change the `Days Since Last Login` from `2` to `26` and `Days Since Last Trade` from `4` to `50`. Click `Submit`.

__96. Notice that the separation risk changes to `High`. It's been a while since we've seen this customer and there is little account activity. In this case, there is a high probability this customer might leave.



Attribute	Value
Age	26
Gender	F
Status	S
Children	0
Home Owner	N
Estimated Income	55000
Net Realized Gains YTD	1500
Net Realized Losses YTD	0
Smallest Single Transaction	150
Largest Single Transaction	1500
Total Dollar Value Traded	8500
Total Units Traded	350
Days Since Last Login	26
Days Since Last Trade	50
Percentage Change Calculation	50

__97. Regardless of your problem classification domain, ICP-D provides the tools needed to access, clean, shape and govern your data in preparation for ML model development, release and continuous improvement framework.

__98. From automatic ML model generation (no coding) to comprehensive tool sets to develop complex models (with coding), ICP-D provides the tools necessary to produce your specific ML model classification.

__99. Once ML models are deployed, they are easily consumed by applications as RESTful-compliant services. The deployed models are easily scalable to meet usage demands as they are packaged as independent scalable docker containers.

__100. As probabilities are consumed from one or more models, application logic can be applied to allow applications to make intelligent business decisions based on a set of predefined business rules and workflows.

__101. Let's deep dive in **Collect**, **Organize**, **Analyze** and **Deploy** features of the ICP-D as an integrated data platform.

	**Note:** Do you talk about getting value from the "Dark Data"?

**** End of Lab 02: Executive Demo**

Lab 03 Collect

Collect data regardless where it resides. This is the key aspect of data virtualization to access, govern and analyze it without duplication and massive data movement.

One step further is to use streaming data from IoT devices, banking transactions and more. The Db2 Event Store's add-on feature in ICP-D provides this capability.

3.1 Multiple Sources

__1. ICP-D has add-in capabilities for Db2 OLTP, Db2 Warehouse and Db2 Event Store to install and upgrade at any time to in the platform itself. The add-on capability will support additional data sources from MongoDB, DataMeer, and others in the future.

__2. The support for data sources available outside ICP-D platform are available for `BigSQL`, `Oracle`, `Db2 for z/OS`, `Hive` for Hortonworks and Cloudera, `HDFS` and `Informix`. Additional support for data sources is available through the `Organize` category. The list of connected data sources is designed to grow.

3.2 Integration with the Hadoop Platform

__3. The Hadoop cluster has several services which are insecure (using HTTP) and hence the Hadoop cluster is behind the firewall of an organization.

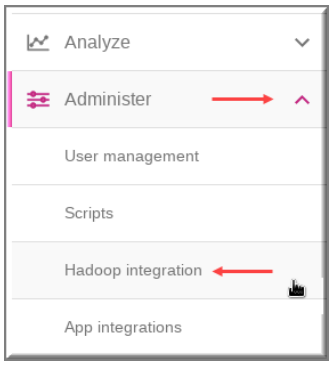
__4. ICP-D provides integration to Hadoop cluster for Hortonworks and Cloudera through a secured gateway built and customized using Apache KNOX.

__5. The integration with the Hadoop cluster requires an additional software package that needs to be installed on one of the Hadoop edge nodes (AKA Gateway).

__6. This has already been done in this lab environment. You can refer to Appendix-C for the necessary details for integration software to be installed on Hadoop edge node for the integration with ICP-D.

__7. Switch the ICP-D web console tab in your browser. If you do not have web console running, double-click `IBM Cloud Private for Data` on the desktop.

__8. Hover your mouse over the left menu bar and click the **Administer** down arrow to expand it and then click `Hadoop Integration`.



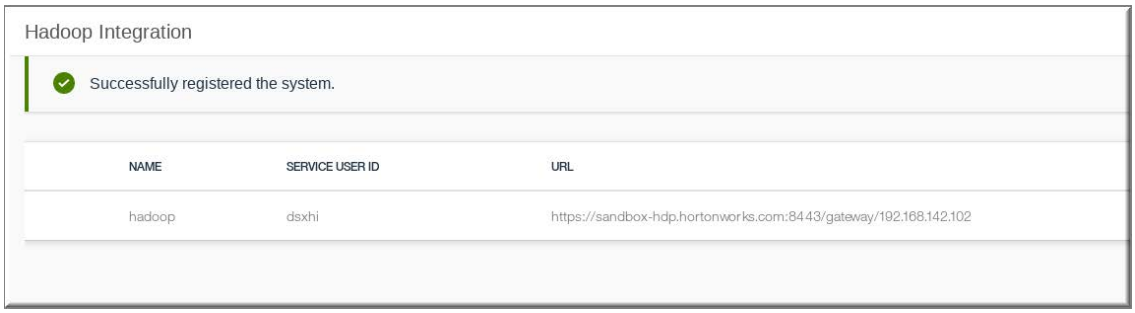
__9. Determine whether our existing Hadoop cluster is registered.

__10. If not, click `Add Registration`. [top, right-hand corner.]

__11. Type:

__a. Display Name: `Hadoop`
__b. Service URL: `https://sandbox-hdp.hortonworks.com:8443/gateway/192.168.142.101`
__c. Service User ID: `dsxhi`
__d. Click `Add`.

__12. You should see the registration success message.



3.3 Data Virtualization from Hadoop Platform

__13. Let's verify a file in Hadoop platform that we will use for data virtualization.

3.3.1 Copy a File to Hadoop

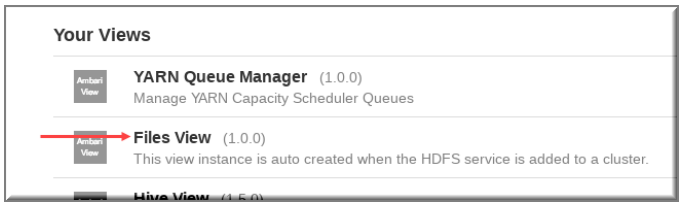
__14. Double-click `Hadoop Web Console` from the VM desktop.

__15. Click `Launch Dashboard` when the HDP Web UI appears.

__16. Type Username: `admin` and Password: `password`.

__17. From main menu bar, click the 3x3 checkerboard .

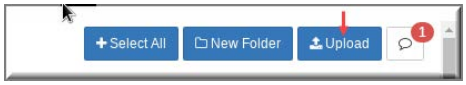
__18. Click `Files View`.



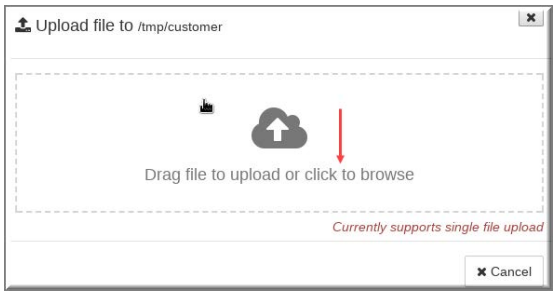
__19. Click `tmp` folder.

__20. Click `customer`.

__21. Maximize your browser screen, if not done already. Click `Upload`.



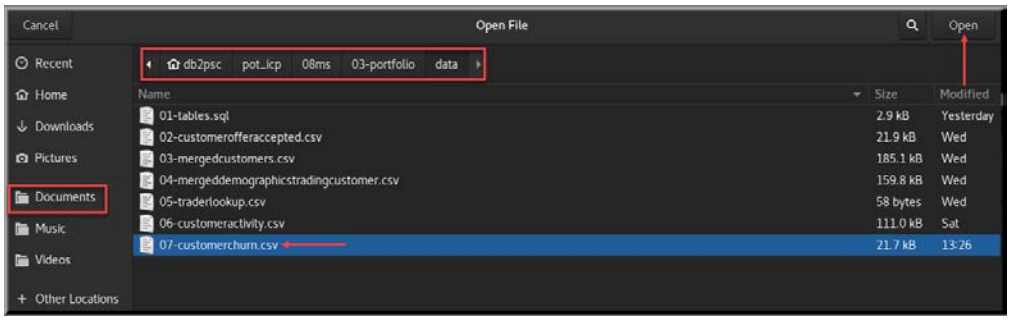
__22. Click to browse.



__23. Click `Documents` in the left pane and click `pot_icp` in right pane.

__24. Click each `08ms ⇨ 03-portfolio ⇨ data ⇨ 07-customerchurn.csv`

__25. Click `Open`

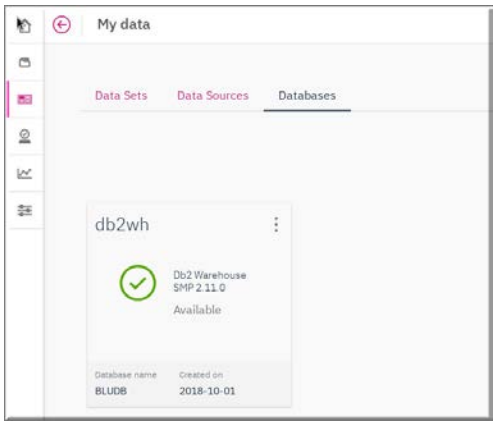


__26. Close the Hadoop browser window and the Hortonworks tab from the main browser window.

3.4 Virtualization of Db2 in ICP-D

__27. Note that ICP-D has an add-on for Db2 Warehouse. This means that the Db2 Warehouse data from within ICP-D platform is already available.

__28. Click `Collect` ⇨ `My Data` ⇨ `Databases`



__29. Let's virtualize Db2 OLTP database which we did not add as an add-on in this case.

__30. Click `Projects` ⇨ `TradingCustomerChurn`

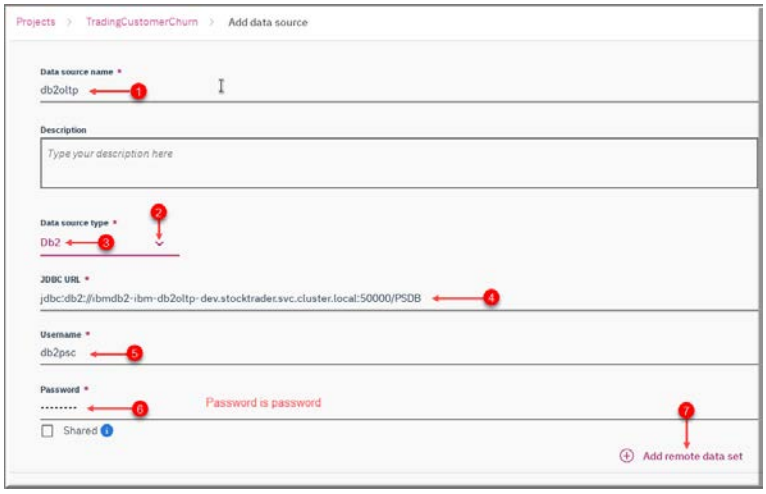
__31. Click `Data Sources`.

__32. Click `add data source`. [Towards right side of Data Sources].

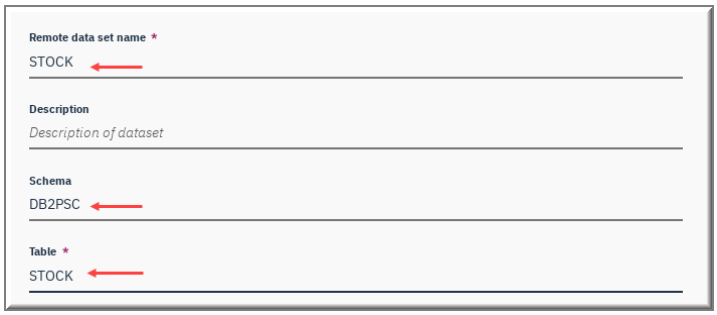
__33. Type the data source name `db2oltp`, Select `Db2` from the drop-down menu, and type the following JDBC URL: [No typos please.]

`jdbc:db2://ibmdb2-ibm-db2oltp-dev.stocktrader.svc.cluster.local:50000/PSDB`

__34. Type Username: `db2psc` and Password: `password`. Click `Add remote data set`



__35. Type *Remote data set name* `STOCK`, *Schema* `DB2PSC` and *Table* `STOCK`. [Far right bottom of screen.] (The names should be in uppercase.)



__36. Click `Create`

__37. Click `db2oltp` data source again, scroll down and click `Add data set`.

__38. Type *Remote data set name* `PORTFOLIO`, *Schema* `DB2PSC` and *Table* `PORTFOLIO` and click `Create.`

__39. Click `Save`.

3.5 Virtualization of Oracle in ICP-D

__40. Click `Add data source`

__41. Type the data source name `oracle`, Select `Oracle` from the dropdown, and type the following JDBC URL: [No typos please.]

`jdbc:oracle:thin:@192.168.142.106:1521:XE`

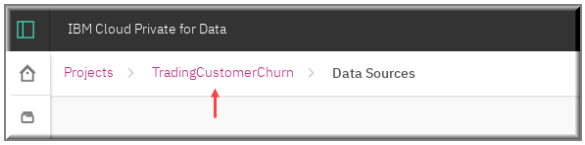
Notice: Next, you need to add `:` and `@`.

__42. Type Username: `scott` and Password: `tiger`. Click `Add remote data set`

__43. Type: *Remote data set name* `ALL_TABLES`, *Schema* `SYS` and *Table* `ALL_TABLES` and click `Create.`

3.6 Test Data Virtualization

__44. Click `TradingCustomerChurn` project.



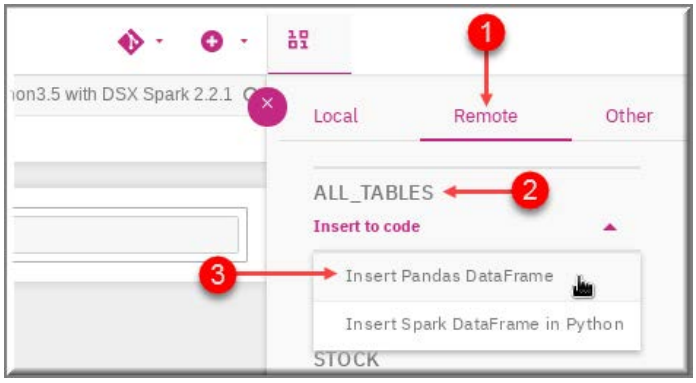
__45. Click `Assets` ⇨ `Notebooks`

__46. Click `Test` to open.

__47. Click `scissor` icon on the menu bar to delete all cells.

__48. Click `binary number` icon.

__49. Select middle tab `Remote`. Insert `ALL_TABLES` (From Oracle 11g database) Panda DataFrame to a cell.



__50. Press `Shift-Enter` to execute the cell.

__51. You should see the `ALL_TABLES` data from the Oracle database.

```
In [12]:
import dsx_core_utils, requests, jaydebeapi, os, io, sys
from pyspark.sql import SparkSession
import pandas as pd
df1 = None
dataSet = dsx_core_utils.get_remote_data_set_info('ALL_TABLES')
dataSource = dsx_core_utils.get_data_source_info(dataSet['datasource'])
if (sys.version_info >= (3, 0)):
    conn = jaydebeapi.connect(dataSource['driver_class'], dataSource['URL'], [dataSource['user'], dataSource['password']])
else:
    conn = jaydebeapi.connect(dataSource['driver_class'], [dataSource['URL'], dataSource['user'], dataSource['password']])
query = 'select * from ' + (dataSet['schema'] + '.' if (len(dataSet['schema'].strip()) != 0) else '') + dataSet['table']
df1 = pd.read_sql(query, con=conn)
df1.head()
```

Out[12]:

	OWNER	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	...	SKIP_CORRUPT	MONITORING	CLUSTER_OWNER	DEPENDENCIES	COMPRES
0	SYS	ICOL$	SYSTEM	C_OBJ#	None	VALID	0.0	0.0	0.0	0.0	...	DISABLED	YES	SYS	DISABLED	DISABLED
1	SYS	CON$	SYSTEM	None	None	VALID	10.0	40.0	1.0	255.0	...	DISABLED	YES	None	DISABLED	DISABLED
2	SYS	UNDO$	SYSTEM	None	None	VALID	10.0	40.0	1.0	255.0	...	DISABLED	YES	None	DISABLED	DISABLED
3	SYS	PROXY_ROLE_DATA$	SYSTEM	None	None	VALID	10.0	40.0	1.0	255.0	...	DISABLED	YES	None	DISABLED	DISABLED
4	SYS	FILE$	SYSTEM	None	None	VALID	10.0	40.0	1.0	255.0	...	DISABLED	YES	None	DISABLED	DISABLED

__52. Add Db2 `PORTFOLIO` in a cell, press `Shift-Enter` to get the data from Db2 data source.

```
In [13]:
df2 = None
dataSet = dsx_core_utils.get_remote_data_set_info('PORTFOLIO')
dataSource = dsx_core_utils.get_data_source_info(dataSet['datasource'])
if (sys.version_info >= (3, 0)):
    conn = jaydebeapi.connect(dataSource['driver_class'], dataSource['URL'], [dataSource['user'], dataSource['password']])
else:
    conn = jaydebeapi.connect(dataSource['driver_class'], [dataSource['URL'], dataSource['user'], dataSource['password']])
query = 'select * from ' + (dataSet['schema'] + '.' if (len(dataSet['schema'].strip()) != 0) else '') + dataSet['table']
df2 = pd.read_sql(query, con=conn)
df2.head()
```

Out[13]:

	OWNER	TOTAL	LOYALTY
0	TechStock	141500.0	Gold

__53. Similarly Add `STOCK` data set in a cell and press `Shift-Enter` to execute the cell.

```
In [14]:
df3 = None
dataSet = dsx_core_utils.get_remote_data_set_info('STOCK')
dataSource = dsx_core_utils.get_data_source_info(dataSet['datasource'])
if (sys.version_info >= (3, 0)):
    conn = jaydebeapi.connect(dataSource['driver_class'], dataSource['URL'], [dataSource['user'], dataSource['password']])
else:
    conn = jaydebeapi.connect(dataSource['driver_class'], [dataSource['URL'], dataSource['user'], dataSource['password']])
query = 'select * from ' + (dataSet['schema'] + '.' if (len(dataSet['schema'].strip()) != 0) else '') + dataSet['table']
df3 = pd.read_sql(query, con=conn)
df3.head()
```

Out[14]:

	OWNER	SYMBOL	SHARES	PRICE	TOTAL	DATEQUOTED
0	TechStock	IBM	1000	141.5	141500.0	2018-10-15

Note: The data virtualization gives you the ability to `Collect` your hard to reach data in a single ICP-D platform.

**** End of Lab 03: Collect**

Lab 04 Organize

Most organizations find it difficult to understand their own data because it is dispersed across many silos and controlled by different teams.

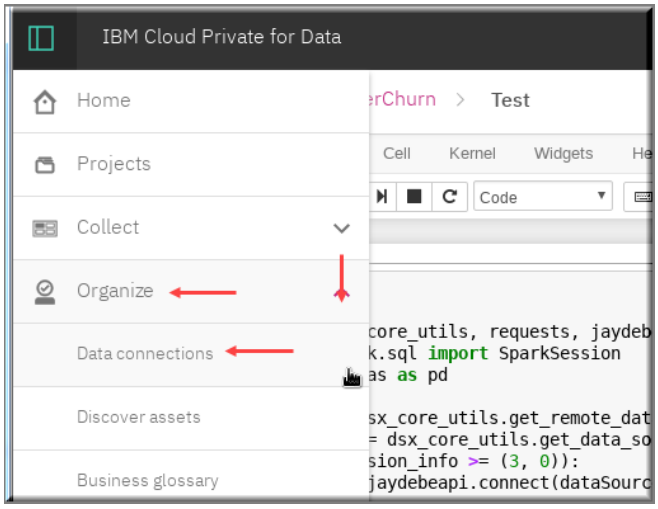
The lab shows you how to uncover hidden data and to build a lineage that is otherwise difficult to establish. ICP-D helps you move from manual processes to establish relationships between data to automation, aided by machine learning.

Let's begin to understand how this task is simplified by ICP-D.

4.1 Create Connections

__1. Switch to the `ICP-D` Web UI.

__2. Hover your mouse on the left-menu bar and click `Organize`, then click the down arrow and select `Data connections`.



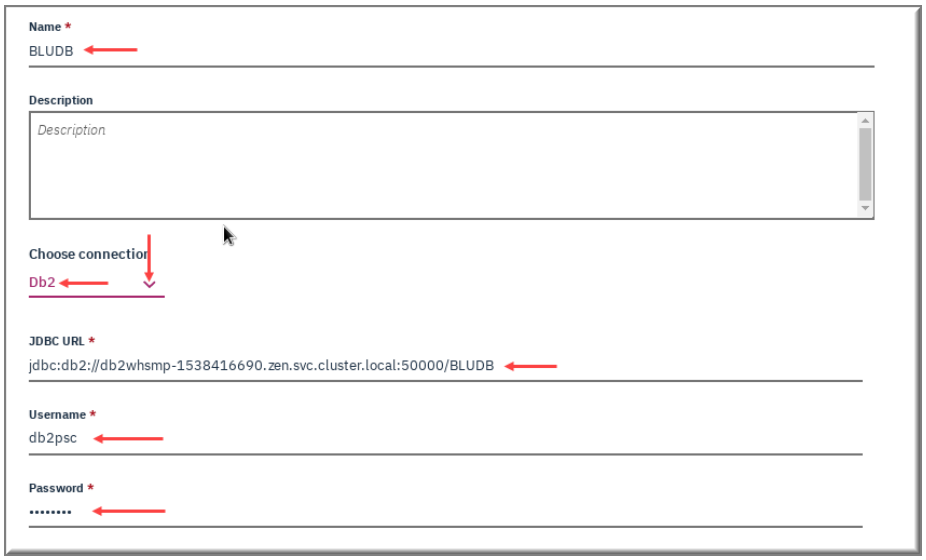
__3. Click `Create Connections`.

__4. Type: Name `BLUDB`, select `Db2` from the drop-down menu and type JDBC URL exactly as shown below:

`jdbc:db2://db2whsmp-1538416690.zen.svc.cluster.local:50000/BLUDB`

__5. [No typos please.]

__6. Enter Username: `db2psc` and Password: `password`.

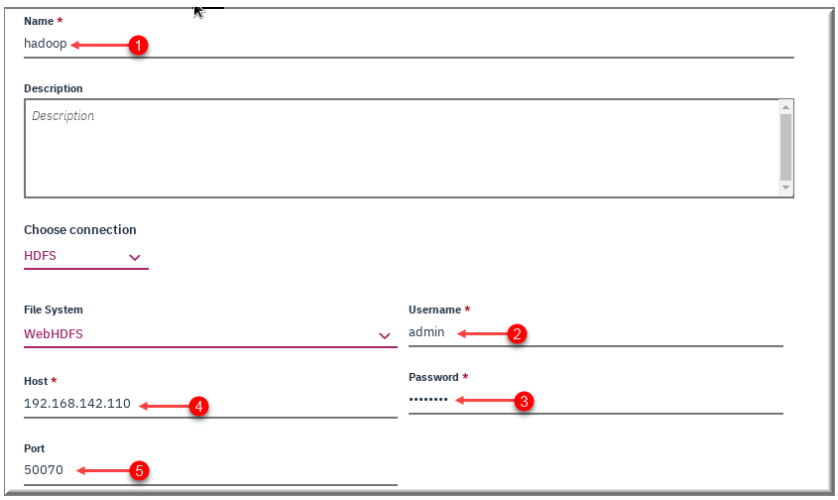


__7. Click `Test Connection`.

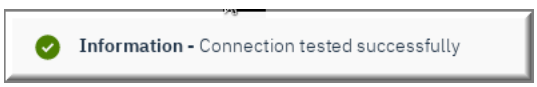
__8. After successful connection, click `Save Connection`.

__9. Under `Organize ⇨ Data connections`, click `Create Connection`.

__10. Enter *Name* `hadoop`, keep *connection* `HDFS`, and keep *File System* `WebHDFS.` Enter *Username:* `admin`, *Password:* `password`, *Host:* `192.168.142.110` and *Port:* `50070`.

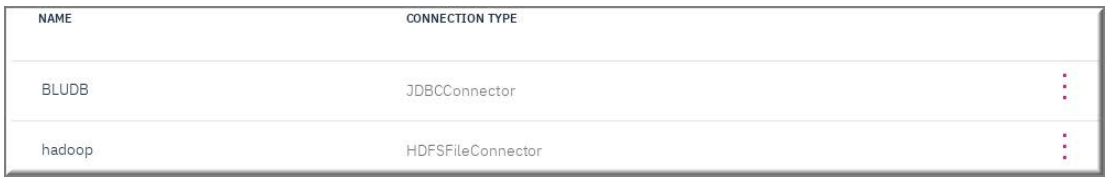


__11. Click `Test Connection`.



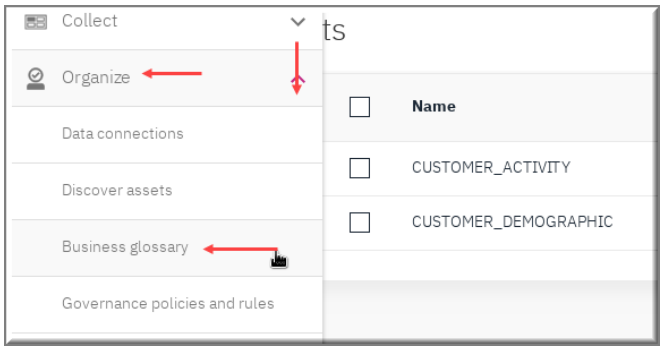
__12. After successful connection, click `Save Connection`.

__13. Two connections are created.



4.2 Create Business Glossary

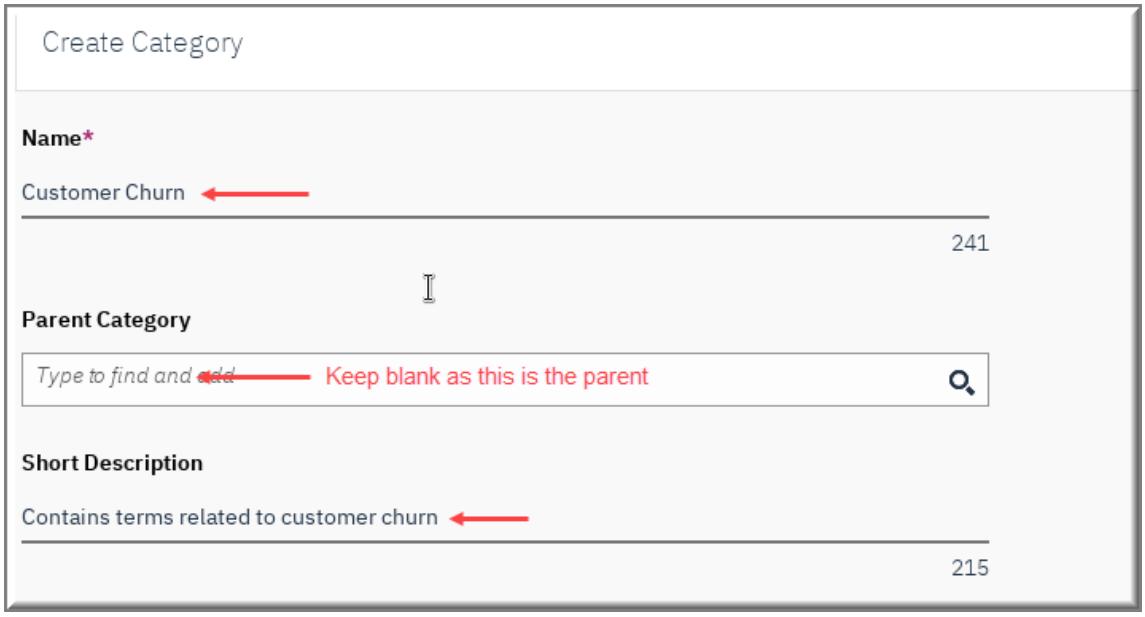
__14. Navigate on the left menu to `Organize` ⇨ `Business Glossary`



	Note: The business terms that we enter help users understand the data. These will automatically be associated with our data sources in a later step. Generally speaking, these terms would either already exist within a client's environment, or there would be a governance board tasked with creating a glossary, prior to onboarding new data sources.

__15. Click `Create Category`.

__16. Type *Name:* `Customer Churn` and description: `Contains terms related to customer churn.`

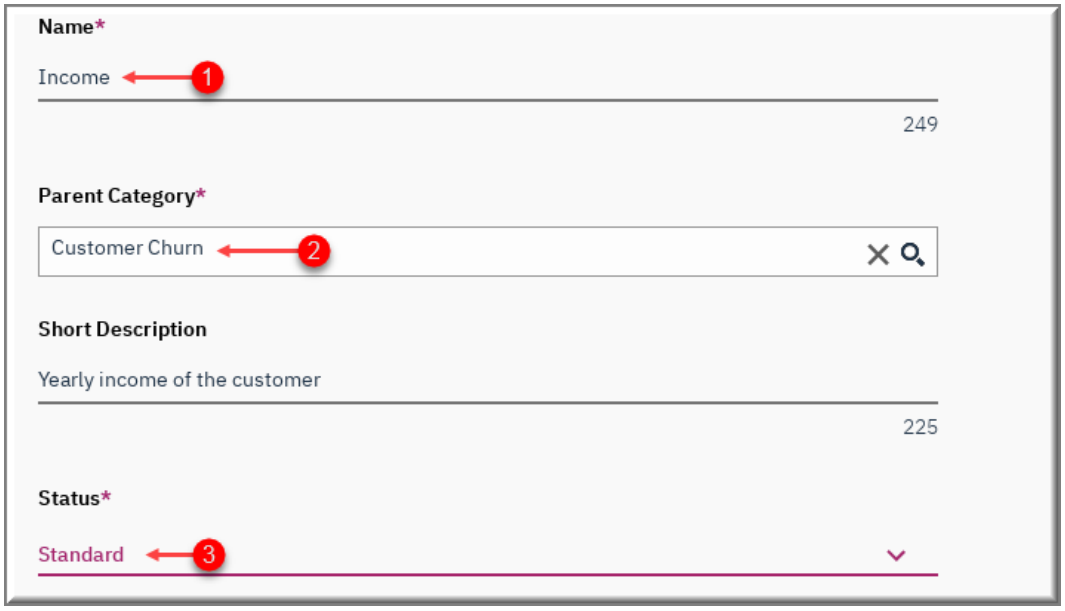


__17. Click `Save` [Towards right bottom of the screen.]

__18. Go to the second tab `Terms` and click `Create Term`.

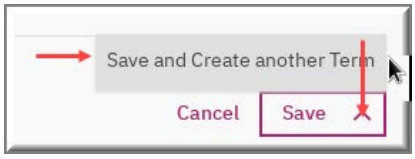
	Note: What is a Term? – Think of it as a building block of a business glossary.

__19. Create a term `Income` with a parent category `Customer Churn` (Type `C` in the box), provide a description as: `Yearly income of the customer` and choose *Status* as `Standard` from the drop-down menu.



Note: You can create your own glossary with terms and categories or you can use Industry Models from IBM for industries such as finance, banking, healthcare, and insurance.

__20. Click the down arrow next to `Save` and click `Save and Create another Term`.



__21. Now, repeat above exercise to create the following `Terms`. [Remember to use `Save and Create another Term`.] (The description for each term is found in the next table.)

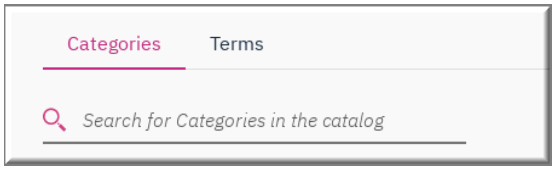
Name	Category	Status
Home Owner	Customer Churn	Standard
Net Realized Losses	Customer Churn	Standard
Net Realized Gains	Customer Churn	Standard
Gender	Customer Churn	Standard
Days Since Last Trade	Customer Churn	Standard
Total Dollar Value Traded	Customer Churn	Standard

__22. Take the `Description` from the second column for the `Term` being entered.

Name	Description
Home Owner	Flag indicating whether the customer owns a home.
Net Realized Losses	The net dollar value of losses realized for a customer, usually a year to date metric.
Net Realized Gains	The net realized gains for a customer, usually a year to date metric.
Gender	The gender of the customer, either 'M' or 'F'
Days Since Last Trade	The number of days since the customer last executed a trade.
Total Dollar Value Traded	The total amount the customer has traded with us since onboarding.

__23. For the last term, just click `Save` instead of Save and Create another Term

__24. Notice the categories `Customer Churn` that you just created and the `Terms` associated with that category. Review the `Categories` and `Terms` tab.

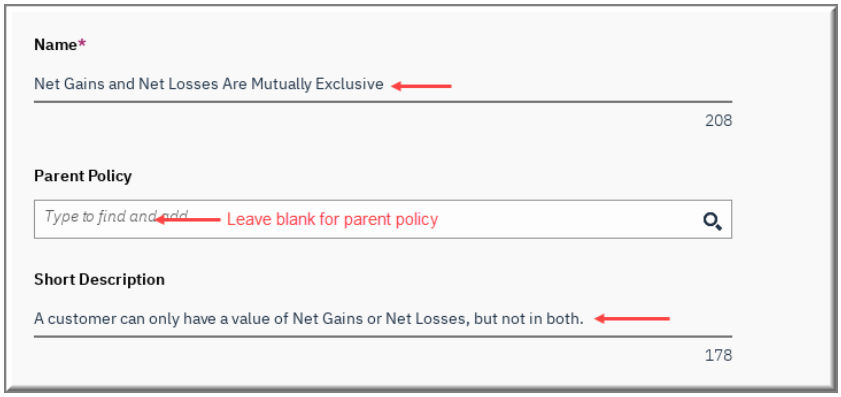


4.3 Create Governance Policies and Rules

__25. Navigate on the left menu to `Organize` ⇨ `Governance policies and rules`

__26. Click `Policies` and `Create Policy`.

__27. Type *Name*: `Net Gains and Net Losses Are Mutually Exclusive` and *Description:* `A customer can only have a value in Net Gains or Net Losses, but not in both`.



