

# DSX Local

## Overview & Roadmap

July 2018

Joel Patterson  
jpatter@us.ibm.com

<https://us45001.iacs.ibm.com>

user01 - user20 / IBMdsxl!



## Explore at scale

- Scale out on-demand
- No Dev-ops/engineering setup

## Reproducibility

- Process of tracking
- Reproduce results easily

## Secure

- Governed Access
- Administration capabilities



## Collaborate

- Understand what's been done
- Share and accelerate learning

## Publish Efforts

- Models as APIs out of the box
- Avoid Engineering re-work

## Discovery to Production

- Minimal efforts
- Seamless scale
- Integration with business process



## Open

- Use desired tool of choice
- Interoperability across tools

## Review Results

- Stakeholder review
- Via Dashboards/Static reports

## Monitoring

- QA/QC on-demand
- Retrain





## Develop & Collaborate

**Notebook servers and RStudio** for interactivity & data visualizing with Python, R, & Scala for coder data scientists



## Deploy at Scale

**Spark** parallelizes & accelerates data science tasks.  
Automate, Deploy scoring servers at scale and monitor model health

### Watson Studio on IBM **Public Cloud**

- PayGo consumption with **as-a-service** delivery, up & running in seconds
- Integrated with IBM Spark-as-a-Service & Watson Machine Learning as a service for compute,
  - IBM Object Store & other cloud services for data,
- Publish and collaborate in the cloud

### DSX **Desktop**

- Easily installed on your **laptop or PC**
  - Won't scale beyond the hardware available on your machine
- Access to RStudio, Zeppelin and Jupyter notebooks, and one small Spark worker operating locally on your machine
- Load CSV data files into Data Frames

### DSX **Local** on Private Cloud

- Scalable DSX cluster deployed on your **private infrastructure**
  - Dockerized containers via Kubernetes & deployable on IBM Cloud Private
- DSX Local can also deploy with **Hortonworks Data Platform** & IIAS appliance on-premises
- LDAP for user management and authentication, easy collaboration with Projects enabled by git.

➤ Use DSX *wherever* it makes sense for you (or where your data is) - and you can easily collaborate on the *same* project across these environments

*and with v1.2: SPSS Modeler for Clickers and Visual Coders*



Machine Learning & Data Science



# Introducing IBM Data Science Experience Local



IBM Data Science Experience

## Community

- Find tutorials and datasets
- Connect with other Data Scientists
- IBM ML Hub for expert assistance
- Open Source evangelism
- Fork and share projects, samples

## Open Source

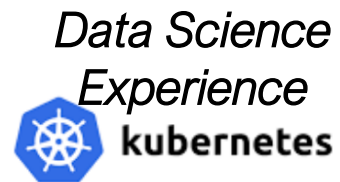
- Code in Scala/Python/R/SQL
- Zeppelin & Jupyter Notebooks
- RStudio IDE
- Anaconda distribution
- Add your favorite libraries

## IBM Added Value

- Projects and Version Control , for dev->test->prod continuous engineering
- Relational & Hadoop data sources connectivity
- Machine Learning & Deep Learning - manage/monitor & deploy models
- Spark-in-DSX and Remote Spark (Hadoop) as well as Python & R based Analytics, ML.
- Publish notebooks and other assets, Host R Shiny Apps, schedule jobs
- Compute Elasticity support, manage CPU/GPU & memory resources
- Data Science Elite team



# DSX is an Open Platform



- ✓ Add your favorite libraries
- ✓ Publish Open APIs for secure ML applications





Learn

Connect to Enterprise data sources easily

Collaborate

Working on cluster safer than desktops for leader

Safe behind the firewall

Big SQL, Db2 (warehouse/z/LUW),  
Hive for HDP, HDFS for HDP  
Hive for Cloudera (CDH)  
HDFS for Cloudera (CDH)  
Informix, Netezza, Oracle

## Community

Notebooks View All >

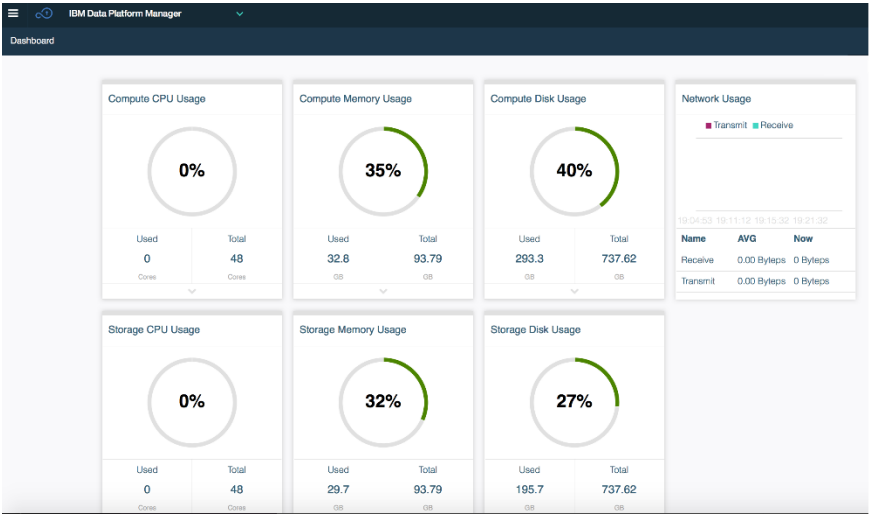
<p><b>NOTEBOOK</b></p> <p>Sudoku</p> <p>AUTHOR IBM</p> <p>DATE Mar 21, 2017</p> <p>TOPIC Leisure</p> <p>DATA SOURCE Self-Contained</p>	<p><b>NOTEBOOK</b></p> <p>How to make targeted offers to customers</p> <p>AUTHOR IBM</p> <p>DATE Mar 07, 2017</p> <p>TOPIC Economy &amp; Business</p> <p>DATA SOURCE Self-Contained</p>	<p><b>NOTEBOOK</b></p> <p>The Unit Commitment Problem</p> <p>AUTHOR IBM</p> <p>DATE Mar 07, 2017</p> <p>TOPIC Economy &amp; Business</p> <p>DATA SOURCE Self-Contained</p>	<p><b>NOTEBOOK</b></p> <p>Use decision optimization to help a sports...</p> <p>AUTHOR IBM</p> <p>DATE Mar 02, 2017</p> <p>TOPIC Leisure</p> <p>DATA SOURCE Self-Contained</p>
<p><b>NOTEBOOK</b></p> <p>Use Spark for R to load data and run SQL...</p> <p>AUTHOR IBM</p> <p>DATE Feb 17, 2017</p> <p>TOPIC Transportation</p> <p>DATA SOURCE Self-Contained</p>	<p><b>NOTEBOOK</b></p> <p>Use Spark for Python to load data and run...</p> <p>AUTHOR IBM</p> <p>DATE Feb 17, 2017</p> <p>TOPIC Transportation</p> <p>DATA SOURCE External</p>	<p><b>NOTEBOOK</b></p> <p>Use Spark for Scala to load data and run SQL...</p> <p>AUTHOR IBM</p> <p>DATE Feb 17, 2017</p> <p>TOPIC Transportation</p> <p>DATA SOURCE External</p>	<p><b>NOTEBOOK</b></p> <p>Learn basics about notebooks and Apache Spark</p> <p>AUTHOR IBM</p> <p>DATE Jan 17, 2017</p> <p>TOPIC Environment</p> <p>DATA SOURCE External</p>





