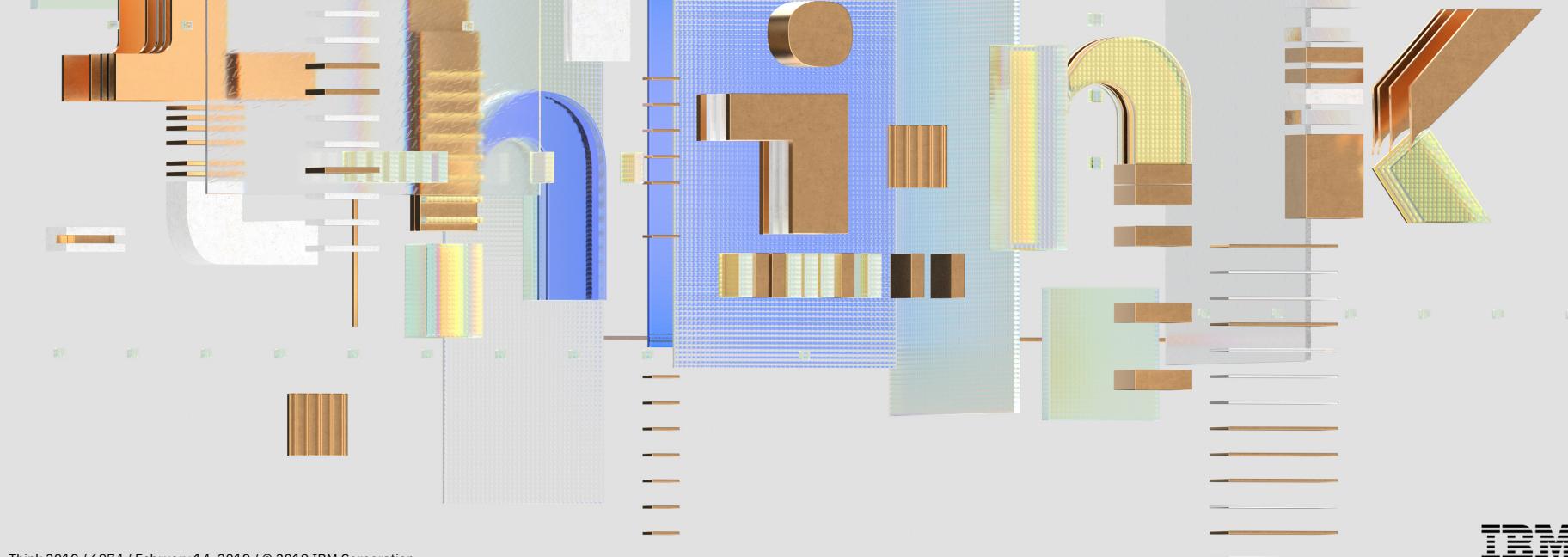


Watson Studio Architecture, Design and Innovation

Emmanuel Génard – genard@fr.ibm.com

Cloud Developer Advocate Europe,
IBM Business Solution Center Nice, France



Agenda

Presentation

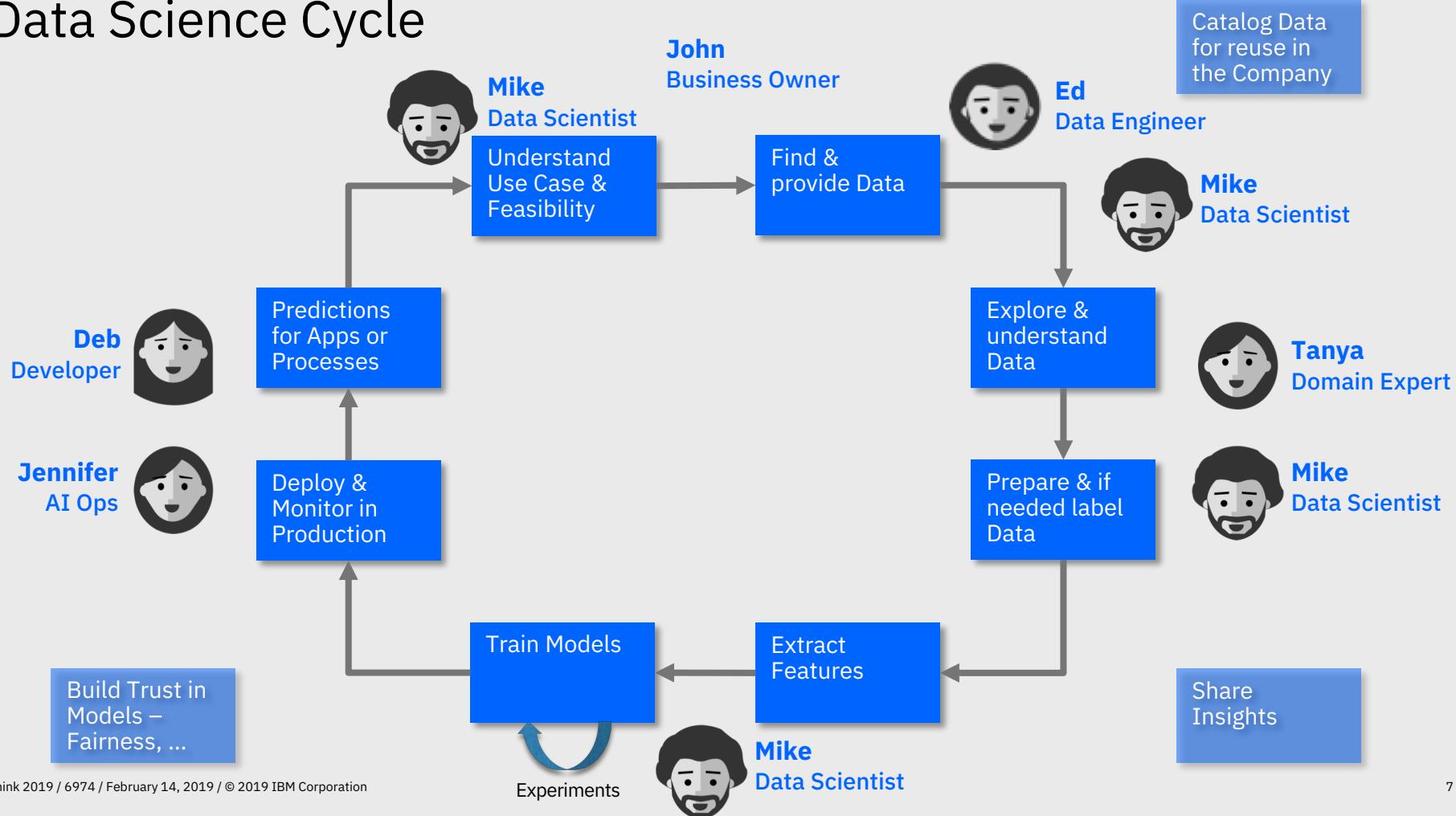
- Overview and Architecture
- Design – How it works for users
- Auto AI – Coming soon
- Architecture Details

Labs

- Github repo: ibm.biz/WatsonStudio4e-i
- Wrap-up

Overview and Architecture

Data Science Cycle



Data Science and AI Teams building AI

Need to be enabled for productive collaboration on Insights and Models



Tanya
Domain Expert

Job:
Contribute domain expertise,
label data, images, text, ...

What she does:

- Shares knowledge with data scientists
- Helps guide analysis of data and training models
- Labels data to provide training data

Sometimes known as:
Subject matter expert



Mike
Data Scientist

Job:
Transform data into knowledge and
models for solving business problems.

What he does:

- Analyzes data to gain insights
- Runs experiments to build models
- Use Machine Learning techniques
- Works with experts to validate models

Sometimes known as:
ML/DL engineer, Modeler, Data Miner



Ed
Data Engineer

Job:
Architects how data is organized and
ensures operability

What he does:

- Builds data infrastructure, ETL pipelines, ...
- Works with Data Flows, Spark, Hadoop, and HDFS, ...

Sometimes known as:
Data infrastructure engineer

Business Owners, Ops, Developers integrating AI into the Business

Need managed scalable Deployment under Control of IT and Business



Jennifer
AI Ops

Job:
Responsible for deploying and running models in production at scale

What she does:

- Based on business direction, create deployments of models and related assets
- Work with data science teams to obtain models and assets.
- Make sure models perform and scale in production environment



John
Business Owner

Job:
Responsible for applications or business processes in his line of business

What he does:

- Initiates creation of Models based on business requirements
- Has his subject matter experts work with data scientists in creating models
- Determines what models are to be used by what applications / business processes over what time periods



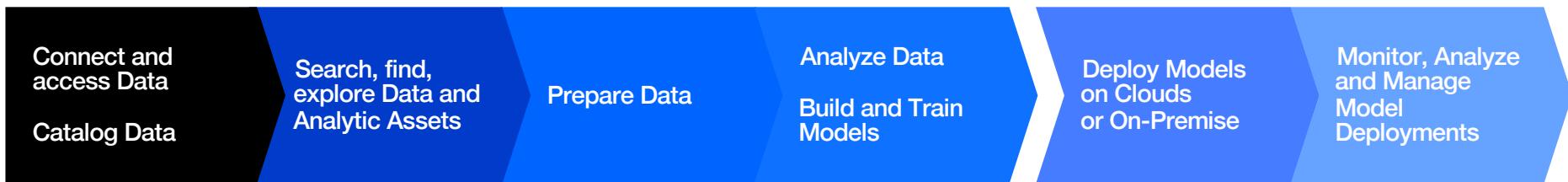
Deb
The Developer

Job:
Builds application or process flows that use AI to meet the business requirements

What she does:

- Build apps for her company or its customers to use
- Build out business processes
- Connect apps or processes to AI by invoking models

Integrated Workflow with Watson Studio + Watson Machine Learning



Create a **Project**.

Connect and discover data from many data sources on public or private cloud.

Catalog data to share across the enterprise.

Find data and analytic assets such as flows, notebooks, models, dashboards, ... in the **Catalog** through an explorative UI honoring policies

Clean and prepare data using **Refinery** to create data flows visually, or **Notebooks/Scripts** for data prep through code.

Use popular open source libraries to prepare structured or unstructured data.

Analyze data using **Dashboards** or **Notebooks**.

Create & train models through **Model Builder**, **Flows**, or through code in **Notebooks**, using popular **open source** ML/DL frameworks and IBM tools.

Train at scale with ability to use **GPUs** and **distributed** compute.

Deploy models for production and **scale automatically** for online, batch or streaming use cases.