__28. Click `Save`.

__29. We just created a policy and now create a rule. Click `Create Rule` under `Rules`.

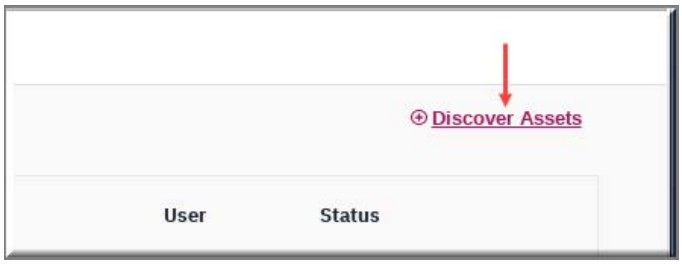
__30. Type Name: `Net Realized Gains and Losses Validity Check`, type **'n'** to select policy and type Description as `If Net Realized gains > 0 then Net Realized Losses = 0 else if Net Realized Losses > 0 then Net Realized Gains = 0.`

__31. Click `Save`.

4.4 Discover Assets

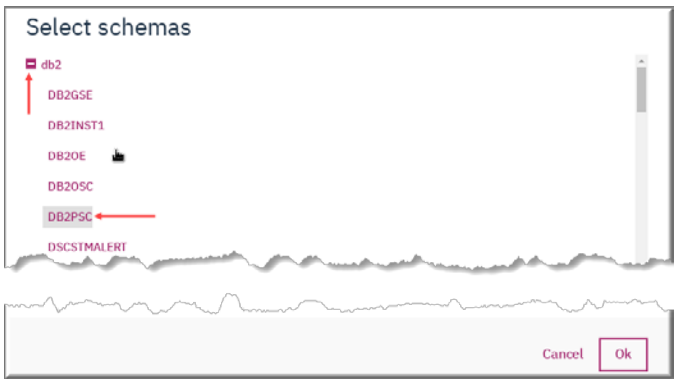
__32. Click `Organize` and expand it and click `Discover assets`.

__33. Click `Discover Assets`. [Towards top right]



__34. Select `BLUDB` from the drop-down menu and click `Browse`.

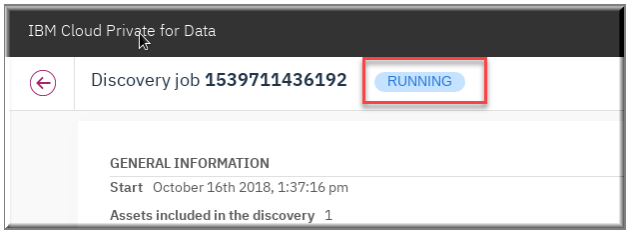
__35. Expand `db2` and click `DB2PSC` to select it and click `Ok`.



__36. Check the boxes for `Profile and classify data`, `Assign business terms`, `Analyze data quality`, and `Publish to enterprise catalog`.

__37. Click **Discover**. [Far bottom right of the screen.]

__38. A job is created and begins to execute. It may take few minutes to complete.

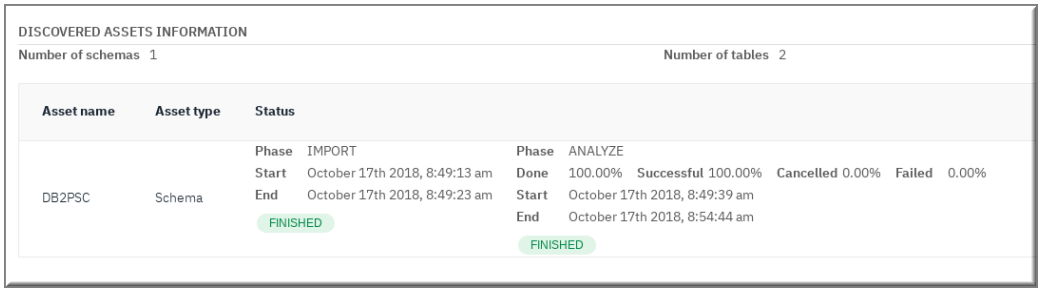


__39. Click `Refresh` to see if the job is still running and wait for this to complete.

__40. The Discovery phase has two stages – 1. `Import` and 2. `Analyze`.

__41. When it is finished, you will see that the 'Analyze' phase has a flag marked `FINISHED.`

__42. As you refresh the page, you can see the percentage complete in each phase.



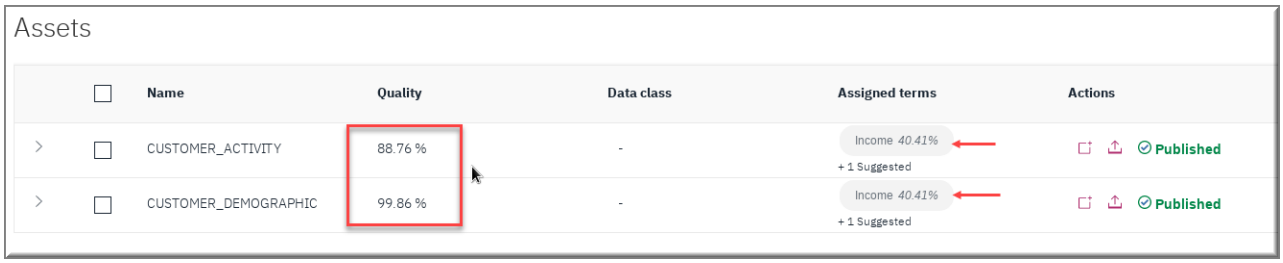
__43. Wait for this to complete. It may take 4-6 minutes.

	**Note:** What just happened? – An automated discovery, assignment a quality score and assignment of terms if there was a match with a TERM that we created. This process is incredibly powerful. What just happened is that, simply by pointing to a data source[ML1]: ICP - D has automatically profiled the data to highlight data quality, classified the data so that the business purpose is easily understood, that is, do columns contain PII data? Are some columns more significant to our analytics project than others? And finally, it has automatically assigned these data columns to business terms that we previously added to our glossary. This allows business users to easily find or 'shop' for data that they want to research or exploit.

__44. When it completes, you will see that the `Analyze` phase has a flag marked `FINISHED.`

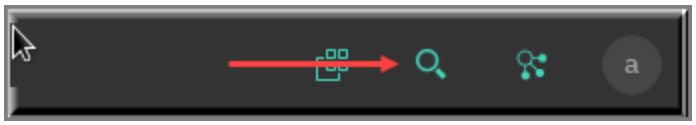
__45. Click `Review` under *Review results*.

__46. Expand `CUSTOMER_ACTIVITY` and notice the quality score assigned to a table, individual data items and match with a known data class and possible match with a Term.

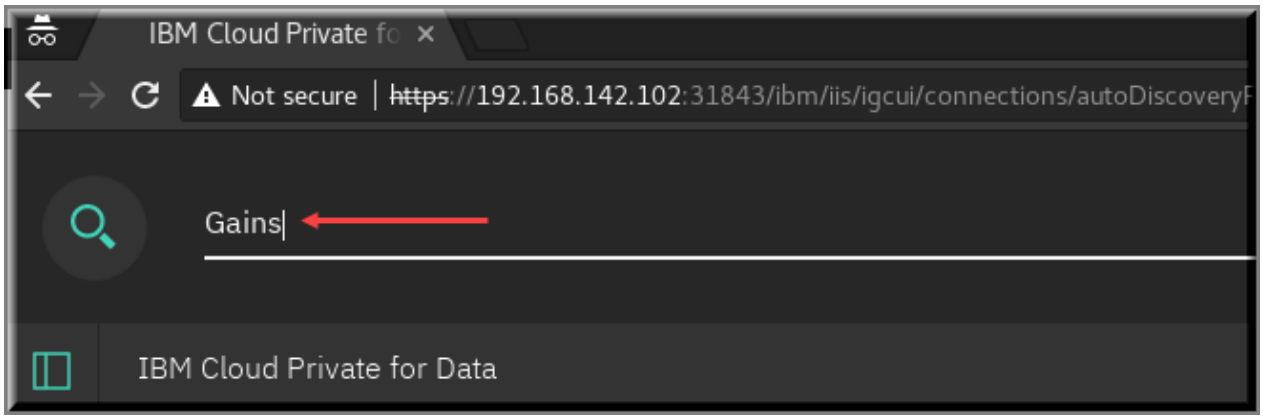


__47. The discovered assets are published to the catalog so that they are searchable.

__48. Click `Search` [Middle icon on the top menu bar towards the right.]

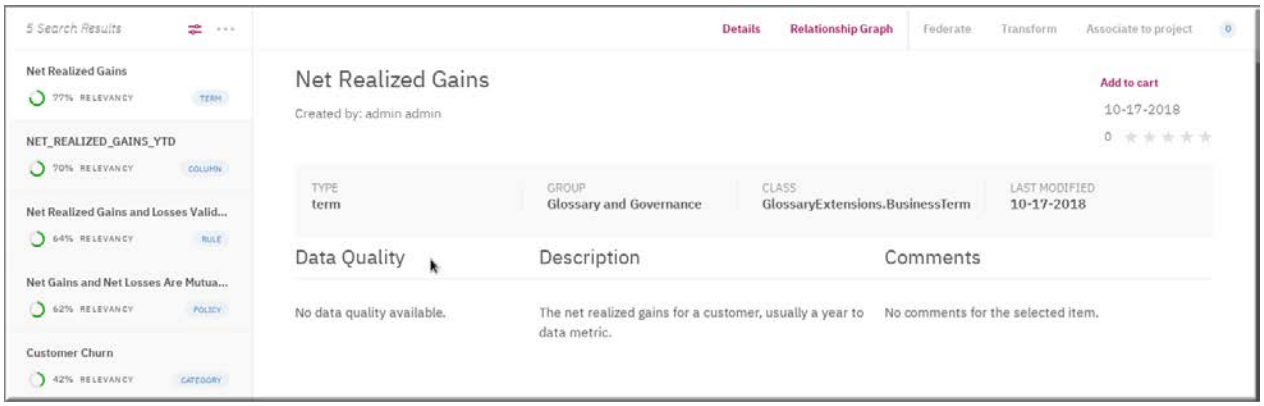


__49. Type `Gains` and click `Enter`.



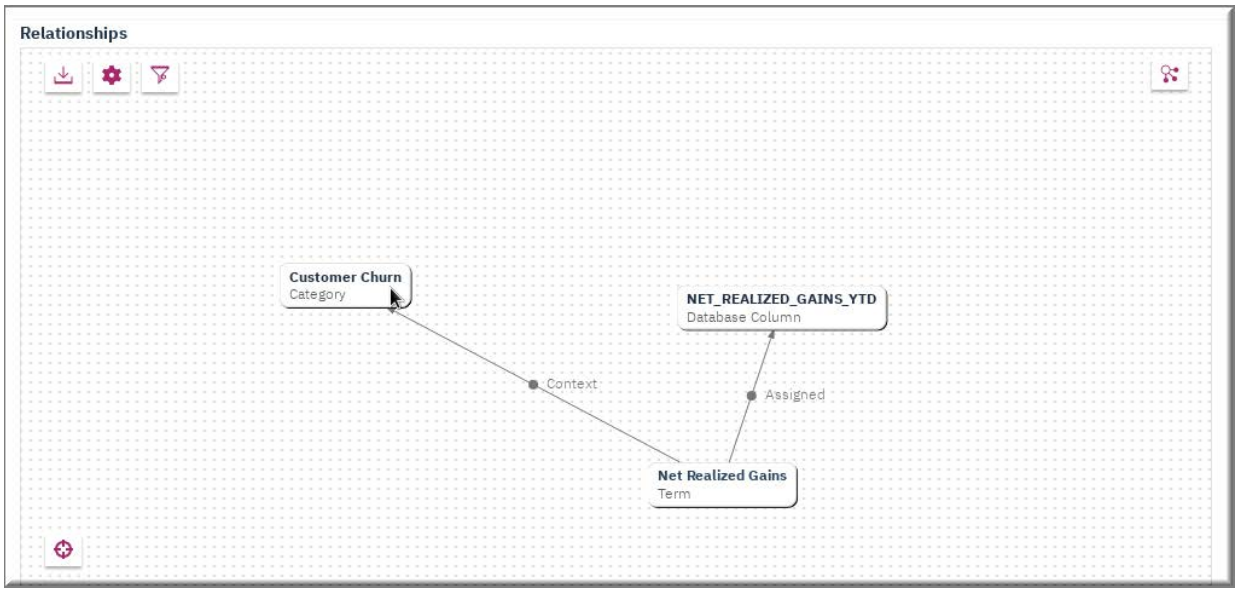
	Note: We're looking to improve our ability to predict churn probability and try to prevent it. Let's see if we have any data that indicates customer's past trading results.

__50. You see number of results and one of them is `Net Realized Gains`.



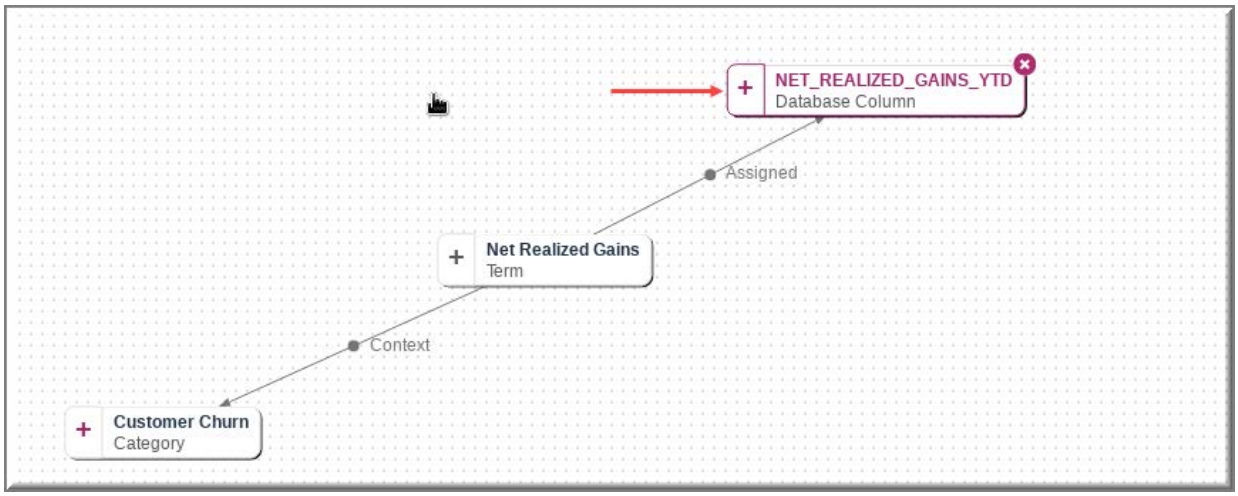
__51. Click `Relationship Graph`.

__52. The relationship graph between `Category`, `Term` and the associated `Database Column` displays.

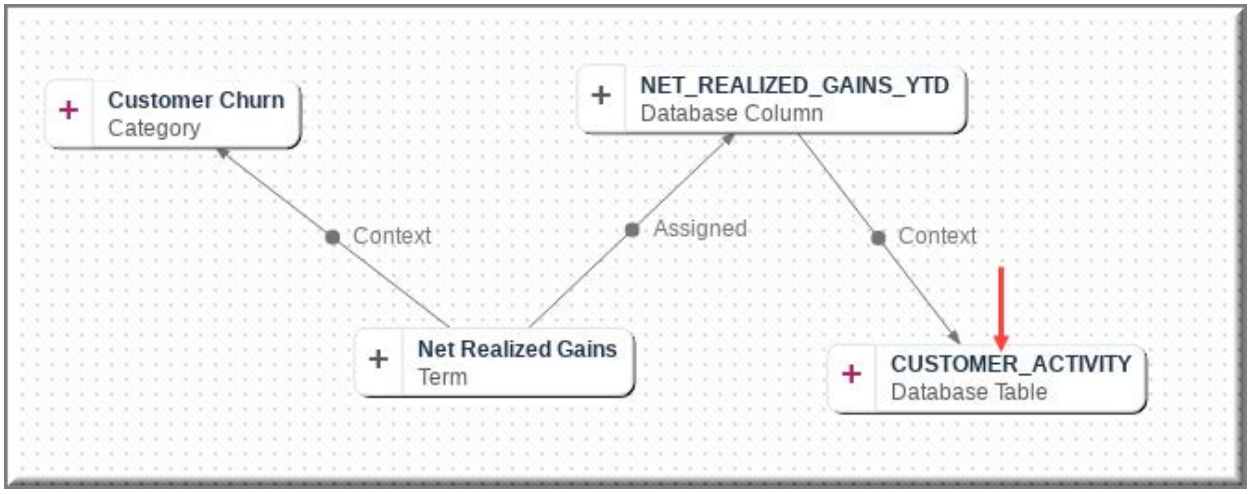


__53. Click `Explore relationships` icon at the top-right corner of the screen.

__54. Hover your mouse over `NET_REALIZED_GAINS_YTD` and click `+` sign.



__55. What did you just learn? A visual relationship matrix between a business TERM and the database table and column it finds its association.



	Note: Deriving this relationship manually is a very cumbersome, labor-intensive job, should you have thousands of tables. Let ICP-D platform do this for you with a degree of confidence that you can see, validate and collaborate with comments and star ratings.

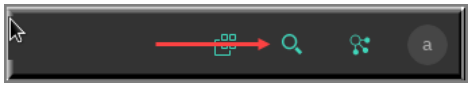
__56. Click the `CUSTOMER_ACTIVITY` box, to display the details on the right-side with the connection information.

__57. A solid discovery and classification process of your data is a recipe to find your `dark data` and build machine learning models that we will cover in next lab exercise.

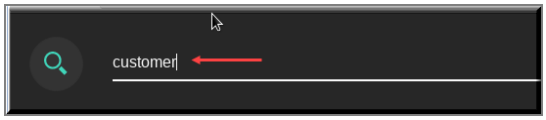
	Note: Gartner defines `dark data` as the information assets organizations collect, process and store during regular business activities, but generally fail to use for other purposes (for example, analytics, business relationships and direct monetizing).

4.5 Shopping for Data

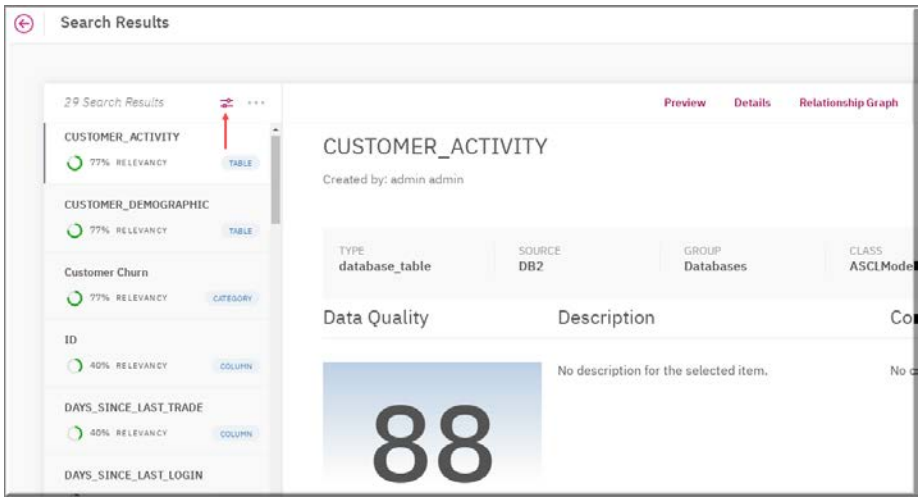
__58. Click `Search` icon.



__59. Type `customer` and click **Enter**.

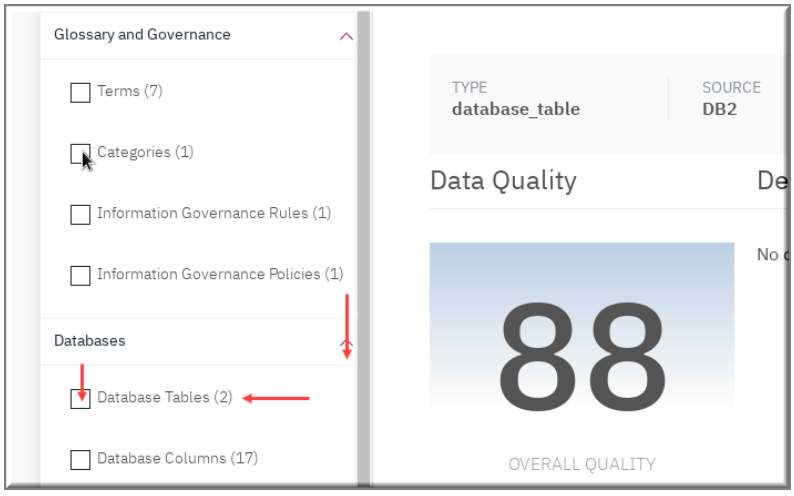


__60. Assets related to `customer` display. Click 



__61. Expand `Databases`.

__62. Check `Database Tables`.



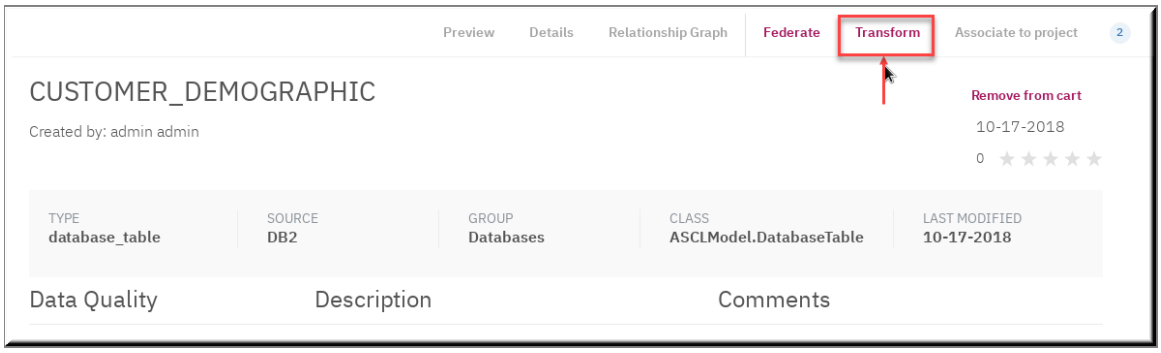
__63. Click `Properties` icon, `CUSTOMER_ACTIVITY` and then click `Add to cart`.

__64. In the same way, click `CUSTOMER_DEMOGRAPHIC` and click `Add to cart`.

	Note: We've now chosen two tables to extract. From here, we build an ETL job to extract, join and transform this data. We will also add previous customer churn data from Hadoop to create our final data set for analysis.

4.6 Transform Data

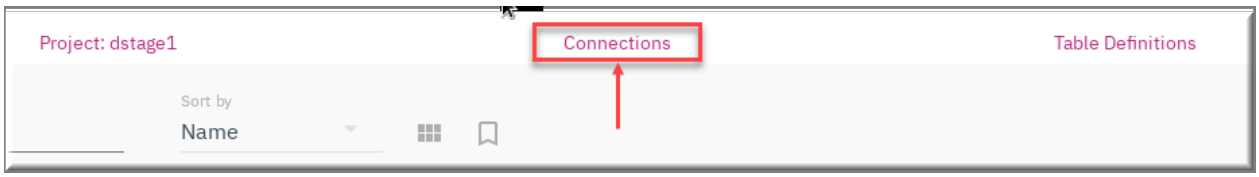
__65. Click `Transform`.



__66. A list of ETL projects displays, click `dstage1`.

	Note: After a few seconds, the screen repaints and displays the ETL canvas, with our two data sources that we just added to our cart placed onto the canvas, so that we may further develop our ETL job.

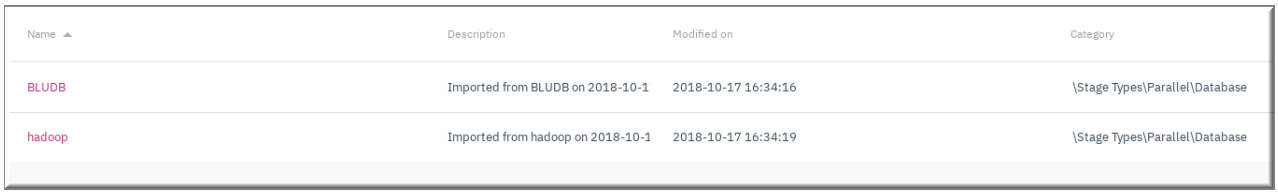
__67. Click `Connections`



__68. Click `Down arrow` [toward the right.]

__69. Check `hadoop` data source and click `Import`.

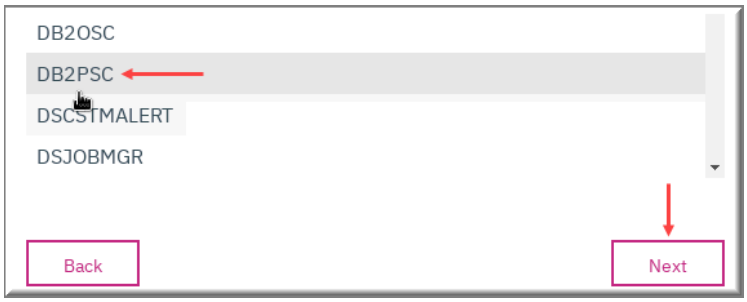
__70. You should see two connections imported.



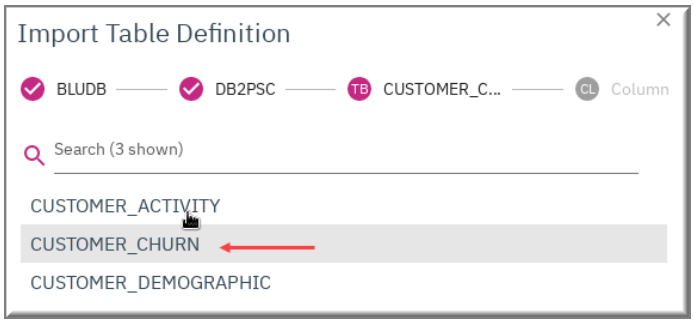
__71. Next click `Table Definitions` and click `Down arrow`.

__72. The connection `BLUDB` is selected, click `Next`.

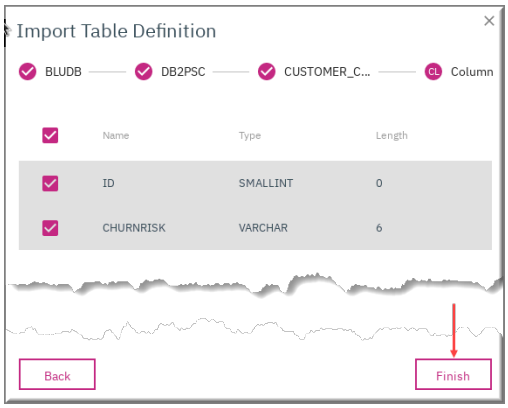
__73. Select DB2PSC schema and click Next.



__74. Select CUSTOMER_CHURN and click Next.



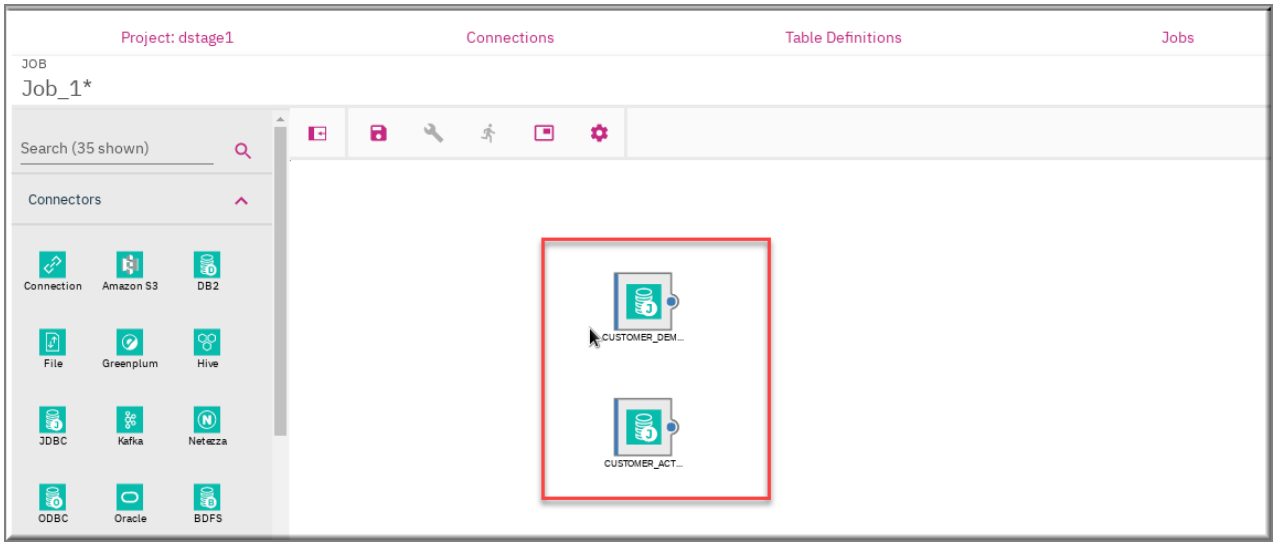
__75. Column names from the table display. Click Finish.



__76. Click Save.

__77. Click Jobs_1*.

__78. Since we had two tables CUSTOMER_ACTIVITY and CUSTOMER_DEMOGRAPHICS already in the shopping cart, those tables are added automatically to the Job_1*.



__79. Click the CUSTOMER_DEMOGRAPHICS icon on the canvas.

__80. Navigate to the palette on the left and do the following:

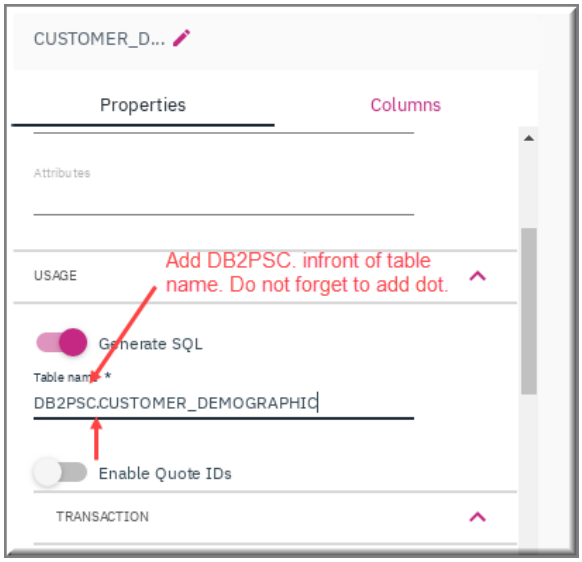
- Find the Join icon (in the Stages section.)
- Click it once
- Hold your mouse button down and drag the stage to the canvas
  - A link (shown as a grey line) should be drawn from CUSTOMER_DEMOGRAPHIC to the join stage. [If you do not see the link being drawn, remove the Join and ensure that you click CUSTOMER_DEMOGRAPHIC first.] When you drag a new stage onto the canvas (Join, in this case), the interface automatically draws a link from whatever stage on the canvas is highlighted (in this case CUSTOMER_DEMOGRAPHIC) to the stage you've just placed it on.
- Next, click the CUSTOMER_ACTIVITY stage
- Click and hold on the stage and drag a link to the join stage

__81. You should see the join between CUSTOMER_DEMOGRAPHIC and CUSTOMER_ACTIVITY.

__82. If you do not get it right it the first time, delete the Join from canvas and try again.

__83. Double-click CUSTOMER_DEMOGRAPHIC to edit some attributes. The right pane will slide-in.

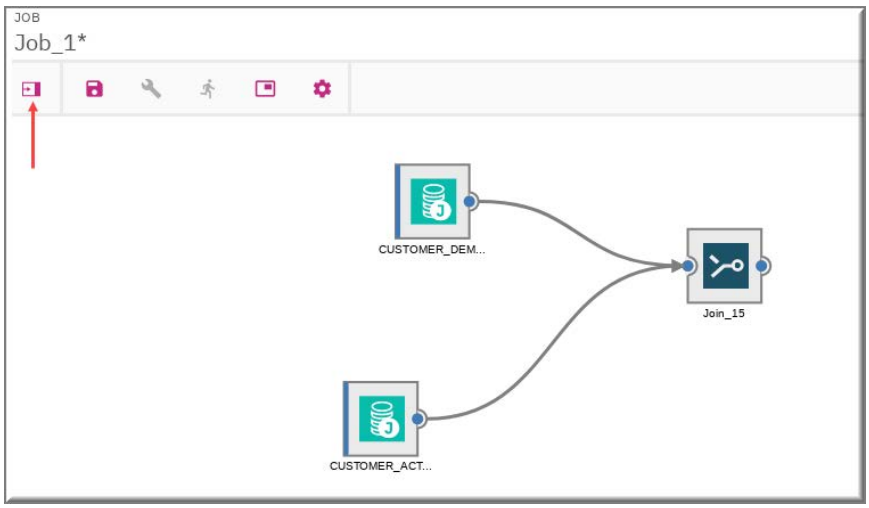
__84. Go to the `Table name` field and add the schema name `DB2PSC.` (include the **dot** after) in front of the table name `CUSTOMER_DEMOGRAPHIC`.



__85. Scroll all the way down and click `OK`.

__86. Double-click `CUSTOMER_ACTIVITY` and add `DB2PSC.` in front of the table name and click `OK`.

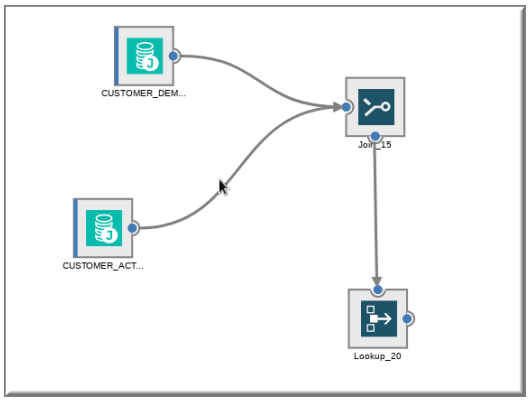
__87. Remember – you can hide/unhide the palette to use different stage artifacts, using the icon shown by the red arrow below.