- DSX Local simplifies distribution of team work based on skills
- DSX Local increases knowledge sharing and knowledge retention
- Currently based on open source notebooks, productivity tools in the future
- DSX Local simplifies cluster management for teams

The screenshot shows the DSX Local interface. At the top, there's a navigation bar with 'Projects > dsx'. Below this, a dropdown menu is open, showing options: 'Commit', 'Pull project', and 'Push project'. To the right of the dropdown, there are buttons for 'Export', 'Create notebook', 'Add model', and 'Add data set'. Further right, there's a button for 'Add new collaborator'. Below these buttons, there are statistics: '10 Assets', '2 Environments', '2 Jobs', '1 Data Sources', and '1 Collaborator'. The main content area shows 'dsx' with a description 'Description not available.' and a 'Date created' of 'Thu Oct 19 2017'.





# Data Science is a team sport



IBM Data Science Experience Local 43 Trial Days Left

Projects > Project Zen

**Project Zen**  
This project includes a few notebooks and related datasets to get you started with DSX Local.

**Date created**  
Sun Feb 18 2018

**11** Assets  
**5** Environments  
**4** Jobs  
**0** Data Sources  
**1** Collaborator

## Collaborators



admin

## Recent Assets

NAME	ASSET TYPE	LAST MODIFIED
01-Introduction to Spark and Notebooks - Python	Jupyter Notebook	2/21/2018, 12:37:01 PM
03-Decision tree churn analysis	Jupyter Notebook	2/20/2018, 6:44:57 AM
Churn Model for Telco-model-evaluation-1518983543678	scripts	2/18/2018, 11:52:43 AM
Telco Churn Prediction Model-model-evaluation-1518983276715	scripts	2/18/2018, 11:48:38 AM
Telco Churn Prediction Model-model-evaluation-1518983196960	scripts	2/18/2018, 11:48:56 AM

IBM Data Science Experience Local 43 Trial Days Left

Projects > Project Zen > All

All Notebooks RStudio Models Scripts Data Sets Other Files Published Assets

**Notebooks** view all (3) + add notebook

NAME	STATUS	ENVIRONMENT	TOOL	LAST MODIFIED
01-Introduction to Spark and Notebooks - Python		Jupyter with Python 2.7, Scala 2.11, R 3.4.1	JUPYTER	02-21-2018
03-Decision tree churn analysis		Jupyter with Python 2.7, Scala 2.11, R 3.4.1	JUPYTER	02-20-2018
02-Insights from New-York car accident reports		Jupyter with Python 2.7, Scala 2.11, R 3.4.1	JUPYTER	02-18-2018

**RStudio** view all (0) + open RStudio

NAME	TYPE	LAST MODIFIED
you have no rstudio files		

**Models** view all (1) + add model

NAME	TYPE	STATUS	LAST MODIFIED
Telco Churn Prediction Model v2	spark-2.0	trained	18 Feb 2018, 11:45 AM

**Scripts** view all (4) + add script

Collaborate within git-backed projects





Facilitate creation of machine learning models

Facilitate deployment of models as API endpoints

Automation of Batch Scoring, Training and Evaluation scripts as schedulable jobs

GIT integration to collaborate with engineers in their favorite environment

Publish content to others in pdf / html / R-Shiny app

The image displays two screenshots of the IBM Data Science Experience interface. The top screenshot shows the 'Select a technique' step in an ML project, with a sidebar for 'Select Data', 'Prepare', 'Train', and 'Evaluate'. The bottom screenshot shows the 'ML deployment' step, featuring an 'Input' form with fields for 'x' (value 5) and 'y' (value 3), and a 'Result' section displaying a pie chart with four segments labeled 0, 1, 2, and 3. Segment 1 is highlighted with a label '1: 04%'.



# Essential tools for Data Scientists



IBM Data Science Experience Local

Projects > Project Zen > 02-Insights from New-York car accident reports

File Edit View Insert Cell Kernel Widgets Help

Code CellToolbar

```
#adjust more settings
plt.title('Motor Vehicle Collisions in New York City by borough', size=20)
plt.xlim((-74.26,-73.7))
plt.ylim((40.5,40.92))
plt.xlabel('Longitude',size=20)
plt.ylabel('Latitude',size=20)
plt.show()
```

Motor Vehicle Collisions in New York City by borough

Manhattan  
Bronx  
Brooklyn  
Staten Island  
Queens

Jupyter notebook Environment  
Python 2.7/3.5 with Anaconda  
Scala, R

IBM Data Science Experience Local

Projects > Project Zen > Bank Analysis

Interpreters

Bank Analysis

```
val year = 1999 // year
if (r(2) != "") {
  val month = r(2)
} else {
  //...and peel out the month
  val month = "00"
}

//format the date
val date = r(2) + "-" + get(r(3)) + "-" + get(r(4)) + " " + get(r(5)) + "T" + get(r(6)) + "Z" + ".00"

//extract interesting earthquake data: id, date, depth, magnitude, "uncorrected" magnitude
earthquakeC
r(0).toString, // id
date,
get(r(9)).toDouble,
get(r(10)).toDouble,
get(r(11)).toDouble, // depth
get(r(12)).toDouble, // mag
get(r(13)).toDouble // mag unc
}

}

val earthquake = globalInstrumentalCatalogData.filter(!_.startsWith("#")).map{e =>
  val f = e.split(",")
  earthquakeC
r(2).trim,
r(0).trim,
r(3).toDouble,
r(2).toDouble,
r(7).toDouble,
r(8).toDouble,
r(11).toDouble
}
}.union(historical).toDF //make a dataframe
//make a table to query
earthquake.registerTempTable("eq")
```