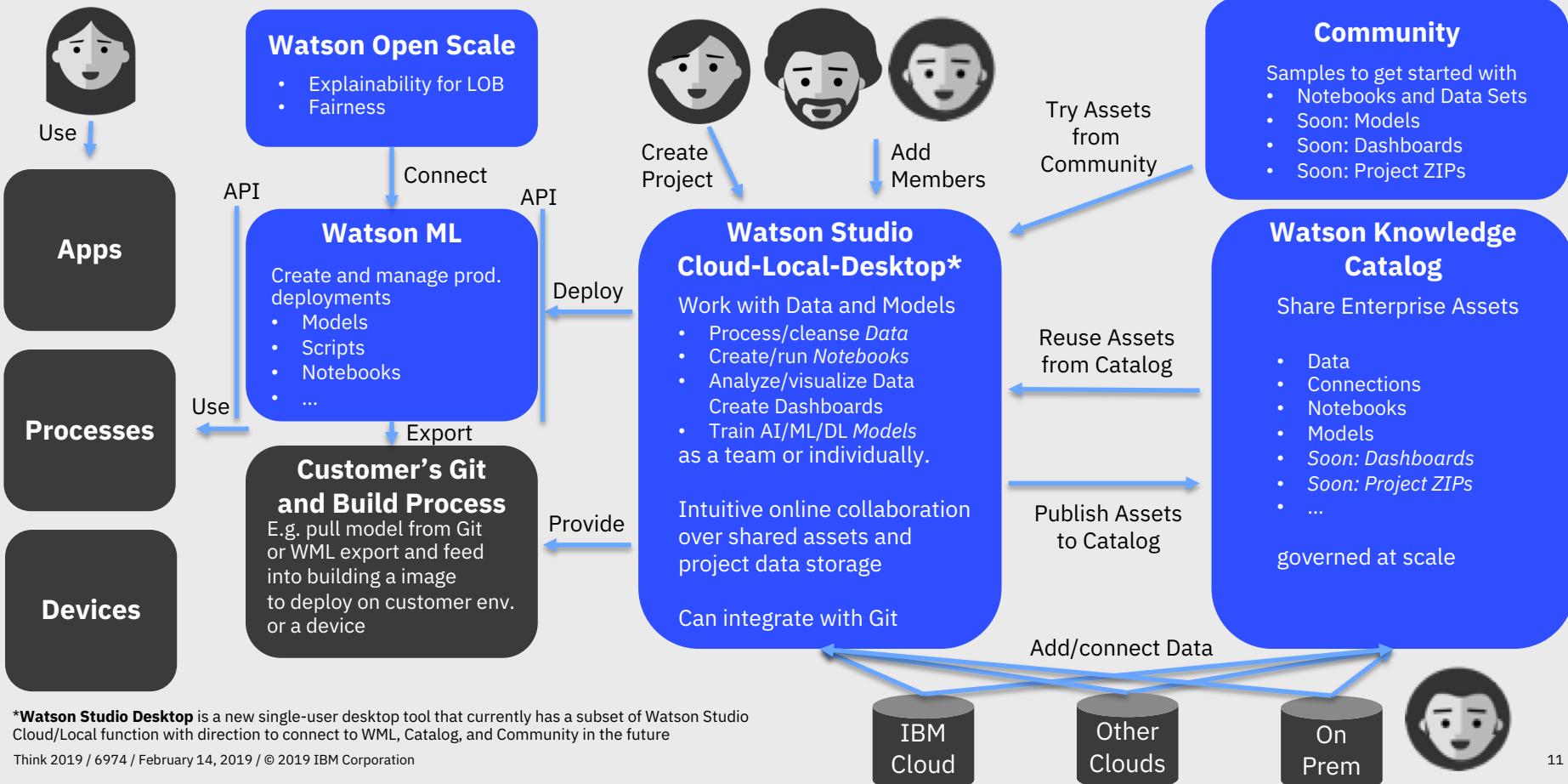
Typically done by AI Ops staff, not by data scientists.

Monitor performance of models in production.

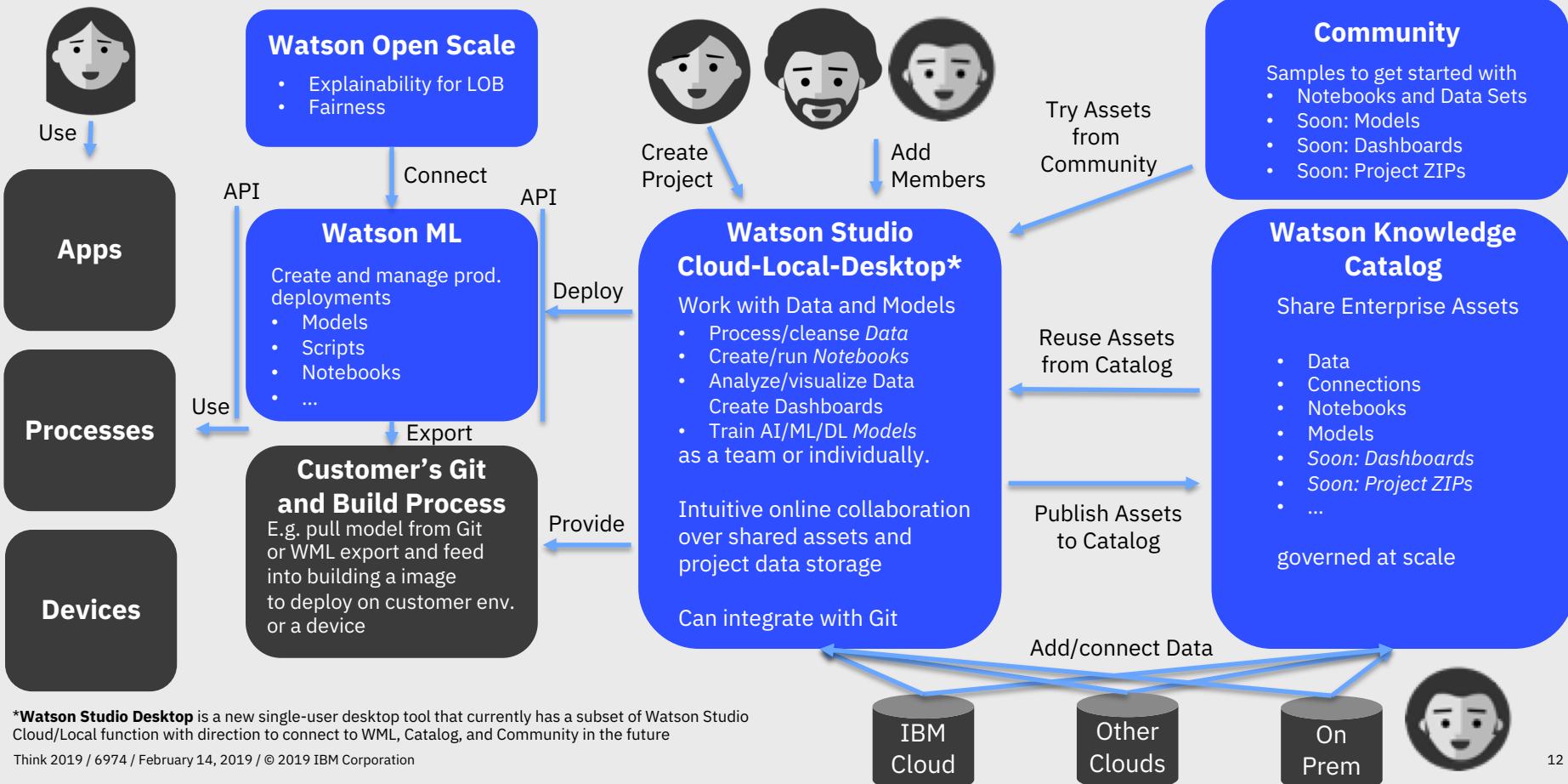
Can trigger automatic retraining and redeployment of models if desired.

Build **Enterprise Trust** with Bias Detection, Fairness, ... using **Watson Open Scale**.

Watson Studio and other IBM Data and AI Capabilities



Watson Studio and other IBM Data and AI Capabilities



***Watson Studio Desktop** is a new single-user desktop tool that currently has a subset of Watson Studio Cloud/Local function with direction to connect to WML, Catalog, and Community in the future

Watson Studio

- **AI & data science as a team effort** - Lets AI experts, data scientists, analysts, stakeholders collaborate to collect, share, explore, analyze data to derive insights, train models, and share/deploy resulting assets
- **Projects** provide a secure environment in which teams or individuals work with and analyze data using
 - **Connect** cloud and on prem data sources
 - **Refine** – clean and shape data for analysis/ ML
 - **Dashboards** – visual analytics and sharing of insights
 - **ML/DL** – train with Spark ML, ScikitLearn, SPSS, ... and deploy to Watson Machine Learning service for prod
 - **Flows** – create flows running on Spark or SPSS
 - **Notebooks** – Jupyter + Spark int., comments, versions, share as link, Git integration, Remote Spark integration, ... now with flexible Environment options
 - **Decision Optimization** – solve complex problems
- **RStudio** integrated with Spark
- **Integration** with other IBM AI and Data capabilities
 - Integrates with Watson Knowledge Catalog, Watson Machine Learning, and Watson Open Scale
 - Integrates with Watson AI Services & IBM Analytics Engine
 - Integrates with IBM/third-party cloud & on-prem data services

Try it at <https://www.ibm.com/cloud/watson-studio>
Now available in Dallas, Frankfurt, Tokyo, London

The screenshot shows the IBM Watson Studio web interface. At the top, there's a navigation bar with links for Projects, Catalog, Community, Services, Docs, Support, Manage, and a user profile. A "Get started" button is also visible.

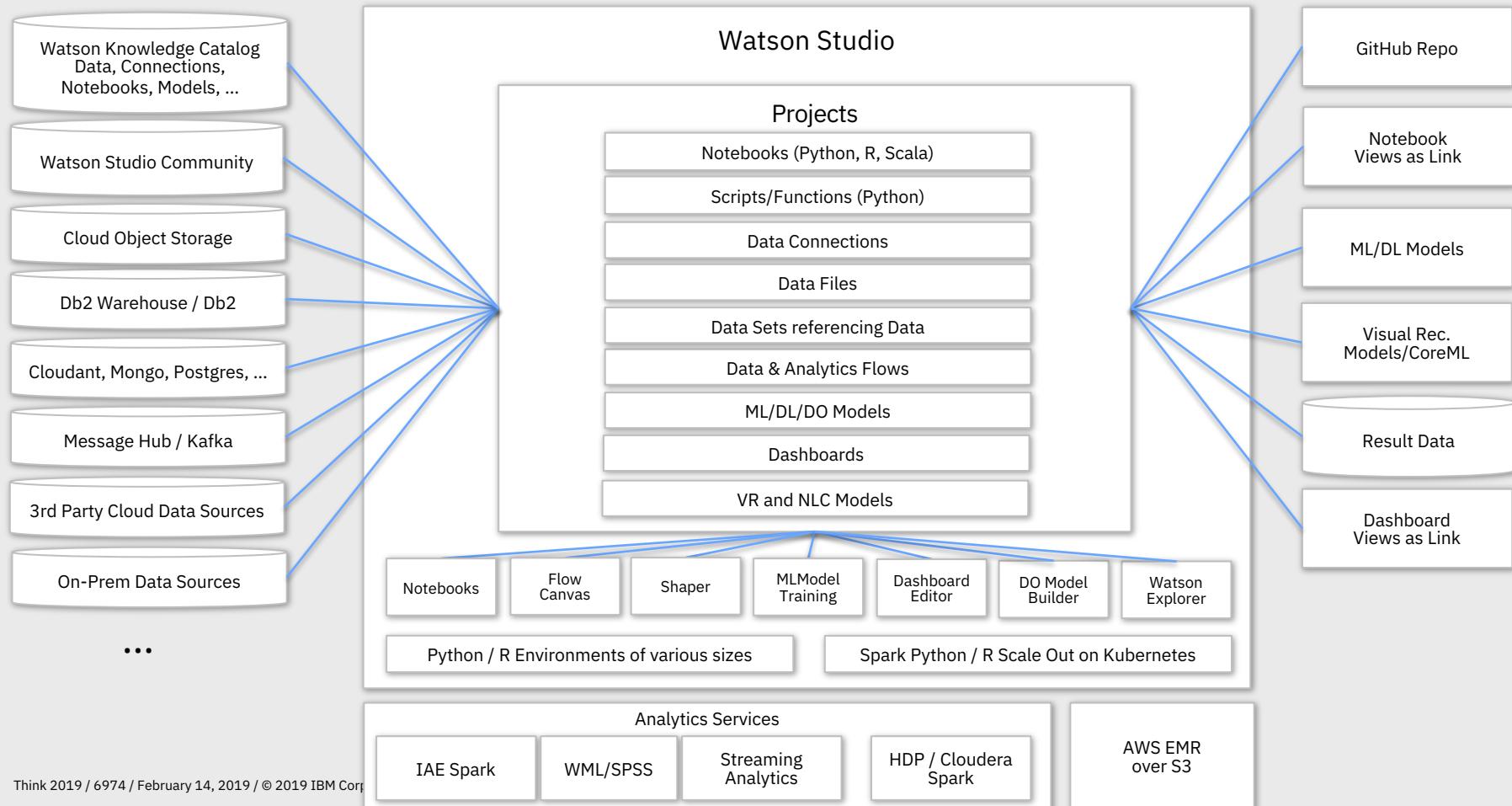
Recently updated projects: This section displays a table of recent project entries. Each entry includes the project name, role (Admin), number of collaborators, date created, and last updated.

NAME	ROLE	COLLABORATORS	DATE CREATED	LAST UPDATED
Project_lib_for_V1projects	Admin	2	Nov 09, 2017	Dec 17, 2018
Watson_Studio_Think_20180319	Admin	31	Feb 07, 2018	Dec 14, 2018
Natural Language Classifier Project	Admin	1	Dec 13, 2018	Dec 13, 2018
DO Project 1	Admin	1	Dec 08, 2018	Dec 08, 2018
Auto Insurance Claims	Admin	90	Mar 15, 2018	Dec 07, 2018
Test 2932386	Admin	1	Dec 03, 2018	Dec 03, 2018

Your catalogs: This section shows a list of catalogs. Each catalog entry includes the creator, date created, and a brief description.