__88. Click `Join` on the canvas to select it. [Make sure you select it before proceeding to the next step.]

__89. On the palette, click **Lookup** to select and then double-click it. This draws the link between `Join` stage and the `Lookup` stage.

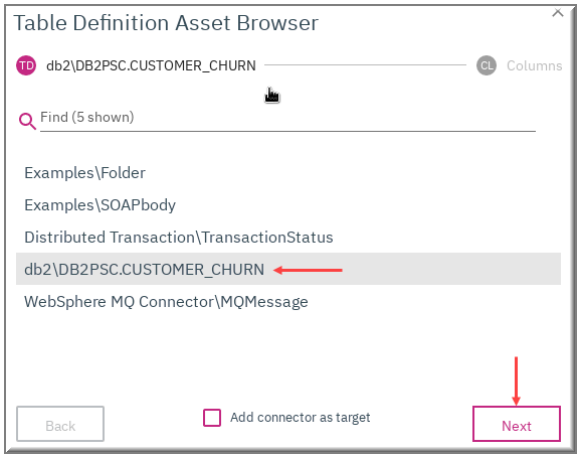
__90. Re-arrange the different stages so that `Lookup` stage is underneath the `Join` stage.



4.6.1 Join with lookup Data from HDFS

__91. **Make sure you have not selected any node.** In the Connectors palette, click `File` (not sequential file) and drag it below the `Lookup` stage. This opens a window.

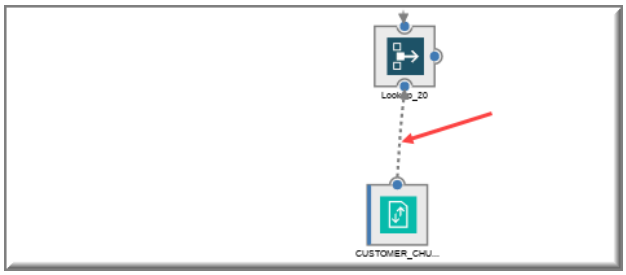
__92. Select `CUSTOMER_CHURN` and click `Next`.



__93. Click `Add to Job`.

__94. The `File` stage has a little circle link on the right side. Click that little circle and the cursor will change to a line and then click `Lookup` stage to draw the connection between two.

__95. This should be a dotted line as shown – which means that it is a *Reference*.



__96. Double-click this File stage to open edit box on the right.

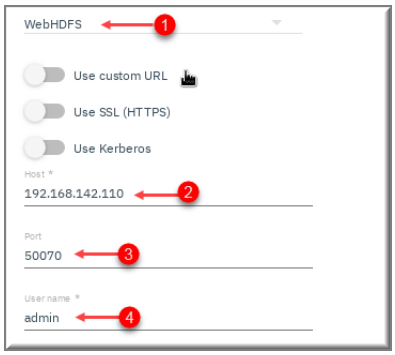
__97. Under the Connection property:

__a. Select `WebHDFS` from the drop-down menu.

__b. Type Host: `192.168.142.110`

__c. Port: `50070`

__d. Username: `admin`

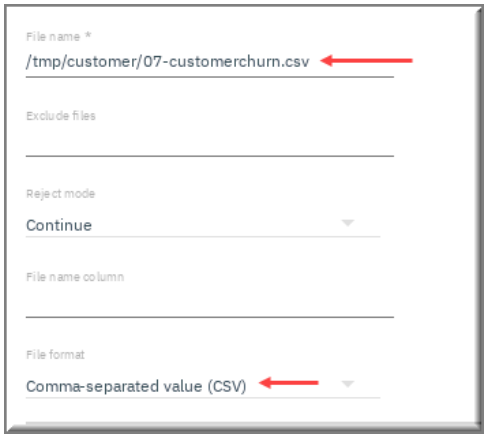


__98. Under Usage property:

__a. Type file name: `/tmp/customer/07-customerchurn.csv` [all lower case.]

__b. Select file format `Comma-Separated value (CSV)` from the drop-down menu.

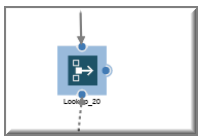
__c. Slide the slider towards right to indicate that the `First row is header`.



__99. Scroll down and click `OK`.

	**Note:** Now we need to do a few things to finish the properties in our job. First, we need to let ICP-D know what columns to use from Hadoop as a key to match on.

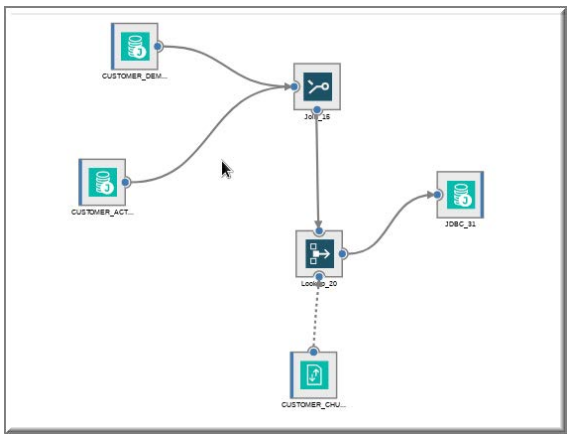
__100. On the canvas, click the Lookup stage to select it.



__101. From the palette, drag a Connection stage and drop it to the right side of the lookup stage.

__102. Check Add selected connection as target. Click Add to Job.

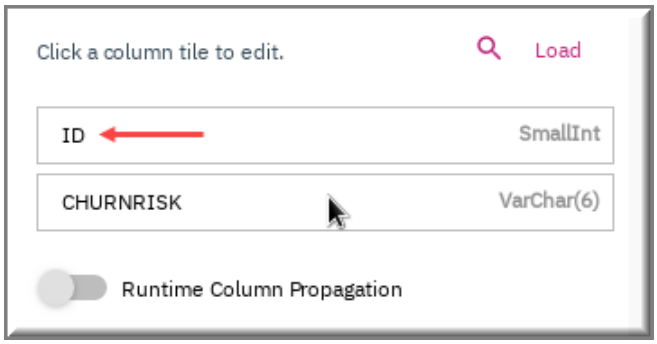
__103. This should link to the Lookup stage. Your flow should display the screenshot below.



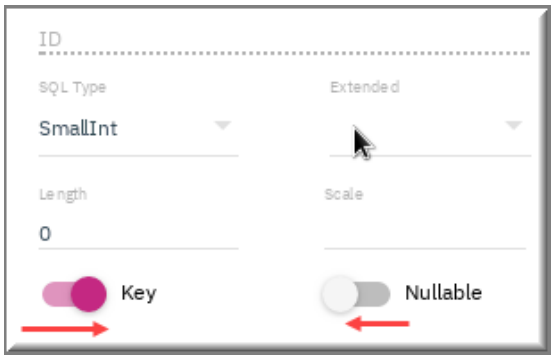
__104. Double-click CUSTOMER_CHURN stage.

__105. Select Columns tab from the right-hand pane.

__106. Click the ID field.



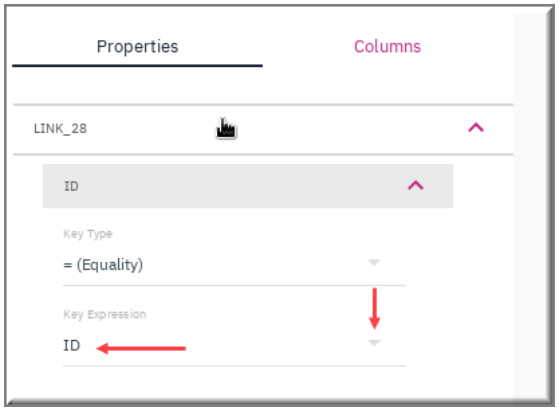
__107. Slide the `Key` slide to the right to make it a key and slide the `Nullable` slide to the left to make it non-nullable.



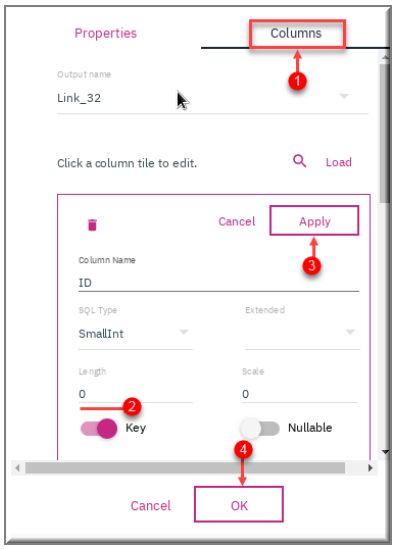
__108. Click `Apply` in the *column tile* and then click `OK`.

__109. Double-click the `Lookup` stage.

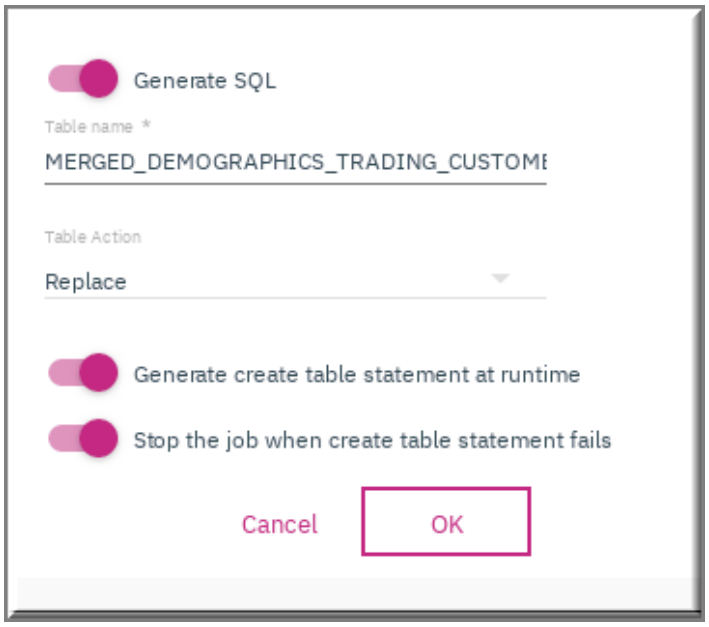
__110. On `Properties`, click key expression drop-down menu and select `ID` to join.



__111. Click `Columns` and click **ID field**, slide key slider to the right, click `Apply` and the `OK`.

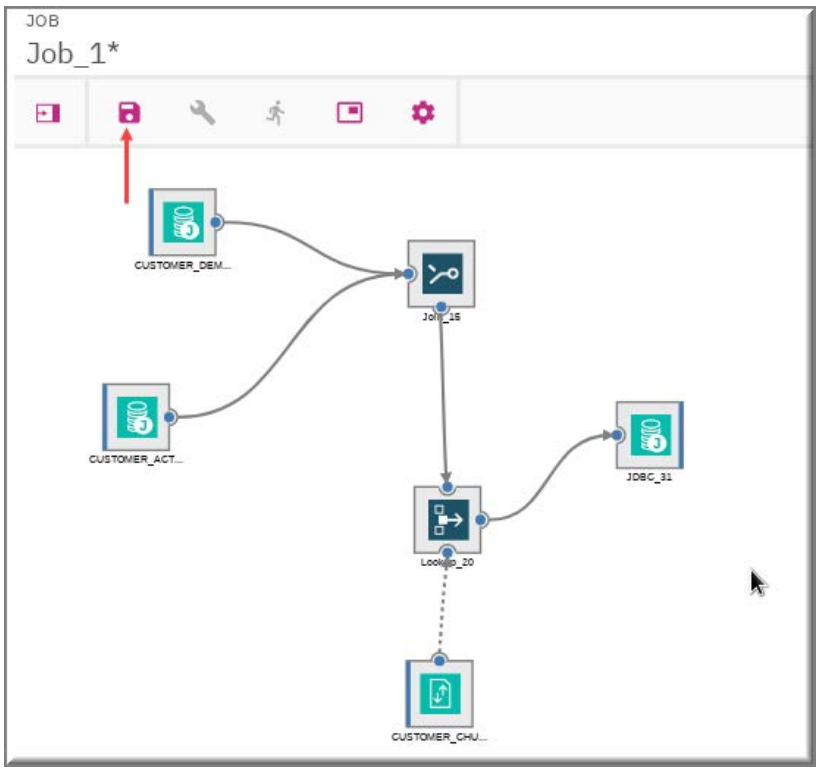


__112. Double-click the `Target` stage [Look for the name as `JDBC_XX`.]. Navigate down to the table name and type `MERGED_DEMOGRAPHICS_TRADING_CUSTOMER` and select table action `Replace` from the drop-down menu. Click `OK` to save.

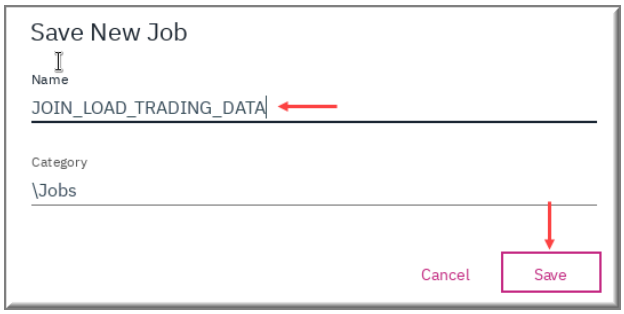


4.6.2 Save, Compile and Run Transformation Job

__113. Click `Save`.

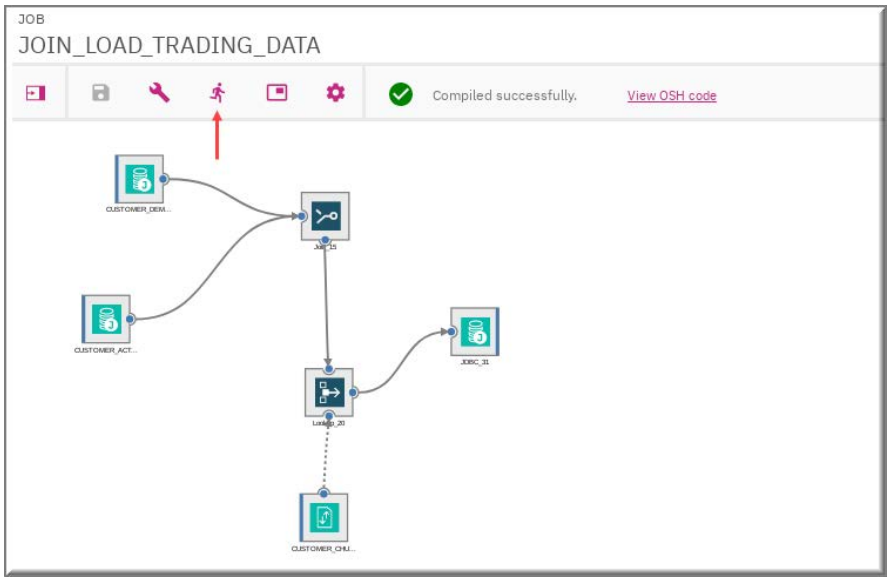


__114. Type name: JOIN_LOAD_TRADING_DATA and click Save.



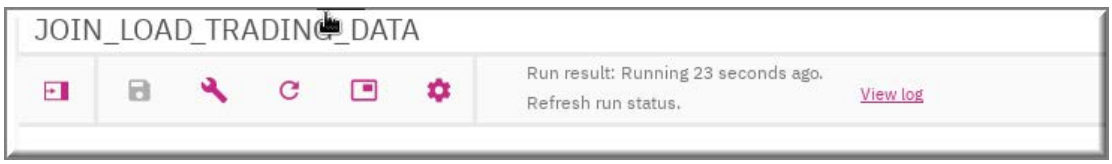
__115. Click the wrench to compile the job.

__116. Click the **runner** icon to Run the job.



__117. Click Run to confirm the run.

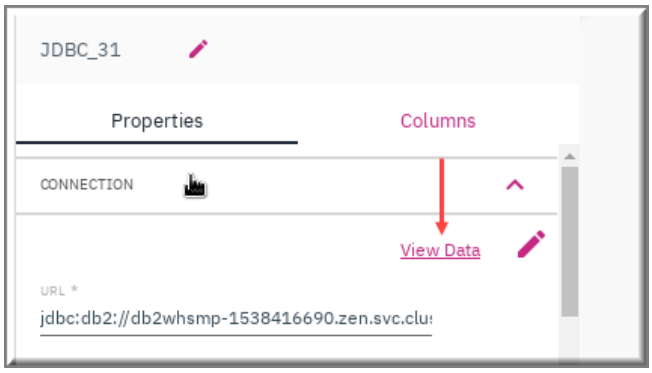
__118. It may take a minute or two to run the job. You can click the **refresh** icon to see the status.



__119. Click `Refresh`.

__120. Double-click `Target` stage [`JDBC_XX`] to open detailed pane on right side.

__121. Click `View Data`.



__122. The data transformation job between two Db2 Warehouse tables and data from Hadoop/HDFS was joined and created new table in Db2 Warehouse.

__123. We will use Analyze the data from this table in next lab exercise. We will build a business dashboard from [ML2]and then build machine learning models that we will use to modernize our application from analytics perspective.

Note: We have seen the value in applying a glossary, terms and classification to make data searchable so that data scientist, data engineers or business analyst can shop for data.

The ICP-D platform reduces the time-consuming data organization task, making it easier for data scientist. We will see that value in next lab exercise.

This lab has shown you how to automatically discover data sources, automatically classify those sources with business classifications using ML methods, automatically assign business terms to those data sources using ML and fuzzy matching methods.

It shows how to shop for specific data, and join and move that data to our analytics warehouse for further use by data scientists.

The steps covered here used to take months, and sometimes years, to complete using traditional manual methods.

ICP-D automates them so that you can accelerate the time to value of your analytics projects.

Credit for Lab: Jan Van Buren, Business Unit Exec, NA Info Integration and Governance Technical Sales

**** End of Lab 04: Organize**

Lab 05 Analyze

When embarking on machine learning projects, most organizations immediately engage their data scientists hoping to gain insight into their data. They quickly realize that the data that they need is not available in the format they require.

`Analyze` is our third pillar in our Integrated Data Platform. This is where data engineers (who prepare data), business analysts (who make sense out of data), data scientists (who build machine learning and artificial intelligence models) join forces to gain insights from their data.

The data stewards help `Organize` the data (as shown in the previous lab) through discovery, business glossary, policy, rules and governance. This is the foundation on which business analysts and data scientists build.

Now, let's look at the role – business analysts and data scientists would play in using IBM Cloud Private for Data to accomplish Analytics Modernization.

5.1 Business Analysist

We will assume the role of two personas who are key to driving business decisions. Let's start by examining the role that a business analyst plays in the discovery of business problems. We will seek solutions in later sections.

5.1.1 Dashboard – Identify Focus Area

__1. In our Executive Demo lab, you were presented with Stock Opening Bell business analysis that painted the picture of how the business is doing. You noticed a gradual decline in revenue and flat share volume.

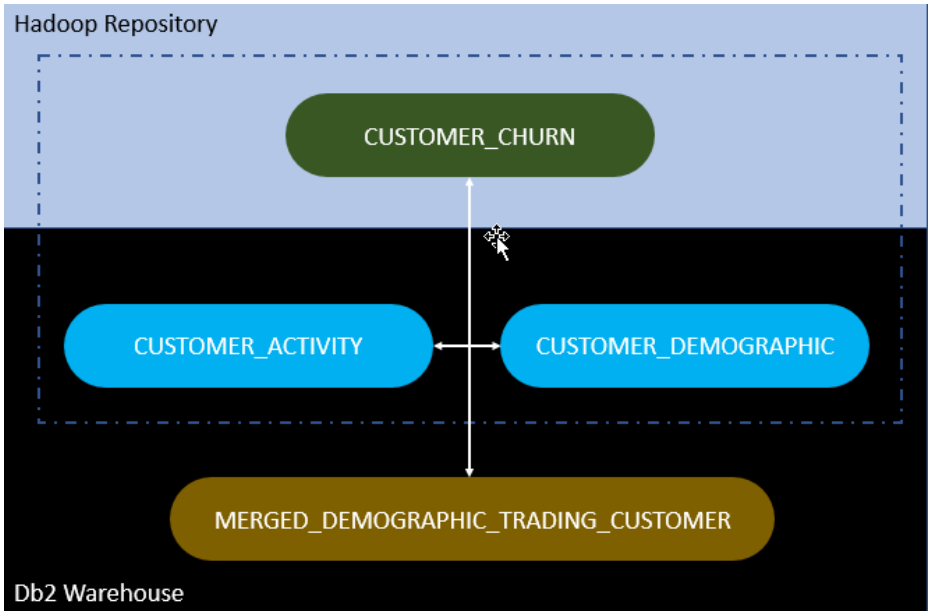
__2. Refer to Appendix-'A' if you would like to learn how to build this dashboard. You will find step-by-step directions to analyze current trends of customer visits and daily trade volume generated through the Stock Trader application.

__3. This is our Stock Trade Opening Bell dashboard derived from the data.

5.1.2 Data Preparation

__4. In previous lab, we collected Customer Risk data. In our fictitious scenario, the Customer Risk legacy application assigns a `Low`, `Medium` and `High` risks to customers based on the company's past complaint history. This data is stored in a Hadoop repository.

__5. In that lab, you'll recall that we joined `CUSTOMER_ACTIVITY` and `CUSTOMER_DEMOGRAPHIC` data from Db2 Warehouse with `CUSTOMER_CHURN` data from Hadoop and produced the `MERGED_DEMOGRAPHICS_TRADING_CUSTOMER` table.



__6. The table `MERGED_DEMOGRAPHIC_TRADING_CUSTOMER` has data that business analysts will visualize with the business dashboard. Then, the data scientists will apply ML/AI capabilities over the data.

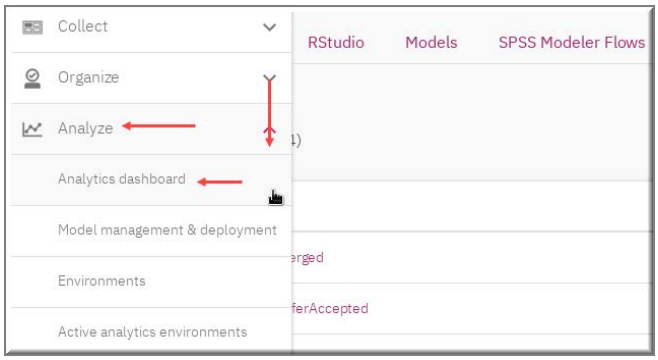
__7. Let's begin with business analysts role to analyze customer demographics.

5.1.3 Business Dashboard – Analyze Customer Demographics

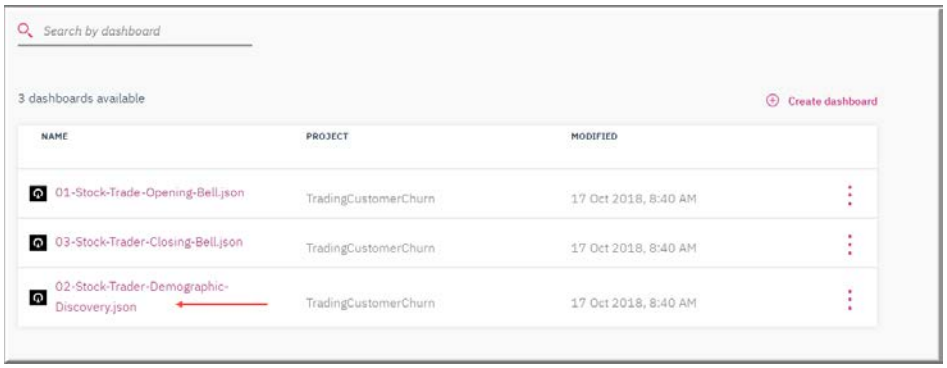
__8. Click `Projects` from the left menu bar.

__9. Click `TradingCustomerChurn` project.

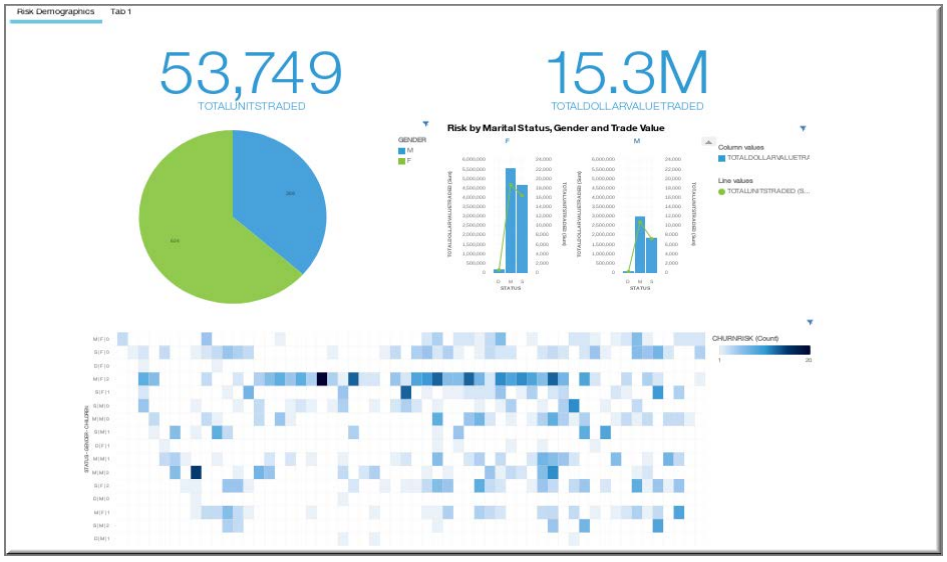
__10. Click `Analyze ⇨ Analytics Dashboard` from the left menu bar.



__11. We pre-built `02-Stock-Trader-Demographic-Discovery` dashboard.



__12. Click `02-Stock-Trader-Demographic-Discovery` to open the dashboard.



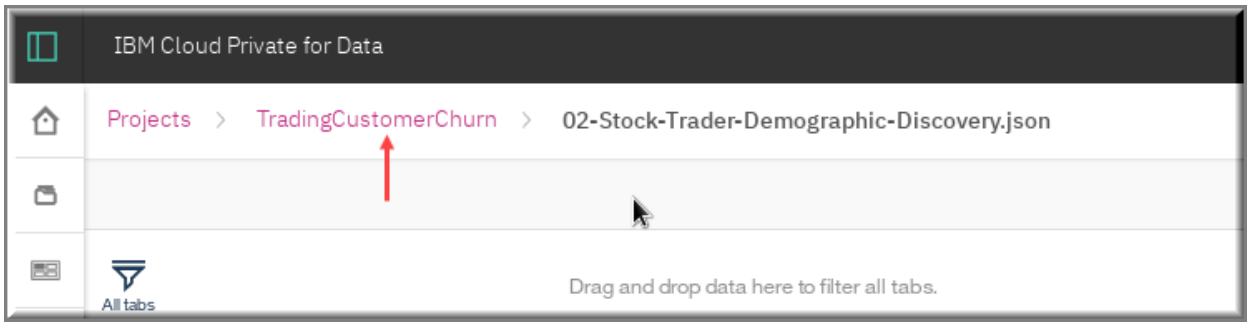
__13. Notice the metrics that are present in this dashboard. **1.** Know your customers (male or female), **2.** Separation Risk by marital status, gender and trade value are used to generate a heat map to help us understand which type of customers are more likely to separate from us.

__14. This dashboard will paint a high-level picture of the customer demographic we need to target to reverse our declining revenue trend.

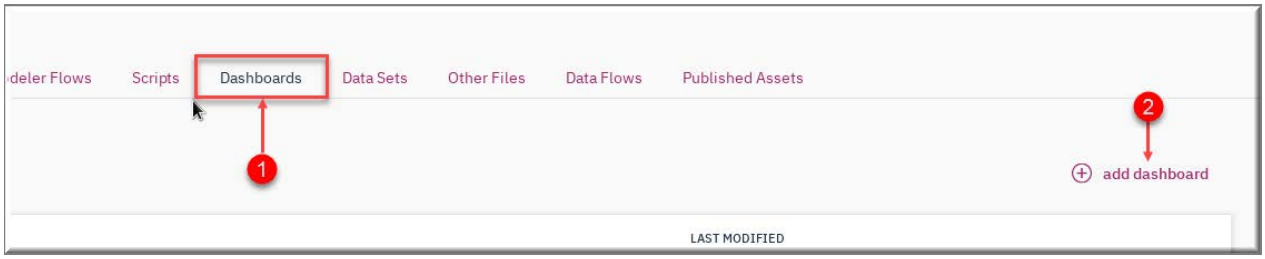
__15. The dashboards allow you to apply a filter `Low`, `Medium` and `High` to the Risk score to help you understand the data.

__16. Let's build this dashboard from the beginning.

__17. Click Project `TradingCustomerChurn`.

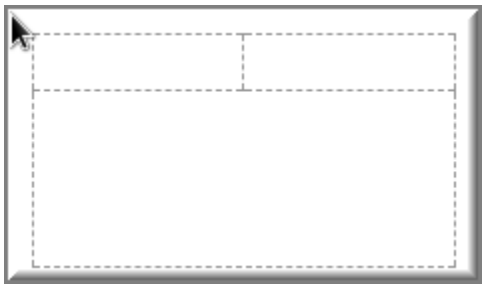


__18. Click `Assets,` then click `Dashboard` and then click `add dashboard`.

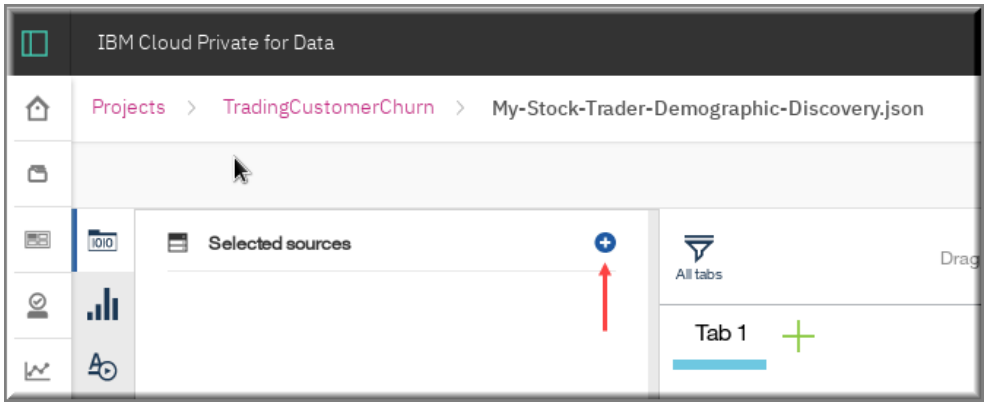


__19. Enter `My-Stock-Trader-Demographic-Discovery` in the Name field and click `Create`. [bottom right.]

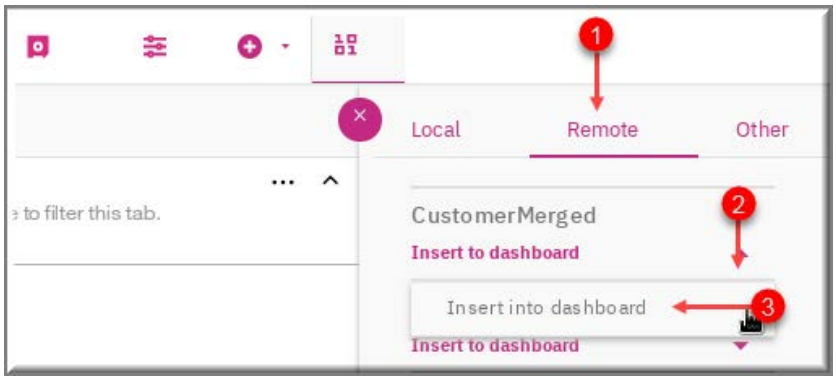
__20. Select this template and click `Ok`.



__21. Click the `+` next to `Selected Sources`

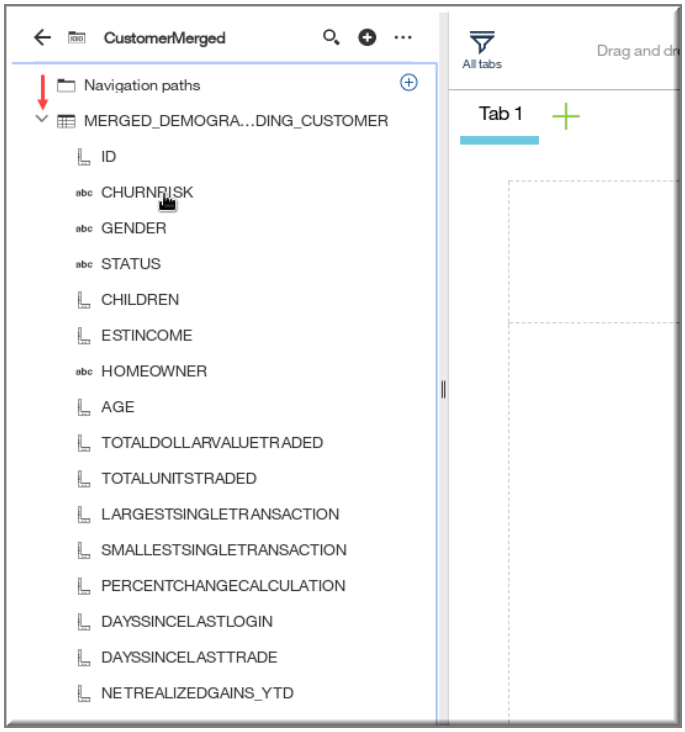


__22. Click `Remote` on the flyout menu on the right side. Expand `CustomerMerged` and click `Insert to dashboard`.

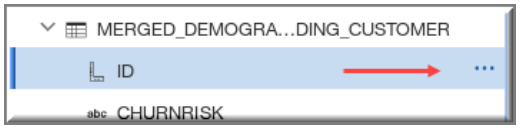


__23. This `CustomerMerged` Db2 warehouse table is inserted in the `Selected sources`. Click to select it.

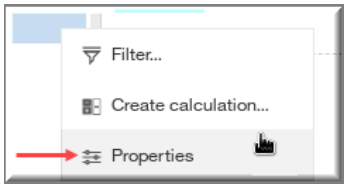
__24. Expand the table details by clicking the **down arrow** / caret.