Zeppelin notebook  
with Python 2.7 with Anaconda

R Studio Environment with >  
300 packages, R Markdown, R  
Shiny

IBM Data Science Experience Local

Projects > Project Zen > RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Run App

```
1 #
2 # This is a Shiny web application. You can run the application by clicking
3 # the 'Run App' button above.
4 #
5 # Find out more about building applications with Shiny here:
6 #
7 # http://shiny.rstudio.com/
8 #
9
10 library(shiny)
11
12 # Define UI for application that draws a histogram
13 ui <- fluidPage(
14
15   # Application title
16   titlePanel("Old Faithful Geyser Data"),
17
18   # Sidebar with a slider input for number of bins
19   sidebarLayout(
20     sidebarPanel(
21       sliderInput("bins",
22         "Number of bins:",
23         min = 1,
24         max = 50,
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Console

```
> shiny::runApp("SampleShiny")
Loading required package: shiny
Listening on http://127.0.0.1:324
```



# Self serve Compute Environments



Servers/IDEs - lifecycle easily controlled by each Data Scientist

## Environments (5)

NAME	TYPE	STATUS	CPU CORES	GPUS	GB MEMORY ALLOCATED	DATE STARTED
Jupyter with Python 2.7, Scala 2.11, R 3.4.1	Jupyter	Running	1.0	0	5.0	18 Feb 2018, 10:53 AM
Jupyter with Python 3.5 for GPU	Jupyter	Stopped	1.0	0	5.0	
Jupyter with Python 3.5	Jupyter	Stopped	1.0	0	5.0	
RStudio with R 3.3.2	RStudio	Stopped	2.0	0	5.0	
Zeppelin with Anaconda2, Python 2.7	Zeppelin	Stopped	1.0	0	5.0	

## Edit Jupyter environment

### Image \*

Jupyter with Python 2.7, Scala 2.11, R 3.4.1

### Description

Jupyter Notebook Server with Anaconda2 4.4, Python 2.7, Scala 2.11, R 3.4.1

### CPU cores \*

1.0  8.0

### Memory \*

5  GB (2.0 minimum)

Total allocated: **31.3 GB RAM**  
■ This runtime: **5.0 GB RAM** ■ Other runtimes: **15.0 GB RAM** ■ Available: **11.3 GB RAM**

Self-serve reservations of compute resources

Worker compute resources – for batch jobs run on-demand or on schedule

## Workers 3

NAME	TYPE	CPU CORES	GPUS	GB MEMORY ALLOCATED	STATUS
Jupyter 4.4 and Python 3.5 for GPU	Python 3.5 with GPU	1	0	5	
Jupyter and Python 3.5	Python 3.5	1	0	5	
Jupyter, Python 2.7, Scala 2.11, R 3.4.1	Python 2.7	1	0	5	

Environments are essentially Kubernetes pods – with High Availability & Compute scale-out baked in (load-balancing/auto-scaling is being planned for a future spring)



# Extend ..

## – Roll your own Environments



- Add libs/packages to the existing Jupyter, Rstudio , Zeppelin IDE Environments or introduce new Job “Worker” environments

<https://content-dsxlocal.mybluemix.net/docs/content/local/images.html>

DSX Local provides a Docker Registry (and replicated for HA) as well.

These images get managed by DSX and is used to help build out custom Environments

Image management										
Image list										