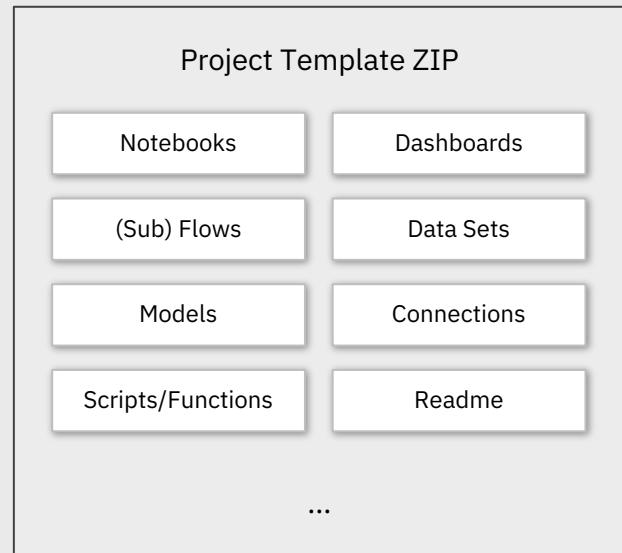
Creator	Date created	Description
watsondemo	Mar 14 2018 11:50 PM	Catalog for Think 2018 Conference demos
Ajay Gupta	Jun 11 2018 7:34 AM	Catalog Created in Armada Production for testfest.

Projects - the Place for Users to work and collaborate



Project Templates – coming soon

- Project Templates are collections of assets such as notebooks, models, dashboards, sample data, ...
- Creators of Templates:
 - IBM and Partners and Academia for Project Templates in the Community
 - Customers and their partners for Project Templates in their Catalogs
- Users can create Project Templates from their projects or create projects or add assets to projects from project templates
- Project Template Files are ZIP files following conventions for packaging of project assets and metadata
- Project Templates can be shared in *Community* and be re-used by users to create their projects
- Project Templates can be shared in the *Catalog* and can be re-used from the catalog

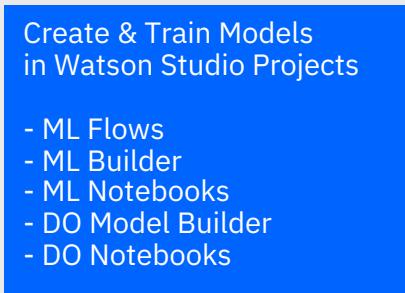


Training ML Models → Operationalization in Prod. Deployments

Data Scientists and Subject Matter Experts

Create & train Models

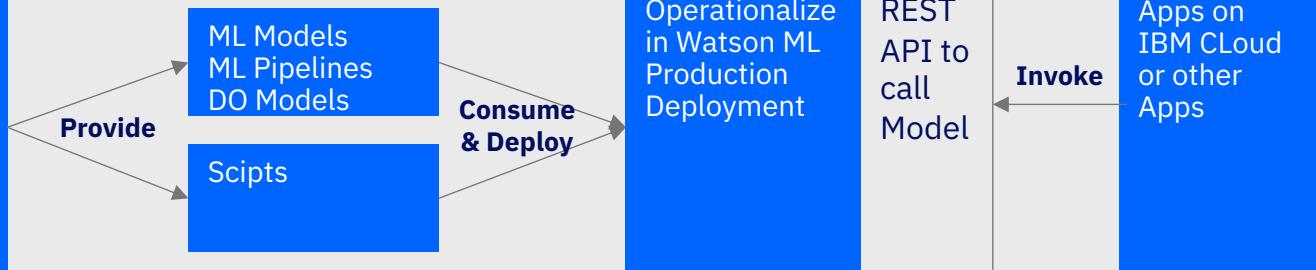
Make them available to be consumed by IT



IT / Application Owners

Get Models and related assets

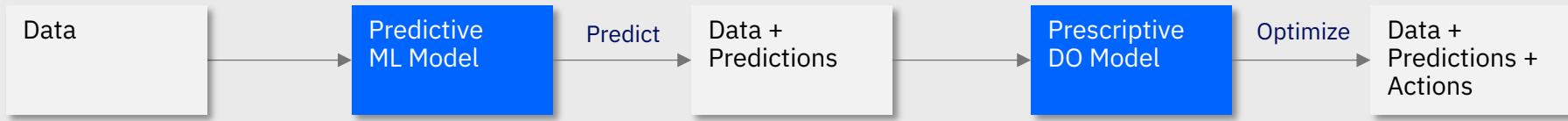
Deploy and run in Production



- **Data Scientists** can create, train, validate ML models in Projects
 - Use ML Builder for guided creation and training of models using common patterns and algorithms
 - Use Notebooks or Flows to train models for more advanced use cases with more flexibility
- **Operations team** can deploy models to production
 - Use REST API to invoke your models for online scoring / predictions
 - Batch scoring of data e.g. stored in relational database tables

Analyze → Predict → Optimize

Combining Machine Learning and Decision Optimization

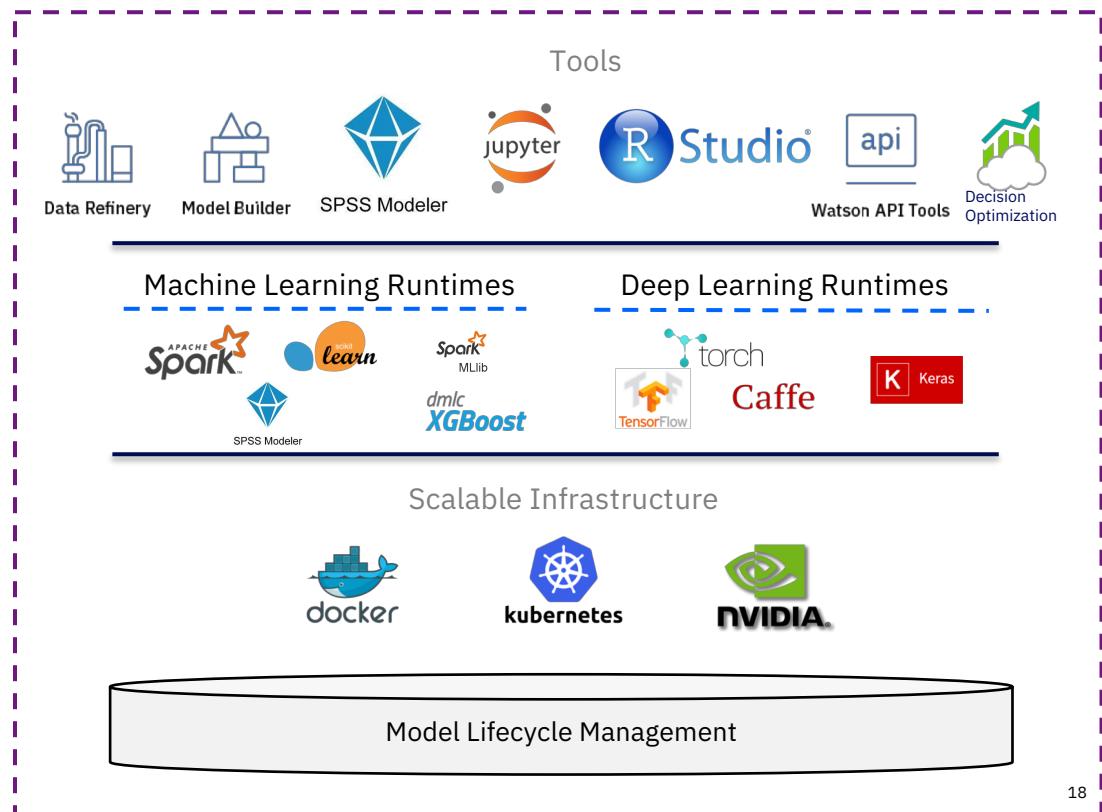


- **ML can make useful predictions**, e.g. which customers are likely to buy certain products, which parts are likely to fail within a month, which customers are at risk to churn, ...
- **However, actions on predictions typically are constrained**, e.g. limited marketing and sales budget, limited maintenance staffing, limited budget for discounts, limited time, ...
- → **Often there is a need to optimize actions under given constraints to achieve the best outcome**
- Combining **Machine Learning + Decision Optimization** allows to **predict and optimize to take the best actions**
- Example: <https://dataplatform.cloud.ibm.com/exchange/public/entry/view/37d8b0ccf1abb870fde60a6ccdd2ec63>

Watson Studio

Pre-Integrated Tools, Algorithms, Libraries for Data Science, ML/DL

- Best of breed open source & IBM tools
- Code (R, Python or Scala) and no-code/visual modeling tools
- Most popular open source frameworks
- IBM best-in-class frameworks
- Container-based resource management
- On IBM Cloud: Elastic pay as you go CPU/GPU use



Design – How it works for Users

Try it at <https://www.ibm.com/cloud/watson-studio>

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with links for Projects, Catalog, Community, Services, Docs, Support, and Manage. On the right side of the header, there's a user profile for 'Thomas Sch.' and a 'Get started' button.

The main area features a large blue banner with the text 'Welcome Thomas!' and 'Watson Studio • Watson Knowledge Catalog'. Below the banner, there are two main calls-to-action: 'Create a project' and 'Search a catalog'. The 'Create a project' section includes a sub-instruction: 'A project is how you organize your resources to work with data and collaborate with team members'. To the right of these buttons is a large circular icon containing a magnifying glass over a computer monitor, symbolizing data analysis.

Below the main banner, there's a section titled 'Recently updated projects' with a link to 'View all (90)'. It includes a 'New project' button with a plus sign. A table lists three recent projects:

NAME	ROLE	COLLABORATORS	DATE CREATED	LAST UPDATED
Watson_Studio_Think_20180319	Admin	31	Feb 07, 2018	Dec 19, 2018
Project-lib_for_V1projects	Admin	2	Nov 09, 2017	Dec 17, 2018
Natural Language Classifier Project	Admin	1	Dec 13, 2018	Dec 13, 2018

A blue speech bubble icon is located in the bottom right corner of the main content area.

Watson Studio Community

- Get started with Notebooks, Data Sets, and Tutorials that you can try in Watson Studio and learn from
- Use filters and search to find what you are interested in
- Add Notebooks from the Community directly to your Project ready to run
- Give feedback
- If you want to contribute assets as a partner, let us know

IBM Watson Studio Projects Catalog Community Services Docs Support Manage

Machine Learning X Decision Optimization X Notebook X Data Science X What are you looking for?

Popular filters: Spark Deep Learning Brunel

Sort by: Most Related

Search results (90)

Notebook Title	Author	Date	Topic	Views	Actions
A TensorFlow regression model to predict...	IBM	Apr 06, 2018	Economy & Business	17	
Access bb2 Warehouse on Cloud and bb2 with...	IBM	Jun 26, 2018	Economy & Business	27	
Access MySQL with Python	IBM	Mar 27, 2018	Transportation	22	
Access MySQL with R	IBM	Mar 27, 2018	Transportation	12	
Access PostgreSQL with Python	IBM	Mar 20, 2018	Transportation	4	
Access PostgreSQL with R	IBM	Mar 20, 2018	Transportation	3	
Analyze Facebook Data Using IBM Watson and...	IBM	Mar 20, 2018	Economy & Business	17	
Analyze accident reports on Amazon EMR Spark	IBM	Oct 12, 2017	Transportation	6	
Analyze data, build a dashboard with Spark...	IBM	DATE	DATE	1	
Analyze open data sets with Spark & PixieDust	IBM	DATE	DATE	1	
Analyze open data sets with pandas DataFrames	IBM	DATE	DATE	1	
Analyze precipitation data	IBM	DATE	DATE	1	

IBM Watson Studio Projects Catalog Community Services Docs Support Manage

In [5]: # Visualize Price vs. Accommodates
options(repr.plot.width=15, repr.plot.height=4)
p1 <- ggplot(portland, aes(x = log(portland\$price), y = accommodates, color = room_type)) + geom_point(position = 'itter', shape = 1) +
 labs(x = "Nightly Rental Price (Log)", y = "Accommodates", title = "Nightly Price vs. Persons Accommodated") +
 theme_minimal()
p2 <- ggplot(portland, aes(x = log(portland\$price), color = room_type)) + geom_density(aes(fill = room_type, alpha = 0.5)) +
 labs(x = "Nightly Rental Price (Log)", y = "Density", title = "Price Density by Room Type") + theme_minimal()
grid.arrange(p1, p2, ncol = 2, nrow = 1)

The above plots show us that indeed, larger accommodations generally fetch a higher price. What about location? Here are the average prices for a number of neighborhoods with at least 30 listings.