__25. Click `ID` and then click ellipse (right side).



__26. Click `Properties`.



__27. Select `Identifier` from the *Usage* drop-down menu and `Count` from the *Aggregate* drop-down menu. Click `Close`.

__28. Repeat same for the following columns, setting their Usage and Aggregate values as follows:

Column Name	Usage	Aggregate
CHURNRISK	Attribute	None
GENDER	Attribute	None
STATUS	Attribute	None
CHILDREN	Attribute	None
ESTINCOME	Measure	None
HOMEOWNER	Attribute	None
AGE	Attribute	None
TOTALDOLLARVALUETRADED	Measure	Total
TOTALUNITSTRADED	Measure	Total
LARGESTSINGLETRANSACTION	Measure	None
SMALLESTSINGLETRANSACTION	Measure	None
PERCENTCHANGECALCULATION	Measure	None
DAYSSINCELASTLOGIN	Measure	None
DAYSSINCELASTTRADE	Measure	None
NETRALIZEDGAINS_YTD	Measure	None

__29. Drag `TOTALUNITSTRADED` towards the text `Drop here to manage` at top left of the screen. The icon color will change to light blue. Drop the field so that it occupies the full space.

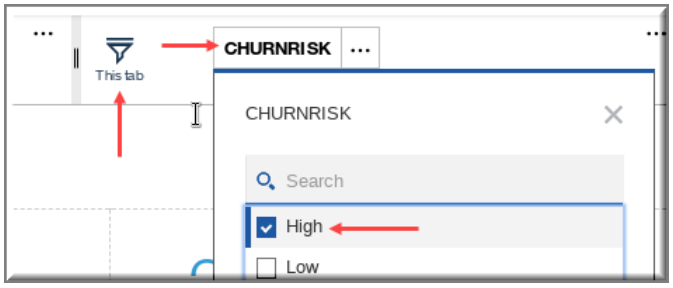
__30. The result should display the total units traded as follows:

__31. Similarly drag `TOTALDOLLARVALUETRADED` to the top right and drop it when the icon turns light blue as you hover the text *Drag here*.

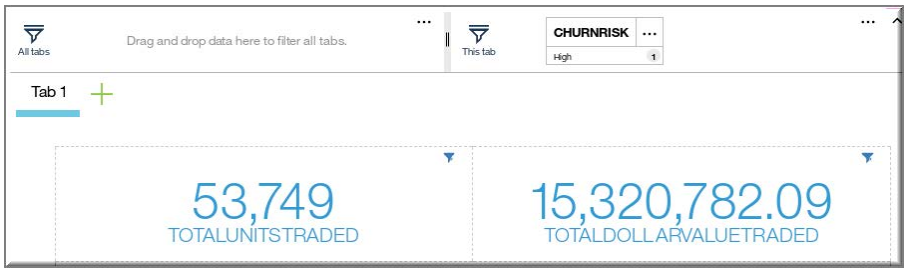
__32. The result should display the following.

__33. Now that we have scored a historical churn risk of existing customers, let's discover traits of who those customers are.

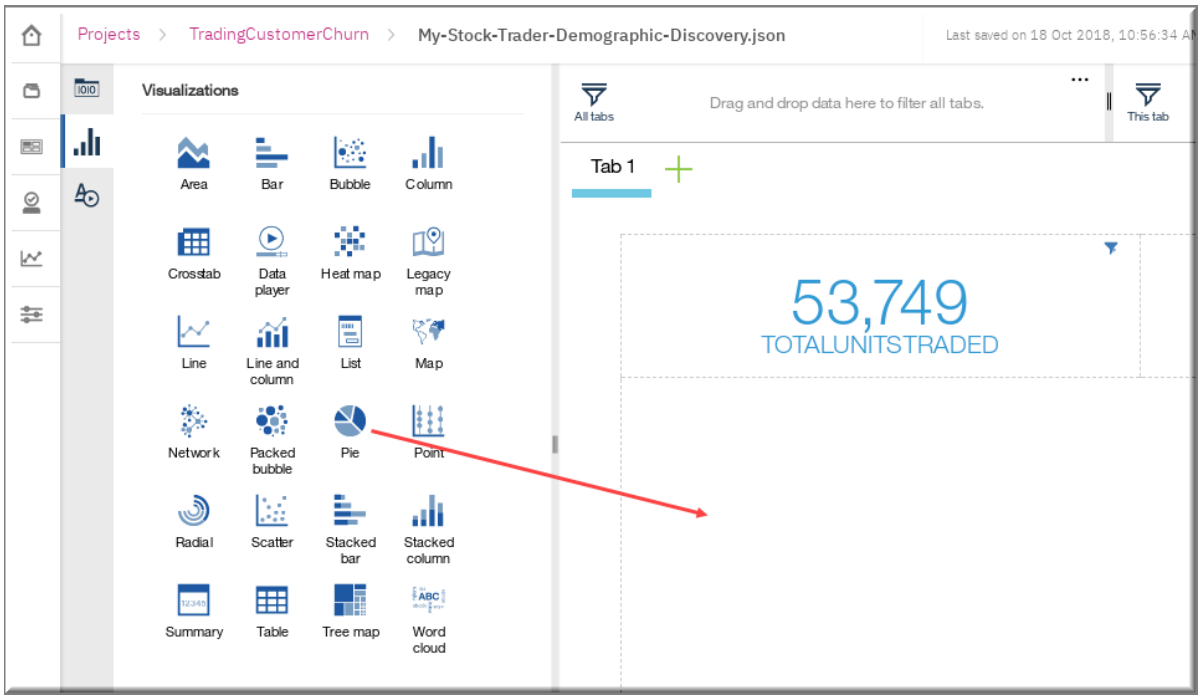
__34. Drag CHURNRISK and drop it on the **This Tab** text at the top right. Click CHURNRISK and check High. Click OK.



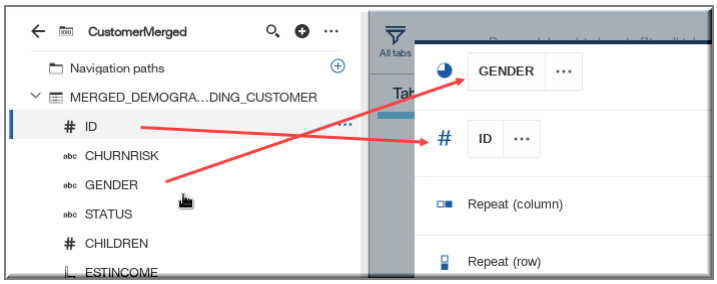
__35. Notice the amount changes after applying a filter of High risk customers.



__36. Click Middle Chart (left menu bar) icon and drag the **Pie chart** from the flyout menu to the bottom left.

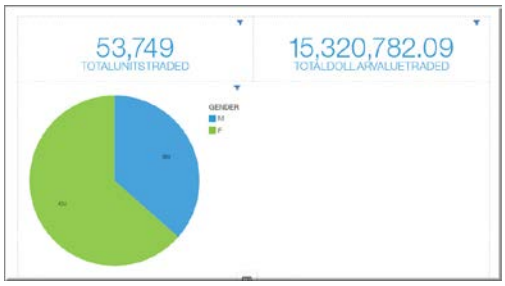


__37. Drag `GENDER` to the top `Segment` section (next to the pie icon) and drag **ID** to the `#` section.



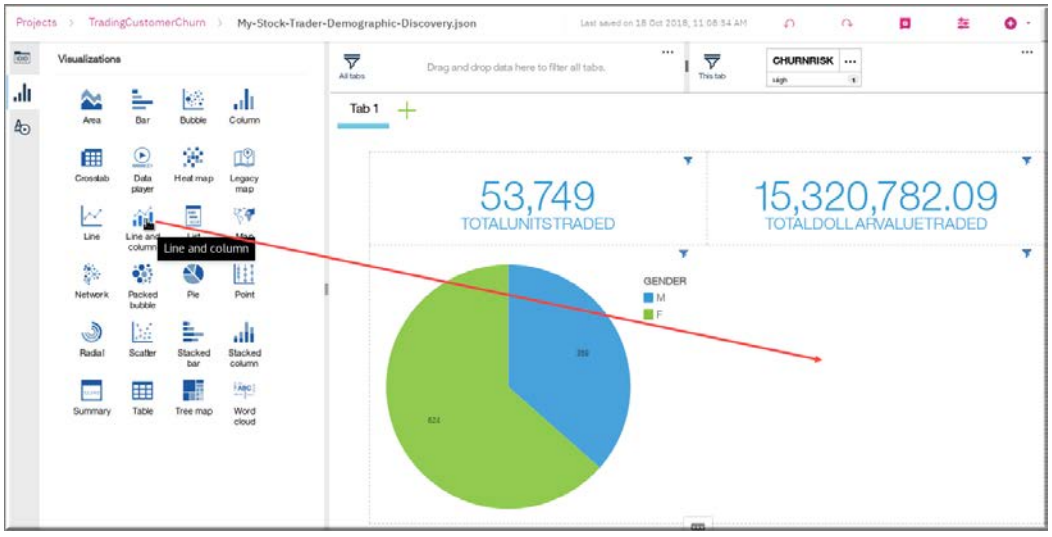
__38. Click `Collapse` (top right corner of the pie chart.)

__39. Your dashboard should display the following. Notice the relative proportion of female and male customers.

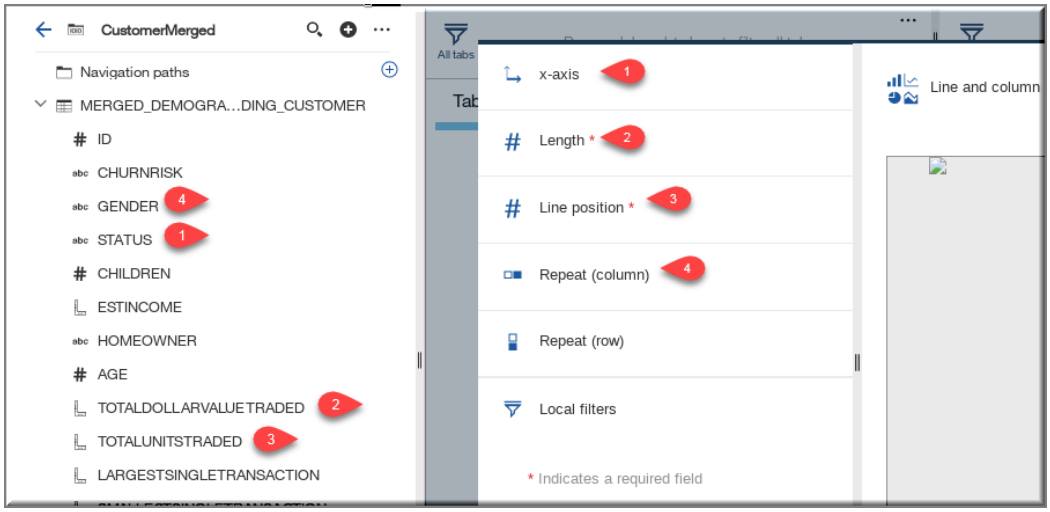


__40. Let's find out who makes the most trades for the most amount of trade value.

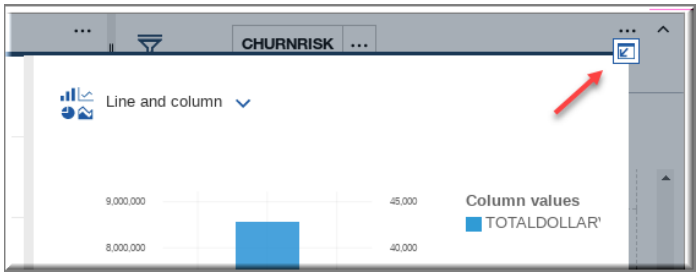
__41. From the `Chart` palette, drag **Line and Column** to bottom right of the canvas.



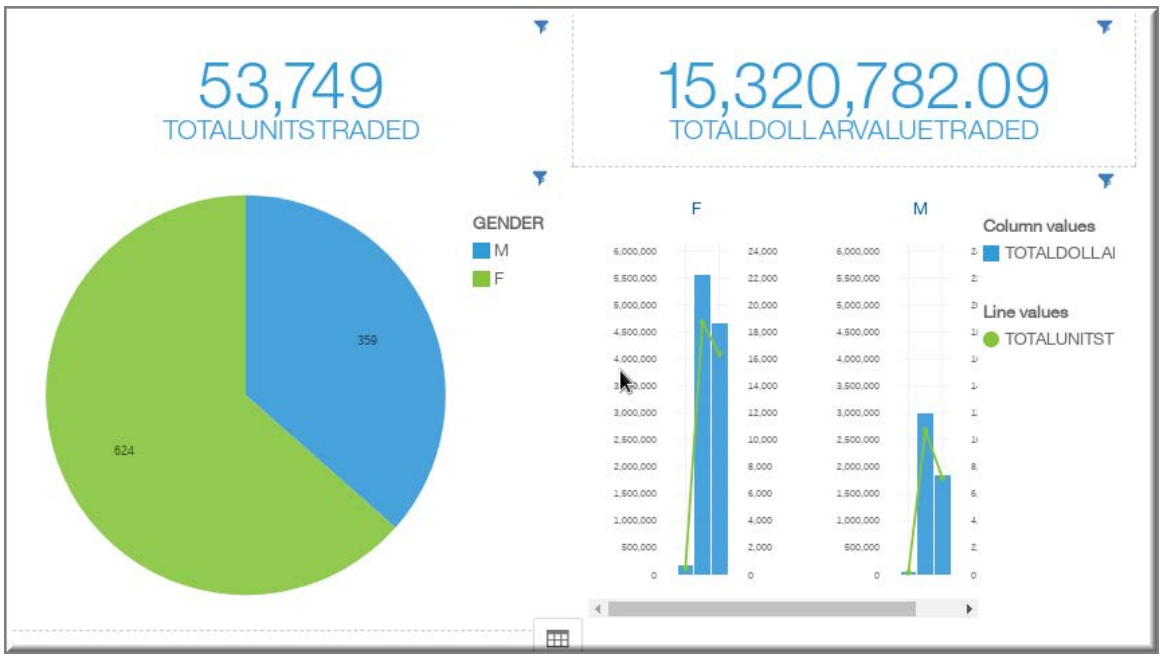
__42. Drag each of the numbered items to the right to the position indicated (that is, Drag status to the top, then Length and so on.)



__43. Collapse the chart.

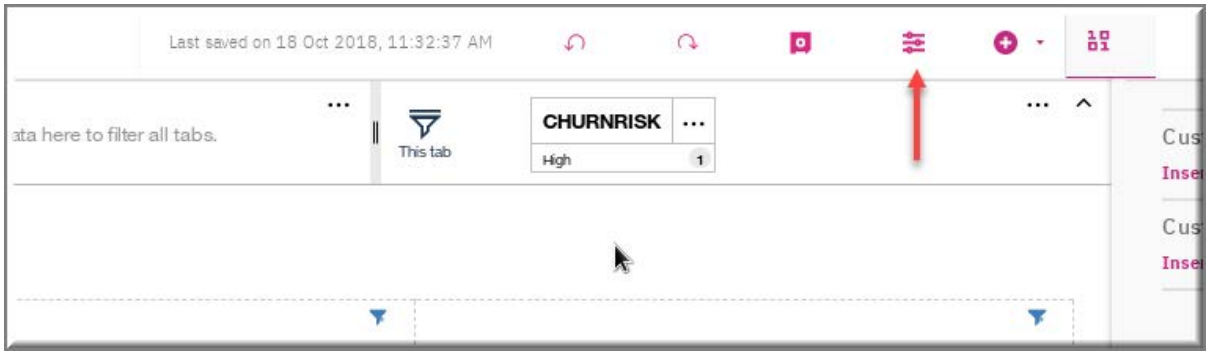


__44. The dashboard should display the following figure:

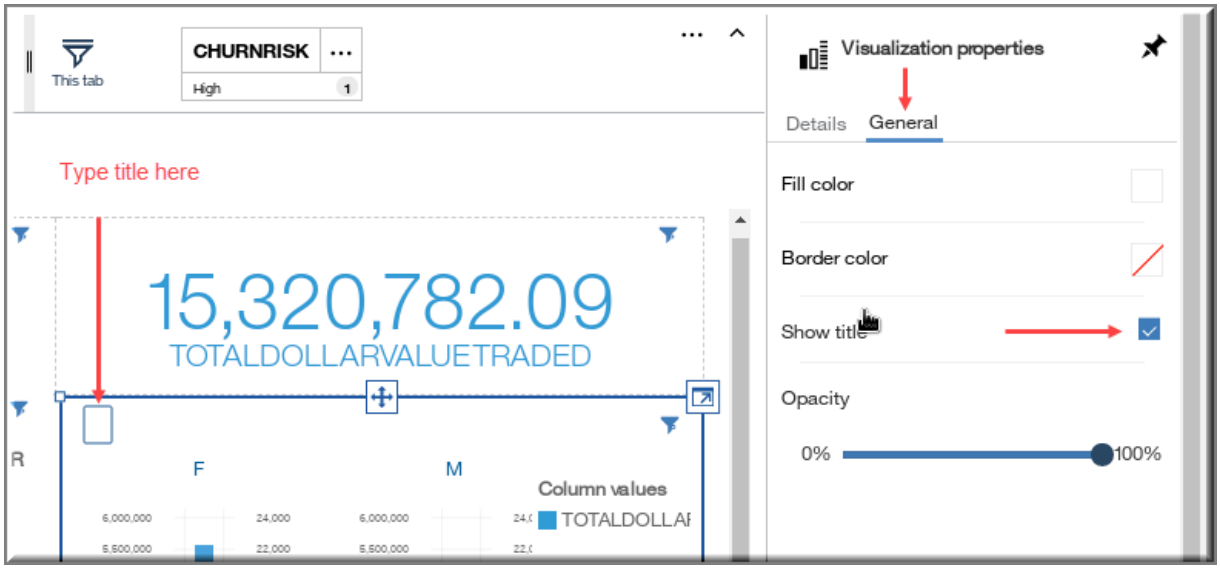


__45. Make some room available by hiding the right palette by clicking `Find Data` icon.

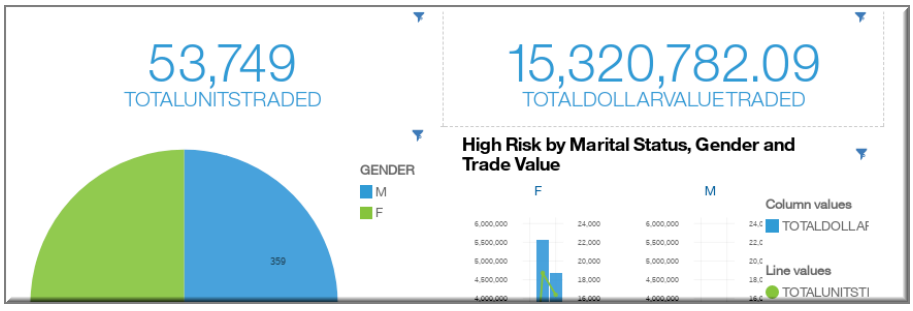
__46. Select the line column chart that you just added and click `Properties` from top menu bar.



__47. Click the `General` tab. Check `Show Title` and type: `High Risk by Marital Status, Gender and Trade Value`



__48. Your dashboard is beginning to take shape. It should display as follows:

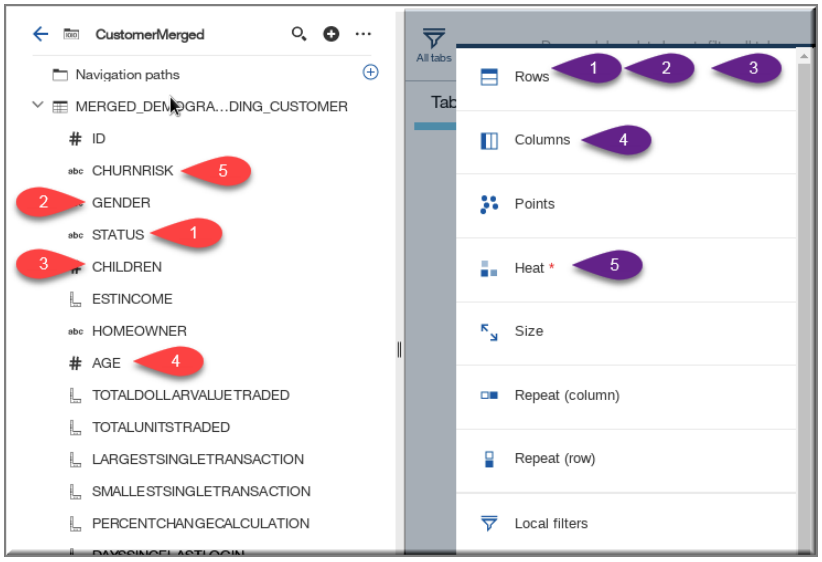


__49. It appears from the chart that the greatest number of high-risk customers who trade and spend the most with us are Married and Single Females.

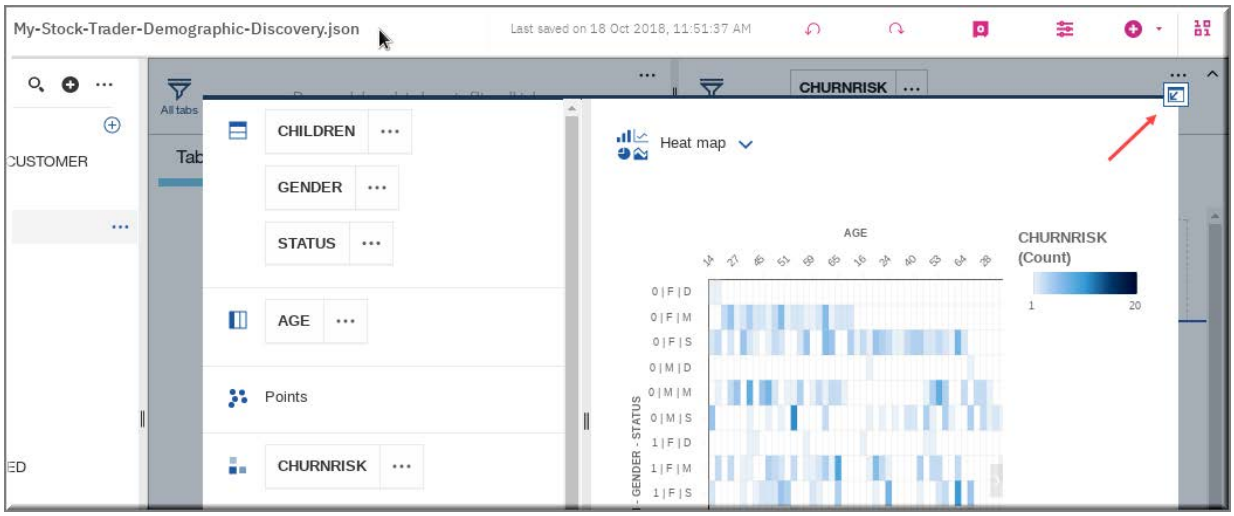
__50. Is there any other insight we can gain? For example, does the number of children or home ownership plays a part?

__51. From the `Chart` palette, `drag Heat Map` to the botton of the pie chart.

__52. Drag the `STATUS`, `GENDER,` and `CHILDREN` individually to the `Rows` section. Drag `AGE` to the `Column` section and drag `CHURNRISK` to the `Heat`. [As indicated in the number tags below]



__53. Collapse the `Heat Map` Chart.



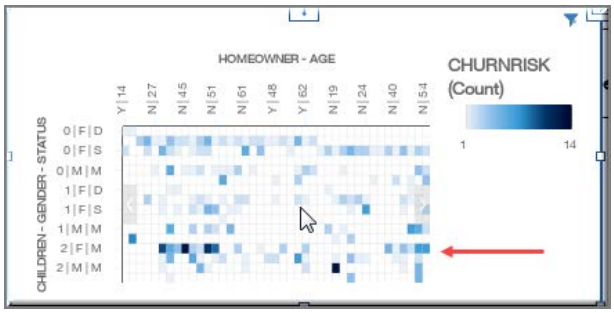
__54. The resultant Heap Map Chart shows that married couple with two children account for most of our customers.

__55. Does Home Ownership play a part?

__56. Expand the `Heat Map` chart and drag `HOMEOWNER` above `AGE` (Note: Do not replace Age).

__57. Collapse the chart.

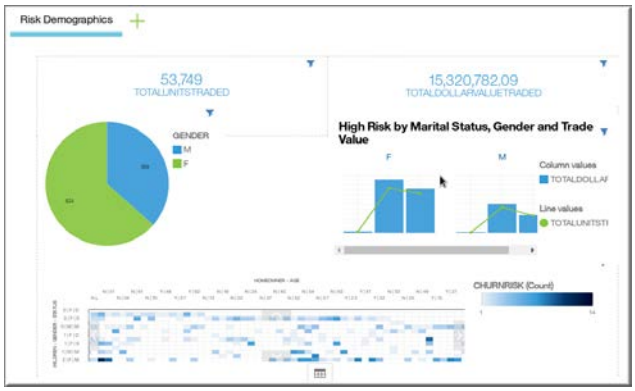
__58. The `Heat Map` visual shows that single female homeowners over 40 years old with two children are most likely to be high risk.



	Note: The key finding – Single females aged 40 and over, who own a home and have two children are most at risk to stop buying. Let's focus on these customers.

__59. Change the title of the first tab to `Risk Demographics`.

__60. Adjust the different visuals in the first tab. For example: Place the heat map below the pie and column charts.

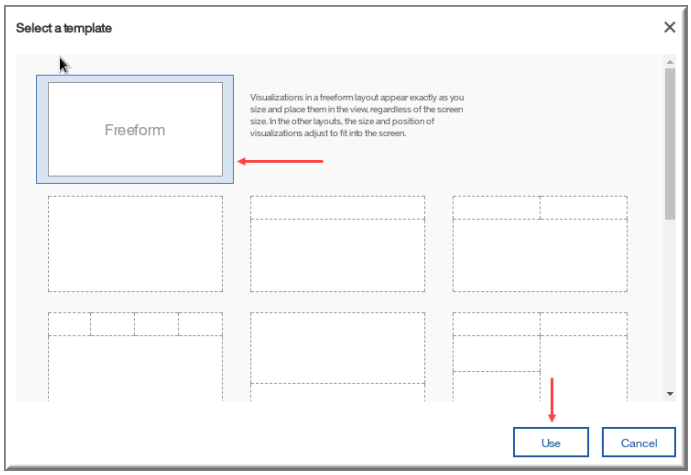


5.1.4 Analyze Risk to the Business

__61. Let's create another tab to see the risk to the business.

__62. Click + to add a tab.

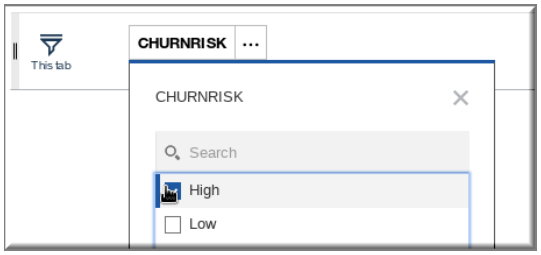
__63. Select the **Free Form** template and click Use.



__64. Change the title to **Risk to the Business**. [Click the pencil in the tab.]

__65. Let's filter our results so that we focus on this group.

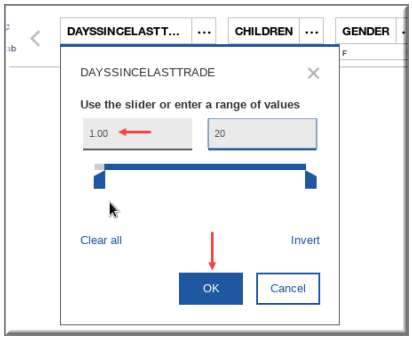
__66. Drag column CHURNRISK to the *Drag and Drop Data to Filter THIS tab*. [Top right of the canvas.] Click it and check High and click OK.



__67. Drag GENDER and click after you drop in the filter area and check Female. Click OK.

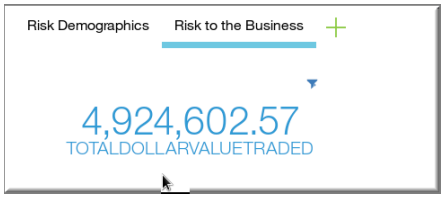
__68. Drag CHILDREN and click after you drop in the filter area and select 2 and click OK.

__69. Drag DAYSSINCELASTRADE and click after you drop in the filter area and select 2 for the lower bound and keep the upper bound to 20. Click OK.



__70. Now we have applied filters as per our previous analysis, let's build the dashboard.

__71. Drag TOTALDOLLARVALUETRADED to the top left area of the canvas. Resize the chart so it appears in the top left corner. This creates a single value field that is filtered based upon our tab filter.



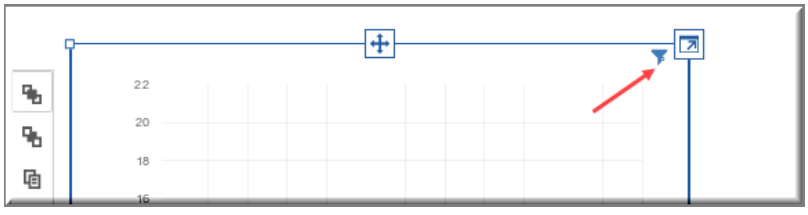
__72. It looks like this group trades roughly $5M.

__73. Pull over TOTALUNITSTRADED over to the top right area of the canvas. Resize the chart so that it appears in the top right corner.

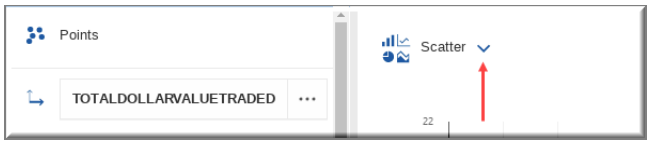
__74. Let's see the TOTALDOLLARVALUE at risk by this group based upon the days since they last traded with us and the date they last logged in to our system.

__75. Hold control key down and click DAYSSINCELASTTRADE and TOTALDOLLARVALUETRADED and drag to the right.

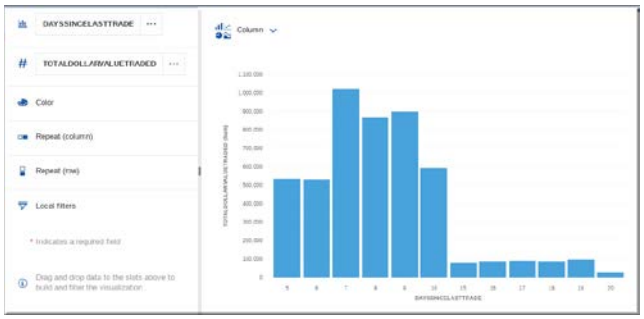
__76. Select Expand.



__77. Change the chart type to Column. [Click and select from the chart palette.]



__78. The chart gets drawn as shown below:



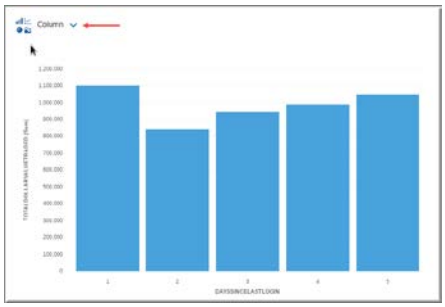
__79. Collapse the screen by clicking the collapse icon to the top right of the frame.

__80. Resize this chart and fit into the bottom left of the canvas using the chart movement arrows shown on the border of the chart.

__81. Hold control key down and click `DAYSSINCEDLASTLOGIN` and `TOTALDOLLARVALUETRADED` and drag to the right.

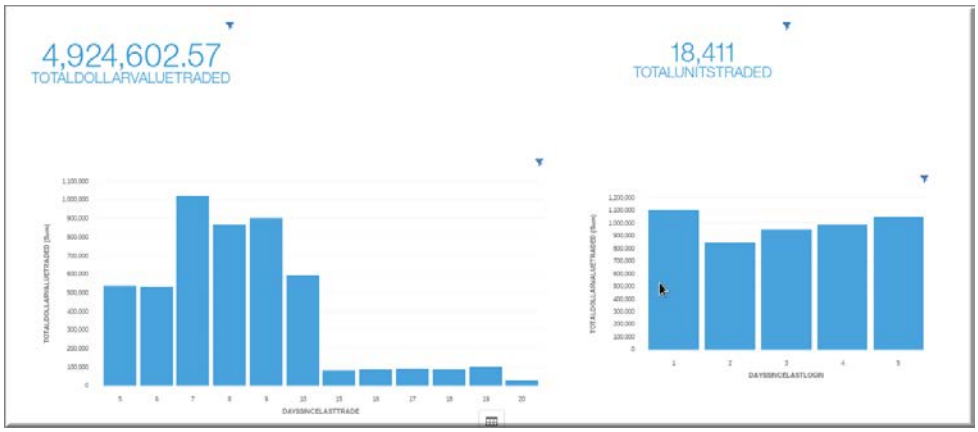
__82. Expand the chart.

__83. Change the Chart type to `Column.` It should display the following:

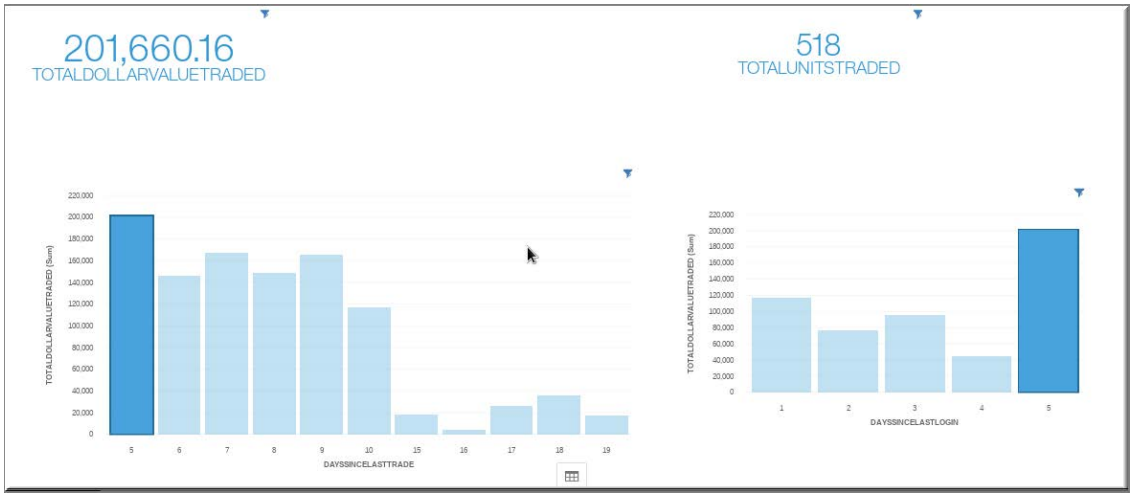


__84. Collapse the chart by clicking the top right icon on the chart and use resize arrows to fit in the bottom right of the screen.