NAME	TAG	AUTHOR	TYPE	BASE	DATE UPLOADED	VALIDATED	VALIDATION CHANGE DATE	VALIDATION CHANGE USER	DESCRIPTION	RUNNING
hdzzeppelin-dsx-d8a2ls2x	v1.0.84	IBM	zeppelin	n/a	n/a	validated	n/a	n/a	Zeppelin Notebook Server 0.7.3 with Anaconda2 4.4, Python 2.7	No
jupyter-dsx-d8a2ls2x	v1.0.98	IBM	jupyter	n/a	n/a	validated	n/a	n/a	Jupyter Notebook Server with Anaconda2 4.4, Python 2.7, Scala 2.11, R 3.4.1	Yes
jupyter-dsx-d8a3ls2x	v1.0.91	IBM	jupyter-py35	n/a	n/a	validated	n/a	n/a	Jupyter Notebook Server with Anaconda3 4.2, Python 3.5	No
jupyter-gpu-py35	v1.0.35	IBM	jupyter-gpu-py35	n/a	n/a	validated	n/a	n/a	Jupyter Notebook Server 4.4 with Python 3.5 for GPU (CUDA 8.0)	No
privatecloud-rstudio	v3.13	IBM	rstudio	n/a	n/a	validated	n/a	n/a	RStudio Server 1.0 with R 3.3.2	No



Jobs Workers

Search by job name

Jobs – trigger on-demand or by a schedule. such as for Model Evaluations, Batch scoring or even continuous (re-) training

Jobs 4 create job

NAME	TYPE	SOURCE ASSET	CREATOR	DATE CREATED	SCHEDULED TO RUN	LAST RUN	STATUS
Churn Model for Telco-1518983543678	Model evaluation	scripts/Churn Model for Telco-model-evaluation-1518983543678.py	admin	18 Feb 2018, 11:52 A M	On demand	18 Feb 2018, 11:52 A M	
Telco Churn Prediction Model-1518983276715	Model evaluation	scripts/Telco Churn Prediction Model-model-evaluation-1518983276715.py	admin	18 Feb 2018, 11:48 A M	On demand	18 Feb 2018, 11:48 A M	
Telco Churn Prediction Model-1518983196960	Model evaluation	scripts/Telco Churn Prediction Model-model-evaluation-1518983196960.py	admin	18 Feb 2018, 11:46 A M	On demand	18 Feb 2018, 11:46 A M	
Telco Churn Prediction Model-1518982994199	Model evaluation	scripts/Telco Churn Prediction Model-model-evaluation-1518982994199.py	admin	18 Feb 2018, 11:43 A M	On demand	18 Feb 2018, 11:43 A M	

<https://us45001.iacs.ibm.com>

user01 - user20 / IBMdsxl!

[https://github.com/jpatter/DSX\\_Local\\_Workshop\\_V12/](https://github.com/jpatter/DSX_Local_Workshop_V12/)



# Deploy, monitor and manage



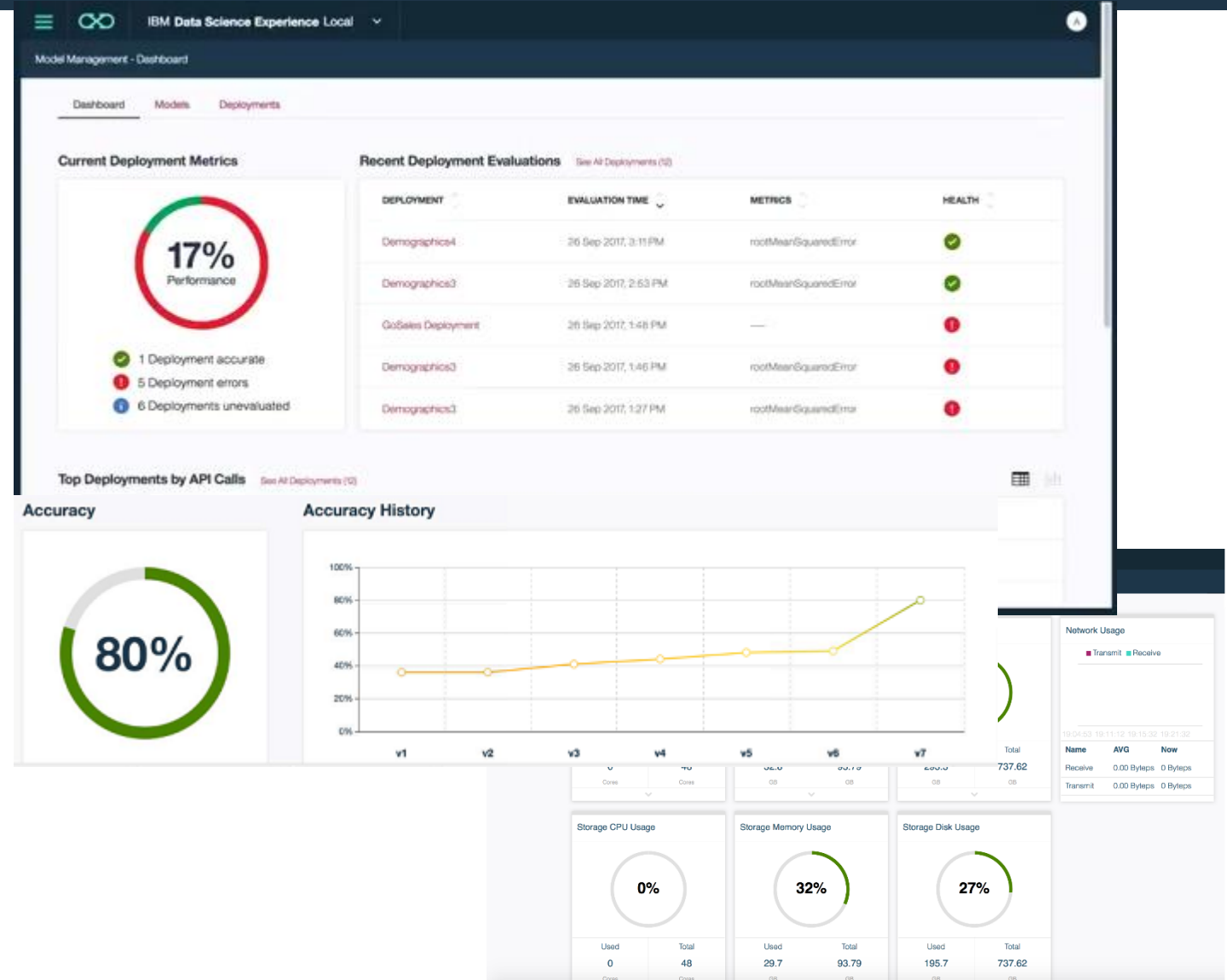
Monitor models through a dashboard

Model versioning, evaluation history

Publish versions of models, supporting dev/stage/production paradigm

Monitor scalability through cluster dashboard

Adapt scalability by redistributing compute/memory/disk resources





# What's new in v1.2.1



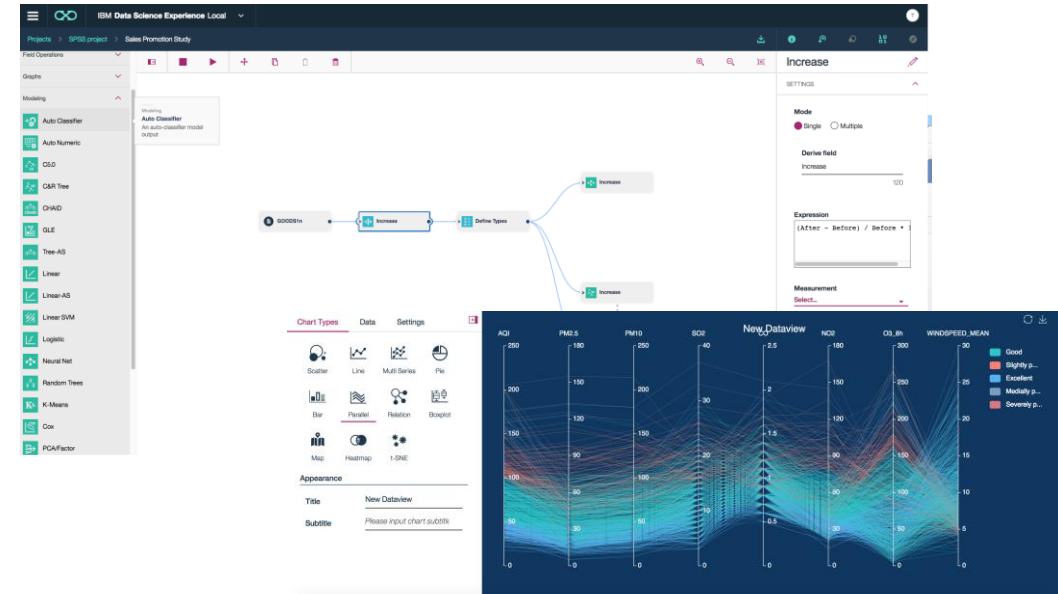
- Cognos Dashboard Embedded
- Watson Explorer oneWEX
- Spark Canvas (beta)
- Python Jupyter Terminals
- Enhanced Data Refinery
- Enhanced Data Connection
- Model Groups
- Enhanced Git support
- Model Management and Deployment Enhancements
- 



# SPSS Modeler for DSX



- Pain Points:
  - Lack of skills around coding
  - Environment for quick prototyping/experimentation
  - Easy path to deployment from visual productivity
- 
- Value Proposition:
  - Visual productivity tool around data science
  - Quicker time to value