In [6]: pAggs <- group_by(portland, neighbourhood_cleaned) %>%
 summarise(avr_price = mean(price), count = n()) %>%
 filter(count > 29) %>%
 arrange(desc(avr_price))

options(repr.plot.width=15, repr.plot.height=5)

ggplot(pAggs, aes(x = reorder(neighbourhood_cleaned, -avr_price), y = avr_price, fill = neighbourhood_cleaned)) +
 geom_bar(stat = "identity") +
 geom_text(label = paste("\$", round(pAggs\$avr_price))) +
 scale_x_discrete(label = abbreviate) +
 labs(x = "Neighborhood", y = "Average Nightly Rental Price (USD)", title = "Portland Rental Prices by Neighborhood")
 theme_minimal() +
 theme(panel.position = "none")

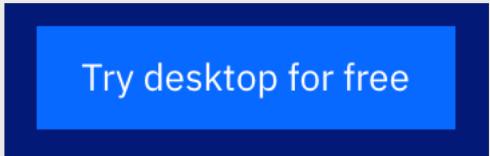
Portland Rental Prices by Neighborhood

1716985 - Thomas Sch... ▾

21

Watson Studio Desktop

- Allows a user to create Projects on their Desktop
- In their project, the user can
 - **Load Data**
 - **Preview and Shape Data**
 - **Flows** (SPSS) to train models or analyze data
 - Soon: **Notebooks** to train models or analyze data
 - Soon: **Decision Optimization** in Notebooks together with ML in Watson Studio Desktop
 - Soon: **Export Project ZIP file** that can be imported to WS Cloud or WS Local
- Free trial available through Watson Studio site at <https://www.ibm.com/cloud/watson-studio>



The screenshot shows the IBM Watson Studio Desktop application. At the top, there's a navigation bar with "IBM Watson Studio" and other options like "Edit", "View", "Window", and "Help". The top right corner shows system status like battery level (76%), signal strength, and the date/time (Mon 14:57).

The main interface has a sidebar on the left labeled "Projects" with "View All Projects" and a list of "Test Project 1" items: T3, Test 343443, Test 980928, and Test 23702938. Below the sidebar are "Support" and "Account" buttons.

The central area shows "Test Project 1" details: ASSETS 7, DATE CREATED Oct 11 2018, LOCATION View Folder. There are tabs for "Assets" (selected) and "Settings". A search bar says "What assets are you looking for?"

A section titled "Data assets" lists five files:

NAME	TYPE	LAST MODIFIED	ACTIONS
wine_data.csv	Data Asset	11 Oct 2018, 7:32:27 am	⋮
productsales.csv	Data Asset	11 Oct 2018, 7:31:53 am	⋮
precipitation.csv	Data Asset	11 Oct 2018, 7:31:47 am	⋮
cars-5.csv	Data Asset	11 Oct 2018, 7:31:12 am	⋮
sql_quick_overview.pptx	Data Asset	11 Oct 2018, 7:17:57 am	⋮

Below the data assets is a section titled "Modeler flows" with two entries:

NAME	TYPE	LAST MODIFIED	ACTIONS
Test Flow 1	SPSS	19 Oct 2018, 3:38:13 am	⋮
Drug Study Example	SPSS	11 Oct 2018, 7:27:50 am	⋮

At the bottom, there's a "Submit feedback" button and a "72%" progress bar. The bottom right corner shows the date/time (Mon 15:03) and a page number 22.

The bottom half of the screen shows the "Record Operations" palette on the left with options like Select, Sample, Sort, Balance, Distinct, Aggregate, Merge, and Append. To the right is a "Flow" diagram consisting of several nodes connected by arrows: DRUG1n → Na_to_K → Discard F... → Define T... → Drug. There are also multiple "Drug" nodes connected to the flow.

Work with Projects

- Browse or search for existing projects under *My Projects*
- Create new Projects based on project starter type
 - Standard Project - customize as you like
 - Data Science Project
 - Visual Recognition Project
 - Deep Learning Project
 - Modeler Project
 - Business Analytics Project
 - Data Engineering Project
 - Streams Flow Project
- You are now the administrator of your new project
- Regardless of based on which project starter type you created your project, you can always switch on more capabilities in your project

The screenshot shows the IBM Watson Studio interface. At the top, there's a navigation bar with links for Projects, Catalog, Community, Services, Docs, Support, Manage, and a user profile. Below the navigation bar is a search bar labeled "Find project by name" and a dropdown menu set to "All projects". On the left, there's a sidebar titled "My Projects" with a list of recent projects. The main area displays a table of projects with columns for Name, Role, Storage, Collaborators, Creator, Date Created, and Actions. The table lists various projects such as "Test 2932388", "Data Engineering", "Notebook Project", etc., each with its creator (e.g., Thomas Schaeck, Jihyoung Kim), role (e.g., Admin, Editor), storage type (e.g., COS), and collaborator count. On the right, there's a "New project" dialog box. It has tabs for "New project" (selected), "Define project details", and "Storage". In the "Define project details" tab, there's a "Name" field containing "Test 12345" and a "Description" field with the word "Test". In the "Storage" tab, it says "Cloud Object Storage-s4-lite". Under "Choose project options", there's a checked checkbox for "Restrict who can be a collaborator" and a note that the project will integrate with Cloud Object Storage. At the bottom right of the dialog box are "Cancel" and "Create" buttons.

NAME	ROLE	STORAGE	COLLABORATORS	CREATOR	DATE CREATED	ACTIONS
Test 2932388	Admin	COS	Thomas Schaeck	3 Dec 2018		
Data Engineering	Admin	COS	Thomas Schaeck	22 Nov 2018		
Notebook Project	Admin	COS	Thomas Schaeck	22 Nov 2018		
Test Standard Project	Admin	COS	Thomas Schaeck	22 Nov 2018		
T12345	Admin	COS	Thomas Schaeck	11 Sep 2018		
AI Sphere - ECM	Admin	COS	Jihyoung Kim +6	29 Jun 2018		
AI ECM Demo	Editor	COS	Ryan Davis +1	26 Jun 2018		
Thomas Test Basic Project	Admin	COS	Thomas Schaeck	14 Jun 2018		
Test289328937	Admin	COS	Thomas Schaeck	24 Apr 2018		
Test 219370193	Admin	COS	Thomas Schaeck	9 Apr 2018		
Product Team Watson Demo Sandbox	Admin	COS	WatsonDemo +47	24 Mar 2018		
Visual Recognition Project 1	Admin	COS	Thomas Schaeck	18 Mar 2018		
Test NLU Project 232323	Admin	COS	Thomas Schaeck	16 Mar 2018		
Test224	Admin	COS	Thomas Schaeck	1 Mar 2018		
Test 2932893	Admin	COS	Thomas Schaeck	1 Mar 2018		
Test233	Admin	COS	Thomas Schaeck	26 Feb 2018		

Manage Project Members

- After creating a project you see the Overview Page
 - Summary about Assets and Members
 - Recent Activity
 - Readme Info

IBM Watson Studio Projects Catalog Community Services Docs Support Manage

My Projects / Auto Insurance Claims

Overview Assets Environments Bookmarks Deployments Access Control Settings

Launch IDE Add to project

Auto Insurance Claims
Last Updated: Dec 07 2018

352 1 93
Assets Bookmarks Collaborators

Date created Mar 15 2018

Description No description available

Storage 9.6% of 25 GB used

Collaborators View all (93)

Sean Tabbert Admin

PATRICK COUGHLIN Admin

Erel Sharf Admin

Recent activity

Search activities

ACTIVITY DATE

watsondemo watsondemo added Leslie Rodriguez to Auto Insurance Claims 2018/11/20 @ 06:43p

watsondemo watsondemo added Katrin Ellice Heintze, Marion Brülls, Michael Friess, Philipp Brucker... 2018/11/14 @ 11:54a

Joe Plumb added Simon Woodcock to Auto Insurance Claims 2018/11/13 @ 03:36p

See More

- Add members, e.g. data scientists, business analysts, domain experts, stakeholders, ... whom you want to collaborate in your project
- **Members** can have one of the following roles
 - **Admin** – all rights incl deleting project
 - **Editor** – can add and edit assets and run things
 - **Viewer** – can only view assets but not change anything
- Project members can collaboratively add and work with assets and tools in the project

IBM Watson Studio Projects Catalog Community Services Docs Support Manage

My Projects / Auto Insurance Claims

Overview Assets Environments Bookmarks Deployments Access Control Settings

Launch IDE Add to project

Find collaborators

Collaborators

NAME	EMAIL	PERMISSION	STATUS	ACTIONS
Adam J. Massachi	adam.massachi@ibm.com	Admin	Active	⋮
Angshuman Roy	aroy@us.ibm.com	Viewer	Active	⋮
Anthony Casaleotto	acasaleotto@uk.ibm.com	Editor	Active	⋮
Armand Ruiz Gabernet	armand.ruijz@us.ibm.com	Admin	Active	⋮
Arron La	arrona@us.ibm.com	Admin	Active	⋮
Awaiting Acceptance	john.e.popham@us.ibm.com	Editor	Pending	⋮
Awaiting Acceptance	jorgen.lindeman@ibm.com	Admin	Pending	⋮
Brandon Mackenzie	brandonn@ca.ibm.com	Admin	Active	⋮
Brandon Swink	swink@us.ibm.com	Admin	Active	⋮
Brent Walsh	walshbr@us.ibm.com	Admin	Active	⋮
Brock Coughlin	brock.coughlin@us.ibm.com	Admin	Active	⋮
Campbell Robertson	cir@ca.ibm.com	Admin	Active	⋮

Work with Assets & Tools

- Overview tab shows project stats and recent activity
- **Assets** that can exist in a project
 - **Data**
 - Data Assets
 - Connections
 - **Flows**
 - Modeller Flows
 - Data Flows
 - Neural Network Flows
 - **Notebooks**
 - **Scripts**
 - **Models**
 - Watson Machine Learning Models
 - Decision Optimization Models (Add-On)
 - Visual Recognition Models (cloud only)
 - Natural Language Classification Models (cloud only)
 - **Dashboards (Add-On)**
 - Future: Add Workflow oriented View

The image displays two screenshots of the IBM Watson Studio interface, illustrating the management of assets and models within a project.

Top Screenshot (Assets Tab): This screenshot shows the 'Assets' tab selected in the navigation bar. It lists various data assets and connections, each with a preview icon, name, type, creator, last modified date, and actions. The data assets include 'airports.csv', 'Customer Orders with PII.csv', 'New COS connection for NN_test_121218', 'Test2 from Projects', 'Cloudant NoSQL DB-rq for Project', 'BM_COS_EDITOR_Oct 31th', 'Prospect copy.csv', 'Auto Insurance Claims CSV', 'Sales Forecast Data', and 'Db2 Warehouse-fq'. The connections listed are 'Cloudant NoSQL DB-rq for Project', 'BM_COS_EDITOR_Oct 31th', and 'Db2 Warehouse-fq'. The interface also shows sections for Collaborators, Storage, Bucket, and Spark Services.