__85. The second tab will display as shown below.



__86. You can click any column from each chart to show the aggregate value for total dollar traded and total units traded. For example: Click first column on first chart and last column on second chart to see the values.



	Note: As a business analyst, you identified the group likely to be at risk to separate from the business. An intelligent scientific solution needs to be crafted to mitigate the risks. Data scientists are now engaged to move our analysis forward.

Credit for the Lab: Kent Rubin, Analytics and Cognitive Solution Architect

5.2 Data Scientists – Approach to a Solution

As seen in the previous labs, the business analysts have visualized the data so that it can be understood. The search of a solution begins now by engaging data scientists from the organization.

We have uncovered few key points along the way.

- Data resides everywhere and collecting the data in one lake (or platform) is not realistic.
- Data virtualization within ICP-D expands the capability to collect the data.
- Data governance is key to comply with rules and regulations but is difficult to implement, which hinders discovery and use of data.
- Dark Data is what leads many organizations to not take an action when it was necessary.
- Data preparation consumes too much of the data scientists' time today. They often spend more time preparing data than developing models.

We have gone through the journey of automation, machine learning while discovering and cataloging of data, it is now time to allow data scientists to provide solutions.

__87. Let's begin through the next phase of Analytics Modernization.

5.2.1 Analyzing Data and Toolset

__88. The recent contribution of universities, research institutes, corporations and individuals to Open Source has led to the development of innovative tools in data science. Most notable of them are:

- Jupyter and Spark Notebooks
- Zeppelin and Spark Notebooks
- R-Studio

IBM Cloud Private for Data supports all three.

__89. Python has become a language a choice for many data scientists mainly due to a very extensive ecosystem of libraries built for the Python language platform. It is no surprise that today Python commands 35 percent of the market share amongst all languages.

__90. We will use Jupyter notebooks and Spark for this lab exercise.

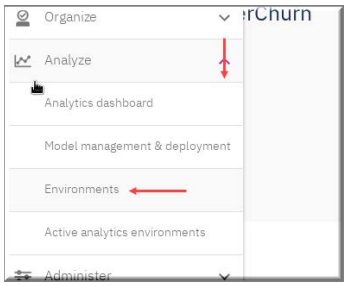
__91. ICP-D platform provides sample notebooks that show real-world scenarios and contain many useful examples of computations and visualizations that will jumpstart your analytics projects.

	**Note:** The non-obvious value of the IBM Cloud Platform is to provide an agile "cloud-native" infrastructure within your firewall.

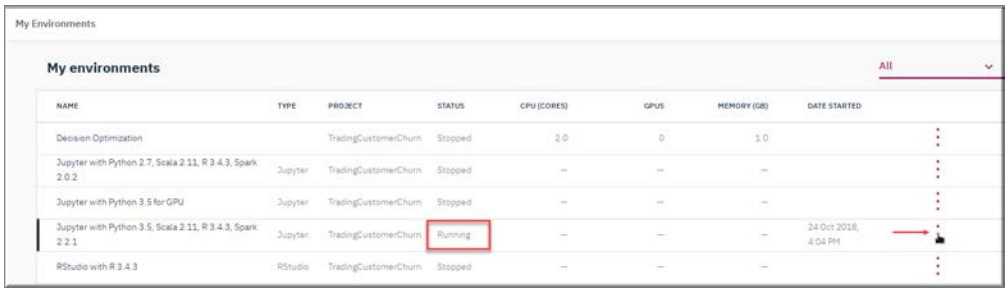
5.2.2 Walk Through the Model

__92. Developing, training, and deploying a machine learning lab from scratch is beyond the scope of this introduction to ICP for Data, so we will walk through a Jupyter notebook that represents an example of the model development process. If you are familiar with the process you will recognize the tools used and will learn how ICP for Data speeds and facilitates collaboration throughout the process. If you're new to machine learning it will serve as a gentle introduction to the process. We will develop, test and score a model that addresses our business problem.

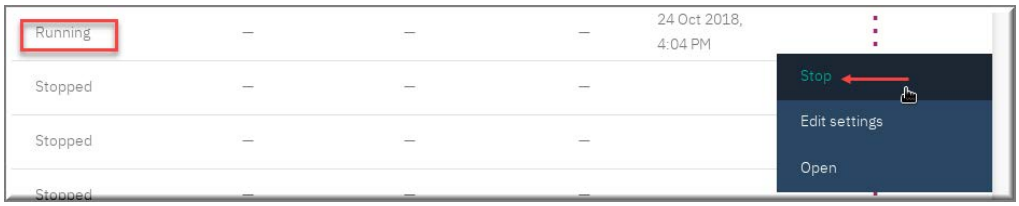
__93. Switch back to the ICP-D Web UI and click `Analyze ⇨ Environments`.



__94. We need to stop <u>all</u> `Running` environments to free-up resources. Click **three vertical dots** to the right of the `Running` environment.

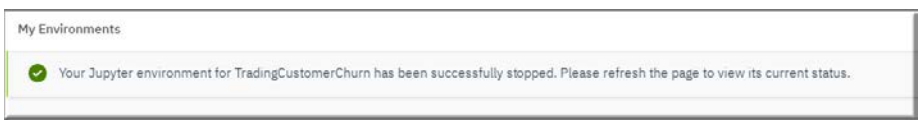


__95. Click `Stop`.



__96. Repeat same for any other environment, which is `Running` in your case.

__97. You will see the message of the environment getting stopped.



__98. Click `Projects` from the left menu bar.

__99. Click `TradingCustomerChurn` to open it.

__100. Click `Assets`. [Your assets number might be different than shown below].

__101. Click Jupyter Notebook – `01TradingCustomerChurnClassifierSparkML` to open it.

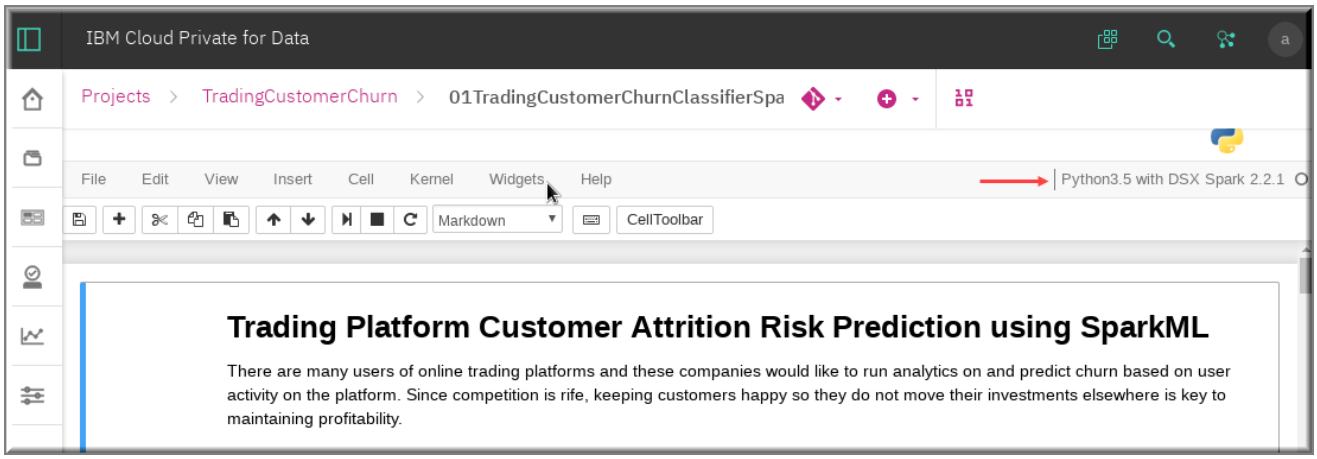
	Note: The Jupyter notebook process initiates a request to Kubernetes to launch a pod that will run on any machine where resources are available. The launch may take a few minutes as it takes time to load the 7 GB image from the registry and launch it.

__102. If you see the free resources message, please ignore it. It is normal due to the restricted resources in the lab environment.

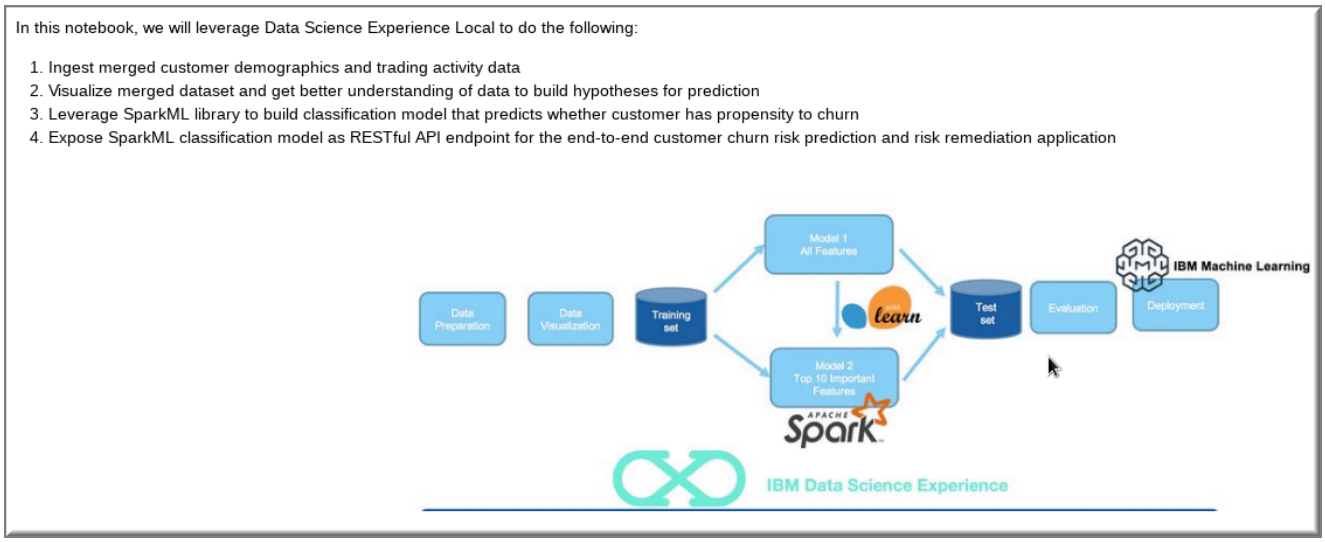
__103. If you are an experienced user of Jupyter notebooks, you may skip steps that are review for you.

__104. A notebook a collection of compute cells that can be run individually, or as a group. You can pause execution between steps to review results, make adjustments to your code and re-run. One of the features of notebooks is that your notes and markdown cells can be interspersed with your code. This allows notebooks to be self-documenting.

__105. A notebook can use environment of your choice. For example, we chose to use Python 3.5 with Spark 2.2.1. Other choices will be available, depending on your cluster configuration and hardware. For example, other levels of Python, Spark, R, GPUs, might be available. These options are chosen at the time of notebook creation.

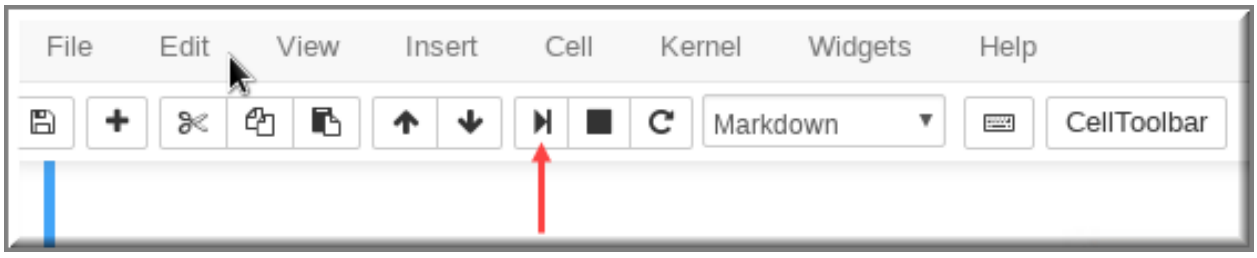


__106. Read through the introduction to learn what this notebook provides.



__107. The notebook explains two type of cells – markdown and code. The documentation is done using markdown (easy to learn) and code cells are actual code that will execute when we want them to.

__108. To execute a cell (even including markdown cell), you can either press menu icon `Run cell select below` or use keyboard `Shift-Enter` to execute a cell.



__109. You are on a first cell, press `Shift-Enter` to execute the cell.

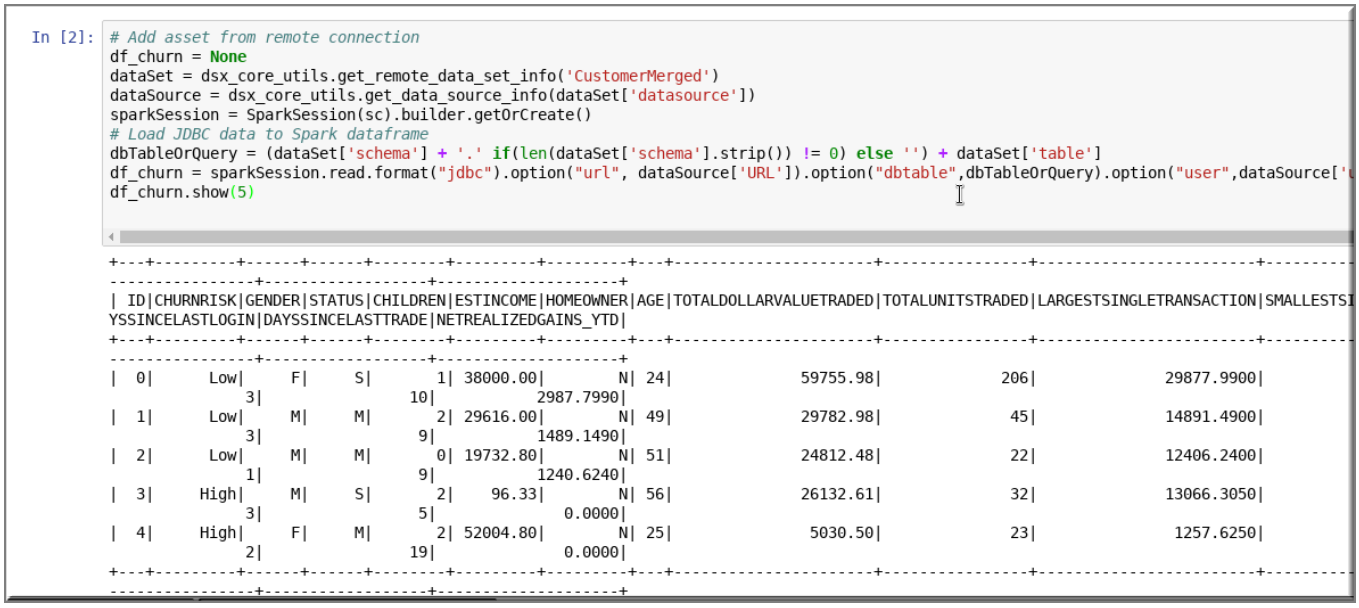
__110. When you execute a cell, you will see an * in the cell square bracket. This is an indication that the cell is currently running. You can execute cell one at a time, in which case execution pauses at the next cell. You can also "Run All" cells which causes a serial execution from cell to cell without pause.

__111. When a cell completes its execution, you will see a number displayed in its cell boundary as shown.

__112. The next cell is a document cell that explains the procedure to insert a data frame which can be either a Pandas or Spark data frame.

__113. Press `Shift-Enter` to execute a cell and watch its output and then go to the next cell by pressing `Shift-Enter` again and keep repeating it until we reach to the end of this notebook.

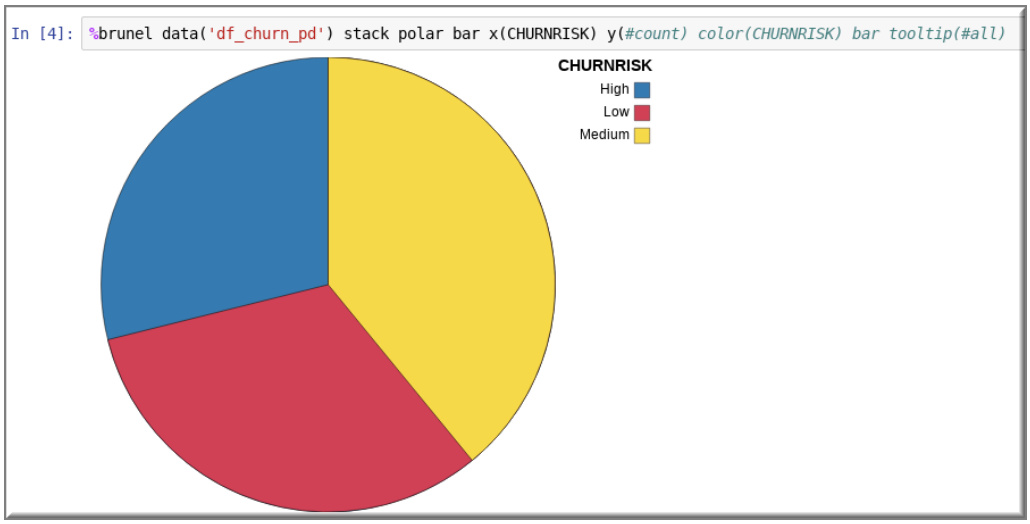
__114. The next data cell is the inserted Panda data frame for connecting to Db2 Warehouse.



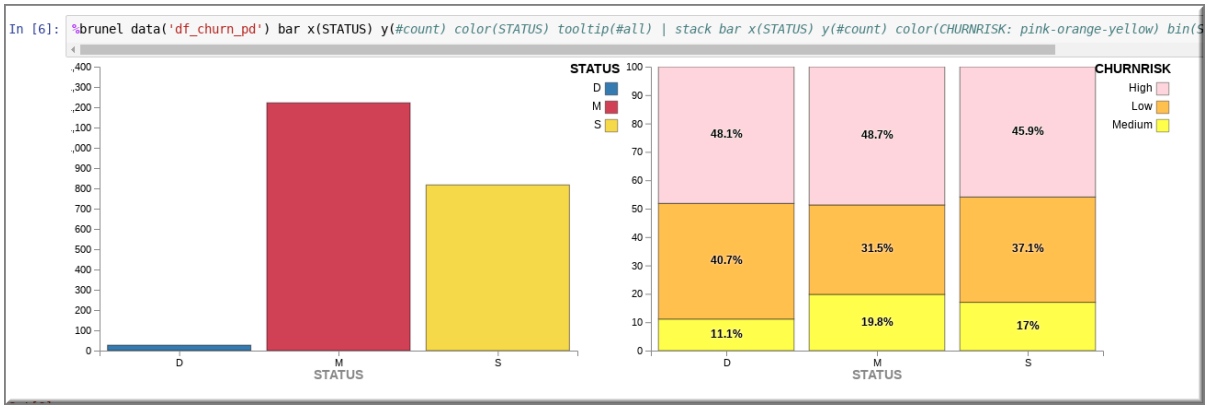
__115. The data above shows the output from `MERGED_DEMOGRAPHICS_TRADING_CUSTOMER` table which was the result of the previous lab exercise and the focus of the business analysis that we did in `Business Analyst` section.

	**Note:** You might wonder – do I have to write this much code to connect to my data source? The answer is No – you just define a data source and drag it here from your remote data set and the code gets generated automatically.

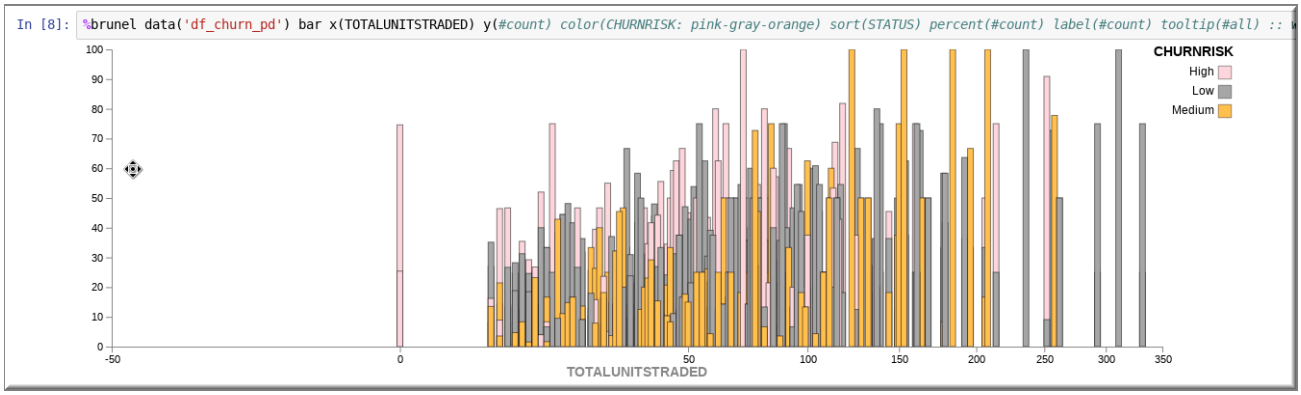
__116. The next cell contains code to visualize the data using the Brunel library. Brunel is just one of many open source visualization libraries available for Python and Jupyter. You may also have heard of matplotlib, plotly, Pixiedust and others. New libraries are constantly being released.



__117. The next visualization shows the separation risk of different category and status.



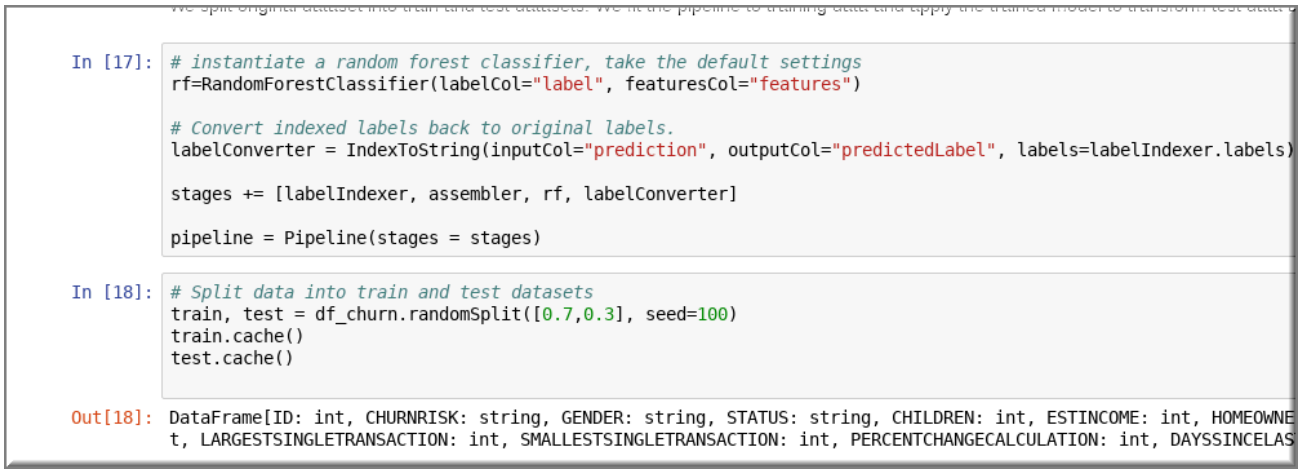
__118. Next visualization is total units traded by each separation risk category.



__119. The next stage in model development is data preparation, which involves the indexing of categories and assembling them into a feature vector to train models.

5.2.3 Build Spark ML Model using Random Forest Classification

__120. These steps show the definition of a pipeline chains together transformations and estimators for machine learning algorithms.

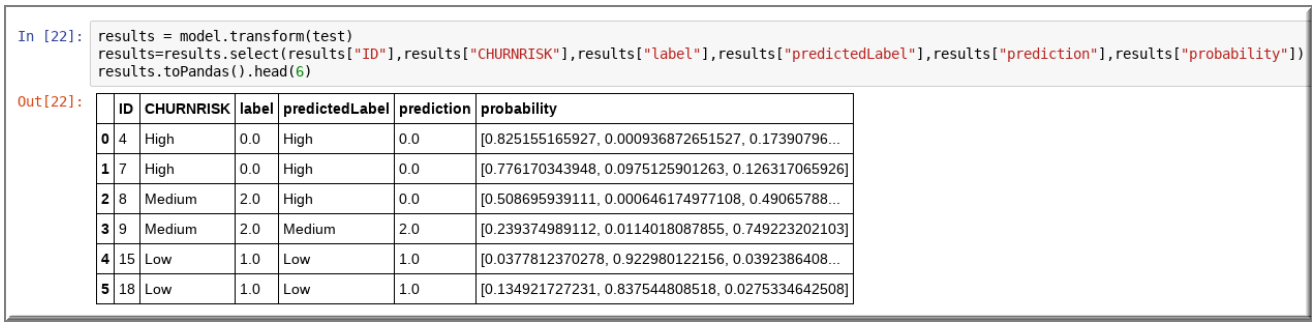


In [17]:
```
# instantiate a random forest classifier, take the default settings
rf=RandomForestClassifier(labelCol="label", featuresCol="features")

# Convert indexed labels back to original labels.
labelConverter = IndexToString(inputCol="prediction", outputCol="predictedLabel", labels=labelIndexer.labels)

stages += [labelIndexer, assembler, rf, labelConverter]

pipeline = Pipeline(stages = stages)
```

In [18]:
```
# Split data into train and test datasets
train, test = df_churn.randomSplit([0.7,0.3], seed=100)
train.cache()
test.cache()
```

Out[18]:
```
DataFrame[ID: int, CHURNRISK: string, GENDER: string, STATUS: string, CHILDREN: int, ESTINCOME: int, HOMEOWNE
t, LARGESTSINGLETRANSACTION: int, SMALLESTSINGLETRANSACTION: int, PERCENTCHANGECALCULATION: int, DAYSSINCELAS
```

__121. The next process builds the models using the pipeline, transforms and shows the results.



In [22]:
```
results = model.transform(test)
results=results.select(results["ID"],results["CHURNRISK"],results["label"],results["predictedLabel"],results["prediction"],results["probability"])
results.toPandas().head(6)
```

Out[22]:

	ID	CHURNRISK	label	predictedLabel	prediction	probability
0	4	High	0.0	High	0.0	[0.825155165927, 0.000936872651527, 0.17390796...
1	7	High	0.0	High	0.0	[0.776170343948, 0.0975125901263, 0.126317065926]
2	8	Medium	2.0	High	0.0	[0.508695939111, 0.000646174977108, 0.49065788...
3	9	Medium	2.0	Medium	2.0	[0.239374989112, 0.0114018087855, 0.749223202103]
4	15	Low	1.0	Low	1.0	[0.0377812370278, 0.922980122156, 0.0392386408...
5	18	Low	1.0	Low	1.0	[0.134921727231, 0.837544808518, 0.0275334642508]

__122. The model results and interpretation are described in the documentation cell.

Model results

In a supervised classification problem such as churn risk classification, we have a true output and a model-generated predicted output for each data point. For this reason, the results for each data point can be assigned to one of four categories:

1. True Positive (TP) - label is positive and prediction is also positive
2. True Negative (TN) - label is negative and prediction is also negative
3. False Positive (FP) - label is negative but prediction is positive
4. False Negative (FN) - label is positive but prediction is negative

These four numbers are the building blocks for most classifier evaluation metrics. A fundamental point when considering classifier evaluation is that pure accuracy (i.e. was the prediction correct or incorrect) is not generally a good metric. The reason for this is because a dataset may be highly unbalanced. For example, if a model is designed to predict fraud from a dataset where 95% of the data points are not fraud and 5% of the data points are fraud, then a naive classifier that predicts not fraud, regardless of input, will be 95% accurate. For this reason, metrics like precision and recall are typically used because they take into account the type of error. In most applications there is some desired balance between precision and recall, which can be captured by combining the two into a single metric, called the F-measure.

__123. The model precision for this is 91 percent.

```
In [24]: print('Model Precision = {:.2f}.'.format(results.filter(results.label == results.p
         Model Precision = 0.91.
```

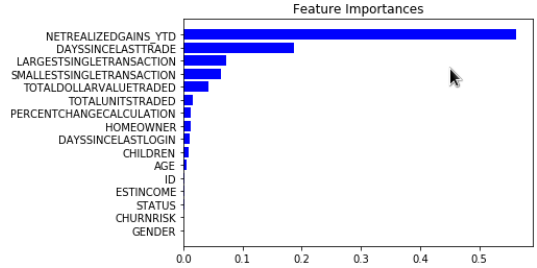
5.2.4 Evaluate Different Models

__124. The process of building model is incomplete unless it can be evaluated with a test data to see its accuracy.

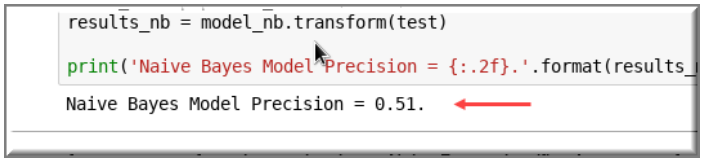
```
Overall Statistics
Model F-measure = 0.9096774193548387

Statistics by Class
Class 0.0 F-Measure = 0.9177631578947368
Class 1.0 F-Measure = 0.9864253393665159
Class 2.0 F-Measure = 0.7052631578947368
```

__125. The next cell shows the “Feature Importance” which shows how much of an impact each feature has on prediction.

```
Out[29]: Text(0.5, 0, 'Relative Importance')
```



__126. Next is Naïve Bayes classifier trained on the training data set.



__127. The above indicates that Random Forest Classifier shows a high F-measure upon evaluation, which shows strong performance.

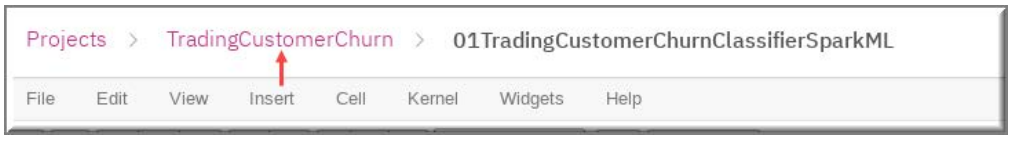
5.2.5 Save the Model in Repository

__128. After comparison of two models, the Random Forest model is saved into the repository.

	Note: The model is saved in the platform repository, providing governance to the data and model.

5.2.6 Test, Batch Score and Evaluate Saved Model

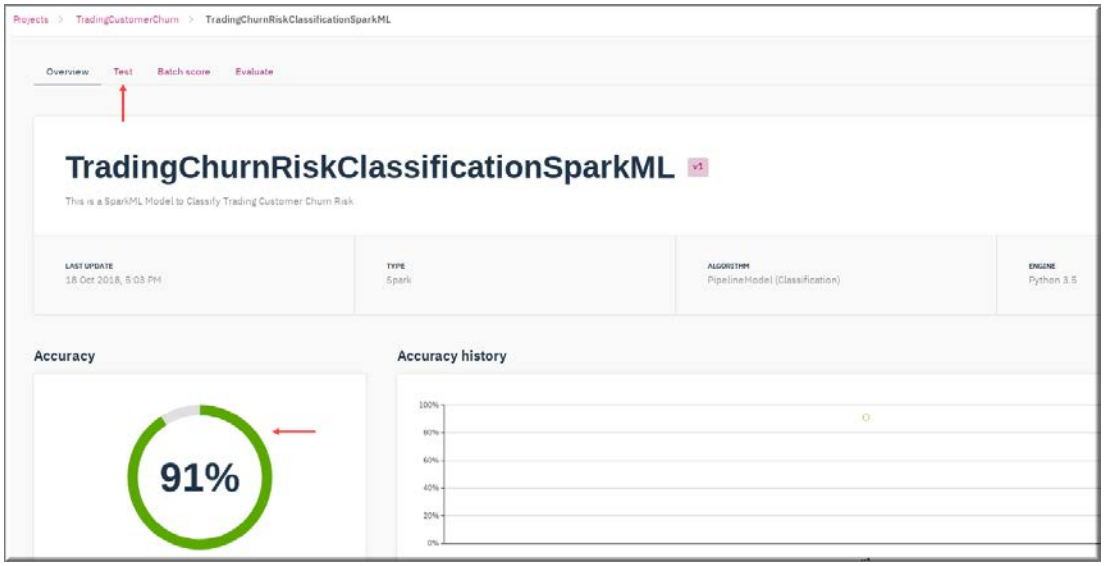
__129. Click `TradingCustomerChurn` Project.



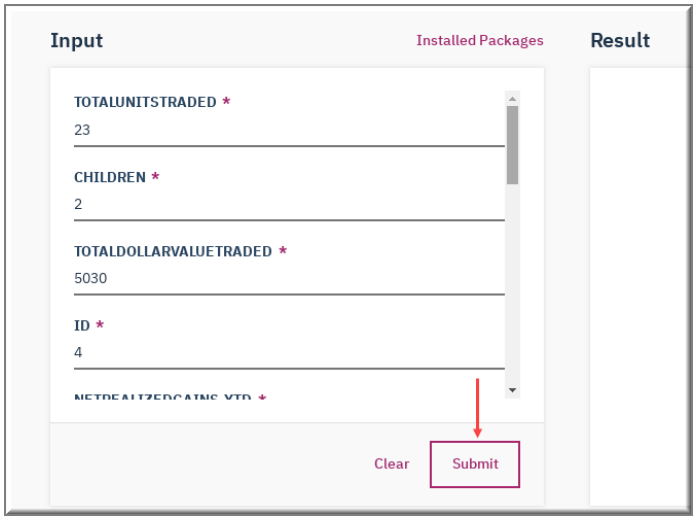
__130. Click `Models`.