  - Inclusion of full-fledged data preparation and many machine learning algorithms



- Features:
- Newly rebuilt interface with improved navigation and ease of use
- Totally new interactive visualization
- Ability to deploy results in Model Management and Deployment



# Decision Optimization for DSX

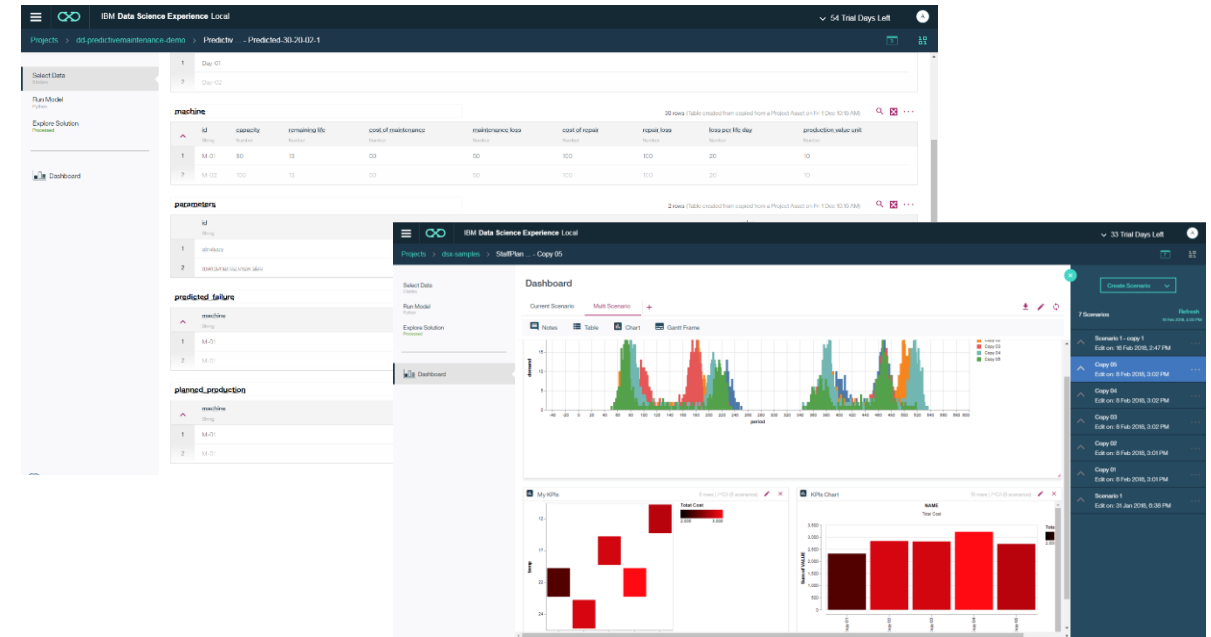


- **Pain Points:**

- Operationalize Data Science and drive higher ROI
- One platform for predictive and prescriptive tools
- Enable team on prescriptive tools and approaches
- Productivity of expensive resources

- **Value Proposition:**

- Transform insights from ML into actions using DO
- Add-on to DSX local with a mix of predictive and prescriptive in unified environment
- Community cards, tutorials, etc.
- Validate models using dashboards and execute what-if



- **Features:**

- Model building workflow
- What-if analysis and dashboards
- Beta: Modeling assistant (limited domains)



# MMD: Introducing Manage Deployments feature



- Supports Dev->Test->Staging->Production CI/CD paradigm
  - Access control & Separation of duties
  - Enables a clear path for promotion of assets to production, while enabling Development to continue on separately.
- Dev: (DSX Local Projects)
  - Build & test out assets
    - Notebooks
    - Python & R Scripts
      - Run as a Job in DSXL cluster or against the Hadoop Integration Service
      - REST based tests against User authored Scripts run as a Web Service  
(invoke Python or R functions – for example for Custom “Scorers”, to infuse custom data prep)
    - Models – Train & Test from scripts/notebooks
    - Shiny Apps – Build & Preview
    - SPSS Modeler Flows - Design & Test
- A Project “Release” **Tag**
  - identifies a stable checkpoint in the history of the Project that marks its (possible) *promotion* to Production
  - implemented as a git tag (also applies to github/bitbucket repositories or exported project .zip/.tar.gz archives)



# MMD: Introducing Manage Deployments feature



- Production with “MMD”: either in the same cluster as Dev or use a separate cluster
  - Pull from the release git tag(or export/import)
  - Create deployments
    - Create online scoring end-points for Models (or versions of a model)
    - Schedule Batch Jobs such as SPSS Modeler Flows, Python/R scoring or evaluations, against Hadoop etc.
    - Schedule execution & externalize Jupyter Notebooks
    - Externalize Shiny Apps
    - Expose User authored Scripts as a Web Service
      - API based access to Python or R functions –such as Custom Scorers with custom data prep
  - Define resource requirements & SLAs for Deployments
    - Per deployment – choose to “reserve” compute (or acquire when needed)
    - Load balancing and latency - choose to run replicas of Scoring servers, Scripts-as-Web Services for high throughput needs
- Variations:
  - Export a “Deployment Manifest” from a “Staging” MMD and Import - modify in a Production MMD