Name	Type	Created By	Last Modified	Actions
airports.csv	Data Asset	Miri Choi	18 Dec 2018, 9:55:42 am	[More]
Customer Orders with PII.csv	Data Asset	OLENA WOOLF	17 Dec 2018, 5:32:15 pm	[More]
New COS connection for NN_test_121218	Connection	Rohan Vaidyanathan	14 Dec 2018, 2:36:01 am	[More]
Test2 from Projects	Data Asset	OLENA WOOLF	3 Dec 2018, 9:07:42 pm	[More]
Cloudant NoSQL DB-rq for Project	Connection	OLENA WOOLF	3 Dec 2018, 9:07:35 pm	[More]
BM_COS_EDITOR_Oct 31th	Connection	OLENA WOOLF	3 Dec 2018, 9:07:35 pm	[More]
Prospect copy.csv	Data Asset	OLENA WOOLF	1 Nov 2018, 3:22:58 pm	[More]
Auto Insurance Claims CSV	Data Asset	OLENA WOOLF	22 Oct 2018, 9:52:12 pm	[More]
Sales Forecast Data	Data Asset	Thomas Schaeck	18 Oct 2018, 9:42:32 am	[More]
Db2 Warehouse-fq	Connection	GARRETT ROWE	17 Sep 2018, 10:21:19 pm	[More]

Bottom Screenshot (Models Tab): This screenshot shows the 'Models' tab selected. It lists Watson Machine Learning models, Natural Language Classifier models, and Functions. The Watson Machine Learning models section shows 47 entries, including 'Test', 'Notebook Telco Churn Prediction Model', 'Automation', 'Demo Model 11_15', 'Auto Modeling Demonstration', 'Test Model 1', 'DPS Predictive Model', 'Ciber Demo Model', and 'car evaluation'. The Natural Language Classifier models section shows one entry: 'Natural Language Classifier models'. The Functions section shows two entries: 'CARS4U - Satisfaction Prediction - AI Function' and 'CARS4U - Business area and Action Prediction - AI Function', both created by Miri Choi on 5 Oct 2018.

Name	Status	Type	Runtime	Last Modified	Actions
Test	untrained	model builder		7 Dec 2018	[More]
Notebook Telco Churn Prediction Model	trained	mllib-2.3	spark-2.3	4 Dec 2018	[More]
Notebook Telco Churn Prediction Model	trained	mllib-2.3	spark-2.3	4 Dec 2018	[More]
Automation	untrained	model builder		19 Nov 2018	[More]
Demo Model 11_15	untrained	model builder		15 Nov 2018	[More]
Auto Modeling Demonstration	untrained	model builder		14 Nov 2018	[More]
Test Model 1	trained	mllib-2.1	spark-2.1	18 Oct 2018	[More]
DPS Predictive Model	trained	wml-3.1	spark-2.1	27 Sep 2018	[More]
Ciber Demo Model	untrained	model builder		20 Sep 2018	[More]
car evaluation	trained	wml-1.1	spark-2.1	17 Sep 2018	[More]

Name	Type	Last Modified	Actions
CARS4U - Satisfaction Prediction - AI Function	python	5 Oct 2018	[More]
CARS4U - Business area and Action Prediction - AI Function	python	5 Oct 2018	[More]

Connect to Data on IBM Cloud, On Prem, or on 3rd Party Clouds

IBM Watson Studio Projects ▾ Catalog ▾ Community Services ▾ Docs Support ▾ Manage ▾ 1716985 - Thomas Sch... ▾  

New connection

IBM services

 BigInsights HDFS	 Cloud Object Storage	 Cloud Object Storage (infrastructure)	 Cloudant
 Compose for MySQL	 Compose for PostgreSQL	 Db2	 Db2 Big SQL
 Db2 for i	 Db2 for z/OS	 Db2 Hosted	 Db2 on Cloud
 Db2 Warehouse	 Informix	 Object Storage OpenStack Swift	 Object Storage OpenStack Swift (infrastructure)
 PureData for Analytics	 Watson Analytics		

Third-party services

 Amazon Redshift	 Amazon S3	 Apache Hive	 Cloudera Impala
 Dropbox	 FTP	 Google BigQuery	 Hortonworks HDFS
 Looker	 Microsoft Azure Data Lake Store	 Microsoft Azure SQL Database	 Microsoft SQL Server
 MySQL	 Oracle	 Pivotal Greenplum	 PostgreSQL
 Salesforce.com	 Sybase	 Sybase IQ	 Tableau
 Teradata			

Connect Data, add Data Asset

- **Add *Connections* to your project**
 - Add Connections from Watson Knowledge Catalog
 - Create new Connections to your data sources
- **Discover data**
 - Browse Connections using the data asset browser
 - Search for data assets in Knowledge Catalog
 - Find relevant data or subsets of data
- **Add as *Data Asset* to the project**

The screenshot shows the 'Select connection source' dialog in IBM Watson Studio. At the top, there are three tabs: 'Auto Insurance Claims', 'Db2 Warehouse', and 'GOSALES'. Below each tab is a list of connections, schemas, and tables. The 'Db2 Warehouse' tab is currently selected, showing 'Schemas (17)' and 'Tables (25)'. A specific schema entry, 'CONVERSION_RATE', is highlighted in grey. At the bottom right of the dialog are 'Cancel' and 'Select' buttons.

Auto Insurance Claims	Db2 Warehouse	GOSALES
Connections (15)	Schemas (17)	Tables (25)
Insurance Claims App Db	AIOS	BRANCH
Greg Test Cat_DataCatalog	AUDIT	CONVERSION_RATE
BM_COS_EDITOR_Oct 31th	AUTO_INSURANCE	COUNTRY
Data Warehouse on Cloud	BILUADMIN	CURRENCY_LOOKUP
Db2 Warehouse	DTE_INSURANCE_DEMO	EURO_CONVERSION
New COS connection for NN...	GOSALES	INVENTORY_LEVELS
Db2 Warehouse on Cloud-z7	GOSALESDW	ORDER_DETAILS
WDP HOL Db2 Warehouse o...	GOSALESHR	ORDER_HEADER
Db2 Warehouse-WatsonDe...	GOSALESMR	ORDER_METHOD
Think Demo_DataCatalog	GOSALESTRT	PRODUCT
AWS Warehouse	IBM_RTMON_DATA	PRODUCT_BRAND
BM_COS_EDITOR_Enterpr...	JROY	PRODUCT_COLOR_LOOKUP
BM_COS_EDITOR_Fs1Bs	NULLIDR1	PRODUCT_FORECAST
Db2 Warehouse-fq	NULLIDRA	PRODUCT_LINE
Cloudant NoSQL DB-rq for ...	PUBLIC	PRODUCT_NAME_LOOKUP
	SAMPLES	PRODUCT_SIZE_LOOKUP
	ST_INFORMTN_SCHEMA	PRODUCT_TYPE
		RETURNINFO_TTFM

The screenshot shows the 'Add asset from connection' dialog in IBM Watson Studio. It has fields for 'Source' (set to 'Db2 Warehouse'), 'Name*' (set to 'Sales Data'), and a 'Description' text area containing the placeholder text 'This data set contains sales data ...'. At the bottom right are 'Create' and 'Cancel' buttons.

My Projects / Auto Insurance Claims / Add asset from connection

Add asset from connection

Source: Db2 Warehouse Change source

Db2 Warehouse : /GOSALES/CONVERSION_RATE

Name*
Sales Data

Description
[This data set contains sales data ...]

263

27

Preview and refine Data

- **Preview and visualize Data**

- Table View
- Profile
- Visualizations

The screenshot shows the IBM Watson Studio interface with the project 'Auto Insurance Claims' selected. A preview of the 'Auto Insurance Claims CSV' file is displayed, showing a schema with 5 columns and 5000+ rows. The columns are: CUSTOMER (Type: String), CLAIM_AMOUNT (Type: String), RESPONSE (Type: String), CLAIM_REASON (Type: String), and TOTAL_CLAIM_AMOUNT (Type: String). The data includes various entries such as BU97986, QZ44356, AI49188, WW63253, HB64268, OC83172, XZB7318, CFB5061, DYB7989, BQ94931, SX51350, VQ65197, DP39365, SJ95423, IL66569, and BW63560. The interface includes a 'Refine' button and a message bubble icon.

- **Refine data** to prepare for analysis or ML

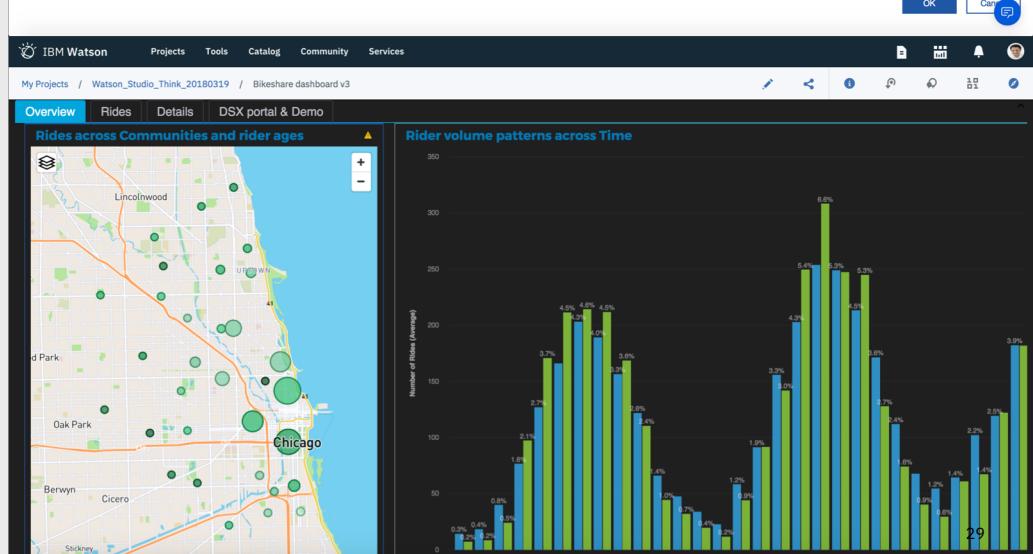
- Filter or anonymize, change column types, combine data, based on sample, resulting in a *Data Flow*
- Run the Data Flow to generate result data set by processing the full data
- Data Flow remains available in project to re-run later if needed

The screenshot shows the IBM Watson Studio interface with the project 'Auto Insurance Claims' selected. A 'Data Refinery' flow is being edited, titled 'Auto Insurance Claims CSV_flow'. The flow consists of several steps: 'Search operations' (remove duplicates), 'Remove empty rows', 'Replace missing values', 'Replace substring', 'ORGANIZE' (with sub-steps 'Aggregate', 'Concatenate', 'Conditional replace'), 'Join', 'Sample', 'Split column', and 'NATURAL LANGUAGE' (with sub-steps 'Remove stop words', 'Tokenize'). The output table shows the same data as the preview, with columns CUSTOMER, CLAIM_AMOUNT, RESPONSE, CLAIM_REASON, and TO. The interface includes tabs for 'Details' and 'Help', and sections for 'DATA REFINERY FLOW DETAILS' (Location: Auto Insurance Claims, Name: Auto Insurance Claims CSV_flow) and 'DATA REFINERY FLOW OUTPUT' (Location: Auto Insurance Claims/Data ref...).