__131. Click to open `Trading Custom Churn` model that we saved in the previous section.

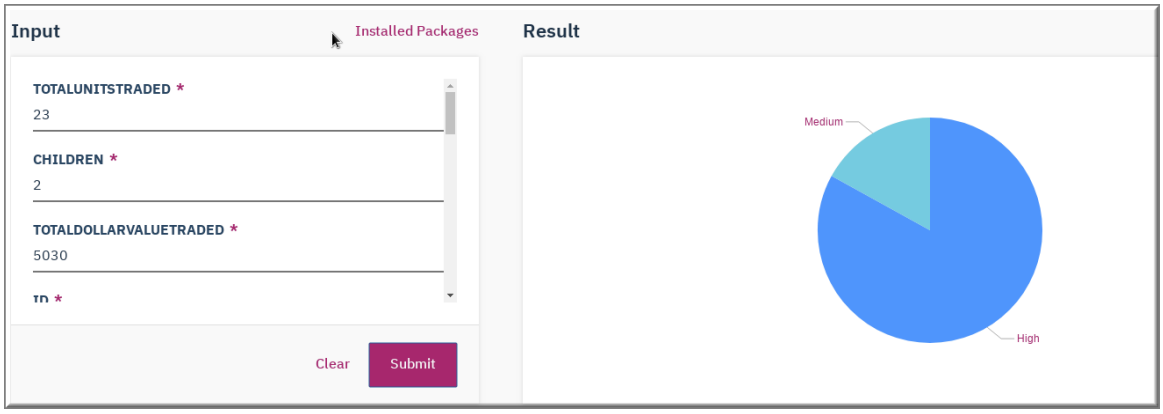
__132. We noticed that the Random Forest Classification gave us 91 percent and Bayes classifier was at 51 percent accuracy. Click `Test`.



__133. Scroll down to the input section where sample input data can be entered to test the model. Click `Submit`.

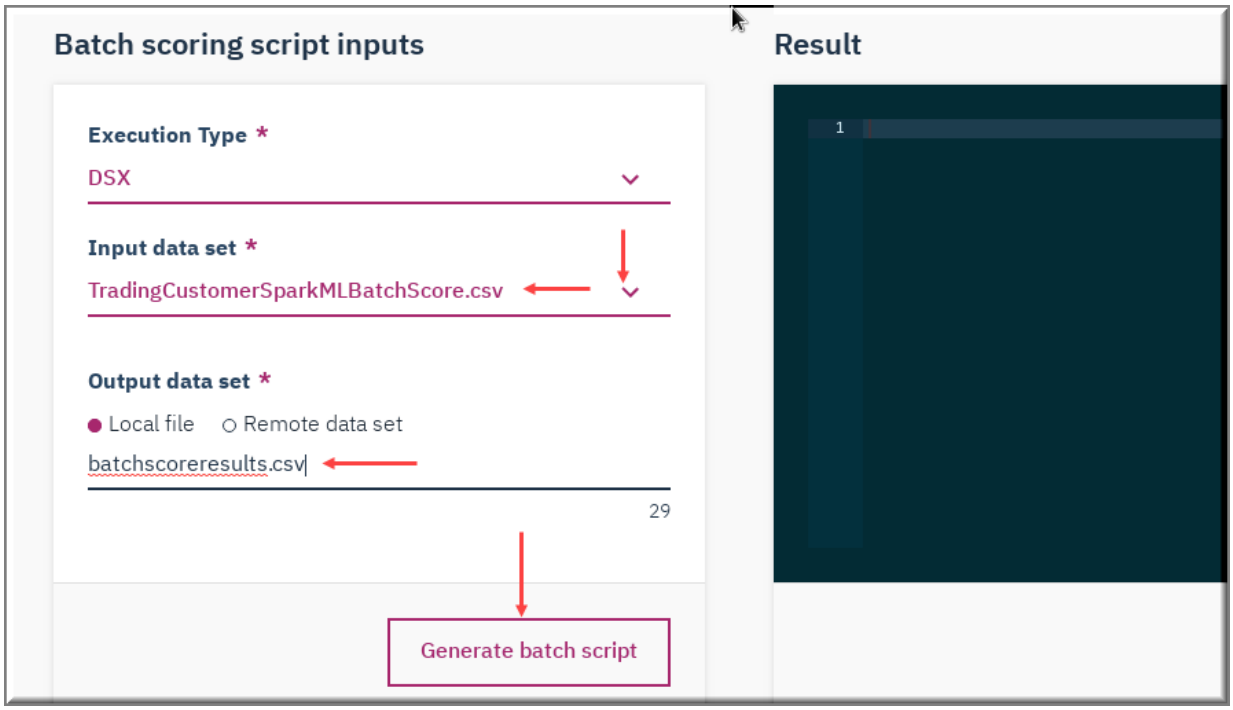


__134. The model online score returns the results indicating percentage of high or medium separation risk.



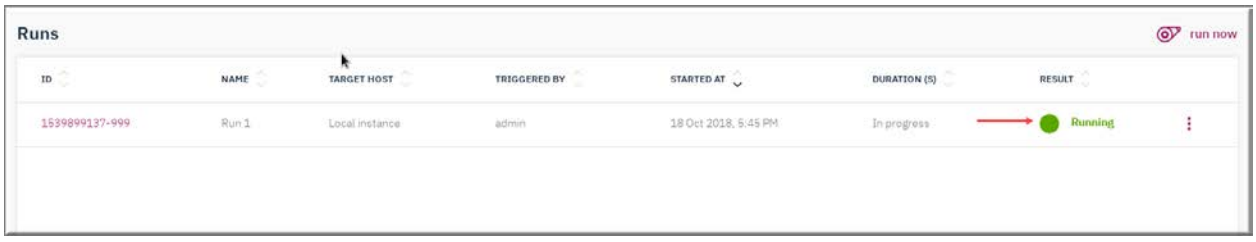
__135. Scroll up and click `Batch Score`.

__136. Select Input data set `TradingCustomerSparkMLBatchScore.csv` and type the output file name `batchscoreresults.csv`. Click `Generate batch script`.

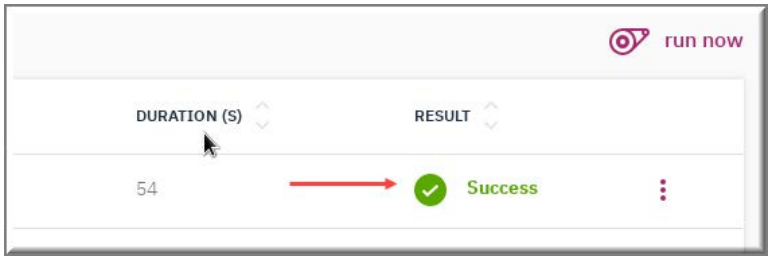


__137. Click `Run now`.

__138. Scroll down and you will see that a batch job is running.



__139. Wait for this to complete and you should see the `Success` message.

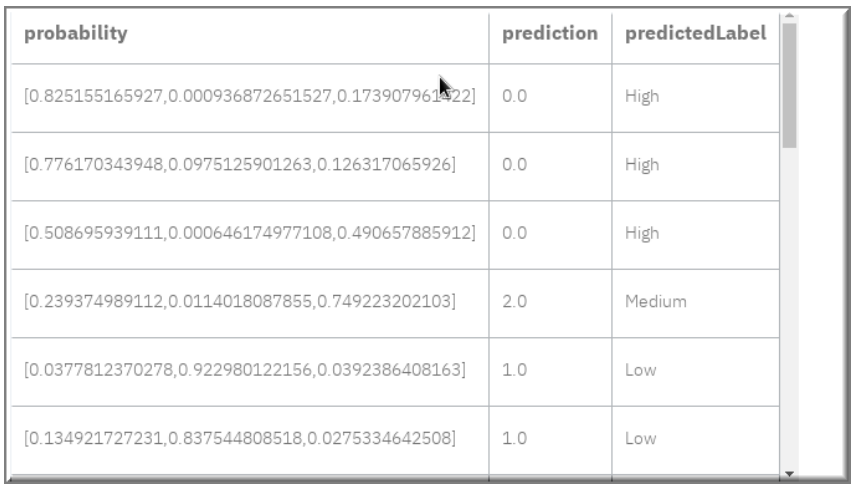


__140. Click `TradingCustomeChurn` project [Top left corner].

__141. Click `Assets ⇨ Data Sets`.

__142. Click `batchscoreresults` to open and preview it.

__143. Scroll to the right and check the prediction for the sample data for different IDs.



probability	prediction	predictedLabel
[0.825155165927,0.000936872651527,0.173907961422]	0.0	High
[0.776170343948,0.0975125901263,0.126317065926]	0.0	High
[0.508695939111,0.000646174977108,0.490657885912]	0.0	High
[0.239374989112,0.0114018087855,0.749223202103]	2.0	Medium
[0.0377812370278,0.922980122156,0.0392386408163]	1.0	Low
[0.134921727231,0.837544808518,0.0275334642508]	1.0	Low

Note: Based upon demographics information of customers and some information from legacy risk applications, the model predicts the probability of separation.

Models must be periodically tested, trained and redeployed.

This is where you will see the value of ICP-D as an integrated data platform.

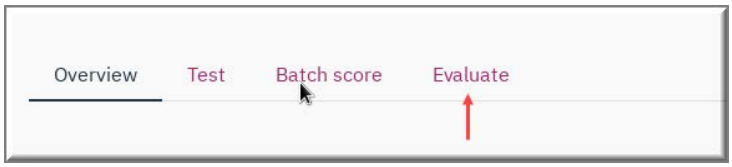
__144. Click `Close` to preview the batch results.

5.2.7 Create an Evaluation

__145. Click `Models`

__146. Click to open the model that we just previewed in previous steps.

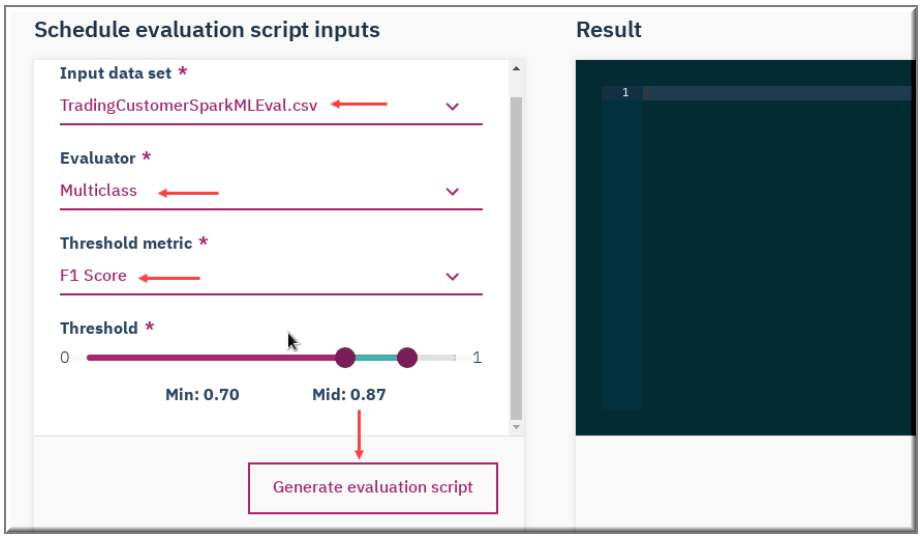
__147. Click `Evaluate`



__148. Scroll down and select input data set `TradingCustomerSparkMLEval.csv` from the drop-down menu.

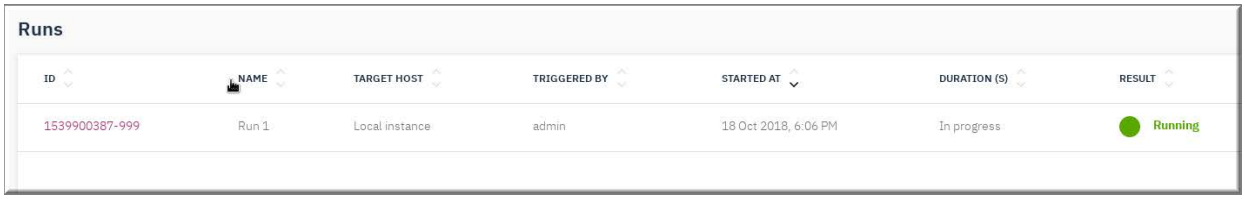
__149. Select `Multiclass` as the evaluator, `F1 score` as the threshold metric.

__150. Change the `Threshold` criteria between `0.71` to `0.89`. Click `Generate evaluation script`.



__151. The script is generated and click `Run now`.

__152. The evaluation job is scheduled. Scroll down and see the job `Running`.



__153. Wait for this job to complete and the Result should show as `Success`.

__154. Click `TradingCustomeChurn` project [top left corner].

__155. Click `Assets ⇨ Data Sets`.

__156. Click `TradingCustomerSparkMLEval` to open and preview it.

__157. Notice the high-risk category prediction – which matches with the business analysis through the heat map.

	**Note:** The journey does not end here but is the beginning. The success is going to be through a repeatable process of prepare, test, evaluate and deploy the model. The platform provides agility that you need from ICP-D. The next lab exercise will show the deployment and the repeatable process that can fit into your CICD pipe line of DevOps.

Lab Credits: Rui Fan and Anjali Saha, IBM Data Scientists.

**** End of Lab 05: Analyze**

Lab 06 Deploy

In our previous lab exercise, we created a model, tested it and scored it using the ICP-D Web UI. Business value is achieved when a successful model is deployed and easily updated.

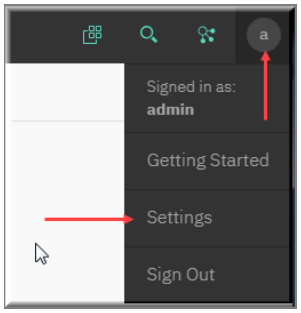
In this lab exercise, we describe the procedure of deploying our model so that it is available for use by application developers.

We will have a much better appreciation of the capabilities that ICP-D platform provides once we finish this lab exercise. We will show automation in harvesting the wealth of information from our data through **Analytics Modernization**.

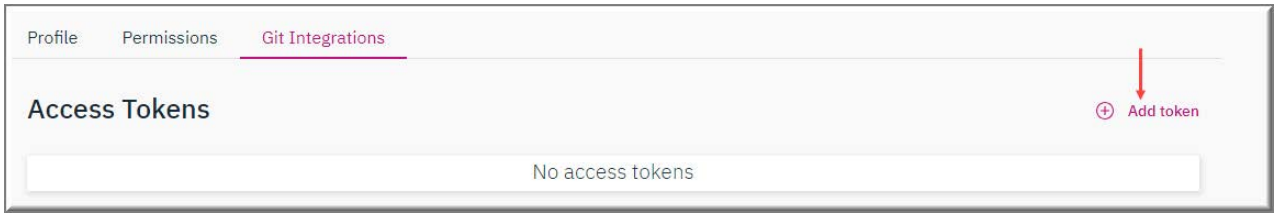
Let's complete this journey with the last pillar of our ICP-D platform.

6.1 Integration with Source Control

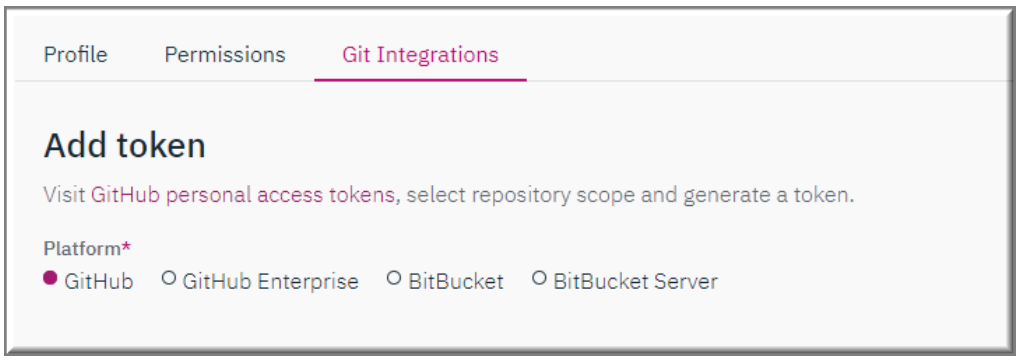
__1. Click a. This is located towards right on the menu bar, then click `Settings`.



__2. Click `Git Integrations` and click `Add Token`.



__3. Notice that you can manage repository with `GitHub`, `GitHub Enterprise`, `BitBucket` and `BitBucket Server`.



6.2 Deploy the Model

__4. You can think of model deployment as the equivalent of writing a self-service application that takes the model and makes it available through a REST API interface. Application developers will access and consume the model through the same interface.

__5. This is a manual process in most organizations. But ICP-D automates deploying and maintaining models without writing a single line of code.

__6. The IBM Cloud Private platform provides an easy method to deploy the model by eliminating the need for the following:

- Write the code to perform this capability and use a runtime to deploy it.
- The runtime could be bare metal machine that requires OS installation, provisioning, network, storage and so on.
- The runtime could be spinning up a virtual machine in a VMware on Intel or an IBM POWER VM® on a POWER platform.
- The runtime could be deploying a Docker image – which requires someone to build the image and deploy it on some platform.

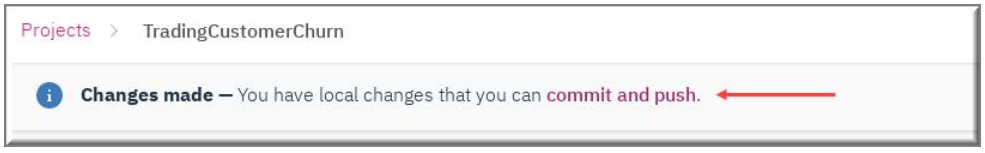
__7. Each of the above requires manpower and machine resources. Using the IBM ICP-D platform, you can bypass this and quickly harvest insight from your data in a repetitive fashion by integrating it with your CICD.

__8. Let's examine the automated delivery and node deployment which you can integrate with your DevOps implementation.

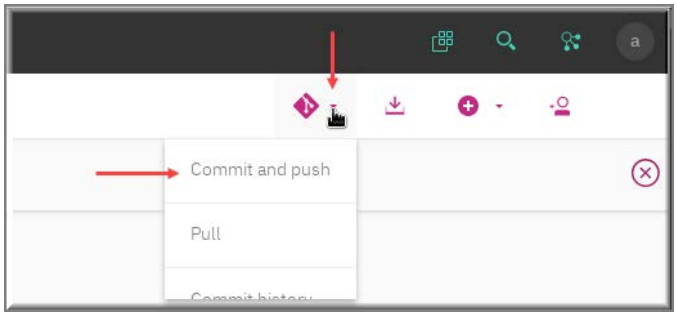
__9. Click `Projects` from the left menu bar.

__10. Click `TradingCustomerChurn` to open it.

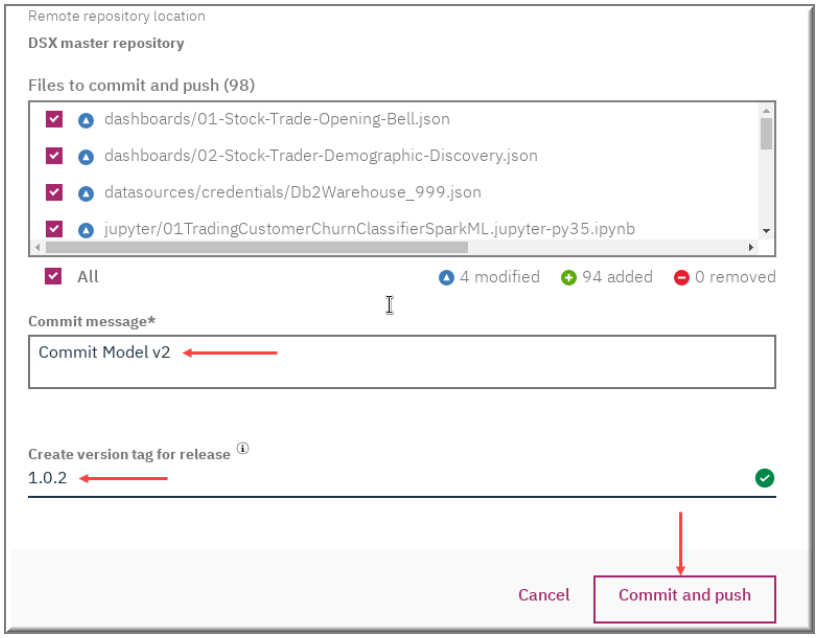
__11. You might see this message.



__12. If you do, click `Git` icon and click `Commit and push`.



__13. Type: *Commit message* `Commit Model v2,` type version tag as: `1.0.2, and` click `Commit and push`.



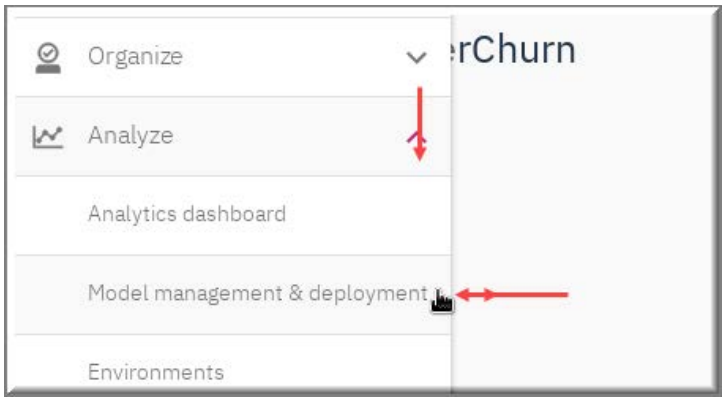
__14. It may take a minute or two. Wait for it to complete. When completed. The following message displays:

__15. Click `View`.

__16. Scroll down to see the collaboration team who has worked on this including: admin (you), Anjali, Rui, Kikuchi, Rubin and so on.

6.2.1 Model Management and Deployment

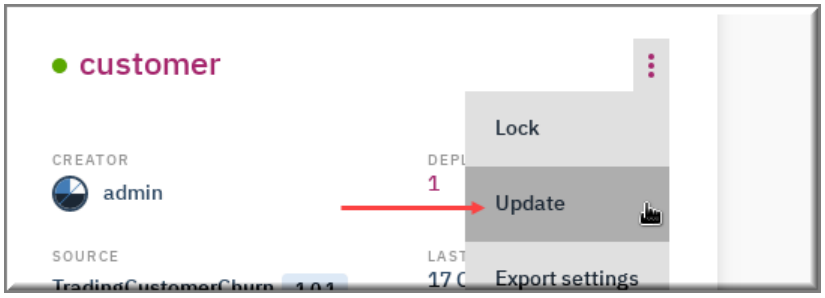
__17. Click **Analyze ⇨ Model management and deployment**



__18. You would notice that a customer release is already up and running.

__19. For the purpose of the Executive Demo, we deployed the model already. We now use the same `Project Release` to deploy the second version.

__20. Click the three vertical dots and click `Update`.



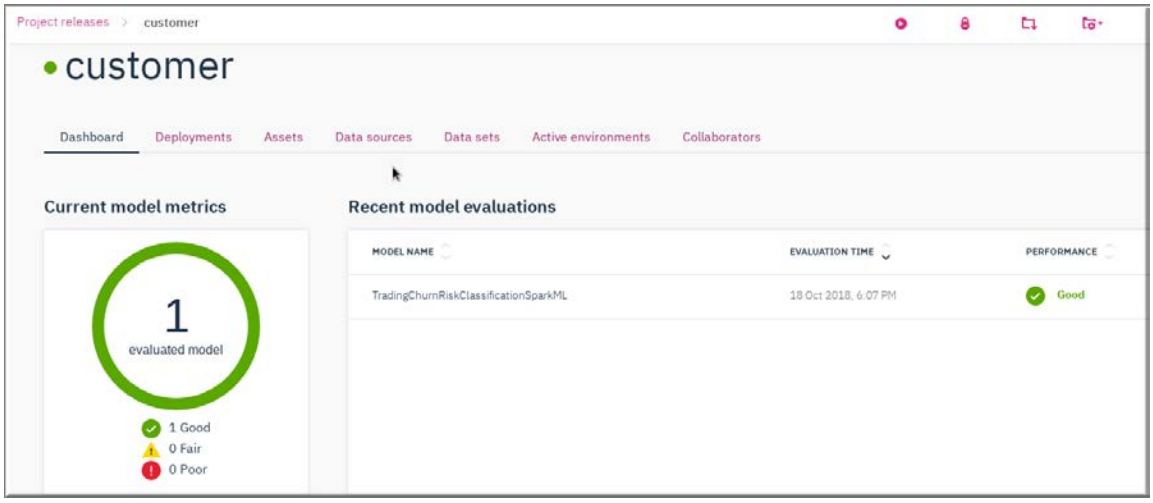
__21. Select `TradingCustomerChurn` from the drop-down menu and Tag `1.0.2`.

__22. Notice the *Name* `customer` and *Route* `award.` These are the name and route we used to deploy the model. The name identifies the model and route is used by the proxy within ICP-D to route the request.

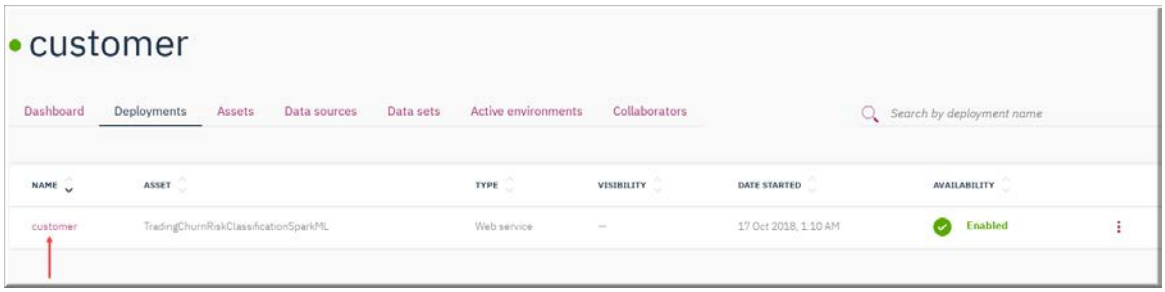
__23. Click `Next` to see the list of assets to be updated.

__24. Click `Update`.

__25. You will see current model metrics and the evaluations that we ran in our previous lab exercises.



__26. Click `Deployments`

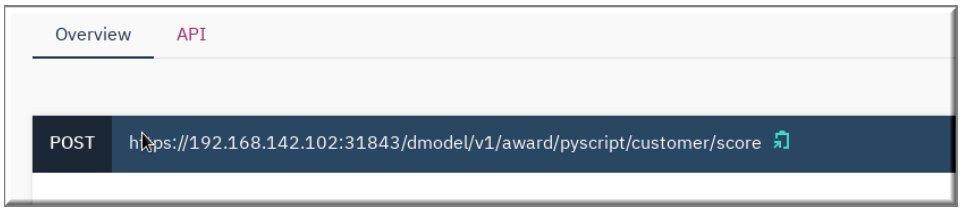


__27. The model is deployed and enabled.

	Note: ICP-D platform builds a Docker container and using the underlying IBM Cloud Private, the container and its Kubernetes pod are deployed onto the cluster. Once deployed they are ready to service the API requests and provide results back from the model. This is an example of the automation that our ICP-D platform provides in support of integrating data and application development pipelines.

__28. Click `customer`.

__29. The `Overview` section shows the `POST` REST API that an application can consume.

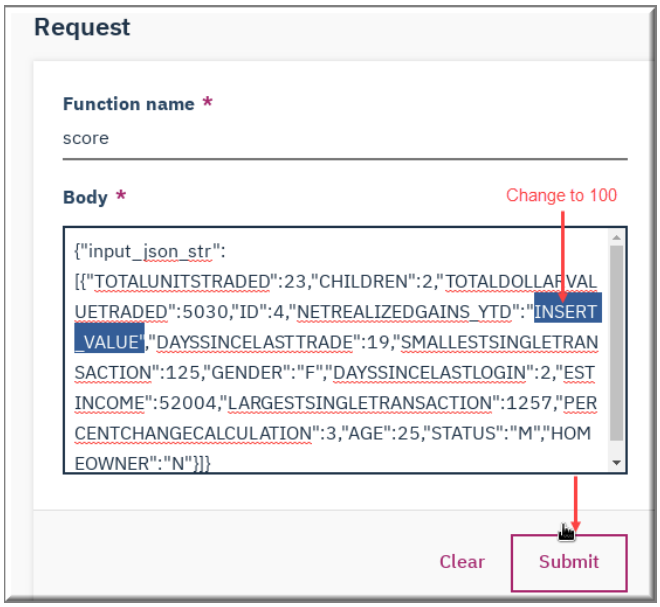


6.3 Test Deployment

__30. The model deployment through ICP-D is secured with an authentication token. Any applications using the model need to authenticate themselves by providing this token.

__31. Scroll down to the **Function** metrics. These give an overview of the REST API consumption metrics.

__32. Click `API` and scroll down to the `Request` section. Look for `INSERT_VALUE` and change it to `100`. Click `Submit`.



__33. You can see the `Response` from the model (which was deployed). The `JSON` output is returned by the `REST API`.


```
{
  "result": [
    "{predictions: [High], probabilities: [[0.572096381241254, 0.24983552631578942, 0.17806809244295652]],
      classes: [High, Low, Medium]}",
    "",
    0
  ],
  "stdout": [
    "+---+--------+-----------------+-----------------+---------+------+---------+---+-----------------------
    "|AGE|CHILDREN|DAYSSINCELASTLOGIN|DAYSSINCELASTTRADE|ESTINCOME|GENDER|HOMEOWNER| ID|LARGESTSINGLETRANSACTION
    "+---+--------+-----------------+-----------------+---------+------+---------+---+-----------------------
    "| 25|       2|                2|               19|    52004|     F|        N|  4|                   1257
    "+---+--------+-----------------+-----------------+---------+------+---------+---+-----------------------
       +--------------------+-----------------------+-------------------------+------+---------------------+----------------+
       |NETREALIZEDGAINS_YTD|PERCENTCHANGECALCULATION|SMALLESTSINGLETRANSACTION|STATUS|TOTALDOLLARVALUETRADED|TOTALUNITSTRADED|
       +--------------------+-----------------------+-------------------------+------+---------------------+----------------+
     |                 100|                      3|                      125|     M|                 5030|              23|
     +--------------------+-----------------------+-------------------------+------+---------------------+----------------+
     -----------+-------------+-----------+-------------+-------------+----------------+--------------------+--------------------+
     GENDERIndex|GENDERclassVec|STATUSIndex|STATUSclassVec|HOMEOWNERIndex|HOMEOWNERclassVec|            features|       rawPrediction|
     -----------+-------------+-----------+-------------+-------------+----------------+--------------------+--------------------+
            0.0| (1,[0],[1.0])|        0.0| (2,[0],[1.0])|          0.0|    (1,[0],[1.0])|[1.0,1.0,0.0,1.0,...|[11.4419276248250...|"
     -----------+-------------+-----------+-------------+-------------+----------------+--------------------+--------------------+
     --------------------+----------+--------------+",
             probability|prediction|predictedLabel|",
     --------------------+----------+--------------+",
     [0.5720963812412 5...|       0.0|          High|","
     --------------------+----------+--------------+",
  ],
  "stderr": []
}
```


__34. Notice within the above response, the prediction probability is returned, based on the input JSON that is given to the model.

__35. Click `generate code`

__36. The attached curl command is generated. This can be used to test the REST API and can be given to the application developers for them to implement the API in their application.

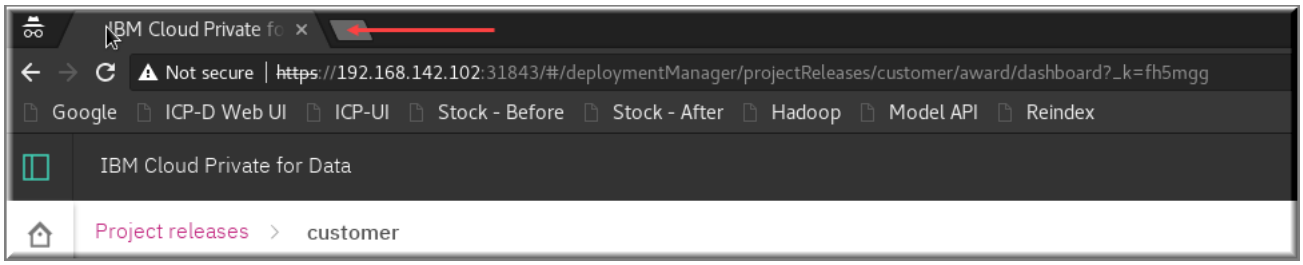
6.4 Consume Deployment in Stock Trader Application

__37. The machine learning model was created by data scientists and deployed using ICP-D platform. Now, let's see how a modern microservices-based application can consume it.

__38. We have shown the `Stock - Before` and `Stock - After` application in the Executive Demo lab.

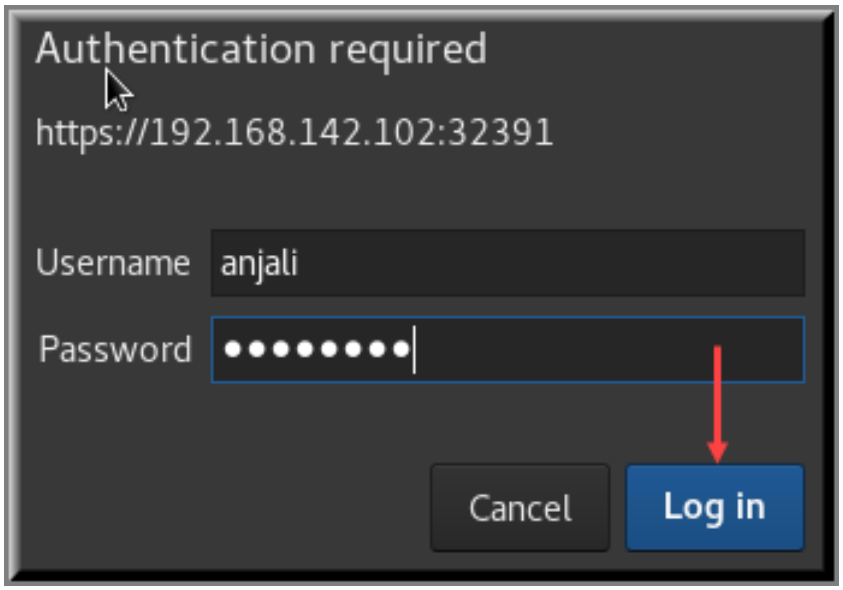
__39. Let's revisit it more detail.