  - Use the same MMD instance for both staging and production – just create two “Project Releases” with different URL routes.



# Deployment manager - Project Releases



IBM Deployment Manager

59 Trial Days Left

A

Project releases

Search by project release name

Project releases 5

project release

○ ashwintest

CREATOR  
admin

DEPLOYMENTS  
0

SOURCE  
Demo v1.6

LAST MODIFIED  
12 Apr 2018, 8:34 PM

○ pytf

CREATOR  
--

DEPLOYMENTS  
0

SOURCE  
pytf v1.0

LAST MODIFIED  
12 Apr 2018, 5:45 PM

● AnalyticsRelease

CREATOR  
admin

DEPLOYMENTS  
4

SOURCE  
Demo v1.6

LAST MODIFIED  
12 Apr 2018, 5:45 PM

Lock  
Update  
Export settings  
Delete

● zlati

CREATOR  
admin

DEPLOYMENTS  
2

SOURCE  
test v1.9

LAST MODIFIED  
12 Apr 2018, 5:36 PM

● testDemo

CREATOR  
admin

DEPLOYMENTS  
2

SOURCE  
Demo v1.1.3.01

LAST MODIFIED  
12 Apr 2018, 4:54 PM

Current git tag

Project releases Deployed & (delta)updatable



# Bring in a new “release” to production



## Create project release

From DSX   From repository   From file

**Name \***  
Demo release ✓  
88

**Route \* ⓘ**  
demoapp ✓  
19

**Source project \***  
Demo ▼

**Tag \***  
v1.6 ▼

New Releases  
- from a “Source”  
Project created  
from a .tar.gz  
package

New Releases  
- from a “Source”  
Project in the  
same cluster

Project releases > Create project release

## Create project release

From DSX   From repository   From file

**Name \***  
Demo release ✓  
100

**Route \* ⓘ**  
demoapp ✓  
26

**Location \***  
☒ GitHub  
☐ BitBucket

**Repository URL \***  
https://github.ibm.com/PrivateCloud/dsx-samples !  
Not a valid git repository URL.

**Token \***  
Select a token ▼

**Release tag \***  
v1.7

New Releases  
- from a “Source”  
Project pulled  
from  
github/bitbucket



# Expose a ML model via a REST API



## Deploy CreditCardDefaultModel as a web service

**Name \***  
ccard-risk ✓ 16

**URL**  
https://apollo-113-master-1.fyre.ibm.com/dmodel/v1/testdemo/pyscript/ccard-risk

**Model version \***  
Use latest version ▼

**Web service environment \***  
Python 2.7 - Script as a Service ▼

**Reserve resources**  
☒

**CPU cores \***  
1

**GB Memory \***  
0.5

**Replicas**  
2 ▲▼

scoring end-point

Model pre-loaded into memory  
inside scoring containers

pick a version to  
expose  
(multiple  
deployments are  
possible too..)

Optionally reserve  
compute

replicas for load  
balancing





# Expose Python and R scripts as a Web Service



Custom scripts can be externalized as a REST service - say for custom prediction functions

IBM Deployment Manager

Project releases > testDemo > Deploy user-script.r as a web service

## Deploy user-script.r as a web service

**Name \***  
obesity-risk-custom-scorer 0

**URL**  
https://apollo-113-master-1.fyre.ibm.com/dsvc/v1/testdemo/rscript/obesity-risk-custom-scorer