Visualize and Analyze Data

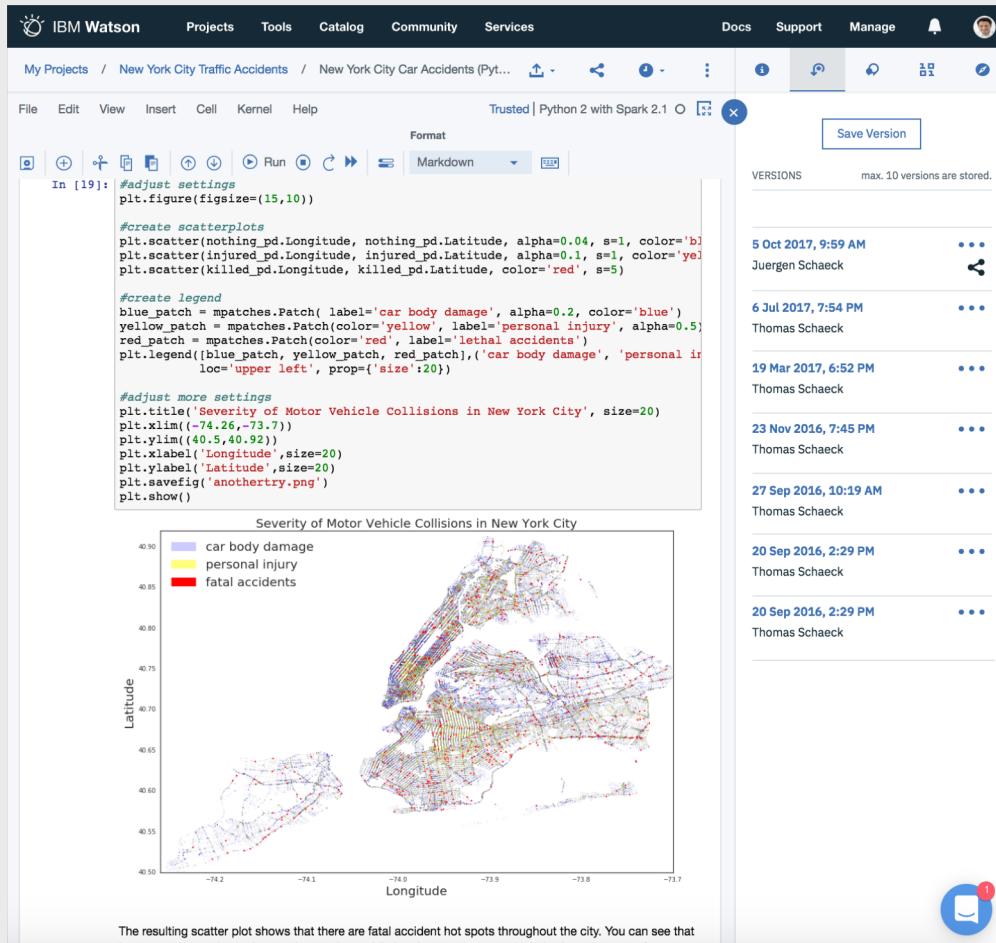
- **Create Dashboard** and connect to data in the project
 - Uploaded CSV files
 - Db2 Warehouse connection or data asset
 - Postgres connection or data asset
- **Pick data** from connection or subset of data asset
- **Compose** dashboards visualizing data leveraging a broad range of widgets and visualizations
- **Share** Dashboard with other project members or optionally get view-only link to share with anyone
- **Export** useful Dashboards as file (.json) for re-use in other Projects
- **Import** dashboard file to your project and re-link to data in your project
- **Publish** dashboard files from project to Watson Knowledge Catalog, add from catalog to your project
- **Soon: Community** Dashboard samples

The screenshot shows the 'Select a template' dialog in IBM Watson Studio. On the left, there are four categories: 'Dashboard' (single page), 'Tabbed' (Freeform and grid-based), and 'Infographic'. The 'Tabbed' category is selected. It shows a 'Freeform' layout where visualizations appear exactly as you size and place them in the view, regardless of the screen size. Below it is a 4x4 grid of preview boxes representing different tabbed layout configurations.



Jupyter Notebooks

- **Active docs that contain insights and code to get them**
 - Text to explain what the Notebook does and document insights
 - Code cells for accessing, processing, and analyzing data
 - Result cells with output and visualizations
- **Leverages Jupyter <http://jupyter.org>**
 - Most popular OS Notebook project, de facto standard
 - Allows to create, run, edit notebooks in web browsers
 - Can run with Python, R, Scala, and many other kernels
 - Notebooks are stored as .ipynb JSON objects
- **Popular for publishing interactive tutorials & demos with runnable code inside, and reproducible research**
- **IBM Value Add in Watson Studio**
 - Project Library for Python and R
 - Versions and Comments
 - Schedule a Notebook version to regularly run
 - Publish as URL to view most recent Notebook run
 - Insert-to-Code to connect to data
 - Share to Git repo



Environments

Built-in compute environments

- **Python / R**
- **Python / R + Spark**
- **Python / R + GPUs**
 - Existing function on Watson Studio Local
 - New on Watson Studio Cloud – Open Beta

Notebooks can optionally use Spark from

- IBM Analytics Engine or AWS EMR (WS Cloud)
- Hortonworks or Cloudera Clusters (WS Local)

The screenshot shows the IBM Watson environments interface. At the top, there's a navigation bar with links for Projects, Tools, Catalog, Community, Services, Docs, Support, Manage, and a user profile icon. Below the navigation is a breadcrumb trail: My Projects / Test Project. A top banner displays account statistics: 0.0 capacity unit hours used this month, 41.5 remaining account capacity unit hours included in plan.

The main content area has tabs for Overview, Assets, Environments (which is selected), Bookmarks, Deployments, Access Control, and Settings. The Environments tab contains sections for environments and environment definitions.

Environments: A search bar asks "Which environment are you looking for?". It includes a note: "In this release you can use default environments or you can create custom environments. Learn more." Below this is a table titled "Active environment runtimes" which currently shows no active runtimes.

Environment definitions: A table lists six environment definitions with columns for Name, Tool, Hardware Configuration, Last Modified, and Actions. All listed environments are Notebooks with specific hardware configurations and were last modified on Feb 16, 19, or 2018.

Name	Tool	Hardware Configuration	Last Modified	Actions
Default R 3.4 S	Notebook	4 vCPU and 16 GB RAM	19 Feb 2018	⋮
Default R 3.4 XS	Notebook	2 vCPU and 8 GB RAM	19 Feb 2018	⋮
Default Python 3.5 Free	Notebook	1 vCPU and 4 GB RAM	16 Feb 2018	⋮
Default Python 3.5 S	Notebook	4 vCPU and 16 GB RAM	16 Feb 2018	⋮
Default Python 3.5 XS	Notebook	2 vCPU and 8 GB RAM	16 Feb 2018	⋮

Create & train ML Models

- **Create Model** from Project Assets tab
 - Automatic – prepare data and create model
 - Manual – user prepares data and selects model
 - Modeller Flow – create and train model in a flow
 - Notebook – create and train model through code
- **Deploy model** to WML
 - the model becomes available through REST API
 - get URL & code snippets from Implementation tab
 - try model with different values in test tab
- **Invoke model** from any client
 - Notebooks in Watson Studio or elsewhere
 - Apps on IBM Cloud
 - Other apps or other clients
- **Alternatively: Export model** and run where needed
 - Own container images
 - Mobile apps

The screenshot displays two main sections of the IBM Watson Studio interface.

New model (Top Section):

- Define model details:**
 - Name: New Model
 - Description: New Model
 - Machine Learning Service: Machine Learning-WatsonDemoEnv
- Select model type:**
 - Model builder
 - From file
 - From sample

Spark Service or Environment: Spark-WatsonDemoEnv-Lite-1

Automatic: Prepare my data and create a model automatically

Manual: Let me prepare my data and select which models to train

Need something more flexible? Create a [notebook](#) or design a [Modeler flow](#)

Churn Model 1 (Bottom Section):

- Overview:** Churn Model 1
- Implementation:** Web Service
- Test:** DEPLOY_SUCCESS
- Deployment:**

Name	Churn Model 1
Type	Web Service
Deployment ID	db51b2bb-7d61-4b63-bf8e-735865213333
Status	DEPLOY_SUCCESS
Machine learning service	Machine Learning-WatsonDemoEnv
Created	02 May 2018 04:57pm
Last modified	04 Jul 2018 01:47pm
- Model:**

Name	Churn Model 1
Model ID	7b24546b-f75a-4826-9b65-a7d6a7d8b85

Flows

- Create new Flow in the project
 - Modeler Flow
 - Neural Network Flow
- Run Modeler flows with a WML runtime option
 - SPSS
 - Spark
- Download Neural Network Flow as
 - Flow file
 - TensorFlow, Keras, PyTorch, or Caffe model

IBM Watson Projects Tools Catalog Community Services Manage Support Docs Thomas Schäck's Acco...

Modeler

New From file From example

Name*
My Flow 43

Description
Neural Network Flow 481

Select flow type

Modeler Flow Neural Network Modeler BETA

Create

IBM Watson Projects Tools Catalog Community Services Docs Support Manage

My Projects / Auto Insurance Claims / VGG19

```
graph LR; ImageData[ImageData] --> Conv2D1[Conv 2D]; Conv2D1 --> Conv2D2[Conv 2D]; Conv2D2 --> Pool2D1[Pool 2D]; Pool2D1 --> Conv2D3[Conv 2D]; Conv2D3 --> Conv2D4[Conv 2D]; Conv2D4 --> Pool2D2[Pool 2D]; Pool2D2 --> Conv2D5[Conv 2D]; Conv2D5 --> Conv2D6[Conv 2D]; Conv2D6 --> Pool2D3[Pool 2D]; Pool2D3 --> Conv2D7[Conv 2D]; Conv2D7 --> Conv2D8[Conv 2D]; Conv2D8 --> Pool2D4[Pool 2D]; Pool2D4 --> Flatten[Flatten]; Flatten --> Dense1[Dense]; Dense1 --> Dense2[Dense]; Dense2 --> Dense3[Dense]; Dense3 --> Softmax[Softmax]; Softmax --> Accuracy[Accuracy]; Accuracy --> Sigmoid[Sigmoid Cross-E...]; Sigmoid --> Adam[Adam]; Adam --> Dense4[Dense]
```

Human-in-the-Loop with Figure8 and Defined Crowd

- To be useful for ML, training data needs to have labels to learn from
 - What is in the text ?
 - What's the sentiment of text ?
 - What's in a picture ?
 - What's the human decision based on a data record ?
 - ...
- Crowd sourcing allows engaging humans in activities such as labelling data, images, text, ...
- Watson Studio integrates with Figure8 and Defined Crowd in order to drive labelling of data in Projects through these services

The image displays two side-by-side screenshots of the IBM Watson Studio interface, specifically the 'Create an annotation job' section.

Screenshot 1 (Top): This screenshot shows the initial steps of creating an annotation job. At the top, there is a navigation bar with links for 'IBM Watson Studio', 'Projects', 'Tools', 'Catalog', 'Community', 'Services', 'Manage', 'Support', 'Docs', and a bell icon. Below the navigation bar, the title 'Create an annotation job' is displayed, followed by a horizontal progress bar with three steps: 'Basic' (highlighted in blue), 'Design job', and 'Review'. On the left, there is a section for 'Defined Crowd' with its logo and a brief description: 'DefinedCrowd is a smart data platform that enables data scientists to collect, refine, and structure training data for AI and ML applications.' A link to 'Learn more at definedcrowd.com' is provided. On the right, under the heading 'Basics', there is a field labeled 'Name your job *' with the placeholder 'This will appear in your account on DefinedCrowd's Enterprise Portal.' Below it is a 'Job name' input field containing 'Job name' with a character count of 255. Further down, there is a section titled 'Select a project template' with a dropdown menu containing 'Text Sentiment Annotation' (which has a checked checkbox) and a description: 'Determine the sentiment of a sentence.'

Screenshot 2 (Bottom): This screenshot shows the continuation of the annotation job creation process. The top navigation bar is identical. The title 'Create an annotation job' is repeated. On the left, there is a section for 'Figure Eight' with its logo and a brief description: 'Figure Eight is the essential Human-in-the-Loop AI platform for data science and machine learning teams.' A link to 'Learn more at figure-eight.com' is provided. On the right, under the heading 'Choose a job template', there are two options: 'Sentiment Analysis' (checked) and 'Judge the Relevance and Sentiment of Content'. Both descriptions mention that contributors analyze text (e.g., social media posts or forum comments) about a specific topic. At the bottom right of this section, there are 'Cancel' and 'Create job' buttons.