__40. Open a new browser window tab.

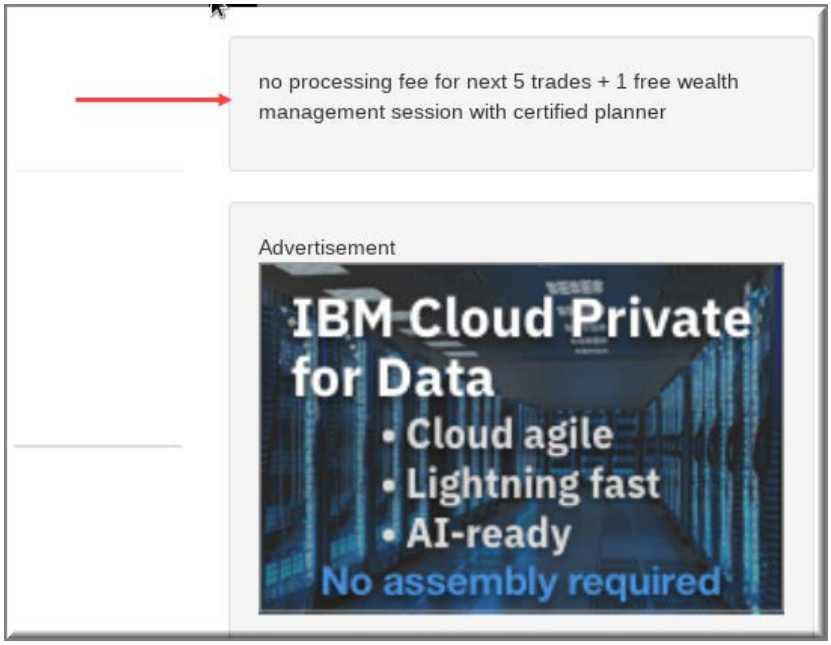


__41. Click `Stock - After` desktop icon.

__42. Log in as *Username*: `anjali` and *Password*: `password`. Click `Log in`.

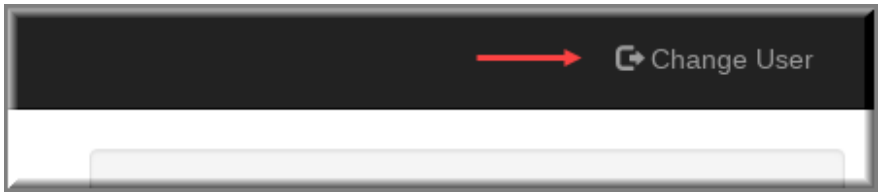


__43. Notice the offer box that shows after logging in to the application.

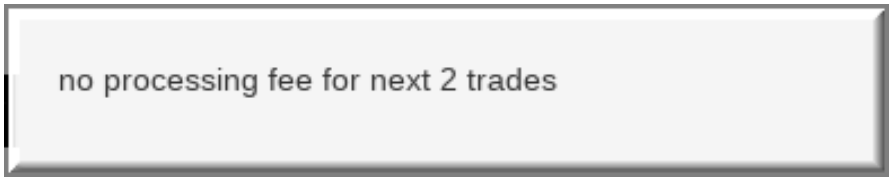


__44. Based on user ID of the login, we get our demographic information about that user from our database, provide that information to a REST API call from the backend of this microservice. The demographic information is given to the model running on the ICP-D platform. The model returns a response (churn prediction value), based on the response business logic, is invoked to present a retention offer to the user. All of this is performed in real time, the user then logs into the application.

__45. Click `Change User`.



__46. Type User: `dan` and Password: `password` and see the offer.



__47. The offer is based on the risk assessment of the ID of the user.

__48. Let's go further into the REST API call.

__49. Open one more browser tab and click `Model API` desktop icon.

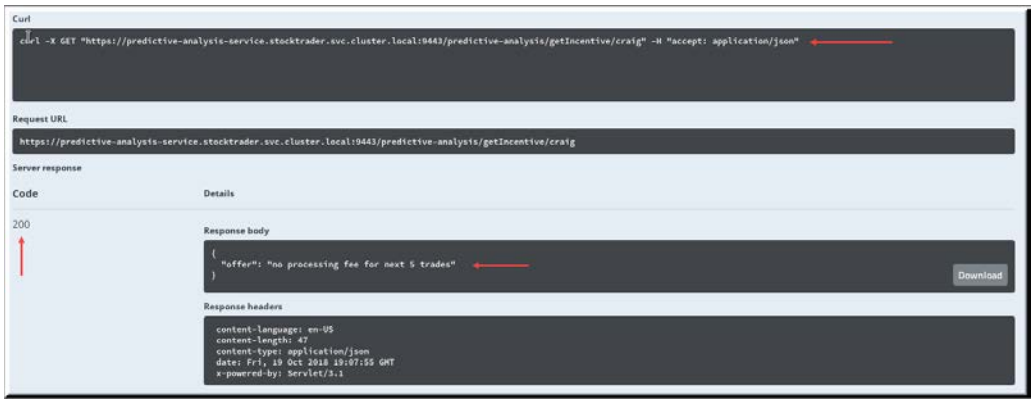
__50. Click `GET` to test the REST API.

__51. Select the HTTPS version of the `REST API` as `http` will be rejected by the proxy server of ICP-D.

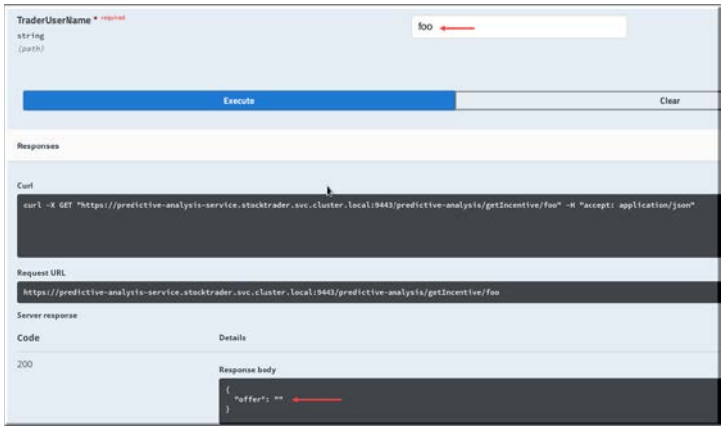
__52. Click `Try it out`.

__53. Type `craig` and click `Execute`.

__54. Notice the responses: `curl` command with GET verb, 200 response code, and the response body which is consumed by the application.



__55. Try user `foo`. You will notice no response as we have no demographic information for this user.



6.5 Conclusion

__56. As an organization, we collect, and have access to, more data than we can possibly analyze.

__57. Data is spread all over, in many silos, and is hard to find. Locating that one person in an organization who has the breadth and depth of knowledge of our data is like looking for a unicorn.

__58. IBM Cloud Private for Data can be that unicorn. It provides an integrated data platform where data scientists can shop for data, Data engineers help to prepare the data. Business analysts provide visual insights of data to the stakeholders. Data stewards can discover, classify, provide governance, rules and policies to meet regulations.

__59. Turning your data into a corporate asset with end-to-end data integration on modern, cloud-native platform. We have completed the IBM Journey to Cloud and AI: Analytics Modernization.

**** End of Lab 06: Deploy**

Appendix A. Stock – Opening Bell Dashboard

__1. Let's begin by analyzing current trends of customer visits and daily trades in our Stock Trader application.

__2. We requested data engineers to provide a file with historical totals of visits and trades for the past year. This file was provided and deposited in our project where we all collaborate with each other.

__3. The data steward guidelines of the company do not allow data to be migrated to individual's laptop for analysis.

__4. Let's build a dashboard with this data to see if we see any trends.

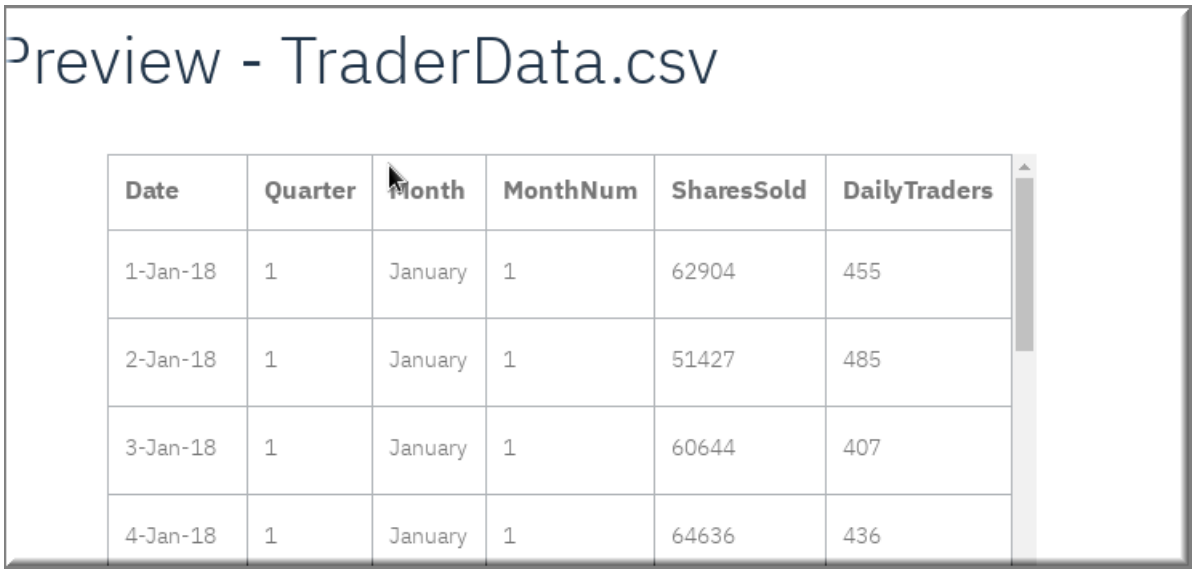
__5. Click `Projects` from the left menu bar.

__6. Click `TradingCustomerChurn` project.

__7. Click `Assets` and then click `Data Sets`.

__8. Notice `TraderData.csv` available in our project. We will use this data file to build the `Stock – Opening Bell` Dashboard.

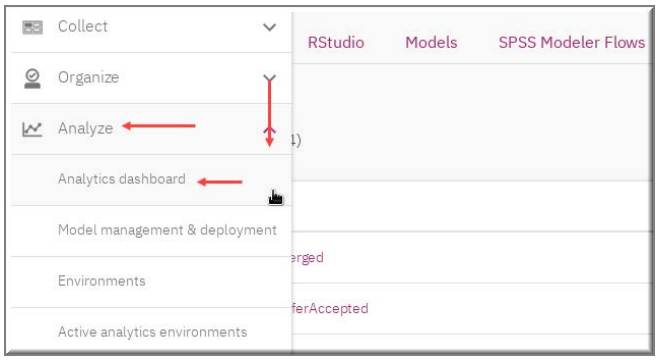
__9. Click the link `TraderData.csv` to preview the data.



Date	Quarter	Month	MonthNum	SharesSold	DailyTraders
1-Jan-18	1	January	1	62904	455
2-Jan-18	1	January	1	51427	485
3-Jan-18	1	January	1	60644	407
4-Jan-18	1	January	1	64636	436

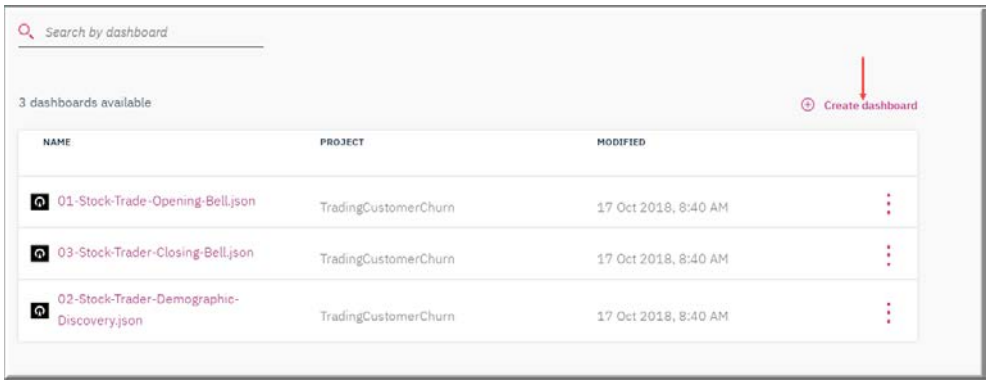
__10. Click `Close` after preview.

__11. Click `Analyze ⇨ Analytics` **dashboard** from the left menu bar.

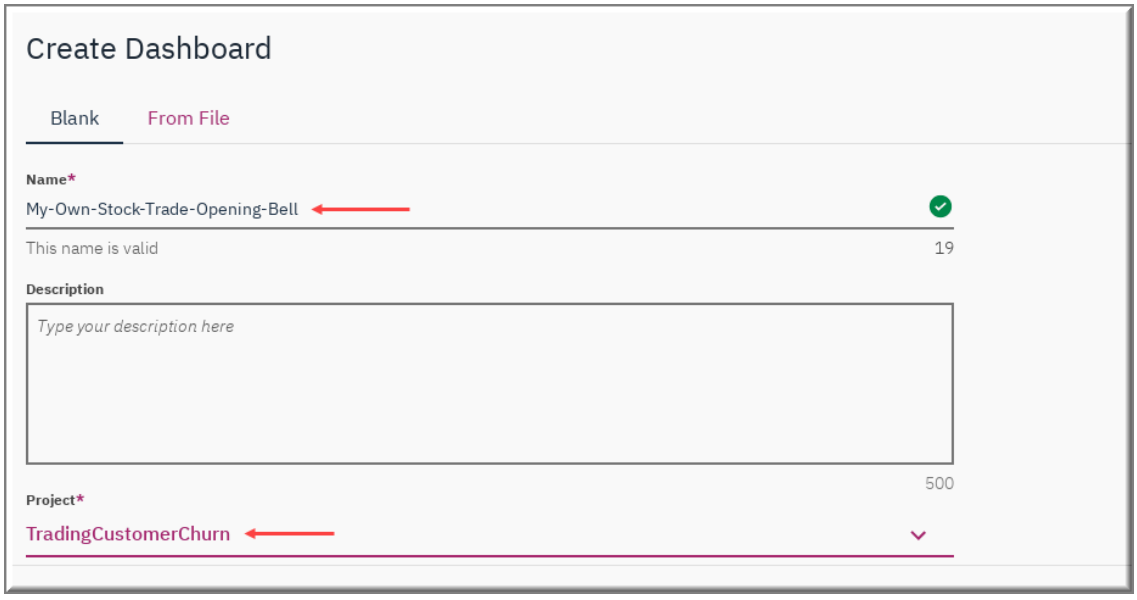


__12. We used the `01-Stock-Trade-Opening-Bell` dashboard in the Executive Demo lab and we will build this exact same dashboard here.

__13. Click `Create dashboard`.

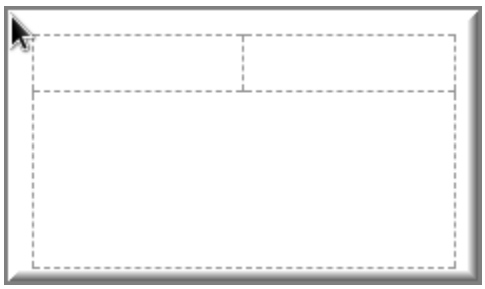


__14. Type name: `My-Own-Stock-Opening-Bell` and choose `TradingCustomerChurn` project from the drop-down menu.



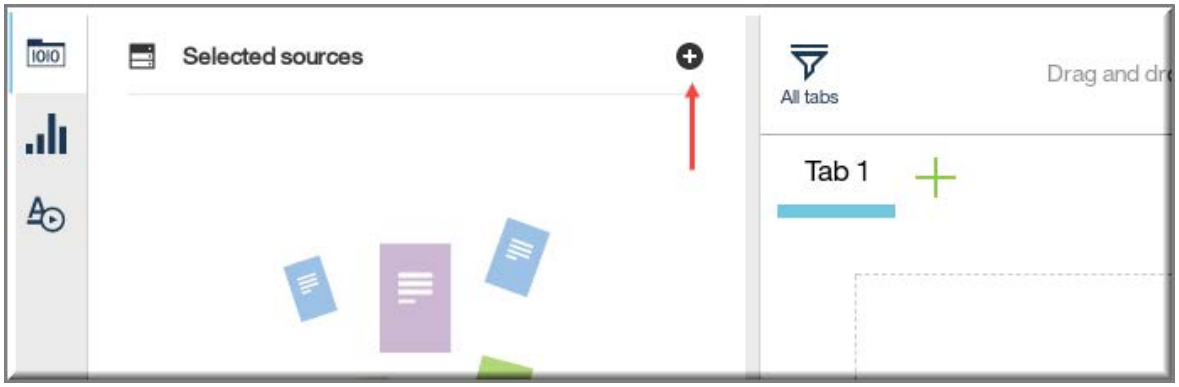
__15. Clock `Create`. [bottom-right corner.]

__16. You are now presented with a choice of canvas templates. Select the one that looks as shown:

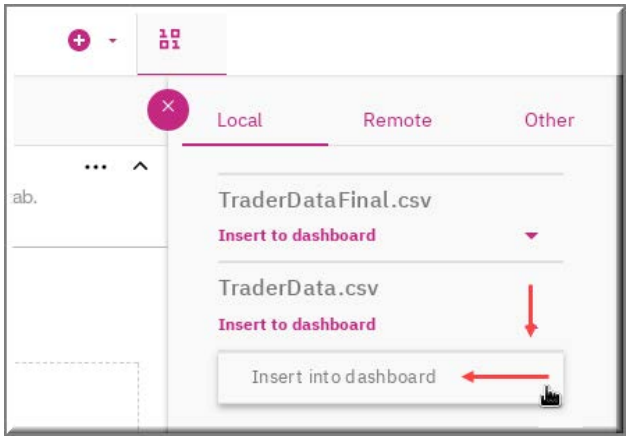


__17. Click `OK`.

__18. From the `Selected sources` area near the top left, click 



__19. The right pane will slide into the canvas. Click the down arrow of `TraderData.csv` and click `Insert Into dashboard`.

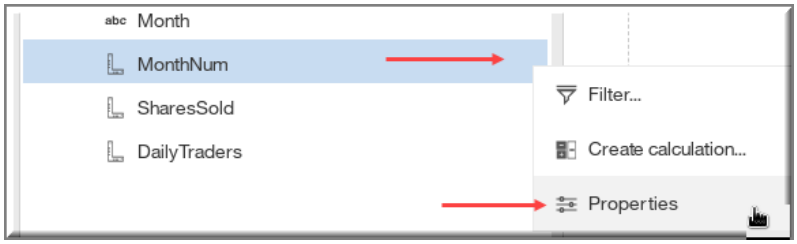


__20. You will now see `TraderData` in the `Selected sources` area.

__21. Click `TraderData` and expand it to show all the data items in that data source file.

__22. Let's do some housekeeping so that our data is represented correctly within our dashboard.

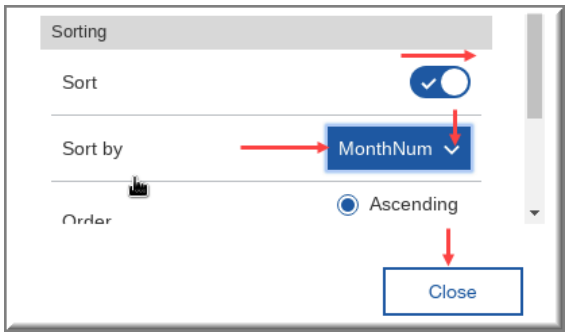
__23. Click the `MonthNum` ellipse, then select `Properties` from the flyout menu.



__24. `MonthNum` in this case is not a measure. It's simply an attribute on which we sort our months. Change `MonthNum` usage to be an `Attribute` and click `Close.`

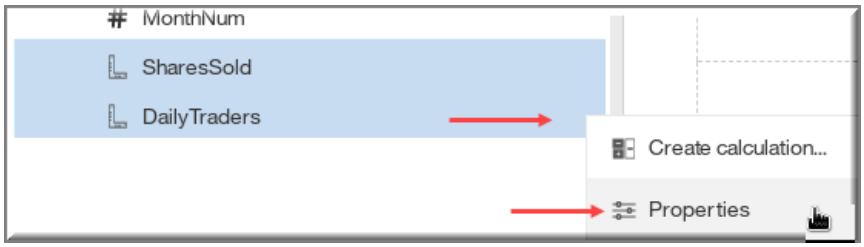
__25. Now, let's select the Properties of `Month` by selecting the ellipse next to `Month`.

__26. Slide the `Sort` icon to turn it on and sort by `MonthNum` in `Ascending` order, then click `Close`.



__27. Next, we set the aggregation of our measures for totaling.

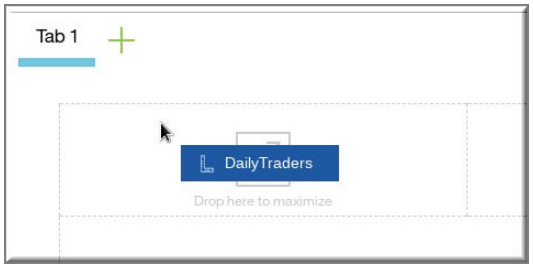
__28. Hold the shift key down and click `SharesSold` and `DailyTraders` to select both. Click the ellipse and then click `Properties`.



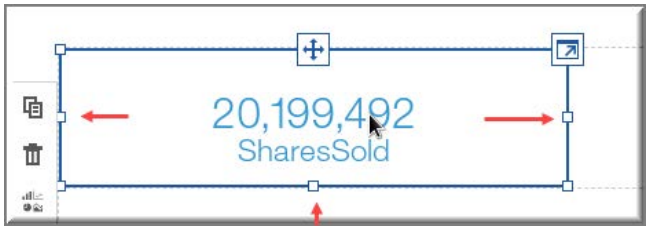
__29. Change *Aggregate* to `Total` from the drop-down menu and click `Close`.

8.1 Build Dashboard

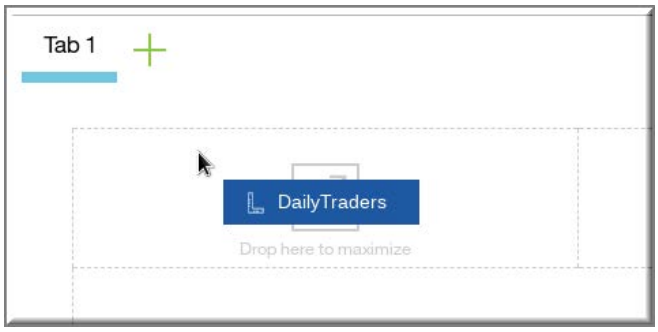
__30. Drag `SharesSold` to the top left box, hovering over the *Drop here to maximize* area when it turns blue. This gives us a total of `Shares sold` over all time.



__31. If you did not drop at the right place, adjust the guide to match with the outline of the left box.

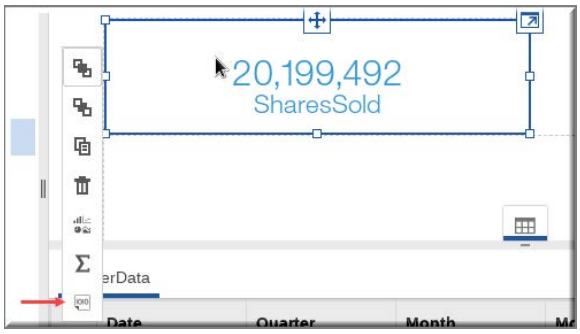


__32. Drag `DailyTraders` to the top right box, hovering over the *Drop here to maximize* area when it turns blue as well. This maximizes this metric in this box. This gives us a total of trades over all time.

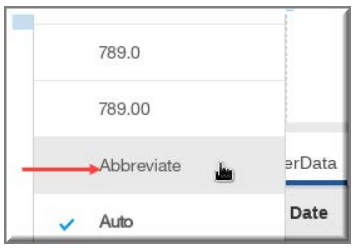


__33. After you complete both top boxes, the dashboard should display as shown:

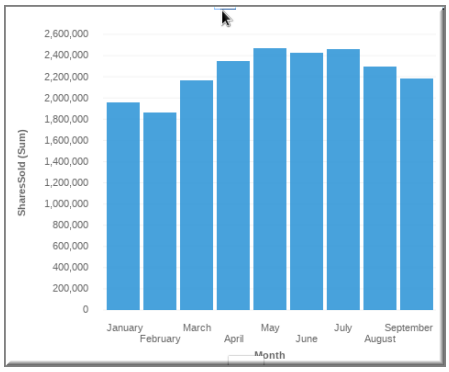
__34. Click `SharesSold` metric and select the `format` from the flyout menu.



__35. Click `Abbreviate`.



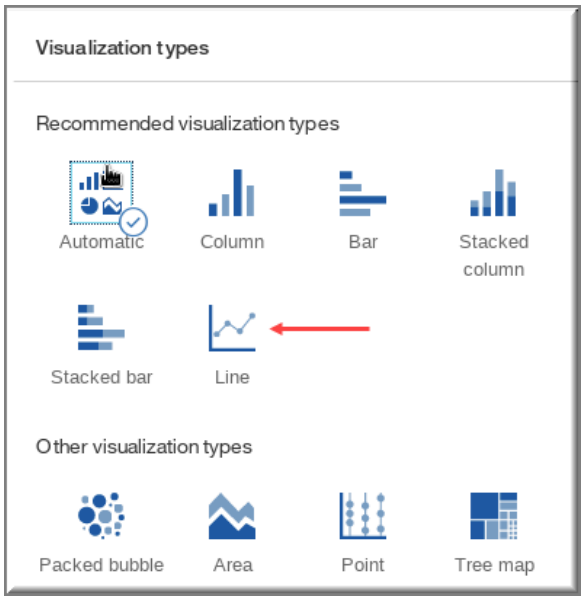
__36. Hold the shift key down, click `Month` and `SharesSold` and drag the two onto the lower canvas area. [Do not drop in the *Drop here to maximize* area this time.]



__37. Click (on top right of the chart). This will expand the chart and allow you to make adjustments.

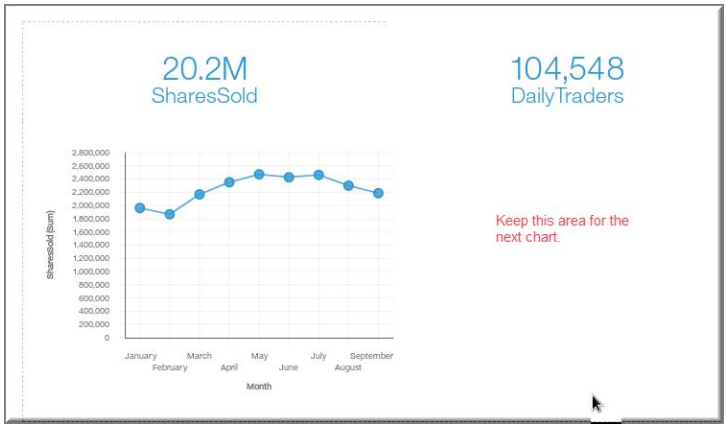
__38. Click the drop-down menu to change the type of chart from Column to **Line Chart**.

__39. Select `Line` visualization type.



__40. Now select the `Expand` button again to return it to it's original size.

__41. Drag the chart so it aligns with the left side of the template area and squeeze in the right side of the template so it aligns with the middle of the screen.

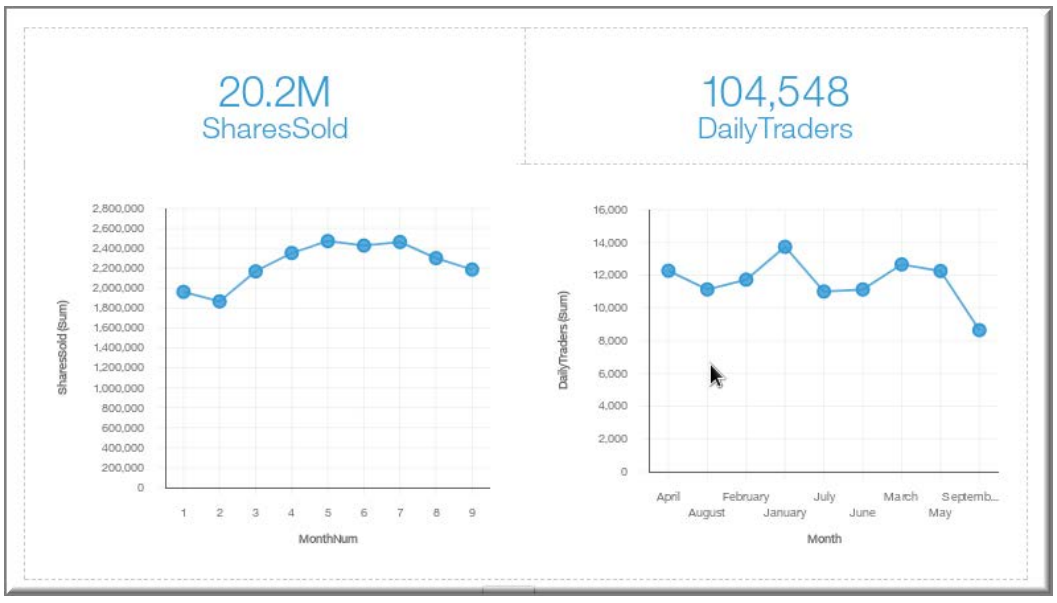


__42. Hold the shift key down, click `Month` and `DailyTraders` and drag the two onto the lower canvas area. [Do not drop in the *Drop here to maximize* area this time.]

__43. Change the visualization from `Column` to `Line` as we did with the first graph. [Click **Expand**, change the type and click the same button to bring its in original size.]

__44. Adjust the top, bottom, left and right of the chart boundaries so that all the boxes are aligned.

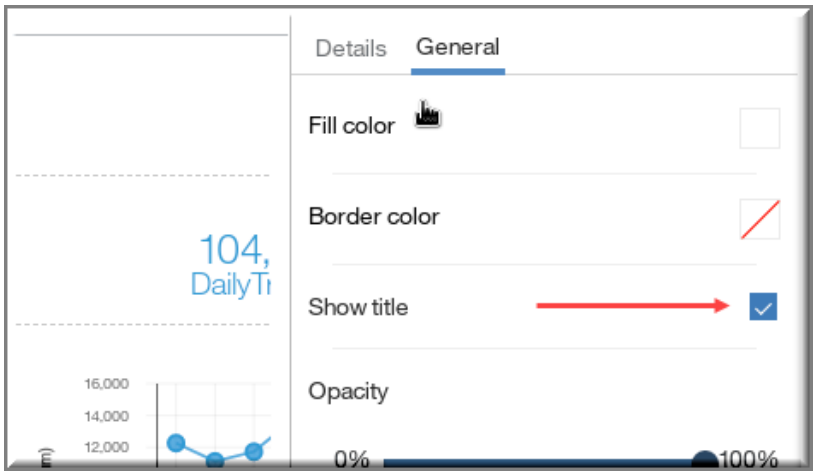
__45. Your final dashboard should display as below.



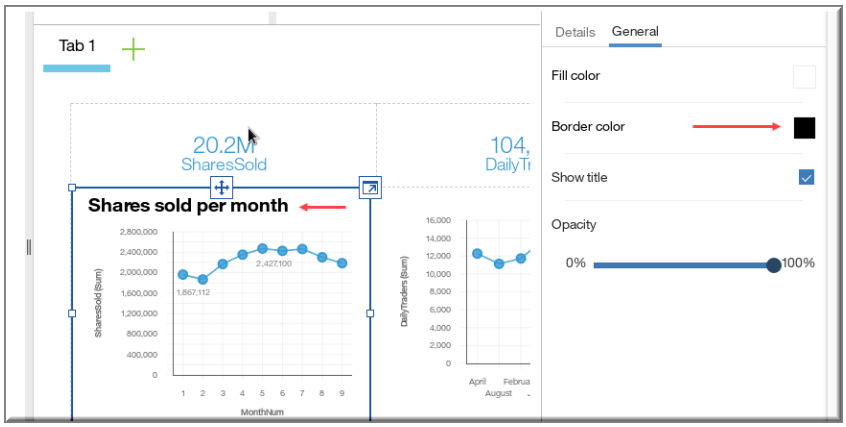
__46. Select the bottom left chart and then select the `Properties` button at the top.