**Web service environment \***  
R 3.4.3 - Script as a Service

**Environment variables**   
VARIABLE\_1=value 1 100

**Reserve resources**  
☐

**Replicas**  
1



# Deploy a script as a schedulable Job



Deploy CreditCardDefaultModel-batch-scoring-1523570886065.py as a job

Scripts can be deployed as a job..

**Name \***  
churn-batch-score ✓

**URL**  
<https://apollo-113-master-1.fyre.ibm.com/djob/v1/testdemo/churn-batch-score>

**Description**

URL exposed for the job for external triggering

Can be run against a DSX Hadoop Integration Service environment

**Type \***  
Batch scoring

**Worker \***  
Jupyter with Python 2.7, Scala 2.11, R 3.4.3

**Target host \***  
Local instance

Local instance
cdhkanchedge
doge1

And use the convenient API panel to understand how to invoke the job..

Overview API

### ccbatch

<https://apollo-113-master-1.fyre.ibm.com/djob/v1/ar1/ccbatch/trigger>  
Deployment token

TYPE	ASSET	ALLOCATED CPU	ALLOCATED MEMORY	TARGET HOST
Job	CreditCardDefaultModel-batch-scoring-1523570886065	Unallocated	Unallocated	Local instance

**Request** **Start** **Response** [generate code](#)

Environment variables

Command line arguments

Start

Status

Stop



# Deploy a Notebook or R Shiny App



IBM Deployment Manager

Project releases > zlati > Deploy retirement as an app

## Deploy retirement as an app

**Name \***  
retirementapp 13

**URL**  
<https://apollo-113-master-1.fyre.ibm.com/dapp/v1/zlati/shinyapp/retirementapp/>

**Environment \***  
Shiny Server  
Used to serve R shiny apps

**Reserve resources** ⓘ  
☒

**CPU cores \***  
1

**GB Memory \***  
4

**Shared with \***  
☒ Anyone with the link  
☐ Any authenticated user  
☐ Deployment admin

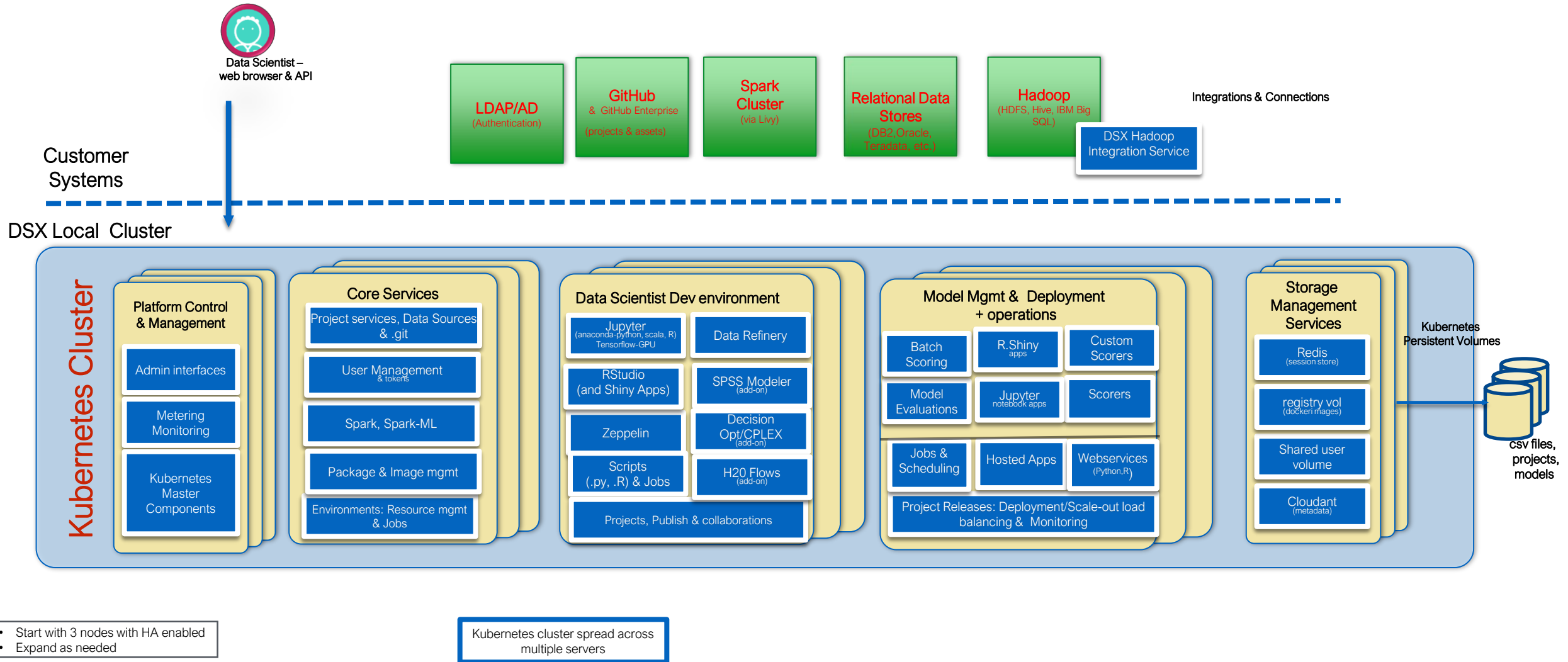
Host a notebook  
or a Shiny App

Optionally reserve  
compute

Define visibility  
rules



# DSX Local Architecture overview



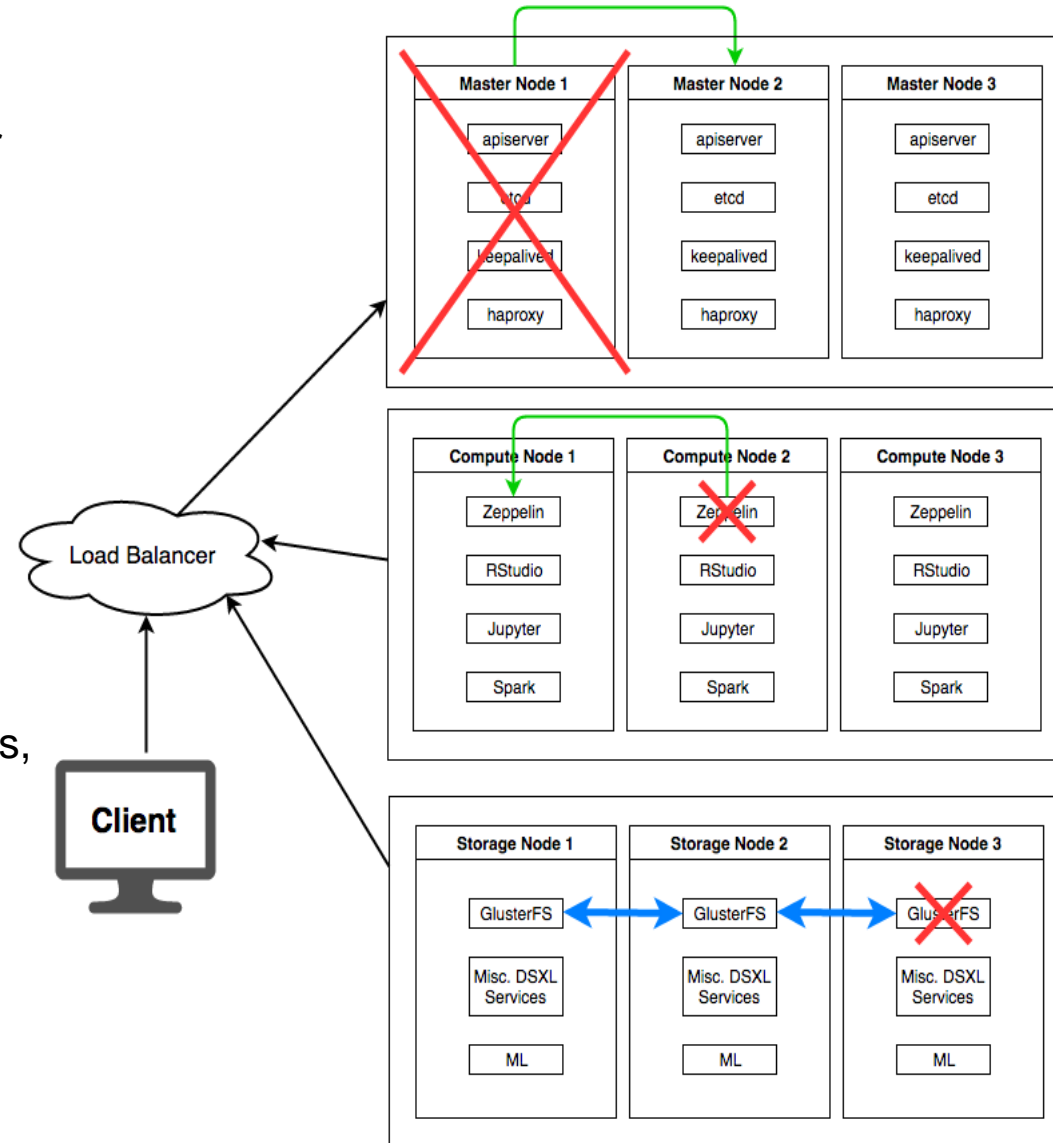


- **Platform HA**

- By default, the cluster is configured for HA: minimum 3-node
- Can tolerate failure of
  - 1 node in a 3-node configuration
  - 4 nodes in a 9-node configuration

- **Service HA**

- For services that are deployed as pods, Kubernetes will monitor and redeploy
- No session failover, which means that some services may be down for a few minutes





# Roadmap Highlights

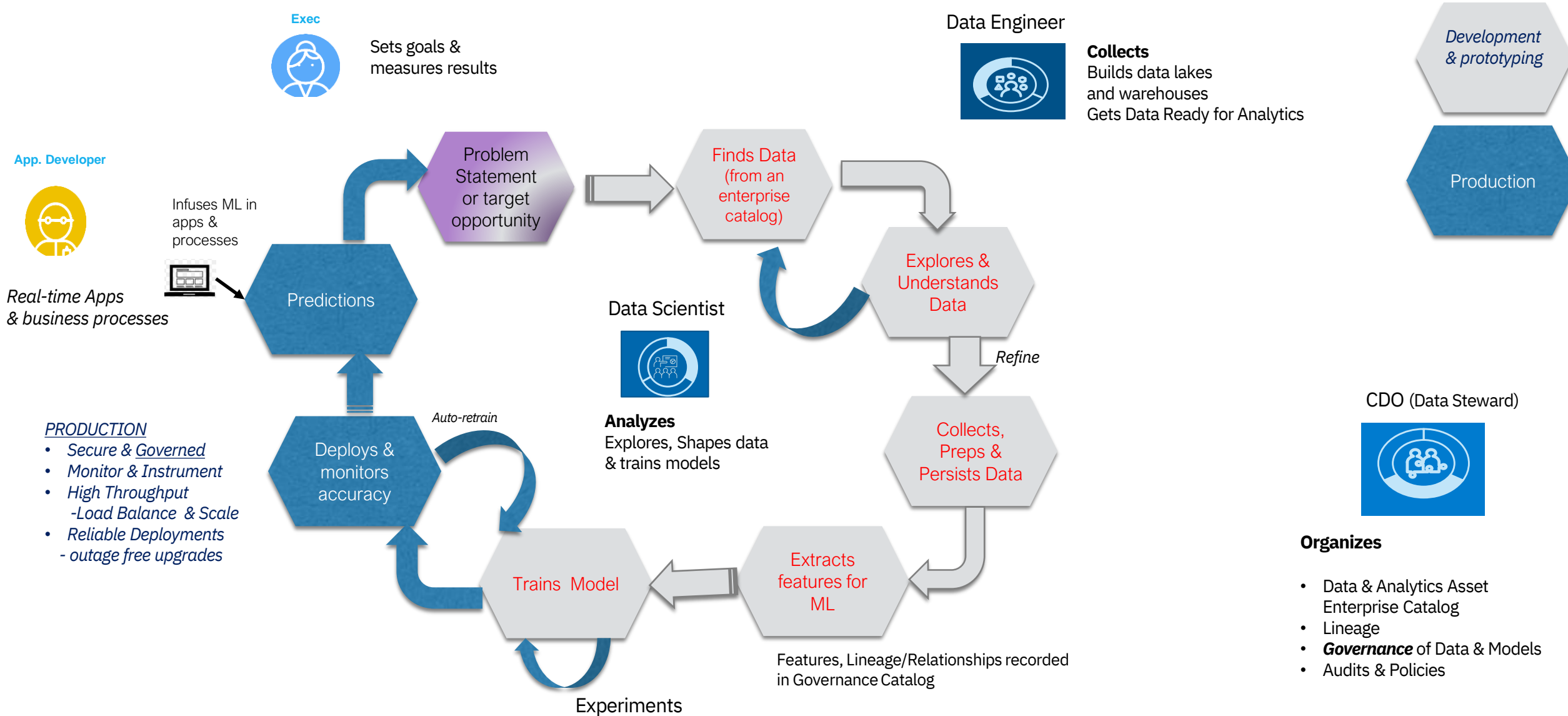




Jan 2018 (Delivered)	Mar 2018 (Delivered)	Q2 2018	2h 2018
<p>New data connections to ease data access</p> <ul style="list-style-type: none"> <li>– Support for IBM BigSQL on HDP as a data source</li> <li>– Support for Cloudera data sources</li> </ul> <p>Enhanced ML capabilities</p> <ul style="list-style-type: none"> <li>– Train, Deploy and Monitor Scikit-learn, xgboost, Tensorflow+Keras and spark-ml models</li> <li>– Automation - Batch Scoring and Evaluation scripts as schedulable jobs</li> <li>– Model versioning and evaluation history</li> <li>– Publish models</li> </ul> <p>New Deep Learning and GPU support</p> <ul style="list-style-type: none"> <li>– Runtime Environments can now be associated with GPU(s)</li> <li>– Train Tensorflow+Keras models or execute other jobs that need GPUs in these Environments.