Decision Optimization (DO) in Notebooks

- **Notebooks** – Develop models directly in notebooks...
- **Python** – Leverage the power of optimization in the most popular data science language.
- **All Data Science** – Combine Decision Optimization with Machine Learning models

In [6]: `mdl = CpoModel(name="trucks")`

4.3 Define the decision variables
The comments in the code provide details about the variables.

In [7]: `# Configuration of the truck for each delivery
truckConfigs = IntegerVarList(nbDeliveries, 0, nbTruckConfigs - 1, "truckConfigs")
Association between a delivery and an order
where = IntegerVarList(nbOrders, 0, nbDeliveries - 1, "where")
Truck Load
load = IntegerVarList(nbDeliveries, 0, maxLoad, "load")
Number of deliveries per customer
nbDeliveries = IntegerVar(0, nbDeliveries)
Association between a customer and a delivery
customerOfDelivery = IntegerVarList(nbDeliveries, 0, nbCustomers, "customerOfTruck")
Transition cost for each delivery
transitionCost = IntegerVarList(nbDeliveries - 1, 0, 1000, "transitionCost")`

4.4 Specify the business constraints
The comments in the code provide details about the constraints.

In [8]: `# transitionCost[i] = transition cost between configurations i and i+1
for i in range(0, nbDeliveries):
 mdl.addConstraint(transitionCost[i], truckConfig[i], truckConfig[i + 1], transitionCost[i + 1])
 mdl.addAllowedAssignment(truckConfig, CONFIGURATION_TRANSITION_COST)

Constraint on the volume of orders in each truck
mdl.add(packLoad, where, volumes, nbDeliveries)
for i in range(0, nbDeliveries):
 mdl.add(load[i] <= element(truckConfig[i], maxTruckConfigLoad))

Relationship between the product type of an order and the configuration of its truck
for j in range(0, nbOrders):
 configContainer = IntegerVar(ALLOWED_CONTAINER_CONFIGS[productType[j]])
 mdl.add(configContainer == element(truckConfig[j], where[j]))

No more than one customer can deliver.
No more than one customer can deliver.`

IBM Watson Studio Projects Services Community Docs Support Manage Sans Confiance Python 3.5 O General Environment

ENVIRONMENT DEFINITION Default Python 3.5 XS + Beta of DO LANGUAGE Python 3.5 HARDWARE CONFIGURATION 2 vCPUs and 8 GB RAM SOFTWARE CONFIGURATION VIEW DETAILS

RUNTIME STATUS Running

Entrée [28]: `from docplex.mp.model import Model
mdl = Model(name="PredictiveMaintenance")`
/opt/conda/envs/DSX-Python3.5/lib/python3.5/site-packages/docplex/mp/model.py:14: Warning: These editions of Docplex and CPLEX for Watson Studio are in beta.
warnings.warn('These editions of Docplex and CPLEX for Watson Studio are in beta.', Warning)

Create a new optimization model.

Entrée [29]: `production = mdl.continuous_var_matrix(keys=all_machines, keys2=all_days, name=lambda ns: "Production_5s_5s" % (ns[0],ns[1]))
df_production = pd.DataFrame({'production': production})
df_production.index.names = ['all_machines', 'all_days']`
maintenance = mdl.binary_var_matrix(keys=all_machines, keys2=all_days, name=lambda ns: "Maintenance_5s_5s" % (ns[0],ns[1]))
df_maintenance = pd.DataFrame({'maintenance': maintenance})
df_maintenance.index.names = ['all_machines', 'all_days']

Add some constraints linking real production with planned production and maintenance.

Entrée [30]: `for machine in all_machines:
 maintenance_loss = int(df_machine[df_machine['id']==machine]['maintenance loss'])/100.
 capacity = int(df_machine[df_machine['id']==machine]['capacity'])
 for day in all_days:
 prod = df_planned_production.production[machine, day]
 maint = df_maintenance[maintenance[machine, day] == 1, production[machine, day]]
 prod.addIfThen(maintenance[machine, day] == 1, production[machine, day] == df_production[production.machine==machine]/df_production.day==day) .`

DO Model Builder (in WS Local, soon on Cloud)

- Model Builder**
Simple 4-step guided builder
- OPL (Optimization Programming Language)**
Import and run OPL models.
- Interactive Dashboard**
Understand and share the optimization results and insights through a visual, interactive interface.
- Scenario Comparison**
Explore trade-offs between different action plans in a single view.
- Debug, tune, validate with Scenarios and Dashboard**
- Scenario Management from Python**
Create, duplicate, update, solve scenarios directly from Python.

The screenshot shows the IBM Watson Studio interface with the DO Model Builder extension. The top section displays three tables: diet_nutrients, diet_food_nutrients, and diet_food. The diet_nutrients table shows nutrients like Calories and Calcium with min and max values. The diet_food_nutrients table shows food items like Roasted Chicken and Spaghetti W/ Sauce with their nutritional values. The diet_food table shows food items with unit cost and quantity ranges. To the right, a code editor shows an imported Python script named model.py, which contains OPL modeling code for diet planning. Below this, a scenario comparison dashboard is shown for a bridge construction project, featuring a Gantt chart, Vega Charts, and a pie chart of activity types.

```
Imported from "model.py" on 13 Dec 2017, 6:15 PM | Python | Run
```

```
1 #!/usr/bin/python
2 # Food = inputs["diet_food"]
3 # nutrients = inputs["diet_nutrients"]
4 # diet_food_nutrients = inputs["diet_food_nutrients"]
5 # food_nutrients.set_index("Food", inplace=True)
6 # add cells
7 from doplex import Model
8
9 mdo = Model(name="diet")
10
11 # Create decision variables, limited to be >= Food_min and <= Food_max
12 #ty = Food[[ "name", "min", "max" ]].copy()
13 #ty["var"] = qtry.apply(lambda x: mdo.continuous_var(x["min"], x["max"], name=x["name"]))
14 #ty.set_index("name", inplace=True)
15
16 # make the name the index
17 ty.set_index("name", inplace=True)
18
19 # Create range of nutrients, and mark them as KP1
20 #for nutrient in nutrients:
21 #    amount = mdo.sum(ty.loc[nutrient]["var"] * food_nutrients.loc[nutrient][name])
22 #    mdo.add_constraint(amount == 1, "FoodNutrientSum")
23 #    mdo.add_range(amount, 0, 1, "FoodNutrientRange")
24 #
25 # minimize cost
26 obj = mdo.sum(ty.loc["var"] * f.unit_cost for f in food.itertuples())
27 mdo.set_objective(obj, "Minimize", "cost")
28 mdo.optimize()
29
30 mdo.print_information()
31 mdo.write_lp("diet.lp")
32
33 # Add printout
34 mdo.print_out()
35
36 ok = mdo.solve()
37 if ok:
38     mdo.print_out()
39     mdo.print_solution()
40     mdo.print_dataframe_from_solution()
41
42 # Add imports
43 import pandas
44
45 solution_df = pandas.DataFrame(columns=["name", "value"])
46
47 for index, row in enumerate(mdo.solution_iter.variables()):
48     solution_df.loc[index, "name"] = row.name
49     solution_df.loc[index, "value"] = row.value
50
51 solution_df.to_csv("solution.csv")
```

IBM Watson Studio Projects : dsx-samples : Diet - Scenario 1

Prepare Input Data

Select Data

Prepare 3 tables Run Model Explore Solution Processed

Dashboard

diet_nutrients

name	qmin	qmax
Calories	2,000	2,500
Calcium	800	1,600

diet_food_nutrients

Food	Calories	Iron	Vit_A	Dietary_Fiber
Roasted Chicken	277.4	21.9	77.4	0
Spaghetti W/ Sauce	250.2	80.2	2.3	3,055.2

diet_food

name	unit_cost	qmin	qmax
Roasted Chicken	0.84	0	10
Spaghetti W/ Sauce	0.78	0	10

IBM Watson Studio Projects : dsx-samples : BridgeSch... - Scenario 1

Select Data

Input Schedule +

Notes Table Charts Vega Charts Gantt

The Gantt Chart shows when activities are scheduled, and how they are assigned to equipment.

Table E...

Name	Required equipment
bricklaying	1
carpentry	1
caterpillar	1
concrete mixer	1
crane	1
excavator	1
pile driver	1
concrete mixer	1
crane	1
excavator	1
pile driver	1

Gantt

Vega Charts Activity

Charts Scheduled activities report

Pie Chart

Watson Knowledge Catalog

Share & reuse Assets across the Enterprise
Unlock tribal knowledge to unleash your data professionals

- Discover** - Intelligent discovery of data, advanced classification and profiling to provide context
- Catalog** - A rich metadata index of all data, with social collaboration and enhanced findability
- Govern** - Powerful governance policy tools to control and protect access to data with visibility to data use

The screenshot displays the Watson Knowledge Catalog interface. At the top, there's a navigation bar with links for IBM Watson, Projects, Tools, Catalog, Community, and Services. Below the navigation is a search bar with the term "auto". The main area shows a grid of "Available Assets" under the heading "Watson Recommends". Each asset card includes a thumbnail, name, owner, add date, and a star rating. For example, the first asset is "Auto Insurance Customers" owned by "watsondemo watsondemo" added on Mar 15, 2018 at 11:49 PM, with a 3-star rating from 3 reviews. To the left of the asset grid is a "Filter" sidebar with sections for Asset types (Data Asset, Connection, Notebook) and Tags. Below the filter is another navigation bar with links for Catalogs, Think Demo, Auto Insurance Customers, and tabs for Overview, Access, Profile, and Review. On the right side, there's a detailed view of the "Auto Insurance Customers" asset, showing its description ("All US auto insurance customers. This asset contains sensitive information and is protected by the Sensitive Data governance policy and the Financial Accounts and Government Identities rules for anonymized data."), overall rating (3 stars from 3 reviews), tags (insurance, auto, customers), connection (Source: Think Demo, DataCatalog; Type: Cloud Object Storage), and classification (SPI). The schema section shows 28 columns and 2 anonymized columns, with a preview of 1000 rows. The bottom right corner of the screenshot has a blue circular badge with the number 37.

Publish/Re-use Assets

- Pick Asset in a Project to publish to catalog
 - Connections
 - Data Assets
 - Notebooks
 - Models
 - Soon: Dashboards
- Published assets become available in the catalog and can be found by other users
- Find Asset in a Catalog and add to project to re-use
 - Add Connection or Data Asset to Project
 - Create new notebook from catalog notebook asset
 - Create new model from catalog model asset
 - Soon: Create new dashboard from catalog dashboard asset

The screenshot shows the IBM Watson Catalog interface. At the top, there are navigation links: Projects, Tools, Catalog, Community, Services, Docs, Support, Manage, and a user profile icon. Below this, the title "My Projects / Auto Insurance Claims" is displayed. A dropdown menu "Models" is open, showing "View all (27)". On the right, there is a "New model" button. The main area is a table with columns: NAME, STATUS, TYPE, RUNTIME, LAST MODIFIED, and ACTIONS. The table lists various models like "Churn Model1", "Auto Claim Test Model KAL", etc. In the "Actions" column for the "Mark's Test ML Model", a "Publish to Catalog" button is highlighted with a blue box.

NAME	STATUS	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
Churn Model1	trained	wml-1.1	spark-2.1	28 Jun 2018	⋮
Auto Claim Test Model KAL	untrained	model builder		19 Jun 2018	⋮
Madison Test Model	untrained	model builder		19 Jun 2018	⋮
Sample	trained	wml-1.1	spark-2.1	15 Jun 2018	⋮
Mark's Test ML Model	trained	wml-1.1	spark-2.1	14 Jun 2018	⋮
High Jump Model	untrained	model builder		11 Jun 2018	⋮
LMG Model June 11	trained	wml-1.1	spark-2.1	11 Jun 2018	⋮
kytydfjytd	untrained	model builder		8 Jun 2018	⋮
BA demo	untrained	model builder		7 Jun 2018	⋮
DPS Predictive Model	trained	wml-1.1	spark-2.1	7 Jun 2018	⋮

The screenshot shows a detailed view of a published model in the IBM Watson Catalog. The top navigation bar is identical to the previous screenshot. The title "Catalogs / Great Outdoors Sandbox / Frida Earthquake Prediction 0623" is displayed. A "Add to Catalog" button is present. The page has tabs: Overview (highlighted), Access, and Review. Below the tabs, there is a "Model" section with a thumbnail and the title "Frida Earthquake Prediction 0623". Action buttons include "Remove", "Download", and "Add to Project". The main content area is divided into sections: "Description" (Earthquake Prediction Model), "Summary" (Training date: 29 Jun 2018, 1:54 PM; Framework: wml 1.1), "Overall Rating" (0 reviews), "Tags" (earthquake), and "Input Schema". The "Input Schema" table has columns: COLUMN and TYPE, with rows: Date (date) and Time (timestamp). A small red circle with the number 1 is visible in the bottom right corner.