__47. Check `Show Value Labels` from the Details tab.

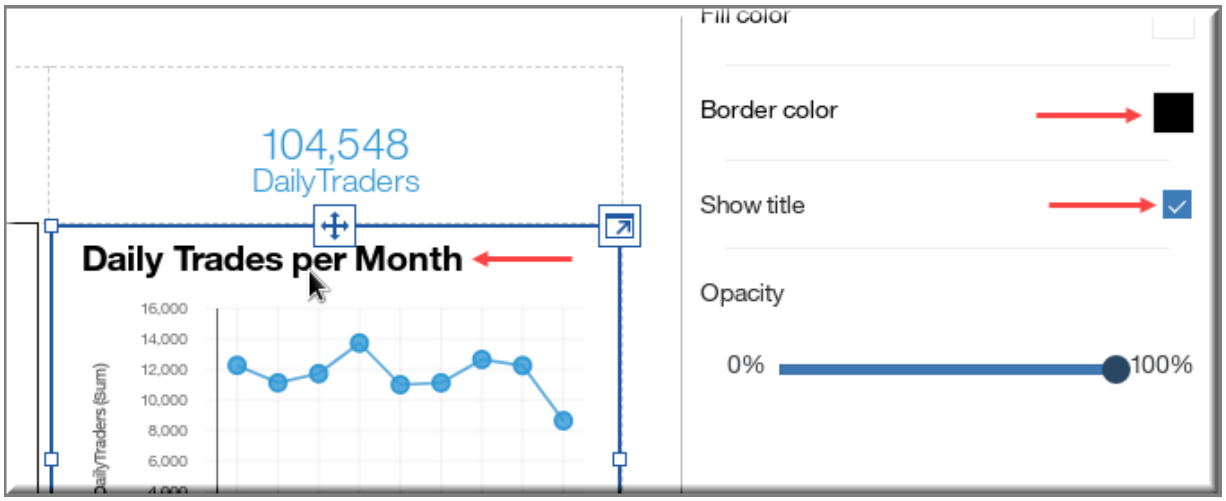
__48. Now select the **General** tab and check to `Show title.`



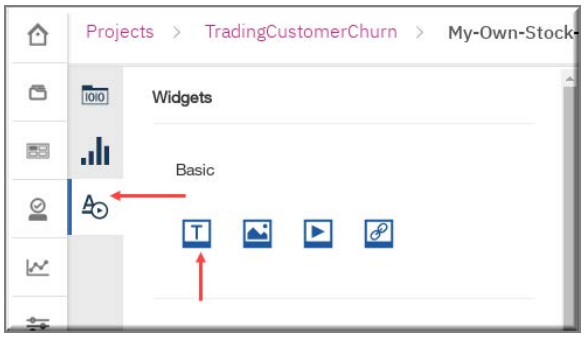
__49. Enter `Shares sold per month` for the title of the left chart and change the border color to **Black**.



__50. Click the bottom right-hand chart and enter `Daily Trades per month` for the title of the right chart and change the border color to **Black**.

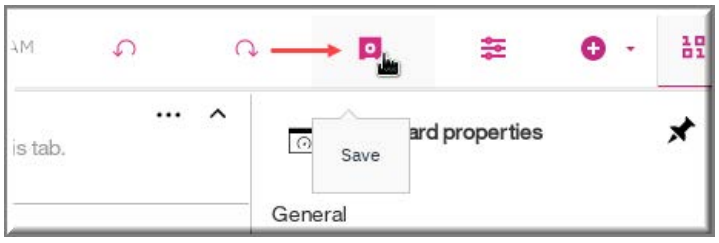


__51. On the left side of the canvas, click the `Widgets` menu and drag a `Text box` between the top two charts.



__52. Title it **Stock Trades**. Adjust the box so the title stands out.

__53. Click `Save`.



__54. You notice from examining the charts that `Shares sold` is relatively flat and daily trades are falling off. We need to use IBM Cloud Private for Data to discover the WHY behind this trend.

Appendix B. Integrated Data Platform

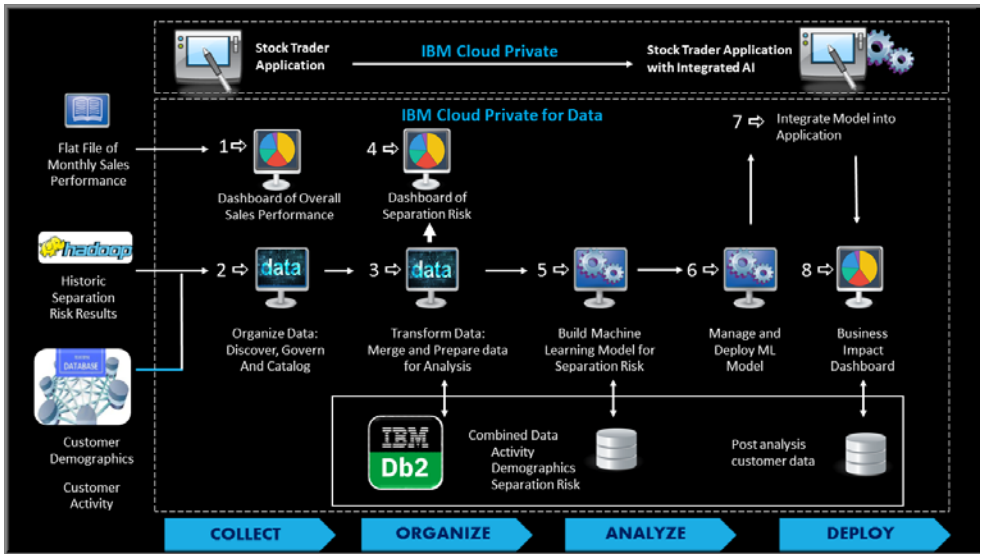
This workshop covers end-to-end analytics modernization with examples to help you on this journey.

Regardless of the role you play in your organization, you are probably aware of difficulties faced by people throughout the Lines of Business and IT. Some of these may sound familiar to you:

- Our data is very siloed. It is a huge challenge to search for it, understand the 'business context' of it, determine its quality and if it can be trusted, and knowing the right people across our business to resolve these challenges.
- The data owner can't give me access to production data, so we have to copy full data sets into our own environment. This is time consuming and a duplication of effort. Also, the data is stale the moment I get it. Note: A case study found that one bank had 82 data warehouses and 63 percent of its data is a duplicate of other data.[1]
- The data we get from different silos is unorganized and it is a complex effort to clean and prepare it. It takes 80 percent of our time to just prepare it. For some projects, this takes many months. In one case study, it was found that 73 percent of the time is spent in Extract Transform and Load (ETL), working with DBAs and infrastructure support teams to get the data.
- We do not have borderless data and we do now know the lineage of the data.
- Building good models takes good data. I'm not always confident in the data I'm using because it's curated by tribal knowledge. Once I do find a good source, or spend effort curating data, it's not discoverable and readily available to others.
- Once I create a well-performing, predictive model, I struggle with the operational aspects of making the model easily consumable by application developers and keeping the production model tuned over time. Multiply this by dozens (or hundreds) of models from multiple lines of lines of business across our enterprise and you see our struggle.
- New programming languages and their resulting tools for data scientists are becoming available annually. We need a way to efficiently support all of this tooling, and the multiple versions of each individual tool.

Let's start working with an example and see how ICP-D helps address these challenges.

For the purpose of this workshop, let's examine the following workflow involving `Business Analyst`, `Data Engineer`, `Data Scientist` and `Data Steward`. Each one of these personas has a role in each of the four lab exercises of `Collect`, `Organize`, `Analyze` and `Deploy`.



[1] Eric Watson, IBM Corporation

8.2 Application with Analytics Modernization

__55. Boatswain is the hypothetical company for which we work. It is an on-line stock trading company. We have approximately 100,000 customers who maintain their portfolios and buy/sell stocks through our website. One of our bigger challenges is retaining customers. Their cost of switching to a new provider is very low. As a business, we need to understand our 'Customer Separation Risk'. Specifically, how many customers are we losing to competition, how much is this costing us, what are the main drivers and/or customer profiles who leave, what actions will we take to retain these individuals, and what are the costs and benefits of our corrective actions?

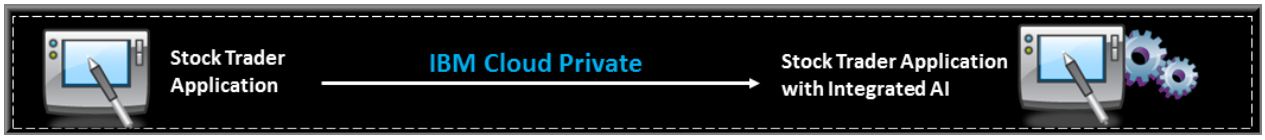
__56. Our corporate strategy is to offer 'Individually customized' retention offers and other services so that we differentiate our website and customer experience from our competition. With this approach we will 'Market to a Segment of One Customer' versus marketing to groups or subsegments of our overall customer base.

__57. As a first step in our journey, we successfully ported our core customer-facing application from a 'legacy monolithic architecture' to a 'cloud-native, microservices architecture' running in Docker containers on a Kubernetes cluster. The next step in our journey is to add artificial intelligence so that we can effectively 'Market to a customer segment of one customer'.

__58. The name of our application is `Stock Trader`. At the start of this workshop, the application does not yet include AI and we are calling the application "Stock – Before". This application can be deployed in any Kubernetes-based cloud environment. For this workshop, we are running the application in `IBM Cloud Private`.

__59. The outcome of our workshop is to enhance our application with a machine learning model (artificial intelligence) to drive increased revenue and profit by significantly improving our customer retention problem. We will name this application "Stock – After".

__60. This is the top box in our workflow.



Note: `Stock – Before –` there are no individual retention offers.

`Stock – After` – we get `high`, `medium` or `low` risk separation probability through our machine learning prediction model for the logged-in user. The offers are made 'in real time' based upon risk.

Low – 3 free trades
Medium – 5 free trades
High – 5 free trades plus a free consult from a wealth management advisor

8.3 Collect

The purpose of the Collect lab is to show ways to get access to data that is locked in different silos. To get started, we simply upload a local .csv file to build our first dashboard.

'Passing of data between users' is a common practice in many enterprises that can lead to unmanageability. When data is 'passed around' on thumb drives, it is generally not secure nor governed. It may include Personally Identifiable Information (PII) such as: customer name, age, tax id number, address, and more. This practice can lead to security breaches, tarnishing a company's reputation in the marketplace, and potentially legal action.

ICP-D provides a better way to protect and govern your data, while at the same time making it easily accessible to business analysts and data scientists.

Through this workshop, we will connect to different data sources including: Hadoop and DB2. Note that ICP-D can currently connect to additional sources in a "Secure and Governed way' including: Oracle, SQL Server, Teradata, and Event Store.

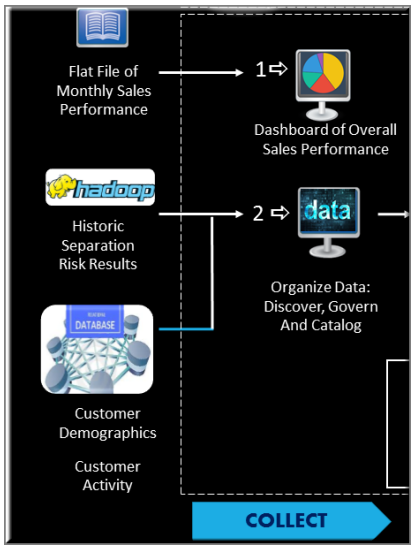
Stay tuned for the next version of this workshop. It will contain new capability on 'virtualizing SQL Queries' across these data sources.

__61. Boatswain has realized that revenue and profits are declining. This will be highlighted in a Sales Performance Dashboard that you will build. Throughout our workshop, we justify additional investment in our project based upon hidden insight that can be uncovered from the voluminous data dispersed in different silos.

__62. The business analyst takes the aggregated historical transactional data from our website and displays a dashboard of declining revenue and number of daily traders. We will see this in the lab exercise entitled Collect `Stock – Opening Bell` business dashboard.

__63. As part of this exercise, you log in to the `Stock Trader – Before` application and work with the application to understand the workflow and service that we provide to our customers.

__64. In the Work Map Index – the `Collect` process is described as below:

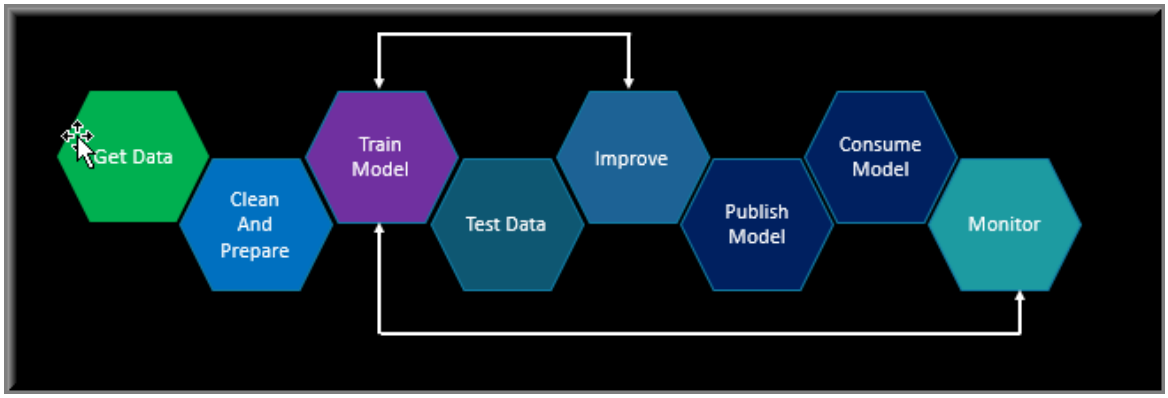


8.4 Organize

__65. The purpose of the `Organize` lab is to show how ICP-D helps to govern data and make it accessible to business analysts and data scientists. One of the biggest challenges companies face when trying to govern their data is to define the inventory of their data. To accomplish this, companies need to execute the painstaking process of looking at all the columns in all of the database tables, understanding the quality of what is in each column, correctly mapping business terms to each column, and making these business terms (and their respective columns) available to business analysts and data scientists in a secure way.

__66. Once the right data is located, the next step is to combine multiple sources into one file. It has been stated that “Simple models based on a lot of data is better than more complex models based on less data.” Said differently leveraging all available, relevant data sources will produce the best performing model

__67. ICP-D helps to accelerate these challenges. The Organize Category has a very important role. The repetitive workflows of `Find` ⇨ `Classify` ⇨ `Quality Score` ⇨ `Load to Catalog`, and `Get Data` ⇨ `Combine Sources` ⇨ `Transform` ⇨ `Load`



__68. A model can be made successful one time with herculean efforts, but this process is not scalable without a strong `Organize` capability. A model can only be as good as the quality of data that it uses.

__69. Building a good model requires a great deal of time and effort. The `Organize` capability of ICP-D helps you to scale and govern the process and ensures your modeling pipeline is being fed by high quality data.

__70. In this lab exercise, we will go through the process:

- Gathering of data across disparate sources
- Classification of data
- Assignment of terms to data
- Application of quality rules to data
- Transformation of data

__71. The above process is critical for a successful machine learning model development. Without automation, most organizations struggle to successfully implement a clean, end-to-end machine learning pipeline.

__72. The success of the machine learning model depends on connecting to data, regardless of where it is stored. Classifying data manually is cumbersome. ICP-D provides an automated way of classifying the data based on more than 200 prebuilt classifications.

__73. Confidence levels are assigned to the data based on automated classifications. This automation process leads to a much faster way of getting to know the data and where it resides.

__74. Quality rules are applied to the data. For example: Whether or not a credit card number is actually chargeable or valid. Run those rules against the columns, if it matches with the classification of data and if it matches associated business terms.

__75. Most organizations have been unsuccessful in properly classifying the data simply due to the extensive manual efforts required. These efforts are reduced significantly in this platform.

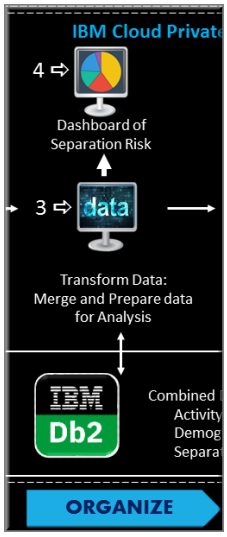
__76. The ICP-D platform assigns business `Terms` that exist already in the business glossary to the data that it has classified automatically to the data elements. This process is about simply assigning a column name of Tables to the descriptive names in the business glossary automatically.

__77. Finally - moving that data to the target using transformation capabilities so that it is available to data scientists.

__78. This capability of `Organize` is the strongest link for a successful and repetitive model deployment for the data scientist to `Search` and `Shop for Data` – a capability most organizations have been lacking due to the enormous amount of manual effort required.

__79. IBM studies with clients have shown a 300 percent average, 5-year ROI with adoption of automation in `Organize` process.

__80. In the Work Map Index – the `Organize` process is described as below:



__81. We will examine the process in detail as described above in `Organize` Lab.

8.5 Analyze

The purpose of the analyze lab is to show how ICP-D helps data scientists can: accomplish their job using language and their tooling of choice, improve their access to trusted, governed data sources, and be positioned with additional tooling to make their models available in a production environment for consumption by application developers.

	Note: We are using a Jupyter Notebook with a Python runtime to build the machine learning model for this workshop. ICP-D also supports other user-friendly GUI interfaces for both coders and non-coders R-Studio, Scala, Spark, Zeppelin with Anaconda and more.

__82. Once the data has been collected and organized properly, with search and classification capabilities, it is much easier for data scientists to shop for the data.

__83. The ICP-D platform reduces that 80 percent manual labor that is spent on data preparation to 10-20 percent.

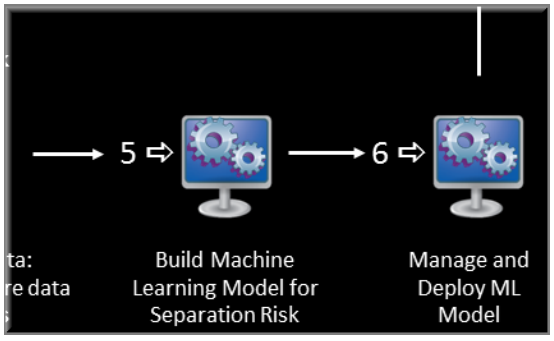
__84. In the Analyze lab exercise, we will develop, test and validate a machine learning model to better predict which customers are at the highest risk to separate from us.

- The model is built using Jupyter notebook and imports the merged customer table which was a result of hybrid data coming from relational and Hadoop data platforms.
- The data for the model will be prepared for the machine learning algorithm.
- 'Separation Risk' will be identified as the critical target.
- The proper variables will be identified as input predictors and a supervised learning algorithm using data as inputs will be executed.
- The model will be tested and validated for performance ensuring that we select the best model.
- The model will be saved to the model repository.

__85. The ICP-D platform provides an integrated platform for data scientists to work together and collaborate instead of each data scientist working on their own set of analytical tools without strong governance of the data.

__86. Collaboration using GitHub or BitBucket is integrated into the platform, which brings a cohesiveness to the work culture and helps to automate the CICD pipeline.

__87. In the Work Map Index – the `Analyze` process is described as below:



__88. We will examine the process in detail as described above in `Analyze` Lab.

8.6 Deploy

__89. The purpose of the Deploy lab is to make the model available in a production environment for application developers to consume. We will enhance our Stock Trader application to consume the model. Here you will work with the '`Stock - After`' application and witness our new AI capability.

Note: The final result of the end-to-end, integrated data platform with machine learning capabilities is demonstrated by consuming the REST API into our existing Stock Trader application.

__90. When a user logs into the application, in real time, we look up the current demographics of the user from our master file, run the demographics through our machine learning model to get a 'churn prediction' score, then based on that score, we provide the user with one of the following four retentions offers

- No offer
- Next 3 trades free
- Next 5 trades free
- Next 5 trades free and a free wealth advisor consultation

__91. As a company, we are differentiating ourselves by offering individualized retention offers in real time on our website. Based on our cloud-native platform, we can easily scale this application based on user/business need and can support tens or hundreds of thousands of users.

__92. The final result of the end-to-end integrated data platform with ML/AI capabilities are demonstrated by consuming the REST API into our existing Stock Trader application.

__93. In this `Deploy` lab exercise, we will show:

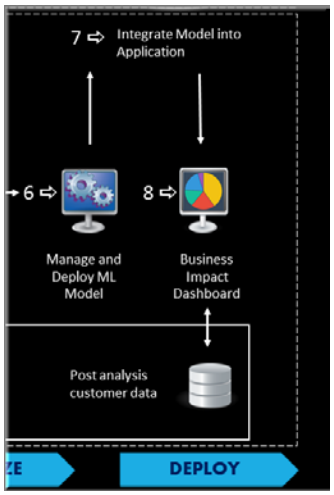
__a. Integrate model into Stock Trader Application

__b. Predict separation risk for high value customers through the REST API's endpoint

__c. Customized promotions offered to customers based on their loyalty and separation risk

__94. The impact of the model deployment to the business is then highlighted. The business analyst takes the data after a three-month period and analyzes the impact of embedding AI into the Stock Trader application model. They see that it has led to a significant change in our business' overall performance.

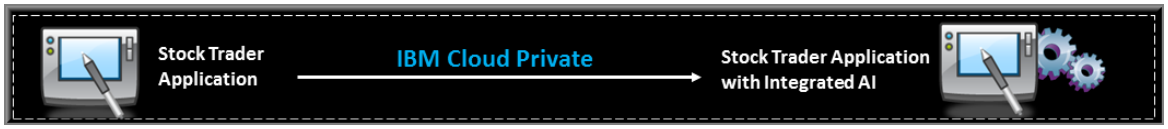
__95. In the Work Map Index – the `Deploy` process is described as below:



8.7 Business Impact

__96. The purpose of our last lab is to provide a dashboard quantifying the business impact of our project over time. We will turn the clock forward 90 days and measure business impact. Specifically, how many retention offers were made, how many customers took advantage of them, what was the cost of the offers (that is lost revenue for free trades and free wealth consulting sessions) AND what was the incremental revenue we brought to our company by significantly improving our 'customer separation' problem.

__97. The goal of this workshop has been to demonstrate analytics modernization and enhance applications with machine learning models, which drive our organization's journey to cloud and AI.



Appendix C. Integration of Hadoop Cluster with ICP-D

__98. Note: These instructions are for IT professionals who will integrate their ICP-D with a Hadoop cluster.

__99. Data virtualization is the key strength of IBM Cloud Private for Data.

__100. The Collect of ICP-D is to get data regardless where it resides. If data is in a Hadoop cluster of either Hortonworks or Cloudra, it can be accessed without actually moving it.

__101. The integration of ICP-D with Hadoop cluster will be useful for the following tasks:

- __a. Accessing the HDFS file system to read and write files
- __b. Connecting to the Hive database to access table data
- __c. Using Big SQL on the Hortonworks data platform to access table data
- __d. Push down Spark processing to the Spark2 server on the HDP

8.8 Access to Hadoop Behind Firewall

__102. When the Hadoop services behind a firewall, the authentication via Knox and/or Kerberos is implemented. In such situation, it is recommended to implement the ICP-D Hadoop Integration package (DSX-HI) on one of the edge nodes of the cluster.

__103. In a secure Hadoop cluster, only authentication services are exposed to the outside world. The insecure services such as native HDFS, WebHDFS, WebHCat, Livy, and so on. are behind a firewall and this can only be accessed through a DSX-HI gateway.

__104. The DSX-HI gateway is built on Apache Knox.

8.9 Download Software

__105. The Hadoop Integration software can be downloaded from Passport Advantage. At the time of writing of this lab document, we used ICP 3.1 with ICPD 1.1.0.2. The part number of the *Hadoop integration software for IBM Cloud Private for Data 1.1.0.2* is `CNW0BEN`.

__106. In our lab environment, Hortonworks Sandbox (Hadoop VM) is available in the tenth VM.

__107. Log in to the Hadoop VM and use `root` password: `hadoop`.

```
# ssh 192.168.142.110
```

8.10 Prepare Hadoop Cluster for ICP-D Integration

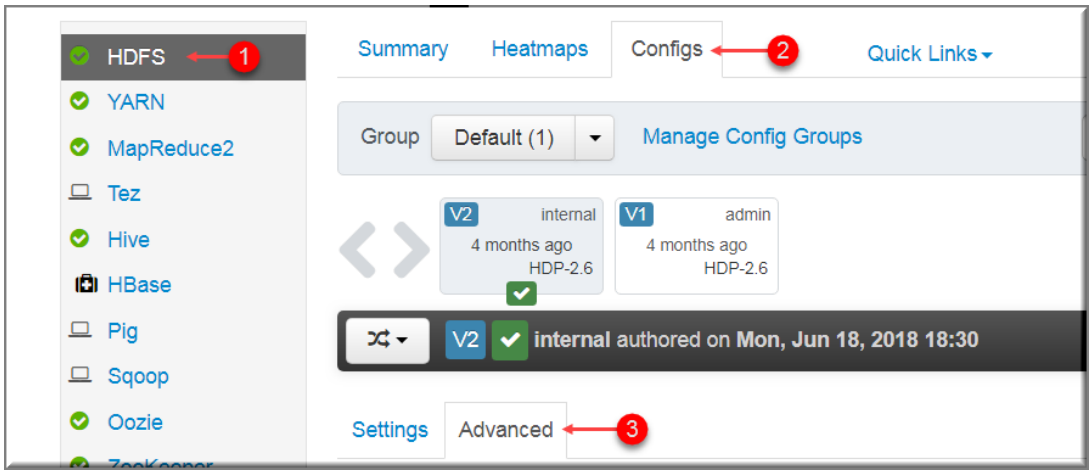
__108. The DSX-HI services connect to the Hadoop cluster with special privileges to impersonate other users to ensure that YARN jobs are submitted on behalf of the user and that HDFS authorization settings are respected. We will use user `dsxhi` as a proxy user to do jobs on behalf of requests coming from ICP-D.

__109. Log in to the Hortonworks Sandbox. Open URL `http://192.168.142.110:1080` using the Chrome browser from VM `icp01`.

__110. Click `Launch Dashboard`.

__111. Login using Username: `admin` and Password: `password`.

__112. Click `HDFS ⇨ Configs ⇨ Advanced`.

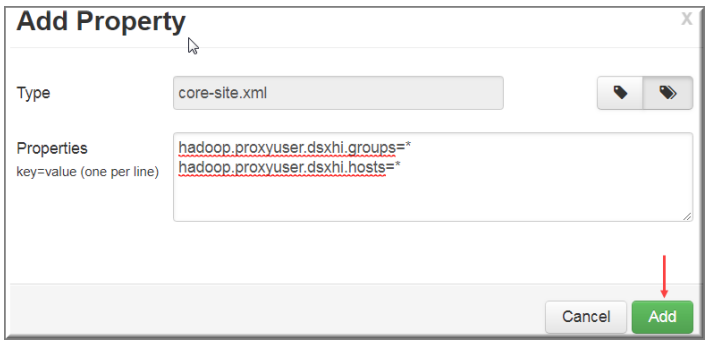


__113. Scroll down to `Custom core-sites` and expand it. Go to the end of property list and click `Add Property`.

__114. Type in the box, the following properties.

```
hadoop.proxyuser.dsxhi.groups=*
hadoop.proxyuser.dsxhi.hosts=*
```

__115. Click `Add`.



__116. Scroll to the top and click `Save` twice and click `Proceed Anyway` and click `OK`.

__117. Click `Hive ⇨ Configs ⇨ Advanced`.

__118. Scroll down to `Custom webhcat-site.`

__119. Expand it and go to the end of list and click `Add Property`.

__120. Add the following two lines in the Properties box and click `Add`.

```
webhcat.proxyuser.dsxhi.hosts=*
webhcat.proxyuser.dsxhi.groups=*
```

__121. Scroll to top and click `Save` twice and click `Proceed Anyway` and click `OK`.

__122. Click `Spark2 ⇨ Configs` and scroll down to `Custom livy2-conf`. Expand it and click `Add Property`. Add the following line and click `Add`.

```
livy.superusers=dsxhi
```

__123. Scroll to top and click `Save` twice and click `Proceed Anyway` and click `OK`.

8.11 Create `dsxhi` User

__124. Switch back to the command line window in the Hadoop VM to run the following commands as root.

__125. Login to the Docker container `sandbox-hdp-security`

```
# docker exec -it sandbox-hdp-security bash
```

__126. Run the following to create dsxhi user

```
# useradd -u 3001 -g hdfs dsxhi
```

__127. Create `dsxhi` HDFS home directory and change permissions.

```
# hdfs dfs -mkdir /user/dsxhi
# hdfs dfs -chown dsxhi:hdfs /user/dsxhi
```

__128. Create a user for every user registered in ICP-D. In our case, we are using only one user `admin`.

```
# useradd -u 3002 admin
```

__129. Change permissions.

```
# hdfs dfs -chown admin:admin /user/admin
```

8.12 Install Software in the Docker Container

__130. You can `scp` RPM to the Docker container or put it on any web server and use `rpm -ivh http:..` to install it.

__131. For example: I copied the RPM to my local web server at https://www.zinox.com/wp-content/uploads/dsxhi-icpdata-ee-1.0.1.0-49.noarch.rpm

__132. Install dsxhi on Hadoop edge node.

```
# rpm -ivh dsxhi-icpdata-ee-1.0.1.0-49.noarch.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:dsxhi-1.0.1.0-49                 ################################# [100%]
```

8.13 Create dsxhi Configuration File.

__133. Switch to conf directory.

```
# cd /opt/ibm/dsxhi/conf
```

__134. Copy the Hortonworks template

```
# cp dsxhi_install.conf.template.HDP dsxhi_install.conf
```

__135. Modify the config file to specify service user, webchat URL, ICP-D server name and so on, in dsxhi_install.conf.

__136. The dsxhi_install.conf as per our configuration is as shown.

```
# sed -e '/^$/d' -e '/^#.*$/d' dsxhi_install.conf
dsxhi_license_acceptance=a
dsxhi_serviceuser=dsxhi
dsxhi_serviceuser_group=hdfs
dsxhi_gateway_port=8443
dsxhi_rest_port=8082
cluster_manager_url=http://192.168.142.110:8080
cluster_admin=admin
exposed_hadoop_services=webhdfs,webhcat
existing_webhcat_url=http://192.168.142.110:50070
dsxhi_livyspark_port=8998
dsxhi_livyspark2_port=8999
known_dsx_list=https://192.168.142.101:31843
package_installer_tool=
packages=lapack
cluster_nodes=
cluster_ssh_user=
cluster_ssh_key_path=
```

8.14 Install DSXHI

__137. Save the existing keystore

```
# cd /etc/pki/ca-trust/extracted/java
# mv cacerts cacerts.orig
```

__138. Install DSXHI

```
# cd /opt/ibm/dsxhi/bin
# ./install.py --dsxhi_gateway_master_password=password --
dsxhi_java_cacerts_password=password --password=password
```

__139. The installation does prerequisite check and if they pass, it will configure knox gateway and start all services.

```
IBM Cloud Private for Data Enterprise Edition Hadoop Integration Service
-------------------------------------------------------
Terms and Conditions: http://www14.software.ibm.com/cgi-
bin/weblap/lap.pl?la_formnum=&li_formnum=L-KMRY-
B2632F&title=IBM+Cloud+Private+for+Data+-
+Enterprise+Edition+V1.1.0.1+(bundles+ICP+Foundation)&l=en
--Determining properties
--Running the prechecks

--Configure gateway
--Create template for gateway
--Install dsxhi_rest
--Start all services
--Setting up known DSX Local clusters
--Install finished. Check status in /var/log/dsxhi/dsxhi.log
```

__140. Check if Gateway, HDFSWeb and file system services ports are in use.

```
# netstat -nlp | grep 8443
tcp        0      0 0.0.0.0:8443      0.0.0.0:*   LISTEN      75285/java
# netstat -nlp | grep 50070
tcp        0      0 172.18.0.2:50070  0.0.0.0:*   LISTEN      6772/java
# netstat -nlp | grep 8020
tcp        0      0 172.18.0.2:8020   0.0.0.0:*   LISTEN      6772/java
```

__141. The modified knox gateway should be used and the default knox gateway should be kept disabled.

8.15 Create DSXHI Systemd Service

__142. The dsxhi gateway should start automatically whenever the Hadoop edge (gateway) node starts.

__143. Create a service as per the following.

```
cat << 'EOT' > /usr/lib/systemd/system/icpdhi.service
[Unit]
Description=ICPD DSXHIO Service
After=ambari-server.service ambari-agent.service mysql.service dbus.service

[Service]
Type=oneshot
ExecStart=/opt/ibm/dsxhi/bin/start.py
TimeoutStartSec=1200
```

```
[Install]
WantedBy=multi-user.target
EOT
```

__144. Enable the service so that the gateway is started automatically whenever the Docker container is restarted.

```
# systemctl enable icpdhi
```

8.16 Commit Docker Container to Save the Installation

__145. Exit from container

```
# exit
```

__146. Commit the image with same tag.

```
ID=$(docker ps --format '{{.ID}}' --filter Name=sandbox-hdp-security)
IMAGE=$(docker ps --format '{{.Image}}' --filter Name=sandbox-hdp-security)
docker commit $ID $IMAGE
```

__147. The image is saved and the `dsxhi` install will not disappear if the container is stopped or removed.

8.17 Troubleshooting

__**148.** When the registration of the Hadoop edge node fails, **Administer ⇨ Hadoop Integration in ICP-D Web UI**.

__149. The reason for this failure is the ICP-D `utils-api` pod is unable to resolve the name of the Hadoop edge node.

__150. Make sure that the name of the Hadoop edge server is defined and resolvable through the name server that ICP-D is accessing.

__151. The troubleshooting can be done by looking at the log files from the `utils-api` pod in ICP-D and the log files from the Hadoop edge node.

```
/var/log/dsxhi/gateway
/var/log/dsxhi
```

__152. Commands to log onto Hadoop sandbox in our cluster.

```
# ssh Hadoop
# docker exec -it sandbox-hdp-security bash
# cd /opt/ibm/dsxhi/bin
```

__153. The commands `start.py`, `status.py` and `stop.py` can be used to start, status and stop the `dsxhi` gateway server.

__154. Run command `manage_known_dsx.py --list` to find the service URL that we need to use in ICP-D.

```
# ./manage_known_dsx.py --list
DSX Local Cluster URL           DSXHI Service URL
https://192.168.142.102:31843   https://sandbox-hdp.hortonworks.com:8443/gateway/192.168.142.102
https://192.168.142.101:31843   https://sandbox-hdp.hortonworks.com:8443/gateway/192.168.142.101
```

__155. Check the logs of the `utils-api` pod

```
# kubectl -n zen get pods  | grep utils
utils-api-5966956c44-t5qcn   1/1   Running   0  1h
# kubectl -n zen logs utils-api-5966956c44-t5qcn
```

Credit: Based on document prepared by Frank Ketelaars, Technical Lead for IBM Cloud Private for Data.

**** End of Appendix:**

Appendix AA. 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
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

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

Appendix BB. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	Db2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	Tivoli	WebSphere	Workplace	System p	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

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

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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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

NOTES

NOTES

IBM Software