</li> </ul> <p>Collaborate</p> <ul style="list-style-type: none"> <li>– Publish and share the URL to any content (pdf, html, csv etc.) with other users</li> </ul>	<p>IBM SPSS Modeler</p> <ul style="list-style-type: none"> <li>– Refreshed GUI</li> <li>– High priority nodes</li> </ul> <p>Decision Optimization for Data Science integration</p> <p>Model Management and Deployment package</p> <ul style="list-style-type: none"> <li>– Deploy/Manage model versions</li> <li>– Deploy SPSS Modeler Streams + batch scoring</li> <li>– Deploy Notebooks, Scripts &amp; Shiny Apps</li> <li>– Add support for R based Models</li> <li>– Management access control</li> </ul> <p>Hadoop</p> <ul style="list-style-type: none"> <li>– Python, R job push-down to Yarn in a secure Hadoop cluster (in-addition to Livy-Spark)</li> </ul>	<p>SPSS Modeler enhancements</p> <ul style="list-style-type: none"> <li>– Additional nodes available</li> <li>– Extensions and scripting available with new GUI</li> </ul> <p>Model Management and Deployment enhancement</p> <ul style="list-style-type: none"> <li>– SPSS Modeler Streams – real-time scoring</li> <li>– A/B testing &amp; experimentation</li> <li>– Scale-out Deployments - replicas &amp; load-balancing(Scoring servers &amp; Shiny Apps etc.)</li> </ul> <p>Apache Atlas, IGC and Apache Ranger integration</p> <p>Add-on/Extensions</p> <ul style="list-style-type: none"> <li>– pick from a “market-place”, mix-n-map Environments easily</li> </ul>	<p>WEX integration - Text analytics – including support for ML annotators, classifiers, and custom annotators</p> <p>HDP/Yarn 3.1’ s Kubernetes native support provides for Data Plane and DSX running directly on HDP-Kubernetes</p> <p>Enable partner and third-party tool integrations</p> <p>Support DL frameworks/libraries: DL4J, Caffe2, CNTK with GPU support.</p>



# Key scenario: Governed Enterprise Data Science







- Efficient Compute Resource Management for large-scale Analytics, Machine Learning and Deep Learning workloads
  - Enable Data Scientists to *procure* resources from a shared compute “grid” for any kind of activity – from interactive notebooks & IDEs to training Jobs or scheduled scripts and Apps.
  - All compute manifested as Docker containers/Kubernetes pods
- HDP/Yarn as *the* Resource Manager
  - Enable *all* workloads, whether Map Reduce or Spark Jobs or DSX/ML activities to be uniformly handled by the HDP/Yarn scheduler.
  - Manage Queue Priorities, balancing of workloads and scale-out for the whole cluster providing best utilization of all resources.

➤ Yarn and Kubernetes - the best of both worlds !