COLUMN	TYPE
Date	date
Time	timestamp

Train Visual Recognition (Watson Studio Cloud only)

Collect training data for Visual Recognition and train/test Visual Recognition Models

Upload ZIPs with training images

Typically having at least 10 images per class suffices to get good results

The screenshot shows the 'Create a class' screen in Watson Studio. At the top, there's a search bar labeled 'Search classes' and a note about the total file size being 94.5/250 MB. Below this, there's a section for 'Add zipped files from your project.' A large dashed box labeled 'Create a class' contains a blue plus sign. To its right, there are four rows of image thumbnails, each representing a different car model: 'BMW M3 Coupe' (10 images), 'BMW X3 SUV' (10 images), 'Hyundai Elantra Touring...' (20 images), 'Volvo C30 Hatchback' (10 images), 'Volvo XC90 SUV' (10 images), and 'Volvo XC90 SUV 2007' (86 images). Each row has a green circular progress indicator.

Train a Visual Recognition Model

Pick the classes you want to include in training the models, Watson Studio automatically connects to Watson Visual Recognition to train with the selected data

Use the Visual Recognition Model

Get the REST API URL plus code snippets to invoke the Visual Recognition model from your application

The screenshot shows the 'Default Custom Model' screen in Watson Studio. At the top, it says 'Associated Service : watson_vision_combined-dsx'. There are tabs for 'Overview', 'Test', and 'Implementation'. The 'Implementation' tab is active. On the left, there's a sidebar with 'Code Snippets' sections for 'curl', 'Java', 'Node', 'Python', and 'Core ML'. The 'Core ML' section is expanded, showing code snippets for 'Download model file', 'Core ML documentation', and a link to 'https://github.com/watson-developer-cloud/visual-recognition-with-discovery-coreml'. Below this, there's a note about Core ML status and a code snippet for 'Core ML' in Python. At the bottom, there's a note about updating the local Core ML model and another code snippet for 'Core ML' in Python.

Natural Language Classification

(Watson Studio Cloud only)

Note: Requires paid IBM Cloud account

*Collect training data for
Natural Language Classification
and train/test NLC Classifiers*

*Get endpoint info and code snippets
to invoke NLC via public API*

The screenshot shows the IBM Watson Studio Cloud interface. At the top, there's a navigation bar with links for Projects, Tools, Catalog, Community, Services, Manage, Support, and Docs. Below the navigation bar, the current project is 'NLC_0627'. The main area is titled 'Email_Spam_Classifier'. Under the 'My classes' tab, there are three examples shown:

- ham (8)**: Subject: start date : 2 / 1 / 02 ; hour/lead hour : 18 ; no ancillary schedule awarded . no varieties denoted . no specific time period . > 0.1 portland \ westdesk \ california scheduling \ iso final schedules \ 2002020118 .txt
- spam (9)**: Subject: another time ? hey , i am back in olympia now . my friends wedding was sunday and i drove
- Another example snippet is partially visible.

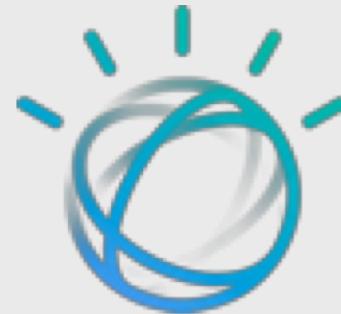
A sidebar on the right has sections for 'Upload to project' (with a 'Browse' button) and 'Add from project' (with a 'Browse' button). It also shows a list of selected files: 'spamDemo.csv' (29 Jun 2018, 10:09:55 am, 5.85 KB).

The screenshot shows the IBM Watson Studio Cloud interface for the 'Email_Spam_Classifier' project. The 'Implementation' tab is selected. On the left, there's a sidebar for 'Code Snippets' with options for cURL, Java, Node, and Python. To the right, there's documentation for the API:

- API endpoint**: <https://gateway.watsonplatform.net/natural-language-classifier/api>
- IAM authentication**: Replace {apikey} with your service credentials.
curl -u "apikey:{apikey}" "https://gateway.watsonplatform.net/natural-language-classifier/api/{method}"
- Basic authentication**: Replace {username} and {password} with your service credentials.
curl -u "{username}:{password}" "https://gateway.watsonplatform.net/natural-language-classifier/api/{method}"

Auto AI Preview

Join the waitlist for a chance
to provide feedback and co-
develop capabilities with IBM



<https://datascienceex.typeform.com/to/shx0eB>

Architecture Details

Watson Studio Cloud, Local, Desktop

Catalog Integration

Model Interop

Releases

Project Template ZIP

Git Integration

Watson Studio Gallery

Watson Studio – Desktop

Watson Studio Desktop Navigation

Data Assets

Shaper

Dashboards

Flows – initially SPSS

Projects
(Personal)

Notebooks

Decision Optimization

Catalog Integration

Python/R
Environments
opt GPUs

SPSS
Runtime

Remote
Spark
Integration

Desktop Integration

Windows OS

Mac OS

Watson Studio Local – Private Cloud

Watson Studio Web Site + Catalog Integration

Data Assets

Shaper

Dashboards

Flows – WML/SPSS>NNM

Projects
(Teams)

Notebooks

RStudio

Decision Optimization

Watson Explorer

Catalog Integration

Python/R
Environments
opt GPUs
opt Spark

WML/SPSS
Integration

Remote
Spark &
Hadoop
Integration
(Hortonworks,
Cloudera, ...)

ICP4Data/ICP Integration

x86

Power

zLinux

Watson Studio – Public Cloud

Watson Studio Web Site + Catalog Integration

Data Assets

Shaper

Dashboards

Flows – WML/SPSS>NNM

Projects
(Teams)

Notebooks

RStudio

Decision Optimization

AI Tools (VR, NLC)

Catalog Integration

Python/R
Environments
opt GPUs
opt Spark

WML/SPSS
Integration

Remote
Spark
Integration
(IAE, AWS
EMR, ...)

IBM Cloud Integration

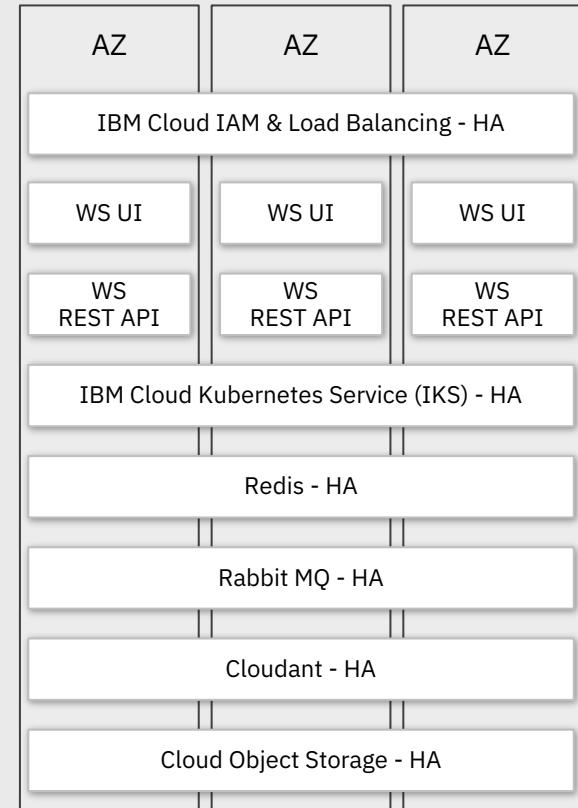
Kubernetes (Armada)

IBM Cloud

Watson Studio Cloud

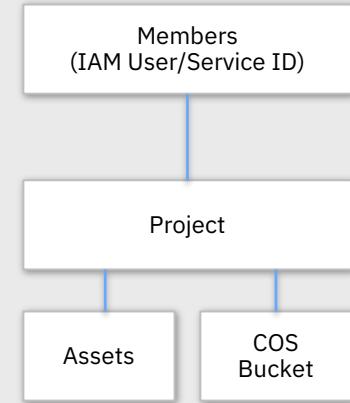
High Availability – Watson Studio Cloud

- Fully Managed by Watson Studio Cloud
→ „just works“ for customers
- All WS micro services deployed on IKS across 3 AZs per region
 - UI Microservices Node.js/Liberty 3-way active-active HA
 - REST API Microservices Node.js/Liberty 3-way active-active HA
- Using 3-way HA IBM Cloud services
 - Redis for in memory caching 3-way HA
 - RabbitMQ for message queueing and notifications 3-way HA
 - Cloudant for metadata persistence 3-way HA with online data replication across three AZs and online data replication and back up to other region
 - Cloud Object Storage 3-way HA with online replication of files across AZs



Security – Watson Studio Cloud

- Watson Studio integrates with **IBM Cloud IAM** for identity & access management
 - Customers can manage users and groups in IBM Cloud account admin
 - Customers can use own identity providers through **identity federation** (SAML)
- Users can only access a Project and assets & data in it if they have a valid IAM User ID and are a member of the Project
- Project data is stored in per-project Cloud Object Storage buckets that can be encrypted with **customer managed keys using Key Protect**
- Users can be Reader, Editor, or Admin for a Project
 - Admins** have all rights on the project, incl. managing project members, associating services, deleting the project
 - Editors** can add/edit/delete assets in the project and using Notebooks, Shaper, or Flows create/change/delete data in the project
 - Readers** can only view / read information in the project
- Admins can also add IAM Service IDs as Reader, Editor, or Admin, to enable access using that Service ID through the Project REST API



Disaster Resilience – Watson Studio Cloud

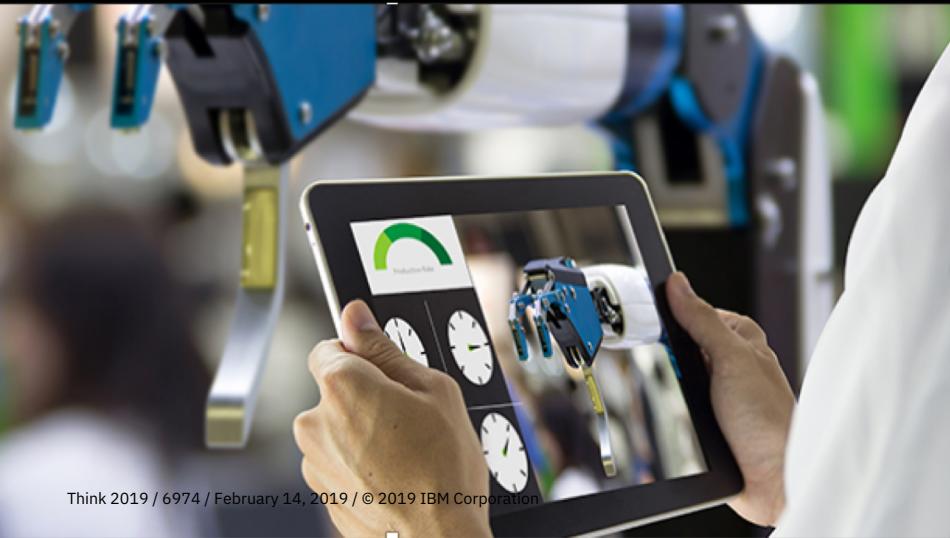
- Watson Studio metadata is internally stored in Cloudant which is 3-way HA and disaster resilient by replicating data across three AZs
- Additionally the WS Cloudant DBs are online replicated and backed up to a remote region
- Data that customers upload to their projects or create/manipulate in their projects is stored in IBM Cloud Object Storage buckets configured to persist to 3 AZs for HA and DR
- RStudio personal data is backed up for DR

Scalability – Watson Studio Cloud

- Watson Studio uses internal Kubernetes clusters for WS micro services and WS compute runtimes for Python, R, Spark, ...
- Customers / end users can run notebooks, flows, train models, etc without having to worry about scalability
- Watson Studio Cloud scales by internally adding more VMs to Kubernetes clusters to increase capacity as needed to meet aggregate customer compute demand
- Depending on Watson Studio Cloud plan a customer is using, a number of Compute Unit Hours are included
- If Watson Studio usage in an IBM Cloud account exceeds the number of included CUHs, extra CUHs will be billed for
- For Watson Studio plans, #included CUHs, and CUH cost refer to the Watson Studio plan information in the IBM Cloud Catalog.

IBM AI Learning & Certification: AI Literacy for ALL

Sharing our deep AI knowledge & experience from working with hundreds of enterprise clients.



Now Live:

Online learning with our AI Learning Catalog

Custom, in-lab AI Learning Experiences

Coming Soon:

IBM AI Certification for the end-to-end enterprise AI workflow.

Invest in your future. Start your AI Learning now.

<http://community.ibm.com/aiskills>

For information visit us at booth 347

Thank you

